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 可以繼續正常使用服務。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？

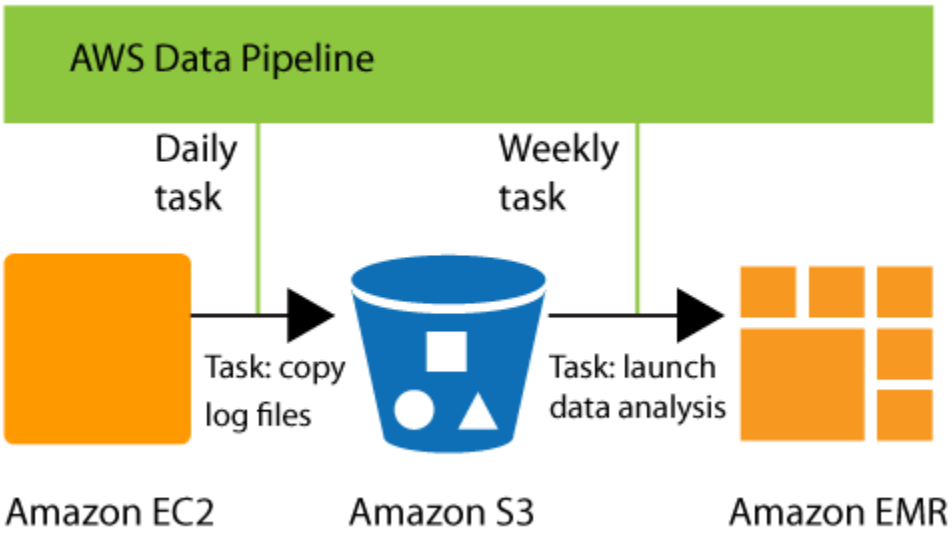
> **Note**
>
> AWS Data Pipeline 服務處於維護模式，未規劃新功能或區域擴展。若要進一步了解並了解如何遷移現有的工作負載，請參閱 從 遷移工作負載 AWS Data Pipeline。

AWS Data Pipeline 是一種 Web 服務，可用來自動化資料的移動和轉換。使用 AWS Data Pipeline，您可以定義資料驅動型工作流程，以便任務可以依賴於成功完成先前的任務。您可以定義資料轉換的參數，並 AWS Data Pipeline 強制執行您設定的邏輯。

下列 元件共同 AWS Data Pipeline 運作以管理您的資料：

- 「管道定義」指定您資料管理的商業邏輯。如需詳細資訊，請參閱管道定義檔案語法。
- 管道透過建立 Amazon EC2 執行個體來執行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。
- 任務執行器輪詢任務，然後執行這些任務。例如，Task Runner 可以將日誌檔案複製到 Amazon S3 並啟動 Amazon EMR 叢集。任務執行器已安裝並在管道定義建立的資源上自動執行。您可以撰寫自訂任務執行器應用程式，也可以使用 提供的任務執行器應用程式 AWS Data Pipeline。如需詳細資訊，請參閱任務執行器。

例如，您可以使用 AWS Data Pipeline 每天將 Web 伺服器的日誌封存至 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。 會 AWS Data Pipeline 排程每日任務以複製資料，以及每週任務以啟動 Amazon EMR 叢集。AWS Data Pipeline 也會確保 Amazon EMR 會等待最後一天的資料上傳到 Amazon S3，再開始分析，即使上傳日誌時發生無法預期的延遲。



目錄

從 遷移工作負載 AWS Data Pipeline

AWS 已在 2012 年推出 AWS Data Pipeline 此服務。當時，客戶正在尋找一種服務，以協助他們使用各種運算選項在不同資料來源之間可靠地移動資料。現在，還有其他 服務可為客戶提供更好的體驗。例如，您可以使用 AWS Glue 執行和協調 Apache Spark 應用程式、使用 AWS Step Functions 協助協調 AWS 服務元件，或使用 Amazon Managed Workflows for Apache Airflow (Amazon MWAA) 協助管理 Apache Airflow 的工作流程協調。

本主題說明如何從 遷移 AWS Data Pipeline 至替代選項。您選擇的選項取決於您目前的工作負載 AWS Data Pipeline。您可以將 的一般使用案例遷移 AWS Data Pipeline 至 AWS Glue AWS Step Functions 或 Amazon MWAA。

將工作負載遷移至 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。其中包括撰寫、執行中任務和協調工作流程的工具。使用 AWS Glue，您可以探索並連線至超過 70 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。

我們建議您在下列 AWS Glue 情況下將 AWS Data Pipeline 工作負載遷移至 ：

- 您正在尋找支援各種資料來源的無伺服器資料整合服務、撰寫包括視覺化編輯器和筆記本的界面，以及進階資料管理功能，例如資料品質和敏感資料偵測。
- 您的工作負載可以遷移到 AWS Glue 工作流程、任務 (Python 或 Apache Spark) 和爬蟲程式 （ 例如，您現有的管道建置在 Apache Spark 之上 ）。
- 您需要單一平台來處理資料管道的所有層面，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 主控台上的預先定義範本建立，例如將 DynamoDB 資料表匯出至 Amazon S3，而您正在尋找相同的用途範本。
- 您的工作負載不依賴特定的 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 對於爬蟲程式 （ 探索資料 ） 和 ETL 任務 （ 處理和載入資料 ） 以每小時費率計費。 AWS Glue Studio 是 AWS Glue 資源的內建協同運作引擎，免費提供。進一步了解 定價。 AWS Glue

將工作負載遷移至 AWS Step Functions

AWS Step Functions 是一種無伺服器協同運作服務，可讓您為業務關鍵型應用程式建立工作流程。使用 Step Functions，您可以使用視覺化編輯器來建置工作流程，並直接與超過 250 種 AWS 服務的 11，000 個動作整合，例如 AWS Lambda、Amazon EMR、DynamoDB 等。您可以使用 Step Functions 來協調資料處理管道、處理錯誤，以及使用基礎 AWS 服務的限流限制。您可以建立工作流程來處理和發佈機器學習模型、協調微服務，以及控制 AWS 服務 AWS Glue，例如建立擷取、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。

同樣地 AWS Data Pipeline， AWS Step Functions 是由 提供的全受管服務 AWS。您不需要管理基礎設施、修補工作者、管理作業系統版本更新或類似項目。

我們建議您在下列情況下將 AWS Data Pipeline 工作負載遷移至 AWS Step Functions：

- 您正在尋找無伺服器、高可用性的工作流程協同運作服務。

- 您正在尋找經濟實惠的解決方案，以單一任務執行的精細程度收費。
- 您的工作負載正在協調多個 AWS 其他服務的任務，例如 Amazon EMR AWS Glue、Lambda 或 DynamoDB。
- 您正在尋找具有drag-and-drop視覺化設計工具的低程式碼解決方案，以建立工作流程，而且不需要學習新的程式設計概念。
- 您正在尋找一項服務，該 AWS 服務提供與超過 250 個其他服務的整合，涵蓋超過 11，000 個out-of-the-box動作，並允許與自訂非AWS 服務和活動整合。

AWS Data Pipeline 和 Step Functions 都使用 JSON 格式來定義工作流程。這可讓 將您的工作流程存放在來源控制中、管理版本、控制存取，以及使用 CI/CD 自動化。Step Functions 使用的語法稱為 Amazon State Language，完全以 JSON 為基礎，並允許在工作流程的文字和視覺呈現之間無縫轉換。

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

對於遷移 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。

若要遷移內部部署伺服器、使用者受管 EC2 執行個體或使用者受管 EMR 叢集上的活動，您可以將 SSM 代理程式安裝到執行個體。您可以透過來自 Step Functions 的 AWS Systems Manager Run Command 啟動命令。您也可以從 Amazon EventBridge 中定義的排程啟動狀態機器。

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，會根據執行應用程式所需的狀態轉換次數向您收費。對於快速工作流程，會根據工作流程的請求數量及其持續時間向您收費。進一步了解 AWS Step Functions 定價。

將工作負載遷移至 Amazon MWAA

Amazon MWAA (Apache Airflow 的受管工作流程） 是 Apache Airflow 的受管協同運作服務，可讓您更輕鬆地大規模在雲端中設定和操作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、排程和監控稱為「工作流程」的程序和任務序列。使用 Amazon MWAA，您可以使用 Airflow 和 Python 程式設計語言來建立工作流程，而無需管理基礎基礎設施以實現可擴展性、可用性和安全性。Amazon MWAA 會自動擴展其工作流程執行容量以符合您的需求，並與 AWS 安全服務整合，協助您快速且安全地存取資料。

同樣地 AWS Data Pipeline，Amazon MWAA 是由 提供的全受管服務 AWS。雖然您需要了解這些服務特有的幾個新概念，但您不需要管理基礎設施、修補工作者、管理作業系統版本更新或類似內容。

我們建議您在下列情況下將 AWS Data Pipeline 工作負載遷移至 Amazon MWAA：

- 您正在尋找受管、高可用性的服務來協調以 Python 撰寫的工作流程。
- 您想要轉換到全受管、廣泛採用的開放原始碼技術 Apache Airflow，以獲得最大的可攜性。
- 您需要單一平台來處理資料管道的所有層面，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道協同運作而設計的服務，其功能包括提供可觀測性的豐富 UI、重新啟動失敗的工作流程、回填，以及重試任務。
- 您正在尋找具有 800 多個預先建置的運算子和感應器的服務，涵蓋 AWS 和非AWS 服務。

Amazon MWAA 工作流程使用 Python 定義為有向無環圖 (DAGs)，因此您也可以將其視為原始程式碼。Airflow 的可擴展 Python 架構可讓您建置與幾乎任何技術連線的工作流程。它具有豐富的使用者介面，用於檢視和監控工作流程，並且可以輕鬆地與版本控制系統整合，以自動化 CI/CD 程序。

使用 Amazon MWAA，您可以選擇您目前使用的相同 Amazon EMR 版本 AWS Data Pipeline。

AWS Airflow 環境執行時間的費用加上任何其他自動擴展，以提供更多的工作者或 Web 伺服器容量。進一步了解 Amazon Managed Workflows for Apache Airflow 定價中的定價。

映射概念

下表包含 服務使用的主要概念映射。它將協助熟悉資料管道的人員了解 Step Functions 和 MWAA 術語。

Data Pipeline	連接詞	步驟函數	Amazon MWAA
管道	工作流程	工作流程	直式 acylic 圖形
管道定義 JSON	工作流程定義或 Python 型藍圖	Amazon 狀態語言 JSON	Python 型
活動	任務	狀態和任務	任務 (操作員和感應器)
執行個體	任務執行	執行	DAG 執行
Attempts	重試嘗試	擷取器和重試器	重試
管道排程	排程觸發條件	EventBridge 排程器任務	Cron、時間表、資料感知
管道表達式和函數	藍圖程式庫	Step Functions 內部函數和 AWS Lambda	可擴展的 Python 架構

範例

下列各節列出您可以參考從 遷移 AWS Data Pipeline 至個別 服務的公有範例。您可以參考它們做為範例，並根據您的使用案例更新和測試管道，在個別服務上建置自己的管道。

AWS Glue 範例

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

- 執行 Spark 任務
- 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 DynamoDB
- 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 範例

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

- 管理 Amazon EMR 任務
- 在 Amazon EMR Serverless 上執行資料處理任務
- 執行 Hive/Pig/Hadoop 任務
- 查詢大型資料集 (Amazon Athena、Amazon S3 AWS Glue)
- 使用 Amazon Redshift 執行 ETL 工作流程
- 協調 AWS Glue 爬蟲程式

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

Amazon MWAA 範例

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

- 執行 Amazon EMR 任務
- 為 Apache Hive 和 Hadoop 建立自訂外掛程式

- 將資料從 Amazon S3 複製到 Redshift
- 在遠端 EC2 執行個體上執行 Shell 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 使用下列 服務來存放資料。

- 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。如需詳細資訊，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，讓您輕鬆且符合成本效益地分析大量資料。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、耐用且高度可擴展的物件儲存。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南。

AWS Data Pipeline 使用下列運算服務來轉換資料。

- Amazon EC2 — 提供可調整大小的運算容量，實際上是 Amazon 資料中心的伺服器，用於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。
- Amazon EMR — 可讓您使用 Apache Hadoop 或 Apache Spark 等架構，在 Amazon EC2 伺服器上輕鬆、快速且經濟實惠地分配和處理大量資料。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 管理主控台— 提供可用來存取的 Web 界面 AWS Data Pipeline。
- AWS Command Line Interface (AWS CLI) — 為廣泛的 AWS 服務提供命令，包括 Windows AWS Data Pipeline、macOS 和 Linux 支援 和 。如需安裝 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface。如需 的命令清單 AWS Data Pipeline，請參閱資料管道。

- 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 請求呼叫的低階 APIs。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，您需要根據活動和先決條件的排程執行頻率及其執行位置來支付管道的費用。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

當 AWS Data Pipeline 執行管道時，它會編譯管道元件，以建立一組可執行的 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詳細資訊，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 執行個體類型矩陣。

下表說明 AWS Data Pipeline 支援的執行個體類型。您可以使用 在任何區域中 AWS Data Pipeline 啟動 Amazon EC2 執行個體，包括 AWS Data Pipeline 不支援 的區域。如需 AWS Data Pipeline 支援的區域資訊，請參閱 AWS 區域和端點。

目錄

依 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 AWS Data Pipeline 支援 的區域中，預設 AWS Data Pipeline 使用 的 Amazon EC2 執行個體。

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 AWS Data Pipeline 不支援 的區域中預設 AWS Data Pipeline 啟動的 Amazon EC2 執行個體。

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
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

區域名稱	區域	執行個體類型
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 且可建立的 Amazon EC2 執行個體，如果指定的話。

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 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

Amazon EMR 叢集支援的 Amazon EC2 執行個體

此表格列出 AWS Data Pipeline 支援 的 Amazon EC2 執行個體，如果指定，可以為 Amazon EMR 叢集建立 和 。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

開始之前，請閱讀 的重要概念和元件 AWS Data Pipeline。

目錄

管道定義

管道定義是您傳達業務邏輯的方式 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

從您的管道定義中， AWS Data Pipeline 決定任務、排程任務，並將其指派給任務執行器。如果任務未成功完成， 會根據您的指示 AWS Data Pipeline 重試任務，並在必要時將其重新指派給另一個任務執行器。如果任務重複失敗，您可以設定管道來接收通知。

例如，在您的管道定義中，您可以指定由應用程式產生的日誌檔案在 2013 年每個月封存至 Amazon S3 儲存貯體。 接著 AWS Data Pipeline 會建立 12 個任務，每個任務都會複製一個月的資料值，無論該月是否包含 30、31、28 或 29 天。

您可以透過下列方式建立管道定義：

- 以圖形方式，使用 AWS Data Pipeline 主控台
- 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。

Resources

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，請參閱管道定義檔案語法。

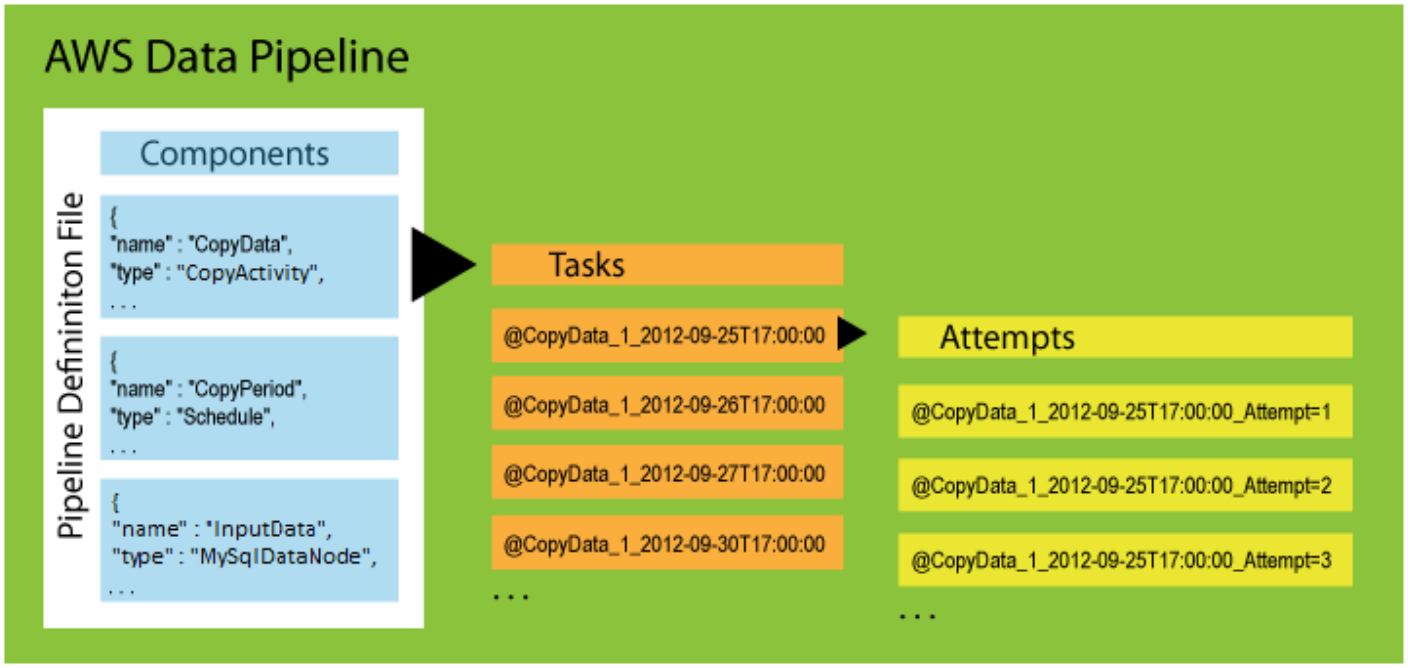
管道元件、執行個體和嘗試

排程管道的相關項目類型有三種：

- 管道元件 — 管道元件代表管道的商業邏輯，並以管道定義的不同區段表示。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。
- 執行個體 — 當 AWS Data Pipeline 執行管道時，它會編譯管道元件以建立一組可執行的執行個體。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道的待辦事項清單。 AWS Data Pipeline 會將執行個體移出任務執行器進行處理。

- 嘗試 — 為了提供強大的資料管理， 會 AWS Data Pipeline 重試失敗的操作。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)。基本上，它是具有 counter. AWS Data Pipeline performs 的執行個體，會使用先前嘗試的相同資源重試，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，請仔細考慮何時適合超過您用來控制重試和相關設定的 AWS Data Pipeline 預設設定。

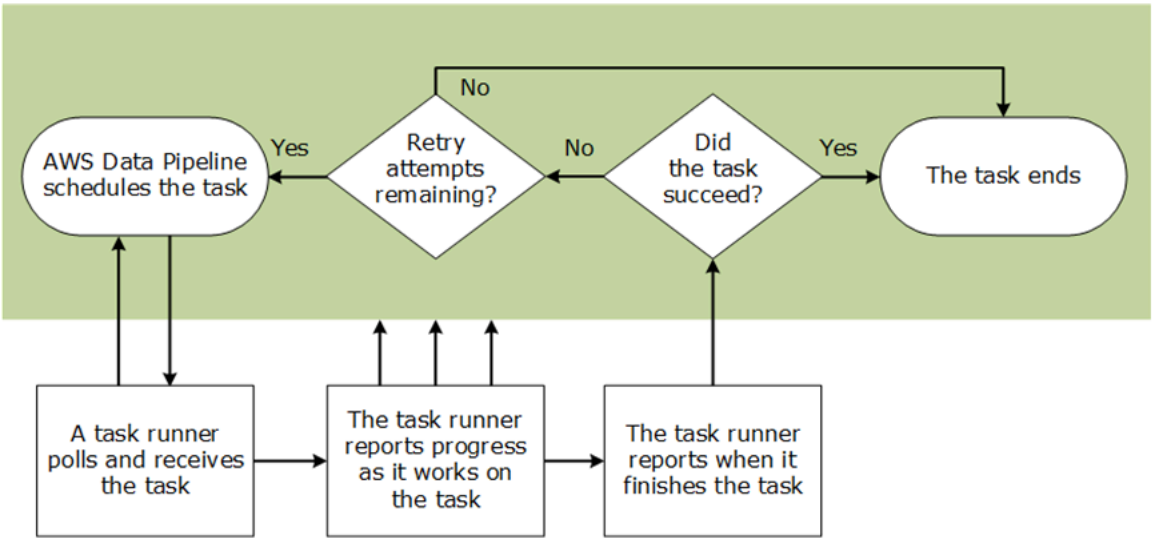


任務執行器

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

Task Runner 是由 提供的任務執行器的預設實作 AWS Data Pipeline。安裝並設定 Task Runner 時，它會輪詢與您已啟用的管道相關聯的 AWS Data Pipeline 任務。當任務指派給任務執行器時，它會執行該任務並將其狀態回報給 AWS Data Pipeline。

下圖說明 AWS Data Pipeline 和 任務執行器如何互動來處理排定的任務。任務是 AWS Data Pipeline 服務與任務執行器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。



有兩種方式可以使用 Task Runner 來處理管道：

- AWS Data Pipeline 在 AWS Data Pipeline Web 服務啟動和管理的資源上安裝 Task Runner。
- 您在管理的運算資源上安裝 Task Runner，例如長時間執行的 EC2 執行個體或內部部署伺服器。

如需使用任務執行器的詳細資訊，請參閱使用任務執行器。

資料節點

在 中 AWS Data Pipeline，資料節點會定義管道活動用作輸入或輸出的位置和資料類型。 AWS Data Pipeline 支援下列類型的資料節點：

DynamoDBDataNode

DynamoDB 資料表，其中包含 HiveActivity或 EmrActivity 要使用的資料。

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**
>
> 之前使用 MySqlDataNode。請改用 SqlDataNode。

RedshiftDataNode

Amazon Redshift 資料表，其中包含RedshiftCopyActivity供 使用的資料。

S3DataNode

Amazon S3 位置，其中包含一或多個檔案供管道活動使用。

資料庫

AWS Data Pipeline 支援下列類型的資料庫：

JdbcDatabase

JDBC 資料庫。

RdsDatabase

Amazon RDS 資料庫。

RedshiftDatabase

Amazon Redshift 資料庫。

活動

在 中 AWS Data Pipeline，活動是一種管道元件，可定義要執行的工作。 AWS Data Pipeline 提供數個預先封裝的活動，以因應常見案例，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 Hive 查詢。

HiveCopyActivity

在支援進階資料篩選的 Amazon EMR 叢集上執行 Hive 查詢，並支援 S3DataNode和 DynamoDBDataNode。

PigActivity

在 Amazon EMR 叢集上執行 Pig 指令碼。

RedshiftCopyActivity

在 Amazon Redshift 資料表之間複製資料。

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，先決條件是管道元件，其中包含條件式陳述式，必須先為 true，活動才能執行。例如，先決條件可以在管道活動嘗試複製來源資料之前檢查是否存在。 AWS Data Pipeline 提供數個預先封裝的先決條件，以適應常見案例，例如資料庫資料表是否存在、Amazon S3 金鑰是否存在等。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。系統管理的先決條件是由 AWS Data Pipeline Web 服務代表您執行，不需要運算資源。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

DynamoDBDataExists

檢查特定 DynamoDB 資料表中是否存在資料。

DynamoDBTableExists

檢查 DynamoDB 資料表是否存在。

S3KeyExists

檢查 Amazon S3 金鑰是否存在。

S3PrefixNotEmpty

檢查 Amazon S3 字首是否為空。

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。

Resources

在 中 AWS Data Pipeline，資源是執行管道活動指定之工作的運算資源。 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

執行管道活動所定義工作的 Amazon EMR 叢集，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 會擴展以容納大量並行任務，您可以將其設定為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 自動建立 20 節點的 Amazon EMR 叢集來處理資料，而您的 AWS 帳戶將 EC2 執行個體限制設為 20，您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》中的支援的平台。

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。

具有 Amazon EMR 叢集和 的 Amazon EC2 Spot 執行個體 AWS Data Pipeline

管道可以將 Amazon EC2 Spot 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。

當您使用 Spot 執行個體時， 會在叢集啟動時將您的 Spot 執行個體最高價格 AWS Data Pipeline 提交至 Amazon EMR。它會自動將叢集的工作配置到您使用 `taskInstanceCount` 欄位定義的 Spot 執行個體任務節點數量。 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

當您搭配 使用 Spot 執行個體時 AWS Data Pipeline，適用下列考量：

- 當 Spot 執行個體價格超過執行個體的最高價格，或由於 Amazon EC2 容量原因，您的 Spot 執行個體可以終止。不過，您不會遺失資料，因為 AWS Data Pipeline 會使用具有核心節點的叢集，這些節點一律為隨需執行個體，且不會受到終止的影響。
- 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 動作是管道元件在特定事件發生時採取的步驟，例如成功、失敗或延遲活動。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 通知做為主要方式，以無人看管的方式指出管道及其元件的狀態。如需詳細資訊，請參閱 Amazon SNS。除了 SNS 通知之外，您還可以使用 AWS Data Pipeline 主控台和 CLI 來取得管道狀態資訊。

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。您可以設定管道元件以通知您某些情況或事件，例如管道元件故障或不以其排定的開始時間開始。 透過提供可與 Amazon SNS 通知建立關聯

的管道元件上的事件欄位， AWS Data Pipeline 讓您輕鬆設定通知，例如 onSuccess、 OnFail和 onLateAction。

設定 的 AWS Data Pipeline

AWS Data Pipeline 第一次使用 之前，請先完成下列任務。

任務

- 註冊 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 ） 執行必要的動作
- 授予程式設計存取權

完成這些任務後，您可以開始使用 AWS Data Pipeline。如需基本教學，請參閱入門 AWS Data Pipeline。

註冊 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 用量費率的詳細資訊，請參閱 AWS Data Pipeline。

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 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或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 會在註冊程序完成後傳送確認電子郵件給您。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。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護您的 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS 管理主控台身分登入 。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)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者 （主控台） 啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

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 Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 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 Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》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 Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

為 AWS Data Pipeline 和管道資源建立 IAM 角色

AWS Data Pipeline 需要決定執行動作和存取 AWS 資源許可的 IAM 角色。管道角色會決定 AWS Data Pipeline 具有的許可，而資源角色會決定在管道資源上執行的應用程式所擁有的許可，例如 EC2 執行個體。您可以在建立管道時指定這些角色。即使您未指定自訂角色並使用預設角色 `DataPipelineDefaultRole` 和 `DataPipelineDefaultResourceRole`，仍必須先建立角色並連接許可政策。如需詳細資訊，請參閱的 IAM 角色 AWS Data Pipeline。

允許 IAM 主體（使用者和群組）執行必要的動作

若要使用管道，您必須允許帳戶中的 IAM 主體（使用者或群組）為管道定義的其他服務執行必要的AWS Data Pipeline 動作和動作。

為了簡化許可，AWSDataPipeline_FullAccess 受管政策可供您連接至 IAM 主體。此受管政策允許委託人執行使用者所需的所有動作，以及在未指定自訂角色 AWS Data Pipeline 時用於 的預設角色上的`iam:PassRole`動作。

強烈建議您仔細評估此受管政策，並僅將許可限制為使用者所需的許可。如有必要，請使用此政策作為起點，然後移除許可以建立更嚴格的內嵌許可政策，您可以將這些政策連接到 IAM 主體。如需詳細資訊和範例許可政策，請參閱 的範例政策 AWS Data Pipeline

類似下列範例的政策陳述式必須包含在連接到任何使用管道的 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。

JSON

```
{
    "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 受管政策連接至群組，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

建立使用者群組 **DataPipelineDevelopers** 並連接 AWSDataPipeline_FullAccess 政策

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. 輸入群組名稱，例如 **DataPipelineDevelopers**，然後選擇下一步。
4. **AWSDataPipeline_FullAccess** 針對篩選條件輸入 ，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，將使用者新增至群組。
   c. 從清單中選擇您要新增的使用者，然後選擇將使用者新增至群組。

授予程式設計存取權

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

若要授予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	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。

哪個使用者需要程式設計存取權？	到	根據
		• 如需 AWS CLI，請參閱AWS Command Line Interface 《使用者指南》中的登入以進行 AWS 本機開發。 • AWS SDKs，請參閱 AWS SDKs 和工具參考指南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 • AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。

哪個使用者需要程式設計存取權？	到	根據
IAM	(不建議使用) 使用長期憑證簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《 IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

入門 AWS Data Pipeline

AWS Data Pipeline 可協助您可靠且符合成本效益地排序、排程、執行和管理經常性資料處理工作負載。此服務可讓您根據您的商業邏輯，輕鬆使用現場部署及雲端中的結構化和非結構化資料來設計擷取-轉換-負載 (ETL) 活動。

若要使用 AWS Data Pipeline，您可以建立管道定義，指定資料處理的商業邏輯。典型管道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。此管道每隔 15 分鐘執行一個小時，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 的 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。

S3DataNode (輸入)

內含輸入日誌檔案的 S3 儲存貯體。

S3DataNode (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。

Schedule

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 的最快速方法是 AWS Data Pipeline 使用稱為範本的管道定義。

建立管道

1. 在 https：//https://console.aws.amazon.com/datapipeline/ 開啟 AWS Data Pipeline 主控台。
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 資源專屬於一個區域，但 AWS Data Pipeline 可讓您使用與管道不同區域中的資源。
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中建立管道，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，主控台會顯示一個頁面，列出您區域的管道。選擇建立新的管道。
4. 在名稱中，輸入管道的名稱。
5. （選用）在描述中，輸入管道的描述。
6. 針對 Source (來源)，選取 Build using a template (使用範本建置)，然後選取以下範本：Getting Started using ShellCommandActivity (使用 ShellCommandActivity 開始使用)。
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. 在管道組態下，保持啟用記錄。選擇 S3 位置下日誌的資料夾圖示，選取其中一個儲存貯體或資料夾，然後選擇選取。

   如果您願意，可以改為停用記錄。

10. 在安全/存取下，將 IAM 角色設定為預設。
11. 按一下 Activate (啟動)。

    如果您願意，可以在 Architect 中選擇編輯來修改此管道。例如，您可以新增先決條件。

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，請刪除您的管道。刪除管道會刪除管道定義和所有相關聯的物件。

刪除管道

1. 在列出管道頁面上，選取您的管道。
2. 按一下動作，然後選擇刪除。
3. 出現確認提示時，請選擇刪除。

如果您已完成本教學課程的輸出，請從 Amazon S3 儲存貯體中刪除輸出資料夾。

使用管道

您可以使用命令列界面 (CLI) 或 AWS SDK 來管理、建立和修改管道。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 的 AWS Data Pipeline。

目錄

建立管道

AWS Data Pipeline 提供多種方法來建立管道：

- 使用 AWS Command Line Interface (CLI) 搭配為方便起見提供的範本。如需詳細資訊，請參閱使用 CLI 從資料管道範本建立管道。
- 使用 AWS Command Line Interface (CLI) 搭配 JSON 格式的管道定義檔案。
- 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 從資料管道範本建立管道

Data Pipeline 提供數個預先設定的管道定義，稱為 範本。您可以使用 範本 AWS Data Pipeline 快速開始使用 。這些範本可在 Amazon S3 位置的公有儲存貯體中取得：`s3://datapipeline-us-east-1/templates/`。這些預先定義的範本是為了達成特定使用案例而建立，可用於建立管道。您可以使用 `aws s3 ls --recursive "s3://datapipeline-us-east-1/templates/"` 列出所有可用的範本。

使用 CLI 從範本建立管道

假設您想要建立管道，將 DynamoDB 資料表匯出至 Amazon S3。在此情況下要使用的範本位於：`s3://datapipeline-us-east-1/templates/DynamoDB Templates/Export DynamoDB table to S3.json`。

下載範本 JSON 並使用 CLI 建立管道

1. 使用 CLI `aws s3 cp` 或 curl 下載範本。例如：

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

2. 視需要變更下載的範本。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 CLI `create-pipeline` 建立管道。例如：

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

4. 請注意建立的管道 ID。
5. 使用 `put-pipeline-definition` 上傳定義。提供您要使用 `--parameter-values`選項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

- 開始使用 ShellCommandActivity
- 執行 AWS CLI 命令
- 將 DynamoDB 資料表匯出至 S3
- 從 S3 匯入 DynamoDB 備份資料
- 在 Amazon EMR 叢集上執行任務
- 將 Amazon RDS MySQL 資料表完整複製到 Amazon S3
- 將 Amazon RDS MySQL 資料表增量複製到 Amazon S3
- 將 S3 資料載入 Amazon RDS MySQL 資料表
- 將 Amazon RDS MySQL 資料表完整複製到 Amazon Redshift
- 將 Amazon RDS MySQL 資料表增量複製到 Amazon Redshift
- 將資料從 Amazon S3 載入 Amazon Redshift

開始使用 ShellCommandActivity

Getting Started using ShellCommandActivity (使用 ShellCommandActivity 開始使用) 範本會執行殼層命令指令碼，計算日誌檔案中的 GET 請求數。輸出會在每次排定的管道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
- S3InputNode
- S3OutputNode
- Ec2Resource

執行 AWS CLI 命令

此範本會以排定的間隔執行使用者指定的 AWS CLI 命令。

將 DynamoDB 資料表匯出至 S3

匯出 DynamoDB 資料表至 S3 範本會排程 Amazon EMR 叢集，將資料從 DynamoDB 資料表匯出至 Amazon S3 儲存貯體。此範本使用 Amazon EMR 叢集，其大小會與 DynamoDB 資料表可用的輸送量

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。在過去，匯出會使用 HiveActivity，但現在它會使用原生的 MapReduce。

範本使用下列管道物件：

- EmrActivity
- EmrCluster
- DynamoDBDataNode
- S3DataNode

從 S3 匯入 DynamoDB 備份資料

從 S3 範本匯入 DynamoDB 備份資料會排程 Amazon EMR 叢集，將先前在 Amazon S3 中建立的 DynamoDB 備份載入 DynamoDB 資料表。DynamoDB 資料表中的現有項目會使用備份資料中的項目進行更新，並將新項目新增至資料表。此範本使用 Amazon EMR 叢集，其大小會與 DynamoDB 資料表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。在過去，匯入會使用 HiveActivity，但現在它會使用原生的 MapReduce。

範本使用下列管道物件：

- EmrActivity
- EmrCluster
- DynamoDBDataNode
- S3DataNode
- S3PrefixNotEmpty

在 Amazon EMR 叢集上執行任務

Elastic MapReduce 叢集範本上的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

將 Amazon RDS MySQL 資料表完整複製到 Amazon S3

RDS MySQL 資料表到 S3 範本的完整複本會複製整個 Amazon RDS MySQL 資料表，並將輸出存放在 Amazon S3 位置。輸出會以 CSV 檔案形式儲存在指定 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
- S3DataNode

將 Amazon RDS MySQL 資料表增量複製到 Amazon S3

RDS MySQL 資料表至 S3 範本的增量複本會從 Amazon RDS MySQL 資料表執行資料的增量複本，並將輸出存放在 Amazon S3 位置。Amazon RDS MySQL 資料表必須具有上次修改的資料欄。

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。排程類型是時間序列，因此如果某個小時已排定複本， 會 AWS Data Pipeline 複製具有上次修改時間戳記的資料表列，該時間戳記落在該小時內。對資料表進行的實體刪除則不會複製。輸出會在每次排程執行時，以時間戳記子資料夾寫入 Amazon S3 位置下方。

範本使用下列管道物件：

- CopyActivity
- Ec2Resource
- SqlDataNode
- S3DataNode

將 S3 資料載入 Amazon RDS MySQL 資料表

將 S3 資料載入 RDS MySQL 資料表範本會排程 Amazon EC2 執行個體，將 CSV 檔案從以下指定的 Amazon S3 檔案路徑複製到 Amazon RDS MySQL 資料表。CSV 檔案不應具備標頭列。範本會使用 Amazon S3 資料中的項目更新 Amazon RDS MySQL 資料表中的現有項目，並將 Amazon S3 資料中的新項目新增至 Amazon RDS MySQL 資料表。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
- S3DataNode

Amazon RDS 到 Amazon Redshift 範本

下列兩個範本會使用轉譯指令碼將資料表從 Amazon RDS MySQL 複製到 Amazon Redshift，該指令碼會使用來源資料表結構描述搭配下列注意事項來建立 Amazon Redshift 資料表：

- 如果未指定分佈索引鍵，Amazon RDS 資料表的第一個主索引鍵會設定為分佈索引鍵。
- 當您將副本複製到 Amazon Redshift 時，您無法略過存在於 Amazon RDS MySQL 資料表中的資料欄。
- （選用）您可以提供 Amazon RDS MySQL 到 Amazon Redshift 資料欄資料類型映射，做為範本中的其中一個參數。如果指定此選項，則指令碼會使用此指令碼來建立 Amazon Redshift 資料表。

如果正在使用 `Overwrite_Existing` Amazon Redshift 插入模式：

- 如果未提供分發金鑰，則會使用 Amazon RDS MySQL 資料表上的主金鑰。
- 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)。只有第一個複合索引鍵會設定為 Amazon Redshift 資料表中的主索引鍵。
- 如果未提供分發金鑰，且 Amazon RDS MySQL 資料表上沒有主金鑰，則複製操作會失敗。

如需 Amazon Redshift 的詳細資訊，請參閱下列主題：

- Amazon Redshift 叢集
- Amazon Redshift COPY
- 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 Amazon Redshift 之間的資料類型轉譯

MySQL 資料類型	Amazon Redshift 資料類型	備註
TINYINT, TINYINT (size)	SMALLINT	MySQL：-128 到 127。可在括弧內指定位數上限。 Amazon Redshift：INT2。帶正負號的 2 位元組整數
TINYINT UNSIGNED, 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。 Amazon Redshift：INT2。帶正負號的 2 位元組整數
SMALLINT, SMALLINT(size)	SMALLINT	MySQL：一般的 -32768 到 32767。可在括弧內指定位數上限。 Amazon Redshift：INT2。帶正負號的 2 位元組整數
SMALLINT UNSIGNED, 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 Amazon Redshift：INT4。帶正負號的 4 位元組整數
MEDIUMINT, MEDIUMINT(size)	INTEGER	MySQL：388608 到 8388607。可在括弧內指定位數上限 Amazon Redshift：INT4。帶正負號的 4 位元組整數
MEDIUMINT UNSIGNED, MEDIUMINT(size)	INTEGER	MySQL：0 到 16777215。可在括弧內指定位數上限

MySQL 資料類型	Amazon Redshift 資料類型	備註
UNSIGNED		Amazon Redshift：INT4。帶正負號的 4 位元組整數
INT, INT(size)	INTEGER	MySQL：147483648 到 2147483647 Amazon Redshift：INT4。帶正負號的 4 位元組整數
INT UNSIGNED, INT(size) UNSIGNED	BIGINT	MySQL：0 到 4294967295 Amazon Redshift：INT8。帶正負號的 8 位元組整數
BIGINT BIGINT(size)	BIGINT	Amazon Redshift：INT8。帶正負號的 8 位元組整數
BIGINT UNSIGNED BIGINT(size) UNSIGNED	VARCHAR(20*4)	MySQL：0 到 18446744073709551615 Amazon Redshift：沒有原生對等項目，因此請使用字元陣列。
FLOAT FLOAT(size,d) 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。 Amazon Redshift：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。 Amazon Redshift：FLOAT8

MySQL 資料類型	Amazon Redshift 資料類型	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。 Amazon Redshift：沒有原生對等項目。
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。 字串右側則會填補空格。 Amazon Redshift：CHAR 資料類型不支援多位元組字元，因此會使用 VARCHAR。 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。 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	Amazon Redshift 資料類型	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 BOOL 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) "1000-01-01" 到 "9999-12-31"

MySQL 資料類型	Amazon Redshift 資料類型	備註
TIME	VARCHAR(10*4)	(hh:mm:ss) "-838:59:59" 到 "838:59:59"
DATETIME	TIMESTAMP	(YYYY-MM-DD hh:mm:ss) 1000-01-01 00:00:00" 到 "9999-12-31 23:59:59"
TIMESTAMP	TIMESTAMP	(YYYYMMDDhhmmss) 19700101000000 到 2037+
YEAR	VARCHAR(4*4)	(YYYY) 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。 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。 column GENERATED BY DEFAULT 相當於： CREATE SEQUENCE name; CREATE TABLE table (column INTEGER NOT NULL DEFAULT nextval(name));

MySQL 資料類型	Amazon Redshift 資料類型	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。 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。 column GENERATED BY DEFAULT 相當於： 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 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。

將 Amazon RDS MySQL 資料表完整複製到 Amazon Redshift

Amazon RDS MySQL 資料表至 Amazon Redshift 範本的完整副本會透過在 Amazon S3 資料夾中暫存資料，將整個 Amazon RDS MySQL 資料表複製到 Amazon Redshift 資料表。 Amazon S3 Amazon S3 預備資料夾必須與 Amazon Redshift 叢集位於相同的區域。如果 Amazon RDS MySQL 資料表不存在，則會使用與來源 Amazon RDS MySQL 資料表相同的結構描述建立 Amazon Redshift 資料表。請將您要在建立 Amazon Redshift 資料表期間套用的任何 Amazon RDS MySQL 提供給 Amazon Redshift 資料欄資料類型覆寫。

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

將 Amazon RDS MySQL 資料表增量複製到 Amazon Redshift

Amazon RDS MySQL 資料表至 Amazon Redshift 範本的增量複本會透過在 Amazon S3 資料夾中暫存資料，將資料從 Amazon RDS MySQL 資料表複製到 Amazon Redshift 資料表。 Amazon S3

Amazon S3 預備資料夾必須與 Amazon Redshift 叢集位於相同的區域。

AWS Data Pipeline 如果來源 Amazon RDS MySQL 資料表尚不存在， 會使用轉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 資料表。您必須將您要在建立 Amazon Redshift 資料表期間套用的任何 Amazon RDS MySQL 提供給 Amazon Redshift 資料欄資料類型覆寫。

此範本會在排程間隔之間複製對 Amazon RDS MySQL 資料表所做的變更，從排程的開始時間開始。不會複製對 Amazon 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。

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

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3DataNode
- SqlDataNode
- RedshiftDataNode
- RedshiftDatabase

將資料從 Amazon S3 載入 Amazon Redshift

將資料從 S3 載入 Redshift 範本，將資料從 Amazon S3 資料夾複製到 Amazon Redshift 資料表。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。

根據 Amazon Redshift COPY選項複製資料。Amazon Redshift 資料表必須與 Amazon S3 中的資料具有相同的結構描述。如需COPY選項，請參閱《Amazon Redshift 資料庫開發人員指南》中的 COPY。

範本使用下列管道物件：

- CopyActivity
- RedshiftCopyActivity
- S3DataNode
- RedshiftDataNode
- RedshiftDatabase
- Ec2Resource

使用參數化範本建立管道

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 myVariables新增至管道定義

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，以下管道定義檔案 (pipeline-definition.json) 包含下列變數：*myShellCmd*、*myS3InputLoc* 和 *myS3OutputLoc*。

> **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，以下 JSON 檔案 (parameters.json) 包含上述範例管道定義中 *myShellCmd*、*myS3InputLoc* 和 *myS3OutputLoc* 變數的參數物件。

```
{
  "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	Type	Description
id	String	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	String	參數的描述。
type	String、Integer、Double 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。

屬性	Type	Description
預設	String	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，以下 JSON 檔案 (`file://values.json`) 包含上述範例管道定義中 *myS3OutputLoc* 變數的值。

```
{
  "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-definition AWS 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

您可以使用命令列界面 (CLI) 檢視管道。

使用 檢視您的管道 AWS CLI

- 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條件。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，或在可以執行 AWS Data Pipeline 之前取消。當此元件所依賴的不同元件或資源發生故障時，就會自動發生這種情況。

CASCADE_FAILED

元件或資源由於其中一個相依項的層疊失敗而取消，但元件可能不是故障的原始來源。

DEACTIVATING

正在停用管道。

FAILED

元件或資源發生錯誤並停止運作。當元件或資源失敗時，可能會導致取消和失敗串聯到其他相依元件。

FINISHED

元件已完成其指派的工作。

INACTIVE

管道已停用。

PAUSED

元件已暫停，且目前未執行其工作。

PENDING

管道已準備好首次啟用。

RUNNING

資源正在執行並準備好接收工作。

SCHEDULED

資源已排程執行。

SHUTTING_DOWN

成功完成其工作後，資源會關閉。

SKIPPED

使用晚於目前排程的時間戳記啟動管道之後，元件略過了執行間隔。

TIMEDOUT

資源超過terminateAfter閾值且已由 停止 AWS Data Pipeline。資源達到此狀態後，會 AWS Data Pipeline 忽略該資源的 retryDelay、 actionOnResourceFailure和 retryTimeout值。此狀態僅適用於 資源。

VALIDATING

正在驗證管道定義 AWS Data Pipeline。

WAITING_FOR_RUNNER

元件正在等待工作者用戶端擷取工作項目。元件和工作者用戶端關係是由該元件定義的 runsOn或 workerGroup 欄位所控制。

WAITING_ON_DEPENDENCIES

元件正在驗證在執行其工作之前，是否符合其預設和使用者設定的先決條件。

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。您可以在執行詳細資訊頁面上按一下HEALTHY狀態，查看最近成功完成的管道元件執行個體。

ERROR

管道中至少有一個元件的運作狀態為 ERROR。您可以在執行詳細資訊頁面上按一下ERROR狀態，查看最近失敗的管道元件執行個體。

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管道元件執行個體。

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管道元件執行個體。

`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

使用命令列界面 (CLI) 來檢視您的管道定義。CLI 會以 JSON 格式列印管道定義檔案。如需管道定義檔案語法和使用方式的資訊，請參閱管道定義檔案語法。

使用 CLI 時，建議您在提交修改之前擷取管道定義，因為在您上次使用管道定義之後，其他使用者或程序可能會變更管道定義。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，您可以取得管道的兩個不同版本。`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 取得管道定義 AWS CLI

若要取得完整的管道定義，請使用以下的 get-pipeline-definition 命令。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 監控管道的進度 AWS CLI

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

在管道建立時，透過在主控台中指定 Amazon S3 位置，或在 SDK/CLI 的預設物件pipelineLogUri中指定 ，支援管道層級記錄。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 (包含那些資源的 TaskRunner 日誌) 會遵循上述的管道記錄日誌結構。

若要檢視指定管道執行的日誌：

1. ObjectId 呼叫 以取得確切的物件 ID query-objects 來擷取 。例如：

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

query-objects 是分頁 CLI，如果指定 的執行次數較多，則可能會傳回分頁字符pipeline-id。您可以使用權杖進行所有嘗試，直到您找到預期的物件為止。例如，傳回的 ObjectId 看起來會像是：@TableBackupActivity_2023-05-020T18:05:18_Attempt=1。

2. 使用 ObjectId，使用 擷取日誌位置：

```
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

若要取得錯誤訊息，請先使用 取得 ObjectIdquery-objects。

擷取失敗的 ObjectId 之後，請使用 describe-objects CLI 取得實際錯誤訊息。

```
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記為已完成。

首先，使用 CLI query-objects 取得物件 ID。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 狀態且未啟用時，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 編輯管道 AWS CLI

您可以使用命令列工具編輯管道。

首先，請使用 get-pipeline-definition 命令下載目前管道定義的複本。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 命令來更新您的管道定義。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 CLI AWS 複製管道：

1. 使用新名稱和唯一 ID 建立新的管道。請注意傳回的管道 ID。
2. 使用 get-pipeline-definition CLI 取得要複製之現有管道的管道定義，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 將管道定義從現有管道複製到新管道。
4. 使用 get-pipeline-definition CLI 取得新管道的定義，以驗證管道定義。

```
# 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至其基礎資源（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。根據預設，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，管道預設會從上次完成的執行繼續，或者您可以指定繼續管道的日期和時間。

使用 停用您的管道 AWS CLI

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 刪除管道 AWS CLI

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 可以在管道中暫存輸入和輸出資料，以便更輕鬆地使用某些活動，例如 `ShellCommandActivity`和 `HiveActivity`。

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。

使用活動 shell 命令或 Hive 指令碼中的特殊變數，即可取得 Amazon EMR 或 Amazon EC2 資源上的暫存資料。

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，資料欄位可供活動使用 AWS Data Pipeline 表達式語法進行參考，只有在滿足相依性時才會發生這種情況。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。

使用 ShellCommandActivity 進行資料預備

考慮使用 ShellCommandActivity S3DataNode 物件做為資料輸入和輸出的案例。AWS Data Pipeline 會自動將資料節點分段，使其可存取 shell 命令，就像是使用環境變數${INPUT1_STAGING_DIR}的本機檔案資料夾${OUTPUT1_STAGING_DIR}一樣，如下列範例所示。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

考慮使用 HiveActivity 搭配 S3DataNode 物件做為資料輸入和輸出的案例。 AWS Data Pipeline 會自動分階段資料節點，使其可存取 Hive 指令碼，就像是使用 變數${input1}的 Hive 資料表${output1}，如下列 範例所示HiveActivity。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。沒有資料預備可供 使用DynamoDBDataNode，因此您必須先使用變數名稱來#{input.tableName}參考 DynamoDB 資料表，在 hive 指令碼中手動建立資料表。如果 DynamoDB 資料表是輸出，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 # （雜湊） 字元字首，因為 AWS Data Pipeline 使用表達式來存取 tableName或 directoryPath。如需表達式評估如何在 中運作的詳細資訊 AWS Data Pipeline，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 相同的區域中執行 AWS Data Pipeline，但 AWS Data Pipeline 支援跨多個區域協調資料流程的功能，例如在一個區域中執行資源，以合併來自另一個區域的輸入資料。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 AWS Data Pipeline 上使用 region 欄位，將資源設定為在與 不同的區域中執行EmrCluster。

下列範例管道 JSON 檔案顯示如何在歐洲 （愛爾蘭） 區域執行 EmrCluster 資源，假設叢集要處理的大量資料存在於相同的區域中。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 可在其中協調工作流程並啟動 Amazon EMR 或 Amazon EC2 資源的區域。這些區域 AWS Data Pipeline 可能不支援 。如需 AWS Data Pipeline 支援的區域資訊，請參閱 AWS 區域和端點。

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

區域名稱	區域代碼
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 可讓您設定管道物件在相依性失敗或使用者取消時的行為方式。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。

Resources

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。

重新執行層疊失敗的物件

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。

層疊失敗和回填

如果您啟用階層失敗，且管道建立許多回填，則管道執行時間錯誤可能會導致資源快速連續建立和刪除，而無需執行有用的工作。當您儲存管道時， AWS Data Pipeline 會嘗試使用以下警告訊息提醒您此情況： *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. 這是因為階層失敗可能會快速將下游活動設定為 ，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。

管道定義檔案語法

本節中的指示適用於使用 AWS Data Pipeline 命令列界面 (CLI) 手動使用管道定義檔案。這是使用 AWS Data Pipeline 主控台以互動方式設計管道的替代方案。

您可以使用支援使用 UTF-8 檔案格式儲存檔案的任何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 也支援管道定義中的各種複雜表達式和函數。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"
}
```

使用者定義的欄位

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。以下範例顯示新增到 S3DataNode 物件，名為 myCustomField 和 my_customFieldReference 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**
>
> 在使用者定義的欄位中， AWS Data Pipeline 僅檢查對其他管道元件的有效參考，而不是您新增的任何自訂欄位字串值。

使用 API

> **Note**
>
> 如果您不是撰寫與 互動的程式 AWS Data Pipeline，則不需要安裝任何 AWS SDKs。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 的 AWS Data Pipeline

撰寫與 互動 AWS Data Pipeline 或實作自訂任務執行器之應用程式最簡單的方式，就是使用其中一個 AWS SDKs。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 SDKs 提供的函數可包裝 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。SDKs也包含範例程式碼、教學課程和其他資源，協助您開始撰寫 呼叫的應用程式 AWS。在 SDK 中呼叫包裝函式可以大幅簡化撰寫 AWS 應用程式的程序。如需如何下載和使用 AWS SDKs的詳細資訊，請前往範例程式碼和程式庫。

AWS Data Pipeline 支援適用於下列平台SDKs：

- 適用於 Java 的 AWS SDK
- 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 SDK
- 適用於 .NET 的 AWS SDK

向 提出 HTTP 請求 AWS Data Pipeline

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

如果您不使用其中一個 AWS SDKs，您可以使用 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 在 HTTP 請求的 標頭中需要以下資訊：

- `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，系統會在請求身分驗證時略過任何 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 資料結構描述進行格式化。 AWS Data Pipeline JSON 資料結構描述會定義每個操作可用的資料類型和參數（例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 — 此標頭後面接著狀態碼。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，如果您需要對請求進行故障診斷，您可以使用該 ID AWS Data Pipeline。請求 ID 範例為 K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG。
- x-amz-crc32 —AWS Data Pipeline 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"}
```

中的安全性 AWS Data Pipeline

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用的合規計劃 AWS Data Pipeline，請參閱合規計劃的 AWS 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。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。

本文件可協助您了解如何在使用 時套用共同責任模型 AWS Data Pipeline。下列主題說明如何設定 AWS Data Pipeline 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS Data Pipeline 資源。

主題

中的資料保護 AWS Data Pipeline

AWS 共同責任模型適用於 中的資料保護 AWS Data Pipeline。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。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更多相關資訊，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 共同責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與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：

- 每個帳戶均要使用多重要素驗證 (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 設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 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 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-2 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。
- AWS Data Pipeline 支援 Amazon EMR 和 Amazon EC2 資源的 IMDSv2。 Amazon EC2 若要將 IMDSv2 與 Amazon 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 或更新版本。如需詳細資訊，請參閱設定 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，例如名稱欄位。這包括當您使用 AWS Data Pipeline 或使用主控台、API AWS CLI或其他 AWS 服務 AWS SDKs 時。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。

的 Identity and Access Management AWS Data Pipeline

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。您可以使用 AWS Data Pipeline 和 AWS Identity and Access Management (IAM) 的功能，允許 AWS Data Pipeline 和其他使用者存取您的 AWS Data Pipeline 資源，而無需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) 整合，提供廣泛的功能：

- 在 中建立使用者和群組 AWS 帳戶。

- 在 中的使用者之間輕鬆共用您的 AWS 資源 AWS 帳戶。
- 為每個使用者指派唯一的安全登入資料。
- 控制每個使用者對 服務和資源的存取。
- 取得您 中所有使用者的單一帳單 AWS 帳戶。

透過搭配 使用 IAM AWS Data Pipeline，您可以控制組織中的使用者是否可以使用特定 API 動作來執行任務，以及是否可以使用特定 AWS 資源。您可以根據管道標籤和工作者群組使用 IAM 政策，與其他使用者共用您的管道，並控制他們擁有的存取層級。

目錄

的 IAM 政策 AWS Data Pipeline

根據預設，IAM 實體沒有建立或修改 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 實體使用所需特定資源和 API 動作的許可，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 政策的一般資訊，請參閱《IAM 使用者指南》中的許可和政策。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。

目錄

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。根據預設，IAM 實體沒有使用資源和 API 動作的許可，因此所有請求都會遭到拒絕。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 Pipeline API 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 實作全 AWS 內容金鑰 （請參閱適用於 條件的可用金鑰)，以及下列服務特定的金鑰。

  - datapipeline:PipelineCreator — 將存取權授予建立管道的使用者。如需範例，請參閱將完整存取授予管道擁有者。
  - datapipeline:Tag — 根據管道標記授予存取權。如需詳細資訊，請參閱使用標籤控制管道的存取。
  - datapipeline:workerGroup — 根據工作者群組的名稱授予存取權。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

您可以建立參考管道標籤的 IAM 政策。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，管理員會將讀取/寫入存取權授予生產 IAM 群組中的使用者，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。

為了達成此案例，管理員會使用「環境=生產」標籤來標記生產管道，並將下列政策連接至開發人員 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。

JSON

```
{
  "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連接至生產 IAM 群組。此陳述式會授予所有管道的完整存取。

JSON

```
{
```

```
  "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

您可以建立建立參考工作者群組名稱的 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至生產資源的管道，而且開發 IAM 群組中的使用者可以建立將任務推送至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連接至開發人員 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。

此外，管理員會將下列政策連接至生產 IAM 群組，以授予對「資源」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。

的範例政策 AWS Data Pipeline

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。

目錄

- 範例 4：授予使用者對 AWS Data Pipeline 主控台的存取權

範例 1：根據標籤將唯讀存取授予使用者

下列政策允許使用者使用唯讀 AWS Data Pipeline API 動作，但僅限於標籤為 "environment=production" 的管道。

ListPipelines API 動作不支援以標籤為基礎的授權。

JSON

```
{
  "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

下列政策允許使用者使用 ListPipelines 以外的所有 AWS Data Pipeline API 動作，但僅限於標籤為 "environment=test" 的管道。

JSON

```
{
  "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

下列政策允許使用者使用所有 AWS Data Pipeline API 動作，但只能搭配自己的管道。

範例 4：授予使用者對 AWS Data Pipeline 主控台的存取權

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。

此政策包含與 `roleARN` AWS Data Pipeline 需求綁定之特定資源的`PassRole`許可動作。如需以身分為基礎 (IAM) 的 `PassRole` 許可詳細資訊，請參閱部落格文章：使用 IAM 角色授予啟動 EC2 執行個體的許可 (PassRole 許可)。

JSON

```
{
 "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"
   ]
  }
 ]
}
```

的 IAM 角色 AWS Data Pipeline

AWS Data Pipeline 使用 AWS Identity and Access Management 角色。連接至 IAM 角色的許可政策會決定 AWS Data Pipeline 您的應用程式可執行哪些動作，以及他們可以存取哪些 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：

- 管道角色控制對 AWS 資源的 AWS Data Pipeline 存取。在管道物件定義中， `role` 欄位會指定此角色。
- EC2 執行個體角色控制在 EC2 執行個體上執行的應用程式必須存取 AWS 資源的存取權，包括 Amazon EMR 叢集中的 EC2 執行個體。在管道物件定義中， `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， `DataPipelineDefaultRole`會為您 AWS Data Pipeline 建立 ，並將 `AWSDataPipelineRole`受管政策連接至角色。自 2022 年 10 月 3 日起， `AWSDataPipelineRole`受管政策已棄用，且必須使用 主控台為管道指定管道角色。我們建議您檢閱現有的管道，並判斷 `DataPipelineDefaultRole` 是否與管道相關聯，以及是否`AWSDataPipelineRole`連接到該角色。若是如此，請檢閱此政策允許的存取權，以確保其符合您的安全需求。視需要新增、更新或取代連接到此角色的政策和政策陳述式。或者，您可以更新管道，以使用您透過不同許可政策建立的角色。

AWS Data Pipeline 角色的許可政策範例

每個角色都連接一或多個許可政策，以決定角色可存取 AWS 的資源和角色可執行的動作。本主題提供管道角色的範例許可政策。它也提供 的內容`AmazonEC2RoleforDataPipelineRole`，這是預設 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是允許 AWS Data Pipeline 需要 以 Amazon EC2 和 Amazon EMR 資源執行管道的基本函數。它也提供許可來存取許多管道所需的其他 AWS 資源，例如 Amazon Simple Storage Service 和 Amazon Simple Notification Service。如果管道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許可。例如，如果您的管道未定義 DynamoDBDataNode或使用 SnsAlarm動作，建議您移除這些動作的允許陳述式。

- 將 取代*111122223333*為 AWS 您的帳戶 ID。
- *NameOfDataPipelineRole* 將 取代為管道角色的名稱 （此政策所連接的角色）。
- *NameOfDataPipelineResourceRole* 將 取代為 EC2 執行個體角色的名稱。
- *us-west-1* 將 取代為適合您應用程式的 區域。

EC2 執行個體角色的預設受管政策

的內容AmazonEC2RoleforDataPipelineRole如下所示。這是連接到 的預設資源角色的受管政策 AWS Data PipelineDataPipelineDefaultResourceRole。當您定義管道的資源角色時，建議您從此許可政策開始，然後移除非必要 AWS 服務動作的許可。

政策第 3 版隨即顯示，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。

JSON

```
{
  "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 建立 IAM 角色 AWS Data Pipeline 並編輯角色許可

使用下列程序來建立 AWS Data Pipeline 使用 IAM 主控台的 角色。程序包含兩個步驟。首先，您可以建立連接到角色的許可政策。接著，建立角色並連接政策。建立角色之後，您可以透過連接和分離許可政策來變更角色的許可。

> **Note**
>
> 當您 AWS Data Pipeline 使用主控台建立 角色時，IAM 會建立並連接角色所需的適當信任政策。

建立許可政策以搭配 的角色使用 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. 選擇 JSON 標籤。
4. 如果您要建立管道角色，請在 中複製並貼上政策範例的內容管道角色許可政策範例，並根據您的安全需求進行編輯。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 EC2 執行個體角色的預設受管政策。
5. 選擇檢閱政策。
6. 輸入政策的名稱，例如 `MyDataPipelineRolePolicy-` 和選用的描述，然後選擇建立政策。
7. 請記下政策的名稱。建立角色時需要它。

為 建立 IAM 角色 AWS Data Pipeline

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，選擇角色，然後選擇建立角色。
3. 在選擇使用案例下，選擇資料管道。
4. 在選取您的使用案例下，執行下列其中一項操作：
   - 選擇 以`Data Pipeline`建立管道角色。
   - 選擇 `EC2 Role for Data Pipeline` 以建立資源角色。
5. 選擇下一步：許可。
6. 如果 AWS Data Pipeline 列出 的預設政策，請繼續以下步驟來建立角色，然後根據下一個程序中的指示進行編輯。否則，請輸入您在上述程序中建立的政策名稱，然後從清單中選擇。
7. 選擇下一步：標籤，輸入要新增至角色的任何標籤，然後選擇下一步：檢閱。
8. 輸入角色的名稱，例如，`MyDataPipelineRole`以及選用的描述，然後選擇建立角色。

為 的 IAM 角色連接或分離許可政策 AWS Data Pipeline

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色
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，執行下列動作：
   - 若要分離許可政策，請在許可政策下，選擇政策項目最右側的移除按鈕。在出現確認提示時，選擇分離。
   - 若要連接您先前建立的政策，請選擇連接政策。在搜尋方塊中，開始輸入您要編輯的政策名稱，從清單中選擇，然後選擇連接政策。

變更現有管道的角色

如果您想要將不同的管道角色或資源角色指派給管道，您可以在 AWS Data Pipeline 主控台中使用架構師編輯器。

使用主控台編輯指派給管道的角色

1. 在 https：//https://console.aws.amazon.com/datapipeline/ 開啟 AWS Data Pipeline 主控台。
2. 從清單中選擇管道，然後選擇動作、編輯。
3. 在架構師編輯器的右窗格中，選擇其他。
4. 從資源角色和角色清單中，選擇要指派 AWS Data Pipeline 的角色，然後選擇儲存。

在 中記錄和監控 AWS Data Pipeline

AWS Data Pipeline 已與 服務整合 AWS CloudTrail，此服務可提供使用者、角色或 AWS 服務在其中採取之動作的記錄 AWS Data Pipeline。CloudTrail 會將 的所有 API 呼叫擷取 AWS Data Pipeline 為事件。擷取的呼叫包括來自 AWS Data Pipeline 主控台的呼叫，以及對 API 操作的 AWS Data Pipeline 程式碼呼叫。如果您建立線索，您可以將 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，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 所收集的資訊來判斷提出的請求 AWS Data Pipeline、提出請求的 IP 地址、提出請求的人員、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AWS CloudTrail 《使用者指南》。

AWS Data Pipeline CloudTrail 中的資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 中發生時 AWS Data Pipeline，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。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 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶中的事件，包括 的事件 AWS Data Pipeline，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 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。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 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 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 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

CloudTrail 會記錄所有 AWS Data Pipeline 動作，並記錄在 AWS Data Pipeline API 參考動作章節中。例如，呼叫 CreatePipeline 動作會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案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：

- 請求是使用根或 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。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AWS Data Pipeline 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。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 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，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
  ]
}
```

中的事件回應 AWS Data Pipeline

的事件回應 AWS Data Pipeline 是 AWS 的責任。 AWS 具有正式、有文件記錄的政策和計劃，可管理事件回應。

AWS Service Health Dashboard 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。

的合規驗證 AWS Data Pipeline

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。

中的彈性 AWS Data Pipeline

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置的。 AWS 區域提供多個實體隔離和隔離的可用區域，這些可用區域以低延遲、高輸送量和高度備援的網路連接。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、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

中的基礎設施安全 AWS Data Pipeline

作為受管服務， AWS Data Pipeline 受到 Amazon Web Services：安全程序概觀白皮書中所述的 AWS 全球網路安全程序的保護。

您可以使用 AWS 發佈的 API 呼叫， AWS Data Pipeline 透過網路存取 。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 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

中的組態和漏洞分析 AWS Data Pipeline

組態和 IT 控制是客戶 AWS 與您之間的共同責任。如需詳細資訊，請參閱 AWS 共同的責任模型。

教學

以下教學可引導您逐步完成使用 AWS Data Pipeline建立和使用管道的程序。

教學

- 搭配 Hadoop Streaming 使用 Amazon EMR 處理資料
- 使用 在 Amazon S3 儲存貯體之間複製 CSV 資料 AWS Data Pipeline
- 使用 將 MySQL 資料匯出至 Amazon S3 AWS Data Pipeline
- 使用 將資料複製到 Amazon Redshift AWS Data Pipeline

搭配 Hadoop Streaming 使用 Amazon EMR 處理資料

您可以使用 AWS Data Pipeline 來管理 Amazon EMR 叢集。透過 AWS Data Pipeline ，您可以指定在啟動叢集之前必須符合的先決條件 （ 例如，確保今天的資料已上傳至 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。

在本教學課程中，您會為簡單的 Amazon EMR 叢集建立管道，以執行 Amazon EMR 提供的預先存在 Hadoop 串流任務，並在任務成功完成後傳送 Amazon SNS 通知。您可以使用 AWS Data Pipeline 為此任務提供的 Amazon EMR 叢集資源。範例應用程式稱為 WordCount，也可以從 Amazon EMR 主控台手動執行。請注意， 代 AWS Data Pipeline 您產生的叢集會顯示在 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 （ 執行 Amazon EMR 提供的預先存在 Hadoop 串流任務 ）。

EmrCluster

資源 AWS Data Pipeline 使用 來執行此活動。

叢集是一組 Amazon EC2 執行個體。 會 AWS Data Pipeline 啟動叢集，然後在任務完成後將其終止。

Schedule

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 通知傳送至您指定的主題。

目錄

開始之前

請務必完成下列步驟。

- 完成 設定 的 AWS Data Pipeline 中的任務。
- (選用) 為叢集設定 VPC，並為 VPC 設定安全群組。
- 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 Simple Notification Service 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 日誌或分析科學資料，您可以使用 AWS Data Pipeline 來管理您的 Amazon EMR 叢集。使用 時 AWS Data Pipeline，您可以指定在叢集啟動之前必須符合的先決條件 （ 例如，確保今天的資料已上傳至 Amazon S3。) 本教學課程會逐步引導您啟動叢集，該叢集可以是簡單 Amazon EMR 型管道的模型，或做為更涉及管道的一部分。

先決條件

在您可以使用 CLI 之前，必須完成下列步驟：

1. 安裝和設定命令列界面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確保名為 DataPipelineDefaultRole 和 DataPipelineDefaultResourceRole 的 IAM 角色存在。AWS Data Pipeline 主控台會自動為您建立這些角色。如果您至少尚未使用 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，請參閱的 IAM 角色 AWS Data Pipeline。

任務

- 監控管道執行

建立管道定義檔案

下列程式碼是簡單 Amazon EMR 叢集的管道定義檔案，該叢集會執行 Amazon EMR 提供的現有 Hadoop 串流任務。此範例應用程式稱為 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請將 設定為過去一天的日期，endDateTime將 設定為未來的一天。 AWS Data Pipeline 然後 會立即開始啟動「逾期」叢集，嘗試將其視為工作待處理項目。此回填表示您不需要等待一小時，即可看到 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`，代表用來執行叢集的一組 Amazon EC2 執行個體。您可以指定執行為叢集之 EC2 執行個體的大小和數量。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，例如引導操作，將其他軟體載入 Amazon EMR 提供的 AMI。
- `MyEmrActivity`，代表使用叢集處理的運算。Amazon EMR 支援多種類型的叢集，包括串流、串聯和指令碼 Hive。`runsOn` 欄位是指返回 MyCluster，使用它做為叢集基礎的規格。

上傳並啟動管道定義

您必須上傳管道定義並啟用管道。在下列範例命令中，將 *`pipeline_name`* 取代為管道的標籤，並將 *`pipeline_file`* 取代為管道定義`.json`檔案的完整路徑。

AWS CLI

若要建立管道定義並啟用管道，請使用下列 create-pipeline 命令。請注意管道的 ID，因為您會將此值與大多數 CLI 命令搭配使用。

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

若要上傳管道定義，請使用下列 put-pipeline-definition 命令。

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

如果您管道驗證成功，`validationErrors` 欄位會是空的。您應該檢閱任何警告。

若要啟用管道，請使用下列 activate-pipeline 命令。

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

您可以使用下列 list-pipelines 命令，驗證管道是否出現在管道清單中。

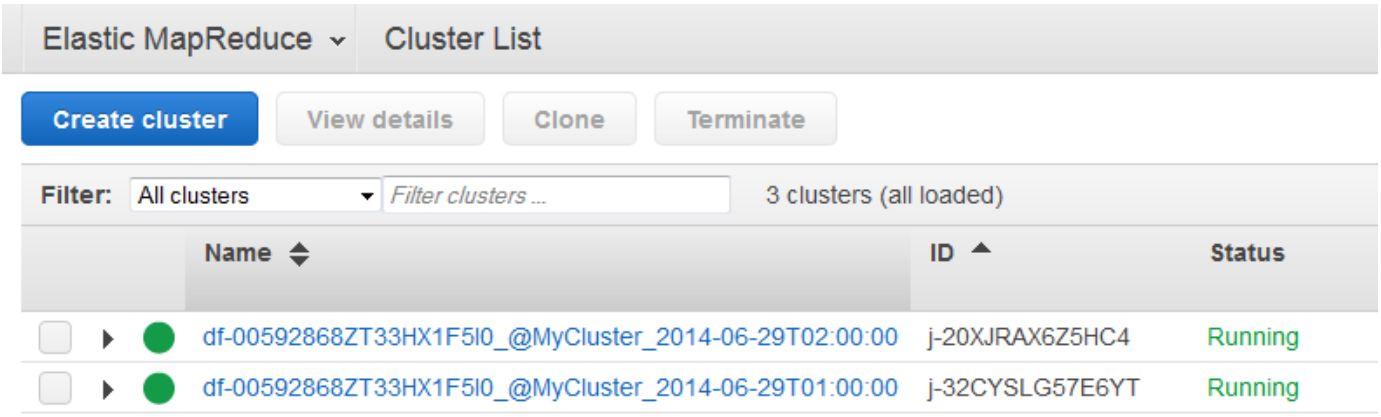
```
aws datapipeline list-pipelines
```

監控管道執行

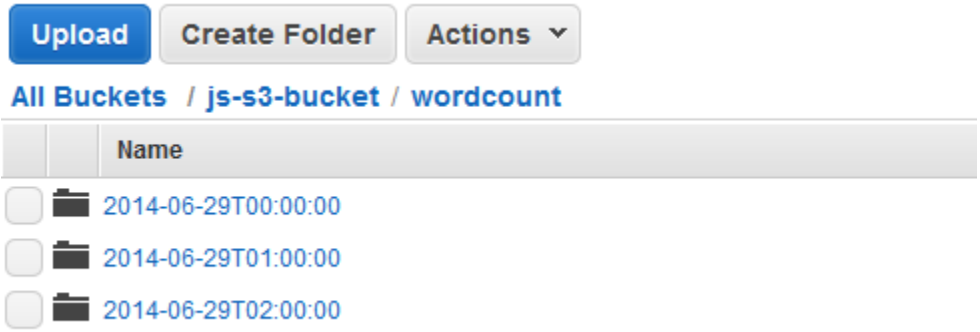
您可以使用 Amazon EMR 主控台檢視啟動 AWS Data Pipeline 的叢集，也可以使用 Amazon S3 主控台檢視輸出資料夾。

若要檢查 啟動的叢集進度 AWS Data Pipeline

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



3. 其中一個執行完成後，請開啟 Amazon S3 主控台，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。



使用 在 Amazon S3 儲存貯體之間複製 CSV 資料 AWS Data Pipeline

在您讀取什麼是 AWS Data Pipeline？並決定要使用 AWS Data Pipeline 來自動化資料的移動和轉換之後，就可以開始建立資料管道。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。

本教學課程將逐步引導您建立資料管道，以將資料從一個 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。您可以針對此複製活動使用 管理 AWS Data Pipeline 的 EC2 執行個體。

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。

Schedule

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）。

S3DataNode

此管道的輸入和輸出節點 (Amazon S3 儲存貯體）。

SnsAlarm

動作 AWS Data Pipeline 必須在符合指定條件時採取（任務成功完成後傳送 Amazon SNS 通知至主題）。

目錄

開始之前

請務必完成下列步驟。

- 完成 設定 的 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。
- 將您的資料上傳至 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 Simple Notification Service 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，請遵循中所述的指示的 IAM 角色 AWS Data Pipeline。

使用命令列複製 CSV 資料

您可以建立並使用管道，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. 安裝和設定命令列界面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確保名為 DataPipelineDefaultRole 和 DataPipelineDefaultResourceRole 的 IAM 角色存在。AWS Data Pipeline 主控台會自動為您建立這些角色。如果您至少尚未使用 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，請參閱的 IAM 角色 AWS Data Pipeline。

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

此範例案例說明如何使用 JSON 管道定義和 CLI， AWS Data Pipeline 在特定時間間隔排程在兩個 Amazon S3 儲存貯體之間複製資料。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://amzn-s3-demo-bucket/source/inputfile.csv"
    },
    {
      "id": "S3Output",
      "type": "S3DataNode",
      "schedule": {
        "ref": "MySchedule"
      },
      "filePath": "s3://amzn-s3-demo-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"
      }
    }
  ]
}
```

Schedule

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"
},
```

Amazon S3 資料節點

接下來，輸入 S3DataNode 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 S3DataNode 元件是由下列欄位定義：

```
{
  "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)。

Type

在 Amazon S3S3DataNode」。

Schedule

我們在上述 JSON 檔案的程式碼行中已建立的排程元件參考，標示為 “MySchedule”。

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資料表的不同語法。

接著，輸出 S3DataNode 元件會定義資料的輸出目的地位置。其採用與輸入 S3DataNode 元件相同的格式，不同之處在於元件的名稱，以及表示目標檔案的不同路徑。

```
{
  "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 應該自動建立 EC2 執行個體來執行複製任務，並在任務完成後終止資源。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。

Type

要執行工作的運算資源類型；在本例中是 EC2 執行個體。您可以使用其他資源類型，例如 EmrCluster 類型。

Schedule

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。請確定您設定適當大小的 EC2 執行個體，最適合您要執行的工作負載 AWS Data Pipeline。在本例中，我們將設定 m1.medium EC2 執行個體。如需不同執行個體類型以及何時使用每個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 https：//http://aws.amazon.com/ec2/instance-types/。

Role

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

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。Role 和 ResourceRole 可以是相同的角色，但不同的角色可提高您安全組態的精細程度。

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 檔案複製到另一個檔案。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。

Type

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 runs0n 欄位 AWS Data Pipeline 可讓 為您建立 EC2 執行個體。runs0n 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。

Output

目標位置資料。

Schedule

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。在下列範例命令中，將 *pipeline_name* 取代為管道的標籤，並將 *pipeline_file* 取代為管道定義.json檔案的完整路徑。

AWS CLI

若要建立管道定義並啟用管道，請使用下列 create-pipeline 命令。請注意管道的 ID，因為您會將此值與大多數 CLI 命令搭配使用。

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

若要上傳管道定義，請使用下列 put-pipeline-definition 命令。

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

如果您管道驗證成功， validationErrors 欄位會是空的。您應該檢閱任何警告。

若要啟用管道，請使用下列 activate-pipeline 命令。

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

您可以使用下列 list-pipelines 命令，驗證管道是否出現在管道清單中。

```
aws datapipeline list-pipelines
```

使用 將 MySQL 資料匯出至 Amazon S3 AWS Data Pipeline

本教學課程將逐步引導您建立資料管道，以將資料 （資料列） 從 MySQL 資料庫的資料表複製到 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 AWS Data Pipeline 針對此複製活動使用 提供的 EC2 執行個體。

管道物件

管道會使用下列物件：

- CopyActivity
- Ec2Resource
- MySqlDataNode
- S3DataNode
- SnsAlarm

目錄

開始之前

請務必完成下列步驟。

- 完成 設定 的 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。

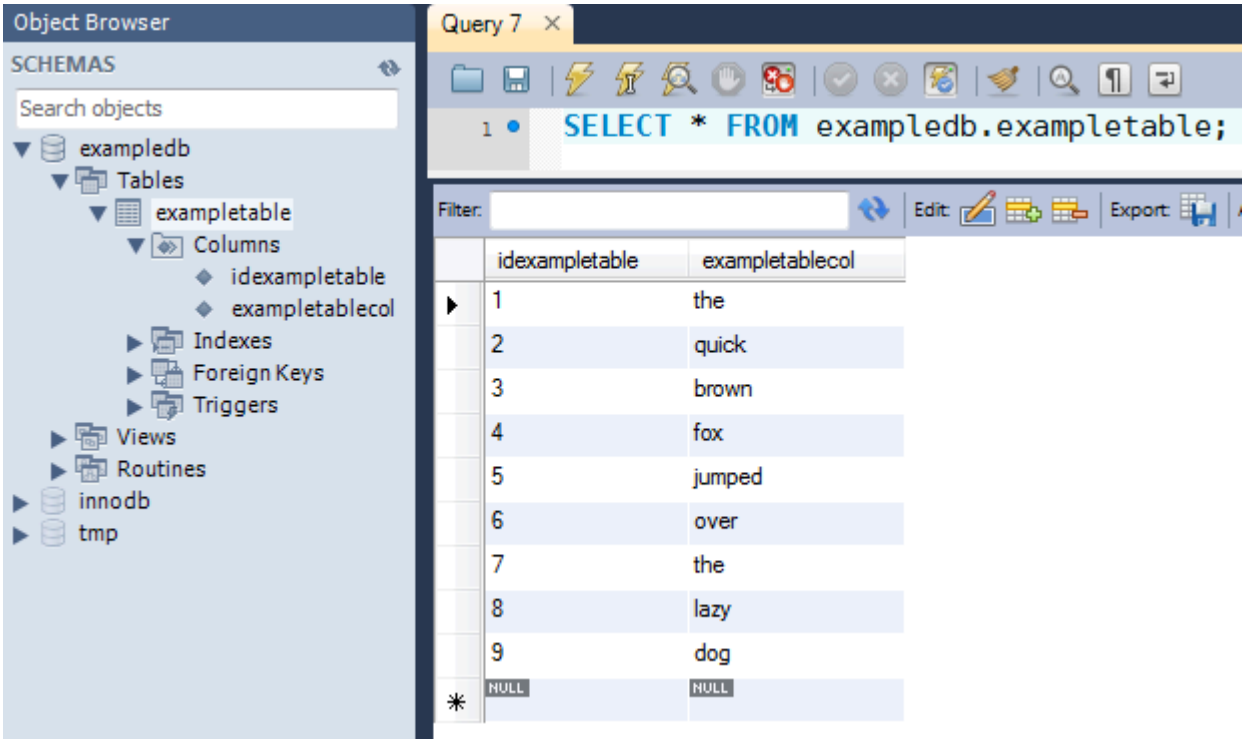
  如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的建立儲存貯體。
- 建立和啟動 MySQL 資料庫執行個體做為您的資料來源。

  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。在您擁有 Amazon RDS 執行個體之後，請參閱 MySQL 文件中的建立資料表。

  > **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 Simple Notification Service 入門指南》中的建立主題。
- （選用）本教學課程使用 建立的預設 IAM 角色政策 AWS Data Pipeline。如果您想要建立和設定 IAM 角色政策和信任關係，請遵循中所述的指示的 IAM 角色 AWS Data Pipeline。

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. 安裝和設定命令列界面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確保名為 DataPipelineDefaultRole 和 DataPipelineDefaultResourceRole 的 IAM 角色存在。AWS Data Pipeline 主控台會自動為您建立這些角色。如果您至少尚未使用 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，請參閱的 IAM 角色 AWS Data Pipeline。
3. 設定 Amazon S3 儲存貯體和 Amazon RDS 執行個體。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

此範例案例示範如何使用 JSON 管道定義和 CLI， AWS Data Pipeline 在指定的時間間隔將資料 （資料列） 從 MySQL 資料庫中的資料表複製到 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://amzn-s3-demo-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 管道元件會定義輸入資料的位置；在此情況下為 Amazon RDS 執行個體。輸入 MySqlDataNode 元件是由下列欄位定義：

```
{
  "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。以您的使用者名稱取代 *my-username*。

Schedule

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

具有星號字首的資料庫帳戶密碼，表示 AWS Data Pipeline 必須加密密碼值。將 *my-password* 取代為您的使用者正確的密碼。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

CopyActivity 物件用來連接至資料庫的 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。

Type

SqlDataNode 類型，在此範例中是使用 MySQL 的 Amazon RDS 執行個體。

> **Note**
>
> MySqlDataNode 類型已移除。雖然您仍可使用 MySqlDataNode，但我們建議您使用 SqlDataNode。

Amazon S3 資料節點

接著，S3Output 管道元件會定義輸出檔案的位置；在此情況下，Amazon S3 儲存貯體位置中的 CSV 檔案。輸出 S3DataNode 元件是由下列欄位定義：

```
{
  "id": "S3DataNodeId114",
  "schedule": {
    "ref": "ScheduleId113"
  },
  "filePath": "s3://amzn-s3-demo-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。

Schedule

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。

Type

管道物件類型，也就是與 Amazon S3 儲存貯體中資料所在的位置相符的 S3DataNode。

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 應該自動建立 EC2 執行個體來執行複製任務，並在任務完成後終止資源。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。

Schedule

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。

Role

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

Type

要執行工作的運算資源類型；在本例中是 EC2 執行個體。您可以使用其他資源類型，例如 EmrCluster 類型。

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。Role 和 ResourceRole 可以是相同的角色，但不同的角色可提高您安全組態的精細程度。

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

Schedule

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 AWS Data Pipeline 可讓 為您建立 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

Output

CSV 輸出檔案的 Amazon S3 位置

Type

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。在下列範例命令中，將 *pipeline_name* 取代為管道的標籤，並將 *pipeline_file* 取代為管道定義.json檔案的完整路徑。

AWS CLI

若要建立管道定義並啟用管道，請使用下列 create-pipeline 命令。請注意管道的 ID，因為您會將此值與大多數 CLI 命令搭配使用。

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

若要上傳管道定義，請使用下列 put-pipeline-definition 命令。

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

如果您管道驗證成功， validationErrors 欄位會是空的。您應該檢閱任何警告。

若要啟用管道，請使用下列 activate-pipeline 命令。

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

您可以使用下列 list-pipelines 命令，驗證管道是否顯示在管道清單中。

```
aws datapipeline list-pipelines
```

使用 將資料複製到 Amazon Redshift AWS Data Pipeline

本教學課程將逐步引導您建立管道，以定期使用 AWS Data Pipeline 主控台中的複製到 Redshift 範本，或使用 CLI 將資料從 Amazon S3 移至 AWS Data Pipeline Amazon Redshift 的管道定義檔案。

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

Amazon Redshift 是雲端中的資料倉儲服務。如需詳細資訊，請參閱 Amazon Redshift 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 內將資料複製到 Amazon Redshift 之前 AWS Data Pipeline，請確定您：

- 從 Amazon S3 載入資料。
- 在 Amazon Redshift 中設定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 COPY。

如需從 Amazon S3 載入資料的步驟，請參閱《Amazon 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 命令會建立新的資料表，名為 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、建立安全群組和建立 Amazon Redshift 叢集

針對教學設定

1. 完成 設定 的 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-Classic】 選取 No VPC VPC。
   f. [EC2-VPC] 針對 VPC 選取您 VPC 的 ID。
   g. 按一下 Create (建立)。
3. 【EC2-Classic】 建立 Amazon Redshift 叢集安全群組，並指定 Amazon EC2 安全群組。
   a. 開啟 Amazon Redshift 主控台。
   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，從連線類型中選取 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 Amazon Redshift 叢集安全群組，並指定 VPC 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. 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。

   若要建立新的叢集並載入範例資料，請遵循 Amazon Redshift 入門中的步驟。如需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. 開啟 Amazon Redshift 主控台。

   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 Amazon Redshift

本教學課程示範如何將資料從 Amazon S3 複製到 Amazon Redshift。您將在 Amazon Redshift 中建立新的資料表，然後使用 從公有 Amazon S3 儲存貯體 AWS Data Pipeline 傳輸資料至此資料表，其中包含 CSV 格式的範例輸入資料。日誌會儲存到您擁有的 Amazon S3 儲存貯體。

Amazon S3 是一種 Web 服務，可讓您將資料存放在雲端。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南。Amazon Redshift 是雲端中的資料倉儲服務。如需詳細資訊，請參閱 Amazon Redshift 管理指南。

先決條件

開始之前，您必須完成下列步驟：

1. 安裝和設定命令列界面 (CLI)。如需詳細資訊，請參閱存取 AWS Data Pipeline。
2. 確保名為 DataPipelineDefaultRole 和 DataPipelineDefaultResourceRole 的 IAM 角色存在。AWS Data Pipeline 主控台會自動為您建立這些角色。如果您至少尚未使用 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，請參閱的 IAM 角色 AWS Data Pipeline。
3. 在 Amazon Redshift 中設定 COPY命令，因為當您在其中執行複製時，將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. 設定 Amazon Redshift 資料庫。如需詳細資訊，請參閱設定管道、建立安全群組和建立 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

此範例案例示範如何將資料從 Amazon S3 儲存貯體複製到 Amazon Redshift。

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S3DataNode。

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管道元件會定義輸出資料的位置；在此情況下，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`

Amazon Redshift 資料庫的參考。

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。在此範例中， AWS Data Pipeline 應該自動建立 EC2 執行個體來執行複製任務，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 資源之帳戶的 IAM 角色，例如存取 Amazon S3 儲存貯體以擷取資料。

`logUri`

從 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。Role 和 ResourceRole 可以是相同的角色，但不同的角色可提高您安全組態的精細程度。

活動

JSON 檔案的最後部分是代表所要執行工作的活動定義。在此情況下，我們使用`RedshiftCopyActivity`元件將資料從 Amazon S3 複製到 Amazon Redshift。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`

Amazon 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`

Amazon Redshift 目的地資料表的參考。

上傳和啟用管道定義

您必須上傳管道定義並啟用管道。在下列範例命令中，將 *pipeline_name* 取代為管道的標籤，並將 *pipeline_file* 取代為管道定義.json檔案的完整路徑。

AWS CLI

若要建立管道定義並啟用管道，請使用下列 create-pipeline 命令。請注意管道的 ID，因為您會將此值與大多數 CLI 命令搭配使用。

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

若要上傳管道定義，請使用下列 put-pipeline-definition 命令。

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

如果您管道驗證成功， validationErrors 欄位會是空的。您應該檢閱任何警告。

若要啟用管道，請使用下列 activate-pipeline 命令。

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

您可以使用下列 list-pipelines 命令，驗證管道是否顯示在管道清單中。

```
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
- Period
- String

DateTime

AWS Data Pipeline 僅支援 UTC/GMT 格式的「YYYY-MM-DDTHH：MM：SS」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 同時支援整數和浮點值。

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 node 的相關資訊，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。

Period

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"
```

String

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://amzn-s3-demo-bucket/#{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，確保所有表達式都以表達式的值取代。

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槽 ID，例如 Amazon S3 備份位置的槽 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://amzn-s3-demo-bucket/ExampleDataNode.csv"。

```
{
  "id" : "ExampleDataNode",
  "type" : "S3DataNode",
  "schedule" : {"ref" : "ExampleSchedule"},
  "filePath" : "s3://amzn-s3-demo-bucket/#{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 可讓您巢狀值以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}，則會評估為 ["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 中使用#{node.*}表達式，PreCondition以回溯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的傳輸參考，但不支援執行時間欄位。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，請參閱使用者定義的欄位。

```
{
  "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，您可以用來計算欄位的值。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。

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

函式	Description
	結果：4.0

字串函數

下列函數可用於處理字串值。

函式	Description
+	串連。非字串值會先轉換成字串。 範例：#{"hel" + "lo"} 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**
>
> 的日期/時間格式 AWS Data Pipeline 是 Joda Time，這是 Java 日期和時間類別的替代項目。如需詳細資訊，請參閱 Joda Time - Class DateTimeFormat。

函式	Description
int day(DateTime myDateTime)	取得 DateTime 值的日 (以整數表示)。 範例：#{day(myDateTime)} 結果：24

函式	Description
`int dayOfYear(DateTime myDateTime)`	取得 DateTime 值的年度日 (以整數表示)。 範例：`#{dayOfYear(myDateTime)}` 結果：`144`
`DateTime firstOfMonth(DateTime myDateTime)`	在指定的 DateTime 中建立月份起始的 DateTime 物件。 範例：`#{firstOfMonth(myDateTime)}` 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。 範例：`#{format(myDateTime,'YYYY-MM-dd HH:mm:ss z')}` 結果：`"2011-05-24T17:10:00 UTC"`
`int hour(DateTime myDateTime)`	取得 DateTime 值的小時 (以整數表示)。 範例：`#{hour(myDateTime)}` 結果：`17`

函式	Description
`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。 範例：`#{makeDate(2011,5,24)}` 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。 範例：`#{makeDateTime(2011,5,24,14,21)}` 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： 範例：`#{midnight(myDateTime)}` 結果：`"2011-05-25T0:00:00z"`

函式	Description
`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。 範例：`#{minusDays(myDateTime,1)}` 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。 範例：`#{minusHours(myDateTime,1)}` 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。 範例：`#{minusMinutes(myDateTime,1)}` 結果：`"2011-05-24T17:09:00z"`

函式	Description
`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。 範例：`#{minusMonths(myDateTime,1)}` 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。 範例：`#{minusWeeks(myDateTime,1)}` 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。 範例：`#{minusYears(myDateTime,1)}` 結果：`"2010-05-24T17:10:00z"`
`int minute(DateTime myDateTime)`	取得 DateTime 值的分鐘 (以整數表示)。 範例：`#{minute(myDateTime)}` 結果：`10`

函式	Description
`int month(DateTime myDateTime)`	取得 DateTime 值的月 (以整數表示)。 範例：`#{month(myDateTime)}` 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。 範例：`#{plusDays(myDateTime,1)}` 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。 範例：`#{plusHours(myDateTime,1)}` 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。 範例：`#{plusMinutes(myDateTime,1)}` 結果：`"2011-05-24 17:11:00z"`

函式	Description
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。 範例：#{plusMonths(myDateTime,1)} 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。 範例：#{plusWeeks(myDateTime,1)} 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。 範例：#{plusYears(myDateTime,1)} 結果："2012-05-24T17:10:00z"

函式	Description
`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。 範例：`#{sunday(myDateTime)}` 結果：`"2011-05-22 17:10:00 UTC"`
`int year(DateTime myDateTime)`	取得 DateTime 值的年 (以整數表示)。 範例：`#{year(myDateTime)}` 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。 範例：`#{yesterday(myDateTime)}` 結果：`"2011-05-23T17:10:00z"`

特殊字元

AWS Data Pipeline 使用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	Description	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 @failureReason @resourceStatus
#	表達式。運算式以 "#{" 和 "}" 分隔，括號的內容則由 評估 AWS Data Pipeline。如需詳細資訊，請參閱表達式。	#{format(myDateTime,'YYYY-MM-dd hh:mm:ss')} s3：//amzn-s3-demo-bucket/#{id}.csv
*	加密欄位。此字元是欄位名稱字首，表示 AWS Data Pipeline 應該加密主控台或 CLI 與 AWS Data Pipeline 服務之間傳輸的此欄位內容。	*password

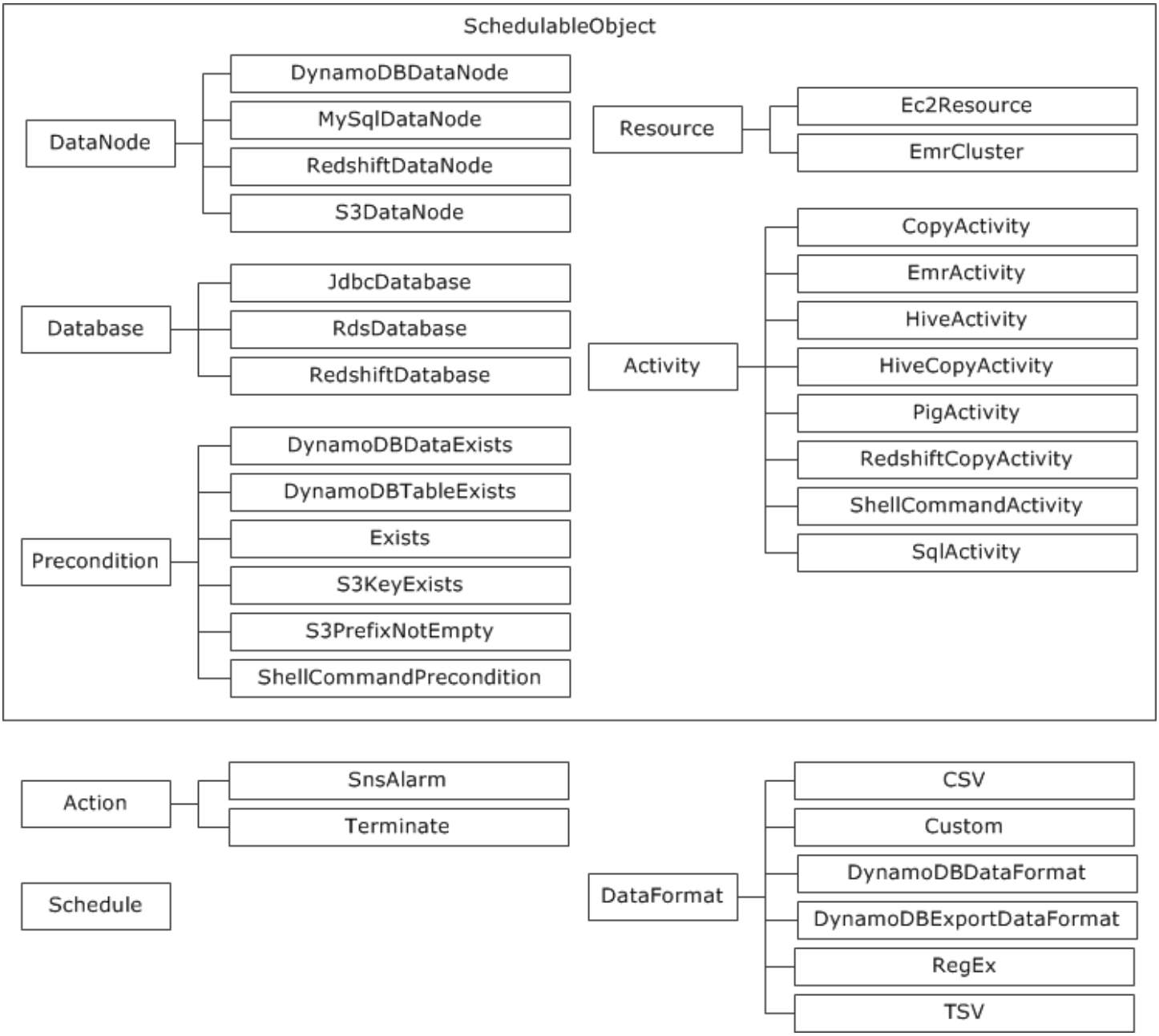
管道物件參考

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 開發套件的範例應用程式，請參閱 GitHub 上的資料管道 DynamoDB 匯出 Java 範例。

以下是 的物件階層 AWS Data Pipeline。



資料節點

以下是 AWS Data Pipeline 資料節點物件：

物件

- DynamoDBDataNode
- MySqlDataNode
- RedshiftDataNode

- S3DataNode
- SqlDataNode

DynamoDBDataNode

使用 DynamoDB 定義資料節點，該 DynamoDB 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	Description	槽類型
tableName	DynamoDB 資料表。	String

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程	參考物件，例如 "schedule":{"ref": "myScheduleId"}

物件呼叫欄位	Description	槽類型
	以滿足這項需求，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，請參閱排程。	

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
dataFormat	此資料節點描述之資料的 DataFormat。目前支援 HiveActivity 和 HiveCopyActivity。	參考物件，"dataFormat":{"ref":"myDynamoDBDataFormatId"}
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"}
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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 在 Hive 中執行 DynamoDB 資料表接移時會使用此項目。	列舉
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

選用欄位	Description	槽類型
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。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 runsOn 值，且 workerGroup 存在，則會忽略 workerGroup。	String
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	Description	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

執行時間欄位	Description	槽類型
cancellationReason	若此物件已取消，會提供 cancellationReason。	String
@cascadeFailedOn	物件失敗所在相依性鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

MySqlDataNode

定義使用 MySQL 的資料節點。

> **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	Description	槽類型
資料表	MySQL 資料庫中的資料表名稱。	String

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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/	參考物件，例如 "schedule":{"ref": "myScheduleId"}

物件呼叫欄位	Description	槽類型
	datapipeline/latest/DeveloperGuide/dp-object-schedule.html。	

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
createTableSql	可建立資料表的 SQL create table 表達式。	String
資料庫	資料庫的名稱。	參考物件，例如 "database":{"ref": "myDatabaseId"}
dependsOn	指定與其他可執行物件的相依性。	參考物件，例如 "dependsOn":{"ref" :"myActivityId"}
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為。	列舉
insertQuery	可將資料載入資料表的 SQL 陳述式。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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":"m yActionId"}

選用欄位	Description	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，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，例如 "runsOn":{"ref":"myResourceId"}

選用欄位	Description	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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉
schemaName	保留資料表的結構描述名稱	String
selectQuery	可從資料表擷取資料的 SQL 陳述式。	String
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。	String

執行時間欄位	Description	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。	String
@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	Description	槽類型
		{"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- S3DataNode

RedshiftDataNode

使用 Amazon Redshift 定義資料節點。 `RedshiftDataNode`代表資料庫內資料的屬性，例如您的管道所使用的資料表。

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	Description	槽類型
資料庫	資料表所在的資料庫。	參考物件，例如 "database":{"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 createTableSql，則會建立資料表。	String

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
createTableSql	在資料庫建立資料表的 SQL 表達式。我們建議您指定應建立資料表的結構描述，例如：CREATE TABLE mySchema.myTable (bestColumn varchar(25) 主索引鍵 distkey， numberOfWins integer sortKey)。如果 tableName 指定的資料表不存在於結構schemaName中，則 會在 createTableSql 欄位中 AWS Data Pipeline 執行指令碼。例如，如果您將 schemaName 指定為 mySchema，但 createTableSql 欄位中未包含 mySchema，則建立資料表的結構描述錯誤 (預設會在 PUBLIC 中建立)。這是因為 AWS Data Pipeline 不會剖析您的 CREATE TABLE 陳述式。	String
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"}
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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	Description	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，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"ref":"myPreconditionId"}
primaryKeys	如果您未指定 `RedShiftCopyActivity` 目的地資料表的 primaryKeys，您可以使用將做為 mergeKey 的 primaryKeys 來指定資料行清單。不過，如果您在 Amazon Redshift 資料表中定義了現有的 primaryKey，則此設定會覆寫現有的金鑰。	String
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，例如 "runsOn":{"ref":"myResourceId"}

選用欄位	Description	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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。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，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》。	String
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。	String

執行時間欄位	Description	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。	String

執行時間欄位	Description	槽類型
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

S3DataNode

使用 Amazon S3 定義資料節點。根據預設，S3DataNode 會使用伺服器端加密。若您要停用此設定，請將 s3EncryptionType 設為 NONE。

> **Note**
>
> 當您使用 `S3DataNode` 做為針對 `CopyActivity` 的輸入時，僅支援 CSV 及 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://amzn-s3-demo-bucket/#{@scheduledStartTime}.csv"
}
```

語法

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period

選用欄位	Description	槽類型
壓縮	S3DataNode 所描述的資料壓縮類型。"none" 表示未使用任何壓縮，而 "gzip" 表示使用 gzip 演算法壓縮。此欄位僅支援使用 Amazon Redshift，以及當您搭配使用 S3DataNode 和 CopyActivity 時。	列舉
dataFormat	此 S3DataNode 描述之資料的 DataFormat。	參考物件，例如 "dataFormat":{"ref":"myDataFormatId"}
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"}
directoryPath	Amazon S3 目錄路徑 URI：s3://my-bucket/my-key-for-directory。您必須提供 filePath 或 directoryPath 值。	String
failureAndRerunMode	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 中的物件路徑 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 directoryPath 值，以容納目錄中的多個檔案。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
manifestFilePath	資訊清單檔案的 Amazon S3 路徑，採用 Amazon Redshift 支援的格式。 AWS Data Pipeline 會使用資訊清單檔案將指定的 Amazon S3 檔案複製到資料表。此欄位僅在 RedShiftCopyActivity 參考 S3DataNode 時有效。	String
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

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，例如 "runsOn":{"ref":"myResourceId"}

選用欄位	Description	槽類型
s3EncryptionType	覆寫 Amazon S3 加密類型。值是 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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。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 runsOn 值，且 workerGroup 存在，則會忽略 workerGroup。	String

執行時間欄位	Description	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。	String
@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	Description	槽類型
		{"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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	Description	槽類型
資料表	SQL 資料庫中的資料表名稱。	String

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
createTableSql	可建立資料表的 SQL create table 表達式。	String
資料庫	資料庫的名稱。	參考物件，例如 "database":{"ref": "myDatabaseId"}
dependsOn	指定與其他可執行物件的相依性。	參考物件，例如 "dependsOn":{"ref" :"myActivityId"}
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為。	列舉
insertQuery	可將資料載入資料表的 SQL 陳述式。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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":"m yActionId"}
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":{"r ef":"myActionId"}

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，例如 "runsOn":{"ref":"myResourceId"}

選用欄位	Description	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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉
schemaName	保留資料表的結構描述名稱	String
selectQuery	可從資料表擷取資料的 SQL 陳述式。	String
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。	String

執行時間欄位	Description	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。	String
@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	Description	槽類型
		{"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

另請參閱

- S3DataNode

活動

以下是 AWS Data Pipeline 活動物件：

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 支援 S3DataNode 和 SqlDataNode 做為輸入及輸出，並且複製操作一般會針對每筆記錄逐一執行。不過，當符合下列所有條件時，CopyActivity會提供高效能的 Amazon S3 到 Amazon S3 複本：

- 輸入及輸出皆為 S3DataNode。
- 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，以及將檔案複製到目錄，但逐筆記錄複製會在將目錄複製到檔案時發生。最後， CopyActivity 不支援複製分段 Amazon S3 檔案。

CopyActivity 的 CSV 支援具有特定限制。當您使用 S3DataNode 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。
- 預設逸出字元是 ASCII 值 92 (反斜線)。
- 記錄結尾識別符是 ASCII 值 10 (或 "\n")。

Windows 類型系統通常會使用不同的記錄結尾字元序列：一個歸位字元及一個換行字元 (ASCII 值 13 及 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。

使用 CopyActivity 從 PostgreSQL RDS 物件匯出到 TSV 資料格式時，預設 NULL 字元是 \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	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，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。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
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"}
failureAndRerunMode	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。	參考物件，例如 "input":{"ref":"myDataNodeId"}
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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	Description	槽類型
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。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

選用欄位	Description	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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉

執行時間欄位	Description	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。	String
@cascadeFailedOn	物件失敗所在相依性鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnableObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String

執行時間欄位	Description	槽類型
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件使用的管道版本。	String

執行時間欄位	Description	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" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

另請參閱

- ShellCommandActivity
- EmrActivity
- 使用 將 MySQL 資料匯出至 Amazon S3 AWS Data Pipeline

EmrActivity

執行 EMR 叢集。

AWS Data Pipeline 針對步驟使用與 Amazon EMR 不同的格式；例如，在`EmrActivity`步驟欄位中的 JAR 名稱後面 AWS Data Pipeline 使用逗號分隔引數。下列範例顯示針對 Amazon EMR 格式化的步驟，後面接著其 AWS Data Pipeline 對等項目：

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

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

範例

以下為此物件類型的範例。此範例使用舊版 Amazon EMR。驗證此範例與您正在使用的 Amazon EMR 叢集版本是否正確。

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**
>
> 在此範例中，您可以將 step 欄位取代成您所需的叢集字串，該字串可以是 Pig 指令碼、Hadoop 串流叢集、您自己的 JAR (包含參數) 等。

Hadoop 2.x (AMI 3.x)

```
{
  "id" : "MyEmrActivity",
  "type" : "EmrActivity",
  "runsOn" : { "ref" : "MyEmrCluster" },
  "preStepCommand" : "scp remoteFiles localFiles",
  "step" : ["s3://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg,-files,s3://
amzn-s3-demo-bucket/myPath/myFile.py,-input,s3://myinputbucket/path,-output,s3://
myoutputbucket/path,-mapper,myFile.py,-reducer,reducerName","s3://amzn-s3-demo-bucket/
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，若您在 JSON 使用 `File\.gz` 做為引數，您可以使用 `File\\\\.gz` 來逸出它。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	Description	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"}

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	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` 。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
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"}
failureAndRerunMode	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。	參考物件，例如 "input":{"ref":"myDataNodeId"}
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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"}

選用欄位	Description	槽類型
output	輸出資料的位置。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	用於上傳管道日誌的 Amazon S3 URI，例如 's3：//BucketName/Prefix/'。	String
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。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	String
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。	Period

選用欄位	Description	槽類型
resizeClusterBeforeRunning	在執行此活動之前調整叢集的大小，以容納指定為輸入或輸出的 DynamoDB 資料表。 **Note** 如果您的 EmrActivity 使用 DynamoDBDataNode 做為輸入或輸出資料節點，而且如果您resizeClusterBeforeRunning 將 設定為 TRUE，則 會使用m3.xlarge 執行個體類型 AWS Data Pipeline 開始。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
resizeClusterMaxInstances	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。	Period
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

選用欄位	Description	槽類型
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 名稱後方，使用逗號分隔的引數，例如 `s3://amzn-s3-demo-bucket/MyWork.jar, arg1,arg2,arg3` 。	String

執行時間欄位	Description	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。	String
@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	String
errorId	若此物件失敗，會提供 `errorId`。	String
errorMessage	若此物件失敗，會提供 `errorMessage` 。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String

執行時間欄位	Description	槽類型
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件時使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- ShellCommandActivity
- CopyActivity
- EmrCluster

HadoopActivity

在叢集上執行 MapReduce 任務。叢集可以是受 AWS Data Pipeline 管理的 EMR 叢集，或是另一個資源 (若您使用 TaskRunner 的話)。請在您希望平行執行工作時使用 HadoopActivity。這可讓您使用 YARN 框架的排程資源，或是 Hadoop 1 中的 MapReduce 資源交涉程式。如果您想要使用 Amazon EMR Step 動作依序執行工作，您仍然可以使用 EmrActivity。

範例

使用 管理的 EMR 叢集的 HadoopActivity AWS Data Pipeline

以下 HadoopActivity 物件會使用 EmrCluster 資源來執行程式：

```
{
  "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://amzn-s3-demo-bucket/MyHadoopActivity/#{@pipelineId}/
#{format(@scheduledStartTime,'YYYY-MM-dd')}"
   ],
   "maximumRetries": "0",
   "postActivityTaskConfig":{"ref":"postTaskScriptConfig”},
   "hadoopQueue" : “high”
 }
```

以下是對應的 *MyEmrCluster*，它會為 Hadoop 2 類型的 AMI 設定 YARN 中的 FairScheduler 和佇列。

```
{
  "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”]
}
```

此為您用來在 Hadoop 1 中設定 FairScheduler 的 EmrCluster：

```
{
      "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 EmrCluster 會為 Hadoop 2 類型的 AMI 設定 CapacityScheduler：

```
{
      "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"
    }
```

使用現有 EMR 叢集的 HadoopActivity

在此範例中，您可以使用工作者群組和 TaskRunner 來執行現有 EMR 叢集上的程式。以下管道定義會使用 HadoopActivity 來：

- 只在 *myWorkerGroup* 資源上執行 MapReduce 程式。如需工作者群組的詳細資訊，請參閱使用任務執行器在現有資源上執行工作。
- 執行 preActivityTaskConfig 和 postActivityTaskConfig

```
{
  "objects": [
    {
      "argument": [
        "-files",
        "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
        "-mapper",
        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input/",
        "-output",
        "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/scripts/preTaskScript.sh",
      "name": "preTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "ShellScriptConfig",
      "scriptUri": "s3://amzn-s3-demo-bucket/scripts/postTaskScript.sh",
      "name": "postTaskScriptConfig",
      "scriptArgument": [
        "test",
        "argument"
      ],
      "type": "ShellScriptConfig"
    },
    {
      "id": "Default",
      "scheduleType": "cron",
      "schedule": {
        "ref": "SchedulePeriod"
      },
      "name": "Default",
      "pipelineLogUri": "s3://amzn-s3-demo-bucket/
logs/2015-05-22T18:02:00.343Z642f3fe415",
      "maximumRetries": "0",
      "workerGroup": "myWorkerGroup",
      "preActivityTaskConfig": {
        "ref": "preTaskScriptConfig"
      },
```

```
      "postActivityTaskConfig": {
        "ref": "postTaskScriptConfig"
      }
    }
  ]
}
```

語法

必要欄位	Description	槽類型
jarUri	在 Amazon S3 或叢集的本機檔案系統中，要搭配執行 HadoopActivity 的 JAR 位置。	String

物件呼叫欄位	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	要在其中執行此任務的 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。	String

選用欄位	Description	槽類型
argument	要傳遞給 JAR 的引數。	String
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
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"}
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
hadoopQueue	要提交活動至其中的 Hadoop 排程器佇列名稱。	String
input	輸入資料的位置。	參考物件，例如 "input":{"ref":"myDataNodeId"}
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
mainClass	搭配執行 HadoopActivity 的 JAR 主要類別。	String

選用欄位	Description	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。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
postActivityTaskConfig	要執行的活動後組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "postActivityTaskConfig":{"ref":"myShellScriptConfigId"}
preActivityTaskConfig	要執行的活動前組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "preActivityTaskConfig":{"ref":"myShellScriptConfigId"}

選用欄位	Description	槽類型
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉

執行時間欄位	Description	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

執行時間欄位	Description	槽類型
cancellationReason	若此物件已取消，會提供 cancellationReason。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- ShellCommandActivity
- CopyActivity
- EmrCluster

HiveActivity

在 EMR 叢集上執行 Hive 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 RDS 的輸入資料自動建立 Hive 資料表。您只需要指定要在來源資料上執行的 HiveQL。 會根據HiveActivity物件中的輸入欄位，以 ${input1}、 ${input2}等方式 AWS Data Pipeline 自動建立 Hive 資料表。

對於 Amazon S3 輸入， dataFormat 欄位用於建立 Hive 資料欄名稱。

對於 MySQL (Amazon RDS) 輸入，SQL 查詢的資料欄名稱用於建立 Hive 資料欄名稱。

> **Note**
>
> 此活動使用 Hive CSV Serde。

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	Description	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	參考物件，例如 "schedule":{"ref": "myScheduleId"}

物件呼叫欄位	Description	槽類型
	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。	

必要的群組 (下列其中之一為必要)	Description	槽類型
hiveScript	要執行的 Hive 指令碼。	String
scriptUri	要執行的 Hive 指令碼位置 (例如 s3://scriptLocation)。	String

必要群組	Description	槽類型
runsOn	要在其上執行此 `HiveActivity` 的 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` 。	String
input	輸入資料來源。	參考物件，例如 "input":{"ref":"myDataNodeId"}

必要群組	Description	槽類型
output	輸出資料來源。	參考物件，例如 "output":{"ref":"myDataNodeId"}

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
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"}
failureAndRerunMode	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。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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"}

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
postActivityTaskConfig	要執行的活動後組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "postActivityTaskConfig":{"ref":"myShellScriptConfigId"}
preActivityTaskConfig	要執行的活動前組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "preActivityTaskConfig":{"ref":"myShellScriptConfigId"}
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period

選用欄位	Description	槽類型
resizeClusterBeforeRunning	在執行此活動之前調整叢集的大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。 **Note** 如果您的活動使用 DynamoDBDataNode 做為輸入或輸出資料節點，而且如果您resizeClusterBeforeRunning 將 設定為 TRUE，則 會使用m3.xlarge 執行個體類型 AWS Data Pipeline 開始。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
resizeClusterMaxInstances	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。	Period
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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉

選用欄位	Description	槽類型
scriptVariable	指定執行指令碼時要傳遞給 Hive 的 Amazon EMR 指令碼變數。例如，以下範例指令碼變數會將 SAMPLE 和 FILTER_DATE 變數傳遞給 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。	String
stage	決定在執行指令碼之前或之後是否啟用預備。Hive 11 不允許，因此請使用 Amazon EMR AMI 3.2.0 版或更新版本。	Boolean

執行時間欄位	Description	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。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnableObjectId"}

執行時間欄位	Description	槽類型
emrStepLog	Amazon EMR 步驟日誌僅適用於 EMR 活動嘗試。	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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

執行時間欄位	Description	槽類型
@status	此物件的狀態。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- ShellCommandActivity
- EmrActivity

HiveCopyActivity

在 EMR 叢集上執行 Hive 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，以在資料欄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	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	指定要在其中執行的叢集。	參考物件，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` 。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
dependsOn	指定與其他可執行物件的相依性。	參考物件，例如 "dependsOn":{"ref":"myActivityId"}
failureAndRerunMode	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 或 Amazon S3 資料子集。篩選條件應該只包含述詞，而不是以`WHERE`子句開頭，因為 會自動 AWS Data Pipeline 新增它。	String
input	輸入資料來源。此必須為 `S3DataNode` 或 `DynamoDBDataNode` 。如果您使用 `DynamoDBNode` ，請指定 `DynamoDBExportDataFormat` 。	參考物件，例如 "input":{"ref":"myDataNodeId"}

選用欄位	Description	槽類型
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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。如果輸入是 S3DataNode ，這必須為 DynamoDBDataNode 。否則，此項目可以是 S3DataNode 或 DynamoDBDataNode 。如果您使用 DynamoDBNode ，請指定 DynamoDBExportDataFormat 。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	用於上傳管道日誌的 Amazon S3 URI 's3://BucketName/Key/' ，例如 。	String
postActivityTaskConfig	要執行的活動後組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "postActivityTaskConfig":{"ref":"myShellScriptConfigId"}

選用欄位	Description	槽類型
preActivityTaskConfig	要執行的活動前組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "preActivityTaskConfig":{"ref":"myShellScriptConfigId"}
precondition	可選擇性定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period
resizeClusterBeforeRunning	在執行此活動之前調整叢集的大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。 **Note** 如果您的活動使用 `DynamoDBDataNode` 做為輸入或輸出資料節點，而且如果您`resizeClusterBeforeRunning` 將 設定為 `TRUE`，則 會使用`m3.xlarge` 執行個體類型 AWS Data Pipeline 開始。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
resizeClusterMaxInstances	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。	Period

選用欄位	Description	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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉

執行時間欄位	Description	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。	String
@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。	String
errorId	若此物件失敗，會提供 errorId。	String

執行時間欄位	Description	槽類型
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件使用的管道版本。	String

執行時間欄位	Description	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" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- ShellCommandActivity
- EmrActivity

PigActivity

PigActivity 在 中為 Pig 指令碼提供原生支援， AWS Data Pipeline 無需使用 `ShellCommandActivity`或 `EmrActivity`。此外，PigActivity 支援資料預備。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，以選取幾欄資料並將其上傳至 Amazon S3。

- MyPigActivity2 會載入第一個輸出、選取幾欄和三列資料，並將其上傳到 Amazon S3 做為第二個輸出。
- MyPigActivity3 會載入第二個輸出資料、將兩列資料插入 Amazon RDS，以及僅將名為 "fifth" 的資料欄插入 Amazon RDS。
- MyPigActivity4 會載入 Amazon RDS 資料、選取第一列資料，並將其上傳至 Amazon S3。

```
{
  "objects": [
    {
      "id": "MyInputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "directoryPath": "s3://amzn-s3-demo-bucket/pigTestInput",
      "name": "MyInputData1",
      "dataFormat": {
        "ref": "MyInputDataType1"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyPigActivity4",
      "scheduleType": "CRON",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
      "input": {
        "ref": "MyOutputData3"
      },
      "pipelineLogUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-bucket/PigActivityOutput2",
      "dataFormat": {
        "ref": "MyOutputDataType2"
      },
      "type": "S3DataNode"
    },
    {
      "id": "MyOutputData1",
      "schedule": {
        "ref": "MyEmrResourcePeriod"
      },
```

```
      "name": "MyOutputData1",
      "directoryPath": "s3://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-bucket/path",
      "scriptUri": "s3://amzn-s3-demo-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	Description	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。使用者可以明確設定物件的排程以滿足這項需求，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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
script	要執行的 Pig 指令碼。	String
scriptUri	要執行 Pig 指令碼的位置 (例如 s3://scriptLocation)。	String

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	要在其中執行此 PigActivity 的 EMR 叢集。	參考物件，例如 "runsOn":{"ref":"myEmrClusterId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
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 。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
dependsOn	指定與其他可執行物件的相依性。	參考物件，例如 "dependsOn":{"ref":"myActivityId"}
failureAndRerunMode	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。	參考物件，例如 "input":{"ref":"myDataNodeId"}
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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	Description	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"}
output	輸出資料來源。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	用於上傳管道日誌的 Amazon S3 URI（例如 's3：//BucketName/Key/')。	String
postActivityTaskConfig	要執行的活動後組態指令碼。這包含 Amazon S33 中 shell 指令碼的 URI 和引數清單。	參考物件，例如 "postActivityTaskConfig":{"ref":"myShellScriptConfigId"}
preActivityTaskConfig	要執行的活動前組態指令碼。這包含 Amazon S3 中的 shell 指令碼 URI 和引數清單。	參考物件，例如 "preActivityTaskConfig":{"ref":"myShellScriptConfigId"}
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	Period

選用欄位	Description	槽類型
resizeClusterBeforeRunning	在執行此活動之前調整叢集的大小，以容納指定為輸入或輸出的 DynamoDB 資料節點。 **Note** 如果您的活動使用 DynamoDBDataNode 做為輸入或輸出資料節點，而且如果您resizeClusterBeforeRunning 將 設定為 TRUE，則 會使用m3.xlarge 執行個體類型 AWS Data Pipeline 開始。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
resizeClusterMaxInstances	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。	Period
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。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，您只要針對每次後續執行呼叫 ActivatePipeline 操作即可。值為：Cron、ondemand 和 timeseries。	列舉
scriptVariable	要傳遞給 Pig 指令碼的引數。您可以搭配使用 scriptVariable 和 script 或 scriptUri。	String

選用欄位	Description	槽類型
stage	決定是否啟用接移，並讓您的 Pig 指令碼存取暫存資料的資料表，例如 ${INPUT1} 和 ${OUTPUT1}。	Boolean

執行時間欄位	Description	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。	String
@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。	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String

執行時間欄位	Description	槽類型
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件時使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

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. 首先使用 AWS Data Pipeline 在 Amazon S3 中暫存您的資料。
2. 使用 將資料從 Amazon RDS 和 Amazon EMR `RedshiftCopyActivity`移至 Amazon Redshift。

   這可讓您將資料載入 Amazon Redshift，以便進行分析。
3. 使用 SqlActivity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S3DataNode。

範例

以下為此物件類型的範例。

為了確保格式轉換，此範例使用 中的 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 將資料複製到 Amazon Redshift AWS Data Pipeline。

語法

必要欄位	Description	槽類型
insertMode	決定 AWS Data Pipeline 如何處理目標資料表中與要載入資料中的資料列重疊的預先存在資料。 有效值為：KEEP_EXISTING 、OVERWRITE_EXISTING 、TRUNCATE 和 APPEND。 KEEP_EXISTING 會將新列新增至資料表，並保持任何現有列不變。 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。 TRUNCATE 會刪除目的地資料表中的所有資料，再寫入新資料。 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。	列舉

物件呼叫欄位	Description	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。 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":	參考物件，例如： "schedule":{"ref": "mySchedu leId"}

物件呼叫欄位	Description	槽類型
	`"DefaultSchedule"}` 在物件上明確設定排程。 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。 如需範例選用排程組態的詳細資訊，請參閱排程。	

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，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。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
commandOptions	接受參數以在`COPY`操作期間傳遞至 Amazon Redshift 資料節點。如需參數的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 COPY。 `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	String

選用欄位	Description	槽類型
	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。 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。 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。 此外，在某些情況下，當需要從 Amazon Redshift 叢集卸載資料並在 Amazon S3 中建立檔案時，RedshiftCopyActivity 依賴 Amazon Redshift UNLOAD的操作。 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。如需參數的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 UNLOAD。	
dependsOn	指定與另一個可執行物件的相依性。	參考物件：`"dependsOn":{"ref":"myActivityId"}`
failureAndRerunMode	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"}`

選用欄位	Description	槽類型
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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：`"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	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String

選用欄位	Description	槽類型
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件：`"precondition":{"ref":"myPreconditionId"}`
佇列	對應至 Amazon Redshift 中的 `query_group` 設定，可讓您根據並行活動在佇列中的置放來指派並排定其優先順序。 Amazon Redshift 會將同時連線數限制在 15。如需詳細資訊，請參閱《Amazon RDS 資料庫開發人員指南》中的將查詢指派給佇列。	String
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

選用欄位	Description	槽類型
scheduleType	允許您指定管道中物件的排程。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。這表示您不必複製或重新建立管道，然後再執行一次。 若要使用 ondemand 管道，請針對每次後續執行呼叫 ActivatePipeline 操作。 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
transformSql	用於轉換輸入資料的 SQL SELECT 表達式。 在資料表執行名為 staging 的 transformSql 表達式。 當您從 DynamoDB 或 Amazon S3 複製資料時， AWS Data Pipeline 會建立名為「預備」的資料表，並最初載入其中的資料。此資料表中的資料用於更新目標資料表。 transformSql 的輸出結構描述，必須與最終目標表格的結構描述相符。 如果您指定 transformSql 選項，則會從指定的 SQL 陳述式建立第二個臨時資料表。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。	String

執行時間欄位	Description	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	若此物件已取消，會提供 cancellationReason。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件：`"cascadeFailedOn":{"ref":"myRunnableObjectId"}`
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String

執行時間欄位	Description	槽類型
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件： `"waitingOn": {"ref":"myRunnableObjectId"}`

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String

系統欄位	Description	槽類型
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

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 搭配使用時S3DataNode， ShellCommandActivity支援預備資料的概念，這表示您可以將資料從 Amazon S3 移至階段位置，例如 Amazon EC2 或您的本機環境、使用指令碼和 對資料執行工作ShellCommandActivity，並將其移回 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，殼層命令的輸出可以暫存在輸出目錄中，以自動推送到 Amazon S3，由 ${OUTPUT1_STAGING_DIR}、 ${OUTPUT2_STAGING_DIR}等參考。

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	Description	槽類型
schedule	在 `schedule` 間隔的執行期間會呼叫此物件。 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。 若要滿足這項需求，請明確設定物件的 `schedule`，例如指定 `"schedule": {"ref": "DefaultSchedule"}` 。 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。 若要分散負載， 會稍微提前 AWS Data Pipeline 建立實體物件，但會按排程執行。 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"}

必要的群組 (下列其中之一為必要)	Description	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。	String
scriptUri	要下載並以 shell 命令執行之檔案的 Amazon S3 URI 路徑。僅指定一個 `scriptUri` 或	String

必要的群組 (下列其中之一為必要)	Description	槽類型
	command 欄位。scriptUri 無法使用參數，請改為使用 command。	

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	執行活動或命令的運算資源，例如 Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，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 。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
dependsOn	指定與其他可執行物件的相依性。	參考物件，例如 "dependsOn":{"ref":"myActivityId"}
failureAndRerunMode	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。	參考物件，例如 "input":{"ref":"myDataNodeId"}

選用欄位	Description	槽類型
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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尚未完成時，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。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
pipelineLogUri	Amazon S3 URI，例如`'s3://BucketName/Key/'` 用於上傳管道的日誌。	String
precondition	可選擇性定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"ref":"myPreconditionId"}

選用欄位	Description	槽類型
reportProgressTimeout	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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
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。 值為：cron、ondemand 和 timeseries 。 如果設為 timeseries ，則執行個體會排程在每個間隔的結尾。 如果設為 Cron，則執行個體會排程在每個間隔的開頭。 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	JSON 格式的字串陣列，用於傳遞給由命令指定的命令。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 ： "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。	String

選用欄位	Description	槽類型
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
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 `runsOn` 欄位，這必須是 Amazon S3 路徑，因為執行活動之資源的暫時性性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。	String
stdout	從 命令接收重新導向輸出的 Amazon S3 路徑。如果您使用 `runsOn` 欄位，這必須是 Amazon S3 路徑，因為執行活動之資源的暫時性性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。	String

執行時間欄位	Description	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` 。	String
@cascadeFailedOn	物件失敗所在之相依鏈的描述。	參考物件，例如 "cascadeFailedOn":

執行時間欄位	Description	槽類型
		{"ref":"myRunnable ObjectId"}
emrStepLog	Amazon EMR 步驟日誌僅適用於 Amazon EMR 活動嘗試。	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage 。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	物件完成其執行的時間。	DateTime
hadoopJobLog	Amazon EMR 型活動的嘗試時，會提供 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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	物件的狀態。	String
@version	用來建立物件的 AWS Data Pipeline 版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

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	Description	槽類型
資料庫	要執行所提供 SQL 指令碼的資料庫。	參考物件，例如 "database":{"ref": "myDatabaseId"}

物件呼叫欄位	Description	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"}` 。 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。 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。	參考物件，例如 "schedule":{"ref": "myScheduleId"}

必要的群組 (下列其中之一為必要)	Description	槽類型
script	要執行的 SQL 指令碼。您必須指定 script 或 scriptUri。當指令碼存放在 Amazon S3 中時，指令碼不會評估為表達式。當指令碼存放在 Amazon S3 中時，指定 scriptArgument 的多個值很有幫助。	String
scriptUri	URI，指定要在此活動中執行的 SQL 指令碼位置。	String

必要的群組 (下列其中之一為必要)	Description	槽類型
runsOn	執行活動或命令的可運算資源。例如，Amazon EC2 執行個體或 Amazon EMR 叢集。	參考物件，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` 。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
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"}

選用欄位	Description	槽類型
failureAndRerunMode	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。	參考物件，例如 "input":{"ref":"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。	Period
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	若物件尚未進行排程，或是在管道排程啟動之後 'lateAfterTimeout' 所指定的時間期間內仍未完成時，應觸發的動作。	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。此項目只在從指令碼內部進行參考 (例如 `#{output.tablename}`) 及透過在輸出資料節點中設定 'createTableSql' 來建立輸出資料表時才有用。SQL 查詢的輸出不會寫入輸出資料節點。	參考物件，例如 "output":{"ref":"myDataNodeId"}
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

選用欄位	Description	槽類型
pipelineLogUri	上傳管道日誌的 S3 URI (例如 's3://BucketName/Key/')。	String
precondition	選擇是否定義先決條件。在所有先決條件滿足前，資料節點不會標示為"READY"。	參考物件，例如 "precondition":{"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。	String
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

選用欄位	Description	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。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，針對 scriptArgument 指定多個值會很有用。範例：#{format(@scheduledStartTime, "YY-MM-DD HH:MM:SS"}\n#{format(plusPeriod(@scheduledStartTime, "1 day"), "YY-MM-DD HH:MM:SS"}	String

執行時間欄位	Description	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	Description	槽類型
@actualStartTime	此物件執行開始的時間。	DateTime
cancellationReason	若此物件已取消，會提供 cancellationReason。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	執行完成最近一次執行的時間。	DateTime

執行時間欄位	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

Resources

以下是 AWS Data Pipeline 資源物件：

物件

- Ec2Resource

- EmrCluster
- HttpProxy

Ec2Resource

執行管道活動所定義工作的 Amazon EC2 執行個體。

AWS Data Pipeline 現在支援 Amazon EC2 執行個體的 IMDSv2，該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好地處理身分驗證。工作階段會開始和結束一系列請求，在 Amazon EC2 執行個體上執行的軟體會使用這些請求來存取本機存放的 Amazon EC2 執行個體中繼資料和登入資料。軟體使用對 IMDSv2 的簡單 HTTP PUT 請求啟動工作階段。IMDSv2 會傳回秘密權杖給在 Amazon EC2 執行個體上執行的軟體，這會使用權杖做為密碼，向 IMDSv2 提出中繼資料和憑證的請求。

> **Note**
>
> 若要將 IMDSv2 用於 Amazon EC2 執行個體，您需要修改設定，因為預設 AMI 與 IMDSv2 不相容。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`。

如需有關未指定執行個體時所 AWS Data Pipeline 建立的預設 Amazon EC2 執行個體的資訊，請參閱依 AWS 區域的預設 Amazon EC2 執行個體。

範例

EC2-Classic

> **Important**
>
> 只有 2013 年 12 月 4 日之前建立 AWS 的帳戶支援 EC2-Classic 平台。如果您有其中一個帳戶，您可以選擇在 EC2-Classic 網路而非 VPC 中為管道建立 EC2Resource 物件。 EC2-Classic 我們強烈建議您為 VPCs 中的所有管道建立資源。此外，如果您在 EC2-Classic 中有現有資源，建議您將其遷移至 VPC。

下列範例物件會在 EC2-Classic 中啟動 EC2 執行個體，並設定一些選用欄位。

```
{
  "id" : "MyEC2Resource",
  "type" : "Ec2Resource",
  "actionOnTaskFailure" : "terminate",
  "actionOnResourceFailure" : "retryAll",
  "maximumRetries" : "1",
  "instanceType" : "m5.large",
  "securityGroups" : [
    "test-group",
    "default"
  ],
  "keyPair" : "my-key-pair"
}
```

EC2-VPC

以下範例物件會在非預設 VPC 中啟動 EC2 執行個體，並設定一部分的選用欄位。

```
{
  "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	Description	槽類型
resourceRole	控制 Amazon EC2 執行個體可存取之資源的 IAM 角色。	String

必要欄位	Description	槽類型
role	AWS Data Pipeline 用來建立 EC2 執行個體的 IAM 角色。	String

物件呼叫欄位	Description	槽類型
schedule	在排程間隔的執行期間會呼叫此物件。 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： • 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。 • 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": "mySchedule Id"}`

選用欄位	Description	槽類型
actionOnResourceFailure	此資源的資源故障之後所採取的動作。有效值為 `"retryall"` 和 `"retrynone"` 。	String
actionOnTaskFailure	此資源的任務失敗之後所採取的動作。有效值為 `"continue"` 或 `"terminate"` 。	String

選用欄位	Description	槽類型
associatePublicIpAddress	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 Amazon VPC 中，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。	String
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。	Period
availabilityZone	要在其中啟動 Amazon EC2 執行個體的可用區域。	String
disableIMDSv1	預設值為 false，並同時啟用 IMDSv1 和 IMDSv2。如果您將其設定為 true，則會停用 IMDSv1，並且只提供 IMDSv2s	Boolean
failureAndRerunMode	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 服務的代理主機。	參考物件，例如 `"httpProxy": {"ref":"myHttpProxyId"}`
imageId	用於執行個體的 AMI ID。根據預設， AWS Data Pipeline 會使用 HVM AMI 虛擬化類型。使用之特定 AMI ID 是以區域為基礎。您可以指定所選擇的 HVM AMI 來覆蓋預設的 AMI。如需 AMI 類型的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Linux AMI 虛擬化類型和尋找 Linux AMI。	String
initTimeout	等候資源啟動的時間長短。	Period
instanceCount	已廢除。	Integer
instanceType	要啟動的 Amazon EC2 執行個體類型。	String

選用欄位	Description	槽類型
keyPair	金鑰對的名稱。如果您在未指定金鑰對的情況下啟動 Amazon EC2 執行個體，則無法登入。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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 S3 URI（例如 `'s3://BucketName/Key/'`)。	String

選用欄位	Description	槽類型
region	Amazon EC2 執行個體應執行之區域的程式碼。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
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
runAsUser	執行 TaskRunner 的使用者。	String
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。 數值為： • `timeseries` 。 執行個體會排程在每個間隔結束時。 • `cron`。 執行個體會排定在每個間隔的開頭。 • `ondemand`。 可讓您在每次啟用時執行管道一次。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	一或多個 Amazon EC2 安全群組IDs，用於資源集區中的執行個體。	String

選用欄位	Description	槽類型
securityGroups	一或多個 Amazon EC2 安全群組，用於資源集區中的執行個體。	String
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。	String
subnetId	要在其中啟動執行個體的 Amazon EC2 子網路 ID。	String
terminateAfter	在此小時數後終止資源。	Period
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。	String

執行時間欄位	Description	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` 。	String
@cascadeFailedOn	物件失敗所在相依性鏈的描述。	參考物件，例如 `"cascadeF`

執行時間欄位	Description	槽類型
		ailedOn": {"ref":"myRunnableObjectId"}
emrStepLog	步驟日誌僅適用於 Amazon EMR 活動嘗試。	String
errorId	若此物件失敗，會提供錯誤 ID。	String
errorMessage	若此物件失敗，會提供錯誤訊息。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@failureReason	資源故障的原因。	String
@finishedTime	此物件完成其執行的時間。	DateTime
hadoopJobLog	Amazon EMR 活動嘗試時可使用 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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	Description	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。	String
@version	建立物件使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

EmrCluster

代表 Amazon EMR 叢集的組態。EmrActivity 和 使用此物件HadoopActivity來啟動叢集。

目錄

- 語法
- 範例
- 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 — 嘗試在大量期間內平均排程資源。
- CapacityScheduler — 使用佇列來允許叢集管理員將使用者指派給具有不同優先順序和資源配置的佇列。
- 預設 — 由叢集使用，可由您的網站設定。

Amazon EMR 版本

Amazon EMR 版本是一組來自大數據生態系統的開源應用程式。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。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`。根據發行標籤`emr-4.0.0`和更新版本的 Amazon EMR 叢集會使用 `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**
>
> 所有使用發行版本 5.22.0 或更新版本建立的 Amazon EMR 叢集都會使用 Signature 第 4 版來驗證對 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。我們強烈建議您使用支援 Signature 第 4 版的 Amazon EMR 發行版本。使用早期版本時，從 EMR 4.7.x 開始，系列中的最新版本均已更新，可支援簽章版本 4。使用早期版本的 EMR 版本時，我們建議您使用此系列中的最新版本。此外，請避免使用 EMR 4.7.0 以下版本。

考量事項與限制

使用最新版本的 Task Runner

如果您使用具有發行標籤的自我管理EmrCluster物件，請使用最新的任務執行器。如需 Task Runner 的詳細資訊，請參閱使用任務執行器。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。

支援 IMDSv2

舊版，僅 AWS Data Pipeline 支援 IMDSv1。現在， AWS Data Pipeline 支援 Amazon EMR 5.23.1、5.27.1 和 5.32 或更新版本，以及 Amazon EMR 6.2 或更新版本中的 IMDSv2。IMDSv2 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好地處理身分驗證。您應該使用 TaskRunner-2.0 建立使用者受管資源，將執行個體設定為進行 IMDSv2 呼叫。 TaskRunner-2

Amazon EMR 5.32 或更新版本和 Amazon EMR 6.x

Amazon EMR 5.32 或更新版本和 6.x 發行系列使用 Hadoop 3.x 版，與 Hadoop 2.x 版相比，Hadoop 的 classpath 評估方式發生重大變化。Joda-Time 等常見程式庫已從 classpath 中移除。

如果 EmrActivity或 HadoopActivity執行的 Jar 檔案對 Hadoop 3.x 中移除的程式庫具有相依性，則步驟會失敗，並顯示錯誤 `java.lang.NoClassDefFoundError`或 `java.lang.ClassNotFoundException`。使用 Amazon EMR 5.x 發行版本執行沒有問題的 Jar 檔案可能會發生這種情況。

若要修正此問題，您必須在啟動 EmrActivity或 之前，將 Jar 檔案相依性複製到EmrCluster物件上的 Hadoop classpathHadoopActivity。我們提供 bash 指令碼來執行此操作。bash 指令碼可在下列位置使用，其中 *MyRegion* 是EmrCluster物件執行 AWS 的區域，例如 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 EmrActivity或 是否在由 管理的資源HadoopActivity上執行， AWS Data Pipeline 還是在自我管理的資源上執行。

如果您使用 管理的資源 AWS Data Pipeline，請將 bootstrapAction 新增至 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，而且可以將bootstrapAction欄位新增至已有引導動作的EmrCluster物件。

若要將此指令碼指定為引導操作，請使用下列語法，其中 JarFileRegion是儲存 Jar 檔案的區域，而每個 *MyJarFile* n都是要複製到 Hadoop classpath 之 Jar 檔案的 Amazon S3 中的絕對路徑。根據預設，請勿指定 Hadoop classpath 中的 Jar 檔案。

```
s3://datapipeline-MyRegion/MyRegion/bootstrap-actions/latest/TaskRunner/copy-jars-to-
hadoop-classpath.sh,JarFileRegion,MyJarFile1,MyJarFile2[, ...]
```

下列範例會指定引導操作，在 Amazon S3： my-jar-file.jar和 中複製兩個 Jar 檔案emr-dynamodb-tool-4.14.0-jar-with-dependencies.jar。範例中使用的 區域是 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"]
}
```

您必須儲存並啟用管道，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，並使用 SSH 從命令列執行指令碼。指令碼會建立名為 datapipeline-jars.sh的目錄，/etc/hadoop/conf/shellprofile.d以及該目錄中名為 的檔案。以命令列引數提供的 jar 檔案會複製到指令碼建立名為的目錄/home/hadoop/datapipeline_jars。如果您的叢集設定不同，請在下載後適當修改指令碼。

在命令列上執行指令碼的語法與使用上一個範例中bootstrapAction顯示的 略有不同。在引數之間使用空格而非逗號，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

當您建立自訂 IAM 角色時，請仔細考慮叢集執行其工作所需的最低許可。請務必授予必要資源的存取權，例如 Amazon S3 中的檔案或 Amazon 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`角色的聯集做為`EmrCluster`物件角色，或為此目的建立自己的角色。

語法

物件呼叫欄位	Description	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	Description	槽類型
actionOnResourceFailure	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` "。	String
actionOnTaskFailure	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)"。	String

選用欄位	Description	槽類型
additionalMasterSecurityGroupIds	EMR 叢集額外主安全群組的識別符，遵循 sg-01XXXX6a 格式。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。	String
additionalSlaveSecurityGroupIds	EMR 叢集額外從屬安全群組的識別符，遵循 `sg-01XXXX6a` 格式。	String
amiVersion	Amazon EMR 用來安裝叢集節點的 Amazon Machine Image (AMI) 版本。如需詳細資訊，請參閱 Amazon EMR 管理指南。	String
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 EMR 4.0 版及更新版本。	String
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
availabilityZone	叢集執行所在的可用區域。	String
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。	String
組態	Amazon EMR 叢集的組態。此參數僅適用於 Amazon EMR 4.0 版及更新版本。	參考物件，例如 `"configuration":{"ref":"myEmrConfigurationId"}`

選用欄位	Description	槽類型
coreInstanceBidPrice	您願意為 Amazon EC2 執行個體支付的最高 Spot 價格。如果已指定出價，Amazon EMR 會使用執行個體群組適用的 Spot 執行個體。指定貨幣為 USD。	String
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 Amazon EMR 叢集支援的 Amazon EC2 執行個體。	String
coreGroupConfiguration	Amazon EMR 叢集核心執行個體群組的組態。此參數僅適用於 Amazon EMR 4.0 版及更新版本。	參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”}
coreEbsConfiguration	將連接至 Amazon EMR 叢集中核心群組中每個核心節點的 Amazon EBS 磁碟區的組態。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的支援 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 5.7.0 版及更新版本。指定 Amazon EMR 佈建 Amazon EC2 執行個體時要使用之自訂 AMI 的 AMI ID。它也可以用來取代引導操作來自訂叢集節點組態。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的下列主題。使用自訂 AMI	String

選用欄位	Description	槽類型
EbsBlockDeviceConfig	與執行個體群組相關聯的請求 Amazon EBS 區塊型設備組態。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： • `volumesPerInstance` 是具備特定磁碟區組態的 EBS 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。 • `volumeSpecification` 是 Amazon EBS 磁碟區規格，例如 Amazon EMR 叢集中連接至 EC2 執行個體的 EBS 磁碟區請求的磁碟區類型、IOPS 和大小，以 Gigibyte (GiB) 為單位。	參考物件，例如 `“EbsBlockDeviceConfig”: {“ref”: “myEbsBlockDeviceConfig”}`
emrManagedMasterSecurityGroupId	Amazon EMR 叢集主安全群組的識別符，其格式為 `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。	String
emrManagedSlaveSecurityGroupId	Amazon EMR 叢集從屬安全群組的識別符，其格式為 `sg-01XXXX6a`。	String
enableDebugging	在 Amazon EMR 叢集上啟用偵錯。	String
failureAndRerunMode	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

選用欄位	Description	槽類型
httpProxy	用戶端用來連線到 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。	Period
keyPair	用來登入 Amazon EMR 叢集主節點的 Amazon EC2 金鑰對。	String
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
masterInstanceBidPrice	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)。指定貨幣為 USD。設定此值可為 Amazon EMR 叢集主節點啟用 Spot 執行個體。如果已指定出價，Amazon EMR 會使用執行個體群組適用的 Spot 執行個體。	String
masterInstanceType	用於主節點的 Amazon EC2 執行個體類型。請參閱 Amazon EMR 叢集支援的 Amazon EC2 執行個體。	String
masterGroupConfiguration	Amazon EMR 叢集主要執行個體群組的組態。此參數僅適用於 Amazon EMR 4.0 版及更新版本。	參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”}
masterEbsConfiguration	將連接至 Amazon EMR 叢集中主群組中每個主節點的 Amazon EBS 磁碟區的組態。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的支援 EBS 最佳化的執行個體類型。	參考物件，例如 “masterEbsConfiguration”: {“ref”: “myEbsConfiguration”}

選用欄位	Description	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"}`
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 S3 URI（例如 's3：//BucketName/Key/')。	String
region	Amazon EMR 叢集應在其中執行的區域程式碼。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 叢集的版本標籤。	String

選用欄位	Description	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。	Period
resourceRole	AWS Data Pipeline 用來建立 Amazon EMR 叢集的 IAM 角色。預設角色為 `DataPipelineDefaultRole` 。	String
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
role	傳遞至 Amazon EMR 以建立 EC2 節點的 IAM 角色。	String
runsOn	此物件不允許此欄位。	參考物件，例如 `"runsOn": {"ref":"myResourceId"}`
securityConfiguration	要套用至叢集的 EMR 安全組態識別符。此參數僅適用於 Amazon EMR 4.8.0 版及更新版本。	String
serviceAccessSecurityGroupId	Amazon EMR 叢集之服務存取安全群組的識別符。	字串. 它遵循 `sg-01XXXX6a` 格式，例如 `sg-1234abcd` 。

選用欄位	Description	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
subnetId	要在其中啟動 Amazon EMR 叢集的子網路識別符。	String
supportedProducts	在 Amazon EMR 叢集上安裝第三方軟體的參數，例如 Hadoop 的第三方分佈。	String
taskInstanceBidPrice	您願意為 EC2 執行個體支付的 Spot 價格上限。介於 0 到 20.00 的小數值 (不含 0 和 20.00)。指定貨幣為 USD。如果已指定出價，Amazon EMR 會使用執行個體群組適用的 Spot 執行個體。	String
taskInstanceCount	要用於 Amazon EMR 叢集的任務節點數量。	Integer
taskInstanceType	用於任務節點的 Amazon EC2 執行個體類型。	String
taskGroupConfiguration	Amazon EMR 叢集任務執行個體群組的組態。此參數僅適用於 Amazon EMR 4.0 版及更新版本。	參考物件，例如 “configuration”: {“ref”: “myEmrConfigurationId”}

選用欄位	Description	槽類型
taskEbsConfiguration	Amazon EMR 叢集中將連接至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》中的支援 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
VolumeSpecification	Amazon EBS 磁碟區規格，例如 Amazon EMR 叢集中連接至 Amazon EC2 執行個體的 Amazon EBS 磁碟區請求的磁碟區類型、IOPS 和大小，以 Gigibyte (GiB) 為單位。節點可以是核心節點、主節點或任務節點。 VolumeSpecification 包括： • iops() 整數。Amazon EBS 磁碟區支援的每秒 I/O 操作數 (IOPS)，例如 1000。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 EBS I/O 特性。 • sizeinGB() 。 整數。Amazon EBS 磁碟區大小，以 GB (GiB) 為單位，例如 500。如需磁碟區類型和硬碟大小的有效組合資訊，請參閱《Amazon EC2 使用者指南》中的 EBS 磁碟區類型。 • volumetType 。 字串。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”}

選用欄位	Description	槽類型
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。	String

執行時間欄位	Description	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。	String
@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。	String
errorId	若此物件失敗，會提供錯誤 ID。	String
errorMessage	若此物件失敗，會提供錯誤訊息。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
@failureReason	資源故障的原因。	String
@finishedTime	此物件完成其執行的時間。	DateTime

執行時間欄位	Description	槽類型
hadoopJobLog	Amazon EMR 活動嘗試時可使用 Hadoop 任務日誌。	String
@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。	String
@healthStatusFromInstanceId	已達終止狀態之最後一個執行個體物件的 ID。	String
@healthStatusUpdatedTime	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。	String
@lastDeactivatedTime	此物件最後停用的時間。	DateTime
@latestCompletedRunTime	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。	String
@version	建立物件使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

範例

以下為此物件類型的範例。

目錄

使用 hadoopVersion 啟動 Amazon EMR 叢集

Example

下列範例使用 AMI 版本 1.0 和 Hadoop 0.20 啟動 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"]
}
```

使用發行標籤 emr-4.x 或更高版本啟動 Amazon EMR 叢集

Example

下列範例使用較新的`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 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`supportedProducts`的欄位，例如，它可讓您安裝自訂 Hadoop 分佈，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

根據預設，建立的 Hadoop 2.x 版EmrCluster活動 AWS Data Pipeline 會啟用伺服器端加密。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"
   }
```

設定 Hadoop KMS ACL 並在 HDFS 中建立加密區域

Example

下列物件會建立 Hadoop KMS 的 ACL，並在 HDFS 中建立加密區域及對應的加密金鑰：

```
{
      "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

根據預設，`DataPipelineDefaultRole`會以 Amazon EMR 服務角色和 Amazon EC2 執行個體描述檔`DataPipelineDefaultResourceRole`的形式 AWS Data Pipeline 傳遞，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描述檔，並改用它們。 AWS Data Pipeline 應該有足夠的許可來使用自訂角色建立叢集，而且您必須新增 AWS Data Pipeline 做為信任的實體。

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"
}
```

在適用於 Java 的 AWS 開發套件中使用 EmrCluster 資源

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 開發套件執行 Spark 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

此範例包含組態，該組態會在 VPC 內的私有子網路中啟動叢集。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的在 VPC 中啟動 Amazon EMR 叢集。 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，請在EmrCluster組態serviceAccessSecurityGroupId中指定SubnetId、emrManagedSlaveSecurityGroupId、emrManagedMasterSecurityGroupId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 磁碟區連接到叢集節點

Example

您可以將 EBS 磁碟區連接到您管道中 EMR 叢集內任何類型的節點。若要將 EBS 磁碟區連接到節點，請在您的 `EmrCluster` 組態中使用 `coreEbsConfiguration`、`masterEbsConfiguration` 和 `TaskEbsConfiguration`。

此 Amazon EMR 叢集範例會將 Amazon EBS 磁碟區用於其主節點、任務和核心節點。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的 Amazon EMR 中的 Amazon EBS 磁碟區。

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。

在管道中，按一下 `EmrCluster` 物件組態，選擇 Master EBS Configuration (主要 EBS 組態)、Core EBS Configuration (核心 EBS 組態) 或 Task EBS Configuration (任務 EBS 組態)，然後輸入與以下範例相似的組態詳細資訊。

```
{
  "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 可讓您設定自己的代理，並讓 Task Runner 透過它存取 AWS Data Pipeline 服務。您不需要使用此資訊設定執行中的 Task Runner。

TaskRunner 中的 HttpProxy 範例

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	Description	槽類型
hostname	用戶端用來連線到 AWS 服務的代理主機。	String
port	用戶端用來連線到 AWS 服務的代理主機連接埠。	String

選用欄位	Description	槽類型
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

選用欄位	Description	槽類型
*password	代理的密碼。	String
s3NoProxy	在連線到 Amazon S3 時停用 HTTP 代理	Boolean
使用者名稱	代理的使用者名稱。	String
windowsDomain	NTLM 代理的 Windows 網域名稱。	String
windowsWorkgroup	NTLM 代理的 Windows 工作群組名稱。	String

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

先決條件

以下是 AWS Data Pipeline 先決條件物件：

物件

- DynamoDBDataExists
- DynamoDBTableExists
- 存在

- S3KeyExists
- S3PrefixNotEmpty
- ShellCommandPrecondition

DynamoDBDataExists

檢查 DynamoDB 資料表中是否存在資料的先決條件。

語法

必要欄位	Description	槽類型
role	指定要用來執行先決條件的角色。	String
tableName	要檢查的 DynamoDB 資料表。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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"}

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"m yBaseObjectId"}
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	Period
reportProgressTime out	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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

執行時間欄位	Description	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":"myRunnable ObjectId"}
@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。	String
@cascadeFailedOn	物件失敗所在相依性鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}

執行時間欄位	Description	槽類型
currentRetryCount	在這個嘗試中，已嘗試過先決條件的次數。	String
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
lastRetryTime	在這個嘗試中，上次嘗試先決條件的時間。	String
節點	即將執行此先決條件的節點	參考物件，例如 "node":{"ref":"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。	String
@version	建立物件使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

DynamoDBTableExists

檢查 DynamoDB 資料表是否存在的先決條件。

語法

必要欄位	Description	槽類型
role	指定要用來執行先決條件的角色。	String
tableName	要檢查的 DynamoDB 資料表。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
maximumRetries	故障時嘗試重試的次數上限	Integer

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"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	Period
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

執行時間欄位	Description	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

執行時間欄位	Description	槽類型
cancellationReason	若此物件已取消，會提供 cancellationReason。	String
@cascadeFailedOn	物件失敗所在相依性鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}
currentRetryCount	在這個嘗試中，已嘗試過先決條件的次數。	String
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
lastRetryTime	在這個嘗試中，上次嘗試先決條件的時間。	String
節點	即將執行此先決條件的節點	參考物件，例如 "node":{"ref":"myR unnableObjectId"}
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。	String
@version	建立物件使用的管道版本。	String

執行時間欄位	Description	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" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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://amzn-s3-demo-bucket/InputData/#{@scheduledStartTime.format('YYYY-
MM-dd-hh:mm')}.csv",
```

```
  "precondition" : { "ref" : "Ready" }
},
{
  "id" : "Ready",
  "type" : "Exists"
}
```

語法

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為。	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

選用欄位	Description	槽類型
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	Period
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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

執行時間欄位	Description	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。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnableObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String

執行時間欄位	Description	槽類型
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
節點	即將執行此先決條件的節點。	參考物件，例如 "node":{"ref":"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。	String
@version	建立物件使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- ShellCommandPrecondition

S3KeyExists

檢查金鑰是否存在於 Amazon S3 資料節點中。

範例

以下為此物件類型的範例。當 s3Key 參數所參考的鍵 (s3://amzn-s3-demo-bucket/mykey) 存在時，便會觸發先決條件。

```
{
"id" : "InputReady",
"type" : "S3KeyExists",
"role" : "test-role",
"s3Key" : "s3://amzn-s3-demo-bucket/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. 在第一個管道完成時，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	Description	槽類型
role	指定要用來執行先決條件的角色。	String
s3Key	Amazon S3 金鑰。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String

選用欄位	Description	槽類型
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為。	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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。	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。	參考物件，例如 "parent":{"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。	Period
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。	Period
retryDelay	兩次連續嘗試之間的逾時持續時間。	Period

執行時間欄位	Description	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。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnableObjectId"}
currentRetryCount	在這個嘗試中，已嘗試過先決條件的次數。	String
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
lastRetryTime	在這個嘗試中，上次嘗試先決條件的時間。	String
節點	即將執行此先決條件的節點	參考物件，例如 "node":{"ref":"myRunnableObjectId"}
reportProgressTime	遠端活動最近報告進度的時間。	DateTime

執行時間欄位	Description	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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

另請參閱

- ShellCommandPrecondition

S3PrefixNotEmpty

檢查具有指定字首（表示為 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	Description	槽類型
role	指定要用來執行先決條件的角色。	String
s3Prefix	用來檢查物件是否存在的 Amazon S3 字首。	String

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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"}

選用欄位	Description	槽類型
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。	參考物件，例如 "parent":{"ref":"m yBaseObjectId"}
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	Period
reportProgressTime out	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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period

執行時間欄位	Description	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":"myRunnable ObjectId"}
@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。	String
@cascadeFailedOn	物件失敗所在的相依鏈的描述。	參考物件，例如 "cascadeFailedOn": {"ref":"myRunnable ObjectId"}

執行時間欄位	Description	槽類型
currentRetryCount	在這個嘗試中，已嘗試過先決條件的次數。	String
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
lastRetryTime	在這個嘗試中，上次嘗試先決條件的時間。	String
節點	即將執行此先決條件的節點。	參考物件，例如 "node":{"ref":"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。	String
@version	建立物件使用的管道版本。	String
@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	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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)	Description	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。	String
scriptUri	要下載並以 shell 命令執行之檔案的 Amazon S3 URI 路徑。只應使用一個 scriptUri 或	String

必要的群組 (下列其中之一為必要)	Description	槽類型
	command 欄位。scriptUri 無法使用參數，請改用 command。	

選用欄位	Description	槽類型
attemptStatus	遠端活動最新回報的狀態。	String
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。	Period
failureAndRerunMode	描述相依性故障或重新執行時的消費者節點行為	列舉
lateAfterTimeout	物件必須完成的管道啟動後經過的時間。只有在排程類型未設定為 時，才會觸發它ondemand。	Period
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。	參考物件，例如 "parent":{"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	Period

選用欄位	Description	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。	Period
retryDelay	兩次重試嘗試之間的逾時持續時間。	Period
scriptArgument	要傳遞給 shell 指令碼的引數	String
stderr	從 命令接收重新導向系統錯誤訊息的 Amazon S3 路徑。如果您使用 `runsOn` 欄位，這必須是 Amazon S3 路徑，因為執行活動之資源的暫時性性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。	String
stdout	從 命令接收重新導向輸出的 Amazon S3 路徑。如果您使用 `runsOn` 欄位，這必須是 Amazon S3 路徑，因為執行活動之資源的暫時性性質。不過，如果您指定 `workerGroup` 欄位，則允許使用本機檔案路徑。	String

執行時間欄位	Description	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。	String
@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	Description	槽類型
		{"ref":"myRunnable ObjectId"}
emrStepLog	只在 EMR 活動嘗試時才可使用的 EMR 步驟日誌	String
errorId	若此物件失敗，會提供 errorId。	String
errorMessage	若此物件失敗，會提供 errorMessage。	String
errorStackTrace	如果此物件失敗，則為錯誤堆疊追蹤。	String
hadoopJobLog	嘗試 EMR 型活動可用的 Hadoop 任務日誌。	String
hostname	選取任務嘗試之用戶端的主機名稱。	String
節點	即將執行此先決條件的節點	參考物件，例如 "node":{"ref":"myR unnableObjectId"}
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。	String
@version	建立物件使用的管道版本。	String
@waitingOn	此物件等待之相依性清單的描述。	參考物件，例如 "waitingOn":{"ref" :"myRunnableObject Id"}

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

另請參閱

- ShellCommandActivity
- 存在

資料庫

以下是 AWS Data Pipeline 資料庫物件：

物件

- JdbcDatabase
- RdsDatabase
- RedshiftDatabase

JdbcDatabase

定義 JDBC 資料庫。

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	Description	槽類型
connectionString	存取資料庫的 JDBC 連線字串。	String
jdbcDriverClass	在建立 JDBC 連線之前要載入的驅動程式類別。	String
*password	要提供的密碼。	String
使用者名稱	連線至資料庫時要提供的使用者名稱。	String

選用欄位	Description	槽類型
databaseName	要連接的邏輯資料庫的名稱	String
jdbcDriverJarUri	用來連線到資料庫之 JDBC 驅動程式 JAR 檔案在 Amazon S3 中的位置。AWS Data Pipeline 必須具有讀取此 JAR 檔案的許可。	String
jdbcProperties	會在此資料庫 JDBC 連線上設為屬性的 A=B 形式對。	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

RdsDatabase

定義 Amazon RDS 資料庫。

> **Note**
>
> RdsDatabase 不支援 Aurora。改為使用the section called “JdbcDatabase”適用於 Aurora 的 。

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。針對 SQL Server 引擎，jdbcDriverJarUri 欄位是必要欄位，並且您可以指定以下驅動程式：https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774。針對 MySQL 和 PostgreSQL 引擎，jdbcDriverJarUri 欄位是選擇性的。

語法

必要欄位	Description	槽類型
*password	要提供的密碼。	String
rdsInstanceId	資料庫執行個體的 `DBInstanceIdentifier` 屬性。	String
使用者名稱	連線至資料庫時要提供的使用者名稱。	String

選用欄位	Description	槽類型
databaseName	要連接的邏輯資料庫的名稱	String
jdbcDriverJarUri	用來連線到資料庫之 JDBC 驅動程式 JAR 檔案在 Amazon S3 中的位置。AWS Data Pipeline 必須具有讀取此 JAR 檔案的許可。針對 MySQL 和 PostgreSQL 引擎，若未指定此欄位，則會使用預設驅動程式，但您可以使用這個欄位來覆寫預設值。若是 Oracle 和 SQL Server 引擎，則此為必要欄位。	String
jdbcProperties	會在此資料庫 JDBC 連線上設為屬性的 A=B 形式對。	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
region	資料庫所在的區域代碼。例如 us-east-1。	String

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

RedshiftDatabase

定義 Amazon Redshift 資料庫。 `RedshiftDatabase`代表管道所使用的資料庫屬性。

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。若要使用 Amazon Redshift 驅動程式，請在 `connectionString` 欄位中從 Amazon Redshift 主控台指定 Amazon Redshift 資料庫連線字串 （開頭為 "jdbc：redshift："）。

語法

必要欄位	Description	槽類型
*password	要提供的密碼。	String
使用者名稱	連線至資料庫時要提供的使用者名稱。	String

必要的群組 (下列其中之一為必要)	Description	槽類型
clusterId	建立 Amazon Redshift 叢集時，使用者提供的識別符。例如，如果 Amazon Redshift 叢集的端點是 mydb.example.us-east-1.redshift.amazonaws.com，則正確的識別符是 mydb。在 Amazon Redshift 主控台中，您可以從叢集識別符或叢集名稱取得此值。	String
connectionString	用於連線至與管道不同的 帳戶所擁有之 Amazon Redshift 執行個體的 JDBC 端點。您不能同時指定 connectionString 和 clusterId 。	String

選用欄位	Description	槽類型
databaseName	要連接的邏輯資料庫的名稱。	String
jdbcProperties	要在此資料庫 JDBC 連線上設為屬性的 A=B 形式對。	String
parent	目前物件的父系，其插槽已被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
region	資料庫所在的區域代碼。例如 us-east-1。	列舉

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

資料格式

以下是 AWS Data Pipeline 資料格式物件：

物件

- CSV 資料格式
- 自訂資料格式
- DynamoDBDataFormat
- DynamoDBExportDataFormat
- 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	Description	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。	String
escapeChar	可指示剖析器忽略下一個字元的字元 (例如 "\")。	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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	Description	槽類型
columnSeparator	指出資料檔案中資料行結尾的字元。	String

選用欄位	Description	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。	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

選用欄位	Description	槽類型
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。	String

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

DynamoDBDataFormat

將結構描述套用至 DynamoDB 資料表，讓 Hive 查詢可存取。 `DynamoDBDataFormat` 會與`HiveActivity`物件和`DynamoDBDataNode`輸入和輸出搭配使用。 `DynamoDBDataFormat`要求您指定 Hive 查詢中的所有資料欄。如需在 Hive 查詢或 Amazon S3 支援中指定特定資料欄的更多彈性，請參閱 DynamoDBExportDataFormat。

> **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	Description	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。	String
parent	目前物件的父系，其槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	用來建立物件的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

DynamoDBExportDataFormat

將結構描述套用至 DynamoDB 資料表，讓 Hive 查詢可存取。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 和 Amazon S3 支援
- 可讓您在 Hive 查詢中透過特定資料行篩選資料
- 即使您有稀疏結構描述，也會從 DynamoDB 匯出所有屬性

> **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	Description	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	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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	Description	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。	String
inputRegEx	用來剖析 S3 輸入檔的規則表達式。inputRegEx 提供一種方法，可從檔案中相對非結構化的資料擷取資料欄。	String

選用欄位	Description	槽類型
outputFormat	inputRegEx 擷取的欄位，但使用 Java 格式化語法則參考為 %1$s %2$s。	String
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

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	Description	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)。	String
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'。	String
escapeChar	可指示剖析器忽略下一個字元的字元 (例如 "\")。	String
parent	目前物件的父系，其插槽已被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
recordSeparator	分隔記錄的字元。預設為 '\n'。	String

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String

系統欄位	Description	槽類型
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

動作

以下是 AWS Data Pipeline 動作物件：

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	Description	槽類型
message	Amazon SNS 通知的內文文字。	String
role	用來建立 Amazon SNS 警示的 IAM 角色。	String
subject	Amazon SNS 通知訊息的主旨行。	String
topicArn	訊息的目的地 Amazon SNS 主題 ARN。	String

選用欄位	Description	槽類型
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
節點	即將執行此動作的節點。	參考物件，例如 "node":{"ref":"myRunnableObjectId"}
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String

系統欄位	Description	槽類型
@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。	String

終止

觸發取消待處理或未完成的活動、資源或資料節點的動作。如果活動、資源或資料節點不是以 `lateAfterTimeout`值開頭，則會 AWS Data Pipeline 嘗試將活動、資源或資料節點置於取消狀態。

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://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg",
    "s3://amzn-s3-demo-bucket/myPath/myOtherStep.jar,anotherArg"
  ]
},
{
  "name" : "TerminateTasks",
```

```
  "id" : "DefaultAction1",
  "type" : "Terminate"
}
```

語法

選用欄位	Description	槽類型
parent	目前物件的父系，其插槽已被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
節點	即將執行此動作的節點。	參考物件，例如 "node":{"ref":"myRunnableObjectId"}
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

Schedule

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過時， 會 AWS Data Pipeline 回填您的管道，並立即開始從指定的開始時間開始排程執行。針對測試/開發，請使用相對較短的間隔。否則， 會 AWS Data Pipeline 嘗試將管道的所有執行排入佇列並排程在該間隔內。如果管道元件`scheduledStartTime`早於 1 天，則 AWS Data Pipeline 嘗試封鎖管道啟用，以防止意外回填。

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	Description	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"，	Period

必要欄位	Description	槽類型
	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。	

必要的群組 (下列其中之一為必要)	Description	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，若要建立隨需管道則可將其移除。	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	Description	槽類型
endDateTime	結束執行排程的日期和時間。此日期和時間必須晚於 startDateTime 或 startAt 值。預設行為是排程執行直到管道關閉為止。	DateTime
occurrences	啟動管道之後的管道執行次數。您不能搭配使用 occurrences 與 endDateTime。	Integer
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@firstActivationTime	建立物件的時間。	DateTime
@pipelineId	此物件所屬管道的 ID	String
@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	String

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

搭配 Activity 使用，來執行 preActivityTaskConfig 和 postActivityTaskConfig 的殼層指令碼。此物件可供 HadoopActivity、HiveActivity、HiveCopyActivity 及 PigActivity 使用。您可以指定 S3 URI 及指令碼的引數清單。

範例

具備引數的 ShellScriptConfig：

```
{
   "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	Description	槽類型
parent	目前物件的父系，其插槽已被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}
scriptArgument	可搭配使用 shell 指令碼的引數清單。	String
scriptUri	Amazon S3 中應下載並執行的指令碼 URI。	String

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

EmrConfiguration

EmrConfiguration 物件是 EMR 4.0.0 版本或更新版本叢集所使用的組態。組態 (做為清單) 是一個傳遞給 RunJobFlow API 呼叫的參數。Amazon EMR 的組態 API 採用分類和屬性。 AWS Data Pipeline 會使用 EmrConfiguration 搭配對應的屬性物件，在管道執行中啟動的 EMR 叢集上設

定EmrCluster應用程式，例如 Hadoop、Hive、Spark 或 Pig。由於您只能為新叢集變更組態，因此您無法為現有的資源提供 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"
      }
   }
]
```

對應管道物件定義會使用 EmrConfiguration 物件，並在 `property` 欄位中使用 Property 物件的清單：

```
{
  "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 EMR 叢集的 Hive 特定屬性：

```
{
    "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	Description	槽類型
分類	組態的分類。	String

選用欄位	Description	槽類型
組態	此組態的子組態。	參考物件，例如 "configuration":{" ref":"myEmrConfigu rationId"}
parent	目前物件的父系，其插槽會被繼承。	參考物件，例如 "parent":{"ref":"m yBaseObjectId"}

選用欄位	Description	槽類型
屬性	組態屬性。	參考物件，例如 "property":{"ref": "myPropertyId"}

執行時間欄位	Description	槽類型
@version	建立物件使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤	String
@pipelineId	此物件所屬管道的 ID	String
@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	String

另請參閱

- EmrCluster
- 屬性
- Amazon EMR 版本指南

屬性

搭配 EmrConfiguration 物件使用的單一鍵/值屬性。

範例

以下管道定義顯示了一個 EmrConfiguration 物件及對應的 Property 物件，來啟動 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	Description	槽類型
金鑰	金鑰	字串
value	value	String

選用欄位	Description	槽類型
parent	目前物件的父系，其插槽已被繼承。	參考物件，例如 "parent":{"ref":"myBaseObjectId"}

執行時間欄位	Description	槽類型
@version	建立物件時使用的管道版本。	String

系統欄位	Description	槽類型
@error	描述格式錯誤物件的錯誤。	String
@pipelineId	此物件所屬管道的 ID。	String
@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。	String

另請參閱

- EmrCluster

- EmrConfiguration
- Amazon EMR 版本指南

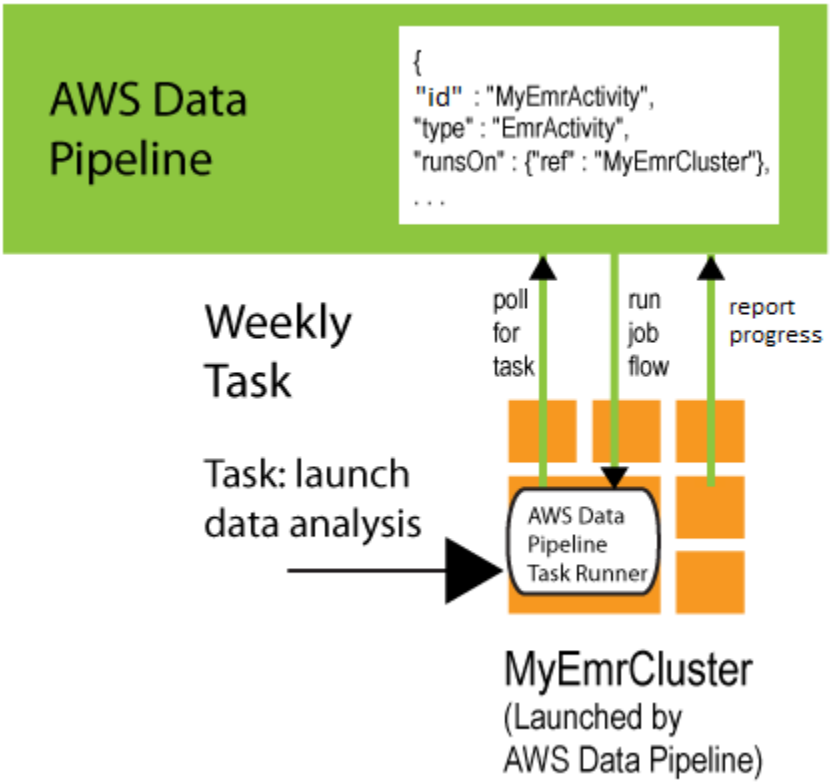
使用任務執行器

任務執行器是一種任務代理程式應用程式，可輪詢 AWS Data Pipeline 排程任務，並在 Amazon EC2 執行個體、Amazon EMR 叢集或其他運算資源上執行它們，並依此報告狀態。根據您的應用程式，您可以選擇：

- 允許 為您 AWS Data Pipeline 安裝和管理一或多個 Task Runner 應用程式。當管道啟動時，活動 runsOn 欄位參考的預設`Ec2Instance`或`EmrCluster`物件會自動建立。 AWS Data Pipeline 會負責在 EC2 執行個體或 EMR 叢集的主節點上安裝 Task Runner。在此模式中， AWS Data Pipeline 可以為您執行大部分的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。您可以安裝任務執行器 （ 可以是任務執行器或您自己裝置自訂任務代理程式），只要它可以與 AWS Data Pipeline Web 服務通訊即可。在此模式中，您幾乎可以完全控制使用的資源及其管理方式，而且您必須手動安裝和設定 Task Runner。若要執行此作業，請使用本節中的程序，如使用任務執行器在現有資源上執行工作中所述。

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`runs0n`的 欄位指定運算資源 (Amazon EC2 執行個體或 Amazon EMR 叢集）。當 AWS Data Pipeline 啟動此資源時，它會在該資源上安裝 Task Runner，並將其設定為處理其`runs0n`欄位設定為該資源的所有活動物件。當 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。當 AWS Data Pipeline 處理該活動時，它會啟動 Amazon EMR 叢集，並將 Task Runner 安裝到主節點。然後，此任務執行器會針對其runsOn欄位設定為該EmrCluster物件的活動處理任務。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://amzn-s3-demo-bucket/myPath/myStep.jar,firstArg,secondArg",
  "step" : "s3://amzn-s3-demo-bucket/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。

如果您在管道中有多個 AWS Data Pipeline受管資源，任務執行器會安裝在每個資源上，而且它們都會輪詢 AWS Data Pipeline 任務以進行處理。

使用任務執行器在現有資源上執行工作

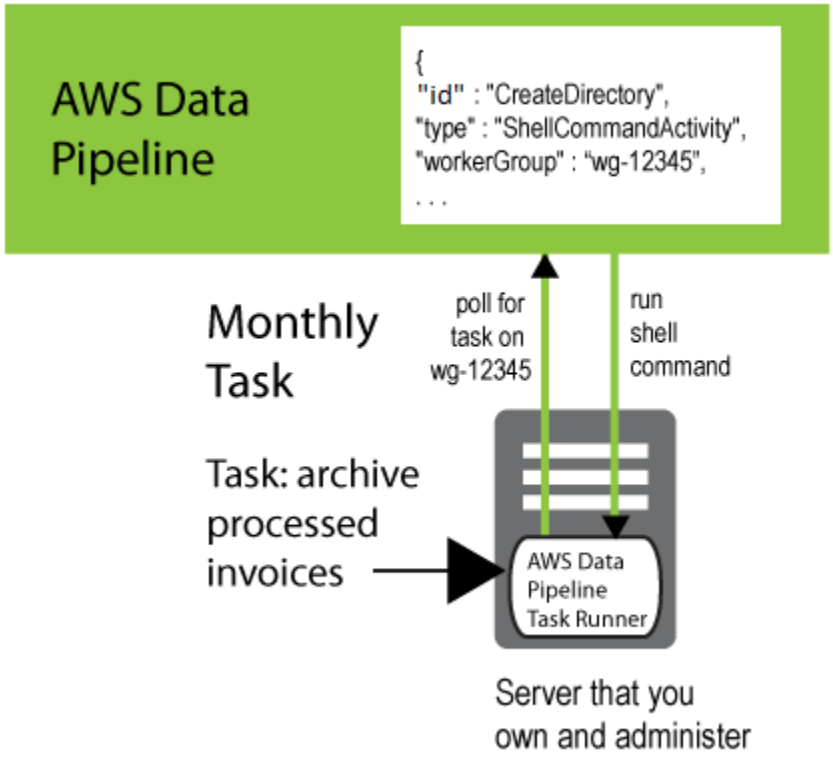
您可以在您管理的運算資源上安裝 Task Runner，例如 Amazon EC2 執行個體，或實體伺服器或工作站。任務執行器可以安裝在任何相容硬體或作業系統的任何位置，但前提是它可以與 AWS Data Pipeline Web 服務通訊。

例如，當您想要使用 AWS Data Pipeline 來處理存放在組織防火牆內的資料時，這種方法非常有用。透過在本機網路的伺服器上安裝 Task Runner，您可以安全地存取本機資料庫，然後輪詢 以執行 AWS Data Pipeline 下一個任務。當 AWS Data Pipeline 結束處理或刪除管道時，Task Runner 執行個體仍會在您的運算資源上執行，直到您手動將其關閉為止。Task Runner 日誌會在管道執行完成後保留。

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運算資源上。

> **Note**
>
> 您只能在 Linux、UNIX 或 macOS 上安裝 Task Runner。Windows 作業系統不支援 Task Runner。
>
> 若要使用 Task Runner 2.0，所需的最低 Java 版本為 1.7。

若要將已安裝的 Task Runner 連線到其應處理的管道活動，請將`workerGroup`欄位新增至物件，並設定 Task Runner 輪詢該工作者群組值。您可以在執行 Task Runner JAR 檔案時，以參數（例如 `--workerGroup=wg-12345`) 形式傳遞工作者群組字串來執行此操作。




```
{
  "id" : "CreateDirectory",
  "type" : "ShellCommandActivity",
  "workerGroup" : "wg-12345",
  "command" : "mkdir new-directory"
}
```

安裝任務執行器

本節說明如何安裝和設定 Task Runner 及其先決條件。安裝過程是一個相當直接的手動程序。

安裝 Task Runner

1. Task Runner 需要 Java 1.6 或 1.8 版。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
   ```

如果您的電腦上未安裝 Java 1.6 或 1.8，請從 https：//http://www.oracle.com/technetwork/java/index.html 下載其中一個版本。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/software/latest/TaskRunner/TaskRunner-1.0.jar 下載，然後將其複製到目標運算資源上的資料夾。對於執行`EmrActivity`任務的 Amazon EMR 叢集，請在叢集的主節點上安裝 Task Runner。
3. 使用 Task Runner 連線至 AWS Data Pipeline Web 服務來處理命令時，使用者需要對具有建立或管理資料管道許可的角色進行程式設計存取。如需詳細資訊，請參閱授予程式設計存取權。
4. Task Runner 會使用 HTTPS 連線至 AWS Data Pipeline Web 服務。如果您使用的是 AWS 資源，請確定已在適當的路由表和子網路 ACL 中啟用 HTTPS。若您使用防火牆或代理，請確認連接埠 443 已開啟。

（選用）授予 Amazon RDS 的任務執行器存取權

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，讓 Task Runner 存取您的 Amazon RDS 執行個體。Task Runner 會從其執行的執行個體取得 Amazon RDS 存取權，因此您新增至 Amazon RDS 執行個體的帳戶和安全群組取決於您安裝 Task Runner 的位置。

授予 EC2-Classic 中 Task Runner 的存取權

1. 開啟 Amazon RDS 主控台。
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 Task Runner 執行的位置，如下所述：

   - `Ec2Resource`
     - Connection Type (連線類型)：`EC2 Security Group`

       Details (詳細資訊)：*`my-security-group-name`* (您為 EC2 執行個體建立的安全群組名稱)
   - `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`：

    Details (詳細資訊)：*`my-ip-address`* (您電腦的 IP 地址或您網路的 IP 地址範圍 (若您的電腦位於防火牆後方的話))

5. 按一下 Add (新增)。

授予 EC2-VPC 中 Task Runner 的存取權

1. 開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇執行個體。
3. 選取資料庫執行個體的詳細資訊圖示。在安全與網路下，開啟安全群組的連結，這會帶您前往 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 Task Runner 執行的位置，如下所述：
   - `Ec2Resource`
     - *`my-security-group-id`* (您為 EC2 執行個體建立的安全群組 ID)
   - `EmrResource`
     - *`master-security-group-id`* (`ElasticMapReduce-master` 安全群組的 ID)
     - *`slave-security-group-id`* (`ElasticMapReduce-slave` 安全群組的 ID)
   - 您的本機環境 (現場部署)
     - *`ip-address`* (您電腦的 IP 地址或您網路的 IP 地址範圍 (若您的電腦位於防火牆後方的話))
5. 按一下 Save (儲存)。

啟動任務執行器

在設定為您安裝 Task Runner 之目錄的新命令提示視窗中，使用以下命令啟動 Task Runner。

```
java -jar TaskRunner-1.0.jar --config ~/credentials.json --workerGroup=myWorkerGroup --
region=MyRegion --logUri=s3://amzn-s3-demo-bucket/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。

`--logUri` 選項用於將壓縮日誌推送到 Amazon S3 中的位置。

當任務執行器處於作用中狀態時，它會列印日誌檔案寫入終端機視窗中的路徑。下列是 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 任務執行器組態選項。

驗證任務執行器記錄

驗證 Task Runner 是否正常運作的最簡單方法是檢查它是否正在寫入日誌檔案。Task Runner 會將每小時日誌檔案寫入安裝 Task Runner 的目錄 `output/logs`下。檔案名稱為 `Task Runner.log.YYYY-MM-DD-HH`，其中 HH 的範圍介於 00 到 23 間 (UDT)。為節省儲存空間，任何超過八個小時的日誌檔案都會以 GZip 壓縮。

任務執行器執行緒和先決條件

任務執行器會針對每個任務、活動和先決條件使用執行緒集區。的預設設定`--tasks`為 2，這表示從任務集區配置了兩個執行緒，每個執行緒都會輪詢新任務 AWS Data Pipeline 的服務。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。

先決條件的管道重試邏輯發生在任務執行器中。配置兩個先決條件執行緒來輪詢 AWS Data Pipeline 先決條件物件。Task Runner 會遵守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。

任務執行器組態選項

這些是當您啟動 Task Runner 時，可從命令列取得的組態選項。

命令列參數	Description
`--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`	Task Runner 在提出請求時使用的 AWS 存取金鑰 ID。 `--accessID` 和 `--secretKey` 選項提供使用 credentials.json 檔案的替代方案。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`	Task Runner 在提出請求時使用的 AWS 私密金鑰。如需詳細資訊，請參閱`--accessID` 。
`--endpoint`	端點是指 web 服務進入點的 URL。在您提出請求的區域中 AWS Data Pipeline 的服務端點。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 AWS Data Pipeline區域和端點AWS 一般參考。
`--workerGroup`	Task Runner 擷取工作之工作者群組的名稱。必要. 當 Task Runner 輪詢 Web 服務時，它會使用您提供的登入資料和 的值`workerGroup` 來選取要擷取哪些（如果有的話）任務。您可以使用對您有意義的任何名稱；唯一的要求是 字串必須與任務執行器及其對應的管道活動相符。工作

命令列參數	Description
	者群組名稱會與區域繫結。即使其他區域中有相同的工作者群組名稱，任務執行器仍會從 中指定的區域取得任務--region。
--taskrunnerId	報告進度時要使用的任務執行器 ID。選用。
--output	日誌輸出檔案的 Task Runner 目錄。選用。日誌檔案會存放在本機目錄中，直到推送到 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，任務執行器會從預設服務區域 擷取任務us-east-1 。 其他支援的區域包含：eu-west-1 、ap-northeast-1 、ap-southeast-2 、us-west-2 。
--logUri	Task Runner 每小時將日誌檔案備份到 的 Amazon S3 目的地路徑。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	Task Runner 用戶端用來連線至 AWS 服務的代理主機。
--proxyPort	Task Runner 用戶端用來連線至 AWS 服務的代理主機連接埠。
--proxyUsername	代理的使用者名稱。
--proxyPassword	代理的密碼。
--proxyDomain	NTLM 代理的 Windows 網域名稱。
--proxyWorkstation	NTLM 代理的 Windows 工作站名稱。

搭配代理使用 Task Runner

若您使用代理主機，您可以在呼叫 Task Runner 時指定其組態，或是設定環境變數 HTTPS_PROXY。搭配 Task Runner 使用的環境變數接受與用於 AWS 命令列界面相同的組態。

任務執行器和自訂 AMIs

當您為管道指定`Ec2Resource`物件時， 會使用為您安裝和設定任務執行器的 AMI，為您 AWS Data Pipeline 建立 EC2 執行個體。此案例中需要與 PV 相容的執行個體類型。或者，您可以使用任務執行器建立自訂 AMI，然後使用 `Ec2Resource` 物件`imageId`的 欄位指定此 AMI 的 ID。如需詳細資訊，請參閱Ec2Resource。

自訂 AMI 必須滿足下列要求， AWS Data Pipeline 才能成功將其用於任務執行器：

- 在執行執行個體相同的區域內建立 AMI。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的建立您自己的 AMI。
- 確認您要使用的執行個體類型支援 AMI 的虛擬化類型。例如，I2 和 G2 執行個體類型需要 HVM AMI，T1、C1、M1 及 M2 執行個體類型則需要 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
- 建立和設定名為 的使用者`ec2-user`。

疑難排解

當您遇到 問題時 AWS Data Pipeline，最常見的症狀是管道未執行。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 主控台是一種方便的工具，可讓您以視覺化方式監控管道的狀態，並輕鬆找到與管道執行失敗或不完整相關的任何錯誤。

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`)。

識別提供管道的 Amazon EMR 叢集

如果 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 主控台來識別為管道提供服務的 Amazon EMR 叢集。這可協助您找到 Amazon EMR 提供的日誌，以取得所發生錯誤的詳細資訊。

若要查看更詳細的 Amazon 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，在名稱中搜尋具有相符 instanceParent 值的叢集，然後選擇偵錯。

   > **Note**
   >
   > 若要讓 Debug (除錯) 按鈕運作，您的管道定義必須將 EmrActivity `enableDebugging` 選項設為 `true`，並將 `EmrLogUri` 選項設為有效的路徑。

5. 現在您知道哪些 Amazon EMR 叢集包含導致管道故障的錯誤，請遵循 Amazon EMR 開發人員指南中的疑難排解秘訣。

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條件。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，或在可以執行 AWS Data Pipeline 之前取消。當此元件所依賴的不同元件或資源發生故障時，就會自動發生這種情況。

CASCADE_FAILED

元件或資源由於其中一個相依項的層疊失敗而取消，但元件可能不是故障的原始來源。

DEACTIVATING

正在停用管道。

FAILED

元件或資源發生錯誤並停止運作。當元件或資源失敗時，可能會導致取消和失敗串聯到其他相依元件。

FINISHED

元件已完成其指派的工作。

INACTIVE

管道已停用。

PAUSED

元件已暫停，且目前未執行其工作。

PENDING

管道已準備好首次啟用。

RUNNING

資源正在執行並準備好接收工作。

SCHEDULED

資源已排程執行。

SHUTTING_DOWN

成功完成其工作後，資源會關閉。

SKIPPED

使用晚於目前排程的時間戳記啟動管道之後，元件略過了執行間隔。

`TIMEDOUT`

資源超過`terminateAfter`閾值且已由 停止 AWS Data Pipeline。資源達到此狀態後，會 AWS Data Pipeline 忽略該資源的 `retryDelay`、 `actionOnResourceFailure`和`retryTimeout`值。此狀態僅適用於 資源。

`VALIDATING`

正在驗證管道定義 AWS Data Pipeline。

`WAITING_FOR_RUNNER`

元件正在等待工作者用戶端擷取工作項目。元件和工作者用戶端關係是由該元件定義的 `runsOn`或 `workerGroup` 欄位所控制。

`WAITING_ON_DEPENDENCIES`

元件正在驗證在執行其工作之前，是否符合其預設和使用者設定的先決條件。

尋找錯誤日誌

本節說明如何尋找 AWS Data Pipeline 寫入的各種日誌，您可以使用這些日誌來判斷特定失敗和錯誤的來源。

管道日誌

我們建議您設定管道在持久性位置建立日誌檔案，例如在下列範例中，您在管道的 `Default` 物件上使用 `pipelineLogUri` 欄位，讓所有管道元件預設使用 Amazon S3 日誌位置 （您可以透過在特定管道元件中設定日誌位置來覆寫此項目）。

> **Note**
>
> 依預設，Task Runner 會將其日誌存放在不同的位置，這可能在管道完成且執行 Task Runner 的執行個體終止時無法使用。如需詳細資訊，請參閱驗證任務執行器記錄。

若要在管道 JSON 檔案中使用 AWS Data Pipeline CLI 設定日誌位置，請使用下列文字開始您的管道檔案：

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

```
{
  "id":"Default",
  "pipelineLogUri":"s3://amzn-s3-demo-bucket/error_logs"
},
...
```

設定管道日誌目錄之後，Task Runner 會在您的目錄中建立日誌的副本，其格式和檔案名稱與上一節有關任務執行器日誌的描述相同。

Hadoop 任務和 Amazon EMR 步驟日誌

透過任何 Hadoop 型活動HiveActivity，例如 HadoopActivity、 或 PigActivity ，您可以在執行時間槽中傳回的位置檢視 Hadoop 任務日誌，hadoopJobLog。 EmrActivity 有自己的記錄功能，而這些日誌會使用 Amazon EMR 選擇的位置存放，並由執行時間槽 emrStepLog 傳回。如需詳細資訊，請參閱《Amazon EMR 開發人員指南》中的檢視日誌檔案。

解決常見的問題

本主題提供各種 AWS Data Pipeline 問題症狀，以及解決問題的建議步驟。

目錄

- 增加 AWS Data Pipeline 限制

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。當您使用 CLI AWS Data Pipeline 提交管道或嘗試使用 AWS Data Pipeline 主控台儲存或啟用管道時，請確定您未收到任何錯誤。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，請確認指派給管道元件的 workerGroup 值與您為 Task Runner 設定的 workerGroup 值完全相同。

> **Note**
>
> 如果您提供 runsOn 值，且 workerGroup 存在，則會忽略 workerGroup。

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 CLI AWS Data Pipeline 工具的 AWS Data Pipeline 主控台或電腦不同。您可能已建立沒有可見錯誤的新管道，但 Task Runner 會因為登入資料的差異而輪詢錯誤的位置，或輪詢許可不足的正確位置，以識別和執行管道定義指定的工作。

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。在此情況下，InputData 物件具備先決條件「Ready (就緒)」，指定資料必須存在，InputData 物件才會完成。如果資料不存在，InputData 物件會維持在 WAITING_ON_DEPENDENCIES 狀態，等待路徑欄位指定的資料變成可用。任何相依於 InputData 的物件同樣會維持在 WAITING_ON_DEPENDENCIES 狀態，等待 InputData 物件到達 FINISHED 狀態。

```
{
    "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，如果登入資料欄位中的資訊沒有存取路徑欄位所指定資料的許可，InputData 物件會卡在 WAITING_ON_DEPENDENCIES 狀態，因為該物件無法存取路徑欄位所指定的資料，即使該資料存在也一樣。

與 Amazon S3 通訊的資源也可能沒有相關聯的公有 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，確認您已在排程物件中正確指定日期，而且 startDateTime 和 endDateTime 值為 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，如中所述的 IAM 角色 AWS Data Pipeline。

存取資源的許可不足

您在 IAM 角色上設定的許可會決定 AWS Data Pipeline 是否可以存取您的 EMR 叢集和 EC2 執行個體來執行您的管道。此外，IAM 提供信任關係的概念，可進一步代表您建立資源。例如，當您建立使用 EC2 執行個體執行命令來移動資料的管道時， AWS Data Pipeline 可以為您佈建此 EC2 執行個體。如果您遇到問題，特別是涉及您可以手動存取的資源，但 AWS Data Pipeline 無法驗證您的 IAM 角色、政策和信任關係，如中所述的 IAM 角色 AWS Data Pipeline。

狀態碼：400 錯誤碼：PipelineNotFoundException

此錯誤表示您的 IAM 預設角色可能沒有讓 AWS Data Pipeline 正常運作所需的必要許可。如需詳細資訊，請參閱的 IAM 角色 AWS Data Pipeline。

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 – 包含在請求中的安全權杖無效。

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適用於管道的排程開始日期，無論管道何時提交。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

此錯誤表示 Task Runner 無法存取 Amazon S3 中的檔案。請驗證：

- 您已正確設定登入資料
- 您嘗試存取的 Amazon S3 儲存貯體已存在
- 您獲授權存取 Amazon S3 儲存貯體

拒絕存取 – 無權執行函數 datapipeline：

在任務執行器日誌中，您可能會看到類似以下的錯誤：

- 錯誤狀態碼：403
- AWS 服務：DataPipeline
- AWS 錯誤碼：AccessDenied
- AWS 錯誤訊息：使用者：arn:aws:sts::XXXXXXXXXXXX:federated-user/i-XXXXXXXX 無權執行：datapipeline:PollForTask。

> **Note**
>
> 在此錯誤訊息中，PollForTask 可能會取代為其他 AWS Data Pipeline 許可的名稱。

此錯誤訊息指出您指定的 IAM 角色需要與 互動所需的額外許可 AWS Data Pipeline。請確定您的 IAM 角色政策包含以下幾行，其中 PollForTask 會取代為您要新增的許可名稱（使用 * 授予所有許可）。如需如何建立新的 IAM 角色並將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 Amazon EMR AMIs 可能會為大型 CSV 檔案建立假資料

在 3.9 之前的 Amazon EMR AMIs (3.8 及更新版本） AWS Data Pipeline 上使用自訂 InputFormat 來讀取和寫入 CSV 檔案，以搭配 MapReduce 任務使用。當服務分階段進出 Amazon S3 的資料表時，會使用此選項。在所偵測到的 InputFormat 問題中，從大型 CSV 檔案讀取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 版本中修正。請使用 Amazon EMR AMI 3.9 或 Amazon EMR 4.0.0 版或更新版本。

增加 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， AWS Data Pipeline 會限制您可以配置的資源，以及您可以配置資源的速率。

目錄

帳戶限制

下列限制適用於單一 AWS 帳戶。如果您需要額外的容量，您可以使用 Amazon Web Services Support Center 請求表單來增加容量。

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 AWS Data Pipeline 代理程式，例如主控台、CLI 和任務執行器。

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 的一般速率為每五秒呼叫一次。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。工作者 Task Runner 的預設實作會自動重試發生調節例外狀況而失敗的 API 呼叫。任務執行器有退避，因此後續嘗試呼叫 API 的間隔會越來越長。如果您要編寫工作程式，我們建議您實作類似的重試邏輯。

這些限制會套用至個別 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

API	一般速率限制	高載限制
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 會擴展以容納大量並行任務，您可以將其設定為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 AWS Data Pipeline 將 設定為自動建立 20 節點的 Amazon EMR 叢集來處理資料，而 AWS 您的帳戶的 EC2 執行個體限制設為 20，您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。

如果您需要額外的容量，您可以使用 Amazon Web Services Support Center 請求表單來增加容量。

AWS Data Pipeline 資源

下列資源有助您使用 AWS Data Pipeline。

- AWS Data Pipeline 產品資訊 - 有關 資訊的主要網頁 AWS Data Pipeline。
- AWS Data Pipeline 技術常見問答集 – 涵蓋開發人員針對此產品提出的前 20 個問題。
- 版本備註 – 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 – 以社群為基礎的論壇，供開發人員討論與 Amazon Web Services 相關的技術問題。

- 課程與研討會 – 連結至以角色為基礎的特殊課程，以及自主進度實驗室，以協助強化您的 AWS 技能並取得實際經驗。
- AWS 開發人員中心 – 探索教學課程、下載工具，並了解 AWS 開發人員事件。
- AWS 開發人員工具 – 開發人員工具、SDKs、IDE 工具組和命令列工具的連結，用於開發和管理 AWS 應用程式。
- 入門資源中心 – 了解如何設定您的 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 - 按照逐步教學課程在 AWS上啟動第一個應用程式。
- AWS 白皮書 – 技術 AWS 白皮書的完整清單連結，涵蓋架構、安全和經濟等主題，並由 AWS 解決方案架構師或其他技術專家撰寫。
- AWS 支援 中心 – 建立和管理 AWS 支援 案例的中樞。也包含其他實用資源的連結，例如論壇、技術 FAQs、服務運作狀態和 AWS Trusted Advisor。
- 支援 – 相關資訊的主要網頁 支援，一對一 one-on-one 的快速回應支援管道，可協助您在雲端中建置和執行應用程式。
- 聯絡我們 – 查詢有關 AWS 帳單、帳戶、事件、濫用與其他問題的中心聯絡點。
- AWS 網站條款 – 有關我們的著作權和商標、您的帳戶、授權和網站存取，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	Description	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 可以繼續正常使用服務。進一步了解	2025 年 7 月 25 日
新增使用 CLI AWS 執行特定程序的文件。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 從資料管道範本建立管道。	2023 年 5 月 26 日
新增更多從 遷移 AWS Data Pipeline 到其他替代服務的內容和範例。	更新遷移 AWS Data Pipeline 至 AWS Glue、 AWS Step Functions 或 Amazon MWAA 的主題，其中包含每個替代方案、服務和範例之間的概念映射的詳細資訊。如需詳細資訊，請參閱從 遷移工作負載 AWS Data Pipeline。	2023 年 3 月 31 日
新增 AWS Data Pipeline 支援 IMDSv2 的資訊。	AWS Data Pipeline 支援 Amazon EMR 和 Amazon EC2 資源的 IMDSv2。 Amazon EC2 如需詳細資訊，請參閱中的資料保護 AWS Data Pipeline、EmrCluster及Ec2Resource。	2022 年 12 月 16 日
新增從 遷移 AWS Data Pipeline 到其他替代服務的主題。	現在還有其他 AWS 服務可為客戶提供更好的資料整合體驗。您可以將 的一般使用案例遷移 AWS Data Pipeline 至 AWS Glue、 AWS Step Functions 或 Amazon MWAA。如需詳細資訊，請參閱從 遷移工作負載 AWS Data Pipeline。	2022 年 12 月 16 日
更新支援 Amazon EC2 和 Amazon EMR 執行個體的清單。	更新支援 Amazon EC2 和 Amazon EMR 執行個體的清單。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 11 月 9 日

變更	Description	版本日期
更新用於執行個體的 HVM (硬體虛擬機器) AMI ID 清單。	更新用於執行個體的 HVM (硬體虛擬機器) AMI ID 清單。如需詳細資訊，請參閱 語法 並搜尋 `imageId`。	
新增將 Amazon EBS 磁碟區連接至叢集節點，以及將 Amazon EMR 叢集啟動至私有子網路的組態。	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。 使用 `coreEbsConfiguration` 、 `masterEbsConfiguration` 和 `TaskEbsConfiguration` 欄位來設定 Amazon EBS 磁碟區與 Amazon EMR 叢集中核心、主節點和任務節點的連接。如需詳細資訊，請參閱將 EBS 磁碟區連接到叢集節點。 使用 `emrManagedMasterSecurityGroupId` 、 `emrManagedSlaveSecurityGroupId` 和 `ServiceAccessSecurityGroupId` 欄位來設定私有子網路中的 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。 如需 `EMRcluster` 語法的詳細資訊，請參閱 EmrCluster。	2018 年 4 月 19 日
新增支援的 Amazon EC2 和 Amazon EMR 執行個體清單。	如果您未在管道定義中指定執行個體類型，則根據預設新增 AWS Data Pipeline 建立的執行個體清單。新增支援 Amazon EC2 和 Amazon EMR 執行個體的清單。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
額外支援 RDS 資料庫	• 新增 `rdsInstanceId` 、 `region` 和 `jdbcDriverJarUri` 至 RdsDatabase。 • 更新了 SqlActivity 中的 `database`，以同時支援 `RdsDatabase` 。	2015 年 8 月 17 日

變更	Description	版本日期
其他 JDBC 支援	• 更新了 SqlActivity 中的 `database`，以同時支援 `JdbcDatabase` 。 • 新增 `jdbcDriverJarUri` 至 JdbcDatabase。 • 新增 `initTimeout` 至 Ec2Resource 和 EmrCluster。 • 已新增 `runAsUser` 到 Ec2Resource。	2015 年 7 月 7 日
HadoopActivity、可用區域和 Spot 支援	• 新增支援提交平行工作到 Hadoop 叢集。如需詳細資訊，請參閱HadoopActivity。 • 新增使用 Ec2Resource 和 EmrCluster 請求 Spot 執行個體的功能。 • 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	已新增範本。更新入門章節，以使用 Getting Started with ShellCommandActivity (ShellCommandActivity 入門) 範本。如需詳細資訊，請參閱使用 CLI 從資料管道範本建立管道。	2014 年 11 月 25 日
VPC 支援	新增支援在虛擬私有雲端 (VPC) 中啟動資源。	2014 年 3 月 12 日
區域支援	新增對多個服務區域的支援。除了 之外`us-east-1` ，AWS Data Pipeline 還支援 `eu-west-1` 、`ap-southeast-2` 、 `ap-northeast-1` 和 `us-west-2` 。	2014 年 2 月 20 日

變更	Description	版本日期
Amazon Redshift 支援	在 中新增對 Amazon Redshift 的支援 AWS Data Pipeline，包括新的主控台範本（複製至 Redshift) 和示範範本的教學課程。如需詳細資訊，請參閱使用 將資料複製到 Amazon Redshift AWS Data Pipeline、RedshiftDataNode、RedshiftDatabase 和 RedshiftCopyActivity。	2013 年 11 月 6 日
PigActivity	新增 PigActivity，可原生支援 Pig。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	新增 CrossRegion DynamoDB Copy (CrossRegion DynamoDB 複製) 主控台範本，包括新的 HiveCopyActivity 和 DynamoDBExportDataFormat。	2013 年 8 月 21 日
串聯失敗和重新執行	新增 AWS Data Pipeline 有關串聯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	新增 AWS Data Pipeline 基本故障診斷影片。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_ON_DEPENDENCIES 狀態	CHECKING_PRECONDITIONS 狀態變更為 WAITING_ON_DEPENDENCIES，並新增管道物件的 @waitingOn 執行時間欄位。	2013 年 5 月 20 日
DynamoDBDataFormat	新增 DynamoDBDataFormat 範本。	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「使用 AWS Data Pipeline、Amazon EMR 和 Hive 處理 Web Logs」和 Amazon EC2 Spot 執行個體支援的影片。	2013 年 2 月 21 日

變更	Description	版本日期
	AWS Data Pipeline 開發人員指南的初始版本。	2012 年 12 月 20 日