aws

使用者指南

AWS IoT Analytics

AWS IoT Analytics: 使用者指南

Table of Contents

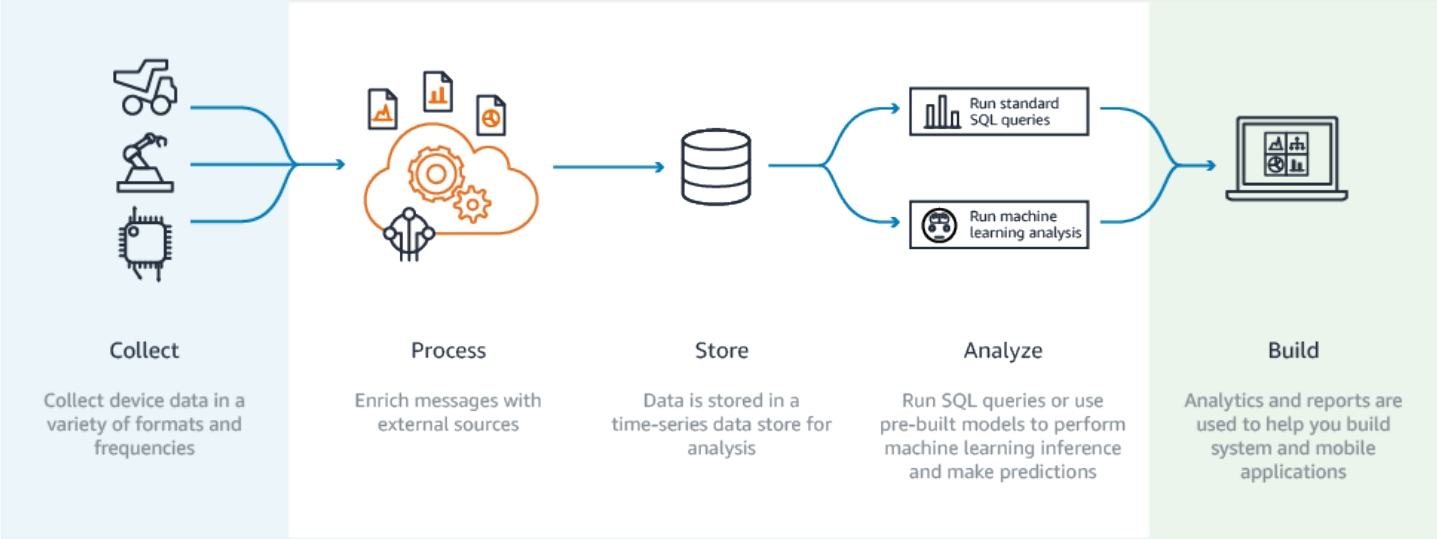
什麼是 AWS IoT Analytics？

AWS IoT Analytics自動化分析 IoT 裝置資料所需的步驟。 AWS IoT Analytics先篩選、轉換並擴展 IoT 資料，再將其存放在時間序列資料存放區中進行分析。您可以設定服務，以只從裝置上收集所需的資料、進行數學轉換來處理資料，然後為資料增加裝置專屬的中繼資料 (例如裝置類型和位置)，再予以存放。然後，您可以使用內建的 SQL 查詢引擎執行查詢來分析資料，或執行更複雜的分析和機器學習推論。 AWS IoT Analytics通過與 Jupyter 筆記本的集成實現高級數據探索。 AWS IoT Analytics還可以透過與 Amazon 整合來實現資料視覺化 QuickSight。Amazon QuickSight 在以下區域提供使用。

傳統的分析和商業智慧工具都是專門用來處理結構化資料。原始 IoT 資料通常來自記錄較低結構化資料的裝置 (例如溫度、動作或聲音)。由於來自這些裝置的資料經常會有明顯的差異、損毀的訊息及錯誤的讀數，因此必須先清理之後才能進行分析。此外，IoT 資料通常只對外部來源的其他資料有意義。AWS IoT Analytics可讓您解決這些問題並收集大量裝置資料、處理訊息並儲存訊息。然後，您可以查詢並對其進行分析。 AWS IoT Analytics包含針對常見 IoT 使用案例的預先建置模型，以便您回答哪些裝置即將發生故障或哪些客戶有可能放棄穿戴式裝置的問題。

如何使用 AWS IoT Analytics

下圖顯示如何使用您可以使用的概觀AWS IoT Analytics。



主要功能

收集

- 與集成AWS IoT Core —AWS IoT Analytics 完全集成，AWS IoT Core因此它可以在流式傳輸時從連接的設備接收消息。

- 使用批次 API 新增來自任何來源的資料，AWS IoT Analytics可透過 HTTP 接收來自任何來源的資料。這意味著連接到互聯網的任何設備或服務都可以將數據發送到AWS IoT Analytics。如需詳細資訊，請參閱《AWS IoT Analytics API 參考》 中的 BatchPutMessage。
- 僅收集您要儲存和分析的資料 — 您可以使用主AWS IoT Analytics控台設定AWS IoT Analytics為透過各種格式和頻率的 MQTT 主題篩選器接收來自裝置的訊息。 AWS IoT Analytics驗證資料是否在您定義的特定參數內並建立通道。然後，服務會將該通道路由至適合的管道進行訊息處理、轉換和增加。

處理

- 清理和篩選 — 可AWS IoT Analytics讓您定義在AWS IoT Analytics偵測到遺失資料時觸發的AWS Lambda函數，以便您可以執行程式碼來估計和填補空白。您也可以定義最大和最小篩選器，以及百分位數閾值，以移除資料中的異常值。
- Transform—AWS IoT Analytics 可以使用您定義的數學或條件邏輯來轉換消息，這樣就可以執行常見的計算，如攝氏轉換為華氏度。
- Enrich—AWS IoT Analytics 可以使用外部資料來源 (例如天氣預報) 來豐富資料，然後將資料路由到資AWS IoT Analytics料存放區。

存放

- 時間序列資料儲AWS IoT Analytics存：將裝置資料儲存在最佳化的時間序列資料存放區中，以加快擷取和分析速度。您也可以管理存取許可、實作資料保留政策，再將您的資料匯出至外部存取點。
- 儲存已處理和原始資料：AWS IoT Analytics儲存已處理的資料，並自動儲存原始擷取的資料，以便您稍後處理。

分析

- 執行臨機操作 SQL query —AWS IoT Analytics 提供 SQL 查詢引擎，讓您可以執行臨機操作查詢並快速取得結果。此服務可讓您使用標準 SQL 查詢從資料存放區擷取資料，以回答問題，例如連線車輛車隊的平均行駛距離，或者在晚上 7 點之後，智慧建築中有多少個門被鎖定。即使連線裝置、機群大小和分析需求變更，這些查詢都仍可重複使用。
- 時間序列分析 —AWS IoT Analytics 支援時間序列分析，因此您可以分析裝置隨時間推移的效能，並瞭解裝置的使用方式和位置，持續監控裝置資料以預測維護問題，並監控感測器以預測環境條件並做出反應。
- 用於複雜分析和機器學習的託管筆記本-AWS IoT Analytics 包括對 Jupyter Notebook 中託管筆記本的支持，以進行統計分析和機器學習。此服務包含一組筆記本範本，其中包含AWS已編寫的機器學習模型和視覺效果。您可以使用這些範本來開始使用與裝置故障分析相關的 IoT 使用案例、預測事件 (例如可能表示客戶將放棄產品的低使用量)，或依客戶使用量層級 (例如繁重使用者、

週末使用者) 或裝置健康狀況劃分裝置。編寫筆記本之後，您可以根據指定的排程將其容器化並執行。如需詳細資訊，請參閱自動化您的工作流程。

- 預測 — 您可以透過稱為邏輯迴歸的方法來進行統計分類。您也可以使用長短期記憶 (LSTM) 這種強大的類神經網路技術，預測隨時間變化的程序輸出或程序狀態。預先建置的筆記本範本也支援用於裝置區隔的 K-means 集群演算法，這會將您的裝置聚集成相似的裝置群。這些範本通常用於分析裝置運作狀態和裝置狀態，例如巧克力工廠的 HVAC 裝置或風力渦輪機葉片的磨損。同樣地，這些筆記本範本可以包含並按排程執行。

構建和可視化

- Amazon QuickSight 整合 —AWS IoT Analytics 提供連接到 Amazon 的連接器，以 QuickSight 便您可以在 QuickSight 儀表板中視覺化您的資料集。
- 主控台整合 — 您也可以在AWS IoT Analytics「主控台」的內嵌 Jupyter 筆記本中，以視覺化方式呈現結果或臨機操作分析。

AWS IoT Analytics組件和概念

頻道

頻道會從 MQTT 主題收集資料，並會在將資料發佈到管道前，先將未處理的原始訊息封存。您也可以使用 BatchPutMessageAPI 直接將訊息傳送至頻道。未處AWS IoT Analytics理的訊息存放在您或管理的 Amazon Storage Service (Amazon S3) 儲存貯體中。

管道

管道會從通道取用訊息，並可讓您在將訊息存放在資料存放區之前，先處理該訊息。這些處理步驟稱為活動 (Pipeline 活動) 可對訊息執行轉換，例如移除、重新命名或新增訊息屬性、根據屬性值篩選訊息、在訊息上叫用 Lambda 函數以進行進階處理，或執行數學轉換以標準化裝置資料。

資料存放區

管道會將其處理完的訊息存放在資料存放區。資料存放區不是資料庫，但卻是可擴展且可查詢的訊息儲存庫。您可使用多個資料存放區存放不同裝置或位置的訊息，或是存放根據您的管道組態和需求依訊息屬性篩選的訊息。與未處理的通道訊息一樣，資料存放區已處理的訊息會存放在您或AWS IoT Analytics管理的 Amazon S3 儲存貯體中。

資料集

您可以透過建立資料集從資料存放區擷取資料。 AWS IoT Analytics可讓您建立 SQL 資料集或容器資料集。

建立資料集之後，您可以透過使用 Amazon 進行整合，探索並深入瞭解資料 QuickSight。您還可以通過與 Jupyter 筆記本的集成來執行更高級的分析功能。Jupyter Notebook 提供了強大的數據科學工具，可以執行機器學習和一系列統計分析。如需詳細資訊，請參閱筆記本範本。

您可以將資料集內容傳送到 Amazon S3 儲存貯體，以便與現有的資料湖整合，或從內部應用程式和視覺化工具存取。您也可以將資料集內容作為輸入傳送至服務 AWS IoT Events，此服務可讓您監視裝置或程序是否發生故障或作業變更，並在此類事件發生時觸發其他動作。

SQL 資料集

SQL 資料集類似於 SQL 資料庫的具體化畫面。您可以套用 SQL 動作來建立 SQL 資料集。透過指定觸發與重複排程可自動產生 SQL 資料集。

容器資料集

容器資料集可讓您自動執行分析工具並產生結果。如需詳細資訊，請參閱自動化您的工作流程。它結合 SQL 資料集做為輸入，含有分析工具及所需的程式庫檔案的 Docker 容器，輸入和輸出變數，以及選用的排程觸發。輸入和輸出變數會告知可執行的映像要在何處取得資料和存放結果。觸發可以在 SQL 資料集完成內容的建立時或根據時間排程表達式來執行您的分析。容器資料集將會自動執行、產生，然後儲存分析工具的結果。

觸發條件

您可以透過指定觸發來自動建立資料集。觸發程序可以是時間間隔 (例如，每兩個小時建立一次此資料集)，也可以是建立另一個資料集的內容時 (例如，建立此資料集的內容`myOtherDataset`完成時建立)。或者，您也可以使用 CreateDatasetContentAPI 手動產生資料集內容。

Docker 容器

您可以建立自己的 Docker 容器來封裝您的分析工具或使用 SageMaker 提供的選項。如需詳細資訊，請參閱 Docker 容器。您可以建立自己的 Docker 容器來封裝您的分析工具或使用提供的選項SageMaker。您可以在指定的 Amazon ECR 登錄中存放容器，讓它可安裝在您想要的平台上。碼頭集裝箱能夠運行與馬特實驗室，八度，智慧，SPSS，R，Fortran，Python，斯卡拉，Java，C ++ 等準備的自定義分析代碼。如需詳細資訊，請參閱容器化筆記本。

差異時段

差異時段是一系列使用者定義、非重疊和接續的時間間隔。差異視窗可讓您使用建立資料集內容，其中包含建立資料集內容，其中包含自上次分析以來已到達資料存放區的新資料。您可以透過在資料集的一`filters``queryAction`部分`deltaTime`中設定來建立差異視窗。如需詳細資訊，請參閱 `CreateDataset` API。通常，您還希望通過設置時間間隔觸發器（`triggers:schedule:expression`）來自動創建數據集內容。這使您可以過濾在特定時間段

內到達的消息，因此上一時間窗口中的消息中包含的數據不會被計算兩次。如需詳細資訊，請參閱範例 6-使用差異視窗 (CLI) 建立 SQL 資料集。

存取 AWS IoT Analytics

作為其中的一部分AWS IoT，AWS IoT Analytics提供下列介面，讓您的裝置能夠產生資料，並讓您的應用程式與其產生的資料互動：

AWS Command Line Interface (AWS CLI)

AWS IoT Analytics在視窗、OS X 和 Linux 上執行的指令。這些命令可讓您建立並管理事物、憑證、憑證、憑證、規則和政策。若要開始使用，請參閱《使用者指南AWS Command Line Interface》。如需命令的詳細資訊AWS IoT，請參閱AWS Command Line Interface參考中的 iot。

> **Important**
>
> 使用指`aws iotanalytics`令與之互動AWS IoT Analytics。使用此`aws iot`指令與 IoT 系統的其他部分互動。

AWS IoT API

使用 HTTP 或 HTTPS 請求建置您的 IoT 應用程式。這些 API 動作可讓您建立並管理事物、憑證、憑證、憑證、規則和政策。如需詳細資訊，請參閱 AWS IoTAPI 參考中的動作。

AWS SDK

使用語言特定的 API 建立AWS IoT Analytics應用程式。這些 SDK 包裝了 HTTP 和 HTTPS API，並使您能夠使用任何支持的語言編寫程序。如需詳細資訊，請參閱 AWS 開發套件與工具。

AWS IoT 裝置 SDK

建立在裝置上執行的應用程式，以AWS IoT Analytics便在裝置上執行，其中 如需詳細資訊，請參閱 AWS IoT 開發套件。

AWS IoT Analytics 主控台

您可以構建組件以在AWS IoT Analytics控制台中可視化結果。

使用案例

預測維護

AWS IoT Analytics提供範本以建立預測性維護模型，並將其套用至您的裝置。例如，您可以用AWS IoT Analytics來預測連接貨運車輛上的加熱和冷卻系統何時可能故障，以便可以重新佈線車輛以防止貨物損壞。或者，車輛製造商可偵測出哪位客戶的煞車踏板已耗損並予以警示，提示其尋求車輛維護。

主動補貨供給

AWS IoT Analytics可讓您建置可即時監控庫存的 IoT 應用程式。例如，食品和飲料公司可以分析食品販賣機的資料，並在供應品量少時，主動再訂購商品。

程序效率評分

您可以透過建置 IoT 應用程式，持續監控不同程序的效率AWS IoT Analytics，並採取行動來改善程序。例如，採礦公司可提高每趟運送的裝載量，以提升其礦車的效率。公司可以透過一段時間的推移AWS IoT Analytics，識別位置或貨車最有效率的負載，然後即時比較與目標負載的任何偏差，並更妥善地規劃領導準則，以提高效率。

智慧農業

AWS IoT Analytics可以使用AWS IoT登錄資料或公開資料來源使用關聯式中繼資料來豐富 IoT 裝置資料，讓您在時間、位置、溫度、高度和其他環境條件下進行分析因素。使用該分析，您就可以撰寫輸出建議動作的模型，供您的裝置在實地情況下採取。例如，為了確定何時用水，灌溉系統可能會使用降雨數據來豐富濕度傳感器數據，從而實現更有效的用水。

開始使用 AWS IoT Analytics (主控台)

使用此教學課程建立所需的 AWS IoT Analytics 資源 (也稱為元件)，以探索有關 IoT 裝置資料的實用見解。

> **備註**
>
> - 如果您在以下自學課程中輸入大寫字元， AWS IoT Analytics 會自動將其變更為小寫字母。
> - 主 AWS IoT Analytics 控台具有一鍵式入門功能，可建立通道、管道、資料存放區和資料集。您可以在登入 AWS IoT Analytics 主控台時找到此功能。
>   - 本教學課程將引導您逐步完成建立 AWS IoT Analytics 資源的每個步驟。

按照以下說明建立 AWS IoT Analytics 通道、管道、資料存放區和資料集。本教學課程也會示範如何使用 AWS IoT Core 主控台傳送要擷取的訊息。 AWS IoT Analytics

主題

登入 AWS IoT Analytics 主控台

要開始使用，您必須擁有一個 AWS 帳戶。如果您已經有一個 AWS 帳戶，請導航到 https://console.aws.amazon.com/iotanalytics/。

如果您沒有 AWS 帳戶，請依照下列步驟建立帳戶。

建立 AWS 帳號

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

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

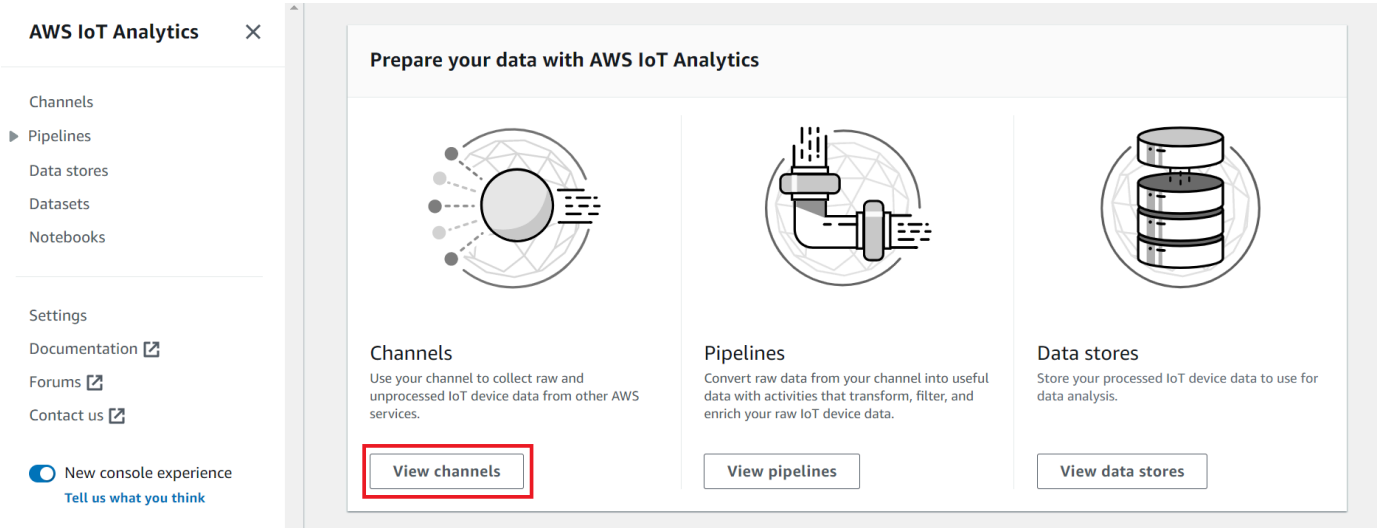
   當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。
3. 登入 AWS Management Console 並瀏覽至 https://console.aws.amazon.com/iotanalytics/。

建立頻道

通道會收集和封存原始、未處理和非結構化 IoT 裝置資料。請依照下列步驟建立頻道。

建立頻道

1. 在 https://console.aws.amazon.com/iotanalytics/ 的「準備資料 AWS IoT Analytics」區段中，選擇「檢視頻道」。



> **Tip**
>
> 您也可以從導覽窗格中選擇「頻道」。

2. 在 Channels (頻道) 頁面上，選擇 Create new queue (建立新頻道)。
3. 在「指定頻道詳細資料」頁面上，輸入頻道的詳細資料。

   a. 輸入唯一且可以輕鬆識別的頻道名稱。

   b. (選擇性) 如果是「標記」，請在頻道中新增一或多個自訂標記 (金鑰/值配對)。標籤可協助您識別為其建立的資源 AWS IoT Analytics。

   c. 選擇下一步。
4. AWS IoT Analytics 將原始、未處理的 IoT 裝置資料存放在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。您可以選擇自己的 Amazon S3 儲存貯體，以存取和管理儲存貯體，也 AWS IoT Analytics 可以為您管理 Amazon S3 儲存貯體。

   a. 在本教學課程中，針對儲存類型，選擇服務受管理的儲存區。

   b. 對於 [選擇儲存原始資料的時間長度]，請選擇 [無限期]。

   c. 選擇下一步。
5. 在 [設定來源] 頁面上，輸入 AWS IoT Analytics 要從中收集郵件資料的資訊 AWS IoT Core。

   a. 輸入 AWS IoT Core 主題篩選器，例如，`update/environment/dht1`。在本教程的後面，您將使用此主題過濾器將消息數據發送到您的頻道。

   b. 在 IAM 角色區域中，選擇 [建立新的]。在「建立新角色」視窗中，輸入角色的名稱，然後選擇「建立角色」。這會自動建立附加適當原則的角色。

   c. 選擇下一步。
6. 檢閱您的選擇，然後選擇 [建立頻道]。
7. 確認您的新頻道出現在「頻道」頁面上。

建立資料存放區

資料倉庫會接收並儲存您的訊息資料。資料存放區不是資料庫。相反地，資料存放區是 Amazon S3 儲存貯體中可擴展且可查詢的儲存庫。您可以將多個資料存放區用於來自不同裝置或位置的訊息。或者，您可以根據管線組態和需求篩選訊息資料。

請依照下列步驟建立資料倉庫。

建立資料倉庫的步驟

1. 在 https://console.aws.amazon.com/iotanalytics/ 的「準備資料 AWS IoT Analytics」區段中，選擇「檢視資料倉庫」。

2. 在 [資料倉庫] 頁面上，選擇 [建立資料倉庫]。
3. 在「指定資料倉庫詳細資訊」頁面上，輸入有關資料倉庫的基本資訊。

   a. 對於「資料倉庫 ID」，請輸入唯一的資料倉庫 ID。建立此 ID 之後，就無法變更它。

   b. (選擇性) 對於標籤，請選擇「新增標籤」以將一個或多個自訂標籤（鍵值對）新增至資料倉庫。標籤可協助您識別為其建立的資源 AWS IoT Analytics。

   c. 選擇下一步。
4. 在 [設定儲存類型] 頁面上，指定儲存資料的方式。

   a. 針對儲存區類型，選擇服務受管理儲存區。

   b. 對於 [設定您要保留已處理資料的時間長度]，選擇 [無限期]。

   c. 選擇下一步。
5. AWS IoT Analytics 資料存放區支援 JSON 和實木地板檔案格式。對於您的資料倉庫資料格式，請選擇 JSON 或實木地板。檔案格式如需 AWS IoT Analytics 支援檔案類型的詳細資訊，請參閱。

   選擇下一步。
6. (選擇性) AWS IoT Analytics 支援資料存放區中的自訂分割區，因此您可以查詢已修剪的資料以改善延遲。如需支援的自訂分割區的詳細資訊，請參閱自訂分割區。

   選擇下一步。
7. 檢閱您的選擇，然後選擇 [建立資料存放區]。
8. 確認您的新資料倉庫顯示在「資料存放區」頁面上。

建立管道

您必須建立管道才能將通道連接到資料存放區。基本管線僅指定收集資料並識別訊息傳送至的資料存放區的通道。如需詳細資訊，請參閱管線活動。

在本自學課程中，您將建立僅將通道連接至資料倉庫的管道。稍後，您可以新增管線活動來處理此資料。

請遵循下列步驟來建立配管。

建立管道

1. 在 https://console.aws.amazon.com/iotanalytics/ 的「準備資料 AWS IoT Analytics」區段中，選擇「檢視管線」。

> **Tip**
>
> 您也可以從導覽窗格中選擇「管線」。

2. 在「配管」頁面上，選擇「建立配管」。
3. 輸入有關管道的詳細資訊。
   a. 在設定管線 ID 和來源中，輸入管線名稱。
   b. 選擇管道的來源，這是您的管 AWS IoT Analytics 道將從中讀取訊息的通道。
   c. 指定管線的輸出，也就是儲存已處理訊息資料的資料存放區。
   d. (選用) 對於標籤，請將一或多個自訂標籤 (鍵值配對) 新增至管線。
   e. 在 [推論訊息屬性] 頁面上，輸入屬性名稱和範例值，從清單中選擇資料類型，然後選擇 [新增屬性]。
   f. 視需要重複上一個步驟，然後選擇 [下一步]。
   g. 您現在不會新增任何管線活動。在 [豐富、轉換和篩選郵件] 頁面上，選擇 [下一步]。
4. 檢閱您的選擇，然後選擇 [建立管道]。
5. 確認您的新管線出現在「管線」頁面上。

> **Note**
>
> 您已建立 AWS IoT Analytics 資源，以便他們可以執行下列作業：
>
> - 透過通道收集原始、未處理的 IoT 裝置訊息資料。
> - 將您的 IoT 裝置訊息資料儲存在資料存放區中。
> - 使用管道清理、篩選、轉換和豐富您的資料。
>
> 接下來，您將建立 AWS IoT Analytics SQL 資料集，以探索有關 IoT 裝置的實用見解。

建立資料集

> **Note**
>
> 資料集通常是資料的集合，可能會或可能不會以表格形式組織。相反地， AWS IoT Analytics 透過將 SQL 查詢套用至資料存放區中的資料來建立資料集。

您現在有一個通道，可將原始訊息資料路由到管線，該管道將資料儲存在可供查詢的資料存放區中。若要查詢資料，請建立資料集。資料集包含用來查詢資料存放區的 SQL 陳述式和運算式，以及選擇性排程，該排程會在您指定的日期和時間重複查詢。您可以使用類似 Amazon CloudWatch 排程運算式的運算式來建立可選排程。

若要建立資料集

1. 在 https://console.aws.amazon.com/iotanalytics/ 的左側導覽窗格中，選擇 [資料集]。
2. 在 [建立資料集] 頁面上，選擇 [建立 SQL]。
3. 在 [指定資料集詳細資料] 頁面上，指定資料集的詳細資料。
   a. 輸入資料集的名稱。
   b. 對於資料倉庫來源，請選擇識別您先前建立之資料倉庫的唯一 ID。
   c. (選擇性) 對於標籤，請在資料集中新增一或多個自訂標籤 (鍵值配對)。
4. 使用 SQL 運算式查詢資料並回答分析問題。查詢結果會儲存在此資料集中。
   a. 在「作者查詢」欄位中，輸入使用萬用字元顯示最多五列資料的 SQL 查詢。

      ```
      SELECT * FROM my_data_store LIMIT 5
      ```

      如需中支援 SQL 功能的詳細資訊 AWS IoT Analytics，請參閱中的 SQL 表達式AWS IoT Analytics。
   b. 您可以選擇「測試查詢」來驗證您的輸入是否正確，並在查詢後面的表格中顯示結果。

      > **Note**
      >
      > - 此時在教學課程中，您的資料存放區可能是空的。在空白資料存放區上執行 SQL 查詢不會傳回結果，因此您可能只會看到__dt。

- 您必須小心將 SQL 查詢限制為合理的大小，以免長時間執行，因為 Athena 會限制執行中查詢的數目上限。因此，您必須小心將 SQL 查詢限制為合理的大小。

  我們建議在測試期間在查詢中使用`LIMIT`子句。測試成功之後，您可以移除這個子句。

5. (選擇性) 使用指定時間範圍中的資料建立資料集內容時，部分資料可能無法及時送達處理。若要允許延遲，您可以指定偏移或差值。如需詳細資訊，請參閱 透過 Amazon CloudWatch 活動取得延遲資料通知。

   此時您不會設定資料選取篩選器。在 [設定資料選取篩選] 頁面上，選擇 [下一步]。

6. (選擇性) 您可以排程此查詢定期執行以重新整理資料集。您可以隨時建立和編輯資料集排程。

   此時您不會排程查詢的週期性執行，因此在 [設定查詢排程] 頁面上選擇 [下一步]。

7. AWS IoT Analytics 將建立此資料集內容的版本，並儲存指定期間內的分析結果。我們建議 90 天，不過您可以選擇設定自訂保留政策。您也可以限制資料集內容的儲存版本數量。

   您可以將預設資料集保留期限設為「無限期」，並保持停用「版本控制」。在 [設定分析] 頁面的結果上，選擇 [下一步]。

8. (選擇性) 您可以設定資料集結果的傳遞規則至特定目的地，例如 AWS IoT Events。

   您不會在本教學課程的其他地方傳送結果，因此請在 [設定資料集內容傳遞規則] 頁面上選擇 [下一步]。

9. 檢閱您的選擇，然後選擇 [建立資料集]。
10. 確認您的新資料集出現在 [資料集] 頁面上。

發送消息數據 AWS IoT

如果您有一個將資料路由到管道的通道，該管道會將資料儲存在可查詢的資料存放區中，那麼您就可以將 IoT 裝置資料傳送到 AWS IoT Analytics。您可以使用下列選項將 AWS IoT Analytics 資料傳送至：

- 使用 AWS IoT 訊息代理程式。
- 使用 AWS IoT Analytics BatchPutMessage API 操作。

在下列步驟中，您會從 AWS IoT Core 主控台中的 AWS IoT 訊息代理程式傳送訊息資料， AWS IoT Analytics 以便擷取此資料。

> **Note**
>
> 當您為郵件建立主題名稱時，請注意下列事項：
>
> - 主題名稱不區分大小寫。名為`example`和`EXAMPLE`在相同有效負載中的欄位被視為重複項目。
> - 主題名稱不能以`$`字元開頭。以開頭的主題`$`是保留的主題，只能由使用 AWS IoT。
> - 請勿在主題名稱中包含可識別個人身分的資訊，因為這些資訊可能會出現在未加密的通訊和報告中。
> - AWS IoT Core 無法在 AWS 帳戶或 AWS 區域之間傳送訊息。

若要傳送訊息資料 AWS IoT

1. 登入 AWS IoT 主控台。
2. 在瀏覽窗格中，選擇 [測試]，然後選擇 [MQTT 測試用戶端]。
3. 在 MQTT 測試用戶端頁面上，選擇「發佈至主題」。
4. 在「主題名稱」中，輸入與您在建立頻道時輸入的主題篩選器相符的名稱。此範例使用 `update/environment/dht1`。
5. 針對「訊息」承載，輸入下列 JSON 內容。

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (選擇性) 針對其他訊息通訊協定選項，選擇新增組態。
7. 選擇 Publish (發佈)。

   這會發布由您的頻道捕獲的消息。然後，您的管道將郵件路由傳送到您的資料存放區。

檢查訊 AWS IoT 息的進度

您可以按照以下步驟檢查訊息是否已擷取到您的頻道中。

若要檢查 AWS IoT 訊息的進度

1. 請登入以下網址：https://console.aws.amazon.com/iotanalytics/
2. 在功能窗格中，選擇 [頻道]，然後選擇您先前建立的頻道名稱。
3. 在頻道的詳細信息頁面上，向下滾動到監控部分，然後調整顯示的時間範圍（1 小時 3h 12h 1d 3d 1 w）。選擇一個值 (例如 1w) 可檢視上週的資料。

您可以使用類似的功能來監視管線活動執行時間和 P ipeline 詳細資訊頁面上的錯誤。在本教學課程中，您尚未將活動指定為管線的一部分，因此您不應該看到任何執行階段錯誤。

若要監視管線活動

1. 在導覽窗格中，選擇「配管」，然後選擇您先前建立的管線名稱。
2. 在管道的詳細資訊頁面上，向下捲動至「監控」區段，然後選擇其中一個時間範圍指示器 (1 小時 3h 12h 1d 3d 1 w) 來調整顯示的時間範圍。

存取查詢結果

資料集內容是包含查詢結果的檔案 (CSV 格式)。

1. 在 https://console.aws.amazon.com/iotanalytics/ 的左側導覽窗格中，選擇 [資料集]。
2. 在 [資料集] 頁面上，選擇您先前建立的資料集名稱。
3. 在資料集資訊頁面的右上角，選擇 [立即執行]。
4. 若要檢查資料集是否已準備就緒，請查看資料集下方是否有類似「您已成功啟動資料集查詢」的訊息。[資料集內容] 索引標籤包含查詢結果並顯示 [成功]。
5. 若要預覽成功查詢的結果，請在 [資料集內容] 索引標籤上選取查詢名稱。若要檢視或儲存包含查詢結果的 CSV 檔案，請選擇 [下載]。

> **Note**
>
> AWS IoT Analytics 可以在資料集內容頁面中嵌入 Jupyter 記事本的 HTML 部分。如需詳細資訊，請參閱 視覺化AWS IoT Analytics數據與控制台。

探索您的資料

您有多種儲存、分析和視覺化資料的選項。

Amazon Simple Storage Service

您可以將資料集內容傳送到 Amazon S3 儲存貯體，以便與現有的資料湖整合，或從內部應用程式和視覺化工具存取。請參閱CreateDataset作業`contentDeliveryRules::destination::s3DestinationConfiguration`中的欄位。

AWS IoT Events

您可以將資料集內容作為輸入傳送至 AWS IoT Events，這項服務可讓您監視裝置或程序是否發生故障或作業變更，並在此類事件發生時起始其他動作。

若要這麼做，請使用CreateDataset作業建立資料集，並在欄位中指定 AWS IoT Events 輸入`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`。您還必須指定角色`roleArn`的，以授予執行 AWS IoT Analytics 權限`iotevents:BatchPutMessage`。無論何時建立資料集內容，都 AWS IoT Analytics 會將每個資料集內容項目當做訊息傳送至指定的 AWS IoT Events 輸入。例如，如果您的資料集包含下列內容。

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

然後 AWS IoT Analytics 發送包含如下字段的消息。

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

您將需要創建一個 AWS IoT Events 輸入來識別您感興趣的字段（一個或多個，`dt`）`what``who`，並創建一個 AWS IoT Events 檢測器模型，該模型使用事件中的這些輸入字段來觸發操作或設置內部變量。

Jupyter 筆記本

Jupyter 筆記本是使用腳本語言來運行臨時數據探索和高級分析的開源解決方案。您可以深入探索並套用更複雜的分析，並在 IoT 裝置資料上使用機器學習方法，例如 k 均值叢集和迴歸模型進行預測。

AWS IoT Analytics 使用 Amazon SageMaker 筆記本實例託管其 Jupyter 筆記本。在創建筆記本實例之前，您必須創建 AWS IoT Analytics 和 Amazon 之間的關係 SageMaker：

1. 瀏覽至SageMaker 主控台並建立筆記本執行個體：
   a. 填入詳細資訊，然後選擇 Create a new role (建立新角色)。請記下角色 ARN。
   b. 建立筆記本執行個體。
2. 前往 IAM 主控台並修改 SageMaker角色：
   a. 開啟角色。它應該有一個受管政策。
   b. 選擇 [新增內嵌原則]，然後針對 [服務] 選擇 [物聯網分析]。選擇 [選取動作]，然後**`GetDatasetContent`**在搜尋方塊中輸入並加以選擇。選擇檢閱政策。
   c. 檢閱原則的正確性，輸入名稱，然後選擇 [建立原則]。

這會提供新建立的角色讀取資料集的權限 AWS IoT Analytics。

1. 返回 https://console.aws.amazon.com/iotanalytics/，然後在左側導覽窗格中選擇 [記事本]。在 [記事本] 頁面上選擇 [建立記事本]。
2. 在 [選取範本] 頁面上，選擇 IOTA 空白範本。
3. 在 [設定記事本] 頁面上，輸入記事本的名稱。在 [選取資料集來源] 中，選擇並選擇您先前建立的資料集。在選取記事本執行個體中，選擇您在其中建立的記事本執行個體 SageMaker。
4. 檢閱您的選擇之後，請選擇「建立記事本」。
5. 在筆記本頁面上，您的筆記本執行個體將在 Amazon 主 SageMaker控台中開啟。

筆記本模板

AWS IoT Analytics 筆記本範本包含 AWS 編寫的機器學習模型和視覺化，可協助您開始 AWS IoT Analytics 使用使用案例。您可以使用這些筆記本範本深入瞭解或重複使用它們，以符合您的 IoT 裝置資料並立即提供價值。

您可以在 AWS IoT Analytics 主控台中找到下列筆記本範本：

- 偵測上下文異常 — 利用 Poisson 指數加權移動平均線 (PEWMA) 模型在測量風速中應用上下文異常偵測。
- 太陽能電池板輸出預測 — 應用分段，季節性和線性時間序列模型，以預測太陽能電池板的輸出。
- 噴氣式引擎的預測性維護 — 應用多變量長短期記憶（LSTM）神經網絡和邏輯回歸以預測噴氣引擎故障。
- 智能家居客戶細分 — 應用 k 均值和主成分分析（PCA）分析，以檢測智能家居使用數據中的不同客戶細分。
- 智慧城市擁堵預測 — 應用 LSTM 預測城市高速公路的使用率。
- 智慧城市空氣品質預測 — 應用 LSTM 預測市中心的微粒污染

AWS IoT Analytics 入門

本節討論您用來收集、儲存、處理和查詢裝置資料的基本指令AWS IoT Analytics。此處顯示的範例使用AWS Command Line Interface (AWS CLI)。若要取得有關的更多資訊AWS CLI，請參閱《AWS Command Line Interface使用指南》。如需有關可用之 CLI 命令的詳細資訊AWS IoT，請參閱AWS Command Line Interface參考資料中的 iot。

> **Important**
>
> 使用指`aws iotanalytics`令可與AWS IoT Analytics使用的互動AWS CLI。使用此`aws iot`命令與 IoT 系統的其他部分互動，使用AWS CLI.

> **Note**
>
> 請注意，當您在接下來的範例中輸入AWS IoT Analytics實體 (通道、資料集、資料存放區和管道) 的名稱時，系統會自動將您使用的任何大寫字母變更為小寫字母。實體的名稱必須以小寫字母開頭，並且只包含小寫字母，下劃線和數字。

建立頻道

頻道會收集並封存未處理的原始訊息資訊，然後再將此資料發佈至管道。傳入的消息會發送到頻道，因此第一步是為您的數據創建一個渠道。

```
aws iotanalytics create-channel --channel-name mychannel
```

如果您想要擷取AWS IoT郵件AWS IoT Analytics，您可以建立 RuAWS IoT les Engine 規則，將訊息傳送到這個通道。這會在稍後顯示將資料導入 AWS IoT Analytics。將資料傳入通道的另一種方法是使用指AWS IoT Analytics令`BatchPutMessage`。

若要列出您已建立的頻道：

```
aws iotanalytics list-channels
```

取得更多頻道的詳細資訊。

```
aws iotanalytics describe-channel --channel-name mychannel
```

未處理的通道訊息存放在管理的 Amazon S3 儲存貯體或存放在您管理的一個儲存貯體中。AWS IoT Analytics使用 channelStorage 參數來指定要在存放在哪一個 Amazon S3 儲存貯體中。預設值為服務受管 Amazon S3 儲存貯體。如果您選擇將通道訊息存放在您管理的 Amazon S3 儲存貯體中，則必須授予AWS IoT Analytics權限，以代表您在 Amazon S3 儲存貯體上執行這些動作：s3:GetBucketLocation(驗證儲存貯體位置)s3:PutObject (存放)、s3:GetObject (讀取)、s3:ListBucket (重新處理)。

Example

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my-iot-analytics-bucket",
                "arn:aws:s3:::my-iot-analytics-bucket/*"
            ]
        }
    ]
}
```

如果您變更客戶管理的通道儲存空間的選項或權限，您可能需要重新處理頻道資料，以確保先前擷取的資料包含在資料集內容中。請參閱重新處理通道資料。

建立資料倉庫

資料存放區會接收和存放您的訊息。它不是數據庫，而是消息的可擴展和可查詢的存儲庫。您可以建立多個資料倉庫來儲存來自不同裝置或位置的訊息，也可以使用單一資料倉庫來接收所有訊AWS IoT息。

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

列示您已建立的資料倉庫。

```
aws iotanalytics list-datastores
```

取得更多資料存放區的詳細資訊。

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

使用 Amazon S3 政策AWS IoT Analytics資源

您可以將處理過的資料存放區訊息存放在 Amazon S3 儲存貯體中AWS IoT Analytics或者在您管理的一個中。建立資料存 Amazon S3，請使用`datastoreStorage`API 參數。預設值為服務受管 Amazon S3 儲存貯體。

如果您選擇將資料存放區訊息存放在您管理的 Amazon S3 儲存貯體中，則必須授予AWS IoT Analytics允許您在 Amazon S3 儲存貯體上執行下列動作：

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

如果您使用資料存放區做為 SQL 查詢資料集的來源，請設定授與的 Amazon S3 儲存貯體政策AWS IoT Analytics對儲存貯體內容叫用 Amazon Athena 查詢的權限。

> **Note**
>
> 建議您指定`aws:SourceArn`在您的儲存貯體政策中，以協助預防混淆代理人安全問題。這會限制存取權限，只允許來自指定帳戶的要求。如需有關混淆代理人問題的詳細資訊，請參閱the section called “預防跨服務混淆代理人”。

下列是授予這些必要許可的儲存貯體政策範例。

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱《》跨帳戶存取在Amazon Athena 使用者指南。

> **Note**
>
> 如果您更新客戶管理資料存放區的選項或權限，您可能需要重新處理通道資料，以確保先前擷取的資料都包含在資料集內容中。如需詳細資訊，請參閱《》重新處理頻道資料。

檔案格式

AWS IoT Analytics資料存放區當前支援 JSON 和 Parquet 檔案格式。預設檔案格式是 JSON。

- JSON (JavaScript 物件標記法)-支持名稱-值對和有序值列表的文本格式。
- Apache Parquet-一種列式存儲格式，用於高效存儲和查詢大量數據。

要配置AWS IoT Analytics數據存儲，則可以使用`FileFormatConfiguration`對象時創建數據存儲。

`fileFormatConfiguration`

包含檔案格式的組態資訊。AWS IoT Analytics資料存放區支援 JSON 和 Parquet。

預設檔案格式是 JSON。您只能指定一種格式。建立資料存放區後，您無法變更檔案格式。

`jsonConfiguration`

包含 JSON 格式的組態資訊。

`parquetConfiguration`

包含 Parquet 格式的組態資訊。

`schemaDefinition`

定義結構描述所需的資訊。

`columns`

指定一個或多個存放資料的欄。

每個結構描述最多可以有 100 列。每列最多可以有 100 個巢狀類型。

`name`

欄位的名稱。

長度限制：1-255 個字符。

`type`

資料的類型。如需支援資料類型的詳細資訊，請參常用資料類型中的AWS Glue開發人員指南。

長度限制：1-131072 個字符。

AWS IoT Analytics支援Amazon Athena 中的資料類型頁面，除了`DECIMAL(`*`precision`*`,` *`scale`*`)`-*`precision`*。

建立資料存放區 (控制台)

下列程序將向您說明如何建立 Parquet 格式保存資料存放區的資料存放區。

創建數據存儲

1. 登入https://console.aws.amazon.com/iotanalytics/。
2. 在導覽窗格中，選擇資料存放區。
3. 在資料存放區頁面上，選擇建立資料存放區。
4. 在指定資料存放區詳情頁面上，輸入有關數據存儲的基本信息。
   a. 適用於資料存放區 ID下，輸入唯一的資料存放區 ID。您無法在建立之後變更此 ID。
   b. （可選）對於標籤，選擇添加新標記將一個或多個自訂標籤 (鍵/值對) 新增至資料存放區。標籤可協助您識別您為AWS IoT Analytics。
   c. 選擇 Next (下一步)。
5. 在設定儲存體方案頁面上，指定如何存儲數據。
   a. 適用於儲存體類型，選擇管理服務儲存體。
   b. 適用於設定您想要保留處理過的資料的時間，選擇無限期。
   c. 選擇 Next (下一步)。
6. 在設定資料格式頁面上，定義數據記錄的結構和格式。
   a. 適用於分類，選擇Parquet。您無法在建立資料存放區後變更此格式。
   b. 適用於推論源，選擇JSON 字符串對於您的資料存放區。
   c. 適用於字串下，請以 JSON 格式輸入架構，如下列範例。

```
{
    "device_id": "0001",
    "temperature": 26,
    "humidity": 29,
    "datetime": "2018-01-26T07:06:01"
}
```

d. 選擇推論架構。

e. 根據設定 Parquet 架構，請確認格式與您的 JSON 示例匹配。如果格式不匹配，請手動更新實木複合架構。

- 如果希望架構顯示更多列，請選擇新增資料行，輸入列名稱，然後選擇資料類型。

> **Note**
>
> 默認情況下，您的架構可以有 100 列。如需詳細資訊，請參閱 AWS IoT Analytics 配額。

- 您可以變更現有列的資料類型。如需支援資料類型的詳細資訊，請參常用資料類型中的AWS Glue開發人員指南。

> **Note**
>
> 在您建立資料存放區後，就無法變更現有列的資料類型。

- 要刪除現有列，請選擇移除資料欄。

f. 選擇 Next (下一步)。

7. (選用)AWS IoT Analytics支持數據存儲中的自定義分區，因此您可以查詢修剪的數據以提高延遲。如需支援的自訂分區的詳細資訊，請參自訂分割區。

   選擇 Next (下一步)。

8. 在Review and create (檢閱和建立)頁面上，檢您的選擇，然後選擇建立資料存放區。

> **Important**
>
> 您無法在建立資料存放區後變更資料存放區 ID、檔案格式或資料類型。

9. 驗證您的新數據存儲是否顯示在資料存放區(憑證已建立！) 頁面上的名稱有些許差異。

自訂分割區

AWS IoT Analytics支持數據分區，以便您可以在數據存儲中組織數據。使用數據分區來組織數據時，您可以查詢已修剪的數據。這樣可以減少每次查詢所掃描的資料量，並提高延遲。

您可以根據消息數據屬性或通過管道活動添加的屬性對數據進行分區。

要開始操作，請在數據存儲中啟用數據分區。指定一個或多個數據分區維，並將分區數據存儲連接到AWS IoT Analytics管道。然後，編寫利用WHERE子句來最佳化效能。

建立資料存放區 (控台)

下列程序將向您演示如何建立自訂分割區的資料存放區。

創建數據存儲

1. 登入 AWS IoT Analytics 主控台。
2. 在導覽窗格中，選擇資料存放區。
3. 在資料存放區頁面上，選擇建立資料存放區。
4. 在指定資料存放區頁面上，輸入有關數據存儲的基本信息。
   a. 適用於資料存放區 ID中，輸入唯一資料存放區。您無法在建立此 ID 後變更此 ID。
   b. （可選）對於標籤，選擇添加新標記將一個或多個自訂標籤 (鍵/值對) 新增至資料存放區。標籤可以幫助您識別為AWS IoT Analytics。
   c. 選擇 Next (下一步)。
5. 在設定儲存類型頁面上，指定如何存儲數據。
   a. 適用於儲存體類型，選擇服務管理儲存。
   b. 適用於設定您希望保留處理過的資料的時間，選擇無限期。
   c. 選擇 Next (下一步)。
6. 在設定資料格式頁面上，定義數據記錄的結構和格式。
   a. 對於資料存放區資料格式分類，選擇JSON或者Parquet。如需有關 的詳細資訊AWS IoT Analytics支持的文件類型，請參閱檔案格式。

   > **Note**
   > 您無法在建立資料存放區後變更此格式。

b. 選擇 Next (下一步)。

7. 為此數據存儲創建自定義分區。

a. 適用於新增資料分割區，選擇啟用。

b. 適用於資料分割區源中，指定有關分區源的基本信息。

選擇樣本來源，然後選擇AWS IoT Analytics通道，為此數據存儲收集消息。

c. 適用於訊息範例屬性中，選擇要用於對數據存儲進行分區的消息屬性。然後，將您的選擇作為屬性分區維或時間戳分區維添加到動作。

> **Note**
>
> 您可以將一個時間戳分割區新增至資料存放區。

d. 適用於自定義數據存儲分區維下，定義分割區維度的基本資訊。您在上一步中選擇的每個消息示例屬性都將成為分區的維度。使用以下選項自定義每個維：

- 分割區類型-指定此分區維度是否為屬性或時間戳記分割區類型。
- 屬性名稱和維度名稱-在預設情況下，AWS IoT Analytics將使用您選擇的消息示例屬性的名稱作為屬性分區維的標識符。編輯屬性名稱以自定義分區維的名稱。您可以在`WHERE`子句來最佳化查詢效能。
  - 任何分區屬性維的名稱都以`__partition_`。
  - 對於時間戳分區類型，AWS IoT Analytics創建以下四個維，其名稱為`__year`、`__month`、`__day`、`__hour`。
- ORDERING-重新排列分區維度以提高查詢的延遲。

適用於時間戳記格式中，通過匹配消息數據中攝入的時間戳來指定時間戳分區的格式。您可以選擇AWS IoT Analytics列出的格式選項，或者指定與數據格式匹配的格式選項。進一步了解日期時間格式化程序。

要添加不是消息屬性的新維度，請選擇新增分割區。

e. 選擇 Next (下一步)。

8. 在Review and create (檢閱和建立)頁面上，檢您的選擇，然後選擇建立資料存放區。

> **Important**
>
> - 您無法在建立資料存放區後變更資料存放區。
> - 要編輯現有分區，必須創建另一個數據存儲並通過管道重新處理數據。

9. 驗證您的新數據存儲是否顯示在資料存放區(憑證已建立！) 頁面上的名稱有些許差異。

建立管道

管道會從通道取用訊息，讓您在將訊息存放於資料存放區之前，先處理和篩選這類訊息。若要將頻道連線到資料存放區，請建立管道。最簡易的管道只會指定要收集資料的通道、及識別訊息要傳送的目標資料存放區。如需更複雜管線的相關資訊，請參閱管線活動。

開始時，我們建議您建立單純負責將頻道連接到資料存放區的管道。然後，在確認原始資料流入資料存放區之後，您可以引進其他管道活動來處理此資料。

執行下列命令來建立管道。

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

該mypipeline.json文件包含以下內容。

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "mychannelactivity",
                "channelName": "mychannel",
                "next": "mystoreactivity"
            }
        },
        {
            "datastore": {
                "name": "mystoreactivity",
                "datastoreName": "mydatastore"
            }
        }
    ]
```

```
}
```

執行下列命令列出現有管道。

```
aws iotanalytics list-pipelines
```

執行下列命令來檢視個別管道的組態。

```
aws iotanalytics describe-pipeline  --pipeline-name mypipeline
```

將資料導入 AWS IoT Analytics

如果您有一個通道將資料路由到管道，該管道會將資料儲存在可查詢的資料存放區中，那麼您就可以將訊息資料傳送至AWS IoT Analytics。在這裡，我們展示了兩種獲取數據的方法AWS IoT Analytics。您可以使用訊AWS IoT息代理程式或使用AWS IoT AnalyticsBatchPutMessage API 來傳送訊息。

主題

使用AWS IoT訊息代理程式

若要使用AWS IoT訊息代理程式，請使用規則引擎建立AWS IoT規則。規則會將具有特定主題的郵件路由傳送至AWS IoT Analytics。不過，首先此規則會要求您建立一個角色，授與所需的許可。

建立 IAM 角色

若要將AWS IoT郵件路由傳送至AWS IoT Analytics頻道，您必須設定規則。但首先，您必須建立 IAM 角色，以授與該規則權限，才能將訊息資料傳送至AWS IoT Analytics通道。

執行下列 命令以建立角色。

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://
arpd.json
```

arpd.json檔案看起來應該如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iot.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

接著，將政策文件連接至角色。

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --
policy-document file://pd.json
```

pd.json檔案看起來應該如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
               "arn:aws:iotanalytics:us-west-2:your-account-number:channel/mychannel"
            ]
        }
    ]
}
```

建立AWS IoT規則

建立將訊息傳送至頻道的AWS IoT規則。

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://
rule.json
```

rule.json檔案看起來應該如下所示。

```
{
    "sql": "SELECT * FROM 'iot/test'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [ {
        "iotAnalytics": {
            "channelName": "mychannel",
            "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
        }
    } ]
}
```

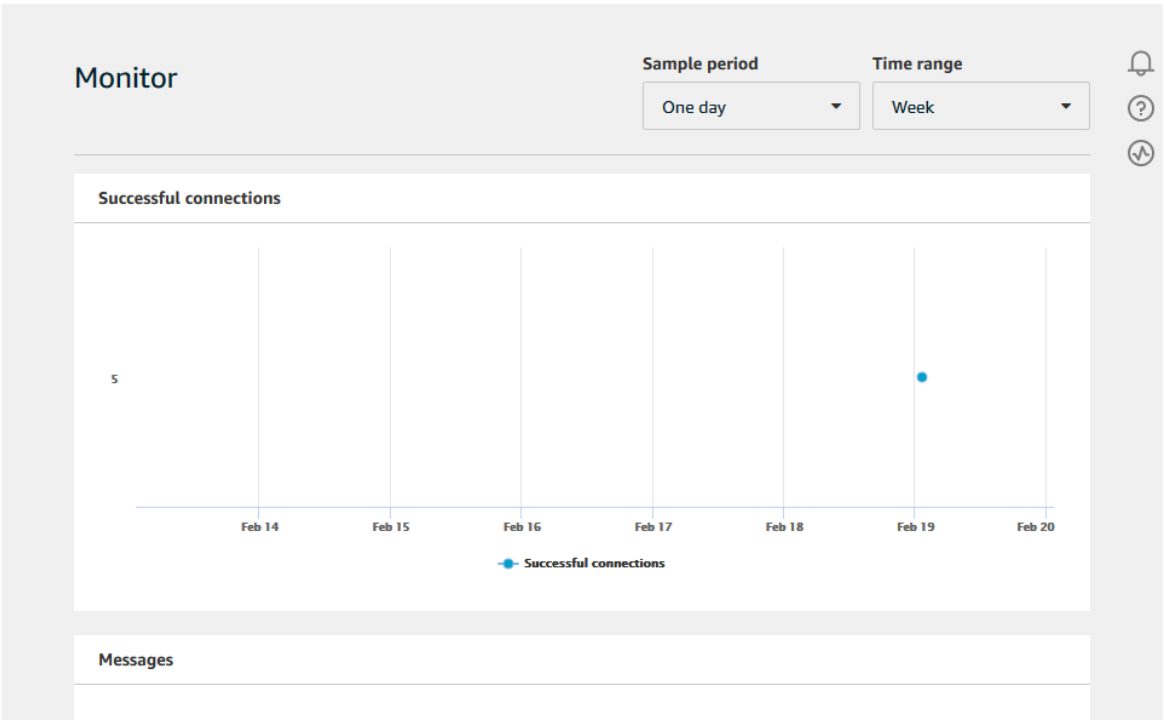
將 `iot/test` 取代為應該路由之訊息的 MQTT 主題。將頻道名稱和角色取代為您在前一節建立的頻道名稱和角色。

傳送 MQTT 訊息至AWS IoT Analytics

將規則連接到通道、管線的通道以及將管線連接到資料存放區之後，符合規則的任何資料現在都會流過AWS IoT Analytics資料存放區，以供查詢。為了測試這一點，您可以使用AWS IoT控制台發送消息。

> **Note**
>
> 您發送到的消息有效載荷（數據）的字段名稱AWS IoT Analytics。
>
> - 僅能包含英數字元和底線 (_)；不得使用其他特殊字元。
> - 開頭必須為字母字元或一個底線 (_)。
> - 不可包含連字號 (-)。
> - 在正則表達式術語中：「`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`」。
> - 不能超過 255 個字元
> - 不區分大小寫。名為foo和FOO在相同有效負載中的欄位被視為重複項目。
>
> 例如，在訊息承載中，`{"temp_01": 29}` 或 `{"_temp_01": 29}` 為有效值，但 `{"temp-01": 29}`、`{"01_temp": 29}` 或 `{"__temp_01": 29}` 皆為無效值。

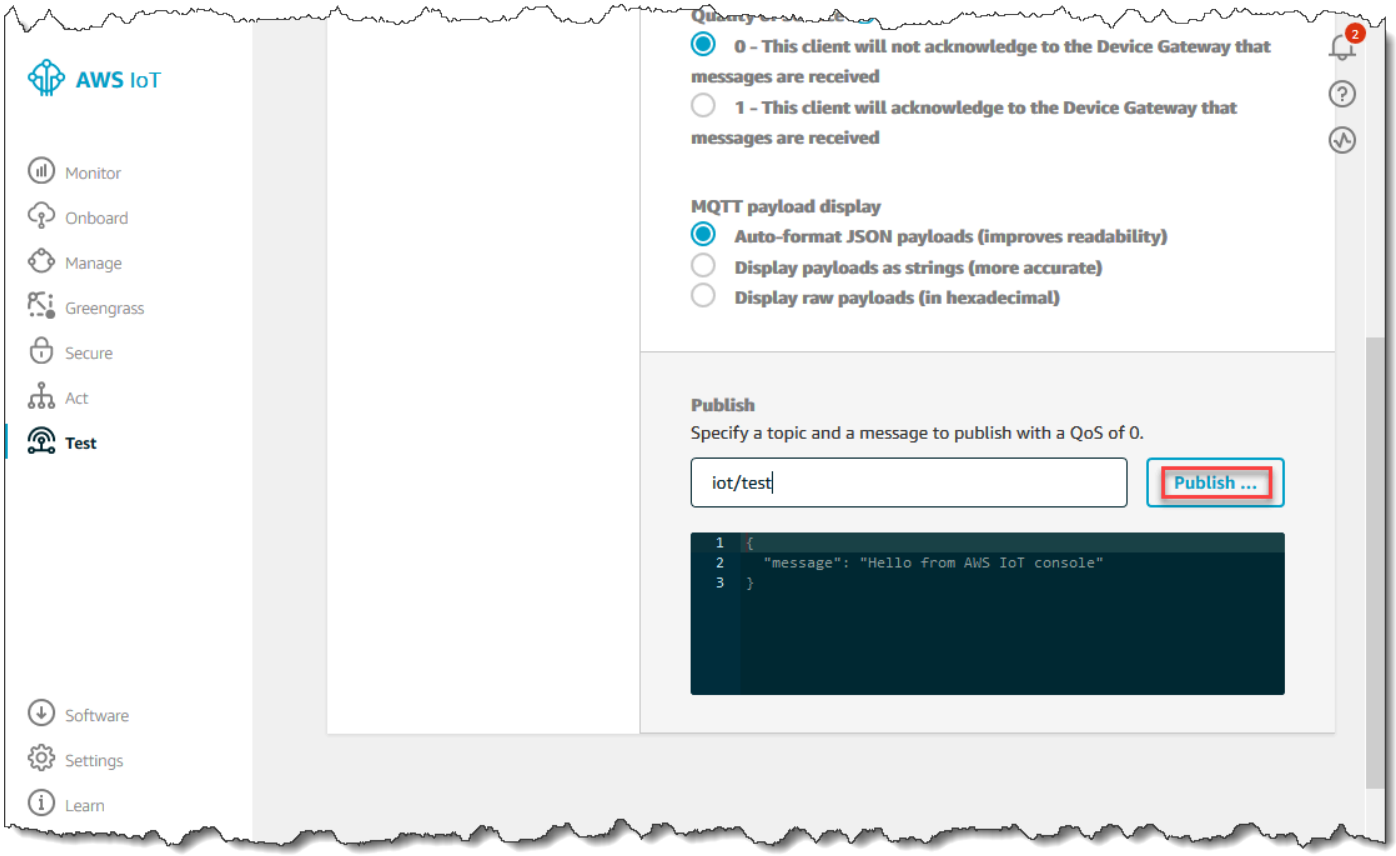
1. 請在 AWS IoT 主控台 左側的導覽窗格中，選擇 Test (測試)。



2. 在 MQTT client (MQTT 用戶端) 頁面，於 Publish (發佈) 部分的 Specify a topic (指定主題) 項目輸入 **`iot/test`**。在訊息承載區段中，確認下列 JSON 內容是否存在，或者如果沒有，請輸入它們。

```
{
    "message": "Hello from the IoT console"
}
```

3. 選擇 Publish to topic (發佈至主題)。



這樣會發佈訊息，並路由傳送到您稍早建立的資料存放區。

應用 BatchPutMessage 程式介面

獲取消息數據的另一種方法AWS IoT Analytics是使用BatchPutMessage API 命令。此方法不需要設定AWS IoT規則，即可將具有特定主題的郵件路由傳送至您的頻道。但它確實要求將其數據/消息發送到頻道的設備能夠運行使用AWS SDK 創建的軟件或能夠使用AWS CLI來調用BatchPutMessage。

1. 創建一個包messages.json含要發送的消息的文件（在此示例中只發送一條消息）。

```
[
    { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI
\" }" }
]
```

2. 執行 batch-put-message 命令。

```
aws iotanalytics batch-put-message  --channel-name mychannel  --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

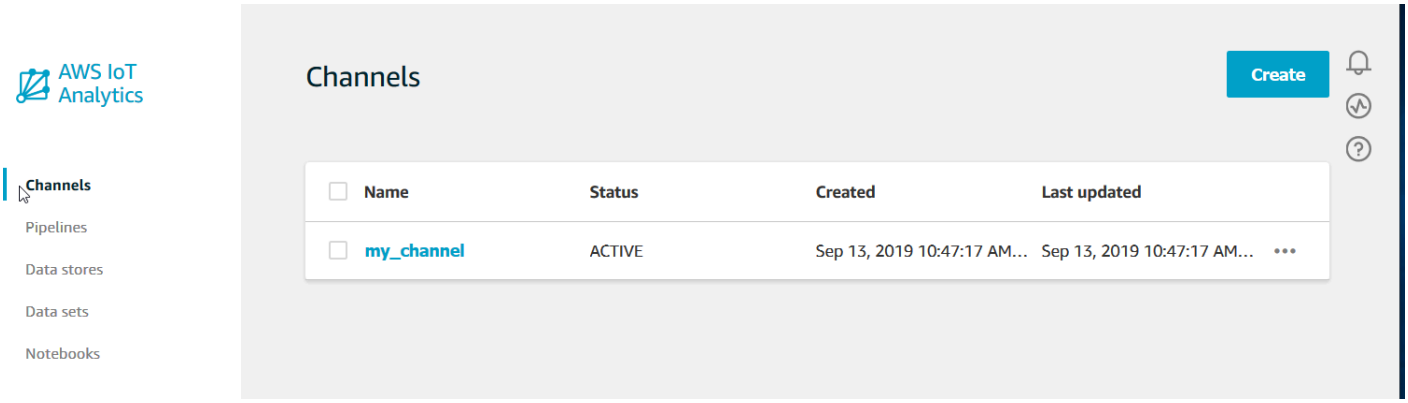
如果沒有錯誤，您會看到下列輸出。

```
{
    "batchPutMessageErrorEntries": []
}
```

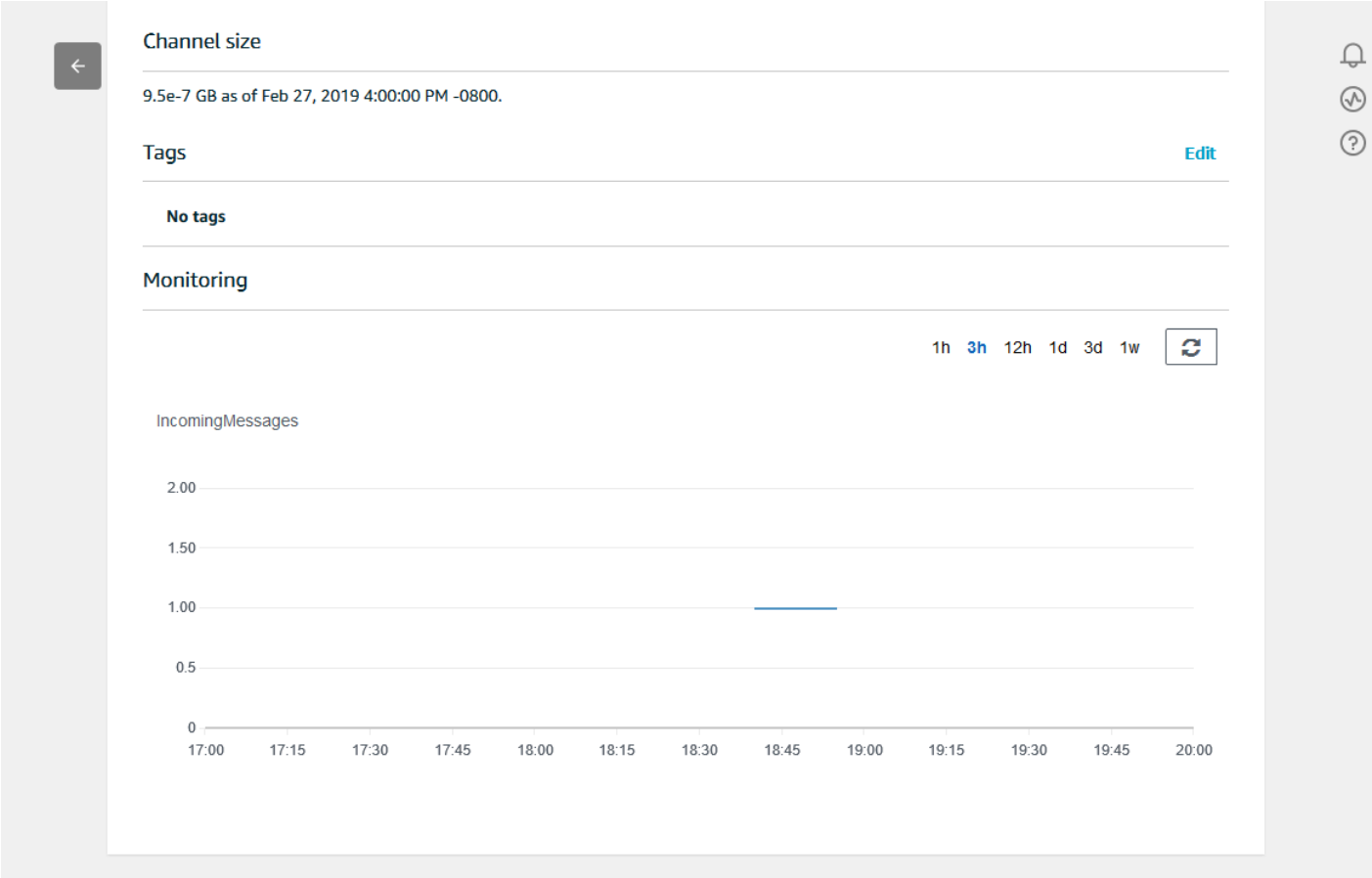
監控擷取的資料

您可以使用AWS IoT Analytics主控台檢查您傳送的訊息是否已擷取到頻道中。

1. 在AWS IoT Analytics主控台的左側導覽窗格中，選擇「準備」，然後選擇「頻道」(C hannel)，然後選擇您先前建立的頻道名稱。

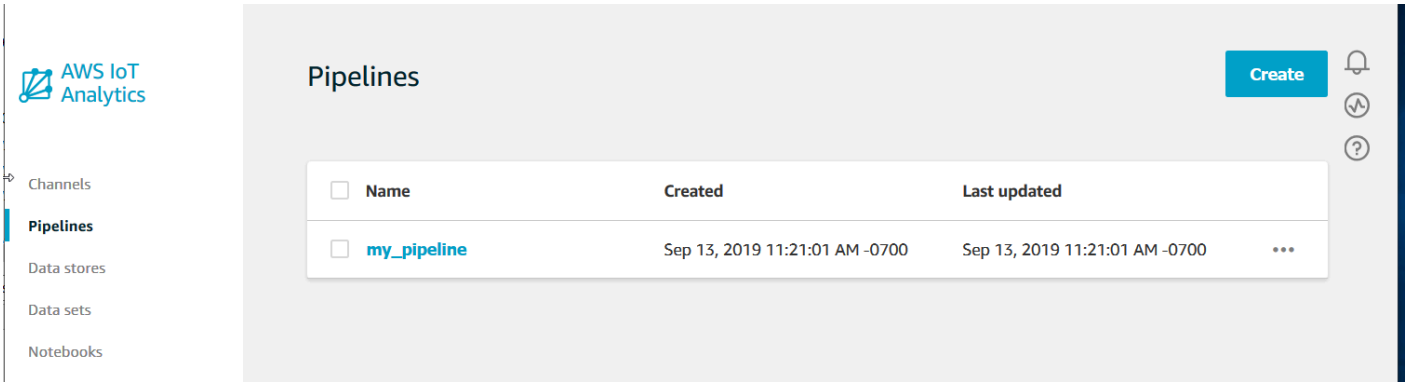


2. 在頻道詳細資訊頁面上，向下捲動到 Monitoring (監控) 區段。視需要調整顯示時間範圍，方法為選擇其中一個時間範圍指標 (1h 3h 12h 1d 3d 1w)。您應該看到一條圖線，指示在指定時間範圍內攝入此通道的消息數量。

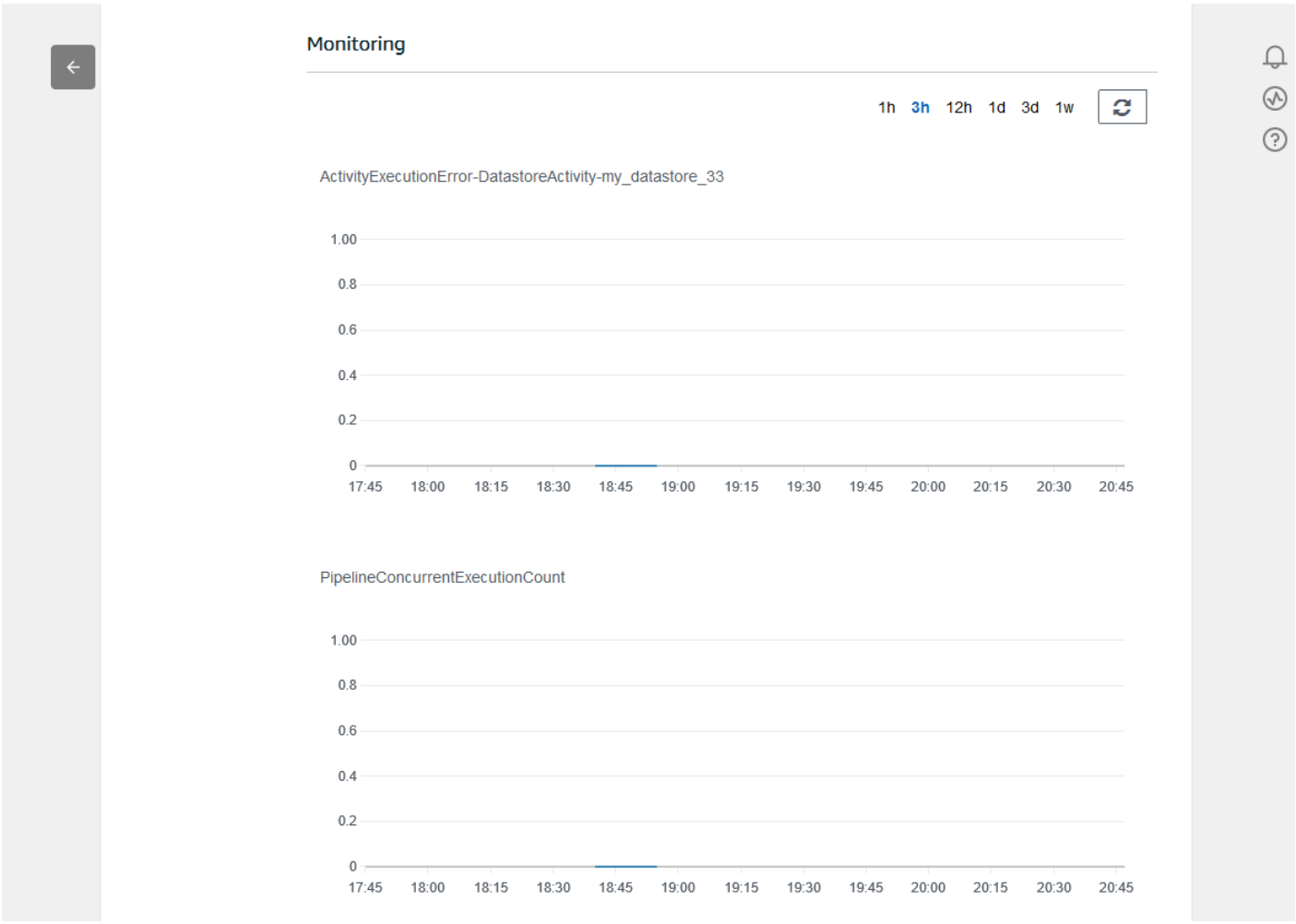


存在類似的監控功能，可用於檢查管道活動執行。您可以在管道詳細資訊頁面上監控活動執行錯誤。如果您尚未將活動指定為管道的一部分，則應顯示 0 個執行錯誤。

1. 在AWS IoT Analytics主控台的左側導覽窗格中，選擇「準備」，然後選擇「管線」，然後選擇您先前建立的管線名稱。



2. 在管道詳細資訊頁面上，向下捲動到 Monitoring (監控) 區段。視需要調整顯示時間範圍，方法為選擇其中一個時間範圍指標 (1h 3h 12h 1d 3d 1w)。您應該會看到一條圖形線，指出指定時間範圍內管線活動執行錯誤的數目。



建立資料集

您可以透過建立 SQL 資料集或容器資料集，從資料存放區擷取資料。 AWS IoT Analytics可以查詢數據以回答分析問題。雖然資料倉庫不是資料庫，但您可以使用 SQL 運算式來查詢資料並產生儲存在資料集中的結果。

主題

- 查詢資料
- 訪問查詢的數據

查詢資料

若要查詢資料，請建立資料集。資料集包含您用來查詢資料存放區的 SQL，以及在您選擇的日期和時間重複查詢的選擇性排程。您可以使用類似於 Amazon CloudWatch 排程運算式的運算式來建立可選排程。

執行下列命令來建立資料料料料料料集。

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

mydataset.json檔案包含下列內容的位置。

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName":"myaction",
            "queryAction": {
                "sqlQuery": "select * from mydatastore"
            }
        }
    ]
}
```

執行下列命令，以執行查詢來建立資料料料料料料料集內容。

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

請稍後幾分鐘建立資料集內容，再繼續進行。

訪問查詢的數據

查詢的結果是以 CSV 格式儲存為檔案的資料集內容。系統會透過 Amazon S3 提供該檔案。以下範例會說明如何檢查結果是否準備就緒，接著下載該檔案。

執行下列 get-dataset-content 命令。

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

如果您的數據集包含任何數據，則來自get-dataset-content的輸出"state": "SUCCEEDED"在status字段中，如下面的示例所示。

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "someEntry",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

dataURI 是輸出結果的已簽署 URL。它的有效期間很短 (幾個小時)。根據您的工作流程，在存取內容前您可能會想要一律呼叫 get-dataset-content，因為呼叫此命令會產生新的已簽署 URL。

瀏覽AWS IoT Analytics資料

您有多種存儲，分析和可視化AWS IoT Analytics數據的選項。

本頁主題：

- Simple Storage Service (Amazon Simple Storage Service (Amazon S3))
- AWS IoT Events
- 亞馬遜 QuickSight
- Jupyter 筆記本

Simple Storage Service (Amazon Simple Storage Service (Amazon S3))

您可以將資料集內容傳送到 Amazon Simple Storage Service (Amazon S3) 儲存貯體，以便與現有的資料湖整合，或從內部應用程式和視覺化工具存取。請參閱中的欄contentDeliveryRules::destination::s3DestinationConfiguration位CreateDataset。

AWS IoT Events

您可以將資料集內容作為輸入傳送至AWS IoT Events，讓您監控設備和裝置或變更的營運是否故障或變更，以及在這類事件發生時觸發其他動作。

若要這麼做，請使用建立資料集，CreateDataset並在欄位中指定AWS IoT Events輸入`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`。您還必須指定授與執行「iotevents：BatchPutMessage」AWS IoT Analytics 權限`roleArn`的角色。無論何時建立資料集的內容，都AWS IoT Analytics會將每個資料集內容項目當做訊息傳送至指定的AWS IoT Events輸入。例如，如果資料集包含：

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

然後AWS IoT Analytics將發送包含這樣字段的消息：

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

並且您將希望創建一個AWS IoT Events輸入來識別您感興趣的字段（一個或多個，`dt`）`what``who`，並創建一個AWS IoT Events檢測器模型，該模型使用事件中的這些輸入字段來觸發操作或設置內部變量。

亞馬遜 QuickSight

AWS IoT Analytics提供與亞馬遜直接集成 QuickSight。Amazon QuickSight 是一項快速的商業分析服務，可用來建置視覺化效果、執行臨機操作分析，及從資料獲取商業洞察。Amazon QuickSight 讓組織能夠擴展到數十萬名使用者，並使用強大的記憶體內引擎 (SPICE) 提供回應迅速的效能。亞馬遜 QuickSight 在這些區域提供。

Jupyter 筆記本

AWS IoT AnalyticsJupyter 筆記本也可以直接使用資料集，以執行進階分析和資料探索。Jupyter 筆記本是一個開源的解決方案。您可以前往 http://jupyter.org/install.html，安裝並下載該筆記本。另外還提供與 SageMaker Amazon 託管筆記型電腦解決方案進行其他整合。

保留多個版本的資料集

您可以在叫用和 UpdateDatasetAPI 時指定資料集`retentionPeriod and versioningConfiguration`欄位的值，來選擇要保留的資料集內容版本數量以CreateDataset及保留的時間長度：

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
...
```

這兩個參數的設定會一起使用，以下列方式決定要保留多少版本的資料集內容，以及保留的時間長度。

	retentionPeriod [未指定]	retentionPeriod： 無限 = 真，numberOfDays = 未設置	retentionPeriod： 無限 = 假，numberOfDays = X
versioningConfiguration： [未指定]	只有最新版本加上最新的成功版本 (如果不同) 保留 90 天。	只有最新版本加上最新的成功版本 (如果不同) 無限期保留。	只有最新版本加上最新的成功版本 (如果不同) 保留 X 天。
versioningConfiguration： unlimited = TRUE，maxVersions 未設定	過去 90 天的所有版本都將保留，無論有多少。	保留的版本數目沒有限制。	最後 X 天的所有版本都將保留，無論有多少。

versioningConfiguration： unlimited = FALSE，maxVersions = Y	過去 90 天沒有超過 Y 個版本將會保留。	高達 Y 個版本將會保留，無論它們已保留多久。	過去 X 天不超過 Y 個版本將會保留。

訊息承載語法

您發送到的消息有效載荷（數據）的字段名稱AWS IoT Analytics：

- 只能包含英數字元和底線 (_)；不允許使用其他特殊字元
- 開頭必須為字母字元或一個底線 (_)。
- 不可包含連字號 (-)。
- 在正則表達式術語中：「^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$」。
- 不得超過 255 個字元。
- 不區分大小寫。同一有效負載中名為「foo」和「FOO」的字段被認為是重複的。

例如，在訊息承載中，{"temp_01": 29} 或 {"_temp_01": 29} 為有效值，但 {"temp-01": 29}、{"01_temp": 29} 或 {"__temp_01": 29} 皆為無效值。

使用AWS IoT SiteWise資料

AWS IoT SiteWise是一項受管服務，您可以使用此服務來大規模地收集、建模、分析和可視化工業裝置的資料。此服務提供資產模型框架，來建立工業裝置、程序和設施的顯示方式。

搭配AWS IoT SiteWise資產模型，您可以定義要使用哪些工業裝置資料，以及如何處理您的資料，來將其轉化為複雜的指標。您可以配置資產模型以收集和處理AWS雲端。如需詳細資訊，請參閱 。AWS IoT SiteWise使用者指南。

AWS IoT Analytics與 整合AWS IoT SiteWise，以便您可以在AWS IoT SiteWise資料。要開始查詢您的AWS IoT SiteWise數據創建數據存儲，請按照設定儲存組態中的AWS IoT SiteWise使用者指南。然後，請依照建立資料集AWS IoT SiteWise資料 (主控台)或建立資料集AWS IoT SiteWise資料 (AWS CLI)若要建立AWS IoT Analytics數據集並對您的工業數據運行 SQL 查詢。

主題

- 建立AWS IoT Analytics資料集AWS IoT SiteWise資料
- 存取資料集內容
- 教學課程：查詢 AWS IoT SiteWise 資料 AWS IoT Analytics

建立AWS IoT Analytics資料集AWS IoT SiteWise資料

同時AWS IoT Analytics資料集包含您用來查詢資料存放區中資料的 SQL 語句和表達式，以及在您指定的日期和時間重複查詢的選用排程。您可以使用類似於亞馬遜 CloudWatch 排程表達式創建可選計劃。

> **Note**
>
> 數據集通常是以表格形式組織的數據集合。相比之下,AWS IoT Analytics通過將 SQL 查詢應用於數據存儲中的數據來創建數據集。

請依照下列步驟開始建立資料集AWS IoT SiteWise資料。

主題

- 建立資料集AWS IoT SiteWise資料 (主控台)
- 建立資料集AWS IoT SiteWise資料 (AWS CLI)

建立資料集AWS IoT SiteWise資料 (主控台)

使用下列步驟來建立資料集AWS IoT Analytics控制台AWS IoT SiteWise資料。

建立資料集

1. 在 中https://console.aws.amazon.com/iotanalytics/，在左側導航窗格，選擇資料集。
2. 在建立資料集頁面上，選擇建立 SQL。
3. 在指定資料集頁面上，指定資料集的詳細資訊。
   a. 為資料集輸入名稱。
   b. 適用於資料存放區源下，選擇用來標識您的AWS IoT SiteWise資料存放區。
   c. （可選）對於標籤中，將一個或多個自訂標籤 (鍵/值對) 新增至資料集
4. 使用 SQL 表達式查詢數據並回答分析性問題。
   a. 在 中作者查詢字段中，輸入使用通配符顯示最多五行數據的 SQL 查詢。

```
SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
```

如需支援的 SQL 功能的詳細資訊，請參AWS IoT Analytics，請參中的 SQL 表達式AWS IoT Analytics。或者，請參教學課程：查詢 AWS IoT SiteWise 資料 AWS IoT Analytics，瞭解可以提供數據洞察的統計查詢示例。

b. 您可以選擇測試查詢來驗證您的輸入是否正確，並在查詢後面的表中顯示結果。

> **Note**
>
> 由於Amazon Athena 限制正在運行的查詢的最大數量，則應將 SQL 查詢限製為合理的大小，以便它不會長時間運行。

5. （可選）使用指定時間範圍內的數據創建數據集內容時，某些數據可能無法及時到達以進行處理。要允許延遲，可以指定偏移或增量。如需詳細資訊，請參閱 透過 Amazon CloudWatch 活動取得延遲資料通知。

   配置數據選擇過濾器後配置數據選擇過濾器頁面上，選擇下一頁。

6. （可選）在「設置查詢調度」頁，您可以安排此查詢定期運行以刷新數據集。可以隨時建立和編輯資料集排程。

   > **Note**
   >
   > 資料來自AWS IoT SiteWise攝取到AWS IoT Analytics每六個小時。我們建議選擇六小時或更長時間的頻率。

   選擇和選項頻率，然後選擇下一頁。

7. AWS IoT Analytics將創建此數據集內容的版本並存儲指定時間段內的分析結果。我們建議 90 天，但您可以選擇設置自定義保留策略。您還可以限制數據集內容的存儲版本數。

   選擇您的選項後，在配置數據集的結果頁面上，選擇下一頁。

8. （可選）您可以將數據集結果的傳遞規則配置到特定目標，例如AWS IoT Events。

   選擇您的選項後，在配置數據集內容傳輸規則頁面上，選擇下一頁。

9. 檢視您的選擇，然後選擇建立資料集。

10. 驗證您的新數據集是否顯示在資料集(憑證已建立！) 頁面上的名稱有些許差異。

建立資料集AWS IoT SiteWise資料 (AWS CLI)

執行下列命令AWS CLI命令開始查詢您的AWS IoT SiteWise資料。

此處顯示的示例使用AWS Command Line Interface(AWS CLI。如需詳細資訊，請參AWS CLI，請參AWS Command Line Interface使用者指南。如需 CLI 命令的詳細資訊，請參AWS IoT Analytics，請參IotAnalytics中的AWS Command Line Interface參考。

建立資料集

1. 執行下列命令`create-dataset`命令來建立資料集

   ```
   aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
   ```

   其中：`my_dataset.json`文件包含下列內容。

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName":"my_action",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 5"
               }
           }
       ]
   }
   ```

   如需支援的 SQL 功能的詳細資訊，請參AWS IoT Analytics，請參中的 SQL 表達式AWS IoT Analytics。或者，請參教學課程：查詢 AWS IoT SiteWise 資料 AWS IoT Analytics，瞭解可以提供數據洞察的統計查詢示例。

2. 執行下列命令`create-dataset-content`命令通過運行查詢來創建數據集內容。

   ```
   aws iotanalytics create-dataset-content --dataset-name my_dataset
   ```

存取資料集內容

SQL 查詢的結果是您的資料集內容，存儲為 CSV 格式的文件。系統會透過 Amazon S3 提供該檔案。下列步驟將顯示如何檢查結果是否已準備就緒，然後下載該文件。

主題

存取資料集內容AWS IoT Analytics(主控台)

如果您的資料集包含任何資料，則可以在AWS IoT Analytics主控台。

若要存取您的AWS IoT Analytics資料集結果

1. 在主控台中，在資料集頁面上，選擇您要存取的資料集的名稱。
2. 在資料集摘要頁面上，選擇內容選項卡。
3. 在 中資料集內容表中，選擇要預覽結果或下載結果的 csv 文件的查詢名稱。

存取資料集內容AWS IoT Analytics(AWS CLI)

如果您的資料集包含任何資料，則可以預覽並下載您的 SQL 查詢結果。

此處顯示的示例使用AWS Command Line Interface(AWS CLI。如需詳細資訊，請參AWS CLI，請參AWS Command Line Interface使用者指南。如需 CLI 命令的詳細資訊，請參AWS IoT Analytics，請參IotAnalytics中的AWS Command Line Interface參考。

若要存取您的AWS IoT Analytics資料集結果 (AWS CLI)

1. 執行下列命令`get-dataset-content`命令查看查詢結果。

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. 如果您的資料集包含任何資料，則`get-dataset-content`, 具有`"state": "SUCCEEDED"`中的`status`欄位，如以下範例所示。

```
{
    "timestamp": 1508189965.746,
    "entries": [
```

```
        {
          "entryName": "my_entry_name",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-
Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

3. 來自`get-dataset-content`包含`dataURI`，它是輸出結果的已簽署 URL。它的有效期間很短 (幾個小時)。訪問`dataURI`訪問 SQL 查詢結果的 URL。

> **Note**
>
> 根據您的工作流程，在存取內容前您可能會想要一律呼叫 `get-dataset-content`，因為呼叫此命令會產生新的已簽署 URL。

教學課程：查詢 AWS IoT SiteWise 資料 AWS IoT Analytics

本教學課程將示範如何在中查詢 AWS IoT SiteWise 資料 AWS IoT Analytics。本教學課程使用中 AWS IoT SiteWise 示範的資料，為風力發電場提供資料範例集。

> **Important**
>
> 您將為此示範建立和使用的資源付費。

主題

必要條件

在此教學課程中，您需執行下列資源：

- 您必須擁有一個 AWS 帳戶才能開始使用 AWS IoT SiteWise 和 AWS IoT Analytics。如果您沒有帳戶，請依照建立 AWS 帳戶中的程序執行。
- 執行 Windows、macOS、Linux 或 Unix 的開發電腦，用來存取 AWS Management Console。如需詳細資訊，請參閱 AWS Management Console入門。
- AWS IoT SiteWise 定義 AWS IoT SiteWise 模型和資產的資料，並串流代表風電場設備資料的資料。若要建立資料，請依照《AWS IoT SiteWise 使用指南》中的〈建立 AWS IoT SiteWise 示範〉中的步驟執行。
- 您管理的現有資料存放區中的 AWS IoT SiteWise 示範風電場設備資料。有關如何為資料建立資料倉庫的詳細 AWS IoT SiteWise 資訊，請參閱《AWS IoT SiteWise 使用指南》中的〈規劃儲存設定〉。

> **Note**
>
> 您的中 AWS IoT SiteWise 繼 AWS IoT SiteWise 資料會在建立後立即顯示在資料倉庫中；不過，原始資料最多可能需要六小時才會顯示。同時，您可以建立 AWS IoT Analytics 資料集並對中繼資料執行查詢。

下一步驟

載入並驗證資料

載入並驗證資料

您在本教學課程中查詢的資料是一組範例 AWS IoT SiteWise 資料，可為風力發電場中的風力發電機模型進行模型。

> **Note**
>
> 您將在本自學課程中查詢資料倉庫中的三個表格：
>
> - `raw`-包含每個資產的原始、未處理的資料。
> - `asset_metadata`-包含有關每個資產的一般資訊。
> - `asset_hierarchy_metadata`-包含資產之間關係的相關資訊。

若要在此自學課程中執行 SQL 查詢

1. 請依照建立資料集AWS IoT SiteWise資料 (主控台)或中建立資料集AWS IoT SiteWise資料 (AWS CLI)的步驟建立資 AWS IoT SiteWise 料 AWS IoT Analytics 集。
2. 若要在本教學課程中更新資料集查詢，請執行下列動作。

   a. 在 AWS IoT Analytics 主控台的 [資料集] 頁面上，選擇您在上一頁建立的資料集名稱。

   b. 在資料集摘要頁面上，選擇 [編輯] 以編輯您的 SQL 查詢。

   c. 若要在查詢之後的表格中顯示結果，請選擇 [測試查詢]。

   或者，您也可以執行下列update-dataset命令，以修改 SQL 查詢 AWS CLI。

   ```
   aws iotanalytics update-dataset --cli-input-json file://update-query.json
   ```

   update-query.json 的內容：

   ```
   {
       "datasetName": "my_dataset",
       "actions": [
           {
               "actionName": "myDatasetUpdateAction",
               "queryAction": {
                   "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
    LIMIT 3"
               }
           }
       ]
   }
   ```

3. 在 AWS IoT Analytics 主控台或使用 AWS CLI，對您的資料執行下列查詢，以確認資料asset_metadata表是否已成功載入。

   ```
   SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
   ```

   同樣，您可以驗證您的asset_hierarchy_metadata和raw表格不是空的。

後續步驟

資料探索

資料探索

建立資 AWS IoT SiteWise 料並將其載入資料存放區後，您可以在中建立資料 AWS IoT Analytics 集並執行 SQL 查詢， AWS IoT Analytics 以探索有關資產的深入解析。下列查詢示範如何在執行統計查詢之前瀏覽資料。

使用 SQL 查詢探索資料的步驟

1. 檢視每個表格中的資料欄和值範例，例如原始資料表。

```
SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
```

2. 用SELECT DISTINCT於查詢asset_metadata表格並列出 AWS IoT SiteWise 資產的 (唯一) 名稱。

```
SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY
 assetname
```

3. 若要列示有關特定 AWS IoT SiteWise 資產之性質的資訊，請使用WHERE子句。

```
SELECT assetpropertyname,
     assetpropertyunit,
     assetpropertydatatype
FROM my_iotsitewise_datastore.asset_metadata
WHERE assetname = 'Demo Turbine Asset 2'
```

4. 使用 AWS IoT Analytics，您可以聯結資料倉庫中兩個或多個表中的資料，例如以下範例所示。

```
SELECT * FROM my_iotsitewise_datastore.raw AS raw
JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
ON raw.seriesId = asset_metadata.timeseriesId
```

若要檢視資產之間的所有關係，請使用下列查詢中的JOIN功能。

```
SELECT DISTINCT parent.assetName as "Parent name",
    child.assetName AS "Child name"
FROM (
    SELECT sourceAssetId AS parent,
           targetAssetId AS child
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
```

```
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
    ON relations.child = child.assetId
JOIN my_iotsitewise_datastore.asset_metadata AS parent
    ON relations.parent = parent.assetId
```

下一步驟

[執行統計查詢]

執行統計查詢

現在您已經探索了 AWS IoT SiteWise 資料，您可以執行統計查詢，為您的工業設備提供寶貴的見解。下列查詢示範的是一些您可以擷取的資訊。

執行 AWS IoT SiteWise 示範風電場資料的統計查詢

1. 執行以下 SQL 指令，以尋找具有特定資產之數值的所有性質的最新值 (展示渦輪資產 4)。

```
SELECT assetName,
    assetPropertyName,
    assetPropertyUnit,
    max_by(value, timeInSeconds) AS Latest
FROM (
    SELECT *,
        CASE assetPropertyDataType
        WHEN 'DOUBLE' THEN
        cast(doubleValue AS varchar)
        WHEN 'INTEGER' THEN
        cast(integerValue AS varchar)
        WHEN 'STRING' THEN
        stringValue
        WHEN 'BOOLEAN' THEN
        cast(booleanValue AS varchar)
        ELSE NULL
        END AS value
    FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
    JOIN my_iotsitewise_datastore.raw AS raw
        ON raw.seriesId = asset_metadata.timeSeriesId
    WHERE startYear=2021
        AND startMonth=7
```

```
        AND startDay=8
        AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. 結合中繼資料表和原始資料表，以識別所有資產的最大風速屬性，以及其父資產。

```
SELECT child_assets_data_set.parentAssetId,
        child_assets_data_set.childAssetId,
        asset_metadata.assetPropertyId,
        asset_metadata.assetPropertyName,
        asset_metadata.timeSeriesId,
        raw_data_set.max_speed
FROM (
   SELECT sourceAssetId AS parentAssetId,
        targetAssetId AS childAssetId
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
    ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
    SELECT seriesId, MAX(doubleValue) AS max_speed
    FROM my_iotsitewise_datastore.raw
    GROUP BY seriesId
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC
```

3. 若要尋找資產 (示範渦輪機資產 2) 的特定性質 (風速) 的平均值，請執行以下 SQL 指令。您必須更換my_bucket_id為值區的 ID。

```
SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw
WHERE seriesId =
    (SELECT timeseriesId
    FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
    WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
            AND asset_metadata.assetpropertyname = 'Wind Speed')
```

下一步驟

清理您的教學課程資源

清理您的教學課程資源

完成教學課程後，請清理資源以避免產生費用。

若要刪除您的 AWS IoT SiteWise 示範

示 AWS IoT SiteWise 範會在一周後自行刪除。如果您使用的示範資源完成，您可以先刪除示範。若要手動刪除示範，請使用下列步驟。

1. 導覽至 AWS CloudFormation 主控台。
2. IoTSiteWiseDemoAssets從「堆疊」清單中選擇。
3. 選擇刪除。當您刪除堆疊時，為示範建立的所有資源均會受到刪除。
4. 在確認對話方塊中，輸入刪除。

   堆疊大約需要 15 分鐘的時間來刪除。如果示範無法刪除，請再次選擇右上角的 Delete (刪除)。如果示範無法再次刪除，請依照 AWS CloudFormation 主控台中的步驟略過刪除失敗的資源，然後再試一次。

刪除您的資料倉庫

- 若要刪除受管理的資料存放區，請執行 CLI 命令`delete-datastore`，如以下範例所示。

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

若要刪除資 AWS IoT Analytics 料集

- 若要刪除資料集，請執行 CLI 命令`delete-dataset`，如下列範例所示。您不需要在執行此作業之前刪除資料集的內容。

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

> **Note**
>
> 此命令不會產生輸出。

管道活動

最簡單的功能性管道是將頻道連接至資料存放區，成為具有兩個活動的管道：一個 channel 活動和一個 datastore 活動。您可以新增其他活動至管道，以獲得更強大的訊息處理。

您可以使用RunPipelineActivity操作，以模擬在您提供的消息有效負載上運行管道活動的結果。在開發和偵錯管道活動時，這可能相當實用。RunPipelineActivity 例子演示了如何使用它。

頻道活動

管道中的第一個活動必須是channel決定要處理之訊息來源的活動。

```
{
    "channel": {
        "name": "MyChannelActivity",
        "channelName": "mychannel",
        "next": "MyLambdaActivity"
    }
}
```

Data Datastore 活動

datastore 活動指定已處理資料的儲存位置，這是最後一個活動。

```
{
    "datastore": {
        "name": "MyDatastoreActivity",
        "datastoreName": "mydatastore"
    }
}
```

AWS Lambda活動

您可以使用**lambda**活動對訊息執行複雜的處理。例如，您可以使用來自外部 API 操作輸出的資料來豐富訊息，或根據 Amazon DynamoDB 的邏輯篩選訊息。不過，在進入資料存放區之前，您無法使用此管線活動來新增其他訊息或移除現有訊息。

所以此AWS Lambda在一個使用的函數**lambda**活動必須接收並返回 JSON 對象的數組。如需範例，請參閱the section called “Lambda 函數範例 1”。

授予AWS IoT Analytics允許叫用 Lambda 函數，您必須新增原則。例如，執行下列 CLI 命令並取代*exampleFunctionName*使用 Lambda 函數的名稱，取代*123456789012*與您的AWS帳戶 ID，並使用管道叫用給定 Lambda 函數的管道的 Amazon Resource Name (ARN)。

```
aws lambda add-permission --function-name exampleFunctionName --
action lambda:InvokeFunction --statement-id iotanalytics --principal
 iotanalytics.amazonaws.com --source-account 123456789012 --source-arn
 arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

命令會傳回下列：

```
{
    "Statement": "{\"Sid\":\"iotanalyticsa\",\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"iotanalytics.amazonaws.com\"},\"Action\":
\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:aws-region:aws-
account:function:exampleFunctionName\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceAccount\":\"123456789012\"},\"ArnLike\":{\"AWS:SourceArn\":
\"arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline\"}}}"
}
```

如需詳細資訊，請參閱對 使用以資源為基礎的政策AWS Lambda在AWS Lambda開發人員指南。

Lambda 函數範例 1

在此範例中，Lambda 函數會根據原始訊息中的資料新增資訊。裝置發佈承載的訊息，如下的範例所示：

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
```

```
}
```

並且該設備具有以下管道定義。

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "foobar_channel",
                    "name": "foobar_channel_activity",
                    "next": "lambda_foobar_activity"
                }
            },
            {
                "lambda": {
                    "lambdaName": "MyAnalyticsLambdaFunction",
                    "batchSize": 5,
                    "name": "lambda_foobar_activity",
                    "next": "foobar_store_activity"
                }
            },
            {
                "datastore": {
                    "datastoreName": "foobar_datastore",
                    "name": "foobar_store_activity"
                }
            }
        ],
        "name": "foobar_pipeline",
        "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
    }
}
```

下面的 Lambda 函數 (`MyAnalyticsLambdaFunction`) 將 GMaps URL 和溫度 (以華氏度為單位) 新增至訊息。

```
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Lambda 函數範例 2

有用的技巧就是壓縮和序列化訊息承載，以降低傳輸和存放成本。在第二個範例中，Lambda 函數假設訊息承載代表 JSON 原始資料，該 JSON 原始資料經過壓縮，然後以 base64 編碼 (序列化) 為字串。它返回原始的 JSON。

```
import base64
import gzip
import json
import logging
import sys

# Configure logging
logger = logging.getLogger()
```

```
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

AddAttributes 活動

addAttributes 活動根據訊息中的現有屬性來新增屬性。這使您可以在存儲之前更改消息的形狀。例如，您可以使用 addAttributes，標準化來自不同世代之裝置韌體的資料。

請看下列輸入訊息。

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ]
    }
}
```

所以此addAttributes活動看起來如下。

```
{
    "addAttributes": {
        "name": "MyAddAttributesActivity",
        "attributes": {
            "device.id": "id",
            "device.coord[0]": "lat",
            "device.coord[1]": "lon"
        },
        "next": "MyRemoveAttributesActivity"
    }
}
```

此活動會將裝置 ID 移至根層級，並擷取coord陣列，將它們提升為稱為頂層屬性lat和lon。此活動的結果會將輸入訊息轉換為下列範例。

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

原始裝置屬性仍然存在。您如果想要將其移除，可以使用 removeAttributes 活動。

RemoveAttributes 活動

removeAttributes 活動從訊息移除屬性。例如，給定的消息是結果addAttributes活動。

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

要標準化該消息，使其僅包含根級別的所需數據，請使用以下命令removeAttributes活動。

```
{
    "removeAttributes": {
        "name": "MyRemoveAttributesActivity",
        "attributes": [
            "device"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

成果如以下訊息沿著管道流。

```
{
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

SelectAttributes 活動

selectAttributes 活動僅會使用原始訊息的指定屬性來建立新訊息，而其他每個屬性皆會遭捨棄。此外，selectAttributes 只會在訊息的根目錄下建立新屬性。因此，假定此訊息：

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ],
        "temp": 50,
        "hum": 40
    },
    "light": 90
}
```

與此活動：

```
{
    "selectAttributes": {
        "name": "MySelectAttributesActivity",
```

```
        "attributes": [
            "device.temp",
            "device.hum",
            "light"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

結果是下列訊息流經管線。

```
{
    "temp": 50,
    "hum": 40,
    "light": 90
}
```

同樣地，selectAttributes 只能建立根層級的物件。

篩選活動

filter 活動會根據其屬性篩選訊息。此活動中使用的表達式看起來像一個 SQLWHERE子句，它必須返回一個布爾值。

```
{
    "filter": {
        "name": "MyFilterActivity",
        "filter": "temp > 40 AND hum < 20",
        "next": "MyDatastoreActivity"
    }
}
```

DeviceRegistryEnrich 活動

所以此deviceRegistryEnrich活動可讓您從AWS IoT裝置登錄到您的訊息承載。舉例而言，若為以下的 訊息：

```
{
    "temp": 50,
```

```
    "hum": 40,
    "device" {
        "thingName": "my-thing"
    }
}
```

deviceRegistryEnrich 活動如下所示：

```
{
    "deviceRegistryEnrich": {
        "name": "MyDeviceRegistryEnrichActivity",
        "attribute": "metadata",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

輸出訊息現在看起來像這個範例。

```
{
    "temp" : 50,
    "hum" : 40,
    "device" {
        "thingName" : "my-thing"
    },
    "metadata" : {
        "defaultClientId": "my-thing",
        "thingTypeName": "my-thing",
        "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",
        "version": 1,
        "thingName": "my-thing",
        "attributes": {},
        "thingId": "aaabbbccc-dddeeef-gghh-jjkk-llmmnnoopp"
    }
}
```

您必須在活動定義的 roleArn 欄位中指定角色，且該角色需連接適當許可。此令命的權限政策如下的範例所示：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

該角色的信任政策則如下所示：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

DeviceShadowEnrich 活動

一個deviceShadowEnrich活動將資訊從AWS IoTDevice Shadow 服務到訊息。例如，假定訊息：

```
{
    "temp": 50,
    "hum": 40,
    "device": { "thingName": "my-thing" }
}
```

以及下列 deviceShadowEnrich 活動：

```
{
    "deviceShadowEnrich": {
        "name": "MyDeviceShadowEnrichActivity",
        "attribute": "shadow",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

結果是一則訊息，看起來會如以下範例。

```
{
    "temp": 50,
    "hum": 40,
    "device": {
        "thingName": "my-thing"
    },
    "shadow": {
        "state": {
            "desired": {
                "attributeX": valueX, ...
            },
            "reported": {
                "attributeX": valueX, ...
            },
            "delta": {
                "attributeX": valueX, ...
            }
        },
        "metadata": {
            "desired": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            },
            "reported": ": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            }
```

```
        },
        "timestamp": timestamp,
        "clientToken": "token",
        "version": version
    }
}
```

您必須在活動定義的 roleArn 欄位中指定角色，且該角色需連接適當許可。此角色必須具有類似以下的權限政策。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
            ],
            "Resource": [
                "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
            ]
        }
    ]
}
```

該角色的信任政策則如下所示：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

數學活動

math 活動會透過訊息的屬性來運算數學表達式。表達式必須傳回數字。舉例而言，假定是以下輸入訊息：

```
{
    "tempF": 50,
}
```

透過以下 math 活動處理後：

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "(tempF - 32) / 2",
        "attribute": "tempC",
        "next": "MyDatastoreActivity"
    }
}
```

產生的訊息看起來如下：

```
{
    "tempF" : 50,
    "tempC": 9
}
```

數學活動運算子和函數

您可以在 math 活動中使用以下運算子：

+	加法
-	減法
*	乘法
/	除法

%	模數

您可以在 math 活動中使用以下函數：

- abs(Decimal)
- acos(Decimal)
- asin(Decimal)
- atan(Decimal)
- atan2(Decimal, Decimal)
- ceil(Decimal)
- cos(Decimal)
- cosh(Decimal)
- exp(Decimal)
- ln(Decimal)
- log(Decimal)
- mod(Decimal, Decimal)
- power(Decimal, Decimal)
- round(Decimal)
- sign(Decimal)
- sin(Decimal)
- sinh(Decimal)
- sqrt(Decimal)
- tan(Decimal)
- tanh(Decimal)
- 主幹（十進制，整數）

abs(Decimal)

傳回某個數字的絕對值。

範例:abs(-5)傳回 5。

引數類型	結果
Int	Int，引數的絕對值。
Decimal	Decimal，引數的絕對值
Boolean	Undefined 。
String	Decimal。結果為引數的絕對值。如果字串無法轉換，則結果為 Undefined 。
Array	Undefined 。
物件	Undefined 。
Null	Undefined 。
未定義	Undefined 。

acos(Decimal)

以弧度傳回數字的反餘弦值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:acos(0) = 1.5707963267948966

引數類型	結果
Int	Decimal (使用雙精度)，引數的反向餘弦值。傳回的虛數結果為 Undefined 。
Decimal	Decimal (使用雙精度)，引數的反向餘弦值。傳回的虛數結果為 Undefined 。
Boolean	Undefined 。
String	Decimal(使用雙精度)，引數的反向餘弦值。如果字串無法轉換，則結果為 Undefined 。傳回的虛數結果為 Undefined 。

引數類型	結果
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

asin(Decimal)

以弧度傳回數字的反正弦值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:`asin(0)` = 0.0

引數類型	結果
Int	Decimal (使用雙精度)，引數的反向正弦值。傳回的虛數結果為 Undefined 。
Decimal	Decimal (使用雙精度)，引數的反向正弦值。傳回的虛數結果為 Undefined 。
Boolean	Undefined .
String	Decimal (使用雙精度)，引數的反向正弦值。如果字串無法轉換，則結果為 Undefined 。傳回的虛數結果為 Undefined 。
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

atan(Decimal)

以弧度傳回數字的反正切值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:`atan(0)` = 0.0

引數類型	結果
Int	Decimal (使用雙精度)，引數的反向正切值。傳回的虛數結果為 Undefined 。
Decimal	Decimal (使用雙精度)，引數的反向正切值。傳回的虛數結果為 Undefined 。
Boolean	Undefined .
String	Decimal (使用雙精度)，引數的反向正切值。如果字串無法轉換，則結果為 Undefined 。傳回的虛數結果為 Undefined 。
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

atan2(Decimal, Decimal)

以弧度傳回 x 軸正軸和兩個引數所定義的 (x, y) 點之間的角度。逆時鐘角度 (上半平面，y > 0) 的角度為正，順時鐘角度為負Decimal引數在套用函數前會四捨五入至雙精度。

範例:`atan(1, 0)` = 1.5707963267948966

引數類型	引數類型	結果
Int / Decimal	Int / Decimal	Decimal(使用雙精度)，x 軸與指定的 (x, y) 點之間的角度

引數類型	引數類型	結果
Int / Decimal / String	Int / Decimal / String	Decimal，所述之點的反向正切值。如果字串無法轉換，則結果為 Undefined 。
其他值	其他值	Undefined .

ceil(Decimal)

將指定的 Decimal 無條件進位至最近的 Int。

範例：

ceil(1.2) = 2

ceil(11.2) = -1

引數類型	結果
Int	Int，引數值。
Decimal	Int，該字符串被轉換為Decimal並四捨五入到最接近的Int。如果字串無法轉換為 Decimal，則結果為 Undefined 。
其他值	Undefined .

cos(Decimal)

以弧度傳回數字的餘弦值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:cos(0) = 1

引數類型	結果
Int	Decimal (使用雙精度)，引數的餘弦值。傳回的虛數結果為 Undefined 。

引數類型	結果
Decimal	Decimal (使用雙精度)，引數的餘弦值。傳回的虛數結果為 Undefined 。
Boolean	Undefined .
String	Decimal (使用雙精度)，引數的餘弦值。如果字串無法轉換為 Decimal，則結果為 Undefined 。傳回的虛數結果為 Undefined 。
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

cosh(Decimal)

以弧度傳回數字的雙曲餘弦值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:cosh(2.3) = 5.037220649268761

引數類型	結果
Int	Decimal (使用雙精度)，引數的雙曲餘弦值。傳回的虛數結果為 Undefined 。
Decimal	Decimal (使用雙精度)，引數的雙曲餘弦值。傳回的虛數結果為 Undefined 。
Boolean	Undefined .
String	Decimal (使用雙精度)，引數的雙曲餘弦值。如果字串無法轉換為 Decimal，則結果為 Undefined 。傳回的虛數結果為 Undefined 。

引數類型	結果
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

exp(Decimal)

傳回e引發到十進制參數。Decimal引數在套用函數前會四捨五入至雙精度。

範例:exp(1) = 1

引數類型	結果
Int	Decimal(使用雙精度)，e ^ 引數。
Decimal	Decimal(使用雙精度)，e ^ 引數
String	Decimal(使用雙精度)，e ^ 引數。如果String無法轉換為Decimal，結果如果Undefined 。
其他值	Undefined .

ln(Decimal)

傳回引數的自然對數。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:ln(e) = 1

引數類型	結果
Int	Decimal (使用雙精度)，引數的自然對數。
Decimal	Decimal(使用雙精度)，引數的自然對數

引數類型	結果
Boolean	Undefined .
String	Decimal (使用雙精度)，引數的自然對數。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined .
物件	Undefined .
Null	Undefined .
未定義	Undefined .

log(Decimal)

傳回引數以 10 為底的對數。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:log(100) = 2.0

引數類型	結果
Int	Decimal (使用雙精度)，引數以 10 為底的對數。
Decimal	Decimal (使用雙精度)，引數以 10 為底的對數。
Boolean	Undefined .
String	Decimal (使用雙精度)，引數以 10 為底的對數。如果 String 無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined .
物件	Undefined .

引數類型	結果
Null	Undefined .
未定義	Undefined .

mod(Decimal, Decimal)

傳回第二個引數第一個引數除法的餘數。您也可以使用%作為相同模功能的中綴運算符。

範例:mod(8, 3) = 3

左運算元	右運算元	輸出
Int	Int	Int，第二個參數的第一個參數模數。
Int / Decimal	Int / Decimal	Decimal，第二個參數的第一個參數模數。
String / Int / Decimal	String / Int / Decimal	如果所有字符串轉換為Decimals，如果第一個引數以模為第二個引數。否則為 Undefined 。
其他值	其他值	Undefined .

power(Decimal, Decimal)

傳回第一個引數次方的第二個引數值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:power(2, 5) =

引數類型 1	引數類型 2	輸出
Int / Decimal	Int / Decimal	Decimal (使用雙精度)，第一個引數次方的第二個引數值。

引數類型 1	引數類型 2	輸出
Int / Decimal / String	Int / Decimal / String	Decimal (使用雙精度)，第一個引數次方的第二個引數值。任何字串都會轉換為Decimals。如果任何 String 無法轉換為 Decimal，則結果為 Undefined 。
其他值	其他值	Undefined .

round(Decimal)

將指定的 Decimal 無條件進位至最接近的 Int。如果 Decimal 與兩個 Int 值 (例如，0.5) 等距，則 Decimal 會無條件進位。

範例：

Round(1.2) = 1

Round(1.5) = 2

Round(1.7) = 2

Round(-1.1) = -1

Round(-1.5) = -2

引數類型	結果
Int	引數
Decimal	Decimal 是要向下捨入到最接近的 Int。
String	Decimal 是要向下捨入到最接近的 Int。如果字串無法轉換為 Decimal，則結果為 Undefined 。

引數類型	結果
其他值	Undefined .

sign(Decimal)

傳回所給數字的符號。當引數的符號為正值時，傳回 1。當引數的符號為負值時，傳回 -1。如果引數為 0，傳回 0。

範例：

`sign(-7) = -1`

`sign(0)= 0`

`sign(13) = 1`

引數類型	結果
Int	Int，Int 值的符號。
Decimal	Int，Decimal 值的符號。
String	Int，Decimal 值的符號。如果轉換為字符串Decimal價值，和符號Decimal返回值。如果 String 無法轉換為 Decimal，則結果為 Undefined 。
其他值	Undefined .

sin(Decimal)

以弧度傳回數字的正弦值。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:`sin(0) = 0.0`

引數類型	結果
Int	Decimal (使用雙精度)，引數的正弦值。

引數類型	結果
Decimal	Decimal (使用雙精度)，引數的正弦值。
Boolean	Undefined 。
String	Decimal，引數的正弦值。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined 。
Object	Undefined 。
Null	Undefined 。
Undefined	Undefined 。

sinh(Decimal)

以弧度傳回數字的雙曲正弦值。Decimal 值在套用函數前會四捨五入至雙精度。結果為雙精度的 Decimal 值。

範例:`sinh(2.3)` = 4.936961805545957

引數類型	結果
Int	Decimal (使用雙精度)，引數的雙曲正弦值。
Decimal	Decimal (使用雙精度)，引數的雙曲正弦值。
Boolean	Undefined 。
String	Decimal，引數的雙曲正弦值。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined 。
Object	Undefined 。
Null	Undefined 。

引數類型	結果
Undefined	Undefined .

sqrt(Decimal)

傳回數字的平方根。Decimal 引數在套用函數前會四捨五入至雙精度。

範例:sqrt(9) = 3.0

引數類型	結果
Int	引數的平方根。
Decimal	引數的平方根。
Boolean	Undefined .
String	引數的平方根。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

tan(Decimal)

以弧度傳回數字的正切值。Decimal 值在套用函數前會四捨五入至雙精度。

範例:tan(3) = -0.1425465430742778

引數類型	結果
Int	Decimal (使用雙精度)，引數的正切值。

引數類型	結果
Decimal	Decimal (使用雙精度)，引數的正切值。
Boolean	Undefined 。
String	Decimal (使用雙精度)，引數的正切值。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined 。
Object	Undefined 。
Null	Undefined 。
Undefined	Undefined 。

tanh(Decimal)

以弧度傳回數字的雙曲正切值。Decimal 值在套用函數前會四捨五入至雙精度。

範例:tanh(2.3) = 0.9800963962661914

引數類型	結果
Int	Decimal (使用雙精度)，引數的雙曲正切值。
Decimal	Decimal (使用雙精度)，引數的雙曲正切值。
Boolean	Undefined 。
String	Decimal (使用雙精度)，引數的雙曲正切值。如果字串無法轉換為 Decimal，則結果為 Undefined 。
Array	Undefined 。
Object	Undefined 。

引數類型	結果
Null	Undefined .
Undefined	Undefined .

主幹（十進制，整數）

將第一個引數截為第二的引數所指定的 Decimal 位數。如果第二個引數小於零，則會設定為零。如果第二個引數大於 34，則會設定為 34。結果中會刪除尾隨零。

範例：

trunc(2.3, 0) = 2

trunc(2.3123, 2)= 2.31

trunc(2.888, 2)= 2.88

trunc(2.00, 5) = 2

引數類型 1	引數類型 2	結果
Int	Int	來源值。
Int / Decimal / String	Int / Decimal	第一個引數會截短為第二個引數所指定的長度。第二個引數若非 Int，會無條件捨去至最接近的 Int。字符串被轉換為Decimal值。如果字串轉換失敗，則結果為 Undefined 。
其他值		未定義。

RunPipelineActivity

以下為您將如何使用RunPipelineActivity用於測試管線活動的命令。在此範例中，我們測試數學活動。

1. 建立maths.json檔案，其中包含您要測試之管線活動的定義。

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

2. 建立檔案payloads.json檔案，其中包含用來測試管線活動的範例承載。

```
[
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. 呼叫RunPipelineActivities從命令列執行作業。

```
aws iotanalytics run-pipeline-activity  --pipeline-activity file://maths.json  --
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

成果如以下。

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

結果中列出的承載是 Base64 編碼字串。當解碼這些字串時，您會得到下列結果。

```
{"humidity":52,"temp":68,"tempC":20}
{"humidity":52,"temp":32,"tempC":0}
```

重新處理頻道消息

AWS IoT Analytics使您能夠重新處理通道數據。在下列情況中，這可能非常有用：

- 您想要重新播放現有的擷取資料，而不是從頭開始。
- 您對管道進行更新，並想將現有的資料 up-to-date 有變更的。
- 您希望包括在更改客户託管存儲選項、渠道權限或數據存儲之前攝入的數據。

參數

當您通過管道重新處理通道消息時AWS IoT Analytics，則必須指定下列資訊：

`StartPipelineReprocessing`

通過管道開始重新處理頻道消息。

`ChannelMessages`

指定要重新處理的一組或多組通道消息。

如果您使用channelMessages物件，則不能指定startTime和endTime。

`s3Paths`

指定標識保存頻道消息的 Amazon Simple Storage Service (Amazon S3) 物件的一個或多個密鑰。您必須使用密鑰的完整路徑。

路徑範
例：`00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.`

類型：字串陣列

陣列成員約束：1-100 個項目。

長度限制：1-1024 個字符。

`endTime`

重新處理的頻道資料結束時間 (不包括)。

如果您指定的endTime參數，則不得使用channelMessages物件。

類型：時間戳記

`startTime`

經重新處理的原始訊息資料開始時間 (包含)。

如果您指定的`startTime`參數，則不得使用`channelMessages`物件。

類型：時間戳記

`pipelineName`

要開始重新處理的管道名稱。

類型：字串

長度限制：1-128 個字元。

重新處理頻道消息 (主控台)

本教程介紹如何重新處理 Amazon S3 在AWS IoT Analytics主控台。

開始之前，請確定您想要重新處理的頻道消息已保存在客户管理的 Amazon S3 存儲體中。

1. 登入 AWS IoT Analytics 主控台。
2. 在導覽窗格中，選擇管道。
3. 選擇您的目標管道。
4. 選擇重新處理消息從動作。
5. 在管道重新處理頁面上，選擇S3 物件為了重新處理消息。

   所以此AWS IoT Analytics主控台還提供下列選項：

   - 所有可用範圍-重新處理頻道中的所有有效數據。
   - 過去 120 天-重新處理過去 120 天內到達的數據。
   - 過去 90 天-重新處理過去 90 天內到達的資料。
   - 過去 30 天-重新處理過去 30 天內到達的資料。
   - 定製範圍-重新處理在指定時間範圍內到達的數據。您可以選擇任何時間範圍。
6. 輸入存儲頻道消息的 Amazon S3 obejct 的密鑰。

   若要查找項，請執行下列作業：

a. 前往Amazon S3 主控台。
b. 選擇目標 Amazon S3 物件。
c. 根據屬性，在物件概觀部分中，複製密鑰。

7. 選擇開始重新處理。

重新處理頻道消息 (API)

當您使用`StartPipelineReprocessing`API，請留意以下事項：

- 所以此`startTime`和`endTime`參數指定原始資料的採集時間，但這只是粗略估計。您可以四捨五入到最接近的 1 小時。所以此`startTime`是包含的，但`endTime`是排他的。
- 命令會以非同步方式啟動重新處理，並立即傳回。
- 不保證重新處理訊息會依其原本收到的順序來處理。只會大致相同，但不完全一樣。
- 你最多可以彌補 1000`StartPipelineReprocessing`API 請求每 24 小時通過管道重新處理相同頻道消息。
- 重新處理您的原始資料會產生額外的成本。

如需詳細資訊，請參閱 。StartPipelineReprocessingAPI, 在AWS IoT AnalyticsAPI 參考。

取消渠道重新處理活動

要取消管道重新處理活動，請使用CancelPipelineReprocessingAPI 或選擇取消重新處理在活動頁面中的AWS IoT Analytics主控台。如果取消重新處理，則不會重新處理剩餘的數據。您必須啟動另一個重新處理請求。

使用DescribePipelineAPI 來檢查重新處理的狀態。請參`reprocessingSummaries`欄位。

自動化您的工作流程

AWS IoT Analytics提供進階資料分析AWS IoT。您可以自動收集 IoT 資料、處理它、存放它，以及使用資料分析和機器學習工具分析它。您可以執行裝載您自己的自訂分析程式碼或 Jupyter Notebook 的容器，或使用協力廠商自訂程式碼容器，因此您不必重新建立現有的分析工具。您可以使用以下功能，從資料存放區取得輸入資料，並將其饋送至自動化工作流程：

按照重複排程建立資料集內容

通過在調用時指定觸發器來安排自動創建數據集內容`CreateDataset(triggers:schedule:expression`。資料倉庫中的資料會用來建立資料集內容。您可以通過使用 SQL 查詢選擇所需的字段（`actions:queryAction:sqlQuery`。

定義非重疊、連續的時間間隔，以確保新資料集內容僅包含上次到達的資料。使用`actions:queryAction:filters:deltaTime`和`:offsetSeconds`用於指定差異時間間隔的欄位。然後指定一個觸發器，以在時間間隔過後建立資料集內容。請參閱the section called “範例 6 — 使用差異時段（CLI）建立 SQL 資料集”。

在另一個資料集完成時建立資料集內容

當另一個資料集的內容建立完成時，觸發新資料集內容的建立`triggers:dataset:name`。

自動執行您的分析應用程式

容器化您自己的自訂資料分析應用程式，並在建立另一個資料集的內容時觸發這些應用程式執行。如此一來，您就可以將根據週期性排程建立的資料集內容中的資料提供應用程式。您可以在應用程式中自動對分析結果採取行動。 (`actions:containerAction`)

在另一個資料集完成時建立資料集內容

當另一個資料集的內容建立完成時，觸發新資料集內容的建立`triggers:dataset:name`。

自動執行您的分析應用程式

容器化您自己的自訂資料分析應用程式，並在建立另一個資料集的內容時觸發這些應用程式執行。如此一來，您就可以將根據週期性排程建立的資料集內容中的資料提供應用程式。您可以在應用程式中自動對分析結果採取行動。 (`actions:containerAction`)

使用案例

將產品品質量測量自動化以降低 OpEx

您有一個系統具有測量壓力、濕度和溫度的智慧閥。系統會定期整理事件，以及在某些事件發生時 (例如，值開啟和關閉時)。搭配AWS IoT Analytics，您可以自動化分析，從這些週期性視窗彙總非重疊資料，並建立最終產品品質的重要績效指標報表。在處理每個批次之後，您可以通過最大化的運行量來衡量整體產品質量並降低運營支出。

將裝置機群的分析自動化

您每 15 分鐘針對 100 部裝置產生的資料執行分析 (演算法、資料科學或 KPI 的 ML)。每個分析週期都會產生和儲存狀態，以便下次執行分析。對於每個分析，您只想要使用指定時段內收到的資料。搭配AWS IoT Analytics您可以協調分析並為每次運行創建 KPI 和報告，然後存儲數據以備 future 分析。

將異常偵測自動化

AWS IoT Analytics可讓您將異常偵測工作流程自動化，您必須每 15 分鐘針對已到達資料存放區的新資料手動執行一次。您也可以將顯示指定期間內的裝置使用量和常用使用者的儀表板自動化。

預測工業製程結果

您有工業生產線。使用傳送至的資料AWS IoT Analytics，包括可用的製程測量，您可以操作分析工作流程以預測流程結果。模型的資料可以排列成 M x N 矩陣，其中每一列都包含來自擷取實驗室樣本的不同時間點的資料。AWS IoT Analytics透過建立差異視窗，並使用資料科學工具建立 KPI 並儲存量測裝置的狀態，協助您操作分析工作流程。

使用碼頭容器

本節包含如何建立 Docker 容器的相關資訊。如果您重複使用第三方建置的 Docker 容器，有安全風險：這些容器可以使用您的使用者許可來執行任意程式碼。在使用任何第三方容器之前，請確定您信任其撰寫者。

以下是您對最後一次執行分析後抵達的資料，設定定期資料分析所要採取的步驟：

1. 建立 Docker 容器，其中包含您的資料應用程式以及任何必要的程式庫或其他相依性。

   所以此 IotAnalytics Jupyter 擴充功能提供容器化 API，以協助進行容器化程序。您也可以執行您自己建立的影像，在其中建立或組合應用程式工具集，以執行所需的資料分析或計算。AWS IoT

Analytics可讓您透過變數來定義容器化應用程式的輸入資料來源，以及 Docker 容器輸出資料的目的地。(自定義Docker 容器輸入/輸出變量包含有關在自訂容器中使用變數的詳細資訊。)

2. 將容器上傳至 Amazon ECR 登錄檔。
3. 建立資料存放區以接收和儲存來自裝置的訊息 (資料) (iotanalytics: CreateDatastore)
4. 創建一個發送消息的渠道 (iotanalytics: CreateChannel。
5. 建立管道以將通道連接至資料存放區 (iotanalytics: CreatePipeline。
6. 建立一個可授與許可的 IAM 角色，以傳送訊息資料至AWS IoT Analytics頻道iam: CreateRole.)
7. 建立使用 SQL 查詢將通道連接至訊息資料來源的 IoT 規則 (iot: CreateTopicRule領域topicRulePayload:actions:iotAnalytics。當設備發送帶有適當主題簽證 MQTT 的消息時，它將被路由到您的頻道。或者，您可以使用iotanalytics: BatchPutMessage將消息直接從能夠使用AWS開發套件AWS CLI。
8. 建立由時間排程觸發的 SQL 資料集 (iotanalytics: CreateDataset,領域actions: queryAction:sqlQuery。

   您也可以指定要套用到訊息資料的預先篩選條件，以協助限制訊息為那些自上次執行動作後抵達的資料。(Field Setactions:queryAction:filters:deltaTime:timeExpression給出可以確定消息時間的運算式。而字段actions:queryAction:filters:deltaTime:offsetSeconds指定訊息到達時可能的延遲。)

   預過濾器和觸發器時間表決定了您的差異窗口。每個新的 SQL 資料集都是使用自上次建立 SQL 資料集後收到的訊息來建立。(第一次創建 SQL 數據集怎麼辦？ 根據排程和預先篩選器，估計上次建立資料集的時間。)
9. 創建由創建第一個觸發的另一個數據集 (CreateDataset領域trigger:dataset。對於此資料集，您可以指定容器動作 (已提交actions:containerAction) 指向您在第一步中創建的 Docker 容器，並提供運行所需的信息。您還可以指定：

   - 存儲在您帳戶中 docker 容器的 ARN (image。)
   - 提供系統存取所需資源之許可的角色 ARN，以便執行容器動作 (executionRoleArn)。
   - 執行容器動作的資源組態 (resourceConfiguration。)
   - 如果用於執行容器動作的運算資源類型 (computeType與可能的值：ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]。
   - 用於執行容器動作的資源執行個體可使用的持久性儲存大小 (GB) (volumeSizeInGB。

- 在應用程序執行的上下文中使用的變量的值（基本上，傳遞給應用程序的參數）（variables。

  這些變數會在執行容器時進行替換。這可讓您使用不同的變數 (參數) 來執行相同的容器，這些變數會在建立資料集內容時提供。所以此 IotAnalytics Jupyter 擴充功能會自動辨識筆記本中的變數，並將其作為容器化程序的一部分提供，簡化此程序。您可以選擇已是別的變數或新增您自己的自訂變數。在其執行容器之前，系統會在執行時以當時的值取代每一個這些變數。
- 其中一個變數是資料集的名稱，其最新內容會用作應用程式的輸入 (這是您在上一個步驟中建立的資料集名稱) (datasetContentVersionValue:datasetName。

使用 SQL 查詢和增量窗口生成數據集，並與您的應用程序一起生成容器，AWS IoT Analytics會建立排程的生產資料集，該資料集會依照您在差異視窗中指定的資料間隔執行，產生所需的輸出並傳送通知。

您可以暫停生產資料集應用程式，並在選擇時繼續執行。當您繼續生產資料集應用程式時，AWS IoT Analytics默認情況下，捕獲自上次執行以來已到達但尚未分析的所有數據。您還可以通過執行一系列連續運行來配置恢復生產數據集工作窗口長度的方式）。或者，您可以只擷取符合差異視窗指定大小的新到達資料，以恢復生產資料集應用程式。

建立或定義由建立另一個資料集而觸發的資料集時，請注意下列限制：

- SQL 資料集只能觸發容器資料集。
- SQL 資料集最多可觸發 10 個容器資料集。

建立由 SQL 資料集觸發的容器資料集時，可能會傳回下列錯誤：

- 「觸發資料集只能在容器資料集新增」
- 「只能有一個觸發資料集」

  如果您嘗試定義由兩個不同 SQL 資料集觸發的容器資料集，就會發生這個錯誤。
- 「觸發數據集<dataset-name>不能由容器數據集觸發」

  如果您嘗試定義另一個容器資料集觸發的容器資料集，就會發生這個錯誤。
- 「<N>資料集已經依賴於<dataset-name>資料集。」

  如果您嘗試定義另一個容器資料集由已觸發 10 個容器資料集的 SQL 資料集觸發的容器資料集，就會發生這個錯誤。
- 「應該只提供一個觸發類型」

當您嘗試定義由排程觸發程序和資料集觸發程序所觸發的資料集時，就會發生這個錯誤。

自定義Docker 容器輸入/輸出變量

本章節示範透過您的自訂 Docker 影像執行的程式，如何讀取輸入變數和上傳其輸出。

程式檔案

輸入變數和您要上傳輸出的目的地都儲存在 JSON 檔案中，此檔案位在執行您的 Docker 影像的執行個體上的 `/opt/ml/input/data/iotanalytics/params`。以下為該文件內容範例：

```
{
   "Context": {
       "OutputUris": {
           "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.html",
           "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.ipynb"
       }
   },
   "Variables": {
       "source_dataset_name": "mydataset",
       "source_dataset_version_id": "xxxx",
       "example_var": "hello world!",
       "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.txt"
   }
}
```

除了您的資料集的名稱和版本 ID 外，`Variables` 區段還包含在 `iotanalytics:CreateDataset` 叫用中指定的變數，在此範例中，變數 `example_var` 取得了值 `hello world!`。`custom_output` 變數中也提供自訂輸出 URI。`OutputUris` 欄位包含容器可以上傳其輸出的預設位置，在這個範例中，對於 ipynb 和 html 輸出都會提供預設輸出 URI。

輸入變數

由您的 Docker 影像啟動的程式可以讀取來自 `params` 檔案的變數。這是回應範例：`params`文件，解析它，並打印`example_var`變數。

```
import json
```

```
with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]
print(example_var)
```

上傳輸

由 Docker 映像啟動的程式也可能會將其輸出存放在 Amazon S3 位置。輸出必須載入」bucket-owner-full-control「存取控制清單。存取清單會授予AWS IoT Analytics對上傳輸出的服務控制。在這個例子中，我們擴展了前一個上傳的內容`example_var`到由下列項目定義的 Amazon S3 位置`custom_output`在`params`檔案。

```
import boto3
import json
from urllib.parse import urlparse

ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]

outputUri = params["Variables"]["custom_output"]
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

許可

您必須建立兩個 角色。一個角色授予啟動權限 SageMaker 實例以容器化筆記本。而執行容器需要另一個角色。

您可以自動或手動建立第一個角色。如果您建立新的 SageMaker執行個體與AWS IoT Analytics控制台，您可以選擇自動創建一個新角色，該角色授予執行所需的所有權限 SageMaker 執行個體和容器化筆記本。或者，您可以手動使用這些權限建立角色。若要執行此操作，請使用`AmazonSageMakerFullAccess`附加政策，並新增以下政策。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:BatchGetImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DescribeRepositories",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::iotanalytics-notebook-containers/*"
    }
  ]
}
```

您必須手動建立第二個角色，授予執行容器的許可。即使您使用了AWS IoT Analytics控制台自動創建第一個角色。建立連接下列政策和信任政策的角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::aws-*-dataset-*/*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "iotanalytics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

信任政策範例如下。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": ["sagemaker.amazonaws.com", "iotanalytics.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

使用 CreateDataset 通過 Java 和AWS CLI

建立資料集。透過套用資料集，儲存從資料存放區擷取的資料queryAction（一個 SQL 查詢）或containerAction（執行容器化應用程序）。此作業會建立資料集的架構。可以通過調用手動填充數據集CreateDatasetContent或根據自動trigger您指定。如需詳細資訊，請參閱《》CreateDataset和CreateDatasetContent。

主題

- 實施例 1-創建一個 SQL 數據集（Java）
- 示例 2-使用增量窗口（java）創建 SQL 數據集
- 範例 3 — 使用自己的排程觸發程序建立容器資料集 (java)
- 範例 4 — 使用 SQL 資料集作為觸發程序建立容器資料集 (java)
- 範例 5 — 建立 SQL 資料集 (CLI)
- 範例 6 — 使用差異時段（CLI）建立 SQL 資料集

實施例 1-創建一個 SQL 數據集（Java）

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
 DataStoreName"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
```

```
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

成功輸出：

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

示例 2-使用增量窗口（java）創建 SQL 數據集

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Filter for DeltaTime
QueryFilter deltaTimeFilter = new QueryFilter();
deltaTimeFilter.withDeltaTime(
                new DeltaTime()
                .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)
                .withTimeExpression("from_unixtime(timestamp)"));

//Create Action
action.setActionName("SQLActionWithDeltaTime");
action.setQueryAction(new SqlQueryDatasetAction()
                .withSqlQuery("SELECT * from DataStoreName")
                .withFilters(deltaTimeFilter));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);
```

```
//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

成功輸出：

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

範例 3 — 使用自己的排程觸發程序建立容器資料集 (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);
```

```
//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

成功輸出：

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

範例 4 — 使用 SQL 資料集作為觸發程序建立容器資料集 (java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
```

```
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
        .withDataset(new TriggeringDataset()
                .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

成功輸出：

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>}
```

範例 5 — 建立 SQL 資料集 (CLI)

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --dataset-
name="<dataSetName>" --actions="[{\"actionName\":\"<ActionName>\", \"queryAction\":
{\"sqlQuery\":\"<SQLQuery>\"}}]" --retentionPeriod numberOfDays=10
```

成功輸出：

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
    "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

範例 6 — 使用差異時段（CLI）建立 SQL 資料集

差異視窗是一系列使用者定義的、非重疊和連續的時間間隔。差異時段可讓您使用建立資料集內容，並對自上次分析以來已到達資料存放區的新資料執行分析。您可以透過設定建立差異視窗`deltaTime`在`filters`的一部分`queryAction`資料集的 (CreateDataset。通常，您還希望通過設置時間間隔觸發器來自動創建數據集內容（`triggers:schedule:expression`。基本上，這使您可以過濾在特定時間段內到達的消息，因此來自上一個時間窗口的消息中包含的數據不會被計算兩次。

在這個範例中，我們建立一個新的資料集，每 15 分鐘就會使用上次到達的資料，自動建立新的資料集內容。我們指定 3 分鐘 (180 秒) deltaTime 偏移，允許訊息延遲 3 分鐘，送達指定的資料存放區。因此，如果資料集內容是在上午 10:30 建立的，則使用的資料 (包含在資料集內容中) 就是時間戳記介於上午 10:12 和 10:27 AM 之間 (即上午 10:30-15 分鐘-3 分鐘-上午 10 點 30 分鐘-3 分鐘)。

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --cli-input-
json file://delta-window.json
```

文件的位置delta-window.json包含以下項目。

```
{
  "datasetName": "delta_window_example",
  "actions": [
    {
      "actionName": "delta_window_action",
      "queryAction": {
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",
        "filters": [
          {
            "deltaTime": {
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(timestamp)"
            }
          }
        ]
      }
    }
  ],
  "triggers": [
    {
      "schedule": {
        "expression": "cron(0/15 * * * ? *)"
      }
    }
  ]
}
```

成功輸出：

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
```

```
}
```

容器化筆記本

本節包含如何使用 Jupyter 記事本建立 Docker 容器的相關資訊。如果您重複使用第三方建置的筆記本，有安全風險：包含的容器可以使用您的使用者許可來執行任意程式碼。此外，筆記本產生的 HTML 可以顯示在AWS IoT Analytics主控台，在顯示 HTML 的電腦上提供潛在的攻擊媒介。在使用任何第三方筆記本之前，請確定您信任其撰寫者。

執行進階分析功能的一個選項是使用 Jupyter 筆記本。Jupyter Notebook 提供了強大的數據科學工具，可以執行機器學習和一系列統計分析。如需詳細資訊，請參閱《》筆記本。（請注意，我們目前不支持內部容器化 JupyterLab。）您可以將 Jupyter 筆記本和庫打包到一個容器中，該容器在接收到的新批次數據上定期運行AWS IoT Analytics在您定義的差異時間範圍內。您可以排定使用容器的分析工作，以及在指定時間範圍內擷取的新分段資料，然後儲存工作的輸出，以供日 future 排定的分析使用。

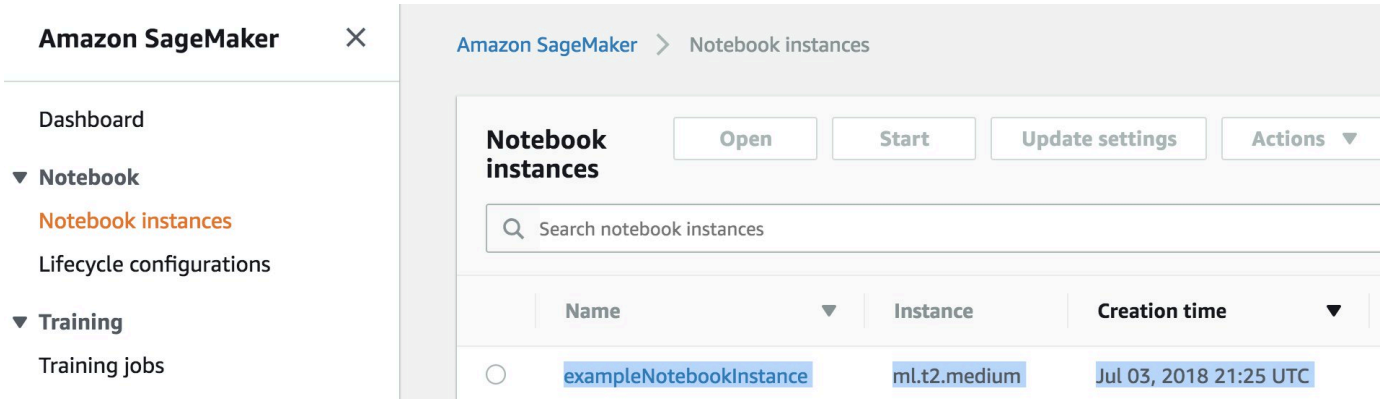
如果您已建立 SageMaker 執行個體使用AWS IoT Analytics控制台在 2018 年 8 月 23 日之後，容器化擴展的安裝已自動為您完成你可以開始創建一個容器化的圖像。否則，請按照本節中列出的步驟操作，啟用筆記型電腦容器化 SageMaker 實例。在以下內容中，您修改了 SageMaker 執行角色可讓您將容器映像上傳到 Amazon EC2，並安裝容器化擴充功能。

啟用未透過以下方式建立的筆記本執行個體的容器化AWS IoT Analytics安慰

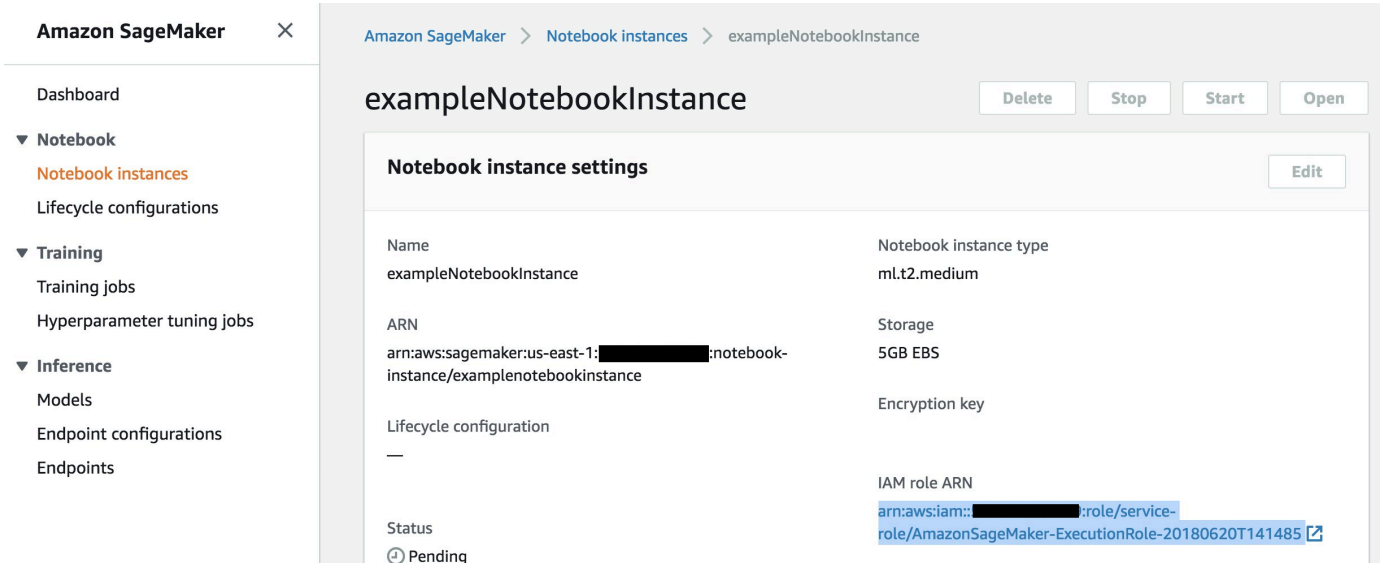
建議您建立新 SageMaker 執行個體AWS IoT Analytics控制台，而不是遵循這些步驟。新的執行個體自動支援容器化。

如果您重新啟動 SageMaker 執行個體在如下所示啟用容器化之後，您不必重新新增 IAM 角色和政策，但您必須重新安裝擴充功能，如最後一個步驟所示。

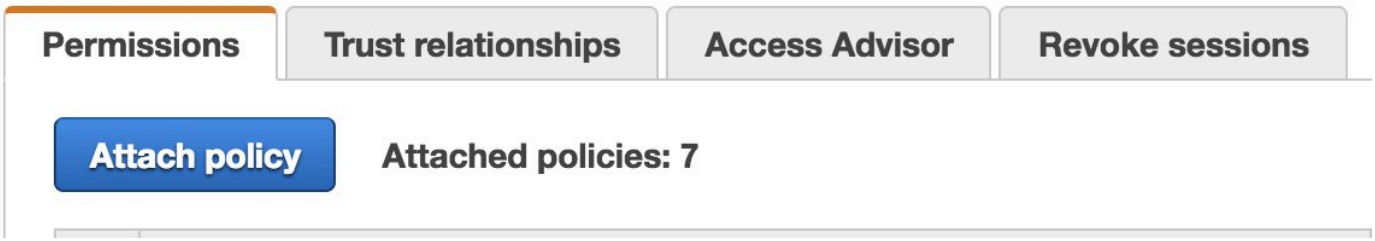
1. 若要授與您的筆記型電腦執行個體存取 Amazon ECS，請選取您的 SageMaker 執行個體上的 SageMaker 頁面：



2. 下IAM 角色，選擇 SageMaker 執行角色。

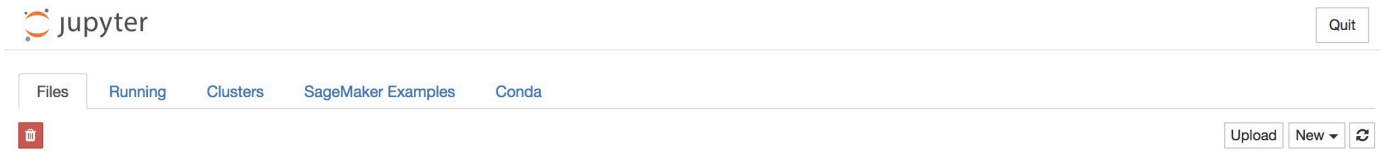


3. 選擇 Attach Policy (連接政策)，然後定義並連接 Permissions (許可)中所顯示的政策。如果AmazonSageMakerFullAccess策略尚未附加，也請附加它。



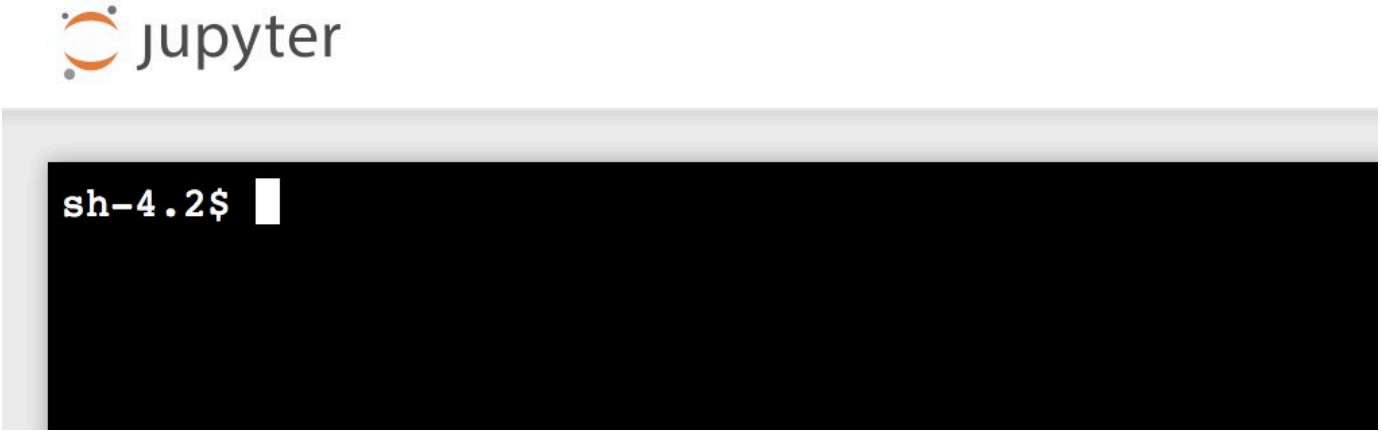
您也必須從 Amazon S3 下載容器化程式碼，並將其安裝在筆記本執行個體上，第一步是存取 SageMaker 執行個體的終端。

1. 在木皮特內，選擇全新。



2. 從出現的選單中，選擇航站。

3. 在終端機內，輸入以下命令來下載程式碼，將其解壓縮並進行安裝。請注意，這些命令會殺死您的筆記本電腦正在運行的任何進程 SageMaker 實例。




```
cd /tmp

aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp

unzip iota_notebook_containers.zip

cd iota_notebook_containers

chmod u+x install.sh

./install.sh
```

等待一兩分鐘以進行驗證和安裝延伸。

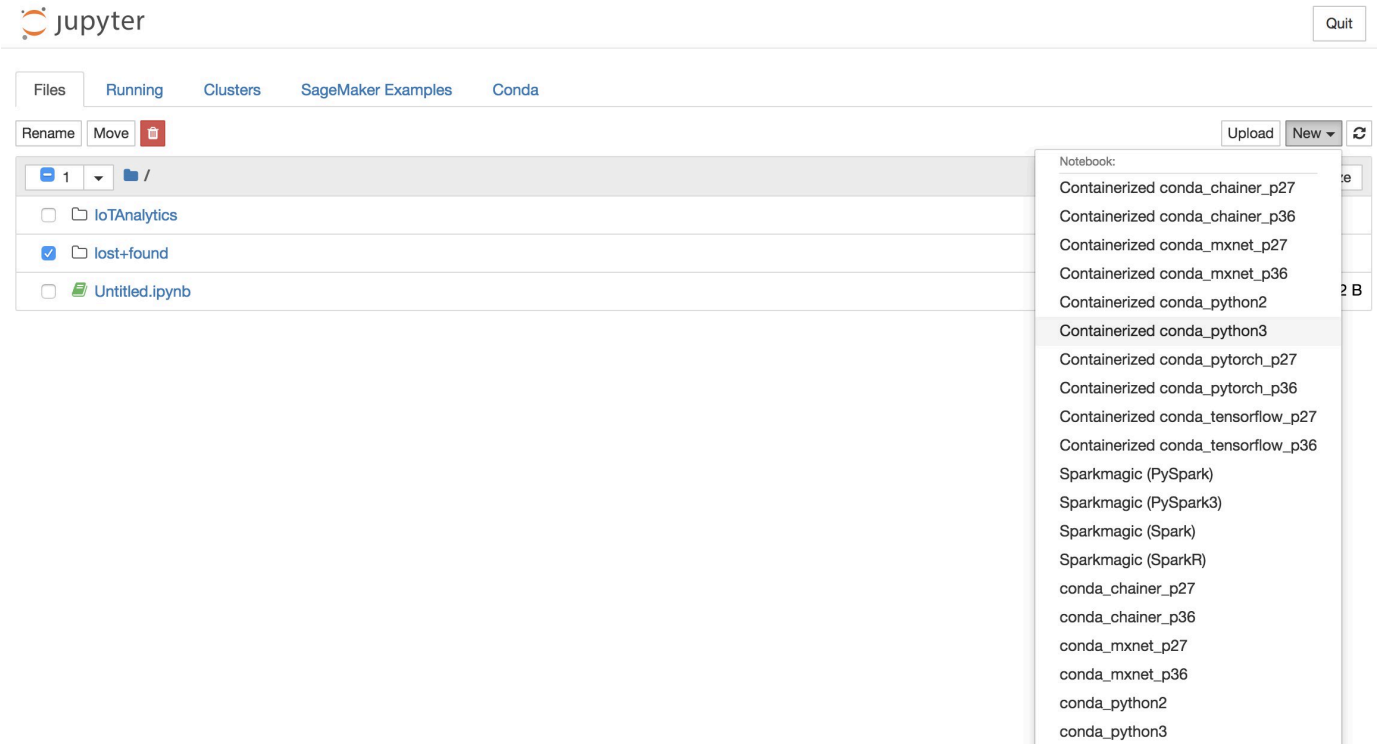
更新您的筆記型電腦容器化擴充功能

如果您建立了 SageMaker 透過執行個體AWS IoT Analytics控制台在 2018 年 8 月 23 日之後，然後自動安裝容器化擴充功能。您可以通過重新啟動實例來更新擴展程序 SageMaker 主控台。如果您手動安裝了擴展程序，則可以通過重新運行「啟用未創建的筆記本實例的容器化」中列出的終端命令來更新擴展程序AWS IoT Analytics主控台。

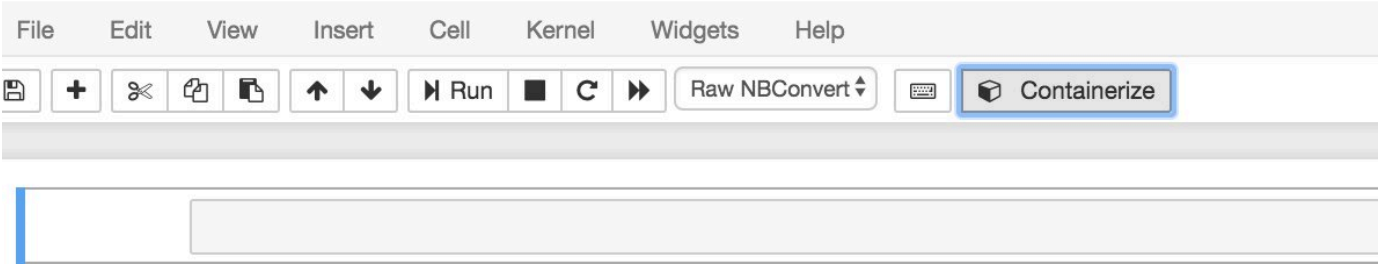
建立容器化映像

在本章節中，我們示範容器化 筆記本所需的步驟。若要開始，請移至您的 Jupyter 筆記本，使用容器化核心建立筆記本。

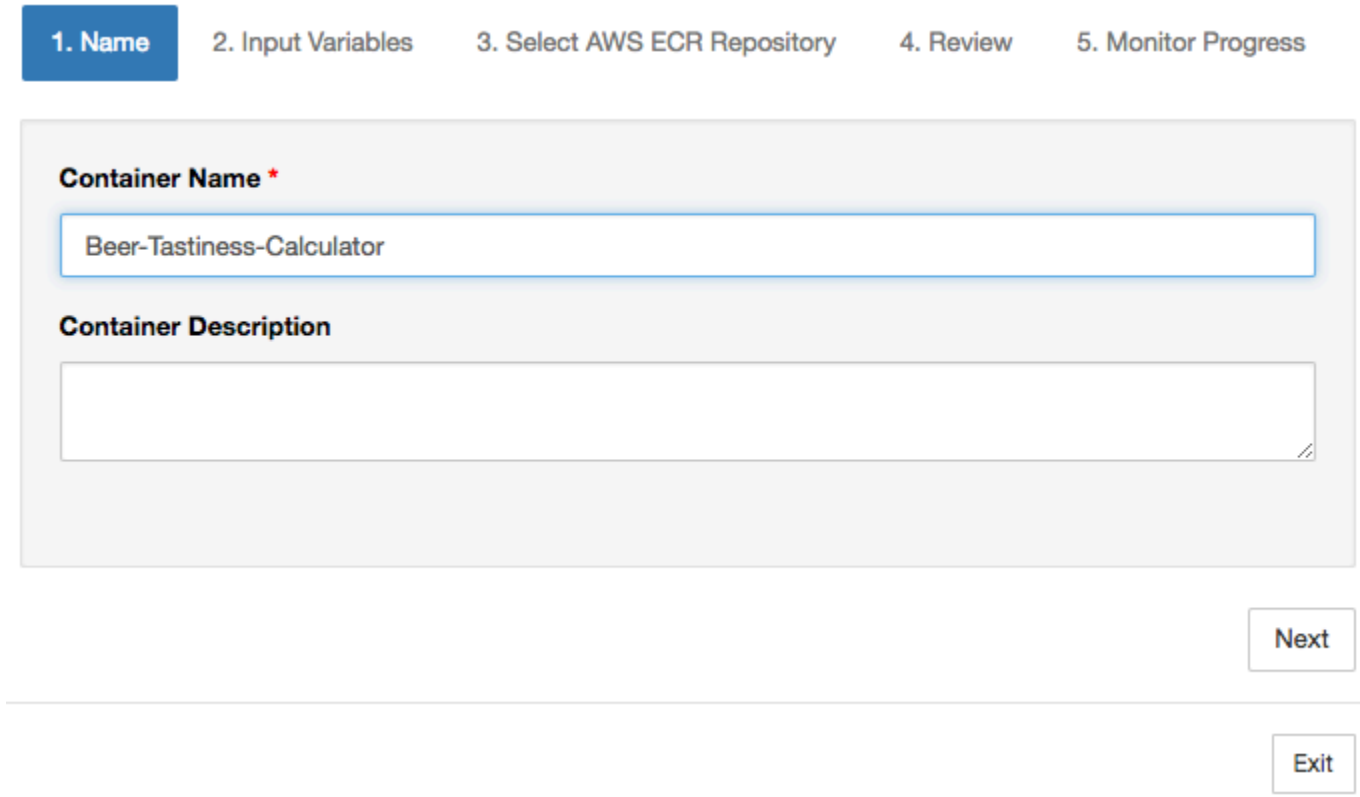
1. 在您的 Jupyter 筆記本中，選擇 New (新增)，然後從下拉式清單選擇您想要的核心類型。（內核類型應以「容器化」開頭，並以您原本選擇的任何內核結束。 例如，如果您只想要一個普通的 Python 3.0 環境，如「conda_python3」，請選擇「容器化的 conda_python3」）。



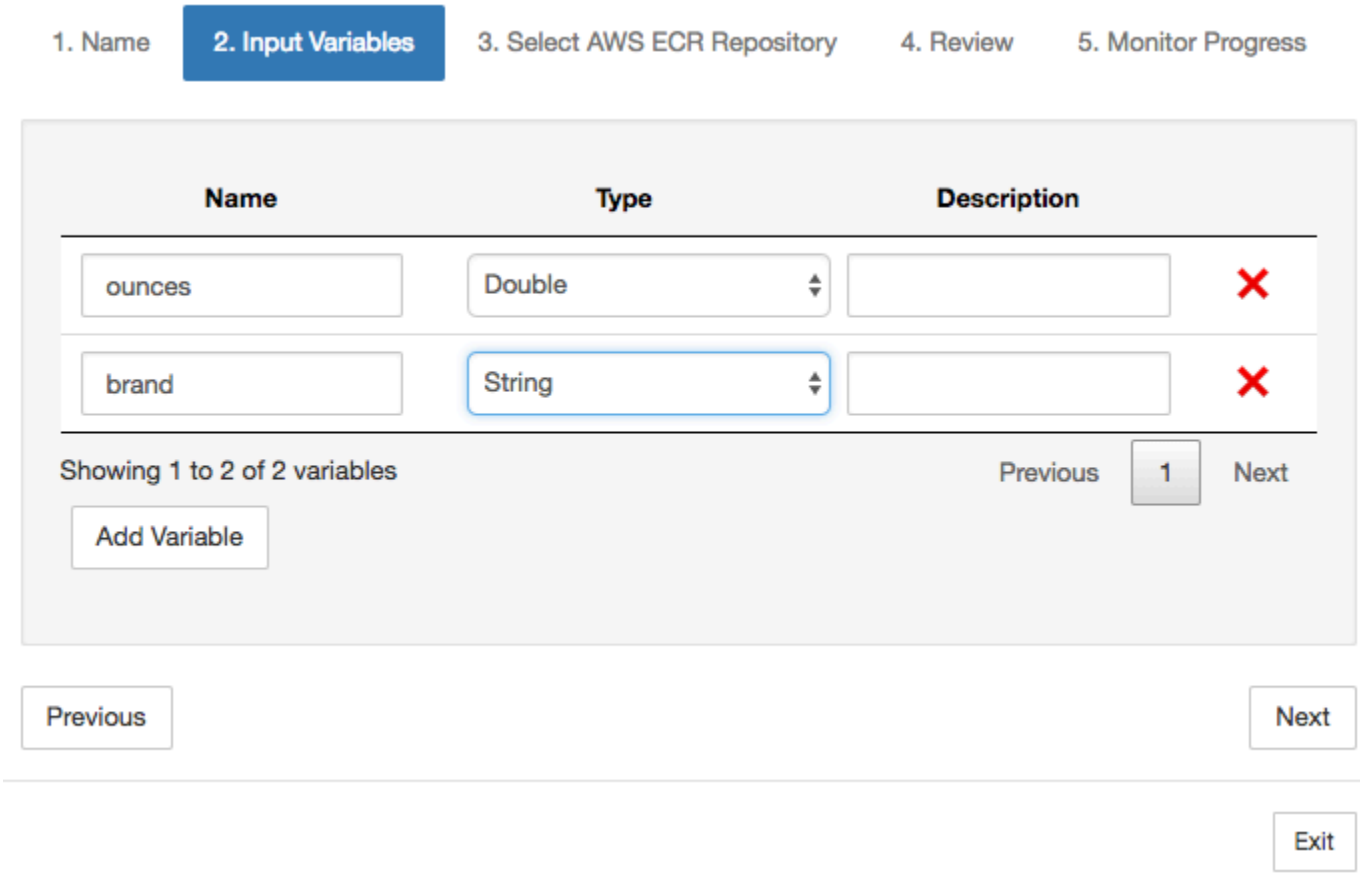
2. 在您完成筆記型電腦的工作並想要將其容器化之後，請選擇容器化。



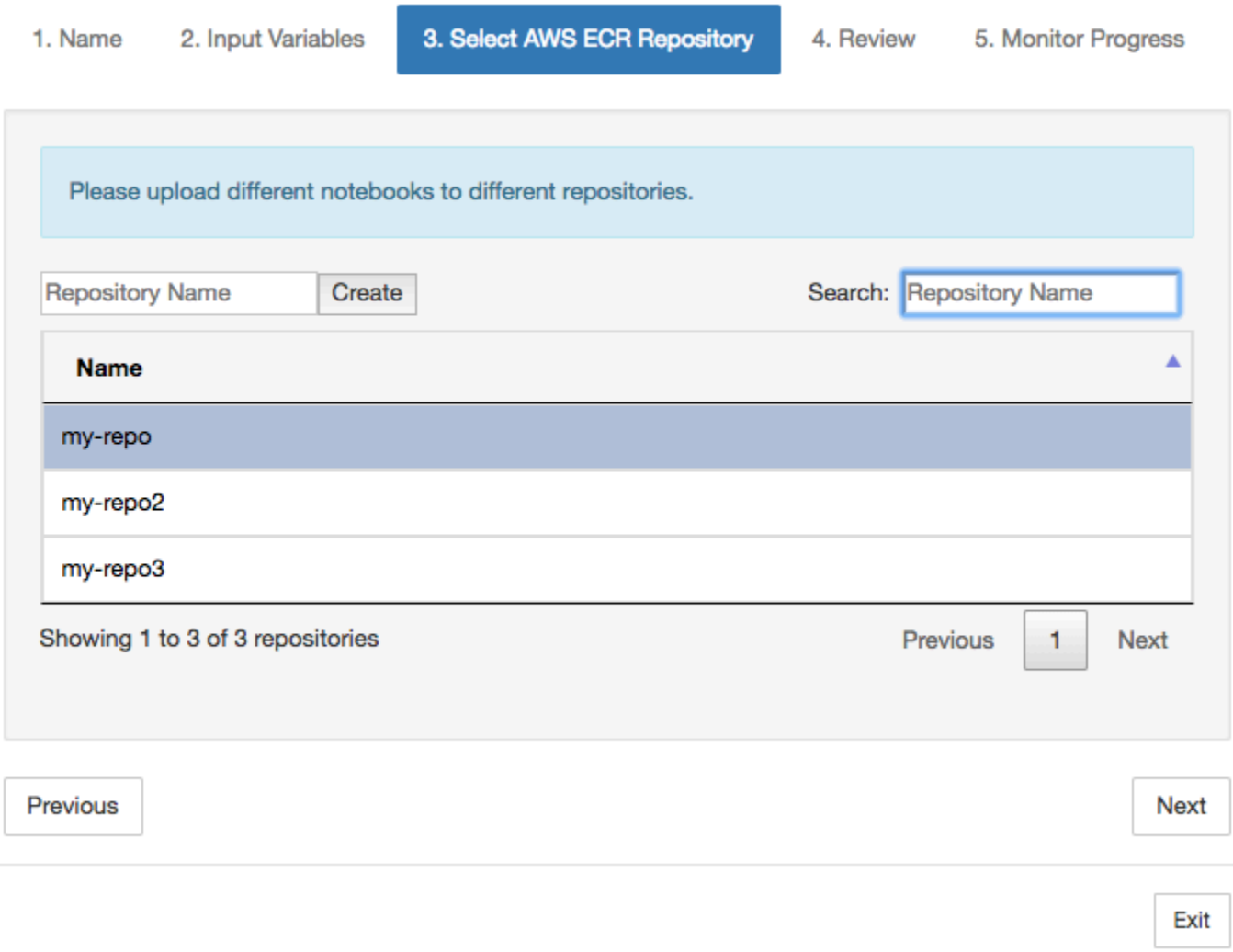
3. 輸入容器化筆記本的名稱。您也可以輸入選用說明。



4. 指定您的筆記本應叫用的 Input Variables (輸入變數) (參數)。您可以選擇自動從您的筆記本偵測到的輸入變數，或定義自訂變數。(請注意，如果您有之前已執行的筆記本，則只會偵測到輸入變數)。對於每個輸入變數選擇類型。您也可以輸入輸入變數的選擇性描述。

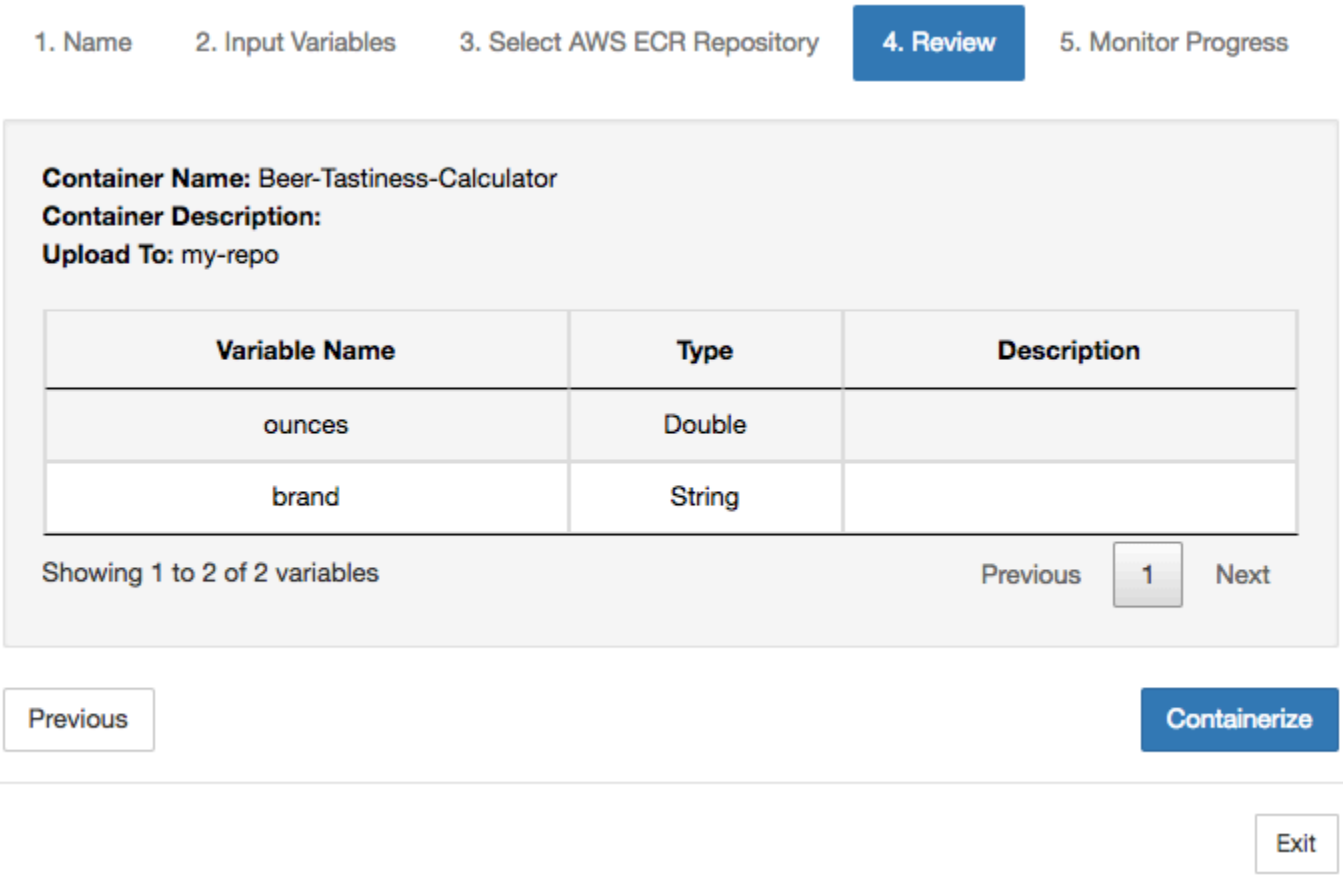


5. 選擇應將從筆記本建立的映像上傳到的 Amazon ECR 儲存庫。

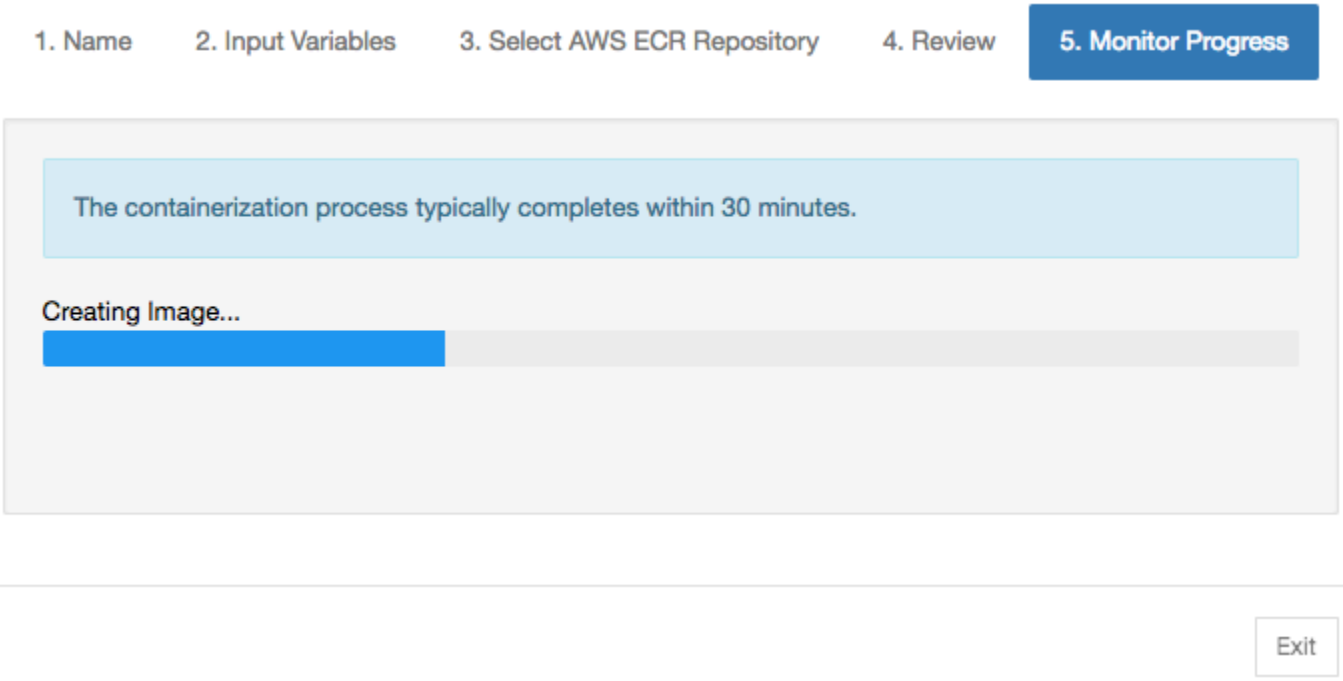


6. 選擇容器化開始該過程。

   您會看到一個概述，總結您的輸入。請注意，啟動該過程之後，您便無法將其取消。該過程可能持續長達一個小時。

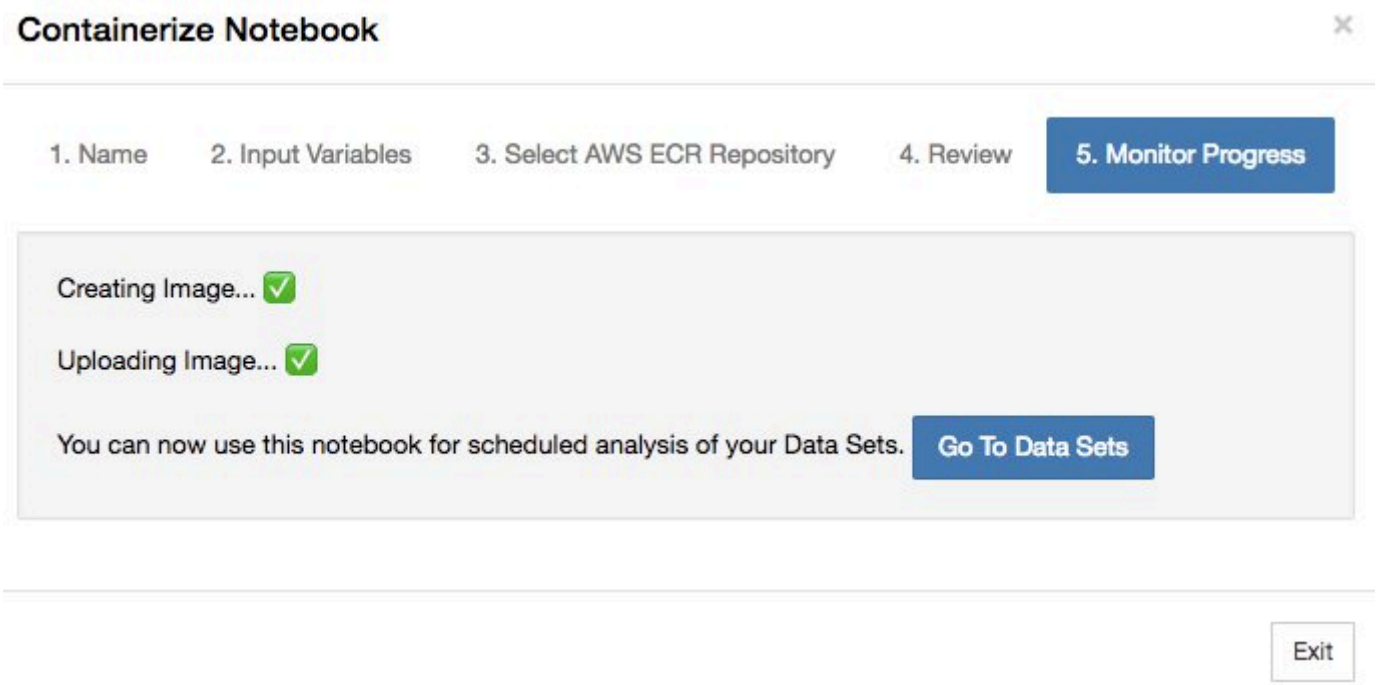


7. 下一頁顯示進度。



8. 如果您不小心關閉瀏覽器，您可以從筆記本的區段AWS IoT Analytics主控台。

9. 程序完成後，容器化映像會儲存在 Amazon ECR 上，供您使用。



使用自訂容器進行分析

本節包含如何使用 Jupyter 記事本建立 Docker 容器的相關資訊。如果您重複使用第三方建置的筆記本，有安全風險：包含的容器可以使用您的使用者許可來執行任意程式碼。此外，筆記本產生的 HTML 可以顯示在AWS IoT Analytics主控台，在顯示 HTML 的電腦上提供潛在的攻擊媒介。在使用任何第三方筆記本之前，請確定您信任其撰寫者。

您可以建立自己的自訂容器，並使用AWS IoT Analytics服務。為此，您可以設置 Docker 映像並將其上傳到 Amazon ECR，然後設置數據集以運行容器操作。本章節提供使用 Octave 的程序範例。

此教學課程假設您擁有：

- 安裝在本機電腦的 Octave
- 在本機電腦設定 Docker 帳戶
- 同時AWS使用 Amazon ECR 帳戶AWS IoT Analytics存取

步驟 1：設定 Docker 映像

在此教學課程中您需要三個主要檔案。其名稱和內容在此：

- `Dockerfile`— Docker 容器化程序的初始設定。

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- `run-octave.py`— 剖析 JSON 來源AWS IoT Analytics，執行八度指令碼，然後將成品上傳到 Amazon S3。

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)

variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
```

```
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
 local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
 'rb'), ACL='bucket-owner-full-control')
```

- moment— 一個簡單的八度腳本，它根據輸入或輸出文件和指定的順序計算時刻。

```
#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename,'M')
```

1. 下載每個檔案的內容。創建一個新目錄並將所有文件放在其中，然後cd到該目錄。
2. 執行下列命令。

```
docker build -t octave-moment .
```

3. 您應應在 Docker 儲存庫中看到新映像。執行下列命令來驗證。

```
docker image ls | grep octave-moment
```

步驟 2：上傳 Docker 映像至 Amazon ECR 儲存庫

1. 在亞馬遜 ECR 中創建一個存儲庫。

```
aws ecr create-repository --repository-name octave-moment
```

2. 獲取登錄到您的碼頭環境。

```
aws ecr get-login
```

3. 複製輸出並執行它。輸出應類似以下所示。

```
docker login -u AWS -p password -e none https://your-aws-account-
id.dkr.ecr..amazonaws.com
```

4. 使用 Amazon ECR 儲存庫標籤為您建立的映像加上標籤。

```
docker tag your-image-id  your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-
moment
```

5. 將映像推送至 Amazon ECR。

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

步驟 3：將範例資料上傳至 Amazon S3 儲存貯體

1. 下載以下文件到文件input.txt。

```
0.857549  -0.987565  -0.467288  -0.252233  -2.298007
 0.030077  -1.243324  -0.692745   0.563276   0.772901
-0.508862  -0.404303  -1.363477  -1.812281  -0.296744
-0.203897   0.746533   0.048276   0.075284   0.125395
 0.829358   1.246402  -1.310275  -2.737117   0.024629
 1.206120   0.895101   1.075549   1.897416   1.383577
```

2. 建立名為的 Amazon S3 儲存貯體octave-sample-data-*your-aws-account-id*。
3. 上傳檔案input.txt到您剛建立的 Amazon S3 儲存貯體。現在您應已擁有名為的儲存貯體octave-sample-data-*your-aws-account-id*包含input.txt檔案。

步驟 4：建立容器執行角色

1. 將下列項目複製到名為的檔案`role1.json`。取代*`your-aws-account-id`*與您的AWS帳戶ID*`aws-region`*與AWS您的地區AWS資源。

> **Note**
>
> 此範例包含全域條件內容索引鍵，以防範例：如需詳細資訊，請參閱 the section called “預防跨服務混淆代理人”。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "iotanalytics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
              "StringEquals": {
                "aws:SourceAccount": "your-aws-account-id"
              },
              "ArnLike": {
                "aws:SourceArn": "arn:aws:iotanalytics:aws-region:your-aws-account-
id:dataset/DOC-EXAMPLE-DATASET"
              }
            }
    ]
}
```

2. 建立授與存取權限的角色 SageMaker 和AWS IoT Analytics，使用檔案`role1.json`您下載的。

```
aws iam create-role --role-name container-execution-role --assume-role-policy-
document file://role1.json
```

3. 將以下內容下載到名為的文件`policy1.json`並取代*`your-account-id`*使用您的帳戶 ID（請參閱下面的第二個 ARN`Statement:Resource`。

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:GetBucketLocation",
       "s3:PutObject",
       "s3:GetObject",
       "s3:PutObjectAcl"
     ],
     "Resource": [
       "arn:aws:s3:::*-dataset-*/*",
       "arn:aws:s3:::octave-sample-data-your-account-id/*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "iotanalytics:*"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "ecr:GetAuthorizationToken",
       "ecr:GetDownloadUrlForLayer",
       "ecr:BatchGetImage",
       "ecr:BatchCheckLayerAvailability",
       "logs:CreateLogGroup",
       "logs:CreateLogStream",
       "logs:DescribeLogStreams",
       "logs:GetLogEvents",
       "logs:PutLogEvents"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "s3:GetBucketLocation",
       "s3:ListBucket",
       "s3:ListAllMyBuckets"
```

```
      ],
      "Resource" : "*"
    }
  ]
}
```

4. 使用建立 IAM 政策policy.json您剛剛下載的文件。

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
 file://policy1.json
```

5. 將 政策連接到角色。

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
 arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

步驟 5：使用容器動作建立資料集

1. 將以下內容下載到名為 fiecli-input.json並替換所有實例*your-account-id*和*region*具有適當的值。

```
{
    "datasetName": "octave_dataset",
    "actions": [
        {
            "actionName": "octave",
            "containerAction": {
                "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",
                "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
                "resourceConfiguration": {
                    "computeType": "ACU_1",
                    "volumeSizeInGB": 1
                },
                "variables": [
                    {
                        "name": "octaveResultS3URI",
                        "outputFileUriValue": {
                            "fileName": "output.mat"
                        }
                    },
```

```
                    {
                        "name": "inputDataS3BucketName",
                        "stringValue": "octave-sample-data-your-account-id"
                    },
                    {
                        "name": "inputDataS3Key",
                        "stringValue": "input.txt"
                    },
                    {
                        "name": "order",
                        "stringValue": "3"
                    }
                ]
            }
        }
    ]
}
```

2. 使用檔案建立資料集cli-input.json您剛剛下載並編輯。

```
aws iotanalytics create-dataset –cli-input-json file://cli-input.json
```

步驟 6：叫用資料集內容產生

1. 執行下列命令。

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

步驟 7：取得資料集

1. 執行下列命令。

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \
$LATEST
```

2. 您可能需要等待幾分鐘，直到DatasetContentState是SUCCEEDED。

步驟 8：在八度上打印輸出

1. 使用八度外殼通過運行以下命令打印從容器的輸出。

```
bash> octave
octave> load output.mat
octave> disp(M)
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```

視覺化AWS IoT Analytics資料

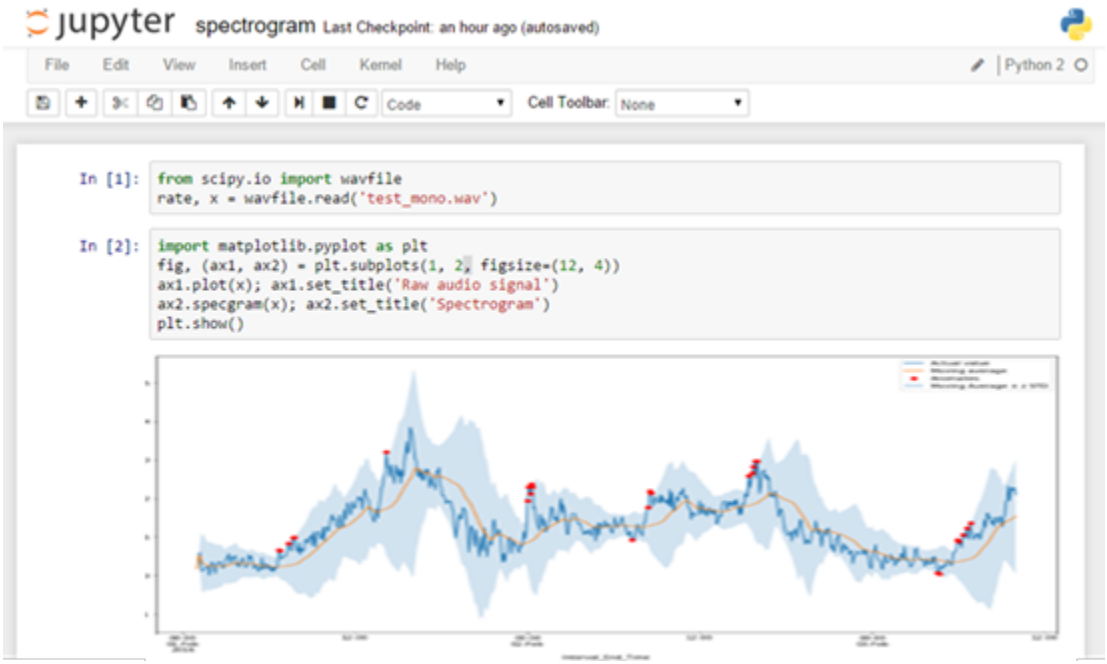
若要視覺化您的AWS IoT Analytics資料，您可以使用AWS IoT Analytics主控台或 Amazon QuickSight。

主題

- 視覺化AWS IoT Analytics數據與控制台
- 視覺化AWS IoT AnalyticsAmazon QuickSight 的資料

視覺化AWS IoT Analytics數據與控制台

AWS IoT Analytics可以嵌入容器數據集的 HTML 輸出（在`output.html`）的容器數據集內容頁面上的AWS IoT Analytics安慰。例如，如果您定義一個運行 Jupyter 筆記本的容器資料集，並在 Jupyter 筆記本中建立視覺化，則您的資料可能會如下所示。



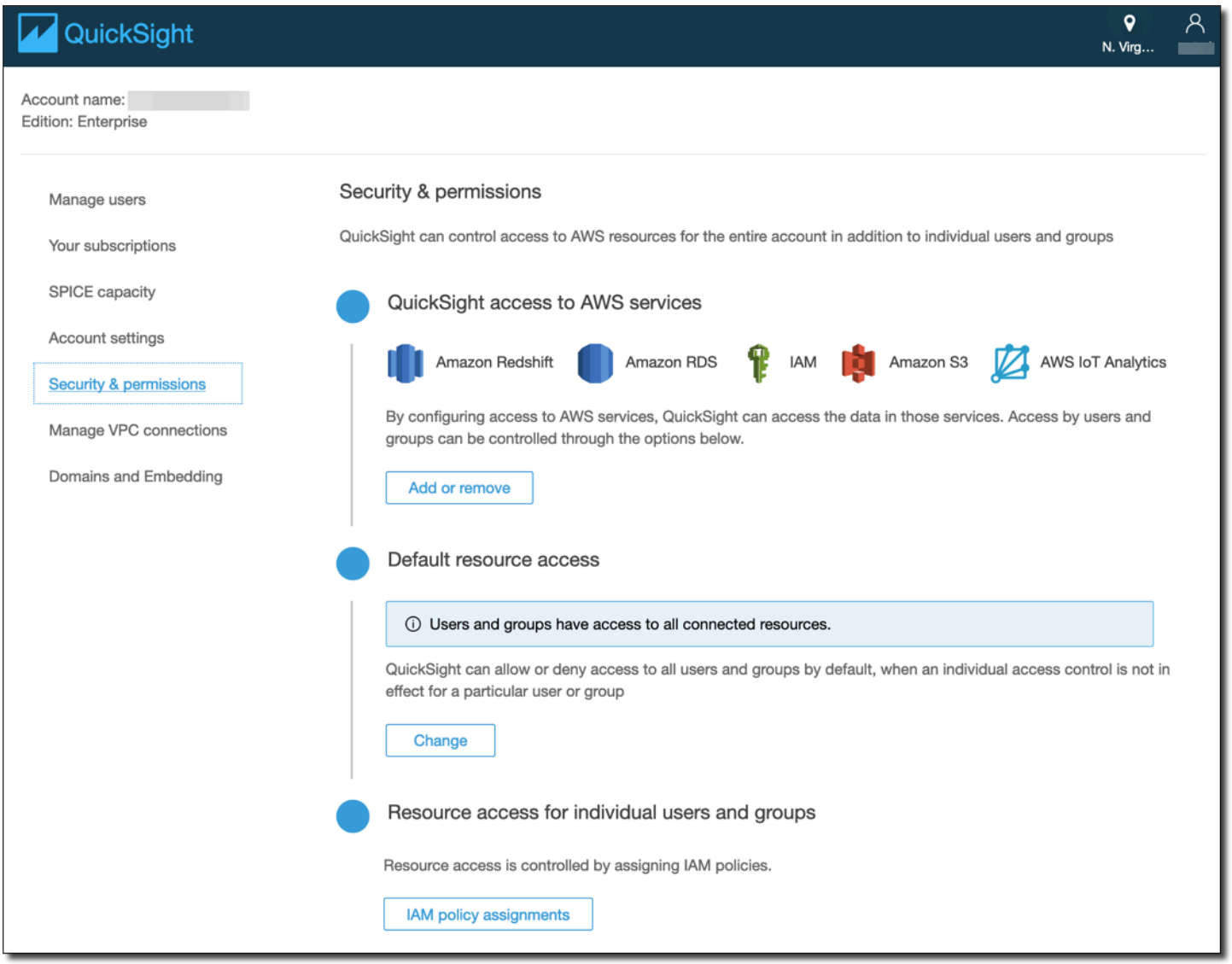
接著在建立容器資料集內容後，您就可以在主控台的資料集內容頁面。

有關建立運行 Jupyter 筆記本的容器資料集的信息，請參閱自動化您的工作流程。

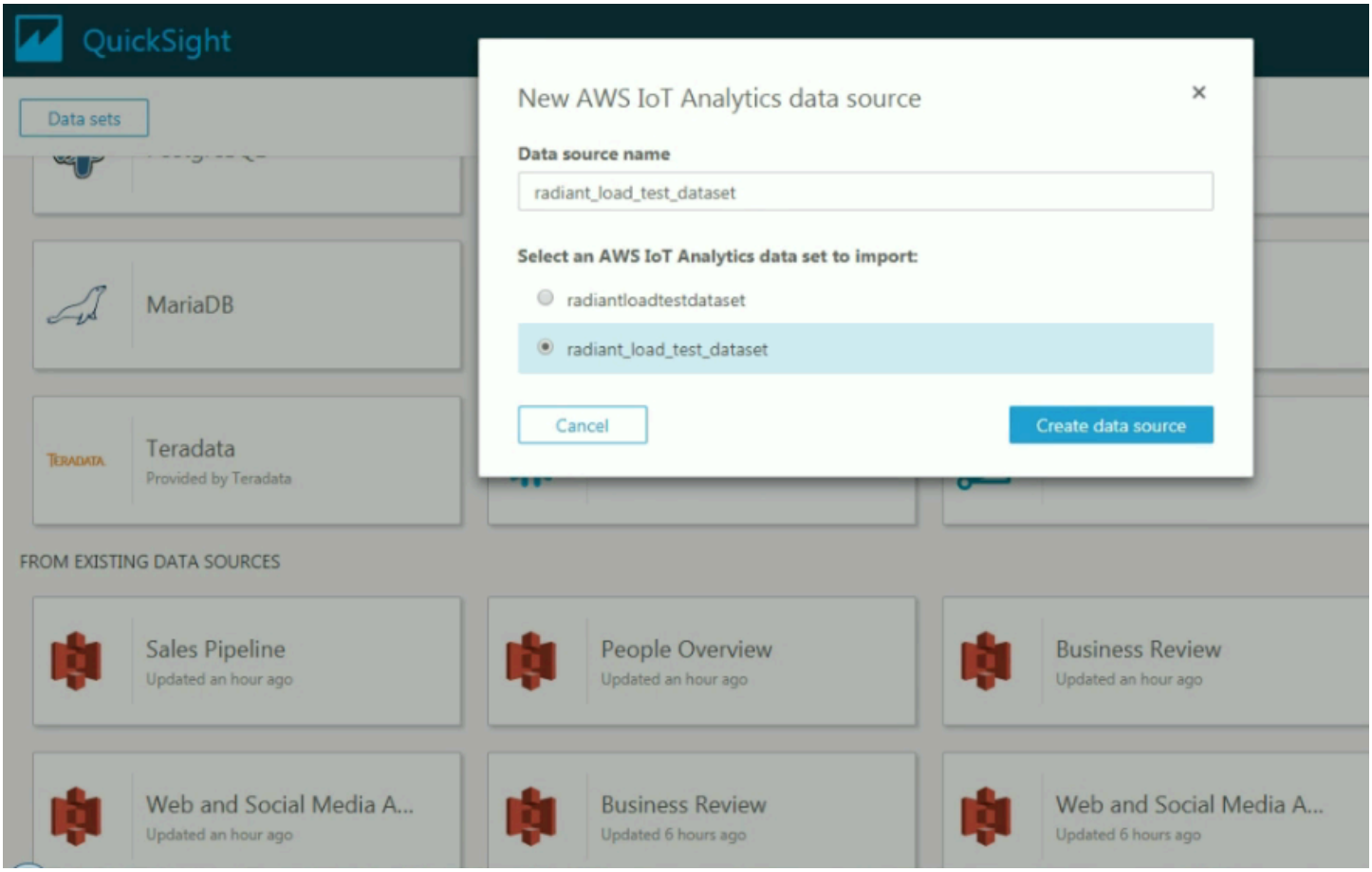
視覺化AWS IoT AnalyticsAmazon QuickSight 的資料

AWS IoT Analytics提供直接集成Amazon QuickSight。亞馬遜 QuickSight 是一項快速的商業分析服務，可用來建置視覺化效果、執行臨機操作分析，及快速從資料獲取商業見解。亞馬遜 QuickSight 讓組織能夠擴展到數十萬使用者，並透過使用強大的記憶體內引擎 (SPICE) 提供靈敏的效能。您可以選擇您的AWS IoT Analytics亞馬遜中的數據集 QuickSight 控制台並開始創建儀錶板和可視化效果。亞馬遜 QuickSight 可在 中使用。這些地區。

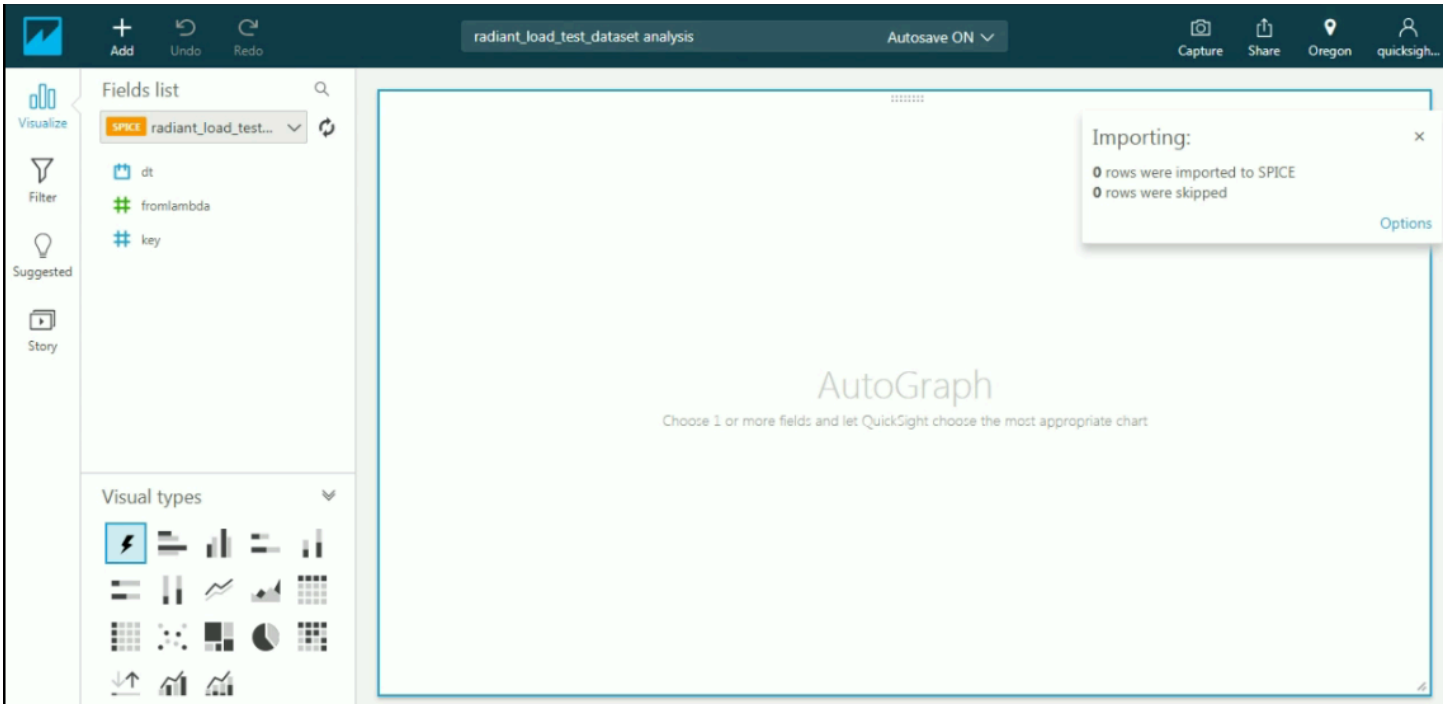
開始使用您的亞馬遜 QuickSight 可視化效果，您必須創建一個亞馬遜 QuickSight 帳戶。確保您向亞馬遜提供 QuickSight 存取您的AWS IoT Analytics當您設定帳戶時的資料。如果您已經擁有帳戶，請將 QuickSight 存取您的AWS IoT Analytics數據，方法是選擇管理員、QuickSight、安全性和權限。UdSightQuickSight 存取AWS服務，選擇新增或移除，接著選取AWS IoT Analytics並選擇更新。



當您的帳戶建立好後，來自 Amazon 的管理員 QuickSight 選擇主控台頁面新增分析和新的資料集，然後選擇AWS IoT Analytics作為來源。輸入您的資料來源名稱，選擇要導入的資料集，然後選擇建立資料來源。



當資料來源建立好後，您就可以在 Amazon QuickSight 中建立視覺化效果。



有關亞馬遜的信息 QuickSight 儀錶板和數據集的詳細信息，請參閱亞馬遜 QuickSight 文件。

標記您的 AWS IoT Analytics 資源

為協助您管理您的頻道、資料集、資料存放區及管道，您可以選擇性將您自己的中繼資料，以標籤的形式指派給這些資源。本章說明標籤並示範如何建立此類標籤。

主題

標籤基本概念

標籤可讓您以不同的方式分類您的 AWS IoT Analytics 資源，例如依據目的、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為您可以依據先前指派的標籤，快速識別特定的資源。每個標籤皆包含由您定義的一個「索引鍵」與選擇性的「值」。例如，您可以為您的頻道定義一組標籤，協助您追蹤負責每個頻道訊息來源的裝置類型。我們建議您為每種資源類型建立符合您需求的標籤金鑰。使用一致的標籤金鑰組可讓您更輕鬆的管理您的資源。您可以根據您新增的標籤搜尋和篩選資源。

您也可以使用標籤來分類和追蹤您的成本。當您將標籤套用至通道、資料集、資料存放區或管道，AWS會以逗號分隔值 (CSV) 檔案格式產生一份成本分配報告，其中包含依標籤彙總的用量與成本。您可以套用代表業務類別 (例如成本中心、應用程式名稱或擁有者) 的標籤，來整理多個服務中的成本。如需有關使用成本分配的詳細資訊，請參閱使用AWS Billing者指南中的使用成本分配標籤。

為了易用性，請使用AWS Billing and Cost Management主控台中的標籤編輯器，此編輯器可讓您集中、整合地建立和管理標籤。如需詳細資訊，請參閱入門中的使用標籤編輯器AWS Management Console。

您還可使用 AWS CLI 和 AWS IoT Analytics API 來使用標籤。當您建立標籤時，可在下列命令中使用標籤欄位，將標籤關聯至頻道、資料集、資料存放區及管道：

- CreateChannel
- CreateDataset
- CreateDatastore
- CreatePipeline

您可以新增、修改或刪除支援標籤的現有資源。使用下列命令：

- TagResource
- ListTagsForResource
- UntagResource

您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為空白字串，但您無法將標籤的值設為 Null。若您將與現有標籤具有相同值的標籤新增到該資源，則新值會覆寫舊值。如果您刪除資源，也會刪除與該資源相關聯的任何標籤。

搭配 IAM 政策使用標籤

您可以使用 Condition 元素 (也稱為 Condition 區塊)，與 IAM 政策中的以下條件內容金鑰/值搭配，來根據資源標籤控制使用者存取 (許可)：

- 使用iotanalytics:ResourceTag/<tag-key>: <tag-value> yo 允許或拒絕具有特定標籤的資源上的使用者動作。
- 使用 aws:RequestTag/<tag-key>: <tag-value> 以在提出 API 請求時，要求使用 (或不使用) 特定標籤，以建立或修改允許標籤的資源。
- 使用 aws:TagKeys: [<tag-key>, ...] 以在提出 API 請求時，要求使用 (或不使用) 特定標籤金鑰集，以建立或修改允許標籤的資源。

> **Note**
>
> IAM 策略中的條件內容鍵/值僅適用於那些AWS IoT Analytics動作，其中能夠標記的資源的識別碼是必要參數。例如，根據條件內容金鑰/值，不允許/拒絕使用，因為在此請求中所參照的項目沒有可標記資源 (通道、資料集、資料存放區或管道)。DescribeLoggingOptions

如需詳細資訊，請參閱《IAM 使用者指南》中的使用標籤控制存取權限。該指南的 IAM JSON 政策參照部分具有 IAM 中 JSON 政策的詳細語法、描述和範例。

以下範例政策會套用兩個限制。受此政策限制的使用者：

1. 無法提供資源「env=prod」標籤 (請參閱範例"aws:RequestTag/env" : "prod"中的行)。
2. 無法修改或存取具有現有標籤「env=prod」的資源 (請參閱範例"iotanalytics:ResourceTag/env" : "prod"中的行)。

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    },
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}
```

您也可以將多個標籤值封閉在清單中，以便為指定的標籤鍵指定多個標籤值，如下列範例所示。

```
"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> 如果您允許/拒絕使用者根據標籤存取資源，請務必考慮明確拒絕使用者將這些標籤新增至相同資源或從中移除的能力。否則，使用者可能透過修改標籤來避開您的限制，並取得資源的存取。

標籤限制

以下基本限制適用於標籤：

- 每個資源的標籤數上限：50
- 金鑰長度上限：127 個 UTF-8 Unicode 字元
- 值長度上限：255 個 UTF-8 Unicode 字元
- 標籤金鑰與值皆區分大小寫。
- 標籤名稱或值不可使用，因為它只保留給AWS使用。`aws: prefix`您不可編輯或刪除具此字首的標籤名稱或值。具此字首的標籤，不算在受來源限制的標籤計數內。
- 如果您的標記結構描述是跨多項服務和資源使用，請記得其他服務可能會有字元使用限制。通常，允許使用的字元為：可用 UTF-8 表示的英文字母、空格和數字，加上以下特殊字元：+ - = . _ : / @。

中的 SQL 表達式AWS IoT Analytics

資料集是對資料存放區的資料使用 SQL 運算式而產生。AWS IoT Analytics使用與 Amazon Athena 相同的 SQL 查詢，函數和運算子

AWS IoT Analytics支援 ANSI 標準 SQL 語法的子集。

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

如需參數的描述，請參參數中的Amazon Athena 文檔。

AWS IoT Analytics，Amazon Athena 不支援以下項目：

- WITH子句。
- CREATE TABLE AS SELECT 陳述式
- INSERT INTO 陳述式
- 準備好的語句，您無法運行EXECUTE取代為USING。
- CREATE TABLE LIKE
- DESCRIBE INPUT 與 DESCRIBE OUTPUT
- EXPLAIN 陳述式
- 使用者定義函數 (UDF 或 UDAF)
- 預存程序
- 聯合連接器

主題

- 支援的 SQL 功能AWS IoT Analytics
- 針對 SQL 查詢的常見問題進行故障診斷AWS IoT Analytics

支援的 SQL 功能AWS IoT Analytics

數據集是通過在數據存儲中的數據使用 SQL 表達式生成的。您在運行的查詢基AWS IoT Analytics於普雷斯托 0.217。

支援的資料類型

AWS IoT Analytics和 Amazon Athena 支持這些數據類型。

- primitive_type
  - TINYINT
  - SMALLINT
  - INT
  - BIGINT
  - BOOLEAN
  - DOUBLE
  - FLOAT
  - STRING
  - TIMESTAMP
  - DECIMAL(precision, scale)
  - DATE
  - CHAR（具有指定長度的固定長度字符數據）
  - VARCHAR（具有指定長度的可變長度字符數據）
- array_type
  - ARRAY<data_type>
- map_type
  - MAP<primitive_type, data_type>
- struct_type
  - STRUCT<col_name:data_type[COMMENT col_comment][,...]>

> **Note**
>
> AWS IoT Analytics和 Amazon Athena 不支持某些數據類型。

支援的函數

Amazon Athena 和AWS IoT Analytics SQL 功能是基於普雷斯托 0.217. 如需有關相關函數、運算子和表達式的資訊，請參閱 Presto 和運算子，以及 Presto 文件的以下具體章節。

- 邏輯運算子
- 比較函數和運算子
- 條件表達式
- 轉換函數
- 數學函數和運算子
- 位元函數
- Decimal 函數和運算子
- 字串函數和運算子
- 二進位函數
- 日期與時間函數和運算子
- 規則運算式函數
- JSON 函數和運算子
- URL 函數
- 彙總函數
- 視窗函數
- 色彩函數
- 陣列函數和運算子
- 對應函數和運算子
- Lambda 表達式和函數
- Teradata 函數

> **Note**
>
> AWS IoT Analytics而且 Amazon Athena 不支援使用者定義函式 (UDF 或 UDAFs) 或預存程序。

針對 SQL 查詢的常見問題進行故障診斷AWS IoT Analytics

使用下列資訊來協助您診斷AWS IoT Analytics。

- 若要轉義單引號，在其前面加另一個單引號。請勿將此與雙引號混淆。

  Example 範例

  ```
  SELECT 'O''Reilly'
  ```

- 要轉義底線中，使用反引號括住以底線開頭的數據存儲欄名稱。

  Example 範例

  ```
  SELECT `_myMessageAttribute` FROM myDataStore
  ```

- 使用數字轉義名稱中，用雙引號括住包含數字的數據存儲名稱。

  Example 範例

  ```
  SELECT * FROM "myDataStore123"
  ```

- 要轉義預留關鍵字中，用雙引號括住預留關鍵字。如需詳細資訊，請參閱「」預留關鍵字清單在SQL選擇陳述式。

中的安全性 AWS IoT Analytics

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。共同的責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲的安全性- AWS 負責保護在 AWS 雲中運行 AWS 服務的基礎設施。 AWS 還為您提供可以安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 AWS 合規計劃的一部分。若要瞭解適用於的合規計劃 AWS IoT Analytics，請參閱合規計劃的AWS 服務範圍。
- 雲端中的安全性-您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的敏感度、您組織的需求和適用的法律及法規。

本文檔將幫助您了解如何在使用時應用共同責任模型 AWS IoT Analytics。下列主題說明如何設定 AWS IoT Analytics 以符合安全性與合規性目標。您還將學習如何使用其他可以幫助您監控和保護 AWS IoT Analytics 資源的 AWS 服務。

AWS Identity and Access Management 在 AWS IoT Analytics

AWS Identity and Access Management (IAM) 是協助管理員安全地控制 AWS 資源存取的 AWS 服務。IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 AWS IoT Analytics 資源。IAM 是一項無需額外付費即可使用的 AWS 服務。

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 AWS IoT Analytics。

服務使用者 — 如果您使用 AWS IoT Analytics 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS IoT Analytics 功能來完成工作時，您可能需要其他權限。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 AWS IoT Analytics中的某項功能，請參閱 疑難排解 AWS IoT Analytics 身分和存取。

服務管理員 — 如果您負責公司的 AWS IoT Analytics 資源，您可能擁有完整的存取權 AWS IoT Analytics。決定您的服務使用者應該存取哪些 AWS IoT Analytics 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM AWS IoT Analytics，請參閱如何與 IAM AWS IoT Analytics 搭配使用。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS IoT Analytics存取權的詳細資訊。若要檢視可在 IAM 中使用的 AWS IoT Analytics 基於身分的政策範例，請參閱。AWS IoT Analytics 以識別為基礎的原則範例

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源類型政策的差異。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。

- 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理

的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 實體許可範圍。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

如何與 IAM AWS IoT Analytics 搭配使用

在您使用 IAM 管理存取權限之前 AWS IoT Analytics，您應該瞭解哪些 IAM 功能可搭配使用 AWS IoT Analytics。若要深入瞭解如何以 AWS IoT Analytics 及其他 AWS 服務如何與 IAM 搭配使用，請參閱 IAM 使用者指南中的與 IAM 搭配使用的AWS 服務。

本頁主題：

- AWS IoT Analytics 以身分為基礎的原則
- AWS IoT Analytics 資源型政策
- 基於 AWS IoT Analytics 標籤的授權

- AWS IoT Analytics IAM 角色

AWS IoT Analytics 以身分為基礎的原則

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及允許或拒絕動作的條件。 AWS IoT Analytics 支援特定動作、資源和條件索引鍵。若要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

動作

IAM 身分類型政策的 `Action` 元素會描述政策將允許或拒絕的特定動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。在策略中使用這些動作來授與執行關聯作業的權限。

原則動作在動作前 AWS IoT Analytics 使用下列前置詞：`iotanalytics:`例如，若要授予某人使用 AWS IoT Analytics `CreateChannel` API 作業建立 AWS IoT Analytics 頻道的權限，您可以將該`iotanalytics:BatchPuMessage`動作納入他們的政策中。原則陳述式必須包含`Action`或`NotAction`元素。 AWS IoT Analytics 定義了它自己的一組動作，描述您可以使用此服務執行的任務。

若要在單一陳述式中指定多個 動作，請用逗號分隔，如下所示。

```
"Action": [
    "iotanalytics:action1",
    "iotanalytics:action2"
    ]
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，如需指定開頭是 `Describe` 文字的所有動作，請包含以下動作：

```
"Action": "iotanalytics:Describe*"
```

若要查看 AWS IoT Analytics 動作清單，請參閱 IAM 使用者指南 AWS IoT Analytics中定義的動作。

資源

`Resource` 元素可指定動作套用的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。您可以使用 ARN 來指定資源，或是使用萬用字元 (*) 來指定陳述式套用到所有資源。

資 AWS IoT Analytics 料集資源具有以下 ARN。

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

如需 ARN 格式的詳細資訊，請參閱 Amazon Resource Name (ARN) 和 AWS 服務命名空間。

例如，若要在陳述式中指定 Foobar 資料集，請使用以下 ARN。

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

如需指定屬於特定帳戶的所有執行個體，請使用萬用字元 (*)。

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

某些 AWS IoT Analytics 動作 (例如用來建立資源的動作) 無法在特定資源上執行。在這些情況下，您必須使用萬用字元 (*)。

```
"Resource": "*"
```

某些 AWS IoT Analytics API 動作涉及多個資源。例如，以通道和資料集的形式CreatePipeline參照，因此使用者必須擁有使用頻道和資料集的權限。若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN。

```
"Resource": [
    "resource1",
    "resource2"
     ]
```

若要查看 AWS IoT Analytics 資源類型及其 ARN 的清單，請參閱《IAM 使用者指南》 AWS IoT Analytics中所定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS IoT Analytics定義的動作。

條件索引鍵

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建置使用條件運算子的條件表達式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。若您為單一條件索引鍵指定多個值， AWS 會使用邏輯 OR 操作評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其使用者名稱標記時，將存取資源的許可授予該使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 政策元素：變數和標籤。

AWS IoT Analytics 不提供任何 sevice 特定的條件鍵，但它確實支持使用一些全局條件鍵。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

範例

若要檢視以 AWS IoT Analytics 身分為基礎的原則範例，請參閱。AWS IoT Analytics 以識別為基礎的原則範例

AWS IoT Analytics 資源型政策

AWS IoT Analytics 不支援以資源為基礎的政策。若要檢視詳細資源型政策頁面的範例，請參閱AWS Lambda 開發人員指南 AWS Lambda中的〈使用以資源為基礎的政策〉。

基於 AWS IoT Analytics 標籤的授權

您可以將標籤附加至 AWS IoT Analytics 資源，或將要求中的標籤傳遞給 AWS IoT Analytics。若要根據標籤控制存取，請使用`iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}`或`aws:TagKeys`條件索引鍵在原則的條件元素中提供標籤資訊。有關標記 AWS IoT Analytics 資源的詳細資訊，請參閱標記資 AWS IoT Analytics 源。

若要檢視以身分識別為基礎的範例政策，以根據該資源上的標籤限制對資源的存取，請參閱根據標籤檢視 AWS IoT Analytics 頻道。

AWS IoT Analytics IAM 角色

IAM 角色是您 AWS 帳戶 中具備特定許可的實體。

使用臨時認證 AWS IoT Analytics

您可以搭配聯合使用暫時憑證、擔任 IAM 角色，或是擔任跨帳戶角色。您可以透過呼叫 AWS Security Token Service (AWS STS) API 作業 (例如AssumeRole或 GetFederationToken) 來取得臨時安全登入資料。

AWS IoT Analytics 不支援使用臨時登入資料。

服務連結角色

服務內容角色可讓 AWS 服務存取其他服務中的資源，以代表您完成動作。服務連結角色會顯示在您的 IAM 帳戶中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

AWS IoT Analytics 不支援服務連結角色。

服務角色

此功能可讓服務代表您擔任服務角色。此角色可讓服務存取其他服務中的資源，以代表您完成動作。服務角色會出現在您的 IAM 帳戶中，且由該帳戶所擁有。這表示 IAM 管理員可以變更此角色的許可。不過，這樣可能會破壞此服務的功能。

AWS IoT Analytics 支援服務角色。

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在 AWS 中，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了防止這種情況，AWS提供的各種工具可協助您保護所有服務的資料，而這些服務主體已獲得帳戶中資源的存取權。

我們建議您使用aws:SourceArn和aws:SourceAccount資源策略中的全域條件內容金鑰。這限制了權限AWS IoT Analytics為資源提供另一項服務。如果同時使用全域條件內容索引鍵，則在相同政策陳述式中使用aws:SourceAccount 值和 aws:SourceArn 值中的帳戶時，必須使用相同的帳戶 ID。

防範混淆代理人問題的最有效方法是使用 aws:SourceArn 全域條件內容索引鍵，其中包含資源的完整 Amazon Resource Name (ARN)。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 aws:SourceArn 全域條件內容索引鍵，同時使用萬用字元 (*) 表示 ARN 的未知部分。例如：arn:aws:*iotanalytics*::*123456789012*:*。

主題

- 適用於 Amazon S3水桶
- 亞馬遜預防 CloudWatch 日誌
- 客戶管理人AWS IoT Analytics資源

適用於 Amazon S3水桶

如果您使用客戶管理的 Amazon S3 儲存AWS IoT Analytics資料存放區 (存放資料的 Amazon S3 儲存貯體) 可能會面臨混淆的副問題。

例如，尼克沃爾夫使用客戶擁有的 Amazon S3 存儲桶稱為*#####*。存儲桶存儲的信息AWS IoT Analytics在區域中建立的資料倉庫*us-east-1*。她指定了一個策略，以啟用AWS IoT Analytics要查詢的服務主體*#####*代表她 妮琪的同事, 李娟, 查詢*#####*從她自己的帳戶並創建一個包含結果的數據

集。其結果是，AWS IoT Analytics服務負責人代表李查詢 Nikki 的 Amazon S3 桶，即使李從她的帳戶中運行了查詢。

為了防止這種情況，Nikki 可以指定aws:SourceAccount條件或aws:SourceArn政策中的條件#####。

指定**aws:SourceAccount**條件-下列儲存貯體政策範例僅指定AWS IoT Analytics來自尼克賬戶的資源（*123456789012*) 可以存取*#####*。

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

指定**aws:SourceArn**條件-或者，尼克可以使用aws:SourceArn條件。

```
{
    "Version": "2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-
EXAMPLE-DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-
EXAMPLE-DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

亞馬遜預防 CloudWatch 日誌

您可以防止責任混淆代理人問題 CloudWatch 記錄檔。以下資源策略顯示了如何防止混淆的副問題：

- 全域條件內容金鑰`aws:SourceArn`

- 所以此`aws:SourceAccount`與您的AWS帳戶 ID
- 與`sts:AssumeRole`要求AWS IoT Analytics

取代*123456789012*與您的AWS帳戶 ID，以及*us-east-1*與您的區域AWS IoT Analytics在下面的例子中的帳戶。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
                }
            }
    ]
}
```

如需啟用和設定 Amazon 的詳細資訊 CloudWatch 記錄檔，請參閱the section called “記錄和監控”。

客戶管理人AWS IoT Analytics資源

如果你授予AWS IoT Analytics對您執行動作的許可AWS IoT Analytics資源，資源可能會暴露於混淆的副問題。為了防止混淆的副問題，您可以限制給予的權限AWS IoT Analytics具有以下示例資源策略。

主題

- 預防AWS IoT Analytics渠道和資料存放區
- 跨服務混淆代理人AWS IoT Analytics資料集內容交付規則

預防AWS IoT Analytics渠道和資料存放區

您使用 IAM 角色，控制AWS資源AWS IoT Analytics可以代表您存取權。若要防止將您的角色暴露在混淆的副問題中，您可以指定AWS帳戶中`aws:SourceAccount`元素和的 ARNAWS IoT Analytics中的資源`aws:SourceArn`您附加至角色之信任原則的元素。

在下列範例中，替換*123456789012*與您的AWS帳戶 ID 和*ARN: AW: #####:aws-region:123456789012###/####-##*具有一個 ARNAWS IoT Analytics通道或資料存放區。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL"
        }
      }
    }
  ]
}
```

若要進一步了解通道和資料存放區的客戶受管 S3 儲存選項，請參閱CustomerManagedChannelS3Storage和CustomerManagedDatastoreS3Storage在AWS IoT AnalyticsAPI 參考。

跨服務混淆代理人AWS IoT Analytics資料集內容交付規則

IAM 角色AWS IoT Analytics假設將資料集查詢結果交付給 Amazon S3 或AWS IoT Events可以暴露於混淆的副問題。為了預防混淆代理人問題，請指定AWS帳戶中`aws:SourceAccount`元素和的 ARNAWS IoT Analytics中的資源`aws:SourceArn`您附加到角色的信任原則元素。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:dataset/DOC-
EXAMPLE-DATASET"
        }
      }
    }
  ]
}
```

如需設定資料集內容傳遞規則的詳細資訊，請參閱contentDeliveryRules在AWS IoT AnalyticsAPI參考。

AWS IoT Analytics 以識別為基礎的原則範例

根據預設，使用者和角色不具備建立或修改 AWS IoT Analytics 資源的權限。他們也無法使用 AWS Management Console AWS CLI、或 AWS API 執行工作。IAM 管理員必須建立 IAM 政策，授予使用者和角色在指定資源上執行特定 API 操作的所需許可。管理員接著必須將這些政策連接至需要這些許可的使用者或群組。

若要了解如何使用這些 JSON 政策文件範例建立 IAM 身分型政策，請參閱 IAM 使用者指南中的 JSON索引標籤上建立政策

本頁主題：

- 政策最佳實務
- 使用控 AWS IoT Analytics 制台
- 允許使用者檢視他們自己的許可

- 存取一個 AWS IoT Analytics 輸入
- 根據標籤檢視 AWS IoT Analytics 頻道

政策最佳實務

身分型政策相當強大。他們決定是否有人可以建立、存取或刪除您帳戶中的 AWS IoT Analytics 資源。這些動作可能會讓您的 AWS 帳戶產生成本。當您建立或編輯身分類型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策-若要 AWS IoT Analytics 快速開始使用，請使用 AWS 受管理的政策提供員工所需的權限。這些政策已在您的帳戶中提供，並由其維護和更新 AWS。如需詳細資訊，請參閱 IAM 使用者指南中的 AWS 受管政策開始使用許可。
- 授與最少權限-當您建立自訂原則時，僅授與執行工作所需的權限。以最小一組許可開始，然後依需要授予額外的許可。這比一開始使用太寬鬆的許可，稍後再嘗試將他們限縮更為安全。如需詳細資訊，請參閱《IAM 使用者指南》中的授予最低權限。
- 針對敏感作業啟用 MFA-為了提高安全性，使用者必須使用多重要素驗證 (MFA) 來存取敏感資源或 API 作業。如需詳細資訊，請參閱《IAM 使用者指南》中的在 AWS中使用多重要素驗證 (MFA)。
- 使用策略條件以獲得額外的安全性-在可行的範圍內，定義以身份為基礎的策略允許存取資源的條件。例如，您可以撰寫條件來指定要求必須來自的允許 IP 位址範圍。您也可以撰寫條件，只在指定的日期或時間範圍內允許請求，或是要求使用 SSL 或 MFA。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

使用控 AWS IoT Analytics 制台

若要存取 AWS IoT Analytics 主控台，您必須擁有最少一組權限。這些權限必須允許您列出和檢視有關 AWS 帳戶. AWS IoT Analytics 如果您建立的以身分識別為基礎的原則，而該原則的限制性比所需的最低權限更嚴格。主控台將無法如預期用於具有該原則的實體 (使用者或角色) 運作。

若要確保這些實體仍可使用 AWS IoT Analytics 主控台，請同時將下列 AWS 受管理的原則附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "iotanalytics:BatchPutMessage",
```

```
                "iotanalytics:CancelPipelineReprocessing",
                "iotanalytics:CreateChannel",
                "iotanalytics:CreateDataset",
                "iotanalytics:CreateDatasetContent",
                "iotanalytics:CreateDatastore",
                "iotanalytics:CreatePipeline",
                "iotanalytics:DeleteChannel",
                "iotanalytics:DeleteDataset",
                "iotanalytics:DeleteDatasetContent",
                "iotanalytics:DeleteDatastore",
                "iotanalytics:DeletePipeline",
                "iotanalytics:DescribeChannel",
                "iotanalytics:DescribeDataset",
                "iotanalytics:DescribeDatastore",
                "iotanalytics:DescribeLoggingOptions",
                "iotanalytics:DescribePipeline",
                "iotanalytics:GetDatasetContent",
                "iotanalytics:ListChannels",
                "iotanalytics:ListDatasetContents",
                "iotanalytics:ListDatasets",
                "iotanalytics:ListDatastores",
                "iotanalytics:ListPipelines",
                "iotanalytics:ListTagsForResource",
                "iotanalytics:PutLoggingOptions",
                "iotanalytics:RunPipelineActivity",
                "iotanalytics:SampleChannelData",
                "iotanalytics:StartPipelineReprocessing",
                "iotanalytics:TagResource",
                "iotanalytics:UntagResource",
                "iotanalytics:UpdateChannel",
                "iotanalytics:UpdateDataset",
                "iotanalytics:UpdateDatastore",
                "iotanalytics:UpdatePipeline"
            ],
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
            "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
        }
    ]
```

```
}
```

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合您嘗試執行之 API 操作的動作就可以了。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 使用者檢視連接到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

存取一個 AWS IoT Analytics 輸入

在此範例中，您想要授與使用者 AWS 帳戶 存取您其中一個 AWS IoT Analytics 頻道的權限，`exampleChannel`. 您還希望允許使用添加，更新和刪除頻道。

此原則會將`iotanalytics:ListChannels, iotanalytics:DescribeChannel, iotanalytics:CreateChannel, iotanalytics:DeleteChannel, and iotanalytics:UpdateChannel`權限授與使用者。如需授與使用者許可並使用主控台測試權限的 Amazon S3 服務範例逐步解說，請參閱[範例逐步解說：使用使用者政策控制儲存貯體的]存取。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListChannelsInConsole",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:ListChannels"
         ],
         "Resource":"arn:aws:iotanalytics:::*"
      },
      {
         "Sid":"ViewSpecificChannelInfo",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:DescribeChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel"
      },
      {
         "Sid":"ManageChannels",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:CreateChannel",
            "iotanalytics:DeleteChannel",
            "iotanalytics:DescribeChannel",
            "iotanalytics:ListChannels",
            "iotanalytics:UpdateChannel"
         ],
         "Resource":"arn:aws:iotanalytics:::exampleChannel/*"
      }
```

```
    ]
}
```

根據標籤檢視 AWS IoT Analytics 頻道

您可以使用以身分識別為基礎的原則中的條件，根據標籤來控制 AWS IoT Analytics 資源的存取。此範例會示範如何建立政策，允許檢視 channel。不過，只有當channel標籤Owner具有該使用者的使用者名稱值時，才會授與權限。此政策也會授予在主控台完成此動作的必要許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListChannelsInConsole",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "*"
        },
        {
            "Sid": "ViewChannelsIfOwner",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "arn:aws:iotanalytics:*:*:channel/*",
            "Condition": {
                "StringEquals": {"iotanalytics:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

您可以將此政策連接到您帳戶中的 使用者。如果名為的使用者richard-roe嘗試檢視 AWS IoT Analytics channel，則channel必須加上標籤Owner=richard-roe or owner=richard-roe。否則，他便會被拒絕存取。條件標籤金鑰 Owner 符合 Owner 和 owner，因為條件金鑰名稱不區分大小寫。如需更多資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。

疑難排解 AWS IoT Analytics 身分和存取

使用下列資訊可協助您診斷及修正使用時可能會遇到的常見問題 AWS IoT Analytics。

主題

- 我沒有執行操作的授權 AWS IoT Analytics

- 我未獲得執行 iam:PassRole 的授權
- 我想允許我以外的人訪 AWS 帳戶 問我的 AWS IoT Analytics 資源

我沒有執行操作的授權 AWS IoT Analytics

如果 AWS Management Console 告訴您您沒有執行動作的授權，您必須聯絡您的系統管理員尋求協助。您的管理員是為您提供使用者名稱和密碼的人員。

當使用mateojackson者嘗試使用主控台來檢視有關channel但沒有iotanalytics:ListChannels權限的詳細資料時，就會發生下列範例錯誤。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

在此情況下，Mateo 會要求管理員更新其策略，以允許他使用此iotanalytics:ListChannel動作存取my-example-channel資源。

我未獲得執行 **iam:PassRole** 的授權

如果您收到錯誤，告知您未獲授權執行 iam:PassRole 動作，您的政策必須更新，允許您將角色傳遞給 AWS IoT Analytics。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 AWS IoT Analytics中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪 AWS 帳戶 問我的 AWS IoT Analytics 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。對於支援以資源為基礎的政策或存取控制清單 (ACL) 的服務，您可以使用這些政策授與人員存取您資源的權限。

如需進一步了解，請參閱以下內容：

- 若要了解是否 AWS IoT Analytics 支援這些功能，請參閱如 AWS IoT Analytics 何使用 IAM。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源型政策的差異。

AWS IoT Analytics 中的記錄和監控

AWS 提供可讓您用來監控 AWS IoT Analytics 的工具。您可設定部分這些工具以為您執行監控工作。部分工具將需要手動操作。建議您盡可能自動化監控任務。

自動化監控工具

您可以使用下列自動化監控工具來監看 AWS IoT，並在發生錯誤時進行回報：

- Amazon CloudWatch Logs-監控、存放及存取來自AWS CloudTrail或其他來源的日誌檔案。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的AWS CloudTrail監控日誌檔案。
- AWS CloudTrail日誌監控-在帳戶之間共用日誌檔、將日 CloudTrail 誌檔傳送至 Logs 以對其進行即時監控、以 Java 撰寫 CloudWatch 日誌處理應用程式，以及驗證傳遞日誌檔之後尚未對其進行變更 CloudTrail。若要取得更多資訊，請參閱《使用指南》中的〈AWS CloudTrail使用 CloudTrail 記錄檔〉。

手動監控工具

監控 AWS IoT 的另一個重要部分包含手動監控 CloudWatch 警示未涵蓋的項目。AWS IoT CloudWatch、和其他AWS服務主控台儀表板可提供您AWS環境狀態的 at-a-glance 檢視。建議您也查看 AWS IoT Analytics 上的日誌檔。

- AWS IoT Analytics 主控台會顯示：
  - 頻道

- 管道
- 資料存放區
- 資料集
- 筆記本
- 設定
- 了解

- CloudWatch 首頁會顯示：
  - 目前警示與狀態
  - 警示與資源的圖表
  - 服務運作狀態

  此外，您還可以使用執 CloudWatch 行下列動作：
  - 建立自定儀表板來監控您注重的服務
  - 繪製指標資料圖表，以對問題進行故障診斷並探索趨勢
  - 搜尋與瀏覽您所有的 AWS 資源指標
  - 建立與編輯要通知發生問題的警示

使用亞馬遜 CloudWatch 日誌監控

AWS IoT Analytics支持使用亞馬遜記錄 CloudWatch。您可以使用 `PutLoggingOptions`API 操作啟用和設定亞馬遜 CloudWatch 日誌記錄。AWS IoT Analytics本節說明如何使`PutLoggingOptions`用 AWS Identity and Access Management (IAM) 來設定和啟用 Amazon CloudWatch 日誌記錄AWS IoT Analytics。

如需 CloudWatch 日誌的詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》。如需AWS IAM 的詳細資訊，請參閱《AWS Identity and Access Management使用者指南》。

> **Note**
>
> 在啟用AWS IoT Analytics記錄功能前，請確保您了解 Lo CloudWatch gs 存取許可。具有 CloudWatch Logs 存取權限的使用者可以看到您的除錯資訊。如需詳細資訊，請參閱 Amazon CloudWatch Logs 的身分驗證和存取控制。

建立 IAM 角色以啟用記錄

建立 IAM 角色以啟用 Amazon 的日誌記錄 CloudWatch

1. 使用 AWSIAM 主控台或下列AWS IAM CLI 命令建立具有信任關係政策 (信任政策) 的新 IAM 角色。CreateRole信任政策會將採用該角色的許可授予 Amazon CloudWatch 之類的許可授予實體 (例如 Amazon)。

```
aws iam create-role --role-name exampleRoleName --assume-role-policy-document
       exampleTrustPolicy.json
```

該exampleTrustPolicy.json文件包含以下內容。

> **Note**
>
> 此範例包含全域條件內容金鑰，以防範混淆代理人安全問題。將 *123456789012* 取代為您的AWS帳戶識別碼和 *aws-region*，並以您的AWS資源所在AWS地區取代。如需詳細資訊，請參閱the section called “預防跨服務混淆代理人”。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Service": "iotanalytics.amazonaws.com"
         },
         "Action": "sts:AssumeRole",
         "Condition": {
           "StringEquals": {
             "aws:SourceAccount": "123456789012"
           },
           "ArnLike": {
             "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:*"
           }
         }

      }
   ]
```

```
}
```

稍後當您呼叫AWS IoT AnalyticsPutLoggingOptions命令時，您可以使用此角色的 ARN。

2. 使用AWS IAM PutRolePolicy將許可政策 (arole policy) 附加到您在步驟 1 中建立的角色。

```
aws iam put-role-policy --role-name exampleRoleName --policy-name
examplePolicyName --policy-document exampleRolePolicy.json
```

exampleRolePolicy.json 檔案包含下列內容。

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream"
   ],
   "Resource": [
    "arn:aws:logs:*:*:*"
   ]
   }
   ]
}
```

3. 要AWS IoT Analytics授予將日誌記錄事件放到亞馬遜的權限 CloudWatch，請使用亞馬遜 CloudWatch 命令PutResourcePolicy。

> **Note**
>
> 為了避免混淆的副安全性問題，我們建議您aws:SourceArn在資源策略中指定。這會限制存取權限，只允許來自指定帳戶的要求。如需混淆代理人問題的更多資訊，請參閱the section called “預防跨服務混淆代理人”。

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

exampleResourcePolicy.json檔案包含下列資源策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition":{
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/
*"
                },
                "StringEquals":{
                    "aws:SourceAccount":"123456789012"
                }
            }
    ]
}
```

設定及啟用記錄

使用命PutLoggingOptions令設定和啟用 Amazon CloudWatch 日誌記錄AWS IoT Analytics。loggingOptions 欄位中的 roleArn，應為您在前一節所建立的角色 ARN。您也可以使用 DecribeLoggingOptions 命令來檢查記錄選項設定。

PutLoggingOptions

設定或更新記AWS IoT Analytics錄選項。如果您更新任何loggingOptions欄位的值，變更最多需要一分鐘才會生效。此外，如果您變更附加至roleArn欄位中指定角色的原則 (例如，要更正無效的策略)，該變更最多可能需要五分鐘才會生效。如需詳細資訊，請參閱PutLoggingOptions。

DescribeLoggingOptions

擷取AWS IoT Analytics記錄選項的目前設定。如需詳細資訊，請參閱 DescribeLoggingOptions

命名空間、量度和維度

AWS IoT Analytics將以下指標放入 Amazon CloudWatch 儲存庫中：

命名空間
AWS/物IoTAnalytics

指標	描述
ActionExecution	執行的動作數目。
ActionExecutionThrottled	調節的動作次數。
ActivityExecutionError	執行管道活動時產生的錯誤數量。
IncomingMessages	進入頻道的訊息數量。
PipelineConcurrentExecutionCount	已同時執行的管線活動數目。

維度	描述
ActionType	正在監控的動作類型。
ChannelName	正在監控的頻道名稱。
DatasetName	正在監控的資料集名稱。
DatastoreName	正在監控的資料存放區名稱。
PipelineActivityName	正在監控的管道活動名稱。
PipelineActivityType	正在監控的管道活動類型。
PipelineName	正在監控的管道名稱。

使用亞馬遜 CloudWatch 活動監控

AWS IoT Analytics活動期間發生執行階段錯誤時，會自AWS Lambda動將事件發佈至 Amazon CloudWatch 事件。此事件包含詳細的錯誤訊息，以及存放未處理通道訊息的 Amazon Simple Storage Service (Amazon S3) 物件。您可以使用 Amazon S3 金鑰重新處理未處理的通道訊息。如需詳細資

訊重新處理頻道消息，請參閱 StartPipelineReprocessingAPI 參考中的AWS IoT Analytics API 以及 Amazon CloudWatch 事件使用者指南中的「什麼是亞馬遜 CloudWatch 事件」。

您也可以設定讓 Amazon CloudWatch 事件傳送通知或採取進一步動作的目標。例如，您可以將通知傳送至 Amazon Simple Queue Service (Amazon SQS) 佇列，然後呼叫`StartReprocessingMessage` API 處理 Amazon S3 物件中儲存的通道訊息，以處理儲存在 Amazon S3 物件中的通道訊息。Amazon CloudWatch 活動支援多種類型的目標，例如：

- Amazon Kinesis Streams
- AWS Lambda 函式
- Amazon Simple Notification Service (Amazon SNS) 主題
- Amazon Simple Queue Service (Amazon SQS) 佇列

如需支援的目標清單，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 目標。

您的 CloudWatch 事件資源和關聯的目標必須位於您建立AWS IoT Analytics資源的AWS區域中。如需詳細資訊，請參閱《AWS 一般參考》中的服務端點和配額。

針對AWS Lambda活動中的執行階段錯誤，傳送給 Amazon E CloudWatch vents 的通知會使用下列格式。

```
{
    "version": "version-id",
    "id": "event-id",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "aws-account",
    "time": "timestamp",
    "region": "aws-region",
    "resources": [
        "pipeline-arn"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "pipeline-name",
        "error-code": "LAMBDA_FAILURE",
        "message": "error-message",
        "channel-messages": {
            "s3paths": [
                "s3-keys"
```

```
            ]
        },
        "activity-name": "lambda-activity-name",
        "lambda-function-arn": "lambda-function-arn"
    }
}
```

通知範例：

```
{
    "version": "0",
    "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "123456789012",
    "time": "2020-10-15T23:47:02Z",
    "region": "ap-southeast-2",
    "resources": [
        "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "test_pipeline_failure",
        "error-code": "LAMBDA_FAILURE",
        "message": "Temp unavaliable",
        "channel-messages": {
        "s3paths": [
            "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
 00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
        ]
    },
    "activity-name": "LambdaActivity_33",
    "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
    }
}
```

透過 Amazon CloudWatch 活動取得延遲資料通知

使用指定時間範圍中的資料建立資料集內容時，部分資料可能無法及時送達處理。若要允許延遲，您可以套用queryAction (SQL 查詢) 來指定建立資料集QueryFilter時的deltaTime偏移量。 AWS

IoT Analytics仍會處理在增量時間內到達的資料，而您的資料集內容有時間延遲。延遲資料通知功能可讓您AWS IoT Analytics在資料超過差異時間後到達時，透過 Amazon E CloudWatch vents 傳送通知。

您可以使用AWS IoT Analytics主控台、API、AWS Command Line Interface(AWS CLI) 或 AWSSDK 來指定資料集的後期資料規則。

在AWS IoT Analytics API 中，`LateDataRuleConfiguration`物件代表資料集的後期資料規則設定。此物件是與`CreateDataset`和`UpdateDataset` API 作業相關聯之`Dataset`物件的一部分。

參數

當您為資料集建立延遲資料規則時AWS IoT Analytics，請務必指定下列資訊：

`ruleConfiguration` (**`LateDataRuleConfiguration`**)

包含延遲資料規則之組態資訊的結構。

`deltaTimeSessionWindowConfiguration`

包含差異時間工作階段時段之組態資訊的結構。

DeltaTime 指定時間間隔。您可以使用 `DeltaTime` 建立資料集內容，其中包含自上次執行以來已到達資料存放區的資料。如需範例`DeltaTime`，請參閱使用差異時段 (CLI) 建立 SQL 資料集。

`timeoutInMinutes`

時間間隔。您可以使用，AWS IoT Analytics以`timeoutInMinutes`便可以批次處理自上次執行以來產生的延遲資料通知。 AWS IoT Analytics同時傳送一批通知給 E CloudWatch vents。

類型：整數

有效範圍：1-60

`ruleName`

延遲資料規則的名稱。

類型：字串

> **Important**
>
> 若要指定`lateDataRules`，資料集必須使用`DeltaTime`篩選條件。

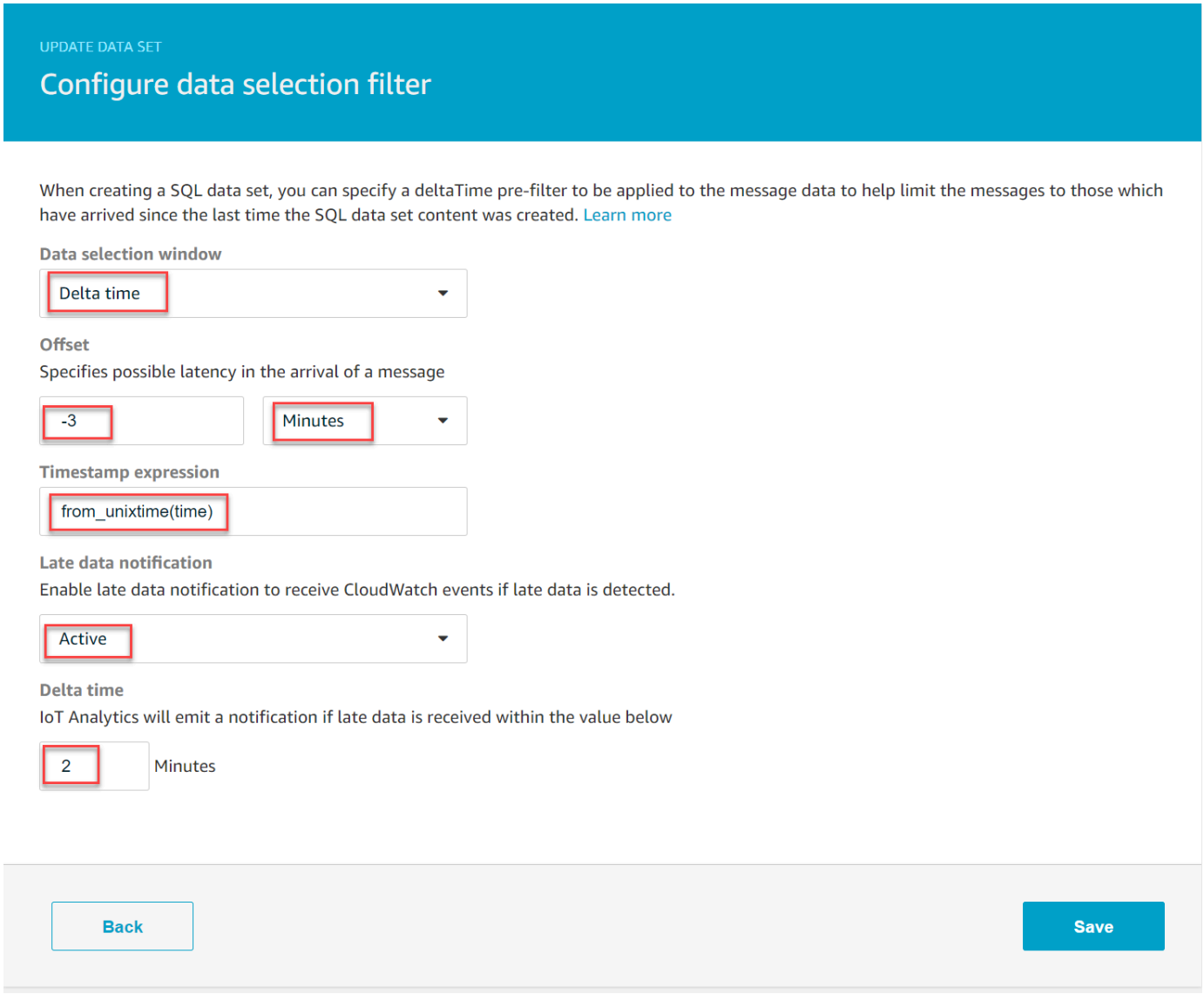
設定延遲資料規則 (主控台)

下列程序顯示如何設定AWS IoT Analytics主控台中資料集的延遲資料規則。

若要設定延遲資料規則

1. 登入 AWS IoT Analytics 主控台。
2. 在導覽窗格中，選擇資料集。
3. 在 [資料集] 下，選擇目標資料集。
4. 在導覽窗格中，選擇詳細資訊。
5. 在「差異」視窗區段中，選擇 「編輯」。
6. 在設定資料選取篩選條件之下，執行下列操作：
   a. 在「資料選取」視窗中，選擇差異時間。
   b. 在「偏移」 中，輸入時間週期，然後選擇一個單位。
   c. 在時間戳記表示式中，輸入表示式。這可以是時間戳記欄位的名稱，或可導出時間的 SQL 運算式，例如 *`from_unixtime (##)`*。

      如需如何編寫時間戳記表達式的更多資訊，請參閱《Presto 0.172 文件》中的日期和時間函數和運算子。
   d. 對於延遲資料通知，請選擇作用中。
   e. 對於差值時間，輸入整數。有效範圍為 1-60。
   f. 選擇 儲存 。



設定延遲資料規則 (CLI)

在AWS IoT Analytics API 中，`LateDataRuleConfiguration`物件代表資料集的後期資料規則設定。此物件是與和相關聯之`Dataset`物件的一部`CreateDataset`分`UpdateDataset`。您可以使用 API AWS CLI、或 AWSSDK 來指定資料集的後期資料規則。下列為使用 AWS CLI 的範例。

若要建立具有指定延遲資料規則的資料集，請執行下列命令。此命令會假設`dataset.json`檔案位於目前的目錄。

> **Note**
>
> 您可以使用 UpdateDatasetAPI 更新現有資料集。

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

dataset.json檔案應包含以下內容：

- 以目標*####### demo_* dataset。
- 將*#######*取代為目標資料存放區名稱。
- 用*###############*間的 SQL 運算式，以導出時間。

  如需如何編寫時間戳記表達式的更多資訊，請參閱《Presto 0.172 文件》中的日期和時間函數和運算子。
- 用 1-60 之間的整數替換*##*。
- 用任何名稱替換*####*。

```
{
    "datasetName": "demo_dataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "filters": [
                    {
                        "deltaTime": {
                            "offsetSeconds": -180,
                            "timeExpression": "from_unixtime(time)"
                        }
                    }
                ],
                "sqlQuery": "SELECT * FROM demo_datastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": false,
        "numberOfDays": 90
```

```
    },
    "lateDataRules": [
        {
            "ruleConfiguration": {
                "deltaTimeSessionWindowConfiguration": {
                    "timeoutInMinutes": timeout
                }
            },
            "ruleName": "demo_rule"
        }
    ]
}
```

訂閱接收延遲資料通知

您可以在 CloudWatch 事件中建立規則，以定義如何處理從傳送的延遲資料通知AWS IoT Analytics。當 CloudWatch 事件收到通知時，它會調用規則中定義的指定目標動作。

建立 CloudWatch 事件規則的先決條件

在您為其建立 E CloudWatch vents 規則之前AWS IoT Analytics，請執行下列操作：

- 熟悉 Events 中 CloudWatch 的事件、規則和目標。
- 建立和設定 CloudWatch 事件規則呼叫的目標。規則可以叫用許多類型的目標，例如以下類型：
  - Amazon Kinesis Streams
  - AWS Lambda 函式
  - Amazon Simple Notification Service (Amazon SNS) 主題
  - Amazon Simple Queue Service (Amazon SQS) 佇列

  您的 CloudWatch 事件規則和相關聯的目標必須位於您建立AWS IoT Analytics資源的「AWS區域」中。如需詳細資訊，請參閱《AWS 一般參考》中的服務端點和配額。

如需詳細資訊，請參閱什麼是 E CloudWatch vents？ 並在亞馬遜 CloudWatch 活動用戶指南中開始使用亞馬遜 CloudWatch 活動。

延遲資料通知事件

延遲資料通知的事件使用下列格式。

```
{
```

```
 "version": "0",
 "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
 "detail-type": "IoT Analytics Dataset Lifecycle Notification",
 "source": "aws.iotanalytics",
 "account": "123456789012",
 "time": "2020-05-14T02:38:46Z",
 "region": "us-east-2",
 "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
 "detail": {
  "event-detail-version": "1.0",
  "dataset-name": "demo_dataset",
  "late-data-rule-name": "demo_rule",
  "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
  "message": null
 }
}
```

建立 CloudWatch 事件規則以接收延遲資料通知

下列程序說明如何建立規則，以便將AWS IoT Analytics延遲資料通知傳送至 Amazon SQS 佇列。

建立 E CloudWatch vents 規則

1. 登錄到亞馬遜 CloudWatch控制台。
2. 在導覽窗格的 Events (事件) 下，選擇 Rules (規則)。
3. 在 [規則] 頁面上，選擇 [建立規則]。
4. 在「事件來源」下選擇「事件模式」。
5. 在 [建立依服務比對事件的事件模式] 區段中，執行下列動作：

   a. 針對「服務名稱」，選擇 IoT Analytics

   b. 對於事件類型，請選擇 IoT Analytics 資料集生命週期通知。

   c. 選擇 [特定資料集名稱]，然後輸入目標資料集的名稱。
6. 在「目標」下，選擇「新增目標 *」。
7. 選擇 SQS 佇列，然後執行下列動作：

   - 在佇列 * 中，選擇目標佇列。
8. 選擇 Configure details (設定詳細資訊)。
9. 在 [步驟 2：設定規則詳細資料] 頁面上，輸入名稱和說明。
10. 選擇 Create rule (建立規則)。

使用 AWS CloudTrail 記錄 AWS IoT Analytics API 呼叫

AWS IoT Analytics與整合AWS CloudTrail，這項服務可提供由使用者、角色或中AWS服務所採取之動作的記錄AWS IoT Analytics。 CloudTrail 擷取AWS IoT Analytics為事件的 API 呼叫，包括來自AWS IoT Analytics主控台以及來自對 API 發出的程AWS IoT Analytics式碼呼叫，以及來自對 API 發出的程式碼呼叫。如果您建立追蹤，就可以將事件持續交付至 Amazon S3 儲存貯體，包括的 CloudTrail 事件，包括的事件，包括的事件AWS IoT Analytics。即使未設定追蹤，您依然可以在 [事件歷史記錄] 中的 CloudTrail 主控台檢視最新的事件。使用由收集的資訊 CloudTrail，您就可以判斷提出的請求、提出請求的對象AWS IoT Analytics、提出請求的對象、提出請求的對象、提出請求的對象、提出請求的對象、提出請求的對象、提出請求的對象、提出請求

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

AWS CloudTrail 中的 AWS IoT Analytics 資訊

CloudTrail 當您建立AWS帳戶時，系統會在您的帳戶中啟用。此外，此外AWS IoT Analytics，系統便會將該活動記錄至 CloudTrail 事件，並將其他AWS服務事件記錄到事件中。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 事件歷程記錄檢視事件。

如需您 AWS 帳戶中正在進行事件的記錄 (包含 AWS IoT Analytics 的事件)，請建立線索。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區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您還能設定其他AWS服務，以進一步分析和處理 CloudTrail 日誌中收集的事件資料，並採取相應動作。如需詳細資訊，請參閱：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定的 Amazon SNS 通知 CloudTrail
- 接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案

AWS IoT Analytics支援將下列 API 動作記錄為 CloudTrail 日誌檔案中的事件：

- CancelPipelineReprocessing
- CreateChannel
- CreateDataset
- CreateDatasetContent
- CreateDatastore

- CreatePipeline
- DeleteChannel
- DeleteDataset
- DeleteDatasetContent
- DeleteDatastore
- DeletePipeline
- DescribeChannel
- DescribeDataset
- DescribeDatastore
- DescribeLoggingOptions
- DescribePipeline
- GetDatasetContent
- ListChannels
- ListDatasets
- ListDatastores
- ListPipelines
- PutLoggingOptions
- RunPipelineActivity
- SampleChannelData
- StartPipelineReprocessing
- UpdateChannel
- UpdateDataset
- UpdateDatastore
- UpdatePipeline

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

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

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

了解 AWS IoT Analytics 日誌檔案項目

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

以下範例顯示的是展示CreateChannel動作的 CloudTrail 日誌項目。

```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsChannelTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:43:12Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:55:14Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateChannel",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"channelName": "channel_channeltest"
},
```

```
"responseElements": {
"retentionPeriod": {
"unlimited": true
},
"channelName": "channel_channeltest",
"channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

以下範例顯示的是展示CreateDataset動作的 CloudTrail 日誌項目。

```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsDatasetTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:41:36Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:53:39Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateDataset",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
```

```
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"datasetName": "dataset_datasettest"
},
"responseElements": {
"datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/
dataset_datasettest",
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```


符合性驗證 AWS IoT Analytics

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。

- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

韌性在 AWS IoT Analytics

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。 AWS 區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援聯網功能相互連結。使用可用區域，您可以設計和操作應用程式和資料庫，在可用區域之間自動容錯移轉，而不會中斷。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。

如需區域和可用區域的詳 AWS 細資訊，請參閱AWS 全域基礎結構。

基礎結構安全 AWS IoT Analytics

作為託管服務， AWS IoT Analytics 受到 AWS 全球網絡安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路進行存取。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。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，您可以使用 AWS Security Token Service (AWS STS) 以產生暫時安全憑證以簽署請求。

AWS IoT Analytics 配額

本AWS 一般參考指南提供帳戶的預設配AWS額。AWS IoT Analytics除非另有說明，否則每項配額都是依AWS區域計 如需詳細資訊，請參閱AWS 一般參考指南中的AWS IoT Analytics端點和配額以及AWS服務配額。

若要請求提高服務配額，請求提高服務配額，請求提高支援案例。如需詳細資訊，請參閱《Service Quotas 使用者指南》中的請求提高配額。

AWS IoT Analytics 命令

了解用於AWS IoT Analytics，包括受支持的 Web 服務協議的示例請求、響應和錯誤。

AWS IoT Analytics 動作

您可以使用AWS IoT AnalyticsAPI 命令來收集、處理、存儲和分析您的 IoT 數據。如需詳細資訊，請參閱 。動作支援AWS IoT Analytics中的AWS IoT AnalyticsAPI 參考。

所以此AWS IoT Analytics部分中的AWS CLI命令參考包含AWS CLI命令，您可以用這些命令來管理和操作AWS IoT Analytics。

AWS IoT Analytics 資料

您可以使用AWS IoT Analytics用於執行高級活動的數據 API 命令AWS IoT Analytics `channel`、`pipeline`、`datastore`，以及`dataset`。如需詳細資訊，請參閱 。資料類型支援AWS IoT Analytics中的數據AWS IoT AnalyticsAPI 參考。

AWS IoT Analytics 疑難排解

請參閱下一節以疑難排解錯誤，並尋找與解決問題的可能解決方案AWS IoT Analytics。

主題

如何知道可以在訊息增加AWS IoT Analytics？

檢查透過規則引擎將資料插入通道的規則是否設定正確。

```
aws iot get-topic-rule --rule-name your-rule-name
```

回應應應如下所示。

```
{
    "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
```

```
    "rule": {
        "awsIotSqlVersion": "2016-03-23",
        "sql": "SELECT * FROM 'iot/your-rule-name'",
        "ruleDisabled": false,
        "actions": [
            {
                "iotAnalytics": {
                    "channelArn":
 "arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
                }
            }
        ],
        "ruleName": "your-rule-name"
    }
}
```

確認用於規則的區域和頻道名稱是否正確。為了確保您的資料到達規則引擎並且規則正確執行，您可能需要新增一個新目標，以暫時將內送訊息存放在 Amazon S3 儲存貯體中。

為什麼我的管道丟失消息？ 我該如何修正這個問題？

- 活動收到無效的 JSON 輸入：

  除 Lambda 活動外，所有活動都特別需要有效的 JSON 字串作為輸入。如果活動收到的 JSON 無效，則訊息會被捨棄，無法進入資料存放區。請確定您擷取有效的 JSON 訊息至服務。若是二進位輸入，請確保您管道中的第一個活動是 Lambda 活動，它可將二進位資料轉換為有效的 JSON，然後再傳遞給下一個活動或儲存到資料存放區。如需詳細資訊，請參閱 Lambda Function 2。

- Lambda 活動叫用的 Lambda 函數沒有足夠的許可：

  請確定 Lambda 活動中的每個 Lambda 函數都具有從AWS IoT Analytics服務叫用的權限。可以使用下列AWS CLI命令授與許可。

  ```
  aws lambda add-permission --function-name <name> --region <region> --statement-id
   <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
  ```

- 篩選條件或 removeAttribute 活動的定義不正確：

  確保定義是否有任何`filter`或`removeAttribute`活動正確。如果您篩選掉訊息或移除訊息的所有屬性，該訊息不會新增到資料存放區。

為什麼我的資料存放區中沒有資料？

- 資料擷取和資料可供使用之間存在延遲：

  資料擷取至頻道後可能需要幾分鐘的時間，資料才能在資料存放區提供使用。所需時間不一，取決於管道活動數量和管道中的任何自訂 Lambda 活動的定義。

- 訊息在您的管道中被篩選掉：

  確定您未刪除管道中的訊息。(請參閱上一個問題和回覆。)

- 您的資料集查詢不正確：

  請確定從資料存放區產生資料集的查詢正確無誤。從查詢移除任何不必要的篩選條件，以確保您的資料可以到達您的資料存放區。

為什麼我的資料集只會顯示__dt？

- 此欄由服務自動新增，並包含資料的大約擷取時間。它可用來最佳化您的查詢。如果您的資料集只包含此內容，請參閱上一個問題和回覆。

如何編寫由數據集完成驅動的事件？

- 您必須根據`describe-dataset`命令設定輪詢，以檢查具有特定時間戳記的資料集狀態是否已成功。

如何正確設定要使用的筆記本執行個體AWS IoT Analytics？

遵循以下步驟，確保您用來建立筆記本執行個體的 IAM 角色具有所需許可：

1. 移至 SageMaker 主控台並建立筆記本執行個體。
2. 填入詳細資訊，然後選擇 create a new role (建立新角色)。請記下角色 ARN。
3. 建立筆記本執行個體。這也會建立 SageMaker 可以使用的角色。
4. 前往 IAM 主控台並修改新建立的 SageMaker 角色。當您開啟該角色，應該會有一個受管政策。
5. 按一下 [新增內嵌原則]，選擇 [IoTAnalytics] 做為服務，然後在 [讀取權限] 下選取GetDatasetContent。

6. 檢閱政策、新增政策名稱，然後create (建立)政策。新建立的角色現在具有讀取資料集的原則權限AWS IoT Analytics。
7. 前往AWS IoT Analytics主控台並在筆記本執行個體中建立筆記本。
8. 等待筆記本執行個體處於「In Service」(服務中) 狀態。
9. 選擇 create notebooks (建立筆記本)，然後選擇您建立的筆記本執行個體。這會建立 Jupyter 筆記本，其中包含可存取資料集的所選範本。

為什麼我無法在執行個體中建立筆記本？

- 請務必使用正確的 IAM 政策來建立筆記本執行個體。(按照上一個問題中的步驟進行。)
- 確定筆記本執行個體處於「In Service」(服務中) 狀態。當您建立執行個體時，它會以「擱置中」狀態啟動。通常大約需要 5 分鐘時間，才會進入「In Service」(服務中) 狀態。如果筆記本執行個體在大約五分鐘後進入「失敗」狀態，請再次檢查權限。

為什麼我在亞馬遜上看不到我的數據集 QuickSight？

Amazon QuickSight 可能需要許可才能讀取您的AWS IoT Analytics資料集內容。若要授予許可，請依照下列步驟執行。

1. 在 Amazon 右上角選擇您的帳戶名稱， QuickSight 然後選擇「管理」 QuickSight。
2. 在左側導覽窗格中，選擇 [安全性和許可]。在 [QuickSight 存取AWS服務] 下方，確認存取權已授與給AWS IoT Analytics。

   a. 如果AWS IoT Analytics沒有存取權，請選擇 [新增] 或 [移除]。

   b. 選擇旁邊的核取方塊，AWS IoT Analytics然後選取 [更新]。如此一來，Amazon 就 QuickSight 可以讀取您的資料集內容。
3. 再次嘗試視覺化您的資料。

確保您為AWS IoT Analytics和亞馬遜選擇相同的AWS區域 QuickSight。否則，您可能會在存取AWS資源時遇到問題。如需支援區域的清單，請參閱AWS IoT AnalyticsAmazon Web Services 一般參考. QuickSight

為什麼我在現有的 Jupyter 筆記本上沒有看到容器化按鈕？

- 這是由於缺少AWS IoT Analytics容器化插件引起的。如果您在 2018 年 8 月 23 日之前建立 SageMaker筆記本執行個體，則需要依照容器化筆記本中的指示手動安裝外掛程式。
- 如果您在從AWS IoT Analytics主控台建立 SageMaker 筆記本執行個體或手動安裝之後，看不到 [容器化] 按鈕，請連絡AWS IoT Analytics技術支援。

為什麼我的容器化插件安裝失敗？

- 通常，由於 SageMaker筆記本實例中缺少權限，插件安裝失敗。有關筆記本執行個體的必要許可，請參閱許可，然後將必要的許可新增至筆記本執行個體角色。如果問題仍然存在，請從AWS IoT Analytics主控台建立新的筆記本執行個體。
- 如果在安裝插件期間出現，則可以放心地忽略日誌中的以下消息：「每次筆記本（或其他應用程序）加載時在瀏覽器中初始化此擴展。」

為什麼我的容器化插件拋出錯誤？

- 有多個原因會造成容器化失敗並產生錯誤。在容器化您的筆記本之前，請確認您使用正確的核心。容器化的核心會以「Containerized」字首開頭。
- 由於外掛程式會在 ECR 儲存庫建立和儲存 Docker 影像，請確認您的筆記本執行個體角色具有足夠的許可，以讀取、列出及建立 ECR 儲存庫。有關筆記本執行個體的必要許可，請參閱許可，然後將必要的許可新增至筆記本執行個體角色。
- 同時確認儲存庫的名稱符合 ECR 的規定。ECR 儲存庫名稱必須以字母開頭，並且只能包含小寫字母、數字、連字號、底線和斜線。
- 如果容器化程序失敗並顯示錯誤：「此執行個體沒有足夠的可用空間來執行容器化」，請嘗試使用較大的執行個體來解決問題。
- 如果您看到連線錯誤或映像建立錯誤，請再試一次。如果問題仍存在，請重新啟動執行個體並安裝最新版本的外掛程式。

為什麼我在容器化期間看不到我的變數？

- AWS IoT Analytics容器化外掛程式會在使用「容器化」核心執行筆記本之後，自動辨識筆記本中的所有變數。使用容器化核心之一來執行筆記本，然後執行容器化。

可以在容器增加哪些變數作為輸入？

- 您可以將您要在執行時間修改值的任何變數新增至您的容器做為輸入。這可讓您使用不同的參數來執行相同的容器，這些參數需要在建立資料集時提供。AWS IoT Analytics容器化 Jupyter 插件通過自動識別筆記本中的變量並將其作為容器化過程的一部分提供，從而簡化了此過程。

如何將容器輸出設置為後續分析的輸入？

- 每次執行您容器的資料集時，就會建立可存放已執行之成品的特定 S3 位置。若要存取此輸出位置，請在您的容器資料集中建立 `outputFileUriValue` 類型的變數。此變數的值應該是 S3 路徑，它用於存放額外的輸出檔。若要在後續執行中存取這些儲存的成品，您可以使用`getDatasetContent` API 並挑選後續執行所需的適當輸出檔案。

為什麼我的容器數據集失敗？

- 請確定您`executionRole`將正確的資料傳遞至容器資料集。的信任政策`executionRole`必須同時包含`iotanalytics.amazonaws.com`和`sagemaker.amazonaws.com`。
- 如果您看`AlgorithmError`到失敗的原因，請嘗試手動偵錯容器程式碼。如果容器程式碼有錯誤或執行角色沒有執行容器的許可，就會發生此錯誤。如果您使用AWS IoT Analytics Jupyter 外掛程式進行容器化，請建立與容器資料集之 ExecutionRole 相同角色的新 SageMaker 筆記本執行個體，然後嘗試手動執行筆記本。如果容器是在 Jupyter 外掛程式之外建立的，請嘗試手動執行程式碼，並限制對於 executionRole 的許可。

文件歷史紀錄

下表說明了對AWS IoT Analytics使用者指南2020 年 11 月 3 日後。如需獲得此文件所更新的詳細資訊，您可以訂閱 RSS 摘要。

變更	描述	日期
區域啟動	AWS IoT Analytics 現已在亞太地區 (孟買) 區域提供。	2021 年 8 月 18 日
查詢JOIN	此更新可讓您使用JOIN查詢AWS IoT Analytics資料集。	2021 年 7 月 27 日
與 AWS IoT SiteWise 整合	您現在可以使用AWS IoT Analytics查詢AWS IoT SiteWise資料。	2021 年 7 月 27 日
自訂分割區	AWS IoT Analytics現在通常支持根據消息屬性或通過管道活動添加的屬性對數據進行分區。	2021 年 6 月 14 日
重新處理通道訊息	此更新可讓您重新處理指定 Amazon S3 物件中的通道資料。	2020 年 12 月 15 日
Parquet 模式	AWS IoT Analytics資料倉庫現在支援實木複合地板檔案格式。	2020 年 12 月 15 日
使用 進行監控 CloudWatch 活動	AWS IoT Analytics自動將活動發佈到亞馬遜 CloudWatch 發生執行階段錯誤時的事件AWS Lambda活動。	2020 年 12 月 15 日
遲到資料通知	您可以使用此功能透過 Amazon 接收通知 CloudWatch 遲到數據到達時的事件。	2020 年 11 月 9 日

區域啟動	啟動AWS IoT Analytics中國(北京)。	2020 年 11 月 4 日

舊版更新

下表說明的重要變更AWS IoT Analytics使用者指南2020 年 11 月 4 日前。

變更	描述	日期
區域啟動	啟動AWS IoT Analytics在亞太區域 (Sydney) Region。	2020 年 7 月 16 日
更新	重組文件中)。	2020 年 5 月 7 日

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。