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 會自動執行從 IoT 裝置分析資料所需的步驟。在將 IoT 資料存放在時間序列資料存放區進行分析之前， 會先 AWS IoT Analytics 篩選、轉換和擴充 IoT 資料。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 透過與 Jupyter Notebook 的整合來實現進階資料探索。AWS IoT Analytics 也透過與 Quick Suite 的整合來實現資料視覺化。Quick Suite 可在下列區域使用。

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不結構化資料 （例如溫度、動作或聲音） 的裝置。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 資料通常只有在來自外部來源的其他資料內容中才有意義。 AWS IoT Analytics 可讓您解決這些問題，並收集大量裝置資料、處理訊息和存放它們。然後，您可以查詢資料並對其進行分析。 AWS IoT Analytics 包含常見 IoT 使用案例的預先建置模型，以便您可以回答問題，例如哪些裝置即將故障，或哪些客戶有捨棄其可穿戴裝置的風險。

如何使用 AWS IoT Analytics

下圖顯示如何使用 的概觀 AWS IoT Analytics。

主要功能

收集

- 與 整合 AWS IoT Core-AWS IoT Analytics 與 完全整合， AWS IoT Core 因此可以在連線裝置傳入時接收訊息。

- 使用批次 API 從任何來源新增資料 -AWS IoT Analytics 可以透過 HTTP 從任何來源接收資料。這表示連線至網際網路的任何裝置或服務都可以傳送資料至 AWS IoT Analytics。如需詳細資訊，請參閱《 AWS IoT Analytics API 參考》中的 BatchPutMessage。
- 僅收集您要存放和分析的資料 - 您可以使用 AWS IoT Analytics 主控台，透過各種格式和頻率的 MQTT 主題篩選條件 AWS IoT Analytics ，設定 從裝置接收訊息。 會 AWS IoT Analytics 驗證資料是否在您定義和建立頻道的特定參數內。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。

流程

- 清理和篩選 -AWS IoT Analytics 可讓您定義在 AWS IoT Analytics 偵測到遺失資料時觸發的 AWS Lambda 函數，因此您可以執行程式碼來估計和填補差距。您也可以定義最大和最小篩選條件和百分位數閾值，以移除資料中的極端值。
- 轉換 -AWS IoT Analytics 可以使用您定義的數學或條件邏輯轉換訊息，以便您可以執行常見的計算，例如攝氏轉換為華氏。
- 豐富 -AWS IoT Analytics 可以使用天氣預測等外部資料來源豐富資料，然後將資料路由到 AWS IoT Analytics 資料存放區。

存放

- 時間序列資料存放區 — 將裝置資料AWS IoT Analytics 存放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。
- 存放已處理和原始資料 -AWS IoT Analytics 存放已處理的資料，也會自動存放原始擷取的資料，以便稍後處理。

分析

- 執行臨機操作 SQL 查詢 -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，例如連線車輛機群的平均行駛距離，或智慧型建築物中有多少門在晚上 7 點後鎖定。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的效能，並了解裝置使用方式和位置、持續監控裝置資料以預測維護問題，以及監控感應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 筆記本中託管筆記本的支援，用於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撰寫的機器學習模型和視覺化。您可以使用 範本來開始使用與裝置故障分析相關的 IoT 使用案例、預測低用量等事件，這些事件可能表示客戶會捨棄產品，或依客戶用量層級 （ 例如重度使用者、週末使用

者）或裝置運作狀態來分割裝置。撰寫筆記本之後，您可以按照指定的排程進行容器化和執行。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在排程中並執行。

建置和視覺化

- Quick Suite 整合 -AWS IoT Analytics 提供 Quick Suite 的連接器，讓您可以在 QuickSight 儀表板中視覺化資料集。
- 主控台整合 - 您也可以在 AWS IoT Analytics主控台的內嵌 Jupyter 筆記本中視覺化結果或臨機操作分析。

AWS IoT Analytics 元件和概念

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 BatchPutMessage API 直接傳送訊息至頻道。未處理的訊息會存放在您或 AWS IoT Analytics 管理的 Amazon Simple Storage Service (Amazon S3) 儲存貯體中。

管道

管道會取用來自頻道的訊息，並可讓您在將訊息存放在資料存放區之前進行處理。處理步驟稱為活動 (管道活動)，對您的訊息執行轉換，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。如同未處理的頻道訊息，資料存放區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。

資料集完成後，您可以使用 Quick Suite 透過整合來探索和深入了解資料。您也可以透過與 Jupyter Notebook 整合來執行更進階的分析功能。Jupyter 筆記本提供強大的資料科學工具，可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至 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進行存取。您也可以將資料集內容做為輸入傳送到 AWS IoT Events，此服務可讓您監控裝置或程序操作失敗或變更，並在發生此類事件時觸發其他動作。

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。觸發可以是時間間隔（例如，每兩小時建立此資料集）或建立另一個資料集的內容時（例如，`myOtherDataset`在完成建立其內容時建立此資料集）。或者，您可以使用 CreateDatasetContent API 手動產生資料集內容。

Docker 容器

您可以建立自己的 Docker 容器來封裝分析工具，或使用 SageMaker AI 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 容器來封裝分析工具，或使用 SageMaker AI 提供的選項。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。Docker 容器能夠執行與 Matlab、Octave、Wise.io、SPSS、R、Fortran、Python、Scala、Java、C++ 等一起準備的自訂分析程式碼。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。Delta 視窗可讓您使用自上次分析以來抵達資料存放區的新資料來建立資料集內容，並對其進行分析。您可以透過在資料集的 `filters`部分`deltaTime`中設定 `queryAction`來建立差異視窗。如需詳細資訊，請參閱 `CreateDataset` API。通常，您還需要設定時間間隔觸發 () 來自動建立資料集內

容`triggers:schedule:expression`。這可讓您篩選在特定時段內到達的訊息，因此前一個時段的訊息中包含的資料不會計算兩次。如需詳細資訊，請參閱範例 6 -- 使用 Delta 視窗 (CLI) 建立 SQL 資料集。

存取 AWS IoT Analytics

作為 的一部分 AWS IoT， AWS IoT Analytics 提供下列界面，讓您的裝置能夠產生資料，而您的應用程式可以與其產生的資料互動：

AWS Command Line Interface (AWS CLI)

在 Windows、OS X 和 Linux AWS IoT Analytics 上執行 的 命令。這些命令可讓您建立和管理物件、憑證、規則和政策。若要開始使用，請參閱《AWS Command Line Interface 使用者指南》。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 `aws iotanalytics`命令與 互動 AWS IoT Analytics。使用 `aws iot`命令與 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 動作可讓您建立和管理物件、憑證、規則和政策。如需詳細資訊，請參閱 AWS IoT API 參考中的動作。

AWS SDKs

使用特定語言 APIs建置您的 AWS IoT Analytics 應用程式。這些 SDKs會包裝 HTTP 和 HTTPS API，並可讓您使用任何支援的語言進行程式設計。如需詳細資訊，請參閱 AWS 開發套件與工具。

AWS IoT 裝置 SDKs

建置在您的裝置上執行並傳送訊息的應用程式 AWS IoT Analytics。如需詳細資訊，請參閱 AWS IoT 開發套件。

AWS IoT Analytics 主控台

您可以建置元件以在 AWS IoT Analytics 主控台中視覺化結果。

使用案例

預測性維護

AWS IoT Analytics 提供 範本來建置預測性維護模型，並將其套用至您的裝置。例如，您可以使用 AWS IoT Analytics 來預測連網貨運車輛上的供暖和冷卻系統何時可能會故障，以便車輛可以重新路由以防止貨物損壞。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。

主動補充供應品

AWS IoT Analytics 可讓您建置 IoT IoT 應用程式，以即時監控庫存。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

透過 AWS IoT Analytics，您可以建置 IoT 應用程式，持續監控不同程序的效率，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。透過 AWS IoT Analytics，公司可以識別一段時間內位置或卡車最有效率的負載，然後即時比較與目標負載的任何偏差，並更好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 登錄檔資料或公有資料來源，以內容中繼資料豐富 IoT 裝置資料，讓您的分析在時間、位置、溫度、海拔高度和其他環境條件上具有因素。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，為了判斷何時供水，灌溉系統可能會使用降雨資料來充實濕度感應器資料，從而實現更有效率的用水量。

AWS IoT Analytics 終止支援

在仔細考慮之後，我們決定終止對 的支援 AWS IoT Analytics，自 2025 年 12 月 15 日起。 自 2024 年 7 月 24 日起 AWS IoT Analytics 將不再接受新客戶。身為在 2024 年 7 月 23 日之前註冊服務的現有客戶，您可以繼續使用 AWS IoT Analytics 功能。2025 年 12 月 15 日之後，您將無法再使用 AWS IoT Analytics。

隨著 2025 年 12 月 15 日 AWS IoT Analytics 接近end-of-service，客戶務必了解其遷移選項。此頁面提供 主要功能的概觀， AWS IoT Analytics 並將其映射至用於複寫功能的替代 AWS 服務。透過了解這些替代服務的功能，客戶可以規劃和執行順暢的遷移，確保其 AWS IoT 資料分析工作流程持續不間斷。

主題

遷移選項

考慮從 遷移時 AWS IoT Analytics，請務必了解此轉移背後的優點和原因。下表提供替代選項和現有 AWS IoT Analytics 功能的映射。

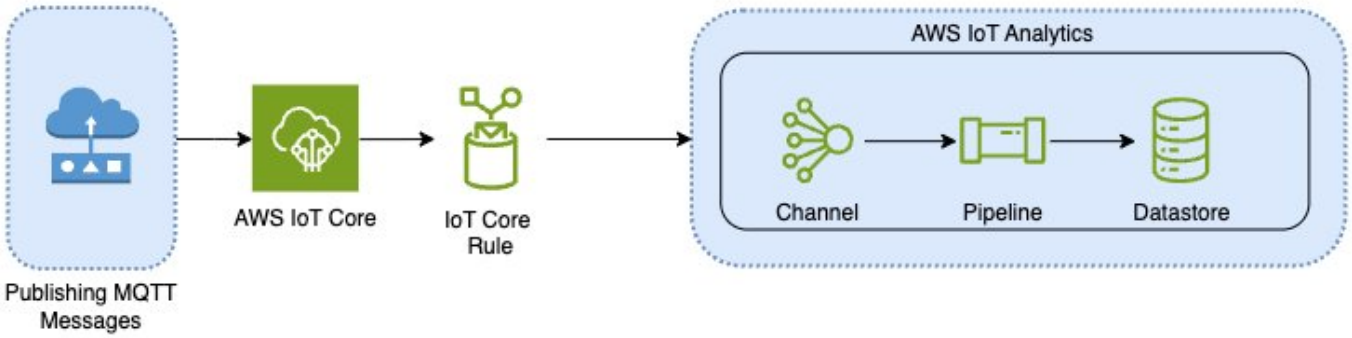
Action	AWS IoT Analytics	替代服務	原因
收集	AWS IoT Analytics 可讓您輕鬆地使用 `BatchPutMessage` API 直接從 AWS IoT Core 或其他來源擷取資料。此整合可確保資料從您的裝置無縫流向分析平台。	• Amazon Kinesis Data Streams • Amazon Data Firehose	Amazon Kinesis Data Streams 提供強大的解決方案。Kinesis 可即時串流資料，實現即時處理和分析，這對需要即時洞察和異常偵測的應用程式至關重要。 Amazon Data Firehose 可在串流資料落入 Amazon S3 之前，簡化擷取和轉換

Action	AWS IoT Analytics	替代服務	原因
			串流資料的程序，並自動擴展以符合您的資料輸送量。
流程	在 中處理資料 AWS IoT Analytics 涉及清理、篩選、轉換，以及使用外部來源來充實資料。	• Amazon Managed Service for Apache Flink • Amazon Data Firehose	Amazon Managed Service for Apache Flink 支援複雜的事件處理，例如模式比對和彙總，這對複雜的 AWS IoT Analytics 案例至關重要。 Amazon Data Firehose 可處理更簡單的轉換，並可叫用 AWS Lambda 函數進行自訂處理，提供彈性，而無須複雜的 Flink。
存放	AWS IoT Analytics 使用針對資料最佳化的時間序列 AWS IoT 資料存放區，其中包括資料保留政策和存取管理等功能。	• Simple Storage Service (Amazon Simple Storage Service (Amazon S3)) • Amazon Timestrea m	Amazon S3 提供可擴展、耐用且經濟實惠的儲存解決方案。Amazon S3 與其他 AWS 服務的整合，使其成為長期儲存和分析大量資料集的理想選擇。 Amazon Timestream 是專門建置的時間序列資料庫。您可以從 Amazon S3 批次載入資料。

Action	AWS IoT Analytics	替代服務	原因
分析	AWS IoT Analytics 提供內建的 SQL 查詢功能、時間序列分析和託管 Jupyter 筆記本的支援，讓您輕鬆執行進階分析和機器學習。	• AWS Glue • Amazon Athena	AWS Glue 簡化 ETL 程序，讓您輕鬆擷取、轉換和載入資料，同時提供與 Athena 整合的資料目錄，以利查詢。 Amazon Athena 可讓您直接在存放在 Amazon S3 中的資料上執行 SQL 查詢，而不需要管理任何基礎設施，從而進一步執行此操作。
視覺化	AWS IoT Analytics 與 Quick Suite 整合，可建立豐富的視覺化效果和儀表板。	• Amazon Quick Suite	根據您決定使用的替代資料存放區繼續使用 Quick Suite，例如 Amazon S3。

遷移指南

在目前的架構中， AWS IoT 資料 AWS IoT Analytics 會透過 AWS IoT Core 規則從 流 AWS IoT Core 向 。 AWS IoT Analytics 會處理擷取、轉換和儲存。

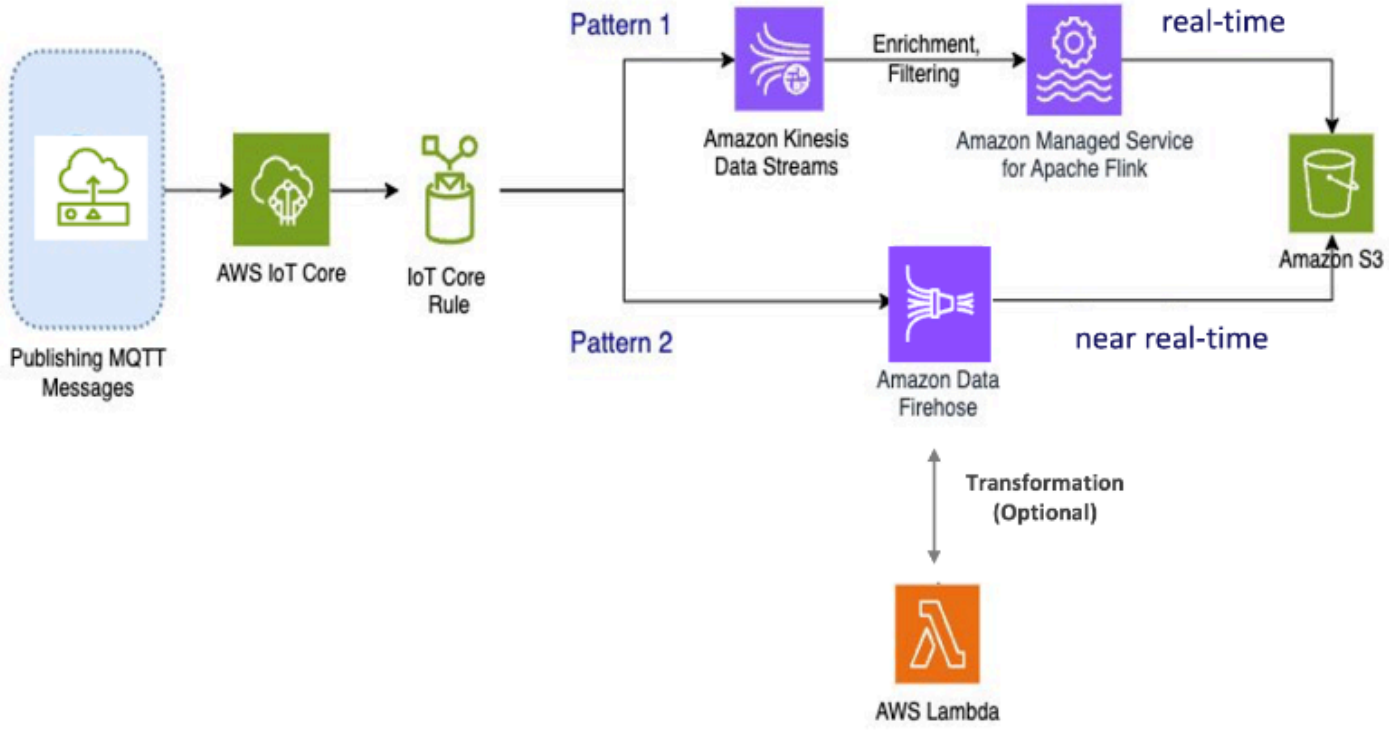


若要完成遷移，請遵循兩個步驟：

主題

步驟 1：重新導向持續的資料擷取

遷移的第一步是將進行中的資料擷取重新導向至新服務。我們根據您的特定使用案例建議兩種模式：



模式 1：Amazon Kinesis Data Streams with Amazon Managed Service for Apache Flink

在此模式中，您會先發佈與 Amazon Kinesis Data Streams 整合的資料 AWS IoT Core ，以便即時收集、處理和分析大量資料頻寬。

指標和分析

1. 擷取資料： AWS IoT 資料會即時擷取至 Amazon Kinesis Data Streams。Amazon Kinesis Data Streams 可以處理數百萬部 AWS IoT 裝置的高傳輸量資料，實現即時分析和異常偵測。

2. 處理資料：使用 Amazon Managed Service for Apache Flink 從 Amazon Kinesis Data Streams 處理、擴充和篩選資料。Flink 為複雜的事件處理提供了強大的功能，例如彙總、聯結和暫時操作。
3. 儲存資料：Flink 會將處理的資料輸出至 Amazon S3 以進行儲存和進一步分析。然後，您可以使用 Amazon Athena 查詢這些資料，或與其他 AWS 分析服務整合。

如果您的應用程式涉及高頻寬串流資料，且需要進階處理，例如模式比對或視窗調整，則使用此模式是最合適的。

模式 2：使用 Amazon Data Firehose

在此模式中，資料會發佈至 AWS IoT Core，其與 Amazon Data Firehose 整合，可讓您直接將資料存放在 Amazon S3 中。此模式也支援使用 進行基本轉換 AWS Lambda。

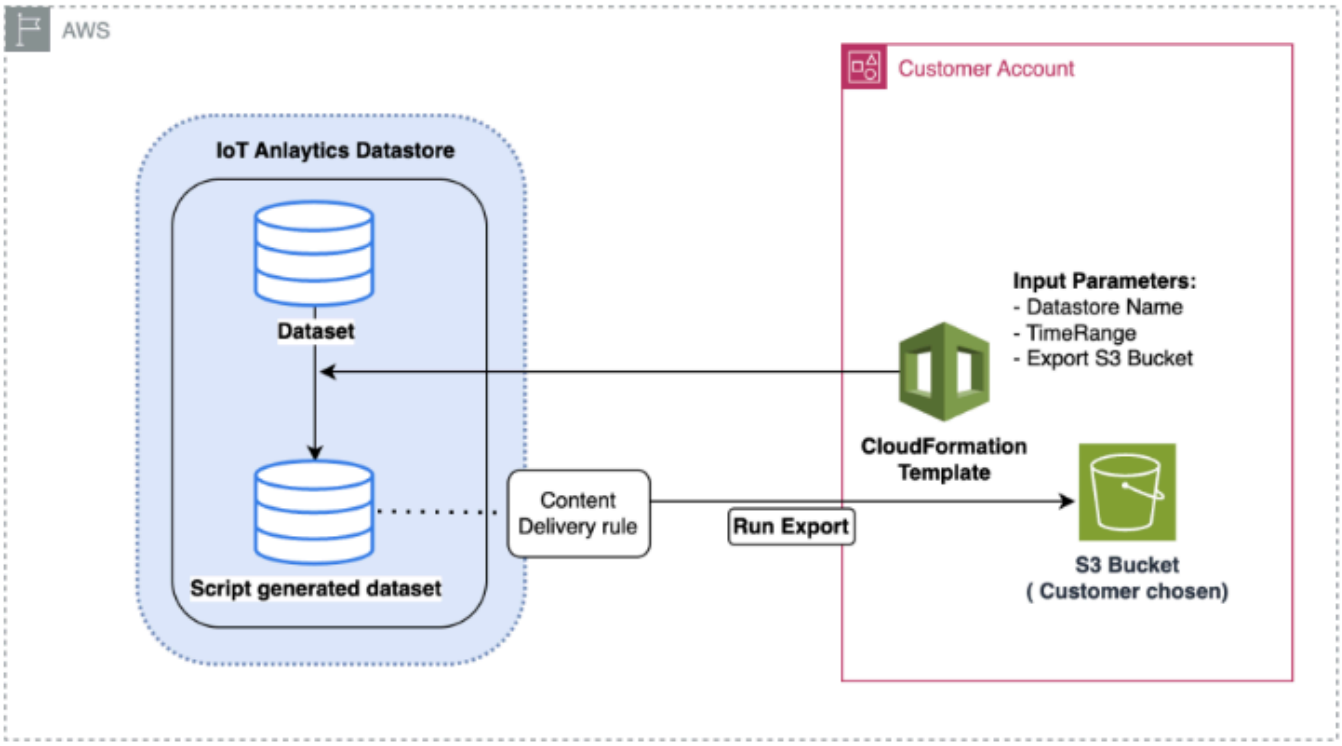
指標和分析

1. 擷取資料： AWS IoT 資料會直接從您的裝置擷取或擷取 AWS IoT Core 至 Amazon Data Firehose。
2. 程序資料：Amazon Data Firehose 會對資料執行基本轉換和處理，例如格式轉換和擴充。您可以設定 Firehose 資料轉換以叫用 AWS Lambda 函數來轉換傳入的來源資料，然後再將其交付至目的地。
3. 存放資料：處理的資料會以近乎即時的方式交付至 Amazon S3。Amazon Data Firehose 會自動擴展以符合傳入資料的輸送量，以確保可靠且有效率的資料交付。

將此模式用於需要基本轉換和處理的工作負載。此外，Amazon Data Firehose 透過為存放在 Amazon S3 中的資料提供資料緩衝和動態分割功能，簡化程序。

步驟 2：匯出先前擷取的資料

對於先前擷取並存放於 的資料 AWS IoT Analytics，您需要將其匯出至 Amazon S3。若要簡化此程序，您可以使用 CloudFormation 範本來自動化整個資料匯出工作流程。您可以使用指令碼進行部分（時間範圍型） 資料擷取。



CloudFormation 將資料匯出至 Amazon S3 的 範本

上圖說明使用 CloudFormation 範本在相同資料 AWS IoT Analytics 存放區中建立資料集的程序，以根據時間戳記進行選擇。這可讓使用者在所需的時間範圍內擷取特定資料點。此外，還會建立內容交付規則，將資料匯出至 Amazon S3 儲存貯體。

以下程序說明步驟。

1. 準備 CloudFormation 範本並將其儲存為 YAML 檔案。例如 `migrate-datasource.yaml`。

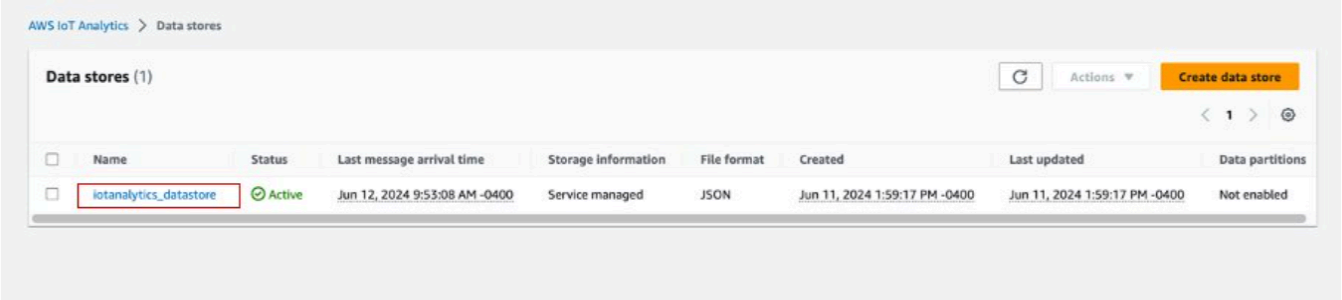
```
# Cloudformation Template to migrate an AWS IoT Analytics datastore to an external
 dataset
AWSTemplateFormatVersion: 2010-09-09
Description: Migrate an AWS IoT Analytics datastore to an external dataset
Parameters:
  DatastoreName:
    Type: String
    Description: The name of the datastore to migrate.
    AllowedPattern: ^[a-zA-Z0-9_]+$
  TimeRange:
    Type: String
```

```
    Description: |
      This is an optional argument to split the source data into multiple files.
      The value should follow the SQL syntax of WHERE clause.
      E.g. WHERE DATE(Item_TimeStamp) BETWEEN '09/16/2010 05:00:00' and '09/21/2010
 09:00:00'.
    Default: ''
  MigrationS3Bucket:
    Type: String
    Description: The S3 Bucket where the datastore will be migrated to.
    AllowedPattern: (?!(^xn--|.+-s3alias$))^[a-z0-9][a-z0-9-]{1,61}[a-z0-9]$
  MigrationS3BucketPrefix:
    Type: String
    Description: The prefix of the S3 Bucket where the datastore will be migrated
 to.
    Default: ''
    AllowedPattern: (^([a-zA-Z0-9.\-_]*\/)*$)|(^$)
Resources:
  # IAM Role to be assumed by the AWS IoT Analytics service to access the external
 dataset
  DatastoreMigrationRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: iotanalytics.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: AllowAccessToExternalDataset
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - s3:GetBucketLocation
                  - s3:GetObject
                  - s3:ListBucket
                  - s3:ListBucketMultipartUploads
                  - s3:ListMultipartUploadParts
                  - s3:AbortMultipartUpload
                  - s3:PutObject
                  - s3:DeleteObject
```

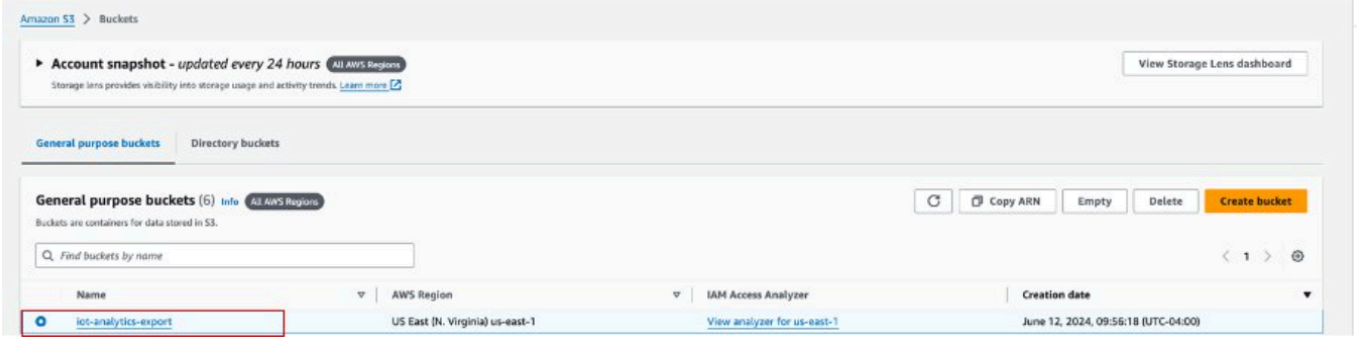
```
                Resource:
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}/
${MigrationS3BucketPrefix}*

  # This dataset that will be created in the external S3 Export
  MigratedDataset:
    Type: AWS::IoTAnalytics::Dataset
    Properties:
      DatasetName: !Sub ${DatastoreName}_generated
      Actions:
        - ActionName: SqlAction
          QueryAction:
            SqlQuery: !Sub SELECT * FROM ${DatastoreName} ${TimeRange}
      ContentDeliveryRules:
        - Destination:
            S3DestinationConfiguration:
              Bucket: !Ref MigrationS3Bucket
              Key: !Sub ${MigrationS3BucketPrefix}${DatastoreName}/!
{iotanalytics:scheduleTime}/!{iotanalytics:versionId}.csv
              RoleArn: !GetAtt DatastoreMigrationRole.Arn
      RetentionPeriod:
        Unlimited: true
      VersioningConfiguration:
        Unlimited: true
```

2. 決定需要匯出資料的資料 AWS IoT Analytics 存放區。針對本指南，我們將使用名為 的範例資料存放區`iot_analytics_datastore`。

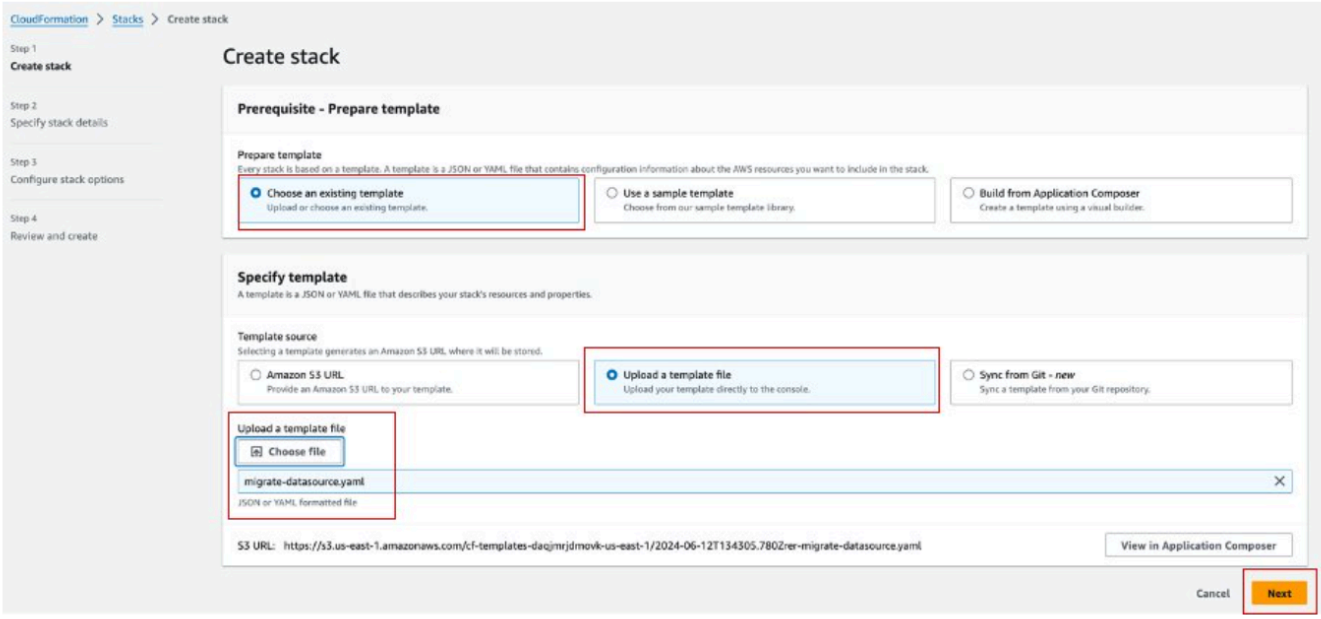


3. 建立或識別要匯出資料的 Amazon S3 儲存貯體。針對本指南，我們將使用 儲存`iot-analytics-export`貯體。



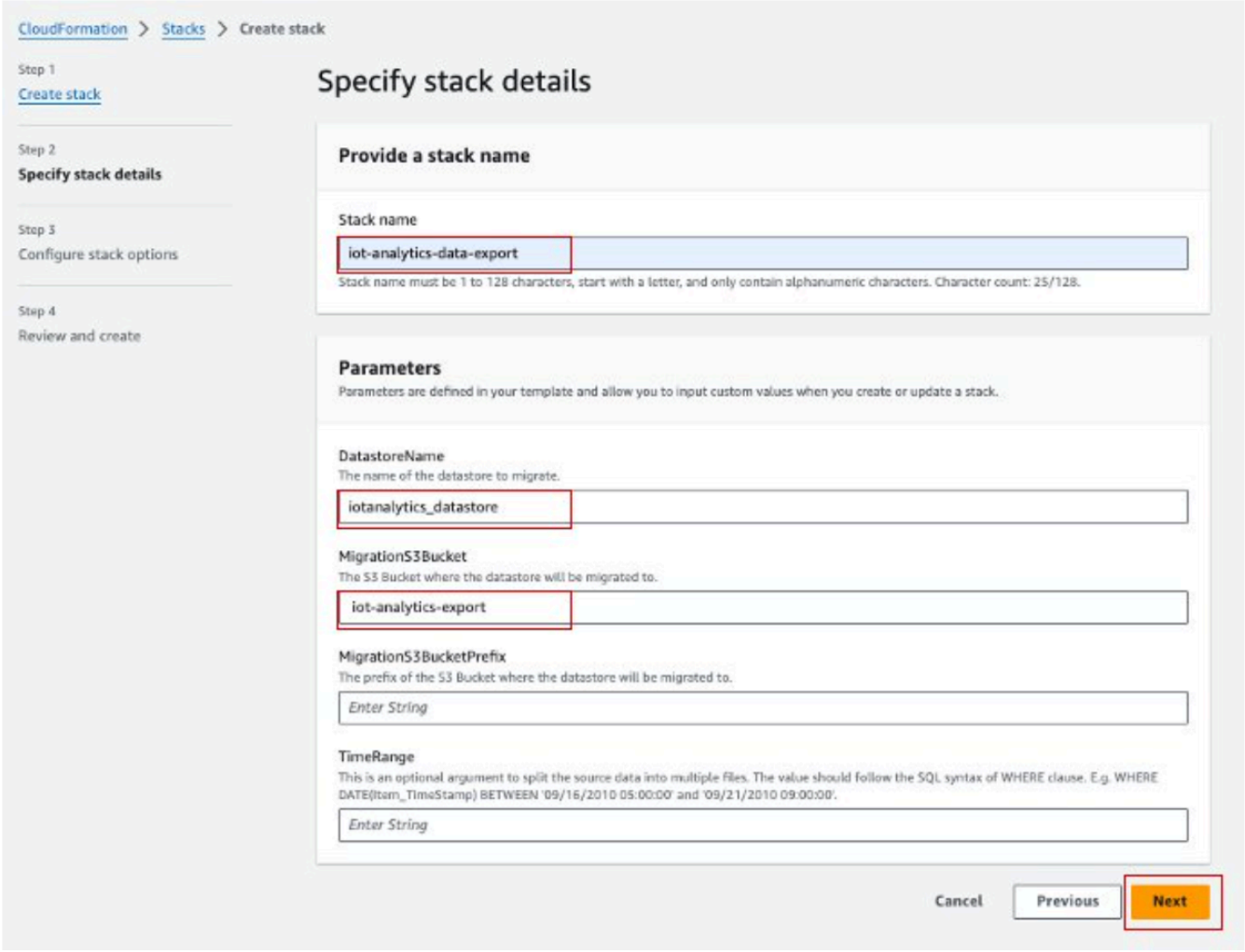
4. 建立 CloudFormation 堆疊。

   - 導覽至 https：//https://console.aws.amazon.com/cloudformation。
   - 按一下建立堆疊，然後選取使用新資源 （標準）。
   - 上傳 `migrate-datasource.yaml` 檔案。

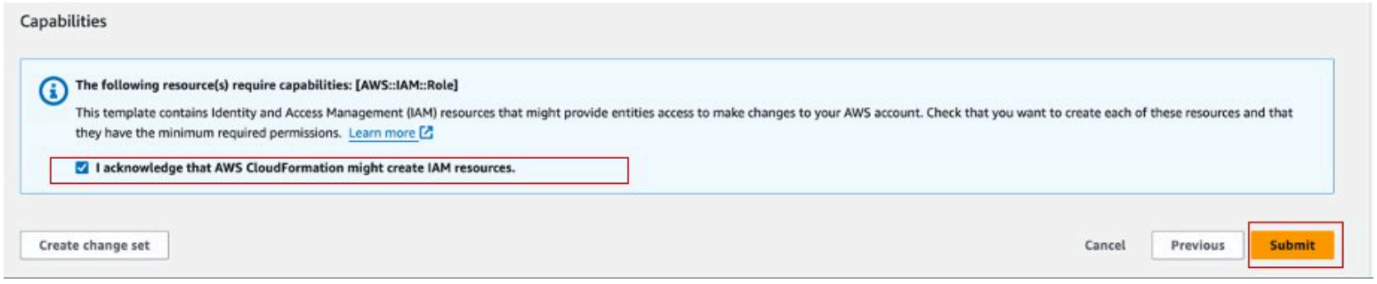


5. 輸入堆疊名稱並提供下列參數：

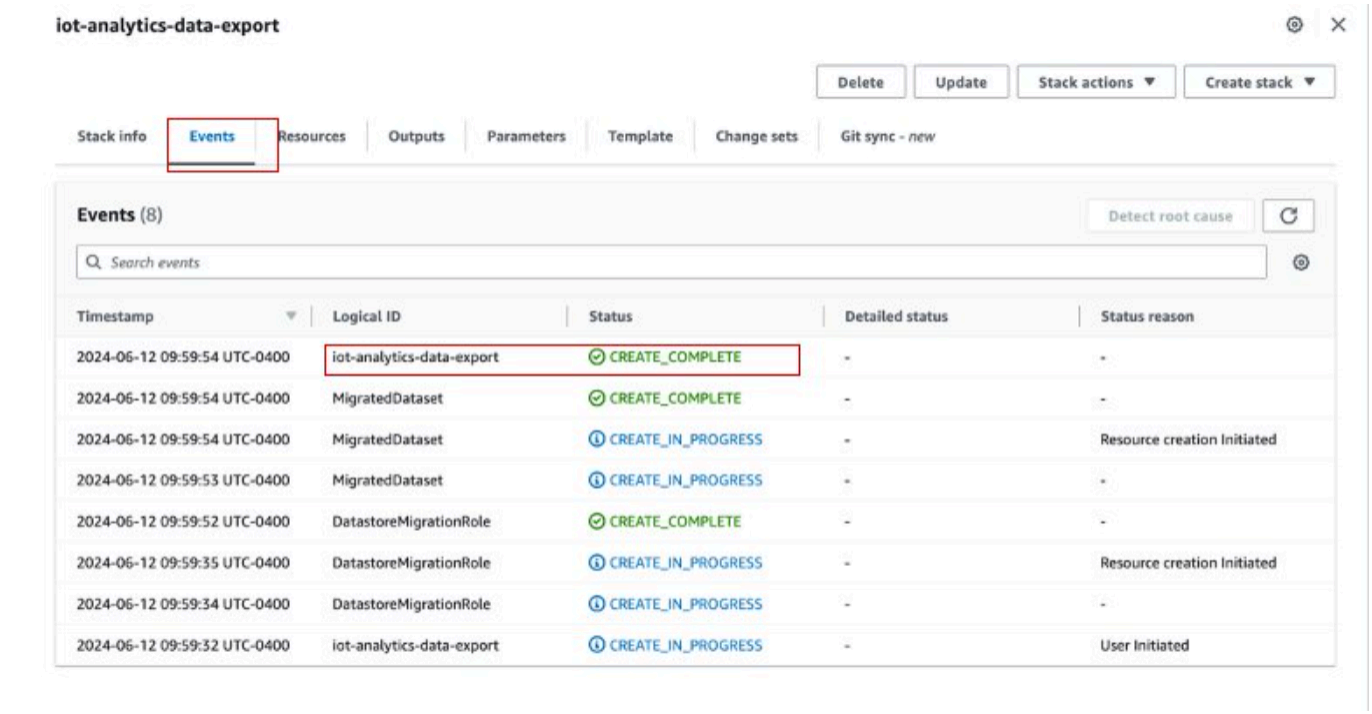
   - DatastoreName：您要遷移 AWS IoT Analytics 的資料存放區名稱。
   - MigrationS3Bucket：存放遷移資料的 Amazon S3 儲存貯體。
   - MigrationS3BucketPrefix （選用） ：Amazon S3 儲存貯體的字首。
   - TimeRange （選用） ：用於篩選匯出資料的SQL `WHERE`子句，允許根據指定的時間範圍將來源資料分割成多個檔案。



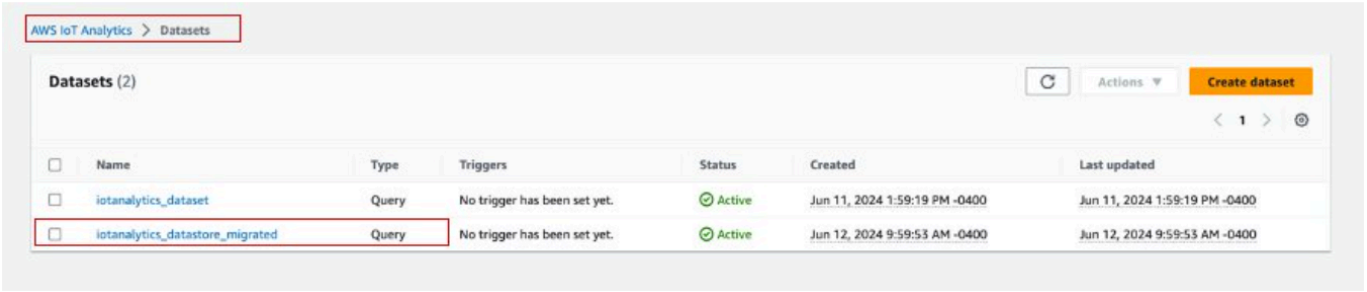
6. 在設定堆疊選項畫面上按一下下一步。
7. 選取核取方塊以確認建立 IAM 資源，然後按一下提交。



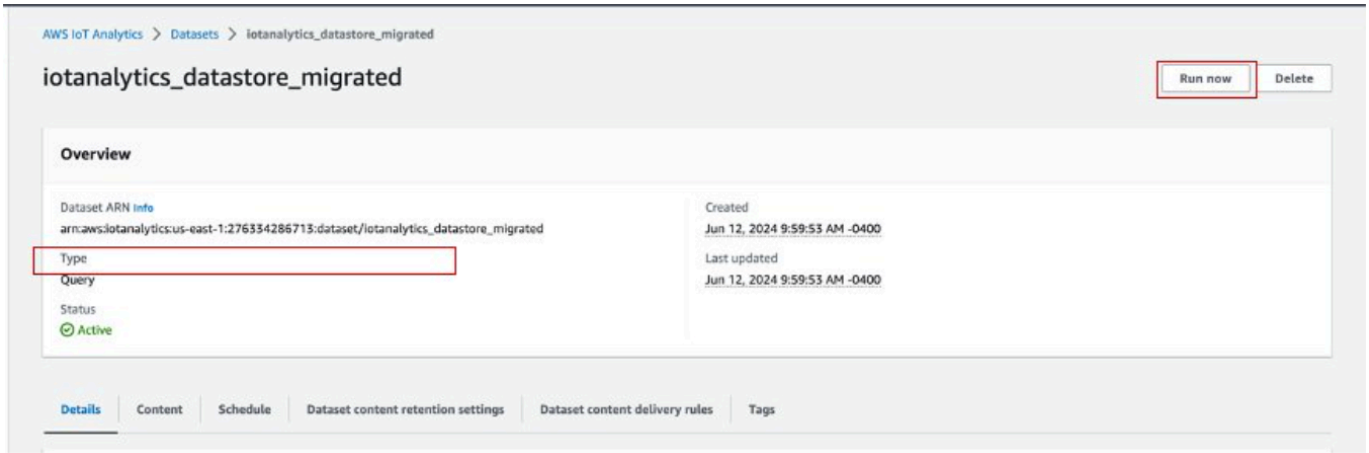
8. 檢閱事件索引標籤上的堆疊建立是否完成。



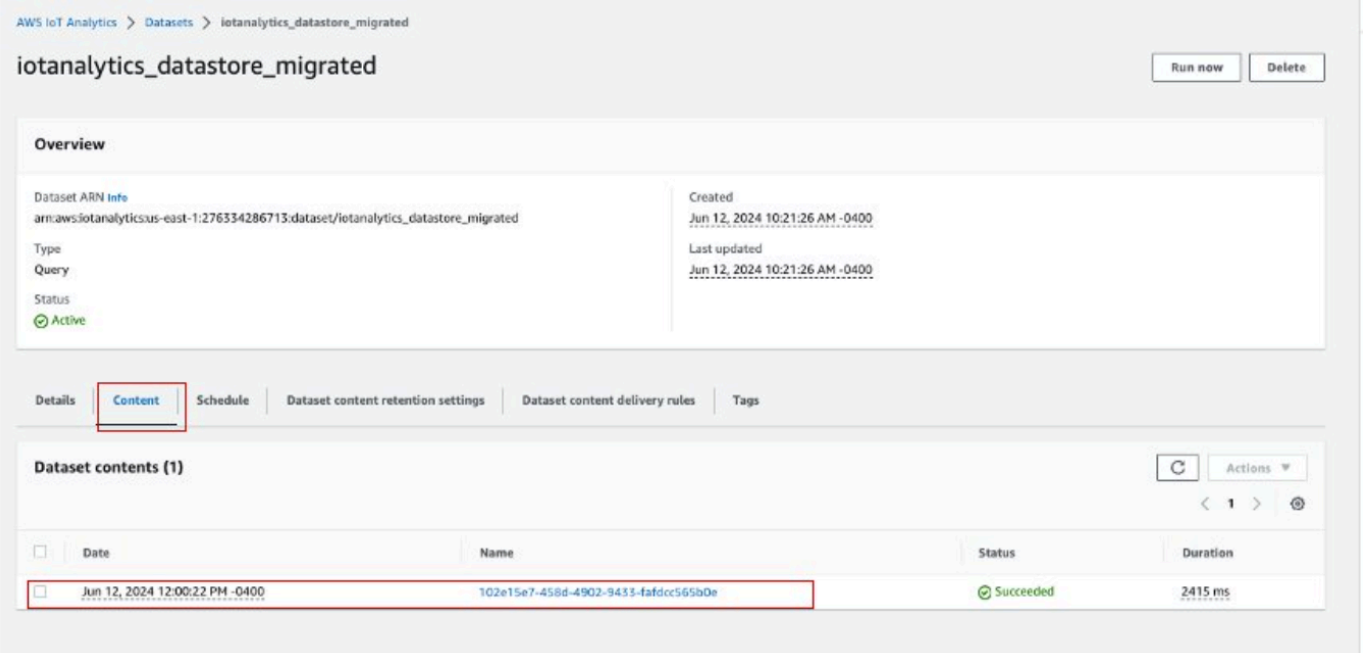
9. 成功完成堆疊後，導覽至 AWS IoT Analytics → 資料集以檢視遷移的資料集。



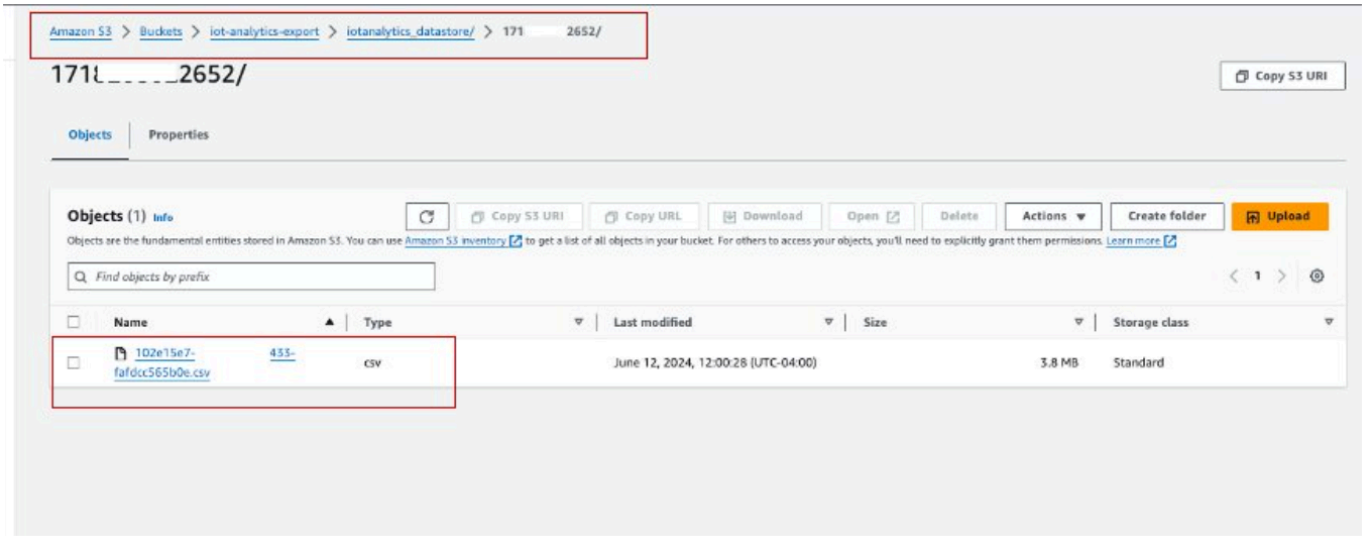
10. 選取產生的資料集，然後按一下立即執行以匯出資料集。



11. 您可以在資料集的內容索引標籤上檢視內容。

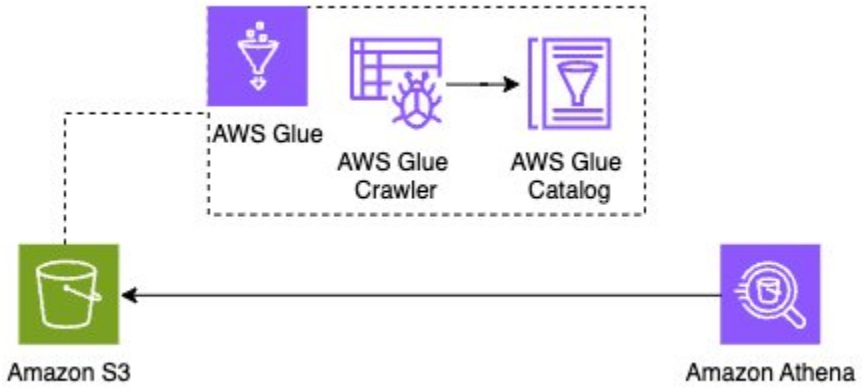


12. 最後，在 Amazon S3 主控台中開啟 iot-analytics-export 儲存貯體來檢閱匯出的內容。



執行兩種模式的隨需查詢

當您將 AWS IoT Analytics 工作負載遷移至 Amazon Kinesis Data Streams 或 Amazon Data Firehose 時，利用 AWS Glue 和 Amazon Athena 可進一步簡化資料分析程序。 AWS Glue 可簡化資料準備和轉換，而 Amazon Athena 則可快速無伺服器地查詢您的資料。它們共同提供強大、可擴展且經濟實惠的解決方案，用於分析 AWS IoT 資料。



Summary

將 AWS IoT Analytics 工作負載從 遷移 AWS IoT Analytics 至 Amazon Kinesis Data Streams、Amazon S3，並增強您處理大規模複雜 AWS IoT 資料的能力。此架構提供可擴展、耐用的儲存體和強大的分析功能，可讓您即時從 IoT 資料獲得更深入的洞見。

清除使用 建立的資源對於避免遷移完成後的意外成本 CloudFormation 至關重要。

如需資料遷移所涉及的成本，請參閱 AWS IoT Analytics 定價頁面。完成後，請考慮刪除新建立的資料集，以避免任何不必要的費用。

完整資料集匯出：若要在沒有以時間為基礎的分割的情況下匯出完整資料集，您也可以使用 AWS IoT Analytics 主控台並相應地設定內容交付規則。

遵循遷移指南，您可以順暢地轉換資料擷取和處理管道，確保持續且可靠的資料流程。利用 AWS Glue 和 Amazon Athena 可進一步簡化資料準備和查詢，讓您執行複雜的分析，而無需管理任何基礎設施。

這種方法可讓您有效地擴展 AWS IoT Analytics 工作，讓您更輕鬆地適應業務不斷增長的需求，並從 AWS IoT 資料中擷取最大價值。

AWS IoT Analytics （主控台） 入門

使用此教學課程來建立您需要 AWS IoT Analytics 的資源 （也稱為元件），以探索 IoT IoT 裝置資料的實用洞見。

> **備註**
>
> - 如果您在下列教學課程中輸入大寫字元， AWS IoT Analytics 會自動將它們變更為小寫。
> - AWS IoT Analytics 主控台具有一鍵式入門功能，可用來建立頻道、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。
>   - 本教學課程會逐步引導您建立 AWS IoT Analytics 資源的每個步驟。

請依照下列指示建立 AWS IoT Analytics 頻道、管道、資料存放區和資料集。本教學課程也說明如何使用 AWS IoT Core 主控台來傳送要導入的訊息 AWS IoT Analytics。

主題

登入 AWS IoT Analytics 主控台

若要開始使用，您必須擁有 AWS 帳戶。如果您已有 AWS 帳戶，請導覽至 https：//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 帳戶

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

   部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

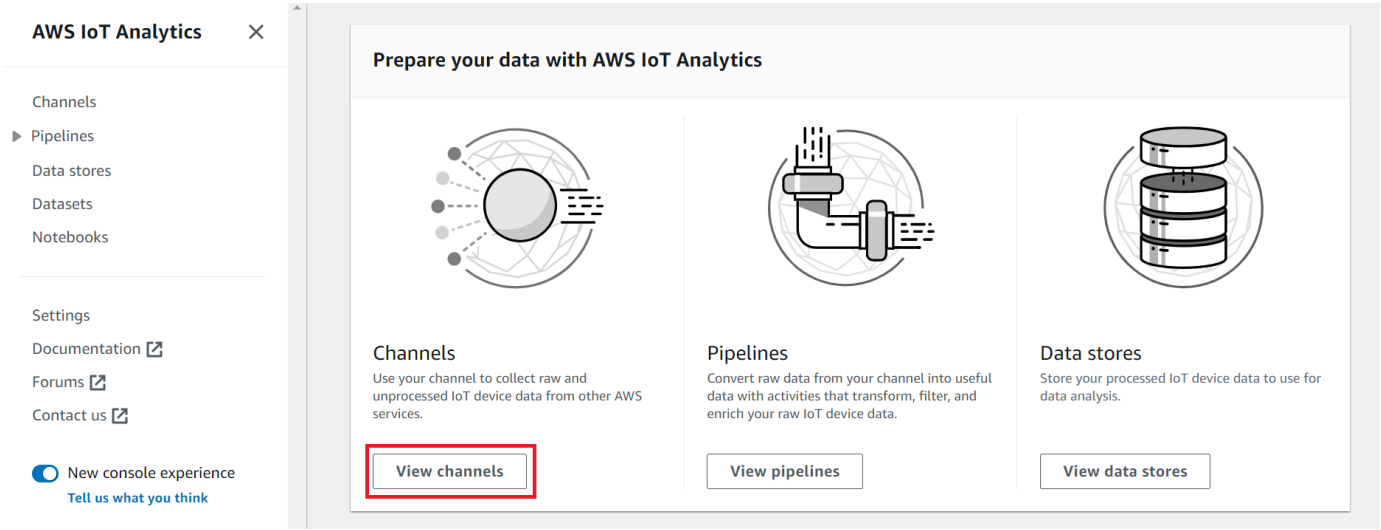
   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。
3. 登入 AWS 管理主控台 並導覽至 https：//https://console.aws.amazon.com/iotanalytics/。

建立頻道

頻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您的頻道。

建立頻道

1. 在 https://console.aws.amazon.com/iotanalytics/ 的「使用 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. 在指定頻道詳細資訊頁面上，輸入頻道的詳細資訊。

   a. 輸入唯一且您可以輕鬆識別的頻道名稱。

   b. （選用）針對標籤，將一或多個自訂標籤（索引鍵/值對）新增至您的頻道。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 會將未處理的原始 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，您可以存取和管理，也可以為您 AWS IoT Analytics 管理 Amazon S3 儲存貯體。

   a. 在本教學課程中，針對儲存類型，選擇服務受管儲存。

   b. 針對選擇儲存原始資料的時間長度，選擇無限期。

   c. 選擇下一步。
5. 在設定來源頁面上，輸入 的資訊 AWS IoT Analytics 以收集訊息資料 AWS IoT Core。

   a. 輸入 AWS IoT Core 主題篩選條件，例如 `update/environment/dht1`。在本教學課程稍後，您將使用此主題篩選條件將訊息資料傳送到您的頻道。

   b. 在 IAM 角色區域中，選擇建立新角色。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政策的角色。

   c. 選擇下一步。
6. 檢閱您的選擇，然後選擇建立頻道。
7. 確認您的新頻道顯示在頻道頁面上。

建立資料存放區

資料存放區會接收和存放您的訊息資料。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。您可以針對來自不同裝置或位置的訊息使用多個資料存放區。或者，您可以根據您的管道組態和需求來篩選訊息資料。

請依照下列步驟來建立資料存放區。

建立資料存放區

1. 在 https://console.aws.amazon.com/iotanalytics/ 的「使用 準備資料」 AWS IoT Analytics區段中，選擇檢視資料存放區。

2. 在資料存放區頁面上，選擇建立資料存放區。
3. 在指定資料存放區詳細資訊頁面上，輸入資料存放區的基本資訊。

   a. 針對資料存放區 ID，輸入唯一的資料存放區 ID。您無法在建立後變更此 ID。

   b. （選用）針對標籤，選擇新增標籤，將一或多個自訂標籤（鍵/值對）新增至您的資料存放區。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，指定如何儲存您的資料。

   a. 針對儲存類型，選擇服務受管儲存。

   b. 針對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 Parquet 檔案格式。針對資料存放區資料格式，選擇 JSON 或 Parquet。如需 AWS IoT Analytics 支援檔案類型的詳細資訊檔案格式，請參閱 。

   選擇下一步。
6. （選用）AWS IoT Analytics 支援資料存放區中的自訂分割區，因此您可以查詢已刪除的資料以改善延遲。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. 確認您的新資料存放區顯示在資料存放區頁面上。

建立管道

您必須建立管道，將頻道連線至資料存放區。基本管道只會指定收集資料的頻道，並識別訊息傳送到的資料存放區。如需詳細資訊，請參閱管道活動。

在本教學課程中，您會建立僅將頻道連線至資料存放區的管道。稍後，您可以新增管道活動來處理此資料。

請依照下列步驟建立管道。

建立管道

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

> **Tip**
>
> 您也可以從導覽窗格中選擇管道。

2. 在管道頁面上，選擇建立管道。
3. 輸入管道的詳細資訊。
   a. 在設定管道 ID 和來源中，輸入管道名稱。
   b. 選擇管道的來源，這是管道將從中讀取訊息的 AWS IoT Analytics 管道。
   c. 指定管道的輸出，這是存放已處理訊息資料的資料存放區。
   d. （選用）對於標籤，將一或多個自訂標籤（索引鍵/值對）新增至管道。
   e. 在推斷訊息屬性頁面上，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. 視需要對任意數量的屬性重複上述步驟，然後選擇下一步。
   g. 您現在不會新增任何管道活動。在富集、轉換和篩選訊息頁面上，選擇下一步。
4. 檢閱您的選擇，然後選擇建立管道。
5. 確認您的新管道顯示在管道頁面上。

> **Note**
>
> 您已建立 AWS IoT Analytics 資源，以便他們可以執行下列動作：
>
> - 使用頻道收集原始、未處理的 IoT 裝置訊息資料。
> - 將 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。
>
> 接著，您將建立 AWS IoT Analytics SQL 資料集，以探索有關 IoT 裝置的實用洞見。

建立資料集

> **Note**
>
> 資料集通常是資料集合，不一定會以表格形式組織。相反地， 會透過將 SQL 查詢套用至資料存放區中的資料來 AWS IoT Analytics 建立資料集。

您現在有一個頻道，將原始訊息資料路由到管道，將資料存放在可以查詢的資料存放區中。若要查詢資料，您可以建立資料集。資料集包含您用來查詢資料存放區的 SQL 陳述式和表達式，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 Amazon CloudWatch 排程表達式的表達式來建立選用的排程。

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訊頁面上，指定資料集的詳細資訊。
   a. 輸入資料集的名稱。
   b. 針對資料存放區來源，選擇可識別您先前建立之資料存放區的唯一 ID。
   c. （選用）對於標籤，將一或多個自訂標籤（索引鍵/值對）新增至資料集。
4. 使用 SQL 表達式查詢您的資料並回答分析問題。查詢的結果會存放在此資料集。
   a. 在作者查詢欄位中，輸入使用萬用字元來顯示最多五列資料的 SQL 查詢。

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

      如需 中支援之 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，並在查詢之後的表格中顯示結果。

      > **Note**
      >
      > - 目前在教學課程中，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 查詢限制在合理的大小，使其不會長時間執行，因為 Athena 會限制執行中查詢的數量上限。因此，您必須小心將 SQL 查詢限制在合理的大小。

  我們建議在測試期間在查詢中使用 `LIMIT`子句。測試成功後，您可以移除此子句。

5. （選用）當您使用指定時間範圍的資料建立資料集內容時，某些資料可能無法及時送達進行處理。若要允許延遲，您可以指定位移或差異。如需詳細資訊，請參閱透過 Amazon CloudWatch Events 取得延遲資料通知。

   此時，您不會設定資料選擇篩選條件。在設定資料選擇篩選條件頁面上，選擇下一步。

6. （選用）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，您不會排程重複執行查詢，因此在設定查詢排程頁面上選擇下一步。

7. AWS IoT Analytics 會建立此資料集內容的版本，並在指定的期間內存放您的分析結果。我們建議 90 天，但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。

   您可以使用預設資料集保留期為無限期，並保持停用版本控制。在設定分析結果頁面上，選擇下一步。

8. （選用）您可以設定資料集結果的交付規則到特定目的地，例如 AWS IoT Events。

   您不會在本教學課程中的其他位置交付結果，因此在設定資料集內容交付規則頁面上，選擇下一步。

9. 檢閱您的選擇，然後選擇建立資料集。

10. 確認您的新資料集顯示在資料集頁面上。

使用 傳送訊息資料 AWS IoT

如果您有將資料路由到管道的頻道，該管道會將資料存放在可查詢的資料存放區中，您就可以將 IoT 裝置資料傳送到其中 AWS IoT Analytics。您可以使用下列選項 AWS IoT Analytics 將資料傳送至 ：

- 使用 AWS IoT 訊息中介裝置。
- 使用 AWS IoT Analytics BatchPutMessage API 操作。

在下列步驟中 AWS IoT Core ，您會從 主控台中的 AWS IoT 訊息中介裝置傳送訊息資料，讓 AWS IoT Analytics 可以擷取此資料。

> **Note**
>
> 當您為訊息建立主題名稱時，請注意下列事項：
>
> - 主題名稱不區分大小寫。相同承載EXAMPLE中名為 example和 的欄位視為重複。
> - 主題名稱不能以 $ 字元開頭。開頭為 的主題$是預留主題，只能由 使用 AWS IoT。
> - 請勿在主題名稱中包含個人身分識別資訊，因為此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。

使用 傳送訊息資料 AWS IoT

1. 登入 AWS IoT 主控台。
2. 在導覽窗格中，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. 針對主題名稱，輸入與您在建立頻道時輸入的主題篩選條件相符的名稱。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. （選用）選擇新增組態以取得其他訊息通訊協定選項。
7. 選擇 Publish (發布)。

   這會發佈頻道擷取的訊息。然後，您的管道會將訊息路由到您的資料存放區。

檢查 AWS IoT 訊息的進度

您可以依照下列步驟檢查訊息是否正在擷取到您的頻道。

檢查 AWS IoT 訊息的進度

1. 登入 https：//https://console.aws.amazon.com/iotanalytics/。
2. 在導覽窗格中，選擇頻道，然後選擇您先前建立的頻道名稱。
3. 在頻道的詳細資訊頁面上，向下捲動至監控區段，然後調整顯示的時間範圍 (1h 3h 12h 1d 3d 1w)。選擇 1w 之類的值，以檢視上週的資料。

您可以使用類似的功能來監控管道活動執行時間和管道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道的一部分，因此不應看到任何執行時間錯誤。

監控管道活動

1. 在導覽窗格中，選擇管道，然後選擇您先前建立的管道名稱。
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標 (1h 3h 12h 1d 3d 1w) 來調整顯示的時間範圍。

存取查詢結果

資料集內容是以 CSV 格式包含查詢結果的檔案。

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就緒，請查看資料集下是否有與您已成功啟動資料集查詢類似的訊息。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上嵌入 Jupyter 筆記本的 HTML 部分。如需詳細資訊，請參閱使用主控台視覺化 AWS IoT Analytics 資料。

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至 Amazon S3 儲存貯體，以便與現有資料湖整合，或從內部應用程式和視覺化工具存取。請參閱 CreateDataset 操作`contentDeliveryRules::destination::s3DestinationConfiguration`中的 欄位。

AWS IoT Events

您可以傳送資料集內容做為輸入 AWS IoT Events，此服務可讓您監控裝置或程序操作失敗或變更，並在發生此類事件時啟動其他動作。

若要這樣做，請使用 CreateDataset 操作建立資料集，並在 欄位 中指定 AWS IoT Events 輸入`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`。您也必須指定角色`roleArn`的 ，以授予執行 的 AWS IoT Analytics 許可`iotevents:BatchPutMessage`。每當資料集內容建立時， AWS IoT Analytics 都會將每個資料集內容項目做為訊息傳送至指定的 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 傳送包含如下欄位的訊息。

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

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

您會想要建立可辨識您感興趣的欄位的 AWS IoT Events 輸入 (`what`、、`who`、 的一或多個`dt`)，並建立偵測器 AWS IoT Events 模型，在事件中使用這些輸入欄位來觸發動作或設定內部變數。

Jupyter 筆記本

Jupyter Notebook 是一種開放原始碼解決方案，可使用指令碼語言來執行臨機操作資料探索和進階分析。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，例如用於預測的 k 平均值叢集和迴歸模型。

AWS IoT Analytics 使用 Amazon SageMaker AI 筆記本執行個體託管其 Jupyter 筆記本。建立筆記本執行個體之前，您必須在 AWS IoT Analytics 和 Amazon SageMaker AI 之間建立關係：

1. 導覽至 SageMaker AI 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 AI 角色：
   a. 開啟角色。它應該有一個受管政策。
   b. 選擇新增內嵌政策，然後在服務中選擇 iotAnalytics。選擇選取動作，然後在**GetDatasetContent**搜尋方塊中輸入 ，然後選擇它。選擇檢閱政策。
   c. 檢閱政策的準確性，輸入名稱，然後選擇建立政策。

這提供新建立的角色從中讀取資料集的許可 AWS IoT Analytics。

1. 返回 https://console.aws.amazon.com/iotanalytics/，然後在左側導覽窗格中，選擇筆記本。在筆記本頁面上，選擇建立筆記本。
2. 在選取範本頁面上，選擇 IoTA 空白範本。
3. 在設定筆記本頁面上，輸入筆記本的名稱。在選取資料集來源中，選擇 ，然後選擇您先前建立的資料集。在選取筆記本執行個體中，選擇您在 SageMaker AI 中建立的筆記本執行個體。
4. 檢閱您的選擇後，選擇建立筆記本。
5. 在筆記本頁面上，您的筆記本執行個體將在 Amazon SageMaker AI 主控台中開啟。

筆記本範本

AWS IoT Analytics 筆記本範本包含 AWS 撰寫的機器學習模型和視覺化，可協助您開始使用 AWS IoT Analytics 使用案例。您可以使用這些筆記本範本來進一步了解或重複使用這些範本，以符合您的 IoT 裝置資料並提供立即價值。

您可以在 AWS IoT Analytics 主控台中找到下列筆記本範本：

- 偵測情境異常 – 使用 Poisson 指數加權移動平均 (PEWMA) 模型，以測量風速套用情境異常偵測。
- 太陽面板輸出預測 – 套用分段、季節性和線性時間序列模型來預測太陽面板的輸出。
- 噴射引擎的預測性維護 – 套用多變量長短期記憶體 (LSTM) 神經網路和邏輯回歸來預測噴射引擎故障。
- 智慧家庭客戶區隔 – 套用 k 平均值和主體元件分析 (PCA) 分析，以偵測智慧家庭使用量資料中的不同客戶客群。
- 智慧城市擁塞預測 – 使用 LSTM 預測城市高速公路的使用率。
- 智慧城市空氣品質預測 – 使用 LSTM 預測城市中心的粒子污染。

入門 AWS IoT Analytics

本節討論您用來收集、存放、處理和查詢裝置資料的基本命令 AWS IoT Analytics。此處顯示的範例使用 AWS Command Line Interface (AWS CLI)。如需 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface 使用者指南。如需 CLI 命令的詳細資訊 AWS IoT，請參閱 AWS Command Line Interface 參考中的 iot。

> **Important**
>
> 使用 `aws iotanalytics`命令來 AWS IoT Analytics 使用 與 互動 AWS CLI。使用 `aws iot`命令來使用 與 IoT 系統的其他部分互動 AWS CLI。

> **Note**
>
> 請注意，當您在下列範例中輸入 AWS IoT Analytics 實體的名稱（頻道、資料集、資料存放區和管道）時，系統會自動將您使用的任何大寫字母變更為小寫。實體的名稱必須以小寫字母開頭，且僅包含小寫字母、底線和數字。

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。傳入的訊息會傳送到頻道，因此第一個步驟是為資料建立頻道。

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

如果您想要擷取 AWS IoT 訊息 AWS IoT Analytics，您可以建立 AWS IoT 規則引擎規則，將訊息傳送到此頻道。這稍後會顯示在 中將資料擷取至 AWS IoT Analytics。將資料傳入頻道的另一種方法是使用 AWS IoT Analytics 命令 `BatchPutMessage`。

若要列出您已建立的頻道：

```
aws iotanalytics list-channels
```

取得頻道的詳細資訊。

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

未處理的頻道訊息會存放在由 管理的 Amazon S3 儲存貯體中 AWS IoT Analytics，或存放在由您管理的儲存貯體中。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。如果您選擇將頻道訊息存放在您管理的 Amazon S3 儲存貯體中，則必須授予代表您在 Amazon S3 儲存貯體上執行這些動作的 AWS IoT Analytics 許可：s3:GetBucketLocation（驗證儲存貯體位置） s3:PutObject（存放區）、 s3:GetObject（讀取）、 s3:ListBucket（重新處理）。

Example

JSON

```
{
    "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:*:*:*:your-iot-analytics-bucket",
                "arn:aws:s3:*:*:your-iot-analytics-bucket/*"
            ]
        }
    ]
}
```

如果您變更客戶受管頻道儲存的選項或許可，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。請參閱重新處理頻道資料。

建立資料存放區

資料存放區會接收和存放您的訊息。它不是資料庫，而是訊息的可擴展且可查詢儲存庫。您可以建立多個資料存放區來存放來自不同裝置或位置的訊息，或者您的 可以使用單一資料存放區來接收您的所有 AWS IoT 訊息。

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

列出您已建立的資料存放區。

```
aws iotanalytics list-datastores
```

取得有關資料存放區的詳細資訊。

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

AWS IoT Analytics 資源的 Amazon S3 政策

您可以將處理的資料存放區訊息存放在由 管理的 Amazon S3 儲存貯體中， AWS IoT Analytics 或存放在您管理的儲存貯體中。當您建立資料存放區時，請使用 `datastoreStorage` API 參數選取您想要的 Amazon S3 儲存貯體。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儲存在您管理的 Amazon S3 儲存貯體中，您必須授予 AWS IoT Analytics 許可，才能在 Amazon S3 儲存貯體上執行這些動作：

- `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 Amazon S3 儲存貯體政策，以 AWS IoT Analytics 授予許可來叫用儲存貯體內容的 Amazon Athena 查詢。

> **Note**
>
> 建議您在儲存貯體政策`aws:SourceArn`中指定 ，以協助防止混淆代理人安全問題。這透過僅允許來自指定帳戶的請求來限制存取。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”。

以下是授予這些必要許可的儲存貯體政策範例。

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastore"
                    ]
                }
            }
        }
    ]
}
```

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

> **Note**
>
> 如果您更新客戶受管資料存放區的選項或許可，您可能需要重新處理頻道資料，以確保任何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 資料存放區目前支援 JSON 和 Parquet 檔案格式。預設檔案格式是 JSON。

- JSON (JavaScript 物件標記法) - 支援名稱值對和值排序清單的文字格式。
- Apache Parquet - 用於有效存放和查詢大量資料的單欄式儲存格式。

若要設定 AWS IoT Analytics 資料存放區的檔案格式，您可以在建立資料存放區時使用 `FileFormatConfiguration` 物件。

`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 個字元。

`type`

資料的類型。如需支援的資料類型的詳細資訊，請參閱《 AWS Glue 開發人員指南》中的常見資料類型。

長度限制：1-131072 個字元。

AWS IoT Analytics 支援 Amazon Athena 中資料類型頁面上列出的所有資料類型，`DECIMAL(precision, scale)`- 除外*precision*。

建立資料存放區 （主控台）

下列程序說明如何建立以 Parquet 格式儲存資料的資料存放區。

建立資料存放區

1. 登入 https：//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細資訊頁面上，輸入資料存放區的基本資訊。
   a. 針對資料存放區 ID，輸入唯一的資料存放區 ID。您無法在建立後變更此 ID。
   b. （選用） 針對標籤，選擇新增標籤，將一或多個自訂標籤 （索引鍵/值對） 新增至您的資料存放區。標籤可協助您識別為其建立的資源 AWS IoT Analytics。
   c. 選擇下一步。
5. 在設定儲存類型頁面上，指定如何儲存您的資料。
   a. 針對儲存類型，選擇服務受管儲存。
   b. 針對設定您想要保留已處理資料的時間長度，選擇無限期。
   c. 選擇下一步。
6. 在設定資料格式頁面上，定義資料記錄的結構和格式。
   a. 針對分類，選擇 Parquet。您無法在建立資料存放區之後變更此格式。
   b. 針對推論來源，選擇資料存放區的 JSON 字串。
   c. 針對字串，以 JSON 格式輸入您的結構描述，例如下列範例。

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

d. 選擇推論結構描述。

e. 在設定 Parquet 結構描述下，確認格式符合您的 JSON 範例。如果格式不相符，請手動更新 Parquet 結構描述。

- 如果您希望結構描述顯示更多資料欄，請選擇新增資料欄，輸入資料欄名稱，然後選擇資料類型。

> **Note**
>
> 根據預設，您的結構描述可以有 100 個資料欄。如需詳細資訊，請參閱 AWS IoT Analytics 配額。

- 您可以變更現有資料欄的資料類型。如需支援資料類型的詳細資訊，請參閱《 AWS Glue 開發人員指南》中的常見資料類型。

> **Note**
>
> 建立資料存放區之後，您無法變更現有資料欄的資料類型。

- 若要移除現有資料欄，請選擇移除資料欄。

f. 選擇下一步。

7. （選用）AWS IoT Analytics 支援資料存放區中的自訂分割區，因此您可以查詢已刪除的資料以改善延遲。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。

8. 在檢閱和建立頁面上，檢閱您的選擇，然後選擇建立資料存放區。

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

9. 確認您的新資料存放區顯示在資料存放區頁面上。

自訂分割區

AWS IoT Analytics 支援資料分割，因此您可以在資料存放區中組織資料。當您使用資料分割來組織資料時，您可以查詢已刪除的資料。這可減少每個查詢掃描的資料量，並改善延遲。

您可以根據透過管道活動新增的訊息資料屬性或屬性來分割資料。

若要開始使用，請在資料存放區中啟用資料分割。指定一或多個資料分割區維度，並將分割的資料存放區連接到 AWS IoT Analytics 管道。然後，撰寫利用 `WHERE`子句來最佳化效能的查詢。

建立資料存放區（主控台）

下列程序說明如何使用自訂分割區建立資料存放區。

建立資料存放區

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. 在指定資料存放區詳細資訊頁面上，輸入資料存放區的基本資訊。
   a. 針對資料存放區 ID，輸入唯一的資料存放區 ID。您無法在建立後變更此 ID。
   b. （選用）針對標籤，選擇新增標籤，將一或多個自訂標籤（鍵/值對）新增至資料存放區。標籤可協助您識別為其建立的資源 AWS IoT Analytics。
   c. 選擇下一步。
5. 在設定儲存類型頁面上，指定如何儲存您的資料。
   a. 針對儲存類型，選擇服務受管儲存。
   b. 針對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。
6. 在設定資料格式頁面上，定義資料記錄的結構和格式。
   a. 針對資料存放區資料格式分類，選擇 JSON 或 Parquet。如需 AWS IoT Analytics 支援檔案類型的詳細資訊，請參閱 檔案格式。

   > **Note**
   > 您無法在建立資料存放區之後變更此格式。

b. 選擇下一步。

7. 為此資料存放區建立自訂分割區。

a. 對於新增資料分割區，選取啟用。

b. 針對資料分割區來源，指定分割區來源的基本資訊。

選擇範例來源，然後選取收集此資料存放區訊息的 AWS IoT Analytics 頻道。

c. 針對訊息範例屬性，選取您要用來分割資料存放區的訊息屬性。然後，在動作下將選取項目新增為屬性分割區維度或時間戳記分割區維度。

> **Note**
>
> 您只能將一個時間戳記分割區新增至資料存放區。

d. 針對自訂資料存放區分割區維度，定義分割區維度的基本資訊。您在上一個步驟中選取的每個訊息範例屬性都會成為分割區的維度。使用以下選項自訂每個維度：

- 分割區類型 - 指定此分割區維度是屬性還是時間戳記分割區類型。
- 屬性名稱和維度名稱 - 預設情況下， AWS IoT Analytics 將使用您選取做為屬性分割區維度識別符的訊息範例屬性名稱。編輯屬性名稱以自訂分割區維度的名稱。您可以使用 `WHERE`子句中的維度名稱來最佳化查詢效能。
  - 任何分割區屬性維度的名稱字首為 `__partition_`。
  - 對於時間戳記分割區類型， 會使用名稱 `__year`、`__month`、`__day`、 AWS IoT Analytics 建立下列四個維度`__hour`。
- 排序 - 重新排列分割區維度，以改善查詢的延遲。

對於時間戳記格式，請比對從訊息資料擷取的時間戳記，指定時間戳記分割區的格式。您可以選擇其中一個 AWS IoT Analytics 列出的格式選項，或指定符合資料格式的選項。進一步了解指定日期時間格式工具。

若要新增非訊息屬性的新維度，請選擇新增分割區。

e. 選擇下一步。

8. 在檢閱和建立頁面上，檢閱您的選擇，然後選擇建立資料存放區。

> **Important**
>
> - 您無法在建立資料存放區之後變更資料存放區 ID。
> - 若要編輯現有的分割區，您必須建立另一個資料存放區，並透過管道重新處理資料。

9. 確認您的新資料存放區顯示在資料存放區頁面上。

建立管道

管道會使用來自頻道的訊息，並可讓您在將訊息存放在資料存放區之前處理和篩選訊息。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。如需更複雜管道的資訊，請參閱管道活動。

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
```

mypipeline.json 檔案包含下列內容。

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

```
}
```

執行下列命令來列出現有的管道。

```
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
```

將資料擷取至 AWS IoT Analytics

如果您有將資料路由到管道的管道，將資料存放在可查詢的資料存放區中，您就可以將訊息資料傳送到其中 AWS IoT Analytics。在這裡，我們顯示將資料傳入的兩種方法 AWS IoT Analytics。您可以使用訊息中介裝置或使用 AWS IoT Analytics `BatchPutMessage` API AWS IoT 傳送訊息。

主題

- 使用 AWS IoT 訊息中介裝置
- 使用 BatchPutMessage API

使用 AWS IoT 訊息中介裝置

若要使用 AWS IoT 訊息中介裝置，您可以使用規則引擎建立 AWS IoT 規則。規則會將具有特定主題的訊息路由到其中 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 訊息路由到 AWS IoT Analytics 頻道，您可以設定規則。但首先，您必須建立 IAM 角色，授予該規則將訊息資料傳送至 AWS IoT Analytics 頻道的許可。

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` 檔案的內容看起來應該如下所示。

JSON

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

然後，將政策文件連接到角色。

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

pd.json 檔案的內容看起來應該如下所示。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
                "arn:aws:iotanalytics:us-east-1:111122223333:channel/your-
channel"
            ]
        }
    ]
}
```

建立 AWS IoT 規則

建立 AWS IoT 規則，將訊息傳送到您的頻道。

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

`rule.json` 檔案的內容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

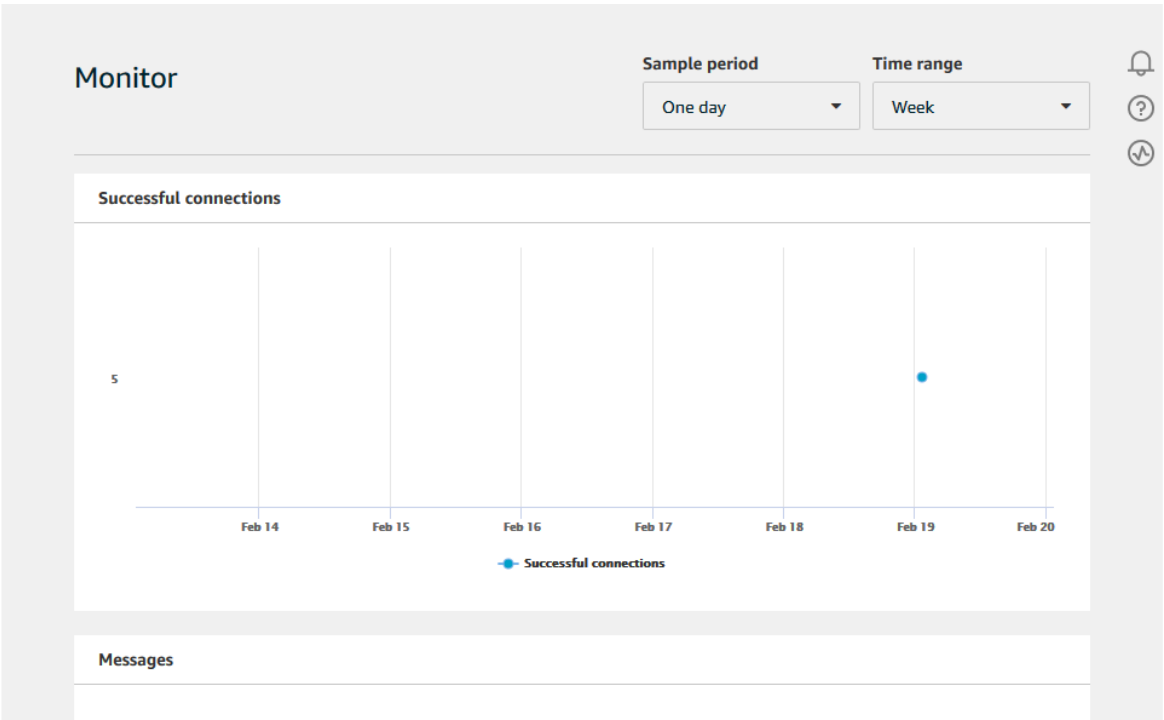
將規則加入頻道、管道管道和資料存放區的管道之後，符合規則的任何資料現在都會流 AWS IoT Analytics 經資料存放區，準備進行查詢。若要測試此項目，您可以使用 AWS IoT 主控台來傳送訊息。

> **Note**
>
> 您傳送的訊息承載 （ 資料 ） 欄位名稱 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 (-)。
> - 在規則表達式術語中："^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$"。
> - 不能大於 255 個字元

- 不區分大小寫。相同承載FOO中名為 foo和 的欄位視為重複。

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。

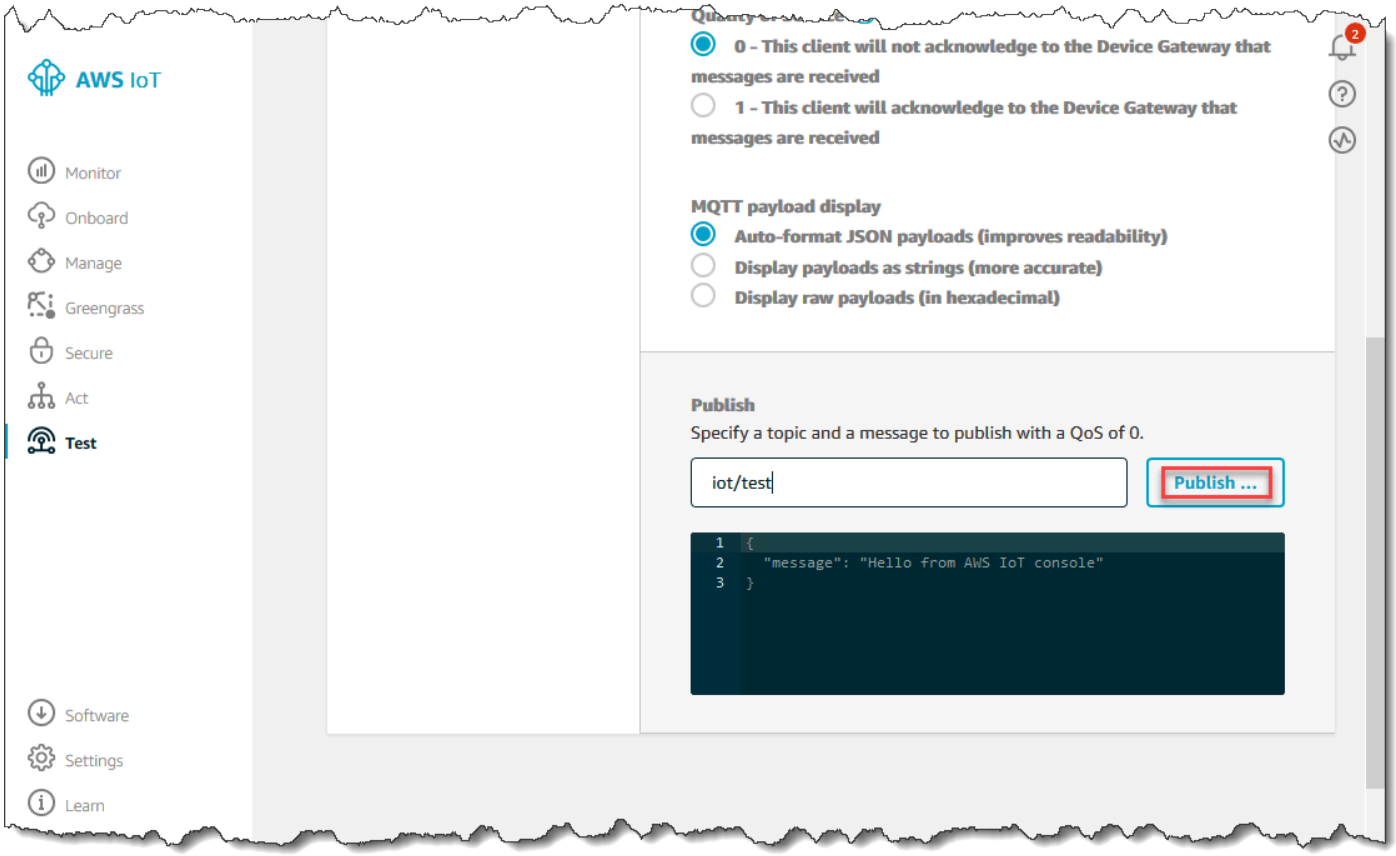
1. 請在 AWS IoT 主控台 左側的導覽窗格中，選擇 Test (測試)。



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 內容存在，否則請輸入內容。

```
{
    "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。

使用 BatchPutMessage API

取得訊息資料的另一個方法是 AWS IoT Analytics 使用 `BatchPutMessage` API 命令。此方法不需要您設定 AWS IoT 規則，將具有特定主題的訊息路由到您的頻道。但是，它確實需要將資料/訊息傳送到頻道的裝置能夠執行使用 AWS SDK 建立的軟體，或能夠使用 AWS CLI 呼叫 `BatchPutMessage`。

1. 建立檔案`messages.json`，其中包含要傳送的訊息（在此範例中，只會傳送一則訊息）。

```
[
    { "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
```

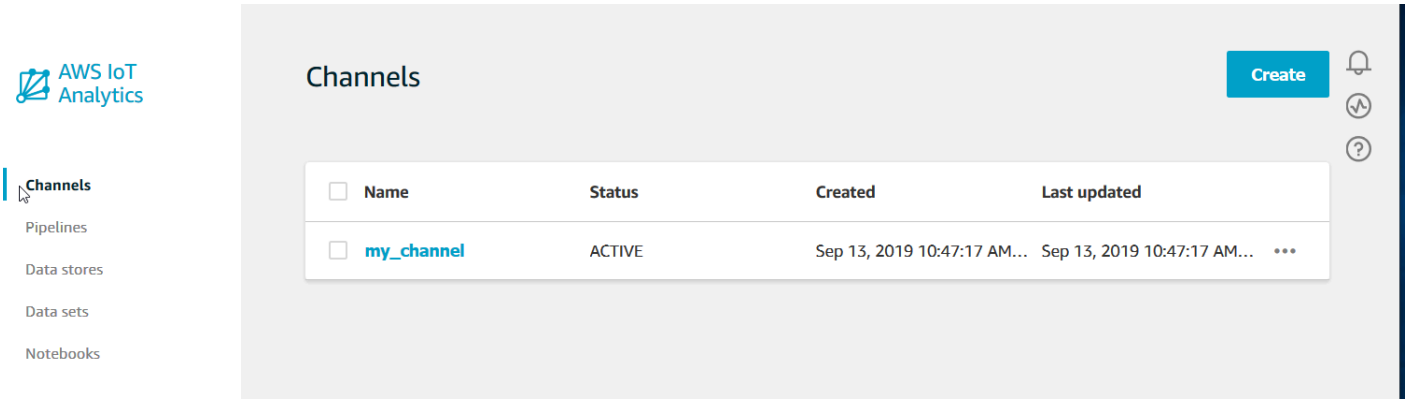
如果沒有錯誤，您會看到以下輸出。

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

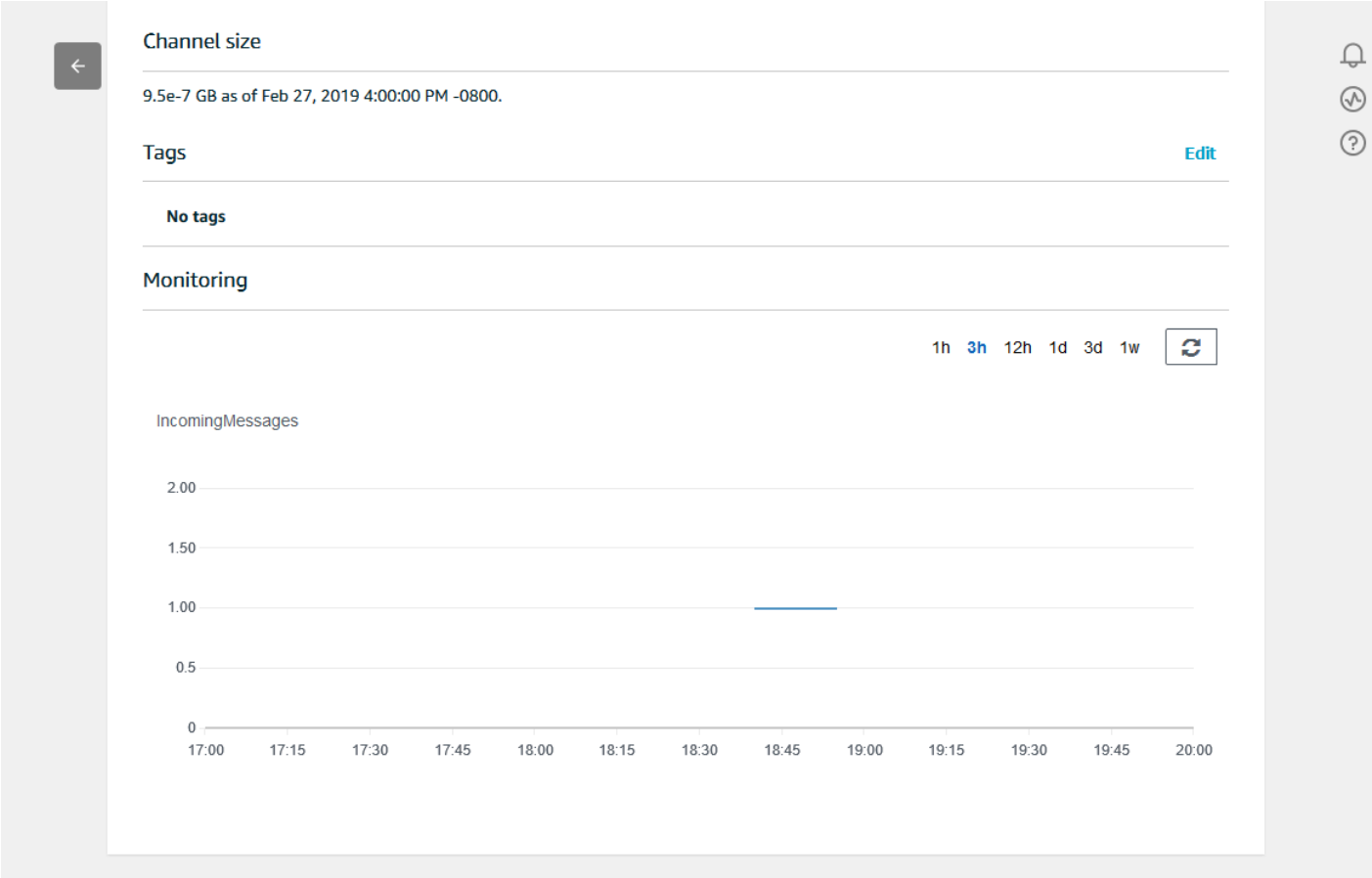
監控擷取的資料

您可以使用 AWS IoT Analytics 主控台，檢查您傳送的訊息是否正在擷取到您的頻道。

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 （ 如有必要），然後選擇您先前建立的頻道名稱。

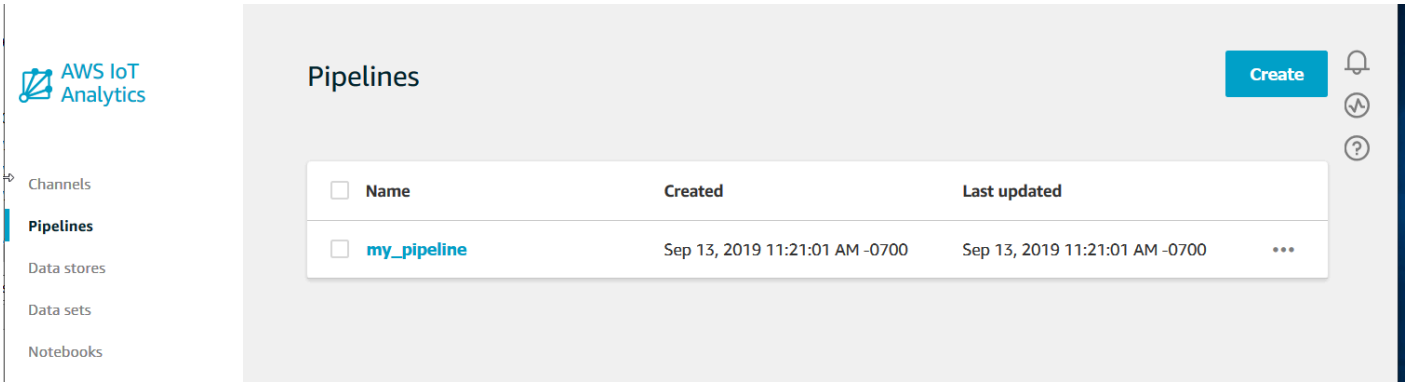


2. 在頻道詳細資訊頁面上，向下捲動到 Monitoring (監控) 區段。視需要調整顯示時間範圍，方法為選擇其中一個時間範圍指標 (1h 3h 12h 1d 3d 1w)。您應該會看到圖形線，指出在指定時間範圍內擷取到此頻道的訊息數量。

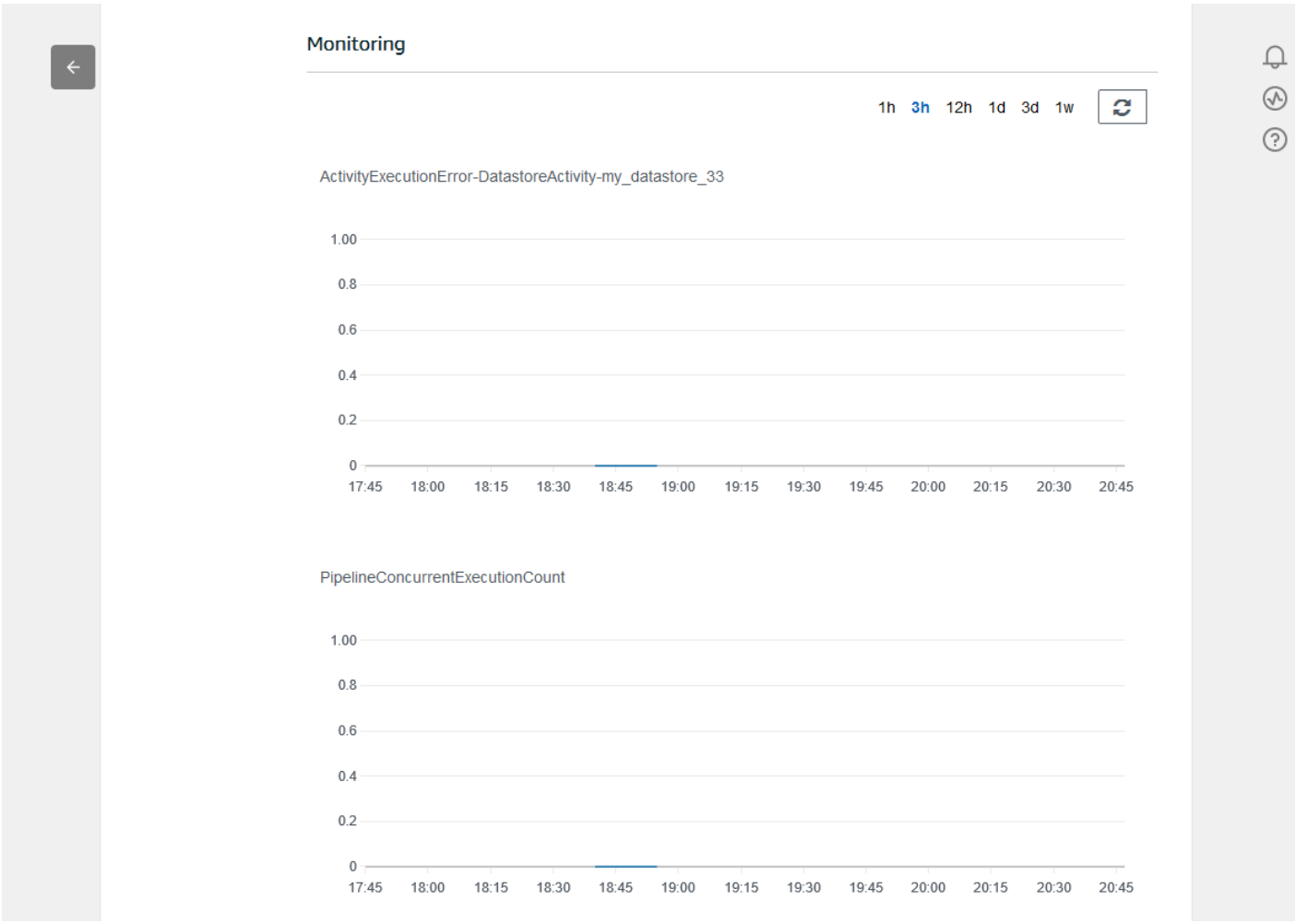


存在類似的監控功能，可用於檢查管道活動執行。您可以在管道詳細資訊頁面上監控活動執行錯誤。如果您尚未在管道中指定活動，則應該會顯示 0 個執行錯誤。

1. 在AWS IoT Analytics 主控台的左側導覽窗格中，選擇準備，然後選擇管道，然後選擇您先前建立的管道名稱。



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)。您應該會看到圖形列，指出指定時間範圍內管道活動執行錯誤的數量。



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 可以查詢資料以回答分析問題。雖然資料存放區不是資料庫，但您可以使用 SQL 表達式來查詢資料，並產生存放在資料集中的結果。

主題

- 查詢資料
- 存取查詢的資料

查詢資料

若要查詢資料，您可以建立資料集。資料集包含您用來查詢資料存放區的 SQL，以及在您所選的日期和時間重複查詢的選用排程。您可以使用類似 Amazon CloudWatch 排程表達式的表達式來建立選用排程。

執行下列命令來建立資料集。

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

其中mydataset.json檔案包含下列內容。

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

執行下列命令，透過執行查詢來建立資料集內容。

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

請等待幾分鐘再繼續建立資料集內容。

存取查詢的資料

查詢的結果是您的資料集內容，以 CSV 格式儲存為 檔案。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
```

如果您的資料集包含任何資料，則來自 的輸出get-dataset-content在 status "state": "SUCCEEDED"欄位中具有 ，如以下範例所示。

```
{
    "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。

探索 AWS IoT Analytics 資料

您有多種儲存、分析和視覺化 AWS IoT Analytics 資料的選項。

本頁主題：

- Simple Storage Service (Amazon Simple Storage Service (Amazon S3))
- AWS IoT Events
- 快速套件
- Jupyter 筆記本

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

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

AWS IoT Events

您可以傳送資料集內容做為輸入 AWS IoT Events，此服務可讓您監控裝置或程序操作失敗或變更，並在發生此類事件時觸發其他動作。

若要這樣做，請使用 CreateDataset 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 ，該角色授予執行 "iotevents：BatchPutMessage" 的 AWS IoT Analytics 許可。每當資料集的內容建立時， AWS IoT Analytics 都會將每個資料集內容項目做為訊息傳送至指定的 AWS IoT Events 輸入。例如，如果您的資料集包含：

```
"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 會傳送訊息，其中包含如下的欄位：

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

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

而且您會想要建立 AWS IoT Events 輸入，以辨識您感興趣的欄位 (`what`、、 `who`的一或多個`dt`)，並建立 AWS IoT Events 偵測器模型，在事件中使用這些輸入欄位來觸發動作或設定內部變數。

快速套件

AWS IoT Analytics 提供與 Quick Suite 的直接整合。Quick Suite 是一種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，以及快速從資料中取得商業洞見。Quick Suite 可讓組織擴展到數十萬使用者，並使用強大的記憶體內引擎 (SPICE) 提供回應效能。這些區域提供 Quick Suite。

Jupyter 筆記本

AWS IoT Analytics Jupyter Notebook 也可以直接使用資料集，以執行進階分析和資料探勘。Jupyter 筆記本是一種開放原始碼解決方案。您可以前往 http://jupyter.org/install.html，安裝並下載該筆記本。也提供與 Amazon 託管筆記本解決方案 SageMaker AI 的額外整合。

保留多個版本的資料集

您可以在叫用 CreateDataset 和 UpdateDataset APIs 時指定資料集`retentionPeriod and versioningConfiguration`欄位的值，以選擇要保留多少版本的資料集內容，以及保留多久：

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

這兩個參數的設定會一起運作，以下列方式判斷保留了多少版本的資料集內容，以及保留多久。

	retentionPeriod [未指定]	retentionPeriod： unlimited = TRUE，numberOfDays = 未設定	retentionPeriod： unlimited = FALSE，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

您傳送至的訊息承載 （ 資料 ） 欄位名稱 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 (-)。
- 在規則表達式術語中："^[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。相同承載中名為「foo」和「FOO」的欄位視為重複。

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，可用來大規模收集、建模、分析和視覺化工業設備的資料。此服務提供資產建模架構，用於建置工業裝置、程序和設施的表示法。

透過 AWS IoT SiteWise 資產模型，您可以定義要使用的工業設備資料，以及如何將資料處理成複雜的指標。您可以設定資產模型來收集和處理 AWS 雲端中的資料。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 ，因此您可以對 AWS IoT SiteWise 資料執行和排程 SQL 查詢。若要開始查詢 AWS IoT SiteWise 資料，請依照AWS IoT SiteWise 《 使用者指南》中的設定儲存設定中的程序來建立資料存放區。然後，遵循 使用 AWS IoT SiteWise 資料建立資料集 （ 主控台 ） 或 中的步驟使用 AWS IoT SiteWise 資料建立資料集 (AWS CLI)來建立 AWS IoT Analytics 資料集，並在工業資料上執行 SQL 查詢。

主題

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

使用 AWS IoT SiteWise 資料建立 AWS IoT Analytics 資料集

AWS IoT Analytics 資料集包含用於查詢資料存放區中資料的 SQL 陳述式和表達式，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 Amazon CloudWatch 排程表達式的表達式來建立選用的排程。

> **Note**
>
> 資料集通常是資料集合，不一定會以表格形式組織。相反地， 會透過將 SQL 查詢套用至資料存放區中的資料來 AWS IoT Analytics 建立資料集。

請依照下列步驟，開始為您的 AWS IoT SiteWise 資料建立資料集。

主題

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

使用 AWS IoT SiteWise 資料建立資料集 （ 主控台 ）

使用這些步驟在 AWS IoT Analytics 主控台中為您的 AWS IoT SiteWise 資料建立資料集。

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訊頁面上，指定資料集的詳細資訊。
   a. 輸入資料集的名稱。
   b. 針對資料存放區來源，選擇可識別 AWS IoT SiteWise 資料存放區的唯一 ID。
   c. （選用） 對於標籤，將一或多個自訂標籤 （ 索引鍵/值對 ） 新增至資料集。
4. 使用 SQL 表達式查詢您的資料並回答分析問題。
   a. 在作者查詢欄位中，輸入使用萬用字元來顯示最多五列資料的 SQL 查詢。

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

如需 中支援之 SQL 功能的詳細資訊 AWS IoT Analytics，請參閱 中的 SQL 表達式 AWS IoT Analytics。或者，教學課程：查詢 中的 AWS IoT SiteWise 資料 AWS IoT Analytics如需可以提供資料洞見的統計查詢範例，請參閱 。

b. 您可以選擇測試查詢來驗證您的輸入是否正確，並在查詢之後的表格中顯示結果。

> **Note**
>
> 由於 Amazon Athena 限制執行中查詢的數量上限，因此您應該將 SQL 查詢限制為合理的大小，使其不會長時間執行。

5. （選用）當您使用指定時間範圍的資料建立資料集內容時，某些資料可能無法及時送達進行處理。若要允許延遲，您可以指定位移或差異。如需詳細資訊，請參閱透過 Amazon CloudWatch Events 取得延遲資料通知。

   在設定資料選擇篩選條件頁面上設定資料選擇篩選條件後，選擇下一步。

6. （選用）在設定查詢排程頁面上，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。

> **Note**
>
> 資料從 AWS IoT SiteWise 每六小時擷取到 AWS IoT Analytics 。我們建議選取六個小時或更長時間的頻率。

   選擇頻率的 和 選項，然後選擇下一步。

7. AWS IoT Analytics 會建立此資料集內容的版本，並在指定的期間內存放您的分析結果。我們建議 90 天，但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。

   在設定資料集的結果頁面上選取您的選項後，選擇下一步。

8. （選用）您可以設定資料集結果的交付規則到特定目的地，例如 AWS IoT Events。

   在設定資料集內容交付規則頁面上選取您的選項後，選擇下一步。

9. 檢閱您的選擇，然後選擇建立資料集。

10. 確認您的新資料集顯示在資料集頁面上。

使用 AWS IoT SiteWise 資料建立資料集 (AWS CLI)

執行下列 AWS CLI 命令以開始查詢 AWS IoT SiteWise 您的資料。

此處顯示的範例使用 AWS Command Line Interface (AWS CLI)。如需 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface 使用者指南。如需可用 CLI 命令的詳細資訊 AWS IoT Analytics，請參閱《 AWS Command Line Interface 參考》中的 iotanalytics。

建立資料集

1. 執行下列`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`檔案包含下列內容。

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

如需 中支援之 SQL 功能的詳細資訊 AWS IoT Analytics，請參閱 中的 SQL 表達式 AWS IoT Analytics。或者，教學課程：查詢 中的 AWS IoT SiteWise 資料 AWS IoT Analytics如需可以提供資料洞見的統計查詢範例，請參閱 。

2. 執行下列`create-dataset-content`命令，透過執行查詢來建立資料集內容。

```
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，以 CSV 格式儲存為檔案。系統會透過 Amazon S3 提供該檔案。下列步驟說明如何檢查結果是否就緒並下載 檔案。

主題

在 AWS IoT Analytics （ 主控台 ） 中存取資料集內容

如果您的資料集包含任何資料，您可以在 AWS IoT Analytics 主控台中預覽和下載 SQL 查詢結果。

存取 AWS IoT Analytics 資料集結果

1. 在 主控台的資料集頁面上，選擇您要存取的資料集名稱。
2. 在資料集摘要頁面上，選擇內容索引標籤。
3. 在資料集內容表格中，選擇您要預覽結果的查詢名稱，或下載結果的 csv 檔案。

在 AWS IoT Analytics (AWS CLI) 中存取資料集內容

如果您的資料集包含任何資料，您可以預覽和下載 SQL 查詢結果。

此處顯示的範例使用 AWS Command Line Interface (AWS CLI)。如需 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface 使用者指南。如需 CLI 命令的詳細資訊 AWS IoT Analytics，請參閱《AWS Command Line Interface 參考》中的 iotanalytics。

存取 AWS IoT Analytics 資料集結果 (AWS CLI)

1. 執行下列`get-dataset-content`命令以檢視查詢的結果。

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

2. 如果您的資料集包含任何資料，則來自 的輸出`get-dataset-content`在 `status` `"state": "SUCCEEDED"`欄位中具有 ，例如在下列範例中。

```
{
    "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. 的輸出`get-dataset-content`包含 `dataURI`，這是輸出結果的簽章 URL。它的有效期間很短 (幾個小時)。請造訪 `dataURI` URL 以存取您的 SQL 查詢結果。

> **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 中的 AWS IoT SiteWise 資料 AWS IoT Analytics

本教學課程示範如何在 中查詢 AWS IoT SiteWise 資料 AWS IoT Analytics。本教學課程使用 中示範的資料 AWS IoT SiteWise ，提供風力發電廠的範例資料集。

> **Important**
>
> 您將為此示範建立和使用的資源付費。

主題

- 先決條件
- 載入並驗證資料
- 資料探勘
- 執行統計查詢
- 清除您的教學課程資源

先決條件

在此教學課程中，您需執行下列資源：

- 您必須擁有 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 中的程序建立 AWS 帳戶。
- 執行 Windows、macOS、Linux 或 Unix 的開發電腦，用來存取 AWS 管理主控台。如需詳細資訊，請參閱 AWS 管理主控台入門。
- AWS IoT SiteWise 資料，定義 AWS IoT SiteWise 模型和資產，並串流代表風力發電廠設備資料的資料。若要建立您的資料，請遵循AWS IoT SiteWise 《使用者指南》中的建立 AWS IoT SiteWise 示範中的步驟。
- 您管理的現有資料存放區中的 AWS IoT SiteWise 示範風力發電廠設備資料。如需如何為 AWS IoT SiteWise 資料建立資料存放區的詳細資訊，請參閱AWS IoT SiteWise 《使用者指南》中的設定儲存設定。

> **Note**
>
> 建立後， AWS IoT SiteWise 中繼資料很快就會出現在 AWS IoT SiteWise 資料存放區中；不過，原始資料最多可能需要六個小時才會出現。同時，您可以建立 AWS IoT Analytics 資料集，並在中繼資料上執行查詢。

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，可模擬風力發電廠中的風力引擎渦輪機。

> **Note**
>
> 在本教學課程中，您將在資料存放區中查詢三個資料表：
>
> - `raw` - 包含每個資產的原始未處理資料。
> - `asset_metadata` - 包含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。

執行本教學課程中的 SQL 查詢

1. 遵循 使用 AWS IoT SiteWise 資料建立資料集 （主控台）或 中的步驟使用 AWS IoT SiteWise 資料建立資料集 (AWS CLI)，為您的 AWS IoT SiteWise 資料建立 AWS IoT Analytics 資料集。
2. 若要在本教學課程中更新您的資料集查詢，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 SQL 查詢。

   c. 若要在查詢後的資料表中顯示結果，請選擇測試查詢。

   或者，您可以使用 執行下列update-dataset命令來修改 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，對資料執行下列查詢，以確認您的asset_metadata資料表已成功載入。

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

   同樣地，您可以驗證 asset_hierarchy_metadata和 raw資料表不是空的。

後續步驟

資料探勘

資料探勘

建立 AWS IoT SiteWise 資料並載入資料存放區後，您可以在 中建立 AWS IoT Analytics 資料集並執行 SQL 查詢 AWS IoT Analytics ，以探索資產的洞見。下列查詢示範如何在執行統計查詢之前探索資料。

使用 SQL 查詢探索您的資料

1. 檢視每個資料表中的資料欄和值範例，例如原始資料表中的 。

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

2. 使用 SELECT DISTINCT 查詢asset_metadata資料表並列出 AWS IoT SiteWise 資產的 （唯一） 名稱。

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

3. 若要列出特定 AWS IoT SiteWise 資產屬性的相關資訊，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，您可以從資料存放區中的兩個或多個資料表聯結資料，例如下列範例。

```
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

現在您已探索 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洞見。下列查詢示範的是一些您可以擷取的資訊。

對 AWS IoT SiteWise 示範風力發電廠資料執行統計查詢

1. 執行下列 SQL 命令，以特定資產 （ 示範渦輪資產 4) 的數值尋找所有屬性的最新值。

```
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. 加入中繼資料資料表和原始資料表，以識別除了其父系資產之外，所有資產的最大風速屬性。

```
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 （示範渦輪資產 2) 特定屬性 （風速） 的平均值，請執行下列 SQL 命令。您必須my_bucket_id以儲存貯體的 ID 取代 。

```
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

[清除您的教學課程資源]

清除您的教學課程資源

完成教學課程後，請清除資源以避免產生費用。

刪除您的 AWS IoT SiteWise 示範

AWS IoT SiteWise 示範會在一週後自行刪除。如果您已完成使用示範資源，則可以提早刪除示範。若要手動刪除示範，請使用下列步驟。

1. 導覽至 CloudFormation 主控台。
2. IoTSiteWiseDemoAssets 從 Stacks 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)。如果示範再次無法刪除，請依照 CloudFormation 主控台中的步驟略過無法刪除的資源，然後再試一次。

刪除資料存放區

- 若要刪除受管資料存放區，請執行 CLI 命令 `delete-datastore`，例如下列範例中的 。

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

刪除 AWS IoT Analytics 資料集

- 若要刪除資料集，請執行 CLI 命令 `delete-dataset`，例如下列範例中的 。在執行此操作之前，您不需要刪除資料集的內容。

```
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 操作來模擬在您提供的訊息承載上執行管道活動的結果。在開發和偵錯管道活動時，這可能相當實用。RunPipelineActivity 範例示範如何使用。

頻道活動

管道中的第一個活動必須是決定要處理之訊息來源channel的活動。

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

資料存放區活動

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 操作輸出的資料來充實訊息，或根據來自 Amazon DynamoDB 的邏輯篩選訊息。不過，在進入資料存放區之前，您無法使用此管道活動來新增其他訊息，或移除現有的訊息。

lambda 活動中使用的 AWS Lambda 函數必須接收並傳回 JSON 物件陣列。如需範例，請參閱「the section called “Lambda 函數範例 1”」。

若要授予叫用 Lambda 函數的 AWS IoT Analytics 許可，您必須新增政策。例如，執行下列 CLI 命令，並將 *exampleFunctionName* 取代為 Lambda 函數的名稱，將 *123456789012* 取代為 AWS 您的帳戶 ID，並使用呼叫指定 Lambda 函數之管道的 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
```

命令會傳回下列項目：

```
{
    "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\"}}}"
}
```

詳情請參閱 AWS Lambda 開發人員指南中的為 AWS Lambda使用資源型政策。

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。裝置會發佈承載類似下列範例的訊息。

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

裝置具有下列管道定義。

```
{
    "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"
    }
}
```

下列 Lambda Python 函數 (`MyAnalyticsLambdaFunction`) 會將 GMaps URL 和以華氏為單位的溫度新增至訊息。

```
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。在此第二個範例中，Lambda 函數假設訊息承載代表已壓縮的 JSON 原始檔，然後以 base64 編碼（序列化）做為字串。它會傳回原始 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。這可讓您在儲存訊息之前變更訊息的形狀。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。

請考慮下列輸入訊息。

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

addAttributes 活動如下所示。

```
{
    "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陣列中的值，將它們提升為名為 lat和 的頂層屬性lon。由於此活動，輸入訊息會轉換為下列範例。

```
{
    "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，假設訊息是addAttributes活動的結果。

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

若要將訊息標準化，使其只包含根層級的必要資料，請使用下列removeAttributes活動。

```
{
```

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

這會導致下列訊息沿著管道流動。

```
{
    "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"
    }
}
```

結果是以下訊息流經管道。

```
{
    "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 SQL WHERE子句，必須傳回布林值。

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

DeviceRegistryEnrich 活動

deviceRegistryEnrich 活動可讓您將資料從 AWS IoT 裝置登錄檔新增至訊息承載。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。角色必須具有如下所示的許可政策。

JSON

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

```
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

該角色的信任政策則如下所示：

JSON

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

DeviceShadowEnrich 活動

`deviceShadowEnrich` 活動會將 AWS IoT Device Shadow 服務中的資訊新增至訊息。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"
    }
}
```

結果是類似下列範例的訊息。

```
{
    "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。角色必須具有如下所示的許可政策。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
            ],
            "Resource": [
            "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

該角色的信任政策則如下所示：

JSON

```
{
    "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)
- trunc(Decimal，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` .

引數類型	結果
String	Decimal （使用雙精度） 引數的反餘弦。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 .

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 .

引數類型	結果
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。逆時針角度的角度為正值 （ 上半平面，y > 0)，順時針角度引Decimal數的負值在套用函數之前會四捨五入為雙精確度。

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 (x，y) 點之間的角度
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，字串會轉換為 Decimal，並四捨五入至最接近的 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 。
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 .

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

exp(Decimal)

傳回e增加到十進位引數。 Decimal引數在套用函數之前會四捨五入到雙精確度。

範例： exp(1) = 1

引數類型	結果
Int	Decimal （具有雙精度）、e^argument。
Decimal	Decimal （使用雙精度）、e^argument
String	Decimal （具有雙精度）、e^argument。如果 String 無法轉換為 Decimal，則結果為 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），引數的自然日誌
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 。

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

mod(Decimal, Decimal)

傳回第二個引數中第一個引數分割的剩餘部分。您也可以使用 %做為相同模數功能的固定運算子。

範例： mod(8, 3) = 2

左運算元	右運算元	輸出
Int	Int	Int，第二個引數的第一個引數模數。
Int / Decimal	Int / Decimal	Decimal，第二個引數的第一個引數模數。
String / Int / Decimal	String / Int / Decimal	如果所有字串都轉換為 Decimals，則第一個引數模數為第二個引數時的結果。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) = 32.0

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。
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。

引數類型	結果
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 。
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Decimal值，則會傳回Decimal值的符號。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。
Decimal	Decimal (使用雙精度)，引數的正弦值。
Boolean	Undefined 。
String	Decimal，引數的正弦。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。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 .

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。
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 。

引數類型	結果
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

trunc(Decimal，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。追蹤零會從結果中分割。

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。字串會轉換為Decimal值。如果字串轉換失敗，則結果為 Undefined 。
其他值		未定義。

RunPipelineActivity

以下是如何使用 RunPipelineActivity命令來測試管道活動的範例。在此範例中，我們會測試數學活動。

1. 建立 maths.json 檔案，其中包含您要測試之管道活動的定義。

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

2. 建立 payloads.json 檔案，其中包含用於測試管道活動的範例承載。

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

3. 從命令列呼叫 RunPipelineActivities操作。

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

這會產生下列結果。

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

結果中列出的承載是 Base64-encoded字串。當這些字串解碼時，您會得到以下結果。

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

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

重新處理頻道訊息

AWS IoT Analytics 可讓您重新處理頻道資料。這在下列情況下很有用：

- 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，並想將現有的資料更新至最新變更。
- 您想要包含在您變更客戶受管儲存選項、頻道許可或資料存放區之前擷取的資料。

參數

當您透過管道使用 重新處理頻道訊息時 AWS IoT Analytics，您必須指定下列資訊：

`StartPipelineReprocessing`

開始透過管道重新處理頻道訊息。

`ChannelMessages`

指定您要重新處理的一組或多組頻道訊息。

如果您使用 `channelMessages` 物件，則不得指定 `startTime`和 的值`endTime`。

`s3Paths`

指定一或多個金鑰，以識別儲存頻道訊息的 Amazon Simple Storage Service (Amazon S3) 物件。您必須使用金鑰的完整路徑。

路徑範例：
`00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.jso`

類型：字串陣列

陣列成員限制條件：1-100 個項目。

長度限制：1-1024 個字元。

`endTime`

重新處理之頻道資料的結束時間 （專屬）。

如果您指定 `endTime` 參數的值，則不得使用 `channelMessages` 物件。

類型：Timestamp

`startTime`

經重新處理的原始訊息資料開始時間 (包含)。

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

類型：Timestamp

`pipelineName`

要開始重新處理的管道名稱。

類型：字串

長度限制條件：1-128 個字元。

重新處理頻道訊息（主控台）

本教學課程說明如何重新處理存放在 AWS IoT Analytics 主控台中指定 Amazon S3 物件中的頻道資料。

開始之前，請確定您要重新處理的頻道訊息儲存在客戶受管的 Amazon S3 儲存貯體中。

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. 從動作中選擇重新處理訊息。
5. 在管道重新處理頁面上，選擇重新處理訊息的 S3 物件。

   AWS IoT Analytics 主控台也提供下列選項：

   - 所有可用範圍 - 重新處理頻道中的所有有效資料。
   - 過去 120 天 - 重新處理過去 120 天內抵達的資料。
   - 過去 90 天 - 重新處理過去 90 天內抵達的資料。
   - 過去 30 天 - 重新處理過去 30 天內抵達的資料。
   - 自訂範圍 - 重新處理抵達指定時間範圍的資料。您可以選擇任何時間範圍。
6. 輸入存放頻道訊息的 Amazon S3 obejct 金鑰。

若要尋找金鑰，請執行下列動作：

a. 前往 Amazon S3 主控台。
b. 選擇目標 Amazon S3 物件。
c. 在屬性的物件概觀區段中，複製金鑰。

7. 選擇開始重新處理。

重新處理頻道訊息 (API)

當您使用 `StartPipelineReprocessing` API 時，請注意下列事項：

- `startTime` 和 `endTime` 參數會指定原始資料擷取的時間，但這些是粗略的預估值。您可以四捨五入到最接近的 1 小時。`startTime` 包含 ，但 `endTime`是唯一的。
- 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。
- 您每 24 小時最多可以提出 1000 個 `StartPipelineReprocessing` API 請求，以透過管道重新處理相同的頻道訊息。
- 重新處理您的原始資料會產生額外費用。

如需詳細資訊，請參閱 API 參考中的 StartPipelineReprocessing API。 AWS IoT Analytics

取消頻道重新處理活動

若要取消管道重新處理活動，請使用 CancelPipelineReprocessing API，或在 AWS IoT Analytics 主控台的活動頁面上選擇取消重新處理。如果您取消重新處理，則不會重新處理剩餘的資料。您必須開始另一個重新處理請求。

使用 DescribePipeline API 檢查重新處理的狀態。請參閱回應中的 `reprocessingSummaries` 欄位。

自動化您的工作流程

AWS IoT Analytics 提供 的進階資料分析 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。您可以執行容器來託管自己的自訂分析程式碼或 Jupyter 筆記本，或使用第三方自訂程式碼容器，這樣您就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：

依週期性排程建立資料集內容

呼叫 `CreateDataset()` 時指定觸發條件，以排程自動建立資料集內容`triggers:schedule:expression`。資料存放區中具有 的資料用於建立資料集內容。您可以使用 SQL 查詢 () 選取您想要的欄位`actions:queryAction:sqlQuery`。

定義非重疊的連續時間間隔，以確保新資料集內容僅包含自上次到達的資料。使用 `actions:queryAction:filters:deltaTime`和 `:offsetSeconds` 欄位來指定差異時間間隔。然後指定觸發，以在經過時間間隔後建立資料集內容。請參閱 the section called “範例 6 -- 使用差異視窗 (CLI) 建立 SQL 資料集”。

在完成另一個資料集時建立資料集內容

當另一個資料集的內容建立完成時，觸發建立新資料集內容`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應用程式執行。如此一來，您就可以使用重複排程建立之資料集內容的資料來饋送應用程式。您可以從應用程式內自動對分析結果採取動作。 (`actions:containerAction`)

在完成另一個資料集時建立資料集內容

當另一個資料集的內容建立完成時，觸發建立新資料集內容`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應用程式執行。如此一來，您就可以使用重複排程建立之資料集內容的資料來饋送應用程式。您可以從應用程式內自動對分析結果採取動作。 (`actions:containerAction`)

使用案例

自動化產品品質測量，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。系統會定期和在某些事件發生時整理事件，例如當值開啟和關閉時。使用 AWS IoT Analytics，您可以自動化從這些定期時段彙總非重疊資料的分析，並建立終端產品品質的 KPI 報告。處理每個批次之後，您會測量整體產品品質，並透過最大化的執行量降低營運支出。

將裝置機群的分析自動化

您可以每 15 分鐘對 100 個裝置所產生的資料執行分析 （演算法、資料科學或 ML for KPI)。每個分析週期都會產生和儲存狀態，以供下次分析執行。對於每個分析，您只想要使用指定時段內收到的資料。透過 AWS IoT Analytics ，您可以協調分析並建立每次執行的 KPI 和報告，然後存放資料以供未來分析。

將異常偵測自動化

AWS IoT Analytics 可讓您針對已送達資料存放區的新資料，每 15 分鐘手動執行一次，以自動化異常偵測工作流程。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，包括可用的程序測量，您可以操作分析工作流程來預測程序結果。模型的資料可以 M x N 矩陣形式排列，其中每一列都包含從實驗室樣本的各個時間點取得的資料。 透過建立差異視窗並使用資料科學工具來建立 KPIs並儲存測量裝置的狀態， AWS IoT Analytics 協助您操作分析工作流程。

使用 Docker 容器

本節包含如何建置您自己的 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。

   IotAnalytics Jupyter 擴充功能提供容器化 API 來協助容器化程序。您也可以執行自己的建立映像，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運算。 AWS IoT Analytics 可讓

您透過變數定義 Docker 容器之輸出資料的輸入資料來源和目的地。(自訂 Docker 容器輸入/輸出變數包含有關搭配自訂容器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存放來自裝置 () 的訊息 （ 資料iotanalytics: CreateDatastore)
4. 建立傳送訊息的頻道 (iotanalytics: CreateChannel)。
5. 建立管道，將頻道連線至資料存放區 (iotanalytics: CreatePipeline)。
6. 建立 IAM 角色，授予將訊息資料傳送至 AWS IoT Analytics 頻道的許可 (iam: CreateRole.)
7. 建立使用 SQL 查詢將頻道連線至訊息資料來源的 IoT 規則 (iot: CreateTopicRule 欄位 topicRulePayload:actions:iotAnalytics)。當裝置使用適當的主題 Visa MQTT 傳送訊息時，它會路由到您的頻道。或者，您可以使用 從能夠使用 AWS SDK 或 的裝置，iotanalytics: BatchPutMessage將訊息直接傳送到頻道 AWS CLI。
8. 建立由時間排程 (iotanalytics: CreateDataset, 欄位) 觸發建立的 SQL 資料集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。（ 欄位actions:queryAction:filters:deltaTime:timeExpression提供可決定訊息時間的表達式。 欄位actions:queryAction:filters:deltaTime:offsetSeconds指定訊息到達時的可能延遲。)

   預先篩選與觸發排程會決定您的差異時段。每個新的 SQL 資料集都是使用自上次建立 SQL 資料集以來收到的訊息來建立。（ 第一次建立 SQL 資料集會如何？ 上次建立資料集的估計時間是根據排程和預先篩選。)
9. 建立另一個由建立第一個 (CreateDataset 欄位 trigger:dataset) 觸發的資料集。對於此資料集，您可以指定容器動作 （ 已提交 actions:containerAction)，指向您在第一個步驟中建立的 Docker 容器，並提供執行所需的資訊。您還可以指定：

   - 存放在您帳戶中之 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.)
   - 如果運算資源用於執行容器動作 (computeType具有可能值：)，則為 類型ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]。
   - 用於執行容器動作 () 的資源執行個體可用的持久性儲存體大小 (GBvolumeSizeInGB)。
   - 在應用程式執行內容中使用的變數值 （ 基本上是傳遞給應用程式的參數 ） ()variables。

這些變數會在執行容器時進行替換。這可讓您執行具有不同變數（參數）的相同容器，這些變數會在建立資料集內容時提供。IotAnalytics Jupyter 擴充功能簡化了此程序，方式是透過自動識別筆記本中的變數，並讓這些變數可在容器化程序中使用。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 查詢和差異視窗產生資料集，以及使用應用程式的容器， AWS IoT Analytics 會建立排程生產資料集，以您在差異視窗中指定的資料間隔執行，產生所需的輸出和傳送通知。

您可以暫停生產資料集應用程式，並在選擇這麼做時繼續執行。當您繼續生產資料集應用程式時 AWS IoT Analytics，根據預設， 會追上自上次執行後抵達但尚未分析的所有資料。您也可以透過執行一系列的連續執行，設定您希望繼續生產資料集任務時段長度的方式）。或者，您可以僅擷取符合差異視窗指定大小的新到達資料，以繼續生產資料集應用程式。

建立或定義由建立另一個資料集所觸發的資料集時，請注意下列限制：

- SQL 資料集只能觸發容器資料集。
- SQL 資料集最多可以觸發 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的 SQL 資料集觸發的容器資料集，就會發生此錯誤。
- 「觸發資料集 <dataset-name> 無法由容器資料集觸發」

  如果您嘗試定義由另一個容器資料集觸發的另一個容器資料集，就會發生此錯誤。
- 「<N> 資料集已相依於 <dataset-name> 資料集。」

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

  發生此錯誤時，您嘗試定義由排程觸發和資料集觸發觸發的資料集。

自訂 Docker 容器輸入/輸出變數

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`。以下是該檔案內容的範例。

```
{
   "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。以下是開啟 `params` 檔案、剖析檔案，以及列印 `example_var`變數值的範例程式。

```
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)
```

上傳輸出

Docker 映像啟動的程式也可能將其輸出存放在 Amazon S3 位置。輸出必須使用 "bucket-owner-full-control" 存取控制清單載入。存取清單會授予對上傳輸出 AWS IoT Analytics 的服務控制。在此範例中，我們延伸先前的 ，將 的內容上傳至 `params` 檔案中由 `example_var` 定義的 Amazon S3 `custom_output` 位置。

```
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。一個角色會授予啟動 SageMaker AI 執行個體的許可，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。如果您使用 AWS IoT Analytics 主控台建立新的 SageMaker AI 執行個體，您可以選擇自動建立新的角色，以授予執行 SageMaker AI 執行個體和容器化筆記本所需的所有權限。或者，您可以手動使用這些權限建立角色。若要這樣做，請建立已連接 `AmazonSageMakerFullAccess` 政策的角色，並新增下列政策。

JSON

```
{
```

```
  "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 AWS IoT Analytics 主控台自動建立第一個角色，仍必須執行此操作。建立已連接下列政策和信任政策的角色。

JSON

```
{
    "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。

JSON

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

透過 Java 和 使用 CreateDataset API AWS CLI

建立資料集。資料集會套用 queryAction(SQL 查詢） 或 containerAction（執行容器化應用程式）來存放從資料存放區擷取的資料。此操作會建立資料集的骨架。您可以呼叫 CreateDatasetContent或根據您trigger指定的 自動填入資料集。如需詳細資訊，請參閱 CreateDataset 和 CreateDatasetContent。

主題

- 範例 1 -- 建立 SQL 資料集 (java)
- 範例 2 -- 建立具有差異視窗的 SQL 資料集 (java)
- 範例 3 -- 使用自己的排程觸發條件建立容器資料集 (java)
- 範例 4 -- 建立以 SQL 資料集做為觸發條件的容器資料集 (java)
- 範例 5 -- 建立 SQL 資料集 (CLI)
- 範例 6 -- 使用差異視窗 (CLI) 建立 SQL 資料集

範例 1 -- 建立 SQL 資料集 (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}}
```

範例 2 -- 建立具有差異視窗的 SQL 資料集 (java)

```
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條件建立容器資料集 (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 -- 建立以 SQL 資料集做為觸發條件的容器資料集 (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 -- 使用差異視窗 (CLI) 建立 SQL 資料集

Delta 時段是一系列使用者定義、不重疊且持續的時間間隔。Delta Windows 可讓您使用自上次分析以來抵達資料存放區的新資料建立資料集內容，並對其進行分析。您可以透過在資料集 queryAction (CreateDataset) 的 filters部分deltaTime中設定 來建立差異視窗。通常，您還需要設定時間間隔觸發 () 來自動建立資料集內容triggers:schedule:expression。基本上，這可讓您篩選在特定時段內到達的訊息，因此先前時段訊息中包含的資料不會計算兩次。

在此範例中，我們會建立一個新的資料集，每 15 分鐘使用自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，則使用的資料（包含在資料集內容中）會是時間戳記介於上午 10：12 到上午 10：27 之間（即上午 10：30 - 15 分鐘 - 3 分鐘到上午 10：30 - 3 分鐘）。

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

其中檔案delta-window.json包含下列項目。

```
{
  "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 筆記本建置 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 主控台中，在顯示 HTML 的電腦上提供潛在的攻擊向量。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 筆記本提供強大的資料科學工具，可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。(請注意，我們目前不支援 JupyterLab 內的容器化。) 您可以將 Jupyter 筆記本和程式庫封裝到容器，該容器會在您定義的差異時段 AWS IoT Analytics 內收到新批次的資料時定期執行。您可以排程使用容器的分析任務，以及在指定時段內擷取的新分段資料，然後存放任務的輸出以供未來排程的分析。

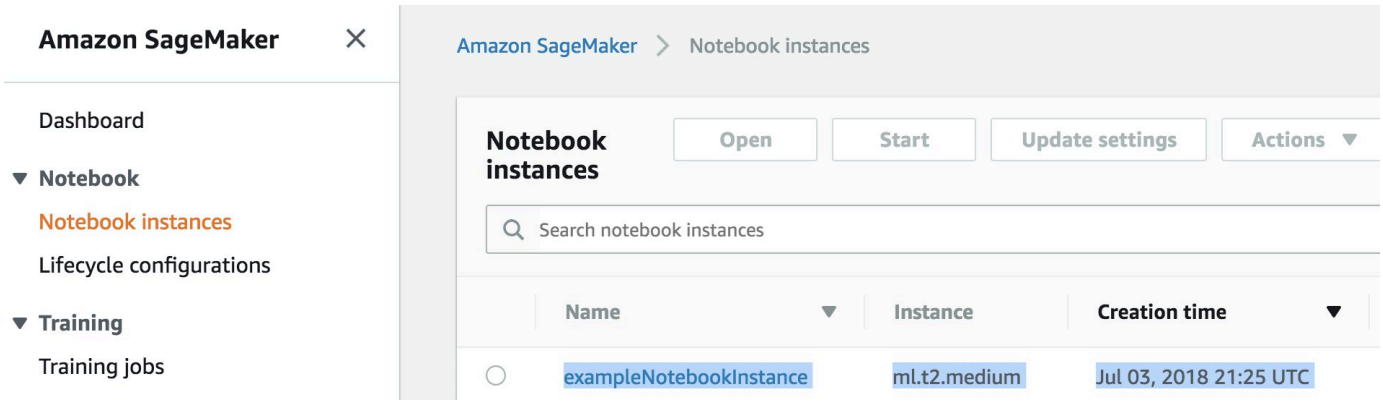
如果您在 2018 年 8 月 23 日之後使用 AWS IoT Analytics 主控台建立 SageMaker AI 執行個體，則容器化擴充功能的安裝已自動為您完成，您可以開始建立容器化映像。否則，請遵循本節中列出的步驟，在 SageMaker AI 執行個體上啟用筆記本容器化。接下來，您將修改 SageMaker AI 執行角色，以允許您將容器映像上傳至 Amazon EC2，並安裝容器化擴充功能。

啟用非透過 AWS IoT Analytics 主控台建立之筆記本執行個體的容器化

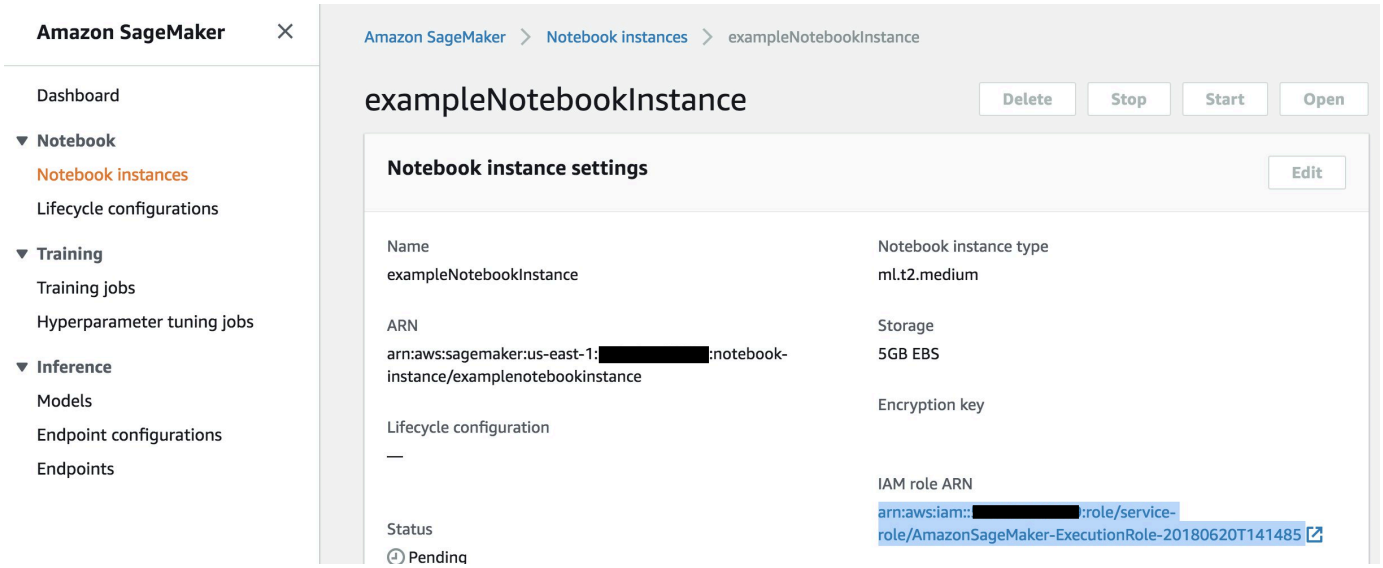
我們建議您透過 AWS IoT Analytics 主控台建立新的 SageMaker AI 執行個體，而不是遵循這些步驟。新的執行個體自動支援容器化。

如果您在如此處所示啟用容器化後重新啟動 SageMaker AI 執行個體，則不需要重新新增 IAM 角色和政策，但必須重新安裝擴充功能，如最終步驟所示。

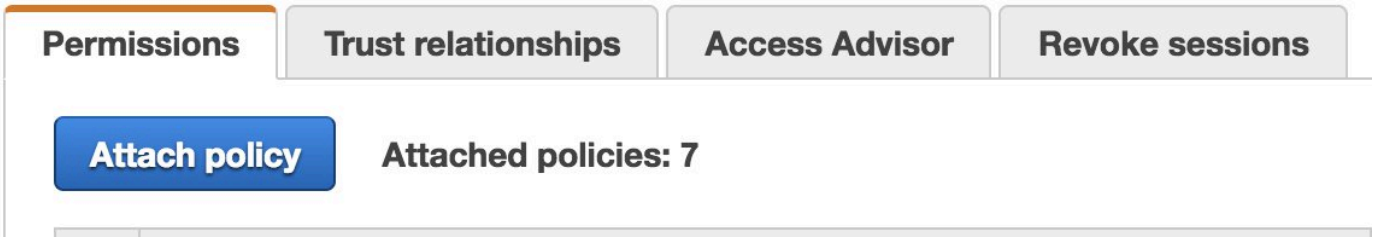
1. 若要授予筆記本執行個體對 Amazon ECS 的存取權，請在 SageMaker AI 頁面上選取您的 SageMaker AI 執行個體：



2. 在 IAM 角色 ARN 下，選擇 SageMaker AI 執行角色。

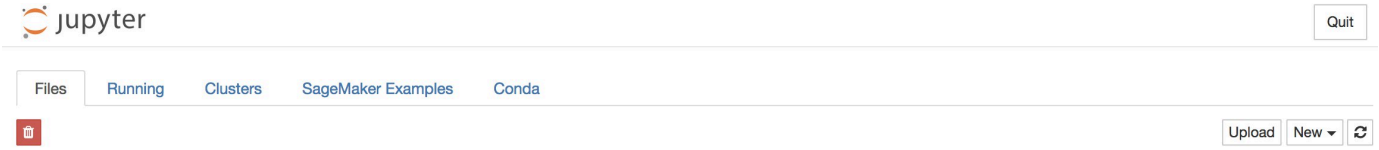


3. 選擇 Attach Policy (連接政策)，然後定義並連接 Permissions (許可)中所顯示的政策。如果`AmazonSageMakerFullAccess`政策尚未連接，也請連接它。



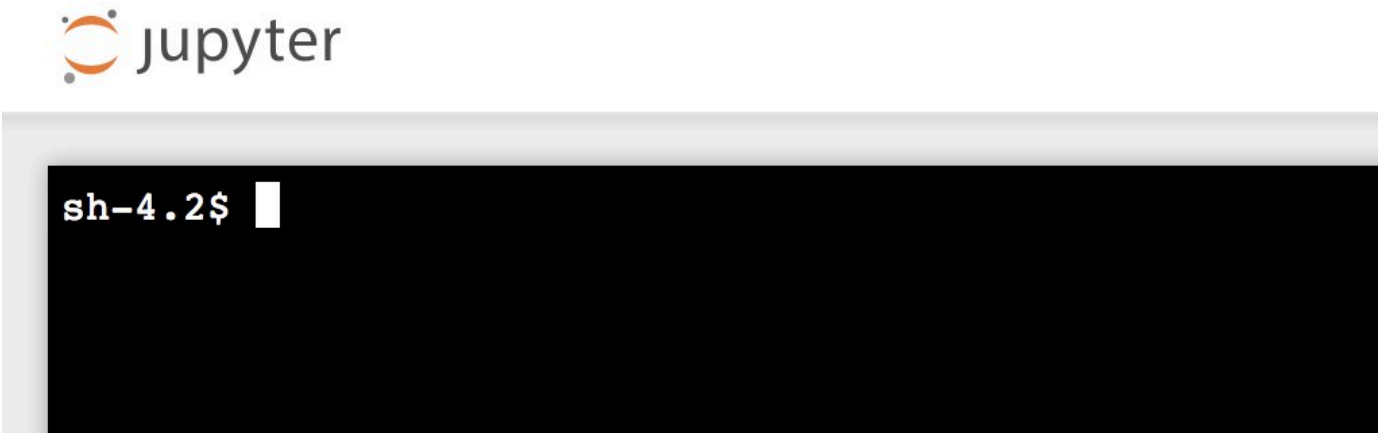
您還必須從 Amazon S3 下載容器化程式碼，並將其安裝在筆記本執行個體上，第一步是存取 SageMaker AI 執行個體的終端機。

1. 在 Jupyter 中，選擇新增。



2. 在出現的功能表中，選擇終端機。

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，這些命令會終止筆記本在此 SageMaker AI 執行個體上執行的任何程序。




```
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。

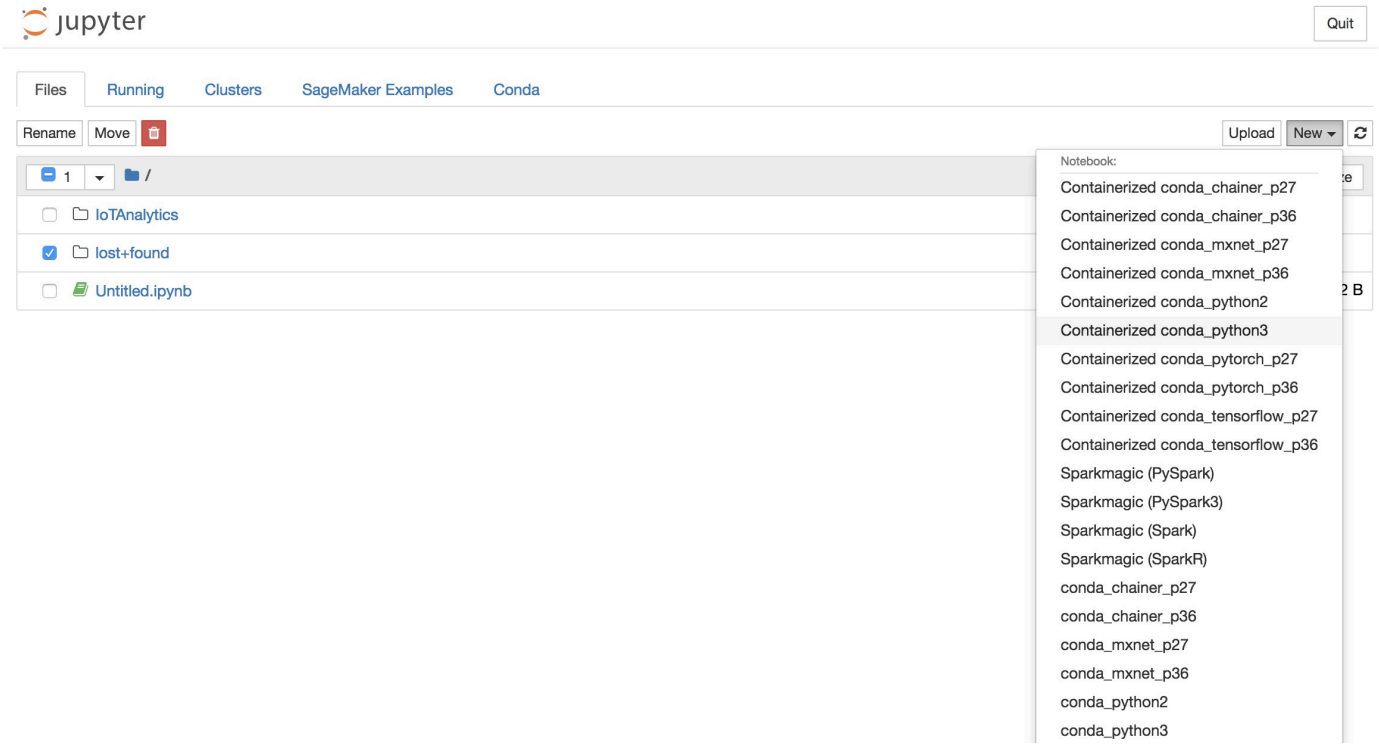
更新您的筆記本容器化擴充功能

如果您在 2018 年 8 月 23 日之後透過 AWS IoT Analytics 主控台建立 SageMaker AI 執行個體，則會自動安裝容器化擴充功能。您可以從 SageMaker AI 主控台重新啟動執行個體來更新擴充功能。如果您手動安裝擴充功能，則可以重新執行在啟用未透過 AWS IoT Analytics 主控台建立的筆記本執行個體容器化中列出的終端機命令來更新擴充功能。

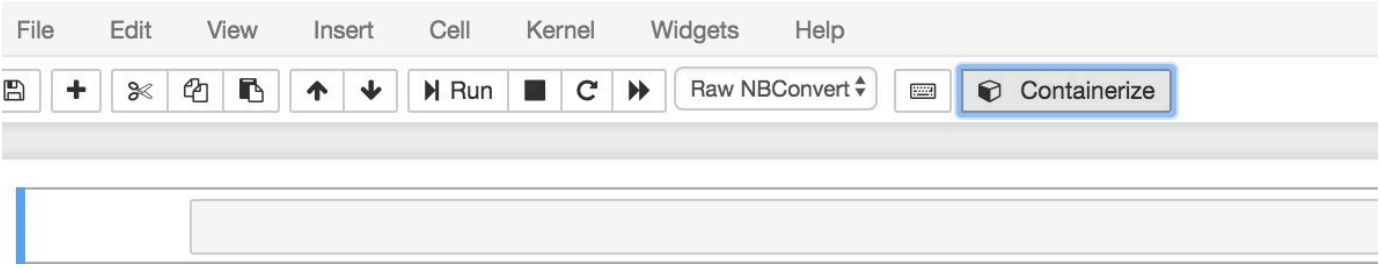
建立容器化映像

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。

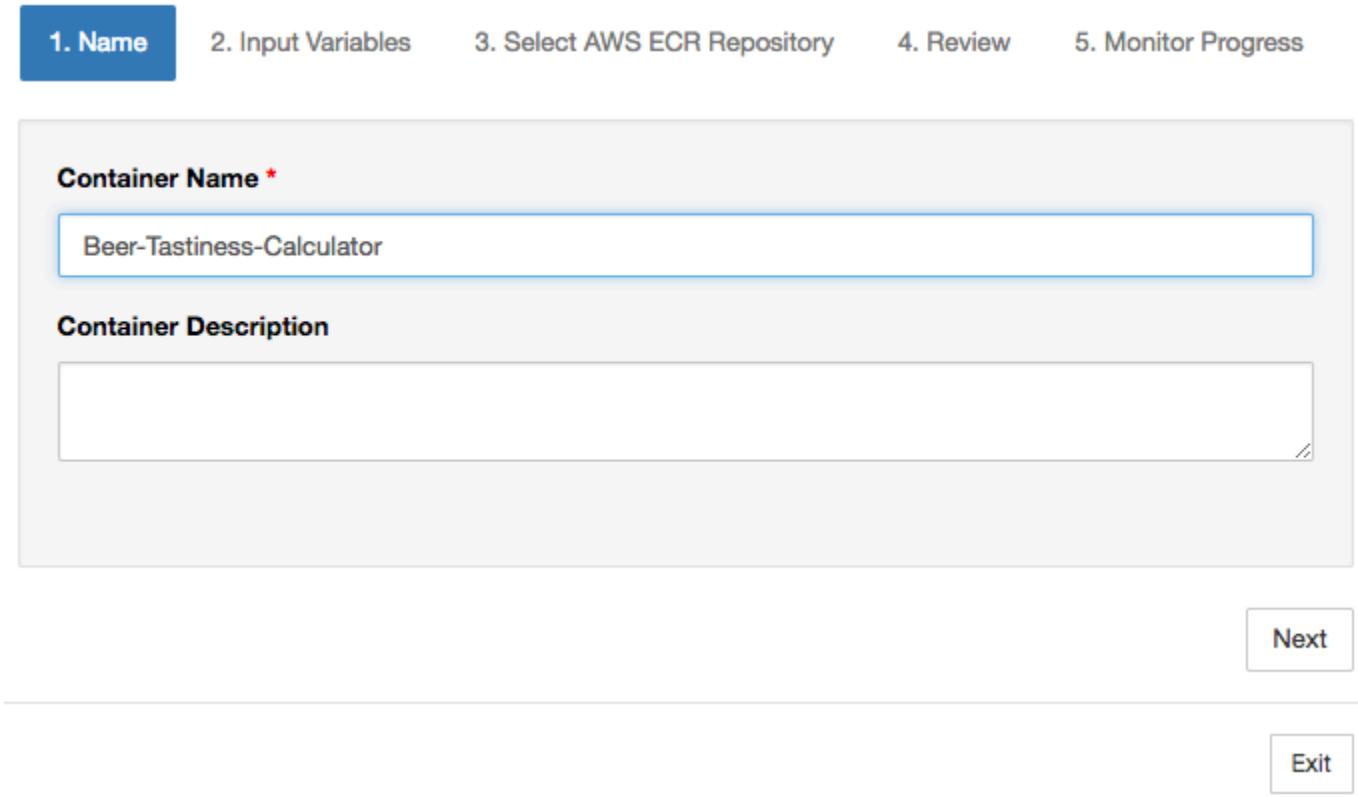
1. 在您的 Jupyter 筆記本中，選擇 New (新增)，然後從下拉式清單選擇您想要的核心類型。（核心類型應以「容器化」開頭，並以您原本選擇的任何核心結尾。 例如，如果您只想要像是 "conda_python3" 的純 Python 3.0 環境，請選擇 "Containerized conda_python3")。



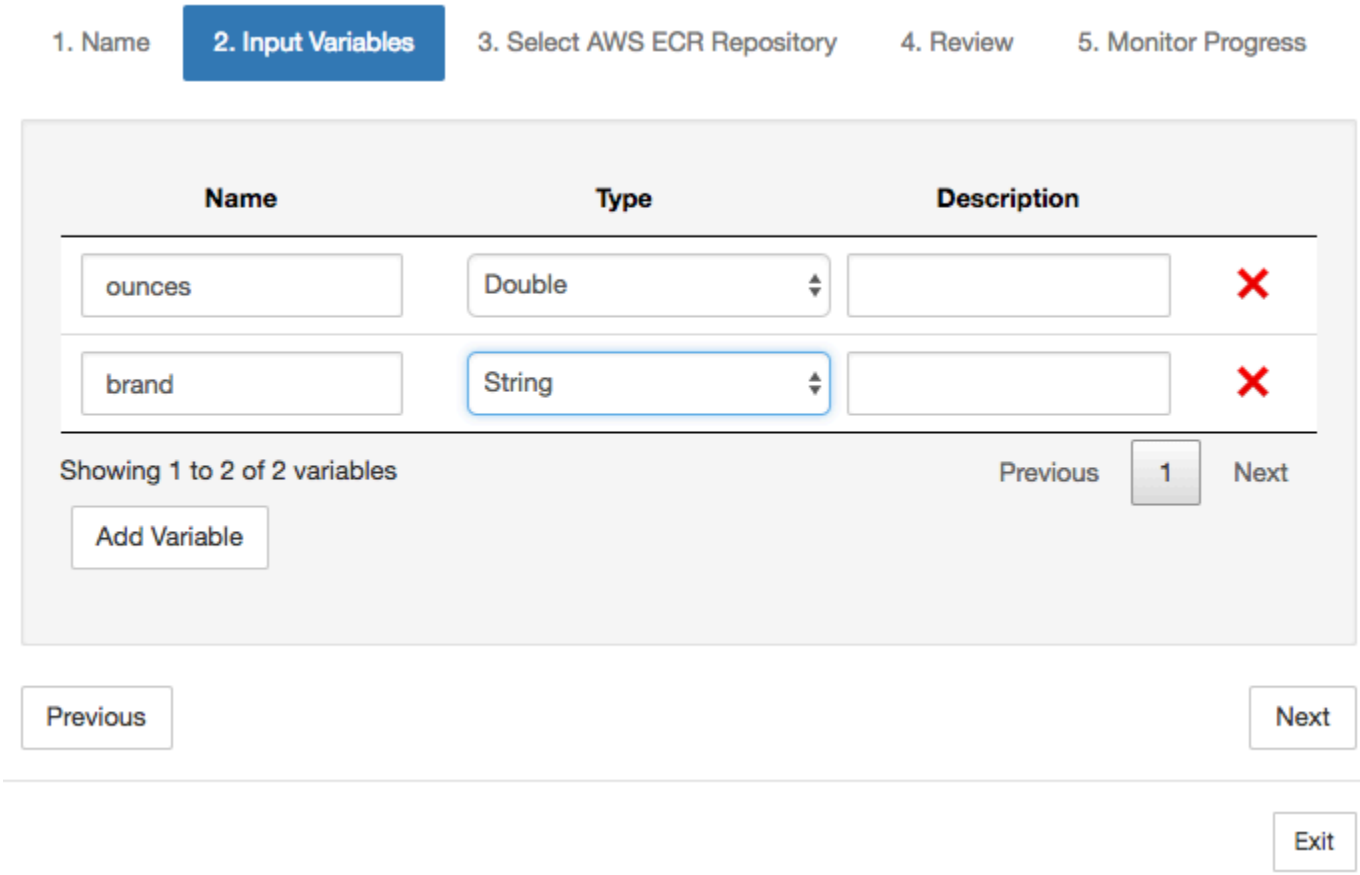
2. 在筆記本上完成工作並想要容器化後，請選擇容器化。



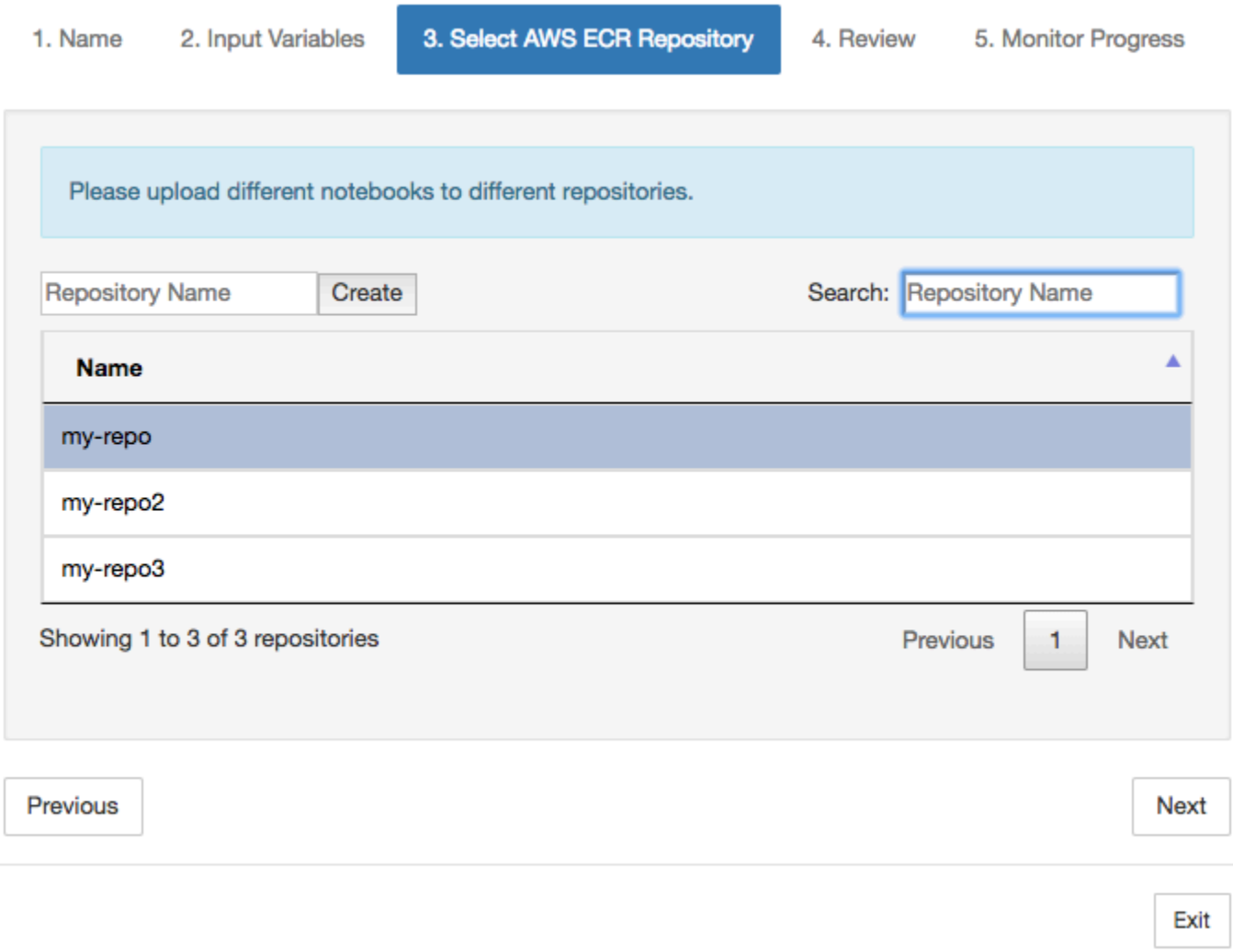
3. 輸入容器化筆記本的名稱。您也可以選擇輸入描述。



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用描述。

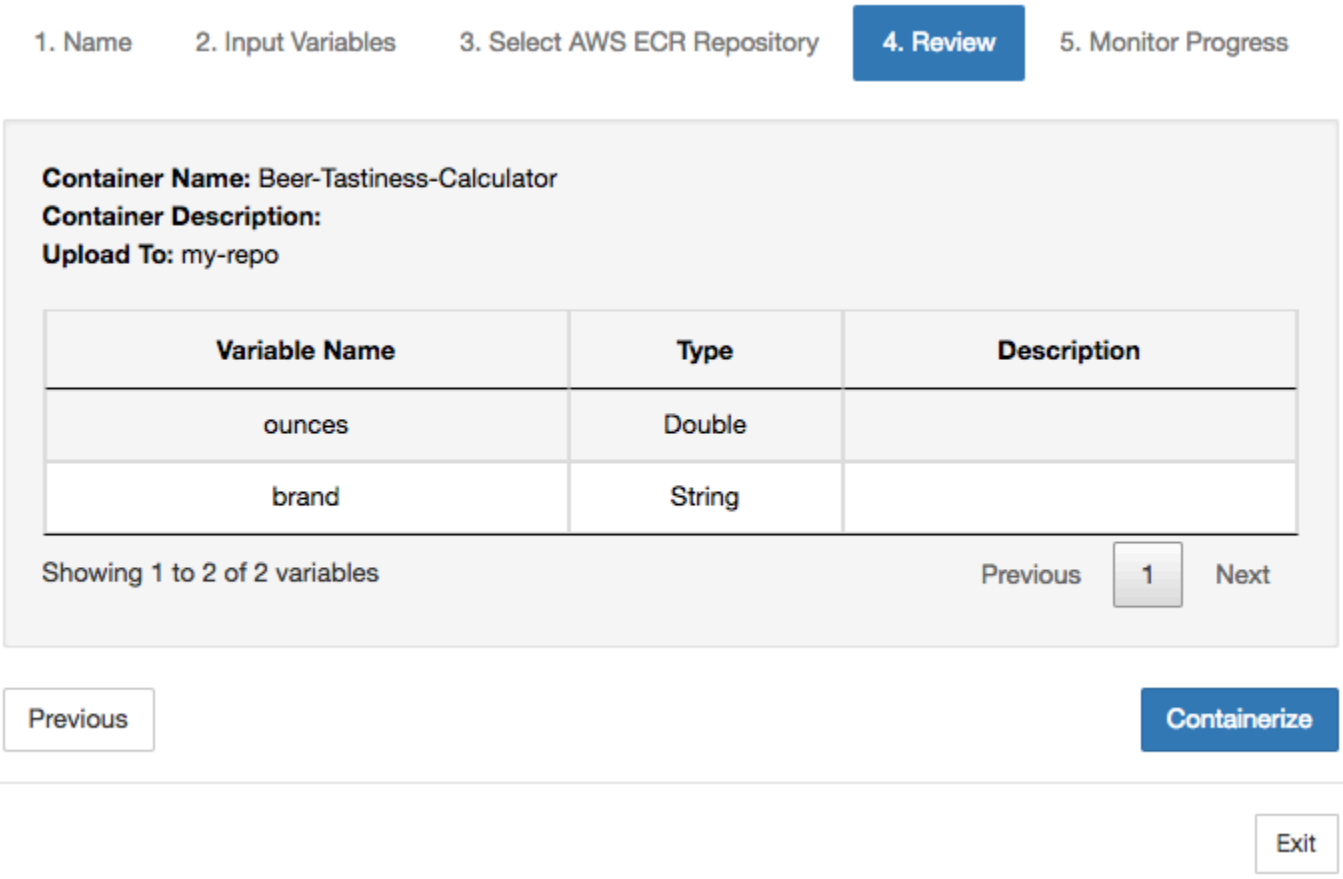


5. 選擇從筆記本建立的映像應上傳的 Amazon ECR 儲存庫。

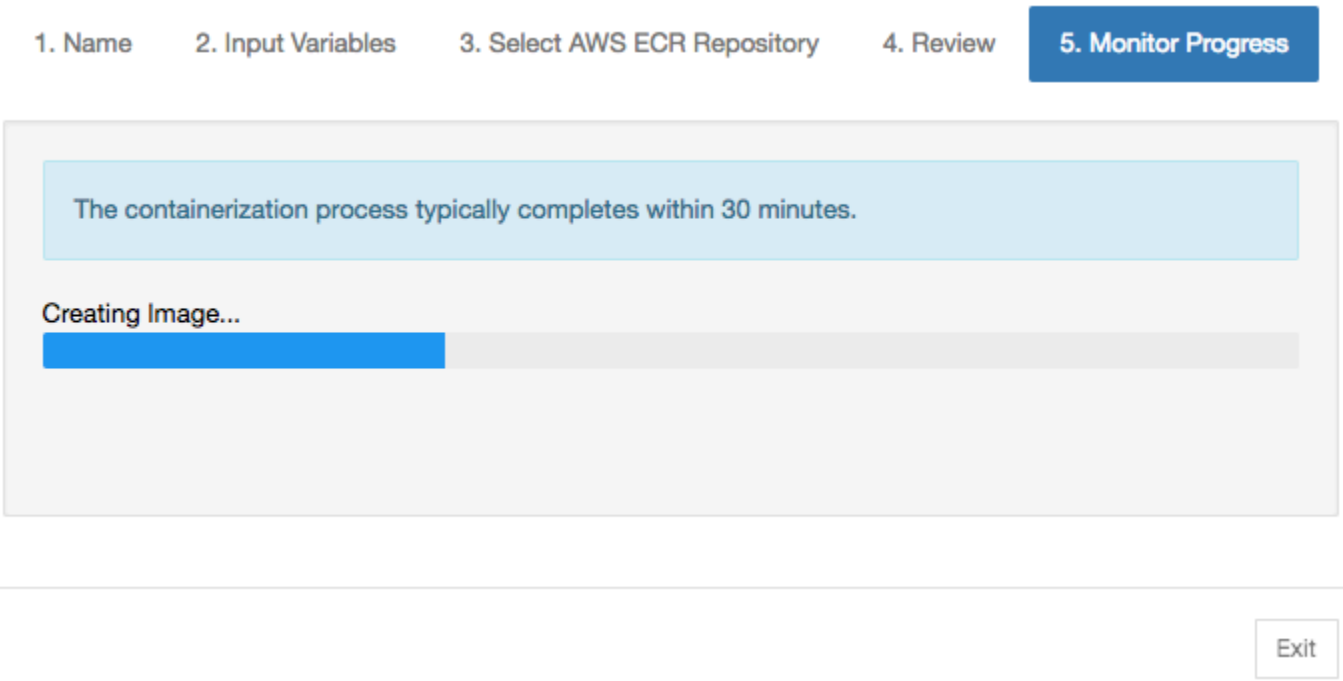


6. 選擇容器化以開始程序。

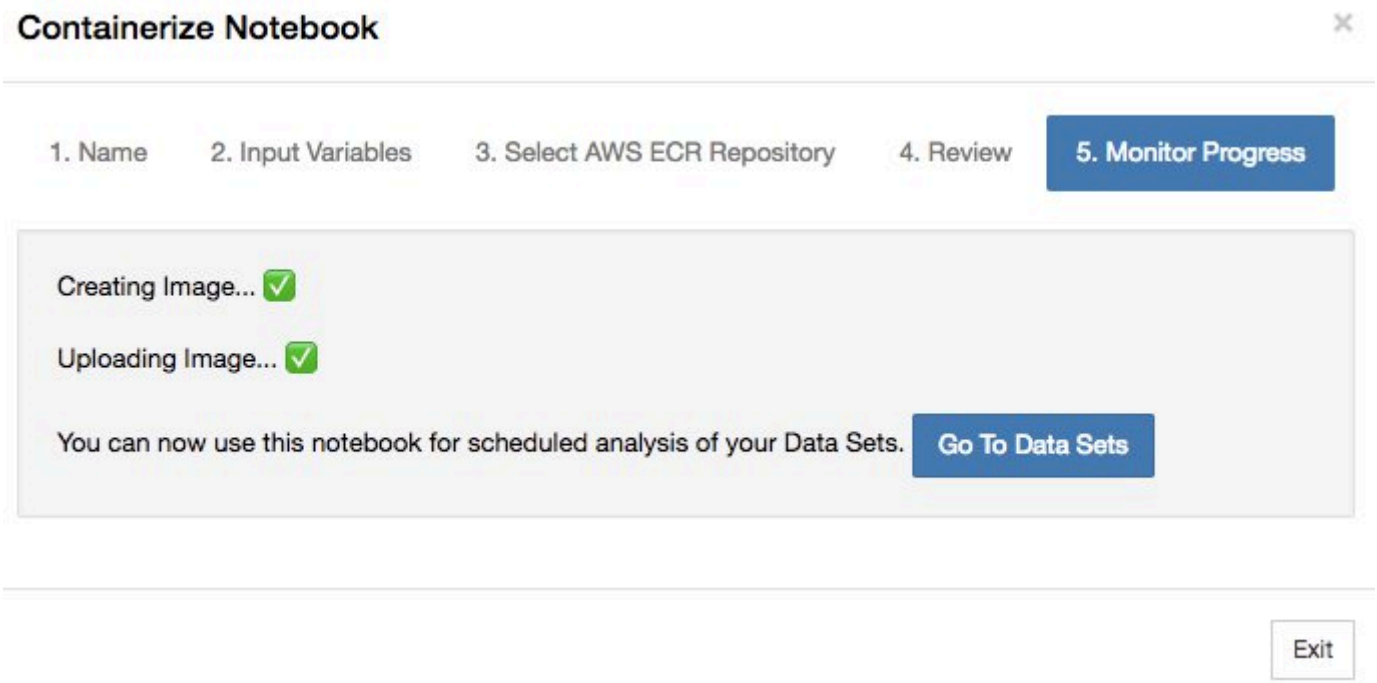
   您會收到摘要輸入內容的概觀。請注意，在您開始程序之後，就無法取消它。程序可能持續長達一小時。


1. Name
2. Input Variables
3. Select AWS ECR Repository
4. Review
5. Monitor Progress
Container Name: Beer-Tastiness-Calculator
Container Description:
Upload To: my-repo
Variable Name
Type
Description
ounces
Double
brand
String
Showing 1 to 2 of 2 variables
Previous
1
Next
Previous
Containerize
Exit

7. 下一頁顯示進度。


1. Name
2. Input Variables
3. Select AWS ECR Repository
4. Review
5. Monitor Progress
The containerization process typically completes within 30 minutes.
Creating Image...
Exit

8. 如果您不小心關閉瀏覽器，您可以從 主控台的 AWS IoT Analytics 筆記本區段監控容器化程序的狀態。
9. 程序完成後，容器化映像會存放在 Amazon ECR 上以供使用。



使用自訂容器進行分析

本節包含如何使用 Jupyter 筆記本建置 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 主控台中，在顯示 HTML 的電腦上提供潛在的攻擊向量。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 服務執行它。若要這樣做，您可以設定 Docker 映像並將其上傳至 Amazon ECR，然後設定資料集以執行容器動作。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
- 在本機電腦上設定的 Docker 帳戶
- 具有 Amazon ECR 或 AWS IoT Analytics 存取權 AWS 的帳戶

步驟 1：設定 Docker 影像

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` – 從 剖析 JSON AWS IoT Analytics、執行 Octave 指令碼，並將成品上傳至 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 – 簡單的 Octave 指令碼，可根據輸入或輸出檔案和指定的順序來計算時刻。

```
#!/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。建立新的目錄，並將所有檔案放入其中，然後cd放入該目錄。
2. 執行下列命令。

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

3. 您應該會在 Docker 儲存庫中看到新的映像。執行下列命令來驗證它。

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

步驟 2：將 Docker 映像上傳至 Amazon ECR 儲存庫

1. 在 Amazon ECR 中建立儲存庫。

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

2. 取得 Docker 環境的登入。

```
aws ecr get-login
```

3. 複製輸出並執行它。輸出看起來應該如下。

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

4. 標記您使用 Amazon ECR 儲存庫標籤建立的映像。

```
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. 下載以下內容以存檔 `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. 將檔案上傳至您剛建立的 Amazon S3 `input.txt`儲存貯體。您現在應該有一個名為 的儲存貯`octave-sample-data-`*`your-aws-account-id`*體，其中包含 `input.txt` 檔案。

步驟 4：建立容器執行角色

1. 將以下內容複製到名為 的檔案`role1.json`。將 *`your-aws-account-id`* 取代為 AWS 您的帳戶 ID，並將 *`aws-region`* 取代為您 AWS 資源 AWS 的區域。

> **Note**
>
> 此範例包含全域條件內容金鑰，以防止混淆代理人安全問題。如需詳細資訊，請參閱the section called “預防跨服務混淆代理人”。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "iotanalytics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset"
                }
            }
        }
    ]
}
```

2. 使用您下載的檔案建立角色 AWS IoT Analytics，以授予對 SageMaker AI 和 `role1.json`的存取許可。

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

3. 將以下內容下載至名為 的檔案，`policy1.json`並以*your-account-id*您的帳戶 ID 取代（請參閱 下的第二個 ARN`Statement:Resource`)。

   JSON

```
{
    "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-123456789012/*"
            ]
        },
        {
            "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. 使用您剛下載policy.json的檔案建立 IAM 政策。

```
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. 將以下內容下載至名為 的 file*region*，cli-input.json並以適當的值取代 *your-account-id*和 的所有執行個體。

```
{
    "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cli-input.json您剛下載和編輯的檔案建立資料集。

```
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：取得資料集內容

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為 SUCCEEDED。

步驟 8：在 Octave上列印輸出

1. 執行下列命令，使用 Octave shell 從容器列印輸出。

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

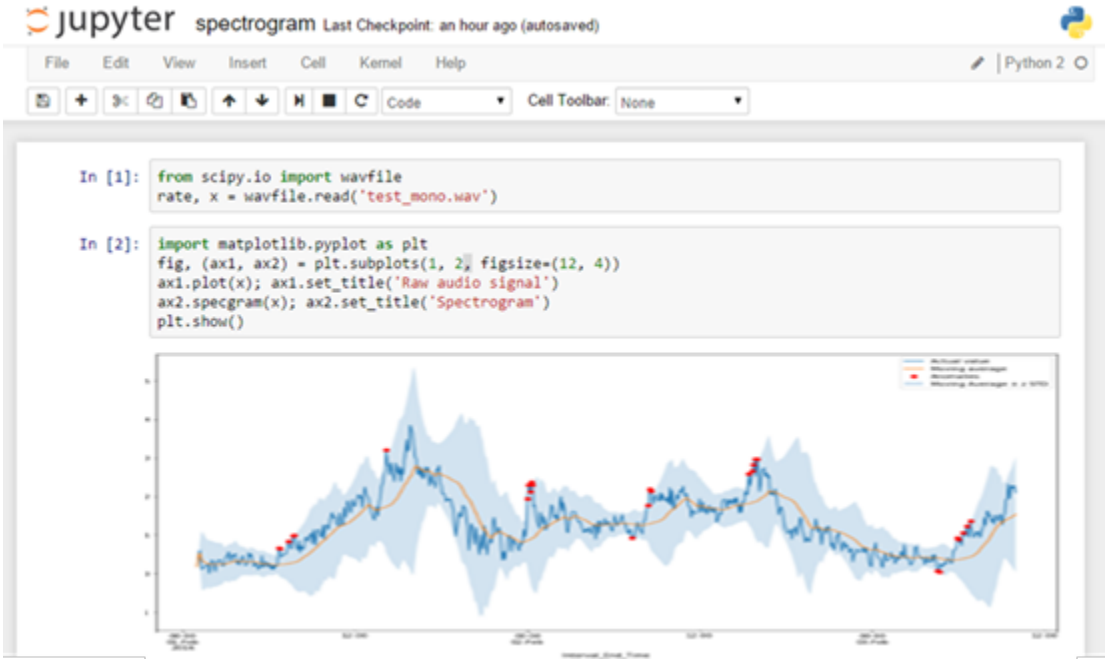
視覺化 AWS IoT Analytics 資料

若要視覺化 AWS IoT Analytics 您的資料，您可以使用 AWS IoT Analytics 主控台或 Quick Suite。

主題

使用主控台視覺化 AWS IoT Analytics 資料

AWS IoT Analytics 可以在AWS IoT Analytics 主控台的容器資料集內容頁面上嵌入容器資料集的 HTML 輸出 （位於 檔案 `output.html`)。例如，如果您定義執行 Jupyter 筆記本的容器資料集，並在 Jupyter 筆記本中建立視覺化，您的資料集可能如下所示。



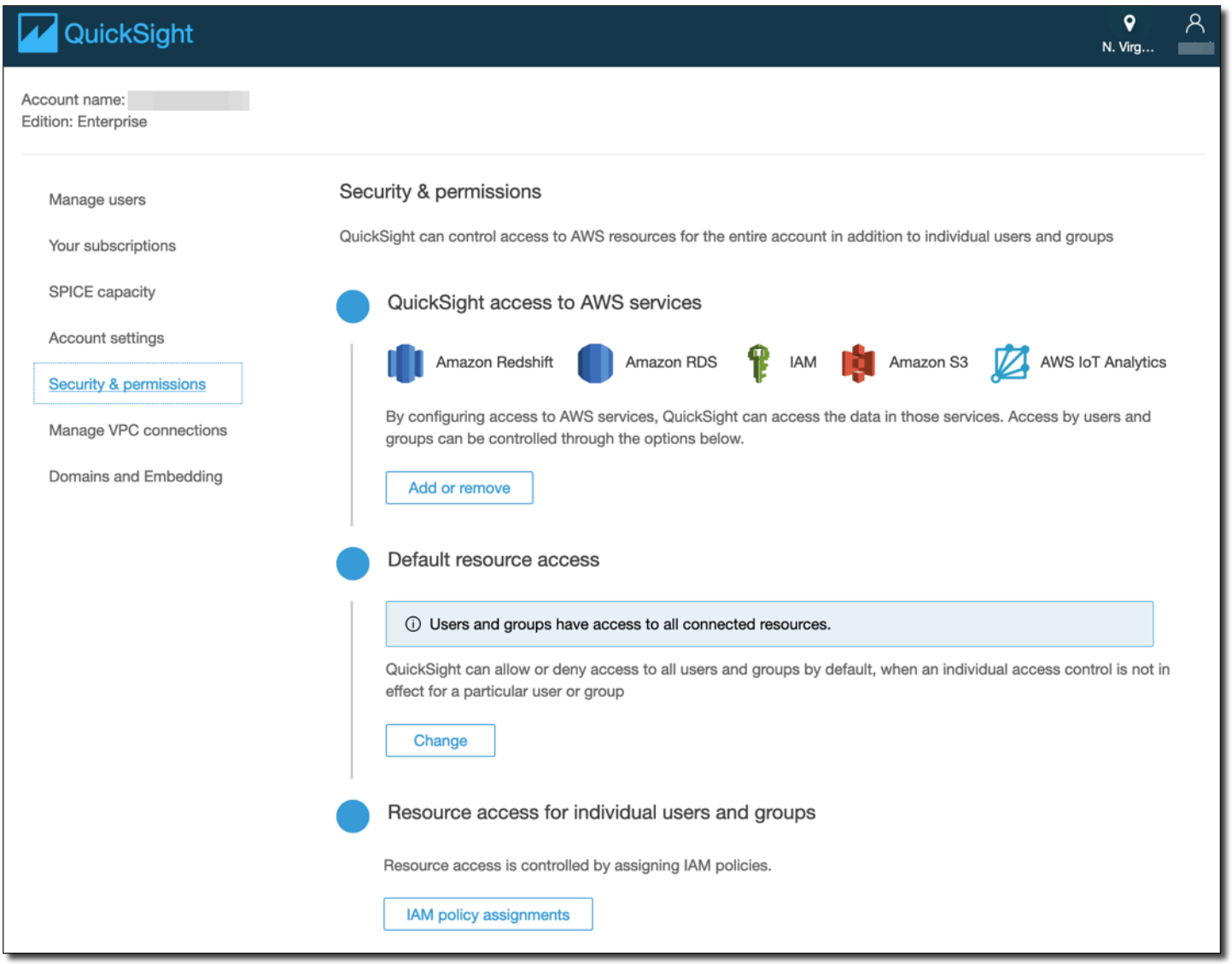
然後，在建立容器資料集內容之後，您可以在主控台的資料集內容頁面上檢視此視覺化效果。

如需建立執行 Jupyter 筆記本之容器資料集的相關資訊，請參閱自動化工作流程。

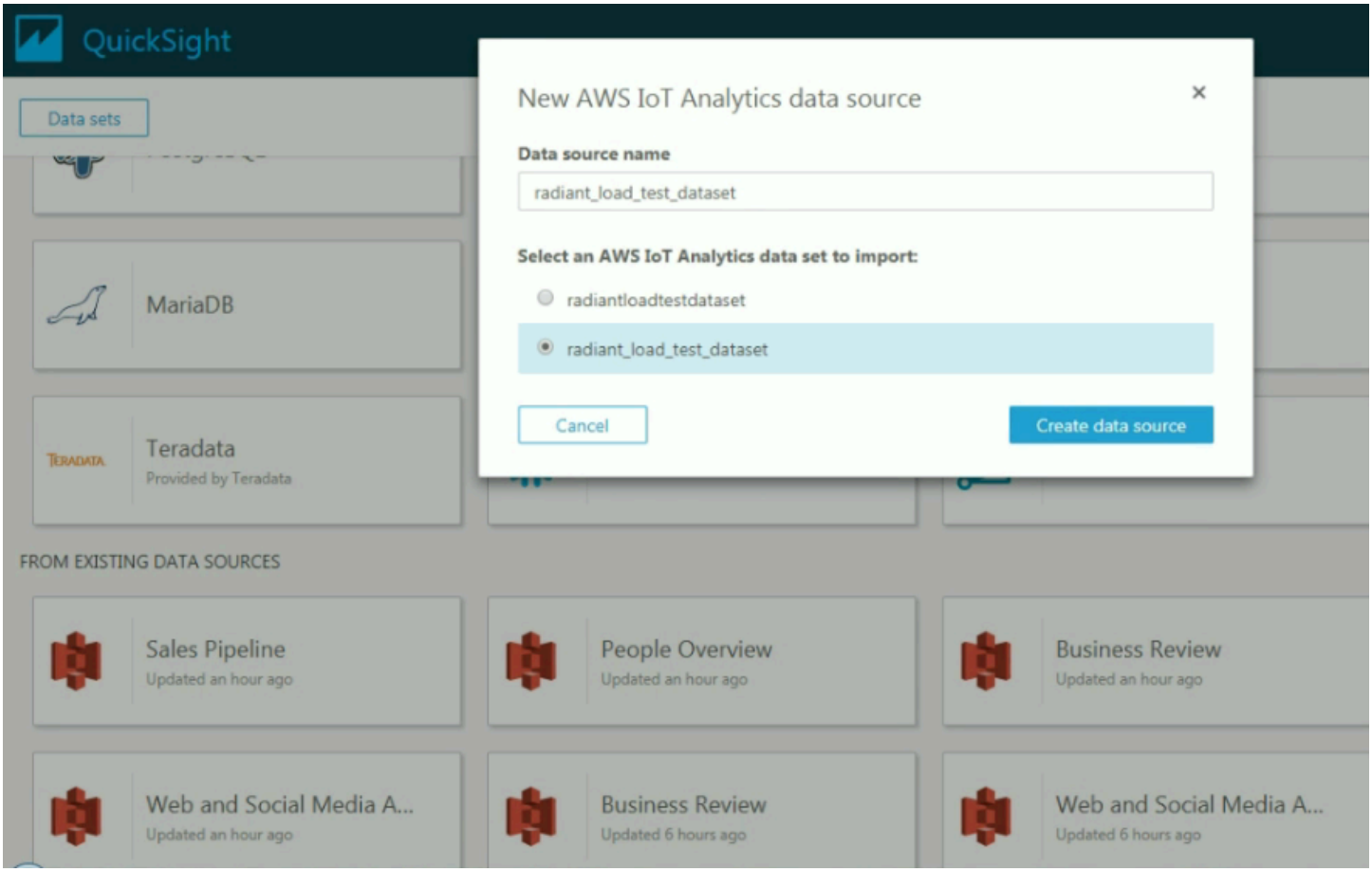
使用 Quick Suite 視覺化 AWS IoT Analytics 資料

AWS IoT Analytics 提供與 Quick Suite 的直接整合。Quick Suite 是一種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，以及快速從資料中取得商業洞見。Quick Suite 可讓組織擴展到數十萬使用者，並使用強大的記憶體內引擎 (SPICE) 提供回應效能。您可以在 Quick Suite 主控台中選取 AWS IoT Analytics 資料集，並開始建立儀表板和視覺化效果。這些區域提供 Quick Suite。

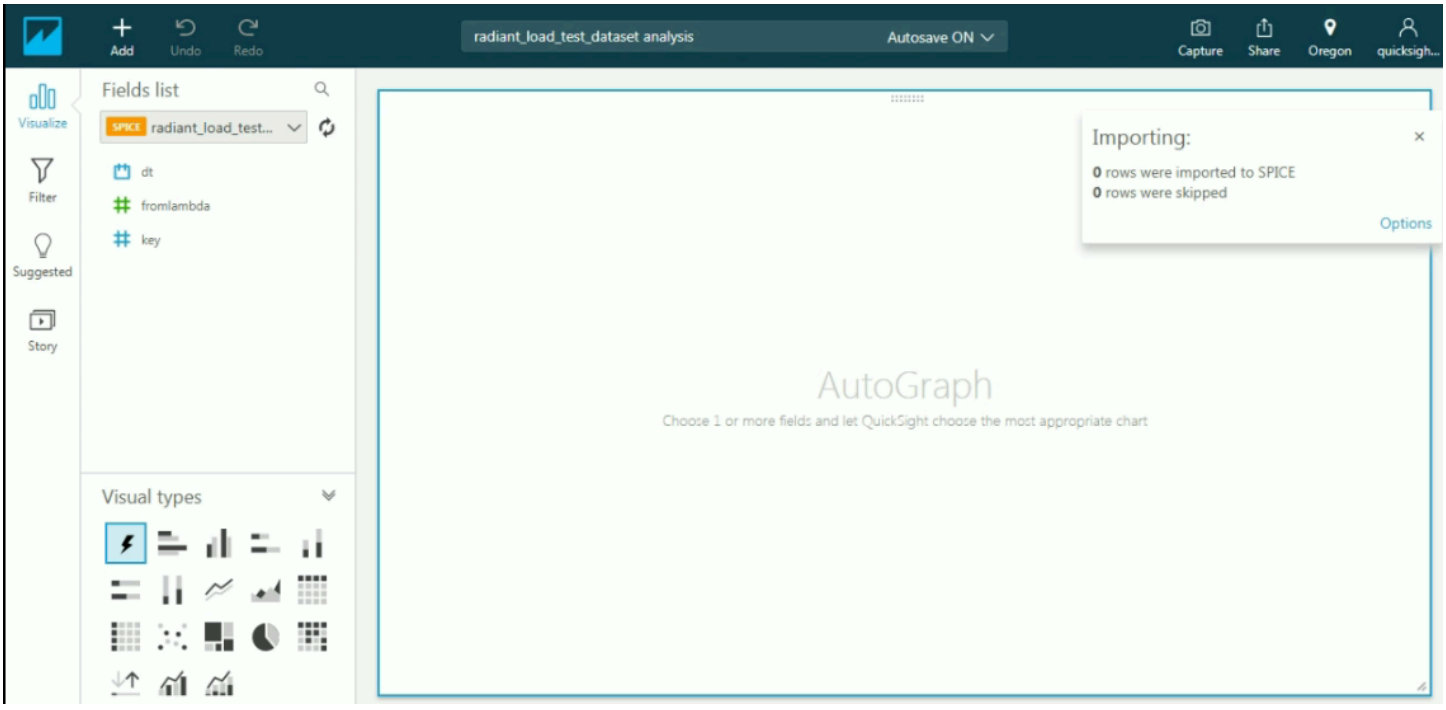
若要開始使用 Quick Suite 視覺化效果，您必須建立 Quick Suite 帳戶。設定帳戶時，請務必授予 Quick Suite AWS IoT Analytics 存取資料的權限。如果您已有 帳戶，請選擇管理員、管理 QuickSight、安全性和許可，讓 Quick Suite 存取 AWS IoT Analytics 您的資料。在 QuickSight 存取 AWS 服務下，選擇新增或移除，然後選取旁邊的核取方塊AWS IoT Analytics，然後選擇更新。



設定您的帳戶後，從管理員 Quick Suite 主控台頁面選擇新分析和新資料集，然後選擇 AWS IoT Analytics 作為來源。輸入資料來源的名稱，選擇要匯入的資料集，然後選擇建立資料來源。



建立資料來源之後，您可以在 Quick Suite 中建立視覺化效果。



如需 Quick Suite 儀表板和資料集的相關資訊，請參閱 Quick Suite 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，並說明如何建立標籤。

主題

標籤基本概念

標籤可讓您以不同的方式分類 AWS IoT Analytics 資源，例如依用途、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。當您將標籤套用至頻道、資料集、資料存放區或管道時，會以逗號分隔值 (CSV) 檔案的形式 AWS 產生成本分配報告，其中包含標籤彙總的用量和成本。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。如需使用標籤進行成本分配的詳細資訊，請參閱AWS Billing 《 使用者指南》中的使用成本分配標籤。

為了方便使用，請使用 AWS 帳單與成本管理 主控台中的標籤編輯器，提供集中、統一的方式來建立和管理標籤。如需詳細資訊，請參閱《 入門 AWS 管理主控台》中的使用標籤編輯器。

您也可以使用 AWS CLI 和 AWS IoT Analytics API 處理標籤。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

您可以為支援標記的現有資源新增、修改或刪除標籤。使用下列命令：

- TagResource
- ListTagsForResource
- UntagResource

您可以編輯標籤索引鍵和值，也可以隨時從資源中移除標籤。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。如果您新增的標籤具有與該資源上現有標籤相同的索引鍵，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)：

- `iotanalytics:ResourceTag/<tag-key>: <tag-value>` 在具有特定標籤的資源上使用 yo 允許或拒絕使用者動作。
- 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 政策中的條件內容索引鍵/值僅適用於資源識別符為必要參數的那些 AWS IoT Analytics 動作。例如，根據條件內容金鑰/值不允許/拒絕使用 DescribeLoggingOptions，因為在此請求中未參考任何可標記資源 (頻道、資料集、資料存放區或管道)。

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 政策參考區段具有 IAM 中 JSON 政策之元素、變數和評估邏輯的詳細語法、描述和範例。

下列範例政策會套用以兩個為基礎的限制。此政策限制的使用者：

1. 無法將標籤 "env=prod" 提供給資源 （請參閱範例中`"aws:RequestTag/env" : "prod"`的行）。
2. 無法修改或存取具有現有標籤 "env=prod" 的資源 （請參閱範例中`"iotanalytics:ResourceTag/env" : "prod"`的 行）。

JSON

```
{
  "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": "*"
    }
  ]
}
```

您也可以將指定標籤索引鍵括在清單中，為指定標籤索引鍵指定多個標籤值，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
- 標籤鍵與值皆區分大小寫。
- 請勿在標籤名稱或值`aws:` `prefix`中使用 ，因為其已保留供 AWS 使用。您不可編輯或刪除具此字首的標籤名稱或值。具有此字首的標籤不會計入每個來源限制的標籤。
- 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

資料集是使用資料存放區中資料的 SQL 表達式產生。 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 ] ]
```

如需參數的說明，請參閱 Amazon Athena 文件中的參數。

AWS IoT Analytics 和 Amazon Athena 不支援下列項目：

- `WITH` 子句。
- `CREATE TABLE AS SELECT` 陳述式
- `INSERT INTO` 陳述式
- 已準備的陳述式，您無法使用 `EXECUTE` 執行`USING`。
- `CREATE TABLE LIKE`
- `DESCRIBE INPUT` 和 `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

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

中支援的 SQL 功能 AWS IoT Analytics

資料集是透過在資料存放區中的資料上使用 SQL 表達式來產生。您在 中執行的查詢 AWS IoT Analytics 是以 Presto 0.217 為基礎。

支援的資料類型

AWS IoT Analytics 和 Amazon Athena 支援這些資料類型。

- primitive_type
  - TINYINT
  - SMALLINT
  - INT
  - BIGINT
  - BOOLEAN
  - DOUBLE
  - FLOAT
  - STRING
  - TIMESTAMP
  - DECIMAL(precision, scale)
  - DATE
  - CHAR（指定長度的固定長度字元資料）
  - VARCHAR（指定長度的可變長度字元資料）
- 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 不支援某些資料類型。

支援的函數

Amazon Athena 和 AWS IoT Analytics SQL 功能是以 Presto 0.217 為基礎。如需有關相關函數、運算子和表達式的資訊，請參閱 Presto 文件中的函數和運算子和下列特定章節。

- 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 和 Amazon Athena 不支援使用者定義的函數 (UDFs或 UDAFs) 或預存程序。

對 中 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疑難排解 中 SQL 查詢的問題 AWS IoT Analytics。

- 若要逸出單引號，請在它前面加上另一個單引號。請勿將此與雙引號混淆。

  Example範例

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

- 若要逸出底線，請使用底線括住以底線開頭的資料存放欄名稱。

  Example範例

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

- 若要以數字逸出名稱，請括住包含雙引號中數字的資料存放區名稱。

  Example範例

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

- 若要逸出預留關鍵字，請以雙引號括住預留關鍵字。如需詳細資訊，請參閱 SQL SELECT 陳述式中的預留關鍵字清單。

中的安全性 AWS IoT Analytics

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

安全性是 AWS 與您之間的共同責任。 共同責任模型將此描述為雲端安全性和雲端安全性：

- 雲端的安全性 - AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 還提供您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。若要了解適用的合規計劃 AWS IoT Analytics，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 - 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的敏感度、您組織的需求和適用的法律及法規。

本文件將協助您了解如何在使用 時套用共同責任模型 AWS IoT Analytics。下列主題說明如何設定 AWS IoT Analytics 以符合您的安全與合規目標。您也將了解如何使用其他 AWS 服務，協助您監控和保護 AWS IoT Analytics 資源。

AWS Identity and Access Management 在 中 AWS IoT Analytics

AWS Identity and Access Management (IAM) 是一種 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證 （登入） 和授權 （具有許可） 來使用 AWS IoT Analytics 資源。IAM 是一項服務 AWS ，您可以免費使用。

目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 - 如果您無法存取功能，請向系統管理員請求許可權 (請參閱 對 AWS IoT Analytics 身分和存取進行故障診斷)
- 服務管理員 - 判斷使用者存取權限，並提交許可權請求 (請參閱 AWS IoT Analytics 如何使用 IAM)
- IAM 管理員 - 撰寫政策以管理存取權 (請參閱 AWS IoT Analytics 身分型政策範例)

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

對於程式設計存取， AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS 第 4 版簽署程序。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務 和 資源的完整存取權。強烈建議您不要以根使用者處理日常任務。有關需要根使用者憑證的任務，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 使用者是一種身分具備單人或應用程式的特定許可權。我們建議使用臨時憑證，而不是具有長期使用權憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的要求人類使用者使用聯合身分提供者，以 AWS 使用臨時憑證存取 。

IAM 群組會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色的身分具有特定許可權，其可以提供臨時憑證。您可以透過從使用者切換到 IAM 角色 （主控台） 或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權、跨服務存取，以及在 Amazon EC2 上執行應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS ， 會評估這些政策。大多數政策會以 JSON 文件 AWS 形式存放在 中。如需有關 JSON 政策文件的詳細資訊，請參閱《IAM 使用者指南》中的 JSON 政策概觀。

使用政策時，管理員會定義哪些主體可以對哪些資源執行動作，以及在哪些條件下執行動作，藉此指定誰可以存取哪些內容。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策，並將其新增至角色，然後使用者就可以擔任該角色。IAM 政策會定義該動作的許可條件，但與使用何種方法進行操作無關。

身分型政策

身分型政策是可以連接身分 (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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可以是內嵌政策 (直接嵌入單一身分) 或受管政策 (連接多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的在受管政策與內嵌政策之間選擇。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 – 設定身分型政策可授予 IAM 實體的最大許可權。如需詳細資訊，請參閱《 IAM 使用者指南》中的 IAM 實體許可界限。
- 服務控制政策 (SCP) – 指定 AWS Organizations中的組織或組織單位的最大許可權。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) – 定義組織中資源可用的最大許可權。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的資源控制政策 (RCP)。
- 工作階段政策 – 這是一種在為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參《IAM 使用者指南》中的工作階段政策。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 決定是否在涉及多個政策類型時允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS IoT Analytics 如何使用 IAM

在您使用 IAM 管理對 的存取之前 AWS IoT Analytics，您應該了解可使用哪些 IAM 功能 AWS IoT Analytics。若要全面了解 AWS IoT Analytics 和其他 AWS 服務如何與 IAM 搭配使用，請參閱《AWS IAM 使用者指南》中的與 IAM 搭配使用的 服務。

本頁主題：

- AWS IoT Analytics 身分型政策

- AWS IoT Analytics 資源型政策
- 以 AWS IoT Analytics 標籤為基礎的授權
- AWS IoT Analytics IAM 角色

AWS IoT Analytics 身分型政策

透過 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。政策動作通常具有與相關聯 AWS API 操作相同的名稱。動作用於政策中，以授予執行相關聯操作的許可。

中的政策動作在動作之前 AWS IoT Analytics 使用以下字首：`iotanalytics:`例如，若要授予某人使用 API 操作建立 AWS IoT Analytics 頻道的 AWS IoT Analytics `CreateChannel`許可，請在其政策中包含 `iotanalytics:BatchPuMessage`動作。政策陳述式必須包含 `Action`或 `NotAction`元素。AWS IoT Analytics 會定義自己的一組動作，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》中的 定義的動作 AWS IoT Analytics。

Resources

`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。

AWS IoT Analytics 資料集資源具有下列 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/*"
```

有些 AWS IoT Analytics 動作無法對特定資源執行，例如用於建立資源的動作。在這些情況下，您必須使用萬用字元 (*)。

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

有些 AWS IoT Analytics API 動作涉及多個資源。例如，CreatePipeline參考 做為頻道和資料集，因此使用者必須具有使用頻道和資料集的許可。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 ARNs的清單，請參閱《IAM 使用者指南》中的 定義的資源 AWS IoT Analytics。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 特定條件金鑰，但支援使用一些全域條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

範例

若要檢視 AWS IoT Analytics 身分型政策的範例，請參閱 AWS IoT Analytics 身分型政策範例。

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》中的針對 使用資源型政策 AWS Lambda。

以 AWS IoT Analytics 標籤為基礎的授權

您可以將標籤連接至 AWS IoT Analytics 資源，或在請求中將標籤傳遞至其中 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 條件索引鍵，在政策的條件元素中提供標籤資訊。 `aws:TagKeys`如需標記 AWS IoT Analytics 資源的詳細資訊，請參閱標記資源 AWS IoT Analytics。

若要檢視以身分為基礎的政策範例，以根據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。

搭配 使用臨時登入資料 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。您取得暫時安全登入資料的方式是透過呼叫 AWS Security Token Service (AWS STS) API 操作 (例如，AssumeRole 或 GetFederationToken)。

AWS IoT Analytics 不支援使用臨時登入資料。

服務連結角色

服務定義角色可讓 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問題，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 中 AWS，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 提供工具，協助您保護所有 服務的資料，讓服務主體能夠存取您帳戶中的資源。

我們建議在資源政策中使用 `aws:SourceArn`和 `aws:SourceAccount`全域條件內容索引鍵。這會限制將另一個服務 AWS IoT Analytics 提供給資源的許可。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

- Amazon S3 儲存貯體的預防
- 使用 Amazon CloudWatch Logs 進行預防
- 客戶受管 AWS IoT Analytics 資源的混淆代理人預防

Amazon S3 儲存貯體的預防

如果您將客戶受管的 Amazon S3 儲存用於 AWS IoT Analytics 資料存放區，則存放資料的 Amazon S3 儲存貯體可能會面臨混淆代理人問題。

例如，Nikki Wolf 使用客戶擁有的 Amazon S3 儲存貯體，稱為 *DOC-EXAMPLE-BUCKET*。儲存貯體會存放在 *us-east-1* 區域中建立之 AWS IoT Analytics 資料存放區的資訊。她指定一個政策，讓 AWS

IoT Analytics 服務主體能夠代表她查詢 *DOC-EXAMPLE-BUCKET*。Nikki 的同事 Li Juan 從自己的帳戶查詢 *DOC-EXAMPLE-BUCKET*，並使用結果建立資料集。因此， AWS IoT Analytics 服務主體代表 Li 查詢 Nikki 的 Amazon S3 儲存貯體，即使 Li 從她的帳戶執行查詢。

為了避免這種情況，Nikki 可以在 *DOC-EXAMPLE-BUCKET* 的政策中指定 `aws:SourceAccount` 條件或 `aws:SourceArn`條件。

指定**`aws:SourceAccount`**條件 - 下列儲存貯體政策範例指定只有來自 Nikki 帳戶 (*123456789012*) AWS IoT Analytics 的資源可以存取 *DOC-EXAMPLE-BUCKET*。

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
```

```
    ]
}
```

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

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastore"
                    ]
                }
            }
        }
```

```
    ]
}
```

使用 Amazon CloudWatch Logs 進行預防

使用 Amazon CloudWatch Logs 監控時，您可以避免混淆代理人問題。下列資源政策顯示如何預防與的混淆代理人問題：

- 全域條件內容索引鍵、 `aws:SourceArn`
- `aws:SourceAccount` 具有您 AWS 帳戶 ID 的
- 中與`sts:AssumeRole`請求相關聯的客戶資源 AWS IoT Analytics

將 *123456789012* 取代為 AWS 您的帳戶 ID，並將 *us-east-1* 取代為您 AWS IoT Analytics 帳戶的區域，如下列範例所示。

JSON

```
{
    "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 CloudWatch Logs 的詳細資訊，請參閱 the section called “日誌記錄和監控”。

客戶受管 AWS IoT Analytics 資源的混淆代理人預防

如果您授予對 AWS IoT Analytics 資源執行動作的 AWS IoT Analytics 許可，資源可能會面臨混淆代理人問題。若要避免混淆代理人問題，您可以使用下列範例資源政策來限制授予 AWS IoT Analytics 的許可。

主題

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

防止 AWS IoT Analytics 頻道和資料存放區

您可以使用 IAM 角色來控制 AWS IoT Analytics 可代表您存取 AWS 的資源。為了避免將您的角色暴露在混淆代理人問題中，您可以在 `aws:SourceAccount`元素中指定 AWS 帳戶，並在您連接到角色的信任政策的 `aws:SourceArn`元素中指定 AWS IoT Analytics 資源的 ARN。

在下列範例中，將 *123456789012* 取代為 AWS 您的帳戶 ID，並將 *arn#aws#iotanalytics#aws-region#123456789012#channel/DOC-EXAMPLE-CHANNEL* 取代為 AWS IoT Analytics 頻道或資料存放區的 ARN。

JSON

```
{
    "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:us-
east-1:123456789012:channel/your-channel"
                }
            }
        }
    ]
}
```

若要進一步了解 頻道和資料存放區的客戶受管 S3 儲存選項，請參閱 AWS IoT Analytics API 參考CustomerManagedDatastoreS3Storage中的 CustomerManagedChannelS3Storage和 。

AWS IoT Analytics 資料集內容交付規則的跨服務混淆代理人預防

AWS IoT Analytics 擔任將資料集查詢結果交付至 Amazon S3 或 的 IAM 角色 AWS IoT Events 可能會面臨混淆代理人問題。為避免混淆代理人問題，請在 aws:SourceAccount元素中指定 AWS 帳戶，並在您連接至角色的信任政策的 aws:SourceArn元素中指定 AWS IoT Analytics 資源的 ARN。

JSON

```
{
    "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:us-
east-1:123456789012:dataset/your-dataset"
                }
            }
        }
    ]
```

```
}
```

如需設定資料集內容交付規則的詳細資訊，請參閱《 AWS IoT Analytics API 參考`contentDeliveryRules`》中的 。

AWS IoT Analytics 身分型政策範例

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 管理主控台 AWS CLI或 AWS API 執行任務。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範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的在 JSON 標籤上建立政策

本頁主題：

- 政策最佳實務
- 使用 AWS IoT Analytics 主控台
- 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。他們會判斷您帳戶中的某個人員是否可以建立、存取或刪除 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 受管政策為您的員工提供所需的許可。這些政策已在您的 帳戶中提供，並由 維護和更新 AWS。如需詳細資訊，請參閱《IAM 使用者指南》中的使用許可搭配 AWS 受管政策。
- 授予最低權限 - 當您建立自訂政策時，僅授予執行任務所需的許可。從一組最低許可開始，並視需要授予其他許可。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。
- 為敏感操作啟用 MFA - 為了提高安全性， 要求使用者使用多重驗證 (MFA) 來存取敏感資源或 API 操作。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)。

- 使用政策條件來提高安全性 - 在實際可行的範圍內，定義以身分為基礎的政策允許存取資源的條件。例如，您可以撰寫條件來指定請求必須來自的允許 IP 地址範圍。您也可以撰寫條件來僅允許指定日期或時間範圍內的請求，或要求使用 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 AWS IoT Analytics 主控台

若要存取 AWS IoT Analytics 主控台，您必須擁有一組最低的許可。這些許可必須允許您列出和檢視中 AWS IoT Analytics 資源的詳細資訊 AWS 帳戶。如果您建立的身分型政策比最低必要許可更嚴格。主控台對於具有該政策的實體（使用者或角色）將無法如預期運作。

為了確保這些實體仍然可以使用 AWS IoT Analytics 主控台，請將下列 AWS 受管政策連接至實體。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。

JSON

```
{
     "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:aws:iotanalytics:us-
east-1:123456789012:channel/your-channel-name",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-datasetName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastoreName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:pipeline/your-pipelineName"
        }
    ]
 }
```

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。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。此政策包含在主控台或使用 或 AWS CLI API 以程式設計方式完成此動作的 AWS 許可。

JSON

```
{
    "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/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，您想要將 AWS 帳戶 存取權授予您其中一個 AWS IoT Analytics 頻道 中的使用者`exampleChannel`。您也想要允許使用 新增、更新和刪除頻道。

政策會將`iotanalytics:ListChannels, iotanalytics:DescribeChannel, iotanalytics:CreateChannel, iotanalytics:DeleteChannel, and iotanalytics:UpdateChannel`許可授予使用者。如需將許可授予使用者並使用 主控台測試使用者之 Amazon S3 服務的範例逐步解說，請參閱範例逐步解說：使用使用者政策來控制對儲存貯體的存取。

JSON

```
{
   "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

您可以在身分型政策中使用條件，根據標籤控制對 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，只有在channel標籤Owner具有該使用者名稱的值時，才會授予許可。此政策也會授予在主控台完成此動作的必要許可。

JSON

```
{
    "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`，`channel`則必須標記 `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。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

對 AWS IoT Analytics 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 時可能遇到的常見問題 AWS IoT Analytics。

主題

- 我無權在 中執行動作 AWS IoT Analytics
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 AWS 帳戶 以外的人員存取我的 AWS IoT Analytics 資源

我無權在 中執行動作 AWS IoT Analytics

如果 AWS 管理主控台 告知您無權執行 動作，您必須聯絡您的管理員尋求協助。您的管理員是為您提供使用者名稱和密碼的人員。

當mateojackson使用者嘗試使用主控台檢視有關 的詳細資訊，channel但沒有iotanalytics:ListChannels許可時，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政策，以允許他使用 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 可讓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 沒有將角色傳遞給服務的許可。

```
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 管理員。您的管理員提供您的簽署憑證。

我想要允許 AWS 帳戶 以外的人員存取我的 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。對於支援資源型政策或存取控制清單 (ACL) 的服務，您可以使用這些政策來授予人員存取您的資源。

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 AWS IoT Analytics 如何使用 IAM。

- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需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。

在 中記錄和監控 AWS IoT Analytics

AWS 提供可用來監控的工具 AWS IoT Analytics。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，在發生錯誤時監看 AWS IoT 和報告 ：

- Amazon CloudWatch Logs - 從 AWS CloudTrail 或其他來源監控、存放和存取您的日誌檔案。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的什麼是 AWS CloudTrail監控日誌檔案。
- AWS CloudTrail 日誌監控 - 在帳戶之間共用日誌檔案、透過將日誌檔案傳送到 CloudTrail CloudWatch Logs 來即時監控 CloudTrail 日誌檔案、在 Java 中寫入日誌處理應用程式，以及驗證您的日誌檔案在 CloudTrail 交付後並未變更。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的使用 CloudTrail 日誌檔案。

手動監控工具

監控的另一個重要部分 AWS IoT 包括手動監控 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讓您at-a-glance檢視 AWS 環境的狀態。我們建議您也檢查 上的日誌檔案 AWS IoT Analytics。

- AWS IoT Analytics 主控台會顯示：
  - 頻道
  - 管道
  - 資料存放區

  - 資料集
  - Notebooks
  - 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，您可以使用 CloudWatch 執行下列動作：
  - 建立自定儀表板來監控您注重的服務
  - 用於疑難排解問題以及探索驅勢的圖形指標資料。
  - 搜尋和瀏覽您的所有 AWS 資源指標
  - 建立與編輯要通知發生問題的警示

使用 Amazon CloudWatch Logs 進行監控

AWS IoT Analytics 支援使用 Amazon CloudWatch 記錄。您可以使用 `PutLoggingOptions` API 操作啟用和設定 的 Amazon CloudWatch AWS IoT Analytics 記錄。本節說明如何使用 `PutLoggingOptions`搭配 AWS Identity and Access Management (IAM) 來設定和啟用 Amazon CloudWatch 記錄 AWS IoT Analytics。

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

> **Note**
>
> 啟用 AWS IoT Analytics 記錄之前，請確定您了解 CloudWatch Logs 存取許可。具有 CloudWatch Logs 存取權限的使用者，皆可查看偵錯資訊。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 CloudWatch 的記錄

1. 使用 AWS IAM 主控台或下列 AWS IAM CLI 命令 CreateRole，建立具有信任關係政策（信任政策）的新 IAM 角色。信任政策會授予實體擔任該角色的許可，例如 Amazon CloudWatch。

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

exampleTrustPolicy.json 檔案包含下列內容。

> **Note**
>
> 此範例包含全域條件內容金鑰，以防止混淆代理人安全問題。將 *123456789012* 取代為您的 AWS 帳戶 ID，並將 *aws-region* 取代為您的 AWS 資源 AWS 區域。如需詳細資訊，請參閱the section called “預防跨服務混淆代理人”。

JSON

```
{
    "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:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

稍後當您呼叫 命令時， AWS IoT Analytics PutLoggingOptions您會使用此角色的 ARN。

2. 使用 AWS IAM PutRolePolicy 將許可政策 (role policy) 連接至您在步驟 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。

JSON

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

3. 若要授予將記錄事件放入 Amazon CloudWatch 的 AWS IoT Analytics 許可，請使用 Amazon CloudWatch 命令 PutResourcePolicy。

> **Note**
>
> 為了協助防止混淆代理人安全問題，建議您在資源政策aws:SourceArn中指定 。這會將存取限制為僅允許來自指定帳戶的請求。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”。

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

exampleResourcePolicy.json 檔案包含下列資源政策。

JSON

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

設定和啟用記錄

使用 PutLoggingOptions命令來設定和啟用 的 Amazon CloudWatch 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 AWS IoT Analytics 記錄選項。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，如果您變更連接到您在 roleArn 欄位中指定之角色的政策（例如，更正無效政策），則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 日誌記錄選項的目前設定。如需詳細資訊，請參閱DescribeLoggingOptions

命名空間、指標和維度

AWS IoT Analytics 將下列指標放入 Amazon CloudWatch 儲存庫：

命名空間
AWS/IoTAnalytics

指標	說明
ActionExecution	執行的動作數目。
ActionExecutionThrottled	已調節的動作數目。
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	同時執行的管道活動數量。

維度	說明
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。

維度	說明
PipelineName	正在監控的管道名稱。

使用 Amazon CloudWatch Events 監控

AWS IoT Analytics 在 AWS Lambda 活動期間發生執行時間錯誤時， 會自動將事件發佈至 Amazon CloudWatch Events。此事件包含詳細的錯誤訊息，以及存放未處理頻道訊息的 Amazon Simple Storage Service (Amazon S3) 物件金鑰。您可以使用 Amazon S3 金鑰來重新處理未處理的頻道訊息。如需詳細資訊，請參閱 重新處理頻道訊息、《 API 參考》中的 StartPipelineReprocessing API，以及《Amazon CloudWatch Events 使用者指南》中的什麼是 Amazon CloudWatch Events。 AWS IoT Analytics

您也可以設定讓 Amazon CloudWatch Events 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，然後調用 `StartReprocessingMessage` API 來處理儲存在 Amazon S3 物件中的頻道訊息。Amazon CloudWatch Events 支援許多類型的目標，例如：

- Amazon Kinesis Streams
- AWS Lambda 函數
- 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。EventBridge

您的 CloudWatch Events 資源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。

針對 AWS Lambda 活動中的執行時間錯誤，傳送至 Amazon CloudWatch Events 的通知使用下列格式。

```
{
    "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 Events 取得延遲資料通知

當您使用來自指定時間範圍的資料建立資料集內容時，某些資料可能無法及時送達進行處理。若要允許延遲，您可以在建立資料集QueryFilter時套用 queryAction(SQL 查詢） 來指定的deltaTime位移。 AWS IoT Analytics 仍會處理在差異時間內抵達的資料，而且您的資料集內容會有時間延遲。延遲資料通知功能可讓 在資料於差異時間後送達時，透過 Amazon CloudWatch Events AWS IoT Analytics 傳送通知。

您可以使用 AWS IoT Analytics 主控台、API、AWS Command Line Interface (AWS CLI) 或 AWS SDK 來指定資料集的延遲資料規則。

在 AWS IoT Analytics API 中， LateDataRuleConfiguration 物件代表資料集的延遲資料規則設定。此物件是與 CreateDataset和 UpdateDataset API 操作相關聯的Dataset物件的一部分。

參數

當您使用 建立資料集的延遲資料規則時 AWS IoT Analytics，您必須指定下列資訊：

ruleConfiguration (**LateDataRuleConfiguration**)

包含延遲資料規則組態資訊的結構。

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 的範例DeltaTime，請參閱使用差異視窗 (CLI) 建立 SQL 資料集。

timeoutInMinutes

時間間隔。您可以使用 timeoutInMinutes讓 AWS IoT Analytics 批次處理自上次執行後產生的延遲資料通知。 AWS IoT Analytics 會一次傳送一批次通知至 CloudWatch Events。

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。

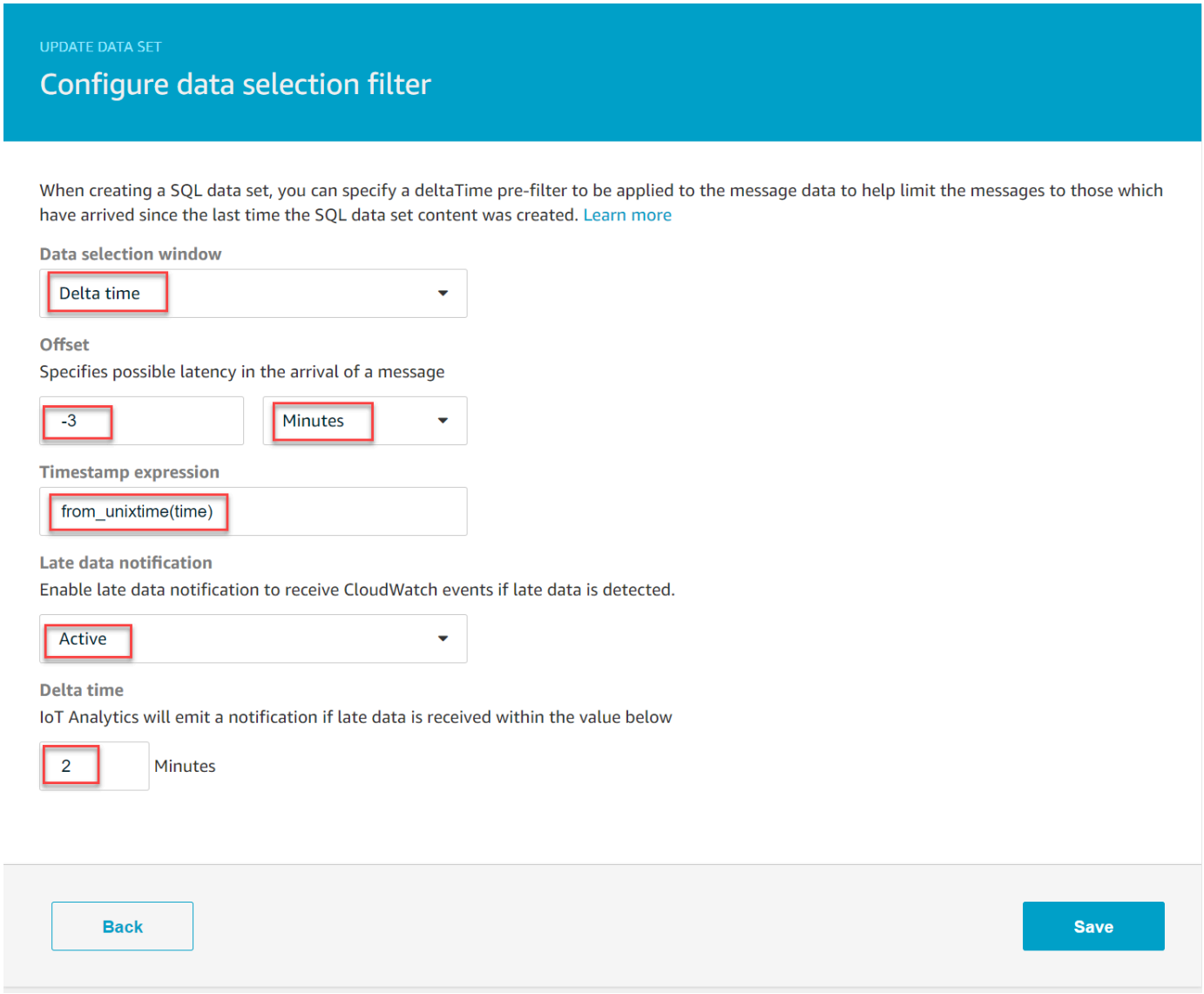
設定延遲資料規則（主控台）

下列程序說明如何在 AWS IoT Analytics 主控台中設定資料集的延遲資料規則。

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 Delta 視窗區段中，選擇編輯。
6. 在設定資料選取篩選條件下，執行下列動作：
   a. 針對資料選擇視窗，選擇 Delta 時間。
   b. 針對位移，輸入時段，然後選擇單位。
   c. 針對時間戳記表達式，輸入表達式。這可以是時間戳記欄位的名稱或可衍生時間的 SQL 表達式，例如 *`from_unixtime(time)`*。

      如需如何撰寫時間戳記表達式的詳細資訊，請參閱 Presto 0.172 文件中的日期和時間函數和運算子。
   d. 針對延遲資料通知，選擇作用中。
   e. 針對 Delta 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` 物件代表資料集的延遲資料規則設定。此物件是與 `CreateDataset`和 相關聯的`Dataset`物件的一部分`UpdateDataset`。您可以使用 APIAWS CLI、 或 AWS SDK 來指定資料集的延遲資料規則。下列為使用 AWS CLI的範例。

若要使用指定的延遲資料規則建立資料集，請執行下列命令。命令假設`dataset.json`檔案位於目前的目錄中。

> **Note**
>
> 您可以使用 UpdateDataset API 來更新現有的資料集。

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

dataset.json 檔案應包含下列項目：

- 將 *demo_dataset* 取代為目標資料集名稱。
- 將 *demo_datastore* 取代為目標資料存放區名稱。
- 將 *from_unixtime(time)* 取代為時間戳記欄位的名稱或可衍生時間的 SQL 表達式。

  如需如何撰寫時間戳記表達式的詳細資訊，請參閱 Presto 0.172 文件中的日期和時間函數和運算子。
- 將*##*取代為介於 1–60 之間的整數。
- 以任何名稱取代 *demo_rule*。

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

訂閱以接收延遲資料通知

您可以在 CloudWatch Events 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 Events 收到通知時，它會叫用規則中定義的指定目標動作。

建立 CloudWatch Events 規則的先決條件

為 建立 CloudWatch Events 規則之前 AWS IoT Analytics，您應該執行下列動作：

- 熟悉 CloudWatch Events 中的事件、規則和目標。
- 建立和設定 CloudWatch Events 規則調用的目標。規則可以叫用許多類型的目標，例如：
  - Amazon Kinesis Streams
  - AWS Lambda 函數
  - Amazon Simple Notification Service (Amazon SNS) 主題
  - Amazon Simple Queue Service (Amazon SQS) 佇列

  您的 CloudWatch Events 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《Amazon CloudWatch Events 使用者指南》中的什麼是 CloudWatch Events？和 Amazon CloudWatch Events 入門。 Amazon CloudWatch

延遲資料通知事件

延遲資料通知的事件使用以下格式。

```
{
```

```
 "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 Events 規則以接收延遲資料通知

下列程序說明如何建立將 AWS IoT Analytics 延遲資料通知傳送至 Amazon SQS 佇列的規則。

建立 CloudWatch Events 規則

1. 登入 Amazon CloudWatch 主控台。
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. 在依服務比對事件的建置事件模式中，執行下列動作：
   a. 針對服務名稱，選擇 IoT Analytics
   b. 針對事件類型，選擇 IoT Analytics 資料集生命週期通知。
   c. 選擇特定資料集名稱（多個），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：
   - 針對佇列*，選擇目標佇列。
8. 選擇設定詳細資訊。
9. 在步驟 2：設定規則詳細資訊頁面上，輸入名稱和描述。
10. 選擇建立規則。

使用 記錄 AWS IoT Analytics API 呼叫 AWS CloudTrail

AWS IoT Analytics 已與 服務整合 AWS CloudTrail，此服務可提供使用者、角色或 AWS 服務在其中採取之動作的記錄 AWS IoT Analytics。CloudTrail 會將 的 API 呼叫子集擷取 AWS IoT Analytics 為事件，包括來自 AWS IoT Analytics 主控台的呼叫，以及來自對 AWS IoT Analytics APIs的程式碼呼叫。如果您建立線索，您可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 的事件 AWS IoT Analytics。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 所收集的資訊來判斷提出的請求 AWS IoT Analytics、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

AWS IoT Analytics 中的資訊 AWS CloudTrail

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 中發生時 AWS IoT Analytics，該活動會與事件歷史記錄中的其他服務 AWS 事件一起記錄在 CloudTrail 事件中。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶中的事件，包括 的事件 AWS IoT Analytics，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立線索時，線索會套用到所有 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案傳送到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中收集的事件資料。如需詳細資訊，請參閱：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

AWS IoT Analytics 支援將下列動作記錄為 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

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

- 是否使用根登入資料或 AWS Identity and Access Management 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

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

了解 AWS 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 日誌檔案包含一或多個日誌專案。事件代表來自任何來源的單一請求，並包含所請求動作、動作的日期和時間、請求參數等相關資訊。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 `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"
}
```

的合規驗證 AWS IoT Analytics

若要了解 是否 AWS 服務 在特定合規計劃範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 AWS 安全文件。

中的彈性 AWS IoT Analytics

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。 AWS 區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您可以設計和操作在可用區域之間自動容錯移轉的應用程式和資料。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

中的基礎設施安全 AWS IoT Analytics

作為受管服務， AWS IoT Analytics 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫， AWS IoT Analytics 透過網路存取 。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。

AWS IoT Analytics 配額

AWS 一般參考 指南提供 AWS IoT Analytics AWS 帳戶的預設配額。除非另有說明，否則每個配額都是每個 AWS 區域。如需詳細資訊，請參閱《 AWS 一般參考 指南》中的AWS IoT Analytics 端點、配額AWS 和服務配額。

若要請求提高服務配額，請在支援中心主控台中提交支援案例。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

閱讀本主題以了解 的 API 操作 AWS IoT Analytics，包括支援 Web 服務通訊協定的範例請求、回應和錯誤。

AWS IoT Analytics 動作

您可以使用 AWS IoT Analytics API 命令來收集、處理、存放和分析 IoT 資料。如需詳細資訊，請參閱 AWS IoT Analytics API 參考 AWS IoT Analytics 中 支援的動作。

AWS CLI 命令參考中的AWS IoT Analytics 各節包含您可以用來管理和操作的 AWS CLI 命令 AWS IoT Analytics。

AWS IoT Analytics 資料

您可以使用 AWS IoT Analytics Data API 命令，透過 AWS IoT Analytics `channel`、`datastore`、`pipeline`和 執行進階活動`dataset`。如需詳細資訊，請參閱 AWS IoT Analytics API 參考中 AWS IoT Analytics 資料支援的資料類型。

故障診斷 AWS IoT Analytics

請參閱下一節，針對錯誤進行故障診斷，並尋找解決 問題的可能解決方案 AWS IoT Analytics。

主題

如何知道我的訊息是否進入 AWS IoT Analytics？

檢查透過規則引擎將資料注入頻道的規則是否已正確設定。

```
aws iot get-topic-rule --rule-name your-rule-name
```

回應應如下所示。

```
{
    "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且規則正確執行，您可能需要新增目標，以暫時將傳入訊息存放在 Amazon S3 儲存貯體中。

為什麼我的管道遺失訊息？ 我該如何修正這個問題？

- 活動收到無效的 JSON 輸入：

  除了 Lambda 活動之外，所有活動都特別需要有效的 JSON 字串做為輸入。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 函數範例 2。

- Lambda 活動叫用的 Lambda 函數沒有足夠的許可：

  確定 Lambda 活動中的每個 Lambda 函數都有從 AWS IoT Analytics 服務叫用的許可。您可以使用下列 AWS CLI 命令來授予許可。

  ```
  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：

  如果有任何 `filter`或 `removeAttribute`活動是正確的，請確定定義。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：

  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。如果您的資料集不包含此項目，請參閱上一個問題和回應。

如何為資料集完成所驅動的事件編寫程式碼？

- 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為 SUCCEEDED。

如何正確設定筆記本執行個體以使用 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. 前往 SageMaker AI 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 AI 可以使用的角色。
4. 前往 IAM 主控台並修改新建立的 SageMaker AI 角色。當您開啟該角色，應該會有一個受管政策。
5. 按一下新增內嵌政策，選擇 IoTAnalytics 做為服務，然後在讀取許可下，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政策許可 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「待定」狀態開始。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，請再次檢查許可。

為什麼我在 Quick Suite 中看不到我的資料集？

Quick Suite 可能需要讀取 AWS IoT Analytics 資料集內容的許可。若要提供許可，請遵循下列步驟。

1. 選擇 Quick Suite 右上角的帳戶名稱，然後選擇管理 QuickSight。
2. 在左導覽窗格中，選擇安全和許可。在 QuickSight 存取 AWS 服務下，驗證是否已授予存取權 AWS IoT Analytics。

    a. 如果 AWS IoT Analytics 沒有存取權，請選擇新增或移除。

    b. 選擇旁邊的方塊，AWS IoT Analytics然後選取更新。這提供 Quick Suite 讀取資料集內容的許可。
3. 再次嘗試視覺化您的資料。

請務必為 AWS IoT Analytics 和 Quick Suite 選擇相同的 AWS 區域。否則，您可能無法存取 AWS 資源。如需支援的區域清單，請參閱 中的AWS IoT Analytics 端點和配額，以及 Quick Suite 端點和配額Amazon Web Services 一般參考。

為什麼我在現有的 Jupyter 筆記本上看不到容器化按鈕？

- 這是因為缺少 AWS IoT Analytics 容器化外掛程式所致。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 主控台建立或手動安裝 SageMaker 筆記本執行個體後沒有看到容器化按鈕，請聯絡 AWS IoT Analytics 技術支援。

為什麼我的容器化外掛程式安裝失敗？

- 外掛程式安裝失敗通常是因為 SageMaker 筆記本執行個體缺少許可。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。
- 如果在安裝外掛程式期間出現以下訊息，您可以安全地忽略日誌中的以下訊息：「每次筆記本（或其他應用程式）載入時，在瀏覽器中初始化此擴充功能。」

為什麼我的容器化外掛程式擲回錯誤？

- 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 使用「容器化」核心執行筆記本後，容器化外掛程式會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。

我可以將哪些變數新增至我的容器做為輸入？

- 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。這可讓您執行具有不同參數的相同容器，這些參數需要在建立資料集時提供。 AWS IoT Analytics 容器化 Jupyter 外掛程式透過自動識別筆記本中的變數並將其作為容器化程序的一部分提供，來簡化此程序。

如何將容器輸出設定為後續分析的輸入？

- 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。

為什麼我的容器資料集會失敗？

- 請確定您將正確的 傳遞`executionRole`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`。
- 如果您看到 `AlgorithmError`是失敗的原因，請嘗試手動偵錯您的容器程式碼。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 外掛程式容器化，請使用與 containerDataset 的 executionRole 相同的角色建立新的 SageMaker 筆記本執行個體，並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。

文件歷史記錄

下表說明 AWS IoT Analytics 使用者指南在 2020 年 11 月 3 日之後的重要變更。如需有關此文件更新的詳細資訊，您可以訂閱 RSS 摘要。

變更	描述	日期
終止支援通知	終止支援通知：在 2025 年 12 月 15 日， AWS 將結束對 的支援 AWS IoT Analytics 。2025 年 12 月 15 日之後，您將無法再存取 AWS IoT Analytics 主控台或 AWS IoT Analytics 資源。如需詳細資訊，請參閱AWS IoT Analytics 終止支援。	2025 年 5 月 20 日
AWS IoT Analytics 不再提供給新客戶	AWS IoT Analytics 不再提供給新客戶。的現有客戶 AWS IoT Analytics 可以繼續正常使用服務。進一步了解	2024 年 8 月 8 日
區域啟動	AWS IoT Analytics 現可於亞太區域 （ 孟買 ） 區域使用。	2021 年 8 月 18 日
使用 查詢 JOIN	此更新可讓您使用 JOIN 來查詢 AWS IoT Analytics 資料集。	2021 年 7 月 27 日
與 整合 AWS IoT SiteWise	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 現在通常支援根據透過管道活動新增的訊息屬性來分割資料。	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 物件中的頻道資料。	2020 年 12 月 15 日
Parquet 結構描述	AWS IoT Analytics 資料存放區現在支援 Parquet 檔案格式。	2020 年 12 月 15 日
使用 CloudWatch Events 進行監控	AWS IoT Analytics 在 AWS Lambda 活動期間發生執行時間錯誤時，會自動將事件發佈至 Amazon CloudWatch Events。	2020 年 12 月 15 日
延遲資料通知	您可以使用此功能，在延遲資料送達時透過 Amazon CloudWatch Events 接收通知。	2020 年 11 月 9 日
區域啟動	AWS IoT Analytics 在中國（北京）推出。	2020 年 11 月 4 日

舊版更新

下表說明 AWS IoT Analytics 使用者指南在 2020 年 11 月 4 日之前的重要變更。

變更	描述	日期
區域啟動	AWS IoT Analytics 在亞太區域（雪梨）區域推出。	2020 年 7 月 16 日
更新	重新整理文件。	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。