Amazon IoT Analytics

用户指南

Amazon IoT Analytics: 用户指南

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅中国的 Amazon Web Services 服务入门。

Table of Contents

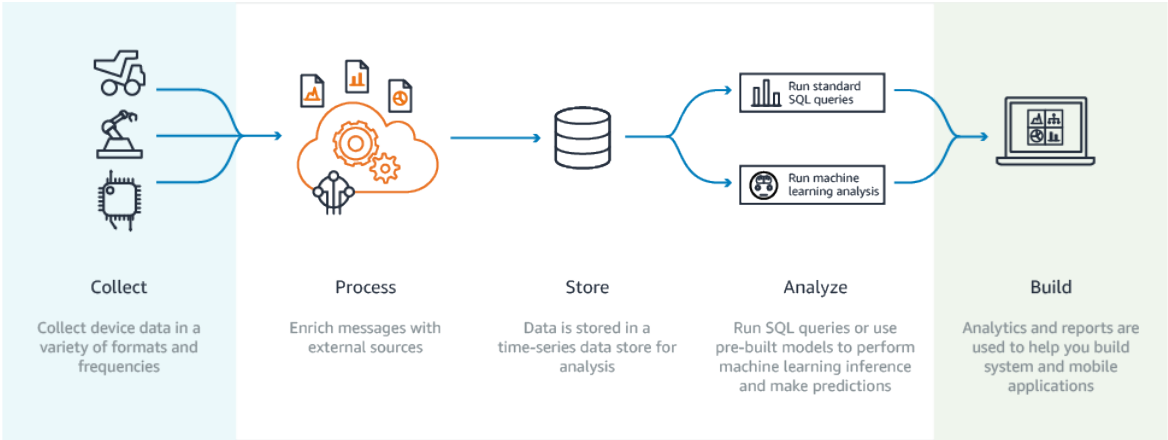
什么是 Amazon IoT Analytics？

Amazon IoT Analytics自动执行分析来自 IoT 设备的数据所需的步骤。Amazon IoT AnalyticsIoT 数据进行筛选、转换和丰富，然后再将其存储在时间序列数据存储中以供分析。您可以将服务设置为只从您的设备中收集所需数据，应用数学转换来处理数据，并使用设备特定的元数据 (例如设备类型和位置) 来丰富数据，然后再存储这些数据。接下来，您可以使用内置的 SQL 查询引擎通过运行查询来分析您的数据，或执行更复杂的分析和机器学习推理。Amazon IoT Analytics与集成以实现高级数据探索Jupyter Notebook.Amazon IoT Analytics还通过与集成以实现数据可视化Amazon QuickSight. 亚马逊 QuickSight 可以在以下内容中使用区域.

传统分析和商业智能工具设计用于处理结构化数据。原始 IoT 数据通常来自记录较少结构化数据（如温度、动作或声音）的设备。因而来自这些设备的数据可能具有大量空白、损坏的消息和错误的读数，必须先清除这些，才能进行分析。另外，IoT 数据通常仅在来自外部源的其他数据的上下文中有意义。Amazon IoT Analytics您可以解决这些问题，收集大量设备数据，处理消息并存储它们。然后，您可以查询数据并对其进行分析。Amazon IoT Analytics包括适用于常见 IoT 使用案例的预构建模型，以便您可以回答一些问题，如哪些设备即将出现故障或哪些客户有放弃其可穿戴设备的风险。

如何使用 Amazon IoT Analytics

下图显示了如何使用。Amazon IoT Analytics.



主要功能

收集

- 与 集成Amazon IoT Core—Amazon IoT Analytics与完全集成Amazon IoT Core因此它可以在流入时从连接设备接收消息。
- 使用批处理 API 从任何源添加数据 —Amazon IoT Analytics可以通过 HTTP 接收来自任何来源的数据。这意味着连接到 Internet 的任何设备或服务可以将数据发送到Amazon IoT Analytics. 有关更多信息，请参阅 。BatchPutMessage中的Amazon IoT AnalyticsAPI 参考.
- 仅收集您想要存储的数据并分析您可以使用Amazon IoT Analytics配置控制台Amazon IoT Analytics通过各种格式和频率的 MQTT 主题过滤器接收来自设备的消息。Amazon IoT Analytics验证数据是否在您定义的特定参数内并创建通道。然后，服务将通道路由到适当的管道来对消息进行处理、转换和丰富。

过程

- 清洁和过滤Amazon IoT Analytics让您定义Amazon Lambda在以下情况下触发的函数Amazon IoT Analytics检测缺少的数据，因此您可以运行代码来估算并填补空白。您还可以定义最大值和最小值筛选条件以及百分位数阈值以删除数据中的异常值。
- 转换Amazon IoT Analytics您可以使用您定义的数学或条件逻辑来转换消息，以便您可以执行常见计算，例如摄氏温度到华氏温度的换算。
- 丰富 —Amazon IoT Analytics可以使用外部数据源 (如天气预报) 来丰富数据，然后将数据路由到Amazon IoT Analytics数据存储。

存储

- 时序列数据存储 —Amazon IoT Analytics将设备数据存储在优化的时间序列数据存储中，以更快地进行检索和分析。您还可以管理访问权限，实施数据保留策略以及将数据导出到外部访问点。
- 存储处理过的和原始数据 —Amazon IoT Analytics存储处理的数据，并且还会自动存储原始提取的数据，以便您稍后对其进行处理。

分析

- 运行临时 SQL 查询 —Amazon IoT Analytics提供了 SQL 查询引擎，因此您可以运行临时查询并快速获得结果。通过使用该服务，您可以使用标准 SQL 查询从数据存储中提取数据以回答一些问题，如互联车队的平均行驶距离，或者在晚上 7 点以后将智能建筑中的多少个门锁上。这些查询可重复使用，即使互连设备、队列大小和分析要求发生了更改也是如此。
- 时间序列分析 —Amazon IoT Analytics支持时间序列分析，因此您可以分析设备随时间推移的性能并了解如何以及在哪里使用它们，持续监控设备数据来预测维护问题，以及监控传感器来预测环境条件并相应地作出反应。
- 托管笔记本用于复杂的分析和机器学习 —Amazon IoT Analytics支持托管在 Jupyter Notebook 中的笔记本，用于统计分析和机器学习。该服务包括一组包含的笔记本模板Amazon-创作的机器学习模型和可视化效果。您可以使用模板开始使用与设备故障分析相关的 IoT 使用案例，预测可能表明客户将放弃产品的低使用量事件，或者按客户使用水平（例如，重量用户、周末用户）或设备健康状况对设备进行分类。在创作笔记本之后，可以将其容器化并按指定计划执行它。有关更多信息，请参阅 。自动执行工作流.
- 预测-您可以通过称为逻辑回归的方法进行统计分类。您还可以使用长短期记忆 (LSTM)，这是一种强大的神经网络技术，用于预测随时间变化的过程的输出或状态。预构建的笔记本模板还支持用于设备细分的 K-means 集群算法，它将您的设备划分为类似设备组。这些模板通常用于剖析设备运行状况和设备状态，如巧克力工厂中的 HVAC 装置的运行状况或风力涡轮机叶片的磨损状态。同样，这些笔记本模板可以按计划包含和执行。

构建并可视化

- 亚马逊 QuickSight 集成 —Amazon IoT Analytics提供亚马逊的连接器 QuickSight 这样你就可以在 QuickSight 控制面板。
- 控制台集成 — 您还可以在中的嵌入式 Jupyter 笔记本中可视化结果或临时分析。Amazon IoT Analytics'控制台。

Amazon IoT Analytics组件和概念

通道

通道收集来自 MQTT 主题的数据，并在将数据发布到管道之前将原始未处理消息归档。您也可以使用BatchPutMessageAPI。未处理的消息存储在一个 Amazon Simple Storage Service (Amazon S3) 储桶中，您或Amazon IoT Analytics管理。

管道

管道使用来自通道的消息，并且您可以在将消息存储在数据存储之前处理消息。处理步骤，称为活动(Pipeline 活动) 对您的消息执行转换，例如，删除、重命名或添加消息属性，根据属性值筛选消息，对消息调用 Lambda 函数以进行高级处理，或者执行数学变换以使设备数据标准化。

数据存储

管道将它们处理过的消息存储在数据存储中。数据存储不是数据库，但它是一个可扩展且可查询的消息存储库。对于来自不同设备或位置的消息，您可以有多个数据存储，或者根据您的管道配置和要求，通过消息属性进行筛选。与未处理的通道消息一样，数据存储的处理消息存储在Amazon S3存储桶你或Amazon IoT Analytics管理。

数据集

您可以通过创建数据集来从数据存储中检索数据。Amazon IoT Analytics您可以创建 SQL 数据集或容器数据集。

在拥有数据集之后，您可以通过使用以下方式进行集成来探索数据并获得见解。Amazon QuickSight. 您还可以通过与集成来执行更高级的分析功能。Jupyter Notebook. Jupyter 笔记本提供强大的数据科学工具，可以执行机器学习和一系列统计分析。有关更多信息，请参阅 。笔记本模板.

您可以将数据集内容发送到Amazon S3存储桶，允许与现有数据湖集成，或者从内部应用程序和可视化工具中进行访问 您也可以将数据集内容作为输入发送到Amazon IoT Events，该服务允许您监控设备或进程故障或操作更改，并在发生此类事件时触发其他操作。

SQL 数据集

SQL 数据集类似于 SQL 数据库的具体化视图。您可以通过应用 SQL 操作来创建 SQL 数据集。SQL 数据集可以通过指定触发器，按定期计划自动生成。

容器数据集

通过容器数据集，您可以自动运行分析工具并生成结果。有关更多信息，请参阅 。自动执行工作流. 它将作为输入的 SQL 数据集、Docker 容器及分析工具和所需库文件、输入和输出变量以及可选的调度触发器组合到一起。输入和输出变量告知可执行映像获取数据和存储结果的位置。触发器可以在 SQL 数据集完成创建其内容时或者按照时间调度表达式来运行分析。容器数据集自动运行，生成并随后保存分析工具的结果。

触发器

您可以通过指定触发器来自动创建数据集。此触发器可以是时间间隔（例如，每隔两小时创建此数据集）或在创建另一个数据集的内容时（例如，在以下情况下创建此数据集）。`myOtherDataset`完成创建内容）。或者，您可以使用以下方式手动生成数据集内容。CreateDatasetContentAPI。

Docker 容器

您可以创建自己的 Docker 容器来打包您的分析工具，或使用以下选项： SageMaker 提供。有关更多信息，请参阅 。Docker 容器. 您可以创建自己的 Docker 容器以打包您的分析工具，或使用SageMaker. 您可以将容器存储在您指定的 Amazon ECR 注册表中，以便可以将其安装在所需的平台上。Docker 容器能够运行您使用 Matlab、Otave、Wise.io、SPSS、R、Fortran、Python、Scala、Java、C++ 等准备的自定义分析代码。有关更多信息，请参阅 。容器化笔记本.

增量时段

增量时段是一系列用户定义的不重叠且连续的时间间隔。通过使用增量时段，您可以使用在上次分析后到达数据存储的新数据创建数据集内容，并对新数据执行分析。您可以通过设置`deltaTime`中的`filters`的部分`queryAction`的数据集。有关更多信息，请参阅 `CreateDataset` API。通常，您还希望设置时间间隔触发器以自动创建数据集内容 (`triggers:schedule:expression`)。这使您可以筛选在特定时间窗口中到达的消息，因此来自以前时间窗口的消息中包含的数据不会重复计数。有关更多信息，请参阅 。示例 6 — 创建具有增量时段的 SQL 数据集 (CLI).

访问 Amazon IoT Analytics

作为的一部分Amazon IoT、Amazon IoT Analytics提供以下界面以使您的设备能够生成数据，并允许您的应用程序与它们生成的数据进行交互：

Amazon Command Line Interface (Amazon CLI)

运行命令Amazon IoT Analytics在 Windows、OS X 和 Linux 上。您可以使用这些命令创建和管理事物、证书、规则及策略。要开始使用，请参阅 Amazon Command Line Interface 用户指南。有关的命令的更多信息Amazon IoT，请参阅iot中的Amazon Command Line Interface参考.

Important

使用`aws iotanalytics`要与之交互的命令Amazon IoT Analytics. 使用`aws iot`命令与 IoT 系统的其他部分交互。

Amazon IoT API

使用 HTTP 或 HTTPS 请求构建您的 IoT 应用程序。您可以使用这些 API 操作创建和管理事物、证书、规则及策略。有关更多信息，请参阅 。操作中的Amazon IoTAPI 参考.

Amazon 软件开发工具包

构建您的Amazon IoT Analytics使用语言特定 API 的应用程序。这些软件开发工具包中封装了 HTTP 和 HTTPS API，并且您可以使用任何受支持的语言进行编程。有关更多信息，请参阅 Amazon 软件开发工具包和工具。

Amazon IoT Device SDK

构建在设备上运行的应用程序，以便向发送消息。Amazon IoT Analytics. 有关更多信息，请参阅 Amazon IoT 软件开发工具包。

Amazon IoT Analytics 控制台

您可以构建组件以在Amazon IoT Analytics控制台.

使用案例

预测性维护

Amazon IoT Analytics提供模板来构建预测性维护模型并将其应用于您的设备。例如，您可以使用Amazon IoT Analytics预测互联货车上的供热制冷系统何时可能发生故障，以便可以重新安排车辆路线以防止货物损坏。或者，汽车制造商可以检测哪些客户的刹车片已磨损并通知他们维修车辆。

主动补充物资

Amazon IoT Analytics您可以构建可实时监控库存的 IoT 应用程序。例如，食品和饮料公司可以分析食品自动售货机中的数据并在供应不足时主动再订购商品。

流程效率评分

与Amazon IoT Analytics，您可以构建不断监控不同过程的效率并采取操作来改进过程的 IoT 应用程序。例如，矿业公司可以通过最大化每次行程的装载量来提高其运矿卡车的效率。与Amazon IoT Analytics时间推移，公司可以确定位置或卡车随时间推移的最高效装载量，并实时比较与目标装载量的任何偏差，并更好地规划领先指南以提高效率。

智能农业

Amazon IoT Analytics可以使用上下文元数据丰富 IoT 设备数据Amazon IoT注册表数据或公共数据源，以便您的分析可以包括时间、位置、温度、海拔及其他环境条件。通过该分析，您可以编写模型来输出建议设备在现场执行的操作。例如，要确定何时浇水，灌溉系统可能使用降雨量数据来丰富湿度传感器数据，从而更高效地使用水资源。

开始使用Amazon IoT Analytics（控制台）

利用本教程创建Amazon IoT Analytics发现有关 IoT 设备数据的有用见解所需的资源（也称为组件）。

注意

- 如果您在以下教程中输入大写字符，Amazon IoT Analytics会自动将它们更改为小写。
- 这些区域有：Amazon IoT Analytics控制台具有一键入门功能，用于创建频道、管道、数据存储和数据集。登录后即可找到此功能Amazon IoT Analytics控制台。
  - 本教程将指导您完成创建自己的Amazon IoT Analytics资源。

按照以下说明创建Amazon IoT Analytics渠道、管道、数据存储和数据集。本教程还向您展示了如何使用Amazon IoT Core控制台发送将要被摄入的消息Amazon IoT Analytics.

主题

登录到 Amazon IoT Analytics 控制台

首先，您必须有一个Amazon账户。如果您已有Amazon账户，导航至https://console.aws.amazon.com/iotanalytics/.

如果您没有Amazon帐户，请执行以下步骤以创建帐户。

创建 Amazon 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

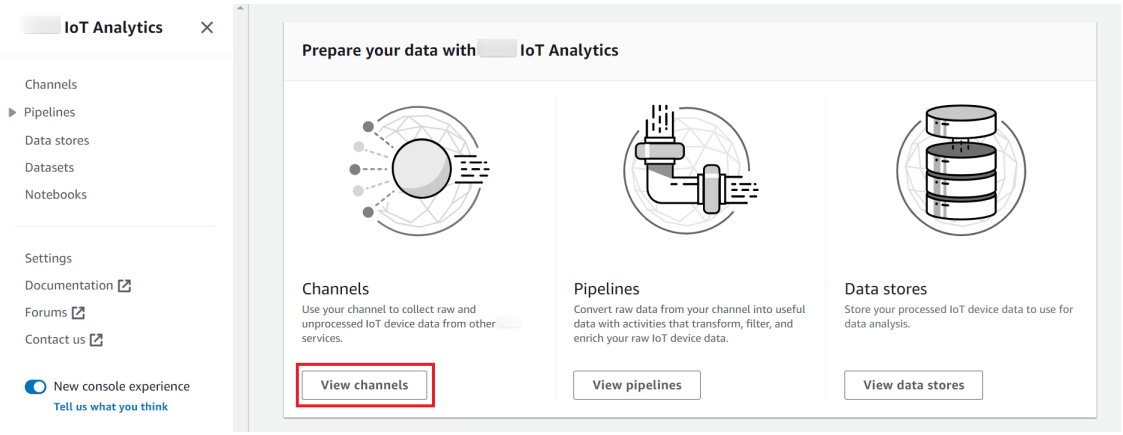
   在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。
3. 登录Amazon Web Services Management Console然后导航到https://console.aws.amazon.com/iotanalytics/.

创建通道

渠道收集和存档原始、未处理和非结构化的 IoT 设备数据。按照以下步骤创建通道。

创建通道

1. 在https://console.aws.amazon.com/iotanalytics/，在使用准备数据Amazon IoT Analytics部分，选择查看通道.



   Tip

   您还可以选择通道从导航窗格中，

2. 在 Channels (通道) 页面上，选择 Create channel (创建通道)。
3. 在指定通道详细信息页面，输入有关您的频道的详细信息。

   a. 输入一个独一无二且易于识别的频道名称。
   b. （可选）对于标签，将一个或多个自定义标签（键值对）添加到通道。标签可帮助您标识您为之创建的资源Amazon IoT Analytics.
   c. 选择 Next（下一步）。

4. Amazon IoT Analytics将未处理的原始IoT ot设备数据存储在 Amazon S3 (Amazon S3) 存储桶中。您可以选择自己的 Amazon S3 存储桶，您可以访问和管理该存储桶，或者Amazon IoT Analytics可以为您管理 Amazon S3 存储桶。

   a. 在本教程中，对于存储类型，选择服务管理存储.
   b. 对于选择存储原始数据的时间，选择无限期.
   c. 选择 Next（下一步）。

5. 在配置来源页面，输入以下信息Amazon IoT Analytics从中收集消息数据Amazon IoT Core.

   a. 输入Amazon IoT Core主题过滤器，例如`update/environment/dht1`. 在本教程的后面部分，您将使用此主题过滤器向您的频道发送消息数据。
   b. 在IAM 角色区域，选择创建新的. 在创建一个新角色窗口，输入名称对于该角色，然后选择创建角色. 这会自动创建一个附加了相应策略的角色。
   c. 选择 Next（下一步）。

6. 查看您的选择，然后选择创建通道.
7. 验证您的新频道是否出现在通道页面。

创建数据存储

数据存储接收和存储您的消息数据。数据存储不是数据库。相反，数据存储是 Amazon S3 存储桶中的可扩展和可查询的存储库。您可以使用多个数据存储来存储来自不同设备或位置的消息。或者，您可以根据您的工作流配置和要求筛选消息数据。

按照以下步骤创建数据存储。

创建数据存储

1. 在https://console.aws.amazon.com/iotanalytics/，在使用准备数据Amazon IoT Analytics部分，选择查看数据存储.
2. 在数据存储页面，选择创建数据存储.
3. 在指定数据存储详细信息页面上，输入有关数据存储的基本信息。

   a. 对于数据存储 ID，输入唯一的数据存储 ID。在创建此 ID 后，您无法更改 ID。
   b. （可选）对于标签，选择添加新标签将一个或多个自定义标签（键值对）添加到数据存储。标签可帮助您标识您为之创建的资源Amazon IoT Analytics.
   c. 选择 Next（下一步）。
4. 在配置存储类型页面，指定如何存储数据。

   a. 对于存储类型，选择服务管理存储.
   b. 对于配置您希望将处理后的数据保留多长时间，选择无限期.
   c. 选择 Next（下一步）。
5. Amazon IoT Analytics数据存储支持 JSON 和 Parquet 文件格式。对于您的数据存储数据格式，选择JSON要么Parquet. 请参阅文件格式 (p. 15)了解有关更多信息Amazon IoT Analytics支持的文件类型。

   选择 Next（下一步）。
6. （可选）Amazon IoT Analytics支持数据存储中的自定义分区，因此您可以查询已修剪的数据以改善延迟。有关支持的自定义分区的更多信息，请参阅自定义分区 (p. 17).

   选择 Next（下一步）。
7. 查看您的选择，然后选择创建数据存储.
8. 验证您的新数据存储是否显示在数据存储页面。

创建管道

必须创建管道才能将通道连接到数据存储。基本管道仅指定收集数据的通道并标识消息发送到的数据存储库。有关更多信息，请参阅 。管道活动.

在本教程中，您将创建一个仅将通道连接到数据存储的管道。稍后，您可以添加管道活动来处理这些数据。

按照以下步骤创建管道。

创建管道

1. 在https://console.aws.amazon.com/iotanalytics/，在使用准备数据Amazon IoT Analytics部分，选择查看管线.

   Tip

   您还可以选择管线从导航窗格中，
2. 在管线页面，选择创建管.
3. 输入有关管道的详细信息。

   a. 中设置管道 ID 和来源，输入管道名称。
   b. 选择您的管道的来源，这是Amazon IoT Analytics您的管道将从中读取消息的频道。
   c. 指定管道的输出，即存储已处理消息数据的数据存储。
   d. （可选）对于标签，将一个或多个自定义标签（键值对）添加到管道。
   e. 在推断消息属性页面，输入属性名称和示例值，从列表中选择一种数据类型，然后选择添加属性.

f. 根据需要对任意数量的属性重复上面的步骤，然后选择下一页.
g. 你现在不会添加任何管道活动。在丰富、转换和筛选消息页面，选择下一页.

4. 查看您的选择，然后选择创建管.
5. 验证您的新管道是否出现在管线页面。

Note

你创建了Amazon IoT Analytics资源，以便他们可以执行以下操作：

- 使用以下命令收集未经处理的原始 IoT 设备消息数据渠道.
- 将您的 IoT 设备消息数据存储在数据存储.
- 清理、筛选、转换和丰富您的数据管道.

接下来，您将创建一个Amazon IoT AnalyticsSQL 数据集用于发现有关您的 IoT 设备的有用见解。

创建数据集

Note

数据集通常是数据集合，这些数据可能以表格形式组织，也可能不以表格形式组织。相比之下，Amazon IoT Analytics通过对数据存储中的数据应用 SQL 查询来创建数据集。

您现在有了一个通道，可以将原始消息数据路由到管道，该管道将数据存储在数据存储库中，可以在那里进行查询。要查询数据，您需要创建一个数据集。数据集包含用于查询数据存储的 SQL 语句和表达式，以及一个在您指定的日期和时间重复查询的可选计划。您可以使用类似于的表达式亚马逊 CloudWatch 计划表达式创建可选时间表。

创建数据集

1. 在https://console.aws.amazon.com/iotanalytics/，在左侧导航窗格，选择数据集.
2. 在创建数据集页面，选择创建 SQL.
3. 在指定数据集的详细信息页面上，指定数据集的详细信息。

   a. 输入数据集的名称。
   b. 对于数据存储源，选择用于标识您之前创建的数据存储的唯一 ID。
   c. （可选）对于标签，将一个或多个自定义标签（键值对）添加到数据集。

4. 使用 SQL 表达式查询您的数据并回答分析问题。您的查询结果存储在此数据集中。

   a. 在作者查询字段中，输入一个 SQL 查询，该查询使用通配符最多显示五行数据。

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

   有关支持的 S3 功能的更多信息，请参阅Amazon IoT Analytics，请参阅中的 SQL 表达式Amazon IoT Analytics (p. 86).

   b. 您可以选择测试查询以验证您的输入是否正确，并在查询后的表格中显示结果。

   Note

   - 此时，在本教程中，您的数据存储可能为空。在空数据存储上运行 SQL 查询不会返回结果，因此您可能只能看到__dt.
   - 必须谨慎地将 SQL 查询限制到合理的大小，这样它就不会长时间运行，因为 Athena限制正在运行的查询的最大数量. 因此，必须谨慎地将 SQL 查询限制为合理的大小。

我们建议使用`LIMIT`测试期间查询中的子句。测试成功后，您可以删除此子句。

5. （可选）当您使用指定时间范围内的数据创建数据集内容时，某些数据可能无法及时到达以进行处理。要允许延迟，可以指定偏移量或增量。有关更多信息，请参阅 通过亚马逊获取延迟数据通知CloudWatch 事件 (p. 108)。

   此时您不会配置数据选择过滤器。在配置数据选择过滤器页面，选择下一页.
6. （可选）您可以计划定期运行此查询以刷新数据集。可以随时创建和编辑数据集计划。

   此时你不会计划重复运行查询，所以设置查询计划页面选择下一页.
7. Amazon IoT Analytics将创建此数据集内容的版本并在指定时间段内存储您的分析结果。我们建议使用90 天，但您可以选择设置自定义保留政策。您也可以限制数据集内容的存储版本数量。

   您可以将默认的数据集保留期用作无限期并保持版本控制已禁用。在配置分析结果页面，选择下一页.
8. （可选）您可以将数据集结果的传输规则配置到特定目的地，例如Amazon IoT Events.

   你不会在本教程的其他地方提供结果，所以在配置数据集内容分发规则页面，选择下一页.
9. 查看您的选择，然后选择创建数据集.
10. 验证您的新数据集是否出现在数据集页面。

使用发送消息数据Amazon IoT

如果您有一个将数据路由到管道的通道，该通道将数据存储在可以查询的数据存储中，那么您就可以将IoT设备数据发送到Amazon IoT Analytics. 你可以将数据发送到Amazon IoT Analytics通过使用以下选项：

- 使用Amazon IoT消息代理。
- 使用Amazon IoT Analytics BatchPutMessageAPI 操作。

在以下步骤中，您将从发送消息数据Amazon IoT消息代理Amazon IoT Core控制台这样Amazon IoT Analytics可以摄取这些数据。

> **Note**
>
> 在为消息创建主题名称时，请注意以下几点：
>
> - 主题名称不区分大小写。字段已命名`example`和`EXAMPLE`在相同的有效载荷中被视为重复的。
> - 主题名称不能以`$`字符。以开头的话题`$`是保留的主题，只能由以下人员使用Amazon IoT.
> - 不要在主题名称中包含可识别个人身份的信息，因为这些信息可能出现在未加密的通信和报告中。
> - Amazon IoT Core无法在两者之间发送消息Amazon账户或Amazon区域。

使用以下命令发送消息数据Amazon IoT

1. 登录到 Amazon IoT 控制台。
2. 在导航窗格中，选择测试，然后选择MQTT 测试客户端.
3. 在MQTT 测试客户端页面，选择向主题发布.
4. 对于主题名称，输入一个与您在创建频道时输入的主题过滤器相匹配的名称。此示例使用 `update/environment/dht1`。
5. 对于消息负载，输入以下 JSON 内容。

```
{
  "thingid": "dht1",
```

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

6. （可选）选择添加配置获取其他消息协议选项。
7. 选择 Publish。

   这会发布一条由您的频道捕获的消息。然后，您的管道将消息路由到您的数据存储。

查看进度Amazon IoT消息

您可以按照以下步骤检查消息是否被收录到您的频道中。

检查进度Amazon IoT消息

1. 登录https://console.aws.amazon.com/iotanalytics/.
2. 在导航窗格中，选择通道，然后选择您之前创建的频道名称。
3. 在通道的详细信息页面，向下滚动到监控部分，然后调整显示的时间范围 (1h 3h 12h 1d 3d 1w)。选择一个值，例如1w查看上周的数据。

您可以使用类似的功能来监视管道活动运行时和错误管道的详细信息页面。在本教程中，您没有将活动指定为管道的一部分，因此您应该不会看到任何运行时错误。

监控管道活动

1. 在导航窗格中，选择管线，然后选择您之前创建的管道的名称。
2. 在管道的详细信息页面，向下滚动到监控部分，然后通过选择其中一个时间范围指示器来调整显示的时间范围（1h 3h 12h 1d 3d 1w)。

访问查询结果

数据集内容是一个包含 CSV 格式的查询结果的文件。

1. 在https://console.aws.amazon.com/iotanalytics/，在左侧导航窗格中，选择数据集.
2. 在数据集页面上，选择您之前创建的数据集的名称。
3. 在数据集信息页面上，在右上角，选择立即运行.
4. 要检查数据集是否准备就绪，请在数据集下方查找一条类似于以下内容的消息您已成功启动数据集查询.这些区域有：数据集内容选项卡包含查询结果并显示成功了.
5. 要预览成功查询的结果，请在数据集内容选项卡上，选择查询名称。要查看或保存包含查询结果的 CSV 文件，请选择下载.

   Note

   Amazon IoT Analytics可以将 Jupyter 笔记本的 HTML 部分嵌入到数据集内容页面。有关更多信息，请参阅 可视化Amazon IoT Analytics使用控制台进行数据 (p. 81)。

探索您的数据

您可以通过多种方式存储、分析和可视化数据。

Amazon Simple Storage Service

您可以将数据集内容发送到Amazon S3bucket，支持与现有数据湖集成或从内部应用程序和可视化工具进行访问。请参阅字段中返回的子位置`contentDeliveryRules::destination::s3DestinationConfiguration`在里面CreateDataset操作。

Amazon IoT Events

您可以将数据集内容作为输入发送到Amazon IoT Events，该服务可让您了解如何监控设备或进程中的故障情况或操作中的更改，并在发生此类事件时启动其他措施。

为此，请使用创建数据集CreateDataset操作并指定Amazon IoT Events在字段中输入`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. 还必须指定`roleArn`一个角色，它授予Amazon IoT Analytics运行权限`iotevents:BatchPutMessage`. 每当创建数据集内容时，Amazon IoT Analytics会将每个数据集内容条目作为消息发送给指定的Amazon IoT Events输入。例如，如果您的数据集包含以下内容。

```
"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"
...
```

然后Amazon IoT Analytics发送包含如下字段的消息。

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

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

你会想要创建一个Amazon IoT Events可识别您感兴趣的字段的输入（其中一个或多个`what,who,dt`）并创建一个Amazon IoT Events探测器模型，在事件中使用这些输入字段来触发动作或设置内部变量。

Jupyter Notebook

Japyter notebook是一个开源解决方案，用于使用脚本语言进行临时数据探索和高级分析。您可以深入研究并应用更复杂的分析，并在 IoT 设备数据上使用机器学习方法，例如 k-means 聚类和回归模型进行预测。

Amazon IoT Analytics使用 Amazon SageMaker 笔记本实例来托管其 Jupyter 笔记本。在创建笔记本实例之前，您必须创建笔记本实例之间的关系Amazon IoT Analytics和Amazon SageMaker：

1. 导航到SageMaker 控制台并创建笔记本实例：
   a. 填写详细信息，然后选择 Create a new role (创建新角色)。记录角色 ARN。
   b. 创建笔记本实例。
2. 转到IAM 控制台并修改 SageMaker角色：
   a. 打开角色。它应该具有一个托管策略。
   b. 选择添加内联策略，然后对于服务，选择IotAnalyts. 选择选择动作，然后输入**`GetDatasetContent`**在搜索框中进行选择。请选择查看策略。
   c. 查看策略的准确性，输入名称，然后选择创建策略.

这为新创建的角色提供了从中读取数据集的权限Amazon IoT Analytics.

1. 返回https://console.aws.amazon.com/iotanalytics/，然后在左侧导航窗格中，选择笔记本. 在笔记本页面，选择创建笔记本.
2. 在选择模板页面，选择Iota 空白模板.
3. 在设置笔记本页面上，输入笔记本的名称。中选择数据集源，选择然后选择您之前创建的数据集。中选择笔记本实例，选择您在中创建的笔记本实例 SageMaker.
4. 查看您的选择后，选择创建笔记本.
5. 在笔记本页面，您的笔记本实例将在亚马逊 SageMaker控制台。

笔记本模板

这些区域有：Amazon IoT Analytics笔记本模板包含Amazon创作机器学习模型和可视化以帮助您开始Amazon IoT Analytics使用案例。您可以使用这些笔记本模板来了解更多信息，也可以重复使用它们以适应您的IoT 设备数据并立即创造价值。

您可以在以下笔记本模板中找到Amazon IoT Analytics控制台：

- 检测上下文异常— 使用泊松指数加权移动平均线 (PEWMA) 模型在测得的风速中应用情境异常检测。
- 太阳能电池板产量预测— 应用分段、季节和线性时间序列模型来预测太阳能电池板的输出。
- 喷气发动机的预测性维护— 应用多变量长短期记忆 (LSTM) 神经网络和逻辑回归来预测喷气发动机故障。
- 智能家居客户细分— 应用k-means和主成分分析（PCA）分析来检测智能家居使用数据中的不同客户群体。
- 智慧城市拥堵预测— 应用LSTM预测城市高速公路的利用率。
- 智慧城市空气质量预测— 应用LSTM预测城市中心的颗粒物污染。

Amazon IoT Analytics 入门

本节讨论了用于收集、存储、处理和查询设备数据时使用的基本命令。Amazon IoT Analytics. 此处显示的示例使用Amazon Command Line Interface(Amazon CLI)。有关Amazon CLI，请参阅Amazon Command Line Interface用户指南. 有关可用于的 CLI 命令的更多信息Amazon IoT，请参阅iot中的Amazon Command Line Interface参考.

Important

使用`aws iotanalytics`要与之交互的命令Amazon IoT Analytics使用Amazon CLI. 使用`aws iot`命令与 IoT 系统的其他部分交互，并使用Amazon CLI.

Note

当你输入的名字时请注意Amazon IoT Analytics系统会自动将您使用的所有大写字母更改为小写字母的实体（通道、数据集、数据存储和管道）。实体的名称必须以小写字母开头，并且只能包含小写字母、下划线和数字。

创建通道

通道收集并存档未处理的原始消息数据，然后再将该数据发布到管道。传入消息被发送到一个通道，因此，第一步是为您的数据创建一个通道。

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

如果要Amazon IoT要摄取的消息Amazon IoT Analytics，您可以创建Amazon IoT规则引擎规则将消息发送到此频道。这将在稍后显示在将数据提取到 Amazon IoT Analytics (p. 19). 将数据传送到通道的另一种方法是使用Amazon IoT Analytics命令`BatchPutMessage`.

列出您已经创建的通道：

```
aws iotanalytics list-channels
```

获取有关通道的更多信息。

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

未处理的通道消息存储在 Amazon S3 存储桶中，Amazon IoT Analytics，或者在你管理的一个中。可以使用 `channelStorage` 参数指定相应的存储桶。默认设置是服务管理的 Amazon S3 存储桶。如果您选择将渠道消息存储在您管理的 Amazon S3 存储桶中，则必须授予Amazon IoT Analytics代表您在 Amazon S3 存储桶上执行以下操作的权限：`s3:GetBucketLocation`（验证存储桶位置）`s3:PutObject`（商店），`s3:GetObject`（阅读），`s3:ListBucket`（再处理）。

Example

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

```
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::my-iot-analytics-bucket",
                "arn:aws:s3:::my-iot-analytics-bucket/*"
            ]
        }
    ]
}
```

如果更改客户管理的通道存储的选项或权限，您可能需要重新处理通道数据，以确保以前提取的数据包含在数据集内容中。请参阅重新处理通道数据.

创建数据存储

数据存储接收并存储您的消息。它不是数据库，而是一个可扩展且可查询的消息存储库。您可以创建多个数据存储来存储来自不同设备或位置的消息，也可以使用单个数据存储接收所有消息：Amazon IoT消息。

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

列出您已经创建的数据存储。

```
aws iotanalytics list-datastores
```

获取有关数据存储的更多信息。

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

Amazon S3 策略Amazon IoT Analytics资源

您可以将处理过的数据存储消息存储在 Amazon S3 存储桶中Amazon IoT Analytics或者是你管理的。创建数据存储时，使用选择所需的 Amazon S3 存储桶`datastoreStorage`API 参数。默认设置是服务管理的 Amazon S3 存储桶。

如果您选择将数据存储消息存储在您管理的 Amazon S3 存储桶中，则必须授予Amazon IoT Analytics允许您在 Amazon S3 存储桶上执行这些操作：

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

如果您使用数据存储作为 SQL 查询数据集的源，请设置一个 Amazon S3 存储桶策略来授予Amazon IoT Analytics允许对您的存储桶中的内容调用 Amazon Athena 查询。

> **Note**
>
> 我们建议您指定`aws:SourceArn`在您的存储桶策略中，有助于避免出现混淆代理安全问题。这通过仅允许来自指定账户的请求来限制访问。有关混淆代理人问题的更多信息，请参阅the section called “跨服务混淆代理问题防范” (p. 94).

以下是授予这些所需权限的存储桶策略示例。

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

有关更多信息，请参阅 。跨账户访问权限在Amazon Athena 用户指南.

Note

如果您更新客户管理的数据存储的选项或权限，则可能需要重新处理渠道数据，以确保数据集内容中包含以前摄取的所有数据。有关更多信息，请参阅 。重新处理通道数据.

文件格式

Amazon IoT Analytics数据存储目前支持 JSON 和 Parquet 文件格式。默认文件格式为 JSON。

- JSON（JavaScript 对象表示法）-支持名称-值对和有序值列表的文本格式。
- Apache Parquet-用于高效存储和查询大量数据的列式存储格式。

配置文件格式Amazon IoT Analytics数据存储，你可以使用`FileFormatConfiguration`创建数据存储时的对象。

`fileFormatConfiguration`

包含文件格式的配置信息。Amazon IoT Analytics数据存储支持 JSON 和 Parquet。

默认文件格式为 JSON。只能指定一种格式。创建数据存储后，无法更改文件格式。

`jsonConfiguration`

包含 JSON 格式的配置信息。

`parquetConfiguration`

包含 Parquet 格式的配置信息。

`schemaDefinition`

定义架构所需的信息。

`columns`

指定存储数据的一个或多个列。

每个架构最多可有 100 列。每列最多可有 100 种嵌套类型。

`name`

列的名称。

长度限制：1-255 个字符。

`type`

数据的类型。有关支持的数据类型的更多信息，请参阅。常见数据类型中的Amazon Glue开发人员指南.

长度限制：1-131072 个字符。

Amazon IoT Analytics支持列在Amazon Athena 中的数据类型页面，除了`DECIMAL(`*`precision`*`,` *`scale`*`)-`*`precision`*.

创建数据存储（控制台）

以下过程演示如何创建数据存储，以 Parquet 格式保存数据存储。

创建数据存储

1. 登录到https://console.aws.amazon.com/iotanalytics/.
2. 在导航窗格中，选择数据存储.
3. 在存储库的数据存储页面上，选择创建数据存储.
4. 在存储库的指定数据存储页面中，输入有关数据存储的基本信息。
   a. 适用于数据存储 ID中，输入唯一数据存储 ID。您在创建此 ID 之后无法更改其。
   b. （可选）对于标签，选择添加新标签将一个或多个自定义标签（键值对）添加到数据存储中。标签可帮助您标识您为其创建的资源Amazon IoT Analytics.
   c. 选择 Next（下一步）。
5. 在存储库的配置存储类型页面中，指定如何存储数据。
   a. 适用于存储类型，选择服务托管存储.
   b. 适用于配置您希望将处理后的数据保留多长时间，选择无限期.
   c. 选择 Next（下一步）。
6. 在存储库的配置数据格式页面中，定义数据记录的结构和格式。
   a. 适用于Classification，选择Parquet. 创建数据存储后，无法更改此格式。
   b. 适用于推理来源，选择JSON 字符串对于数据存储。

c. 适用于字符串以 JSON 格式输入架构，例如以下示例。

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

d. 选择推断架构.

e. UNDER配置 Parquet 架构中，确认格式与您的 JSON 示例匹配。如果格式不匹配，请手动更新 Pincia 模式。

- 如果您希望模式显示更多列，请选择添加新列，输入列名称，然后选择数据类型。

  Note

  默认情况下，架构可使用 100 列。有关更多信息，请参阅 Amazon IoT Analytics 配额。

- 您可以更改现有列的数据类型。有关支持的数据类型的更多信息，请参阅。常见数据类型中的Amazon Glue开发人员指南.

  Note

  在您创建数据存储之后，便无法更改现有列的数据类型。

- 要删除现有列，请选择删除列.

f. 选择 Next（下一步）。

7. （可选）Amazon IoT Analytics支持数据存储中的自定义分区，因此您可以查询修剪的数据以提高延迟。有关支持的定制分区的更多信息，请参阅自定义分区 (p. 17).

   选择 Next（下一步）。

8. 在存储库的审核和创建页面，查看您的选择，然后选择创建数据存储.

   Important

   创建数据存储后，无法更改列的数据存储 ID、文件格式或数据类型。

9. 验证您的新数据存储是否显示在数据存储页.

自定义分区

Amazon IoT Analytics支持数据分区，以便您可以在数据存储中组织数据。当您使用数据分区来组织数据时，可以查询修剪的数据。这样可以减少每个查询扫描的数据量，并延长延迟。

您可以根据消息数据属性或通过管道活动添加的属性对数据进行分区。

要开始使用，请在数据存储中启用数据分区。指定一个或多个数据分区维度，然后将分区数据存储连接到Amazon IoT Analytics管道。然后，编写利用`WHERE`子句优化性能。

创建数据存储（控制台）

下面的过程演示如何使用自定义分区创建数据存储。

创建数据存储

1. 登录到 Amazon IoT Analytics 控制台。
2. 在导航窗格中，选择数据存储.
3. 在存储库的数据存储页面上，选择创建数据存储.

4. 在存储库的指定数据存储详情页面中，输入有关数据存储的基本信息。

   a. 适用于数据存储 ID中，输入唯一的数据存储 ID。创建此 ID 后，您无法更改其 ID。
   b. （可选）对于标签，选择添加新标签将一个或多个自定义标签（键值对）添加到数据存储中。标签可帮助您识别为其创建的资源Amazon IoT Analytics.
   c. 选择 Next（下一步）。
5. 在存储库的配置存储类型页面中，指定如何存储数据。

   a. 适用于存储类型，选择服务托管存储.
   b. 适用于配置您希望将处理后的数据保留多长时间，选择无限期.
   c. 选择 Next（下一步）。
6. 在存储库的配置数据格式页面中，定义数据记录的结构和格式。

   a. 对于数据存储格式Classification，选择JSON要么Parquet. 有关 的更多信息Amazon IoT Analytics支持的文件类型，请参阅文件格式 (p. 15).

      Note

      创建数据存储后，无法更改此格式。
   b. 选择 Next（下一步）。
7. 为此数据存储创建自定义分区。

   a. 适用于添加数据分区，选择启用.
   b. 适用于数据分区源中，指定有关分区源的基本信息。

      选择源示例，然后选择Amazon IoT Analytics为此数据存储收集消息的渠道。
   c. 适用于消息示例属性中，选择要用于对数据存储进行分区的消息属性。然后，将您的选择添加为属性分区维度或时间戳分区维操作.

      Note

      您只能将一个时间戳分区添加到数据存储中。
   d. 适用于自定义数据存储分区维度中，定义有关分区维度的基本信息。您在上一步中选择的每个消息示例属性都将成为分区的维度。使用以下选项自定义每个维度：

      - 分区类型-指定此分区维度是否为属性或者时间戳分区类型。
      - 属性名称和维度名称-默认为Amazon IoT Analytics将使用您选择的消息示例属性的名称作为属性分区维的标识符。编辑属性名称以自定义分区维的名称。您可以在`WHERE`子句优化查询性能。
        - 任何分区属性维度的名称都带有前缀`__partition_`.
        - 对于时间戳分区类型，Amazon IoT Analytics使用名称创建以下四个维度`__year`、`__month`、`__day`、`__hour`.
      - Orter-重新排列分区维度以提高查询的延迟时间。

      适用于时间戳格式中，通过匹配消息数据中提取的时间戳来指定时间戳分区的格式。您可以选择其中之一Amazon IoT Analytics列出的格式选项，或者指定与数据格式匹配的格式选项。了解有关的详细信息日期时间格式化器.

      要添加不是消息属性的新维度，请选择添加新分区.
   e. 选择 Next（下一步）。
8. 在存储库的审核和创建页面，查看您的选择，然后选择创建数据存储.

   Important

   - 创建数据存储后，无法更改数据存储 ID。
   - 要编辑现有分区，必须创建另一个数据存储并通过管道重新处理数据。

9. 验证您的新数据存储是否显示在数据存储页.

创建管道

管道使用来自通道的消息，并在将消息存储在数据存储之前允许您处理和筛选消息。要将通道连接到数据存储，请创建一个管道。最简单的管道除了指定收集数据的通道和标识将向其发送消息的数据存储之外，不包含任何其他活动。有关更复杂管道的信息，请参阅Pipeline 活动.

在开始时，我们建议您创建一个管道，该管道仅将通道连接到数据存储，而不执行任何其他操作。然后，在确认原始数据流到数据存储后，您可以引入额外的管道活动以处理该数据。

运行以下命令以创建管道。

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

这些区域有：mypipeline.json文件包含以下内容。

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

运行以下命令列出现有管道。

```
aws iotanalytics list-pipelines
```

运行以下命令查看单个管道的配置。

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

将数据提取到 Amazon IoT Analytics

如果您具有一个通道以将数据路由到管道，该管道将数据存储在数据存储以在其中对其进行查询，则可以将消息数据发送到Amazon IoT Analytics. 此处我们介绍了两种将数据传送到的方法Amazon IoT Analytics. 你可以使用Amazon IoT消息代理或使用Amazon IoT Analytics `BatchPutMessage`API。

主题

使用Amazon IoT消息代理

使用Amazon IoT消息代理，请使用Amazon IoT规则引擎。该规则将具有特定主题的消息路由到Amazon IoT Analytics. 但首先，该规则要求您创建一个角色以授予所需的权限。

创建 IAM 角色

要拥有Amazon IoT发送到的消息Amazon IoT Analytics频道，你设置了一条规则。但首先，您必须创建一个 IAM 角色，用于授予该规则权限以将消息数据发送到Amazon IoT Analytics频道。

运行以下 命令以创建角色。

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

的内容`arpd.json`您的文件应类似以下内容。

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

然后，将策略文档附加到该角色。

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

的内容`pd.json`您的文件应类似以下内容。

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

创建Amazon IoT规则

创建Amazon IoT将消息发送到您的通道的规则。

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

的内容`rule.json`您的文件应类似以下内容。

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

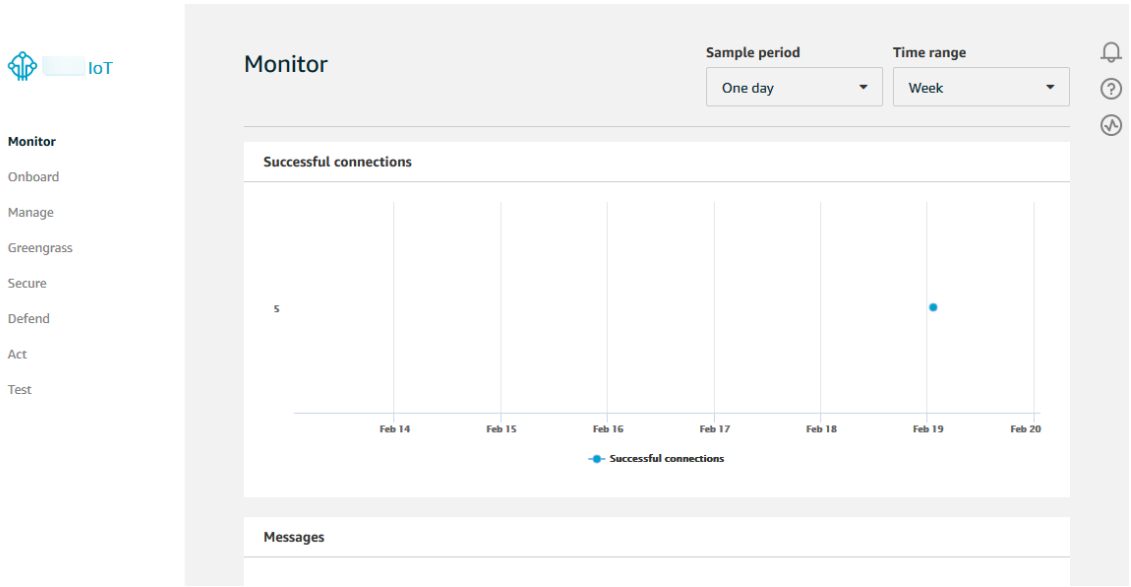
将 `iot/test` 替换为应路由的消息的 MQTT 主题。将通道名称和角色替换为您在前面章节中创建的通道和角色。

将 MQTT 消息发送到Amazon IoT Analytics

在将规则加入通道、将通道加入管道并将管道加入数据存储后，与规则匹配的所有数据现在流入。Amazon IoT Analytics到数据存储以供查询。要测试此内容，您可以使用Amazon IoT发送邮件的控制台。

> **Note**
>
> 您发送至的消息负载（数据）的字段名称Amazon IoT Analytics.
>
> - 必须仅包含字母数字字符和下划线 (_)；不允许使用其他特殊字符。
> - 必须以字母字符或单个下划线 (_) 开头。
> - 不能包含连字符 (-)。
> - 在正则表达式术语中："`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`"。
> - 长度不能超过 255 个字符
> - 不区分大小写。命名的字段`foo`和`FOO`在同一负载中被认为是重复的。
>
> 例如，在消息负载中，`{"temp_01": 29}` 或 `{"_temp_01": 29}` 有效，但 `{"temp-01": 29}`、`{"01_temp": 29}` 或 `{"__temp_01": 29}` 无效。

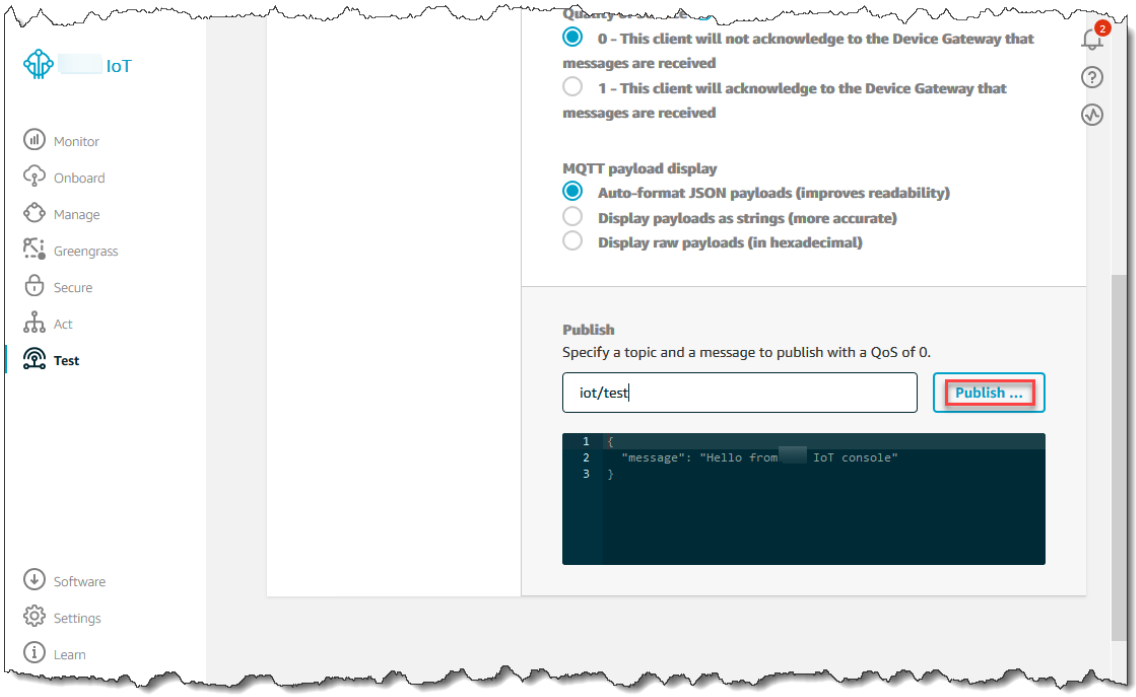
1. 在 Amazon IoT 控制台的左侧导航窗格中选择 Test (测试)。



2. 在 MQTT 客户端页面上，在 Publish 部分的 Specify a topic 中，键入 **`iot/test`**。在消息负载部分，请验证以下 JSON 内容是否存在，如果不存在，则键入它们。

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

3. 选择 Publish to topic（发布到主题）。



这会发布一条消息，该消息路由到您之前创建的数据存储。

使用 BatchPutMessage API

将消息数据输入的另一种方法Amazon IoT Analytics是使用`BatchPutMessage`API 命令。此方法不要求您设置Amazon IoT将具有特定主题的消息路由到您的通道。但它确实要求将数据/消息发送到通道的设备能够运行使用AmazonSDK 或能够使用Amazon CLI调用`BatchPutMessage`.

1. 创建文件`messages.json`其中包含要发送的消息（在本示例中，只发送一条消息）。

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

2. 运行 `batch-put-message` 命令。

```
aws iotanalytics batch-put-message  --channel-name mychannel  --messages file://
messages.json
```

如果没有错误，将显示以下输出。

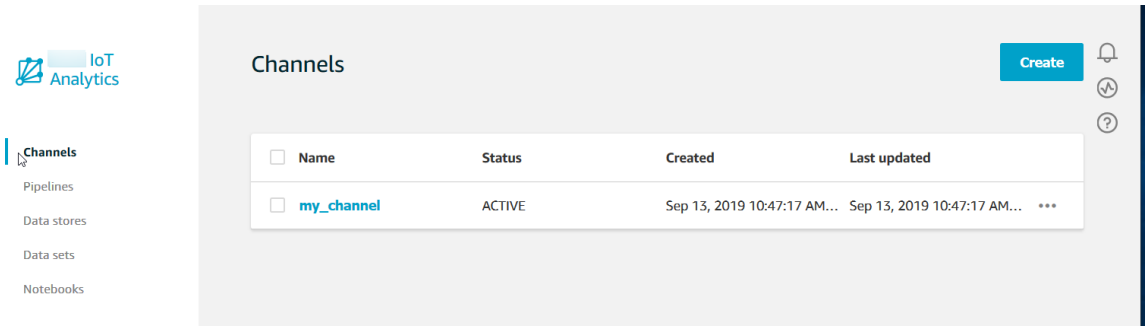
```
{
```

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

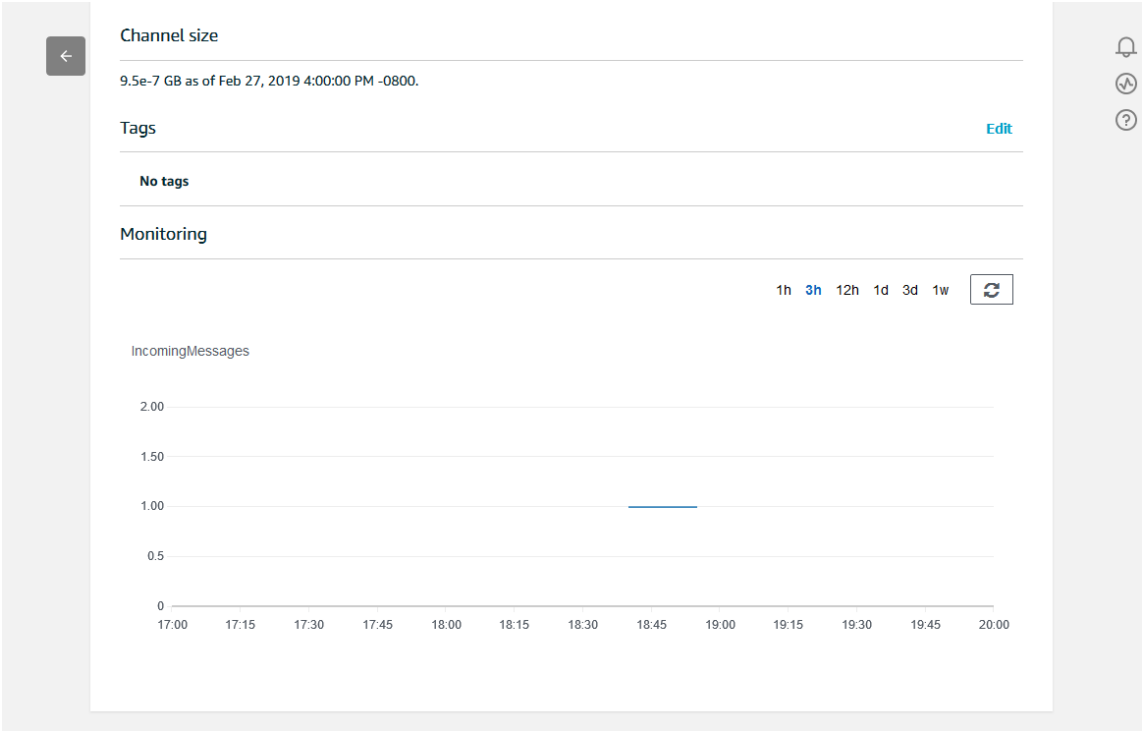
监控摄入的数据

您可以使用Amazon IoT Analytics控制台。

1. 在Amazon IoT Analytics控制台，在左侧导航窗格中，选择准备和（如有必要）选择Channel，然后选择之前创建的通道的名称。

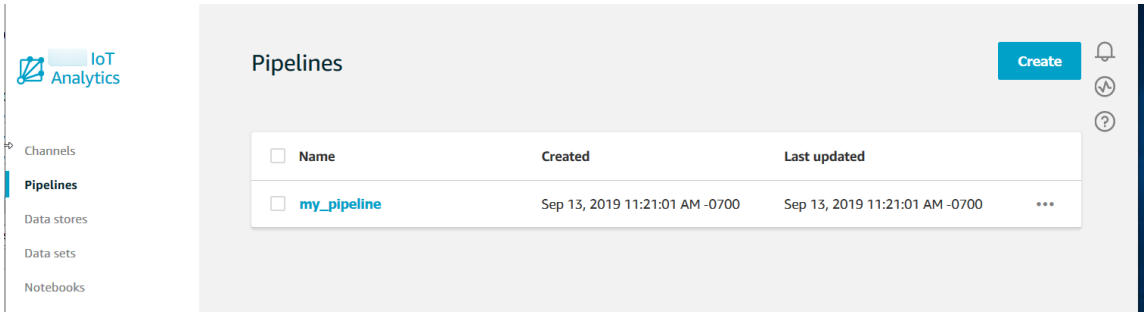


2. 在通道详细信息页面上，向下滚动到监控部分。根据需要，选择其中的一个时间范围指示符（1 小时 3 小时 12 小时 1 天 3 天 1 周）以调整显示的时间范围。将会看到一条图形线，以指示在指定时间范围内提取到该通道的消息数。

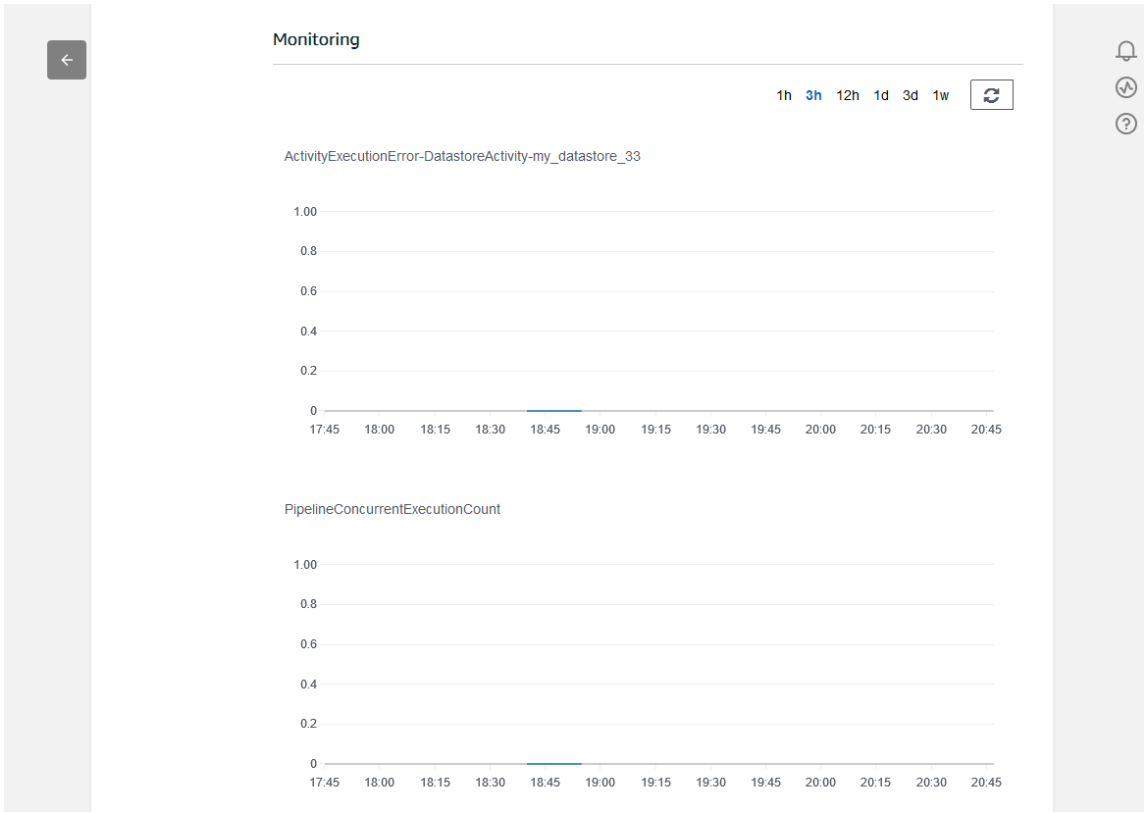


可以使用类似的监控功能以检查管道活动执行。您可以在管道的详细信息页面上监控活动执行错误。如果未将活动指定为管道的一部分，将显示 0 个执行错误。

1. 在Amazon IoT Analytics控制台，在左侧导航窗格中，选择准备然后选择管道，然后选择之前创建的管道的名称。



2. 在管道详细信息页面上，向下滚动到监控部分。根据需要，选择其中的一个时间范围指示符（1 小时 3 小时 12 小时 1 天 3 天 1 周）以调整显示的时间范围。将会看到一条图形线，以指示指示指定时间范围内的管道活动执行错误数。



创建数据集

您可以通过创建 SQL 数据集或容器数据集来从数据存储中检索数据。Amazon IoT Analytics可以查询数据来回答分析问题。尽管数据存储不是数据库，但您可以使用 SQL 表达式来查询数据并生成存储在数据集中的结果。

主题

查询数据

要查询数据，您需要创建一个数据集。数据集包含用于查询数据存储的 SQL 以及在您选择的日期和时间重复该查询的可选计划。您可以使用类似于以下的表达式创建可选亚马逊 CloudWatch 计划表达式.

运行以下命令以创建数据集。

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

其中mydataset.json文件包含以下内容。

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

运行以下命令以通过执行查询创建数据集内容。

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

等待几分钟，在创建完数据集内容后，便可继续。

访问查询的数据

查询结果是以 CSV 文件格式存储的数据集内容。将通过 Amazon S3 向您提供该文件。以下示例说明了如何检查结果是否已准备就绪、文件是否已下载。

运行以下 get-dataset-content 命令：

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

如果您的数据集包含任何数据，则输出来自get-dataset-content，有"state": "SUCCEEDED"中的status字段中，类似以下示例。

```
{
    "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` 是输出结果的签名 URL。它在较短的一段时间内有效 (几个小时)。根据您的工作流，您可能需要在访问内容之前始终调用 `get-dataset-content`，因为调用此命令会生成新的签名 URL。

了解Amazon IoT Analytics数据

您可以通过多种方法存储、分析和可视化Amazon IoT Analytics数据。

本页面上的主题：

Amazon S3

您可以将数据集内容发送到Amazon Simple Storage Service (Amazon S3)存储桶，允许与现有数据湖集成，或者从内部应用程序和可视化工具中进行访问 查看该领域`contentDeliveryRules::destination::s3DestinationConfiguration`在CreateDataset.

Amazon IoT Events

您可以将数据集内容作为输入发送到Amazon IoT Events，该服务允许您监控设备或进程中的故障情况或操作更改，并在发生此类事件时触发其他操作。

为此，请使用创建数据集CreateDataset然后指定Amazon IoT Events字段中的输入`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. 还必须指定`roleArn`授予的角色Amazon IoT Analytics执行 “iotevent: BatchputMessage” 的权限。无论何时创建数据集的内容，Amazon IoT Analytics将将每个数据集内容条目作为消息发送到指定的Amazon IoT Events输入。例如，如果数据集包含以下内容：

```
"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"
...
```

然后Amazon IoT Analytics将发送包含这样的字段的消息：

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

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

你会想创建一个Amazon IoT Events识别您感兴趣的字段的输入 (一个或多个`what`、`who`、`dt`) 然后创建一个Amazon IoT Events在事件中使用这些输入字段来触发操作或设置内部变量的检测器模型。

Jupyter Notebook

Amazon IoT Analytics数据集也可以由 Jupyter 笔记本直接使用，以执行高级分析和数据探索。Jupyter 笔记本是一个开源解决方案。您可以从 http://jupyter.org/install.html 安装并下载。还额外提供与 SageMaker 的集成，后者是一项 Amazon 托管的笔记本解决方案。

保留数据集的多个版本

您可以通过指定数据集的值来选择要保留的数据集内容的版本以及保留多长时间。`retentionPeriod` and `versioningConfiguration`调用时的字段CreateDataset和UpdateDatasetAPI：

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

这两个参数的设置一起使用，以通过以下方式确定保留的数据集内容版本数和保留时间。

	retentionPeriod [未指定]	retentionPeriod： 无限 = TRUE，numberOfDays = 未设置	retentionPeriod： 无限 = FALSE，numberOfDays = X
versioningConfiguration： [未指定]	仅将最新版本以及最新的成功版本（如果不同）保留 90 天。	仅无限期保留最新版本以及最新的成功版本（如果不同）。	仅将最新版本以及最新的成功版本（如果不同）保留 X 天。
versioningConfiguration： unlimited = TRUE，未设置 maxVersions	将保留过去 90 天的所有版本，而无论具有多少个版本。	对保留的版本数没有任何限制。	将保留过去 X 天的所有版本，而无论具有多少个版本。
versioningConfiguration： unlimited = FALSE，maxVersions = Y	将保留过去 90 天内不超过 Y 个版本。	将最多保留 Y 个版本，而无论它们存在多长时间。	将保留过去 X 天内不超过 Y 个版本。

消息负载语法

您发送至的消息负载（数据）的字段名称Amazon IoT Analytics：

- 必须仅包含字母数字字字符和下划线 (_)；不允许使用其他特殊字符。
- 必须以字母字符或单个下划线 (_) 开头。
- 不能包含连字符 (-)。
- 在正则表达式术语中：”`^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`“。
- 长度不能超过 255 个字符。
- 不区分大小写。在同一负载中，名为 “foo” 和 “FOO” 的字段被视为重复字段。

例如，在消息负载中，`{"temp_01": 29}` 或 `{"_temp_01": 29}` 有效，但 `{"temp-01": 29}`、`{"01_temp": 29}` 或 `{"__temp_01": 29}` 无效。

使用Amazon IoT SiteWise数据

Amazon IoT SiteWise是一种托管式服务，可用于从工业设备中大规模收集、建模、分析和可视化显示来自工业设备的数据。该服务提供了一个资产建模框架，用于构建工业设备、流程和设施的表示形式。

与Amazon IoT SiteWise使用资产模型，您可以定义要使用的工业设备数据以及如何将数据处理为复杂的指标。您可以配置资产模型以在Amazon云。有关更多信息，请参阅 Amazon IoT SiteWise 用户指南。

Amazon IoT Analytics已与 集成Amazon IoT SiteWise所以你可以运行和安排 SQL 查询Amazon IoT SiteWiseDATA。开始查询你的Amazon IoT SiteWisedata，请按照中的过程创建数据存储配置存储设置中的Amazon IoT SiteWise用户指南. 然后，请按照中的步骤操作使用创建数据集Amazon IoT SiteWise数据（控制台）(p. 28)或者使用创建数据集Amazon IoT SiteWiseDATA (数据Amazon CLI）(p. 29)来创建Amazon IoT Analytics数据集并对您的工业数据运行 SQL 查询。

主题

创建Amazon IoT Analytics数据集Amazon IoT SiteWise数据

网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的Amazon IoT Analytics数据集包含用于查询数据存储中的数据的 SQL 语句和表达式以及在您指定的日期和时间重复该查询的可选计划。你可以使用类似于亚马逊 CloudWatch 计划表达式以创建可选的时间表。

> Note
>
> 数据集通常是可能或可能不以表格形式组织的数据集合。相比之下，Amazon IoT Analytics通过对数据存储中的数据应用 SQL 查询来创建数据集。

按照以下步骤开始为您的数据集创建数据集Amazon IoT SiteWiseDATA。

主题

使用创建数据集Amazon IoT SiteWise数据（控制台）

使用以下步骤在Amazon IoT Analytics你的控制台Amazon IoT SiteWiseDATA。

创建数据集

1. 在https://console.aws.amazon.com/iotanalytics/，在左侧导航窗格，选择数据集.
2. 在存储库的创建数据集页面上，选择创建 SQL.
3. 在存储库的指定数据集页面上，指定数据集的详细信息。
   a. 输入数据集的名称。
   b. 适用于数据存储源，选择识别您的唯一 IDAmazon IoT SiteWise数据存储。
   c. （可选）对于标签，将一个或多个自定义标签（键值对）添加到数据集中。
4. 使用 SQL 表达式查询数据并回答分析问题。
   a. 在作者查询字段中，输入使用通配符最多显示五行数据的 SQL 查询。

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

有关支持的 SQL 功能的更多信息Amazon IoT Analytics，请参阅中的 SQL 表达式Amazon IoT Analytics (p. 86). 或者，请参阅教程：查询Amazon IoT SiteWise中的数据Amazon IoT Analytics (p. 31)以了解可以深入了解数据的统计查询示例。

b. 您可以选择测试查询以验证您的输入是否正确，并在查询后的表中显示结果。

Note

由于Amazon Athena 限制正在运行的查询的最大数量，您应将 SQL 查询限制为合理的大小，以便它不会在很长时间内运行。

5. （可选）使用指定时间范围内的数据创建数据集内容时，某些数据可能无法及时到达以进行处理。要允许延迟，您可以指定偏移量或增量。有关更多信息，请参阅 通过亚马逊获取延迟数据通知 CloudWatch 事件 (p. 108)。

在上配置数据选择筛选器后配置数据选择过滤器页面上，选择下一步.

6. （可选）在设置查询计划页，您可以安排此查询定期运行以刷新数据集。可以随时创建和编辑数据集计划。

Note

来自的数据Amazon IoT SiteWise摄取Amazon IoT Analytics每六个小时。我们建议选择六个小时或更长时间的频率。

选择和选择Frequency然后选择下一步.

7. Amazon IoT Analytics将创建此数据集内容的版本并存储指定时间段内的分析结果。我们建议 90 天，但是您可以选择设置自定义保留策略。您还可以限制数据集内容的存储版本的数量。

在配置数据集的结果页面上，选择下一步.

8. （可选）您可以将数据集结果的传输规则配置到特定目标，例如Amazon IoT Events.

在配置数据集内容交付规则页面上，选择下一步.

9. 检查您的选择，然后选择创建数据集.
10. 验证您的新数据集是否显示在数据集页.

使用创建数据集Amazon IoT SiteWiseDATA (数据Amazon CLI)

运行以下命令Amazon CLI命令来开始查询Amazon IoT SiteWiseDATA。

此处显示的示例使用Amazon Command Line Interface(Amazon CLI)。有关Amazon CLI，请参阅Amazon Command Line Interface用户指南. 有关可用于的 CLI 命令的更多信息，请参阅Amazon IoT Analytics，请参阅iotAnalytics中的Amazon Command Line Interface参考.

创建数据集

1. 运行以下命令`create-dataset`命令来创建数据集。

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

在哪里`my_dataset.json`文件包含以下内容。

```
{
    "datasetName": "my_dataset",
    "actions": [
```

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

有关支持的 SQL 功能的更多信息Amazon IoT Analytics，请参阅中的 SQL 表达式Amazon IoT Analytics (p. 86). 或者，请参阅教程：查询Amazon IoT SiteWise中的数据Amazon IoT Analytics (p. 31)以了解可以深入了解数据的统计查询示例。

2. 运行以下命令`create-dataset-content`命令可通过运行查询来创建数据集内容。

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

访问数据集内容

SQL 查询的结果是以 CSV 文件格式存储的数据集内容。将通过 Amazon S3 向您提供该文件。请参阅以下步骤，如何检查结果是否已准备就绪并下载文件。

主题

- 在中访问数据集内容Amazon IoT Analytics(控制台) (p. 30)
- 在中访问数据集内容Amazon IoT Analytics(Amazon CLI) (p. 30)

在中访问数据集内容Amazon IoT Analytics(控制台)

如果您的数据集包含任何数据，则可以在Amazon IoT Analytics控制台。

要访问您的Amazon IoT Analytics数据集结果

1. 在控制台中，数据集页面上，选择要访问的数据集的名称。
2. 在数据集摘要页面上，选择内容选项卡。
3. 在数据集内容表中，选择要预览结果的查询的名称或下载结果的 csv 文件。

在中访问数据集内容Amazon IoT Analytics(Amazon CLI)

如果您的数据集包含任何数据，则可以预览并下载 SQL 查询结果。

此处显示的示例使用Amazon Command Line Interface(Amazon CLI)。有关Amazon CLI，请参阅Amazon Command Line Interface用户指南. 有关可用于的 CLI 命令的更多信息，请参阅Amazon IoT Analytics，请参阅iotAnalytics中的Amazon Command Line Interface参考.

要访问您的Amazon IoT Analytics数据集结果 (Amazon CLI)

1. 运行以下命令`get-dataset-content`命令来查看查询的结果。

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

2. 如果您的数据集包含任何数据，则的输出将`get-dataset-content`，有`"state": "SUCCEEDED"`中的`status`字段，如下例所示。

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

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

3. 输出的get-dataset-content包括dataURI，是输出结果的签名 URL。它在较短的一段时间内有效(几个小时)。访问dataURI用于访问 SQL 查询结果的 URL。

   Note

   根据您的工作流，您可能需要在访问内容之前始终调用 get-dataset-content，因为调用此命令会生成新的签名 URL。

教程：查询Amazon IoT SiteWise中的数据Amazon IoT Analytics

本教程演示如何查询Amazon IoT SiteWise中的数据Amazon IoT Analytics. 本教程使用了中演示中的数据Amazon IoT SiteWise提供风电场示例数据集。

Important

此演示创建和使用的资源是收费的。

主题

先决条件

在此教程中，您需要以下资源：

- 您必须具有Amazon开始使用的帐户Amazon IoT SiteWise和Amazon IoT Analytics. 如果您没有账户，请按照中的过程操作创建Amazon帐户 (p. 5).
- 运行 Windows、macOS、Linux 或 Unix 的开发计算机（用于访问 Amazon Web Services Management Console）。有关更多信息，请参阅 Amazon Web Services Management Console 入门。
- Amazon IoT SiteWise定义的数据Amazon IoT SiteWise模型和资产并流式传输表示风电场设备数据的数据。要创建数据，请按照中的步骤操作创建Amazon IoT SiteWise演示中的Amazon IoT SiteWise用户指南.
- 您的Amazon IoT SiteWise在您管理的现有数据存储中演示风电场设备数据。有关如何为您创建数据存储的更多信息Amazon IoT SiteWiseDATA，请参阅配置存储设置中的Amazon IoT SiteWise用户指南.

Note

您的Amazon IoT SiteWise元数据显示在您的Amazon IoT SiteWise数据存储在创建后不久；但是，您的原始数据最多可能需要六个小时才能显示。同时，您可以创建Amazon IoT Analytics数据集并对元数据运行查询。

下一步

加载和验证数据 (p. 32)

加载和验证数据

您在本教程中查询的数据是一组示例Amazon IoT SiteWise对风电场中风力发动机涡轮机进行模型的数据。

Note

在本教程中，您将在数据存储中查询三个表：

- `raw`-包含每个资产的未处理的原始数据。
- `asset_metadata`-包含有关每种资产的一般信息。
- `asset_hierarchy_metadata`-包含有关资产之间关系的信息。

运行本教程中的 SQL 查询

1. 按中的步骤操作。使用创建数据集Amazon IoT SiteWise数据（控制台）(p. 28)要么使用创建数据集Amazon IoT SiteWiseDATA (数据Amazon CLI）(p. 29)创建Amazon IoT Analytics您的数据集Amazon IoT SiteWiseDATA。
2. 要在本教程中更新您的数据集查询，请执行以下操作。

   a. 在Amazon IoT Analytics控制台，在数据集页面上，选择您在上一页面上创建的数据集的名称。
   b. 在数据集摘要页面上，选择编辑以编辑 SQL 查询。
   c. 要在查询之后的表格中显示结果，请选择测试查询.

   或者，您也可以运行以下命令`update-dataset`用于修改 SQL 查询的命令Amazon CLI.

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

   `update-query.json` 的内容：

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

3. 在Amazon IoT Analytics控制台或使用Amazon CLI对您的数据运行以下查询以验证您的`asset_metadata`表已成功加载。

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

同样地，您可以验证您的`asset_hierarchy_metadata`和`raw`表不为空。

下一个步骤

数据探究 (p. 33)

数据探究

在您的Amazon IoT SiteWise数据已创建并加载到数据存储中，您可以创建Amazon IoT Analytics数据集并在中运行 SQL 查询Amazon IoT Analytics以了解有关资产的见解。以下查询演示了如何在运行统计查询之前探索数据。

使用 SQL 查询探索数据

1. 查看每个表中的列和值的示例，例如在原始表中。

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

2. 使用`SELECT DISTINCT`查询您的`asset_metadata`表格并列出您的（唯一）名称Amazon IoT SiteWise资产。

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

3. 列出有关特定属性的信息Amazon IoT SiteWise资产，请使用`WHERE`子句。

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

4. 与Amazon IoT Analytics，您可以联接数据存储中两个或多个表中的数据，例如以下示例中的数据。

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

要查看资产之间的所有关系，请使用`JOIN`以下查询中的功能。

```
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

运行统计查询 (p. 34)

运行统计查询

现在你已经探索了你的Amazon IoT SiteWise数据，您可以运行统计查询，为工业设备提供宝贵的见解。以下查询演示了您可以检索的一些信息。

要在上运行统计查询Amazon IoT SiteWise演示风电场数据

1. 运行以下 SQL 命令以查找所有属性的最新值，其中包含特定资产的数值（演示涡轮资产 4）。

```
SELECT assetName,
    assetPropertyName,
    assetPropertyUnit,
    max_by(value, timeInSeconds) AS Latest
FROM (
    SELECT *,
        CASE assetPropertyDataType
        WHEN 'DOUBLE' THEN
        cast(doubleValue AS varchar)
        WHEN 'INTEGER' THEN
        cast(integerValue AS varchar)
        WHEN 'STRING' THEN
        stringValue
        WHEN 'BOOLEAN' THEN
        cast(booleanValue AS varchar)
        ELSE NULL
        END AS value
    FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
    JOIN my_iotsitewise_datastore.raw AS raw
        ON raw.seriesId = asset_metadata.timeSeriesId
    WHERE startYear=2021
        AND startMonth=7
        AND startDay=8
        AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. 加入元数据表和原始表格，以确定所有资产的最大风速属性，此外他们的父资产.

```
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. 要查找资产（演示涡轮机资产 2）的特定属性（风速）的平均值，请运行以下 SQL 命令。你必须更换my_bucket_id使用您的存储桶的 ID。

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

下一步

清理教程资源 (p. 35)

清理教程资源

完成本教程后，清理资源以避免产生费用。

删除Amazon IoT SiteWise演示

这些区域有：Amazon IoT SiteWise演示会在一周后自行删除。如果您已完成演示资源的使用，则可以提前删除演示。要手动删除演示，请使用以下步骤。

1. 导航到 Amazon CloudFormation 控制台。
2. 选择IoTSiteWiseDemoAssets从列表中堆栈.
3. 请选择 Delete（删除）。当您删除堆栈时，为演示创建的所有资源都将被删除。
4. 在确认对话框中，输入Delete.

   删除堆栈约需 15 分钟时间。如果演示无法删除，请再次选择右上角的 Delete (删除)。如果演示仍然无法删除，请按照 Amazon CloudFormation 控制台中的步骤跳过删除失败的资源，然后重试。

删除数据存储

- 要删除托管数据存储，请运行 CLI 命令delete-datastore，例如在以下示例中。

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

删除Amazon IoT Analytics数据集

- 要删除数据集，请运行 CLI 命令delete-dataset，例如在以下示例中。在执行此操作之前，您无需删除数据集的内容。

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

> **Note**
>
> 此命令不会生成任何输出。

Pipeline Activity

最简单的正常运行的管道可将一个通道连接到数据存储，这使其成为具有两个活动的管道：一个 channel 活动和一个 datastore 活动。您可以通过向管道添加额外的活动，实现更强大的消息处理功能。

您可以使用RunPipelineActivity操作，模拟在您提供的消息负载上运行管道活动的结果。在开发和调试管道活动时，您会发现这非常有用。RunPipelineActivity 示例 (p. 55)演示了如何将其用法。

频道活动

管道中的第一个活动必须是channel活动，它确定要处理的消息的来源。

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

数据存储活动

datastore 活动是最后一个活动，指定将处理后的数据存储到何处。

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

Amazon Lambda活动

您可以使用lambda活动，以对消息执行更复杂的处理。例如，您可以使用来自外部 API 操作输出的数据来丰富消息，或根据 Amazon DynamoDB 的逻辑筛选消息。不过，您不能使用此活动来执行任何类型的基于消息的处理，包括筛选哪些消息将存储在数据存储中。

这些区域有：Amazon Lambda此活动中使用的函数必须接收并返回一组 JSON 对象。在以下示例中，Lambda 函数修改并返回它的event参数。

```
{
    "lambda": {
        "name": "MyLambdaActivity",
        "lambdaName": "mylambda",
        "batchSize": 10,
        "next": "MyDatastoreActivity"
```

```
    }
}
```

Note

batchSize 确定 Lambda 函数在每次调用时接收多少消息。设置时，请注意 Lambda 函数的最大超时为 900 秒。因此，Lambda 函数必须能够在不到 900 秒的时间内处理批处理中的所有消息。

你必须添加策略才能授予Amazon IoT Analytics调用 Lambda 函数的权限。运行以下 CLI 命令并替换：*###*使用 Lambda 函数的名称，*123456789012*与您的Amazon账户 ID，然后使用将调用给定 Lambda 函数的管道的 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
```

此命令将返回以下：

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

有关更多信息，请参阅 。使用 的基于资源的策略Amazon Lambda中的Amazon Lambda开发人员指南.

Lambda 函数示例 1

在此示例中，Lambda 函数根据原始消息中的数据添加信息。设备发布一条消息，其负载类似于以下示例。

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

设备的管道定义如下。

```
{
    "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"
    }
}
```

该 f遵循 Lambda Python 函数 (`MyAnalyticsLambdaFunction`) 向消息中添加 GMaps URL 和华氏温度。

```
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 函数示例 2

一种有用的方法是压缩并序列化消息负载，以降低传输和存储成本。在该第二个示例中，Lambda 函数假定消息负载表示已压缩并以字符串形式进行 Base64 编码（序列化）的 JSON 原始数据。它返回原始 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 活动

addAttributes 活动根据消息中现有的属性添加属性。这样，您就可以在存储之前更改消息的形状。例如，您可以使用 addAttributes 规范化来自不同代的设备固件的数据。

请考虑以下输入消息。

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

这些区域有：addAttributes活动看起来与以下内容类似。

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

```
}
```

此活动会将设备 ID 移到根级别，并提取coordarray，将它们提升为顶级属性lat和lon. 作为此活动的结果，输入消息将转换为以下示例。

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

原始设备属性仍然存在。如果要删除它，您可以使用 removeAttributes 活动。

“移除属性” 活动

removeAttributes 活动从消息中删除属性。例如，给定消息是由addAttributes活动。

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

要规范化该消息，使其只包含根级别所需的数据，请使用以下命令：removeAttributes活动。

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

这会产生沿管道流动的以下消息。

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

Select Attributes 活动

selectAttributes 活动仅使用原始消息中的指定属性创建新消息。其他属性都将被丢弃。selectAttributes 只会在消息的根位置下创建新属性。因此，给定此消息：

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

和此活动：

```
{
    "selectAttributes": {
        "name": "MySelectAttributesActivity",
        "attributes": [
            "device.temp",
            "device.hum",
            "light"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

结果将是流经管道的以下消息。

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

同样，`selectAttributes` 只能创建根级别对象。

筛选活动

`filter` 活动根据消息属性筛选消息。此活动中使用的表达式看起来像一个 SQL。`WHERE`子句，它必须返回一个布尔值。

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

DeviceRegistryEnrich 活动

这些区域有：`deviceRegistryEnrich`活动使您能够从Amazon IoT设备注册表到您的消息负载。例如，给定了以下 消息：

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

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

和类似下面这样的 deviceRegistryEnrich 活动：

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

输出消息现在如下所示。

```
{
    "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"
    }
}
```

您必须在活动定义的 roleArn 字段中指定已附加适当权限的角色。该角色必须具有类似于以下示例的权限策略。

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

和类似如下的信任策略：

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

```
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

DeviceShadowEnrich 活动

一个deviceShadowEnrich活动添加来自Amazon IoT将 Device Shadow 服务转换为消息。例如，给定以下消息：

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

和以下 deviceShadowEnrich 活动：

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

结果将是一条类似于以下示例的消息。

```
{
    "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
    }
}
```

您必须在活动定义的 `roleArn` 字段中指定已附加适当权限的角色。该角色必须具有类似如下的权限策略。

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

和类似如下的信任策略：

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

数学活动

`math` 活动使用消息的属性计算算术表达式。表达式必须返回数字。例如，给定以下输入消息：

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

经过以下 `math` 活动处理后：

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

生成的消息类似于：

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

数学活动运算符和函数

您可以在 `math` 活动中使用以下运算符：

+	加
-	减
*	乘
/	除
%	取模

您可以在 `math` 活动中使用以下函数：

abs(Decimal)

返回数字的绝对值。

示例：`abs(-5)`返回 5.

参数类型	结果
`Int`	`Int`，参数的绝对值。
`Decimal`	`Decimal`，参数的绝对值
`Boolean`	`Undefined.`
`String`	`Decimal`。结果是参数的绝对值。如果字符串无法转换，则结果为 `Undefined`。
数组	`Undefined.`
对象	`Undefined.`
Null	`Undefined.`
未定义	`Undefined.`

acos(Decimal)

以弧度为单位返回数字的反余弦值。在代入函数之前，`Decimal` 参数舍入到双精度。

示例：`acos(0)`= 1.5707963267948966

参数类型	结果
`Int`	`Decimal` (双精度)，参数的反余弦值。虚数结果返回 `Undefined`。
`Decimal`	`Decimal` (双精度)，参数的反余弦值。虚数结果返回 `Undefined`。
`Boolean`	`Undefined.`
`String`	`Decimal`(双精度) 参数的逆余弦值。如果字符串无法转换，则结果为 `Undefined`。虚数结果返回 `Undefined`。
数组	`Undefined.`
对象	`Undefined.`
Null	`Undefined.`

参数类型	结果
未定义	Undefined.

asin(Decimal)

以弧度为单位返回数字的反正弦值。在代入函数之前，Decimal 参数舍入到双精度。

示例：asin(0)= 0.0

参数类型	结果
Int	Decimal (双精度)，参数的反正弦值。虚数结果返回 Undefined。
Decimal	Decimal (双精度)，参数的反正弦值。虚数结果返回 Undefined。
Boolean	Undefined.
String	Decimal (双精度)，参数的反正弦值。如果字符串无法转换，则结果为 Undefined。虚数结果返回 Undefined。
数组	Undefined.
对象	Undefined.
Null	Undefined.
未定义	Undefined.

atan(Decimal)

以弧度为单位返回数字的反正切值。在代入函数之前，Decimal 参数舍入到双精度。

示例：atan(0)= 0.0

参数类型	结果
Int	Decimal (双精度)，参数的反正切值。虚数结果返回 Undefined。
Decimal	Decimal (双精度)，参数的反正切值。虚数结果返回 Undefined。
Boolean	Undefined.
String	Decimal (双精度)，参数的反正切值。如果字符串无法转换，则结果为 Undefined。虚数结果返回 Undefined。
数组	Undefined.
对象	Undefined.

参数类型	结果
Null	Undefined.
未定义	Undefined.

atan2(Decimal, Decimal)

以弧度的形式返回 x 轴正方向与由两个参数定义的 (x,y) 点之间的角度。逆时针的角，角度为正数（上半平面，y > 0），顺时针的角，角度为负数（ ）。Decimal在代入函数之前，参数舍入到双精度。

示例：atan(1, 0)= 1.5707963267948966

参数类型	参数类型	结果
Int / Decimal	Int / Decimal	Decimal(双精度)，x 轴和指定的 (x, y) 点之间的角度。
Int / Decimal / String	Int / Decimal / String	Decimal，所描述点的反正切值。如果字符串无法转换，则结果为 Undefined。
其他值	其他值	Undefined.

ceil(Decimal)

将给定的 Decimal 向上舍入到最近的 Int。

示例：

ceil(1.2) = 2

ceil(11.2) = -1

参数类型	结果
Int	Int，参数值。
Decimal	Int，该字符串被转换为Decimal并向上舍入到最近的Int. 如果字符串无法转换为 Decimal，则结果为 Undefined。
其他值	Undefined.

cos(Decimal)

以弧度为单位返回数字的余弦值。在代入函数之前，Decimal 参数舍入到双精度。

示例：cos(0)= 1

参数类型	结果
Int	Decimal (双精度)，参数的余弦值。虚数结果返回 Undefined。

参数类型	结果
`Decimal`	`Decimal` (双精度)，参数的余弦值。虚数结果返回 `Undefined`。
`Boolean`	`Undefined`.
`String`	`Decimal` (双精度)，参数的余弦值。如果字符串无法转换为 `Decimal`，则结果为 `Undefined`。虚数结果返回 `Undefined`。
数组	`Undefined`.
对象	`Undefined`.
Null	`Undefined`.
未定义	`Undefined`.

cosh(Decimal)

以弧度为单位返回数字的双曲余弦值。在代入函数之前，`Decimal` 参数舍入到双精度。

示例：`cosh(2.3)`= 5.037220649268761

参数类型	结果
`Int`	`Decimal` (双精度)，参数的双曲余弦值。虚数结果返回 `Undefined`。
`Decimal`	`Decimal` (双精度)，参数的双曲余弦值。虚数结果返回 `Undefined`。
`Boolean`	`Undefined`.
`String`	`Decimal` (双精度)，参数的双曲余弦值。如果字符串无法转换为 `Decimal`，则结果为 `Undefined`。虚数结果返回 `Undefined`。
数组	`Undefined`.
对象	`Undefined`.
Null	`Undefined`.
未定义	`Undefined`.

exp(Decimal)

返回 e 提高到decimal 参数。`Decimal`在代入函数之前，参数舍入到双精度。

示例：`exp(1)`= 1

参数类型	结果
`Int`	`Decimal`(双精度)，^ 参数。

参数类型	结果
Decimal	Decimal(双精度)，^ 参数
String	Decimal(双精度)，^ 参数。如果String无法转换为Decimal，如果结果如果Undefined.
其他值	Undefined.

ln(Decimal)

返回参数的自然对数。在代入函数之前，Decimal 参数舍入到双精度。

示例：ln(e)= 1

参数类型	结果
Int	Decimal (双精度)，参数的自然对数。
Decimal	Decimal(双精度)，参数的自然对数
Boolean	Undefined.
String	Decimal (双精度)，参数的自然对数。如果字符串无法转换为 Decimal，则结果为 Undefined。
数组	Undefined.
对象	Undefined.
Null	Undefined.
未定义	Undefined.

log(Decimal)

返回参数的以 10 为底的对数。在代入函数之前，Decimal 参数舍入到双精度。

示例：log(100)= 2.0

参数类型	结果
Int	Decimal (双精度)，参数以 10 为底的对数。
Decimal	Decimal (双精度)，参数以 10 为底的对数。
Boolean	Undefined.
String	Decimal (双精度)，参数以 10 为底的对数。 如果 String 无法转换为 Decimal，则结果为 Undefined。
数组	Undefined.
对象	Undefined.
Null	Undefined.

参数类型	结果
未定义	Undefined.

mod(Decimal, Decimal)

返回第二个参数的除法的余数。您还可以使用%作为相同模功能的中缀运算符。

示例：`mod(8, 3)`= 3

左侧操作数	右侧操作数	输出
Int	Int	Int，第二个参数的第一个参数对第一个参数取模。
Int / Decimal	Int / Decimal	Decimal，第二个参数的第一个参数对第一个参数取模。
String / Int / Decimal	String / Int / Decimal	如果所有字符串都转换为Decimals，如果第一个参数对第二个参数取模，则结果将是结果。否则为 Undefined。
其他值	其他值	Undefined.

power(Decimal, Decimal)

返回第一个参数的第二个参数次幂的值。在代入函数之前，Decimal 参数舍入到双精度。

示例：`power(2, 5)`= 32.0

参数类型 1	参数类型 2	输出
Int / Decimal	Int / Decimal	Decimal (双精度)，返回第一个参数的第二个参数次幂的值。
Int / Decimal / String	Int / Decimal / String	Decimal (双精度)，返回第一个参数的第二个参数次幂的值。所有字符串均转换为Decimals. 如果任何 String 无法转换为 Decimal，则结果为 Undefined。
其他值	其他值	Undefined.

round(Decimal)

将给定的 Decimal 舍入到最近的 Int。如果 Decimal 与上下两个 Int 值距离相同 (例如 0.5)，Decimal 将向上进位。

示例：

`Round(1.2)` = 1

`Round(1.5)` = 2

`Round(1.7)` = 2

`Round(-1.1)` = -1

`Round(-1.5)` = -2

参数类型	结果
`Int`	参数
`Decimal`	`Decimal` 会向下舍入至最近的 `Int`。
`String`	`Decimal` 会向下舍入至最近的 `Int`。如果字符串无法转换为 `Decimal`，则结果为 `Undefined`。
其他值	`Undefined`.

sign(Decimal)

返回给定数字的符号。当参数的符号为正时，将返回 1。当参数的符号为负时，将返回 -1。如果参数为 0，则返回 0。

示例：

`sign(-7)` = -1

`sign(0)`= 0

`sign(13)` = 1

参数类型	结果
`Int`	`Int`，`Int` 值的符号。
`Decimal`	`Int`，`Decimal` 值的符号。
`String`	`Int`，`Decimal` 值的符号。如果将字符串转换为`Decimal`，以及符号`Decimal`将返回值。如果 `String` 无法转换为 `Decimal`，则结果为 `Undefined`。
其他值	`Undefined`.

sin(Decimal)

以弧度为单位返回数字的正弦值。在代入函数之前，`Decimal` 参数舍入到双精度。

示例：`sin(0)`= 0.0

参数类型	结果
`Int`	`Decimal` (双精度)，参数的正弦值。

参数类型	结果
Decimal	Decimal (双精度)，参数的正弦值。
Boolean	Undefined.
String	Decimal，参数的正弦值。如果字符串无法转换为 Decimal，则结果为 Undefined。
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

sinh(Decimal)

以弧度为单位返回数字的双曲正弦值。在代入函数之前，Decimal 值舍入到双精度。结果是双精度的 Decimal 值。

示例：sinh(2.3)= 4.936961805545957

参数类型	结果
Int	Decimal (双精度)，参数的双曲正弦值。
Decimal	Decimal (双精度)，参数的双曲正弦值。
Boolean	Undefined.
String	Decimal，参数的双曲正弦值。如果字符串无法转换为 Decimal，则结果为 Undefined。
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

sqrt(Decimal)

返回数字的平方根。在代入函数之前，Decimal 参数舍入到双精度。

示例：sqrt(9)= 3.0

参数类型	结果
Int	参数的平方根。
Decimal	参数的平方根。
Boolean	Undefined.

参数类型	结果
String	参数的平方根。如果字符串无法转换为 Decimal，则结果为 Undefined。
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

tan(Decimal)

以弧度为单位返回数字的正切值。在代入函数之前，Decimal 值舍入到双精度。

示例：tan(3)= -0.1425465430742778

参数类型	结果
Int	Decimal (双精度)，参数的正切值。
Decimal	Decimal (双精度)，参数的正切值。
Boolean	Undefined.
String	Decimal (双精度)，参数的正切值。如果字符串无法转换为 Decimal，则结果为 Undefined。
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

tanh(Decimal)

以弧度为单位返回数字的双曲正切值。在代入函数之前，Decimal 值舍入到双精度。

示例：tanh(2.3)= 0.9800963962661914

参数类型	结果
Int	Decimal (双精度)，参数的双曲正切值。
Decimal	Decimal (双精度)，参数的双曲正切值。
Boolean	Undefined.
String	Decimal (双精度)，参数的双曲正切值。如果字符串无法转换为 Decimal，则结果为 Undefined。
Array	Undefined.

参数类型	结果
`Object`	`Undefined.`
`Null`	`Undefined.`
`Undefined`	`Undefined.`

trunc (Decimal, 整数)

按照第二个参数指定的 `Decimal` 位数截断第一个参数。如果第二个参数小于零，则会设置为零。如果参数第二大于 34，则会设置为 34。结果中删除结尾的零。

示例：

`trunc(2.3, 0)` = 2

`trunc(2.3123, 2)`= 2.31

`trunc(2.888, 2)`= 2.88

`trunc(2.00, 5)` = 2

参数类型 1	参数类型 2	结果
`Int`	`Int`	源值。
`Int` / `Decimal` / `String`	`Int` / `Decimal`	第一个参数被截断到由第二个参数所指定的长度。第二个参数如果不是 `Int`，将向下舍入至最近的 `Int`。所有字符串均转换为`Decimal`值。如果字符串转换失败，则结果为 `Undefined`。
其他值		Undefined。

RunPipelineActivity

下面是一个示例说明如何将`RunPipelineActivity`命令来测试管道活动。在这个例子中，我们测试一个数学活动。

1. 创建`maths.json`文件，其中包含您要测试的管道活动的定义。

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

2. 创建文件`payloads.json`文件，其中包含用于测试管道活动的示例负载。

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

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

3. 调用RunPipelineActivities从命令行执行操作。

```
aws iotanalytics run-pipeline-activity  --pipeline-activity file://maths.json  --payloads
 file://payloads.json
```

这会产生以下结果。

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

结果中列出的负载为 Base64 编码的字符串。解码这些字符串时，您将获得以下结果。

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

重新处理通道消息

Amazon IoT Analytics使您能够重新处理频道数据。这在以下情况下会很有用：

- 您希望重放现有已提取的数据，而不是重新开始。
- 您想要更新管道并希望带来现有数据 up-to-date 随着更改。
- 您希望包括在更改客户托管存储选项、渠道权限或数据存储之前提取的数据。

参数

当你通过渠道重新处理频道消息时Amazon IoT Analytics，您必须指定以下信息：

`StartPipelineReprocessing`

通过管道开始重新处理通道消息。

`ChannelMessages`

指定要重新处理的一组或多组频道消息。

如果您将`channelMessages`对象，您不得为指定值`startTime`和`endTime`.

`s3Paths`

指定一个或多个密钥，以标识用于保存您的通道消息的 Amazon Simple Storage Service (Amazon S3) 对象。您必须使用密钥的完整路径。

示例路径：`00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.json.gz`

类型: 字符串数组

数组成员约束：1-100 个项目。

长度限制：1-1024 个字符。

`endTime`

重新处理的通道数据的结束时间（不包括）。

如果为指定值`endTime`参数，则不能使用`channelMessages`对象。

类型: 时间戳

`startTime`

重新处理的原始消息数据的开始时间 (含)。

如果为指定值`startTime`参数，则不能使用`channelMessages`对象。

类型: 时间戳

`pipelineName`

要开始重新处理的管道的名称。

类型: 字符串

长度约束：1-128 个字符。

重新处理通道消息（控制台）

本教程向您展示如何重新处理存储在指定 Amazon S3 对象中的频道数据Amazon IoT Analytics控制台。

在开始之前，请确保要重新处理的渠道消息保存在客户托管的 Amazon S3 存储桶中。

1. 登录到 Amazon IoT Analytics 控制台。
2. 在导航窗格中，选择管道.
3. 选择目标管道。
4. 选择重新处理消息从操作.
5. 在存储库的管道重新处理页面，选择S3 对象为了重新处理消息.

   这些区域有：Amazon IoT Analytics控制台还提供以下选项：

   - 所有可用的范围-重新处理频道中的所有有效数据。
   - Last 120 Days-重新处理过去 120 天内到达的数据。
   - 最近 90 天-重新处理最近 90 天内到达的数据。
   - 最近 30 天-重新处理最近 30 天内到达的数据。
   - 自定义范围-重新处理在指定时间范围内到达的数据。你可以选择任何时间范围。
6. 输入存储频道消息的 Amazon S3 对象的密钥。

   要查找密钥，请执行以下操作：

   a. 转至Amazon S3 控制台.
   b. 选择目标 Amazon S3 对象。
   c. UNDER属性，在对象概述部分中，复制密钥。
7. 选择开始再处理.

重新处理频道消息 (API)

当您使用`StartPipelineReprocessing`API，请注意以下几点：

- 这些区域有：`startTime`和`endTime`参数指定提取原始数据的时间，不过这些是粗略估计的时间。您可以舍入到最接近的小时。这些区域有：`startTime`是包容性的，但`endTime`是独家的。
- 该命令异步启动重新处理并立即返回。
- 不保证重新处理消息的顺序与接收时的顺序相同。这大致相同，但不完全一致。
- 你最多可以弥补 1000`StartPipelineReprocessing`API 请求每 24 小时通过管道重新处理相同的频道消息。
- 重新处理原始数据会产生额外的成本。

有关更多信息，请参阅 。StartPipelineReprocessingAPI，在Amazon IoT AnalyticsAPI 参考.

取消渠道再处理活动

要取消管道再处理活动，请使用CancelPipelineReprocessingAPI 或者选择取消重新处理在活动中的页面Amazon IoT Analytics控制台。如果取消再处理，剩余的数据将不会被重新处理。你必须开始另一个再处理请求。

使用DescribePipeline查看重新处理的状态的 API。请参阅`reprocessingSummaries`响应中的字段。

自动化您的工作流程

Amazon IoT Analytics为以下内容提供高级数据分析Amazon IoT. 您可以使用数据分析和机器学习工具自动收集 IoT 数据，处理数据以及存储和分析数据。您可以执行托管自己的自定义分析代码或 Jupyter Notebook 的容器，也可以使用第三方自定义代码容器，这样就不必重新创建现有的分析工具。您可以使用以下功能从数据存储中提取输入数据，并将其提供给自动化工作流程：

按照重复计划创建数据集内容

通过在调用时指定触发器来计划自动创建数据集内容`CreateDataset`(`triggers:schedule:expression`)。数据存储中的数据用于创建数据集内容。您可以使用 SQL 查询选择所需的字段 (`actions:queryAction:sqlQuery`)。

定义一个不重叠的连续时间间隔，以确保新的数据集内容仅包含自上次以来到达的数据。使用`actions:queryAction:filters:deltaTime`和`:offsetSeconds`用于指定增量时间间隔的字段。然后指定一个触发器，在时间间隔过后创建数据集内容。请参阅 the section called “示例 6 — 使用增量窗口 (CLI) 创建 SQL 数据集” (p. 67)。

完成另一个数据集后创建数据集内容

当另一个数据集的内容创建完成时，触发新数据集内容的创建`triggers:dataset:name`.

自动运行您的分析应用程序

将您自己的自定义数据分析应用程序容器化，并在创建其他数据集的内容时触发它们运行。这样，您可以为应用程序提供来自定期创建的数据集内容的数据。您可以在应用程序中自动对分析结果采取行动。(`actions:containerAction`)

完成另一个数据集后创建数据集内容

当另一个数据集的内容创建完成时，触发新数据集内容的创建`triggers:dataset:name`.

自动运行您的分析应用程序

将您自己的自定义数据分析应用程序容器化，并在创建其他数据集的内容时触发它们运行。这样，您可以为应用程序提供来自定期创建的数据集内容的数据。您可以在应用程序中自动对分析结果采取行动。(`actions:containerAction`)

使用案例

自动测量产品质量以降低产品质量 OpEx

您的系统具有可测量压力、湿度和温度的智能值。系统会定期整理事件，也会在某些事件发生时（例如值的打开和关闭）进行整理。与Amazon IoT Analytics，您可以自动进行分析，汇总来自这些定期窗口的非重叠数据，并创建有关最终产品质量的 KPI 报告。处理完每个批次后，您可以衡量整体产品质量，并通过最大化运行量来降低运营支出。

自动分析设备队列

您每 15 分钟对数百台设备生成的数据运行一次分析（算法、数据科学或 KPI 的 ML）。每个分析周期都会生成和存储状态以供下一次分析运行。对于每个分析，您都希望只使用在指定时间窗口内收到的数据。与Amazon IoT Analytics您可以协调分析，为每次运行创建 KPI 和报告，然后存储数据以供future分析。

自动执行异常检测

Amazon IoT Analytics使您能够自动执行异常检测工作流程，您必须每 15 分钟对到达数据存储的新数据手动运行该工作流程。您还可以自动更新控制面板，其中显示在指定时间段内的设备使用情况和顶级用户。

预测工业过程的成果

您具有工业生产线。使用发送到的数据Amazon IoT Analytics，包括可用的过程测量，您可以运用分析工作流程来预测过程结果。模型的数据可以排列在 M x N 矩阵中，其中每行包含来自实验室样本采集的不同时间点的数据。Amazon IoT Analytics通过创建增量窗口和使用数据科学工具创建 KPI 和保存测量设备的状态，帮助您操作分析工作流程。

使用 Docker 容器

本节包括有关如何构建自己的 Docker 容器的信息。如果重复使用第三方构建的 Docker 容器，则存在安全风险：这些容器可能使用您的用户权限执行任意代码。在使用之前，请确保您信任任何第三方容器的作者。

使用以下步骤可设置针对自上次执行分析以来到达的数据的定期数据分析：

1. 创建一个 Docker 容器，其中包含您的数据应用程序以及任何必要的库或其他依赖项。

   这些区域有： IotAnalytics Jupyter 扩展提供了容器化 API 来协助容器化过程。您还可以运行自己创建的图像，在其中创建或汇编应用程序工具集以执行所需的数据分析或计算。Amazon IoT Analytics允许您通过变量定义容器化应用程序的输入数据源和 Docker 容器输出数据的目的地。(自定义 Docker 容器输入/输出变量包含有关在自定义容器中使用变量的更多信息。)
2. 将容器上传到 Amazon ECR 注册表。
3. 创建数据存储库以接收和存储来自设备的消息（数据）（`iotanalytics: CreateDatastore`）
4. 创建发送消息的频道 (`iotanalytics: CreateChannel`)。
5. 创建管道以将通道连接到数据存储 (`iotanalytics: CreatePipeline`)。
6. 创建一个 IAM 角色，该角色授予将消息数据发送到Amazon IoT Analytics通道 (`iam: CreateRole`.)
7. 创建使用 SQL 查询将通道连接到消息数据源的 IoT 规则 (`iot: CreateTopicRule`领域`topicRulePayload:actions:iotAnalytics`)。当设备通过 MQTT 发送带有相应主题的消息时，该消息将被路由到您的频道。或者，您可以使用`iotanalytics: BatchPutMessage`将消息直接从能够使用的设备发送到频道Amazon软件开发工具开发工具Amazon CLI.
8. 创建一个由时间表触发创建的 SQL 数据集 (`iotanalytics: CreateDataset,`领域`actions: queryAction:sqlQuery`)。

   您还可以指定要应用于消息数据的预筛选器，以帮助将消息限制为自上次执行操作以来到达的消息。（字段`actions:queryAction:filters:deltaTime:timeExpression`给出了用于确定消息时间的表达式。while field`actions:queryAction:filters:deltaTime:offsetSeconds`指定消息到达时可能出现的延迟。)

   预过滤器以及触发时间表决定了您的增量窗口。每个新的 SQL 数据集都是使用自上次创建 SQL 数据集以来收到的消息创建的。（第一次创建 SQL 数据集怎么样？ 根据计划和预过滤器估算上次创建数据集的时间。)
9. 创建另一个由创建第一个数据集触发的数据集 (CreateDataset领域`trigger:dataset`)。对于此数据集，您可以指定容器操作（已归档）`actions:containerAction`），它指向您在第一步中创建的 Docker 容器，并提供运行该容器所需的信息。在这里，您还可以指定：

   - 存储在您的账户中的 docker 容器的 ARN (`image`。)
   - 角色的 ARN，该角色用于授予访问所需资源所需的系统权限，以便运行容器操作 (`executionRoleArn`)。
   - 执行容器操作的资源的配置 (`resourceConfiguration`。)
   - 用于执行容器操作的计算资源的类型 (`computeType`使用可能的值：`ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]`)。
   - 用于执行容器操作的资源实例可以使用的持久性存储的大小 (GB)`volumeSizeInGB`)。
   - 在执行应用程序的上下文中使用的变量的值（基本上是传递给应用程序的参数）（`variables`)。

在执行容器时，将替换这些变量。这使您能够使用创建数据集内容时提供的不同变量（参数）运行同一个容器。这些区域有： IotAnalytics Jupyter 扩展通过自动识别笔记本中的变量并将它们作为容器化过程的一部分提供来简化此过程。您可以选择已识别的变量或添加自己的自定义变量。在运行容器之前，系统会将每个变量的值替换为执行时的当前值。

- 其中一个变量是使用其最新内容作为应用程序输入的数据集的名称（这是您在上一步中创建的数据集的名称）（`datasetContentVersionValue:datasetName`)。

使用 SQL 查询和增量窗口生成数据集，使用容器生成应用程序，Amazon IoT Analytics创建按您指定的间隔在 delta 窗口中的数据上运行的预定生产数据集，生成所需的输出并发送通知。

您可以暂停生产数据集应用程序，并在选择时将其恢复。当您恢复生产数据集应用程序时，Amazon IoT Analytics，默认情况下，它会捕获自上次执行以来到达但尚未分析的所有数据。您还可以通过执行一系列连续运行来配置恢复生产数据集的方式（作业窗口长度）。或者，您可以通过仅捕获新到达且符合增量窗口指定大小的数据来恢复生产数据集应用程序。

在创建或定义由创建另一个数据集触发的数据集时，请注意以下限制：

- SQL 数据集只能触发容器数据集。
- 一个 SQL 数据集最多可以触发 10 个容器数据集。

创建由 SQL 数据集触发的容器数据集时可能会返回以下错误：

- “Triggering dataset can only be added on a container dataset”(只能在容器数据集中添加触发数据集)
- “There can only be one triggering dataset”(只能存在一个触发数据集)

  如果您尝试定义由两个不同的 SQL 数据集触发的容器数据集，就会出现此错误。
- “触发数据集<dataset-name>无法由容器数据集触发”

  如果您尝试定义另一个由另一个容器数据集触发的容器数据集，则会出现此错误。
- “<N>数据集已经依赖于<dataset-name>数据集了。”

  如果您尝试定义另一个由已经触发 10 个容器数据集的 SQL 数据集触发的容器数据集，则会出现此错误。
- “Exactly one trigger type should be provided”(应确切提供一种触发器类型)

  当您尝试定义由计划触发器和数据集触发器触发的数据集时，就会出现此错误。

自定义 Docker 容器输入/输出变量

本节演示您的自定义 Docker 映像运行的程序如何读取输入变量并上传其输出。

参数文件

输入变量以及要将输出上传到的目的地存储在一个 JSON 文件中，它位于执行 Docker 映像的实例上的 `/opt/ml/input/data/iotanalytics/params` 中。以下是该文件内容的示例。

```
{
   "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"
    }
}
```

除了数据集的名称和版本 ID 外，`Variables` 部分还包含在 `iotanalytics:CreateDataset` 调用中指定的变量 - 在此示例中，为变量 `example_var` 给定值 `hello world!`。在 `custom_output` 变量中还提供了自定义输出 URI。该 `OutputUris` 字段包含容器可将其输出上传到的默认位置 - 在本示例中，同时提供了 ipynb 和 html 输出的默认输出 URI。

输入变量

通过 Docker 映像启动的程序可从 `params` 文件中读取变量。以下是一个示例程序，它可以打开`params`文件，对其进行解析，然后输出其值`example_var`变量。

```
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)
```

上传输出

您的 Docker 镜像启动的程序也可能将其输出存储在 Amazon S3 位置。必须使用" 加载输出bucket-owner-full-control"访问控制列表. 访问列表授予Amazon IoT Analytics对上传输出的服务控制。在此示例中，我们扩展了前一个示例以上传的内容`example_var`到由定义的 Amazon Storage S3 位置`custom_output`在里面`params`文件。

```
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)
```

权限

您必须创建两个 角色。一个角色授予启动权限 SageMaker 实例，以便对笔记本进行容器化。执行容器需要使用另一个角色。

您可以自动或手动创建第一个角色。如果你创建了新的 SageMaker使用实例Amazon IoT Analytics 控制台，你可以选择自动创建一个新角色，该角色授予执行所需的所有权限 SageMaker 实例和容器化笔记本。或者，您也可以手动创建具有这些权限的角色。为此，请使用以下命令创建一个角色`AmazonSageMakerFullAccess`附加策略并添加以下策略。

```
{
  "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/*"
    }
  ]
}
```

您必须手动创建第二个角色，该角色将授予执行容器所需的权限。即使你使用了Amazon IoT Analytics控制台自动创建第一个角色。创建附加以下策略和信任策略的角色。

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

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

以下是信任策略的示例。

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

使用 CreateDataset API 通过 Java 和Amazon CLI

创建数据集。数据集通过应用以下方法存储从数据存储中检索到的数据`queryAction`（一个 SQL 查询）或`containerAction`（执行容器化应用程序）。此操作创建数据集的框架。可以通过调用手动填充数据集`CreateDatasetContent`或者根据 a 自动生成`trigger`您指定。有关更多信息，请参阅 。CreateDataset和`CreateDatasetContent`.

主题

示例 1 — 创建 SQL 数据集 (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);
```

成功时的输出：

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

示例 2 — 使用增量窗口创建 SQL 数据集 (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);
```

成功时的输出：

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

示例 3 — 使用自己的调度触发器 (java) 创建容器数据集

```
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);
```

成功时的输出：

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

示例 4 — 使用 SQL 数据集作为触发器创建容器数据集 (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);
```

成功时的输出：

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

示例 5 — 创建 SQL 数据集 (CLI)

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

成功时的输出：

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

示例 6 — 使用增量窗口 (CLI) 创建 SQL 数据集

增量窗口是一系列用户定义的、非重叠的连续时间间隔。增量窗口使您可以使用上次分析后到达数据存储的新数据创建数据集内容，并对这些数据执行分析。您可以通过设置来创建增量窗口`deltaTime`在里面`filter`sa的一部分`queryAction`数据集 (CreateDataset)。通常，你需要通过设置时间间隔触发器来自动创建数据集内容 (`triggers:schedule:expression`)。基本上，这使您能够筛选在特定时间段内到达的消息，因此来自先前时间窗口的消息中包含的数据不会被计算两次。

在此示例中，我们创建了一个新的数据集，该数据集仅使用自上次以来到达的数据，每 15 分钟自动创建新的数据集内容。我们指定 3 分钟（180 秒）的 deltaTime 偏差，以允许到达指定数据存储的消息延迟 3 分钟。因此，如果数据集内容是在上午 10:30 创建的，则使用的数据（包含在数据集内容中）将是时间戳在上午 10:12 到上午 10:27（即上午 10:30-15 分钟-3 分钟到上午 10:30-3 分钟）之间的数据。

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

文件在哪里delta-window.json包含以下内容。

```
{
  "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 * * * ? *)"
      }
    }
  ]
}
```

成功时的输出：

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

对笔记本进行容器化笔记本

本节包括有关如何使用 Jupyter 笔记本构建 Docker 容器的信息。如果重复使用第三方构建的笔记本，则存在安全风险：包含的容器可能使用您的用户权限执行任意代码。此外，笔记本生成的 HTML 可以显示在Amazon IoT Analytics控制台，在显示 HTML 的计算机上提供潜在的攻击向量。在使用之前，请确保您信任任何第三方笔记本的作者。

执行高级分析函数的一个选项是使用 Jupyter 笔记本。Jupyter Notebook 提供强大的数据科学工具，可以执行机器学习和一系列统计分析。有关更多信息，请参阅 。笔记本模板.（请注意，我们目前不支持内部容器化 JupyterLab。）你可以将你的 Jupyter Notebook 和库打包到一个容器中，该容器在收到新一批数据时会定期运行这些数据Amazon IoT Analytics在您定义的增量时间窗口期间。您可以安排一个使用容器和在指定时间窗口内捕获的新分段数据的分析作业，然后存储该作业的输出以供future 计划分析之用。

如果你已经创建了一个 SageMaker 实例使用Amazon IoT Analytics控制台 2018 年 8 月 23 日之后，容器化扩展的安装已自动为您完成然后你就可以开始创建容器化镜像了. 否则，按照本节中列出的步骤在您的计算机

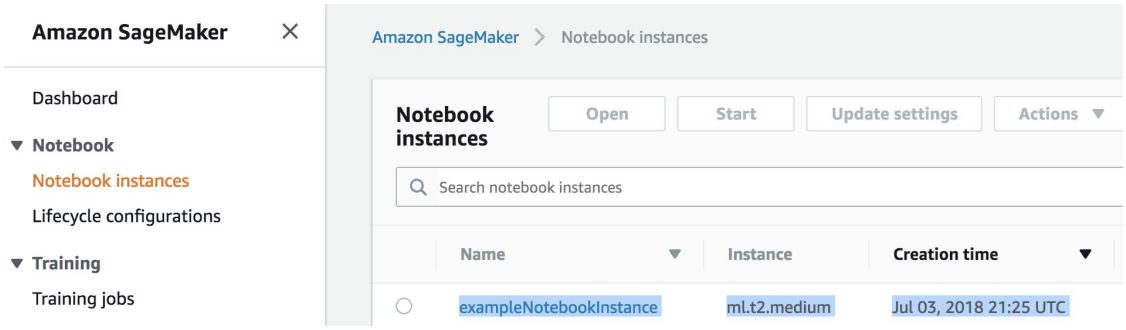
上启用笔记本容器化 SageMaker 实例。在接下来的内容中，你修改你的 SageMaker 执行角色允许您将容器映像上传到 Amazon EC2 并安装容器化扩展程序。

对不是通过创建的笔记本实例启用容器化Amazon IoT Analytics控制台

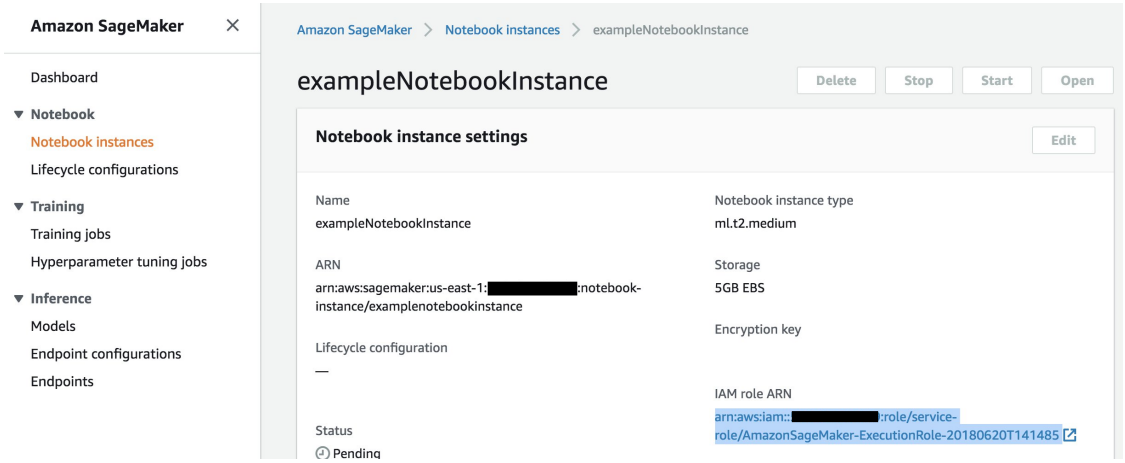
建议您创建新的 SageMaker 通过运行实例Amazon IoT Analytics控制台，而不是按照这些步骤操作。新实例将自动支持容器化。

如果你重启你的 SageMaker 实例启用容器化后（如下所示），您无需重新添加 IAM 角色和策略，但必须重新安装扩展，如最后一步所示。

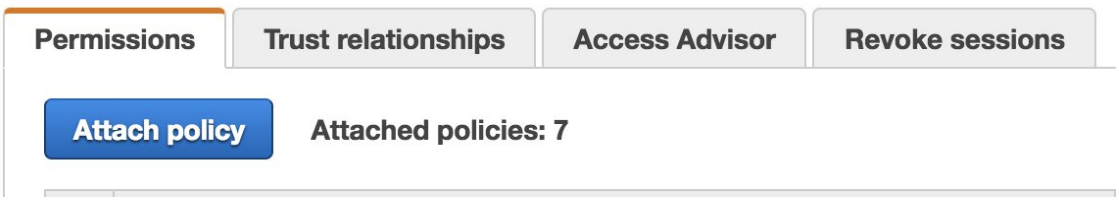
1. 要向您的笔记本实例授予 Amazon ECS 访问权限，请选择您的 SageMaker 上的实例 SageMaker 页面：



2. 下面IAM 角色 ARN，请选择 SageMaker 执行角色。

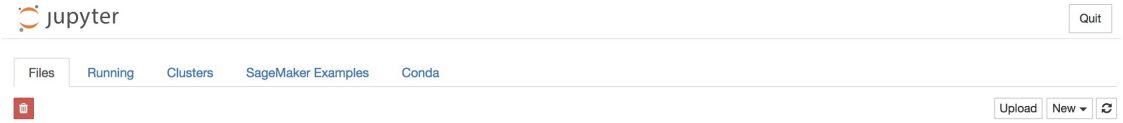


3. 选择 Attach Policy (附加策略)，然后定义并附加权限中显示的策略。如果`AmazonSageMakerFullAccess`政策尚未附上，请附上。



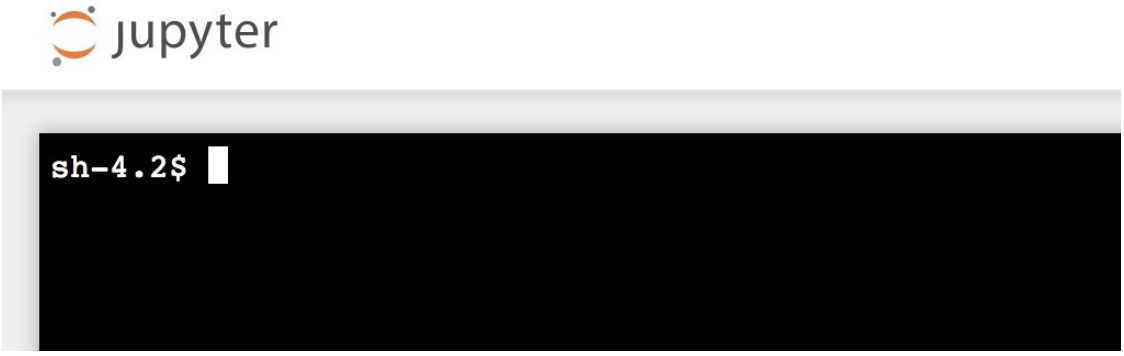
您还必须从 Amazon S3 下载容器化代码并将其安装在您的笔记本实例上，第一步是访问 SageMaker 实例的终端。

1. 在 Jupyter 里面，选择new.



2. 在出现的菜单中，选择终端站.

3. 在终端内，输入以下命令来下载、解压缩和安装代码。请注意，这些命令会杀死您的笔记本电脑在此上运行的所有进程 SageMaker 实例。




```
cd /tmp

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

unzip iota_notebook_containers.zip

cd iota_notebook_containers

chmod u+x install.sh

./install.sh
```

等待一两分钟，以便对扩展进行验证和安装。

更新您的笔记本容器化扩展

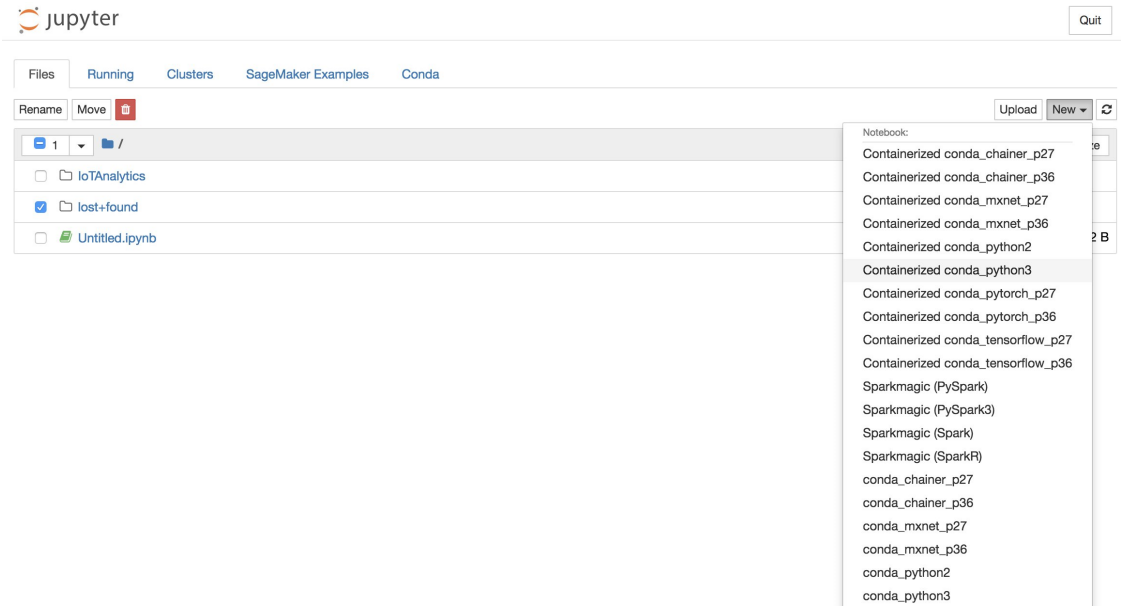
如果你创建了你的 SageMaker 通过运行实例Amazon IoT Analytics控制台在 2018 年 8 月 23 日之后，容器化扩展程序自动安装。您可以通过从重启实例来更新扩展程序 SageMaker 控制台。如果您手动安装了扩展

程序，则可以通过重新运行启用未通过以下方式创建的笔记本实例的容器化中列出的终端命令来对其进行更新Amazon IoT Analytics控制台。

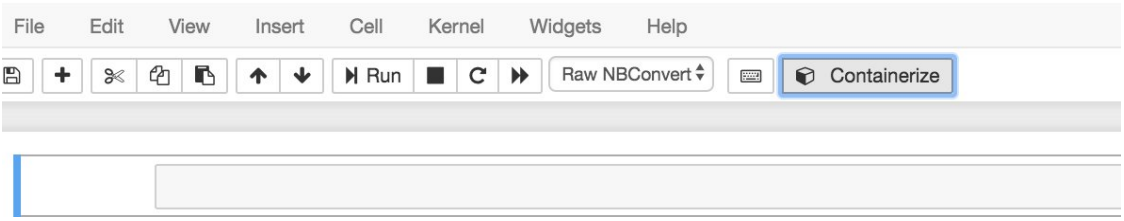
创建容器化映像

在本节中，我们将演示实现笔记本容器化所需的步骤。首先，请转至您的 Jupyter 笔记本，创建具有容器化内核的笔记本。

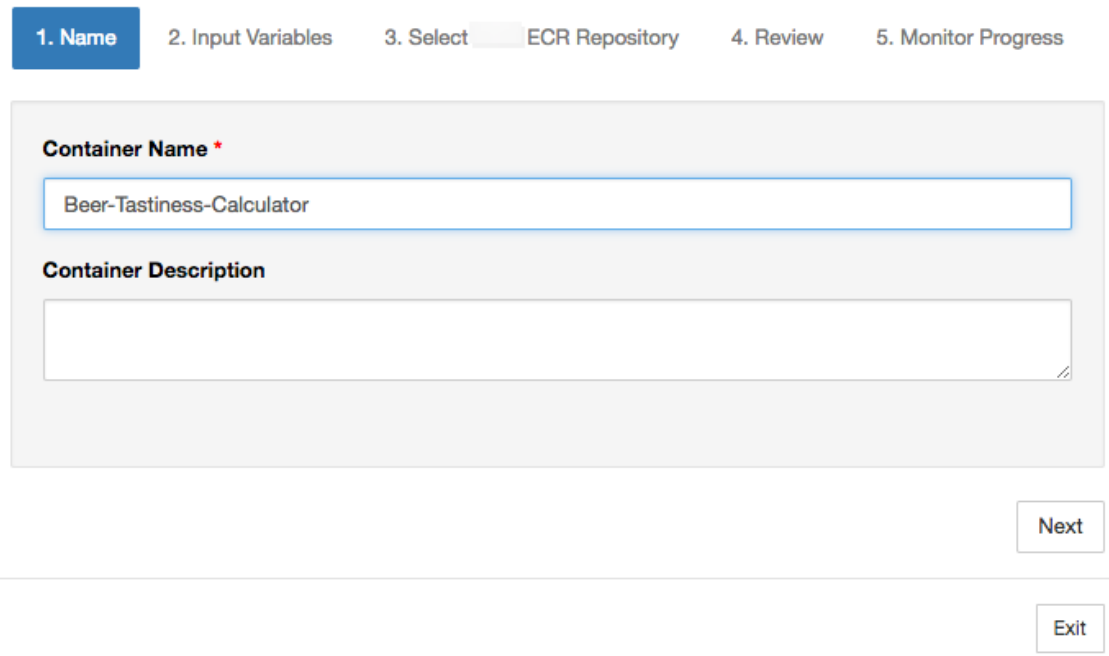
1. 在您的 Jupyter 笔记本中，选择 New (新建)，然后从下拉列表中选择所需内核类型。（内核类型应以“容器化”开头，以原本会选择的任何内核结尾。 例如，如果你只想要一个像 “conda_python3” 这样的普通的 Python 3.0 环境，请选择 “容器化 conda_python3”）。



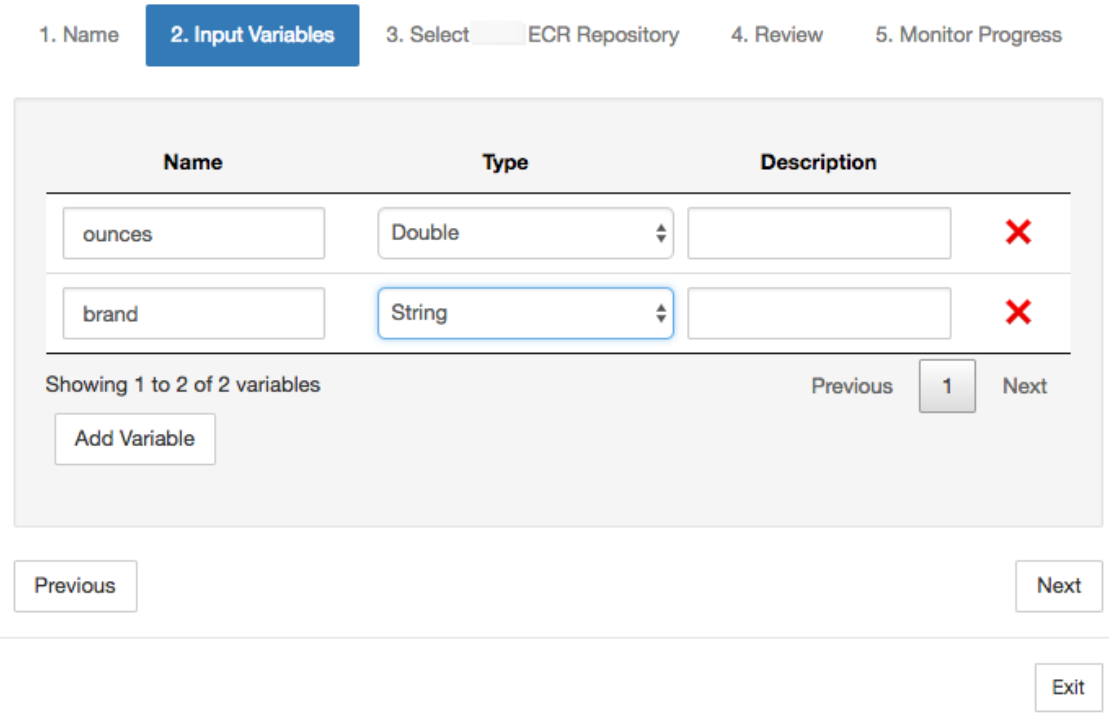
2. 在完成笔记本上的工作并想要将其容器化后，选择容器化.



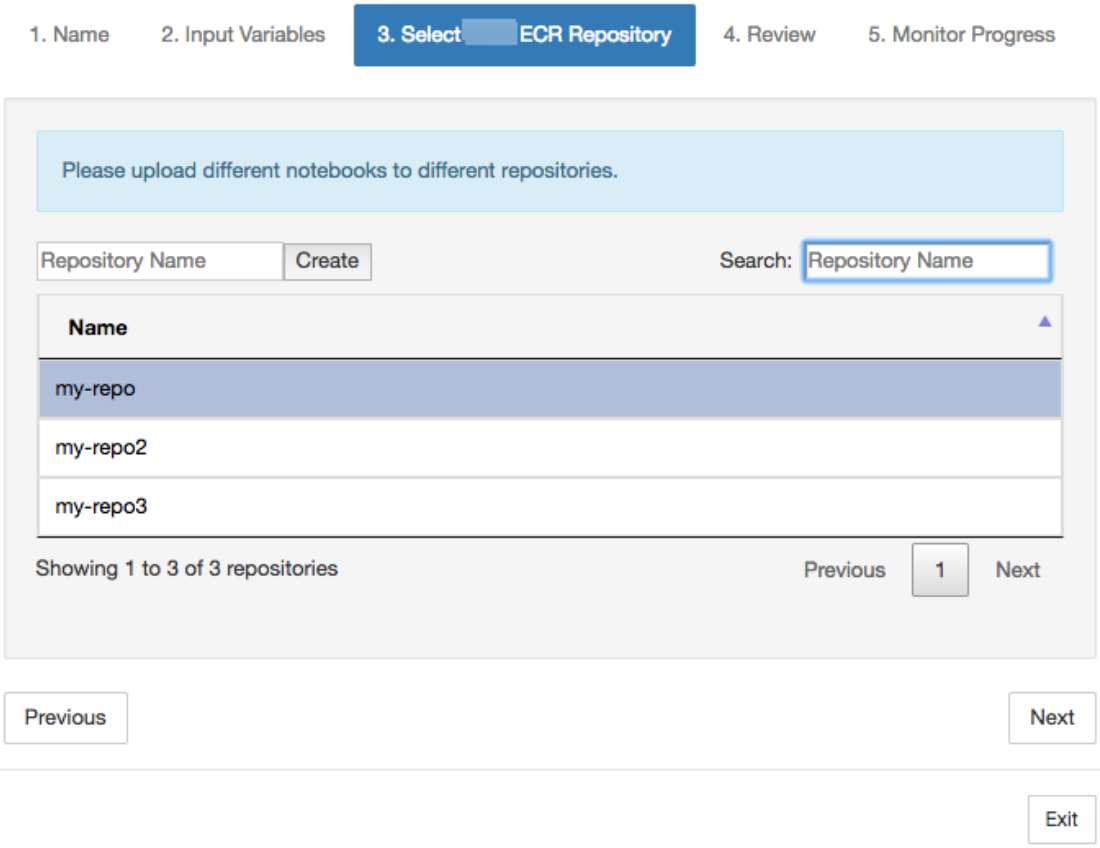
3. 输入容器化笔记本的名称。您还可以输入可选说明。



4. 指定应用于调用笔记本的 Input Variables (输入变量)（参数）。您可以选择从笔记本中自动检测到的输入变量，或定义自定义变量。（请注意，仅当之前已执行过笔记本时，才能检测到输入变量。）对于每个输入变量，选择一种类型。您也可以输入输入变量的可选描述。

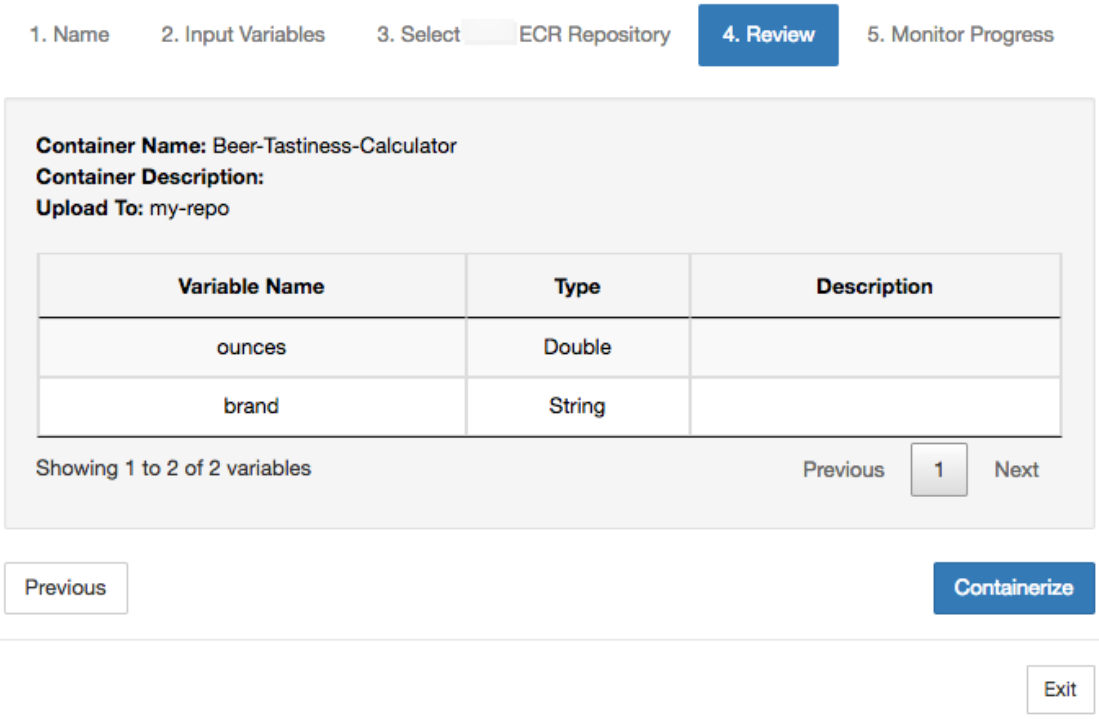


5. 选择 Amazon ECR 存储库，在该存储库中上载从笔记本创建的镜像。

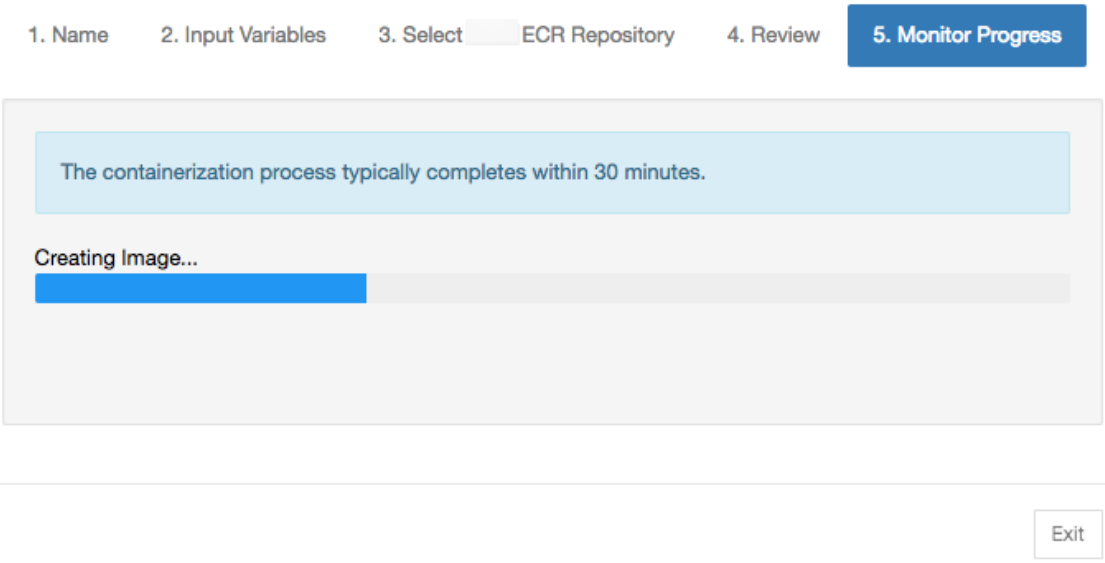


6. 选择容器化开始这个过程。

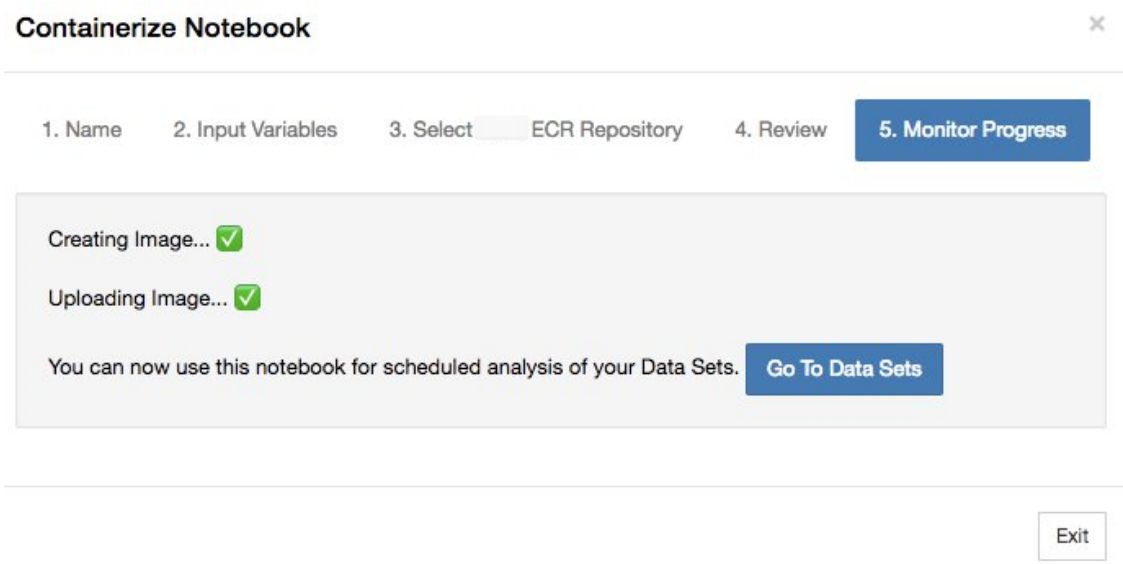
   您将看到一个概述，总结您的输入。请注意，启动此过程之后，将无法取消它。该过程可能持续长达一个小时。



7. 下一页显示进度。



8. 如果您不小心关闭了浏览器，您可以通过以下方式监控容器化过程的状态笔记本部分Amazon IoT Analytics控制台。
9. 该过程完成后，容器化映像将存储在 Amazon ECR 上以备使用。



使用自定义容器进行分析

本节包括有关如何使用 Jupyter 笔记本构建 Docker 容器的信息。如果重复使用第三方构建的笔记本，则存在安全风险：包含的容器可能使用您的用户权限执行任意代码。此外，笔记本生成的 HTML 可以显示在Amazon IoT Analytics控制台，在显示 HTML 的计算机上提供潜在的攻击向量。在使用之前，请确保您信任任何第三方笔记本的作者。

您可以创建自己的自定义容器并运行它Amazon IoT Analytics服务。为此，您需要设置一个 Docker 镜像并将其上传到 Amazon ECR，然后设置一个数据集来运行容器操作。本节提供一个使用 Octave 的过程示例。

本教程假定：

- 本地计算机上已安装 Octave
- 在本地计算机上设置的 Docker 帐户
- 网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的Amazon在Amazon ECR 开设账户Amazon IoT Analytics访问

步骤 1: 设置 Docker 映像

本教程需要三个主要文件。其名称和内容如下：

- `Dockerfile`— Docker 容器化过程的初始设置。

```
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`— 解析来自的 JSONAmazon IoT Analytics，运行 Octave 脚本并将构件上传到 Amazon S3。

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

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

variables = params['Variables']

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

local_input_filename = "input.txt"
local_output_filename = "output.mat"

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

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

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

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

- `moment`— 一个简单的 Octave 脚本，它根据输入或输出文件和指定的顺序计算力矩。

```
#!/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. 下载每个文件的内容。创建一个新目录并将所有文件放入其中，然后`cd`到那个目录。
2. 运行以下命令。

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

3. 你应该在 Docker 存储库中看到一个新镜像。运行以下命令对其进行验证。

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

步骤 2: 将 Docker 镜像上载到 Amazon ECR 存储库

1. 在亚马逊 ECR 中创建存储库。

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

2. 登录您的 Docker 环境。

```
aws ecr get-login
```

3. 复制输出并运行它。输出应与以下内容类似。

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

4. 使用 Amazon ECR 存储库标签标记您创建的映像。

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

5. 将映像推送到 Amazon ECR

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

步骤 3: 将示例数据上载到 Simple StoragAmazon S3 存储桶

1. 将以下内容下载到文件`input.txt`.

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

2. 创建名为的 Amazon S3 存储桶`octave-sample-data-`*`your-aws-account-id`*.
3. 上传文件`input.txt`存储到您刚刚创建的 Amazon S3 存储桶。现在，您应该有了一个名为的存储桶`octave-sample-data-`*`your-aws-account-id`*其中包含`input.txt`文件。

步骤 4: 创建容器执行角色

1. 将以下内容复制到名为的文件中`role1.json`. Replace（替换）*`your-aws-account-id`*用你的Amazon账户 ID 和*`aws-region`*用Amazon您所在区域Amazon资源。

   Note

   此示例包括一个全局条件上下文密钥，用于防止混淆代理安全问题。有关更多信息，请参阅 the section called “跨服务混淆代理问题防范” (p. 94)。

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

2. 创建一个向其授予访问权限的角色 SageMaker 和Amazon IoT Analytics，使用文件`role1.json`您下载的。

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

3. 将以下内容下载到名为的文件中`policy1.json`并更换*your-account-id*使用您的账户 ID（参见下面的第二个 ARN）`Statement:Resource`)。

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

```
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets"
      ],
      "Resource" : "*"
    }
  ]
}
```

4. 创建 IAM policy`policy.json`你刚刚下载的文件。

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

5. 将 策略附加到该角色。

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

步骤 5：使用容器操作创建数据集

1. 将以下内容下载到名为的文件中`cli-input.json`并替换所有实例*your-account-id*和*region*使用适当的值。

```
{
    "datasetName": "octave_dataset",
    "actions": [
        {
            "actionName": "octave",
            "containerAction": {
                "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-moment",
                "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
                "resourceConfiguration": {
                    "computeType": "ACU_1",
                    "volumeSizeInGB": 1
                },
                "variables": [
                    {
                        "name": "octaveResultS3URI",
                        "outputFileUriValue": {
                            "fileName": "output.mat"
                        }
                    },
                    {
                        "name": "inputDataS3BucketName",
                        "stringValue": "octave-sample-data-your-account-id"
                    },
                    {
                        "name": "inputDataS3Key",
                        "stringValue": "input.txt"
```

```
                    },
                    {
                        "name": "order",
                        "stringValue": "3"
                    }
                ]
            }
        }
    ]
}
```

2. 使用文件创建数据集`cli-input.json`你刚刚下载并编辑。

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

步骤 6：调用数据集内容生成

1. 运行以下命令。

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

步骤 7：获取数据集内容

1. 运行以下命令。

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

2. 您可能需要等待几分钟，直到`DatasetContentState`是`SUCCEEDED`.

步骤 8：在 Octave 上打印输出

1. 通过运行以下命令，使用 Octave shell 打印容器的输出。

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

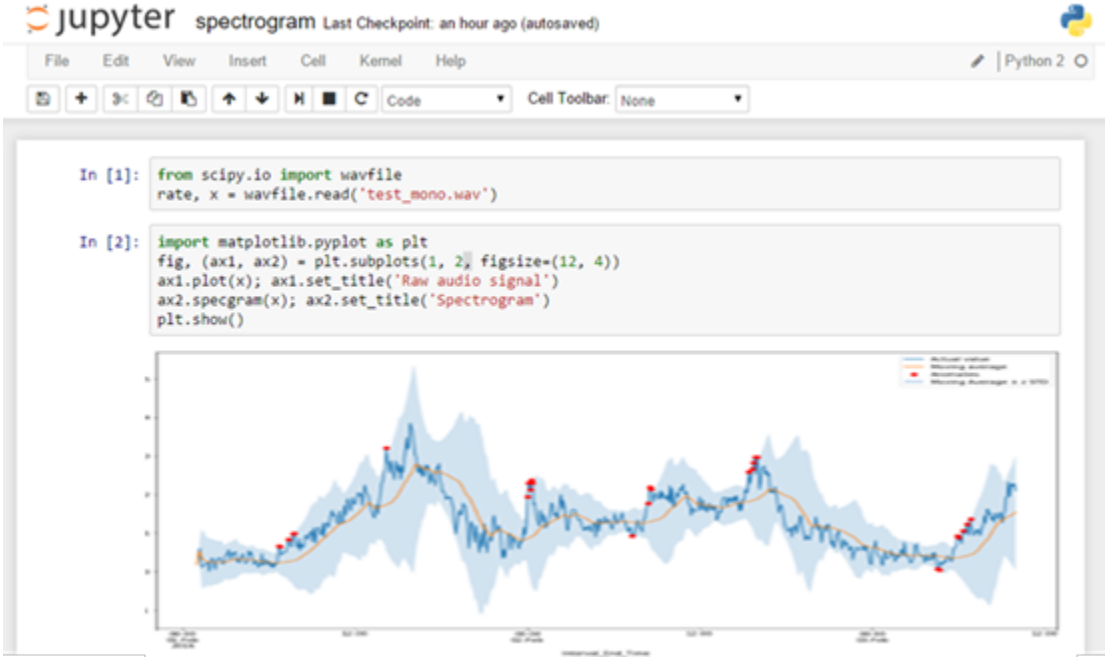
可视化Amazon IoT Analytics数据

要可视化Amazon IoT Analytics数据，您可以使用Amazon IoT Analytics控制台或 Amazon QuickSight。

主题

可视化Amazon IoT Analytics使用控制台进行数据

Amazon IoT Analytics可以嵌入容器数据集的 HTML 输出（在文件中找到）`output.html`) 在Amazon IoT Analytics控制台. 例如，如果您定义一个运行 Jupyter 笔记本的容器数据集，并在 Jupyter 笔记本中创建可视化内容，则您的数据集可能如下所示。



然后，在创建容器数据集内容后，您便可以在控制台上查看此可视化内容了数据集内容页面。

有关创建运行 Jupyter 笔记本的容器数据集的信息，请参阅自动执行工作流程.

给您的 Amazon IoT Analytics 资源加标签

为了方便管理您的通道、数据集、数据存储和管道，您可以选择通过标签的形式为每个资源分配您自己的元数据。本章介绍标签并说明如何创建标签。

主题

有关标签的基本知识

标签可让您按各种标准 (例如用途、所有者或环境) 对 Amazon IoT Analytics 资源进行分类。这在您拥有许多同类型资源时很有用 - 您可以根据分配给资源的标签快速识别特定资源。每个标签都包含您定义的一个键和一个可选值。例如，您可以为通道定义一组标签，以跟踪负责每个通道的消息源的设备类型。我们建议您针对每类资源设计一组标签，以满足您的需要。使用一组连续的标签键，管理资源时会更加轻松。您可以根据添加的标签搜索和筛选资源。

您还可以使用标签对成本进行分类和跟踪。当您将标签应用于频道、数据集、数据存储或管道时，Amazon 以逗号分隔值 (CSV) 文件格式生成一份成本分配报告，其中包括按标签汇总的使用率和成本。您可以设置代表业务类别 (例如成本中心、应用程序名称或所有者) 的标签，以便整理多种服务的成本。有关使用成本分配标签的更多信息，请参阅使用成本分配标签中的Amazon Billing用户指南.

为了便于使用，请使用标签编辑器中的Amazon Billing and Cost Management该控制台提供了一个集中而统一的方法来创建和管理您的标签。有关更多信息，请参阅 。使用标签编辑器在开始使用Amazon Web Services Management Console.

您也可以使用 Amazon CLI 和 Amazon IoT Analytics API 处理标签。可在创建标签时将其与通道、数据集、数据存储和管道关联；可在以下命令中使用 Tags (标签) 字段：

- CreateChannel
- CreateDataset
- CreateDatastore
- CreatePipeline

您可以为支持标记的现有资源添加、修改或删除标签。使用以下命令：

- TagResource
- ListTagsForResource
- UntagResource

您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将其设为空值。如果您添加的标签的键与该标签的键相同，则新值将覆盖旧值。如果删除资源，则所有与资源相关的标签都将被删除。

在 IAM 策略中使用标签

您可以在 IAM 策略中将 Condition 元素（也称为 Condition 块）与以下条件上下文键/值一起使用，以根据资源的标签控制用户访问权限（权限）：

- 使用iotanalytics:ResourceTag/<tag-key>: <tag-value>您允许或拒绝对带特定标签的资源的用户操作。
- 使用 aws:RequestTag/<tag-key>: <tag-value> 可要求在发出创建或修改允许标签的资源的 API 请求时使用（或不使用）特定标签。
- 使用 aws:TagKeys: [<tag-key>, ...] 可要求在发出创建或修改允许标签的资源的 API 请求时使用（或不使用）一组特定标签键。

  Note

  IAM 策限制中的条件上下文键/值仅适用于以下条件上下文键/值。Amazon IoT Analytics能够标记的资源的标识符是必需参数的操作。例如，不会根据条件上下文键/值允许/拒绝使用 DescribeLoggingOptions，因为在该请求中没有引用任何可标记的资源（通道、数据集、数据存储或管道）。

有关更多信息，请参阅《IAM 用户指南》中的使用标签控制访问。这些区域有：IAM JSON 策略参考该指南的一节包含 IAM 中的 JSON 策略的元素、变量和评估逻辑的详细语法、描述和示例。

以下示例策略应用两个基础的限制。受此策略限制的 IAM 用户：

1. 无法为资源提供标签 “env=prod”（请参阅行）"aws:RequestTag/env" : "prod"在示例中）。
2. 无法修改或访问具有现有标签 “env=prod” 的资源（请参阅行）"iotanalytics:ResourceTag/env" : "prod"在示例中）。

```
{
  "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": "*"
```

```
    }
  ]
}
```

您还可以为给定标签键指定多个标签值，方法是将它们括在列表中，如下示例所示。

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

Note

如果您根据标签允许/拒绝用户访问资源，请务必考虑显式拒绝用户在相同资源中添加或删除这些标签的功能。否则，用户可能会修改资源标签以绕过您的限制，并获得该资源的访问权限。

标签限制

下面是适用于标签的基本限制：

- 每个资源的最大标签数 - 50
- 最大密钥长度 - 127 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度 - 255 个 Unicode 字符（采用 UTF-8 格式）
- 标签键和值区分大小写。
- 请勿使用`aws: prefix`请参阅标签名称或值，因为它专为Amazon使用。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个来源的标签数限制。
- 如果您的标记方案针对多个 服务和资源使用，请记得其它服务可能对允许使用的字符有限制。通常允许使用的字符包括：可用 UTF-8 格式表示的字母、空格和数字以及以下特殊字符 + - = . _ : / @。

中的 SQL 表达式Amazon IoT Analytics

数据集是使用 SQL 表达式从数据存储中的数据生成的。Amazon IoT Analytics使用与 Amazon Athena 相同的 SQL 查询、函数和运算符。

Amazon IoT Analytics支持 ANSI 标准 SQL 语法的子集。

```
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 ] ]
```

有关参数的说明，请参阅参数中的Amazon Athena 文档.

Amazon IoT AnalyticsAmazon Athena 不支持以下内容：

- `WITH`子句。
- `CREATE TABLE AS SELECT` 语句
- `INSERT INTO` 语句
- 准备好的陈述，你无法运行`EXECUTE`和`USING`.
- `CREATE TABLE LIKE`
- `DESCRIBE INPUT` 和 `DESCRIBE OUTPUT`
- `EXPLAIN` 语句
- 用户定义的函数 (UDF 或 UDAF)
- 存储过程
- 联合连接器

主题

中支持的 SQL 功能Amazon IoT Analytics

数据集是通过使用 SQL 表达式从数据存储中的数据生成的。你运行的查询Amazon IoT Analytics基于Presto 0.217.

支持的数据类型

Amazon IoT Analytics而且 Amazon Athena 支持这些数据类型。

- primitive_type

    - `TINYINT`
    - `SMALLINT`
    - `INT`
    - `BIGINT`
    - `BOOLEAN`
    - `DOUBLE`
    - `FLOAT`
    - `STRING`
    - `TIMESTAMP`
    - `DECIMAL[()precision, scale]`
    - `DATE`
    - `CHAR`(具有指定长度的长度固定的字符数据)
    - `VARCHAR`(具有指定长度的长度可变的字符数据)
- array_type
    - `ARRAY<data_type>`
- map_type
    - `MAP<primitive_type, data_type>`
- struct_type
    - `STRUCT<col_name:data_type[COMMENT col_comment][,...]>`

    Note

    Amazon IoT Analytics而且 Amazon Athena 不支持某些数据类型。

支持的函数

Amazon Athena 和Amazon IoT AnalyticsSQL 功能基于Presto 0.217. 有关相关函数、运算符和表达式的信息，请参见函数和运算符以及 Presto 文档中的以下具体章节。

- 逻辑运算符
- 比较函数和运算符
- 条件表达式
- 转换函数
- 数学函数和运算符
- 按位函数
- 十进制函数和运算符
- 字符串函数和运算符
- 二进制函数
- 日期与时间函数和运算符
- 正则表达式函数
- JSON 函数和运算符
- URL 函数
- 聚合函数
- 窗口函数
- 颜色函数
- 数组函数和运算符
- 映射函数和运算符

- Lambda 表达式和函数
- Teradata 函数

Note

Amazon IoT AnalyticsAmazon Athena 不支持用户定义的函数 (UDF 或 UDAF) 或存储过程。

排查中 SQL 查询的常见问题Amazon IoT Analytics

可以使用以下信息帮助解决中 SQL 查询的问题：Amazon IoT Analytics.

- 要转义单引号，在它之前再加上另一个单引号。不要将这种情况与双引号混淆。

  Example 示例

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

- 要转义下划线，可以使用反引号将以下划线开头的数据存储列名称括起来。

  Example 示例

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

- 用数字转义姓名，将包含数字的数据存储名称括起来。

  Example 示例

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

- 要转义保留关键字，将保留关键字括起来。有关更多信息，请参阅 。保留关键字的列表在SQL SELECT 语句.

Amazon IoT Analytics 中的安全性

Amazon 十分重视云安全性。作为 Amazon 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是Amazon和您的共同责任。这些区域有：责任共担模式将其描述为安全的云和安全性在云：

- 云的安全性-Amazon负责保护正在运行的基础设施Amazon中的服务Amazon云。Amazon还向您提供可安全使用的服务。作为 Amazon 合规性计划的一部分，我们的安全措施的有效性定期由第三方审计员进行测试和验证。要了解适用于 Amazon IoT Analytics 的合规性计划，请参阅合规性计划范围内的 Amazon 服务。
- 云中的安全性-您的责任由Amazon您使用的服务。您还需要对其他因素负责，包括您的数据的敏感性、您组织的要求以及适用的法律法规。

此文档将帮助您了解如何在使用 Amazon IoT Analytics 时应用责任共担模型。以下主题说明如何配置 Amazon IoT Analytics 以实现您的安全性和合规性目标。你还将学习如何使用其他Amazon服务可帮助您监控和保护您的Amazon IoT Analytics资源。

Amazon Identity and Access Management中的 Amazon IoT Analytics

Amazon Identity and Access Management (IAM) 是一种 Amazon 服务，可以帮助管理员安全地控制对 Amazon 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）来使用 Amazon IoT Analytics 资源。IAM 是一个可以免费使用的 Amazon 服务。

Audience

如何使用Amazon Identity and Access Management(IAM) 因您可以在中执行的操作而异Amazon IoT Analytics.

服务用户 – 如果您使用 Amazon IoT Analytics 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。随着你的使用量越来越多Amazon IoT Analytics功能要完成工作，您可能需要额外权限。了解如何管理访问权限可帮助您向管理员请求适合的权限。如果您无法访问 Amazon IoT Analytics 中的功能，请参阅 对 Amazon IoT Analytics 身份和访问进行故障排除 (p. 101)。

服务管理员 – 如果您在公司负责管理 Amazon IoT Analytics 资源，则您可能具有 Amazon IoT Analytics 的完全访问权限。您有责任确定您的员工应访问哪些 Amazon IoT Analytics 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon IoT Analytics 搭配使用的更多信息，请参阅Amazon IoT Analytics 如何与 IAM 协同工作 (p. 92)。

IAM 管理员-如果您是 IAM 管理员，您可能希望了解有关您可以如何编写可在 IAM 中使用的策略的详细信息，请参阅，请参阅Amazon IoT Analytics 基于身份的策略示例 (p. 98).

使用身份进行身份验证

身份验证是您使用身份凭证登录 Amazon 的方法。有关使用登录的更多信息Amazon Web Services Management Console，请参阅iam Console 和登录页面在IAM 用户指南.

您必须作为 Amazon 账户根用户、IAM 用户或代入 IAM 角色以进行身份验证（登录到 Amazon）。您还可以使用公司的单一登录身份验证方法，甚至使用 Google 或 Facebook 登录。在这些情况下，您的管理员以前使用 IAM 角色设置了联合身份验证。在您使用来自其它公司的凭证访问 Amazon 时，您间接地代入了角色。

要直接登录到Amazon Web Services Management Console，请使用您的密码和根用户电子邮件地址或 IAM 用户名。您可以使用根用户或 IAM 用户访问密钥以编程方式访问 Amazon。Amazon 提供了开发工具包和命令行工具，可使用您的凭证对您的请求进行加密签名。如果您不使用 Amazon 工具，则必须自行对请求签名。使用 Signature Version 4（用于对入站 API 请求进行验证的协议）完成此操作。有关验证请求的更多信息，请参阅《Amazon 一般参考》中的 Signature Version 4 签名流程。

无论使用何种身份验证方法，您可能还需要提供其它安全信息。例如，Amazon 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《IAM 用户指南》中的在 Amazon 中使用多重身份验证 (MFA)。

Amazon 账户根用户

当您首次创建Amazon账户时，最初使用的是一个对账户中所有Amazon服务和资源有完全访问权限的单点登录身份。此身份称为 Amazon 账户根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得该身份。强烈建议您不使用根用户执行日常任务，即使是管理任务。相反，请坚持仅使用根用户创建您的第一个 IAM 用户的最佳实践. 然后请妥善保存根用户凭证，仅用它们执行少数账户和服务管理任务。

IAM 用户和组

IAM 用户是 Amazon 账户内对某个人员或应用程序具有特定权限的一个身份。IAM 用户可能具有长期凭证，例如用户名和密码或一组访问密钥。要了解如何生成访问密钥，请参阅管理 IAM 用户的访问密钥在IAM 用户指南. 为 IAM 用户生成访问密钥时，请确保查看并安全保存key pair。您无法恢复丢失的秘密访问密钥。而是必须生成新的访问密钥对。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 `IAMAdmins` 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的何时创建 IAM 用户（而不是角色）。

IAM 角色

IAM 角色是 Amazon 账户中具有特定权限的实体。它类似于 IAM 用户，但与特定人员不关联。您可以通过切换角色，在 Amazon Web Services Management Console 中暂时代入 IAM 角色。您可以调用 Amazon CLI 或 Amazon API 操作或使用自定义 URL 以代入角色。有关使用角色的方法的更多信息，请参阅 IAM 用户指南 中的使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 临时 IAM 用户权限 - IAM 用户可以担任 IAM 角色，以暂时获得不同的权限以执行特定的任务。
- 联合身份用户访问-您也可以使用来自、的现有用户身份，而不创建 IAM 用户，而不创建 IAM 用户 Amazon Directory Service、您的企业用户目录或 Web 身份提供商。它们被称为联合身份.Amazon在通过访问请求访问权限时，将为联合身份用户分配角色身份提供者. 有关联合身份用户的更多信息，请参阅，请参阅联合身份用户和角色在IAM 用户指南.
- 跨账户访问 - 您可以使用 IAM 角色以允许不同账户中的某个人（可信委托人）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式，请参阅，请参阅IAM 角色与基于资源的策略有何不同在IAM 用户指南.
- Amazon服务访问-服务角色是一个 IAM 角色，服务担任该角色以代表您在您的账户中执行操作。在设置一些 Amazon 服务环境时，您必须为服务定义要代入的角色。此服务角色必须包含该服务访问所需的所有权限Amazon它需要的资源。服务角色因服务而异，但只要您满足服务记录在案的要求，许多服务都允许您选择权限。服务角色只在您的账户内提供访问权限，不能用于为访问其他账户中的服务授权，而不在您的账户内提供访问权限，而不在您的账户内提供访问权限，不能用于 您可以从 IAM 中创建、修改和删除服务

角色。例如，您可以创建一个角色以允许 Amazon Redshift 代表您访问 Amazon S3 存储桶，然后将该存储桶中的数据加载到 Amazon Redshift 集群中。有关更多信息，请参阅 IAM 用户指南中的创建向Amazon服务委派权限的角色。

- 在 Amazon EC2 上运行的应用程序-您可以使用 IAM 角色管理在 EC2 实例上运行并制作的应用程序的临时凭证Amazon CLI要么AmazonAPI 请求。这优先于在 EC2 实例中存储访问密钥。要将 Amazon 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是否使用 IAM 角色，请参阅何时在中创建 IAM 角色（而不是用户）IAM 用户指南.

使用策略管理访问

您将创建策略并将其附加到 IAM 身份或Amazon资源，以便控制Amazon中的访问。策略是Amazon中的对象；在与身份或资源相关联时，策略定义它们的权限。在某个实体（根用户、IAM 用户或 IAM 角色）发出请求时，Amazon将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 Amazon 中存储为 JSON 文档。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概述。

IAM 管理员可以使用策略来指定哪些用户有权访问 Amazon 资源，以及他们可以对这些资源执行哪些操作。每个 IAM 实体（用户或角色）最初没有任何权限。换言之，预设情况下，用户什么都不能做，甚至不能更改他们自己的密码。要为用户授予执行某些操作的权限，管理员必须将权限策略附加到用户。或者，管理员可以将用户添加到具有预期权限的组中。当管理员为某个组授予访问权限时，该组内的全部用户都会获得这些访问权限。

IAM policy 定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。具有该策略的用户可以从 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、角色或组）的 JSON 权限策略文档。这些策略控制身份可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅创建 IAM policy在IAM 用户指南.

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是可以附加到 Amazon 账户中的多个用户、组和角色的独立策略。托管式策略包括 Amazon 托管式策略和客户托管式策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM 用户指南 中的在托管式策略与内联策略之间进行选择。

其它策略类型

Amazon支持额外的、不太常用的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体的基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅 IAM 用户指南中的 IAM 实体的权限边界。
- 服务控制策略数 (SCP)SCP 是 JSON 策略，用于指定中的组织或组织单位 (OU) 的最大权限Amazon Organizations.Amazon Organizations是一项用于分组和集中管理多个服务器的服务Amazon您的企业拥有的账户。如果在组织中启用所有功能，则可以将 SCP 应用于您的任何或所有账户。SCP 限制成员账户中实体的权限，包括每个Amazon账户根用户。有关的更多信息Amazon Organizations和 SCP，请参阅SCP的工作理在Amazon Organizations用户指南.
- 会话策略 - 会话策略是高级策略，在以编程方式为角色或联合身份用户创建临时会话时，这些策略将作为参数进行传递。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解 Amazon 如何确定在涉及多种策略类型时是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Amazon IoT Analytics 如何与 IAM 协同工作

在使用 IAM 管理对 Amazon IoT Analytics 的访问之前，您应了解哪些 IAM 功能可与 Amazon IoT Analytics 结合使用。要大致了解 Amazon IoT Analytics 和其他 Amazon 服务如何与 IAM 一起使用，请参阅 IAM 用户指南中的与 IAM 一起使用的 Amazon 服务。

本页面上的主题：

Amazon IoT Analytics 基于身份的策略

使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。Amazon IoT Analytics 支持特定的操作、资源和条件密钥。要了解您在 JSON 策略中使用的所有元素，请参阅 IAM 用户指南中的 IAM JSON 策略元素参考。

操作

基于 IAM 身份的策略的 `Action` 元素描述该策略将允许或拒绝的特定操作。策略操作通常与关联的 Amazon API 操作同名。这些策略用于策略中以授予执行关联操作的权限。

中的策略操作Amazon IoT Analytics在操作前使用以下前缀：`iotanalytics:`例如，向某人授予创建权限，以创建一个Amazon IoT Analytics通道Amazon IoT Analytics `CreateChannel`API 操作，你包括`iotanalytics:BatchPuMessage`在他们的政策中采取行动。策略语句必须包括 `Action` 或 `NotAction` 元素。Amazon IoT Analytics 定义了自己的一组操作，这些操作描述了可使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示。

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

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，请包括以下操作。

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

若要查看清单Amazon IoT Analytics操作，请参阅操作定义于Amazon IoT Analytics在IAM 用户指南.

资源

`Resource` 元素指定要向其应用操作的对象。语句必须包含 `Resource` 或 `NotResource` 元素。您可使用 ARN 来指定资源，或使用通配符 (*) 以指明该语句适用于所有资源。

这些区域有：Amazon IoT Analytics数据集资源拥有以下 ARN。

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

有关 ARN 格式的更多信息，请参阅 Amazon Resource Name (ARN) 和 Amazon 服务命名空间。

例如，要在语句中指定 Foobar 数据集，请使用以下 ARN。

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

要指定属于特定账户的所有实例，请使用通配符 (*)。

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

无法对特定资源执行某些 Amazon IoT Analytics 操作，例如，用于创建资源的操作。在这些情况下，您必须使用通配符 (*)。

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

一些 Amazon IoT Analytics API 操作涉及多种资源。例如，CreatePipeline引用作为频道和数据集，从而使 IAM 用户必须获得相应权限才能使用频道和数据集，从而获得相应权限才能使用频道和数据集。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

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

若要查看清单Amazon IoT Analytics资源类型及其 ARN，请参阅定义的资源Amazon IoT Analytics在IAM 用户指南. 要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 Amazon IoT Analytics 定义的操作。

条件键

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以构建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 Amazon 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则 Amazon 使用逻辑 OR 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅 。IAM 策略元素：变量和标签在IAM 用户指南.

Amazon IoT Analytics不提供任何服务特定的条件键，但支持使用某些全局条件键，但支持使用某些全局条件键。查看全部Amazon全局条件键，请参阅Amazon全局条件上下文键. 在IAM 用户指南.

示例

要查看 Amazon IoT Analytics 基于身份的策略的示例，请参阅 Amazon IoT Analytics 基于身份的策略示例 (p. 98)。

Amazon IoT Analytics 基于资源的策略

Amazon IoT Analytics 不支持基于资源的策略。要查看详细的基于资源的策略页面的示例，请参阅，请参阅使用 的基于资源的策略Amazon Lambda在Amazon Lambda开发人员指南.

基于 Amazon IoT Analytics 标签的授权

您可以将标签附加到 Amazon IoT Analytics 资源或将请求中的标签传递到 Amazon IoT Analytics。要基于标签控制访问，您需要在中提供标签信息，请在中提供标签信息条件元素使用`iotanalytics:ResourceTag/{key-name}, aws:RequestTag/{key-name}`要么`aws:TagKeys`条件键。有关标记的更多信息Amazon IoT Analytics资源，请参阅为添加标签Amazon IoT Analytics资源.

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅，请参阅查看Amazon IoT Analytics根据标签进行通道.

Amazon IoT Analytics IAM 角色

IAM 角色是 Amazon 账户中具有特定权限的实体。

将临时凭证用于Amazon IoT Analytics

您可以使用临时凭证进行联合身份登录，担任 IAM 角色或担任跨账户角色。您可以通过调用获取临时安全凭证Amazon Security Token Service(Amazon STS) API 操作，例如AssumeRole要么GetFederationToken.

Amazon IoT Analytics 不支持使用临时凭证。

服务相关角色

服务相关角色允许Amazon服务访问其他服务中的资源以代表您完成操作。服务相关角色显示在您的 IAM 账户中，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

Amazon IoT Analytics 不支持服务相关角色。

服务角色

此功能允许服务代表您担任服务角色。此角色允许服务访问其它服务中的资源以代表您完成操作。服务角色显示在您的 IAM 账户中，并归该账户所有。这意味着，IAM 管理员可以更改该角色的权限。但是，这样做可能会中断服务的功能。

Amazon IoT Analytics 支持服务角色。

跨服务混淆代理问题防范

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在 Amazon 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务)时，可能会发生跨服务模拟。可以操纵调用服务以使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为了防止这种情况，Amazon提供可帮助您保护所有服务主体有权限访问账户中的资源。

我们建议使用`aws:SourceArn`和`aws:SourceAccount`资源策略中的上下文密钥。这限制了以下权限Amazon IoT Analytics为资源提供另一项服务。如果使用两个全局条件上下文键，在同一策略语句中使用时，`aws:SourceAccount` 值和 `aws:SourceArn` 值中的账户必须使用相同的账户 ID。

防止混淆代理问题最有效的方法是使用具有资源完整 Amazon Resource Name (ARN) 的`aws:SourceArn` 全局条件上下文键。如果您不知道资源的完整 ARN，或正在指定多个资源，请针对 ARN 未知部分使用带有通配符 (*) 的 `aws:SourceArn` 全局上下文条件键。例如，`arn:aws:iotanalytics::123456789012:*`。

主题

- Amazon S3 的防范桶 (p. 95)
- 亚马逊的预防 CloudWatch 日志 (p. 96)

- 防止责任管理混淆代理问题防范Amazon IoT Analytics资源 (p. 97)

Amazon S3 的防范桶

如果您使用客户管理的 Amazon S3 存储Amazon IoT Analytics数据存储，存储您的数据的 Amazon S3 存储桶可能会出现混乱的副手问题。

例如，Nikki Wolf 使用客户拥有的名为 Amazon S3 存储桶*#######*. 存储桶存储以下信息Amazon IoT Analytics在该地区创建的数据存储*us-east-1*. 她指定了一项政策，该策略允许Amazon IoT Analytics要查询的服务主体*#######*代表她。Nikki 的同事李娟询问*#######*从她自己的账户中创建一个包含结果的数据集。因此，Amazon IoT Analytics服务负责人代表 Li 查询了 Nikki 的 Amazon S3 存储桶，尽管 Li 通过她的账户进行了查询。

为了防止这种情况，Nikki 可以指定`aws:SourceAccount`条件或`aws:SourceArn`保单中的条件*#######*.

指定**`aws:SourceAccount`**条件-以下存储桶策略示例指定只有Amazon IoT Analytics来自 Nikki 账户的资源 (*123456789012*) 可以访问*#######*.

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

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

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

```
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-EXAMPLE-
DATASET",
                        "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-EXAMPLE-
DATASTORE"
                    ]
                }
            }
        }
    ]
}
```

亚马逊的预防 CloudWatch 日志

在使用Amazon 进行监控时，您可以防止责任混淆代理问题 CloudWatch 日志。以下资源策略显示了如何防止出现混淆的副手问题：

- 全局条件上下文密钥，`aws:SourceArn`
- 这些区域有：`aws:SourceAccount`用你的Amazon账户 ID
- 与之关联的客户资源`sts:AssumeRole`请求Amazon IoT Analytics

Replace（替换）*123456789012*用你的Amazon账户 ID 和*us-east-1*与您所在的地区Amazon IoT Analytics以下示例中的账户。

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

```
}
```

有关启用和配置 Amazon 的更多信息 CloudWatch 日志，请参阅the section called “日志记录和监控” (p. 103).

防止责任管理混淆代理问题防范Amazon IoT Analytics资源

如果你授予Amazon IoT Analytics允许您执行操作Amazon IoT Analytics资源，资源可能会遇到混乱的副手问题。为了防止出现混淆副手的问题，你可以限制授予的权限Amazon IoT Analytics以下是资源策略示例。

主题

预防Amazon IoT Analytics信道和数据存储

您可以使用 IAM 角色来控制Amazon这些资源Amazon IoT Analytics可以代表您访问。为了防止你的角色面临困惑的副手问题，你可以指定Amazon账户在`aws:SourceAccount`元素和的 ARNAmazon IoT Analytics中的资源`aws:SourceArn`您附加到角色的信任策略中的元素。

在以下示例中，替换*123456789012*用你的Amazon账户 ID 和*arn: aws: #####:aws-region:123456789012: Channel/doc-example-Channel*使用的 ARNAmazon IoT Analytics频道或数据存储。

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

要了解有关通道和数据存储的客户托管 S3 存储选项的更多信息，请参阅`CustomerManagedChannelS3Storage`和`CustomerManagedDatastoreS3Storage`在Amazon IoT AnalyticsAPI 参考.

防止跨服务混淆代理问题防范Amazon IoT Analytics数据集内容分发规则

IAM 角色Amazon IoT Analytics假设将数据集查询结果提供给 Amazon S3 或Amazon IoT Events可能会遇到混乱的副手问题。为防止责任混淆代理问题，指定Amazon账户在`aws:SourceAccount`元素和的 ARNAmazon IoT Analytics中的资源`aws:SourceArn`您附加到角色的信任策略的元素。

```
{
```

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

有关配置数据集内容分发规则的更多详细信息，请参阅`contentDeliveryRules`在Amazon IoT AnalyticsAPI 参考.

Amazon IoT Analytics 基于身份的策略示例

预设情况下，IAM 用户和角色没有创建或修改 Amazon IoT Analytics 资源的权限。它们还无法使用 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 执行任务。IAM 管理员必须创建 IAM policy，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

要了解如何使用这些示例 JSON 策略文档创建基于身份的 IAM 策略，请参阅在“JSON”选项卡上创建策略在IAM 用户指南

本页面上的主题：

策略最佳实践

基于身份的策略非常强大。它们确定某个人是否可以创建、访问或删除您账户中的 Amazon IoT Analytics 资源。这些操作可能会使 Amazon 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 入门Amazon托管策略-开始使用Amazon IoT Analytics快点，使用Amazon托管策略，为您的员工授予他们所需的权限。这些策略已在您的账户中提供，并由维护和更新Amazon. 有关更多信息，请参阅 IAM 用户指南中的开始使用 Amazon 托管式策略中的权限。
- 授予最低权限-当您创建自定义策略时，仅授予执行任务所需的许可。最开始只授予最低权限，然后根据需要授予其它权限。这样做比起一开始就授予过于宽松的权限而后再尝试收紧权限来说更为安全。有关更多信息，请参阅《IAM 用户指南》中的授予最低权限。
- 为敏感操作启用 MFA-为增强安全性，要求 IAM 用户使用多重身份验证 (MFA) 来访问敏感资源或 API 操作。要了解更多信息，请参阅 IAM 用户指南中的在 Amazon 中使用多重身份验证 (MFA)。
- 使用策略条件来增强安全性-在切实可行的范围内，定义基于身份的策略在哪些情况下允许访问资源。例如，您可编写条件来指定请求必须来自允许的 IP 地址范围。您也可以编写条件，以便仅允许指定日

期或时间范围内的请求，或者要求使用 SSL 或 MFA。有关更多信息，请参阅 。IAM JSON 策略元素：Condition在IAM 用户指南.

使用 Amazon IoT Analytics 控制台

要访问 Amazon IoT Analytics 控制台，您必须拥有一组最低的权限。这些权限必须允许您列出和查看有关您的 Amazon 账户中的 Amazon IoT Analytics 资源的详细信息。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（IAM 用户或角色），控制台将无法按预期正常运行。

要确保这些实体仍可使用 Amazon IoT Analytics 控制台，也可向实体附加以下 Amazon 托管策略。有关更多信息，请参阅 IAM 用户指南 中的为用户添加权限。

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "iotanalytics:BatchPutMessage",
                 "iotanalytics:CancelPipelineReprocessing",
                 "iotanalytics:CreateChannel",
                 "iotanalytics:CreateDataset",
                 "iotanalytics:CreateDatasetContent",
                 "iotanalytics:CreateDatastore",
                 "iotanalytics:CreatePipeline",
                 "iotanalytics:DeleteChannel",
                 "iotanalytics:DeleteDataset",
                 "iotanalytics:DeleteDatasetContent",
                 "iotanalytics:DeleteDatastore",
                 "iotanalytics:DeletePipeline",
                 "iotanalytics:DescribeChannel",
                 "iotanalytics:DescribeDataset",
                 "iotanalytics:DescribeDatastore",
                 "iotanalytics:DescribeLoggingOptions",
                 "iotanalytics:DescribePipeline",
                 "iotanalytics:GetDatasetContent",
                 "iotanalytics:ListChannels",
                 "iotanalytics:ListDatasetContents",
                 "iotanalytics:ListDatasets",
                 "iotanalytics:ListDatastores",
                 "iotanalytics:ListPipelines",
                 "iotanalytics:ListTagsForResource",
                 "iotanalytics:PutLoggingOptions",
                 "iotanalytics:RunPipelineActivity",
                 "iotanalytics:SampleChannelData",
                 "iotanalytics:StartPipelineReprocessing",
                 "iotanalytics:TagResource",
                 "iotanalytics:UntagResource",
                 "iotanalytics:UpdateChannel",
                 "iotanalytics:UpdateDataset",
                 "iotanalytics:UpdateDatastore",
                 "iotanalytics:UpdatePipeline"
             ],
             "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
             "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
             "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
             "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
         }
     ]
```

```
 }
```

对于只需要调用 Amazon CLI 或 Amazon API 的用户，无需为其提供最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上完成此操作或者以编程方式使用 Amazon CLI 或 Amazon API 所需的权限。

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

访问一个Amazon IoT Analytics输入

在本示例中，您希望在您的账户中授予一个 IAM 用户Amazon账户访问某个账户Amazon IoT Analytics通道，`exampleChannel`. 您还希望允许用户添加、更新和删除频道。

该政策授予`iotanalytics:ListChannels`, `iotanalytics:DescribeChannel`, `iotanalytics:CreateChannel`, `iotanalytics:DeleteChannel`, and `iotanalytics:UpdateChannel`权限。有关 Amazon S3 服务向用户授予权限并使用控制台测试这些权限的示例演练，请参阅，请参阅演练示例：使用用户策略控制对存储桶的访问.

```
{
   "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/*"
      }
   ]
}
```

查看Amazon IoT Analytics根据标签进行通道

您可以在基于身份的策略中使用条件，以便基于标签控制对 Amazon IoT Analytics 资源的访问。此示例显示如何创建策略以允许查看 channel。但是，只有在以下情况下才会授予权限channel标签Owner具有该用户的用户名值。该策略还会授予在控制台上完成该操作所需的权限。

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

您可以将该策略附加到您账户中的 IAM 用户。如果用户名为richard-roe尝试查看Amazon IoT Analytics channel，channel必须标记Owner=richard-roe or owner=richard-roe. 否则，他将被拒绝访问。条件标签键 Owner 匹配 Owner 和 owner，因为条件键名称不区分大小写。有关更多信息，请参阅 。IAM JSON 策略元素：Condition在IAM 用户指南.

对 Amazon IoT Analytics 身份和访问进行故障排除

使用以下信息可帮助您诊断和修复在使用 Amazon IoT Analytics 时可能遇到的常见问题。

主题

我无权在 Amazon IoT Analytics 中执行操作

如果Amazon管理控制台告诉您，您无权执行某个操作，因此必须联系您的管理员寻求帮助，以获得帮助，以获得帮助。您的管理员是指为您提供用户名和密码的那个人。

以下示例错误发生在`mateojackson`IAM 用户尝试使用控制台查看有关`channel`但没有`iotanalytics:ListChannels`权限。

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

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他访问`my-example-channel`资源使用`iotanalytics:ListChannel`操作。

我无权执行 iam：PassRole

如果您收到错误消息，提示您无权执行 `iam:PassRole` 操作，则必须联系您的管理员寻求帮助。您的管理员是指为您提供用户名和密码的那个人。请求该人员更新您的策略，以便允许您将角色传递给 Amazon IoT Analytics。

有些 Amazon 服务允许您将现有角色传递到该服务，而不是创建新服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 Amazon IoT Analytics 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，Mary 请求她的管理员来更新其策略，以允许她执行 `iam:PassRole` 操作。

我想要查看我的访问密钥

在创建 IAM 用户访问密钥后，您可以随时查看您的访问密钥 ID。但是，您无法再查看您的秘密访问密钥。如果您丢失了私有密钥，则必须创建一个新的访问密钥对。

访问密钥由两个部分组成：访问密钥 ID（例如`AKIAIOSFODNN7EXAMPL`) 和秘密访问密钥（例如，`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`)。与用户名和密码一样，您必须同时使用访问密钥 ID 和秘密访问密钥对请求执行身份验证。像对用户名和密码一样，安全地管理访问密钥。

> Important
>
> 请不要向第三方提供访问密钥，即便是为了帮助找到您的规范用户 ID 也不行。如果您这样做，可能会向某人提供对您的账户的永久访问权限。

当您创建访问key pair 时，系统会提示您将访问密钥 ID 和秘密访问密钥保存在一个安全位置。秘密访问密钥仅在您创建它时可用。如果丢失了您的秘密访问密钥，您必须为 IAM 用户添加新的访问密钥。您最多可拥有两个访问密钥。如果已有两个访问密钥，则必须删除一个key pair，然后才能创建新的密钥。要查看说明，请参阅 IAM 用户指南中的管理访问密钥。

我是管理员并希望允许其他人访问 Amazon IoT Analytics

要允许其他人访问 Amazon IoT Analytics，您必须为需要访问权限的人员或应用程序创建一个 IAM 实体（用户或角色）。它们将使用该实体的凭证访问 Amazon。然后，您必须将策略附加到实体，以便在 Amazon IoT Analytics 中向其授予正确的权限。

要正确入门，请参阅创建您的第一个 IAM 委派用户和组在IAM 用户指南.

我希望允许我的 Amazon 账户之外的人员访问我的 Amazon IoT Analytics 资源

您可以创建一个角色，以便其它账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权，而不支持基于资源的策略或访问控制列表 (ACL) 的服务，而不支持基于资源的策略或访问控制列表

要了解更多信息，请参阅以下内容：

- 如需了解Amazon IoT Analytics支持这些功能，请参阅如何Amazon IoT Analytics与 IAM 协同工作.
- 要了解如何为您拥有的 Amazon 账户中的资源提供访问权限，请参阅IAM 用户指南中的为您拥有的另一个 Amazon 账户中的 IAM 用户提供访问权限。
- 要了解如何为第三方 Amazon 账户提供您的资源的访问权限，请参阅IAM 用户指南中的 为第三方拥有的 Amazon 账户提供访问权限。
- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅 IAM 用户指南中的 IAM 角色与基于资源的策略有何不同。

Amazon IoT Analytics 中的日志记录和监控

Amazon 为您提供了各种可用于监控 Amazon IoT Analytics 的工具。您可以配置其中的一些工具以便进行监控。一些工具需要手动干预。建议您尽可能实现监控任务自动化。

自动监控工具

您可以使用以下自动化监控工具来监控 Amazon IoT 并在出现错误时报告：

- 亚马逊 CloudWatch 日志-监控、存储和访问您的日志文件Amazon CloudTrail或者其他源。有关更多信息，请参阅 。是什么Amazon CloudTrail监控日志文件亚马逊 CloudWatch 用户指南.
- Amazon CloudTrail日志监控-在账户间共享日志文件，监控 CloudTrail 通过将文件发送到来实时记录文件 CloudWatch 日志、使用 Java 编写日志处理应用程序，以及验证您的日志文件在由 CloudTrail 传送后未发生更改。有关更多信息，请参阅 。使用 CloudTrail 日志文件中的Amazon CloudTrail用户指南.

手动监控工具

监控 Amazon IoT 的另一个重要环节是手动监控 CloudWatch 警报未涵盖的项目。这些区域有：Amazon IoT、CloudWatch 和其他Amazon服务控制台仪表板提供 at-a-glance 查看您的状态Amazon环境。建议您还要查看 Amazon IoT Analytics 上的日志文件。

- 该 Amazon IoT Analytics 控制台显示：
  - 通道

- 管道
- 数据存储
- 数据集
- 笔记本
- 设置
- 学习

- 这些区域有： CloudWatch 主页显示：
  - 当前告警和状态
  - 告警和资源图表
  - 服务运行状况

  此外，您还可以使用 CloudWatch 执行以下操作：
  - 创建自定义控制面板以监控您关心的服务
  - 绘制指标数据图，以排除问题并弄清楚趋势
  - 搜索并浏览您所有的 Amazon 资源指标
  - 创建和编辑警报以接收有关问题的通知

使用 Amazon 进行监控 CloudWatch 日志

Amazon IoT Analytics支持使用 Amazon CloudWatch 进行记录。您可以启用和配置 Amazon CloudWatch 正在登录Amazon IoT Analytics通过使用`PutLoggingOptions`API 操作. 本节说明如何将`PutLoggingOptions`和Amazon Identity and Access Management(IAM) 来配置和启用亚马逊 CloudWatch 正在登录Amazon IoT Analytics.

有关 的更多信息 CloudWatch 日志，请参阅亚马逊 CloudWatch 日志用户指南. 有关 的更多信息 AmazonIAM，请参阅Amazon Identity and Access Management用户指南.

> Note
>
> 在启用之前Amazon IoT Analytics日志记录中，请务必了解 CloudWatch 日志访问权限。有权访问的用户 CloudWatch 日志可以查看您的调试信息。有关更多信息，请参阅 。Amazon 的身份验证和访问控制 CloudWatch 日志.

创建 IAM 角色以启用日志记录

要创建 IAM 角色以便为 Amazon 启用日志记录 CloudWatch

1. 使用AmazonIAM 控制台或以下权限AmazonIAM CLI 命令，CreateRole，以使用信任关系策略（信任策略）创建新的 IAM 角色。此信任策略为 Amazon CloudWatch 等实体授予权限以担任该角色。

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

   这些区域有： `exampleTrustPolicy.json`文件包含以下内容。

   > Note
   >
   > 此示例包括一个全局条件上下文密钥，用于防止混淆代理安全问题。Replace*123456789012*与您的Amazon和账户 ID*aws-region*使用Amazon您的区域Amazon资源的费用。有关更多信息，请参阅 the section called “跨服务混淆代理问题防范” (p. 94)。

   ```
   {
   ```

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

      }
   ]
}
```

稍后，在调用时，您可以使用此角色的 ARN。Amazon IoT Analytics `PutLoggingOptions`命令。

2. 使用AmazonIAMPutRolePolicy附加权限策略（`arole policy`）到您在步骤 1 中创建的角色。

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

   Simple OolePolicy .json 文件包含以下内容。

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
      {
      "Effect": "Allow",
      "Action": [
       "logs:CreateLogGroup",
       "logs:CreateLogStream"
      ],
      "Resource": [
       "arn:aws:logs:*:*:*"
      ],
       "Condition":{
       "ArnLike":{
        "aws:SourceArn":"arn:aws:iotanalytics:*"
       },
       "StringEquals":{
        "aws:SourceAccount":"123456789012"
       }
      }
      }
      ]
   }
   ```

3. 为了给予Amazon IoT Analytics向 Amazon CloudWatch 提供日志记录事件的权限，请使用亚马逊 CloudWatch 命令PutResourcePolicy.

   Note

   为了帮助避免出现混淆代理安全问题，我们建议您指定`aws:SourceArn`在你的资源策略中。这限制了访问权限，以仅允许来自指定账户的那些请求。有关混淆代理问题的更多信息，请参阅the section called “跨服务混淆代理问题防范” (p. 94).

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

这些区域有：exampleResourcePolicy.json文件包含以下资源策略。

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

配置并启用日志记录

使用PutLoggingOptions用于配置和启用亚马逊的命令 CloudWatch 正在登录Amazon IoT Analytics. loggingOptions 字段中的 roleArn 应为您在上一节中创建的角色的 ARN。还可以使用 DecribeLoggingOptions 命令来检查您的日志记录选项设置。

PutLoggingOptions

设置或更新Amazon IoT Analytics日志记录选项。如果你更新了任何的值loggingOptions字段，更改最多需要一分钟才能生效。此外，如果您更改附加到您在roleArn字段（例如，要更正政策这是无效的），则最多可能需要五分钟时间，此更改才能生效。有关更多信息，请参阅 PutLoggingOptions。

DescribeLoggingOptions

检索的当前设置Amazon IoT Analytics日志记录选项。有关更多信息，请参阅 。DescribeLoggingOptions

命名空间、指标和维度

Amazon IoT Analytics将以下指标放入亚马逊： CloudWatch 存储库：

命名空间
Amazon/IoTAnalytics

指标	描述
ActionExecution	操作数被处决.
操作执行受限	受限制的操作数。
ActivityExecutionError	执行管道活动时生成的错误数。

指标	描述
IncomingMessages	进入通道的消息数量。
PipelinE 并发执行计数	同时执行的管道活动的数量。

维度	描述
ActionType	正在监视的操作类型。
ChannelName	正在监视的通道的名称。
DatasetName	正在监视的数据集的名称。
DatastoreName	正在监视的数据存储的名称。
PipelineActivityName	正在监视的管道活动的名称。
PipelineActivityType	正在监视的管道活动的类型。
PipelineName	正在监视的管道的名称。

使用 Amazon 进行监控 CloudWatch 事件

Amazon IoT Analytics自动向亚马逊发布活动 CloudWatch 在Amazon Lambda活动。此事件包含详细的错误消息以及存储未处理的通道消息的 Amazon Simple Storage Service (Amazon S3) 对象的密钥。您可以使用 Amazon S3 密钥重新处理未处理的频道消息。有关更多信息，请参阅 。重新处理通道消息 (p. 57)，StartPipelineReprocessing中的 APIAmazon IoT AnalyticsAPI 参考, 和什么是 Amazon CloudWatch 事件中的亚马逊 CloudWatch 事件用户指南.

您也可以配置启用 Amazon 的目标。 CloudWatch 发送通知或采取进一步行动的事件。例如，您可以将通知发送到 Amazon Simple Queue Service (Amazon SQS) 队列，然后调用`StartReprocessingMessage`用于处理保存在 Amazon S3 对象中的频道消息的 API。亚马逊 CloudWatch 事件支持许多类型的目标，如下所示：

- Amazon Kinesis Streams
- Amazon Lambda 函数
- Amazon Simple Notification Service (Amazon SNS) 主题
- Amazon Simple Queue Service (Amazon SQS) 队列

有关支持的目标列表，请参阅亚马逊 EventBridge 目标中的亚马逊 EventBridge 用户指南.

您的 CloudWatch 事件资源和相关目标必须位于Amazon创建您的区域Amazon IoT Analytics资源的费用。有关更多信息，请参阅 。服务终端节点和配额中的Amazon一般参考.

发送给亚马逊的通知 CloudWatch 中的运行时错误的事件Amazon LambdaActivity 采用以下格式。

```
{
"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"
}
}
```

示例通知：

```
{
"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"
}
}
```

通过亚马逊获取延迟数据通知 CloudWatch 事件

当您使用指定时间范围内的数据创建数据集内容时，某些数据可能无法及时到达以进行处理。要允许延迟，可以指定`deltaTime`对于的偏移量`QueryFilter`当您时创建数据集通过应用`queryAction`（SQL 查询）。Amazon IoT Analytics仍然处理在增量时间内到达的数据，并且您的数据集内容有时滞。延迟数据通知功能启用Amazon IoT Analytics通过发送通知亚马逊 CloudWatch 事件当数据在增量时间之后到达时。

您可以使用Amazon IoT Analytics控制台，API、Amazon Command Line Interface(Amazon CLI），或者Amazon开发工具包以便为数据集指定晚期数据规则。

在Amazon IoT AnalyticsAPI，`LateDataRuleConfiguration`对象表示数据集的晚期数据规则设置。这个对象是`Dataset`与关联的对象`CreateDataset`和`UpdateDataset`API 操作。

参数

当您为数据集创建延迟数据规则时Amazon IoT Analytics，则必须指定以下信息：

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

包含延迟数据规则的配置信息的结构。

`deltaTimeSessionWindowConfiguration`

包含增量时间会话窗口的配置信息的结构。

DeltaTime指定时间间隔。您可以通过 `DeltaTime` 使用上次执行后到达数据存储的数据创建数据集内容。对于一个例子`DeltaTime`，请参阅创建具有增量时段的 SQL 数据集 (CLI)。.

`timeoutInMinutes`

一个时间间隔。您可以使用`timeoutInMinutes`以便Amazon IoT Analytics可以批量处理上次执行后生成的延迟数据通知。Amazon IoT Analytics向发送一批通知 CloudWatch 同时举办活动。

类型: 整数

有效范围：1-60

`ruleName`

延迟数据规则的名称。

类型: 字符串

Important

指定`lateDataRules`，数据集必须使用`DeltaTime`筛选条件。

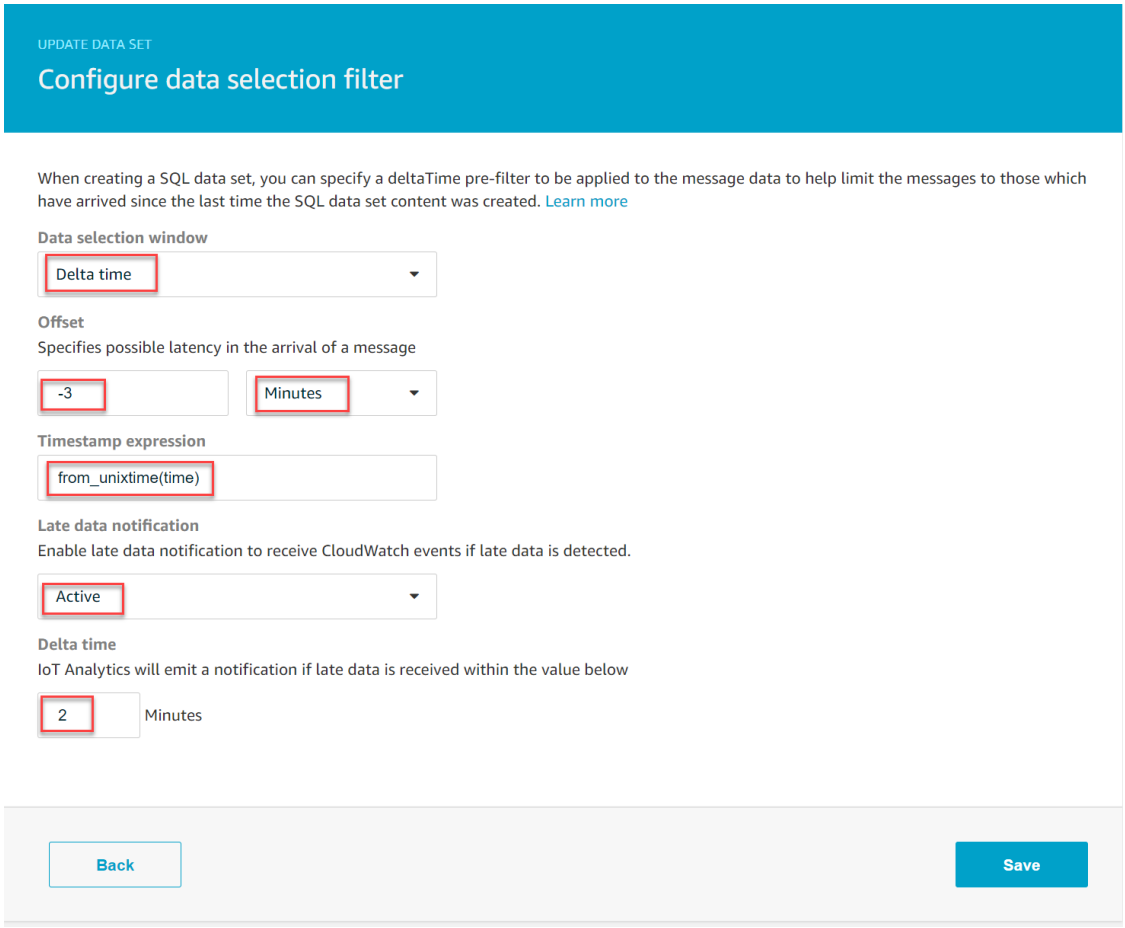
配置延迟数据规则（控制台）

以下过程显示如何配置数据集的延迟数据规则。Amazon IoT Analytics控制台。

配置迟期数据规则

1. 登录到 Amazon IoT Analytics 控制台。
2. 在导航窗格中，选择和。数据集.
3. 在数据集中，选择目标数据集。
4. 在导航窗格中，选择和。详细信息.
5. 在增量窗口部分，选择编辑.
6. 在配置数据选择过滤器中，执行以下操作：
   a. 适用于数据选择窗口，选择增量时间.
   b. 适用于Offset中，输入时间段，然后选择一个单位。
   c. 适用于Timestamp中，输入表达式。这可以是时间戳字段的名称，或可以推断时间的 SQL 表达式，例如：*`from_unixtime####`*.

      有关如何编写时间戳表达式的更多信息，请参阅。日期与时间函数和运算符中的Presto 0.172 文档.
   d. 适用于延迟数据通知，选择处于活动状态.
   e. 适用于增量时间中，输入整数。有效范围为 1-60。
   f. 选择 Save（保存）。



配置迟期数据规则 (CLI)

在Amazon IoT AnalyticsAPI，`LateDataRuleConfiguration`对象表示数据集的晚期数据规则设置。这个对象是`Dataset`与关联的对象`CreateDataset`和`UpdateDataset`. 您可以使用API、Amazon CLI，或者Amazon开发工具包以便为数据集指定晚期数据规则。以下示例使用 Amazon CLI。

要创建具有指定延迟数据规则的数据集，请运行以下命令。该命令假定`dataset.json`文件位于当前目录中。

> **Note**
>
> 您可以使用UpdateDataset更新现有数据集的 API。

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

这些区域有：`dataset.json`文件应包含以下内容：

- Replace*`demo_data`*使用目标数据集名称。
- Replace*`demo_data ##`*使用目标数据存储名称。
- Replace*`from_unixtime####`*具有时间戳字段的名称，或可推断时间的 SQL 表达式。

  有关如何编写时间戳表达式的更多信息，请参阅。日期与时间函数和运算符中的Presto 0.172 文档.

- Replace*##*整数介于 1—60 之间。
- Replace*demo_rules*有任何名字。

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

订阅接收延迟数据通知

您可以在中创建规则 CloudWatch 定义如何处理发送的延迟数据通知的事件Amazon IoT Analytics. 何时 CloudWatch 事件接收通知，它会调用规则中定义的指定目标操作。

创建的先决条件 CloudWatch 事件规则

在创建 CloudWatch 的事件规则Amazon IoT Analytics，则应执行以下操作：

- 熟悉中的事件、规则和目标。 CloudWatch 事件。
- 创建并配置目标由您调用 CloudWatch 事件规则。规则可以调用许多类型的目标，例如：
  - Amazon Kinesis Streams
  - Amazon Lambda 函数
  - Amazon Simple Notification Service (Amazon SNS) 主题
  - Amazon Simple Queue Service (Amazon SQS) 队列

  您的 CloudWatch 事件规则，并且关联的目标必须位于Amazon创建您的区域Amazon IoT Analytics资源的费用。有关更多信息，请参阅 。服务终端节点和配额中的Amazon一般参考.

有关更多信息，请参阅 。是什么 CloudWatch 事件？和开始使用亚马逊 CloudWatch 事件中的亚马逊 CloudWatch 事件用户指南.

延迟数据通知事件

延迟数据通知的事件采用以下格式。

```
{
 "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
 }
}
```

创建 CloudWatch 事件规则用于接收延迟数据通知

以下过程显示如何创建发布规则。Amazon IoT Analytics延迟发布数据通知到 Amazon SQS 队列。

创建 CloudWatch 事件规则

1. 登录到Amazon CloudWatch 控制台.
2. 在导航窗格中的事件下，选择规则。
3. 在存储库的Rule页面，选择。创建规则.
4. 在事件源，选择事件模式.
5. 在依次构建事件模式，以匹配服务。部分中，执行以下操作：
   a. 适用于服务名称，选择IoT Analytics
   b. 适用于事件类型，选择IoT Analytics 数据集生命周期通.
   c. 选择特定数据集名称，然后输入目标数据集的名称。
6. 在目标，选择添加目标 *.
7. 选择SQS 队列，然后执行以下操作：
   - 适用于队列 *中，选择目标队列。
8. 选择 Configure details (配置详细信息)。
9. 在存储库的步骤 2: 配置规则详细信息页面上，输入名称和描述。
10. 请选择 Create rule (创建规则)。

使用 Amazon IoT Analytics 记录 Amazon CloudTrail API 调用

Amazon IoT Analytics已与集成Amazon CloudTrail，提供用户、角色或用户所执行操作的记录的服务Amazon中的服务Amazon IoT Analytics. CloudTrail 捕获的 API 调用的子集Amazon IoT Analytics作为事件，包括来自Amazon IoT Analytics控制台和从代码调用到Amazon IoT AnalyticsAPI。如果您创建跟踪，则可以使 CloudTrail 发送到 Amazon S3 存储桶的事件，包括Amazon IoT Analytics. 如果您不配置跟踪，则仍可在 CloudTrail 控制台中事件记录. 使用 CloudTrail 收集的信息，您可以确定向 Amazon IoT Analytics 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅《Amazon CloudTrail 用户指南》。

Amazon CloudTrail 中的 Amazon IoT Analytics 信息

在您创建 Amazon 账户时，将在该账户上启用 CloudTrail。当活动发生在Amazon IoT Analytics，该活动将记录在 CloudTrail 活动以及其他Amazon中的服务事件事件记录. 您可以在 Amazon 账户中查看、搜索和下载最新事件。有关更多信息，请参阅 。使用查看事件 CloudTrail 事件记录.

要持续记录 Amazon 账户中的事件（包括 Amazon IoT Analytics 的事件），请创建跟踪。跟踪启用 CloudTrail 将日志文件传送到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有区域。此跟踪记录在 Amazon 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）存储桶。此外，您还可以配置其他Amazon服务，进一步分析在中收集的事件数据并采取措施。 CloudTrail 日志。有关更多信息，请参阅：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 接收 CloudTrail 来自多个区域的日志文件和接收 CloudTrail 来自多个账户的日志文件

Amazon IoT Analytics支持将以下操作记录为事件： CloudTrail 日志文件：

- 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

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management 用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 Amazon 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 Amazon IoT Analytics 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。 CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

以下示例显示了一个 CloudTrail 说明`CreateChannel`action.


```
{
"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"
}
```

以下示例显示了一个 CloudTrail 说明`CreateDataset`action.

```
{
"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"
}
```

Amazon IoT Analytics 的合规性验证

作为多个 Amazon 合规性计划的一部分，第三方审核员将评估 Amazon Web Services的安全性与合规性，例如 SOC、PCI、FedRAMP 和 HIPAA。

要了解此服务或其他 Amazon Web Services 是否在特定合规性计划范围内，请参阅合规性计划范围内的 Amazon Web Services。有关常规信息，请参阅Amazon Web Services合规性计划。

您可以使用 Amazon Artifact 下载第三方审计报告。有关更多信息，请参阅、在 Amazon Artifact 中下载报告。

您使用 Amazon Web Services 的合规性责任取决于您数据的敏感度、贵公司的合规性目标以及适用的法律法规。Amazon 提供以下资源来帮助满足合规性：

- 安全性与合规性快速入门指南安全性与合规性快速入门指南 – 这些部署指南讨论了架构注意事项，并提供了在Amazon上部署注重安全性和合规性的基准环境的步骤。
- Amazon Web Services 上的 HIPAA 安全性和合规性架构设计– 该白皮书介绍了公司如何使用 Amazon Web Services 创建符合 HIPAA 标准的应用程序。

  Note

  并非所有 Amazon Web Services都符合 HIPAA 要求。有关更多信息，请参阅符合 HIPAA 要求的服务参考。
- Amazon合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- Amazon Config 开发人员指南中的使用规则评估资源 – 此 Amazon Config 服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- Amazon Security Hub：此 Amazon Web Service 提供了 Amazon 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践规范。

Amazon IoT Analytics 中的故障恢复能力

Amazon全球基础设施围绕Amazon区域和可用区构建。Amazon区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 Amazon 区域和可用区的更多信息，请参阅 Amazon 全球基础设施。

Amazon IoT Analytics 中的基础设施安全性

作为一项托管服务，Amazon IoT Analytics受到Amazon全局网络安全程序，如中所述Amazon Web Services：安全过程概述白皮书。

您可以使用 Amazon 发布的 API 调用通过网络访问 Amazon IoT Analytics。客户端必须支持传输层安全性 (TLS) 1.0 或更高版本。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service (Amazon STS) 生成临时安全凭证来对请求进行签名。

Amazon IoT Analytics 配额

这些区域有：Amazon一般参考指南提供的默认配额Amazon IoT Analytics对于Amazonaccount. 每个配额针对的是每个配额针对Amazon区域。有关更多信息，请参阅 。Amazon IoT Analytics终端节点和配额和Amazon服务配额中的Amazon一般参考指南.

要请求提高服务配额，请在Support 中心控制台。有关更多信息，请参阅《Service Quotas 用户指南》中的请求增加配额。

Amazon IoT Analytics 命令

了解有关 API 操作的信息，请阅读此主题Amazon IoT Analytics，包括受支持的 Web 服务协议的示例请求、响应和错误。

Amazon IoT Analytics 操作

您可以使用Amazon IoT Analytics用于收集、处理、存储和分析 IoT 数据的 API 命令。有关更多信息，请参阅 。行动支持Amazon IoT Analytics中的Amazon IoT AnalyticsAPI 参考.

这些区域有：Amazon IoT Analytics部分中的Amazon CLI命令参考加入Amazon CLI您可以使用管理和操作的命令Amazon IoT Analytics.

Amazon IoT Analytics 数据

您可以使用Amazon IoT Analytics用于执行高级活动的 Data API 命令Amazon IoT Analytics `channel`、`pipeline`、`datastore`, 和`dataset`. 有关更多信息，请参阅 。数据类型支持Amazon IoT Analytics中的数据Amazon IoT AnalyticsAPI 参考.

排除 Amazon IoT Analytics 的故障

请参阅以下部分来解决错误并查找解决问题的可能解决方案Amazon IoT Analytics.

主题

如何知道我的消息是否正在进入？Amazon IoT Analytics?

检查是否正确配置了通过规则引擎将数据注入通道的规则。

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

该响应当与以下内容相似。

```
{
    "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"
```

```
    }
}
```

确保规则中使用的区域和通道名称正确。要确保数据进入了规则引擎并且正确执行了规则，您可能需要添加新目标，以在 Amazon S3 存储桶中临时存储传入消息。

为什么管道会丢失消息？ 如何修复此问题？

- 活动收到了无效的 JSON 输入：

  除了 Lambda 活动之外的所有活动，专门需要有效 JSON 字符串作为输入。如果活动收到的 JSON 无效，则将丢弃消息，并且消息不会传入数据存储。确保您将有效的 JSON 消息提取到服务中。在使用二进制输入时，确保管道中的第一个活动是将二进制数据转换为有效 JSON 的 Lambda 活动，然后再将其传递到下一个活动或者存储到数据存储中。有关更多信息，请参阅 。Lambda 函数示例 2.
- Lambda 活动调用的 Lambda 函数权限不足：

  确保对 Lambda 活动中的每个 Lambda 函数具有权限，可以从Amazon IoT Analytics服务。您可以使用以下命令：Amazon CLI命令授予权限。

  ```
  aws lambda add-permission --function-name <name> --region <region> --statement-id <id> --
  principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
  ```
- 不正确地定义了某个筛选条件或 removeAttribute 活动：

  如果有的话确保定义`filter`要么`removeAttribute`活动正确无误。如果您筛选出了某条消息或者从某条消息中删除了所有属性，该消息不会添加到数据存储。

为什么我的数据存储中没有任何数据？

- 数据提取与数据可用之间存在延迟：

  将数据提取到通道中之后，可能需要几分钟时间，然后数据才会在数据存储中可用。该时间因管道活动数以及管道中的任何自定义 Lambda 活动的定义而异。
- 管道正将消息筛选掉：

  确保您未在管道中丢弃消息。(请参阅前面的问题和答复。)
- 您的数据集查询不正确：

  确保从数据存储生成数据集的查询正确无误。从查询中删除任何不必要的筛选条件，确保数据可以进入数据存储。

为什么我的数据集只是显示__dt?

- 此列由服务自动添加，并包含数据的大致提取时间。它可用于优化查询。如果数据集中除此之外不含任何内容，请参阅前面的问题和答复。

如何编写由数据集完成操作驱动的事件代码？

- 您必须基于`describe-dataset`命令检查具有特定时间戳的数据集的状态是否为成功了.

如何正确配置笔记本实例以使用？Amazon IoT Analytics?

按照以下步骤确保您用于创建笔记本实例的 IAM 角色具有所需权限：

1. 转至 SageMaker 控制台并创建笔记本实例。
2. 填写详细信息，然后选择 create a new role (创建新角色)。记下角色的 ARN。
3. 创建笔记本实例。这还会创建一个角色： SageMaker 您可以使用。
4. 转到 IAM 控制台并修改新创建的 SageMaker 角色。当您打开该角色时，它应具有一个托管策略。
5. 单击添加内联策略，选择IoTAnalytics作为服务，然后在读取权限下，选择GetDatasetContent.
6. 检查策略，添加策略名称，然后对它选择 create (创建)。新创建的角色现在具有策略权限，可从中读取数据集。Amazon IoT Analytics.
7. 转到Amazon IoT Analytics控制台并在笔记本实例中创建笔记本。
8. 等待笔记本实例进入“In Service”(正在服务) 状态。
9. 选择 create notebooks (创建笔记本)，然后选择您创建的笔记本实例。这将使用可访问您的数据集的选定模板创建 Jupyter 笔记本。

为什么我无法在实例中创建笔记本？

- 确保您使用正确的 IAM 策略创建了笔记本实例。(按照上一个问题中的步骤操作。)
- 请确保笔记本实例处于“In Service”(正在服务) 状态。当您创建实例时，它开始处于 “挂起” 状态。通常大约需要 5 分钟进入“In Service”(正在服务) 状态。如果笔记本实例在大约 5 分钟之后进入 “Failed” (失败) 状态，请重新检查权限。

为什么我在 Amazon QuickSight 中看不到我的数据集？

> **Important**
>
> 亚马逊 QuickSight 此类商品在中国（北京）区域中不可用。有关支持的区域列表，请参阅亚马逊 QuickSight 终端节点和配额中的Amazon Web Services 一般参考.

亚马逊 QuickSight 可能需要许可才能阅读Amazon IoT Analytics数据集内容。要授予权限，请按照以下步骤操作。

1. 在亚马逊右上角选择您的账户名称 QuickSight 然后选择管理 QuickSight.
2. I在左侧导航窗格中，选择安全性和权限. UNDERQuickSight 访问Amazon服务，验证是否已授予访问权限Amazon IoT Analytics.
   a. 如果Amazon IoT Analytics选择没有访问权限，请选择添加或删除.
   b. 选择旁边的框Amazon IoT Analytics然后选择更新. 这给了亚马逊 QuickSight 读取数据集内容的权限。
3. 请重试以可视化您的数据。

确保选择相同的Amazon两者的地区Amazon IoT Analytics和 Amazon QuickSight。否则，您可能在访问Amazon资源的费用。有关支持的区域列表，请参阅Amazon IoT Analytics终端节点和配额和亚马逊 QuickSight 终端节点和配额中的Amazon Web Services 一般参考.

为什么我在现有的 Jupyter 笔记本上看不到容器化按钮？

- 这是由失踪造成的Amazon IoT Analytics容器化插件。如果您的 SageMaker 笔记本实例是在 2018 年 8 月 23 日之前创建的，则需按照中的说明手动安装插件。容器化笔记本.
- 如果您在创建后未看到容器化按钮 SageMaker 来自笔记本实例Amazon IoT Analytics控制台或手动安装它，请联系Amazon IoT Analytics技术支持。

为什么我的容器化插件安装失败？

- 通常，插件安装失败是因为 SageMaker 笔记本实例缺少权限。有关笔记本实例需要的权限，请参阅权限以及向笔记本实例角色添加所需权限。如果问题仍然存在，请从Amazon IoT Analytics控制台。
- 如果安装插件期间出现以下消息，您可以安全地忽略日志中的这一消息：“Ta notebook (或其他应用) 加载时，在浏览器中初始化此扩展。”

为什么我的容器化插件引发错误？

- 容器化失败并生成错误的原因有很多。在进行笔记本容器化之前，确保您使用的是正确的内核。容器化内核以“Containerized”前缀开头。
- 由于插件会在 ECR 存储库中创建并保存一个 Docker 映像，请确保您的笔记本实例角色有足够权限来读取、列出和创建 ECR 存储库。有关笔记本实例需要的权限，请参阅权限以及向笔记本实例角色添加所需权限。
- 此外，还要确保存储库的名称符合 ECR 要求。ECR 存储库名称必须以字母开头，并且只能包含小写字母、数字、连字符、下划线和正斜杠。
- 如果容器化过程失败并显示错误 “Tthis instance instance is run container erization” (此实例用于运行容器化的可用空间不足)，请尝试使用更大的实例来解决问题。
- 如果显示连接错误或映像创建错误，请重试。如果问题仍然存在，请重新启动实例并安装最新的插件版本。

为什么我在容器化期间看不到我的变量？

- 这些区域有：Amazon IoT Analytics容器化插件在运行具有 “容器化” 内核的笔记本后，可自动识别笔记本中的所有变量。使用其中一个容器化内核运行笔记本，然后执行容器化。

我可以将哪些变量作为输入添加到我的容器中？

- 您可以将要在运行时期间修改其值的任何变量作为输入添加到容器中。这样，您就可以使用需要在创建数据集时提供的不同参数来运行相同的容器。这些区域有：Amazon IoT Analytics容器化 Jupyter 插件通过在笔记本中自动识别变量并将其标记为可用来简化此过程。

如何将我的容器输出设置为后续分析的输入？

- 每次运行容器数据集时，都会创建一个可存储已执行构件的特定 S3 位置。要访问此输出位置，请在您的容器数据集中创建一个类型为 `outputFileUriValue` 的变量。该变量的值应该是用于存储其他输出文件

的 S3 路径。要在后续运行时访问这些已保存的构件，您可以使用`getDatasetContent`API 并选取后续运行所需的相应输出文件。

为什么我的容器数据集失败？

- 确保您正确传递的是正确的`executionRole`转到容器数据集。的信任政策`executionRole`必须包括两者`iotanalytics.amazonaws.com`和`sagemaker.amazonaws.com`.
- 如果您明白`AlgorithmError`作为失败原因，请尝试手动调试容器代码。如果容器代码中有错误或执行角色无权执行容器，则会出现这种情况。如果你通过使用Amazon IoT AnalyticsJupyter 插件，创建一个新的 SageMaker 笔记本实例与 containerDataset 的 executionRole 具有相同角色，请尝试手动运行该笔记本。如果容器是在 Jupyter 插件外创建的，请尝试手动运行代码并限制 executionRole 的权限。

文档历史记录

下表介绍了对Amazon IoT Analytics用户指南2020 年 11 月 3 日之后。如需获取对此文档的更新的更新的更多信息，您可以订阅 RSS 源。

update-history-change	update-history-description	update-history-date
地区启动	Amazon IoT Analytics 现已在亚太地区 (孟买) 提供。	2021 年 8 月 18 日
带有查询的JOIN	此更新使你能够使用JOIN查询Amazon IoT Analytics数据集。	2021 年 7 月 27 日
与 Amazon IoT SiteWise 集成	现在，您可以使用Amazon IoT Analytics查询Amazon IoT SiteWiseDATA。	2021 年 7 月 27 日
自定义分区	Amazon IoT Analytics现在通常支持根据消息属性或通过管道活动添加的属性对数据进行分区。	2021 年 6 月 14 日
重新处理通道消息	此更新使您能够重新处理指定 Amazon S3 对象中的频道数据。	2020 年 12 月 15 日
Parquet 模式	Amazon IoT Analytics数据存储现在支持 Publia 文件格式。	2020 年 12 月 15 日
使用 进行监控 CloudWatch 事件	Amazon IoT Analytics自动向亚马逊发布活动 CloudWatch 在 Amazon Lambda活动。	2020 年 12 月 15 日
延迟数据通知	您可以使用此功能通过亚马逊接收通知 CloudWatch 延迟数据到达时的事件。	2020 年 11 月 9 日
地区启动	推出Amazon IoT Analytics在中国（北京）。	2020 年 11 月 4 日

早期更新

下表介绍了对Amazon IoT Analytics用户指南2020 年 11 月 4 日之前。

更改	说明	日期
地区启动	推出Amazon IoT Analytics亚太地区（悉尼）区域。	2020 年 7 月 16 日
更新	重新组织了文档。	2020 年 5 月 7 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。