GUIDE NO. 4

AWS IoT Analytics

Analytics and Visualization

AWS IoT Analytics Mini-User Guide

Introduction

As IoT applications scale, so does the data generated from these various IoT devices. This data is raw, unstructured, and full of noise. It needs to be cleansed and translated into a language that can be easily interpreted in order to process that data to gain insights and intelligence quickly and efficiently. There are many common data management and analytics tasks across IoT applications, including processing and enriching data, provisioning and partitioning databases, and writing complex queries that need to be constantly updated as devices evolve, fleet sizes change, and new analytics requirements emerge. We've gone over these processes in the previous guides. This guide discusses analyzing and visualizing IoT data, both of which play a critical role in the overall IoT analytics architecture model.

AWS IoT Analytics can perform simple ad hoc queries as well as complex analysis, and is the easiest way to run IoT analytics for use cases, such as understanding the performance of devices, predicting device failures, and machine learning. IoT Analytics processes, stores, and performs advanced analysis on the massive amount of IoT data.

AWS IoT Analytics lets you schedule execution of your containers hosting custom authored analytical code or Jupyter Notebooks on a continuous basis. Customers have custom analyses in Matlab, Octave, R, Python, etc. and can automatically execute that analysis in their preferred scheduling windows. Customers can schedule execution and orchestration of stateful machine learning models.

IoT Analytics containerizes customers' data science tools and allows customers to execute their stateful analytical workflows at a recurring schedule to generate insights, giving customers more time to focus on their core competencies instead of operational improvements.

In this guide, we will go through the process of containerizing your Jupyter Notebooks-based analysis, scheduling the execution of analysis, and importing a simple SQL data set, created on the AWS IoT Analytics platform, to QuickSight. To see how to create a container data set, see the AWS IoT Analytics Mini-User Guide: Data Stores & Data Sets.

Adding Value to Your Business

Understand how your IoT devices are performing in near real time so you can respond quickly, preventing downtime, costly repairs, and even catastrophes.

Pre-built notebook templates that contain AWS-authored machine learning models and visualizations help you get started with IoT. Use cases related to device failure profiling, forecasting events such as low usage that might signal the customer will abandon the product, or segmenting devices by customer usage levels (for example heavy users, weekend users), or device health.

Access to high-performance, scalable, machine learning algorithms from Amazon SageMaker, provide you with the intelligence needed to build new services and business models, improve products and services over time, enjoy better relationships with your customers, and run your business more efficiently. Make intelligent decisions faster, and, over time, develop a data driven discipline in your company's culture.

What are Analytics and Visualization?

Analytics and visualization are important for:

1. Discovering patterns, trends, correlations and anomalies
2. Raising alerts
3. Building machine learning models
4. Improving communication between stakeholders
5. Driving capital gains

AWS IoT Analytics automates all the steps required to run analytics on IoT data. From the previous guides, we know that IoT Analytics offers an API so you can send data into the service from any source. It is fully integrated with AWS IoT Core so it is easy to collect data and begin performing analytics.

For review, first, you define a channel by using MQTT topic filters to specify only the data you want to store and analyze. Once the channel is set up, you configure a pipeline to process your data. The pipeline can perform data transformations, execute conditional statements, and enrich messages with data from external sources.

After processing the data, AWS IoT Analytics stores it in a time-series data store for analysis. Then, you can run ad hoc queries using the built-in SQL query engine to answer specific business questions or perform more sophisticated analysis and machine learning.

How to Gain Value

AWS IoT Analytics leverages the power of Amazon SageMaker to make it easy to apply machine learning to your IoT data with hosted Jupyter Notebooks, the industry standard and tool of choice for developers and data scientists. You can directly connect your IoT data to the notebook and build, train, and execute models at any scale right from the IoT Analytics console without having to manage any of the underlying infrastructure. You can also persist your Jupyter Notebooks and workspace for future use. Using AWS IoT Analytics, you can apply machine learning algorithms to your device data to produce a health score for each device in your fleet, prevent fraud and cyber intrusion by detecting anomalies on IoT devices, predict device failures, segment fleets of devices, and identify other rare events that may have great significance but are hard to find. For example, an auto manufacturer can detect which of their customers have worn brake pads and alert them to seek maintenance for their vehicles.

IoT Analytics provides a connector to Amazon QuickSight so you can visualize your data sets in a QuickSight dashboard. You can also visualize the results or your ad hoc analysis in the embedded Jupyter Notebooks within the IoT Analytics' console.

Run Ad Hoc SQL Queries

AWS IoT Analytics provides a built-in SQL query engine so you can run ad hoc queries and get results quickly. For example, you may want to run a quick query to find out how many monthly active users there are for each device in your fleet.

Time-Series Analysis

AWS IoT Analytics supports time-series analysis so you can analyze the performance of devices over time and understand how and where they are being used, continuously monitor device data to predict maintenance issues, and monitor sensors to predict and react to environmental conditions.

AWS IoT Analytics includes support for fully managed instances running Jupyter Notebooks from Amazon SageMaker, for training data exploration and preprocessing. IoT Analytics also includes a set of pre-built notebook templates that contain AWS-authored machine learning models and visualizations to help you get started with IoT use cases related to device failure profiling, forecasting events such as low usage that might signal the customer will abandon the product, or segmenting devices by customer usage levels (for example, heavy users vs. weekend users) or device health.

AWS IoT also provides access to the high-performance, scalable machine learning algorithms from Amazon SageMaker, optimized for speed, scale, and accuracy. You can do statistical classification through a method called logistic regression. You can also use Long-Short-Term Memory (LSTM) which is a powerful neural network technique for predicting the output or state of a process that varies over time. The pre-built notebook templates also support the K-means clustering algorithm for device segmentation, which clusters your devices into cohorts of like devices. These templates are typically used to profile device health and device state such as HVAC units in a chocolate factory or wear and tear of blades on a wind turbine.

Accessing AWS IoT Analytics

As part of AWS IoT, AWS IoT Analytics provides the following interfaces to interact with your devices and the data they generate:

AWS Command Line Interface (AWS CLI)
Run commands for AWS IoT Analytics on Windows, OS X, and Linux. These commands allow you to create and manage things, certificates, rules, and policies. To get started, see the AWS Command Line Interface User Guide.

AWS IoT API
Build your IoT applications using HTTP or HTTPS requests. These API actions allow you to create and manage things, certificates, rules, and policies.

Step by Step

Scheduling Execution of Analysis

AWS IoT Analytics uses DeltaTime Windows, a series of user-defined, non-overlapping and contiguous time intervals, to create data set contents with, and perform analysis on, new data that has arrived in the data store since the last analysis. IoT Analytics allows you to filter messages that have arrived during a specific time window, so the data contained in messages from previous time windows doesn't get counted twice.

You create a delta window by setting the DeltaTime in the filters portion of a queryAction of a data set. This allows you to filter messages that have arrived during a specific time window, so the data contained in messages from previous time windows doesn't get counted twice.

AWS IoT Analytics allows you to take input data from a data store and feed it into an automated workflow, using the following capabilities:

Create data set contents on a recurring schedule.
You can schedule the automatic creation of data set contents by specifying a trigger when you call CreateDataset (triggers:schedule:expression). Data which has arrived in a specified data store is used to create the data set contents, with the fields you want selected by a SQL query (actions:queryAction:sqlQuery).

You can also define a DeltaTime interval that ensures the new data set contents contain only that data which has arrived since the last time. Use the actions:queryAction:filters:deltaTime field to specify the DeltaTime interval. You will need to specify the trigger that creates the data set contents when the time interval has elapsed.

Create data set contents upon completion of another data set.
You can trigger creation of new data set contents upon completion of another data set contents creation (triggers:dataset:name).

Automatically run your analysis applications.
You can containerize your own, custom data analysis applications and trigger them to run upon creation of another data set contents. In this way, you can feed your application with data from data set contents created on a recurring schedule, and you can automatically take action on the results of your analysis from within your application. (actions:containerAction)

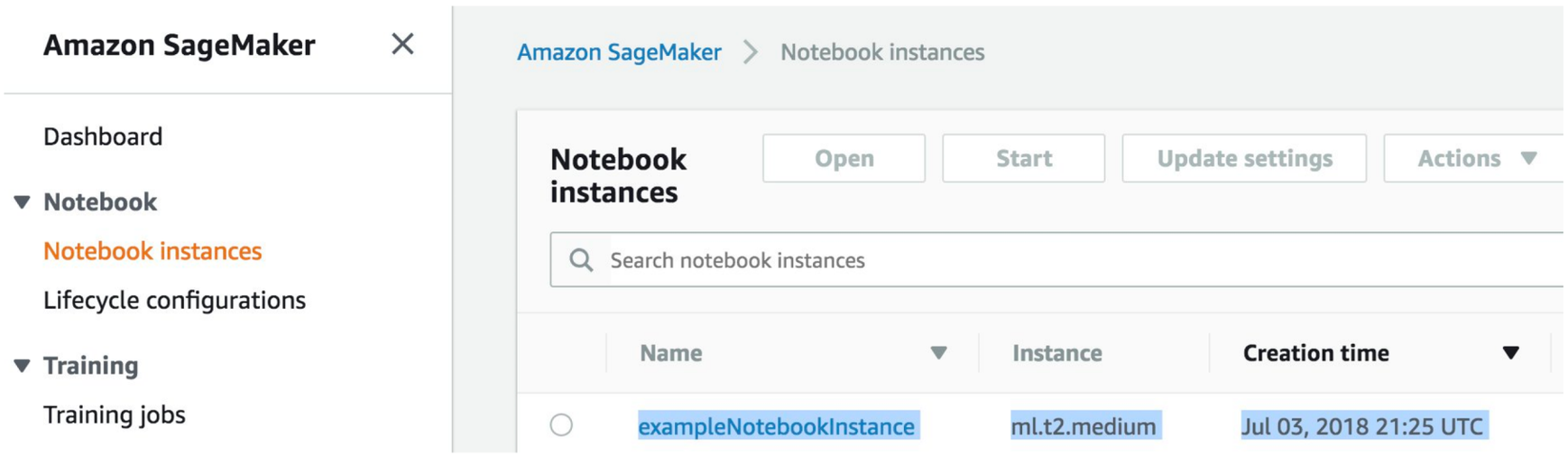
Once you have containerized your custom data analysis, you can set up periodic data analysis on data which has arrived since the last analysis was performed. See more information in the AWS IOT Analytics User Guide.

After you have a data set, you can explore and gain insights into your data through integration with Amazon QuickSight or you can perform more advanced analytical functions through integration with Jupyter Notebooks. Jupyter Notebooks provide powerful data science tools that can perform machine learning and a range of statistical analyses.

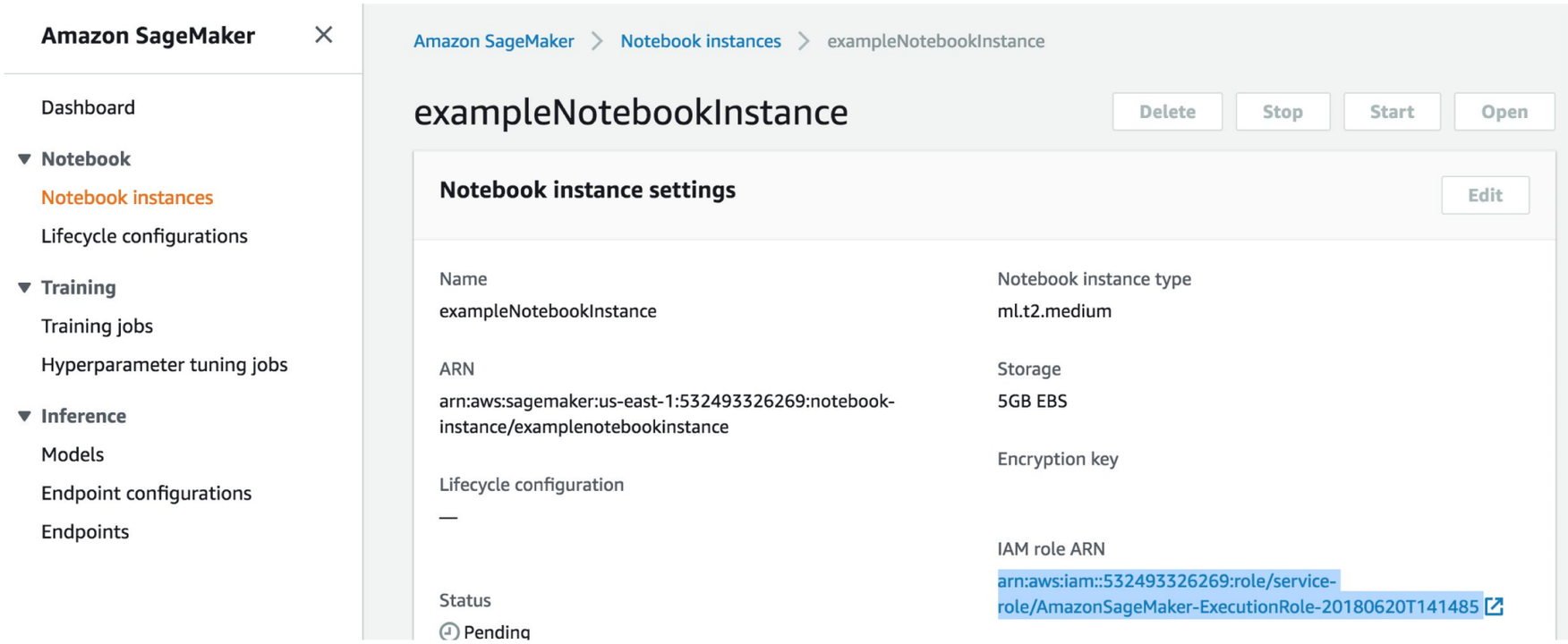
How to Containerize your Jupyter Notebooks-based Analysis

Enable Containerization Of Notebook Instances Not Created Via the AWS IoT Analytics Console

Step 1: To grant your notebook instance access to AWS ECS, select your Sagemaker instance on the Amazon Sagemaker page

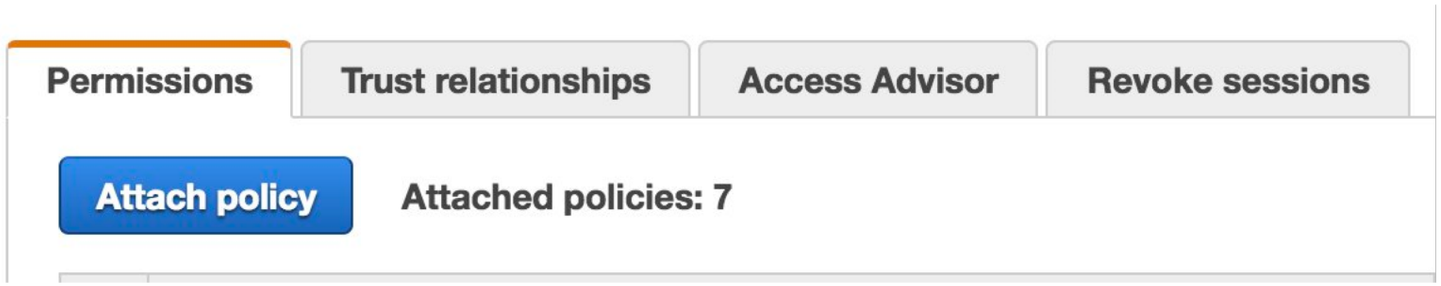


Step 2: Under IAM role ARN choose the Sagemaker Execution Role



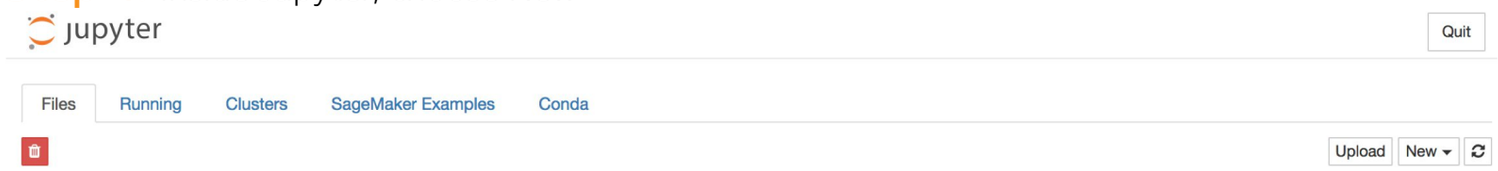
Step 3: Choose Attach Policy, then define and attach the policy shown in Permissions.

If the "AmazonSageMakerFullAccess" policy is not already attached, attach it as well.



You also must download the containerization code from S3 and install it on your notebook instance. The first step is to access the Sagemaker instance's terminal.

Step 4: Inside Jupyter, choose New



Step 5: In the dropdown menu that appears, select Terminal

Step 6: Inside the terminal, enter the following commands to download the code, unzip it, and install it

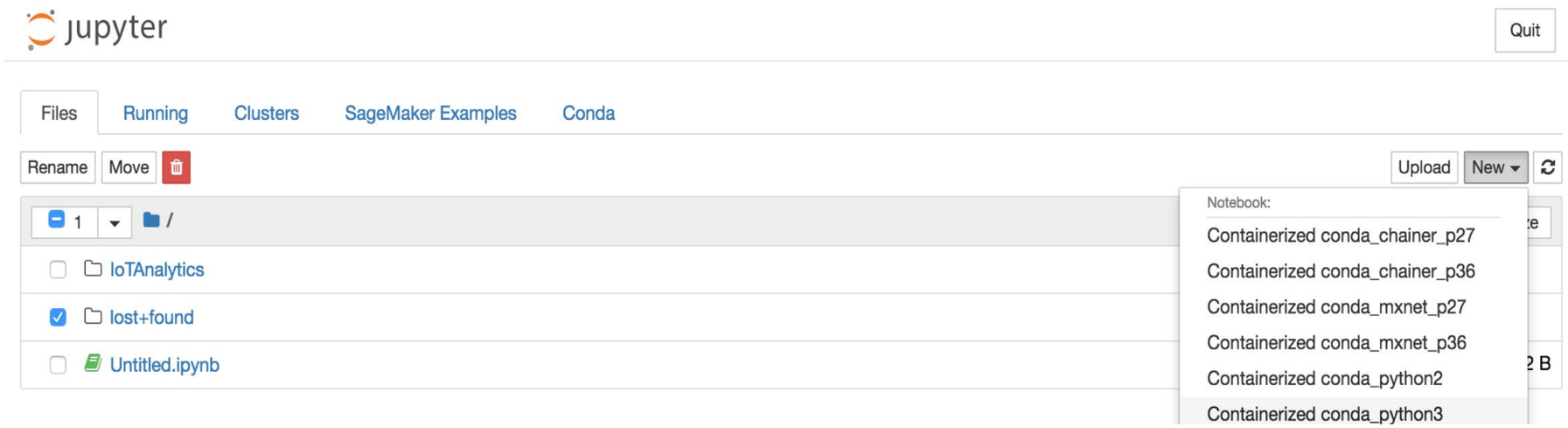

```
cd /tmp

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
```

Wait for a minute or two for the extension to be validated and installed.

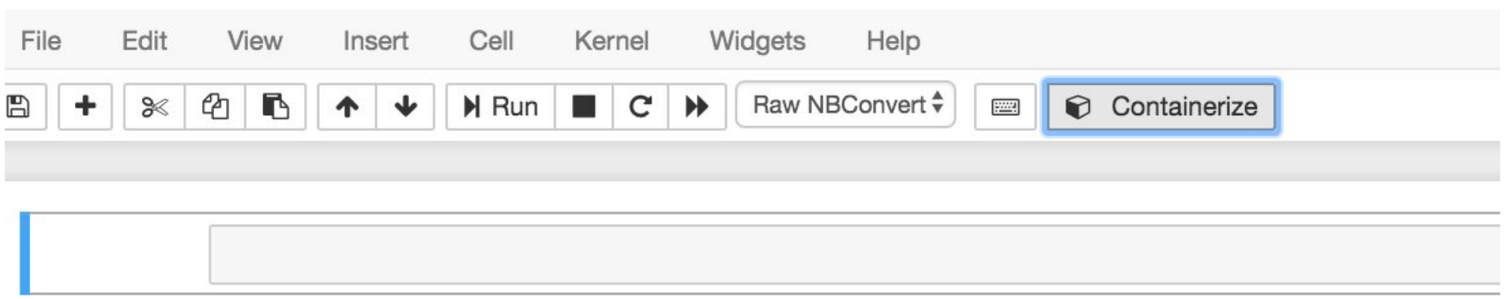
Create a Containerized Image

To containerize a notebook, go to your Jupyter Notebook and create a notebook with a containerized kernel.

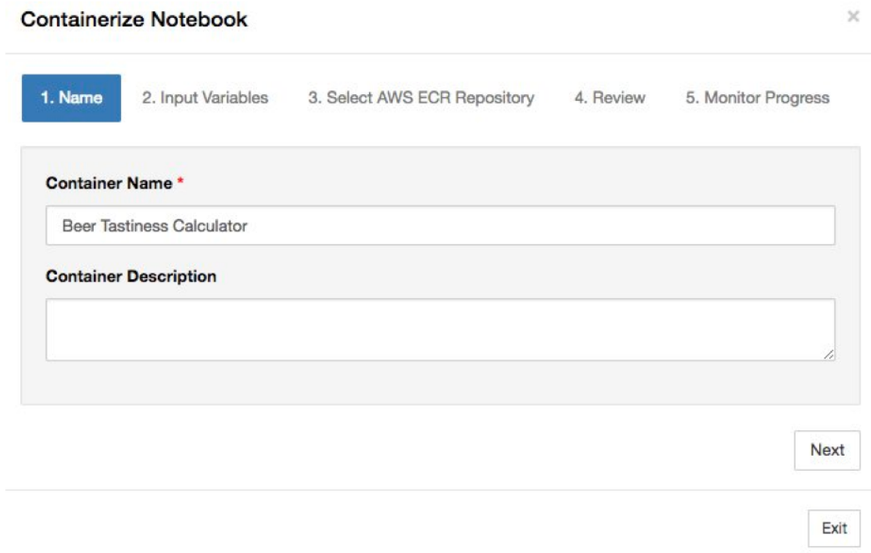
Step 1: In your Jupyter Notebook, choose New, then choose the kernel type you want from the dropdown list. (The kernel type should start with "Containerized" and end with whatever kernel you would have otherwise selected. For example, if you just want a plain python3 environment like "conda_python3", choose "Containerized conda_python3").



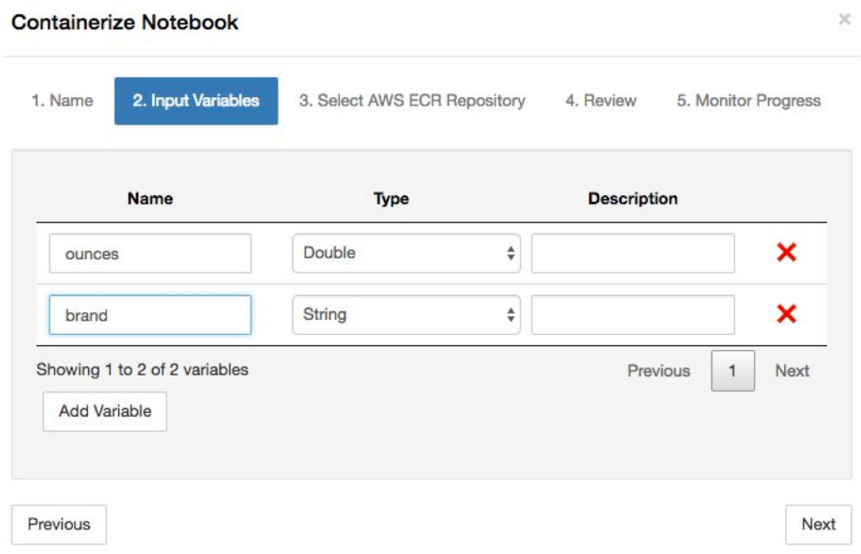
Step 2: After you have completed work on your notebook and you want to containerize it, choose the containerize button.



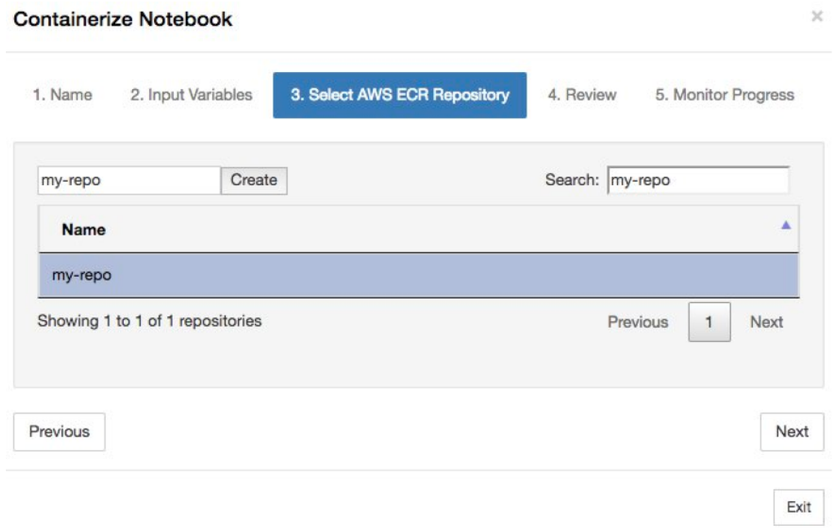
Step 3: Enter a name for the containerized notebook. You may also enter an optional description.



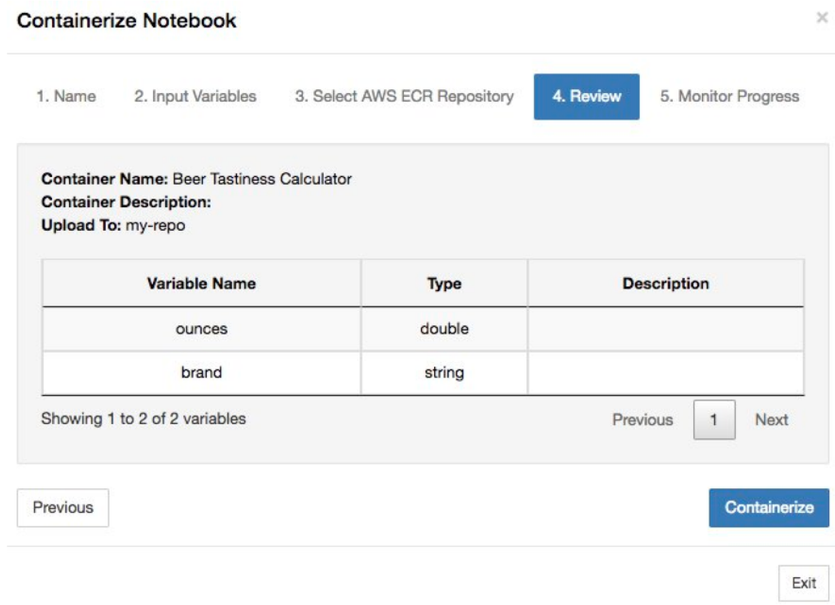
Step 4: Specify the Input Variables (parameters) that your notebook should be invoked with. You can select the input variables that are automatically detected from your notebook or define custom variables. (Note that input variables are only detected if you have previously executed your notebook.) For each input variable choose a type. You can also enter an optional description of the input variable.



Step 5: Choose the AWS ECR repository where the image created from the notebook should be uploaded:



Step 6: You are presented with with an overview summarizing your input. Note that after you have started the containerization process you cannot cancel it. The process may last for over an hour. Choose containerize to begin the containerization process.



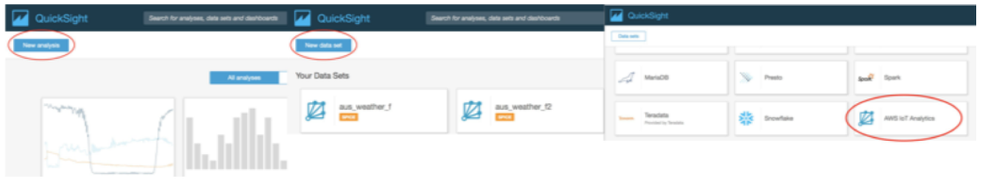
You can monitor the status of the containerization process from the Notebooks section of the IoT Analytics console. After the process is complete, the containerized image is stored on AWS ECR ready for use.

Importing AWS IoT Analytics Data Set to QuickSight

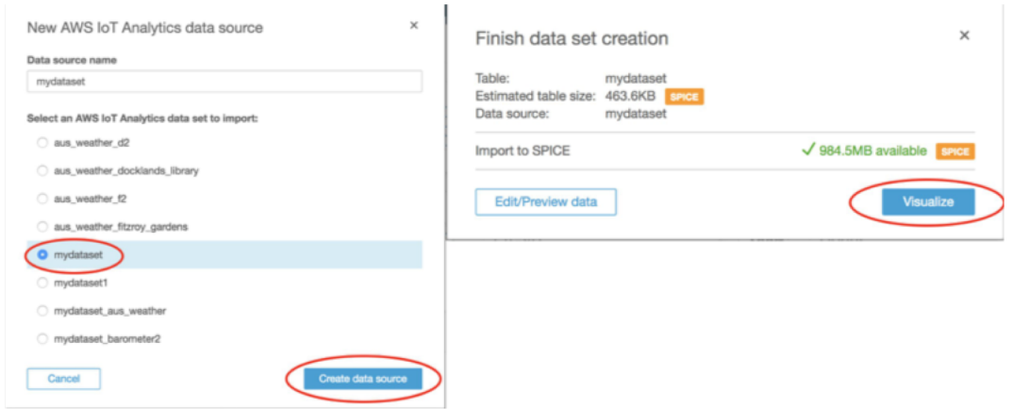
Assume that we created a data set in AWS IoT Analytics called myDataset. We created this data set using sensor devices that measure humidity, light, temperature and emit timestamps in (day, time) format with each measurement. The data set has 4 columns

1. humdity_avg (eg 45.3 %)
2. light_avg (eg 97.4 lms)
3. temp_avg (eg 21.3 C)
4. timestamp (eg 2014-12-15T01:45:00.000)

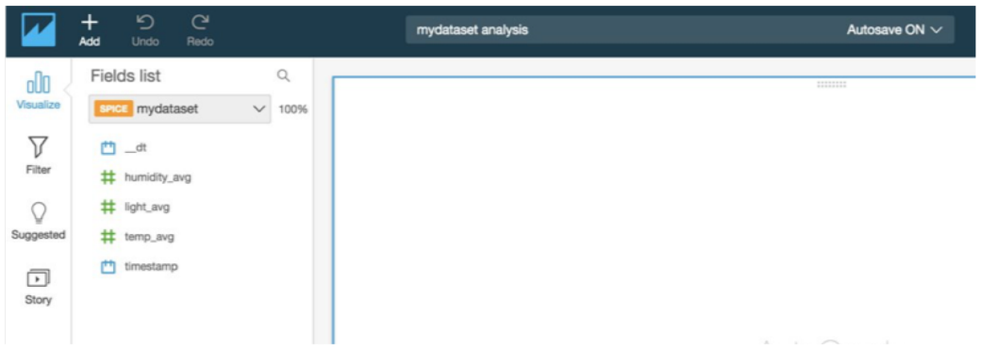
For information on how to create a data set, please refer to refer to the mini guide on Data Stores and Data Sets. We will now import this data set onto QuickSight. On QuickSight, click on New analysis, followed by New Dataset. Next, click on AWS IoT Analytics.



Choose myDataset and click on Create Datastore. On the next window click on Visualize.



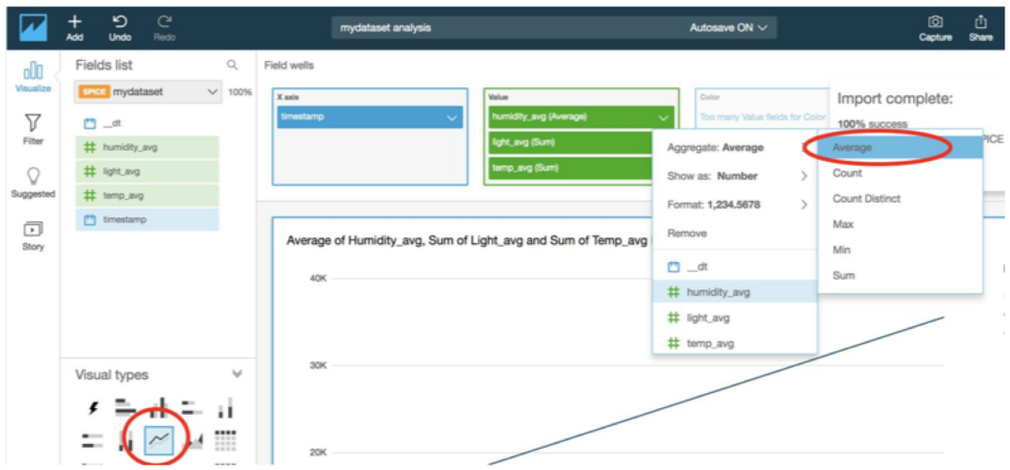
The QuickSight visualization window will appear as shown below:



We are now ready to visualize our data on QuickSight.

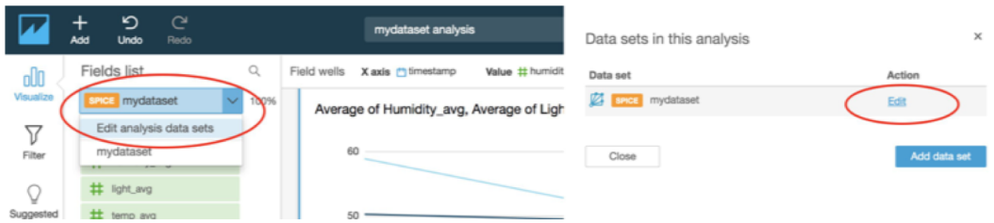
Visualizing Data on QuickSight

QuickSight has been designed with the objective of being as user-friendly as possible. Let us try to make a line plot on QuickSight with our data set. In this line plot we will plot the variation of our data (humidity_avg, light_avg and temperature_avg) with time (timestamp). Click on the line plot, located at the lower left-hand corner. Next drag the timestamp field onto the X axis at the top and the humdity_avg, light_avg, temp_avg field on the Value column as shown below. For each of the Value field, choose aggregation as average (default is sum).



Our visualization is ready. However, We can see that by default QuickSight plots our timestamp on a day granularity as shown below:

IoT data is usually in minute or even second granularity. Our timestamp field is in date format (mm/dd/yyyy HH:MM:SS). Unfortunately QuickSight doesn't allow us to by default change the granularity of the X axis for a date formatted field. We can however change the format of timestamp from date to string and then try plotting. To do this, click on the myDataset name in the left-hand corner. A pop up window appears. Click Edit analysis data sets. On the new window, click on edit next to myDataset:



We can now change the format of the timestamp format from date to string in the edit window as shown below:

Click on Save and visualize at the top. It will now have the timestamp in string (mm/dd/yyyyTHH:MM:SS) format. Below the visual, there is a blue window which can be scrolled from left to right in order observe the variation of data with time. The window can also be expanded to increase the timespan of the visual:

Conclusion

This mini-user guide serves as a stepping stone for readers to set up DeltaTime Windows to create data set contents with, and perform analysis on, new data that has arrived in a data store since the last analysis. It also covers the creation of a Jupyter Notebook. AWS IoT Analytics allows you to bring your customer created code, find out more in our IoT Analytics Documentation. Once you have captured your time-series data, visualize it on QuickSight. There are other features not discussed here that are provided by QuickSight for better visualization. For more information about QuickSight tools, please visit QuickSight documentation.

Easily analyze data for deeper insights to make better, more accurate decisions for IoT applications and machine learning use cases. With AWS IoT Analytics, you can collect, pre-process, enrich, store and analyze your IoT data.

Start using Analytics and Visualization with AWS IoT Analytics in minutes:

aws.amazon.com/iot-analytics