**<https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/README.md>**

**DAI-DP 302 Unlocking the IoT promise- Design and architect solutions with real-world impact**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/iothub.jpg)

Azure IoT Hub is a fully managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a solution back end. Azure IoT Hub:

* Provides multiple device-to-cloud and
* cloud-to-device communication options. These options include one-way messaging, file transfer, and request-reply methods.
* Provides built-in declarative message routing to other Azure services.
* Provides a queryable store for device metadata and synchronized state information.
* Enables secure communications and access control using per-device security keys or X.509 certificates.
* Provides extensive monitoring for device connectivity and device identity management events.
* Includes device libraries for the most popular languages and platforms.

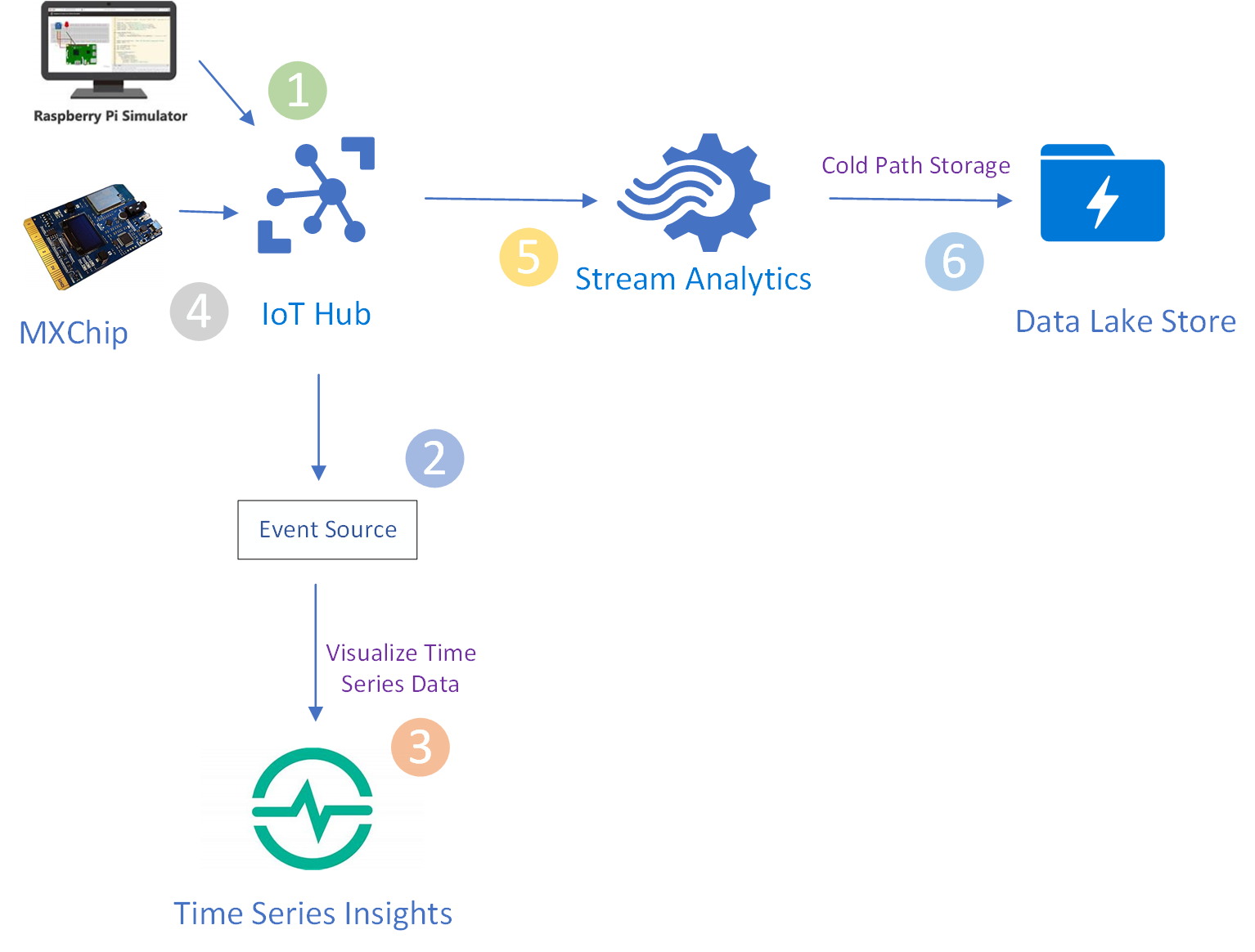
**IoTHub: Connect, monitor, and manage billions of IoT assets**

* **Establish** bi-directional communication with billions of IoT devices
* **Authenticate** per device for security-enhanced IoT solutions
* **Register** devices at scale with IoT Hub Device Provisioning Service
* **Manage** your IoT devices at scale with device management
* **Extend** the power of the cloud to your edge device

<iframe src="<https://channel9.msdn.com/Shows/Azure-Friday/Azure-IoT-Hub/player>" width="480" height="270" allowFullScreen frameBorder="0"></iframe>

**In this lab you will**

* Learn to Create IoT Hub
* Learn to use Simulator to connect to IoT Hub and send Data
* Learn to setup MXChip, connect to IoT Hub and send data

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/Lab1.png)

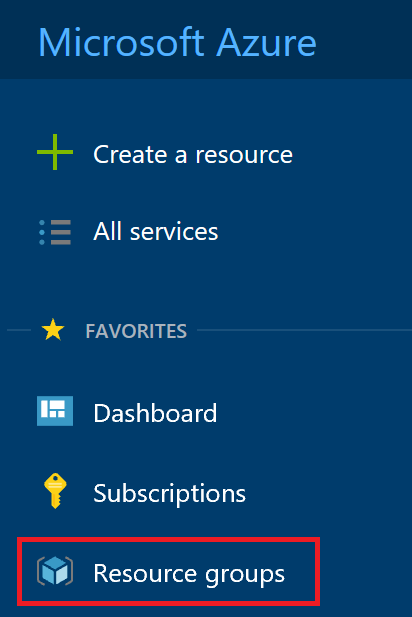
**Create Resource Group**

The infrastructure for your application is typically made up of many components – maybe a virtual machine, storage account, and virtual network, or a web app, database, database server, and 3rd party services.

You do not see these components as separate entities, instead you see them as related and interdependent parts of a single entity. You want to deploy, manage, and monitor them as a group. Azure Resource Manager enables you to work with the resources in your solution as a group. You can deploy, update, or delete all the resources for your solution in a single, coordinated operation.

You use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment.

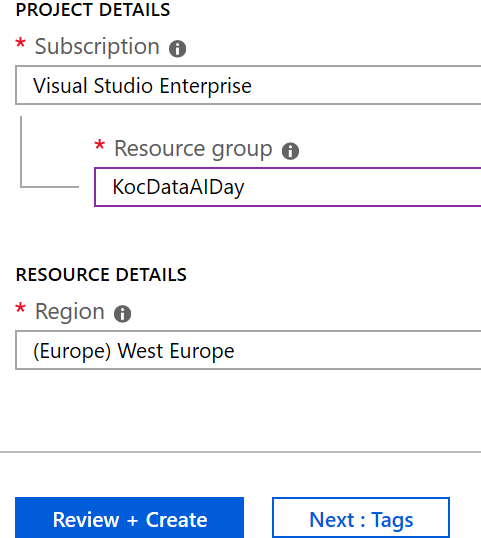
Create a resource group to collect and manage all your application resources for this lab

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/01_Create_Resource_Group.png)

Click on **+ Add** button

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/02_Create_Resource_Group_Create.png)

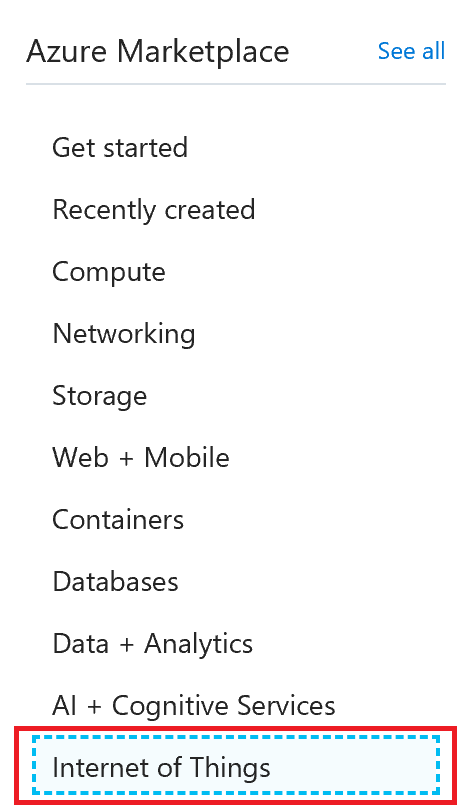
Enter **Resource group name**, Select **subscription** and **region**



**Create IoThub**

Create an IoT Hub to connect your real device or simulator to this IoTHub and start sending data.

Click on **Create a resource** and click on **Internet of Things**

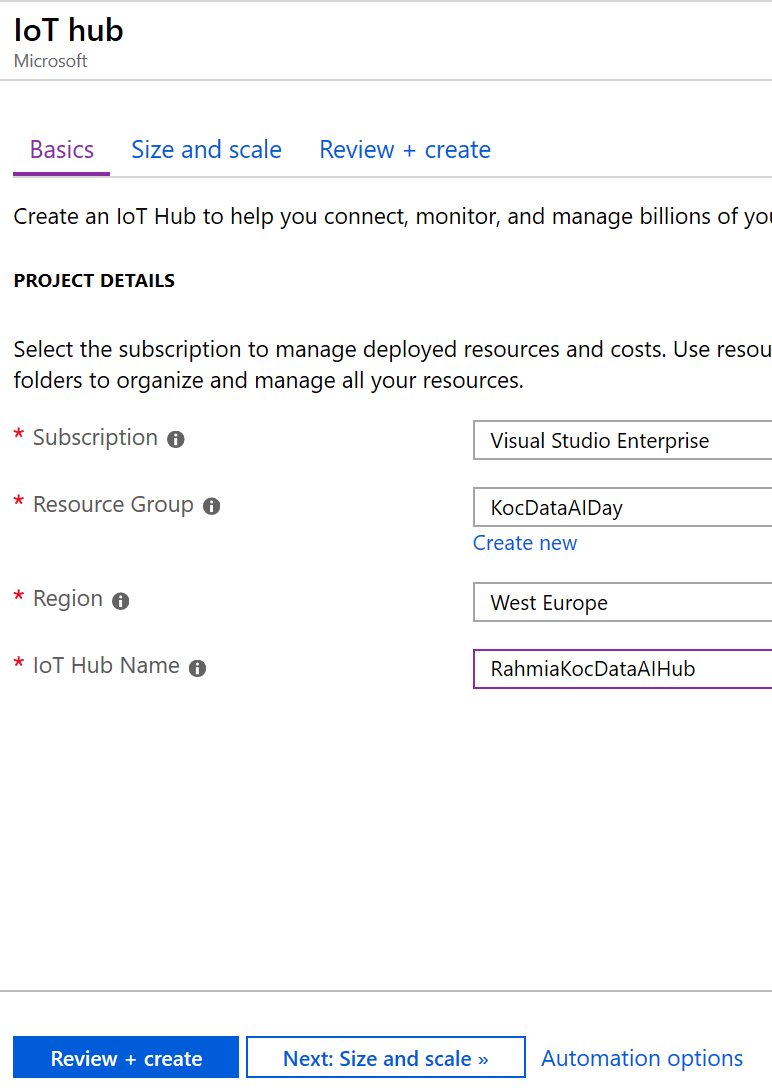
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/iot.png)

Click on **IoTHub**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/04_Create_IoTHub.png)

Make sure you select the resource group you created in previous step.

In the Name field, enter a unique name for your IoT hub. The name of your IoT hub must be **unique** across all IoT hubs. For this lab chose **West Europe**.



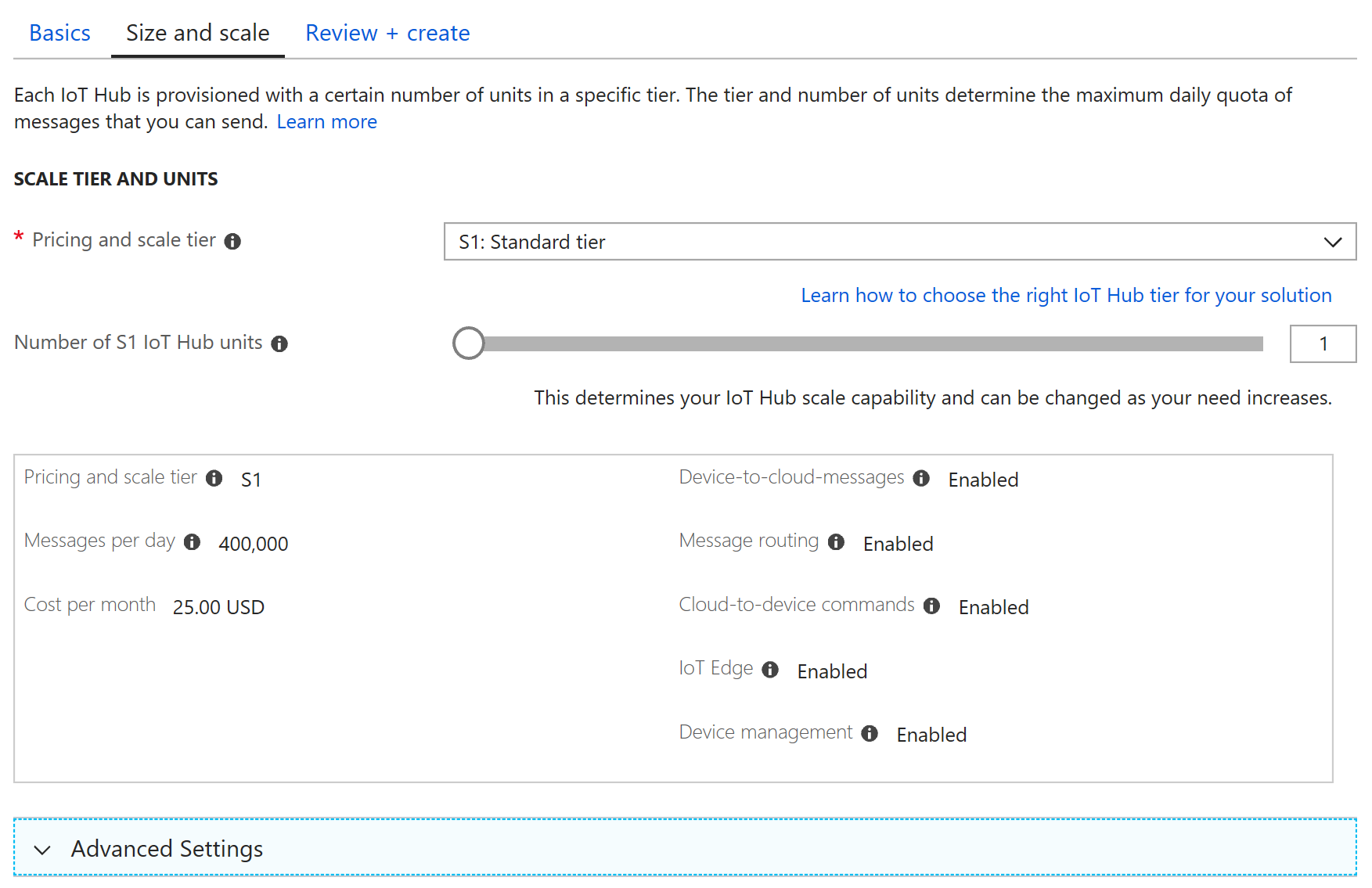
Click **Size and Scale** button.

In the Tier filed, select **S1 tier**.

You can choose from several tiers depending on how many features you want and how many messages you send through your solution per day. The free tier is intended for testing and evaluation. It allows 500 devices to be connected to the IoT hub and up to 8,000 messages per day. Each Azure subscription can create one IoT Hub in the free tier.

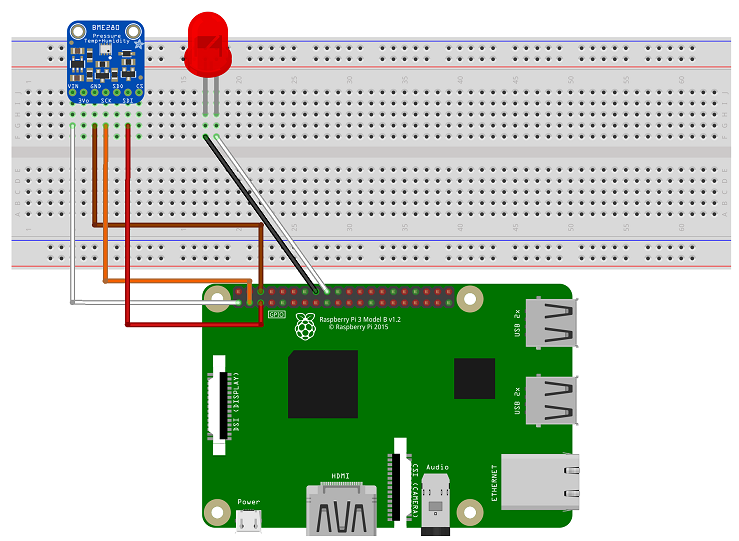
The **S1** tier allows total of 400,000 messages per unit per day.

For details about the other tier options, see [Choosing the right IoT Hub tier](https://azure.microsoft.com/en-us/pricing/details/iot-hub/).

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/05_Create_IoTHub_Submit_2_review.png)

Click **Review and Create** button. Review the shown configuration and clicl the **Create** button

**Connect PI Simulator to IoT Hub**

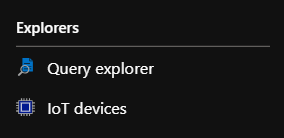
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/pi_simulator.png)

Connect a Simulator to your IoT Hub and stream data.

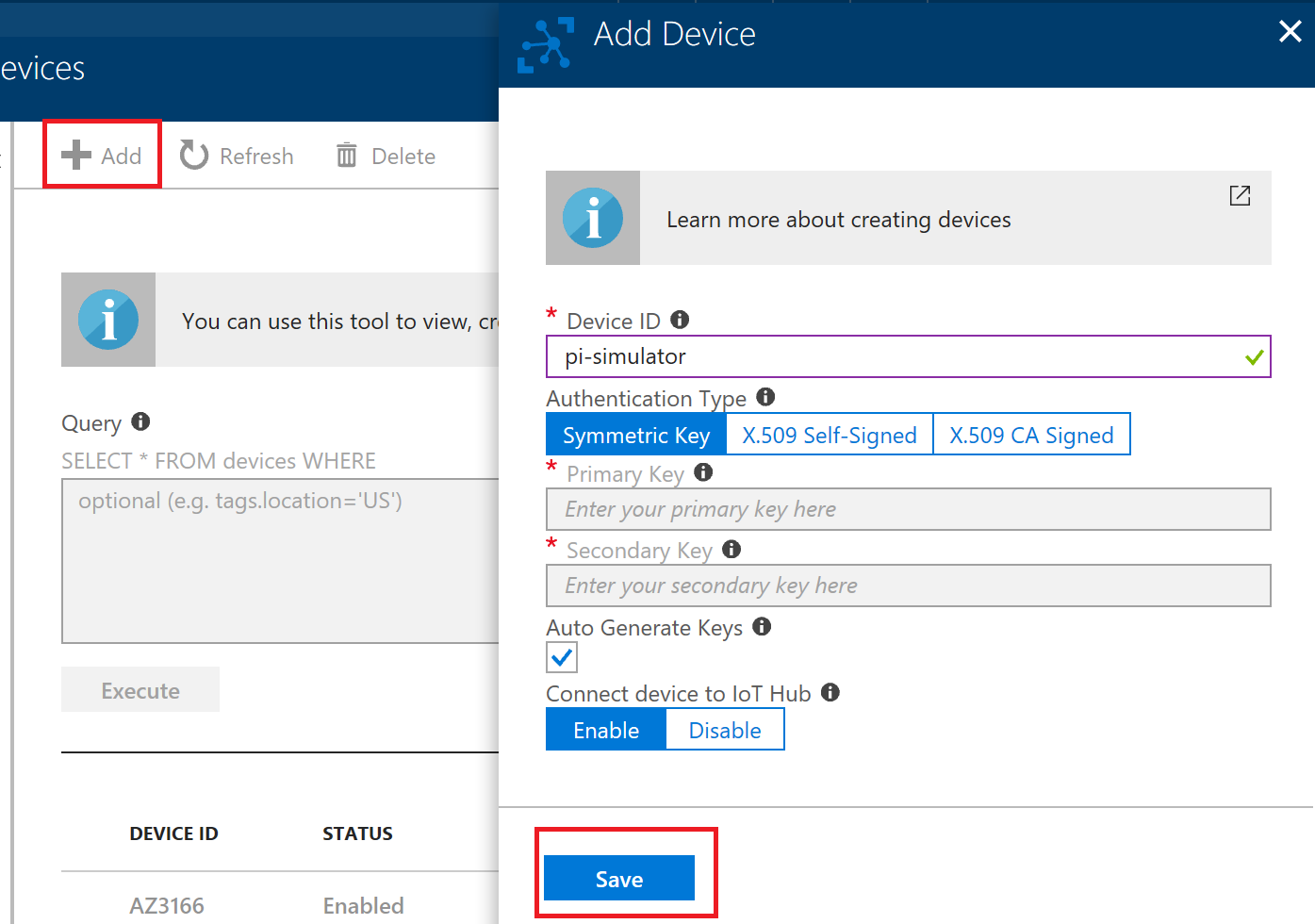
* Learn to create a device using Azure Portal
* Connect the simulator to IoT Hub
* Send telemetry data to Azure

**Create a Device**

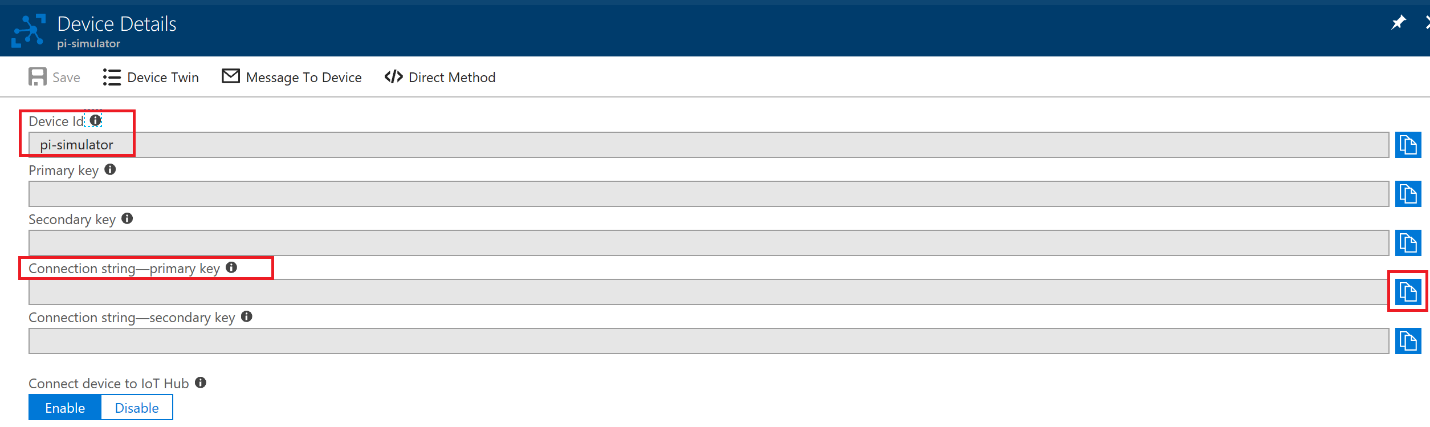
Go To your IoT Hub in the portal and click on **IoT Devices**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/iot_devices.png)

Click on **+ Add** and enter a **Device ID** and click **Save**.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/add_device.png)

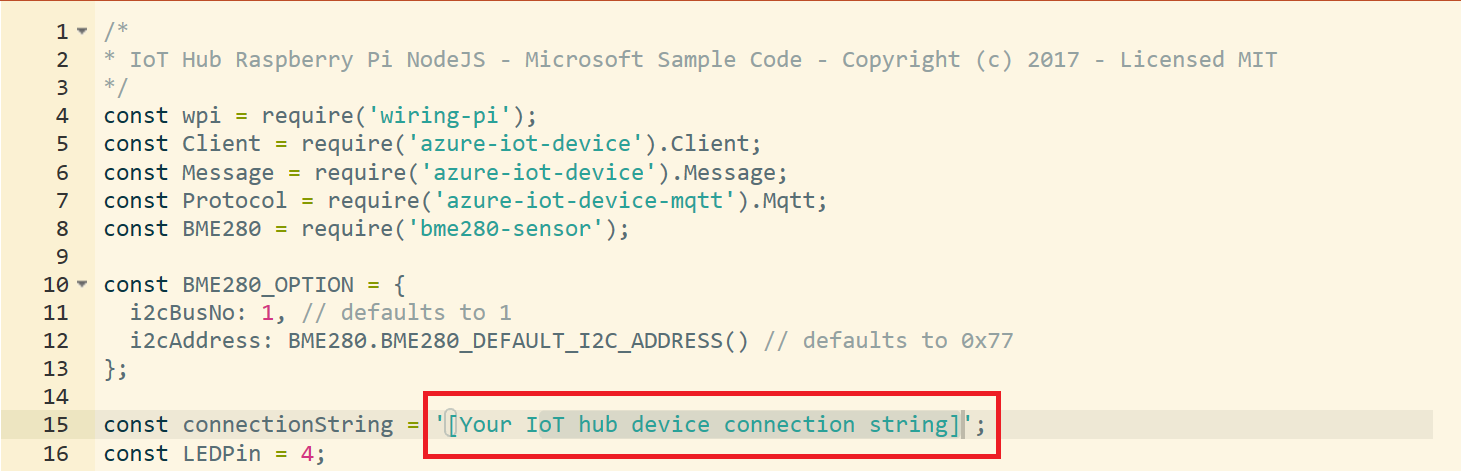
Click on the device and copy the primary key connection string.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/connection-string.png)

Open below link on a new window to go to the PI Simulator

[PI Simulator](https://azure-samples.github.io/raspberry-pi-web-simulator/#GetStarted)

Replace the connection string with the primary key connection string copied in the previous steps

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/pi_connection_string_before.png)

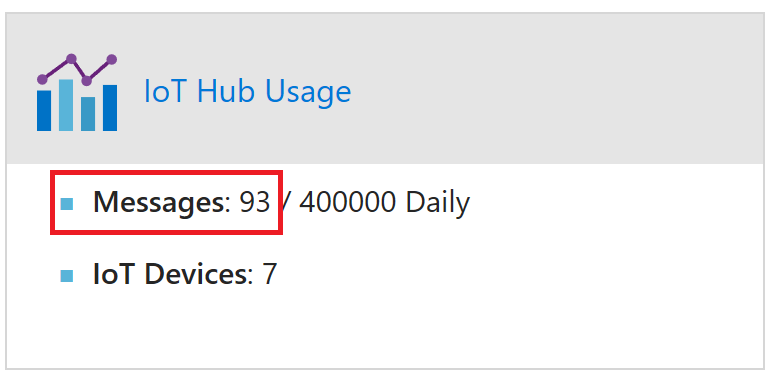
After you copy the connection string should look like below

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/pi_connection_string_after.png)

Click Run and start sending messages. LED will start blinking

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/pi_message.png)

Messages will start flowing into IoT Hub

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/iothub_messages.png)

**You will Visualize the Data flowing into IoT Hub by connecting to Time Series Insights**

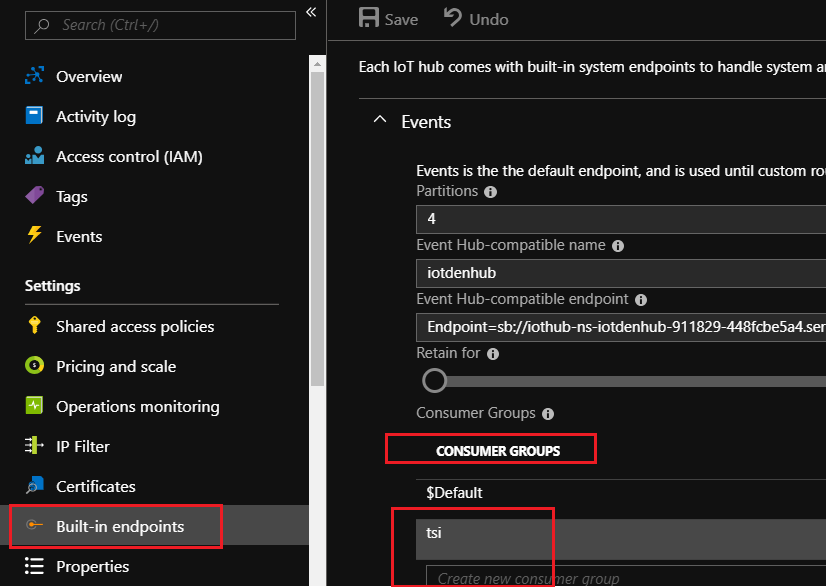
**Create Consumer Groups to Route Data**

Under IoTHub, Settings click on **Build-in endpoints**

Expand **Events** and go to **CONSUMER GROUPS**

Create two consumer groups for

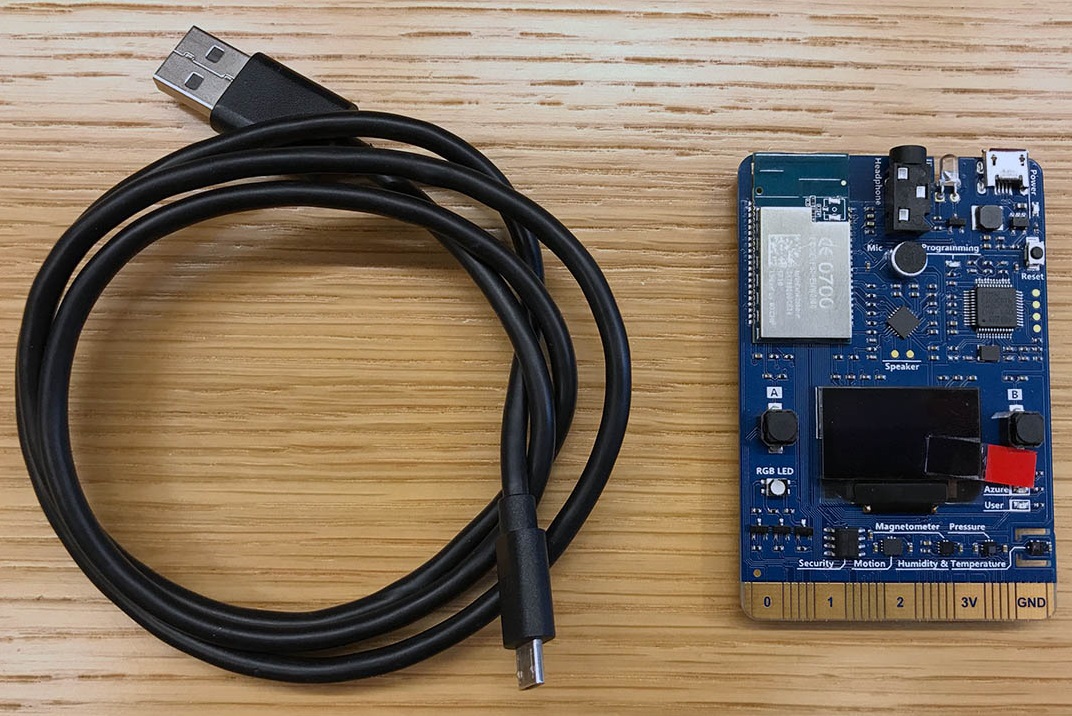
* Time Series Insights *(name can be anything, ie we chose tsi in below image)*
* Stream Analytics

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/customEndpoint.png)

**Connect Device and Send Data to IoThub**

This Lab assumes you are using MXChip as the Device.

If you do not have MXChip, you can open the [PI Simulator](https://azure-samples.github.io/raspberry-pi-web-simulator/#GetStarted) again in a different window, edit the deviceId: 'Raspberry Pi Web Client' (line 29) part to give a new name to your device, and create a new device twin on IoT Hub, and update the **connectionString** with the primary key info of the new devide twin and run.

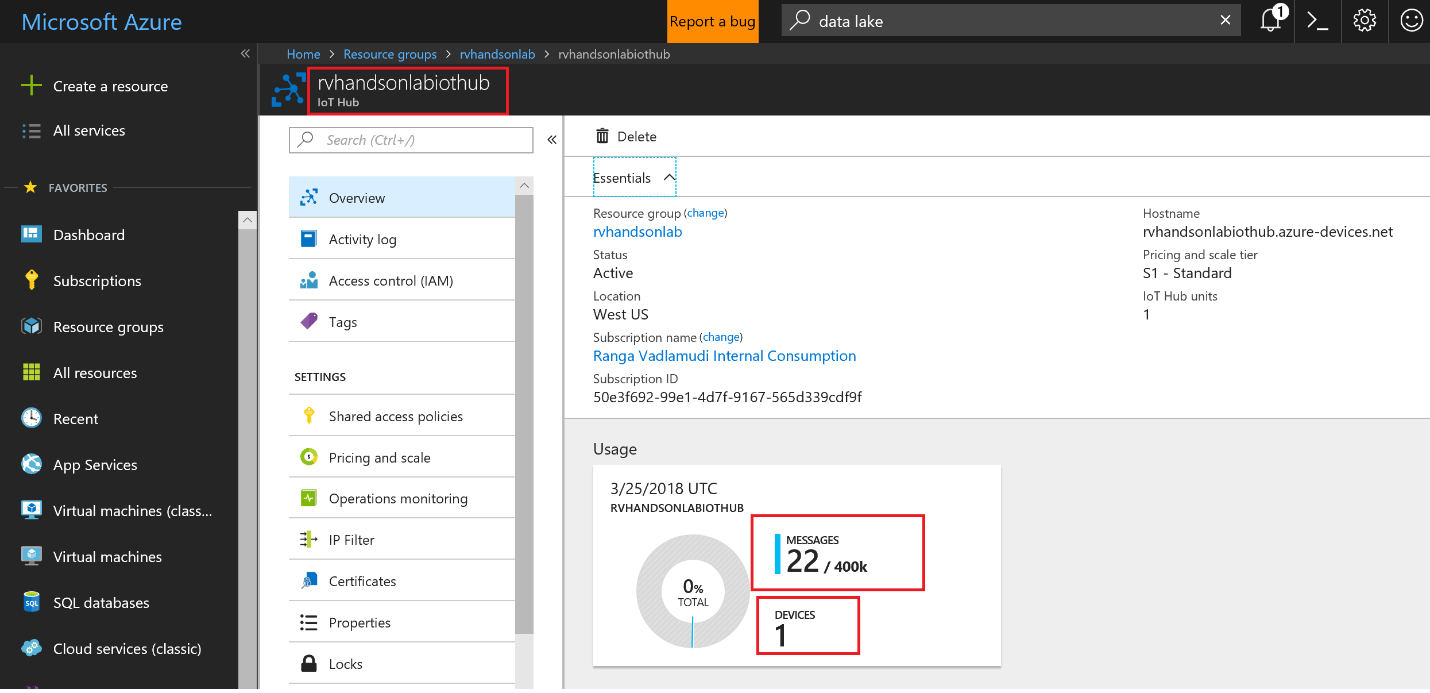
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/MxChip.jpg)

**Prepare the MXChip by**

* updating firmware
* connecting to Wifi
* connecting to Azure to select a subscription and IoTHub
* uploading device code

[Prepare MXChip to Connect to IoTHub](https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-arduino-iot-devkit-az3166-get-started)

Once Device Connects to IoTHub, messages flow into IoThub

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/06_IoTHub_DeviceCreated_Data_Flowing.png)

**Create Azure Time Series Insights and Visualize Device Data**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/timeseriesinsights.jpg)

**Create Time Series Insights**

Azure Time Series Insights is a fully managed analytics, storage, and visualization service for managing IoT-scale time-series data in the cloud. It provides massively scalable time-series data storage and enables you to explore and analyze billions of events streaming in from all over the world in seconds. Use Time Series Insights to store and manage terabytes of time-series data, explore and visualize billions of events simultaneously, conduct root-cause analysis, and to compare multiple sites and assets.

Time Series Insights has four key jobs:

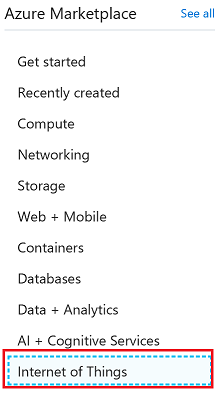
* First, it's fully integrated with cloud gateways like Azure IoT Hub and Azure Event Hubs. It easily connects to these event sources and parses JSON from messages and structures that have data in clean rows and columns. It joins metadata with telemetry and indexes your data in a columnar store.
* Second, Time Series Insights manages the storage of your data. To ensure data is always easily accessible, it stores your data in memory and SSDâ€™s for up to 400 days. You can interactively query billions of events in seconds â€“ on demand.
* Third, Time Series Insights provides out-of-the-box visualization via the TSI explorer.
* Fourth, Time Series Insights provides a query service, both in the TSI explorer and by using APIs that are easy to integrate for embedding your time series data into custom applications.

<iframe src="<https://channel9.msdn.com/Shows/Internet-of-Things-Show/Time-Series-Insight-for-IoT-apps/player>" width="480" height="270" allowFullScreen frameBorder="0"></iframe>

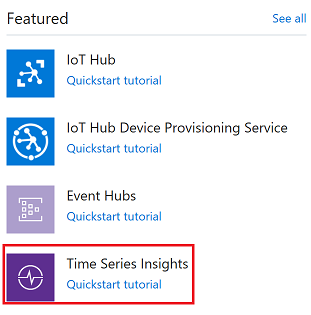
In this lab you will learn

* how to set up a Time Series Insights environment
* explore
* analyze time series data of your IoT solutions or connected things

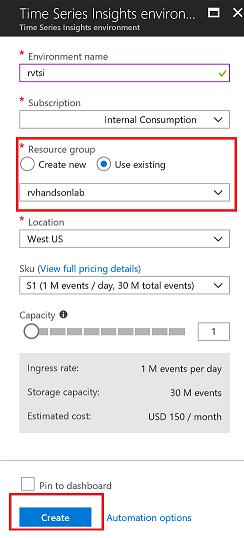
Click on **Create a Resource** and click on **Internet of Things**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/01_Create_Time_Series_Insights.png)

Click on **Time Series Insights**

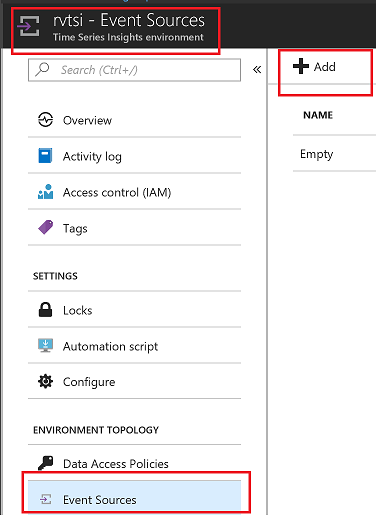
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/tsi.png)

Select the resource group you previously created and click **Create** button

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/02_Create_Time_Series_Inisghts_Submit.png)

**Create Event Source**

Create Event Source to connect to IoTHub. Please make sure you use a unique Consumer Group. Time Series Insights has a requirement to have its own unique consumer group

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/03_Create_Event_Source.png)

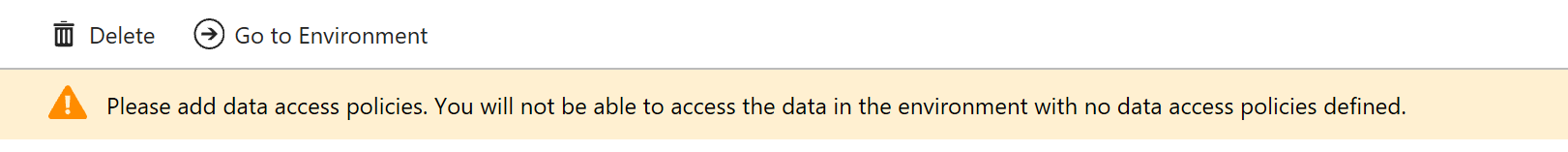
Select the appropriate consumer group and click Create button

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/04_Create_Event_Source_Submit.png)

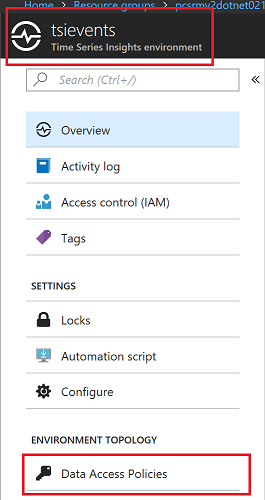
**Setup Time Series Insights**

Click on **Overview** to go to main page and click on Go To Environment which will take you to Time Series Insights Explorer

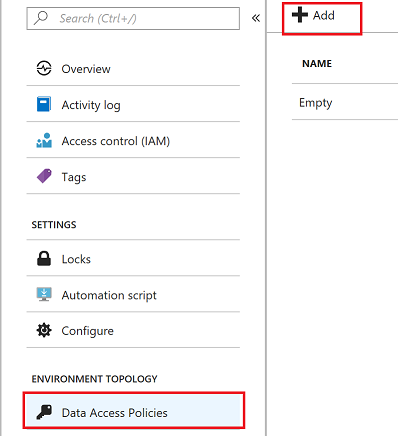
If you get Data Access Policy Error execute the following steps

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/16_data_access_poliy_error.png)

Go To Environment Topology and

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/15_data_access_policy.png)

Click on Add Button

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/17_add_user_role.png)

Select Contributor Role

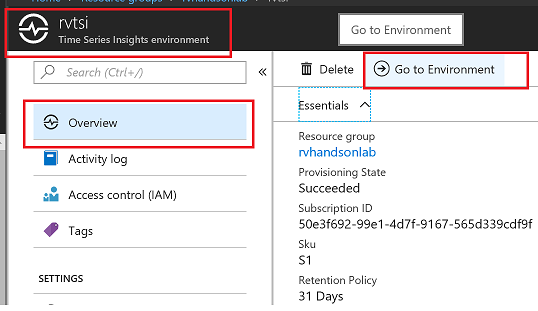
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/18_select_controbutor_role.png)

Select User

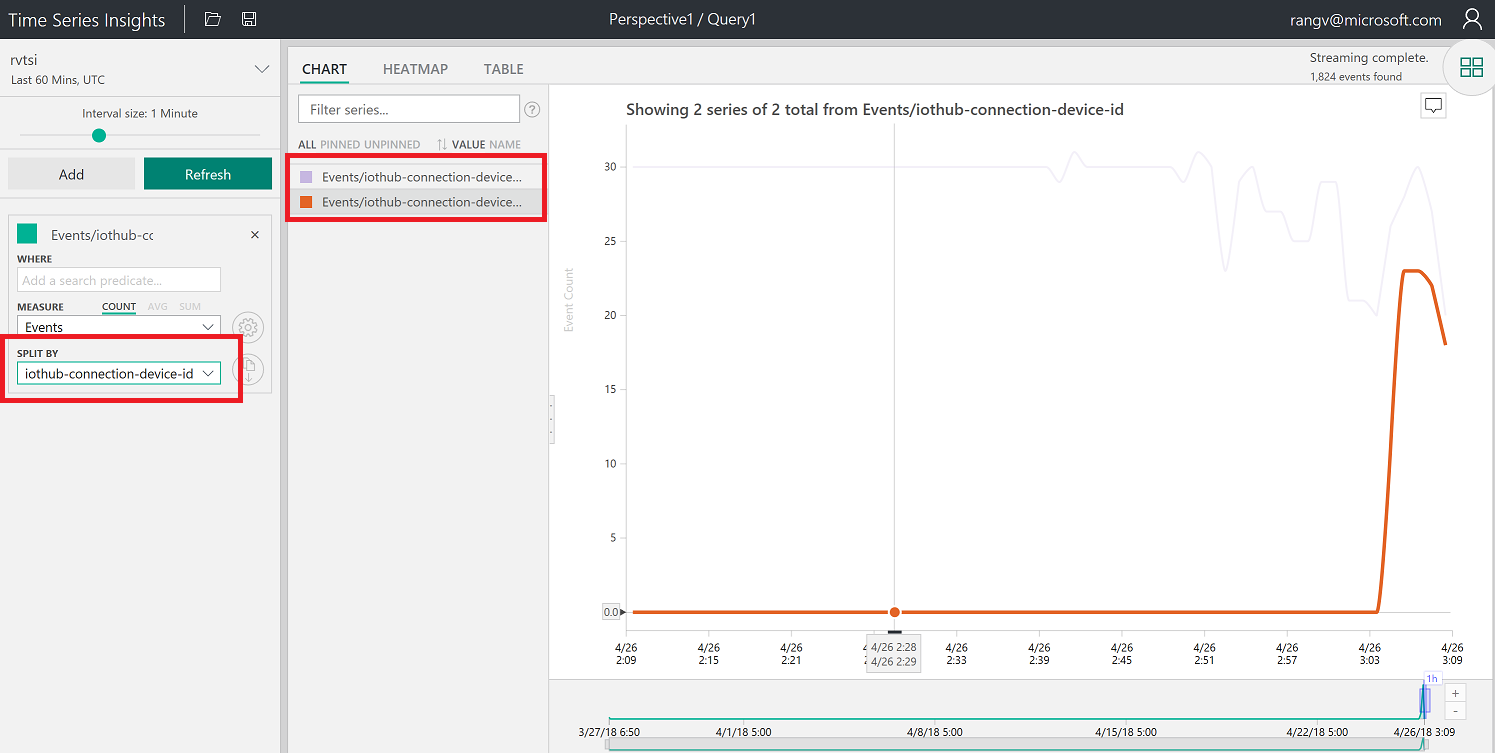
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/19_select_user.png)

**Time Series Insights Explorer**

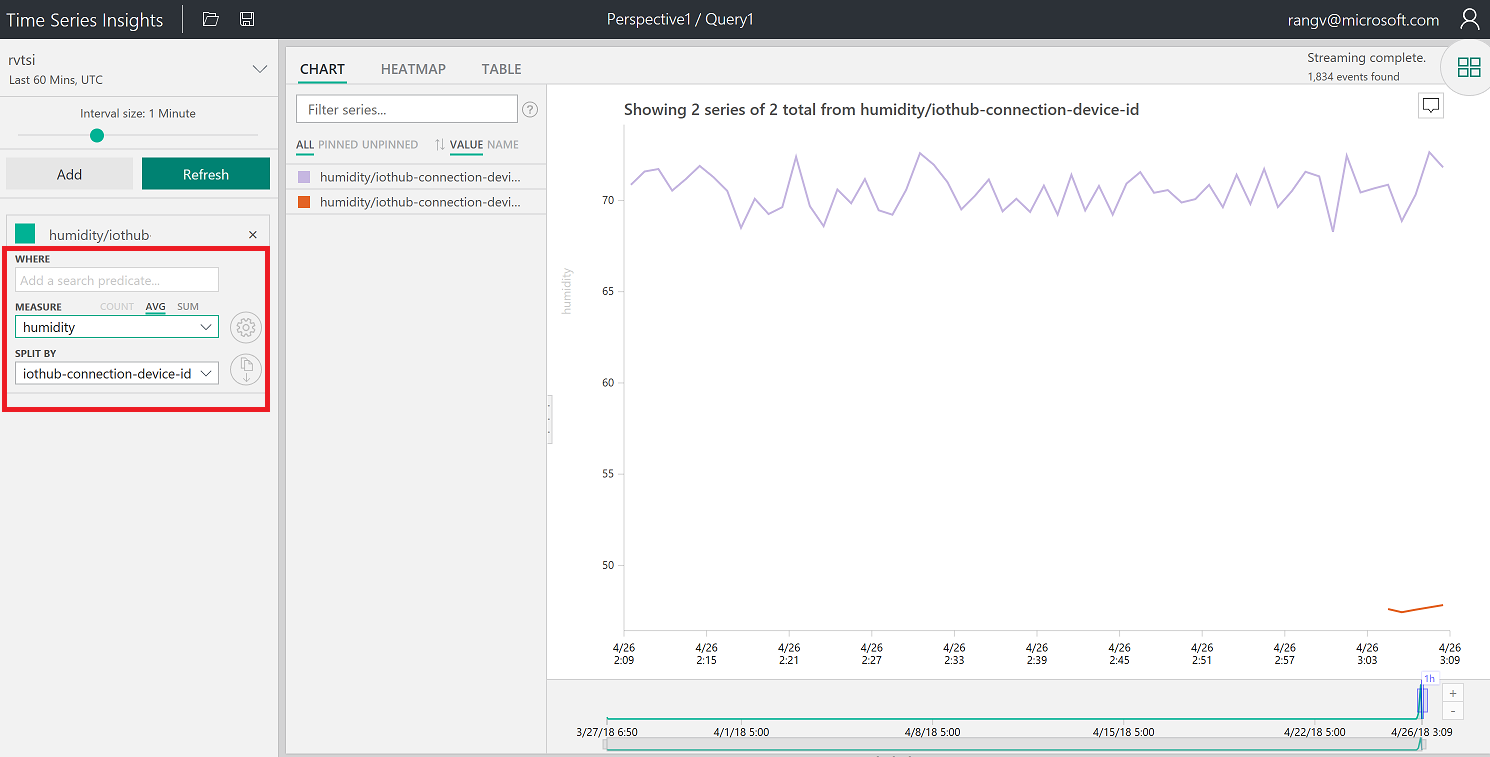
Go To Time Series Insights Explorer

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/05_GoTo_TSI_Explorer.png)

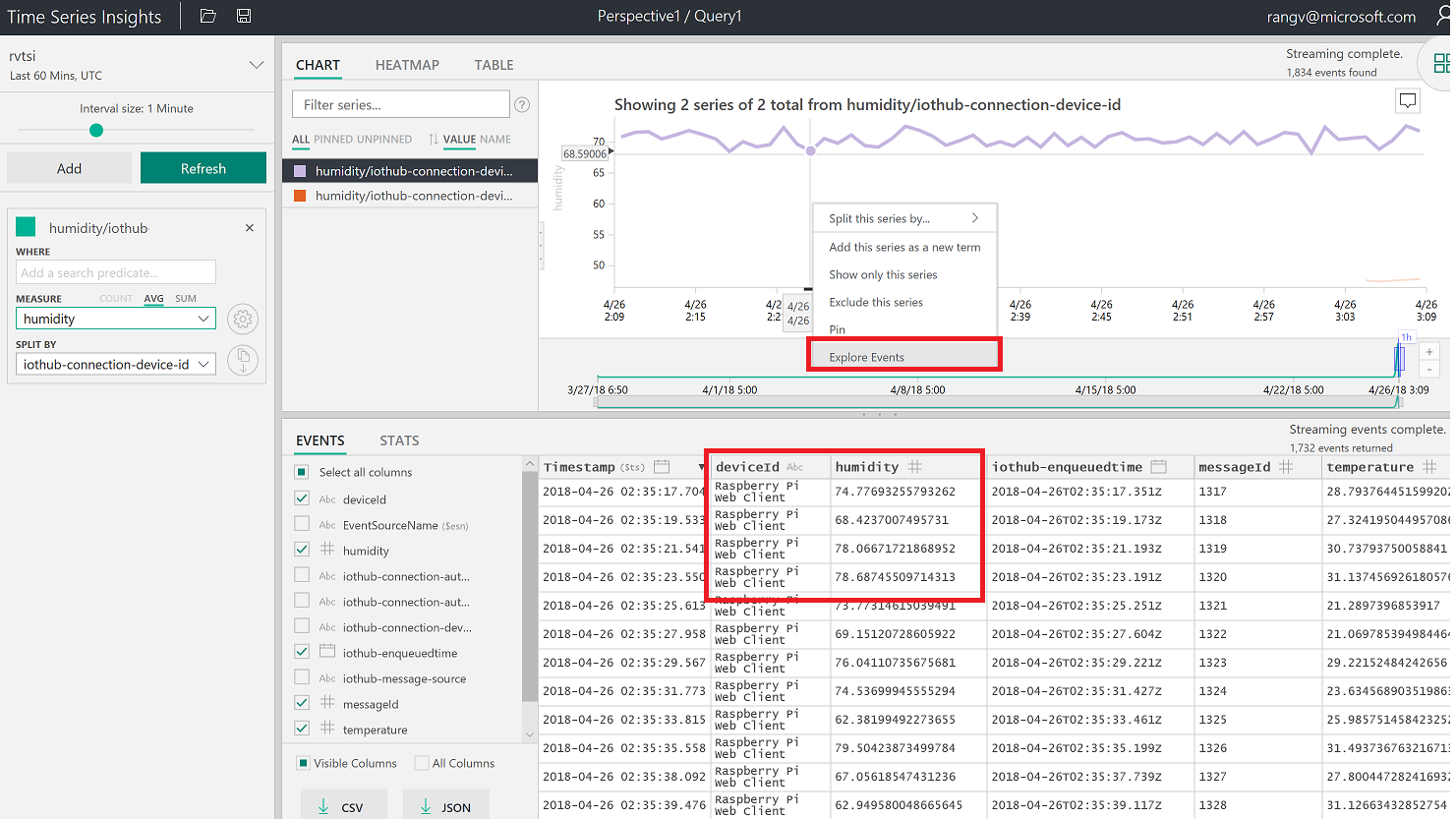
Split By ID. You will see data flowing from two devices. MXChip and Pi Simulator.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/06_Visual1.png)

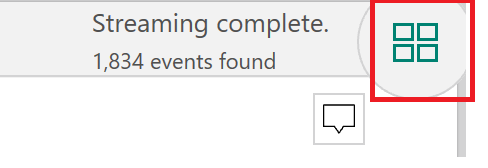
Select humidity and Split By ID. You will see data flowing from two devices. MXChip and Pi Simulator.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/07_Visual2.png)

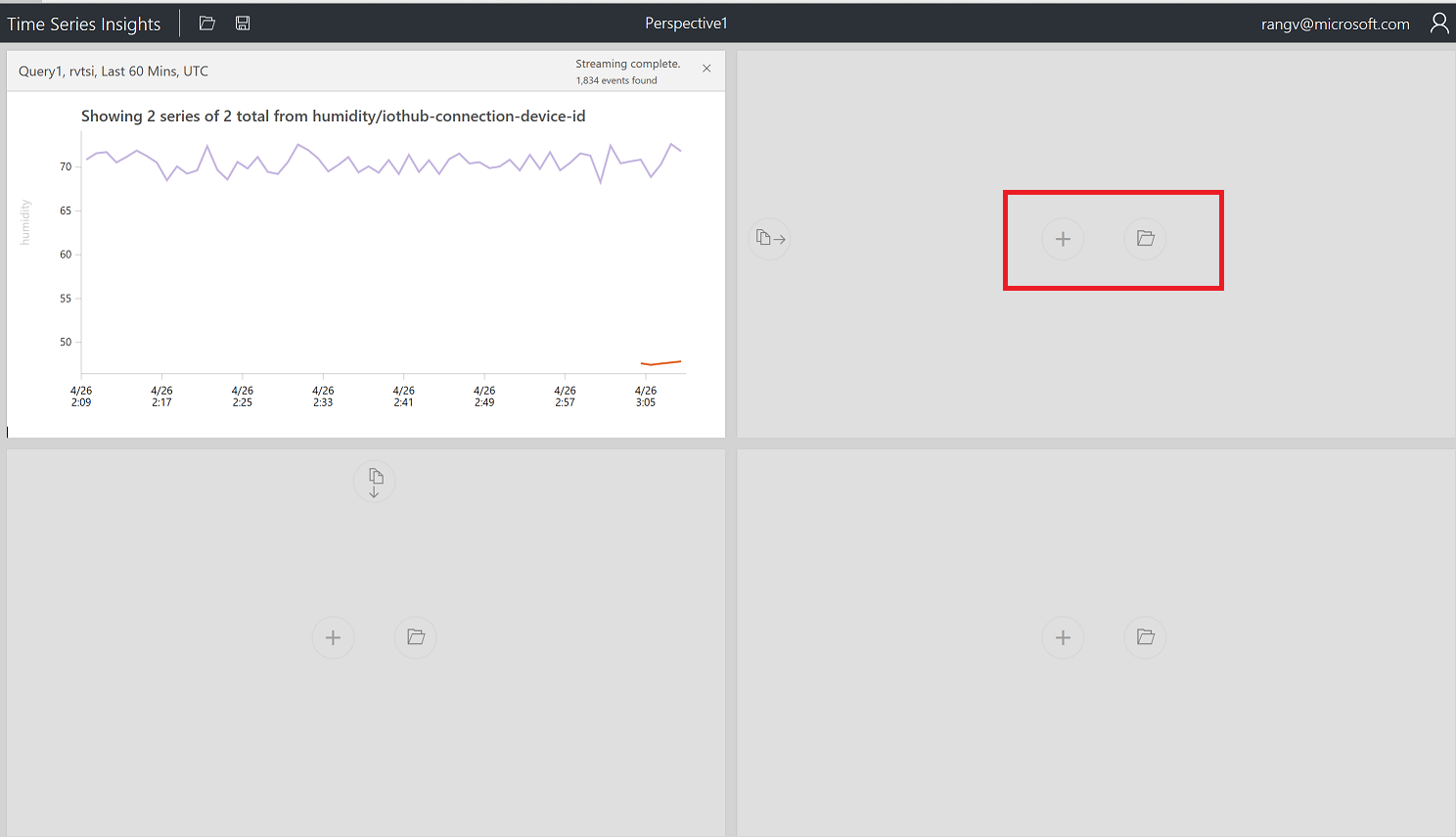
Right Click to Explore events. You can download events in CSV and JSON format by clicking on **CSV or JSON** buttons

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/08_Visual3.png)

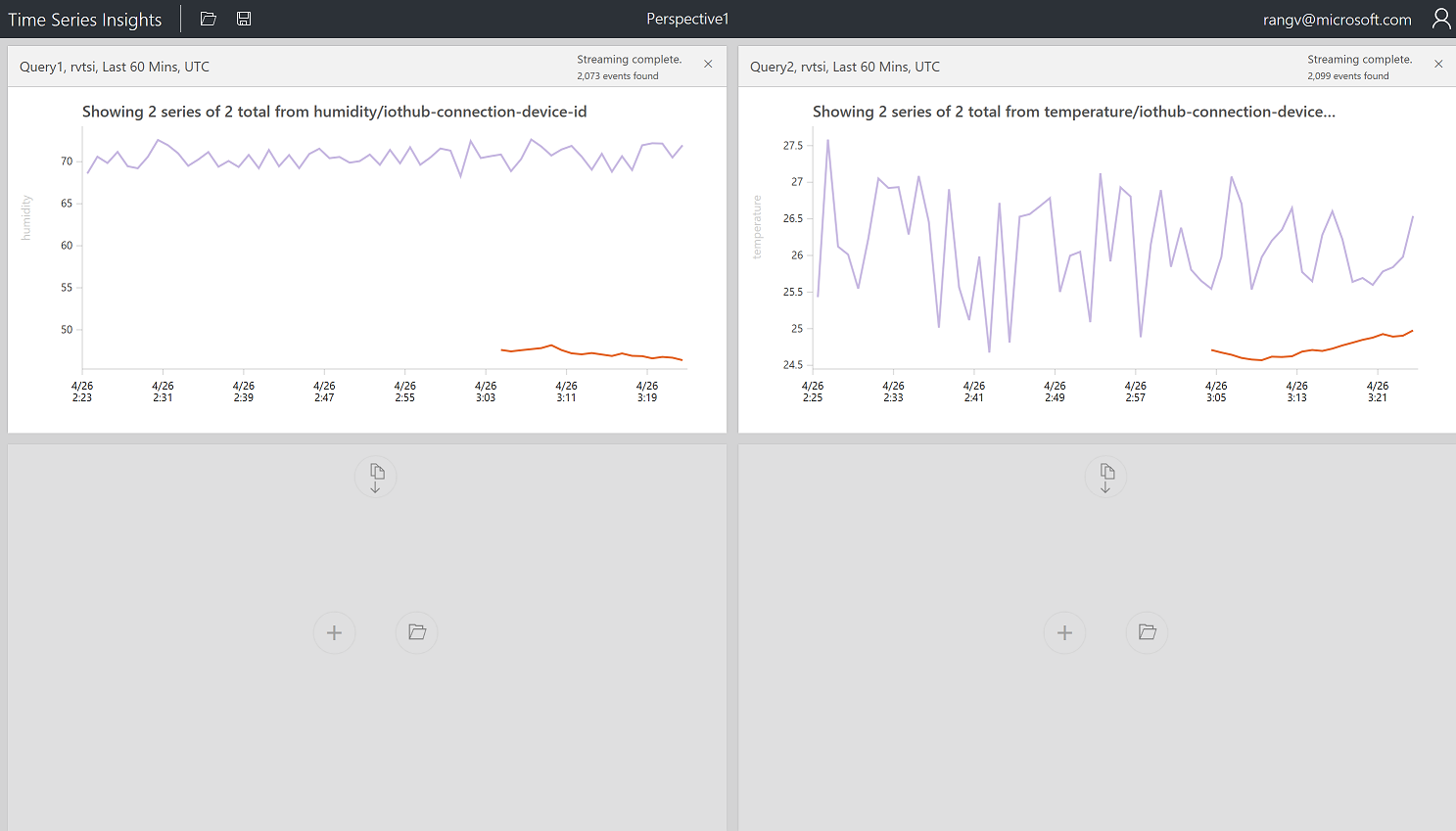
Create a perspective by clicking on the image shown below

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/perspective.png)

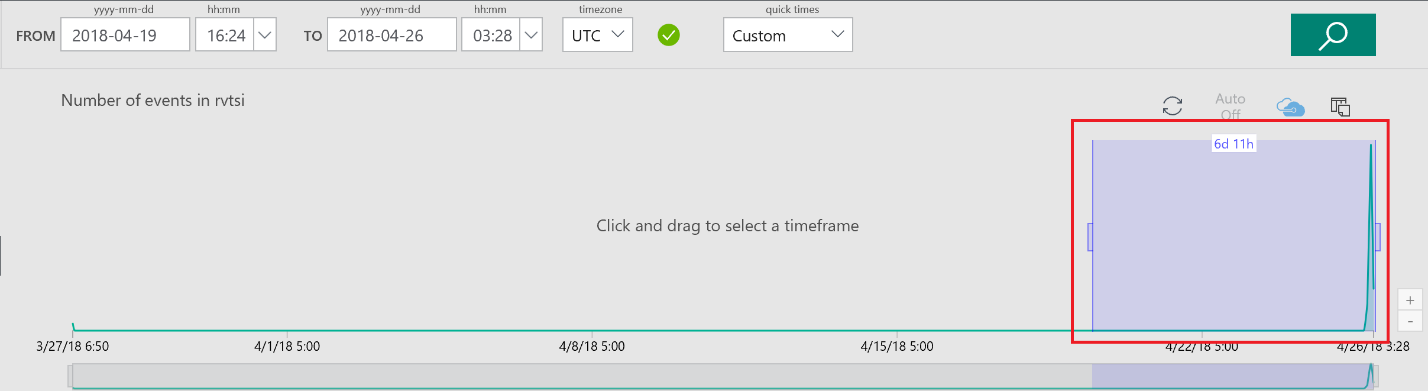
Click **+** to add a new query

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/10_visual10.png)

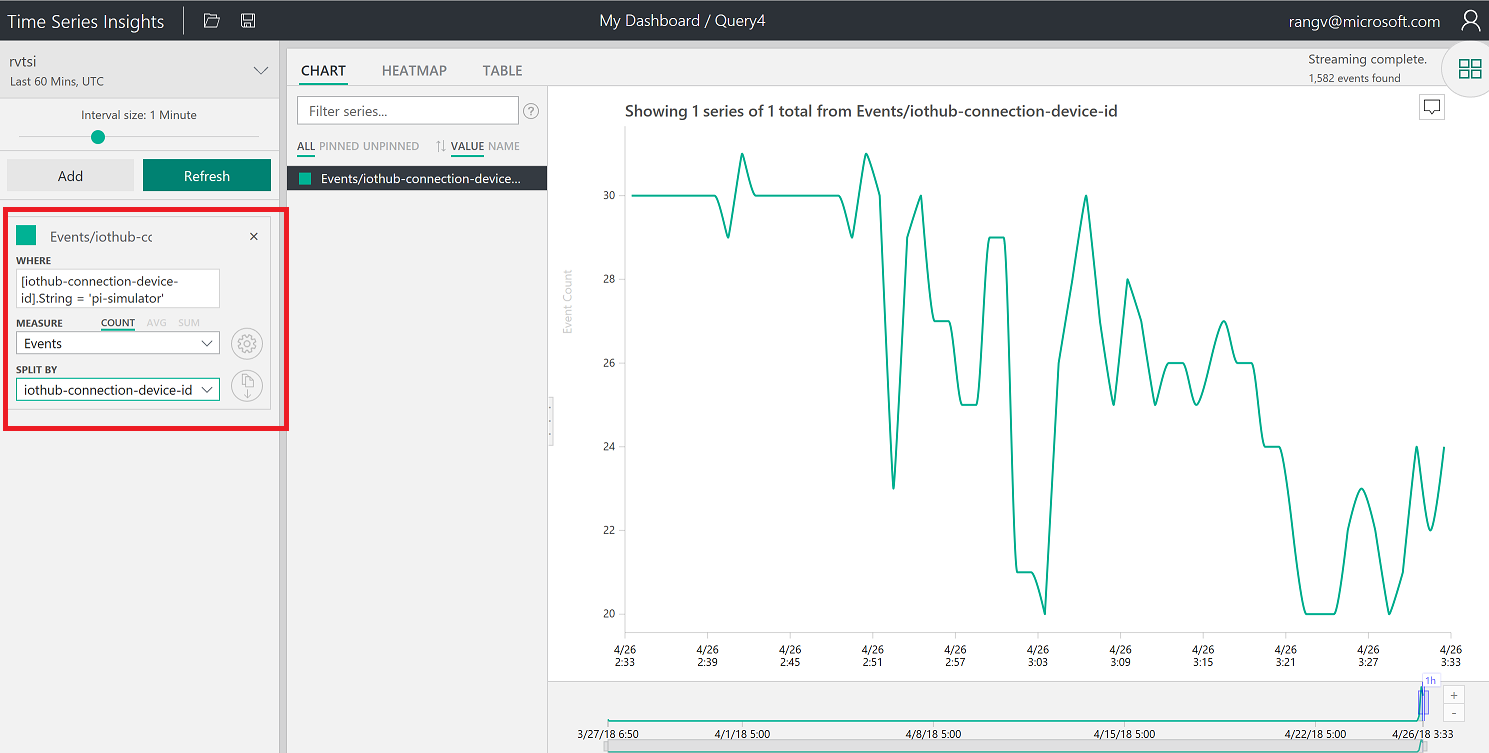
Select Temperature and split by Device ID and click on perspective image.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/11_visual11.png)

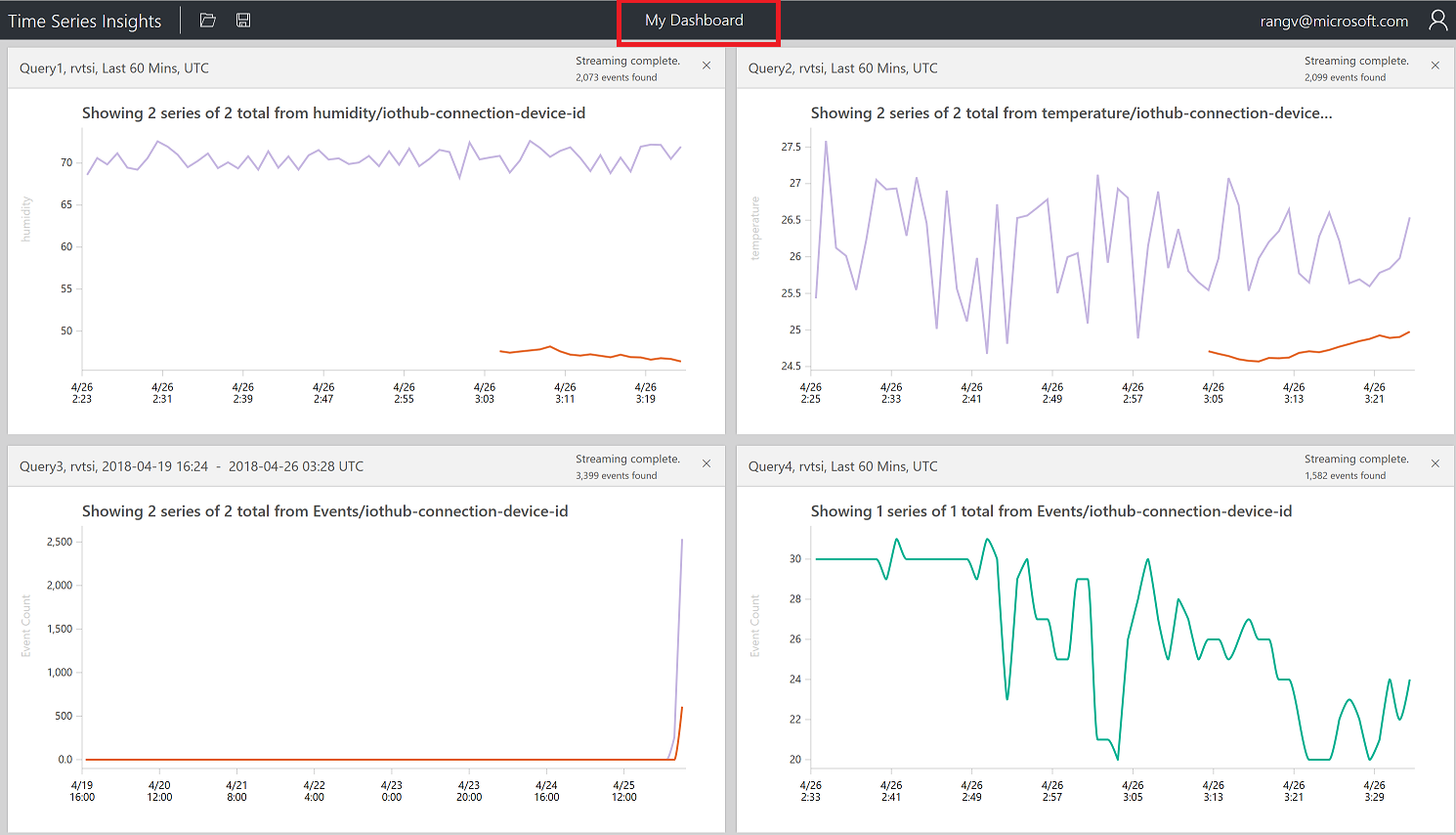
Create a chart by selecting a timeframe with drag feature

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/12_Visual12.png)

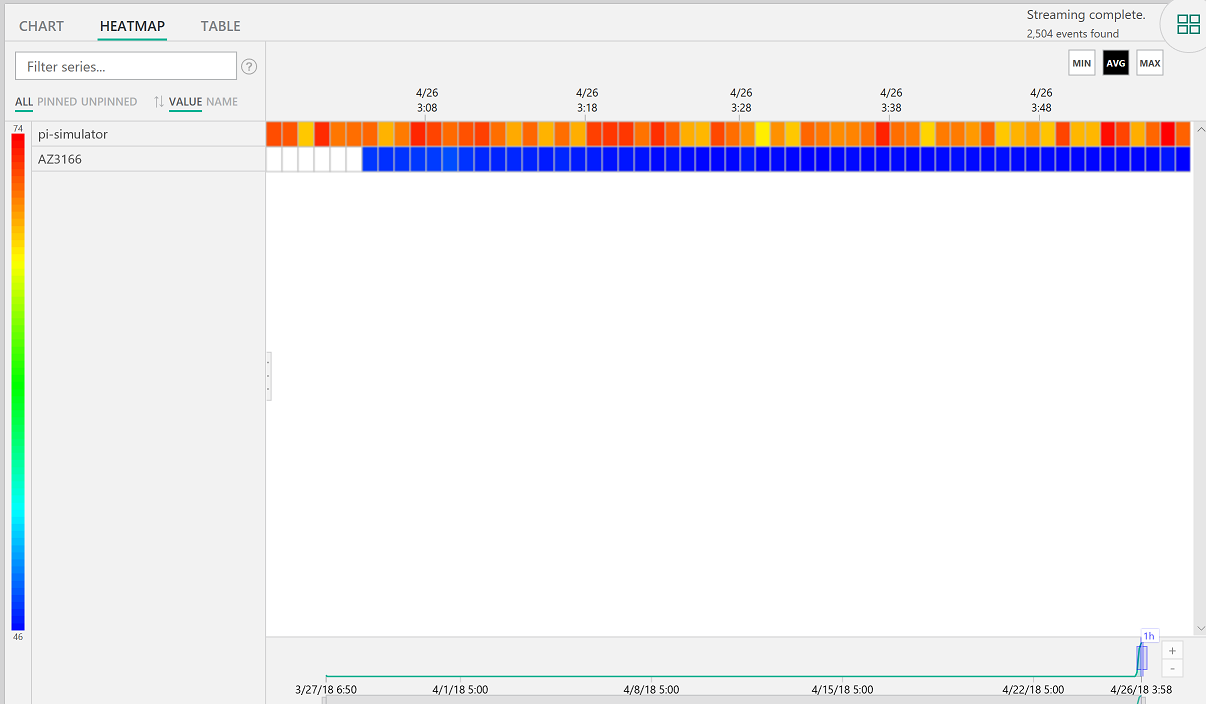
Create a Chart by adding a predicate

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/predicate.png)

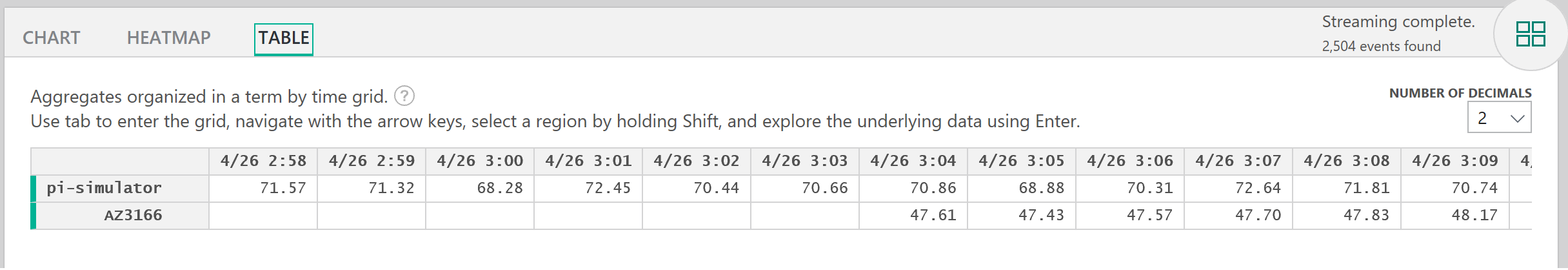
Perspective with 4 different charts and also changed Title

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/14_Visual_dashboard.png)

Click on Heatmap

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/heatmap.png)

View data in a table

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/table.png)

**Cold Path Storage**

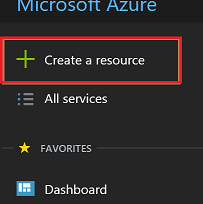
**Create Data Lake Store and Stream Data from IoTHub using Azure Stream Analytics**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/datalakestore.jpg)

Azure Data Lake Store is an enterprise-wide hyper-scale repository for big data analytic workloads. Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. Data Lake Store can store trillions of files. A single file can be larger than one petabyte in size. This makes Data Lake Store ideal for storing any type of data including massive datasets like high-resolution video, genomic and seismic datasets, medical data, and data from a wide variety of industries.

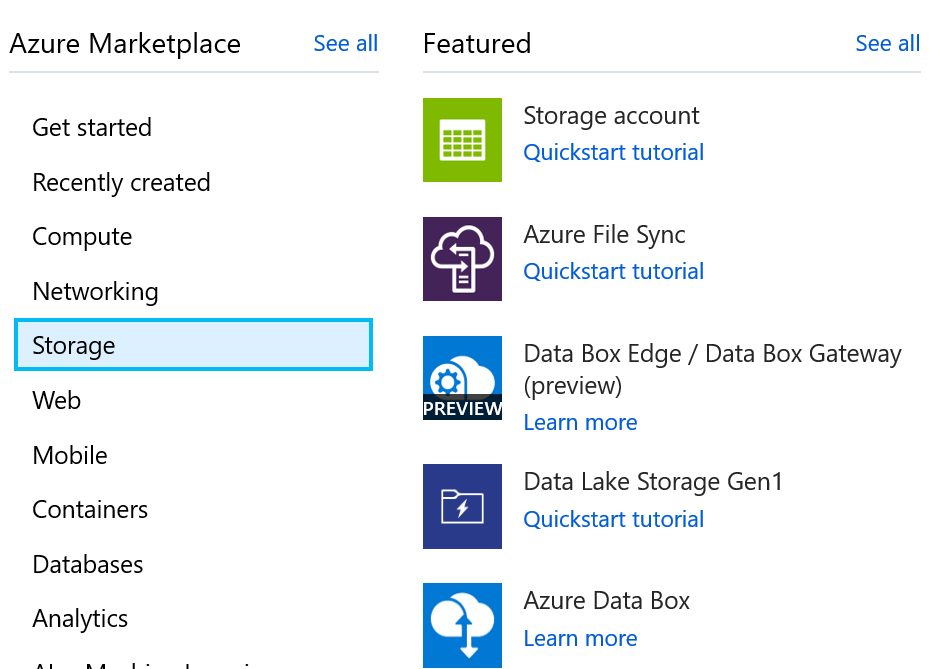
**Create Azure Data Lake Store**

Create a hyper scale data lake store to store IoT Data. Click on **Create a resource**

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/create_resource.png)

Click on **Storage**

Click on **Data Lake Storage Gen 1**



During creation of data lake you have the choice to encrypt the store

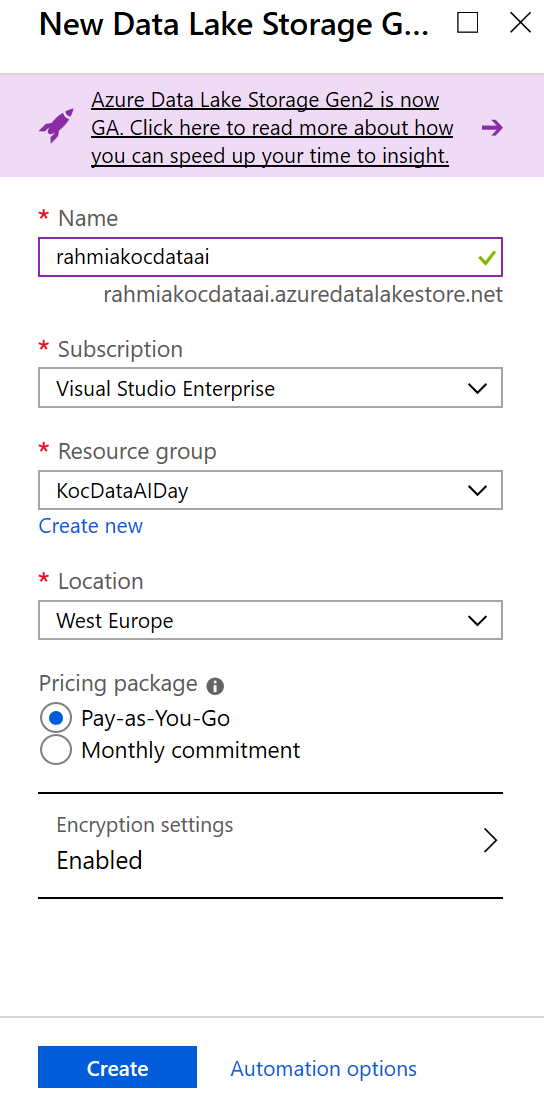
Data Lake Store protects your data assets and extends your on-premises security and governance controls to the cloud.

Your data is

* always encrypted
* while in motion using SSL
* at rest using service or user-managed HSM-backed keys in Azure Key Vault.

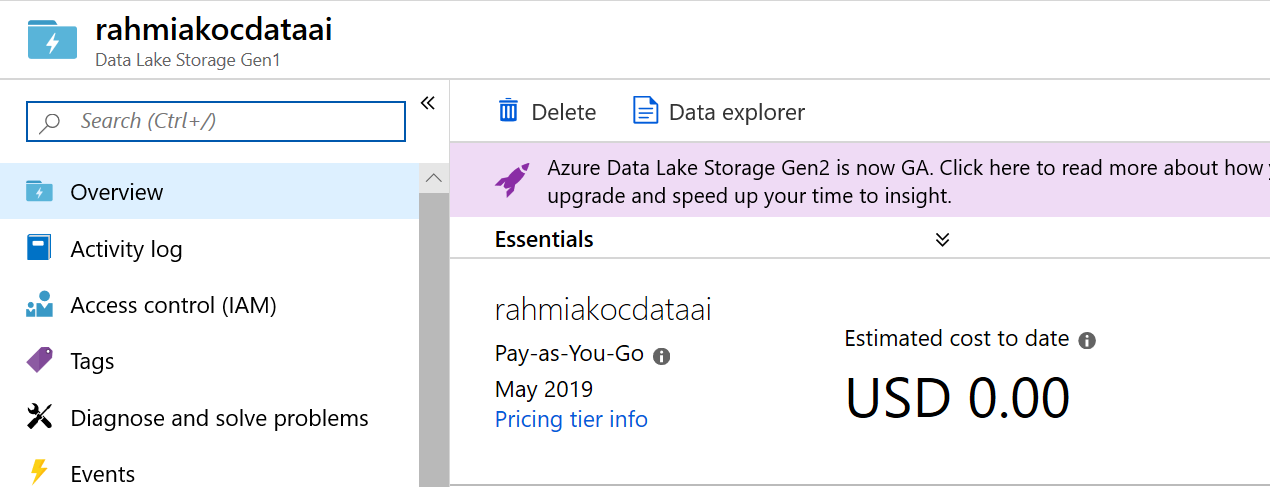
Single sign-on (SSO), multi-factor authentication, and seamless management of millions of identities is built-in through Azure Active Directory. Authorize users and groups with fine-grained POSIX-based ACLs for all data in your store and enable role-based access controls. Meet security and regulatory compliance needs by auditing every access or configuration change to the system.

Click on **Create** button



**Explore Data in Data Lake Store**

**Create Folders in Data Lake Store**



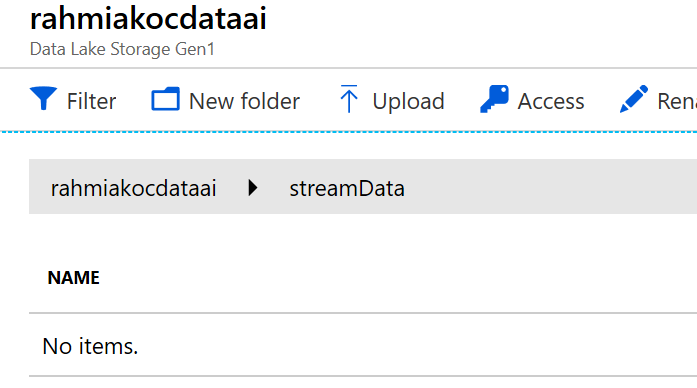
Create /workshop/streaming folder to store Streaming data coming from your device through IoTHub using Stream Analytics Job

Create /workshop folder

Create /workshop/streaming folder



You should have the folder structure below in place to start streaming data to data lake store

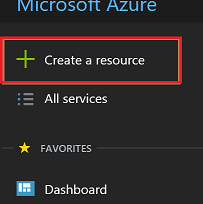


**Create Stream Analytics Job**

Azure Stream Analytics is a managed event-processing engine set up real-time analytic computations on streaming data. The data can come from devices, sensors, web sites, social media feeds, applications, infrastructure systems, and more

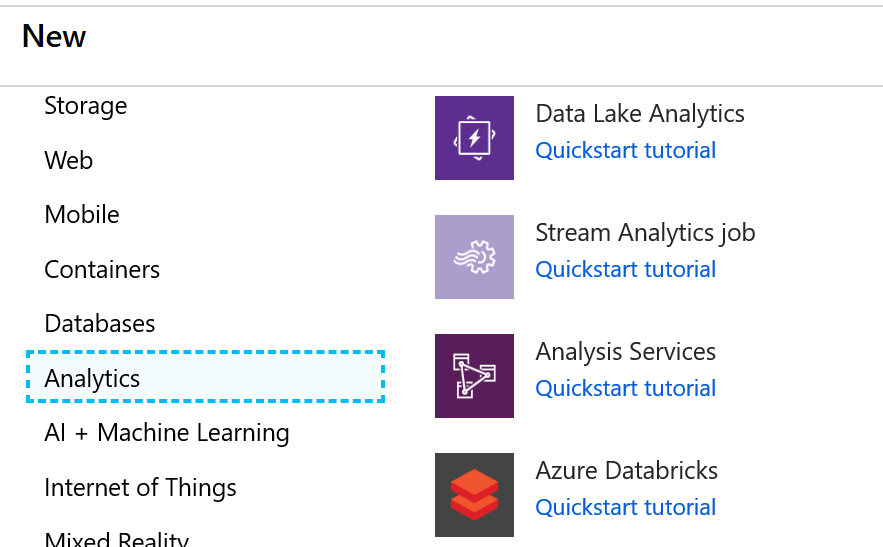
<iframe src="<https://channel9.msdn.com/Shows/Internet-of-Things-Show/Stream-Analytics-in-IoT-solutions/player>" width="480" height="270" allowFullScreen frameBorder="0"></iframe>

Create a hyper scale data lake store to store IoT Data. Click on **Create a resource**

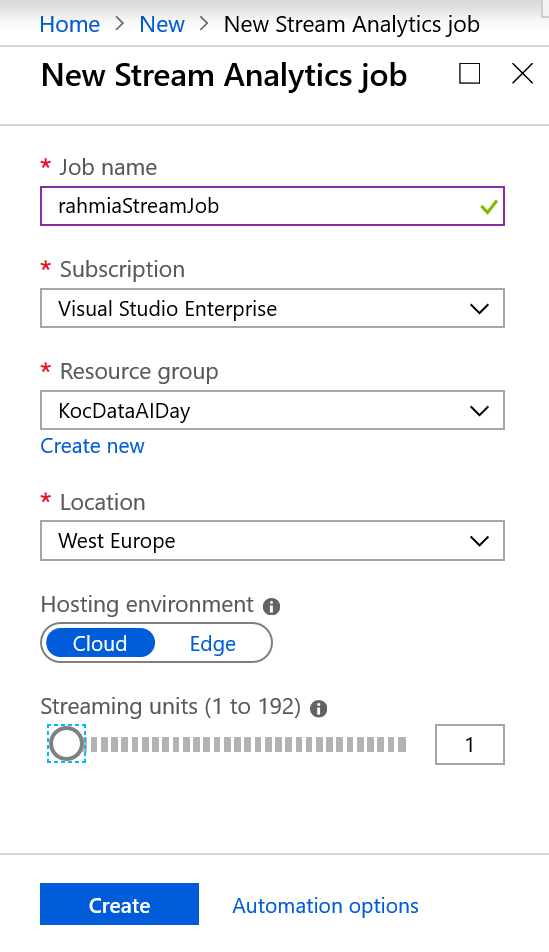
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/create_resource.png)

Click on **Analytics**

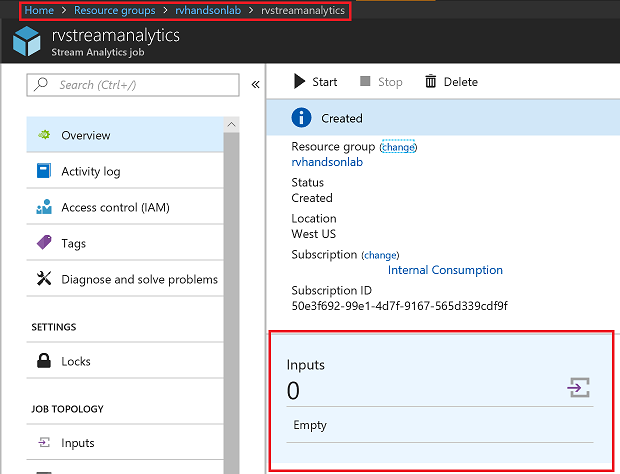
Click on **Stream Analytics Job**



Stream Analytics job cab be created to run on the cloud as well as on the Edge. You will chose to run this on the cloud



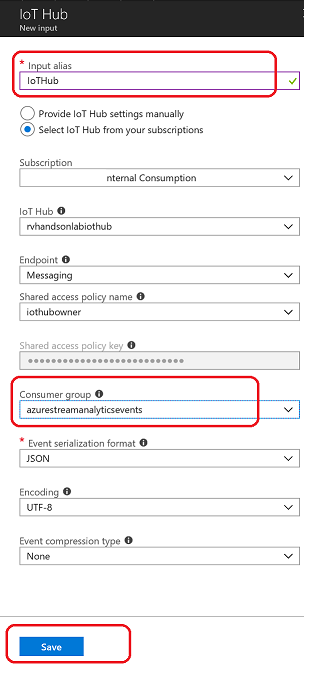
Add Input for Streaming Job

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/09_Add_Input.png)

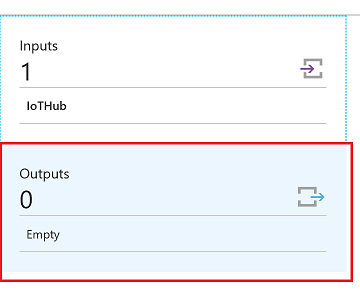
Select IoTHub as Input

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/10_Add_IoTHub.png)

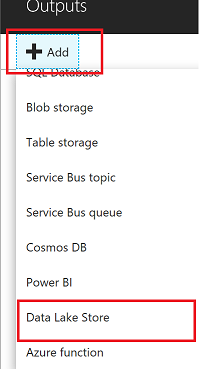
Make sure to provide a consumer group. Each consumer group allows up to 5 output sinks/consumers. Make sure you create a new consumer group for every 5 output sinks and you can create up to 32 consumer groups.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/11_Save_IoTHub.png)

Add Data Lake Store as Output for Streaming Job

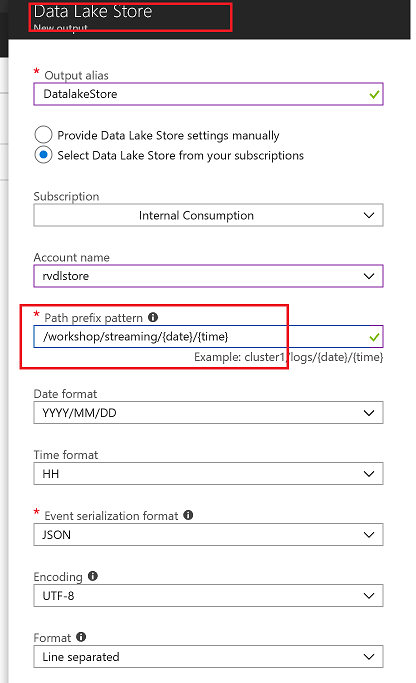
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/12_Add_Data_Lake_Store.png)

Select Data Lake Store as output sink

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/13_Add_Output.png)

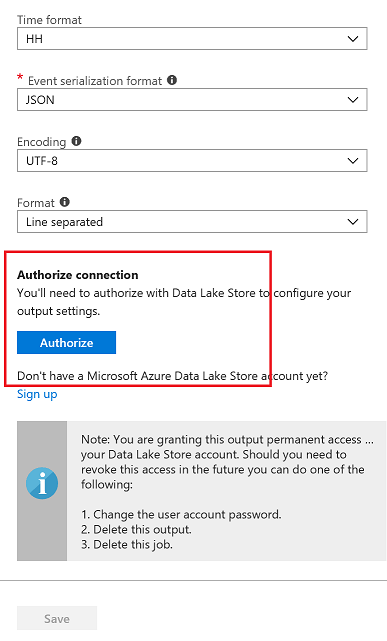
Select the Data Lake Store account you created in previous steps and provide folder structure to stream data to the store

/workshop/streaming/{date}/{time} with Date=YYYY/MM/DD format and Time=HH format will equate to /workshop/streaming/2018/03/30/11 on the store

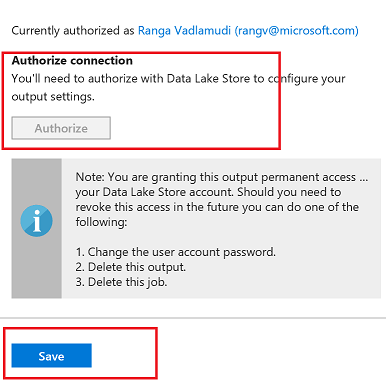
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/14_Save_Output.png)

You will have to Authorize data lake store connection for Stream analytics to have access to be able to write to data lake store

1. Multi-factor authentication based on OAuth2.0
2. Integration with on-premises AD for federated authentication
3. Role-based access control
4. Privileged account management
5. Application usage monitoring and rich auditing
6. Security monitoring and alerting
7. Fine-grained ACLs for AD identities

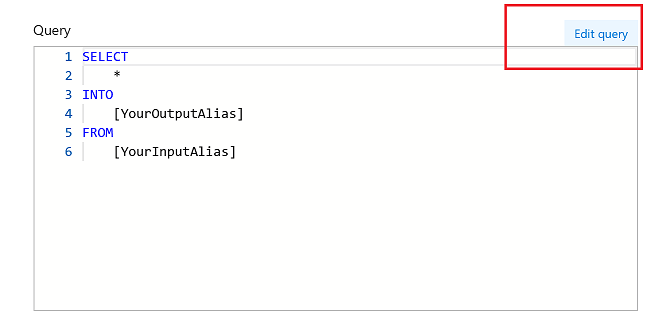
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/15_Save_Output_2.png)

You will see a popup and once the popup closes Authorize button will be greyed out after azuthorization is complete. There are exception cases where popup doesnt appear.In this case try again in incognito mode

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/16_Save_Output_3.png)

**Edit Stream Analytics Query**

Edit Query for Streaming Job, Stream Data from IoTHub to Datalake Store

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/17_Edit_Query.png)

Query

SELECT

\*, System.Timestamp as time

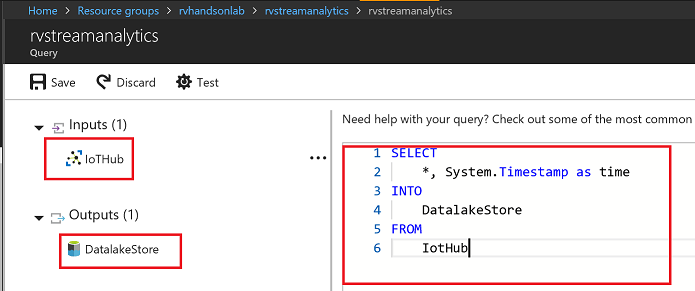
INTO

DatalakeStore

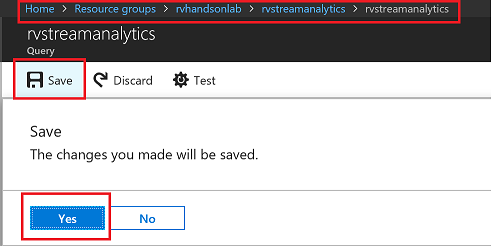
FROM

IotHub

Save the query

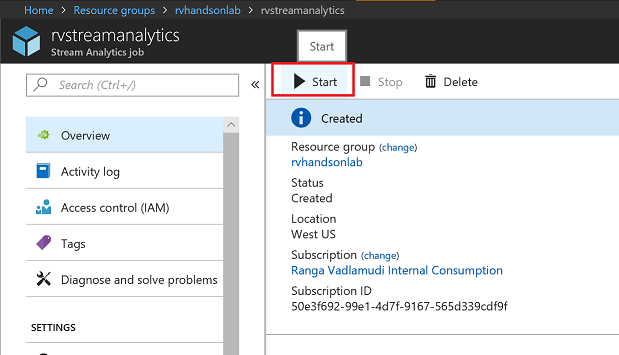
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/18_Save_Query.png)

Accept by pressing yes

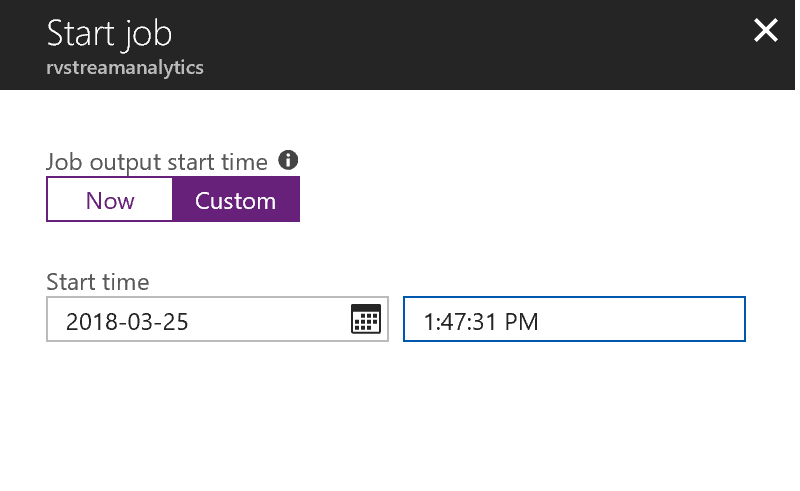
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/19_Save_Query_Yes.png)

**Start Streaming Analytics Job**

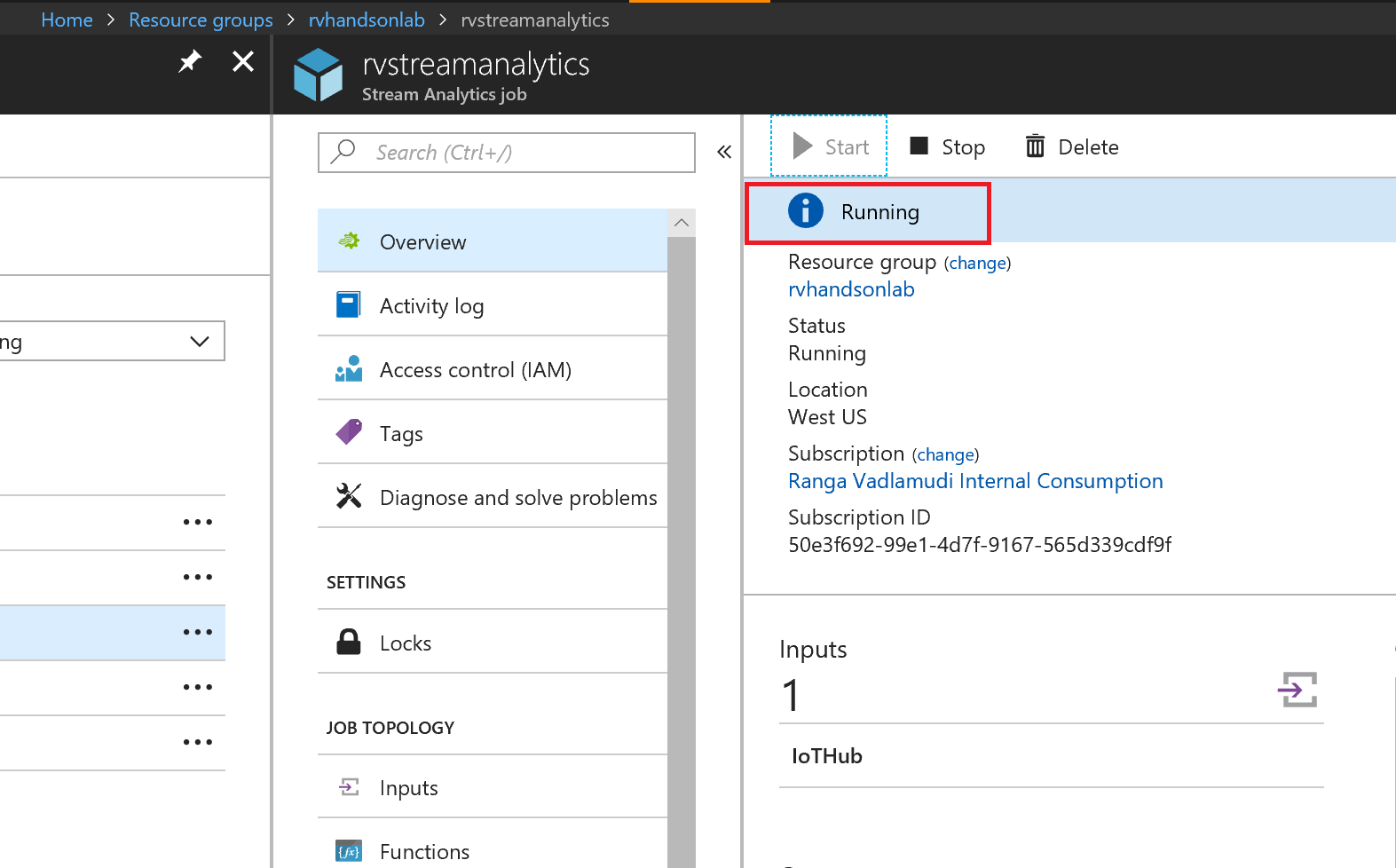
Start the stream job which will read data from IoTHub and store data in Data lake Store

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/20_Start_Stream_Analytics_Job.png)

You can pick custom time to go back a few hours to pick up data from when your device has started streaming

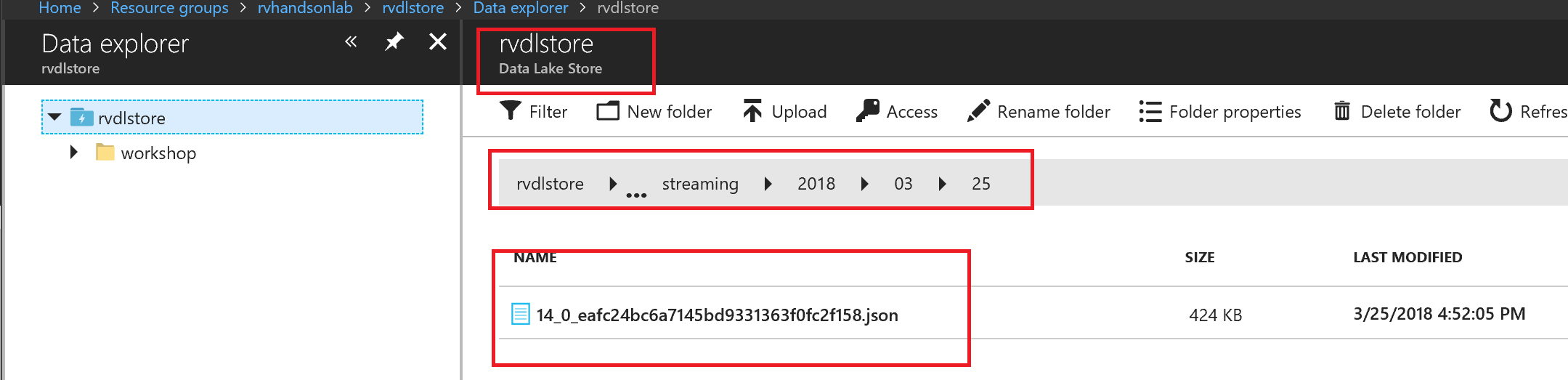
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/21_Start_custom.png)

Wait till job goes into running state, if you see errors could be from your query, make sure syntax is correct

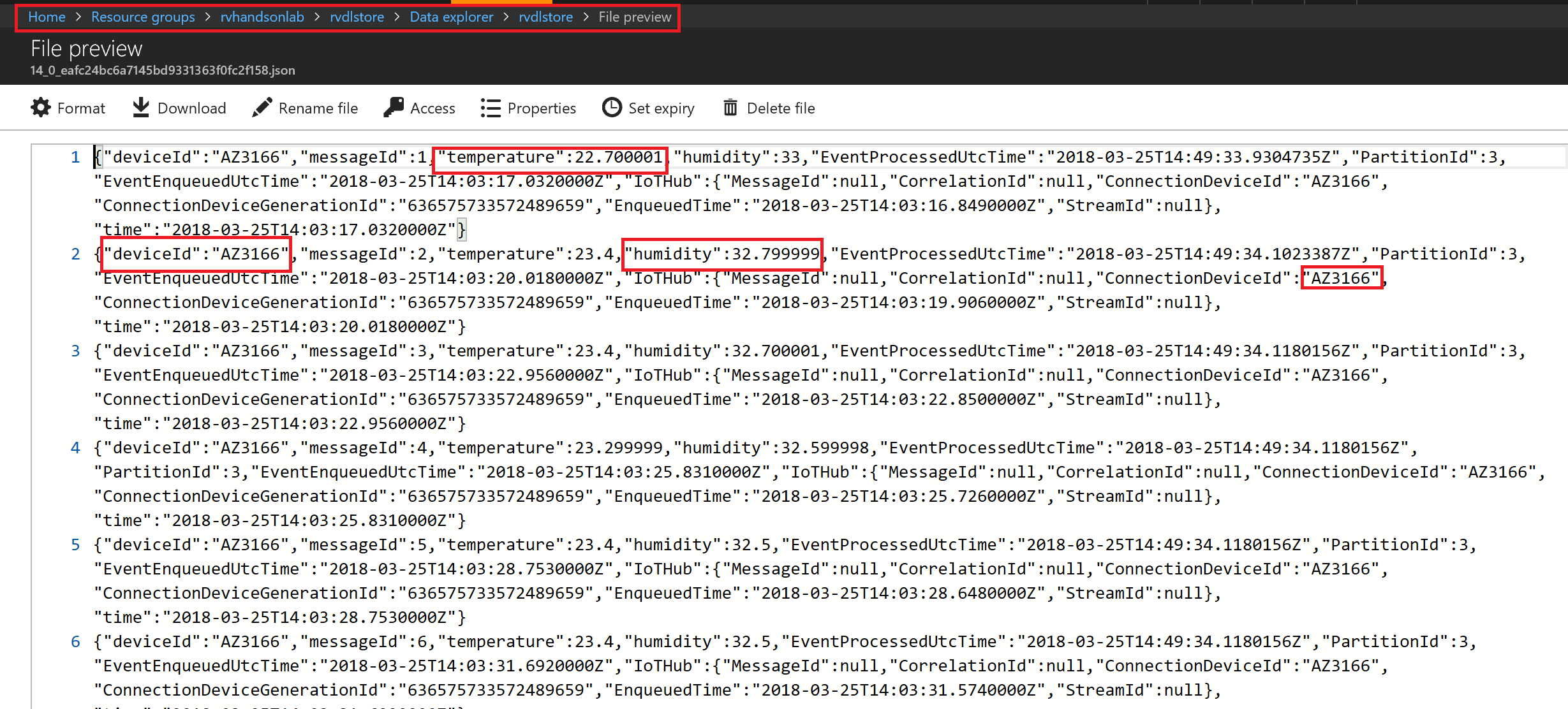
[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/22_running.png)

**Explore Streaming Data**

Go to Data Lake store data explorer and drill down to /workshop/streaming folder.You will see folders created with YYYY/MM/DD/HH format.

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/23_datalake_store_explore_streaming_data.png)

You will see json files, with one file per hour, explore the data

[](https://github.com/sasever/MSReady2018/blob/master/01DAIDP-302/images/24_datalake_file.png)