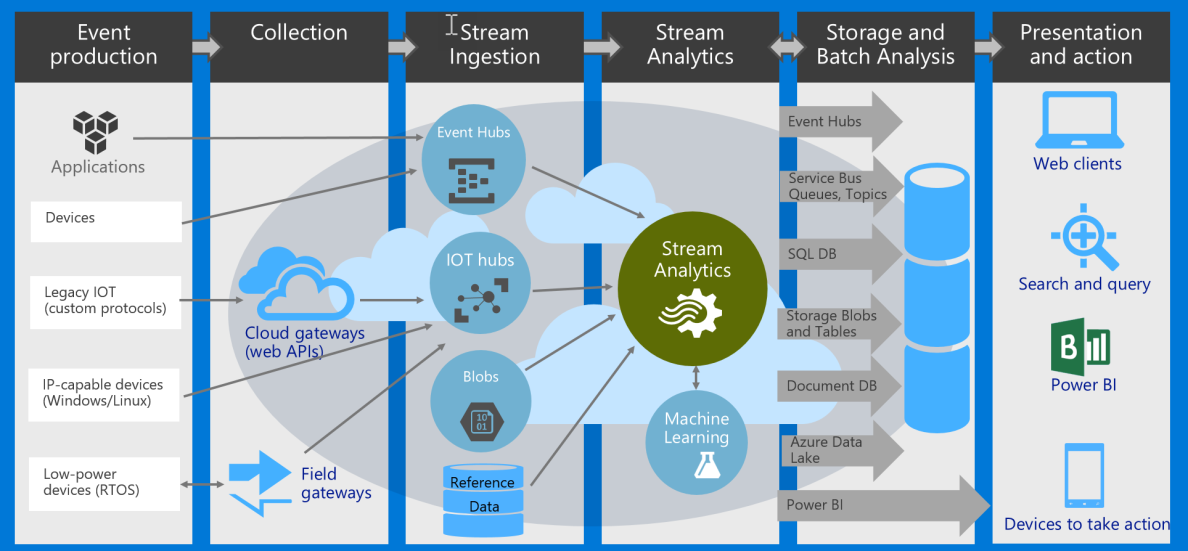
**Setup Guide for FT900 Microsoft Azure IoT demo**

**I. MICROSOFT AZURE IoT Architecture**

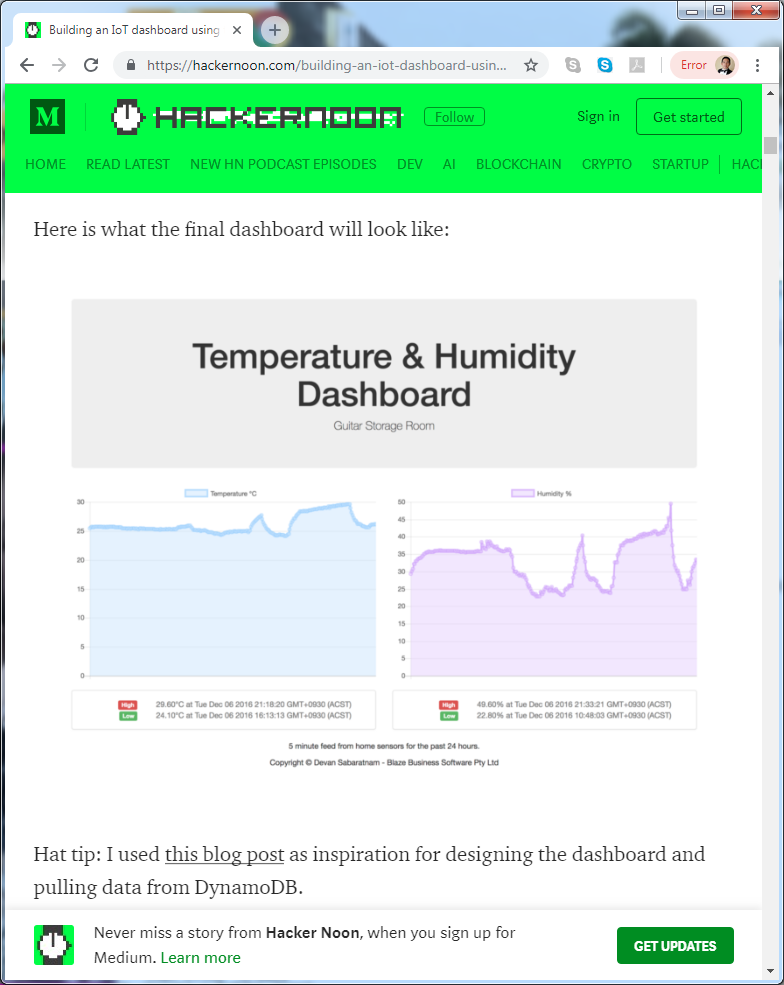
****

BACK-END**: FT900 ->** **IoT Hub -> Stream Analytics -> CosmosDB (aka DocumentDB)**

FRONT-END: **Browser -> (Dashboard webpage in) Storage -> CosmosDB 🡨 [NG: TODO]**

TODO: Integrate the provided Node.JS script to the sample AWS IoT Dashboard at

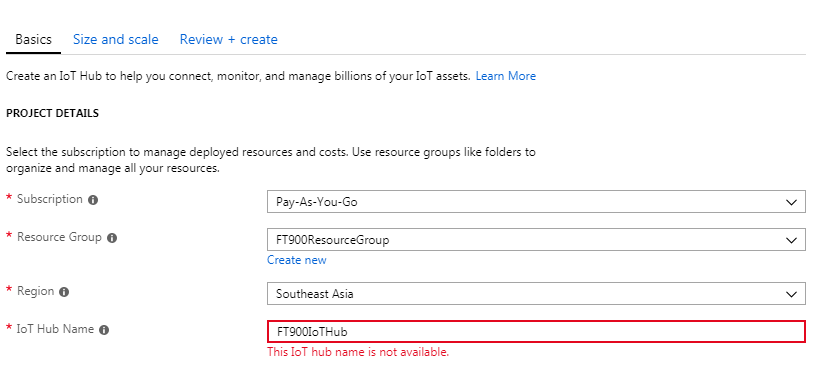
<https://hackernoon.com/building-an-iot-dashboard-using-the-onion-omega-and-amazon-aws-a3520f850c9>

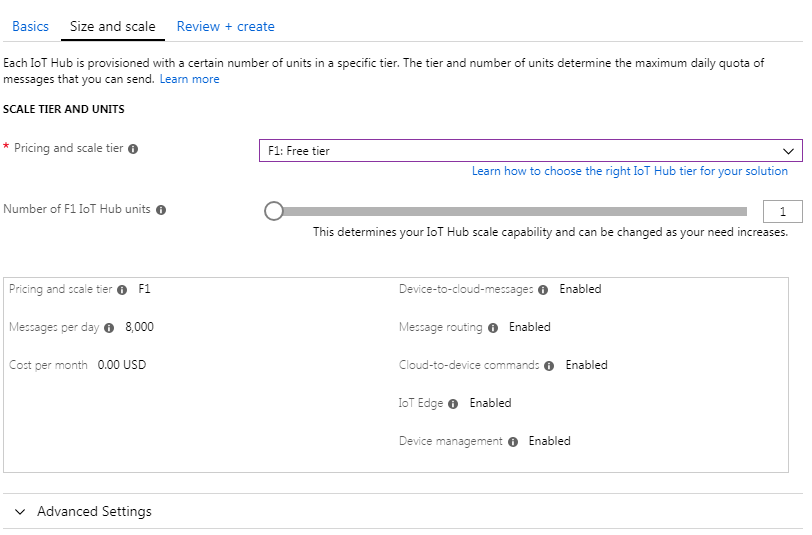


**II. BACKEND CLOUD Setup Guide:**

1. Create an IoT Hub resource ‘FT900IoTHub’

Create a resource > Internet of Things > IoT Hub



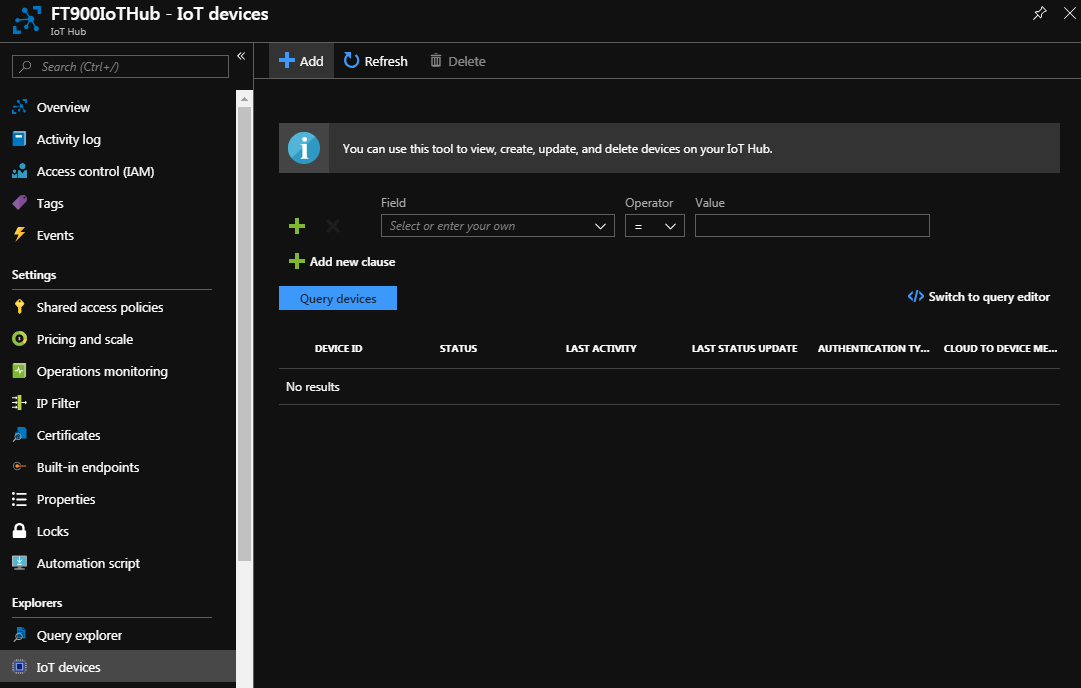


If free tier is not available, select the basic tier.

Creating an IoT Hub will take about 3-5 minutes.

1. Create a Device (under IoT Hub) called ‘ft900device1’ – will use SAS token authentication

IoT Hub > IoT devices > Add



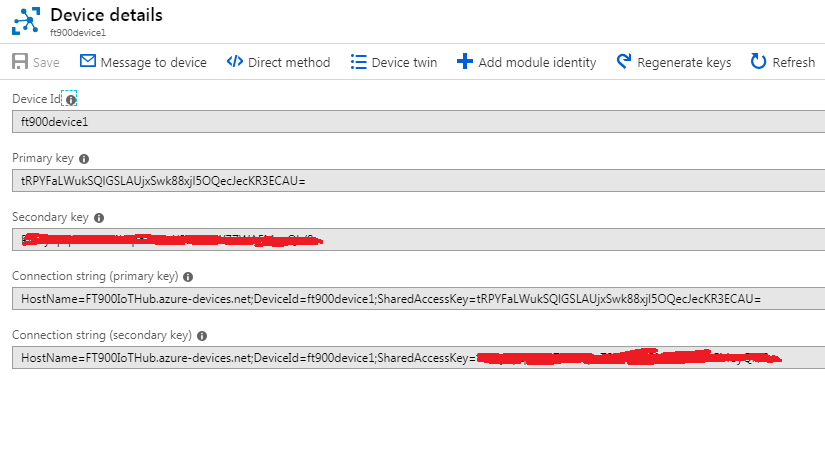
Create Device1 using the following details:

DeviceID: ft900device1

Authentication type: Symmetric Key

Auto-generate keys: YES

Click Save button

Device 1 will use SAS token authentication. Copy the generated Primary Key and update ft900device1\_sas\_azure.pem 

1. (Optional) Create a Device (under IoT Hub) called ‘ft900device2’ – X509 certificate authentication

This is optional.

Create Device2 using the following details:

DeviceID: ft900device2

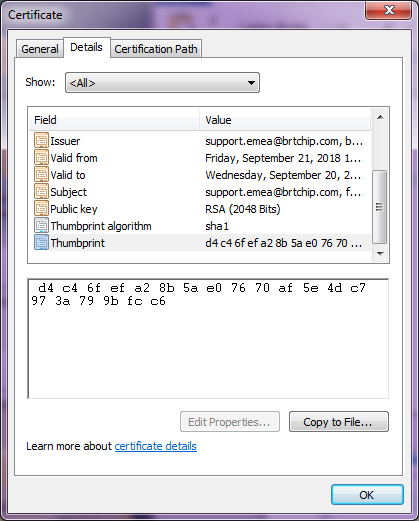
Authentication type: x.509 Self-Signed

Primary Thumbprint: d4c46fefa28b5ae07670af5e4dc7973a799bfcc6

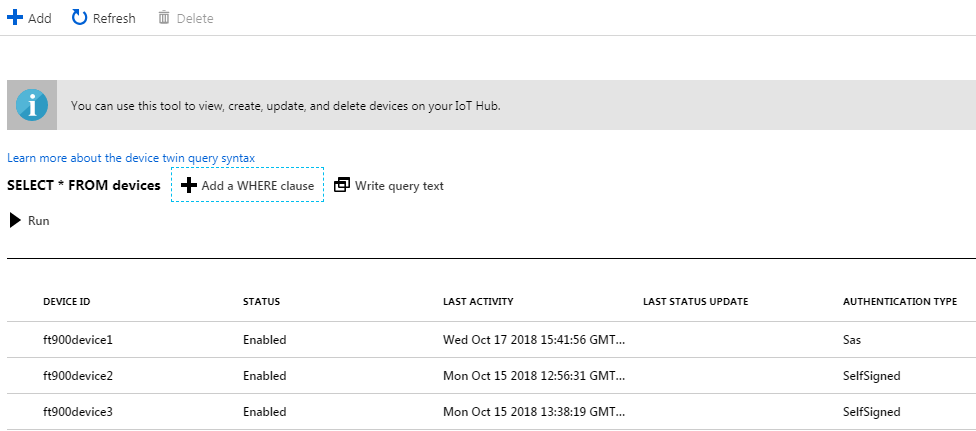
Secondary Thumbprint: d4c46fefa28b5ae07670af5e4dc7973a799bfcc6

Click Save button

Device 2 will use X509 certificate authentication. The thumbprint above is retrieved using the certificate hash of ft900device2\_cert.pem.



After creating device1 and device2, both devices should appear as below



At this point, you can now check if FT900 device can connect successfully to Azure IoT using either ft900device1 or ft900device2.

Changes for iot\_config.h

#define USE\_MQTT\_BROKER MQTT\_BROKER\_MAZ\_IOT

#define USE\_MQTT\_DEVICE SAMPLE\_DEVICE\_1 // or SAMPLE\_DEVICE\_2

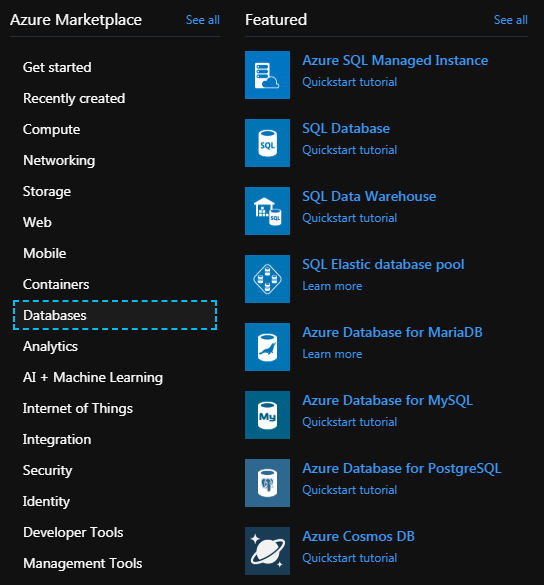
SAMPLE\_DEVICE\_1 refers to ft900device1 // configured to use SAS authentication

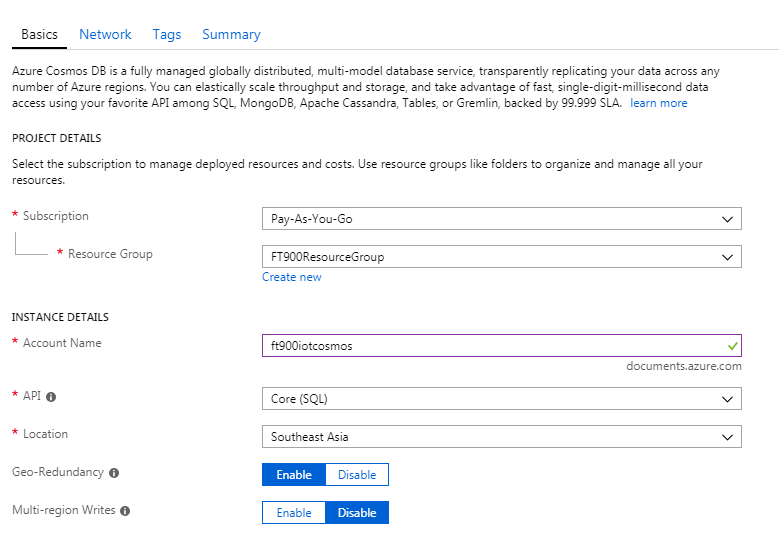
SAMPLE\_DEVICE\_2 refers to ft900device2 // configured to use X509 authentication

Please confirm that both devices are able to connect successfully. If yes, then proceed to the next step.

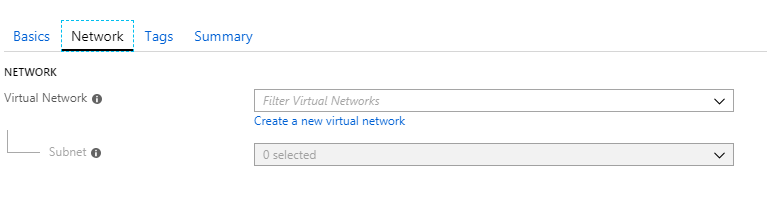
1. Create a CosmosDB resource ‘ft900iotcosmosdb’

Create a resource > Databases > Azure Cosmos DB

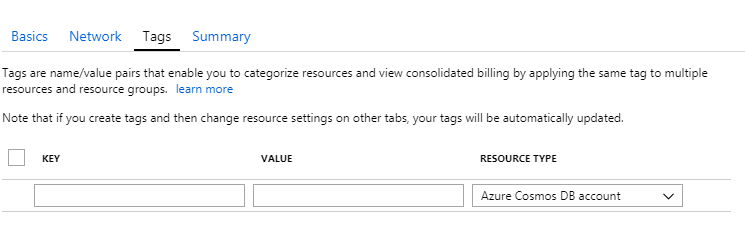




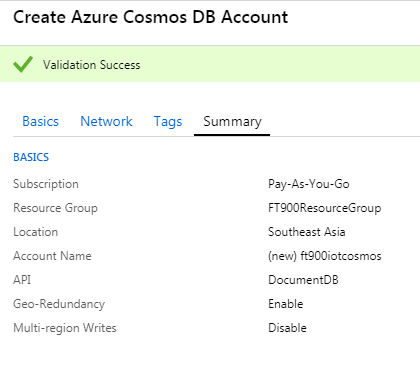
Click “Next: Network” button



Click “Next: Tags” button



Click “Next: Summary” button

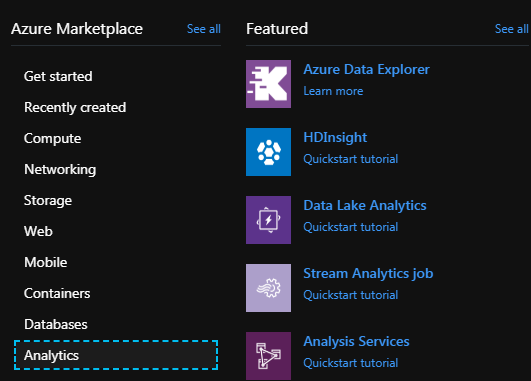


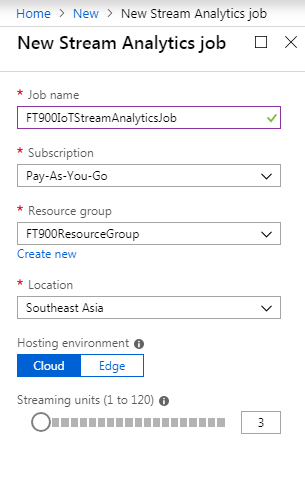
Click “Create” button

Creating the CosmosDB database will take about 6 minutes.

1. Create a Stream Analytics resource ‘FT900IoTStreamAnalytics’

Create a resource >Analytics > Stream Analytics job

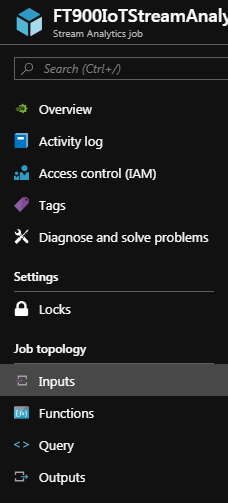




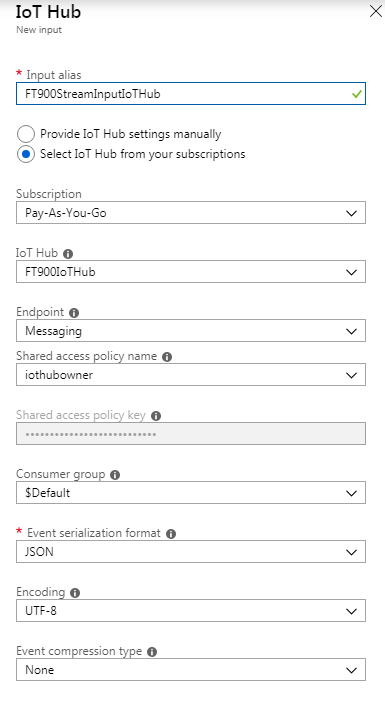
Click “Create” button.

Create the **Job Input**

Go to Job topology > Inputs > Add stream input > IoT Hub



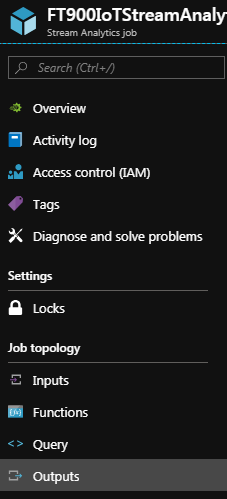
Add Input alias: FT900StreamInputIoTHub



Click “Save” button

Create the **Job Output**

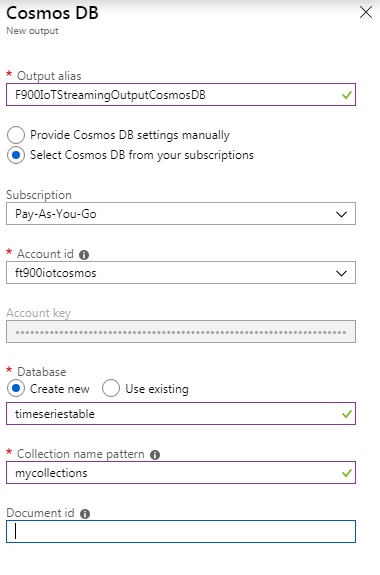
Go to Job topology > Outputs > Add > Cosmos DB



Add Output alias: FT900StreamOutputCosmosDB

Database Create new: timeseriestable

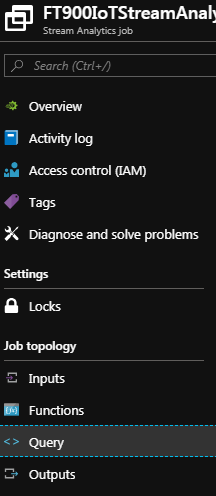
Collection name pattern: mycollections



Click “Save” button

Update the **Query**

Go to Job topology > Query > Add > Cosmos DB



Update the query

SELECT

\*

INTO

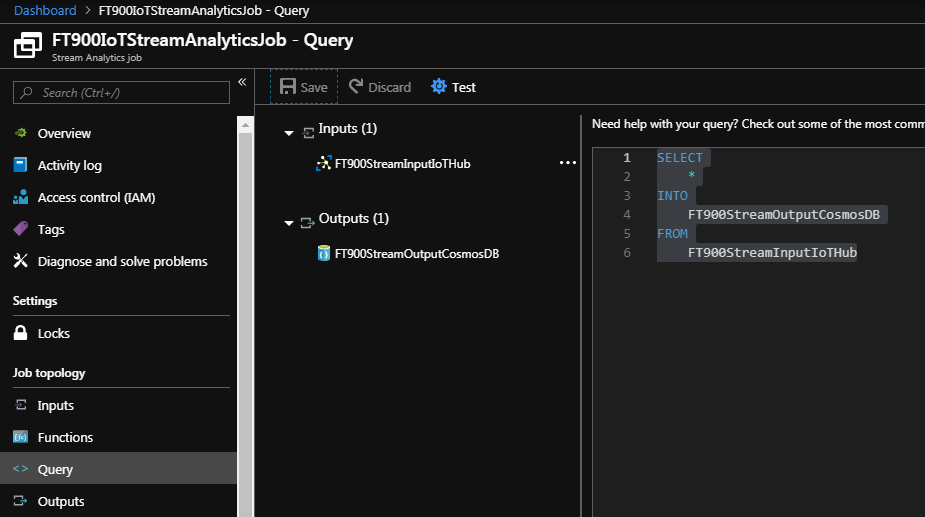
FT900StreamOutputCosmosDB

FROM

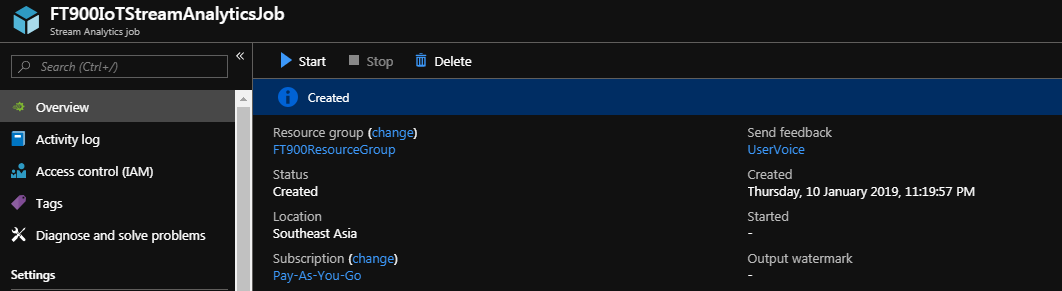
FT900StreamInputIoTHub

Click “Save” button

Check that it should look like this below

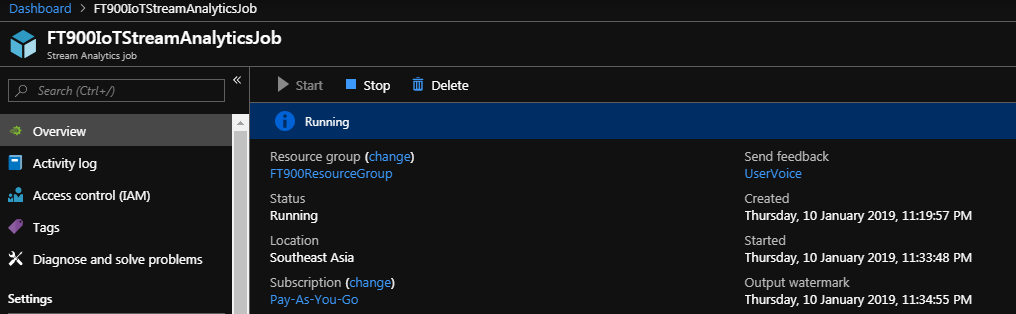


Click “Start” button > Start



Wait for the Streaming job to start successfully! Status will change from Starting to Running

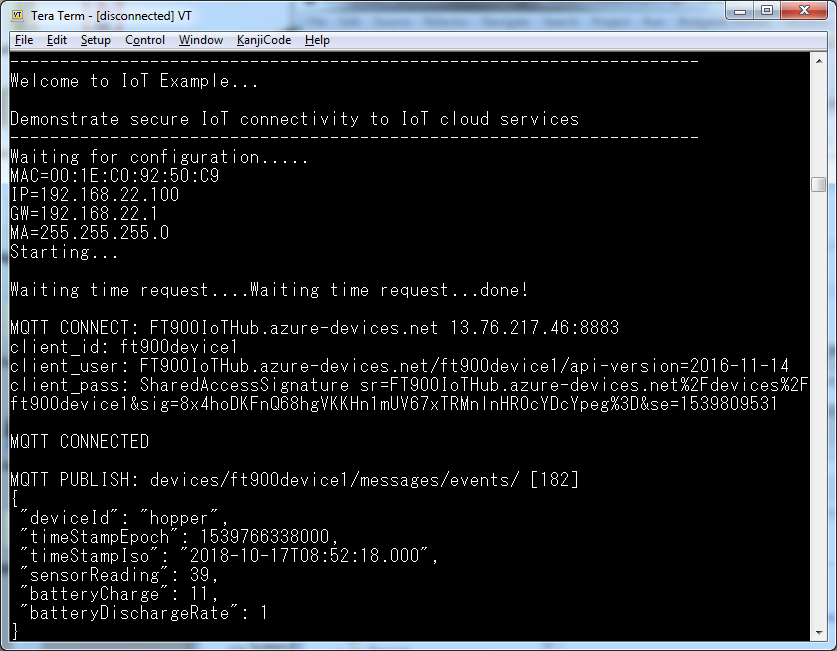
This will take about 2-3 minutes.



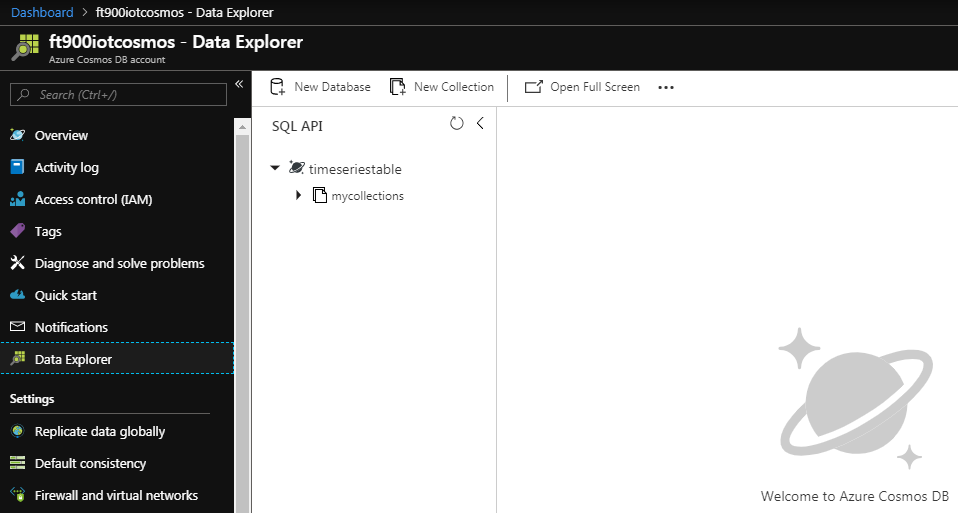
1. Verify FT900 MQTT messages are saved in Cosmos DB.

1. Run FT900 IoT Demo for 1 minute.

(make sure MQTT\_BROKER\_MAZ\_IOT and SAMPLE\_DEVICE\_1 is selected)



2. Go to CosmosDB > Data Explorer

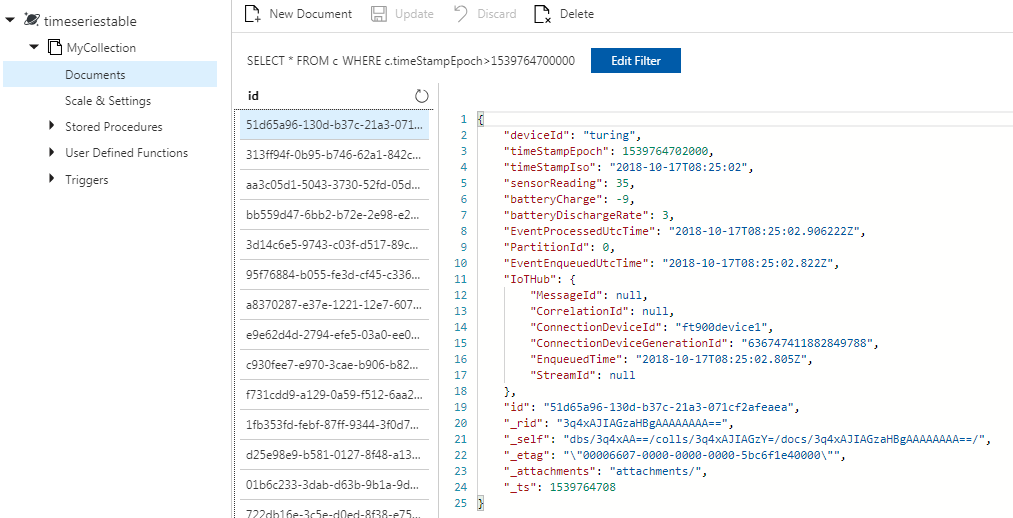


Under timeseriestable > mycollections > Documents

run the following query “SELECT \* FROM c”

Verify that the data appears as screenshot below.

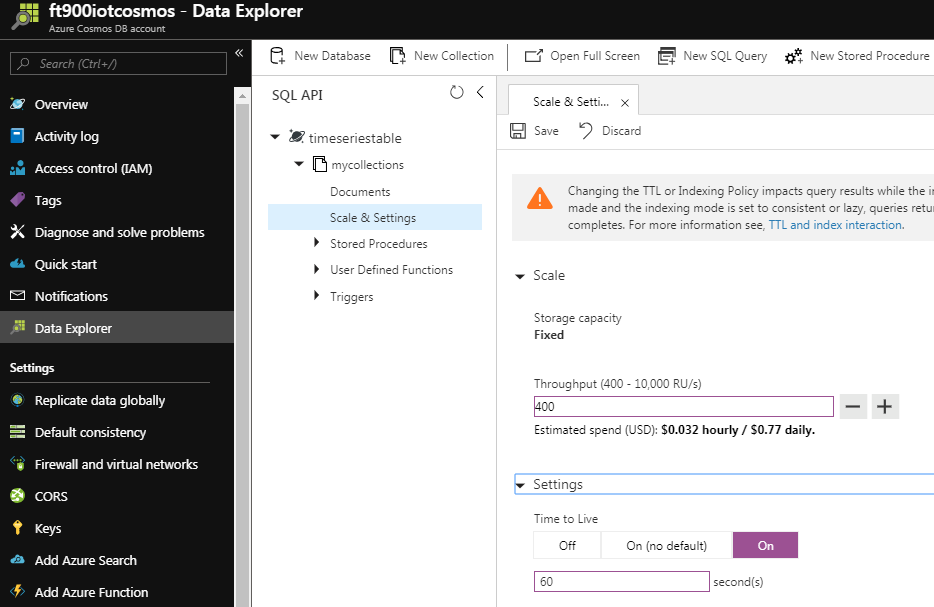
These are the packets sent by FT900 device



3. NOTE: To save database memory, change the following values:

Throughput: 400

Time to Live: On (60 seconds) or depending on the dashboard use-case.



Then click Save button

**III. FRONTEND CLOUD Setup Guide:**

A. Install the prerequisites

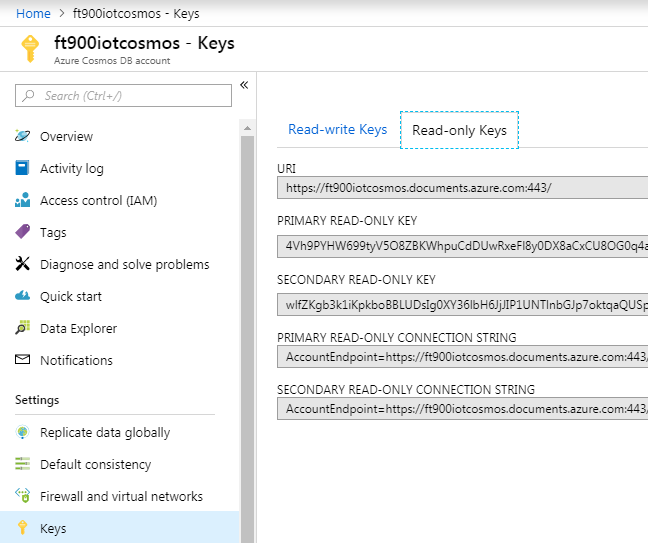
1. NodeJS

2. Azure Cosmos API

npm install async –g

npm install @azure/cosmos -g

B. Get the access in key CosmosDB



C. Update and run the Node.JS script “cosmosdbclient.js”

1. Update the values for

endpoint = "https://ft900iotcosmos.documents.azure.com:443/";

masterKey = " 9eAPFh6cZ4geusIykcyvz1Epy2y95A7e1hqSk13XFdgvEeHoUF92g5YjvJHFhC3EIVBjG0QQnFwfJ5FfsoHcbA==";

databaseId = "timeseriestable";

containerId = "mycollections";

2. Run FT90x for 1-2 minutes

3. Run cosmosdbclient.js using “node cosmosdbclient.js”

