Wind Farm POV Documentation

1. Update the **WeatherStationArchitecture** Architecture Diagram and add subsystems that satisfies the requirements (use PowerPoint or Visio). Save this version as **WindFarmArchitecture-[YOUR STUDENT ID]** for assessment.

2. Update the **WeatherStationThreatModel** threat model and note any key factors that drive configuration of the solution.

In short:

Wind Farm -> IOT Hub -> Stream Analytics -> CosmosDB -> PowerBI

3. Install **Wind Farm Dashboard** App as directed in the **Lab Setup** task in the **Introduction**.

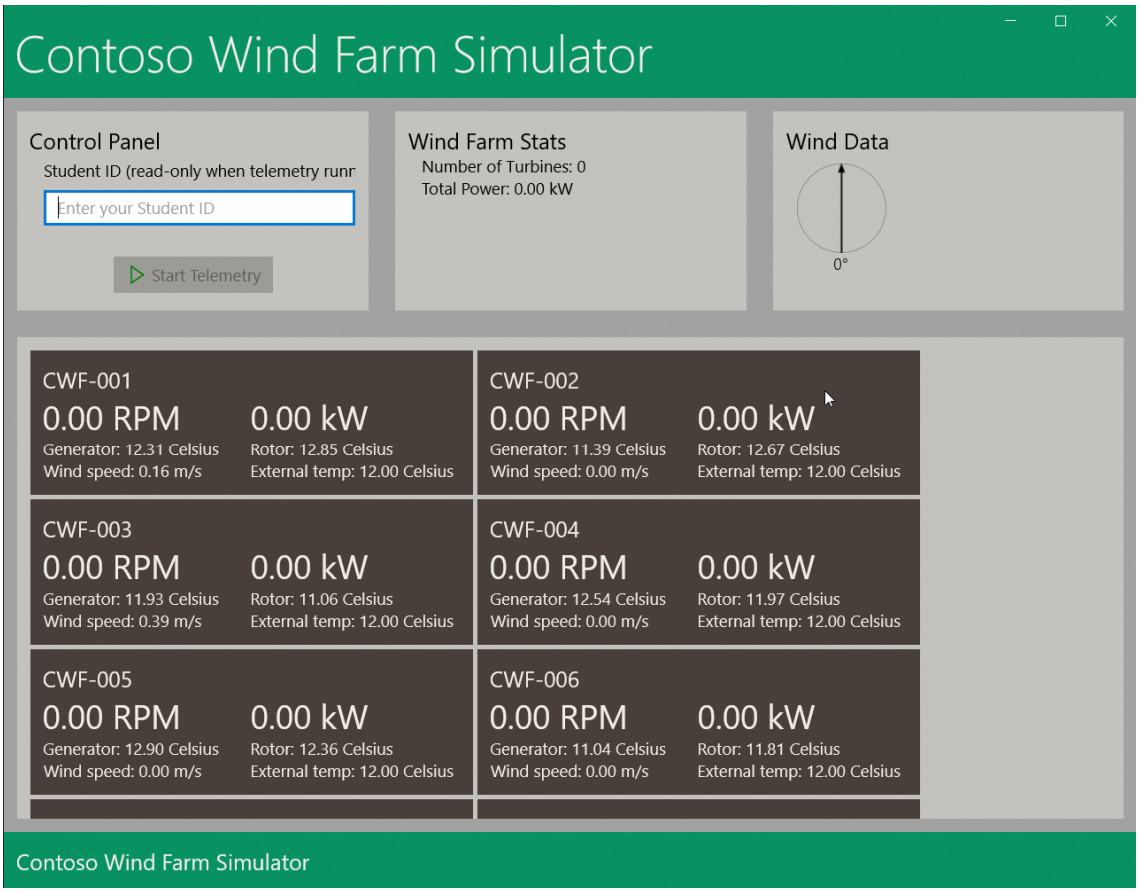
4. To install the **Wind Farm Dashboard** UWP application, download the zip file below and extract the contents to a directory.

5. Within the directory, find the **Add-AppDevPackage.ps1** file - this PowerShell Script will install the application using “side-load”

6. Right-click the **Add-AppDevPackage.ps1** file, and click **Run with powershell** from the context menu.

7. Enter **R** to run the script once.

8. Hit **Enter** to continue and close the PowerShell window.



9. Connect Wind Farm Dashboard App to IoT Hub.

   You will need to create 10 devices:

      CWF-001.

      CWF-002.

      CWF-003.

      CWF-004.

      CWF-005.

      CWF-006.

      CWF-007.

      CWF-008.

      CWF-009.

      CWF-010.

10. Review the following materials:

   Course: DEV326x IoT Data Analytics and Storage

   Module: Advanced Analytics

   Lab: Device Management and Analytics

   Topic: Managing IoT Devices, tags and desired configurations with IoT Hub

and

   Course: DEV312x Business Intelligence for IoT Solutions

   Module: Time Series Data

   Lab: Producing Simulated Data

   Topic: Set up the device simulation

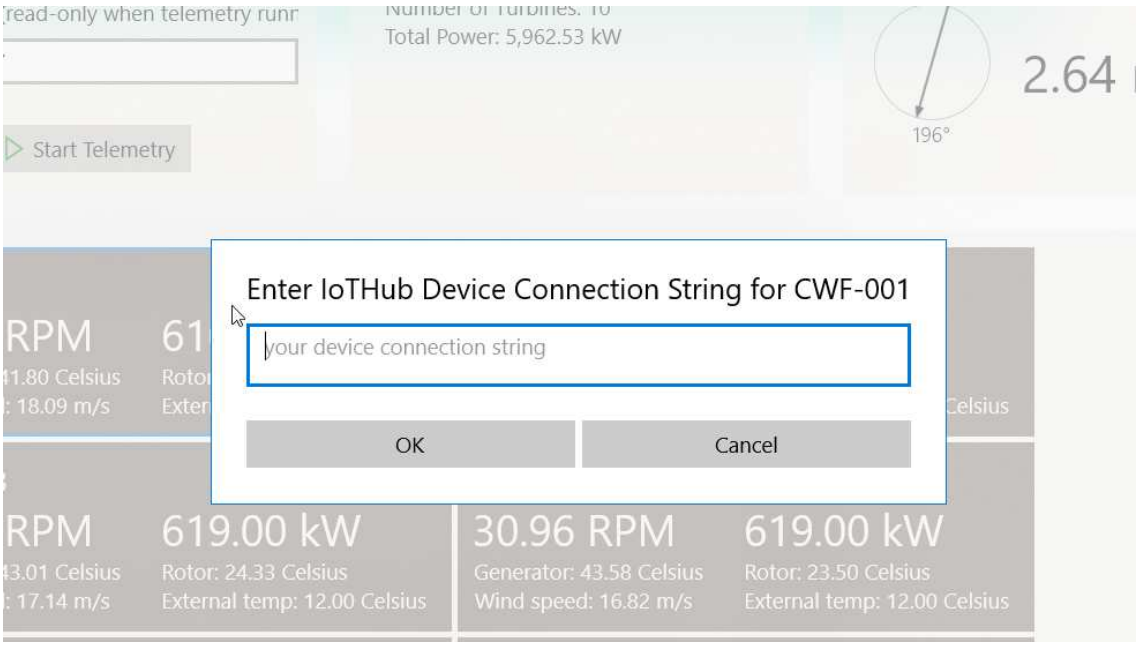
DEVICE ID	STATUS	LAST ACTIVITY TIME (UTC)	LAST STATUS UPDATE (UTC)	AUTHENTICATION TYPE	CLOUD TO D.
CWF-005	Enabled	--	--	Sas	0
CWF-MX-001	Enabled	Jul 4, 2019 5:43 AM	--	Sas	0
CWF-010	Enabled	--	--	Sas	0
CWF-004	Enabled	--	--	Sas	0
CWF-003	Enabled	--	--	Sas	0
CWF-008	Enabled	--	--	Sas	0
CWF-002	Enabled	--	--	Sas	0
CWF-006	Enabled	--	--	Sas	0
CWF-007	Enabled	--	--	Sas	0
CWF-009	Enabled	--	--	Sas	0
CWF-001	Enabled	--	--	Sas	0

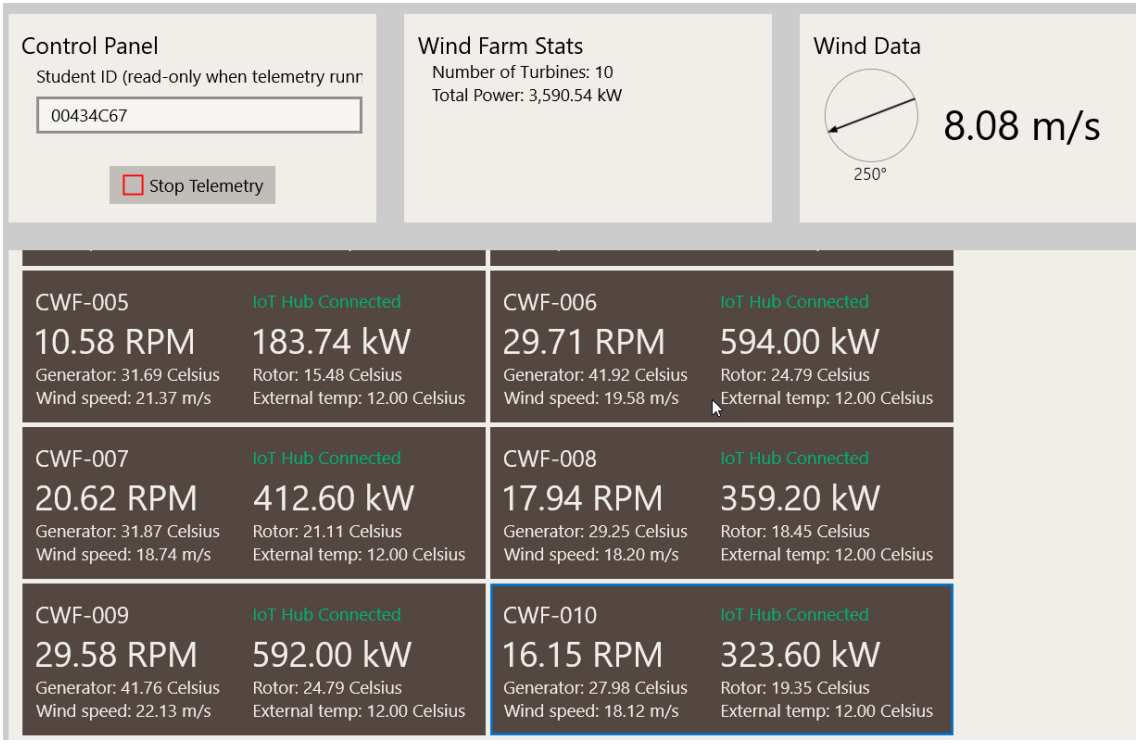
11. Start the **Wind Farm Dashboard** App

12. Enter your student ID in to the appropriate field - this will be saved between sessions so you wil only need to do this once.

13. Click on each of the Turbine tiles - a dialog will be displayed. Copy the **Device Connection String** for each device into the dialog and close it - this connection string will be saved between sessions so you will only need to do this once.

14. Start the Telemetry so that data begins to be sent to your IoT Hub - note that each turbine should display an **IoT Hub Connected** message. **Let the telemetry run at least one hour**.





15. Create a Time Series Insights resource and configure it to connect to the IoT Hub.

    Note: Pay attention to the SKU you select - TSI can get expensive quickly.

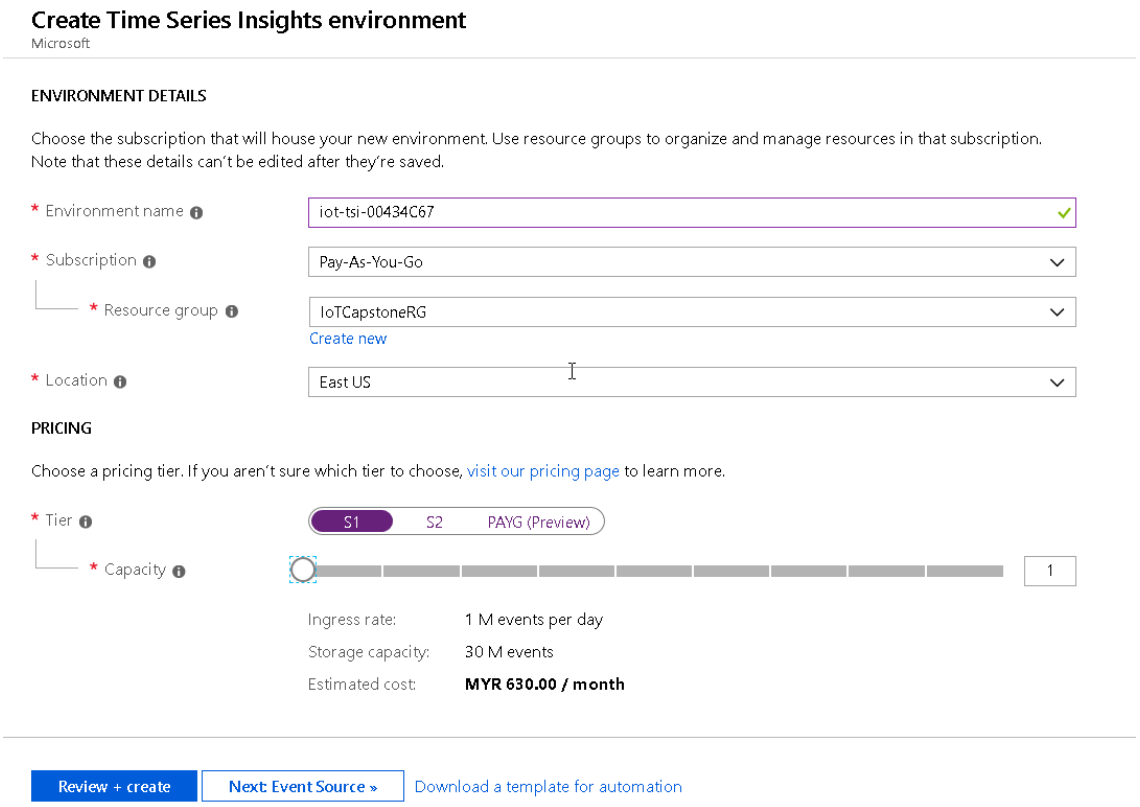
    Tip: Review the following materials:

        Course: DEV312x Business Intelligence for IoT Solutions

        Module: Time Series Insights

        Lab: Provisioning Time Series Insights

16. Under **Sku**, select Standard 1 (**S1**)., Under **Capacity**, select 1.



17. Select **Next: Event Source** to advance to the **Event Source** tab.

18. Make certain **Yes** is selected under **Create an event source?**

19. Under **Name**, enter **IotHubSource**.

- Under **Source type**, select **Iot Hub**.

- Under **Select a hub**, select **Select existing**.

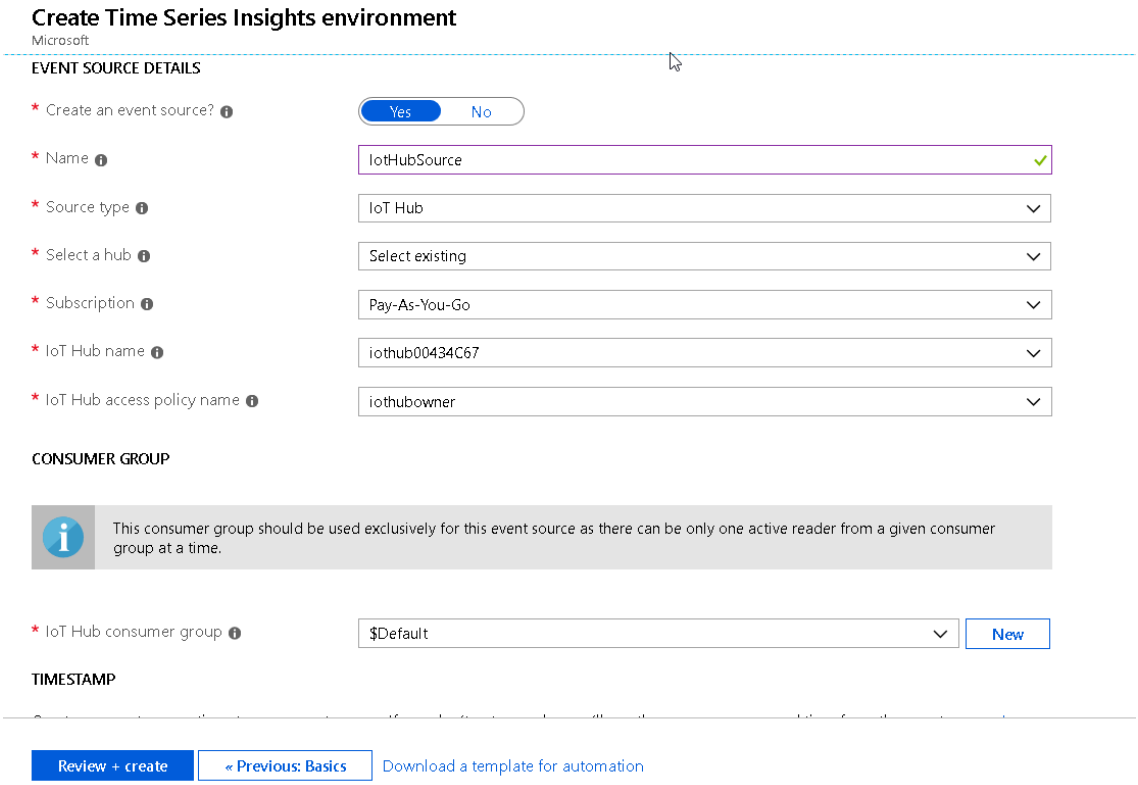
- Under **Subscription**, make certain to select the subscription you used to create your IoT Hub in the last lesson.

- Under **Iot hub name**, select the IoT hub that you created for this module (e.g. **IoTBIHubxx**).

- Under **Iot hub policy name**, select **iothubowner**.

- Under **IoT Hub consumer group**, select **$Default**.

- Under **Timestamp property name**, leave the textbox blank.



20. Use the Time Series Explorer to create the **Interesting Telemetry** query detailed below. Set the query to display the **last 30 minutes**. Use an interval of 2 seconds (or as small as possible if 2 seconds is unavailable).

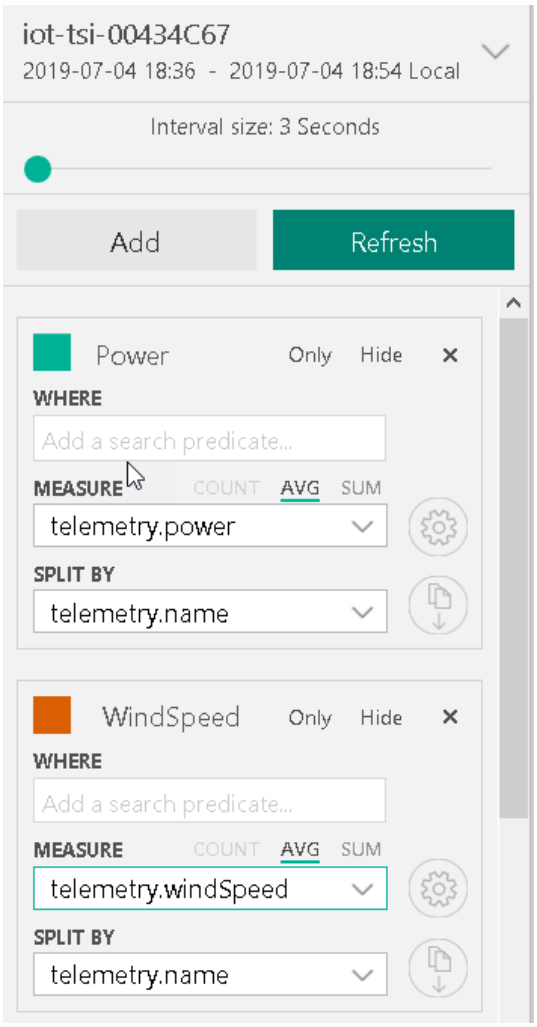
21. Interesting Telemetry. Create a query that displays the following measures:

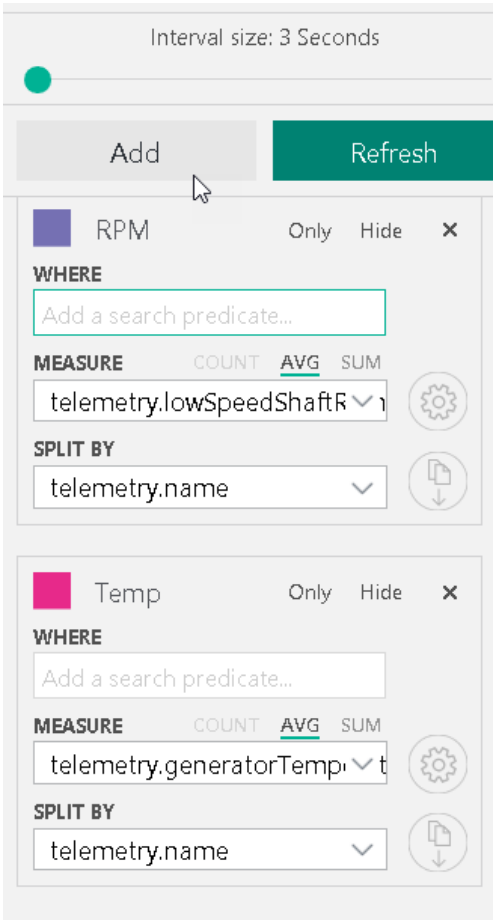
AVG Telemetry.Power SPLIT BY Telemetry.Name with a display name of Power

AVG Telemetry.WindSpeed SPLIT BY Telemetry.Name with a display name of WindSpeed

AVG Telemetry.LowSpeedShaftRpm SPLIT BY Telemetry.Name with a display name of RPM

AVG Telemetry.GeneratorTemperatureCelsius SPLIT BY Telemetry.Name with a display name of Temp


iot-tsi-00434C67
2019-07-04 18:36 - 2019-07-04 18:54 Local
Interval size: 3 Seconds
Add
Refresh
Power
Only
Hide
WHERE
Add a search predicate...
MEASURE
COUNT
AVG
SUM
telemetry.power
SPLIT BY
telemetry.name
WindSpeed
Only
Hide
WHERE
Add a search predicate...
MEASURE
COUNT
AVG
SUM
telemetry.windSpeed
SPLIT BY
telemetry.name



22. Examine the displayed data - do you notice a particular turbine and telemetry that appears to be operating outside of the others?

23. DELIVERABLE: Select the turbine and data element that seems to be a problem and take a screen shot of the view and save it **as InterestingTelemetry-[YOUR STUDENT ID]-anomaly.png** for submission. Save this file in the Lab2 folder within your GitHub repository (ensure you add, commit and push your changes). Ensure that the screenshot includes the following:

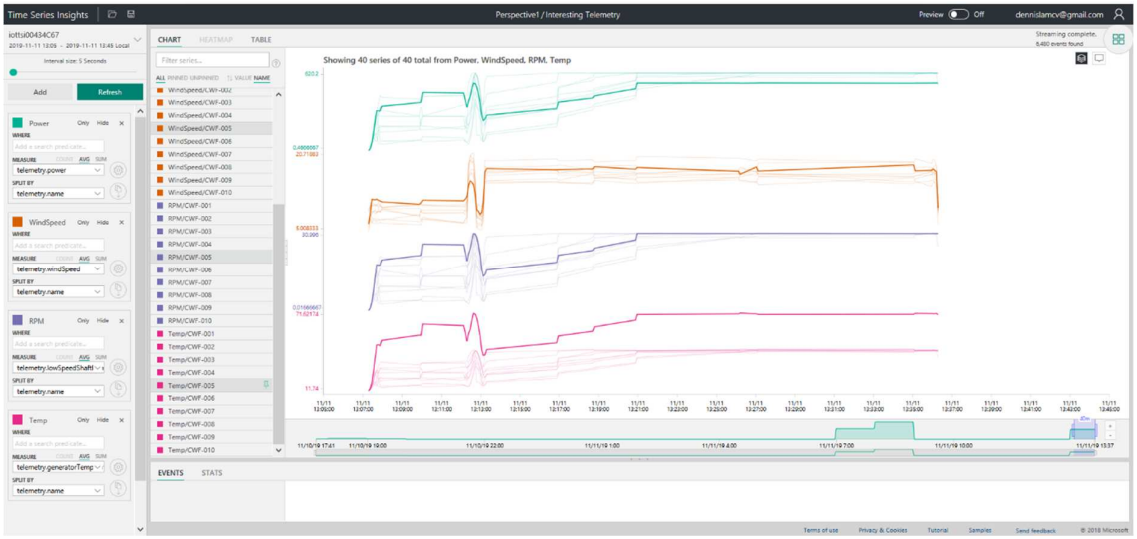
All of the terms on the left hand side of the view.

The charts should be displayed stacked.

The chart title should be visible.

The charts should be displaying at least 10 minutes of data.

The suspect telemetry term and turbine should be selected.



24. Create another query and name **is SuspectTelemetry**. Set the query to display the last 30 minutes. Use an interval of 2 seconds (or as small as possible if 2 seconds is unavailable). Add two terms to the query:

The AVG measure you suspect is out of the ordinary, named the same as the measure in the Interesting Telemetry query. Add a WHERE clause that only displays the suspect turbine:

The WHERE clause will be similar to [telemetry.name] has 'CWF-XXX'.

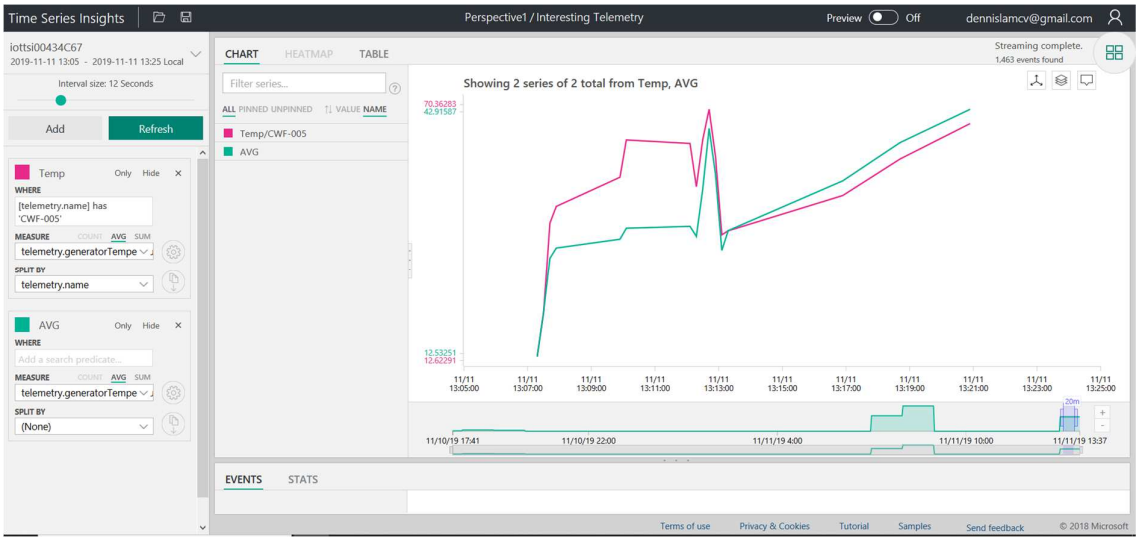
The same AVG measure you added above without a WHERE value or SPLIT BY - this will display the average for that measure across all turbines. Name this measure AVG.

25. DELIVERABLE: Take a screen shot of the SuspectTelemetry view and save it as **SuspectTelemetry-[YOUR STUDENT ID].png** for submission. Save this file in the Lab2 folder within your GitHub repository (ensure you add, commit and push your changes). Ensure that the screenshot includes the following:

All of the terms on the left hand side of the view - ensure the name of the turbine and the measure is clearly visible.

The chart title should be visible.

The chart should be displaying at least 10 minutes of data.



25. Create and configure CosmosDB for use as your Warm Storage.

Note: The **Wind Farm Dashboard** will stream a large volume of data that will exceed the default 400 RUs capacity of a container. Set the RUs to 5000 to ensure all the data is captured. Remember to delete the resources when you are no longer using them as you will incur more cost with the higher RUs.

26. Configure your Azure Streaming Analytics instance to stream telemetry to CosmosDB.

    Confirm data is flowing to Cosmos DB

    Tip: Review the following materials:

        Course: DEV326x IoT Data Analytics and Storage

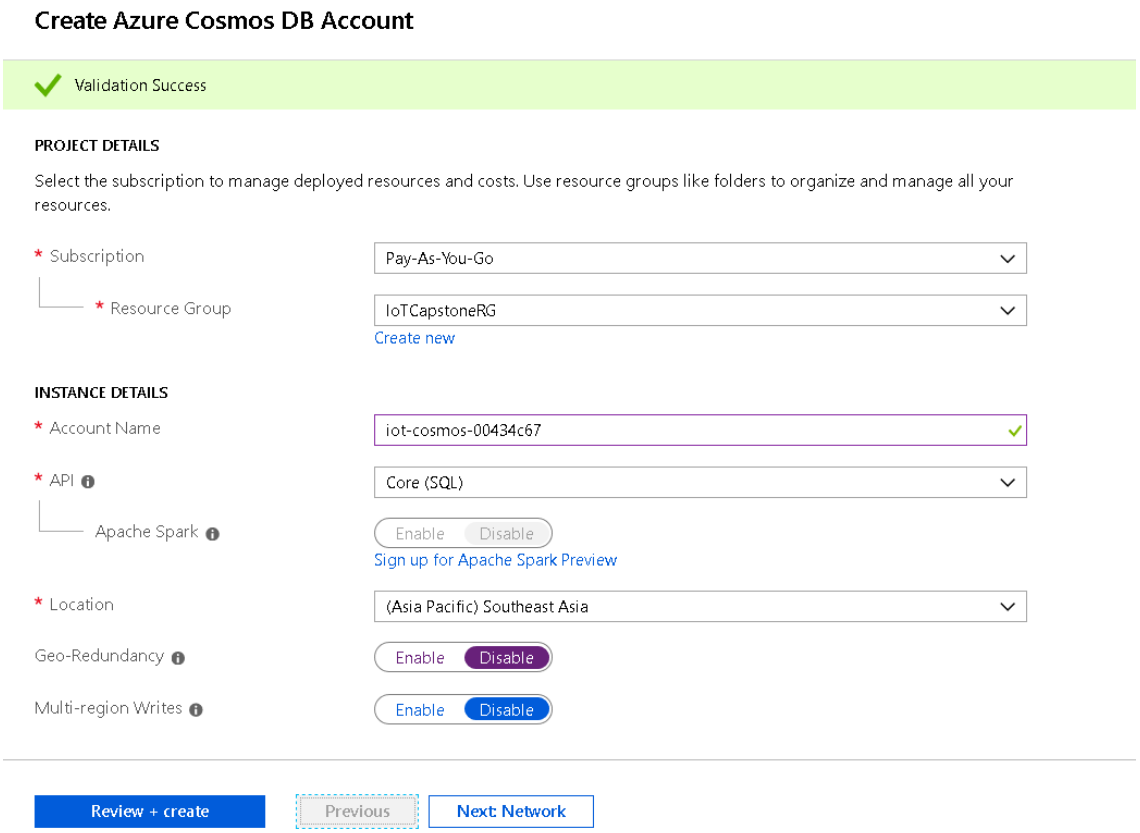
        Module: Warm Storage

        Lab: Getting Started with Warm Storage

27. On the **New Account** blade, enter a unique name to use for the **ID** field.

28. Under **API**, select **SQL**

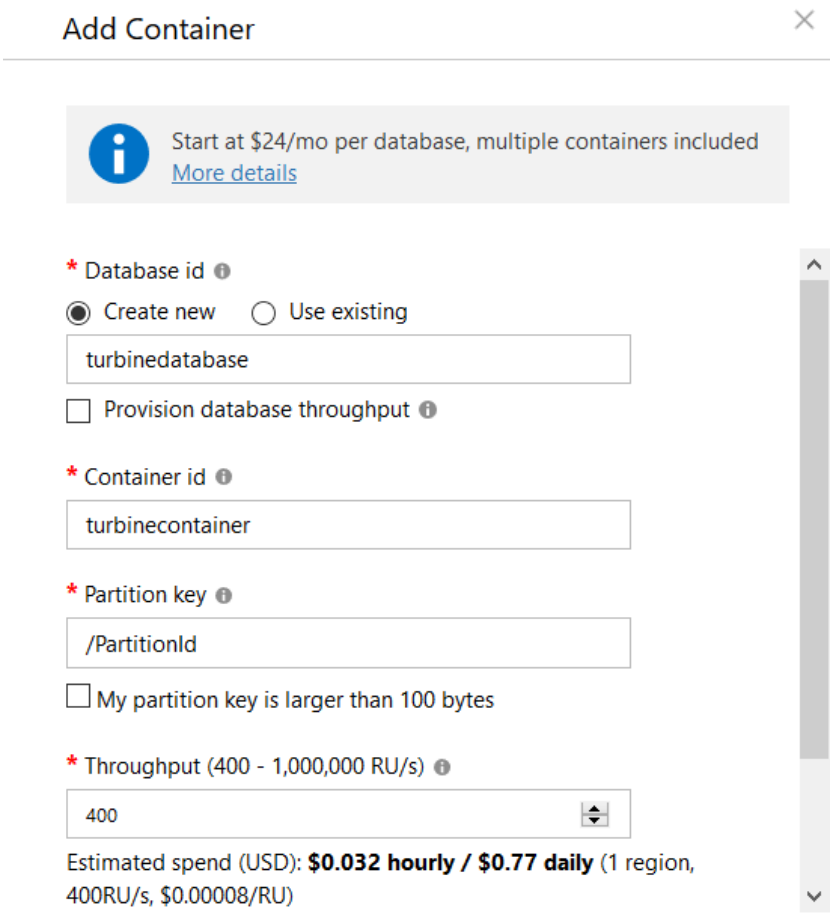
29. make sure **Enable geo-redundancy** is unchecked.



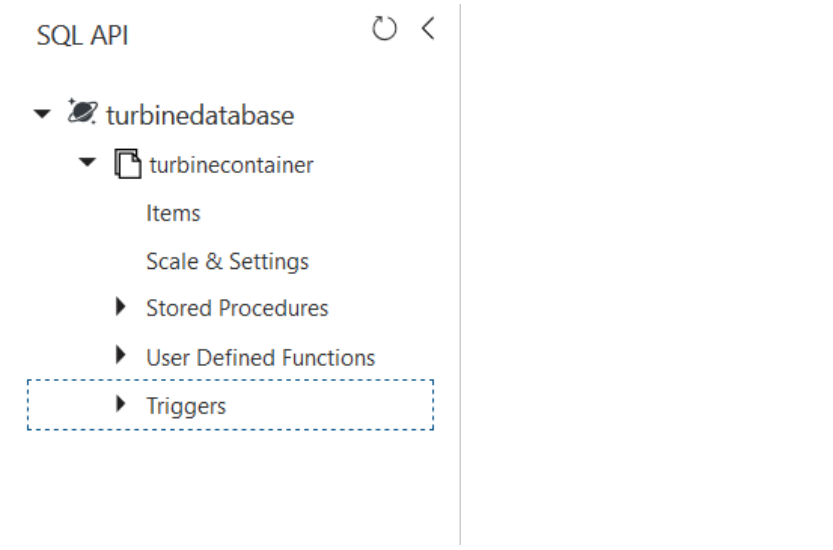
30. In Cosmos DB, data is stored within a container. Complete the following steps to add a container.

The most critical part is Partition key because if it's wrong the database has no items inside.

31. Wait and look at new directory created.

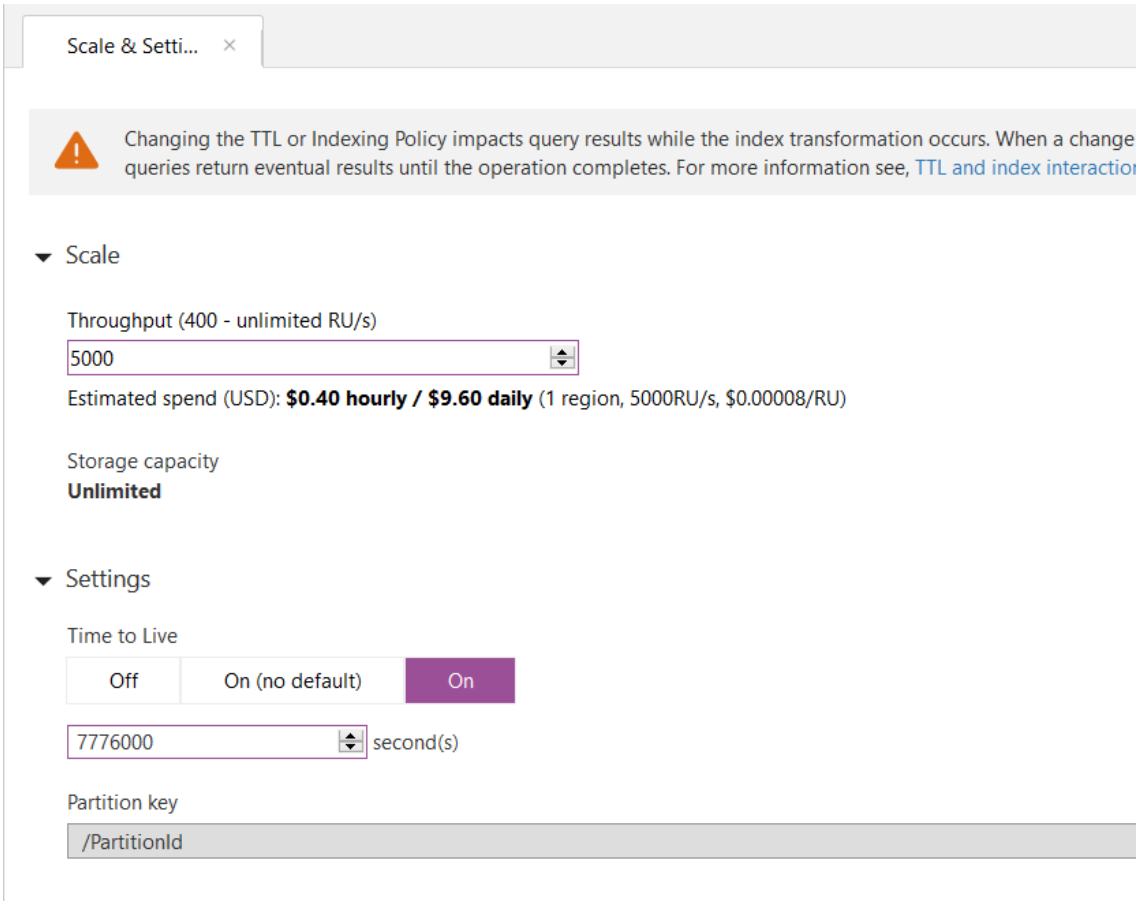


After setup, it looks like this:



On the **Settings** tab, under Settings, in Time to live, click On

Under **Time to live**, in second(s), enter **7776000**



32. Delete the previous outputs in Stream Analytics.

33. Add Cosmos DB as output.

At this time, Azure Stream Analytics only supports unlimited containers with partition keys at the top level. For example, `/region` is supported. Nested partition keys (e.g. `/region/name`) are not supported.

https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-documentdb-output

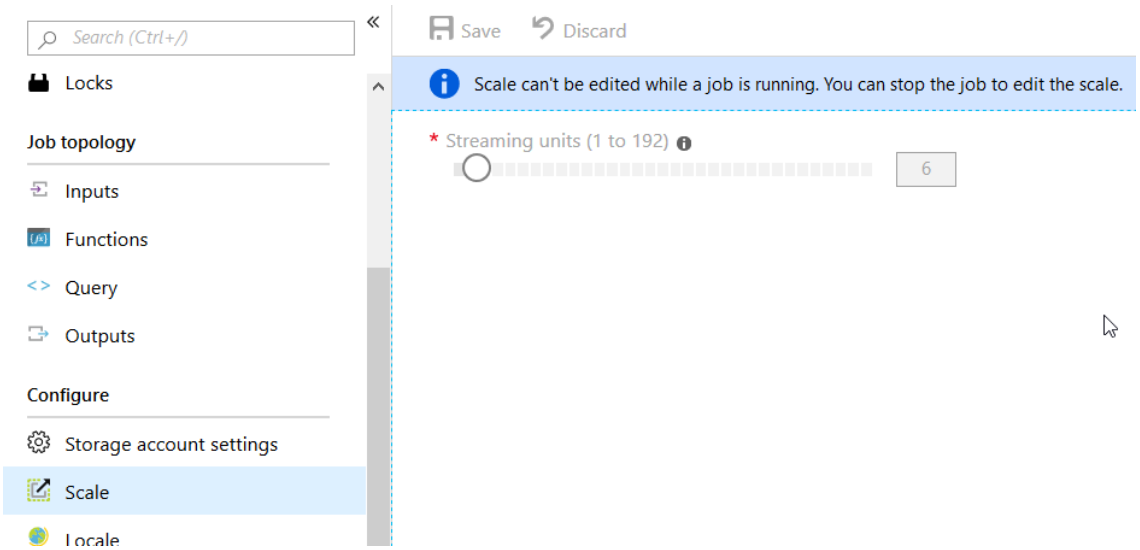
34. Refer to this link: https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-documentdb-output

35. Configure your Azure Streaming Analytics instance to stream telemetry to CosmosDB.

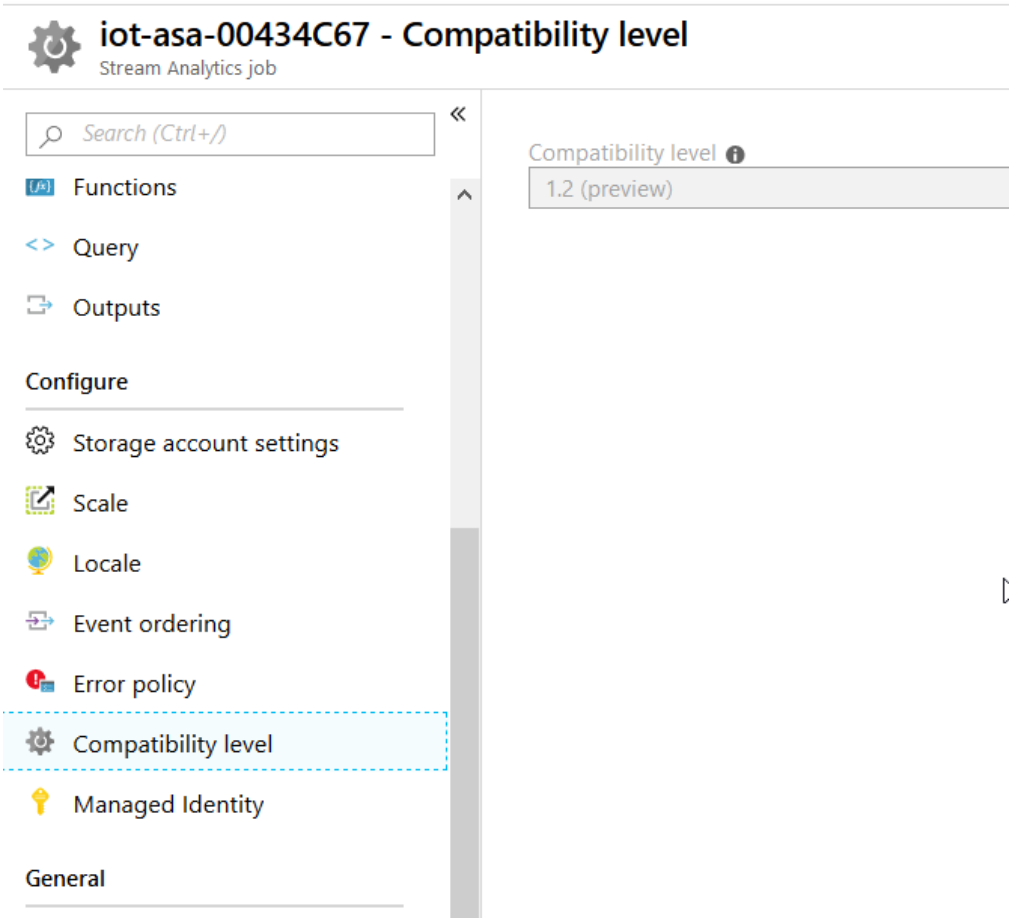
----------UPDATE NOV 2019 (IGNORE SOME INSTRUCTIONS)----------

36. Before streaming there are **some modifications** need to be done.

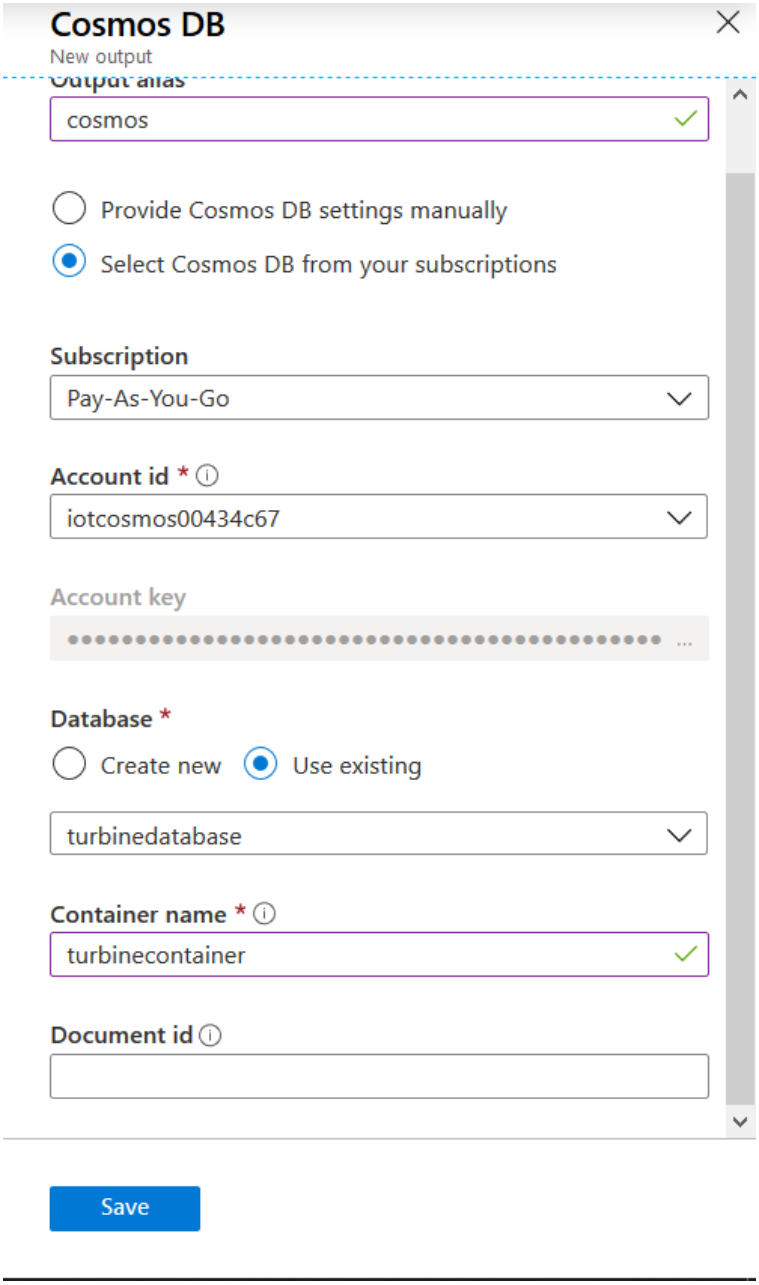
37. Scale up Azure Stream Analytics (max to 6) for faster processing



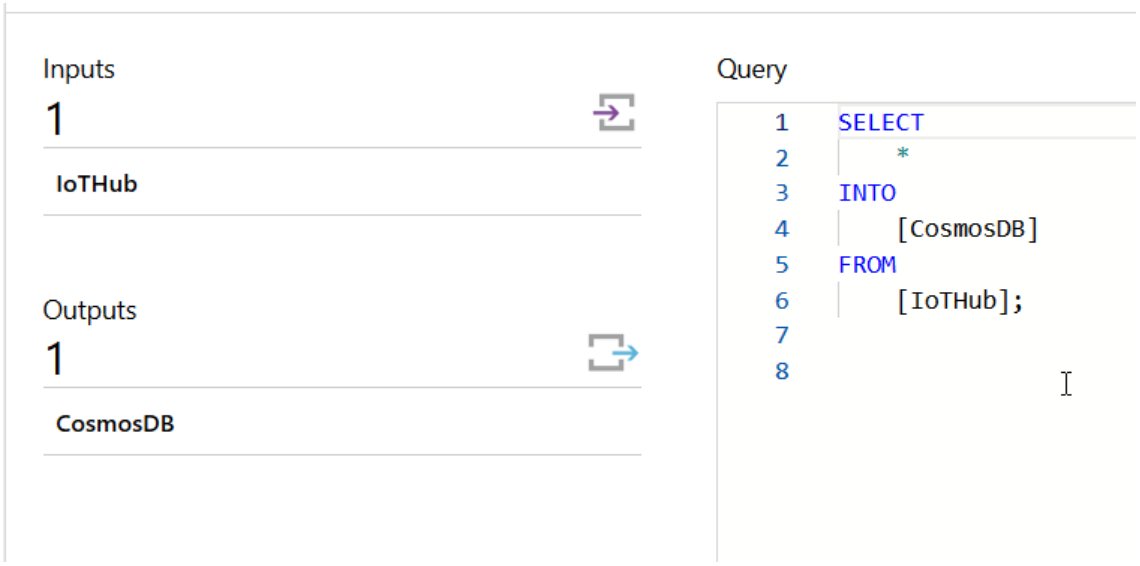
37. Raise compatibility level to v1.2 for SQL queries.



37a: Add CosmosDB output to Stream Analytics

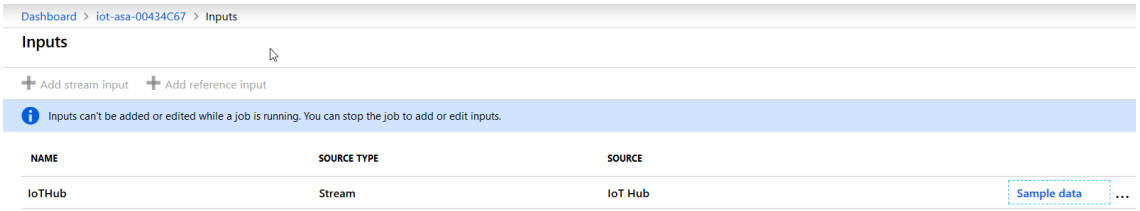


Check and make sure Input and Output Connectivity is OK

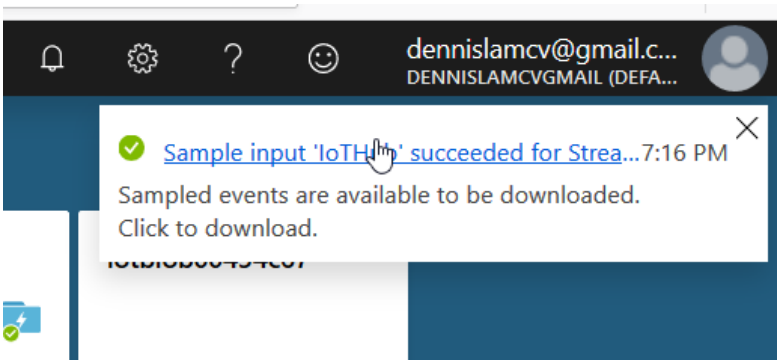


----------IGNORE THIS PORTION ----------

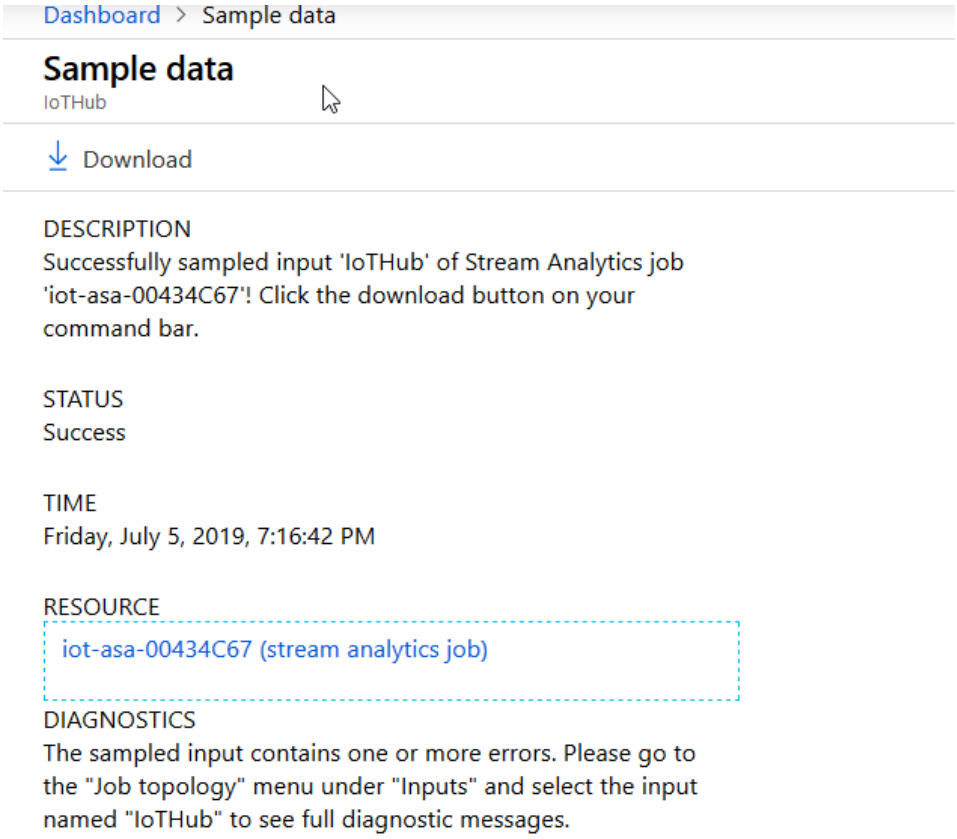
38. We also check the input telemetry by Sample Data from IOT Hub here:



39. Click on the link after sampling finished.



40. Download the file.



41. At the top of the Sample data blade, click Download

41a. When prompted, save the sample data a local folder location.

The sample data is formatted as JSON file and has a record containing values. We will be using Visual Studio Code to view the data.

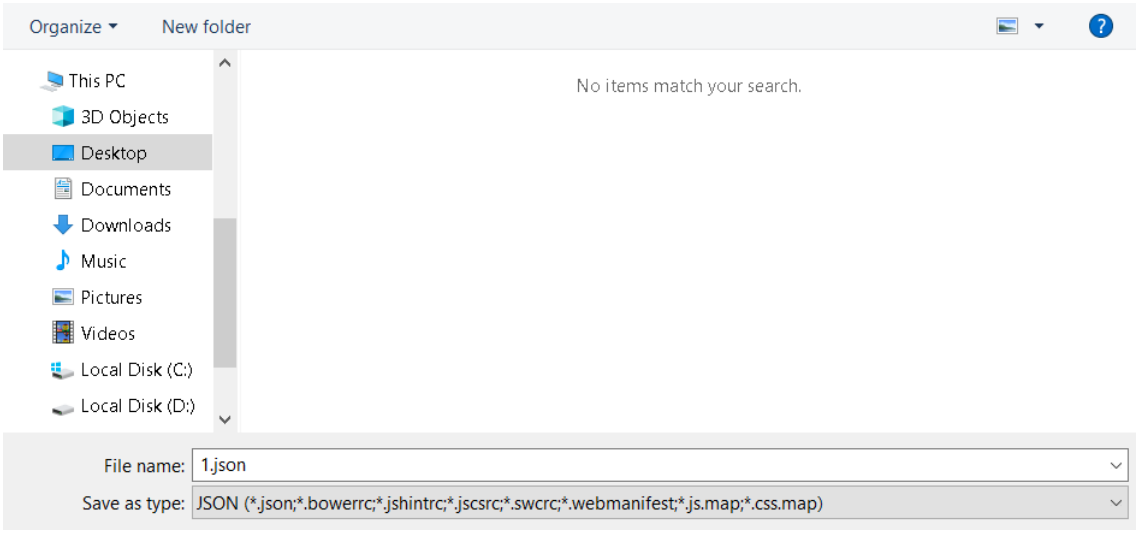
41b. Open Visual Studio Code, and then open the sample data file that you just saved.

41c. On the File menu, click Save As

The file is saved with a .txt file extension, or with no extension at all. We will change that file extension to JSON, which will help VSCode to format the data for us.

On the Save As dialog, open the Save as type list, click JSON, and then click Save

With the file extension changed to JSON, VSCode understands how to format the file for us.



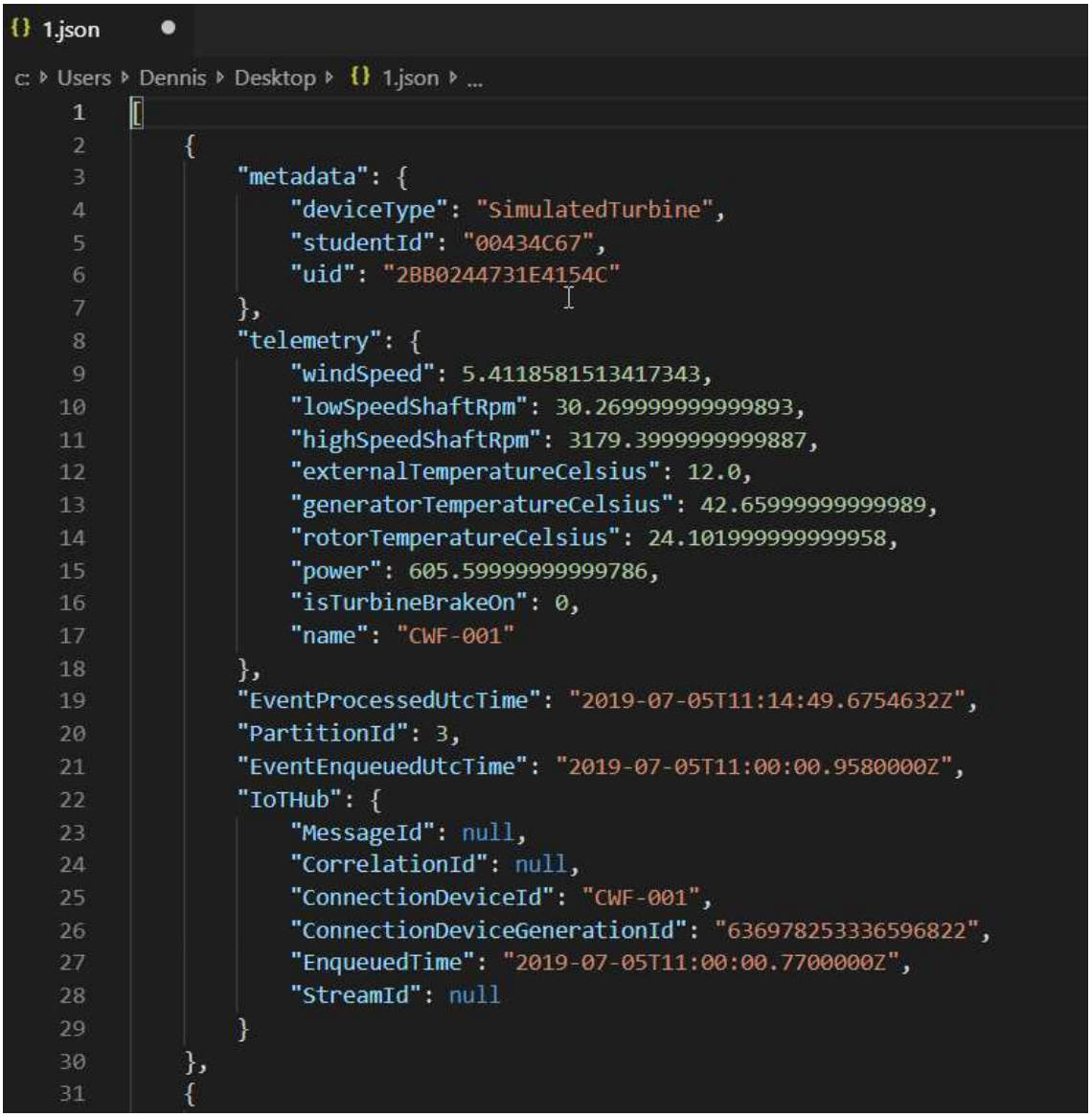
41e. To format the data file, use the keyboard shortcut appropriate to your platform:

On Windows: Shift + Alt + F

On Mac: Shift + Option + F

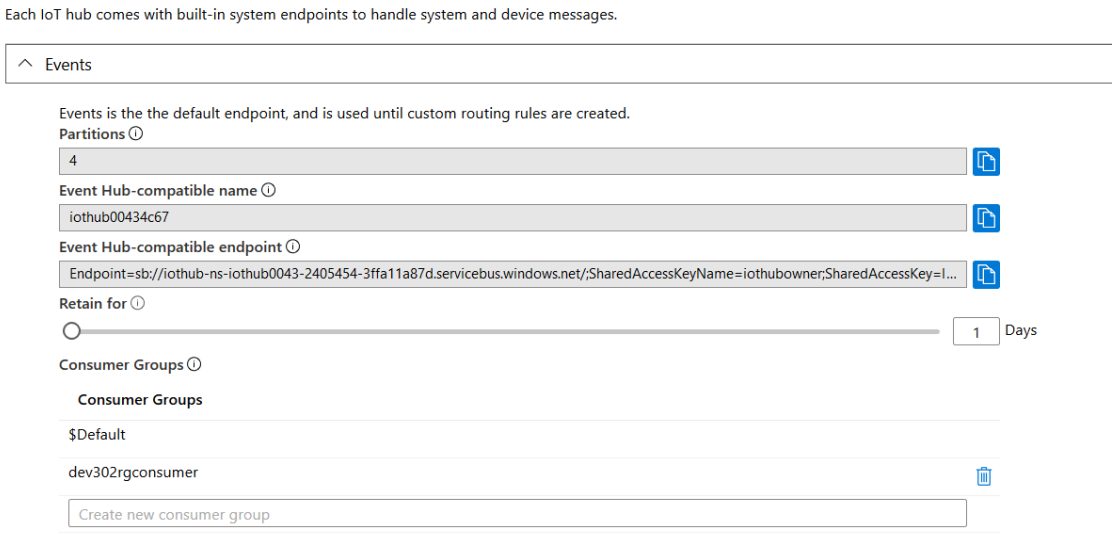
On Ubuntu: Ctrl + Shift + I

42. This is to check if the telemetry is correct

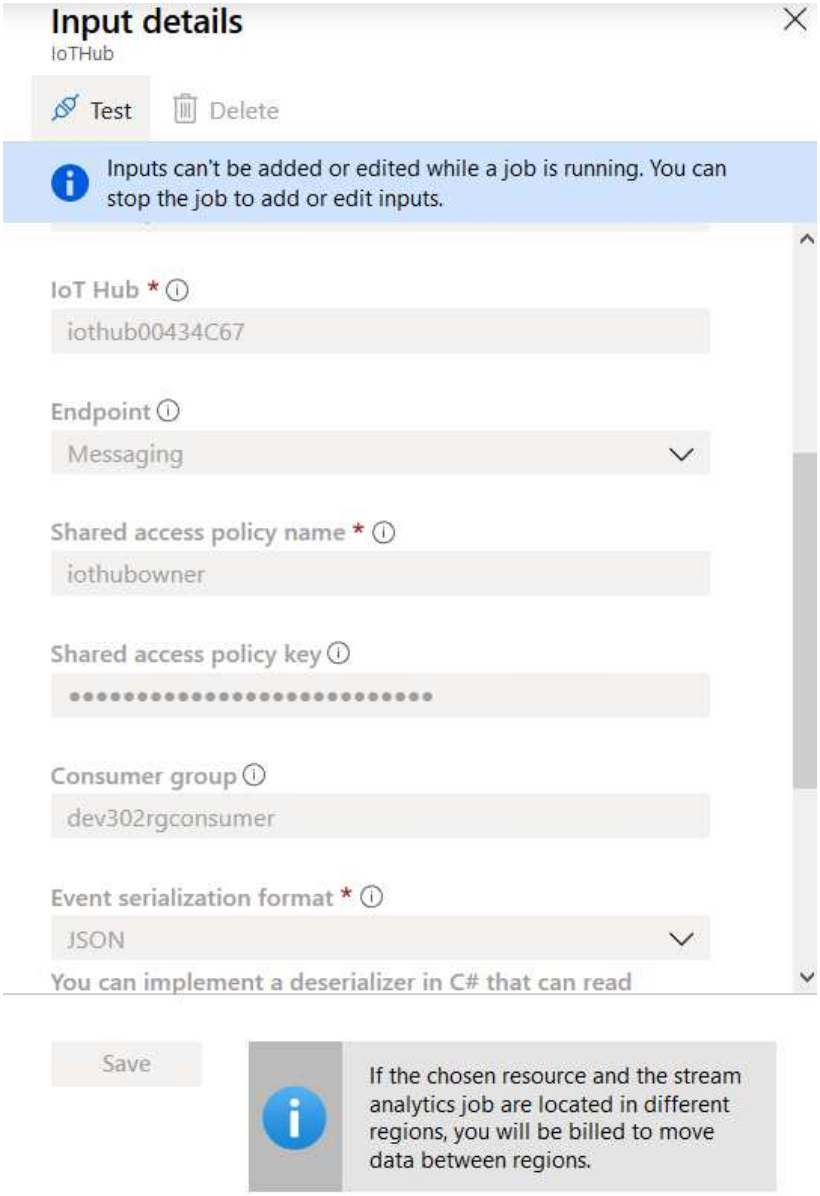


---

43. Go to IOT Hub -> Built in endpoints. Create a new consumer group for CWF-001 to CWF-010 devices streaming usage.



44. On Azure Streaming Input, make sure **dev302rgconsumer** is selected and Save.

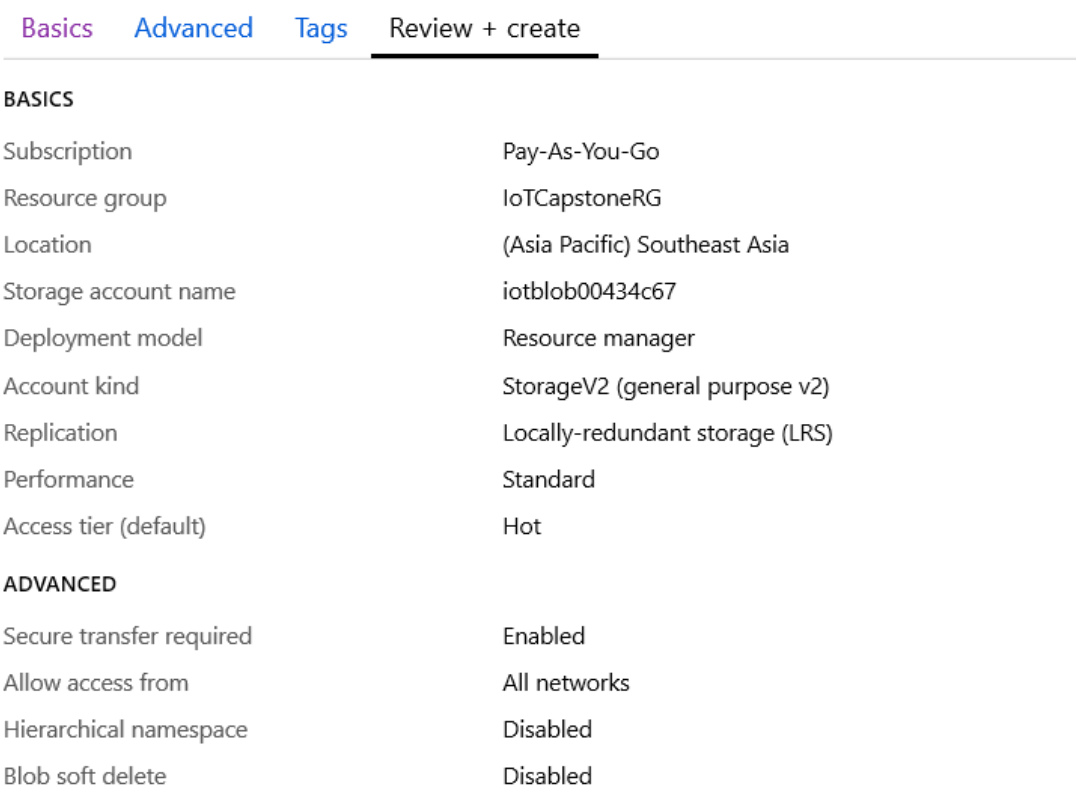


----------IGNORE THIS PORTION ----------

44a. Created one blob storage account as “backup”. Set Access Tier as Hot.

45. In the left hand nav area, under Settings, click Access keys.

The Access keys pane opens. You will see the name of the storage displayed as well as the key and connection string information for two keys. Having two keys available supports key rotation scenarios - resources can be using the key2 data to connect to the storage, allowing the key1 data to be regenerated, etc.



47. Add a route that will send all messages to a blob storage endpoint.

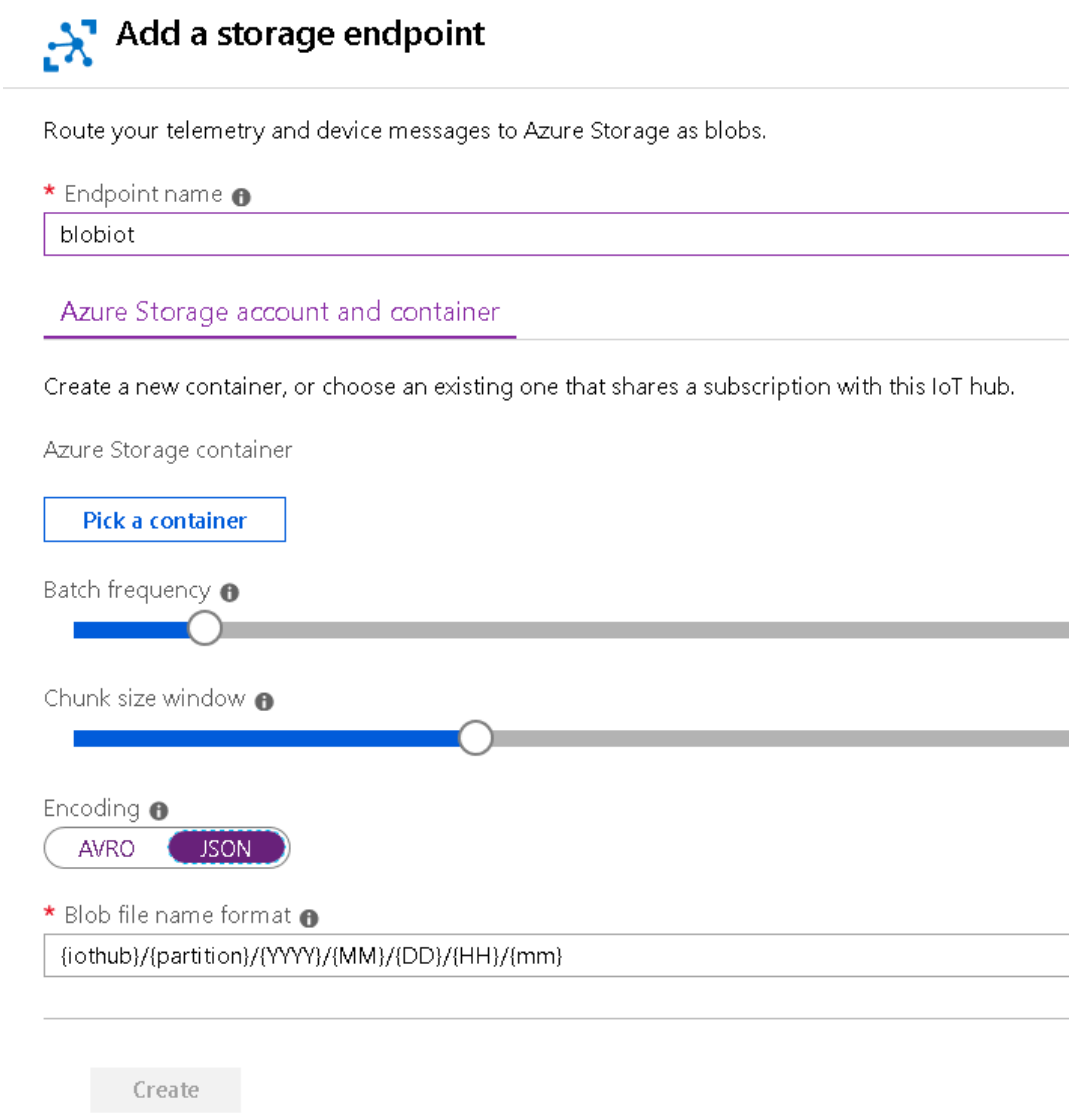
48. In the **IoT Hub** blade left hand nav area, under **Messaging**, click **Message routing**.

49. Ensure the **Routes** tab is selected and click **Add**.

50. In the **Add a route** pane, next to the **Endpoint** selection, click **Add**, then select **Blob storage**.

51. On the **Add a storage endpoint** pane, under **Endpoint name**, enter a name you can remember easily. Select JSON.

52. Click **Pick a container**.



53. From the list of, choose the storage account you created earlier. The **Containers** blade will open, listing the containers within the storage account you just chose. As this is a new storage account, the list is empty - we will add a container now.

54. In the **Containers** blade, in the toolbar, click **+ Container**.

55. In the **New container** popup, under **Name**, enter **allmessages**.

56. Under **Public access level**, select **Private** and then click **OK**.

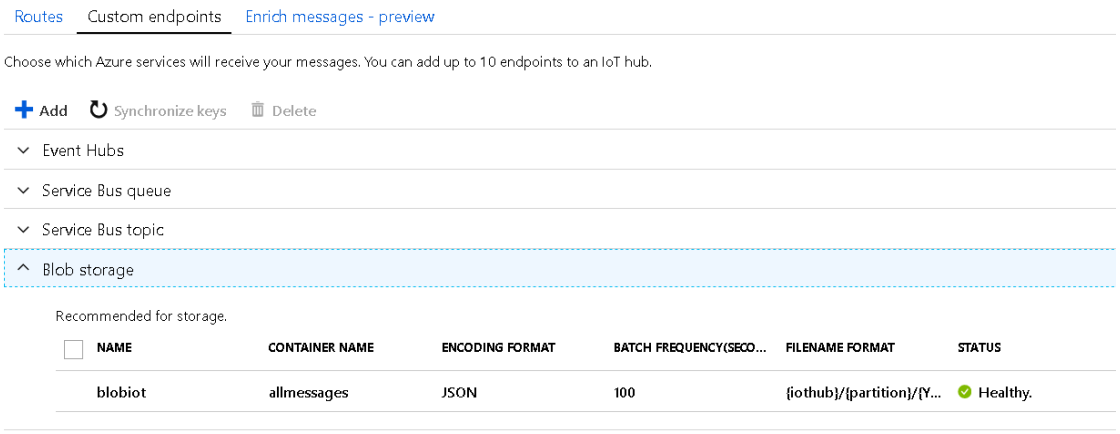
57. In the **Containers** list, click **allmessages**. In the **Containers** blade, at the bottom of the page, click **Select**. The **Containers** blade will close and the **Add a storage** blade is shown. The blade has updated to display the URI for the **Azure storage container** above **Pick a container**.

58. At the bottom of the **Add a storage endpoint** blade, click **Create**.

59. On the **Add a route** blade, under **name**, enter **AllMessages.** Under **Data source**, select **Device Telemetry Messages**.

60. Under **Routing query**, enter **true. I**n **Add a route** blade, at the bottom of the page, click **Save**.

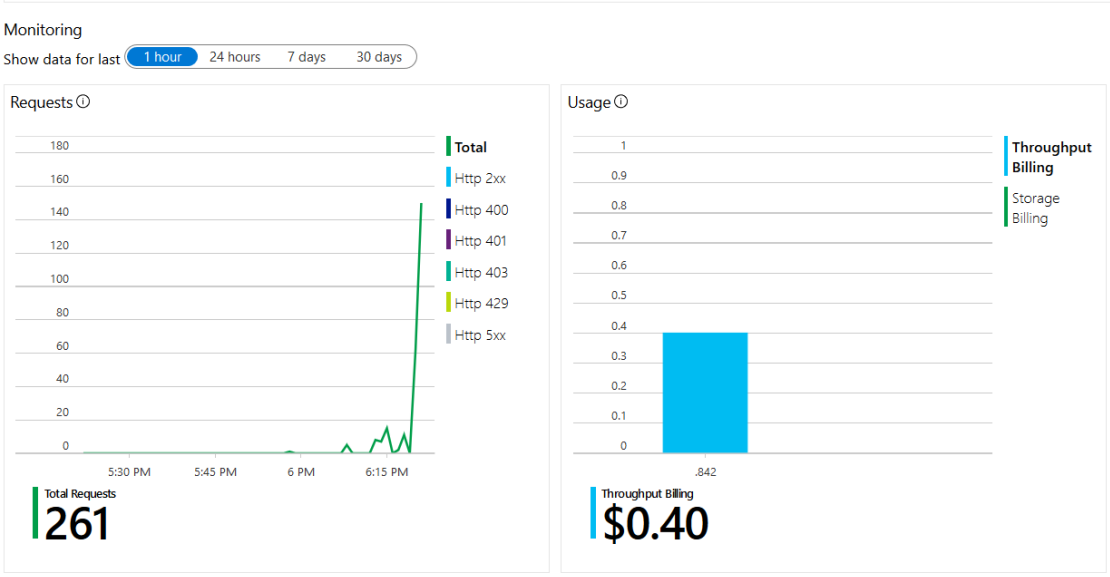
61. If you switch to **Custom endpoints** and click **Blob storage**, you will see the storage endpoint we created (the status may be **Unknown**, wait a few moments and it will show as **Healthy**).

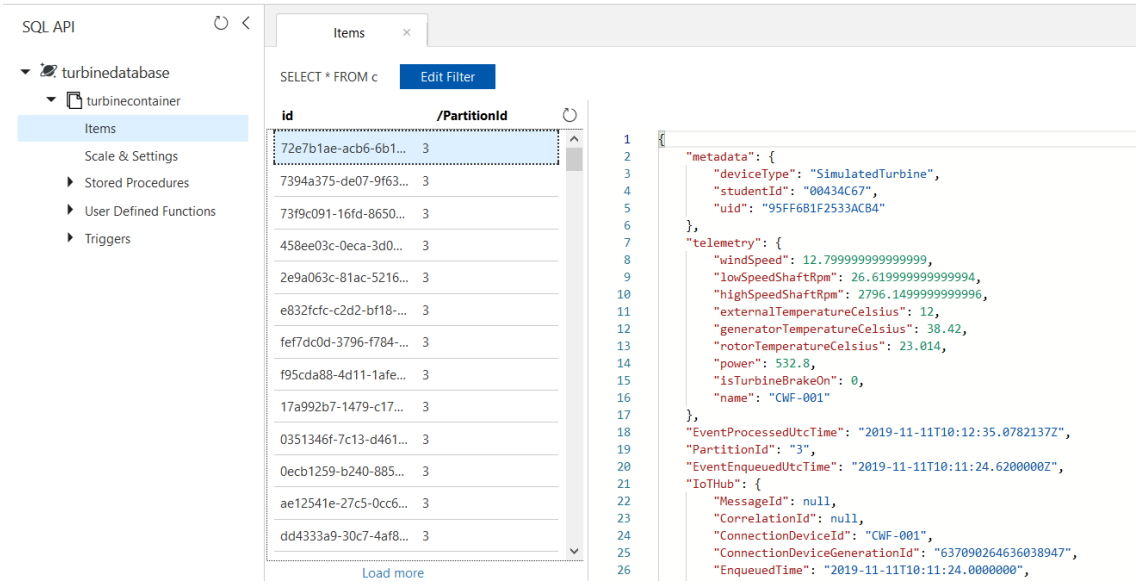


63.Start the wind farm simulator, let it run at least **10 mins**

64. We are at least looking at 4000 to 5000 telemetry counts or more during that period

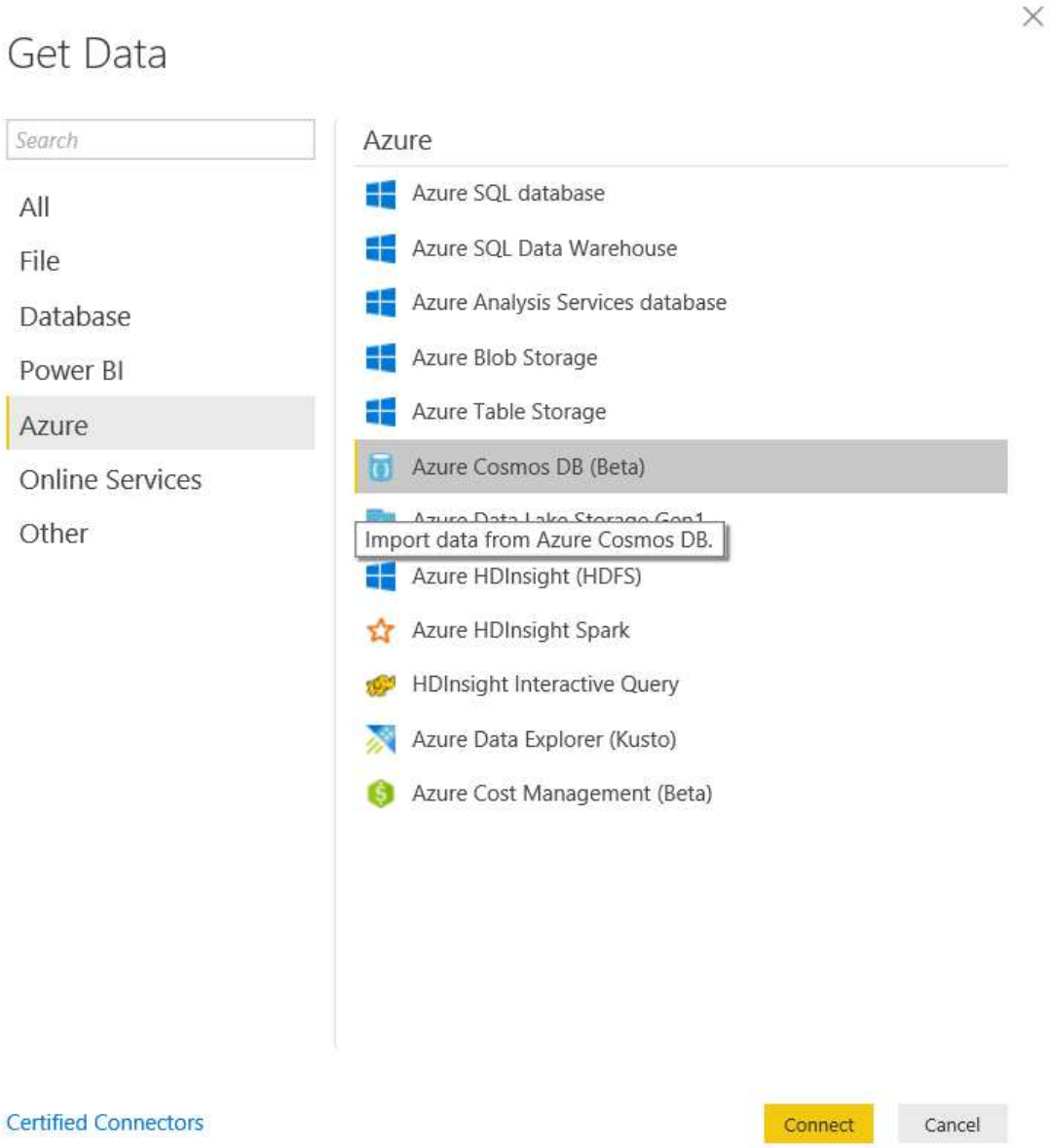
65. Inside Cosmos DB container you should have items populated. Confirm data is flowing to Cosmos DB

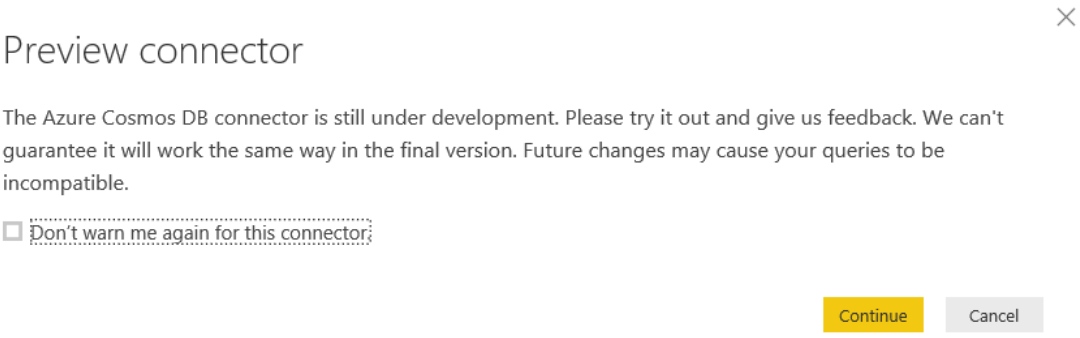


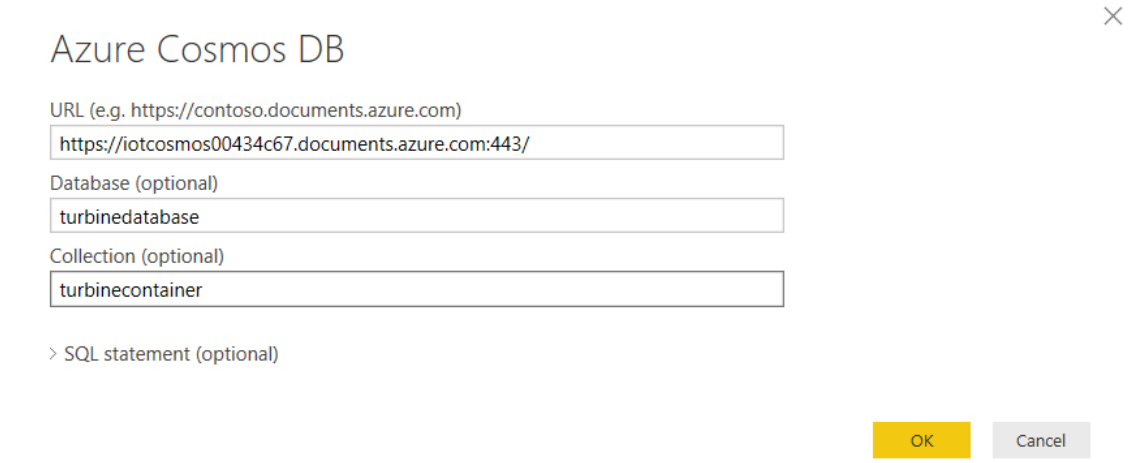


66. Install Power BI and use the Cosmos DB account and database as the source of data.

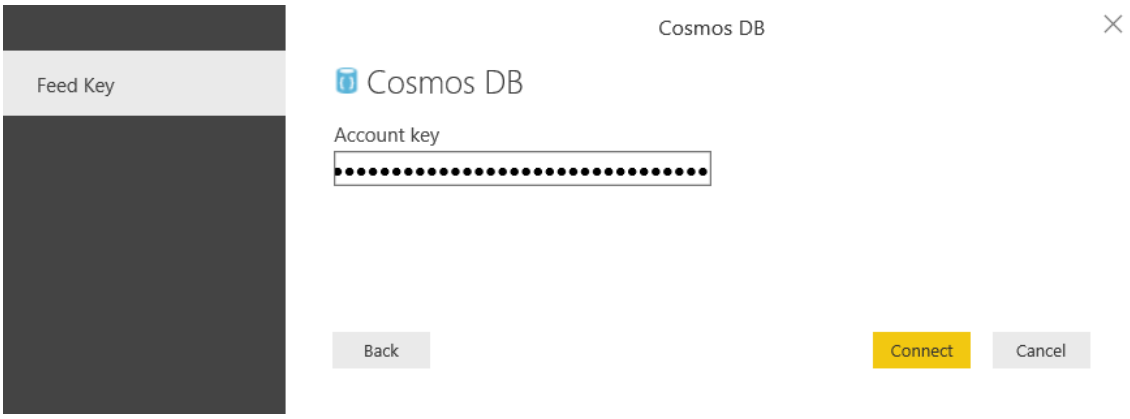
67. Choose Azure -> Azure Cosmos DB


Get Data
Search
All
File
Database
Power BI
Azure
Online Services
Other
Azure
Azure SQL database
Azure SQL Data Warehouse
Azure Analysis Services database
Azure Blob Storage
Azure Table Storage
Azure Cosmos DB (Beta)
Import data from Azure Cosmos DB.
Azure HDInsight (HDFS)
Azure HDInsight Spark
HDInsight Interactive Query
Azure Data Explorer (Kusto)
Azure Cost Management (Beta)
Certified Connectors
Connect
Cancel


Preview connector
The Azure Cosmos DB connector is still under development. Please try it out and give us feedback. We can't guarantee it will work the same way in the final version. Future changes may cause your queries to be incompatible.
Don't warn me again for this connector
Continue
Cancel



68. Feed key get from Key section in CosmosDB, choose Read-Write Keys tab, PRIMARY KEY



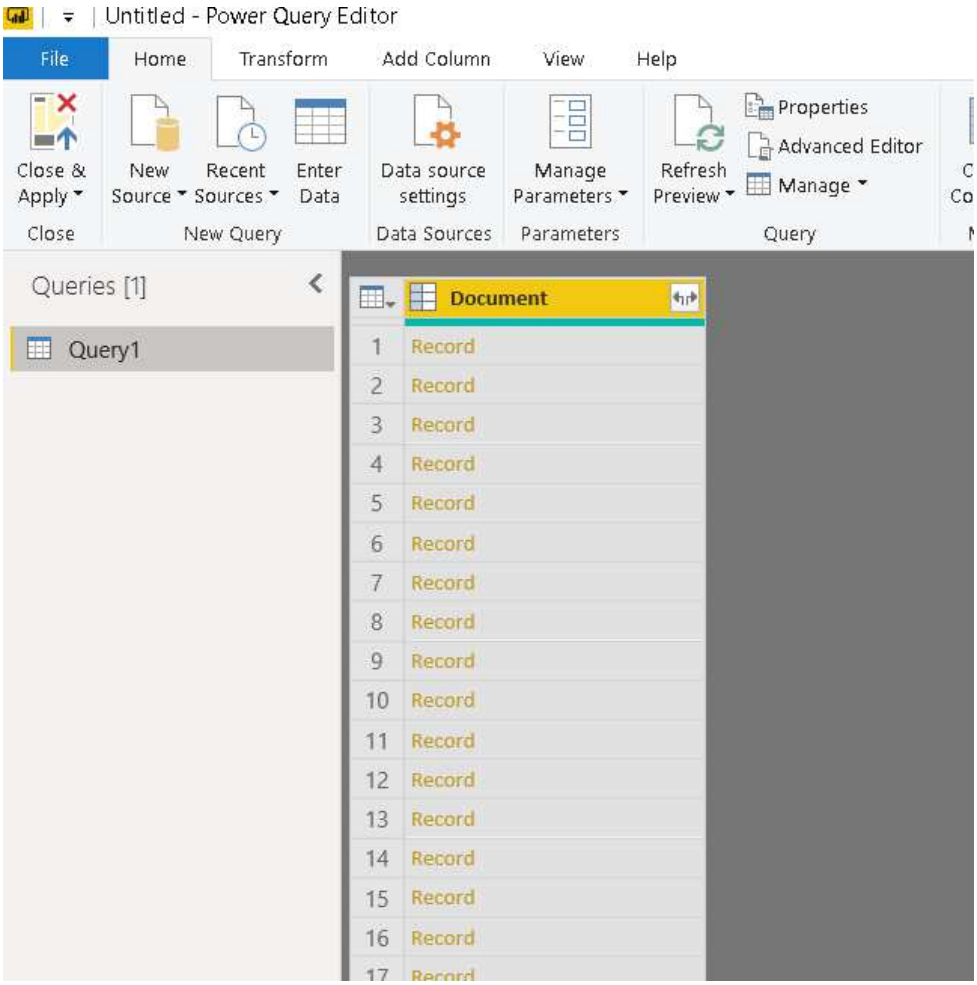
69. Select Edit to open the data.

https://iot-cosmos-00434c67.documents.azure.com:443/

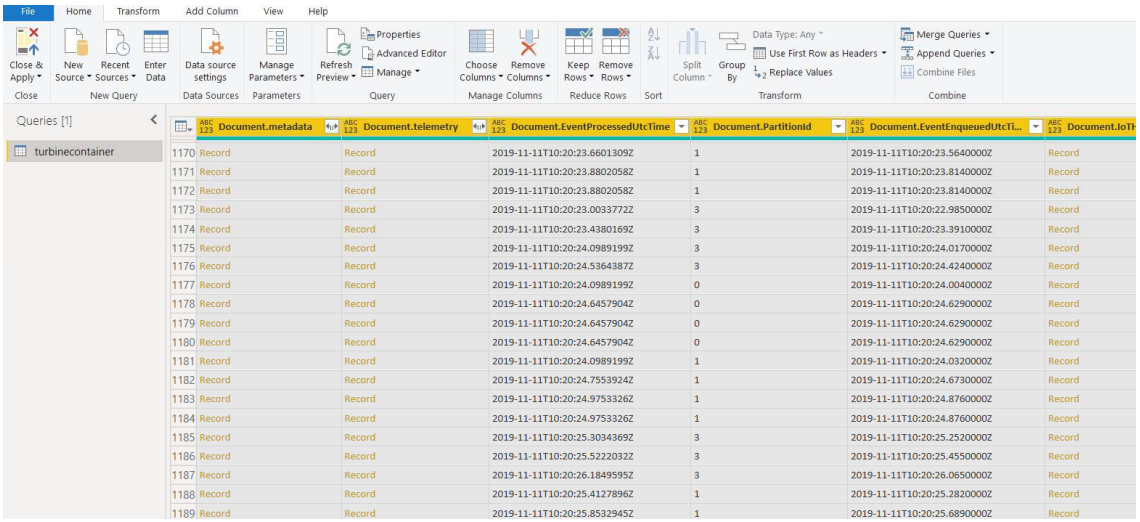
Document
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record
Record

The data in the preview has been truncated due to size limits.

Load Edit Cancel



70. The **Power Query Editor** will show a single column of data. In the header of the column, to the right of the **Document** title is a button that expands the record - click it.



71. A selection popup appears that lists the columns that can be expanded. These are the top level properties in the JSON. Unselect all columns except:

telemetry

EventEnqueuedUtcTime

72. This time, we need to expand the Document.telemetry column - click the expand button.

In the popup, unselect all columns except:

windSpeed

lowSpeedShaftRpm

power

name

Click OK. You will see these columns now displayed, again containing data.

Document.EventEnqueuedUtcTime	Document.telemetry.windSpeed	Document.telemetry.lowSpeedShaftRpm	Document.telemetry.power	Document.telemetry.name
2019-11-11T10:11:24.8390000Z	13.22	29.07	581.2	CWF-008
2019-11-11T10:11:25.2450000Z	13.84	29.19	583.6	CWF-008
2019-11-11T10:11:25.8550000Z	13.82	29.12	583	CWF-008
2019-11-11T10:11:26.6840000Z	13.24	29.19	584.2	CWF-008
2019-11-11T10:11:27.9190000Z	13.81	29.1	582	CWF-008
2019-11-11T10:11:28.9530000Z	13.47	29.2	583.6	CWF-008
2019-11-11T10:11:29.7820000Z	11.97	29.09	582.2	CWF-008
2019-11-11T10:11:30.8130000Z	11.88	28.93	578.2	CWF-008
2019-11-11T10:11:31.8290000Z	12.14	28.74	574.2	CWF-008
2019-11-11T10:11:33.0640000Z	12.04	28.44	568.2	CWF-008
2019-11-11T10:11:33.8920000Z	12.6	28.02	560.8	CWF-008
2019-11-11T10:11:34.9230000Z	12.31	29.2	583.6	CWF-008
2019-11-11T10:11:36.1730000Z	11.91	29.09	582.2	CWF-008

73. Proceed to follow and create the PowerBI Dashboard file and submit.

74. This is the final resources created in IOTCapstone Group