

## 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**.

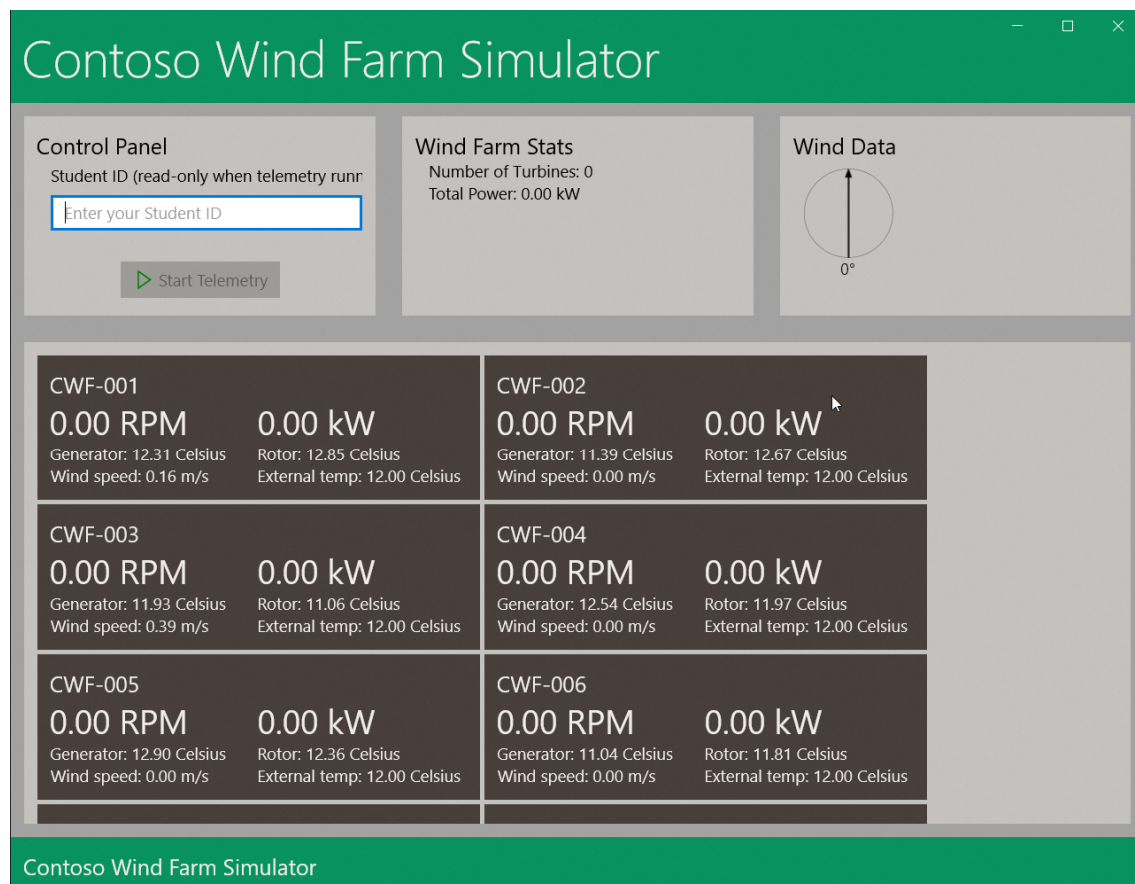
4. To install the **Wind Farm Dashboard** UWP application, download the zip file below and extract the contents to a directory.

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

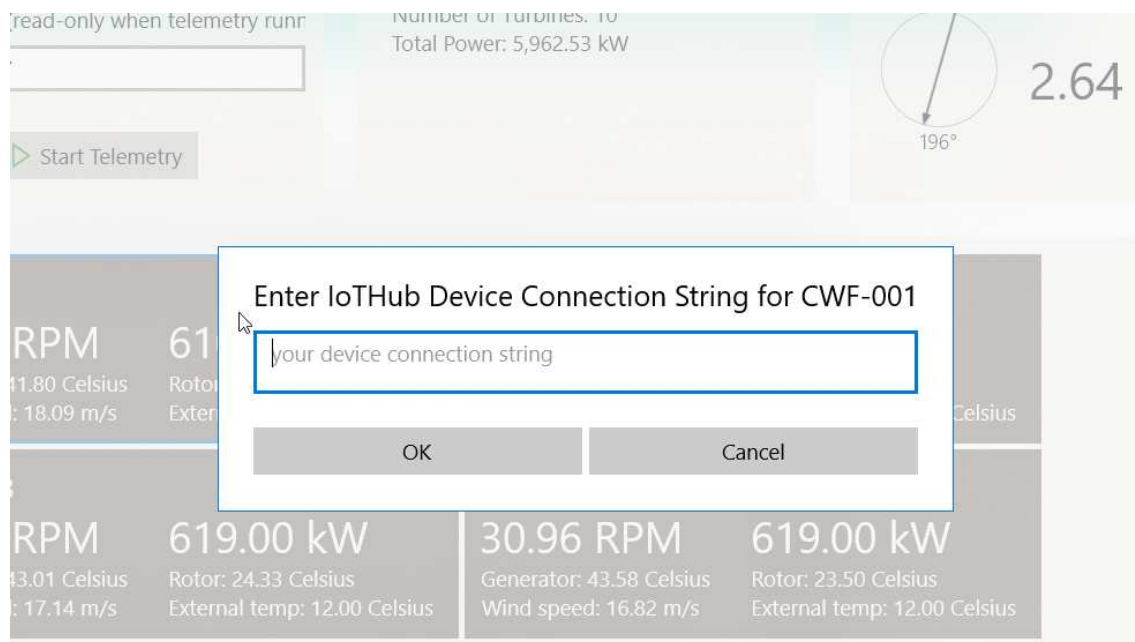
DEVICE ID	STATUS	LAST ACTIVITY TIME (UTC)	LAST STATUS UPDATE (UTC)	AUTHENTICATION TYPE	CLOUD TO D.
CWF-005	Enabled	--	--	Sas	0
CWF-MX-001	Enabled	Jul 4, 2019 5:43 AM	--	Sas	0
CWF-010	Enabled	--	--	Sas	0
<input type="checkbox"/> CWF-004	Enabled	--	--	Sas	0
CWF-003	Enabled	--	--	Sas	0
CWF-008	Enabled	--	--	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

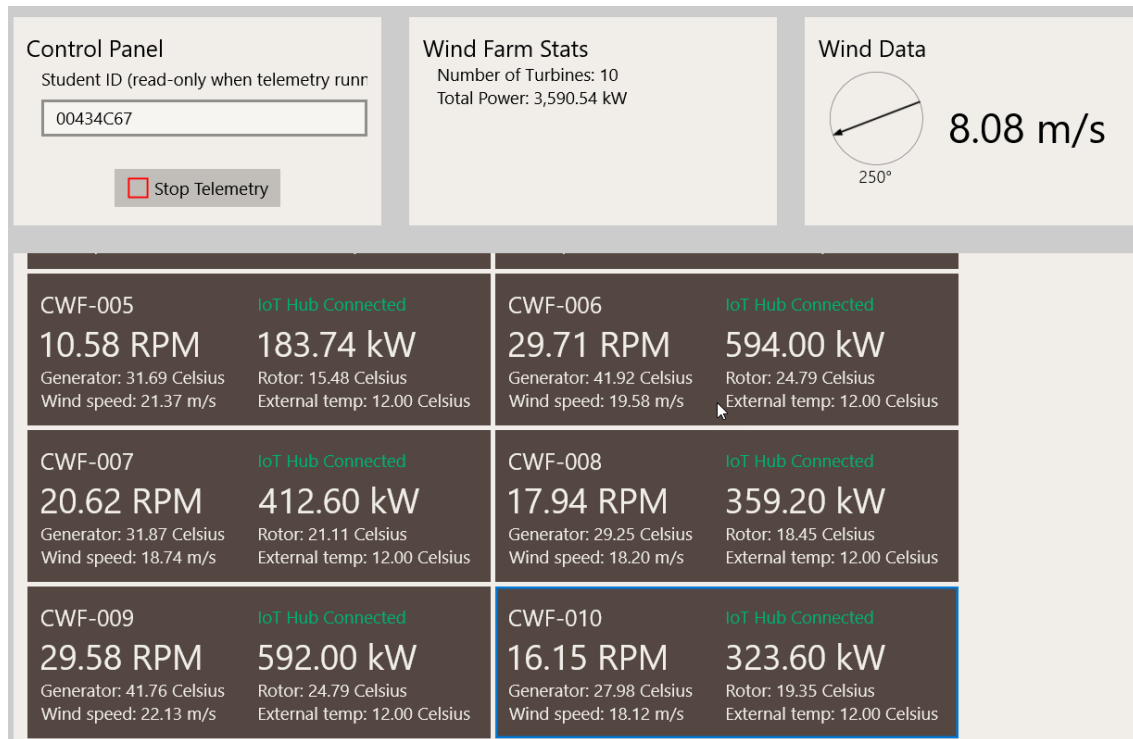
11. Start the **Wind Farm Dashboard App**

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

## Create Time Series Insights environment

Microsoft

### ENVIRONMENT DETAILS

Choose the subscription that will house your new environment. Use resource groups to organize and manage resources in that subscription. Note that these details can't be edited after they're saved.

\* Environment name ⓘ  ✓

\* Subscription ⓘ  ▼

\* Resource group ⓘ  ▼  
[Create new](#)

\* Location ⓘ  ▼

### PRICING

Choose a pricing tier. If you aren't sure which tier to choose, [visit our pricing page](#) to learn more.

\* Tier ⓘ S1 S2 PAYG (Preview)

\* Capacity ⓘ  1

Ingress rate: 1 M events per day  
Storage capacity: 30 M events  
Estimated cost: **MYR 630.00 / month**

[Review + create](#)

[Next: Event Source »](#)

[Download a template for automation](#)

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**.
20. Under **Source type**, select **IoT Hub**.
20. Under **Select a hub**, select **Select existing**.
20. Under **Subscription**, make certain to select the subscription you used to create your IoT Hub in the last lesson.
20. Under **IoT hub name**, select the IoT hub that you created for this module (e.g. **IoTBIHubxx**).
20. Under **IoT hub policy name**, select **iothubowner**.
20. Under **IoT Hub consumer group**, select **\$Default**.
20. Under **Timestamp property name**, leave the textbox blank.

## Create Time Series Insights environment

Microsoft

### EVENT SOURCE DETAILS

\* Create an event source? ⓘ

Yes

No

\* Name ⓘ

IoHubSource ✓

\* Source type ⓘ

IoT Hub ▼

\* Select a hub ⓘ

Select existing ▼

\* Subscription ⓘ

Pay-As-You-Go ▼

\* IoT Hub name ⓘ

iothub00434C67 ▼

\* IoT Hub access policy name ⓘ

iothubowner ▼

### CONSUMER GROUP



This consumer group should be used exclusively for this event source as there can be only one active reader from a given consumer group at a time.

\* IoT Hub consumer group ⓘ

\$Default ▼

New

### TIMESTAMP

Review + create

« Previous: Basics

[Download a template for automation](#)

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 x

WHERE

Add a search predicate...

MEASURE COUNT AVG SUM

telemetry.power v

SPLIT BY

telemetry.name v

WindSpeed Only Hide x

WHERE

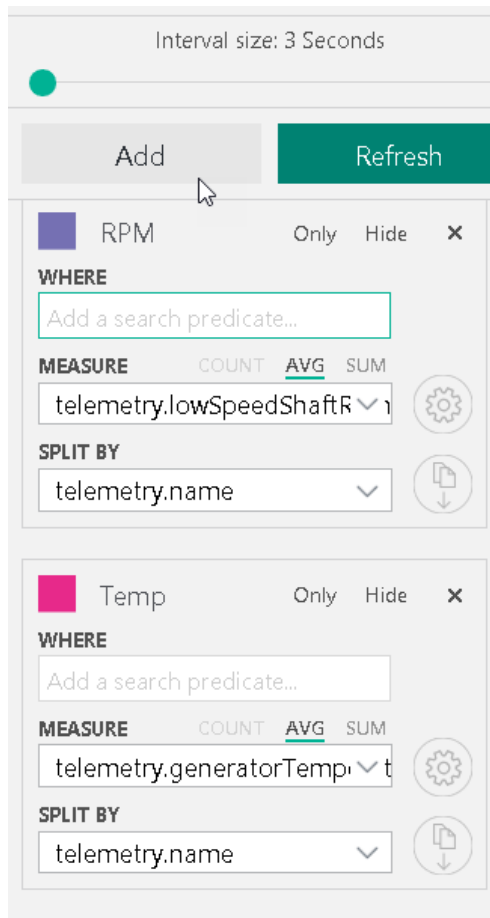
Add a search predicate...

MEASURE COUNT AVG SUM

telemetry.windSpeed v

SPLIT BY

telemetry.name v



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 screenshot 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 it **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'.

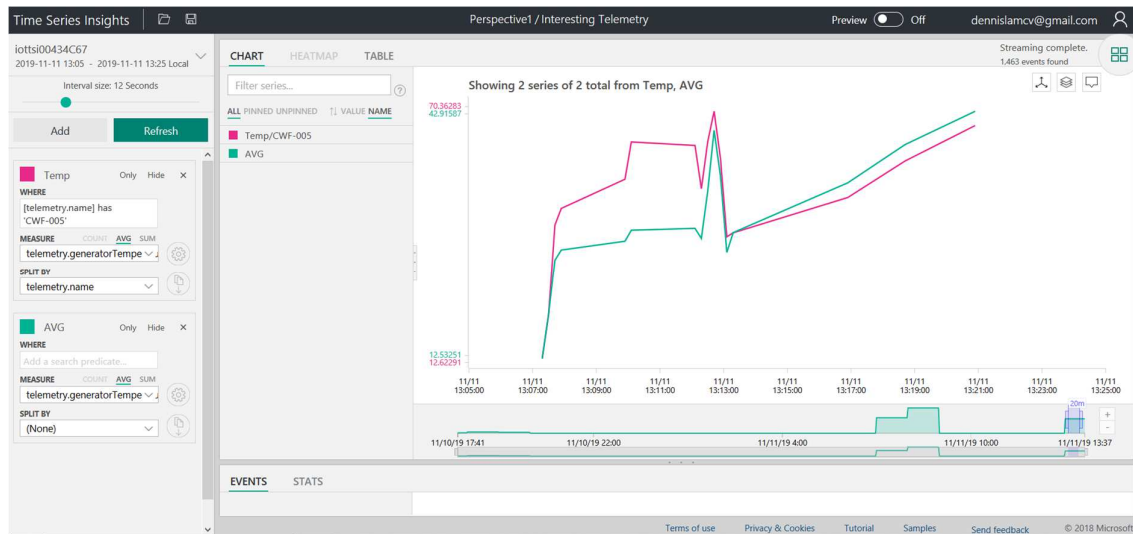
The same AVG measure you added above without a WHERE value or SPLIT BY - this will display the average for that measure across all turbines. Name this measure AVG.

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.

## Create Azure Cosmos DB Account

✓ Validation Success

### PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription	Pay-As-You-Go
* Resource Group	IoTCapstoneRG

[Create new](#)

### INSTANCE DETAILS

* Account Name	iot-cosmos-00434c67
* API ⓘ	Core (SQL)
Apache Spark ⓘ	<div>Enable Disable</div> <a href="#">Sign up for Apache Spark Preview</a>
* Location	(Asia Pacific) Southeast Asia
Geo-Redundancy ⓘ	<div>Enable Disable</div>
Multi-region Writes ⓘ	<div>Enable Disable</div>

[Review + create](#) [Previous](#) [Next: Network](#)

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.

## Add Container



Start at \$24/mo per database, multiple containers included  
[More details](#)

### \* Database id ⓘ

☒ Create new ☐ Use existing

turbinedatabase

☐ Provision database throughput ⓘ

### \* Container id ⓘ

turbinecontainer

### \* Partition key ⓘ

/PartitionId

☐ My partition key is larger than 100 bytes

### \* Throughput (400 - 1,000,000 RU/s) ⓘ



400





Estimated spend (USD): **\$0.032 hourly / \$0.77 daily** (1 region, 400RU/s, \$0.00008/RU)

OK

After setup, it looks like this:

SQL API  

- ▼  turbinedatabase
  - ▼  turbinecontainer
    - Items
    - Scale & Settings
    - ▶ Stored Procedures
    - ▶ User Defined Functions
    - ▶ Triggers

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

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

Scale & Setti... x

Changing the TTL or Indexing Policy impacts query results while the index transformation occurs. When a change queries return eventual results until the operation completes. For more information see, [TTL and index interaction](#)

▼ Scale

Throughput (400 - unlimited RU/s)

5000

Estimated spend (USD): **\$0.40 hourly / \$9.60 daily** (1 region, 5000RU/s, \$0.00008/RU)

Storage capacity

**Unlimited**

▼ Settings

Time to Live

Off On (no default) **On**

7776000 second(s)

Partition key

/PartitionId

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

Search (Ctrl+/)

Save Discard

Scale can't be edited while a job is running. You can stop the job to edit the scale.

Streaming units (1 to 192)

6

Locks

Job topology

Inputs

Functions

Query

Outputs

Configure

Storage account settings

Scale

Locale

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

iot-asa-00434C67 - Compatibility level

Stream Analytics job

Search (Ctrl+/)

Compatibility level

1.2 (preview)

Functions

Query

Outputs

Configure

Storage account settings

Scale

Locale

Event ordering

Error policy

Compatibility level

Managed Identity

General

37a: Add CosmosDB output to Stream Analytics

Cosmos DB

New output

Output alias

cosmos

☐ Provide Cosmos DB settings manually

☒ Select Cosmos DB from your subscriptions

Subscription

Pay-As-You-Go

Account id \* ⓘ

iotcosmos00434c67

Account key

.....

Database \*

☐ Create new

☒ Use existing

turbinedatabase

Container name \* ⓘ

turbinecontainer

Document id ⓘ

Save

Check and make sure Input and Output Connectivity is OK

Inputs

1

IoT Hub

Outputs

1

CosmosDB

Query

```

1 SELECT
2 *
3 INTO
4 [CosmosDB]
5 FROM
6 [IoT Hub];
7
8

```

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

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

Dashboard > iot-asa-00434C67 > Inputs

Inputs

+ Add stream input
+ Add reference input

*Inputs can't be added or edited while a job is running. You can stop the job to add or edit inputs.*

NAME	SOURCE TYPE	SOURCE	
IoT Hub	Stream	IoT Hub	<a href="#">Sample data</a> ...

39. Click on the link after sampling finished.

dennislamcv@gmail.c...

[Sample input 'IoT Hub' succeeded for Stream...](#) 7:16 PM

Sampled events are available to be downloaded.  
Click to download.

40. Download the file.



[Dashboard](#) > Sample data

## Sample data

IoTHub

[Download](#)

**DESCRIPTION**

Successfully sampled input 'IoTHub' of Stream Analytics job 'iot-asa-00434C67'. Click the download button on your command bar.

**STATUS**

Success

**TIME**

Friday, July 5, 2019, 7:16:42 PM

**RESOURCE**

[iot-asa-00434C67 \(stream analytics job\)](#)

**DIAGNOSTICS**

The sampled input contains one or more errors. Please go to the "Job topology" menu under "Inputs" and select the input named "IoTHub" to see full diagnostic messages.

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.

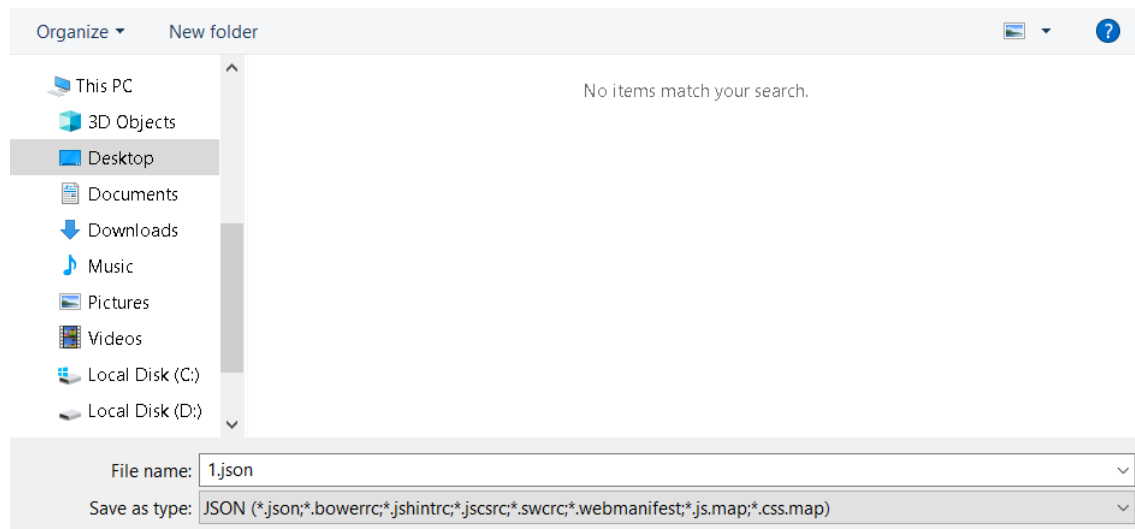
41b. Open Visual Studio Code, and then open the sample data file that you just saved.

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

```
{ } 1.json
c: > Users > Dennis > Desktop > { } 1.json > ...

1  [
2  {
3      "metadata": {
4          "deviceType": "SimulatedTurbine",
5          "studentId": "00434C67",
6          "uid": "2BB0244731E4154C"
7      },
8      "telemetry": {
9          "windSpeed": 5.4118581513417343,
10         "lowSpeedShaftRpm": 30.269999999999893,
11         "highSpeedShaftRpm": 3179.3999999999887,
12         "externalTemperatureCelsius": 12.0,
13         "generatorTemperatureCelsius": 42.65999999999989,
14         "rotorTemperatureCelsius": 24.101999999999958,
15         "power": 605.59999999999786,
16         "isTurbineBrakeOn": 0,
17         "name": "CWF-001"
18     },
19     "EventProcessedUtcTime": "2019-07-05T11:14:49.6754632Z",
20     "PartitionId": 3,
21     "EventEnqueuedUtcTime": "2019-07-05T11:00:00.9580000Z",
22     "IoTHub": {
23         "MessageId": null,
24         "CorrelationId": null,
25         "ConnectionDeviceId": "CWF-001",
26         "ConnectionDeviceGenerationId": "636978253336596822",
27         "EnqueuedTime": "2019-07-05T11:00:00.7700000Z",
28         "StreamId": null
29     }
30 },
31 {
```

---

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

Each IoT hub comes with built-in system endpoints to handle system and device messages.

^

Events

Events is the the default endpoint, and is used until custom routing rules are created.

Partitions ⓘ

4

Event Hub-compatible name ⓘ

iothub00434c67

Event Hub-compatible endpoint ⓘ

Endpoint=sb://iothub-ns-iothub0043-2405454-3ffa11a87d.servicebus.windows.net;/SharedAccessKeyName=iothubowner;SharedAccessKey=l...

Retain for ⓘ

1 Days

Consumer Groups ⓘ

Consumer Groups

\$Default

dev302rgconsumer

Create new consumer group

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

Input details

IoT Hub

Test

Delete

Inputs can't be added or edited while a job is running. You can stop the job to add or edit inputs.

IoT Hub \* ⓘ

iothub00434C67

Endpoint ⓘ

Messaging

Shared access policy name \* ⓘ

iothubowner

Shared access policy key ⓘ

.....

Consumer group ⓘ

dev302rgconsumer

Event serialization format \* ⓘ

JSON

You can implement a deserializer in C# that can read

Save

If the chosen resource and the stream analytics job are located in different regions, you will be billed to move data between regions.

-----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.

#### BASICS

Subscription	Pay-As-You-Go
Resource group	IoTCapstoneRG
Location	(Asia Pacific) Southeast Asia
Storage account name	iotblob00434c67
Deployment model	Resource manager
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)
Performance	Standard
Access tier (default)	Hot

#### ADVANCED

Secure transfer required	Enabled
Allow access from	All networks
Hierarchical namespace	Disabled
Blob soft delete	Disabled

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**.



## Add a storage endpoint

Route your telemetry and device messages to Azure Storage as blobs.

\* Endpoint name ⓘ

blobiot

### Azure Storage account and container

Create a new container, or choose an existing one that shares a subscription with this IoT hub.

Azure Storage container

Pick a container

Batch frequency ⓘ



Chunk size window ⓘ



Encoding ⓘ

AVRO

JSON

\* Blob file name format ⓘ

{iothub}/{partition}/{YYYY}/{MM}/{DD}/{HH}/{mm}

Create

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**.

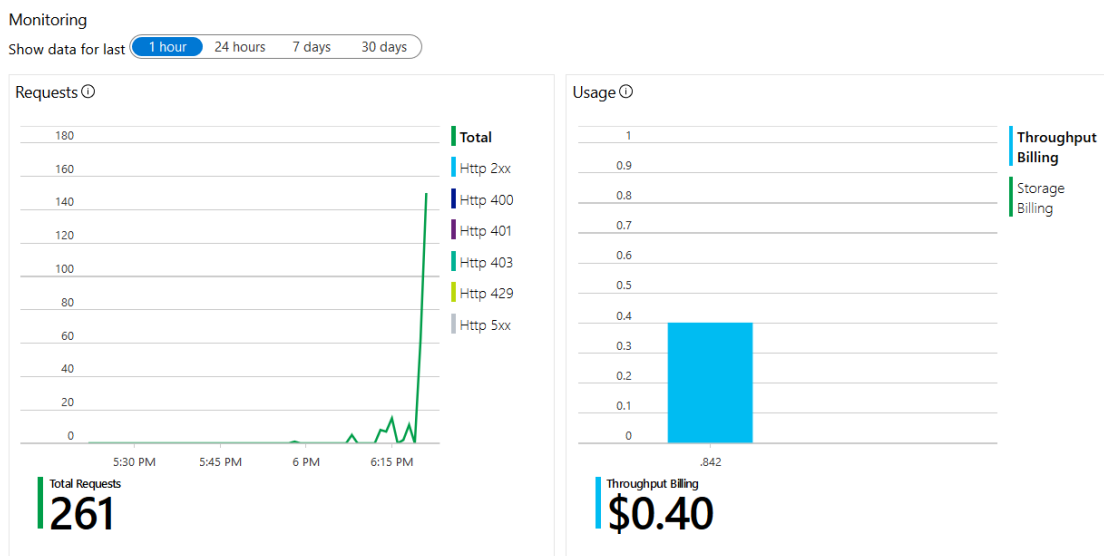
57. In the **Containers** list, click **allmessages**. In the **Containers** blade, at the bottom of the page, click **Select**. The **Containers** blade will close and the **Add a storage** blade is shown. The blade has updated to display the URI for the **Azure storage container** above **Pick a container**.

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**.

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**).

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





SQL API

turbinedatabase

turbinecontainer

Items

Scale & Settings

Stored Procedures

User Defined Functions

Triggers

Items

SELECT \* FROM c

Edit Filter

id	/PartitionId
72e7b1ae-acb6-6b1...	3
7394a375-de07-9f63...	3
73f9c091-16fd-8650...	3
458ee03c-0eca-3d0...	3
2e9a063c-81ac-5216...	3
e832fcfc-c2d2-bf18-...	3
fef7dc0d-3796-f784-...	3
f95cda88-4d11-1afe...	3
17a992b7-1479-c17...	3
0351346f-7c13-d461...	3
0ecb1259-b240-885...	3
ae12541e-27c5-0cc6...	3
dd4333a9-30c7-4af8...	3

Load more

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

```

{
  "metadata": {
    "deviceType": "SimulatedTurbine",
    "studentId": "00434C67",
    "uid": "95FF681F2533ACB4"
  },
  "telemetry": {
    "windSpeed": 12.799999999999999,
    "lowSpeedShaftRpm": 26.619999999999994,
    "highSpeedShaftRpm": 2796.1499999999996,
    "externalTemperatureCelsius": 12,
    "generatorTemperatureCelsius": 38.42,
    "rotorTemperatureCelsius": 23.014,
    "power": 532.8,
    "isTurbineBrakeOn": 0,
    "name": "CWF-001"
  }
},
"EventProcessedUtcTime": "2019-11-11T10:12:35.0782137Z",
"PartitionId": "3",
"EventEnqueuedUtcTime": "2019-11-11T10:11:24.6200000Z",
"IoTHub": {
  "MessageId": null,
  "CorrelationId": null,
  "ConnectionDeviceId": "CWF-001",
  "ConnectionDeviceGenerationId": "637090264636038947",
  "EnqueuedTime": "2019-11-11T10:11:24.0000000",

```

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



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)
- Azure Data Lake Storage Gen1
- 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

## Azure Cosmos DB

URL (e.g. <https://contoso.documents.azure.com>)

Database (optional)

Collection (optional)

> SQL statement (optional)

OK

Cancel

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

Feed Key

Cosmos DB

Account key

.....

Back

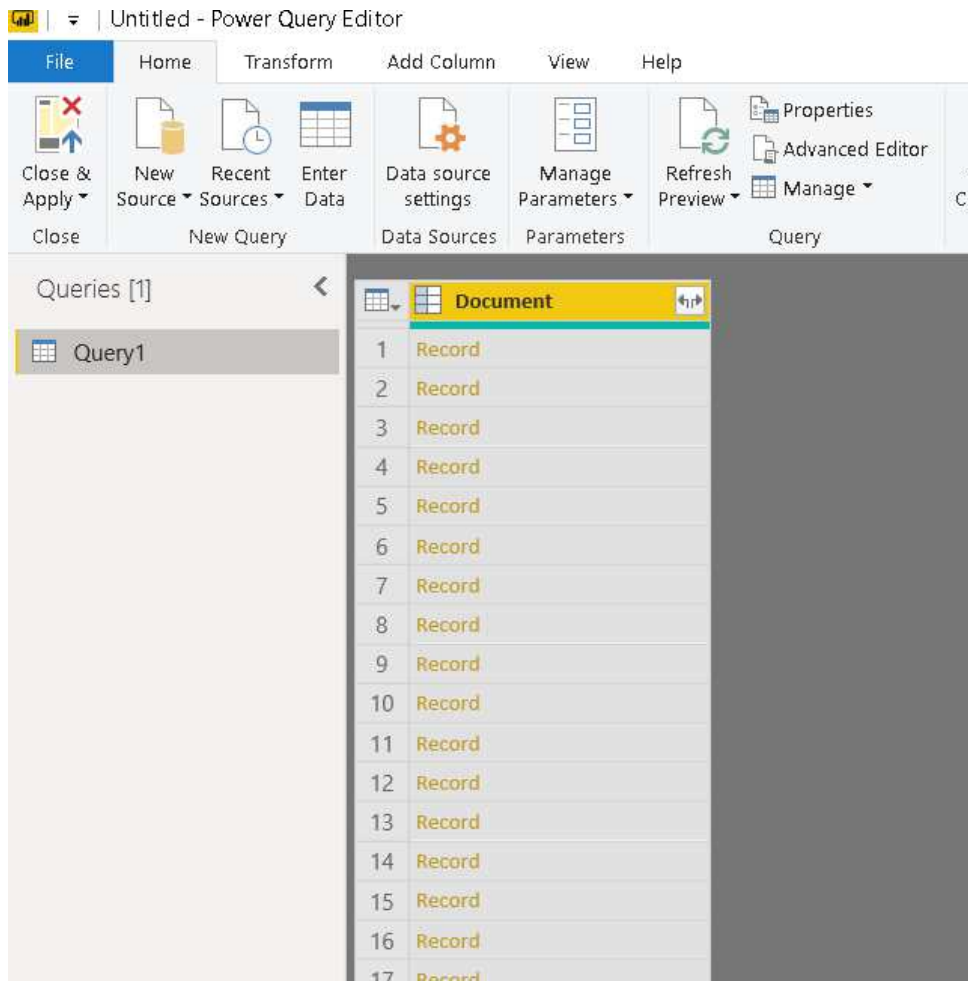
Connect

Cancel

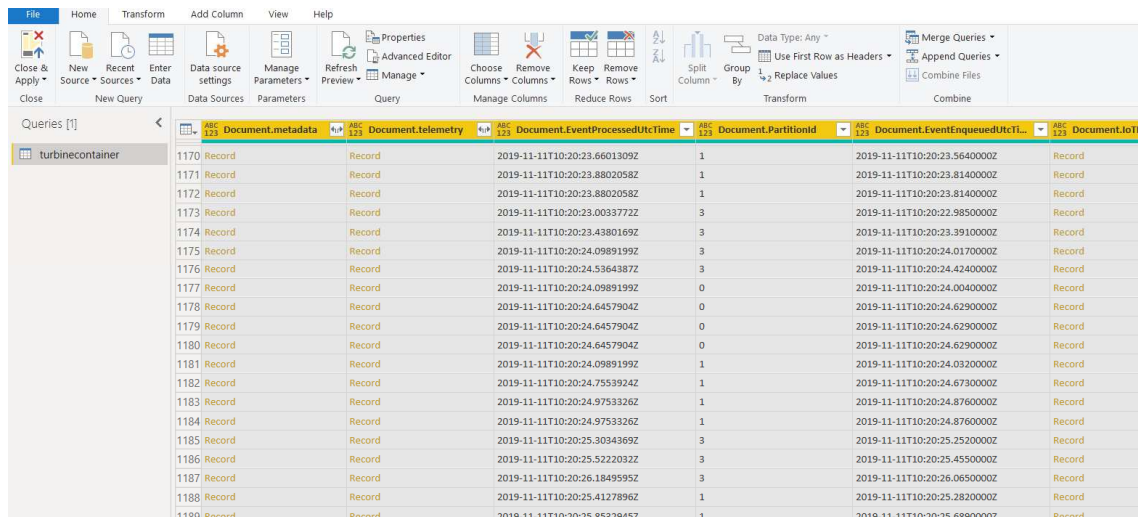
69. Select Edit to open the data.

□ ×

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

Resources

IoTCapstoneRG

Refresh

 <a href="#">iotasa00434C67</a>	Stream Analytics job	Southeast Asia
 <a href="#">iotcosmos00434c67</a>	Azure Cosmos DB account	Southeast Asia
 <a href="#">iothub00434C67</a>	IoT Hub	Southeast Asia
 <a href="#">iottsi00434C67</a>	Time Series Insights environment	Southeast Asia
 <a href="#">iotdla00434c67</a>	Data Lake Analytics	East US 2
 <a href="#">iotdls00434c67</a>	Data Lake Storage Gen1	East US 2
 <a href="#">iotstore00434c67</a>	Storage account	Southeast Asia
 <a href="#">iotfunc00434C67</a>	Application Insights	Southeast Asia
 <a href="#">iotfunc00434C67</a>	App Service	Southeast Asia
 <a href="#">IoTHubSource (iottsi00434C67/IotHubSource)</a>	Time Series Insights event source	Southeast Asia
 <a href="#">SoutheastAsiaPlan</a>	App Service plan	Southeast Asia