

Objective: In this tutorial, you will learn how to use Stream Analytics and Power BI to stream sensor data in real-time.

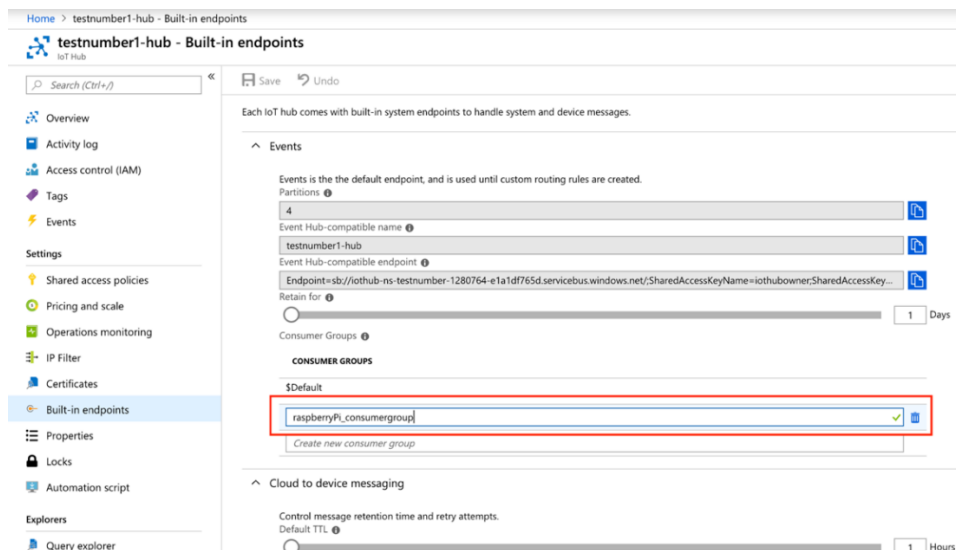
Required Setup: Azure account, GrovePi libraries, IoT hub setup(primary key), IoT device setup(primary key)

Parts:

- RPi 3 B+, GrovePi+ board
- Grove connection wires, and temperature&humidity sensor

Part A: Azure Stream Analytics

Step 1. On your home page of your Azure portal, click on, in my case, 'testnumber1-hub' (this is your IoT hub's name) from the 'recent resources' section. From the navigation menu of the IoT hub, click on 'Built-in endpoints' under settings. Under 'Events', find 'consumer group'. We need to add a new consumer group called 'raspberrypi_consumergroup'. After type into the blank, click on the blank space besides it.



Step 2. Click on '+Create a resource' in the main navigation menu. In the Search box, type 'Stream Analytics'. Choose 'Stream Analytics job' in the drop down menu. After click 'Create' at the bottom, you should see something similar to the screenshot on the right. Enter 'raspberrypi' for the Job name and select 'test-group' for the resource group (we created a resource group in week4 lab step 6). Click on 'Create' on the bottom.

New Stream Analytics job
 ✕

* Job name

* Subscription

* Resource group

Create new

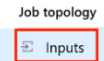
* Location

Hosting environment

Streaming units (1 to 120)

Step 3. You will need to define the **Inputs**, **Query**, and **Outputs** for any Stream Analytics job. Inputs are the sources of your data, which will be the IoT hub we created last week. Query is used to specify a way filtering the incoming data. Outputs are the destinations where you want the data to go, which will be the Power BI in our case here.

Step 4. On the left navigation menu of Stream Analytics, click 'Inputs' under 'Job topology' section.



Step 5. Click '+Add stream Input' -> 'IoT hub' and a secondary window should appear from the right side. You only need to change two fields. For the 'Input alias', type 'iothubinput; for the 'Consumer Group', select the 'raspberrypi_consumergroup' which we created it in step 2. You can pick your name to fill in these two fields but avoid using special characters because we are going to use these names in the SQL query later on and the query won't recognize special characters. Other fields should be filled automatically. The IoT Hub should be selected to the IoT Hub created last week and Endpoint should be 'Messaging'.

There is another setting for Endpoint called 'Operations Monitoring' which is used to monitor events. You can find more information about that option from [here](#). We will stick with 'Messaging' for now. Click on 'Save' at the bottom. You can click on the notification icon in the menu at the top to see if the input is tested and connected successfully. The input method is completed.

Step 6. From the navigation menu, under 'Job topology', click 'Outputs'. Click '+Add' -> 'Power BI' and a window should appear from the right side.

Step 7. For the 'Output alias' field, type 'powerbioutput; for the 'Dataset name' field, type 'tempdataset; for the 'Table name' field, type 'temptable. Of course, you can pick any name you want. Just make sure that you remember them because we need them later on. Click 'Authorize' -> click 'Save'. The authorization may require you to login again. The output method is completed.

Power BI
 ×

New output

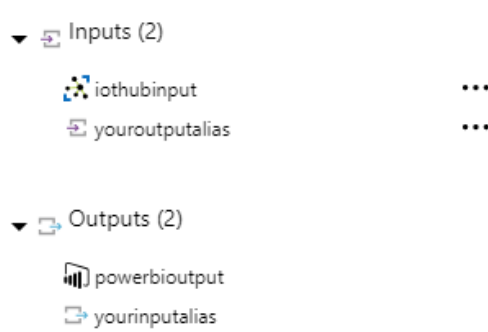
* Output alias

Group workspace

* Dataset name ⓘ

* Table name


Step 8. Now we only have the last piece of the puzzle----Query. From the navigation menu, under 'Job topology', click 'Query' to confirm that both input and output are showed up correctly.



Step 9. Now we will pause the Query set-up for now. Don't close the browser. Let's switch to the Raspberry pi. Download 'stream.py' from the folder and open it using the Thonny on the Raspberry Pi. Read through the comments to understand the code and try to finish the two tasks in the file.

```
## Azure IoT Hub connection
## The first connection string--primary key in week4
##### Task 1 -- Modify this part
URI = 'YOUR_IOT_HUB_NAME.azure-devices.net'
KEY = 'YOUR_IOT_HUB_PRIMARY_KEY'
IOT_DEVICE_ID = 'YOUR_IOT_DEVICE_ID'
POLICY = 'iothubowner'
```

```
## get the temperature and Humidity from the DHT sensor
## get the sound level from the sound sensor
##### Task 2 -- Complete this function
def read_temp():
    ## Use the previous knowledge to fetch data from DHT sensor
    ## hint: week2_partA.py
    return temp
```

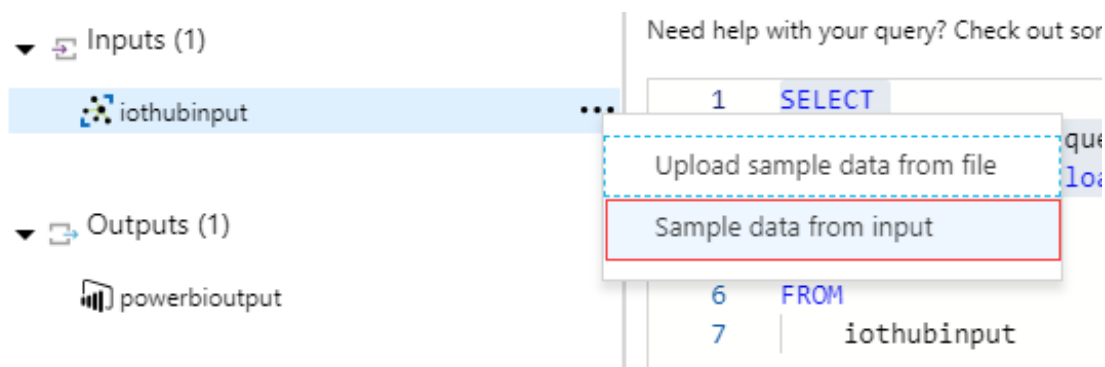
Step 10. After you complete the tasks, save  the python script on the desktop, named it 'stream.py'. We are not going to use Thonny to run it because one of the packages does not work very well in python3. Open the terminal, type 'cd Desktop' to get into the Desktop folder. Then run 'python stream.py'. (If you just use python in the terminal, it's going to use python2 as default.) If the data is '-1', reboot Raspberry Pi.

[illegible]

Step 11. Back to Azure Stream Analytics job. From the navigation menu, under 'Job topology', click 'Query'. Type the following SQL commands into your query window. If you have different name for input and output, change 'powerbioutput' and 'iothubinput' to the names you picked for input and output.

```
SELECT
    CAST(iothub.EnqueueedTime As datetime) AS event_date,
    CAST(temp AS float) AS temp
INTO
    powerbioutput
FROM
    iothubinput
```

Step 12. Click on the three little dots next to your IoT Hub input and choose 'Sample data from input'.

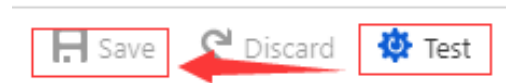


Step 13. Set 'Duration' to 1 minute in the prompted window. Click 'ok' at the bottom. It will take exactly 1 minute to fetch the data. Make sure the python script is running because it's uploading the temperature data to the Azure.

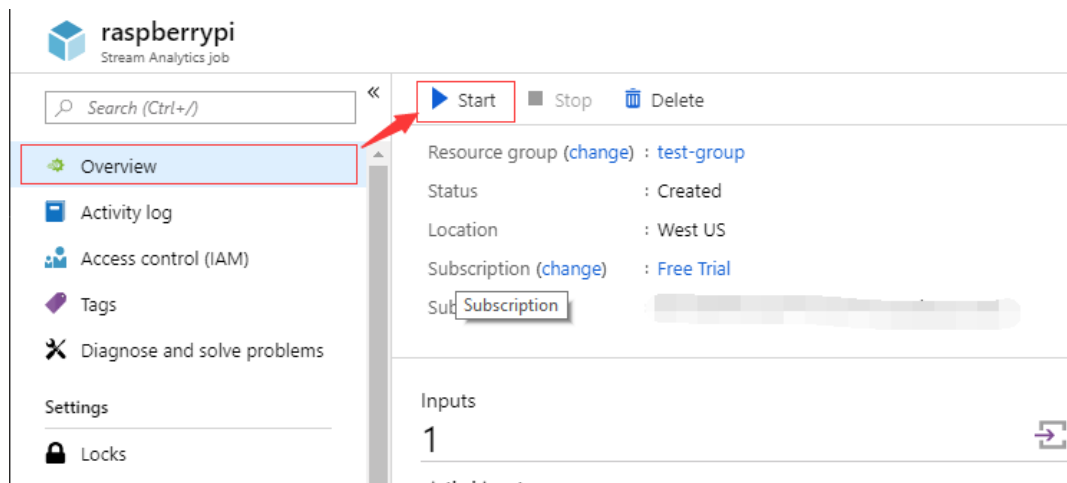
* Duration ⓘ

| Days | Hours | Minutes | Seconds |
|------|-------|---------|---------|
| 0 | 0 | 1 | 0 |

Step 14. Wait until the following notification shows up. Click 'Test' from the top left side of the window. You should see 11 rows of data in the 'Results' section. These temperature data should be the same set you got from the console output in the terminal on Raspberry Pi. Then click 'Save'.

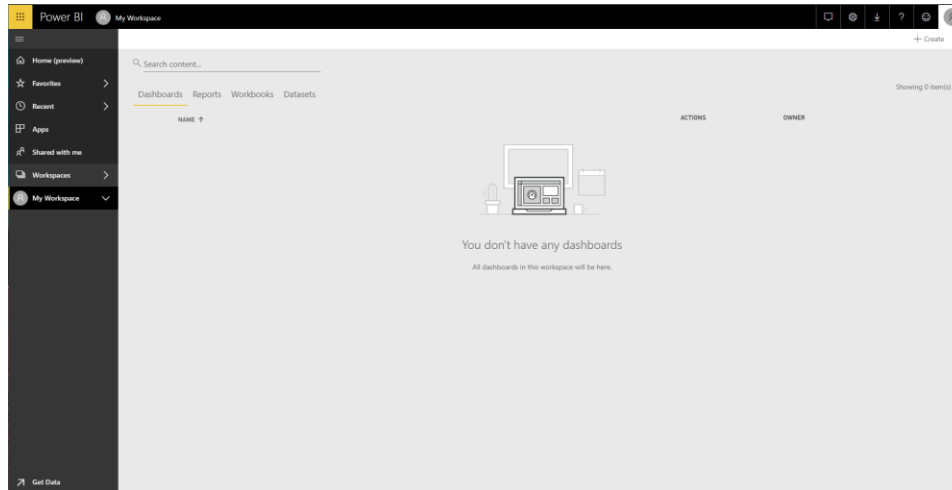


Step 15. Click 'Overview' from the navigation menu of 'Stream Analytics job'. Click 'Start' and click 'Start' again.



Part B: Visualization with Power BI

Step 16. Open a new tab in the browser and use Google to search 'Power BI'. Click on the first link and click 'sign in' at the top right corner. Use the same login credential of Azure to log in. You should see the following screen.





Step 17. Tab on 'My Workspace' on the left navigation menu. Click on '+Create' at the top right corner. Then choose 'Dashboard'. Put 'Raspberry Pi' as Dashboard name. (Of course, you can pick any name you want.)

Step 18. We want to add two tiles to the dashboard to visualize our temperature data. Click '+Add tile' in the top menu. Pick 'Custom Streaming Data' from 'REAL-TIME DATA' section. Click 'Next' at the bottom.



Step 19. Select 'tempdataset' from 'YOUR DATASETS'. (Pick corresponding dataset if you used different name.) Click 'Next' at the bottom.

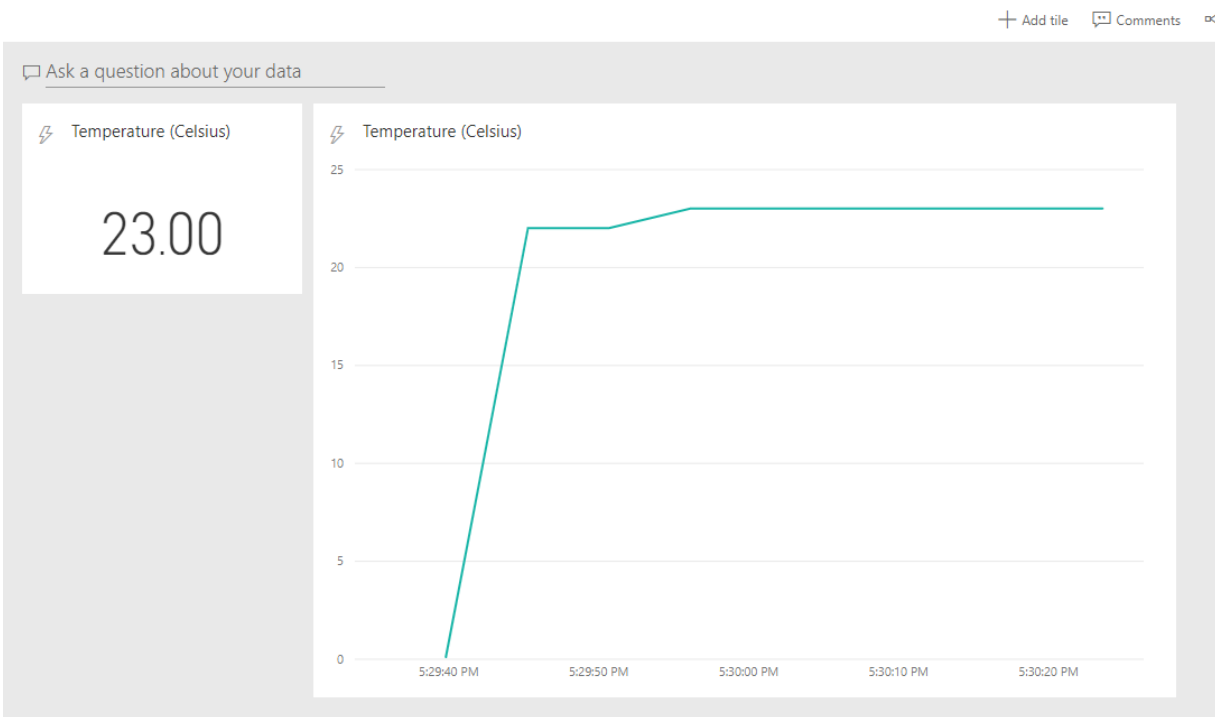
Step 20. Click '+Add value' in the 'Fields' section. Choose 'temp' in the drop-down menu. Choose the 'setting'   from the submenu. Put '2' in 'Value decimal places'. Click 'Next'.

Step 21. Put 'Temperature (Celsius)' in the 'Title' field. Click 'Apply'. We just create a 'Card' tile on the dashboard. Now let's try to create a 'Line Chart' tile. Redo **Step 18** and **Step 19**.

Step 22. We are going to choose 'Line chart' from 'Visualization type'. For 'Axis', click '+Add value' and choose 'event_date' from the drop down menu; for 'Values', choose 'temp'. Leave 'Time window to display' as '1 minutes'. Click 'Next'. Put 'Temperature (Celsius)' in the 'Title' field. Click 'Apply'.

Step 23. Your line chart should be updating very frequently. If your line chart does not show anything, check if the python script is still running properly on your raspberry Pi. Sometimes, the DHT sensor would freeze after a long-term usage. Unplug, then plug the sensor back, reboot Raspberry Pi and re-run the script until you see appropriate temperature output. Also, check if your 'Stream analytics job' is still running properly. Restart it if the job is stopped. As soon as the script and the 'Stream analytics job' are both

running, you should be seeing the following screen. Congratulation! Now you are streaming temperature data in real-time using Azure Stream Analytics and Power BI!



Step 24. Next, instead of visualizing the data directly through PowerBI, we are going to learn how to use Azure SQL database to collect/store the data for the further usage.

References:

1. https://courses.edx.org/courses/course-v1:Microsoft+DEV225x+3T2018/courseware/c67f4dcf-86dc-9926-ca39-d7d7300e8c0f/ae8dfac3-fc4d-0fa4-a1c9-4edd729b40da/1?activate_block_id=block-v1%3AMicrosoft%2BDEV225x%2B3T2018%2Btype%40vertical%2Bblock%40121d369e-4326-4b0e-03b7-a336e4234060
2. <https://www.taygan.co/blog/2018/03/12/streaming-sensor-data-in-real-time-with-azure-iot-hub>
3. <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi>