

Getting started with Azure, Send Some Data on Cloud

1. Create an account
2. Login to account
3. Then find the azure CLI (command line interface) Also known as Cloud Shell
4. Find Cloud shell button, click it
5. Create Storage (subscription will be automatically selected as Free)
6. It'll ask BASH or Powershell, select BASH here
7. Cloud Shell Started
8. Run this command on Azure IoT Shell ***az extension add --name azure-cli-iot-ext***
9. Now the extension is added
10. Close the shell for now
11. Create a new resource
12. Look for internet of things and click on azure iot hub
13. Now give the details about subscription, resource group and name of IoT hub usually a simple name like *mypihub*, click on **size and scale**
14. Select the FREE TIER
15. Click review+ create and then create
16. Install the required python packages on raspberry pi
17. `sudo apt-get update`
18. `sudo pip3 install azure-iot-device`
19. `sudo pip3 install azure-iot-hub`
20. `sudo pip3 install azure-iothub-service-client`
21. `sudo pip3 install azure-iothub-device-client`
22. go to iot hub
23. then go to exploreres → iot devices and create a new device
24. look for primary connection string
25. Now go to iot hub
26. Go to shared access policy
27. Click on iothubowner and copy the primary connection string
28. Use this connection string in `iothub_registry_manage_sample.py`

Getting started with sense HAT

<https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat>

Installation of Azure Libraries

changes : current azure docs support pip installations

`sudo apt-get update`

`sudo pip3 install azure-iot-device`

`sudo pip3 install azure-iot-hub`

`sudo pip3 install azure-iothub-service-client`

`sudo pip3 install azure-iothub-device-client`

my connection string

HostName=amit.azure-
devices.net;DeviceId=myspi;SharedAccessKey=BNfUR6dUXhgEzac6w8tFtClBRxHt3bqr2vQHjWY3q0o=

To view telemetry data on Azure IOT, use below commands

az extension add --name azure-cli-iot-ext

az iot hub monitor-events --hub-name **myspihub** --device-id **myspi**

official telemetry tutorial

<https://docs.microsoft.com/en-us/azure/iot-hub/quickstart-send-telemetry-python>

For registry manager sample, you need the IOTHub connection string

Once the hub is available, go to settings - Shared access policies, click iothubowner and copy the connection string - primary key. Use this key in the registry_manager_sample

Case Study for today

<https://projects.raspberrypi.org/en/projects/fetching-the-weather/11>

Telemetry Example

1. we'll create an iot hub
2. Go to home → all resources → click on your iot hub
3. IoT Devices → New IoT Device → only give a device ID; change nothing else → save
4. Go to device and then see various connection string data shown
5. copy the device primary connection string
6. HostName=mypihub.azure-devices.net;DeviceId=myspi;SharedAccessKey=gaD2du1+L3yUMcmJ6NUfvH68xNtoyZ7jL8xg7SUDu8o=
7. write down a python code (edit) with connection string (SimulatedDevice.py)
8. <https://github.com/amitrana3348/IoTCodes/blob/master/SimulatedDevice.py>
9. run the code --> start sending data to azure IoT hub
10. now we want to observe the data coming to iot hub
11. cloud shell
12. And run a sample command to view the data
13. To create CLI iot extension, use
14. az extension add --name azure-cli-iot-ext
15. to view the incoming data on iot hub, use this on Azure CLI
16. az iot hub monitor-events --hub-name mypihub --device-id myspi

File Uploading

1. Create a log of sensor values / whatever vital information that you have in a file
2. Abc.txt file created
3. Goto iot hub and then enable file upload as instructed
4. Goto iot hub
5. Goto messaging → file upload
6. Goto azure storage accounts → create a container → give it a unique name → select the container
7. Now save settings in file upload option
8. Now make necessary changes to your file upload code
9. File Upload Code is available at
10. https://github.com/amitrana3348/IoTCodes/blob/master/upload_file_azure.py
11. Change the primary connection string here in this code as well with Device primary connection string
12. Modify PATHTOFILE with the path to your file on windows / rpi
13. Change the file name you want to appear in azure IoT hub
14. Once the code is run
15. Goto resources → storage account → containers
16. Check the contents here

Creating storage Endpoint and a custom gateway

1. Go to iot hub
2. In messaging → message routing click here
3. You'll see 3 tabs, routes, custom endpoint and enrich message
4. First, go to custom endpoint
5. Add and choose storage endpoint, it'll open a new page
6. Here you'll have to give an end point name, give some sensible name which can be referred later
7. Pick a container, it'll open in another new page
8. It'll open your storage accounts
9. Select your storage account
10. Create a new container
11. Then select your container and click on "select"
12. You'll come back to add storage end point page
13. Select JSON format and click on create
14. Custom end point is created
15. Now to first tab, which is routes
16. You now click on + symbol to create a route
17. Give it a name
18. Then select an endpoint from drop down list
19. Keep the query as True
20. And save it.
21. Goto resources
22. Storage account
23. Containers and open the created container for end point

CLI Instructions for Storage Account

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-configure-file-upload-cli>

Graphical Instructions with python program

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-python-python-file-upload>

Stream Analytics

<https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-get-started-with-azure-stream-analytics-to-process-data-from-iot-devices>