**AWS IoT Raspberry Pi Tutorial**

This is a collection of resources that will help get you started with the AWS IoT, giving you some background on the assembled parts and walking you through the process of getting the parts assembled and configured to send data to AWS through LTE-M

**Let’s start with some background**

**What you need** -

Hardware:

1. Raspberry Pi Zero W /w Headers– The Raspberry Pi Zero is half the size of a Model A+, with twice the utility. A tiny Raspberry Pi that’s affordable enough for any project! **Make sure your Raspberry Pi Comes with 2x20 GPIO Headers or you will not be able to connect you application shield**
2. Micro SD Card
3. SixFab RPi Cellular IoT Application Shield – An addon for the Raspberry Pi that has the combined LTE technologies Cat.M1, Cat.NB1 (NB-IoT), and eGPRS for Raspberry Pi, based on Quectel’s BG96 module.

Software:

Raspbian Lite – Free lightweight version of the Raspbian no GUI( Terminal only)

AWS Account – Students can create a free account and receive free credits. Follow this link <https://aws.amazon.com/education/awseducate/> to join AWS Educate

Green Grass SDK – Python version as the SDK for the SixFab shield is also in Python

**Overview of what the AWS Process Looks like**



**Configuring your IoT Hardware:**

Flash Raspbian Lite on your SD Card

Follow Official documentation on getting SSH to work

<https://www.raspberrypi.org/documentation/remote-access/ssh/>

Follow the AWS Documentation on getting your device to work Download the ARM6l version of

Greengrass as the RPi **Zero** is ARM6

<https://docs.aws.amazon.com/greengrass/latest/developerguide/module1.html>

You can use the provided GG\_Sensor\_Data lambda function and upload that as a lambda function. The provided GG\_init.sh file can be used to simplify some process of restarting your Greengrass daemon if you need to restart your RPi.

Follow this guide on setting up IoT analytics to begin redirecting your data to other AWS Services

<https://docs.aws.amazon.com/iotanalytics/latest/userguide/quickstart.html>

The Sensor\_Data.py code can be thrown into a Jupyter Notebook (AWS Service) To pull data from IoT Analytics to visualize your data.

You’re connected to AWS and can now start to explore more AWS Services!

Tips:

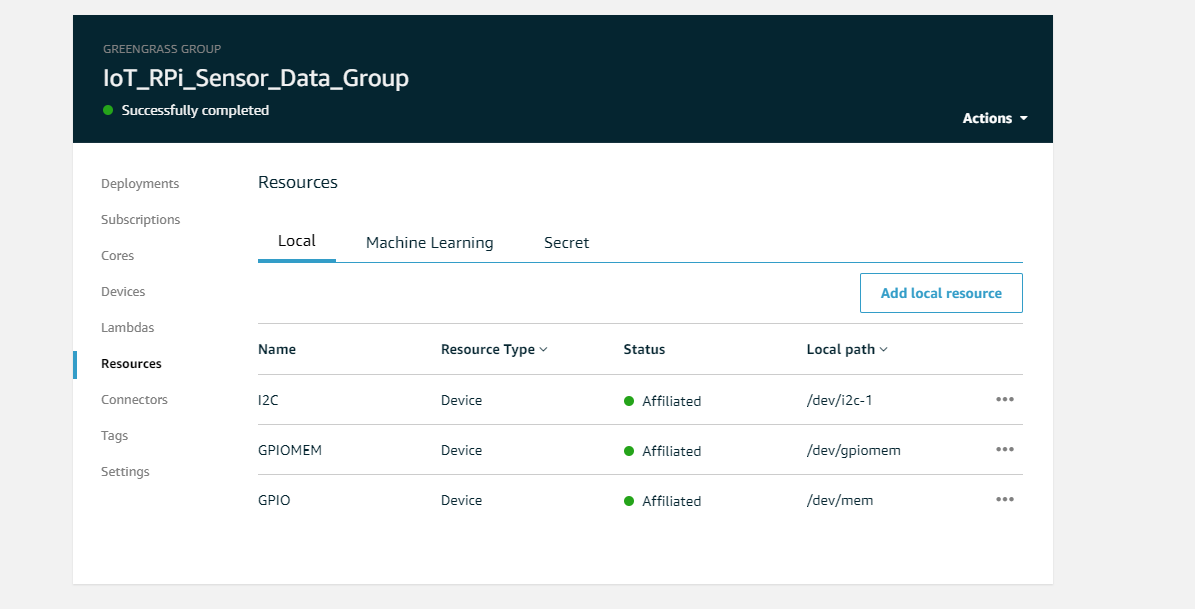
If you want to connect to your Raspberry Pi without any cables follow this guide on setting up your RPi:

<https://dev.to/vorillaz/headless-raspberry-pi-zero-w-setup-3llj>

Can use nmap -sP <your:ip:addrerss>/24 to find the RPi as long as it’s connected to your network.

Remember to allowFunctionsToRunAsRoot To allow the Lambda function access to files to access some restricted files.

Some resources that you may want to add onto your device to allow some access:



When running into issues with your Lambda functions Enable logging byu following this the “**Configure Logging for AWS IoT Greengrass**” section of this documentation <https://docs.aws.amazon.com/greengrass/latest/developerguide/greengrass-logs-overview.html#config-logs-api>

allowFunctionsToRunAsRoot within runtime

runnin greengrass daemon connects device to greengrass group

install python-pip

pip install RPi.GPIO to allow python to communicate with GPIO on Raspberry Pi