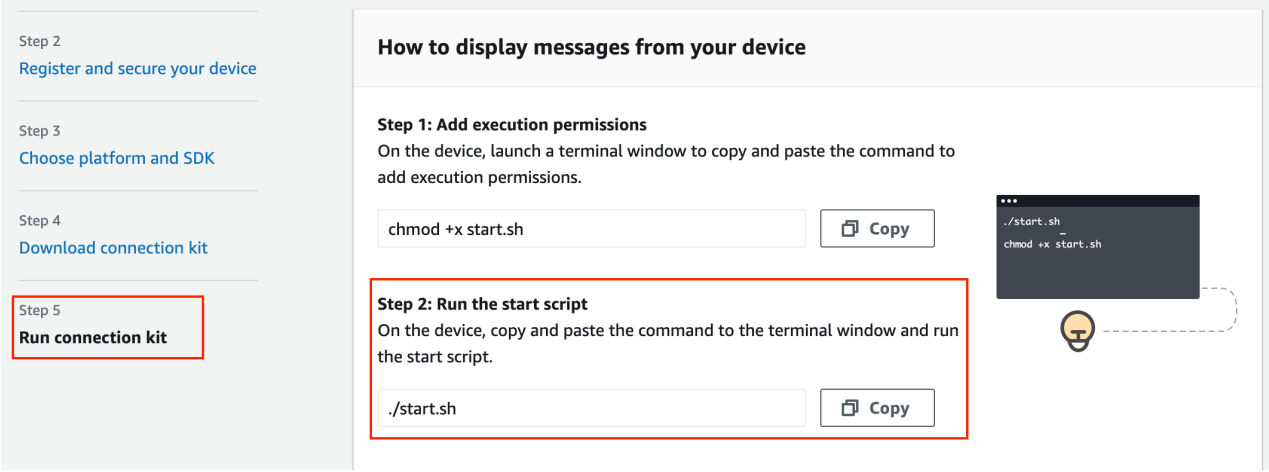
Step1:

- Connect PM2.5 sensor to LoRA. The detail is in the experiment guide CH5 section 5.2.2.
- Next, please connect LoRA to RPi.

Step2:
Please do steps 8.1 – 8.2 from the experiment guide CH 8
[note]: The new version of AWS interface is slightly different from the one shown in experiment guide, but it does not affect us doing the experiment.

Before you run the start.sh
Please modify line 26 as “python3 -m pip install awsiotsdk” first.



[note]: If you cannot install successfully, please refer to (1) or (2)
(1)
https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html

Do the steps:

- Install the required tools and libraries for the AWS IoT Device SDK
- Install AWS IoT Device SDK

(2)

To install from Github:

```
git clone https://github.com/awslabs/aws-crt-python.git
cd aws-crt-python
git submodule update --init
python3 -m pip install .
```

https://pypi.org/project/awscrt/

Step3:
Please put the file testPM2.5_aws.py[download] under the path aws-iot-device-sdk-python-v2/samples/
Then run the command:

(1) cd ../..
(2) python3 aws-iot-device-sdk-python-v2/samples/testPM2.5_aws.py --endpoint [your API endpoint] --ca_file [your Root CA] --cert [your certificate]--key [your private key]--client_id [your client id] --topic sdk/test/Python --count 0

For example:

```
pi@raspberrypi:~/aws $ python3 aws-iot-device-sdk-python-v2/samples/testPM2.5_aws.py --e
ndpoint a3c7o1yavmawig-ats.iot.ap-northeast-1.amazonaws.com --ca_file root-CA.crt --cert
 myRPi.cert.pem --key myRPi.private.key --client_id basicPubSub --topic sdk/test/Python
 --count 0
```

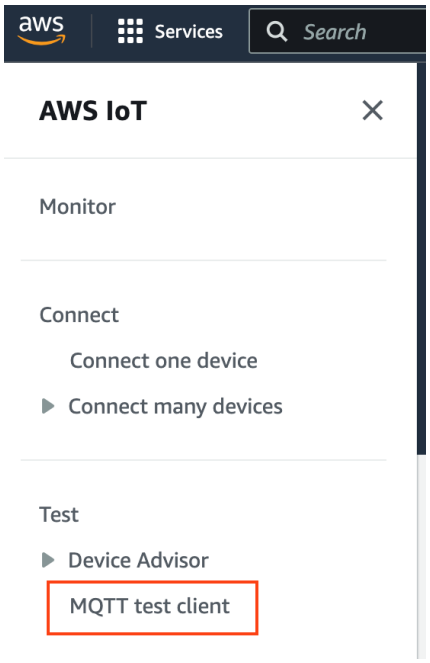
The full details, such like [your API endpoint] / [your Root CA]/ [your certificate] / [your private key] / [your client id] can be found in the start.sh(in line 36).

If it connects to the AWS, the terminal may show the progress like that:

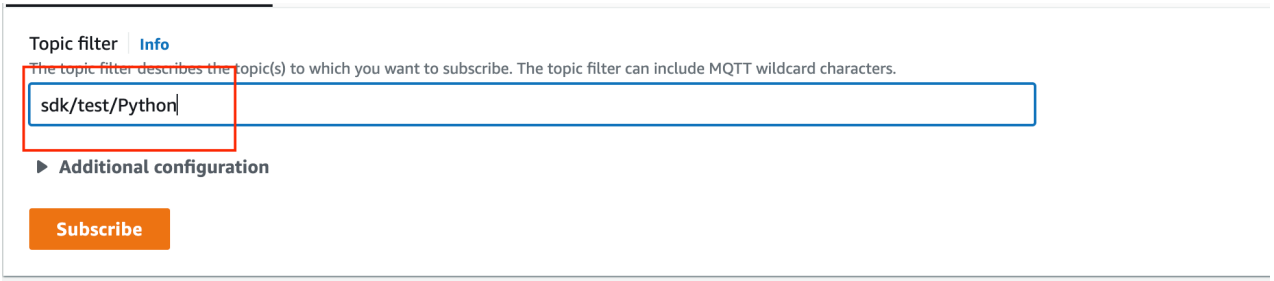
```
Connecting to a3c7o1yavmawig-ats.iot.ap-northeast-1.amazonaws.com with client ID 'basicP
ubSub'...
Connected!
Subscribing to topic 'sdk/test/Python'...
Subscribed with QoS.AT_LEAST_ONCE
Sending messages until program killed
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [1]
Received message from topic 'sdk/test/Python': b'"-149.52 [1]"'
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [2]
Received message from topic 'sdk/test/Python': b'"-149.52 [2]"'
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [3]
Received message from topic 'sdk/test/Python': b'"-149.52 [3]"'
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [4]
Received message from topic 'sdk/test/Python': b'"-149.52 [4]"'
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [5]
Received message from topic 'sdk/test/Python': b'"-149.52 [5]"'
-149.52 0
Publishing message to topic 'sdk/test/Python': -149.52 [6]
Received message from topic 'sdk/test/Python': b'"-149.52 [6]"'
```

TA's Arduino is broken, so the value of PM 2.5 is wrong ☹. Normally, the value is non-negative.

Step 4: Here, we will use the client service to check whether the device successfully connect to the AWS. Pleae return to your AWS platform then enter to the Aws IoT > Test > MQTT test client



Step 5: subscribe "sdk/test/Python"



If AWS successfully connects to RPi, it will show: