



Lab no 08 –Connect AWS Core to Raspberry pi

The purpose of this lab is to connect Raspberry Pi in pervious labs to publish/subscribe to amazon AWS core.

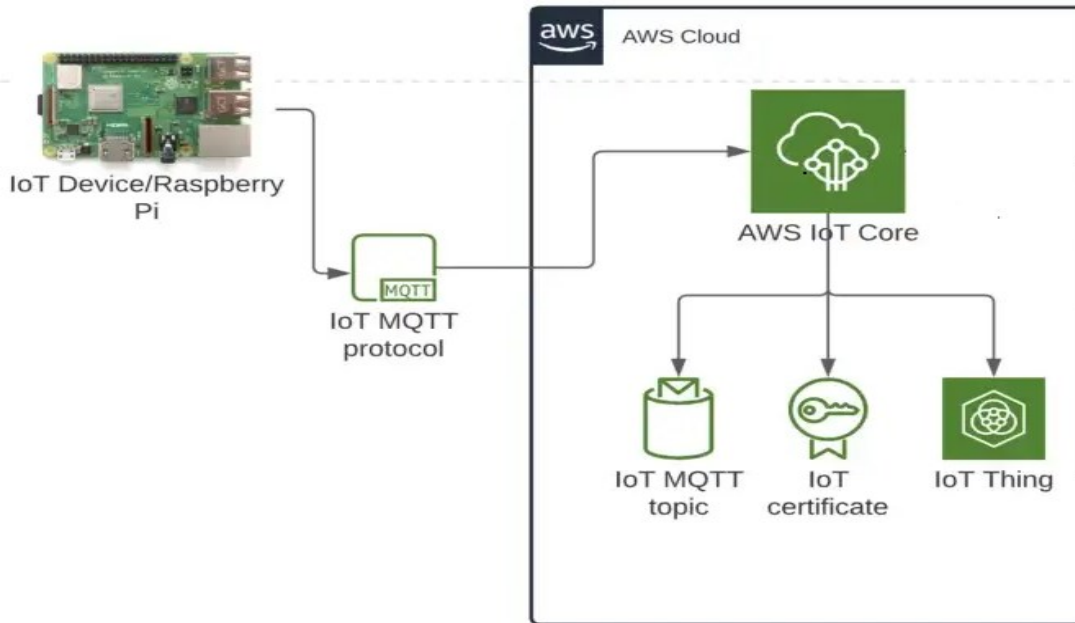
Parts: -

1. Create IoT Core Thing and Download new Certificates.
2. Create policy.
3. Associate policy.
4. Create Cognito.
5. Configure and run python code in raspberry pi.
6. subscribe AWS core to Raspberry pi.

Required Resources

- 1 PC with Internet access.
- Account in AWS Management Console.
- Raspberry Pi.
- Adapter.
- Ethernet Cable.
- Jumpers.
- Breadboard.

Technical Architecture



Part 1: Create IoT Core Thing and download new Certificates.

1. In the AWS Management Console, click Services, and then click IoT Core to open the IoT Console.
2. Click Get started.
3. Expand Manage on the left menu.
4. Click Things.
5. Click Register a thing.
6. Click Create a single thing (any of the two buttons with that name works).
7. For Name, enter Raspberry pi and click Next.
8. Click Create certificate to automatically generate a Certificate, a Public Key and a Private Key using AWS IoT's Certificate Authority that you will then download. Note that you could create your own Certificate Authority. You could also create your own Private Key on your end and generate a Certificate Signing Request that you would upload here to be signed by AWS IoT's Certificate Authority. Both could work with the exercise, but you would have to adapt some of the later commands. For simplicity, use the One-click certificate creation.
9. Make sure that you take the following steps before going to the next screen as you would have to restart the creation of this Certificate.
10. Click the Activate button to activate the Certificate so it can be used later to connect to AWS IoT Core by your Thing.
11. Click the Download link next to A certificate for this thing.
12. Rename this file to certificate.pem.crt.
13. Click the Download link next to A private key.

Lab no 08 –Connect AWS Core to Raspberry pi

14. Rename this file to private.pem.key.
15. Click the Download link next to A public key.
16. Rename this file to public.pem.key.
17. Click the Download link next to Root RSA.
18. Rename this file to root-ca.pem.
19. Save each of the downloaded files to the certificates sub-directory name “AWS IoT”.
20. Raspberry pi has now been created.
21. Connect to raspberry pi as shown in lab 3 and copy folder “AWS IoT” and paste folder to raspberry pi.

Part 2: Create policy

1. Open the AWS IoT Core console.
2. In the left navigation pane, choose Secure.
3. Under Secure, choose Policies.
4. If you have existing AWS IoT Core policies, then choose Create to create a new policy.
5. On the Create a policy page, enter a Name for your policy “Raspberry pi policy”.
Under Add statements, do the following:
6. For Action, enter iot:*.
Important: Allowing all AWS IoT actions (iot:*) is useful for testing.
7. For Resource ARN, enter *.
8. For Effect, select the Allow check box.
9. Choose Create.

Part 3: Associate policy.

1. Open the AWS IoT Core console.
2. In the left navigation pane, choose Secure.
3. Under Secure, choose certificates.
4. In certificates, choose ... and choose attach policy “raspberry pi policy” next click attach.
5. In certificates, choose ... and choose attach thing “raspberry pi” next click attach.

Part 4: Create Cognito.

1. In the AWS Management Console, click Services, and then click Cognito to go to the Cognito console.
2. Click Manage Identity Pools. It will automatically start the screen of creation of an Identity Pool as this is your first. If it's not, click the Create new identity pool button.
3. In the Identity pool name field, enter AWSIdentityPool.
4. Click the check mark next to Enable access to unauthenticated identities as this is what you will be using to authenticate the web application.
5. Click Create Pool.
6. Expand the View Details link to see the name of the roles that will be created. You can also see the IAM Policy that will be attached by clicking on the View Policy Document.
7. Edit in policy attached, in resources between ** add ARN of thing.

Note to get ARN follow these steps:

Lab no 08 –Connect AWS Core to Raspberry pi

- a. In the AWS Management Console, click Services, and then click IoT Core to open the IoT Console.
 - b. Click Get started.
 - c. Expand Manage on the left menu.
 - d. Click Things.
 - e. Click in raspberry pi and copy ARN, example arn:aws:iot:us-east-1:123456789012:thing.
8. Past ARN between ** in resources
 9. Click Allow.

Part 5: Configure and run python code in raspberry pi.

- **Install pip and the AWS IoT SDK for Python**

Install the AWS IoT SDK for Python v2 by running the following from the command line:

```
pip install awsiotsdk  
or
```

Install the AWS IoT Device SDK for Python (the previous SDK version) by running the following command:

```
pip install AWSIoTPythonSDK
```

- **Create a Python program file**

Python code to publish to AWS in topic “home/helloworld” random values

```
import time  
from random import randrange, uniform  
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient  
myMQTTClient = AWSIoTMQTTClient("RishabClientID") #random key, if another connection using the same key  
is opened the previous one is auto closed by AWS IOT  
myMQTTClient.configureEndpoint("a36ppf9m9zgs9-ats.iot.us-east-1.amazonaws.com", 8883)  
  
myMQTTClient.configureCredentials("/home/pi/AWSIoT/root-ca.pem", "/home/pi/AWSIoT/private.pem.key",  
"/home/pi/AWSIoT/certificate.pem.crt")  
  
myMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing  
myMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz  
myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec  
myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec  
print ('Initiating Realtime Data Transfer From Raspberry Pi...')  
myMQTTClient.connect()  
while True:  
    randomNumber = uniform(20.0, 21.0)  
    print("publish from raspberry pi From Raspberry Pi..." +str(randomNumber))  
    myMQTTClient.publish(  
        topic="home/helloworld",  
        QoS=1,  
        payload=randomNumber)  
    time.sleep(2)
```

Lab no 08 –Connect AWS Core to Raspberry pi

Part 6: subscribe AWS core to Raspberry pi.

1. Open the AWS IoT Core console.
2. In the left navigation pane, choose test mqtt client.
3. Subscribe topic “home/helloworld” and receive raspberry pi messages as shown in <https://drive.google.com/file/d/1X5z1iZS-j31dRen-ICqiLkFfMFtA2sa8/view?usp=sharing>

Note:

Connect a Raspberry Pi or another device

<https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html>

Connecting Raspberry Pi to AWS Iot Core : Setup and code Using Python and AWS IOT(video)

<https://www.youtube.com/watch?v=kPLafcrng-c>