Networking the Physical World: Lab 2

Samir Farhat Dominguez

March 25, 2022

1 Discussion

Read up on & discuss the role of Policies in Amazon IoT

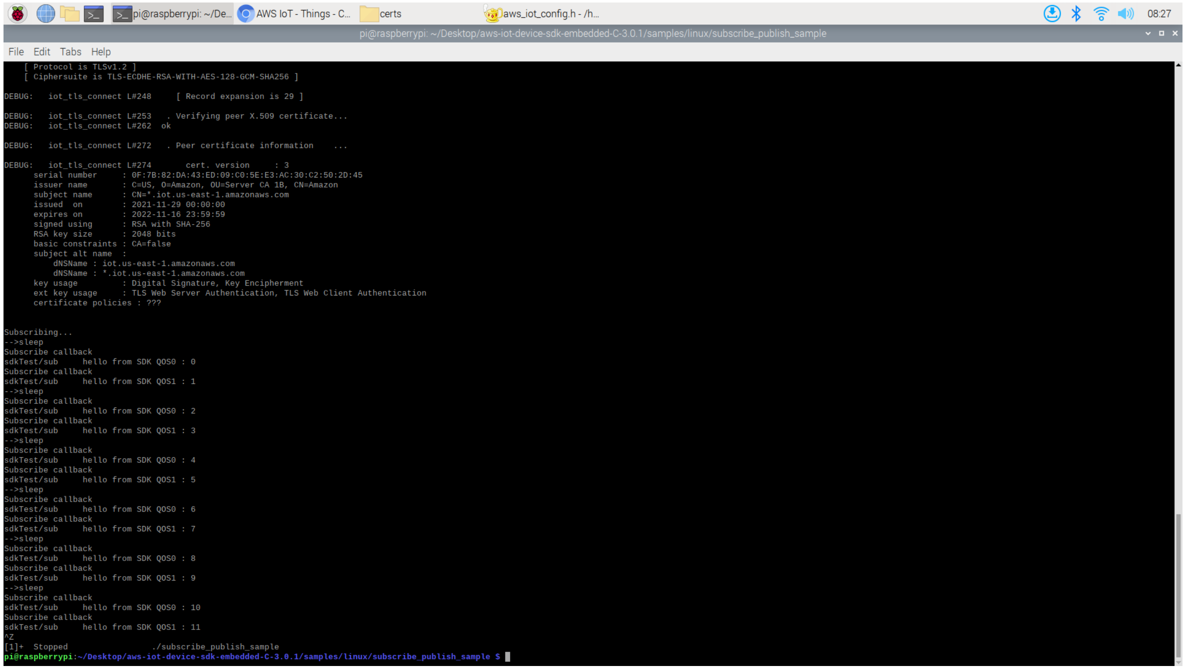
In the context of AWS, and particularly Amazon IoT, policies are employed in order to manage access and permissions. Policies are created with a set of rules and permissions, and then assigned to users, groups, roles, or IoT devices/Things. There are several policy types employed in the Amazon ecosystem. These are Identity-based, resource-based, permission boundaries, session policies and access control policies. Within IoT we can use identity based policies to allow a device to identify itself to access permissions or functions; resource based to give devices access to DBMS servers or DBs; permission boundaries and access control boundaries to restrict the capabilities of the particular IoT device or IoT device users; as well as session policies to restrict permissions of users interacting with an IoT device. All these come together in order to protect and secure hardware, keys, data, networking and many other necessary points of interest for IoT systems.

Read up on & discuss the role of Thing (Device) Shadows in Amazon IoT and what role they play when something is not connected to the Internet

Thing shadows are a way to extend the usability of IoT devices, and multiple can be assigned to a single registered IoT device/Thing. They allow the state data and functionality of the device to remain even when not connected to AWS IoT. Creation and assignment of shadows is at the discretion of users, although a hidden standard shadow is created at the initial creation if the Thing, just for the sake of consistency. These shadows facilitate the continuous creation and exchange of data for the IoT device, and can allow Amazon IoT to keep updated indirectly by tracking a connected device that is able to connect with the shadow. Shadows can be named and unnamed, of which the latter type can only have a single one per device. The APIs that access these differ slightly, as named shadows need to enable the creation of different views and resolutions of the device, while unnamed cannot.

2 Screenshots of "Run Sample Applications"

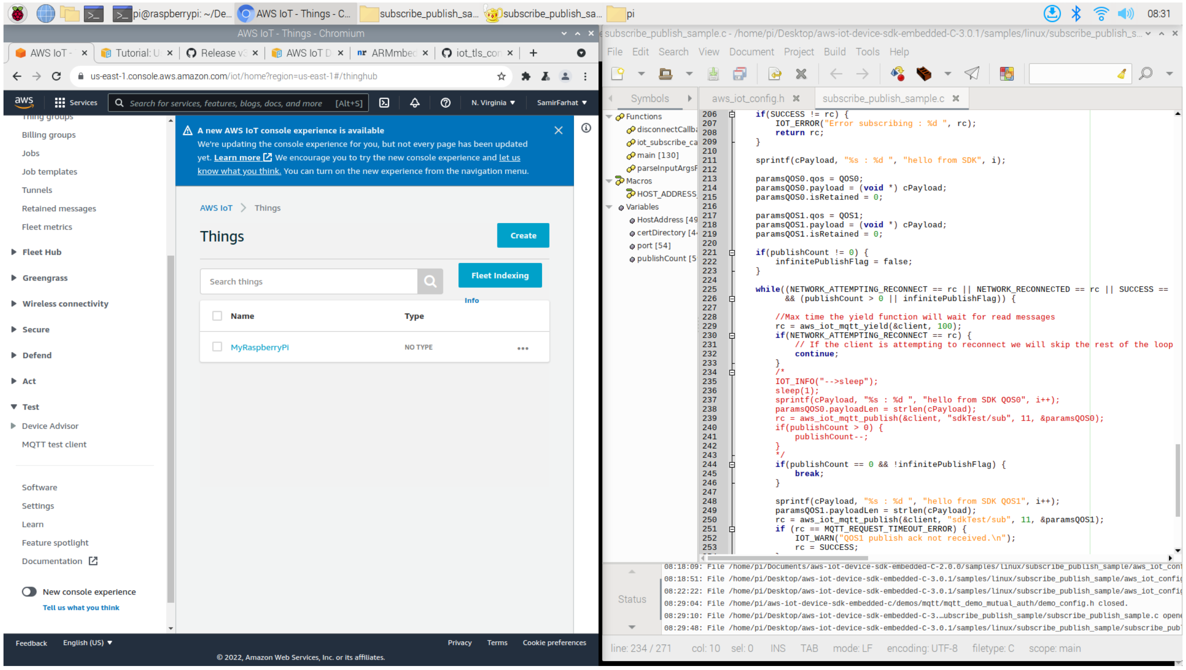
5


pi@raspberrypi: ~/De...
AWS IoT - Things - C...
certs
aws_iot_config.h - /h...
08:27
pi@raspberrypi: ~/Desktop/aws-iot-device-sdk-embedded-C-3.0.1/samples/linux/subscribe_publish_sample
File
Edit
Tabs
Help

```
    [ Protocol is TLSv1.2 ]
    [ Ciphersuite is TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256 ]

DEBUG:   iot_tls_connect L#248     [ Record expansion is 29 ]

DEBUG:   iot_tls_connect L#253   . Verifying peer X.509 certificate...
DEBUG:   iot_tls_connect L#262  ok

DEBUG:   iot_tls_connect L#272   . Peer certificate information    ...

DEBUG:   iot_tls_connect L#274       cert. version     : 3
      serial number     : 0F:7B:82:DA:43:ED:09:C0:5E:E3:AC:30:C2:50:2D:45
      issuer name       : C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
      subject name      : CN=*.iot.us-east-1.amazonaws.com
      issued  on        : 2021-11-29 00:00:00
      expires on        : 2022-11-16 23:59:59
      signed using      : RSA with SHA-256
      RSA key size      : 2048 bits
      basic constraints : CA=false
      subject alt name  :
          dNSName : iot.us-east-1.amazonaws.com
          dNSName : *.iot.us-east-1.amazonaws.com
      key usage         : Digital Signature, Key Encipherment
      ext key usage     : TLS Web Server Authentication, TLS Web Client Authentication
      certificate policies : ???


Subscribing...
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 0
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 2
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 3
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 4
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 5
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 6
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 7
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 8
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 9
-->sleep
Subscribe callback
sdkTest/sub     hello from SDK QOS0 : 10
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 11
^Z
[1]+  Stopped                 ./subscribe_publish_sample
pi@raspberrypi:~/Desktop/aws-iot-device-sdk-embedded-C-3.0.1/samples/linux/subscribe_publish_sample $ 
```

Figure 1

6



A new AWS IoT console experience is available

We're updating the console experience for you, but not every page has been updated yet. Learn more We encourage you to try the new console experience and let us know what you think. You can turn on the new experience from the navigation menu.

AWS IoT > Things

Things

Create

Search things

Fleet Indexing

Info

□ Name	Type
□ MyRaspberryPi	NO TYPE

© 2022, Amazon Web Services, Inc. or its affiliates.

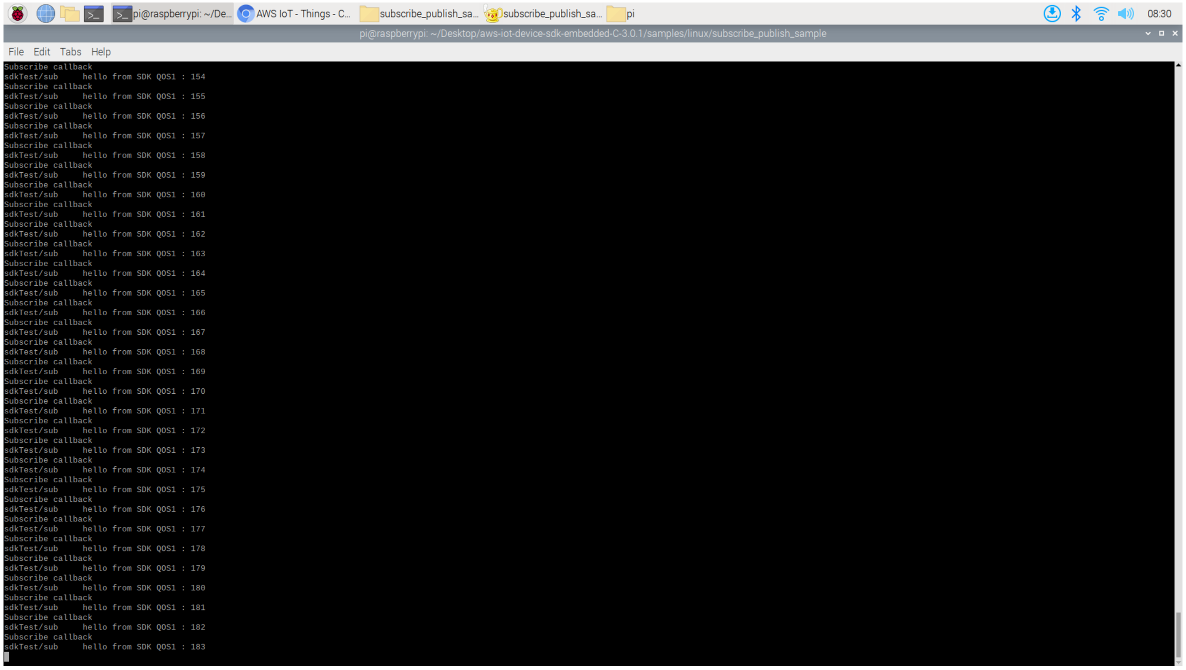
```c
    if(SUCCESS != rc) {
        IOT_ERROR("Error subscribing : %d ", rc);
        return rc;
    }

    sprintf(cPayload, "%s : %d ", "hello from SDK", i);

    paramsQOS0.qos = QOS0;
    paramsQOS0.payload = (void *) cPayload;
    paramsQOS0.isRetained = 0;

    paramsQOS1.qos = QOS1;
    paramsQOS1.payload = (void *) cPayload;
    paramsQOS1.isRetained = 0;

    if(publishCount != 0) {
        infinitePublishFlag = false;
    }

    while((NETWORK_ATTEMPTING_RECONNECT == rc || NETWORK_RECONNECTED == rc || SUCCESS ==
          && (publishCount > 0 || infinitePublishFlag)) {

        //Max time the yield function will wait for read messages
        rc = aws_iot_mqtt_yield(&client, 100);
        if(NETWORK_ATTEMPTING_RECONNECT == rc) {
            // If the client is attempting to reconnect we will skip the rest of the loop
            continue;
        }
        /*
        IOT_INFO("-->sleep");
        sleep(1);
        sprintf(cPayload, "%s : %d ", "hello from SDK QOS0", i++);
        paramsQOS0.payloadLen = strlen(cPayload);
        rc = aws_iot_mqtt_publish(&client, "sdkTest/sub", 11, &paramsQOS0);
        if(publishCount > 0) {
            publishCount--;
        }
        */
        if(publishCount == 0 && !infinitePublishFlag) {
            break;
        }

        sprintf(cPayload, "%s : %d ", "hello from SDK QOS1", i++);
        paramsQOS1.payloadLen = strlen(cPayload);
        rc = aws_iot_mqtt_publish(&client, "sdkTest/sub", 11, &paramsQOS1);
        if (rc == MQTT_REQUEST_TIMEOUT_ERROR) {
            IOT_WARN("QOS1 publish ack not received.\n");
            rc = SUCCESS;
```

```
08:18:09: File /home/pi/Documents/aws-iot-device-sdk-embedded-C-2.0.0/samples/linux/subscribe_publish_sample/aws_iot_conf
08:18:51: File /home/pi/Desktop/aws-iot-device-sdk-embedded-C-3.0.1/samples/linux/subscribe_publish_sample/aws_iot_config
08:22:22: File /home/pi/Desktop/aws-iot-device-sdk-embedded-C-3.0.1/samples/linux/subscribe_publish_sample/aws_iot_config
08:29:04: File /home/pi/aws-iot-device-sdk-embedded-c/demos/mqtt/mqtt_demo_mutual_auth/demo_config.h closed.
08:29:10: File /home/pi/Desktop/aws-iot-device-sdk-embedded-C-3.…ubscribe_publish_sample/subscribe_publish_sample.c opene
08:29:48: File /home/pi/Desktop/aws-iot-device-sdk-embedded-C-3.0.1/samples/linux/subscribe_publish_sample/subscribe_pub
```

line: 234 / 271   col: 10   sel: 0   INS   TAB   mode: LF   encoding: UTF-8   filetype: C   scope: main

Figure 2

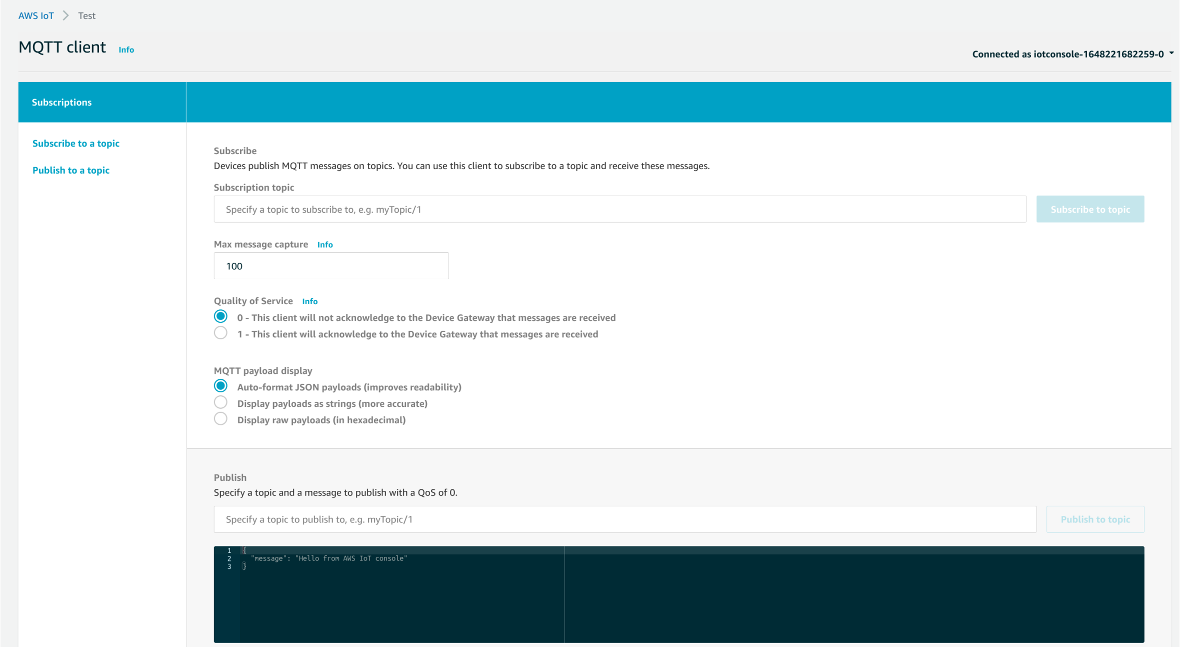
7



```
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 154
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 155
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 156
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 157
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 158
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 159
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 160
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 161
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 162
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 163
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 164
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 165
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 166
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 167
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 168
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 169
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 170
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 171
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 172
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 173
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 174
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 175
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 176
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 177
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 178
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 179
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 180
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 181
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 182
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 183
```

Figure 3

8 and 9



MQTT client

Connected as iotconsole-1648221682259-0

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

Specify a topic to subscribe to, e.g. myTopic/1

Max message capture: 100

Quality of Service

● 0 - This client will not acknowledge to the Device Gateway that messages are received

○ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

● Auto-format JSON payloads (improves readability)

○ Display payloads as strings (more accurate)

○ Display raw payloads (in hexadecimal)

Publish

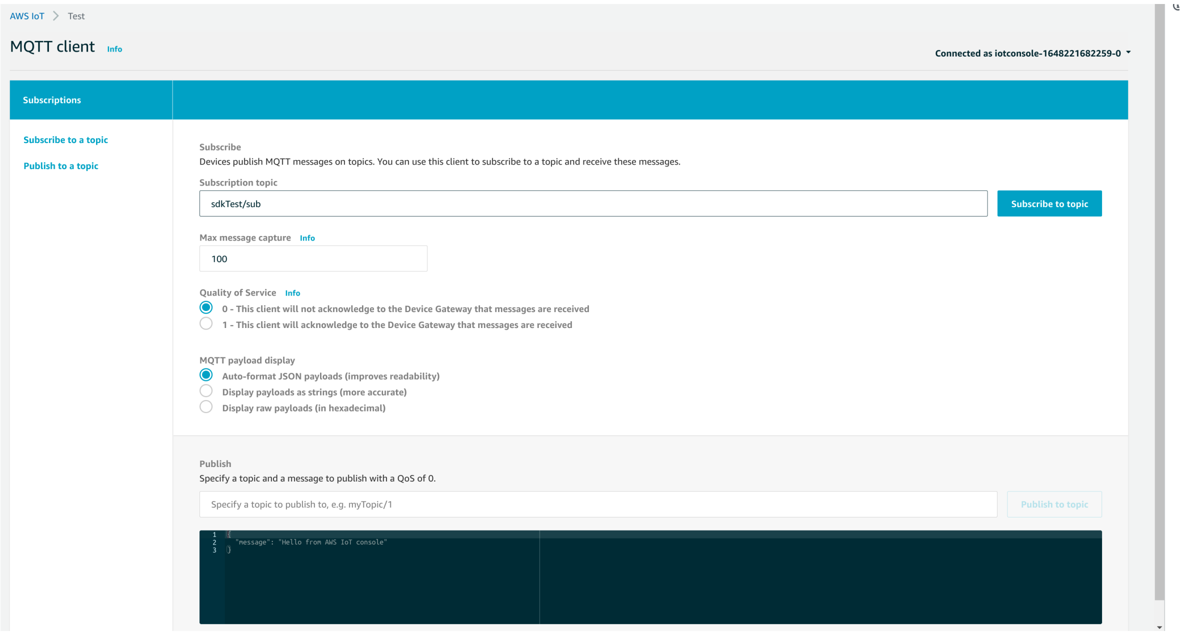
Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

```
{
  "message": "Hello from AWS IoT console"
}
```

Figure 4

10



MQTT client

Connected as iotconsole-1648221682259-0

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic: sdkTest/sub

Max message capture: 100

Quality of Service

● 0 - This client will not acknowledge to the Device Gateway that messages are received

○ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

● Auto-format JSON payloads (improves readability)

○ Display payloads as strings (more accurate)

○ Display raw payloads (in hexadecimal)

Publish

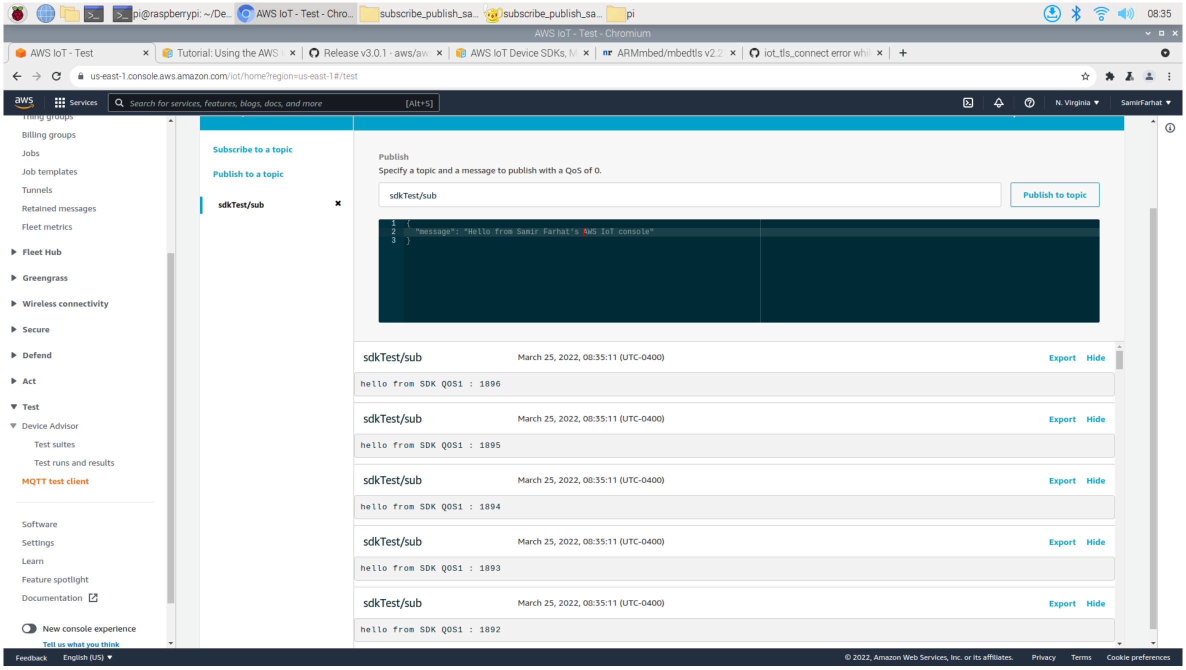
Specify a topic and a message to publish with a QoS of 0.

Specify a topic to publish to, e.g. myTopic/1

```
{
  "message": "Hello from AWS IoT console"
}
```

Figure 5

11 and 12



Publish

Specify a topic and a message to publish with a QoS of 0.

sdkTest/sub

```
{
  "message": "Hello from Samir Farhat's AWS IoT console"
}
```

sdkTest/sub March 25, 2022, 08:35:11 (UTC-0400)

```
hello from SDK QOS1 : 1896
```

sdkTest/sub March 25, 2022, 08:35:11 (UTC-0400)

```
hello from SDK QOS1 : 1895
```

sdkTest/sub March 25, 2022, 08:35:11 (UTC-0400)

```
hello from SDK QOS1 : 1894
```

sdkTest/sub March 25, 2022, 08:35:11 (UTC-0400)

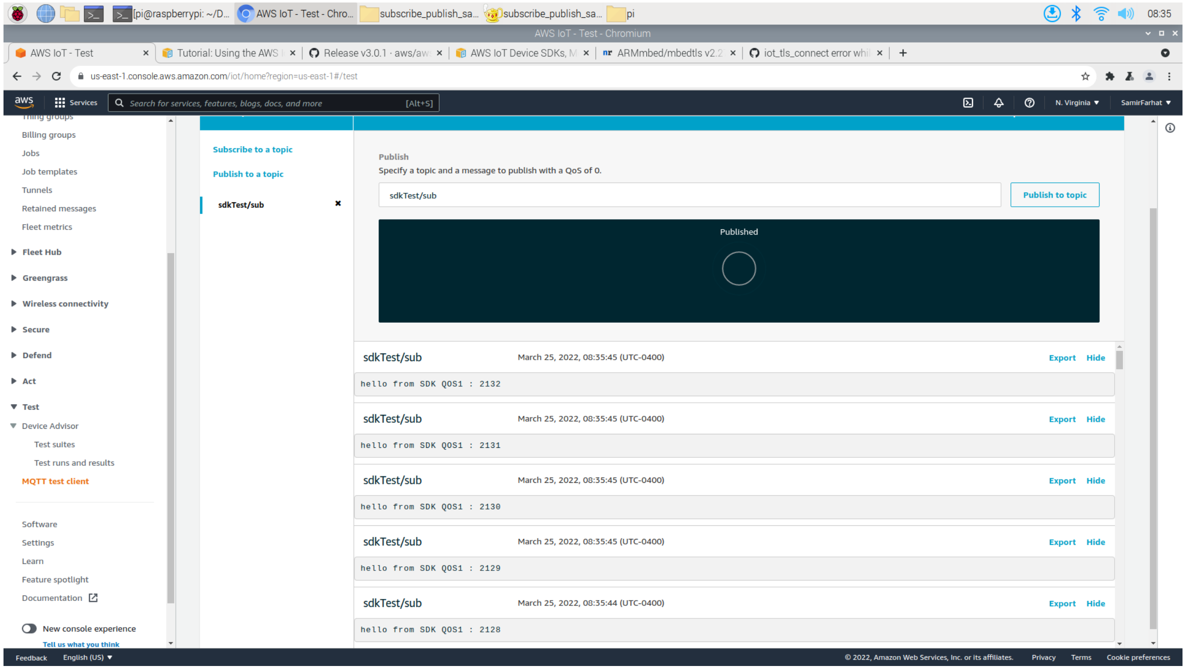
```
hello from SDK QOS1 : 1893
```

sdkTest/sub March 25, 2022, 08:35:11 (UTC-0400)

```
hello from SDK QOS1 : 1892
```

Figure 6

13



sdkTest/sub March 25, 2022, 08:35:45 (UTC-0400)

```
hello from SDK QOS1 : 2132
```

sdkTest/sub March 25, 2022, 08:35:45 (UTC-0400)

```
hello from SDK QOS1 : 2131
```

sdkTest/sub March 25, 2022, 08:35:45 (UTC-0400)

```
hello from SDK QOS1 : 2130
```

sdkTest/sub March 25, 2022, 08:35:45 (UTC-0400)

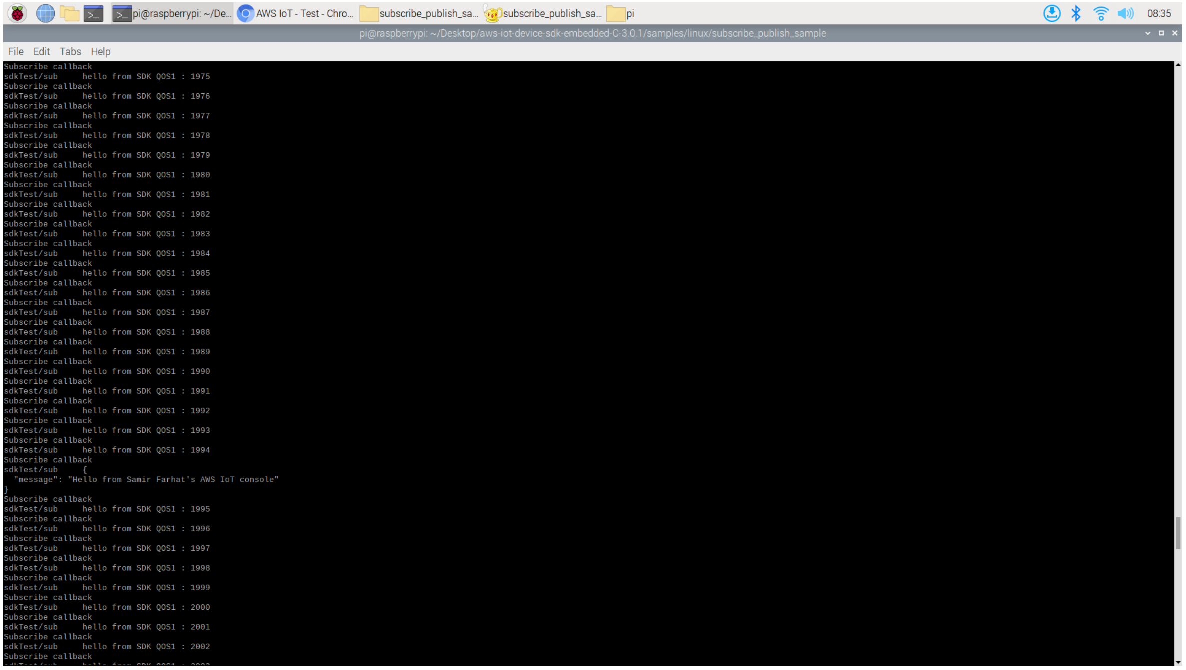
```
hello from SDK QOS1 : 2129
```

sdkTest/sub March 25, 2022, 08:35:44 (UTC-0400)

```
hello from SDK QOS1 : 2128
```

Figure 7

14



```
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1975
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1976
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1977
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1978
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1979
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1980
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1981
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1982
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1983
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1984
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1985
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1986
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1987
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1988
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1989
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1990
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1991
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1992
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1993
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1994
Subscribe callback
sdkTest/sub     {
  "message": "Hello from Samir Farhat's AWS IoT console"
}
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1995
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1996
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1997
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1998
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 1999
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 2000
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 2001
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 2002
Subscribe callback
sdkTest/sub     hello from SDK QOS1 : 2003
```

Figure 8