Integration of MQTTfx client to AWS IoT Console

|  |  |
| --- | --- |
| Title | Integration of MQTTfx client to AWS IoT Console |
| By | Meenakshi Murugappan |
| Student Id: | L00156569 |

# Summary

This document describes the implementation of the virtual devices in MQTTfx client and publishes the message to AWS IoT Console

It describes the following tasks:

1. Install MQTTfx client in the system.
2. Creating the AWS IoT Policy
3. Attaching the certificate to IoT Policy
4. Publish the message from MQTTfx client to AWS IoT console.
5. Subscribe and test the topic using the MQTT client in the AWS IoT console.

Contents

[Summary 1](#_Toc45359045)

[Resources and Prerequisites 3](#_Toc45359046)

[Introduction 3](#_Toc45359047)

[Steps 4](#_Toc45359048)

[Problems Faced 9](#_Toc45359049)

[Summary 9](#_Toc45359050)

[References 9](#_Toc45359051)

# Resources and Prerequisites

1. Valid AWS credentials to sign in to the AWS console to access the services

# Introduction

This document provides the instruction to integrate the MQTTfx client [[1]](https://mqttfx.jensd.de/index.php/download) to act as a virtual device and publish the message to AWS IoT console. To publish the message to AWS IoT console, as all Amazon web services strict security considerations were in place for any device to connect to AWS IoT core services [[2]](https://aws.amazon.com/iot-core/). To connect any devices with AWS we need to attach an IoT policy and TLS 1.2 CA certification must be attached to the policy, which can be downloaded from the AWS website. A high-level architecture diagram for the project “AWSIoTCoreforColdFactorySME” is shown in the below image. In this document, since it is a project with less time consideration, all temperature sensors are created using MQTTfx client (virtual devices) to connect with AWS IoT services.

A picture containing map

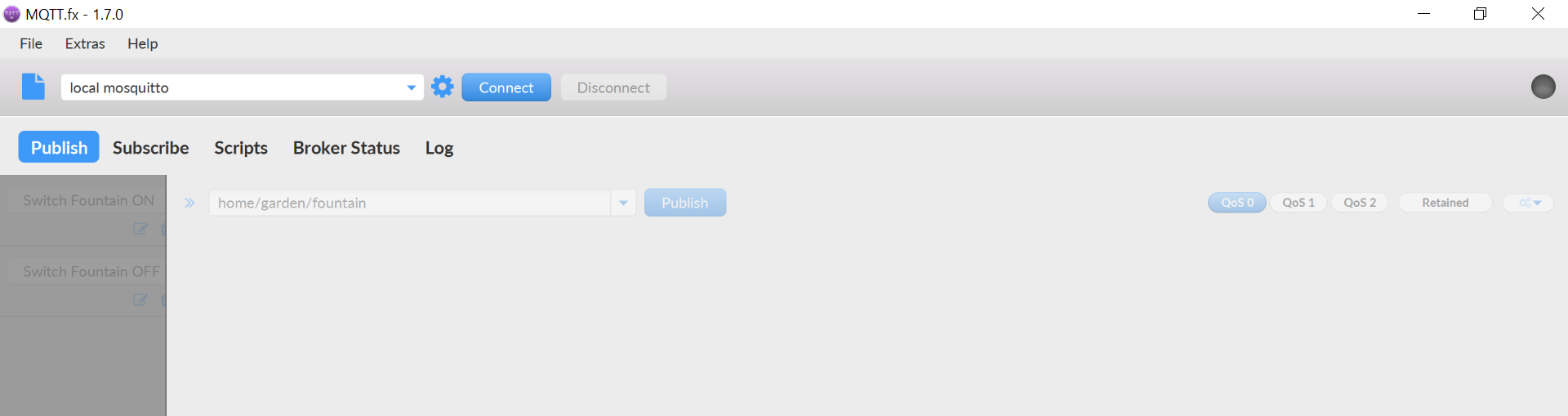
Description automatically generated

**Figure 1: Sample Architecture for AWS IoT services in a Cold Storage SME**

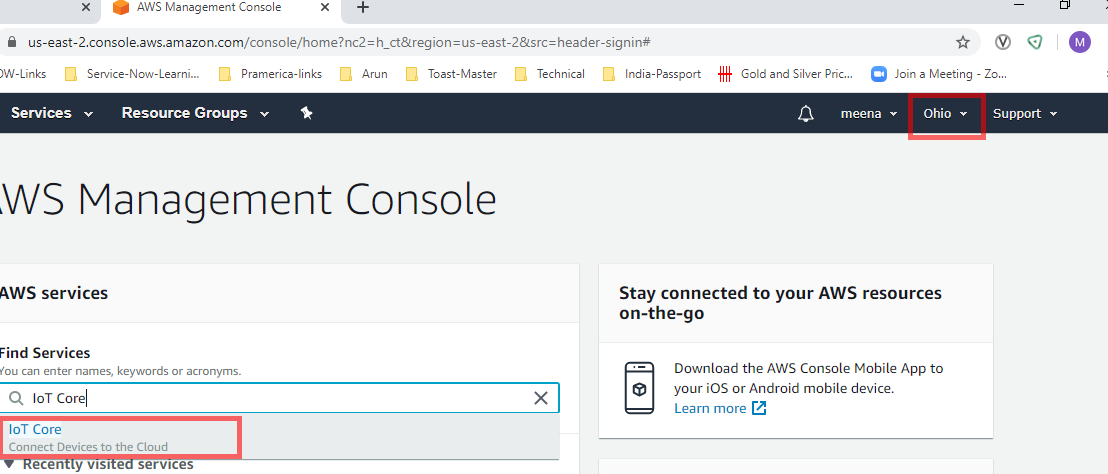
The below steps described in this document explains the sample scenario of publishing the device topics from MQTTfx client and tested by subscribing to the topic in AWS MQTT test client.

# Steps

1. Installed the JavaFX based MQTT Client in a system to publish the topic to AWS by downloading from the link.

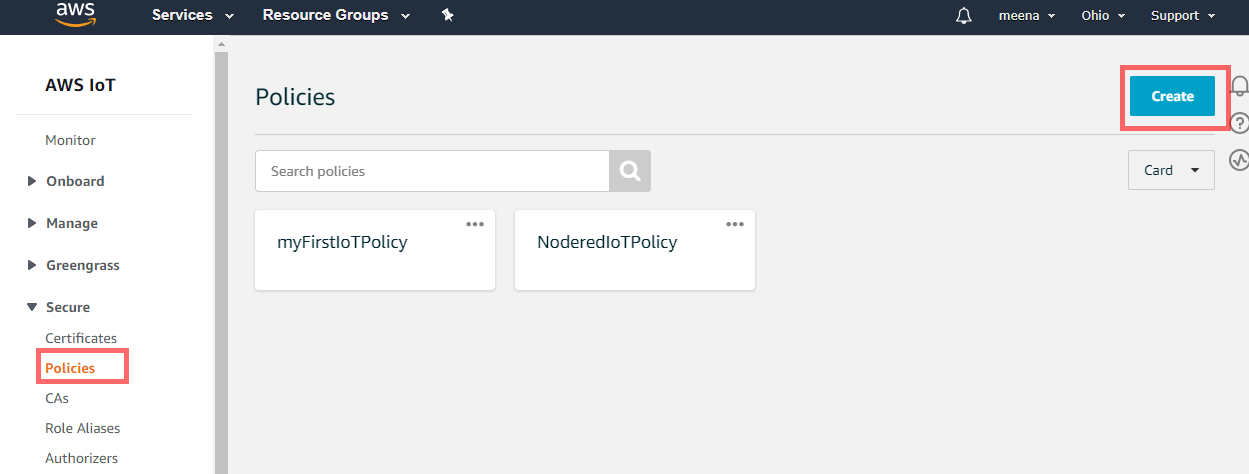


1. Once log in to the AWS Console, navigate to the “AWS IoT Core” services as shown below

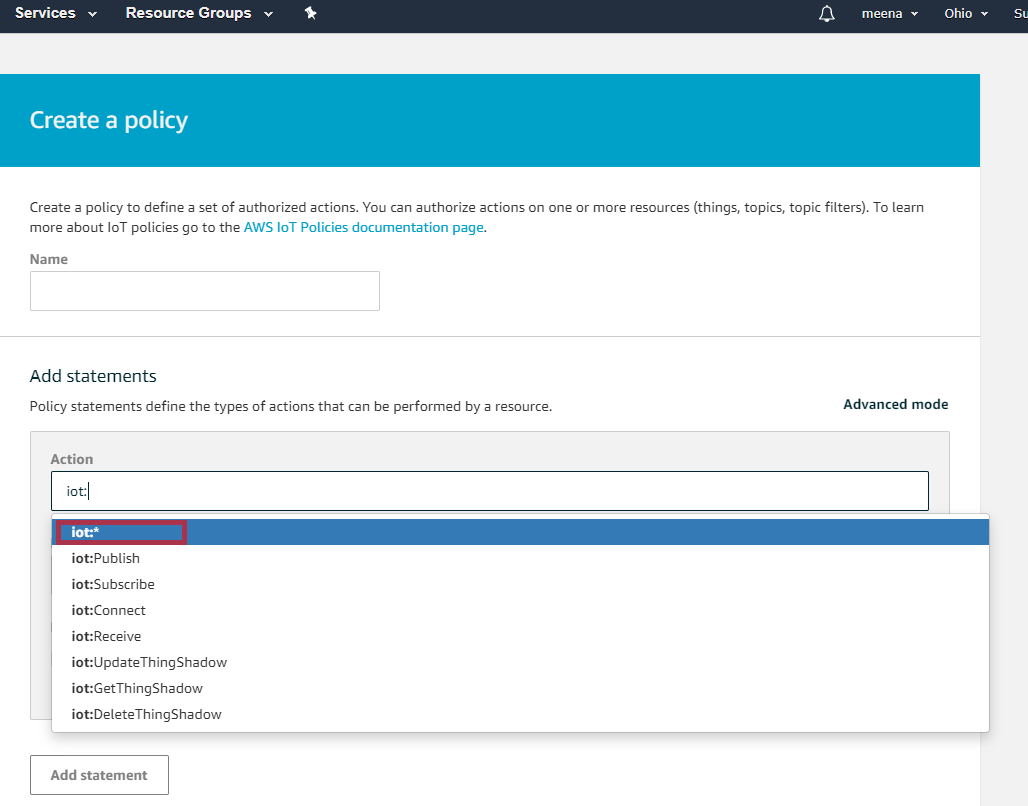


Note: Use the default region US-east-2, this is the first region where all new services are introduced. In AWS there are multiple regions. Ensure the region is the same across the lab exercises.

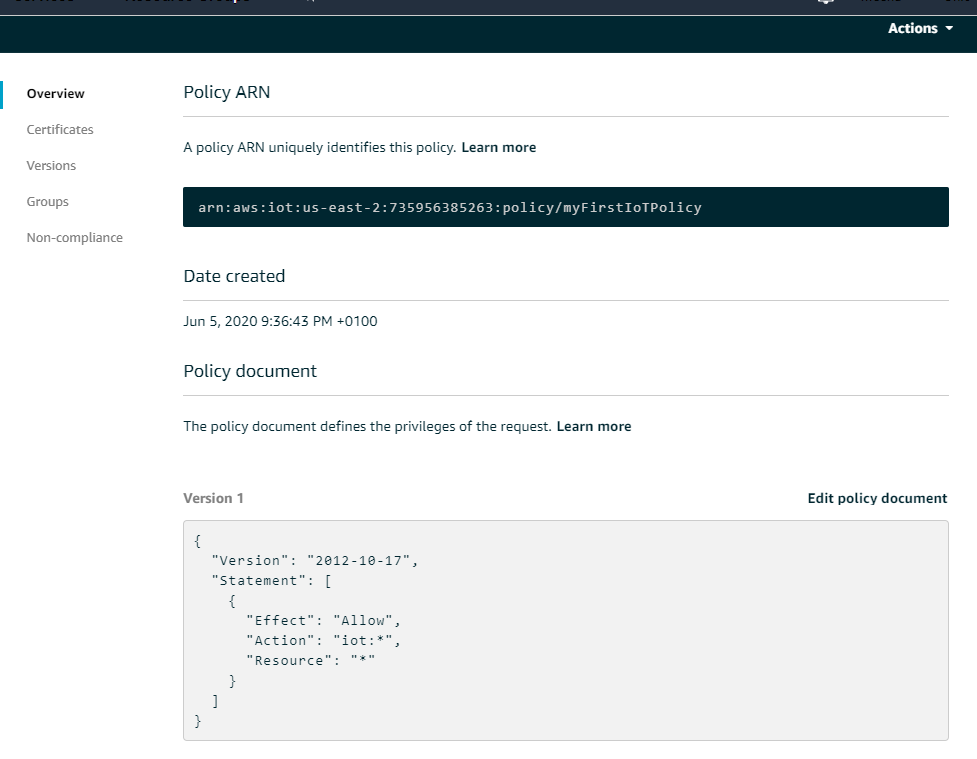
1. Click the create policy “myFirstIoTPolicy” on the left navigation of the AWS IoT secure to create a policy. Please refer to the link [[3]](https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html?icmpid=docs_iot_console) for more details on policy action.

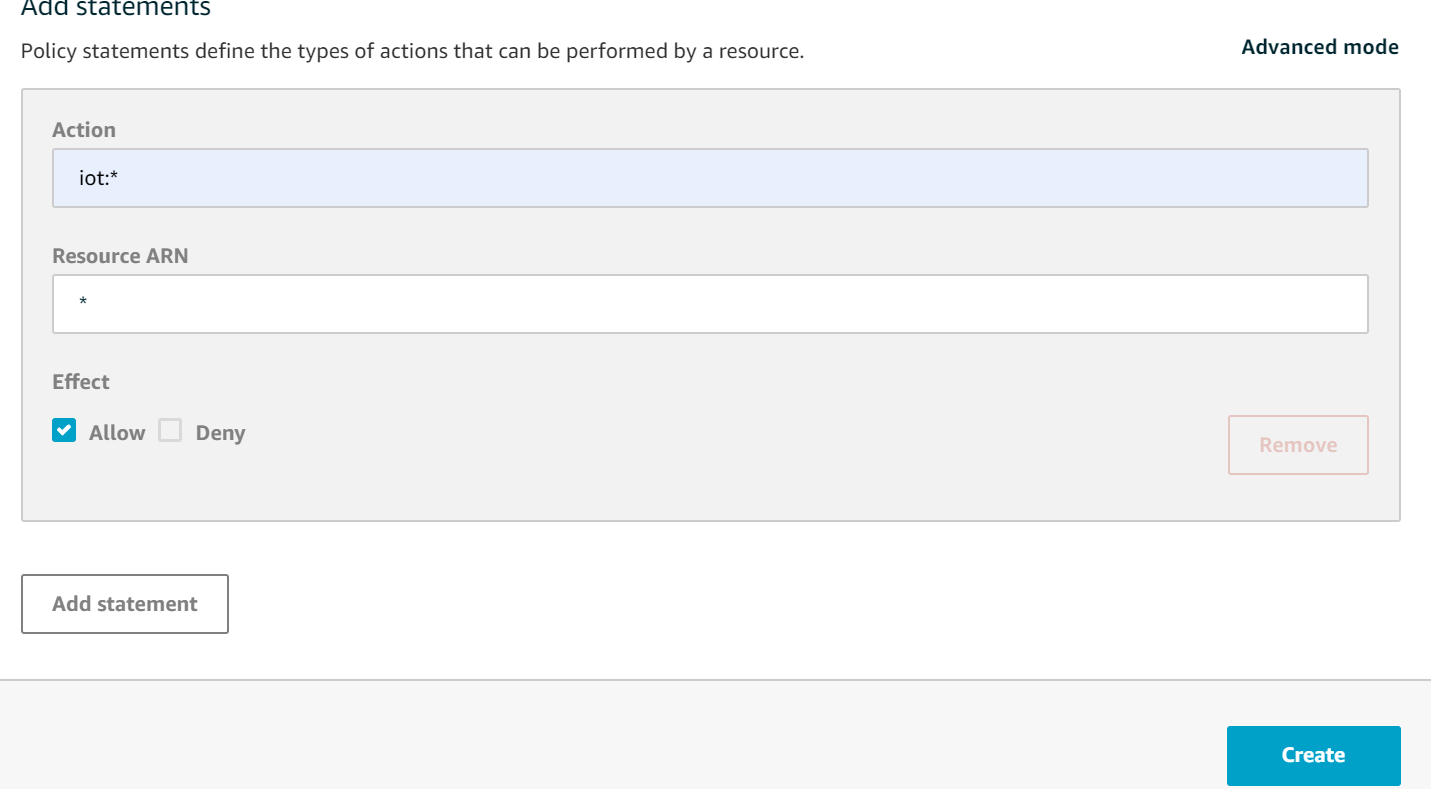


1. Attached a statement to the policy to allow IoT action, for this exercise we allowed IoT operations by IoT:\*.

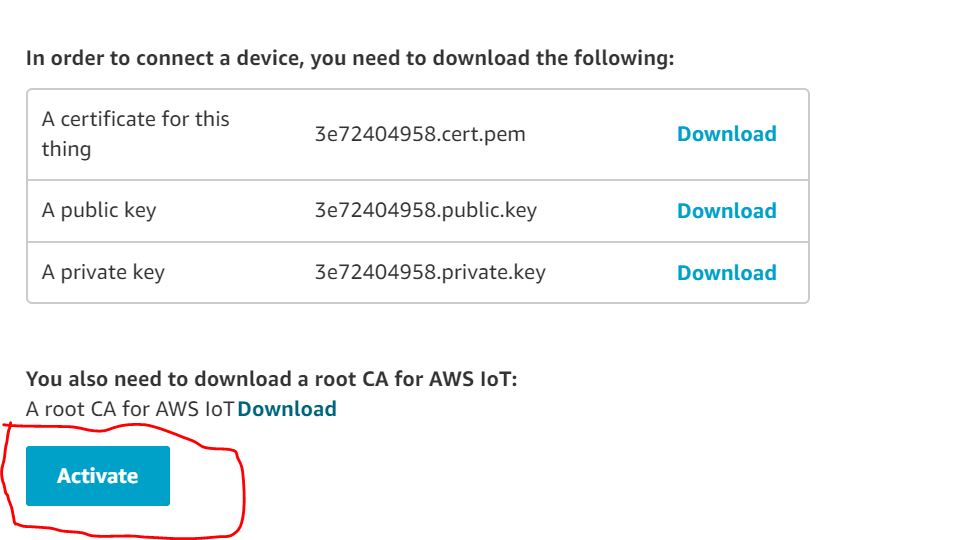


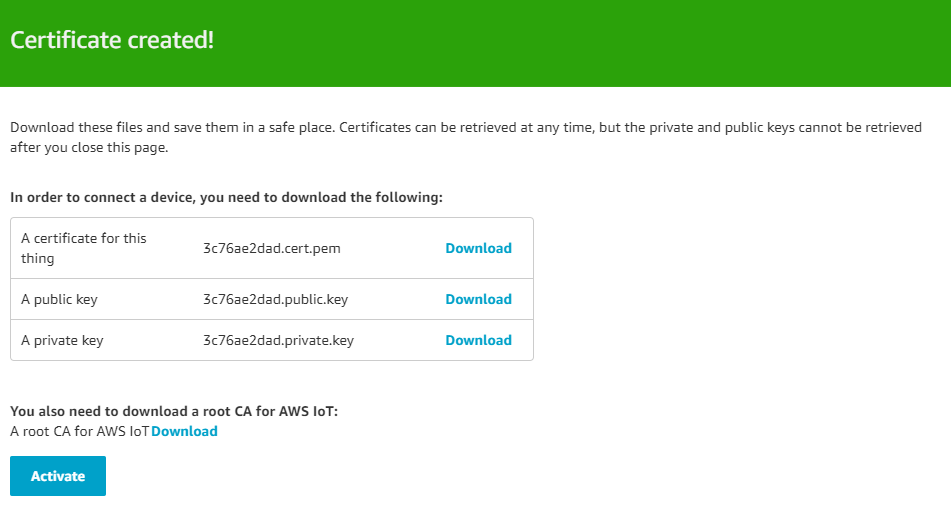
1. The above created in policy is available as JSON format, attached the same in the git hub project in the JSON folder and ensure the Allow statement has been selected while creating the AWS.



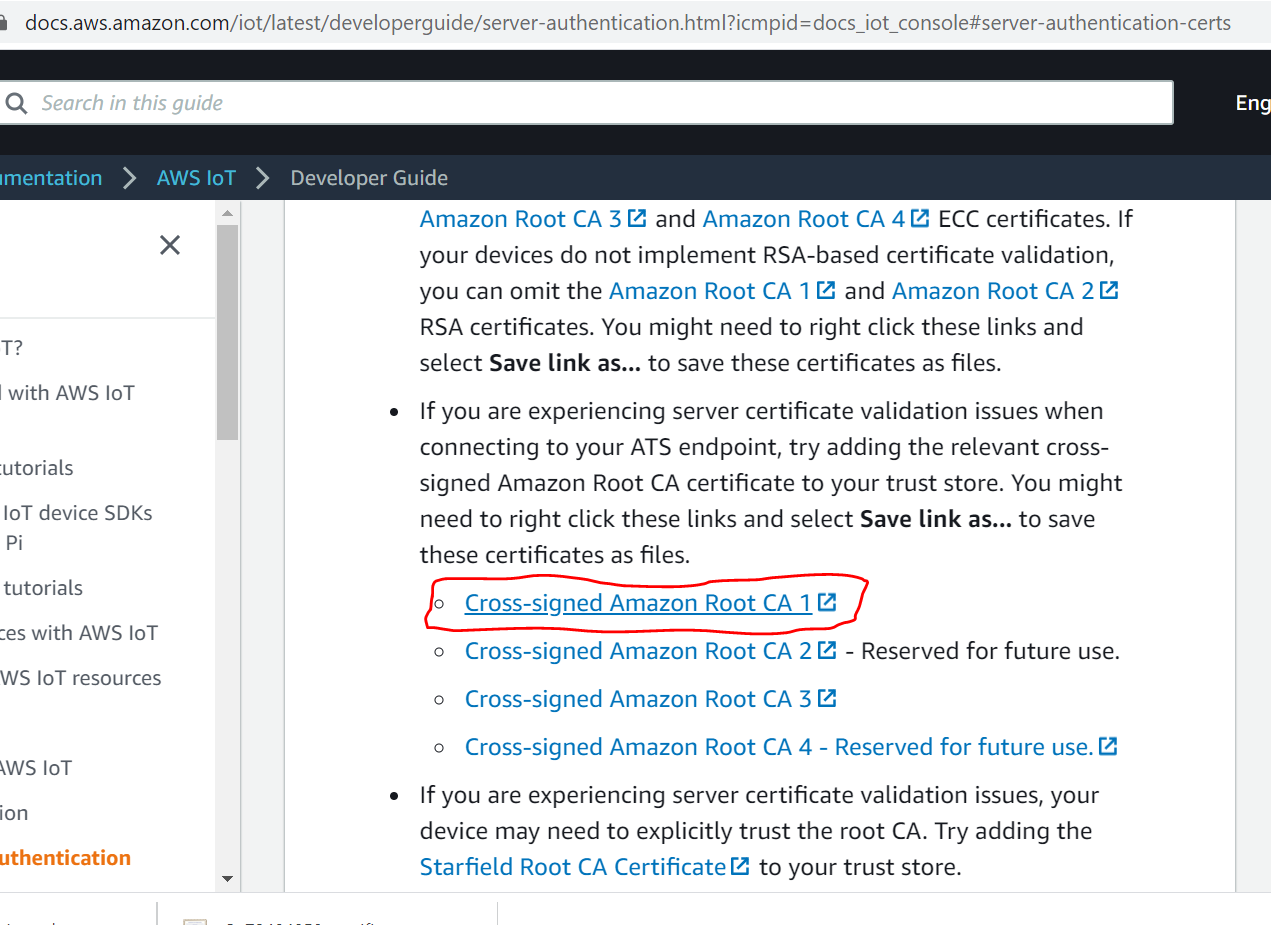
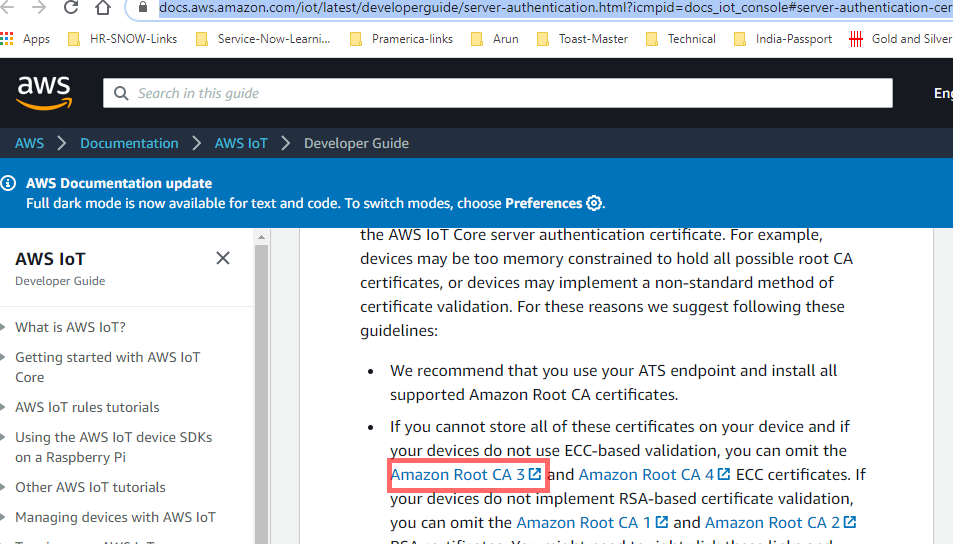


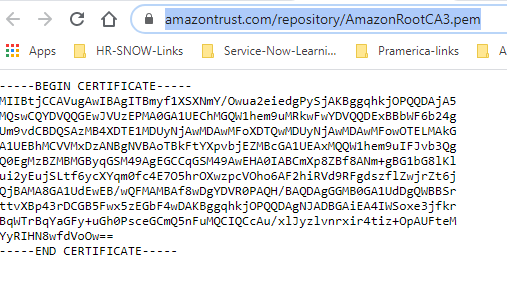
1. To publish the message from devices policy has been created in the above step and create a certificate as shown below to generate a certificate, public key, and private key using AWS IoT's certificate authority. Downloaded the public certificate and a key to attach in the MQTTfx client.



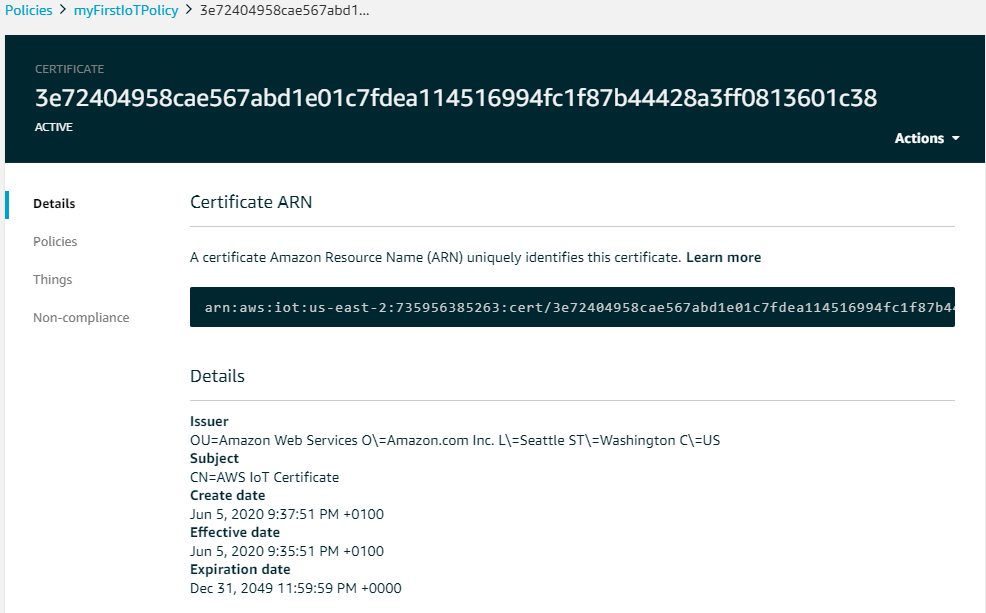


1. Download the root CA certificate from the link [[4]](https://docs.aws.amazon.com/iot/latest/developerguide/server-authentication.html?icmpid=docs_iot_console#server-authentication-certs) as shown in the below figure.

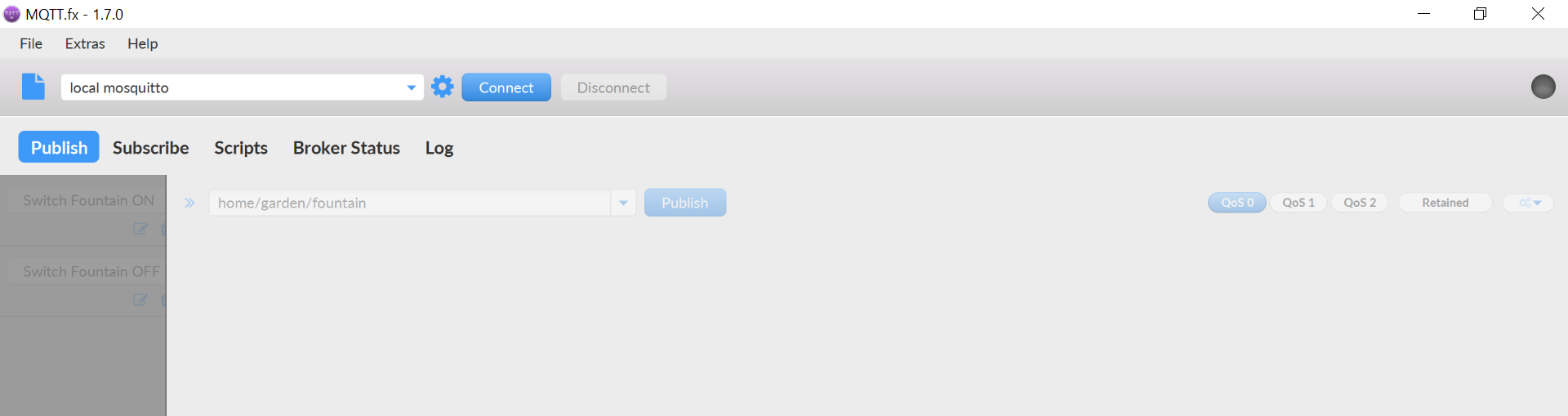
 

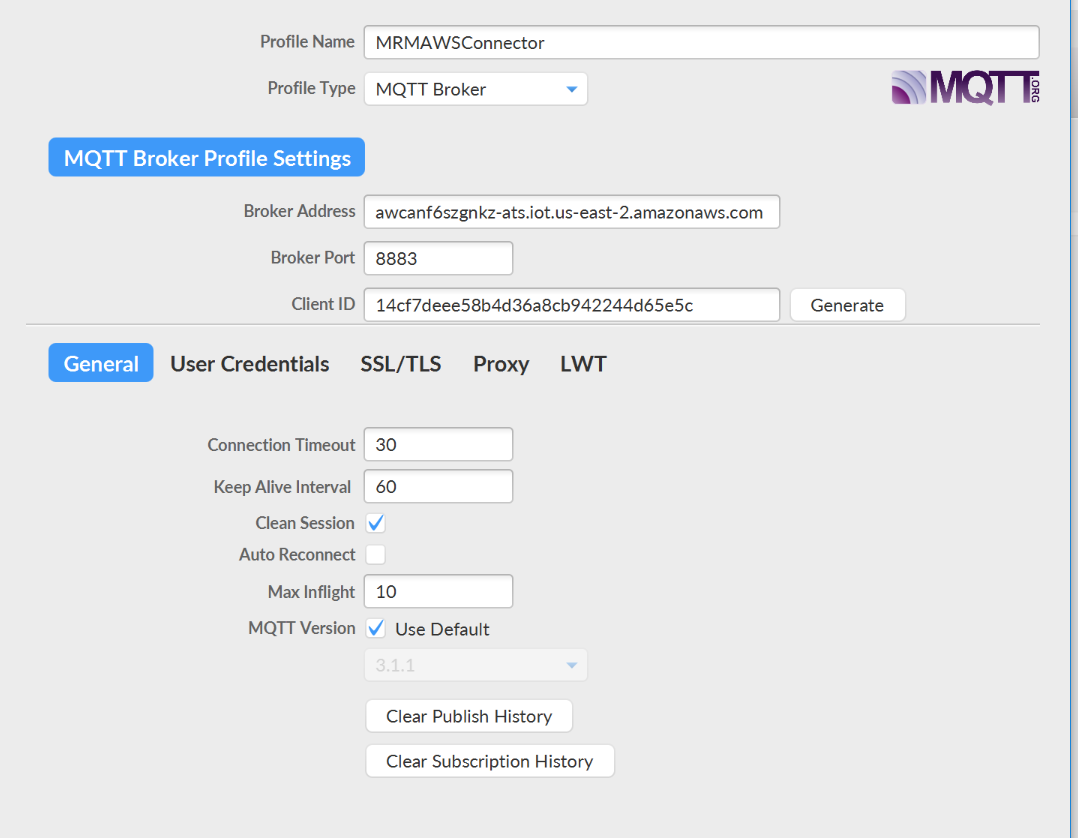


1. Once the certificate is created, attach to “myFirstIoTPolicy” IoT policy created in step 3.
2. All the certificates downloaded to the system and attached to the github in the certificates folder used for this project. The certificates created are shown in the below image.

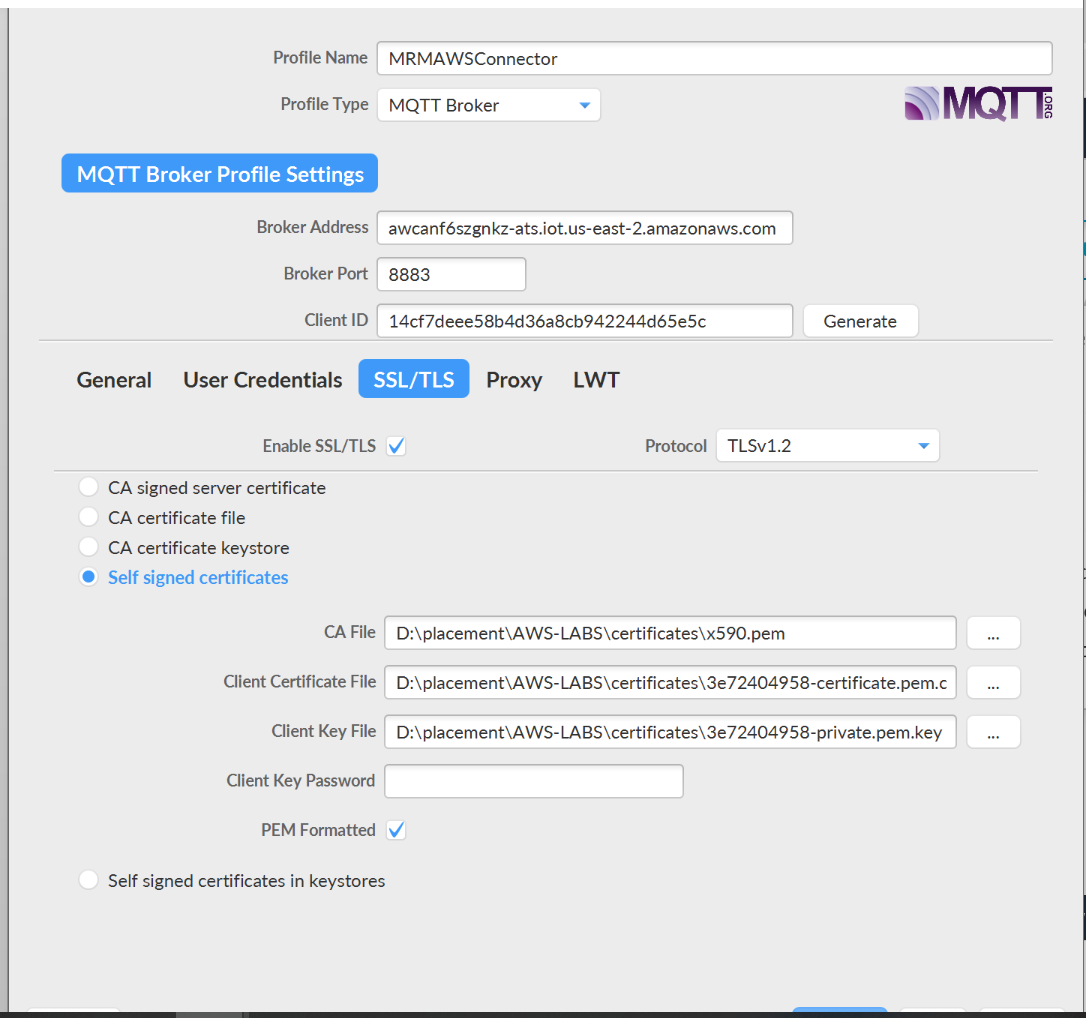


1. To connect AWS IoT with MQTTfx with the client by clicking the setting icon, create a connector as shown below.



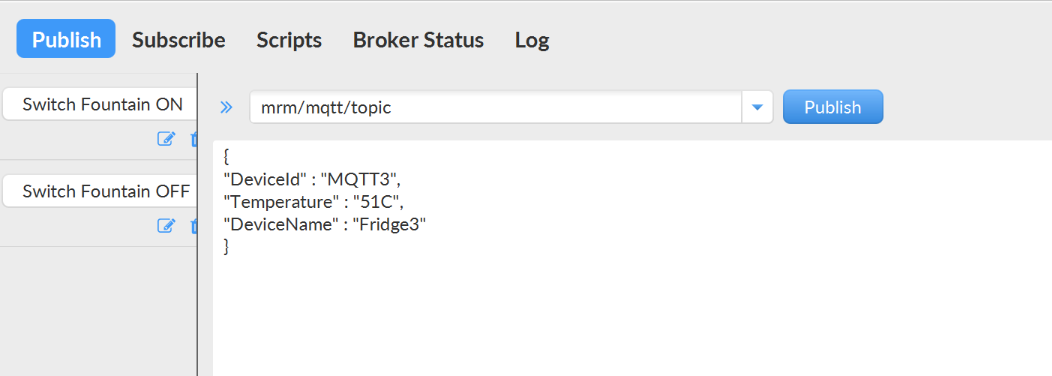


1. Attach the SSL/TLS certificate in the connector using the TLSv1.2 protocol as shown below.

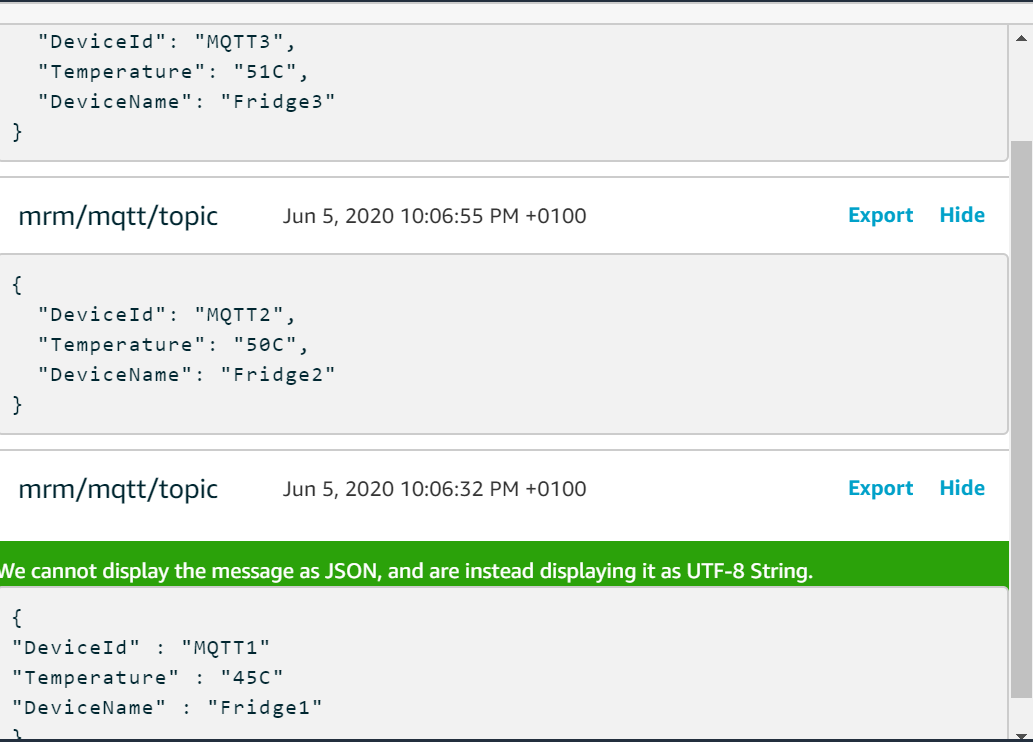


Note: TLSv1.2 and above is the supported TLS version from AWS, Any lower version of TLS will not work.

1. Publish the topic “mrm/mqtt/topic” by clicking the Publish button as shown in the below figure.



1. Tested the same, by subscribing to “mrm/mqtt/topic” in the MQTT client, the published messages will be displayed as shown in the below figure.



# Problems Faced

1. In this exercise, failed to attach the certificate to policy, so the published message in MQTTfx is not visible in the AWS MQTT client console.
2. Failed to attach the SSL information “self-signed certificates” as shown in step 11, so the connection could not

# Summary

The purpose of the above exercise is to understand the basic mechanism of MQTT publish and subscribe to the topic. By completing the above exercise will help to understand how to publish the message from the devices in the Industrial IoT world. Later this concept can be used to publish the message virtual MQTT client or simulated MQTT client or real embedded devices.

# References

[1] MQTT FX client, <https://mqttfx.jensd.de/index.php/download>, 02-June-2020

[2] AWS IoT Core, <https://aws.amazon.com/iot-core/>, 01-June-2020.

[3] AWS IoT Policies, <https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html?icmpid=docs_iot_console>, 06-June-2020.