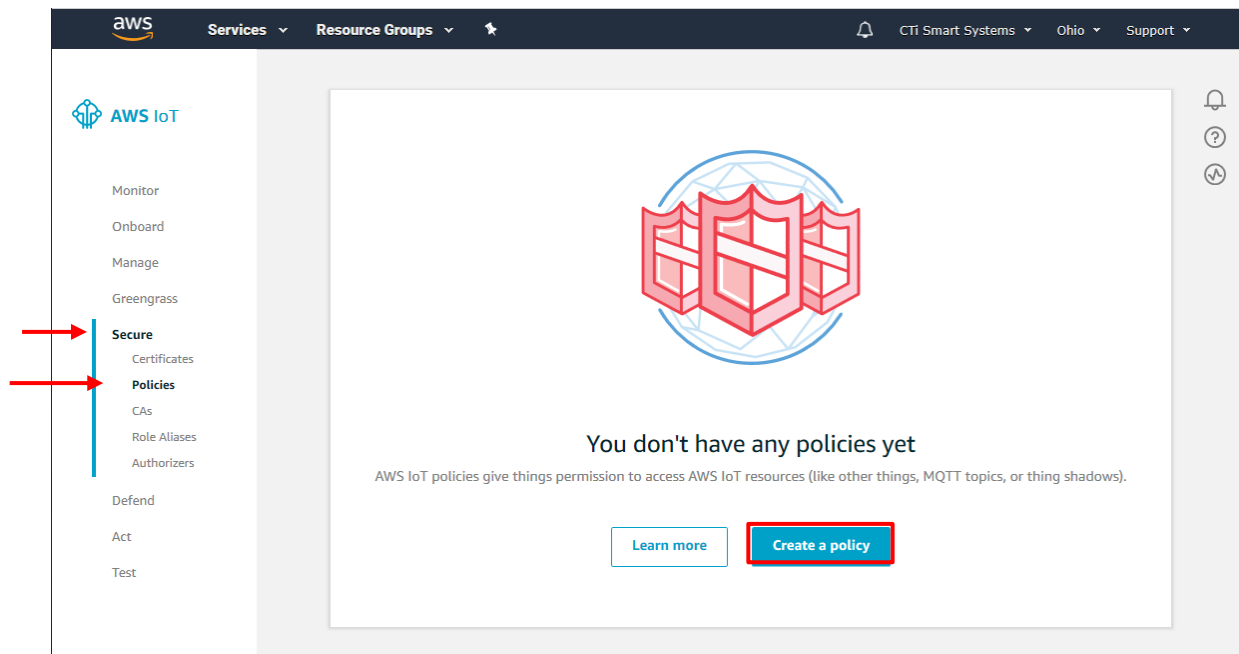


## AWS MQTT/TLS Sample

The AWS MQTT/TLS sample connects an AWS MQTT broker securely. The code of this sample is attached with this document at the end of this section.

To connect your nRF9160 to an AWS broker you need to have an AWS account. If you don't have an AWS account already go to [AWS IoT Console](#) and create one, then follow the direction below to setup the account.

- 1- After creating an AWS account, the first thing you need to do is creating a policy, so select **Secure>>Policies>> Create a Policy**.



Creating a policy

---

2- Add the policy statements you need, and if you are not sure what policies are needed for your device, you can use the one in the figure.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name

Policy1

Add statements

Policy statements define the types of actions that can be performed by a resource. Advanced mode

Action	Resource ARN	Effect
iot:*	*	<input checked="" type="checkbox"/> Allow <input type="checkbox"/> Deny <span>Remove</span>

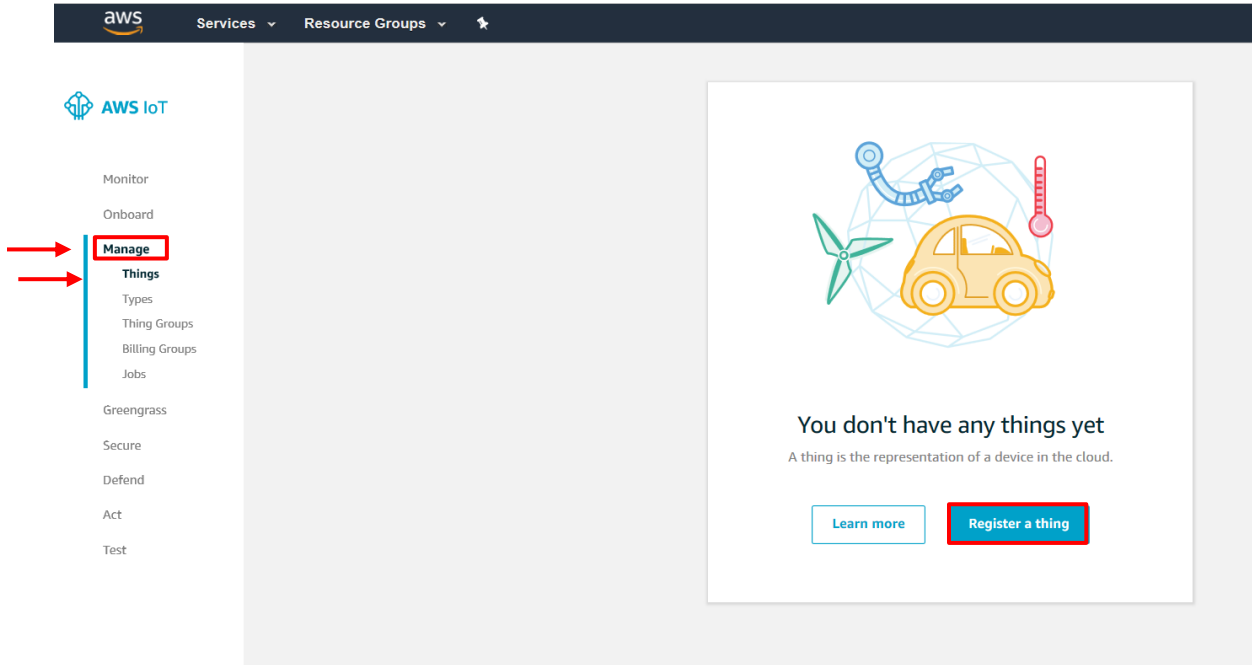
Add statement

Create

create a policy

The \* in figure 5.53 means that you are using **all** available resources.

- 
- 
- 2- After creating a policy, you need to create a thing that matches your client id. Select **Manage>>Things>>Register a thing**, then choose **Create a single thing**.



AWS IoT

## Creating AWS IoT things

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. [Learn more](#).

### Register a single AWS IoT thing

Create a thing in your registry

Create a single thing

### Bulk register many AWS IoT things

Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

Create many things

## Creating AWS IoT Thing

- 
- 3- Add your device to the thing registry by writing your client id as a thing name.

CREATE A THING

## Add your device to the thing registry

STEP 1/3

This step creates an entry in the thing registry and a thing shadow for your device.

Name

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected » [Create a type](#)

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups / [Create group](#) [Change](#)

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key	Value	
<input type="text" value="Provide an attribute key, e.g. Manufacturer"/>	<input type="text" value="Provide an attribute value, e.g. Acme-Corporation"/>	<a href="#">Clear</a>
<a href="#">Add another</a>		

Show thing shadow ▾

Cancel [Back](#) [Next](#)

Creating AWS IoT Thing

#### 4- You need to create a certificate for your thing.

CREATE A THING STEP 2/3

### Add a certificate for your thing

A certificate is used to authenticate your device's connection to AWS IoT.

**One-click certificate creation (recommended)**  
This will generate a certificate, public key, and private key using AWS IoT's certificate authority. [Create certificate](#)

---

**Create with CSR**  
Upload your own certificate signing request (CSR) based on a private key you own. [Create with CSR](#)

---

**Use my certificate**  
Register your CA certificate and use your own certificates for one or many devices. [Get started](#)

---

**Skip certificate and create thing**  
You will need to add a certificate to your thing later before your device can connect to AWS IoT. [Create thing without certificate](#)

#### Adding a certificate

#### 5- Download and Activate the certificate and the keys.

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	51aedab585.cert.pem	<a href="#">Download</a>
A public key	51aedab585.public.key	<a href="#">Download</a>
A private key	51aedab585.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:  
A root CA for AWS IoT [Download](#)

[Activate](#)

Cancel Done [Attach a policy](#)

#### Certificates Download

---

6- Before attaching policy, you need to download an **AWS root CA**, Click Download, and click on one of Amazon root certificates then copy the certificate and save it where you have the keys and the client certificate. We used **Amazon Root CA 1**.

**Server Authentication**

Server certificates allow your devices to verify that they're communicating with AWS IoT and not another server impersonating AWS IoT. Service certificates must be copied onto your device and referenced when devices connect to AWS IoT. For more information, see the [AWS IoT Device SDKs](#).

AWS IoT server certificates are signed by one of the following CA certificates:

**VeriSign Endpoints (legacy)**

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

**Amazon Trust Services Endpoints (preferred)**

- RSA 2048 bit key: [Amazon Root CA 1](#).
- RSA 4096 bit key: [Amazon Root CA 2](#) - Reserved for future use.
- ECC 256 bit key: [Amazon Root CA 3](#).
- ECC 384 bit key: [Amazon Root CA 4](#) - Reserved for future use.



Amazon Root CA

7- You can go back and click Attach Policy.

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

**In order to connect a device, you need to download the following:**

A certificate for this thing	51aedab585.cert.pem	<a href="#">Download</a>
A public key	51aedab585.public.key	<a href="#">Download</a>
A private key	51aedab585.private.key	<a href="#">Download</a>

**You also need to download a root CA for AWS IoT:**

A root CA for AWS IoT [Download](#)

[Deactivate](#)

Cancel [Done](#) [Attach a policy](#)

Attach a Policy

8- Select your policy and click **Register Thing**

CREATE A THING

## Add a policy for your thing

STEP 3/3

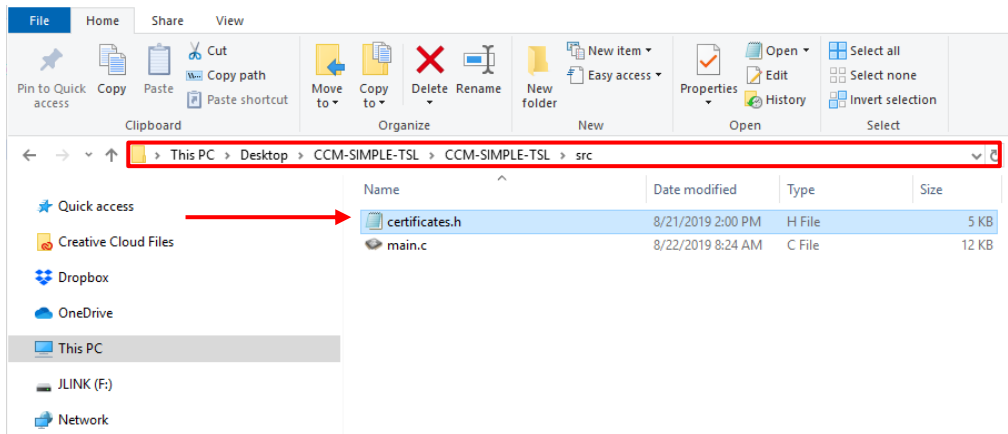
Select a policy to attach to this certificate:

- Policy1 View

1 policy selected Register Thing

Adding your policy

9- You need to add the certificates you downloaded to your device configuration. Go to **CC-MSIMPLE-TLS>>src>>certificates.h**



Certificates.h directory

10- Copy the client certificate, private key and the Amazon root CA to certificates.h as strings then save. You don't need to use the public key here.

```
certificates.h - Notepad
File Edit Format View Help
/*
 * Copyright (c) 2018 Nordic Semiconductor ASA
 *
 * SPDX-License-Identifier: BSD-5-Clause-Nordic
 */

#define CLIENT_ID "nRF9160-DK"

#define CA_CERTIFICATE \
"-----BEGIN CERTIFICATE-----\n" \
"MIIDQTCCAIegAwIBAgITBeyfz5w/3AoS4vB41kPw1jZbyjANBgkqhkiG9w0BAQsF\n" \
"ADASMQswCQYDVQQGEwJWUzZEPMAIGAMGQ1UECnMGQ1hew9uP9kufwYDVQDDEwBB\n" \
"l24gIw9vKBDQSAwPB4XDTI1MDUyNjA0PDAuPFoKDTMAMEsNzAwPDAuPFowOTE\n" \
"lWkGA1UEBHMPCVVMxOzANBgNVBAoTBkFtYXNjYXN0eS5kaWZlbnR1eS5kaWZl\n" \
"bnR1eS5kaWZlbnR1eS5kaWZlbnR1eS5kaWZlbnR1eS5kaWZlbnR1eS5kaWZl\n" \
"ca0HgFB0W7Y14h29j1o91ghYP10hAeVrAItht0gQ3p0sqTQ9ro8vo3bSPgWfz2M\n" \
"906I1Bc+6zf1tRn45W1w3te5dJgdY26k/oI2peVKVuRF4Fn9tBb6d9qczuSL/qr\n" \
"jFAGbHrQgLKw+a/sRxpUDgH3KXHVj4urtlp+UhrPCbu1HheB4eJucAuhmahRMe6\n" \
"V0uJwSH55Nz/BegwLX0tdHA114gk957EhM67c4cXBj3GKLHD+rcdqsq88p8kD111\n" \
"93FcRm/6pKCyZ1K+1A4b9v7LkIbxcceVDF34GfID5yHI9Y/QCB/IIDEgEw+OyQ\n" \
"jgSub3rIggBCAwEAAaHCEAwDwYDVR0TAQH/BALAwEB/zAOBgNVHQ8BAF8EBAMC\n" \
"AYYwAQYDVRR0BBYEFIQVzIUW7LwP13QuCFmcx7IQTgoIMAGCSqGSIb3QQEBCwU\n" \
"AAIBAQCYBjdaQZCHGvV2U5ggN1PDruYou6r41KS1p0B/G/wk3JuaByKX9rbxenDE\n" \
"USPHCCjJwCXP16T531HTFIU3+U6adT+CC2q3eHZERxh1bI1BjJt/wsv0tadQ1wJ\n" \
"lwgG563pYaAcbvXy8P4y7Vu33Pq00HweE6V/Uq2V8v1T09GLXFvKw13bYKBU90\n" \
"v/rUq3VtPVTBQcPHR8j+dkPSHCa2XV4cdFyQzR1b1d2wg3c3wApzyKZFo6IQ6XJ\n" \
"5MsI+yMQ+HDXJ1oa1dXgJAKK642M4wrt9VBob2x3NDd22hulnoQdeXeGADbkpy\n" \
"nqKRfboQno2sG4q5WTP4685QuvG5\n" \
"-----END CERTIFICATE-----\n" \

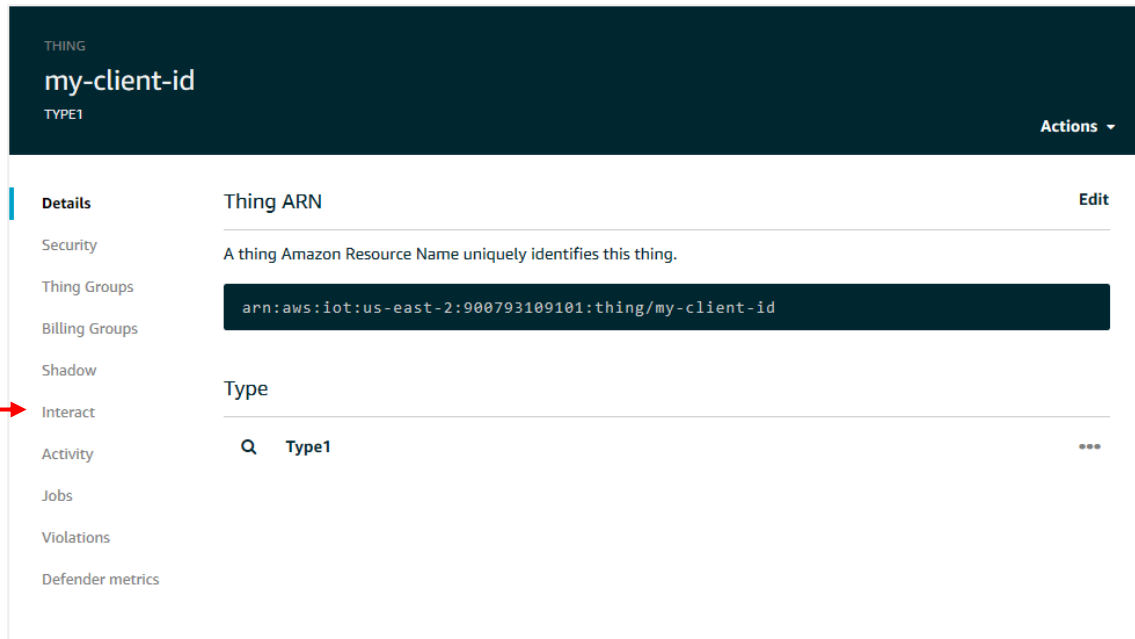
#define CLIENT_PUBLIC_CERTIFICATE \
"-----BEGIN CERTIFICATE-----\n" \
"MIIDNjCCAKgAwIBAgIVAPPh3JNS0rp1a5vd714a/A751EerPMAGCSqGSIb3QQE\n" \
"CuIMPEBx5z8jB8gNVBAwMQkFtYXNjYXN0eS5kaWZlbnR1eS5kaWZlbnR1eS5\n" \
"IE1uYy4gTD1TZWZhdGx1IFNUPVdhc2hpbmd9b24gQz1VUzAeFw0TAAPjIwODI\n" \
"lWkGA1UEBHMTEyYzZEPMAIGAMGQ1UEBHMCAwEAAaHCEAwDwYDVR0TAQH/BAL\n" \
"AWEB/zAOBgNVHQ8BAF8EBAMCAYYwAQYDVRR0BBYEFIQVzIUW7LwP13QuCFmcx7\n" \
"IGAggEIPMAGCSqGSIb3QQEBAQIAA4IBDwAggEKAoIBAQDSr1zY9n3IVDx/eS/\n" \
"TruPP6suhAzEcsvpfhL3vDGSCO22nx3PAb9aPC4/B18Pglz60P93y65AHj+acv\n" \
"yXKv1obE3+tb0wrbko891zbr1s5BFH7E+v235CLYNAn31HEyt1DY7rMe+Q9vC\n" \
"TwDccgcPxY1eL2o+c4ek8gVTUrtCw5y7oNe20719Lh3QYF+w5RQ02k2G1uDLQ\n" \
"FKRRlgeSet3zbY7kzxvG7FSKcT+4s2w4cxKkmydFMAnxjxtu05LEp1gPVoTX+fi\n" \
"-----END CERTIFICATE-----\n" \
```

Certificates.h file



---

11- Go back to your AWS account, click on the thing you created and select Interact. The thing shadow is the host broker that you will need to add to your configuration.



THING  
**my-client-id**  
TYPE1

Actions ▾

Details  
Security  
Thing Groups  
Billing Groups  
Shadow  
**Interact**  
Activity  
Jobs  
Violations  
Defender metrics

Thing ARN Edit

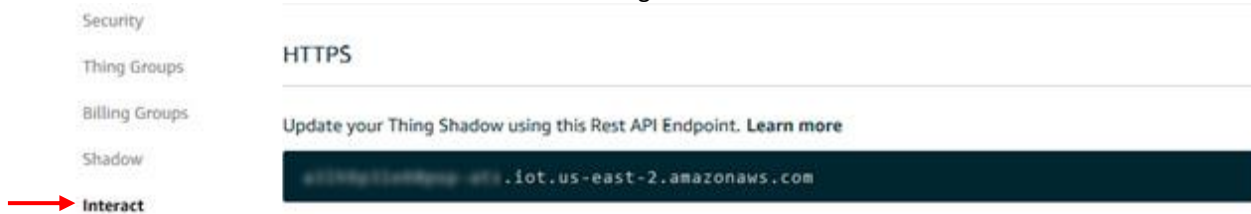
A thing Amazon Resource Name uniquely identifies this thing.

```
arn:aws:iot:us-east-2:900793109101:thing/my-client-id
```

Type

🔍 Type1 ...

Thing Details



Security  
Thing Groups  
Billing Groups  
Shadow  
**Interact**

HTTPS

Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

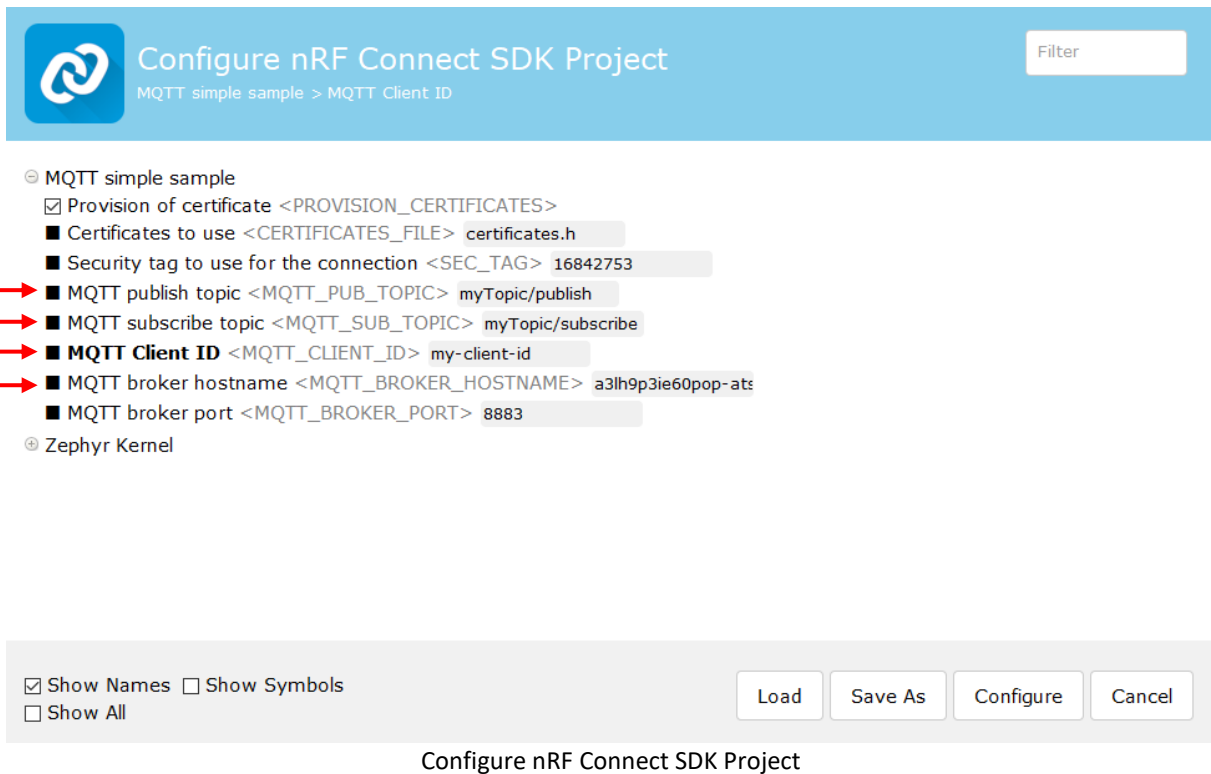
```
https://my-client-id-900793109101.iot.us-east-2.amazonaws.com
```

Host Broker

12- Open your **Segger embedded studio**, select file, **Open nRF connect SDK Project** and choose our **zip file**.

13- Select **Project>>Configure nRF Connect SDK Project>>menuconfig>>MQTT simple sample**.

14- Use your new Host Broker, then choose publish and subscribe topics, and configure.



15- Plug the nRF9160 DK to your computer and go to **Build > Build Solution**.

16- Go to **Target > Connect J-Link**, when its connected go back to **Target** and choose **Erase All** (Be sure to erase the board every time you need to reprogram it).

17- Program and run your program by clicking the green arrow.

18- To make sure that your nRF9160 is connected to your AWS account, open TeraTerm. It should look like figure 5.67.

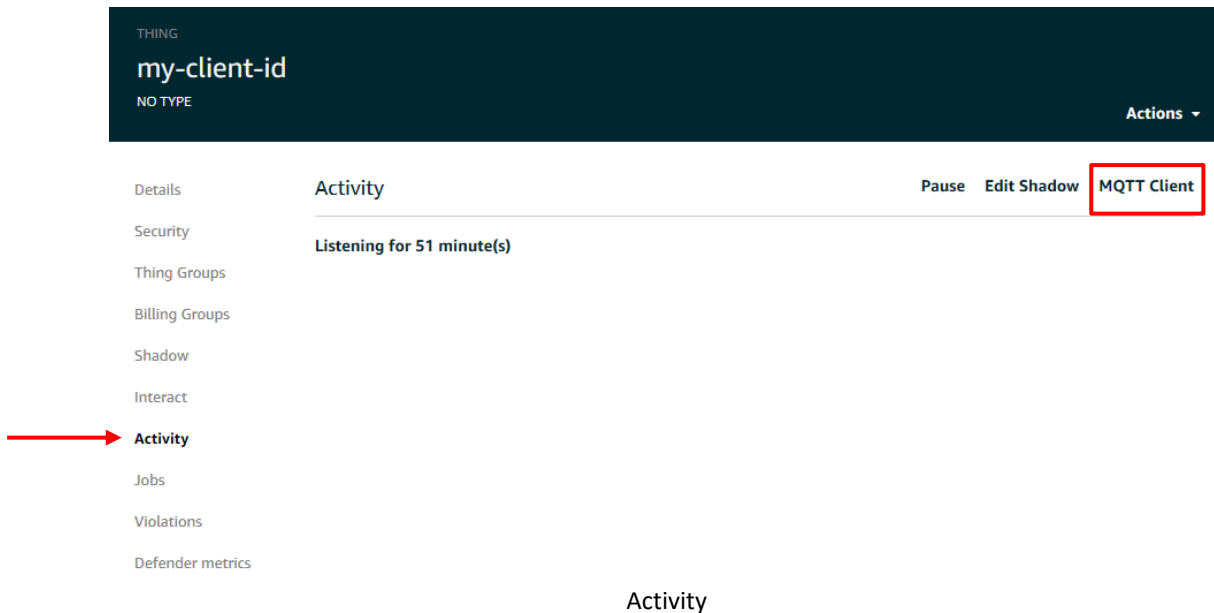
```

COM11 - Tera Term VT
File Edit Setup Control Window Help
The MQTT simple sample started
Deleting certs sec_tag: 16842753
nrf_inbuilt_key_delete(16842753, 0) => result=0
Deleting certs sec_tag: 16842753
nrf_inbuilt_key_delete(16842753, 1) => result=0
Deleting certs sec_tag: 16842753
nrf_inbuilt_key_delete(16842753, 2) => result=0
Deleting certs sec_tag: 16842753
nrf_inbuilt_key_delete(16842753, 3) => result=2
Deleting certs sec_tag: 16842753
nrf_inbuilt_key_delete(16842753, 4) => result=2
Write ca certs sec_tag: 16842753
Write private cert sec_tag: 16842753
Write public cert sec_tag: 16842753
LTE Link Connecting ...
LTE Link Connected!
IPv4 Address found 0x6441dd12
[mqtt_evt_handler:146] MQTT client connected!
Subscribing to: myTopic/subscribe len 17
Publish: Test
to topic: myTopic/publish len: 15
[mqtt_evt_handler:191] SUBACK packet id: 1234
[mqtt_evt_handler:181] PUBACK packet id: 4151

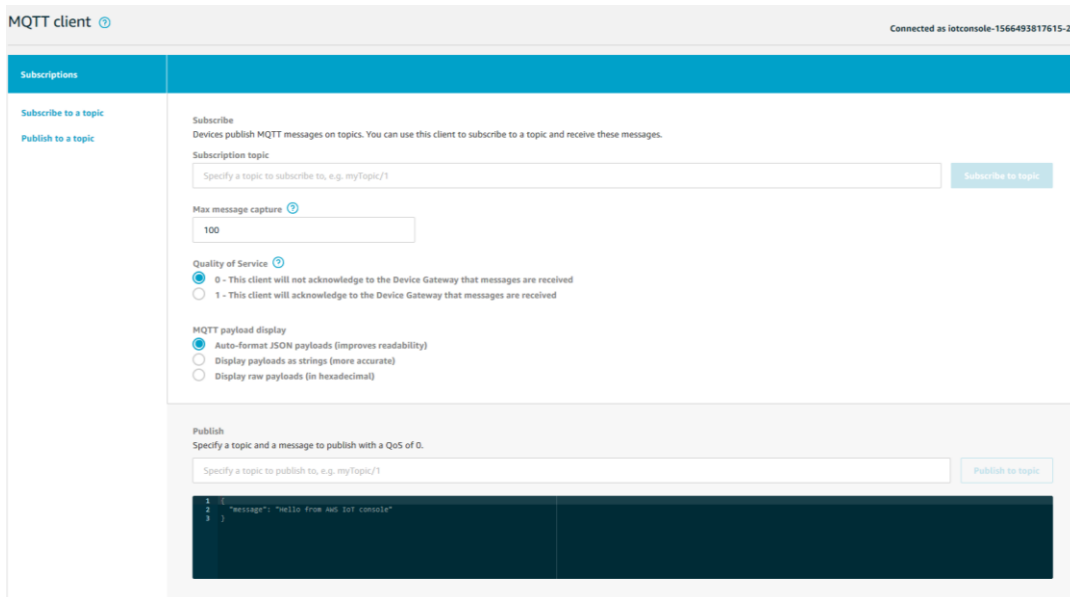
```

Tera Term

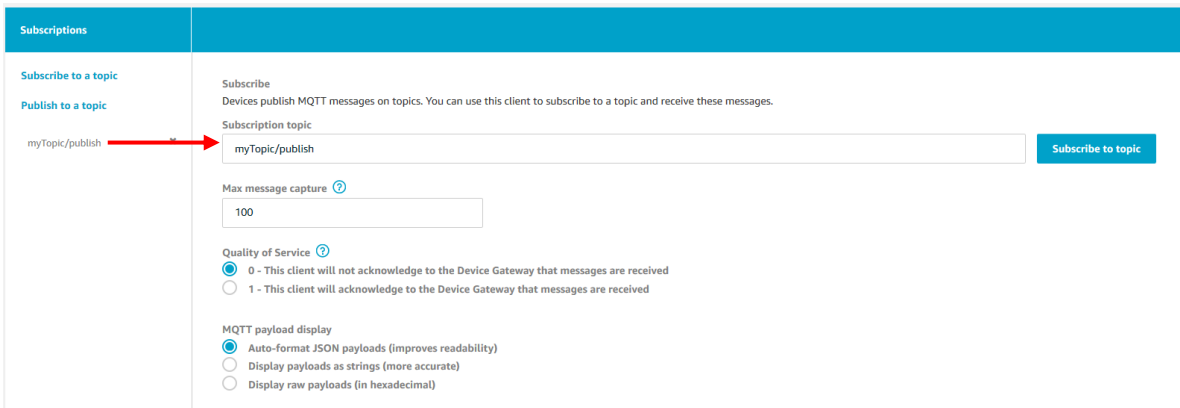
19- To test publishing and subscribing using AWS MQTT Client, go back to your **AWS account**, select **Activity>>MQTT Client**.



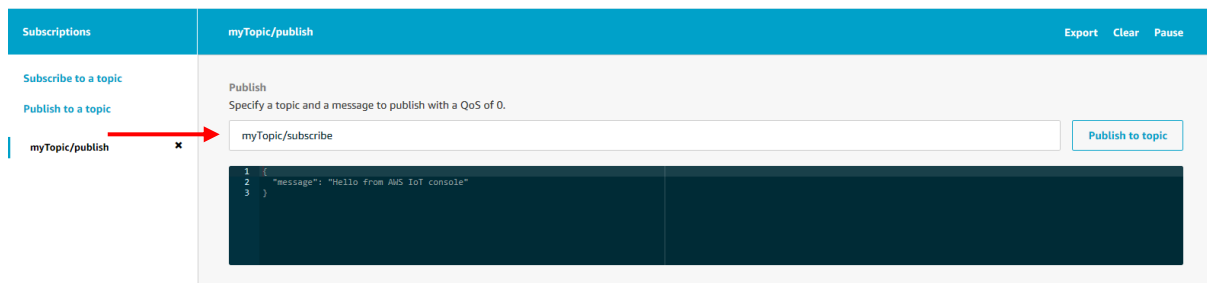
20- Choose a publish and subscribe topics based on your publish and subscribe topics in the **menuconfig (Segger Studio)**. **Remember** that you publish to a subscribe topic and you subscribe to a publish topic.



Subscribe and publish topics



Subscription Topic



Publish to Topic

---

21- To Publish a Test message press on Button 2 on your nRF9160 and you will see the message published in your AWS account.

myTopic/publish Export Clear Pause

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

myTopic/subscribe Publish to topic

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

myTopic/publish Aug 27, 2019 9:41:11 AM -0400 Export Hide

We cannot display the message as JSON, and are instead displaying it as UTF-8 String.

Test

Test message

[Download the code here](#)