BG95 Series AWS IoT Platform Access User Guide

LPWA Module Series

Rev. 1.0

Date: 2020-08-25

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236 Email: info@quectel.com

Or our local office. For more information, please visit: http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm or email to support@quectel.com.

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

DISCLAIMER

WHILE QUECTEL HAS MADE EFFORTS TO ENSURE THAT THE FUNCTIONS AND FEATURES UNDER DEVELOPMENT ARE FREE FROM ERRORS, IT IS POSSIBLE THAT THESE FUNCTIONS AND FEATURES COULD CONTAIN ERRORS, INACCURACIES AND OMISSIONS. UNLESS OTHERWISE PROVIDED BY VALID AGREEMENT, QUECTEL MAKES NO WARRANTIES OF ANY KIND, IMPLIED OR EXPRESS, WITH RESPECT TO THE USE OF FEATURES AND FUNCTIONS UNDER DEVELOPMENT. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUECTEL EXCLUDES ALL LIABILITY FOR ANY LOSS OR DAMAGE SUFFERED IN CONNECTION WITH THE USE OF THE FUNCTIONS AND FEATURES UNDER DEVELOPMENT, REGARDLESS OF WHETHER SUCH LOSS OR DAMAGE MAY HAVE BEEN FORESEEABLE.

COPYRIGHT

About the Document

Revision History

Version	Date	Author	Description
1.0	2020-08-25	Jaryoung LI	Initial

Contents About the Document 2

Table Index

Figure Index

1 Introduction

This document provides users with AWS IoT Cloud platform access method, including how to connect Quectel module to AWS IoT Cloud platform with MQTTS and the related AT command involved in the AWS IoT platform access process.

1.1. Brief Introduction on AWS IoT

AWS IoT provides secure, bi-directional communication between Internet-connected devices such as sensors, actuators, embedded micro-controllers, or smart appliances and the AWS Cloud. This enables to collect telemetry data from multiple devices, and store and analyze the data.

AWS IoT Cloud platform supports TLS dual authentication for client certificates, in which MQTT can act as a message broker which provides a secure mechanism for devices and AWS IoT applications to publish and receive messages from each other. After importing certificates into Quectel module, the module can access to AWS IoT Cloud platform through MQTTS.

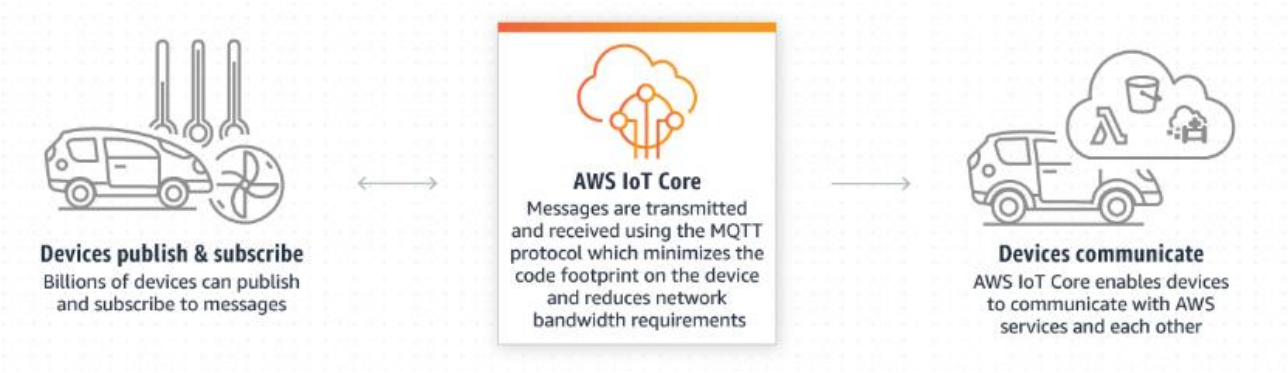


Figure 1: Communication between AWS IOT and the Device

2 AWS IoT Platform Access

AWS IoT platform supports TLS client and server certificates authentication, with the MQTT protocol as a message broker. The certificates imported by the module can be used to connect the AWS IoT platform with MQTTS.

NOTE

Before using AWS IoT services, an AWS account must be created. Please refer to the AWS official link https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html#aws-registration for details on how to create an AWS account.

2.1. Register Device and Get Certificates

Sign in to the AWS IoT platform and register a device in the registry. Certificates will be created in the process of device registration.

NOTE

The certificates created in the process of device registration to be imported into the module later should downloaded to the local computer.

2.1.1. Register Device

1. Navigate to the AWS IoT Console at https://console.aws.amazon.com/iot/home, in the navigation pane, choose Manage, and then choose Things.

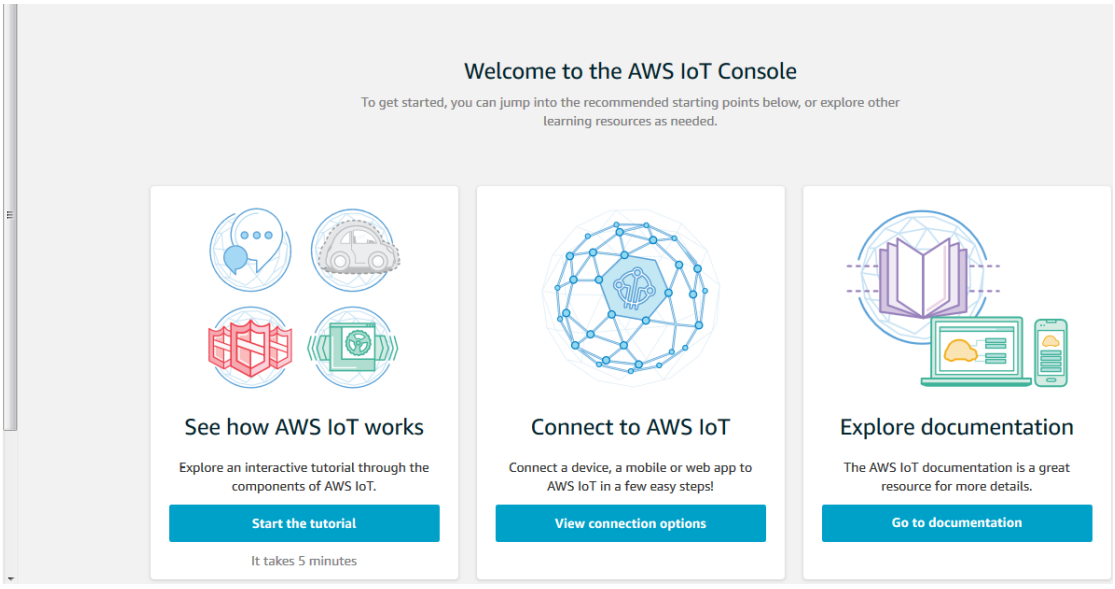
1. 

Figure 2: AWS IoT Console

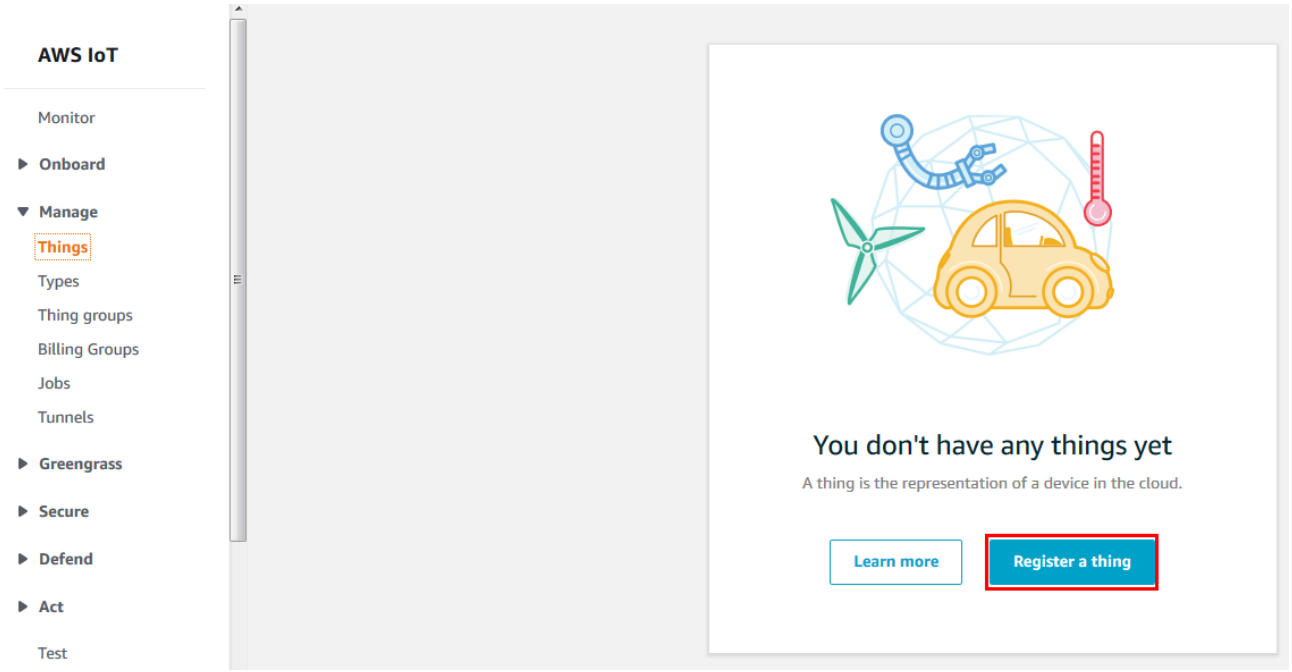


Figure 3: Register a Thing

2. In the Creating AWS IoT things page, click "**Create a single thing**".

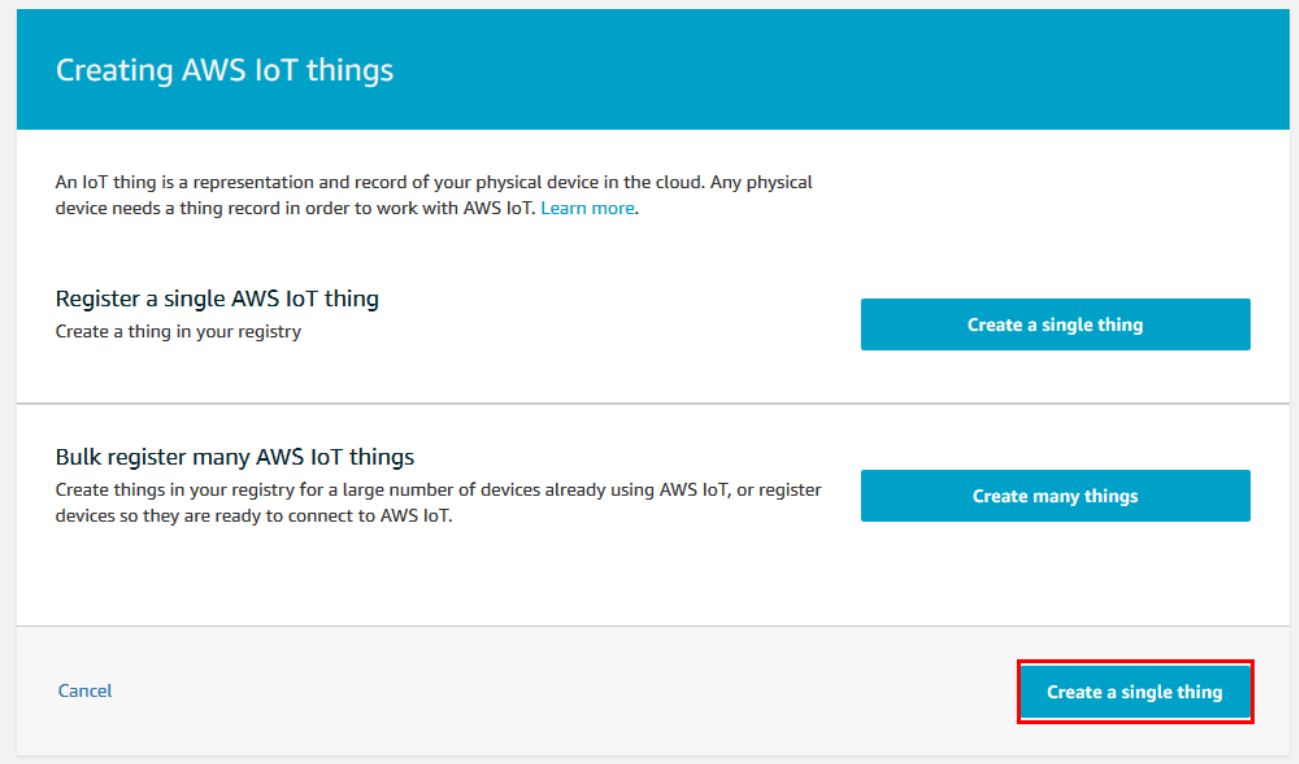


Figure 4: Create a Single Thing

3. Then add your device to the thing registry according to the provided steps. Take the device name MyIoTDevice as an example:

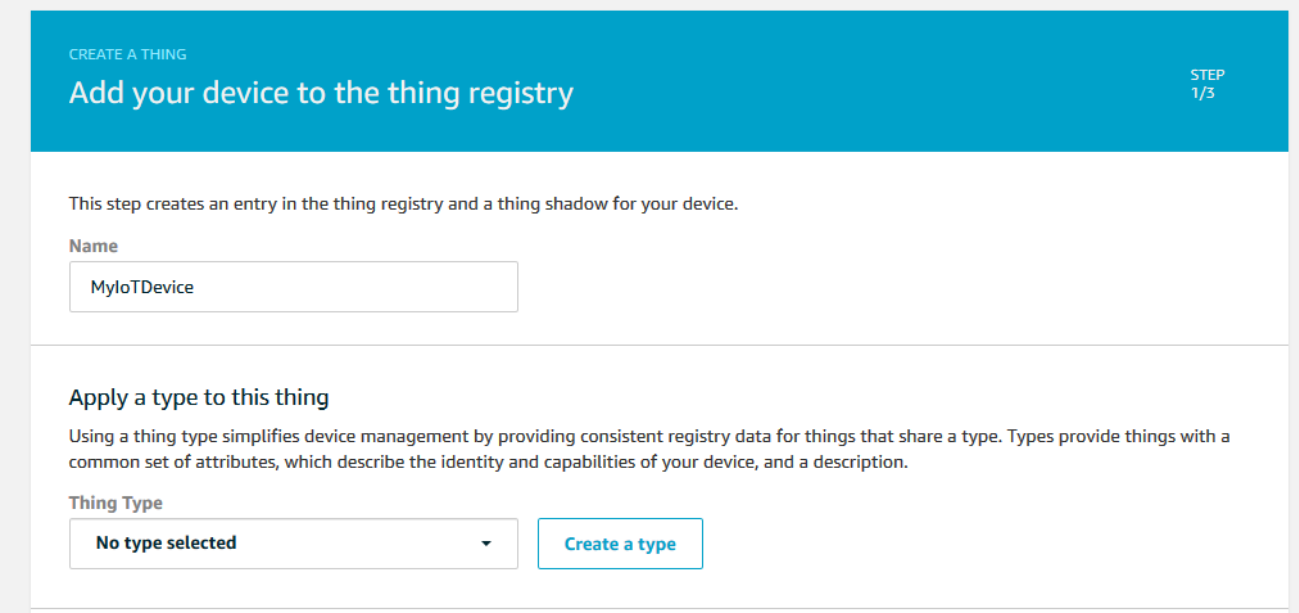


Figure 5: Add Device to the Thing Registry

2.1.2. Get Certificates

X.509 certificates protects the connection between device and AWS IoT platform. The certificates have to be activated before using.

1. On the Add a certificate for your thing page, under One-click certificate creation, choose Create certificate.

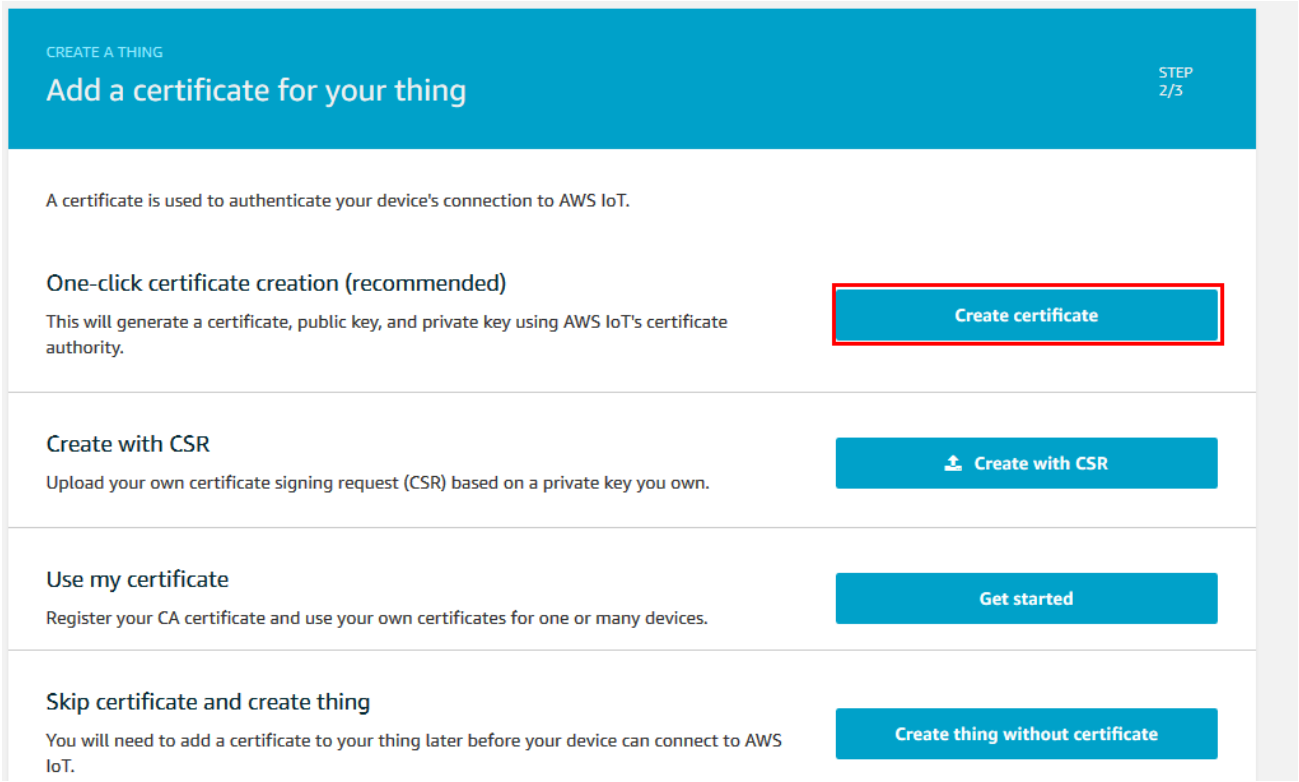


Figure 6: Add Certificate for Thing

2. Download certificates, keys and root CA and save them in your PC. If your device supports it, you should select the RSA 2048 bit key: Amazon Root CA 1 These are cross-signed by Starfield. https://docs.aws.amazon.com/iot/latest/developerguide/server-authentication.html.

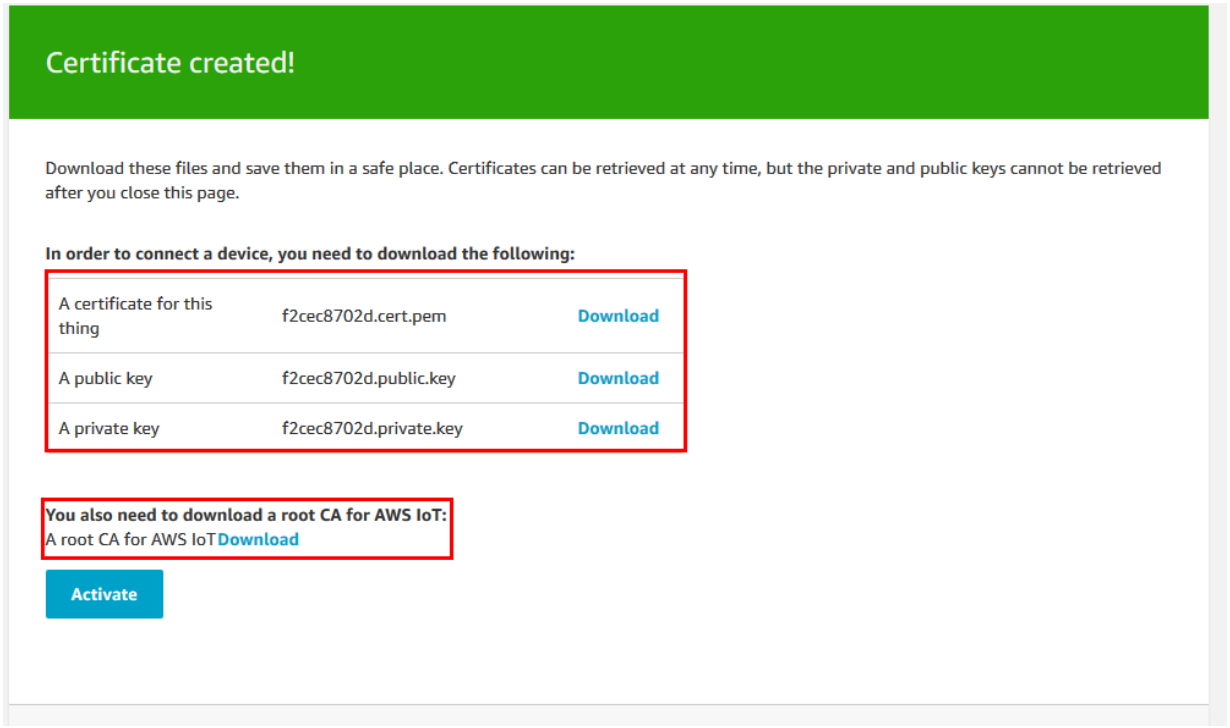


Figure 7: Download Certificates

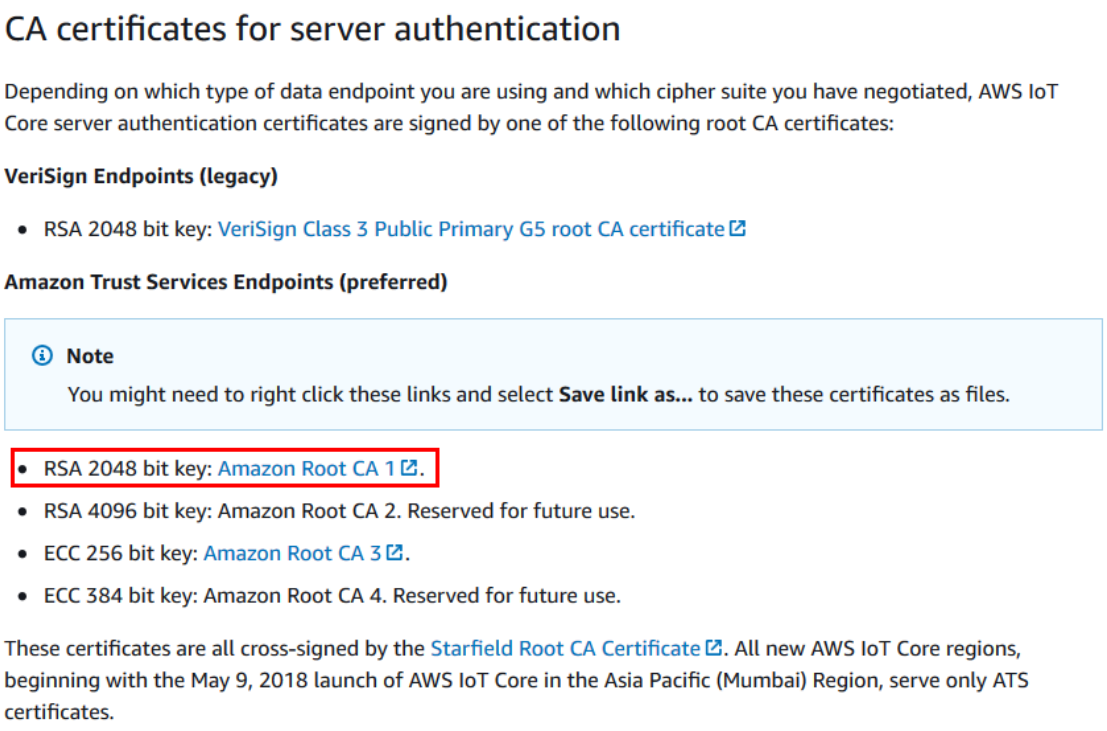


Figure 8: Select Root CA

2.2. Create Policies and Attach

1. Create a policy.

   In the AWS IoT console page, select "**Secure**" in the left navigation bar, click "**Policies**" and "**Create**" to create a policy. In the "Create a policy" page, input policy name, take policy name MyIoTDevicePolicy as an example:

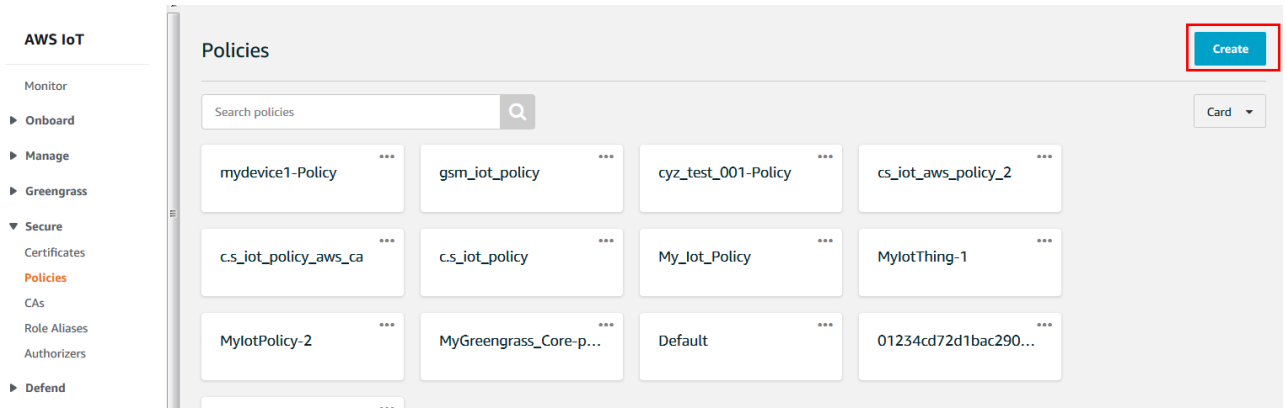


Figure 9: Policies

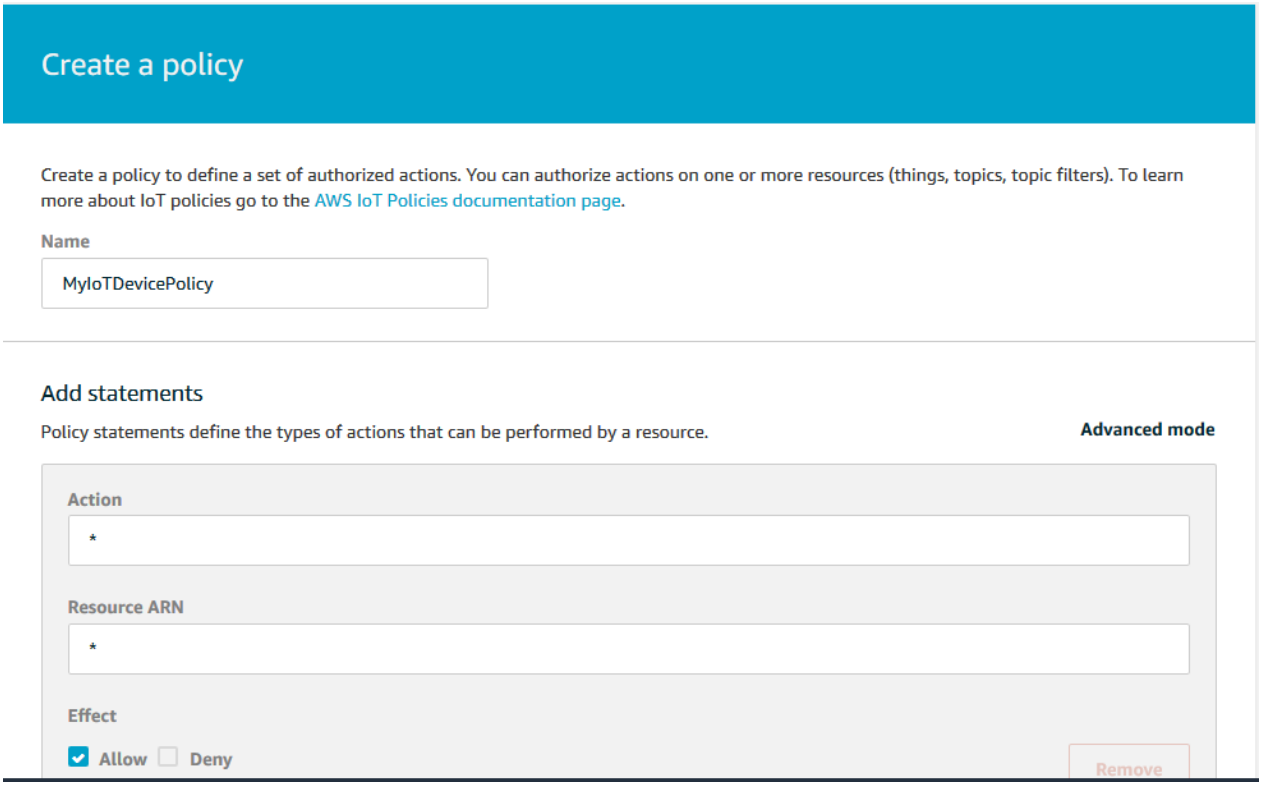


Figure 10: Create a Policy

NOTE

This policy grants unrestricted access for all iot operations, and is to be used only in a development environment. For non-dev environments, all devices in your fleet must have credentials with privileges that authorize intended actions only, which include (but not limited to) AWS IoT MQTT actions such as publishing messages or subscribing to topics with specific scope and context. The specific permission policies can vary for your use cases. Identify the permission policies that best meet your business and security requirements.
For sample policies, refer to
https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html.
Also refer to
https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html.

2. Attach the policy to a certificate.

   In the AWS IoT console page, select "**Secure**" in the left navigation bar, click "**Certificates**", choose a certificate:

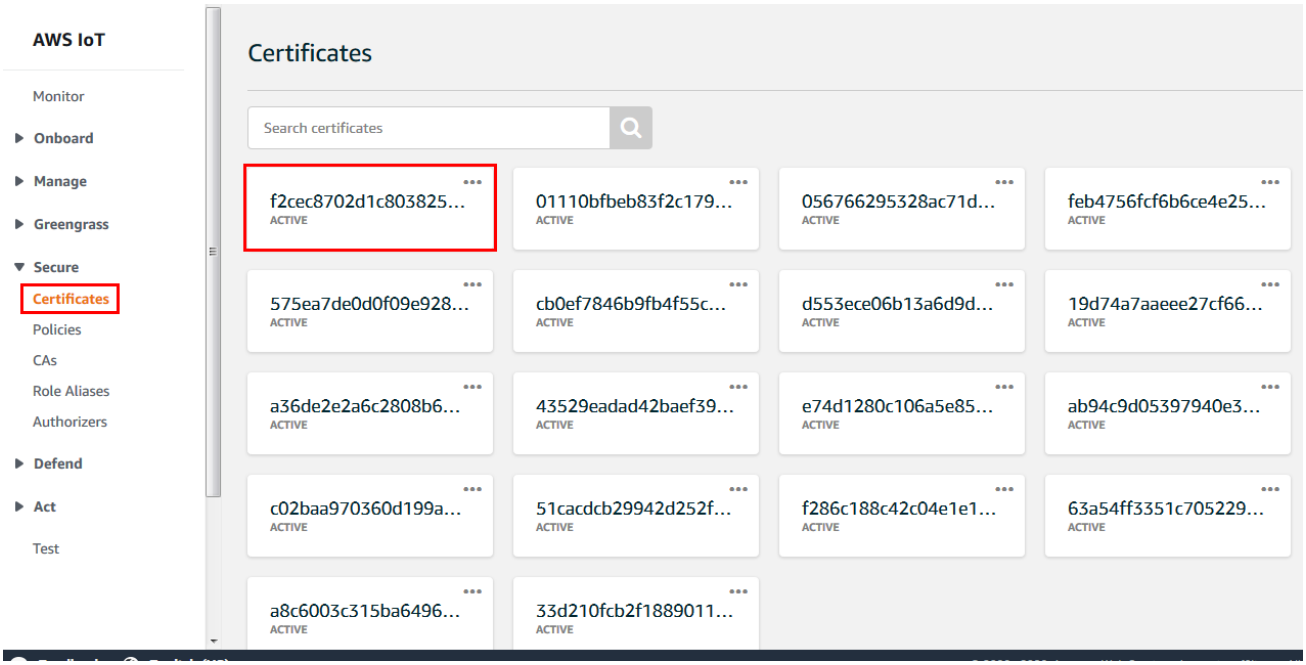


Figure 11: Select Certificate

Click "**…**" to open the drop-down menu and click "**Attach policy**":

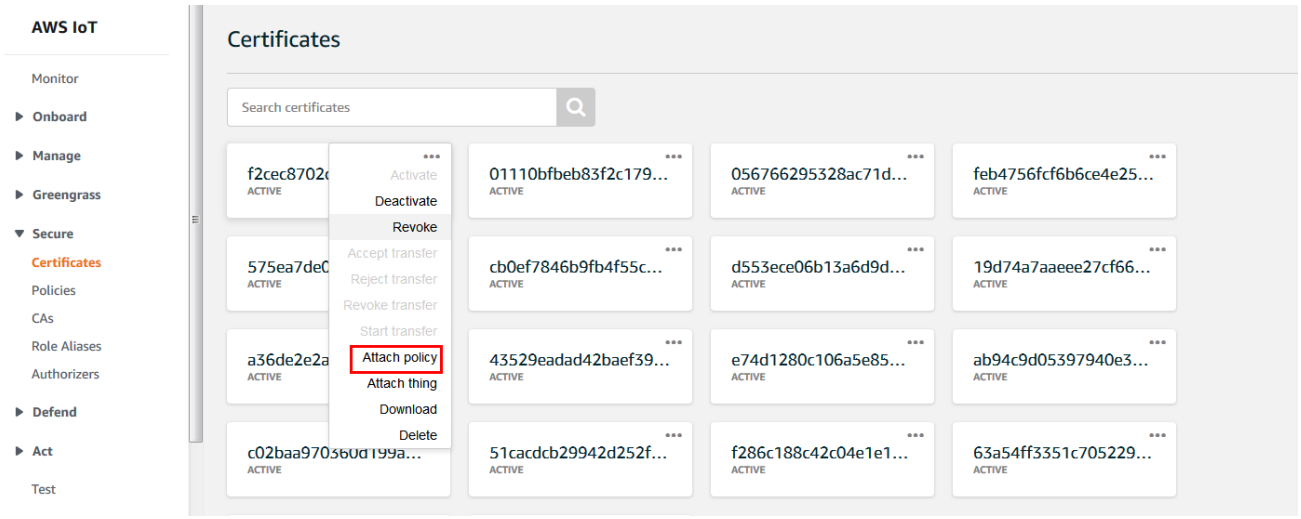


Figure 12: Attach Policy

Choose the policy you have created previously and click "**Attach**":

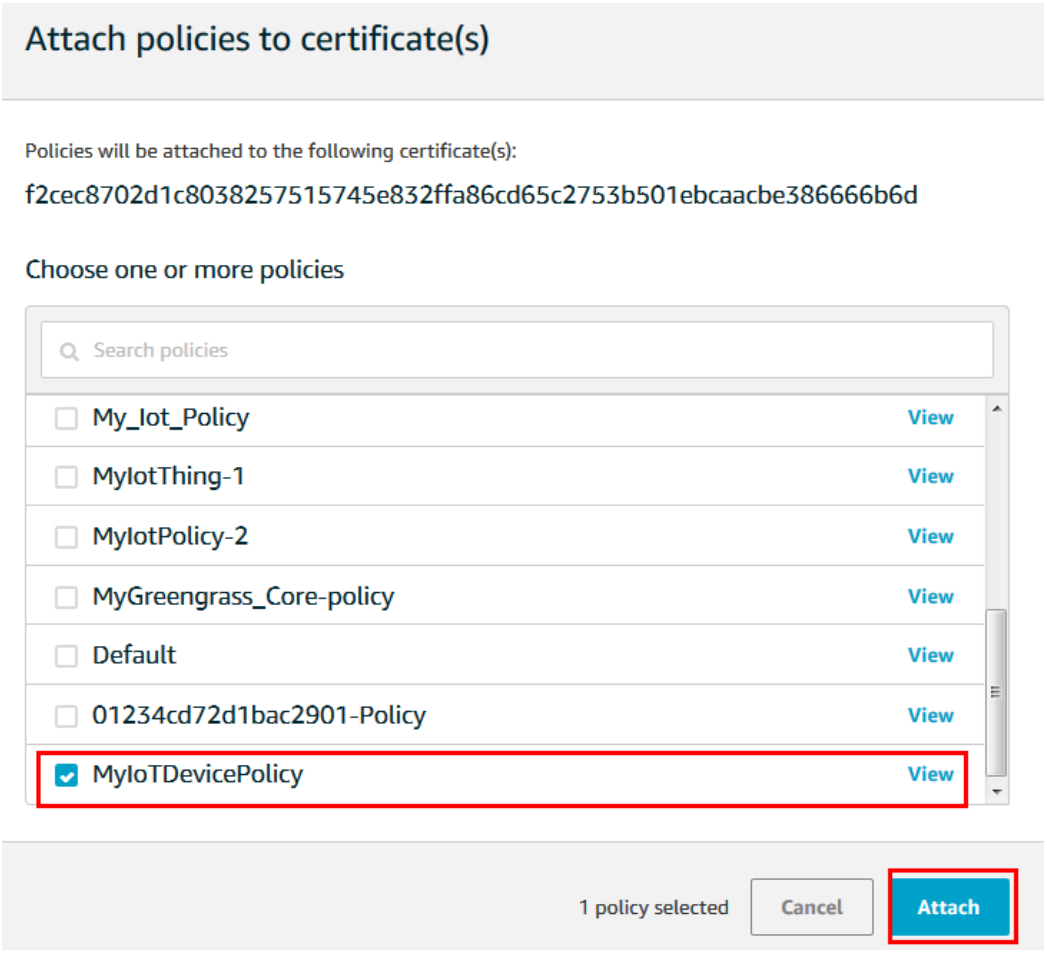


Figure 13: Attach Policy to the Certificate

3. Attach the certificate to a thing.

   Click "**…**" of the specified certificate to open the drop-down menu and click "**Attach thing**":

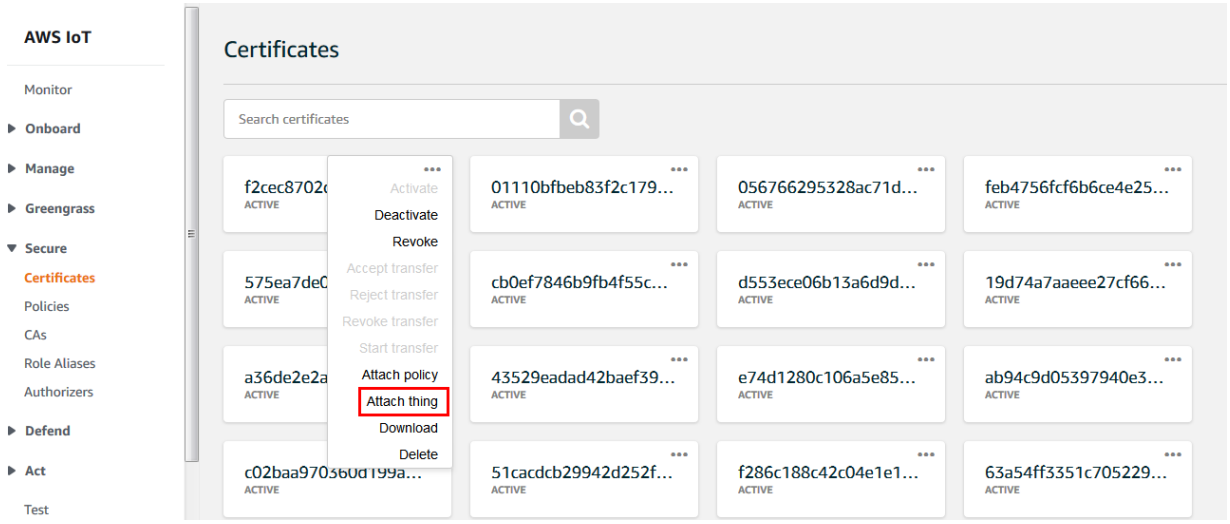


Figure 14: Attach Thing

Choose the thing you have created previously and click "**Attach**":

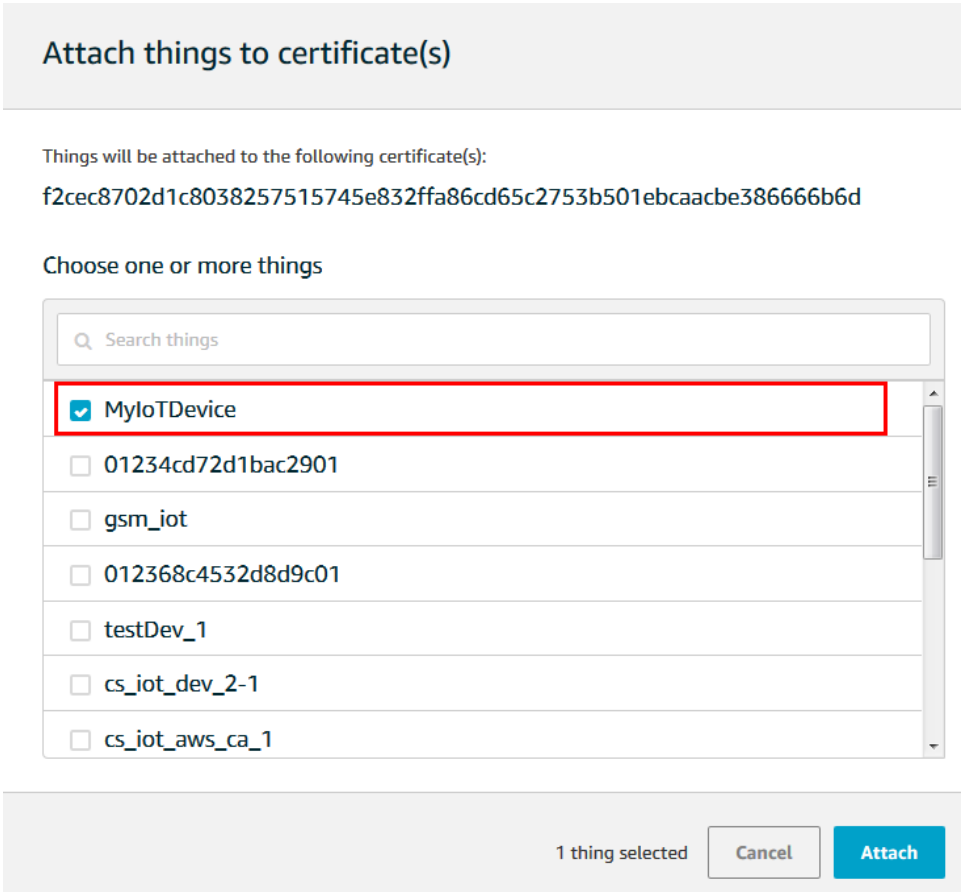


Figure 15: Attach Thing to the Certificate

4. Verification.

   To verify whether the policy and thing are attached successfully, select the certificate to list all attached policies and things.

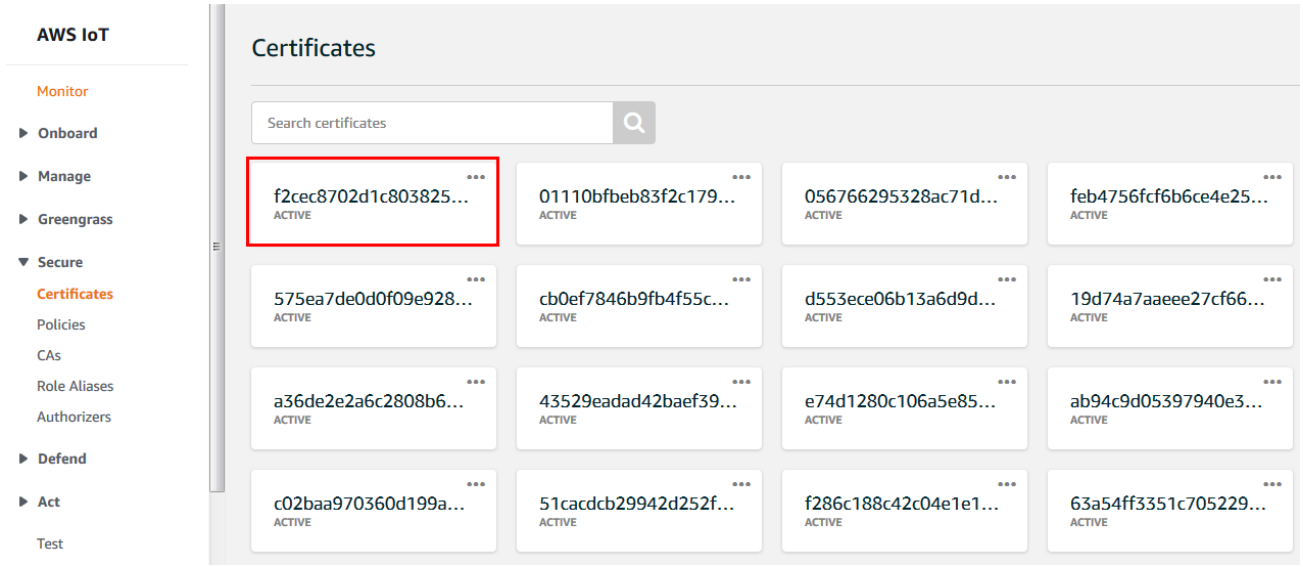


Figure 16: Select the Certificate

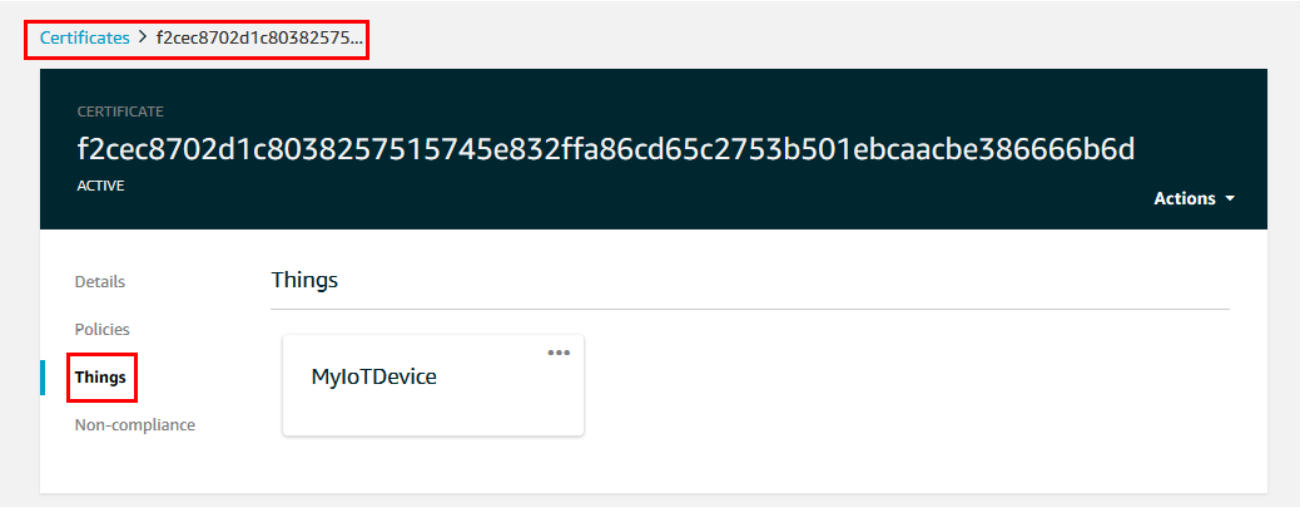


Figure 17: List Attached Things

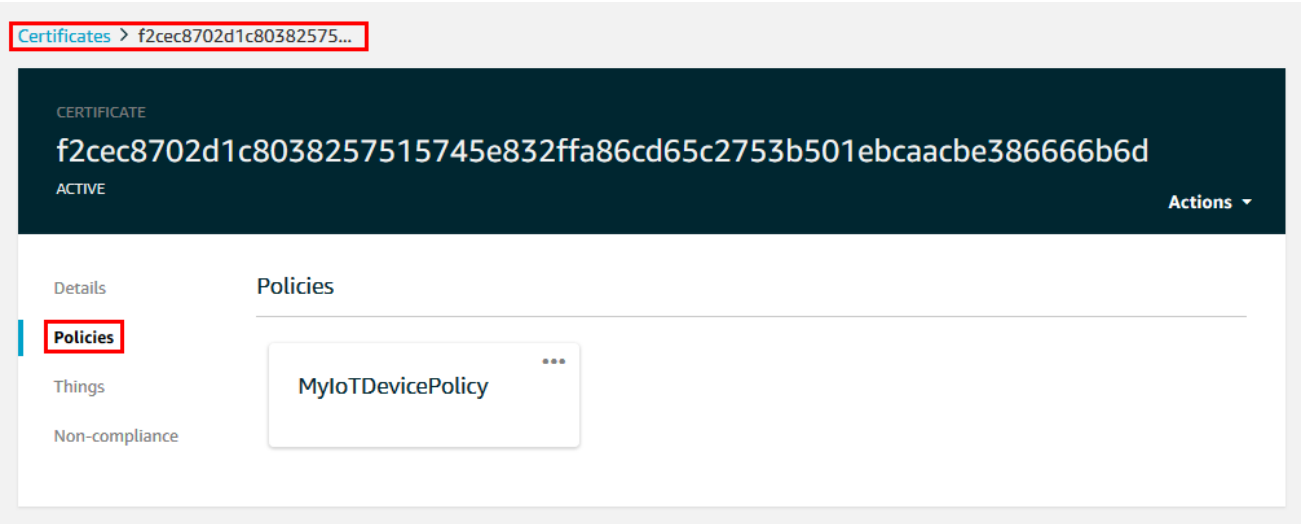


Figure 18: List Attached Policies

2.3. Find MQTT Connection Address and Port

Find the MQTT connection server as below:

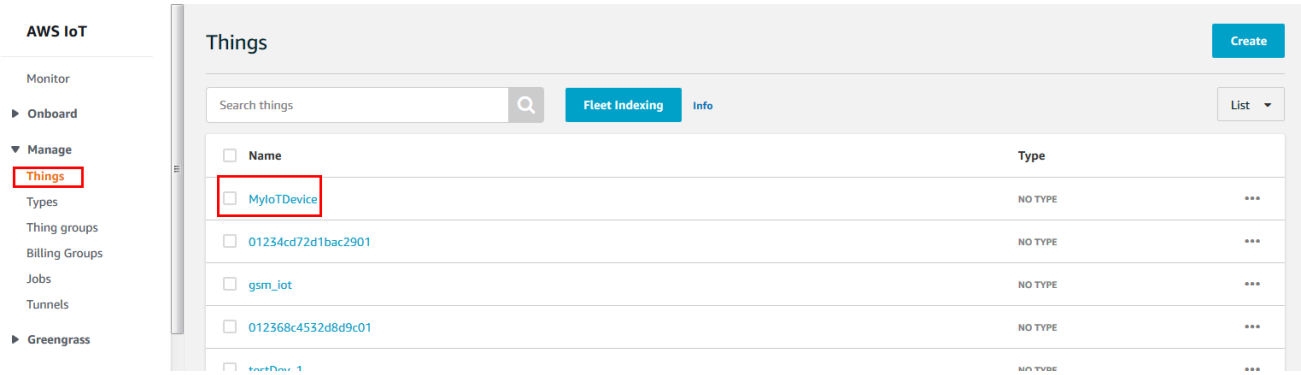


Figure 19: Things Interface

Choose the things you just created and choose Interact.

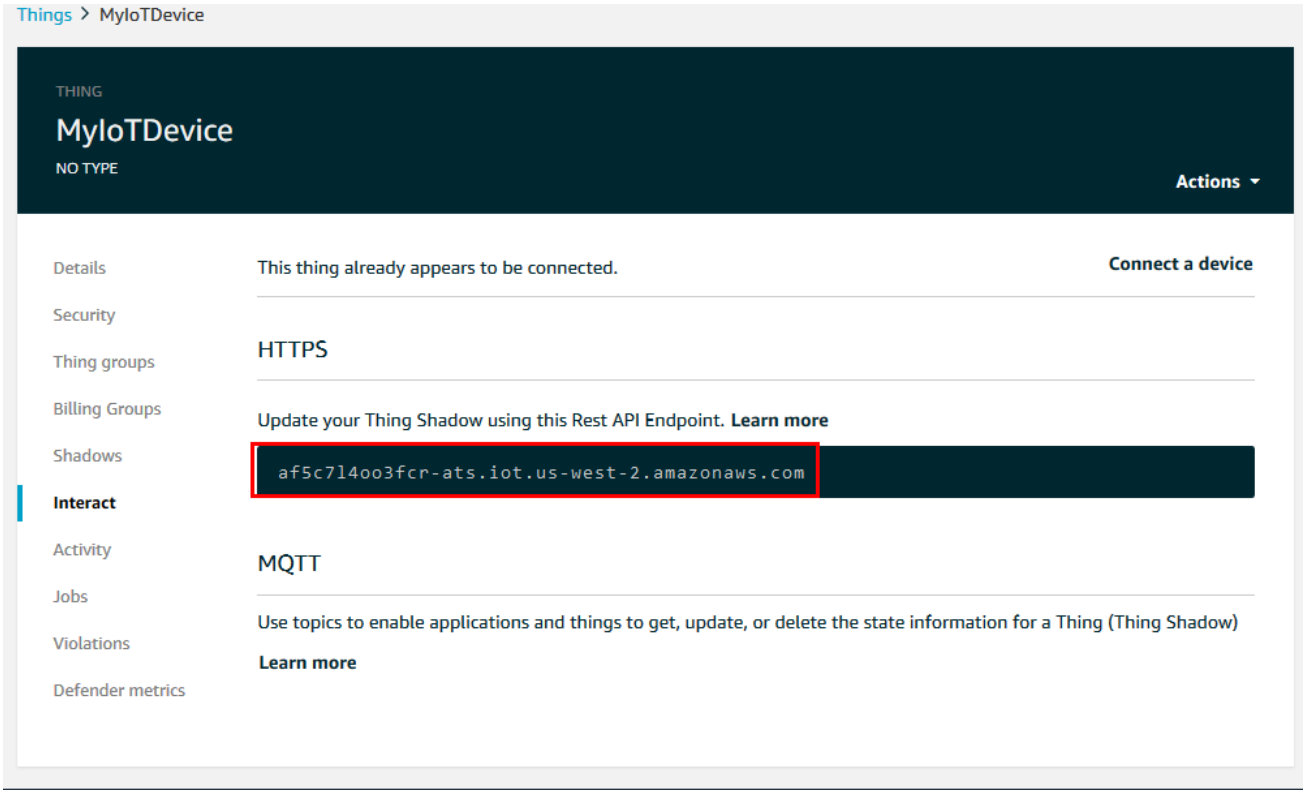


Figure 20: Find MQTT Connection Address

The MQTTS connection port supported by AWS IoT platform is shown as below.

Protocols, Port Mappings, and Authentication

The following table shows each protocol supported by AWS IoT, the authentication method, and port used for each protocol.

Protocol, Authentication, and Port Mappings

Protocol	Authentication	Port	ALPN ProtocolName
MQTT	X.509 client certificate	8883, 443†	`x-amzn-mqtt-ca`
HTTPS	X.509 client certificate	8443, 443†	`x-amzn-http-ca`
HTTPS	SigV4	443	N/A
MQTT over WebSocket	SigV4	443	N/A

Figure 21: MQTT Connection Port

2.4. Import Certificates and Connect to AWS IoT Platform

Quectel EVB can test the connection between AWS IoT platform and the module. The EVB has antenna, SIM card and USB power as the figure below shows. Such as BG95-M3 can use GSM/CAT-M/NB-IoT network. The USB port can be a virtual port to connect with the serial port tool. Please install the *Quectel_LTE_Windows_USB_Driver* in your PC first. Then you can use the *QCOM* tool to communicate with the EVB.

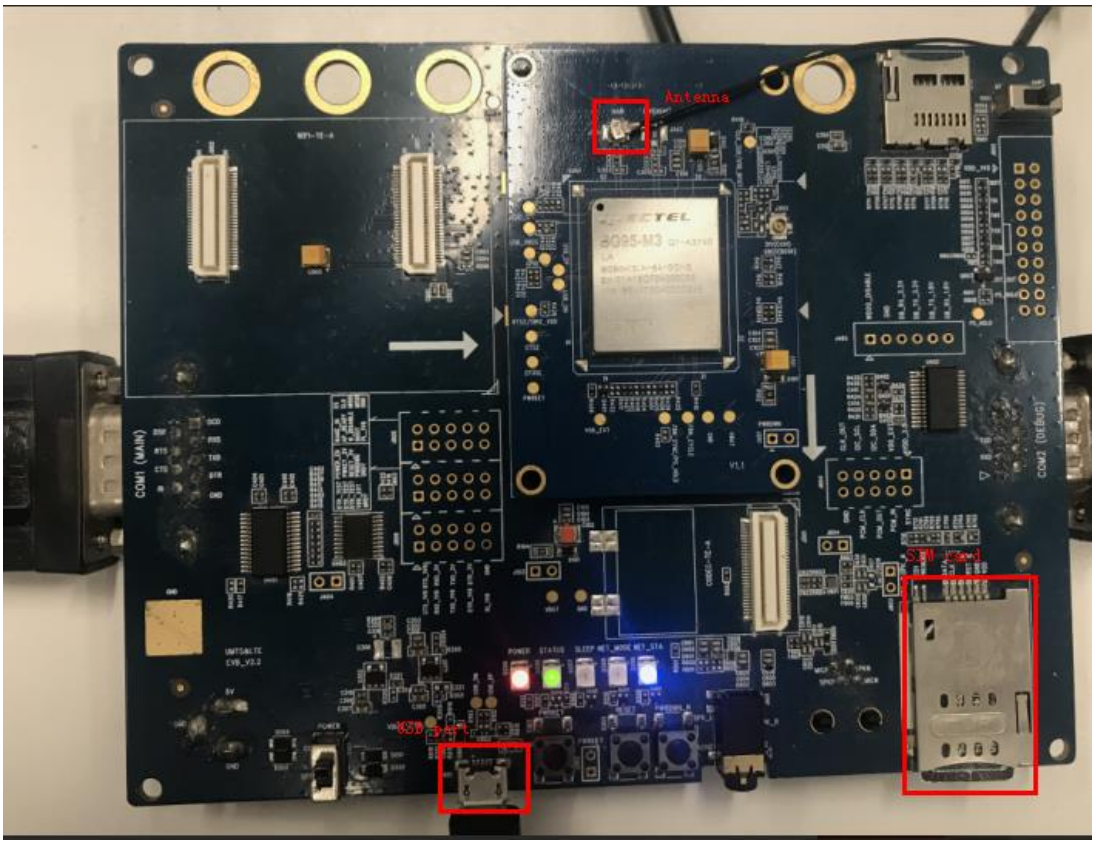


Figure 22: Quectel EVB

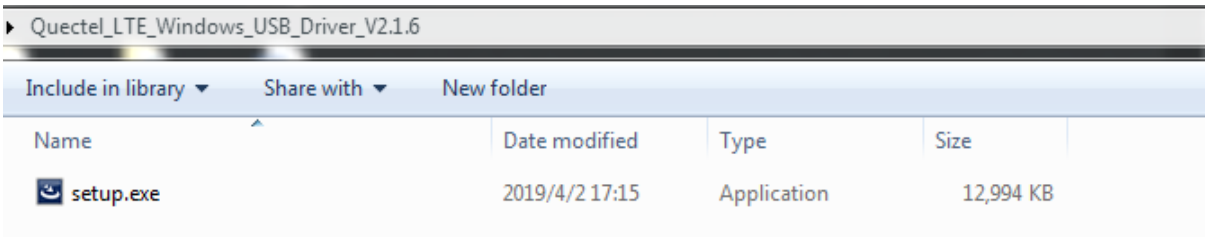


Figure 23: Quectel LTE Windows USB *Driver*

Select Setup.exe to install *Quectel_LTE_Windows_USB_Driver,* after install finished PC serial port as below:

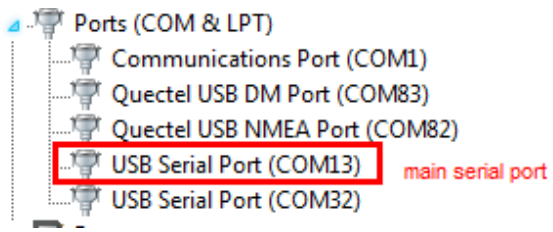


Figure 24: PC serial port

Import certificates into the module and connect the module to AWS IoT platform with AT command related to MQTTS. The process is shown as below.

Step 1: Upload the certificates into the module

Figure 25: AWS Connection Certificates

Upload the certificates into the module:

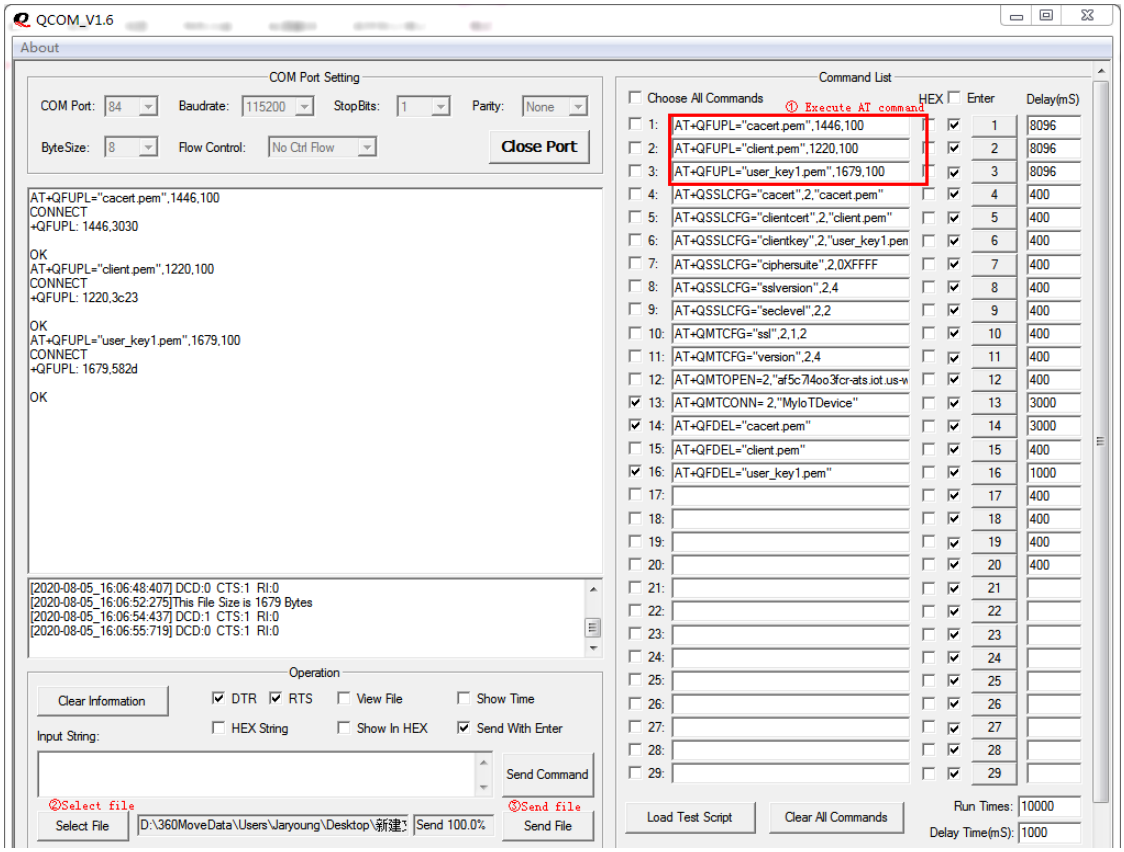


Figure 26: Upload Certificates to Module

Step 2: Configure TLS option:

```
AT+QSSLCFG="cacert",2,"cacert.pem"
OK
AT+QSSLCFG="clientcert",2,"client.pem"
OK
AT+QSSLCFG="clientkey",2,"user_key1.pem"
OK
AT+QSSLCFG="ciphersuite",2,0XFFFF
OK
AT+QSSLCFG="sslversion",2,4
OK
AT+QSSLCFG="seclevel",2,2
OK
AT+QMTCFG="ssl",2,1,2
OK
```

Figure 27: Configure TLS Option

Step 3: Configure MQTTS and MQTT protocol version

```
AT+QMTCFG="ssl",2,1,2
OK
AT+QMTCFG="version",2,4
OK
```

Figure 28: Configure MQTT Option

Step 4: Connect to AWS IoT with MQTTS.

```
AT+QMTOPEN=2,"af5c7l4oo3fcr-ats.iot.us-west-2.amazonaws.com",8883
OK

+QMTOPEN: 2,0
AT+QMTCONN= 2,"MyIoTDevice"
OK

+QMTCONN: 2,0,0
```

Figure 29: Connect Server

2.5. Use the Device Shadow Service

Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow).

Refer to https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html

Shadow topics

The topics in this section are used by named and unnamed shadows. The topics used by each differ only in the topic prefix. This table shows the topic prefix used by each shadow type.

ShadowTopicPrefix value	Shadow type
$aws/things/*thingName*/shadow	Unnamed (classic) shadow
$aws/things/*thingName*/shadow/name/*shadowName*	Named shadow

To create a complete topic, select the *ShadowTopicPrefix* for the type of shadow to which you want to refer, replace *thingName*, and *shadowName* if applicable, with their corresponding values, and then append that with the topic stub as shown in the following table. Remember that topics are case sensitive.

Figure 30: Shadow Topics

Step 5: Subscribe and publish messages:

```
AT+QMTPUB=2,1,1,0,"$aws/things/MyIoTDevice/shadow/get/accepted"
> Hello AWS IoT!
OK

+QMTPUB: 2,1,0

+QMTRECV: 2,1,"$aws/things/MyIoTDevice/shadow/get/accepted","Hello AWS IoT!"
```

Figure 31: Publish message

3 Example

This chapter provides examples for AWS IoT platform access authentication. The following shows the whole process of accessing to AWS IoT with MQTTS.

3.1. Configure the Network

```
//AT+QCFG="nwscanmode",[,<scanmode>]
//<scanmode>:    0    Automatic
//               1    GSM only
//               3    LTE only
AT+QCFG="nwscanmode",3                  //Configure the scan mode to LTE only.
OK

//AT+QCFG="iotopmode"[,<mode>]
//<mode>     Number format. Network category to be searched under LTE RAT.
//           0    eMTC
//           1    NB-IoT
//           2    eMTC and NB-IoT
AT+QCFG="iotopmode",1                   //Configure the network to NB-IoT.

OK

//AT+QCFG="band"[,<gsmbandval>,<emtcbandval>,<nbiotbandval>]
//<gsmbandval>:  A hexadecimal value that specifies the GSM frequency band. If it is set to 0, it means
                 not to change GSM frequency band.
//<emtcbandval>: A hexadecimal value that specifies the eMTC frequency band. If it is set to 0 or
                 0x40000000, it means not to change the frequency band
//<nbiotbandval>: A hexadecimal value that specifies the NB-IoT frequency band. If it is set to 0 or
                 0x40000000, it means not to change the frequency band
//For the supported band list, please refer to Chapter 4.

AT+QCFG="band",0,0,10                   //Configure NB-IoT network BAND 5, the hexadecimal value
                                          is 0x10, only need input 10.
OK
AT+CEREG?;+QNWINFO;+QCSQ                //Query the network status.
+CEREG: 0,1                             //Registered NB-IoT network
```

```
+QNWINFO: "CAT-NB1","46011","LTE BAND 5",2506

+QCSQ: "CAT-NB1",-80,-94,103,-15

OK
```

3.2. Load Certificates

```
//If the module already has certificates, please delete the certificates with AT+QFDEL first.
AT+QFDEL="cacert.pem"
OK
AT+QFDEL="client.pem"
OK
AT+QFDEL="user_key1.pem"
OK
//The server certificate size is 1446 bytes, the timeout is 5000 s, upload the certificates after echoing CONNECT.
AT+QFUPL="cacert.pem",1446,5000
CONNECT
-----BEGIN CERTIFICATE-----
MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dGhvcml0eSAtIEcyMB4XDTA5
MDkwMTAwMDAwMFoXDTM3MTIzMTIzNTk1OVowgZgxCzAJBgNVBAYTAlVTMRAwDgYD
VQQIEwdBcml6b25hMRMwEQYDVQQHEwpTY290dHNkYWxlMSUwIwYDVQQKExxTdGFy
ZmllbGQgVGVjaG5vbG9naWVzLCBJbmMuMTswOQYDVQQDEzJTdGFyZmllbGQgU2Vy
dmljZXMgUm9vdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkgLSBHMjCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBANUMOsQq+U7i9b4Zl1+OiFOxHz/Lz58gE20p
OsgPfTz3a3Y4Y9k2YKibXlwAgLIvWX/2h/klQ4bnaRtSmpDhcePYLQ1Ob/bISdm2
8xpWriu2dBTrz/sm4xq6HZYuajtYlIlHVv8loJNwU4PahHQUw2eeBGg6345AWh1K
Ts9DkTvnVtYAcMtS7nt9rjrnvDH5RfbCYM8TWQIrgMw0R9+53pBlbQLPLJGmpufe
hRhJfGZOozptqbXuNC66DQO4M99H67FrjSXZm86B0UVGMpZwh94CDklDhbZsc7tk
6mFBrMnUVN+HL8cisibMn1lUaJ/8viovxFUcdUBgF4UCVTmLfwUCAwEAAaNCMEAw
DwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFJxfAN+q
AdcwKziIorhtSpzyEZGDMA0GCSqGSIb3DQEBCwUAA4IBAQBLNqaEd2ndOxmfZyMI
bw5hyf2E3F/YNoHN2BtBLZ9g3ccaaNnRbobhiCPPE95Dz+I0swSdHynVv/heyNXB
ve6SbzJ08pGCL72CQnqtKrcgfU28elUSwhXqvfdqlS5sdJ/PHLTyxQGjhdByPq1z
qwubdQxtRbeOlKyWN7Wg0I8VRw7j6IPdj/3vQQF3zCepYoUz8jcI73HPdwbeyBkd
iEDPfUYd/x7H4c7/I9vG+o1VTqkC50cRRj70/b17KSa7qWFiNyi2LSr2EIZkyXCn
0q23KXB56jzaYyWf/Wi3MOxw+3WKt21gZ7IeyLnp2KhvAotnDU0mV3HaIPzBSlCN
```

sSi6
-----END CERTIFICATE-----
+QFUPL: 1188,2d13

OK

//The client certificate size is 1220 bytes, the timeout is 5000 s, upload the certificates after echoing **CONNECT**.
AT+QFUPL="client.pem",1220,5000
CONNECT
-----BEGIN CERTIFICATE-----
MIIDWTCCAkGgAwIBAgIUEU8Sdtdxv7TSMa+qSJctGwP/ef4wDQYJKoZIhvcNAQEL
BQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi5jb20g
SW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDgwNTA2MTgz
NloXDTQ5MTIzMTIzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0
ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANyZlfrksZfD2Wz07SHU
BMSGhxABZFmJeW8+/R1j88imNG+EK32pDiS6foD1zoYkpZF5bZyqSgn+GW4WYLPB
yU2AAG2IewsF9eOXc87Nk4GdtS+1Qrbjf+JkcmrDMHDFq6ugzEhjSP3CHMlU5SJy
TJTdY0HxuR+aQbty8d6zyqNe6nFLe8NuUbQ7AmSMAtJQDODTPsVGv3dzbO6FI3TR
IQCCp33sLZ4AYhRE9y+jhJZ0hHk/ehe9D68sbeWOfbAkFySCmIDFdBGLIaQ6z+jW
7uOxZhBaxbxoS1dqT7j3lNqsoR59hrwz8lKD29HQPRrjSFv1x+nSV/tfUAl/c+Tn
0psCAwEAAaNgMF4wHwYDVR0jBBgwFoAUoJJ7YR4/P2r0/vbPNTRM7EXwcoAwHQYD
VR0OBBYEFOrBYZ82vY5gWGR2AoWDRLvmUHpyMAwGA1UdEwEB/wQCMAAwDgYDVR0P
AQH/BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IBAQCeSbUzBjF9wVN8x+a0F7Zjtq0Z
/CJAK6g+iNOcSpXBCXeER0mjijFfiGQUN0By6kTf67yv5RS3yZZtFS+3g62psf7q
LDWm/vnClp4B4ZehNiGyZHfC5X+mN9HCe2ej+npukHBwPFgzf0e0PsY25LrnNTtW
sRcWzQAdG1YJuk0u4ai+NcThYrJKNcNj2BHRk1+rZiSRp19Jwzks6xaJrYAT2zcx
tcT5gJqI6yCq1HSmF603JxksSlYFdSSc/Yx+7O5zR5g/sNehh/BL43SZFAhIy2r1
tOg3g6iwILrKD0s3/VdoK8p8xfZwF8MksKpKi14LKwAb2eGnCAhgfB8JKvjK
-----END CERTIFICATE-----
+QFUPL: 1224,7a6c

OK

//The client key size is 1679 bytes, the timeout is 5000 s, upload the client key after echoing **CONNECT**.
AT+QFUPL="user_key1.pem",1679,5000
CONNECT
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA3JmV+uSxl8PZbPTtIdQExIaHEAFkWYl5bz79HWPzyKY0b4Qr
fakOJLp+gPXOhiSlkXltnKpKCf4ZbhZgs8HJTYAAbYh7CwX145dzzs2TgZ21L7VC
tuN/4mRyasMwcMWrq6DMSGNI/cIcyVTlInJMlN1jQfG5H5pBu3Lx3rPKo17qcUt7
w25RtDsCZIwC0lAM4NM+xUa/d3Ns7oUjdNEhAIKnfewtngBiFET3L6OElnSEeT96
F70Pryxt5Y59sCQXJIKYgMV0EYshpDrP6Nbu47FmEFrFvGhLV2pPuPeU2qyhHn2G
vDPyUoPb0dA9GuNIW/XH6dJX+19QCX9z5OfSmwIDAQABAoIBAB7Zh6TfiIptxCE9

```
m0/ow4XsPkUZvLvcwtOw3lrW2IfN3nVd6WsYUjcGsZw9Q9V0mIbgkVigY9xD4bCI
hpt81Tb2WJj4xiRDgCCE1JguHZu5v1XdpmfHJuI1I98UTtjme9xzjz3sTI/YLohl
S+nmTBlhYHkfYcEupFSffX2kbQktgWotYbUX9f9h2KlXAyvASekUz6EgyZXsHRV2
963uuJgZl2kYinQ9zaku01x14qtMIFfAZRLI/xZol4Qr65d+582FO1+EjZmffV55
TfV2a2wnE/RtGRQyq9Z8I2EPOsP8DuHogsHZwAV7761Z5n7DqF4DoKanR7W8SqCV
zH2FbjkCgYEA7g4DlpxsrG8sH3nnUeb5V09Skdy1i1IdRRsGELTpyfXRFKnYsEIA
e51GMNd19L5Q+7e87LpjBZLKsV3DwlFNiRIwS6YHp6b2WuOxVkwHnsthqLe+zfac
xmi/WFCkEcwGEoCkEW55uut2X1UzZ8zt6ZTGuiREYa1mp3MYo2u/MlUCgYEA7Tq6
j4bEjhOJ1+fxaDCnak7rcOIv3hECnBemXrp5PnA4+a8ffDhg8KHxC3gR4Jd0BHGa
U9jPjfC0XMjyJqzIjXH1/JyfNZOjovjGIcy7LUZ4PRESsbXE0INL6pvmeAvAJZeG
l59U4aTIy85+gMM9/ZxOy1c5Qta6Yx+Jna7uQS8CgYEAtiRC2poVDFqDiBHdy+zO
Gt+2baRRw6ewfY+PtHi36K8MqqLKCRJ3DB3p9rTkq75yeCa9huOVoE4QiGUVwzgx
+w7PpRIECWLWW/SbW0wRCI7UyiLM+woWKjk7LneFEZjpjH6hCjVdLoe6qnamPmWu
I2qSIxpct9/VC4ok7+UhBYECgYEA5Wr9Tp4jac8FaHJbqMocrZeC45kqd5R1tKS+
dS/a3wJCZ1zvkv8m6K+D3/aPO2bcgQuoFtw+5OLaWjzOyY4hKYEDRfflDeicZIq
cjd+8KsMzum67Xd+zbOb8Fgyive6K+Cc/fbNbKYitc6N0lJ+mcvW+5jvvG7Ss9el
C4RHz2sCgYBhykXUv0zuAYHhmUQhu/5YspuacwZTm0P27TQm7iX9wPaa7vwUPCEB
v4AWjORZqCURbRiaonLCQ15ZZwg/GV1IfzsX1+Wlrv3V4LhQ6gIkpuN8pGOMv3vA
GQEUj6zg6ALKdYwmFd8J3BYE3m3OGeOifk5OzDqCyyuNxOwAwh1peQ==
-----END RSA PRIVATE KEY-----
+QFUPL: 1675,150

OK
AT+QFLST                                    //Query the list file.
+QFLST: "cacert.pem",1446
+QFLST: "client.pem",1220
+QFLST: "user_key1.pem",1679

OK
```

3.3. Active PDP Context

```
//AT+QICSGP=<contextID>[,<context_type>,<APN>[,<username>,<password>[,<authentication>]]]
//<contextID>           Integer type. The context ID. Range: 1–16.
//< context_type >:     Integer type. The protocol type.
                        1   IPv4
//<APN> :               String type. The access point name.
//<username>:           String type. The username.
//<password> :          String type. The password
//<authentication>      Integer type. The authentication methods.
//                      0   None
//                      1   PAP
```

```
//                    2   CHAP
//                    3   PAP or CHAP
AT+QICSGP=1,1,"m2m64.com.attz","","",0
OK
AT+QIACT=1                          //Active PDP contextID 1.
OK
```

3.4. Configure SSL Option

```
AT+QSSLCFG="cacert",2,"cacert.pem"     //Configure the path of server certificate for SSL context 2.
OK
AT+QSSLCFG="clientcert",2,"client.pem" //Configure the path of client certificate for SSL context 2.
OK
AT+QSSLCFG="clientkey",2,"user_key1.pem" //Configure the path of client private key for SSL
                                          context 2.
OK
AT+QSSLCFG="seclevel",2,2              //Configure the authentication mode for SSL context 2,
                                         SSL authentication mode: server and client
                                         authentication if requested by the remote server
OK
AT+QSSLCFG="sslversion",2,4            //SSL authentication version
OK
AT+QSSLCFG="ciphersuite",2,0xFFFF          //Cipher suite
OK
AT+QSSLCFG="ignorelocaltime",2,1       //Ignore the time of authentication
OK
```

3.5. Configure MQTT Option

```
AT+QMTCFG="SSL",2,1,2                  //Configure MQTT session to SSL mode for SSL context 2
OK
AT+QMTCFG="version",2,4                //Configure MQTT protocol version to MQTT v3.1.1
OK
```

3.6. MQTT Connection and Data Interaction

```
//Start MQTT SSL connection, please refer to chapter 2.3.
AT+QMTOPEN=2,"a2sgasbshsff52-ats.iot.us-west-2.amazonaws.com",8883
```

```
OK

+QMTOPEN: 2,0
AT+QMTCONN=2,"Quectel"                    //Connect to MQTT server
OK

+QMTCONN: 2,0,0
AT+QMTSUB=2,1,"$aws/things/MyIoTDevice/shadow/get/accepted",0 //Subscribe to topics,
                                                              //MyIoTDevice is device
                                                              //name
OK

+QMTSUB: 2,1,0,

//Publish messages. After echoing >, input the payload, and tap “ctrl+Z” send the data.
AT+QMTPUB=2,1,1,0,"$aws/things/MyIoTDevice/shadow/get/accepted "
>Hello AWS IoT!
OK

+QMTPUB: 2,1,0

+QMTRECV: 2,1,"$aws/things/MyIoTDevice/shadow/get/accepted","Hello AWS IoT!"
AT+QMTCLOSE=2                             //Close MQTTS connection
OK

+QMTCLOSE: 2,0
```

NOTE

For the details of above commands, please refer to ***Document [1]***, ***Document [2]***, ***Document [3]*** and ***Document [4]***.

4 Support Band List

Table 1: Support Band List

Parameter	Supported List		Note
<gsmbandval>	00000000 No change 00000001 GSM 900MHz 00000002 GSM 1800MHz 00000004 GSM 850MHz 00000008 GSM 1900MHz 0000000F Any frequency band		eg.: 0x0a=0x02(GSM1800)+0x08(GSM1900) This parameter is valid on BG95-M3 module only.
<emtcbandval>	0x1 (BAND_PREF_LTE_BAND1) 0x2 (BAND_PREF_LTE_BAND2) 0x4 (BAND_PREF_LTE_BAND3) 0x8 (BAND_PREF_LTE_BAND4) 0x10 (BAND_PREF_LTE_BAND5) 0x80 (BAND_PREF_LTE_BAND8) 0x800 (BAND_PREF_LTE_BAND12) 0x1000 (BAND_PREF_LTE_BAND13) 0x2000 (BAND_PREF_LTE_BAND14) 0x20000 (BAND_PREF_LTE_BAND18) 0x40000 (BAND_PREF_LTE_BAND19) 0x80000 (BAND_PREF_LTE_BAND20) 0x1000000 (BAND_PREF_LTE_BAND25) 0x2000000 (BAND_PREF_LTE_BAND26) 0x4000000 (BAND_PREF_LTE_BAND27) 0x8000000 (BAND_PREF_LTE_BAND28) 0x40000000 (BAND_PREF_LTE_BAND31) 0x20000000000000000 (BAND_PREF_LTE_BAND66) 0x800000000000000000 (BAND_PREF_LTE_BAND72) 0x1000000000000000000 (BAND_PREF_LTE_BAND73) 0x10000000000000000000 (BAND_PREF_LTE_BAND85)	LTE B1 LTE B2 LTE B3 LTE B4 LTE B5 LTE B8 LTE B12 LTE B13 LTE B14 LTE B18 LTE B19 LTE B20 LTE B25 LTE B26 LTE B27 LTE B28 LTE B31 LTE B66 LTE B72 LTE B73 LTE B85	eg.: 0x15=0x01(LTEB1)+0x04(LTE B3)+0x10(LTE B5)
<nbiotbandval>	0x1 (BAND_PREF_LTE_BAND1)	LTE B1	

0x2 (BAND_PREF_LTE_BAND2)	LTE B2
0x4 (BAND_PREF_LTE_BAND3)	LTE B3
0x8 (BAND_PREF_LTE_BAND4)	LTE B4
0x10 (BAND_PREF_LTE_BAND5)	LTE B5
0x80 (BAND_PREF_LTE_BAND8)	LTE B8
0x800 (BAND_PREF_LTE_BAND12)	LTE B12
0x1000 (BAND_PREF_LTE_BAND13)	LTE B13
0x2000 (BAND_PREF_LTE_BAND14)	LTE B14
0x20000 (BAND_PREF_LTE_BAND18)	LTE B18
0x40000 (BAND_PREF_LTE_BAND19)	LTE B19
0x80000 (BAND_PREF_LTE_BAND20)	LTE B20
0x1000000 (BAND_PREF_LTE_BAND25)	LTE B25
0x2000000 (BAND_PREF_LTE_BAND26)	LTE B26
0x4000000 (BAND_PREF_LTE_BAND27)	LTE B27
0x8000000 (BAND_PREF_LTE_BAND28)	LTE B28
0x40000000 (BAND_PREF_LTE_BAND31)	LTE B31
0x20000000000000000 (BAND_PREF_LTE_BAND66)	LTE B66
0x800000000000000000 (BAND_PREF_LTE_BAND72	LTE B72
0x1000000000000000000 (BAND_PREF_LTE_BAND73)	LTE B73
0x100000000000000000000 (BAND_PREF_LTE_BAND85)	LTE B85

5 Appendix A References

Table 2: Related Documents

SN	Document Name	Remark
[1]	Quectel_BG95&BG77_AT_Commands_Manual	AT command manual
[2]	Quectel_BG95&BG77&BG600L_Series_MQTT_Application_Note	MQTT application note
[3]	Quectel_BG95&BG77_SSL_Application_Note	SSL application note
[4]	Quectel_BG95&BG77_FILE_Application_Note	FILE application note

NOTE

The *Quectel_LTE_Windows_USB_Driver*, *QCOM* tool and documents please obtain from Quectel. https://www.quectel.com/support/download.htm

Table 3: Terms and Abbreviations

Abbreviation	Description
AWS	Amazon Web Services
IoT	Internet of Things
MQTT(S)	Message Queuing Telemetry Transport (Security)
TLS	Transport Layer Security
SSL	Secure Sockets Layer
QCOM	Serial port tool
EVB	Evaluation Board