

ESP32_TO_AWS



IoT Core ☆

Connect Devices to the Cloud

Monitor

Connect

Connect one device

▶ Connect many devices

Test

▶ Device Advisor

MQTT test client

Device Location [New](#)

Manage

▼ All devices

Things

Thing groups

Thing types

Fleet metrics

Things (1) [Info](#)

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

[Refresh](#) [Advanced search](#) [Run aggregations](#) [Edit](#) [Delete](#) [Create things](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	Thing type
--------------------------	------	------------

步驟

1. Create single thing
2. Thing name: {自訂}
3. Auto-generate a new certificate (recommended)
4. Create policy

Policies (1)

Select up to 10 policies to attach to this certificate.

[Refresh](#) [Create policy](#)

< 1 > [Settings](#)

Create policy [Info](#)

AWS IoT Core policies allow you to manage access to the AWS IoT Core data plane operations.

Policy properties

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name

自訂

PolicyName

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

► Tags - optional

Policy statements

Policy examples

Policy document [Info](#)

Builder

JSON

An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.

Policy document

policy.txt内容取代

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "",  
7       "Resource": ""  
8     }  
9   ]  
10 }
```



policy.txt

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive",
        "iot:PublishRetain"
      ],
      "Resource": "arn:aws:iot:us-east-1:{accountid}:topic/esp32/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": "arn:aws:iot:us-east-1:{accountid}:topicfilter/esp32/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": "arn:aws:iot:us-east-1:{accountid}:client/*"
    }
  ]
}
```

可用區&accountid需修改

N. Virginia ▼

voclabs/user2174082=10

US East (N. Virginia)

us-east-1

voclabs/user2174082=1090

Account ID: [REDACTED]
Federated user: voclabs/user2174082=109021085@live.asia.edu.tw

Download certificates and keys

×

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate

Oca3b25315f...te.pem.crt

Deactivate certificate

Download

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file

Oca3b25315f31c36ecb6818...7ed0e21-public.pem.key

Download

Private key file

Oca3b25315f31c36ecb6818...ed0e21-private.pem.key

Download

✔ Key downloaded

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint

RSA 2048 bit key: Amazon Root CA 1

Download

Amazon trust services endpoint

ECC 256 bit key: Amazon Root CA 3

Download

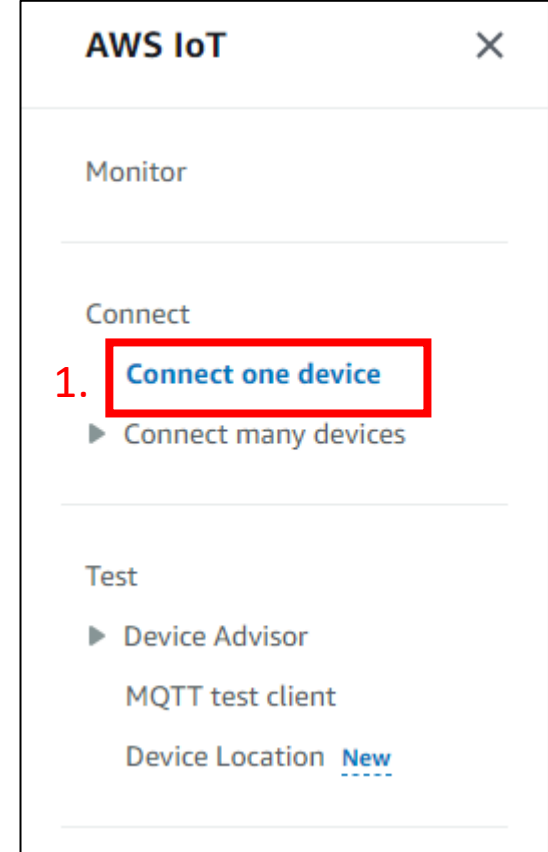
If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

Done

- 1.選擇剛建立的policy按下create thing
- 2.出現此視窗需先下載(此視窗只出現一次)
- 3.按下done後即完成

公鑰暫時沒用到

```
1  #include <pgmspace.h>
2
3  #define THINGNAME "esp32_1" // replace with thing name
4
5  const char WIFI_SSID[] = ""; // replace with wifi ssid
6  const char WIFI_PASSWORD[] = ""; // replace with wifi password
7
8  const char AWS_IOT_ENDPOINT[] = ""; // replace with iot endpoint
9
10 // Amazon Root CA 1 :
11 static const char AWS_CERT_CA[] PROGMEM = R"EOF(
12 -----BEGIN CERTIFICATE-----
13
14 -----END CERTIFICATE-----
15 )EOF";
16
17 // Device Certificate
18 static const char AWS_CERT_CRT[] PROGMEM = R"KEY(
19 -----BEGIN CERTIFICATE-----
20
21 -----END CERTIFICATE-----
22 )KEY";
23
24 // Device Private key
25 static const char AWS_CERT_PRIVATE[] PROGMEM = R"KEY(
26 -----BEGIN RSA PRIVATE KEY-----
27
28 -----END RSA PRIVATE KEY-----
29 )KEY";
```



4. From the terminal window, enter this command:

2.
ping



Test

▼ Device Advisor

Test suites

Test runs and results

MQTT test client

Device Location [New](#)

Subscribe to a topic

Publish to a topic

Topic filter [Info](#)

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard

esp32/pub_2

► Additional configuration

Subscribe

```
#define AWS_IOT_PUBLISH_TOPIC    "esp32/pub_2"
#define AWS_IOT_SUBSCRIBE_TOPIC "esp32/sub_2"
```

Subscriptions

esp32/pub_2

esp32/pub_2



Message payload

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

► Additional configuration

Publish

▼ esp32/pub_2

```
{
  "metrics": 25
}
```

資料來源:<https://awsacademy.instructure.com/courses/57479/modules/items/5049636>

影片參考https://www.youtube.com/watch?v=QctUfGqdR20&ab_channel=codejam