

# Lab3

# Outline

## ▶ AWS IoT

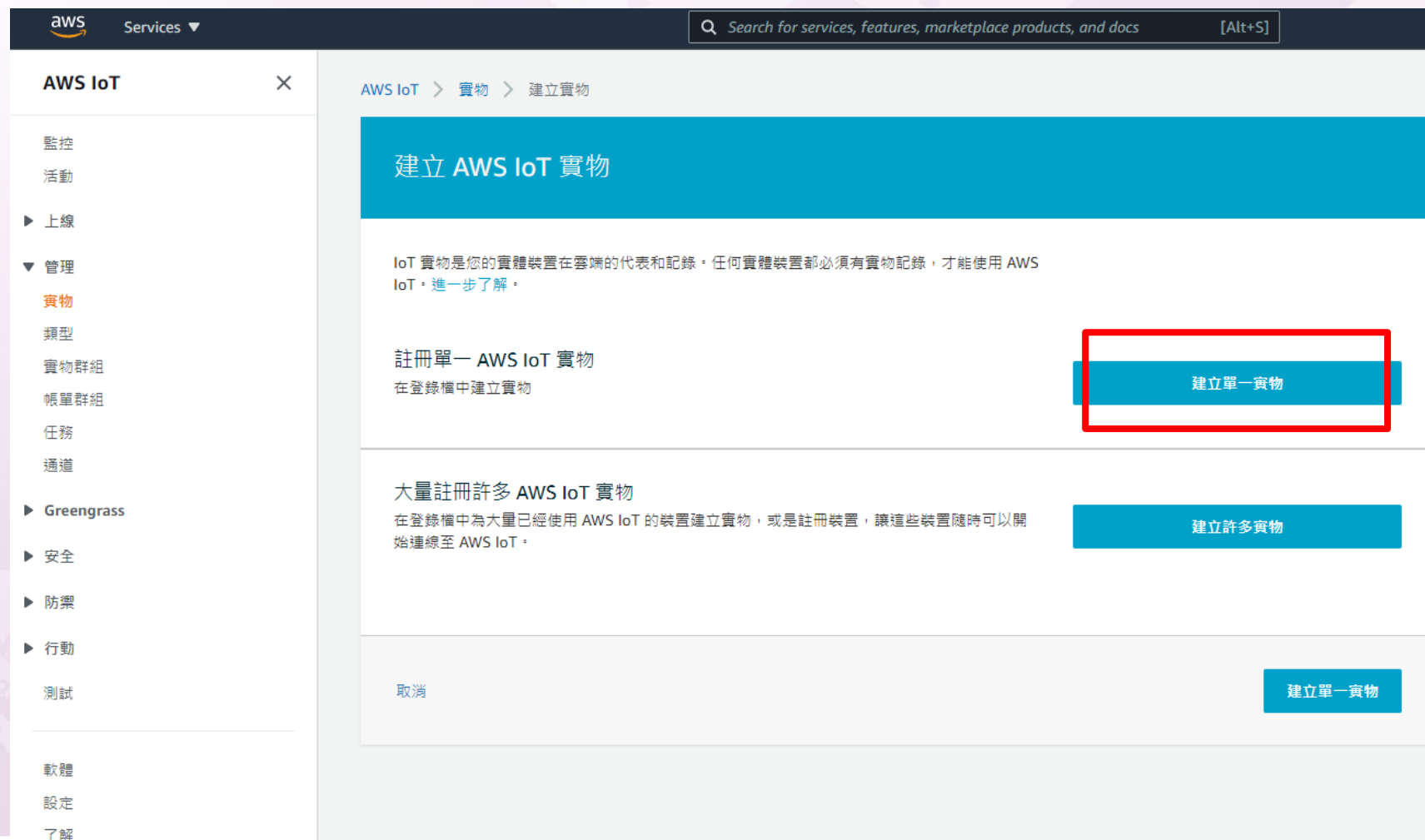
- ▶ IoT 實物與shadow

- ▶ 連接Lambda

## ▶ Homework

# AWS IoT

- ▶ 前往 AWS IoT core 服務
- ▶ 選擇“實物” -> ”建立實物”
- ▶ 點選“建立單一實物”



The screenshot displays the AWS IoT console interface. On the left, a navigation sidebar lists various services, with 'AWS IoT' selected. The main content area is titled '建立 AWS IoT 實物' (Create AWS IoT Thing). It provides an overview of IoT Things and offers two primary options: '註冊單一 AWS IoT 實物' (Register a single AWS IoT Thing) and '大量註冊許多 AWS IoT 實物' (Bulk register many AWS IoT Things). The '註冊單一 AWS IoT 實物' option is highlighted with a red rectangle, and its corresponding '建立單一實物' (Create single thing) button is also highlighted. The '大量註冊許多 AWS IoT 實物' option has a '建立許多實物' (Create many things) button. At the bottom, there are '取消' (Cancel) and '建立單一實物' (Create single thing) buttons.

aws Services

Search for services, features, marketplace products, and docs [Alt+S]

AWS IoT

監控  
活動  
▶ 上線  
▼ 管理  
實物  
類型  
實物群組  
帳單群組  
任務  
通道  
▶ Greengrass  
▶ 安全  
▶ 防禦  
▶ 行動  
測試

AWS IoT > 實物 > 建立實物

## 建立 AWS IoT 實物

IoT 實物是您的實體裝置在雲端的代表和記錄。任何實體裝置都必須有實物記錄，才能使用 AWS IoT。 [進一步了解](#)。

註冊單一 AWS IoT 實物  
在登錄檔中建立實物

建立單一實物

大量註冊許多 AWS IoT 實物  
在登錄檔中為大量已經使用 AWS IoT 的裝置建立實物，或是註冊裝置，讓這些裝置隨時可以開始連線至 AWS IoT。

建立許多實物

取消

建立單一實物

# AWS IoT

- ▶ 輸入名稱
- ▶ 點選下一步

## 將您的裝置新增至實物登錄檔

1/3

此步驟會在實物註冊表中創建一個條目，並為您的設備創建一個實物影子。

名稱

為您的實物指定名稱

### 將類型套用到此實物

使用實物類型可為共用一個類型的實物提供一致的登錄檔資料，藉以簡化裝置管理。類型提供實物一組共同的屬性，這些屬性描述裝置的身分識別和功能，以及品名種類。

實物類型

未選擇類型

建立類型

### 將此實物新增至群組

將實物新增至群組，可讓您在遠端使用工作來管理裝置。

實物群組

群組 /

建立群組

變更

### 設定可搜尋的實物屬性 (選用)

為這其中一個或多個屬性輸入一個值，讓您可以在登錄檔中搜尋您的實物。

屬性索引鍵

提供屬性索引鍵，例如：Manufacturer

值

提供屬性索引鍵，例如：Acme-Corporation

清除

新增另一個

顯示實物影子 ▼

取消

返回

下一步

# AWS IoT

▶ 點選”建立憑證”

AWS IoT > 實物 > 建立實物 > 將您的裝置新增至實物登錄檔 > 新增憑證

建立實物

新增實物的憑證

步驟 2/3

憑證用於驗證您的裝置與 AWS IoT 的連線。

**單鍵建立憑證作業 (建議)**  
這將會使用 AWS IoT 的憑證授權機構來產生憑證、公有金鑰和私有金鑰。

**建立憑證**

**使用 CSR 建立**  
根據您擁有的私有金鑰上傳您自己的憑證簽署要求 (CSR)。

**使用 CSR 建立**

**使用我的憑證**  
註冊您的 CA 憑證，並將自己的憑證用於一個或多個裝置。

**開始使用**

**略過憑證並建立實物**  
稍後必須將憑證新增至實物，您的裝置才能連線至 AWS IoT。

**建立沒有憑證的實物**

# AWS IoT

- ▶ 下載憑證以及公私鑰
- ▶ 請務必確實下載保存關閉，頁面後將無法再存取
- ▶ 之後點選 root CA 的下載

成功  
已成功建立實物。

成功  
已成功產生憑證。請下載憑證檔案。

憑證已建立！

下載這些檔案，然後存放在安全的地方。可以隨時擷取憑證，但是關閉此頁面後，就無法擷取私有及公有金鑰。

為了進行裝置連線，您需要下載以下項目：

此實物的憑證	2eb00753f7.cert.pem	<a href="#">下載</a>
公有金鑰	2eb00753f7.public.key	<a href="#">下載</a>
私有金鑰	2eb00753f7.private.key	<a href="#">下載</a>

您還需要下載 **AWS IoT 的根 CA**：

AWS IoT 的根 CA [下載](#)

啟用

取消

完成

連接政策

# AWS IoT

- ▶ 選擇 root CA 1
- ▶ 將內容存成 .txt 檔案

## CA certificates for server authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT is signed by one of the following root CA certificates:

### VeriSign Endpoints (legacy)

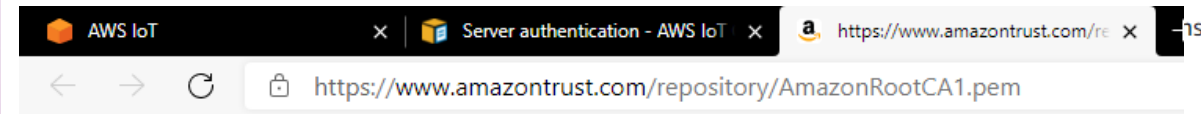
- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

### Amazon Trust Services Endpoints (preferred)

#### Note

You might need to right click these links and select **Save link as...** to save these certificates as files.

- 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.



```
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPm1jZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDEExBBBwF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDExNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxHzANBgNVBAoTBkFtYXNjYXZlZm9udGVzZm9udGVzZm9udGVz
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHhKXj
ca9HgFB0fW7Y14h29J1o91ghYPL0hAEvrAItHtOgQ3p0sqTQNroBvo3bSMgHFzZM
906II8c+6zf1tRn4SWiw3te5djdYZ6k/oI2peVKVuRf4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbu1Hheb4mjUcAwhmahRwa6
VOUjw5H5SNz/0egwLX0tdHA114gk957EwW67c4cX8jJGKLhD+rcdqsq08p8kd11L
93FcXmn/6pUCyzikr1A4b9v7LWIbxcceV0F34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jg5SubJrIqg0CAwEAaAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zA0BgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwM1JQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBChUA
A4IBAQC8jdaQZChGsV2USggNiMOruYou6r41K5IpDB/G/wkju0yKX9rbxenDI
U5PMCCjJmCXPi6T53iHTfIUJrU6adTrCC2qJehZERxhlb11bjjt/msv0tadQ1wUs
N+gDS63pYaACbvXy8Mwly7Vu33PqUXHeeE6V/Uq2V8vIT096LXFvKW1JbYK8U90vv
o/urfQJVTMT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJc3mApzyMZFo6IQ6XU
5MSI+yMRQ+hdKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXegADbkpy
rQXRfboQnoZsG4q5WTP4685QvvG5
-----END CERTIFICATE-----
```

# AWS IoT

- ▶ 點選”啟用”
- ▶ 之後選擇”完成”

成功  
已成功建立實物。

成功  
已成功產生憑證。請下載憑證檔案。

憑證已建立！

下載這些檔案，然後存放在安全的地方。可以隨時擷取憑證，但是關閉此頁面後，就無法擷取私有及公有金鑰。

為了進行裝置連線，您需要下載以下項目：

此實物的憑證	2eb00753f7.cert.pem	<a href="#">下載</a>
公有金鑰	2eb00753f7.public.key	<a href="#">下載</a>
私有金鑰	2eb00753f7.private.key	<a href="#">下載</a>

您還需要下載 AWS IoT 的根 CA：  
[AWS IoT 的根 CA 下載](#)

啟用

取消

完成

連接政策



# AWS IoT

- ▶ 選擇”政策”->”建立政策”
- ▶ 輸入名稱
- ▶ 設定操作權限: ”iot:\*”
- ▶ 設定ARN: ”\*”
- ▶ 選擇”建立”

AWS IoT > 政策 > 建立政策

## 建立政策

建立政策以定義一組獲得授權的操作。您可以授權一個或多個資源 (實體、主題、主題篩選條件) 可執行的操作。如需進一步了解 IoT 政策，請前往 [AWS IoT 政策文件頁面](#)。

名稱

lab3

### 新增陳述式

政策陳述式定義資源可以執行的操作類型。

進階模式

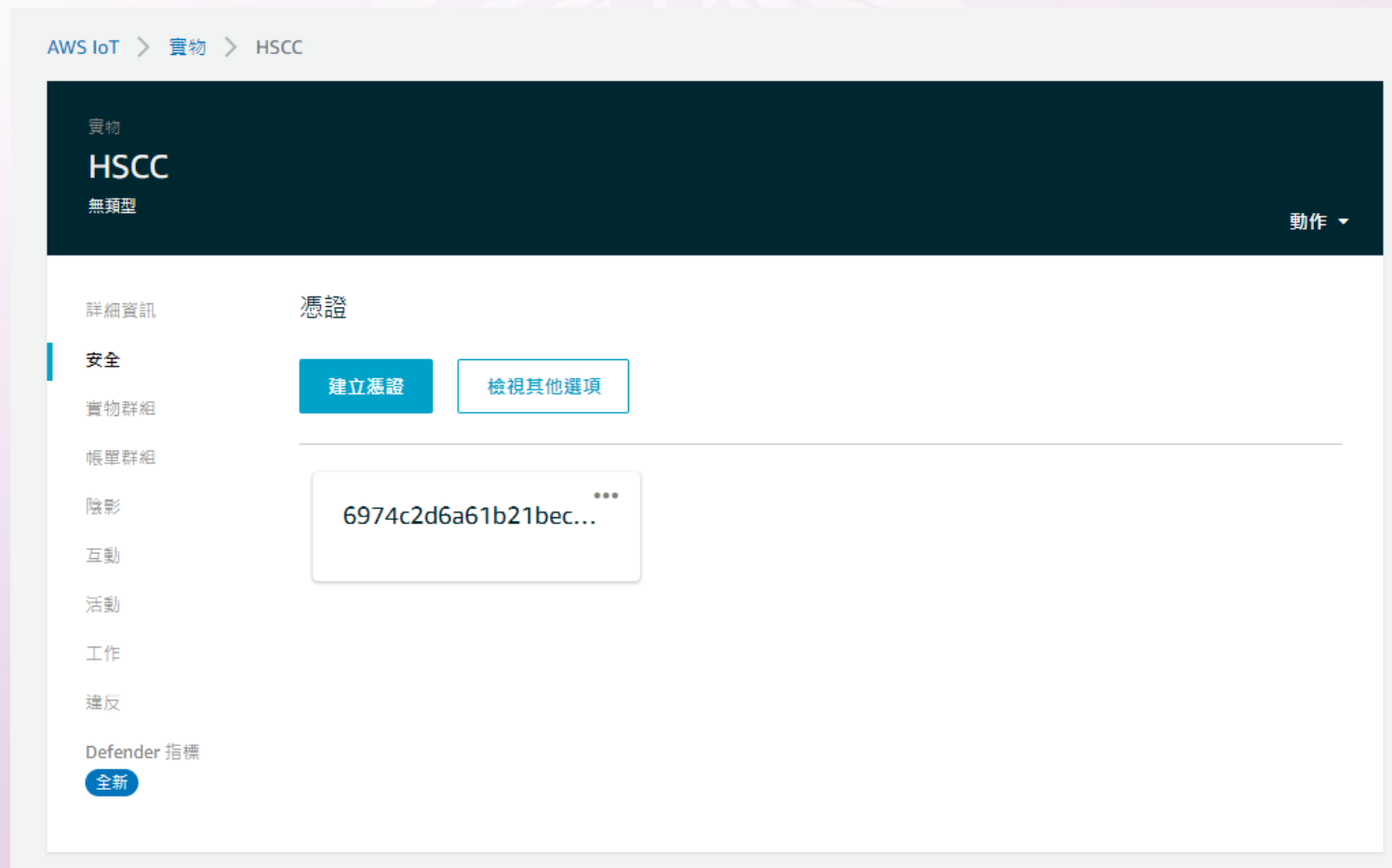
操作	
iot:*	
資源 ARN	
*	
生效	
<input checked="" type="checkbox"/> 允許 <input type="checkbox"/> 拒絕	
	移除

新增陳述式

建立

# AWS IoT

- ▶ 回到剛剛的實物
- ▶ 選擇”安全”
- ▶ 點選剛才建立的憑證



# AWS IoT

## ▶ 選擇“連接政策”

AWS IoT > 憑證 > 6974c2d6a61b21bec5358f8ba5c4be3bf30a2dc7381c893000bdbffd944fbe5c

憑證  
6974c2d6a61b21bec5358f8ba5c4be3bf30a2dc7381c893000bdbffd944fbe5c  
作用中

動作 ▾

- 啟用
- 停用
- 撤銷
- 接受傳輸
- 拒絕傳輸
- 撤銷傳輸
- 開始傳輸
- 連接政策
- 連接實物
- 下載
- 刪除

詳細資訊

政策

實物

不合規

憑證 ARN

憑證 Amazon Resource Name (ARN) 可唯一識別出此憑證。 [進一步了解](#)

`arn:aws:iot:us-east-2:996759461321:cert/6974c2d6a61b21bec5358f8ba5c4b`

詳細資訊

發行者  
OU=Amazon Web Services O\=Amazon.com Inc. L\=Seattle ST\=Washington C\=US  
主旨  
CN=AWS IoT Certificate  
建立日期  
3月 23, 2021, 17:28:51 (UTC+0800)  
生效日期  
3月 23, 2021, 17:26:51 (UTC+0800)  
過期日期  
1月 01, 2050, 07:59:59 (UTC+0800)

# AWS IoT

- ▶ 選擇剛才建立的政策
- ▶ 按下連接

### 將政策連接至憑證

政策將會連接至以下憑證：  
6974c2d6a61b21bec5358f8ba5c4be3bf30a2dc7381c893000bdbffd944fbe5c

選擇一個或多個政策

<input type="checkbox"/> test2	<a href="#">檢視</a>
<input type="checkbox"/> test	<a href="#">檢視</a>
<input checked="" type="checkbox"/> lab3	<a href="#">檢視</a>

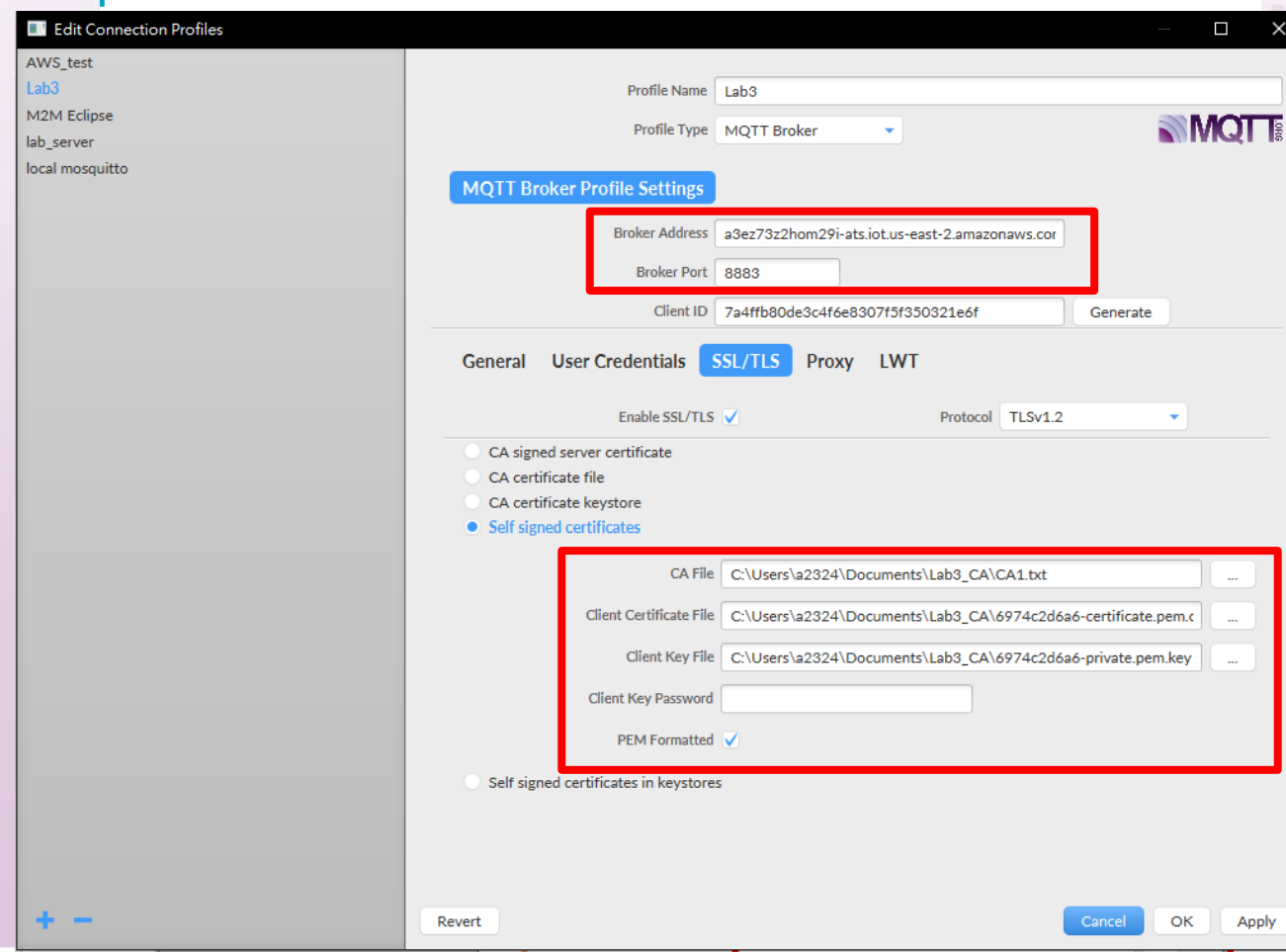
1 已選擇 個政策

取消

連接

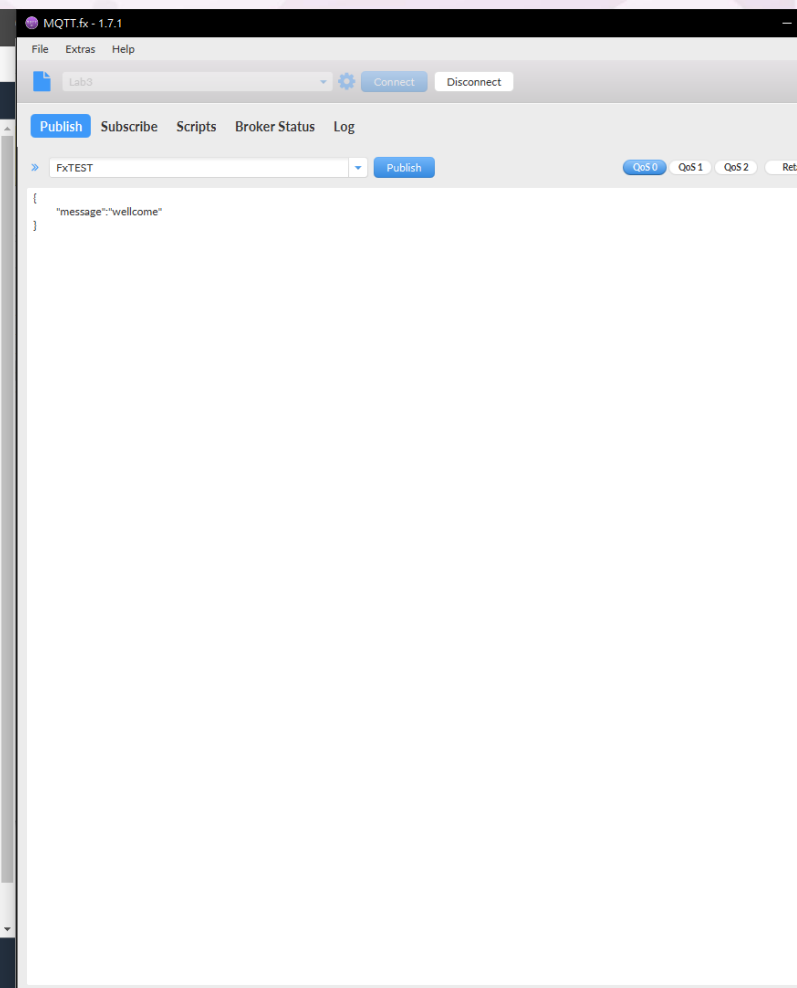
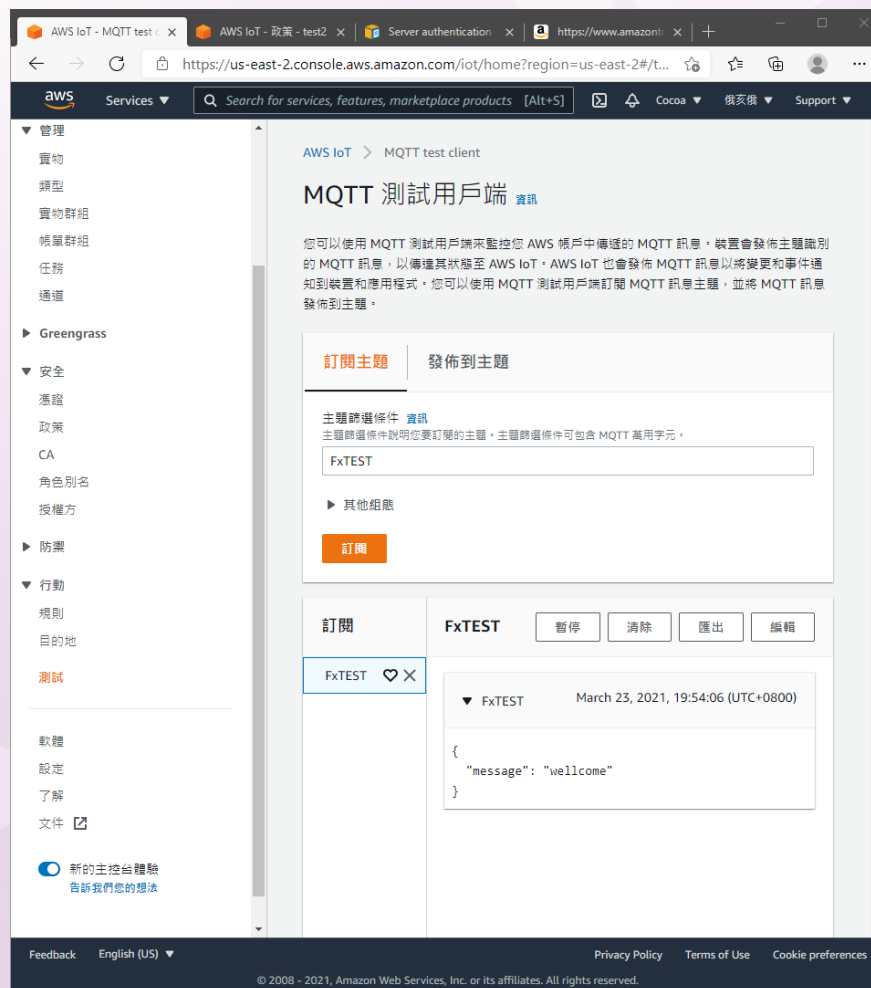
# AWS IoT

- ▶ 打開 MQTT.fx
- ▶ 點選上方的齒輪
- ▶ 增加一個新的 Profile
- ▶ 輸入名稱
- ▶ Address 填上你AWS IoT的URL
  - ▶ 可以在實物中的”互動”找到
- ▶ Port 8883
- ▶ 在SSL/TLS選擇self signed certificates
  - ▶ 依序選擇檔案
    - ▶ Root CA
    - ▶ .crt 檔案
    - ▶ 私鑰
- ▶ 打勾 PEM formatted



# AWS IoT

- ▶ 回到 MQTT.fx 的首頁按下connect，有綠燈代表成功接上broker了
- ▶ 接著在AWS選擇測試
- ▶ 訂閱一個主題
- ▶ 到MQTT.fx 選擇Publish
- ▶ 發布內容到同一個主題
- ▶ 測試畫面就會顯示發布的內容



# AWS IoT

- ▶ 回到“實物”
- ▶ 選擇“陰影”
- ▶ 點選“Classic shadow”
- ▶ 將內容編輯成右邊的形式
- ▶ 關於陰影可以參考[這裡](#)

AWS IoT > 實物 > HSCC > Classic Shadow

實物

## HSCC

無類型

動作 ▾

詳細資訊

安全

實物群組

帳單群組

**陰影**

互動

活動

工作

違反

Defender 指標

全新

### 影子 ARN

影子 ARN 可唯一識別出此實物的影子。

[返回陰影清單](#)

### 影子文件

上次更新：3月 23, 2021, 20:11:26 (UTC+0800)

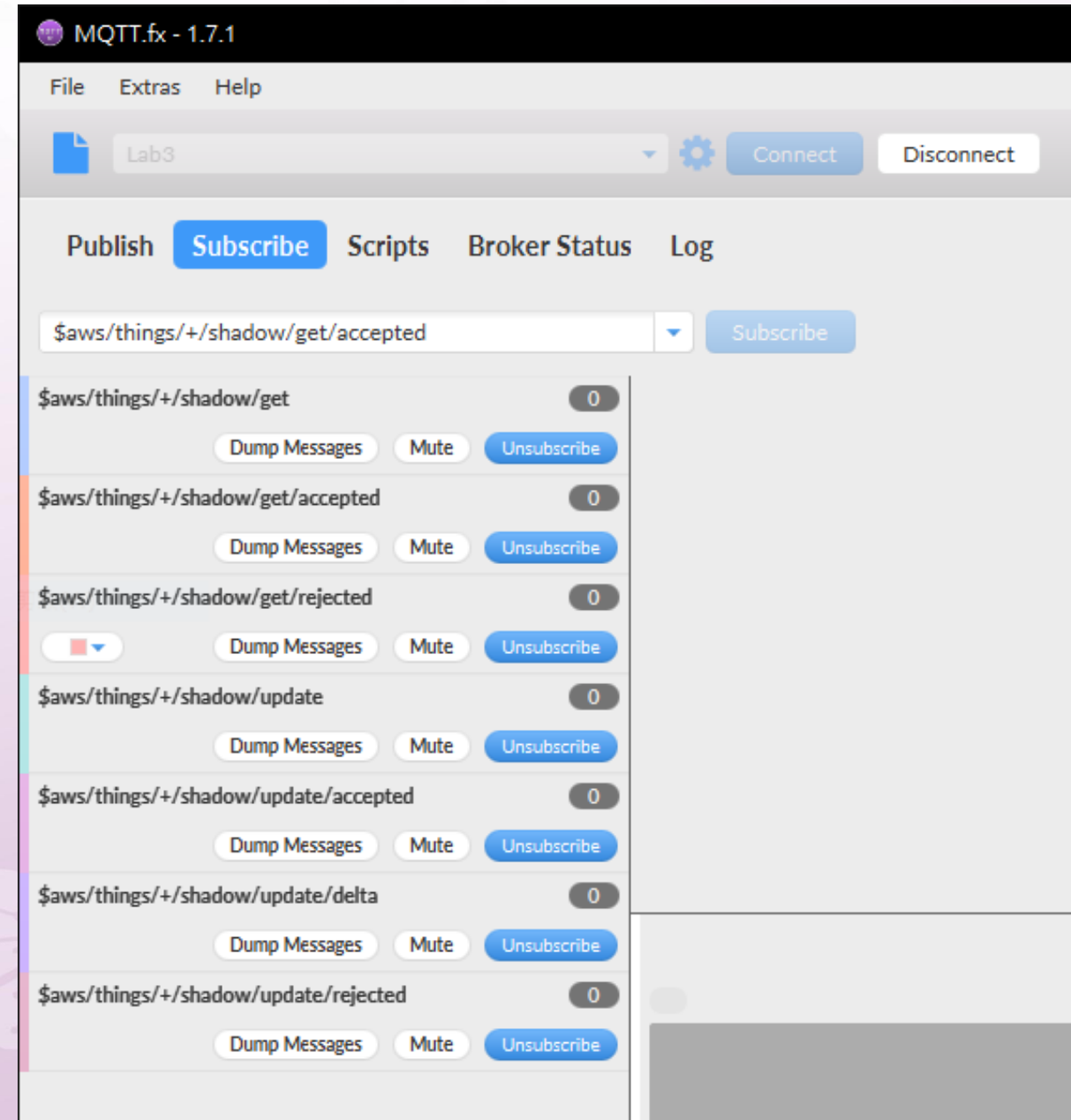
刪除 編輯

影子狀態：

```
{
  "desired": {
    "welcome": "aws-iot",
    "ring": 0
  },
  "reported": {
    "welcome": "aws-iot",
    "humidity": 0,
    "ring": 0
  }
}
```

# AWS IoT

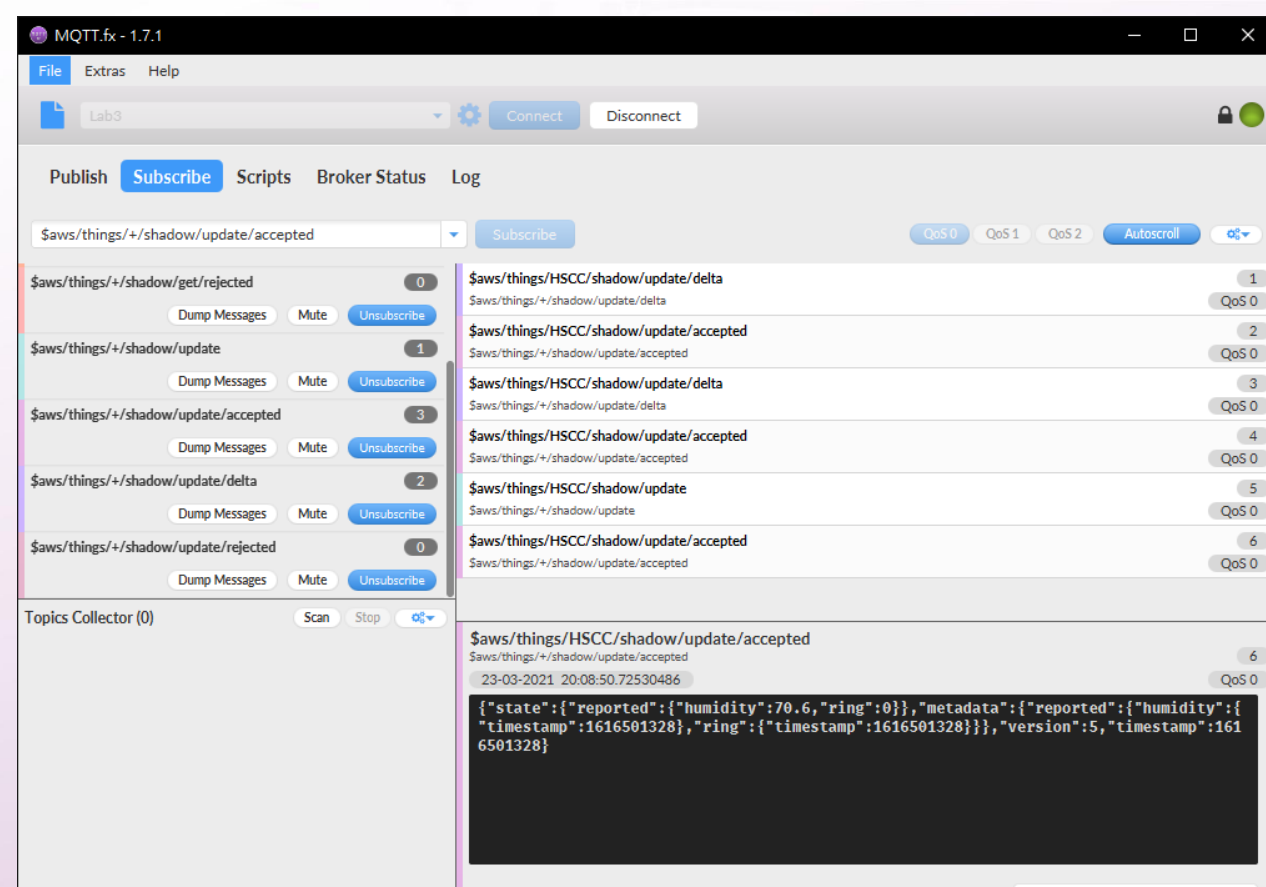
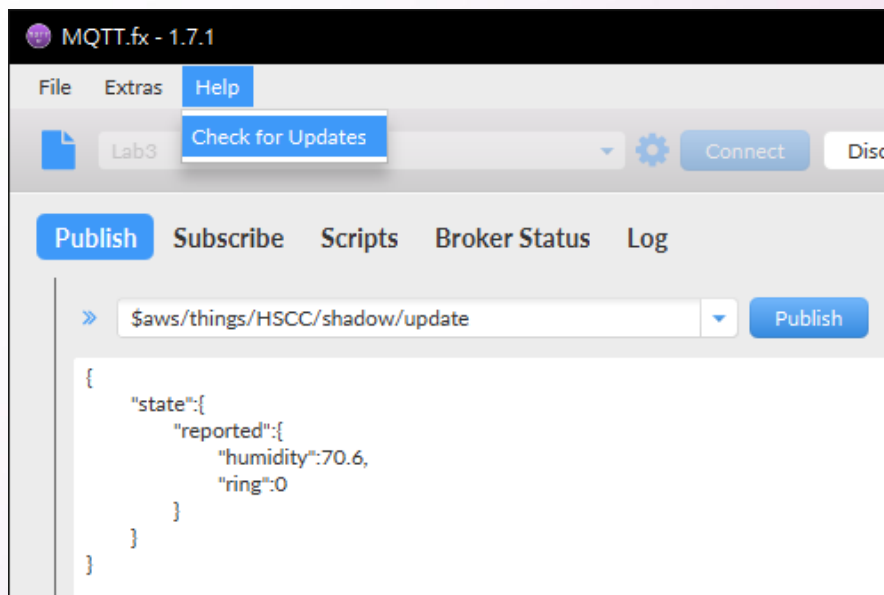
- ▶ 回到MQTT.fx
- ▶ 訂閱以下主題
- ▶ \$aws/things/+/shadow/get
- ▶ \$aws/things/+/shadow/get/accepted
- ▶ \$aws/things/+/shadow/get/rejected
- ▶ \$aws/things/+/shadow/update
- ▶ \$aws/things/+/shadow/update/accepted
- ▶ \$aws/things/+/shadow/update/rejected
- ▶ \$aws/things/+/shadow/update/delta





# AWS IoT

- ▶ 往 \$aws/things/<your device name>/shadow/update 發送更新
- ▶ 注意 payload 格式要與 shadow 相符
- ▶ \*記得要包在 state 裡面
- ▶ \*注意逗號及冒號



影子文件

刪除 編輯

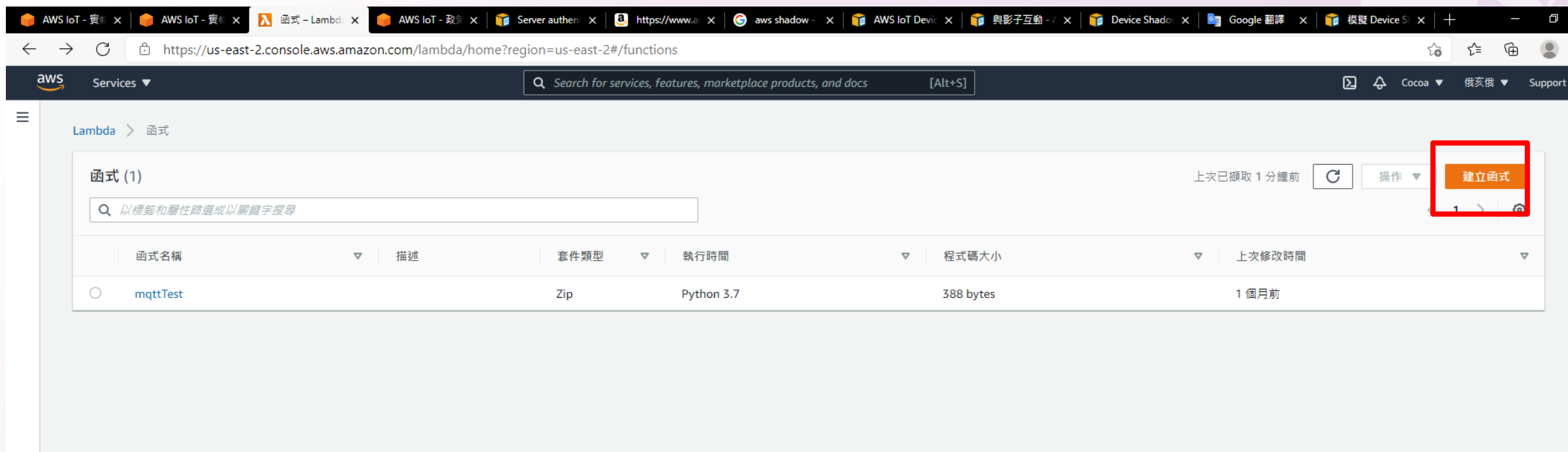
上次更新：3月23, 2021, 20:08:48 (UTC+0800)

影子狀態：

```
{
  "desired": {
    "welcome": "aws-iot",
    "ring": 0
  },
  "reported": {
    "welcome": "aws-iot",
    "humidity": 70.6,
    "ring": 0
  }
}
```

# AWS IoT -- Lambda

- ▶ 選擇“Lambda”
- ▶ 建立新函式

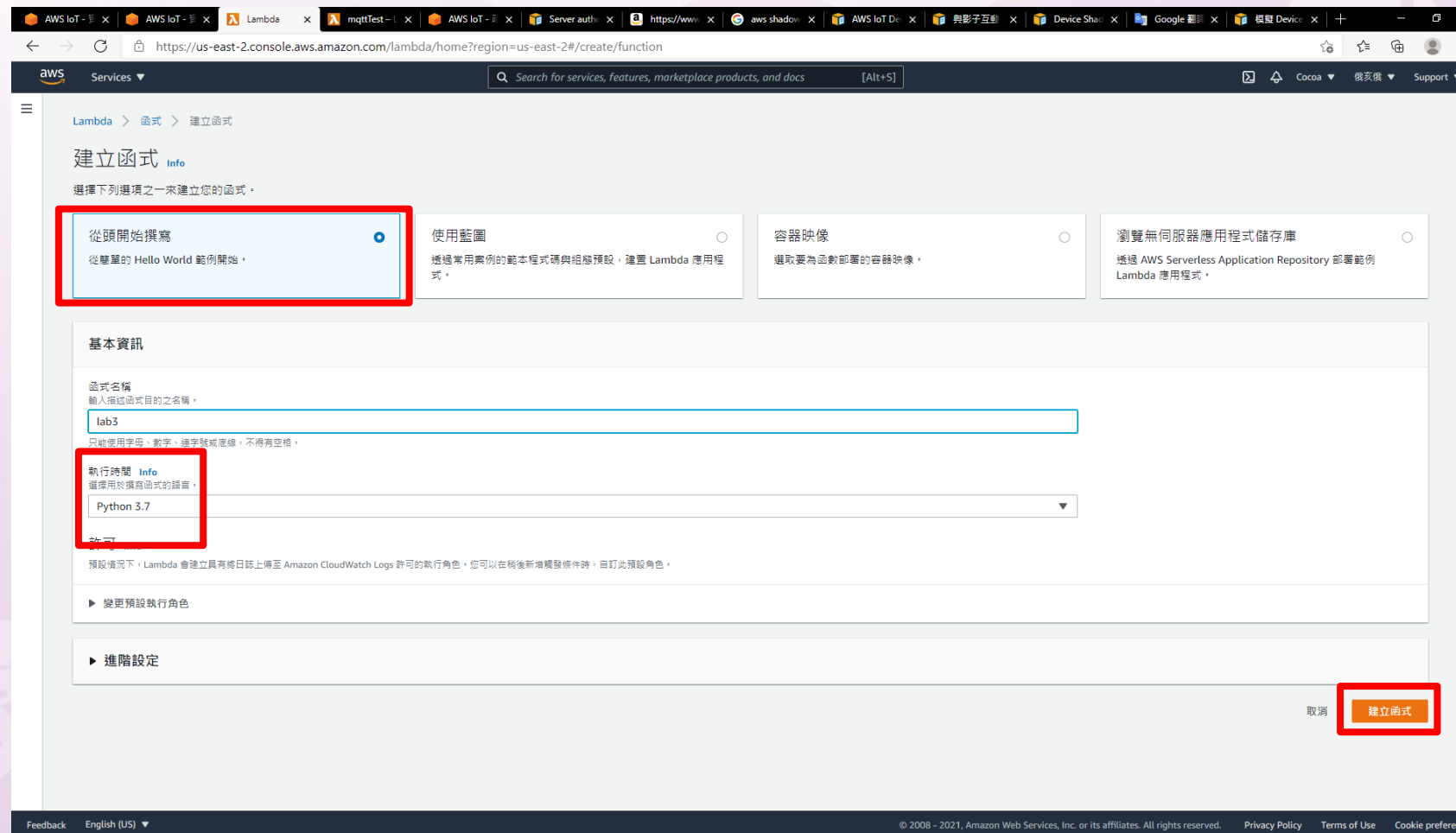


The screenshot shows the AWS Lambda console interface. The browser address bar displays the URL: <https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions>. The console header includes the AWS logo, a search bar, and navigation links. The main content area is titled 'Lambda > 函式' (Functions). It features a search bar with the placeholder text '以標籤和屬性篩選或以關鍵字搜尋' (Filter by tags and attributes or search by keyword). Below the search bar is a table listing functions. The table has columns for 'Function Name', 'Description', 'Runtime', 'Execution Time', 'Code Size', and 'Last Modified Time'. A single function named 'mqttTest' is listed with a runtime of 'Python 3.7' and a code size of '388 bytes'. In the top right corner of the console, there is a 'Create Function' button, which is highlighted with a red box. Other buttons visible include '上次已擷取 1 分鐘前' (Last fetched 1 minute ago), '操作' (Actions), and '建立函式' (Create Function).

Function Name	Description	Runtime	Execution Time	Code Size	Last Modified Time
mqttTest		Zip	Python 3.7	388 bytes	1 個月前

# AWS IoT -- Lambda

- ▶ 選擇從頭開始撰寫
- ▶ 輸入名稱
- ▶ Run time 選擇 Python 3.7
- ▶ 點選建立函式



# AWS IoT -- Lambda

- ▶ 選擇“組態”
- ▶ 在執行角色點選角色名稱

The screenshot displays the AWS Lambda console interface for a function named 'lab3'. The breadcrumb navigation shows 'Lambda > 函式 > lab3'. The function's configuration is shown in the 'lab3' section, including a 'Layers' list and a 'Function ARN' of 'arn:aws:lambda:us-east-2:996759461321:function:lab3'. The 'Execution role' section is highlighted with a red box, showing the role name 'lab3-role-raonm1dz'. The 'Resource summary' section shows the function is associated with Amazon CloudWatch Logs. The left sidebar contains a navigation menu with options like '一般組態', '觸發器', '許可', '目的地', '環境變數', '標籤', 'VPC', 'Monitoring and operations tools', '並行數量', and '非同步叫用'.

# AWS IoT -- Lambda

- ▶ 在角色政策選擇”新增內嵌政策”

角色 > lab3-role-raonm1dz

摘要 刪除角色

角色 ARN	arn:aws:iam::996759461321:role/service-role/lab3-role-raonm1dz
角色說明	<a href="#">編輯</a>
執行個體描述檔 ARN	<a href="#">🔗</a>
路徑	/service-role/
建立時間	2021-03-23 20:18 UTC+0800
上次活動	未在追蹤期間進行存取
工作階段持續時間上限	1 小時 <a href="#">編輯</a>

許可 信任關係 標籤 存取顧問 撤銷工作階段

▼ Permissions policies (1 套用的政策)

[連接政策](#) 新增內嵌政策

政策名稱 ▼	政策類型 ▼	
▶ AWSLambdaBasicExecutionRole-8def08ed-9753-4bf8-a344-e3b89dacb354	受管政策	✕

▶ 權限界限 (not set)

# AWS IoT -- Lambda

- ▶ 搜尋服務iot
- ▶ 找到Publish的操作
- ▶ 資源選擇指定
- ▶ 點選檢閱政策就可以看到新加入的權限了

## 建立政策

政策會定義您可指派給使用者、群組或角色的 AWS 許可。您可以在視覺化編輯器中使用 JSON 建立和編輯政策。 [進一步了解](#)

視覺化編輯器

JSON

[匯入受管政策](#)

[全部展開](#) | [全部折疊](#)

▼ IoT (1 個操作)

[複製](#) | [移除](#)

▶ 服務 IoT

▶ 操作 寫入

Publish

▼ 資源 ☒ 指定

[關閉](#) ☐ 所有資源

topic ?

arn:aws:iot:\*:996759461321:topic/\*

[編輯](#)

☒ 本帳戶中的任一者

▶ 請求條件 [指定請求條件 \(選用\)](#)

[+ 新增其他許可](#)

字元計數：150 個 (最多 10,240 個)。  
包含在 role 內所有內嵌政策字元的目前字元計數：lab3-role-raonm1dz。

[取消](#)

[檢閱政策](#)

# AWS IoT -- Lambda

- ▶ 輸入名稱
- ▶ 點選”建立政策”

## 建立政策

1

2

### 檢閱政策

在建立此政策之前，請提供所需的資訊並檢閱此政策。

名稱\*

Lab3

最多 128 個字元。請使用英數和「+,-,@,\_」字元。

摘要

Q 篩選條件：

服務 ▼

存取層級

資源

請求條件

允許 (1 / 276 個服務) [顯示其餘 275](#)

IoT

限制：寫入

TopicName | string like | All

無

\* 必要

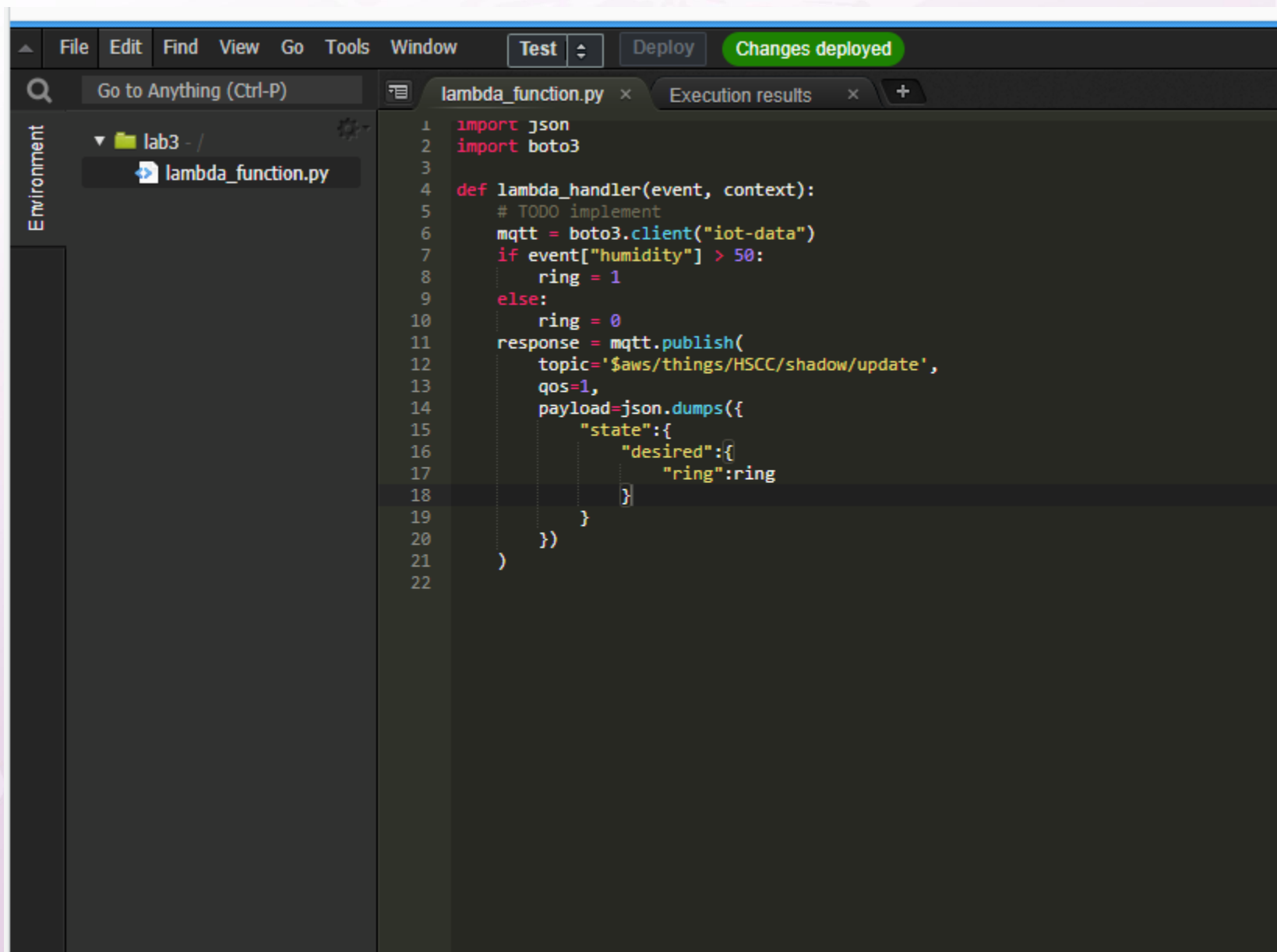
取消

上一個

建立政策

# AWS IoT -- Lambda

- ▶ 回到Lambda的編輯器
- ▶ 輸入右側的code
- ▶ 這個程式是將判斷溼度的值，若濕度太高則響鈴。
- ▶ 注意payload的格式要用json。
- ▶ \*要按下Deploy程式才會發布
- ▶ 關於boto3可以參考[這裡](#)



```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # TODO implement
6     mqtt = boto3.client("iot-data")
7     if event["humidity"] > 50:
8         ring = 1
9     else:
10        ring = 0
11    response = mqtt.publish(
12        topic='$aws/things/HSCC/shadow/update',
13        qos=1,
14        payload=json.dumps({
15            "state":{
16                "desired":{
17                    "ring":ring
18                }
19            }
20        })
21    )
22
```



# AWS IoT -- Lambda

- ▶ 回到 IoT Core 選擇”行動”->”規則”
- ▶ 建立新規則
- ▶ 輸入名稱
- ▶ 輸入IoT SQL將溼度及響鈴狀態抓出來
- ▶ 點選新增操作，選擇lambda函數
- ▶ 選擇剛剛建立的函數

規則查詢陳述式

表示您要以此規則處理之訊息的來源。

使用 SQL 版本

2016-03-23

規則查詢陳述式

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>。例如：SELECT temperature FROM 'iot/topic' WHERE temperature > 50。如需進一步了解如何建構 SQL 陳述式，請參閱 [AWS IoT SQL 參考](#)。

```
1 SELECT state.reported.humidity,state.reported.ring FROM '$aws/things/HSCC/shadow/update/accepted'
```

設定一個或多個操作

選擇一個或多個要在傳入訊息符合上述規則時發生的操作。動作會定義在訊息抵達時進行的其他活動，例如：將這些訊息儲存在資料庫、叫用雲端函數，或傳送通知。（\*必要）

新增操作



叫用傳遞訊息資料的 Lambda 函數  
LAMBDA

我們會為您設定 Lambda 函數的許可。

[新建 Lambda 函數](#)

\*函數名稱

lab3

清除

從顯示的新下拉式清單選取

取消

新增操作

# AWS IoT -- Lambda

- ▶ 往\$aws/things/<your device name>/shadow/update發送更新
- ▶ 如果規則有被執行就會看到Lambda發回來的響鈴狀態

The screenshot shows the MQTT.fx - 1.7.1 interface. The 'Subscribe' button is highlighted. The subscription list on the left includes '\$aws/things/+/shadow/update' with a count of 7. The message list on the right shows a message from '\$aws/things/HSCC/shadow/update/accepted' with a count of 18. The message payload is displayed in a black box:

```
{ "state": { "desired": { "ring": 0 } } }
```

The payload is decoded by the 'Plain Text Decoder'.

The screenshot shows the MQTT.fx - 1.7.1 interface. The 'Subscribe' button is highlighted. The subscription list on the left includes '\$aws/things/+/shadow/update' with a count of 7. The message list on the right shows a message from '\$aws/things/HSCC/shadow/update/accepted' with a count of 19. The message payload is displayed in a black box:

```
{ "state": { "desired": { "ring": 0 } }, "metadata": { "desired": { "ring": { "timestamp": 1616503204 } } }, "version": 11, "timestamp": 1616503204 }
```

The payload is decoded by the 'Plain Text Decoder'.

# Homework

- ▶ Host

- ▶ 140.114.78.132

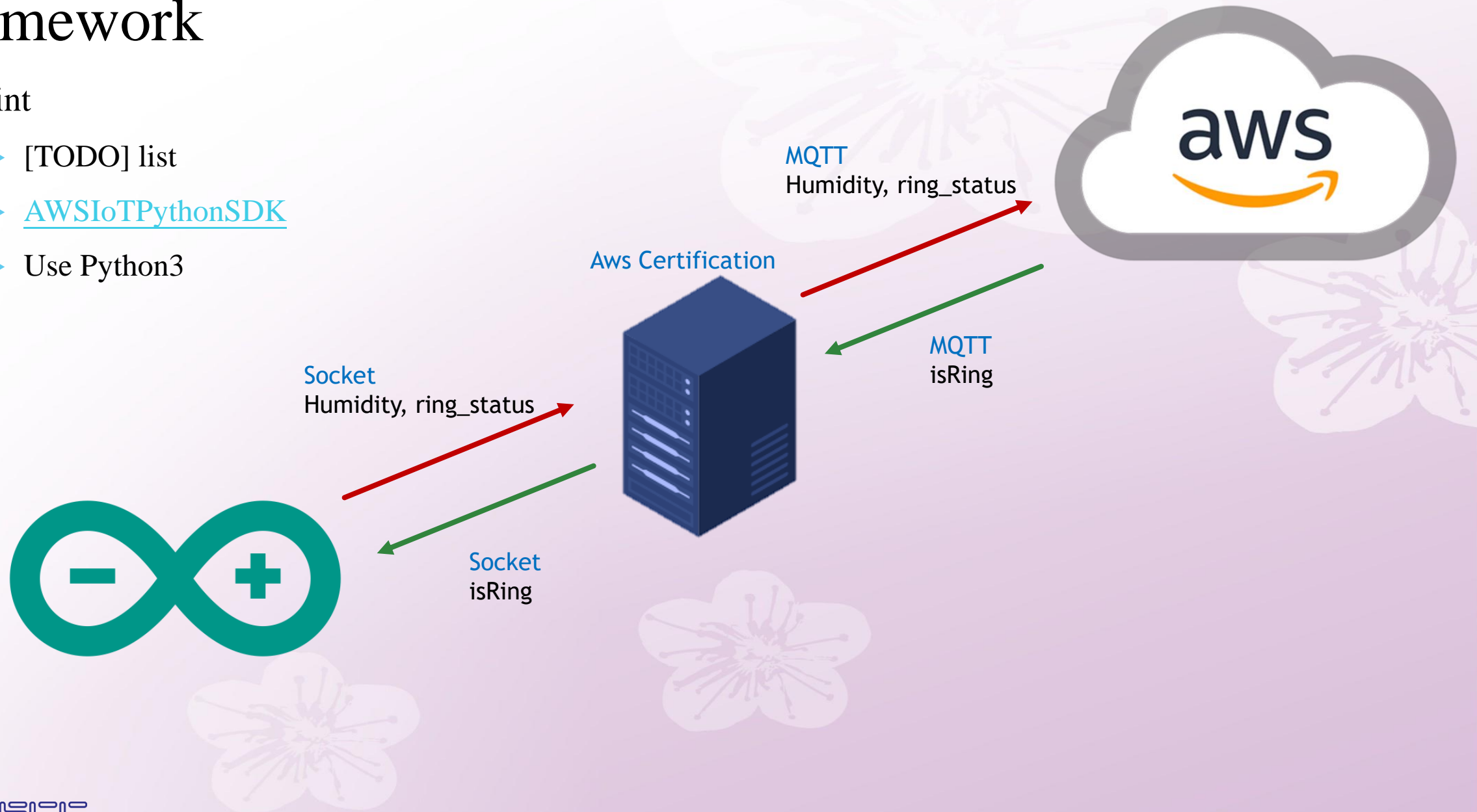
- ▶ 指令

- ▶ ssh <帳號>@140.114.78.132
  - ▶ 帳號: 學號
  - ▶ 預設密碼: 學號
  - ▶ 第一次登入請重新設定密碼

# Homework

## ▶ Hint

- ▶ [TODO] list
- ▶ [AWSIoTPythonSDK](#)
- ▶ Use Python3



# Homework

- ▶ 讀出濕度和響鈴狀態(或LED)
- ▶ 將資料送上Server (140.114.78.132)
- ▶ 在Server上撰寫MQTT程式
- ▶ 利用MQTT將資料 publish 到 AWS
- ▶ 透過Lambda決定響鈴狀態(或LED)
- ▶ 再用MQTT將狀態 subscribe回 Server
- ▶ Server將資料送到BC20
- ▶ BC20把響鈴狀態解碼出來，決定是否響鈴(或LED)
  
- ▶ **請繳交Demo影片 以及Report到elearn**