

AWS Educate 資源融入課 堂上應用

實踐大學資訊科技與管理學系

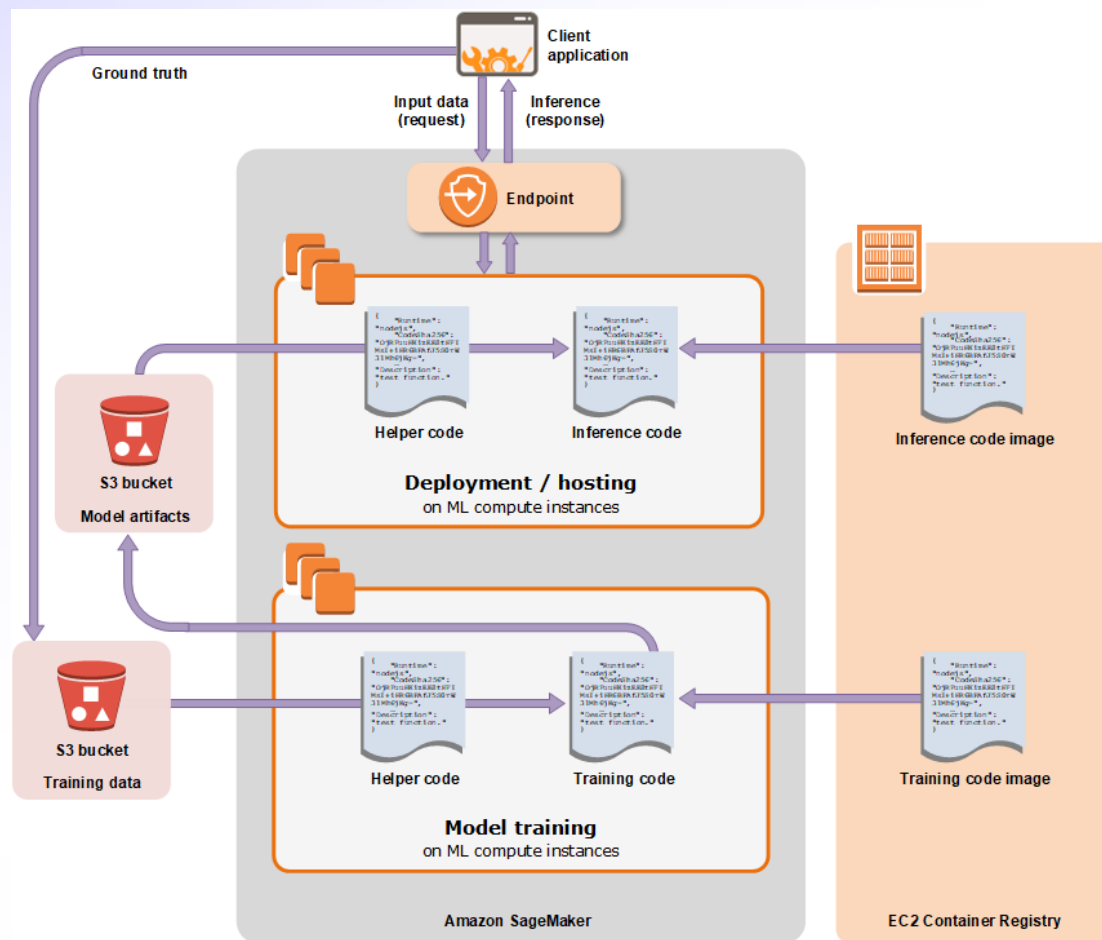
金力鵬

2021/03/22

1. 專題-人臉辨識
2. 專題-物聯網與大數據分析

人臉辨識

架構





温州大学
WENZHOU UNIVERSITY

选取 Notebook instances

The screenshot displays the AWS Management Console interface for Amazon SageMaker. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar shows the 'Amazon SageMaker' menu with options like 'Amazon SageMaker Studio', 'Dashboard', 'Search', 'Ground Truth', and 'Notebook'. The 'Notebook' section is expanded, and 'Notebook instances' is highlighted with a red rectangular box. The main content area shows the 'Amazon SageMaker Studio' dashboard with an overview section containing icons for Ground Truth, Notebook, Training, Inference, and AWS Marketplace.

Amazon SageMaker X

- Amazon SageMaker Studio
- Dashboard
- Search
- ▼ Ground Truth
 - Labeling jobs
 - Labeling datasets
 - Labeling workforces
- ▼ Notebook
 - Notebook instances**
 - Lifecycle configurations
 - Git repositories

Amazon SageMaker > Dashboard

Amazon SageMaker Studio X

Explore Amazon SageMaker Studio, a machine learning Integrated Development Environment (IDE) to build, train and debug models, track experiments, deploy models in production, and monitor drifts.

[Learn more](#)

Overview Hide

- Ground Truth
- Notebook
- Training
- Inference
- AWS Marketplace



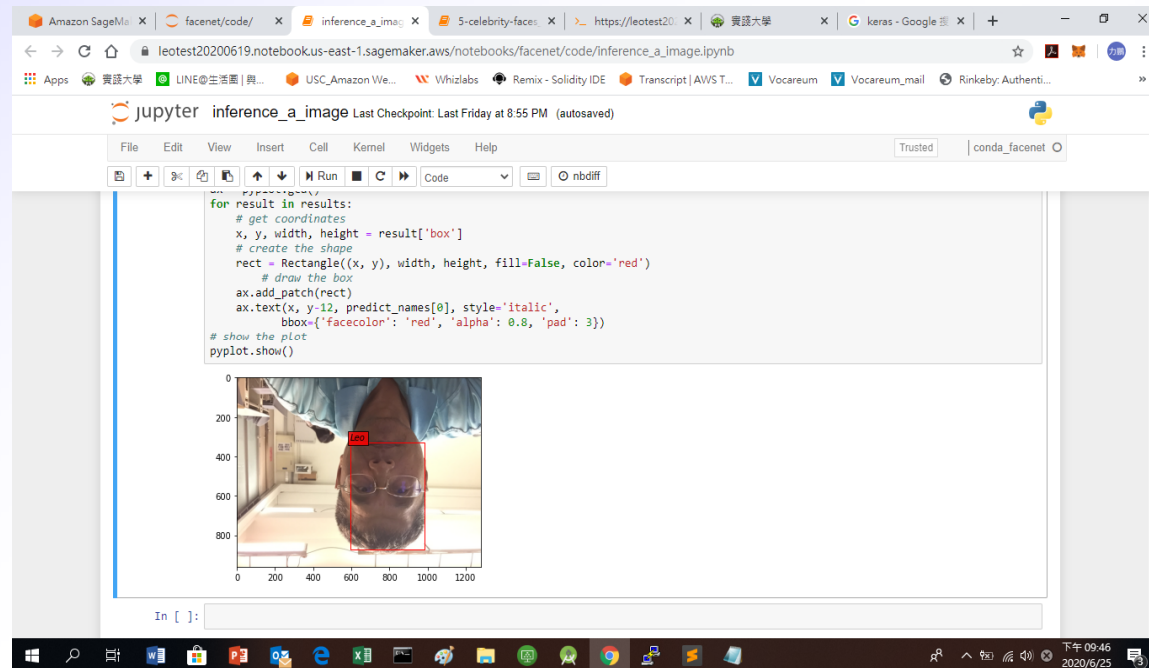
建立環境

- `conda create -n facenet python=3.6 numpy=1.16.1 jupyter`
- `source activate facenet`
- `pip install tensorflow==1.3`
- `pip install keras==2.1.2`
- `pip install opencv-python`
- `pip install scikit-learn scikit-image imageio`
- `pip install mtcnn==0.0.8`
- `pip install matplotlib h5py Pillow`

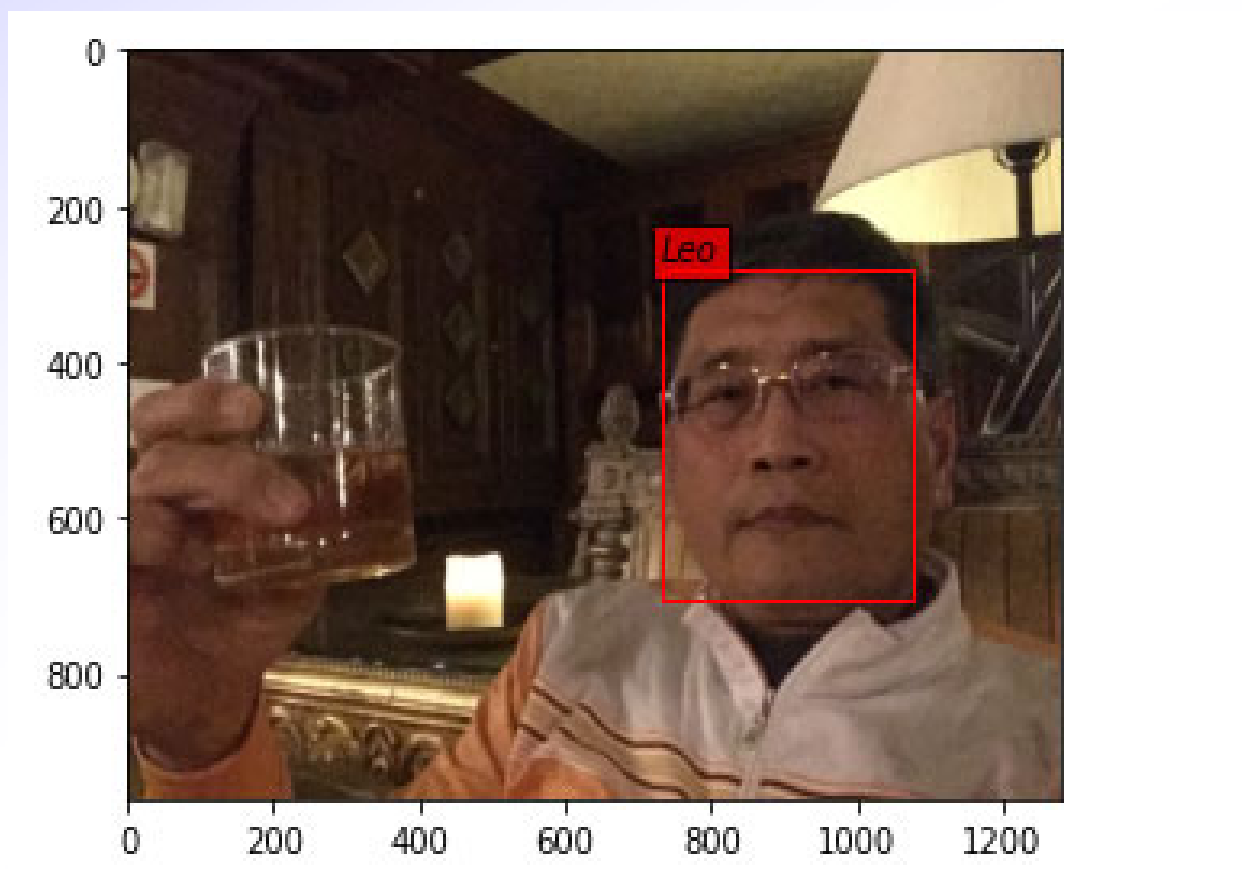


實踐大學
WEN CHIEN UNIVERSITY

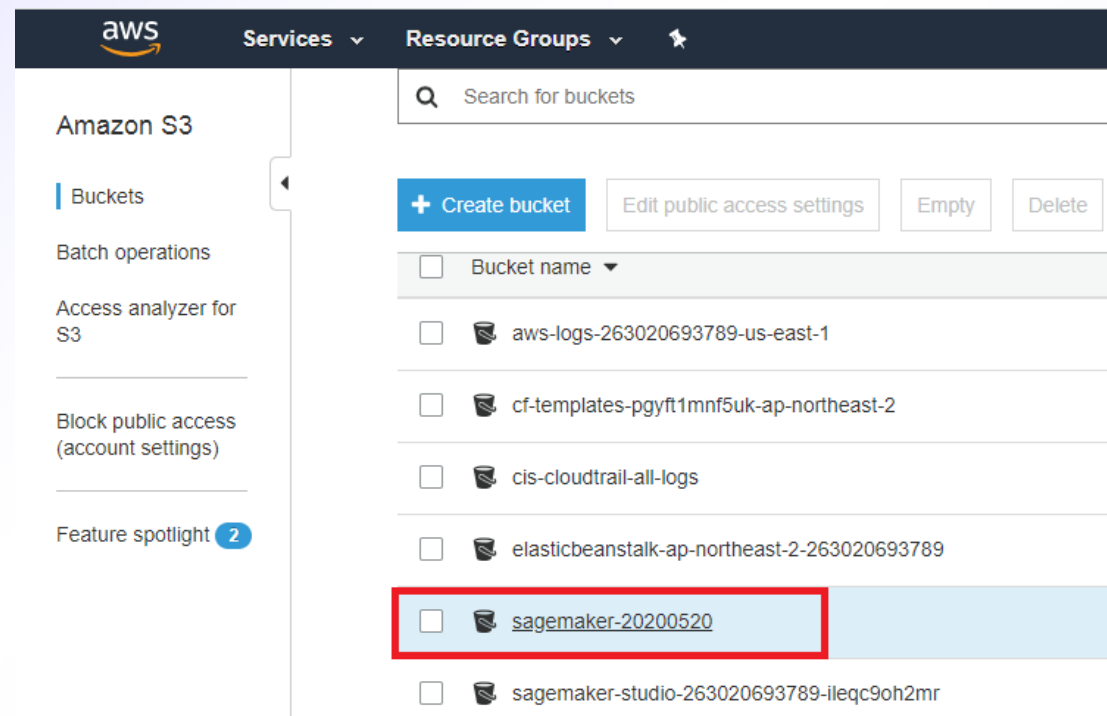
透過Sagemaker執行訓練



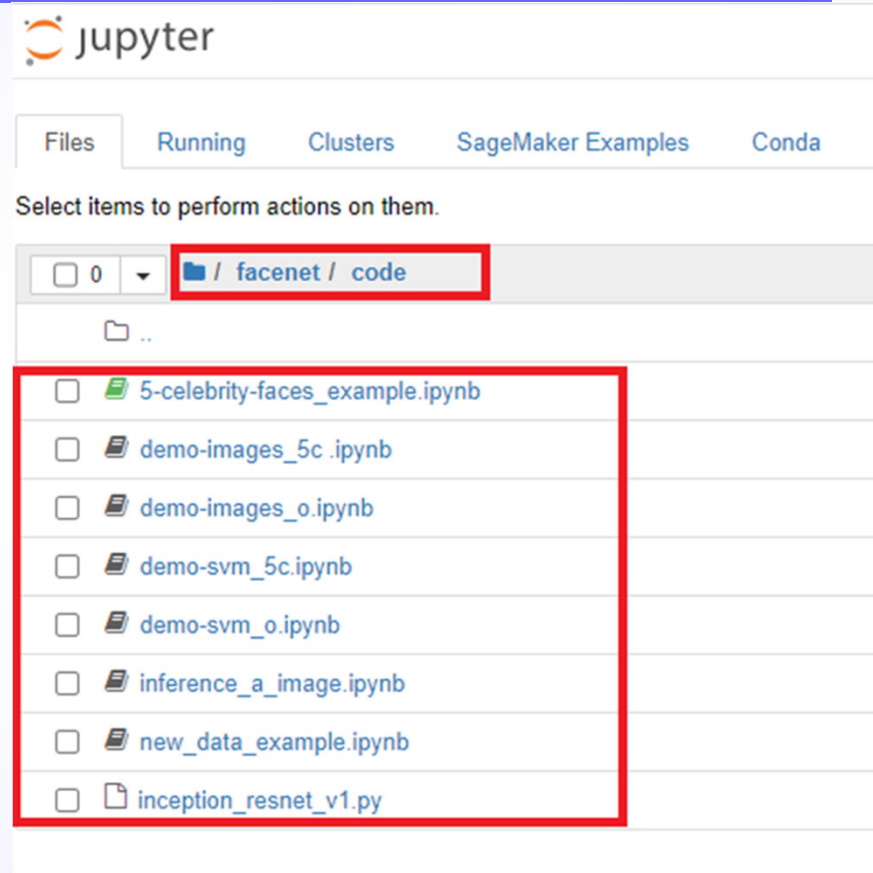
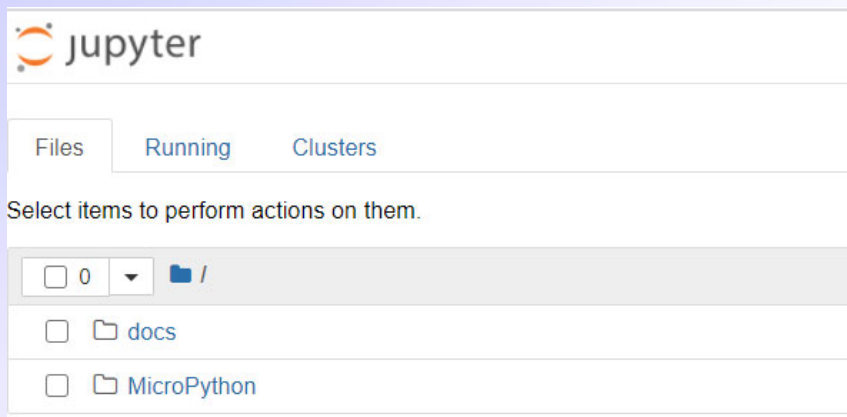
辨識成果



在S3建立共享資料夾



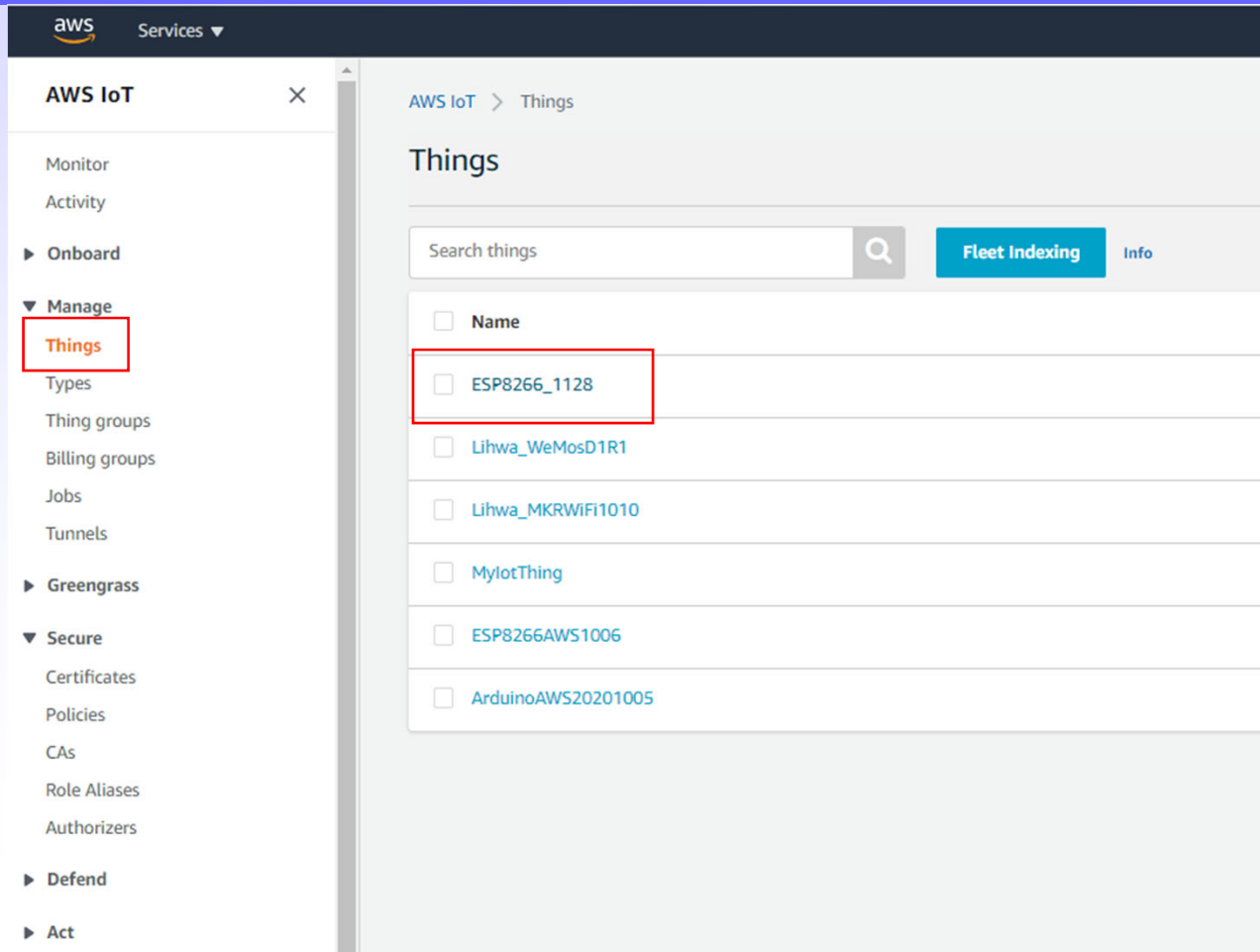
無移植的難度



物聯網

ArduinoESP8266+AWS

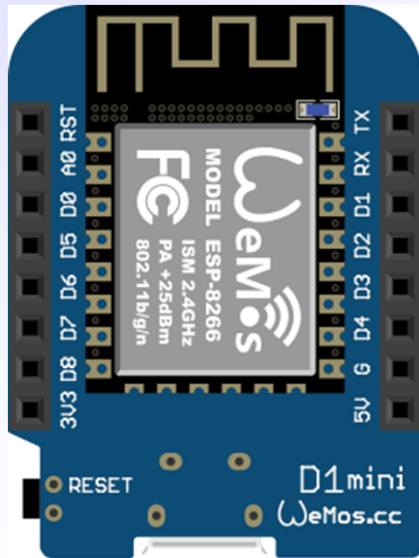
AWS IoT 建立



The screenshot displays the AWS IoT console interface. On the left, the 'AWS IoT' sidebar is visible, with the 'Manage' section expanded and 'Things' highlighted. The main content area shows the 'Things' page, which includes a search bar and a list of IoT things. The first item in the list, 'ESP8266_1128', is highlighted with a red box.

Name
ESP8266_1128
Lihwa_WeMosD1R1
Lihwa_MKRWIFI1010
MyIotThing
ESP8266AWS1006
ArduinoAWS20201005

Arduino 套件



Arduino ESP8266



溫度感應元件

建立MQTT Thing



AWS IoT > Things > Create things

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

Cancel

Create a single thing

下載Certificate

✓ **Success**
Successfully created thing.

✓ **Success**
Successfully generated certificate. Please download certificate files.

Certificate created!

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:

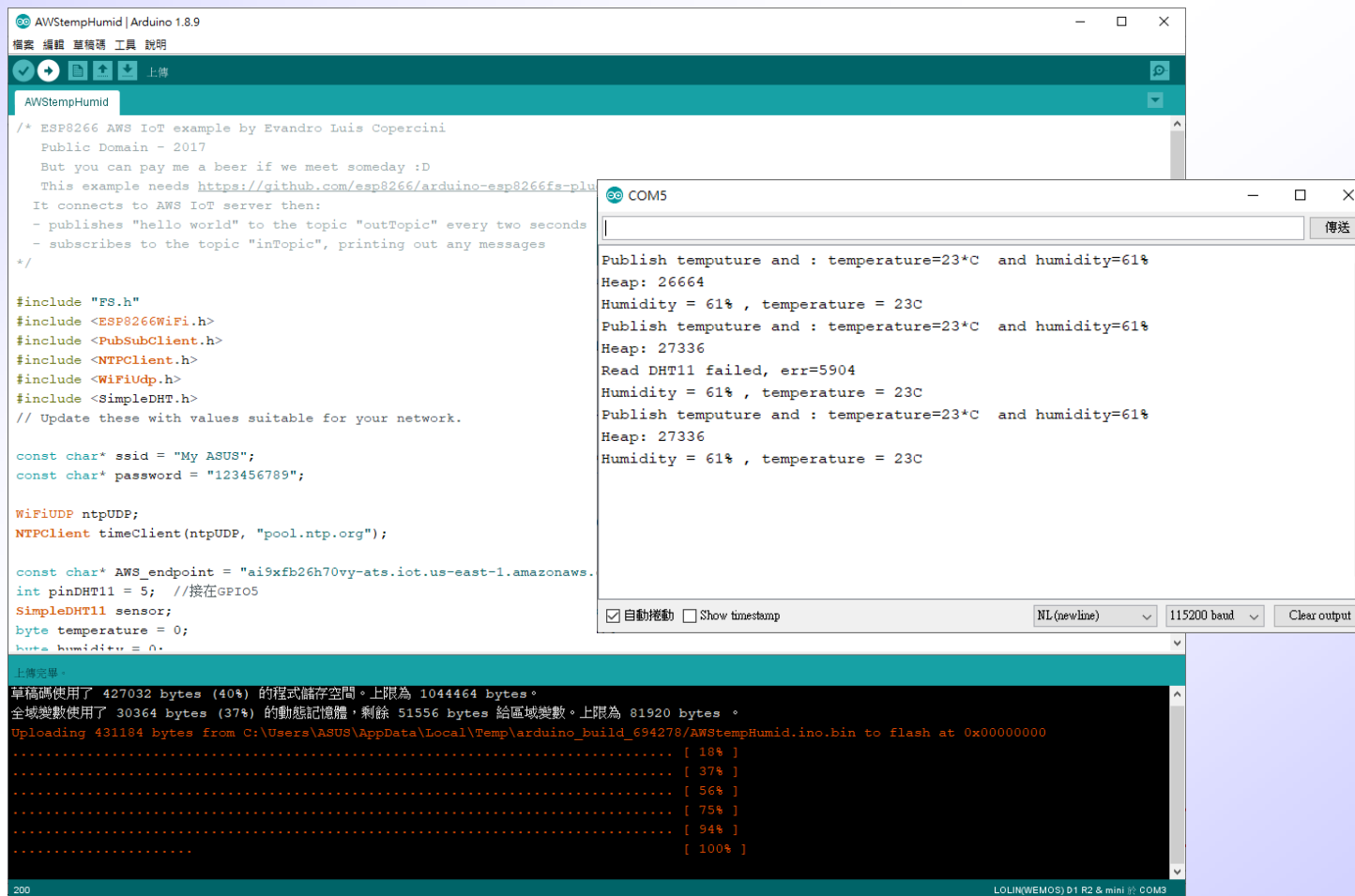
A certificate for this thing	aa1576b047.cert.pem	Download
A public key	aa1576b047.public.key	Download
A private key	aa1576b047.private.key	Download

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

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

[Activate](#)

Arduino IDE



The screenshot displays the Arduino IDE environment. The main editor window shows the 'AWStempHumid' sketch by Evandro Luis Copercini. The sketch includes libraries for WiFi, MQTT, and DHT11, and is configured to connect to an AWS IoT server. The bottom status bar indicates the board is 'LOLIN(WEMOS) D1 R2 & mini' and the port is 'COM5'.

The 'Serial Monitor' window (COM5) shows the output of the sketch, which publishes temperature and humidity data to an MQTT topic every two seconds. The output includes heap memory usage and a DHT11 read error.

```
/* ESP8266 AWS IoT example by Evandro Luis Copercini
Public Domain - 2017
But you can pay me a beer if we meet someday :D
This example needs https://github.com/esp8266/arduino-esp8266fs-plus
It connects to AWS IoT server then:
- publishes "hello world" to the topic "outTopic" every two seconds
- subscribes to the topic "inTopic", printing out any messages
*/

#include "FS.h"
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <SimpleDHT.h>
// Update these with values suitable for your network.

const char* ssid = "My ASUS";
const char* password = "123456789";

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

const char* AWS_endpoint = "ai9xfb26h70vy-ats.iot.us-east-1.amazonaws.com";
int pinDHT11 = 5; //接在GPIO5
SimpleDHT11 sensor;
byte temperature = 0;
byte humidity = 0;

void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println("Starting...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  timeClient.begin();
  sensor.begin(DHT11);
}

void loop() {
  Serial.print("Publish tempature and : temperature=");
  Serial.print(temperature);
  Serial.print(" and humidity=");
  Serial.print(humidity);
  Serial.println();
  Serial.print("Heap: ");
  Serial.print(heap);
  Serial.println();
  Serial.print("Humidity = ");
  Serial.print(humidity);
  Serial.print(", temperature = ");
  Serial.print(temperature);
  Serial.println();
  Serial.print("Publish tempature and : temperature=");
  Serial.print(temperature);
  Serial.print(" and humidity=");
  Serial.print(humidity);
  Serial.println();
  Serial.print("Heap: ");
  Serial.print(heap);
  Serial.println();
  Serial.print("Humidity = ");
  Serial.print(humidity);
  Serial.print(", temperature = ");
  Serial.print(temperature);
  Serial.println();
  delay(2000);
}
```

上一步: 上传完成。
草稿使用了 427032 bytes (40%) 的程式储存空间。上限为 1044464 bytes。
全域变数使用了 30364 bytes (37%) 的动态記憶體。剩餘 51556 bytes 给区域变数。上限为 81920 bytes。
Uploading 431184 bytes from C:\Users\ASUS\AppData\Local\Temp\arduino_build_694278\AWStempHumid.ino.bin to flash at 0x00000000
[18%]
[37%]
[56%]
[75%]
[94%]
[100%]

整合至Arduino 套件

WEMOS - wemos.ino | Arduino 1.8.9

檔案 編輯 草稿碼 工具 說明



wemos

```
#include "FS.h"
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <SimpleDHT.h>
```

匯入需要的程式庫

```
// Update these with values suitable for your network.
```

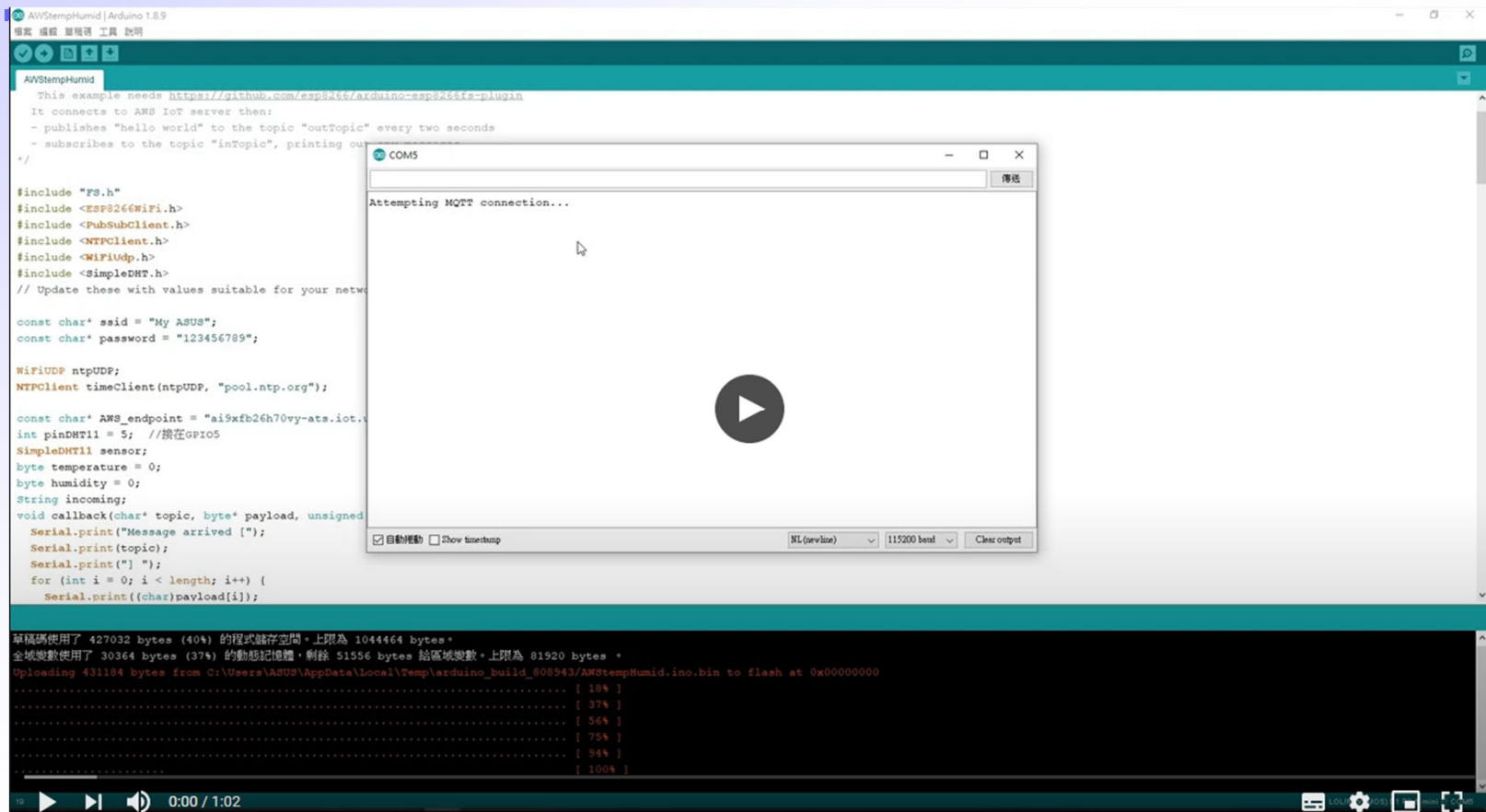
```
const char* ssid = "My ASUS";
const char* password = "123456789";
```

輸入 網路名稱
密碼

```
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
```

```
const char* AWS_endpoint = "ai9xfb26h70vy-ats.iot.us-east-1.amazonaws.com"; //MQTT broker ip
```

詳細資訊請看下一頁



Writes the data from an MQTT message to an AWS S3

The screenshot shows the AWS Management Console interface. At the top, a search bar is visible. On the left sidebar, the 'Services' menu is highlighted with a red box and a circled '1'. Below it, the 'Storage' category is expanded, and the 'S3' service is highlighted with a red box and a circled '2'. In the main content area, the 'S3 Buckets' page is displayed. The 'Create bucket' button is highlighted with a red box and a circled '3'. Below the button, a table lists existing buckets with their names, regions, access status, and creation dates.

Name	Region	Access	Creation date
a-20200005	Asia Pacific (Seoul) ap-northeast-2	Unknown Error	November 4, 2020, 14:34 (UTC+08:00)
aws-logs-263020693789-us-east-1	US East (N. Virginia) us-east-1	Unknown Error	November 25, 2019, 14:55 (UTC+08:00)
cf-templates-pgyft1mnf5uk-ap-northeast-2	Asia Pacific (Seoul) ap-northeast-2	Unknown Error	June 3, 2019, 10:24 (UTC+08:00)
cis-cloudtrail-all-logs	US East (N. Virginia) us-east-1	Unknown Error	June 10, 2019, 21:07 (UTC+08:00)

優點

- 容易擴充功能
- 服務連接方便
- 資訊安全
- 高可靠度
- 容易管理

謝謝聆聽