

# 利用AWS建立Open Weather ETL 並整合至 Apache Airflow

郭益華

**GitHub**

# 目錄

- 1. 簡介說明
- 2. 建立 AWS EC2 並整合至 Local端 Virtual Studio Code 編譯
- 3. Apache Airflow 端口設定
- 4. 建立 AWS S3
- 5. 整合ETL至Apache Airflow
- 6. 遇到Error設定EC2及S3 的IAM policy
- 7. 修正error重新執行ETL

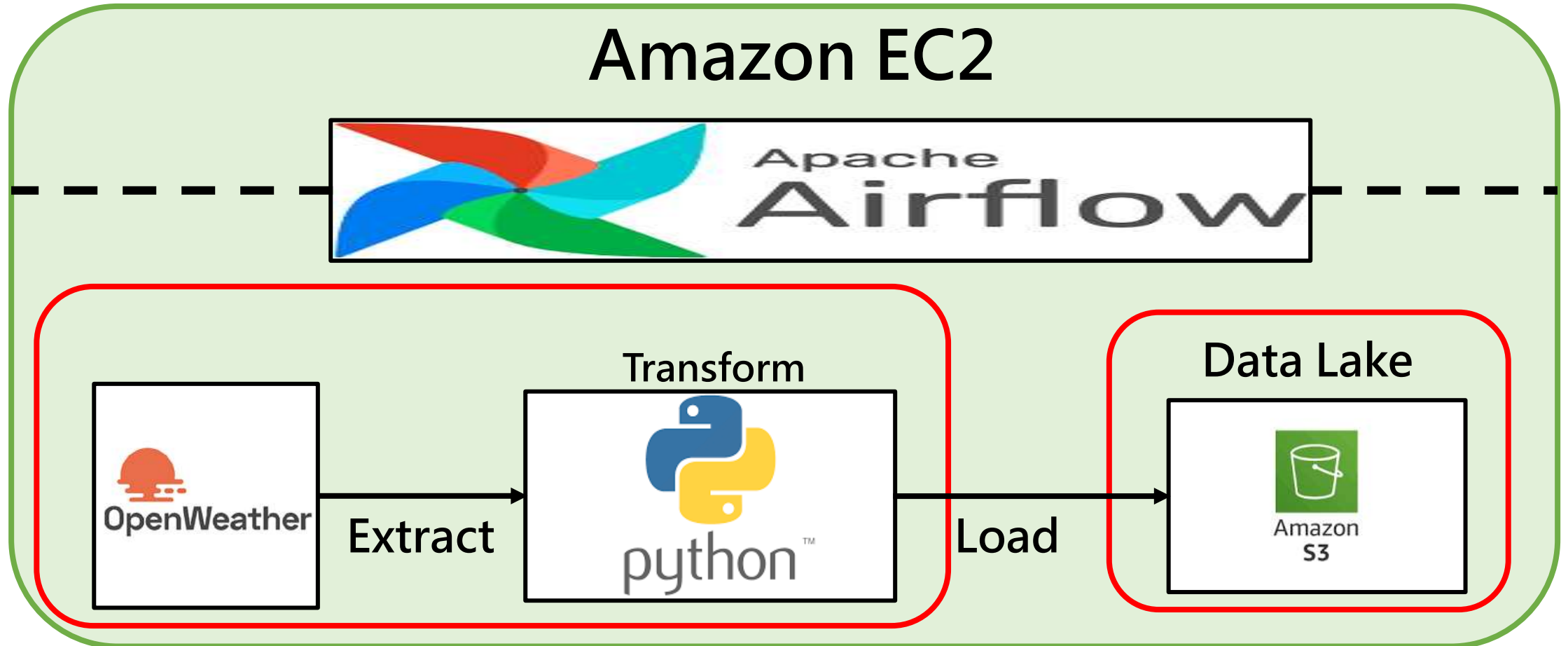
# 1. 簡介說明

# 實作說明

## 利用AWS開發自動化抓取OpenWeather天氣資訊的ETL

- 建立 AWS EC2 並整合至 Local端 Virtual Studio Code 編譯
- Apache Airflow 端口設定
- 建立 AWS S3
- 整合ETL至Apache Airflow
- 遇到Error設定EC2及S3 的IAM policy
- 修正error重新執行ETL

# Flow



# 前置準備(1/2)

- 註冊open\_weather帳號獲得API Keys

<https://openweathermap.org/>

- 註冊AWS 帳號

<https://aws.amazon.com/tw/>

- 建立一個 EC2(後面有建立EC2教學)

# 前置準備(2/2)

- AWS EC2 Ubuntu需安裝的套件:

```
sudo apt update
```

```
sudo apt install python3-pip
```

```
sudo pip install pandas
```

```
sudo pip install s3fs
```

```
sudo pip install apache-airflow
```

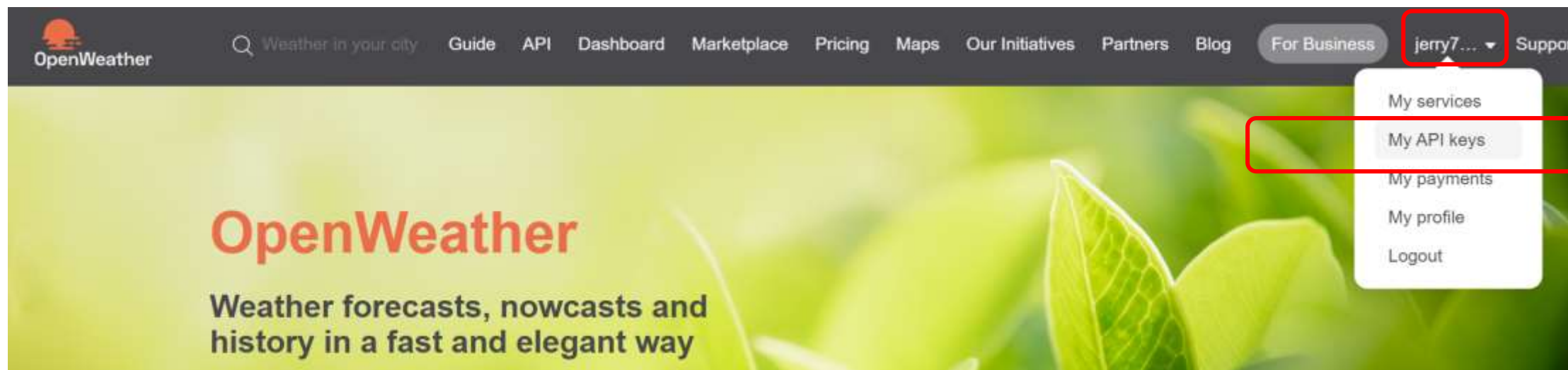
# 實務知識

- Apache Airflow
- AWS EC2 建立
- AWS S3 建立
- AWS port 設定
- AWS security policy 設定
- 利用 AWS 建立 ETL
- 將AWS EC2 整合到 Local端 Visual Studio Code編譯



# OpenWeather

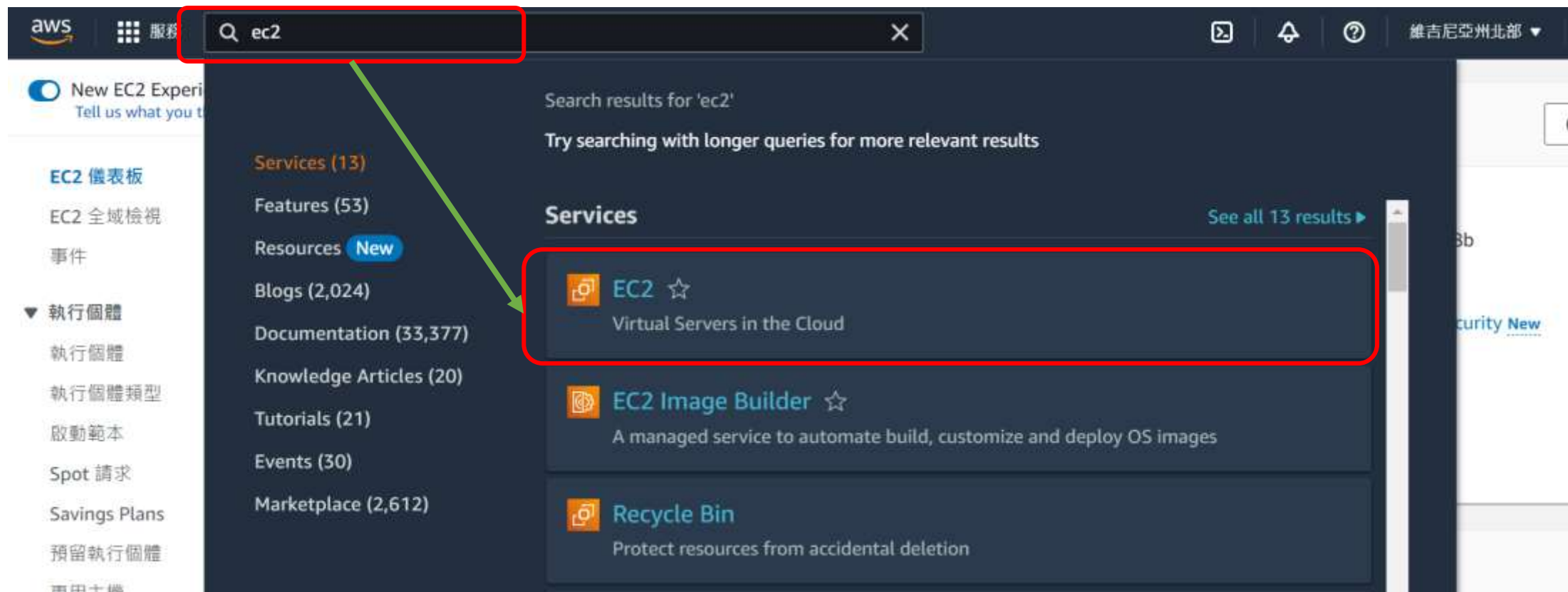
註冊完畢點選 My API keys 即可獲得金鑰



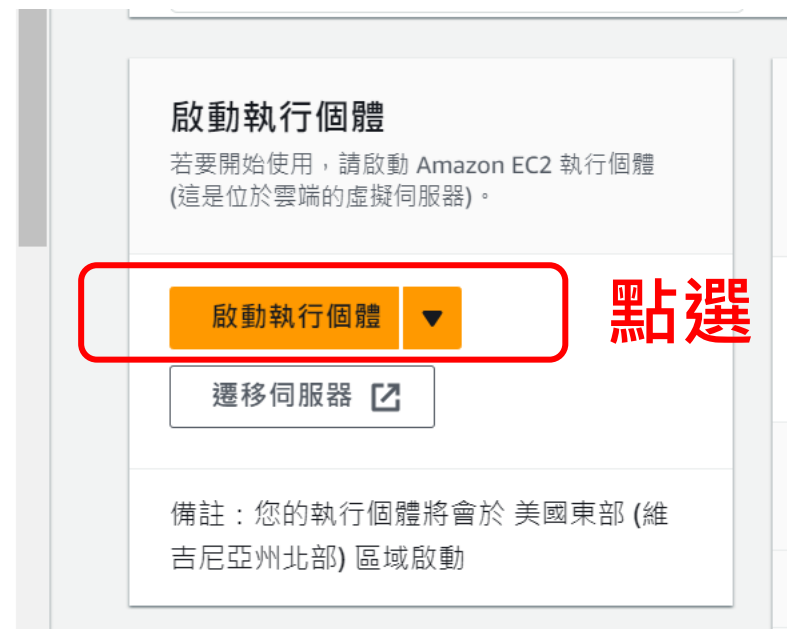
## 2. 建立 AWS EC2 並整合至 Local 端 Visual Studio Code 編譯

# 搜尋 EC2

## DashBoard搜尋 EC2



# 點選後進入畫面



# 為 EC2 命名

[EC2](#) > [執行個體](#) > 啟動執行個體

## 啟動執行個體 [資訊](#)

Amazon EC2 可讓您建立在 AWS 雲端上執行的虛擬機器或執行個體。請按下方的簡易步驟快速開始使用。

### 名稱和標籤 [資訊](#)

名稱

[新增其他標籤](#)

▼ 應用程式和作業系統映像 (Amazon Machine Image) [資訊](#)

# 選擇作業系統

這裡選擇 Ubuntu

快速入門

Amazon Linux  
aws

macOS  
Mac

**Ubuntu**  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

瀏覽更多 AMI  
包括來自 AWS、Marketplace 和社群的 AMI

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type 符合免費方案資格  
ami-053b0d53c279acc90 (64 位元 (x86)) / ami-0a0c8eebcdd6dcdbd0 (64 位元 (ARM))  
虛擬化: hvm 已啟用 ENA: true 根裝置類型: ebs

描述

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

架構

AMI ID

64 位元 (x86) ▼

ami-053b0d53c279acc90

已驗證的供應商

# 選擇規格

實作ETL需選擇

t2.small規格

否則後續會有記憶體不足的情形發生

如果只是實作

連線到 Local端 Visual Studio Code

選擇 t2.micro 免費規格即可

▼ 執行個體類型 [資訊](#)

執行個體類型

t2.micro

符合免費方案資格

系列: t2 1 vCPU 1 GiB 記憶體 目前世代: true

隨需 Windows base 定價: 0.0162 USD 每小時

隨需 SUSE base 定價: 0.0116 USD 每小時

隨需 RHEL base 定價: 0.0716 USD 每小時 隨需 Linux base 定價: 0.0116 USD 每小時

Additional costs apply for AMIs with pre-installed software

☐ 所有世代

[比較執行個體類型](#)

▼ 金鑰對 (登入) [資訊](#)

您可以使用金鑰對安全地連線到您的執行個體。在啟動執行個體之前，請確定您有權存取所選取的金鑰對。

金鑰對名稱 - 必要

選取

▼

[建立新的金鑰對](#)

# 建立金鑰

▼ 執行個體類型 [資訊](#)

執行個體類型

t2.micro

符合免費方案資格

系列: t2 1 vCPU 1 GiB 記憶體 目前世代: true  
隨需 Windows base 定價: 0.0162 USD 每小時  
隨需 SUSE base 定價: 0.0116 USD 每小時  
隨需 RHEL base 定價: 0.0716 USD 每小時 隨需 Linux base 定價: 0.0116 USD 每小時

所有世代

比較執行個體類型

Additional costs apply for AMIs with pre-installed software

▼ 金鑰對 (登入) [資訊](#)

您可以使用金鑰對安全地連線到您的執行個體。在啟動執行個體之前，請確定您有權存取所選取的金鑰對。

金鑰對名稱 - 必要

選取

▼

↻ 建立新的金鑰對

建立金鑰對

×

金鑰對名稱

金鑰對可讓您安全地連線到您的執行個體。

open\_weather\_ETL

名稱最多可包含 255 個 ASCII 字元，不能包含前置或尾端空格。

金鑰對類型

☒ RSA

RSA 加密的私有和公有金鑰對

☐ ED25519

ED25519 加密的私有和公有金鑰對

私有金鑰檔案格式

☒ .pem

搭配 OpenSSH 使用

☐ .ppk

搭配 PuTTY 使用

⚠ 出現提示時，請將私有金鑰存放在電腦上安全且可存取的位置。您稍後將需要使用此資訊來連線到執行個體。 [進一步了解](#)


取消

建立金鑰對



# 產生金鑰

✓ 今天

<input checked="" type="checkbox"/> 	open_weather_ETL.pem	2023/9/14 上午 09:30	PEM 檔案	2 KB
---	----------------------	--------------------	--------	------

[查看詳情](#)

**請將此檔案妥善儲存，勿遺失**

▼ 設定儲存 資訊 進階

1x 8 GiB gp2 根磁碟區 (未加密)

符合免費方案資格的客戶可獲得最多 30 GB 的 EBS 一般用途 (SSD) 或磁性儲存空間

新增新磁碟區

選取的 AMI 包含比執行個體允許的數量更多的執行個體存放磁碟區。只有來自 AMI 的第一個 0 執行個體存放磁碟區才能從執行個體存取

0 x 檔案系統 編輯

▼ 網路設定 資訊 編輯

網路 資訊  
vpc-0aaa6ecc293a2118b

子網路 資訊  
沒有進行設定 (任何可用區域中的預設子網路)

自動指派公有 IP 資訊  
啟用

防火牆 (安全群組) 資訊  
安全群組是一組防火牆規則，可控制執行個體的流量。新增規則以允許特定流量到達您的執行個體。

建立安全群組 選擇現有的安全群組

我們將建立名為 'launch-wizard-1' 的新安全群組，其中包含下列規則：

- ☒ 允許 SSH 流量，來自 隨處 0.0.0.0/0
- ☒ 允許來自網際網路的 HTTPS 流量  
若要設定端點，例如建立 Web 伺服器端
- ☒ 允許來自網際網路的 HTTP 流量  
若要設定端點，例如建立 Web 伺服器端


全部勾選

啟動執行個體

檢視命令

# 成功建立 EC2

[EC2](#) > [執行個體](#) > 啟動執行個體

 成功  
已成功啟動執行個體 ([i-003973a3ae2df453b](#))

▶ 啟動日誌

後續步驟

執行個體 (1) 資訊							
<input type="text" value="依屬性或標籤 (case-sensitive) 尋找 執行個體"/>				< 1 >		⚙️	
<input type="checkbox"/>	Name ▼	執行個體 ID	執行個體狀態 ▼	執行個體類型 ▼	狀態檢查	警示狀態	可
<input type="checkbox"/>	open_weather...	<a href="#">i-003973a3ae2df453b</a>	✔️ 執行中	t2.micro	🕒 正在初始化	0 in alarm +	us

# 點選連線

執行個體 (1/1) 資訊

🔄 連線 執行個體狀態 ▼ 動作 ▼ 啟動新執行個體 ▼

🔍 依屬性或標籤 (case-sensitive) 尋找 執行個體

<input checked="" type="checkbox"/>	Name ▼	執行個體 ID	執行個體狀態 ▼	執行個體類型 ▼	狀態檢查	警示狀態	可
<input checked="" type="checkbox"/>	open_weather...	i-003973a3ae2df453b	✅ 執行中	t2.micro	🕒 正在初始化	0 in alarm	+

執行個體 : i-003973a3ae2df453b (open\_weather\_ETL)

詳細資訊 | 安全性 | 聯網 | 儲存 | 狀態檢查 | 監控 | 標籤

▼ 執行個體摘要 資訊

執行個體 ID	公有 IPv4 地址	私有 IPv4 地址
📄 i-003973a3ae2df453b (open_weather_ETL)	📄 54.158.100.87   <a href="#">開啟地址</a>	📄 172.31.43.27

# 點選SSH用戶端

EC2 > 執行個體 > i-003973a3ae2df453b > 連線至執行個體

## 連線至執行個體 資訊

使用任何這些選項連線至執行個體 i-003973a3ae2df453b (open\_weather\_ETL)

EC2 Instance Connect

Session Manager

**SSH 用戶端**

EC2 序列主控台

執行個體 ID

 i-003973a3ae2df453b (open\_weather\_ETL)

1. 開啟 SSH 用戶端。
2. 尋找私有金鑰檔案。用於啟動此執行個體的金鑰是 open\_weather\_ETL.pem
3. 如有必要，請執行此命令，以確保您的金鑰無法公開檢視。  
 `chmod 400 open_weather_ETL.pem`
4. 使用 公有 DNS 連線至執行個體：  
 `ec2-54-158-100-87.compute-1.amazonaws.com`

範例：

 `ssh -i "open_weather_ETL.pem" ubuntu@ec2-54-158-100-87.compute-1.amazonaws.com`

**將此複製**

# 開啟終端機測試連線

金鑰路徑須根據自身存放路徑自行修改

```
PS C:\Users\jerry> ssh -i "C:\Users\jerry\Desktop\master course\dataEngineer\openWeatherETL\open_weather_ETL.pem" ubuntu@ec2-54-158-100-87.compute-1.amazonaws.com
The authenticity of host 'ec2-54-158-100-87.compute-1.amazonaws.com (54.158.100.87)' can't be established.
ED25519 key fingerprint is SHA256:uk9rZDsvu7Aw0/ZP+8w0twzBsA935HXesKBd1/sHrPE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-158-100-87.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

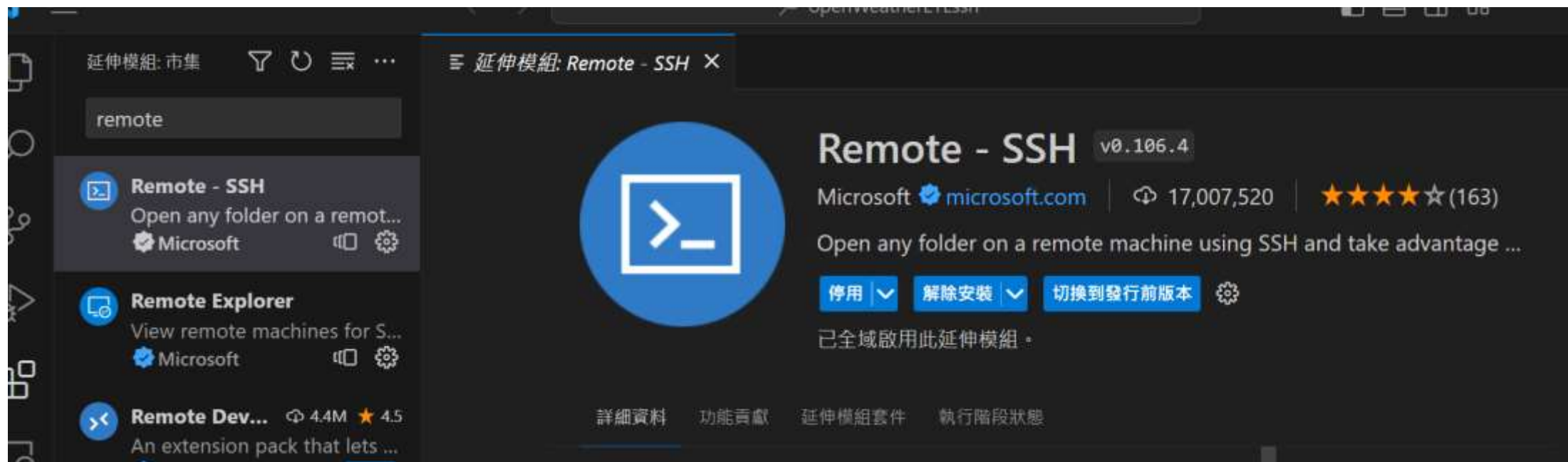
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

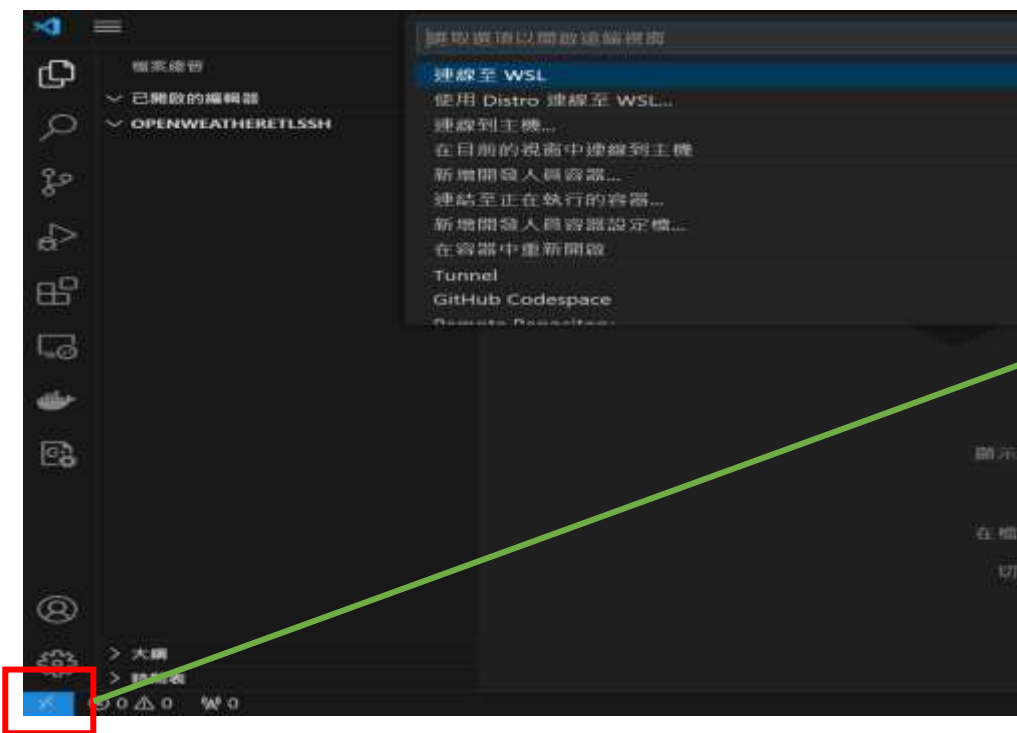
```
ubuntu@ip-172-31-43-27:~$
```

成功連線畫面

# 整合至Local端 Visual Studio Code

開啟 Visual Studio Code 並下載 延伸模組 Remote SSH

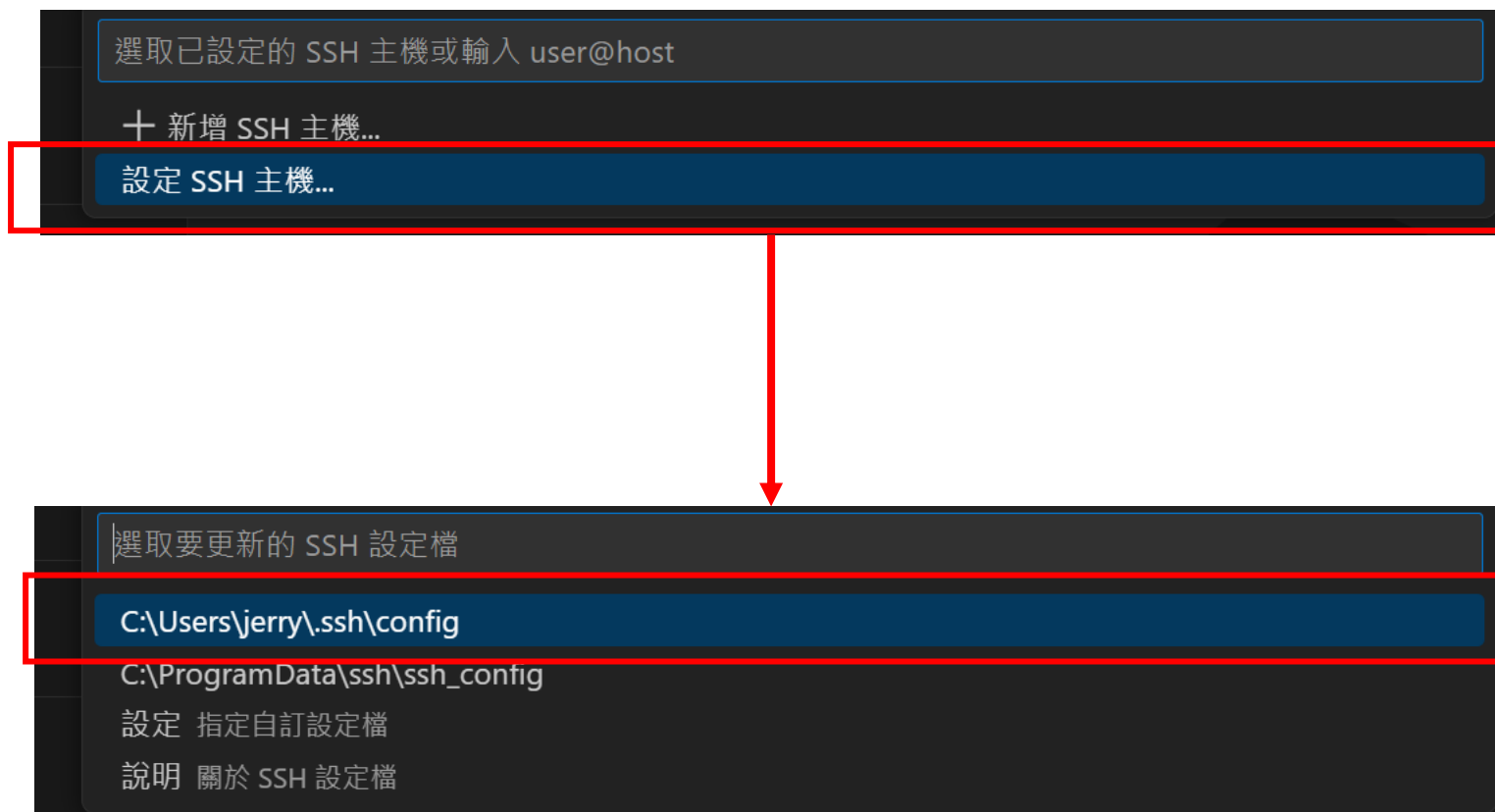




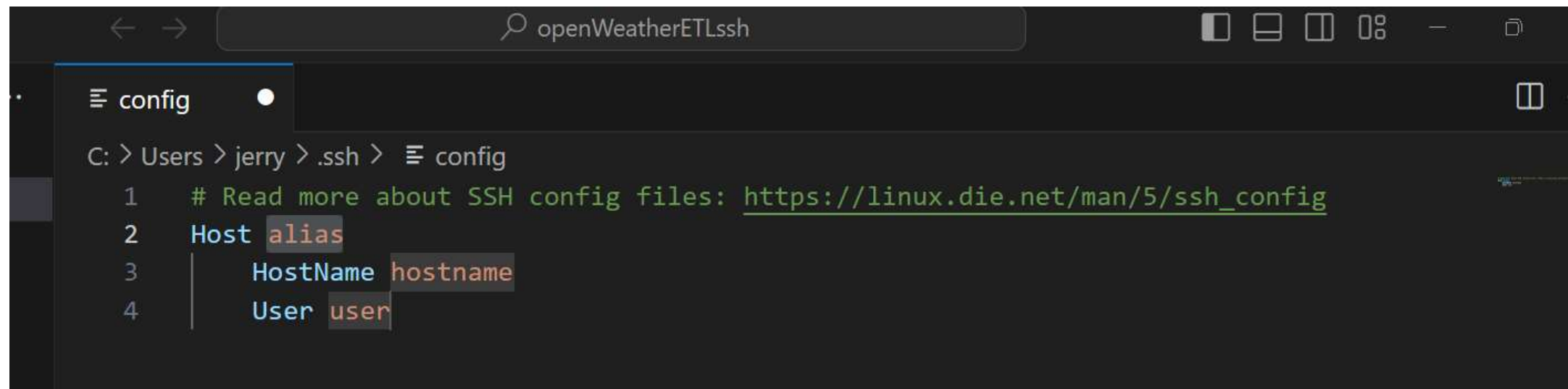
點選左下角藍色選項



# 繼續點選



# 修改此檔案內容

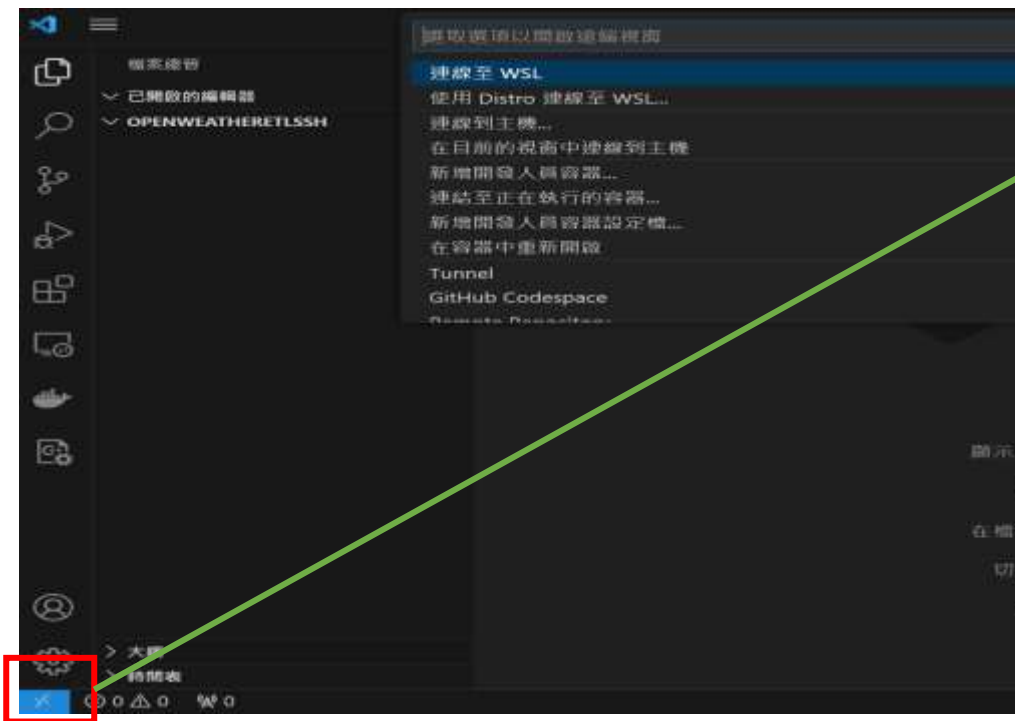


The image shows a code editor window with a dark theme. The title bar at the top contains navigation arrows, a search icon, and the text "openWeatherETLssh". Below the title bar, a tab labeled "config" is active. The main editing area displays the following text:

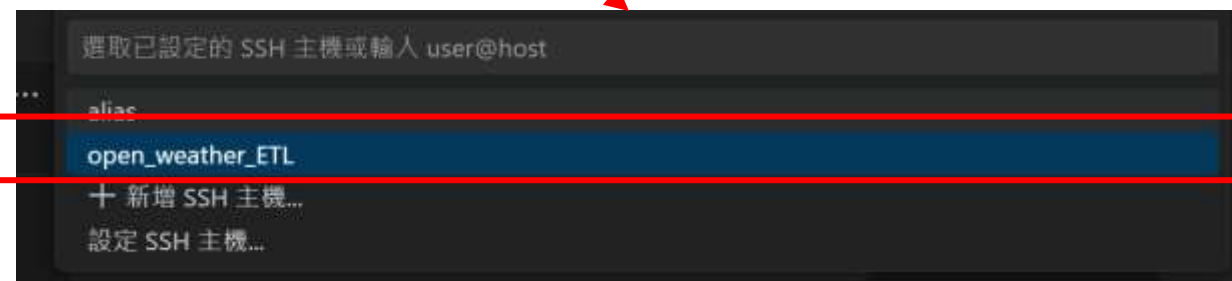
```
C: > Users > jerry > .ssh > config
1  # Read more about SSH config files: https://linux.die.net/man/5/ssh\_config
2  Host alias
3      HostName hostname
4      User user
```

```
config
C: > Users > jerry > .ssh > config
1  # Read more about SSH config files: https://linux.die.net/man/5/ssh\_config
2  Host alias
3      HostName hostname
4      User user
5
6  Host open_weather_ETL
7      HostName 54.158.100.87
8      User ubuntu
9      IdentityFile "C:openWeatherETL\open_weather_ETL.pem"
10
```

Host: EC2 名稱  
HostName: EC2 公有 IPv4 地址  
User: 作業系統  
IdentityFile: 金鑰路徑

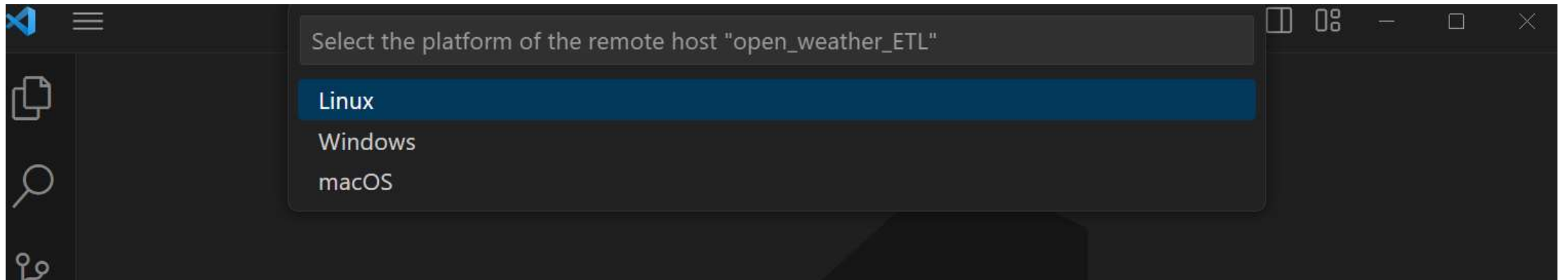


點選左下角藍色選項



- 可看到已經有我們所設定的主機名稱
- 點選即可

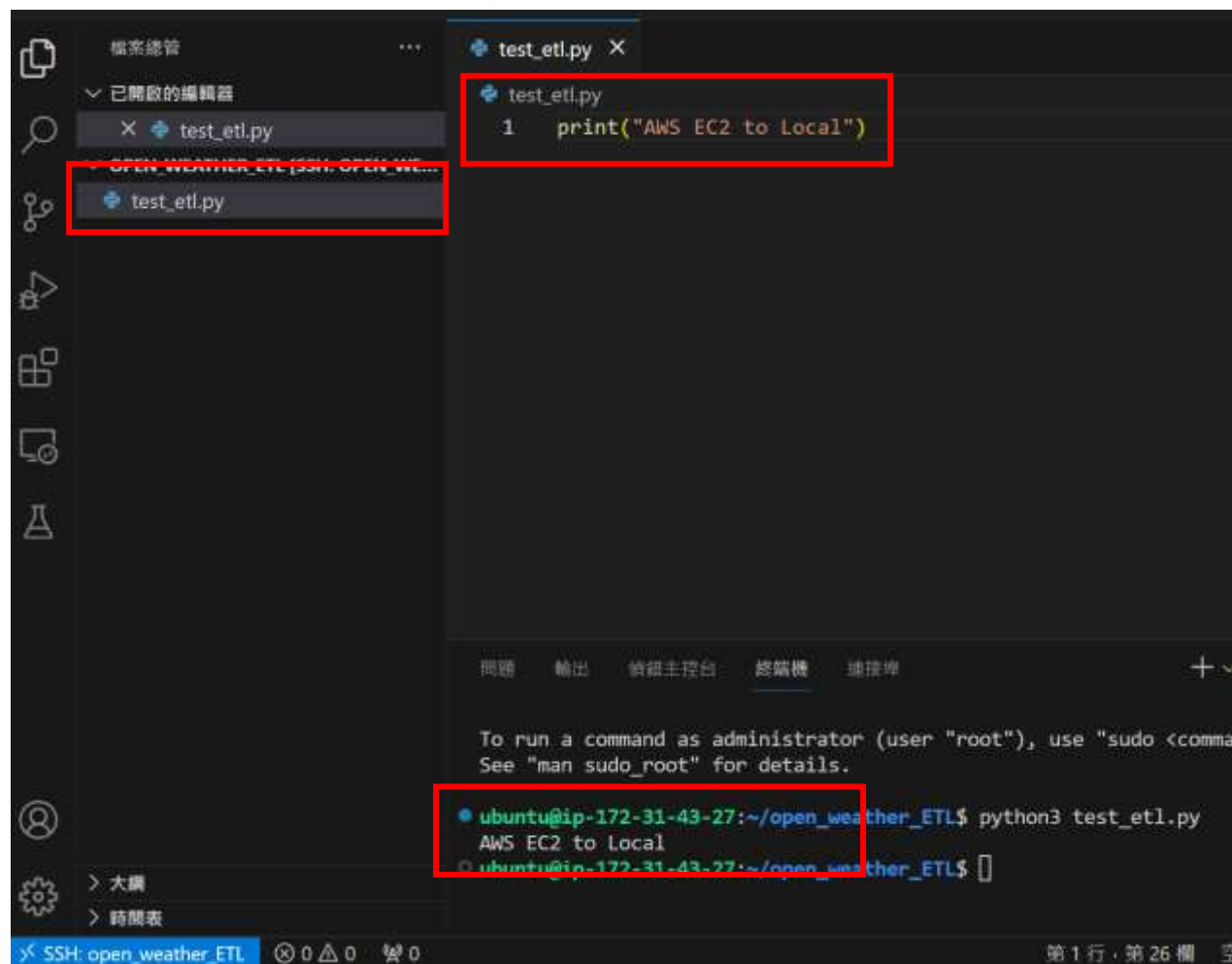
# 點選Linux



# 整合連線成功



# 測試

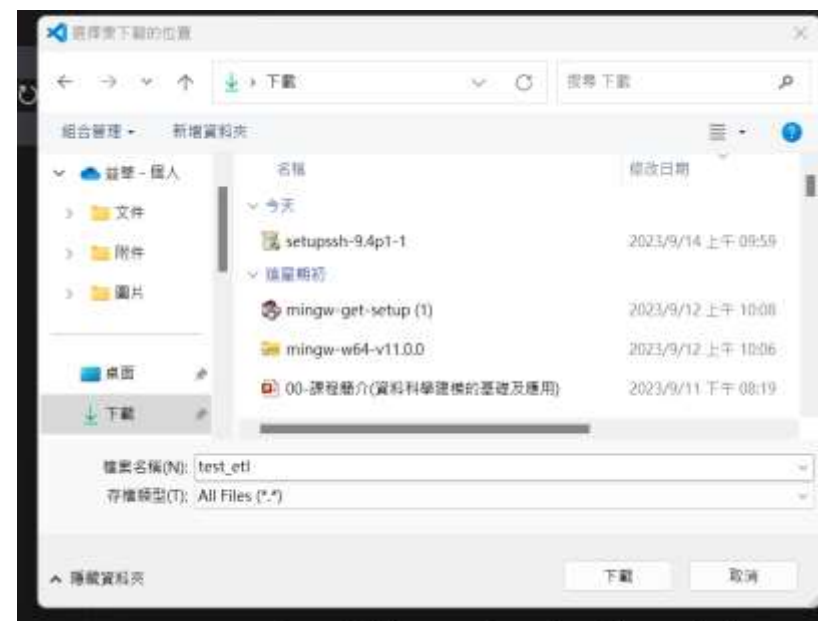
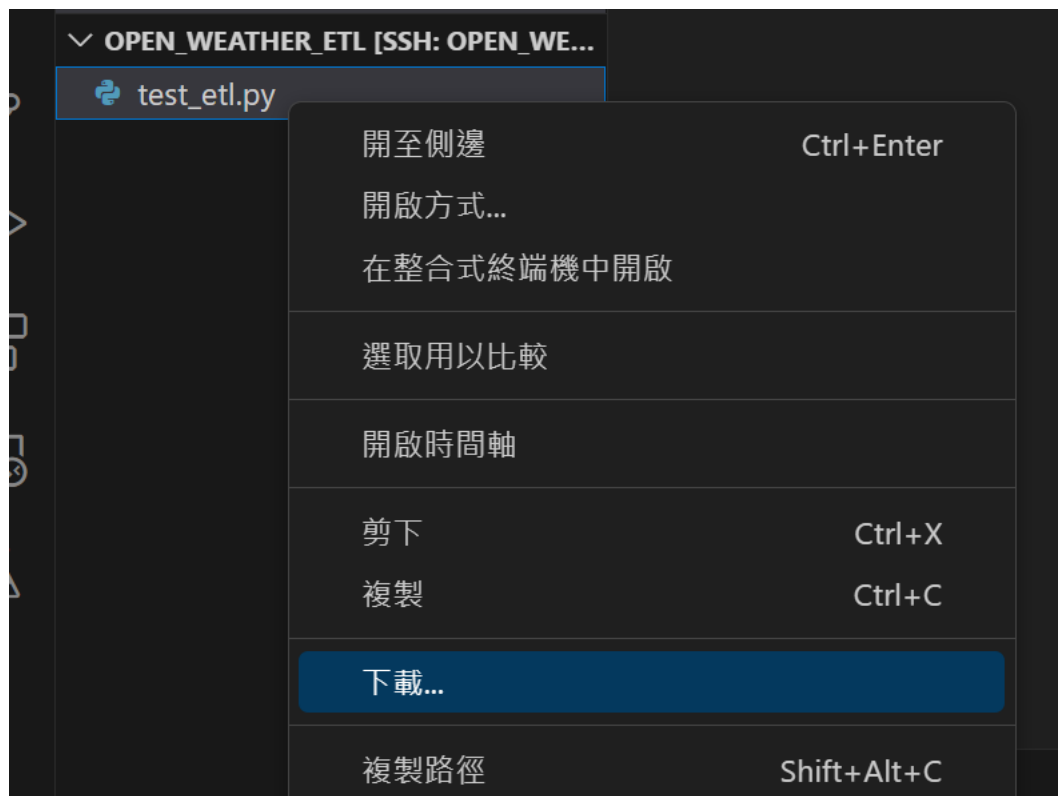


The screenshot shows a code editor interface with a sidebar on the left and a main editor area. The sidebar contains a file explorer with a red box highlighting the file `test_etl.py`. The main editor area shows the content of `test_etl.py`, which is a single line of Python code: `print("AWS EC2 to Local")`. This line is also highlighted with a red box. Below the editor, there is a terminal window showing the command `python3 test_etl.py` being executed, and the output `AWS EC2 to Local`. The terminal output is also highlighted with a red box. The status bar at the bottom indicates the file is `SSH: open_weather_ETL` and shows the current cursor position as `第 1 行, 第 26 欄`.

```
test_etl.py
1 print("AWS EC2 to Local")

ubuntu@ip-172-31-43-27:~/open_weather_ETL$ python3 test_etl.py
AWS EC2 to Local
ubuntu@ip-172-31-43-27:~/open_weather_ETL$
```

# 也可將程式碼下載至Local端



點選程式檔案，並且按滑鼠右鍵 > 下載



# 注意事項



每重新啟動一次EC2皆須修改HostName  
才可成功連線

```
main.py x config x
C: > Users > jerry > .ssh > config
1 # Read more about SSH config files: https://linux.die.net/man/5/ssh\_config
2 Host alias
3     HostName hostname
4     User user
5
6 Host open_weather_ETL
7     HostName 54.224.176.161
8     User ubuntu
9     IdentityFile "C:\Users\jerry\Desktop\master course\dataEngineer\openWeatherETL\open_w
10
```

### 3. Apache Airflow 端口設定

# 回到執行個體區

勾選 EC2

點選 安全性

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, there's a header for '執行個體 (1/2) 資訊' (Instances (1/2) Information) with buttons for '連線' (Connect), '執行個體狀態' (Instance State), '動作' (Actions), and '啟動新執行個體' (Launch New Instance). Below this is a search bar and a table of instances. The first instance, 'open\_weather...', is selected, and its details are shown below the table. The '安全性' (Security) tab is active, showing the '安全詳細資訊' (Security Details) section. The 'IAM 角色' (IAM Role) is highlighted, and the '安全群組' (Security Group) is also highlighted. Blue arrows indicate the flow from the instance selection to the security details and then to the security group.

Name	執行個體 ID	執行個體狀態	執行個體類型	狀態檢查	警示狀態
<input checked="" type="checkbox"/> open_weather...	i-003973a3ae2df453b	已停止	t2.micro	-	0 in alarm
<input type="checkbox"/> opweather_etl	i-06b5f2b178d17ec67	已停止	t2.small	-	0 in alarm

執行個體 : i-003973a3ae2df453b (open\_weather\_ETL)

詳細資訊 | **安全性** | 聯網 | 儲存 | 狀態檢查 | 監控 | 標籤

▼ 安全詳細資訊

IAM 角色: -

擁有者 ID: 288142612271

啟動時間: Thu Sep 14 2023 18:37:10 GMT+0800 (GMT+08:00)

安全群組: sg-0e109e30f9a4fac4f (launch-wizard-1)

記住此編號

# 回到 EC2 點選 安全群組

## ▼ 網路和安全

安全群組

彈性 IP

配置群組

金鑰對

網路界面

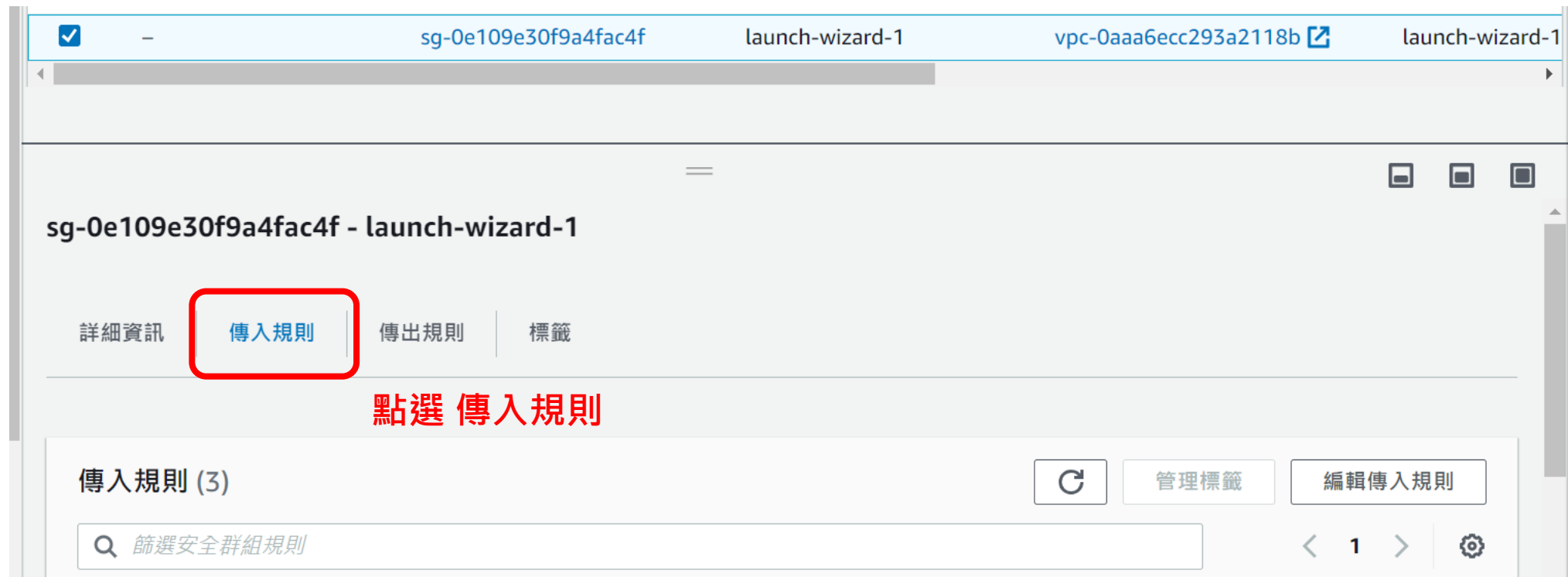
## 啟動執行個體

若要開始使用，請啟動 Amazon EC2 執行個體  
(這是位於雲端的虛擬伺服器)。

啟動執行個體



# 勾選與前面相符的安全組編號



# 自訂 8080 連接，來源 選 隨機ipv4

安全群組規則 ID	類型 資訊	通訊協定 資訊	連接埠範圍 資訊	來源 資訊	描述 - 選用 資訊	
sgr-0d1e6916a47101238	HTTPS ▼	TCP	443	自訂 ▼	<input type="text" value="Q"/> 0.0.0.0/0 ✕	<input type="text"/> 刪除
sgr-06065dc319c1317e4	SSH ▼	TCP	22	自訂 ▼	<input type="text" value="Q"/> 0.0.0.0/0 ✕	<input type="text"/> 刪除
sgr-0e632b6a6c9624c71	HTTP ▼	TCP	80	自訂 ▼	<input type="text" value="Q"/> 0.0.0.0/0 ✕	<input type="text"/> 刪除
-	自訂 TCP ▼	TCP	8080	隨機... ▼	<input type="text" value="Q"/> 0.0.0.0/0 ✕	<input type="text"/> 刪除

新增規則

# 新增成功畫面

Q 依屬性或標籤 (case-sensitive) 尋找 執行個體

<input checked="" type="checkbox"/>	Name ▾	執行個體 ID	執行個體狀態 ▾	執行個體類型 ▾	狀態檢查	警示狀態	可
<input checked="" type="checkbox"/>	open_weather...	i-003973a3ae2df453b	🟢 執行中	t2.micro	🟢 2/2 項檢查通過	0 in alarm	+

執行個體 : i-003973a3ae2df453b (open\_weather\_ETL)

Q 篩選規則

名稱	安全群組規則 ID	連接埠範圍	通訊協定	來源
-	sgr-0d1e6916a47101238	443	TCP	0.0.0.0/0
-	sgr-06065dc319c1317e4	22	TCP	0.0.0.0/0
-	sgr-0e632b6a6c9624c71	80	TCP	0.0.0.0/0
-	sgr-05c4e4612d03d4769	8080	TCP	0.0.0.0/0

# 啟動 Airflow

```
ubuntu@ip-172-31-43-27: ~  
ubuntu@ip-172-31-43-27:~$ airflow standalone
```

於所開啟連線的 AWS EC2 ubuntu中 輸入指令 `airflow standalone` 即可啟動airflow

```
e |  
e | Airflow is ready  
e | Login with username: admin password: mSkYBAbg6Z9nTeFe  
e | Airflow Standalone is for development purposes only. Do not use this in production!
```

成功啟動會給予一組預設的帳號密碼



# 查看 EC2 的 公有 IPv4 DNS 並複製

<input checked="" type="checkbox"/>	Name	執行個體 ID	執行個體狀態	執行個體類型	狀態檢查	警示狀態	可
<input checked="" type="checkbox"/>	open_weather...	i-003973a3ae2df453b	✔ 執行中	t2.micro	✔ 2/2 項檢查通過	0 in alarm	+ us-

### 執行個體 : i-003973a3ae2df453b (open\_weather\_ETL)

執行個體 ID	公有 IPv4 地址	私有 IPv4 地址
i-003973a3ae2df453b (open_weather_ETL)	34.207.55.219   <a href="#">開啟地址</a>	172.31.43.27
IPv6 地址	執行個體狀態	公有 IPv4 DNS
-	✔ 執行中	ec2-34-207-55-219.compute-1.amazonaws.com   <a href="#">開啟地址</a>



12:56 UTC → Log In

於網址後輸入：  
:8080  
前面所設定的連接  
即可進入登入畫面

Sign In

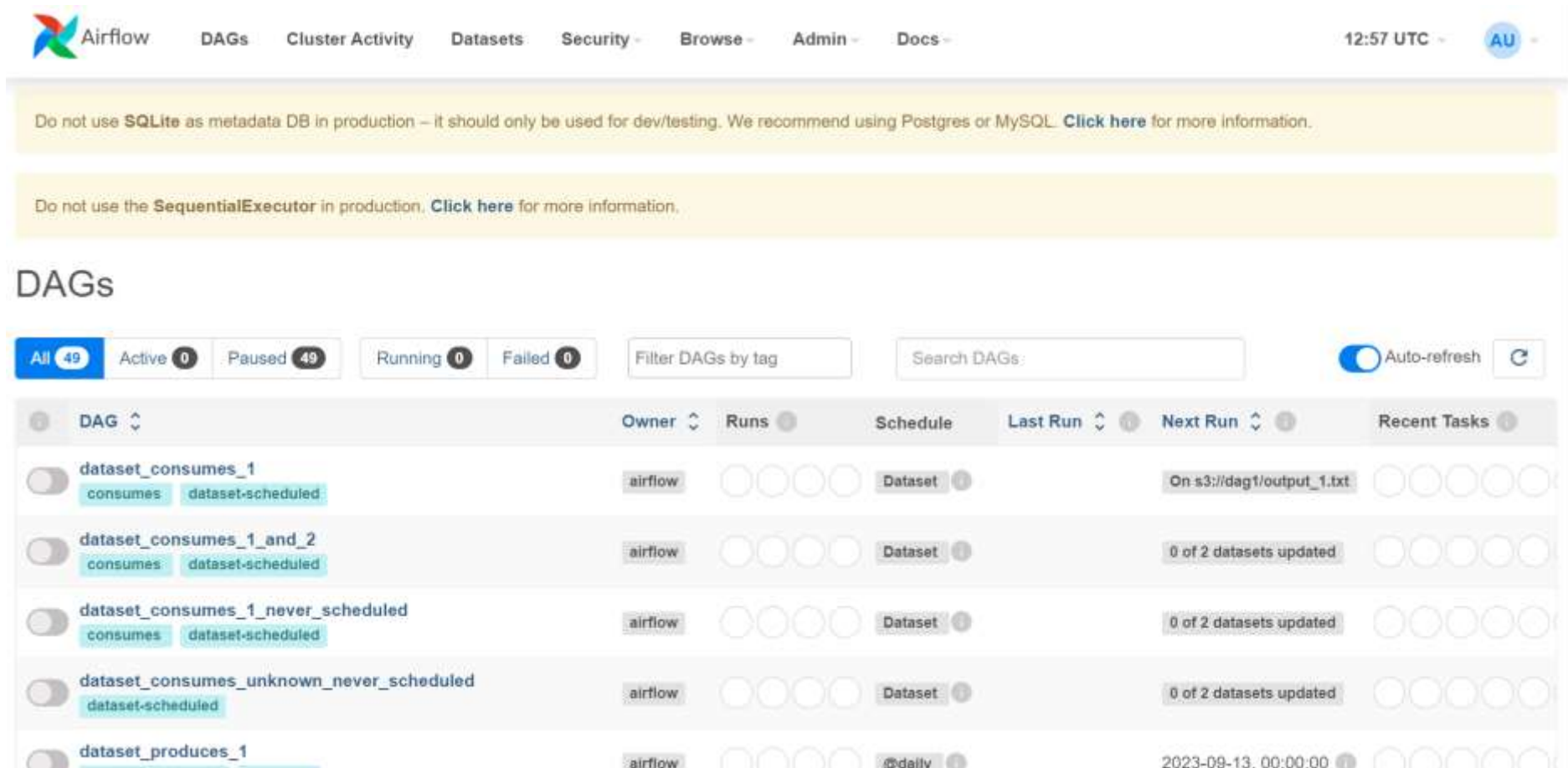
Enter your login and password below: 輸入預設的帳號密碼

Username:

Password:

Sign In

# 成功登入 Airflow

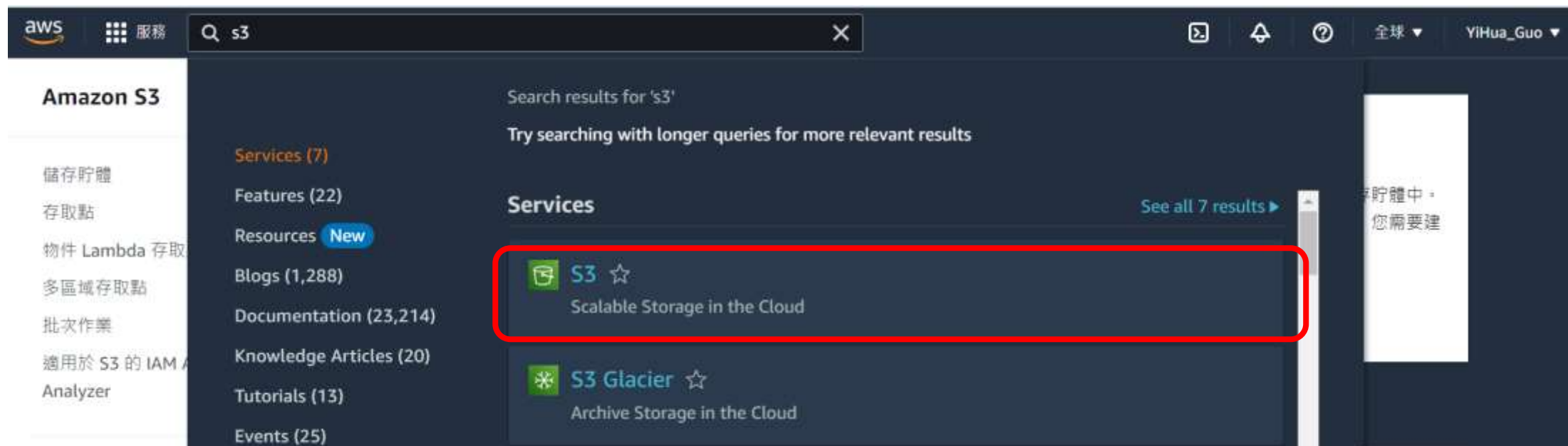


The screenshot displays the Apache Airflow web interface. At the top, the navigation bar includes the Airflow logo, links for DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs, along with the current time (12:57 UTC) and a user profile icon (AU). Below the navigation bar, two yellow warning banners are visible: one advising against using SQLite as a metadata database in production, and another advising against using the SequentialExecutor. The main section is titled "DAGs" and features a filter bar with buttons for "All" (49), "Active" (0), "Paused" (49), "Running" (0), and "Failed" (0). There is also a "Filter DAGs by tag" input field, a "Search DAGs:" search bar, and an "Auto-refresh" toggle switch. The DAGs are listed in a table with columns for DAG name, Owner, Runs, Schedule, Last Run, Next Run, and Recent Tasks. The first five DAGs are all owned by "airflow" and have a "Dataset" schedule. The first four DAGs have a "dataset-scheduled" tag, while the fifth has a "@daily" schedule.

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
<input type="checkbox"/> dataset_consumes_1 consumes dataset-scheduled	airflow	0/0/0/0/0	Dataset		On s3://dag1/output_1.txt	0/0/0/0/0
<input type="checkbox"/> dataset_consumes_1_and_2 consumes dataset-scheduled	airflow	0/0/0/0/0	Dataset		0 of 2 datasets updated	0/0/0/0/0
<input type="checkbox"/> dataset_consumes_1_never_scheduled consumes dataset-scheduled	airflow	0/0/0/0/0	Dataset		0 of 2 datasets updated	0/0/0/0/0
<input type="checkbox"/> dataset_consumes_unknown_never_scheduled dataset-scheduled	airflow	0/0/0/0/0	Dataset		0 of 2 datasets updated	0/0/0/0/0
<input type="checkbox"/> dataset_produces_1	airflow	0/0/0/0/0	@daily		2023-09-13, 00:00:00	0/0/0/0/0

## 4. 建立 *AWS S3*

# 搜尋 S3 點選



# 點選 建立儲存體

The screenshot shows the Amazon S3 console interface. At the top, there is a search bar with 's3' and a close button. To the right of the search bar are icons for a folder, a bell, a question mark, and a dropdown menu labeled '全球' (Global) with the user name 'YiHua\_Guo'. On the left side, there is a sidebar with a close button and three menu items: '取點' (Points), '1 Access', and '公開存取」設定' (Public Access Settings). The main content area has a dark blue background with the text '儲存' (Storage) at the top left. Below it, the heading 'Amazon S3' is displayed in large white font, followed by the subheading '從任何位置存放和擷取任意數量的資料' (Store and retrieve any amount of data from any location). A smaller line of text states: 'Amazon S3 是物件儲存服務，提供業界領先的可擴展性、資料可用性、安全性和效能。' (Amazon S3 is an object storage service that provides industry-leading scalability, data availability, security, and performance). On the right side of the main content area, there is a white box titled '建立儲存貯體' (Create Storage). Inside this box, there is a paragraph: 'S3 中的每個物件都會存放在儲存貯體中。若要將檔案和資料夾上傳至 S3，您需要建立存放物件的儲存貯體。' (Every object in S3 is stored in a storage class. To upload files and folders to S3, you need to create a storage class for your objects). Below this paragraph, there is a yellow button with the text '建立儲存貯體' (Create Storage), which is highlighted by a red rectangular box. At the bottom right of the console, there is a section titled '定價' (Pricing).

# S3命名

建立儲存貯體 [資訊](#)

儲存貯體是存放在 S3 中資料的容器。 [進一步了解](#)

一般組態

儲存貯體名稱

openweather\_airflow

儲存貯體名稱在全域命名空間中必須是唯一的。並遵循儲存貯體命名規則。 [請參閱儲存貯體命名規則](#)

AWS 區域

美國東部 (維吉尼亞北部) us-east-1

從現有儲存貯體複製設定 - 選用

只會複製下列組態中的儲存貯體設定。

[選擇儲存貯體](#)

► 進階設定

[?](#) 建立儲存貯體之後，您可以將檔案與資料夾上傳至儲存貯體，並設定其他儲存貯體設定。

取消 [建立儲存貯體](#)

# 建立成功畫面



 已成功建立儲存貯體 "openweather-airflow"

檢視詳細資訊 

若要上傳檔案和資料夾，或設定其他儲存貯體設定，請選擇檢視詳細資訊。

[Amazon S3](#) > 儲存貯體

▶ 帳戶快照

檢視 Storage Lens 儀表板

Storage Lens 可讓您查看儲存體用量和活動趨勢。 [進一步了解](#)

儲存貯體 (1) [資訊](#)

  複製 ARN 清空 刪除 建立儲存貯體

儲存貯體是存放在 S3 中資料的容器。 [進一步了解](#)

 依名稱尋找儲存貯體

< 1 > 

名稱	AWS 區域	存取	建立日期
 <a href="#">openweather-airflow</a>	美國東部 (維吉尼亞北部) us-east-1	<a href="#">儲存貯體和物件非公開</a>	2023年9月15日 am9:22:43 +08

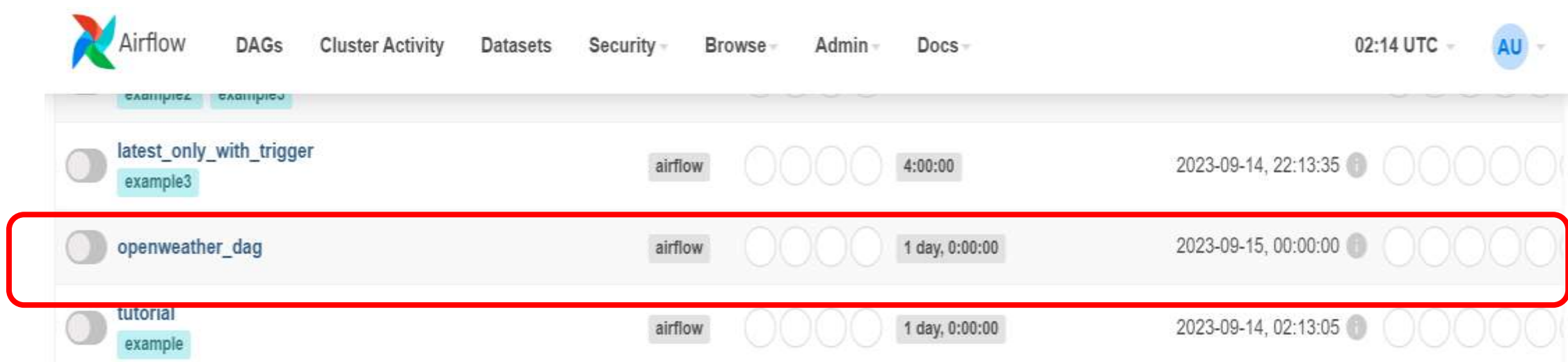


# 5. 整合ETL至 Apache Airflow

# 移至 AWS EC2 Ubuntu 中的 Airflow 資料夾



# 回到 Airflow畫面



可看到已經成功新增我們所建立的ETL檔案

點選即可進入

# 進入後 點選Code可查看程式碼

🔌 DAG: openweather\_dag Our first DAG with ETL process!

Schedule: 1 day, 0:00:00 Next Run: 2023-09-15, 00:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details **<> Code** ▶ 🗑️

Audit Log

2023/09/15 上午 02:16:57 25 All Run Types All Run States Clear Filters Auto-refresh

Press **shift + /** for Shortcuts

deferred failed queued removed restarting running scheduled shutdown skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

« » DAG openweather\_dag

⚠️ Details 🗨️ Graph 📅 Gantt **<> Code**

Parsed at: 2023-09-15, 02:16:44 UTC

```
1 from datetime import timedelta
2 from airflow import DAG
3 from airflow.operators.python import PythonOperator
4 from airflow.utils.dates import days_ago
5 from datetime import datetime
6 from openweather_etl import run_openweather_etl
```

complete\_twitter\_etl

Toggle Wrap

# 點選 Graph > 紅框三角形 即可啟動ETL

The screenshot displays the Apache Airflow web interface. At the top, there is a navigation bar with various views: Grid, Graph (selected), Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, and Code. A red box highlights the 'Run' button (a play icon) in the top right corner. Below the navigation bar, there is a filter section with a date/time picker (2023/09/15 上午 02:22:55), a dropdown for '25', and filters for 'All Run Types' and 'All Run States'. A 'Clear Filters' button and an 'Auto-refresh' toggle are also present. Below the filter section, there is a row of status buttons: deferred, failed, queued, removed, restarting, running, scheduled, shutdown, skipped, success, up\_for\_reschedule, up\_for\_retry, upstream\_failed, and no\_status. The main area shows the DAG 'openweather\_dag' in the Graph view. The DAG is a simple linear graph with a single task 'complete\_openweather\_etl' of type 'PythonOperator'. The task is highlighted with a red box. The layout is set to 'Left -> Right'.

# 執行完畢 發現執行失敗

✈ Landing Times    ≡ Gantt    ▲ Details    <> Code    ▶ 🗑

States ▾    Clear Filters    Auto-refresh ☐

running scheduled shutdown skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

Task  
15, 02:27:12 UTC / complete\_openweather\_etl    Clear task    Mark state as... ▾    Filter Tasks ▾

<> Code    ≡ Logs

Layout: Left -> Right ▾

complete\_openweather\_etl  
❌ failed  
PythonOperator

# 查看錯誤訊息

```
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/s3fs/core.py", line 113, in _error_wrapper
    return await func(*args, **kwargs)
  File "/usr/local/lib/python3.10/dist-packages/aiobotocore/client.py", line 383, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.exceptions.ClientError: An error occurred (AccessDenied) when calling the CreateBucket operation: Access Denied
The above exception was the direct cause of the following exception:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/airflow/operators/python.py", line 192, in execute
    return_value = self.execute_callable()
  File "/usr/local/lib/python3.10/dist-packages/airflow/operators/python.py", line 209, in execute_callable
    return_value = self.python_callable(*self.args, **self.kwargs)
```

推測應為 AWS 上的 EC2 與 S3 沒有建立資料傳輸的權限

## 6. 遇到Error設定EC2及S3 的IAM policy



# 至EC2執行個體畫面

執行個體 (1/2) 資訊

連線 執行個體狀態 動作 啟動新執行個體

依屬性或標籤 (case-sensitive) 尋找 執行個體

Name	執行個體 ID	執行個體狀態	執行個體類型	狀態
open_weather...	i-003973a3ae2df453b	已停止	t2.micro	-
<input checked="" type="checkbox"/> opweather_etl	i-06b5f2b178d17ec67	執行中	t2.small	2

變更安全群組 取得 Windows 密碼 修改 IAM 角色

安全性 映像和範本 監控和故障診斷

執行個體 : i-06b5f2b178d17ec67 (opweather\_etl)

勾選執行個體 並依序點選 動作 > 安全性 > 修改IAM角色

# 點選 建立新IAM角色

[EC2](#) > [執行個體](#) > [i-06b5f2b178d17ec67](#) > 修改 IAM 角色

## 修改 IAM 角色 資訊

將 IAM 角色連接至您的執行個體。

執行個體 ID

 [i-06b5f2b178d17ec67](#) (opweather\_etl)

IAM 角色

選取 IAM 角色以連接至您的執行個體，或者如果您尚未建立，則建立新的角色。您選取的角色會取代目前連接至您的執行個體的任何角色。

選擇 IAM 角色 ▼

 [建立新 IAM 角色](#) 

 如果您選擇 **沒有 IAM 角色**，則會移除目前連接至執行個體的任何 IAM 角色。您確定要從選取的執行個體移除嗎？

取消

更新 IAM 角色

# 點選 建立角色

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with the title 'Identity and Access Management (IAM)' and a search bar. The main content area is titled 'IAM > 角色' (Roles). It features a header section with the title '角色 (2) 資訊' (Roles (2) Information), a refresh button, a delete button, and a '建立角色' (Create Role) button which is highlighted with a red rectangle. Below this is a search bar and a table of existing roles. The table has two columns: '角色名稱' (Role Name) and '信任實體' (Trusted Entity). Two roles are listed: 'AWSServiceRoleForSupport' and 'AWSServiceRoleForTrustedAdvisor'. At the bottom, there is a section titled 'Roles Anywhere 資訊' (Roles Anywhere Information) with a '管理' (Manage) button.

VS 服務 Search [Alt+S] 全球 YiHua\_Guo

Identity and Access Management (IAM)

搜尋 IAM

儀表板

存取管理

使用者群組

使用者

角色

政策

身分供應商

帳戶設定

IAM > 角色

角色 (2) 資訊

IAM 角色是您可以建立的身分，其特定許可具有短期有效的憑證。您信任的實體可以擔任角色。

搜尋

1

建立角色

角色名稱	信任實體
<a href="#">AWSServiceRoleForSupport</a>	AWS 服務: support (服務連結角色)
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS 服務: trustedadvisor (服務連結角色)

Roles Anywhere 資訊

驗證您的非 AWS 工作負載，並安全地提供對 AWS 服務的存取權。

管理

# 依序勾選設定

信任的實體類型

☒ AWS 服務  
允許 AWS 服務 (如 EC2、Lambda 或其他) 在此帳戶中執行動作。

☐ AWS 帳戶  
允許屬於您或第三方的其他 AWS 帳戶中的實體在此帳戶中執行動作。

☐ Web 身分  
允許由指定的外部 Web 身分供應商聯合的使用者擔任此角色，以在您的帳戶中執行動作。

☐ SAML 2.0 聯合  
允許從公司目錄使用 SAML 2.0 聯合身分的使用者在此帳戶中執行動作。

☐ 自訂信任政策  
建立自訂信任政策，讓其他人可在您的帳戶中執行動作。



使用案例

允許 AWS 服務 (如 EC2、Lambda 或其他) 在此帳戶中執行動作。

服務或使用案例

EC2

選擇指定服務的使用案例。

使用案例

☒ EC2  
Allows EC2 instances to call AWS services on your behalf.

☐ EC2 Role for AWS Systems Manager  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

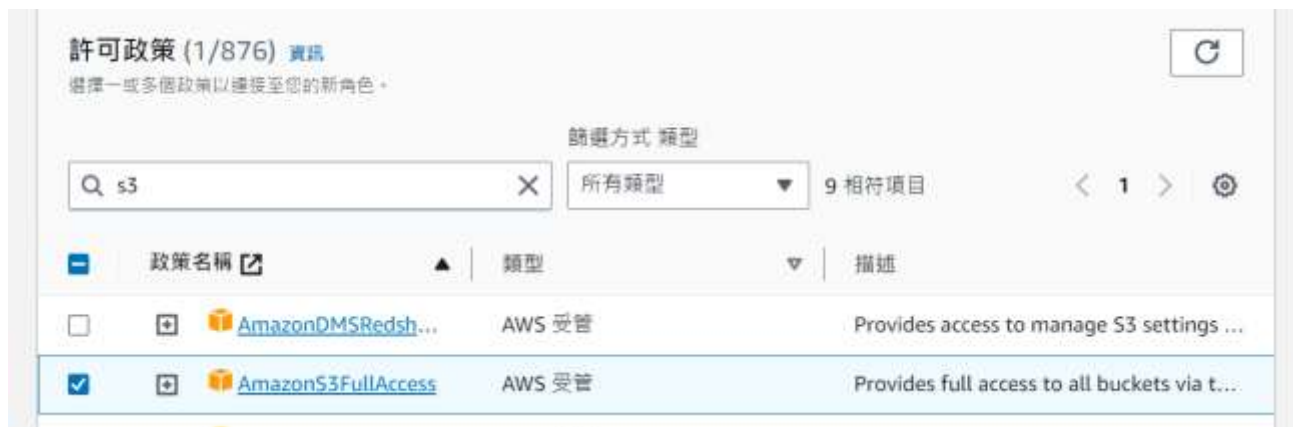
☐ EC2 - Spot Fleet  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

☐ EC2 - Scheduled Instances  
Allows EC2 Scheduled Instances to manage instances on your behalf.

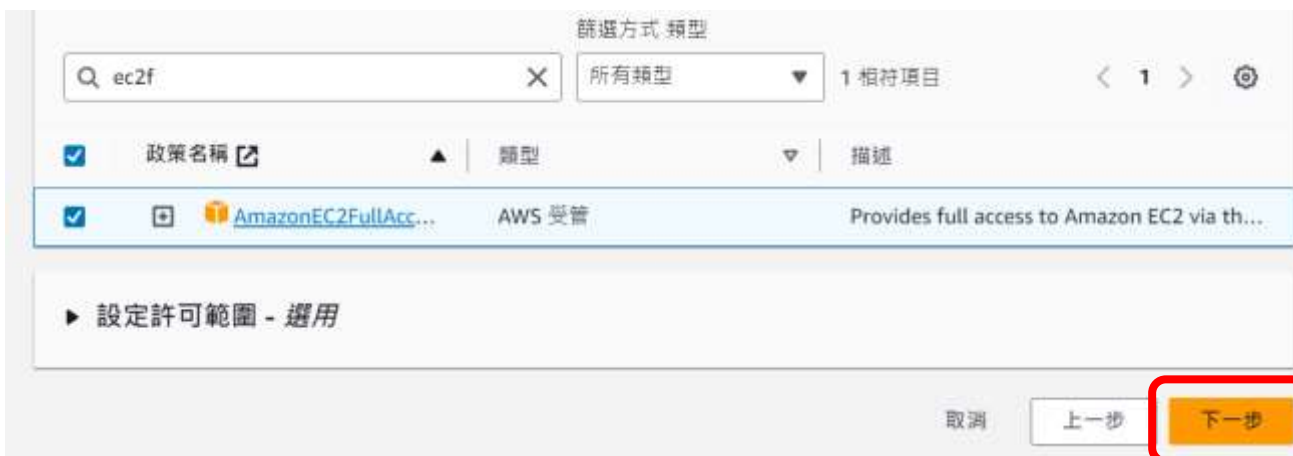
取消 下一步

# 政策設定

搜尋 S3  
勾選 AmazonS3FullAccess



搜尋 EC2  
勾選 AmazonEC2FullAccess



勾選政策完，點選下一步即可

# 為前面所制定的政策 role 命名

## 命名、檢閱和建立

### 角色詳細資訊

#### 角色名稱

輸入有意義的名稱以識別此角色。

ec2\_s3\_airflow\_role

最多 64 個字元。請使用英數字元和 '+=,.,@-\_' 字元。

#### 描述


為此角色新增簡短說明。

Allows EC2 instances to call AWS services on your behalf.

最多 1000 個字元。請使用英數字元和 '+=,.,@-\_' 字元。

# 點選 建立角色

許可政策摘要

政策名稱 	▲	類型	▼	連接為	▼
<a href="#">AmazonEC2FullAccess</a>		AWS 受管		許可政策	
<a href="#">AmazonS3FullAccess</a>		AWS 受管		許可政策	

步驟 3：新增標籤

新增標籤 - 選用 [資訊](#)

標籤是鍵值對，您可以將其新增到 AWS 資源，以協助識別、組織或搜尋資源。

沒有與該資源相關聯的標籤。

新增標籤

您最多可以再新增 50 個標籤。

取消

上一步

建立角色

# 成功建立畫面

✓ 角色 ec2\_s3\_airflow\_role 已建立。

檢視角色

✕

[IAM](#) > 角色

角色 (3) 資訊

刷新

刪除

建立角色

IAM 角色是您可以建立的身分，其特定許可具有短期有效的憑證。您信任的實體可以擔任角色。

搜尋

< 1 > ⚙

<input type="checkbox"/>	角色名稱	▲	信任實體
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>		AWS 服務: support (服務連結角色)
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>		AWS 服務: trustedadvisor (服務連結角色)
<input type="checkbox"/>	<a href="#">ec2_s3_airflow_role</a>		AWS 服務: ec2



# 回到最初 修改IAM角色 畫面

選擇前面所建立的IAM role

[EC2](#) > [執行個體](#) > [i-06b5f2b178d17ec67](#) > 修改 IAM 角色

## 修改 IAM 角色 [資訊](#)


將 IAM 角色連接至您的執行個體。

執行個體 ID

 [i-06b5f2b178d17ec67](#) (opweather\_etl)

IAM 角色

選取 IAM 角色以連接至您的執行個體，或者如果您尚未建立，則建立新的角色。您選取的角色會取代目前連接至您的執行個體的任何角色。

 [建立新 IAM 角色](#) 


沒有 IAM 角色

選擇此選項以分開 IAM 角色

ec2\_s3\_airflow\_role

arn:aws:iam::288142612271:instance-profile/ec2\_s3\_airflow\_role

ec2\_s3\_airflow\_role

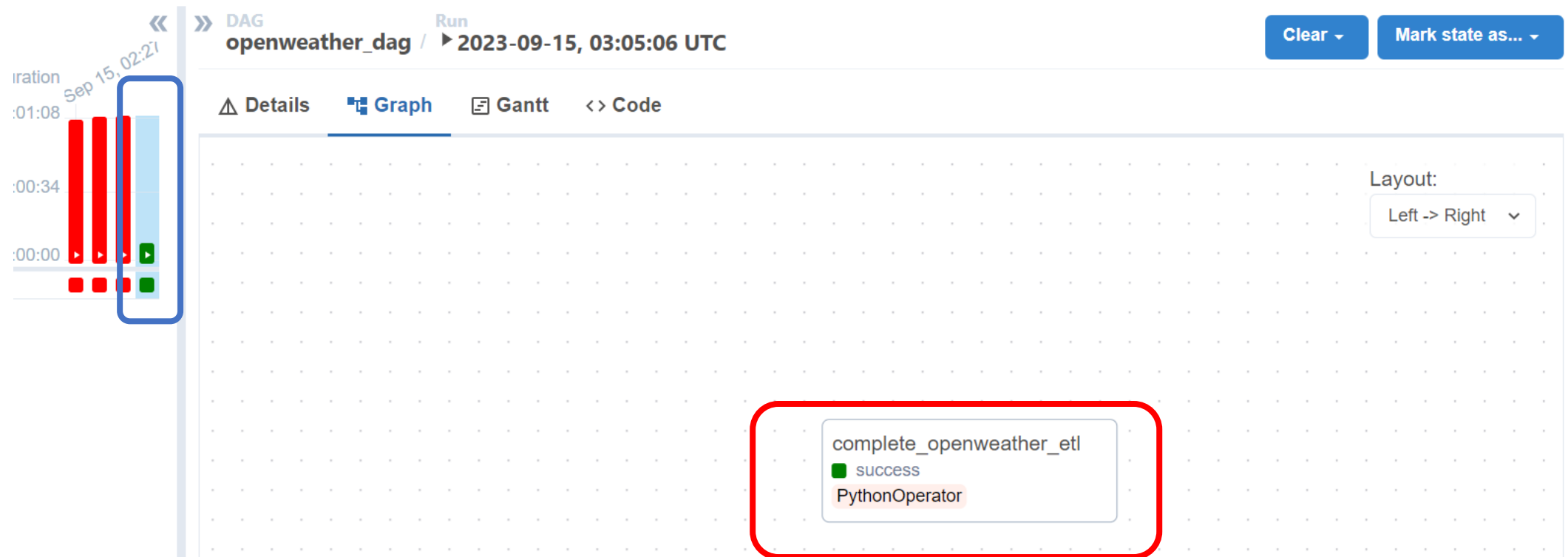


取消

更新 IAM 角色

## 7. 修正error重新執行ETL

# 綠色標示 成功執行



# 成功將EC2抓取的資料傳送儲存至S3

Amazon S3 > 儲存貯體 > openweather-airflow

## openweather-airflow 資訊

物件 屬性 許可 指標 管理 存取點

物件 (1)

物件是存放在 Amazon S3 中的基本實體。您可以使用 [Amazon S3 庫存](#) 取得儲存貯體中所有物件的清單。若要讓其他人存取您的物件，您需要明確授予這些許可。 [進一步了解](#)

複製 S3 URI 複製 URL 下載 開啟 刪除 動作 ▼ 建立資料夾

上傳

🔍 依前綴尋找物件

<input type="checkbox"/>	名稱 ▲	類型 ▼	上次修改時間 ▼	大小 ▼	儲存體類別 ▼
<input type="checkbox"/>	<a href="#">current_weather_data_portland.csv</a>	csv	2023年9月15日 am11:05:13 +08	330.0 B	標準

**End**