

# AI 雲端部署 feat. LINE Bot & Web

Enos Chou

2022/11/09, 11/16

# Enos' Steps

2022/10/24(一) 09:00~15:30

專題分享 by Enos

2022/10/31(一) 09:00~16:30

專題訂定 by Teams feat. Enos

2022/11/09(三) 09:00~16:30

AI 雲端部署 by Enos

2022/11/16(三) 09:00~16:30

AI 雲端部署 by Enos

2022/12/09(二) 09:00~16:30

專題預報 by Teams feat. Enos

# Focus

LINE Bot AI 部署

LINE Bot + AI 部署

Web + AI 部署

LIFF Web + AI 部署

Web TensorFlow.js AI 部署

# 課前準備

## 1. New GCP trial account

- a. 以三個月內新申請的帳號完成登入 <https://console.cloud.google.com>
- b. 申請時需信用卡，預刷 40 元不會請款
- c. 不須升級為完整帳戶，維持免費試用即可

勿點選啟用，請直接關閉



## 2. Ready Azure account

- a. 以班導師提供的序號儲值或新申請帳號並完成登入
- b. 序號兌換流程 <https://www.microsoftazurepass.com/Home/HowTo>
- c. 完成 Azure 登入 <https://portal.azure.com>

# 課前準備

## 3. MobaXterm Home Edition (Windows Only)

- a. 下載並安裝 MobaXterm Home Edition 最新版
- b. <https://mobaxterm.mobatek.net/> > Download > Home Edition Download now

## 4. LINE Messaging API settings

- a. 備妥一組測試用 LINE Bot 並完成 LINE Messaging API 相關設定

## 5. (Optional) TensorFlow 開發環境

- a. 備妥 Python 環境，包含 TensorFlow 2.4.4

# LINE Bot AI 部署

# LINE Bot AI 部署重點

1. 長期可用且廉價的硬體環境 ...

# Why Not Cloud ?

Cloud vs On-Premises

Tech's View

Scalability

Biz's View

CAPEX vs OPEX

# Which Cloud ?

## Magic Quadrant for Cloud

1. Magic Quadrant for Cloud Infrastructure and Platform Services  
(2021/07)
2. Magic Quadrant for Cloud AI Developer Services  
(2022/01)

## Cloud Market Share

IaaS + PaaS Market Share  
(2022/Q2)

# LINE Bot AI 部署重點

1. 長期可用且廉價的硬體環境 ... **Cloud**
2. Flask as Web Server 的替代方案 ...
3. 長期可用且廉價的 SSL 網域方案 ...

# Warning from Flask

```
python3 your_module.py # feat. app.run()
```

```
* Serving Flask app 'test'  
* Debug mode: off
```

**WARNING: This is a development server. Do not use it in a production deployment.**  
**Use a production WSGI server instead.**

```
* Running on http://127.0.0.1:5000
```

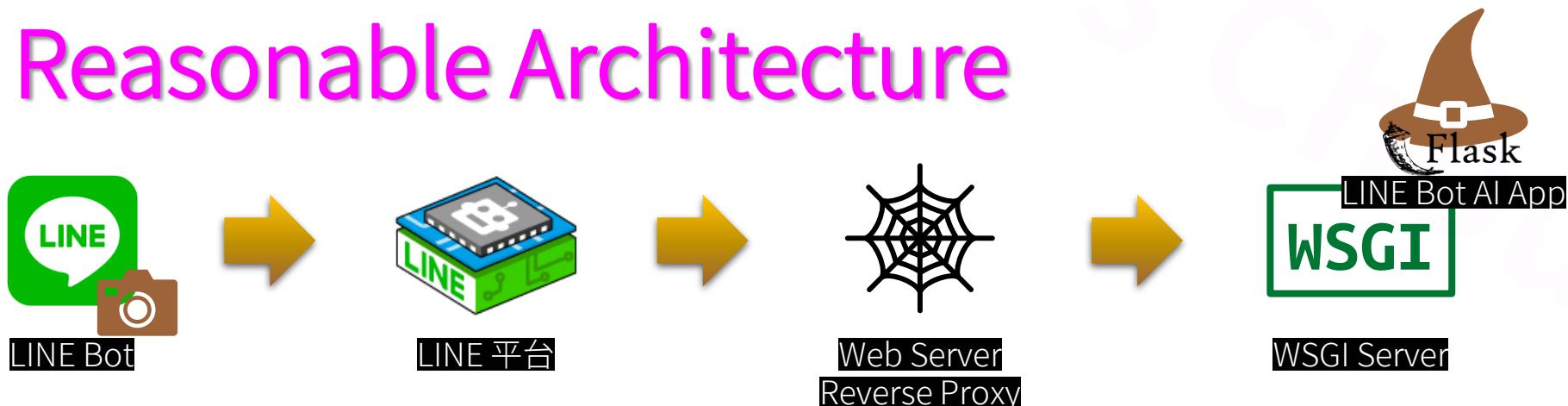
Press CTRL+C to quit

# Architecture

## Warning from Flask

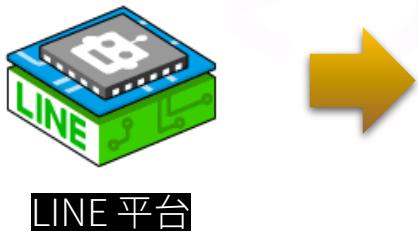
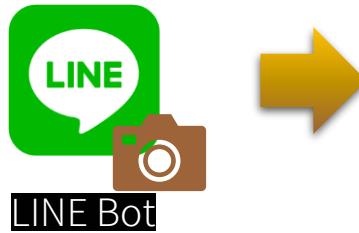


## Reasonable Architecture

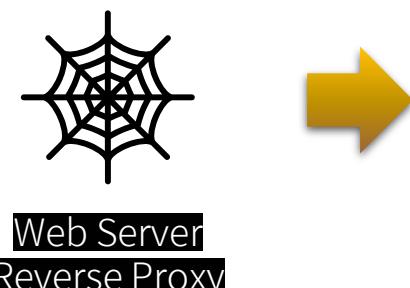
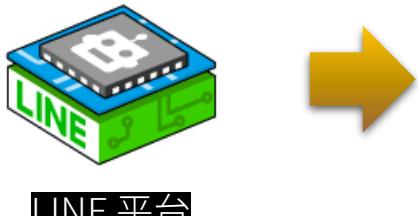


# Architecture

## Imaging Cloud Run

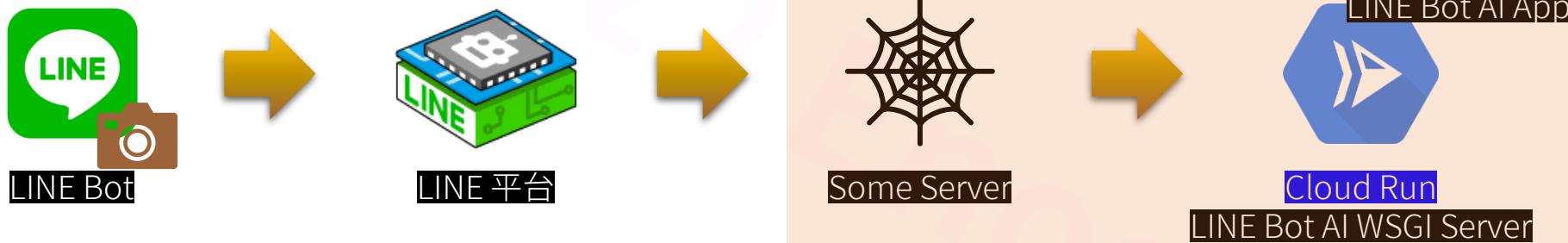


## Reasonable Architecture

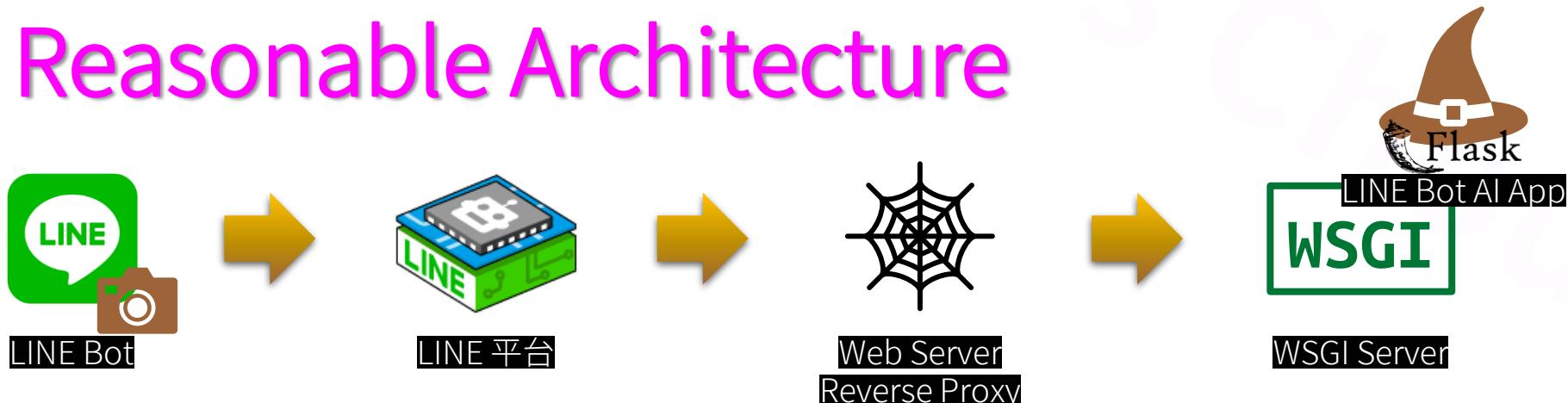


# Architecture

## Reasonable Cloud Run

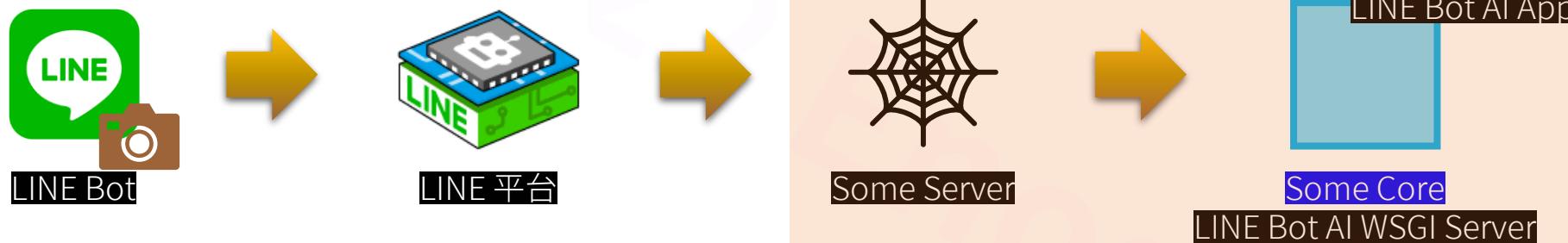


## Reasonable Architecture

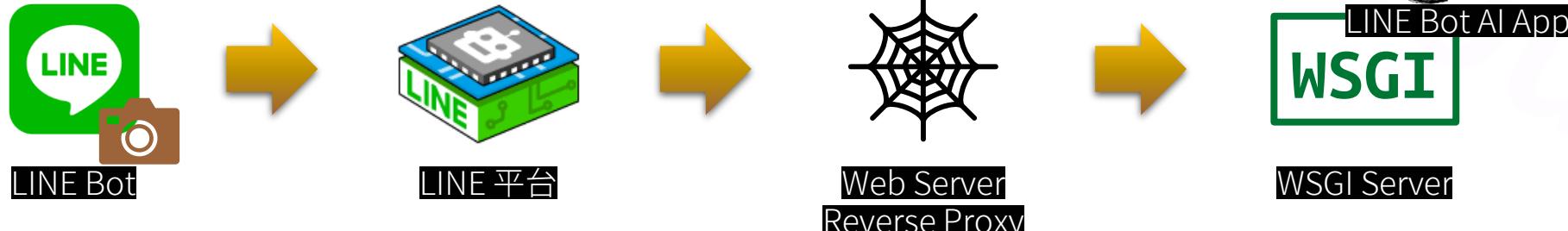


# Architecture

## Serverless

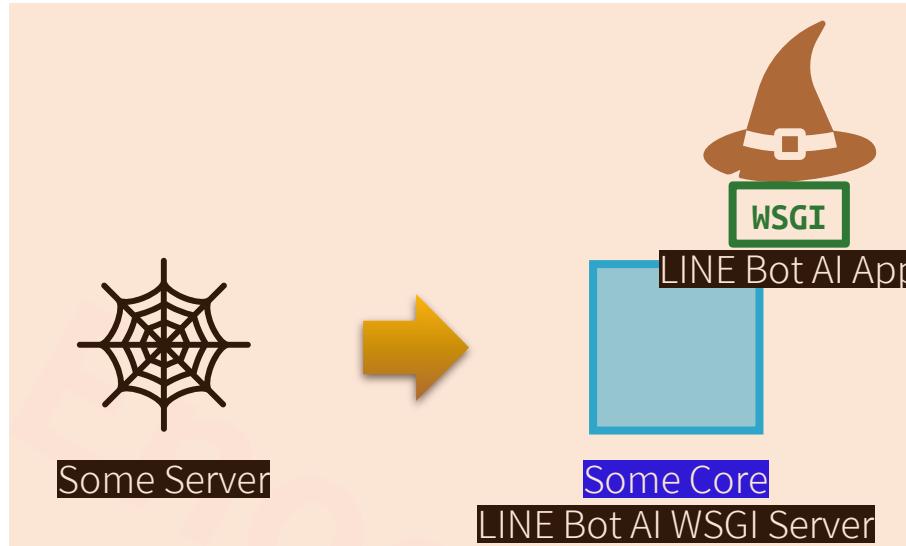
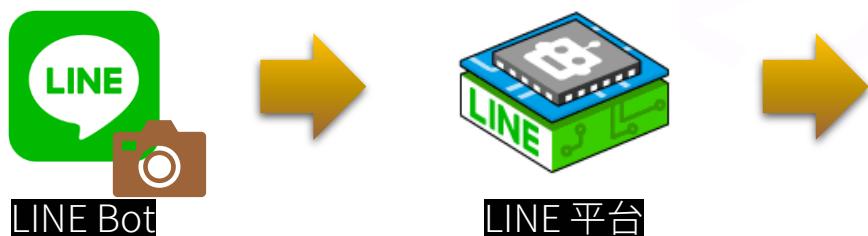


## Reasonable Architecture

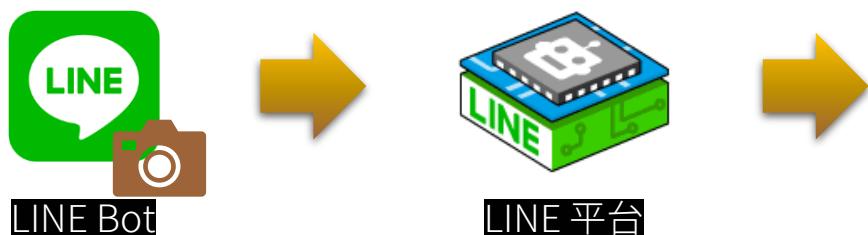


# Architecture

## Serverless

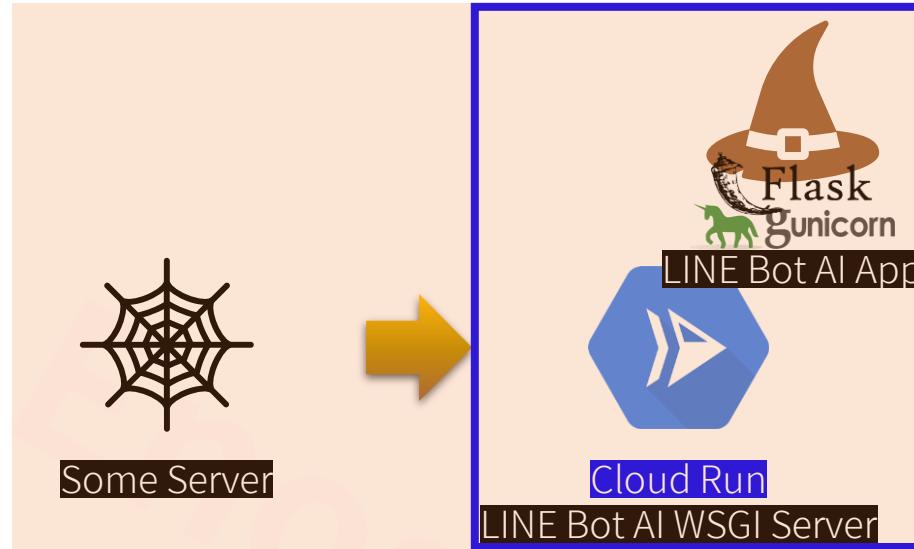
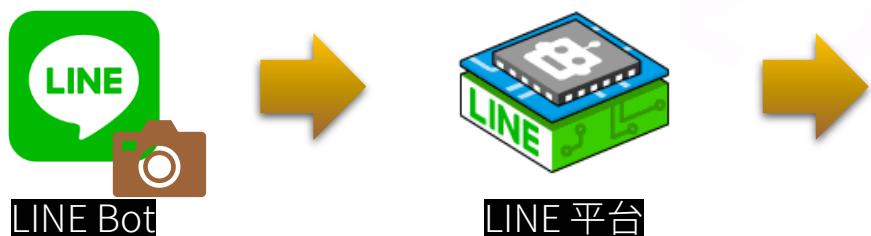


## Server or VM

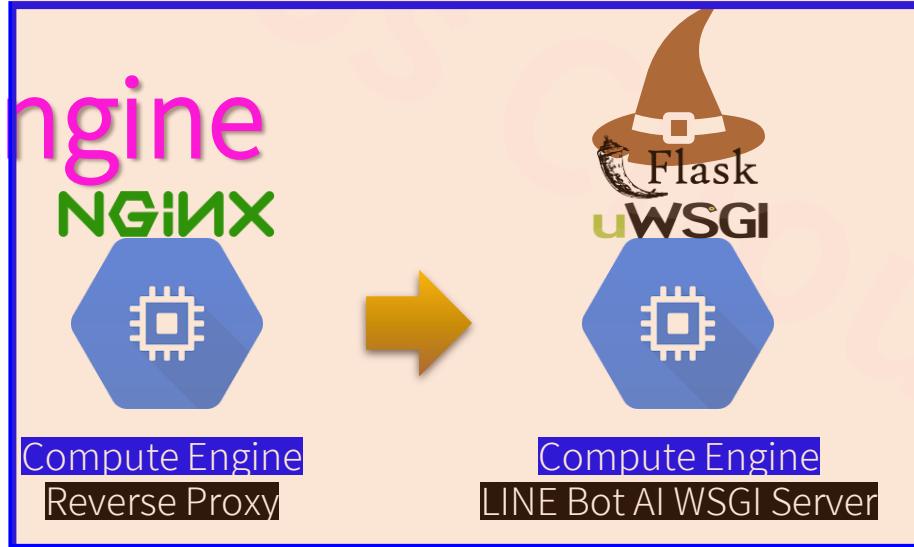


# Architecture

## Google Cloud Run



## Google Compute Engine



# LINE Bot AI 部署重點

1. 長期可用且廉價的硬體環境 ... **Cloud**
2. Flask as Web Server 的替代方案 ... Got it
3. 長期可用且廉價的 SSL 網域方案 ...

# LINE Messaging API 的 SSL 需求

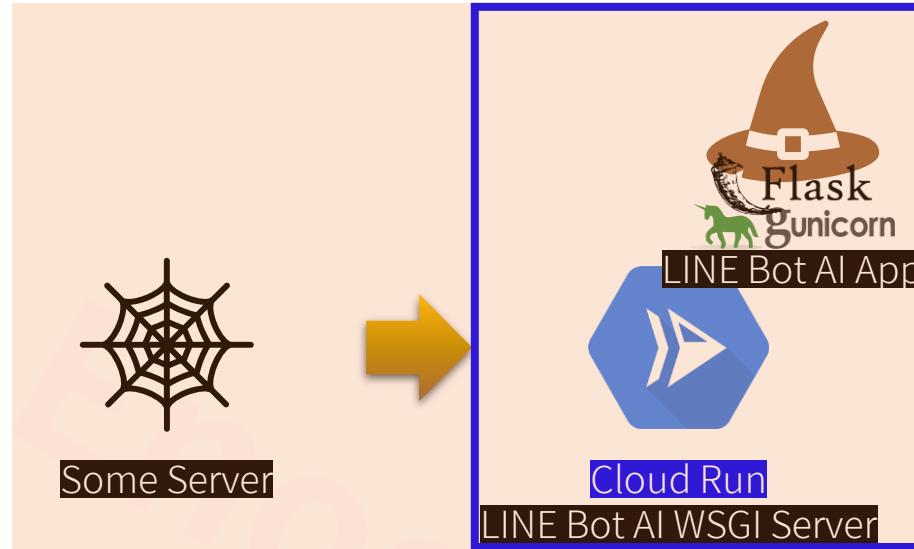
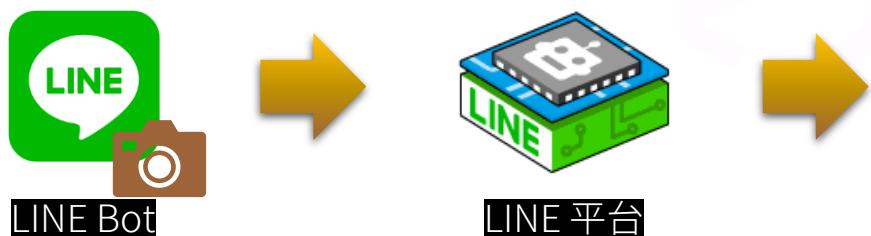
The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (which is expanded to show 'Available APIs'), 'Search...', and user account information ('Admin'). The main content area shows the path 'TOP > test > trees > Messaging API'. Under 'Available APIs', it lists 'REPLY\_MESSAGE' and 'PUSH\_MESSAGE'. Below that, the 'Webhook settings' section is shown. It includes a 'Webhook URL' field containing 'https://tenadv.site/callback' (which is highlighted with a pink rectangle), a 'Verify' button, and an 'Edit' button. A large pink annotation text '對應的 Web Server 需要 Domain Name + SSL 憑證' is overlaid on this section. There are also toggle switches for 'Use webhook' (which is turned on), 'Webhook redelivery', and 'Error statistics aggregation'.

SSL: Security Socket Layer ➔ TLS: Transport Layer Security problems? Please use our [inquiry form](#).

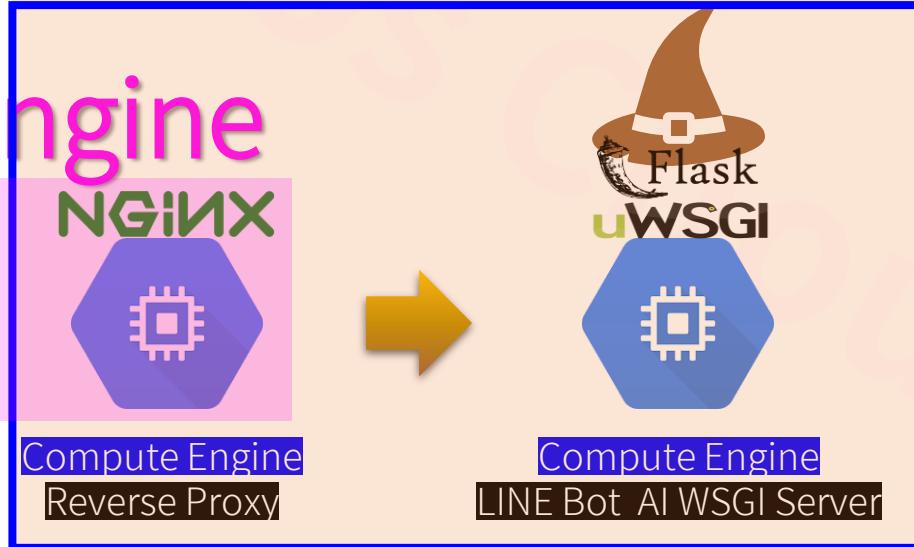
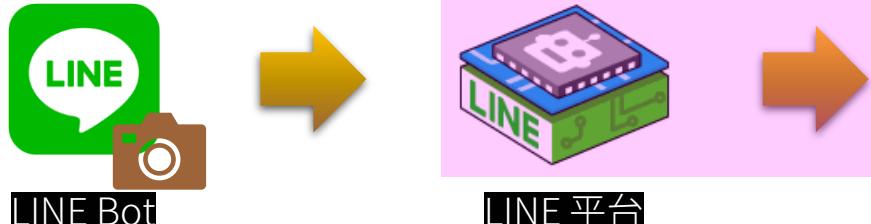
⊕ Family sites English

# Architecture

## Google Cloud Run



## Google Compute Engine



# 一線品牌憑證

## DigiCert

- <https://digicert.com/> > TLS/SSL Certificates > Basic TLS/SSL Certificates
- US\$ 250/ year

# 免費憑證

## Let's Encrypt

- <https://letsencrypt.org/zh-tw/>
- 免費憑證
- 贊助商包含 AWS、IBM、NGINX、Heroku、LINE … 等
- 效期 3 個月
- Line Messaging API 接受

# LINE Bot AI 部署重點

1. 長期可用且廉價的硬體環境 ... Cloud
2. Flask as Web Server 的替代方案 ... Got it
3. 長期可用且廉價的 SSL 網域方案 ... Got it

# LINE Bot AI 雲端部署選項

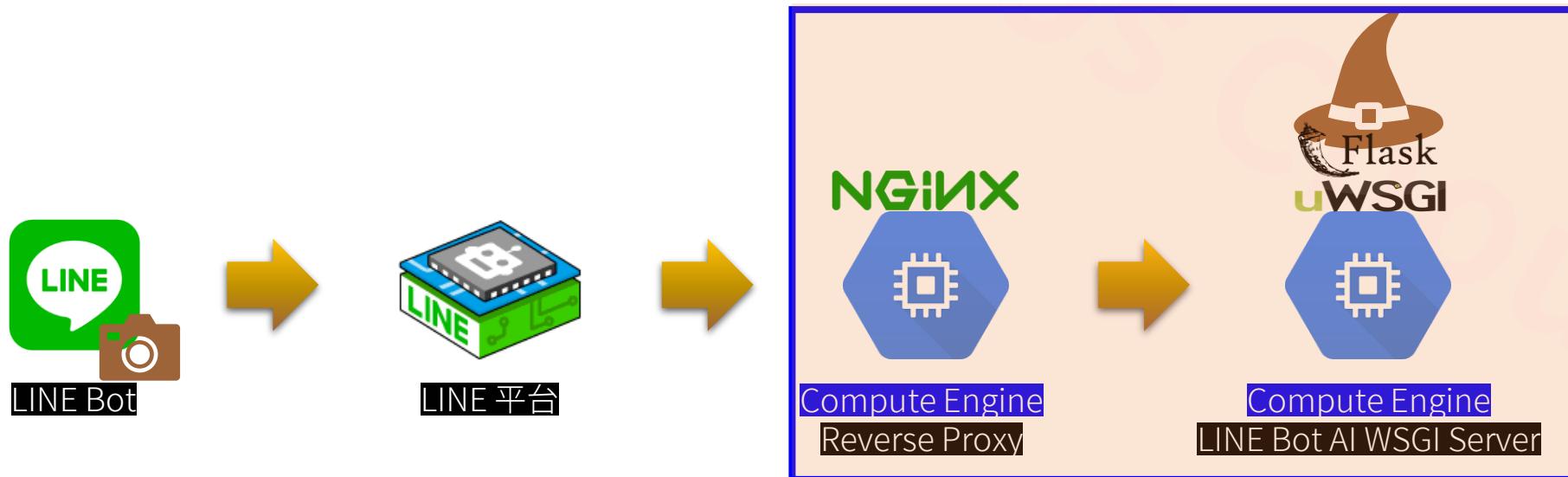
Cloud	Server (VM)	Container-Based	Serverless	Free Resource
GCP	Compute Engine (GCE)	1. GCE feat. Container 2. Cloud Run	1. Cloud Run 2. App Engine	1. free trial: US\$ 300 /3 months feat. different accounts 2. free tier
Azure	VM	1. Container Instance (ACI) 2. Web App	?	1. free trial: US\$ 200 /1 year for just one time 2. free trial: promo code 3. free tier
AWS	EC2	?	?	1. free trial: 1 year for just one time 2. free tier

## Note

1. free trial vs free tier
2. Web App ≈ Cloud Run (when min instance number > 0)

# Solution 1 - VM 部署

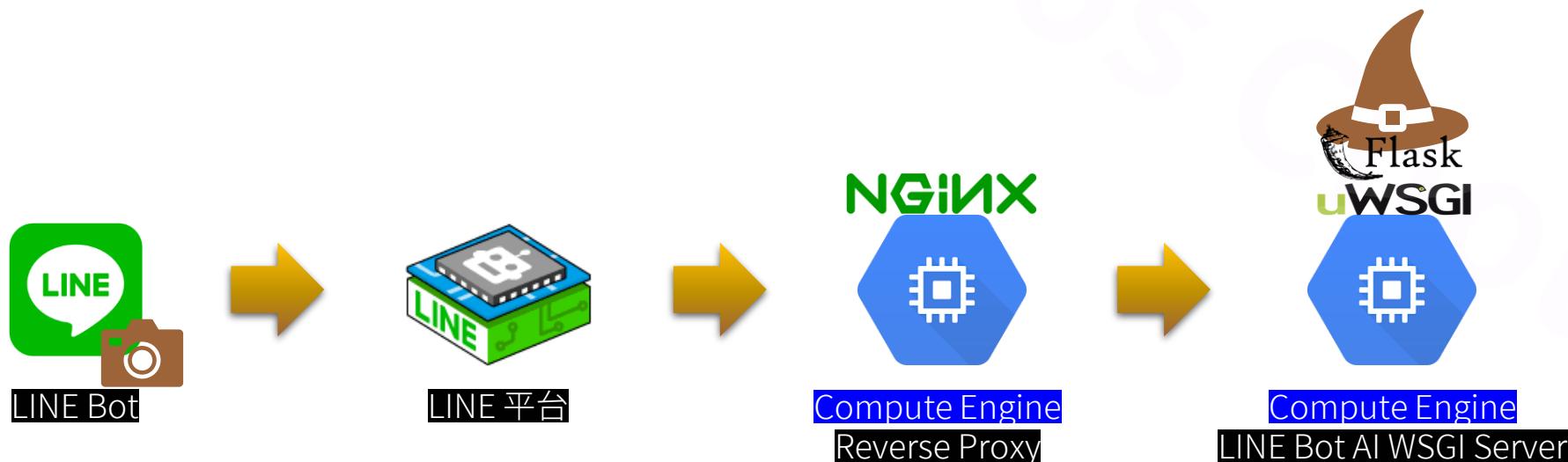
1. 長期可用且廉價的硬體環境 ... Google Compute Engine/ Azure VM
2. Flask as Web Server 的替代方案 ... NGINX + Domain + uWSGI
3. 長期可用且廉價的 SSL 網域方案 ... Let's Encrypt + Certbot

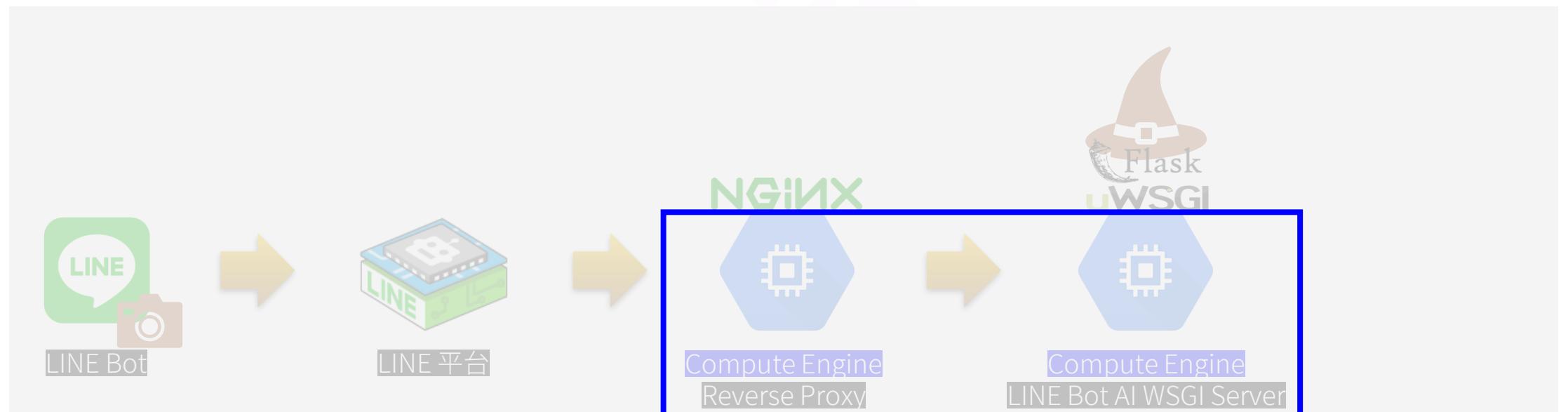


# 任務

1. 於 GCP 與 Azure 建立 VM
2. 以 MobaXterm 連線 VM 進行部署
3. 於 VM 架設 LINE Bot & uWSGI
4. 申購與設定 Domain
5. 於 VM 架設 NGINX
6. 於 VM 運用 Certbot 設定憑證

# 流程





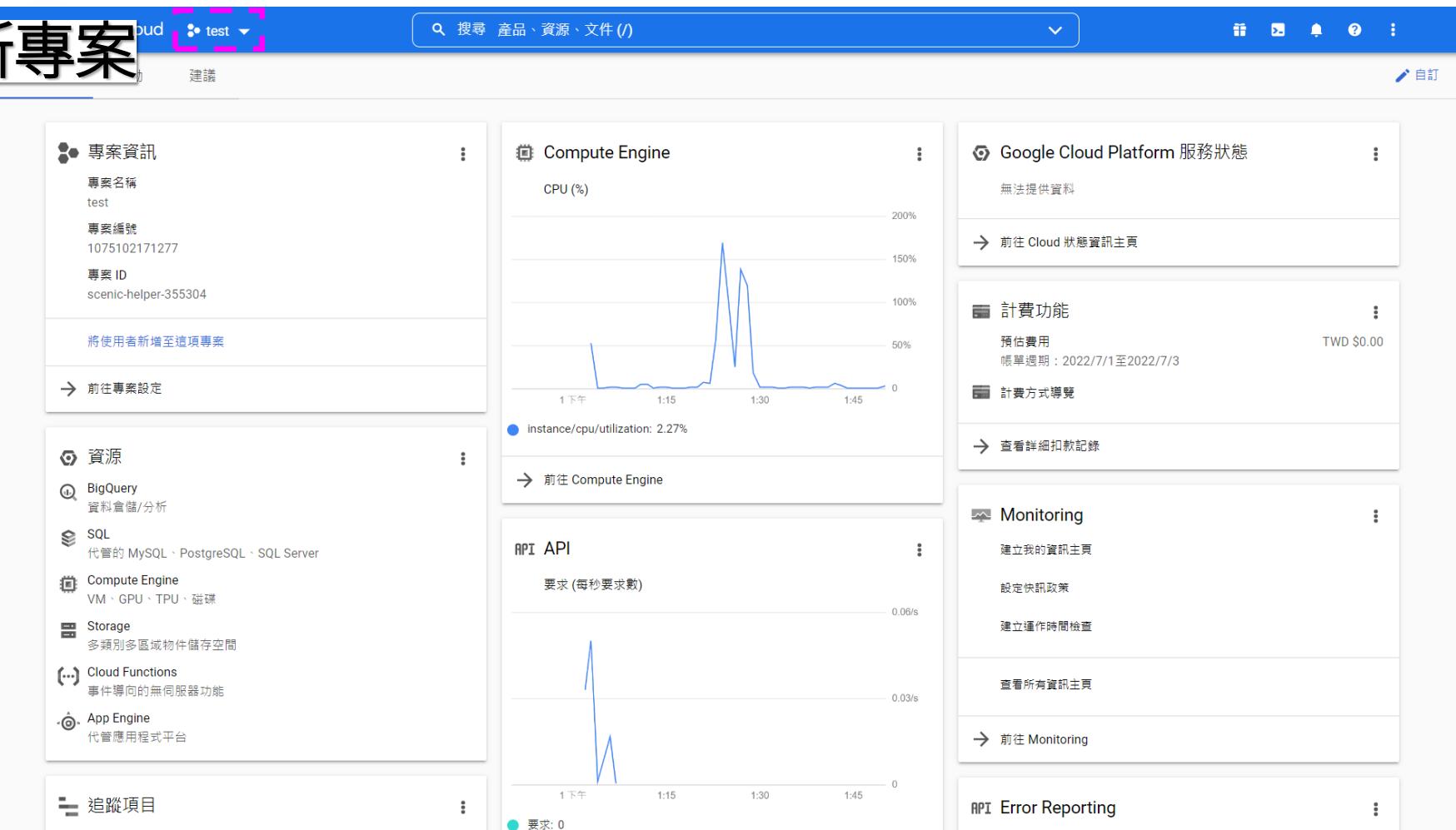
# 架設 VM - GCP

## 1. 登入

<https://console.cloud.google.com>

# 架設 VM - GCP

## 2. 建立並切換至新專案



# 架設 VM - GCP

## 3. 啟用 Compute Engine

The screenshot shows the Google Cloud Platform (GCP) dashboard. On the left, a sidebar menu is open, showing various services like Compute Engine, Kubernetes Engine, Cloud Storage, BigQuery, Cloud Run, SQL, and Google Maps Platform. The 'Compute Engine' item is highlighted with a pink rectangle. The main content area displays two charts: one for CPU utilization and another for API requests per second. The CPU chart shows a sharp spike around 1:25, with the annotation 'instance/cpu/utilization: 2.27%'. The API chart shows a similar sharp spike around the same time, with the annotation '要求: 0'. To the right of the charts, there are sections for Google Cloud Platform service status, billing information, monitoring, and error reporting.

https://console.cloud.google.com/compute?authuser=3&project=scenic-helper-355304

Compute Engine

CPU (%)

instance/cpu/utilization: 2.27%

API API

要求 (每秒要求數)

要求: 0

Google Cloud Platform 服務狀態

無法提供資料

前往 Cloud 狀態資訊主頁

計費功能

預估費用  
帳單週期：2022/7/1至2022/7/3  
TWD \$0.00

計費方式導覽

前往詳細扣款記錄

Monitoring

建立我的資訊主頁  
設定快訊政策  
建立運作時間檢查  
查看所有資訊主頁

前往 Monitoring

API Error Reporting

# 架設 VM - GCP

## 3. 啟用 Compute Engine

The screenshot shows the Google Cloud Platform API library interface. At the top, there's a blue header bar with the text "test" and some icons. Below the header, the title "Compute Engine API" is displayed, along with a "Google Enterprise API" link and a "Compute Engine API" link. A prominent blue button labeled "啟用" (Enable) is highlighted with a pink rectangle. To the right of the button is another button labeled "試用這個 API" (Try this API). Below the title, there are three tabs: "總覽" (Overview), "說明文件" (Documentation), and "支援" (Support). The "總覽" tab is selected and underlined. On the left side, there's a section titled "總覽" with the subtext "Creates and runs virtual machines on Google Cloud Platform." Below this, there's a "教學課程與說明文件" (Training courses and documentation files) section with a "Learn more" link. On the right side, there's a "其他詳細資料" (Other details) section containing information such as "類型: SaaS & APIs", "上次更新時間: 2022/4/30", "類別: Compute, Networking, Google Enterprise APIs", and "服務名稱: compute.googleapis.com".

# 架設 VM - GCP

## 3. 啟用 Compute Engine

The screenshot shows the Google Cloud Platform (GCP) Compute Engine interface. The left sidebar has a tree view with categories like 'VM 執行個體' (selected), '執行個體範本', '單一用戶群節點', '機器映像檔', 'TPU', '承諾使用折扣', and 'Migrate for Compute Eng...'. Below these are sections for '儲存空間' (Storage) and '執行個體群組' (VM Groups). The main content area is titled '虛擬機器 (VM...)' and shows a table with columns: 狀態, 名稱 (sorted up), 區域, 建議, 使用者, 內部 IP, 外部 IP, and 連線. There are several VM instances listed, each represented by a small globe icon with colored dots (red, green, blue, yellow). A large globe icon is centered below the table. To the right, there's a sidebar titled '選取執行個體' with tabs for 'PERMISSIONS', 'LABELS', and 'MONITORING'. A message says '請至少選取一項資源。' (Please select at least one resource.). At the bottom, there are two buttons: '建立執行個體' (Create VM) and '進入快速入門導覽課程' (Enter Quickstart Guide).

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Compute Engine interface for creating a VM. The left sidebar has sections for VM execution instances, storage space, instance group management, Marketplace, and version history. The main area is titled 'VM execution instance' and contains a search bar, a table header with columns for status, name, region, suggestion, user, internal IP, external IP, and connection. Below the table is a large globe icon with colored dots (green, blue, red, yellow) representing regions. A descriptive text block explains Compute Engine's purpose: "Compute Engine 可讓您使用在 Google 基礎架構中運作的虛擬機器。您可以建立微型 VM，或是執行 Debian、Windows 或其他標準映像檔的大型執行個體。請建立第一個 VM 執行個體、使用遷移服務匯入 VM 執行個體，或是透過快速入門導覽課程建構範例應用程式。" At the bottom are two buttons: 'Create instance' (highlighted with a pink box) and 'View quick start guide'.

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM instance. The main title is '建立執行個體' (Create Instance). A large pink box highlights the '名稱\*' input field where 'bot' is typed, with the note '這會成為VM主機名稱設定完畢無法更改' (This will become the VM host name, which cannot be changed after configuration).

The 'Machine Type' section is also highlighted with a pink box, showing the selected 'e2-small' configuration (2 vCPU, 2 GB Memory), with the note '依需求指定主機規格' (Specify machine type as needed).

On the right side, there's a sidebar with a table showing estimated monthly costs for the selected configuration:

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
10 GB 平衡永久磁碟	US\$1.00
Sustained use discount	-US\$0.00
Total	US\$13.23

At the bottom, there are sections for 'Compute Engine 定價' (Compute Engine Pricing) and 'LESS'.

# GCP 一律免費方案 (Free Tier)

<https://cloud.google.com/> > GCP 定價 > 20 項以上的產品

## Compute Engine

每月 1 個 e2-micro 執行個體

- 2 個 vCPU
- 1GB 記憶體
- 地區限制：奧勒岡州：us-west1、愛荷華州：us-central1、南卡羅來納州：us-east1

每月 30GB 標準永久磁碟

每月 1GB 網路輸出流量

注意：預設為已平衡的永久磁碟。移除 VM 後需個別移除磁碟

注意：使用外部 IP 仍會收費？

注意：VM 停止後，VM 不收費但磁碟與外部 IP 仍會收費

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM instance. The top navigation bar includes the Google Cloud logo, project name 'test', a search bar, and various navigation icons.

The main page title is '建立執行個體' (Create Instance). A sidebar on the left lists options: '新增 VM 執行個體' (Create new VM instance), '運用範本建立新的 VM 執行個體' (Create new VM instance from template), '運用機器映像檔建立新的 VM 執行個體' (Create new VM instance from image), and 'Marketplace'.

The main configuration area includes:

- 大小 (Size):** 10 GB
- 授權類型 (Authorization type):** 免費 (Free)
- 映像檔 (Image):** Debian GNU/Linux 11 (bullseye)
- 變更 (Change) button:** Enclosed in a red box.
- 身分及 API 存取權 (Identity & API access rights):**
  - 服務帳戶 (Service account):** Compute Engine default service account
  - 存取權範圍 (Access scope):** 允許預設存取權 (Allow default access) (selected)
  - 防火牆 (Firewall):** Options to allow HTTP and HTTPS traffic.
- 進階選項 (Advanced options):** Includes network, disk, security, and management settings.
- 預估每月費用 (Estimated monthly cost):** US\$13.23 (每小時約為 US\$0.02)
- 項目 (Project) table:**

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
10 GB 平衡永久磁碟	US\$1.00
Sustained use discount	-US\$0.00
Total	US\$13.23
- Compute Engine 定價 (Compute Engine Pricing) link.**

# 架設 VM - GCP

## 4. 建立 VM



The screenshot shows the Google Cloud VM creation interface. On the left, a sidebar lists options: '新增 VM 執行個體' (Create new VM instance), '運用範本建立新的 VM 執行個體' (Create new VM instance from template), '運用機器映像檔建立新的 VM 執行個體' (Create new VM instance from machine image), and 'Marketplace'.

The main panel is titled '開機磁碟' (Boot Disk). It includes tabs for '公開映像檔' (Public Images), '自訂映像檔' (Custom Images), '快照' (Snapshots), and '現有磁碟' (Existing Disks). The 'Ubuntu' image is selected under '公共映像檔'.

Configuration details include:

- 身分及 API 存取權**: Service account is set to 'Compute Engine default'.
- 存取權範圍**: Set to '允許預設存取權' (Allow default access).
- 防火牆**: Options for allowing HTTP and HTTPS traffic.
- 進階選項**: Includes network, disk, security, and boot options.

A large blue button at the bottom right is labeled '選取' (Select). A pink box highlights this button and the '30' value in the '大小 (GB)' (Size (GB)) input field.

**1. 硬碟空間視需求設定，一般應用可設為 30G**

**2. 預設的已平衡的永久硬碟非永久免費，為防未來若轉為正式帳號導致誤出帳可改為標準永久磁碟**

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform interface for creating a new VM instance. The configuration details are as follows:

- 大小 (Size):** 30 GB
- 授權類型 (Authorization Type):** 免費 (Free)
- 映像檔 (Image):** Ubuntu 18.04 LTS
- 預估每月費用 (Estimated Monthly Cost):** US\$13.43 (每小時約為 US\$0.02)
- 身分及 API 存取權 (Identity & API Access):** 使用 Compute Engine default service account
- 存取範圍 (Access Scope):** 允許預設存取權 (Allow default access)
- 防火牆 (Firewall):** 允許 HTTP 流量 (Allow HTTP traffic) 和 允許 HTTPS 流量 (Allow HTTPS traffic) 两个选项均被勾选。
- 進階選項 (Advanced Options):** 包括網路、磁碟、安全性、管理、單一用戶群等。

右侧显示了预估的每月费用表：

項目 (Project)	預估每月費用 (Estimated Monthly Cost)
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

下方有链接：Compute Engine 定價 (Compute Engine Pricing) 和 LESS (Less).

HTTP for Certbot  
HTTPS for LINE Messaging

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform interface for creating a new VM instance. The top navigation bar includes 'Google Cloud' and a project dropdown ('test'). A search bar is at the top right.

The main page title is '建立執行個體' (Create Instance). On the left, a sidebar lists creation options:

- 新增 VM 執行個體** (Create New VM Instance) - Selected.
- 運用範本建立新的 VM 執行個體** (Create New VM Instance from Template)
- 運用機器映像檔建立新的 VM 執行個體** (Create New VM Instance from Image)
- Marketplace** (Marketplace)

The main configuration area includes the following sections:

- 針對各個 API 設定存取權** (Set API Access for Each API): Includes a note about adding tags and firewall rules.
- 防火牆** (Firewall):
  - Allow HTTP traffic (checked)
  - Allow HTTPS traffic (checked)
- 進階選項** (Advanced Options): Includes '網路' (Networking).
- 網路** (Networking): Sub-section for main machine name and network interface.
- 磁碟** (Storage): Sub-section for other disk.
- 安全性** (Security): Sub-section for protected VM and security features.
- 管理** (Management): Sub-section for terms, deletion protection, and policies.
- 單一用戶群** (Single User Group): Sub-section for node placement and CPU oversubscription.

A note at the bottom states: '這個 VM 執行個體將會耗用免費試用額度。 [Google Cloud 免費方案](#)' (This VM instance will consume free trial usage. [Google Cloud Free Trial](#)).

On the right, there's a summary of estimated monthly costs:

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

Below the table are links for 'Compute Engine 定價' (Compute Engine Pricing) and 'LESS'.

At the bottom, there are buttons: '建立' (Create), '取消' (Cancel), and '對等指令列' (Peer-to-Peer Queue).

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM instance. The top navigation bar includes the Google Cloud logo, project name 'test', a search bar, and various navigation icons.

**建立執行個體** (Create Instance) page:

- 如要建立 VM 執行個體，請先選取下列任一選項：**
  - + 新增 VM 執行個體** (Create new VM instance): Selected option.
  - + 運用範本建立新的 VM 執行個體** (Create new VM instance from template):
  - + 運用機器映像檔建立新的 VM 執行個體** (Create new VM instance from image):
- 網路介面** (Network Interface):
  - 網路介面一經設定即無法變更 (The network interface cannot be changed once set).
  - 目前選擇：default default (10.138.0.0/20)
  - 新增網路介面 (Add network interface):
- 磁碟** (Disk): Other disks.
- 安全性** (Security): Protected VM and security features.
- 管理** (Management): Information about deletion, retention, automation, and availability policies.
- 單一用戶群** (Single User Group): Notes on node affinity and CPU oversubscription.

**預估每月費用** (Estimated monthly cost): **US\$13.43**.  
每小時約為 US\$0.02.  
用多少付多少：無須預繳費用，而是以秒計費.

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

**Compute Engine 定價** (Compute Engine Pricing) and **LESS** buttons.

Bottom navigation buttons: **建立** (Create), **取消** (Cancel), and **對等指令列** (Peer-to-peer list).

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM instance. The top navigation bar includes the Google Cloud logo, project name 'test', a search bar, and various navigation icons.

The main section is titled '建立執行個體' (Create Instance). On the left, there's a sidebar with four options:

- 新增 VM 執行個體** (Create New VM Instance): Selected and highlighted in blue.
- 運用範本建立新的 VM 執行個體** (Create New VM Instance from Template)
- 運用機器映像檔建立新的 VM 執行個體** (Create New VM Instance from Image)
- Marketplace** (Marketplace)

The main configuration area is titled '網路介面' (Network Interface). It shows the following settings:

- 編輯網路介面** (Edit Network Interface): A dropdown menu set to 'default'.
- 子網路** (Subnetwork): A dropdown menu set to 'default IPv4 (10.138.0.0/20)'.
- IP 堆疊類型** (IP Stack Type): Radio button selected for 'IPv4 (單一堆疊)' (IPv4 (Single Stack)).
- 主要內部 IP** (Primary Internal IP): Set to '臨時 (自動)' (Temporary (Automatic)).
- 別名 IP 範圍** (Alias IP Range): A dropdown menu set to '臨時' (Temporary).
- 網路服務級別** (Network Service Level): Radio button selected for '進階' (Advanced).

To the right, there's a summary section titled '預估每月費用' (Estimated Monthly Cost) showing:

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

Below the cost summary, there are links for 'Compute Engine 定價' (Compute Engine Pricing) and a 'LESS' link.

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM. The top navigation bar includes 'Google Cloud' and a project dropdown 'test'. A search bar is at the top right. Below the navigation is a breadcrumb trail '建立執行個體'.

The main content area is titled '建立執行個體' and displays the following steps:

- 如要建立 VM 執行個體，請先選取下列任一選項：**
  - + 新增 VM 執行個體** (selected): 從頭開始建立一個 VM 執行個體
  - + 運用範本建立新的 VM 執行個體**: 運用現有範本建立一個 VM 執行個體
  - + 運用機器映像檔建立新的 VM 執行個體**: 運用現有機器映像檔建立一個 VM 執行個體
  - Marketplace**: 將立即可用的解決方案部署至 VM 執行個體
- 網路介面** (selected): 網路介面一經設定即無法變更
- 編輯網路介面** (selected):
  - 網路 \***: default
  - 子網路 \***: default IPv4 (10.138.0.0/20)
  - IP 堆疊類型**:
    - IPv4 (單一堆疊)** (selected)
    - IPv4 和 IPv6 (雙重堆疊)**
  - 別**:
    - 無**
    - 臨時**
  - 建立 IP 位址** (button highlighted with a pink box)
- 網路服務級別**:
  - 進階** (selected)
  - 標準級 (us-west1)**

**預估每月費用**: **US\$13.43**  
每小時約為 US\$0.02  
用多少付多少：無須預繳費用，而是以秒計費

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

**Compute Engine 定價**  
**LESS**

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM instance. The left sidebar lists options: '新增 VM 執行個體' (Create new VM instance), '運用範本建立新的 VM 執行個體' (Create new VM instance from template), '運用機器映像檔建立新的 VM 執行個體' (Create new VM instance from image), and 'Marketplace'.

The main panel shows the configuration for a 'default' network interface. It includes fields for '子網路' (Subnetwork) set to 'default IPv4 (10.138.0.0/20)', 'IP 堆疊類型' (IP stack type) set to 'IPv4 (單一堆疊)' (IPv4 (Single stack)), and a note about IPv6 support. A '保留新的靜態 IP 位址' (Keep new static IP address) dialog is overlaid, asking for a static IP name 'botip' and providing a '說明' (Description) field. The 'Keep' button is highlighted with a pink box.

On the right, a summary table shows estimated monthly costs:

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

Below the table, there's a section for 'Compute Engine 定價' (Compute Engine Pricing) and a 'LESS' link.

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform (GCP) interface for creating a new VM. The top navigation bar includes 'Google Cloud' and a project dropdown ('test'). A search bar is at the top right. The main title is '建立執行個體' (Create Instance). On the left, a sidebar lists options: '新增 VM 執行個體' (Create new VM instance), '運用範本建立新的 VM 執行個體' (Create a new VM instance from a template), '運用機器映像檔建立新的 VM 執行個體' (Create a new VM instance from a machine image), and 'Marketplace'.

The main content area is titled '網路介面' (Network Interface). It shows a summary: '傳出網路頻寬上限 : 1 Gbps'. Below this is a '編輯網路介面' (Edit Network Interface) section. It lists '網路\*' (Network) as 'default' and '子網路\*' (Subnetwork) as 'default IPv4 (10.138.0.0/20)'. A note says: '如要使用 IPv6，您必須擁有 IPv6 子網路範圍。' (If you want to use IPv6, you must have an IPv6 subnet range.)

Below this is an 'IP 堆疊類型' (IP Stack Type) section with 'IPv4 (單一堆疊)' (IPv4 (Single Stack)) selected. It also shows '主要內部 IP' (Main Internal IP) as '臨時 (自動)' (Temporary (Automatic)).

Under '別名 IP 範圍' (Alias IP Range), there is a '新增 IP 範圍' (Add IP Range) button and a listed range '外部 IPv4 位址' (External IPv4 Address) as 'botip (34.105.119.208)'.

At the bottom, there are sections for '網路服務級別' (Network Service Level) set to '進階' (Advanced) and '公開 DNS PTR 記錄' (Public DNS PTR Record).

On the right side, there is a '預估每月費用' (Estimated monthly cost) summary table:

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

Below the table are links for 'Compute Engine 定價' (Compute Engine Pricing) and 'LESS'.

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Platform interface for creating a new VM instance. The top navigation bar includes the Google Cloud logo, project name 'test', a search bar, and various navigation icons.

The main page title is '建立執行個體' (Create Instance). On the left, a sidebar lists options for creating a VM:

- 新增 VM 執行個體** (Create New VM Instance) - Selected.
- 運用範本建立新的 VM 執行個體** (Create New VM Instance from Template)
- 運用機器映像檔建立新的 VM 執行個體** (Create New VM Instance from Image)
- Marketplace** (Marketplace)

The main configuration area includes the following sections:

- IP Address**:
  - 啟用 IPv4
  - PTR 網域名稱
  - 完成** (Finish)
- 新增網路介面** (Add Network Interface)
- 磁碟** (Disk): Other disks
- 安全性** (Security): Protected VM and security features
- 管理** (Management): Documentation, Deletion protection, Project, Automation, and Availability policy
- 單一用戶群** (Single User Group): Quota settings and CPU usage configuration

A note at the bottom states: '這個 VM 執行個體將會耗用免費試用額度。 [Google Cloud 免費方案](#)' (This VM instance will consume free trial usage. [Google Cloud Free Trial](#))

At the bottom right are buttons: **建立** (Create), 取消 (Cancel), and 對等指令列 (Peer-to-Peer List).

**預估每月費用** (Estimated Monthly Cost): **US\$13.43**  
每小時約為 US\$0.02  
用多少付多少：無須預繳費用，而且是以秒計費

項目	預估每月費用
2 vCPU + 2 GB memory	US\$12.23
30 GB 標準永久磁碟	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$13.43

[Compute Engine 定價](#) | [LESS](#)

# 架設 VM - GCP

## 4. 建立 VM

The screenshot shows the Google Cloud Compute Engine interface for managing VM instances. On the left, a sidebar lists categories like 'Compute Engine', 'Virtual Machines', 'Storage', 'Machine Groups', 'Marketplace', and 'Versions'. The main area displays a table of VM instances. One instance, named 'bot', is highlighted with a green checkmark and has its external IP address, 34.105.119.208, under the 'External IP' column. A pink box highlights this IP address. The table also includes columns for Status, Name, Zone,建议 (Recommendation), User, Internal IP, and External IP. Below the table, several related actions are listed: 'Explore Actifio GO' (Backup VM and set disaster recovery), 'View Bill Report' (View and manage Compute Engine bills), 'Monitor VM' (View CPU and network metrics for individual VMs), 'Set Firewall Rules' (Manage VM instance traffic flow), and 'Manage Patching' (Schedule patch updates and view patch status). A large pink box labeled '記錄外部 IP' (Record External IP) is overlaid on the screenshot.

狀態	名稱	可用區	建議	使用者	內部 IP	外部 IP	連線
<input checked="" type="checkbox"/>	<a href="#">bot</a>	us-west1-b			10.138.0.3 (nic0)	34.105.119.208 (nic0)	SSH

記錄外部 IP

# 架設 VM - GCP

## 5. 準備開發環境

### 建立金鑰

<https://cloud.google.com/> > 文件 > 運算 > Compute Engine > Guides: Managing SSH Keys in metadata > Windows

### 設定用戶端

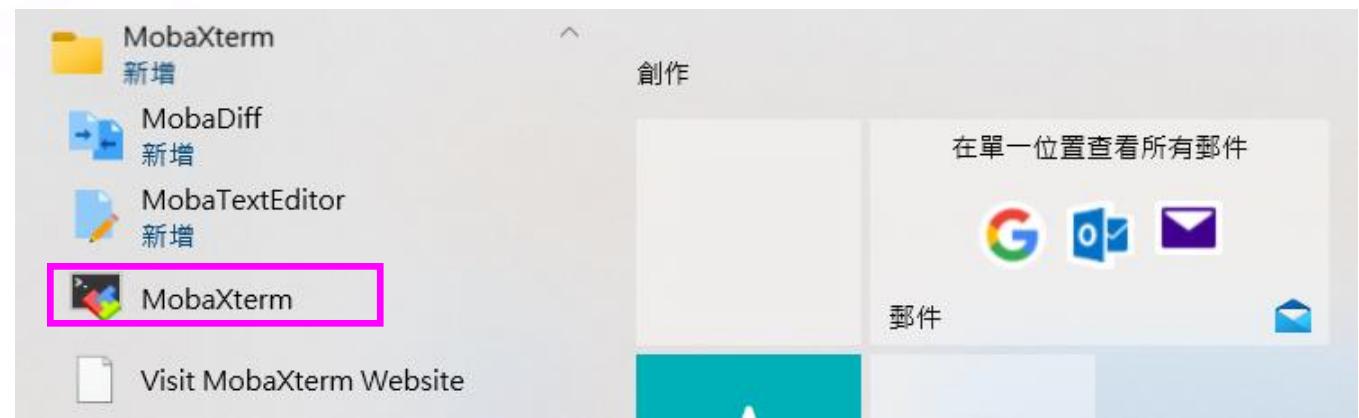
<https://cloud.google.com/> > 文件 > 運算 > Compute Engine > Guides: Connecting to instances > Connecting to Linux VMs using advanced methods > Windows(PuTTY)

# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

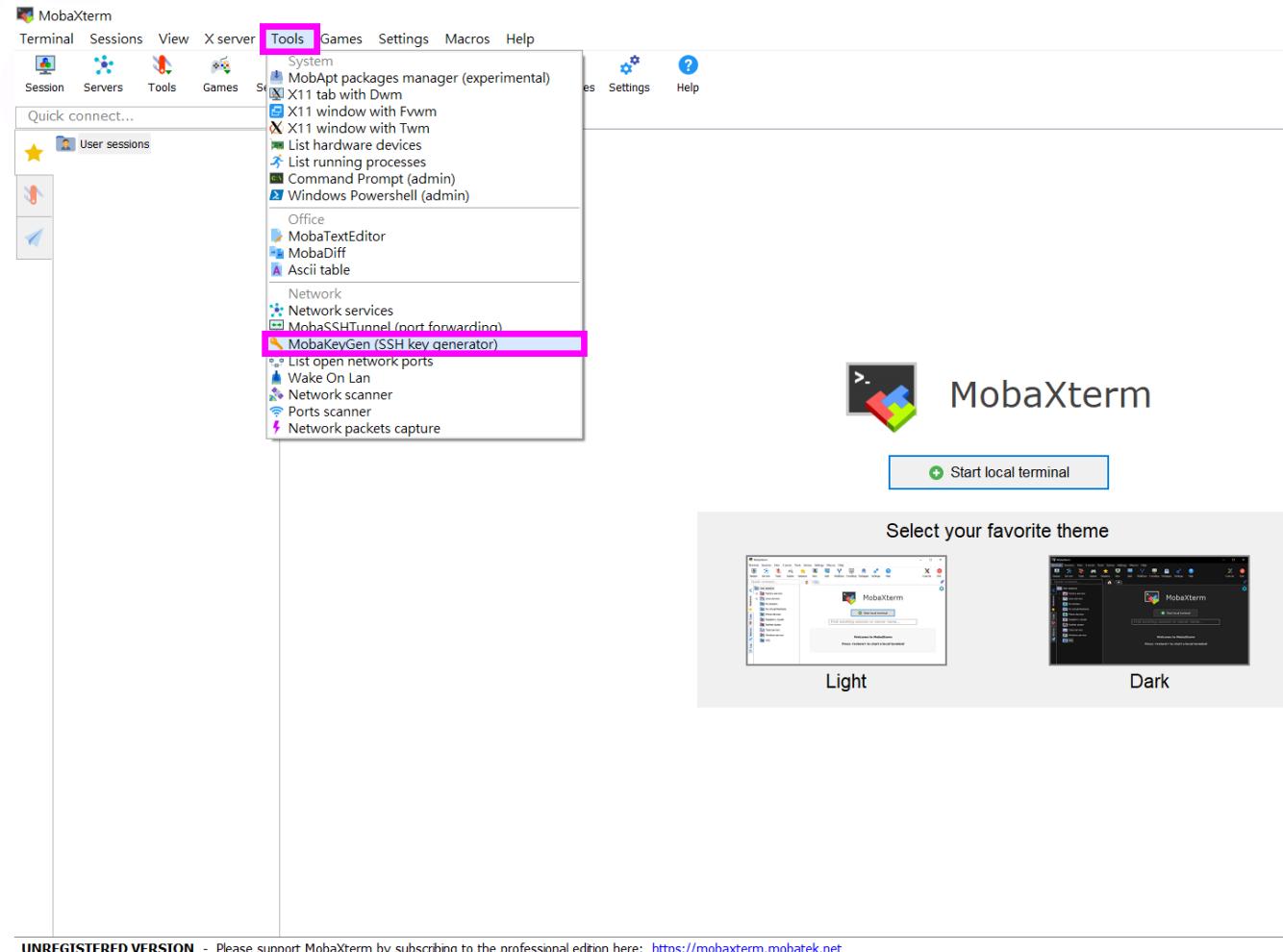
啟動 MobaXterm



# 架設 VM - GCP

## 5. 準備開發環境

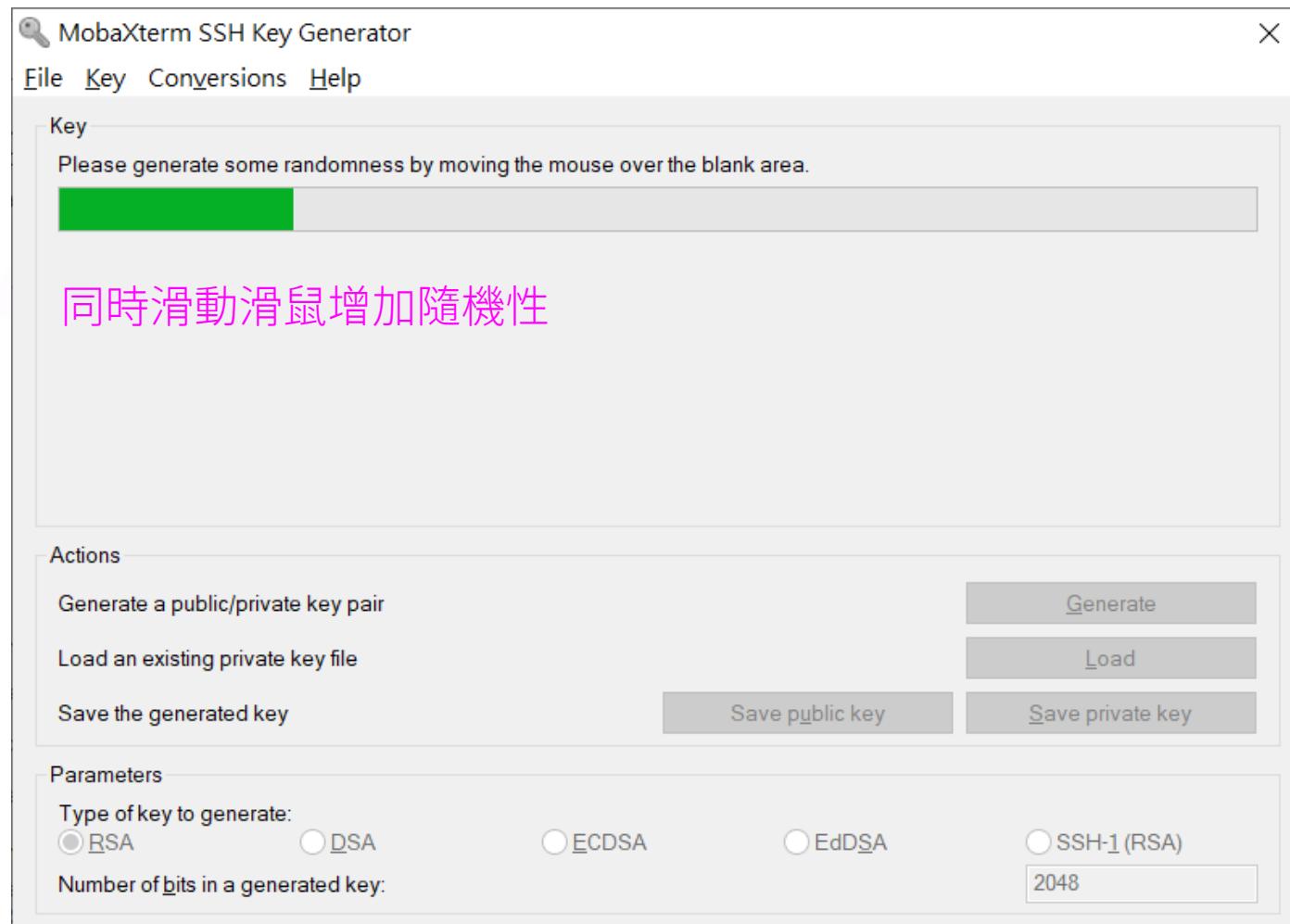
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

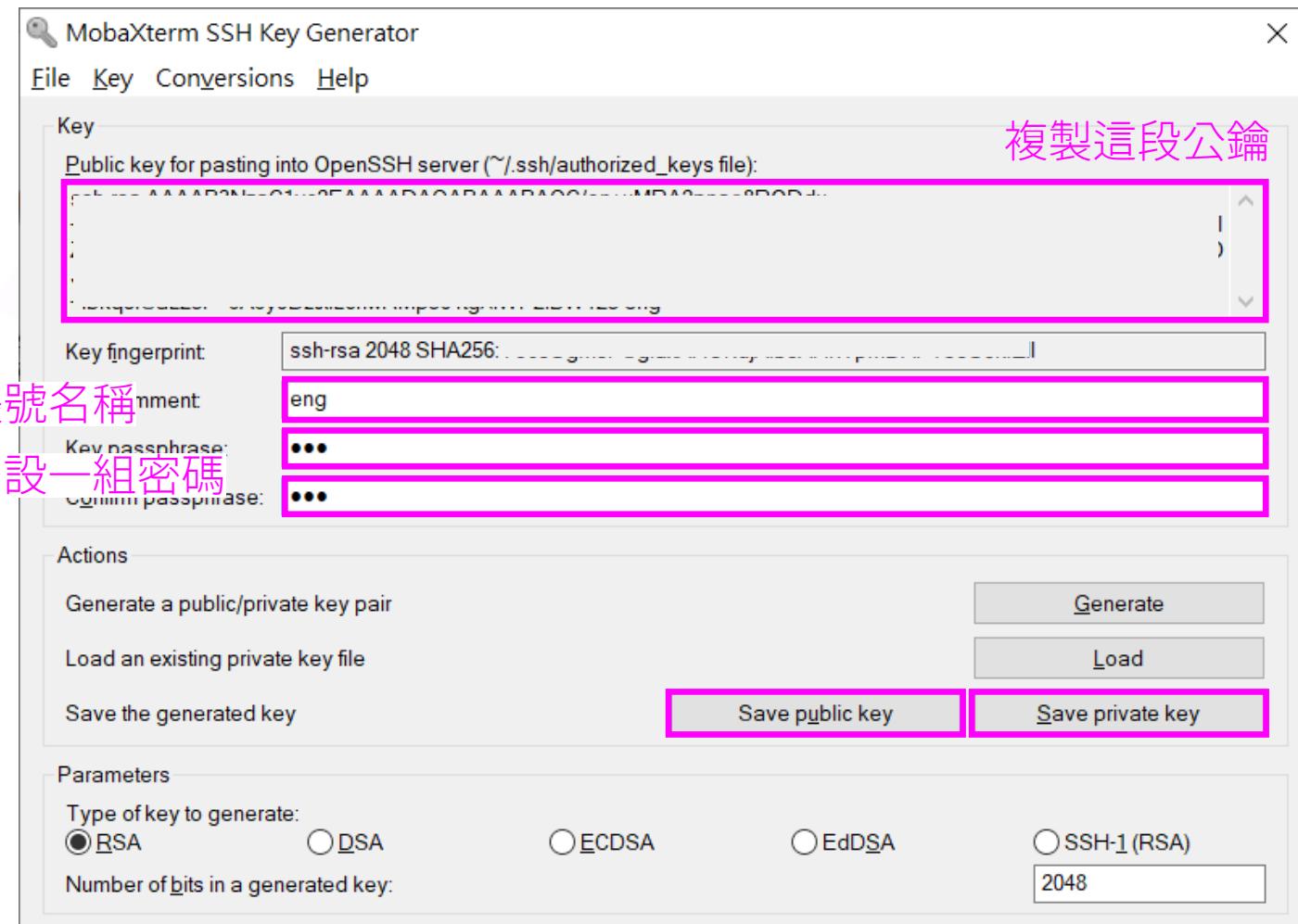
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

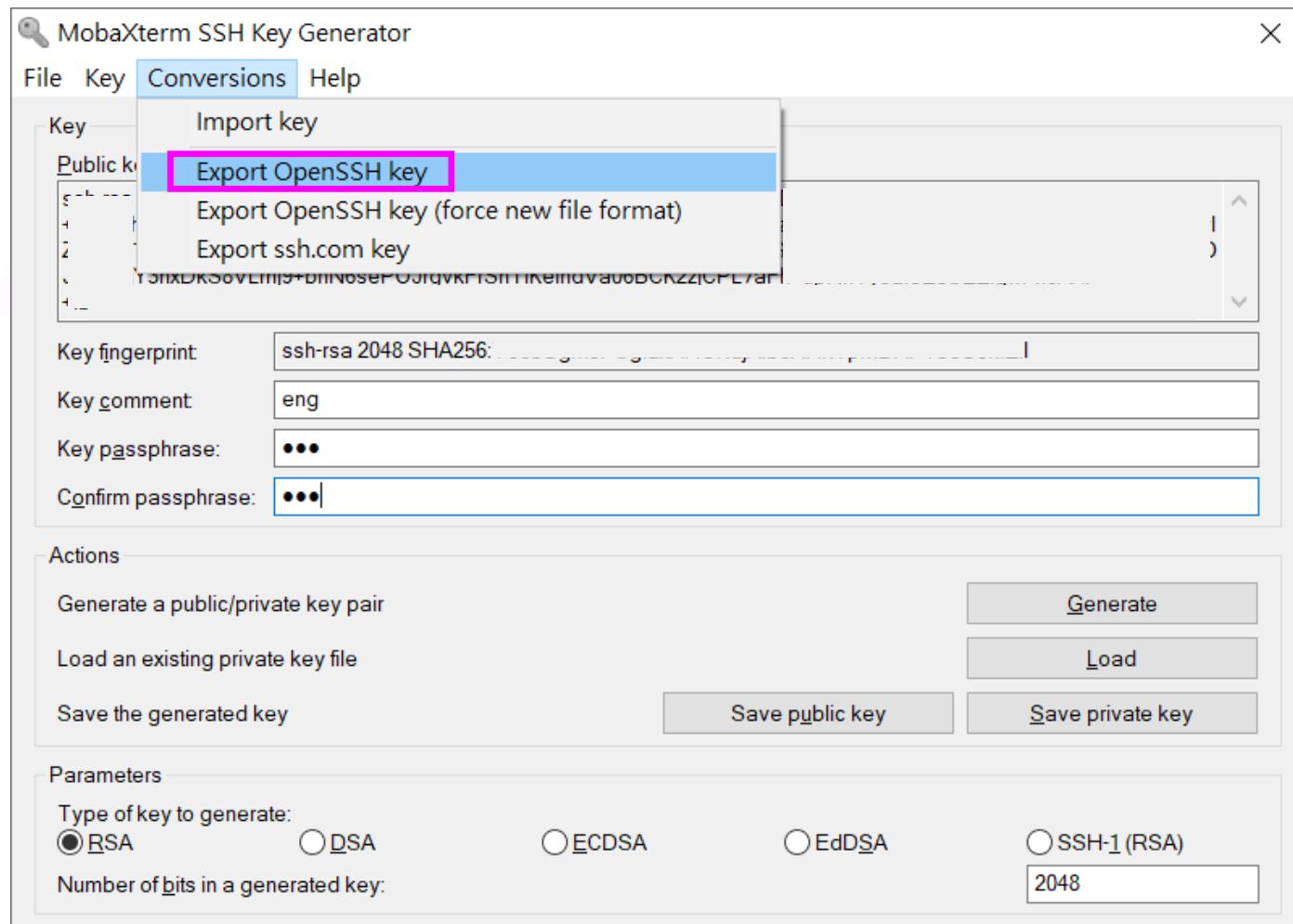


# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

匯出 OpenSSH 格式的 Private Key 稍後於 Cloud Shell 使用；建議儲存為 .pem



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

The screenshot shows the Google Cloud Platform interface for creating a new VM instance. The left sidebar is collapsed, and the main area displays the Compute Engine dashboard. A modal window is open, prompting for VM configuration details. The 'Compute Engine' menu item in the sidebar is highlighted with a pink rectangle.

Google Cloud test

Cloud 總覽

近期

查看所有產品

已固定

API 和服務

帳單

IAM 與管理

Marketplace

Compute Engine

Kubernetes Engine

Cloud Storage

BigQuery

虛擬私有雲網路

Cloud Run

SQL

安全性

Google 地圖平台

儲存空間

磁碟

快照

映像檔

執行個體群組

執行個體群組

健康狀態檢查

VM 管理員

OS 修補程式管理服務

OS 設定管理服務

BARE METAL 解決方案

伺服器

網路

磁碟區

NFS 共用

設定

中繼資料

可用區

網路端點群組

作業

安全性掃描

設定

建立執行個體

匯入 VM

重新整理

開始/繼續

停止

暫停

重設

選取執行個體

PERMISSIONS

LABELS

MONITORING

請至少選取一項資源。

區域 建議 使用者 內部 IP 外部 IP 連線

us-west1-b		10.138.0.3 (nic0)	35.227.191.94 (nic0)	SSH	⋮
------------	--	----------------------	-------------------------	-----	---

▲ HIDE

https://console.cloud.google.com/compute/metadata?authuser=3&project=scenic-helper-355304

# 架設 VM - GCP

## 5. 準備開發環境

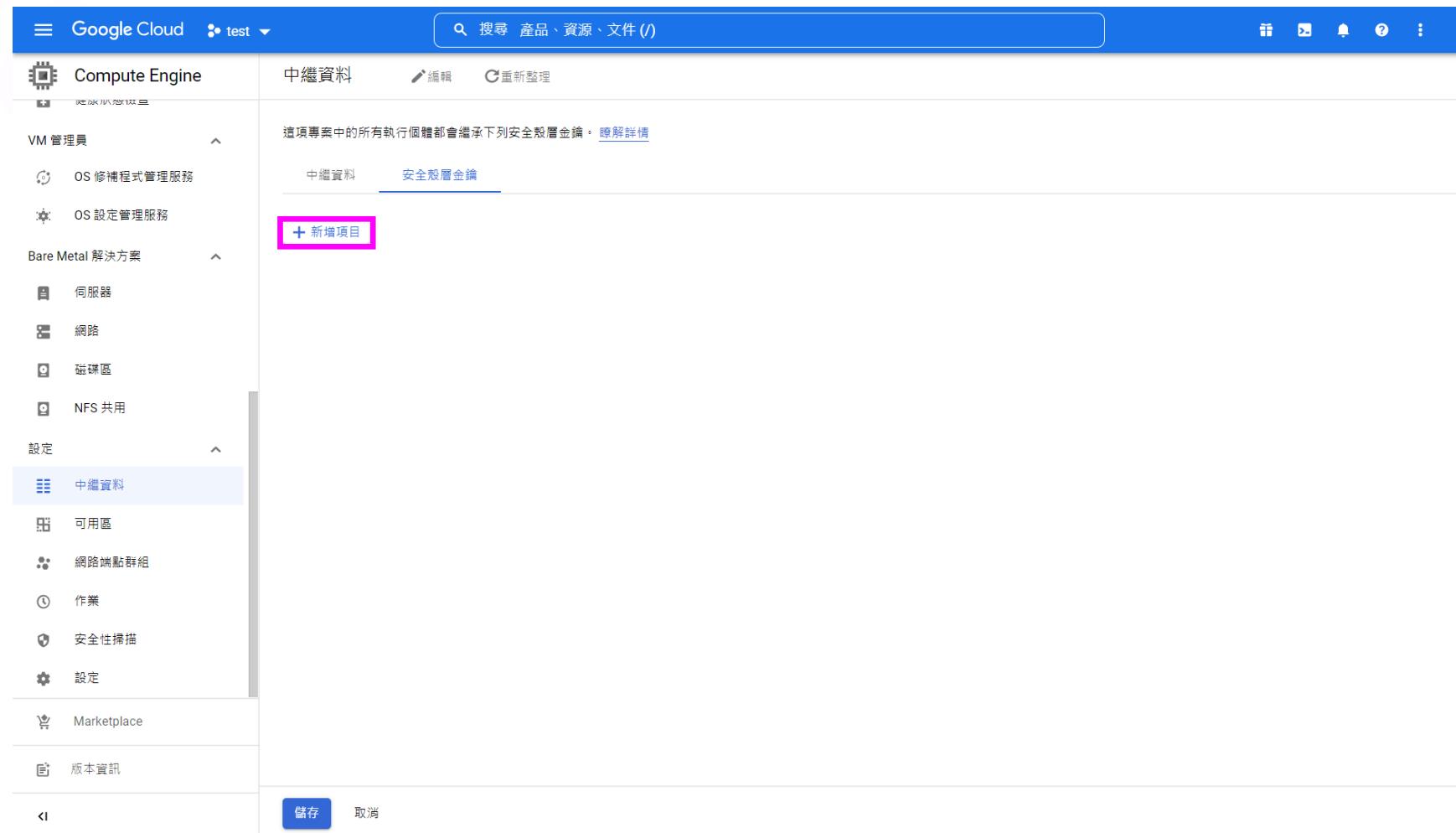
### a. Windows 10

The screenshot shows the Google Cloud Platform (GCP) Compute Engine interface. The left sidebar menu includes 'Compute Engine', 'VM 管理員', 'Bare Metal 解決方案', '設定', and 'Marketplace'. Under '設定', '中繼資料' is selected. The main content area displays a 'Security Keys' section with a sub-section titled '安全殼層金鑰'. A callout box highlights the '新增安全殼層金鑰' (Add Security Key) button. The interface features a blue header bar with the GCP logo and a search bar.

# 架設 VM - GCP

## 5. 準備開發環境

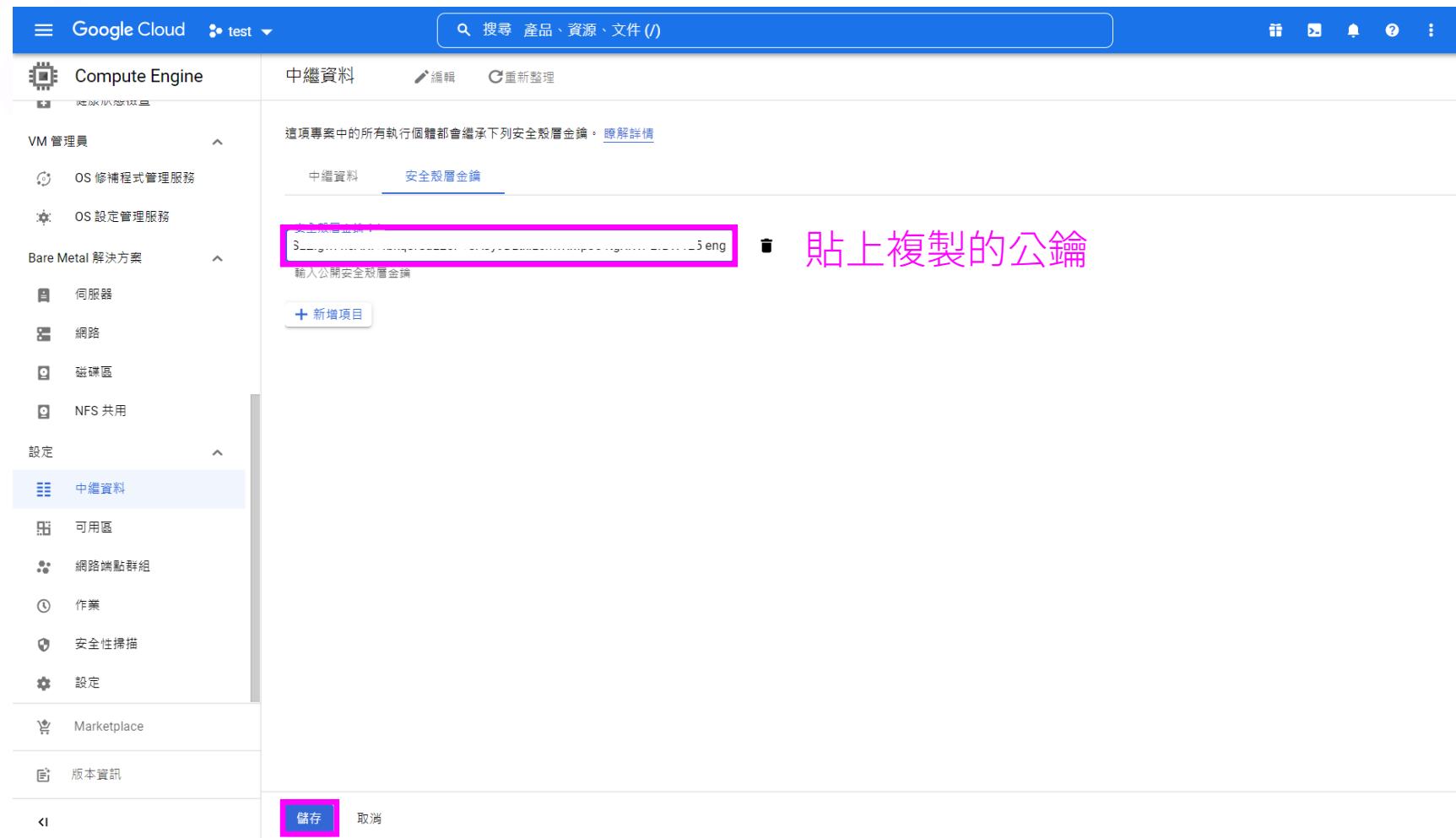
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

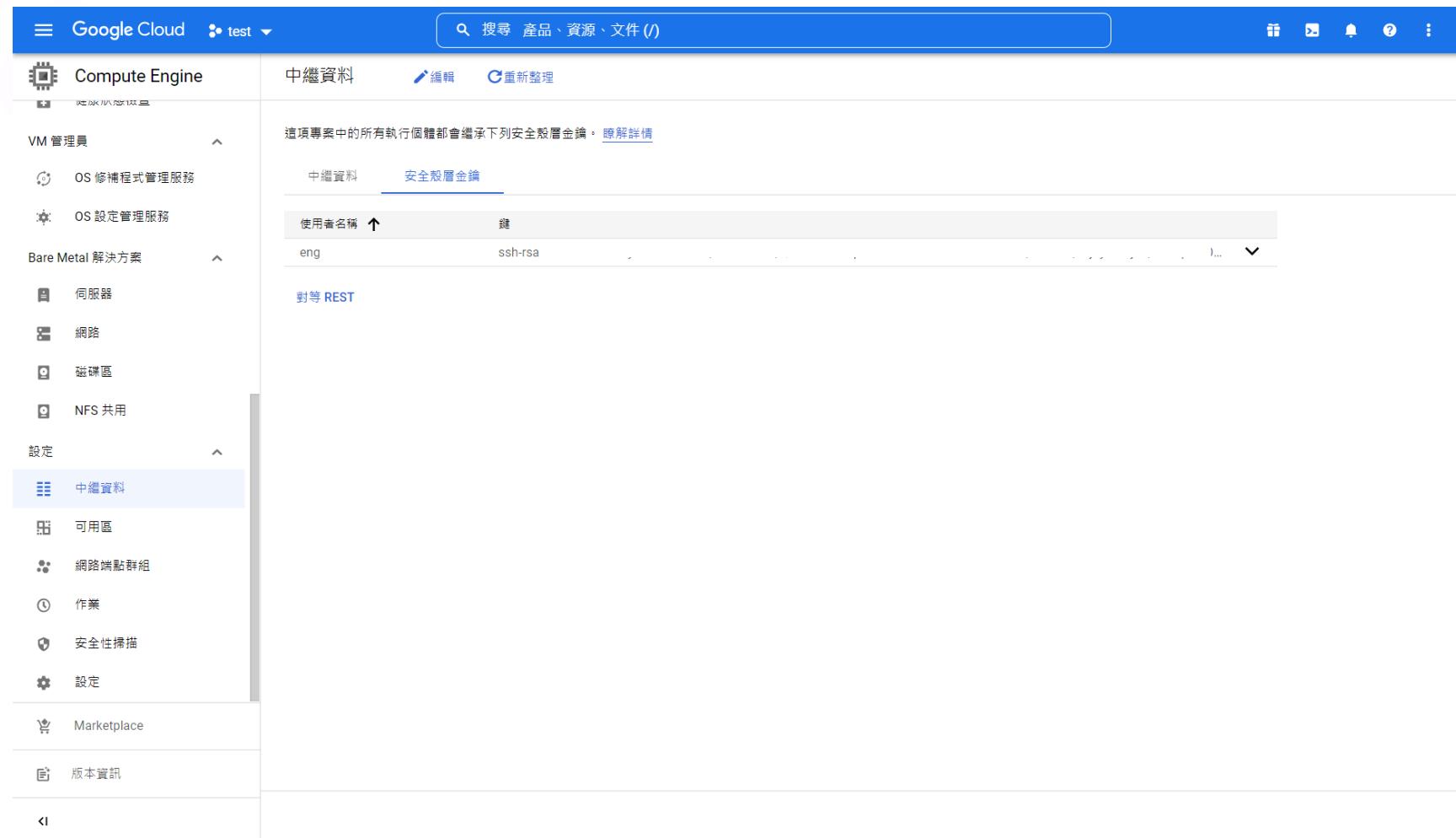


The screenshot shows the Google Cloud Compute Engine interface. On the left, the navigation pane includes sections for VM 管理員, Bare Metal 解決方案, 設定, and Marketplace. Under 設定, 中繼資料 is selected. In the main content area, the '安全殼層金鑰' tab is active, showing a list of keys. One key, '3...eng', is highlighted with a pink rectangle. To the right of the interface, the text '貼上複製的公鑰' is displayed in pink.

# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10



The screenshot shows the Google Cloud Platform Compute Engine interface. The left sidebar navigation includes:

- Compute Engine
- VM 管理員
  - OS 修補程式管理服務
  - OS 設定管理服務
- Bare Metal 解決方案
  - 伺服器
  - 網路
  - 磁碟區
  - NFS 共用
- 設定
  - 中繼資料
  - 可用區
  - 網路端點群組
  - 作業
  - 安全性掃描
  - 設定
- Marketplace
- 版本資訊

The main content area displays the "中繼資料" tab under "安全殼層金鑰". It lists a single key entry:

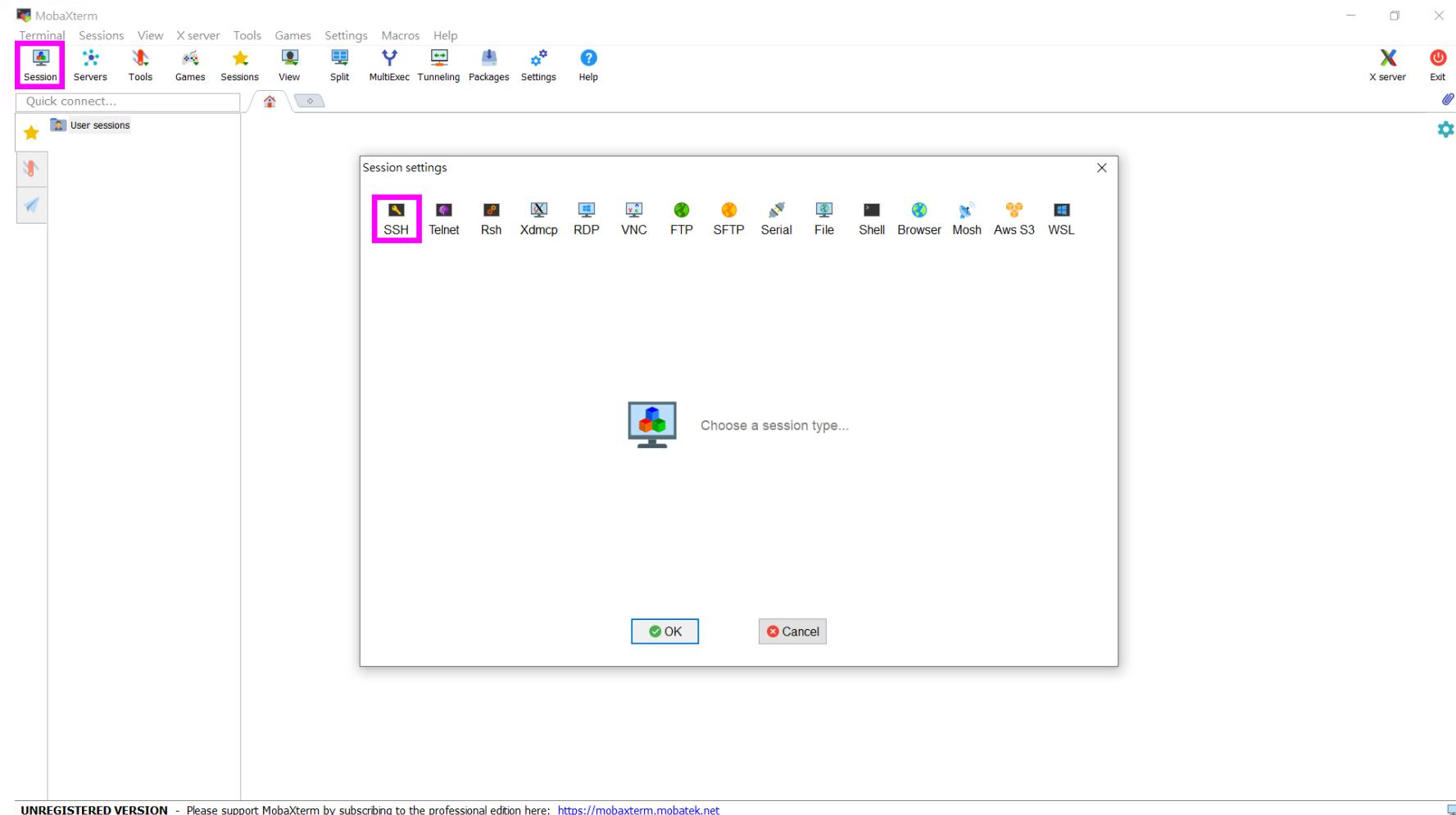
使用者名稱	鍵
eng	ssh-rsa

Below the table, there is a link to "對等 REST".

# 架設 VM - GCP

## 5. 準備開發環境

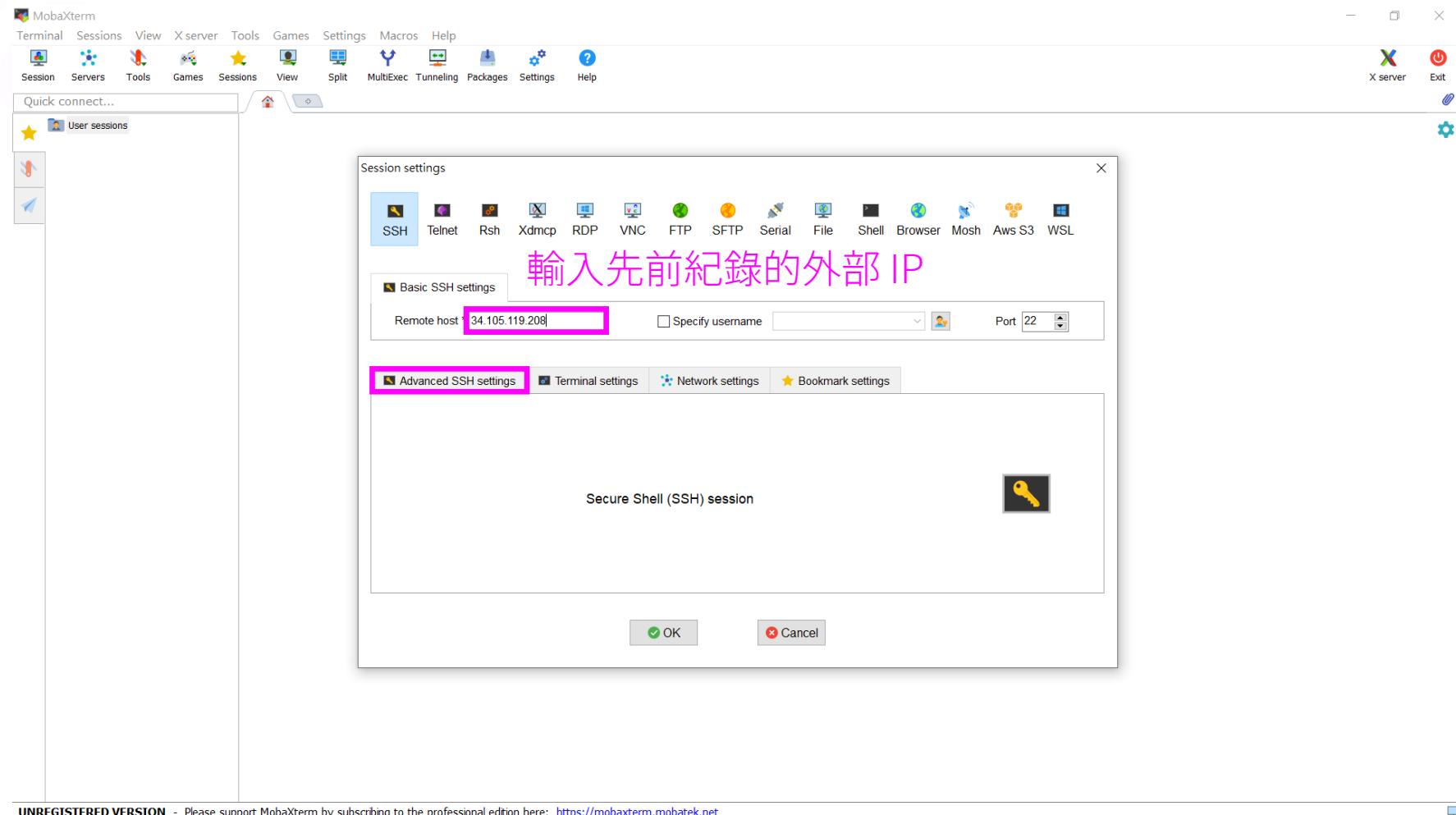
a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

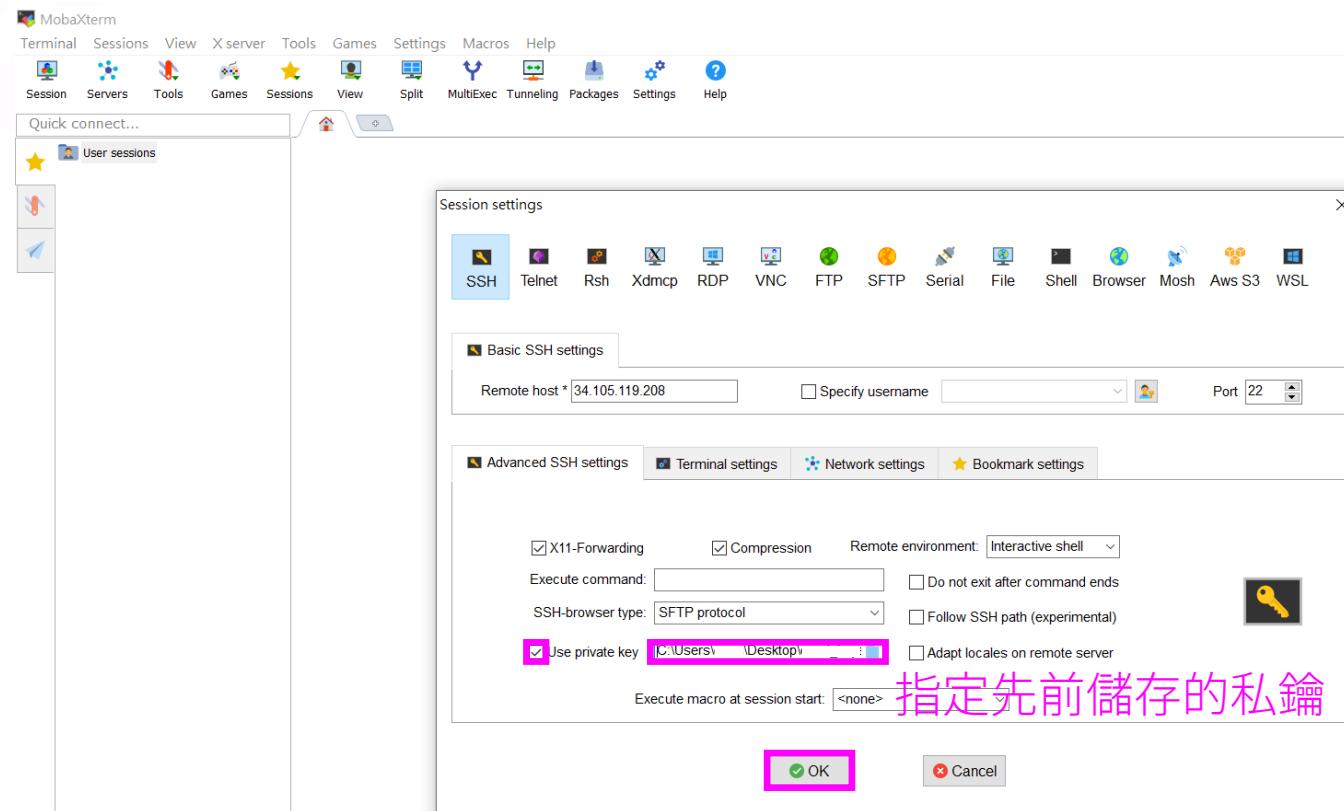
a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

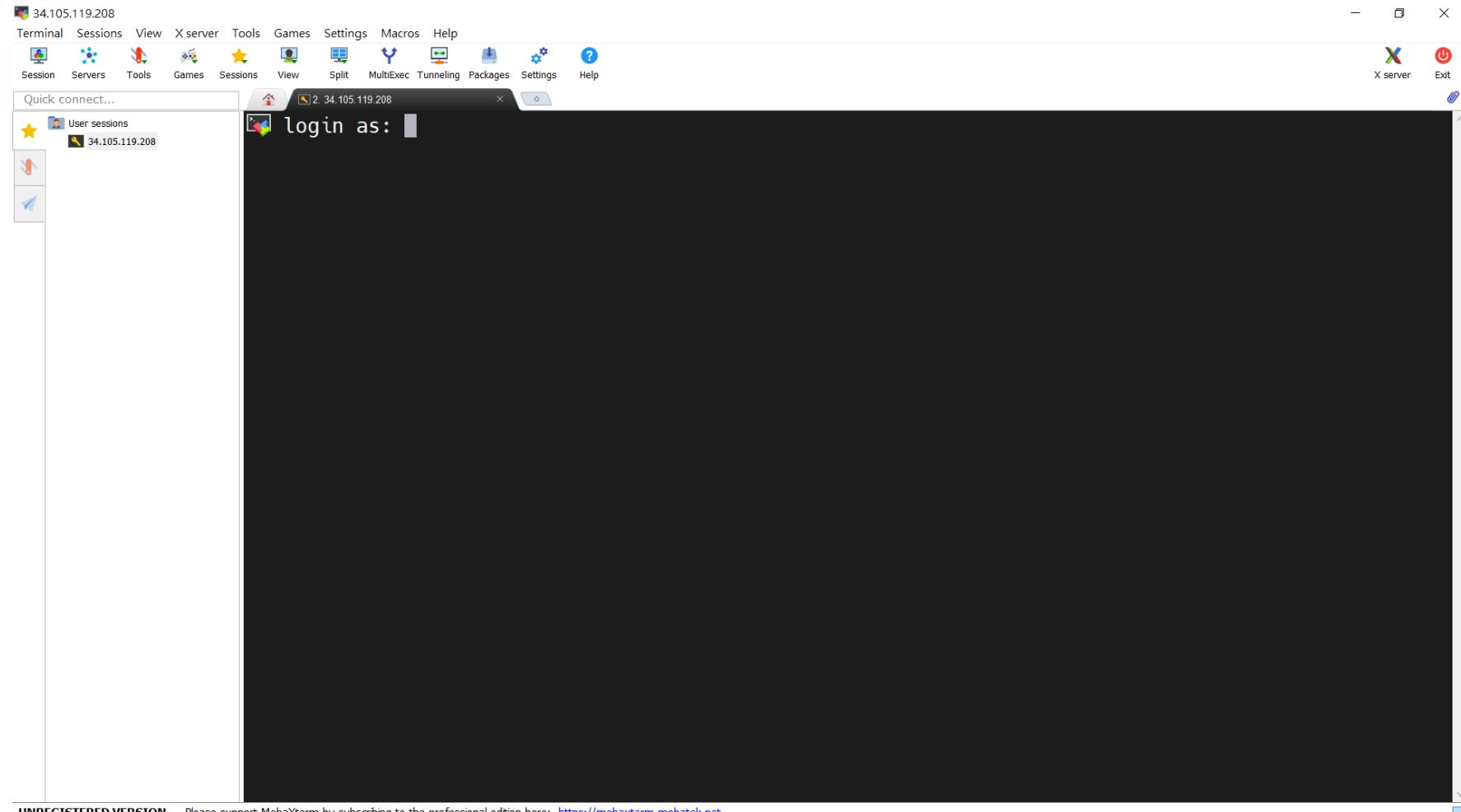
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

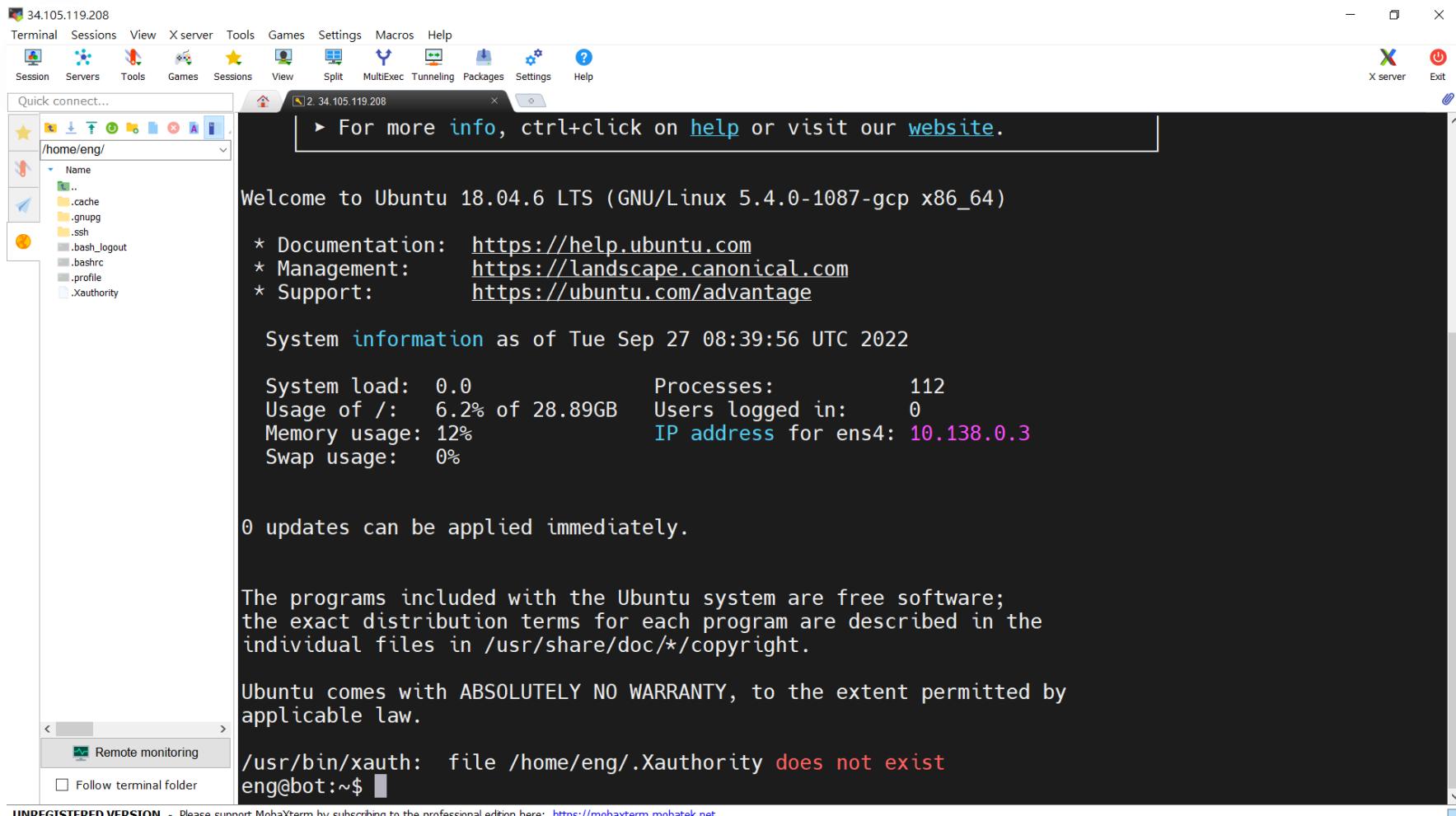
a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### b. macOS/ Linux/ Cloud Shell

- ① 於 MAC 或 Linux 終端機建立金鑰

開啟 終端機

```
ssh-keygen -f your_keyfile -C your_account
ssh-keygen -f eng -C eng # 可選擇追加密碼或留空
```

產出 *your\_keyfile* (私鑰) 以及 *your\_keyfile.pub* (公鑰)

# 架設 VM - GCP

## 5. 準備開發環境

### b. macOS/ Linux/ Cloud Shell

#### ② 於 VM 匯入公鑰

複製 *your\_keyfile.pub* 內容

```
cat your_keyfile.pub
```

```
cat eng.pub
```

貼入 GCE > 中繼資料 > 安全殼層金鑰 ([參閱](#))

# 架設 VM - GCP

## 5. 準備開發環境

b. macOS/ Linux/ Cloud Shell

③ 以 SSH 登入 VM

開啟 終端機

(`chmod 400 your_keyfile`)

(`chmod 400 eng`)

`ssh -i your_keyfile your_account@your_vmidp`

`ssh -i eng eng@34.105.119.208 # 有詢問請輸入 yes`

# 架設 VM - GCP

## 5. 準備開發環境

b. macOS/ Linux/ Cloud Shell

④ 以 SCP 複製資料至 VM

開啟 終端機

```
scp -i your_keyfile your_account@your_vmip:path/to/file client/path  
scp -i eng eng@34.105.119.208:/home/eng/test.log .
```

# 架設 VM - GCP

## 5. 準備開發環境

### c. GCP Cloud Shell

#### ① SSH 登入主機

開啟 Cloud Shell

```
gcloud compute ssh your_account@your_vm
```

授權 Cloud Shell: 授權 > generate key: Y > passphrase twice > Zone: n

# 架設 VM - Azure

## 1. 登入

<https://portal.azure.com>

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for account settings and help. Below the header, the main content area is titled "Azure 服務" (Azure Services) and features a row of service icons: 建立資源 (Create Resource), 虛擬機器 (Virtual Machines), 容器執行個體 (Container Instances), 容器登錄 (Container Registry), App Service 方案 (App Service Plan), 防火牆原則 (Firewall Rules), 應用程式服務 (App Service), 資源群組 (Resource Groups), 認知服務 (Cognitive Services), and a "更多服務" (More Services) button. Underneath this, there's a section titled "Resources" with tabs for "Recent" and "Favorite". It displays a message: "No resources have been viewed recently" with a "View all resources" button. Further down, there are sections for "瀏覽" (Browse) with links to 訂用帳戶 (Billing Account), 資源群組 (Resource Groups), 所有資源 (All Resources), and 儀表板 (Dashboard); and "工具" (Tools) with links to Microsoft Learn, Azure 監視器 (Azure Monitor), 適用於雲端的 Microsoft Defender (Microsoft Defender for Cloud), and 成本管理 (Cost Management). A footer bar at the bottom contains the URL "https://portal.azure.com/#create/hub".

# 架設 VM - Azure

## 2. 建立 VM

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Azure logo, a search bar, and various icons for account settings and help. Below the navigation bar, the main content area has a title 'Azure 服務' (Azure Services). On the left, there's a sidebar with '建立資源' (Create Resource) and '虛擬機器' (Virtual Machine) buttons. The main content area is titled '虛擬機器' (Virtual Machine) and shows a list of recent resources. One item, 'Azure 虛擬機器' (Azure Virtual Machine), is highlighted with a pink rectangle. This item has a sub-menu with options like '建立' (Create), '檢視' (View), and '具有預設設定的 Azure 虛擬機器' (Azure Virtual Machine with pre-defined settings). Other items in the list include 'Azure Arc 虛擬機器' (Azure Arc Virtual Machine) and 'Azure VMWare 解決方案虛擬機器' (Azure VMWare Solution Virtual Machine). Below the list, there's a section for 'viewed recently' and a 'resources' button. At the bottom of the page, there are sections for '訂用帳戶' (Subscription), '資源群組' (Resource Group), '所有資源' (All Resources), '儀表板' (Dashboard), 'Microsoft Learn' (with a link to '透過 Microsoft 推出的免費線上訓練了解 Azure'), 'Azure 監視器' (Azure Monitor), '適用於雲端的 Microsoft Defender' (Microsoft Defender for Cloud), and '成本管理' (Cost Management).

# 架設 VM - Azure

## 2. 建立 VM

The screenshot shows the 'Create Virtual Machine' wizard in the Microsoft Azure portal. The 'Basic' tab is selected. Key configuration options highlighted in pink are:

- VM Name:** bot
- Region:** (US) West US
- Image:** Ubuntu Server 18.04 LTS - Gen2
- Processor Architecture:** x64

Other visible fields include:

- Subscription:** Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d1f3)
- Resource Group:** (新增) 資源群組 (New)
- Availability Options:** 不需要基礎結構備援 (No infrastructure backup required)
- Security Type:** 標準 (Standard)
- VM Size:** Standard\_B2I\_v3 (1 vCore, 2.5 GiB 記憶體 / 每日 \$1.50€ 77)

Annotations on the right side of the interface:

- Setting the host machine name: 設定主機名稱
- Selecting the region: 選擇區域
- Picking the operating system: 選擇作業系統

# 架設 VM - Azure

## 2. 建立 VM

Microsoft Azure

首頁 > 虛擬機器 >

建立虛擬機器

大小 \*

Standard\_DS1\_v2 - 1 個 vcpu, 3.5 GiB 記憶體 (每月 \$1,535.77)

Administrator 帳戶

驗證類型 \*

SSH 公開金鑰 (selected)

密碼

Azure 現在會自動為您產生 SSH 公開金鑰組，並允許您儲存以供未來使用。這是一種快速、簡單且安全的方式，讓您可以連線至虛擬機器。

使用者名稱 \*

azureuser

SSH 公開金鑰來源

產生新的金鑰組

金鑰組名稱 \*

bot\_key

輸入連接埠規則

選取可從公用網際網路存取的虛擬機器網路連接埠。您可以在 [網路] 索引標籤上指定限制範圍更小或更精確的網路存取。

公用輸入連接埠 \*

無 (unchecked)

允許選取的連接埠 (selected)

選取輸入連接埠 \*

SSH (22)

⚠️ 這可讓所有 IP 位址存取您的虛擬機器。建議您只將此項用於測試。使用 [網路功能] 索引標籤中的 [進階] 控制項可建立規則，限制輸入流量只能來自自己知的 IP 位址。

檢閱 + 建立

< 上一步

下一步：磁碟 >

提供意見反應

# 架設 VM - Azure

## 2. 建立 VM

Microsoft Azure

搜尋資源、服務及文件 (G+/)

首頁 > 虛擬機器 >

建立虛擬機器

大小 \* ① Standard\_DS1\_v2 - 1 個 vcpu, 3.5 GiB 記憶體 (每月 \$1,535.77) [查看所有大小](#)

Administrator 帳戶

驗證類型 ① SSH 公開金鑰

Azure 現在會自動為您產生 SSH 公開金鑰組，並允許您儲存以供未來使用。這是一種快速、簡單且安全的方式，讓您可以連線至虛擬機器。

使用者名稱 \* ① eng

SSH 公開金鑰來源 使用現有的公開金鑰

SSH 公開金鑰 \* ① eng

輸入連接埠規則

選取可從公用網際網路存取的虛擬機器網路連接埠。您可以在 [網路] 索引標籤上指定限制範圍更小或更精確的網路存取。

公用輸入連接埠 \* ① 無

允許選取的連接埠

選取輸入連接埠 \* SSH (22)

這可讓所有 IP 位址存取您的虛擬機器。建議您只將此項用於測試。使用 [網路功能] 索引標籤中的 [進階] 控制項可建立規則，限制輸入流量只能來自已知的 IP 位址。

檢閱 + 建立 < 上一步 下一步：磁碟 > 提供意見反應

自訂帳號  
金鑰建立方式，選擇現有  
貼上公鑰

# 架設 VM - Azure

## 2. 建立 VM

The screenshot shows the 'Create Virtual Machine' wizard in the Microsoft Azure portal. The current step is '建立虛擬機器' (Create Virtual Machine). The configuration includes:

- 大小**: Standard\_DS1\_v2 - 1 個 vcpu, 3.5 GiB 記憶體 (每月 \$1,535.77)
- Administrator 帳戶**:
  - 驗證類型: SSH 公開金鑰 (selected)
  - 使用者名稱: eng
  - SSH 公開金鑰來源: 使用現有的公開金鑰
  - SSH 公開金鑰: A dropdown menu showing existing public keys, with one selected.
- 輸入連接埠規則**:
  - 選取可從公用網際網路存取的虛擬機器網路連接埠。可以在 [網路] 索引標籤上指定限制範圍更小或更精確的網路存取。
  - 公用輸入連接埠: 允許選取的連接埠 (selected)
  - 選取輸入連接埠: HTTP (80), HTTPS (443), SSH (22) (selected)

**HTTP for Certbot**  
**HTTPS for LINE Messaging**

At the bottom, there are buttons for '檢閱 + 建立' (Review + Create) and '下一步: 磁碟' (Next: Disks).

# 架設 VM - Azure

## 2. 建立 VM

The screenshot shows the Microsoft Azure VM creation process. At the top, the navigation bar includes 'Microsoft Azure', a search bar ('搜尋資源、服務及文件 (G+)'), and various icons for account and settings. The main title is '建立虛擬機器' (Create Virtual Machine). A green success message '驗證成功' (Verification successful) is displayed. Below it, a navigation bar has tabs: '基本' (Basic), '磁碟' (Disk), '網路' (Network), '管理' (Management), 'Monitoring', '進階' (Advanced), '標籤' (Tags), and '檢閱 + 建立' (Review + Create), with '檢閱 + 建立' currently selected. A note below the tabs states: '以下提供的成本是估計值，而不是最終價格。請使用 [定價計算機](#) 以滿足您所有的定價需求。' (The cost provided is an estimate, not the final price. Please use the [Pricing calculator](#) to meet all your pricing needs.). The 'PRODUCT DETAILS' section shows '1 X Standard DS1 v2 by Microsoft' and 'Subscription credits apply'. It also lists the price '2.1038 TWD/hr' and links to 'Terms of use' and 'Privacy policy'. The 'TERMS' section contains legal text about agreeing to terms and privacy statements. A warning message at the bottom left says: '⚠ 您已將 SSH 個連接埠設定為網際網路。建議您只將此項用於測試。若要變更此設定，請回到 [基本] 索引標籤。' (Warning: You have set the SSH port to Internet. It is recommended to use this setting only for testing. If you want to change this setting, please go back to the [Basic] index tab.). At the bottom, there are tabs for '基本' (Basic), '訂用帳戶' (Subscription), 'Azure Pass - 贊助' (Azure Pass - Sponsor), and '憑據存取' (Access keys). The '建立' (Create) button is highlighted in blue, while the other buttons are greyed out.

# 架設 VM - Azure

## 2. 建立 VM

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons. Below the navigation bar, the page title is "CreateVm-Canonical.UbuntuServer-18\_04-Its-gen2-20220927230325 | 概觀". On the left, there's a sidebar with options like "概觀", "輸入", "輸出", and "範本". The main content area displays a success message: "您的部署已完成" (Deployment completed successfully). It shows deployment details: Deployment name: CreateVm-Canonical.UbuntuServer-18\_04-Its-gen2-2022..., Start time: 27/9/2022 下午11:13:58, User account: Azure Pass - 賽助 (0fdd0a2d-5afb-4589-8206-0a7eae00d...), Resource group: bot\_group\_09272303. There are two expandable sections: "部署詳細資料" (Deployment details) and "後續步驟" (Next steps). Under "後續步驟", there are three links: "設定自動關機" (建议), "監視 VM 健康情況、效能與網路相依性" (建議), and "在虛擬機器內執行指令碼" (建議). At the bottom, there are two buttons: "前往資源" (Go to resources) and "建立另一個 VM" (Create another VM). On the right side of the screen, there are several promotional cards: "成本管理" (Cost management), "適用於雲端的 Microsoft Defender" (Microsoft Defender for Cloud), "免費 Microsoft 教學課程" (Free Microsoft training courses), and "諮詢專家" (Consultant experts).

# 架設 VM - Azure

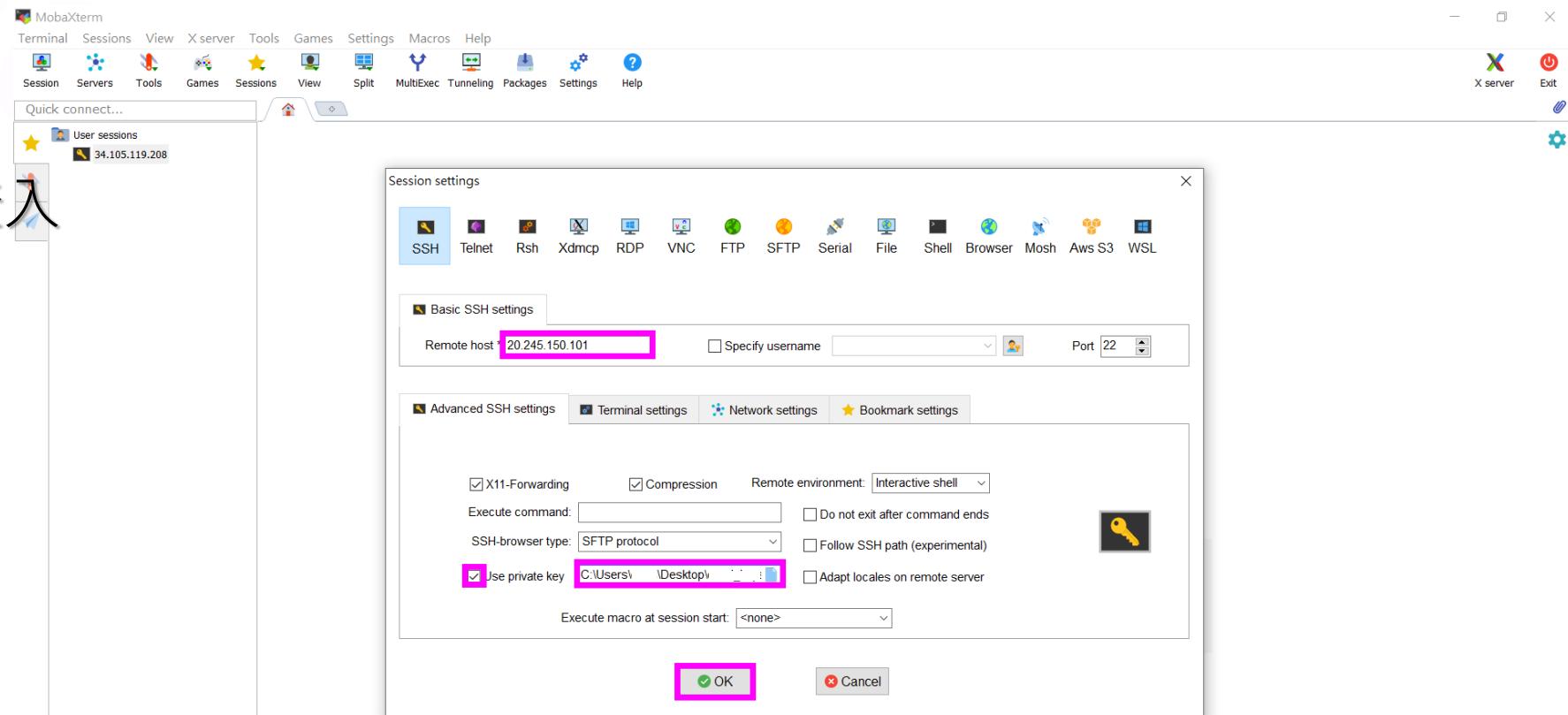
## 2. 建立 VM

The screenshot shows the Microsoft Azure portal interface for managing a virtual machine named 'bot'. The top navigation bar includes the Azure logo, search bar, and various management icons. The main content area displays the VM's overview, including its resource group, status (Running), location (West US), and network information. A pink box highlights the 'Public IP Address' field, which contains the value '20.245.150.101'. To the right of this field, the text '記錄外部 IP' (Record external IP) is overlaid in pink. Below the main summary, there are tabs for '属性' (Properties), '监视' (Monitoring), '功能 (7)' (Features), '建议' (Recommendations), and '教学课程' (Training Courses). Under the '属性' tab, detailed configuration settings are listed for the VM, such as its name ('bot'), operating system ('Linux (ubuntu 18.04)'), publisher ('Canonical'), offer ('UbuntuServer'), sku ('18\_04-Its-gen2'), and size ('Standard DS1 v2'). Other sections visible include '网络' (Network), '大小' (Size), and '磁盘' (Disk). The bottom of the page features a green footer bar with the text '© 2022 Enos Chou'.

# 架設 VM - Azure

## 3. 準備開發環境

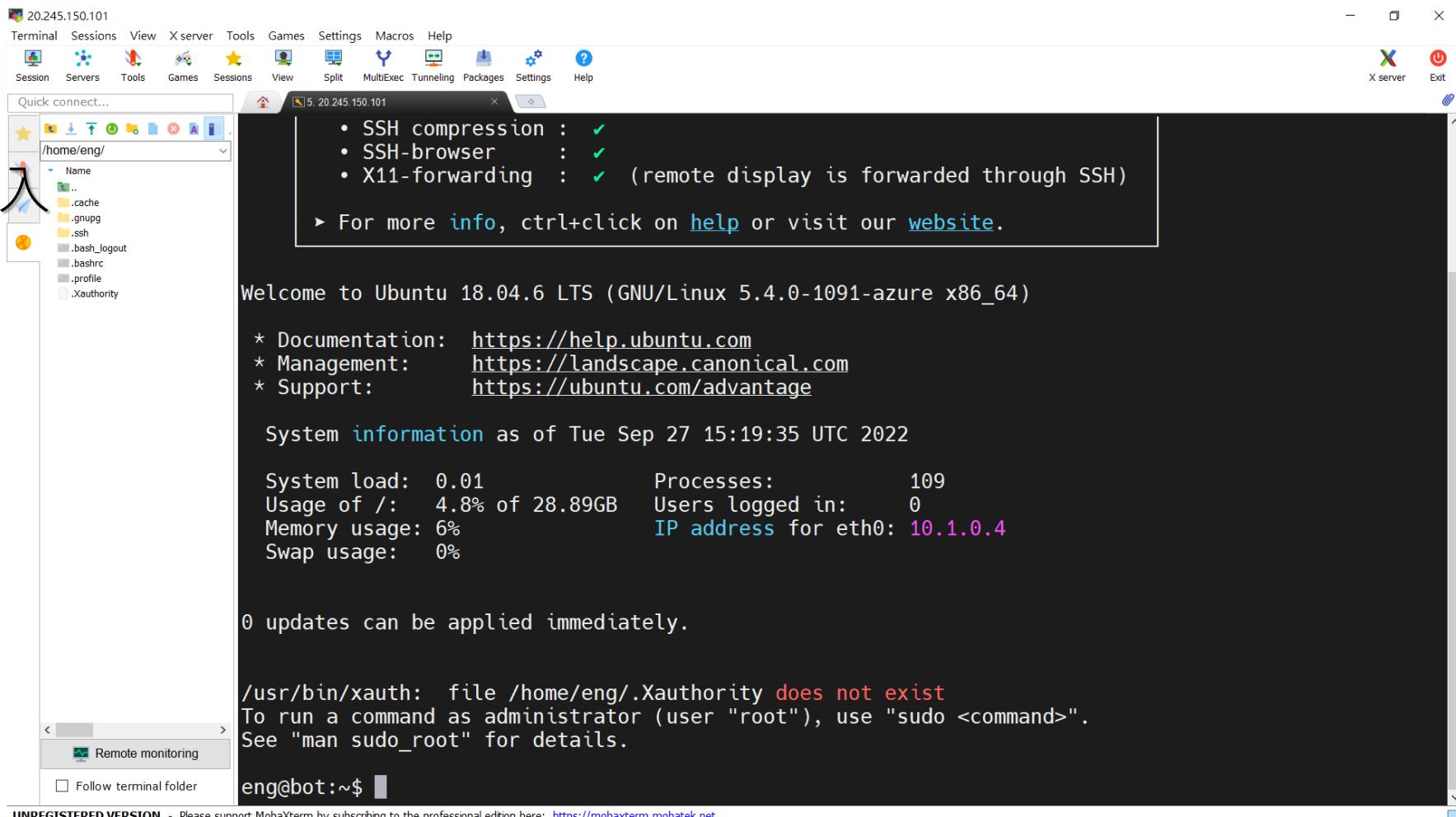
以 MobaXterm 登入



# 架設 VM - Azure

## 3. 準備開發環境

以 MobaXterm 登入



20.245.150.101

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

X server Exit

Quick connect... /home/eng/

Name .. .cache .grupg .ssh .bash\_logout .bashrc .profile .xauthority

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1091-azure x86\_64)

- SSH compression : ✓
- SSH-browser : ✓
- X11-forwarding : ✓ (remote display is forwarded through SSH)

▶ For more info, ctrl+click on help or visit our website.

\* Documentation: <https://help.ubuntu.com>  
\* Management: <https://landscape.canonical.com>  
\* Support: <https://ubuntu.com/advantage>

System information as of Tue Sep 27 15:19:35 UTC 2022

System load: 0.01	Processes: 109
Usage of /: 4.8% of 28.89GB	Users logged in: 0
Memory usage: 6%	IP address for eth0: 10.1.0.4
Swap usage: 0%	

0 updates can be applied immediately.

/usr/bin/xauth: file /home/eng/.Xauthority does not exist  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.

eng@bot:~\$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 架設 VM - Azure

## 3. 準備開發環境

以終端機登入

```
ssh -i your_keyfile your_account@your_vmip
```

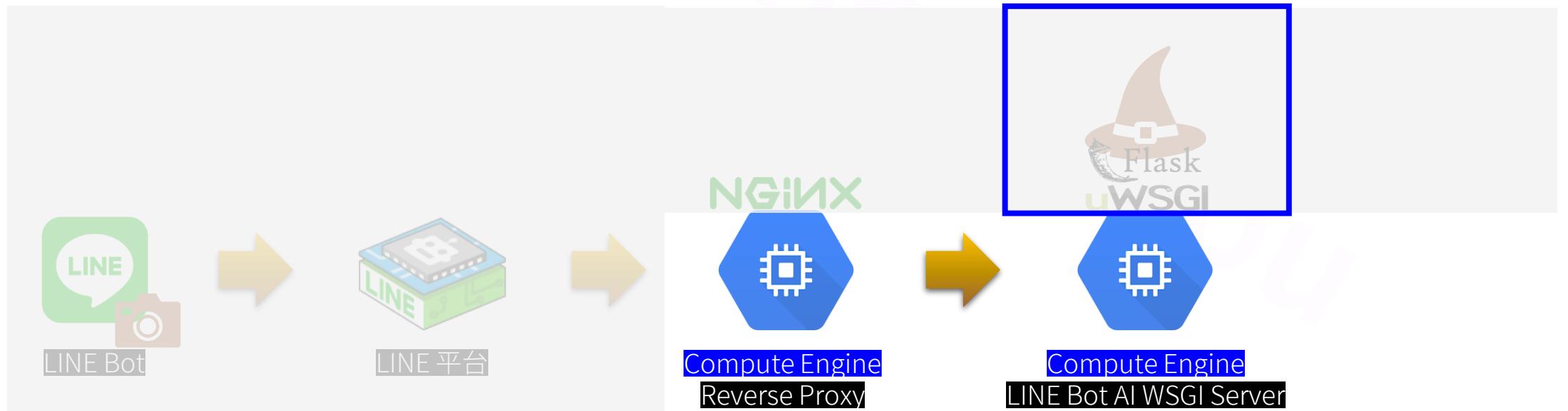
```
ssh -i eng eng@20.245.150.101
      Usage of /: 4.9% of 28.89GB
(ssh -i eng.pem eng@20.245.150.101)
```

### Note

若是將 *your\_keyfile.pem* 上傳至 Cloud Shell 使用，執行 ssh 前須先執行  
chmod 400 *your\_keyfile.pem*

```
Last login: Tue Sep 27 15:47:00 2022 from 34.80.18.167
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

eng@bot:~$
```



# LINE Bot & WSGI

## 1. 準備程式碼

### a. 下載範例程式並調整

- ① treesbot.py
- ② trees17V1.h5
- ③ treeset\_labels.txt
- ④ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_FILE
- ⑤ other tree samples

# LINE Bot & WSGI

## 1. 準備程式碼

### b. 製作 requirements.txt

line-bot-sdk

flask

pillow

tensorflow==2.4.4

uwsgi

# LINE Bot & WSGI

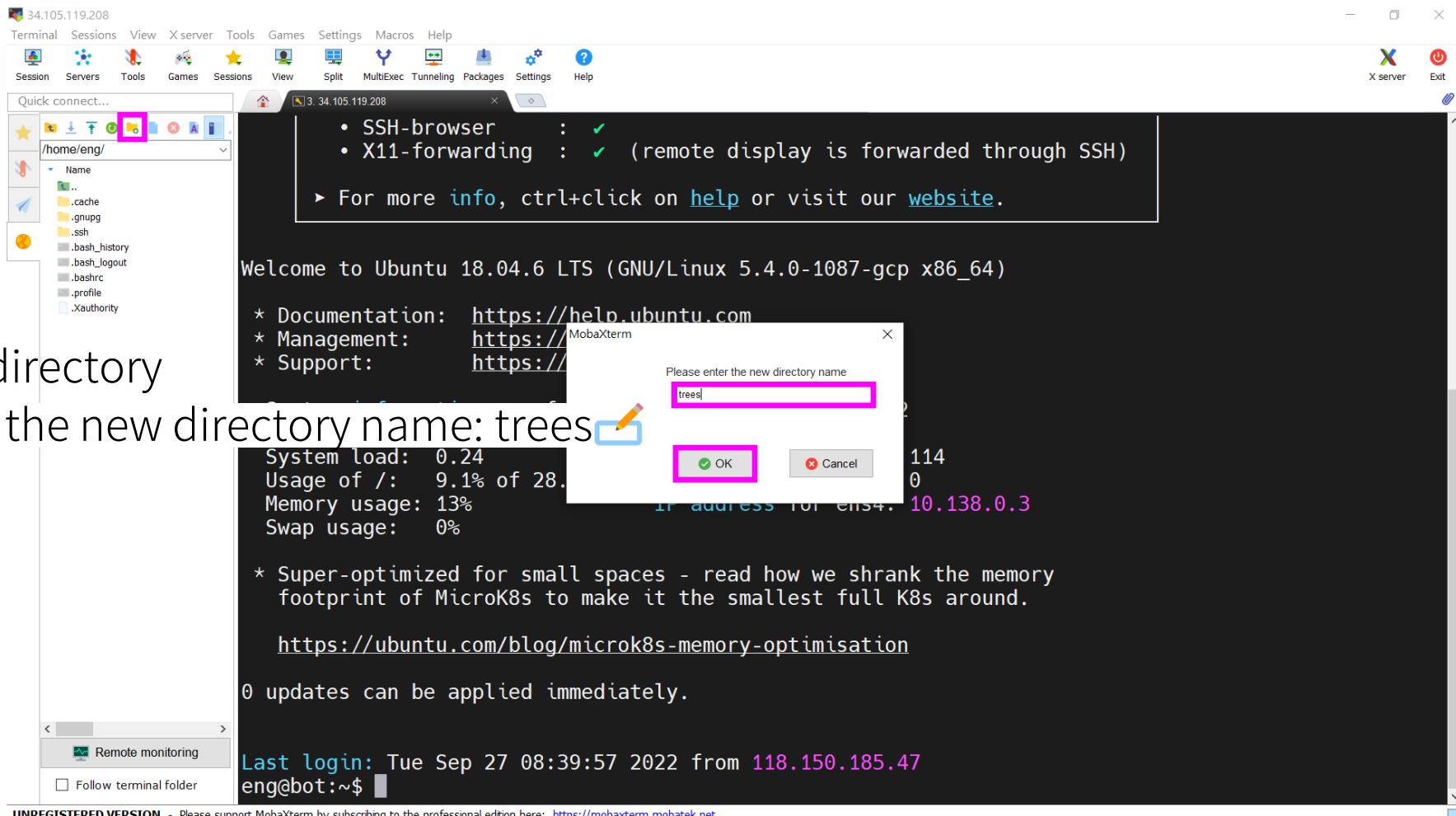
## 1. 準備程式碼

### c. 於 VM 建立專案目錄

① Create new directory

② Please enter the new directory name: trees

③ OK



# LINE Bot & WSGI

## 1. 準備程式碼

### c. 於 VM 建立專案目錄

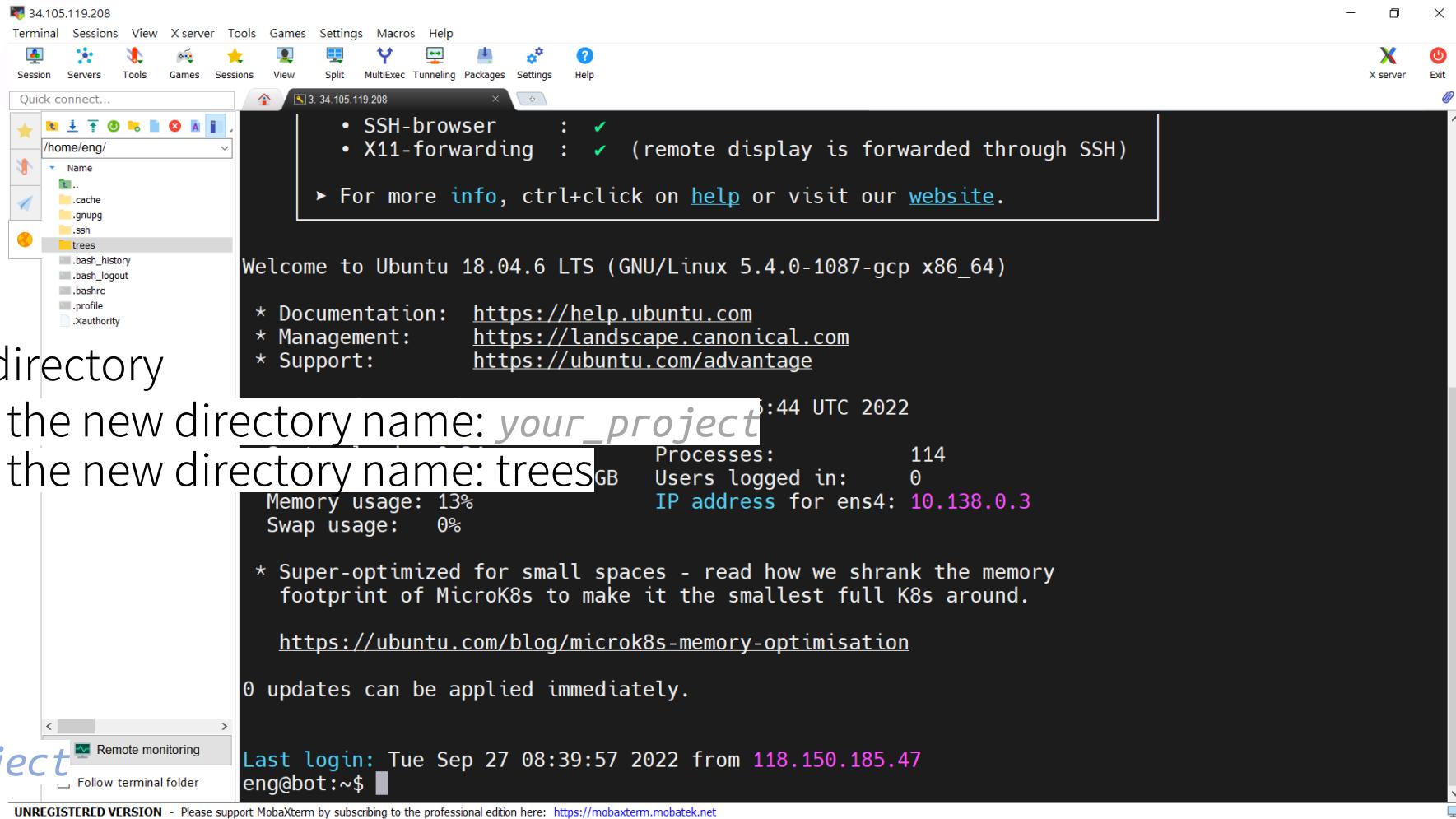
- ① Create new directory
- ② Please enter the new directory name: *your\_project*  
Please enter the new directory name: *trees*
- ③ OK

#### Note

相當於下列指令

`mkdir your_project`

`mkdir trees`



# LINE Bot & WSGI

## 1. 準備程式碼

### c. 於 VM 建立專案目錄

- ① Create new directory
- ② Please enter the new directory name: your\_project  
Please enter the new directory name: trees
- ③ OK
- ④ 進入 trees

```
34.105.119.208
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3. 34.105.119.208
• SSH-browser : ✓
• X11-forwarding : ✓ (remote display is forwarded through SSH)
▶ For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1087-gcp x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

10:44 UTC 2022
Processes: 114
Users logged in: 0
IP address for ens4: 10.138.0.3
Memory usage: 13%
Swap usage: 0%
* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.

Last login: Tue Sep 27 08:39:57 2022 from 118.150.185.47
eng@bot:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# LINE Bot & WSGI

## 1. 準備程式碼

d. 上傳上述檔案  
至專案目錄

- ① treesbot.py
- ② trees17V1.h5
- ③ treeset\_labels.txt
- ④ env.json
- ⑤ requirements.txt

SSH-browser : ✓  
X11-forwarding : ✓ (remote display is forwarded through SSH)  
For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1087-gcp x86\_64)

\* Documentation: <https://help.ubuntu.com>  
\* Management: <https://landscape.canonical.com>  
\* Support: <https://ubuntu.com/advantage>

System information as of Wed Sep 28 13:35:44 UTC 2022

System load: 0.24	Processes: 114
Usage of /: 9.1% of 28.89GB	Users logged in: 0
Memory usage: 13%	IP address for ens4: 10.138.0.3
Swap usage: 0%	

\* Super-optimized for small spaces - read how we shrank the memory footprint of MicroK8s to make it the smallest full K8s around.  
<https://ubuntu.com/blog/microk8s-memory-optimisation>

0 updates can be applied immediately.

Last login: Tue Sep 27 08:39:57 2022 from 118.150.185.47  
eng@bot:~\$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# LINE Bot & WSGI

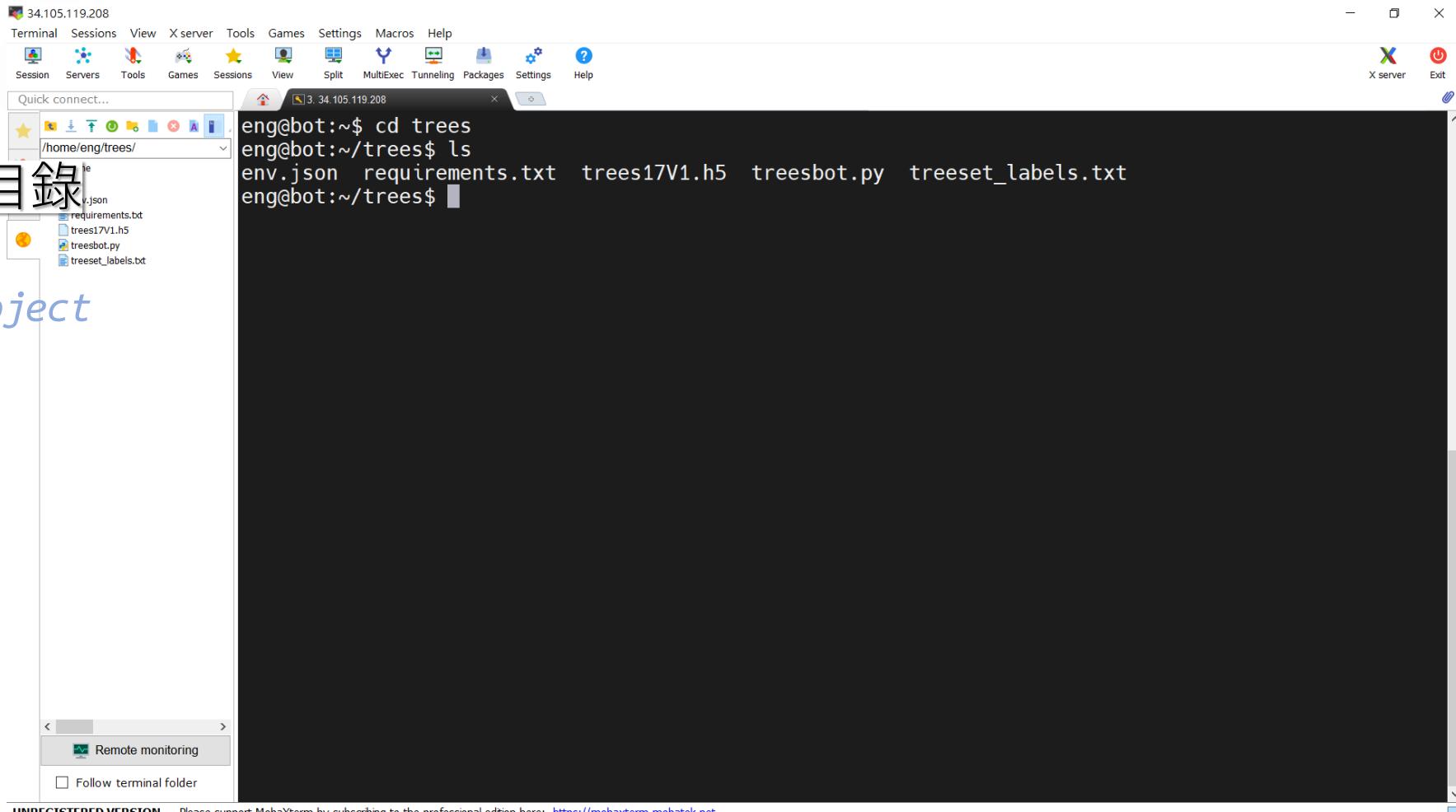
## 2. 部署服務

a. 進入 VM 專案目錄

`cd; cd your_project`

`cd; cd trees`

`ls`



```
eng@bot:~$ cd trees
eng@bot:~/trees$ ls
env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt
eng@bot:~/trees$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

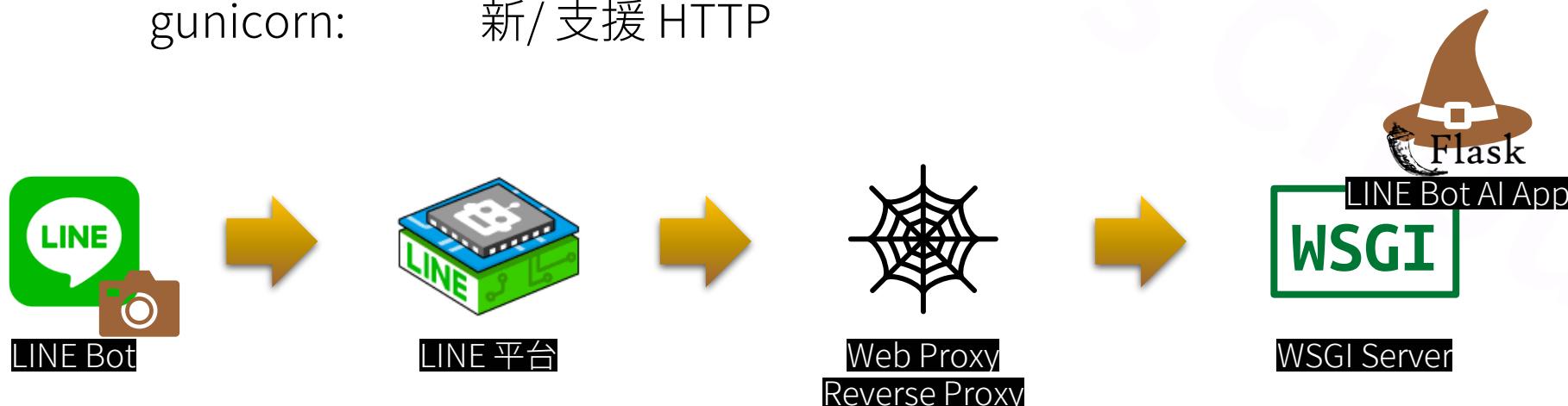
# LINE Bot & WSGI

## WSGI Server 的選擇

uWSGI vs gunicorn

uWSGI: 經典/ 支援 HTTP/ 搭配 NGINX 支援 uwsgi protocol

gunicorn: 新/ 支援 HTTP



# LINE Bot & WSGI

## WSGI Server 的選擇

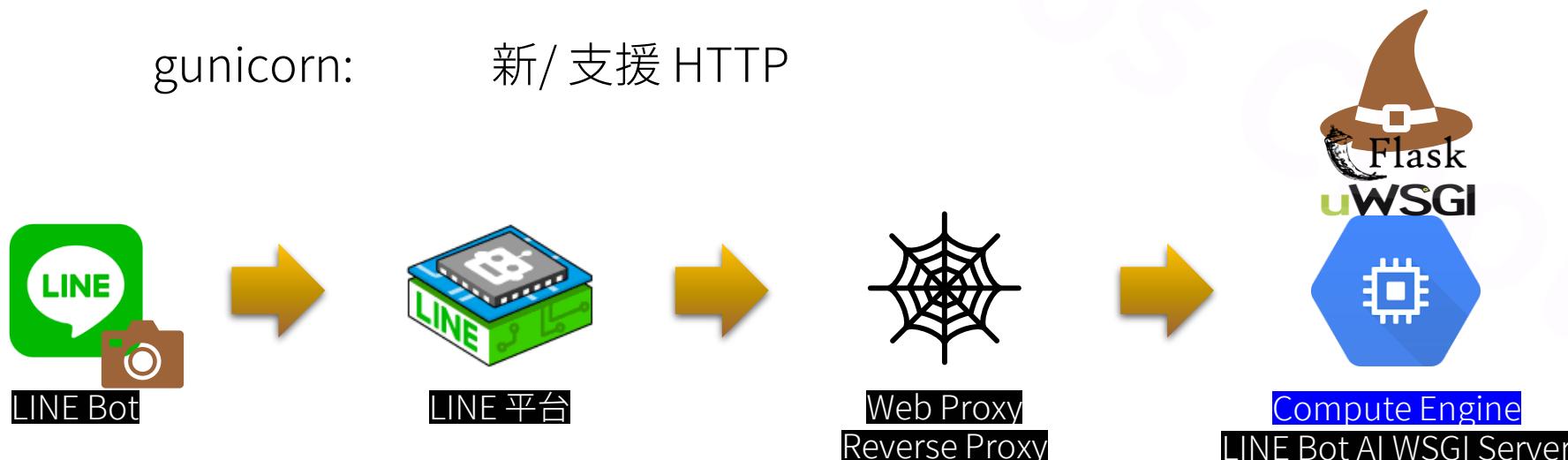
### uWSGI vs gunicorn



uWSGI: 經典/ 支援 HTTP/ 搭配 NGINX 支援 uwsgi protocol

gunicorn:

新/ 支援 HTTP



# LINE Bot & WSGI

## 2. 部署服務

### b. 更新系統同時安裝 LINE Bot 與 WSGI Server (< 4 分)

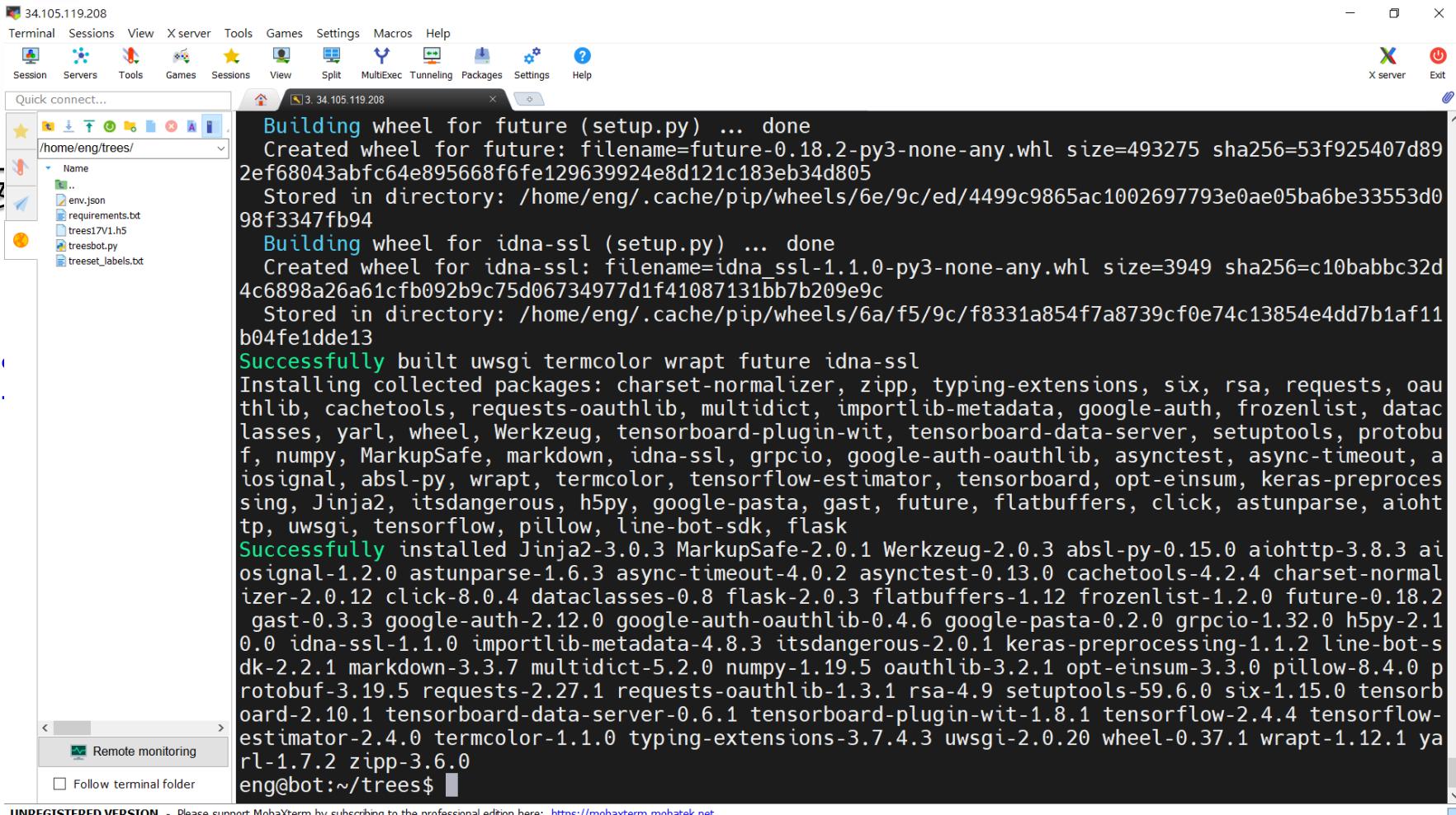
```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade  
pip; sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r  
requirements.txt --no-warn-script-location; source ../../profile
```

# LINE Bot & WSGI

## 2. 部署服務

### b. 更新系統同時安裝

```
sudo apt update  
pip; sudo timed  
requirements.tx
```



The screenshot shows a terminal window titled '34.105.119.208' running on MobaXterm. The terminal displays the output of a pip command:

```
Building wheel for future (setup.py) ... done  
Created wheel for future: filename=future-0.18.2-py3-none-any.whl size=493275 sha256=53f925407d89  
2ef68043abfc64e895668f6fe129639924e8d121c183eb34d805  
Stored in directory: /home/eng/.cache/pip/wheels/6e/9c/ed/4499c9865ac1002697793e0ae05ba6be33553d0  
98f3347fb94  
Building wheel for idna-ssl (setup.py) ... done  
Created wheel for idna-ssl: filename=idna_ssl-1.1.0-py3-none-any.whl size=3949 sha256=c10babbc32d  
4c6898a26a61cfb092b9c75d06734977d1f41087131bb7b209e9c  
Stored in directory: /home/eng/.cache/pip/wheels/6a/f5/9c/f8331a854f7a8739cf0e74c13854e4dd7b1af11  
b04fe1dde13  
Successfully built uwsgi termcolor wrapt future idna-ssl  
Installing collected packages: charset-normalizer, zipp, typing-extensions, six, rsa, requests, oau  
thlib, cachetools, requests-oauthlib, multidict, importlib-metadata, google-auth, frozenlist, data  
classes, yarl, wheel, Werkzeug, tensorboard-plugin-wit, tensorboard-data-server, setuptools, protobu  
f, numpy, MarkupSafe, markdown, idna-ssl, grpcio, google-auth-oauthlib, asynctest, async-timeout, a  
iosignal, absl-py, wrapt, termcolor, tensorflow-estimator, tensorboard, opt-einsum, keras-preproces  
sing, Jinja2, itsdangerous, h5py, google-pasta, gast, future, flatbuffers, click, astunparse, aioht  
tp, uwsgi, tensorflow, pillow, line-bot-sdk, flask  
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 Werkzeug-2.0.3 absl-py-0.15.0 aiohttp-3.8.3 ai  
osignal-1.2.0 astunparse-1.6.3 async-timeout-4.0.2 asynctest-0.13.0 cachetools-4.2.4 charset-normal  
izer-2.0.12 click-8.0.4 dataclasses-0.8 flask-2.0.3 flatbuffers-1.12 frozenlist-1.2.0 future-0.18.2  
gast-0.3.3 google-auth-2.12.0 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.32.0 h5py-2.1  
0.0 idna-ssl-1.1.0 importlib-metadata-4.8.3 itsdangerous-2.0.1 keras-preprocessing-1.1.2 line-bot-s  
dk-2.2.1 markdown-3.3.7 multidict-5.2.0 numpy-1.19.5 oauthlib-3.2.1 opt-einsum-3.3.0 pillow-8.4.0 p  
rotobuf-3.19.5 requests-2.27.1 requests-oauthlib-1.3.1 rsa-4.9 setuptools-59.6.0 six-1.15.0 tensorb  
oard-2.10.1 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.1 tensorflow-2.4.4 tensorflow-  
estimator-2.4.0 termcolor-1.1.0 typing-extensions-3.7.4.3 uwsgi-2.0.20 wheel-0.37.1 wrapt-1.12.1 ya  
rl-1.7.2 zipp-3.6.0  
eng@bot:~/trees$
```

At the bottom of the terminal window, there is a watermark: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

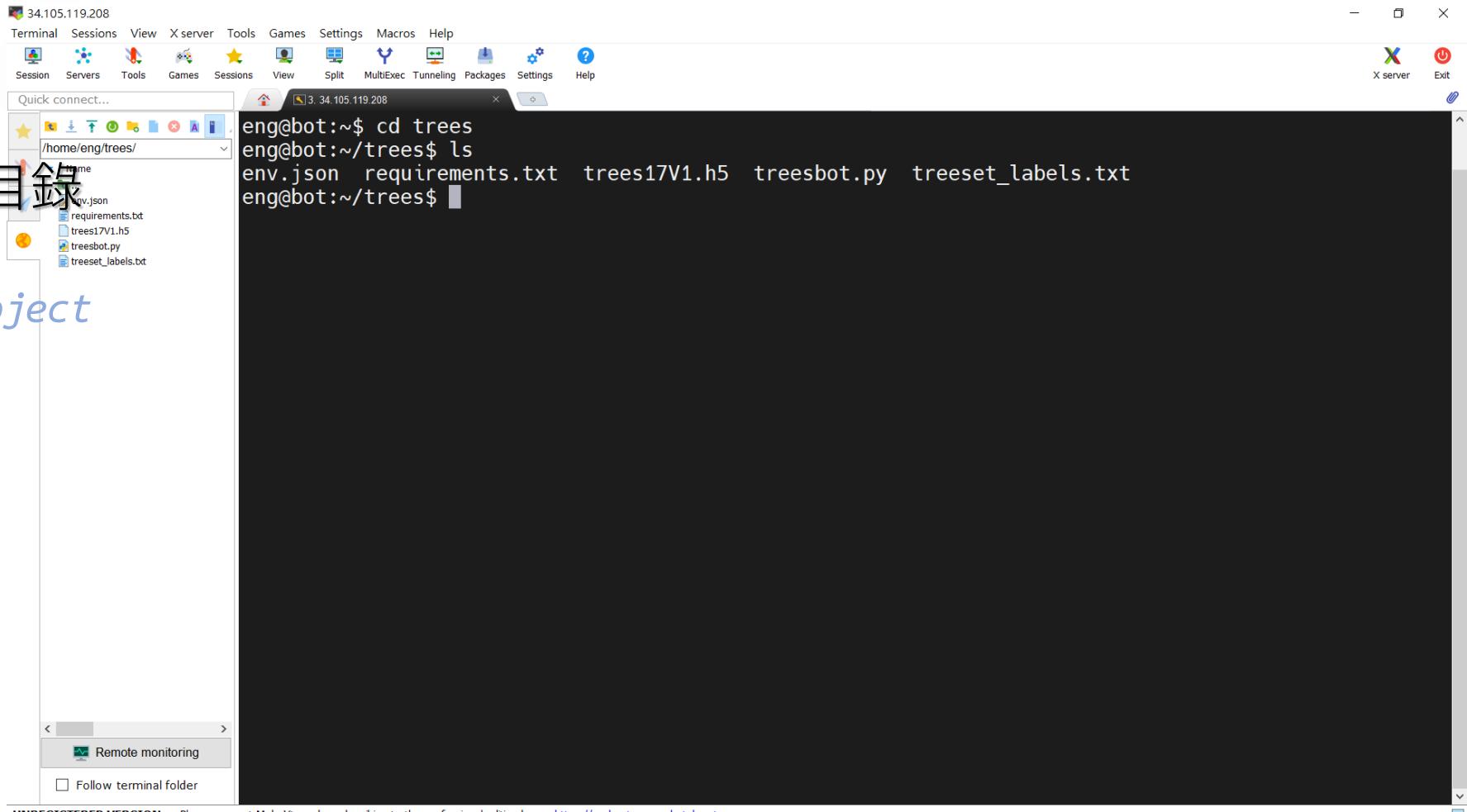
# LINE Bot & WSGI

## 3. 啟動服務

a. 進入 VM 專案目錄

*cd; cd your\_project*

*cd; cd trees*



```
eng@bot:~$ cd trees
eng@bot:~/trees$ ls
env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt
eng@bot:~/trees$
```

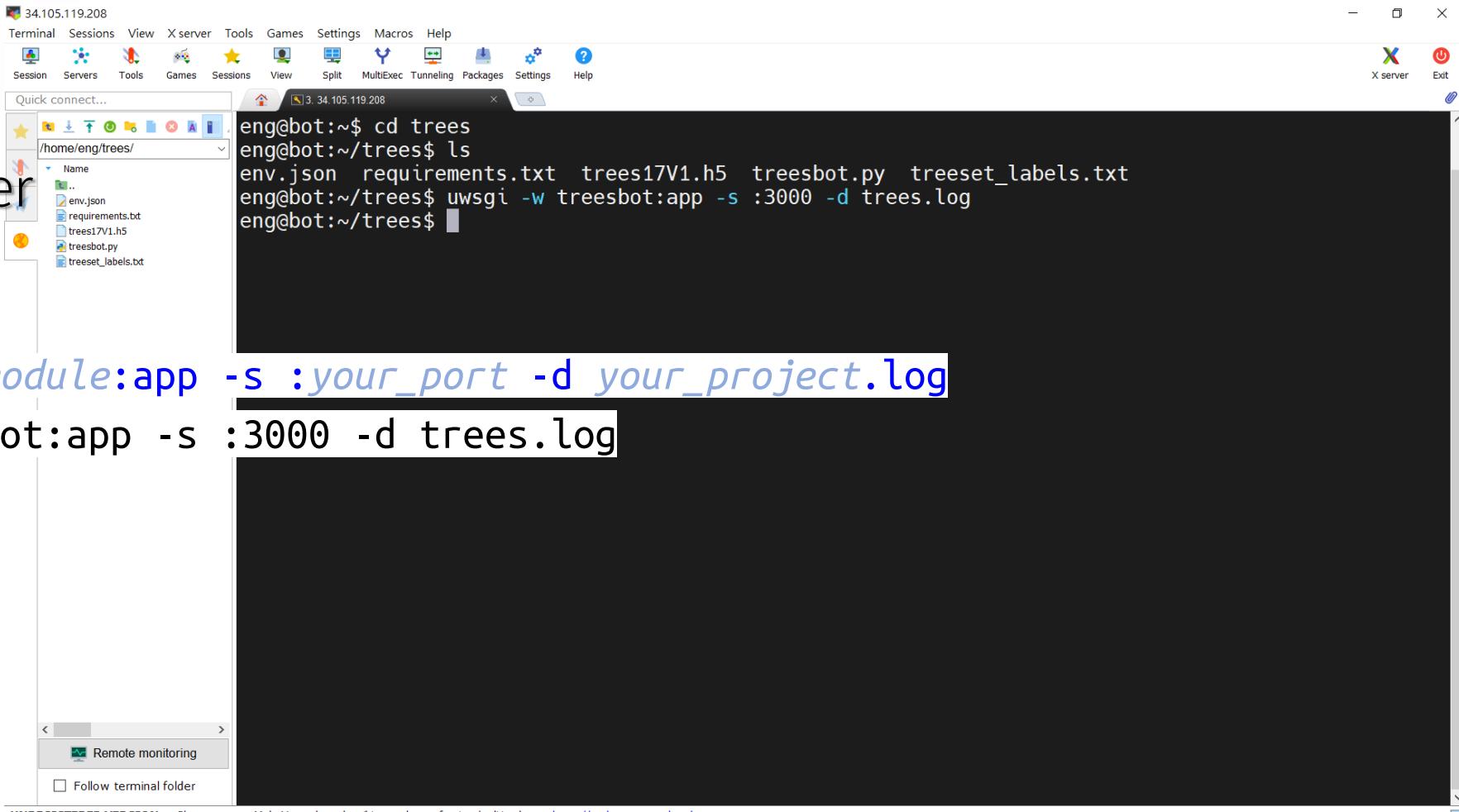
The screenshot shows a MobaXterm interface. The title bar indicates the connection is to 34.105.119.208. The menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. A toolbar below the menu bar contains icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. On the left, a file explorer window titled 'Quick connect...' shows a directory structure under '/home/eng/trees/'. The files listed are env.json, requirements.txt, trees17V1.h5, treesbot.py, and treeset\_labels.txt. At the bottom of the terminal window, there is a status bar with the text 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net'.

# LINE Bot & WSGI

## 3. 啟動服務

- b. 以 WSGI Server  
帶起服務模組

```
uwsgi -w your_module:app -s :your_port -d your_project.log  
uwsgi -w treesbot:app -s :3000 -d trees.log
```



```
34.105.119.208
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3. 34.105.119.208
eng@bot:~$ cd trees
eng@bot:~/trees$ ls
env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt
eng@bot:~/trees$ uwsgi -w treesbot:app -s :3000 -d trees.log
eng@bot:~/trees$ █

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```

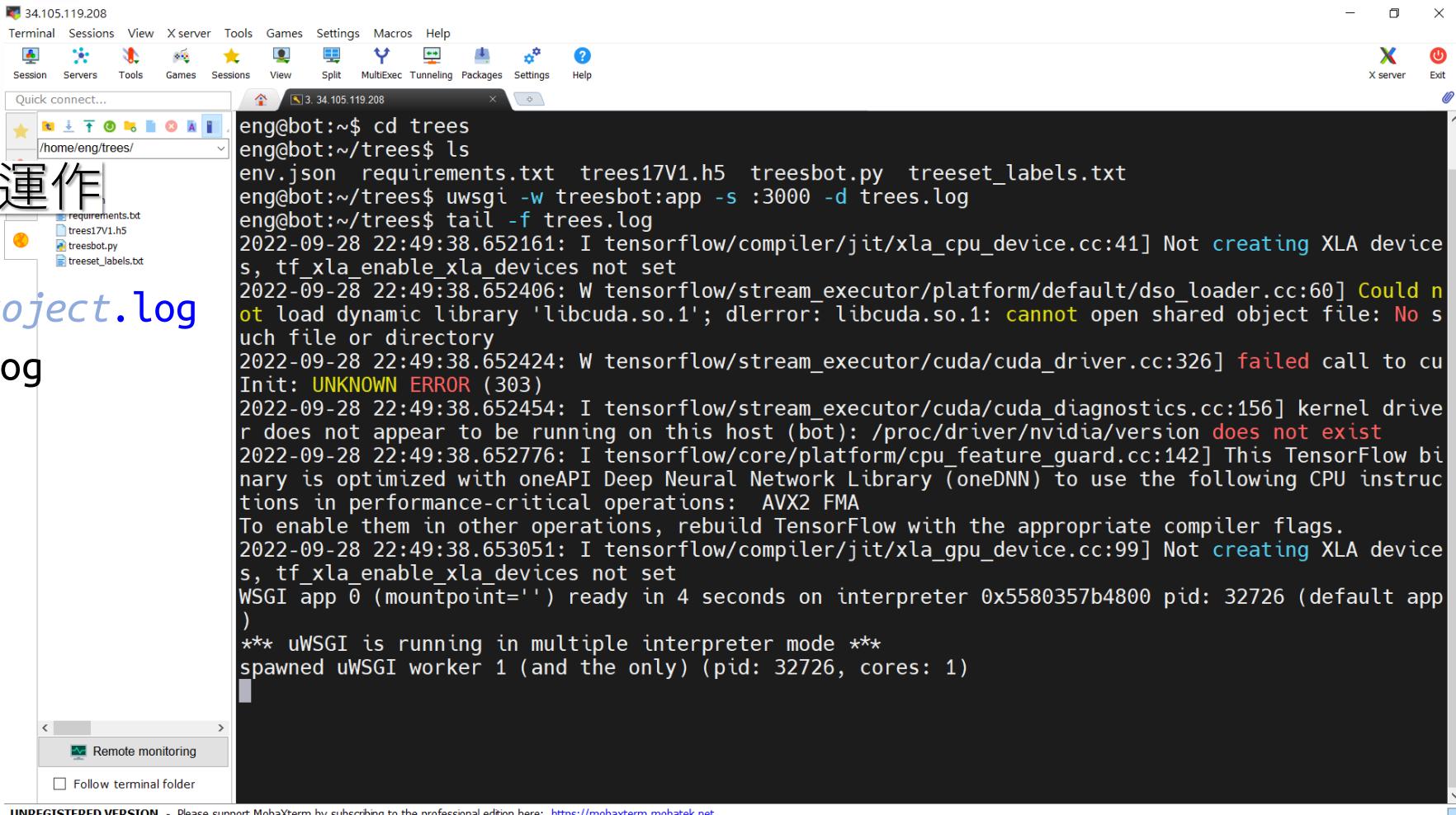
# LINE Bot & WSGI

## 3. 啟動服務

### c. 確認服務正常運作

```
tail -f your_project.log  
tail -f trees.log
```

正常範例



```
eng@bot:~$ cd trees  
eng@bot:~/trees$ ls  
env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt  
eng@bot:~/trees$ uwsgi -w treesbot:app -s :3000 -d trees.log  
eng@bot:~/trees$ tail -f trees.log  
2022-09-28 22:49:38.652161: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices not set  
2022-09-28 22:49:38.652406: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory  
2022-09-28 22:49:38.652424: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)  
2022-09-28 22:49:38.652454: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (bot): /proc/driver/nvidia/version does not exist  
2022-09-28 22:49:38.652776: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA  
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.  
2022-09-28 22:49:38.653051: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set  
WSGI app 0 (mountpoint='') ready in 4 seconds on interpreter 0x5580357b4800 pid: 32726 (default app)  
*** uWSGI is running in multiple interpreter mode ***  
spawned uWSGI worker 1 (and the only) (pid: 32726, cores: 1)
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

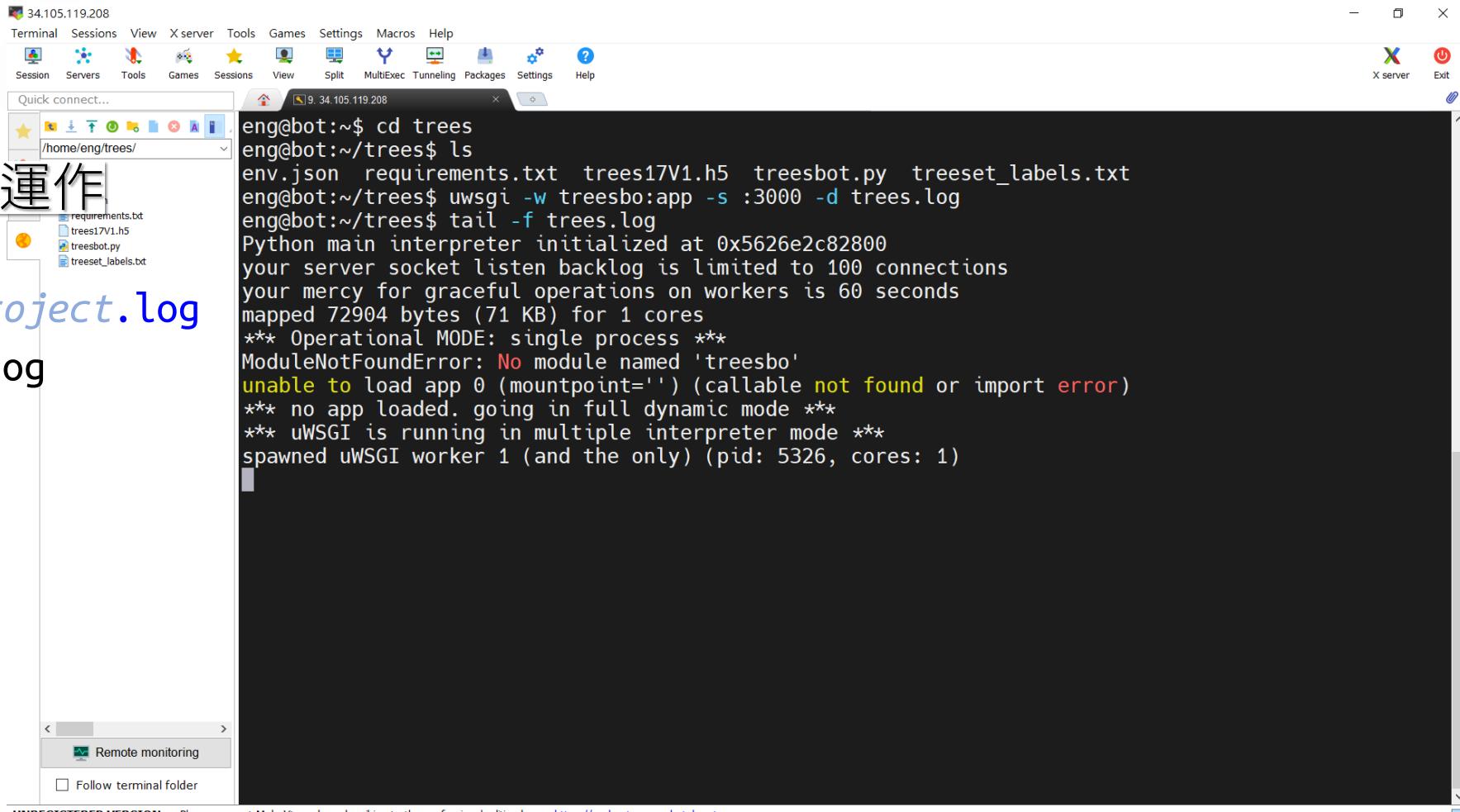
# LINE Bot & WSGI

## 3. 啟動服務

### c. 確認服務正常運作

`tail -f your_project.log`  
`tail -f trees.log`

異常範例



The screenshot shows a MobaXterm window titled '34.105.119.208' with the IP address '9. 34.105.119.208'. The terminal window displays the following command and its output:

```
eng@bot:~$ cd trees
eng@bot:~/trees$ ls
env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt
eng@bot:~/trees$ uwsgi -w treesbo:app -s :3000 -d trees.log
eng@bot:~/trees$ tail -f trees.log
Python main interpreter initialized at 0x5626e2c82800
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 72904 bytes (71 KB) for 1 cores
*** Operational MODE: single process ***
ModuleNotFoundError: No module named 'treesbo'
unable to load app 0 (mountpoint='') (callable not found or import error)
*** no app loaded. going in full dynamic mode ***
*** uWSGI is running in multiple interpreter mode ***
spawned uWSGI worker 1 (and the only) (pid: 5326, cores: 1)
```

The terminal also shows a sidebar with session management icons and a 'Remote monitoring' section.

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# LINE Bot & WSGI

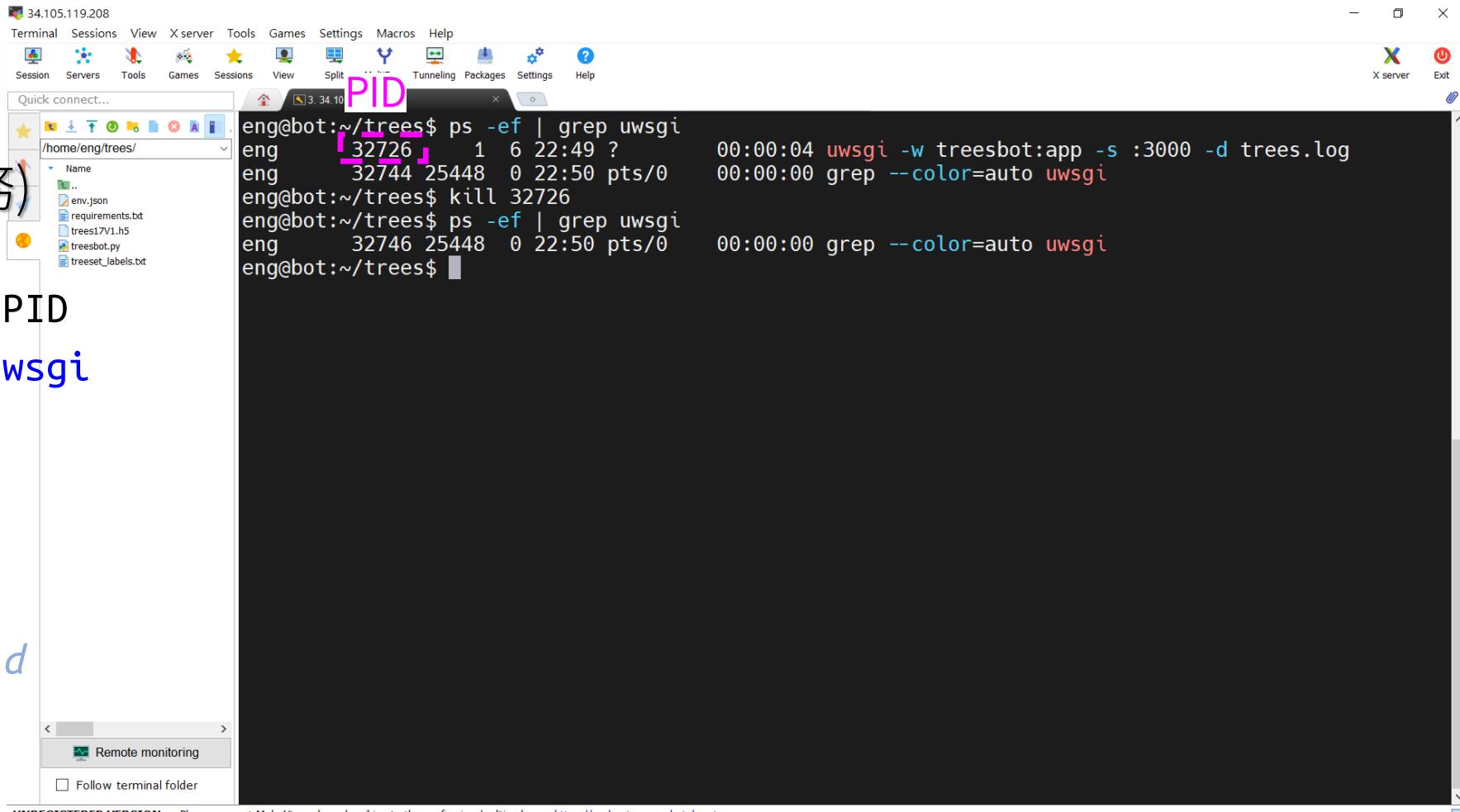
## 3. 啟動服務

d. (日常上下服務)

```
# 取得背景程序 PID  
ps -ef | grep uwsgi
```

```
# 移除背景程序  
kill your_pid
```

```
kill -9 your_pid
```



```
eng@bot:~/trees$ ps -ef | grep uwsgi  
eng 32726 1 6 22:49 ? 00:00:04 uwsgi -w treesbot:app -s :3000 -d trees.log  
eng 32744 25448 0 22:50 pts/0 00:00:00 grep --color=auto uwsgi  
eng@bot:~/trees$ kill 32726  
eng@bot:~/trees$ ps -ef | grep uwsgi  
eng 32746 25448 0 22:50 pts/0 00:00:00 grep --color=auto uwsgi  
eng@bot:~/trees$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### a. 製作服務

① 於 `/lib/systemd/system` 建立 `your_service.service`

```
[Unit]
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=your_script.sh
```

```
RemainAfterExit=yes
```

```
[Install]
```

```
WantedBy=multi-user.target
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### a. 製作服務

② 將 *your\_service.service* 建立 symbolic link 至 */etc/systemd/system*

```
sudo ln -s /lib/systemd/system/your_service.service  
/etc/systemd/system/your_service.service
```

```
sudo ln -s /lib/systemd/system/treesbot.service  
/etc/systemd/system/treesbot.service
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### b. 製作啟動程式

- ① 於專案目錄建立啟動程式 *your\_script.sh*

```
#!/bin/bash
export PATH=/home/your_account/.local/bin:$PATH
cd /home/your_account/your_project

uwsgi -w your_module:app -s :your_port --uid your_account -d your_service.log
--logfile-chown

exit 0
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### b. 製作啟動程式

- ② 調整啟動程式權限為可執行

```
chmod 755 your_script.sh
```

```
chmod 755 treesbot.service.sh
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### c. 設定自動啟動服務

#### ① 建立自動啟動服務

```
sudo systemctl enable your_service
sudo systemctl enable treesbot
```

#### ② 啟動服務

```
sudo systemctl start your_service
sudo systemctl start treesbot
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### c. 設定自動啟動服務

③ 查詢服務狀態

```
sudo systemctl status your_service
```

```
sudo systemctl status treesbot
```

④ 列出所有服務

```
systemctl
```

⑤ 重新載入服務設定

```
sudo systemctl daemon-reload
```

# LINE Bot & WSGI

## 4. (開機自動啟動服務)

### d. 取消自動啟動服務

① 停止服務

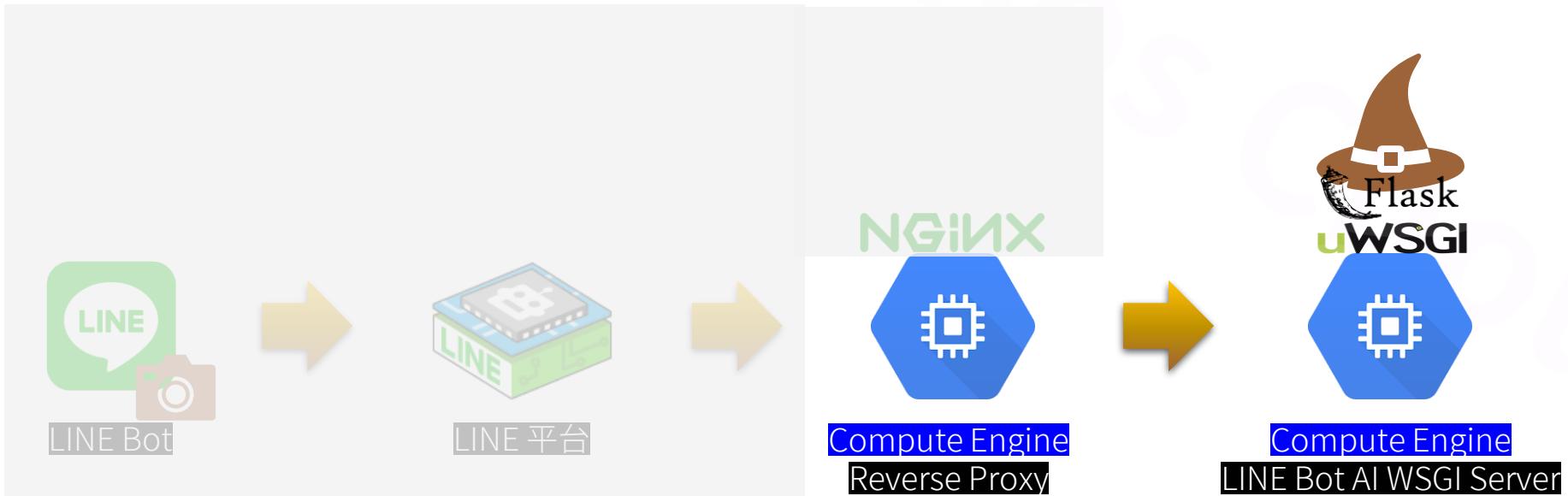
```
sudo systemctl stop your_service
```

```
sudo systemctl stop treesbot
```

② 移除自動啟動服務

```
sudo systemctl disable your_service
```

```
sudo systemctl disable treesbot
```



# 申購並設定網域

## 何謂網域？

- a. 網域為可讀且符合格式的字串，對應特定 IP Address
- b. 主網擁用有者可設定多組子網域

## 中央氣象局

	IP Address	俗稱
cwb.gov.tw		主網域
www.cwb.gov.tw	210.61.218.11	子網域
opendata.cwb.gov.tw	61.56.14.16	子網域
opendata.cwb.gov.tw/dataset		子目錄

# 申購並設定網域

GoDaddy

<https://tw.godaddy.com/>

# 申購並設定網域

GoDaddy

a. 申購網域

The screenshot shows the GoDaddy homepage. At the top, there is a navigation bar with the GoDaddy logo, language selection (台灣), and links for 域名查詢 (Domain Search), 網站及電子商務 (Websites & E-commerce), 網站安全性 (Website Security), and 行銷工具 (Marketing Tools). On the right side of the header are links for 聯絡我們 (Contact Us), 說明 (Description), 登入 (Log In), and a shopping cart icon. The main content area features a large orange banner with the text "運用 .com 帶領企業進軍網路世界，只要 NT\$291/第 1 年" (Use .com to lead your company into the network world, only NT\$291 per year). Below the banner is a search bar with the input "enadv" highlighted by a pink rectangle, and a black "搜尋" (Search) button to its right, also highlighted by a pink rectangle.

更多助您擴展企業規模的工具。

大家都知道我們是網域行家，不過您也可以看看其他讓企業在網路上更能大展身手的重要功能。



# 申購並設定網域

GoDaddy

## a. 申購網域

The screenshot shows the GoDaddy Taiwan website interface. At the top, there's a navigation bar with links for '聯絡我們' (Contact Us), '說明' (Explanation), '登入' (Log In), and a shopping cart icon. Below the navigation is a search bar containing 'enadv'. To the right of the search bar is a magnifying glass icon and a green button labeled '前往購物車' (Go to Cart). The main content area displays a list of domain names and their prices:

Domain Name	Price (NT\$)	Period	Action
enadv.com	NT\$3,653 <sup>②</sup>	第一年	加入購物車
enadv.tw	NT\$734 <sup>②</sup>	第一年	加入購物車
enadv.net	NT\$498 <sup>②</sup>	第一年	加入購物車
enadv.com.tw	NT\$1,070 <sup>②</sup>	隔年將收取相同費用	加入購物車
enadv.org	NT\$319 <sup>②</sup>	第一年	加入購物車
enadv.cc	NT\$175 <sup>②</sup>	第一年	加入購物車

A note above the first row states: '◎ enadv.com 已被他人使用 我們依然有機會能為您取得。看看'.

# 申購並設定網域

GoDaddy

## a. 申購網域

The screenshot shows the GoDaddy checkout process for a .CLUB domain. At the top, it displays the domain name "enadv.club" and the registration type ".CLUB 網域註冊". A dropdown menu shows "1年" (1 year) selected. To the right, the price is listed as "NT\$33" with a note "現省 94%" (Save 94%). Below this, a yellow box contains the text: "每年 170,000 次，這就是罪犯嘗試竊取網域的頻率。保護您的網域。" (Every year, there are 170,000 times when criminals try to steal domains. Protect your domain.) and the price "NT\$199/年, 每個網域" (NT\$199 per year, per domain). A "新增" (Add) button is next to the price. At the bottom left is a link "查看優惠免責聲明" (View promotional disclaimer), and at the bottom right is a "清空購物車" (Empty shopping cart) button.

**小計 (TWD)** **NT\$33**

促銷代碼: [GOKBTW06](#)

讚! 您的訂單成功省下 NT\$474.

**繼續結帳**

包您滿意

如果您對選購內容有任何不滿之處，歡迎洽詢 GoDaddy  
客戶顧問為您解決問題。

### cPanel® 主機服務

- 領先業界的網頁載入時間
- 運作時間保證達到 99.9%
- 屢獲殊榮的客戶支援

NT\$234/月

[新增](#)

### 新增網域

搜尋您的理想網域



### 相符網域可供使用

enadv.net	NT\$592	NT\$340	<a href="#">新增</a>
enadv.co	NT\$1,104	NT\$148	<a href="#">新增</a>
enadv.info	NT\$834	NT\$78	<a href="#">新增</a>

# 申購並設定網域

GoDaddy

a. 申購網域

The screenshot shows the GoDaddy Taiwan website's payment page. At the top, the GoDaddy logo and '台灣' are visible. On the right, there is a link to '聯絡我們'. The main title '購買' is centered above the breadcrumb navigation: 購物籃 > 登入 > 帳單資訊 > 付款 > 完成. The payment method is set to VISA. Below it, the billing information section is labeled '帳單資訊'.

On the right side, the product details are listed:

- enadv.club .CLUB 網域註冊**  
NT\$35  
現省 94%  
於 2022年6月續約，費用為 NT\$537 (1 年效期)
- 小計** NT\$28
- 稅金與費用** NT\$7
- 促銷代碼:** GOKBTW06

The total amount is displayed as **總計 (TWD) NT\$35**. A note at the bottom left says '讚！您的訂單成功省下 NT\$474。' A large red rectangular button at the bottom right is labeled '完成購買'.

按下「完成購買」之後，即表示您同意我們的條款與條件及隱私政策之內容，並同意將您的產品加入我們的自動續約服務，您隨時都可以到帳戶中的「續約和帳單」頁面取消此服務。直到取消為止，系統都會透過您為此訂單選擇的付款方式或備份付款方式自動收取續約的費用。您的付款正在 美國 進行處理。

# 申購並設定網域

GoDaddy

## b. 設定網域

The screenshot shows the GoDaddy website interface. At the top, there is a search bar with placeholder text '找出完美網域' and a teal button labeled '搜尋網域'. Below the search bar, there is a promotional section for 'Virtual Hosting' featuring a photo of a man holding a book titled 'Christopher Lee Wheat and Wood'. The sidebar on the right contains a user's account information, including a PIN number (NT\$319) and a menu with options like '我的產品' (My Products), which is highlighted with a pink rectangle.

.app 網域 NT\$576/第 1 年	Microsoft 365 NT\$52/月	虛擬主機 NT\$62/月	<b>j.info™</b> 特賣中！NT\$130/第 1 年。 讓全世界知道您的知識素養。
WordPress NT\$209/月	網站安全性 NT\$167/月	SSL 一年 NT\$1,739	

# 申購並設定網域

GoDaddy

b. 設定網域

The screenshot shows the GoDaddy dashboard. At the top, there's a navigation bar with the GoDaddy logo, a '我的帳戶' (My Account) link, and various icons for help, cart, notifications, and account settings. Below the navigation is a search bar labeled '搜尋新網域'. A main heading says '開始使用新產品'. In the center, there's a card for a domain named 'enadv.site' with a globe icon, followed by the text '建立網站或電子郵件地址'. Below this, a section titled '網域' (Domains) lists the domain 'enadv.site'. To the right of the domain name is a vertical ellipsis menu with options: '編輯設定' (Edit Settings), '變更隱私' (Change Privacy), and '管理 DNS' (Manage DNS). The '管理 DNS' option is highlighted with a pink rectangle. At the bottom right of the page is a blue button with the text '洽詢我們' (Contact Us).

# 申購並設定網域

GoDaddy

b. 設定網域

The screenshot shows the GoDaddy DNS Management interface for the domain `enadv.site`. The top navigation bar includes links for GoDaddy logo, domains, account, and settings. The main menu has options: 網域 (selected), 買賣, DNS, 設定, and 說明.

The current page is "我的網域 / 網域設定" (My Domains / Domain Settings) under "DNS 管理".

The "DNS 記錄" (DNS Record) section displays two entries:

類型	名稱	資料	TTL	刪除	編輯
A	@	Parked	600 秒		
A			600 秒		

At the top right of the "DNS 記錄" section, there are buttons for "新增" (Add New) and "...".

# 申購並設定網域

GoDaddy

b. 設定網域

The screenshot shows the GoDaddy DNS Management interface for the domain enadv.site. The top navigation bar includes links for GoDaddy logo, account, and menu options: 網域 (Domain), 買賣 (Buy), DNS, 設定 (Settings), and 說明 (Help). The current page is '我的網域 / 網域設定' (My Domain / Domain Settings) under 'DNS 管理' (DNS Management).

In the main content area, a modal window titled 'DNS 記錄' (DNS Record) is open. It contains a brief description: 'DNS 記錄會向網際網路說明如何處理您的網域，如顯示網站內容及發送 email 等。' (DNS records tell the Internet how to handle your domain, such as displaying website content and sending emails.)

The modal has several buttons at the top: '刪除' (Delete), '複製' (Copy), '篩選' (Filter), '新增' (Add), and three dots for more options.

The table below lists the A Record settings:

類型	名稱	內容值	TTL
A	t	34.105.119.208	預設

Annotations in pink highlight specific fields:

- 'A Record' is highlighted in pink.
- '子網域名稱' (Subdomain Name) is highlighted in pink.
- 'VM 外部 IP' (VM External IP) is highlighted in pink.
- The '新增記錄' (Add Record) button is highlighted in pink.
- The '刪除' (Delete) and '複製' (Copy) buttons are also highlighted in pink.

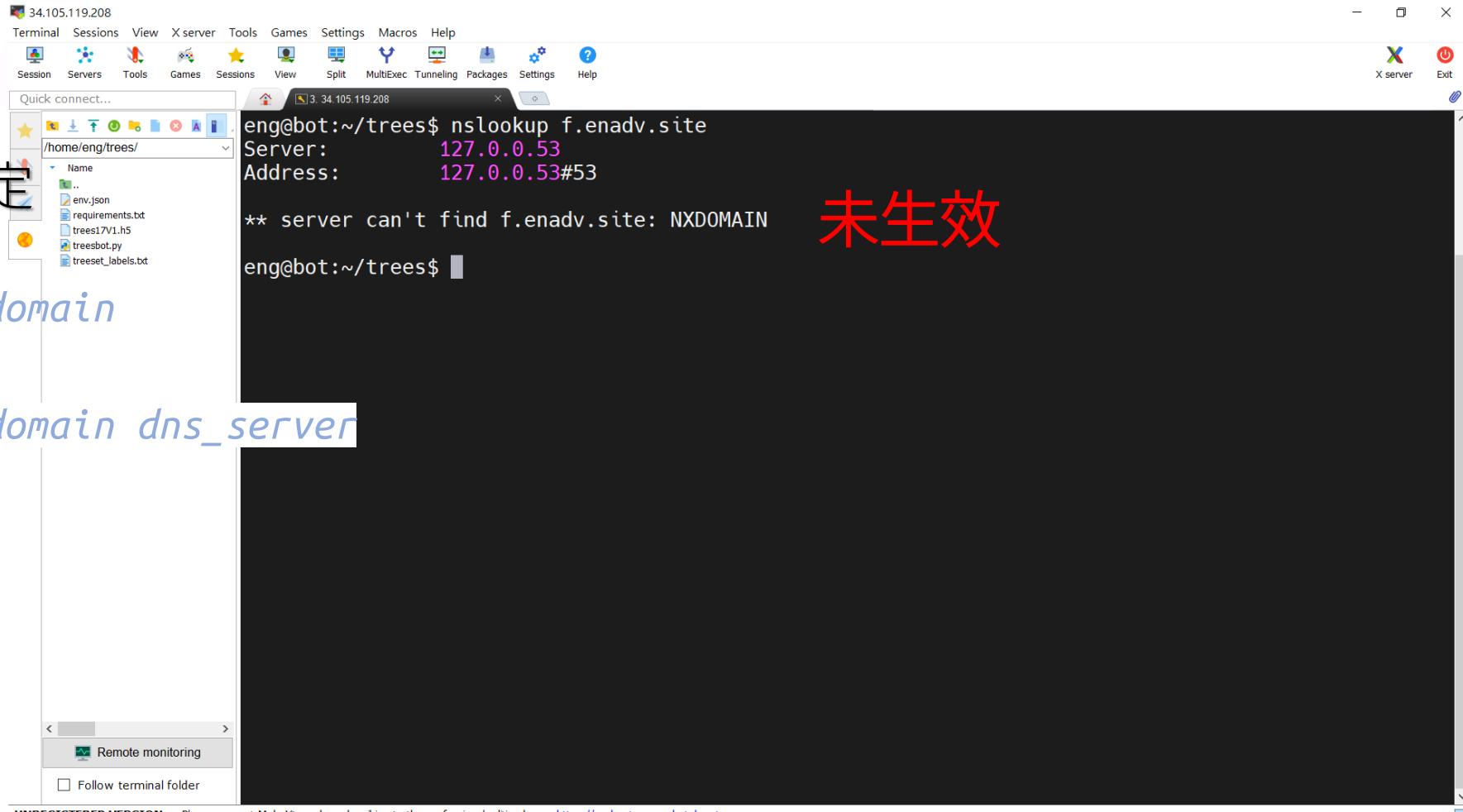
# 申購並設定網域

## 驗證網域

確認網域與 IP 綁定

`nslookup your_domain`

`nslookup your_domain dns_server`



The screenshot shows a terminal window in MobaXterm with the title bar "34.105.119.208". The menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The terminal window displays the command:

```
eng@bot:~/trees$ nslookup f.enadv.site
Server: 127.0.0.53
Address: 127.0.0.53#53

** server can't find f.enadv.site: NXDOMAIN
eng@bot:~/trees$
```

To the right of the terminal window, the text "未生效" (Not Effective) is displayed in red.

At the bottom of the terminal window, there is a status bar with the text "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

# 申購並設定網域

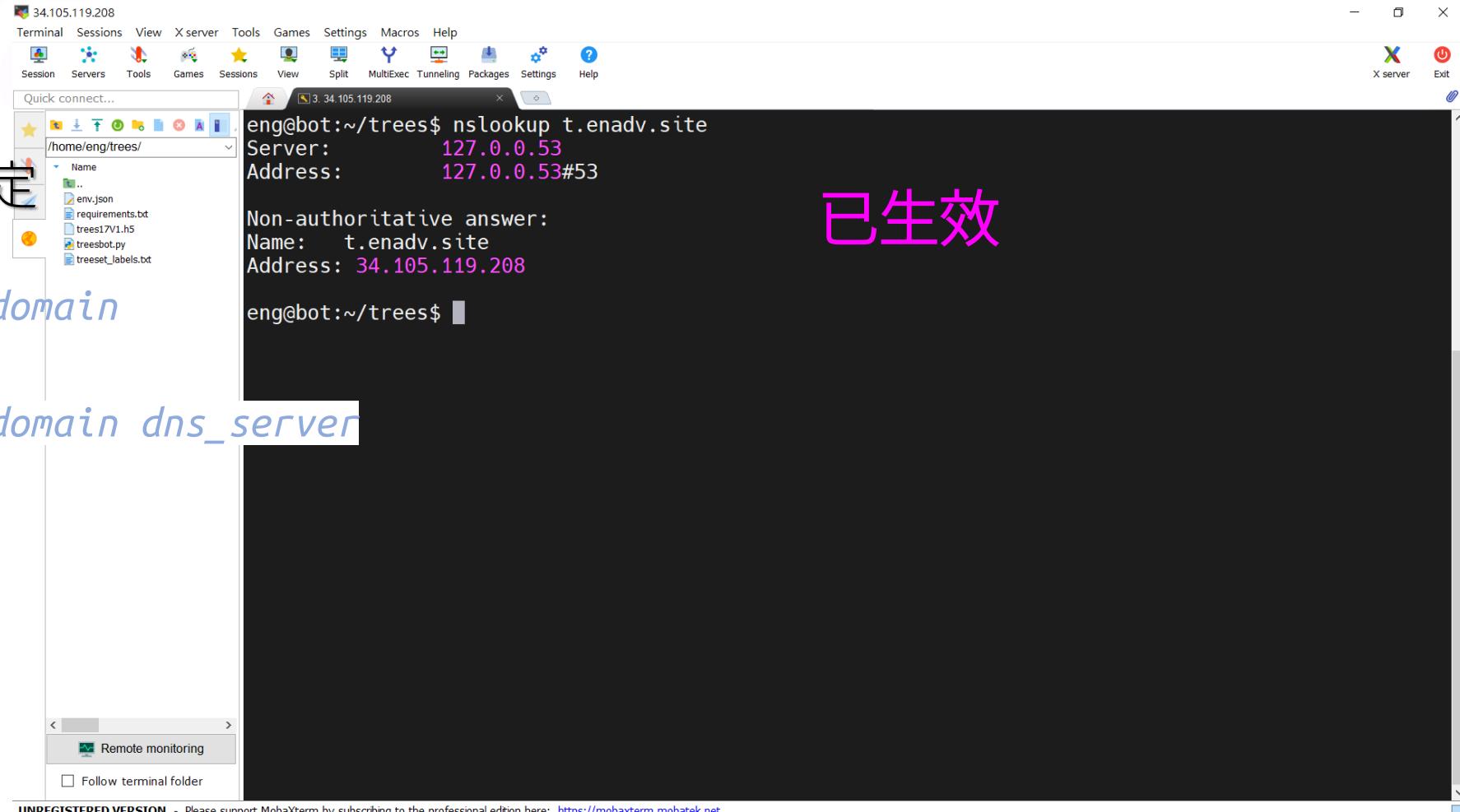
## 驗證網域

確認網域與 IP 綁定

`nslookup your_domain`

`nslookup your_domain dns_server`

已生效



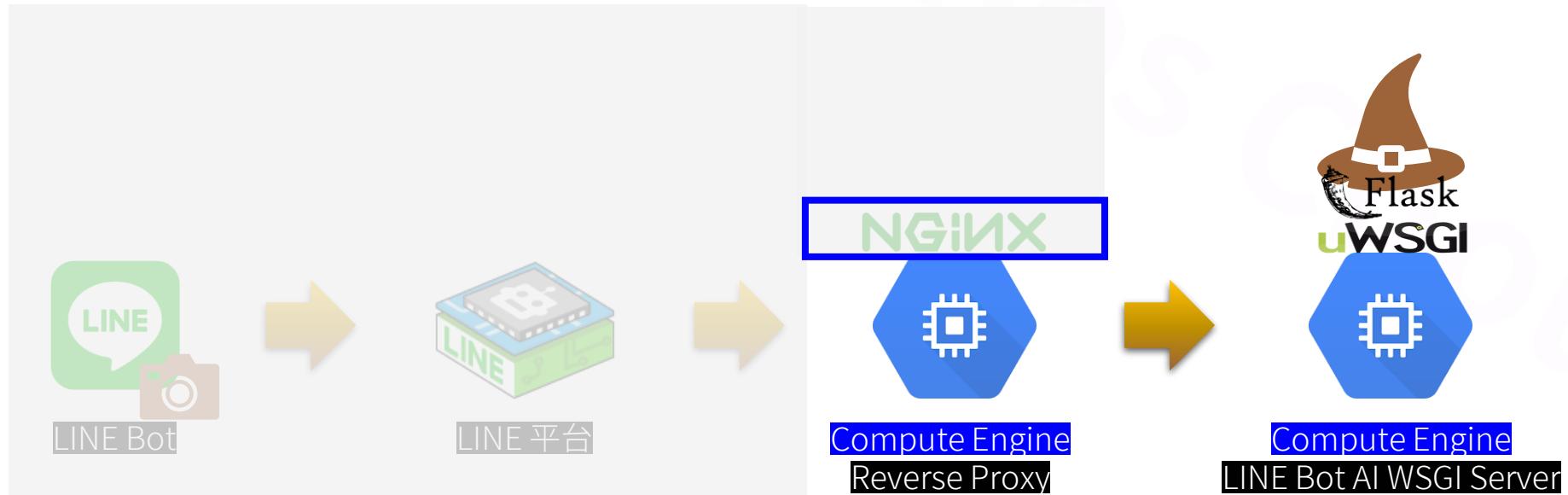
The screenshot shows a terminal window in MobaXterm with the title bar "34.105.119.208". The menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The terminal window displays the following output:

```
eng@bot:~/trees$ nslookup t.enadv.site
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: t.enadv.site
Address: 34.105.119.208

eng@bot:~/trees$
```

The terminal window has a sidebar with a file tree showing files like env.json, requirements.txt, trees17V1.h5, treesbot.py, and treeset\_labels.txt. At the bottom of the terminal window, there is a status bar with the text "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".



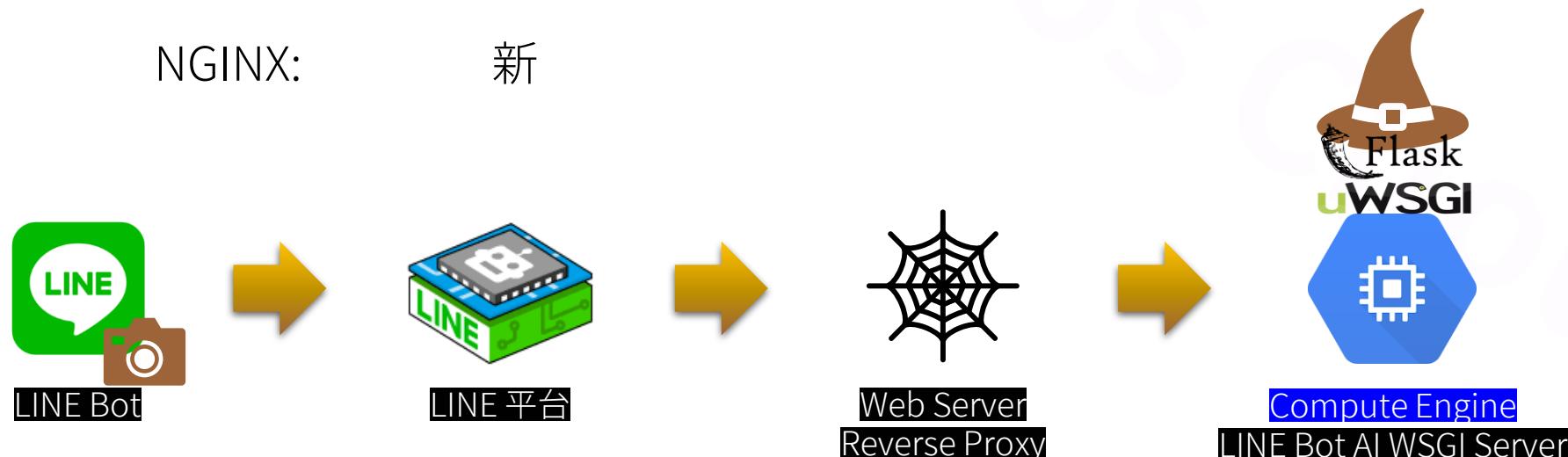
# 網站與憑證

## Web Server 的選擇

Apache vs NGINX

Apache: 經典

NGINX: 新



# 網站與憑證

## Web Server 的選擇

Apache vs NGINX

Apache: 經典



NGINX: 新



LINE Bot



LINE 平台



NGINX

Compute Engine  
Reverse Proxy



Compute Engine  
LINE Bot AI WSGI Server

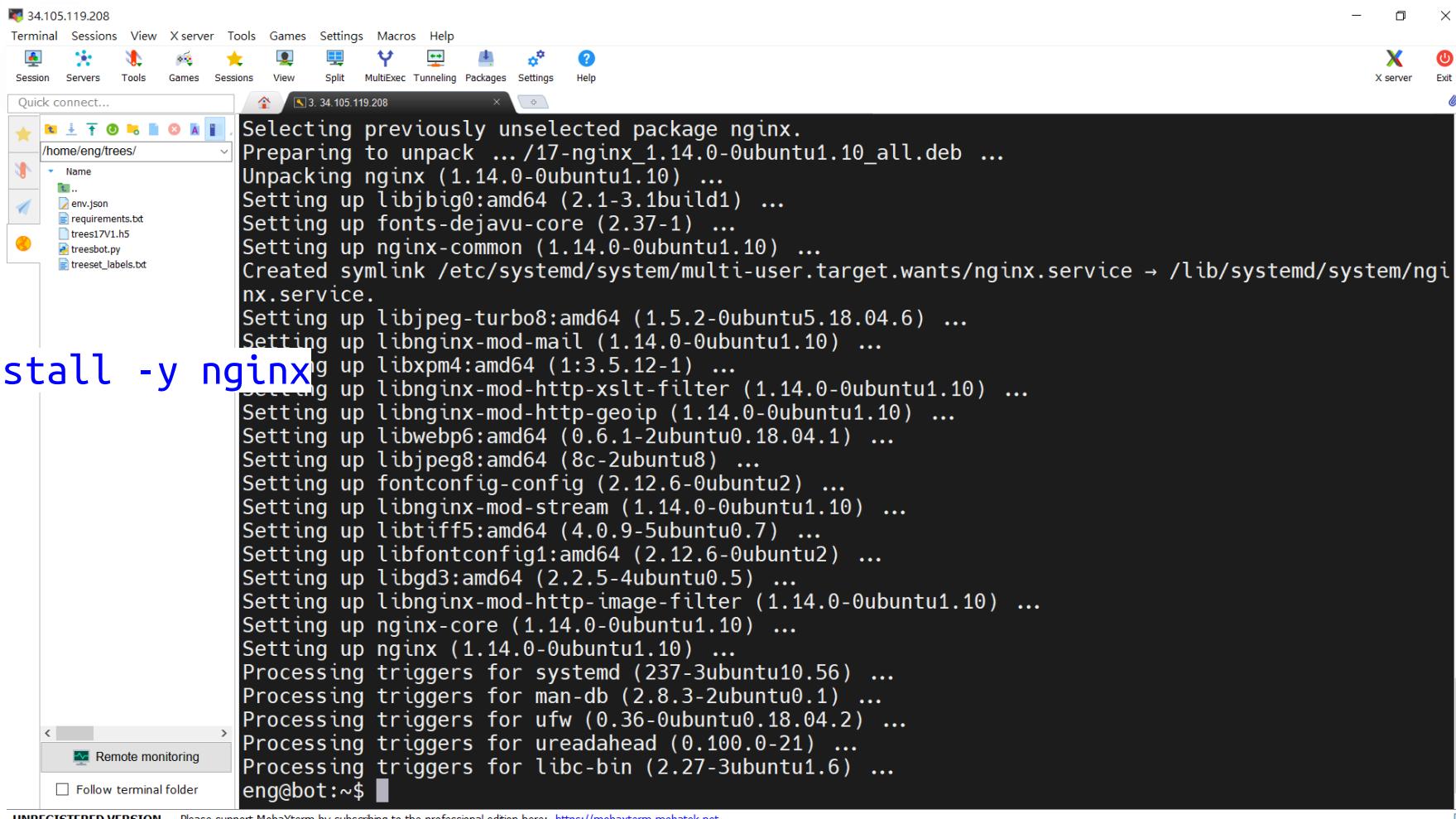
# 網站與憑證

## 1. 架設 NGINX

### a. 安裝 NGINX

# 登入 VM

`sudo apt-get install -y nginx`



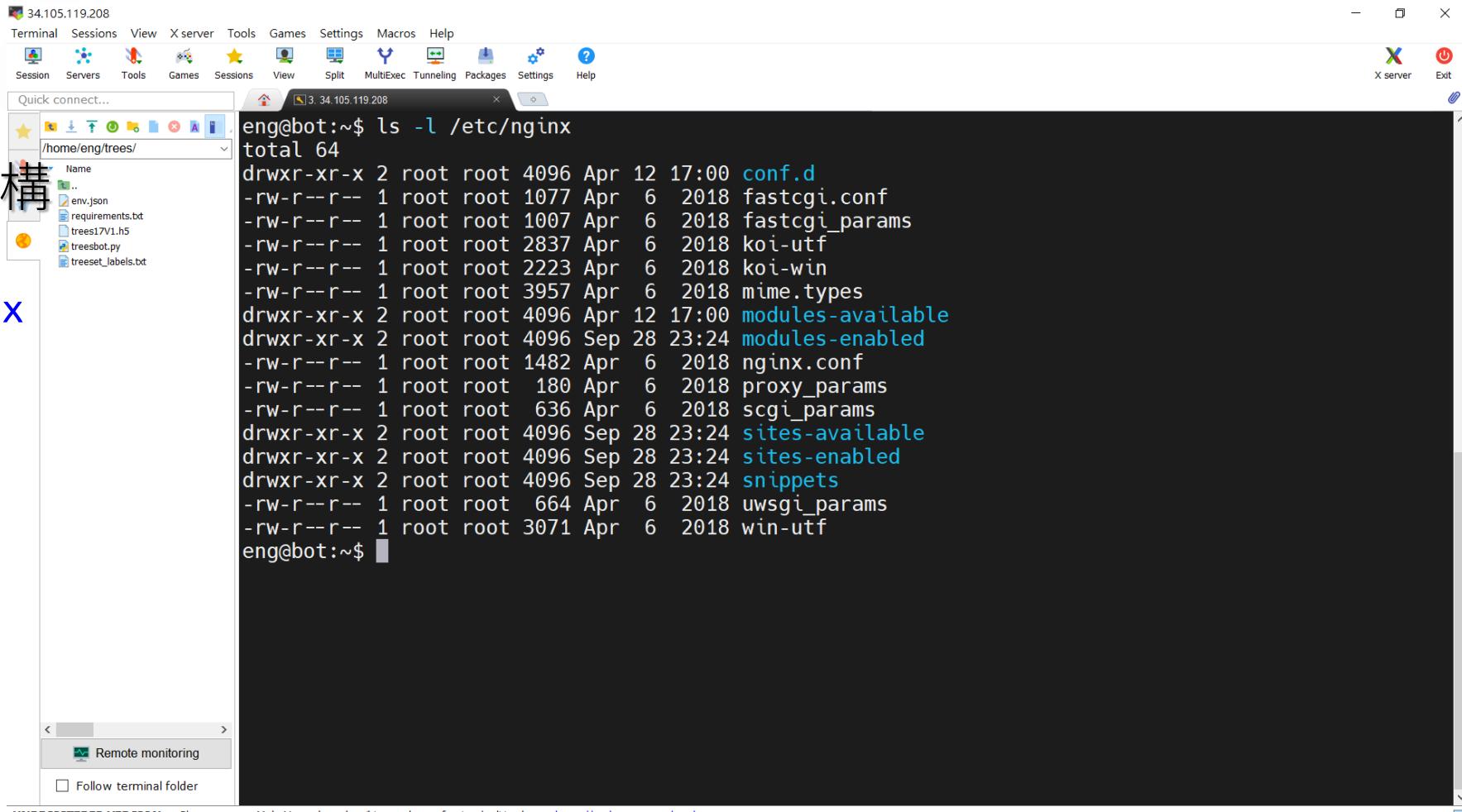
The screenshot shows a terminal window in MobaXterm connected to an Ubuntu 1.10 VM. The terminal displays the output of the command `sudo apt-get install -y nginx`. The output shows the package being unpacked, fonts being set up, symlinks being created, and various libraries and configuration files being installed. The terminal window has a dark background and light-colored text. The top menu bar includes options like Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. A sidebar on the left shows the current directory structure: `/home/eng/trees/`, containing `..`, `env.json`, `requirements.txt`, `trees17V1.h5`, `treebot.py`, and `treeset_labels.txt`. At the bottom of the terminal window, there are buttons for "Remote monitoring" and "Follow terminal folder". The status bar at the bottom of the screen reads "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

# 網站與憑證

## 1. 架設 NGINX

b. 觀察組態檔架構

`ls -l /etc/nginx`



The screenshot shows a terminal window in MobaXterm connected to a host at 34.105.119.208. The terminal displays the output of the command `ls -l /etc/nginx`. The output lists numerous files and directories within the `/etc/nginx` directory, including configuration files like `conf.d`, `fastcgi.conf`, and `mime.types`, as well as modules and snippets. The terminal interface includes a file browser on the left and various MobaXterm settings at the top.

```
eng@bot:~$ ls -l /etc/nginx
total 64
drwxr-xr-x 2 root root 4096 Apr 12 17:00 conf.d
-rw-r--r-- 1 root root 1077 Apr  6 2018 fastcgi.conf
-rw-r--r-- 1 root root 1007 Apr  6 2018 fastcgi_params
-rw-r--r-- 1 root root 2837 Apr  6 2018 koi-utf
-rw-r--r-- 1 root root 2223 Apr  6 2018 koi-win
-rw-r--r-- 1 root root 3957 Apr  6 2018 mime.types
drwxr-xr-x 2 root root 4096 Apr 12 17:00 modules-available
drwxr-xr-x 2 root root 4096 Sep 28 23:24 modules-enabled
-rw-r--r-- 1 root root 1482 Apr  6 2018 nginx.conf
-rw-r--r-- 1 root root 180 Apr  6 2018 proxy_params
-rw-r--r-- 1 root root 636 Apr  6 2018 scgi_params
drwxr-xr-x 2 root root 4096 Sep 28 23:24 sites-available
drwxr-xr-x 2 root root 4096 Sep 28 23:24 sites-enabled
drwxr-xr-x 2 root root 4096 Sep 28 23:24 snippets
-rw-r--r-- 1 root root 664 Apr  6 2018 uwsgi_params
-rw-r--r-- 1 root root 3071 Apr  6 2018 win-utf
eng@bot:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 網站與憑證

## 1. 架設 NGINX

### b. 觀察組態檔架構

```
ls -l /etc/nginx
```

`nginx.conf`

NGINX 啟動時預設讀取的組態檔，內容  
包含掃描 `sites-enabled` 中所有組態檔

`sites-available` 目錄

儲存各種啟動組態檔，但未必會被啟動。  
欲啟動的組態檔應以 symbolic link 方式  
置於 `sites-enabled` 目錄

`sites-enabled` 目錄

實際欲啟動的組態檔

# 網站與憑證

## 1. 架設 NGINX

### c. 建立組態檔

```
sudo nano /etc/nginx/sites-available/your_project.conf  
sudo nano /etc/nginx/sites-available/trees.conf
```

```
server {  
    server_name your_domain;  
    location / {  
        include uwsgi_params;  
        uwsgi_pass your_ip:your_port;  
    }  
}
```

#### Note

調整 *your\_domain*, *your\_ip*, *your\_port*

# 網站與憑證

## 1. 架設 NGINX

### c. 建立組態檔

```
sudo nano /etc/  
sudo nano /etc/
```

```
server {  
    server_name  
    location /  
        include  
        uwsgi_p  
    }  
}
```

The screenshot shows a terminal window titled 'GNU nano 2.9.3' displaying the contents of the file '/etc/nginx/sites-available/trees.conf'. The window is part of the MobaXterm interface, which includes a sidebar for session management and a menu bar at the top.

```
server {  
    server_name t.enadv.site;  
    # for LINE Bot  
    location / {  
        include uwsgi_params;  
        #uwsgi_pass unix:/home/your_account/your_project/your_project.sock;  
        uwsgi_pass 127.0.0.1:3000;  
        #proxy_pass http://your_url;  
    }  
    # for Web or LIFF  
    #location /test {  
        #root /home/your_account/your_project;  
        #alias /home/your_account/your_project;  
    #}  
}
```

The terminal window also displays a set of keyboard shortcuts at the bottom:

- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^C Cur Pos
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Uncut Text
- ^T To Spell
- ^P Go To Line

At the very bottom of the terminal window, there is a footer note: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

# 網站與憑證

## 1. 架設 NGINX

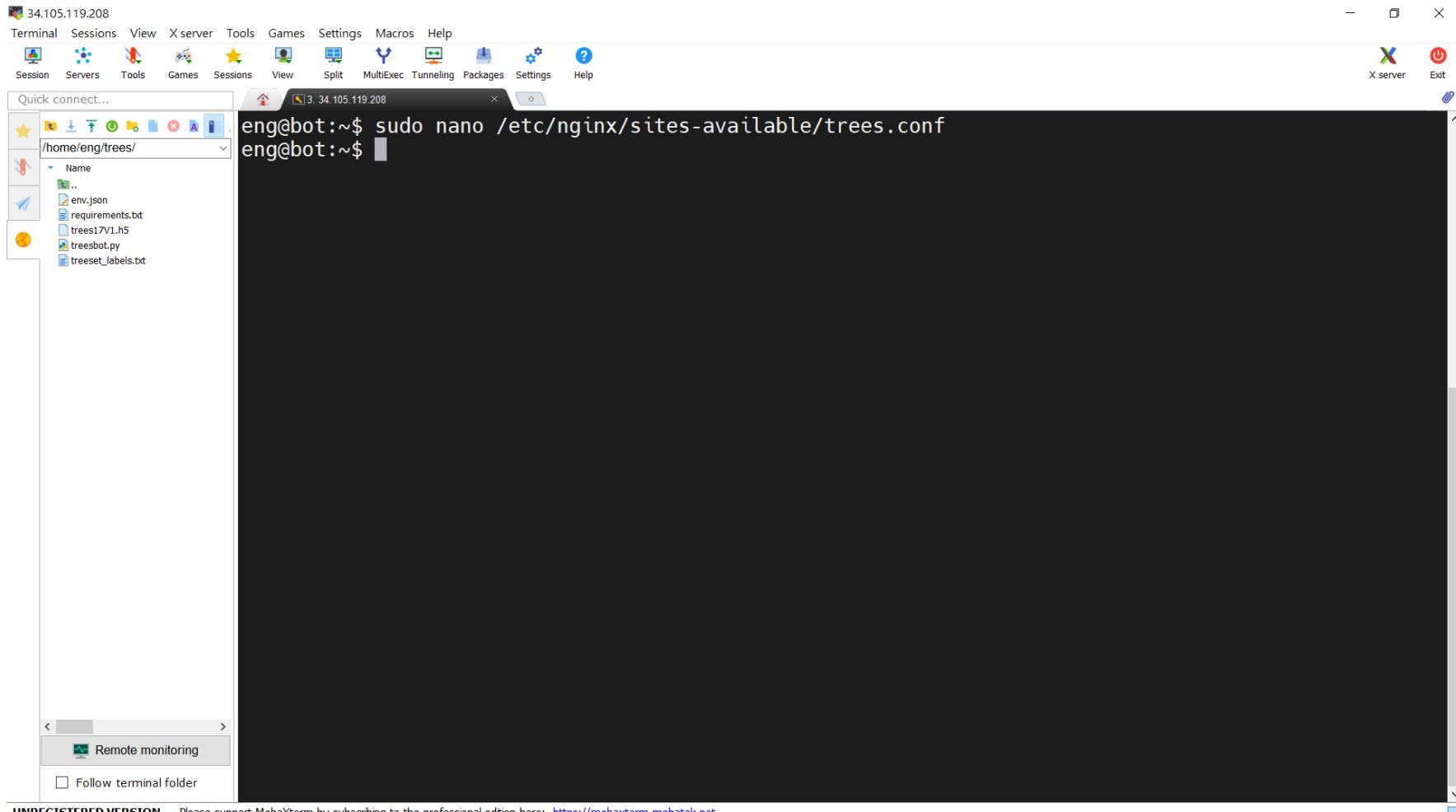
### c. 建立組態檔

# 儲存

Ctrl + O

# 退出

Ctrl + X



The screenshot shows a MobaXterm window titled '34.105.119.208'. The terminal session is running on port 34.105.119.208. The user is in the directory '/home/eng/trees/'. The terminal command shown is:

```
eng@bot:~$ sudo nano /etc/nginx/sites-available/trees.conf
```

The file treebot.py is visible in the file browser on the left.

# 網站與憑證

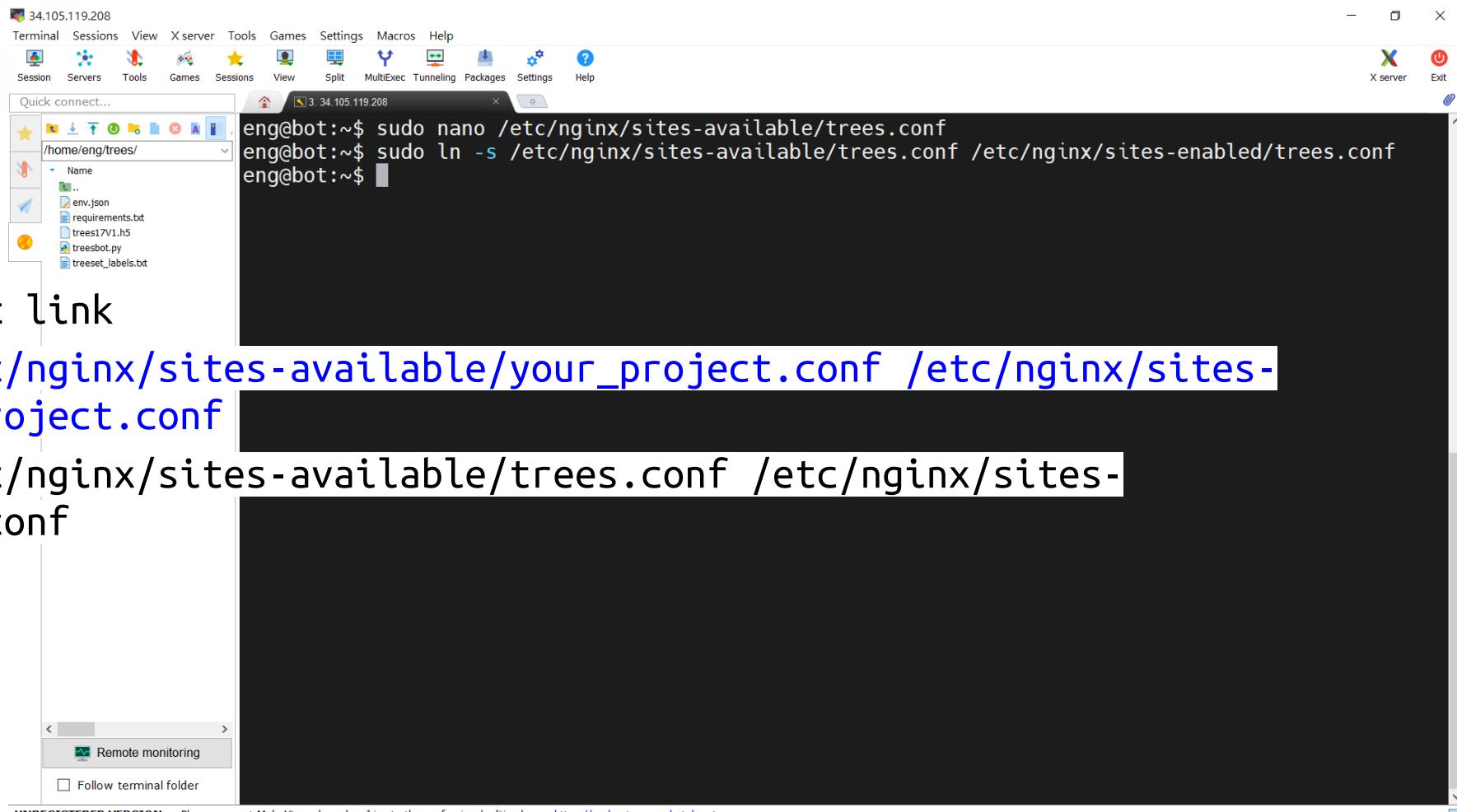
## 1. 架設 NGINX

### c. 建立組態檔

```
# 建立 symbolic link
```

```
sudo ln -s /etc/nginx/sites-available/your_project.conf /etc/nginx/sites-enabled/your_project.conf
```

```
sudo ln -s /etc/nginx/sites-available/trees.conf /etc/nginx/sites-enabled/trees.conf
```



```
eng@bot:~$ sudo nano /etc/nginx/sites-available/trees.conf
eng@bot:~$ sudo ln -s /etc/nginx/sites-available/trees.conf /etc/nginx/sites-enabled/trees.conf
eng@bot:~$
```

# 網站與憑證

## 1. 架設 NGINX

### c. 建立組態檔

```
# 確認 symbolic link
```

```
ls /etc/nginx/sites-enabled/
```

```
# 確認組態檔
```

```
cat /etc/nginx/sites-enabled/your_project.conf
```

```
cat /etc/nginx/sites-enabled/trees.conf
```

```
eng@bot:~$ sudo nano /etc/nginx/sites-available/trees.conf
eng@bot:~$ sudo ln -s /etc/nginx/sites-available/trees.conf /etc/nginx/sites-enabled/trees.conf
eng@bot:~$ ls -l /etc/nginx/sites-enabled/
total 0
lrwxrwxrwx 1 root root 34 Sep 28 23:24 default → /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 37 Sep 29 10:01 trees.conf → /etc/nginx/sites-available/trees.conf
eng@bot:~$ cat /etc/nginx/sites-enabled/trees.conf
server {
    server_name t.enadv.site;
    # for LINE Bot
    location / {
        include uwsgi_params;
        #uwsgi_pass unix:/home/your_account/your_project/your_project.sock;
        uwsgi_pass 127.0.0.1:3000;
        #proxy_pass http://your_url;
    }
    # for Web or LIFF
}
```

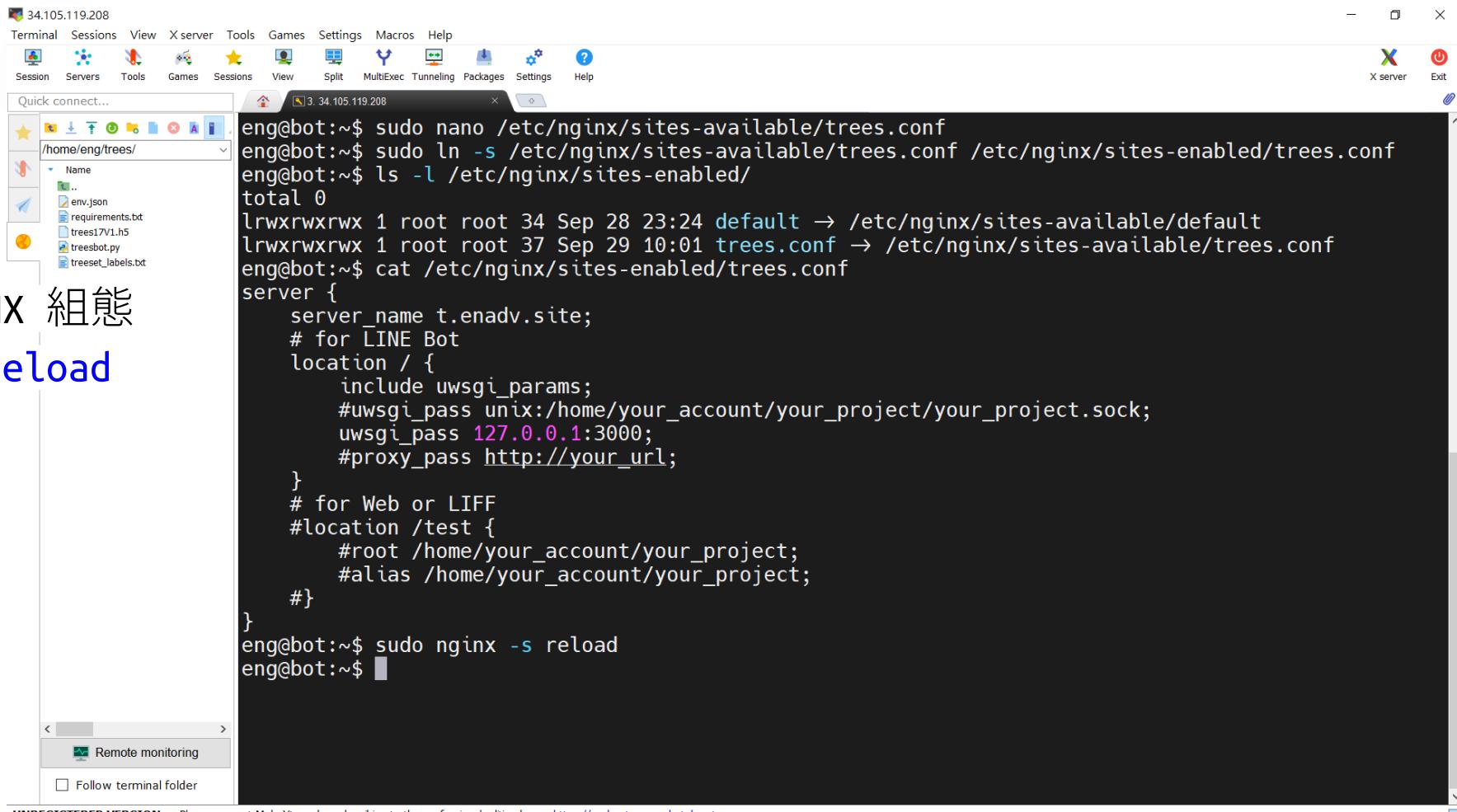
The screenshot shows a terminal session in MobaXterm connected to IP 34.105.119.208. The user has run several commands to manage Nginx configuration files. They first edit the file at /etc/nginx/sites-available/trees.conf, then create a symbolic link to it at /etc/nginx/sites-enabled/trees.conf. They then list the contents of the sites-enabled directory, which shows two symbolic links: 'default' pointing to /etc/nginx/sites-available/default and 'trees.conf' pointing to /etc/nginx/sites-available/trees.conf. Finally, they view the content of the 'trees.conf' file, which contains a basic Nginx server block configuration.

# 網站與憑證

## 1. 架設 NGINX

d. 載入新組態

```
# 重新載入 NGINX 組態  
sudo nginx -s reload
```



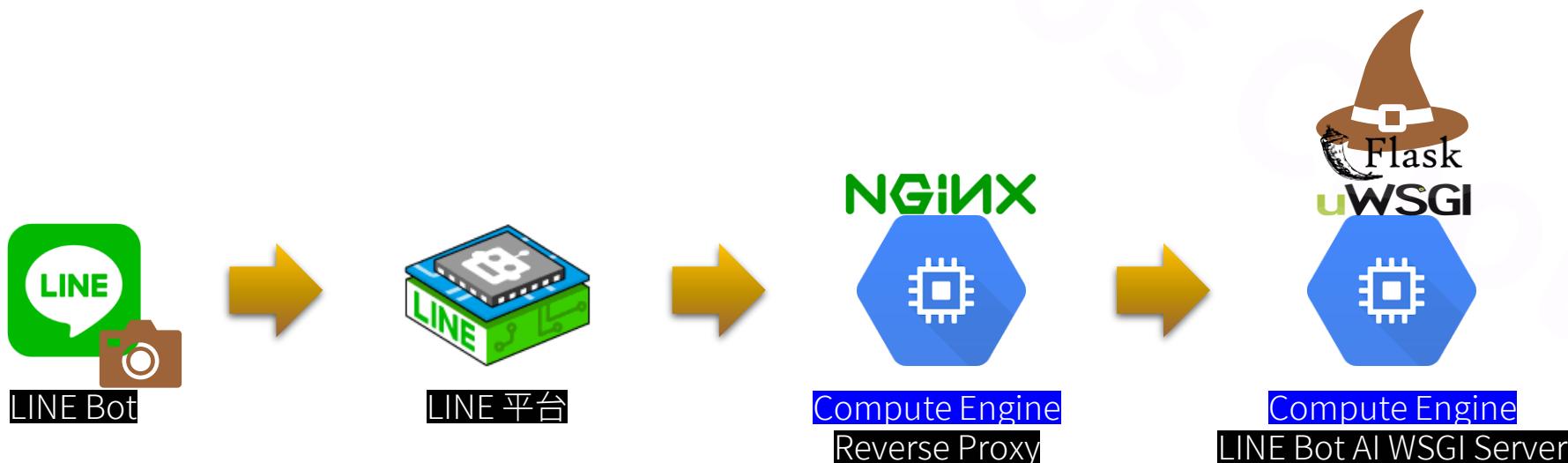
```
eng@bot:~$ sudo nano /etc/nginx/sites-available/trees.conf
eng@bot:~$ sudo ln -s /etc/nginx/sites-available/trees.conf /etc/nginx/sites-enabled/trees.conf
eng@bot:~$ ls -l /etc/nginx/sites-enabled/
total 0
lrwxrwxrwx 1 root root 34 Sep 28 23:24 default → /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root 37 Sep 29 10:01 trees.conf → /etc/nginx/sites-available/trees.conf
eng@bot:~$ cat /etc/nginx/sites-enabled/trees.conf
server {
    server_name t.enadv.site;
    # for LINE Bot
    location / {
        include uwsgi_params;
        #uwsgi_pass unix:/home/your_account/your_project/your_project.sock;
        uwsgi_pass 127.0.0.1:3000;
        #proxy_pass http://your_url;
    }
    # for Web or LIFF
    #location /test {
        #root /home/your_account/your_project;
        #alias /home/your_account/your_project;
    //}
}
eng@bot:~$ sudo nginx -s reload
eng@bot:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 網站與憑證

## 免費憑證機器人 Certbot 特色

- <https://certbot.eff.org/>
- 採用 Let's Encrypt 免費憑證
- 自動安裝且自動更新憑證

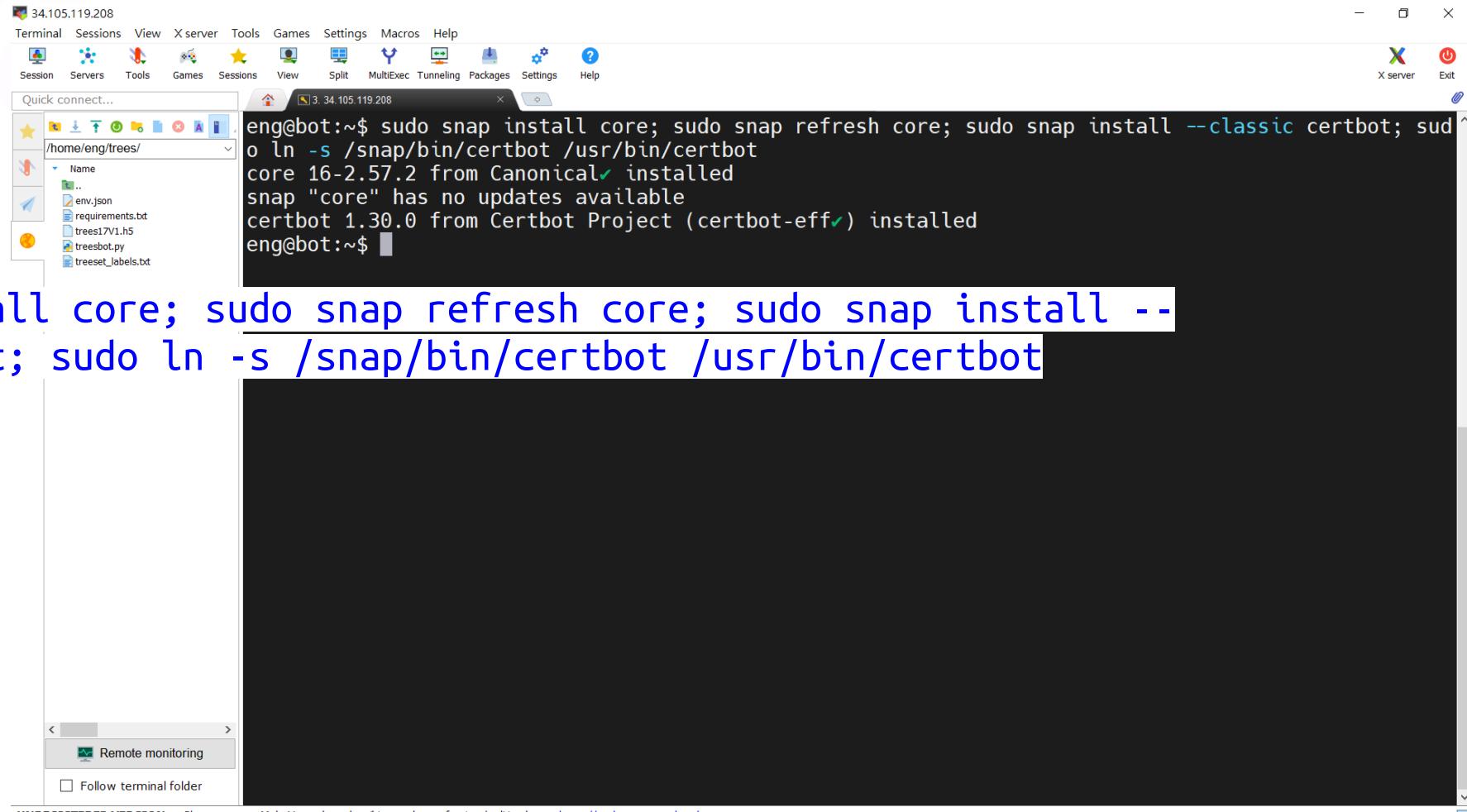


# 網站與憑證

## 2. 架設 Certbot

### a. 安裝 Certbot

```
sudo snap install core; sudo snap refresh core; sudo snap install --classic certbot; sudo ln -s /snap/bin/certbot /usr/bin/certbot
```



The screenshot shows a terminal window titled 'eng@bot:~\$' running on a Linux system. The window displays the following command and its execution:

```
eng@bot:~$ sudo snap install core; sudo snap refresh core; sudo snap install --classic certbot; sudo ln -s /snap/bin/certbot /usr/bin/certbot
core 16-2.57.2 from Canonical✓ installed
snap "core" has no updates available
certbot 1.30.0 from Certbot Project (certbot-eff✓) installed
eng@bot:~$
```

The terminal window is part of a larger interface with a file browser on the left and various application icons at the top. A status bar at the bottom indicates it's an 'UNREGISTERED VERSION'.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

`sudo certbot --nginx`

The screenshot shows a terminal window titled 'eng@bot:~\$' with the command 'sudo certbot --nginx' entered. The terminal output indicates the installation of Certbot via snap packages. A file browser window is also visible on the left, showing a folder structure under '/home/eng/trees/'. The status bar at the bottom of the terminal window displays 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net'.

```
eng@bot:~$ sudo snap install core; sudo snap refresh core; sudo snap install --classic certbot; sudo ln -s /snap/bin/certbot /usr/bin/certbot
core 16-2.57.2 from Canonical✓ installed
snap "core" has no updates available
certbot 1.30.0 from Certbot Project (certbot-eff✓) installed
eng@bot:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): 輸入 email
```

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

`sudo certbot --nginx`

The screenshot shows a terminal window in MobaXterm with the following session details:

- Session: 34.105.119.208
- Servers: X server
- Tools: Sessions
- Games: None
- View: None
- X server: None
- Tools: Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help
- Games: None
- Settings: None
- Macros: None
- Help: None

The terminal window displays the command output:

```
eng@bot:~$ sudo snap install core; sudo snap refresh core; sudo snap install --classic certbot; sudo ln -s /snap/bin/certbot /usr/bin/certbot
core 16-2.57.2 from Canonical✓ installed
snap "core" has no updates available
certbot 1.30.0 from Certbot Project (certbot-eff✓) installed
eng@bot:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): @gmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
(Y)es/(N)o: 接受服務條款，Y
```

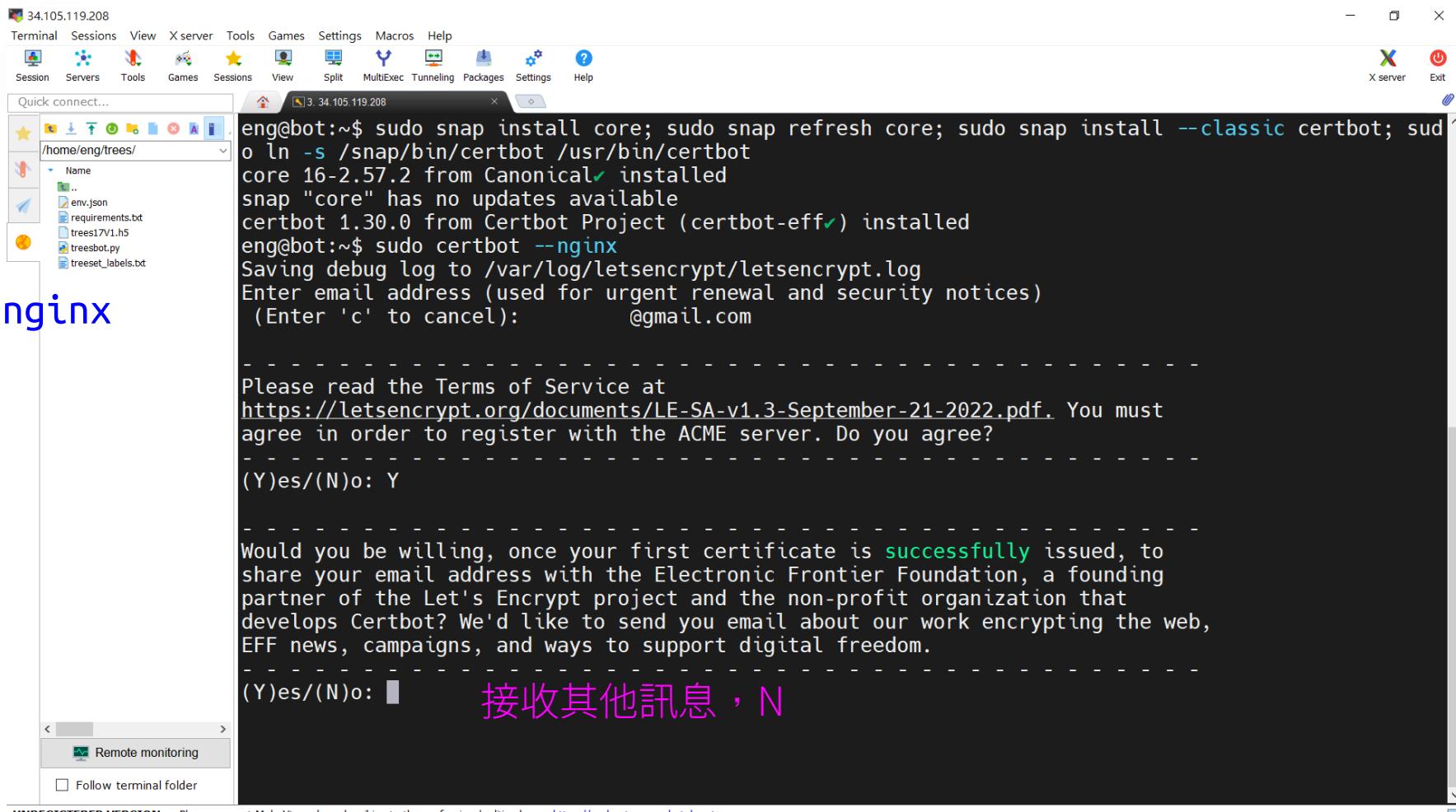
At the bottom of the terminal, there is a note: UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

`sudo certbot --nginx`



The screenshot shows a terminal window titled '34.105.119.208' running on a Linux system. The terminal displays the following command and its execution:

```
eng@bot:~$ sudo snap install core; sudo snap refresh core; sudo snap install --classic certbot; sudo ln -s /snap/bin/certbot /usr/bin/certbot
core 16-2.57.2 from Canonical✓ installed
snap "core" has no updates available
certbot 1.30.0 from Certbot Project (certbot-eff✓) installed
eng@bot:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): @gmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
(Y)es/(N)o: ■
```

A magenta annotation '接收其他訊息，N' is placed next to the '(Y)es/(N)o:' prompt.

At the bottom of the terminal window, there is a footer message: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

`sudo certbot --nginx`

The screenshot shows a terminal window titled 'eng@bot:~\$ sudo certbot --nginx'. The terminal displays the following command and its execution:

```
eng@bot:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): @gmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?

(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.

(Y)es/(N)o: N
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.

1: t.enadv.site
----- 指定憑證域名，1
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```

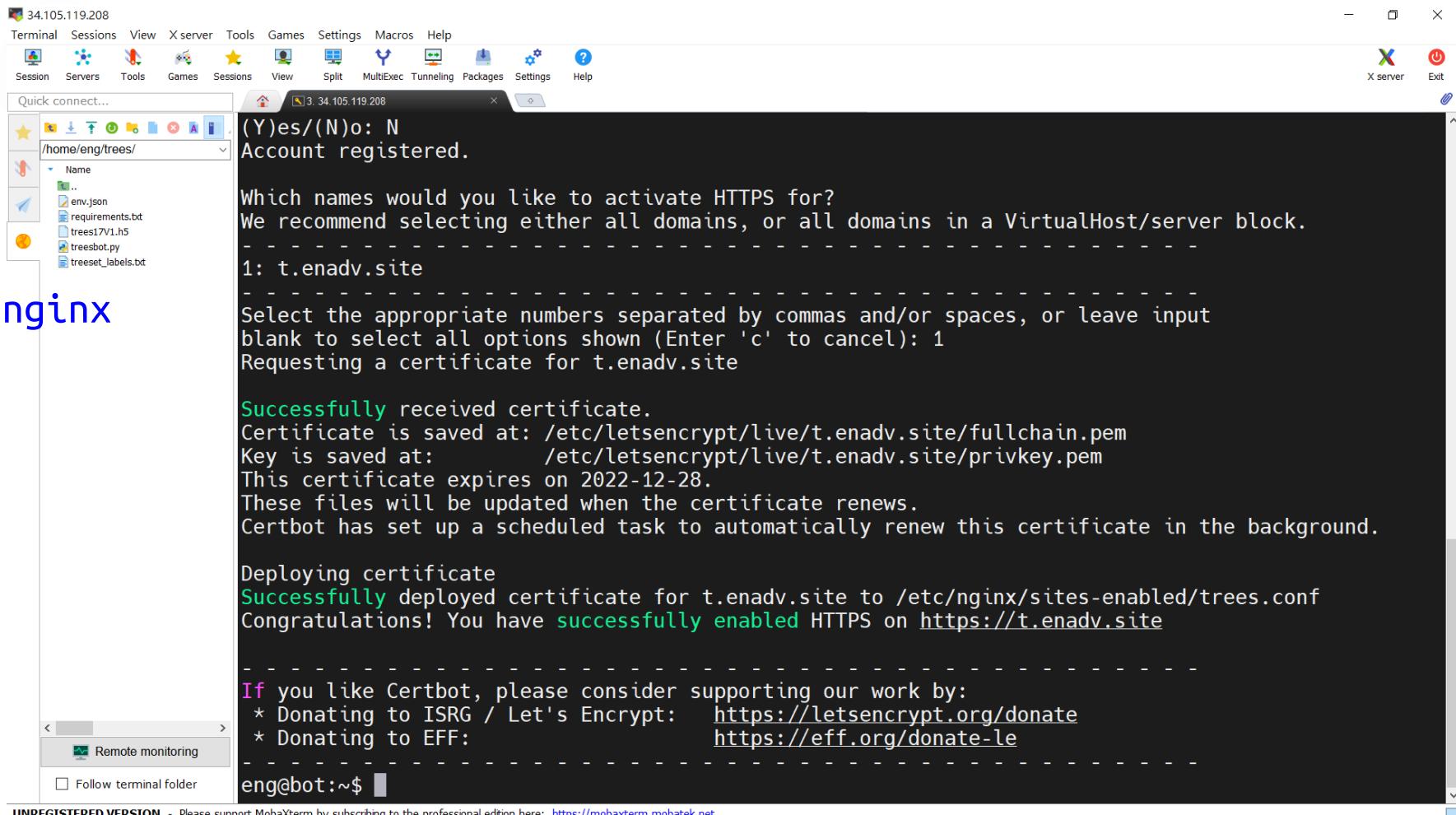
A pink annotation highlights the number '1' next to the domain 't.enadv.site'.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

```
sudo certbot --nginx
```



The screenshot shows a terminal window in MobaXterm with the following output:

```
(Y)es/(N)o: N
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: t.enadv.site
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1
Requesting a certificate for t.enadv.site

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/t.enadv.site/fullchain.pem
Key is saved at: /etc/letsencrypt/live/t.enadv.site/privkey.pem
This certificate expires on 2022-12-28.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for t.enadv.site to /etc/nginx/sites-enabled/trees.conf
Congratulations! You have successfully enabled HTTPS on https://t.enadv.site

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
eng@bot:~$
```

At the bottom of the terminal window, there is a watermark: UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

# 觀察 NGINX 組態檔

more /etc/nginx/sites-enabled/*your\_project.conf*

more /etc/nginx/sites-enabled/trees.conf

The screenshot shows a terminal window titled '34.105.119.208' running on MobaXterm. The window displays two configuration files: 'trees.conf' and 'certbot.conf'. The 'trees.conf' file contains standard NGINX server blocks for 't.enadv.site' and 'trees17V1.h5'. The 'certbot.conf' file is a generated SSL configuration managed by Certbot, containing directives for listening on port 443, using certificates from '/etc/letsencrypt/live/t.enadv.site/fullchain.pem', and returning HTTPS for the specified host.

```
server {
    server_name t.enadv.site;
    # for LINE Bot
    location / {
        include uwsgi_params;
        #uwsgi_pass unix:/home/your_account/your_project/your_project.sock;
        uwsgi_pass 127.0.0.1:3000;
        #proxy_pass http://your_url;
    }
    # for Web or LTFF
}

server {
    if ($host = t.enadv.site) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

--More-- (91%)
```

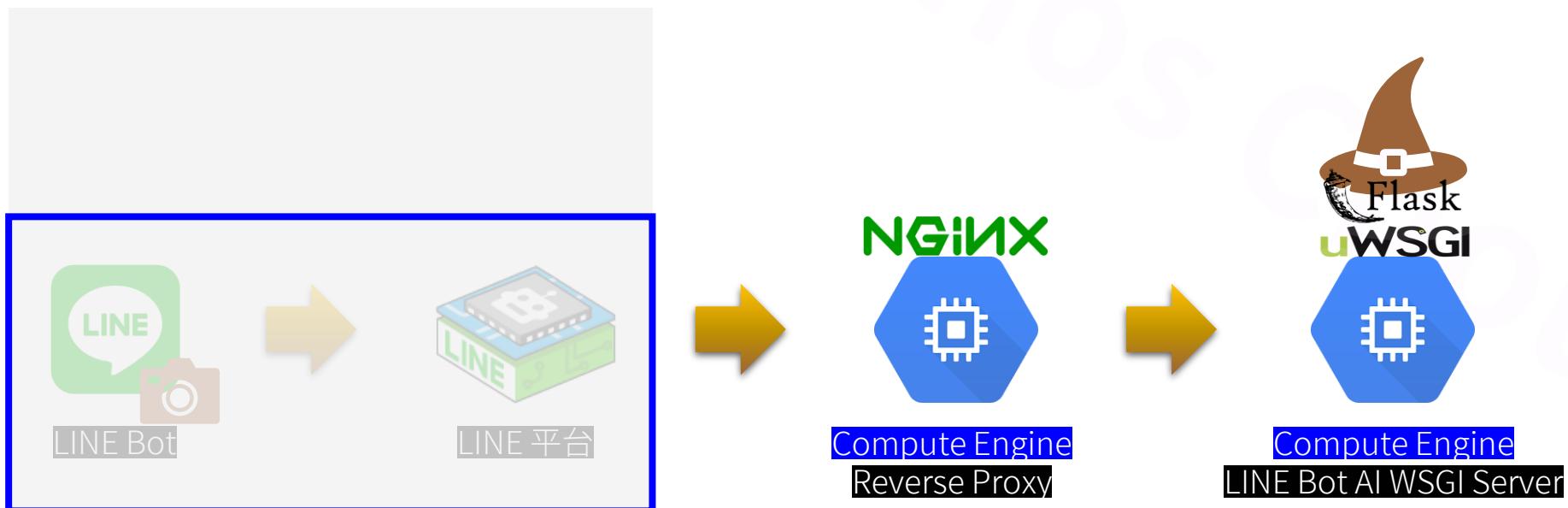
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 網站與憑證

## 2. 架設 Certbot

### c. (移除憑證)

- ① `sudo certbot delete --cert-name your_domain`
- ② 手動移除 `/etc/nginx/sites-enabled/your_project.conf` 中 Certbot  
增加的設定 (# managed by Certbot)



# 設定 LINE Messaging

## 1. 調整 LINE Messaging

### a. 調整 Webhook URL

The screenshot shows the LINE Messaging API settings page. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin', 'Tools', and 'Support'. The main area shows the breadcrumb 'TOP > test > trees > Messaging API'. Under 'Available APIs', it lists 'REPLY\_MESSAGE' and 'PUSH\_MESSAGE'. The 'Webhook settings' section contains a 'Webhook URL' field with the value 'https://t.enadv.site/callback', a 'Verify' button (highlighted with a pink box), and an 'Edit' button. Below that is a 'Use webhook' toggle switch (on). Further down are 'Webhook redelivery' (off) and 'Error statistics aggregation' (on).

TOP > test > trees > Messaging API

Available APIs ⓘ

- REPLY\_MESSAGE
- PUSH\_MESSAGE

Webhook settings

Webhook URL ⓘ https://t.enadv.site/callback

Verify Edit

Use webhook ⓘ

Webhook redelivery ⓘ

Error statistics aggregation ⓘ

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form .

⊕ Family sites English

# 設定 LINE Messaging

## 1. 調整 LINE Messaging

### a. 調整 Webhook URL

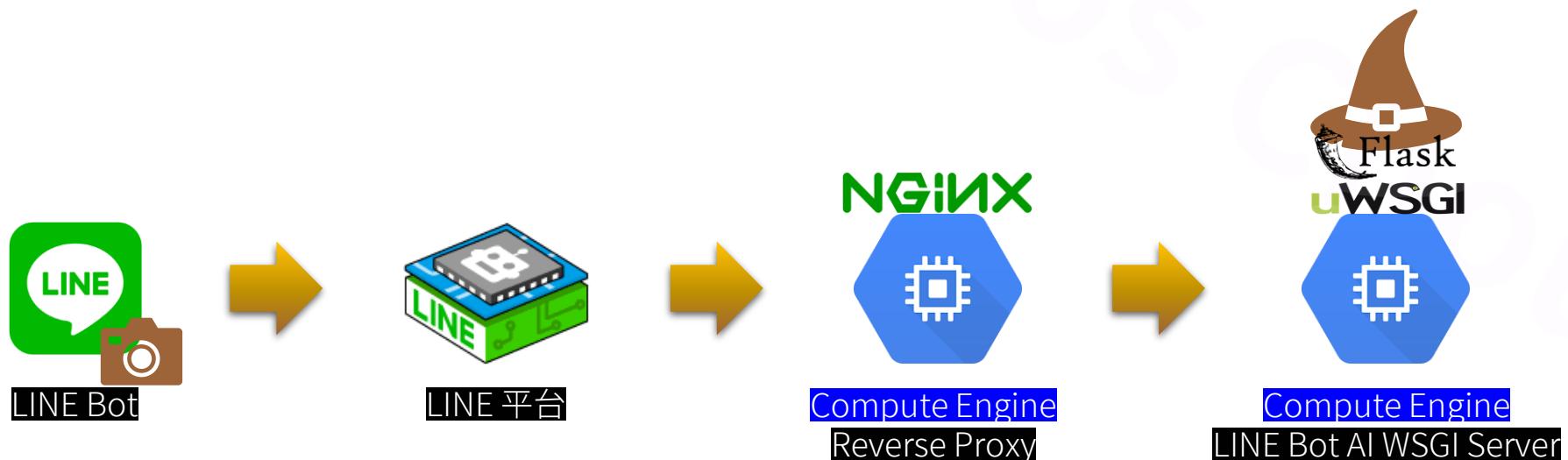
The screenshot shows the LINE Messaging API settings page. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin', 'Tools', and 'Support'. The main area shows 'Available APIs' with 'REPLY\_MESSAGE' and 'PUSH\_MESSAGE'. Under 'Webhook settings', the 'Webhook URL' is set to 'https://tenadv.site/callback', and a 'Verify' button is visible. A modal window titled 'Success' is overlaid on the page. Below the modal, the 'Use webhook' toggle is green, and other options like 'Webhook redelivery' and 'Error statistics aggregation' are also toggled on. At the bottom, there are links for '© LINE Corporation', 'Terms and policies', 'About trademarks', and 'Found any problems? Please use our inquiry form'. There are also buttons for 'Family sites' and 'English'.

# 設定 LINE Messaging

## 1. 調整 LINE Messaging

### b. LINE 測試





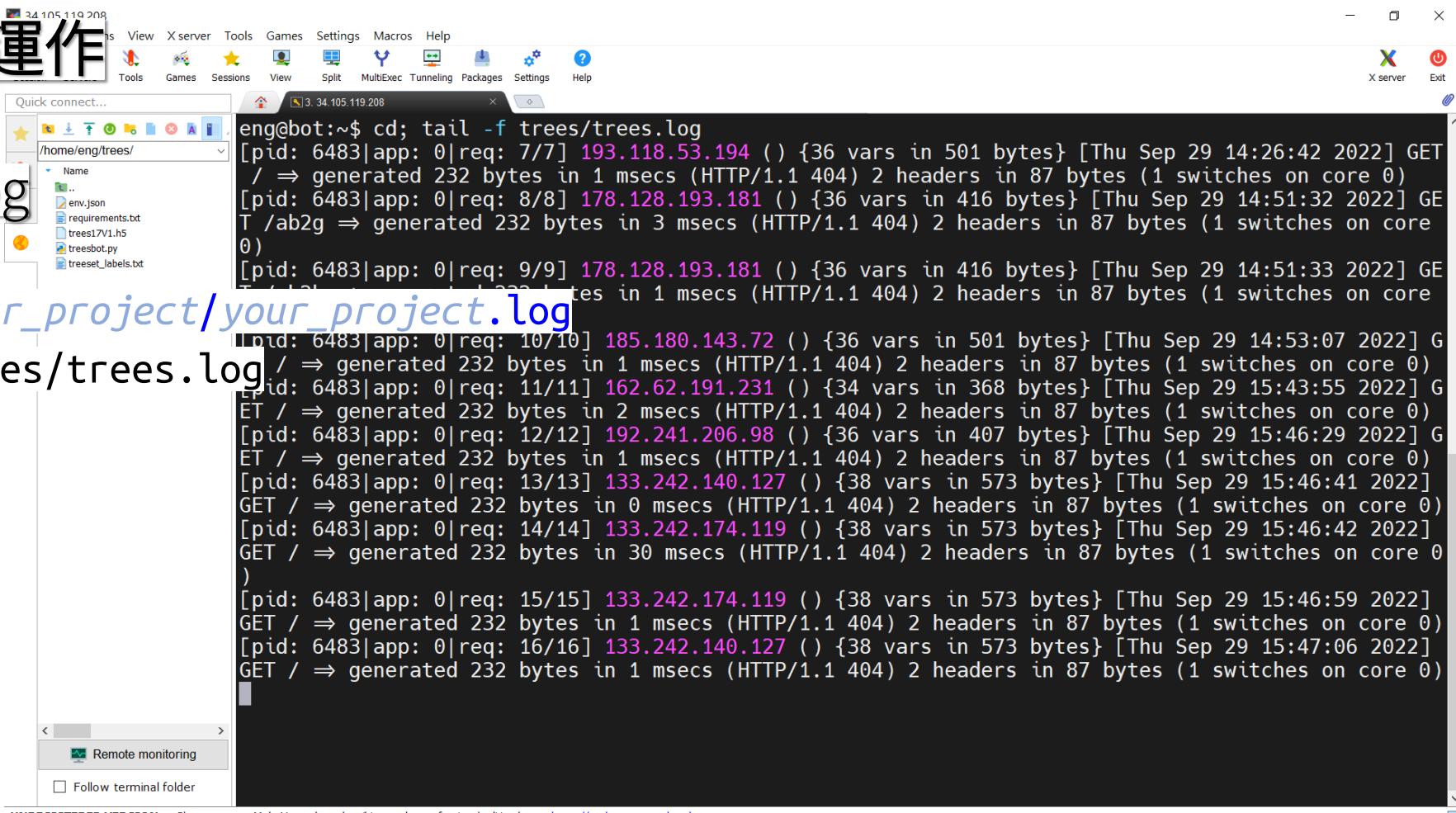
# Debug

## 1. 確認 LINE Bot 運作

### a. 檢視 uWSGI log

`cd; tail -f your_project/your_project.log`

`cd; tail -f trees/trees.log`



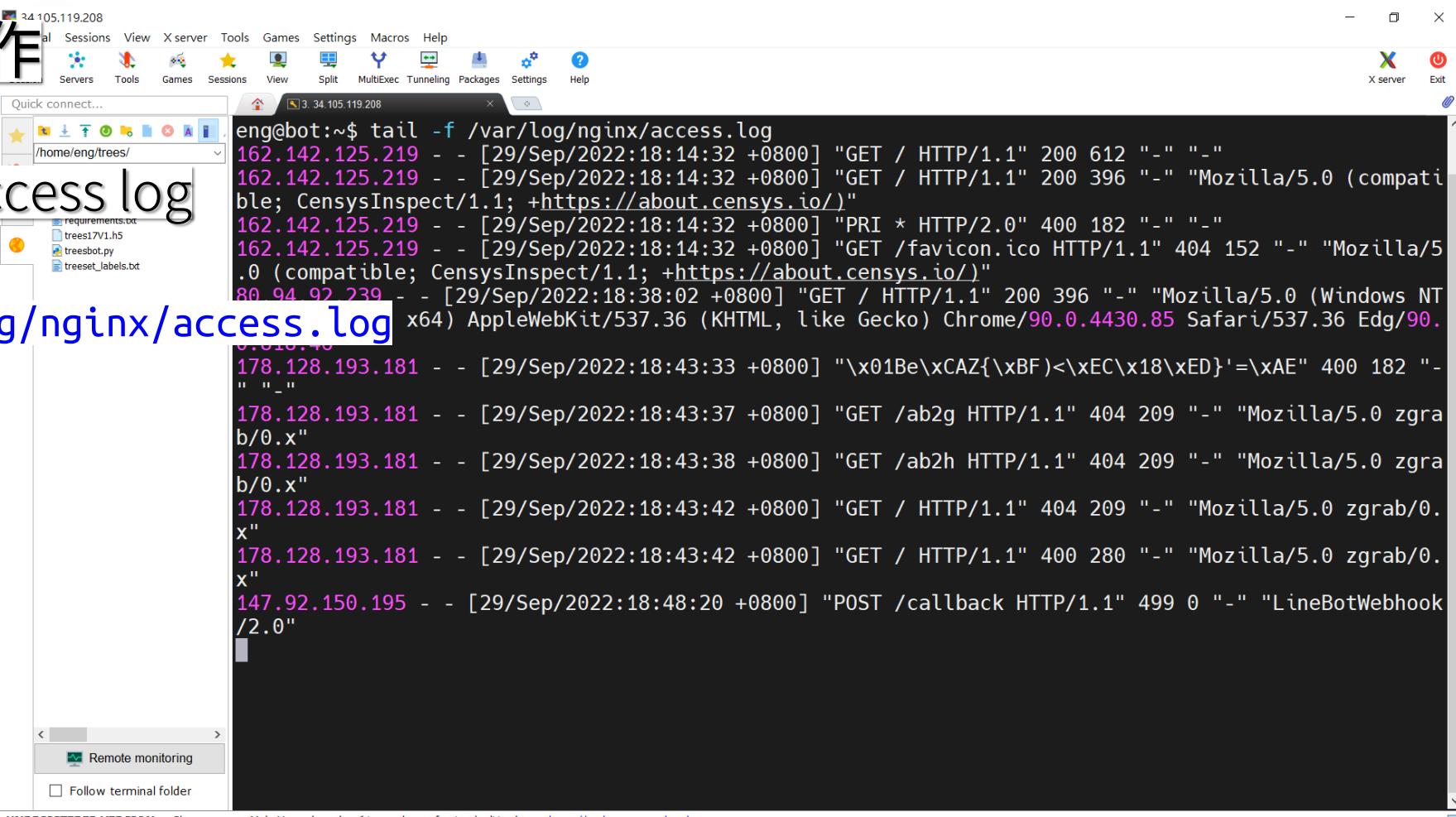
The screenshot shows a MobaXterm window with two panes. The left pane displays a file browser with a directory tree for '/home/eng/trees/'. The right pane is a terminal window titled '3. 34.105.119.208' showing log output from a uWSGI application. The logs show multiple requests from various IP addresses (e.g., 193.118.53.194, 178.128.193.181, 185.180.143.72, 162.62.191.231, 192.241.206.98) with details like request ID, IP, method (GET), path, variables, and response status.

```
eng@bot:~$ cd; tail -f trees/trees.log
[pid: 6483|app: 0|req: 7/7] 193.118.53.194 () {36 vars in 501 bytes} [Thu Sep 29 14:26:42 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 8/8] 178.128.193.181 () {36 vars in 416 bytes} [Thu Sep 29 14:51:32 2022] GET / → generated 232 bytes in 3 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 9/9] 178.128.193.181 () {36 vars in 416 bytes} [Thu Sep 29 14:51:33 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 10/10] 185.180.143.72 () {36 vars in 501 bytes} [Thu Sep 29 14:53:07 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 11/11] 162.62.191.231 () {34 vars in 368 bytes} [Thu Sep 29 15:43:55 2022] GET / → generated 232 bytes in 2 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 12/12] 192.241.206.98 () {36 vars in 407 bytes} [Thu Sep 29 15:46:29 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 13/13] 133.242.140.127 () {38 vars in 573 bytes} [Thu Sep 29 15:46:41 2022] GET / → generated 232 bytes in 0 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 14/14] 133.242.174.119 () {38 vars in 573 bytes} [Thu Sep 29 15:46:42 2022] GET / → generated 232 bytes in 30 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 15/15] 133.242.174.119 () {38 vars in 573 bytes} [Thu Sep 29 15:46:59 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
[pid: 6483|app: 0|req: 16/16] 133.242.140.127 () {38 vars in 573 bytes} [Thu Sep 29 15:47:06 2022] GET / → generated 232 bytes in 1 msecs (HTTP/1.1 404) 2 headers in 87 bytes (1 switches on core 0)
```

# Debug

## 2. 確認 NGINX 運作

### a. 檢視 NGINX access log



The screenshot shows a MobaXterm session titled 'eng@bot:~\$' running on port 105.119.208. The terminal window displays the output of the command 'tail -f /var/log/nginx/access.log'. The logs show multiple entries from different IP addresses, mostly from 162.142.125.219, indicating various HTTP requests like GET / favicon.ico and POST /callback.

```
eng@bot:~$ tail -f /var/log/nginx/access.log
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 612 "-" "-"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "PRI * HTTP/2.0" 400 182 "-" "-"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET /favicon.ico HTTP/1.1" 404 152 "-" "Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)"
80.94.92.239 - - [29/Sep/2022:18:38:02 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (Windows NT x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edge/90.0.818.40"
178.128.193.181 - - [29/Sep/2022:18:43:33 +0800] "\x01Be\xCAZ{\xBF}<\xEC\x18\xED}'=\xAE" 400 182 "-"
178.128.193.181 - - [29/Sep/2022:18:43:37 +0800] "GET /ab2g HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:38 +0800] "GET /ab2h HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:42 +0800] "GET / HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:42 +0800] "GET / HTTP/1.1" 400 280 "-" "Mozilla/5.0 zgrab/0.x"
147.92.150.195 - - [29/Sep/2022:18:48:20 +0800] "POST /callback HTTP/1.1" 499 0 "-" "LineBotWebhook/2.0"
```

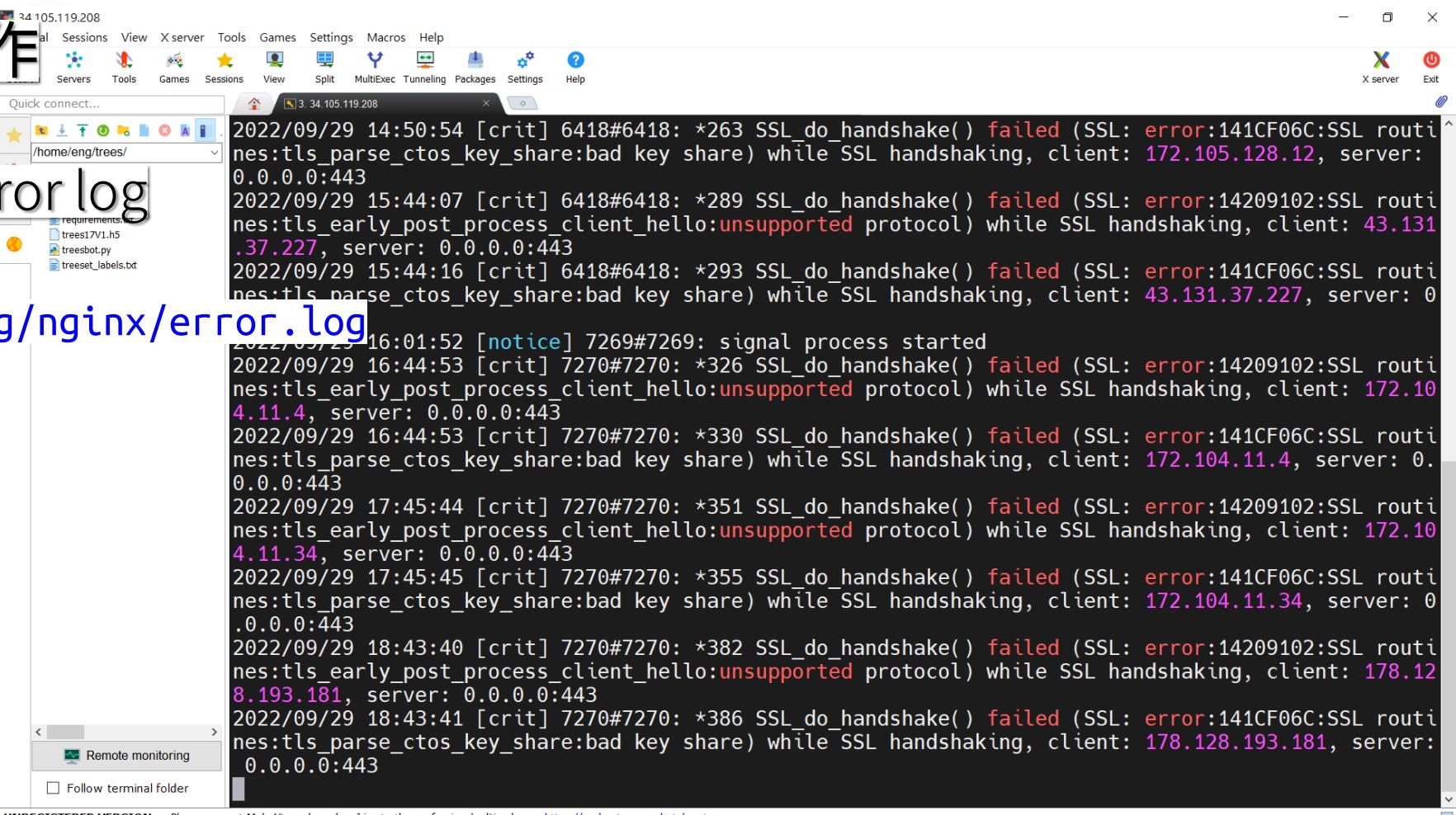
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# Debug

## 2. 確認 NGINX 運作

### b. 檢視 NGINX error log

```
tail -f /var/log/nginx/error.log
```

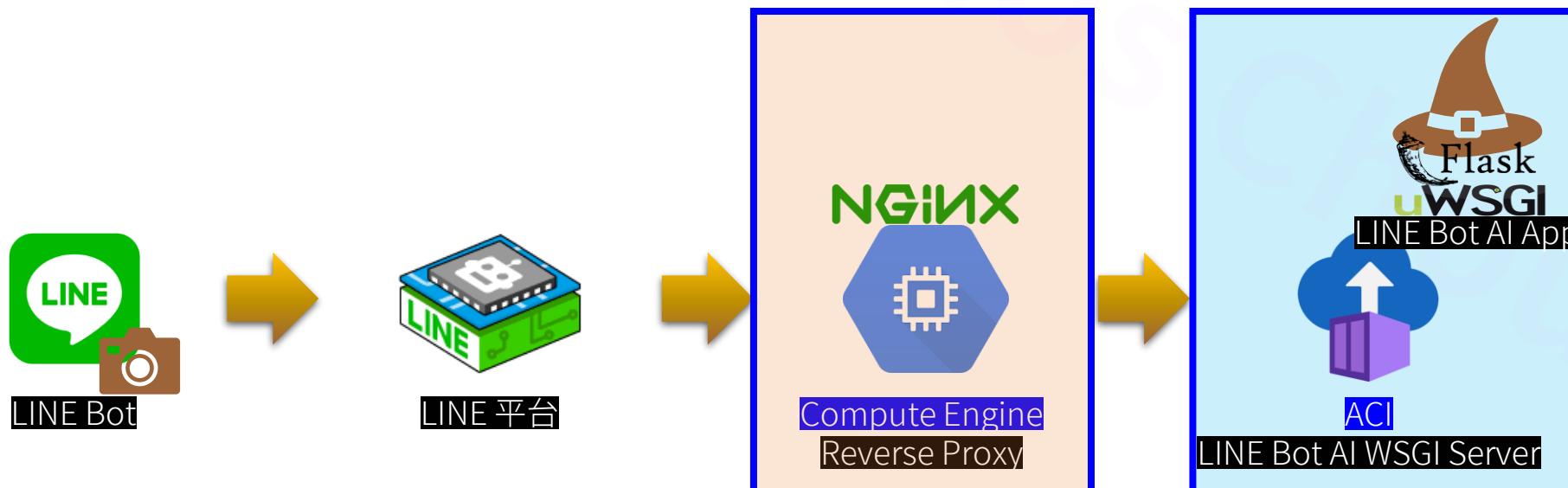


```
2022/09/29 14:50:54 [crit] 6418#6418: *263 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.105.128.12, server: 0.0.0.0:443
2022/09/29 15:44:07 [crit] 6418#6418: *289 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 43.131.37.227, server: 0.0.0.0:443
2022/09/29 15:44:16 [crit] 6418#6418: *293 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 43.131.37.227, server: 0.0.0.0:443
2022/09/29 16:01:52 [notice] 7269#7269: signal process started
2022/09/29 16:44:53 [crit] 7270#7270: *326 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 172.104.11.4, server: 0.0.0.0:443
2022/09/29 16:44:53 [crit] 7270#7270: *330 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.104.11.4, server: 0.0.0.0:443
2022/09/29 17:45:44 [crit] 7270#7270: *351 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 172.104.11.34, server: 0.0.0.0:443
2022/09/29 17:45:45 [crit] 7270#7270: *355 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.104.11.34, server: 0.0.0.0:443
2022/09/29 18:43:40 [crit] 7270#7270: *382 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 178.128.193.181, server: 0.0.0.0:443
2022/09/29 18:43:41 [crit] 7270#7270: *386 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 178.128.193.181, server: 0.0.0.0:443
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# Solution 2 - Mixed 部署

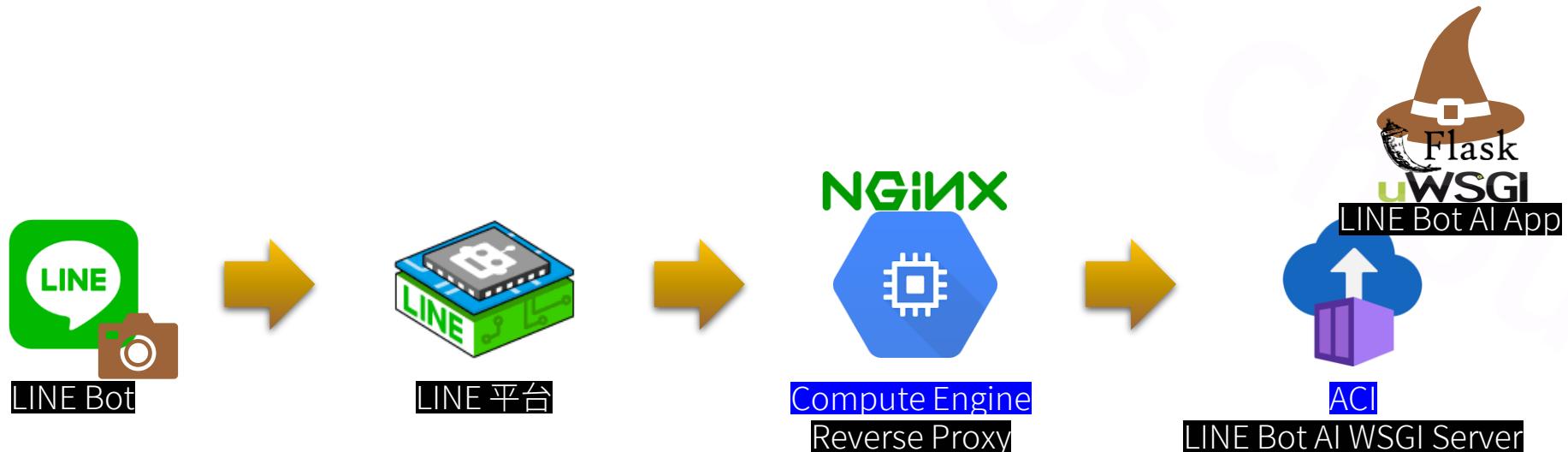
1. 長期可用且廉價的硬體環境 ... Google Compute Engine + Azure ACI
2. Flask as Web Server 的替代方案 ... NGINX + Domain + uWSGI
3. 長期可用且廉價的 SSL 網域方案 ... Let's Encrypt + Certbot

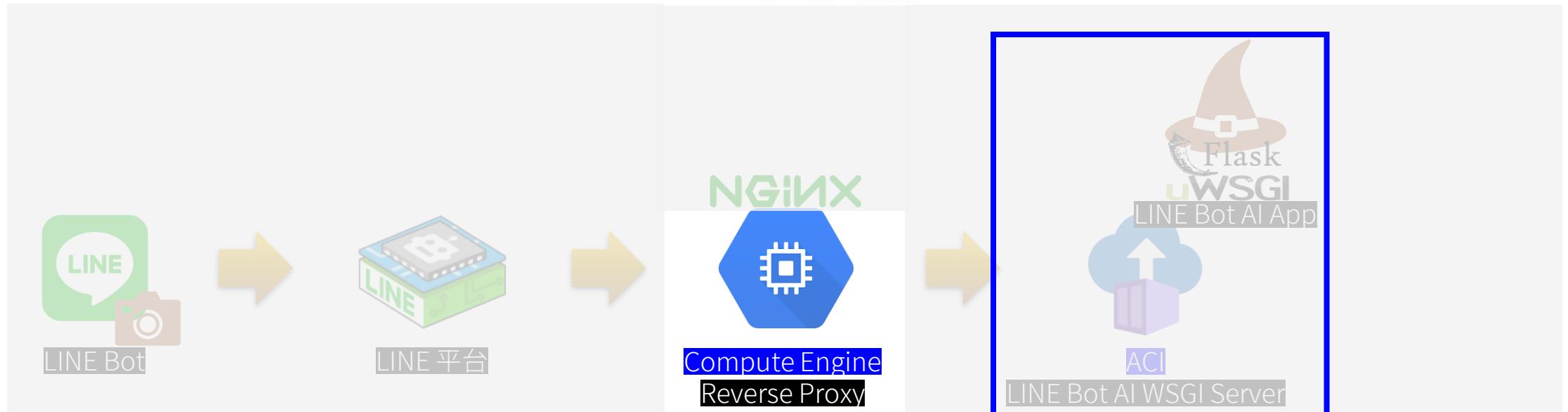


# 任務

1. 上傳程式至 Cloud Shell
2. 建立 Container 並推送至容器登錄
3. 部署 Container 至容器執行個體
4. 調整 Google Compute Engine VM 之 NGINX 轉導設定

# 流程





# LINE Bot & WSGI

## 1. 準備程式碼

### a. 下載範例程式並調整

- ① treesbot.py
- ② trees17V1.h5
- ③ treeset\_labels.txt
- ④ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_FILE
- ⑤ other tree samples

# LINE Bot & WSGI

## 1. 準備程式碼

### b. 製作 requirements.txt

line-bot-sdk

flask

pillow

tensorflow==2.4.4

uwsgi

# LINE Bot & WSGI

## 1. 準備程式碼

### c. 製作 Dockerfile

Note

調整 *your\_module*

```
FROM ubuntu:18.04
WORKDIR /your_module
COPY . .
ENV TZ=Asia/Taipei
RUN mkdir var && \
    ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && \
    echo $TZ > /etc/timezone && \
    apt-get update && \
    apt-get install -y python3-pip tzdata && \
    dpkg-reconfigure -f noninteractive tzdata && \
    python3 -m pip install --upgrade pip && \
    python3 -m pip install -r requirements.txt
CMD uwsgi -w your_module:app --http-socket :$PORT
```

# LINE Bot & WSGI

## 1. 準備程式碼

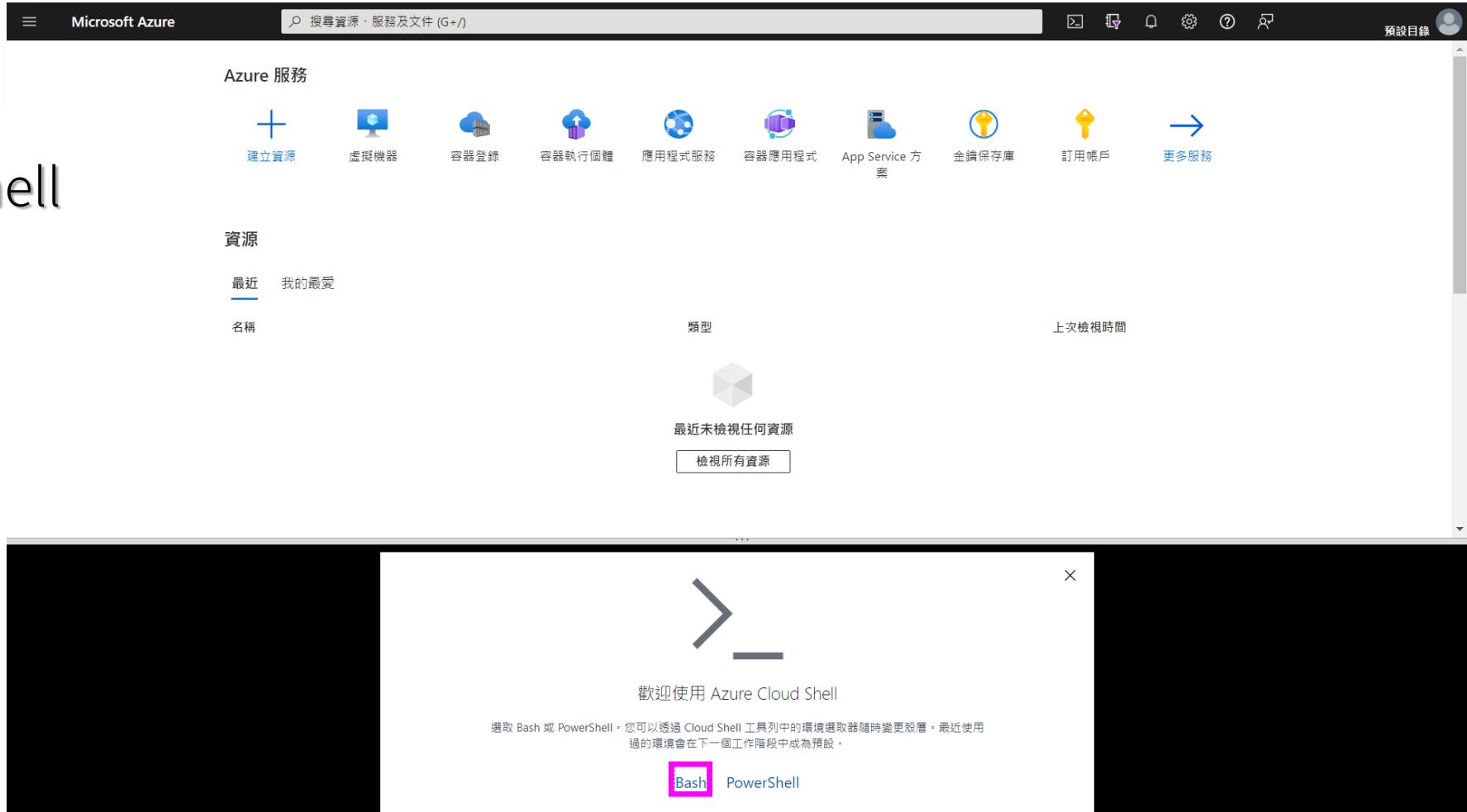
d. 啟動 Cloud Shell

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for account settings and help. Below the navigation bar, the main content area has a title "Azure 服務" (Azure Services). Under "Azure 服務", there are several service icons: 建立資源 (Create Resource), 虛擬機器 (Virtual Machines), 容器登錄 (Container Registry), 容器執行個體 (Container Instances), 應用程式服務 (App Service), 容器應用程式 (Container Apps), App Service 方案 (App Service Plan), 金鑰保存庫 (Key Vault), 訂用帳戶 (Subscription), and 更多服務 (More Services). Below this, there's a section titled "資源" (Resources) with tabs for "最近" (Recent) and "我的最愛" (Favorites). It includes columns for "名稱" (Name), "類型" (Type), and "上次檢視時間" (Last checked time). A message says "最近未檢視任何資源" (No resources checked recently) with a "檢視所有資源" (View all resources) button. At the bottom, there are sections for "瀏覽" (Browse) with links to 訂用帳戶 (Subscription), 資源群組 (Resource Group), 所有資源 (All Resources), and 儀表板 (Dashboard); and "工具" (Tools) with links to Microsoft Learn, Azure 監視器 (Azure Monitor), 適用於雲端的 Microsoft Defender (Microsoft Defender for Cloud), and 成本管理 (Cost Management).

# LINE Bot & WSGI

## 1. 準備程式碼

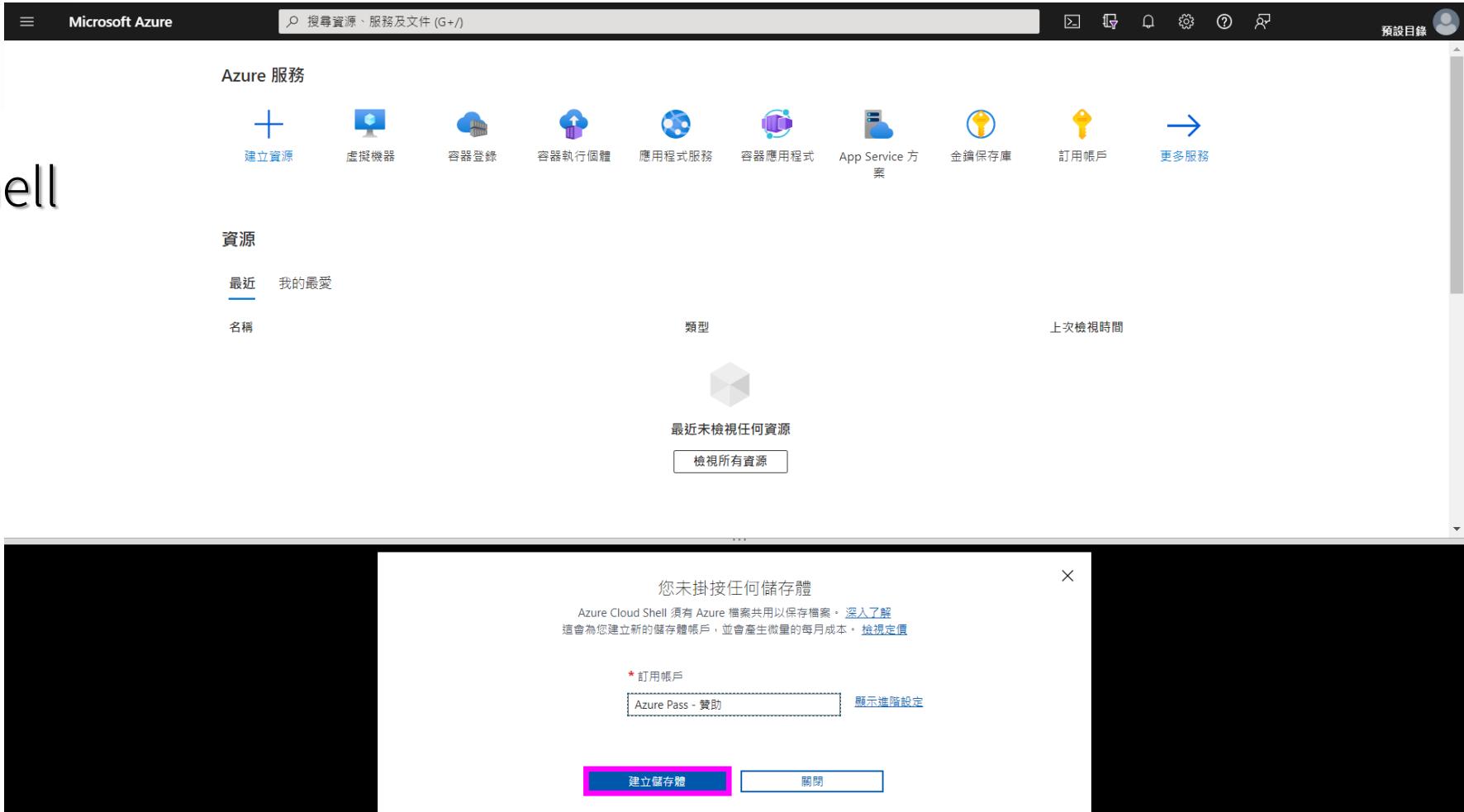
d. 啟動 Cloud Shell



# LINE Bot & WSGI

## 1. 準備程式碼

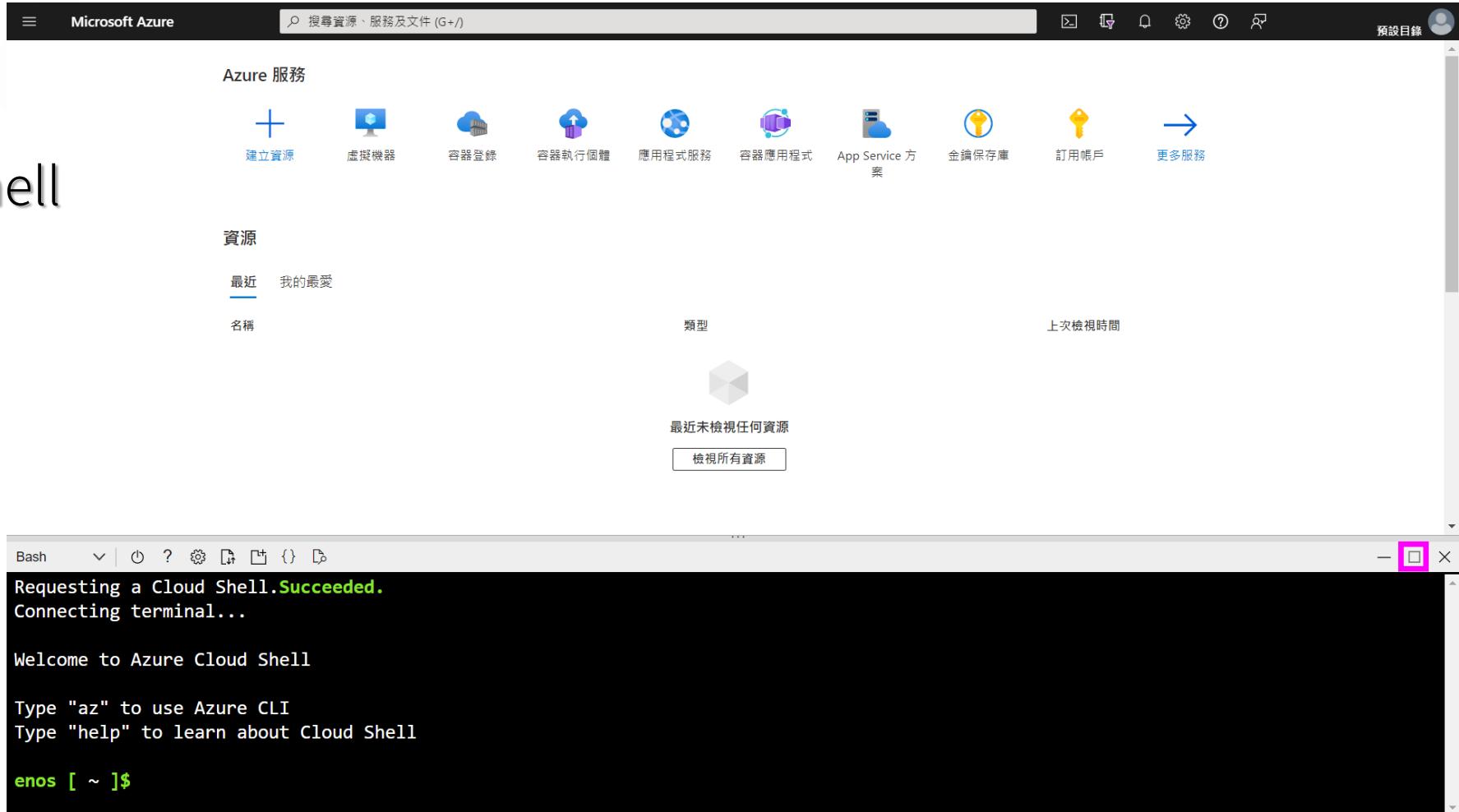
d. 啟動 Cloud Shell



# LINE Bot & WSGI

## 1. 準備程式碼

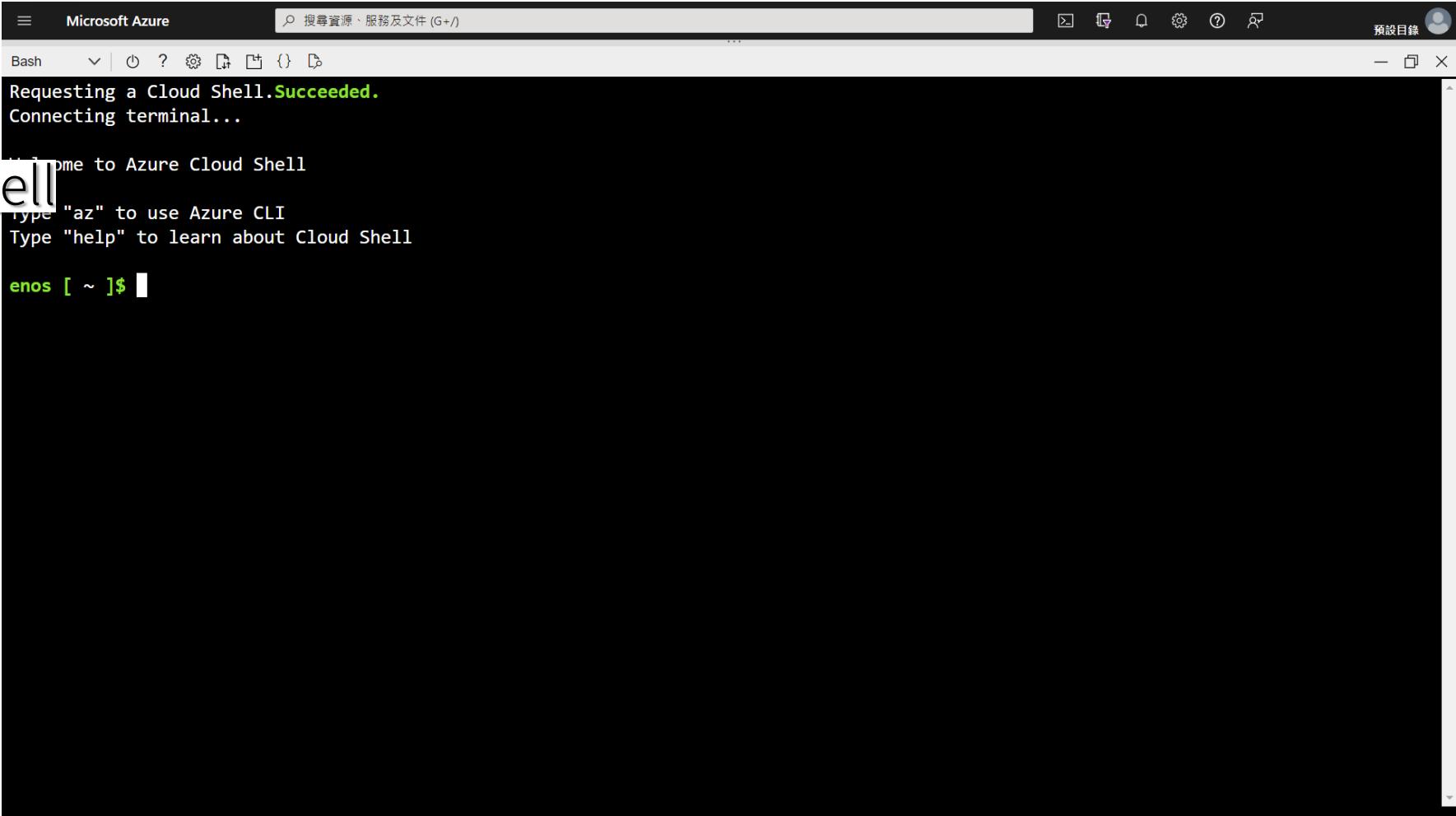
### d. 啟動 Cloud Shell



# LINE Bot & WSGI

## 1. 準備程式碼

### d. 啟動 Cloud Shell



The screenshot shows a Microsoft Azure Cloud Shell terminal window. The title bar says "Microsoft Azure" and "Bash". The main area displays the following text:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

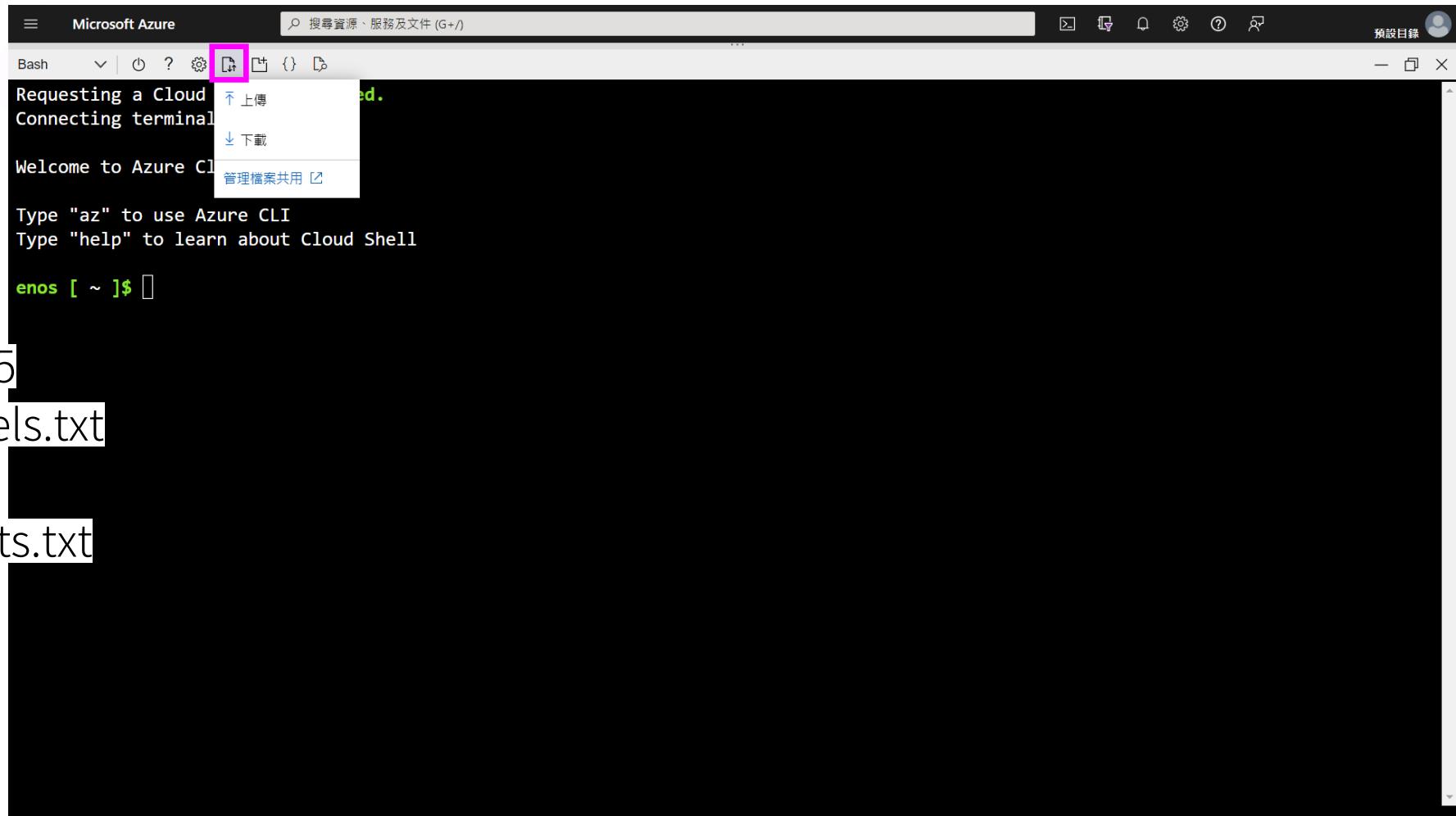
enos [ ~ ]$
```

# LINE Bot & WSGI

## 1. 準備程式碼

e. 上傳下列程式

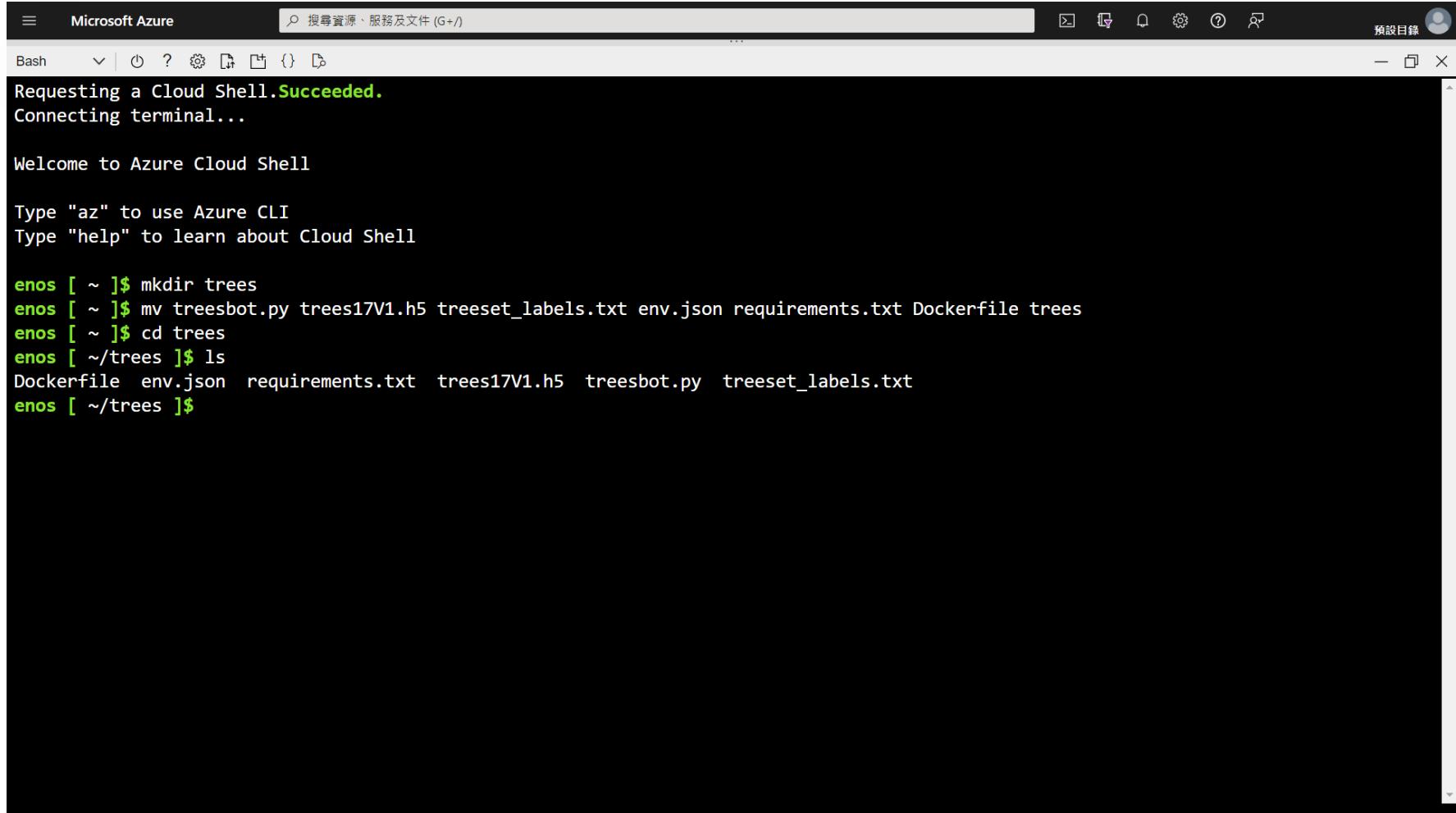
- ① treesbot.py
- ② trees17V1.h5
- ③ treeset\_labels.txt
- ④ env.json
- ⑤ requirements.txt
- ⑥ Dockerfile



# LINE Bot & WSGI

## 1. 準備程式碼

### f. 移至專案目錄



The screenshot shows a Microsoft Azure Cloud Shell interface. The title bar says "Microsoft Azure" and "Bash". The search bar contains "搜尋資源、服務及文件 (G+ /)". The terminal window displays the following text:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

enos [ ~ ]$ mkdir trees
enos [ ~ ]$ mv treesbot.py trees17V1.h5 treeset_labels.txt env.json requirements.txt Dockerfile trees
enos [ ~ ]$ cd trees
enos [ ~/trees ]$ ls
Dockerfile env.json requirements.txt trees17V1.h5 treesbot.py treeset_labels.txt
enos [ ~/trees ]$
```

# LINE Bot & WSGI

## 2. 啟動容器登入

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for account management and help.

The main area is titled "Azure 服務" (Azure Services). On the left, there's a sidebar with sections for "建立資源" (Create Resource), "虛擬機器" (Virtual Machines), and "容器登錄" (Container Registry). Below this is a "資源" (Resources) section with tabs for "最近" (Recent) and "我的最愛" (Favorites).

The central content area is titled "容器登錄" (Container Registry) and contains a "建立" (Create) button highlighted with a pink rectangle. It also includes a "檢視" (View) link and a detailed description of the service.

Below the main content, there are sections for "Microsoft 提供的免費訓練" (Free Microsoft Training) featuring three video links: "使用 ACR 建置及存取容器映像" (6 個單位, 49 分鐘), "使用 Docker 建置容器化 Web 應用程式" (8 個單位, 57 分鐘), and "使用 Azure App Service 部署及執行容器化 ..." (8 個單位, 46 分鐘).

At the bottom, there are several navigation links: "訂用帳戶" (Subscription), "資源群組" (Resource Group), "所有資源" (All Resources), "儀表板" (Dashboard), "Microsoft Learn" (101 lessons), "Azure 監視器" (Monitor), "適用於雲端的 Microsoft Defender" (Cloud-native Microsoft Defender), and "成本管理" (Cost Management).

# LINE Bot & WSGI

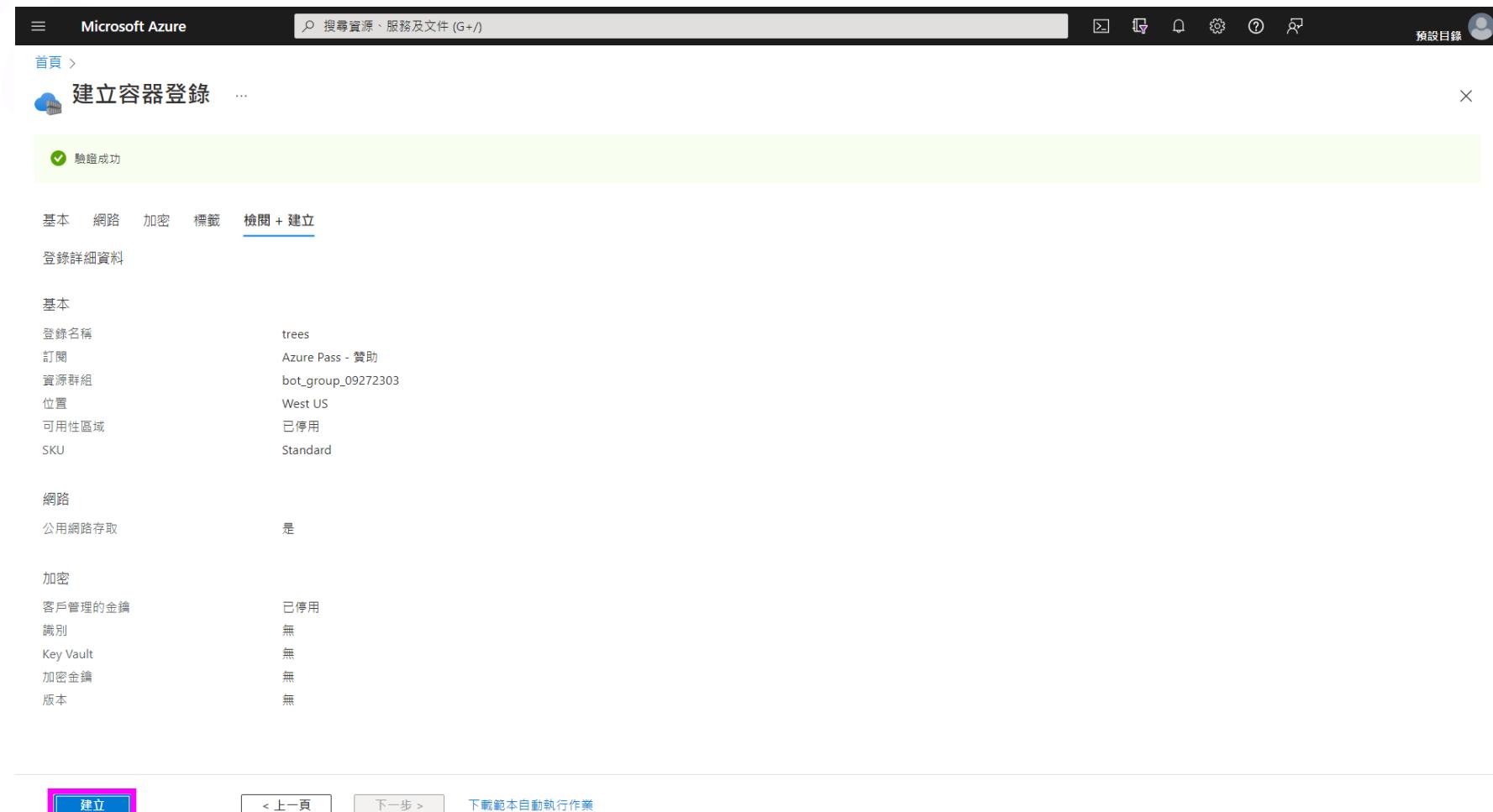
## 2. 啟動容器登入

The screenshot shows the Microsoft Azure portal interface for creating a new container registry. The top navigation bar includes the Microsoft Azure logo, a search bar, and various navigation icons. The main title is "建立容器登錄" (Create Container Registry). The "Basic" tab is selected under the "Container Registry" section. The "Subscription" dropdown is set to "Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d1f3)". The "Resource Group" dropdown is set to "bot\_group\_09272303". The "Registry Name" field contains "trees.azurecr.io". The "Location" dropdown is set to "West US". The "SKU" dropdown is set to "Standard". A note at the bottom states: "In the following registries and supported regions, availability zones are enabled. Learn more". At the bottom of the page are buttons for "檢閱 + 建立" (Review + Create) and "下一步: 網路" (Next: Network).

.azurecr.io # global unique

# LINE Bot & WSGI

## 2. 啟動容器登入



The screenshot shows the Microsoft Azure portal interface for creating a container registry. The title bar reads "Microsoft Azure". The main content area is titled "建立容器登錄" (Create Container Registry) with a "驗證成功" (Verification successful) message. The "檢閱 + 建立" (Review + Create) tab is selected. The "登錄詳細資料" (Login details) section displays the following configuration:

基本	網路	加密	標籤
登錄名稱 訂閱 資源群組 位置 可用性區域 SKU	trees Azure Pass - 贊助 bot_group_09272303 West US 已停用 Standard		
網路	公用網路存取 是		
加密			
客戶管理的金鑰 識別 Key Vault 加密金鑰 版本	已停用 無 無 無		

At the bottom, there are three buttons: "建立" (Create) highlighted with a pink box, "< 上一頁" (Previous page), and "下一步 >" (Next step).

# LINE Bot & WSGI

## 2. 啟動容器登入

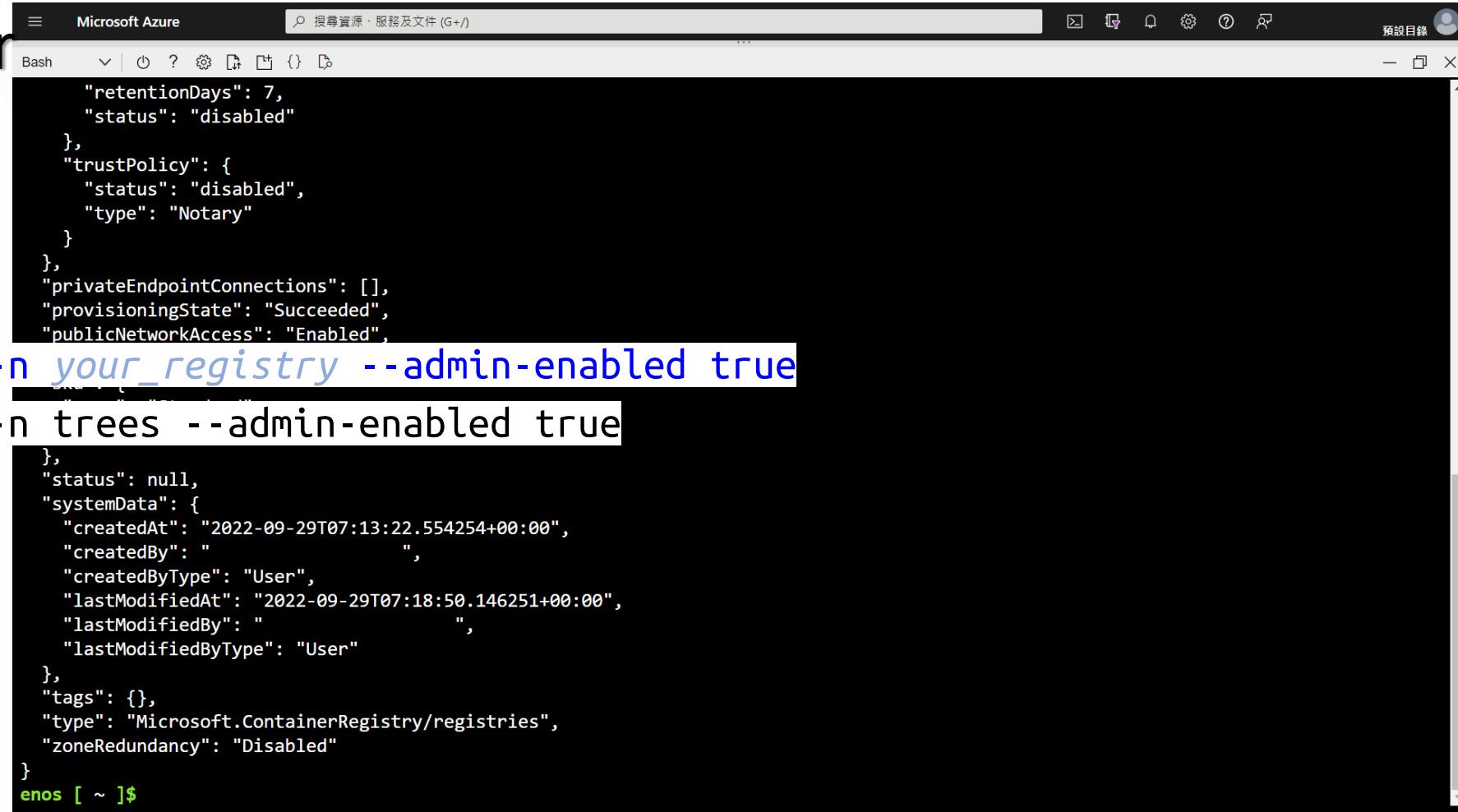
The screenshot shows the Microsoft Azure Container Registry deployment status page. The title bar reads "Microsoft Azure" and the main title is "Microsoft.ContainerRegistry | 概觀". The deployment status is shown as "您的部署已完成" (Deployment completed) with a green checkmark icon. Deployment details include: 部署名稱: Microsoft.ContainerRegistry, 開始時間: 29/9/2022 下午3:13:16, 訂用帳戶: Azure Pass - 賽助 (0fdd0a2d-5af8-4589-8206-0a7eae00d...), 相互關聯識別碼: bf05508a-bbc7-4432-8b4f-44ed0bef2a01, 資源群組: bot\_group\_09272303. Below this, there are sections for "部署詳細資料" (Deployment details) and "後續步驟" (Next steps). A prominent blue button labeled "前往資源" (Go to resource) is centered at the bottom. The right sidebar contains links for "成本管理" (Cost management), "適用於雲端的 Microsoft Defender" (Microsoft Defender for Cloud), "免費 Microsoft 教學課程" (Free Microsoft training courses), and "諮詢專家" (Consultant experts).

# LINE Bot & WSGI

## 3. 建立 Container

### a. 授權容器使用

```
# Cloud Shell
az acr update -n your_registry --admin-enabled true
az acr update -n trees --admin-enabled true
```



The screenshot shows a Microsoft Azure Cloud Shell interface. The title bar says "Microsoft Azure" and "搜尋資源、服務及文件 (G+ /)". The main area is a terminal window titled "Bash". It displays the command "az acr update" being run twice. The first command is "az acr update -n your\_registry --admin-enabled true", where "your\_registry" is highlighted in blue. The second command is "az acr update -n trees --admin-enabled true". Below these commands, the terminal shows a JSON object representing a registry resource, followed by the prompt "enos [ ~ ]\$". The background of the slide features a green gradient at the bottom.

```
"retentionDays": 7,
"status": "disabled"
},
"trustPolicy": {
    "status": "disabled",
    "type": "Notary"
}
},
"privateEndpointConnections": [],
"provisioningState": "Succeeded",
"publicNetworkAccess": "Enabled",
},
"status": null,
"systemData": {
    "createdAt": "2022-09-29T07:13:22.554254+00:00",
    "createdBy": "",
    "createdByType": "User",
    "lastModifiedAt": "2022-09-29T07:18:50.146251+00:00",
    "lastModifiedBy": "",
    "lastModifiedByType": "User"
},
"tags": {},
"type": "Microsoft.ContainerRegistry/registries",
"zoneRedundancy": "Disabled"
}
```

enos [ ~ ]\$

# LINE Bot & WSGI

## 3. 建立 Container

### b. 製作並推送 Container 至容器登錄

```
cd; cd your_project
```

```
cd; cd trees
```

```
az acr build --registry your_registry --image your_container:your_tag .
```

```
az acr build --registry trees --image trees17:0.0.0 .
```

# LINE Bot & WSGI

## 3. 建立 Container

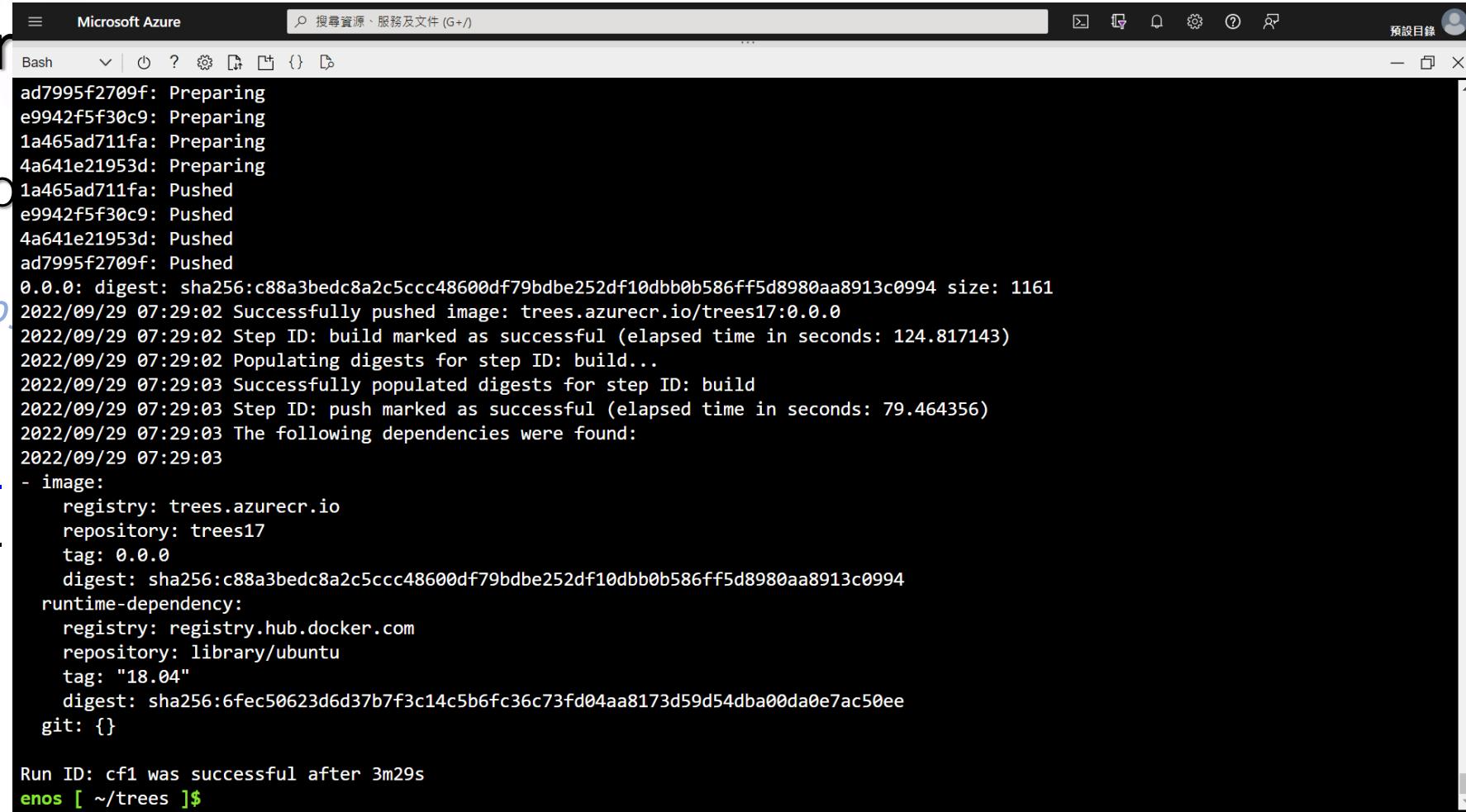
### b. 製作並推送 Container

```
cd; cd your_project_name
```

```
cd; cd trees
```

```
az acr build --
```

```
az acr build --
```



```
ad7995f2709f: Preparing
e9942f5f30c9: Preparing
1a465ad711fa: Preparing
4a641e21953d: Preparing
1a465ad711fa: Pushed
e9942f5f30c9: Pushed
4a641e21953d: Pushed
ad7995f2709f: Pushed
0.0.0: digest: sha256:c88a3bedc8a2c5ccc48600df79bdbe252df10dbb0b586ff5d8980aa8913c0994 size: 1161
2022/09/29 07:29:02 Successfully pushed image: trees.azurecr.io/trees17:0.0.0
2022/09/29 07:29:02 Step ID: build marked as successful (elapsed time in seconds: 124.817143)
2022/09/29 07:29:02 Populating digests for step ID: build...
2022/09/29 07:29:03 Successfully populated digests for step ID: build
2022/09/29 07:29:03 Step ID: push marked as successful (elapsed time in seconds: 79.464356)
2022/09/29 07:29:03 The following dependencies were found:
2022/09/29 07:29:03
- image:
    registry: trees.azurecr.io
    repository: trees17
    tag: 0.0.0
    digest: sha256:c88a3bedc8a2c5ccc48600df79bdbe252df10dbb0b586ff5d8980aa8913c0994
    runtime-dependency:
        registry: registry.hub.docker.com
        repository: library/ubuntu
        tag: "18.04"
        digest: sha256:6fec50623d6d37b7f3c14c5b6fc36c73fd04aa8173d59d54dba00da0e7ac50ee
        git: {}

Run ID: cf1 was successful after 3m29s
enos [ ~/trees ]$
```

# LINE Bot & WSGI

## 4. 部署 ACI

### a. 建立容器執行個體

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons. Below the navigation bar, the main content area has a title 'Azure 服務' (Azure Services) and several service icons: '建立資源' (Create Resource), '虛擬機器' (Virtual Machines), '容器登錄' (Container Registry), '容器執行個體' (Container Instances), and '容器執行個體' (Container Instances) again, which is highlighted with a pink box and has a '建立' (Create) button with a pink border. To the right of this central area are icons for '金鑰保存庫' (Key Vault), '訂用帳戶' (Subscription), and '更多服務' (More Services). Below the central area, there's a section titled '資源' (Resources) with tabs for '最近' (Recent) and '我的最愛' (Favorites). A message says '最近未檢視任何資源' (Recently viewed no resources) with a '檢視所有資源' (View all resources) button. At the bottom, there are sections for '瀏覽' (Browse) with links to '訂用帳戶' (Subscription), '資源群組' (Resource Groups), '所有資源' (All Resources), and '儀表板' (Dashboard); and '工具' (Tools) with links to 'Microsoft Learn' (Microsoft Learn), 'Azure 監視器' (Azure Monitor), '適用於雲端的 Microsoft Defender' (Microsoft Defender for Cloud), and '成本管理' (Cost Management).

<https://portal.azure.com/#create/hub>

# LINE Bot & WSGI

## 4. 部署 ACI

### a. 建立容器執行個體

The screenshot shows the Microsoft Azure portal with the title '建立容器執行個體' (Create Container Instance). The page is in Chinese. The 'Container details' section is visible, containing fields for account, resource group, container name, region, and image source. The 'Image source' section is expanded, showing options for 'Azure Container Registry' and 'Other registries'. The 'Container Registry' field is set to 'trees' and the 'Image' field is set to 'trees17'. The 'Next Step: Network' button is highlighted.

Microsoft Azure

搜尋資源、服務及文件 (G+/)

首頁 >

建立容器執行個體

執行個體 (ACI) 可讓您快速且輕鬆地在 Azure 上執行容器，而不須管理伺服器或學習新工具。ACI 提供以秒為單位的計費，讓在雲端上執行容器的成本降到最低。深入了解 Azure 容器執行個體

專案詳細資料

選取用以管理部署資源及成本的訂用帳戶。使用像資料夾這樣的資源群組來安排及管理您的所有資源。

訂用帳戶 \* ①

Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d1f3)

資源群組 \* ①

bot\_group\_09272303

容器詳細資料

容器名稱 \* ①

treeswsqi

區域 \* ①

(US) West US

可用性區域 ①

None

選取的區域不支援可用性區域。

映像來源 \* ①

快速入門映像

Azure Container Registry

其他登錄

登錄 \* ①

trees

映像 \* ①

trees17

檢閱 + 建立

< 上一步

下一步：網路 >

# LINE Bot & WSGI

## 4. 部署 ACI

### a. 建立容器執行個體

The screenshot shows the '建立容器執行個體' (Create Container Instance) step in the Azure portal. The top navigation bar includes 'Microsoft Azure', a search bar, and various icons. The main area has tabs for '檢閱 + 建立' (Review + Create), '< 上一步' (Previous Step), and '下一步：進階' (Next Step: Advanced). The configuration section for '網路' (Network) is displayed, showing options for network type (Public, Private, None), DNS name, and port mapping. A note states: '個體從三個網路選項中進行選擇' (Select from three network options). Below the note, it says: '• [公用] 則將為容器執行個體建立公用 IP 位址。' (If [Public] is selected, a public IP address will be assigned to the container instance.)

指定開放外部連入的 PORT，  
一併開通防火牆

# LINE Bot & WSGI

## 4. 部署 ACI

a. 建立容器執行個體

重新啟動原則 ①

失敗時

環境變數

標示為安全	金鑰	值
否	PORT	3000
否		

命令覆寫 ①

範例: [ "/bin/bash", "-c", "echo hello; sleep 100000" ]

金鑰管理 ①

Microsoft 管理的金鑰 (MMK)  
 客戶自控金鑰 (CMK)

檢閱 + 建立 < 上一步 下一步 : 標籤 >

以環境變數指定 Container 使用的 PORT

### Note

若監聽的 PORT 寫死於 container  
(未設定為環境變數) 則省略此步驟

# LINE Bot & WSGI

## 4. 部署 ACI

a. 建立容器執行個體

The screenshot shows the '建立容器執行個體' (Create Container Instance) step in the Azure portal. A green success message '驗證成功' (Validation successful) is displayed at the top left. The '檢閱 + 建立' (Review + Create) tab is selected. The '基本' (Basic) section displays the configuration details:

設定	值
訂用帳戶	Azure Pass - 贊助
資源群組	bot_group_09272303
區域	West US
容器名稱	treeswsgi
映像類型	Private
映像登錄登入伺服器	trees.azurecr.io
映像	trees.azurecr.io/trees17:0.0.0
映像登錄使用者名稱	trees
OS 類型	Linux
記憶體 (GiB)	1.5
CPU 核心數	1
GPU 類型 (預覽)	None
GPU 計數	0

The '網路' (Network) section shows:

設定	值
網路類型	公用
連接埠	3000 (TCP)
重複使用 DNS 名稱標籤範圍	任何重複使用 (不安全)

The '進階' (Advanced) section is partially visible.

At the bottom, there are buttons: '建立' (Create) in blue, '< 上一步' (Previous Step), '下一步 >', and '下載自動化的範本' (Download template).

# LINE Bot & WSGI

## 4. 部署 ACI

### a. 建立容器執行個體

The screenshot shows the Microsoft Azure Container Instances overview page for the deployment 'Microsoft.ContainerInstances-20220929154101'. The deployment status is marked as '已完成' (Completed) with a green checkmark icon. Deployment details include:

- 部署名稱: Microsoft.ContainerInstances-20220929154101
- 訂用帳戶: Azure Pass - 賽助 (0fdd0a2d-5afb-4589-8206-0a7eae00d...)
- 開始時間: 29/9/2022 下午3:51:14
- 相互關聯識別碼: b2cd1382-3237-4bad-bb9d-7b027981f509
- 資源群組: bot\_group\_09272303

Below the deployment details, there are two collapsed sections: '部署詳細資料' (Deployment Details) and '後續步驟' (Next Steps). A prominent blue button labeled '前往資源' (Go to Resource) is centered below these sections. On the right side of the page, there are several promotional cards:

- 成本管理**: 接收通知以掌握預算，並避免帳單上出現非預期的費用。 [設定成本警示 >](#)
- 適用於雲端的 Microsoft Defender**: 保護應用程式及基礎結構。 [移至適用於雲端的 Microsoft Defender >](#)
- 免費 Microsoft 教學課程**: 立即開始學習。 [立即開始學習 >](#)
- 諮詢專家**: Azure 專家是服務提供者合作夥伴，可協助您在 Azure 上管理資產，也是您的第一線支援。 [尋找 Azure 專家 >](#)

# LINE Bot & WSGI

## 4. 部署 ACI

### a. 建立容器執行個體

The screenshot shows the Microsoft Azure Container Instances overview page for a container named 'treeswsgi'. The container is currently running in the 'bot\_group\_09272303' resource group in the West US location. It is using an Azure Pass - 賽助 account and has a public IP address of 20.237.248.210. The container is running on a Linux OS. A pink annotation on the right side of the screen highlights the public IP address with the text 'Reverse Proxy 轉導 IP'.

Microsoft Azure

首頁 > Microsoft.ContainerInstances-20220929154101 | 概觀 >

treeswsgi 容器執行個體

搜尋

開始 重新啟動 停止 刪除 重新整理

程式集

資源群組 (移動) : bot\_group\_09272303  
狀態 : 正在執行  
位置 : West US  
訂用帳戶 (移動) : Azure Pass - 賽助  
訂用帳戶識別碼 : 0fdd0a2d-5afb-4589-8206-0a7eae00d1f3  
標籤 (編輯) : 按一下這裡即可新增標籤

OS 類型 : Linux  
IP 位址 (Public) : 20.237.248.210  
FQDN  
容器計數

Reverse Proxy 轉導 IP

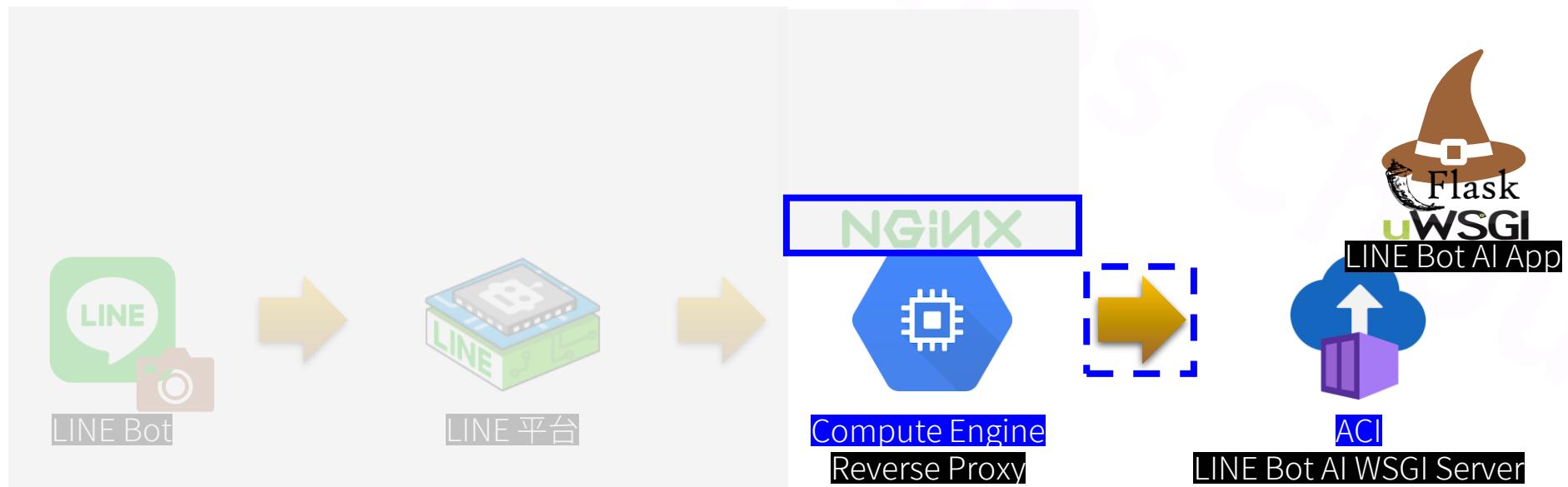
CPU

記憶體

接收的網路位元組

傳輸的網路位元組

JSON 檢視



# 網站與憑證

## 1. 調整 NGINX 轉導

### a. 調整 NGINX 組態

- ① 以 SSH 登入 NGINX VM
- ② 以文字編輯器開啟 NGINX 組態檔 `/etc/nginx/sites-enabled/your_project.conf`
- ③ 調整轉導設定
- ④ 儲存並退出

```
server {  
    server_name your_domain;  
  
    location / {  
        include uwsgi_params;  
        #uwsgi_pass your_ip:your_port;  
        proxy_pass http://your_ip:your_port;  
    }  
}
```

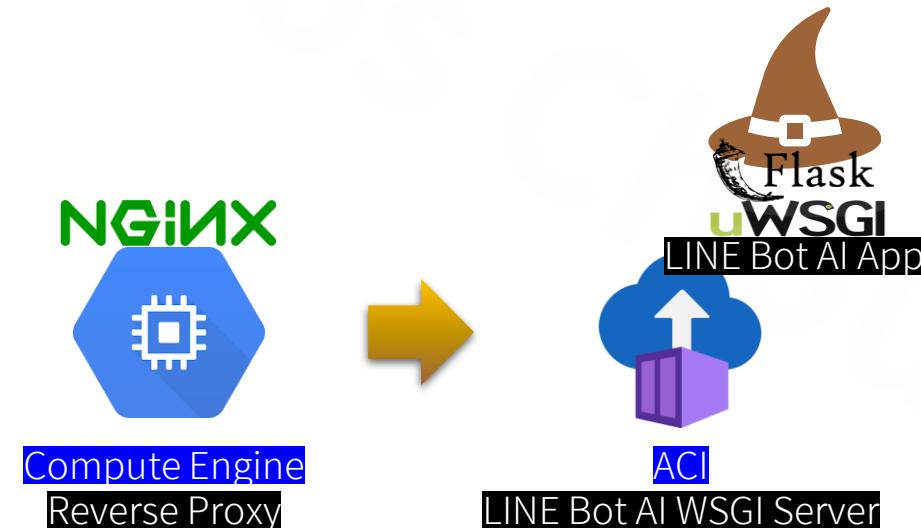
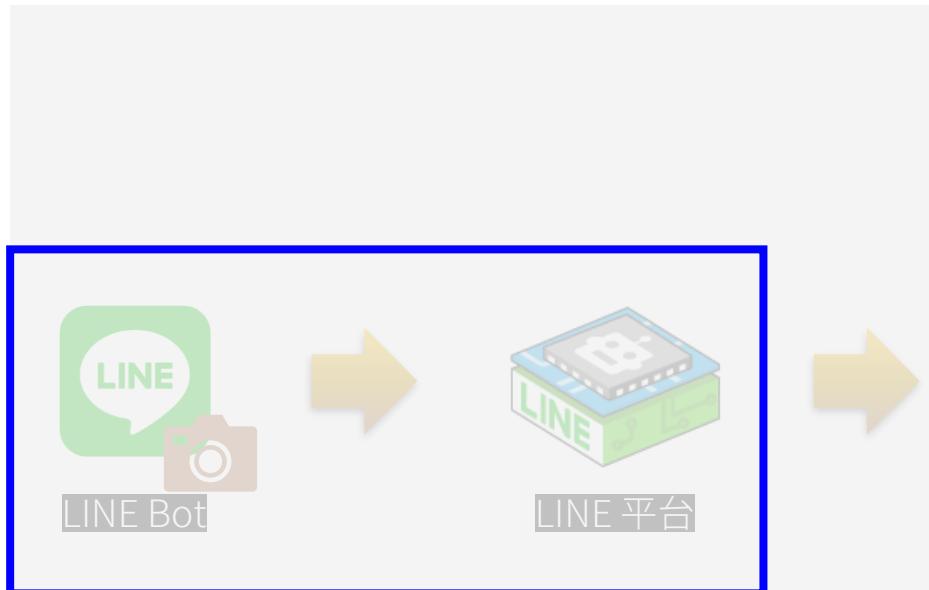
# 網站與憑證

## 1. 調整 NGINX 轉導

### b. 更新 NGINX 組態

- ① (以 SSH 登入 NGINX VM)
- ② 重新載入 NGINX 組態檔

```
sudo nginx -s reload
```

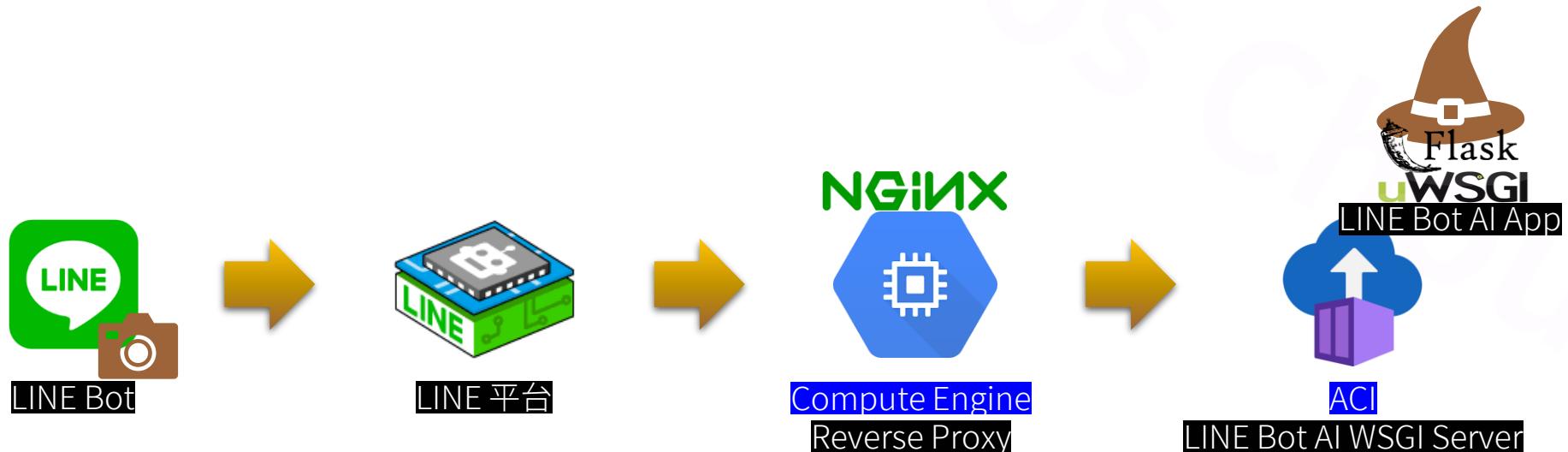


# 設定 LINE Messaging

## 1. 調整 LINE Messaging

a. 調整 Webhook URL

b. LINE 測試



# Debug

## 1. 確認 LINE Bot 運作

### a. 檢視容器執行個體 log

- ① 進入容器執行個體
- ② 容器
- ③ 記錄
- ④ 重新整理
- ⑤ 觀察 log

The screenshot shows the Azure Container Instances portal. On the left, there's a navigation pane with '容器執行個體' (Container Instances) selected. In the center, a detailed view for the 'treeswsgi' container is shown. The top right corner has a search bar and various navigation icons. The main area displays the container's status: '1 個容器及 0 個 Init 容器'. Below this, a table lists the container details: name 'treeswsgi', image 'trees.azurecr.io/trees17...', state 'Running', start time '2022-09-29T07:53:45.82...', and restart count '0'. On the right, the '記錄' (Logs) tab is selected, showing the log output for the container. The log output is as follows:

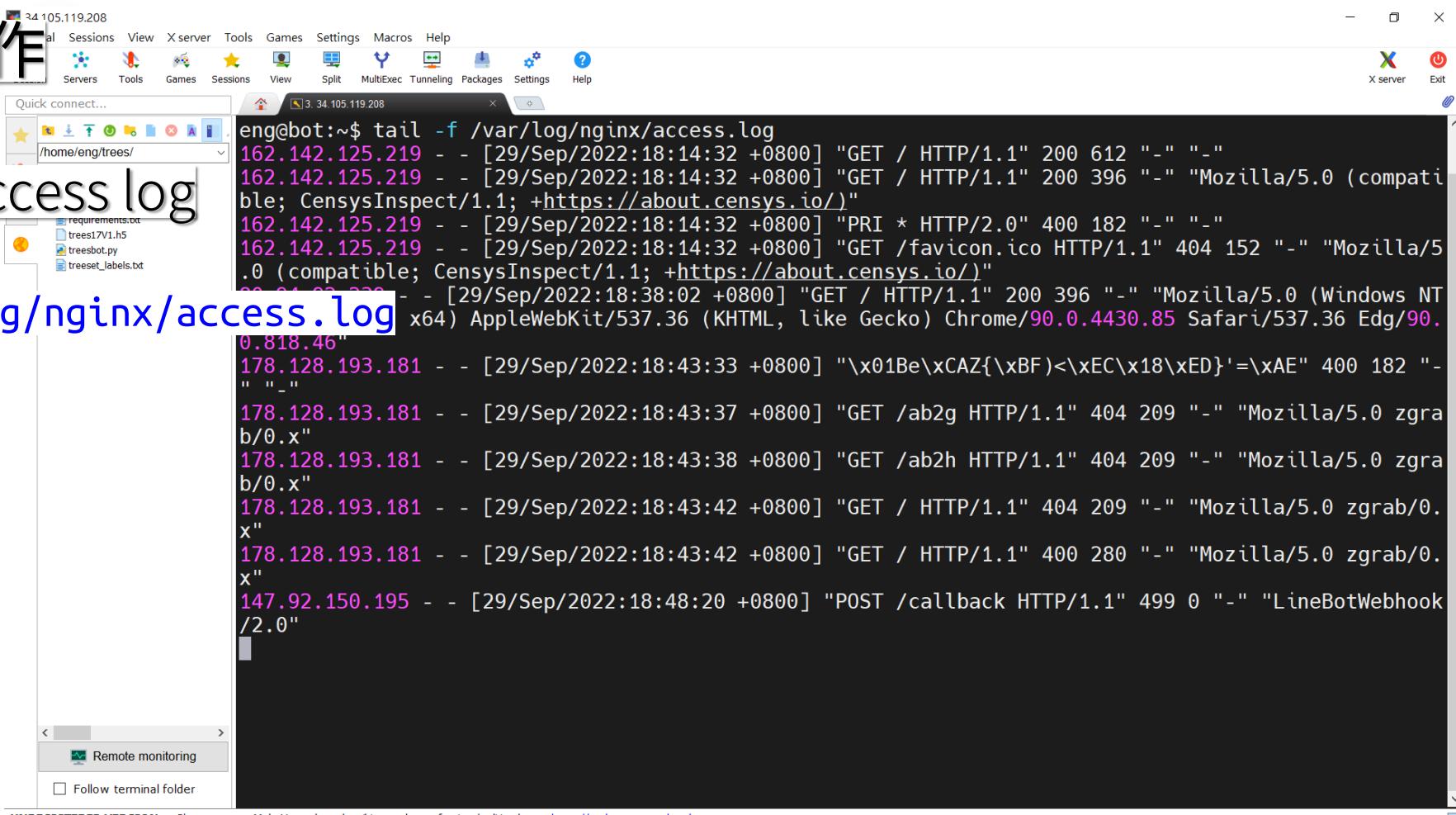
```
*** Starting uWSGI 2.0.20 (64bit) on [Thu Sep 29 15:53:46 2022] ***
compiled with version: 7.5.0 on 29 September 2022 07:26:47
os: Linux-5.10.102.2-microsoft-standard #1 SMP Mon Mar 7 17:36:34 UTC 2022
nodename: SandboxHost-638000346905459478
machine: x86_64
clock source: unix
detected number of CPU cores: 1
current working directory: /treesbot
detected binary path: /usr/local/bin/uwsgi
!!! no internal routing support, rebuild with pcre support !!!
uwsgi running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uwsgi as root !!! (use the --uid flag) ***
*** WARNING: you are running uwsgi without its master process manager ***
your processes number limit is 5932
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uwsgi socket 0 bound to TCP address :3000 fd 3
uwsgi running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uwsgi as root !!! (use the --uid flag) ***
Python version: 3.6.9 (default, Jun 29 2022, 11:45:57) [GCC 8.4.0]
*** Python threads support is disabled. You can enable it with --enable-threads ***
```

# Debug

## 2. 確認 NGINX 運作

### a. 檢視 NGINX access log

```
tail -f /var/log/nginx/access.log
```



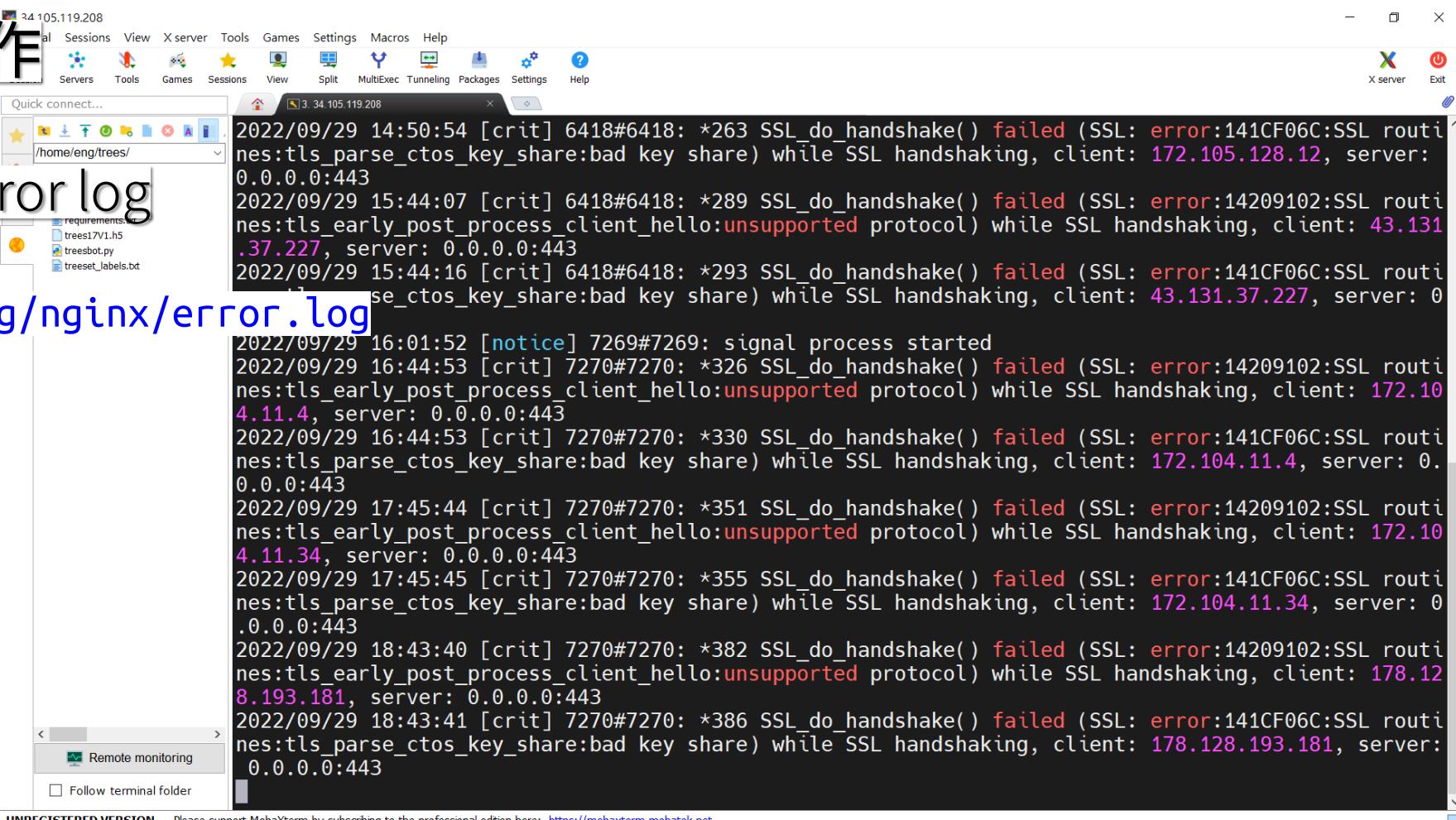
```
eng@bot:~$ tail -f /var/log/nginx/access.log
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 612 "-" "-"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "PRI * HTTP/2.0" 400 182 "-"-
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET /favicon.ico HTTP/1.1" 404 152 "-" "Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)"
20.21.82.220 - - [29/Sep/2022:18:38:02 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (Windows NT x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46"
178.128.193.181 - - [29/Sep/2022:18:43:33 +0800] "\x01Be\xCAZ{\xBF}<\xEC\x18\xED}'=\xAE" 400 182 "-"-
178.128.193.181 - - [29/Sep/2022:18:43:37 +0800] "GET /ab2g HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:38 +0800] "GET /ab2h HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:42 +0800] "GET / HTTP/1.1" 404 209 "-" "Mozilla/5.0 zgrab/0.x"
178.128.193.181 - - [29/Sep/2022:18:43:42 +0800] "GET / HTTP/1.1" 400 280 "-" "Mozilla/5.0 zgrab/0.x"
147.92.150.195 - - [29/Sep/2022:18:48:20 +0800] "POST /callback HTTP/1.1" 499 0 "-" "LineBotWebhook/2.0"
```

# Debug

## 2. 確認 NGINX 運作

### b. 檢視 NGINX error log

```
tail -f /var/log/nginx/error.log
```

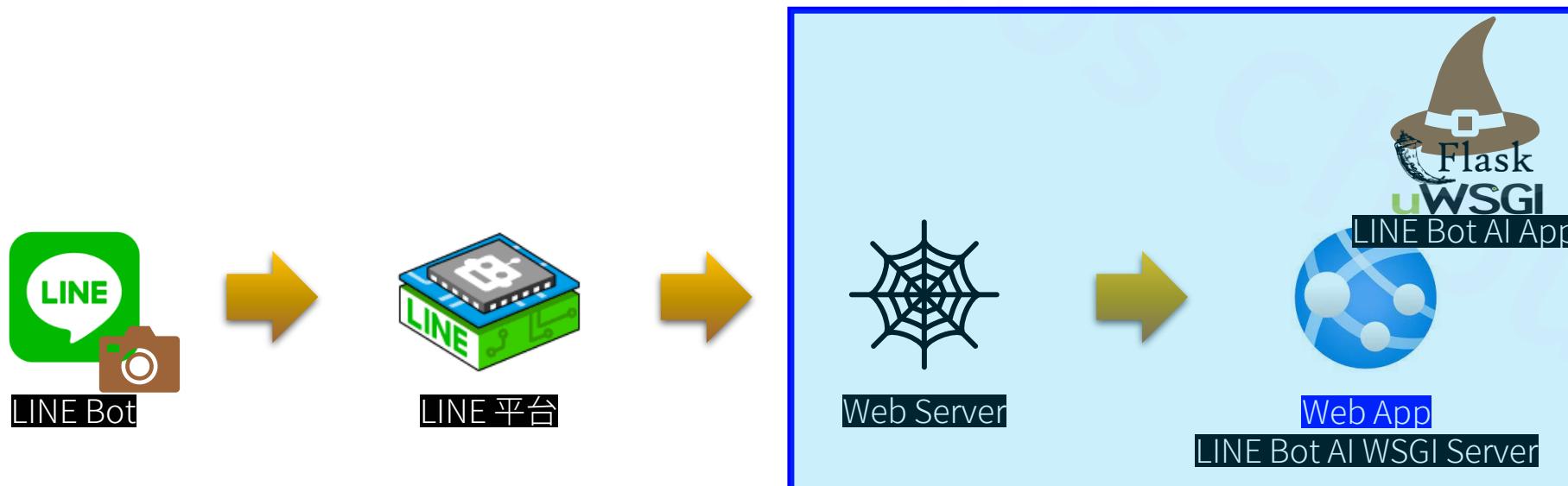


```
2022/09/29 14:50:54 [crit] 6418#6418: *263 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.105.128.12, server: 0.0.0.0:443
2022/09/29 15:44:07 [crit] 6418#6418: *289 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 43.131.37.227, server: 0.0.0.0:443
2022/09/29 15:44:16 [crit] 6418#6418: *293 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 43.131.37.227, server: 0.0.0.0:443
2022/09/29 16:01:52 [notice] 7269#7269: signal process started
2022/09/29 16:44:53 [crit] 7270#7270: *326 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 172.104.11.4, server: 0.0.0.0:443
2022/09/29 16:44:53 [crit] 7270#7270: *330 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.104.11.4, server: 0.0.0.0:443
2022/09/29 17:45:44 [crit] 7270#7270: *351 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 172.104.11.34, server: 0.0.0.0:443
2022/09/29 17:45:45 [crit] 7270#7270: *355 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.104.11.34, server: 0.0.0.0:443
2022/09/29 18:43:40 [crit] 7270#7270: *382 SSL_do_handshake() failed (SSL: error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported protocol) while SSL handshaking, client: 178.128.193.181, server: 0.0.0.0:443
2022/09/29 18:43:41 [crit] 7270#7270: *386 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 178.128.193.181, server: 0.0.0.0:443
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# Solution 3 - Single Container 部署

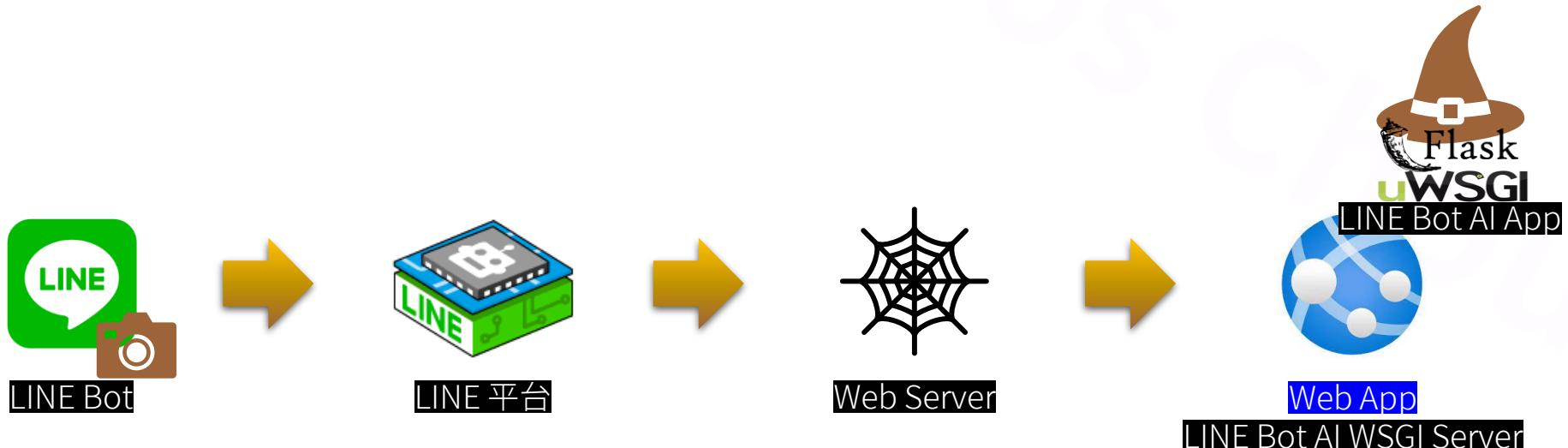
1. 長期可用且廉價的硬體環境 ... [Azure Web App](#)
2. Flask as Web Server 的替代方案 ... [Azure Web App](#)
3. 長期可用且廉價的 SSL 網域方案 ... [Azure Web App](#)

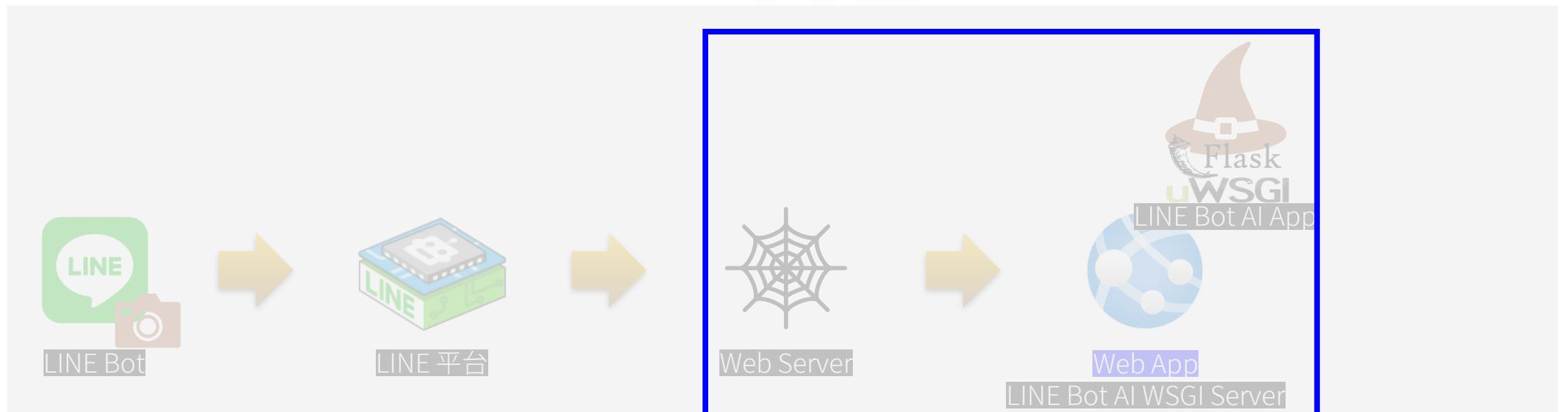


# 任務

1. 上傳程式至 Cloud Shell
2. 建立 Container 並推送至容器登錄
3. 部署 Container 至應用程式服務

# 流程





# LINE Bot & WSGI

## 1. 部署 Web App

### a. 由 Container 建立應用程式服務

The screenshot shows the Microsoft Azure Storage Explorer interface. On the left, there's a sidebar with navigation links like '首頁', '容器登錄', and 'trees | 存放庫'. The main area displays a storage account named 'trees' with a container named 'trees17'. The container details show it was last updated on 29/9/2022 at 3:29 PM (GMT+8). On the right, there's a detailed view of the container, including a '程式集' (Program) section with a single item 'trees17'. A context menu is open over this item, with the option '部署到 Web 應用程式' (Deploy to Web App) highlighted with a pink rectangle.

# LINE Bot & WSGI

## 1. 部署 Web App

The screenshot shows the Microsoft Azure portal interface for creating a new web application. On the left, there's a sidebar for 'trees17' showing a single container named 'trees17'. The main area is titled 'Web App for Containers' and contains the following fields:

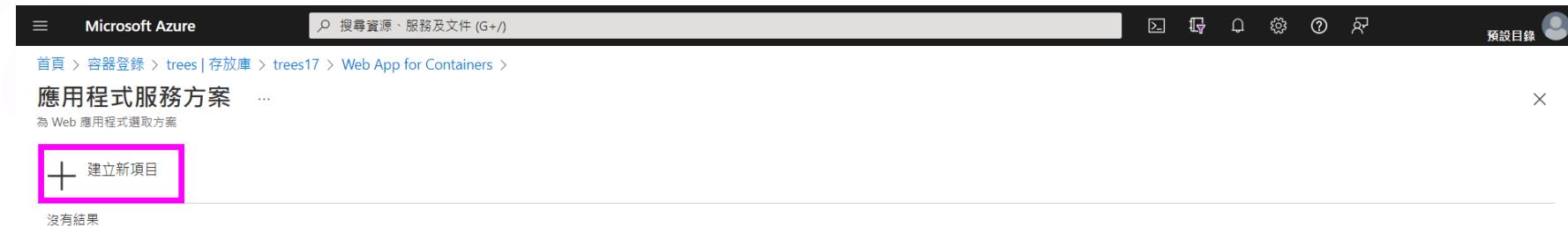
- 站台名稱 \*: treesapp (highlighted with a pink box)
- 訂用帳戶 \*: Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d1f3) (dropdown menu)
- 資源群組 \*: bot\_group\_09272303 (dropdown menu)
- App Service 方案/位置 \*: (button to 'Browse') (highlighted with a pink box)
- 樹狀圖: trees17:0.0.0
- 作業系統 \*: Linux (radio button selected)

A note at the bottom states: "這會建立名為 'treesapp150116' 的 Webhook 供持續部署之用。Webhook 將會在登錄的當前位置 westus 中建立。"

**a. 由 Container 建立應用程式服務**

# LINE Bot & WSGI

## 1. 部署 Web App



### a. 由 Container 建立應用程式服務

# LINE Bot & WSGI

## 1. 部署 Web App



# LINE Bot & WSGI

## 1. 部署 Web App

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar for 'trees17' which includes a '程式集' (Program) section with a 'trees17' entry, and a '存取庫' (Storage) section. The main area is titled 'Web App for Containers' and contains the following fields:

- 站台名稱 \*: treesapp
- 訂用帳戶 \*: Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d1f3)
- 資源群組 \*: bot\_group\_09272303
- App Service 方案/位置 \*: (new) treesol/West US
- 標籤計數: trees17:0.0.0
- 作業系統 \*: Linux (radio button selected)

A note at the bottom states: '這會建立名為 'treesapp150116' 的 Webhook 供持續部署之用。Webhook 將會在登錄的當前位置 westus 中建立。'

At the bottom center is a blue '建立' (Create) button.

**a. 由 Container 建立應用程式服務**

# LINE Bot & WSGI

## 1. 部署 Web App



The screenshot shows the Microsoft Azure Microsoft.Web | 概觀 (Overview) page. At the top, there is a search bar and a navigation bar with icons for search, refresh, and other account settings. Below the header, the title is "Microsoft.Web | 概觀". On the left, there is a sidebar with icons for Overview, Input, Output, and Templates. The main content area displays a green checkmark icon followed by the message "您的部署已完成" (Deployment completed). Below this, it shows deployment details: Deployment name: Microsoft.Web, Deployment account: Azure Pass - 贊助 (0fdd0a2d-5afb-4589-8206-0a7eae00d...), Start time: 29/9/2022 下午 11:05:28, Resource group: bot\_group\_09272303. To the right of the main content, there are three vertical cards: "成本管理" (Cost Management) with a green dollar sign icon, "適用於雲端的 Microsoft Defender" (Microsoft Defender for Cloud) with a shield icon, and "治詢專家" (Azure Experts) with a person icon.

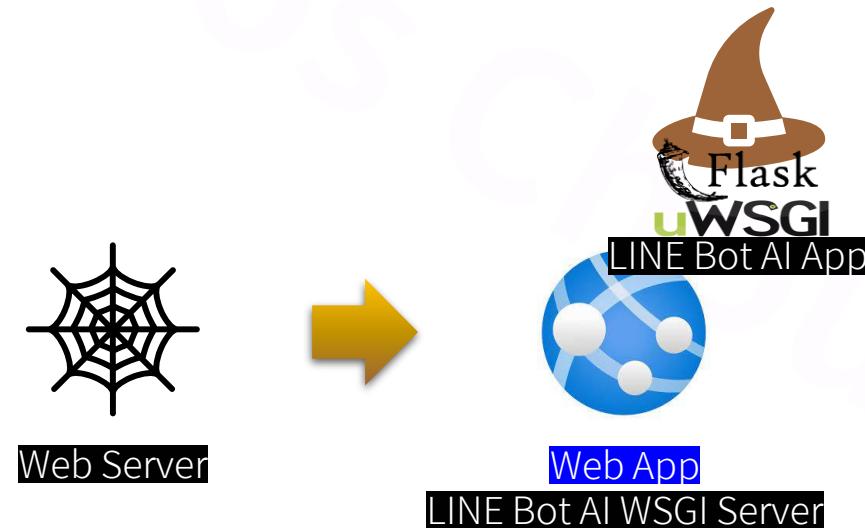
a. 由 Container 建立應用程式服務

# LINE Bot & WSGI

## 1. 部署 Web App

a. 由 Container 建立應用程式服務

The screenshot shows the Microsoft Azure portal interface for an App Service named 'treesapp'. The main title bar reads 'Microsoft Azure' and '搜尋資源、服務及文件 (G+ /)'. The left sidebar has a '概述' (Overview) section with links for '活動記錄' (Activity Log), '存取控制 (IAM)', and '標籤'. Below it is a '事件 (預覽)' (Preview Events) section with links for '部署', '快速入門', '部署認證', '部署位置', and '部署中心'. The main content area displays the 'treesapp' service details, including its status as 'Running' in 'West US'. A large pink callout box highlights the 'LINE Messaging Webhook URL' field, which contains the value <https://treesapp.azurewebsites.net>. Other service details shown include the URL (<https://treesapp.azurewebsites.net>), App Service 方案 ([treesal \(\\$1: 1\)](#)), and various connection and monitoring links. At the bottom, there are three charts: 'HTTP 5xx' (mostly 100s), '資料輸入' (mostly 100B), and '連出的資料' (mostly 100B).

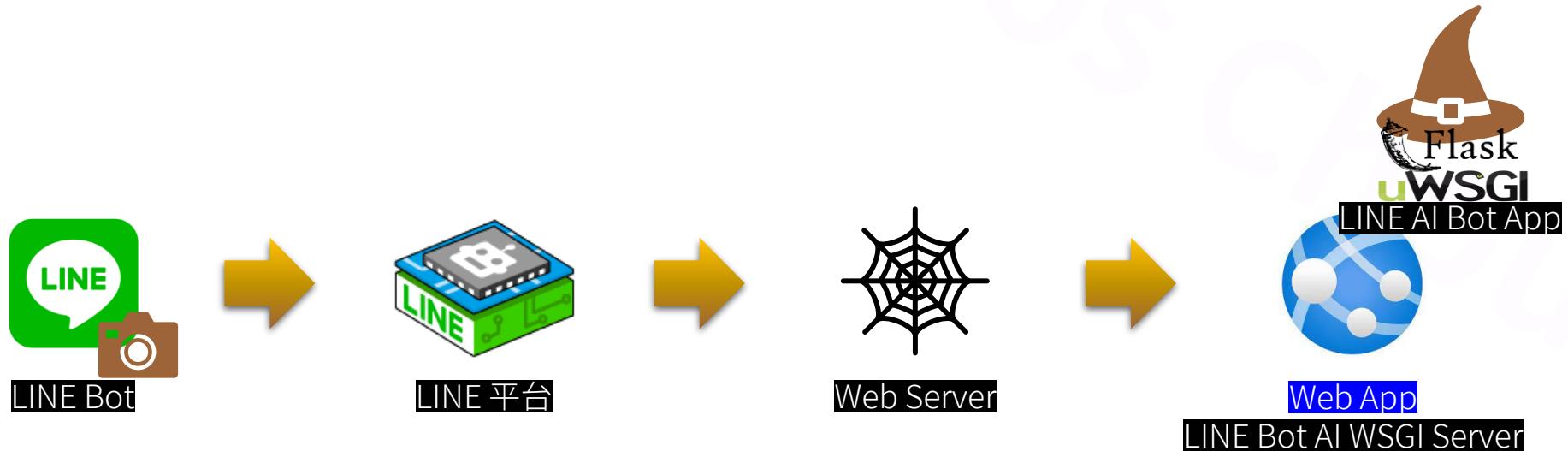


# 設定 LINE Messaging

## 1. 調整 LINE Messaging

a. 調整 Webhook URL

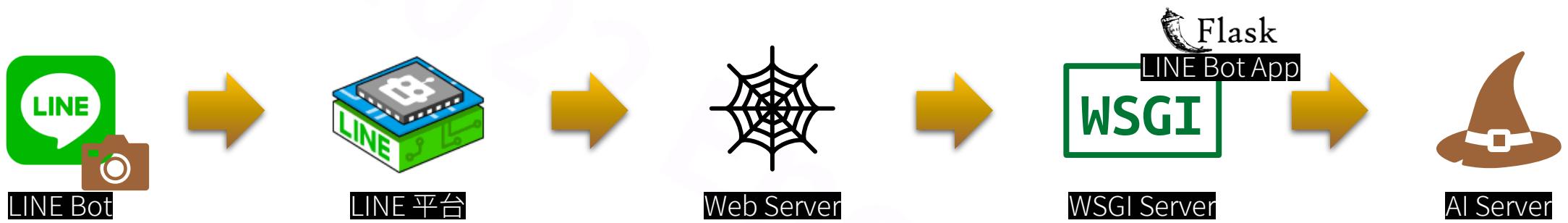
b. LINE 測試



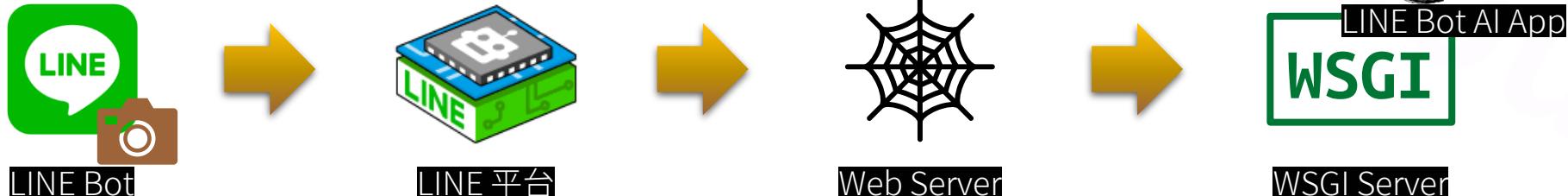
# LINE Bot + AI 部署

# Architecture

## Scalable Architecture



## Reasonable Architecture

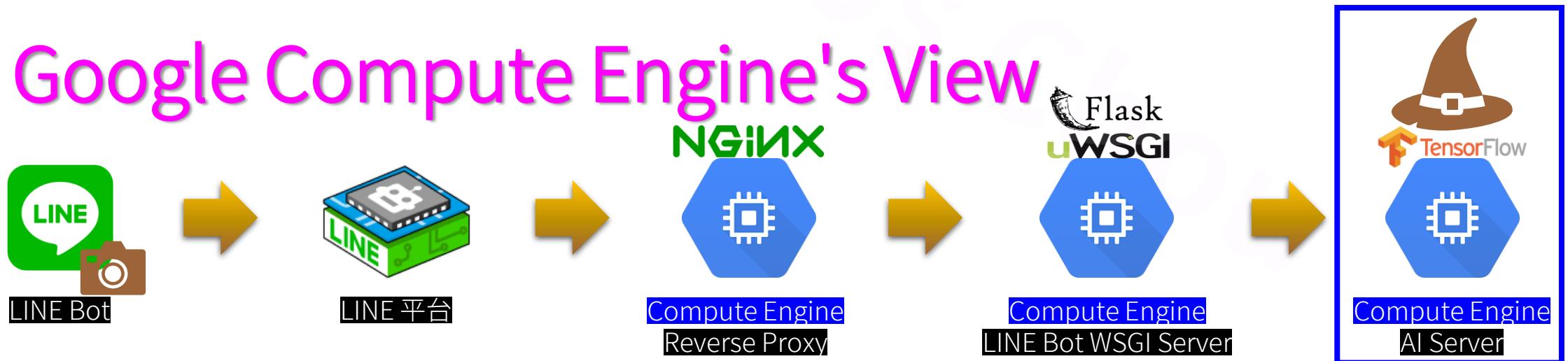


# 部署 TensorFlow Serving - VM

## Scalable Architecture

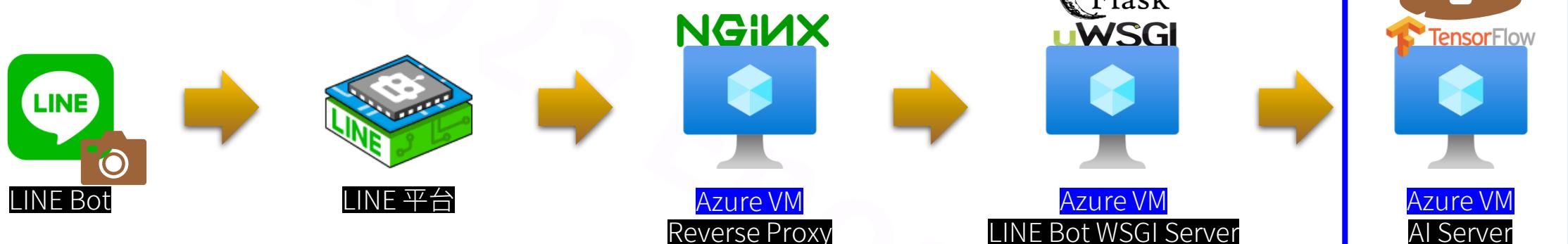


## Google Compute Engine's View

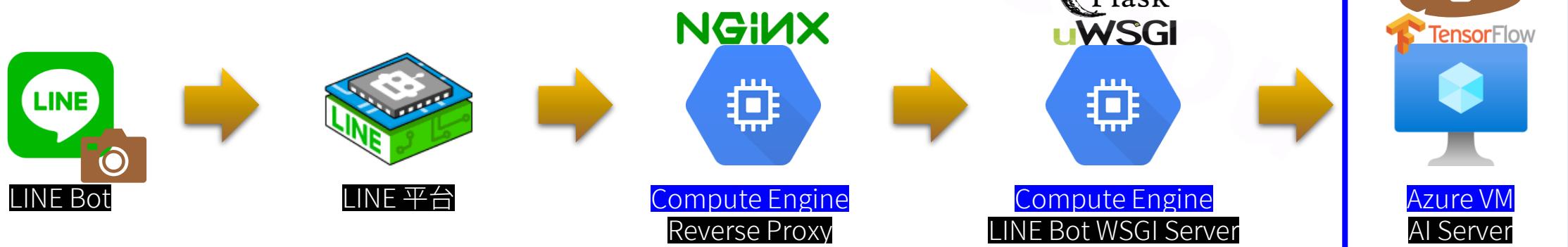


# 部署 TensorFlow Serving - VM

## Azure VM's View



## Mixed View



# 部署 TensorFlow Serving - VM

## 1. 建立 VM

### a. VM 規格

OS: Ubuntu 20.04

RAM: 1GB

HD: 10GB

# 部署 TensorFlow Serving - VM

## 1. 建立 VM

### b. 開通防火牆

PORT: TCP 8500, 8501

#### ① Google Compute Engine 開通防火牆

導覽選單 > 虛擬私有雲網路 > 防火牆 > 建立防火牆規則 >  
名稱："自行命名"；目標："網路中的所有執行個體"；來源 IPv4 範圍："0.0.0.0/0"；  
指定的通訊協定和埠："TCP" "8500, 8501" > 建立

#### ② Azure VM 開通防火牆

虛擬機器 > 進入指定 VM > 網路 > 新增輸入連接埠規則 >  
目的地連接埠範圍："TCP" "8500, 8501"；名稱："自行命名" > 新增

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### a. 轉換模型為 TensorFlow SavedModel 格式

- ① 準備 TensorFlow 2.4.4 環境

```
pip install tensorflow==2.4.4
```

#### Note

Python 3.9 以上版本可安裝 TensorFlow 2.5.3

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### a. 轉換模型為 TensorFlow SavedModel 格式

② 下載範例程式

`hdf5_to_savedmodel.py`  
`trees17V1.h5`

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### a. 轉換模型為 TensorFlow SavedModel 格式

③ 轉換模型

```
python hdf5_to_savedmodel.py your_hdf5.h5 your_savedmodel  
python hdf5_to_savedmodel.py trees17V1.h5 1
```

#### Note

若採用較新的 TensorFlow 版本導致 **protobuf error**，可加入環境變數  
PROTOCOL\_BUFFERS\_PYTHON\_IMPLEMENTATION=python 後重新執行  
(**export PROTOCOL\_BUFFERS\_PYTHON\_IMPLEMENTATION=python**)

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### b. 上傳 SavedModel 模型至 VM (from MobaXterm)

- ① 登入 VM
- ② 建立模型目錄
- ③ 上傳 SavedModel 至模型目錄

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### b. 上傳 SavedModel 模型至 VM

# 路徑範例

/home/your\_account/your\_model/1/assets

/home/your\_account/your\_model/1/saved\_model.pb

/home/your\_account/your\_model/1/variables/variables.data-00000-of-00001

/home/your\_account/your\_model/1/variables/variables.index

#### Note

- ① 模型放在 /home/someone/your\_model/2 ... N/ 之下，系統會選用數字最大的模型
- ② 切勿忽略 variables 目錄下檔案

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### b. 上傳 SavedModel 模型至 VM (from Cloud Shell/ macOS/ Linux)

- ① 登入終端機
- ② (上傳 VM 之 OpenSSH 格式私鑰 *your\_privatekey* 至終端機)
- ③ (調整私鑰權限)

`chmod 400 your_privatekey`

- ④ 建立 VM 模型目錄

```
ssh -i your_privatekey your_account@your_vmid mkdir your_model  
ssh -i ./eng eng@34.127.42.172 mkdir trees
```

- ⑤ 複製 SavedModel 至 VM

```
scp -i your_privatekey -r your_savedmodel your_account@your_vmid:your_model  
scp -i ./eng -r 1 eng@34.127.42.172:trees
```

# 部署 TensorFlow Serving - VM

## 3. 安裝 TensorFlow Serving

安裝最新版本 TensorFlow Server (v2.10.0)

```
echo "deb [arch=amd64] http://storage.googleapis.com/tensorflow-serving-  
apt stable tensorflow-model-server tensorflow-model-server-universal" |  
sudo tee /etc/apt/sources.list.d/tensorflow-serving.list; curl  
https://storage.googleapis.com/tensorflow-serving-apt/tensorflow-  
serving.release.pub.gpg | sudo apt-key add -; sudo apt-get update; sudo  
apt-get install tensorflow-model-server
```

Note

系統不須安裝其他 TensorFlow 套件

# 部署 TensorFlow Serving - VM

## 4. 啟動 TensorFlow Serving

啟動 TensorFlow Serving for RESTful API & gRPC

```
nohup tensorflow_model_server --model_base_path=your_model_path --  
model_name=your_model_name --rest_api_port=your_rest_port --  
port=your_grpc_port &  
nohup tensorflow_model_server --model_base_path=/home/eng/trees --  
model_name=trees --rest_api_port=8501 --port=8500 &
```

### Note

- ① --rest\_api\_port for RESTful API
- ② --port for gRPC
- ③ nohup ... & for 背景執行

# 部署 TensorFlow Serving - VM

## 5. 測試 TensorFlow Serving

### a. 建立測試環境

- ① 安裝 Python 套件

requests

pillow

numpy

tensorflow-serving-api # 欲避免自動更新 TensorFlow 須以 --no-deps 單獨安裝此套件

# 部署 TensorFlow Serving - VM

## 5. 測試 TensorFlow Serving

### a. 建立測試環境

② 下載範例程式

`tf-serving_test.py`

`model_io.py`

`treeset_labels.txt`

`trees samples`

# 部署 TensorFlow Serving - VM

## 5. 測試 TensorFlow Serving

### b. 測試 RESTful API

```
python tf-serving_test.py --help
usage: tf-serving_test.py [-h] [--protocol {gRPC,REST}] [--ssl] [--host HOST]
                          [--port PORT] [--labels LABELS] [--input INPUT]
                          [--output OUTPUT]
                          MODEL PIC
```

```
python tf-serving_test.py --host 34.168.152.9 --port 8501 --protocol REST --
                          labels treeset_labels.txt trees 凤凰木.jpg
```

# 部署 TensorFlow Serving - VM

## 5. 測試 TensorFlow Serving

### c. 測試 gRPC

- ① 確認 SavedModel I/O 名稱

```
python model_io.py your_savedmodel  
python model_io.py 1
```

#### Note

gRPC 需要 model I/O 名稱作為參數

# 部署 TensorFlow Serving - VM

## 5. 測試 TensorFlow Serving

### c. 測試 gRPC

```
python tf-serving-test.py --help
usage: tf-serving-test.py [-h] [--protocol {gRPC,REST}] [--ssl] [--host HOST]
                          [--port PORT] [--labels LABELS] [--input INPUT]
                          [--output OUTPUT]
                          MODEL PIC
```

```
python tf-serving-test.py --host 34.168.152.9 --port 8500 --protocol gRPC --
                          input input_4 --output dense_1 --labels treeset_labels.txt trees 鳳凰木.jpg
```

# 部署 TensorFlow Serving - VM

## 6. (部署多個 Model)

### a. 建立參數檔 *your\_conf*

```
model_config_list {  
    config {  
        name: 'your_model'  
        base_path: 'path/to/model'  
        model_platform: 'tensorflow'  
    }  
}
```

同時啟動兩個模型的參數檔範例 models.conf

```
model_config_list {  
    config {  
        name: 'trees'  
        base_path: '/home/eng/trees/'  
        model_platform: 'tensorflow'  
    }  
    config {  
        name: 'trees2'  
        base_path: '/home/eng/trees2/'  
        model_platform: 'tensorflow'  
    }  
}
```

# 部署 TensorFlow Serving - VM

## 6. (部署多個 Model)

### b. 啟動 TensorFlow Serving for RESTful API & gRPC

```
nohup tensorflow_model_server --model_config_file=your_conf --  
rest_api_port=your_rest_port --port=your_grpc_port &  
nohup tensorflow_model_server --model_config_file=models.conf --  
rest_api_port=8501 --port=8500 &
```

#### Note

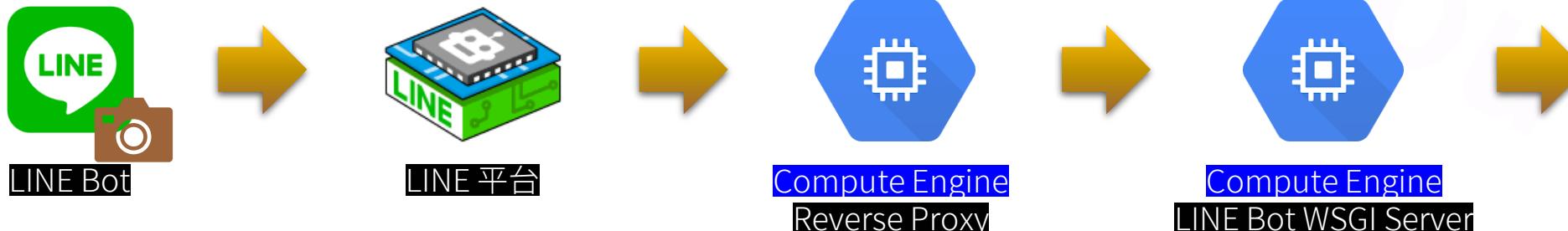
新增 `--model_config_file` 指定參數檔，取代 `--model_base_path` 與 `--model_name`

# 部署 TensorFlow Serving - Cloud Run

## Mixed View



## GCP's View



# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### a. 準備環境 (on GCP Cloud Shell)

- ① 開啟 GCP Cloud Shell
- ② 建立模型轉換虛擬環境

```
python -m venv your_env  
python -m venv test
```

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### a. 準備環境 (on GCP Cloud Shell)

③ 啟動環境

```
source your_env/bin/activate
```

```
source test/bin/activate
```

④ (退出環境)

```
deactivate
```

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### a. 準備環境 (on GCP Cloud Shell)

⑤ 安裝 TensorFlow 2.5.3

```
pip install tensorflow==2.5.3
```

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

b. 轉換模型為 SavedModel 格式 (on GCP Cloud Shell)

① 下載範例程式

```
git clone https://github.com/path_to_example
```

② 進入範例目錄

```
cd trees17
```

③ 轉換模型

```
python hdf5_to_savedmodel.py your_hdf5.h5 your_savedmodel
python hdf5_to_savedmodel.py trees17V1.h5 1
```

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### c. 將 SavedModel 置於模型目錄

# 目錄範例

*your\_model/1/assets*

*your\_model/1/saved\_model.pb*

*your\_model/1/variables/variables.data-00000-of-00001*

*your\_model/1/variables/variables.index*

#### Note

- ① 模型放在 *your\_model/2 ... N/* 之下，系統會選用數字最大的模型
- ② 切勿忽略 **variables** 目錄下檔案

# 部署 TensorFlow Serving - Cloud Run

## 2. 製作 Container

### a. 製作 Dockerfile (gRPC)

```
# Dockerfile 範例 for gRPC
FROM tensorflow/serving:2.8.3
COPY your_model/1/ /models/your_model/1
ENTRYPOINT tensorflow_model_server --port=$PORT --model_name=your_model
--model_base_path=/models/your_model
```

# 部署 TensorFlow Serving - Cloud Run

## 2. 製作 Container

### a. 製作 Dockerfile (RESTful API)

```
# Dockerfile 範例 for RESTful API
FROM tensorflow/serving:2.8.3
COPY your_model/1/ /models/your_model/1
ENTRYPOINT tensorflow_model_server --rest_api_port=$PORT --model_name=your_model
--model_base_path=/models/your_model
```

# 部署 TensorFlow Serving - Cloud Run

## 2. 製作 Container

### b. 將 Dockerfile 置於模型目錄同層級

```
# 目錄範例  
Dockerfile  
your_model/1/assets  
your_model/1/saved_model.pb  
your_model/1/variables/variables.data-00000-of-00001  
your_model/1/variables/variables.index
```

# 部署 TensorFlow Serving - Cloud Run

## 3. 部署

### a. 以 Google Cloud Shell 編輯器部署

- ① (上傳 *your\_model* 目錄與 Dockerfile 至 GCP 專案帳號之 Cloud Shell)
- ② 開啟 Cloud Shell 編輯器，指定 Workspace 為 Dockerfile 所在目錄，包含 trees 目錄
- ③ 以 Cloud Shell 編輯器部署 Cloud Run

Cloud Shell 編輯器 > Coud Code - Cloud Run: Cloud Run >  
(Enable Cloud Run API) > (Login to Google Cloud SDK) >  
(授權 Cloud Shell : 授權) > (Select a project) > Deploy to Cloud Run >  
Create a service > Service name : "自行命名" ; Region : "自行指定" >  
Show Advanced Settings > Memory allocated : "**512**" MiB > Deploy

# 部署 TensorFlow Serving - Cloud Run

## 3. 部署

### b. 以 Google Cloud Shell + Cloud Run 部署

- ① (上傳 *your\_model* 目錄與 *Dockerfile* 至 GCP 專案帳號之 Cloud Shell)
- ② `docker build -t gcr.io/your_project/your_container:your_tag .`
- ③ `docker push gcr.io/your_project/your_container:your_tag`
- ④ 由 Cloud Run 部署

Cloud Run > 建立服務 > 容器映像檔網址："選擇先前推送的 Container"；  
服務名稱："自行命名"；地區："自行指定"；驗證："允許未經驗證的叫用" >  
(容器、變數和密鑰、連線、安全性 > 連線 > 使用 HTTP/2 端對端) > 建立

#### Note

- ① Cloud Run 記憶體需 512M 以上；gRPC 建議採用 HTTP/2
- ② Client 連結時，HOST 須移除 "https://"；PORT 為 443，採用 SSL

# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

### a. 驗證網域



The screenshot shows the Google Cloud Cloud Run service management interface. The top navigation bar includes the Google Cloud logo, a project dropdown set to 'test', a search bar containing '搜尋 container', and various icons for notifications and help. The main navigation tabs are 'Cloud Run' (selected), '服務' (Services), '+ 建立服務' (+ Create Service), and '管理自訂網域' (Manage Custom Domains), which is highlighted with a pink rectangle. Below the tabs, there are three buttons: '服務' (Services), '工作' (Work), and '預覽' (Preview), with '服務' being the active tab. A filter section labeled '篩選' (Filter) is present. The main table lists a single service entry:

名稱	每秒要求數	區域	驗證	輸入流量	最近部署時間	部署者
treestfs	0	us-west1	允許在未經驗證的情況下叫用	全部	1 分鐘前	@gmail.com

# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

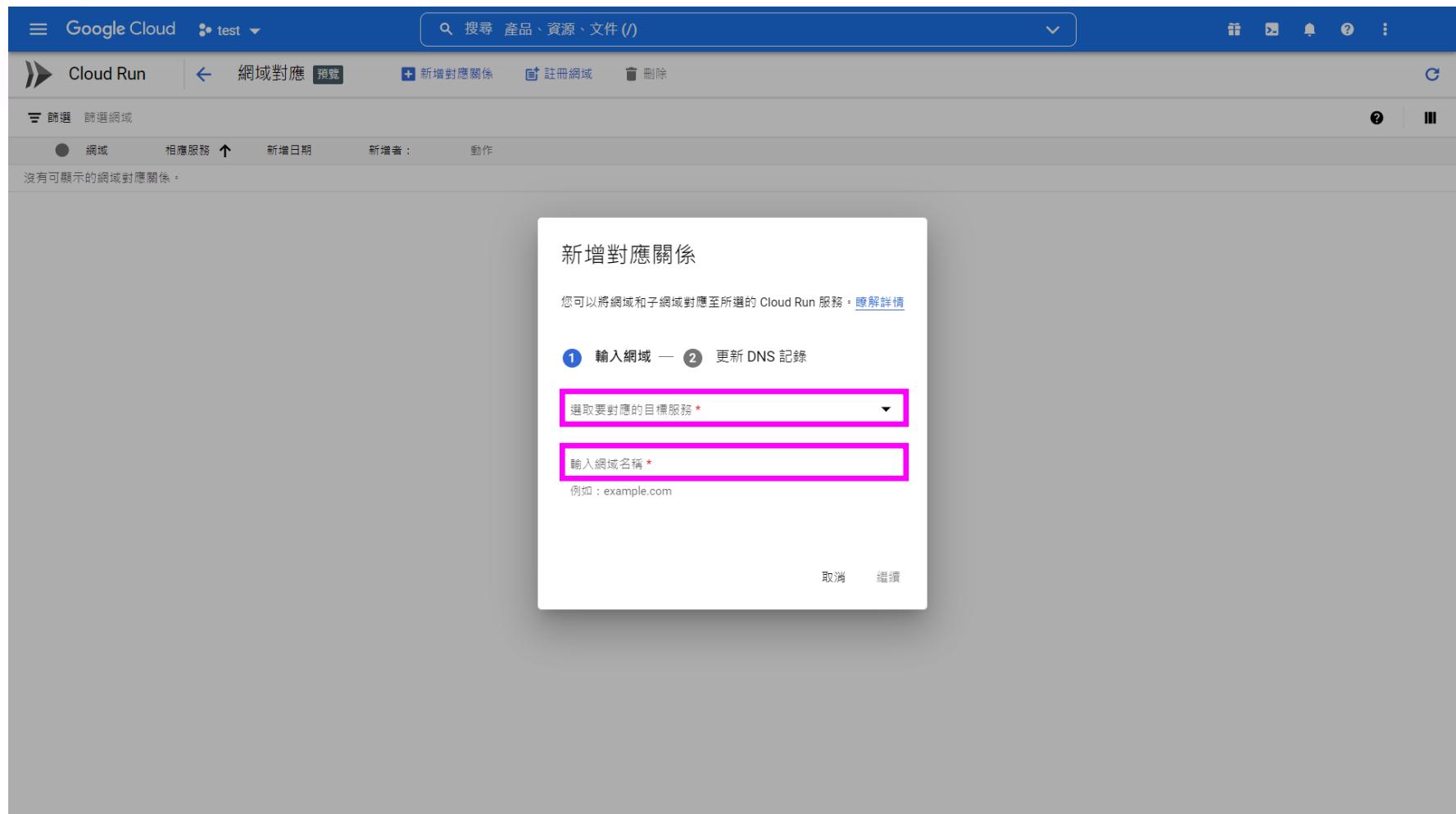
### a. 驗證網域

The screenshot shows the Google Cloud Platform interface for managing domain mappings. The top navigation bar includes the Google Cloud logo, a project dropdown set to 'test', a search bar, and various icons. Below the bar, the 'Cloud Run' service is selected. The main area is titled '網域對應' (Domain Mapping) with a '預覽' (Preview) button. A prominent blue button labeled '+ 新增對應關係' (Add Mapping Relationship) is highlighted with a pink rectangle. Below this, there's a table header with columns: 篩選 (Filter), 網域 (Domain), 相應服務 (Service), 新增日期 (Created Date), 新增者 (Creator), and 動作 (Actions). The message '沒有可顯示的網域對應關係。' (No domain mappings to display) is displayed.

# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換 Domain)

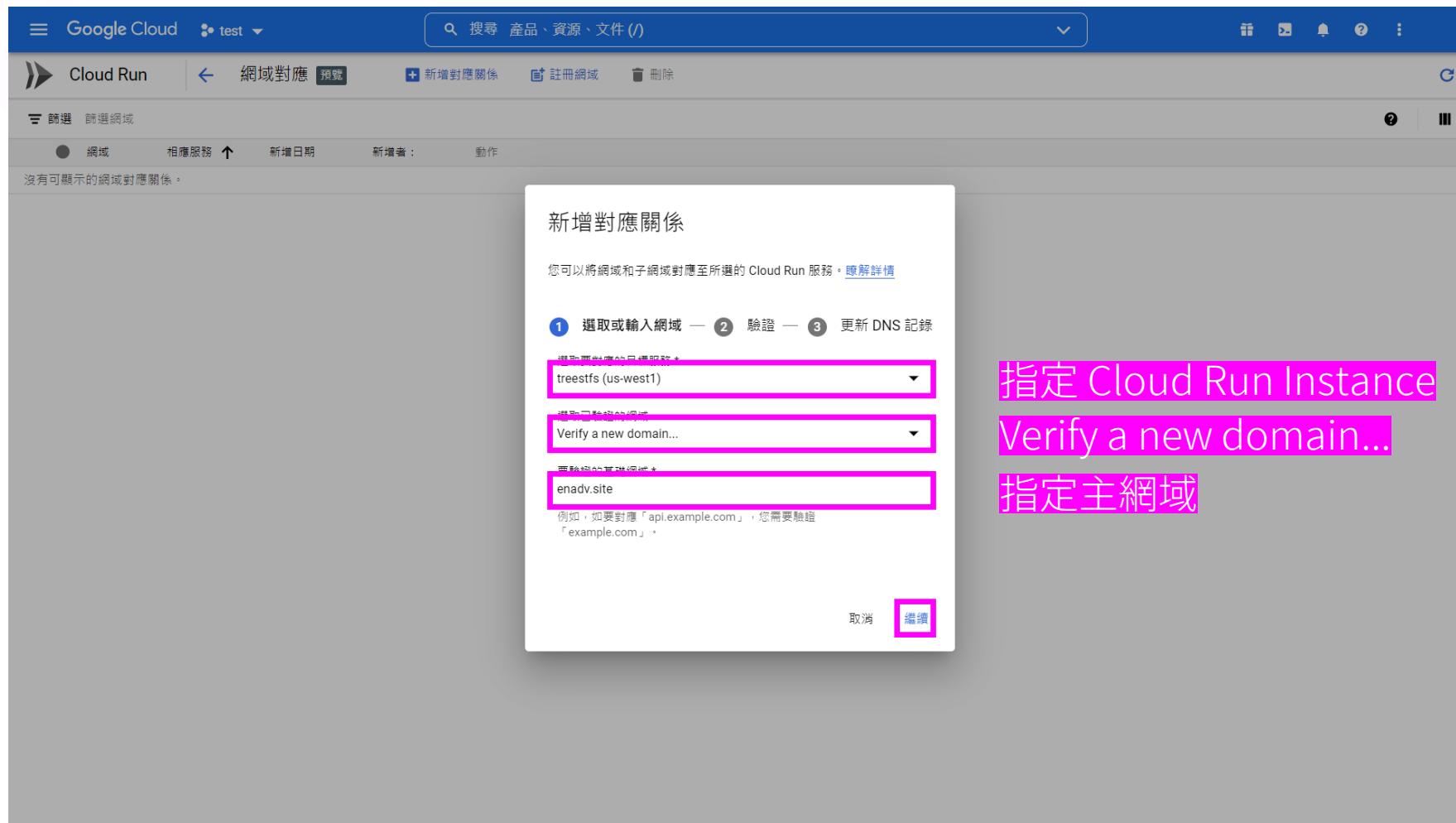
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

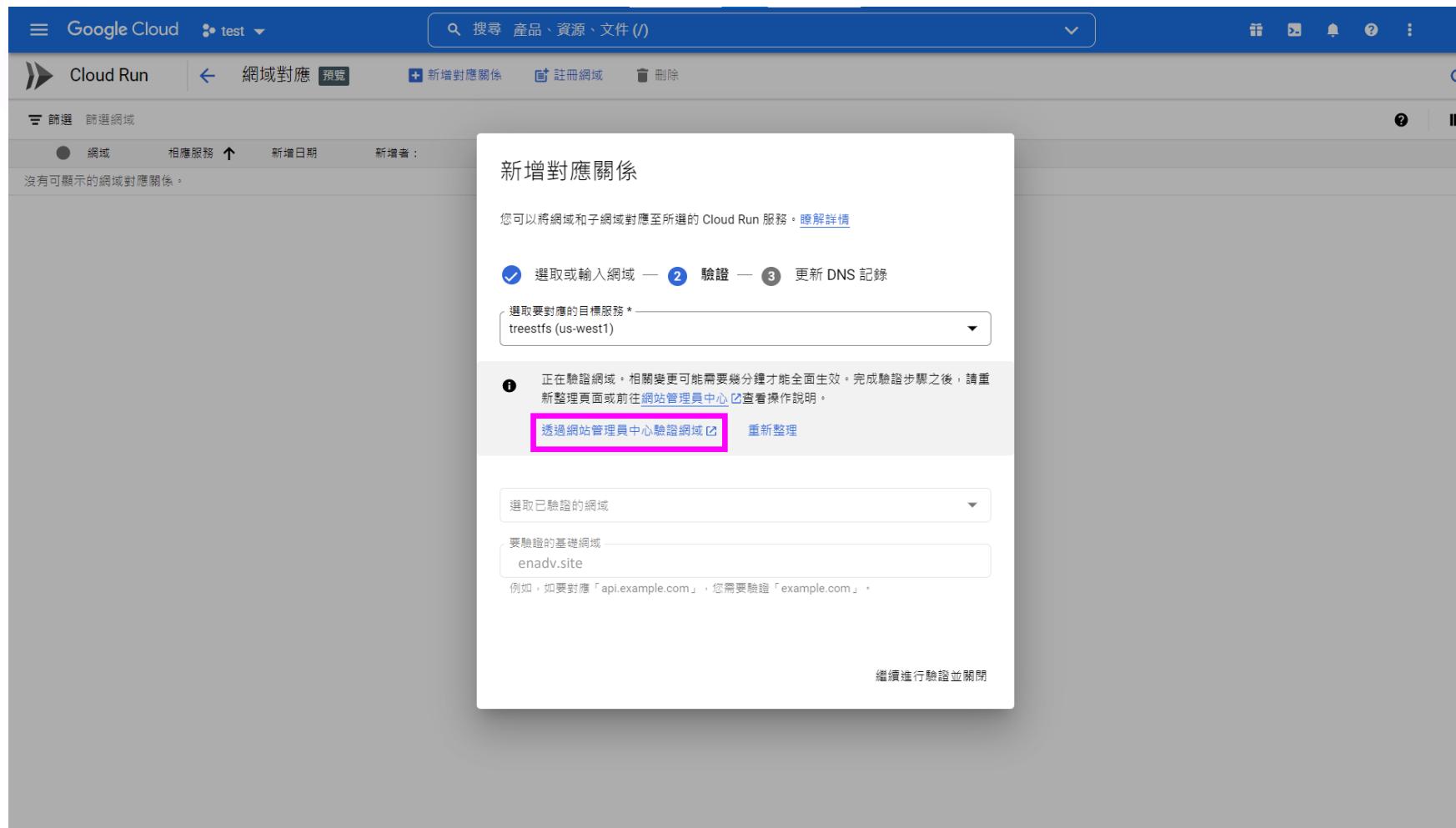
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

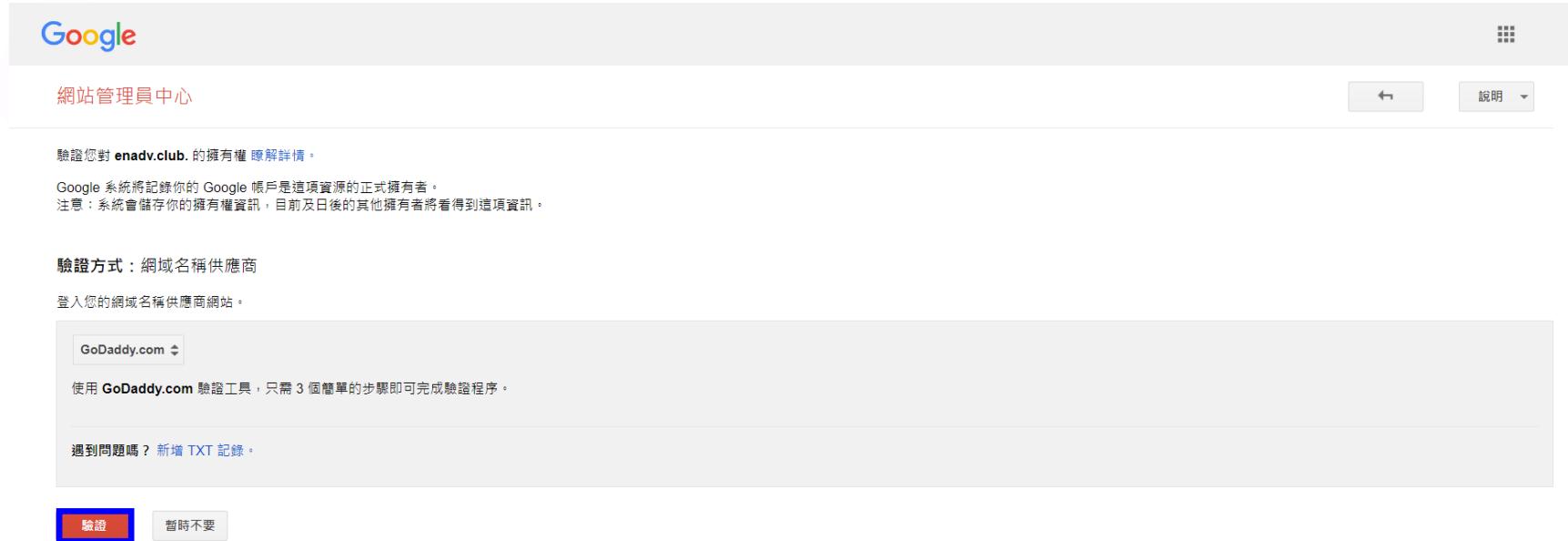
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

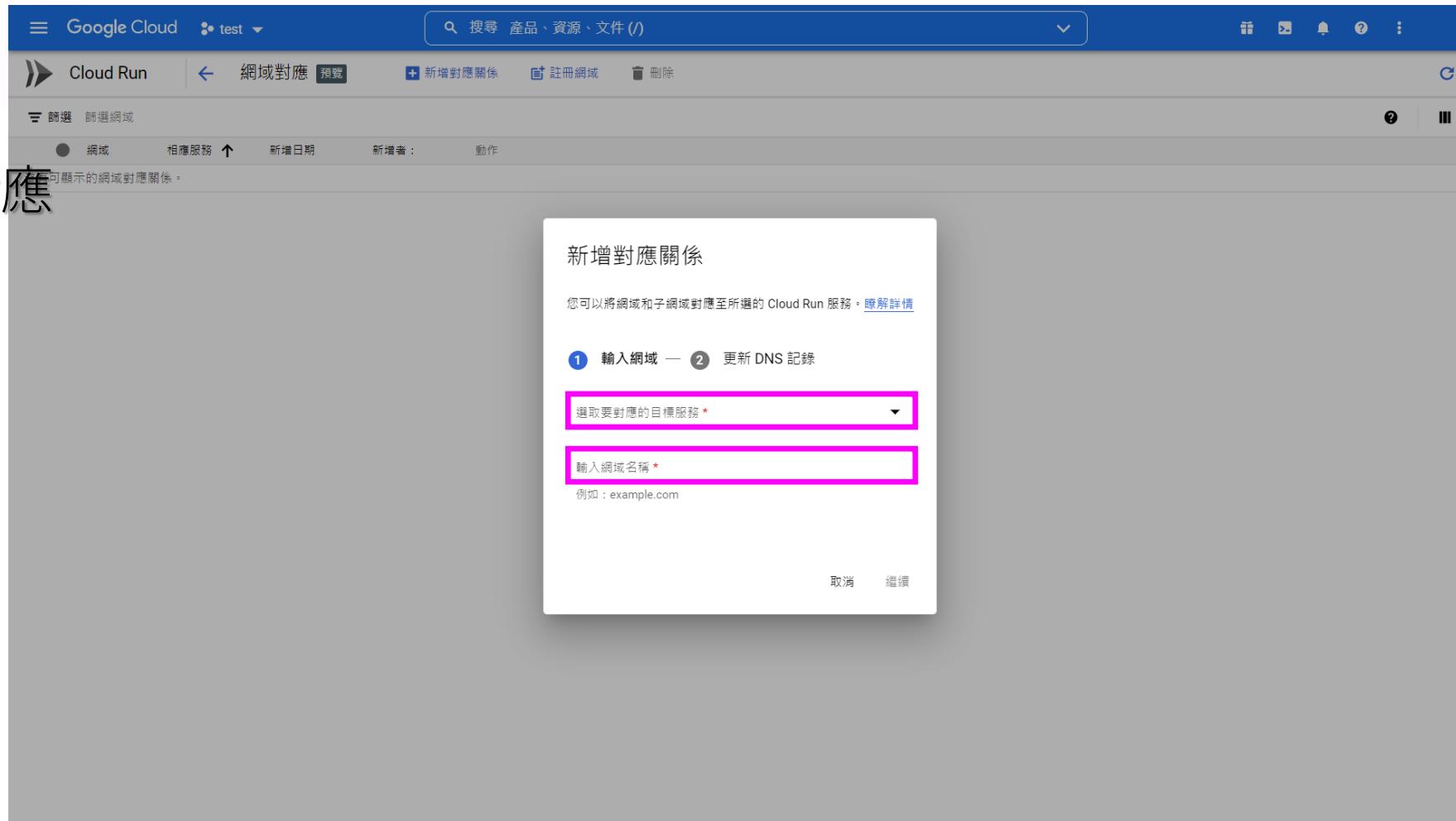
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

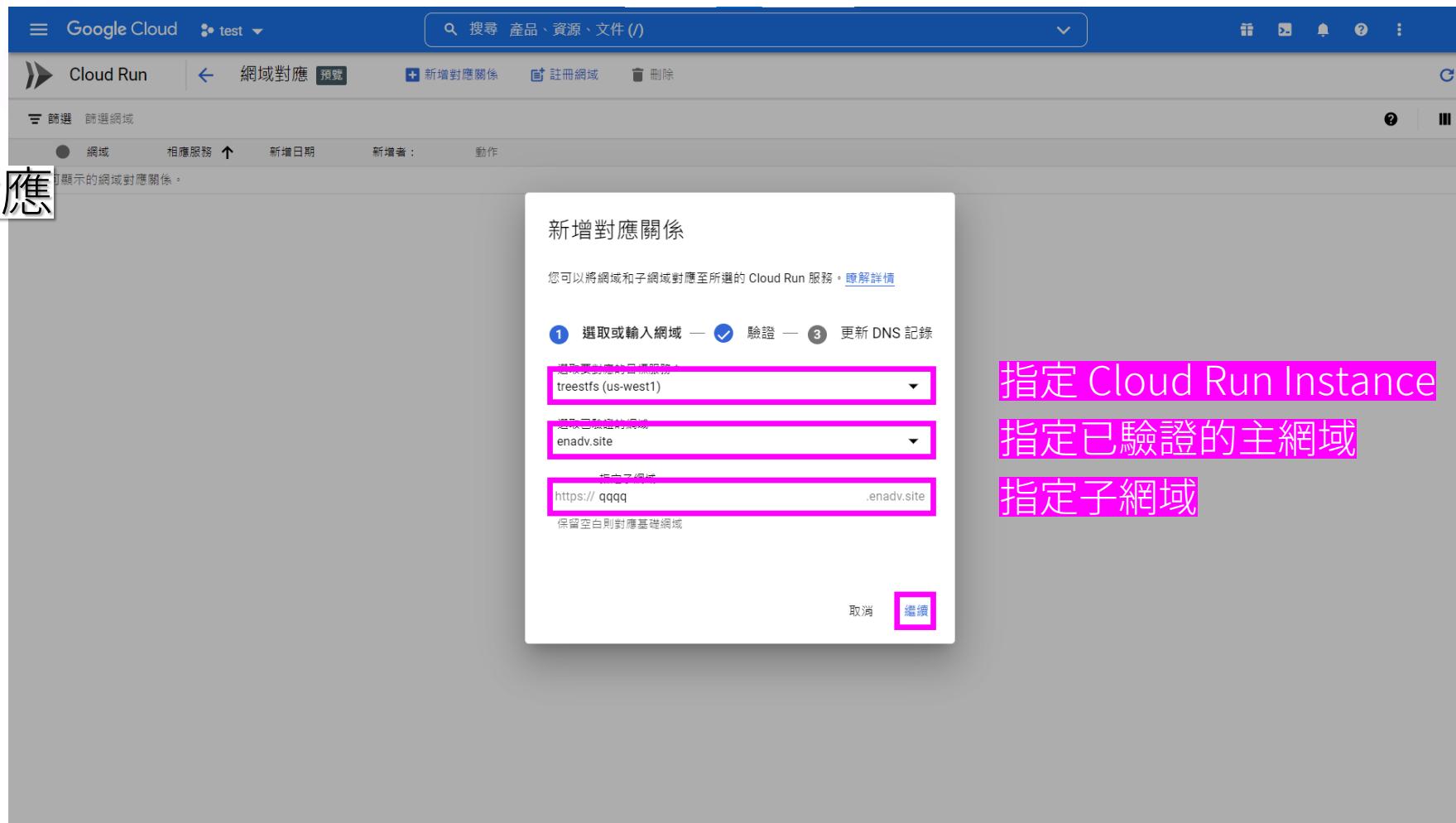
### b. 建立子網域對應



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

### b. 建立子網域對應



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

### b. 建立子網域對應



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

### b. 建立子網域對應

The screenshot shows the GoDaddy DNS Management interface for the domain enadv.site. The main navigation bar includes links for 網域, 買賣, DNS, 設定, and 說明. The current page is '我的網域 / 網域設定' under 'DNS 管理'. A modal window titled 'DNS 記錄' is open, showing a table for managing DNS records. The table has columns for '類型' (Type), '名稱\*' (Name), '內容值\*' (Content Value), and 'TTL'. A new CNAME record is being added, with the type set to 'CNAME', the name set to 'qqqq', the content value set to 'ghs.googlehosted.com.', and the TTL set to '預設'. The modal also contains a note about CNAME records and a '新增記錄' (Add Record) button.

類型	名稱*	內容值*	TTL
CNAME	qqqq	ghs.googlehosted.com.	預設

# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

### b. 建立子網域對應



The screenshot shows the Google Cloud Platform interface for managing Cloud Run domain mappings. The top navigation bar includes 'Google Cloud' with a dropdown for 'test', a search bar with the placeholder '搜尋 container', and various icons for account and project management. The main title is 'Cloud Run' with a sub-section '網域對應'. Below this, there are buttons for '新增對應關係' (Add mapping), '註冊網域' (Register domain), and '刪除' (Delete). A table lists domain mappings:

網域	相應服務	新增日期	新增者	動作
qqqq.enadv.site	treestfs (us-west1)	44 分鐘前	@gmail.com	⋮

# 部署 TensorFlow Serving - Cloud Run

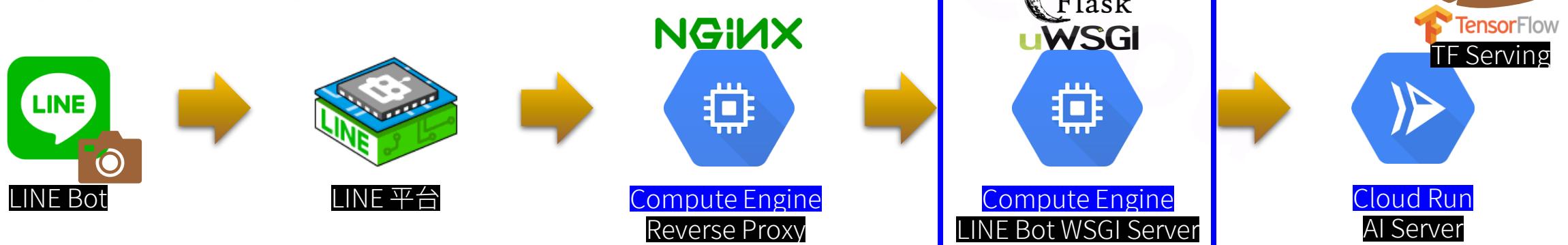
## 5. 測試 TensorFlow Serving

# 部署 LINE Bot + TensorFlow Serving - REST

## Azure's View



## GCP's View



# 部署 LINE Bot + TensorFlow Serving - REST

## 1. 調整 LINE Bot

### a. RESTful API 叫用 TensorFlow Serving 範例

```
rest = 'http://YOUR_REST_HOST:YOUR_REST_PORT/v1/models/YOUR_MODEL_NAME:predict'  
headers = {"content-type": "application/json"}  
data = json.dumps({"instances": img.tolist()})  
r = requests.post(rest, headers=headers, data=data)  
p = np.argmax(r.json()['predictions'])
```

# 部署 LINE Bot + TensorFlow Serving - REST

## 1. 調整 LINE Bot

### b. 下載範例程式並調整

- ① treesbot\_rest.py
- ② treeset\_labels.txt
- ③ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_NAME,  
REST\_SSL, REST\_HOST, REST\_PORT
- ④ other tree samples

# 部署 LINE Bot + TensorFlow Serving - REST

## 1. 調整 LINE Bot

### c. 製作 requirements.txt

```
line-bot-sdk
flask
pillow
tensorflow==2.4.4
requests
numpy
uwsgi
```

# 部署 LINE Bot + TensorFlow Serving - REST

## 2. 重新部署 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

① 上傳檔案至 VM 之專案目錄 (*your\_project*)

`treesbot_rest.py`

`treeset_labels.txt`

`env.json`

`requirements.txt`

# 部署 LINE Bot + TensorFlow Serving - REST

## 2. 重新部署 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

② 更新 VM 系統並安裝 LINE Bot

`cd; cd your_project`

`cd; cd trees`

# 若為新系統，須更新系統並安裝套件；既有系統安裝缺乏的套件即可，不須更新系統

```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade pip;
sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r
requirements.txt --no-warn-script-location; source ../.profile
```

# 部署 LINE Bot + TensorFlow Serving - REST

## 2. 重新部署 LINE Bot

### b. 以 uWSGI 啟動 LINE Bot

```
# 於 VM 以 uWSGI 啟動 LINE Bot
```

```
cd; cd your_project
```

```
cd; cd trees
```

```
uwsgi -w your_module:app -s :your_port -d your_project.log
```

```
uwsgi -w treesbot_rest:app -s :3000 -d trees.log
```

# 部署 LINE Bot + TensorFlow Serving - REST

## 2. 重新部署 LINE Bot

### c. (開通防火牆)

PORT: TCP 3000

#### ① Google Compute Engine 開通防火牆

導覽選單 > 虛擬私有雲網路 > 防火牆 > 建立防火牆規則 >  
名稱："自行命名"；目標："網路中的所有執行個體"；來源 IPv4 範圍："**0.0.0.0/0**"；  
指定的通訊協定和埠："**TCP**" "**3000**" > 建立

#### ② Azure VM 開通防火牆

虛擬機器 > 進入指定 VM > 網路 > 新增輸入連接埠規則 >  
目的地連接埠範圍："**TCP**" "**3000**"；名稱："自行命名" > 新增

# 部署 LINE Bot + TensorFlow Serving - REST

## 2. 重新部署 LINE Bot

### d. (調整 NGINX)

```
# .conf

server {
    server_name your_domain;

    location / {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        #proxy_pass http://your_ip:your_port;
    }
}
```

# 部署 LINE Bot + TensorFlow Serving - REST

## 3. 調整 LINE Messaging

a. (調整 Webhook URL)

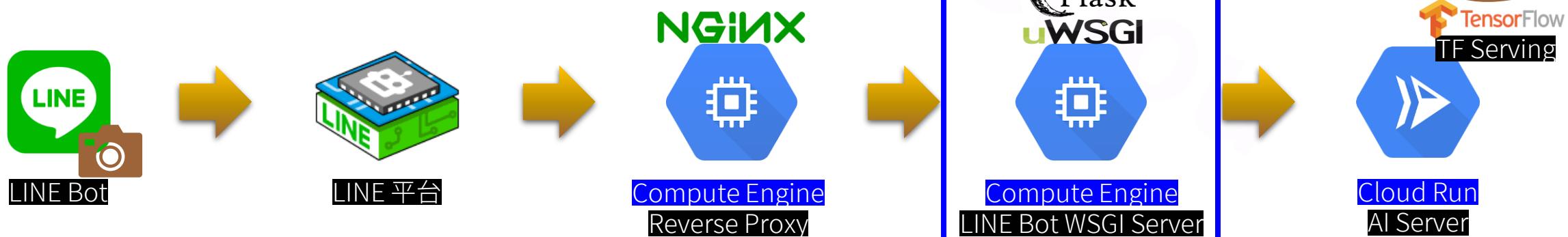
b. LINE 測試

# 部署 LINE Bot + TensorFlow Serving - gRPC

## Azure's View



## GCP's View



# 部署 LINE Bot + TensorFlow Serving - gRPC

## 1. 調整 LINE Bot

### a. gRPC 叫用 TensorFlow Serving 範例

```
grpcurl = 'YOUR_GRPC_HOST:YOUR_GRPC_PORT'  
ssl = YOUR_GRPC_SSL  
model = 'YOUR_MODEL_NAME'  
modelin = 'YOUR_MODEL_IN'  
modelout = 'YOUR_MODEL_OUT'
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 1. 調整 LINE Bot

### a. gRPC 叫用 TensorFlow Serving 範例

```
if ssl:  
    channel = grpc.secure_channel(grpcurl, grpc.ssl_channel_credentials())  
else:  
    channel = grpc.insecure_channel(grpcurl)  
stub = prediction_service_pb2_grpc.PredictionServiceStub(channel)  
req = predict_pb2.PredictRequest()  
req.model_spec.name = model  
req.inputs[modelin].CopyFrom(make_tensor_proto(img))  
r = stub.Predict(req, 13.0)  
p = np.argmax(r.outputs[modelout].float_val)
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 1. 調整 LINE Bot

### b. 下載範例程式並調整

- ① treesbot\_grpc.py
- ② treeset\_labels.txt
- ③ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_NAME,  
MODEL\_IN, MODEL\_OUT, GRPC\_HOST, GRPC\_PORT, GRPC\_SSL
- ④ other tree samples

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 1. 調整 LINE Bot

### c. 製作 requirements.txt

```
line-bot-sdk
flask
pillow
tensorflow==2.4.4
tensorflow-serving-api
requests
numpy
uwsgi
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 2. 重新部署 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

① 上傳檔案至 VM 之專案目錄

`treesbot_grpc.py`

`treeset_labels.txt`

`env.json`

`requirements.txt`

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 2. 重新部署 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

② 更新 VM 系統並安裝 LINE Bot

`cd; cd your_project`

`cd; cd trees`

# 若為新系統，須更新系統並安裝套件；既有系統安裝缺乏的套件即可，不須更新系統

```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade pip;
sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r
requirements.txt --no-warn-script-location; source ../.profile
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 2. 重新部署 LINE Bot

### b. 以 uWSGI 啟動 LINE Bot

```
# 於 VM 以 uWSGI 啟動 LINE Bot
```

```
cd; cd your_project
```

```
cd; cd trees
```

```
uwsgi -w your_module:app -s :your_port -d your_project.log
```

```
uwsgi -w treesbot_grpc:app -s :3000 -d trees.log
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 2. 重新部署 LINE Bot

### c. (開通防火牆)

PORT: TCP 3000

#### ① Google Compute Engine 開通防火牆

導覽選單 > 虛擬私有雲網路 > 防火牆 > 建立防火牆規則 >  
名稱："自行命名"；目標："網路中的所有執行個體"；來源 IPv4 範圍："0.0.0.0/0"；  
指定的通訊協定和埠："TCP" "3000" > 建立

#### ② Azure VM 開通防火牆

虛擬機器 > 進入指定 VM > 網路 > 新增輸入連接埠規則 >  
目的地連接埠範圍："TCP" "3000"；名稱："自行命名" > 新增

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 2. 重新部署 LINE Bot

### d. (調整 NGINX)

```
# .conf

server {
    server_name your_domain;

    location / {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        #proxy_pass http://your_ip:your_port;
    }
}
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 3. 調整 LINE Messaging

a. (調整 Webhook URL)

b. LINE 測試

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

LINE Bot feat. gRPC TensorFlow Serving Issues

① Channel 建立成本高

建立 channel 耗時 ... 重複使用 Channel

② TensorFlow Serving API 叫用 TensorFlow 函式

導致一併安裝 TensorFlow 套件，浪費空間 (> 300MB) ... 去化 TensorFlow

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

### a. 重複使用 Channel

略

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

### b. 去化 TensorFlow

#### ① 自行建立 proto

- git clone https://github.com/tensorflow/tensorflow
- mkdir p; cd tensorflow
- python3 -m grpc.tools.protoc ./tensorflow/core/framework/\*.proto --python\_out=../p --grpc\_python\_out=../p --proto\_path=.
- python3 -m grpc.tools.protoc ./tensorflow/core/example/\*.proto --python\_out=../p --grpc\_python\_out=../p --proto\_path=.
- python3 -m grpc.tools.protoc ./tensorflow/core/protobuf/\*.proto --python\_out=../p --grpc\_python\_out=../p --proto\_path=.
- cd ..; rm -rf tensorflow; mv p/tensorflow .; rmdir p
- manually move tensorflow to project directory

#### Note

- ① 需要套件 grpcio-tools==1.34.1 (or 1.48.2)
- ② 開發環境的 grpcio-tools 與部署環境的 grpcio 版本需一致以避免問題
- ③ 開發環境與部署環境的 protobuf 版本需一致以避免問題
- ④ 可考慮直接於部署環境產出此 protobuf 碼

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

### b. 去化 TensorFlow

```
② 取代 request.inputs[modelin].CopyFrom(make_tensor_proto(img))
from tensorflow.core.framework import types_pb2
from tensorflow.core.framework.tensor_shape_pb2 import TensorShapeProto
from tensorflow.core.framework.tensor_pb2 import TensorProto

d = [TensorShapeProto.Dim(size=x) for x in img.shape]
t_p = TensorProto(dtype=types_pb2.DT_FLOAT, tensor_shape=TensorShapeProto(dim=d),
                  tensor_content=img.tobytes())
req.inputs[modelin].CopyFrom(t_p)
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

### c. 下載範例程式並調整

- ① treesbot\_adv.py
- ② treeset\_labels.txt
- ③ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_NAME,  
MODEL\_IN, MODEL\_OUT, GRPC\_HOST, GRPC\_PORT, GRPC\_SSL
- ④ other tree samples

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 4. 進階調整 LINE Bot

### d. 製作 requirements.txt

line-bot-sdk  
flask  
pillow  
numpy  
grpcio  
protobuf  
uwsgi

#### Note

- ① 須無依賴安裝 tensorflow-serving-api 以免自帶 TensorFlow，故不可將 tensorflow-serving-api 寫入 requirements.txt，須以下列方式分開安裝  
`pip3 install -r requirements.txt`  
  
`pip3 install --no-deps tensorflow-serving-api`
- ② 開發環境的 grpcio-tools 與部署環境的 grpcio 版本需一致
- ③ 開發環境與部署環境的 protobuf 版本需一致

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 5. 重新部署進階 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

① 上傳檔案至 VM 之專案目錄

`treesbot_adv.py`

`treeset_labels.txt`

`env.json`

`requirements.txt`

`tensorflow/ # 經 protoc 編譯的 gRPC 碼`

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 5. 重新部署進階 LINE Bot

### a. 更新系統同時安裝 LINE Bot 與 WSGI Server

② 更新 VM 系統並安裝 LINE Bot

`cd; cd your_project`

`cd; cd trees`

# 若為新系統，須更新系統並安裝套件；既有系統安裝缺乏的套件即可，不須更新系統

```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade pip;
sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r
requirements.txt --no-warn-script-location; source ../.profile
```

```
python3 -m pip install --no-deps tensorflow-serving-api
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 5. 重新部署進階 LINE Bot

### b. 以 uWSGI 啟動 LINE Bot

```
# 於 VM 以 uWSGI 啟動 LINE Bot
```

```
cd; cd your_project
```

```
cd; cd trees
```

```
uwsgi -w your_module:app -s :your_port -d your_project.log
```

```
uwsgi -w treesbot_adv:app -s :3000 -d trees.log
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 5. 重新部署進階 LINE Bot

### c. (開通防火牆)

PORT: TCP 3000

#### ① Google Compute Engine 開通防火牆

導覽選單 > 虛擬私有雲網路 > 防火牆 > 建立防火牆規則 >  
名稱："自行命名"；目標："網路中的所有執行個體"；來源 IPv4 範圍："0.0.0.0/0"；  
指定的通訊協定和埠："TCP" "3000" > 建立

#### ② Azure VM 開通防火牆

虛擬機器 > 進入指定 VM > 網路 > 新增輸入連接埠規則 >  
目的地連接埠範圍："TCP" "3000"；名稱："自行命名" > 新增

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 5. 重新部署進階 LINE Bot

### d. (調整 NGINX)

```
# .conf

server {
    server_name your_domain;

    location / {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        #proxy_pass http://your_ip:your_port;
    }
}
```

# 部署 LINE Bot + TensorFlow Serving - gRPC

## 6. 調整 LINE Messaging

a. (調整 Webhook URL)

b. LINE 測試

# gcloud CLI

## 專案

初始化

```
gcloud init
```

登入

```
gcloud auth login
```

全部登出

```
gcloud auth revoke --all
```

查詢預設專案

```
gcloud config list project
```

設定預設專案代號

```
gcloud config set project your_project_id
```

```
gcloud config set project cryptic-bolt-319014
```

## App Engine

部署

```
gcloud app deploy
```

觀察 log

```
gcloud app logs tail -s default
```

# gcloud CLI

## Compute Engine

SSH 登入

```
gcloud compute ssh your_id@your_vm
```

SCP 複製

```
gcloud compute scp your_file your_id@your_vm:
```

```
gcloud compute scp --recurse your_dir your_id@your_vm:
```

## Cloud Shell

Reset

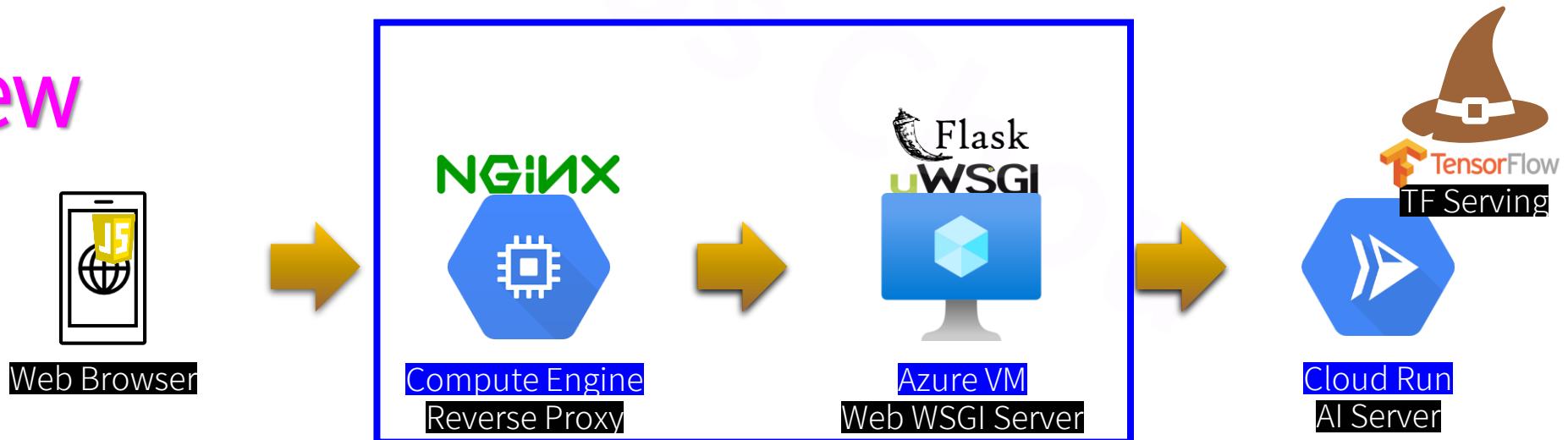
```
sudo rm -rf $HOME
```

重新啟動

# Web + AI 部署

# 部署 Web + TensorFlow Serving

## Mixed View



# 部署 Web + TensorFlow Serving

## 1. TensorFlow Serving 架設

略

# 部署 Web + TensorFlow Serving

## 2. Web 開發

### a. 下載範例程式並調整 (HTML + REST)

- ① treesweb.py
- ② treeset\_labels.txt
- ③ env.json # LABELS, MODEL\_NAME, REST\_SSL, REST\_HOST, REST\_PORT
- ④ other tree samples

# 部署 Web + TensorFlow Serving

## 2. Web 開發

### a. 下載範例程式並調整 (JavaScript + REST)

- ① treesweb.py
- ② treeset\_labels.txt
- ③ env.json # LABELS, MODEL\_NAME, REST\_SSL, REST\_HOST, REST\_PORT
- ④ other tree samples

# 部署 Web + TensorFlow Serving

## 2. Web 開發

### a. 下載範例程式並調整 (JavaScript + gRPC)

- ① treesweb.py
- ② treeset\_labels.txt
- ③ env.json # LABELS, MODEL\_NAME, MODEL\_IN, MODEL\_OUT, GRPC\_SSL,  
GRPC\_HOST, GRPC\_PORT
- ④ other tree samples

# 部署 Web + TensorFlow Serving

## 2. Web 開發

### b. 製作 requirements.txt (for REST)

flask

pillow

numpy

requests

uwsgi

# 部署 Web + TensorFlow Serving

## 2. Web 開發

### b. 製作 requirements.txt (JavaScript + gRPC)

flask  
pillow  
numpy  
grpcio  
protobuf  
uwsgi

#### Note

須無依賴安裝 tensorflow-serving-api 以免自帶 TensorFlow，故不可將 tensorflow-serving-api 寫入 requirements.txt，須以下列方式分開安裝

`pip3 install -r requirements.txt`

`pip3 install --no-deps tensorflow-serving-api`

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### a. 建立 VM

OS: Ubuntu 18.04

RAM: 1GB

HD: 10GB

PORT: TCP 4000

#### Note

此處採用 VM 架設，亦可自行以其他方式架設

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### b. 上傳範例程式至 VM

- ① 於 VM 建立專案目錄

```
cd; mkdir your_project  
cd; mkdir trees
```

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### b. 上傳範例程式至 VM

② 上傳下列檔案至 VM 之專案目錄

`treesweb.py`

`treeset_labels.txt`

`env.json`

③ 上傳 gRPC 碼至 VM 之專案目錄 for gRPC

`tensorflow/ # 經 protoc 編譯的 gRPC 碼`

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### c. 更新系統同時安裝 Web 與 WSGI Server

① 進入專案目錄

```
cd; cd your_project  
cd; cd trees
```

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### c. 更新系統同時安裝 Web 與 WSGI Server

② (若為新系統，須更新系統並安裝套件；既有系統安裝缺乏的套件即可，不須更新系統)

```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade pip;
sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r
requirements.txt --no-warn-script-location; source ../../.profile
```

③ 安裝 TensorFlow Serving API 套件 for gRPC

```
python3 -m pip install --no-deps tensorflow-serving-api
```

# 部署 Web + TensorFlow Serving

## 3. 部署 Web + uWSGI

### d. 以 uWSGI 啟動 Web

```
uwsgi -w your_module:app -s :your_port -d your_project.log
uwsgi -w treesweb:app -s :4000 -d treesweb.log
```

# 部署 Web + TensorFlow Serving

## 4. 部署 NGINX

### a. 建立 VM

OS: Ubuntu 18.04

RAM: 1GB

HD: 10GB

PORT: HTTP 80, HTTPS 443

# 部署 Web + TensorFlow Serving

## 4. 部署 NGINX

### b. 於 VM 架設 NGINX 並申裝憑證

```
# your_project.conf
server {
    server_name your_domain;

    location /your_path {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        client_max_body_size 12M;
    }
}
```

```
# trees.conf
server {
    server_name t.enadv.site;

    location /trees {
        include uwsgi_params;
        uwsgi_pass your_ip:4000;
        client_max_body_size 12M;
    }
}
```

# 部署 Web + TensorFlow Serving

## 5. Browser 測試

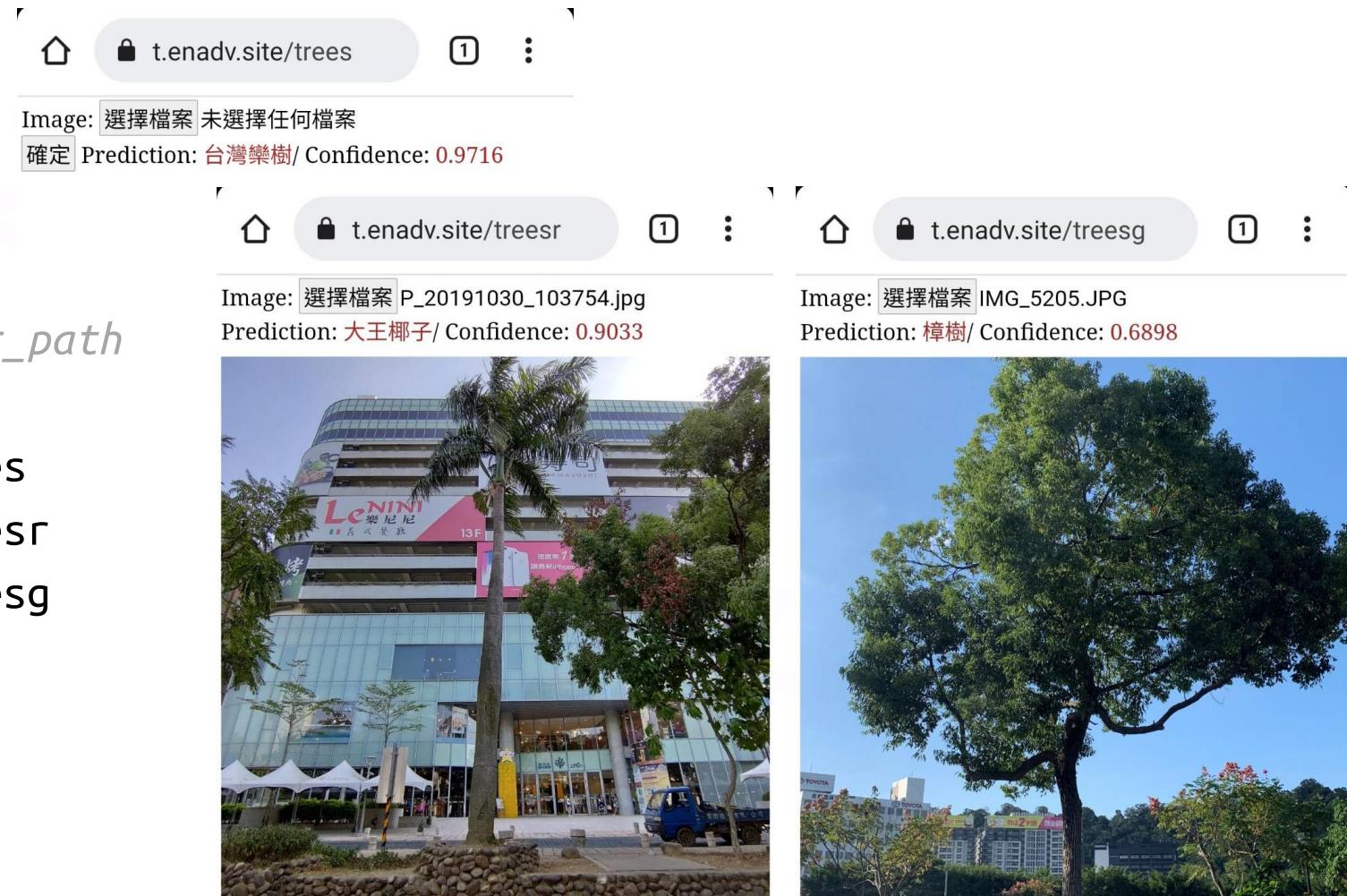
### a. 以瀏覽器開啟網址

`https://your_domain/your_path`

`https://your_domain/trees`

`https://your_domain/treesr`

`https://your_domain/treesg`



# LIFF Web + AI 部署

# LIFF – LINE Front-end Framework

## 背景

- LINE Bot 本質為 interactive
- 選項較多的應用不適合 interactive (poor UX)
- Web 可解決 interactive 問題，但用戶將行為脫離 LINE Bot 掌握

## 方案

- LIFF 為一種 LINE 支援前端 Web 認證的機制，讓用戶能夠以 LINE 帳號登入 Web；登入後 Web 可掌握該用戶的 LINE 資訊

# 建立 LIFF

LINE Developers

News Products Documentation FAQ Glossary Community Blog

Console home

Providers

Search...

Admin

test

Channels Roles Settings

Group by role

Icon	Name	Role
	Create a new channel	
	trees	Admin
		Admin

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our [inquiry form](#).

Family sites English

# 建立 LIFF

LINE Developers

Console home Providers

Search... Admin

test

Channel

Create a new channel

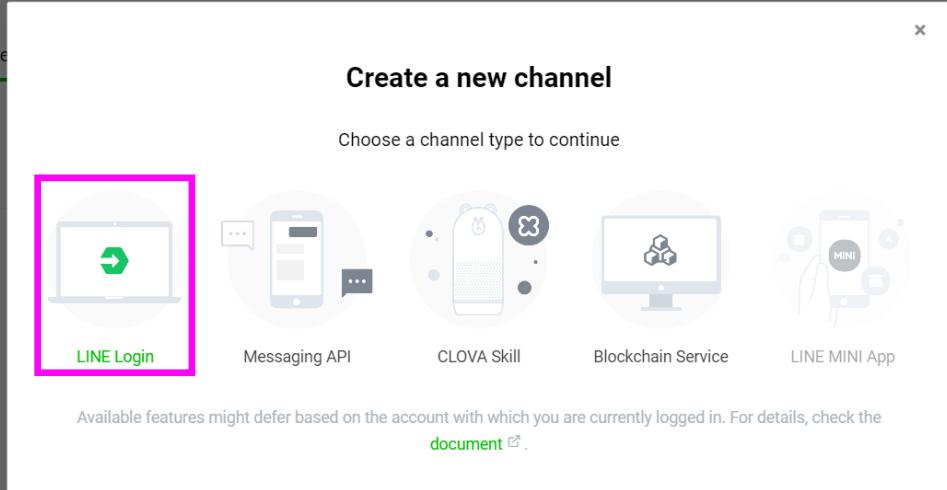
Choose a channel type to continue

LINE Login Messaging API CLOVA Skill Blockchain Service LINE MINI App

Available features might defer based on the account with which you are currently logged in. For details, check the document.

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form. https://developers.line.biz/console/channel/new?type=line-login&provider=16556821...

Family sites English



# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test

TOP Create a new channel

Channel type LINE Login ✓ Don't leave this empty

Provider test ✓ Don't leave this empty

Region to provide the service Japan Thailand Taiwan **Taiwan** Indonesia

Company or owner's country or region Taiwan  
Corporations should select their company's country or region. Individuals should select the country or region of their store, or residence.  
✓ Don't leave this empty

Tools Support Channel icon optional

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

⊕ Family sites English

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test

TOP Register

✓ File type must be one of: PNG,JPG,JPEG,GIF,BMP  
✓ File must be no larger than 3 MB

Channel name treesrealm

✓ Don't leave this empty  
✓ Don't use special characters (4-byte Unicode)  
✓ Enter no more than 20 characters

Channel description treesrealm

✓ Don't leave this empty  
✓ Don't use special characters (4-byte Unicode)  
✓ Enter no more than 500 characters

App types  Web app  Mobile app

✓ Don't leave this empty

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

⊕ Family sites English

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test Tools Support

TOP

Enter a valid email address  
Enter no more than 100 characters

Privacy policy URL optional Enter privacy policy URL  
Enter a valid HTTPS URL  
Enter no more than 500 characters

Terms of use URL optional Enter terms of use URL  
Enter a valid HTTPS URL  
Enter no more than 500 characters

I have read and agree to the [LINE Developers Agreement](#)  
Select the checkbox after reading the related document

Create

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our [inquiry form](#). Family sites English

The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with links like 'Console home', 'Providers' (which is expanded), 'Search...', 'Admin', 'test', 'Tools', and 'Support'. The main area is titled 'TOP' and contains fields for 'Privacy policy URL' and 'Terms of use URL', both of which are marked as 'optional'. Below these fields is a checkbox for accepting the 'LINE Developers Agreement'. At the bottom right of the main area is a large green 'Create' button.

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Admin test Tools Support

TOP > test > treesrealm

treesrealm Admin LINE Login Developing

Basic settings LINE Login LIFF Roles

**Basic settings**

**Basic information**

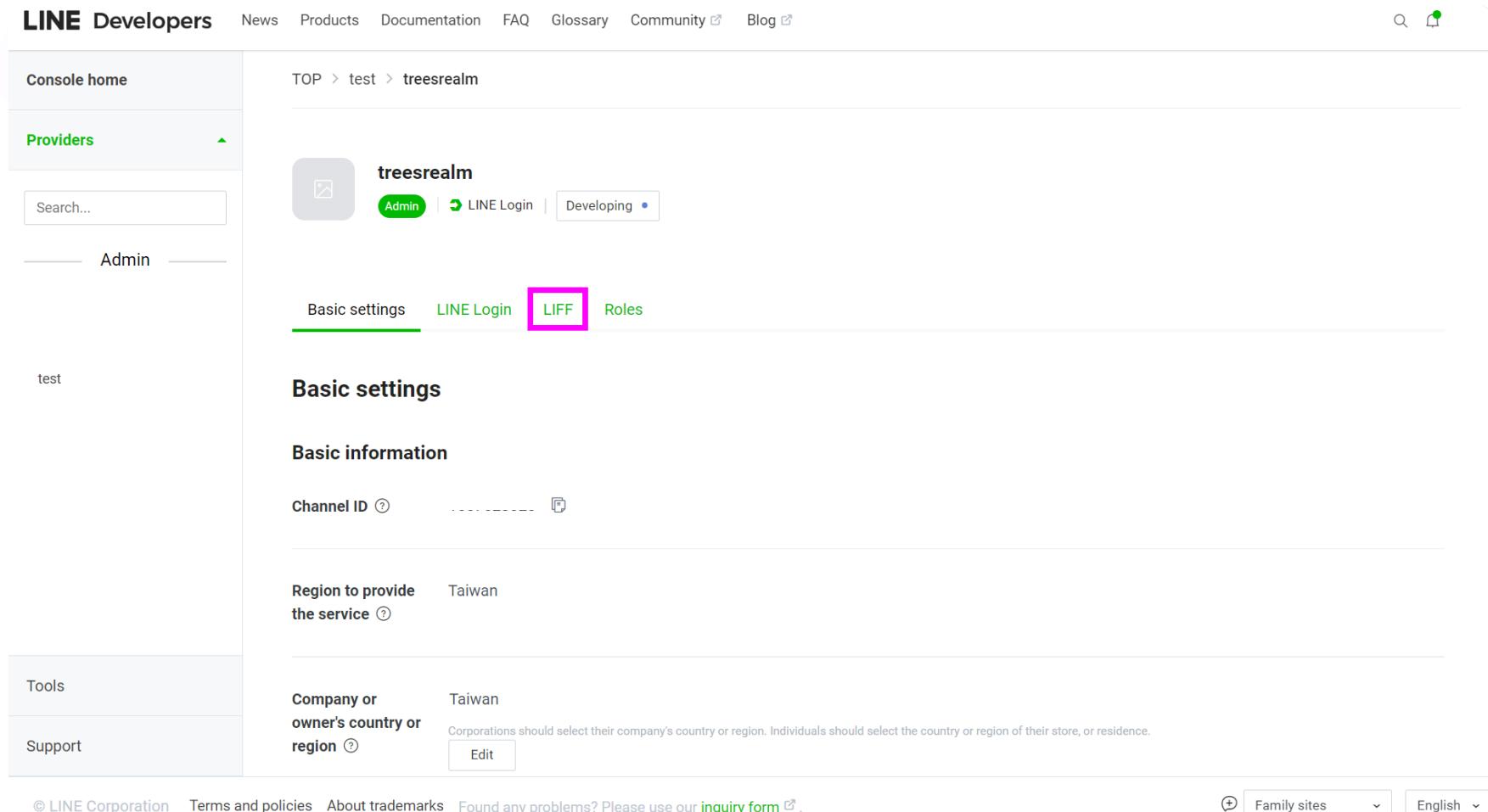
Channel ID ..... Edit

Region to provide the service Taiwan

Company or owner's country or region Taiwan

Corporations should select their company's country or region. Individuals should select the country or region of their store, or residence.

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form. Family sites English



# 建立 LIFF

The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', and sections for 'Admin' and 'test'. The main area shows the 'treesrealm' channel details, including 'Admin', 'LINE Login', and 'Developing'. Below this, tabs for 'Basic settings', 'LINE Login', 'LIFF' (which is selected), and 'Roles' are visible. A section titled 'LIFF applications' contains a bulleted list about LIFF and a link to the 'LINE Developers Agreement'. A message states 'This channel doesn't have any LIFF apps yet' with a 'Add' button. At the bottom, there are links for '© LINE Corporation', 'Terms and policies', 'About trademarks', 'Inquiry form', and language selection for 'Family sites' and 'English'.

LINE Developers

News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test

treesrealm Admin | LINE Login | Developing

Basic settings LINE Login LIFF Roles

**LIFF applications**

- LIFF (LINE Front-end Framework) is a platform for building web apps that can run within LINE. Each LIFF app must be linked to a channel.
- Learn how to add LIFF apps from the [documentation](#)
- The use of a user's LINE account information to deliver advertisements through a third party service is prohibited. For more information, see [LINE Developers Agreement](#)

This channel doesn't have any LIFF apps yet

Add a LIFF app to get started

Add

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our [inquiry form](#). Family sites English

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test Tools Support

TOP > test > treesrealm > LIFF > Add a LIFF app

treesrealm Admin | LINE Login | Developing

Basic settings LINE Login LIFF Roles

Add a LIFF app All fields are required

Basic information

LIFF app name Trees Recognition

Don't leave this empty  
Don't use special characters (4-byte Unicode)  
Enter no more than 256 characters

Size Full Tall Compact

Don't leave this empty

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (selected), a search bar, and sections for 'Admin' and 'test'. Under 'Providers', there are 'Tools' and 'Support'. The main content area shows the navigation path: TOP > test > treesrealm > LIFF > Add a LIFF app. Below this, there's a profile card for 'treesrealm' with roles 'Admin', 'LINE Login', and 'Developing'. A tab bar at the bottom includes 'Basic settings', 'LINE Login', 'LIFF' (which is selected), and 'Roles'. The main form for 'Add a LIFF app' has a required field 'LIFF app name' containing 'Trees Recognition', which is highlighted with a pink border. Below it are validation rules: 'Don't leave this empty', 'Don't use special characters (4-byte Unicode)', and 'Enter no more than 256 characters'. There's also a 'Size' section with radio buttons for 'Full' (selected), 'Tall', and 'Compact', with a note 'Don't leave this empty' below it. At the bottom, there are links for '© LINE Corporation', 'Terms and policies', 'About trademarks', and 'inquiry form', along with language selection for 'Family sites' and 'English'.

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test

Endpoint URL https://tenadv.site/treesrliff

Scopes profile openid

Bot link feature Off

Options Scan QR

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

# 建立 LIFF

LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Search... Admin test Tools Support

TOP > test > treesrealm > LIFF > Add a LIFF app

Enter no more than 1000 characters

Scopes ?  profile  openid  
View all

Don't leave this empty  
Select openid or profile

Bot link feature ?  On (Normal)  On (Aggressive)  Off  
Don't leave this empty

Options

Scan QR ?

Add

Cancel

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

# 建立 LIFF

開發狀態下僅供擁有者測試；  
發布狀態下可供所有人使用

LINE Developers

Console home

Providers

Search...

Admin

test

Tools

Support

TOP > test > treesrealm > LIFF

treesrealm

Admin | LINE Login | Developing

Basic settings LINE Login LIFF Roles

LIFF applications Add

- LIFF (LINE Front-end Framework) is a platform for building web apps that can run within LINE. Each LIFF app must be linked to a channel.
- Learn how to add LIFF apps from the documentation
- The use of a user's LINE account information to deliver advertisements through a third party service is prohibited. For more information, see LINE Developers Agreement

shareTargetPicker

LIFF app name	LIFF ID	LIFF URL	Size
Trees Recognition	REDACTED	https://lfif.line.me/REDACTED	Tall

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

# LIFF Web & LINE Bot

## 1. 重點 LIFF 語法

### a. 載入 LIFF SDK

```
<script charset="utf-8" src="https://static.line-scdn.net/liff/edge/2/sdk.js"></script>
```

Note

單獨一行

# LIFF Web & LINE Bot

## 1. 重點 LIFF 語法

### b. LIFF 初始化

```
<script>
  liff.init({
    liffId: 'YOUR_LIFF_ID'
  })
  .then(() => {
    console.log('LIFF OK');
  })
  .catch((err) => {
    console.log(err);
  });
</script>
```

# LIFF Web & LINE Bot

## 1. 重點 LIFF 語法

### c. 取得用戶資料

```
<script>
    liff.getProfile().then(function(profile) {
        let disp = profile.displayName;
        let uid = profile.userId;
        let pic = profile.pictureUrl;
    });
</script>
```

# LIFF Web & LINE Bot

## 1. 重點 LIFF 語法

### d. 要求 Web 登入

```
<script>
  if (!liff.isLoggedIn()) {liff.login();}
</script>
```

# LIFF Web & LINE Bot

## 2. 準備程式碼

### a. 下載範例程式並調整

- ① treesliff.py
- ② treesbot\_test.py
- ③ treeset\_labels.txt
- ④ env.json # CHANNEL\_SECRET, CHANNEL\_ACCESS\_TOKEN, LABELS, MODEL\_NAME,  
REST\_HOST, REST\_PORT, REST\_SSL, WEB\_URI, LIFF\_ID
- ⑤ other tree samples

# LIFF Web & LINE Bot

## 2. 準備程式碼

### b. 製作 requirements.txt

line-bot-sdk

flask

pillow

requests

numpy

uwsgi

# LIFF Web & LINE Bot

## 3. 部署 Web & LINE Bot

### a. 建立 VM

OS: Ubuntu 18.04

RAM: 1GB

HD: 10GB

PORT: TCP 3000, 4000

#### Note

- ① 此處採用 VM 架設，亦可自行以其他方式架設
- ② 若 Web & LINE Bot 與 NGINX 安裝於相同 VM 則不須開通防火牆

# LIFF Web & LINE Bot

## 3. 部署 Web & LINE Bot

### b. 上傳範例程式至 VM

① 於 VM 建立專案目錄

```
cd; mkdir your_project  
cd; mkdir trees
```

#### Note

若使用既有專案可省略

# LIFF Web & LINE Bot

## 3. 部署 Web & LINE Bot

### b. 上傳範例程式至 VM

② 上傳檔案至 VM 之專案目錄

`treesliff.py`

`treesbot_test.py`

`treeset_labels.txt`

`env.json`

`requirements.txt`

# LIFF Web & LINE Bot

## 3. 部署 Web & LINE Bot

### c. 安裝 Web & LINE Bot

① 進入專案目錄

```
cd; cd your_project
```

```
cd; cd trees
```

② (若為新系統，須更新系統並安裝套件；既有系統安裝缺乏的套件即可，不須更新系統)

```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade pip;
sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r
requirements.txt --no-warn-script-location; source ../.profile
```

# LIFF Web & LINE Bot

## 3. 部署 Web & LINE Bot

### d. 以 uWSGI 啟動 Web & LINE Bot

```
uwsgi -w your_module:app -s :your_port -d your_project.log
```

- ① uwsgi -w treesweb:app -s :4000 -d treesliff.log
- ② uwsgi -w treesbot\_test:app -s :3000 -d treesbot.log

# LIFF Web & LINE Bot

## 4. 部署 NGINX

### a. 建立 VM

OS: Ubuntu 18.04

RAM: 1GB

HD: 10GB

PORT: HTTP 80, HTTPS 443

# LIFF Web & LINE Bot

## 4. 部署 NGINX

### b. 於 VM 架設 NGINX 並申裝憑證

```
# your_project.conf
server {
    server_name your_domain;
    location /your_path {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        client_max_body_size 12M;
    }
}
```

```
# trees.conf
server {
    server_name t.enadv.site;
    include uwsgi_params;

    location /trees {
        uwsgi_pass your_ip:4000;
        client_max_body_size 12M;
    }
    location / {
        uwsgi_pass your_ip:3000;
    }
}
```

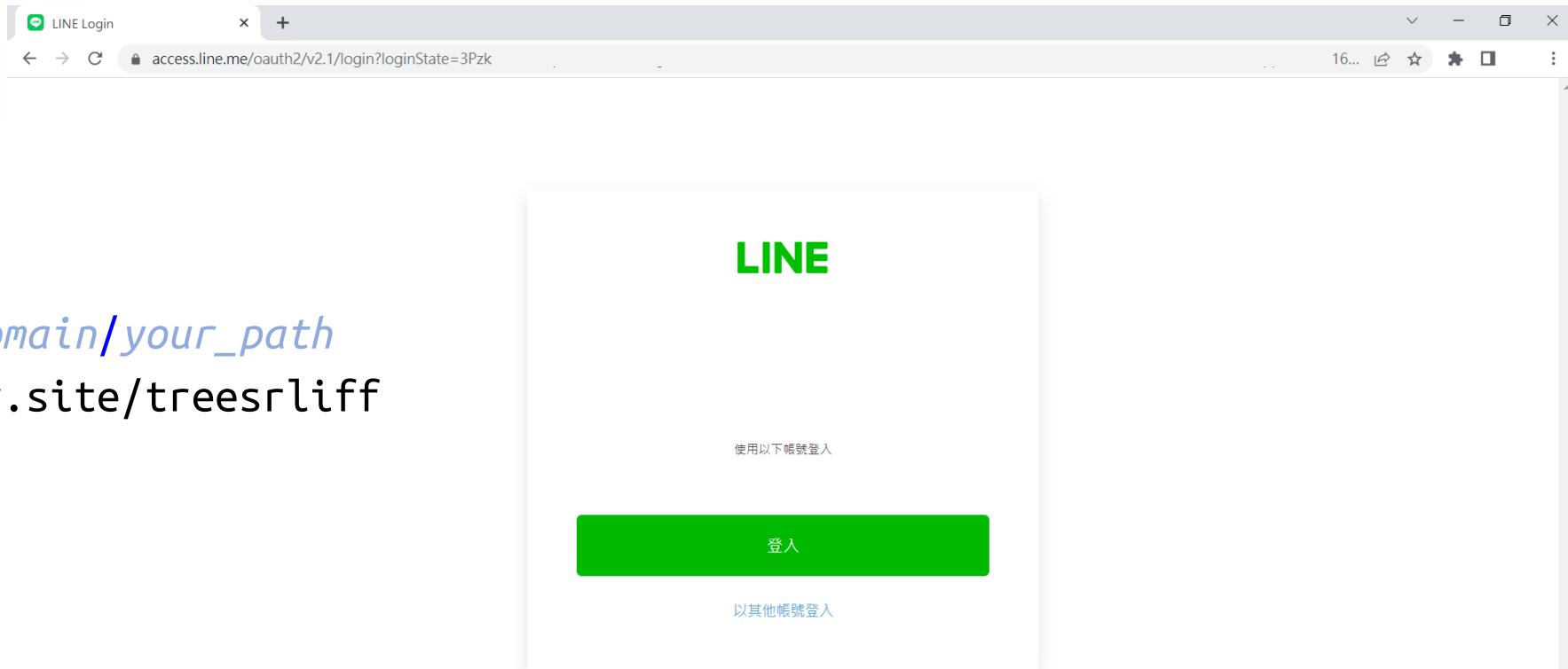
# 啟用 LIFF

## 1. 測試 LIFF

### a. 以 Web 驗證

[https://your\\_domain/your\\_path](https://your_domain/your_path)

<https://t.enadv.site/treesrliff>

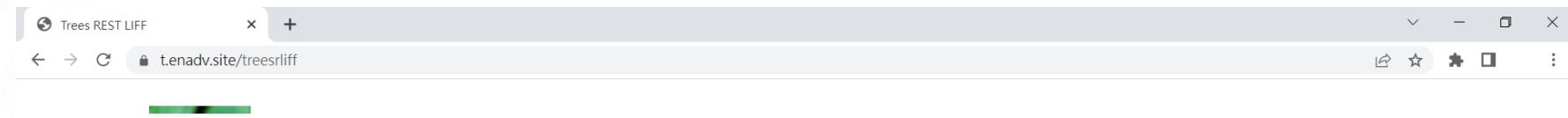


# 啟用 LIFF

## 1. 測試 LIFF

a. 以 Web 驗證

[https://your\\_domain](https://your_domain)  
<https://t.enadv.site/treesrlif>



# 啟用 LIFF

## 1. 測試 LIFF

a. 以 Web 驗證

[https://your\\_domain](https://your_domain)

<https://t.enadv.site/treesrliff>



# 啟用 LIFF

## 1. 測試 LIFF

### b. 以 LINE Bot 驗證

- ① 視需要調整 LINE Messaging 設定
- ② 於 LINE Bot 輸入 fn 啟動功能選單

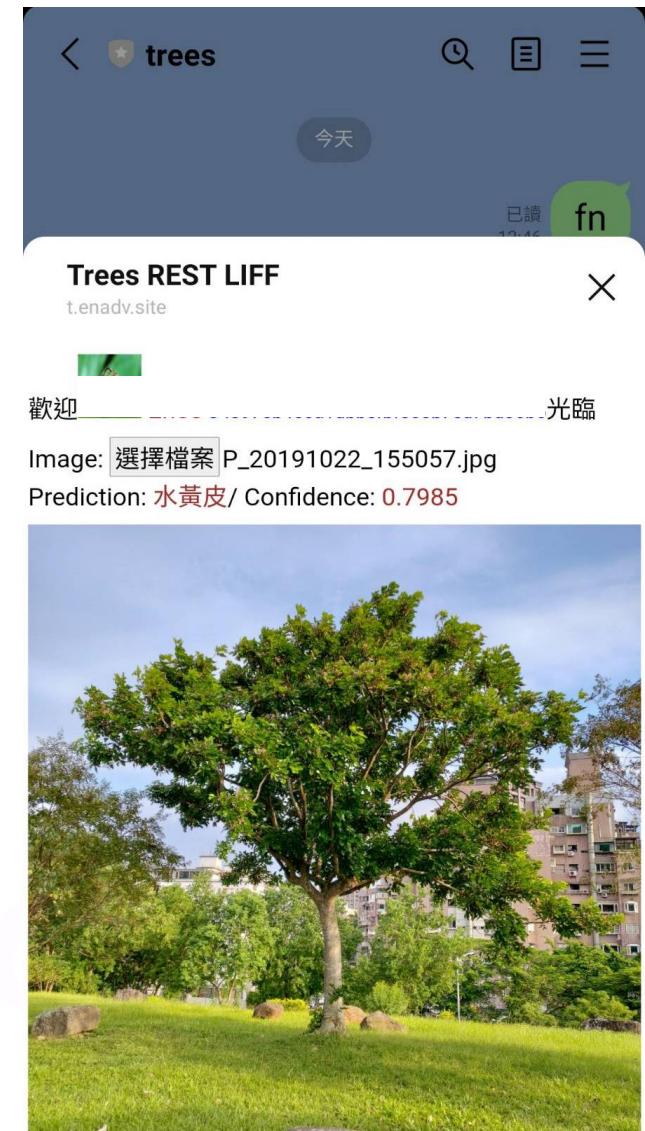


# 啟用 LIFF

## 1. 測試 LIFF

### b. 以 LINE Bot 驗證

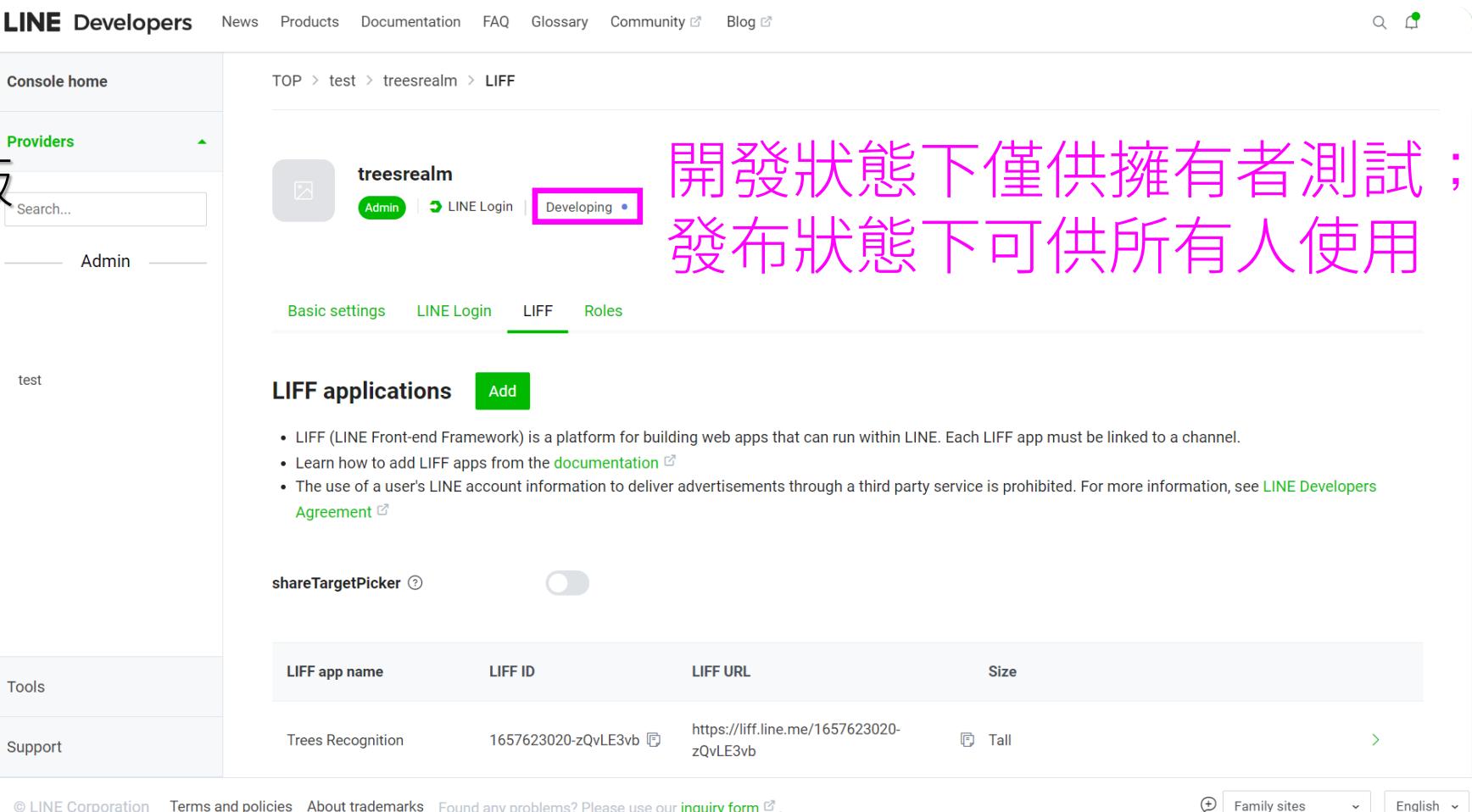
- ① 視需要調整 LINE Messaging 設定
  - ② 於 LINE Bot 輸入 fn 啟動功能選單
  - ③ 選擇 Web 或 LIFF 進行測試



# 啟用 LIFF

## 2. 啟用 LIFF

切換 LIFF 為正式版



LINE Developers

Console home Providers Admin

TOP > test > treesrealm > LIFF

treesrealm Admin LINE Login Developing

Basic settings LINE Login LIFF Roles

LIFF applications Add

- LIFF (LINE Front-end Framework) is a platform for building web apps that can run within LINE. Each LIFF app must be linked to a channel.
- Learn how to add LIFF apps from the documentation
- The use of a user's LINE account information to deliver advertisements through a third party service is prohibited. For more information, see LINE Developers Agreement

shareTargetPicker

LIFF app name	LIFF ID	LIFF URL	Size
Trees Recognition	1657623020-zQvLE3vb	<a href="https://liff.line.me/1657623020-zQvLE3vb">https://liff.line.me/1657623020-zQvLE3vb</a>	<input checked="" type="checkbox"/> Tall

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

# 啟用 LIFF

## 2. 啟用 LIFF

切換 LIFF 為正式版

The screenshot shows the LINE Developers console interface. In the center, a modal dialog box is displayed with the title "Publish this channel?". The dialog contains the following text:  
After you change the channel status from Developing to Published, any LINE user can communicate with the app linked to your channel. You can't undo this. The only way to return to Developing status is to delete the channel and recreate it.  
Below the text are two buttons: "Cancel" and a large green "Publish" button, which is highlighted with a pink rectangle. The background of the modal is white, while the rest of the page is dimmed.

LINE Developers

Console home

Providers

Search...

Admin

test

Tools

Support

TOP > test > treesrealm > LIFF

treesrealm

Admin | LINE Login | Developing

Basic settings LINE Login LIFF Roles

LIFF app

shareTargetPicker

LIFF app name LIFF ID LIFF URL Size

Trees Recognition 1657623020-zQvLE3vb https://lfif.line.me/1657623020-zQvLE3vb Tall

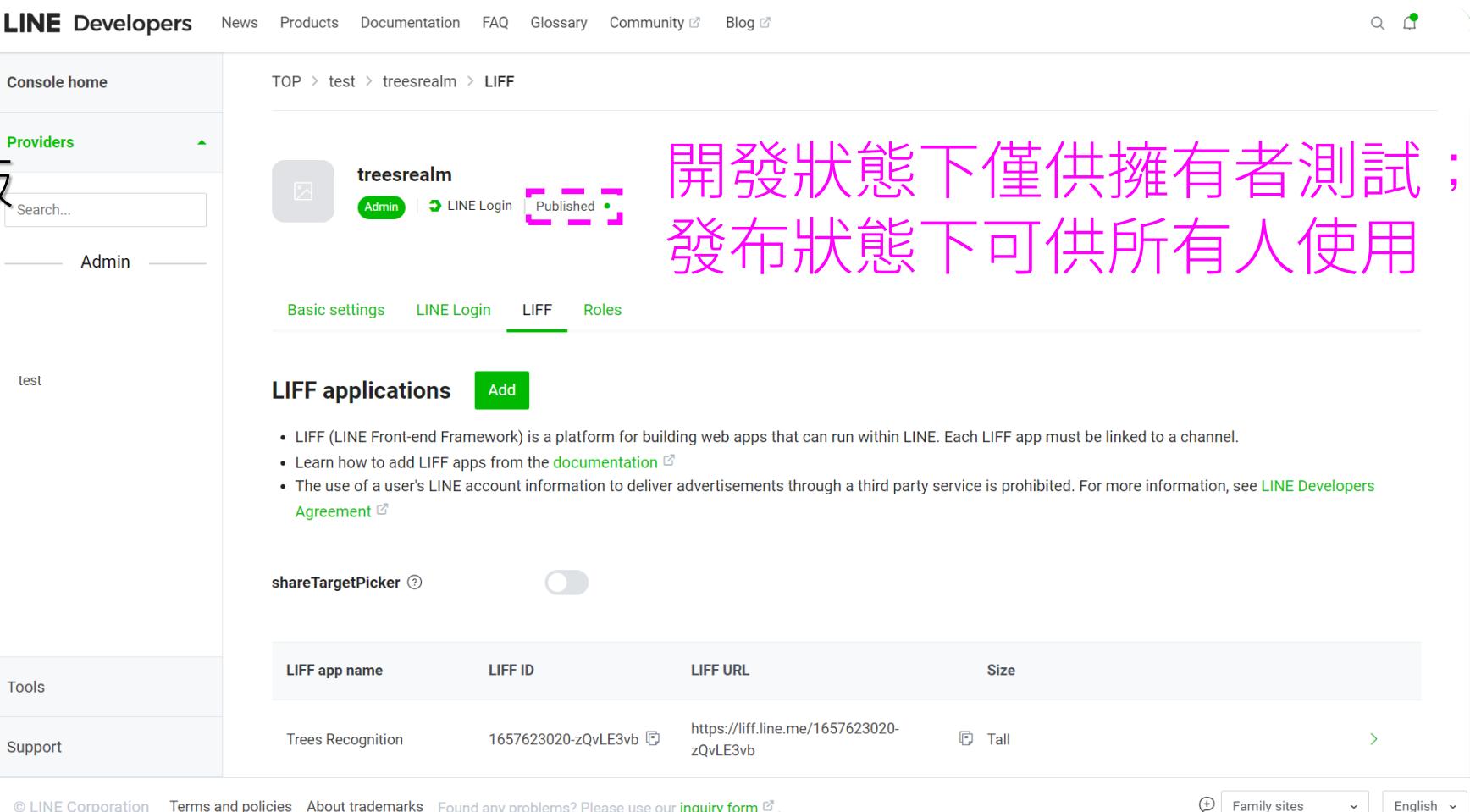
© LINE Corporation Terms and policies About trademarks Found any problems? Please use our inquiry form.

Family sites English

# 啟用 LIFF

## 2. 啟用 LIFF

切換 LIFF 為正式版



LINE Developers

Console home Providers Admin

TOP > test > treesrealm > LIFF

treesrealm Admin LINE Login Published

Basic settings LINE Login LIFF Roles

LIFF applications Add

- LIFF (LINE Front-end Framework) is a platform for building web apps that can run within LINE. Each LIFF app must be linked to a channel.
- Learn how to add LIFF apps from the [documentation](#)
- The use of a user's LINE account information to deliver advertisements through a third party service is prohibited. For more information, see [LINE Developers Agreement](#)

shareTargetPicker

LIFF app name	LIFF ID	LIFF URL	Size
Trees Recognition	1657623020-zQvLE3vb	<a href="https://liff.line.me/1657623020-zQvLE3vb">https://liff.line.me/1657623020-zQvLE3vb</a>	<input type="checkbox"/> Tall >

© LINE Corporation Terms and policies About trademarks Found any problems? Please use our [inquiry form](#).

Family sites English

開發狀態下僅供擁有者測試；  
發布狀態下可供所有人使用

# Web TensorFlow.js AI 部署

# 部署 TensorFlow.js Inference

## 1. 準備模型

### a. 轉換模型為 TensorFlow.js Graph Model 格式

① 安裝 `tensorflowjs`

```
pip install tensorflowjs==3.18.0
```

#### Note

1. `tensorflowjs` 會自帶最新版 `tensorflow`；若先前已安裝 `tensorflow` 則無影響
2. `tensorflowjs 3.19.0` 以上版本會採用較新的 `protobuf` (3.20.0 以上)

# 部署 TensorFlow.js Inference

## 1. 準備模型

### a. 轉換模型為 TensorFlow.js Graph Model 格式

② 下載範例程式並調整

trees17V1.h5

treestfjs.html

treeset\_labels.txt

env.json # LABELS, MODEL\_TFJS

other tree samples

# 部署 TensorFlow.js Inference

## 1. 準備模型

### a. 轉換模型為 TensorFlow.js Graph Model 格式

③ 轉換模型

```
tensorflowjs_converter --input_format keras --output_format tfjs_graph_model  
your_hdf5.h5 your_tfjs
```

```
tensorflowjs_converter --input_format keras --output_format tfjs_graph_model  
trees17V1.h5 tfjs
```

# 部署 TensorFlow.js Inference

## 2. 開發測試

### a. 開發 TensorFlow.js Inference

<https://www.tensorflow.org/js/>

# 部署 TensorFlow.js Inference

## 2. 開發測試

### b. 架設測試 HTTP Server

```
python -m http.server your_port  
python -m http.server 8000
```

# 部署 TensorFlow.js Inference

## 2. 開發測試

### c. 網頁預覽

`http://localhost:your_port/your_html`

# 部署 TensorFlow.js Inference

## 3. 部署

### a. 建立 VM

OS: Ubuntu 18.04

RAM: 1GB

HD: 10GB

PORT: HTTP 80, HTTPS 443

# 部署 TensorFlow.js Inference

## 3. 部署

### b. 上傳範例程式與模型至 VM

- ① 於 VM 建立專案目錄

```
cd; mkdir your_project  
cd; mkdir treestfjs
```

- ② 上傳下列檔案至 VM 之專案目錄

```
tfjs/ # TensorFlow.js 模型目錄  
treestfjs.html  
treeset_labels.txt  
env.json
```

# 部署 TensorFlow.js Inference

## 3. 部署

### c. 於 VM 架設 NGINX 並申裝憑證

```
# your_project.conf
server {
    server_name your_domain;

    location /your_path {
        alias /home/your_account/your_project;
        index your_html;
    }
}
```

```
# trees.conf
server {
    server_name t.enadv.site;

    location /treestfjs {
        alias /home/eng/treestfjs;
        index treestfjs.html;
    }
}
```

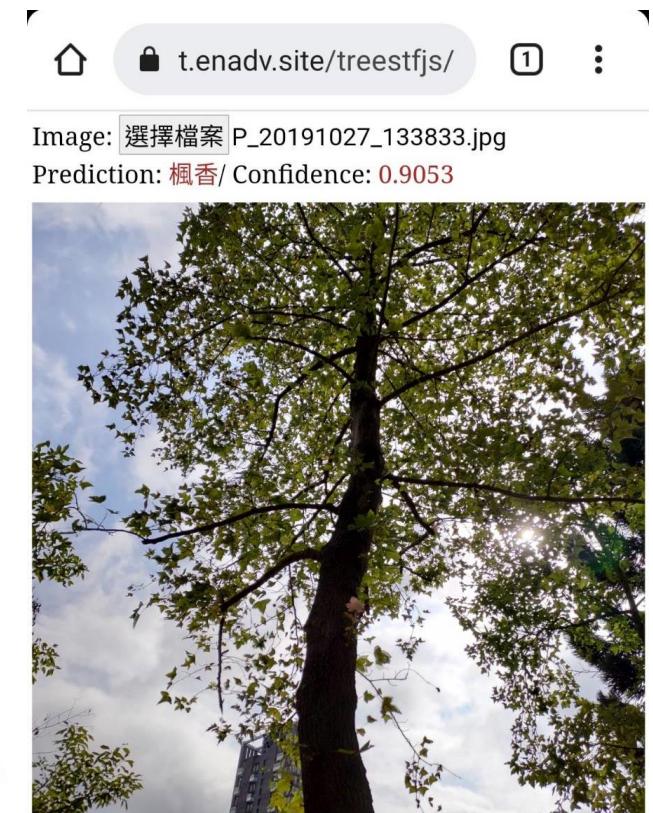
# 部署 TensorFlow.js Inference

## 4. Browser 測試

a. 以瀏覽器開啟網址

*https://your\_domain/your\_path*

*https://your\_domain/treestfjs*





# The End