

# AI 雲端部署 feat. LINE Bot & Web

Enos Chou

2022/10/05, 10/07

# Enos' Steps

2022/09/01(四) 10:00~16:30

專題分享 by Enos

2022/09/07(三) 09:00~16:30

專題訂定 by Teams feat. Enos

2022/10/05(三) 09:00~16:30

AI 雲端部署 by Enos

2022/10/07(五) 09:00~16:30

AI 雲端部署 by Enos

2022/11/01(二) 09:00~16:30

專題預報 by Teams feat. Enos

# Focus

LINE Bot AI 部署

LINE Bot 部署 + AI 部署

Web + AI 部署

Web TensorFlow.js AI 部署

# 課前準備

## 1. New GCP 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

## 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 虛擬環境，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 ?

雲端 vs 地端

To Tech

Scalability

To Biz

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 (app.run())
```

```
* Serving Flask app 'somebot' (lazy loading)
```

```
* Environment: production
```

```
WARNING: This is a development server. Do not use it in a production deployment.
```

```
Use a production WSGI server instead.
```

```
* Debug mode: off
```

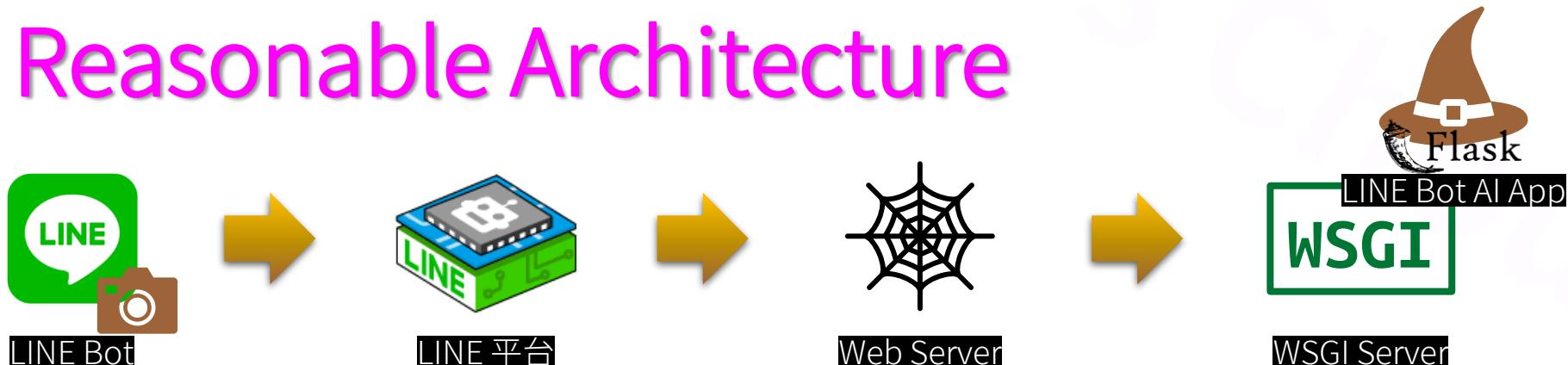
```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

# Architecture

## Warning from Flask

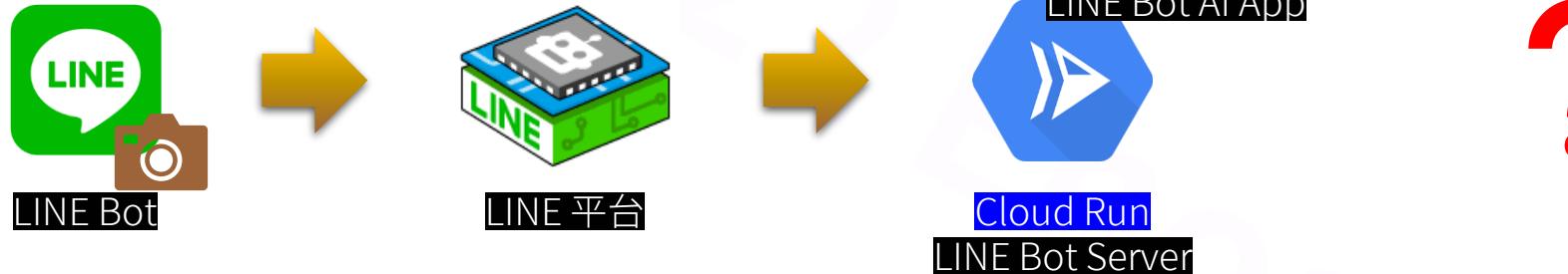


## Reasonable Architecture

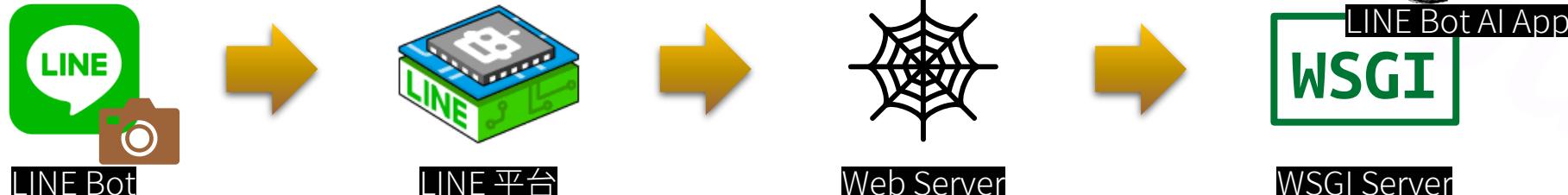


# Architecture

## Imaging Cloud Run

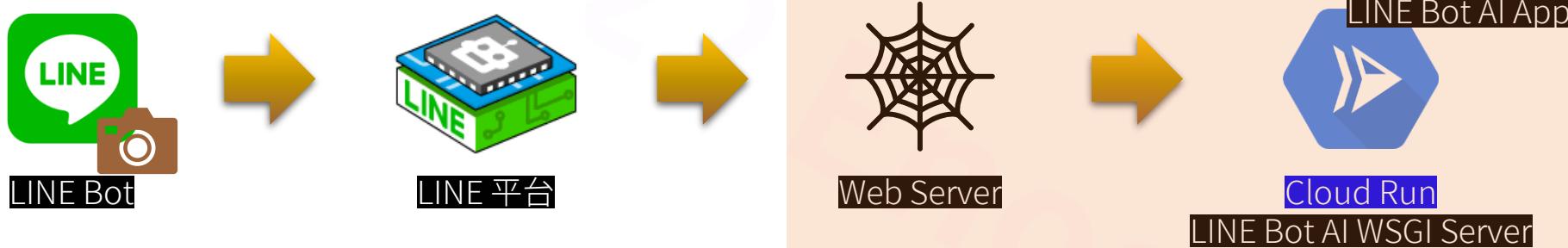


## Reasonable Architecture

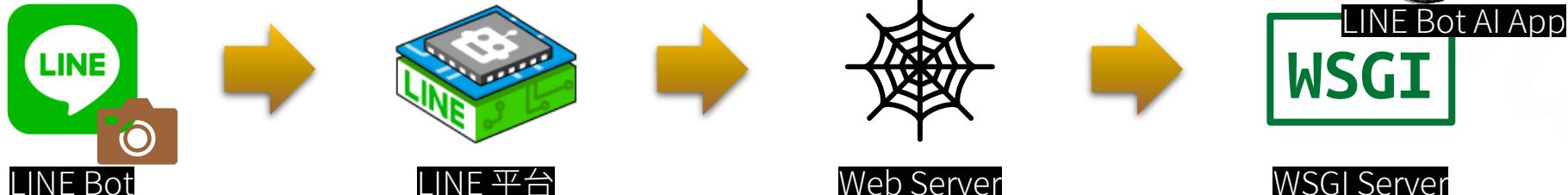


# Architecture

## Reasonable Cloud Run

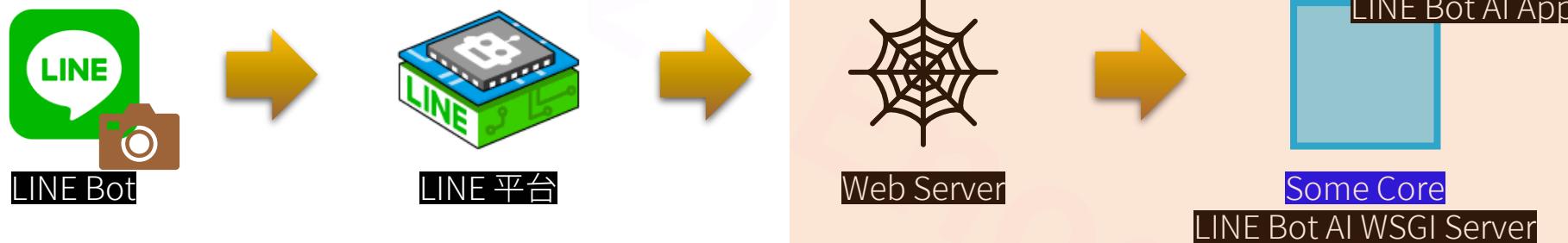


## Reasonable Architecture

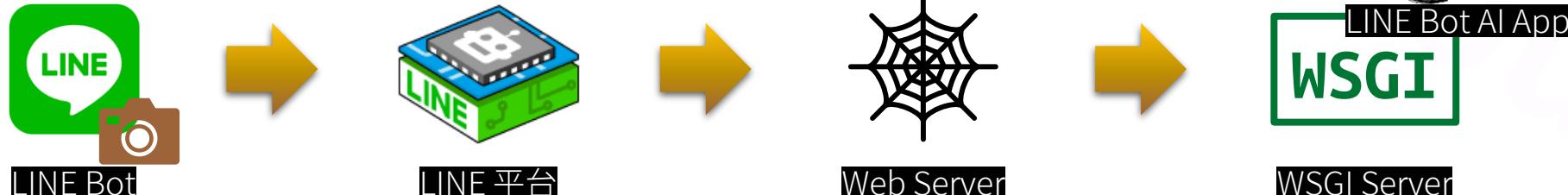


# Architecture

## Serverless

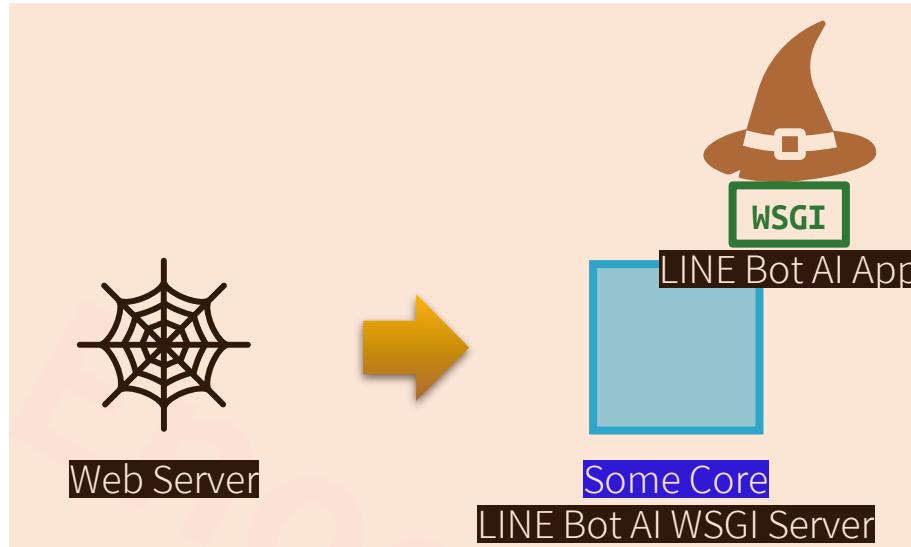
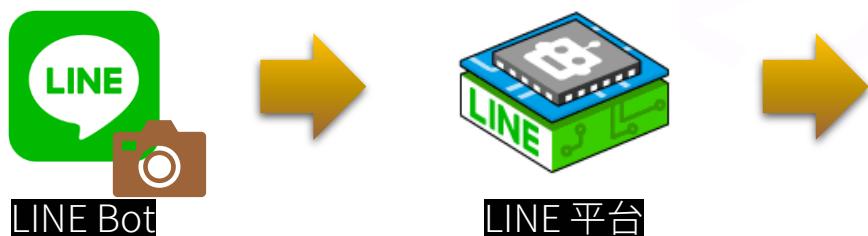


## Reasonable Architecture



# Architecture

## Serverless

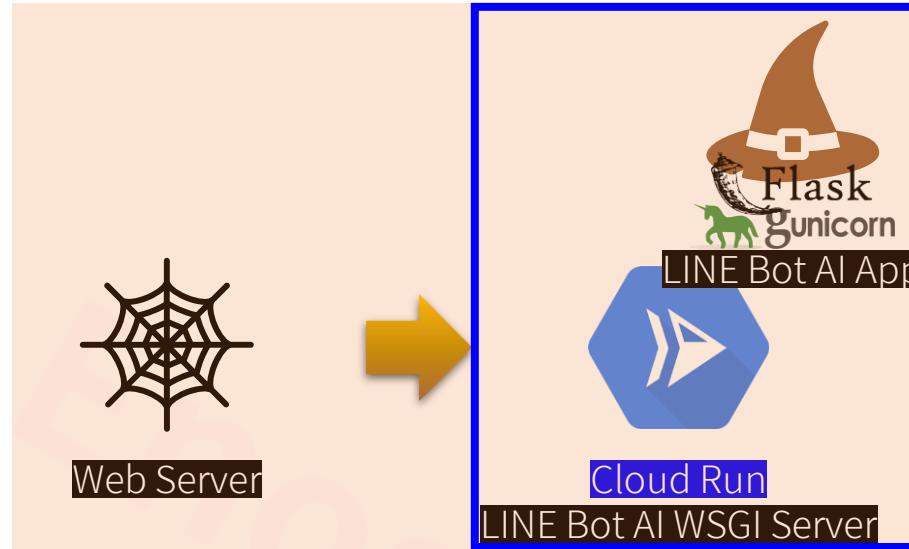
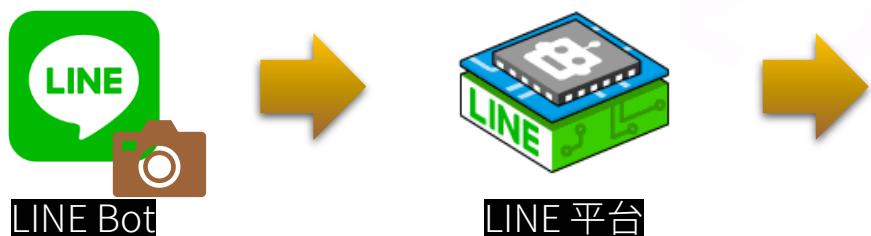


## Server or VM

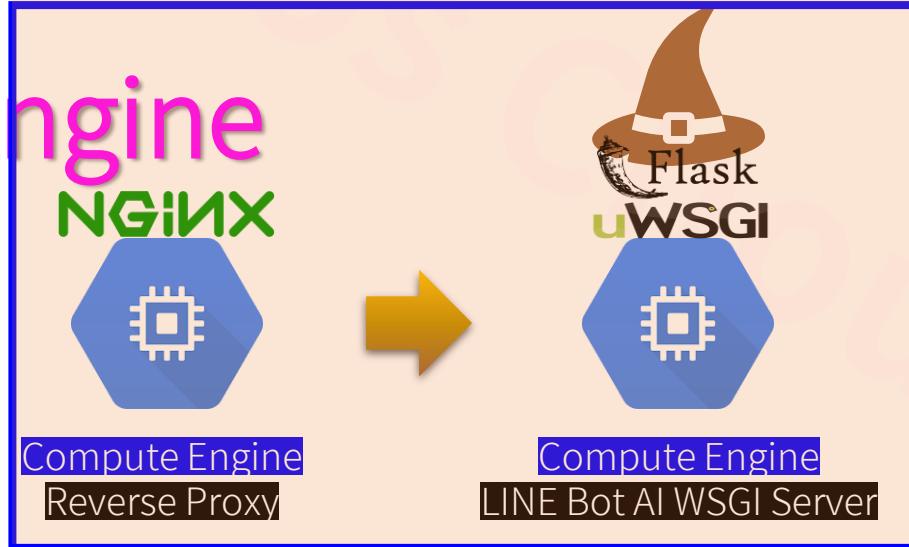


# Architecture

## 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

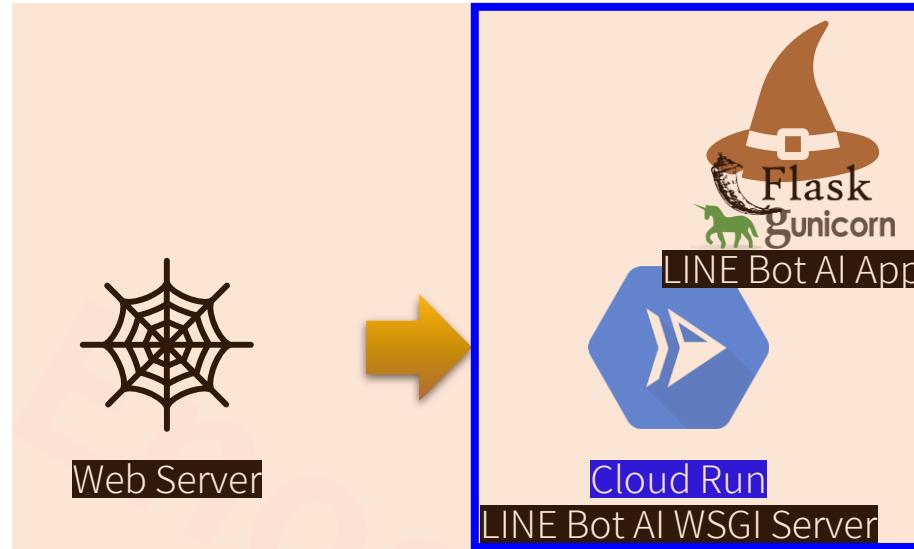
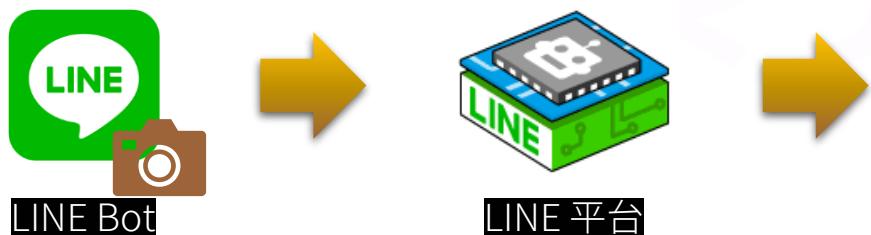
The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (which is expanded to show 'Available APIs'), 'Search...', and user account information ('Admin'). The main content area shows the path 'TOP > test > trees > Messaging API'. Under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

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

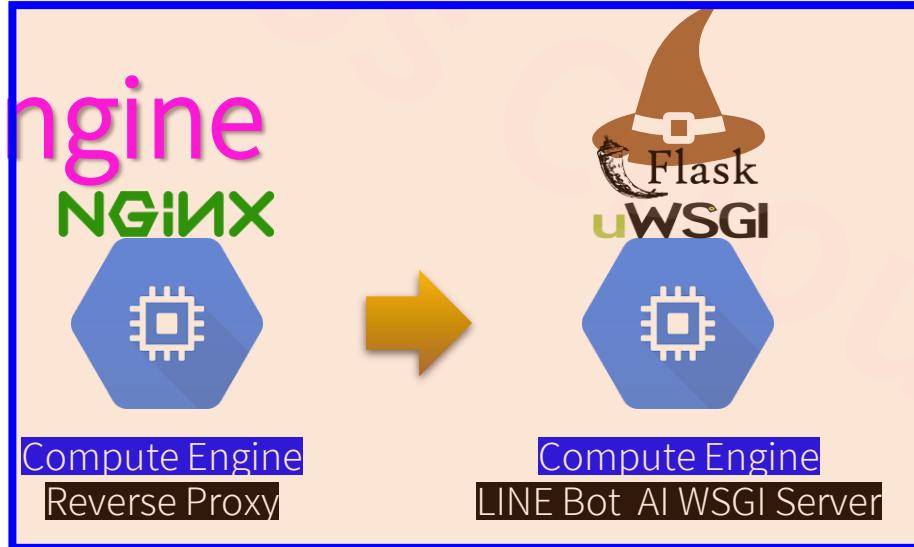
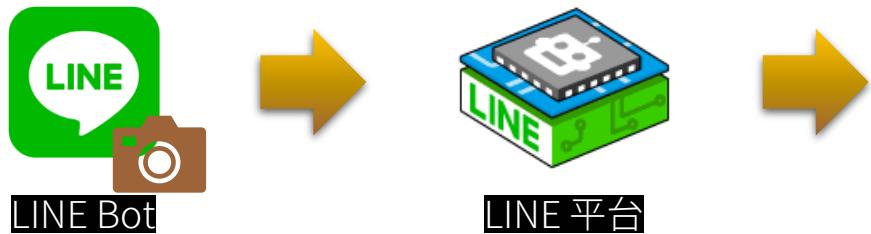
⊕ Family sites English

# 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

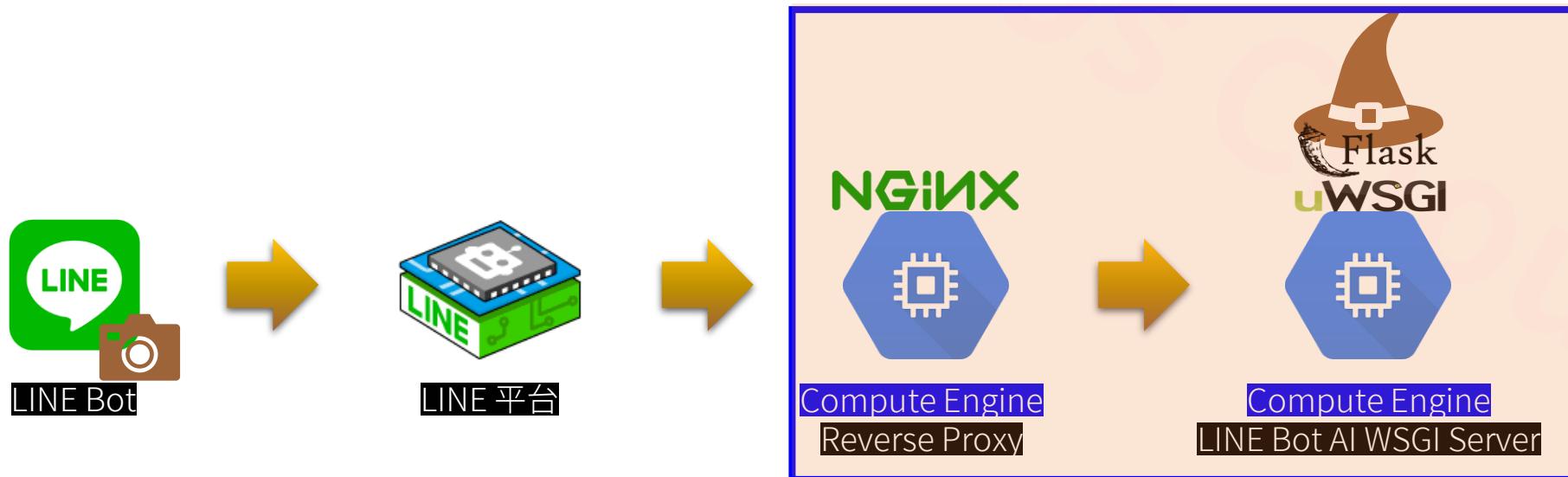
Cloud	Server (VM)	Serverless	Note
Heroku	N/A	Heroku	<a href="http://heroku.com/">http://heroku.com/</a> > Pricing
GCP	Compute Engine	Cloud Run	*GCP 需要每 3 個月換帳號才能免費執行
Azure	VM	Web App	*Azure 需要 Coupon 才能免費執行
AWS	EC2	?	?

## Note

free trial vs free tier

# Solution 1 - VM 部署

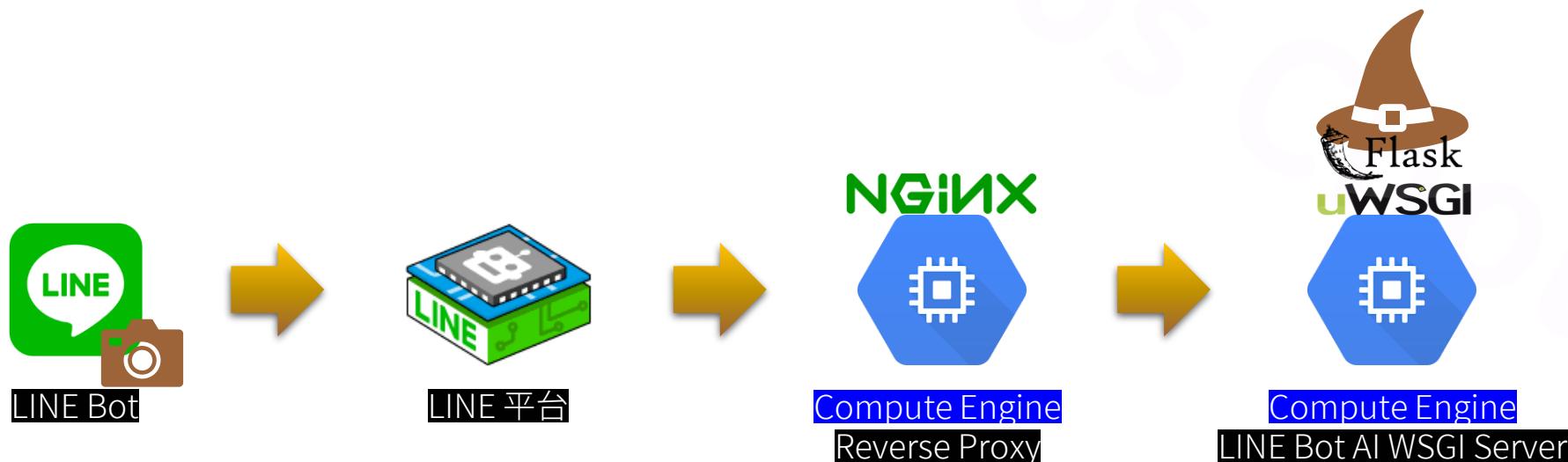
1. 長期可用且廉價的硬體環境 ... Google Compute Engine/ Azure VM
2. Flask as Web Server 的替代方案 ... NGINX + Domain + uWSGI
3. 長期可用且廉價的 SSL 網域方案 ... Let's Encrypt + Certbot

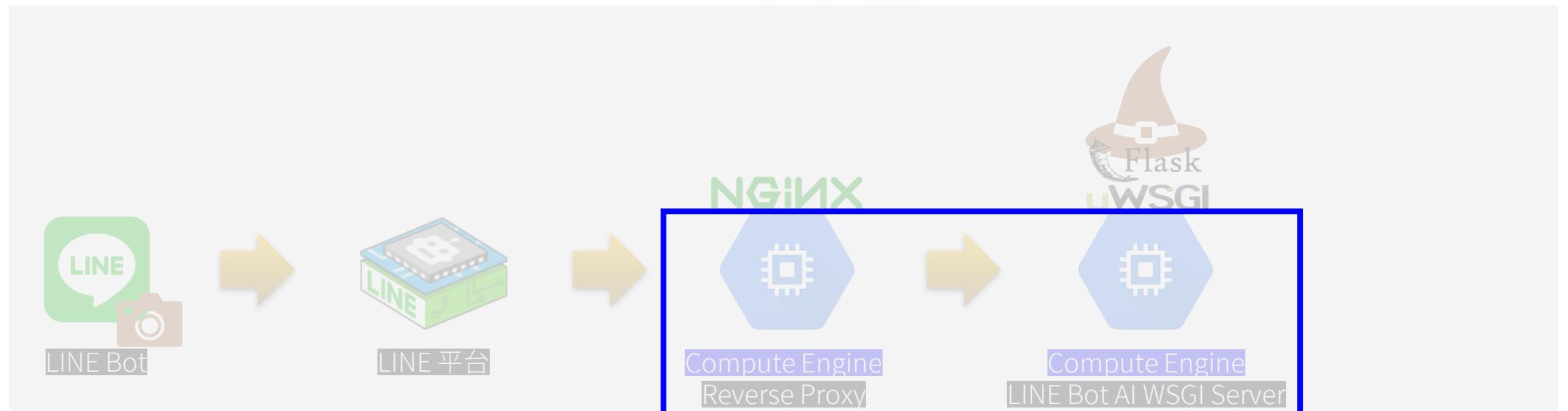


# 任務

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





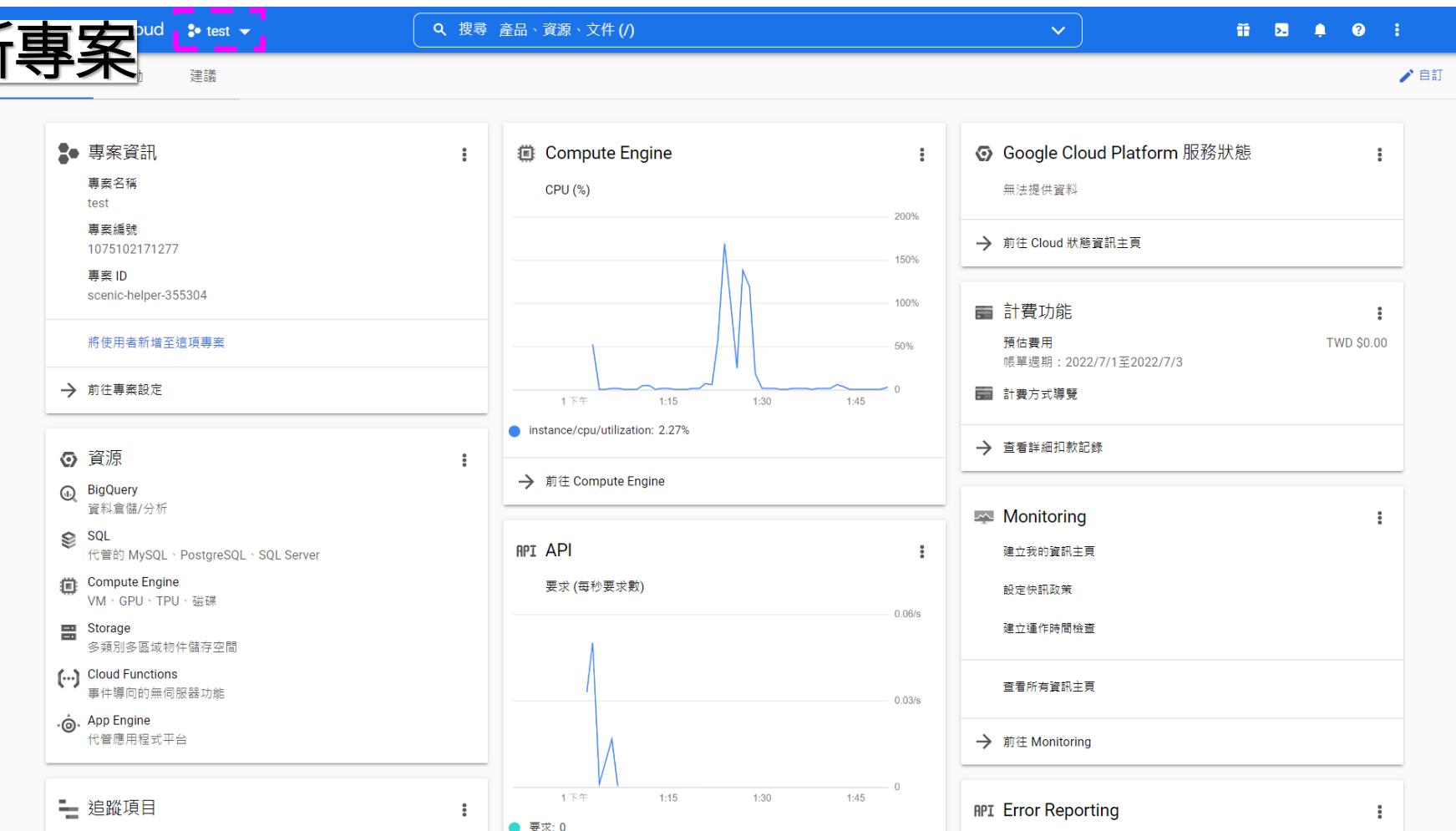
# 架設 VM - GCP

## 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 box. The main content area displays two charts: 'Compute Engine' and 'API API'. The 'Compute Engine' chart shows CPU utilization over time, with a sharp peak around 1:25. The 'API API' chart shows requests per second, also with a sharp peak around 1:25. To the right of the charts, there are sections for 'Google Cloud Platform 服務狀態' (Status), '計費功能' (Billing), 'Monitoring', and 'API Error Reporting'. The URL at the bottom of the page is <https://console.cloud.google.com/compute?authuser=3&project=scenic-helper-355304>.

# 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 tabs for "總覽", "說明文件", and "支援". The "總覽" tab is selected and underlined. The "總覽" section contains a brief description: "Creates and runs virtual machines on Google Cloud Platform." To the right of this, under the heading "其他詳細資料", there are details: "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". Further down, there's a "教學課程與說明文件" section with a "Learn more" link, and a "支援" section with a "瞭解詳情" link.

# 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 displays a table of VM instances with columns for '狀態' (Status), '名稱' (Name), '區域' (Region), '建議' (Recommendation), '使用者' (User), '內部 IP' (Internal IP), '外部 IP' (External IP), and '連線' (Connection). A large globe icon is centered above the table. The bottom right of the main area contains a summary of Compute Engine features and two buttons: '建立執行個體' (Create VM) and '進入快速入門導覽課程' (Enter Quick Start Guide). On the right side, there are tabs for 'PERMISSIONS' (selected), 'LABELS', and 'MONITORING'. A message at the bottom right says '請至少選取一項資源。' (Please select at least one resource.).

# 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).

On the right, there's a sidebar with '預估每月費用' (Estimated monthly cost) showing US\$13.23 and a table of costs for different configurations.

The 'Machine Type' section is highlighted with a pink box, showing '系列' (Series) set to 'E2' and '機器大小' (Machine Size) set to 'e2-small (2 vCPU, 2 GB 記憶體)'. Below it, the 'CPU' and 'Memory' details are shown: 'vCPU' (1 個共用核心) and 'Memory' (2 GB).

Other sections visible include '區域' (Region), '區域' (Zone), '機器設定' (Machine Settings), 'Compute Engine 定價' (Compute Engine Pricing), and 'LESS'.

At the bottom, there are sections for '顯示裝置' (Display Device) and '機密 VM 服務' (Secret VM Service).

# 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 Platform interface for creating a new VM 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 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**: Firewall rules are listed but not yet applied.
- 進階選項**: Advanced options like network, disk, and security are available.

A large blue button at the bottom right says '選取' (Select). A pink box highlights the 'Ubuntu' image selection.

**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 includes:

- 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)
- 進階選項 (Advanced Options):** 包括網路、磁碟、安全性、管理、單一用戶群等。

在界面右侧，有粉色高亮的文字说明：

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 'test' project dropdowns, a search bar, and various icons.

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:

- 針對各個 API 設定存取權** (Set Access for All APIs)
- 防火牆** (Firewall):
  - 允許 HTTP 流量 (Allow HTTP traffic)
  - 允許 HTTPS 流量 (Allow HTTPS traffic)
- 進階選項** (Advanced Options) - Expanded.
- 網路** (Networking):
  - 主機名稱和網路介面 (Host Name and Network Interface)
- 磁碟** (Storage):
  - 其他磁碟 (Other Disks)
- 安全性** (Security):
  - 受防護的 VM 和安全殼層金鑰 (Protected VM and Secure Shell Key)
- 管理** (Management):
  - 說明、防刪除功能、保留項目、自動化和可用性政策 (Description, Delete Protection, Retention Policy, Automation, and Availability Policy)
- 單一用戶群** (Single User Group):
  - 節點相依性標籤和 CPU 超額配置 (Node Labels and CPU Overcommitment)

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 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** (Networking):
  - 網路介面一經設定即無法變更 (Network interface settings 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** (Storage): Other storage.
- 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): Configuration for shared memory and CPU usage.

**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. The top navigation bar includes 'Google Cloud' and a project dropdown ('test'). A search bar is at the top right. Below the header, the page 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 disk 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'. 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 summary, there's a note: '用多少付多少：無須預繳費用，而且是以秒計費' (Pay as you go: no prepayment required, charged by second).

The 'Compute Engine 定價' (Compute Engine Pricing) section includes a 'LESS' link.

The central form for '編輯網路介面' (Edit Network Interface) contains fields for '網路' (Network) set to 'default', '子網路' (Subnetwork) set to 'default IPv4 (10.138.0.0/20)', and 'IP 堆疊類型' (IP Stack Type) set to 'IPv4 (單一堆疊)' (IPv4 (Single stack)). The '主要內部 IP' (Primary Internal IP) dropdown is set to '臨時 (自動)' (Temporary (Automatic)). The '別名 IP 範圍' (Alias IP Range) dropdown is highlighted with a pink rectangle and set to '臨時' (Temporary). The '網路服務級別' (Network Service Level) dropdown is set to '進階' (Advanced).

# 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 header, the page 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'. To the right, a '預估每月費用' (Estimated monthly cost) table shows a total of US\$13.43. The table includes rows for '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), and 'Total' (US\$13.43).

The 'Compute Engine 定價' (Compute Engine Pricing) section contains a note: '如要使用 IPv6，您必須擁有 IPv6 子網路範圍。' (To use IPv6, you must have an IPv6 subnet range.)

The 'IP 堆疊類型' (IP stack type) section has two radio buttons: 'IPv4 (單一堆疊)' (selected) and 'IPv4 和 IPv6 (雙重堆疊)'. Below this is a dropdown menu with '別無' (None) selected, and a button labeled '建立 IP 位址' (Create IP address) which is highlighted with a pink rectangle.

The '網路服務級別' (Network service level) section has two radio buttons: '進階' (selected) and '標準級 (us-west1)'.

# 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 open, asking for a name 'botip' and has a 'Keep' button 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) with 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 instance. 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), '運用範本建立新的 VM 執行個體' (Create VM from Template), '運用機器映像檔建立新的 VM 執行個體' (Create VM from Image), and 'Marketplace'.

The main configuration 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, there's a '編輯網路介面' (Edit Network Interface) section. It lists the '網路' (Network) as 'default' and the '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。' (To use IPv6, you must have an IPv6 subnet range.)

Below this, under 'IP 堆疊類型' (IP Stack Type), 'IPv4 (單一堆疊)' (IPv4 (Single Stack)) is selected. Under '主要內部 IP' (Primary Internal IP), it says '臨時 (自動)' (Temporary (Automatic)).

Under '別名 IP 範圍' (Alias IP Range), there is a button '+ 新增 IP 範圍' (Add IP Range) and a listed entry '外部 IPv4 位址' (External IPv4 Address) with the value 'botip (34.105.119.208)'. At the bottom, '網路服務級別' (Network Service Level) is set to '進階' (Advanced).

On the right side, there is a '預估每月費用' (Estimated Monthly Cost) 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, there 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 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 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 credits. [Google Cloud Free Trial](#))

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

**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. The left sidebar has sections for VM execution instances, storage space (disks, snapshots, images), and VM instance groups. The main area displays a table of VM instances. A pink box highlights the 'External IP' column for the first instance, which is 34.105.119.208. The table includes columns for Status, Name, Zone,建议 (Recommendation), User, Internal IP, External IP, and SSH. Below the table are several related operations: Explore Actifio GO, View Bill Report, Monitor VM, Set Firewall Rules, and Manage Patching. A pink box also highlights the 'Monitor VM' option.

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

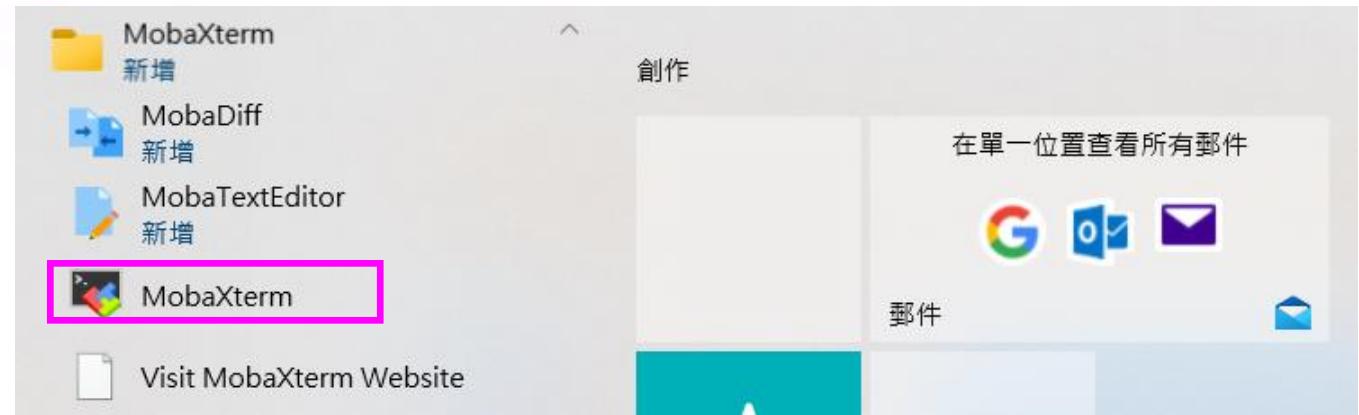
<https://cloud.google.com/> > 文件 > 運算 > Compute Engine > Guides: Connecting to instances > Connecting to Linux VMs using advanced methods > Windows(PuTTY)

# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

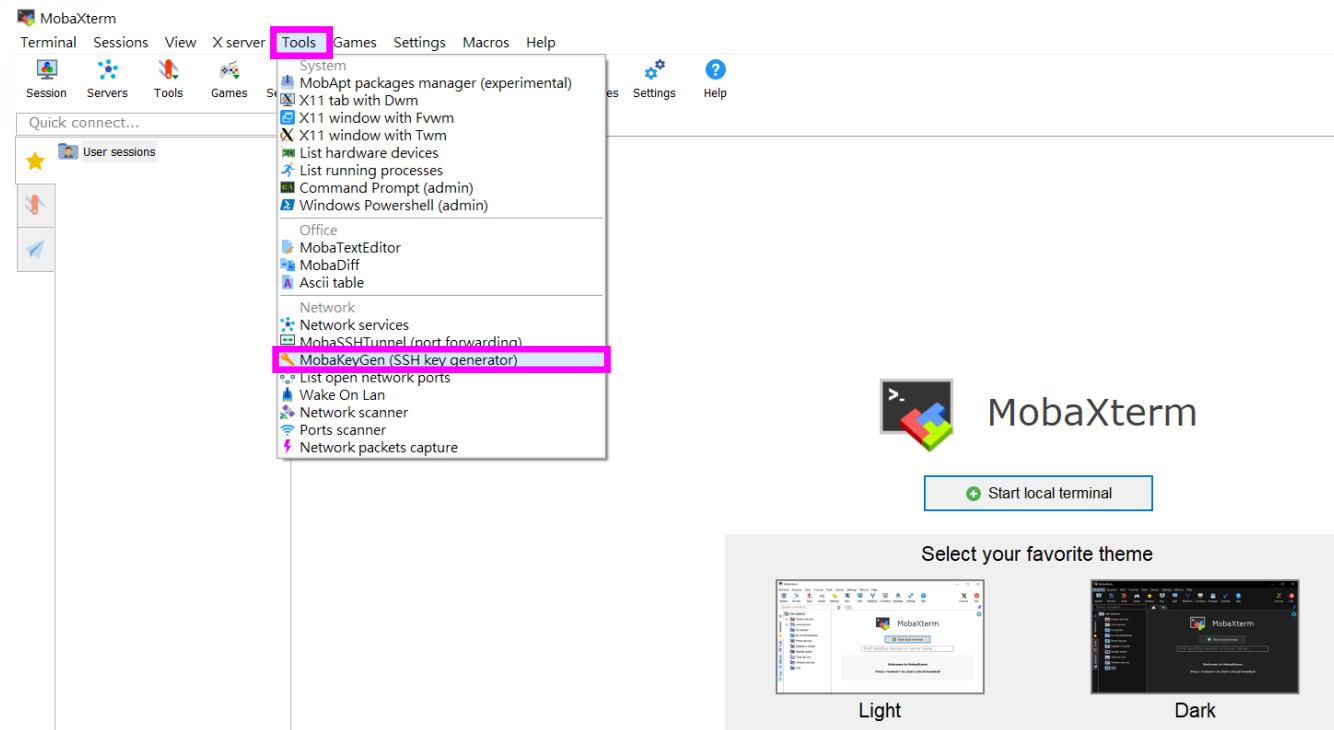
啟動 MobaXterm



# 架設 VM - GCP

## 5. 準備開發環境

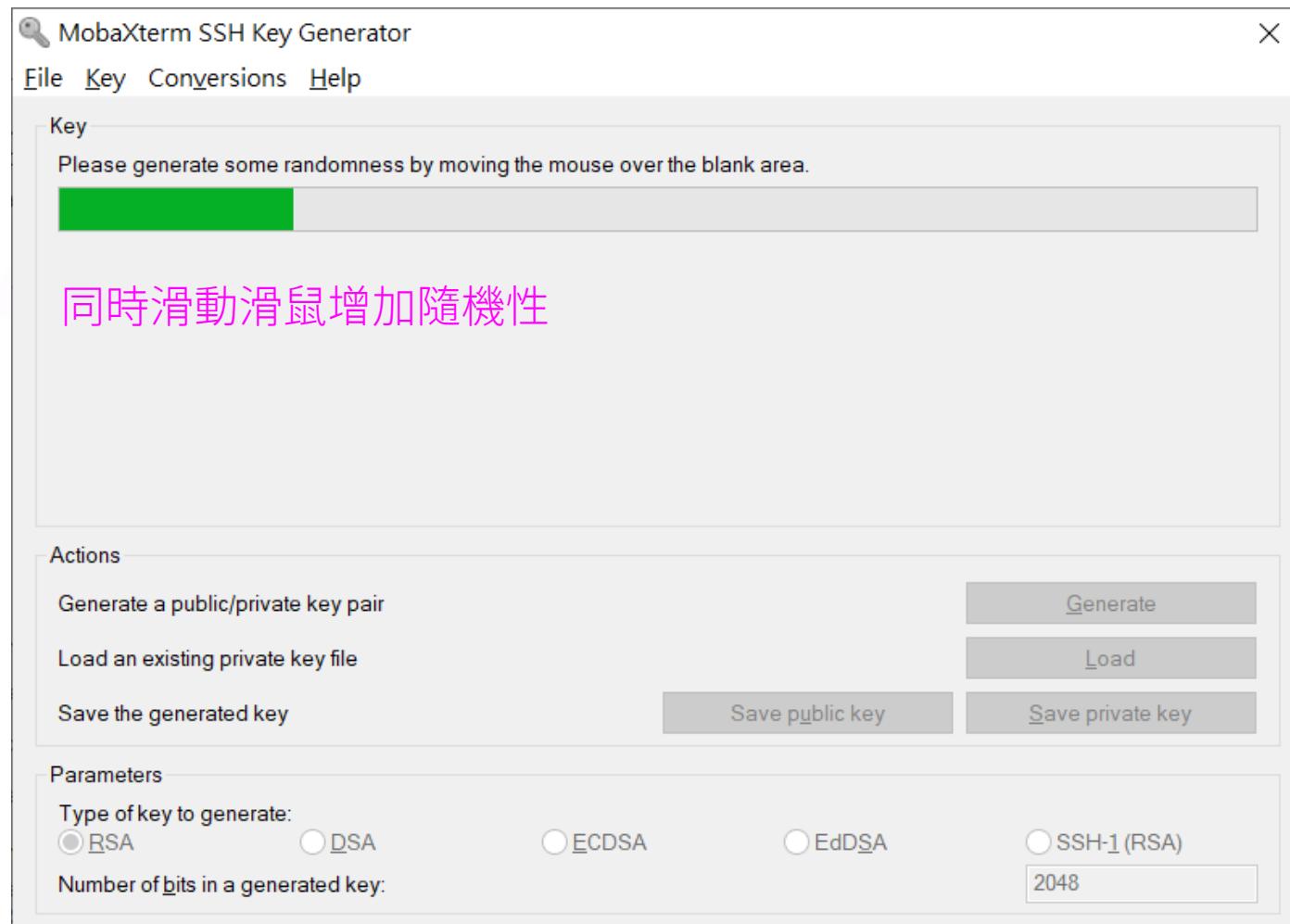
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

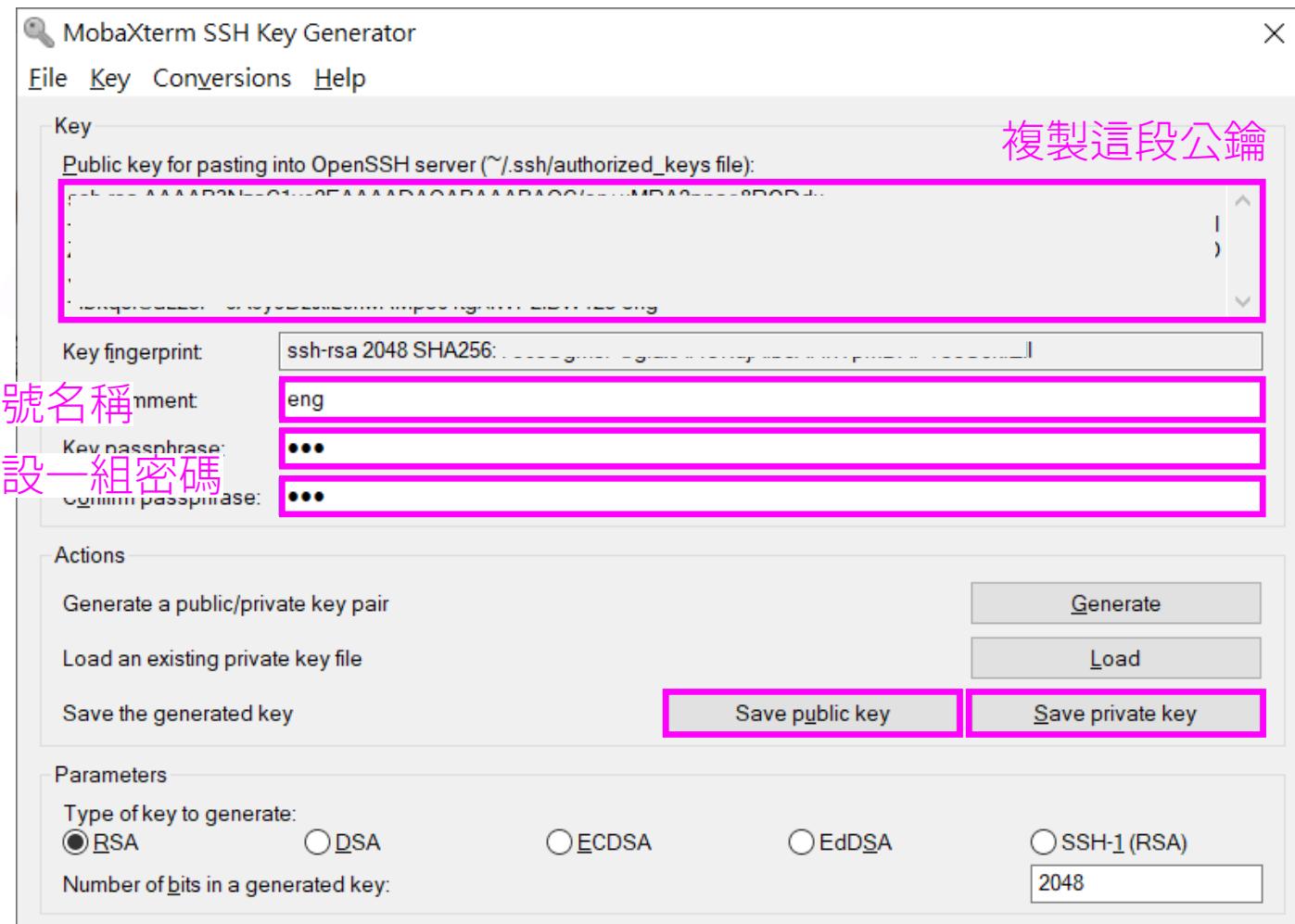
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10



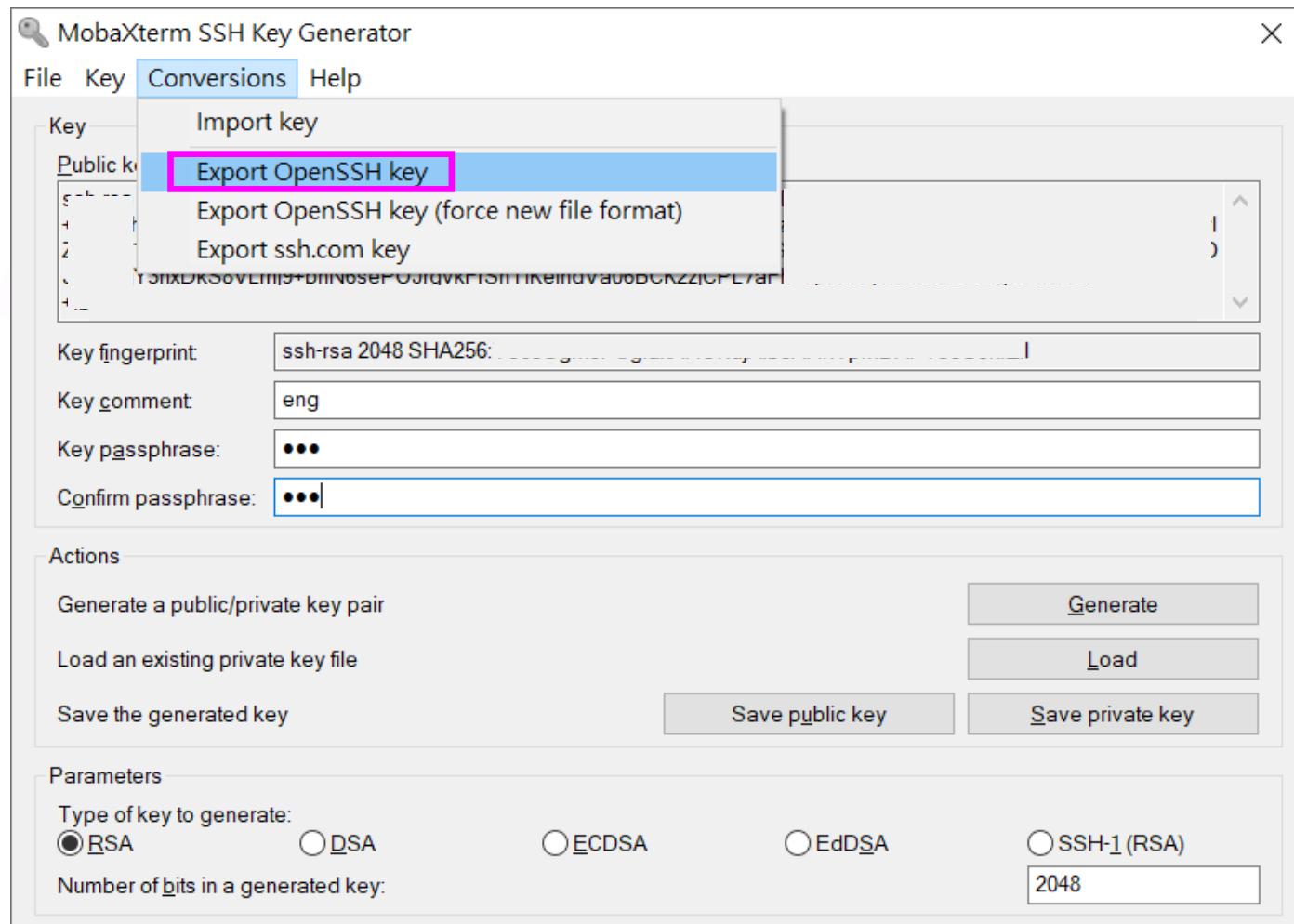
# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

#### Note

匯出 OpenSSH 格式的 Private Key 於 Cloud Shell 使用



# 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 search bar at the top center is set to '搜尋 產品、資源、文件 (/)'. Below it are several control buttons: '建立執行個體' (Create Instance), '匯入 VM' (Import VM), '重新整理' (Refresh), '開始/繼續' (Start/Resume), '停止' (Stop), '暫停' (Pause), and '重設' (Reset). To the right of these buttons is a '選取執行個體' (Select Instance) dropdown menu. At the bottom of the dashboard, there is a table showing existing VM instances. The first instance listed is 'us-west1-b', with details: 建議 (建议) 10.138.0.3, 內部 IP (Internal IP) (nic0), 外部 IP (External IP) 35.227.191.94 (nic0), and 連線 (Connections) SSH. Below the table are three cards: '查看帳單報表' (View Billing Report), '探索 VM 記錄檔' (Explore VM Log Files), and '管理修補程式' (Manage Patching). On the far left, a vertical navigation menu is open, showing various Google services like Cloud Storage, BigQuery, and Compute Engine. The 'Compute Engine' section is currently selected and expanded, showing options like '伺服器' (Server), '網路' (Network), '磁碟區' (Disk), 'NFS 共用' (NFS Share), '設定' (Settings), and '中繼資料' (Metadata). The 'Metadata' option is highlighted with a pink rectangle. At the bottom of the page is a URL bar with the address <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 sections for VM Management, Bare Metal Solutions, and Settings. Under Settings, the 'Cloud Storage' option is highlighted. The main content area displays the 'Security Keys' section, which contains a note about inheritance of security keys across all instances in the project. It features a globe icon and a 'New Security Key' button.

Google Cloud test ▾

Compute Engine 中繼資料 編輯 重新整理

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

中繼資料

這項專案中的所有執行個體都會繼承下列安全殼層金鑰。 [瞭解詳情](#)

中繼資料 安全殼層金鑰

安全殼層金鑰

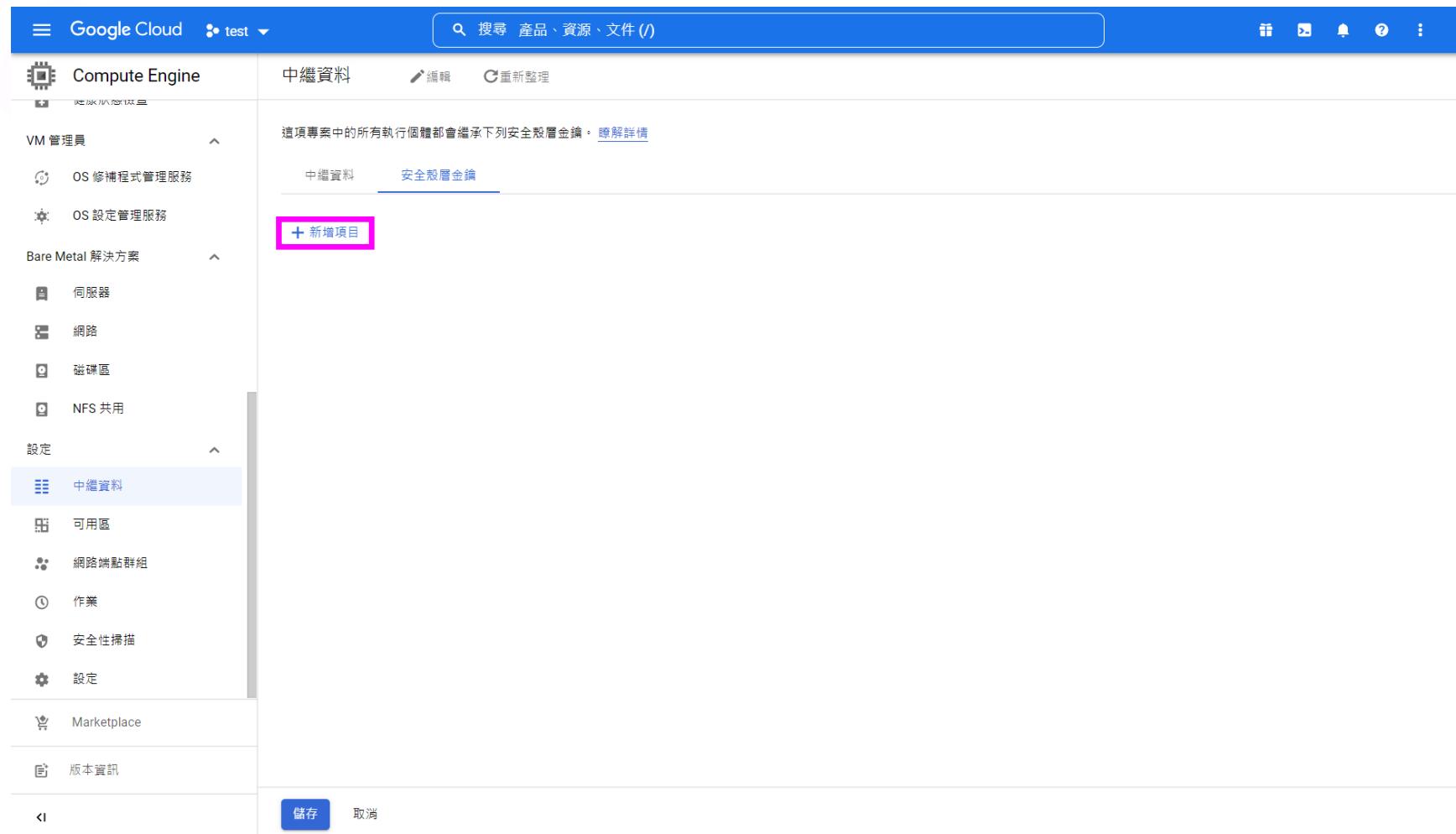
您可以儲存安全殼層金鑰，以便連線至專案中的 VM 執行個體。專案層級金鑰會散佈至「沒有」安全殼層金鑰的所有 VM 執行個體。

新增安全殼層金鑰

# 架設 VM - GCP

## 5. 準備開發環境

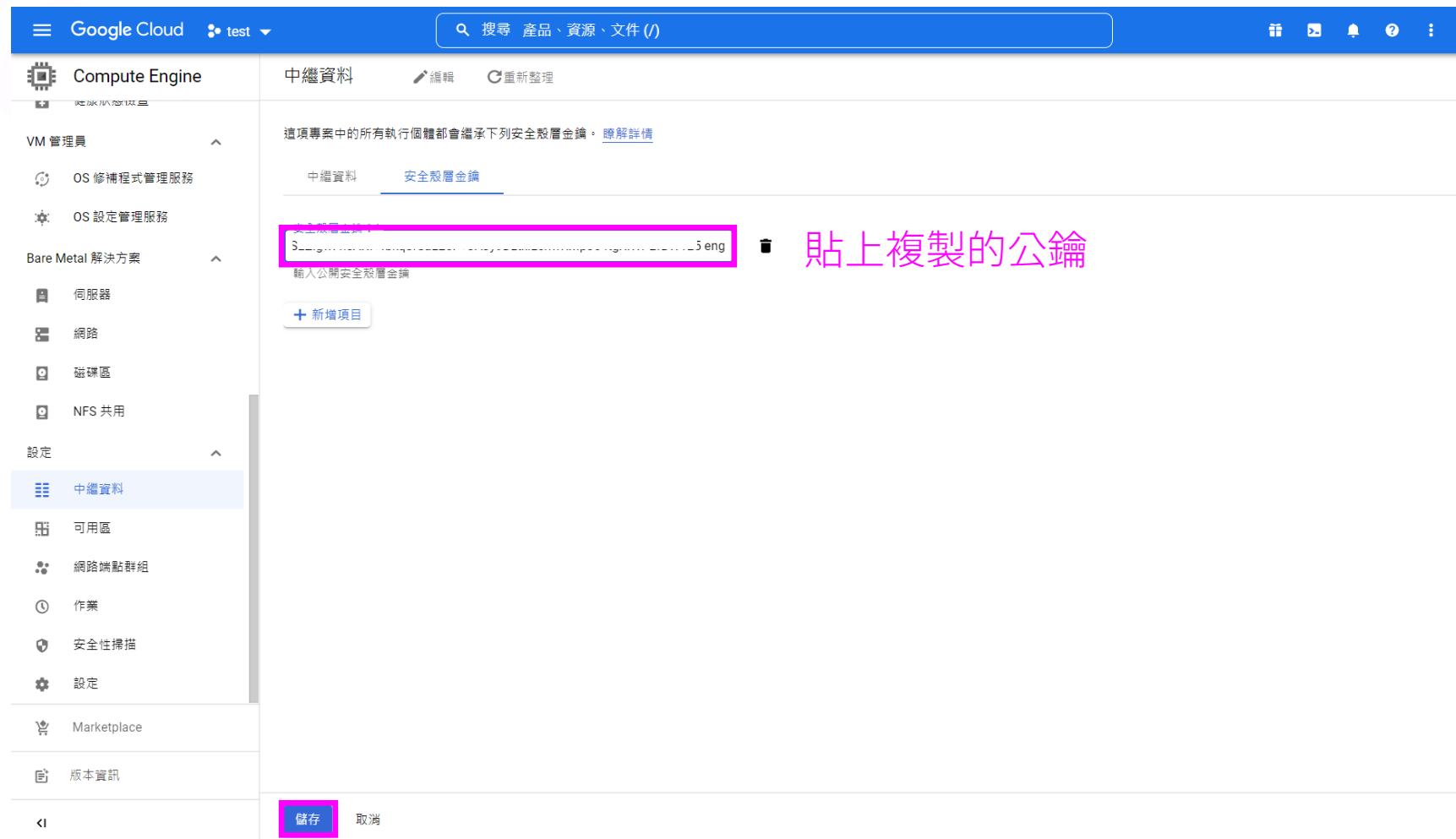
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

### a. Windows 10

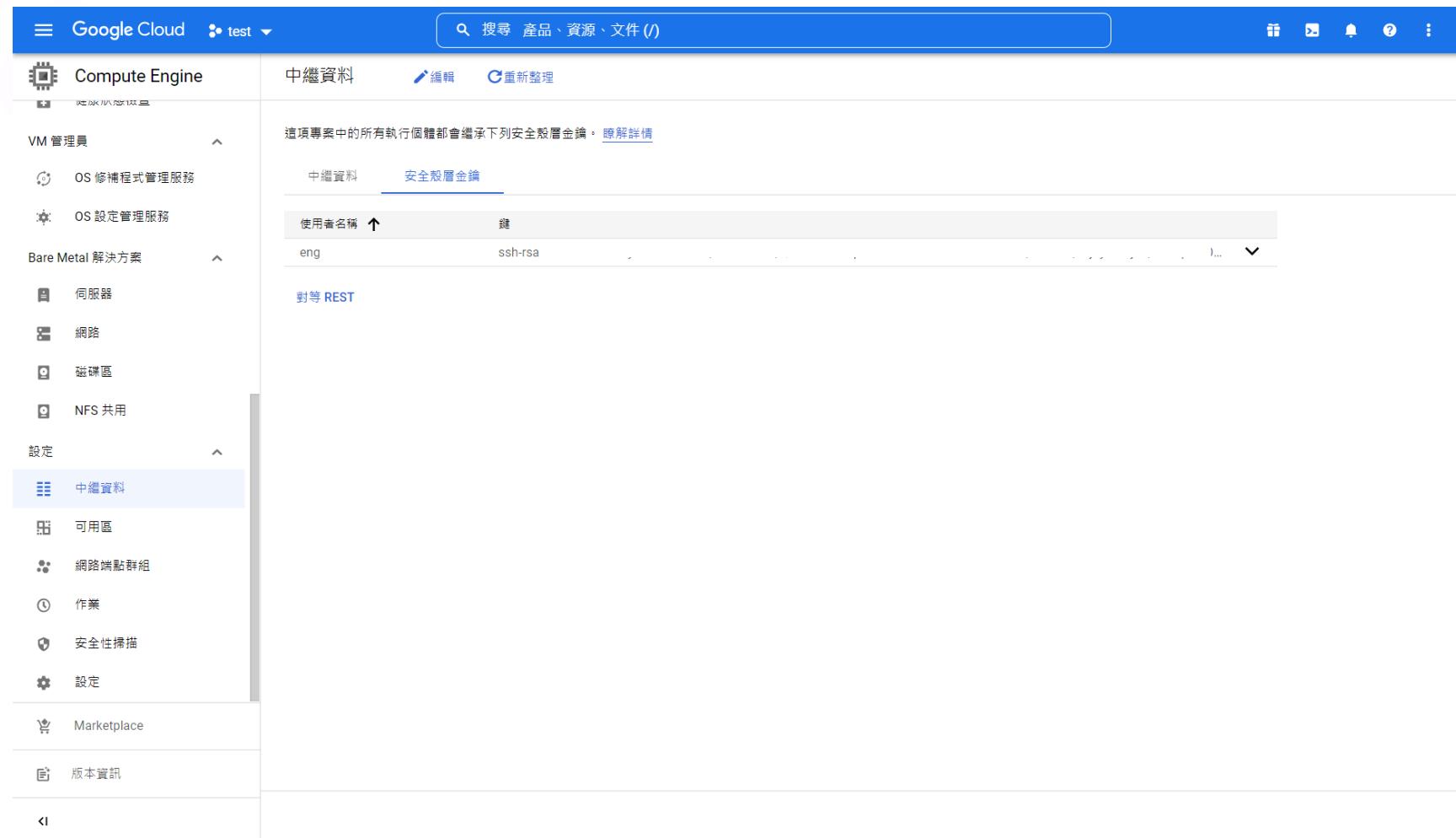


The screenshot shows the Google Cloud Compute Engine interface. On the left, the navigation pane includes sections for VM 管理員, Bare Metal 解決方案, 設定, and Marketplace. Under 設定, 中繼資料 is selected. In the main content area, the '安全殼層金鑰' tab is active, displaying a list of keys. One key, '3...eng', is highlighted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

# 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:

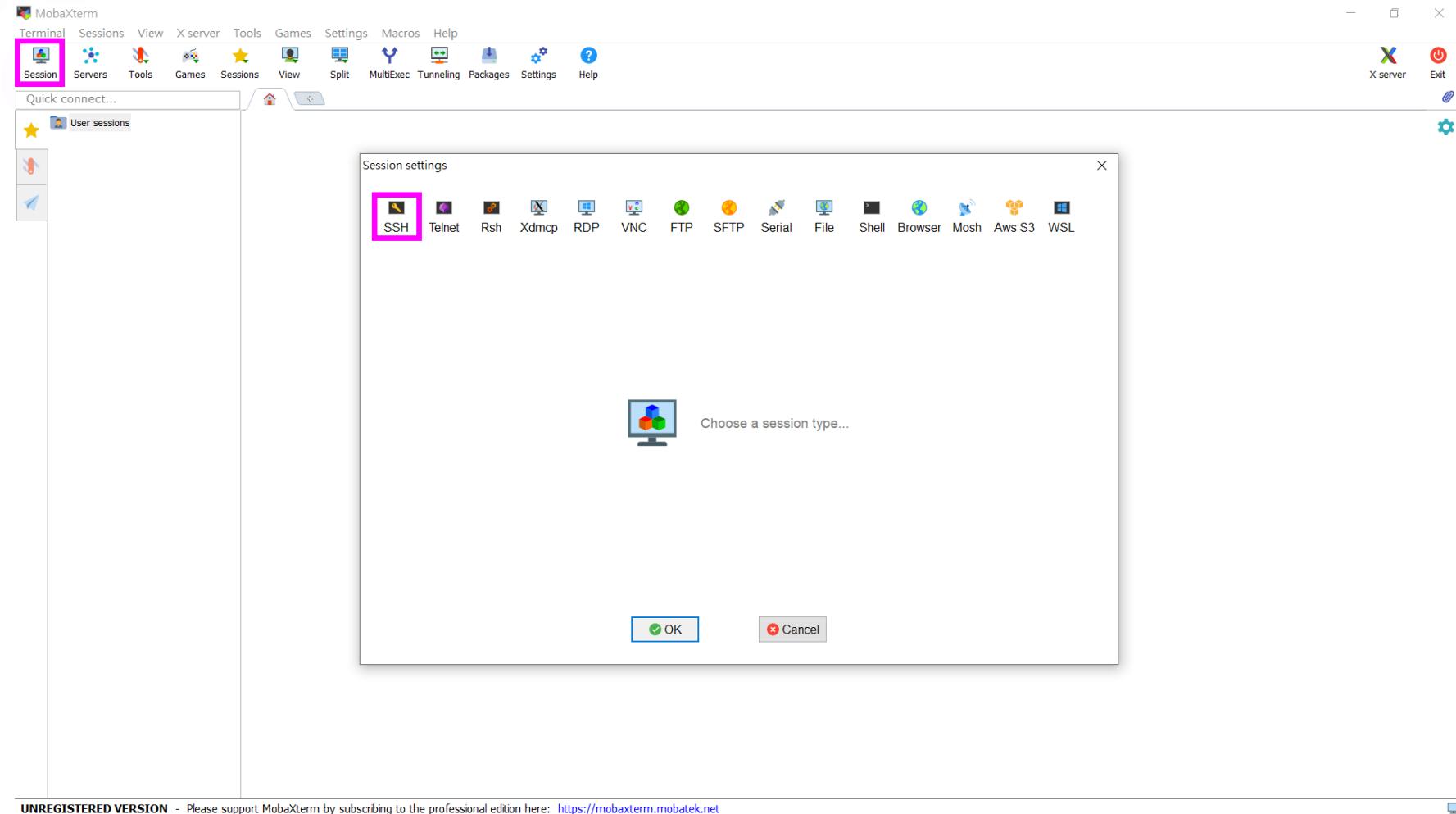
使用者名稱	鍵
eng	ssh-rsa

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

# 架設 VM - GCP

## 5. 準備開發環境

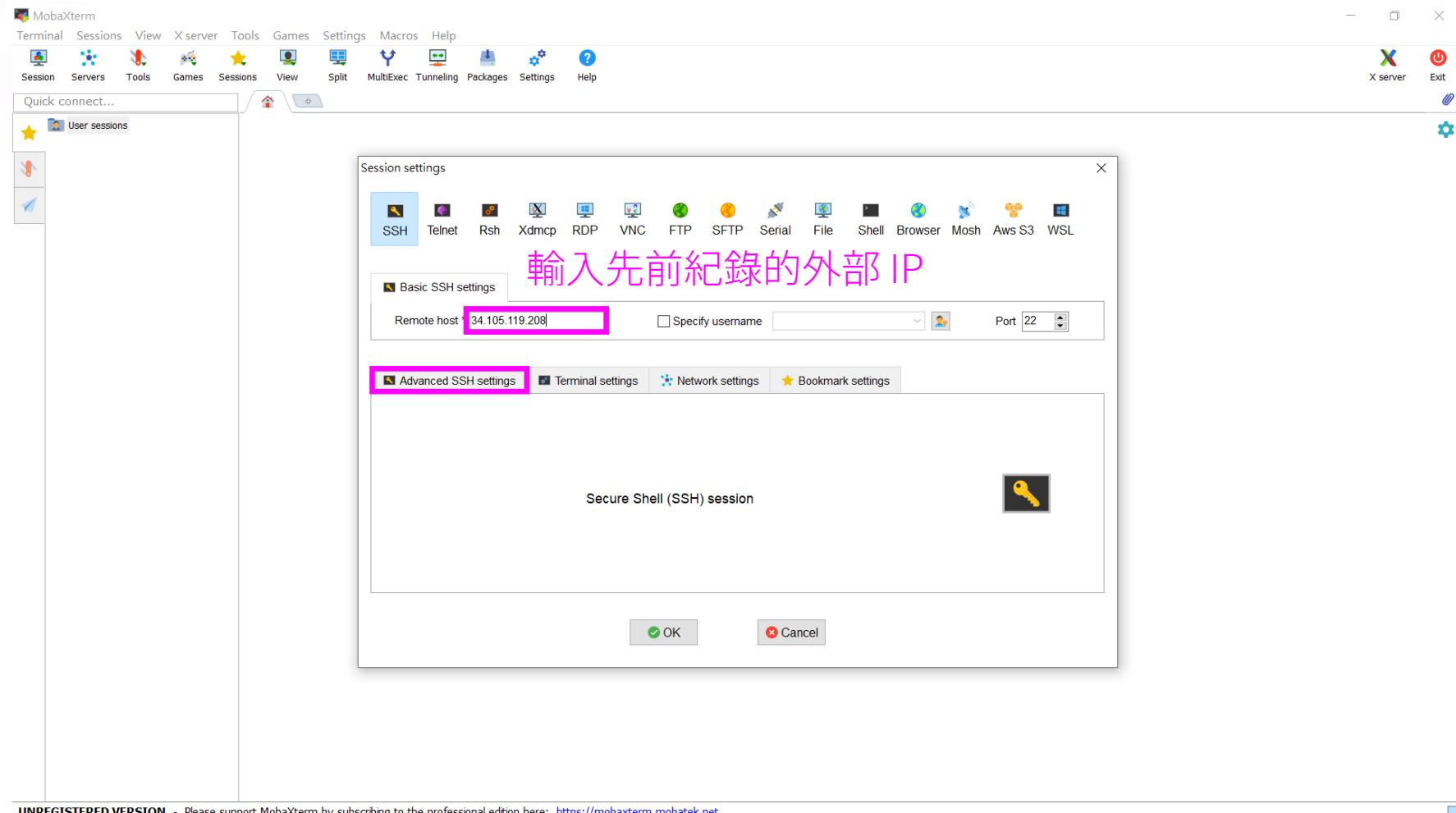
a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

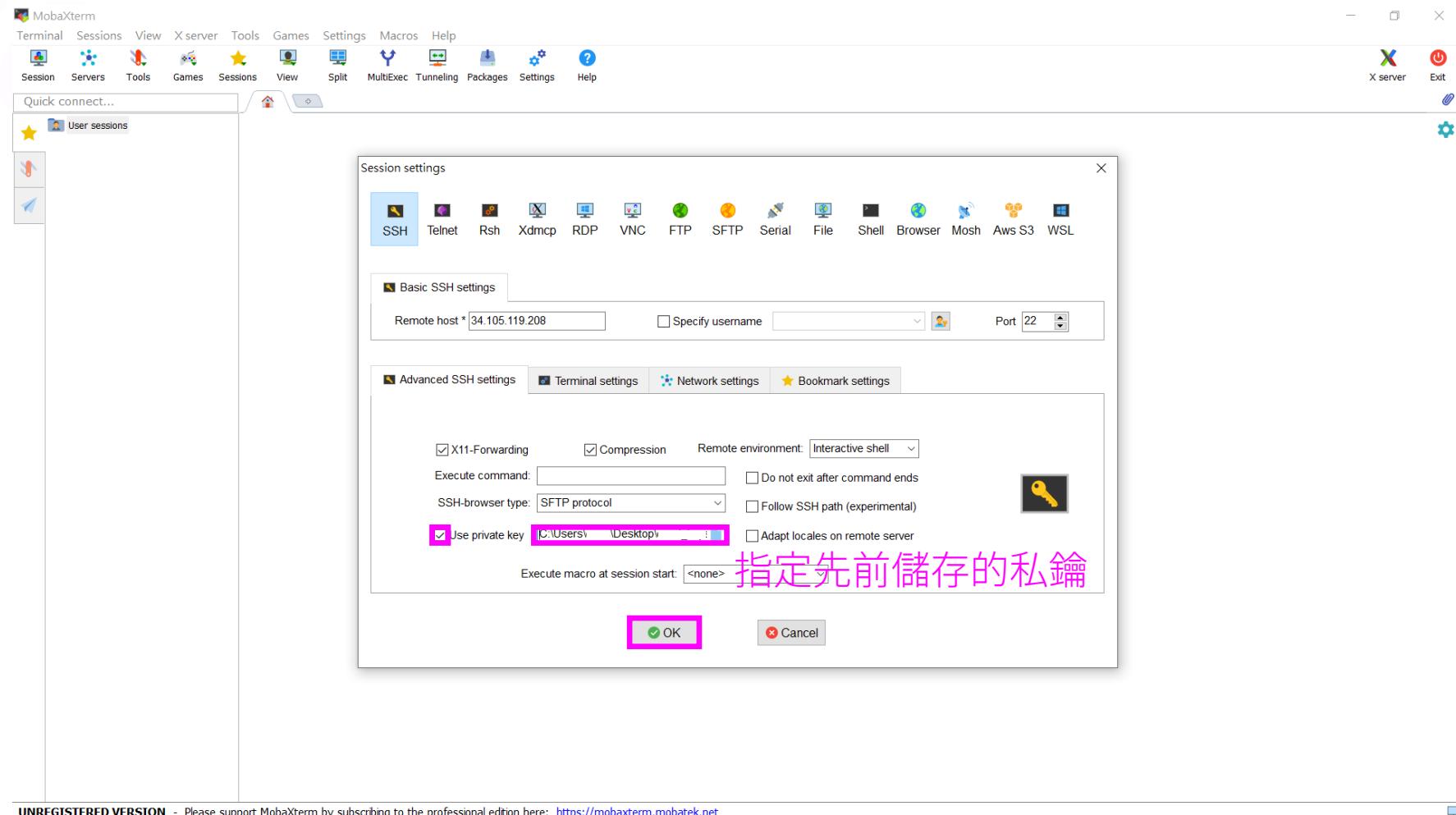
a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

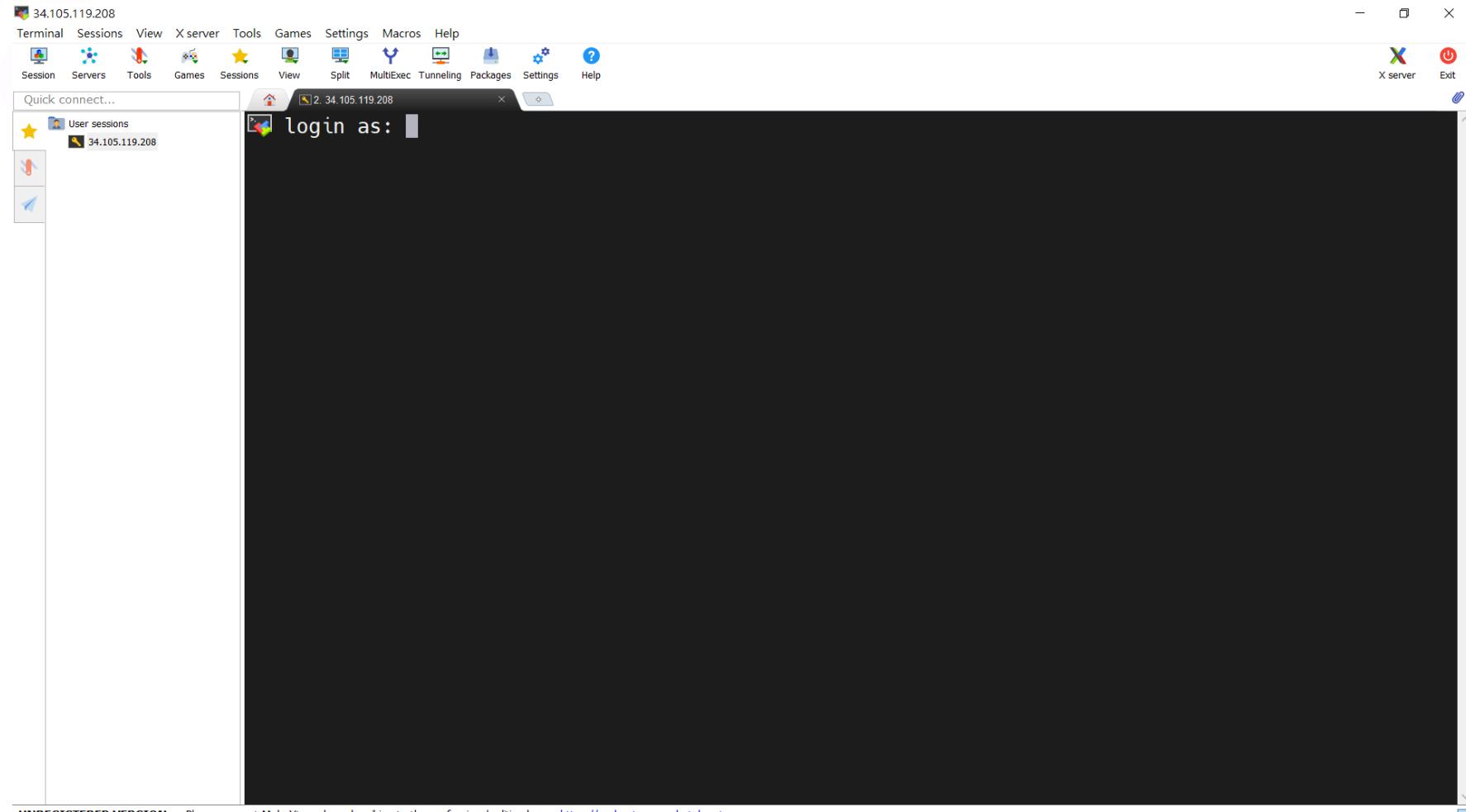
### a. Windows 10



# 架設 VM - GCP

## 5. 準備開發環境

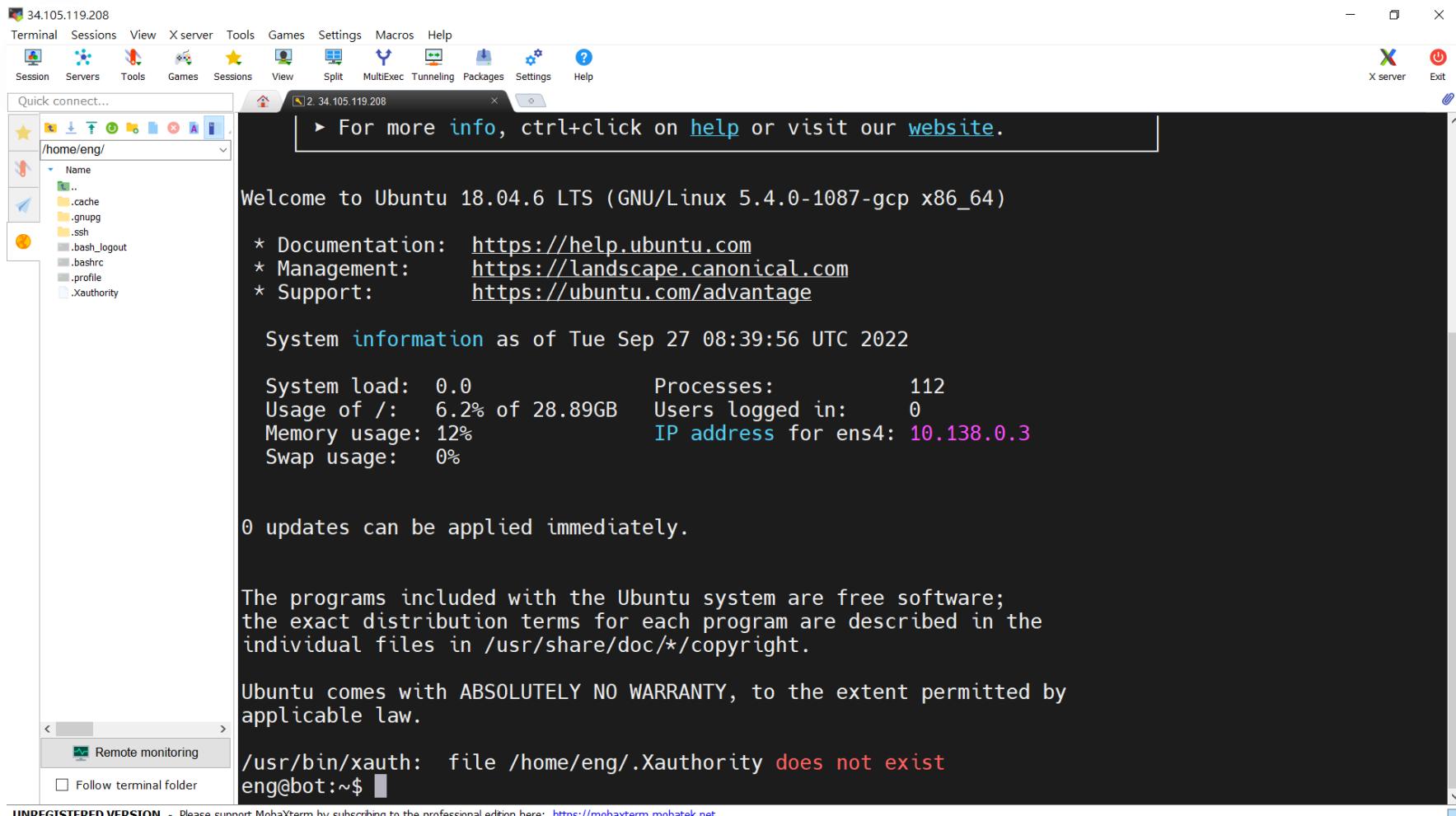
a. Windows 10



# 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 bot -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 Cloud VM 匯入公鑰

複製 *your\_keyfile.pub* 內容

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 Cloud VM

開啟 終端機

```
chmod 400 your_keyfile
```

```
chmod 400 eng
```

```
ssh -i your_keyfile your_account@your_vmiip
```

```
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 Cloud VM

開啟 終端機

```
scp -i your_keyfile your_account@your_vmid: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 management and help. Below the navigation bar is a section titled "Azure 服務" (Azure Services) featuring icons for creating resources, virtual machines, container instances, container registries, App Service, firewalls, application services, resource groups, and cognitive services, along with a "More services" link.

Under "Azure 服務", there's a "Resources" section. It has tabs for "Recent" (which is selected) and "Favorite". It lists resources by name, type, and last viewed time. A message says "No resources have been viewed recently" and includes a "View all resources" button.

Below the resources is a "瀏覽" (Browse) section with links for "訂用帳戶" (Subscription), "資源群組" (Resource Group), "所有資源" (All Resources), and "儀表板" (Dashboard).

At the bottom, there's a "工具" (Tools) section with links for "Microsoft Learn" (with a "Create" button), "Azure 監視器" (Monitoring), "適用於雲端的 Microsoft Defender" (Cloud-native Microsoft Defender), and "成本管理" (Cost Management).

A footer at the bottom left shows 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 title is "Azure 服務". On the left, there's a sidebar with "建立資源" (Create Resource) and "虛擬機器" (Virtual Machine). 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 learn about 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

Annotations on the right side of the screen provide instructions:

- 設定主機名稱** (Set Hostname)
- 選擇區域** (Select Region)
- 選擇作業系統** (Select Operating System)

At the bottom of the wizard, there are buttons for 'Search + Create' and 'Previous Step: Disks'.

# 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

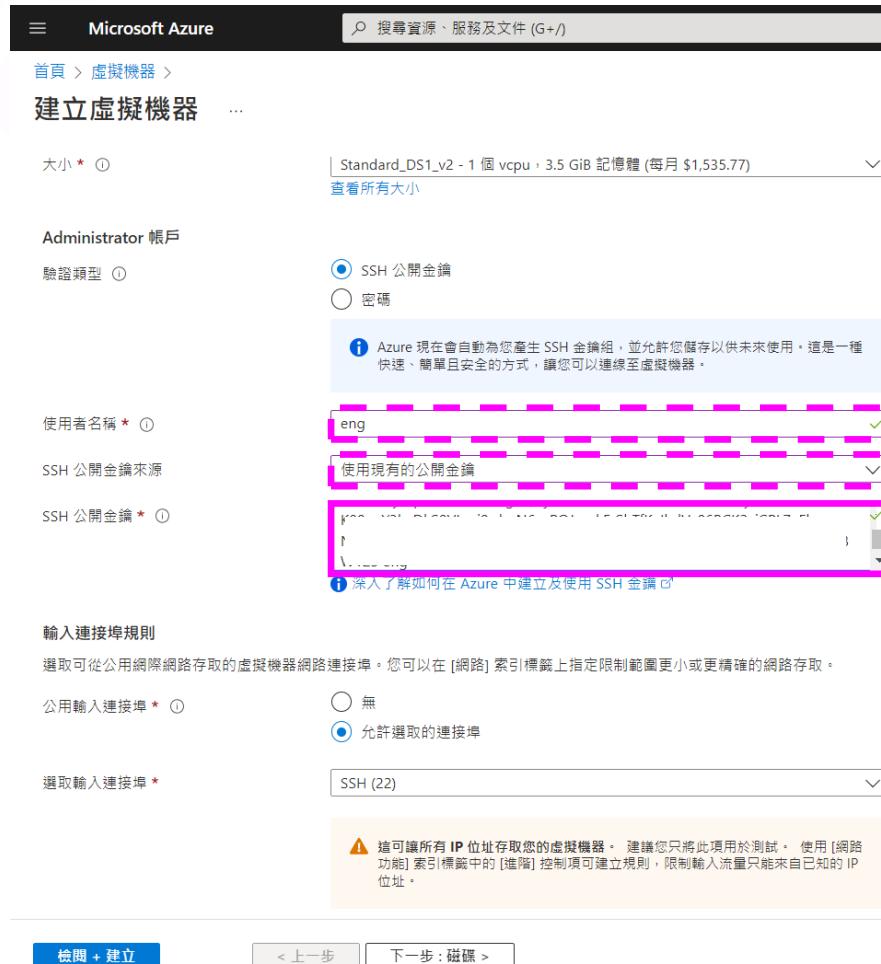
< 上一步

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: 檢閱 + 建立 (Review + Create) and < 上一步 : 磁碟 > (Previous Step: Disk).

# 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, and various icons. 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. Use the [Pricing calculator](#) to meet all your pricing needs.)

**PRODUCT DETAILS**

1 X Standard DS1 v2  
by Microsoft  
[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ  
**2.1038 TWD/hr**  
[Pricing for other VM sizes](#)

**TERMS**

By clicking "建立", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

**警告** 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。

底部有 '建立' (Create) 按钮，以及 '上一步' (Previous)、'下一步' (Next) 和 '下載自動化的範本' (Download template) 按钮。右下角有 '提供意見反應' (Provide feedback) 图标。

# 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). To the right of the main content, 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 settings. 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, detailed tabs provide information about the VM's configuration, such as its size (Standard DS1 v2), operating system (Linux ubuntu 18.04), and disk details.

首頁 > CreateVm-Canonical.UbuntuServer-18\_04-Its-gen2-20220927230325 | 概觀

bot 虛擬機器

搜尋

連接 啟動 重新啟動 停止 指取 刪除 重新整理 在行動裝置中開啟 CLI / PS 意見反應

JSON 檢視

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 - 賽手

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

作業系統 : Linux (ubuntu 18.04)

大小 : Standard DS1 v2 (1 vCPU, 3.5 GiB 記憶體)

公用 IP 位址 : 20.245.150.101

虛擬網路/子網路 : bot\_group\_09272303-vnet/default

DNS 名稱 : 未設定

記錄外部 IP

屬性 監視 功能 (7) 建議 教學課程

虛擬機器

電腦名稱 : bot

健全狀態 : -

作業系統 : Linux (ubuntu 18.04)

發行者 : Canonical

供應項目 : UbuntuServer

方案 : 18\_04-Its-gen2

VM 世代 : V2

VM 架構 : x64

代理程式狀態 : Ready

代理程式版本 : 2.8.0.11

主機群組 : 無

主機 : -

鄰近放置群組 : -

網路

公用 IP 位址 : 20.245.150.101

公用 IP 位址 (IPv6) : -

私人 IP 位址 : 10.1.0.4

私人 IP 位址 (IPv6) : -

虛擬網路/子網路 : bot\_group\_09272303-vnet/default

DNS 名稱 : 設定

大小

大小 : Standard DS1 v2

vCPU : 1

RAM : 3.5 GiB

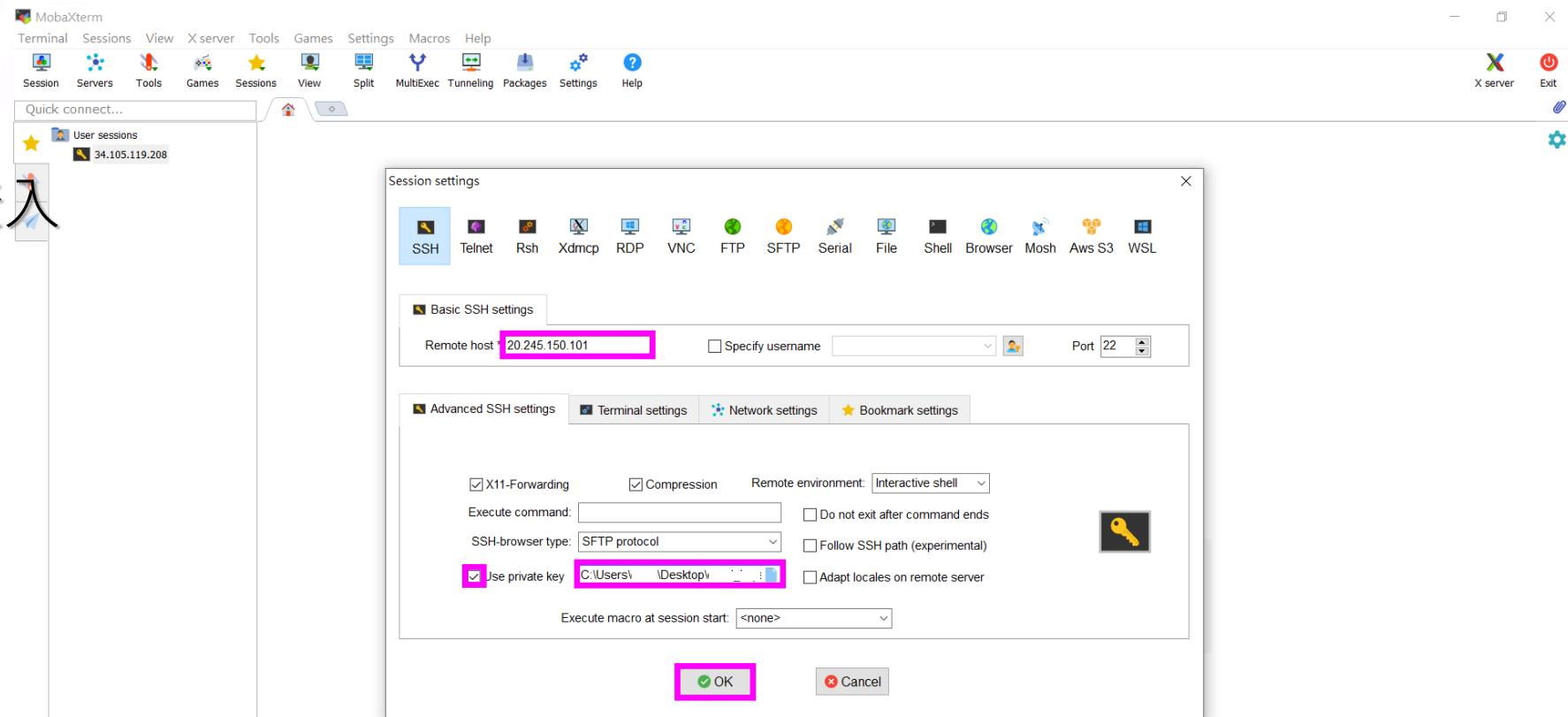
磁碟

作業系統磁碟 : bot\_OsDisk\_1\_a0ac7177ec95420e92af4a05c6442852

# 架設 VM - Azure

## 3. 準備開發環境

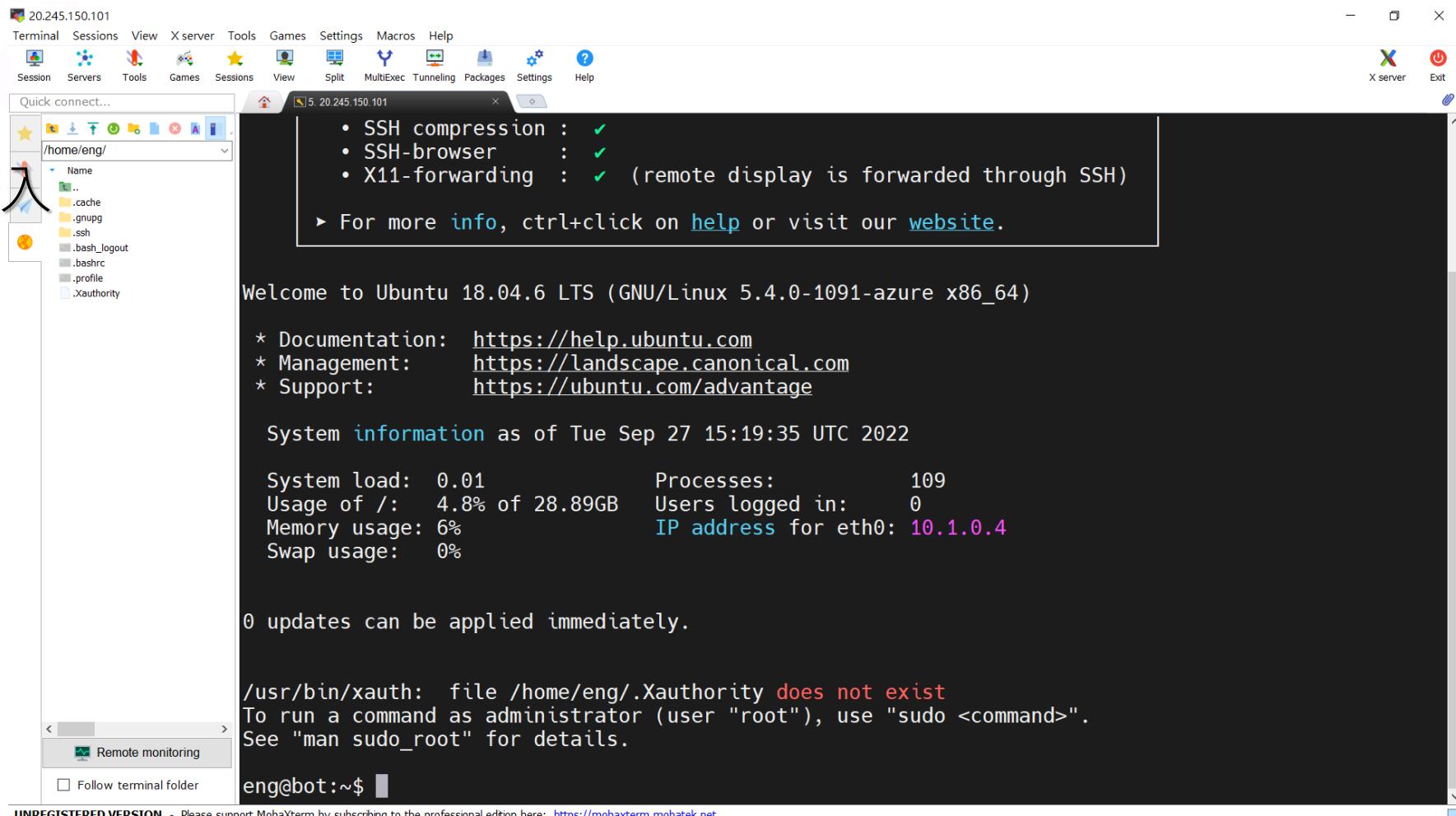
以 MobaXterm 登入



# 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.pem eng@20.245.150.101
```

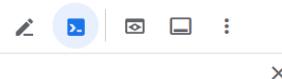
Usage of /:	4.9% of 28.8GB	Processes: 102
Memory usage: 5%		Users logged in: 0
Swap usage: 0%		IP address for eth0: 10.1.0.4

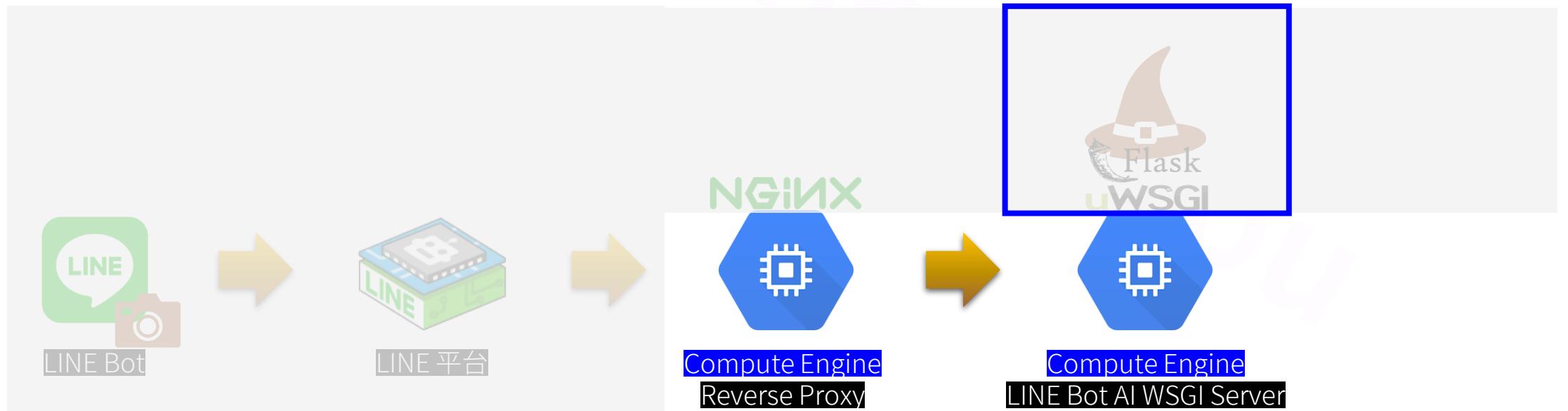
```
0 updates can be applied immediately.
```

```
New release '20.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.
```

```
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

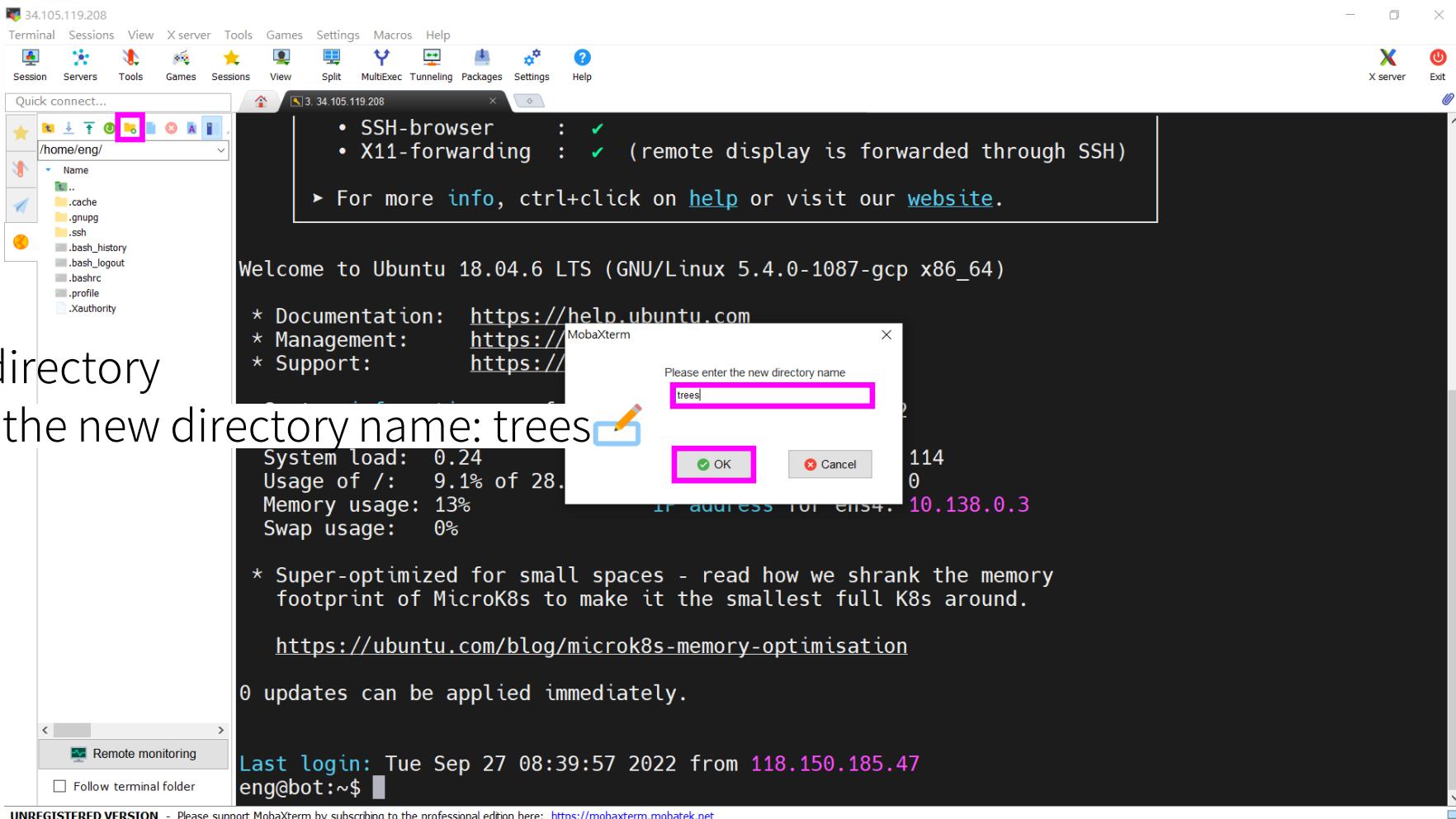
## 1. 準備程式碼

### 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: trees
- ③ OK

#### Note

相當於下列指令

`mkdir your_project`

`mkdir trees`

The screenshot shows a MobaXterm window titled '34.105.119.208'. The terminal window displays the following text:

```
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 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:~$
```

At the bottom of the terminal window, there is a note: "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

### c. 於 VM 建立專案目錄

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

The screenshot shows a MobaXterm window titled '34.105.119.208'. The terminal window displays the following text:

```
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 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:~$
```

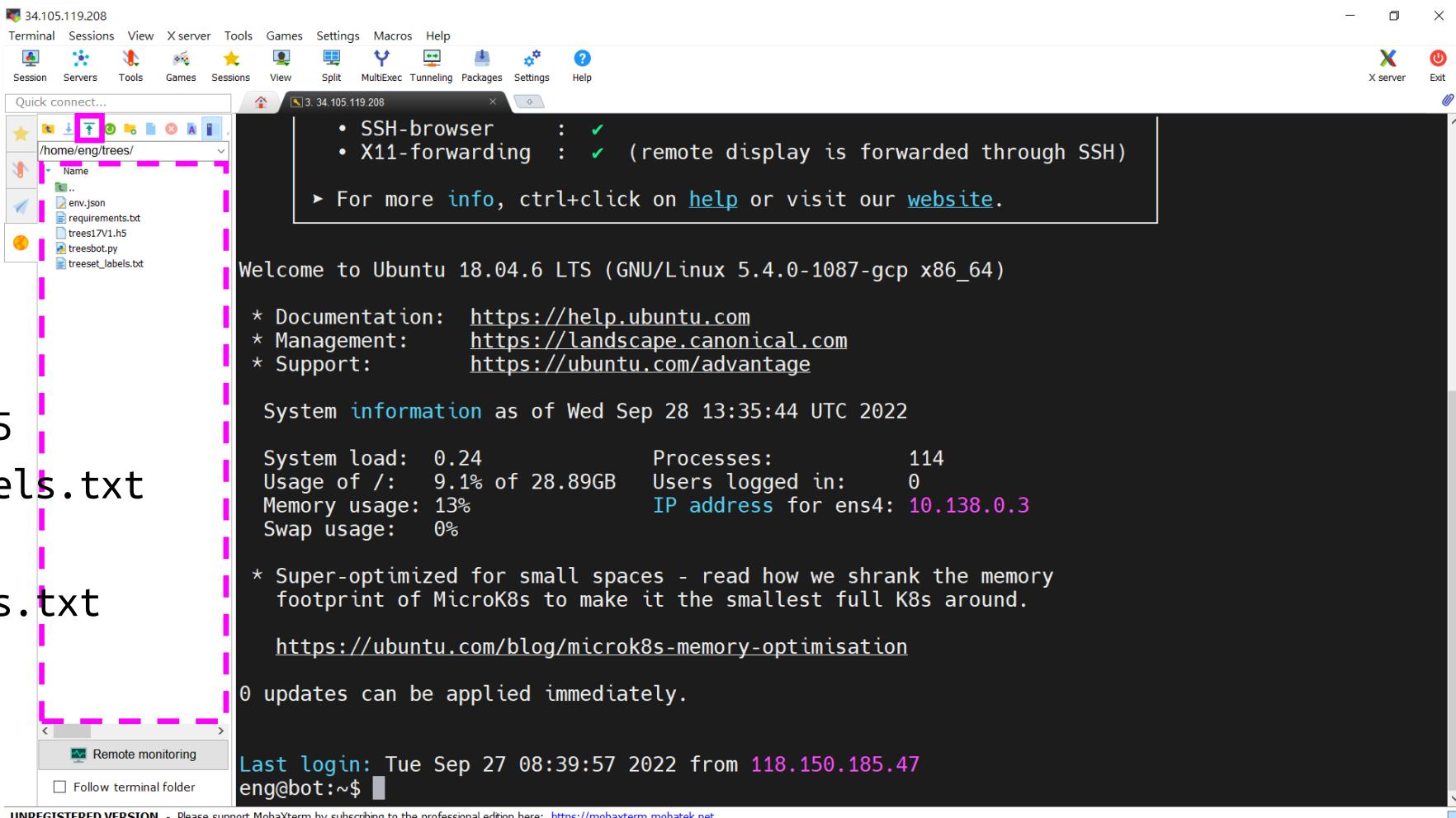
At the bottom of the terminal window, there is a note: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

# 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



```
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

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%
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

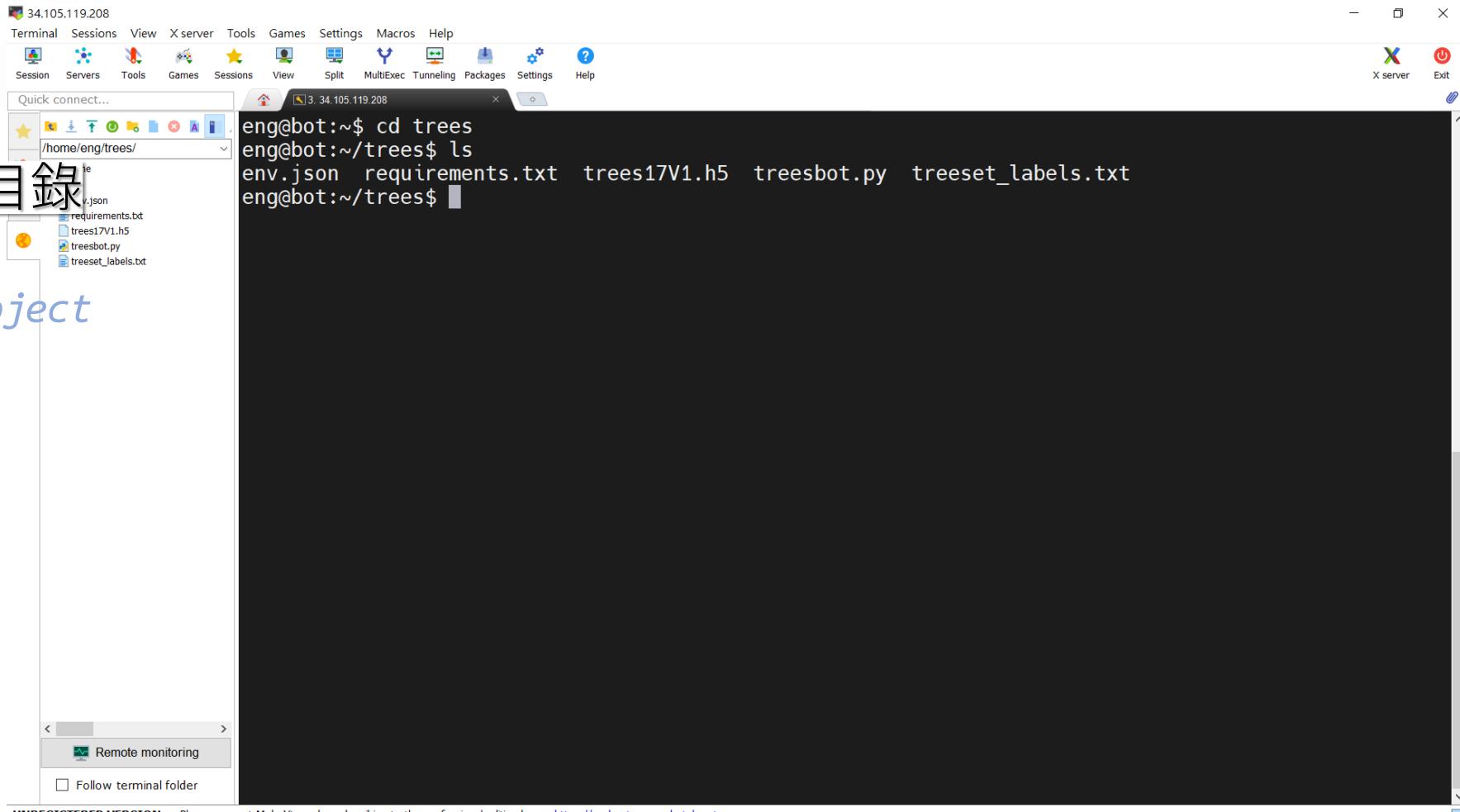
## 2. 部署服務

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>

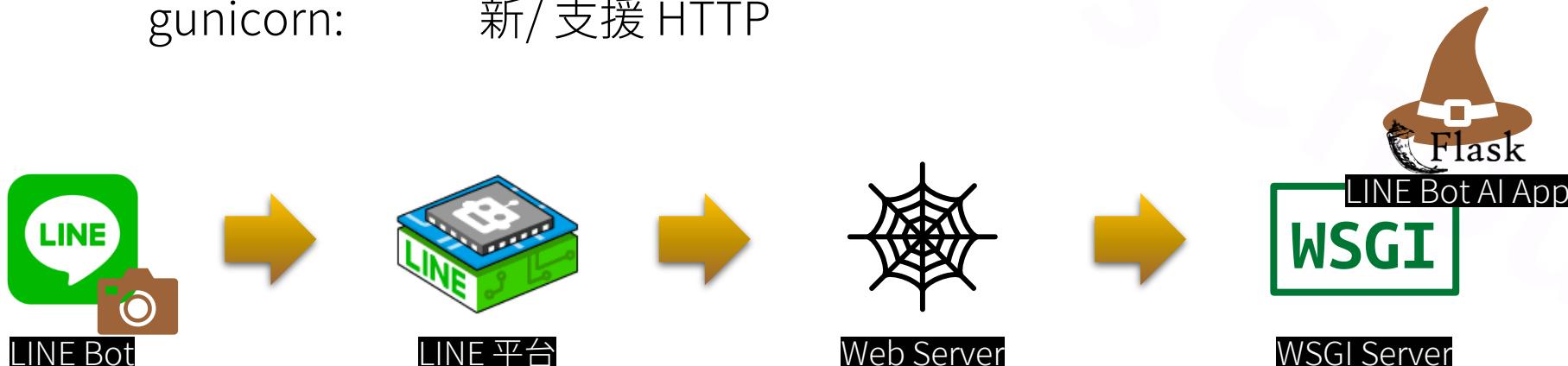
# LINE Bot & WSGI

## WSGI Server 的選擇

uWSGI vs gunicorn

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

gunicorn: 新/ 支援 HTTP



# LINE Bot & WSGI

## WSGI Server 的選擇

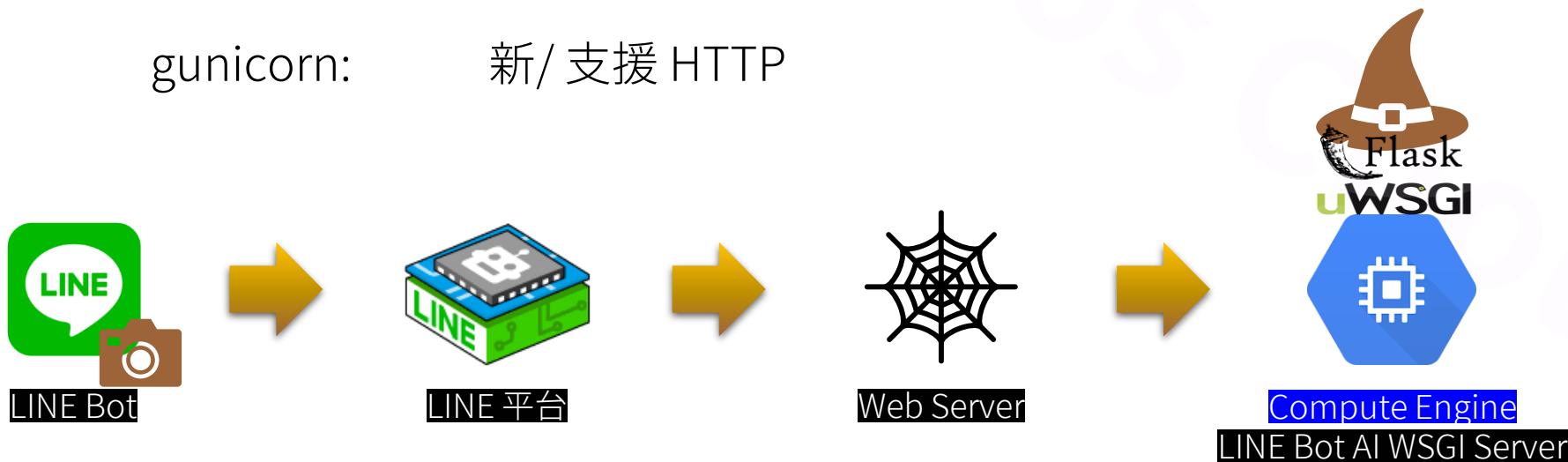
uWSGI vs gunicorn



uWSGI: 經典/ 支援 HTTP/ 可搭配 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)

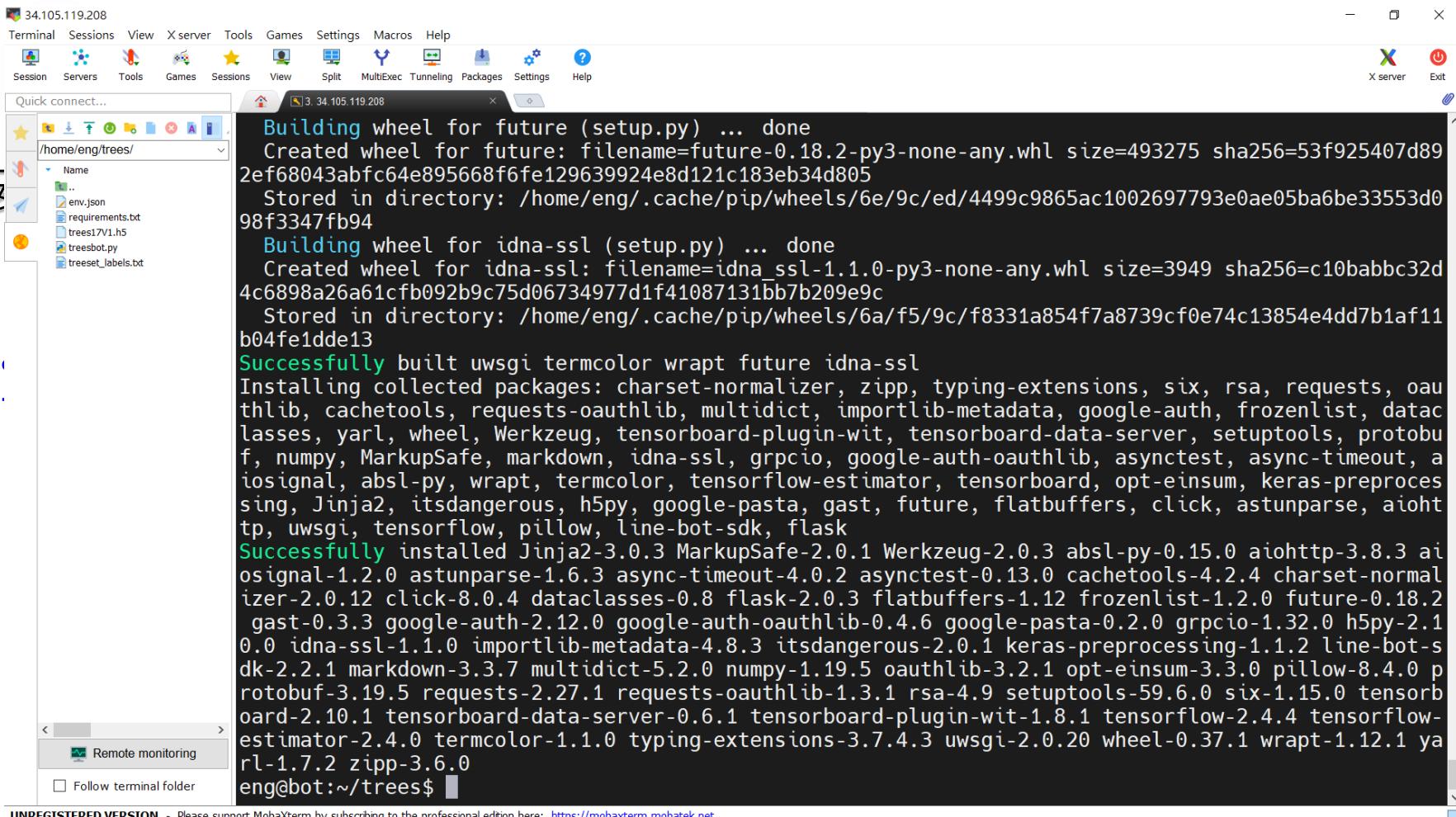
```
sudo apt update; sudo apt install -y python3-pip; pip3 install --upgrade  
pip; sudo timedatectl set-timezone Asia/Taipei; python3 -m pip install -r  
requirements.txt --no-warn-script-location; source ../../profile
```

# 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>'.

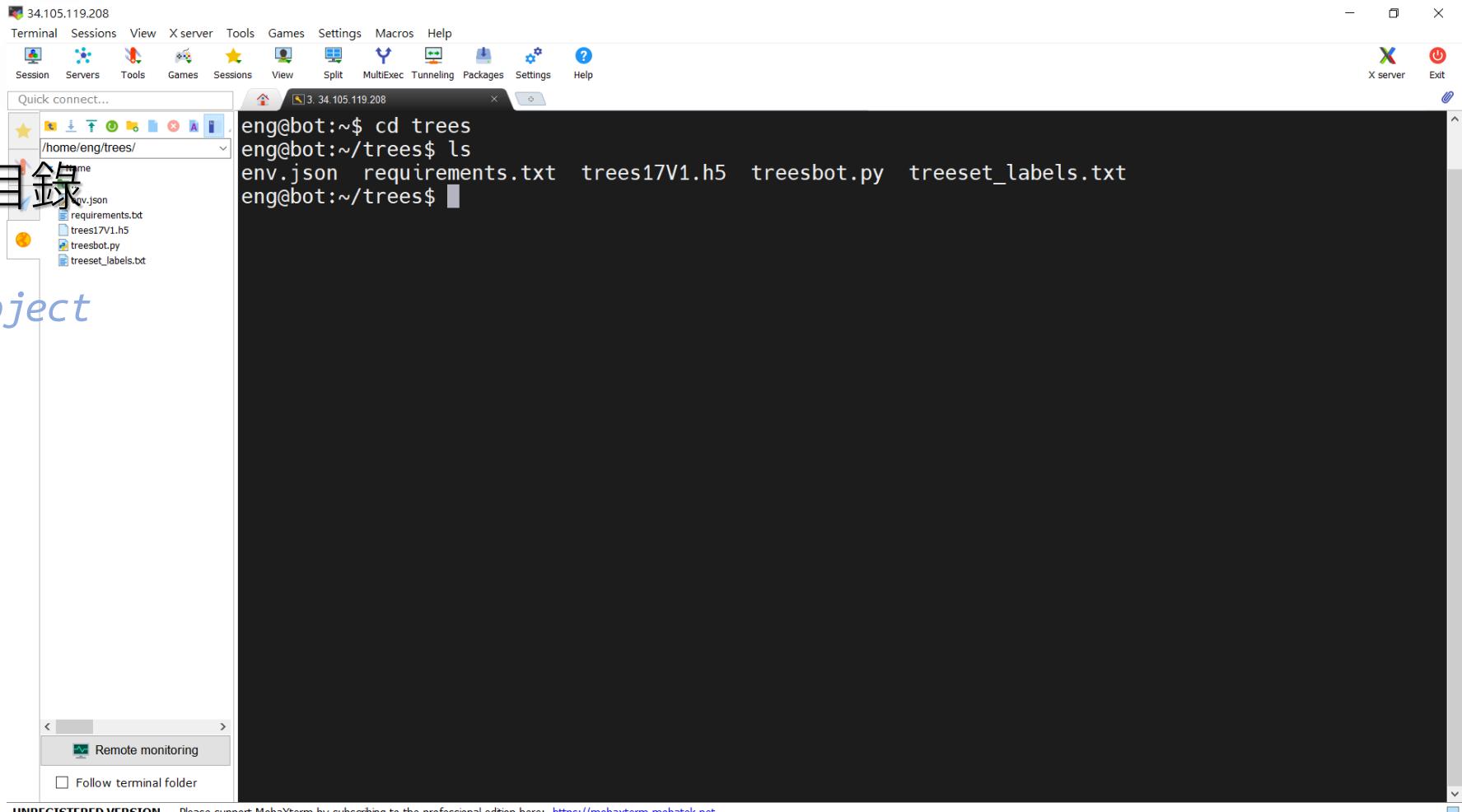
# LINE Bot & WSGI

## 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>

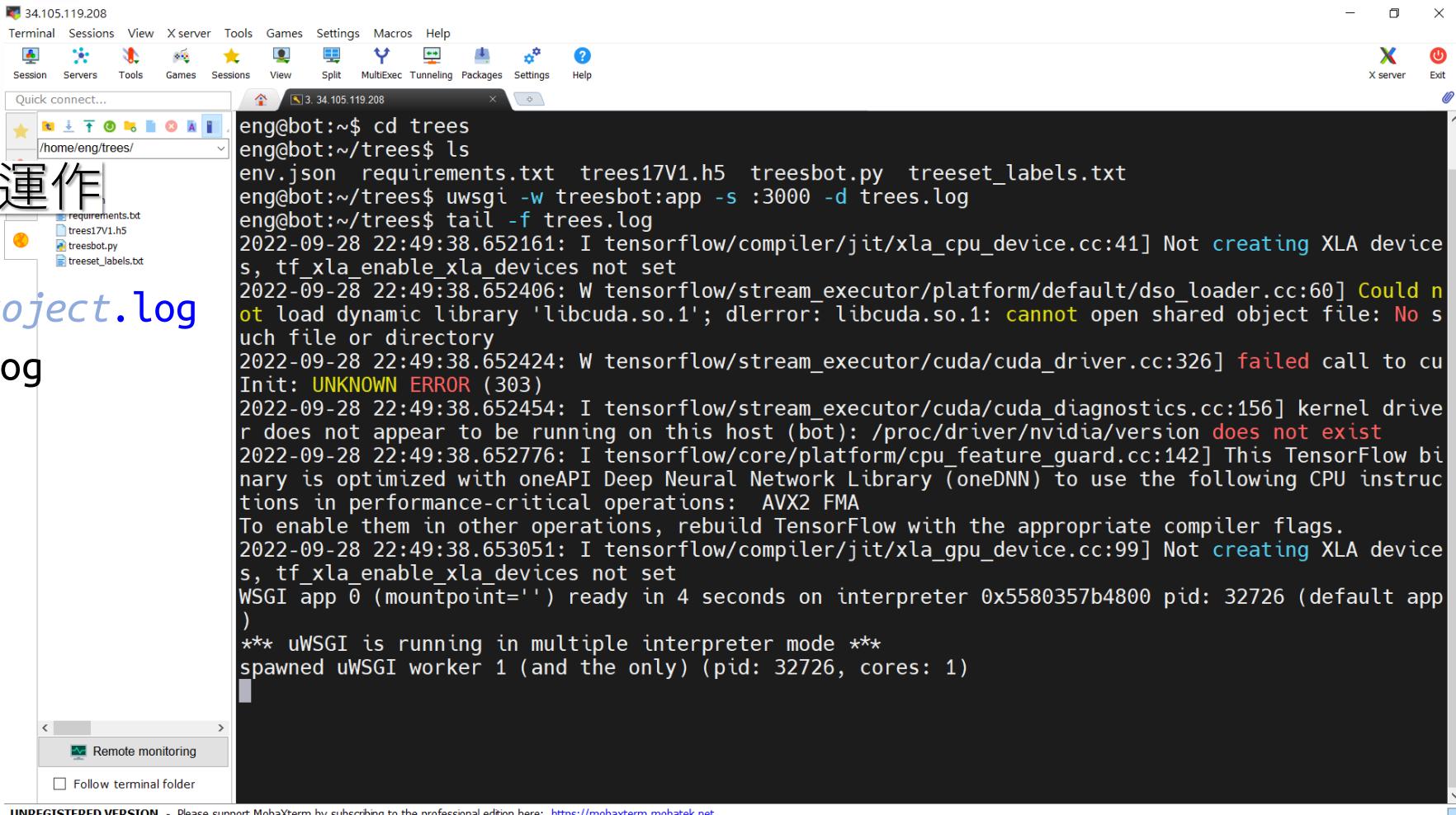
# LINE Bot & WSGI

## 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>

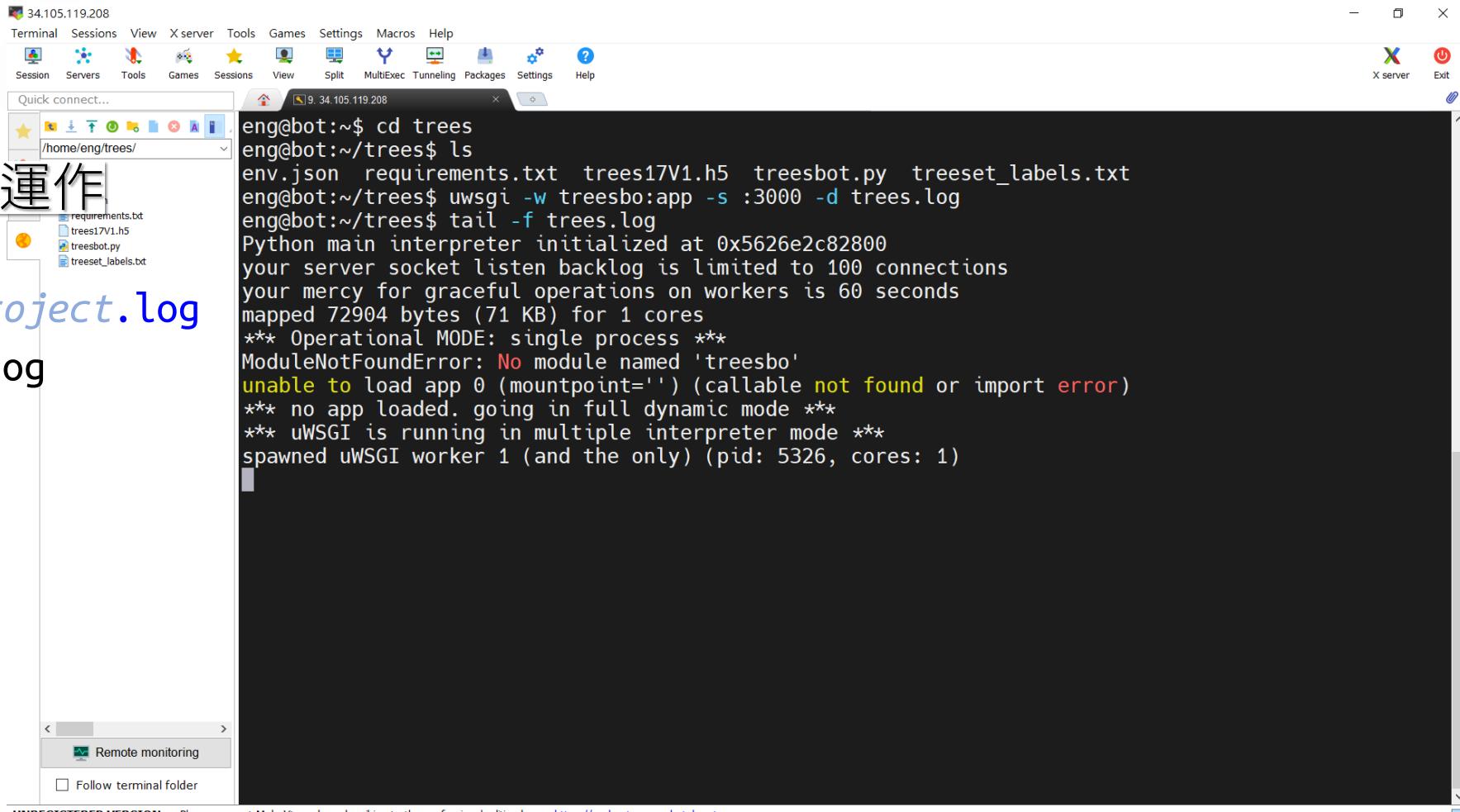
# LINE Bot & WSGI

## 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 terminal window in MobaXterm with the following command and 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 interface includes a file browser sidebar and various MobaXterm menu and toolbars.

# LINE Bot & WSGI

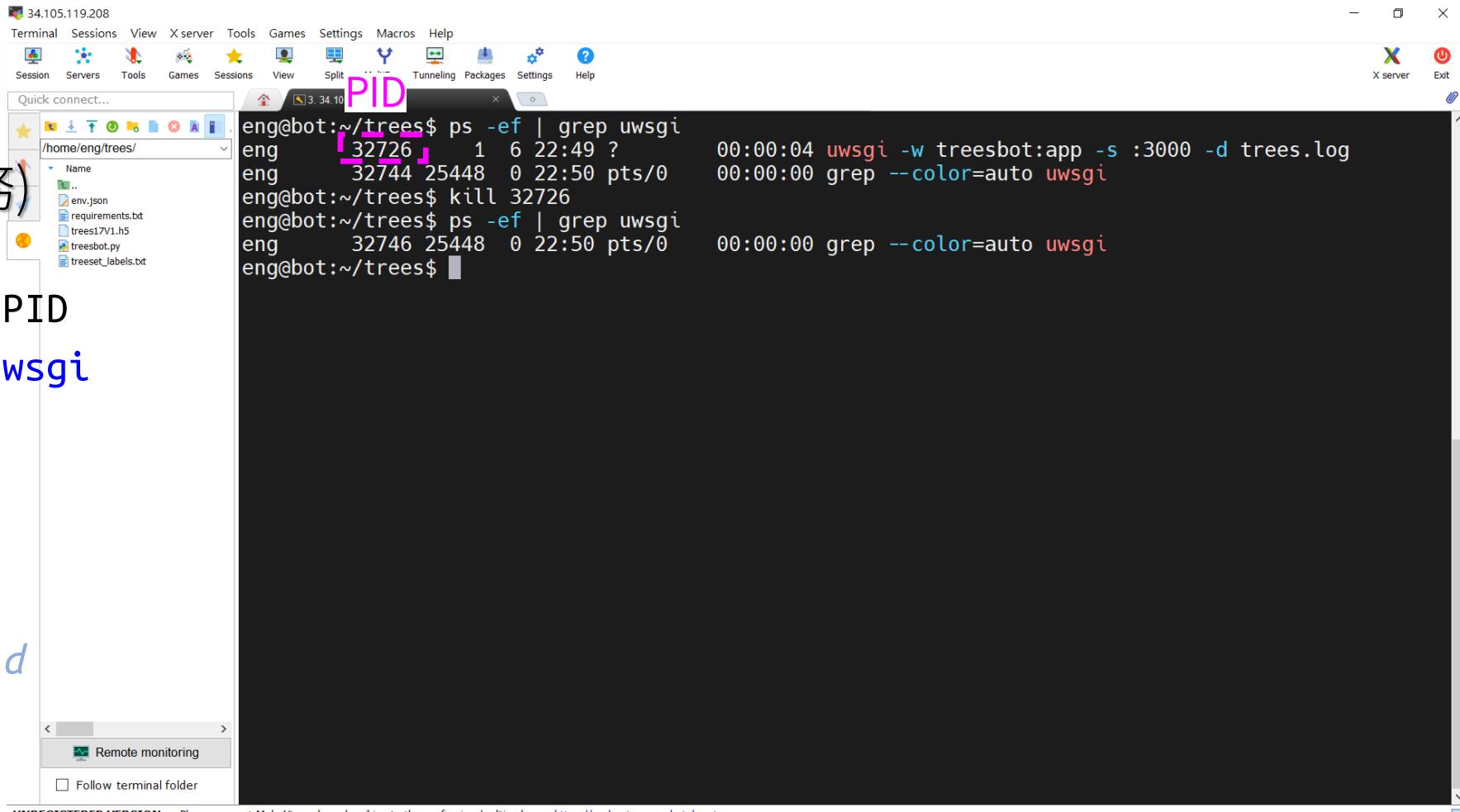
## 3. 啟動服務

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

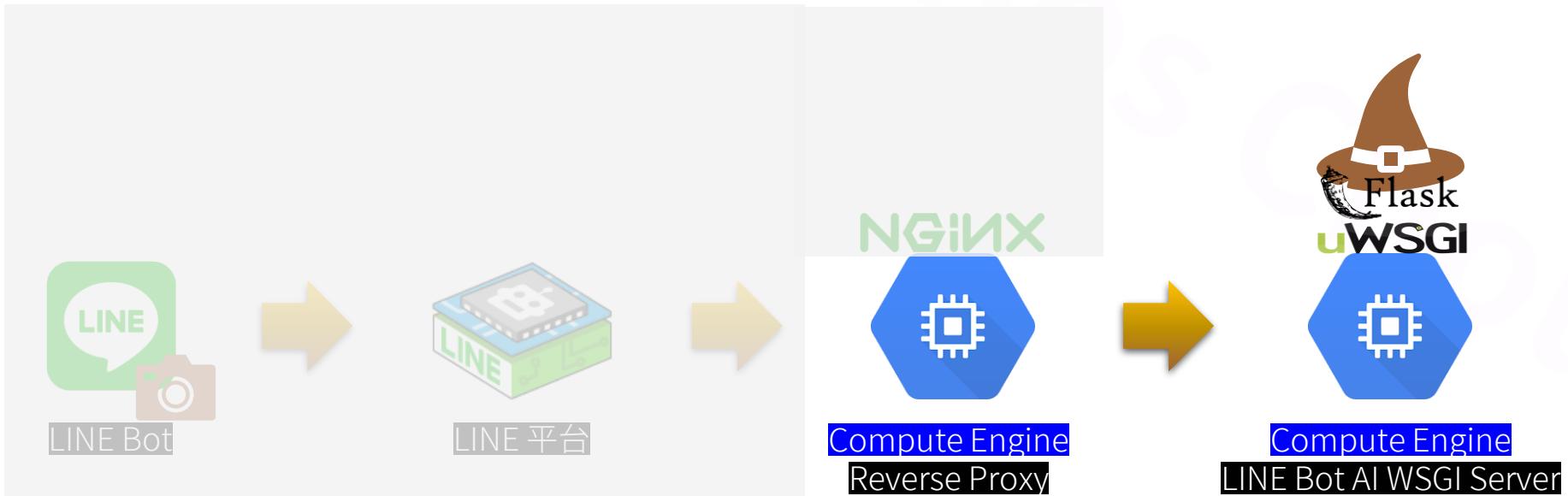
```
sudo systemctl stop your_service
```

```
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, a search bar containing the text 'enadv', and links for '聯絡我們', '說明', '登入', and a shopping cart icon. The main banner features the text '運用 .com 帶領企業進軍網路世界，只要 NT\$291/第 1 年'. Below the banner is a search bar with the placeholder '搜尋'.

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 website interface. At the top, there is a navigation bar with links for '聯絡我們' (Contact Us), '說明' (Help), '登入' (Log In), and a shopping cart icon. Below the navigation is a search bar containing the text 'enadv'. To the right of the search bar are 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	Action
enadv.com	NT\$3,653 <sup>②</sup>	加入購物車
enadv.tw	NT\$1,070 NT\$734 <sup>②</sup> 第一年	加入購物車
enadv.net	NT\$592 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\$676 NT\$319 <sup>②</sup> 第一年	加入購物車
enadv.cc	NT\$460 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 period "1年". To the right, the price is listed as "NT\$33" with a note "現省 94%" and a "NT\$537" crossed-out price. Below this, a yellow box highlights that the domain is "每年 170,000 次" (attempted to be stolen) and offers "全方位網域隱私及保護" at "NT\$199/年, 每個網域". A "新增" button is shown next to this offer. At the bottom of the main section, there are links for "查看優惠免責聲明" and "清空購物車".

**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**

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'.

In the main content area, a product summary for 'enadv.club .CLUB 網域註冊' is displayed. It shows a discount from NT\$537 to NT\$35, indicating a savings of 94%. The renewal period is set to 1 year. A note states: '於 2022年6月續約，費用為 NT\$537 (1 年效期)'.

The total summary at the bottom shows a subtotal of NT\$28, tax of NT\$7, and a promotional code of GOKBTW06. The final total is NT\$35. A message at the bottom left says: '讚！您的訂單成功省下 NT\$474。' A large red button labeled '完成購買' is prominently displayed at the bottom right.

At the very bottom of the page, a small note reads: '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' with the heading '既快又安全的主機服務。' and a subtext '包含免費網域 + email。最低 NT\$62 起。' A button labeled '瞭解更多資訊' is present. To the right of this section is a photograph of a man named Christopher Lee holding a book titled 'Wheat and Wood'. The sidebar on the right contains account information such as '客戶編號: PIN: 查看', a balance of 'NT\$319', and a shopping cart icon with 'T\$224'. A dropdown menu is open, showing options like '我的產品' (highlighted with a pink box), '續約與帳單', '帳戶設定', 'Office 365 Email 登入', and '登入 GoDaddy Webmail'. At the bottom right, there is a 'info' logo and a promotional message: '特賣中！NT\$130/第 1 年。讓全世界知道您的知識素養。' A blue button at the bottom right says '洽詢我們'.

.app 網域 NT\$576/第 1 年	Microsoft 365 NT\$52/月	虛擬主機 NT\$62/月	
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'. Underneath, there's a card for 'enadv.site' featuring a globe icon, the domain name, and the text '建立網站或電子郵件地址'. A '網域' (Domain) section follows, displaying a list of domain names. For 'enadv.site', a context menu is open with options: '建立網站', '設定電子郵件帳戶', '連線至現有網站', '管理我的網域', '編輯設定' (highlighted with a pink box), and '變更隱私'. At the bottom right of the page is a blue button labeled '洽詢我們'.

# 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		

Buttons at the top of the record table include: 刪除 (Delete), 複製 (Copy), 篩選 (Filter), 新增 (Add - highlighted with a pink box), and more options.

# 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.

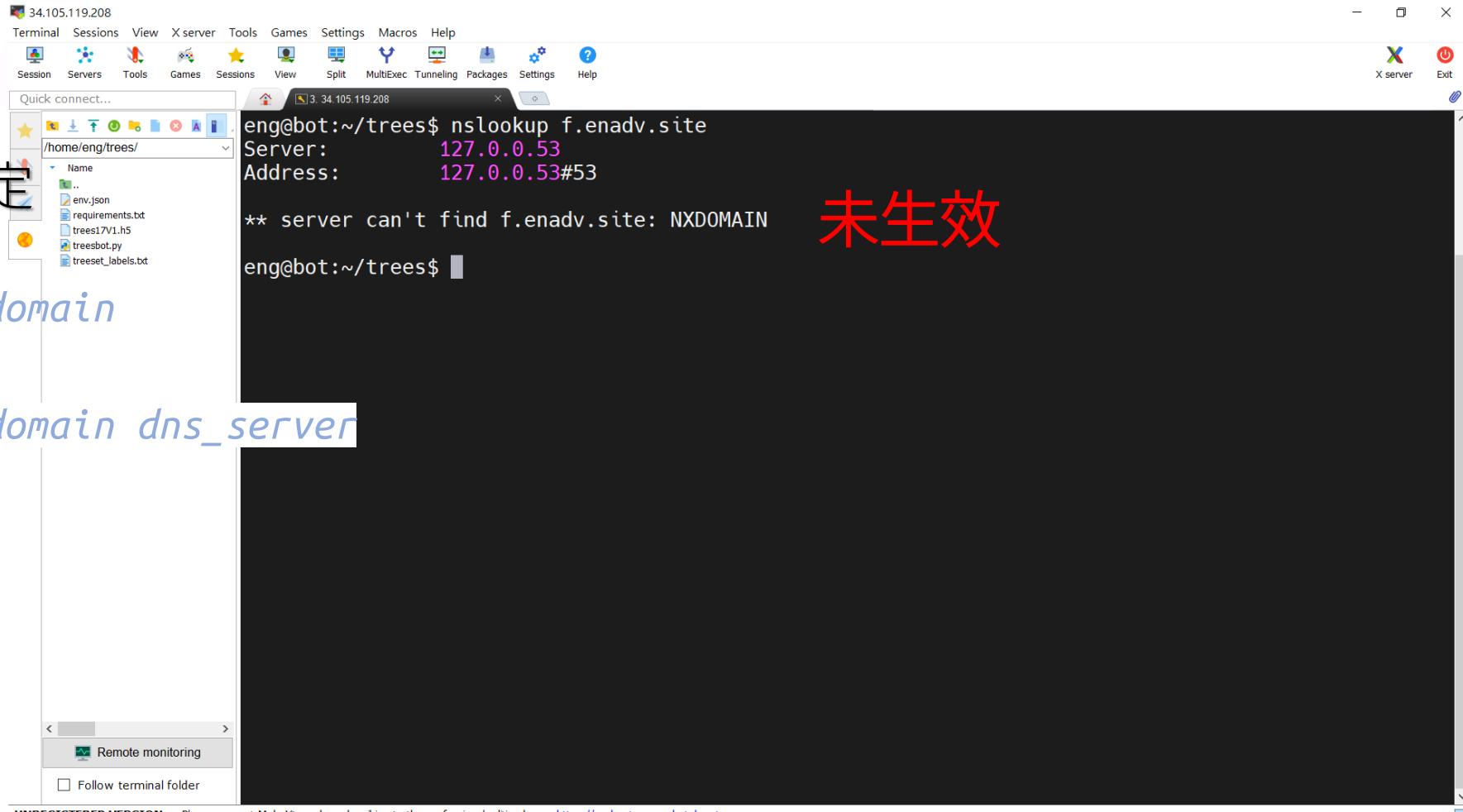
# 申購並設定網域

## 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, 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

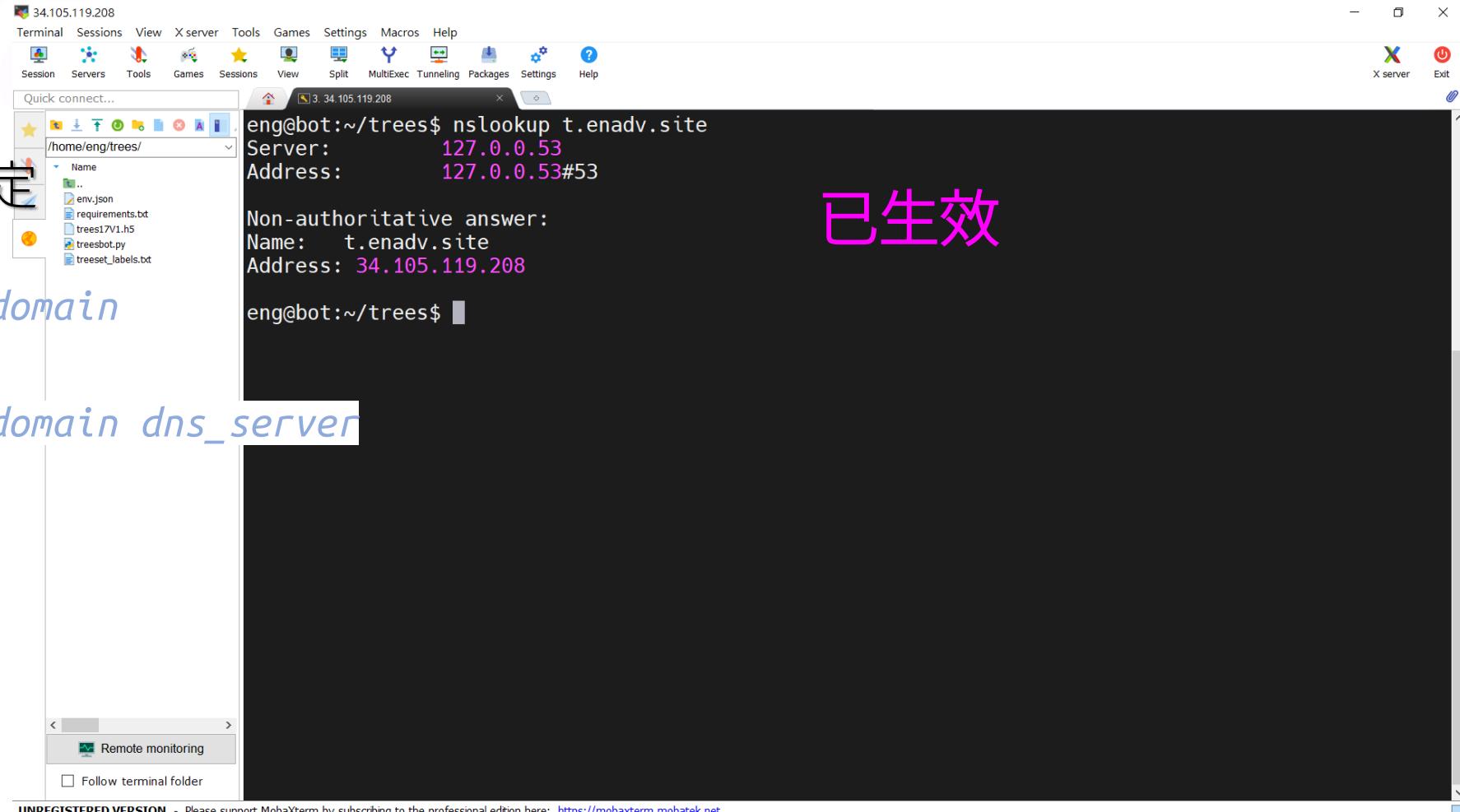
## 驗證網域

確認網域與 IP 綁定

`nslookup your_domain`

`nslookup your_domain dns_server`

已生效



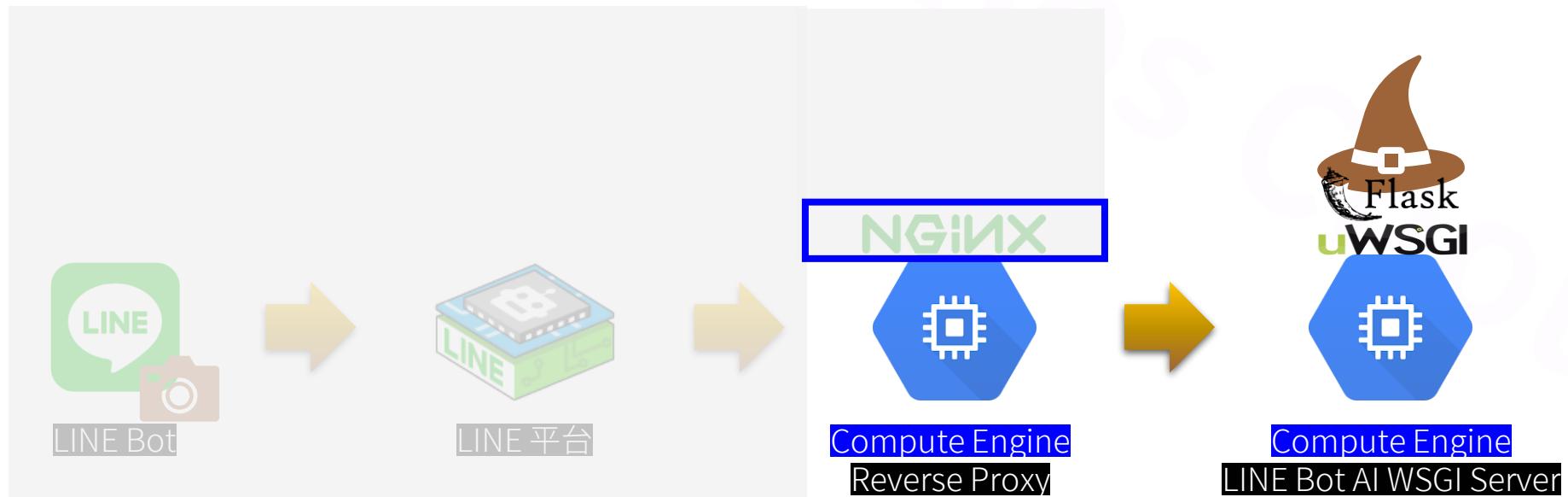
The screenshot shows a terminal window in MobaXterm with the title bar "34.105.119.208". The menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The terminal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



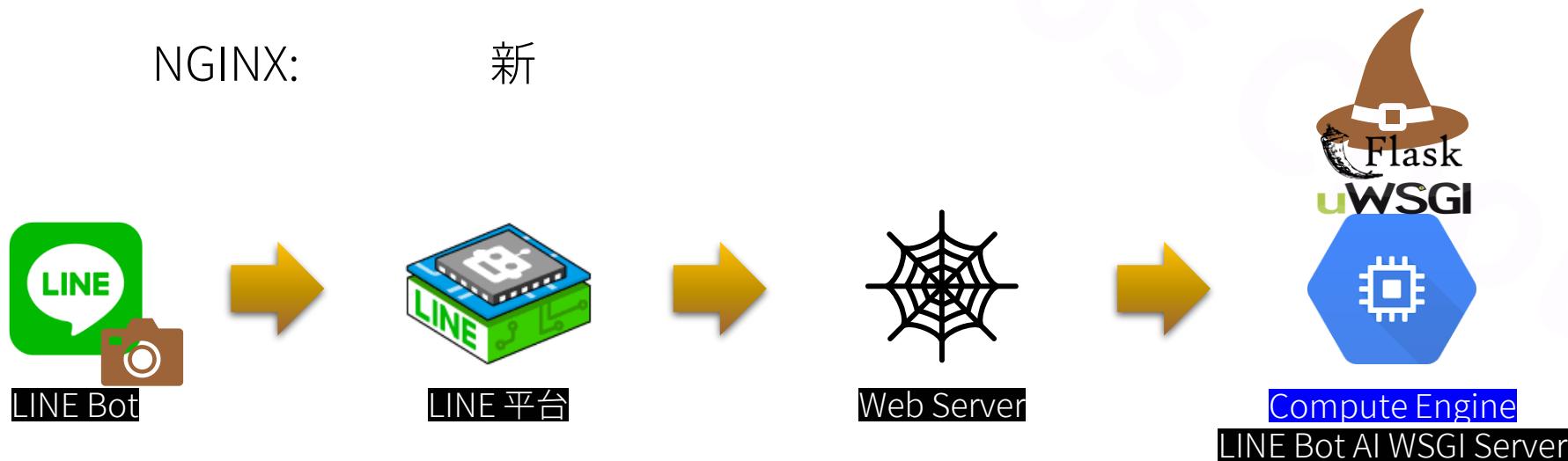
# 網站與憑證

## 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

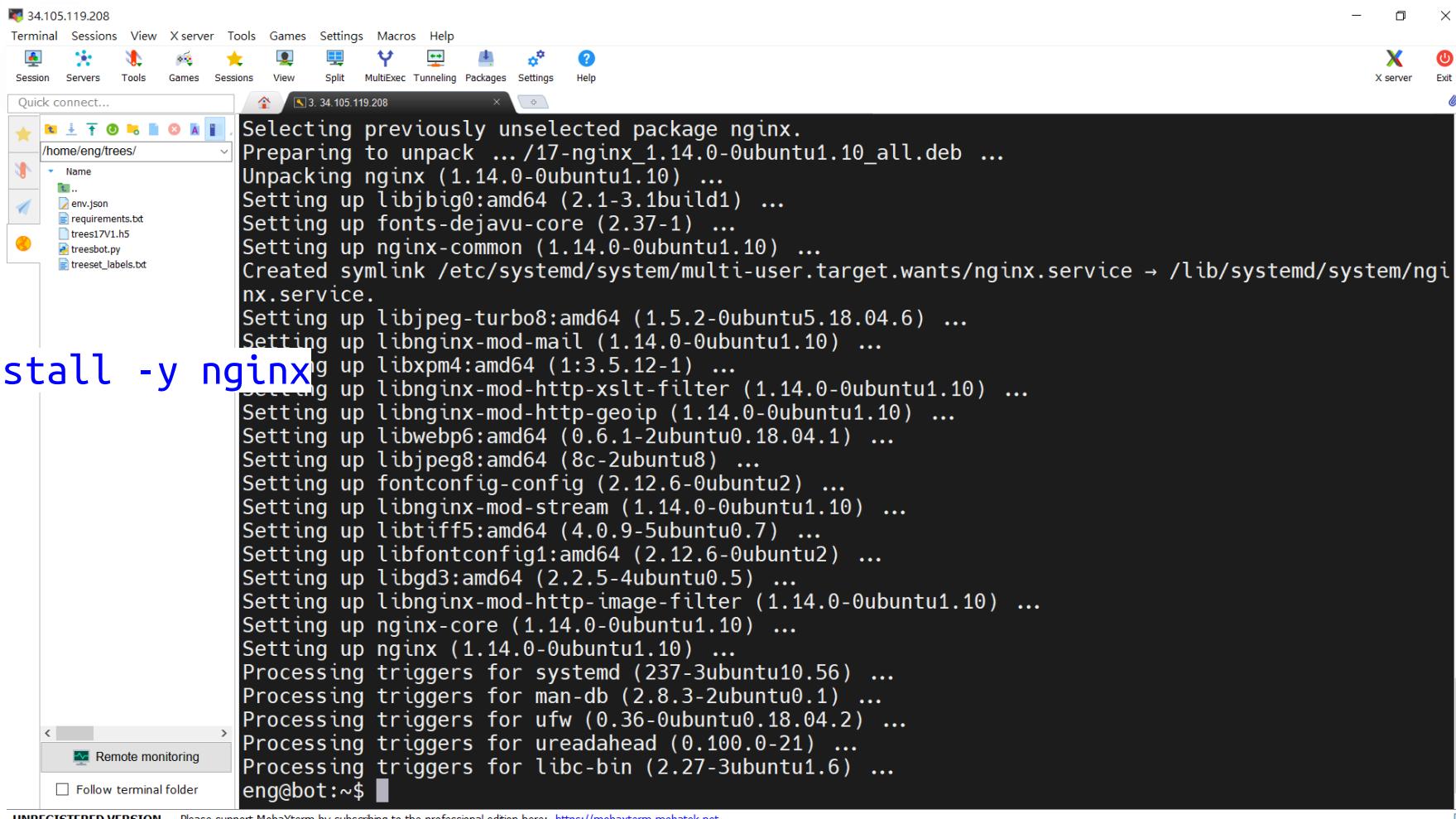
# 網站與憑證

## 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 with the IP address 34.105.119.208. The terminal output displays the process of installing the nginx package via apt-get. The output includes messages like "Preparing to unpack ... /17-nginx\_1.14.0-0ubuntu1.10\_all.deb ...", "Unpacking nginx (1.14.0-0ubuntu1.10) ...", and "Setting up nginx (1.14.0-0ubuntu1.10) ...". The terminal window also shows a file browser sidebar with files like env.json, requirements.txt, trees17V1.h5, treesbot.py, and treeset\_labels.txt.

```
Selecting previously unselected package nginx.
Preparing to unpack ... /17-nginx_1.14.0-0ubuntu1.10_all.deb ...
Unpacking nginx (1.14.0-0ubuntu1.10) ...
Setting up libjbig2:amd64 (2.1-3.1build1) ...
Setting up fonts-dejavu-core (2.37-1) ...
Setting up nginx-common (1.14.0-0ubuntu1.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Setting up libjpeg-turbo8:amd64 (1.5.2-0ubuntu5.18.04.6) ...
Setting up libnginx-mod-mail (1.14.0-0ubuntu1.10) ...
Setting up libxpm4:amd64 (1:3.5.12-1) ...
Setting up libnginx-mod-http-xslt-filter (1.14.0-0ubuntu1.10) ...
Setting up libnginx-mod-http-geoip (1.14.0-0ubuntu1.10) ...
Setting up libwebp6:amd64 (0.6.1-2ubuntu0.18.04.1) ...
Setting up libjpeg8:amd64 (8c-2ubuntu8) ...
Setting up fontconfig-config (2.12.6-0ubuntu2) ...
Setting up libnginx-mod-stream (1.14.0-0ubuntu1.10) ...
Setting up libtiff5:amd64 (4.0.9-5ubuntu0.7) ...
Setting up libfontconfig1:amd64 (2.12.6-0ubuntu2) ...
Setting up libgd3:amd64 (2.2.5-4ubuntu0.5) ...
Setting up libnginx-mod-http-image-filter (1.14.0-0ubuntu1.10) ...
Setting up nginx-core (1.14.0-0ubuntu1.10) ...
Setting up nginx (1.14.0-0ubuntu1.10) ...
Processing triggers for systemd (237-3ubuntu10.56) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.2) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
eng@bot:~$
```

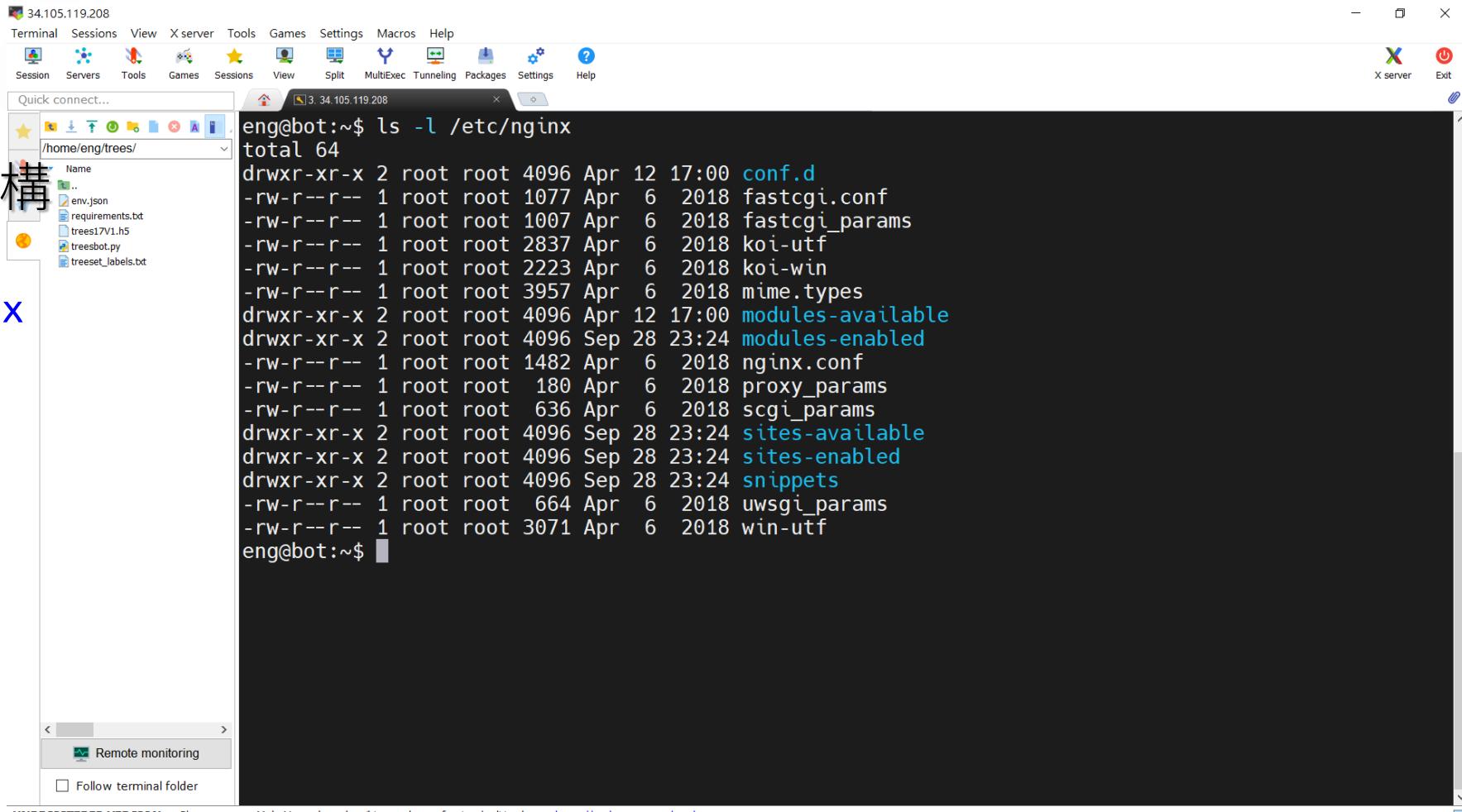
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 網站與憑證

## 1. 架設 NGINX

b. 觀察組態檔架構

`ls -l /etc/nginx`



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

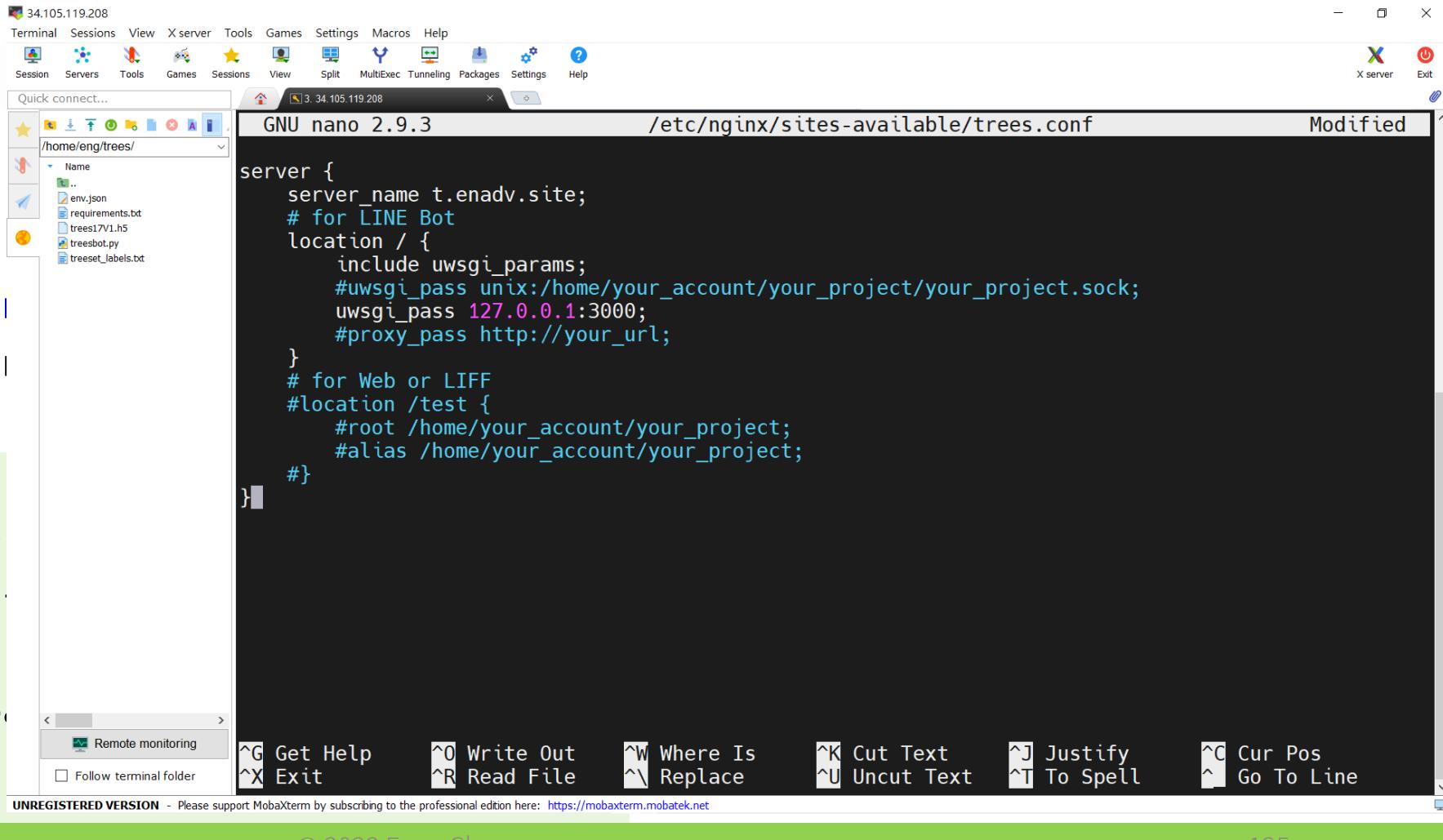
## 1. 架設 NGINX

### c. 建立組態檔

```
sudo nano /etc/
```

```
sudo nano /etc/
```

```
server {  
    server_name t.enadv.site;  
    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 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 application, which includes a file browser sidebar on the left. The configuration file contains code for setting up a server block for 't.enadv.site' using uWSGI and proxying to a local socket or port. The terminal interface includes standard nano keybindings at the bottom.

```
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;  
    }  
}
```

# 網站與憑證

## 1. 架設 NGINX

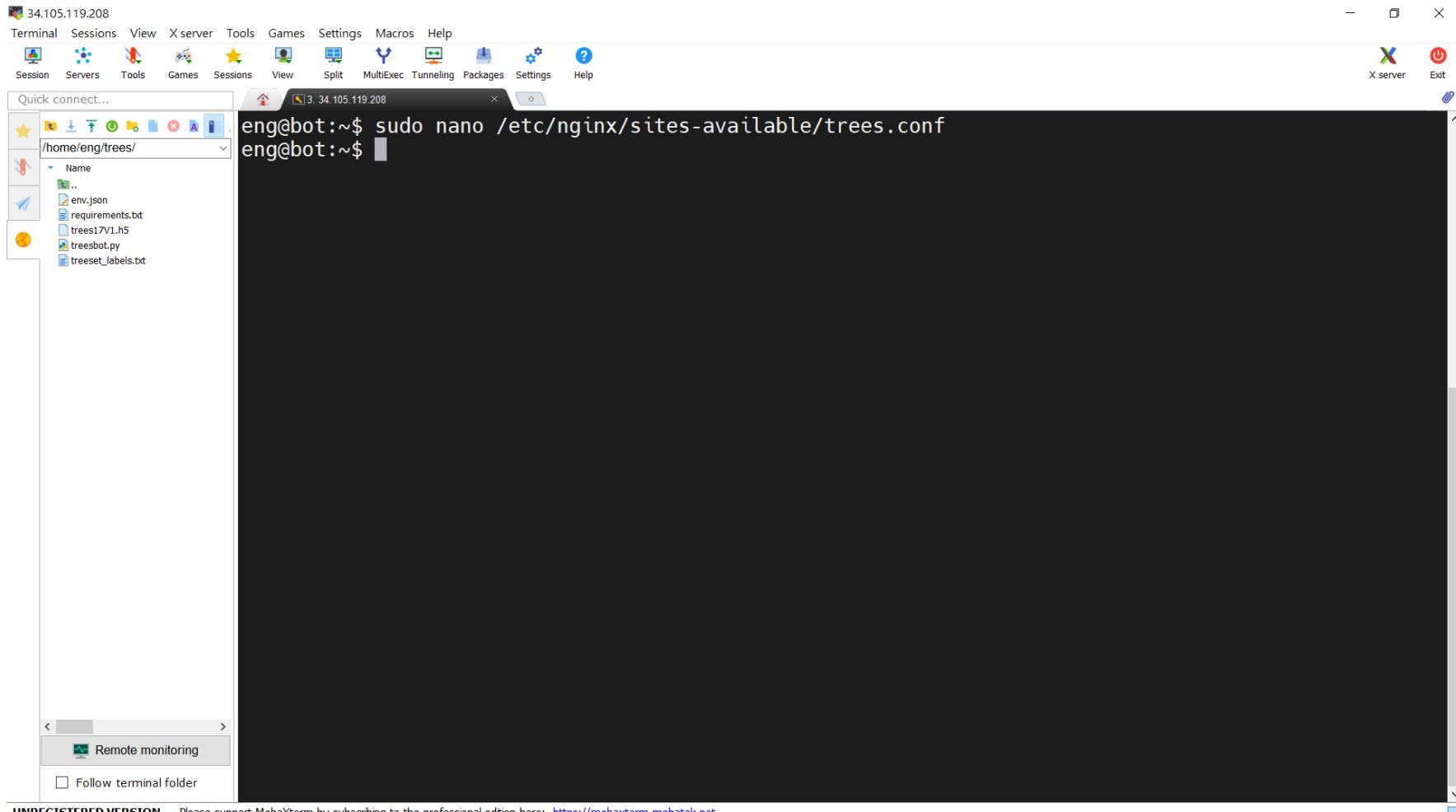
### c. 建立組態檔

# 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

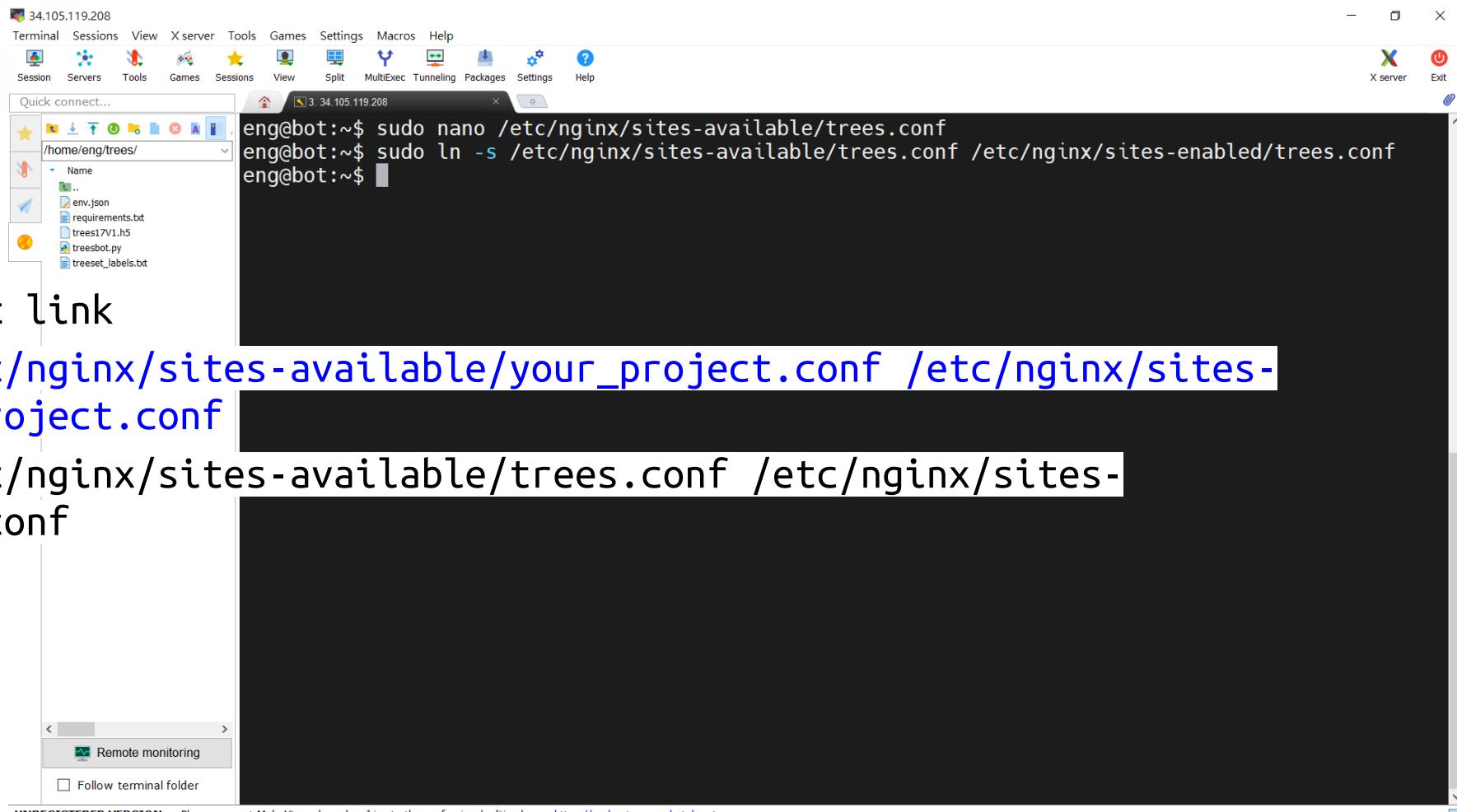
## 1. 架設 NGINX

### 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
```

The screenshot shows a terminal window titled '34.105.119.208' running on MobaXterm. The terminal content is as follows:

```
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 terminal window has a sidebar showing a file tree under '/home/eng/trees/'. The bottom of the terminal window includes a status bar with 'UNREGISTERED VERSION' and a link to support the professional edition.

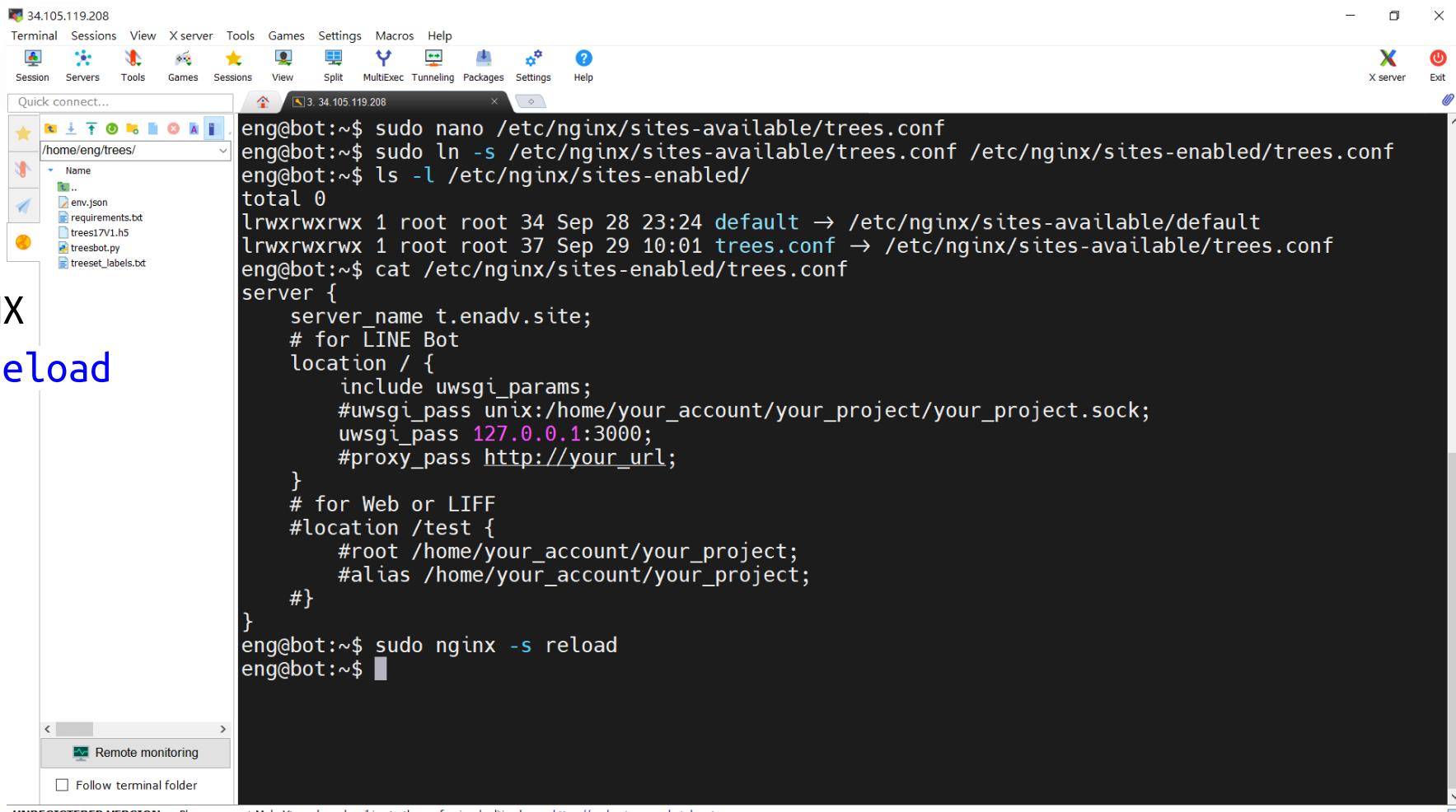
# 網站與憑證

## 1. 架設 NGINX

d. 載入新組態

# 重新啟動 NGINX

`sudo nginx -s reload`



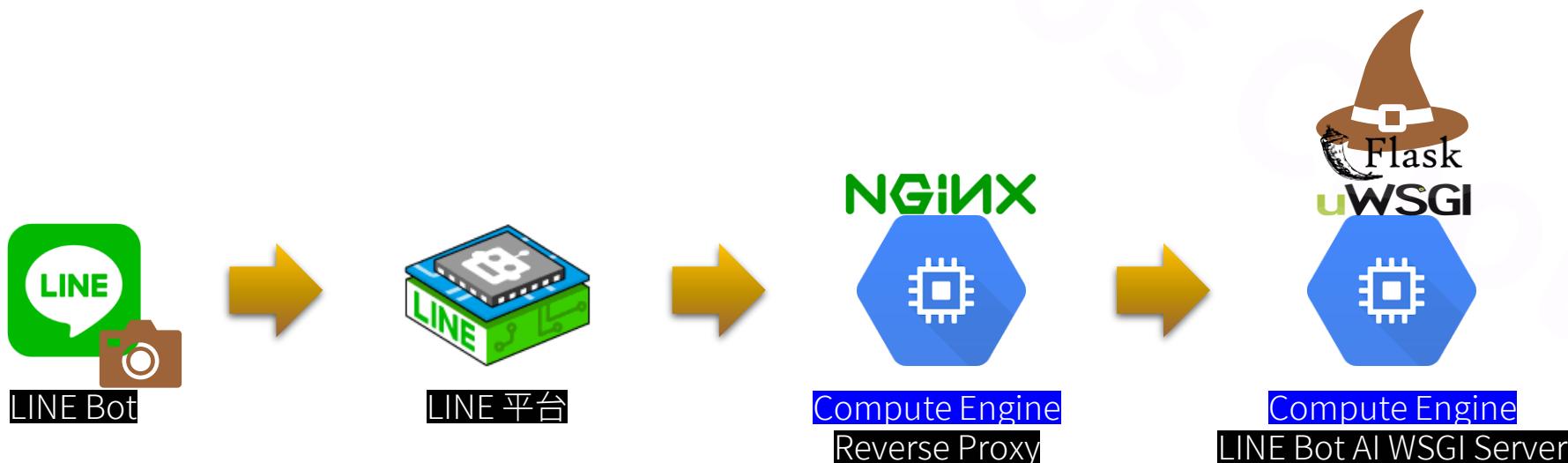
```
eng@bot:~$ sudo nano /etc/nginx/sites-available/trees.conf
eng@bot:~$ sudo ln -s /etc/nginx/sites-available/trees.conf /etc/nginx/sites-enabled/trees.conf
eng@bot:~$ ls -l /etc/nginx/sites-enabled/
total 0
lrwxrwxrwx 1 root root 34 Sep 28 23:24 default → /etc/nginx/sites-available/default
lrwxrwxrwx 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

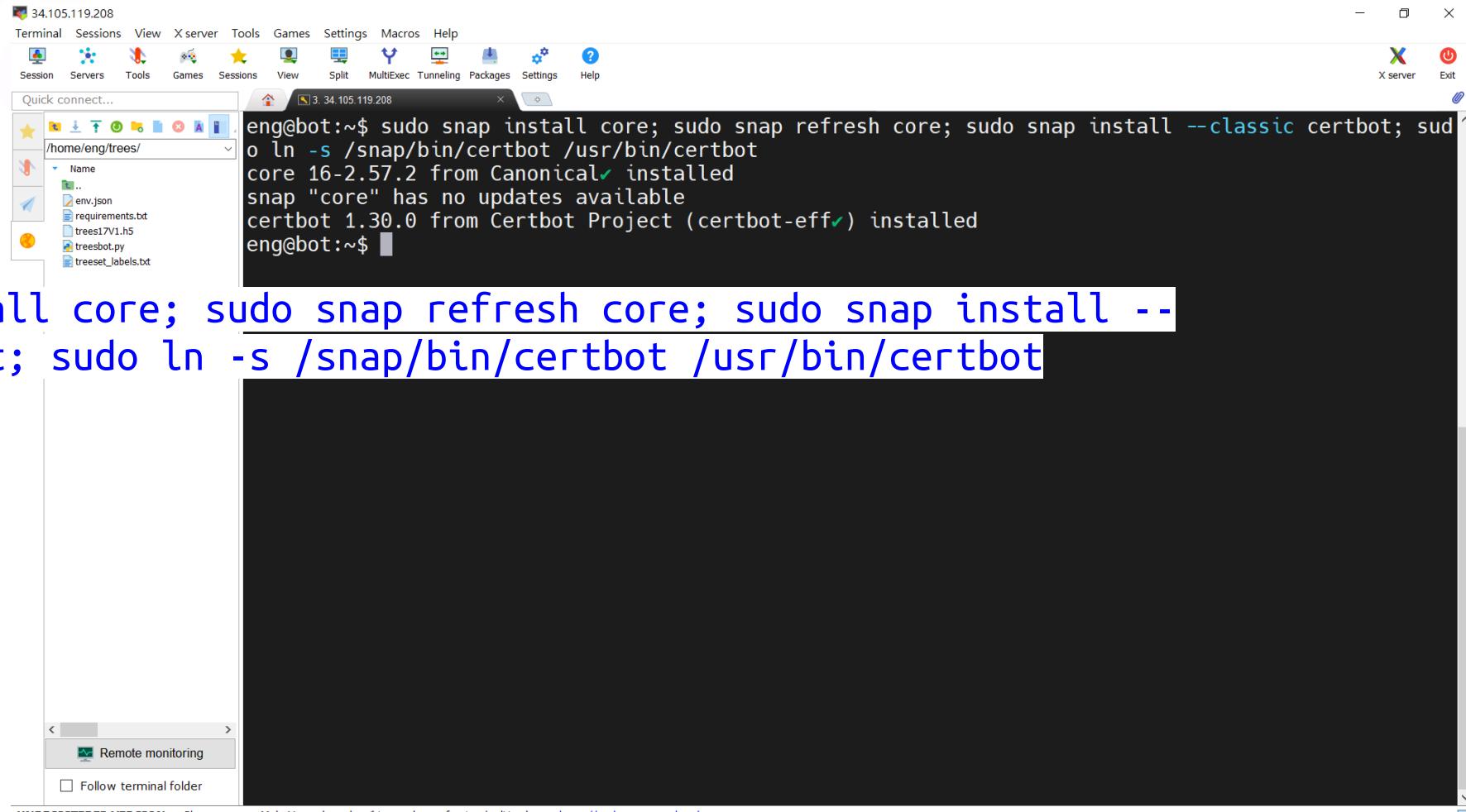


# 網站與憑證

## 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:~\$' running on a Linux system with IP 34.105.119.208. 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): 輸入 email
```

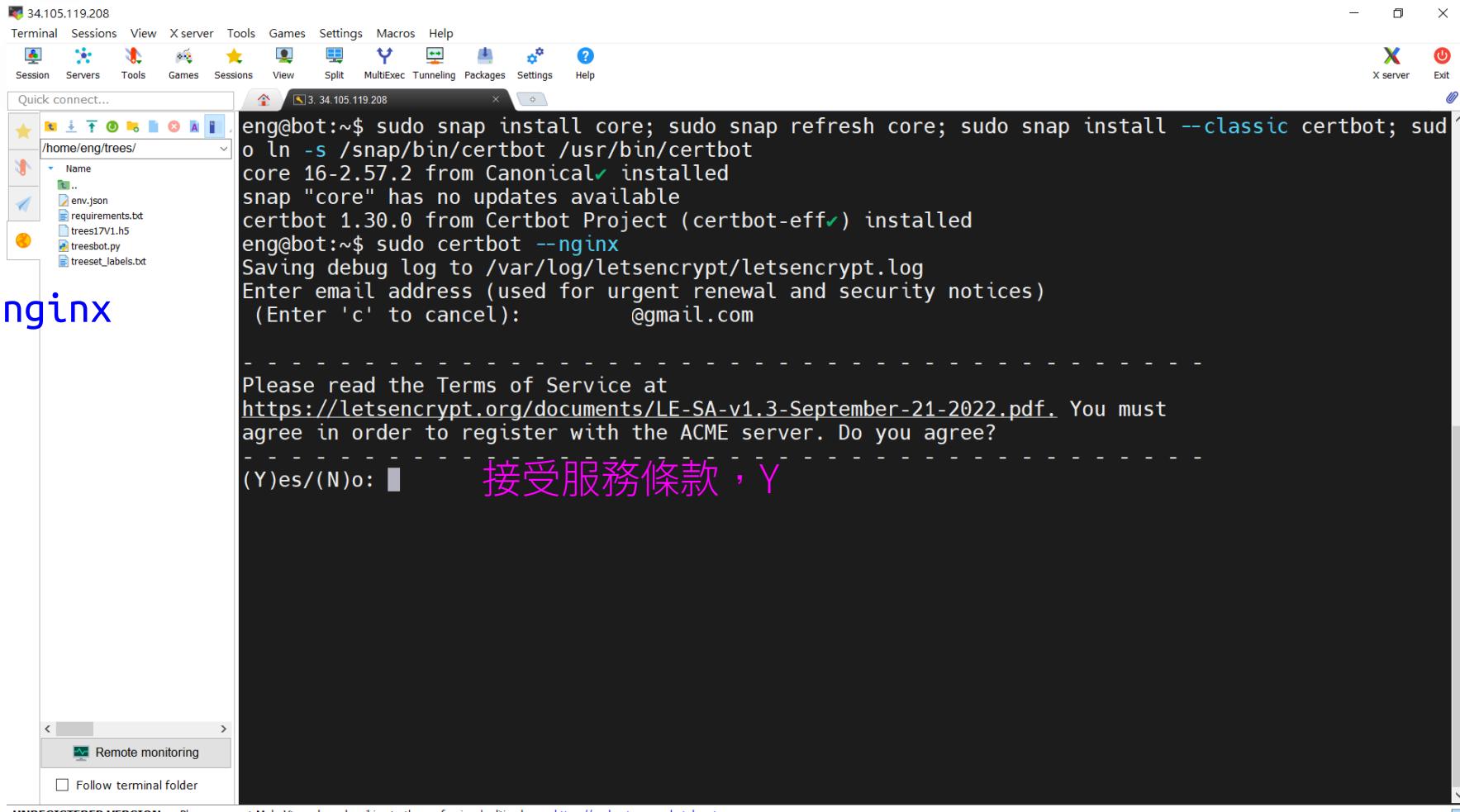
The terminal interface includes a file browser sidebar on the left showing a directory structure under '/home/eng/trees/'. The bottom of the window has status bars for 'Remote monitoring' and 'Follow terminal folder', and a footer note about the unregistered version.

# 網站與憑證

## 2. 架設 Certbot

### b. 申請憑證

`sudo certbot --nginx`



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

```
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
```

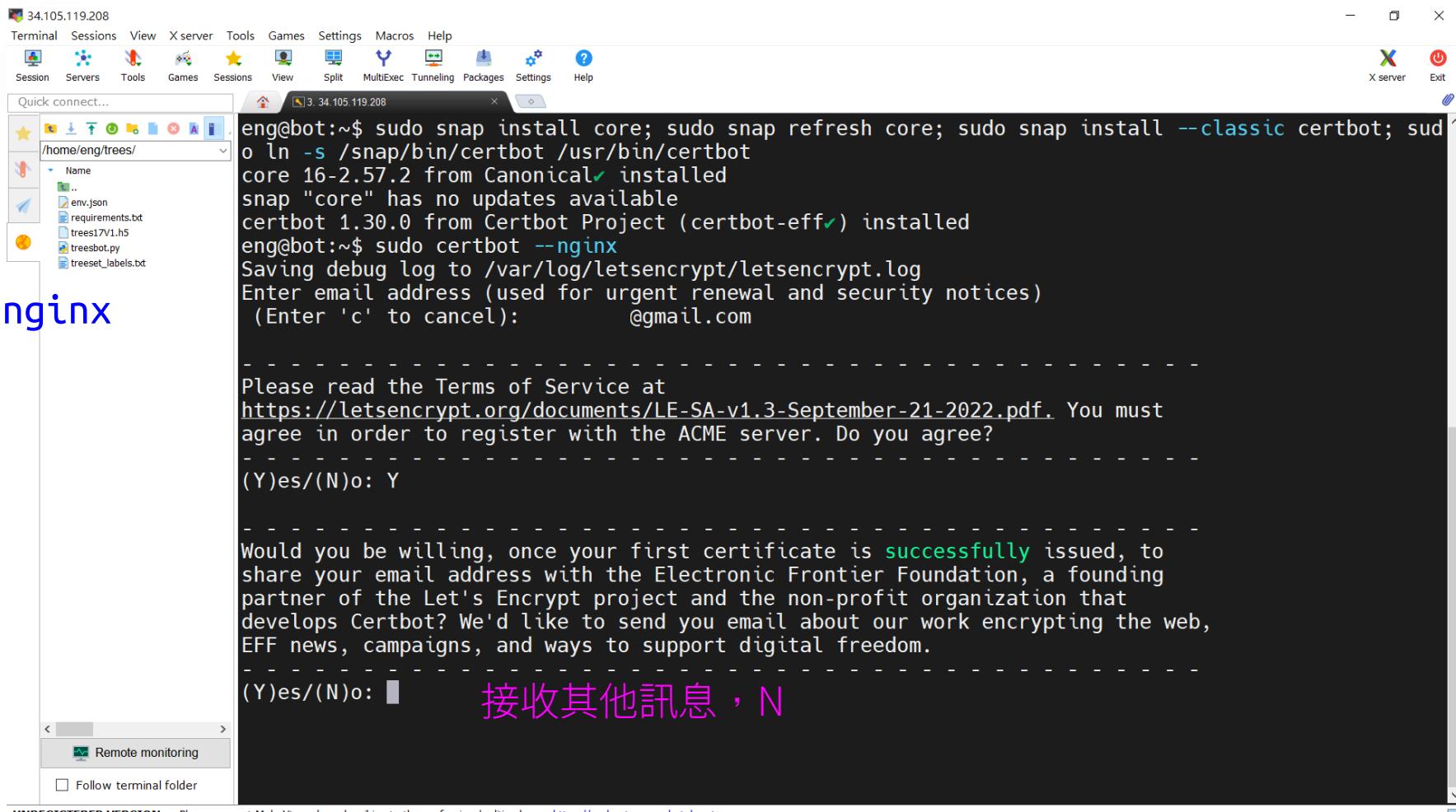
The terminal window title is "34.105.119.208". The menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help, and Exit. The status bar at the bottom says "UNREGISTERED 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):
```

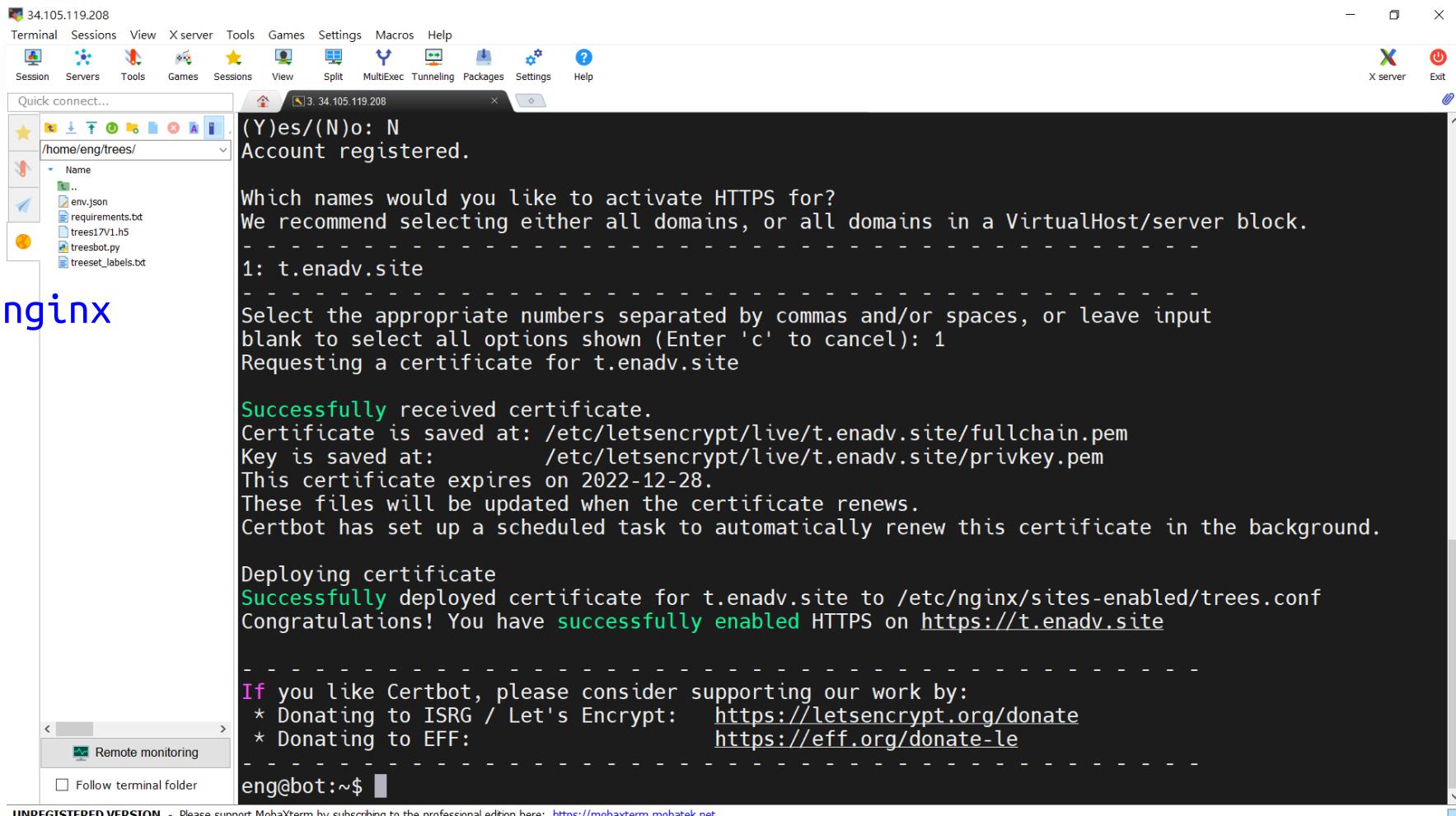
A pink annotation highlights the number '1' next to the domain 't.enadv.site'.

# 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

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/t.enadv.site/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/t.enadv.site/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

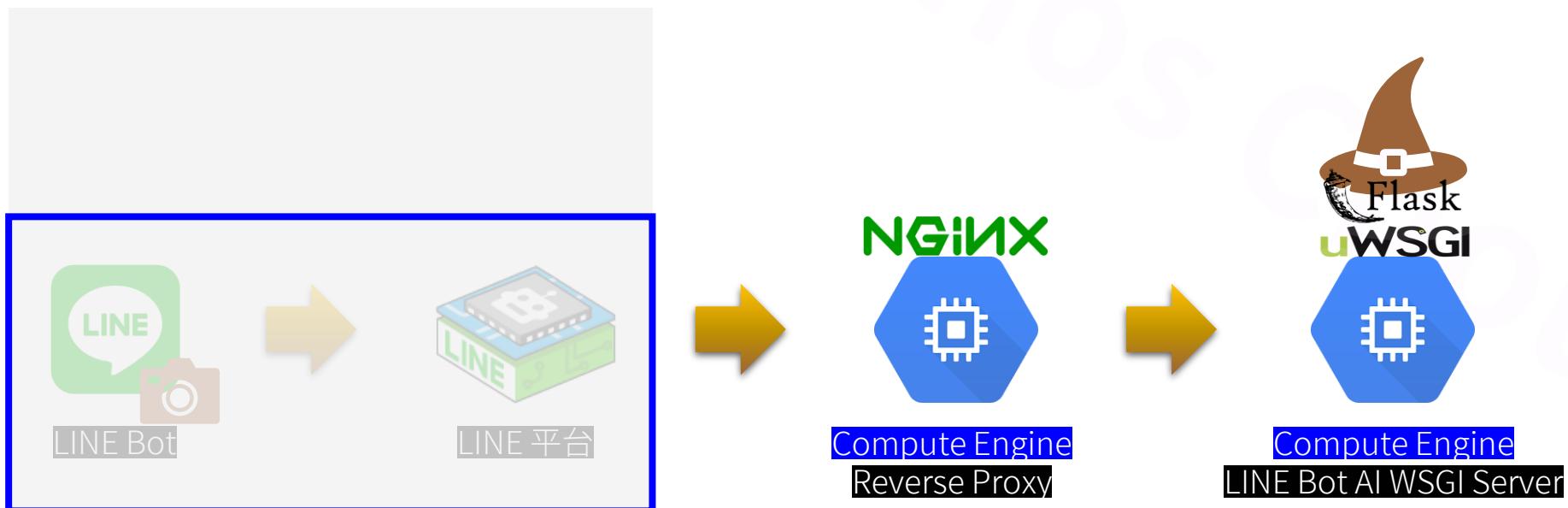
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# 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 in pink), and an 'Edit' button. Below that, 'Use webhook' is turned on (green switch). Further down are 'Webhook redelivery' (off) and 'Error statistics aggregation' (on, green switch). At the bottom, there are links for '© LINE Corporation', 'Terms and policies', 'About trademarks', and 'Found any problems? Please use our inquiry form'.

# 設定 LINE Messaging

## 1. 調整 LINE Messaging

### a. 調整 Webhook URL

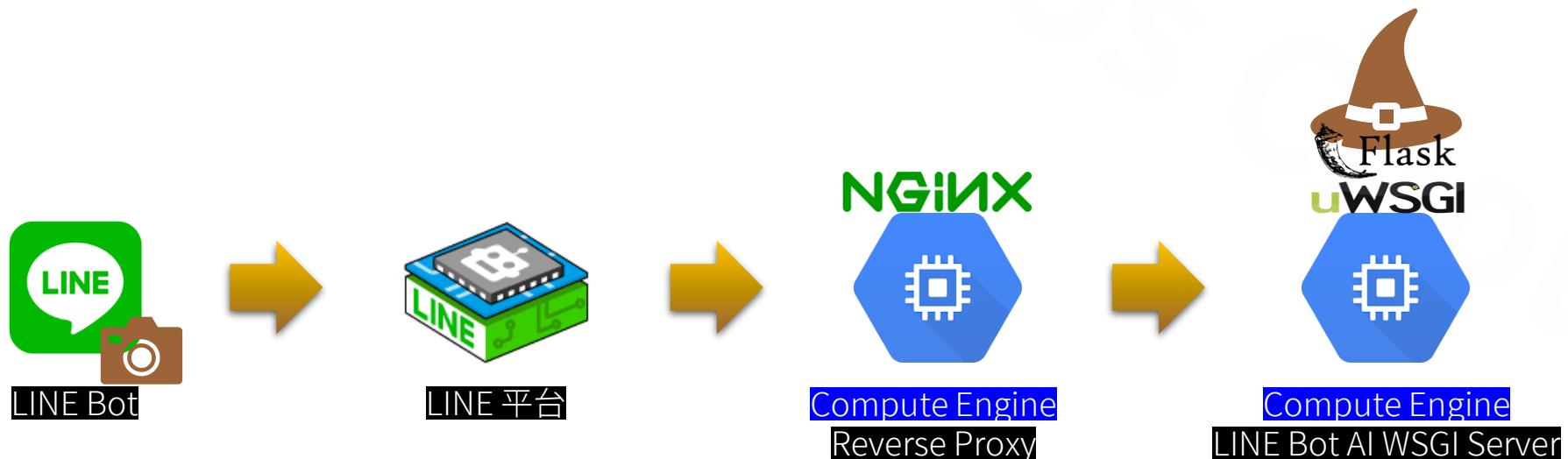
The screenshot shows the LINE Messaging API settings page. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin', 'Tools', and 'Support'. The main area shows 'Available APIs' with 'REPLY\_MESSAGE' and 'PUSH\_MESSAGE'. Under 'Webhook settings', the 'Webhook URL' is set to 'https://tenadv.site/callback', and a 'Verify' button is visible. A modal dialog box is open, displaying 'Success' above a large green 'OK' button. Below the modal, there are toggle switches for 'Use webhook' (on), 'Webhook redelivery' (off), and 'Error statistics aggregation' (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





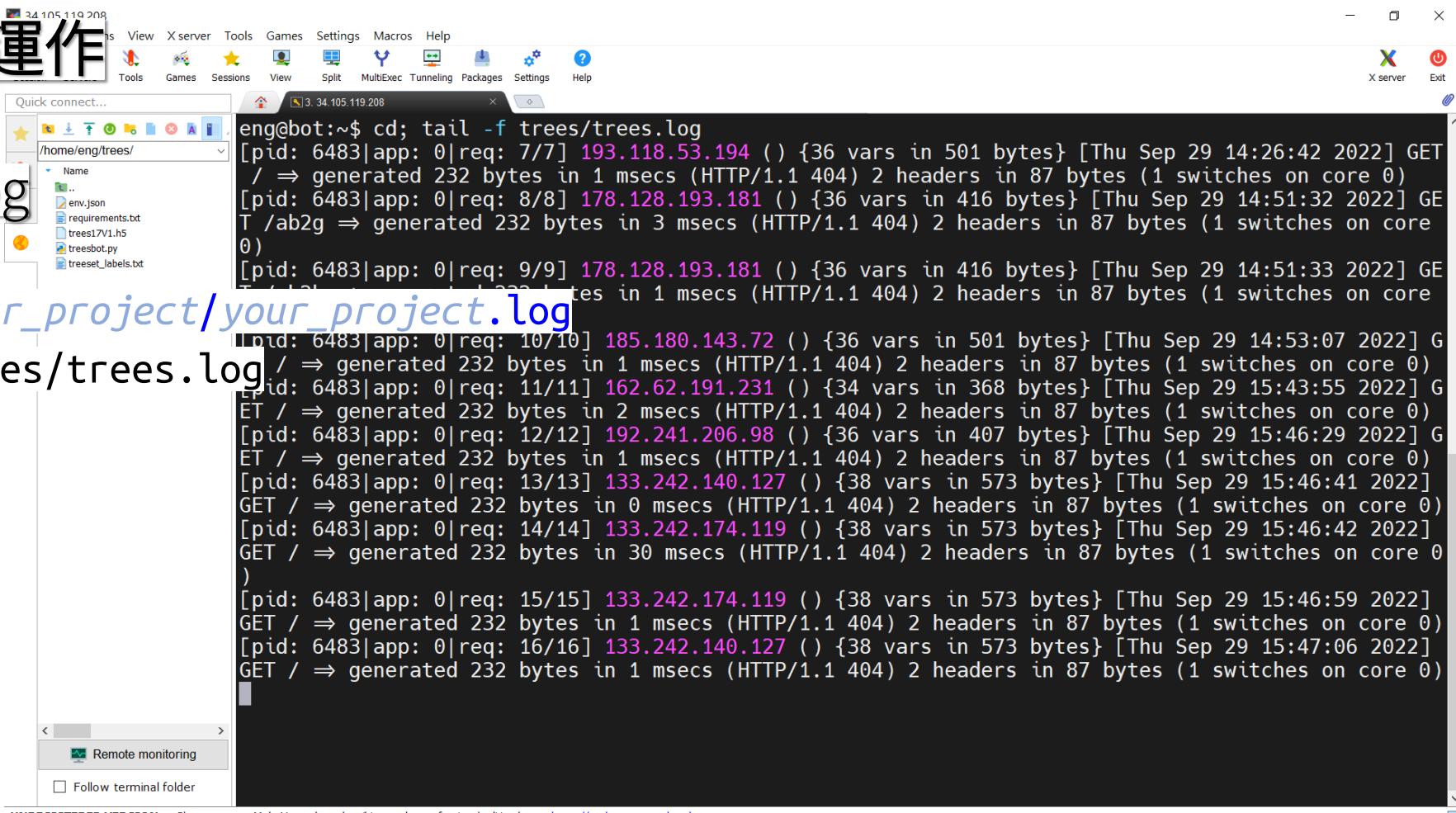
# Debug

## 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`



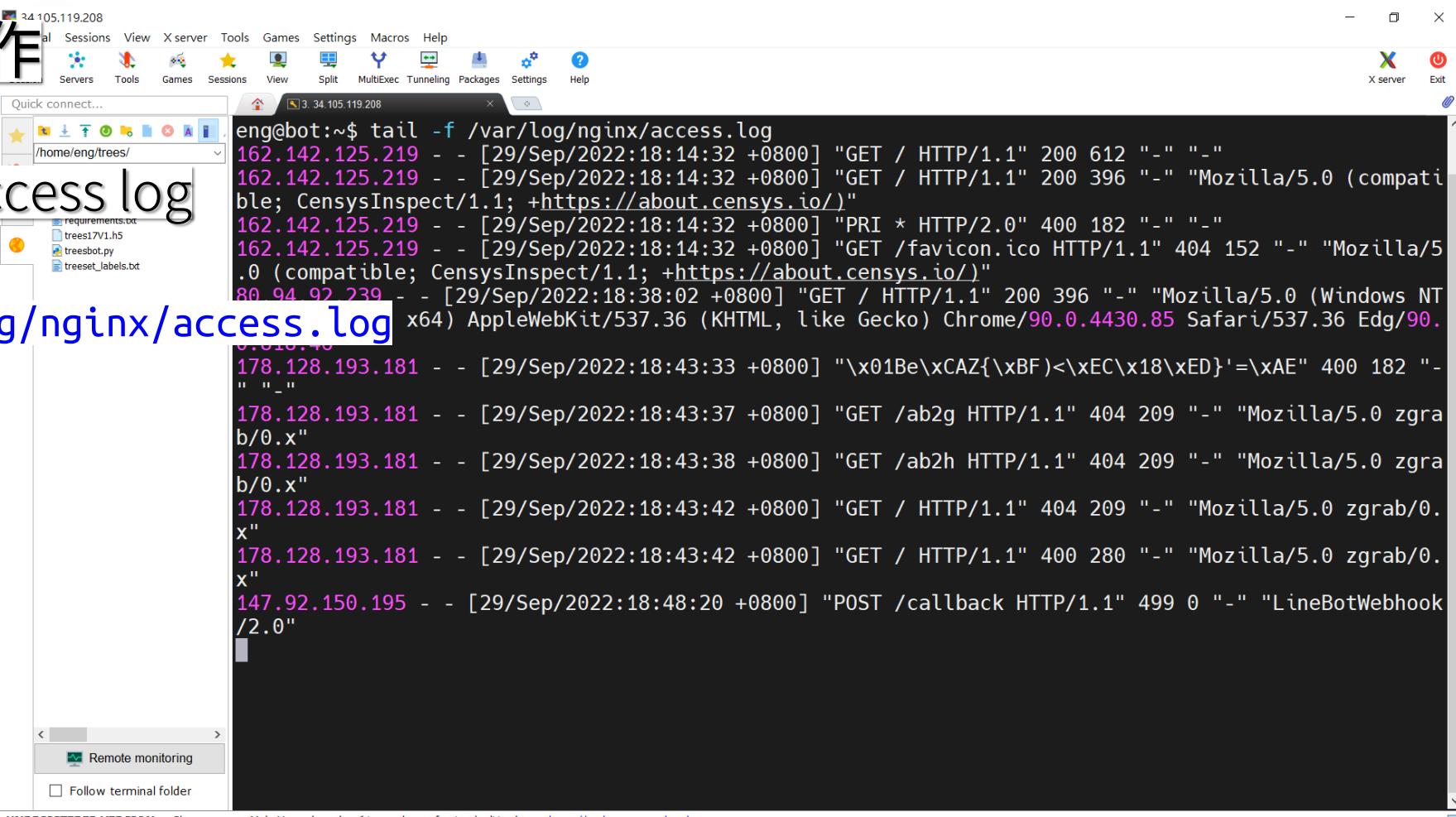
The screenshot shows a MobaXterm window with two panes. The left pane displays a file browser with a directory tree for '/home/eng/trees/'. The right pane is a terminal window titled '3. 34.105.119.208' showing log output from a uWSGI application. The logs show multiple requests from various IP addresses (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:~\$". 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 /, HEAD /, 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"
```

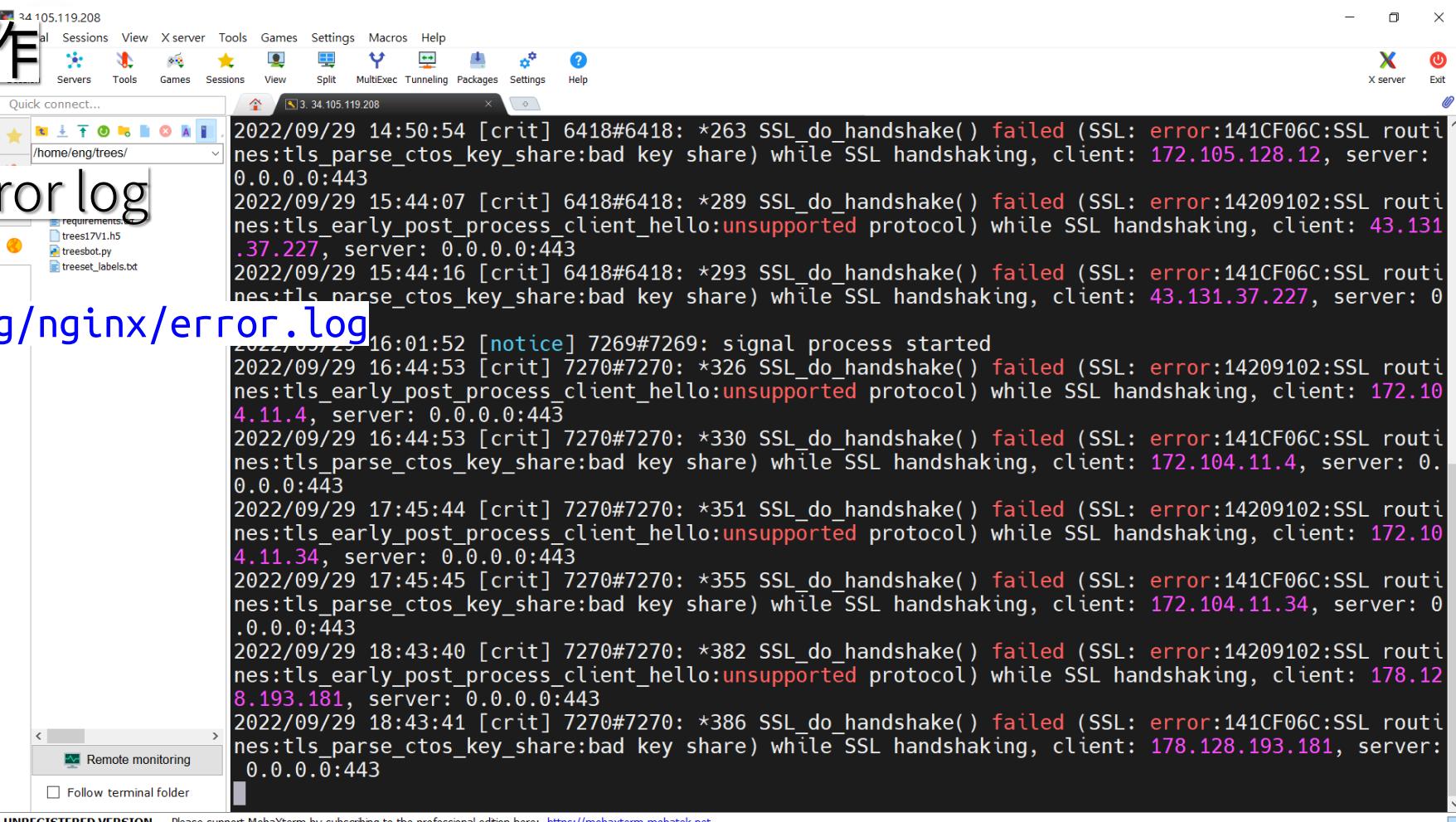
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

# Debug

## 2. 確認 NGINX 運作

### b. 檢視 NGINX error log

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

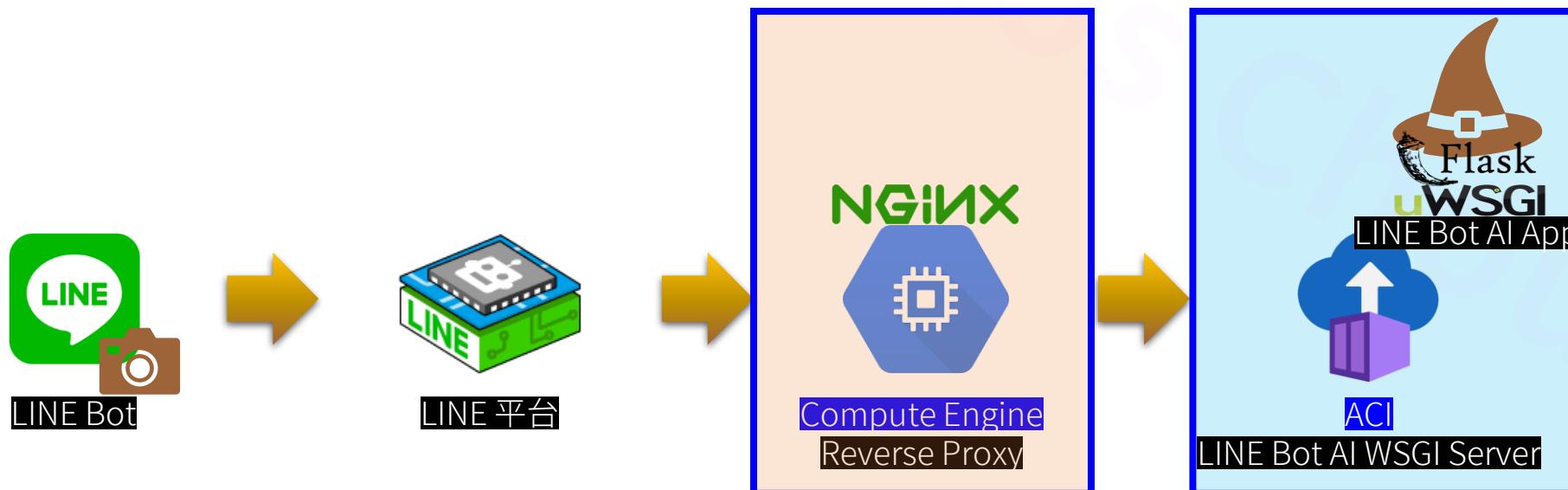


```
2022/09/29 14:50:54 [crit] 6418#6418: *263 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.105.128.12, server: 0.0.0.0:443
2022/09/29 15:44:07 [crit] 6418#6418: *289 SSL_do_handshake() failed (SSL: error:14209102:SSL 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

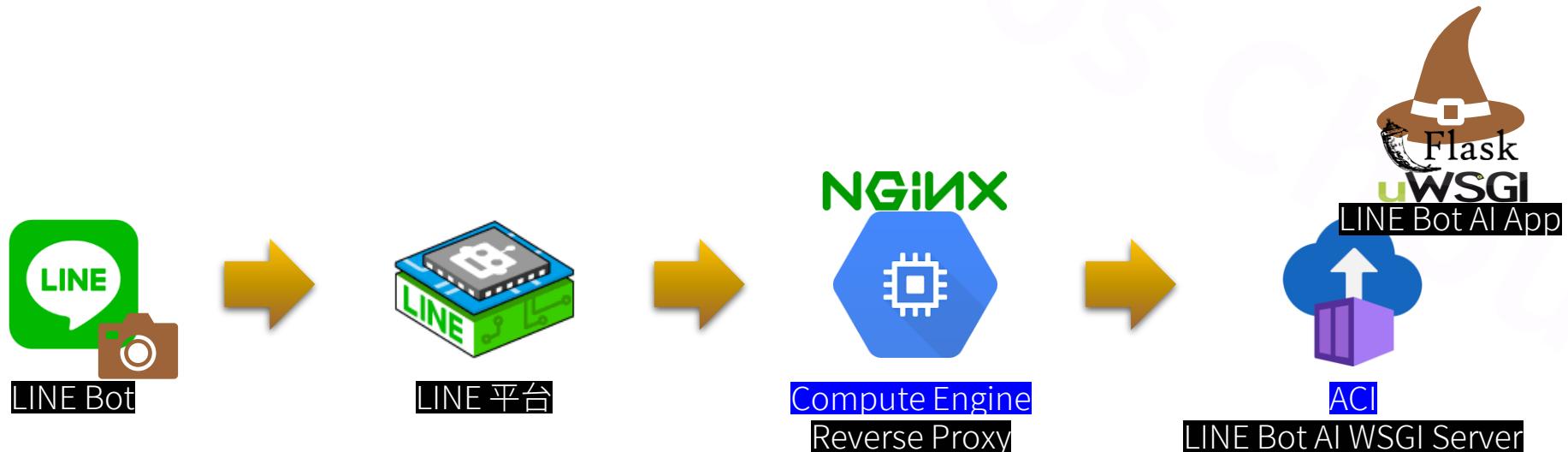
1. 長期可用且廉價的硬體環境 ... Google Compute Engine + Azure ACI
2. Flask as Web Server 的替代方案 ... NGINX + Domain + uWSGI
3. 長期可用且廉價的 SSL 網域方案 ... Let's Encrypt + Certbot

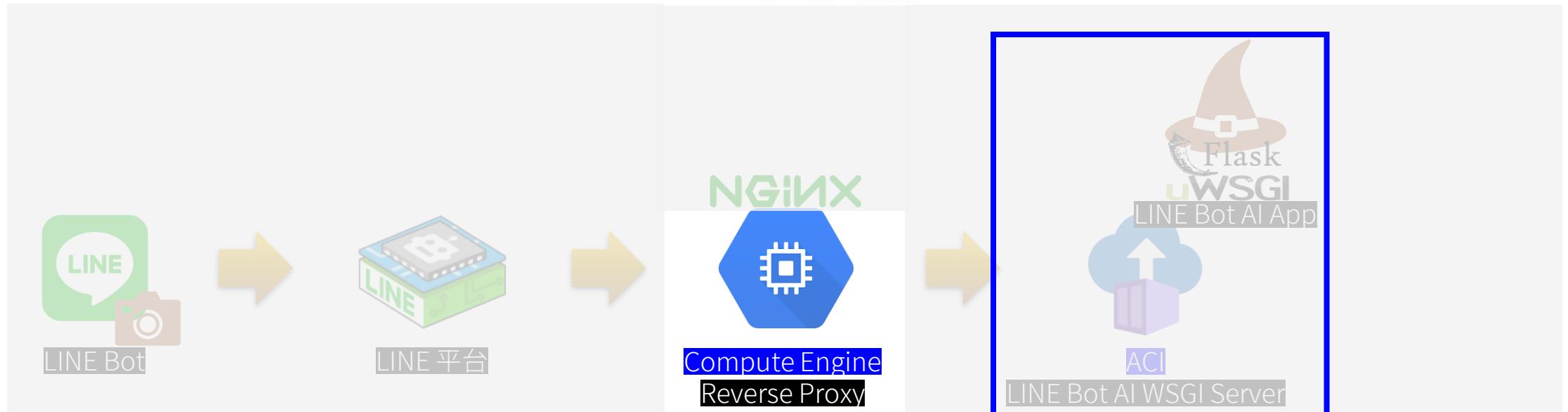


# 任務

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

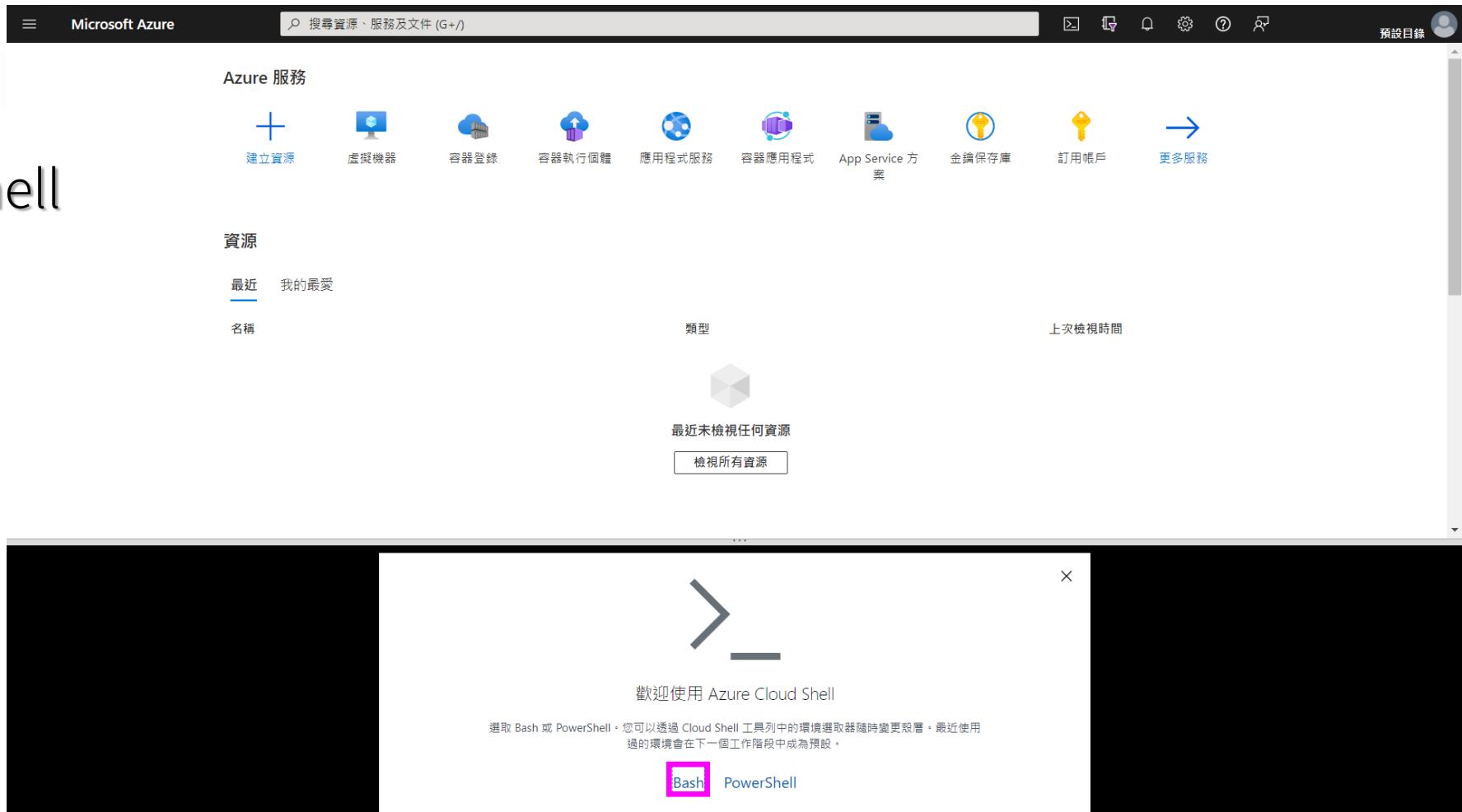
d. 啟動 Cloud Shell

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 this title, 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). A large '+' icon is also present. Below this section is a '資源' (Resources) panel. It has tabs for '最近' (Recent) and '我的最愛' (Favorites), which is currently selected. It includes columns for '名稱' (Name), '類型' (Type), and '上次檢視時間' (Last checked time). A message indicates '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 of the page, there are sections for '瀏覽' (Browse) and '工具' (Tools), each containing several icons for different Azure services like 訂用帳戶 (Subscription), 資源群組 (Resource Group), 所有資源 (All Resources), 儀表板 (Dashboard), 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

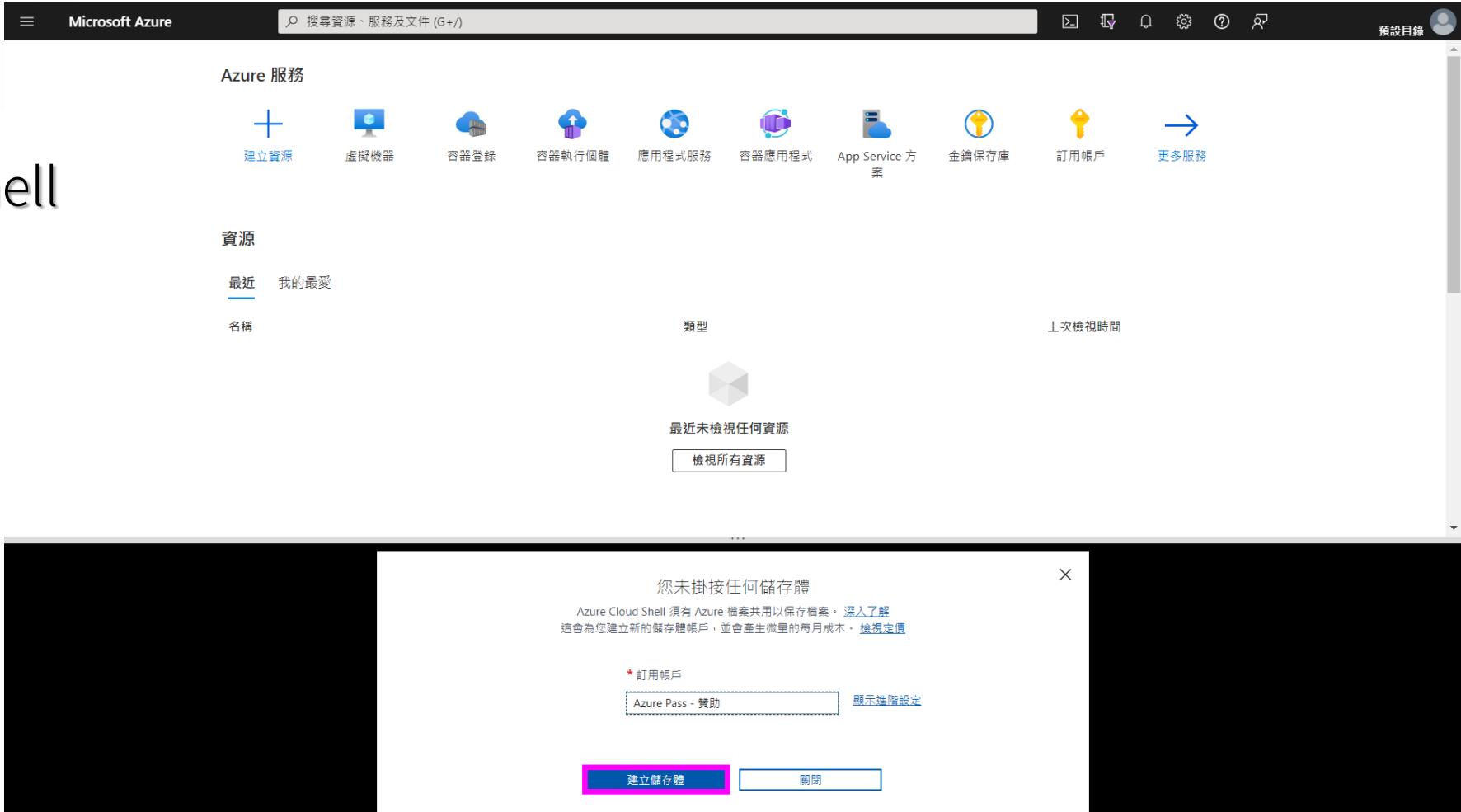
d. 啟動 Cloud Shell



# LINE Bot & WSGI

## 1. 準備程式碼

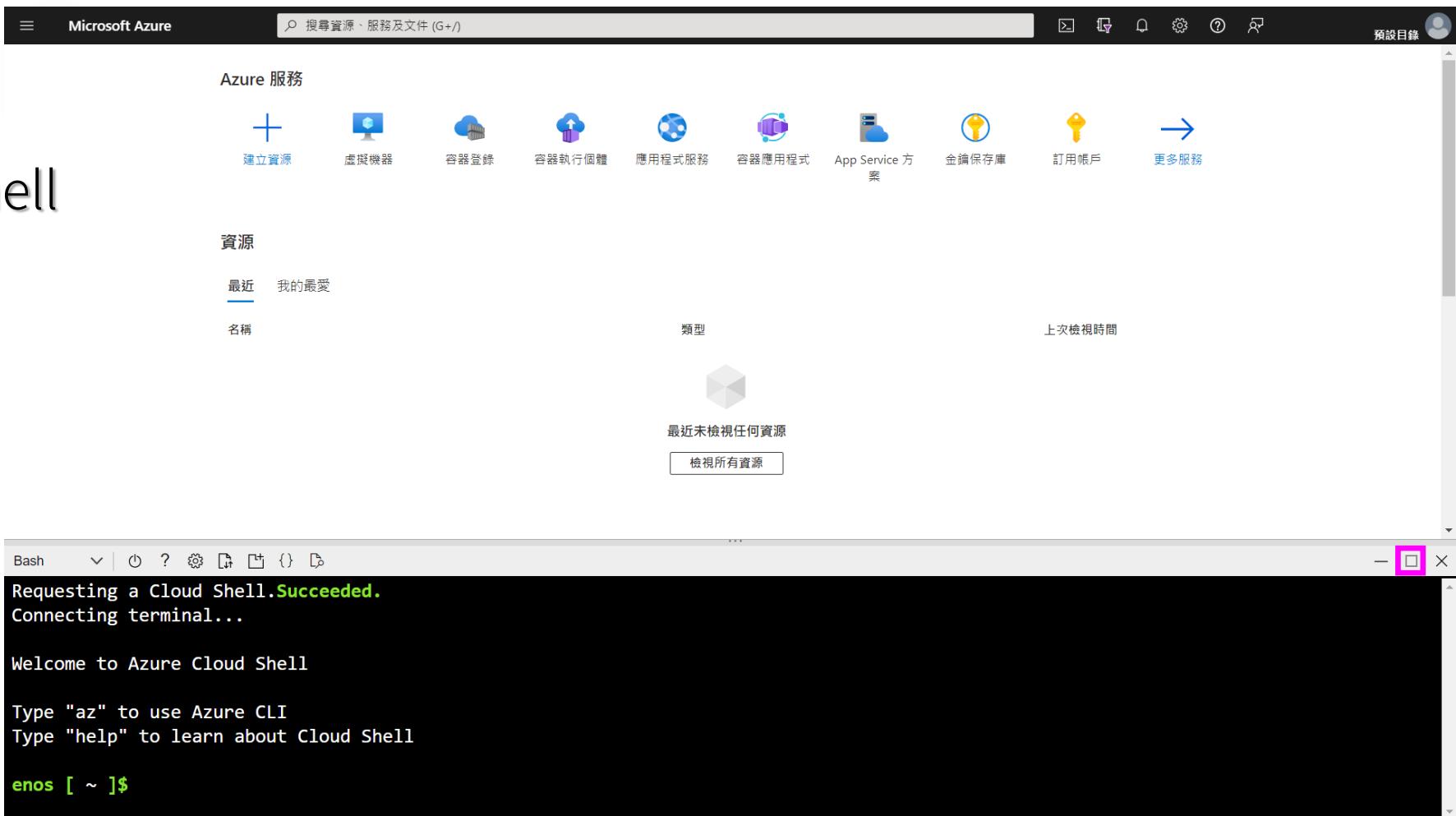
d. 啟動 Cloud Shell



# LINE Bot & WSGI

## 1. 準備程式碼

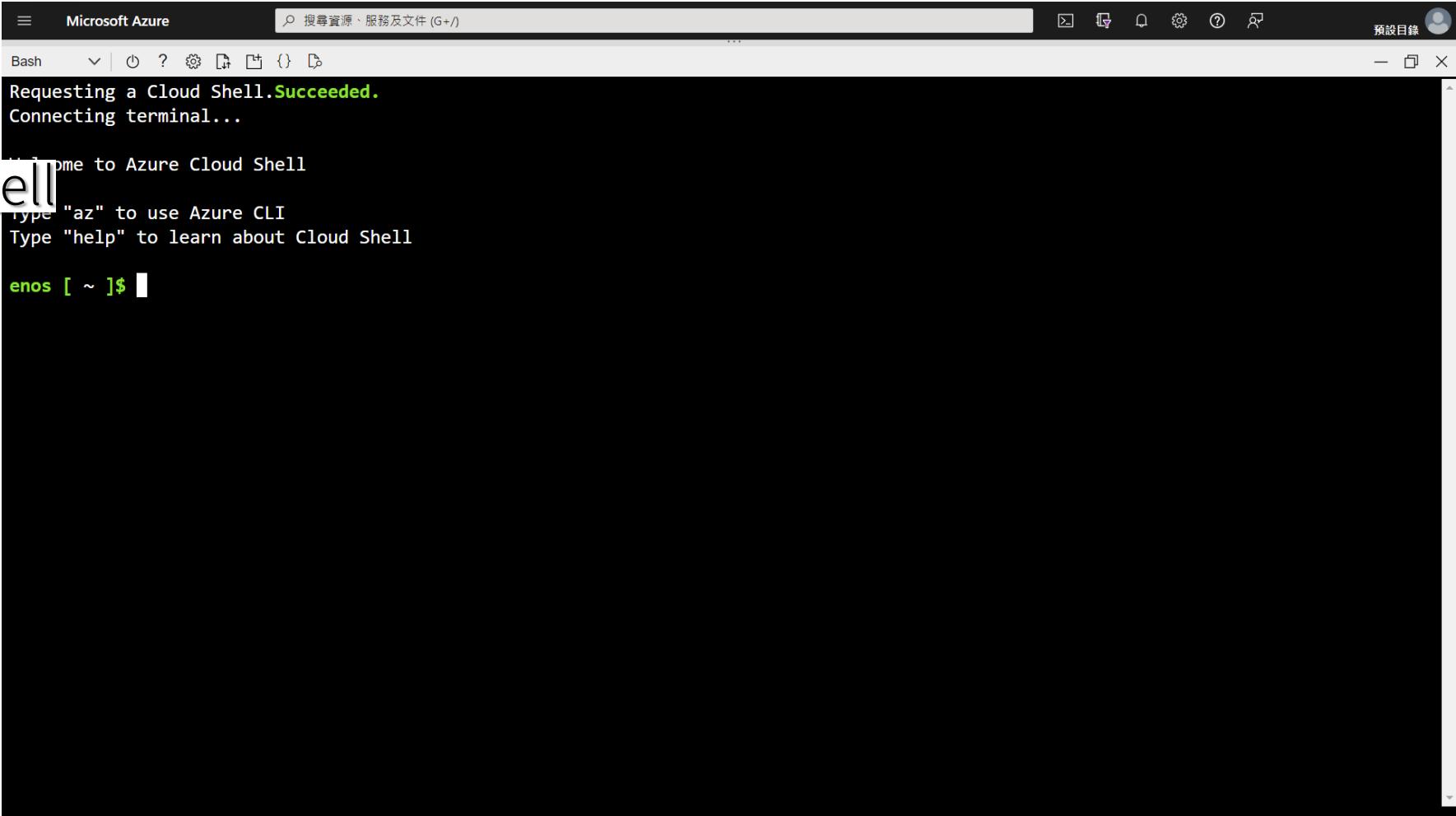
#### d. 啟動 Cloud Shell



# 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

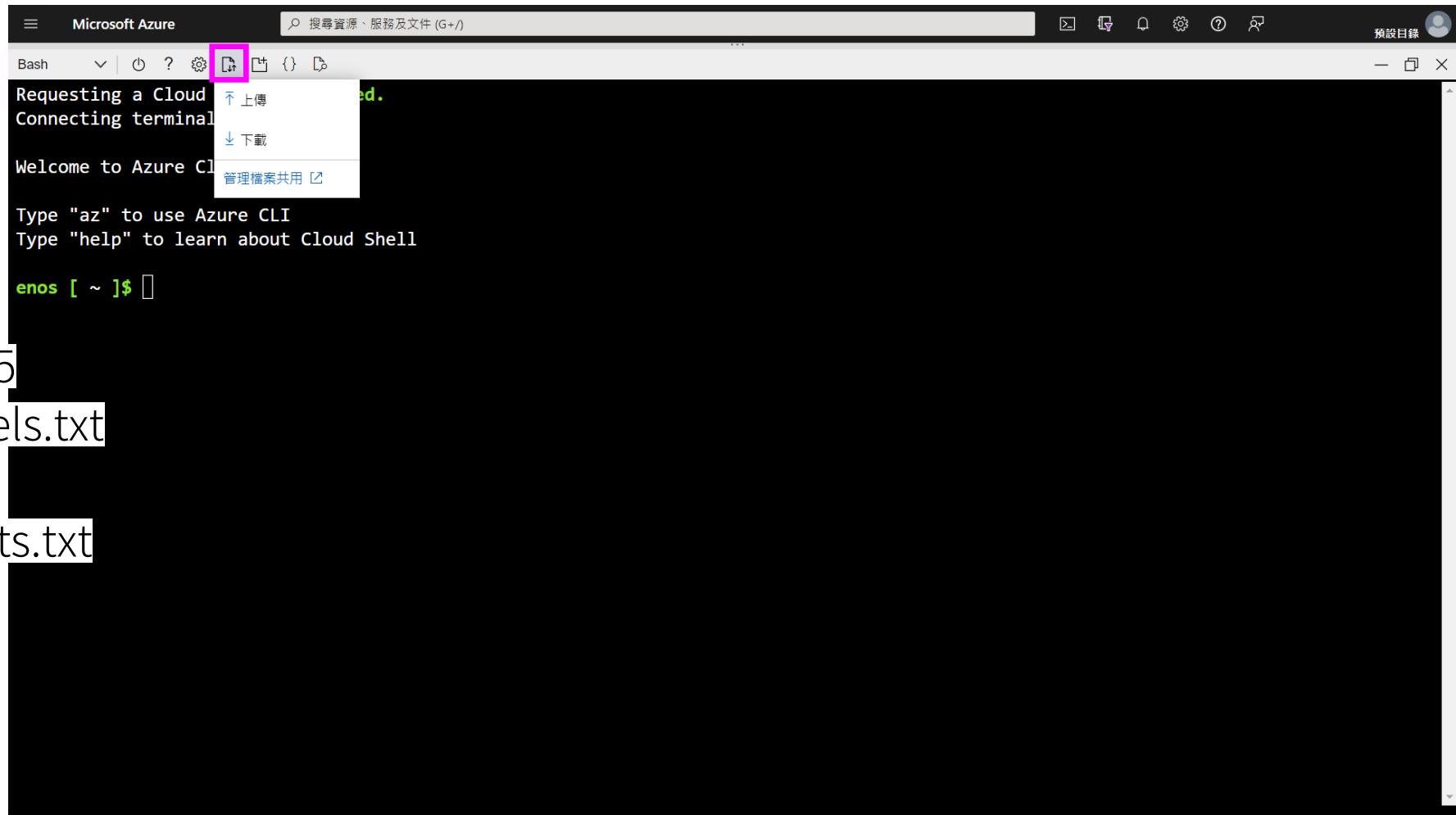
enos [ ~ ]$
```

# LINE Bot & WSGI

## 1. 準備程式碼

e. 上傳下列程式

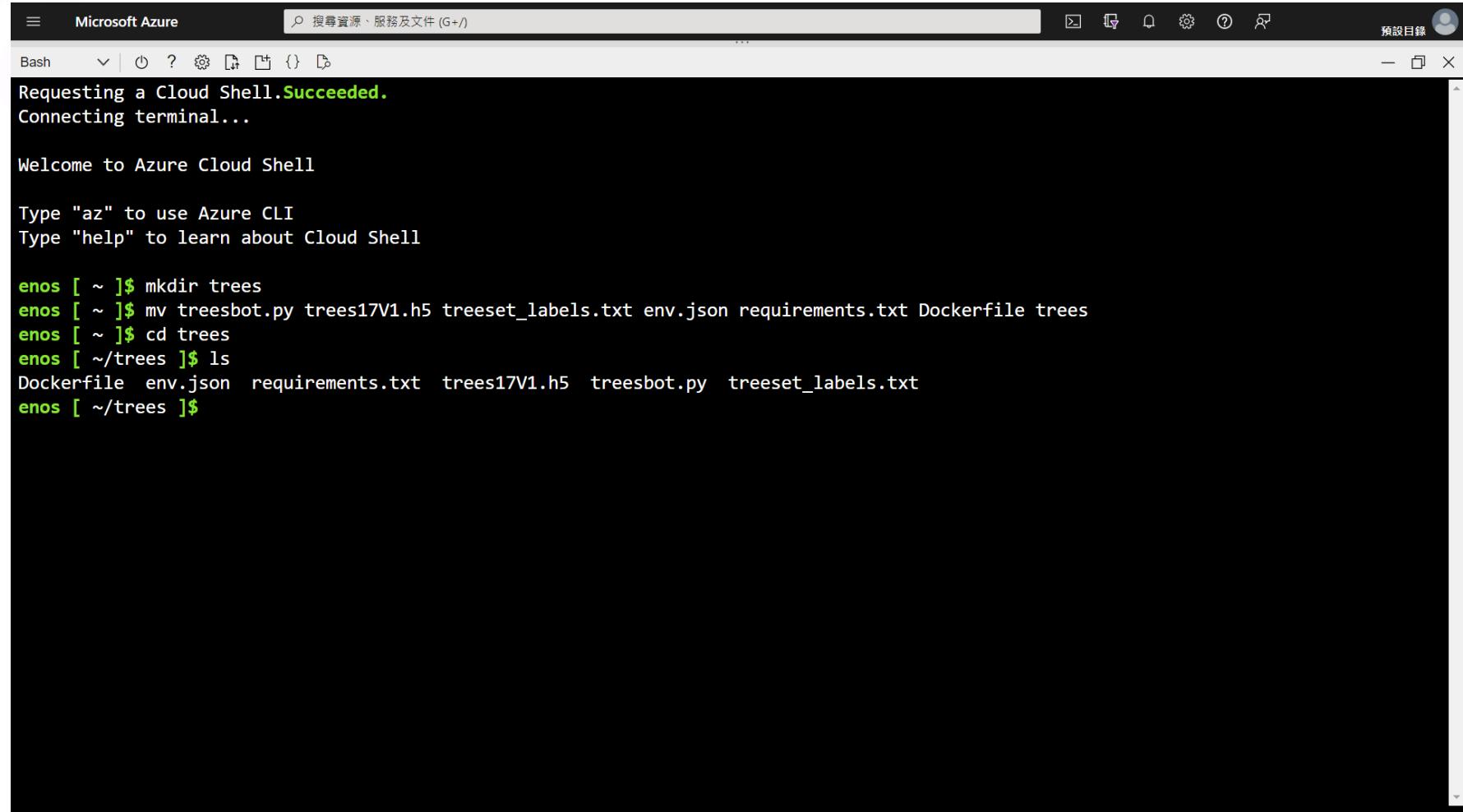
- ① treesbot.py
- ② trees17V1.h5
- ③ treeset\_labels.txt
- ④ env.json
- ⑤ requirements.txt
- ⑥ Dockerfile



# 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's a section for "Microsoft 提供的免費訓練" (Free Microsoft Training) with three items:

- 使用 ACR 建置及存取容器映像 (Build and access container images using ACR)
- 6 個單位 · 49 分鐘
- 使用 Docker 建置容器化 Web 應用程式 (Build containerized web applications using Docker)
- 8 個單位 · 57 分鐘
- 使用 Azure App Service 部署及執行容器化 ... (Deploy and run containerized applications using Azure App Service)
- 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 Groups), "所有資源" (All Resources), "儀表板" (Dashboard), "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).

# LINE Bot & WSGI

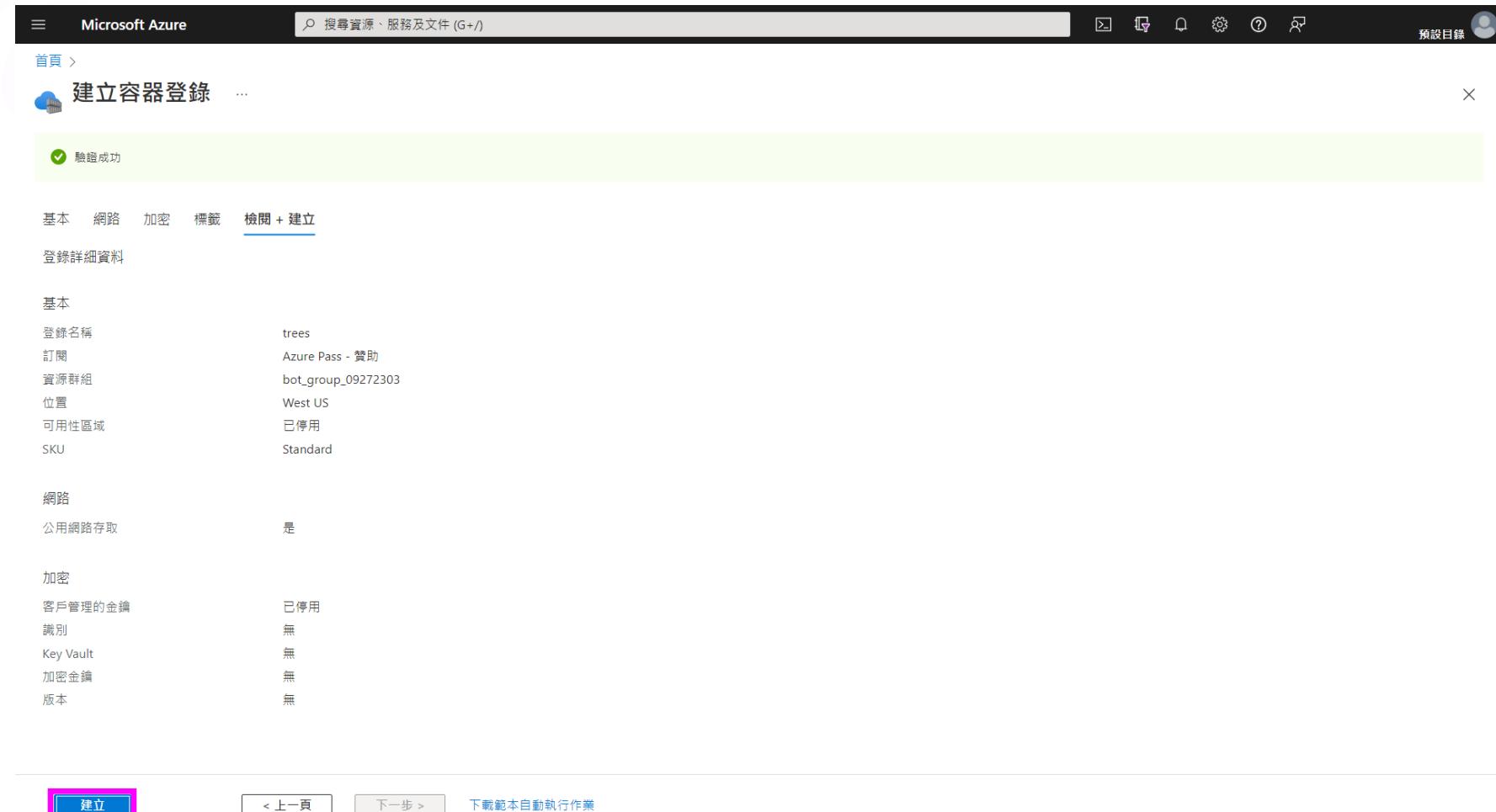
## 2. 啟動容器登入

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

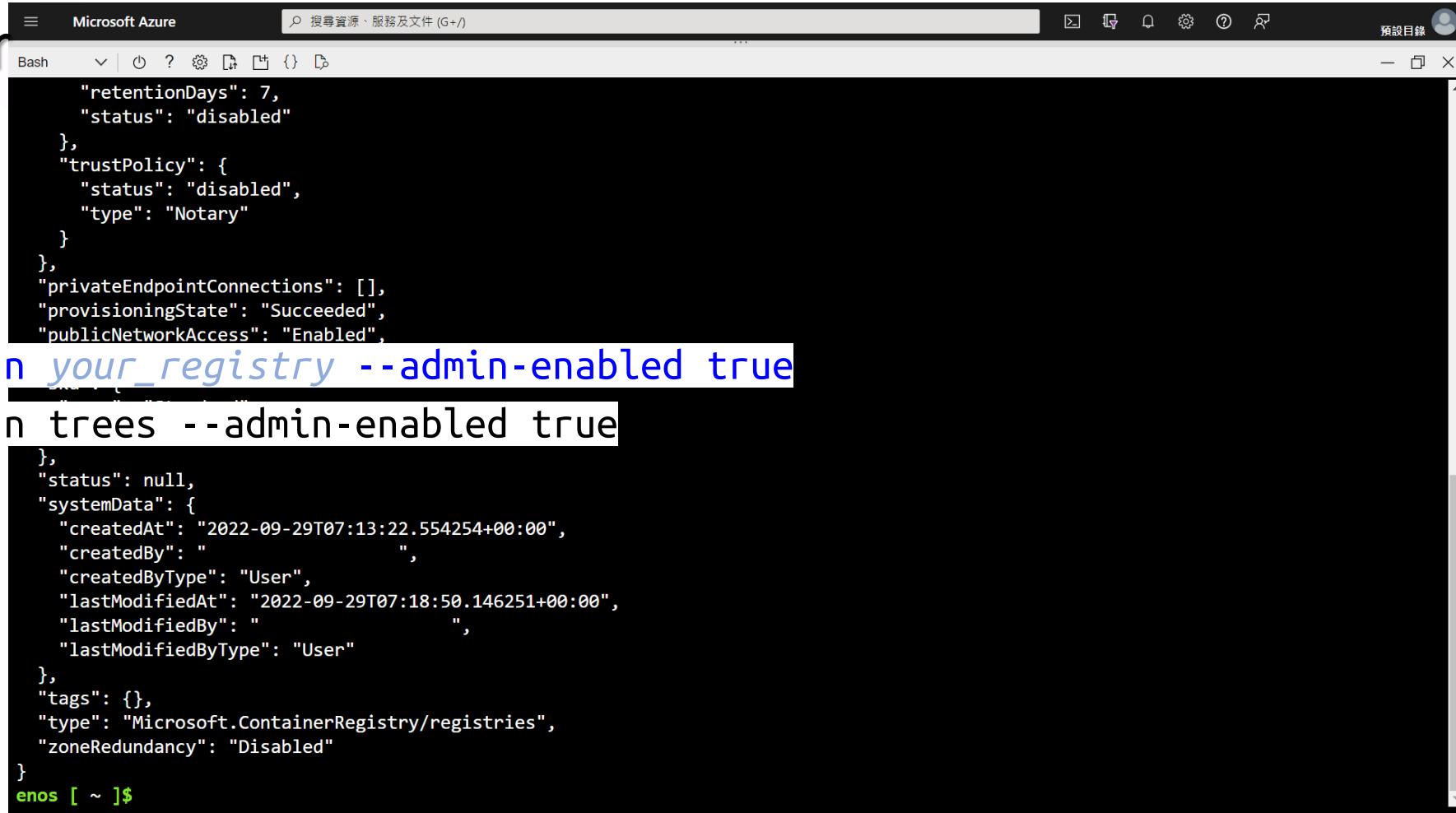
The screenshot shows the Microsoft Azure Container Registry deployment status page. The title bar reads "Microsoft Azure" and the main title is "Microsoft.ContainerRegistry | 概觀". The deployment status is shown as "您的部署已完成" (Deployment completed) with a green checkmark icon. Deployment details include: 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 located at the bottom of the main content area. 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

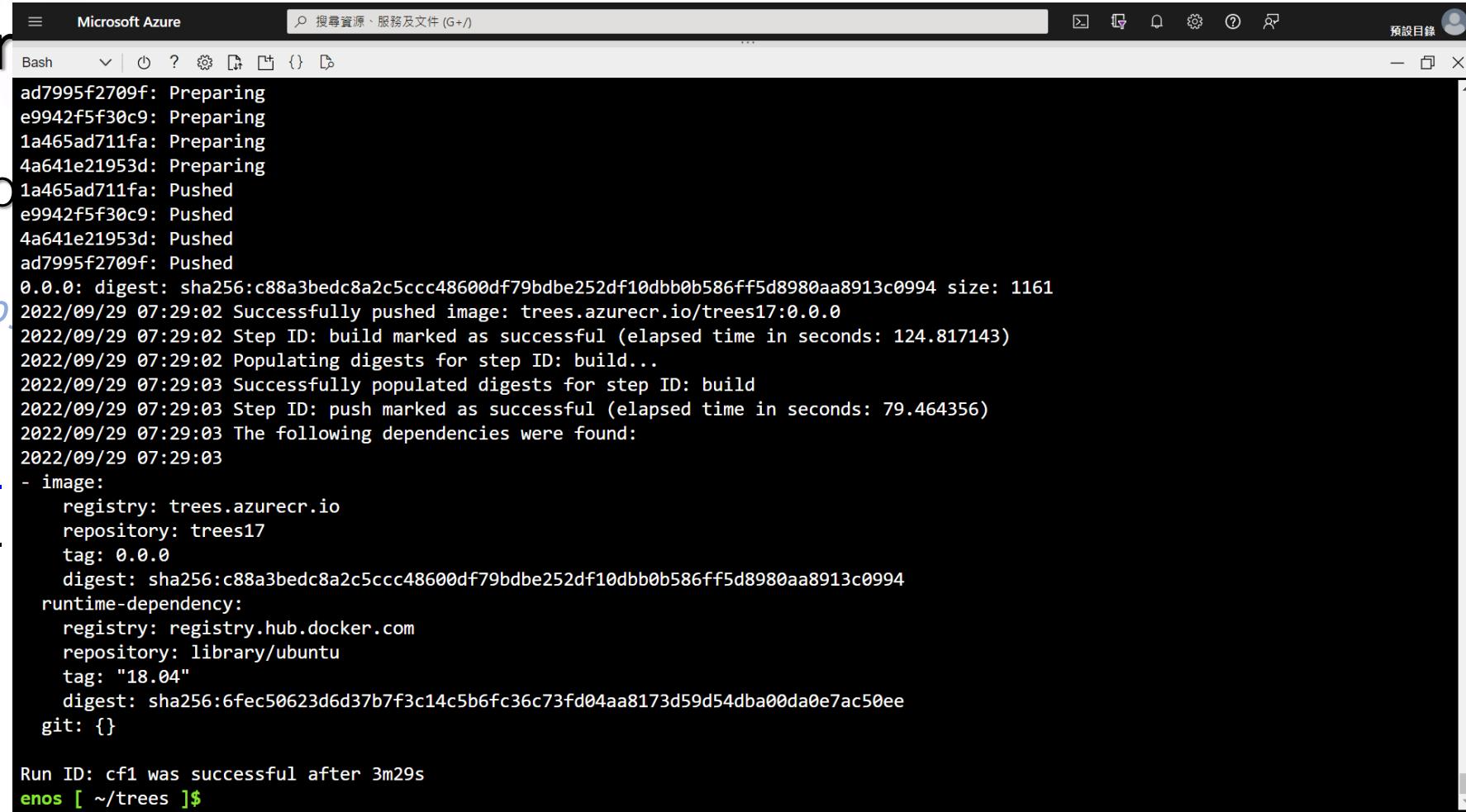
### b. 製作並推送 Container

```
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". 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), and "容器執行個體" (Container Instances) which is highlighted with a pink box. A tooltip for this highlighted icon says "容器執行個體" and "建立". To the right of the tooltip are "檢視" (View) and a star icon. Below the icons, there's a section titled "資源" (Resources) with tabs "最近" (Recent) and "我的最愛" (Favorites). The "最近" tab is selected. Under "最近", there are sections for "名稱" (Name) and "上次檢視時間" (Last checked time). A large central box contains a summary of Container Instances: "使用 Azure 容器執行個體可讓您在 Azure 中建立及管理 Docker 容器，而不需要設定虛擬機器或管理其他基礎結構。若要開始使用，請在 Azure 容器執行個體中建立容器。" At the bottom of this box is a "建立" (Create) button, also highlighted with a pink box. Below this box is a message "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. In the bottom left corner, there's a "瀏覽" (Browse) section 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). In the bottom right corner, there are four tool links: "Microsoft Learn" (Learn), "Azure 監視器" (Monitor), "適用於雲端的 Microsoft Defender" (Microsoft Defender for Cloud), and "成本管理" (Cost Management). A URL at the bottom of the page is <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' (Set up container runtime) step of the Azure Container Instance creation wizard. The interface includes tabs for '檢閱 + 建立' (Review + Create), '< 上一步' (Previous Step), and '下一步：進階 >' (Next Step: Advanced). The main area displays configuration options for the container runtime:

- 網路** (Networking):
  - 說明: 指定容器執行個體的網路連接。
  - 說明: 瀏覽器從三個網路選項中進行選擇：
  - 選項：
    - [公用] 則將為容器執行個體建立公用 IP 位址。
    - [私人] 則允許您為容器執行個體選擇新的或現有的虛擬網路。Windows 容器尚無法使用。
    - [無] 將不會建立公用 IP 或虛擬網路。您仍能使用命令列存取容器記錄檔。
  - 網路類型：選項為「公用」（已選中）。
  - DNS 名稱標籤：輸入框。
  - 重複使用 DNS 名稱標籤範圍：下拉選項為「任何重複使用 (不安全)」。

**連接埠** (Ports):
  - 連接埠：輸入框顯示「3000」，并被粉色高亮。
  - 連接埠通訊協定：下拉選項為「TCP」。

**備註** (Notes):  
此 PORT 不須另行開通防火牆

# 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. It displays the configuration for a container instance named 'linebot'. The '進階' (Advanced) tab is selected. In the '環境變數' (Environment Variables) section, there is a table with one row:

標示為安全	金鑰	值
否	PORT	3000

A pink box highlights the 'PORT' column. Below the table, there is a '命令覆寫' (Overwrite Command) input field containing '[]'. At the bottom, there are two radio buttons for '金鑰管理': 'Microsoft 管理的金鑰 (MMK)' (selected) and '客戶自控金鑰 (CMK)'. The bottom navigation bar includes buttons for '檢閱 + 建立' (Review + Create), '< 上一步' (Previous Step), and '下一步 : 檢籤 >' (Next Step : Tag).

以環境變數指定 Container 使用的 PORT

# 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 configuration is as follows:

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

Below the basic configuration, there are sections for '網路' (Networking) and '進階' (Advanced). The '建立' (Create) button is highlighted in blue at the bottom left.

# 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 summary, 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

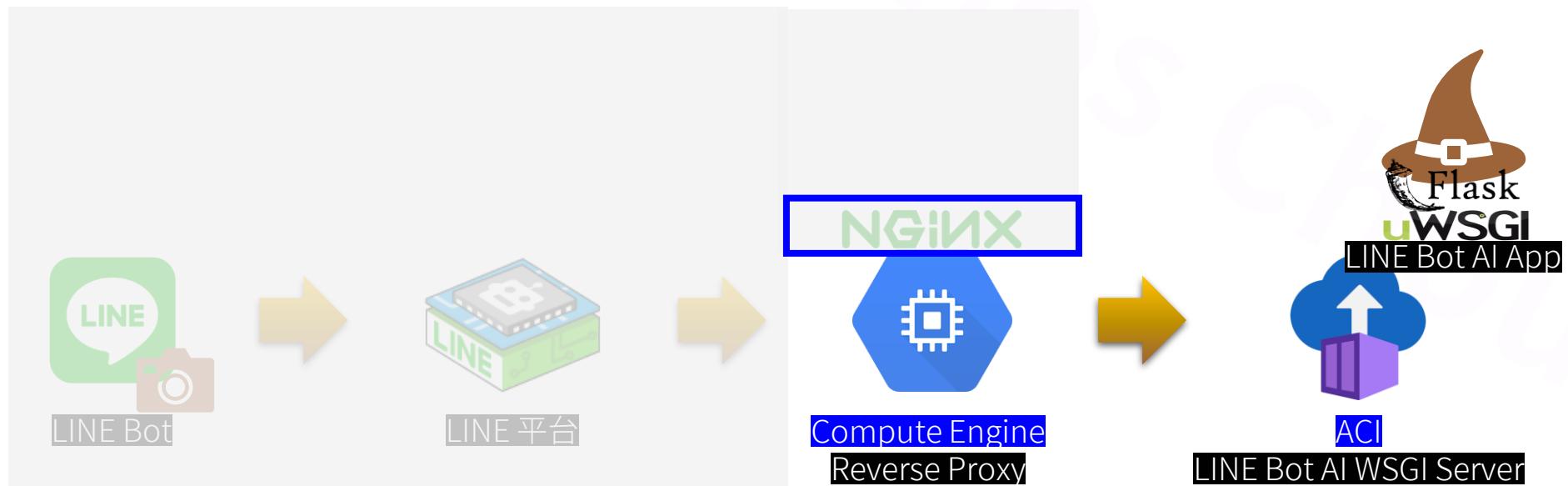
CPU

記憶體

接收的網路位元組

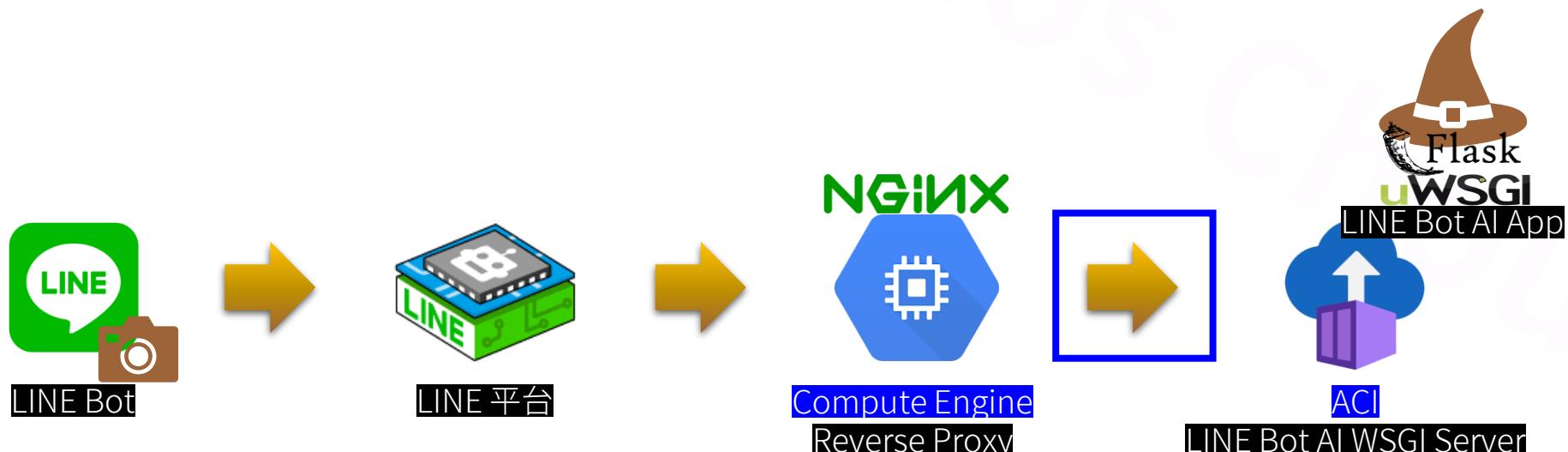
傳輸的網路位元組

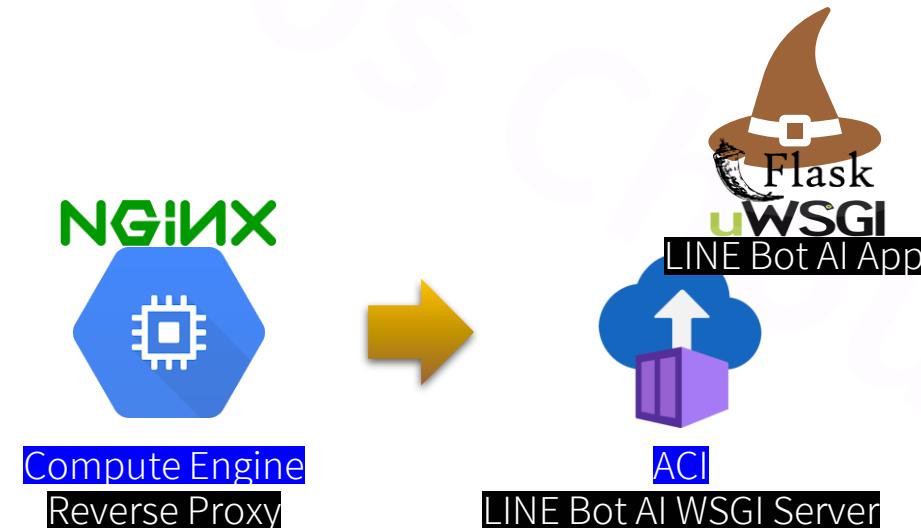
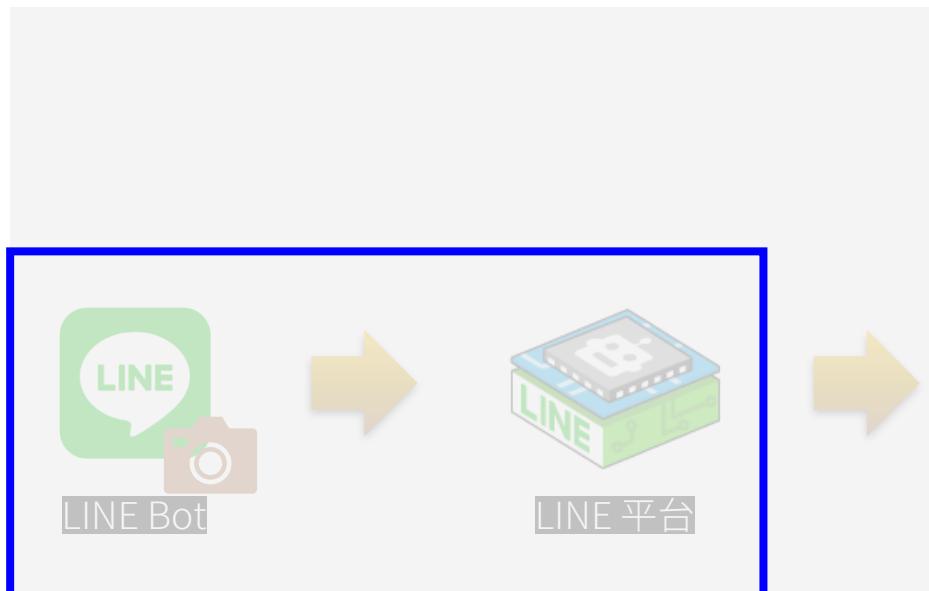
JSON 檢視



# 網站與憑證 - Your Turn

嘗試由 NGINX @VM 轉導 LINE Message 至 LINE Bot + WSGI @ACI

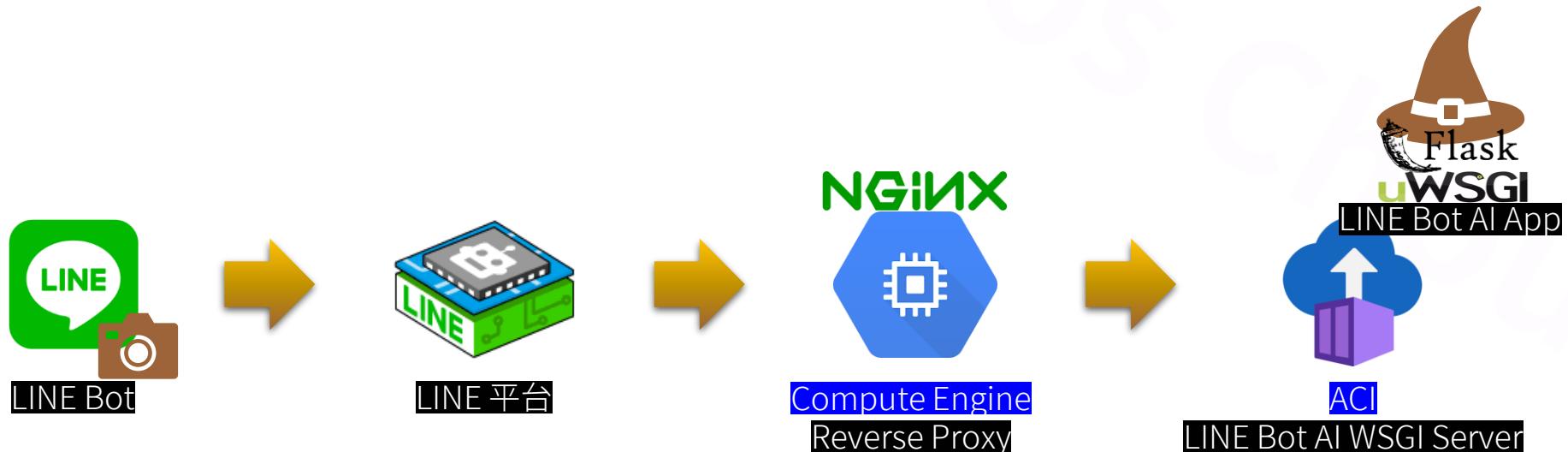




# 設定 LINE Messaging

## 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 sidebar with options like '建立', '管理檢視', and '...'. The main area shows a container named 'treeswsgi' with a status of 'Running'. Below the container name, there's a search bar and a '重新整理' (refresh) button. A table lists the container details: name (treeswsgi), image (trees.azurecr.io/trees17...), state (Running), previous state (-), start time (2022-09-29T07:53:45.82...), and restart count (0). On the right, there's a detailed log viewer with tabs for '事件' (Events), '屬性' (Properties), '記錄' (Logs), and '連線' (Connections). The '記錄' tab is selected, displaying the log output:

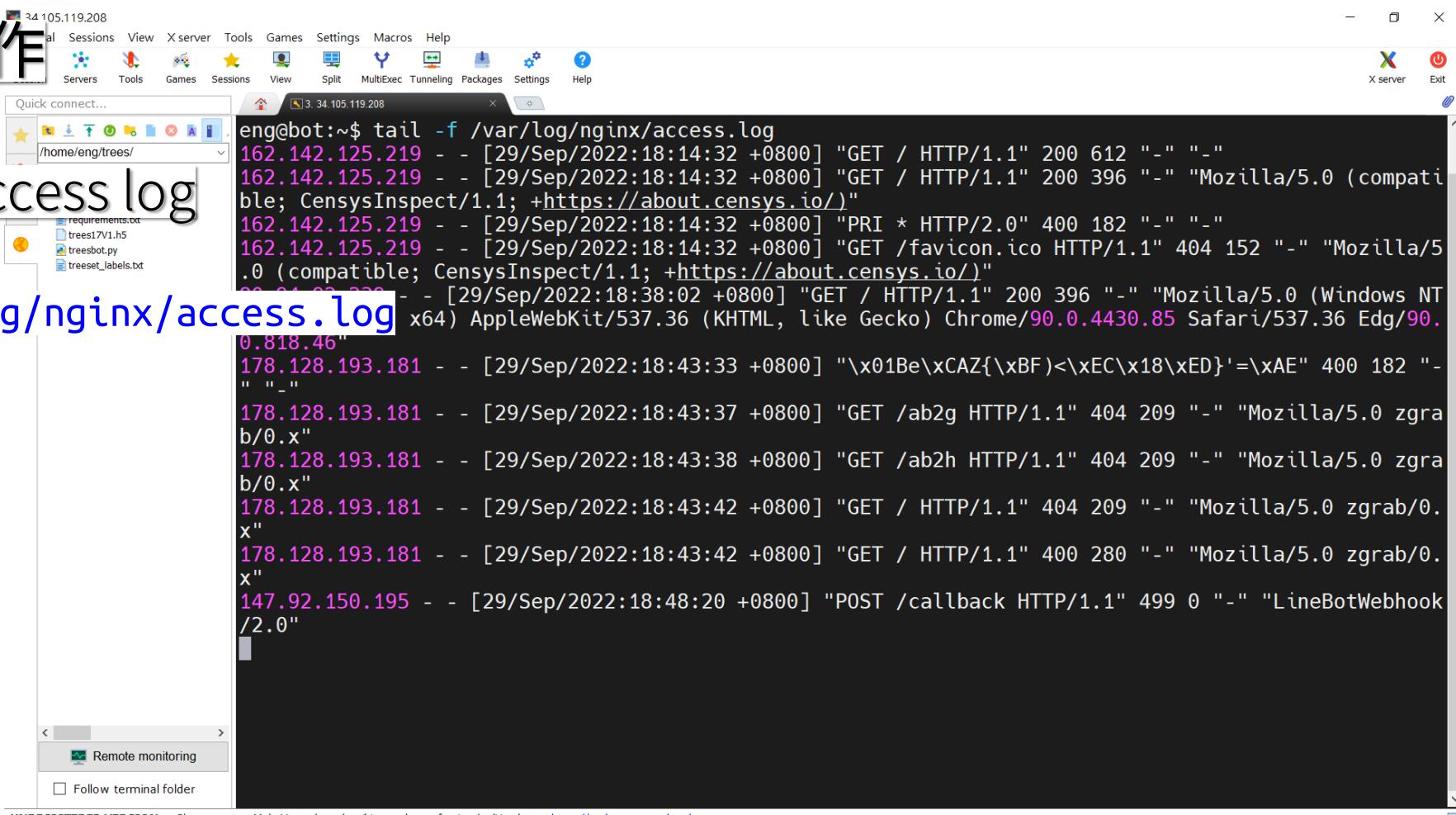
```
*** Starting uWSGI 2.0.20 (64bit) on [Thu Sep 29 15:53:46 2022] ***
compiled with version: 7.5.0 on 29 September 2022 07:26:47
os: Linux-5.10.102.2-microsoft-standard #1 SMP Mon Mar 7 17:36:34 UTC 2022
nodename: SandboxHost-638000346905459478
machine: x86_64
clock source: unix
detected number of 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
```



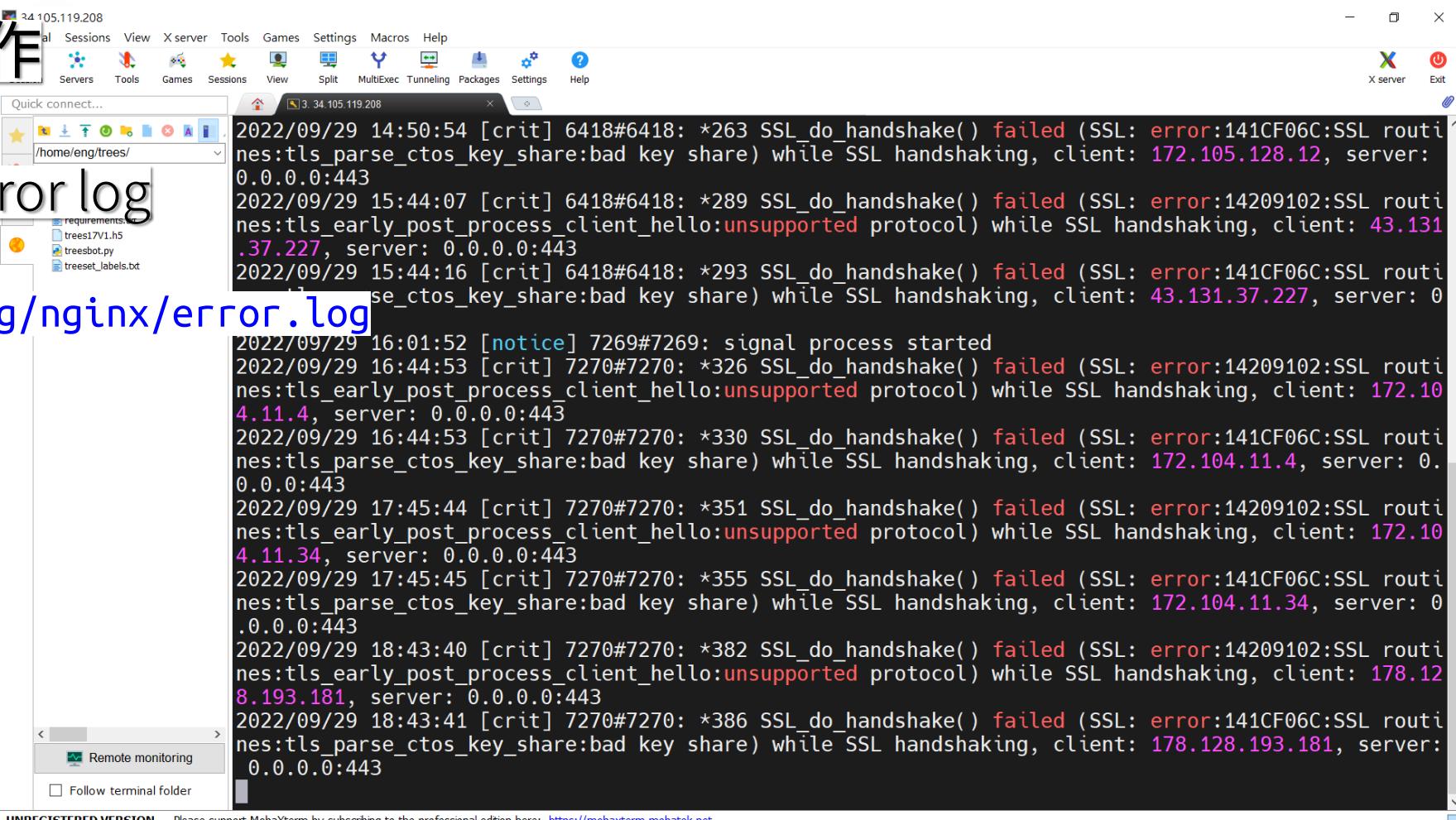
```
eng@bot:~$ tail -f /var/log/nginx/access.log
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 612 "-" "-"
162.142.125.219 - - [29/Sep/2022:18:14:32 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (compatible; CensysInspect/1.1; +https://about.censys.io/)"
162.142.125.219 - - [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
```

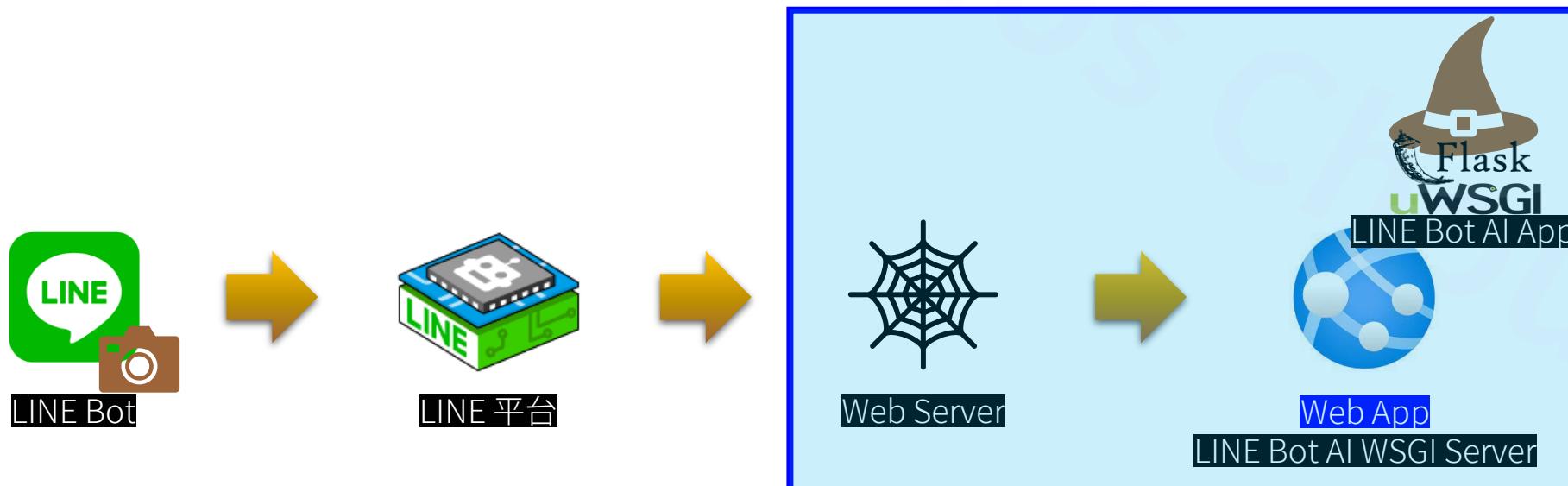


```
2022/09/29 14:50:54 [crit] 6418#6418: *263 SSL_do_handshake() failed (SSL: error:141CF06C:SSL routines:tls_parse_ctos_key_share:bad key share) while SSL handshaking, client: 172.105.128.12, server: 0.0.0.0:443
2022/09/29 15:44:07 [crit] 6418#6418: *289 SSL_do_handshake() failed (SSL: error:14209102:SSL 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 - Serverless 部署

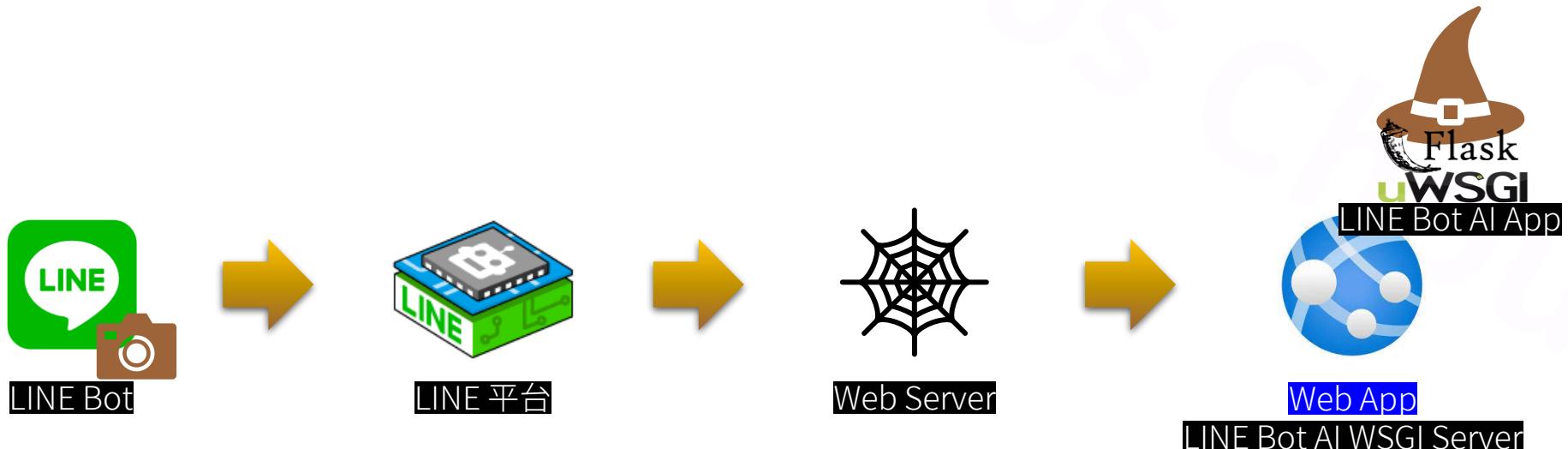
1. 長期可用且廉價的硬體環境 ... [Azure Web App](#)
2. Flask as Web Server 的替代方案 ... [Azure Web App](#)
3. 長期可用且廉價的 SSL 網域方案 ... [Azure Web App](#)

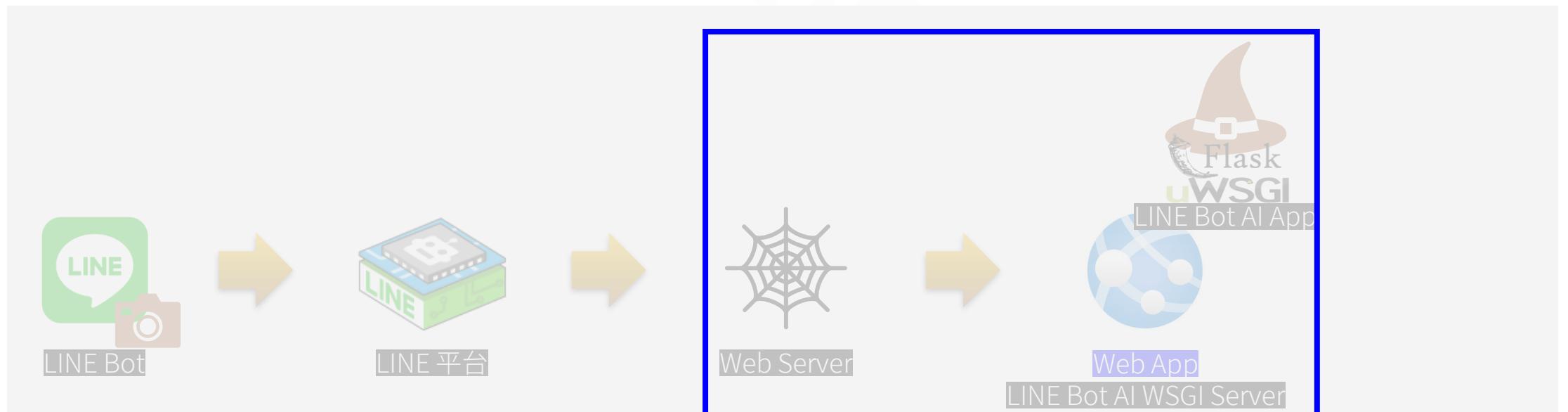


# 任務

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)

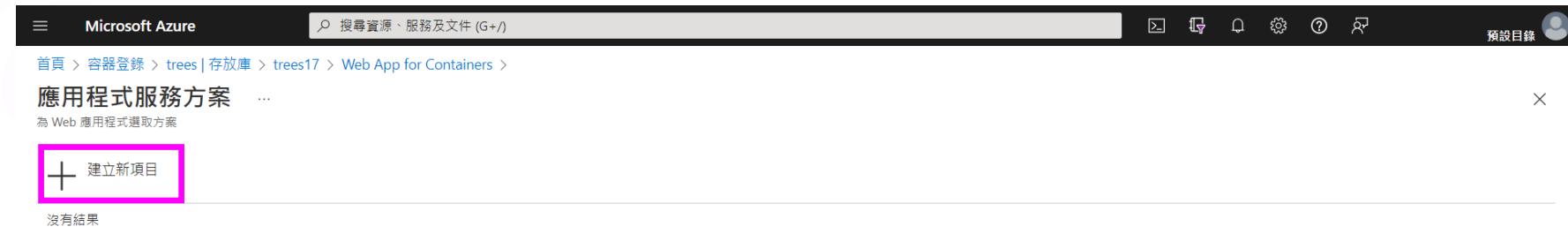
A note at the bottom states: "這會建立名為 'treesapp150116' 的 Webhook 供持續部署之用。Webhook 將會在登錄的當前位置 westus 中建立。"

At the bottom right of the main form 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



### 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 with a 'trees17' entry. 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 page 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.

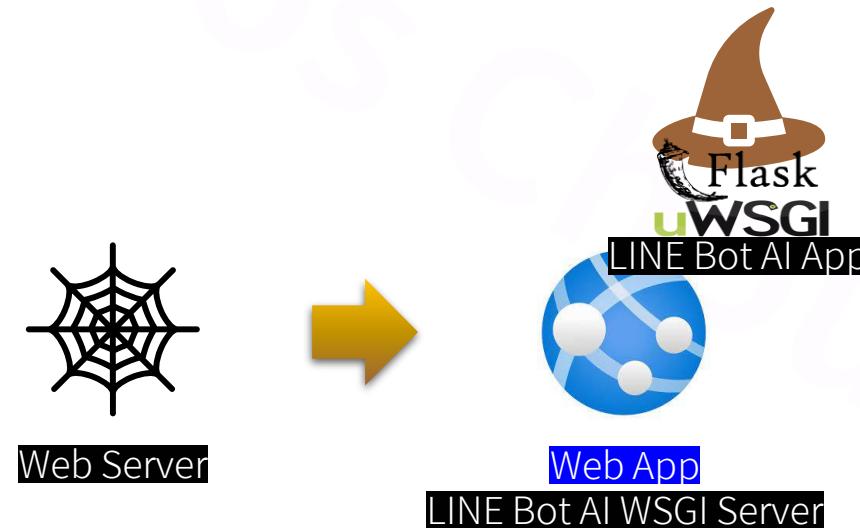
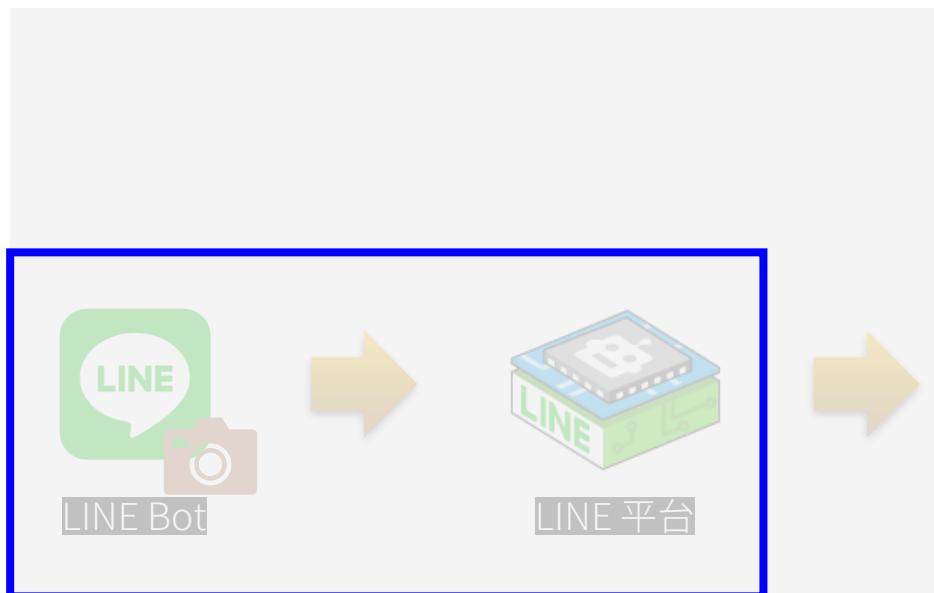
a. 由 Container 建立應用程式服務

# 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 a Web App named 'treesapp'. The main title bar says 'Microsoft Azure' and the sub-header says '搜尋資源、服務及文件 (G+ /)'. The left sidebar has sections like '概述', '活動記錄', '存取控制 (IAM)', and '標籤'. The main content area shows the app's status as 'Running' in 'West US'. It displays the 'LINE Messaging Webhook URL' as <https://treesapp.azurewebsites.net>. Below this, there are sections for 'HTTP 5xx', '資料輸入', and '連出的資料', each showing a chart of data over time. A pink box highlights the 'LINE Messaging Webhook URL'.

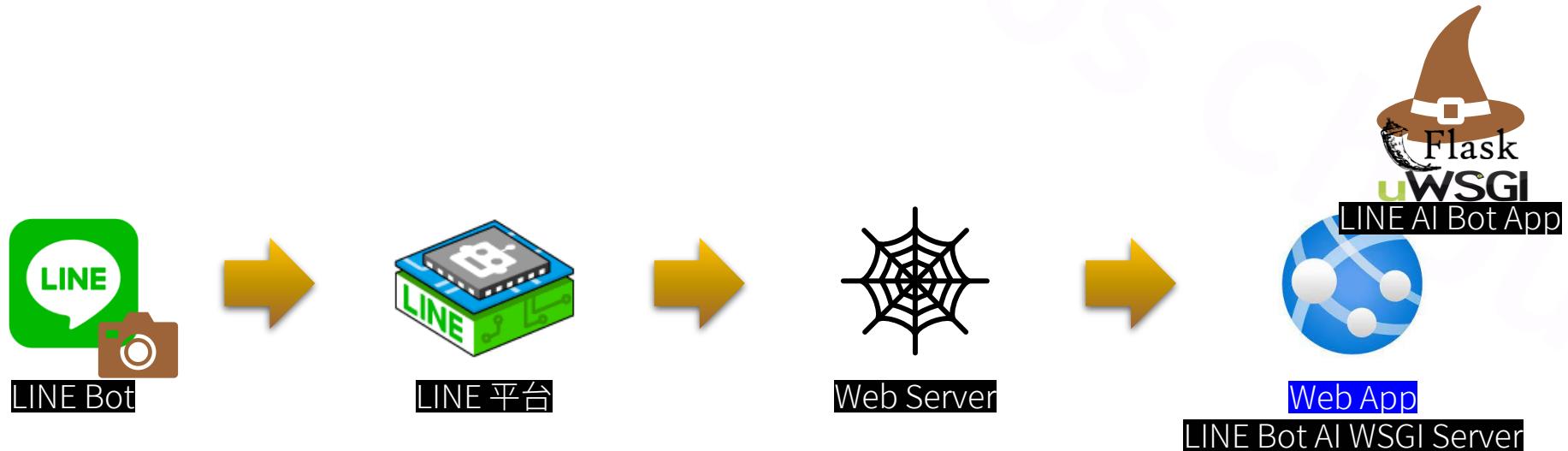


# 設定 LINE Messaging

## 1. 調整 LINE Messaging

a. 調整 Webhook URL

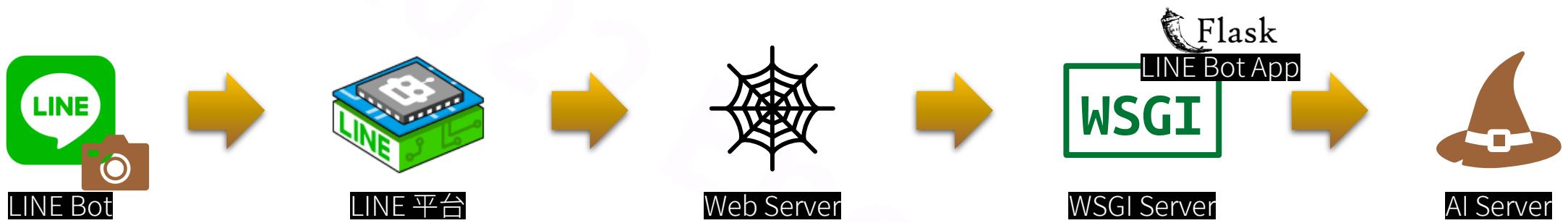
b. LINE 測試



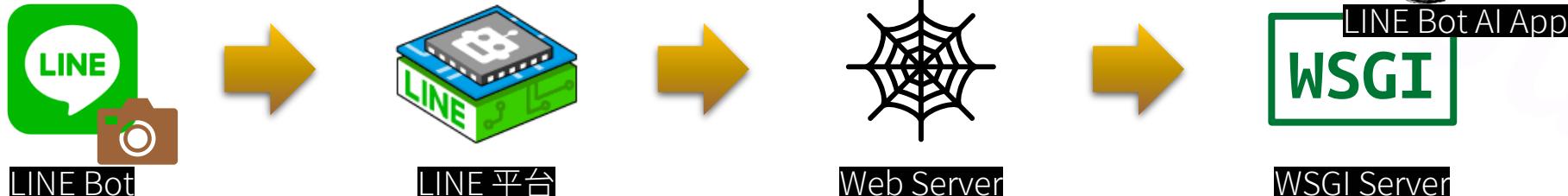
# LINE Bot 部署 + AI 部署

# Architecture

## Scalable Architecture

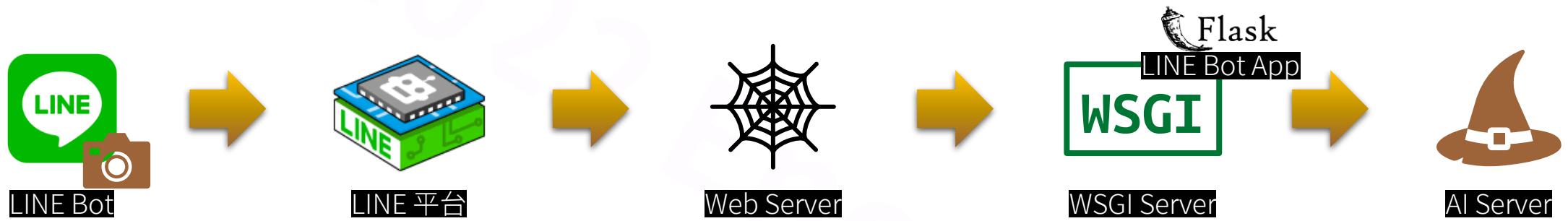


## Reasonable Architecture

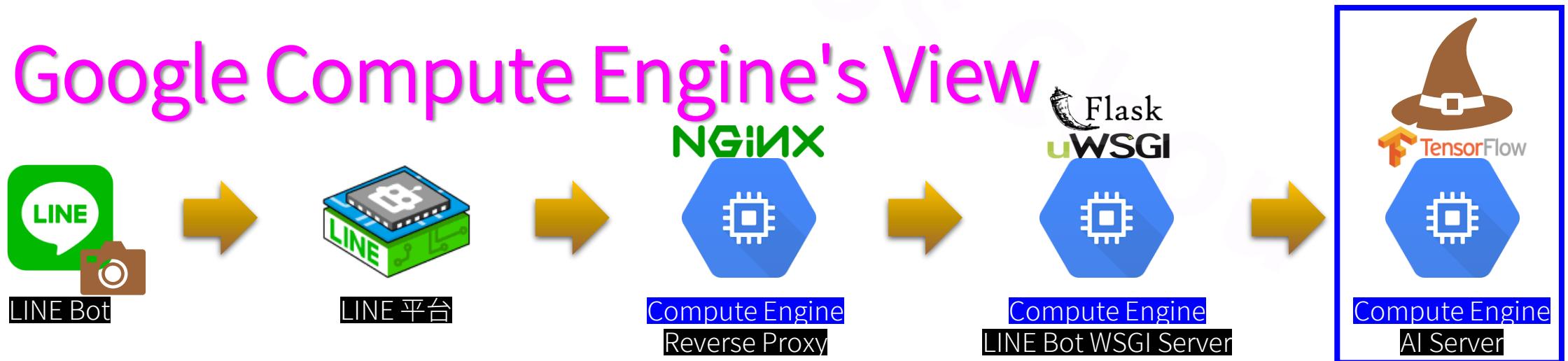


# 部署 TensorFlow Serving - VM

## Scalable Architecture

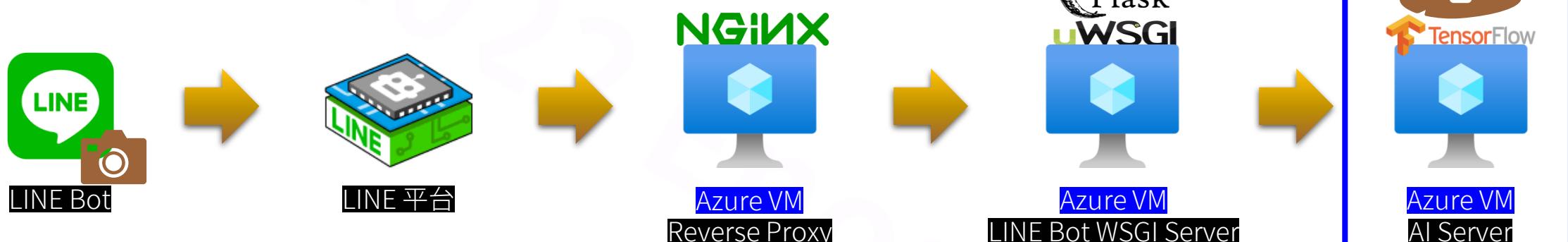


## Google Compute Engine's View

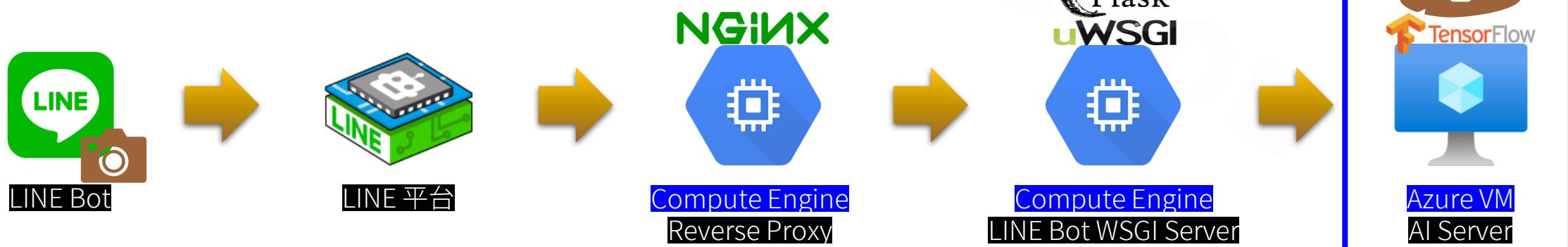


# 部署 TensorFlow Serving - VM

## 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

# 部署 TensorFlow Serving - VM

## 2. 準備模型

### b. 上傳 SavedModel 模型至 VM

- 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

# Cloud Shell 上傳至 VM 範例

- ① 上傳 VM 之 OpenSSH 格式私鑰 *your\_keyfile.pem* 至 Cloud Shell
- ② `chmod 400 your_keyfile.pem`
- ③ `scp -i your_keyfile.pem -r your_model/1 your_account@your_vmid:your_model`

# 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 # 單獨以 --no-deps 安裝可避免自動安裝或更新 TensorFlow

# 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
```

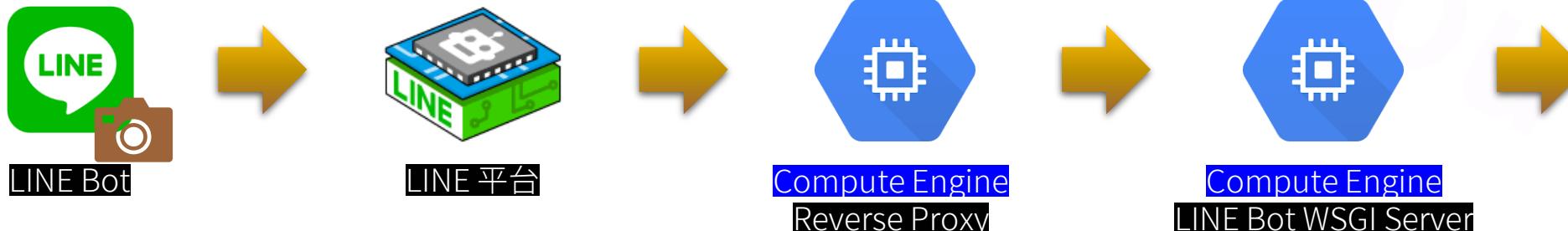
```
python tf-serving-test.py --host 34.168.152.9 --port 8500 --protocol gRPC --
                           input input_4 --output dense_1 --labels treeset_labels.txt trees 鳳凰木.jpg
```

# 部署 TensorFlow Serving - Cloud Run

## Mixed View



## GCP's View



# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### a. 轉換模型為 TensorFlow SavedModel 格式

- ① 準備 TensorFlow 2.4.4 以上環境

```
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 - Cloud Run

## 1. 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 - Cloud Run

## 1. 準備模型

### a. 轉換模型為 TensorFlow SavedModel 格式

③ 轉換模型

```
hdf5_to_savedmodel.py your_hdf5.h5 your_savedmodel  
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

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### b. 將 SavedModel 置於模型目錄

- ① 建立模型目錄
- ② 上傳或複製 SavedModel 至模型目錄

# 部署 TensorFlow Serving - Cloud Run

## 1. 準備模型

### b. 將 SavedModel 置於模型目錄

# 路徑範例

*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

```
# Dockerfile 範例
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
```

#### Note

此範例採用 gRPC，可自行調整為 RESTful API

# 部署 TensorFlow Serving - Cloud Run

## 2. 製作 Container

### b. 將 Dockerfile 置於模型目錄同層級

```
# 路徑示意  
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；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). A '篩選' (Filter) button is also present. The main content area displays a table with one row for a service named 'treestfs'. The columns include '名稱' (Name), '每秒要求數' (Requests per second), '區域' (Region), '驗證' (Verification), '輸入流量' (Input traffic), '最近部署時間' (Last deployment time), and '部署者' (Deployer). The 'treestfs' row shows a green checkmark in the 'Name' column, a value of '0' in the 'Requests per second' column, 'us-west1' in the 'Region' column, and '允許在未經驗證的情況下叫用' (Allow unverified calls) in the 'Verification' column.

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)

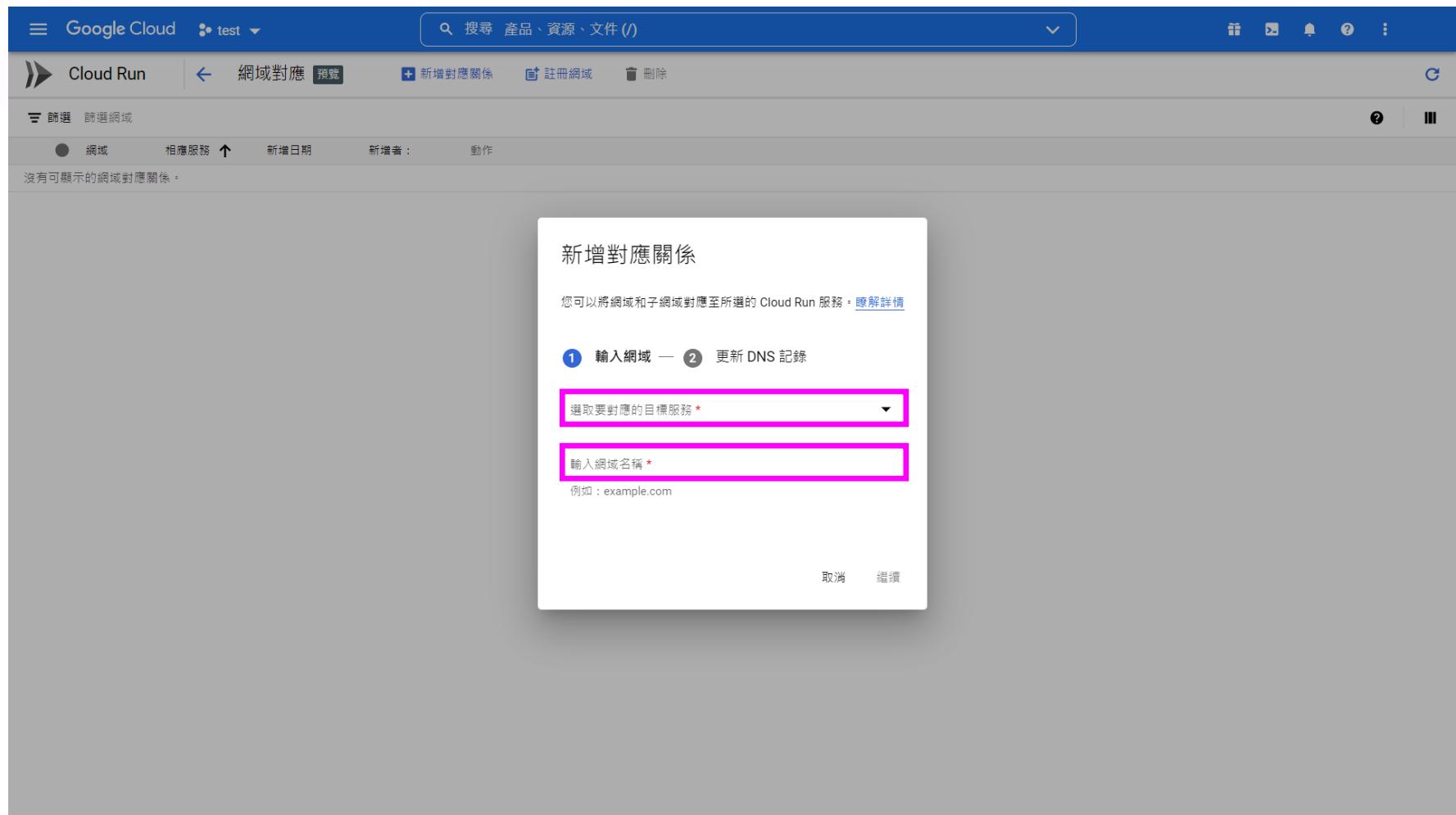
### a. 驗證網域

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 navigation 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: '網域' (Domain), '相應服務' (Associated Service), '新增日期' (Created Date), '新增者' (Creator), and '動作' (Actions). A note at the bottom states '沒有可顯示的網域對應關係' (No domain mappings to display).

# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換 Domain)

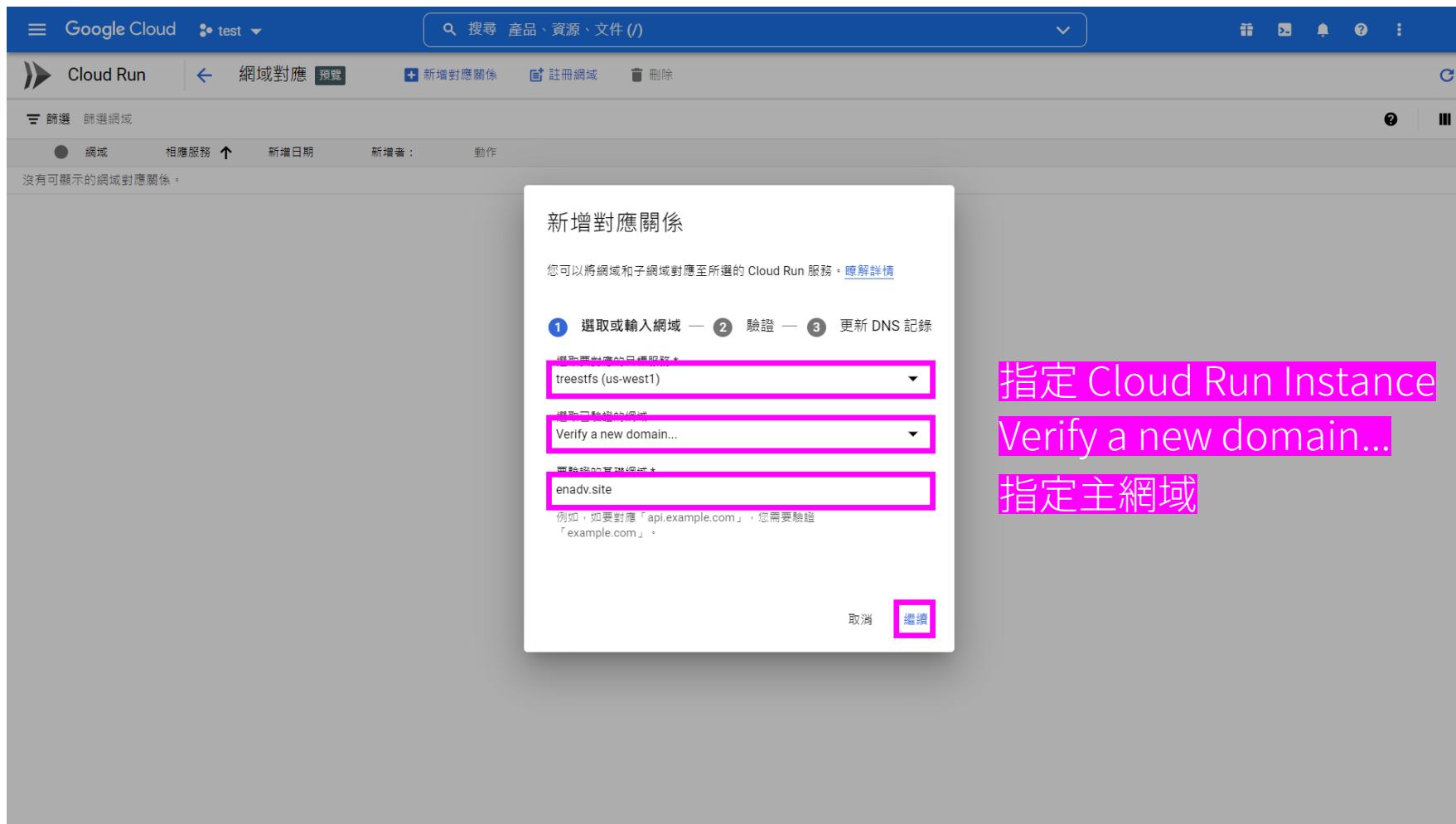
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

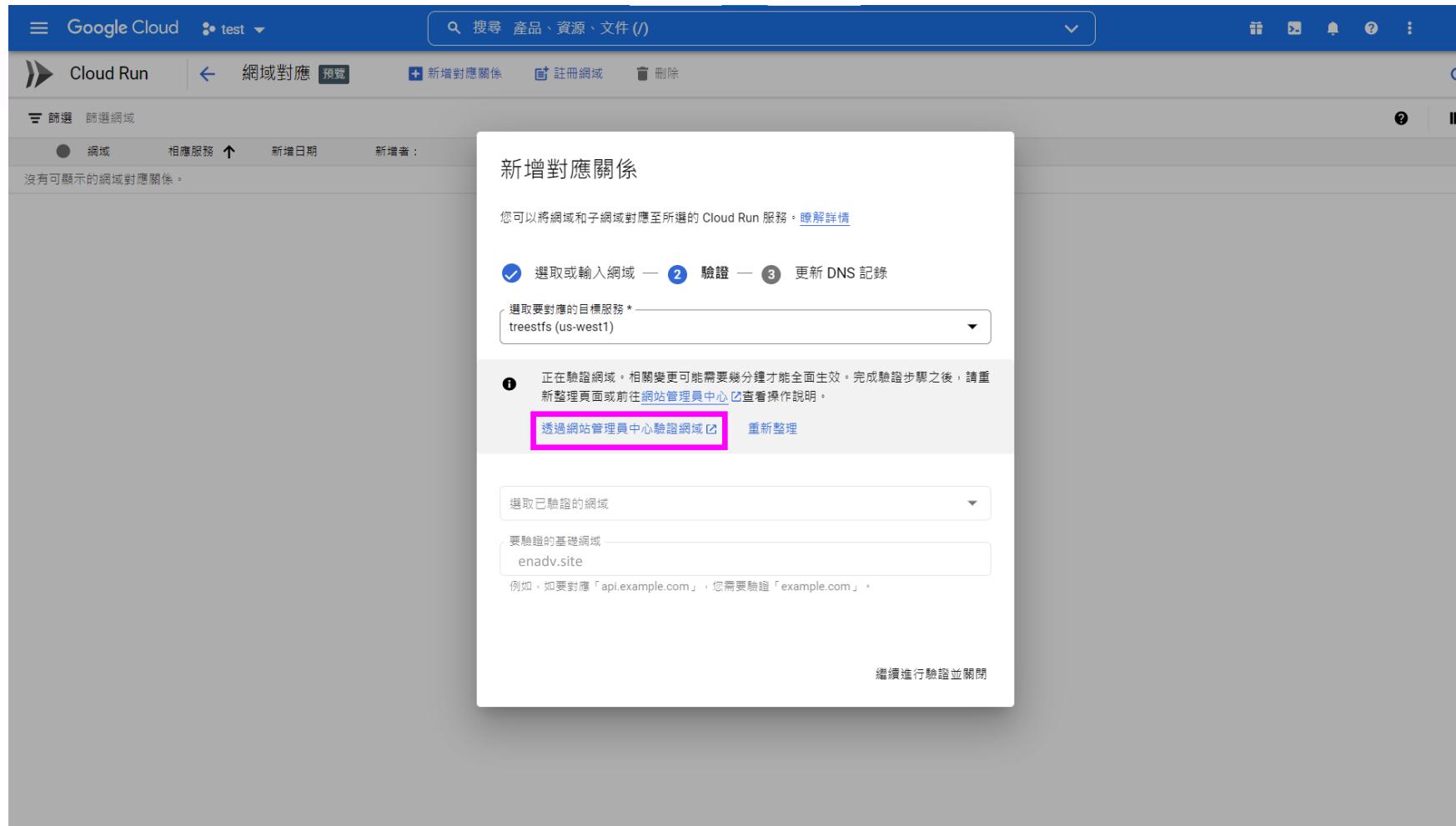
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

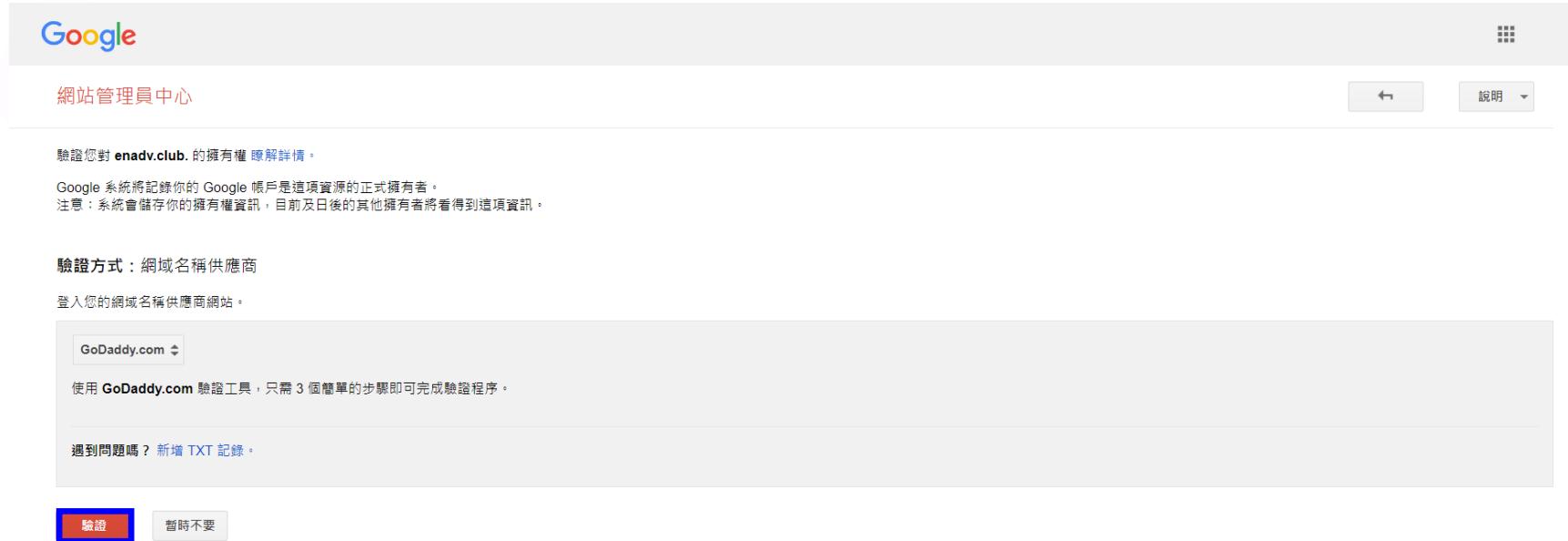
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

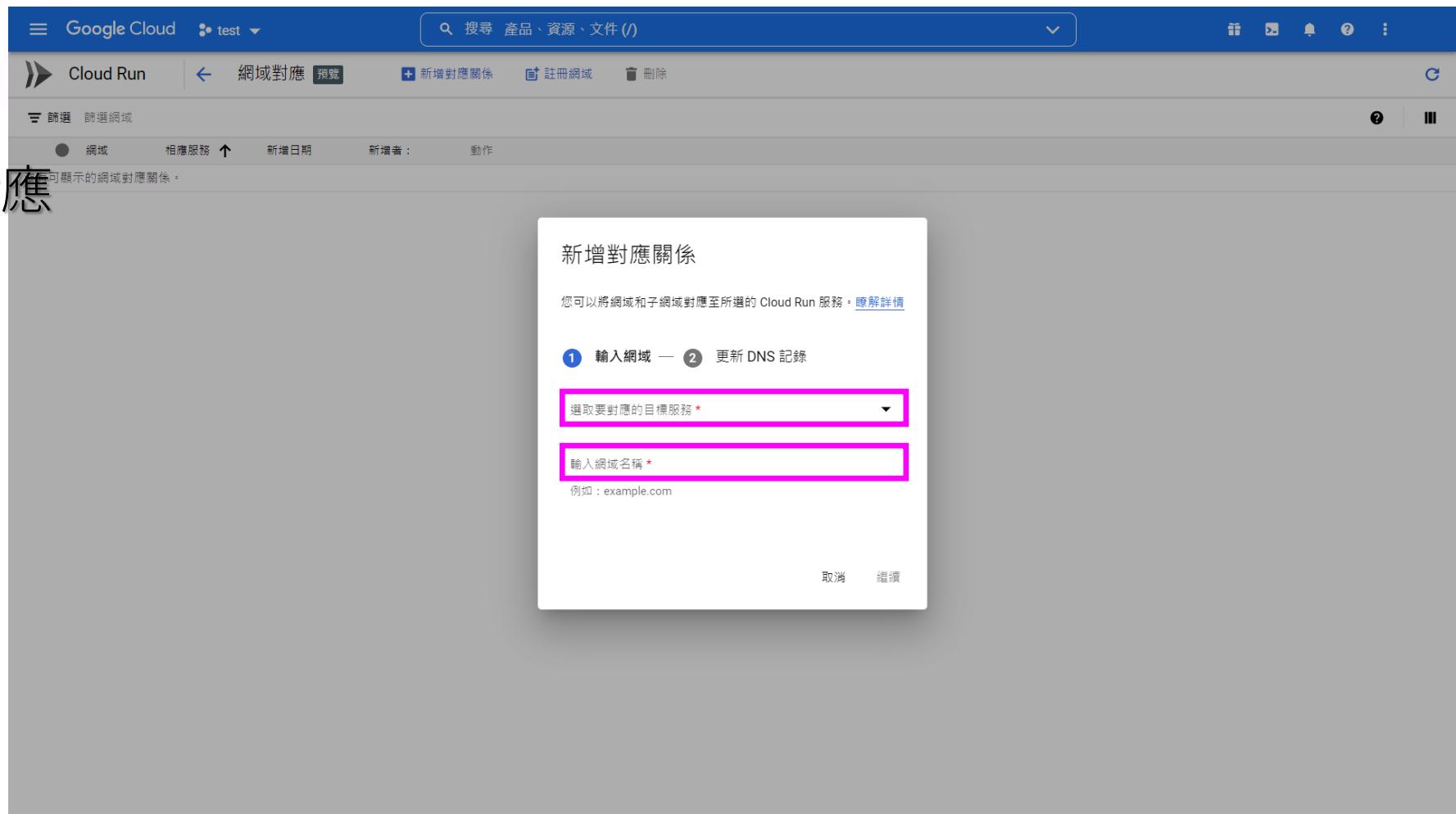
### a. 驗證網域



# 部署 TensorFlow Serving - Cloud Run

## 4. (轉換網域)

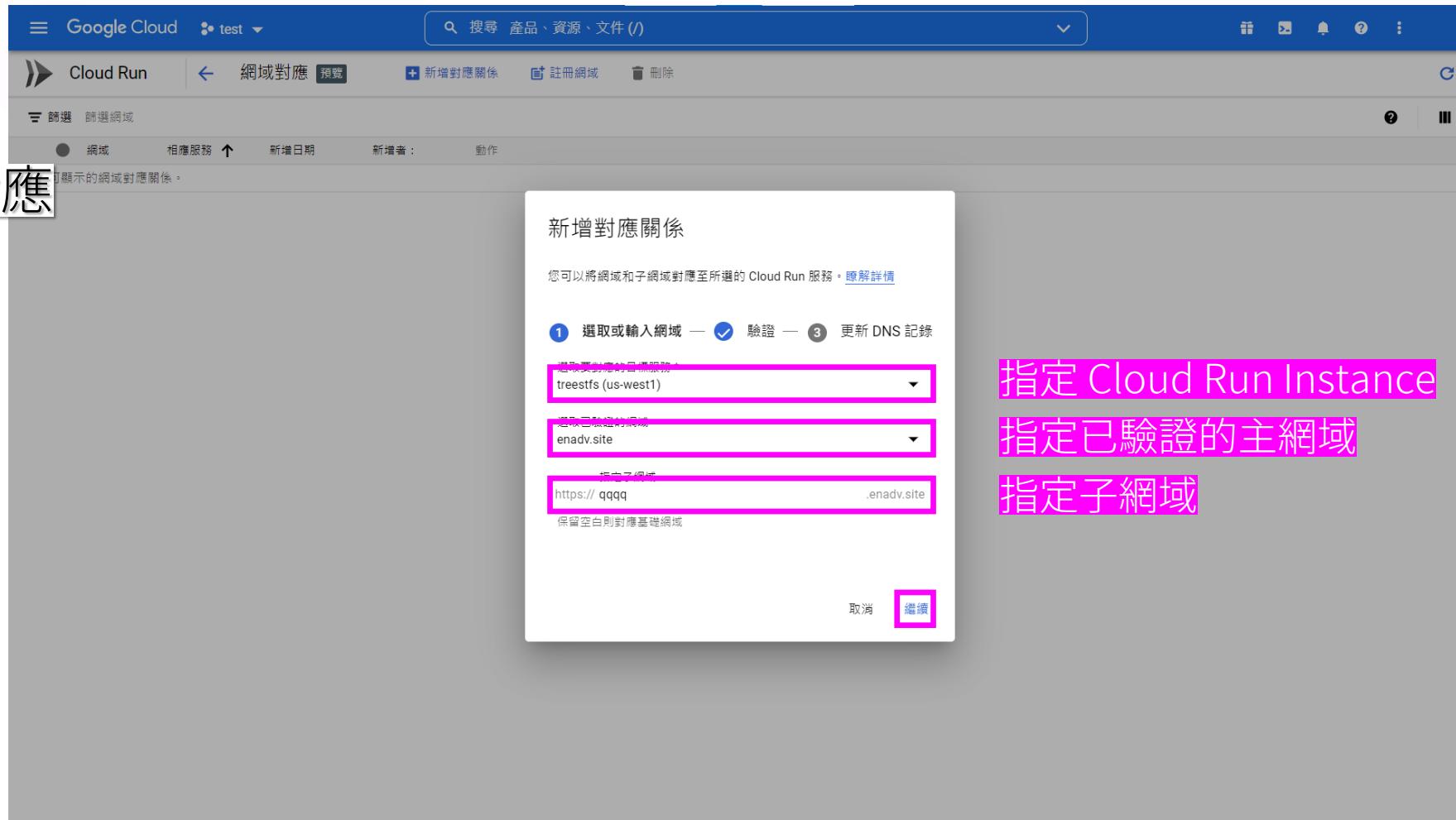
### b. 建立子網域對應



# 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. At the top, there's a navigation bar with links for '網域', '買賣', 'DNS', '設定', and '說明'. Below the navigation, it says '我的網域 / 網域設定' and 'DNS 管理'. Underneath that, it shows the domain 'enadv.site'. A modal window titled 'DNS 記錄' is open. It contains a sub-section titled 'CNAME 記錄' with the note: 'CNAME 記錄是一種子網域或別名類型，可以指向另一個網域名稱。'. The modal has buttons for '刪除' (Delete) and '複製' (Copy). On the right, there are buttons for '新增' (Add), '篩選' (Filter), and three dots. The main table area shows a single CNAME record with the following details:

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

Below the table, there are buttons for '新增記錄' (Add Record) and '清除' (Clear). To the right of the table, there's a link '貼上 CNAME 資料' (Paste CNAME Data).

# 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. Below the navigation is a secondary header with 'Cloud Run' and '網域對應' (Domain Mapping) tabs, along with buttons for '新增對應關係' (Add Mapping), '註冊網域' (Register Domain), and '刪除' (Delete). A '篩選' (Filter) button is also present. The main content area displays a table of domain mappings:

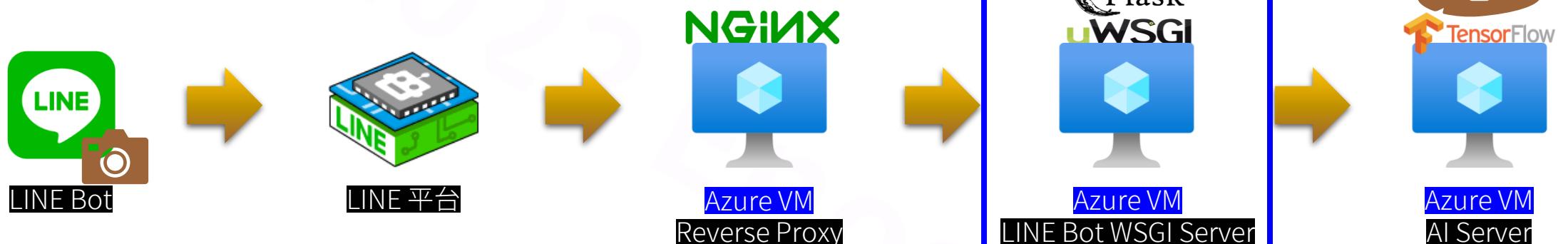
網域	相應服務	新增日期	新增者	動作
qqqq.enadv.site	treestfs (us-west1)	44 分鐘前	@gmail.com	⋮

# 部署 TensorFlow Serving - Cloud Run

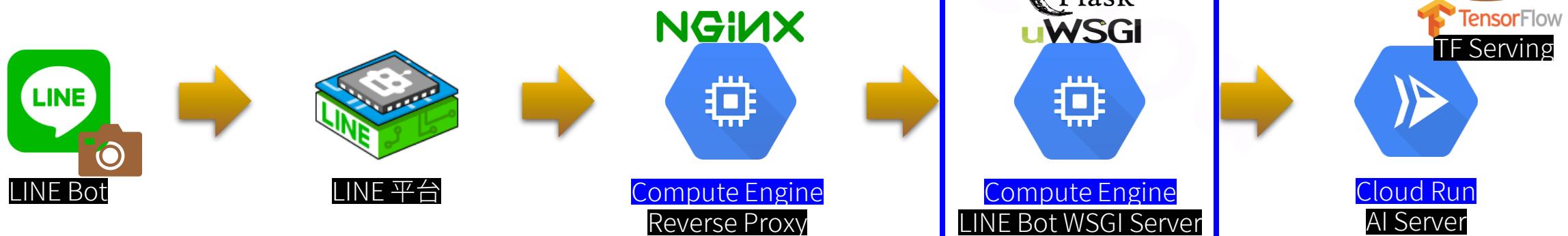
## 5. 測試 TensorFlow Serving

# 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\_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 Cloud VM 之專案目錄

`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 Cloud 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 cloud 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

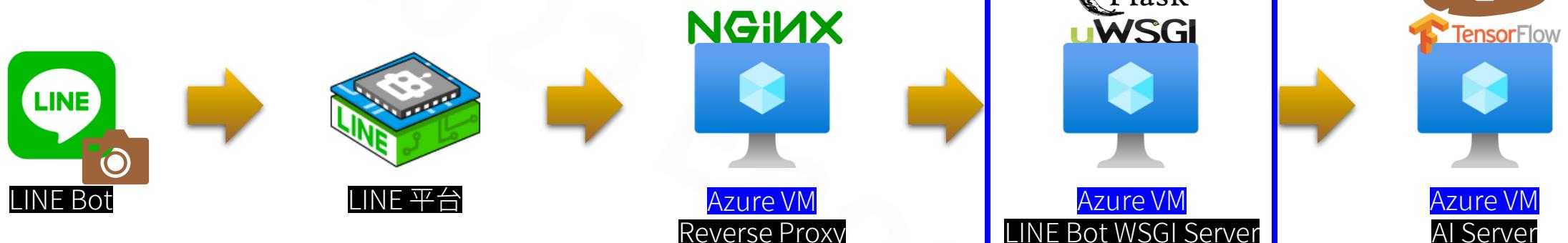
## 3. 調整 LINE Messaging

a. (調整 Webhook URL)

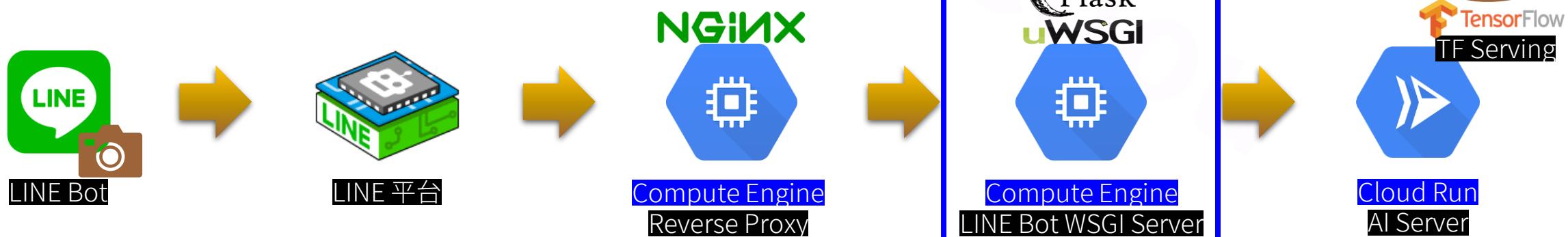
b. LINE 測試

# 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 Cloud 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 Cloud 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 cloud 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 ..; move p/tensorflow to project directory

#### Note

- ① 需要套件 `grpcio-tools==1.34.1`
- ② 建議在開發環境進行，產出上傳於部署環境
- ③ 開發環境的 `grpcio-tools` 與部署環境的 `grpcio` 版本需一致以避免問題
- ④ 開發環境與部署環境的 `protobuf` 版本需一致以避免問題

# 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

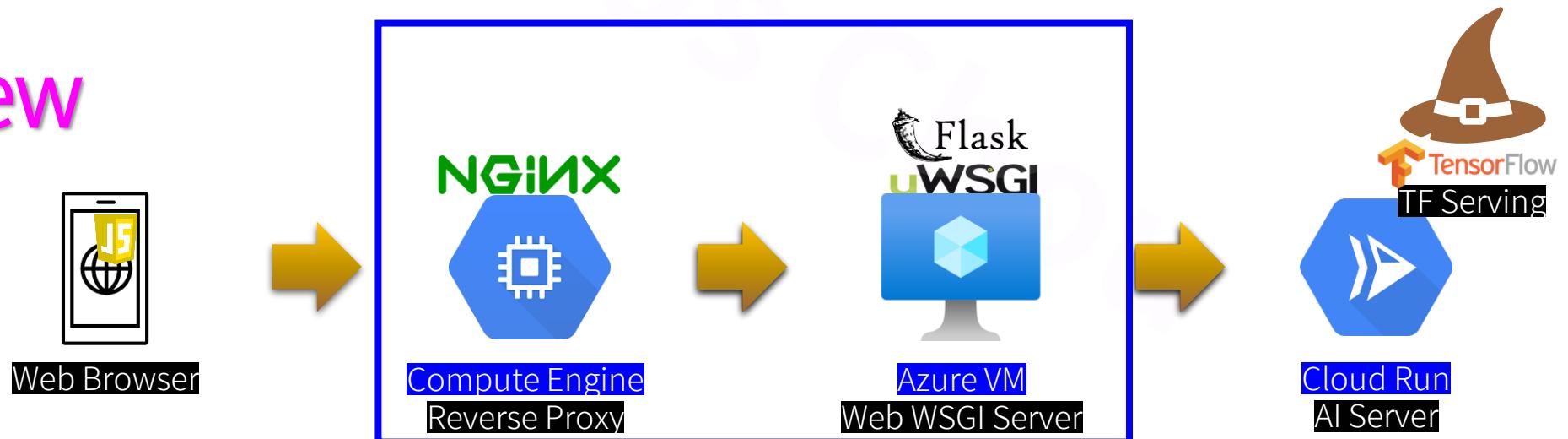
```
sudo rm -rf $HOME
```

重新啟動

# Web + AI 部署

# 部署 Web + TensorFlow Serving - gRPC

## Mixed View



# 部署 Web + TensorFlow Serving - gRPC

## 1. TensorFlow Serving gRPC 架設

略

# 部署 Web + TensorFlow Serving - gRPC

## 2. Web 開發

### a. 下載範例程式並調整

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

# 部署 Web + TensorFlow Serving - gRPC

## 2. Web 開發

### b. 製作 requirements.txt

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 Web + TensorFlow Serving - gRPC

## 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 Serverless 方式架設

# 部署 Web + TensorFlow Serving - gRPC

## 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
```

② 上傳下列檔案至 VM 之專案目錄

```
treesweb.py  
treeset_labels.txt  
env.json  
tensorflow/ # 經 protoc 編譯的 gRPC 碼
```

# 部署 Web + TensorFlow Serving - gRPC

## 3. 部署 Web + uWSGI

### c. 更新系統同時安裝 Web 與 WSGI Server

```
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 Web + TensorFlow Serving - gRPC

## 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 - gRPC

## 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 - gRPC

## 4. 部署 NGINX

### b. 於 VM 架設 NGINX 並申裝憑證

```
# .conf
server {
    server_name your_domain;

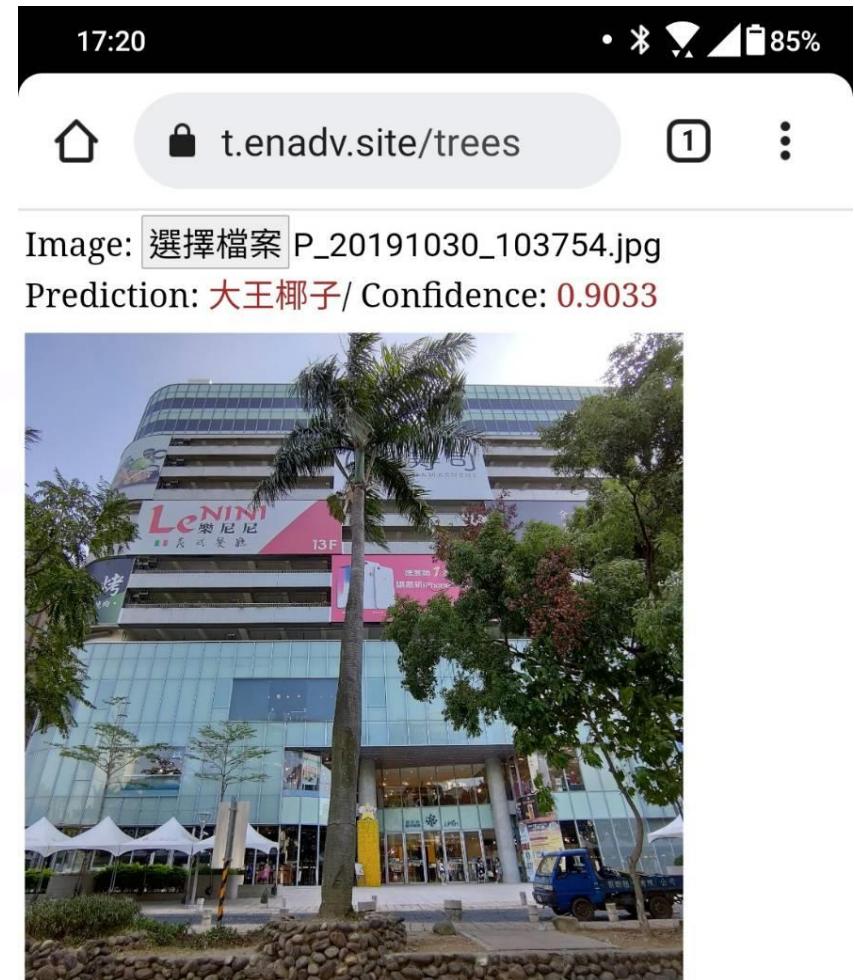
    location /your_path {
        include uwsgi_params;
        uwsgi_pass your_ip:your_port;
        client_max_body_size 12M;
    }
}
```

# 部署 Web + TensorFlow Serving - gRPC

## 5. Browser 測試

a. 以瀏覽器開啟網址

`https://your_domain/your_path`



# 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

```
# .conf
server {
    server_name your_domain;

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





# The End