台灣 IPv6 全面升級

IPv6 升級實作技術手冊

第二十四篇IPv6-Only Wi-Fi 平台建置

財團法人台灣網路資訊中心編撰

中 華 民 國 115 年 8 月 31 日

IPv6 升級實作技術手冊

出版者：財團法人台灣網路資訊中心
網址：https://www.twnic.tw/
地址：105 臺北市松山區八德路四段 123 號 3 樓
總機電話：886-2-25289696
發行人：黃勝雄
編審：余若凡
主編：顧靜恆
編輯：蔡更達、鄭進興、郭政哲、王彥傑、陳玟羽

中華民國 115 年 8 月第 1.0 版

IPv6 升級實作技術手冊

目錄

第二十四篇 IPv6-Only Wi-Fi 平台建置

一、 本篇概述

(一) 建置背景與發展趨勢

隨著全球網際網路持續成長，IPv4 位址資源早已面臨耗盡問題，雖然透過 NAT（Network Address Translation）技術延緩位址不足的衝擊，但亦衍生網路架構複雜化、管理成本增加及端對端通訊受限等問題。相較之下，IPv6 擁有近乎無限的位址空間，並具備自動組態、安全性擴充、路由效率提升及物聯網整合等優勢，已成為全球網路發展的重要方向。

近年來，全球主要網際網路服務供應商（ISP）、雲端服務業者及行動通訊業者均積極推動 IPv6 普及化。例如 Google、Apple、Meta、Microsoft 等國際大型網路服務已全面支援 IPv6；美國 T-Mobile、日本 NTT、印度 Reliance Jio 等業者更逐步導入 IPv6-Only 或 IPv6 Mostly 網路架構，以降低維運成本並提升網路效能。根據 APNIC 的統計資料顯示，全球 IPv6 使用率持續攀升，部分國家甚至已超過七成以上，顯示 IPv6 已由技術驗證階段逐步邁向全面部署階段。

在此趨勢下，傳統雙協定（Dual Stack）架構雖可同時支援 IPv4 與 IPv6，但仍需維護兩套網路系統，增加管理負擔與設備成本。因此，國際間逐漸開始研究與建置 IPv6-Only 網路環境，透過 NAT64、DNS64、464XLAT、PREF64 等技術，使終端設備即使在沒有 IPv4 位址的環境下，仍可順利存取既有 IPv4 網際網路服務，進一步達成簡化網路架構與提升管理效率之目標。

因此，建置 IPv6-Only Wi-Fi 平台不僅符合國際網路發展趨勢，更可作為未來校園、企業及政府機關推動 IPv6 網路轉型的重要驗證環境，具有高度研究與示範價值。

(二) IPv6-Only Wi-Fi 平台建置目標與內容

本平台建置之主要目標，在於建立一套符合國際標準且具備實際驗證能力的 IPv6-Only 無線網路環境，透過完整的網路架構與轉換機制，驗證各類終端設備於純 IPv6 網路中的運作情形，並評估其穩定性、可用性與相容性。

本平台將以 Ubuntu Linux Server 26.04 為核心建置基礎，整合開源軟體與相關網路服務，建構完整之 IPv6-Only 網路環境。平台主要功能包括 IPv6 Router Advertisement（RA）服務、DNS64 名稱解析服務、NAT64 位址轉換服務、DHCP 管理服務以及 Wi-Fi 無線存取環境等。

在位址配置方面，系統透過 Router Advertisement（RA）自動發送 IPv6 網路資訊，使終端設備能自動取得 IPv6 位址與預設閘道器設定。同時導入 RFC 8781 所定義之 PREF64 機制，使終端設備能夠自動獲知 NAT64 Prefix 資訊，進一步提升 IPv4 資源存取能力。

在名稱解析部分，平台採用 DNS64 技術，當使用者存取僅支援 IPv4 的網站時，DNS64 伺服器將自動產生對應之 IPv6 位址紀錄（AAAA Record），並交由 NAT64 系統進行封包轉換。透過此機制，即使終端設備完全不配置 IPv4 位址，仍可正常瀏覽傳統 IPv4 網站及服務。

在位址轉換方面，本平台採用 Jool NAT64 作為核心轉換系統，符合 RFC

6146 標準規範。當 IPv6 用戶端發送封包至 IPv4 服務時，NAT64 將負責進行 IPv6 與 IPv4 之間的協定轉換，使不同協定版本之設備能夠相互通訊，確保既有網際網路服務之可達性。

此外，平台亦將導入 RFC 8925 所規範之 IPv6-Only Preferred Option（DHCP Option 108），讓支援該標準之終端設備可優先選擇 IPv6 網路模式，進一步降低對 IPv4 的依賴程度，驗證未來 IPv6-Only 網路部署之可行性。

為符合實際應用需求，本平台亦將建置 Wi-Fi 無線網路環境，並整合 Captive Portal 身分驗證機制，提供使用者登入管理、設備識別與網路存取控制功能。透過完整之認證與管理流程，可模擬校園、企業及公共場域之實際部署情境。

最後，平台將針對 Windows 11、Android、iOS、Linux 等不同作業系統進行相容性測試，驗證其 IPv6 位址取得、自動組態、DNS64 解析、NAT64 轉換及應用服務存取能力。同時進行多使用者併發連線測試，預計支援至少 60 台終端設備同時接入，以評估系統效能與穩定性。

透過本 IPv6-Only Wi-Fi 平台之建置與驗證，不僅可深入了解 IPv6-Only 網路環境之運作模式，更可建立完整之建置流程、測試方法及管理經驗，作為未來校園網路、政府機關、企業網路及公共無線網路導入 IPv6-Only 架構的重要參考依據。隨著全球網路逐步朝向 IPv6 發展，本平台亦將成為推動數位基礎建設升級及促進下一世代網路技術發展的重要示範場域。

(三) IPv6-Only Wi-Fi 平台系統拓樸與核心主件

本 IPv6-Only Wi-Fi 平台 採用以 IPv6 為核心的網路架構，由 Ubuntu

Server 26.04 擔任網路閘道器（Gateway），整合 Netplan、Jool、Radvd 及 CoreDHCP 等 4 大核心組件，提供完整的 IPv6-Only 無線網路服務。系統透過無線基地台(ASUS Access Point)提供 Wi-Fi 連線，使用者連線後由 Radvd 發送 Router Advertisement（RA）及 PREF64 資訊，並由 CoreDHCP 提供 DHCP Option 108（IPv6-Only Preferred Option）等設定，使終端裝置取得 IPv6 網路參數。未完成認證的裝置會先由 nftables 將 HTTP 流量重新導向至 Python Flask 建置的 Captive Portal，完成認證後再將裝置加入授權清單，開放其網路存取權限。當使用者存取 IPv4 Only 網站時，BIND9 的 DNS64 會自動合成 AAAA 紀錄，再由 Jool 執行 NAT64 位址與封包轉換，使 IPv6-Only 用戶端仍可順利存取 IPv4 網際網路資源。整體系統採用模組化設計，各核心元件分工明確，兼具高相容性、易於維護、擴充性佳及完全採用開源軟體等優點，適合作為 IPv6-Only 網路環境之建置、測試與驗證平台。系統拓樸與核心組件協同架構如圖 1 所示。

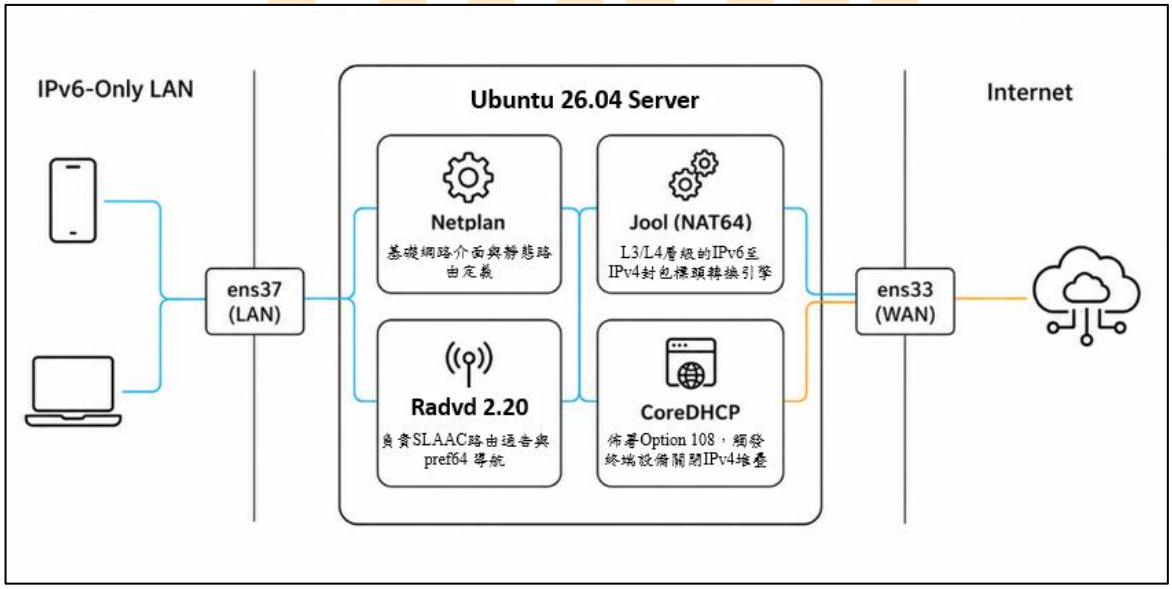


圖 1　系統拓樸與核心組件協同架構

(四) 系統實體網路介面與 IP 位址參數

本系統對照圖 1 的系統拓樸，系統實體網路介面與 IP 位址參數如下表所示，後續的範例系統安裝及設定均依照此表參數進行設定。

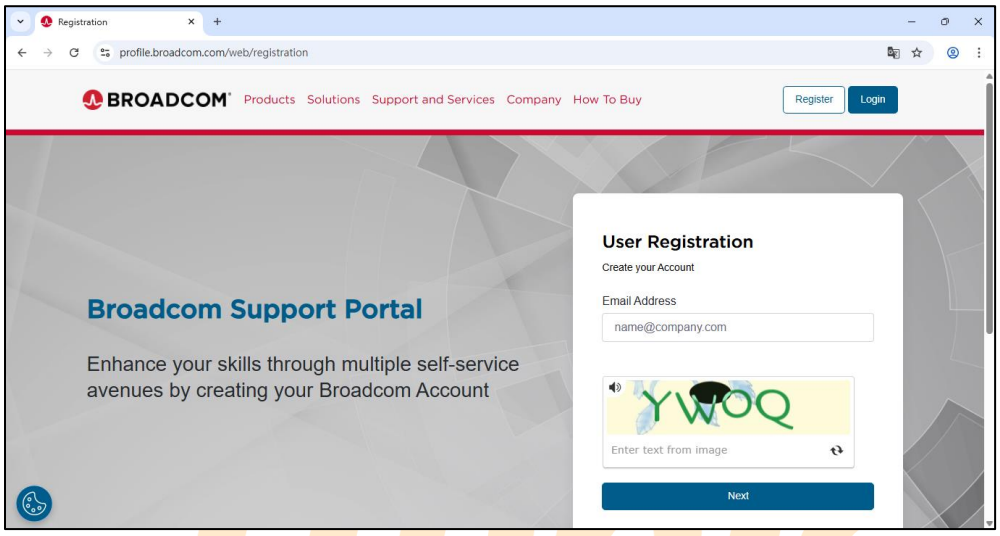
WAN 介面(ens33)	
IPv4 IP	163.18.22.69/24
IPv4 GW	163.18.22.254
IPv4 DNS	163.18.1.7
IPv6 IP	2001:288:8005:22::61/64
IPv6 GW	2001:288:8005:22::254
LAN 介面(ens37)	
IPv6 IP	2001:288:8005:ff00::1
IPv4 狀態	完全停用(IPv6-Only)
轉換前綴(NAT64 Prefix)	
Prefix	64:ff9b::/96 全球標準 NAT64 前綴

二、系統建構前置準備工作

(一) 下載及安裝 VMware Workstation Pro

1. 請到 Broadcom 的官方網站註冊一個帳號：

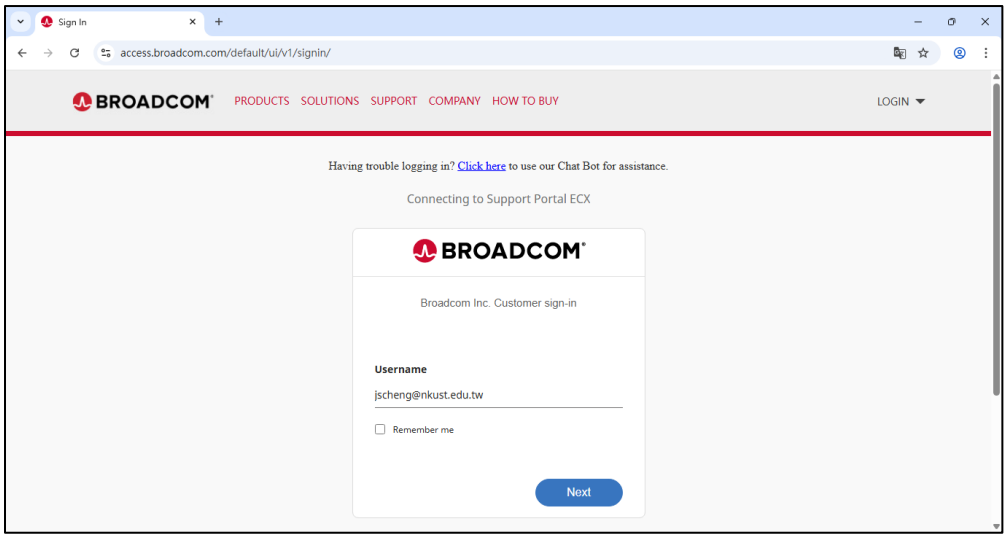
   https://profile.broadcom.com/web/registration



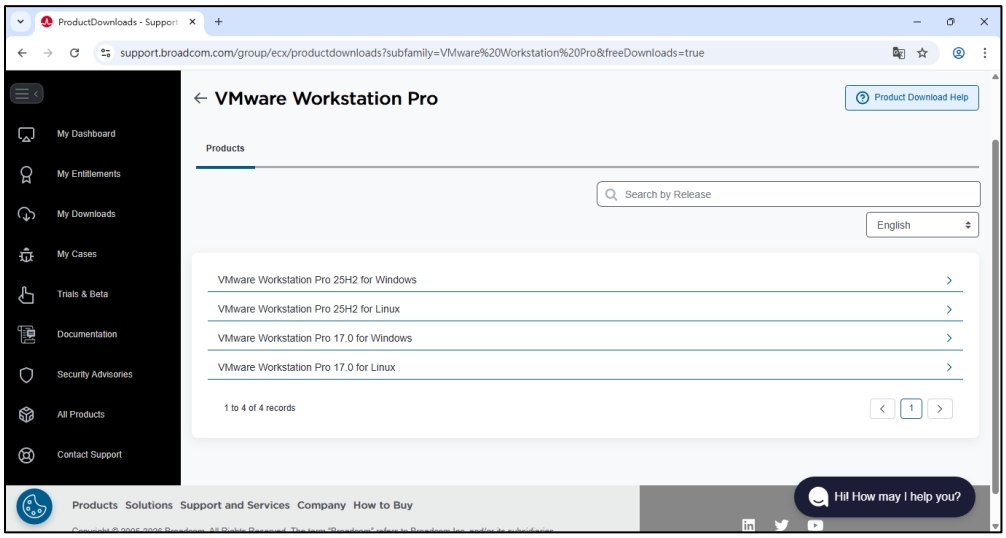
2. 登入 Broadcom

   https://access.broadcom.com/default/ui/v1/signin/

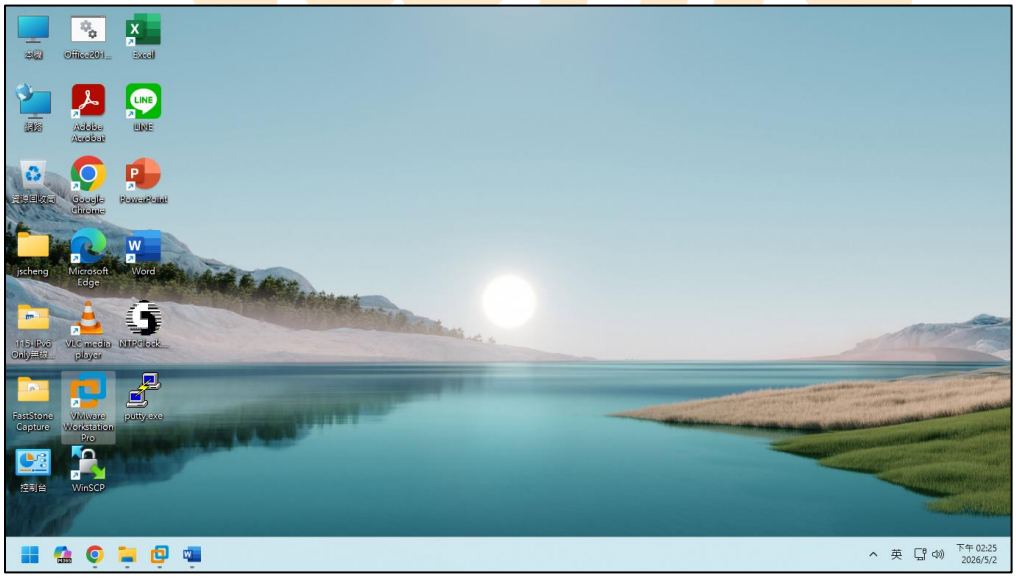
   輸入正確的帳號及密碼登入網站。



3. 下載 VMware Workstation Pro 25H2 for Windows 的安裝檔 VMware-Workstation-Full-25H2-24995812.exe。

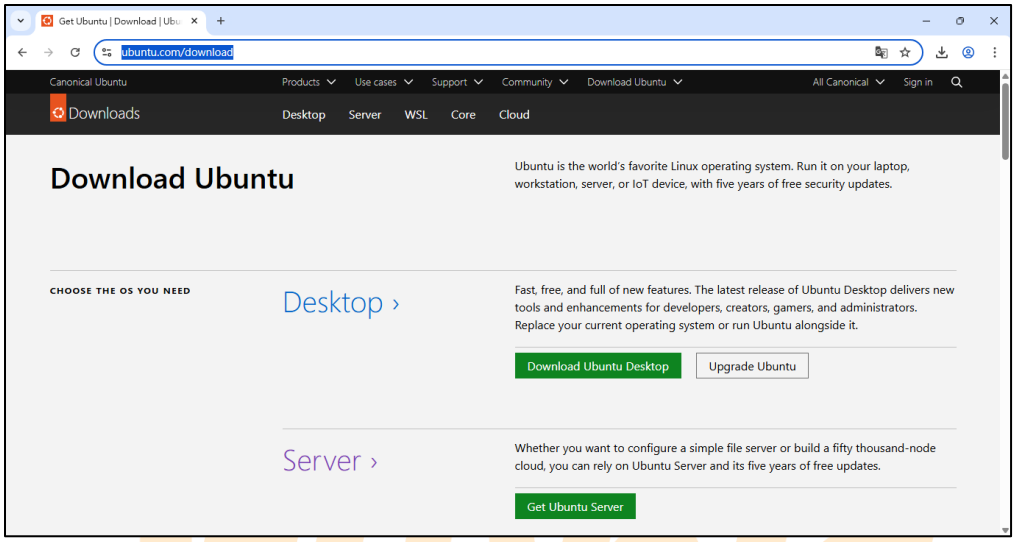


4. 進行安裝在 Windows 11 的實體主機上，安裝完成後，在桌面上會有 VMware Workstation Pro 的 icon。

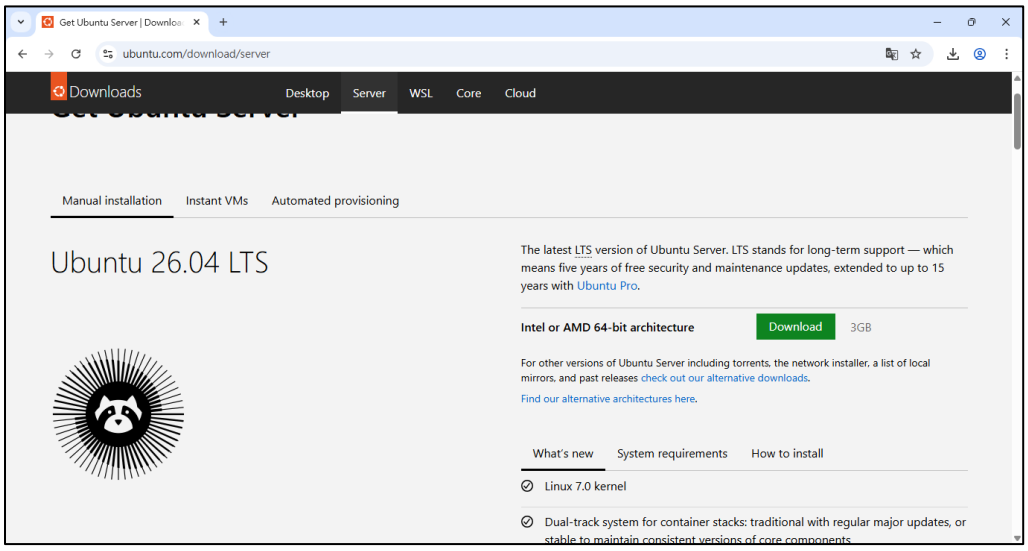


(二) 在 VMware Workstation Pro 前置安裝 Ubuntu Server 26.04 VM

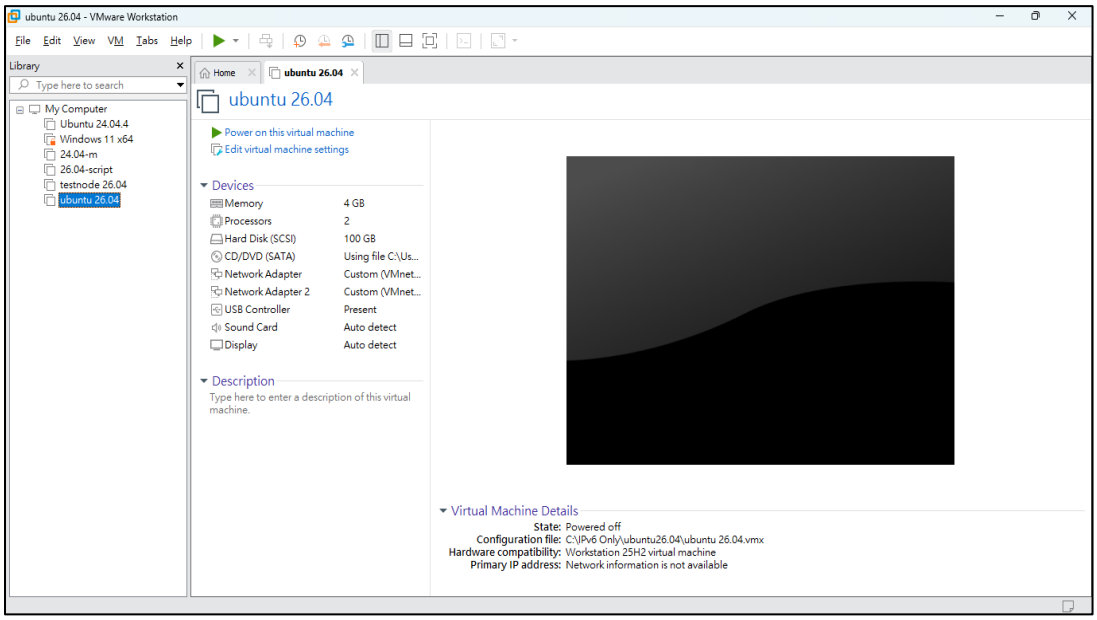
1. 請到 Ubuntu 官方網站 https://ubuntu.com/download 下載 Ubuntu Server 26.04。



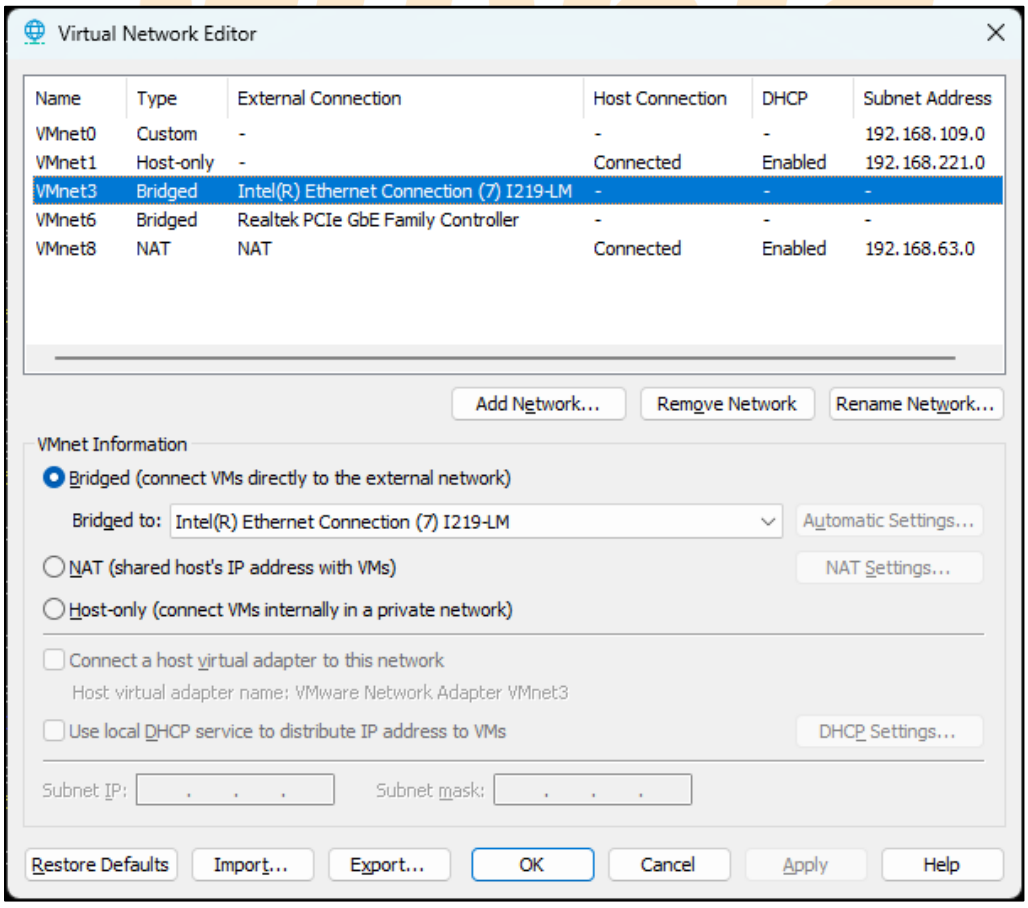
2. 下載 Ubuntu 26.04 LTS server 的安裝檔 ubuntu-26.04-live-server-amd64.iso



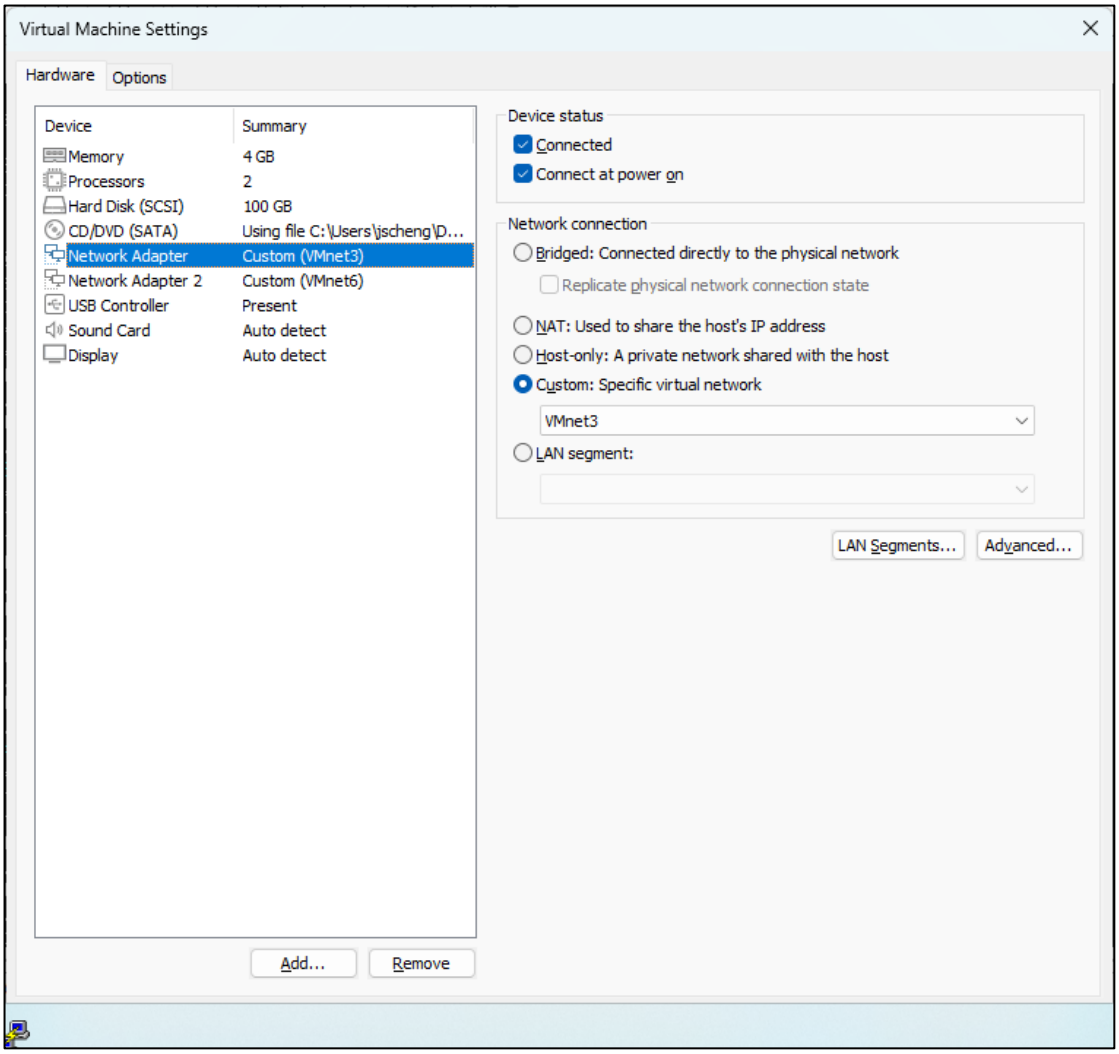
3. 在 VMware Workstation Pro 安裝好 Ubuntu Server 26.04 的 VM。



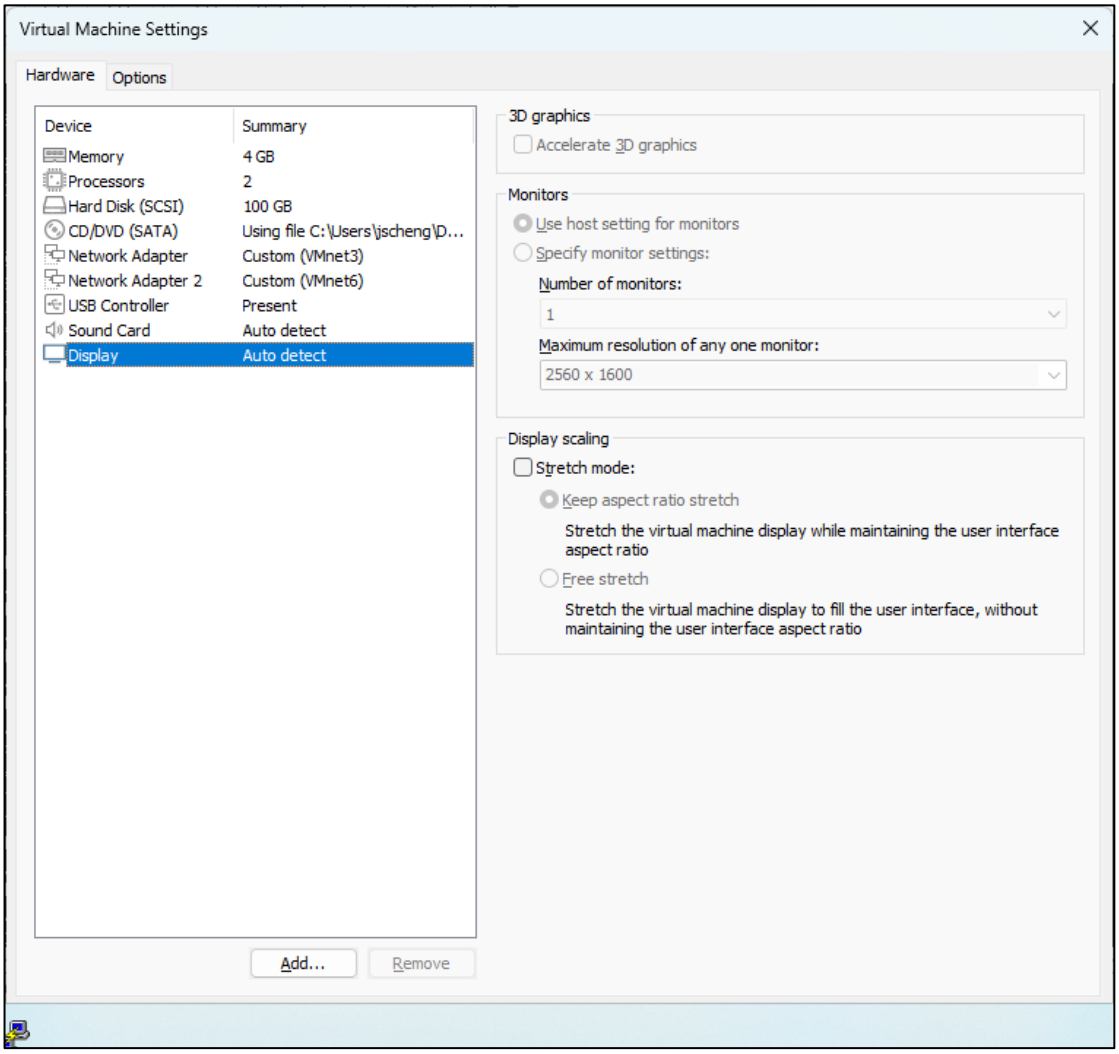
4. 在 VMware Workstation Pro 中的 Virtual network editor 設定中，VMnet3 及 VMnet6 的 Type 均設定為 Bridged。



5. 在 Ubuntu 26.04 這一個 VM 的設定中，有兩片網卡，分別連結到 VMnet3 及 VMnet6。



6. VM 的 Display 設定中，3D graphics 取消 Accelerate 3D graphics 的勾選。以避免開機時可能發生錯誤訊息。



以避免開機時可能發生以下錯誤訊息。

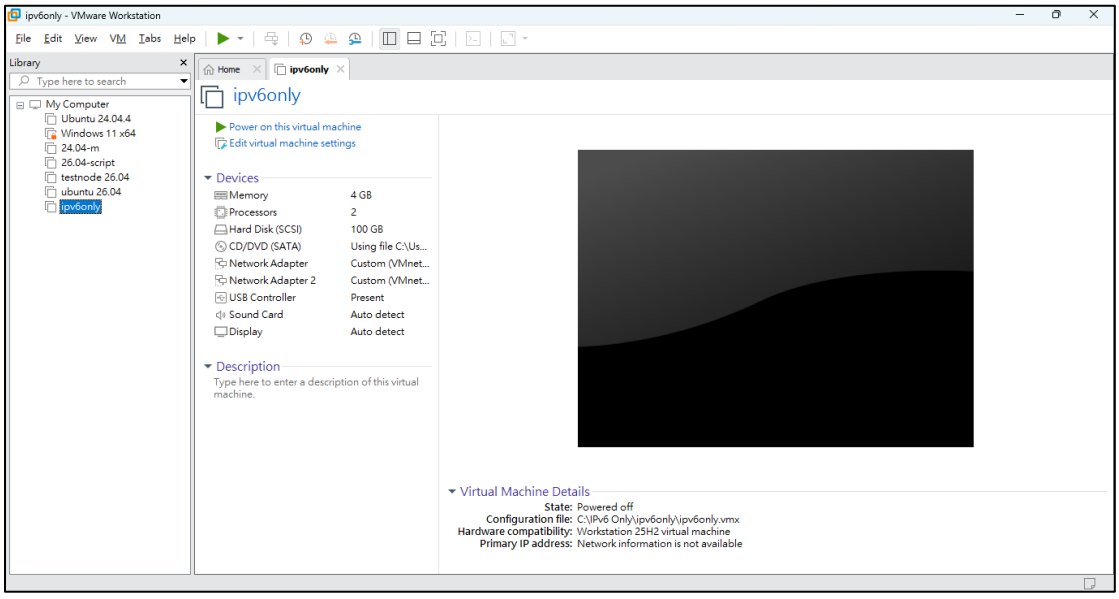
7. 本範例 Ubuntu Server 26.04 所在的 WAN 的 IPv4 網段的位址為 163.18.22.0/24，IPv6 網段位址為 2001:288:8005:22::/64，而 LAN 的 IPv6 網段位址為 2001:288:8005:ff00::/64，則 Ubuntu Server 26.04 的網路卡設定範例如下：

```
root@alpha:~# nano /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      addresses:
      - 163.18.22.69/24
      - 2001:288:8005:22::61/64
      nameservers:
```

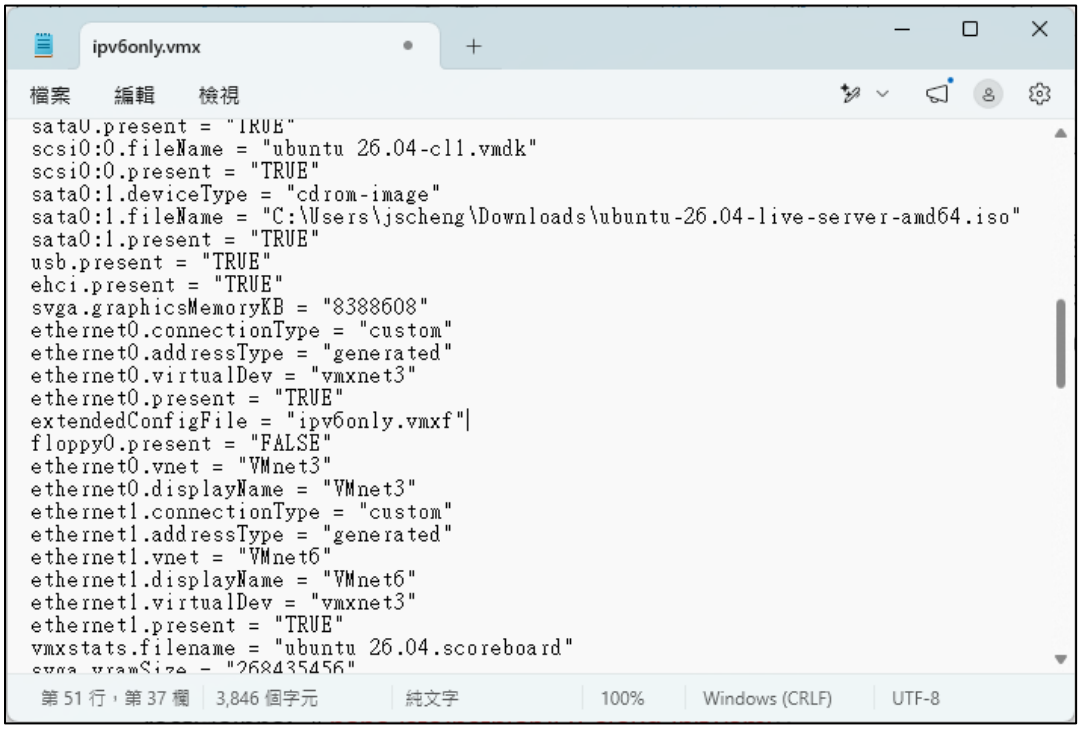
```
        addresses:
        - 163.18.1.7
      routes:
      - to: default
        via: 163.18.22.254
      - to: default
        via: 2001:288:8005:22::254
    ens34:
      addresses:
      - 2001:288:8005:ff00::1/64
  version: 2
```

8. 為了實作一台 IPv6-Only 主機，將 Ubuntu Server 26.04 關機，並複製一台 VM，其顯示名稱為 ipv6only。

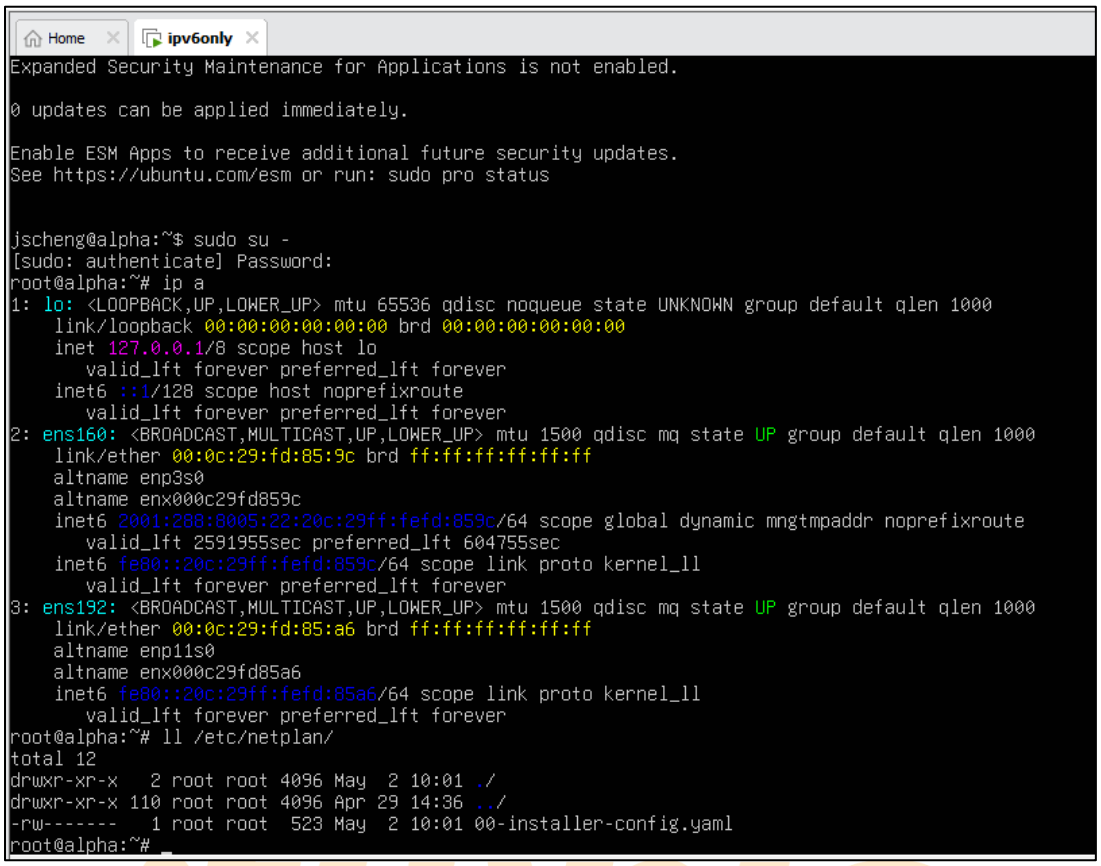
```
root@alpha:~# shutdown -h now
```



9. 在 VMware Workstation Pro 中將 VM 的網卡修改為高效能的 VMXNET3，可以讓 VM 的網路流量效能更好。而最直接的方法是修改虛擬機的設定檔 (.vmx)。請先關閉虛擬機並退出 VMware 程式，找到虛擬機資料夾內的 ipv6only.vmx 檔，用文字編輯器開啟，將 ethernet0.virtualDev = "e1000" (或 e1000e) 改為 ethernet0.virtualDev = "vmxnet3"，及將 ethernet1.virtualDev = "e1000" (或 e1000e) 改為 ethernet1.virtualDev = "vmxnet3"，儲存後重新開機即可。



10. 重新開機後會發現網卡代碼會自動變動為 ens160 及 ens192，網路卡設定檔的名稱也會改成 00-installer-config.yaml



11.重新設定網路卡的內容如下：

root@alpha:~# nano /etc/netplan/00-installer-config.yaml

This is the network config written by 'subiquity'

network:

ethernets:

ens160:

addresses:

- 163.18.22.69/24

- 2001:288:8005:22::61/64

nameservers:

```
        addresses:
        - 163.18.1.7
        search:
        - nkust.edu.tw
      routes:
      - to: default
        via: 163.18.22.254
      - to: default
        via: 2001:288:8005:22::254
    ens192:
      addresses:
      - 2001:288:8005:ff00::1/64
  version: 2
```

12.設定完成後，就可以直接再讓設定檔生效了。

```
root@alpha:~# netplan apply
```

13.設定時區

```
root@alpha:~# timedatectl
```

Local time: Sun 2026-05-03 03:11:07 UTC

Universal time: Sun 2026-05-03 03:11:07 UTC

RTC time: Sun 2026-05-03 03:11:07

Time zone: Etc/UTC (UTC, +0000)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

root@alpha:~# timedatectl set-timezone Asia/Taipei

root@alpha:~# timedatectl

Local time: Sun 2026-05-03 11:13:09 CST

Universal time: Sun 2026-05-03 03:13:09 UTC

RTC time: Sun 2026-05-03 03:13:10

Time zone: Asia/Taipei (CST, +0800)

System clock synchronized: yes

NTP service: active

RTC in local TZ: no

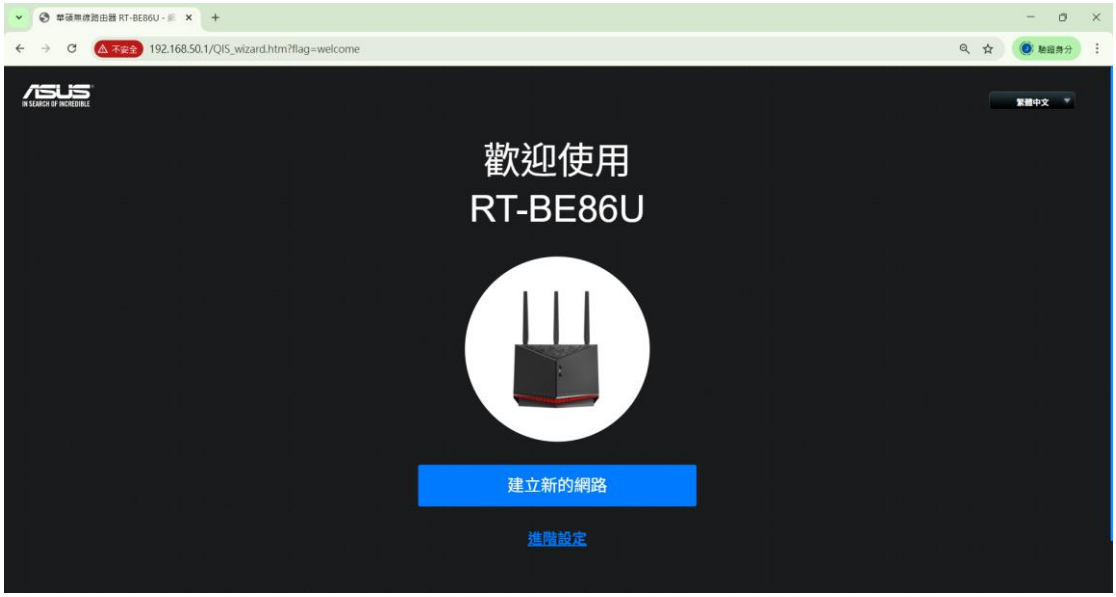
(三) ASUS(華碩) RT-BE86U BE6800 AP 設定

ASUS RT-BE86U BE6800 Wi-Fi 7 無線基地台（AP） 採用新一代 Wi-Fi 7

（IEEE 802.11be） 技術，提供雙頻最高 BE6800（2.4 GHz：1032 Mbps、5 GHz：5764 Mbps）的無線傳輸效能，支援 Multi-Link Operation (MLO)、4096-QAM 與 160 MHz 頻寬，可有效提升傳輸速度、降低延遲並增加多裝置同時連線的穩定性。設備內建 10GbE 網路埠及多個 2.5GbE 高速網路埠，可滿足高速光纖、伺服器及企業級網路環境需求；同時支援 IPv4/IPv6、AiMesh 網狀網路及 Access Point（AP）模式，可作為高速無線基地台部署於校園、實驗室或企業環境，提供穩定且高效能的 Wi-Fi 無線網路服務，特別適合建置 IPv6-Only 或 Wi-Fi 7 測試平台。

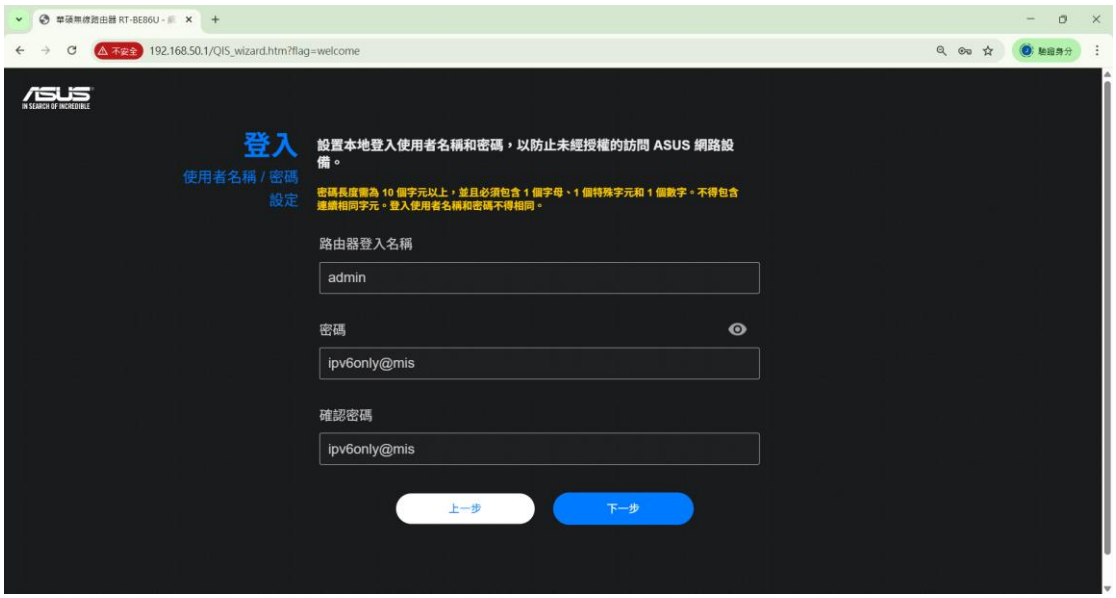
將 AP 接到 Ubuntu Server 26.04 實體機的 ens37 的 LAN Port，然後進行以下的設定步驟，讓 AP 可以正常的工作。

1. 為了進行 ASUS AP 的相關設定，首先打開 Windows 11 實體機的 Chrome 瀏覽器，ASUS AP 預設的管理介面連結為 http://192.168.50.1/， (最後修改成 http://192.168.1.1)。在登入畫面設定模式：按 「進階設定」。

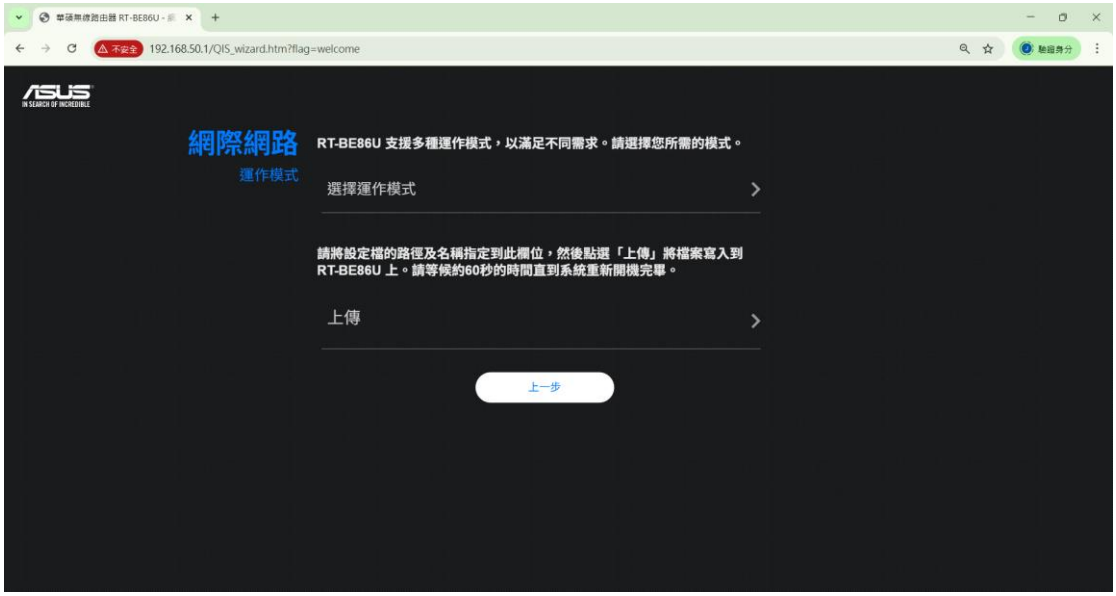


2. 設置本地登入使用者名稱和密碼：本案例設定路由器登入名稱：admin

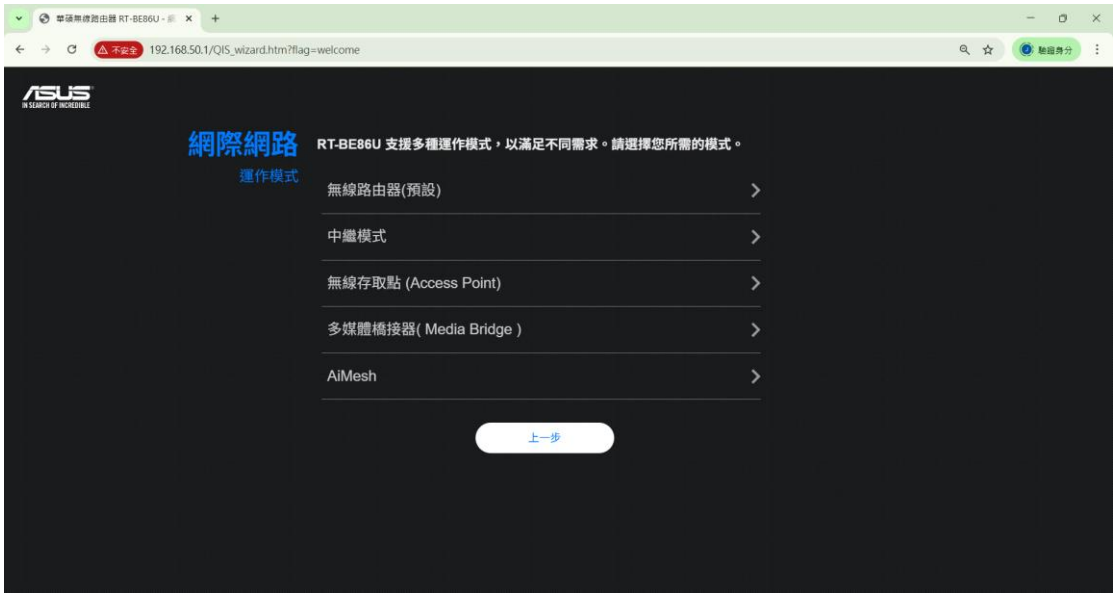
密碼：ipv6only@mis，再按：下一步。



3. 按「選擇運作模式」：



4. 按「無線存取點(Access Point)」



5. 按「手動指定」

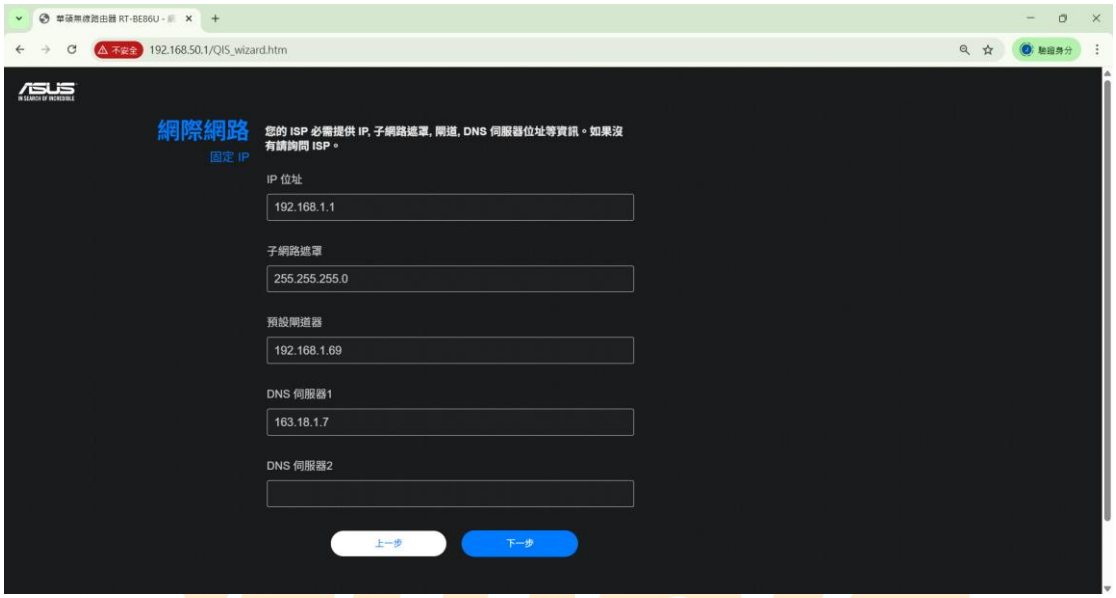
6. 設定相關參數如下，並按下一步：

IP 位址：192.168.1.1

子網路遮罩：255.255.255.0

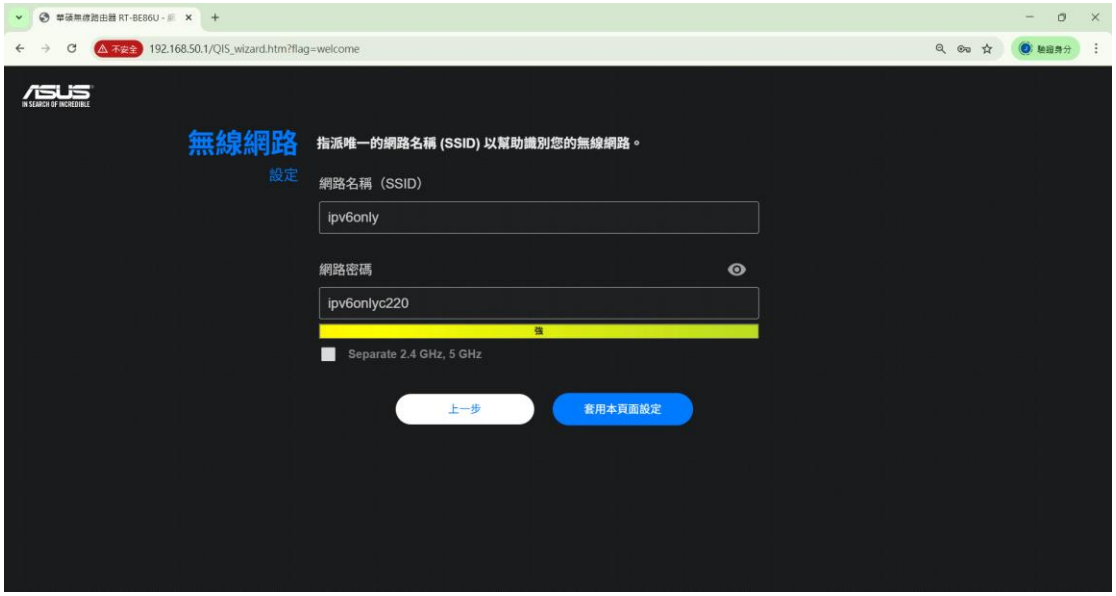
預設閘道器：192.168.1.69

DNS 伺服器 1：163.18.1.7



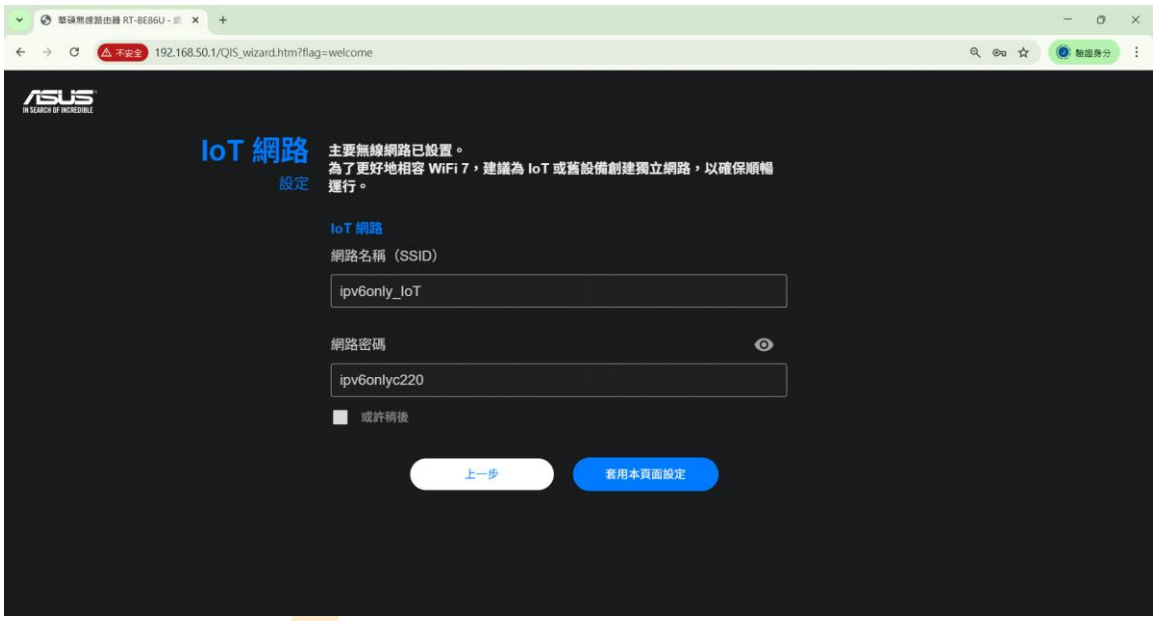
7. 指派唯一的網路名稱 (SSID) 以識別你的無線網路，本案例設定如下，記得要按：套用本頁面設定。本範例設定如下：

網路名稱（SSID）：ipv6only，網路密碼：ipv6onlyc220

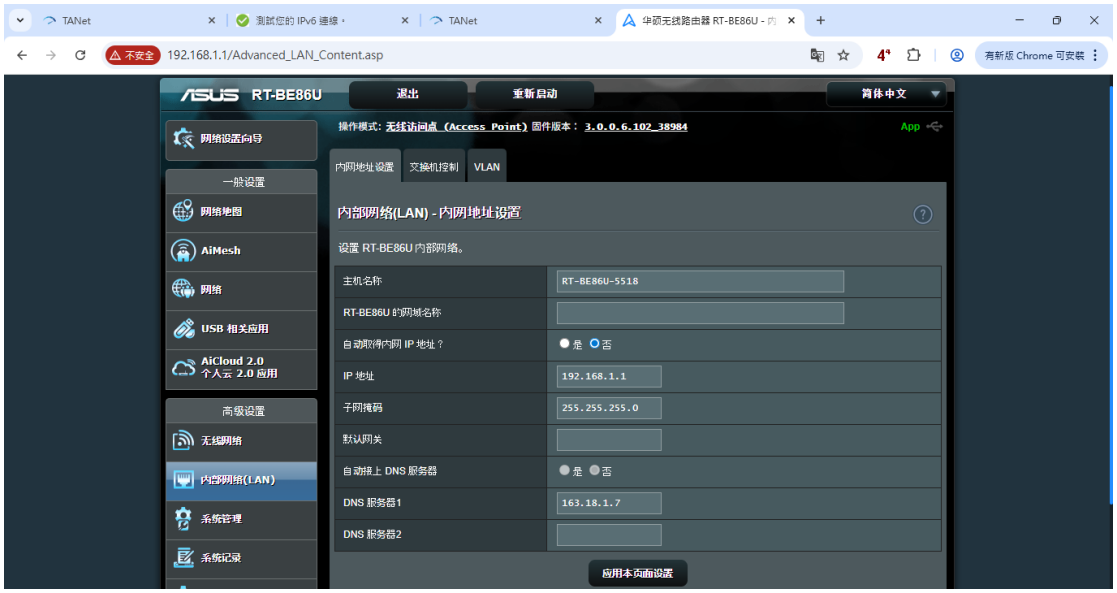


8. IoT 網路設定，本案例設定如下，記得要按：套用本頁面設定。

網路名稱（SSID）：ipv6only_IoT，網路密碼：ipv6onlyc220



9. 最後要取消 gateway(默認網關)設定，亦即留下空白設定，如下圖：



三、Jool 的安裝及設定

Jool 是一套可安裝於 Ubuntu Linux 的開源 NAT64（Network Address Translation 64）與 SIIT（Stateless IP/ICMP Translation）網路位址轉換軟體，主要功能是在 IPv6-Only 網路中，讓僅支援 IPv6 的用戶端仍可存取僅提供 IPv4 的網站與服務。Jool 以 Linux Kernel 模組運作，具備高效能、低延遲及高穩定性的特性，能與 DNS64 服務搭配，自動將 IPv4 位址轉換為 IPv6 位址，實現 IPv6 與 IPv4 網路間的透明互通。由於支援 RFC 6146（Stateful NAT64）及 RFC 7915（SIIT）等國際標準，Jool 已廣泛應用於 IPv6-Only、IPv6 Mostly、464XLAT、雲端平台及校園網路等環境，是建置 IPv6-Only 網路架構的重要核心元件之一。

有關 Jool 的安裝及設定過程分別敘述如下：

(一) Jool 的安裝

Jool 是一套 Linux 核心模組，用來實作 NAT64（IPv6 → IPv4）及 SIIT（Stateless IPv6/IPv4 Translation）。

1. 安裝 jool

```
root@alpha:~# apt update

root@alpha:~# apt upgrade -y

root@alpha:~# apt install -y net-tools

root@alpha:~# apt install -y jool-dkms jool-tools
```

(二) Jool 的設定

1. 開啟核心轉發 (Forwarding)

NAT64 需要同時開啟 IPv4 與 IPv6 的轉發功能：

root@alpha:~# echo "net.ipv4.conf.all.forwarding=1" >> /etc/sysctl.conf

root@alpha:~# echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf

root@alpha:~# echo "net.ipv4.ip_local_port_range=1024 20000" >> /etc/sysctl.conf

2. 確認設定是否正確

root@alpha:~# sysctl -p

net.ipv4.conf.all.forwarding = 1

net.ipv6.conf.all.forwarding = 1

net.ipv4.ip_local_port_range = 1024 20000

3. 啟動 Jool，載入模組並套用設定：

root@alpha:~# echo "jool" | sudo tee -a /etc/modules

jool

root@alpha:~# echo "jool" | sudo tee /etc/modules-load.d/jool.conf

jool

4. 自動載入 Jool 核心模組

編輯 /etc/modules-load.d/jool.conf，讓系統開機就載入 jool 模組。

root@alpha:~# nano /etc/systemd/system/jool.service

[Unit]

Description=Jool NAT64 Service

After=network-online.target

Wants=network-online.target

[Service]

Type=oneshot

RemainAfterExit=yes

啟動時讀取設定檔

ExecStart=/usr/bin/jool file handle /etc/jool/jool.conf

停止時移除實例

ExecStop=/usr/bin/jool instance remove default

[Install]

WantedBy=multi-user.target

5. 重新載入及設定系統開機時自動執行

root@alpha:~# systemctl daemon-reload

root@alpha:~# systemctl enable jool

Synchronizing state of jool.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.

Executing: /usr/lib/systemd/systemd-sysv-install enable jool

(三) 建立 Jool NAT64 設定

1. 建立 /etc/jool 目錄

root@alpha:~# mkdir /etc/jool

2. 設定 jool 設定檔

root@alpha:~# nano /etc/jool/jool.conf

```
{
    "instance": "default",
    "framework": "netfilter",
    "global": {
        "pool6": "64:ff9b::/96"
    },
    "pool4": [
        { "protocol": "TCP", "prefix": "163.18.22.69/32" },
        { "protocol": "UDP", "prefix": "163.18.22.69/32" },
```

```
        { "protocol": "ICMP", "prefix": "163.18.22.69/32" }
    ]
}
```

3. jool 設定檔說明

(1) 使用預設 Jool instance。

(2) 表示 Jool 使用 Linux Netfilter 架構處理封包。

(3) NAT64 使用的 IPv6 prefix。jool_pool6="64:ff9b::/96"

(4) 代表 NAT64 對外轉換時，會使用 WAN 的 IPv4 位址作為來源 IPv4。則 Jool 會用：163.18.22.69 作為 NAT64 出口 IPv4。

4. Jool 設定檔權限：root 可寫，其他使用者可讀。

```
root@alpha:~# chmod 644 /etc/jool/jool.conf
```

四、 Radvd 的安裝及設定

Radvd（Router Advertisement Daemon）是 Ubuntu/Linux 環境中常用的 IPv6 路由通告服務，其主要功能是依據 RFC 4861（Neighbor Discovery Protocol）定期或在用戶端請求時發送 Router Advertisement（RA） 封包，讓 IPv6 用戶端能自動取得網路設定，而無需手動配置。透過 Radvd，可向終端設備公告 IPv6 網路前綴（Prefix）、預設閘道（Default Gateway）、MTU、DNS 相關資訊（搭配 RFC 8106）以及 PREF64 前綴（RFC 8781） 等參數，使支援 IPv6 的作業系統能利用 SLAAC（Stateless Address Autoconfiguration）自動產生 IPv6 位址並建立連線。在 IPv6-Only 或 IPv6 Mostly 網路環境中，

Radvd 更是核心元件之一，可搭配 DNS64 與 NAT64（如 Jool） 建構完整的 IPv6-only 網路架構，讓僅支援 IPv4 的服務仍可透過位址轉換正常存取，同時支援最新的 IPv6 網路部署需求，因此廣泛應用於企業網路、校園網路、資料中心、實驗測試平台及 IPv6-Only Wi-Fi 環境建置。

(一) Radvd 概要說明

Radvd 的核心概念主要負責在區域網路（LAN）中，透過 IPv6 Neighbor Discovery Protocol（NDP） 發送 RA 封包，讓終端設備（例如，電腦、手機）自動完成以下設定：

(1) 自動取得 IPv6 位址（SLAAC）

(2) 取得預設閘道（Default Gateway）

(3) 獲得 DNS / 網路相關資訊（可搭配其他機制）

Radvd 的運作原理為當啟動 Radvd 時，它會在網路上定期發送 RA 訊息，內容包含：

(1) IPv6 網段（Prefix）

(2) 是否允許自動組態（A flag）

(3) 是否使用 DHCPv6（M flag / O flag）

(4) Router 的存在（Default Gateway）

終端設備收到後會自動設定，不需要手動輸入 IP。

Radvd 通常會搭配以下技術一起使用：

技術	功能
SLAAC	自動產生 IPv6 位址

DHCPv6	提供 DNS 等進階資訊
DNS64	將 IPv4 網站轉成 IPv6
NAT64	讓 IPv6-only 主機連 IPv4
PREF64（RFC 8781）	告訴 client NAT64 prefix

在 IPv6-Only 網路中，Radvd + PREF64 非常關鍵

此外 Radvd 的優點可以簡要條列如下：

(1) 不需要 DHCP 就能讓設備連結網路。

(2) 符合 IPv6 設計理念（Plug & Play）。

(3) 設定簡單、輕量。

(4) 非常適合 Wi-Fi / 校園 / IPv6-Only 環境。

Radvd 與 DHCPv6 重點比較比較如下表：

項目	**Radvd（RA）**	**DHCPv6**
IP 分配	自動（SLAAC）	伺服器分配
設定複雜度	低	中
控制能力	較低	較高
IPv6-Only 必備	✓	視需求

總結 Radvd 是 IPv6 網路中負責發送 Router Advertisement 的核心服務，讓終端設備能透過 SLAAC 自動取得 IPv6 位址與預設閘道，是建構 IPv6-Only 無線網路的關鍵基礎元件。

(二) Radvd 的安裝

Ubuntu 26.04 內建的 Radvd 2.20 版開始支援 PREF64 設定，我們是以 Ubuntu 26.04 來撰寫這個手冊。

1. 安裝 Radvd

root@alpha:~# apt install -y radvd

2. 重新載入及設定系統開機時自動執行

root@alpha:~# systemctl daemon-reload

root@alpha:~# systemctl enable radvd

Synchronizing state of radvd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.

Executing: /usr/lib/systemd/systemd-sysv-install enable radvd

(三) 建立 Radvd RA / PREF64 / RDNSS 設定

1. 設定 Radvd 設定檔

root@alpha:~# nano /etc/radvd.conf

```
interface ens192
{
        AdvSendAdvert on;
        MinRtrAdvInterval 30;
        MaxRtrAdvInterval 90;
        AdvCaptivePortalAPI "http://[2001:288:8005:ff00::1]:8080/";
```

```
        prefix 2001:288:8005:ff00::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };

        # NAT64 Prefix 公告
        nat64prefix 64:ff9b::/96 {};

        # DNS 設定
        #RDNSS 2001:288:8005:ff00::1
        RDNSS 2001:4860:4860::6464 2001:4860:4860::64
        {
        };
};
```

2. Radvd 設定檔說明

(1) AdvSendAdvert on; 這表示在 LAN 網卡上啟用 Router Advertisement。

(2) AdvSendAdvert on; LAN 端設備會透過 RA 自動取得 IPv6 網路資訊。

(3) AdvOnLink on; AdvAutonomous on; AdvRouterAddr on; 公告 LAN 端 IPv6 prefix。

(4) 例如：radvd_prefix="2001:288:8005:ff00::/64" 代表 LAN 端 client 可以使用 SLAAC 自動產生 IPv6 位址。

(5) AdvOnLink on 告訴 client 這個 prefix 是本地鏈路可達。

(6) AdvAutonomous on 允許 client 使用 SLAAC 自動產生 IPv6 位址。

(7) AdvRouterAddr on 公告 Router 位址。

(8) 它會透過 RA 公告 NAT64 Prefix，也就是 RFC 8781 的 PREF64 概念。radvd_pref64="64:ff9b::/96"

(9) Client 如果支援 RFC 8781，就可以知道 NAT64 prefix，進而合成 IPv4 對應的 IPv6 位址。

例如 IPv4：

93.184.216.34

可被轉成 NAT64 IPv6：

64:ff9b::5db8:d822

(10) 透過 RA 公告 DNS Server。

例如：radvd_dns64="2001:288:8005:ff00::1"，

代表 LAN 端 client 會使用這台 Gateway 當 DNS Server。

3. Radvd 設定檔權限：root 可寫，其他使用者可讀。

root@alpha:~# chmod 644 /etc/radvd.conf

五、CoreDHCP 的安裝及設定

CoreDHCP 是一套以 Go 語言開發的高效能、模組化 DHCP 伺服器，適合部署於 Ubuntu 等 Linux 作業系統。在 IPv6-Only 網路環境中，CoreDHCP 主要負責提供 DHCP 服務，向用戶端傳遞 IPv6 網路所需的設定資訊，例如 DNS 伺服器位址、網域搜尋清單及其他 DHCP 選項，並可支援 RFC 8925 所定義的 IPv6-Only Preferred Option（Option 108），通知相容的用戶端優先採用 IPv6-Only 網路模式，避免不必要的 IPv4 位址取得流程，加速網

路連線並降低雙協定管理成本。CoreDHCP 可與 Radvd 所提供的 Router Advertisement（RA）、BIND 的 DNS64 服務及 Jool 的 NAT64 功能整合，形成完整的 IPv6-Only 網路架構，使終端設備在沒有 IPv4 位址的情況下，仍能順利存取 IPv6 與 IPv4 網際網路資源，因此廣泛應用於 IPv6-Only Wi-Fi、校園網路、企業網路及 IPv6 技術驗證與研究平台。

(一) CoreDHCP 的安裝

首先安裝 Go 編譯環境，CoreDHCP 是由 Go 語言編寫的，Ubuntu Server 26.04 儲存庫中的版本已經足夠。

1. 安裝 Go 編譯環境

```
root@alpha:~# apt update
root@alpha:~# apt install -y golang git
```

2. 下載 CoreDHCP 及安裝

我們直接從 GitHub 官方網站下載獲得源碼檔案，並進行編譯，以確保包含最新的 IPv6-Only 套件。

```
root@alpha:~# cd /usr/local/src
root@alpha:/usr/local/src# git clone https://github.com/coredhcp/coredhcp.git
root@alpha:/usr/local/src# cd coredhcp/cmds/coredhcp
root@alpha:/usr/local/src/coredhcp/cmds/coredhcp# go build -o coredhcp
```

3. 目錄移動

為了方便之後透過 systemd 呼叫，我們將它移動到 /usr/local/sbin：

root@alpha:/usr/local/src/coredhcp/cmds/coredhcp# cp coredhcp /usr/local/sbin/

(二) 設定 CoreDHCP

1. CoreDHCP 的設定

root@alpha:/usr/local/src/coredhcp/cmds/coredhcp# nano /etc/coredhcp.yml

```
# coredhcp 設定檔
server4:
  #listen: ["0.0.0.0"]
  listen: ["%ens192"]
  plugins:
    # 直接使用 ipv6only 插件，它會自動處理 Option 108，需要注意順序
    - server_id: "163.18.22.69" # 這個一定要加，不然 iPhone 會一直發出 Request
    - lease_time: "3600s"
    - ipv6only: "24h"
```

2. /etc/coredhcp.yml 設定檔整體功能摘要說明

此設定檔的目的，是讓 CoreDHCP 在 ens192 網路介面上提供 DHCPv4 服務，但不分配 IPv4 位址，而是利用 RFC 8925 的 IPv6-Only Preferred（Option 108） 通知支援的終端設備優先切換至 IPv6-Only 模式。設定中的 server_id 可避免 iPhone 等裝置因缺少 DHCP Server Identifier 而持續重送

DHCP Request；lease_time 用於控制 DHCP 租約更新週期；ipv6only: "24h" 則指定終端在接下來 24 小時內優先採用 IPv6 通訊。此設定通常會搭配 radvd、DNS64 及 Jool NAT64 使用，共同建構完整的 IPv6-Only 網路環境，使終端設備即使沒有 IPv4 位址，仍能正常存取 IPv4 與 IPv6 網際網路資源。

3. 設定 CoreDHCP 以系統服務（daemon）方式自動啟動與維運

root@alpha:/usr/local/src/coredhcp/cmds/coredhcp# cd ~

root@alpha:~# nano /etc/systemd/system/coredhcp.service

[Unit]

Description=CoreDHCP IPv6-Only Preferred Service

After=network.target

[Service]

ExecStart=/usr/local/sbin/coredhcp --conf /etc/coredhcp.yml -L debug

Restart=always

StandardOutput=append:/var/log/coredhcp.log

StandardError=append:/var/log/coredhcp.log

[Install]

WantedBy=multi-user.target

4. /etc/systemd/system/coredhcp.service 設定檔整體功能摘要說明

[Unit] 區塊： 表示這個服務是用來支援 IPv6-Only（RFC 8925）環境，表示等網路初始化完成後才啟動
[Service] 區塊：啟動 CoreDHCP。詳細記錄 DHCP 封包，顯示 client 行為。服務掛掉會自動重啟。將：正常輸出（stdout），錯誤輸出（stderr），全部寫入/var/log/coredhcp.log。
[Install] 區塊：表示在系統進入「多使用者模式」時啟動，等同於 開機 → Linux 正常運作 → 自動啟動 CoreDHCP。

5. 重新載入及設定系統開機時自動執行

root@alpha:~# systemctl daemon-reload

root@alpha:~# systemctl enable coredhcp

Created symlink '/etc/systemd/system/multi-user.target.wants/coredhcp.service' → '/etc/systemd/system/coredhcp.service'.

六、 /etc/rc.local 及/etc/systemd/system/rc-local.service 的設定

這份 /etc/rc.local 是一個「開機初始化腳本」，主要用於系統開機後自動初始化 IPv6-Only / NAT64 平台，包含載入系統網路參數、設定 Jool NAT64 的 IPv4 位址與連接埠池、啟動 IPv6 Captive Portal、建立 UDP 轉送通道，以

及透過 nftables 監控使用 NAT64 前綴 64:ff9b::/96 的 IPv6 目的位址，方便後續觀察哪些 IPv4 服務被用戶端透過 NAT64 存取。

/etc/systemd/system/rc-local.service 此設定檔是 systemd 的服務單元(Service Unit)，其主要目的在於讓 Ubuntu 系統重新支援傳統的 /etc/rc.local 開機啟動機制。由於 Ubuntu 20.04 之後（包含 Ubuntu 22.04、24.04、26.04）已不再預設啟用 rc.local，因此需自行建立 rc-local.service，讓 systemd 在系統開機完成並完成網路初始化後，自動執行 /etc/rc.local 腳本。如此即可於每次系統啟動時，自動完成 IPv6-Only 平台所需的初始化工作，例如載入 Linux 核心參數（sysctl）、設定 Jool NAT64 Pool4、啟動 IPv6 Captive Portal、建立 WireGuard UDP Relay，以及設定 nftables 封包監控規則，免除管理者每次開機後需手動執行相關指令，大幅提升系統自動化與管理便利性。

(一) /etc/rc.local 設定內容

1. 設定/etc/rc.local

root@alpha:~# nano /etc/rc.local

```
#!/bin/bash


#sudo ip link set dev ens160 mtu 1280

#sudo ip link set dev ens192 mtu 1280


sysctl -p
```

```
sleep 10


jool pool4 flush

jool pool4 add 163.18.22.69 20001-65535 --tcp

jool pool4 add 163.18.22.69 20001-65535 --udp

jool pool4 add 163.18.22.69 20001-65535 --icmp


#nft -f /etc/nftables.conf

nohup sudo python3 /root/ipv6_captive_portal.py >>
/var/log/ipv6_captive_portal.txt 2>&1 &


# https://www.route64.org/ wireguard

nohup socat UDP4-LISTEN:2000,fork,reuseaddr UDP4:103.172.116.132:50522
&


########################################


# 1. 清除舊規則 (NAT64 IPv6 版本)

nft flush chain ip6 jool_monitor_v6_nat64 prerouting 2>/dev/null
```

2. 建立 IPv6 表格

nft add table ip6 jool_monitor_v6_nat64

3. 建立 IPv6 Set (型別改為 ipv6_addr)

nft 'add set ip6 jool_monitor_v6_nat64 active_v6_dsts { type ipv6_addr; flags dynamic; timeout 300s; }'

4. 建立鏈 (改用 prerouting，在進入 Jool 轉換前攔截)

優先級設為 -100，確保在 Jool (通常是 -90 或 0) 處理前就抓到

nft 'add chain ip6 jool_monitor_v6_nat64 prerouting { type filter hook prerouting priority -100; }'

5. 加入規則：

這裡我們監控目的地為 NAT64 Prefix (64:ff9b::/96) 的封包

nft 'add rule ip6 jool_monitor_v6_nat64 prerouting ip6 daddr 64:ff9b::/96 counter add @active_v6_dsts { ip6 daddr }'

A. 查看抓到的目的地 IPv6 (這會看到 64:ff9b::/96 ... 開頭的位址)

##nft list set ip6 jool_monitor_v6_nat64 active_v6_dsts

##

exit 0

2. 設定/etc/rc.local 為可執行檔案

root@alpha:~# chmod +x /etc/rc.local

(二) /etc/rc.local 設定內容指令簡要說明

設定指令簡要說明如下表所示：

設定指令	說明
#!/bin/bash	指定此檔案以 Bash shell 執行。
#sudo ip link set dev ens160 mtu 1280	註解狀態；原用途是將 ens160 介面的 MTU 設為 1280，這是 IPv6 規定的最小 MTU。
#sudo ip link set dev ens192 mtu 1280	註解狀態；原用途是將 ens192 介面的 MTU 設為 1280，可避免部分 IPv6/NAT64 封包因 MTU 過大造成傳輸問題。
sysctl -p	重新載入 /etc/sysctl.conf 的核心參數，例如 IPv6 forwarding、路由轉送

	等設定。
sleep 10	開機後等待 10 秒，讓網路介面與相關服務先完成初始化。
jool pool4 flush	清除 Jool 既有的 IPv4 位址池設定，避免重複或殘留規則影響 NAT64。
jool pool4 add 163.18.22.69 20001-65535 --tcp	指定 Jool 使用 IPv4 位址 163.18.22.69 與 TCP 連接埠 20001–65535 作為 NAT64 轉換池。
jool pool4 add 163.18.22.69 20001-65535 --udp	指定 UDP NAT64 轉換使用相同 IPv4 位址與連接埠範圍。
jool pool4 add 163.18.22.69 20001-65535 --icmp	指定 ICMP NAT64 轉換使用該 IPv4 位址與可用識別範圍。
#nft -f /etc/nftables.conf	註解狀態；原用途是載入 nftables 防火牆規則。
nohup sudo python3 /root/ipv6_captive_portal.py >> /var/log/ipv6_captive_portal.txt 2>&1 &	背景啟動 IPv6 Captive Portal 程式，並將輸出與錯誤訊息寫入 /var/log/ipv6_captive_portal.txt。
nohup socat UDP4-LISTEN:2000,fork,reuseaddr UDP4:103.172.116.132:50522 &	使用 socat 建立 UDP 轉送，將本機 UDP 2000 port 收到的封包轉送到 103.172.116.132:50522，可用於 WireGuard 或外部 UDP 服務轉接。
nft flush chain ip6 jool_monitor_v6_nat64 prerouting 2>/dev/null	嘗試清除既有 nftables chain 中的舊規則；若不存在則將錯誤訊息丟棄。
nft add table ip6	建立 IPv6 nftables 表格，用於

jool_monitor_v6_nat64	NAT64 流量監控。
nft 'add set ip6 jool_monitor_v6_nat64 active_v6_dsts { type ipv6_addr; flags dynamic; timeout 300s; }'	建立動態 IPv6 位址集合，記錄被存取的 NAT64 目的 IPv6 位址，資料保留 300 秒。
nft 'add chain ip6 jool_monitor_v6_nat64 prerouting { type filter hook prerouting priority -100; }'	建立 prerouting chain，在封包進入 Jool 轉換前先攔截與記錄。
nft 'add rule ip6 jool_monitor_v6_nat64 prerouting ip6 daddr 64:ff9b::/96 counter add @active_v6_dsts { ip6 daddr }'	監控目的位址屬於 NAT64 前綴 64:ff9b::/96 的封包，計數並記錄目的 IPv6 位址。
##nft list set ip6 jool_monitor_v6_nat64 active_v6_dsts	註解狀態；可用來查看目前被記錄的 NAT64 目的位址。
exit 0	正常結束 rc.local 腳本。

(三) /etc/systemd/system/rc-local.service 設定內容

1. 設定/etc/systemd/system/rc-local.service

root@alpha:~# nano /etc/systemd/system/rc-local.service

[Unit]

Description=/etc/rc.local Compatibility

ConditionFileIsExecutable=/etc/rc.local

After=network.target

[Service]

Type=forking

ExecStart=/etc/rc.local start

TimeoutSec=0

RemainAfterExit=yes

GuessMainPID=no

[Install]

WantedBy=multi-user.target

2. 重新載入及設定系統開機時自動執行

root@alpha:~# systemctl daemon-reload

root@alpha:~# systemctl enable rc-local

Created symlink '/etc/systemd/system/multi-user.target.wants/rc-local.service' → '/etc/systemd/system/rc-local.service'.

七、設定 WAN / LAN 網卡 MTU 及備份並重建 Netplan 網路設定

(一) 設定 WAN / LAN 網卡 MTU

1280 是 IPv6 規定的最小 MTU，因此常用於 IPv6-Only、NAT64、隧道或實驗環境，避免封包過大造成傳輸問題。

```
root@alpha:~# ip link set dev ens160 mtu 1280
root@alpha:~# ip link set dev ens192 mtu 1280
```

(二) 備份並重建 Netplan 網路設定

```
root@alpha:~# mkdir -p /etc/netplan/backup01
root@alpha:~# mv /etc/netplan/*.yaml /etc/netplan/backup01/
root@alpha:~# nano /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    ens160:
      dhcp4: no
      dhcp6: no
      addresses:
        - 163.18.22.69/24
        - 2001:288:8005:22::61/64
      routes:
```

```
        - to: default
          via: 163.18.22.254
        - to: default
          via: 2001:288:8005:22::254
      nameservers:
        addresses:
          - 163.18.1.7
    ens192:
      dhcp4: no
      dhcp6: no
      addresses:
        - 2001:288:8005:ff00::1/64
```

```
root@alpha:~# chmod 600 /etc/netplan/01-netcfg.yaml

root@alpha:~# netplan apply
```

八、 啟用 named DNS 服務

BIND9（Berkeley Internet Name Domain 9） 是目前最廣泛使用的開源 DNS（Domain Name System）伺服器軟體，可提供網域名稱解析、權威 DNS（Authoritative DNS）、遞迴 DNS（Recursive DNS）及 DNS 快取等功能。在 IPv6-Only 網路環境中，BIND9 可搭配 DNS64 功能，將 IPv4 Only 網站的 A 紀錄自動轉換為 AAAA 紀錄，使 IPv6-Only 用戶端能透過 NAT64 機制存取 IPv4 網路資源，提供完整的網際網路存取能力。

Bind9-utils 是 BIND9 的管理與診斷工具套件，提供多種 DNS 維護與測試指令，例如 dig、nslookup、named-checkconf 及 named-checkzone 等，可用於查詢 DNS 解析結果、驗證 DNS 設定檔與檢查 Zone 檔案是否正確。在建置 IPv6-Only 與 DNS64 環境時，管理者可利用這些工具確認 DNS 服務是否正常運作，並驗證 DNS64 是否成功產生 AAAA 紀錄，以提升系統建置與維護效率。

(一) Bind9 及 Bind9-utils 的安裝及設定

1. Bind9 及 Bind9-utils 的安裝

root@alpha:~# apt install -y bind9 bind9-dnsutils

2. Bind9 的設定

root@alpha:~# nano /etc/bind/named.conf.options

```
options {
        directory "/var/cache/bind";

        // 1. 安全性設定：只允許你的內網查詢 (避免變成公開 DNS 被攻擊)
        // 請根據你的實際網段修改，例如 localhost 和你的 IPv6/IPv4 網段
            //allow-query { localhost; 192.168.0.0/16; 2001:470:fe10::/48; };
        allow-query { localhost; 192.168.0.0/16; 2001:288:8005::/48; };

        // 2. 轉發設定：把不懂的查詢丟給 Google 或 Cloudflare
        forwarders {
            2001:4860:4860::6464; // Google DNS64 IPv6
```

```
        //2001:4860:4860::8888; // Google DNS IPv6
        //8.8.8.8;                    // Google DNS IPv4
    };

    // 啟用遞迴查詢
    recursion yes;

    // 3. 確保 BIND 監聽 IPv6
    listen-on-v6 { any; };

    // 關閉 DNSSEC 驗證 (建議)：
    // 因為 DNS64 合成的紀錄無法通過嚴格的 DNSSEC 驗證，設為 no
    dnssec-validation no;

    // 4. DNS64 核心設定
    // 這裡的前綴 64:ff9b::/96 必須跟你 Jool 設定的 pool6 一致
    dns64 64:ff9b::/96 {
        clients { any; };               // 允許任何人使用此轉換
        mapped { !10.0.0.0/8; !192.168.0.0/16; !172.16.0.0/12; any; }; // ! 代表「不准轉換」，any 代表「其餘皆可」
        //exclude { 2001:4860:4860::8888; 2001:db8::/32;}; // 忽略特定原生 IPv6，強制 DNS64 合成。
        //exclude { !2001::/22; any; };          // 不要忽略 2001::/22（可能是微軟的連通檢測網段），之外忽略全部原生 IPv6，強制 DNS64 合成
    };
};
```

3. 設定 named 開機自動啟動

root@alpha:~# systemctl enable named

Synchronizing state of named.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.

Executing: /usr/lib/systemd/systemd-sysv-install enable named

(二) Bind9 內容指令說明

1. named.conf.options 設定摘要說明

/etc/bind/named.conf.options 為 BIND9 DNS 伺服器的主要全域設定檔，用於設定 DNS 服務的基本運作方式，包括工作目錄、查詢權限、遞迴查詢、上游 DNS（Forwarders）、監聽介面及 DNSSEC 等功能。在本 IPv6-Only 無線網路平台中，此設定檔最重要的功能是啟用 DNS64，當用戶端查詢僅具有 IPv4 位址的網站時，BIND9 會自動依據 64:ff9b::/96 前綴合成對應的 AAAA 紀錄，使 IPv6-Only 裝置能透過 NAT64（Jool）存取 IPv4 網路資源。此外，設定檔亦限制只有內部網段可以使用本 DNS 服務，以避免成為公開 DNS（Open Resolver）而遭受濫用，同時指定 Google DNS64 作為上游 DNS，提供完整且穩定的網域名稱解析服務，確保 IPv6-Only 網路環境中的各類應用程式皆能正常運作。

2. named.conf.options 設定內容指令說明如下表：

指令	指令說明
options {	開始定義 BIND9 的全域選

	項設定。
directory "/var/cache/bind";	指定 BIND9 執行時使用的工作目錄與快取資料存放位置。
allow-query { localhost; 192.168.0.0/16; 2001:288:8005::/48; };	限制可使用此 DNS 伺服器查詢的來源，只允許本機、IPv4 內網 192.168.0.0/16 及 IPv6 網段 2001:288:8005::/48 查詢，避免成為公開 DNS。
forwarders { ... };	設定上游 DNS 轉發伺服器，當本機無法直接解析時，會將查詢轉送給指定 DNS。
2001:4860:4860::6464;	指定 Google DNS64 IPv6 伺服器作為上游 DNS，可協助 IPv6-Only 環境解析 IPv4 Only 網站。
//2001:4860:4860::8888;	註解行，未啟用。若啟用，表示使用 Google IPv6 DNS 作為上游 DNS。
//8.8.8.8;	註解行，未啟用。若啟用，表示使用 Google IPv4 DNS 作為上游 DNS。
recursion yes;	啟用遞迴查詢，使 BIND9 可代替用戶端向其他 DNS 查詢完整解析結果。

listen-on-v6 { any; };	讓 BIND9 監聽所有 IPv6 位址，接受 IPv6 用戶端的 DNS 查詢。
dnssec-validation no;	關閉 DNSSEC 驗證。因 DNS64 會合成 AAAA 紀錄，可能無法通過嚴格 DNSSEC 驗證，因此此處設定為關閉。
dns64 64:ff9b::/96 {	啟用 DNS64 功能，使用 64:ff9b::/96 作為 IPv6 合成前綴，必須與 NAT64/Jool 的 pool6 設定一致。
clients { any; };	允許所有符合 DNS 查詢權限的用戶端使用 DNS64 合成功能。
mapped { !10.0.0.0/8; !192.168.0.0/16; !172.16.0.0/12; any; };	設定哪些 IPv4 位址可被 DNS64 合成為 IPv6 位址；！表示排除私有 IPv4 網段不轉換，其餘 IPv4 位址皆可轉換。
//exclude { 2001:4860:4860::8888; 2001:db8::/32;};	註解行，未啟用。若啟用，可排除特定 IPv6 位址或網段，不進行 DNS64 處理。
//exclude { !2001::/22; any; };	註解行，未啟用。若啟用，表示不排除 2001::/22，但排除其他原生 IPv6 位址，可用於特定連通性偵測或強制

	DNS64 測試。
};	結束 dns64 設定區塊。
};	結束 options 全域設定區塊。

這份 named.conf.options 的用途：它主要是給 IPv6-Only / NAT64 / DNS64 環境使用，讓沒有 AAAA 紀錄的 IPv4 網站可被合成為 IPv6 位址。這份設定是讓 BIND 作為：IPv6-Only 用戶端的 DNS Server，並具備：

(1) DNS 遞迴查詢

(2) DNS Forwarder 轉送

(3) DNS64 位址合成

(4) 配合 Jool NAT64 使用

(5) 限制內網查詢，避免變成公開 DNS

九、系統安裝完成後注意事項

1. 如果使用 VMware Workstation Pro 有效能考量：請在虛擬機設定中（.vmx 檔），將網卡類型（ethernet?.virtualDev）全部從 e1000 或 e1000e 更改為 vmxnet3。
2. 網卡類型更改後，Ubuntu Server 26.04 VM 的網卡名稱會改變，請將以下/etc/netplan/01-netcfg.yaml， /etc/radvd.conf ，/etc/rc.local 這 3 個檔案的網卡名稱都要跟著調整(可能是 ens160 及 ens192)。
3. 系統安裝完成後請關機，並再重新開機以確認網卡設定的正確性。

十、系統運作測試

實際測試時，首先將智慧型手機（Android 或 iPhone）連線至 IPv6-Only Wi-Fi 無線網路（SSID），手機完成 Wi-Fi 連線後，系統會透過 Router Advertisement（RA） 自動配置 IPv6 位址，並接收網路前綴（Prefix）、預設閘道（Default Gateway）及 PREF64 等相關資訊；若裝置支援 DHCP Option 108（IPv6-Only Preferred Option），則可依設定優先使用 IPv6-Only 網路。由於尚未完成網路認證，當使用者首次開啟瀏覽器或系統自動進行網路連通性檢測時，HTTP 請求會被 nftables 重新導向至 Captive Portal 認證頁面。使用者閱讀說明後點選「開始 IPv6-Only 上網」，系統便會取得裝置的 MAC 位址並加入授權清單，完成網路存取授權。之後再次開啟瀏覽器，即可正常瀏覽網站。若存取的網站僅提供 IPv4 服務，系統會由 BIND9 DNS64 自動合成 AAAA 紀錄，再由 Jool NAT64 將 IPv6 封包轉換為 IPv4 封包，使手機即使未取得任何 IPv4 位址，仍可順利存取 IPv4 Only 網站。此外，可利用手機瀏覽器測試 IPv6 網站（例如 https://test-ipv6.com）及 IPv4 Only 網站，確認 DNS64/NAT64 轉換是否正常；亦可檢查手機 Wi-Fi 詳細資訊，確認僅取得 IPv6 位址而未配置 IPv4 位址，以驗證 IPv6-Only Wi-Fi 平台 的建置與運作是否符合預期。

本測試使用手機連結 IPv6-Only Wi-Fi 平台，並將手機畫面紀錄如下：

1. 手機搜尋無線 AP 的 SSID：ipv6only，並進行連結。

2. 系統登入頁面

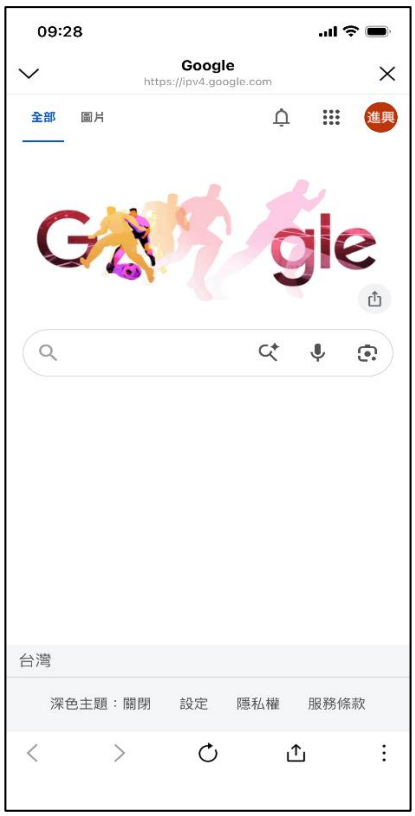
3. 完成登入系統

4. 手機取得的 IPv6 位址

5. 手機取得的 IPv4 位址

6. 連結 IPv4 Only 網頁 http://ipv4.google.com

IPv4 Address: 142.250.77.206



7. 連結 IPv4 Only 網頁 http:// www.ncku.edu.tw

IPv4 Address: 140.116.248.21

8. 連結 Dual stack 網頁 https://www.youtube.com/

IPv4 Address: 142.250.196.206，IPv6 Address: 2404:6800:4012:2::200e

系統測試結果，顯示手機可以正常登入系統，連結網際網路並存取 IPv4 Only 的網頁及 Dual stack(雙協定)的網頁，表示系統能正常運作提供 IPv6-Only 的服務。