MỤC LỤC

[**GIỚI THIỆU VỀ UBUNTU** 1](#_Toc211505645)

[**HOSTNAME** 2](#_Toc211505646)

[**1. Đổi hostname (đặt lại tên máy):** 3](#_Toc211505647)

[**2. Xem hostname hiện tại:** 3](#_Toc211505648)

[**3. Cập nhật hostname vào file /etc/hosts:** 4](#_Toc211505649)

[**GIẢI THÍCH VỀ SHELL PROMPT** 12](#_Toc211505650)

[**ROOT USER VÀ USER** 13](#_Toc211505651)

[**1. "User" (người dùng) trong hệ điều hành là gì?** 13](#_Toc211505652)

[**2. Có mấy loại user trong Linux?** 13](#_Toc211505653)

[**3. Root user là gì?** 13](#_Toc211505654)

[**4. User thường là gì?** 14](#_Toc211505655)

[**5. sudo là gì?** 14](#_Toc211505656)

[**6. Phân biệt nhanh** 14](#_Toc211505657)

[**CHUYỂN TỪ USER SANG ROOT** 15](#_Toc211505658)

[**Cách 1: Chuyển tạm sang root bằng sudo -i** 15](#_Toc211505659)

[**Cách 2: Chạy từng lệnh với sudo (an toàn hơn)** 16](#_Toc211505660)

[**Cách 3: Đăng nhập hẳn vào root (ít dùng)** 16](#_Toc211505661)

[**TẠO USER MỚI TRÊN UBUNTU SERVER VÀ PHÂN QUYỀN SUDO (QUYỀN QUẢN TRỊ)** 18](#_Toc211505662)

[**Bước 1: Tạo user mới** 18](#_Toc211505663)

[**Bước 2: Cấp quyền sudo cho user mới** 18](#_Toc211505664)

[**Bước 3: (Tuỳ chọn) Kiểm tra xem user mới có quyền sudo chưa** 19](#_Toc211505665)

[**Bước 4: (Tùy chọn) Xem danh sách user trên hệ thống** 19](#_Toc211505666)

[**Bước 5: (Tùy chọn) Xóa user nếu không cần nữa** 20](#_Toc211505667)

[**LIỆT KÊ CÁC FILE VÀ THƯ MỤC TRONG MỘT THƯ MỤC** 20](#_Toc211505668)

[**Nội dung 1: xem file và thư mục con của thư mục hiện tại** 20](#_Toc211505669)

[**Nội dung 2: di chuyển sang thư mục khác** 21](#_Toc211505670)

[**Nội dung 3: ký hiệu ./ và ../ là gì?** 22](#_Toc211505671)

[**Tại sao khi đang ở thư mục /home của user, khi dùng câu lệnh ls không hiển thị gì?** 23](#_Toc211505672)

[**Kiểm tra xem mình đang ở thư mục nào** 24](#_Toc211505673)

[**Hiển thị chi tiết, kể cả file ẩn** 25](#_Toc211505674)

[**TẠO THƯ MỤC VÀ FILE** 27](#_Toc211505675)

[**CÁCH GHI NỘI DUNG TRONG FILE** 29](#_Toc211505676)

[**Cách 1: Dùng lệnh echo để ghi nhanh một dòng** 29](#_Toc211505677)

[**Cách 2: Dùng echo để thêm dòng mới vào file (không xoá nội dung cũ)** 29](#_Toc211505678)

[**Cách 3: Dùng trình soạn thảo nano để gõ nội dung trực tiếp** 30](#_Toc211505679)

[**Cách 4 (nâng cao): Dùng vim (nếu bạn muốn học trình soạn thảo chuyên sâu)** 31](#_Toc211505680)

[**vi là gì?** 31](#_Toc211505681)

[**THOÁT TÀI KHOẢN, KHỞI ĐỘNG LẠI, TẮT MÁY CHỦ** 36](#_Toc211505682)

[**1. Thoát người dùng hiện tại bằng dòng lệnh** 36](#_Toc211505683)

[**2. Khởi động lại server** 37](#_Toc211505684)

[**3. Tắt máy theo thời gian chỉ định (shutdown)** 37](#_Toc211505685)

[**4. Tắt máy ngay lập tức:** 38](#_Toc211505686)

[**QUẢN LÝ MẠNG TRONG UBUNTU SERVER** 39](#_Toc211505687)

[**1. Xem địa chỉ IP hiện tại của máy** 39](#_Toc211505688)

[**2. Xem địa chỉ gateway mặc định** 40](#_Toc211505689)

[**3. Xem DNS server mà hệ thống đang dùng** 40](#_Toc211505690)

[**4. Cấu hình địa chỉ IP tĩnh (manual / static IP)** 41](#_Toc211505691)

[**5. Áp dụng cấu hình mạng mới** 42](#_Toc211505692)

[**6. Kiểm tra kết nối internet** 42](#_Toc211505693)

[**CẤU HÌNH IP TĨNH** 43](#_Toc211505694)

[**1. Từ khóa "WARNING"** 44](#_Toc211505695)

[**2. gateway4 has been deprecated** 44](#_Toc211505696)

[**3. Câu: “Use default routes instead”** 44](#_Toc211505697)

[**4. CÁCH VIẾT MỚI (theo tài liệu Netplan hiện đại)** 45](#_Toc211505698)

[**5. Mục đích thay đổi file YAML trong Netplan** 46](#_Toc211505699)

[**6. Nhưng tại sao phải dùng IP tĩnh? Sao không để tự động (DHCP) luôn?** 48](#_Toc211505700)

[**MỤC ĐÍCH CỦA VIỆC DÙNG PING 8.8.8.8 VÀ PING GOOGLE.COM** 49](#_Toc211505701)

[**1. ping 8.8.8.8 là để kiểm tra** 50](#_Toc211505702)

[**2. ping google.com là để kiểm tra** 50](#_Toc211505703)

[**3. Cách kiểm tra – sửa lỗi** 51](#_Toc211505704)

[**4. Cách sửa lại để mạng hoạt động** 54](#_Toc211505705)

[**LỖI CẤU HÌNH MẠNG** 56](#_Toc211505706)

[**1. Destination Host Unreachable (từ IP 192.168.1.100)** 57](#_Toc211505707)

[**2. Temporary failure in name resolution** 57](#_Toc211505708)

[**3. Dùng IP động (cho đơn giản và chắc chắn hơn lúc mới học)** 59](#_Toc211505709)

[**4. Điều kiện để máy ảo ubuntu server kết nối được internet** 60](#_Toc211505710)

[**5. Nếu vẫn chưa được, bạn hãy thử cách DỄ NHẤT:** 63](#_Toc211505711)

[**6. Bạn đã kết nối mạng thành công!** 63](#_Toc211505712)

[**CÂU LỆNH PING** 65](#_Toc211505713)

[**1. "ping" là gì?** 65](#_Toc211505714)

[**2. Câu lệnh ping dùng để làm gì?** 66](#_Toc211505715)

[**3. Cách dùng cơ bản** 66](#_Toc211505716)

[**4. Khi bạn chạy lệnh ping, máy sẽ làm gì?** 66](#_Toc211505717)

[**5. Ping chạy mãi, dừng lại bằng cách nào?** 67](#_Toc211505718)

[**6. Sau khi dừng, nó hiển thị thống kê** 67](#_Toc211505719)

[**7. Nếu ping bị lỗi thì sao?** 67](#_Toc211505720)

[**DEFAULT GATEWAY** 68](#_Toc211505721)

[**SỰ KHÁC NHAU GIỮA HOSTNAME, DOMAIN NAME VÀ FQDN** 71](#_Toc211505722)

[**1. Hostname – TÊN MÁY** 71](#_Toc211505723)

[**2. Domain Name – TÊN MIỀN** 72](#_Toc211505724)

[**3. FQDN – Fully Qualified Domain Name** 73](#_Toc211505725)

[**CÀI PHẦN MỀM** 75](#_Toc211505726)

[**1. Cập nhật danh sách gói phần mềm (apt update)** 75](#_Toc211505727)

[**2. Nâng cấp các gói đã lỗi thời (apt upgrade)** 75](#_Toc211505728)

[**CÀI ĐẶT NGINX** 76](#_Toc211505729)

[**1. Cài đặt Gói Nginx** 76](#_Toc211505730)

[**2. Kiểm tra Trạng thái Dịch vụ** 76](#_Toc211505731)

[**3. Điều chỉnh Tường lửa (Firewall)** 77](#_Toc211505732)

[**4. Kiểm tra Truy cập Web từ Trình duyệt** 78](#_Toc211505733)

[**5. Tổng kết** 79](#_Toc211505734)

[**TƯỜNG LỬA** 81](#_Toc211505735)

[**HIỂN THỊ WEBSITE** 90](#_Toc211505736)

[**Bài tập 1: Sử dụng curl để lấy thông tin các quốc gia và thủ đô của họ trên toàn thế giới với URL: https://countriesnow.space/api/v0.1/countries/capital** 92](#_Toc211505737)

[**1. Tổng quan về Lệnh curl và API** 92](#_Toc211505738)

[**2. Các Bước Thực hiện Bài tập** 92](#_Toc211505739)

[**3. Các Biến thể Lệnh curl Nâng cao (Tùy chọn)** 93](#_Toc211505740)

[**XÓA PHẦN MỀM** 95](#_Toc211505741)

[**THIẾT LẬP THỜI GIAN** 95](#_Toc211505742)

# **GIỚI THIỆU VỀ UBUNTU**

**🔷 Giới thiệu về Ubuntu**

**Ubuntu** là một hệ điều hành **miễn phí** và **mã nguồn mở**, được xây dựng dựa trên nền tảng **Linux**. Đây là một trong những hệ điều hành phổ biến nhất hiện nay, được sử dụng rộng rãi trên nhiều loại nền tảng khác nhau, bao gồm:

* **Máy tính cá nhân** (desktop)
* **Máy chủ** (server)
* **Hạ tầng điện toán đám mây** (cloud computing)

Ubuntu không chỉ phục vụ nhu cầu người dùng thông thường mà còn đáp ứng tốt các yêu cầu khắt khe trong môi trường doanh nghiệp, trung tâm dữ liệu và công nghệ mới nổi như IoT (Internet of Things).

**🔷 Lịch sử và nhà phát triển**

Ubuntu được phát triển và duy trì bởi công ty **Canonical Ltd**, một doanh nghiệp công nghệ có trụ sở tại Anh. Phiên bản đầu tiên của Ubuntu được phát hành vào **tháng 10 năm 2004**. Từ đó đến nay, Ubuntu đã không ngừng phát triển và trở thành một trong những hệ điều hành Linux dễ sử dụng nhất.

**🔷 Nền tảng Debian**

Ubuntu được xây dựng dựa trên **Debian**, một bản phân phối Linux nổi tiếng khác. Tuy nhiên, so với Debian – vốn nổi tiếng với độ ổn định nhưng khá “khó nhằn” với người mới – Ubuntu tập trung mạnh vào:

* **Tính dễ sử dụng**: Giao diện thân thiện, dễ tiếp cận cho cả người mới và người dùng chuyên sâu.
* **Cập nhật định kỳ**: Ubuntu có chu kỳ phát hành rõ ràng (2 lần mỗi năm), giúp người dùng luôn tiếp cận công nghệ mới.
* **Dễ cài đặt**: Quá trình cài đặt Ubuntu đơn giản, nhanh chóng, hỗ trợ tốt cho cả cài đặt thủ công và tự động.

**🔷 Các phiên bản của Ubuntu**

Ubuntu được phân phối dưới nhiều **phiên bản (editions)** khác nhau, phù hợp với từng mục đích sử dụng cụ thể:

**➤ Ubuntu Desktop**

Dành cho **máy tính cá nhân** và **máy tính xách tay (laptop)**.  
Phiên bản này có giao diện đồ họa trực quan, dễ sử dụng, phù hợp với người dùng thông thường thay thế cho Windows hoặc macOS.

**➤ Ubuntu Server**

Dành cho các **môi trường máy chủ**, **điện toán đám mây** và **trung tâm dữ liệu**.  
Phiên bản này không có giao diện đồ họa mặc định, tập trung vào hiệu suất, tính ổn định và bảo mật cho các hệ thống backend hoặc dịch vụ web.

**➤ Ubuntu Core**

Là phiên bản **tối giản**, nhẹ và bảo mật cao, được thiết kế dành riêng cho các thiết bị **IoT (Internet of Things)** và các **ứng dụng dạng container** (như Docker, Snap).  
Phiên bản này rất phù hợp cho các thiết bị thông minh, nhúng, hoặc hệ thống tự động hóa nhỏ gọn.

**✅ Tóm lại**

Ubuntu là một hệ điều hành mạnh mẽ, miễn phí và dễ sử dụng, phù hợp với cả cá nhân lẫn doanh nghiệp. Nhờ vào sự linh hoạt và hỗ trợ tốt từ cộng đồng, Ubuntu ngày càng được tin dùng trong nhiều lĩnh vực từ máy tính để bàn đến hạ tầng đám mây và thiết bị IoT.

# **HOSTNAME**

Sau khi bạn **cài đặt Ubuntu Server trên VMware**, có 3 phần rất quan trọng bạn cần hiểu và làm quen đầu tiên:

1. **Quản lý tên máy (hostname)**
2. **Quản lý địa chỉ IP**
3. **Quản lý ngày giờ hệ thống (DateTime)**

Trong nội dung bạn đưa ra, mình sẽ tập trung **giải thích chi tiết phần hostname**, còn phần IP và DateTime bạn có thể học tiếp sau (mình cũng có thể giải nếu bạn cần).

**🔷 1. Khái niệm cơ bản: Ubuntu Server là gì?**

Trước tiên, bạn cần biết:

* **Ubuntu** là một hệ điều hành mã nguồn mở dựa trên Linux.
* **Ubuntu Server** là phiên bản Ubuntu không có giao diện đồ họa (chỉ có màn hình dòng lệnh), thường dùng cho **máy chủ (server)** để chạy web, ứng dụng, cơ sở dữ liệu, v.v.
* **VMware** là phần mềm giúp bạn tạo một **máy ảo** – tức là một máy tính ảo bên trong máy tính thật, để cài hệ điều hành như Ubuntu Server và học mà không ảnh hưởng đến máy thật.

**🔶 2. Hostname là gì?**

**Hostname** là **tên định danh của máy tính** trong một mạng.

🔧 Ví dụ:

* Máy bạn có thể có hostname là: server01, webserver, ubuntu-test, v.v.
* Giống như tên gọi để các máy tính khác trên mạng nhận diện máy này.

🧠 Tưởng tượng: Nếu các máy tính trong mạng là con người, thì **hostname chính là tên của từng người**.

**✅ Các thao tác cơ bản với hostname:**

Trong Ubuntu Server, mọi thứ bạn làm là **qua dòng lệnh (terminal)**. Dưới đây là các lệnh bạn cần:

## **1. Đổi hostname (đặt lại tên máy):**

**sudo hostnamectl set-hostname NEW\_NAME**

🧾 Giải thích từng phần:

| **Phần lệnh** | **Ý nghĩa** |
| --- | --- |
| sudo | Chạy lệnh với quyền quản trị (admin). Cần nhập mật khẩu. |
| hostnamectl | Công cụ dùng để quản lý hostname và thông tin hệ thống. |
| set-hostname | Lệnh con để thay đổi tên máy |
| NEW\_NAME | Thay bằng tên bạn muốn đặt, ví dụ: myserver, ubuntu01 |

📌 Ví dụ: Muốn đổi hostname thành us2:

**sudo hostnamectl set-hostname us2**

## **2. Xem hostname hiện tại:**

**hostnamectl**

📋 Lệnh này sẽ hiển thị ra thông tin hệ thống, trong đó có dòng:

**Static hostname: us2**

→ Đây là hostname hiện tại của máy bạn.

## **3. Cập nhật hostname vào file /etc/hosts:**

**❓Tại sao cần làm điều này?**

* Khi bạn đổi hostname, bạn cần cập nhật nó vào file /etc/hosts để **hệ thống vẫn biết hostname đó là của "localhost" (máy hiện tại)**.
* Nếu không cập nhật, một số dịch vụ sẽ **không hoạt động đúng**, vì hệ thống không biết tên mới là của máy hiện tại.

**🧱 File /etc/hosts là gì?**

* Đây là một file hệ thống giúp **áp tên (hostname) vào địa chỉ IP**.
* Nó giống như **danh bạ điện thoại**, ánh xạ giữa tên và số IP.

📁 Để xem nội dung file:

**cat /etc/hosts**

📥 Một dòng ví dụ trong file:

**127.0.0.1 localhost**

**127.0.1.1 tên hostname cũ**

→ 127.0.1.1 là IP cục bộ dùng để trỏ đến máy này với tên us2.

**✏️ Cách chỉnh sửa file /etc/hosts:**

1. Mở file bằng trình soạn thảo (ví dụ nano):

**sudo nano /etc/hosts**

1. Tìm dòng có hostname cũ, thay bằng hostname mới:

**Ví dụ: Đổi từ ubuntu sang server01:**

**127.0.0.1 localhost**

**127.0.1.1 us2 (tên hostname mới)**

1. Nhấn Ctrl + O để lưu file → nhấn Enter
2. Nhấn Ctrl + X để thoát

Mình sẽ giúp bạn **hiểu gốc rễ từ nền tảng, từng khái niệm một cách cực kỳ rõ ràng** – kể cả bạn là người **chưa có kiến thức gì trước đó**, thì đọc xong bạn sẽ hiểu **vì sao khi đổi hostname, ta phải sửa file /etc/hosts**.

**🧠 1. Trước tiên: Hostname là gì?**

* **Hostname** là **tên định danh của máy tính trong hệ thống**.
* Nó giống như **"tên gọi"** của máy bạn trong mạng.

📌 Ví dụ:

* Bạn có thể đặt tên máy là: ubuntu-server1, web-server, nammoadidaphat, v.v.
* Khi bạn mở terminal và thấy dòng:

lebao@nammoadidaphat:~$

→ Thì nammoadidaphat chính là **hostname**.

**🧠 2. Vậy /etc/hosts là gì?**

File **/etc/hosts** là một **file đặc biệt trên Linux** (và cả Windows có file tương tự).  
Nó dùng để **liên kết (map) giữa hostname và địa chỉ IP**.

📌 Bạn có thể coi nó như **“danh bạ điện thoại” cho hệ thống**, ví dụ:

127.0.0.1 localhost

127.0.1.1 ubuntu-server1

| **Dòng này nghĩa là gì?** |
| --- |
| 127.0.0.1 localhost → máy bạn tự nhận mình là localhost |
| 127.0.1.1 ubuntu-server1 → máy bạn cũng tự nhận tên là ubuntu-server1 |

💡 Khi chương trình trong máy cần **gửi dữ liệu đến hostname**, nó sẽ **tra trong file /etc/hosts** trước để tìm địa chỉ IP tương ứng.

**💥 3. Vấn đề xảy ra nếu bạn chỉ đổi hostname mà không sửa /etc/hosts**

**🎯 Giả sử:**

* Ban đầu hostname là ubuntu-server1
* Trong /etc/hosts có dòng:

127.0.1.1 ubuntu-server1

→ Mọi thứ OK vì hostname khớp với file hosts.

**😬 Nhưng nếu bạn đổi hostname thành nammoadidaphat bằng lệnh:**

sudo hostnamectl set-hostname nammoadidaphat

→ Lúc này:

* Máy bạn **nghĩ rằng tên của mình là nammoadidaphat**
* Nhưng trong /etc/hosts vẫn ghi 127.0.1.1 ubuntu-server1

👉 Vậy là có **mâu thuẫn giữa tên thực và “danh bạ”** → gây lỗi tiềm ẩn!

**❗ Điều gì có thể xảy ra nếu không sửa /etc/hosts?**

| **Hậu quả tiềm ẩn** | **Vì sao lại xảy ra** |
| --- | --- |
| Một số lệnh không chạy được (như sudo, ping,...) | Vì hệ thống không thể "tự tìm chính mình" qua hostname |
| Lỗi trong các chương trình như Apache, MySQL, Docker,... | Vì nhiều dịch vụ cần biết chính xác địa chỉ máy thông qua hostname |
| Thông báo lỗi kiểu “unable to resolve hostname” | Vì hostname mới không có trong /etc/hosts |

**✅ Kết luận: Vì sao cần sửa /etc/hosts khi đổi hostname?**

**👉 Vì:**

* Hệ thống cần **biết chính xác hostname tương ứng với IP nội bộ**
* Mà /etc/hosts là nơi lưu bản đồ liên kết giữa hostname ↔ IP
* Nếu không sửa, **hostname mới không “có trong danh bạ” của máy**, sẽ dẫn đến lỗi khi dùng nhiều dịch vụ

**✅ Cách sửa /etc/hosts sau khi đổi hostname**

Giả sử bạn vừa đổi hostname thành nammoadidaphat, bạn cần:

**Mở file:**

sudo nano /etc/hosts

**Tìm dòng:**

127.0.1.1 ubuntu-server1

**Đổi lại thành:**

127.0.1.1 nammoadidaphat

⚠️ Không được sửa dòng 127.0.0.1 localhost nhé – dòng đó luôn luôn giữ nguyên!

**📌 Tóm tắt dễ nhớ cho người mới:**

| **Việc làm** | **Mục đích** |
| --- | --- |
| Đổi hostname bằng hostnamectl | Đổi tên máy trong hệ thống |
| Sửa file /etc/hosts | Đảm bảo hostname mới có trong "danh bạ nội bộ" của máy |
| Kiểm tra hostname | Dùng lệnh hostnamectl hoặc hostname |
| Kiểm tra file hosts | Mở bằng cat /etc/hosts |

Bạn có thể thử thực hành:

**1. Đổi tên hostname:**

sudo hostnamectl set-hostname nammoadidaphat

**2. Sửa /etc/hosts:**

sudo nano /etc/hosts

→ Đổi dòng 127.0.1.1 ubuntu-server1 thành 127.0.1.1 nammoadidaphat

**3. Khởi động lại:**

sudo reboot

→ Kiểm tra lại, bạn sẽ thấy tên mới hiển thị:

lebao@nammoadidaphat:~$

Tôi vẫn chưa hiểu lý do tại sao đổi hostname lại liên quan đến Cập nhật file /etc/hosts vì tôi là người mới bắt đầu học từ con số 0, nên không hiểu rõ ngọn ngành gốc gác nó như thế nào cả

**ChatGPT said:**

Câu hỏi của bạn rất hay — và quan trọng với người mới bắt đầu học Linux.

Việc **đổi hostname** và **cập nhật file /etc/hosts** có **mối liên hệ chặt chẽ**, đặc biệt là trong cách hệ điều hành **nhận diện chính nó** và **giao tiếp mạng nội bộ**.

Mình sẽ **giải thích từ gốc đến ngọn**, đúng kiểu người mới bắt đầu từ con số 0 cũng hiểu được. Bắt đầu nha 👇

**🔸 1. Hostname là gì?**

**👉 "Hostname" là tên định danh của máy tính trong mạng.**

* Khi bạn cài Ubuntu, bạn sẽ đặt một cái tên cho máy, ví dụ: ubuntu-server1
* Tên này dùng để:
  + Hiển thị trên dòng lệnh: lebao@ubuntu-server1:~$
  + Giao tiếp với các máy khác trong mạng (giống như tên trong mạng LAN)
  + Ghi log, chạy dịch vụ, quản lý qua SSH, v.v.

🧠 Bạn có thể hiểu đơn giản:

Hostname giống như **tên của con người** để người khác gọi/nhận diện, thay vì dùng số chứng minh nhân dân (IP) khô khan.

**🔸 2. File /etc/hosts là gì?**

**👉 /etc/hosts là một “sổ danh bạ” nội bộ của hệ điều hành.**

* Nó ánh xạ (map) giữa **tên** (hostname) và **địa chỉ IP**.
* Khi bạn dùng một tên máy (hostname), hệ thống sẽ **tra cứu file /etc/hosts** để biết tên đó ứng với IP nào — **trước khi hỏi DNS bên ngoài**.

**🔍 Ví dụ nội dung của /etc/hosts:**

127.0.0.1 localhost

127.0.1.1 ubuntu-server1

* 127.0.0.1: địa chỉ loopback (máy tự gọi chính nó)
* ubuntu-server1: hostname của máy

🧠 Nói cách khác:

**Khi bạn gõ ping ubuntu-server1**, hệ điều hành sẽ xem trong file /etc/hosts để biết ubuntu-server1 là địa chỉ IP nào.

**🔸 3. Tại sao đổi hostname cần cập nhật /etc/hosts?**

**Vì:**

* Nếu bạn đổi hostname từ ubuntu-server1 thành web-server
* Mà file /etc/hosts vẫn giữ dòng:
* 127.0.1.1 ubuntu-server1
* 👉 Thì hệ điều hành **không thể tự nhận diện tên mới** web-server.

📌 Kết quả là:

* Một số chương trình sẽ **không hoạt động đúng** (vì hostname không khớp)
* Có thể gặp lỗi khi khởi động dịch vụ (Apache, MySQL…)
* Gõ ping web-server có thể báo lỗi “host not found”

**🔧 Vì vậy bạn cần sửa lại /etc/hosts:**

**Trước:**

127.0.1.1 ubuntu-server1

**Sau khi đổi hostname:**

127.0.1.1 web-server

✅ Khi đó:

* Hệ điều hành sẽ hiểu web-server chính là máy này.
* Các dịch vụ sẽ hoạt động bình thường.
* Các lệnh như ping, hostname, v.v. sẽ khớp nhau.

**✅ Tóm tắt dễ hiểu:**

| **Thành phần** | **Vai trò** | **Khi đổi hostname thì sao?** |
| --- | --- | --- |
| hostname | Tên máy | Cần đổi bằng hostnamectl |
| /etc/hosts | Sổ địa chỉ tên ↔ IP | Cần sửa tên cũ thành tên mới |
| Không sửa /etc/hosts | Hệ thống không biết tên mới là ai | Dịch vụ có thể lỗi, hostname không khớp |

Nếu bạn muốn mình làm một **minh họa thực tế** (trước – sau khi đổi hostname) hoặc viết **script tự động đổi luôn cả hostname và file hosts**, mình sẵn sàng hỗ trợ!

Bạn thấy phần nào còn mơ hồ không? Cứ hỏi tiếp nhé.

**You said:**

câu lệnh ping ubuntu-server1 có tác dụng gì vậy? tôi gõ và nahasn enter cái nó không dừng lại luôn

**ChatGPT said:**

Rất tốt! Bạn đang **tìm hiểu thực tế** cách lệnh ping hoạt động — đây là một trong những lệnh cơ bản nhưng **rất quan trọng** trong Linux và mạng máy tính nói chung.

Giờ mình sẽ **giải thích chi tiết và dễ hiểu** cho bạn:

**✅ Câu hỏi 1: Lệnh ping ubuntu-server1 có tác dụng gì?**

**👉 Mục đích chính:**

Lệnh ping dùng để:

* **Kiểm tra kết nối mạng** giữa máy của bạn và một máy khác (hoặc chính máy bạn).
* Xác định xem hostname (tên máy) như ubuntu-server1 có được **nhận diện và phân giải thành IP hay không**.

**👇 Khi bạn gõ:**

ping ubuntu-server1

Hệ điều hành sẽ:

1. Tìm xem ubuntu-server1 có ánh xạ với IP nào (trong /etc/hosts hoặc DNS).
2. Gửi một gói tin **ICMP echo request** đến IP đó.
3. Chờ phản hồi (ICMP echo reply).
4. In ra kết quả từng dòng, ví dụ:

PING ubuntu-server1 (127.0.1.1) 56(84) bytes of data.

64 bytes from ubuntu-server1 (127.0.1.1): icmp\_seq=1 ttl=64 time=0.045 ms

64 bytes from ubuntu-server1 (127.0.1.1): icmp\_seq=2 ttl=64 time=0.046 ms

...

🧠 Nếu bạn thấy dòng như trên → **ubuntu-server1 có tồn tại và máy đang kết nối được với nó.**

**✅ Câu hỏi 2: Tại sao lệnh ping không dừng lại?**

**👉 Trên Linux, lệnh ping mặc định sẽ chạy mãi không dừng — trừ khi bạn bấm thủ công để dừng.**

**✅ Cách dừng lệnh ping:**

➡️ **Nhấn Ctrl + C**  
→ Lệnh sẽ dừng lại và hiển thị **thống kê kết quả**, ví dụ:

--- ubuntu-server1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4095ms

**✅ Mẹo nhỏ: Ping trong thời gian hoặc số lần nhất định**

**🔸 Ping chỉ 4 lần:**

ping -c 4 ubuntu-server1

→ Máy sẽ gửi 4 gói ping, rồi tự dừng.

**✅ Tóm lại**

| **Câu hỏi** | **Trả lời** |
| --- | --- |
| ping ubuntu-server1 để làm gì? | Kiểm tra xem máy có nhận diện được hostname ubuntu-server1 và có kết nối tới nó hay không |
| Tại sao không dừng? | Trên Linux, ping chạy liên tục cho đến khi bạn **nhấn Ctrl + C** |
| Muốn nó tự dừng thì sao? | Dùng ping -c [số\_lần], ví dụ ping -c 4 ubuntu-server1 |

# **GIẢI THÍCH VỀ SHELL PROMPT**

Cụm **"lebao@ubuntu-server1:~$"** có một tên gọi chính thức trong thế giới Linux/Unix.

**✅ Tên gọi chính xác: Shell Prompt (hoặc đơn giản là Prompt)**

**👉 "Prompt" nghĩa là dấu nhắc lệnh, là phần:**

* Hiển thị trước mỗi dòng nhập lệnh trong terminal.
* Cho bạn biết **ai đang đăng nhập**, **đang ở đâu**, và **máy nào**.

**✅ Giải thích từng phần của lebao@ubuntu-server1:~$**

A computer screen shot of a computer code

AI-generated content may be incorrect.

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| lebao | Tên người dùng đang đăng nhập |
| @ | Phân tách giữa user và hostname |
| ubuntu-server1 | Tên máy (hostname) đang sử dụng |
| : | Phân tách giữa tên máy và thư mục |
| ~ | Thư mục hiện tại (~ là viết tắt cho thư mục home của user) |
| $ | Dấu nhắc lệnh, báo hiệu hệ thống đang chờ bạn nhập lệnh |
| # | Nếu là root user, dấu nhắc sẽ là # thay vì $ |

**🔁 Tùy chỉnh prompt**

Bạn có thể thay đổi cách hiển thị prompt bằng biến môi trường PS1. Ví dụ:

**export PS1="\u@\h:\w$ "**

* **\u: username**
* **\h: hostname**
* **\w: thư mục hiện tại**

⟶ sẽ hiển thị giống như: lebao@ubuntu-server1:~/Documents$

# **ROOT USER VÀ USER**

## **1. "User" (người dùng) trong hệ điều hành là gì?**

Hệ điều hành như Linux (hoặc Windows, macOS…) luôn cần có **người dùng (user)** để đăng nhập và sử dụng.

* Mỗi user có **tài khoản riêng**, **mật khẩu riêng** và **quyền hạn riêng**.
* Khi bạn đăng nhập vào Ubuntu với tên như lebao, thì bạn đang dùng máy với tư cách **một người dùng (user)**.

## **2. Có mấy loại user trong Linux?**

Trong Linux, có 2 loại chính:

| **Loại** | **Mô tả** | **Dấu hiệu** |
| --- | --- | --- |
| **User thường** *(normal user)* | Dùng để làm việc, học tập, lướt web, lập trình, v.v. | Dấu nhắc lệnh là: $ |
| **Root user** *(siêu người dùng)* | Là **quản trị viên hệ thống** — có quyền cao nhất, làm gì cũng được | Dấu nhắc lệnh là: # |

## **3. Root user là gì?**

**➤ root là tài khoản cao nhất trong Linux**

* Có thể **làm mọi việc**: cài phần mềm, xóa hệ thống, tạo người dùng, chỉnh sửa file hệ thống...
* Không bị giới hạn bởi quyền gì hết
* Giống như **admin tuyệt đối** vậy đó

🧠 Trong Windows thì giống như **Administrator**.

**⚠️ Cảnh báo:**

Làm việc với tài khoản root rất **nguy hiểm** nếu bạn không cẩn thận. Chỉ cần một lệnh sai cũng có thể:

* Xóa toàn bộ hệ thống
* Làm hỏng Ubuntu
* Mất dữ liệu

## **4. User thường là gì?**

* Là người dùng thông thường (như bạn khi đăng nhập bằng lebao).
* Không có quyền can thiệp vào phần hệ thống.
* Muốn làm gì "nặng đô" (ví dụ: cài phần mềm, chỉnh file hệ thống) thì phải:  
  👉 dùng lệnh **sudo**

## **5. sudo là gì?**

sudo = **SuperUser DO**  
→ Tạm hiểu là: *“Làm việc này với quyền root.”*

**Ví dụ:**

| **Mục đích** | **Câu lệnh** |
| --- | --- |
| Chạy bình thường | apt update → ❌ Bị từ chối |
| Chạy với root | sudo apt update → ✅ Chạy được |

Hệ thống sẽ yêu cầu bạn nhập mật khẩu → nếu đúng, thì bạn **tạm thời được mượn quyền root** để chạy lệnh.

## **6. Phân biệt nhanh**

| **Tài khoản** | **Dấu hiệu** | **Quyền hạn** | **Rủi ro** |
| --- | --- | --- | --- |
| **root** | # | Toàn quyền hệ thống | Rất cao nếu dùng sai |
| **user thường (ví dụ: lebao)** | $ | Bị giới hạn | An toàn hơn |
| **sudo** | (lệnh đặc biệt cho user thường) | Tạm mượn quyền root | Trung bình |

**🔍 Ví dụ thực tế:**

**Bạn là user thường:**

lebao@ubuntu-server1:~$ sudo apt install nginx

→ Máy hỏi mật khẩu của lebao. Nếu đúng, lệnh được chạy với quyền root.

**✅ Tóm lại:**

* **User**: Người dùng thông thường, làm việc, học tập, không có quyền can thiệp hệ thống.
* **Root user**: Quản trị viên toàn quyền hệ thống. Có thể làm mọi thứ, kể cả xóa sạch hệ thống.
* **sudo**: Lệnh cho phép user thường tạm "mượn" quyền root để chạy lệnh nâng cao.

# **CHUYỂN TỪ USER SANG ROOT**

## **Cách 1: Chuyển tạm sang root bằng sudo -i**

**🧪 Dùng khi nào?**

* Khi bạn chỉ cần **vào chế độ root tạm thời** để làm gì đó.
* Thoát ra là quay lại user bình thường.

**🧱 Câu lệnh:**

**sudo -i**

**📌 Giải thích:**

* sudo = mượn quyền root
* -i = mở **môi trường shell** giống như bạn là root thật

**🖥️ Ví dụ:**

**lebao@ubuntu-server1:~$ sudo -i**

**[sudo] password for lebao: ← Bạn nhập mật khẩu của user `lebao`**

**root@ubuntu-server1:~# ← Bạn đã thành root rồi!**

✅ Từ đây bạn có thể chạy lệnh mà không cần gõ sudo nữa, ví dụ:

**apt update**

**apt install nginx**

**🔁 Muốn thoát khỏi root?**

**exit**

→ Bạn sẽ quay lại với user lebao.

## **Cách 2: Chạy từng lệnh với sudo (an toàn hơn)**

**🧪 Dùng khi:**

* Bạn chỉ muốn chạy 1–2 lệnh với quyền root, không cần vào hẳn tài khoản root.

**Ví dụ:**

**sudo apt update**

**sudo systemctl restart nginx**

📌 Đây là cách **an toàn hơn**, vì bạn không ở trong môi trường root quá lâu.

## **Cách 3: Đăng nhập hẳn vào root (ít dùng)**

**🛑 Chỉ dùng khi bạn đã thiết lập mật khẩu cho tài khoản root**

Mặc định Ubuntu **không cho đăng nhập root trực tiếp** vì lý do bảo mật.

**📌 Bước 1: Đặt mật khẩu cho root (chỉ cần làm 1 lần)**

**sudo passwd root**

→ Hệ thống sẽ yêu cầu bạn nhập **mật khẩu mới cho root**

**📌 Bước 2: Đăng nhập root**

**su -**

→ Nhập mật khẩu root → vào root shell

**Kết quả:**

**root@ubuntu-server1:~#**

📌 su = switch user  
su - = switch sang user root và tải môi trường của root

**✅ Cách thoát khỏi root và quay lại user thường**

**🧱 Lệnh duy nhất bạn cần:**

**exit**

**📌 Ví dụ thực tế:**

**root@ubuntu-server1:~# exit**

**logout**

**lebao@ubuntu-server1:~$**

* Bạn sẽ thấy từ dòng # (root), chuyển lại về $ (user thường).
* Dấu nhắc shell sẽ quay lại giống lúc đầu: lebao@ubuntu-server1:~$

**💡 Bạn nên dùng cách nào?**

| **Mục đích** | **Cách nên dùng** | **Ghi chú** |
| --- | --- | --- |
| Dùng tạm root để làm vài lệnh | sudo -i | Rất tiện, dễ thoát |
| Chạy 1 lệnh cần quyền root | sudo lệnh | An toàn nhất |
| Đăng nhập hẳn vào root | su - (sau khi passwd root) | Nguy hiểm hơn, ít khuyến khích |

**⚠️ Lưu ý bảo mật cực quan trọng**

* **Không nên làm mọi việc bằng tài khoản root**
* **Chỉ dùng root khi thực sự cần**
* Cẩn thận với lệnh như: rm -rf /, chmod -R, chown, mv, v.v.

**🎯 Dùng exit khi nào?**

| **Tình huống** | **Bạn dùng exit để…** |
| --- | --- |
| Đang trong shell root (root@...#) | Thoát khỏi quyền root |
| Đang ở trong sudo -i | Thoát về user thường |
| Đang dùng su - | Đăng xuất khỏi tài khoản root |

**📝 Mẹo thêm:**

Nếu bạn **đăng nhập SSH vào máy chủ** với tài khoản root, thì gõ exit lần đầu sẽ **thoát khỏi root**,  
gõ exit thêm lần nữa sẽ **thoát khỏi phiên SSH hoàn toàn** (đóng terminal).

# **TẠO USER MỚI TRÊN UBUNTU SERVER VÀ PHÂN QUYỀN SUDO (QUYỀN QUẢN TRỊ)**

## **Bước 1: Tạo user mới**

**📌 Cú pháp:**

**sudo adduser [tên\_user\_mới]**

**📥 Ví dụ:**

**sudo adduser tan**

Hệ thống sẽ yêu cầu bạn nhập:

* Mật khẩu cho user tan
* (Tùy chọn) Họ tên, số điện thoại, thông tin cá nhân (có thể bỏ qua)

**🧠 Lưu ý:**

* Thư mục home sẽ tự động tạo: /home/tan
* Sau khi tạo, user mới vẫn là **user thường**, chưa có quyền sudo

## **Bước 2: Cấp quyền sudo cho user mới**

**📌 Cách nhanh & chuẩn nhất:**

**sudo usermod -aG sudo [tên\_user\_mới]**

**📥 Ví dụ:**

**sudo usermod -aG sudo tan**

* usermod: lệnh để chỉnh sửa user
* -aG: thêm user vào một nhóm (G = group)
* sudo: là nhóm đặc biệt có quyền quản trị
* tan: tên user mới

➡️ Sau lệnh này, user tan sẽ có quyền chạy các lệnh như:

**sudo apt update**

**sudo shutdown now**

## **Bước 3: (Tuỳ chọn) Kiểm tra xem user mới có quyền sudo chưa**

**📥 Cách kiểm tra:**

1. Đăng nhập vào tài khoản tan:

**su - tan**

1. Gõ thử:

**sudo whoami**

→ Nhập mật khẩu của tan  
→ Nếu thấy kết quả là:

**root**

→ 🎉 User tan đã có quyền sudo!

## **Bước 4: (Tùy chọn) Xem danh sách user trên hệ thống**

**cut -d: -f1 /etc/passwd**

## **Bước 5: (Tùy chọn) Xóa user nếu không cần nữa**

**❗ Cẩn thận khi dùng!**

* Xóa user:

**sudo deluser tan**

* Xóa luôn cả thư mục home:

**sudo deluser --remove-home tan**

# **LIỆT KÊ CÁC FILE VÀ THƯ MỤC TRONG MỘT THƯ MỤC**

**🧱 Phần 1: Cấu trúc thư mục trong Ubuntu Server**

Hệ điều hành Ubuntu Server giống như một **cây thư mục**, tương tự như Windows. Có:

* **Thư mục gốc (root)** là /
* Bên trong chứa các thư mục con như: /home, /etc, /var, v.v.
* Mỗi thư mục lại có thể chứa thêm thư mục con hoặc file

📁 Bạn có thể tưởng tượng giống như:

**/**

**├── home/**

**│ └── lebao/**

**│ ├── file1.txt**

**│ └── Documents/**

**│ └── resume.doc**

**├── etc/**

**├── var/**

**└── usr/**

## **Nội dung 1: xem file và thư mục con của thư mục hiện tại**

**✅ Cú pháp:**

**ls**

**📌 Ý nghĩa:**

* ls là viết tắt của “list” – **liệt kê ra** các **file và thư mục con** trong thư mục mà bạn đang đứng hiện tại.

**📦 Ví dụ:**

Giả sử bạn đang ở thư mục /home/lebao, bạn gõ:

ls

Kết quả có thể ra:

Documents file1.txt test.sh

👉 Điều này nghĩa là trong thư mục hiện tại đang có:

* 1 thư mục con tên: Documents
* 2 file tên: file1.txt và test.sh

**🛠️ Một số tùy chọn thường dùng với ls:**

| **Lệnh** | **Ý nghĩa** |
| --- | --- |
| ls -l | Hiển thị dạng danh sách chi tiết (dung lượng, ngày tạo, v.v.) |
| ls -a | Hiển thị cả các file ẩn (bắt đầu bằng dấu chấm .) |
| ls -la | Kết hợp cả hai trên |

📌 Ví dụ:

**ls -la**

Hiển thị chi tiết và cả file ẩn trong thư mục hiện tại.

## **Nội dung 2: di chuyển sang thư mục khác**

**✅ Cú pháp:**

**cd <tên\_thư\_mục>**

**📌 Ý nghĩa:**

* cd là viết tắt của **change directory** – nghĩa là **chuyển sang thư mục khác**

**📦 Ví dụ:**

Giả sử bạn đang ở /home/lebao và bạn muốn vào thư mục Documents bên trong nó, bạn gõ:

**cd Documents**

Giờ bạn đã chuyển sang thư mục:

**/home/lebao/Documents**

Nếu bạn gõ ls ở đây, nó sẽ liệt kê file/thư mục trong Documents.

**📌 Một số trường hợp dùng cd:**

| **Lệnh** | **Ý nghĩa** |
| --- | --- |
| cd | Trở về thư mục chính của bạn (/home/lebao) |
| cd /etc | Đi đến thư mục /etc (tuyệt đối) |
| cd Documents | Vào thư mục Documents (tương đối, nằm trong thư mục hiện tại) |

## **Nội dung 3: ký hiệu ./ và ../ là gì?**

Đây là hai ký hiệu **rất quan trọng** khi bạn làm việc trong dòng lệnh.

**✅ ./ → Thư mục hiện tại**

* Dấu . nghĩa là **"chính là thư mục hiện tại"** bạn đang đứng.

**📌 Ví dụ:**

Giả sử bạn đang đứng ở thư mục /home/lebao, và bạn có một file chương trình run.sh.

Bạn có thể chạy nó bằng:

./run.sh

→ Nghĩa là: “Chạy file run.sh nằm **ngay tại thư mục hiện tại**.”

**✅ ../ → Thư mục cha (thư mục phía trên)**

* Dấu .. nghĩa là **“thư mục cha”** – tức là thư mục chứa thư mục hiện tại.

**📦 Ví dụ:**

Giả sử bạn đang ở thư mục: /home/lebao/Documents

Bạn muốn quay lại thư mục /home/lebao (thư mục cha), bạn gõ:

cd ..

Nếu bạn lại gõ thêm:

cd ..

Thì bạn sẽ lên thêm 1 cấp nữa → đến /home

**📌 Kết hợp:**

| **Lệnh** | **Ý nghĩa** |
| --- | --- |
| ./ | Thư mục hiện tại |
| ../ | Thư mục cha |
| ../../ | Lên 2 cấp thư mục |
| cd ../Downloads | Di chuyển đến thư mục Downloads nằm ở cùng cấp với thư mục hiện tại |

Bạn đã gõ:

**lebao@nammoadidaphat:~$ ls**

**lebao@nammoadidaphat:~$**

💡 Và thấy **không có gì hiện ra** sau lệnh ls.

## **Tại sao khi đang ở thư mục /home của user, khi dùng câu lệnh ls không hiển thị gì?**

Điều này xảy ra khi:

**✅ Thư mục hiện tại KHÔNG có file hay thư mục con nào bên trong.**

Cụ thể hơn:

* Bạn đang ở trong thư mục ~, tức là **thư mục chính của người dùng lebao**, tương đương với /home/lebao
* Khi bạn gõ ls, Ubuntu liệt kê những **file và thư mục con nằm trong đó**
* Nếu **không có gì** bên trong thì lệnh ls **sẽ không hiển thị gì cả** – đây là hành vi **bình thường**

👉 Không phải lỗi, chỉ đơn giản là thư mục đang **trống**

**✅ Cách kiểm tra rõ ràng hơn**

## **Kiểm tra xem mình đang ở thư mục nào**

**pwd**

Kết quả sẽ là:

**/home/lebao**

→ Bạn đang ở thư mục chính của user lebao

**pwd: Là viết tắt của print working directory → Hiển thị đường dẫn thư mục hiện tại.**

**✅ Mục đích:**

Lệnh pwd giúp bạn biết **"hiện tại mình đang đứng ở đâu trong hệ thống file (filesystem)"**.

**⚠️ Điều cần chú ý:**

Bạn nhận được kết quả:

**home/lebao**

Điều này **thiếu dấu / ở đầu**, nên không hoàn toàn đúng.

**✅ Kết quả đúng phải là:**

**/home/lebao**

Dấu / ở đầu là cực kỳ quan trọng. Nó thể hiện **thư mục gốc (root)** – nơi bắt đầu của toàn bộ hệ thống Linux.

**🧱 Giải thích kỹ hơn:**

**🔸 Trong hệ thống Linux:**

* **/** là thư mục gốc (root)
* **/home** là thư mục chứa các thư mục người dùng
* **/home/lebao** là thư mục riêng của bạn (user lebao)

**✅ Vì sao bạn thấy home/lebao mà không có /?**

👉 Có thể là **Ubuntu đang bị lỗi cấu hình nhỏ**, hoặc **một số môi trường shell** (như trong máy ảo hoặc container) bị cấu hình thiếu phần /.

📌 Nhưng **về mặt logic**, thư mục đó thực sự là:

/home/lebao

Bạn có thể xác minh bằng cách gõ thêm lệnh:

cd /

ls

Sẽ thấy các thư mục như:

bin boot dev etc home lib media opt proc root run sbin tmp usr var

→ Tức là /home là một thư mục con của /, chứ không phải home là thư mục hiện tại không có gốc.

**✅ Bạn nên nhớ:**

| **Đường dẫn** | **Ý nghĩa** |
| --- | --- |
| /home/lebao | Đường dẫn tuyệt đối – đầy đủ từ gốc / |
| home/lebao | Đường dẫn tương đối – thiếu phần gốc, nên sai trong ngữ cảnh pwd |

**📌 Tóm gọn lại:**

* ✅ Lệnh pwd giúp bạn biết “mình đang ở thư mục nào”.
* ⚠️ Kết quả đúng phải bắt đầu bằng dấu / → ví dụ /home/lebao
* ❌ Nếu thấy home/lebao là **bị thiếu dấu / đầu**, nhưng vẫn đang ở thư mục đúng.
* 🧠 Trong Linux, mọi thứ đều bắt đầu từ thư mục gốc /.

## **Hiển thị chi tiết, kể cả file ẩn**

**ls -la**

🔍 Lệnh này sẽ hiển thị cả các **file ẩn** (tên bắt đầu bằng dấu .), ví dụ .bashrc, .profile, v.v.

Kết quả có thể giống như:

total 20

drwxr-xr-x 2 lebao lebao 4096 Oct 15 14:00 .

drwxr-xr-x 3 root root 4096 Oct 15 13:59 ..

-rw-r--r-- 1 lebao lebao 220 Oct 15 13:59 .bash\_logout

-rw-r--r-- 1 lebao lebao 3771 Oct 15 13:59 .bashrc

-rw-r--r-- 1 lebao lebao 807 Oct 15 13:59 .profile

➡️ Những file bắt đầu bằng . là **file ẩn**, thường dùng để cấu hình hệ thống.

**✅ Cách thử thêm – tạo một file để thấy ls có hoạt động không**

Bạn có thể thử tạo một file đơn giản bằng lệnh:

**touch hello.txt**

Sau đó gõ lại:

**ls**

Kết quả:

**hello.txt**

→ Giờ bạn sẽ thấy ls hiện ra file mới bạn vừa tạo.

**🧠 Tóm lại:**

| **Hiện tượng** | **Nguyên nhân** |
| --- | --- |
| Gõ ls không hiện gì | Thư mục hiện tại đang trống, không có file hoặc thư mục con |
| ls -la hiện ra nhiều file .bashrc, .profile,... | Đó là file ẩn, thường dùng cho cấu hình |
| Sau khi tạo file bằng touch, ls mới hiển thị | Đúng! Vì bây giờ thư mục đã có nội dung |

Bạn có thể thử thêm:

**mkdir test\_folder**

**ls**

→ Bạn sẽ thấy thư mục test\_folder hiện ra.

# **TẠO THƯ MỤC VÀ FILE**

Bạn đang nhìn vào một **sơ đồ minh hoạ giả lập** cây thư mục (filesystem) của hệ điều hành Ubuntu Server như thế này:

/

**├── home/**

**│ └── lebao/**

**│ ├── file1.txt**

**│ └── Documents/**

**│ └── resume.doc**

**├── etc/**

**├── var/**

**└── usr/**

Đây là một **mô hình tưởng tượng (giả định)** để giúp bạn **hiểu cách hệ thống thư mục hoạt động**, chứ **không phải máy của bạn đã có sẵn những file đó**.

**🧠 Thực tế trên máy bạn**

* Khi bạn mới **cài đặt Ubuntu Server** (đặc biệt là bản không có giao diện đồ họa), thì:
  + **Thư mục Documents** không có sẵn
  + **Các file như file1.txt, resume.doc** cũng không tồn tại
  + **Thư mục /home/lebao** thường **rỗng** hoặc chỉ có một vài **file ẩn cấu hình** như .bashrc, .profile, v.v.

**✅ Cách kiểm tra chính xác trên máy bạn**

**1. Kiểm tra xem bạn đang ở đâu:**

**pwd**

**/home/lebao**

**2. Xem có gì trong đó:**

**ls**

Nếu kết quả là **trống**, tức là **không có gì**.

Hoặc dùng:

**ls -la**

Để thấy cả các **file ẩn** (bắt đầu bằng dấu chấm .)

**3. Kiểm tra thử có thư mục Documents hay file file1.txt không:**

**ls Documents**

Nếu nó báo lỗi:

ls: cannot access 'Documents': No such file or directory

→ Nghĩa là **chưa có thư mục Documents**

**📌 Tạo thủ công các thư mục & file để giống ví dụ**

Nếu bạn muốn tạo đúng như sơ đồ ví dụ, bạn chỉ cần gõ các lệnh sau:

**cd ~ # Đảm bảo bạn đang ở /home/lebao**

**mkdir Documents # Tạo thư mục Documents**

**touch file1.txt # Tạo file file1.txt**

**touch Documents/resume.doc # Tạo file resume.doc trong thư mục Documents**

Sau đó kiểm tra lại:

**ls**

# → sẽ thấy file1.txt và thư mục Documents

**ls Documents**

# → sẽ thấy resume.doc

# **CÁCH GHI NỘI DUNG TRONG FILE**

**🎯 Mục tiêu: Ghi nội dung vào file file1.txt**

## **Cách 1: Dùng lệnh echo để ghi nhanh một dòng**

**📌 Cú pháp:**

**echo "Nội dung bạn muốn ghi" > file1.txt**

**📌 Giải thích:**

* echo in ra nội dung chuỗi
* > dùng để **ghi vào file** (nếu file chưa có thì tạo mới, nếu có rồi thì **xóa nội dung cũ** rồi ghi mới)
* file1.txt là tên file

**📌 Ví dụ:**

**echo "Nam Mô A Di Đà Phật" > file1.txt**

👉 File file1.txt giờ có nội dung:

Nam Mô A Di Đà Phật

📌 Kiểm tra:

**cat file1.txt**

## **Cách 2: Dùng echo để thêm dòng mới vào file (không xoá nội dung cũ)**

**📌 Cú pháp:**

**echo "Dòng mới" >> file1.txt**

Lưu ý: >> là **nối thêm vào cuối file**

**📌 Ví dụ:**

**echo "Nam Mô Đại Bi Quán Thế Âm Bồ Tát" >> file1.txt**

📌 Kiểm tra lại:

**cat file1.txt**

Kết quả:

Nam Mô A Di Đà Phật

Nam Mô Đại Bi Quán Thế Âm Bồ Tát

## **Cách 3: Dùng trình soạn thảo nano để gõ nội dung trực tiếp**

**📌 Bước 1: Mở file bằng nano**

**nano file1.txt**

Bạn sẽ thấy màn hình chuyển sang chế độ soạn thảo như thế này:

GNU nano 6.x file1.txt

| ← Gõ nội dung tại đây

|

|

^G Help ^O Write Out ^X Exit ...

**📌 Bước 2: Gõ nội dung bạn muốn vào**

Ví dụ:

Nam Mô A Di Đà Phật

Nam Mô Đại Bi Quán Thế Âm Bồ Tát

**📌 Bước 3: Lưu và thoát nano**

* **Nhấn Ctrl + O để lưu file (sẽ hiện tên file, bạn nhấn Enter)**
* **Nhấn Ctrl + X để thoát khỏi nano**

**📌 Kiểm tra lại:**

**cat file1.txt**

## **Cách 4 (nâng cao): Dùng vim (nếu bạn muốn học trình soạn thảo chuyên sâu)**

**📌 Tổng kết so sánh các cách**

| **Cách** | **Ưu điểm** | **Nhược điểm** |
| --- | --- | --- |
| echo "..." > file | Nhanh, đơn giản | Ghi đè, không phù hợp nếu cần viết nhiều dòng |
| echo "..." >> file | Nối thêm dòng dễ | Không sửa nội dung giữa file được |
| nano file | Soạn thảo trực tiếp, dễ dùng | Cần học thao tác lưu, thoát |
| vim file | Mạnh mẽ cho lập trình | Hơi khó với người mới |

**👉 Gợi ý cho bạn: Thực hành**

**cd ~**

**touch file1.txt**

**echo "Đây là dòng đầu tiên" > file1.txt**

**echo "Đây là dòng thứ hai" >> file1.txt**

**nano file1.txt # Gõ thêm dòng thứ ba bằng tay**

**cat file1.txt # Xem lại nội dung**

## **vi là gì?**

* vi là một **trình soạn thảo văn bản trong dòng lệnh**, có từ rất lâu và rất mạnh mẽ
* Bạn dùng vi để **mở, đọc, sửa** nội dung của file
* vi chạy **ngay trong terminal**, không cần giao diện đồ họa

**🟩 "Syntax of the command: vi <file\_name>"**

👉 Nghĩa là:

**"Cú pháp của lệnh: vi + tên file"**

Ví dụ:

**vi file1.txt**

**🧪 Khi bạn gõ vi file1.txt, chuyện gì sẽ xảy ra?**

**✅ Nếu file đã tồn tại:**

* vi sẽ **mở file ra**, và bạn có thể đọc và sửa nội dung bên trong

**✅ Nếu file chưa tồn tại:**

* vi sẽ **tạo file mới với tên đó**, cho bạn **soạn nội dung từ đầu**

**🧑‍🏫 Hướng dẫn sử dụng vi cơ bản (cho người mới)**

vi hơi khó với người mới vì nó có nhiều **chế độ điều khiển**, nhưng mình sẽ hướng dẫn từng bước rõ ràng.

**▶️ Bước 1: Mở file bằng vi**

**vi file1.txt**

**▶️ Bước 2: Vào chế độ soạn thảo**

* **Sau khi mở file, bạn không thể gõ gì ngay**
* **Bạn cần bấm phím i (viết thường) để vào chế độ INSERT**

➡️ Bạn sẽ thấy ở cuối màn hình dòng chữ:

**-- INSERT --**

**▶️ Bước 3: Gõ nội dung bạn muốn**

Ví dụ:

Nam Mô A Di Đà Phật

Học Ubuntu từ con số 0

**▶️ Bước 4: Lưu và thoát vi**

**⬅️ Thoát khỏi chế độ insert:**

* Nhấn **Esc** (phím góc trên bên trái)

**💾 Lưu nội dung và thoát:**

* Gõ lệnh:

**:wq**

(Sau khi nhấn Esc, gõ dấu : rồi wq – nghĩa là **write & quit**)

* Sau đó **nhấn Enter**

📌 Giải thích:

* : để bắt đầu lệnh
* w = write (lưu)
* q = quit (thoát)

**❗ Các lệnh nhanh khi dùng vi**

| **Phím / Lệnh** | **Ý nghĩa** |
| --- | --- |
| i | Vào chế độ chèn nội dung (INSERT) |
| Esc | Thoát khỏi chế độ INSERT |
| :w | Lưu file |
| :q | Thoát vi |
| :wq | Lưu và thoát |
| :q! | Thoát không lưu |

**📌 Gợi ý cho bạn: Thực hành ngay**

**cd ~**

**vi file1.txt**

**# Bấm i để nhập:**

**# Nam Mô A Di Đà Phật**

**# Học Ubuntu Server từ con số 0**

**# Nhấn Esc**

**# Gõ :wq rồi Enter**

**cat file1.txt**

**✅ 1. Cách của thầy bạn: Dùng vi (hoặc vim)**

**🔸 Tính chất:**

|  | **Nội dung** |
| --- | --- |
| Công cụ | vi là trình soạn thảo dòng lệnh mạnh mẽ, gắn liền với Linux/Unix |
| Đa năng | Có thể **soạn thảo, sửa dòng, tìm kiếm, xoá, copy,...** |
| Chế độ làm việc | Có nhiều chế độ: lệnh, soạn thảo, dòng lệnh |
| Hơi khó với người mới | Cần nhớ nhiều thao tác bàn phím (nhấn i, Esc, :wq,...) |
| Rất mạnh cho lập trình | Được dùng trong server, lập trình, cấu hình hệ thống chuyên sâu |

**🔸 Thích hợp khi nào?**

* Bạn muốn **chỉnh sửa nội dung file một cách chuyên nghiệp**
* Bạn cần làm việc trong môi trường **server không có giao diện đồ họa**
* Bạn làm việc với **file cấu hình, file code, file shell script,...**

**✅ 2. Các cách mình dạy bạn: echo, cat, nano**

**A. 🟢 echo và >> / > để ghi nhanh**

|  | **Nội dung** |
| --- | --- |
| Cực kỳ đơn giản | Ghi nội dung vào file bằng một dòng lệnh |
| Ghi 1 dòng/lần | Mỗi lệnh ghi 1 dòng mới |
| > là ghi đè | >> là nối thêm |
| Không sửa nội dung cũ | Chỉ thêm hoặc ghi đè toàn bộ |

📌 **Thích hợp khi:**

* Ghi nhanh 1–2 dòng vào file
* Tạo file test, demo, log đơn giản

**B. 🟢 nano – dễ dùng hơn vi**

|  | **Nội dung** |
| --- | --- |
| Giao diện đơn giản hơn | Có thanh hướng dẫn phím ở dưới màn hình |
| Dễ học hơn vi | Dùng mũi tên, gõ trực tiếp, lưu bằng Ctrl + O, thoát bằng Ctrl + X |
| Không có nhiều chế độ như vi | Đơn giản, dễ hiểu |

📌 **Thích hợp khi:**

* Bạn mới học dùng Ubuntu
* Muốn chỉnh sửa nhanh nội dung file mà không rối

**C. 🟢 cat – chỉ để xem file, không chỉnh sửa**

**cat file.txt**

📌 Dùng để **hiển thị nội dung** file ra màn hình – **không chỉnh sửa được**

**🧠 So sánh tổng quan**

| **Cách** | **Dùng để** | **Dễ/Khó** | **Có thể sửa nội dung cũ?** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| vi | Đọc & chỉnh sửa | Hơi khó với người mới | ✅ Có | Rất mạnh mẽ, chuyên nghiệp |
| nano | Đọc & chỉnh sửa | Dễ hơn vi | ✅ Có | Phù hợp người mới |
| echo > | Ghi nội dung mới | Rất dễ | ❌ (Ghi đè) | Ghi 1 dòng mới |
| echo >> | Ghi nối thêm | Rất dễ | ❌ (Không sửa cũ) | Nối dòng vào cuối file |
| cat | Xem nội dung | Dễ | ❌ | Chỉ xem, không sửa |

# **THOÁT TÀI KHOẢN, KHỞI ĐỘNG LẠI, TẮT MÁY CHỦ**

Khi làm việc với **Ubuntu Server**, bạn đôi khi cần:

| **Mục tiêu** | **Lý do** |
| --- | --- |
| Đăng xuất khỏi tài khoản người dùng hiện tại | Vì người dùng đã xong việc |
| Khởi động lại máy chủ (restart) | Sau khi cấu hình hệ thống, cập nhật phần mềm,... |
| Tắt máy chủ (shutdown) | Khi bảo trì, tiết kiệm điện, hoặc không dùng nữa |

Giờ mình sẽ đi **từng dòng lệnh một** mà bạn đưa ra:

## **1. Thoát người dùng hiện tại bằng dòng lệnh**

**pkill -KILL -u <user\_name>**

**✅ Giải thích từng phần:**

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| pkill | Lệnh dùng để “giết” (kill) một tiến trình (process) đang chạy |
| -KILL | Tùy chọn: gửi tín hiệu **KILL** để buộc dừng tiến trình |
| -u <user\_name> | Chỉ định tất cả tiến trình của người dùng <user\_name> sẽ bị kết thúc |

👉 **Khi bạn chạy lệnh này, Ubuntu sẽ buộc đăng xuất toàn bộ phiên làm việc của người dùng đó.**

**📌 Ví dụ thực tế:**

Giả sử bạn đang đăng nhập với tên người dùng là lebao, bạn có thể chạy:

**pkill -KILL -u lebao**

→ Lệnh này **buộc người dùng lebao đăng xuất khỏi hệ thống**, giống như "đá văng ra ngoài".

⚠️ **Lưu ý quan trọng:**

* Nếu bạn dùng lệnh này để **đăng xuất chính bạn**, bạn sẽ bị "văng" ra ngay.
* Dùng lệnh này cẩn thận – thường chỉ dùng khi bạn là **admin** và muốn đăng xuất người khác (ví dụ trong môi trường nhiều người dùng)

## **2. Khởi động lại server**

**sudo reboot**

**✅ Giải thích:**

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| sudo | Cho phép chạy lệnh với quyền **quản trị hệ thống (root)** |
| reboot | Lệnh để **khởi động lại hệ thống ngay lập tức** |

**👉 Điều gì xảy ra?**

Khi bạn chạy:

**sudo reboot**

Hệ thống sẽ:

* **Thoát tất cả người dùng**
* **Tắt tất cả dịch vụ**
* **Khởi động lại máy**

⏱️ Thường mất vài chục giây đến 1–2 phút tùy vào cấu hình máy.

## **3. Tắt máy theo thời gian chỉ định (shutdown)**

**sudo shutdown <number>**

**✅ Giải thích:**

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| sudo | Chạy lệnh với quyền quản trị |
| shutdown | Lệnh để tắt máy (shutdown) |
| <number> | **Số phút** sau khi chạy lệnh, máy sẽ được tắt |

**📌 Ví dụ:**

**sudo shutdown 5**

→ Máy sẽ **tắt sau 5 phút** kể từ lúc bạn gõ lệnh.

Hệ thống sẽ thông báo:

**Shutdown scheduled for Thu 2025-10-15 22:45:00 UTC, use 'shutdown -c' to cancel.**

## **4. Tắt máy ngay lập tức:**

Bạn cũng có thể **tắt máy ngay** bằng:

**sudo shutdown now**

👉 Máy sẽ **tắt ngay lập tức**.

**🎯 Tổng kết 3 lệnh chính:**

| **Mục tiêu** | **Lệnh** | **Ý nghĩa** |
| --- | --- | --- |
| Đăng xuất người dùng | pkill -KILL -u lebao | Buộc user lebao đăng xuất |
| Khởi động lại | sudo reboot | Tắt máy → khởi động lại ngay |
| Tắt máy sau X phút | sudo shutdown 5 | Máy sẽ tắt sau 5 phút |
| Tắt máy ngay | sudo shutdown now | Máy tắt lập tức |

Bạn hãy thử từng lệnh sau trong máy ảo Ubuntu:

**whoami # Kiểm tra bạn là user nào**

**sudo shutdown 2 # Đặt máy tắt sau 2 phút**

**sudo shutdown -c # Hủy lệnh shutdown**

**sudo reboot # Khởi động lại**

**Sau đó thử:**

**sudo shutdown now # Tắt máy ngay (sẽ làm bạn bị thoát khỏi VMware)**

# **QUẢN LÝ MẠNG TRONG UBUNTU SERVER**

Đây là phần **cực kỳ quan trọng** khi bạn làm việc với máy chủ (server), vì **máy chủ mà không kết nối được mạng thì coi như... vô dụng**.

**🔹 KHÁI NIỆM CƠ BẢN**

| **Khái niệm** | **Ý nghĩa** |
| --- | --- |
| **IP Address** | Là "địa chỉ nhà" của máy bạn trong mạng. VD: 192.168.1.100 |
| **Default Gateway** | Là "cổng ra" của mạng, nơi máy bạn sẽ gửi dữ liệu ra internet |
| **DNS Server** | Là máy giúp "dịch tên miền thành IP", VD: google.com thành 142.250.72.14 |
| **Netplan** | Là công cụ trong Ubuntu dùng để cấu hình mạng (IP, DNS, gateway...) qua file YAML |

## **1. Xem địa chỉ IP hiện tại của máy**

**❓ Mục tiêu:**

Xem địa chỉ IP mà máy Ubuntu của bạn đang dùng (nếu bạn dùng DHCP thì IP sẽ được tự gán).

**📌 Câu lệnh:**

**ip a**

hoặc

**ip addr show**

**📘 Giải thích:**

* ip là lệnh quản lý mạng trong Linux
* a là viết tắt của address

**📌 Ví dụ kết quả:**

**2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 ...**

**inet 192.168.1.100/24 brd 192.168.1.255 scope global dynamic enp0s3**

👉 Bạn để ý dòng bắt đầu bằng inet – đó là **địa chỉ IP** của máy bạn (192.168.1.100).

## **2. Xem địa chỉ gateway mặc định**

**❓ Mục tiêu:**

Biết "cổng ra" của mạng là IP nào (thường là router Wi-Fi hoặc modem).

**📌 Câu lệnh:**

**ip route**

**📘 Giải thích:**

Lệnh ip route sẽ hiển thị bảng định tuyến (route table), trong đó bạn sẽ thấy:

default via 192.168.1.1 dev enp0s3 ...

👉 192.168.1.1 chính là **default gateway**.

## **3. Xem DNS server mà hệ thống đang dùng**

**❓ Mục tiêu:**

Xem máy bạn đang dùng DNS nào để phân giải tên miền (như google.com thành IP).

**📌 Câu lệnh:**

**cat /etc/resolv.conf**

**📘 Giải thích:**

File /etc/resolv.conf chứa danh sách DNS server đang được sử dụng. Ví dụ:

**nameserver 8.8.8.8**

**nameserver 1.1.1.1**

👉 Đây là các DNS server của Google và Cloudflare – máy bạn sẽ hỏi chúng để tìm địa chỉ IP của các website.

## **4. Cấu hình địa chỉ IP tĩnh (manual / static IP)**

**❓ Mục tiêu:**

Không dùng IP động từ router nữa, mà **gán cố định một địa chỉ IP cho máy** để dễ kiểm soát (rất quan trọng khi làm server).

**📁 File cần chỉnh sửa:**

**/etc/netplan/50-cloud-init.yaml**

Đây là file cấu hình mạng của Ubuntu Server (dùng công cụ Netplan)

**✅ Ví dụ nội dung file cấu hình:**

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**gateway4: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

**📘 Giải thích từng dòng:**

| **Dòng** | **Ý nghĩa** |
| --- | --- |
| network: | Bắt đầu cấu hình mạng |
| version: 2 | Dùng version 2 của Netplan |
| ethernets: | Định nghĩa các card mạng |
| enp0s3: | Tên card mạng (tùy từng máy) – bạn kiểm tra bằng ip a |
| dhcp4: no | Không dùng DHCP nữa (tức là IP tĩnh) |
| addresses: [192.168.1.100/24] | Đặt IP là 192.168.1.100, subnet mask là /24 (tương đương 255.255.255.0) |
| gateway4: 192.168.1.1 | Cổng ra mạng (thường là modem/router) |
| nameservers: addresses: [...] | Danh sách DNS server |

**🔧 Cách sửa file:**

1. Dùng lệnh mở file bằng sudo:

**sudo nano /etc/netplan/50-cloud-init.yaml**

1. Dán nội dung bạn muốn (như ví dụ trên)
2. Nhấn Ctrl + O để lưu, Enter, rồi Ctrl + X để thoát

## **5. Áp dụng cấu hình mạng mới**

**📌 Câu lệnh:**

**sudo netplan apply**

👉 Lệnh này sẽ áp dụng các thay đổi trong file yaml để cấu hình mạng **có hiệu lực ngay lập tức**.

## **6. Kiểm tra kết nối internet**

**📌 Câu lệnh kiểm tra:**

**ping 8.8.8.8**

→ Nếu có trả lời (Reply from...) nghĩa là **máy đã ra được internet**

**ping google.com**

→ Nếu lệnh này hoạt động, nghĩa là **DNS hoạt động tốt**

📌 Nhấn Ctrl + C để dừng ping.

**🧠 Ghi nhớ quan trọng cho người mới:**

| **Việc** | **Ghi nhớ** |
| --- | --- |
| Xem IP | ip a |
| Xem gateway | ip route |
| Xem DNS | cat /etc/resolv.conf |
| Sửa IP tĩnh | Mở /etc/netplan/50-cloud-init.yaml |
| Áp dụng cấu hình | sudo netplan apply |
| Kiểm tra mạng | ping |

# **CẤU HÌNH IP TĨNH**

Rất tốt, bạn đang làm **rất đúng quy trình cấu hình IP tĩnh** cho Ubuntu Server. Và bạn đã gặp phải một cảnh báo khi dùng lệnh sudo netplan apply.

**🧠 Mục tiêu của bạn:**

Bạn đã dùng sudo nano /etc/netplan/50-cloud-init.yaml để cấu hình mạng thủ công như sau:

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**gateway4: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

=> Ý định của bạn:

* **Cấu hình IP tĩnh: 192.168.1.100**
* **Gateway là: 192.168.1.1**
* **DNS là: 8.8.8.8 và 1.1.1.1**

=> Mọi thứ là đúng **về logic**, máy vẫn **có thể dùng bình thường**.

**❗Nhưng sau khi chạy sudo netplan apply, bạn thấy cảnh báo sau:**

**WARNING \*\*: gateway4 has been deprecated, use default routes instead.**

**See the 'Default routes' section of the documentation for more details**.

CẢNH BÁO \*\*: gateway4 đã không còn được sử dụng nữa, hãy sử dụng các tuyến mặc định thay thế.

Xem phần 'Tuyến mặc định' trong tài liệu để biết thêm chi tiết.

## **1. Từ khóa "WARNING"**

* WARNING = Cảnh báo, chứ **không phải lỗi**.
* Hệ thống chỉ đang **nhắc nhở bạn rằng**: "Tôi vẫn chạy được, nhưng bạn nên cập nhật cách viết mới".

**🧠 Tưởng tượng cho dễ hiểu:**

Giống như giáo viên nhắc bạn: “Cách bạn làm vẫn đúng, nhưng có phương pháp mới tốt hơn, bạn nên học dùng sau này”.

## **2. gateway4 has been deprecated**

* **"gateway4" là một từ khóa cũ** trong cú pháp của Netplan.
* **"deprecated" nghĩa là "đã bị khuyến cáo ngừng dùng"** – vì đã có cách viết hiện đại hơn.

🔸 Ubuntu vẫn hiểu từ gateway4, vẫn hoạt động tốt, nhưng:

"Trong tương lai, Netplan có thể bỏ hỗ trợ gateway4, bạn nên đổi sang cách viết mới để an toàn."

## **3. Câu: “Use default routes instead”**

Nghĩa là:

Hãy **sử dụng cú pháp “default routes” mới** để khai báo Gateway thay vì dùng gateway4.

## **4. CÁCH VIẾT MỚI (theo tài liệu Netplan hiện đại)**

Nếu bạn muốn cập nhật cấu hình cho “sạch”, không còn cảnh báo, thì thay vì dùng:

**gateway4: 192.168.1.1**

→ Bạn nên viết:

**routes:**

**- to: default**

**via: 192.168.1.1**

**✅ File YAML hoàn chỉnh theo chuẩn mới:**

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

**routes:**

**- to: default**

**via: 192.168.1.1**

✅ Ưu điểm:

* Không còn cảnh báo “deprecated”
* Chuẩn Netplan mới nhất
* Tương lai không sợ lệnh bị loại bỏ

**🔧 Làm gì tiếp theo?**

| **Việc bạn nên làm** | **Gợi ý** |
| --- | --- |
| ❌ Không cần sửa ngay nếu máy vẫn chạy mạng tốt | Vì gateway4 **vẫn hoạt động được** |
| ✅ Nên sửa sang cú pháp mới để sạch sẽ, tránh cảnh báo | Nếu bạn muốn học chuẩn, theo tài liệu chính thức |

**🧪 Cách kiểm tra lại:**

Sau khi sửa file /etc/netplan/50-cloud-init.yaml theo cú pháp mới như trên, bạn chạy lại:

**sudo netplan apply**

Nếu không thấy dòng cảnh báo nữa → bạn đã làm đúng!

## **5. Mục đích thay đổi file YAML trong Netplan**

Là để **cấu hình mạng (network)** cho Ubuntu Server – cụ thể là **cài đặt địa chỉ IP, gateway, DNS,...** để máy **kết nối được vào mạng LAN, Internet, hoặc giao tiếp với máy khác.**

Nếu bạn KHÔNG cấu hình mạng đúng thì **máy server không ra được Internet**, không kết nối được, cũng không thể làm gì khác (vì server sống là nhờ có mạng).

**1. Máy tính/server muốn kết nối mạng cần gì?**

Trước hết, để một máy **kết nối vào mạng**, nó cần được cung cấp những thông tin cơ bản sau:

| **Thông tin** | **Vai trò** |
| --- | --- |
| **IP Address** | Là địa chỉ duy nhất của máy trong mạng (như địa chỉ nhà của bạn) |
| **Gateway** | Là “cổng ra” để máy gửi dữ liệu ra ngoài mạng LAN, đến Internet |
| **DNS Server** | Là nơi máy hỏi để chuyển google.com thành địa chỉ IP thật (như 142.250.72.14) |

**2. Ubuntu Server dùng Netplan để cấu hình mạng**

Kể từ Ubuntu 17.10 trở đi, Ubuntu đã chuyển sang dùng **Netplan** để quản lý mạng.

Netplan dùng các **file cấu hình có đuôi .yaml** nằm trong thư mục:

**/etc/netplan/**

Ví dụ:

**/etc/netplan/50-cloud-init.yaml**

Đây là nơi bạn **khai báo mạng** cho máy server:

* Gán IP tĩnh hay IP động (qua DHCP)?
* Cấu hình DNS nào?
* Gateway là gì?

💡 Tóm lại:

File YAML trong Netplan chính là “hồ sơ” định nghĩa cách mà **máy tính này kết nối mạng như thế nào**.

**3. Nội dung của file YAML thể hiện điều gì?**

Giả sử bạn có file như sau:

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**gateway4: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

**👉 Nghĩa là:**

* dhcp4: no → Máy **không lấy IP tự động từ modem/router**, mà bạn sẽ **tự cấu hình IP thủ công**
* addresses: [192.168.1.100/24] → Gán IP là 192.168.1.100, subnet mask /24 (tức là 255.255.255.0)
* gateway4: 192.168.1.1 → Đây là cổng để ra mạng ngoài (thường là địa chỉ của router)
* nameservers: addresses: → Cấu hình máy chủ DNS

👉 Nếu bạn KHÔNG viết file YAML này, Ubuntu sẽ **không biết dùng IP nào, DNS nào, Gateway nào**, → máy sẽ không vào mạng được!

**4. Sau khi sửa file YAML thì làm gì?**

Bạn chạy lệnh:

**sudo netplan apply**

→ Lệnh này giúp **áp dụng các cấu hình mạng** bạn vừa viết trong file YAML.  
Giống như bạn nhấn "Save and Apply" sau khi cài đặt Wi-Fi vậy.

**🧪 Nếu không sửa file YAML thì sao?**

| **Tình huống** | **Kết quả** |
| --- | --- |
| Không có file YAML đúng | Máy **không có địa chỉ IP** → không vào được mạng |
| Cấu hình sai | Máy có thể không giao tiếp được với các máy khác |
| Không có gateway hoặc DNS | Không truy cập được Internet hoặc không resolve được tên miền |

## **6. Nhưng tại sao phải dùng IP tĩnh? Sao không để tự động (DHCP) luôn?**

Vì **server cần IP cố định** để người khác truy cập!

Ví dụ:

* Nếu server của bạn là web server, thì người khác sẽ truy cập bằng IP: 192.168.1.100
* Nếu bạn để IP động (DHCP), IP có thể thay đổi mỗi lần khởi động → không ai kết nối được nữa

**🎯 KẾT LUẬN: Tóm lại cực ngắn cho dễ nhớ**

| **Câu hỏi** | **Trả lời** |
| --- | --- |
| **Vì sao phải sửa file YAML?** | Vì file YAML là nơi cấu hình mạng cho Ubuntu Server |
| **Tác dụng?** | Giúp máy có IP, DNS, gateway để kết nối mạng và Internet |
| **Lệnh áp dụng?** | sudo netplan apply |
| **Có bắt buộc không?** | Có! Nếu không có file đúng, server sẽ không vào mạng được |
| **Sửa sai có sao không?** | Có thể mất kết nối mạng. Nhưng có thể sửa lại dễ dàng |

**📌 Gợi ý cho bạn thực hành (trên máy ảo)**

1. Mở file YAML:

**sudo nano /etc/netplan/50-cloud-init.yaml**

1. Dán nội dung như sau (cập nhật đúng card mạng):

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

**routes:**

**- to: default**

**via: 192.168.1.1**

1. Lưu lại, rồi chạy:

**sudo netplan apply**

1. Kiểm tra IP bằng:

**ip a**

1. Kiểm tra mạng:

**ping 8.8.8.8**

**ping google.com**

# **MỤC ĐÍCH CỦA VIỆC DÙNG PING 8.8.8.8 VÀ PING GOOGLE.COM**

Đây là **bước kiểm tra mạng** sau khi bạn đã cấu hình IP, gateway và DNS trong file YAML.

## **1. ping 8.8.8.8 là để kiểm tra**

Máy bạn có **ra được Internet** chưa – thông qua **địa chỉ IP trực tiếp**.

8.8.8.8 là **DNS server của Google**, rất phổ biến và luôn "sống".

✅ Nếu **ping thành công**, thì nghĩa là:

* Máy bạn đã **giao tiếp được với mạng bên ngoài**
* **Gateway hoạt động tốt** (bạn đi ra khỏi mạng LAN được)

## **2. ping google.com là để kiểm tra**

Máy bạn có thể **chuyển tên miền thành địa chỉ IP được không**  
→ Đây là **kiểm tra DNS**

✅ Nếu ping google.com thành công:

* DNS bạn cấu hình đang **hoạt động tốt**
* Máy có thể **phân giải tên miền**

**😬 Nhưng bạn gặp lỗi như sau:**

**❌ ping 8.8.8.8 → báo:**

ping: connect: Network is unreachable

ping: kết nối: Mạng không thể truy cập được

**❌ ping google.com → báo:**

ping: google.com: Temporary failure in name resolution

ping: google.com: Lỗi tạm thời trong việc phân giải tên

**❌ 1. ping: connect: Network is unreachable**

Nghĩa: **Máy bạn chưa có kết nối mạng** – không thể ra ngoài được, vì không có đường đi (route) nào cả.

**🧠 Hiểu đơn giản:**

Giống như bạn muốn đi đến nhà bạn, nhưng:

* Bạn không có địa chỉ cụ thể
* Không có đường đi (gateway)
* Không biết đi lối nào

👉 Máy bạn **chưa cấu hình gateway đúng**, hoặc Netplan chưa hiểu rõ cách ra mạng.

**❌ 2. ping: google.com: Temporary failure in name resolution**

Nghĩa là: Máy **không thể dịch tên miền google.com ra địa chỉ IP**, vì **DNS không hoạt động được**

**Nguyên nhân phổ biến**:

* Chưa kết nối mạng (đương nhiên DNS cũng không hoạt động)
* Hoặc sai cấu hình DNS

**🔍 Tóm lại, hai lỗi này thường do 3 lỗi chính sau:**

| **Nguyên nhân** | **Mô tả** |
| --- | --- |
| ❌ Cấu hình gateway sai hoặc không đúng cú pháp | Máy không biết đường ra Internet |
| ❌ Cấu hình DNS sai | Không phân giải được tên miền |
| ❌ Dùng sai tên card mạng (enp0s3, eth0,...) | Netplan không áp dụng đúng cấu hình vì dùng sai tên card |

## **3. Cách kiểm tra – sửa lỗi**

**🧪 Bước 1: Kiểm tra tên card mạng thật sự là gì**

Chạy lệnh:

**ip link**

Bạn sẽ thấy danh sách card mạng, ví dụ:

1: lo: ...

2: enp0s3: ...

Nếu card bạn dùng trong file YAML không đúng tên thật → cấu hình **không áp dụng**.

**🧪 Bước 2: Mở lại file YAML để kiểm tra**

Chạy lệnh:

**sudo nano /etc/netplan/50-cloud-init.yaml**

Và kiểm tra xem:

* Tên card mạng đúng chưa (enp0s3, eth0, ens33,... phải đúng với tên ở bước 1)
* Có gateway chưa? Đúng cú pháp chưa?
* Có DNS chưa?

**✅ File YAML mẫu ĐÚNG CHUẨN (Dùng cú pháp mới, tránh cảnh báo):**

Giả sử card mạng tên là enp0s3, bạn cấu hình như sau:

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**routes:**

**- to: default**

**via: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

📌 Giải thích:

* addresses: IP của bạn
* routes.to: default: là chỉ định “đường mặc định để đi ra Internet”
* via: là địa chỉ gateway (thường là modem hoặc router)
* nameservers: máy DNS để phân giải tên miền

**🧪 Bước 3: Áp dụng lại**

**sudo netplan apply**

**🧪 Bước 4: Kiểm tra IP**

**ip a**

Bạn phải thấy dòng có IP 192.168.1.100 gán cho enp0s3

**🧪 Bước 5: Kiểm tra lại kết nối**

**ping 8.8.8.8**

Nếu thấy dòng như:

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=119 time=10.2 ms

→ ✅ Máy bạn đã ra được Internet

Tiếp theo kiểm tra DNS:

**ping google.com**

Nếu ping được → ✅ DNS hoạt động

**📌 Tóm Tắt Dễ Nhớ**

| **Việc bạn làm** | **Mục đích** | **Có lỗi không?** |
| --- | --- | --- |
| ping 8.8.8.8 | Kiểm tra có ra Internet không | Bị lỗi: máy không ra được mạng |
| ping google.com | Kiểm tra DNS có hoạt động không | Bị lỗi: không phân giải được tên miền |
| Nguyên nhân chính | Do cấu hình gateway sai hoặc card mạng không đúng | Đúng |
| Cách sửa | Sửa lại file YAML đúng, apply lại bằng sudo netplan apply | Có thể sửa dễ dàng |

**✅ Tóm tắt kết quả của bạn:**

**2: ens33: ...**

👉 **Tên card mạng thật của bạn là ens33** (chứ **không phải enp0s3** như bạn đã viết trong file YAML trước đó).

**⚠️ Đây chính là nguyên nhân chính khiến:**

* Cấu hình mạng không được áp dụng đúng cách
* ping bị lỗi network is unreachable

**🧠 Vì sao vậy?**

Trong file YAML, bạn đã viết:

**ethernets:**

**enp0s3: ← ❌ Sai tên card mạng**

➡️ Netplan đọc file này, không thấy card nào tên là enp0s3, nên nó **không cấu hình mạng gì cả** → máy không có IP, không có đường ra ngoài.

## **4. Cách sửa lại để mạng hoạt động**

**✅ Bước 1: Mở lại file cấu hình netplan**

**sudo nano /etc/netplan/50-cloud-init.yaml**

**✅ Bước 2: Sửa thành nội dung sau (dùng đúng tên card mạng ens33):**

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**routes:**

**- to: default**

**via: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

📌 Giải thích từng dòng:

* ens33: là tên card mạng thật của bạn
* dhcp4: no: tắt cấp phát IP tự động
* addresses: gán IP tĩnh cho máy là 192.168.1.100, subnet /24 = 255.255.255.0
* routes: định nghĩa gateway ra mạng là 192.168.1.1
* nameservers: khai báo DNS để phân giải tên miền

(Lưu ý: bạn cần đảm bảo 192.168.1.1 là địa chỉ gateway thật – thường là IP của router)

**✅ Bước 3: Lưu và thoát**

* **Nhấn Ctrl + O → Enter để lưu**
* **Nhấn Ctrl + X để thoát nano**

**✅ Bước 4: Áp dụng lại cấu hình:**

**sudo netplan apply**

**✅ Bước 5: Kiểm tra lại IP:**

**ip a**

→ Bạn sẽ thấy card ens33 đã có IP 192.168.1.100

**✅ Bước 6: Kiểm tra kết nối:**

1. **Kiểm tra ra Internet**:

**ping 8.8.8.8**

1. **Kiểm tra DNS**:

**ping google.com**

**🎉 Nếu thấy kết quả như sau là bạn đã thành công:**

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=119 time=10.2 ms

hoặc:

64 bytes from google.com: ...

✅ Nghĩa là:

* Máy đã ra được Internet
* DNS đã hoạt động bình thường

**📌 Tóm lại**

| **Việc bạn làm** | **Trạng thái** | **Lý do** |
| --- | --- | --- |
| Cấu hình netplan | Sai tên card mạng | enp0s3 → ❌ không tồn tại |
| Card mạng đúng | ens33 | ✔️ Bạn cần sửa file YAML theo tên này |
| Gateway và DNS | Đã khai báo đúng | ✔️ chỉ cần sửa tên card là mạng sẽ hoạt động |

# **LỖI CẤU HÌNH MẠNG**

**📍 Tình hình hiện tại của bạn:**

Bạn đã:

* ✅ Sửa đúng **tên card mạng** (ens33)
* ✅ Cấu hình **IP tĩnh, gateway và DNS** trong file YAML
* ✅ Áp dụng cấu hình (sudo netplan apply)

Nhưng khi kiểm tra:

**❌ Lỗi khi ping 8.8.8.8:**

From 192.168.1.100 icmp\_seq=1 Destination Host Unreachable

Từ 192.168.1.100 icmp\_seq=1 Máy chủ đích không thể truy cập được

**❌ Lỗi khi ping google.com:**

ping: google.com: Temporary failure in name resolution

ping: google.com: Lỗi tạm thời trong việc phân giải tên

**🚨 Ý nghĩa của các lỗi này là gì?**

## **1. Destination Host Unreachable (từ IP 192.168.1.100)**

Nghĩa là: **máy bạn có IP**, nhưng **không ra được khỏi mạng nội bộ (LAN)** → không truy cập được Internet.

**Hiểu đơn giản:**

Máy bạn giống như **đã có địa chỉ nhà (IP)**, nhưng **đường đi ra cổng (gateway) thì bị chặn** → không thể ra ngoài khu phố (Internet).

## **2. Temporary failure in name resolution**

Nghĩa là: **DNS không hoạt động** → máy không thể chuyển google.com thành địa chỉ IP để kết nối

**Lý do:** Vì máy **không kết nối được Internet**, nên tất nhiên **DNS cũng không hoạt động được**.

**📌 Điều này cho thấy: Có vấn đề với kết nối giữa máy bạn và Gateway (192.168.1.1)**

Vậy nên **nguyên nhân có thể là một trong các vấn đề dưới đây**:

**📘 PHẦN GIẢI THÍCH CHI TIẾT CHO NGƯỜI MỚI:**

**✅ Máy bạn đã có IP đúng chưa?**

Chạy lại lệnh sau để kiểm tra IP đã được gán:

**ip a**

Bạn sẽ thấy:

**inet 192.168.1.100/24**

→ Nếu thấy IP này trên card ens33 thì **IP đã được gán thành công**.

**❌ Nhưng có thể gateway không hoạt động, hoặc bị sai**

Chúng ta đã cấu hình gateway là:

**via: 192.168.1.1**

💡 Nhưng nếu **192.168.1.1 không phải địa chỉ thật của router** (modem bạn đang dùng), thì máy sẽ không thể ra Internet.

**✅ Bước tiếp theo: KIỂM TRA KẾT NỐI ĐẾN GATEWAY**

Thử ping từ máy bạn tới gateway:

**ping 192.168.1.1**

**Kết quả:**

| **Nếu ping được** | **Nghĩa là gateway hoạt động, vấn đề có thể ở chỗ khác** |
| --- | --- |
| Nếu **ping KHÔNG được** | Gateway không tồn tại hoặc có vấn đề → cần kiểm tra lại |

**❓ Câu hỏi quan trọng: Router/modem của bạn có thực sự dùng IP là 192.168.1.1 không?**

Nếu bạn đang dùng mạng bình thường ở nhà qua modem wifi hoặc router:

* Thử kiểm tra IP gateway của máy host (máy thật – không phải máy ảo)

**Trên máy Windows (host): mở CMD và gõ:**

**ipconfig**

**Tìm dòng:**

**Default Gateway . . . . . . . . . : 192.168.x.1**

→ Dòng đó chính là **IP gateway thật** bạn cần dùng trong file YAML.

📌 **Ghi nhớ**: Gateway phải đúng IP thật của router/modem. Nếu bạn cấu hình sai thì Ubuntu không thể ra Internet.

**✅ Bước sửa lại file YAML (nếu gateway sai)**

Giả sử bạn kiểm tra thấy gateway đúng là 192.168.0.1 thì bạn cần sửa file YAML như sau:

**sudo nano /etc/netplan/50-cloud-init.yaml**

Sửa nội dung:

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: no**

**addresses: [192.168.0.100/24] # phải cùng lớp mạng với gateway**

**routes:**

**- to: default**

**via: 192.168.0.1 # sửa gateway đúng**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

➡️ Sau đó:

**sudo netplan apply**

Rồi kiểm tra lại:

**ping 8.8.8.8**

## **3. Dùng IP động (cho đơn giản và chắc chắn hơn lúc mới học)**

Nếu bạn muốn để máy tự lấy IP từ router, bạn **chỉ cần bật DHCP**, như sau:

Mở file:

**sudo nano /etc/netplan/50-cloud-init.yaml**

Và sửa thành:

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: true**

➡️ Rồi áp dụng:

**sudo netplan apply**

Sau đó kiểm tra:

**ip a # để xem máy đã được cấp IP chưa**

**ping 8.8.8.8**

**ping google.com**

✅ Cách này là **tự động, dễ dùng, nhanh kết nối**, phù hợp cho bạn lúc mới học, chưa rành IP tĩnh.

**🧠 Tóm tắt đơn giản dễ nhớ**

| **Việc bạn làm** | **Ý nghĩa** | **Cần kiểm tra gì** |
| --- | --- | --- |
| Ping 8.8.8.8 | Kiểm tra kết nối Internet | Gateway phải đúng |
| Ping google.com | Kiểm tra DNS | Phụ thuộc vào mạng ra Internet |
| Lỗi Destination Host Unreachable | Máy không ra được ngoài | Sai gateway hoặc kết nối ảo giữa máy ảo và máy thật chưa thông |
| Cách sửa đơn giản | Dùng DHCP | dhcp4: true trong YAML |

**✅ TÓM TẮT NHANH THÔNG TIN BẠN CÓ:**

Dựa vào phần bạn chia sẻ, đây là dòng quan trọng nhất:

**Wireless LAN adapter Wi-Fi:**

**...**

**IPv4 Address. . . . . . . . . . . : 192.168.1.6**

**Default Gateway . . . . . . . . . : 192.168.1.1**

👉 Vậy **gateway thật của bạn là: 192.168.1.1**  
👉 Và máy thật bạn đang dùng mạng **Wi-Fi** với dải mạng: 192.168.1.x

**🧠 Giải thích cho bạn – người mới:**

## **4. Điều kiện để máy ảo ubuntu server kết nối được internet**

**Máy thật của bạn (Windows) đang kết nối Wi-Fi và được cấp IP 192.168.1.6, gateway là 192.168.1.1  
→ Tức là mạng của bạn đang dùng dải IP: 192.168.1.x**

**📌 Vậy nếu bạn muốn máy Ubuntu Server (máy ảo) kết nối được Internet, thì nó phải được đặt IP cùng dải, ví dụ:**

* **IP: 192.168.1.100 (bạn chọn)**
* **Gateway: 192.168.1.1 (router/modem)**
* **DNS: Google DNS (8.8.8.8), hoặc DNS của nhà mạng**

**❗️❗️ NHƯNG...**

Nếu máy ảo của bạn hiện **không được kết nối đến mạng Wi-Fi của máy thật**, thì dù bạn có đặt IP đúng cũng vẫn **không ra được Internet**.

**🔍 KIỂM TRA 1: Cấu hình mạng của máy ảo VMware**

Bạn đang dùng **VMware** và cài **Ubuntu Server**.

→ Bạn cần đảm bảo **kết nối mạng của máy ảo đang dùng chế độ "Bridge" hoặc "NAT"**, để nó **kết nối được với Wi-Fi**.

**✅ Cách kiểm tra & sửa:**

1. Tắt máy ảo Ubuntu
2. Mở VMware
3. Chọn máy ảo Ubuntu của bạn
4. Vào **Settings → Network Adapter**
5. Chọn 1 trong 2 chế độ:

| **Chế độ** | **Ý nghĩa** |
| --- | --- |
| **NAT (Đề xuất)** | Máy ảo "mượn" mạng từ máy thật để ra Internet – an toàn, đơn giản |
| **Bridged** | Máy ảo trở thành 1 máy độc lập trong mạng Wi-Fi như máy thật |

→ Nếu bạn mới học và chỉ cần **máy ảo có Internet**, thì chọn **NAT** là dễ nhất.

📌 Sau khi chọn xong, bật lại máy ảo Ubuntu.

**🛠️ Sau đó: Cập nhật lại file YAML cho đúng**

**⚠️ Bạn đang dùng IP 192.168.1.100 và gateway 192.168.1.1 → ĐÚNG rồi!**

Bây giờ chỉ cần chắc chắn tên card mạng là ens33 (bạn đã kiểm tra rồi).

**✅ Mẫu file YAML cuối cùng – bạn copy và dán:**

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: no**

**addresses: [192.168.1.100/24]**

**routes:**

**- to: default**

**via: 192.168.1.1**

**nameservers:**

**addresses: [8.8.8.8, 1.1.1.1]**

**Thực hiện lại:**

1. Mở file:

**sudo nano /etc/netplan/50-cloud-init.yaml**

1. Dán nội dung như trên
2. Lưu và thoát (Ctrl + O, Enter, rồi Ctrl + X)
3. Áp dụng:

**sudo netplan apply**

1. Kiểm tra lại IP:

**ip a**

1. Ping:

ping 8.8.8.8

ping google.com

## **5. Nếu vẫn chưa được, bạn hãy thử cách DỄ NHẤT:**

**👉 Sử dụng DHCP (máy tự lấy IP, gateway, DNS):**

1. Mở lại file:

**sudo nano /etc/netplan/50-cloud-init.yaml**

1. Dán nội dung:

**network:**

**version: 2**

**ethernets:**

**ens33:**

**dhcp4: true**

1. Lưu và áp dụng:

**sudo netplan apply**

1. Kiểm tra mạng:

**ip a**

**ping 8.8.8.8**

**ping google.com**

## **6. Bạn đã kết nối mạng thành công!**

**👉 1. Khi bạn gõ lệnh:**

**ping google.com**

Nó trả về:

PING google.com (142.250.198.238) 56(84) bytes of data.

64 bytes from nchkgb-ah-in-f14.1e100.net (142.250.198.238): icmp\_seq=1 ttl=128 time=51.1 ms

...

🟢 Điều này nghĩa là:

* Máy bạn đã **kết nối được internet**
* **DNS hoạt động tốt** (vì nó chuyển được từ google.com → địa chỉ IP 142.250.198.238)
* **Mạng đã thông suốt**

**👉 2. Khi bạn gõ:**

**ping 8.8.8.8**

Và thấy:

**64 bytes from 8.8.8.8: icmp\_seq=1 ttl=128 time=32.4 ms**

...

🟢 Nghĩa là:

* Máy bạn **đã ping được đến Internet**, cụ thể là máy chủ của Google (8.8.8.8 là địa chỉ DNS của Google)

📌 Thông số time=32.4 ms nghĩa là: chỉ mất khoảng 32 mili-giây để gửi & nhận phản hồi → mạng rất ổn định!

**🔍 Vậy dòng “20% packet loss” là gì?**

Đây là kết quả:

**5 packets transmitted, 4 received, 20% packet loss**

💡 Nghĩa là:

* Bạn đã gửi 5 gói tin ra mạng
* Nhận lại được 4 gói
* 1 gói bị "rớt" → **20% packet loss**

📌 Đây **không phải lỗi nghiêm trọng** nếu chỉ xảy ra thỉnh thoảng. Nó có thể do:

* Mạng Wi-Fi chập chờn
* Tải mạng cao (đông người dùng)
* Router bận xử lý
* Hoặc đơn giản là **ping bị "timeout" nhẹ** trong lần thử

**🟢 Tóm lại:**

Máy Ubuntu của bạn **đã hoàn toàn kết nối thành công đến Internet. Không còn vấn đề gì cả.**

**📘 Gói kiến thức nhỏ cho bạn:**

**✔️ ping google.com**

* Kiểm tra được 2 thứ cùng lúc:
  + Có ra Internet không?
  + Có dùng được DNS không? (vì phải tra ra IP của google.com)

**✔️ ping 8.8.8.8**

* Chỉ kiểm tra được **kết nối mạng**
* Không cần DNS → dùng để test khi nghi ngờ DNS bị hỏng

**🎯 Bạn đã làm đúng những gì cần làm:**

| **Việc cần làm** | **Kết quả** |
| --- | --- |
| Cấu hình DHCP tự động | ✅ Thành công |
| Kết nối đến Internet | ✅ Ping 8.8.8.8 được |
| Sử dụng DNS tốt | ✅ Ping google.com được |
| Biết cách sửa YAML | ✅ Làm được |
| Biết cách kiểm tra mạng | ✅ Thành thạo lệnh ping |

# **CÂU LỆNH PING**

## **1. "ping" là gì?**

ping là một **lệnh kiểm tra kết nối mạng** giữa **máy bạn** và **một máy khác** trên Internet hoặc trong mạng nội bộ.

**💡 Nó giống như:**

Bạn hét to "Ê, có ai ở đó không?"  
→ Nếu có người trả lời lại: "Có!"  
→ Nghĩa là **kết nối được**, họ **nghe thấy bạn**

## **2. Câu lệnh ping dùng để làm gì?**

| **Mục đích** | **Ví dụ** |
| --- | --- |
| ✔️ Kiểm tra xem máy có **kết nối Internet** không | ping 8.8.8.8 |
| ✔️ Kiểm tra **DNS** có hoạt động không (tức là dịch tên website sang IP được không) | ping google.com |
| ✔️ Kiểm tra máy khác trong cùng mạng có **tồn tại và online** không | ping 192.168.1.1 (ping router) |

## **3. Cách dùng cơ bản**

**ping <địa\_chỉ\_IP\_hoặc\_tên\_miền>**

**Ví dụ:**

**ping 8.8.8.8 # Kiểm tra kết nối tới máy chủ DNS của Google**

**ping google.com # Kiểm tra cả kết nối mạng và DNS**

**ping 192.168.1.1 # Kiểm tra kết nối tới router/modem trong nhà bạn**

## **4. Khi bạn chạy lệnh ping, máy sẽ làm gì?**

Khi bạn chạy lệnh:

**ping google.com**

Nó sẽ:

1. **Tìm IP của google.com** (nhờ vào DNS)
2. Gửi gói tin nhỏ gọi là **ICMP Echo Request** đến địa chỉ đó
3. Nếu Google nhận được, nó sẽ gửi lại **ICMP Echo Reply**
4. Mỗi dòng như sau là một phản hồi:

64 bytes from 142.250.198.238: icmp\_seq=1 ttl=128 time=32.4 ms

Nghĩa là:

* Gửi thành công
* Được phản hồi
* Tốn 32.4 mili-giây để đi và về

## **5. Ping chạy mãi, dừng lại bằng cách nào?**

Ping sẽ gửi liên tục (theo mặc định) để kiểm tra kết nối.

👉 Nhấn tổ hợp phím:

**Ctrl + C**

Để dừng lệnh ping

## **6. Sau khi dừng, nó hiển thị thống kê**

Ví dụ:

7 packets transmitted, 7 received, 0% packet loss

* Đã gửi 7 gói
* Nhận lại đủ 7
* Không bị mất gói nào → mạng OK

## **7. Nếu ping bị lỗi thì sao?**

| **Trường hợp** | **Ý nghĩa** |
| --- | --- |
| ping: unknown host | Không tìm được địa chỉ → lỗi DNS |
| Destination Host Unreachable | Máy kia không online hoặc mất mạng |
| Request timed out | Gửi gói đi nhưng không có trả lời → mạng chập chờn hoặc bị chặn |
| 100% packet loss | Toàn bộ kết nối đều không thành công |

**🟢 Tóm gọn dễ nhớ:**

| **Tên** | **Ý nghĩa** |
| --- | --- |
| ping | Lệnh kiểm tra **kết nối mạng** giữa 2 máy |
| ping 8.8.8.8 | Kiểm tra máy có ra được Internet không |
| ping google.com | Kiểm tra cả kết nối mạng **và** DNS |
| Ctrl + C | Dừng lệnh ping |
| packet loss | Tỷ lệ gói bị rơi mất trong quá trình gửi nhận |

**✨ Mẹo:**

🔍 Nếu bạn thấy máy **ping được IP nhưng không ping được tên miền**, thì khả năng cao là:

* DNS bị lỗi
* Cấu hình /etc/resolv.conf sai
* Không có nameserver hợp lệ

# **DEFAULT GATEWAY**

**✅ Khái niệm đơn giản:**

**Default Gateway** là "cửa ngõ mặc định" để máy tính **ra ngoài mạng** (ra internet hoặc đến mạng khác).

Bạn có thể hiểu:

📦 Máy tính của bạn (Ubuntu) khi muốn **gửi dữ liệu ra ngoài mạng**, nó phải **gửi qua một cái “cửa”** — cửa đó chính là **default gateway**.  
Nếu không có cửa này, nó **không biết gửi đi đâu** → Không kết nối được internet.

**📌 Vậy gateway này là của ai? lấy từ đâu?**

**Có 3 trường hợp cần phân biệt rõ ràng:**

| **Thiết bị** | **Vai trò** |
| --- | --- |
| 💻 **Ubuntu Server (máy ảo)** | Là “máy khách” (client), muốn kết nối mạng |
| 🧑‍💻 **Máy thật của bạn (Windows)** | Là máy đang chạy phần mềm VMware, đóng vai trò như **“host”** |
| 📡 **Router / Modem / Cục Wi-Fi nhà bạn** | Là nơi **thật sự kết nối ra Internet**, cũng thường là nơi làm **default gateway** |

**🧭 Cách Ubuntu (trên máy ảo) xác định Gateway:**

Ubuntu không tự nghĩ ra Gateway. Nó lấy gateway theo **2 cách**:

**✳️ Cách 1: Tự động (DHCP)**

* Nếu bạn dùng cấu hình **dhcp4: yes** trong YAML → Máy Ubuntu **tự xin IP + gateway từ hệ thống mạng (thường là router)**

➡️ Đây là cách bạn vừa dùng thành công!

**✳️ Cách 2: Thủ công (static)**

* Nếu bạn dùng **dhcp4: no** và tự điền gateway4: 192.168.1.1 → Bạn đang **chỉ định một default gateway thủ công**

➡️ Tuy nhiên, nếu bạn điền sai gateway (không tồn tại trong mạng thật), Ubuntu sẽ không ra mạng được!

**📌 Còn dòng 128.0.1.1 là gì?**

Bạn nói rằng:

Trong file YAML, gateway là 128.0.1.1

🛑 **Đây có thể là sai sót**, vì địa chỉ 128.0.1.1 **không thuộc mạng nội bộ của bạn** (mạng gia đình thường là 192.168.x.x hoặc 10.x.x.x)

📛 Nếu bạn dùng gateway này, **Ubuntu sẽ không thể ra internet**, vì nó cố gửi dữ liệu qua 1 "cánh cửa không tồn tại".

**✅ Vậy còn cái bạn vừa chạy: ip route**

default via 192.168.101.2 dev ens33 proto dhcp src 192.168.101.140 metric 100

**Nghĩa là:**

* default via 192.168.101.2 → Đây là **default gateway** của máy Ubuntu hiện tại
* Nó dùng card mạng ens33
* IP máy Ubuntu là 192.168.101.140
* Được cấp qua DHCP → nghĩa là **máy tự động lấy địa chỉ từ mạng**

**❓ Vậy Gateway 192.168.101.2 là ai?**

**Đó là gateway do VMware NAT (VMnet8) cung cấp.**

👉 Vì bạn đang chạy Ubuntu trong VMware, và cấu hình VMware đang dùng **NAT** (Network Address Translation), thì:

* VMware sẽ tạo ra một **mạng ảo riêng** cho máy Ubuntu (thường là 192.168.101.0/24)
* Và nó sẽ **đóng vai trò gateway** ở địa chỉ **192.168.101.2**
* Từ đó, dữ liệu mới đi ra ngoài Internet qua máy Windows → rồi đến modem → rồi ra Internet thật.

💡 Mạng ảo này **khác hoàn toàn** với mạng Wi-Fi thực mà bạn dùng trên Windows.

**🔁 Mối liên hệ tổng quát:**

[Ubuntu Server (VM)]

│

│ (Default Gateway: 192.168.101.2)

▼

[VMware NAT ảo]

│

▼

[Windows máy thật]

│

▼

[Router Wi-Fi: 192.168.1.1]

│

▼

[Internet]

**✅ Tóm tắt dễ hiểu:**

| **Câu hỏi** | **Trả lời** |
| --- | --- |
| Gateway là gì? | Là "cánh cửa" để máy tính ra Internet |
| Gateway lấy từ đâu? | Do bạn cấu hình thủ công hoặc lấy tự động từ DHCP |
| Gateway hiện tại Ubuntu là gì? | 192.168.101.2 – do VMware NAT cấp |
| Tại sao khác 128.0.1.1? | Vì 128.0.1.1 không tồn tại trong mạng ảo của bạn |
| Máy Ubuntu đang ra Internet qua đâu? | Qua gateway ảo của VMware, rồi đến Windows, rồi ra router thật |

Nếu bạn vẫn muốn **hiểu sâu hơn về mô hình mạng ảo trong VMware**, mình có thể giúp bạn vẽ sơ đồ minh họa hoặc hướng dẫn chi tiết.

Bạn có muốn tiếp tục tìm hiểu phần này không? Hay mình gợi ý bài học tiếp theo về quản lý người dùng, file, hay dịch vụ trên Ubuntu cho bạn?

# **SỰ KHÁC NHAU GIỮA HOSTNAME, DOMAIN NAME VÀ FQDN**

**🧠 Bạn cần nắm rõ 3 khái niệm:**

| **Thuật ngữ** | **Viết tắt** | **Nghĩa tiếng Việt** | **Mức độ** |
| --- | --- | --- | --- |
| hostname | – | Tên máy (máy tính / server) | Cơ bản |
| domain name | – | Tên miền (tổ chức, mạng) | Trung bình |
| FQDN | Fully Qualified Domain Name | Tên miền đầy đủ | Kết hợp của 2 cái trên |

## **1. Hostname – TÊN MÁY**

**📌 Định nghĩa:**

Là **tên gọi của máy tính hoặc máy chủ trong một mạng nội bộ** hoặc trong một hệ thống.

**📌 Ví dụ:**

* ubuntu-server
* web01
* nammoadidaphat
* lebao-laptop

**✅ Đặc điểm:**

* Chỉ là **một từ đơn** (không có dấu chấm ., không phải tên miền).
* Được thiết lập bằng lệnh:

**sudo hostnamectl set-hostname ubuntu-server**

* Khi bạn dùng lệnh:

**hostname**

→ nó sẽ trả về đúng hostname bạn đã đặt.

**🧠 Quan trọng:**

* hostname chỉ dùng **trong nội bộ máy hoặc mạng**.
* **Không thể dùng hostname để truy cập từ Internet nếu không có DNS map.**

## **2. Domain Name – TÊN MIỀN**

**📌 Định nghĩa:**

Là **tên đại diện cho một tổ chức, công ty, dịch vụ...** trên Internet.

**📌 Ví dụ:**

* google.com
* vnexpress.net
* lebao.com
* ubuntuserver.local

**✅ Đặc điểm:**

* Gồm **các phần cách nhau bởi dấu chấm**, đọc từ trái sang phải:
  + www.google.com: phần google là tên tổ chức, com là tên miền cấp cao (TLD)
  + example.local: tên miền trong mạng LAN nội bộ
* **Domain name được quản lý thông qua DNS (Domain Name System)**

## **3. FQDN – Fully Qualified Domain Name**

**📌 Định nghĩa:**

Là **tên đầy đủ của một máy chủ** bao gồm:

**Hostname + Domain name**

**✅ Công thức:**

FQDN = Hostname + Domain Name

**📌 Ví dụ:**

| **Hostname** | **Domain Name** | **FQDN** |
| --- | --- | --- |
| web01 | example.com | web01.example.com |
| ubuntu-server | lab.local | ubuntu-server.lab.local |

→ Đây là **địa chỉ định danh chính xác và duy nhất** cho một máy trong hệ thống mạng – dùng trong Internet và DNS.

**🔍 So sánh tổng thể**

| **Đặc điểm** | **Hostname** | **Domain Name** | **FQDN** |
| --- | --- | --- | --- |
| Là gì? | Tên máy cục bộ | Tên miền của tổ chức | Tên đầy đủ của máy |
| Dùng ở đâu? | Nội bộ máy/mạng | Trên Internet hoặc mạng nội bộ | Hệ thống DNS |
| Có dấu chấm .? | ❌ Không | ✅ Có | ✅ Có |
| Truy cập máy từ xa được không? | ❌ Không | ✅ Nếu có DNS | ✅ Chính xác nhất |

**🔧 Kiểm tra trên Ubuntu**

**✅ Xem hostname:**

**hostname**

**✅ Xem FQDN (nếu được thiết lập):**

**hostname -f**

→ Nếu chưa cấu hình domain name đúng, thì FQDN có thể trả về giống hostname.

**📁 Tệp liên quan:**

| **File** | **Mục đích** |
| --- | --- |
| /etc/hostname | Chứa tên hostname |
| /etc/hosts | Ánh xạ hostname ↔ IP nội bộ |
| /etc/resolv.conf | Chứa DNS server để phân giải tên miền |
| /etc/hosts + DNS | Giúp hệ thống hiểu hostname + domain name = FQDN |

**📌 Ví dụ thực tế:**

Giả sử bạn có cấu hình sau trong file /etc/hosts:

**127.0.1.1 ubuntu-server.example.com ubuntu-server**

→ Thì:

* **ubuntu-server là hostname**
* **example.com là domain name**
* **ubuntu-server.example.com là FQDN**

Bạn có thể kiểm tra bằng lệnh:

**hostname -f**

→ Sẽ hiển thị ubuntu-server.example.com

**✅ Tổng kết dễ hiểu:**

| **Bạn cần nhớ** | **Ghi chú** |
| --- | --- |
| hostname là tên cục bộ của máy | Đơn giản, 1 từ, dùng trong máy hoặc mạng LAN |
| domain name là tên miền trên mạng | Dùng để xác định tổ chức (VD: google.com) |
| FQDN là tên đầy đủ: hostname + domain | Giống như địa chỉ chính thức, định danh duy nhất |

# **CÀI PHẦN MỀM**

Trước khi cài đặt bất kỳ phần mềm nào, bước quan trọng nhất và bắt buộc bạn phải làm là **cập nhật danh sách gói phần mềm** để đảm bảo bạn đang cài đặt phiên bản mới nhất và an toàn nhất.

**Các bước chuẩn bị trước khi cài đặt phần mềm**

Bạn sẽ cần chạy hai lệnh sau đây theo thứ tự:

## **1. Cập nhật danh sách gói phần mềm (apt update)**

Lệnh này sẽ tải về danh sách các gói phần mềm mới nhất từ các kho lưu trữ (repositories) trên Internet. Nó giúp hệ thống của bạn biết những phiên bản phần mềm nào có sẵn để cài đặt.

Gõ lệnh sau và nhấn **Enter**:

Bash

**sudo apt update**

* **sudo**: Cho phép bạn chạy lệnh với quyền quản trị viên (cần thiết cho các lệnh quản lý hệ thống).
* **apt**: Là công cụ quản lý gói phần mềm (Advanced Packaging Tool) của Ubuntu.
* **update**: Chỉ thị để cập nhật danh sách gói.

## **2. Nâng cấp các gói đã lỗi thời (apt upgrade)**

Sau khi cập nhật danh sách, bạn nên nâng cấp các gói phần mềm **đã được cài đặt** trên server của bạn (như kernel, thư viện hệ thống,...) lên phiên bản mới nhất.

Gõ lệnh sau và nhấn **Enter**:

Bash

**sudo apt upgrade**

* Hệ thống có thể hỏi bạn **Do you want to continue? [Y/n]**. Bạn gõ **y** (viết tắt của Yes) và nhấn **Enter** để đồng ý nâng cấp.

**Bước Kế Tiếp: Cài đặt Phần mềm**

Sau khi hai lệnh trên chạy xong, hệ thống của bạn đã sẵn sàng. Bạn có thể cài đặt phần mềm bằng cú pháp:

Bash

**sudo apt install ten\_phan\_mem**

Ví dụ, nếu bạn muốn cài đặt trình duyệt web dòng lệnh **Lynx** (để thử nghiệm):

Bash

**sudo apt install lynx**

Bạn muốn cài đặt phần mềm gì cho server của mình? (Ví dụ: Web server **Nginx**, Database **MySQL**, hay công cụ **htop** để theo dõi hiệu suất?)

# **CÀI ĐẶT NGINX**

Tuyệt vời! Cài đặt **Nginx** (phát âm là "Engine-X") là một trong những bài thực hành cơ bản và quan trọng nhất trên Ubuntu Server. Nginx là một máy chủ web rất nhanh và hiệu quả.

Dưới đây là hướng dẫn chi tiết từng bước để bạn cài đặt và kiểm tra Nginx:

**Thực hành: Cài đặt Máy chủ Web Nginx**

Hãy đảm bảo rằng bạn đã chạy lệnh sudo apt update và sudo apt upgrade ở các bước trước để hệ thống sẵn sàng.

## **1. Cài đặt Gói Nginx**

Bạn sử dụng công cụ quản lý gói apt để cài đặt Nginx từ kho lưu trữ của Ubuntu.

Gõ lệnh sau và nhấn **Enter**:

Bash

**sudo apt install nginx**

1. Hệ thống sẽ liệt kê các gói cần cài đặt và hỏi bạn có muốn tiếp tục không.
2. Gõ y (Yes) và nhấn **Enter**.
3. Quá trình này sẽ tải xuống và tự động cài đặt Nginx cùng với các gói phụ thuộc (dependency) cần thiết.

## **2. Kiểm tra Trạng thái Dịch vụ**

Sau khi cài đặt xong, dịch vụ Nginx sẽ tự động khởi động. Bạn cần kiểm tra để đảm bảo nó đang chạy:

Gõ lệnh sau và nhấn **Enter**:

Bash

**sudo systemctl status nginx**

**Kết quả mong đợi:**

* Bạn sẽ thấy dòng chữ Active: active (running) (thường có màu xanh lá cây).
* Nhấn phím q (viết tắt của quit) để thoát khỏi màn hình trạng thái và quay lại dòng lệnh.

Mặc dù Nginx đang chạy bên trong máy ảo, bạn vẫn chưa thể truy cập nó từ trình duyệt trên máy tính vật lý của mình vì hai lý do chính: **Tường lửa** và **Địa chỉ IP**.

## **3. Điều chỉnh Tường lửa (Firewall)**

Mặc định, Ubuntu Server sử dụng tường lửa **UFW** (Uncomplicated Firewall) để bảo vệ server. Để người khác có thể truy cập vào trang web của bạn qua Nginx, bạn cần mở các cổng HTTP (80) và HTTPS (443).

Ubuntu Server có tường lửa UFW (Uncomplicated Firewall) bật sẵn để bảo mật. Bạn cần mở cổng HTTP (cổng 80) để cho phép lưu lượng truy cập web đi qua.

Kích hoạt Tường lửa UFW (Bắt buộc):

**sudo ufw enable**

* Hệ thống có thể hỏi: **Command may disrupt existing ssh connections. Proceed with operation (y|n)?**
* Bạn gõ **y** (Yes) và **Enter**. (Đừng lo lắng, vì bạn đã mở cổng HTTP trước, và SSH cũng sẽ được giữ mở theo mặc định).

Ở bước này, cổng 80 của Ubuntu có thể đã được sử dụng bởi tiến trình khác. Bạn phải liệt kê các tiến trình sử dụng cổng 80 bằng lệnh:

**sudo lsof -i :80**

Nếu bạn dừng httpd hoặc apache, có thể xảy ra lỗi vì bạn đã dừng nhầm tên dịch vụ. Do đó, cần phải tìm đúng tên dịch vụ.

**invcmis@ubuntusvr:~$ systemctl list-units --type=service | grep -i apache**

**snap.nextcloud.apache.service**

**ud.apache**

**nvcmis@ubuntusvr:"$**

* + Hiển thị đầy đủ đúng tên dịch vụ chứa chuỗi apache

**sudo snap set nextcloud apache.enable=false**

Cuối cùng, khởi động và kích hoạt nginx

**sudo systemctl start nginx**

**sudo systemctl enable nginx**

Bạn có thể thấy các cấu hình có sẵn của Nginx bằng lệnh:

Bash

**sudo ufw app list**

**Bạn sẽ thấy các hồ sơ (profiles) sau:**

* **Nginx Full: Mở cả cổng 80 (HTTP) và 443 (HTTPS).**
* **Nginx HTTP: Chỉ mở cổng 80 (HTTP).**
* **Nginx HTTPS: Chỉ mở cổng 443 (HTTPS).**

Bạn nên mở cổng HTTP cơ bản để kiểm tra. Gõ lệnh sau để cho phép lưu lượng truy cập qua HTTP:

Bash

**sudo ufw allow 'Nginx HTTP'**

**sudo ufw allow 'Nginx Full'**

Cho phép truy cập HTTP (cổng 80) qua tường lửa UFW:

Bạn đã mở cổng **80** (cổng tiêu chuẩn cho HTTP) trên tường lửa, cho phép người dùng bên ngoài truy cập vào trang web của bạn.

Tường lửa **UFW** (Uncomplicated Firewall) hiện đã được **bật** (active). Nó sẽ tự động khởi động cùng với hệ điều hành mỗi khi máy chủ của bạn được bật. Điều này cực kỳ quan trọng cho bảo mật server của bạn. Vì bạn đã chạy lệnh sudo ufw allow 'Nginx HTTP' trước đó, tường lửa sẽ cho phép truy cập qua cổng 80 (HTTP).

Sau đó, bạn kiểm tra lại trạng thái tường lửa để đảm bảo nó đã hoạt động và các cổng đã mở:

Bash

**sudo ufw status**

## **4. Kiểm tra Truy cập Web từ Trình duyệt**

Đây là bước cuối cùng để xác nhận Nginx đã hoạt động hoàn hảo.

1. Bạn cần tìm **Địa chỉ IP** của máy chủ Ubuntu của bạn. Gõ lệnh sau trong Terminal:

Bash

**ip a**

1. Tìm dòng có giao diện mạng, thường là ens33 hoặc eth0, và tìm địa chỉ inet (ví dụ: 192.168.101.132 - địa chỉ này có thể khác trên máy của bạn).
2. **Trên máy tính vật lý của bạn** (không phải trong máy ảo):
   * **Mở một trình duyệt web (Chrome, Firefox, Edge).**
   * **Nhập Địa chỉ IP bạn vừa tìm được vào thanh địa chỉ:**
   * **http://[Địa chỉ IP của bạn]**

**Kết quả mong đợi:**

* Nếu thành công, trình duyệt sẽ hiển thị trang chào mừng mặc định của Nginx, với tiêu đề là **"Welcome to nginx!"**.

Chúc mừng, máy chủ web Nginx của bạn đã được cài đặt và hoạt động!

## **5. Tổng kết**

Tuyệt vời! **Xin chúc mừng!** Màn hình này có nghĩa là bạn đã **hoàn thành thành công việc cài đặt và cấu hình cơ bản một máy chủ web** (Web Server) trên máy ảo của mình. Đây chính là **trang chào mừng mặc định** của máy chủ **Nginx** mà bạn đã cài đặt.

Bây giờ, hãy cùng nhau nhìn lại toàn bộ hành trình bạn vừa đi qua, vai trò và tác dụng của từng bước:

**1. Tổng quan: Hành trình Cài đặt Web Server**

| Thứ Tự | Hành Động Bạn Thực Hiện | Mục Tiêu Chính (Vai trò) |
| --- | --- | --- |
| **1.** | **Cài đặt Ubuntu Server trên VMware.** | Tạo ra một **Máy chủ Ảo (Virtual Server)** hoạt động độc lập, mô phỏng một máy chủ vật lý ngoài đời thực. |
| **2.** | **Cài đặt OpenSSH.** | Cho phép bạn **truy cập và điều khiển** server từ xa thông qua dòng lệnh một cách bảo mật (không cần dùng chuột). |
| **3.** | **Chạy**sudo apt update**và**sudo apt upgrade**.** | **Cập nhật và nâng cấp** hệ thống, đảm bảo server của bạn an toàn và có phiên bản phần mềm mới nhất. |
| **4.** | **Cài đặt Nginx (**sudo apt install nginx**).** | Cài đặt **Máy chủ Web (Web Server)**. Đây là phần mềm lắng nghe yêu cầu từ trình duyệt và gửi lại trang web. |
| **5.** | **Kiểm tra Nginx (**sudo systemctl status nginx**).** | **Xác nhận** rằng phần mềm web server đã khởi động và đang chạy ổn định trong hệ điều hành. |
| **6.** | **Bật và cấu hình tường lửa UFW (**sudo ufw enable**,**sudo ufw allow 'Nginx HTTP'**).** | **Bảo mật** server, chỉ cho phép truy cập vào các dịch vụ cần thiết (cổng 80 cho web) và chặn các cổng còn lại. |
| **7.** | **Tìm địa chỉ IP (**ip a**) và truy cập bằng trình duyệt.** | **Kết nối** máy tính vật lý của bạn với máy chủ ảo, **xác nhận** rằng toàn bộ hệ thống (OS, Nginx, Tường lửa, Mạng) đã hoạt động đồng bộ. |

**2. Vai trò và Tác dụng của Từng Thành phần**

Tất cả những gì bạn làm được gọi là thiết lập môi trường **LAMP/LEMP** cơ bản, là nền tảng của hầu hết mọi trang web trên thế giới.

**A. Máy Ảo (VMware)**

* **Tác dụng:** Cho phép bạn thực hành mọi thứ mà không ảnh hưởng đến máy tính chính của mình. Nó là một môi trường biệt lập, mô phỏng chính xác một máy chủ thực sự mà bạn có thể thuê từ các nhà cung cấp dịch vụ như AWS, Google Cloud hay DigitalOcean.

**B. Ubuntu Server (Hệ điều hành)**

* **Tác dụng:** Là nền tảng cơ bản để chạy phần mềm (giống như Windows hay macOS). **Server Edition** được tối ưu hóa cho tốc độ, hiệu suất và bảo mật, **chỉ dùng dòng lệnh** để tiêu thụ ít tài nguyên nhất.

**C. Nginx (Máy chủ Web)**

* **Tác dụng:** Nginx là "người phục vụ" chính.
  + Nó **lắng nghe** (qua cổng 80) khi bạn gõ địa chỉ IP.
  + Nó **nhận** yêu cầu từ trình duyệt của bạn.
  + Nó **tìm** tập tin trang web (hiện tại là trang "Welcome to nginx!").
  + Nó **gửi** tập tin đó trở lại trình duyệt của bạn để hiển thị.

**D. Tường lửa (UFW)**

* **Tác dụng:** Tường lửa là "người bảo vệ" cho server.
  + **Trước khi mở cổng 80:** Tường lửa chặn mọi truy cập, server của bạn an toàn nhưng không ai xem được web.
  + **Sau khi mở cổng 80:** Tường lửa chỉ cho phép các gói tin đến cổng 80 đi qua Nginx, trong khi vẫn chặn các cổng khác, bảo vệ server khỏi các cuộc tấn công không mong muốn.

**3. Bạn đã Học được gì?**

Bạn không chỉ cài đặt một phần mềm. Bạn đã thiết lập thành công một **mô hình client-server** hoạt động hoàn chỉnh:

* **Client (Khách hàng):** Trình duyệt web trên máy tính vật lý của bạn.
* **Server (Máy chủ):** Máy ảo Ubuntu chạy Nginx.

**Kết quả:** Bạn đã biến một máy ảo Ubuntu chỉ biết gõ lệnh thành một **thực thể có thể phục vụ nội dung ra mạng**.

**Bước tiếp theo** là thay thế trang "Welcome to Nginx" bằng trang web của riêng bạn (HTML, CSS) và học cách cài đặt các thành phần khác như cơ sở dữ liệu (MySQL) để lưu trữ dữ liệu.

# **TƯỜNG LỬA**

**🧠 GIẢI THÍCH TOÀN DIỆN TỪ CON SỐ 0**

(dễ hiểu, cụ thể, không cần biết kỹ thuật)

**🧱 PHẦN 1: GỐC RỄ – WEB LÀ GÌ? MẠNG LÀ GÌ?**

**🌐 1. Internet/web hoạt động như thế nào?**

* Web là một mạng **kết nối giữa các máy tính với nhau**.
* Có 2 vai trò:
  + **Client (máy người dùng)** – thường là bạn, dùng trình duyệt
  + **Server (máy phục vụ)** – nơi đặt file web, chạy phần mềm như NGINX

👉 Máy chủ (server) chờ yêu cầu → gửi file về cho client.

**💻 2. Mạng hoạt động theo cách: Gửi dữ liệu từ máy A đến máy B qua địa chỉ IP**

* IP = địa chỉ số, giống như địa chỉ nhà
* Để máy khác (trình duyệt) truy cập web, nó phải biết địa chỉ IP của máy server

**🔐 PHẦN 2: VẬY TẠI SAO PHẢI CÓ “TƯỜNG LỬA”?**

**🔥 Tường lửa là gì?**

🔐 **Tường lửa (firewall)** giống như **bảo vệ ở cổng nhà**, kiểm soát:

* Ai được phép vào
* Ai bị chặn lại

**🧱 Có nhiều loại tường lửa**

| **Loại** | **Chạy ở đâu** | **Vai trò** |
| --- | --- | --- |
| Phần cứng | Ở router hoặc thiết bị mạng | Bảo vệ toàn mạng |
| Phần mềm | Chạy trên máy tính như Ubuntu, Windows | Bảo vệ riêng từng máy |

**🧯 Trong Ubuntu Server, tường lửa phổ biến là UFW**

**✅ PHẦN 3: UFW LÀ GÌ?**

**🔑 UFW = Uncomplicated Firewall**

(UFW nghĩa là "tường lửa đơn giản")

Là **một công cụ rất đơn giản trên Ubuntu Server** để giúp bạn **mở hoặc đóng các cổng mạng**.

**Nói đơn giản:**

* Máy Ubuntu có hàng ngàn "cổng" để máy khác truy cập vào
* Bạn **không muốn ai cũng được phép truy cập toàn bộ**
* UFW giúp bạn **cho phép hoặc chặn các cổng cụ thể**

**🌐 Ví dụ:**

| **Cổng** | **Dịch vụ** | **Có nên mở không?** |
| --- | --- | --- |
| 80 | Web (http) | ✅ cần mở để người dùng truy cập web |
| 443 | Web bảo mật (https) | ✅ nên mở |
| 22 | SSH – để điều khiển từ xa | ✅ nên mở nếu bạn muốn điều khiển server |
| 3306 | MySQL (database) | ❌ Không nên mở nếu không cần |

**🧩 PHẦN 4: UFW CÓ MỐI QUAN HỆ GÌ VỚI UBUNTU SERVER TRÊN VMWARE?**

**👉 Khi bạn cài Ubuntu Server trên VMware, bạn tạo ra một máy tính riêng biệt chạy trong máy Windows.**

* Máy này có **hệ điều hành riêng (Ubuntu)**
* Máy này có **mạng riêng (IP riêng)**
* Máy này có thể **bị tấn công hoặc truy cập từ máy ngoài**

**Do đó, Ubuntu Server trong VMware vẫn cần được bảo vệ bằng tường lửa UFW, y như server ngoài đời thật!**

**🖥️ PHẦN 5: TẠI SAO KHÔNG DÙNG WINDOWS ĐỂ LÀM SERVER?**

❓Tại sao không dùng Windows thật để cài NGINX, sao phải dùng Ubuntu Server?

**Câu trả lời:**

| **Lý do** | **Giải thích** |
| --- | --- |
| 🐧 Ubuntu là hệ điều hành **chuyên dùng cho server** | Gọn nhẹ, không cần giao diện |
| 🧰 Dễ cài các phần mềm server (Nginx, Apache, MySQL...) | Không rườm rà như Windows |
| 🛡️ Bảo mật cao hơn | Vì không có giao diện, ít bị virus |
| 💡 Học Ubuntu = học cách vận hành server thật ngoài đời | Giống như bạn tập làm đầu bếp trong bếp thật vậy |

💡 **Windows là hệ điều hành dành cho người dùng cuối (Client)**  
Ubuntu Server là hệ điều hành **dành cho máy phục vụ (Server)**

**🌐 PHẦN 6: VẬY TẠI SAO PHẢI NHẬP ĐỊA CHỈ IP?**

📡 Vì IP chính là **địa chỉ để máy tính khác biết đường tìm đến Server**

**Khi bạn gõ:**

http://192.168.101.143

Nghĩa là bạn đang nói với Chrome:

“Này, trình duyệt! Hãy đến gõ cửa nhà 192.168.101.143 (Ubuntu Server) và hỏi: Cho tôi xem trang web bạn có không?”

* Máy thật gửi yêu cầu đến IP đó
* NGINX nhận yêu cầu, trả về file HTML
* Trình duyệt Chrome hiển thị nội dung

**✅ TÓM LẠI – BẢN ĐỒ MỐI QUAN HỆ GIỮA CÁC THÀNH PHẦN**

| **Thành phần** | **Vai trò** | **Ví dụ** |
| --- | --- | --- |
| 🖥️ Máy Windows thật | Máy người dùng (Client) | Bạn đang dùng |
| 💻 Ubuntu Server trong VMware | Máy chủ (Server) | Nơi cài NGINX và chứa web |
| 🔐 UFW | Tường lửa của Ubuntu Server | Chặn hoặc cho phép truy cập vào |
| 🌐 IP Address | Địa chỉ nhà của Server | 192.168.101.143 |
| 🌍 Trình duyệt Chrome | Công cụ để "ghé thăm" server | Gõ IP vào để truy cập |
| 📦 NGINX | Phần mềm phục vụ web | Nhận yêu cầu → gửi file HTML |

**🧠 VÍ DỤ TƯỞNG TƯỢNG NHƯ ĐỜI THẬT**

* Ubuntu Server = Quán cà phê
* UFW = Bảo vệ quán cà phê, chỉ cho khách nào hợp lệ vào
* NGINX = Nhân viên phục vụ
* File HTML = Ly cà phê
* IP = Địa chỉ quán
* Chrome = Người khách
* Bạn = người dùng, muốn uống cà phê

**✅ KẾT LUẬN (cô đọng)**

🔒 **UFW là tường lửa trong Ubuntu Server** – bảo vệ server khỏi truy cập không mong muốn.  
🌍 Bạn dùng Chrome trên máy thật, gõ địa chỉ IP để truy cập website.  
💻 Ubuntu Server trong VMware là mô phỏng máy chủ thật ngoài đời – gọn nhẹ, an toàn, chuyên nghiệp hơn Windows.  
🌐 IP là "địa chỉ nhà", để các máy biết tìm nhau trong mạng.

Bạn hỏi rất đúng, đang đi rất vững! 🔥  
Nếu bạn muốn, mình có thể hướng dẫn tiếp:

* ✅ Cách bật/tắt tường lửa UFW
* ✅ Cách cho phép/chặn cổng cụ thể (ví dụ: chỉ mở port 80)
* ✅ Cách bảo vệ server khỏi tấn công

Chỉ cần nói “Mình muốn học tiếp về…” là mình sẽ **giải thích đơn giản và rõ như thế này** nhé!

**You said:**

Việc mở cổng 80 rồi 443 trên tường lửa tức là sao?

**ChatGPT said:**

Bạn hỏi **“Việc mở cổng 80 rồi 443 trên tường lửa tức là sao?”** — và đây chính là **trái tim** của việc xây dựng một web server cơ bản, mình sẽ giải thích **từ gốc, dễ hiểu nhất** cho bạn — như đang nói chuyện trực tiếp vậy nhé!

**🧠 1. HÃY BẮT ĐẦU TỪ ĐIỀU CƠ BẢN: Cổng (Port) là gì?**

**🏡 Hình dung:**

* Mỗi máy tính giống như **một tòa nhà**.
* Trong tòa nhà đó có **nhiều cửa** – mỗi cửa có **số hiệu**: cửa 80, cửa 443, cửa 22...
* Người ngoài (ví dụ trình duyệt) muốn **vào nhà**, thì phải **đến đúng cánh cửa phù hợp với mục đích**.

| **Port số** | **Dùng cho gì?** | **Ví dụ thực tế** |
| --- | --- | --- |
| 80 | Web bình thường (http) | Bạn gõ http://example.com |
| 443 | Web bảo mật (https) | Bạn gõ https://example.com |
| 22 | SSH (truy cập điều khiển từ xa) | Kết nối terminal vào server |
| 3306 | MySQL database | Truy cập cơ sở dữ liệu |

**🔥 2. TƯỜNG LỬA LÀ GÌ?**

Tường lửa (firewall) giống như **bảo vệ gác cổng nhà**, quyết định **ai được đi qua cửa nào**.

* Nếu **cổng bị đóng**, không ai đi vào được → bạn không truy cập được web
* Nếu **cổng được mở**, người ngoài mới gửi yêu cầu đến máy được

**🌐 3. CỤ THỂ CỔNG 80 VÀ 443 DÙNG ĐỂ LÀM GÌ?**

| **Cổng** | **Giao thức** | **Mô tả** |
| --- | --- | --- |
| **80** | HTTP | Giao tiếp web **không bảo mật** |
| **443** | HTTPS | Giao tiếp web **có mã hóa, an toàn** |

Khi bạn mở Chrome và gõ địa chỉ IP, trình duyệt sẽ **ngầm gửi yêu cầu đến cổng 80 hoặc 443** trên server.

**🛡️ 4. VẬY "MỞ CỔNG" TRÊN UFW NGHĨA LÀ GÌ?**

🔓 Bạn **cho phép** người ngoài (như Chrome) **gửi yêu cầu đến máy Ubuntu thông qua các cổng** này.

**Ví dụ:**

Khi bạn chạy lệnh:

sudo ufw allow 'Nginx HTTP'

* Bạn đang mở **cổng 80** cho dịch vụ HTTP
* Máy khác (như máy thật, Chrome...) có thể gửi yêu cầu web vào máy Ubuntu server

Nếu không mở cổng 80:

* Trình duyệt sẽ **không thể truy cập web được**
* Giống như **khách đến quán nhưng cửa bị khóa**

**📦 5. NÓ LIÊN QUAN GÌ ĐẾN NGINX?**

**NGINX** là phần mềm **lắng nghe (listen)** ở các cổng như **80, 443** để **phục vụ web**.

* Nếu **NGINX mở sẵn cửa (port 80), nhưng UFW (tường lửa) chặn lại**, thì vẫn không ai vào được.
* Nên bạn phải **mở tường lửa và khởi động NGINX**, thì khách mới vào xem web được.

**✅ 6. TÓM TẮT DỄ HIỂU**

| **Hành động** | **Ý nghĩa thực tế** |
| --- | --- |
| sudo ufw allow 'Nginx HTTP' | Mở cổng 80 cho người khác xem web (http) |
| sudo ufw allow 'Nginx HTTPS' | Mở cổng 443 cho web bảo mật (https) |
| Cổng bị chặn | Trình duyệt sẽ không vào web được |
| Cổng mở | Trình duyệt gõ IP → thấy trang web thành công |

**🧠 TƯỞNG TƯỢNG CỤ THỂ NHƯ ĐỜI THỰC**

Giả sử bạn mở một **quán cafe web** (Ubuntu Server):

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| Quán cafe = Ubuntu server |  |
| Nhân viên = NGINX |  |
| Khách = Chrome, trình duyệt |  |
| Cửa chính = cổng 80 (http) |  |
| Bảo vệ = tường lửa UFW |  |

👉 Nếu bạn **không cho bảo vệ mở cửa**, dù nhân viên bên trong có sẵn cà phê (file HTML), **khách cũng không vào được** để uống.

**🧪 7. CÁCH KIỂM TRA CỔNG ĐANG MỞ**

Bạn có thể chạy:

sudo ufw status

Sẽ thấy như sau:

Status: active

To Action From

-- ------ ----

Nginx HTTP ALLOW Anywhere

Nginx HTTPS ALLOW Anywhere

👉 Nghĩa là cổng 80 và 443 đã mở, **ai cũng có thể truy cập website của bạn**.

**🧩 HỎI THÊM: CÓ CẦN MỞ CẢ HAI CỔNG KHÔNG?**

* **Chỉ cần mở cổng 80** nếu bạn mới chỉ dùng web thông thường (http)
* Nếu sau này bạn dùng **web bảo mật (https)** → thì mới cần mở thêm cổng 443

**📚 KẾT LUẬN NGẮN GỌN**

| **Mở cổng 80** | **Cho phép trình duyệt (http) vào xem web** |
| --- | --- |
| Mở cổng 443 | Cho phép trình duyệt (https) vào xem web an toàn |
| UFW (tường lửa) | Là "bảo vệ cổng", chặn hoặc cho phép truy cập đến máy Ubuntu Server |
| NGINX | Phần mềm đón tiếp khách, đứng sau cổng 80 và 443 |

Bạn đang hiểu **rất sát gốc rễ**, nếu bạn còn muốn tìm hiểu tiếp:

* Cách chuyển từ http sang https
* Cách dùng domain thay vì IP
* Cách kiểm tra NGINX đang nghe ở cổng nào

Cứ nói: “Mình muốn học tiếp về…” – mình sẽ **giải thích dễ hiểu, từ con số 0** như bây giờ nha! 💪🌱

Việc mở cổng 80 rồi 443 trên tường lửa tức là sao?

**ChatGPT said:**

Việc **mở cổng 80 và 443** trên tường lửa (ví dụ như UFW) nghĩa là bạn **cho phép lưu lượng truy cập từ bên ngoài** (tức là từ Internet) đi vào máy chủ thông qua hai cổng này.

**🔹 Cụ thể:**

| **Cổng** | **Dịch vụ** | **Mô tả** |
| --- | --- | --- |
| **80** | **HTTP** | Giao thức truyền tải trang web không mã hóa. Dùng cho các trang web thông thường. |
| **443** | **HTTPS** | Giao thức truyền tải trang web **có mã hóa** (bảo mật hơn). Dùng cho các trang web bảo mật với SSL/TLS. |

**🔹 Khi bạn “mở cổng” nghĩa là:**

* Tường lửa sẽ **không chặn** các kết nối đến cổng đó.
* Ví dụ, khi người dùng truy cập vào website của bạn bằng trình duyệt (qua HTTP/HTTPS), các yêu cầu sẽ đến máy chủ qua cổng 80 hoặc 443.
* Nếu các cổng này bị **chặn**, trình duyệt sẽ **không thể kết nối**, và người dùng sẽ **không thấy trang web**.

**🔹 Minh họa đơn giản:**

Giả sử bạn có một server chạy web (Apache hoặc Nginx):

| **Trạng thái cổng** | **Kết quả** |
| --- | --- |
| 🔓 Cổng 80 mở | Người dùng có thể truy cập trang web qua http://yourdomain.com |
| 🔓 Cổng 443 mở | Người dùng có thể truy cập trang web qua https://yourdomain.com (an toàn hơn) |
| 🔒 Cổng 80/443 bị chặn | Người dùng **không truy cập được** vào trang web |

**🔹 Lệnh mở cổng với UFW:**

sudo ufw allow 80 # Mở cổng HTTP

sudo ufw allow 443 # Mở cổng HTTPS

**🔹 Kết luận:**

✅ **Mở cổng 80 và 443** là **bắt buộc** nếu bạn muốn triển khai một **web server** mà người dùng từ bên ngoài có thể truy cập.

Nếu bạn có server mà không mở cổng này thì dù webserver có chạy, bên ngoài cũng không thể truy cập được.

Bạn muốn mình kiểm tra cấu hình UFW mẫu cho web server không?

# **HIỂN THỊ WEBSITE**

Rất vui được hướng dẫn bạn! Bây giờ Nginx đã được cài đặt và chạy, vai trò của nó sẽ trở nên rõ ràng hơn thông qua việc bạn **thay thế trang web mặc định bằng nội dung của riêng mình.**

Đây là bài tập minh họa giúp bạn thấy vai trò của **Ubuntu Server** (nơi lưu trữ) và **Nginx** (người phục vụ):

**Bài Tập Minh Họa: Xuất Bản Trang Web Cá Nhân**

Mục tiêu của bài tập này là thay thế trang **"Welcome to nginx!"** bằng một tập tin HTML đơn giản do chính bạn tạo ra.

**1. Vai trò của Ubuntu Server và Nginx**

| Thành Phần | Vai Trò (Tác Dụng) |
| --- | --- |
| **Ubuntu Server** | Là **"tủ hồ sơ"** nơi chứa tất cả các tập tin trang web của bạn. |
| **Nginx** | Là **"người bồi bàn"** biết chính xác vị trí của tập tin đó và phục vụ nó cho khách hàng (trình duyệt web). |
| **Thư mục**/var/www/html | Đây là **vị trí mặc định** mà Nginx tìm kiếm các tập tin trang web. |

**2. Các Bước Thực Hiện**

Bạn sẽ tạo một tập tin HTML mới và đưa nó vào thư mục mà Nginx đang "theo dõi".

**Bước 1: Di chuyển đến Thư mục Web**

Trong Linux, /var/www/html/ là nơi chứa các tập tin web công cộng.

Gõ lệnh sau để chuyển đến thư mục này:

Bash

**cd /var/www/html/**

**Bước 2: Kiểm tra Tập tin Mặc định**

Bạn sẽ thấy tập tin HTML tạo ra trang "Welcome to nginx!".

Gõ lệnh sau để liệt kê các tập tin:

Bash

**ls -l**

Bạn sẽ thấy một tập tin có tên là index.nginx-debian.html.

**Bước 3: Tạo Tập tin Trang Web Mới**

Chúng ta sẽ tạo một tập tin mới có tên là index.html bằng trình soạn thảo nano. Tập tin index.html là tên mặc định mà Nginx sẽ tìm kiếm đầu tiên.

Gõ lệnh sau để mở trình soạn thảo nano với quyền quản trị:

Bash

**sudo nano index.html**

Sau khi màn hình soạn thảo hiện ra, hãy **gõ hoặc dán** đoạn mã HTML đơn giản sau:

HTML

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Chao Mung!</title>**

**</head>**

**<body>**

**<h1>CHUC MUNG! BAN DA THANH CONG XUAT BAN WEB!</h1>**

**<p>Trang web nay dang chay tren Ubuntu Server nho Nginx.</p>**

**<p>Nguoi tao: lebao</p>**

**</body>**

**</html>**

**Bước 4: Lưu và Thoát khỏi**nano

1. Nhấn **Ctrl + O** (Write Out) để lưu.
2. Nhấn **Enter** để xác nhận tên tập tin (index.html).
3. Nhấn **Ctrl + X** để thoát khỏi trình soạn thảo.

**Bước 5: Kiểm tra Kết quả trên Trình Duyệt**

Mở lại trình duyệt trên máy tính vật lý của bạn và truy cập địa chỉ IP:

**http://192.168.101.133**

**Kết quả:** Trình duyệt sẽ hiển thị trang web mới của bạn với dòng chữ **"CHUC MUNG! BAN DA THANH CONG XUAT BAN WEB!"**

**Vai trò Rõ Ràng của Ubuntu Server và Nginx**

* **Ubuntu Server:** Đã cung cấp môi trường (thư mục /var/www/html) để bạn **lưu trữ** tập tin index.html của mình.
* **Nginx:** Đã tự động phát hiện tập tin index.html mới của bạn và **phục vụ** (gửi) nó cho trình duyệt thay vì trang mặc định cũ.

Bạn đã hoàn thành một chu trình phát triển web cơ bản: **Tạo -> Lưu -> Xuất bản -> Truy cập.**

# **Bài tập 1: Sử dụng curl để lấy thông tin các quốc gia và thủ đô của họ trên toàn thế giới với URL: https://countriesnow.space/api/v0.1/countries/capital**

## **1. Tổng quan về Lệnh curl và API**

* curl: Là viết tắt của **C**lient **URL**. Đây là một công cụ dùng để **truyền dữ liệu** đến hoặc từ một máy chủ, sử dụng các giao thức khác nhau (như HTTP, HTTPS, FTP, v.v.). Vai trò chính của nó là giả lập trình duyệt hoặc ứng dụng để giao tiếp với các dịch vụ mạng.
* **API** (Application Programming Interface): Là một "giao diện lập trình ứng dụng." Trong trường hợp này, đường dẫn https://countriesnow.space/api/v0.1/countries/capital là một **API** cung cấp dữ liệu về quốc gia và thủ đô. Khi bạn dùng curl truy cập, API sẽ trả về dữ liệu đó (thường ở định dạng **JSON**).

## **2. Các Bước Thực hiện Bài tập**

**Bước 1: Kiểm tra**curl**đã được cài đặt chưa**

Trên các bản Ubuntu Server mới, curl thường đã được cài đặt sẵn. Tuy nhiên, bạn nên kiểm tra để đảm bảo:

Gõ lệnh sau và nhấn **Enter**:

Bash

**curl --version**

* **Nếu lệnh hiển thị phiên bản** (ví dụ: curl 7.81.0...), bạn đã sẵn sàng.
* **Nếu báo lỗi**command not found, bạn cần cài đặt nó:

Bash

**sudo apt update**

**sudo apt install curl**

**Bước 2: Sử dụng**curl**để Lấy Dữ liệu**

Bây giờ, bạn sẽ dùng curl để gửi yêu cầu đến API.

Gõ chính xác lệnh sau và nhấn **Enter**:

Bash

**curl https://countriesnow.space/api/v0.1/countries/capital**

**Bước 3: Phân tích Kết quả**

Ngay lập tức, Terminal của bạn sẽ hiển thị một lượng lớn dữ liệu.

**Kết quả bạn thấy:**

* Đó là dữ liệu thô (raw data) được API trả về, ở định dạng **JSON** (JavaScript Object Notation).
* Dữ liệu này chứa một danh sách (array) các đối tượng, mỗi đối tượng có các trường như "name", "iso2", "iso3" và quan trọng nhất là "capital".
* Ví dụ bạn sẽ thấy các đoạn như:

JSON

**{"name":"Viet Nam","iso3":"VNM","iso2":"VN","capital":"Ha Noi"}**

**Nếu bạn muốn hiển thị mỗi đối tượng trên một dòng, trong Linux, công cụ jq được sử dụng với tham số -c và lặp qua mảng ‘.[]**

## **3. Các Biến thể Lệnh curl Nâng cao (Tùy chọn)**

Để làm cho bài tập trở nên rõ ràng và dễ đọc hơn, bạn có thể sử dụng các tùy chọn sau:

**a) Hiển thị Thông tin Đầu trang (Headers)**

Lệnh này giúp bạn thấy các thông tin kỹ thuật mà máy chủ trả về (ví dụ: loại dữ liệu, tình trạng kết nối).

Bash

**curl -i https://countriesnow.space/api/v0.1/countries/capital**

* **Tác dụng của**-i**:** Hiển thị cả HTTP headers (như HTTP/1.1 200 OK - nghĩa là thành công) và dữ liệu.

**b) Chỉ Lấy Dữ liệu (Không cần Thanh trạng thái)**

Nếu bạn không muốn thấy thanh tiến trình (progress bar) khi tải dữ liệu.

Bash

**curl -s https://countriesnow.space/api/v0.1/countries/capital**

* **Tác dụng của**-s**:** (Silent) Ẩn thanh tiến trình và thông báo lỗi.

**c) Lưu Kết quả vào Tập tin**

Nếu bạn muốn lưu dữ liệu JSON vào một tập tin để xử lý sau:

Bash

**curl https://countriesnow.space/api/v0.1/countries/capital > country\_capitals.json**

* **Tác dụng của**>**:** Đây là ký hiệu **chuyển hướng** (redirection). Nó không hiển thị kết quả ra màn hình mà ghi thẳng vào tập tin country\_capitals.json trong thư mục hiện tại.

**Tóm lại, thông qua bài tập này, bạn đã học được cách sử dụng Ubuntu Server và lệnh**curl**để lấy dữ liệu từ một dịch vụ web (API) – một kỹ năng cơ bản và quan trọng trong lĩnh vực phát triển và quản trị hệ thống.**

# **XÓA PHẦN MỀM**

A screenshot of a computer program

AI-generated content may be incorrect.

# **THIẾT LẬP THỜI GIAN**

**timedatectl**

A close-up of a phone number

AI-generated content may be incorrect.

Nãy giờ mình cài máy chủ nginx là gì? dùng để làm gì? có tác dụng gì?

Đây là một câu hỏi rất hay! Nãy giờ bạn đã thực hiện một công việc quan trọng: **cài đặt và kích hoạt máy chủ web Nginx.**

Dưới đây là tóm tắt đơn giản về Nginx, vai trò và tác dụng của nó trong bối cảnh bạn vừa thực hành:

**Nginx là gì? (Phát âm: Engine-X)**

Nginx là một phần mềm **máy chủ web (web server)** phổ biến, miễn phí và mã nguồn mở. Nó được thiết kế để xử lý lượng truy cập lớn một cách hiệu quả, giúp trang web tải nhanh và ổn định.

Nginx là một trong những máy chủ web được sử dụng rộng rãi nhất trên thế giới, đứng sau nhiều trang web lớn.

**Nginx Dùng để Làm gì? (Vai trò của Web Server)**

Hãy tưởng tượng Nginx là một **người tiếp tân kiêm bồi bàn** cực kỳ nhanh nhẹn và chuyên nghiệp trong một nhà hàng (máy chủ của bạn).

1. **Lắng nghe Yêu cầu:** Nginx liên tục "lắng nghe" (qua **cổng 80** mà bạn đã mở trên tường lửa) các yêu cầu từ internet.
2. **Nhận Yêu cầu:** Khi bạn gõ địa chỉ IP (http://192.168.101.133) vào trình duyệt, trình duyệt của bạn gửi một **yêu cầu** đến máy chủ. Nginx là người đầu tiên nhận được yêu cầu đó.
3. **Xử lý và Tìm kiếm:** Nginx phân tích yêu cầu, tìm kiếm tập tin trang web tương ứng (trong trường hợp của bạn là tập tin hiển thị chữ "Welcome to nginx!").
4. **Phản hồi:** Nginx **gửi** tập tin trang web đó trở lại trình duyệt của bạn.
5. **Hiển thị:** Trình duyệt nhận được dữ liệu và hiển thị nó dưới dạng trang web mà bạn thấy.

**Tác dụng của Việc Cài đặt Nginx**

Việc cài đặt Nginx trong môi trường thực hành của bạn có ba tác dụng chính:

**1. Phục vụ Trang Web (Web Serving)**

* **Tác dụng cốt lõi:** Biến máy ảo Ubuntu của bạn từ một "chiếc hộp đen" chỉ dùng dòng lệnh thành một **điểm truy cập trên mạng** có khả năng phục vụ nội dung (HTML, hình ảnh, CSS, v.v.).

**2. Mô phỏng Môi trường Thực tế**

* **Tác dụng học tập:** Bạn đang mô phỏng chính xác cấu hình cơ bản của hầu hết các máy chủ web chuyên nghiệp ngoài đời thực. Server không thể hoạt động nếu thiếu một web server như Nginx hoặc Apache.

**3. Cổng vào cho Ứng dụng Web**

* **Tác dụng tương lai:** Nginx là lớp đầu tiên mà mọi người dùng tương tác. Sau này, khi bạn viết các ứng dụng web bằng ngôn ngữ lập trình (như Python, Node.js, PHP), Nginx sẽ đóng vai trò là **Reverse Proxy** (Máy chủ proxy ngược), nhận yêu cầu từ người dùng và chuyển nó một cách an toàn đến ứng dụng của bạn, giúp ứng dụng hoạt động hiệu quả và bảo mật hơn.

Nói tóm lại, **Nginx là phần mềm không thể thiếu** để trang web của bạn có thể hiển thị được trên Internet!

**Bạn có muốn học cách thay đổi trang web mặc định này bằng trang HTML của riêng bạn không?**

sau khi cài đặt xong nginx trên ubuntu server từ rất lâu rồi, thì bật nó lên như thế nào? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức: giải thích từng khái niệm một cách chi tiết, đơn giản, dễ hiểu, dành riêng cho người bắt đầu từ con số 0 như bạn nhé.

**ChatGPT said:**

Chắc chắn rồi! Mình sẽ hướng dẫn bạn cách **bật Nginx sau khi cài đặt xong trên Ubuntu Server**, từng bước một, giải thích đơn giản, dễ hiểu, **dành cho người mới bắt đầu hoàn toàn, không có kiến thức kỹ thuật** nhé.

**💡 Trước hết: Nginx là gì?**

**Nginx (đọc là "en-jin-x")** là một **phần mềm máy chủ web** – hiểu đơn giản là nó giúp **máy tính của bạn (Ubuntu Server)** có thể **phục vụ các trang web** cho người khác truy cập.

* Bạn có thể tưởng tượng nó giống như **một người phục vụ nhà hàng**. Khi ai đó (người dùng web) yêu cầu một món ăn (truy cập trang web), thì **Nginx sẽ lấy món ăn đó từ bếp (máy chủ) và mang ra cho người đó**.

**✅ Giờ thì: Làm sao để “bật” Nginx?**

Sau khi bạn đã cài đặt xong Nginx, bạn cần **“khởi động”** nó (tức là nói với hệ thống: “Hãy chạy chương trình Nginx đi”). Trong thế giới của Ubuntu Server, thao tác này gọi là **start (bắt đầu)** dịch vụ.

**📦 Dịch vụ là gì?**

Một **dịch vụ (service)** là một chương trình chạy **ngầm trong nền** (background), không cần bạn mở cửa sổ ra.

* Ví dụ: Nginx là một dịch vụ web — nó chạy ngầm và chờ có ai đó truy cập vào website thì mới phục vụ nội dung.

**🔧 Công cụ điều khiển dịch vụ: systemctl**

Trên Ubuntu, bạn dùng một công cụ gọi là **systemctl** để điều khiển các dịch vụ. Bạn có thể:

* start – bắt đầu chạy dịch vụ.
* stop – dừng dịch vụ.
* restart – khởi động lại.
* status – kiểm tra xem nó đang chạy không.

**🔥 Bây giờ bạn hãy mở Terminal lên (nơi bạn gõ lệnh) và gõ:**

sudo systemctl start nginx

**Giải thích:**

* sudo: tạm thời cấp quyền quản trị (như admin).
* systemctl: công cụ quản lý dịch vụ.
* start: bắt đầu chạy dịch vụ.
* nginx: tên của dịch vụ bạn muốn chạy.

➡️ Lệnh này sẽ **bật Nginx lên**.

**✅ Kiểm tra xem Nginx đã bật chưa:**

Gõ lệnh sau:

sudo systemctl status nginx

Bạn sẽ thấy một số dòng, ví dụ:

● nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since...

Nếu bạn thấy dòng Active: active (running) thì nghĩa là Nginx đang chạy! 🎉

**💡 Để Nginx tự bật mỗi lần khởi động máy:**

Gõ:

sudo systemctl enable nginx

Giải thích:

* enable: bật chế độ **tự động khởi động khi máy khởi động**.

**🚀 Kiểm tra Nginx có hoạt động không?**

Sau khi bật, bạn có thể kiểm tra bằng cách:

1. Trên trình duyệt (Google Chrome, Firefox, v.v.), gõ địa chỉ IP của máy Ubuntu Server.
2. Nếu hiện ra dòng chữ **"Welcome to nginx!"** thì bạn đã thành công rồi!

**✅ Tóm tắt lại từng bước đơn giản:**

| **Bước** | **Lệnh gõ vào terminal** | **Giải thích ngắn** |
| --- | --- | --- |
| 1 | sudo systemctl start nginx | Bật Nginx |
| 2 | sudo systemctl status nginx | Kiểm tra xem Nginx đang chạy không |
| 3 | sudo systemctl enable nginx | Tự động bật Nginx khi khởi động máy |

Nếu bạn thấy dòng nào báo lỗi hoặc không chắc chắn về bước nào, bạn cứ **chụp ảnh màn hình hoặc copy nội dung ra đây**, mình sẽ giúp bạn từng chút một, không sao cả nhé. 🧡

Bạn muốn mình giải thích tiếp về **cách xem trang web trên Nginx**, hoặc **cách thay đổi nội dung trang web mặc định** không?

**You said:**

Hãy giải thích chi tiết cụ thể cặn kẽ, nguồn gốc gốc gác tại sao lại phải cài đặt nginx để làm gì? nó là gì? có tác dụng và vai trò gì? Dùng để làm gì? Có mối quan hệ gì với ubuntu server cài trên vmware? Và tại sao phải cài đặt nó trên unbuntu server mà không phải là máy windows thật hay gì đó? Và lý do tại sao phải nhập địa chỉ ip các thứ? Hãy giải thích cội nguồn từ con số 0 cho mình hiểu. hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức: giải thích từng khái niệm một cách chi tiết, đơn giản, dễ hiểu, dành riêng cho người bắt đầu từ con số 0 như bạn nhé.

**ChatGPT said:**

Câu hỏi của bạn rất tuyệt vời — bởi vì nó cho thấy bạn **muốn hiểu cặn kẽ tận gốc**, chứ không chỉ làm theo hướng dẫn. Mình sẽ đi từng phần **rất chi tiết**, theo cách **đơn giản nhất, dễ hiểu nhất**, **dành riêng cho người bắt đầu từ con số 0**, như bạn yêu cầu nha. ☘️

**📌 1. Trước tiên: Tại sao lại cài đặt Nginx?**

**🔸 Vì bạn muốn tạo ra một website hoặc một ứng dụng web, đúng không?**

Muốn có một website mà người khác truy cập được qua trình duyệt (Chrome, Firefox...), bạn **phải có một "máy chủ" (server)**.

**📌 2. Vậy: Máy chủ (Server) là gì?**

* **Server** là một máy tính đặc biệt **luôn bật**, luôn hoạt động, có nhiệm vụ **phục vụ yêu cầu từ người dùng**.
* Khi bạn mở trình duyệt và gõ vào địa chỉ web (ví dụ: facebook.com), thì trình duyệt **gửi yêu cầu đến một máy chủ**, rồi máy chủ đó **trả về dữ liệu**, ví dụ: trang chủ Facebook.

**Ví dụ:**

* Bạn = người khách.
* Trình duyệt = nhân viên nhà hàng ghi món.
* Server = đầu bếp (phục vụ món).
* Nginx = hệ thống bếp chuyên nghiệp giúp nấu nhanh hơn, dễ hơn.

**📌 3. Vậy Nginx là gì?**

**Nginx là một phần mềm giúp máy chủ phục vụ nội dung web cho người dùng.**

**Nói cách khác:**

* Nginx là **“người đứng giữa”** kết nối giữa người truy cập web và máy chủ (server).
* Nginx **nhận yêu cầu** từ trình duyệt, rồi **gửi lại trang web hoặc dữ liệu** về cho người dùng.

**📌 4. Tác dụng của Nginx là gì?**

| **Tác dụng** | **Giải thích** |
| --- | --- |
| Phục vụ trang web | Hiển thị nội dung trang web khi ai đó truy cập địa chỉ IP hoặc tên miền |
| Làm cổng vào cho các ứng dụng | Nếu bạn có ứng dụng web (NodeJS, Python Flask, PHP…), Nginx giúp người dùng truy cập vào |
| Chuyển tiếp yêu cầu | Nginx có thể “chuyển hướng” yêu cầu đến các dịch vụ khác bên trong máy chủ |
| Làm cân bằng tải | Nếu bạn có nhiều máy chủ con, Nginx chia đều người dùng ra để không bị quá tải |

**📌 5. Vì sao cài Nginx trên Ubuntu Server, không phải Windows?**

**Trả lời ngắn:**

👉 Vì Ubuntu Server là **hệ điều hành chuyên dùng để làm máy chủ**, nhẹ, miễn phí, ổn định, mạnh mẽ hơn Windows khi dùng cho máy chủ.

**Giải thích đơn giản:**

| **Windows** | **Ubuntu Server** |
| --- | --- |
| Dành cho người dùng phổ thông | Dành cho máy chủ, chuyên dùng cho web/server |
| Nặng, có giao diện đồ họa | Nhẹ, không giao diện (chỉ dòng lệnh), tiết kiệm tài nguyên |
| Không tối ưu cho web server | Rất tối ưu cho web server |
| Có thể cài web server, nhưng không phổ biến | Hầu hết web server trên thế giới dùng Ubuntu (hoặc Linux khác) |

📌 Vì vậy: **Nếu bạn đang học làm website hay app, thì cài Nginx trên Ubuntu là cách tiếp cận đúng, giống như người chuyên nghiệp làm.**

**📌 6. Vậy tại sao lại dùng Ubuntu Server trên VMware, chứ không cài trực tiếp trên máy?**

**Trước tiên: VMware là gì?**

👉 **VMware** là một phần mềm giúp bạn **tạo ra một “máy tính ảo” bên trong máy tính thật.**

Giống như:

* Máy thật: là cái laptop bạn đang dùng.
* VMware: là phần mềm “tạo ra” một máy tính khác bên trong máy đó.

**Lý do bạn dùng VMware:**

| **Vì sao dùng VMware** | **Giải thích** |
| --- | --- |
| Không ảnh hưởng máy thật | Bạn thử nghiệm thoải mái, nếu hư thì xóa, cài lại |
| Giả lập máy chủ | Bạn đang học làm web/server, thì cần 1 môi trường riêng biệt như máy chủ thật |
| Tiện lợi | Dễ dùng, không cần mua máy chủ thật |

**Ubuntu Server trên VMware = Máy chủ ảo trong máy tính của bạn**

* Khi bạn cài Ubuntu Server trong VMware là bạn đang **tạo ra một máy chủ thật sự, nhưng chạy ảo**.
* Bạn có thể cài Nginx lên đó và tập làm web như thật.

**📌 7. Tại sao phải nhập địa chỉ IP?**

**Trả lời đơn giản:**

👉 Vì **mỗi máy tính đều có địa chỉ IP**, giống như **số nhà**, để người khác tìm và kết nối được.

**Trong tình huống của bạn:**

* Bạn có:
  + Máy thật: laptop.
  + Máy ảo: Ubuntu Server trong VMware.
* Máy ảo có một địa chỉ IP riêng (như một máy khác trong mạng).

➡️ Khi bạn mở trình duyệt trên máy thật, bạn **nhập địa chỉ IP của máy ảo**, thì **trình duyệt sẽ truy cập vào Nginx đang chạy trên Ubuntu Server đó.**

**📌 8. Tóm lại: Toàn bộ bức tranh lớn là gì?**

| **Thành phần** | **Vai trò** |
| --- | --- |
| Ubuntu Server (trong VMware) | Là “máy chủ” của bạn |
| Nginx | Phần mềm giúp máy chủ phục vụ website/app |
| VMware | Phần mềm tạo máy ảo để bạn học mà không làm hỏng máy thật |
| Địa chỉ IP | Giúp các máy (thật và ảo) kết nối với nhau |
| Trình duyệt web (Chrome...) | Công cụ bạn dùng để truy cập và xem web |
| Bạn | Là lập trình viên tương lai 😄, đang học cách xây dựng hệ thống web |

**🎁 Hình ảnh minh họa (bằng lời)**

[Bạn (máy thật)] --> (Nhập IP máy ảo vào trình duyệt)

↓

[VMware (chạy máy Ubuntu Server)]

↓

[Nginx đang chạy bên trong]

↓

[Phục vụ nội dung web cho bạn]

**✨ Bạn đã hiểu được tận gốc vấn đề rồi đó!**

Nếu bạn muốn:

* Biết cách thay nội dung web hiển thị bởi Nginx.
* Tạo 1 website đầu tiên.
* Cài thêm ứng dụng phía sau Nginx (NodeJS, Flask…).

👉 Mình có thể hướng dẫn tiếp **từng bước cực dễ hiểu** nhé!

Bạn muốn làm gì tiếp theo nào? 😊

**You said:**

ubuntu server cũng là máy chủ mà? tại sao lại phải tải thêm nginx về? và tại sao phải gõ địa chỉ ip của ubuntu server? sao không gõ địa chỉ ip của nginx? rối rắm quá huhu vẫn không hình dung được cái trình tự mối liên hệ, quan hệ vai trò giữa chúng, kiểu phải hiểu nguồn gốc, gốc gác nó như nào, kiểu phải học từ phép cộng trừ rồi mới đến nhân chia các thứ chứ nhỉ hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức: giải thích từng khái niệm một cách chi tiết, đơn giản, dễ hiểu, dành riêng cho người bắt đầu từ con số 0 như bạn nhé.

**ChatGPT said:**

Bạn ơi, đừng lo lắng nha! Câu hỏi của bạn rất hay và cũng **rất đúng hướng tư duy** – bởi vì bạn muốn **hiểu bản chất gốc rễ của vấn đề**, chứ không chỉ “làm theo mà không biết tại sao”.

Và đúng như bạn nói: "Phải học cộng trừ trước khi nhân chia" – nên bây giờ mình sẽ **giải thích từng khái niệm nhỏ một cách thật đơn giản, dễ hiểu, dễ hình dung**, như là mình đang ngồi kế bên bạn, chỉ tận tay luôn nha! 💡

**🧠 I. Khái niệm cơ bản: Máy chủ là gì?**

**💻 Máy tính & Máy chủ:**

**👉 Bạn có 2 kiểu máy:**

| **Loại máy** | **Là gì?** |
| --- | --- |
| **Máy tính (client)** | Là máy bạn dùng hằng ngày (laptop, điện thoại), để **truy cập thông tin** |
| **Máy chủ (server)** | Là máy chuyên phục vụ người khác. Nó **chứa dữ liệu, website, ứng dụng** để client truy cập |

**❓Vậy "Ubuntu Server" là gì?**

* **Ubuntu Server** là **một hệ điều hành** (giống như Windows, macOS, v.v.)
* Nhưng nó được thiết kế **chuyên để làm máy chủ**.
* Nó không phải là **cái server hoàn chỉnh luôn**, mà **giống như "cái nền nhà" để bạn xây cái server lên**.

➡️ **Ubuntu Server = nền tảng**  
➡️ Còn muốn nó phục vụ web, bạn phải **cài thêm phần mềm máy chủ web** → **chính là NGINX**.

**🏗️ II. Vì sao phải cài thêm NGINX?**

**📦 Ubuntu Server chỉ là nền tảng hệ điều hành**

Giống như:

* Bạn mua một căn nhà trống (Ubuntu Server).
* Nhưng để nấu ăn, bạn cần bếp (NGINX).
* Nếu bạn muốn bán cà phê, bạn cần máy pha cà phê (phần mềm khác: MySQL, NodeJS, v.v.)

👉 Ubuntu Server KHÔNG TỰ ĐỘNG có sẵn chức năng “phục vụ web” đâu nha. Nó chỉ là nền móng.

Muốn **ai đó truy cập được trang web** → bạn phải cài thêm phần mềm như **NGINX** để làm “bếp trưởng phục vụ món web”.

**🔁 III. Vai trò – Mối quan hệ giữa các thành phần**

**🧩 Toàn bộ hệ thống của bạn có 4 phần:**

| **Thành phần** | **Vai trò** | **Giống như** |
| --- | --- | --- |
| **VMware** | Tạo ra máy ảo Ubuntu Server | Một “nhà riêng” trong căn nhà thật |
| **Ubuntu Server** | Hệ điều hành trong máy ảo | Nền nhà trong căn nhà đó |
| **NGINX** | Phần mềm phục vụ web | Bếp trưởng nhận món khách gọi |
| **Địa chỉ IP** | Địa chỉ để tìm đến nhà bạn | Số nhà để đưa khách đến đúng nơi |

**🎯 Mối quan hệ:**

[Trình duyệt Chrome trên máy bạn]

↓ (truy cập địa chỉ IP)

[Ubuntu Server đang chạy trên VMware]

↓ (Ubuntu chứa phần mềm NGINX)

[NGINX nhận yêu cầu, trả về nội dung web]

**🤔 Vậy tại sao lại gõ IP của Ubuntu Server chứ không phải của NGINX?**

**🧠 Giải thích ngắn gọn:**

* **IP là của máy (Ubuntu Server)**, không phải của phần mềm bên trong.
* NGINX là **phần mềm chạy bên trong Ubuntu**, nó **không có IP riêng**.
* Khi bạn gõ IP của Ubuntu, bạn đang gõ:  
  👉 “Tôi muốn nói chuyện với cái máy đó”
* Máy đó (Ubuntu Server) **đang có NGINX lắng nghe yêu cầu web** trên cổng 80 → nên nó phản hồi bằng trang web.

**🎯 Ví dụ thực tế:**

| **Ví dụ đời thường** | **So sánh trong máy tính** |
| --- | --- |
| Bạn đến nhà bạn A (số nhà: 192.168.1.100) | Truy cập IP Ubuntu Server |
| Bạn bấm chuông gọi người trong nhà | Trình duyệt gửi yêu cầu HTTP |
| Trong nhà có ông NGINX ra mở cửa | NGINX nhận yêu cầu |
| Ông ấy đưa bạn ly nước chanh | NGINX trả về nội dung web |

Bạn **không gọi ông NGINX** trực tiếp được — mà phải **gõ địa chỉ nhà (IP)** nơi ông ấy đang sống (Ubuntu Server).

**🧩 Tóm tắt theo trình tự "cộng trừ nhân chia"**

| **Giai đoạn** | **Việc cần làm** | **Giải thích đơn giản** |
| --- | --- | --- |
| **1. Cài VMware** | Để tạo máy ảo | Giống như xây nhà riêng trong máy thật |
| **2. Cài Ubuntu Server vào máy ảo** | Để làm nền hệ điều hành | Giống như lắp sàn, tường, điện |
| **3. Cài NGINX vào Ubuntu** | Để biến nó thành máy chủ web | Giống như lắp bếp để nấu ăn |
| **4. Lấy IP của Ubuntu Server** | Để truy cập từ máy thật | Giống như biết số nhà để ghé thăm |
| **5. Nhập IP vào trình duyệt** | Để xem web do NGINX phục vụ | Bạn gõ đúng số nhà, NGINX mở cửa phục vụ bạn |

**✅ Tóm lại: Bản chất rất đơn giản nếu bạn hiểu như sau:**

**Ubuntu Server = căn nhà rỗng**  
**NGINX = ông bếp trưởng giúp bạn nấu món web**  
**VMware = cách bạn tạo căn nhà riêng biệt này**  
**Địa chỉ IP = số nhà để người khác tìm đến**

**🧶 Hình ảnh minh họa (mô hình mối quan hệ)**

[Máy bạn (Windows thật)]

↕

VMware (Tạo máy ảo)

↕

[Ubuntu Server trong máy ảo]

↕

[Nginx chạy trong Ubuntu]

↕

[Trình duyệt yêu cầu → Nginx trả dữ liệu web]

**🧭 Nếu vẫn còn mơ hồ:**

Bạn có thể hỏi tiếp như:

* "Nội dung trang web nằm ở đâu?"
* "Nginx trả về trang gì?"
* "Làm sao đổi nội dung web trả về?"

Mình sẵn sàng **giải thích từng tí** tiếp cho bạn, đến khi bạn nắm rõ như lòng bàn tay luôn!

Bạn muốn mình vẽ sơ đồ hoặc gửi ví dụ hình ảnh thực tế cho dễ hiểu hơn nữa không? Mình có thể vẽ mô hình giúp bạn thấy rõ luôn nha! 💡

**You said:**

Tôi tưởng nginx là một loại laptop nào đó, chỉ cần mua về và lưu trữ website rồi deloy gì gì đó là được, tôi tưởng nginx có giao diện đồ họa rồi hệ điều hành các thứhãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức: giải thích từng khái niệm một cách chi tiết, đơn giản, dễ hiểu, dành riêng cho người bắt đầu từ con số 0 như bạn nhé.

**ChatGPT said:**

Cảm ơn bạn đã chia sẻ rất chân thật và rõ ràng nha! 🎉  
Câu hỏi của bạn cực kỳ quan trọng, vì **rất nhiều người mới bắt đầu cũng nghĩ như vậy**: tưởng **NGINX là một cái máy tính hay thiết bị gì đó**, như laptop – mua về, cắm điện là chạy website.

Bây giờ, mình sẽ giải thích lại từ đầu, **chậm rãi, đơn giản, dễ hiểu, dễ hình dung**, dành cho **người chưa biết gì về công nghệ**, như bạn yêu cầu nhé. ☀️

**🧠 I. Trước tiên: Nginx KHÔNG PHẢI là một cái laptop hay thiết bị**

**✅ Sự thật:**

**NGINX là một phần mềm** — chứ không phải là một máy tính, không phải là thiết bị, không có màn hình, không có bàn phím.

**❓Vậy thì: NGINX là phần mềm như thế nào?**

**➤ NGINX là một phần mềm máy chủ web (web server software)**

Nói một cách thật đơn giản:

**NGINX giống như một “ông phục vụ”** – khi người dùng từ bên ngoài gõ địa chỉ web vào trình duyệt, ông NGINX sẽ:

* Nhận yêu cầu đó,
* Tìm trang web tương ứng,
* Trả về nội dung cho người dùng xem trên trình duyệt.

**🔥 Vậy tại sao nhiều người tưởng đó là thiết bị?**

Vì trong thế giới thực, những công ty lớn có máy chủ riêng (có thể nhìn thấy được), người ta hay nói:

* “Mua máy chủ để chạy web”
* “Mua server để deploy ứng dụng”

Nhưng cái **máy vật lý đó (máy chủ thật)** chỉ là **phần cứng**, còn NGINX là **phần mềm chạy trên cái máy đó**.

**🧩 II. Tưởng tượng đơn giản: So sánh giữa phần cứng và phần mềm**

| **Bạn tưởng** | **Thực tế là** |
| --- | --- |
| NGINX là 1 cái máy tính đặc biệt để chạy web | ❌ Sai |
| NGINX là 1 phần mềm cài vào **máy tính hoặc máy chủ** | ✅ Đúng |

**🎯 Giống như điện thoại:**

* **Điện thoại** là cái máy vật lý.
* **Zalo, Facebook, TikTok** là phần mềm bạn cài vào để xài.

👉 Cũng như vậy:

* **Máy chủ (hoặc Ubuntu Server)** là cái máy.
* **NGINX** là phần mềm cài vào đó để phục vụ web.

**🔍 III. NGINX có giao diện không?**

❌ Không có giao diện đồ họa như các phần mềm Windows  
✅ NGINX chỉ hoạt động **qua dòng lệnh**, thường dùng trên hệ điều hành **Linux** (như Ubuntu Server)

Nó không có cửa sổ, không có nút bấm đẹp đẽ — nó là công cụ dành cho máy chủ chuyên dụng, chạy "ngầm".

**🧭 IV. Tóm gọn lại: Vai trò, vị trí, và cách dùng NGINX**

| **Thành phần** | **Là gì?** | **Ví dụ dễ hiểu** |
| --- | --- | --- |
| NGINX | Phần mềm máy chủ web | “Ông phục vụ web” |
| Ubuntu Server | Hệ điều hành | “Sàn nhà” để ông NGINX đứng |
| Máy chủ (server) | Máy tính chạy Ubuntu + NGINX | “Ngôi nhà” chứa ông NGINX |
| Trình duyệt (Chrome...) | Công cụ người dùng dùng để truy cập web | “Người khách gọi món” |
| IP Address | Địa chỉ của máy chủ | “Số nhà” để biết tìm đến đâu |

**🏗️ V. Hình ảnh mô hình tổng thể dễ hiểu**

[Trình duyệt trên máy bạn]

↓ (gõ IP máy chủ)

[Ubuntu Server (máy chủ)]

↓ (có cài NGINX)

[NGINX (phần mềm)]

↓

[Phục vụ nội dung trang web]

**📌 VI. Một ví dụ thực tế đời thường**

Giả sử bạn có người bạn tên là Linh đang mở một tiệm bán bánh tại địa chỉ số 25 Nguyễn Văn A.

* Linh = máy chủ (Ubuntu Server)
* Tiệm bánh = ứng dụng web
* Nhân viên phục vụ bánh cho khách = NGINX
* Địa chỉ số 25 Nguyễn Văn A = địa chỉ IP

Bạn đến tiệm, gõ cửa = mở trình duyệt và gõ IP  
Linh mở cửa và nhân viên đưa bánh = Ubuntu + NGINX phản hồi trình duyệt

**✅ Kết luận dễ nhớ:**

| **Câu hỏi của bạn** | **Trả lời dễ hiểu** |
| --- | --- |
| NGINX có phải là laptop không? | ❌ Không. Nó là phần mềm |
| NGINX có hệ điều hành không? | ❌ Không. Nó cần **được cài vào** hệ điều hành khác (như Ubuntu) |
| NGINX dùng để làm gì? | ✅ Giúp trả trang web về cho người dùng |
| Tại sao cần cài NGINX lên Ubuntu Server? | ✅ Vì Ubuntu là nơi để phần mềm như NGINX chạy |
| Tại sao gõ IP của Ubuntu? | ✅ Vì NGINX nằm bên trong Ubuntu – giống như gõ địa chỉ nhà chứ không gọi ông phục vụ riêng |

**🧑‍🏫 Bạn có thể nhớ theo ví dụ sau:**

👉 Máy chủ = Căn nhà

👉 Ubuntu Server = Nền nhà

👉 NGINX = Ông đầu bếp trong nhà

👉 IP Address = Số nhà

👉 Trình duyệt = Khách tới gọi món

Bạn **không gọi đầu bếp trực tiếp**, bạn **tới nhà (IP)**, thì đầu bếp mới phục vụ bạn được.

**❓Bạn muốn tiếp theo là:**

* Cách thay đổi nội dung trang web mà NGINX phục vụ?
* Cách tạo một website nhỏ đầu tiên?
* Cách trỏ tên miền (thay vì phải gõ IP)?

Mình sẵn sàng hướng dẫn tiếp **cực kỳ dễ hiểu**, không cần bạn có kiến thức trước đâu nhé.  
👉 Chỉ cần bạn nói: “Mình muốn tiếp theo học cái gì” – mình sẽ dắt tay bạn đi từng bước 😄

Bạn muốn học tiếp gì sau phần này nào?

Vẫn chưa hiểu lắm? Tại sao cài nginx rồi? sao đó phải gõ địa chỉ của vmware có ubuntu server đã cài trong đó để kiểm tra gì gì đó của nginx ? tức là sao nhỉ? dùng câu lệnh ip a để xem ip của ubuntu server hay của vmware và dùng địa chỉ ip đó, mở chrome gõ trên máy thật rồi nhấn enter để làm gì?hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức: giải thích từng khái niệm một cách chi tiết, đơn giản, dễ hiểu, dành riêng cho người bắt đầu từ con số 0 như bạn nhé.

**ChatGPT said:**

Cảm ơn bạn đã kiên trì nha! ❤️  
Bạn đang rất đúng khi đặt ra câu hỏi như vậy, vì thật sự nó **rất dễ gây nhầm lẫn cho người mới** — đặc biệt khi có nhiều thành phần như: máy thật, máy ảo, Ubuntu Server, Nginx, địa chỉ IP…

👉 Vậy nên mình sẽ **giải thích lại toàn bộ từ đầu**, theo **cách cực kỳ đơn giản, dễ hiểu**, giống như mình đang kể chuyện cho bạn nghe vậy.  
**Chúng ta cùng nhau "vẽ bản đồ tư duy" từ gốc rễ, rồi đi từng bước.**

**🧭 BỐI CẢNH – Bạn đang có những gì?**

**🖥️ 1. Máy thật của bạn: Máy tính Windows**

* Đây là **máy bạn đang dùng hàng ngày**.
* Trên máy này, bạn có cài phần mềm **VMware** để chạy máy ảo.

**🧱 2. Máy ảo: Ubuntu Server đang chạy trong VMware**

* VMware giống như **một “căn nhà ảo” bên trong căn nhà thật**.
* Bên trong căn nhà ảo này, bạn **cài hệ điều hành Ubuntu Server**.
* Ubuntu Server là **một máy chủ** (server), dù nó chỉ đang chạy ảo trong máy thật.

**🔧 3. Bên trong Ubuntu Server, bạn cài thêm phần mềm NGINX**

* NGINX là **phần mềm web server**, giúp Ubuntu Server **có khả năng phục vụ web**.
* NGINX **không có giao diện**, bạn điều khiển nó bằng **dòng lệnh** trong Ubuntu.

**🚦 TỪ ĐÂY: Câu hỏi chính của bạn là...**

"Tại sao sau khi cài NGINX trong Ubuntu Server rồi, mình lại phải chạy lệnh ip a để lấy IP, rồi lên máy thật mở Chrome, gõ IP đó vào để làm gì?"

**✅ CÂU TRẢ LỜI NGẮN GỌN TRƯỚC:**

👉 Là để **kiểm tra xem NGINX đã hoạt động hay chưa**, bằng cách **dùng trình duyệt ở máy thật để truy cập vào máy ảo Ubuntu Server qua IP**.

**👣 BÂY GIỜ GIẢI THÍCH CỤ THỂ – TỪNG KHÁI NIỆM MỘT**

**📌 I. "Nginx là phần mềm web server" nghĩa là gì?**

* Khi bạn gõ địa chỉ web như www.google.com, trình duyệt **gửi yêu cầu đến một máy chủ đang chạy phần mềm như NGINX**.
* NGINX **nhận yêu cầu đó và trả về nội dung web** để bạn thấy được trên trình duyệt.

**📌 II. Vậy bạn cài NGINX vào Ubuntu Server để làm gì?**

👉 Để biến cái máy ảo Ubuntu Server đó thành **một máy chủ web thật sự**.  
Sau này, bạn có thể:

* Đưa website của bạn lên đó.
* Cho người khác truy cập vào từ xa.
* Tập deploy (đưa ứng dụng web lên).

**📌 III. Vậy tại sao lại phải gõ IP của Ubuntu Server?**

👉 Vì để **truy cập vào máy Ubuntu đó từ máy Windows thật**, bạn **phải biết địa chỉ của nó** (IP).  
Giống như muốn ghé nhà bạn bè, bạn cần biết **số nhà** của họ.

**📌 IV. ip a là gì?**

Đây là **một lệnh bạn gõ trong Ubuntu Server** để hỏi:  
“Ê, máy Ubuntu này đang có địa chỉ IP là gì vậy?”

* Nó sẽ hiện ra một loạt dòng.
* Bạn sẽ tìm thấy IP giống như 192.168.1.123 chẳng hạn.

**📌 V. Sau đó, bạn mở Chrome trên máy thật, gõ IP đó vào để làm gì?**

👉 Để xem thử:  
"Mình từ máy thật, có thể **truy cập vào Ubuntu Server** qua địa chỉ IP không?"

Và nếu:

* NGINX đang chạy,
* Ubuntu Server đang bật,
* Máy thật + máy ảo **nằm trong cùng mạng**,

👉 thì bạn sẽ thấy **trang mặc định của NGINX**:

Welcome to nginx!

**🧠 BẠN CÓ THỂ HÌNH DUNG THEO CÁCH NÀY:**

| **Bước** | **Hành động** | **Mục đích** |
| --- | --- | --- |
| 1 | Cài NGINX vào Ubuntu | Để Ubuntu có thể phục vụ web |
| 2 | Gõ ip a trong Ubuntu | Để biết địa chỉ IP của máy Ubuntu |
| 3 | Mở Chrome trên Windows thật | Là trình duyệt để kiểm tra |
| 4 | Gõ IP máy Ubuntu vào Chrome | Để truy cập vào Ubuntu từ bên ngoài |
| 5 | Thấy "Welcome to nginx!" | Nghĩa là NGINX đã hoạt động OK ✅ |

**📷 MỘT VÍ DỤ GIẢ LẬP:**

**Giả sử bạn gõ ip a và thấy Ubuntu Server có IP là:**

192.168.145.130

**Thì bạn làm:**

1. Mở **Chrome trên Windows**
2. Gõ: http://192.168.145.130
3. Nhấn Enter

👉 Nếu thành công, bạn sẽ thấy:

Welcome to nginx!

**🔁 GIỐNG NHƯ CUỘC TRÒ CHUYỆN GIỮA 2 NGƯỜI**

* Máy thật (Windows) = người A
* Máy ảo (Ubuntu Server) = người B
* NGINX = nhân viên trực tổng đài trong nhà của B
* IP = số điện thoại của B

👉 Người A muốn kiểm tra xem nhân viên tổng đài (NGINX) của B có đang trực không → thì cần gọi số điện thoại của B (IP) để kiểm tra.

**✅ KẾT LUẬN RẤT DỄ NHỚ**

| **Bạn hỏi** | **Giải thích dễ hiểu** |
| --- | --- |
| Tại sao cài NGINX rồi còn phải gõ IP? | Vì NGINX là phần mềm **chạy trong Ubuntu**, bạn muốn kiểm tra xem nó có đang chạy không thì **phải truy cập vào máy Ubuntu** qua địa chỉ IP của nó |
| Dùng ip a để làm gì? | Để **tìm ra địa chỉ IP của máy Ubuntu Server** trong máy ảo |
| Gõ IP vào Chrome để làm gì? | Để **kiểm tra xem từ máy thật có thể "nói chuyện" với máy Ubuntu và NGINX hay không** |

**📌 GỢI Ý TIẾP THEO**

Nếu bạn đã thấy màn hình “Welcome to nginx!”, nghĩa là:

🎉 **Bạn đã thành công trong việc:**

* Cài NGINX
* Kết nối từ máy thật đến máy ảo
* Biến Ubuntu thành 1 server đơn giản

👉 Bạn có thể tiếp tục học:

* Cách đổi nội dung web mà NGINX hiển thị
* Cách đưa một website thật lên NGINX
* Cách dùng tên miền thay vì gõ IP

Bạn muốn mình hướng dẫn bước tiếp theo là gì?  
Ví dụ: **“Thay nội dung trang Welcome to nginx thành web của mình”** nhé?

Cứ nói thôi, mình sẽ dắt tay bạn từng bước! 🧡

Thể rốt cuộc addresses và via là của máy windows thật, của máy ảo hay của wifi, modem, router hay gì đó vậy? Mình nontech, chưa học về những cái này nên không hiểu network: version: 2 ethernets: ens33: dhcp4: no addresses: [192.168.1.100/24] routes: - to: default via: 192.168.1.1 nameservers: addresses: [8.8.8.8, 1.1.1.1]

**ChatGPT said:**

Câu hỏi của bạn rất **chuẩn xác và thực tế** – chính xác là điều mà những người mới học Linux và mạng máy tính hay bị **rối và hoang mang**.  
Bây giờ mình sẽ **giải thích chi tiết – tường tận – dễ hiểu như đang kể chuyện**, để bạn **nắm rõ vai trò của addresses, via, DNS…** trong đoạn YAML này, từ **góc nhìn non-tech**.

**🧠 Trước tiên: Đây là đoạn cấu hình mạng trong Ubuntu**

Cụ thể là trong file cấu hình /etc/netplan/50-cloud-init.yaml, dùng để:

✅ Khai báo **cách Ubuntu kết nối mạng**  
✅ Tự đặt địa chỉ IP, cổng kết nối (gateway), DNS...

**🧩 Phân tích từng dòng:**

**✅ 1. addresses: [192.168.1.100/24]**

🔹 **Ý nghĩa:** Gán địa chỉ IP **tĩnh (static)** cho máy Ubuntu

* 192.168.1.100: là **địa chỉ IP mà máy ảo Ubuntu sẽ dùng**
* /24: là **prefix subnet**, tương đương với subnet mask 255.255.255.0

👉 **Đây là IP của Ubuntu**, không phải của máy thật Windows, cũng không phải của router.

📌 Bạn đang nói: "Tôi muốn Ubuntu của tôi là số nhà 100 trong khu phố 192.168.1.x"

**✅ 2. via: 192.168.1.1**

🔹 **Ý nghĩa:** Chỉ định "default gateway" — tức **cửa ngõ để máy Ubuntu gửi dữ liệu ra ngoài**

* 192.168.1.1 thường là **địa chỉ IP của router nhà bạn** (cục Wi-Fi/modem)
* Nó là **cánh cửa chính** để Ubuntu có thể ra Internet

👉 Vậy **via là IP của router**, **không phải của máy Windows thật, cũng không phải máy ảo Ubuntu**.

📌 Tưởng tượng: bạn ra khỏi nhà (máy Ubuntu) để đi chợ (Internet), thì 192.168.1.1 là **cánh cổng nhà bạn (router)**

**✅ 3. nameservers: addresses: [8.8.8.8, 1.1.1.1]**

🔹 **Ý nghĩa:** Chỉ định **DNS Server** dùng để phân giải tên miền (như google.com → thành địa chỉ IP)

* 8.8.8.8: DNS của Google
* 1.1.1.1: DNS của Cloudflare

👉 Đây là địa chỉ của **máy chủ DNS trên Internet** — không liên quan đến máy Windows, máy ảo hay router của bạn.

📌 Máy bạn muốn biết "google.com là IP gì?", thì phải hỏi mấy ông DNS này

**✅ 4. dhcp4: no**

🔹 Nghĩa là **không dùng DHCP**, tức là bạn **không xin IP từ mạng**, mà **tự đặt IP thủ công**

👉 Nếu để dhcp4: yes, thì máy Ubuntu sẽ tự xin IP từ mạng (thường là từ router hoặc VMware)

**📊 Tóm lại:**

| **Thành phần** | **Ý nghĩa** | **Thuộc về thiết bị nào** |
| --- | --- | --- |
| addresses | IP mà máy Ubuntu dùng | 👉 Máy Ubuntu (VMware) |
| via (gateway) | Cổng kết nối để ra Internet | 👉 Router hoặc modem Wi-Fi |
| nameservers | Dịch tên miền → IP | 👉 Máy chủ DNS trên Internet |
| dhcp4 | Có xin IP tự động không | 👉 Nếu yes thì router/VMware cấp IP |

**🌐 Mối quan hệ giữa các thành phần mạng**

[ Ubuntu (VMware) ]

└── IP: 192.168.1.100

└── Gateway: 192.168.1.1 ← (Router / Wi-Fi nhà bạn)

└── DNS: 8.8.8.8, 1.1.1.1 ← (Google, Cloudflare)

[ Máy Windows thật ]

└── Chạy phần mềm VMware

└── Không ảnh hưởng đến IP của máy ảo (nếu dùng mạng NAT/Bridge đúng cách)

[ Router / Modem ]

└── Có IP là 192.168.1.1

└── Là cổng gateway để tất cả máy tính trong nhà ra Internet

**📌 Nếu bạn điền sai addresses hoặc via thì sao?**

| **Sai gì** | **Hậu quả** |
| --- | --- |
| addresses không cùng mạng với gateway | Máy không liên lạc được với gateway → không ra Internet |
| via không đúng địa chỉ router | Không có đường ra mạng ngoài |
| DNS sai | Không mở được website bằng tên miền (chỉ ping được IP) |

**✅ Kết luận dễ nhớ**

✔️ addresses → là IP bạn muốn **Ubuntu dùng**  
✔️ via → là địa chỉ **của router nhà bạn (thường là 192.168.1.1)**  
✔️ nameservers → là **máy chủ trên Internet** giúp dịch tên miền

Bạn đã hiểu phần này chưa? Nếu cần, mình có thể **vẽ sơ đồ kết nối** để bạn hình dung ra cách mọi thứ liên kết với nhau 👇  
Hoặc bạn muốn mình kiểm tra lại file YAML bạn đang dùng để đảm bảo đúng cấu hình nhé?

**You said:**

gIẢI THÍCH CHO MÌNH HIỂU VỀ "routes", "default gateway", "DNS" "WIFI" "MODEM" "CẤU HÌNH MẠNG" "CẤU HÌNH IP" "IP TĨNH" "IP ĐỘNG" "CẤU HÌNH IP TĨNH " "CẤU HÌNH IP ĐỘNG" LÀ GÌ?để làm gì? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức:

**ChatGPT said:**

Câu hỏi của bạn rất hay, rất thiết thực, và đúng chuẩn **người mới học từ con số 0** như bạn nói.  
Mình sẽ giúp bạn **giải thích cặn kẽ từng khái niệm**, với **ngôn ngữ dễ hiểu nhất**, theo thứ tự từng phần:  
Bạn sẽ hiểu:

* 👉 Nó là gì?
* 👉 Dùng để làm gì?
* 👉 Vai trò & ứng dụng ra sao?

**🔶 1. Wi-Fi là gì?**

**✅ Là cách máy tính kết nối mạng không dây (wireless)**

* Không cần dây cáp, chỉ cần sóng từ router phát ra
* Máy tính, điện thoại, laptop đều kết nối Wi-Fi để truy cập Internet

📌 **Wi-Fi chỉ là “phương tiện kết nối”**, không phải là Internet.

**🔶 2. Modem là gì?**

**✅ Là thiết bị giúp kết nối mạng của bạn ra Internet**

* Modem lấy tín hiệu từ **nhà mạng (VNPT, Viettel, FPT...)**
* Sau đó **chuyển thành tín hiệu mạng trong nhà bạn**

🧱 Một modem thường được **gắn chung với router**, gọi là “modem Wi-Fi”

**🔶 3. Router là gì?**

**✅ Là thiết bị chia mạng ra cho nhiều thiết bị dùng chung**

* Bạn có thể tưởng tượng router như **trạm phát sóng trong nhà**
* Router có thể cấp IP cho máy tính, điện thoại, máy in...

📌 Hầu hết các thiết bị bạn gọi là **“cục Wi-Fi”** chính là **modem + router trong 1 thiết bị**

**🔶 4. Cấu hình mạng (network configuration) là gì?**

**✅ Là việc thiết lập cách máy tính kết nối mạng**

* Bao gồm: gán địa chỉ IP, gateway, DNS...
* Cấu hình đúng → máy ra mạng được
* Cấu hình sai → không vào Internet

📌 Đây là “địa chỉ nhà, đường đi, bản đồ” của máy trong mạng

**🔶 5. IP là gì?**

**✅ Là “địa chỉ nhà” của thiết bị trong mạng**

* IP (Internet Protocol address) là **địa chỉ duy nhất** trong mạng LAN
* Giúp máy tính **giao tiếp với nhau**
* IP giống như số nhà trong thành phố

Ví dụ: 192.168.1.100

**🔶 6. Cấu hình IP là gì?**

**✅ Là việc đặt địa chỉ IP cho máy tính**

Có 2 loại:

**🔵 A. IP động (Dynamic IP)**

**✅ Là IP tự động được cấp bởi router**

* Máy tính tự hỏi: “Ai cho tôi địa chỉ IP?”
* Router trả lời: “Đây là IP cho bạn dùng: 192.168.1.12”
* Dùng giao thức **DHCP (Dynamic Host Configuration Protocol)**

📌 IP này **có thể thay đổi mỗi lần bạn bật máy**

📌 **Ưu điểm:** tiện lợi, không cần cài thủ công  
📌 **Nhược điểm:** mỗi lần IP thay đổi → có thể gây lỗi nếu máy cần IP cố định (VD: server)

**🔵 B. IP tĩnh (Static IP)**

**✅ Là IP do bạn tự cài đặt**

* Ví dụ: bạn nói “Máy của tôi luôn là 192.168.1.100”
* Bạn phải tự cấu hình bằng tay (dòng lệnh hoặc giao diện)

📌 **Ưu điểm:** IP không đổi, dùng tốt cho máy chủ (server)  
📌 **Nhược điểm:** nếu cấu hình sai → không kết nối được

**🔶 7. Cấu hình IP động (DHCP) là gì?**

* Bạn để máy **tự lấy IP**
* Trong Ubuntu thì cấu hình:

dhcp4: yes

* Máy sẽ hỏi router: “Tôi cần IP” → router cấp IP, gateway, DNS

**🔶 8. Cấu hình IP tĩnh là gì?**

* Bạn **tự gán IP, gateway, DNS**
* Trong Ubuntu thì viết như:

dhcp4: no

addresses: [192.168.1.100/24]

gateway4: 192.168.1.1

nameservers:

addresses: [8.8.8.8, 1.1.1.1]

📌 Đòi hỏi bạn **biết rõ mạng mình đang dùng là gì**

**🔶 9. Default Gateway là gì?**

**✅ Là “cánh cửa” để máy bạn ra ngoài mạng**

* Nếu không có Gateway → chỉ liên lạc được trong mạng nội bộ
* Thường là IP của router, ví dụ: 192.168.1.1

📌 Máy gửi dữ liệu ra ngoài → sẽ gửi qua gateway

**🔶 10. DNS là gì?**

**✅ DNS (Domain Name System) là “sổ địa chỉ” của Internet**

* Máy không hiểu google.com, chỉ hiểu IP
* DNS giúp bạn **đổi tên → thành IP**

Ví dụ:

* Bạn gõ: google.com
* DNS dịch ra: 142.250.198.238

📌 Không có DNS → bạn chỉ mở web bằng IP được

**🔶 11. Routes là gì?**

**✅ Là đường đi mạng từ máy bạn → đến các mạng khác**

* Dùng để chỉ đường: muốn ra mạng nào → đi qua đâu?

Ví dụ:

routes:

- to: default

via: 192.168.1.1

Nghĩa là:

Muốn đi đến **bất kỳ đâu ngoài mạng nội bộ**, thì đi qua **gateway 192.168.1.1**

**🔶 12. Tóm tắt dễ hiểu toàn bộ:**

| **Tên** | **Là gì?** | **Dùng để làm gì?** |
| --- | --- | --- |
| **Wi-Fi** | Kết nối mạng không dây | Giúp máy kết nối Internet |
| **Modem** | Thiết bị nhận tín hiệu mạng từ nhà mạng | Giúp nhà bạn có mạng |
| **Router** | Chia mạng cho nhiều thiết bị | Phát Wi-Fi, cấp IP |
| **IP** | Địa chỉ của máy trong mạng | Máy nào cũng cần IP để giao tiếp |
| **IP động** | IP do router cấp tự động | Nhanh, tiện |
| **IP tĩnh** | IP do bạn tự đặt | Ổn định, không thay đổi |
| **Cấu hình mạng** | Cách thiết lập IP, DNS, gateway | Giúp máy vào mạng đúng cách |
| **Gateway** | Cổng ra mạng ngoài | Không có = không ra Internet |
| **DNS** | Hệ thống đổi tên miền → IP | Mở web bằng tên dễ nhớ |
| **Routes** | Đường đi dữ liệu trong mạng | Chỉ đường đến mạng khác |

Nếu bạn muốn, mình có thể vẽ **sơ đồ minh họa mạng nhà bạn**, hoặc hướng dẫn bạn **thực hành cấu hình IP động / tĩnh trên Ubuntu** từng bước nhé!

Bạn có muốn mình làm tiếp phần đó không?

**You said:**

gIẢI THÍCH CHO MÌNH HIỂU VỀ "DHCP" "NAT" SUBNETMARK" "CÁC LỚP MẠNG" LÀ GÌ?để làm gì? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức:

**ChatGPT said:**

Rất tốt! Đây là những kiến thức **cốt lõi** trong mạng máy tính – **bạn hiểu được là bạn đã vượt 1 bậc từ “non-tech” sang “có nền tảng”** rồi. Mình sẽ giải thích từng khái niệm **một cách chi tiết, đơn giản, dễ hiểu**, dành riêng cho người **bắt đầu từ con số 0** như bạn nhé.

**✅ 1. DHCP là gì?**

**📌 DHCP = Dynamic Host Configuration Protocol**

Là giao thức giúp máy tính **tự động nhận thông tin mạng** như:

* IP address (địa chỉ IP)
* Default Gateway
* DNS
* Subnet Mask

**🧠 Nói cho dễ hiểu:**

* Bạn tưởng tượng **router/modem là ông tổ trưởng** của khu phố
* Khi 1 thiết bị **(laptop, điện thoại, máy ảo Ubuntu)** kết nối, nó hỏi:

“Chào ông tổ trưởng! Tôi mới đến, cho tôi địa chỉ nhà (IP) được không?”

* Router sẽ trả lời:

“Được rồi! Cậu là 192.168.1.100, gateway là 192.168.1.1, DNS là 8.8.8.8”

📌 Tất cả diễn ra **tự động**, bạn **không cần cấu hình thủ công**  
📌 DHCP **tiện lợi** nhưng mỗi lần kết nối lại, có thể sẽ **được IP khác nhau**

**🧩 Tác dụng:**

| **Tác dụng** | **Giải thích** |
| --- | --- |
| Cấp IP tự động | Không cần cấu hình thủ công |
| Tránh trùng IP | Router tự quản lý, tránh 2 máy trùng địa chỉ |
| Quản lý dễ dàng | Hữu ích cho mạng đông người |

**✅ 2. NAT là gì?**

**📌 NAT = Network Address Translation**

Là kỹ thuật **chuyển đổi địa chỉ IP** từ mạng nội bộ → ra mạng bên ngoài (Internet)

**🧠 Dễ hiểu hơn:**

* Bạn có **nhiều máy trong nhà** (máy ảo, laptop, điện thoại...)
* Nhưng chỉ có **1 đường ra Internet**, do **nhà mạng chỉ cấp 1 IP thật**

📌 NAT giúp router làm "phiên dịch":

* Máy nội bộ (ví dụ 192.168.1.100) gửi yêu cầu ra ngoài
* NAT ghi lại: "OK, yêu cầu từ 192.168.1.100 → google.com"
* Khi Google trả lời, NAT **gửi lại đúng máy đã yêu cầu**

**🧩 Tác dụng:**

| **Tác dụng** | **Giải thích** |
| --- | --- |
| Cho nhiều máy cùng dùng 1 IP thật | Tiết kiệm IP thật |
| Ẩn mạng nội bộ | Máy bên ngoài không thấy trực tiếp IP nội bộ |
| Bảo mật | Hạn chế truy cập trái phép từ ngoài |

**✅ 3. Subnet Mask là gì?**

**📌 Là giá trị giúp phân biệt phần “mạng” và phần “máy” trong địa chỉ IP**

Mỗi địa chỉ IP có 2 phần:

* 🏢 **Phần mạng (Network)**: xác định nhóm máy
* 🧑‍💻 **Phần host (thiết bị)**: máy cụ thể trong mạng đó

Ví dụ IP: 192.168.1.100  
Nếu subnet mask là 255.255.255.0, thì:

* 192.168.1 là **mạng**
* .100 là **máy**

**🧠 Dễ hiểu hơn:**

* Subnet mask giống như **bản đồ vạch ranh giới khu phố**
* Nó bảo rằng: **các IP cùng phần “mạng” thì nằm trong 1 khu**

📌 Thường dùng:

* 255.255.255.0 → mỗi mạng có tối đa **254 máy**
* 255.255.0.0 → mạng lớn hơn, có hàng nghìn máy

**🧩 Tác dụng:**

| **Tác dụng** | **Giải thích** |
| --- | --- |
| Phân chia mạng | Biết máy nào “cùng mạng” |
| Quản lý hiệu quả | Mỗi mạng có số lượng máy giới hạn |
| Giao tiếp | Giúp máy biết khi nào cần gửi ra gateway |

**✅ 4. Các lớp mạng (Class A, B, C)**

Là cách chia địa chỉ IP thành **nhiều loại, dùng cho các quy mô mạng khác nhau**

**📌 Có 3 lớp chính:**

| **Lớp mạng** | **IP bắt đầu** | **Dải địa chỉ** | **Số lượng host (máy)** | **Dùng cho** |
| --- | --- | --- | --- | --- |
| **Class A** | 1.0.0.0 – 126.255.255.255 | 10.0.0.0/8 | ~16 triệu | Rất lớn (công ty lớn) |
| **Class B** | 128.0.0.0 – 191.255.255.255 | 172.16.0.0/12 | ~65.000 | Trung bình |
| **Class C** | 192.0.0.0 – 223.255.255.255 | 192.168.0.0/16 | ~254 | Nhỏ (gia đình, văn phòng) |

📌 Trong thực tế, bạn hay thấy:

* **192.168.x.x** → mạng lớp C → dùng trong gia đình
* **172.16.x.x** → lớp B → các hệ thống lớn hơn
* **10.x.x.x** → lớp A → server, doanh nghiệp lớn

**🧩 Tác dụng:**

| **Tác dụng** | **Giải thích** |
| --- | --- |
| Phân loại địa chỉ IP | Tùy theo số lượng máy |
| Quản lý IP tốt hơn | Mạng lớn cần IP nhiều |
| Giới hạn dùng nội bộ | Các dải trên là IP riêng (không dùng trực tiếp trên Internet) |

**📚 Tổng kết đơn giản**

| **Thuật ngữ** | **Là gì?** | **Dùng để làm gì?** |
| --- | --- | --- |
| **DHCP** | Cấp IP tự động | Máy tự nhận IP, gateway, DNS |
| **NAT** | Dịch địa chỉ IP | Giúp nhiều máy dùng 1 IP ra Internet |
| **Subnet Mask** | Ranh giới mạng | Biết đâu là mạng, đâu là máy |
| **Class A/B/C** | Các loại mạng | Dành cho các quy mô khác nhau |

**🎓 Gợi ý học tiếp:**

* Bạn có muốn mình **vẽ sơ đồ trực quan** về **DHCP, NAT và Subnet** không?
* Hoặc hướng dẫn bạn **thực hành phân biệt IP động và IP tĩnh trên Ubuntu**?

Chỉ cần nói "có", mình sẽ làm từng bước rõ ràng cho bạn nhé!

**You said:**

gIẢI THÍCH CHO MÌNH HIỂU VỀ "cấu hình mạng (network) cho Ubuntu Server – cụ thể là cài đặt địa chỉ IP, gateway, DNS,... để máy kết nối được vào mạng LAN, Internet, hoặc giao tiếp với máy khác." là gì? tức là sao? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức

**ChatGPT said:**

Rất hay! Đây là một **kiến thức nền tảng cực kỳ quan trọng** khi bạn học về **Ubuntu Server**, đặc biệt nếu bạn muốn **kết nối máy ảo vào mạng**, **cài đặt web server**, hoặc **giao tiếp giữa các máy trong mạng LAN hoặc Internet**.

**💡 Câu hỏi của bạn:**

“**Cấu hình mạng (network) cho Ubuntu Server** – cụ thể là **cài đặt địa chỉ IP, gateway, DNS,... để máy kết nối được vào mạng LAN, Internet, hoặc giao tiếp với máy khác**” là gì? Tác dụng ra sao?

**✅ Mình sẽ giải thích cho bạn từ gốc, từng bước, thật dễ hiểu và tường tận, đúng với bạn là người “non-tech” bắt đầu từ con số 0.**

**🧠 1. Mạng là gì? Vì sao Ubuntu Server cần cấu hình mạng?**

* **Mạng (Network)** là khi **nhiều thiết bị** (máy tính, điện thoại, server...) **kết nối với nhau** để:
  + Trao đổi dữ liệu
  + Chia sẻ file, máy in
  + Truy cập Internet
  + Giao tiếp giữa các phần mềm (web server, database…)

👉 Một máy tính **muốn “nói chuyện” được với máy khác hoặc ra Internet**, thì nó **phải kết nối mạng đúng cách**.  
Ubuntu Server **sau khi cài xong**, nó là **một máy tính rỗng** – **bạn phải cấu hình cho nó biết:**

* Tôi có địa chỉ IP gì?
* Tôi sẽ gửi dữ liệu qua đâu (gateway)?
* Ai sẽ giúp tôi tra cứu tên miền (DNS)?

**🔧 2. Cấu hình mạng là gì?**

Là quá trình bạn **thiết lập địa chỉ mạng** cho Ubuntu Server, bao gồm:

* IP Address (Địa chỉ máy)
* Default Gateway (Cổng mặc định để ra ngoài)
* DNS (Tra cứu tên miền như google.com)

**📦 3. Thành phần cấu hình mạng gồm gì?**

| **Thành phần** | **Là gì?** | **Giải thích dễ hiểu** |
| --- | --- | --- |
| **IP Address** | Địa chỉ định danh của máy trong mạng | Giống như **địa chỉ nhà** – để người khác biết bạn ở đâu |
| **Gateway** | Cửa ra để ra ngoài mạng | Giống như **cổng ra đường chính** nếu bạn muốn ra khỏi khu phố |
| **DNS** | Máy chủ tra tên miền | Giúp bạn gõ google.com mà không cần nhớ IP của Google |
| **Netmask (Subnet mask)** | Phạm vi mạng | Cho biết máy nào là “hàng xóm” và máy nào là “ngoài mạng” |

**🏗️ 4. Có mấy cách cấu hình?**

| **Cách cấu hình** | **Mô tả** | **Ưu điểm / Nhược điểm** |
| --- | --- | --- |
| **IP động (DHCP)** | Máy tự động xin IP từ router | ✅ Nhanh, tiện ❌ IP thay đổi, khó quản lý |
| **IP tĩnh (Static IP)** | Bạn tự điền IP, gateway, DNS | ✅ Ổn định, dễ kết nối lại ❌ Phải cẩn thận điền đúng thông tin |

**🖥️ 5. Thực tế: Vì sao cần cấu hình?**

**✅ Trường hợp 1: Ubuntu server kết nối Internet**

* Muốn cài phần mềm (apt update, apt install nginx, v.v)
* Cần có mạng, DNS, gateway

**✅ Trường hợp 2: Ubuntu server làm web server**

* Bạn cần cho máy thật (Windows) truy cập trang web trong máy ảo
* Phải đặt IP tĩnh để **máy Windows biết địa chỉ để truy cập**

**✅ Trường hợp 3: Server giao tiếp trong mạng LAN (nội bộ)**

* Cài file server (Samba), database (MySQL), v.v
* Các máy cần **IP cố định** để nói chuyện với nhau

**📘 6. File cấu hình mạng nằm đâu?**

Trong Ubuntu Server (phiên bản mới), file cấu hình mạng nằm ở:

/etc/netplan/50-cloud-init.yaml

Bạn dùng lệnh:

sudo nano /etc/netplan/50-cloud-init.yaml

**✍️ 7. Ví dụ cấu hình IP tĩnh**

network:

version: 2

ethernets:

ens33:

dhcp4: no

addresses: [192.168.1.100/24]

routes:

- to: default

via: 192.168.1.1

nameservers:

addresses: [8.8.8.8, 1.1.1.1]

Giải thích:

* dhcp4: no: Tắt chế độ tự xin IP → dùng IP tĩnh
* addresses: Địa chỉ IP gán cho máy
* via: Gateway để ra Internet
* nameservers: DNS giúp tra cứu tên miền

Sau đó áp dụng:

sudo netplan apply

**🔍 8. Làm sao biết cấu hình đúng?**

Bạn kiểm tra bằng:

* ip a → xem IP có đúng không
* ip route → xem gateway có đúng không
* ping 8.8.8.8 → kiểm tra Internet
* ping google.com → kiểm tra DNS

**✅ 9. Tổng kết siêu dễ hiểu**

| **Khái niệm** | **Là gì?** | **Vai trò** |
| --- | --- | --- |
| IP Address | Địa chỉ của máy | Giúp máy định danh trong mạng |
| Gateway | Cửa ra mạng khác (Internet) | Để máy biết phải gửi dữ liệu đi đâu |
| DNS | Dịch tên miền thành IP | Gõ google.com → biết IP của Google |
| Netmask | Xác định mạng | Biết máy nào “cùng mạng” để kết nối |
| DHCP | Cấp IP tự động | Giúp bạn không cần điền IP thủ công |
| IP tĩnh | IP cố định | Ổn định, dùng cho server |

**❓Bạn muốn tiếp theo:**

* Hướng dẫn cấu hình IP tĩnh từng bước trong Netplan?
* Cách kiểm tra kết nối mạng, gateway, DNS?
* Cách giao tiếp giữa máy thật (Windows) và Ubuntu Server?

Chỉ cần nói "muốn", mình sẽ giúp bạn từ từ, chắc chắn, dễ hiểu!

**You said:**

"• Nếu server của bạn là web server, thì người khác sẽ truy cập bằng IP: 192.168.1.100 • Nếu bạn để IP động (DHCP), IP có thể thay đổi mỗi lần khởi động → không ai kết nối được nữa " Mình không hiểu? web server là gì? rồi giả sử trang shopee.com thì nó là web server à? hay web gì? rồi tại sao không gõ tên miền mà gõ địa chỉ ip? là gì? tức là sao? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức

**ChatGPT said:**

Rất hay, câu hỏi của bạn **cực kỳ đúng trọng tâm**! 💡 Mình sẽ **giải thích từ gốc đến ngọn**, từng khái niệm một cách **cụ thể – dễ hiểu – không cần kỹ thuật cao**, phù hợp với bạn là người **non-tech – chưa có nền tảng**.

**💻 1. "Web server" là gì?**

**Web server (máy chủ web)** là một **máy tính đặc biệt** (như Ubuntu Server) dùng để **lưu trữ và cung cấp trang web** cho người dùng truy cập qua Internet hoặc mạng LAN.

**🎯 Nói dễ hiểu:**

Khi bạn gõ shopee.com, bạn đang **truy cập vào một web server của Shopee**, máy đó chứa:

* Mã web (HTML, CSS, JavaScript)
* Cơ sở dữ liệu
* Ảnh sản phẩm, thông tin tài khoản v.v...

➡️ Web server sẽ **trả dữ liệu về trình duyệt** của bạn để bạn thấy trang Shopee.

**🛜 2. Vậy web server là máy tính à?**

✅ Đúng. Nó là **một máy tính giống máy bạn**, nhưng:

* Luôn hoạt động 24/7
* Cấu hình đặc biệt để phục vụ web (dùng phần mềm như Nginx, Apache…)
* Có địa chỉ IP để người khác truy cập

**🌐 3. Vậy còn "IP" là gì? Vì sao không gõ tên miền?**

**🌍 IP là địa chỉ của máy trong mạng**

Ví dụ: 192.168.1.100, 8.8.8.8...

* IP giống như **địa chỉ nhà** → muốn đến đúng nhà thì cần biết địa chỉ
* Nếu không có IP → không thể gửi dữ liệu đến máy đó

**🧠 4. Vì sao không gõ tên miền (shopee.com) mà phải gõ IP?**

**👉 Tên miền (domain) là "bí danh" cho IP**

* shopee.com là **tên dễ nhớ** dành cho con người
* Máy tính thì **cần địa chỉ IP thực tế**

Khi bạn gõ shopee.com, trình duyệt sẽ hỏi **máy chủ DNS** để biết:

“Ê, cho tao biết IP thật của shopee.com là gì?”  
Sau đó mới kết nối bằng IP.

🧠 **Gõ IP = truy cập thẳng máy**  
**Gõ tên miền = phải tra IP trước (qua DNS)**

**🧪 5. Giả sử bạn làm web server trong Ubuntu Server**

Bạn cài một website nhỏ vào Ubuntu Server trong VMware.

* Bạn đặt IP tĩnh là 192.168.1.100
* Máy thật (Windows) hoặc máy khác cùng mạng sẽ **truy cập web qua**:  
  http://192.168.1.100 ✅

👉 Nếu bạn dùng IP động (DHCP):

* Lúc này IP là 192.168.1.100
* Nhưng lần sau khởi động có thể đổi thành 192.168.1.103 ❌
* Người khác vẫn truy cập 192.168.1.100 → KHÔNG VÀO ĐƯỢC nữa ❌

=> **Lý do phải dùng IP tĩnh cho web server:**  
➡️ Để **giữ địa chỉ cố định** cho người khác dễ truy cập website.

**📌 6. Tóm tắt siêu dễ hiểu:**

| **Khái niệm** | **Nghĩa đơn giản** | **Ví dụ** |
| --- | --- | --- |
| **Web Server** | Máy tính phục vụ trang web | Máy chứa website Shopee |
| **IP Address** | Địa chỉ nhà của máy | 192.168.1.100, 8.8.8.8 |
| **Tên miền (Domain)** | Tên dễ nhớ thay cho IP | shopee.com |
| **DNS** | Máy tra cứu tên miền ra IP | Gõ shopee.com → DNS trả IP |
| **IP động (DHCP)** | IP tự động thay đổi | Mỗi lần khởi động có thể khác |
| **IP tĩnh** | IP cố định | Giữ nguyên → người khác dễ truy cập |

**✅ Kết luận:**

* Shopee là **web server thật sự**, họ dùng **nhiều server đặt trong trung tâm dữ liệu**
* Bạn cũng có thể **tạo web server trong Ubuntu Server**, để học hoặc chia sẻ nội bộ
* Nếu dùng IP tĩnh, **máy khác có thể truy cập website của bạn dễ dàng**

Nếu bạn muốn, mình có thể:

* Hướng dẫn bạn **tự tạo web server đơn giản**
* Hướng dẫn gán tên miền nội bộ cho IP (dùng /etc/hosts)
* Giải thích thêm về DNS và tên miền chuyên sâu hơn

Chỉ cần bạn nói **"mình muốn học tiếp"** – mình sẽ hướng dẫn từ gốc nhé!

**You said:**

Thế nếu có "Web server" thì có "web client" không? mình vẫn chưa hiểu ấy

**ChatGPT said:**

Câu hỏi của bạn rất thông minh, và đúng là **có sự phân biệt rõ ràng giữa “web server” và “web client”**. Vì bạn bảo chưa hiểu, mình sẽ giải thích theo cách **cực kỳ đơn giản – dễ hiểu – dành cho người chưa có nền tảng công nghệ** nhé!

**🖥️ Web Server là gì?**

* **Web Server** là **máy tính chứa trang web**, có nhiệm vụ **cung cấp nội dung** (giao diện, hình ảnh, văn bản...) cho người khác.
* Nó **phản hồi yêu cầu** từ người dùng.

📦 Ví dụ:

* Ubuntu Server bạn đang học cấu hình có thể là một **web server**
* Shopee, Facebook, Google… đều chạy trên các **web server**

**🧑‍💻 Web Client là gì?**

* **Web Client** là **thiết bị hoặc phần mềm dùng để truy cập website** – thường là trình duyệt web (Chrome, Firefox, Safari…).
* Web Client **gửi yêu cầu** đến Web Server → chờ phản hồi → hiển thị nội dung ra màn hình.

💡 **Web Client KHÔNG lưu trữ nội dung website**, mà **chỉ hiển thị**.

📌 **Web Client là bạn!**

* Khi bạn mở Chrome gõ shopee.com → bạn là **web client**
* Máy tính bạn đang dùng là **client**
* Trình duyệt là công cụ để **gửi yêu cầu** đến **server** chứa web

**🧠 Tóm tắt dễ hiểu:**

| **Thành phần** | **Là cái gì?** | **Vai trò** |
| --- | --- | --- |
| 🖥️ Web Server | Máy chủ chứa website | Phục vụ nội dung web |
| 🧑‍💻 Web Client | Máy tính/trình duyệt của bạn | Truy cập, hiển thị web |
| 🌐 Internet | Cầu nối giữa client và server | Truyền dữ liệu qua lại |

**📦 Ví dụ dễ hình dung**

Bạn đi ăn phở:

* 🧑‍💻 **Bạn = Web Client** (người đặt món)
* 🧑‍🍳 **Quán phở = Web Server** (nơi nấu, phục vụ)
* 🌐 **Internet = Đường đi giữa bạn và quán phở** (giao hàng, phục vụ)

Bạn nói “Cho tôi 1 tô phở!” (yêu cầu truy cập trang)  
→ Quán nấu và mang ra cho bạn (trả về nội dung web)

**✅ Kết luận:**

* Có **web client**, chính là **thiết bị người dùng** (bạn, trình duyệt, ứng dụng).
* Có **web server**, chính là **máy phục vụ website** (như Ubuntu Server bạn đang học).
* Web client và server **giao tiếp với nhau qua Internet**, sử dụng **giao thức HTTP**.

Nếu bạn muốn, mình có thể:

* Làm ví dụ đơn giản: Dùng Ubuntu Server làm web server, còn trình duyệt là client
* Hoặc giải thích thêm các bước **client truy cập web server như thế nào**

Chỉ cần nói: **“Mình muốn làm thử!”**, mình sẽ hướng dẫn chi tiết luôn nha.

**You said:**

Máy thật của mình (Windows) đang kết nối Wi-Fi và được cấp IP 192.168.1.6, gateway là 192.168.1.1 → Tức là mạng của bạn đang dùng dải IP: 192.168.1.x 📌 Vậy nếu bạn muốn máy Ubuntu Server (máy ảo) kết nối được Internet, thì nó phải được đặt IP cùng dải, ví dụ: • IP: 192.168.1.100 (bạn chọn) • Gateway: 192.168.1.1 (router/modem) • DNS: Google DNS (8.8.8.8), hoặc DNS của nhà mạng Vậy mình đặt ip và gateway cho máy ubuntu server như thế nào đây, mình dùng ip a thì nó hiển thị thông tin như sau: 23:31:33 lebao@us2:~$ ip a 1: 10: <LOOPBACK, UP, LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid\_lft forever preferred\_lft forever inet6 :: 1/128 scope host noprefixroute valid\_lft forever preferred\_lft forever 2: ens33: <BROADCAST, MULTICAST, UP, LOWER\_UP> mtu 1500 qdisc fq\_codel state UNKNOWN group default qlen 1000 link/ether 00:0c:29:fc:63:40 brd ffffffff:ff:ff altname enp2s1 inet 192.168.101.143/24 metric 100 brd 192.168.101.255 scope global dynamic ens33 valid\_lft 1433sec preferred\_lft 1433sec inet6 fe80::20c:29ff:fefc:6340/64 scope link valid\_lft forever preferred\_lft forever 23:34:30 lebao@us2:~$

**ChatGPT said:**

Rất tốt bạn đã làm lệnh ip a và gửi kết quả — từ kết quả đó, mình sẽ **phân tích chi tiết** xem máy Ubuntu Server của bạn hiện tại đang ở mạng nào, và từ đó hướng dẫn bạn **cách đặt IP & gateway đúng** để nó kết nối được Internet qua mạng Wi-Fi của máy thật bạn. Mình sẽ giải thích từng bước, dễ hiểu, để bạn hiểu “tại sao” chứ không chỉ “làm theo”.

**📊 Phân tích kết quả ip a bạn gửi**

Bạn chạy:

ip a

và nhận được:

2: ens33: <BROADCAST, MULTICAST, UP, LOWER\_UP> mtu 1500 qdisc fq\_codel state UNKNOWN group default qlen 1000

link/ether 00:0c:29:fc:63:40 brd ffffffff:ff:ff

altname enp2s1

inet 192.168.101.143/24 metric 100 brd 192.168.101.255 scope global dynamic ens33

valid\_lft 1433sec preferred\_lft 1433sec

inet6 fe80::… scope link

**✅ Những điều đáng chú ý:**

1. **Tên card mạng**: ens33 (và altname enp2s1)  
   → Đây là card mạng mà Ubuntu đang sử dụng.
2. **Địa chỉ IPv4 hiện tại**: 192.168.101.143/24  
   → Máy Ubuntu đang nằm trong mạng 192.168.101.x (đã được cấp IP động qua DHCP)  
   → *Không nằm trong mạng bạn muốn (192.168.1.x)*
3. scope global dynamic → IP này được cấp bởi DHCP (tự động)

**❓ Vấn đề của hiện tại**

* Máy Ubuntu đang ở mạng 192.168.101.x, trong khi máy Windows thật của bạn đang ở mạng 192.168.1.x
* Hai máy nằm ở **mạng khác nhau**, nên nếu bạn đặt IP tĩnh 192.168.1.100 mà vẫn để network adapter ở chế độ NAT hay không kết nối đúng mạng, Ubuntu sẽ “lạc” giữa mạng — không kết nối được với mạng Wi-Fi của máy thật
* Cấu hình hiện tại của Ubuntu có thể đang sử dụng NAT do VMware thiết lập mạng ảo VMnet8 hoặc tương tự — nên nó được cấp mạng 192.168.101.x

**🛠️ Cách đặt IP & gateway cho máy Ubuntu Server để nó nằm chung mạng với máy thật (192.168.1.x)**

Muốn làm điều đó, máy ảo Ubuntu của bạn phải:

* Được kết nối với mạng Wi-Fi của máy thật (qua VMware với chế độ Bridged hoặc cấu hình mạng phù hợp)
* Có IP cùng dải mạng: 192.168.1.x
* Có gateway là 192.168.1.1 (router của bạn)
* DNS hợp lý

Các bước cụ thể:

**Bước 1: Kiểm tra thiết lập mạng ảo trong VMware**

Trước khi cấu hình IP, bạn cần đảm bảo:

* Adapter mạng của máy ảo Ubuntu được thiết lập **Bridged** (cầu nối vào mạng Wi-Fi) — để máy ảo được như một máy thật trong mạng 192.168.1.x
  + Nếu để ở chế độ **NAT**, máy ảo sẽ nằm trong mạng nội bộ riêng của VMware (ví dụ mạng 192.168.101.x) và không thể “nhập chung mạng nhà bạn” trực tiếp.
* Nếu đã là Bridged, máy ảo Ubuntu sẽ “nhìn thấy” router Wi-Fi chính của bạn như một gateway.

**Bước 2: Sửa file cấu hình Netplan**

Mở file YAML cấu hình mạng:

sudo nano /etc/netplan/50-cloud-init.yaml

Sửa nội dung thành ví dụ sau (giả sử tên card mạng là ens33):

network:

version: 2

ethernets:

ens33:

dhcp4: no

addresses: [192.168.1.100/24]

routes:

- to: default

via: 192.168.1.1

nameservers:

addresses: [8.8.8.8, 1.1.1.1]

Giải thích:

* addresses: [192.168.1.100/24] → đặt IP tĩnh cho Ubuntu là 192.168.1.100, subnet /24 (tức là mạng có địa chỉ từ 192.168.1.0 đến 192.168.1.255)
* via: 192.168.1.1 → gateway mặc định là router Wi-Fi bạn dùng
* nameservers: [8.8.8.8, 1.1.1.1] → DNS dùng Google & Cloudflare

Lưu lại (Ctrl + O, Enter) và thoát (Ctrl + X).

**Bước 3: Áp dụng cấu hình mới**

sudo netplan apply

**Bước 4: Kiểm tra lại**

ip a

Bạn phải thấy:

* Card ens33 có IP 192.168.1.100/24
* Địa chỉ gateway trong ip route là 192.168.1.1

Rồi thử:

ping 8.8.8.8

ping google.com

Nếu trả lời bình thường — mạng đã làm việc đúng.

**✅ Kết luận dành cho bạn**

* **addresses** là IP bạn đặt cho Ubuntu — để Ubuntu “ở” trong mạng 192.168.1.x như máy thật của bạn
* **via** là gateway mặc định — là router Wi-Fi 192.168.1.1 — để Ubuntu “ra” khỏi mạng nội bộ đi ra Internet
* Vì hiện tại bạn dùng NAT/không ở cùng mạng, Ubuntu được cấp IP 192.168.101.x — bạn cần chuyển máy ảo qua Bridged và đặt IP trong mạng nhà bạn để hai máy có thể “chung mạng”

Nếu bạn muốn, mình có thể hướng dẫn bạn **chọn chế độ mạng Bridged trong VMware**, và viết file cấu hình YAML chính xác cho máy bạn để chỉ cần copy-paste. Muốn mình làm điều đó không?

**You said:**

cho mình hỏi khi mình gõ ping google.com tức là đang làm gì? và nó trả về như này nghĩa là sao? ping google.com Nó trả về: PING google.com (142.250.198.238) 56(84) bytes of data. 64 bytes from nchkgb-ah-in-f14.1e100.net (142.250.198.238): icmp\_seq=1 ttl=128 time=51.1 ms Và cái địa chỉ ip (142.250.198.238) nghĩa là gì? tức là sao? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức

**ChatGPT said:**

Rất tuyệt khi bạn hỏi kỹ để hiểu rõ tận gốc như vậy. Mình sẽ **giải thích chi tiết, từ đầu, từng khái niệm một cách dễ hiểu** — đúng kiểu dành cho người mới bắt đầu, non-tech nhé!

**🎯 Câu hỏi: ping google.com là đang làm gì?**

**✅ Câu lệnh ping là gì?**

* ping là một **câu lệnh kiểm tra kết nối mạng** giữa máy tính của bạn (Ubuntu server) và một máy chủ khác (trong trường hợp này là google.com).
* Nó hoạt động giống như **việc bạn “gọi thử” ai đó để xem họ có đang online không**.

🧠 Tưởng tượng như:

Bạn đứng từ xa và gọi lớn: “Google ơi, có đó không?”, rồi Google trả lời: “Tôi đây!”

**✅ Khi bạn gõ:**

ping google.com

Ubuntu sẽ thực hiện 2 việc quan trọng:

1. **Tra cứu DNS (Domain Name System)**:  
   Nó sẽ hỏi: “Google.com là IP nào vậy?”  
   → DNS sẽ trả về IP thật của google.com (ví dụ: 142.250.198.238)
2. **Gửi gói tin ICMP tới địa chỉ IP đó**:  
   Ubuntu sẽ gửi các gói tin gọi là **ICMP Echo Request** đến IP đó. Nếu Google nhận được, nó sẽ phản hồi lại bằng **ICMP Echo Reply**

**📦 Kết quả bạn nhận được:**

PING google.com (142.250.198.238) 56(84) bytes of data.

64 bytes from nchkgb-ah-in-f14.1e100.net (142.250.198.238): icmp\_seq=1 ttl=128 time=51.1 ms

Giải thích chi tiết:

| **Phần** | **Ý nghĩa** |
| --- | --- |
| PING google.com | Bạn đang ping tên miền google.com |
| (142.250.198.238) | Đây là địa chỉ IP của máy chủ Google (Google có rất nhiều IP, bạn nhận được 1 trong số đó) |
| 56(84) bytes of data | Gói tin gửi đi có 56 byte dữ liệu, tổng cộng 84 byte bao gồm cả phần header |
| 64 bytes from ... | Máy chủ Google đã nhận và trả lời gói tin, phản hồi bạn bằng gói tin 64 byte |
| icmp\_seq=1 | Đây là gói tin thứ nhất được gửi đi |
| ttl=128 | "Time to live" – một giá trị kỹ thuật cho biết gói tin đã qua bao nhiêu thiết bị mạng |
| time=51.1 ms | Mất 51.1 mili-giây để gửi đi và nhận phản hồi lại từ Google |

**✅ Địa chỉ IP 142.250.198.238 là gì?**

* Đây là **địa chỉ IP thật của máy chủ Google** mà bạn đang gửi lệnh ping đến.
* Nó được **tra cứu tự động thông qua DNS**.

👉 Bạn gõ ping google.com, nhưng thật ra đằng sau, máy tính không hiểu "google.com" là gì — nó phải hỏi DNS để lấy ra **địa chỉ IP thật**, giống như địa chỉ nhà của Google, rồi mới gửi “gói tin ping” đến đó.

**🤔 Vậy ping để làm gì?**

Bạn dùng ping để:

| **Mục đích** | **Ý nghĩa** |
| --- | --- |
| ✅ Kiểm tra kết nối Internet | Nếu ping 8.8.8.8 hoặc ping google.com thành công → máy bạn có Internet |
| ✅ Kiểm tra DNS có hoạt động không | Nếu ping 8.8.8.8 được nhưng ping google.com lỗi → DNS có vấn đề |
| ✅ Kiểm tra kết nối giữa 2 máy | Dùng để xem máy này có “nhìn thấy” máy kia không (trong cùng mạng LAN chẳng hạn) |

**🔁 Tóm tắt dễ hiểu**

| **Câu hỏi** | **Trả lời dễ hiểu** |
| --- | --- |
| ping google.com là gì? | Gửi gói tin kiểm tra đến máy chủ của Google để xem có kết nối không |
| 142.250.198.238 là gì? | Là địa chỉ IP của một máy chủ Google thật sự mà bạn đang kiểm tra |
| Tại sao không chỉ dùng tên miền? | Vì máy tính chỉ hiểu địa chỉ IP, nên phải dịch google.com sang IP |
| Dùng ping để làm gì? | Để kiểm tra xem mạng có hoạt động không, có vào Internet được không, có DNS không |

Nếu bạn muốn, mình có thể chỉ luôn cách dùng ping để **kiểm tra kết nối giữa Ubuntu Server và máy Windows thật**, hoặc giữa Ubuntu Server và Router — rất hữu ích khi học mạng cơ bản.

Bạn có muốn mình chỉ luôn không?

**You said:**

"(142.250.198.238): Đây là địa chỉ IP của máy chủ Google (Google có rất nhiều IP, bạn nhận được 1 trong số đó)" Tức là sao?tức là sao? Có tác dụng vai trò gì nhỉ? hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức

**ChatGPT said:**

Tuyệt vời, bạn đang đặt đúng những câu hỏi cực kỳ quan trọng để hiểu **gốc rễ của cách Internet hoạt động**. Mình sẽ **giải thích thật dễ hiểu, chi tiết và từ từ** — đúng kiểu dành cho người chưa có nền tảng công nghệ nhé.

**❓Câu hỏi của bạn:**

“(142.250.198.238): Đây là địa chỉ IP của máy chủ Google (Google có rất nhiều IP, bạn nhận được 1 trong số đó)” — Tức là sao? Để làm gì? Có vai trò gì?

**🧠 Đầu tiên: Máy chủ Google là gì?**

* Máy chủ Google (Google Server) là **một chiếc máy tính cực kỳ mạnh**, được Google cài đặt để:
  + Chạy các dịch vụ như **Google Tìm kiếm**, **Gmail**, **YouTube**, v.v.
  + Khi bạn truy cập https://google.com thì **trình duyệt của bạn gửi yêu cầu tới máy chủ Google**, và máy chủ Google gửi lại dữ liệu (trang web, hình ảnh, kết quả tìm kiếm...)

🧱 Hãy tưởng tượng:

Google có hàng ngàn **“văn phòng” (máy chủ)** trên khắp thế giới. Mỗi văn phòng có một **địa chỉ nhà** cụ thể — đó chính là **địa chỉ IP**.

**🌐 Địa chỉ IP là gì?**

* IP (Internet Protocol) là **“địa chỉ nhà”** của một thiết bị trong mạng.
* Giống như địa chỉ nhà số 123 đường ABC, TP. HCM.  
  → Trong Internet, địa chỉ IP giúp các máy tính biết **nên gửi dữ liệu đến đâu**.

👉 Ví dụ:

| **Tên miền (domain)** | **IP tương ứng** |
| --- | --- |
| google.com | 142.250.198.238 *(chỉ là 1 IP thôi)* |
| youtube.com | 142.250.200.14 |
| facebook.com | 157.240.25.35 |

**🤔 Tại sao Google có “nhiều IP”?**

Vì Google có hàng **nghìn máy chủ** đặt ở các nơi khác nhau trên thế giới.

🔍 Khi bạn gõ google.com, **DNS** sẽ chọn ra **một IP gần bạn nhất** để:

* Giúp bạn truy cập nhanh hơn (ít bị trễ).
* Giảm tải cho các máy chủ Google.

👉 Mỗi người ở mỗi quốc gia có thể thấy IP Google khác nhau.  
Ví dụ:

| **Người dùng** | **IP của google.com nhận được** |
| --- | --- |
| Bạn ở Việt Nam | 142.250.198.238 |
| Người ở Mỹ | 172.217.5.78 |
| Người ở Úc | 142.250.206.14 |

**✅ Vậy IP Google để làm gì?**

* Khi bạn gõ ping google.com, máy sẽ **tra cứu IP của google.com qua DNS**, ví dụ là 142.250.198.238.
* Sau đó **gửi tín hiệu đến đúng IP này** → kiểm tra xem có kết nối được không.

**📦 Tóm lại – Vai trò của IP Google là gì?**

| **Vai trò** | **Giải thích đơn giản** |
| --- | --- |
| Xác định máy chủ nào sẽ xử lý yêu cầu của bạn | Giống như bạn gửi thư — cần ghi đúng địa chỉ |
| Giúp bạn kết nối nhanh hơn | DNS chọn máy chủ gần bạn nhất để tối ưu tốc độ |
| Là đích đến của trình duyệt hoặc lệnh ping | Nếu không có IP, dữ liệu không biết đi đâu |

**🧪 Bonus – Bạn có thể kiểm tra IP này bằng:**

ping google.com

Hoặc:

nslookup google.com

→ Lệnh này sẽ cho bạn biết chính xác IP của google.com là bao nhiêu tại thời điểm đó.

**🎯 Kết luận**

* ✅ 142.250.198.238 là **địa chỉ IP của một máy chủ Google** đang phục vụ bạn.
* ✅ Google có rất nhiều IP vì có nhiều máy chủ.
* ✅ Việc tra IP là để **biết chính xác gửi dữ liệu đến đâu** trong Internet.
* ✅ Mỗi lần bạn ping, bạn có thể nhận được **một IP Google khác nhau**, nhưng tất cả đều là máy chủ của Google.

Nếu bạn muốn, mình có thể vẽ sơ đồ minh họa quá trình:  
**Bạn gõ google.com → DNS trả IP → gửi ping tới máy chủ đó**, rất trực quan.

Bạn muốn không?

**You said:**

Thế máy chủ của google chỉ chứa duy nhất 1 trang web tên là google.com thôi à? Có chứa các trang web khác như shopee.com hay tiki.vn, hay tiktok.vn gì gì đó không?

**ChatGPT said:**

Ôi không nhé! Máy chủ của Google **không chỉ chứa mỗi trang google.com đâu**, và cũng **không chứa các trang web như shopee.com, tiki.vn hay tiktok.vn**.

Mình giải thích chi tiết nha:

**1. Máy chủ của Google chứa gì?**

* Máy chủ của Google chủ yếu **chạy các dịch vụ và trang web của Google** như:
  + Google Search (google.com)
  + Gmail
  + YouTube
  + Google Drive
  + Google Maps
  + Và rất nhiều dịch vụ khác do Google cung cấp

Mỗi dịch vụ này có thể chạy trên nhiều máy chủ khác nhau, khắp nơi trên thế giới.

**2. Shopee.com, Tiki.vn, TikTok.vn thì sao?**

* Những trang web này không thuộc Google, mà là của các công ty khác, họ có **máy chủ riêng hoặc thuê dịch vụ máy chủ (hosting) khác**.
* Ví dụ:
  + Shopee có máy chủ riêng hoặc thuê máy chủ của các nhà cung cấp dịch vụ.
  + Tiki cũng tương tự.
  + TikTok có hệ thống máy chủ riêng của họ hoặc thuê các đám mây (cloud) như AWS, Azure,…

**3. Mạng Internet hoạt động thế nào?**

* Mỗi trang web (ví dụ shopee.com) có **địa chỉ IP riêng**, máy chủ riêng.
* Khi bạn nhập tên miền (ví dụ shopee.com), máy tính bạn hỏi DNS để lấy **IP máy chủ của Shopee**.
* Sau đó, dữ liệu sẽ gửi đến đúng máy chủ đó để lấy nội dung trang web.

**4. Tóm lại:**

| **Website** | **Máy chủ chứa trang web** |
| --- | --- |
| google.com | Máy chủ của Google |
| shopee.com | Máy chủ riêng của Shopee hoặc dịch vụ thuê ngoài |
| tiki.vn | Máy chủ riêng của Tiki hoặc dịch vụ thuê ngoài |
| tiktok.vn | Máy chủ riêng của TikTok hoặc dịch vụ thuê ngoài |

**Bạn hình dung như thế này nha:**

* Google giống như một tòa nhà lớn, bên trong chứa nhiều phòng (dịch vụ Google).
* Shopee, Tiki, TikTok là các tòa nhà khác, không nằm trong tòa nhà Google.

Nếu bạn muốn, mình có thể giải thích thêm về cách DNS hoạt động, giúp bạn hiểu rõ hơn cách “tên miền → địa chỉ IP → máy chủ” hoạt động nhé!

Bạn muốn không?

**You said:**

Ủa thế một máy chủ có 1 trang web duy nhất hay có nhiều trang web khác nhau?hãy giải thích chi tiết tường tận cụ thể cho mình hiểu với nha, do mình non tech, non kiến thức

**ChatGPT said:**

Rất hay câu hỏi! Mình sẽ giải thích thật chi tiết và dễ hiểu nhé!

**1. Máy chủ là gì?**

Máy chủ (server) là một chiếc máy tính mạnh mẽ, luôn kết nối mạng và chứa dữ liệu, ứng dụng để phục vụ người dùng khác qua Internet hoặc mạng nội bộ.

**2. Một máy chủ có thể chứa bao nhiêu trang web?**

* **Có thể chứa một trang web duy nhất**
* **Hoặc có thể chứa rất nhiều trang web khác nhau cùng lúc**

**Tại sao máy chủ có thể chứa nhiều trang web?**

**Ví dụ dễ hiểu:**

* Bạn có một căn nhà (máy chủ).
* Trong căn nhà này có thể bạn chỉ sống một mình (chỉ một trang web).
* Hoặc bạn cho thuê từng phòng nhỏ trong căn nhà đó cho nhiều người (nhiều trang web).

**3. Cách máy chủ chứa nhiều trang web**

Có 2 cách phổ biến:

**a) Máy chủ vật lý riêng biệt cho mỗi trang web**

* Ví dụ: Shopee có máy chủ riêng, Tiki có máy chủ riêng.
* Mỗi máy chủ này chỉ chạy một trang web hoặc dịch vụ lớn.

**b) Chia sẻ máy chủ dùng chung cho nhiều trang web (Shared Hosting)**

* Một máy chủ có thể chứa nhiều trang web nhỏ khác nhau.
* Nhà cung cấp dịch vụ hosting sẽ chia tài nguyên máy chủ cho nhiều khách hàng.
* Mỗi trang web sẽ có tên miền riêng, nhưng dùng chung một máy chủ.

**4. Cách máy chủ phân biệt các trang web khi có nhiều trang**

* Khi bạn gõ tên miền (ví dụ: shopee.com), máy chủ nhận biết bằng cách xem **tên miền được yêu cầu**.
* Máy chủ dựa vào tên miền này để trả về đúng nội dung trang web của từng khách.

**5. Mình lấy ví dụ thực tế**

* Bạn mua hosting (dịch vụ chứa web).
* Bạn có thể tạo trang web shopee.com, tiki.vn và nhiều trang khác trên cùng một máy chủ.
* Máy chủ sẽ phân biệt nội dung dựa trên tên miền, trả về trang tương ứng.

**6. Tóm lại**

| **Mô hình** | **Mô tả** |
| --- | --- |
| Máy chủ một trang web | Máy chủ mạnh, dành riêng cho 1 trang lớn |
| Máy chủ nhiều trang web | Máy chủ chia sẻ, nhiều trang web nhỏ cùng dùng chung |

**7. Vai trò của điều này**

* Giúp tiết kiệm chi phí khi nhiều trang web dùng chung máy chủ.
* Máy chủ riêng cho trang web lớn để đảm bảo hiệu năng cao.

Bạn có muốn mình giải thích thêm cách hoạt động của máy chủ, tên miền, hay cách phân biệt trang web khi dùng chung máy chủ không?

**You said:**

ủa nhưng mà mỗi máy chủ chỉ có 1 địa chỉ ip duy nhất hay nhiều địa chỉ ip?

**ChatGPT said:**

Câu hỏi rất hay! Mình giải thích rõ luôn nhé:

**1. Một máy chủ có thể có 1 địa chỉ IP hoặc nhiều địa chỉ IP**

* **Địa chỉ IP** là "số điện thoại" để máy khác gọi tới máy chủ đó trên mạng.
* Thông thường, mỗi máy chủ được gán **ít nhất 1 địa chỉ IP** duy nhất.
* Nhưng máy chủ cũng có thể có **nhiều địa chỉ IP** cùng lúc, tùy theo cách cấu hình mạng.

**2. Khi nào máy chủ có 1 địa chỉ IP?**

* Máy chủ nhỏ hoặc đơn giản.
* Ví dụ: Bạn thuê 1 máy chủ ảo (VPS) chỉ có 1 IP duy nhất để truy cập.

**3. Khi nào máy chủ có nhiều địa chỉ IP?**

* Máy chủ lớn, cần phục vụ nhiều dịch vụ khác nhau.
* Ví dụ: Một công ty lớn có thể có 1 máy chủ với nhiều IP để chạy nhiều trang web, dịch vụ, hoặc để phân bổ tải.
* Hoặc để có IP riêng cho từng dịch vụ (ví dụ mail, web, database).

**4. Mỗi địa chỉ IP có thể tương ứng với một hoặc nhiều trang web**

* Một IP có thể chứa nhiều trang web (qua kỹ thuật gọi là Virtual Hosting).
* Một trang web cũng có thể có nhiều IP (để tăng độ ổn định, cân bằng tải).

**5. Tóm tắt đơn giản**

| **Số địa chỉ IP trên máy chủ** | **Thông tin** |
| --- | --- |
| 1 | Máy chủ nhỏ, đơn giản |
| Nhiều | Máy chủ lớn, phức tạp hoặc nhiều dịch vụ |

**6. Ví dụ thực tế**

* Google có hàng ngàn máy chủ và hàng ngàn địa chỉ IP để phục vụ các dịch vụ khác nhau.
* Một website nhỏ bạn thuê hosting thường chỉ có 1 địa chỉ IP duy nhất.

Bạn muốn mình giải thích thêm về IP tĩnh, IP động hay cách địa chỉ IP liên quan đến tên miền không?

**You said:**

Thể giả sử web abc.com và web bcd.com đều nằm trên máy chủ XYZ, rồi máy chủ XYZ này chỉ có 1 địa chỉ ip là 145.168.23.44, và khi tôi gõ địa chỉ này trên thanh tìm kiếm thì nó trả về trang web nào?? Hay là tôi bị nhầm lẫn kiến thức gì đó? thế giả sử tôi ping abc.com thì nó trả về 145.168.23.44, rồi trang bcd.com có bị ảnh hưởng gì không? trời ơi sao rối vậy ta huhu

**ChatGPT said:**

Bạn hỏi rất hay, đây là vấn đề về **Virtual Hosting** — một kỹ thuật rất phổ biến trong thế giới web hosting! Mình giải thích kỹ từng điểm cho bạn dễ hiểu nhé:

**1. Một IP có thể chứa nhiều website khác nhau nhờ Virtual Hosting**

* Máy chủ XYZ có IP: **145.168.23.44**
* Trên máy chủ này, có thể cài nhiều website như:
  + abc.com
  + bcd.com
  + xyz.com
  + ...

**2. Khi bạn nhập địa chỉ IP trực tiếp (vd: 145.168.23.44) vào trình duyệt**

* Trình duyệt sẽ gửi yêu cầu tới máy chủ đó, nhưng không gửi kèm tên website.
* Máy chủ sẽ **trả về trang web mặc định** (default website) mà nó cấu hình cho IP đó.
* Thường đây là một trang giới thiệu hoặc trang lỗi, không phải luôn là abc.com hay bcd.com.

**3. Khi bạn gõ abc.com hoặc bcd.com vào trình duyệt**

* Trình duyệt sẽ hỏi hệ thống DNS: "abc.com là IP nào?"
* DNS trả về IP là 145.168.23.44
* Trình duyệt gửi yêu cầu tới 145.168.23.44, **và gửi kèm tên website (Host header)** là "abc.com"
* Máy chủ sẽ đọc tên này, và trả về đúng nội dung website abc.com
* Tương tự với bcd.com, trình duyệt gửi Host header là "bcd.com", máy chủ trả về trang bcd.com

**4. Tóm lại**

| **Bạn nhập** | **Kết quả** |
| --- | --- |
| 145.168.23.44 (IP) | Máy chủ trả về trang mặc định (default site) |
| abc.com | Máy chủ trả về trang abc.com |
| bcd.com | Máy chủ trả về trang bcd.com |

**5. Không bị nhầm đâu!**

* Việc nhiều website chia sẻ 1 IP là chuyện bình thường.
* Kỹ thuật **Virtual Hosting** cho phép máy chủ phân biệt yêu cầu của các website nhờ cái "Host header" trong HTTP.
* Vì vậy, **ping abc.com và ping bcd.com đều trả về 145.168.23.44** là bình thường.

**6. Bạn hình dung thế này:**

* IP = Địa chỉ nhà
* Website = Người sống trong nhà
* Có nhiều người (website) sống chung trong 1 nhà (IP)
* Khi bạn đến nhà (gõ IP), bạn phải nói rõ tên người muốn gặp (gõ tên miền abc.com), thì họ mới ra mở cửa cho bạn.

Bạn muốn mình giải thích thêm về **Host header** hoặc cách Virtual Hosting hoạt động chi tiết hơn không?

Hướng dẫn cái đổi hostname ubuntu server: lebao@ubuntu-server1:~$

**ChatGPT said:**

Để đổi **hostname** (tên máy chủ) trên **Ubuntu Server**, bạn có thể làm theo các bước dưới đây. Lưu ý là bạn cần quyền **sudo** để thực hiện các thay đổi này.

**🧾 Ví dụ:**

Giả sử bạn muốn đổi hostname từ ubuntu-server1 thành web-server.

**✅ Bước 1: Kiểm tra hostname hiện tại**

hostnamectl

Bạn sẽ thấy thông tin như:

Static hostname: ubuntu-server1

Icon name: computer-vm

Chassis: vm

Machine ID: ...

Boot ID: ...

Operating System: Ubuntu 22.04 LTS

Kernel: Linux ...

Architecture: x86-64

**✅ Bước 2: Đổi hostname tạm thời (cho tới khi reboot)**

sudo hostname web-server

Lệnh này sẽ đổi hostname ngay lập tức, nhưng **sẽ bị mất sau khi reboot**.

**✅ Bước 3: Đổi hostname vĩnh viễn**

Chạy lệnh sau:

sudo hostnamectl set-hostname web-server

Sau khi chạy lệnh này, hostname sẽ được đổi và **giữ nguyên sau khi reboot**.

**✅ Bước 4: (Tùy chọn) Cập nhật file /etc/hosts**

Mở file:

sudo nano /etc/hosts

Tìm dòng có hostname cũ, ví dụ:

127.0.1.1 ubuntu-server1

Và sửa lại thành:

127.0.1.1 web-server

Nhấn Ctrl + O để lưu, Enter, rồi Ctrl + X để thoát.

**✅ Bước 5: Khởi động lại (nếu muốn)**

sudo reboot

**✅ Kiểm tra lại sau khi reboot:**

hostnamectl

Hoặc:

hostname

Nếu bạn cần đổi ngược lại hoặc thêm domain (FQDN), mình có thể hướng dẫn thêm.

Bạn có muốn mình viết một script nhỏ để tự động đổi hostname không?

**You said:**

Hướng dẫn cái đổi username dành cho người mới bắt đầu đi

**ChatGPT said:**

Việc **đổi username (tên người dùng)** trên **Ubuntu Server** cần thực hiện cẩn thận vì liên quan đến thư mục home, quyền truy cập và các tiến trình đang chạy. Dưới đây là hướng dẫn đơn giản, từng bước cho **người mới bắt đầu**.

**🛑 Lưu ý trước khi đổi:**

* Không nên đổi username khi đang đăng nhập bằng **chính user đó**.
* Nên tạo một tài khoản tạm thời để đổi (hoặc dùng tài khoản root nếu có).
* Nên **sao lưu** dữ liệu quan trọng trước.

**✅ Ví dụ cụ thể:**

Giả sử bạn muốn đổi username từ lebao thành admin.

**🧱 Bước 1: Tạo user tạm thời để đổi**

(Chỉ cần nếu bạn đang dùng lebao để đăng nhập)

sudo adduser tempuser

Rồi đặt mật khẩu và các thông tin (có thể bỏ qua nếu không cần).

Thêm user này vào sudo để có quyền admin:

sudo usermod -aG sudo tempuser

**🔄 Bước 2: Đăng xuất khỏi user cũ và đăng nhập bằng tempuser**

Bạn có thể dùng lệnh:

exit

Hoặc nếu trên máy chủ thật, bạn thoát SSH rồi đăng nhập lại bằng tempuser.

**✏️ Bước 3: Đổi username**

Cú pháp:

sudo usermod -l [tên\_mới] [tên\_cũ]

Ví dụ:

sudo usermod -l admin lebao

**📁 Bước 4: Đổi tên thư mục home**

Mặc định, Ubuntu không đổi tên thư mục /home/lebao → /home/admin, bạn cần làm thủ công:

sudo mv /home/lebao /home/admin

Rồi cập nhật đường dẫn home trong hệ thống:

sudo usermod -d /home/admin -m admin

**🔐 Bước 5: Cập nhật file sudoers (nếu có cấu hình riêng)**

Nếu user lebao được cấu hình riêng trong /etc/sudoers hoặc thư mục /etc/sudoers.d/, bạn cần cập nhật lại tên lebao thành admin.

**🔁 Bước 6: Đăng xuất và đăng nhập lại bằng user mới**

Bạn có thể logout khỏi tempuser:

exit

Rồi đăng nhập lại bằng user mới admin.

**🧹 Bước 7: Xóa user tạm (nếu muốn)**

sudo deluser --remove-home tempuser

**✅ Kiểm tra lại**

whoami

Bạn sẽ thấy:

admin

Nếu bạn đang dùng SSH key, đừng quên copy file:

sudo cp -r /home/lebao/.ssh /home/admin/

sudo chown -R admin:admin /home/admin/.ssh

**❓Bạn cần tự động hóa không?**

Mình có thể viết một script shell giúp bạn đổi username chỉ với 1-2 lệnh nếu bạn muốn.

Bạn có muốn thử không?

**You said:**

Có câu lệnh nào để xem hiện tại có những thư mục nào không, kiểu như vậy? / ├── home/ │ └── lebao/ ├── etc/ ├── var/ └── usr/

**ChatGPT said:**

Có! Nếu bạn muốn **hiển thị cấu trúc thư mục** như ví dụ của bạn (cây thư mục dạng tree), bạn có thể dùng lệnh:

**✅ 1. Lệnh tree – chính xác như bạn muốn**

**Cài đặt tree (nếu chưa có):**

sudo apt update

sudo apt install tree

**Ví dụ dùng tree để xem thư mục gốc /:**

sudo tree -L 2 /

* -L 2: Chỉ hiện 2 cấp thư mục (cấp 1 là /, cấp 2 là home/, etc/, usr/...)
* Bạn có thể tăng giảm -L để xem sâu hơn hoặc nông hơn.

**✅ 2. Dùng ls + find (nếu không cài được tree)**

Nếu không cài được tree, bạn có thể dùng ls hoặc find để liệt kê:

**Hiện danh sách thư mục trong / (1 cấp):**

ls -l /

**Hiện tất cả thư mục con bên trong /home:**

find /home -maxdepth 2 -type d

**✅ Ví dụ đầu ra tree -L 2 / (giống bạn muốn):**

/

├── bin

├── boot

├── dev

├── etc

├── home

│ └── lebao

├── lib

├── media

├── mnt

├── opt

├── proc

├── root

├── run

├── sbin

├── srv

├── sys

├── tmp

├── usr

└── var

Nếu bạn muốn mình hướng dẫn cách xuất cấu trúc thư mục ra file .txt hoặc tìm thư mục nào đó trong hệ thống, mình có thể hỗ trợ thêm.

**You said:**

ê mà, nó xổ một đống thư mục, nhưng mà tôi không cuộn lên trên để xem được, ubuntu server nó không có chứ nặng cuộn như terminal hay cmd của windows à?

**ChatGPT said:**

Bạn nói đúng — trên **Ubuntu Server** (đặc biệt là khi dùng qua SSH hoặc console thuần túy), **nó không có thanh cuộn như terminal của Windows**. Vậy nên khi bạn chạy lệnh mà ra quá nhiều dòng (như tree / hoặc ls -lR), **nó "trôi tuốt" và bạn không cuộn lên được** — rất khó chịu đúng không?

Nhưng đừng lo, có vài cách rất đơn giản để xử lý chuyện này 👇

**✅ Cách 1: Dùng less để cuộn lên/xuống**

Đây là cách phổ biến nhất, kiểu như “xem từng trang một”.

**Ví dụ:**

sudo tree -L 2 / | less

Bây giờ bạn có thể:

* Nhấn ↑ hoặc k để cuộn lên
* Nhấn ↓ hoặc j để cuộn xuống
* Nhấn q để thoát

📌 Bạn có thể dùng less với bất kỳ lệnh nào in ra nhiều dòng.

**✅ Cách 2: Ghi kết quả ra file để xem sau**

**Ví dụ:**

sudo tree -L 2 / > ~/cay-thu-muc.txt

Sau đó mở xem:

less ~/cay-thu-muc.txt

**✅ Cách 3: Dùng trình SSH có hỗ trợ cuộn**

Nếu bạn SSH từ Windows:

| **SSH Client** | **Có cuộn không?** |
| --- | --- |
| **PuTTY** | ✅ Có |
| **Windows Terminal** | ✅ Có |
| **MobaXterm** | ✅ Có |
| **CMD cũ** | ❌ Không tốt |
| **PowerShell** | ✅ Có |

Nên nếu bạn đang dùng CMD để SSH, hãy thử **Windows Terminal** hoặc **MobaXterm** — cuộn mượt như kéo status Facebook 😄

**✅ Tóm lại**

* **Không cuộn được** là do giới hạn terminal, không phải lỗi.
* Dùng | less hoặc chuyển output ra file để xem thoải mái hơn.
* Cân nhắc dùng **terminal hỗ trợ cuộn** khi SSH từ Windows.

Nếu bạn cho mình biết bạn đang dùng terminal gì để truy cập Ubuntu (CMD, PowerShell, PuTTY, hay cái khác), mình sẽ chỉ cách tối ưu nhất cho trường hợp của bạn luôn.

**You said:**

oke muốn thoát chế độ cuộn thì làm sao?

**ChatGPT said:**

Rất đơn giản! Khi bạn đang ở **chế độ cuộn** (tức là khi dùng less), chỉ cần:

**✅ Nhấn q**

👉 Đây là **lệnh thoát (quit)** khỏi less.

**🔁 Ví dụ tình huống:**

tree -L 2 / | less

Nó sẽ mở ra giao diện cuộn (bạn thấy dấu : ở dưới cùng). Khi xem xong:

➡️ **Nhấn q để quay lại terminal bình thường.**

**⛔ Sai lầm hay gặp:**

* Gõ exit: ❌ Không đúng (nó thoát SSH luôn!)
* Nhấn Ctrl+C: ✅ Có thể dùng được, nhưng **không đẹp** bằng q.

Nếu bạn lỡ bị kẹt trong less, chỉ cần nhớ:  
👉 **ấn q để thoát.**

Còn gì vướng không? Mình chỉ thêm.

**You said:**

tắt máy theo giây mình tự đặt ra được không: kiểu sudo shutdown + số giây mà máy sẽ tắt

**ChatGPT said:**

Câu hỏi rất hay! Nhưng đáng tiếc là **lệnh shutdown không hỗ trợ đếm theo đơn vị giây**, mà chỉ hỗ trợ **phút** hoặc **giờ cụ thể**.

**❌ shutdown không hỗ trợ giây**

Ví dụ lệnh này:

sudo shutdown +10

⟶ là **tắt máy sau 10 phút**, **không phải 10 giây**.

**✅ Giải pháp: Dùng sleep + shutdown để tắt theo giây**

Bạn có thể kết hợp lệnh sleep để chờ số giây bạn muốn, rồi mới tắt:

**🔧 Ví dụ: Tắt máy sau 15 giây**

sleep 15 && sudo shutdown now

* sleep 15: chờ 15 giây
* && sudo shutdown now: sau khi chờ xong thì tắt máy ngay

📌 Bạn có thể thay 15 bằng bất kỳ số giây nào.

**✅ Nếu muốn tắt hẳn không cần xác nhận**

sleep 15 && sudo poweroff

**🧠 Mẹo: Chạy lệnh này trong nền để không bị kẹt terminal**

Nếu bạn muốn chạy nó rồi rảnh tay luôn, dùng dấu & để chạy trong nền:

(sleep 15 && sudo shutdown now) &

**Tóm lại:**

| **Bạn muốn** | **Lệnh** |
| --- | --- |
| Tắt sau 10 phút | sudo shutdown +10 |
| Tắt sau 15 giây | sleep 15 && sudo shutdown now |
| Tắt sau 30 giây (mạnh tay) | sleep 30 && sudo poweroff |

Nếu bạn cần hẹn giờ **khởi động lại (reboot)** thay vì tắt máy, mình chỉ luôn nhé.

**You said:**

Giải thích chi tiết cặn kẽ nội dung dưới đây cho mình hiểu Ubuntu is a free open-source Linux based operating system that is widely used for desktop, server and cloud platforms. It was developed and maintained by Canonical Ltd, and was the first released in October 2004. Ubuntu is based on Debian, another well-known Linux distribution but focuses on usability, regular updates and ease of installation. Ubuntu editions include: ➢ Ubuntu desktop: For personal computers and laptops ➢ Ubuntu server: Designed for server enviroments, cloud computing and data centers ➢ Ubuntu core: Minimal version focused on IoT and containerized applications

**ChatGPT said:**

Chắc chắn rồi! Mình sẽ **giải thích chi tiết, cặn kẽ từng câu, từng ý** trong đoạn bạn đưa ra để bạn nắm rõ từng phần – ngay cả khi bạn là người mới bắt đầu.

**🔹 Câu gốc:**

**"Ubuntu is a free open-source Linux based operating system that is widely used for desktop, server and cloud platforms."**

**✅ Giải thích:**

* **Ubuntu**: Là tên của một **hệ điều hành** (giống như Windows hay macOS).
* **Free**: Hoàn toàn **miễn phí** — bạn có thể tải về, cài đặt, và sử dụng mà không mất tiền.
* **Open-source**: Là phần mềm **mã nguồn mở** — tức là **ai cũng có thể xem, chỉnh sửa và phân phối lại mã nguồn của nó** (rất khác với phần mềm đóng như Windows).
* **Linux based operating system**: Ubuntu được **xây dựng dựa trên Linux**, một nhân hệ điều hành nổi tiếng vì sự ổn định và bảo mật.
* **Widely used for desktop, server and cloud platforms**:
  + **Desktop**: Dùng cho **máy tính cá nhân** (giống Windows).
  + **Server**: Dùng cho **máy chủ**, ví dụ máy chạy website hoặc hệ thống doanh nghiệp.
  + **Cloud platforms**: Dùng trong **điện toán đám mây**, như các máy chủ ảo chạy trên AWS, Azure, Google Cloud...

🧠 **Tóm lại**: Ubuntu là một hệ điều hành **miễn phí**, **mã nguồn mở**, dựa trên **Linux**, dùng được cho **cá nhân, doanh nghiệp và cả đám mây**.

**🔹 Câu gốc:**

**"It was developed and maintained by Canonical Ltd, and was first released in October 2004."**

**✅ Giải thích:**

* **Canonical Ltd**: Là công ty **phát triển và duy trì** Ubuntu.
* **First released in October 2004**: Ubuntu được **ra mắt lần đầu vào tháng 10 năm 2004**.

🧠 **Tóm lại**: Ubuntu được công ty Canonical tạo ra và bắt đầu phát hành từ năm 2004.

**🔹 Câu gốc:**

**"Ubuntu is based on Debian, another well-known Linux distribution but focuses on usability, regular updates and ease of installation."**

**✅ Giải thích:**

* **Based on Debian**: Ubuntu được **phát triển dựa trên Debian**, một bản phân phối (distribution) Linux lâu đời và ổn định.
* **Debian**: Là một hệ điều hành Linux nổi tiếng về độ ổn định, nhưng không thân thiện với người mới.
* Ubuntu **tập trung vào**:
  + **Usability**: Tính **dễ sử dụng** (dễ hơn Debian, nhất là cho người mới).
  + **Regular updates**: Cập nhật **đều đặn** và có lịch trình rõ ràng (2 bản chính mỗi năm).
  + **Ease of installation**: **Dễ cài đặt** — có trình cài đồ họa, không cần gõ lệnh phức tạp.

🧠 **Tóm lại**: Ubuntu lấy nền tảng từ Debian nhưng được làm cho **dễ dùng, dễ cài**, và **cập nhật thường xuyên hơn**.

**🔹 Ubuntu editions include:**

Đây là phần liệt kê **các phiên bản Ubuntu**, dành cho mục đích khác nhau.

**🔸 1. Ubuntu Desktop**

**"For personal computers and laptops"**

**✅ Giải thích:**

* Đây là phiên bản Ubuntu dành cho **máy tính cá nhân** như:
  + Laptop
  + PC để bàn
* Có giao diện đồ họa đẹp (GNOME), dễ sử dụng, có thể thay thế Windows.

🧠 **Dùng khi**: Bạn muốn cài Ubuntu lên máy tính cá nhân.

**🔸 2. Ubuntu Server**

**"Designed for server environments, cloud computing and data centers"**

**✅ Giải thích:**

* Dành cho **máy chủ (server)** — không có giao diện đồ họa mặc định.
* Chạy nhẹ, ổn định, bảo mật — dùng trong:
  + **Cloud computing**: chạy trên các dịch vụ đám mây.
  + **Data centers**: trung tâm dữ liệu lớn.
  + **Web server, database server**, v.v.

🧠 **Dùng khi**: Bạn muốn chạy web, backend, hoặc dịch vụ nào đó không cần giao diện đồ họa.

**🔸 3. Ubuntu Core**

**"Minimal version focused on IoT and containerized applications"**

**✅ Giải thích:**

* Là bản **tối giản** của Ubuntu.
* Tập trung vào:
  + **IoT (Internet of Things)**: Thiết bị thông minh như camera, thiết bị đo nhiệt độ...
  + **Containerized applications**: Ứng dụng đóng gói dạng container (giống Docker).
* Rất nhỏ gọn, chỉ cài những gì cần thiết.

🧠 **Dùng khi**: Bạn đang làm dự án IoT hoặc cần hệ điều hành nhẹ cho thiết bị nhúng/container.

**✅ Tóm tắt toàn bộ dễ hiểu:**

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| Ubuntu là gì? | Hệ điều hành miễn phí, mã nguồn mở, dựa trên Linux |
| Ai tạo ra? | Công ty Canonical Ltd (phát hành từ 2004) |
| Dựa vào đâu? | Dựa trên Debian nhưng dễ dùng hơn |
| Dùng ở đâu? | Máy tính cá nhân, server, cloud, IoT |
| Các phiên bản chính | Desktop (máy cá nhân), Server (máy chủ), Core (nhẹ, cho IoT) |

Nếu bạn muốn, mình có thể so sánh Ubuntu với Windows hoặc giải thích Ubuntu Core kỹ hơn nữa. Bạn muốn tiếp tục phần nào?

Dựa vào nội dung sau, viết lại cho mình chi tiết, đầy đủ, rõ ràng hơn:

**ChatGPT said:**

Tuyệt, bạn vui lòng gửi **nội dung** mà bạn muốn mình viết lại chi tiết hơn nhé.  
Mình đang chờ phần nội dung bạn nhắc tới trong câu:

"Dựa vào nội dung sau, viết lại cho mình chi tiết, đầy đủ, rõ ràng hơn:"

📌 Bạn chỉ cần dán nội dung vào, mình sẽ viết lại giúp ngay.

**You said:**

"Ubuntu is a free open-source Linux based operating system that is widely used for desktop, server and cloud platforms. It was developed and maintained by Canonical Ltd, and was the first released in October 2004. Ubuntu is based on Debian, another well-known Linux distribution but focuses on usability, regular updates and ease of installation. Ubuntu editions include: ➢ Ubuntu desktop: For personal computers and laptops ➢ Ubuntu server: Designed for server enviroments, cloud computing and data centers ➢ Ubuntu core: Minimal version focused on IoT and containerized applications "

**ChatGPT said:**

Dưới đây là phiên bản **viết lại chi tiết, đầy đủ và rõ ràng hơn** của đoạn văn bạn cung cấp. Mình đã diễn đạt lại bằng tiếng Anh chuẩn, đồng thời bổ sung thêm thông tin và chuyển ngữ mượt mà để dễ hiểu hơn, đặc biệt hữu ích cho người mới học về Ubuntu.

**✅ Bản viết lại chi tiết:**

**Ubuntu** is a **free and open-source** operating system built on the **Linux kernel**, and it is one of the most widely used Linux distributions across various platforms, including **personal desktops, servers, and cloud environments**. It provides a flexible, secure, and stable environment for both personal and professional computing needs.

The development and maintenance of Ubuntu are managed by **Canonical Ltd.**, a company based in the United Kingdom. Ubuntu was **first released in October 2004**, and since then, it has become one of the most popular choices for Linux users worldwide.

Ubuntu is **based on Debian**, a long-standing and reliable Linux distribution. However, unlike Debian, Ubuntu emphasizes:

* **User-friendliness**: It offers a more intuitive interface and user experience.
* **Regular updates**: Ubuntu follows a predictable release schedule, with new versions released **every six months**, and long-term support (LTS) versions every **two years**.
* **Ease of installation**: Ubuntu includes graphical installers and out-of-the-box hardware support, making it easier to set up than many other Linux distributions.

To suit different use cases, Ubuntu is available in several **editions**, each tailored to specific environments and needs:

**🔹 Ubuntu Desktop**

This edition is designed for **personal computers and laptops**.  
It includes a full graphical user interface (GUI) using the **GNOME desktop environment**, and comes preloaded with essential applications like web browsers, office tools, and media players. It is ideal for users who want a free, open-source alternative to Windows or macOS.

**🔹 Ubuntu Server**

This edition is built specifically for **server environments**, including **data centers**, **web hosting**, and **cloud-based platforms**.  
Unlike Ubuntu Desktop, Ubuntu Server **does not include a GUI by default**,

**You said:**

Dựa vào nội dung sau, viết lại tiếng Việt cho mình chi tiết, đầy đủ, rõ ràng hơn: "Ubuntu is a free open-source Linux based operating system that is widely used for desktop, server and cloud platforms. It was developed and maintained by Canonical Ltd, and was the first released in October 2004. Ubuntu is based on Debian, another well-known Linux distribution but focuses on usability, regular updates and ease of installation. Ubuntu editions include: ➢ Ubuntu desktop: For personal computers and laptops ➢ Ubuntu server: Designed for server enviroments, cloud computing and data centers ➢ Ubuntu core: Minimal version focused on IoT and containerized applications "

**ChatGPT said:**

Dưới đây là **bản dịch tiếng Việt** được viết lại **chi tiết, đầy đủ và rõ ràng hơn** từ đoạn văn gốc mà bạn cung cấp:

**🔷 Giới thiệu về Ubuntu**

**Ubuntu** là một hệ điều hành **miễn phí** và **mã nguồn mở**, được xây dựng dựa trên nền tảng **Linux**. Đây là một trong những hệ điều hành phổ biến nhất hiện nay, được sử dụng rộng rãi trên nhiều loại nền tảng khác nhau, bao gồm:

* **Máy tính cá nhân** (desktop)
* **Máy chủ** (server)
* **Hạ tầng điện toán đám mây** (cloud computing)

Ubuntu không chỉ phục vụ nhu cầu người dùng thông thường mà còn đáp ứng tốt các yêu cầu khắt khe trong môi trường doanh nghiệp, trung tâm dữ liệu và công nghệ mới nổi như IoT (Internet of Things).

**🔷 Lịch sử và nhà phát triển**

Ubuntu được phát triển và duy trì bởi công ty **Canonical Ltd**, một doanh nghiệp công nghệ có trụ sở tại Anh. Phiên bản đầu tiên của Ubuntu được phát hành vào **tháng 10 năm 2004**. Từ đó đến nay, Ubuntu đã không ngừng phát triển và trở thành một trong những hệ điều hành Linux dễ sử dụng nhất.

**🔷 Nền tảng Debian**

Ubuntu được xây dựng dựa trên **Debian**, một bản phân phối Linux nổi tiếng khác. Tuy nhiên, so với Debian – vốn nổi tiếng với độ ổn định nhưng khá “khó nhằn” với người mới – Ubuntu tập trung mạnh vào:

* **Tính dễ sử dụng**: Giao diện thân thiện, dễ tiếp cận cho cả người mới và người dùng chuyên sâu.
* **Cập nhật định kỳ**: Ubuntu có chu kỳ phát hành rõ ràng (2 lần mỗi năm), giúp người dùng luôn tiếp cận công nghệ mới.
* **Dễ cài đặt**: Quá trình cài đặt Ubuntu đơn giản, nhanh chóng, hỗ trợ tốt cho cả cài đặt thủ công và tự động.

**🔷 Các phiên bản của Ubuntu**

Ubuntu được phân phối dưới nhiều **phiên bản (editions)** khác nhau, phù hợp với từng mục đích sử dụng cụ thể:

**➤ Ubuntu Desktop**

Dành cho **máy tính cá nhân** và **máy tính xách tay (laptop)**.  
Phiên bản này có giao diện đồ họa trực quan, dễ sử dụng, phù hợp với người dùng thông thường thay thế cho Windows hoặc macOS.

**➤ Ubuntu Server**

Dành cho các **môi trường máy chủ**, **điện toán đám mây** và **trung tâm dữ liệu**.  
Phiên bản này không có giao diện đồ họa mặc định, tập trung vào hiệu suất, tính ổn định và bảo mật cho các hệ thống backend hoặc dịch vụ web.

**➤ Ubuntu Core**

Là phiên bản **tối giản**, nhẹ và bảo mật cao, được thiết kế dành riêng cho các thiết bị **IoT (Internet of Things)** và các **ứng dụng dạng container** (như Docker, Snap).  
Phiên bản này rất phù hợp cho các thiết bị thông minh, nhúng, hoặc hệ thống tự động hóa nhỏ gọn.

**✅ Tóm lại**

Ubuntu là một hệ điều hành mạnh mẽ, miễn phí và dễ sử dụng, phù hợp với cả cá nhân lẫn doanh nghiệp. Nhờ vào sự linh hoạt và hỗ trợ tốt từ cộng đồng, Ubuntu ngày càng được tin dùng trong nhiều lĩnh vực từ máy tính để bàn đến hạ tầng đám mây và thiết bị IoT.

Bạn cần mình chuyển thành file Word hoặc PDF không? Mình có thể giúp luôn.

**You said:**

Tôi vẫn chưa hiểu lý do tại sao đổi hostname lại liên quan đến Cập nhật file /etc/hosts vì tôi là người mới bắt đầu học từ con số 0, nên không hiểu rõ ngọn ngành gốc gác nó như thế nào cả

**ChatGPT said:**

Câu hỏi của bạn rất hay — và quan trọng với người mới bắt đầu học Linux.

Việc **đổi hostname** và **cập nhật file /etc/hosts** có **mối liên hệ chặt chẽ**, đặc biệt là trong cách hệ điều hành **nhận diện chính nó** và **giao tiếp mạng nội bộ**.

Mình sẽ **giải thích từ gốc đến ngọn**, đúng kiểu người mới bắt đầu từ con số 0 cũng hiểu được. Bắt đầu nha 👇

**🔸 1. Hostname là gì?**

**👉 "Hostname" là tên định danh của máy tính trong mạng.**

* Khi bạn cài Ubuntu, bạn sẽ đặt một cái tên cho máy, ví dụ: ubuntu-server1
* Tên này dùng để:
  + Hiển thị trên dòng lệnh: lebao@ubuntu-server1:~$
  + Giao tiếp với các máy khác trong mạng (giống như tên trong mạng LAN)
  + Ghi log, chạy dịch vụ, quản lý qua SSH, v.v.

🧠 Bạn có thể hiểu đơn giản:

Hostname giống như **tên của con người** để người khác gọi/nhận diện, thay vì dùng số chứng minh nhân dân (IP) khô khan.

**🔸 2. File /etc/hosts là gì?**

**👉 /etc/hosts là một “sổ danh bạ” nội bộ của hệ điều hành.**

* Nó ánh xạ (map) giữa **tên** (hostname) và **địa chỉ IP**.
* Khi bạn dùng một tên máy (hostname), hệ thống sẽ **tra cứu file /etc/hosts** để biết tên đó ứng với IP nào — **trước khi hỏi DNS bên ngoài**.

**🔍 Ví dụ nội dung của /etc/hosts:**

127.0.0.1 localhost

127.0.1.1 ubuntu-server1

* 127.0.0.1: địa chỉ loopback (máy tự gọi chính nó)
* ubuntu-server1: hostname của máy

🧠 Nói cách khác:

**Khi bạn gõ ping ubuntu-server1**, hệ điều hành sẽ xem trong file /etc/hosts để biết ubuntu-server1 là địa chỉ IP nào.

**🔸 3. Tại sao đổi hostname cần cập nhật /etc/hosts?**

**Vì:**

* Nếu bạn đổi hostname từ ubuntu-server1 thành web-server
* Mà file /etc/hosts vẫn giữ dòng:
* 127.0.1.1 ubuntu-server1
* 👉 Thì hệ điều hành **không thể tự nhận diện tên mới** web-server.

📌 Kết quả là:

* Một số chương trình sẽ **không hoạt động đúng** (vì hostname không khớp)
* Có thể gặp lỗi khi khởi động dịch vụ (Apache, MySQL…)
* Gõ ping web-server có thể báo lỗi “host not found”

**🔧 Vì vậy bạn cần sửa lại /etc/hosts:**

**Trước:**

127.0.1.1 ubuntu-server1

**Sau khi đổi hostname:**

127.0.1.1 web-server

✅ Khi đó:

* Hệ điều hành sẽ hiểu web-server chính là máy này.
* Các dịch vụ sẽ hoạt động bình thường.
* Các lệnh như ping, hostname, v.v. sẽ khớp nhau.

**✅ Tóm tắt dễ hiểu:**

| **Thành phần** | **Vai trò** | **Khi đổi hostname thì sao?** |
| --- | --- | --- |
| hostname | Tên máy | Cần đổi bằng hostnamectl |
| /etc/hosts | Sổ địa chỉ tên ↔ IP | Cần sửa tên cũ thành tên mới |
| Không sửa /etc/hosts | Hệ thống không biết tên mới là ai | Dịch vụ có thể lỗi, hostname không khớp |

Nếu bạn muốn mình làm một **minh họa thực tế** (trước – sau khi đổi hostname) hoặc viết **script tự động đổi luôn cả hostname và file hosts**, mình sẵn sàng hỗ trợ!

Bạn thấy phần nào còn mơ hồ không? Cứ hỏi tiếp nhé.

**You said:**

câu lệnh ping ubuntu-server1 có tác dụng gì vậy? tôi gõ và nahasn enter cái nó không dừng lại luôn

**ChatGPT said:**

Rất tốt! Bạn đang **tìm hiểu thực tế** cách lệnh ping hoạt động — đây là một trong những lệnh cơ bản nhưng **rất quan trọng** trong Linux và mạng máy tính nói chung.

Giờ mình sẽ **giải thích chi tiết và dễ hiểu** cho bạn:

**✅ Câu hỏi 1: Lệnh ping ubuntu-server1 có tác dụng gì?**

**👉 Mục đích chính:**

Lệnh ping dùng để:

* **Kiểm tra kết nối mạng** giữa máy của bạn và một máy khác (hoặc chính máy bạn).
* Xác định xem hostname (tên máy) như ubuntu-server1 có được **nhận diện và phân giải thành IP hay không**.

**👇 Khi bạn gõ:**

ping ubuntu-server1

Hệ điều hành sẽ:

1. Tìm xem ubuntu-server1 có ánh xạ với IP nào (trong /etc/hosts hoặc DNS).
2. Gửi một gói tin **ICMP echo request** đến IP đó.
3. Chờ phản hồi (ICMP echo reply).
4. In ra kết quả từng dòng, ví dụ:

PING ubuntu-server1 (127.0.1.1) 56(84) bytes of data.

64 bytes from ubuntu-server1 (127.0.1.1): icmp\_seq=1 ttl=64 time=0.045 ms

64 bytes from ubuntu-server1 (127.0.1.1): icmp\_seq=2 ttl=64 time=0.046 ms

...

🧠 Nếu bạn thấy dòng như trên → **ubuntu-server1 có tồn tại và máy đang kết nối được với nó.**

**✅ Câu hỏi 2: Tại sao lệnh ping không dừng lại?**

**👉 Trên Linux, lệnh ping mặc định sẽ chạy mãi không dừng — trừ khi bạn bấm thủ công để dừng.**

**✅ Cách dừng lệnh ping:**

➡️ **Nhấn Ctrl + C**  
→ Lệnh sẽ dừng lại và hiển thị **thống kê kết quả**, ví dụ:

--- ubuntu-server1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4095ms

**✅ Mẹo nhỏ: Ping trong thời gian hoặc số lần nhất định**

**🔸 Ping chỉ 4 lần:**

ping -c 4 ubuntu-server1

→ Máy sẽ gửi 4 gói ping, rồi tự dừng.

**✅ Tóm lại**

| **Câu hỏi** | **Trả lời** |
| --- | --- |
| ping ubuntu-server1 để làm gì? | Kiểm tra xem máy có nhận diện được hostname ubuntu-server1 và có kết nối tới nó hay không |
| Tại sao không dừng? | Trên Linux, ping chạy liên tục cho đến khi bạn **nhấn Ctrl + C** |
| Muốn nó tự dừng thì sao? | Dùng ping -c [số\_lần], ví dụ ping -c 4 ubuntu-server1 |

Nếu bạn gõ ping ubuntu-server1 mà nó báo lỗi kiểu như "ping: ubuntu-server1: Name or service not known" thì có thể là:

* Bạn **chưa cập nhật hostname mới trong /etc/hosts**.
* Hostname bị sai chính tả.
* Không có mạng hoặc firewall chặn.

Bạn muốn mình giúp test điều đó không?

**You said:**

cả một cụm này gọi là gì? "lebao@ubuntu-server1:~$?"

**ChatGPT said:**

Rất hay — cụm "lebao@ubuntu-server1:~$" có một tên gọi chính thức trong thế giới Linux/Unix.

**✅ Tên gọi chính xác: Shell Prompt (hoặc đơn giản là Prompt)**

**👉 "Prompt" nghĩa là dấu nhắc lệnh, là phần:**

* Hiển thị trước mỗi dòng nhập lệnh trong terminal.
* Cho bạn biết **ai đang đăng nhập**, **đang ở đâu**, và **máy nào**.

**✅ Giải thích từng phần của lebao@ubuntu-server1:~$**

lebao@ubuntu-server1:~$

| | | |

| | | └── Dấu kết thúc prompt: `$` (user thường), `#` (root)

| | └── Thư mục hiện tại: `~` là thư mục home

| └── Tên máy (hostname): ubuntu-server1

└── Tên người dùng: lebao

**✅ Cụ thể:**

| **Thành phần** | **Ý nghĩa** |
| --- | --- |
| lebao | Tên người dùng đang đăng nhập |
| @ | Phân tách giữa user và hostname |
| ubuntu-server1 | Tên máy (hostname) đang sử dụng |
| : | Phân tách giữa tên máy và thư mục |
| ~ | Thư mục hiện tại (~ là viết tắt cho thư mục home của user) |
| $ | Dấu nhắc lệnh, báo hiệu hệ thống đang chờ bạn nhập lệnh |
| # | Nếu là root user, dấu nhắc sẽ là # thay vì $ |

**🔁 Tùy chỉnh prompt**

Bạn có thể thay đổi cách hiển thị prompt bằng biến môi trường PS1. Ví dụ:

export PS1="\u@\h:\w$ "

* \u: username
* \h: hostname
* \w: thư mục hiện tại

⟶ sẽ hiển thị giống như: lebao@ubuntu-server1:~/Documents$

**✅ Kết luận:**

Cụm "lebao@ubuntu-server1:~$" được gọi là **shell prompt** hay **command prompt**, và nó giúp bạn biết **ai đang làm việc**, **trên máy nào**, **ở đâu**, và **sẵn sàng chờ lệnh**.

Bạn có thể thử gõ:

echo $PS1

Để xem cấu trúc prompt hiện tại của bạn được định nghĩa thế nào.

Muốn mình giúp bạn **tùy chỉnh prompt cho đẹp hơn** hoặc **màu mè dễ nhìn** không?

**You said:**

root user và user là gì? nghĩa là sao?

**ChatGPT said:**

Câu hỏi rất quan trọng và là nền tảng để bạn học Linux chắc chắn từ đầu:

**Root user và user là gì? Nghĩa là sao?**

Mình sẽ **giải thích từ cơ bản đến rõ ràng**, đúng kiểu "người mới học từ con số 0" vẫn hiểu được dễ dàng.

**✅ 1. "User" (người dùng) trong hệ điều hành là gì?**

Hệ điều hành như Linux (hoặc Windows, macOS…) luôn cần có **người dùng (user)** để đăng nhập và sử dụng.

* Mỗi user có **tài khoản riêng**, **mật khẩu riêng** và **quyền hạn riêng**.
* Khi bạn đăng nhập vào Ubuntu với tên như lebao, thì bạn đang dùng máy với tư cách **một người dùng (user)**.

**✅ 2. Có mấy loại user trong Linux?**

Trong Linux, có 2 loại chính:

| **Loại** | **Mô tả** | **Dấu hiệu** |
| --- | --- | --- |
| **User thường** *(normal user)* | Dùng để làm việc, học tập, lướt web, lập trình, v.v. | Dấu nhắc lệnh là: $ |
| **Root user** *(siêu người dùng)* | Là **quản trị viên hệ thống** — có quyền cao nhất, làm gì cũng được | Dấu nhắc lệnh là: # |

**✅ 3. Root user là gì?**

**➤ root là tài khoản cao nhất trong Linux**

* Có thể **làm mọi việc**: cài phần mềm, xóa hệ thống, tạo người dùng, chỉnh sửa file hệ thống...
* Không bị giới hạn bởi quyền gì hết
* Giống như **admin tuyệt đối** vậy đó

🧠 Trong Windows thì giống như **Administrator**.

**⚠️ Cảnh báo:**

Làm việc với tài khoản root rất **nguy hiểm** nếu bạn không cẩn thận. Chỉ cần một lệnh sai cũng có thể:

* Xóa toàn bộ hệ thống
* Làm hỏng Ubuntu
* Mất dữ liệu

**✅ 4. User thường là gì?**

* Là người dùng thông thường (như bạn khi đăng nhập bằng lebao).
* Không có quyền can thiệp vào phần hệ thống.
* Muốn làm gì "nặng đô" (ví dụ: cài phần mềm, chỉnh file hệ thống) thì phải:  
  👉 dùng lệnh **sudo**

**✅ 5. sudo là gì?**

sudo = **SuperUser DO**  
→ Tạm hiểu là: *“Làm việc này với quyền root.”*

**Ví dụ:**

| **Mục đích** | **Câu lệnh** |
| --- | --- |
| Chạy bình thường | apt update → ❌ Bị từ chối |
| Chạy với root | sudo apt update → ✅ Chạy được |

Hệ thống sẽ yêu cầu bạn nhập mật khẩu → nếu đúng, thì bạn **tạm thời được mượn quyền root** để chạy lệnh.

**✅ 6. Phân biệt nhanh:**

| **Tài khoản** | **Dấu hiệu** | **Quyền hạn** | **Rủi ro** |
| --- | --- | --- | --- |
| **root** | # | Toàn quyền hệ thống | Rất cao nếu dùng sai |
| **user thường (ví dụ: lebao)** | $ | Bị giới hạn | An toàn hơn |
| **sudo** | (lệnh đặc biệt cho user thường) | Tạm mượn quyền root | Trung bình |

**🔍 Ví dụ thực tế:**

**Bạn là user thường:**

lebao@ubuntu-server1:~$ sudo apt install nginx

→ Máy hỏi mật khẩu của lebao. Nếu đúng, lệnh được chạy với quyền root.

**✅ Tóm lại:**

* **User**: Người dùng thông thường, làm việc, học tập, không có quyền can thiệp hệ thống.
* **Root user**: Quản trị viên toàn quyền hệ thống. Có thể làm mọi thứ, kể cả xóa sạch hệ thống.
* **sudo**: Lệnh cho phép user thường tạm "mượn" quyền root để chạy lệnh nâng cao.

Nếu bạn muốn mình chỉ luôn cách **chuyển từ user thường sang root**, hoặc **tạo user mới**, mình có thể giúp thêm nhé!

**You said:**

Vậy mình muốn bạn hướng dẫn chi tiết cụ thể về "chuyển từ user thường sang root"

**ChatGPT said:**

Rất sẵn lòng! Đây là một **bước rất quan trọng** để bạn làm chủ Linux, nên mình sẽ hướng dẫn bạn thật chi tiết, từng bước, đúng chuẩn "mới bắt đầu từ con số 0" cũng làm được.

**🎯 Mục tiêu:**

**Chuyển từ user thường (ví dụ: lebao) sang tài khoản root** trong Ubuntu Server

**✅ Cách 1: Chuyển tạm sang root bằng sudo -i**

**🧪 Dùng khi nào?**

* Khi bạn chỉ cần **vào chế độ root tạm thời** để làm gì đó.
* Thoát ra là quay lại user bình thường.

**🧱 Câu lệnh:**

sudo -i

**📌 Giải thích:**

* sudo = mượn quyền root
* -i = mở **môi trường shell** giống như bạn là root thật

**🖥️ Ví dụ:**

lebao@ubuntu-server1:~$ sudo -i

[sudo] password for lebao: ← Bạn nhập mật khẩu của user `lebao`

root@ubuntu-server1:~# ← Bạn đã thành root rồi!

✅ Từ đây bạn có thể chạy lệnh mà không cần gõ sudo nữa, ví dụ:

apt update

apt install nginx

**🔁 Muốn thoát khỏi root?**

exit

→ Bạn sẽ quay lại với user lebao.

**✅ Cách 2: Chạy từng lệnh với sudo (an toàn hơn)**

**🧪 Dùng khi:**

* Bạn chỉ muốn chạy 1–2 lệnh với quyền root, không cần vào hẳn tài khoản root.

**Ví dụ:**

sudo apt update

sudo systemctl restart nginx

📌 Đây là cách **an toàn hơn**, vì bạn không ở trong môi trường root quá lâu.

**✅ Cách 3: Đăng nhập hẳn vào root (ít dùng)**

**🛑 Chỉ dùng khi bạn đã thiết lập mật khẩu cho tài khoản root**

Mặc định Ubuntu **không cho đăng nhập root trực tiếp** vì lý do bảo mật.

**📌 Bước 1: Đặt mật khẩu cho root (chỉ cần làm 1 lần)**

sudo passwd root

→ Hệ thống sẽ yêu cầu bạn nhập **mật khẩu mới cho root**

**📌 Bước 2: Đăng nhập root**

su -

→ Nhập mật khẩu root → vào root shell

**Kết quả:**

root@ubuntu-server1:~#

📌 su = switch user  
su - = switch sang user root và tải môi trường của root

**💡 Bạn nên dùng cách nào?**

| **Mục đích** | **Cách nên dùng** | **Ghi chú** |
| --- | --- | --- |
| Dùng tạm root để làm vài lệnh | sudo -i | Rất tiện, dễ thoát |
| Chạy 1 lệnh cần quyền root | sudo lệnh | An toàn nhất |
| Đăng nhập hẳn vào root | su - (sau khi passwd root) | Nguy hiểm hơn, ít khuyến khích |

**⚠️ Lưu ý bảo mật cực quan trọng**

* **Không nên làm mọi việc bằng tài khoản root**
* **Chỉ dùng root khi thực sự cần**
* Cẩn thận với lệnh như: rm -rf /, chmod -R, chown, mv, v.v.

Nếu bạn muốn, mình có thể:

* Kiểm tra xem tài khoản bạn đang dùng đã được cấp quyền sudo chưa.
* Hướng dẫn cách **thêm người dùng mới và cấp quyền root** luôn.

Bạn muốn tiếp tục theo hướng nào?

**You said:**

Cho mình hỏi ở cách 3 khi đã Đăng nhập hẳn vào root thì sau thoát ra như nào

**ChatGPT said:**

Câu hỏi rất đúng lúc! Khi bạn đã **đăng nhập hẳn vào tài khoản root** (theo **cách 3**) — tức là đang dùng tài khoản có quyền **quản trị hệ thống tối cao** — thì việc **thoát ra đúng cách** cũng rất quan trọng để tránh thao tác nhầm, gây lỗi hệ thống.

**✅ Cách thoát khỏi root và quay lại user thường**

**🧱 Lệnh duy nhất bạn cần:**

exit

**📌 Ví dụ thực tế:**

root@ubuntu-server1:~# exit

logout

lebao@ubuntu-server1:~$

* Bạn sẽ thấy từ dòng # (root), chuyển lại về $ (user thường).
* Dấu nhắc shell sẽ quay lại giống lúc đầu: lebao@ubuntu-server1:~$

**🎯 Dùng exit khi nào?**

| **Tình huống** | **Bạn dùng exit để…** |
| --- | --- |
| Đang trong shell root (root@...#) | Thoát khỏi quyền root |
| Đang ở trong sudo -i | Thoát về user thường |
| Đang dùng su - | Đăng xuất khỏi tài khoản root |

**📝 Mẹo thêm:**

Nếu bạn **đăng nhập SSH vào máy chủ** với tài khoản root, thì gõ exit lần đầu sẽ **thoát khỏi root**,  
gõ exit thêm lần nữa sẽ **thoát khỏi phiên SSH hoàn toàn** (đóng terminal).

**✅ Tóm lại:**

Sau khi đăng nhập hẳn vào root (dùng sudo -i hoặc su -), bạn có thể **gõ lệnh exit** để:

* Thoát khỏi shell root
* Trở về user thường
* Hạn chế thao tác nguy hiểm do vô tình thực hiện với quyền root

Bạn có thể thử:

sudo -i # hoặc su -

exit # quay về user thường

Muốn mình chỉ bạn luôn cách **kiểm tra đang ở tài khoản nào** hay **chuyển đổi nhanh giữa người dùng** không?

**You said:**

Vậy mình muốn bạn hướng dẫn chi tiết cụ thể về "tạo user mới và phân quyền gì gì đó"

**ChatGPT said:**

Tuyệt vời! Đây là bước quan trọng nếu bạn muốn **quản lý nhiều người dùng trên một máy Linux**, hoặc muốn **tạo user mới có quyền quản trị (sudo)**.

Mình sẽ hướng dẫn bạn thật chi tiết, dễ hiểu, **từng bước một**, đúng chuẩn **"người mới bắt đầu từ số 0"** cũng làm được.

**🎯 Mục tiêu:**

Hướng dẫn **tạo user mới trên Ubuntu Server** và **phân quyền sudo** (quyền quản trị)

**✅ Bước 1: Tạo user mới**

**📌 Cú pháp:**

sudo adduser [tên\_user\_mới]

**📥 Ví dụ:**

sudo adduser tan

Hệ thống sẽ yêu cầu bạn nhập:

* Mật khẩu cho user tan
* (Tùy chọn) Họ tên, số điện thoại, thông tin cá nhân (có thể bỏ qua)

**🧠 Lưu ý:**

* Thư mục home sẽ tự động tạo: /home/tan
* Sau khi tạo, user mới vẫn là **user thường**, chưa có quyền sudo

**✅ Bước 2: Cấp quyền sudo cho user mới**

**📌 Cách nhanh & chuẩn nhất:**

sudo usermod -aG sudo [tên\_user\_mới]

**📥 Ví dụ:**

sudo usermod -aG sudo tan

* usermod: lệnh để chỉnh sửa user
* -aG: thêm user vào một nhóm (G = group)
* sudo: là nhóm đặc biệt có quyền quản trị
* tan: tên user mới

➡️ Sau lệnh này, user tan sẽ có quyền chạy các lệnh như:

sudo apt update

sudo shutdown now

**✅ Bước 3: (Tuỳ chọn) Kiểm tra xem user mới có quyền sudo chưa**

**📥 Cách kiểm tra:**

1. Đăng nhập vào tài khoản tan:
2. su - tan
3. Gõ thử:
4. sudo whoami

→ Nhập mật khẩu của tan  
→ Nếu thấy kết quả là:

root

→ 🎉 User tan đã có quyền sudo!

**✅ Bước 4: (Tùy chọn) Xem danh sách user trên hệ thống**

cut -d: -f1 /etc/passwd

**✅ Bước 5: (Tùy chọn) Xóa user nếu không cần nữa**

**❗ Cẩn thận khi dùng!**

* Xóa user:
* sudo deluser tan
* Xóa luôn cả thư mục home:
* sudo deluser --remove-home tan

**✅ Tổng kết các lệnh quan trọng:**

| **Hành động** | **Lệnh** |
| --- | --- |
| Tạo user mới | sudo adduser tên\_user |
| Thêm quyền sudo | sudo usermod -aG sudo tên\_user |
| Đăng nhập user mới | su - tên\_user |
| Kiểm tra quyền sudo | sudo whoami |
| Xóa user | sudo deluser tên\_user |

Nếu bạn muốn:

* Tạo user **chỉ được dùng một số lệnh sudo**
* Giới hạn quyền hoặc thư mục
* Tự động tạo user bằng script

→ Mình có thể hướng dẫn nâng cao hơn.  
Bạn muốn thử tiếp phần nào?

giúp mình tùy chỉnh prompt cho đẹp hơn hoặc màu mè dễ nhìn

**ChatGPT said:**

Tuyệt rồi! Tùy chỉnh **prompt (dòng lệnh)** là một bước nhỏ nhưng cực kỳ hữu ích:  
✅ Giúp dễ nhìn,  
✅ Biết mình đang ở đâu, là ai,  
✅ Tránh nhầm lẫn giữa user thường và root,  
✅ Tạo trải nghiệm cá nhân hóa thú vị hơn trong terminal.

**🎯 Mục tiêu**

Tùy chỉnh **dòng prompt** (lebao@ubuntu-server1:~$) thành phiên bản **đẹp hơn, màu mè, dễ nhìn**.

**✅ 1. Cách prompt hoạt động: biến $PS1**

Prompt được điều khiển bởi biến môi trường tên là PS1.

**📥 Kiểm tra hiện tại:**

echo $PS1

**✅ 2. Các thành phần có thể dùng trong prompt**

| **Mã** | **Ý nghĩa** |
| --- | --- |
| \u | Tên user hiện tại |
| \h | Tên máy (hostname) |
| \w | Thư mục hiện tại |
| \W | Tên thư mục hiện tại (không có đường dẫn đầy đủ) |
| \$ | Hiện $ nếu là user thường, # nếu là root |
| \t | Giờ hiện tại (HH:MM:SS) |
| \d | Ngày hiện tại |
| \n | Xuống dòng mới |
| \[ và \] | Để bọc các mã màu ANSI (bắt buộc khi thêm màu) |

**✅ 3. Màu sắc ANSI (dùng được trong prompt)**

| **Màu** | **Mã ANSI** |
| --- | --- |
| Đen | \e[0;30m |
| Đỏ | \e[0;31m |
| Xanh lá | \e[0;32m |
| Vàng | \e[0;33m |
| Xanh dương | \e[0;34m |
| Tím | \e[0;35m |
| Xanh cyan | \e[0;36m |
| Trắng | \e[0;37m |
| Reset | \e[0m |

Để dùng trong PS1, bạn cần bọc bằng \[ và \].  
Ví dụ: \[\e[0;32m\] → bật màu xanh lá

**✅ 4. Ví dụ prompt màu sắc đẹp — bạn có thể dùng ngay!**

**🌈 Prompt gợi ý (đẹp, đầy đủ, dễ nhìn):**

PS1='\[\e[1;32m\]\u@\h\[\e[0m\]:\[\e[1;34m\]\w\[\e[0m\]\$ '

**🧠 Giải thích:**

* \u@\h: tên user và hostname → **màu xanh lá**
* \w: thư mục hiện tại → **màu xanh dương**
* \$: hiện $ hoặc # tùy quyền
* \[\e[0m\]: reset về màu mặc định

**✅ 5. Thử nghiệm ngay (tạm thời)**

Gõ lệnh sau để thử ngay (chỉ hiệu lực **tạm thời** trong terminal hiện tại):

PS1='\[\e[1;32m\]\u@\h\[\e[0m\]:\[\e[1;34m\]\w\[\e[0m\]\$ '

➡️ Prompt sẽ đổi màu ngay lập tức 🎉

**✅ 6. Lưu lại vĩnh viễn**

Để không bị mất khi tắt máy hoặc mở terminal mới, bạn cần **lưu dòng PS1** vào file cấu hình shell.

**✅ Nếu bạn dùng shell mặc định là bash, hãy làm như sau:**

nano ~/.bashrc

* Tìm dòng có PS1= (hoặc thêm mới nếu chưa có)
* Dán dòng sau vào cuối file:

PS1='\[\e[1;32m\]\u@\h\[\e[0m\]:\[\e[1;34m\]\w\[\e[0m\]\$ '

* Lưu file: nhấn Ctrl + O, Enter, rồi Ctrl + X để thoát
* Cuối cùng, chạy:

source ~/.bashrc

→ Để áp dụng ngay cấu hình mới

**✅ 7. Bonus: prompt có ngày giờ**

Nếu bạn muốn hiển thị thêm thời gian:

PS1='\[\e[0;36m\]\t \[\e[1;32m\]\u@\h:\[\e[0;34m\]\w\[\e[0m\]\$ '

📌 Ví dụ hiển thị:

14:25:36 lebao@ubuntu-server1:~/projects$

**🛠 Nếu đang dùng Zsh, mình có thể hướng dẫn prompt đẹp kiểu “Oh My Zsh” luôn nhé!**

Bạn muốn mình giúp chỉnh theo kiểu đẹp hơn nữa (nhiều dòng, có biểu tượng git, lỗi/success,...) không?  
Hay bạn đang dùng shell khác như Zsh hoặc Fish?

**You said:**

mình : nhập nano ~/.bashrc và nhấn enter thì nó hiện như này và làm sao nữa GNU nano 7.2 /home/le #~/.bashrc: executed by bash (1) for non-login shells. # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc) # for examples # If not running interactively, don't do anything case $- in \*i\*) ;; \*) return;; esac # don't put duplicate lines or lines starting with space in the history. # See bash (1) for more options HISTCONTROL=ignoreboth # append to the history file, don't overwrite it shopt -s histappend # for setting history length see HISTSIZE and HISTFILESIZE in bash (1) HISTSIZE=1000 HISTFILESIZE=2000 # check the window size after each command and, if necessary, # update the values of LINES and COLUMNS. shopt -s checkwinsize # If set, the pattern "\*\*" used in a pathname expansion context will # match all files and zero or more directories and subdirectories. #shopt -s globstar # make less more friendly for non-text input files, see lesspipe (1) [ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)" # set variable identifying the chroot you work in (used in the prompt below) if [ -z "${debian\_chroot:-}" ] && [ -r /etc/debian\_chroot]; then debian\_chroot=$(cat /etc/debian\_chroot) fi # set a fancy prompt (non-color, unless we know we "want" color) case "$TERM" in xterm-color\*-256color) color\_prompt=yes;; esac # uncomment for a colored prompt, if the terminal has the capability; turned # off by default to not distract the user: the focus in a terminal window # should be on the output of commands, not on the prompt #force\_color\_prompt=yes

**ChatGPT said:**

Bạn đang mở file ~/.bashrc bằng nano và thấy nội dung như trên là hoàn toàn bình thường nhé! Đây là file cấu hình mặc định của bash shell, dùng để thiết lập nhiều thứ khi bạn mở terminal.

**Hướng dẫn thao tác tiếp theo trong nano để chỉnh prompt:**

**1. Di chuyển xuống cuối file**

* Dùng phím **mũi tên xuống** (↓) trên bàn phím, bạn kéo xuống tận cuối file.

**2. Thêm dòng cấu hình prompt vào cuối file**

Dán đoạn này vào cuối file (bạn có thể nhấn chuột phải → Paste hoặc gõ thủ công):

PS1='\[\e[1;32m\]\u@\h\[\e[0m\]:\[\e[1;34m\]\w\[\e[0m\]\$ '

Đoạn này sẽ làm prompt của bạn đổi thành màu xanh lá phần user@host, màu xanh dương phần thư mục.

**3. Lưu file và thoát nano**

* Nhấn **Ctrl + O** (phím Control + O): để lưu file
* Nano sẽ hỏi tên file để lưu, bạn nhấn **Enter** luôn (để giữ nguyên file hiện tại)
* Sau đó nhấn **Ctrl + X** để thoát nano.

**4. Áp dụng thay đổi ngay**

Sau khi thoát nano, bạn chạy lệnh sau trong terminal:

source ~/.bashrc

Lệnh này sẽ nạp lại file ~/.bashrc và bạn sẽ thấy prompt thay đổi màu sắc ngay lập tức.

**Nếu bạn làm theo các bước trên mà không thấy prompt đổi màu, hãy báo mình biết nhé, mình giúp bạn sửa tiếp!**

**You said:**

Mình muốn chỉnh giờ theo hà nội việt nam á

**ChatGPT said:**

Tuyệt! Mình sẽ hướng dẫn bạn cách **thiết lập múi giờ Hà Nội (Việt Nam)** trên Ubuntu Server, từng bước một, rất dễ làm.

**🎯 Mục tiêu:**

Chuyển hệ thống Ubuntu về đúng múi giờ Việt Nam: **Asia/Ho\_Chi\_Minh** (GMT+7)

**✅ Cách làm: Dùng lệnh timedatectl**

**🔧 Bước 1: Kiểm tra múi giờ hiện tại**

timedatectl

Kết quả có thể giống như:

Local time: 2025-10-15 15:42:00 UTC

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

→ Nếu thấy Time zone: Etc/UTC thì tức là máy bạn đang chạy theo giờ quốc tế (UTC), **lệch 7 tiếng so với Việt Nam**.

**🔧 Bước 2: Xem danh sách các múi giờ có sẵn**

timedatectl list-timezones

Sau đó, bạn có thể cuộn và tìm dòng:

Asia/Ho\_Chi\_Minh

Hoặc lọc nhanh bằng:

timedatectl list-timezones | grep Ho\_Chi\_Minh

**🔧 Bước 3: Thiết lập múi giờ Việt Nam**

sudo timedatectl set-timezone Asia/Ho\_Chi\_Minh

**🔧 Bước 4: Kiểm tra lại**

timedatectl

Kết quả đúng sẽ là:

Local time: 2025-10-15 22:42:00 +07

Time zone: Asia/Ho\_Chi\_Minh (ICT, +0700)

✅ Lúc này máy bạn đã chạy theo giờ Hà Nội rồi.

**🕒 Bonus: Hiển thị giờ trong Prompt (nếu bạn thích)**

Nếu bạn muốn **hiển thị giờ Hà Nội trong prompt dòng lệnh**, bạn có thể sửa lại ~/.bashrc (như đã hướng dẫn ở trên), và dùng dòng sau:

PS1='\[\e[0;36m\]\t \[\e[1;32m\]\u@\h:\[\e[0;34m\]\w\[\e[0m\]\$ '

Trong đó:

* \t: là giờ (HH:MM:SS)
* Giờ hiển thị sẽ **khớp với múi giờ hệ thống** bạn vừa đặt (tức là giờ Việt Nam)

**✅ Tổng kết**

| **Lệnh** | **Tác dụng** |
| --- | --- |
| timedatectl | Xem giờ và múi giờ hệ thống hiện tại |
| timedatectl list-timezones | Xem tất cả múi giờ |
| sudo timedatectl set-timezone Asia/Ho\_Chi\_Minh | Đặt múi giờ Việt Nam |
| source ~/.bashrc | Nạp lại cấu hình prompt sau khi chỉnh |

Bạn làm tới bước nào rồi? Mình có thể kiểm tra giúp nếu có lỗi.

**You said:**

CÁCH GÕ TIẾNG vIỆT TRONG NANO

**ChatGPT said:**

Việc **gõ tiếng Việt có dấu trong trình soạn thảo nano** trên Ubuntu Server không tự động hoạt động, vì nó còn **phụ thuộc vào cấu hình locale, bộ gõ, encoding của terminal**, v.v.

Mình sẽ hướng dẫn bạn cách **thiết lập để gõ và hiển thị tiếng Việt trong nano** một cách chi tiết.

**🎯 Mục tiêu**

Giúp bạn có thể **gõ tiếng Việt có dấu trong nano** trên Ubuntu Server (thường qua SSH hoặc terminal trực tiếp)

**✅ Bước 1: Đảm bảo hệ thống hỗ trợ UTF-8 (để hiển thị và lưu được tiếng Việt)**

Chạy lệnh kiểm tra:

locale

Kết quả lý tưởng bạn cần thấy là:

LANG=en\_US.UTF-8

**📌 Nếu bạn thấy POSIX hoặc C, thì phải cài lại locale UTF-8 như sau:**

sudo apt update

sudo apt install locales

sudo dpkg-reconfigure locales

Chọn:

* en\_US.UTF-8
* hoặc vi\_VN.UTF-8 (nếu bạn muốn cả hệ thống hiển thị tiếng Việt)

Sau đó chọn en\_US.UTF-8 làm mặc định.

**✅ Bước 2: Cài đặt bộ gõ tiếng Việt (nếu dùng trực tiếp trên Ubuntu Desktop)**

Nếu bạn đang dùng Ubuntu **có giao diện đồ họa**, hãy:

1. Cài bộ gõ tiếng Việt:

sudo apt install ibus-unikey

1. Khởi động lại ibus:

ibus restart

1. Mở Settings > Region & Language > Input Source > Thêm **Vietnamese - Unikey**

→ Sau đó bạn có thể dùng tổ hợp **Ctrl + Space** để bật/tắt gõ tiếng Việt.

**✅ Bước 3: Nếu dùng qua SSH từ máy Windows**

📌 Đây là trường hợp phổ biến nhất với Ubuntu Server.

**➤ Bạn phải bật bộ gõ tiếng Việt trên máy Windows, ví dụ:**

* Unikey (phổ biến nhất)
* Gõ kiểu Telex hoặc VNI
* Dùng SSH client như **MobaXterm**, **Termius**, hoặc **Windows Terminal**

➡️ Sau khi bật Unikey → gõ tiếng Việt → dán vào nano là sẽ được!

**✅ Bước 4: Kiểm tra encoding trong nano**

Mở một file bằng nano với chế độ UTF-8:

nano -encoding=utf-8 test.txt

Hoặc thêm dòng sau vào file cấu hình của nano để mặc định dùng UTF-8:

nano ~/.nanorc

Thêm dòng:

set utf8

**✅ Bước 5: Test thử**

Sau khi đã:

* Gõ tiếng Việt trên client (dùng Unikey)
* Terminal/SSH hỗ trợ UTF-8
* Nano set utf8

→ Bạn mở nano và thử gõ:

Tôi đang gõ tiếng Việt có dấu: ạ á ã â đ ê ô ơ ư

Nếu hiển thị đúng → ✅ OK rồi.

**❌ Lỗi thường gặp:**

| **Lỗi** | **Nguyên nhân** | **Cách xử lý** |
| --- | --- | --- |
| Gõ dấu nhưng ra ký tự lạ | Terminal không hỗ trợ UTF-8 | Đổi sang UTF-8, kiểm tra locale |
| Gõ không ra dấu | Unikey chưa bật | Bật Unikey, kiểm tra kiểu gõ |
| Lưu file xong bị lỗi font | File không được lưu bằng UTF-8 | Dùng nano -encoding=utf-8 hoặc set utf8 |

**✅ Tổng kết cách gõ tiếng Việt trong nano:**

1. **Đảm bảo locale hệ thống là UTF-8**
2. **Dùng bộ gõ tiếng Việt (Unikey trên Windows hoặc ibus-unikey trên Ubuntu Desktop)**
3. **Set encoding UTF-8 cho nano**
4. **Gõ hoặc dán tiếng Việt vào file nano**

Bạn đang dùng Ubuntu Server qua SSH từ Windows hay đang làm việc trực tiếp trên Ubuntu?  
Mình có thể hướng dẫn đúng theo trường hợp của bạn nếu bạn nói rõ hơn.

127.0.0.53 là gì

**ChatGPT đã nói:**

**127.0.0.53** là một địa chỉ IP cục bộ (localhost) đặc biệt thường thấy trong các hệ thống **Linux**, đặc biệt là khi sử dụng **systemd-resolved**, một dịch vụ quản lý DNS (hệ thống phân giải tên miền).

**Cụ thể:**

* **127.0.0.53** là một địa chỉ **loopback** giống như **127.0.0.1** – tức là nó trỏ về chính máy tính của bạn.
* Nó thường được cấu hình là **DNS stub resolver** – một máy chủ DNS giả lập được cài đặt sẵn trong hệ thống để **trung chuyển các truy vấn DNS** từ các ứng dụng tới DNS thực tế.

**Vì sao không phải là 127.0.0.1?**

* Trên các hệ thống dùng systemd-resolved, tệp cấu hình /etc/resolv.conf thường sẽ có dòng:
* nameserver 127.0.0.53

Điều này nghĩa là tất cả các truy vấn DNS từ ứng dụng sẽ đi qua systemd-resolved qua địa chỉ loopback này.

* systemd-resolved sau đó sẽ gửi truy vấn ra internet thông qua các máy chủ DNS thực sự (ví dụ: 8.8.8.8 của Google, hoặc máy chủ DNS do DHCP cung cấp).

**Kiểm tra nhanh:**

* Bạn có thể kiểm tra bằng lệnh:
* systemd-resolve --status

hoặc

resolvectl status

* Để xem DNS thực sự mà systemd-resolved đang dùng (bên dưới lớp stub 127.0.0.53).

**Tóm tắt:**

| **Thành phần** | **Giải thích** |
| --- | --- |
| 127.0.0.53 | Stub resolver của systemd-resolved |
| Vai trò | Trung gian xử lý truy vấn DNS |
| Có phải DNS thực tế? | Không. Nó chuyển tiếp truy vấn DNS |
| Có thể thay đổi được? | Có, nhưng không nên nếu không cần |

Nếu bạn đang cấu hình mạng hoặc gặp lỗi DNS, việc hiểu 127.0.0.53 sẽ giúp bạn xác định đúng nơi cần chỉnh sửa (thường là cấu hình của systemd-resolved hoặc resolv.conf).

**Bạn đã nói:**

thế biết 127.0.0.53 cái này rồi, thì nó dùng để làm gì?

**ChatGPT đã nói:**

Biết địa chỉ **127.0.0.53** là để hiểu được **cách hệ điều hành của bạn xử lý truy vấn DNS** – điều rất quan trọng khi:

**✅ 1. Khắc phục sự cố DNS**

Ví dụ:

* Truy cập web chậm, lỗi “không tìm thấy máy chủ”.
* Ping tên miền (như ping google.com) bị lỗi, nhưng ping IP (như ping 8.8.8.8) thì được.

Khi thấy 127.0.0.53 trong /etc/resolv.conf, bạn biết là:

Truy vấn DNS đang được xử lý qua systemd-resolved, chứ không gửi trực tiếp ra DNS ngoài.

Từ đó, bạn có thể:

* Kiểm tra trạng thái với resolvectl status
* Xem DNS thực tế nó đang dùng (có thể do DHCP cung cấp hoặc cấu hình tay).
* Nếu systemd-resolved gặp lỗi, có thể bypass bằng cách sửa /etc/resolv.conf tạm thời trỏ về 8.8.8.8 chẳng hạn.

**✅ 2. Hiểu luồng truy vấn DNS**

Khi một ứng dụng muốn phân giải tên miền (ví dụ google.com):

1. Nó gửi truy vấn DNS tới **127.0.0.53**
2. Đây là stub resolver do systemd-resolved lắng nghe.
3. systemd-resolved sẽ chọn DNS thực tế (do DHCP hoặc cấu hình tay) và gửi truy vấn ra ngoài.
4. Trả kết quả ngược lại cho ứng dụng.

Biết điều này giúp bạn hiểu rõ **DNS không hẳn bị lỗi chỉ vì thấy 127.0.0.53**, mà phải kiểm tra thêm ai đứng sau nó (DNS upstream).

**✅ 3. Cấu hình DNS nâng cao (VPN, split-DNS, chặn quảng cáo,...)**

Khi bạn dùng:

* VPN (WireGuard, OpenVPN)
* DNS chặn quảng cáo (như Pi-hole)
* Chia DNS theo miền (split-DNS)

Thì hiểu vai trò của 127.0.0.53 giúp bạn:

* Điều chỉnh đúng DNS ở layer systemd-resolved.
* Tránh việc sửa trực tiếp /etc/resolv.conf vì nó có thể bị ghi đè.
* Dùng lệnh như resolvectl dns, resolvectl domain để cấu hình theo interface hoặc theo miền.

**✅ Tóm lại**

**127.0.0.53 là "cửa ngõ DNS" của hệ thống bạn.**

Hiểu nó giúp bạn:

| **Việc cần làm** | **Có ích khi hiểu 127.0.0.53 không?** |
| --- | --- |
| Sửa lỗi không truy cập được web | ✅ |
| Kiểm tra hệ thống dùng DNS nào | ✅ |
| Cấu hình DNS cho VPN/phần mềm khác | ✅ |
| Tránh bị hiểu nhầm là lỗi DNS | ✅ |
| Tăng tốc độ DNS, tránh DNS rò rỉ | ✅ |

Nếu bạn muốn, mình có thể hướng dẫn cách thay đổi DNS thực sự đằng sau 127.0.0.53 theo cách chính xác.

**Bạn đã nói:**

default gateway của tôi là cái nào vậy? Windows IP Configuration Ethernet adapter Ethernet: Media State . . . . . . . . . . . : Media disconnected Connection-specific DNS Suffix . : Wireless LAN adapter Local Area Connection\* 1: Media State . . . . . . . . . . . : Media disconnected Connection-specific DNS Suffix . : Wireless LAN adapter Local Area Connection\* 2: Media State . . . . . . . . . . . : Media disconnected Connection-specific DNS Suffix . : Ethernet adapter VMware Network Adapter VMnet1: Connection-specific DNS Suffix . : Link-local IPv6 Address . . . . . : fe80::9867:40fa:5e37:4627%13 IPv4 Address. . . . . . . . . . . : 192.168.192.1 Subnet Mask . . . . . . . . . . . : 255.255.255.0 Default Gateway . . . . . . . . . : Ethernet adapter VMware Network Adapter VMnet8: Connection-specific DNS Suffix . : Link-local IPv6 Address . . . . . : fe80::2fb:efb7:c9aa:a08e%10 IPv4 Address. . . . . . . . . . . : 192.168.101.1 Subnet Mask . . . . . . . . . . . : 255.255.255.0 Default Gateway . . . . . . . . . : Wireless LAN adapter Wi-Fi: Connection-specific DNS Suffix . : IPv6 Address. . . . . . . . . . . : 2402:800:637c:768b:7e35:42af:c8ba:5e3b Temporary IPv6 Address. . . . . . : 2402:800:637c:768b:d445:91c7:9d5f:2f91 Link-local IPv6 Address . . . . . : fe80::1107:2220:2701:2d25%3 IPv4 Address. . . . . . . . . . . : 192.168.1.6 Subnet Mask . . . . . . . . . . . : 255.255.255.0 Default Gateway . . . . . . . . . : fe80::1%3 192.168.1.1 Ethernet adapter vEthernet (Default Switch): Connection-specific DNS Suffix . : Link-local IPv6 Address . . . . . : fe80::566e:90e8:3611:a616%24 IPv4 Address. . . . . . . . . . . : 172.27.96.1 Subnet Mask . . . . . . . . . . . : 255.255.240.0 Default Gateway . . . . . . . . . : C:\Users\btech>

**ChatGPT đã nói:**

Dựa vào cấu hình IP bạn cung cấp, **default gateway (cổng mặc định)** hiện tại của bạn là:

**192.168.1.1**

**📌 Cụ thể:**

Đây là phần quan trọng trong đầu ra của bạn:

Wireless LAN adapter Wi-Fi:

IPv4 Address. . . . . . . . . . . : 192.168.1.6

Subnet Mask . . . . . . . . . . . : 255.255.255.0

Default Gateway . . . . . . . . . : fe80::1%3

192.168.1.1

* Bạn đang kết nối mạng qua **Wi-Fi**
* Địa chỉ IP máy bạn: **192.168.1.6**
* **Default Gateway**:
  + IPv6: fe80::1%3 *(mang tính nội bộ, không quan trọng trong đa số trường hợp thường dùng)*
  + IPv4: **192.168.1.1** → đây là **default gateway chính**, thường là địa chỉ IP của **modem/router Wi-Fi** mà bạn đang kết nối.

**🔍 Vậy default gateway dùng để làm gì?**

* Nó là **"cánh cửa ra internet"** cho thiết bị của bạn.
* Máy tính gửi các gói dữ liệu ra ngoài mạng nội bộ sẽ **đi qua default gateway**.
* Thường là IP của router, nơi tiếp nhận và định tuyến dữ liệu ra ngoài internet.

Nếu bạn muốn:

* **Truy cập trang cấu hình router**, bạn có thể mở trình duyệt và gõ http://192.168.1.1
* **Kiểm tra hoặc đổi DNS/router**, đây cũng là địa chỉ bạn cần.

Cần mình hướng dẫn gì tiếp không? Ví dụ như kiểm tra router, đổi DNS, sửa lỗi mạng...?

**Bạn đã nói:**

cách kiểm tra địa chỉ IP của router nhà bạn (cục Wi-Fi/modem)

**ChatGPT đã nói:**

Để kiểm tra **địa chỉ IP của router (modem/cục Wi-Fi)** mà bạn đang kết nối, bạn chỉ cần xem **Default Gateway** của kết nối mạng đang sử dụng. Đây chính là IP nội bộ của router.

**✅ Cách kiểm tra IP router trên Windows**

**📌 Cách 1: Dùng lệnh ipconfig**

1. Mở **Command Prompt (CMD)**
   * Bấm Windows + R → gõ cmd → Enter
2. Gõ lệnh:
3. ipconfig
4. Tìm phần đang có kết nối (thường là **Wi-Fi** hoặc **Ethernet**).  
   Dòng:
5. Default Gateway . . . . . . . . . : 192.168.x.x

→ chính là **địa chỉ IP của router/modem**.

🔍 **Ví dụ bạn đưa trước đó:**

Wireless LAN adapter Wi-Fi:

IPv4 Address. . . . . . . . . . . : 192.168.1.6

Default Gateway . . . . . . . . . : 192.168.1.1

📌 Kết luận: **IP của router là 192.168.1.1**

**📌 Cách 2: Dùng Control Panel**

1. Vào **Control Panel** → **Network and Sharing Center**
2. Click vào tên Wi-Fi đang kết nối (ở dòng “Connections”)
3. Chọn **Details**
4. Xem dòng **IPv4 Default Gateway** → đó là IP router.

**✅ Cách kiểm tra trên điện thoại (Android / iPhone)**

**Android:**

* Vào **Cài đặt (Settings)** → **Wi-Fi**
* Nhấn vào mạng Wi-Fi đang kết nối
* Tìm dòng **Gateway** hoặc **Router** → chính là IP của router

**iPhone:**

* Vào **Cài đặt** → **Wi-Fi**
* Nhấn vào dấu (i) bên cạnh mạng Wi-Fi
* Tìm mục **Router** → đó là IP của router

**🔐 Mẹo truy cập router**

Sau khi có IP router (ví dụ: 192.168.1.1):

* Mở trình duyệt web
* Gõ vào thanh địa chỉ: http://192.168.1.1
* Nhập **tên đăng nhập & mật khẩu** (mặc định thường là admin/admin hoặc admin/1234 — tùy nhà mạng)

👉 Sau đó bạn có thể:

* Đổi mật khẩu Wi-Fi
* Xem ai đang dùng Wi-Fi
* Đổi DNS
* Cấu hình port forwarding, giới hạn băng thông,...

Nếu bạn không đăng nhập được, mình có thể giúp bạn **tìm tài khoản mặc định** theo model router hoặc nhà mạng.

**Bạn đã nói:**

IP assignment: Automatic (DHCP) DNS server assignment: Automatic (DHCP) SSID: Bao Ngoc Protocol: Wi-Fi 4 (802.11n) Security type: WPA2-Personal Manufacturer: Intel Corporation Description: Intel(R) Dual Band Wireless-AC 8265 Driver version: 22.200.2.1 Network band: 2.4 GHz Network channel: 6 Link speed (Receive/Transmit): 52/52 (Mbps) IPv6 address: 2402:800:637c:768b:7e35:42af:c8ba:5e3b Link-local IPv6 address: fe80::1107:2220:2701:2d25%3 IPv6 DNS servers: fe80::c251:5cff:fe84:f6c0%3 (Unencrypted) IPv4 address: 192.168.1.6 IPv4 DNS servers: 116.97.90.124 (Unencrypted) 203.113.188.8 (Unencrypted) Physical address (MAC): 38-DE-AD-20-6C-6E - là cái nào?

**ChatGPT đã nói:**

Dựa vào thông tin bạn cung cấp, mình sẽ giúp bạn xác định rõ **IP của router**, **DNS**, và một số chi tiết kỹ thuật mạng mà bạn đang dùng:

**✅ 1. IP của router (cục Wi-Fi/modem) là cái nào?**

IP của router **không được hiển thị trực tiếp ở đây**, nhưng **chắc chắn là 192.168.1.1**, vì:

* IP máy bạn là: 192.168.1.6
* Bạn đang dùng **DHCP** (IP tự động cấp phát)
* Dải địa chỉ 192.168.1.x rất phổ biến, và gateway thường là .1
* Trước đó, bạn đã cung cấp thông tin từ ipconfig cho thấy:
* Default Gateway . . . . . . . . . : 192.168.1.1

📌 **Kết luận**: Router của bạn có IP là **192.168.1.1**

**✅ 2. DNS Server đang dùng là gì?**

Bạn đang **không dùng DNS của router**, mà dùng **DNS từ nhà mạng**, cấp phát qua DHCP:

IPv4 DNS servers:

➡️ 116.97.90.124 (Unencrypted)

➡️ 203.113.188.8 (Unencrypted)

* Đây là DNS của nhà mạng Việt Nam, cụ thể:
  + 116.97.90.124 thuộc **VNPT**
  + 203.113.188.8 cũng thuộc **VNPT**

📌 **Kết luận**: Bạn đang dùng **DNS của nhà mạng VNPT**, được cấp tự động qua DHCP.

**✅ 3. Tóm tắt cấu hình mạng hiện tại của bạn**

| **Mục** | **Thông tin** |
| --- | --- |
| **Tên Wi-Fi (SSID)** | Bao Ngoc |
| **IP máy bạn (IPv4)** | 192.168.1.6 |
| **Gateway (Router IP)** | 192.168.1.1 *(xác định từ cấu hình trước)* |
| **Cấp IP & DNS** | Tự động (DHCP) |
| **DNS đang dùng** | 116.97.90.124, 203.113.188.8 |
| **Băng tần Wi-Fi** | 2.4 GHz |
| **Tốc độ kết nối** | 52/52 Mbps |
| **Chuẩn Wi-Fi** | Wi-Fi 4 (802.11n) |
| **Bảo mật Wi-Fi** | WPA2-Personal |

Nếu bạn muốn:

* **Truy cập vào router:** mở trình duyệt và gõ http://192.168.1.1
* **Đổi DNS sang Google, Cloudflare,...** → mình có thể hướng dẫn cụ thể
* **Kiểm tra ai đang dùng Wi-Fi**, cũng cần truy cập vào router
* **Tăng tốc độ mạng** hoặc đổi sang băng tần 5GHz (nếu router hỗ trợ)

👉 Chỉ cần nói mình biết, mình hướng dẫn chi tiết từng bước.

**Bạn đã nói:**

hẹ thống phân giải tên miền là gì

**ChatGPT đã nói:**

Hệ thống phân giải tên miền (DNS – **Domain Name System**) là **"danh bạ điện thoại của Internet"**.

**✅ Nói đơn giản:**

* Khi bạn gõ **một địa chỉ web** như www.google.com vào trình duyệt,
* **Trình duyệt không hiểu tên miền**, nó cần **địa chỉ IP** (ví dụ: 142.250.191.68) để kết nối.
* Lúc đó, **hệ thống phân giải tên miền (DNS)** sẽ **chuyển đổi tên miền thành địa chỉ IP tương ứng**.

**🧠 Ví dụ thực tế:**

| **Bạn nhập** | **DNS chuyển thành** | **Máy tính kết nối tới** |
| --- | --- | --- |
| google.com | 142.250.191.68 | máy chủ Google |
| facebook.com | 157.240.20.35 | máy chủ Facebook |

Bạn không cần nhớ IP, DNS làm điều đó giúp bạn.

**📌 Quá trình phân giải tên miền (DNS Resolution) gồm các bước:**

1. **Trình duyệt** hỏi hệ điều hành: "IP của google.com là gì?"
2. Hệ điều hành hỏi **DNS resolver** (ví dụ: 8.8.8.8 hoặc 127.0.0.53 nếu dùng systemd-resolved)
3. DNS resolver tra cứu thông tin:
   * Nếu **có sẵn trong cache**, trả lời ngay
   * Nếu không, nó sẽ truy vấn các máy chủ DNS khác (qua nhiều cấp)
4. Nhận được IP → trả về cho trình duyệt
5. Trình duyệt dùng IP đó để kết nối tới website

**🎯 Tóm lại:**

| **Câu hỏi** | **Trả lời ngắn gọn** |
| --- | --- |
| **Hệ thống phân giải tên miền là gì?** | Là hệ thống chuyển đổi tên miền thành địa chỉ IP |
| **Tại sao cần?** | Vì máy tính chỉ hiểu địa chỉ IP, không hiểu tên |
| **Ví dụ?** | google.com → 142.250.191.68 |

**📌 DNS có thể tùy chỉnh được không?**

Có. Bạn có thể chọn các máy chủ DNS tùy thích:

| **Tên DNS** | **Địa chỉ IP** | **Ưu điểm** |
| --- | --- | --- |
| Google DNS | 8.8.8.8, 8.8.4.4 | Nhanh, ổn định |
| Cloudflare DNS | 1.1.1.1, 1.0.0.1 | Riêng tư, nhanh |
| Quad9 DNS | 9.9.9.9 | Bảo mật, lọc mã độc |
| AdGuard DNS | 94.140.14.14 | Chặn quảng cáo |

👉 Nếu bạn muốn, mình có thể hướng dẫn cách đổi DNS trong Windows, Android hoặc router.

Bạn muốn tìm hiểu thêm phần nào trong DNS không? (ví dụ: cache, phân giải ngược, cách tăng tốc mạng qua DNS...)

**Bạn đã nói:**

câu lệnh ping 192.168.1.1 để kiểm tra router modem gì gì đó, tôi không phân biệt được

**ChatGPT đã nói:**

Không sao, mình sẽ giải thích **một cách cực dễ hiểu** để bạn **phân biệt và hiểu rõ** việc dùng câu lệnh ping 192.168.1.1 kiểm tra **router/modem** là để làm gì, và nó có nghĩa gì.

**✅ 1. Câu lệnh ping 192.168.1.1 dùng để làm gì?**

Nó dùng để **kiểm tra xem máy tính của bạn có kết nối được với router/modem (cục Wi-Fi)** hay không.

**👉 Lệnh:**

ping 192.168.1.1

**Kết quả:**

* Nếu bạn **nhận được phản hồi (Reply from...)** → ✅ máy tính **đang kết nối thành công với router**
* Nếu bạn **bị Request Timed Out hoặc Destination Host Unreachable** → ❌ có vấn đề về kết nối nội bộ với router

**✅ 2. 192.168.1.1 là gì?**

* Đây là **địa chỉ IP riêng (private IP)** phổ biến nhất mà các router/modem dùng làm **gateway nội bộ**.
* Có thể gọi là "cửa ra vào" từ máy tính tới mạng Internet.
* **Máy tính, điện thoại, tivi, v.v.** trong nhà sẽ được router phát Wi-Fi và cấp IP như 192.168.1.2, 192.168.1.6, v.v.

**✅ 3. Vậy router với modem khác nhau thế nào?**

| **Thiết bị** | **Vai trò** | **Có thể là 192.168.1.1?** | **Thường gọi là gì?** |
| --- | --- | --- | --- |
| **Modem** | Kết nối nhà mạng (ADSL, cáp quang...) | Có thể, nếu phát luôn Wi-Fi | Cục chuyển mạng |
| **Router** | Phát Wi-Fi, chia mạng cho thiết bị khác | Có thể, nếu bạn dùng Wi-Fi từ nó | Cục Wi-Fi |
| **Modem-Router (2-in-1)** | Kết hợp cả hai chức năng | ✅ Rất phổ biến ở VN | Cục Wi-Fi của nhà mạng |

📌 Ở nhà, hầu hết thiết bị của VNPT, Viettel, FPT,... đều là **modem-router**, tức là một thiết bị duy nhất, và **nó có IP là 192.168.1.1**.

**✅ 4. Ping để làm gì trong thực tế?**

| **Mục đích** | **Kết quả mong đợi** | **Giải thích ngắn gọn** |
| --- | --- | --- |
| Kiểm tra kết nối với router | Có phản hồi | ✅ Kết nối nội bộ vẫn ổn |
| Không có phản hồi (Timeout) | ❌ | Có thể bị ngắt mạng, lỗi Wi-Fi, sai cấu hình |
| Trễ cao (ping >100ms) | ⚠️ | Mạng nội bộ chậm, nghẽn, nhiễu sóng |

**✅ Ví dụ chạy ping trong CMD:**

C:\Users\You> ping 192.168.1.1

Reply from 192.168.1.1: bytes=32 time=2ms TTL=64

Reply from 192.168.1.1: bytes=32 time=1ms TTL=64

📌 Bạn thấy Reply from... → nghĩa là **máy tính bạn đang nói chuyện được với router**, mạng nội bộ hoạt động OK.

**❓Bạn nên dùng lệnh này khi nào?**

* Khi **mạng chậm hoặc mất mạng** → kiểm tra xem có còn kết nối được với router không.
* Khi **không vào được mạng** → xác định xem lỗi nằm ở **nội bộ (Wi-Fi/router)** hay **ngoài Internet**.

Nếu bạn chạy thử lệnh ping 192.168.1.1 và gửi kết quả, mình có thể **giúp bạn phân tích cụ thể luôn** nhé.

**Bạn đã nói:**

như này nghĩa là sao? Microsoft Windows [Version 10.0.22631.5984] (c) Microsoft Corporation. All rights reserved. C:\Users\btech>ping 192.168.1.1 Pinging 192.168.1.1 with 32 bytes of data: Reply from 192.168.1.1: bytes=32 time=3ms TTL=64 Reply from 192.168.1.1: bytes=32 time=2ms TTL=64 Reply from 192.168.1.1: bytes=32 time=2ms TTL=64 Reply from 192.168.1.1: bytes=32 time=2ms TTL=64 Ping statistics for 192.168.1.1: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 2ms, Maximum = 3ms, Average = 2ms C:\Users\btech>

**ChatGPT đã nói:**

Kết quả bạn vừa chạy cho lệnh:

ping 192.168.1.1

👉 **nghĩa là**: **Kết nối giữa máy tính của bạn và router (modem/cục Wi-Fi) đang hoạt động bình thường và ổn định.**

**🔍 Giải thích chi tiết:**

**✅ Dòng này:**

Reply from 192.168.1.1: bytes=32 time=2ms TTL=64

Nghĩa là:

* Máy tính của bạn **gửi gói tin (ping)** tới địa chỉ router (192.168.1.1)
* Router đã **trả lời (Reply)** → kết nối thành công
* time=2ms → mất 2 mili-giây để gửi & nhận lại → **rất nhanh, ổn định**
* TTL=64 → không cần quan tâm nhiều, chỉ là thông số kỹ thuật của gói tin

**📊 Thống kê cuối cùng:**

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

✔️ Gửi 4 gói → Nhận lại 4 gói → **Không mất gói nào** → Kết nối **không bị rớt**  
Tức là giữa **máy tính ↔ router**: ổn định, không lỗi.

**🎯 Kết luận:**

| **Ý nghĩa** | **Tình trạng** |
| --- | --- |
| Kết nối tới router | ✅ Tốt |
| Mạng nội bộ Wi-Fi | ✅ Ổn định |
| Tốc độ phản hồi ping | ⚡ Nhanh (2–3 ms) |
| Mất gói? | ❌ Không |

**✅ Vậy bước tiếp theo?**

* Nếu bạn **vào web vẫn chậm hoặc không vào được**, thì **vấn đề nằm ngoài router**, cụ thể:
  + **DNS** bị lỗi hoặc chậm
  + **Mạng nhà mạng (VNPT, Viettel...) đang có vấn đề**
  + **Tốc độ Internet bị nghẽn**, nhiều thiết bị dùng cùng lúc
* Có thể thử:
  + Ping ra ngoài Internet: ping 8.8.8.8
  + Ping tên miền: ping google.com
  + Đổi DNS để kiểm tra thử tốc độ cải thiện không

👉 Nếu bạn muốn, bạn gửi tiếp kết quả lệnh ping 8.8.8.8 hoặc ping google.com, mình sẽ giúp bạn **xác định lỗi nằm ở đâu**.

**Bạn đã nói:**

mình vẫn chưa phân biệt được model với router là gì? và phát wifi nghĩa là sao? cái nào

**ChatGPT đã nói:**

Rất nhiều người nhầm lẫn giữa **modem**, **router** và việc **phát Wi-Fi**, nên bạn hỏi là rất hay và đáng để làm rõ. Mình sẽ giải thích cực dễ hiểu để bạn **phân biệt một lần là nhớ luôn**:

**✅ 1. Modem là gì?**

**👉 Modem = thiết bị kết nối nhà bạn với mạng Internet của nhà mạng.**

* Nó nhận tín hiệu từ **cáp quang, dây điện thoại, cáp đồng trục**,...
* Sau đó **chuyển đổi tín hiệu** để đưa Internet vào trong nhà.
* Không chia mạng, không phát Wi-Fi (nếu là modem đơn thuần).

📌 Tên đầy đủ là: **MOdulator - DEModulator**

**✅ 2. Router là gì?**

**👉 Router = thiết bị chia Internet cho nhiều thiết bị khác dùng chung (máy tính, điện thoại, TV,...)**

* Nhận Internet từ modem, rồi **phân phối lại** qua Wi-Fi hoặc dây LAN.
* Có thể thiết lập **Wi-Fi, tên mạng, mật khẩu, tường lửa**,...
* Thường có nhiều cổng LAN, và **phát Wi-Fi cho các thiết bị kết nối**.

📌 Router không thể tự truy cập Internet nếu **không có modem**.

**✅ 3. Phát Wi-Fi là gì?**

**👉 Là khả năng tạo mạng không dây (Wi-Fi) để các thiết bị như điện thoại, laptop,... kết nối mà không cần dây.**

* **Chỉ router có Wi-Fi mới phát Wi-Fi được**
* Có thiết bị không phát Wi-Fi (router dùng dây LAN), chỉ có **Wi-Fi router** mới phát.

**✅ 4. Modem và Router có thể là 2 thiết bị khác nhau hoặc chung 1 cục**

**❗ Có 2 trường hợp bạn sẽ gặp:**

| **Trường hợp** | **Thiết bị bạn thấy trong nhà** | **Gọi là gì?** |
| --- | --- | --- |
| 1. **Tách rời** | 1 modem + 1 router | Cài đặt chuyên sâu, phổ biến với mạng mạnh |
| 2. **2 trong 1** (phổ biến ở VN) | 1 cục duy nhất của nhà mạng | Modem kiêm router – thường gọi là **cục Wi-Fi**, **cục mạng** |

**🔍 Tóm gọn:**

| **Tên** | **Chức năng** | **Có phát Wi-Fi không?** |
| --- | --- | --- |
| **Modem** | Kết nối nhà bạn với nhà mạng (Internet) | ❌ Không |
| **Router** | Chia mạng cho các thiết bị, quản lý mạng nội bộ | ✅ Nếu là router có Wi-Fi |
| **Modem-Router** | Kết hợp cả hai chức năng | ✅ Có (đa số ở VN là loại này) |

**📌 Ví dụ thực tế trong nhà bạn:**

* Bạn có một **cục của VNPT/Viettel/FPT** → nó là **modem + router Wi-Fi** trong 1 thiết bị.
* Bạn kết nối Wi-Fi tên là Bao Ngoc → tức là **router đang phát Wi-Fi**.
* IP router bạn ping là 192.168.1.1 → bạn đang nói chuyện với thiết bị đó.

**👉 Câu hỏi phụ:**

* Khi người ta nói "phát Wi-Fi", tức là **router đang tạo ra mạng không dây**, bạn có thể kết nối điện thoại/laptop vào đó mà không cần cắm dây.
* Không phải modem nào cũng phát Wi-Fi.
* Bạn có thể "tắt Wi-Fi" của router để dùng mạng dây thôi.

Nếu bạn muốn, mình có thể gửi hình ảnh minh họa 2 loại thiết bị này để bạn dễ nhận ra trong thực tế. Bạn thấy giải thích như vậy đã dễ hiểu hơn chưa?

**Bạn đã nói:**

sao tôi dùng trong ubuntu server nó lại cho kết quả khác máy thật lúc nãy nhỉ? 23:25:19 lebao@us2:~$ ping 192.168.1.1 ping: connect: Network is unreachable 23:25:31 lebao@us2:~$ -

**ChatGPT đã nói:**

Bạn đang gặp lỗi:

ping: connect: Network is unreachable

Khi chạy trên **Ubuntu Server**, nghĩa là:

⚠️ **Máy chủ Ubuntu của bạn hiện không có kết nối mạng tới router (192.168.1.1), hoặc chưa được cấu hình mạng đúng cách.**

**✅ So sánh với máy thật lúc nãy:**

* Trên **Windows**, bạn ping 192.168.1.1 → **kết nối thành công** → Windows đã kết nối Wi-Fi, có IP, có gateway.
* Trên **Ubuntu Server**, bạn ping 192.168.1.1 → **"Network is unreachable"** → Ubuntu **chưa có mạng**, không có IP nội bộ, chưa kết nối router.

**❓Tại sao lại bị lỗi này?**

Dưới đây là các **lý do thường gặp nhất** khi Ubuntu báo lỗi "Network is unreachable":

**🔸 1. Chưa cắm dây mạng (với máy ảo hoặc server vật lý)**

* Nếu bạn dùng **Ubuntu Server trên máy thật**, kiểm tra xem **có cắm dây LAN vào không**.
* Nếu là **Ubuntu chạy trong máy ảo (VD: VirtualBox, VMware)** → cần **cấu hình network đúng kiểu** (Bridged/NAT).

**🔸 2. Không được gán IP (chưa cấu hình DHCP tĩnh hoặc động)**

* Ubuntu Server không tự kết nối mạng như Desktop
* Bạn phải kiểm tra và cấu hình mạng trong file:  
  **/etc/netplan/\*.yaml**

**🔸 3. Dịch vụ mạng chưa khởi động hoặc bị lỗi**

* Có thể networking.service hoặc NetworkManager chưa chạy

**✅ Cách kiểm tra & khắc phục**

**🧪 Bước 1: Kiểm tra card mạng có hoạt động không**

ip a

👉 Bạn sẽ thấy các interface như: eth0, ens33, enp0s3, v.v.

* Nếu không thấy **IP nào trong dải 192.168.x.x**, hoặc chỉ thấy 127.0.0.1, thì chưa có kết nối mạng.

**🧪 Bước 2: Kiểm tra cài đặt netplan**

Chạy:

ls /etc/netplan/

Bạn sẽ thấy 1 file .yaml, ví dụ: 01-netcfg.yaml

Mở để xem nội dung:

sudo nano /etc/netplan/01-netcfg.yaml

Ví dụ cấu hình DHCP:

network:

version: 2

ethernets:

ens33:

dhcp4: true

Sau đó **áp dụng cấu hình**:

sudo netplan apply

**🧪 Bước 3: Ping lại**

ping 192.168.1.1

Nếu vẫn lỗi → kiểm tra lại kiểu mạng nếu là máy ảo (Bridged mode hay NAT), hoặc cắm lại dây nếu là máy thật.

**📌 Tóm lại:**

| **Nguyên nhân có thể** | **Khắc phục gợi ý** |
| --- | --- |
| Ubuntu chưa kết nối mạng | Kiểm tra ip a |
| Chưa cấu hình netplan | Sửa /etc/netplan/\*.yaml |
| Dùng máy ảo cấu hình sai | Chuyển sang **Bridged Adapter** |
| Không có DHCP hoặc router không cấp IP | Cấu hình IP tĩnh |

Nếu bạn gửi giúp mình đầu ra của lệnh ip a và nội dung file /etc/netplan/\*.yaml, mình sẽ hướng dẫn cấu hình đúng ngay lập tức.

Bạn đang dùng Ubuntu Server **trên máy thật**, hay **trong máy ảo (VMware, VirtualBox)** vậy?