

5

Lab

**PHỤC VỤ MỤC ĐÍCH GIÁO DỤC**  
FOR EDUCATIONAL PURPOSE ONLY

# Cấu hình web server với Nginx

**Thực hành môn Lập trình ứng dụng Web**

Tháng 5/2025

**Lưu hành nội bộ**

<Nghiêm cấm đăng tải trên internet dưới mọi hình thức>

## A. TỔNG QUAN

### 1. Mục tiêu

- Hiểu được vai trò của Web Server trong ứng dụng web.
- Cài đặt và cấu hình cơ bản Web Server Nginx.
- Triển khai ứng dụng web đơn giản thông qua Nginx.

### 2. Thời gian thực hành

- Thực hành tại lớp: 5 tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa 13 ngày.

### 3. Liên quan

- Sinh viên cần nắm các kiến thức nền tảng về HTML, CSS, JavaScript và PHP. Các kiến thức này đã được giới thiệu trong nội dung lý thuyết đã học do đó sẽ không được trình bày lại trong nội dung thực hành này.

## B. CHUẨN BỊ MÔI TRƯỜNG

- Cài đặt máy ảo Ubuntu với VMware hoặc VirtualBox.
- Cài đặt Nginx trên môi trường ảo trên.

## C. THỰC HÀNH

### 1. Giới thiệu Nginx

Nginx là một máy chủ Web Server mã nguồn mở, nó được thiết kế hướng đến mục đích cải thiện tối đa hiệu năng và sự ổn định. Bên cạnh đó, nhờ vào các khả năng của máy chủ HTTP mà NGINX còn có thể hoạt động như một proxy server cho email (IMAP, POP3, và SMTP), reverse proxy, và trung gian để cân bằng tải cho các máy chủ HTTP, TCP, và UDP.

Hướng dẫn cài đặt và cấu hình web server với Nginx.

#### Bước 1: Cài đặt Nginx

Command:

```
sudo apt-get update  
sudo apt install nginx
```

#### Bước 2: Điều chỉnh firewall.

Liệt kê các ứng dụng mà **ufw** biết:

```
sudo ufw app list
```

Kết quả sẽ cần gồm có các profiles của Nginx

```
kian@kian:~$ sudo ufw app list
Available applications:
CUPS
Nginx Full
Nginx HTTP
Nginx HTTPS
```

Cho phép lưu lượng truy cập trên port 80 (http):

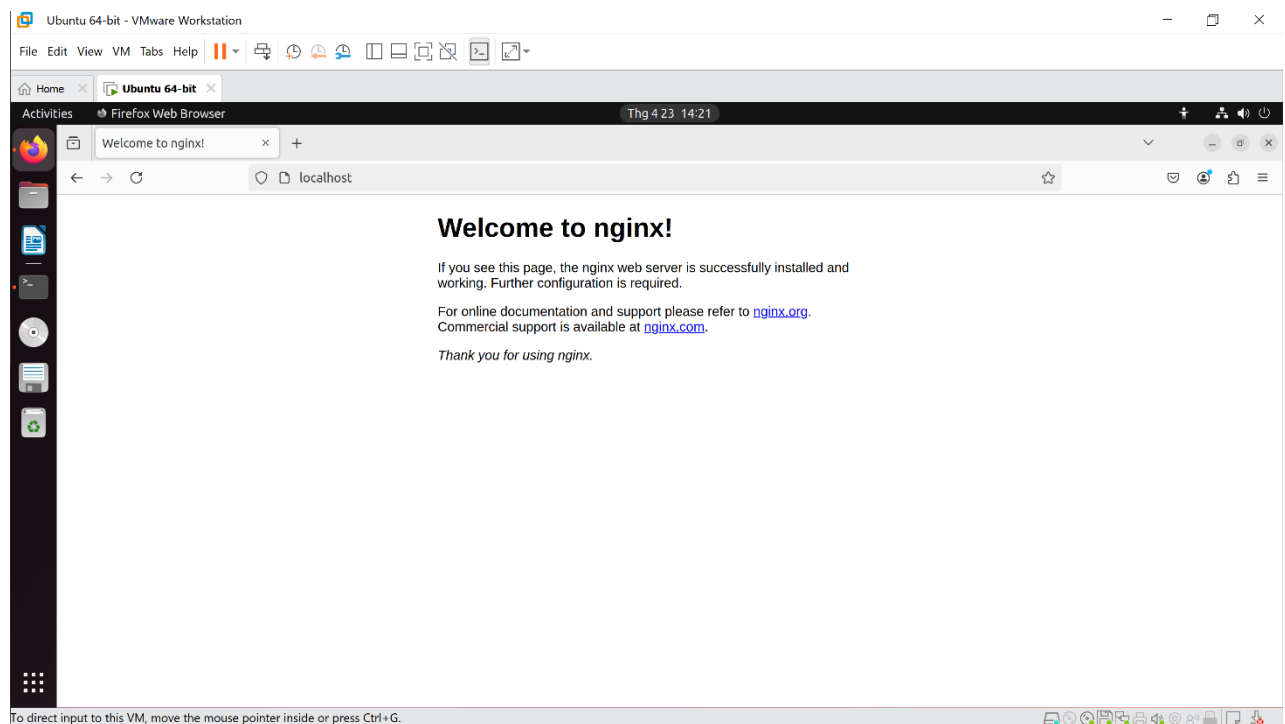
```
sudo ufw enable
sudo ufw allow 'Nginx HTTP'
```

Kiểm tra kết quả cấu hình firewall: `sudo ufw status`

```
kian@kian:~$ sudo ufw status
Status: active

To Action From
--
Nginx HTTP ALLOW Anywhere
Nginx HTTP (v6) ALLOW Anywhere (v6)
```

Truy cập `http://localhost` thành công để xác nhận Nginx đang chạy.



Bước 3: Cấu hình file và thư mục cho bài lab

**Tạo thư mục chung có tên là lab5\_MSSV**

và tạo các thư mục con cho các task

```
sudo mkdir -p /var/www/lab5
sudo mkdir -p /var/www/lab5/task1
```

**Bước 4: Cấu hình Nginx trở đến thư mục mới**

```
sudo nano /etc/nginx/sites-available/lab5_task1
```

Thêm nội dung sau vào file, lưu ý tùy chỉnh lại tên (lab5\_MSSV):

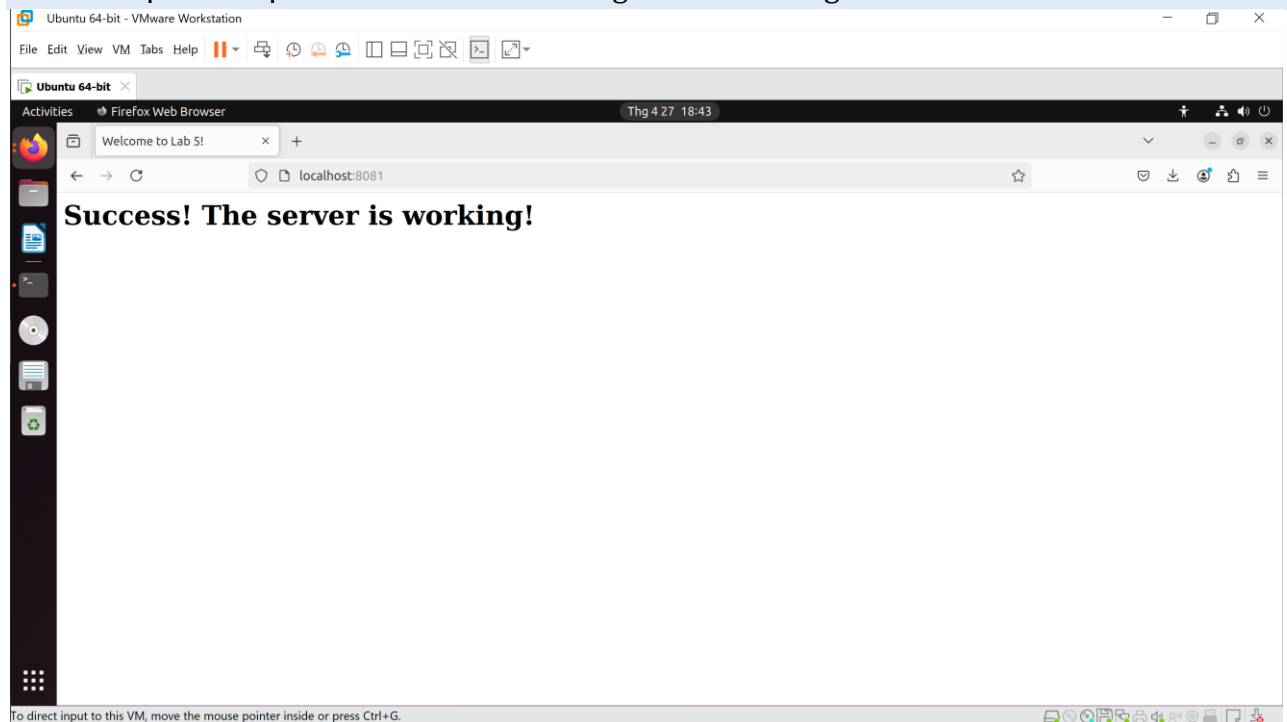
```
server {
    listen 8081;
    server_name localhost;
    root /var/www/lab5/task1;
    index index.html index.php;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Kích hoạt file đã tạo bằng cách tạo liên kết đến thư mục sites-enabled, Nginx sẽ đọc trong quá trình khởi động.

```
sudo ln -s /etc/nginx/sites-available/lab5_task1 /etc/nginx/sites-enabled/
```

Kiểm tra cấu hình các files Nginx và reload lại Nginx:

```
sudo nginx -t
sudo systemctl reload nginx
```

**Task 1: Tạo thư mục con task1 để chứa trang web tĩnh đơn giản.****2. Cấu hình và triển khai web server****a) Triển khai PHP**

**Bước 1: Cài đặt php và service trên môi trường ảo**

```
sudo apt install php
sudo apt install php8.1-fpm
```

**Bước 2: Tạo file và thư mục cho task 2****Lưu ý: Cấp quyền cho thư mục nếu bị lỗi**

```
sudo chown -R $USER:$USER /var/www/lab5
sudo chmod -R 755 /var/www/lab5
```

**Bước 3: Tạo file cấu hình Nginx cho web task2**

```
sudo nano /etc/nginx/sites-available/lab5_task2
```

**Nội dung file:**

```
server {
    listen 8082;
    server_name localhost;

    root /var/www/lab5/task2;
    index index.php index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
    }
}
```

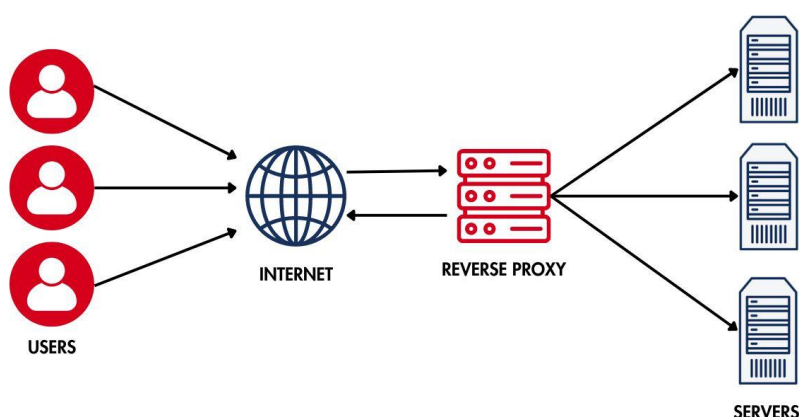
**Bước 4: Kích hoạt file với sites-enabled và reload.**

**Task 2:** Tạo form với method POST nhận dữ liệu từ người dùng gồm các trường: Họ và tên, tuổi, email, số điện thoại. Sau khi người dùng chọn nút gửi, nhảy điều hướng tới trang mới hiển thị lại các thông tin đã nhập.

**b) Thiết lập reverse proxy và load balancing**

**Reverse proxy** là một máy chủ trung gian đứng trước các web server backend, tiếp nhận toàn bộ yêu cầu từ client và chuyển tiếp (proxy) đến backend phù hợp. Nó có thể ẩn toàn bộ hệ thống backend phía sau 1 domain duy nhất, làm cổng vào tập trung (entrypoint). Qua đó giúp tăng bảo mật và dễ kiểm soát backend. Ngoài ra Reverse Proxy có thể được dùng như một tường lửa đơn giản để block hoặc filter các request không mong muốn.

**Load balancing** là kỹ thuật phân phối yêu cầu từ client đến nhiều máy chủ backend khác nhau để tối ưu hiệu năng và đảm bảo hệ thống không bị quá tải. Qua đó, duy trì tính sẵn sàng của hệ thống (high availability) và cân bằng tải nguyên xử lý giữa các backend (round-robin, least-conn, IP-hash,...).



**Task 3:** Cấu hình Nginx để làm reverse proxy phân phối yêu cầu đến nhiều backend web server và quan sát hoạt động thực tế của load balancing.

**Bước 1:** Chuẩn bị 2 backend PHP servers

Sử dụng lại nội dung Task 2 (hoặc tạo nội dung tùy chọn):

```
sudo cp -r /var/www/lab5/task2 /var/www/lab5/task2_2
sudo cp -r /var/www/lab5/task2 /var/www/lab5/task2_3
```

Trong đó,

- Port 8082: task2 (gốc)
- Port 8083: task2\_2
- Port 8084: task2\_3

Đối với mỗi backend chỉnh file index.php tương ứng để hiển thị thông tin port, ví dụ với backend task2\_2 chạy trên port 8083:

```
<?php
echo "Backend hiện tại: PORT 8083";
?>
```

**Bước 2:** Cấu hình 2 backend HTTP server trong Nginx

- Tạo các file cấu hình Nginx riêng cho từng backend:

/etc/nginx/sites-available/task2\_8083

```
server {
    listen 8083;
    server_name localhost;

    root /var/www/lab5/task2_2;
    index index.php index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
```

```

fastcgi_pass unix:/run/php/php8.1-fpm.sock;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
include fastcgi_params;
}
}

```

- Tương tự tạo task2\_8084, lưu ý đổi thư mục root và port của listen.
- Kích hoạt file với sites-enabled và reload.

### Bước 3: Cấu hình reverse proxy load balancer

- Tạo file cấu hình /etc/nginx/sites-available/lab5\_task3

```

upstream backend_servers {
    server 127.0.0.1:8083;
    server 127.0.0.1:8084;
}

server {
    listen 80;

    server_name task3.com;

    location / {
        proxy_pass http://backend_servers;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

- Kích hoạt file với sites-enabled và reload.

### Bước 4: Cấu hình file /etc/hosts

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

Thực hiện map domain tới localhost:

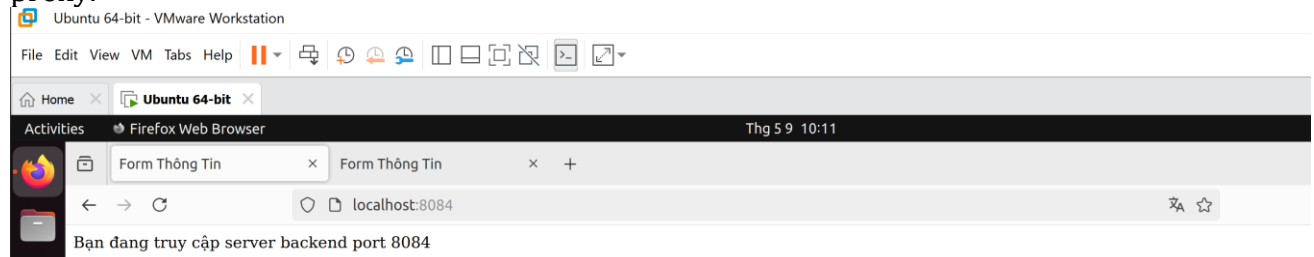
```
127.0.0.1      task3.com
```

### Bước 5: Kiểm tra hoạt động

Truy cập <http://task3.com/> để kiểm tra kiểm quả

Nếu mỗi lần reload lại trang, backend có thay đổi (8083 → 8084 → 8083), tức là load balancing hoạt động thành công.

Tuy nhiên, với cách cấu hình trên ta vẫn có thể truy cập trực tiếp tới backend mà không qua proxy.



Hãy cấu hình để ngăn chặn điều này. Chỉ cho phép truy cập qua proxy, không cho client gọi trực tiếp vào backend.



**Task 4:** Theo mặc định, thuật toán load balancing của Nginx là **round-robin**, tức các yêu cầu (requests) sẽ được phân phối tuần tự theo vòng tròn đến danh sách các máy chủ backend. Do đó, mỗi backend sẽ được nhận số lượng yêu cầu gần như là như nhau, bất kể backend đó mạnh yếu ra sao. Hãy tìm hiểu những thuật toán load balancing khác trong Nginx (**least\_conn, ip\_hash, weight**) và triển khai. Yêu cầu: trình bày cách hoạt động của mỗi thuật toán, ưu nhược điểm và kết quả thực hiện có minh chứng (ảnh chụp, lập thống kê).

Gợi ý: Để kiểm tra hoạt động của thuật toán có thể sử dụng vòng lặp với curl Command để truy cập nhiều lần, ví dụ:

```
for i in {1..20}; do curl http://task3.com; sleep 1; done
```

## D. YÊU CẦU & ĐÁNH GIÁ

- Sinh viên tìm hiểu và thực hành theo hướng dẫn, thực hiện theo nhóm đã đăng ký.
- Nộp báo cáo kết quả gồm chi tiết những việc (Report) mà nhóm đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Báo cáo:
  - File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
  - Đặt tên theo định dạng: [Mã lớp]-LabX\_NhomY.
  - Ví dụ: [NT208.P12.ANTT.1]-Lab1\_Nhom1.
  - Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
  - Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

*Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

*Chúc các bạn hoàn thành tốt!*