BÁO CÁO THỰC HÀNH

**Môn học: AN TOÀN MẠNG**

**Tên chủ đề: Proxy**

*GVHD: Tô Trọng Nghĩa*

**Nhóm: 09**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT140.P11.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Email |
| 1 | Trần Anh Khôi | 22520701 | [22520701@gm.uit.edu.vn](mailto:22520701@gm.uit.edu.vn) |
| 2 | Hồ Vỉ Khánh | 22520633 | [22520633@gm.uit.edu.vn](mailto:22520633@gm.uit.edu.vn) |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-2)**

|  |  |  |
| --- | --- | --- |
| STT | Nội dung | Tình trạng |
| 1 | Yêu cầu 1 | 100% |
| 2 | Yêu cầu 2 | 100% |
| Điểm tự đánh giá 10/10 | | |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

## Cài đặt Apache và NGINX

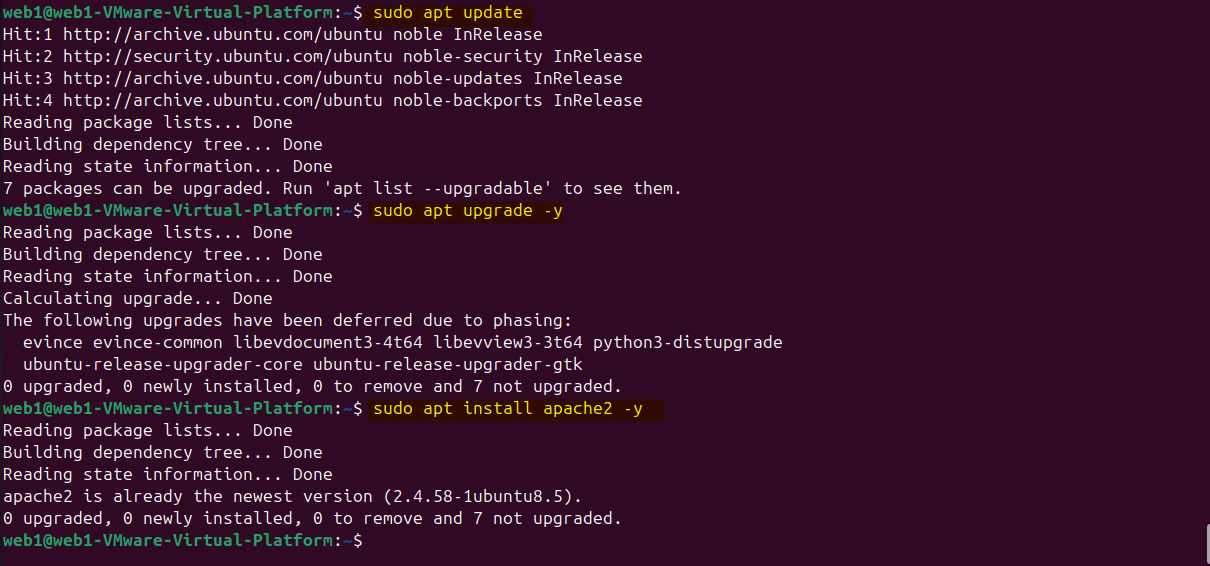
### Cài đặt Apache

Tại máy ảo Web 1, thực hiện cài đặt Apache bằng các câu lệnh sau:

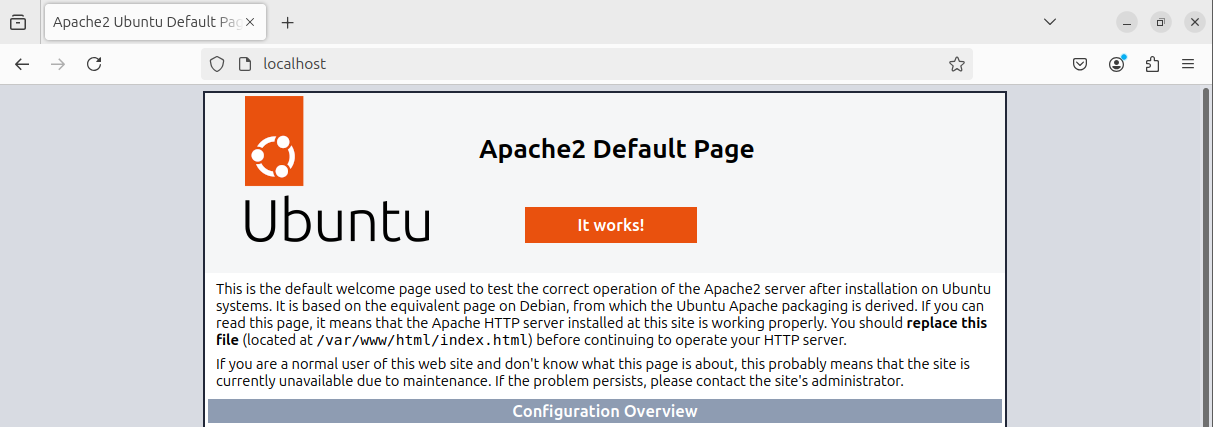
$sudo apt update

$sudo apt upgrade -y

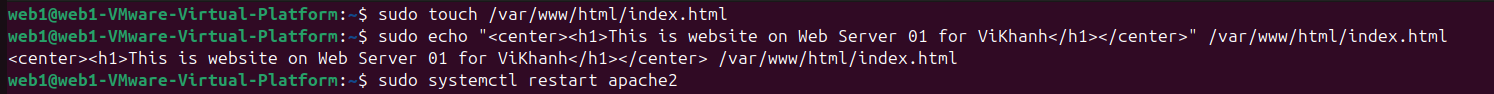
$sudo apt install apache2 -y



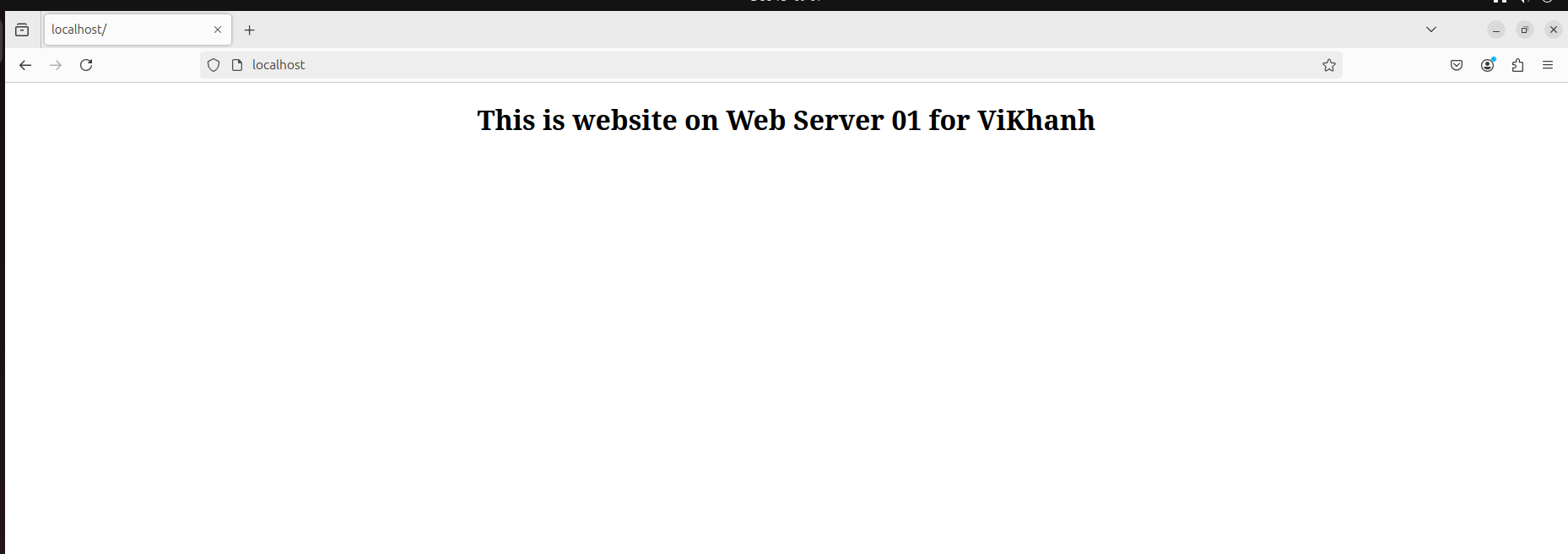
Sau khi cài đặt, truy cập đường dẫn http://localhost để kiếm tra. Nếu website hiển thị như hình dưới chúng ta đã cài đặt **thành công**.



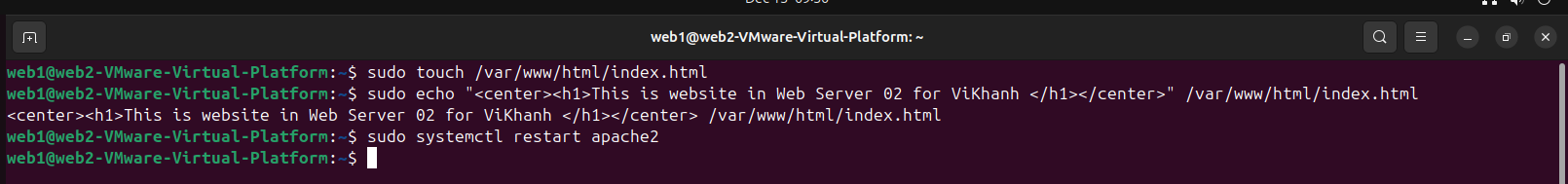
Bây giờ chúng ta sẽ tạo một website đơn giảng trên server Web1.

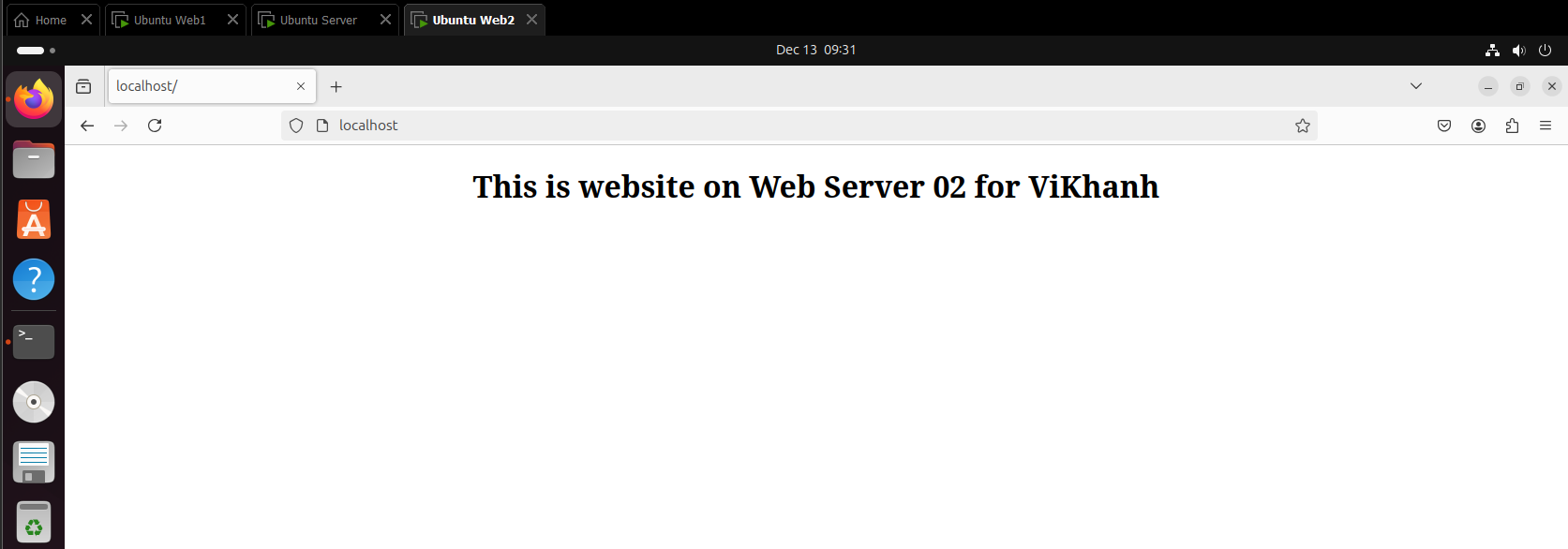


Khởi động lại Apache và truy cập lại đường dẫn <http://localhost> chúng ta sẽ thấy website vừa được cấu hình



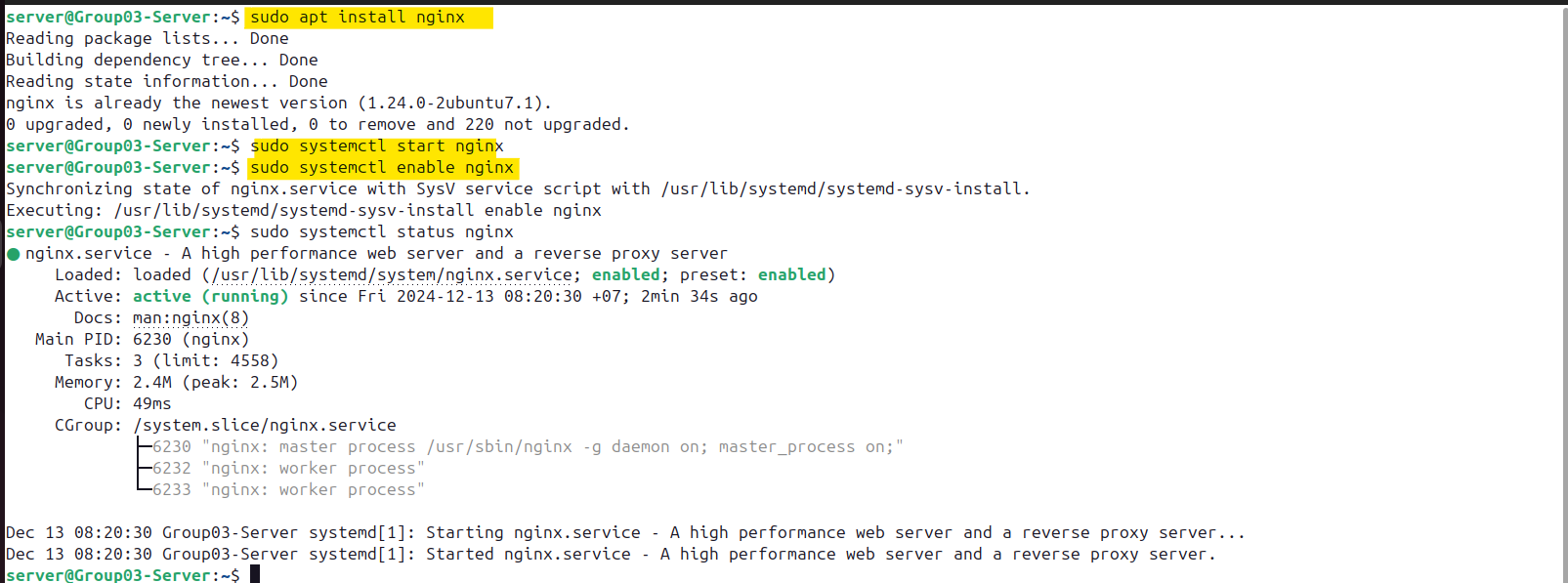
#### **Task 1:** Thực hiện cài đặt Apache và cấu hình web tương tự trên server Web2, thay đổi nội dung của file HTML để website ở server Web2 hiển thị dòng “This is website on Web Server 02”.





### Cài đặt NGINX

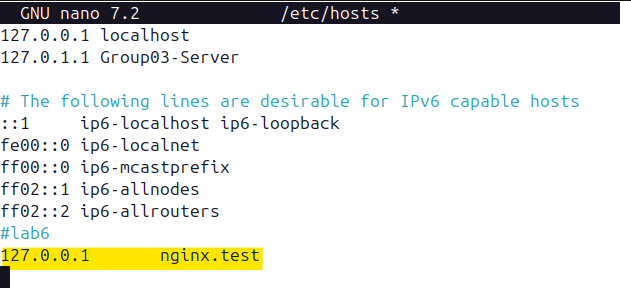
Trên máy ảo NGINX LB, thực hiện cài đặt NGINX

****

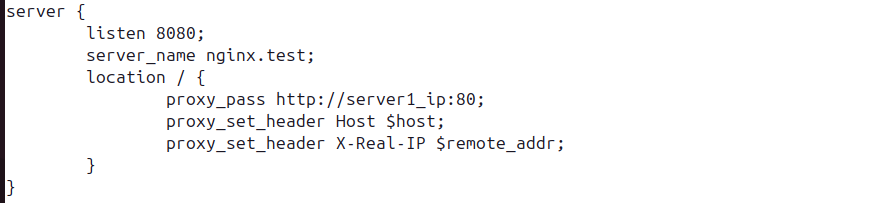
## Cấu hình NGINX

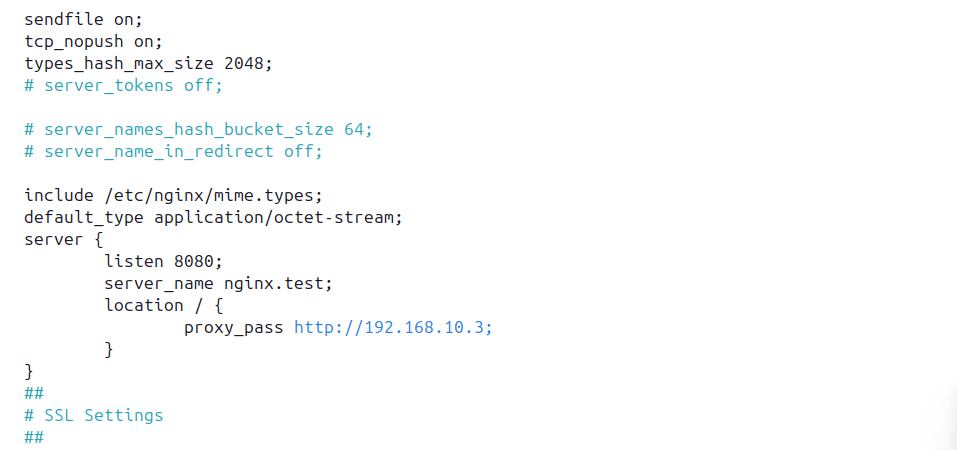
### Thiết lập máy chủ proxy đơn giản

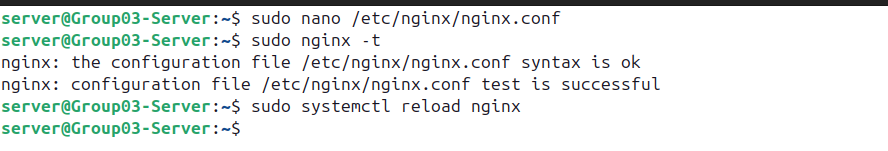
Chỉnh sửa file /etc/hosts, và thêm một ánh xạ tên miền nginx.test tới địa chỉ 127.0.0.1



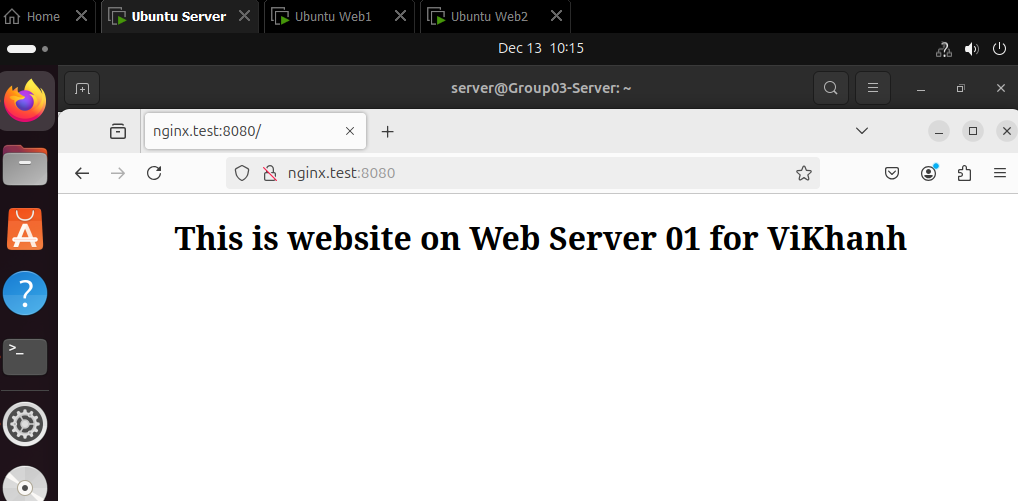
Tại server NGINX, mở file cấu hình /etc/nginx/nginx.conf và chỉnh sửa:





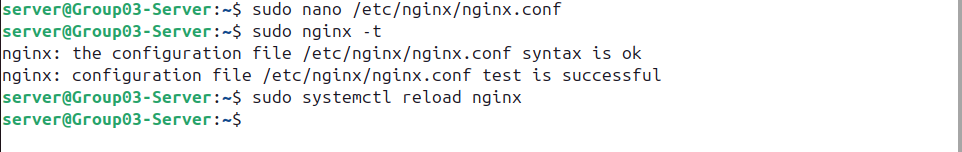


Để kiểm tra cấu hình, dùng trình duyệt web truy cập vào đường dẫn <http://nginx.test:8080> ,chúng ta sẽ nhận được nội dung từ Web server 1.

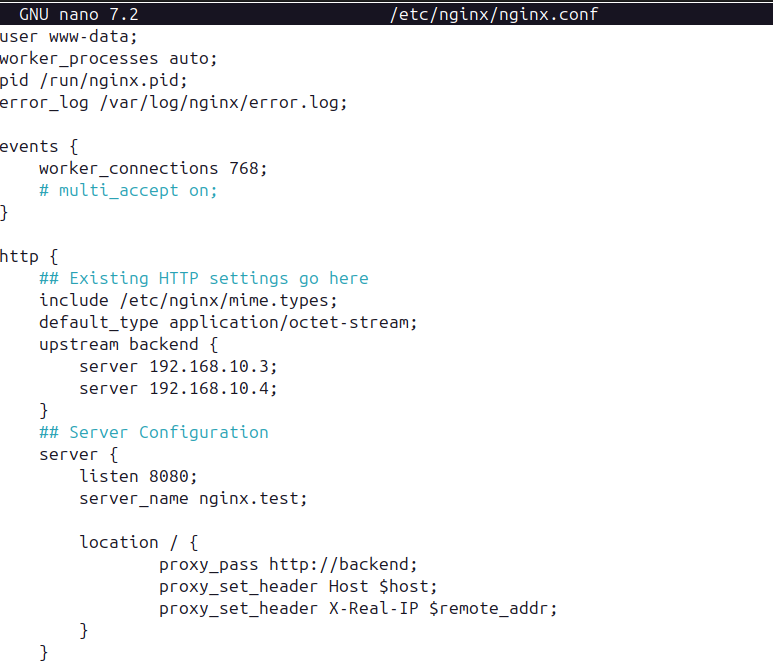


### Load Balacing

Mở file cấu hình trước đó /etc/nginx/nginx.conf

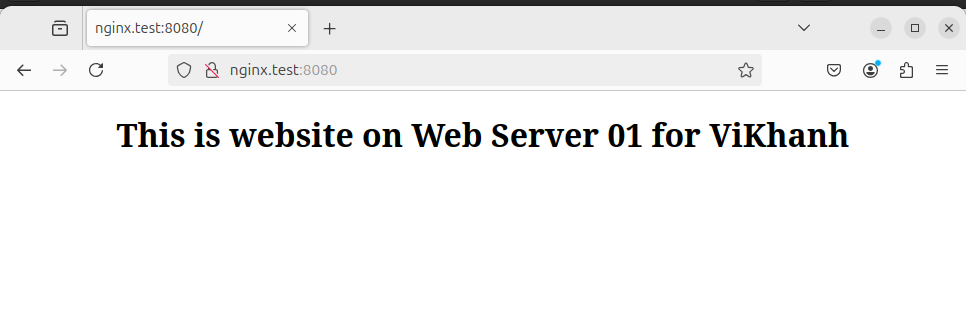


Chỉnh sửa thành



Truy cập lại đường dẫn <http://nginx.test:8080/> và refresh liên tục để thấy sự thay đổi giữa 2 server sau mỗi lần refresh.

**Kết quả**:





**Giải thích**

- Khai báo nhóm các backend server được NGINX sử dụng để chuyển tiếp yêu cầu từ client

- Các server trong danh sách này (192.168.10.130 và 192.168.10.128) sẽ nhận các yêu cầu được chuyển tiếp từ NGINX

- NGINX sử dụng thuật toán định tuyến Round Robin để phân phối yêu cầu giữa các server này.

- Mỗi yêu cầu được gửi luân phiên đến 192.168.10.130, sau đó đến 192.168.10.128, và quay lại từ đầu

- Truy cập nginx thông qua port 8080

- Định nghĩa tên miền nginx.test

- Khối location định nghĩa các yêu cầu đến server

**Cách hoạt động:**

- Client gửi yêu cầu HTTP đến NGINX qua địa chỉ http://<IP-nginx>:8080.

- NGINX nhận yêu cầu và chuyển tiếp đến một trong các backend server (192.168.10.130 hoặc 192.168.10.128) dựa trên thuật toán định tuyến (mặc định là Round Robin).

- Các header như Host và X-Real-IP được truyền từ client đến backend server để đảm bảo backend server nhận được đầy đủ thông tin từ yêu cầu ban đầu.

- Backend server xử lý yêu cầu và trả phản hồi lại cho NGINX.

- NGINX gửi phản hồi về client.

**Nguyên lý:**

- Với cấu hình NGINX hiện tại, nếu một server trong upstream chết (ví dụ: server 192.168.10.130 hoặc 192.168.10.128 không hoạt động), NGINX sẽ tự động chuyển yêu cầu sang server còn lại để đảm bảo dịch vụ vẫn hoạt động. Đây chính là ưu điểm của việc sử dụng load balancing với NGINX

#### **Task 2:** Dựa vào quan sát, hãy giải thích cách hoạt động của thuật toán định tuyến Round Robin. Tìm hiểu tài liệu và cấu hình một thuật toán định tuyến khác. Giải thích cách hoạt động của thuật toán định tuyến này. <https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/>

## Thuật toán Round Robin

**Round Robin**: Mỗi lần refresh sẽ gửi đến 1 server khác trong server-backend. Yêu cầu tuần tự giữa các server, không quan tâm đến trạng thái hoặc dữ liệu cụ thể của từng yêu cầu.

Ưu điểm:

* Giúp cân bằng tải, phù hợp với hệ thống đồng nhất.
* Đơn giản, dễ cấu hình giảm nguy cơ cấu hình lỗi

Nhược điểm:

* Không có khả năng phát hiện và loại bỏ server bị tấn công hoặc hoạt động bất thường
* Chậm khi hệ thống có các server cấu hình khác nhau nhưng lại nhận các request bằng nhau.

**Least Connections**: Chọn server có ít kết nối nhất để xử lí yêu cầu.

Ưu điểm:

* Giảm nguy cơ quá tải trên một server cụ thể, hạn chế khả năng tấn công từ chối dịch vụ (DoS) tập trung vào một server.
* Phù hợp hệ thống lớn, xử lí nhiều loại request

Nhược điểm:

* Không hiệu quả khi có 2 request ngắn với 1 request dài.
* Không kiểm tra loại kết nối, nên các kết nối độc hại vẫn được xem xét như các kết nối hợp lệ

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Round Robin** | **Least Connections** |
| Phương thức định tuyến | Luân phiên từng server | Định tuyến đến server ít kết nối nhất |
| Phù hợp với | Các server có cấu hình giống nhau | Các server cấu hình khác nhau hoặc tải không đồng đều |
| Phân phối yêu cầu | Đồng đều | Tập trung vào tải thực tế |

## Thuật toán Random

Thuật toán **Random** trong định tuyến là một phương pháp phân phối tải đơn giản và ngẫu nhiên, trong đó mỗi yêu cầu HTTP sẽ được gửi đến một backend server được chọn ngẫu nhiên từ danh sách các server có sẵn. Phương pháp này không dựa vào bất kỳ yếu tố nào như số kết nối, thời gian phản hồi hay tài nguyên của server mà chỉ sử dụng yếu tố ngẫu nhiên để quyết định

**Cấu hình**

A computer screen with white text

Description automatically generated

Thêm dòng random;

**Test thử làm 1 kịch bản**

Kết quả truy cập ngẫu nhiên

A close-up of a website

Description automatically generated

Lần 1

A screen shot of a computer

Description automatically generated

Lần 2

A close-up of a website

Description automatically generated  
Lần 3

Nó random ngẫu nhiên

**So sánh với Round Robin**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Round Robin** | **Random** |
| Cách hoạt động | Chuyển yêu cầu đến các server theo vòng tròn. Mỗi server nhận một yêu cầu theo thứ tự. | Chọn ngẫu nhiên một server để xử lý mỗi yêu cầu. |
| Dễ dàng cấu hình | Cấu hình rất đơn giản, là mặc định trong NGINX. | Dễ dàng cấu hình với chỉ một lệnh random. |
| Phân phối tải | Phân phối đều giữa các server, không có yếu tố ưu tiên. | Phân phối không đều, có thể dẫn đến tải không đều. |
| Ứng dụng | Thích hợp khi tất cả các server có khả năng xử lý giống nhau và không cần yêu cầu ưu tiên. | Thích hợp khi bạn muốn phân phối ngẫu nhiên, không theo quy luật cụ thể. |
| Mức độ kiểm soát | Phân phối đều và dễ dự đoán. Không có khả năng kiểm soát tỷ lệ phân phối. | Phân phối ngẫu nhiên, không thể dự đoán được. |
| Độ phân tán yêu cầu | Các yêu cầu phân phối đồng đều theo thứ tự, không có sự thiên lệch. | Các yêu cầu có thể phân tán không đều, phụ thuộc vào sự may mắn. |
| Khả năng tối ưu hóa | Phù hợp với hệ thống có nhiều server đồng đều và không cần phân biệt tài nguyên. | Phù hợp trong những tình huống có sự bất đối xứng giữa các server (ví dụ: các server có tài nguyên khác nhau). |
| Khả năng đáp ứng khi server chết | Nếu một server chết, NGINX sẽ tiếp tục phân phối đến các server còn lại. | Tương tự như Round Robin, nhưng có thể gây ra phân phối không đồng đều nếu server ngẫu nhiên bị bỏ qua quá lâu. |

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-2)