

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
2	Hồ Vi Khánh	22520633	22520633@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

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.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Cài đặt Apache và NGINX

a) Cài đặt Apache

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

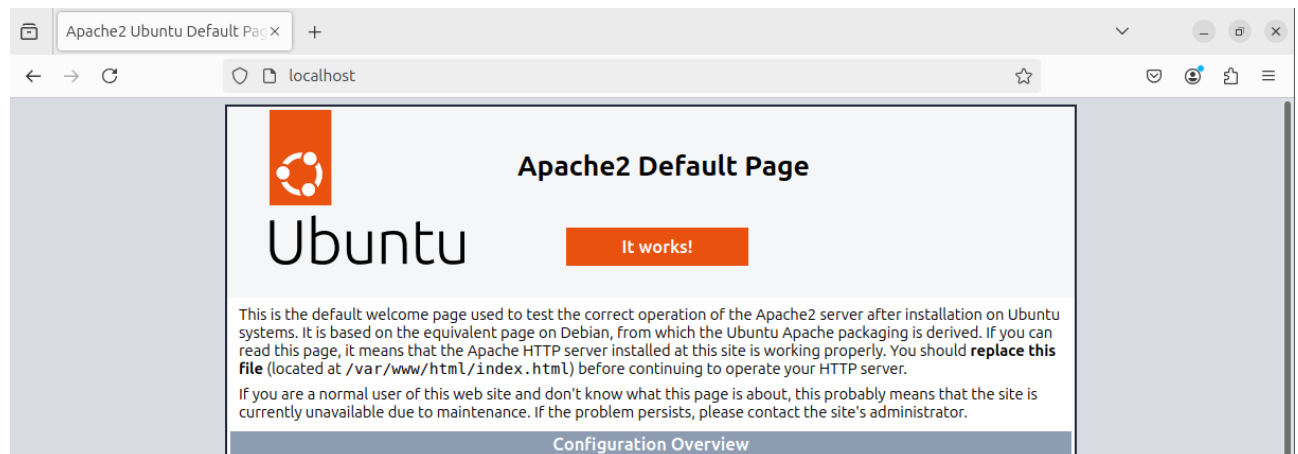
```
$sudo apt update
```

```
$sudo apt upgrade -y
```

```
$sudo apt install apache2 -y
```

```
web1@web1-VMware-Virtual-Platform:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
web1@web1-VMware-Virtual-Platform:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  evince evince-common libevdocument3-4t64 libevview3-3t64 python3-distupgrade
  ubuntu-release-upgrader-core ubuntu-release-upgrader-gtk
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
web1@web1-VMware-Virtual-Platform:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.5).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
web1@web1-VMware-Virtual-Platform:~$
```

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



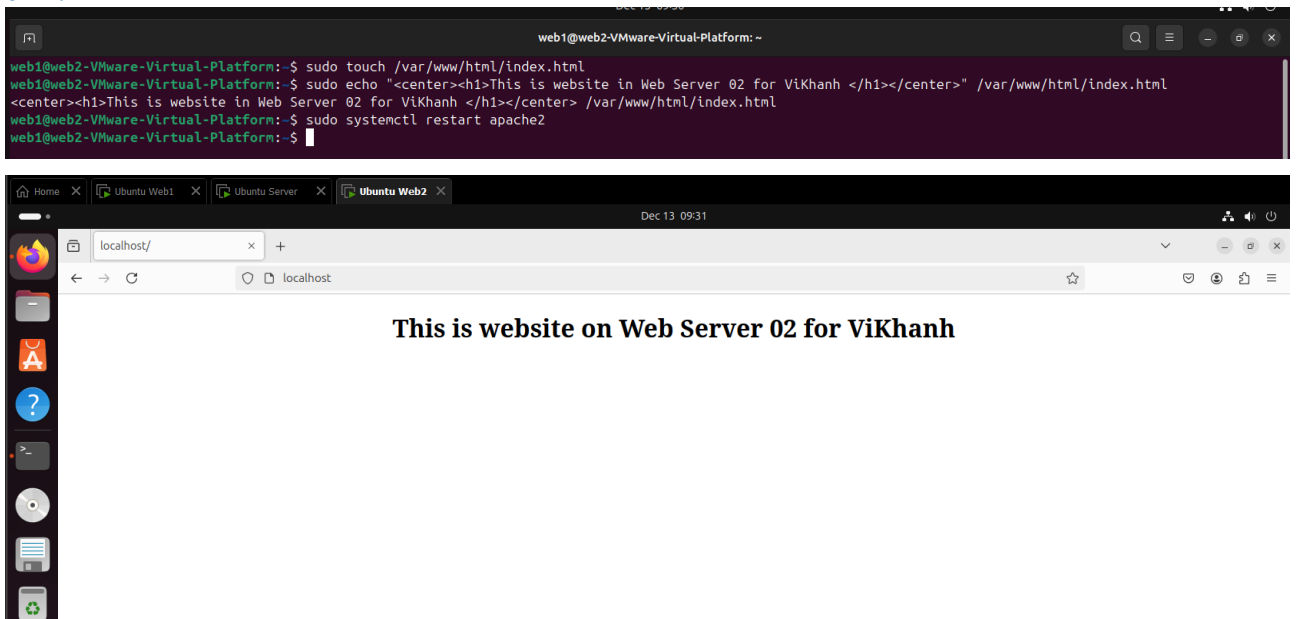
Bây giờ chúng ta sẽ tạo một website đơn giản trên server Web1.

```
web1@web1-VMware-Virtual-Platform:~$ sudo touch /var/www/html/index.html
web1@web1-VMware-Virtual-Platform:~$ sudo echo "<center><h1>This is website on Web Server 01 for ViKhanh</h1></center>" /var/www/html/index.html
web1@web1-VMware-Virtual-Platform:~$ sudo systemctl restart apache2
```

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”.



b) Cài đặt NGINX

Trên máy ảo NGINX LB, thực hiện cài đặt NGINX

```
server@Group03-Server:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.1).
0 upgraded, 0 newly installed, 0 to remove and 220 not upgraded.
server@Group03-Server:~$ sudo systemctl start nginx
server@Group03-Server:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
server@Group03-Server:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-12-13 08:20:30 +07; 2min 34s ago
     Docs: man:nginx(8)
    Main PID: 6230 (nginx)
      Tasks: 3 (limit: 4558)
     Memory: 2.4M (peak: 2.5M)
        CPU: 49ms
    CGroup: /system.slice/nginx.service
            └─6230 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─6232 "nginx: worker process"
                └─6233 "nginx: worker process"

Dec 13 08:20:30 Group03-Server systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Dec 13 08:20:30 Group03-Server systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
server@Group03-Server:~$
```

2. Cấu hình NGINX**a. 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

```
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 Group03-Server

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
#lab6
127.0.0.1 nginx.test
```

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

```
server {
    listen 8080;
    server_name nginx.test;
    location / {
        proxy_pass http://server1_ip:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

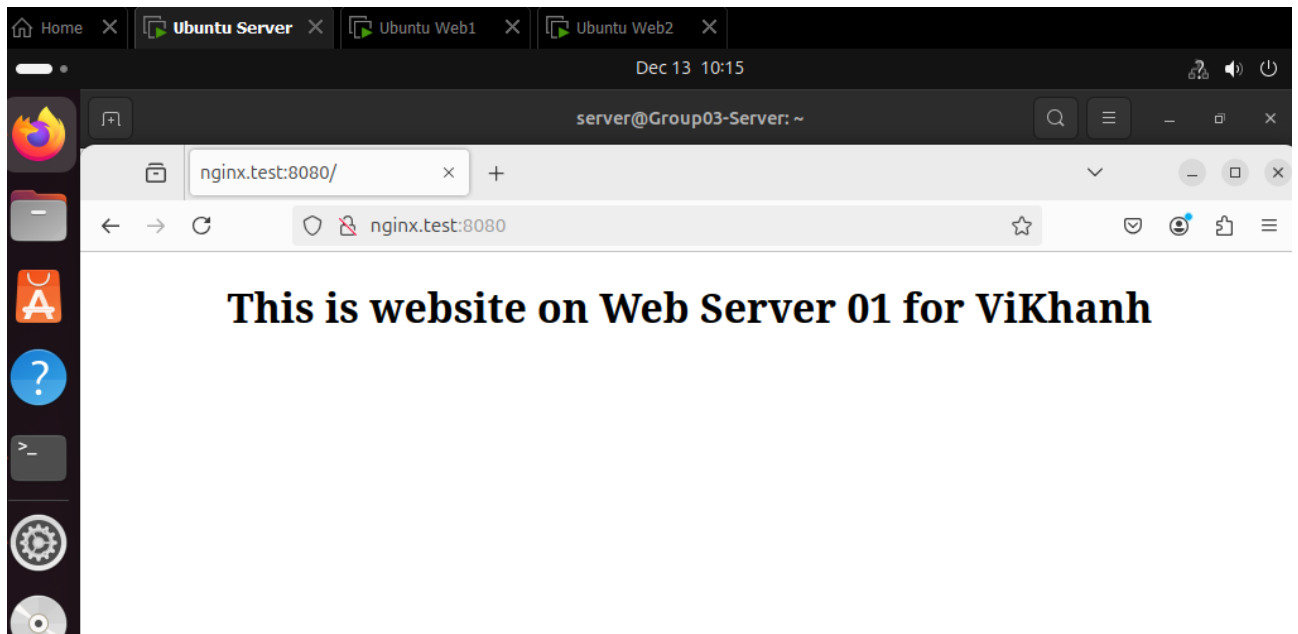
```
sendfile on;
tcp_nopush on;
types_hash_max_size 2048;
# server_tokens off;

# server_names_hash_bucket_size 64;
# server_name_in_redirect off;

include /etc/nginx/mime.types;
default_type application/octet-stream;
server {
    listen 8080;
    server_name nginx.test;
    location / {
        proxy_pass http://192.168.10.3;
    }
}
##
# SSL Settings
##
```

```
server@Group03-Server:~$ sudo nano /etc/nginx/nginx.conf
server@Group03-Server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
server@Group03-Server:~$ sudo systemctl reload nginx
server@Group03-Server:~$
```

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.



b. Load Balancing

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

```
server@Group03-Server:~$ sudo nano /etc/nginx/nginx.conf
server@Group03-Server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
server@Group03-Server:~$ sudo systemctl reload nginx
server@Group03-Server:~$
```

Chỉnh sửa thành

```
GNU nano 7.2 /etc/nginx/nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ## Existing HTTP settings go here
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    upstream backend {
        server 192.168.10.3;
        server 192.168.10.4;
    }
    ## Server Configuration
    server {
        listen 8080;
        server_name nginx.test;

        location / {
            proxy_pass http://backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }
}
```

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:



This is website on Web Server 01 for ViKhanh



This is website on Web Server 02 for ViKhanh

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 luân 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 lý 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 lý 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

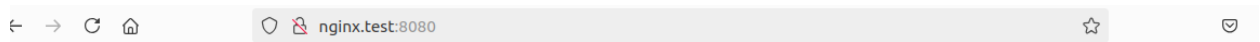
Cấu hình

```
upstream backend {  
    random;  
    server 192.168.10.130;  
    server 192.168.10.128;  
}
```

Thêm dòng random;

Test thử làm 1 kịch bản

Kết quả truy cập ngẫu nhiên



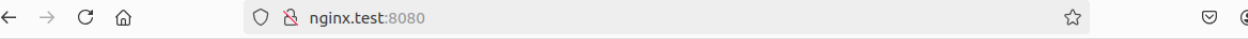
This is website on Web Server 01

Lần 1



This is website on Web Server 01

Lần 2



This is website on Web Server 02

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.