

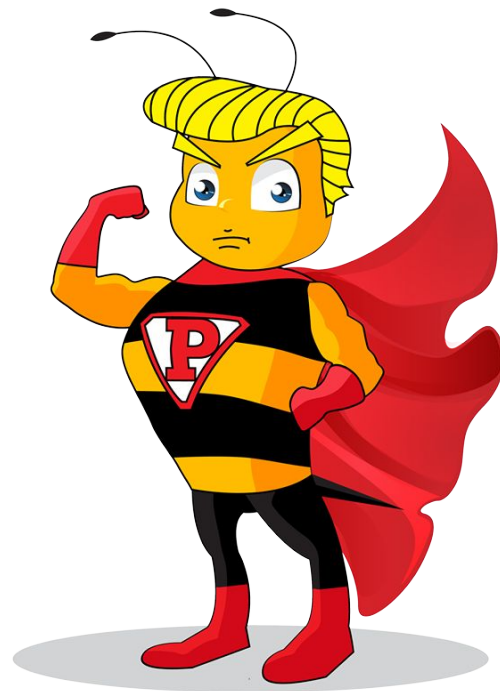
SOA204

PHẦN MỀM MIỄN PHÍ VÀ MÃ NGUỒN MỞ

OPEN SOURCE & FREE SOFTWARE

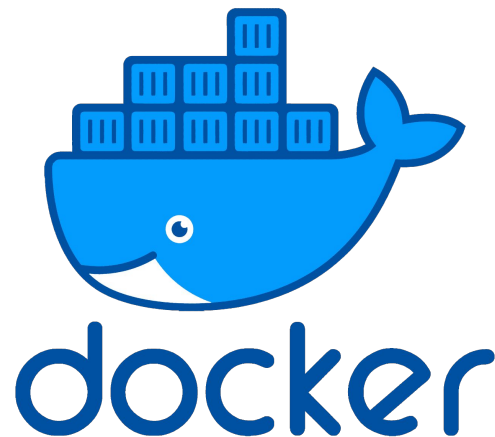
BÀI 5:

Cài đặt và cấu hình Webserver trên
hệ điều hành Linux

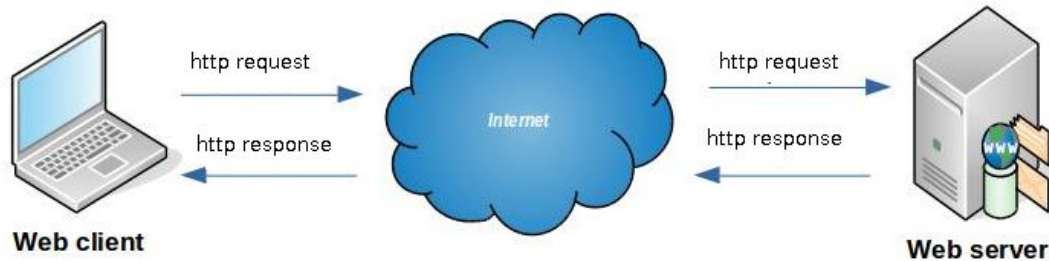


<https://caodang.fpt.edu.vn>

- ❖ Khái niệm ảo hóa
- ❖ Giới thiệu về Docker
- ❖ Hướng dẫn cài đặt và sử dụng Docker



- ❖ Cấu hình TCP/IP trên Ubuntu
- ❖ Giới thiệu về dịch vụ Web Server.
- ❖ Cài đặt và cấu hình nginx
- ❖ Cài đặt và cấu hình MySQL
- ❖ Cài đặt PHP



CẤU HÌNH TCP/IP TRÊN UBUNTU

KIỂM TRA CẤU HÌNH:

```
ip a
// hoặc
ipconfig
```

```
zaira@Zaira:/etc/netplan$ ip a
1: lo: <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
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 3a:53:de:81:be:d7 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 1e:ba:a9:c6:ea:0f brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
   link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
   link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:15:5d:df:c3:ad brd ff:ff:ff:ff:ff:ff
   inet 172.23.199.129/20 brd 172.23.207.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::215:5dff:fedf:c3ad/64 scope link
       valid_lft forever preferred_lft forever
```

XEM CHI TIẾT SUBNET MASK:

```
ifconfig -a
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 172.23.199.129 netmask 255.255.240.0 broadcast 172.23.207.255
      inet6 fe80::215:5dff:fedf:c3ad prefixlen 64 scopeid 0x20<link>
      ether 00:15:5d:df:c3:ad txqueuelen 1000 (Ethernet)
      RX packets 250754 bytes 456747743 (456.7 MB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 98277 bytes 13457632 (13.4 MB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

IP là 172.23.199.129 và subnet mask là 255.255.240.0

THAY ĐỔI CẤU HÌNH:

- **Netplan** là công cụ quản lý mạng mặc định, sử dụng file cấu hình **.yaml**
- Di chuyển đến **/etc/netplan**
- Sử dụng lệnh **ls** để xem danh sách files và thư mục trong **etc/netplan**
- Nếu không tồn tại file có đuôi **.yaml** thì hãy tạo một file **.yaml** và đặt tên, ví dụ: **01-network-manager-all.yaml** và thêm nội dung sau:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    eth0:
      dhcp4: no
      addresses: [172.23.207.254/20]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8,8.8.8.4]
```

THỬ CẤU HÌNH MỚI:

- Nếu không có lỗi gì, bạn sẽ được hỏi nếu muốn áp dụng cấu hình này.

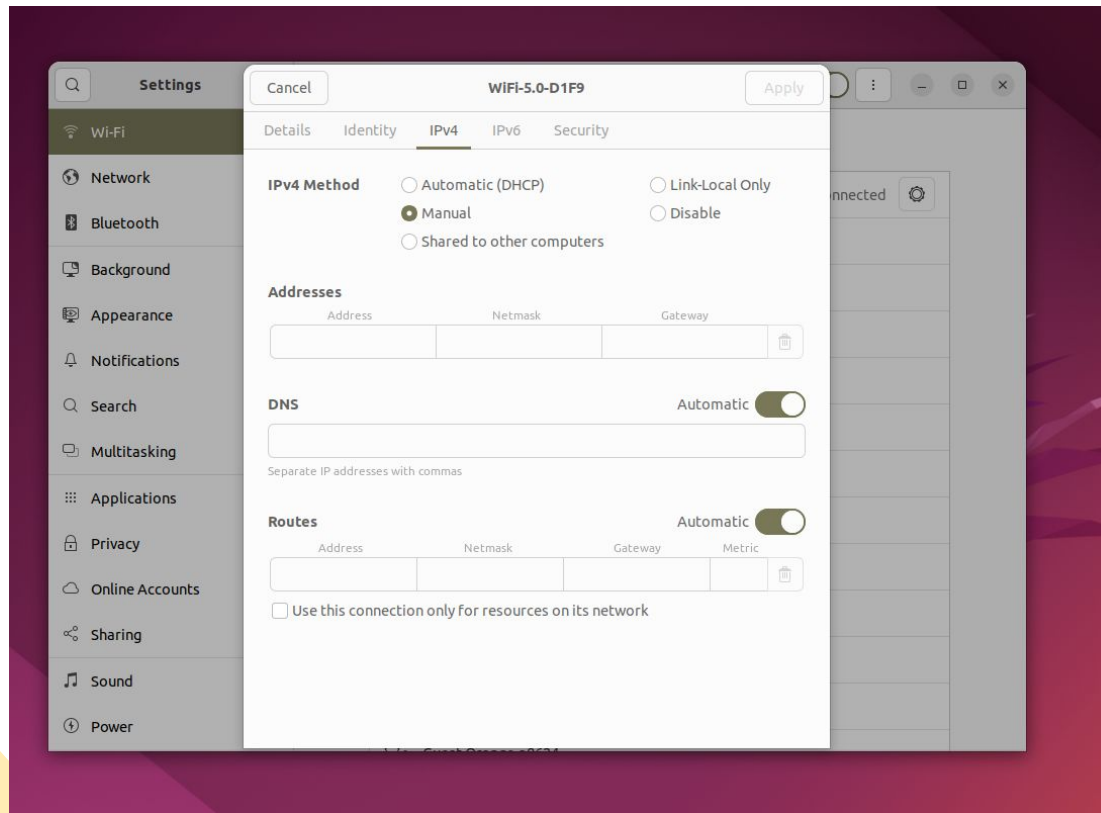
```
sudo netplan try
```


KIỂM TRA THAY ĐỔI:

ip a

```
zaira@Zaira:~$ ip a
1: lo: <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
        valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 3a:53:de:81:be:d7 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 1e:ba:a9:c6:ea:0f brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:df:c3:ad brd ff:ff:ff:ff:ff:ff
    inet 172.23.207.254/20 brd 172.23.207.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fedf:c3ad/64 scope link
        valid_lft forever preferred_lft forever
```

CẤU HÌNH BẰNG GUI:



WEB SERVER & CÀI ĐẶT NGINX TRÊN LINUX



Web server là máy chủ cài đặt các chương trình phục vụ các ứng dụng web. Web server có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao thức HTTP hoặc các giao thức khác. Có nhiều phần mềm web server khác nhau như: *Apache*, *Nginx*, *IIS*, ... Là



VS

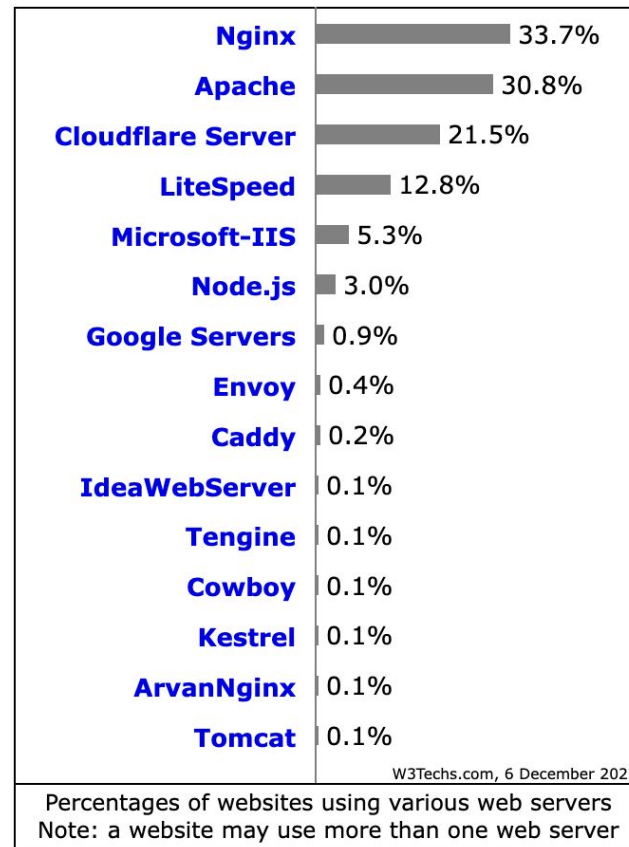


VS



ĐỘ PHỔ BIẾN:

- Nginx đã soán ngôi của Apache để trở thành Web Server phổ biến nhất hiện nay.



APACHE:

- Apache là phần mềm web server miễn phí mã nguồn mở. Nó đang chiếm đến thị phần thứ 2 sau Nginx. Tên chính thức của Apache là Apache HTTP Server, được điều hành và phát triển bởi Apache Software Foundation.
- Nó giúp chủ website đưa nội dung lên web – vì vậy nó hay được gọi là web server. Apache là một trong số những web server lâu đời và đáng tin cậy nhất, phiên bản đầu tiên đã được ra mắt từ hơn 20 năm trước, tận những năm 1995.



IIS:

- IIS được đính kèm với các phiên bản của Windows.
- Microsoft Internet Information Services (các dịch vụ cung cấp thông tin Internet) là các dịch vụ dành cho máy chủ chạy trên nền Hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server,...



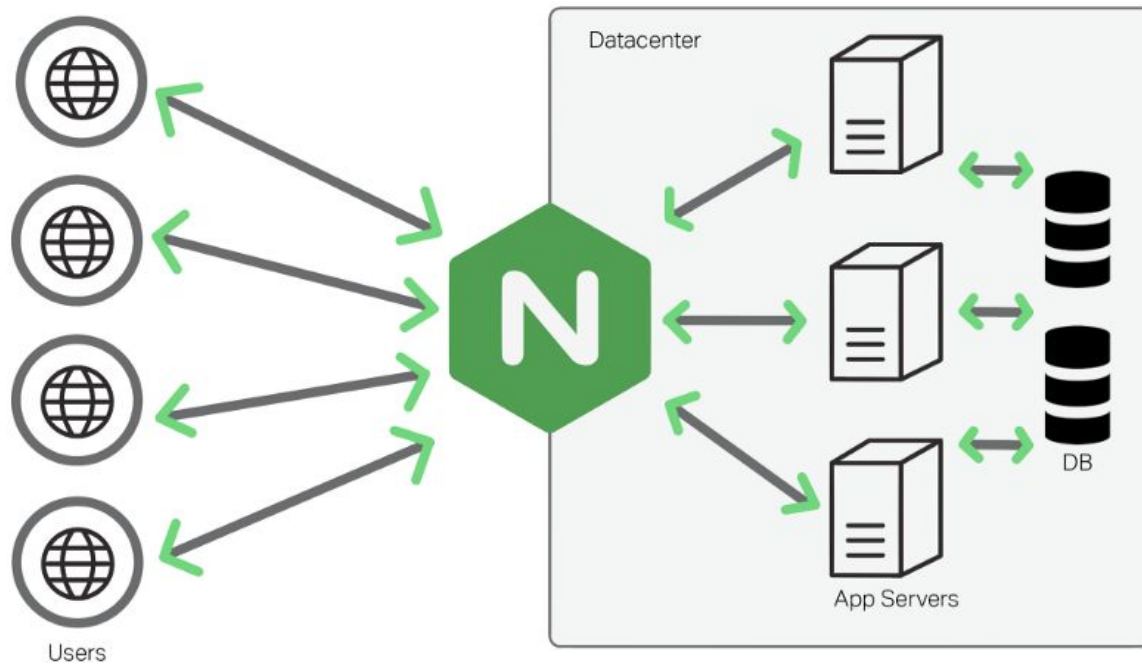
NGINX:

- NGINX là một web server mạnh mẽ mã nguồn mở. Nginx sử dụng kiến trúc đơn luồng, hướng sự kiện vì thế nó hiệu quả hơn Apache server. Ngoài ra nginx được sử dụng như load balancing, HTTP caching, hay sử dụng như một reverse proxy. Nginx là kiến thức không thể thiếu đối với một web developer, system administrator hay devops.

NGINX

Năm 2002, Igor Sysoev phát triển Nginx ban đầu và công bố lần đầu vào năm 2004. Tháng 7 năm 2011, công ty Nginx Inc được thành lập với trụ sở đầu tiên tại San Francisco, California, USA.

NGINX:



TÍNH NĂNG CƠ BẢN:

- Phục vụ các tệp tĩnh và chỉ mục, tự động lập chỉ mục; mở bộ đệm mô tả tập tin.
- Tăng tốc ủy quyền ngược bằng bộ nhớ đệm; cân bằng tải và khả năng chịu lỗi.
- Hỗ trợ tăng tốc với bộ nhớ đệm của máy chủ FastCGI, uwsgi, SCGI và memcached.
- Kiến trúc mô-đun. Các bộ lọc bao gồm gzipping, phạm vi byte, phản hồi chunked, XSLT, SSI và bộ lọc chuyển đổi hình ảnh. Nhiều phần bao gồm SSI trong một trang có thể được xử lý song song nếu chúng được xử lý bởi máy chủ proxy hoặc FastCGI/uwsgi/SCGI.
- Hỗ trợ SSL và TLS SNI.
- Hỗ trợ HTTP/2 với mức độ ưu tiên dựa trên trọng số và phụ thuộc.
- Hỗ trợ HTTP/3.

TÍNH NĂNG HTTP KHÁC:

- Máy chủ ảo dựa trên tên và dựa trên IP.
- Hỗ trợ kết nối liên tục và kết nối pipelined.
- Truy cập các định dạng nhật ký, ghi nhật ký vào bộ đệm, xoay nhật ký nhanh và ghi nhật ký hệ thống;
- Chuyển hướng mã lỗi 3xx-5xx;
- Mô-đun viết lại: URI thay đổi bằng cách sử dụng biểu thức chính quy.
- Thực hiện các chức năng khác nhau tùy thuộc vào địa chỉ khách hàng.
- Kiểm soát truy cập dựa trên địa chỉ IP của khách hàng, bằng mật khẩu (xác thực HTTP Basic) và theo kết quả của subrequest.

TÍNH NĂNG HTTP KHÁC:

- Xác thực HTTP referrer (người giới thiệu).
- Các phương thức PUT, DELETE, MKCOL, COPY và MOVE.
- Truyền phát FLV và MP4.
- Giới hạn tỉ lệ phản hồi.
- Giới hạn số lượng kết nối hoặc yêu cầu đồng thời đến từ một địa chỉ.
- Định vị địa lý dựa trên IP.
- Thử nghiệm A/B.
- Sao chép yêu cầu (request)
- Perl nhúng.
- Ngôn ngữ kịch bản njs.

TÍNH NĂNG MAIL PROXY SERVER:

- Chuyển hướng người dùng đến máy chủ IMAP hoặc POP3 bằng máy chủ xác thực HTTP bên ngoài.
- Xác thực người dùng bằng máy chủ xác thực HTTP bên ngoài và chuyển hướng kết nối đến máy chủ SMTP nội bộ.
- Các phương thức xác thực:
- POP3: USER/PASS, APOP, AUTH LOGIN/PLAIN/CRAM-MD5.
- IMAP: LOGIN, AUTH LOGIN/PLAIN/CRAM-MD5.
- SMTP: AUTH LOGIN/PLAIN/CRAM-MD5.
- Hỗ trợ SSL.
- Hỗ trợ STARTTLS và STLS.

CÁCH CÀI ĐẶT:

- Có 2 cách cài đặt nginx đó là:
 - Sử dụng gói (package) dựng sẵn.
 - Cài đặt từ mã nguồn (source).
- Phương thức đầu tiên dễ và nhanh hơn, nhưng cài đặt từ source cung cấp khả năng cài đặt thêm các module khác giúp NGINX mạnh mẽ hơn. Nó cho phép chúng ta tùy chỉnh cho phù hợp với nhu cầu của ứng dụng.

CÀI ĐẶT BẰNG GÓI:

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

KIỂM TRA SAU CÀI ĐẶT:

```
sudo nginx -v
```

Webserver mới sẽ được cài đặt tại **/etc/nginx/**. Nếu bạn vào trong thư mục này, bạn sẽ thấy nhiều tệp tin và thư mục. Nhưng thứ quan trọng nhất cần chú ý là tệp tin **nginx.conf** và thư mục **sites-available**.

nginx.conf:

```
(1/2)
upstream remoteApplicationServer {
    server 10.10.10.10;
}
upstream remoteAPIServer {
    server 20.20.20.20;
    server 20.20.20.21;
    server 20.20.20.22;
    server 20.20.20.23;
}
server {
    listen 80;
    server_name www.customapp.com customapp.com
    root /var/www/html;
    index index.html

    location / {
        alias /var/www/html/customapp/;
        try_files $uri $uri/ =404;
    }
}
```

```
(2/2)
location /remoteapp {
    proxy_set_header    Host
$host:$server_port;
    proxy_set_header    X-Real-IP        $remote_addr;
    proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_pass http://remoteAPIServer/;
}

location /api/v1/ {
    proxy_pass https://remoteAPIServer/api/v1/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
    proxy_redirect http:// https://;
}
}
```


CẤU HÌNH QUAN TRỌNG:

- **worker_processes:** Thiết lập này định nghĩa số worker processes mà NGINX sẽ sử dụng. Bởi vì NGINX là đơn luồng (single threaded), nó thường bằng với số lõi CPU.
- **worker_connection:** Đây là số lượng tối đa của các kết nối đồng thời cho mỗi worker process và nói cho các worker process của chúng ta có bao nhiêu người có thể được phục vụ đồng thời bởi NGINX.
- **access_log & error_log:** Đây là những tệp tin mà NGINX sẽ sử dụng để log bất kỳ lỗi và số lần truy cập. Các bản ghi này thường được sử dụng để gỡ lỗi hoặc sửa chữa.

CẤU HÌNH QUAN TRỌNG:

- **gzip:** Đây là các thiết lập nén GZIP của các NGINX response. Tính năng này có nhiều thiết lập phụ, phần bị comment bởi mặc định có thể giúp hiệu suất được cải thiện đáng kể. Trong các thiết lập phụ của GZIP, cần quan tâm tới `gzip_comp_level`, nó là mức nén và nằm trong khoảng từ 1 tới 10. Thông thường, giá trị này không nên lớn hơn 6 — trên mức này lợi ích từ việc nén là không đáng kể, vì nó cần sử dụng nhiều CPU hơn. `gzip_types` là một danh sách các kiểu response sẽ được nén.

sites-available:

- NGINX có thể hỗ trợ nhiều hơn một website, và các tệp tin định nghĩa các trang web của bạn ở trong thư mục **/etc/nginx/sites-available**.
- Tuy nhiên, các tệp tin trong thư mục này không “live” — bạn có thể có nhiều tệp tin định nghĩa các trang web ở đây, nhưng NGINX không thực sự làm bất cứ điều gì với chúng trừ khi chúng được **symlink** (*liên kết tượng trưng*) tới thư mục **/etc/nginx/sites-enabled** (bạn cũng có thể *copy chúng tới thư mục này, nhưng symlink đảm bảo rằng chỉ có duy nhất một bản copy của mỗi tệp tin được theo dõi*).
- Việc này cung cấp một phương thức để nhanh chóng đưa các trang web online hoặc offline mà không cần phải thực sự xóa bất kỳ tệp tin nào — khi bạn sẵn sàng cho một trang web online, tạo symlink tới **sites-enabled** và khởi động lại NGINX.

sites-available:

- Cấu hình thông thường:

CẤU HÌNH sites-available:

- Ngữ cảnh **server** định nghĩa một **server ảo** để xử lý các request từ client của bạn. **Bạn có thể có nhiều khối server**, và NGINX sẽ chọn một trong số chúng dựa trên các chỉ thị **listen** và **server_name**.
- Trong một khối server, chúng ta định nghĩa nhiều ngữ cảnh **location** được sử dụng để quyết định cách xử lý các request từ client. Bất cứ khi nào một request đến, NGINX sẽ thử khớp URI tới một trong số các định nghĩa **location** và xử lý nó cho phù hợp.
 - *try_files sẽ cố gắng phục vụ các tệp tin tĩnh được tìm thấy trong thư mục được trỏ tới bởi chỉ thị gốc.*
 - *proxy_pass sẽ gửi request tới một proxy server cụ thể.*
 - *rewrite sẽ viết lại URI tới dựa trên một regular expression để một khối location có thể xử lý nó.*

KHỞI ĐỘNG:

```
sudo service nginx start
```

KHỞI ĐỘNG LẠI:

- Bất cứ khi nào chúng ta thay đổi cấu hình, chúng ta chỉ cần tải lại (không có thời gian downtime) sử dụng lệnh dưới đây:

```
service nginx reload
```

KIỂM TRA TRẠNG THÁI NGINX:

```
service nginx status
```

CÀI ĐẶT MYSQL TRÊN UBUNTU

CÀI ĐẶT MySQL Client:

- Sử dụng để kết nối từ xa tới máy chủ

```
sudo apt update  
sudo apt install mysql-client -y
```

KIỂM TRA VERSION:

```
mysql -V
```

KẾT NỐI TỚI MÁY CHỦ:

```
mysql -u <username> -p <password> -h <hostname hoặc ip>
```


CÀI ĐẶT MySQL Server:

```
sudo apt update  
sudo apt install mysql-server
```

CẤU HÌNH BẢO MẬT TỰ ĐỘNG:

```
sudo mysql_secure_installation
```

CÀI ĐẶT MẬT KHẨU ROOT:

Please set the password for root here.

New password:

Re-enter new password:

CÁC CÂU HỎI SAU KHI ĐẶT MẬT KHẨU:

- Xóa người dùng thử nghiệm ẩn danh
- Tắt đăng nhập từ xa từ người dùng gốc (root user)
- Xóa cơ sở dữ liệu thử nghiệm
- Tải lại bảng đặc quyền (privileges table) để lưu lại toàn bộ cài đặt

ĐĂNG NHẬP:

```
mysql -u <username> -p <password>
```

DANH SÁCH CSDL:

```
SHOW DATABASES;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.00 sec)
```

TẠO CSDL:

```
mysql> CREATE DATABASE pets;  
Query OK, 1 row affected (0.01 sec)
```

SỬ DỤNG CSDL:

```
mysql> USE pets  
Database changed
```

TẠO BẢNG:

```
CREATE TABLE cats
(
  id            INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the
record
  name          VARCHAR(150) NOT NULL,                # Name of the cat
  owner         VARCHAR(150) NOT NULL,                # Owner of the cat
  birth         DATE NOT NULL,                        # Birthday of the cat
  PRIMARY KEY   (id)                                  # Make the id the
primary key
);
```

XEM TẤT CẢ CÁC BẢNG:

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_pets |  
+-----+  
| cats           |  
+-----+  
1 row in set (0.00 sec)
```

XEM TẤT CẢ CỘT TRONG BẢNG:

```
mysql> DESCRIBE cats;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name  | varchar(150)    | NO   |     | NULL    |                |
| owner | varchar(150)    | NO   |     | NULL    |                |
| birth | date            | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

THÊM BẢN GHI VÀO BẢNG:

```
INSERT INTO cats ( name, owner, birth) VALUES
( 'Sandy', 'Lennon', '2015-01-03' ),
( 'Cookie', 'Casey', '2013-11-13' ),
( 'Charlie', 'River', '2016-05-21' );
```

XEM DỮ LIỆU BẢNG:

```
SELECT * FROM cats;
+----+-----+-----+-----+
| id | name   | owner  | birth      |
+----+-----+-----+-----+
|  1 | Sandy  | Lennon | 2015-01-03 |
|  2 | Cookie | Casey  | 2013-11-13 |
|  3 | Charlie | River  | 2016-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```


CÀI ĐẶT PHP

CÀI ĐẶT :

```
sudo apt install php libapache2-mod-php
```

CÀI CÁC GÓI BỔ SUNG:

```
sudo apt install php-cli
```

```
sudo apt install php-cgi
```

```
sudo apt install php-mysql
```

KIỂM TRA CÀI ĐẶT :

```
<?php  
    phpinfo() ;  
?>
```

DEMO

- Giảng viên demo cài đặt môi trường và chạy thử mã nguồn PHP

- Web server là máy chủ cài đặt các chương trình có khả năng tiếp nhận request, gửi phản hồi đến client thông qua HTTP.
- Nginx là phần mềm web server phổ biến nhất hiện nay.

- ❖ **Hệ quản trị quan hệ khách hàng (CRM) mã nguồn mở.**

THANK YOU!