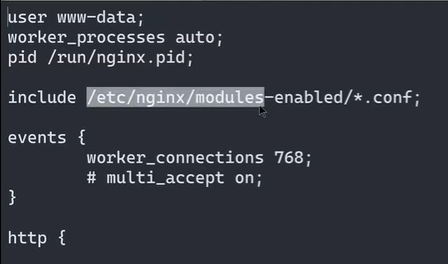
Nginx

Config chính của nginx là ở /etc/nginx/nginx.conf

File này sẽ quyết định **nạp** cấu hình ở các folder khác

VD: nạp cấu hình là ở **include**



Hiện tại n chỉ đang đọc thêm cấu hình ở modules-enabled

Bước 1: xác định các file cấu hình được nạp

include /etc/nginx/sites-enabled/\*;

* Tạo 1 folder cho site mới

VD: /etc/nginx/sites-enabled/vietdung

B2: thiết lập config server site (config phải được nạp, include trong /etc/nginx/nginx.conf)

nano /etc/nginx/sites-enabled/vietdung

    server {

        listen 80 ;

        listen [::]:80 ;

        root /var/www/vietdung/html;

        index index.html index.html;

        server\_name vietdung.com  www.vietdung.com  haha.com;

        location / {

            # alias /usr/share/nginx/html;

            try\_files $uri $uri/ =404;

        }

    }

root /var/www/vietdung/html; : thư mục chứa source code

index index.html index.html; : đọc những file này trong thư mục root

server name: đặt bao nhiêu tên server name cho nó

        location / {

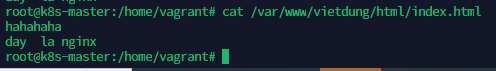
            try\_files $uri $uri/ =404;

        }

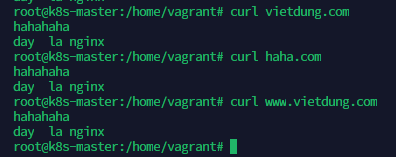
location / : bắt tất cả request

try\_files : nó sẽ cố tìm file trong index, nếu ko có sẽ trả về 404

B3: tạo thư mục đọc (index) trong folder root

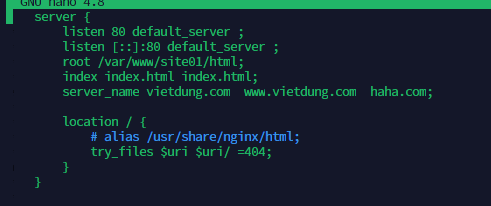


Và check



## Nginx default

Trong config nginx chỉ được config 1 cái cấu hình default server



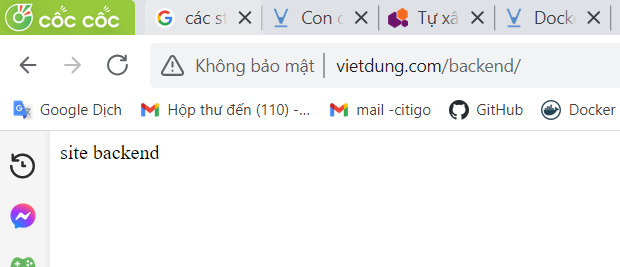
Tức vào nginx port 80 thì sẽ vào đây, ăn cấu hình mặc định ở đây, khi trỏ vào nginx mà ko thuộc rule nào của nginx thì n sẽ đi vào cấu hình của con này, con **vietdung.com**

## Path nginx

Ta có thêm 1 file index.html trong folder backend của thư mục nginx



* Khi vào domain …./backend n sẽ load nội dung trong folder backend



## Public domain

Vì nginx chỉ có nhiệm vụ forward domain, nên nếu muốn gọi domain này từ ngoài public hoặc từ máy nào đó thì ta có 2 cách. Cốt lõi là phải đăng ký domain với DNS server

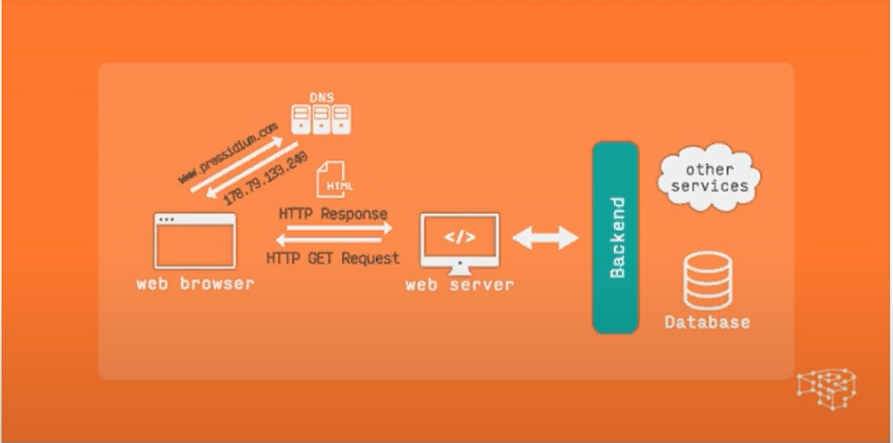
Cách 1: đăng ký domain, assign domain này vs 1 IP, IP này được định nghĩa để các client biết

Cách 2: thêm 192.168.50.10 vietdung.com trong /etc/hosts

Vì bản chất n cũng tương tự như hỏi domain

Hoặc trong app thì NAT IP nginx ra vs 1 public IP

Trong app định nghĩa IP public kia là DNS server

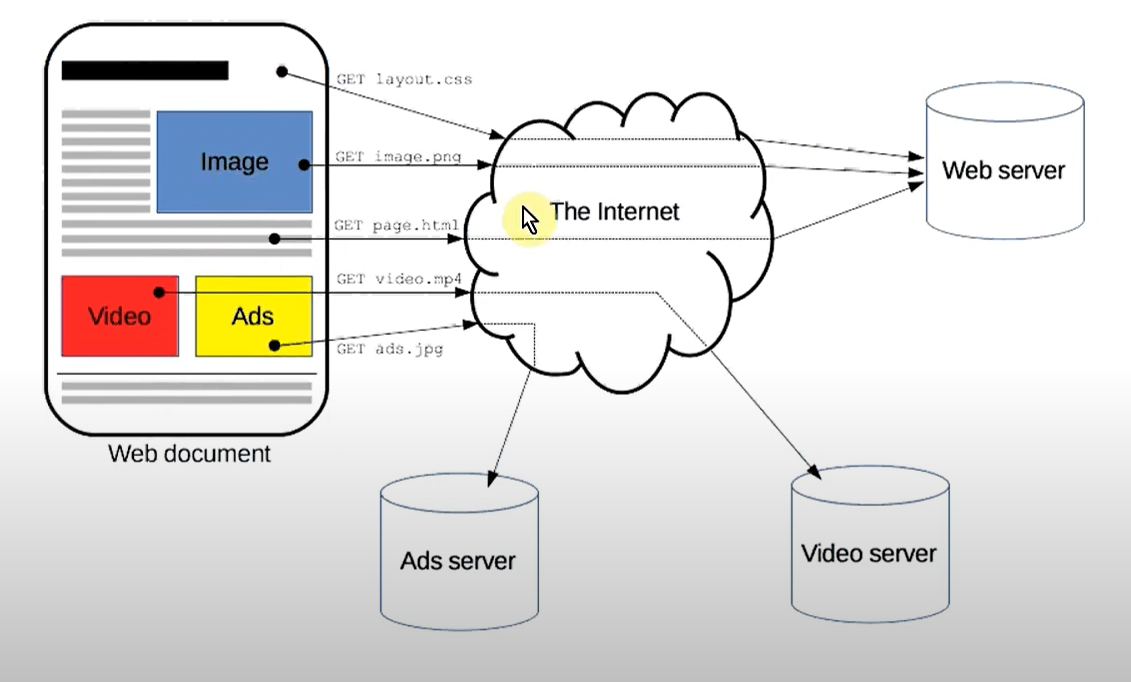


Điều gì xảy ra khi gõ 1 domain vào trình duyệt

Trước hết nó sẽ hỏi thông tin domain với DNS server, rồi DNS server sẽ IP trang web chúng ta truy cập, đó là IP public, IP public này lại được **NAT** vs IP webserver bên trong

Và từ đó nó sẽ đi vào webserver

Cách khác là mình trỏ thẳng domain này tới webserver trong file /etc/hosts



Rất nhiều request đến web server và có thể request rất nặng

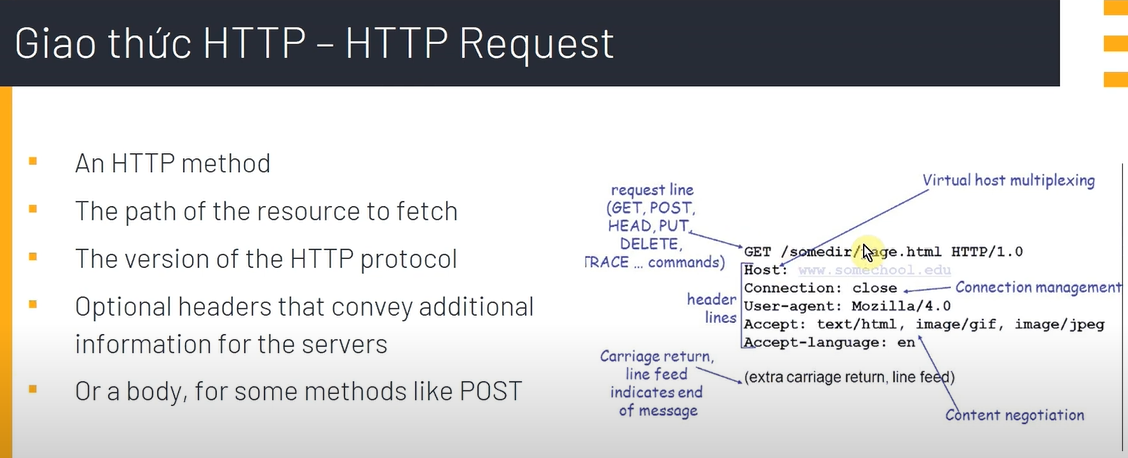
HTTP là giao thức client – server

HTTP là giao thức stateless, but no sessionless

Stateless là khi gửi 1 request tới server và server respone request đó tới client thì yêu cầu này gói gọn trong 1 request

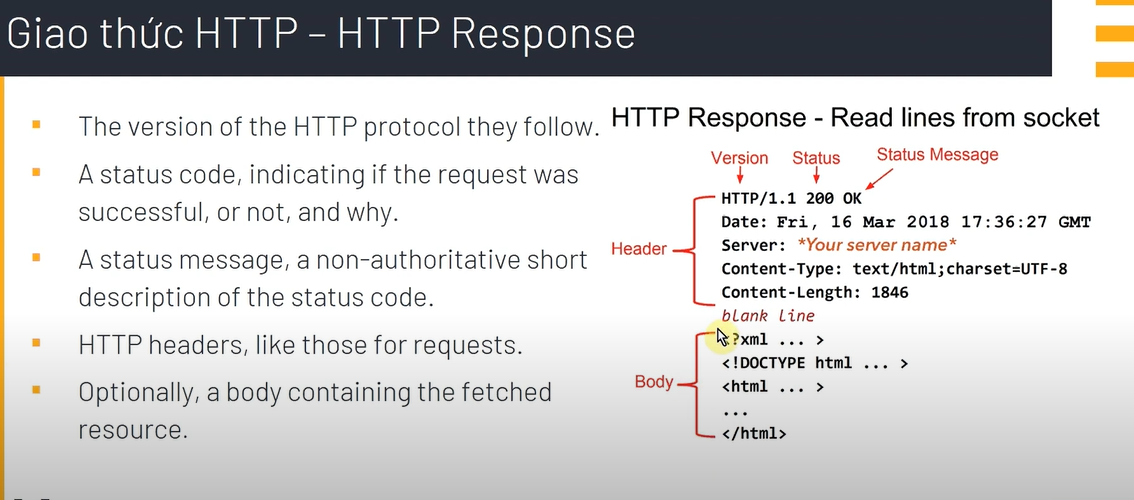
Nếu thực hiện 1 request thứ 2 thì server vẫn trả về request nhưng n sẽ coi client là 1 client khác

Vì vậy sessionless sẽ giữ phiên người dùng, lần đầu tiên request lên, server sẽ sinh ra key để giữ session cho những request sau, để server nhận biết client như ng quen, có 1 số request mà yêu cầu authen, mà request đầu authen rồi, request sau ko giữ session authen thì n sẽ tèo

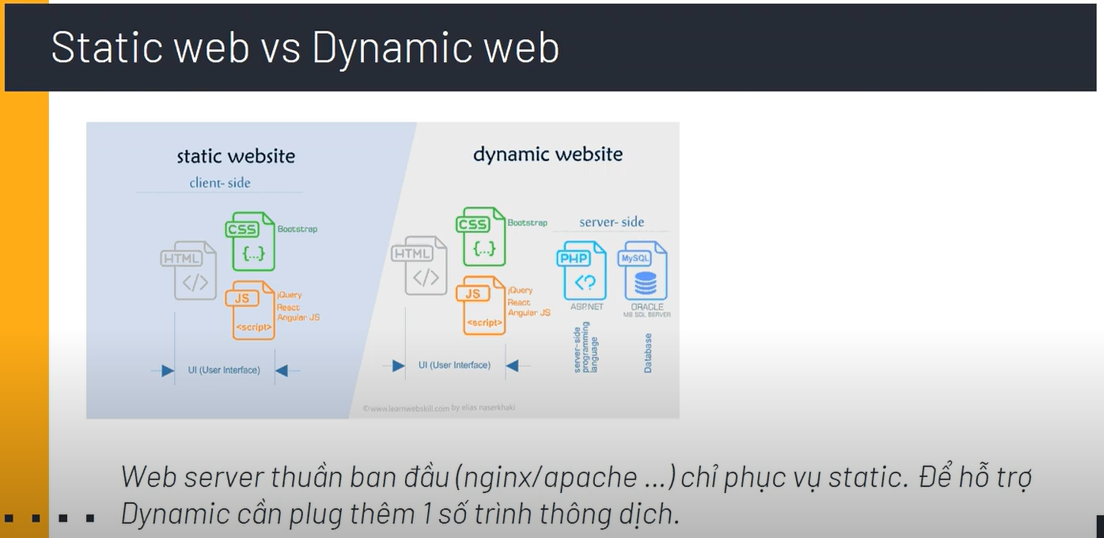


HTTP request gồm 2 phần là header và data

Accept: chấp nhận kiểu dữ liệu trả về là gì text (index.html), image ....



HTTP response cũng 2 phần: header và body



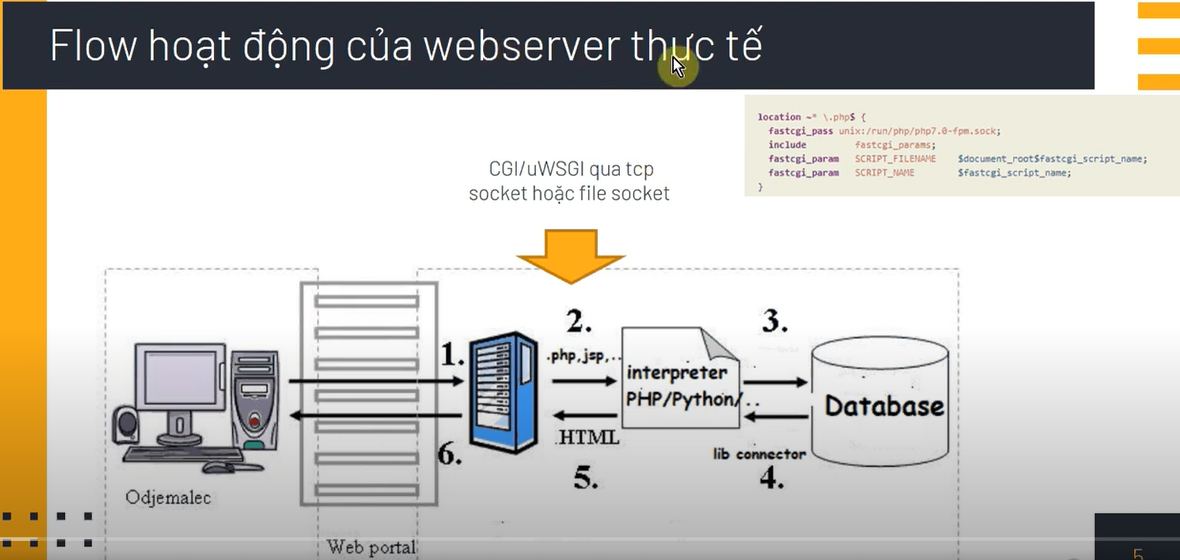
Static web là công nghệ web mà phía server site chỉ là document HTML mà thôi

Vì vậy 10 người vào cùng 1 URL thì n trả kết quả về y hệt nhau như thế, đơn giản nội dung trong HTML đấy đã fix cứng rồi

Ngược lại vs static web thì dynamic web là công nghệ web, phía server site ngoài HTML, CSS, js ra thì ngta còn sử dụng 1 số ngôn ngữ kịch bản sinh ra nội dung web như PHP, ASP, nodejs , nó ko fix cứng

10 người truy cập nhưng với 10 nội dung khác nhau

* Server thuần ban đầu (nginx / apache...) chỉ phục vụ thuần static web, để hỗ trợ Dynamic web thì nginx, apache cần plugin thêm 1 số trình thông dịch như php, nodejs python



Những trang website này sẽ được viết dựa vào ngôn ngữ như php, ASP, python, nodejs

* Nginx, apache ko thể đọc, xử lý file của php, nodejs, python, vì vậy webserver forward file php đó sag trình thông dịch php, python (trình thông dịch này hđ và cần thông tin DB thì n collect DB dựa vào lib connector), và n xử lý trả lại theo nội dung HTML cho web server

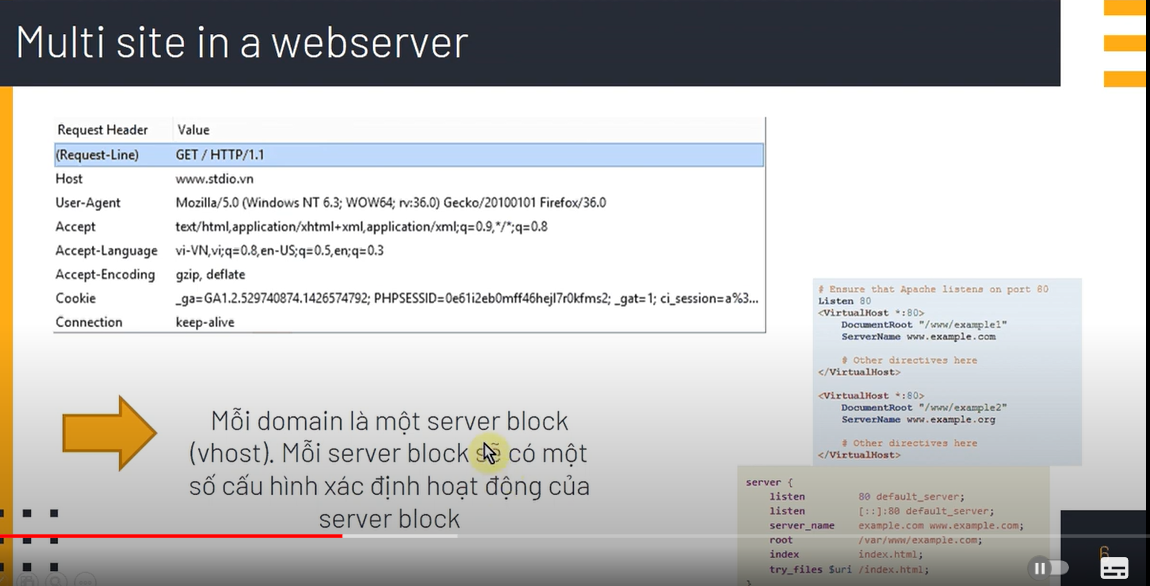
Và việc giao tiếp giữa web server và trình thông dịch PHP sẽ có chuẩn giao tiếp CGI hoặc uWSGI

Giữa web server và trình thông dịch là 2 tiến trình độc lập khác nhau chạy trên 1 con server

Lên việc giao tiếp và lên tiến trình sẽ có 1 số kiểu giao tiếp theo TCP socket, hoặc file socket theo 1 protocol chuẩn là CGI/uWSGI



Ví dụ trong cấu hình nginx, web server n nhìn thấy file php này server nginx n ko xử lý mà forward cho thằng php xử lý dựa vào cái **unix:/run/php/php7.0-fpm.sock (**qua file socket unix**)**

****

****

đây là 1 server block

## proxy pass

config 1 site

  server {

        listen 80 ;

        listen [::]:80 ;

        root /var/www/vietdung/html;

        index index.html index.html;

        server\_name vietdung.com;

        location / {

            # alias /usr/share/nginx/html;

            try\_files $uri $uri/ =404;

        }

    }

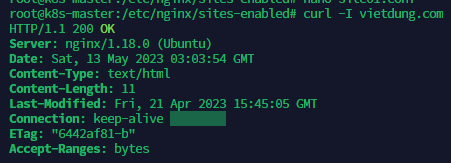
server {

    server\_name www.vietdung.com;

    return 301 $scheme://vietdung.com$request\_uri;

}

## Header



Đây là header mặc định của các website, respone header ta có thể thêm các header vào

    server {

        listen 80 ;

        listen [::]:80 ;

        root /var/www/vietdung/html;

        index index.html index.html;

        server\_name vietdung.com;

        add\_header Cache-Control no-store;     # add header ;

# Cài nginx

Cách 1: apt install nginx

Cách này thì sẽ thiếu 1 số module

Cách 2

Tải source code về để build trên máy => sử dụng make để build

Configuration => tạo ra makefile

Make => build ra binary

Make install để cài

Make uninstall để gỡ

B1: cài 1 số thư viện

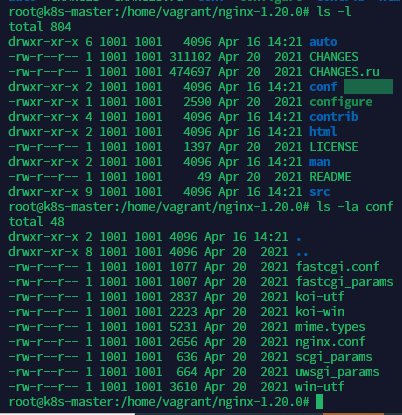
apt update

sudo apt-get install build-essential libpcre3 libpcre3-dev zlib1g zlib1g-dev libssl-dev libgd-dev libxml2 libxml2-dev uuid-dev

B2: download nginx : nginx.org

wget http://nginx.org/download/nginx-1.20.0.tar.gz

tar -zxvf nginx-1.20.0.tar.gz



B2: build vs module

Trong folder có file module

./configure   \

    --prefix=/var/www/html  \

    --sbin-path=/usr/sbin/nginx  \

    --conf-path=/etc/nginx/nginx.conf  \

    --http-log-path=/var/log/nginx/access.log  \

    --error-log-path=/var/log/nginx/error.log  \

    --with-pcre   \

    --lock-path=/var/lock/nginx.lock  \

    --pid-path=/var/run/nginx.pid  \

    --with-http\_ssl\_module  \

    --with-http\_image\_filter\_module=dynamic  \

    --modules-path=/etc/nginx/modules  \

    --with-http\_v2\_module  \

    --with-stream=dynamic  \

    --with-http\_addition\_module  \

    --with-http\_mp4\_module

--with là install module

Install cho bản 1.16

wget http://nginx.org/download/nginx-1.16.1.tar.gz

./configure \

    --prefix=/usr/share/nginx \

    --sbin-path=/usr/sbin/nginx \

    --conf-path=/etc/nginx/nginx.conf \

    --pid-path=/var/run/nginx.pid \

    --lock-path=/var/lock/nginx.lock \

    --error-log-path=/var/log/nginx/error.log \

    --http-log-path=/var/log/nginx/access.log \

    --user=www-data \

    --group=www-data \

    --without-mail\_pop3\_module \

    --without-mail\_imap\_module \

    --without-mail\_smtp\_module \

    --without-http\_fastcgi\_module \

    --without-http\_uwsgi\_module \

    --without-http\_scgi\_module \

    --without-http\_memcached\_module \

    --with-http\_ssl\_module \

    --with-http\_stub\_status\_module \

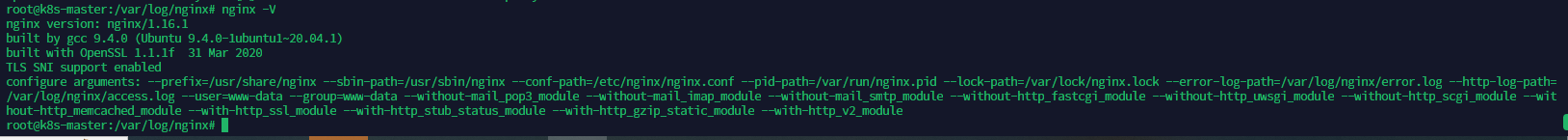
    --with-http\_gzip\_static\_module \

    --with-http\_v2\_module

make

make install

nginx –V



nano /lib/systemd/system/nginx.service

[Unit]

Description=The NGINX HTTP and reverse proxy server

After=syslog.target network-online.target remote-fs.target nss-lookup.target

Wants=network-online.target

[Service]

Type=forking

PIDFile=/var/run/nginx.pid

ExecStartPre=/usr/sbin/nginx -t

ExecStart=/usr/sbin/nginx

ExecReload=/usr/sbin/nginx -s reload

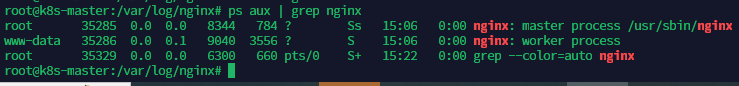
ExecStop=/bin/kill -s QUIT $MAINPID

PrivateTmp=true

[Install]

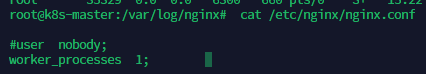
WantedBy=multi-user.target

Nginx đang chạy vs user www-data



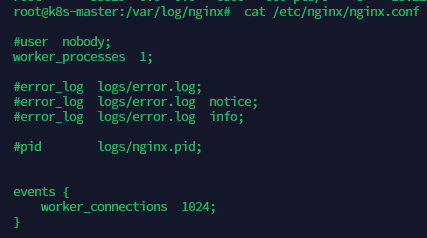
Nhưng ta đang có 1 user root chạy tiến trình của nginx (là n để lấy port 80 của hệ thống)

Còn các tiến trình con chạy user www-data thôi, để bảo mật vì chúng n có thể phi vào server vs user nginx nhưng ko vào server vs user root được, vì user nginx ko su được lên root



Worker\_processes auto; thì worker\_processes = số core của máy

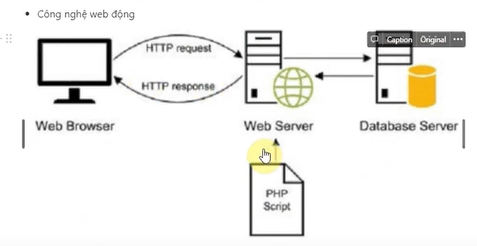
Nên lệnh ps aux chỉ show được 1 tiến trình worker process nginx



Worker\_connections = 1024 \* 1 = 1024 – phục vụ 1024 connections

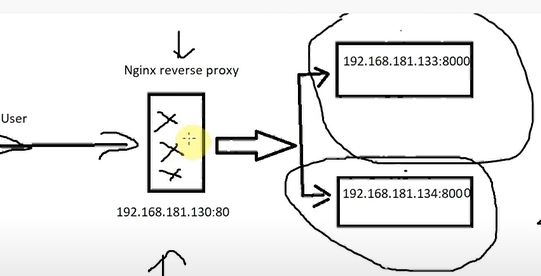
# Cấu hình nginx vs PHP-fpm

Công nghệ web động



Giao tiếp giữa web server vs script qua 1 interface file socket để phục vụ request ng dùng

## Reverse proxy



Cấu hình reverse proxy => option chỉ thị forward cho nginx là cấu hình proxy\_pass

server {

        listen 80 ;

        listen [::]:80 ;

        root /var/www/vietdung/html;

        index index.html index.html;

        server\_name vietdung.com;

        add\_header Cache-Control no-store;

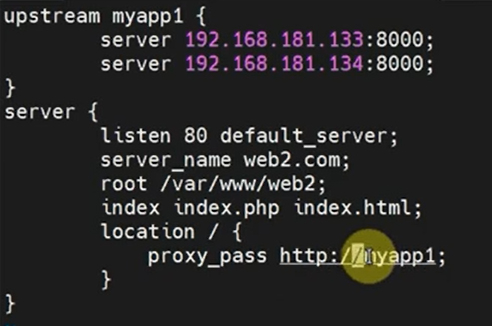
        location / {

            proxy\_pass http://192.168.50.11:8000

        }

    }

Để proxy pass tới 2 backend server thì phải tạo ra 1 **upstream**



## Cookie

Cookie để giữ session request respone của client vs server

Cookie cũng có thể có thiết lập tgian expire, ví dụ như vào app banking, nếu như ko tạo request j nữa, để 1 tgian quay lại thì bị out session, => reload lại để login lại = thiết lập lại cookie mới

* Cookie để giữ 1 session giúp tương tác giữa client và server để client, server request và respone

Nếu 2 thằng client mà có token giống nhau thì server lại chả respone về cho 2 thằng