server {

listen 80;

server\_name ec2-54-160-202-168.compute-1.amazonaws.com www.ec2-54-160-202-168.compute-1.amazonaws.com;

root /var/www/ec2-54-160-202-168.compute-1.amazonaws.com;

index index.html index.htm index.php;

location / {

try\_files $uri $uri/ =404;

}

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass **unix:**/var/run/php/php7.4-fpm.sock;

}

location ~ /\.ht {

deny all;

}

}

server{

#SSL configurations

listen 443 ssl http2 default\_server;

listen [::]:443 ssl http2 default\_server;

include snippets/self-signed.conf;

include snippets/ssl-params.conf;

}

##

# You should look at the following URL's in order to grasp a solid understanding

# of Nginx configuration files in order to fully unleash the power of Nginx.

# https://www.nginx.com/resources/wiki/start/

# https://www.nginx.com/resources/wiki/start/topics/tutorials/config\_pitfalls/

# https://wiki.debian.org/Nginx/DirectoryStructure

#

# In most cases, administrators will remove this file from sites-enabled/ and

# leave it as reference inside of sites-available where it will continue to be

# updated by the nginx packaging team.

#

# This file will automatically load configuration files provided by other

# applications, such as Drupal or Wordpress. These applications will be made

# available underneath a path with that package name, such as /drupal8.

#

# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.

##

# Default server configuration

#

server {

listen 80 default\_server;

listen [::]:80 default\_server;

# SSL configuration

#

# listen 443 ssl default\_server;

# listen [::]:443 ssl default\_server;

#

# Note: You should disable gzip for SSL traffic.

# See: https://bugs.debian.org/773332

#

# Read up on ssl\_ciphers to ensure a secure configuration.

# See: https://bugs.debian.org/765782

#

# Self signed certs generated by the ssl-cert package

# Don't use them in a production server!

#

# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP

index index.php;

server\_name ec2-54-160-202-168.compute-1.amazonaws.com;

return 302 https://$server\_name$request\_uri;

# include snippets/phpmyadmin.conf;

location / {

# First attempt to serve request as file, then

# as directory, then fall back to displaying a 404.

try\_files $uri $uri/ =404;

}

# pass PHP scripts to FastCGI server

#

location ~ \.php$ {

include snippets/fastcgi-php.conf;

#

# # With php-fpm (or other unix sockets):

fastcgi\_pass unix:/var/run/php/php7.4-fpm.sock;

# # With php-cgi (or other tcp sockets):

# fastcgi\_pass 127.0.0.1:9000;

}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

location ~ /\.ht {

deny all;

}

}

server{

#SSL configurations

listen 443 ssl http2 default\_server;

listen [::]:443 ssl http2 default\_server;

include snippets/self-signed.conf;

include snippets/ssl-params.conf;

# include snippets/phpmyadmin.conf;

}

# Virtual Host configuration for example.com

#

# You can move that to a different file under sites-available/ and symlink that

# to sites-enabled/ to enable it.

#

#server {

# listen 80;

# listen [::]:80;

#

# server\_name example.com;

#

# root /var/www/example.com;

# index index.html;

#

# location / {

# try\_files $uri $uri/ =404;

# }

#}

~