Remover Docker Engine antigo

sudo apt-get purge -y docker-engine docker docker.io docker-ce docker-ce-cli
sudo apt-get autoremove -y --purge docker-engine docker docker.io docker-ce

sudo rm -rf /var/lib/docker /etc/docker
sudo rm /etc/apparmor.d/docker
sudo groupdel docker
sudo rm -rf /var/run/docker.sock

Instalar Docker Engine

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

sudo apt install docker-ce

/etc/init.d/docker start

/etc/init.d/docker status

Install Docker Compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.28.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

O que é o Docker?
É uma tecnologia de virtualização de servidores que permite a criação de ambientes padrões para aplicações.

Container: É o local onde a aplicação ficará rodando.
Imagem: É um snapshot do container com seus recursos.

Dockerfile: Arquivo que possui instruções de tipo de S0 e ferramentas que a nossa máquina virtual possuirá.

Docker Compose: É uma ferramenta do Docker que permite subir containers com suas especificações e conectados através de redes internas.

services: Imagens onde programas serão executados.
ports: Vamos apontar a porta local (primeira) para a porta do servidor no docker (segunda).
volumes: Com essa configuração nos dizemos ao Docker para copiar todos os arquivos do nosso diretório local para dentro do diretório do servidor no container.
volumes mysql: Salva os dados do banco para não perdemos eles quando deletarmos o container.
ln: Vamos criar um link simbólico apontando todo o nosso projeto para a pasta html no container.
network: Criamos uma rede que os nossos serviços irão compartilhar, e declaramos o uso dessa rede em cada declaração de serviço no arquivo docker-compose.yaml.

Dockerfile:

```
FROM wyveo/nginx-php-fpm:latest
WORKDIR /usr/share/nginx/
```

docker-compose.yaml:

```
version: '3'

services:
   laravel-app:
       build: .
       ports:
           - "8080:80"
       volumes:
           - ./:/usr/share/nginx
       networks:
           - app-network

   mysql-app:
       image: mysql:5.7.22
       ports:
           - "3306:3306"
```

```
        volumes:
            - .docker/dbdata:/var/lib/mysql
        environment:
            MYSQL_DATABASE: laravel
            MYSQL_ROOT_PASSWORD: laravel
        networks:
            - app-network

networks:
   app-network:
        driver: bridge
```

Access Laravel routes:
RUN: sudo ln -s public html
RUN: sudo docker-compose up -d --build
RUN: sudo docker exec -it newblog_laravel-app_1 bash
RUN: nano /etc/nginx/conf.d/default.conf
RUN: Ctrl + O and Ctrl + X
Edit location:

```
root@2209d85c3f65:/usr/share/nginx# cat /etc/nginx/conf.d/default.conf
server {
    listen   80; ## listen for ipv4; this line is default and implied
    listen   [::]:80 default ipv6only=on; ## listen for ipv6

    root /usr/share/nginx/html;
    index index.php index.html index.htm;

    # Make site accessible from http://localhost/
    server_name _;

    # Disable sendfile as per https://docs.vagrantup.com/v2/synced-folders/virtualbox.html
    sendfile off;

    # Security - Hide nginx version number in error pages and Server header
    server_tokens off;

    # Add stdout logging
    error_log /dev/stdout info;
    access_log /dev/stdout;

    # reduce the data that needs to be sent over network
    gzip on;
    gzip_min_length 10240;
    gzip_proxied expired no-cache no-store private auth;
    gzip_types text/plain text/css text/xml application/json text/javascript application/x-javascript application/xml;
    gzip_disable "MSIE [1-6]\.";

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to index.php
        try_files $uri $uri/ /index.php;
    }

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
```

Change to enable Laravel routes

RUN: service nginx restart

Flowchart: