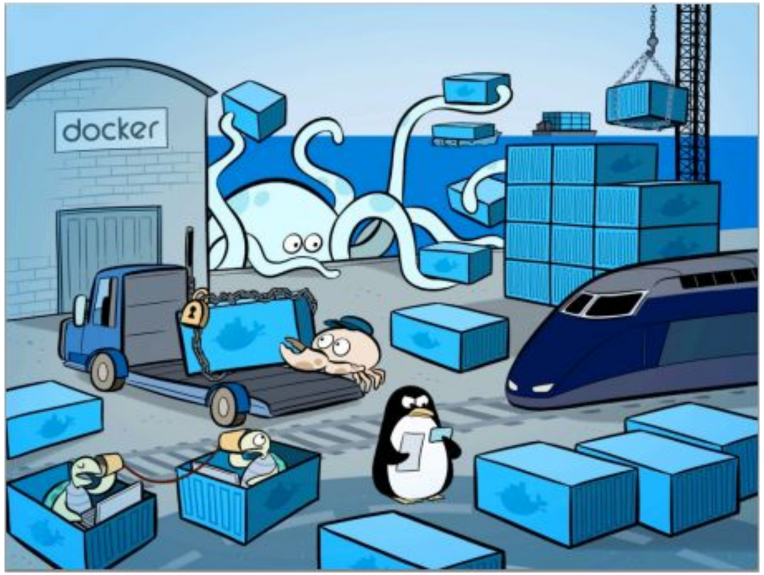
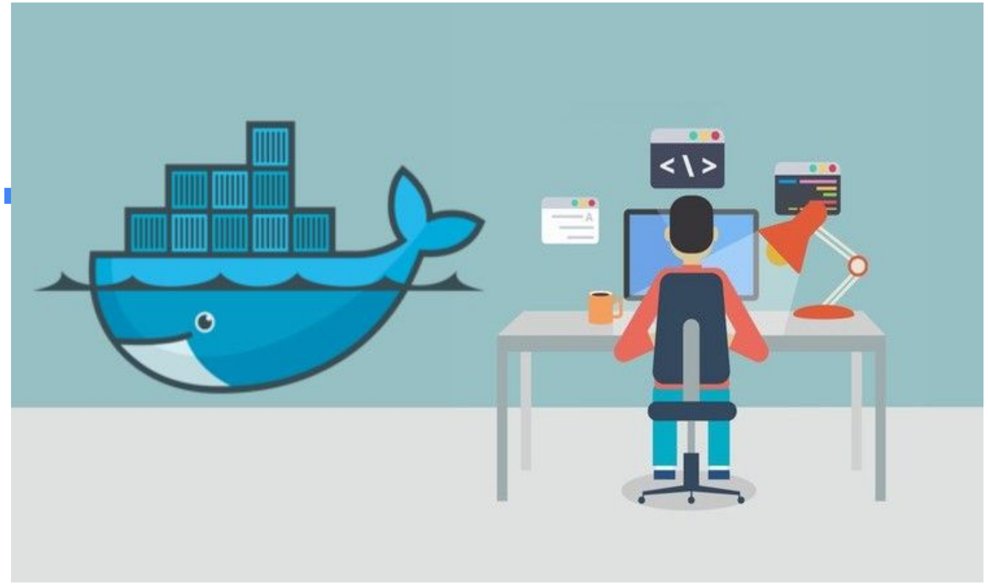
Docker





Quem somos?

- **Diego Bulhões**
- Pet Sistemas
- Interesses
  - Docker
  - Desenvolvimento web
  - infraestrutura
  - Nodejs e JS

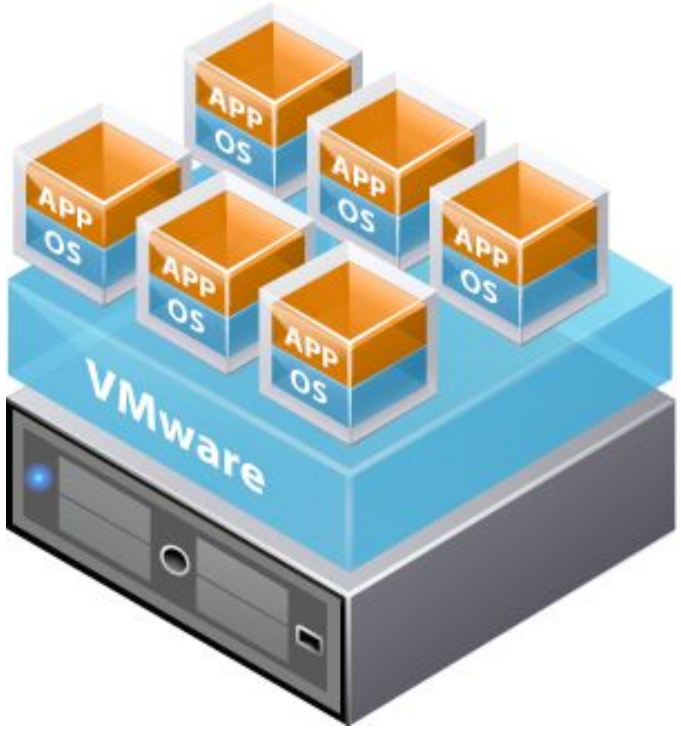
- **Vinicius Espindola**
- Pet Sistemas
- Técnico de Informática (IFMS)
- Interesses
  - Docker e Kubernetes
  - Desenvolvimento web
  - infraestrutura
  - Cloud Computing

- **Hernanes**
- CTEI
- TIMESAT
- Shell Script
- C
- Python

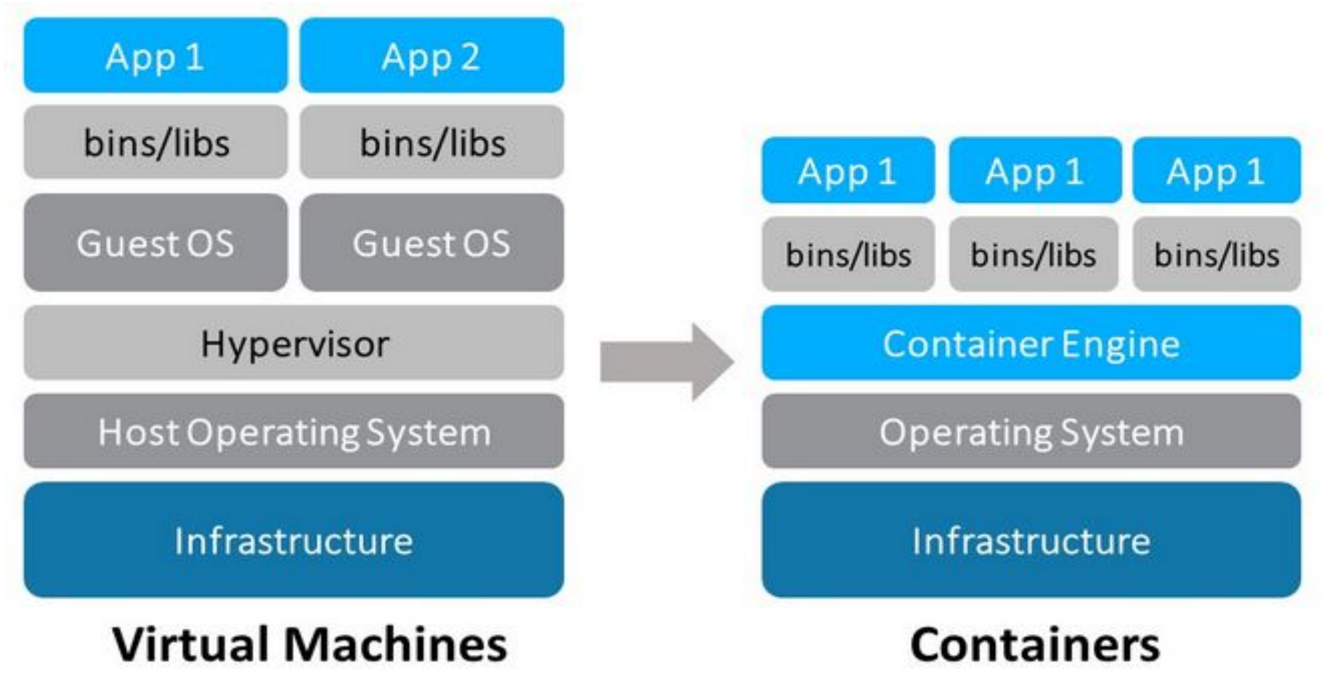
Como vai ser?

- Vai ser divido em 1 dia 4 horas
- Conteúdo
  - Dockerfile
  - Docker-compose
  - Swarm

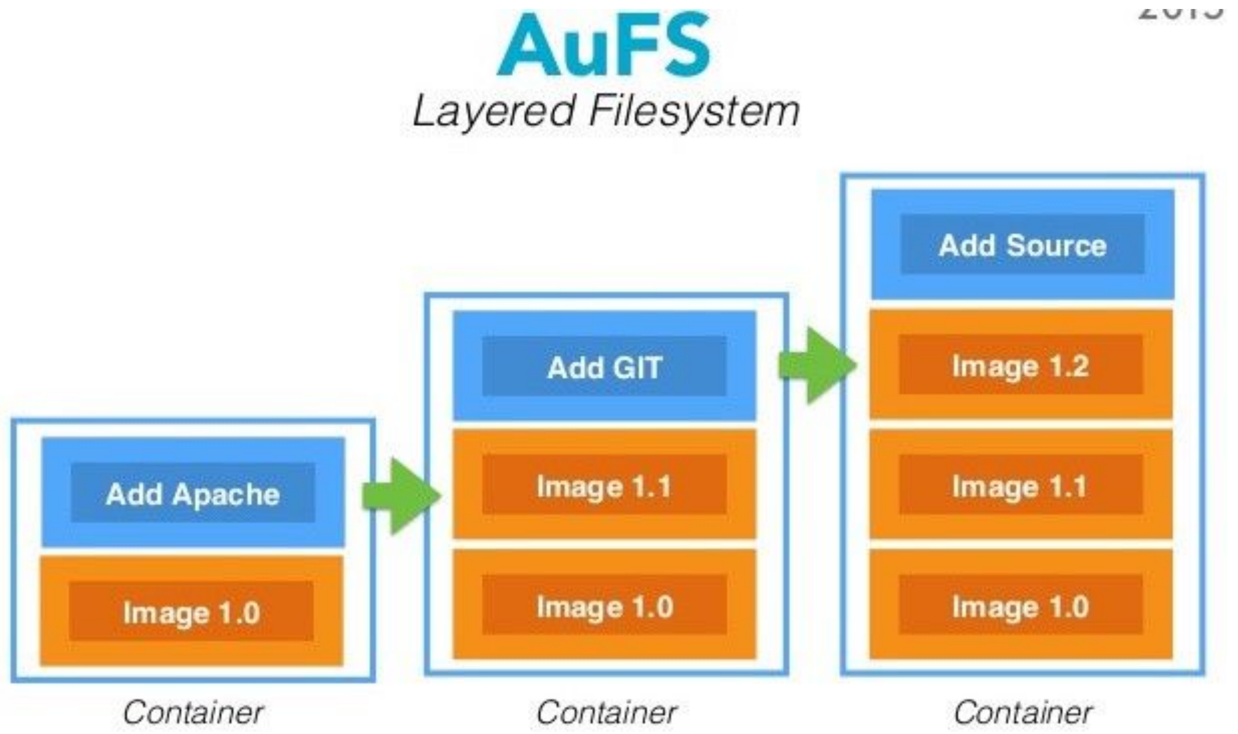
Virtualização de Servidores

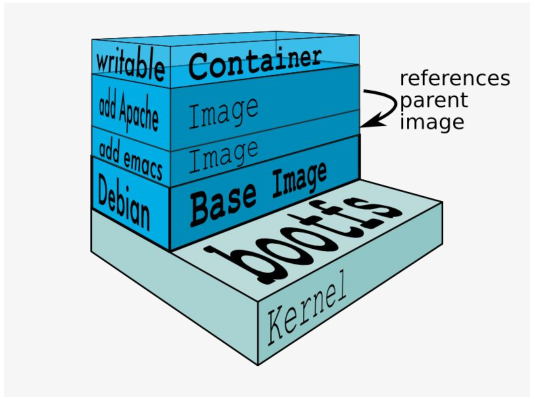


Docker x Virtual Machine



Imagens


2015
AuFS
Layered Filesystem
Add Apache
Image 1.0
Container
Add GIT
Image 1.1
Image 1.0
Container
Add Source
Image 1.2
Image 1.1
Image 1.0
Container


writable
Container
add Apache
Image
add emacs
Image
Debian
Base Image
bootfs
Kernel
references parent image

Docker Windows e Docker Toolbox

não é aconselhável o uso do docker toolbox

baixar o executável e next next …

https://docs.docker.com/toolbox/toolbox_install_windows/
https://docs.docker.com/docker-for-windows/install/

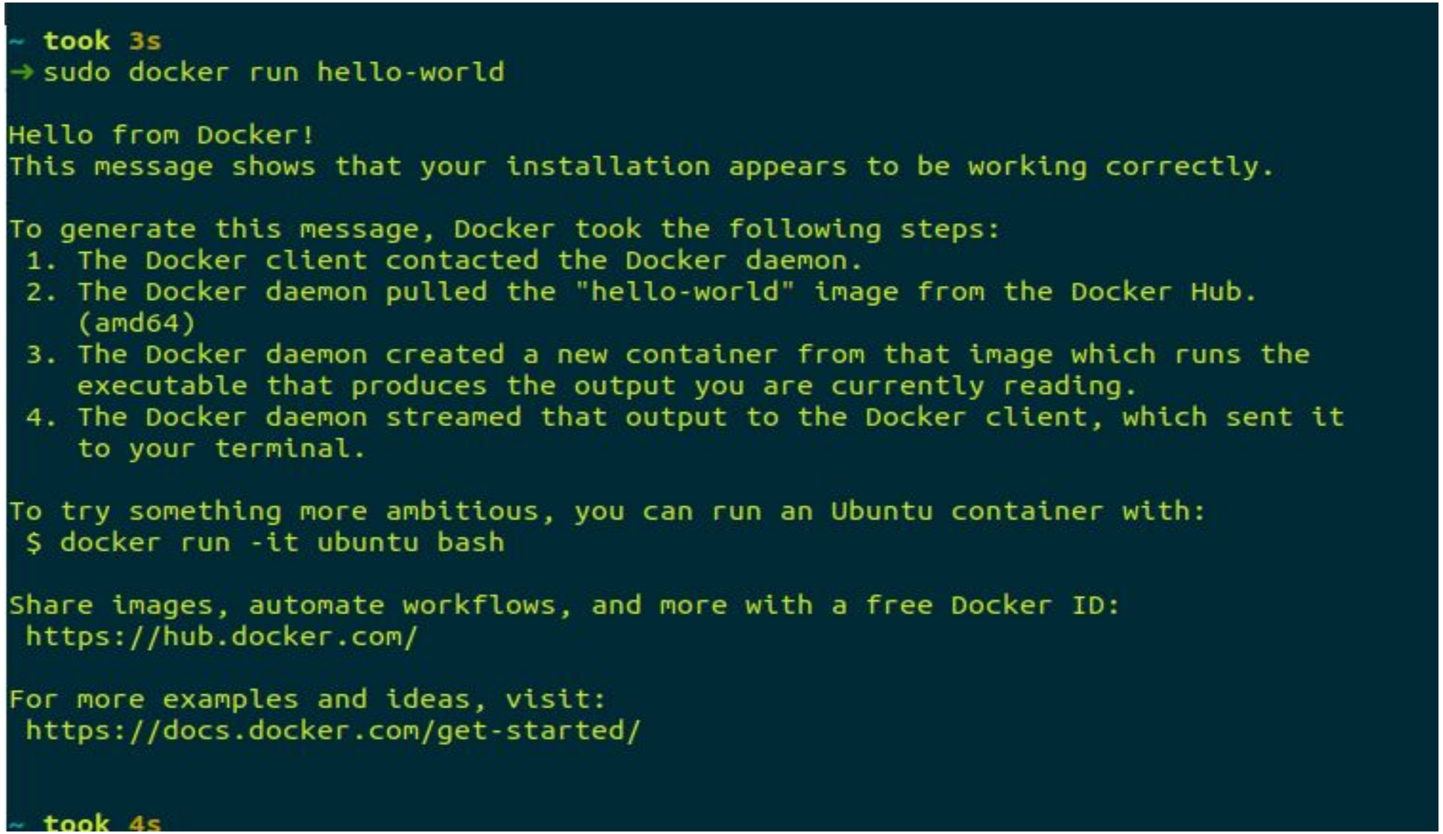
Docker Linux

- $ curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
- $ curl -fsSL https://get.docker.com -o get-docker.sh | sh

https://docs.docker.com/install/linux/docker-ce/ubuntu/

EXECUTANDO HELLO WORLD

docker run hello-world



```
~ took 3s
→ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/


~ took 4s
```

PROCESSO PRINCIPAL DE UM CONTAINER

Exercicio - VOLUME

- Para a atividade de volume iremos instalar o apache em um container e realizar um link de volume, configurar a porta e rede

Executando containers

```
docker exec -it IDContainer
```

VOLUMES E MAIS ALGUNS COMANDOS

* todos os argumentos mostrados anteriormente são válidos para volumes

- docker run -v (caminho do host):(caminho dentro do container)

Exercício

- Subir um container com Ubuntu e um link de diretórios

DOCKER COMANDOS

docker run ARGUMENT ARGUMENT...

SEMPRE EXECUTAR COMO SUDO

docker pull ARGUMENT

docker container inspect NameContainer

docker COMMAND ls

ex: docker container ls -a

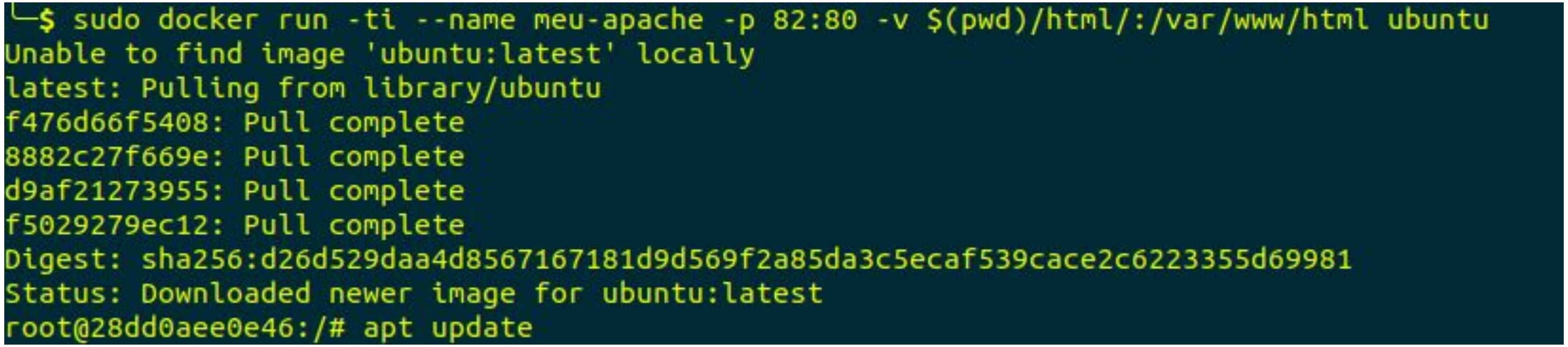
docker network ls

docker volume ls

https://docs.docker.com/

Manual de Instalação

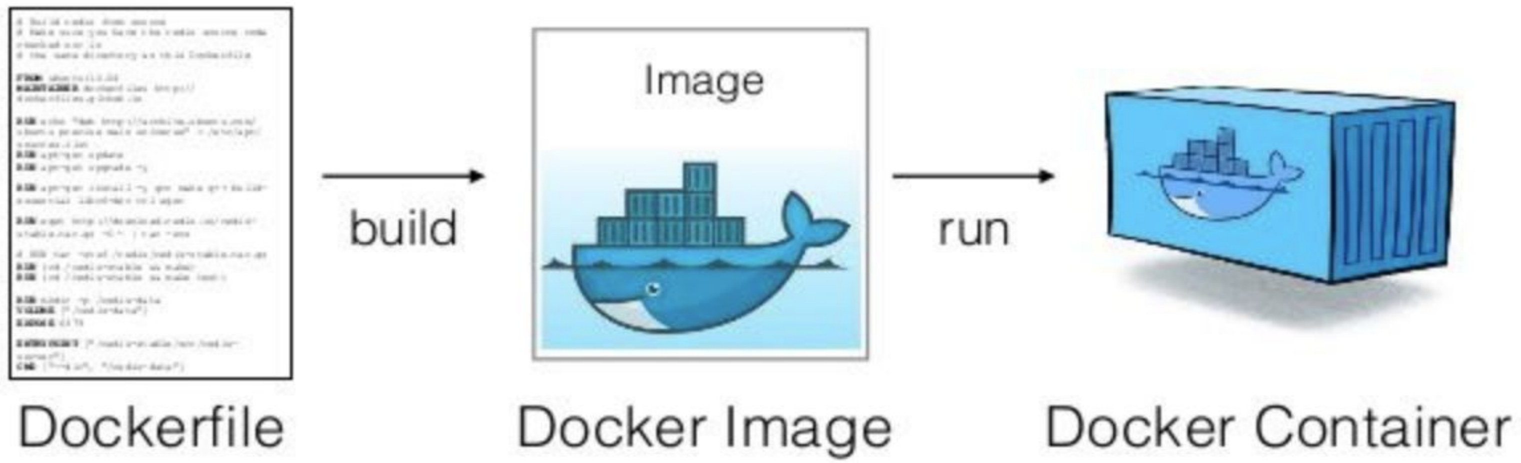
1. docker run -ti --name meu-apache -p 8080:80 -v $(pwd)/html/:/var/www/html ubuntu
2. apt update && apt upgrade && apt install apache2
3. service apache2 start
4. http://ipDoContainer:8080



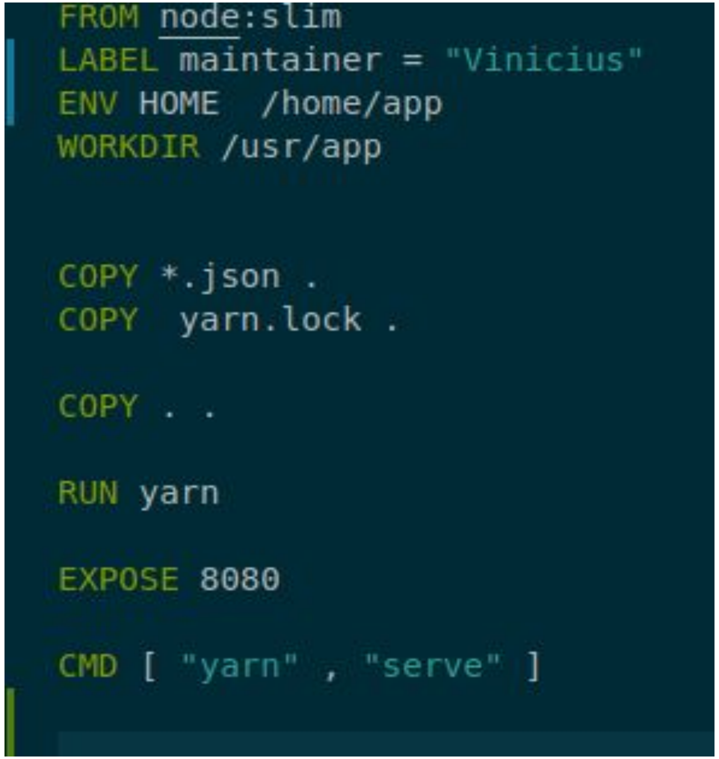
```
└─$ sudo docker run -ti --name meu-apache -p 82:80 -v $(pwd)/html/:/var/www/html ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
f476d66f5408: Pull complete
8882c27f669e: Pull complete
d9af21273955: Pull complete
f5029279ec12: Pull complete
Digest: sha256:d26d529daa4d8567167181d9d569f2a85da3c5ecaf539cace2c6223355d69981
Status: Downloaded newer image for ubuntu:latest
root@28dd0aee0e46:/# apt update
```

Dockerfile

```
docker build -t    userDockerhub/nameImage   pathDockerfile
```

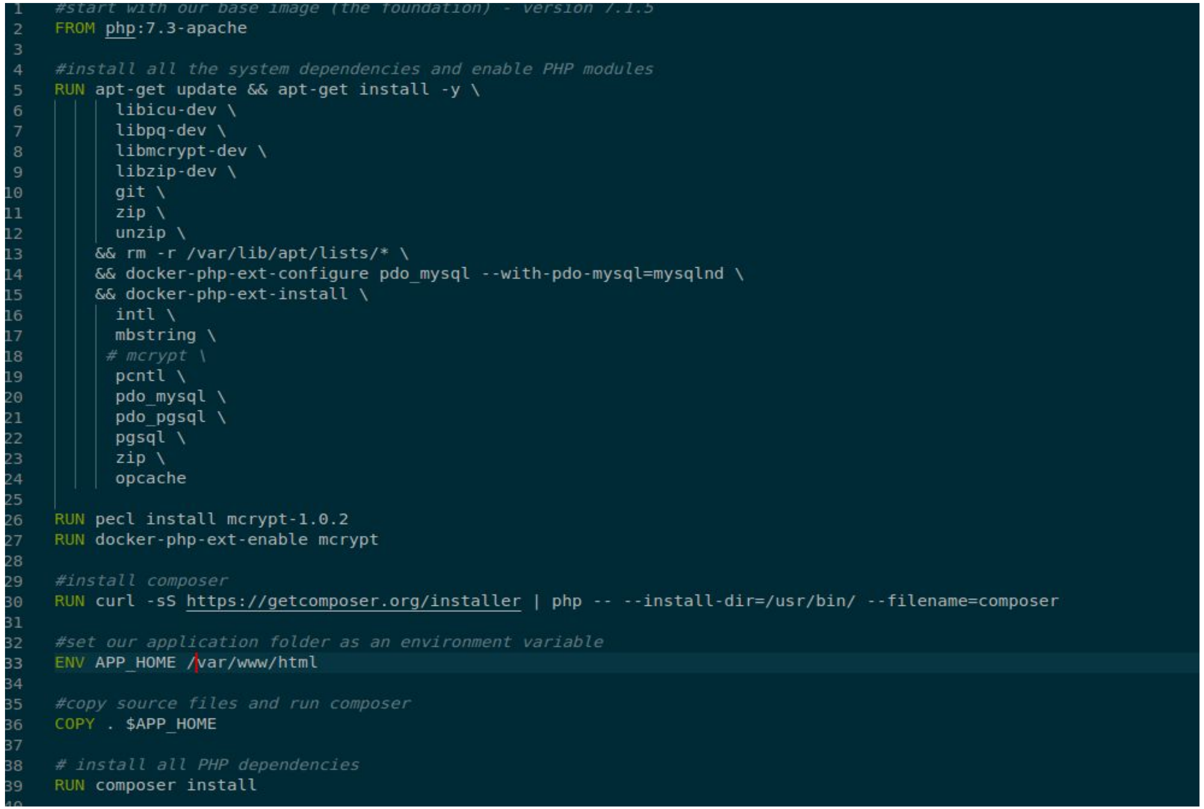


Exemplo Dockerfile NodeJS



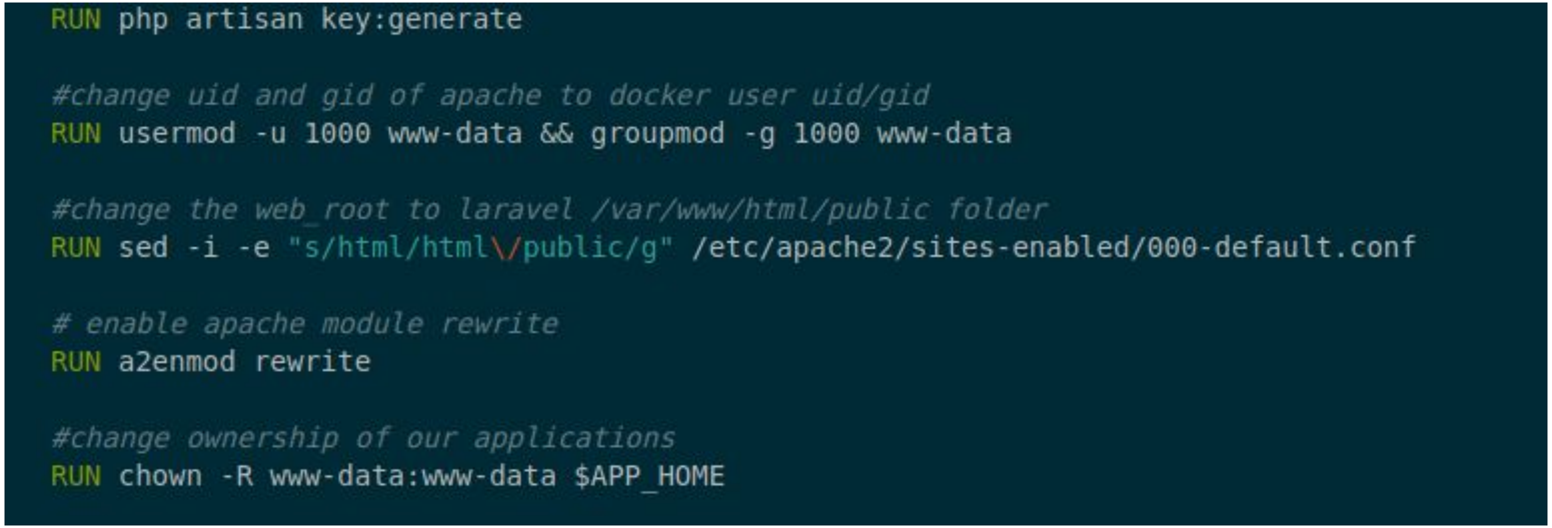
```
FROM node:slim
LABEL maintainer = "Vinicius"
ENV HOME  /home/app
WORKDIR /usr/app


COPY *.json .
COPY  yarn.lock .

COPY . .

RUN yarn

EXPOSE 8080

CMD [ "yarn" , "serve" ]
```

Exemplo Dockerfile PHP - Framework Laravel



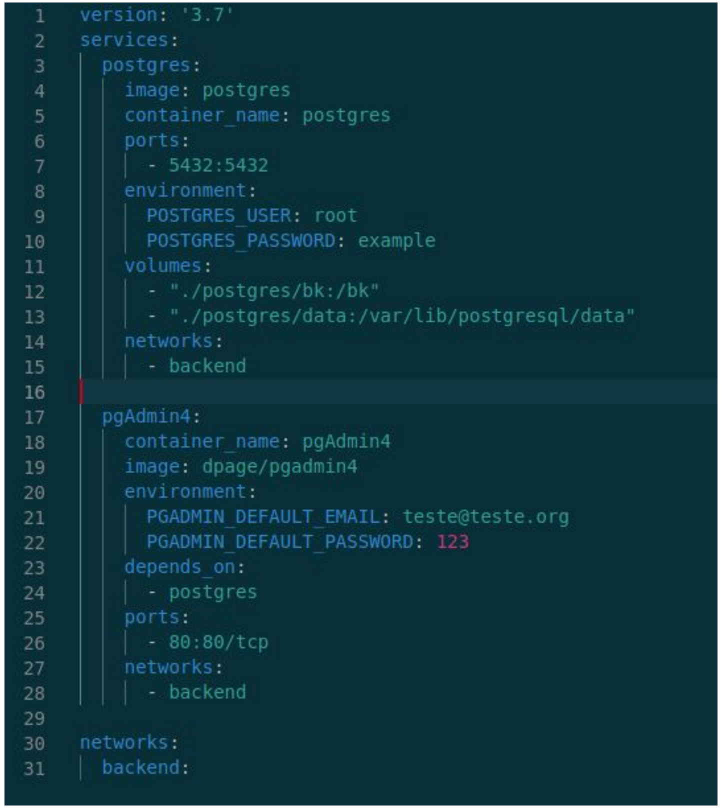
```
#start with our base image (the foundation) - version 7.1.5
FROM php:7.3-apache

#install all the system dependencies and enable PHP modules
RUN apt-get update && apt-get install -y \
      libicu-dev \
      libpq-dev \
      libmcrypt-dev \
      libzip-dev \
      git \
      zip \
      unzip \
    && rm -r /var/lib/apt/lists/* \
    && docker-php-ext-configure pdo_mysql --with-pdo-mysql=mysqlnd \
    && docker-php-ext-install \
      intl \
      mbstring \
     # mcrypt \
      pcntl \
      pdo_mysql \
      pdo_pgsql \
      pgsql \
      zip \
      opcache

RUN pecl install mcrypt-1.0.2
RUN docker-php-ext-enable mcrypt

#install composer
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/bin/ --filename=composer

#set our application folder as an environment variable
ENV APP_HOME /var/www/html

#copy source files and run composer
COPY . $APP_HOME

# install all PHP dependencies
RUN composer install
```

Exemplo Dockerfile PHP - Framework Laravel



```
RUN php artisan key:generate

#change uid and gid of apache to docker user uid/gid
RUN usermod -u 1000 www-data && groupmod -g 1000 www-data

#change the web_root to laravel /var/www/html/public folder
RUN sed -i -e "s/html/html\/public/g" /etc/apache2/sites-enabled/000-default.conf

# enable apache module rewrite
RUN a2enmod rewrite

#change ownership of our applications
RUN chown -R www-data:www-data $APP_HOME
```

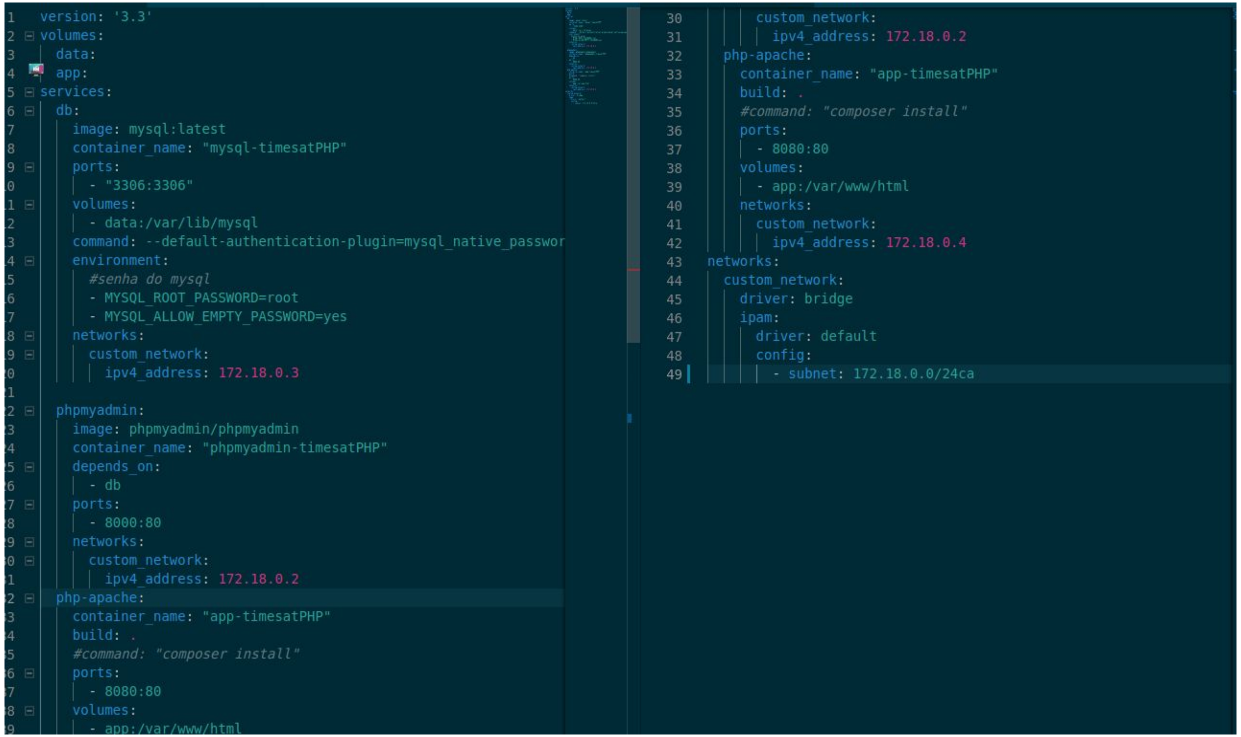
PGAdmin4 e PostgresSQL



```yaml
version: '3.7'
services:
  postgres:
    image: postgres
    container_name: postgres
    ports:
      - 5432:5432
    environment:
      POSTGRES_USER: root
      POSTGRES_PASSWORD: example
    volumes:
      - "./postgres/bk:/bk"
      - "./postgres/data:/var/lib/postgresql/data"
    networks:
      - backend

  pgAdmin4:
    container_name: pgAdmin4
    image: dpage/pgadmin4
    environment:
      PGADMIN_DEFAULT_EMAIL: teste@teste.org
      PGADMIN_DEFAULT_PASSWORD: 123
    depends_on:
      - postgres
    ports:
      - 80:80/tcp
    networks:
      - backend

networks:
  backend:
```

Docker-Compose

Orquestrar containers via arquivo

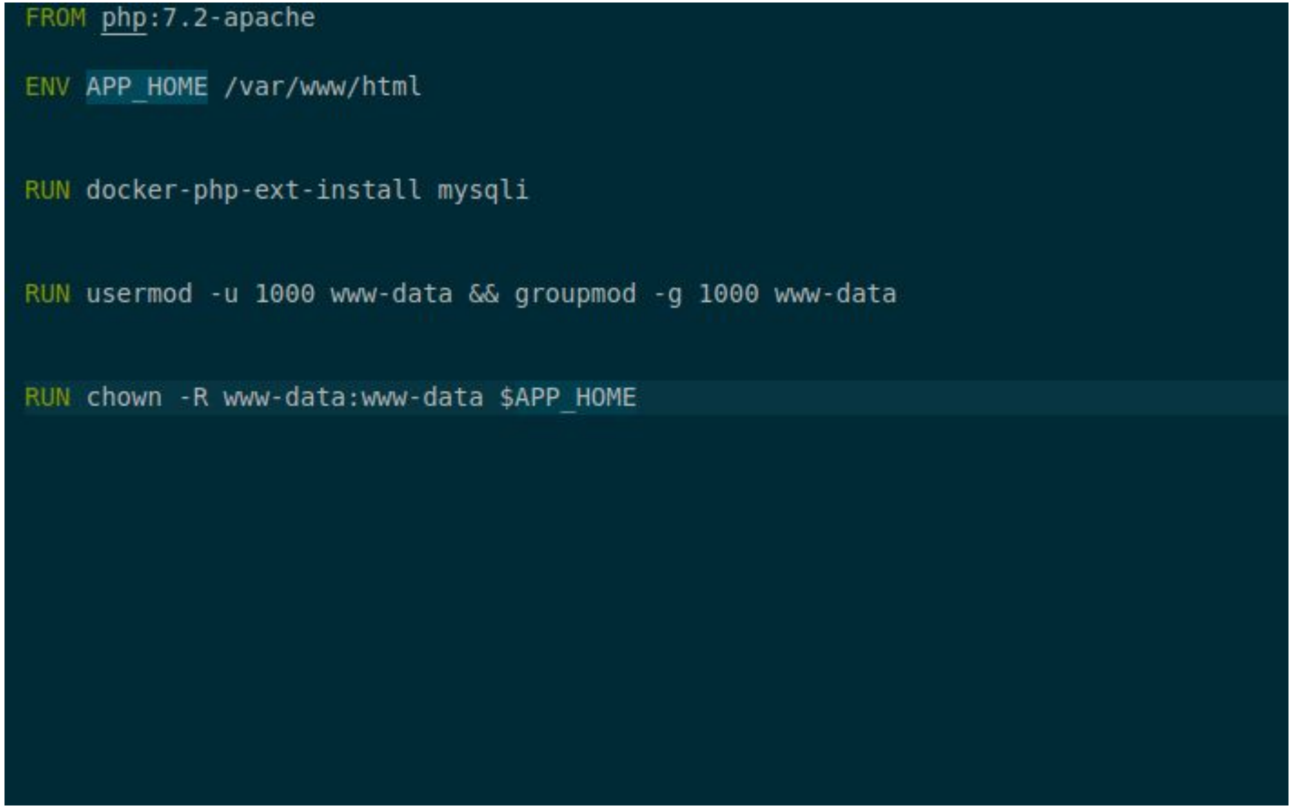


```yaml
version: '3.3'
volumes:
  data:
  app:
services:
  db:
    image: mysql:latest
    container_name: "mysql-timesatPHP"
    ports:
      - "3306:3306"
    volumes:
      - data:/var/lib/mysql
    command: --default-authentication-plugin=mysql_native_passwor
    environment:
      #senha do mysql
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_ALLOW_EMPTY_PASSWORD=yes
    networks:
      custom_network:
        ipv4_address: 172.18.0.3

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    container_name: "phpmyadmin-timesatPHP"
    depends_on:
      - db
    ports:
      - 8000:80
    networks:
      custom_network:
        ipv4_address: 172.18.0.2
  php-apache:
    container_name: "app-timesatPHP"
    build: .
    #command: "composer install"
    ports:
      - 8080:80
    volumes:
      - app:/var/www/html
```

```yaml
      custom_network:
        ipv4_address: 172.18.0.2
  php-apache:
    container_name: "app-timesatPHP"
    build: .
    #command: "composer install"
    ports:
      - 8080:80
    volumes:
      - app:/var/www/html
```

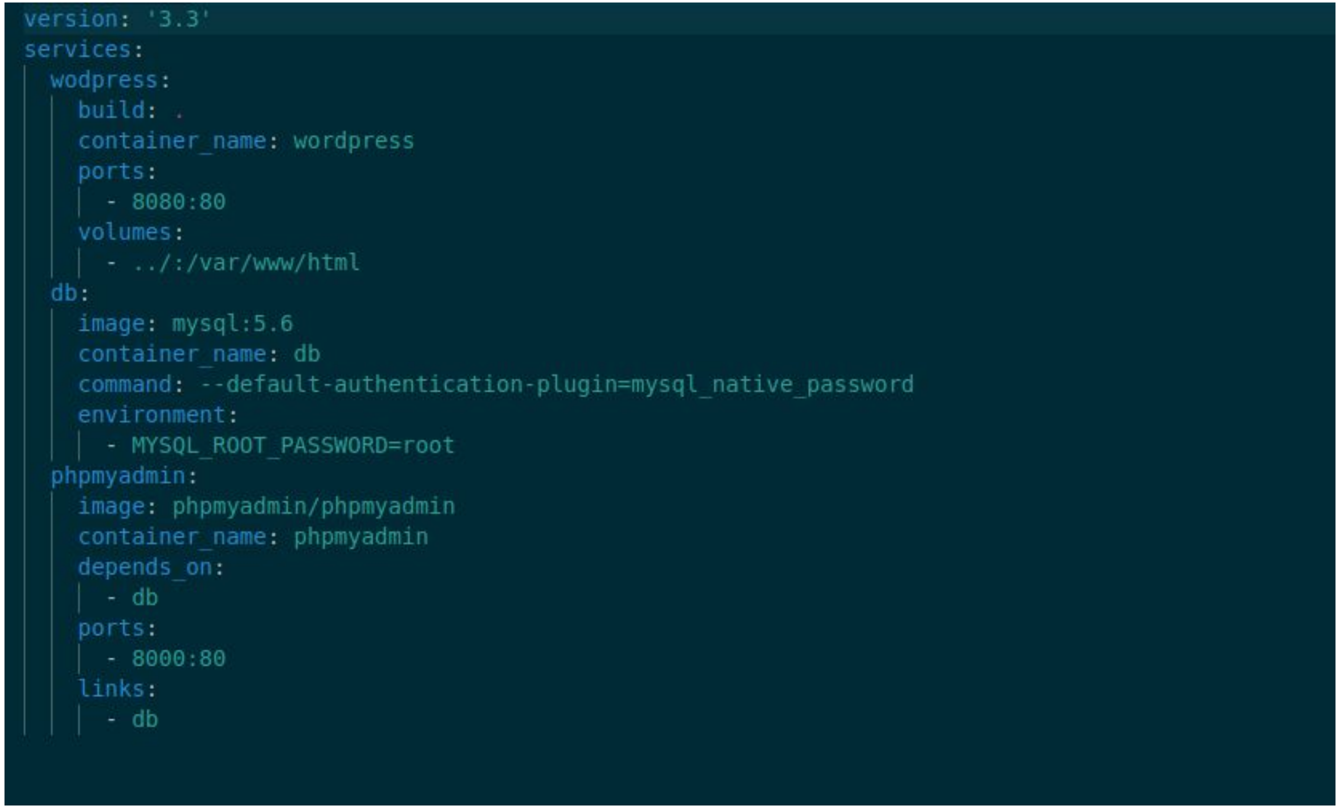
```yaml
    networks:
      custom_network:
        ipv4_address: 172.18.0.4
networks:
  custom_network:
    driver: bridge
    ipam:
      driver: default
      config:
        - subnet: 172.18.0.0/24ca
```

WORDPRESS



```
FROM php:7.2-apache

ENV APP_HOME /var/www/html


RUN docker-php-ext-install mysqli


RUN usermod -u 1000 www-data && groupmod -g 1000 www-data


RUN chown -R www-data:www-data $APP_HOME
```

WORDPRESS



```yaml
version: '3.3'
services:
  wodpress:
    build: .
    container_name: wordpress
    ports:
      - 8080:80
    volumes:
      - ../:/var/www/html
  db:
    image: mysql:5.6
    container_name: db
    command: --default-authentication-plugin=mysql_native_password
    environment:
      - MYSQL_ROOT_PASSWORD=root
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin
    depends_on:
      - db
    ports:
      - 8000:80
    links:
      - db
```