

# Docker Alura - Início

**docker images:** lista toda as imagens, como seus nomes criados aleatoriamente ou não,

**docker run:** cria os containers,

**docker exec :** serve para executar alguma coisa no container,

**docker kill [nome da maquina] :** matando um processo travado,

**docker start [nome do container] :** revivendo containers já criados,

**docker inspect [nome do container] :** quando a máquina estiver UP podemos ver o que tem na máquina, inclusive o IP,

**docker pull:** faz o download de imagens,

**docker run -it -v \$(pwd):/tmp [nome do container] bash :** o **-v** é de volume e o **\$(pwd)** pega o caminho local e cria a pasta **tmp**,

**docker ps:** acompanha os containers em execução,

**docker run --link mysql:db -it ubuntu bash:** o uso do **--link** serve para não fixar o IP, assim criaríamos um **ALIAS db**,

**docker rm \$(docker ps -qa):** removendo todos os containers,

**docker rmi [nome da imagem]:** removendo todas as imagens,

**docker run --rm -it ubuntu bash:** instanciando um container que não será persistido,

**docker commit -m "Instalação do Apache" [nome ou id do container] [imagem]/apache:** está salvando um FORK da máquina UBUNTU com o APACHE,

**docker run -it -p 8080:80 ubuntu/apache bash:** a porta 8080 é da máquina física e o 80 é do container,

**docker exec -it [nome do container] bash:** nós deixa entrar na linha de comando do container.

## Imagem x Máquina(Container)

Quando você está usando o Docker você precisa de uma imagem, isso quer dizer que a imagem poderá ser reutilizada várias vezes, já a máquina não sempre será criar uma nova caso você não chame a antiga pelo NAME.

**docker images:** lista toda as imagens, como seus nomes criados aleatoriamente ou não.

**docker ps -a:** lista toda as máquinas criadas já.

## **.Dockerfile**

**<!-- inicio -->**

FROM ubuntu //imagem base

RUN apt-get update && apt-get install -y apache2 //executa a instrução de instalação do Apache

EXPOSE 80

ADD app/ /var/www/html

COPY app/ /var/www/html/app

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

**<!-- fim -->**

**docker build -t ubuntu/apache:** compilando e gerando a imagem,

**docker run -d -p 80:80 ubuntu/apache:** rodando em primeiro plano com o FOREGROUND.

## **docker-compose.yml**

**<!-- inicio -->**

db:

image: mysql

volumes:

- ~/blog\_alura/database:/var/lib/mysql/

environment:

- MYSQL\_ROOT\_PASSWORD=teste123

blog:

image: wordpress

environment:

- WORDPRESS\_DB\_PASSWORD=teste123

links:

- db:mysql

ports:

- 80:80

**<!-- fim -->**

**docker-compose up -d** : executa uma série de comandos, o **-d** deixa rodando em BACKGROUND,

**docker-compose kill** : força a parada da máquina,

**docker-compose rm** : remove máquina que foi criada via COMPOSE,