

# Docker



# História do Docker

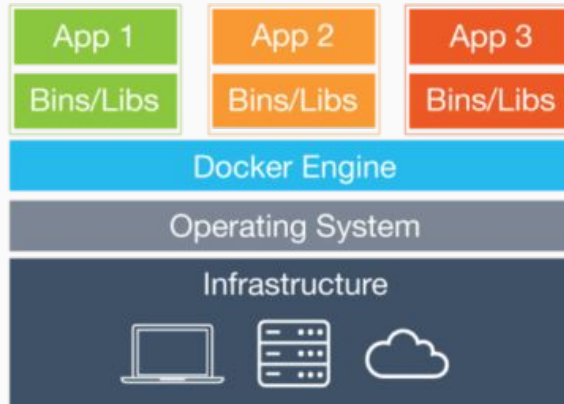
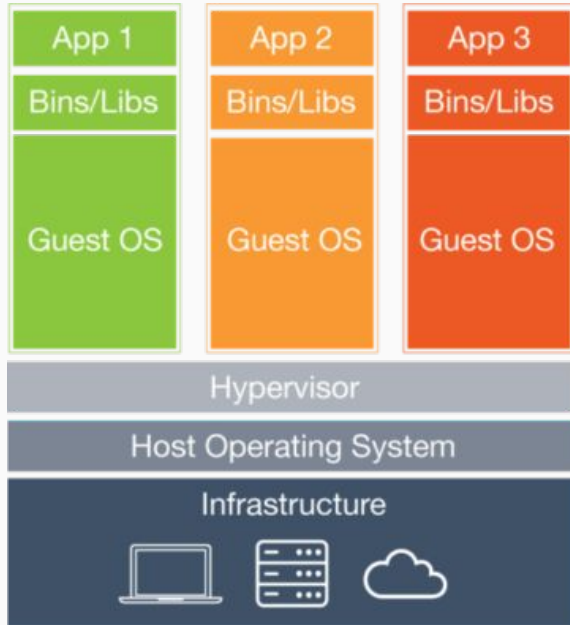


O Linux Container ou LXC como é mais conhecido, foi lançado em 2008 e é uma tecnologia que permite a criação de múltiplas instâncias isoladas de um determinado Sistema Operacional dentro de um único hospedeiro ou em outras palavras, **é uma maneira de virtualizar aplicações dentro de um servidor GNU/Linux.**

Hoje um dos mais conhecidos LXC's do mercado é o **Docker**, escrito em GO, que nasceu como um projeto open source da DotCloud, uma empresa de PaaS (Platform as a Service) que apesar de estar mais interessada em utilizar LXC apenas em suas aplicações, acabou desenvolvendo um produto que foi muito bem aceito pelo mercado.

Tecnologia Open Source que permite criar, executar, testar e implantar aplicações distribuídas dentro de containers para rodar em qualquer ambiente.

# VMs x Containers



- Portabilidade de aplicação
- Isolamento de processos
- Gerenciamento de recursos

# Elementos do Docker

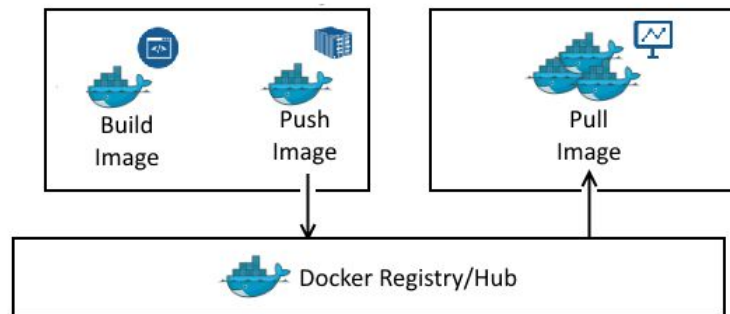
Docker Daemon

Docker CLI

Docker registry

Docker Images

DockerFile



```
FROM openjdk:8-jre-alpine

ENV APP_TARGET target
ENV APP cadastro-cliente-0.0.1-SNAPSHOT.jar

RUN mkdir -p /opt
COPY ${APP_TARGET}/${APP} /opt

CMD java -Xms${JAVA_XMS:-512m} -Xmx${JAVA_XMX:-512m} -jar /opt/${APP}
```

# Docker Compose

O Docker Compose é uma ferramenta para a criação e execução de múltiplos containers de maneira padronizada e que facilite a comunicação entre eles.

Com o Compose, é possível usar um único arquivo para definir como será o ambiente de uma aplicação e usando um único comando possível criará e iniciará todos os serviços definidos.

```
version: '3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
  db_data:
```

# Docker Cheat Sheet

## BUILD

### Construir uma imagem a partir do Dockerfile

```
docker build -t myapp:1.0 .
```

### Listar todas as imagens localmente

```
docker images
```

### Deletar uma imagem

```
docker rmi alpine:3.4
```

## RUN

### Rodando um container do Nginx

```
docker run --name webserver -p 8080:80 -v /home/user/site:/usr/share/nginx/html:ro -d nginx
```

### Logar em um registro (the Docker Hub by default)

```
docker login my.registry.com:8000
```

### Enviar uma imagem para o registro

```
docker push myrepo/myalpine:3.4
```

### Conectando em um container

```
docker exec -it <nome do container em execução> /bin/bash
```

## SHIP

### Obter uma imagem através de um registro

```
docker pull alpine:3.4
```

### Logar em um registro (the Docker Hub by default)

```
docker login my.registry.com:8000
```

### Enviar uma imagem para o registro

```
docker push myrepo/myalpine:3.4
```

# Mão na massa

## 1. **Instalação**

```
curl -sSL https://get.docker.com/ | sh
```

## 2. **Instalar o docker-compose e git**

```
yum install docker-compose git
```

## 3. **Git clone de alguns exemplos para demonstração**

```
git clone https://github.com/danielmrosa/minicurso-docker.git
```

## 4. **Realizar o Deploy de um Wordpress com o Docker Compose**

```
cd wordpress-mysql-compose
```

```
docker-compose up -d
```

```
abrir http://localhost:8000
```

# Próximos passos

## Orquestração de containers



kubernetes



## Service Mesh e Observability



Istio



Grafana



kiali



JAEGER



Prometheus



elastic



# Referências

<https://docs.docker.com/>

<https://dockercheatsheet.painlessdocker.com/>

<https://start.spring.io/>