



Docker

Marco Túlio
Dev backend
Squad de mobilidade

Quanto aqui já tiveram contato com docker ?



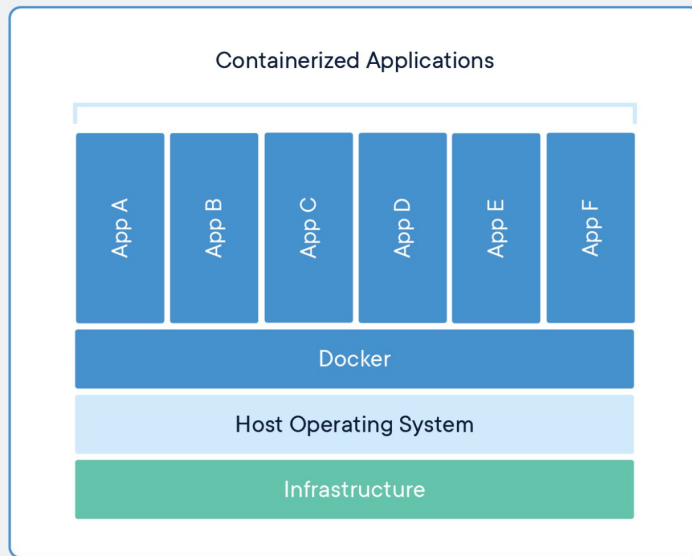
- O que é o docker ?
 - Containers
 - Images
 - Dockerfile
 - Volumes
 - Docker compose
 - Exemplo na prática
 - Dica

O que é o docker ?

- Estamos falando de várias coisas:
 - O software de TI "Docker" é uma tecnologia de containerização para criação e uso de containers Linux®.
 - A comunidade open source do Docker trabalha gratuitamente para melhorar essas tecnologias para todos os usuários.
 - A empresa Docker Inc. se baseia no trabalho realizado pela comunidade do Docker, tornando-o mais seguro, e compartilha os avanços com a comunidade em geral. Depois, ela oferece aos clientes corporativos o suporte necessário para as tecnologias que foram aprimoradas e fortalecidas.

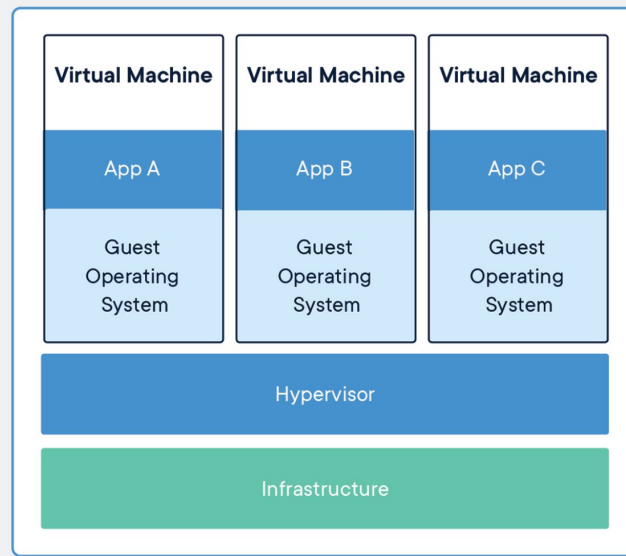
<https://www.redhat.com/pt-br/topics/containers/what-is-docker>

Containers Vs Máquinas Virtual



CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.



VIRTUAL MACHINES

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries - taking up tens of GBs. VMs can also be slow to boot.

Containers

- Uma abstração do SO (Ubuntu, Debian, mint, CentOS, etc)
- Binários/Libs
- Aplicação (ex: Mysql, Oracle, etc).

Images

- Tudo que iremos executar em um container, vem de uma imagem e essa imagem pode ser uma que você mesmo criou ou uma imagem oficial.
- Na imagem temos uma definição da estrutura do nosso container.
- Existe um hub de images para compartilhamento das docker-images
 - [Docker hub](#)

Dockerfile

- Dockerfile é um simples arquivo texto que contém todos os comandos para gerar uma imagem.
- Criando uma nova imagem: `docker build -t imageName .`
- Você pode rodar um container usando uma imagem já pronta ou montar sua própria imagem.

Dockerfile

```
FROM node:10-alpine
```

```
WORKDIR /app
```

```
COPY . .
```

```
RUN yarn install --production
```

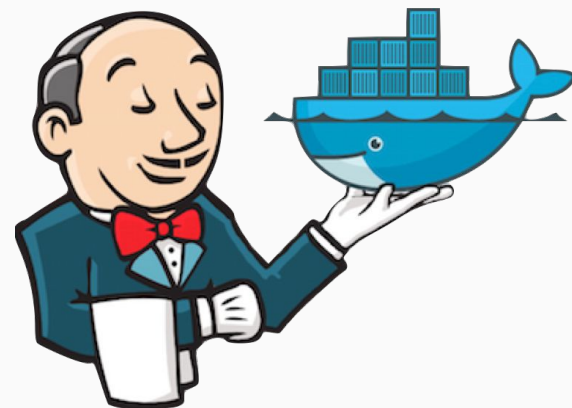
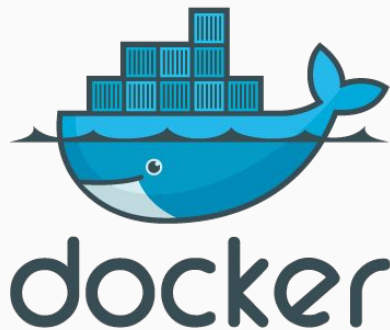
```
CMD ["node", "/app/src/index.js"]
```

Volumes

- Os arquivos que é um container possui, são aqueles que foram copiados para dentro da imagem no momento do build.
- Enquanto o container está rodando, você pode adicionar arquivos ao container enquanto ele está rodando. Porém ele voltará para o estado inicial definido pela image.
- Para resolver essa situação, você pode apontar um diretório do host para dentro do container. Dessa forma, esses dados não são perdidos.

Como trabalhar com vários containers ao mesmo tempo ?

- `docker run -dp 3306:3306 mysql:latest`
- `docker run -dp 8080:8080 jenkins:latest`
- `docker run -dp 8000:8000 openjdk`



Docker-compose

- Docker Compose é a maneira de declararmos vários containers para nossa aplicação e ele é executado pelo comando docker-compose.
- Os containers e suas configurações são definidas em um arquivo denominado: **docker-compose.yml**
- [Exemplo docker-compose](#)

Dicas

- Brinque com docker !
- [Play with docker](#)

Um exemplo real

- Um exemplo simples, porém funcional.
- Subindo um container oracle, para rodar a hybris.
- Tutorial: [Ecommerce-documentation](#)