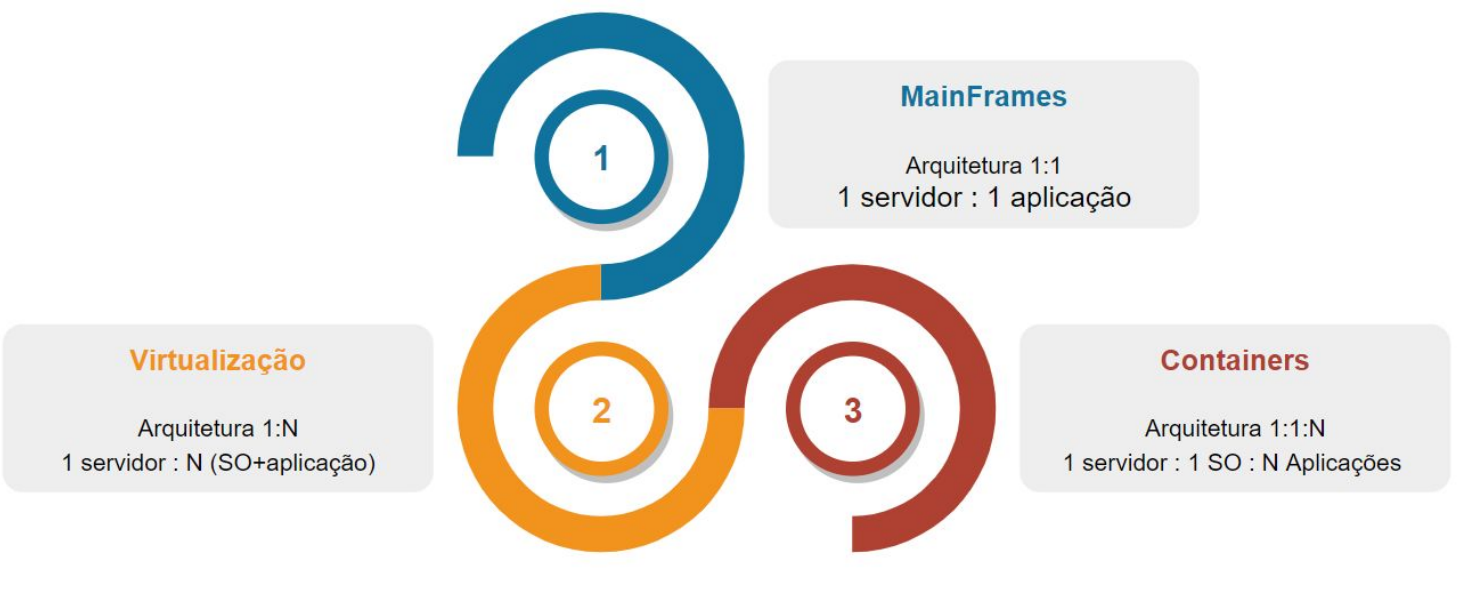
Introdução ao Docker

Um pouco mais sobre os containers

O QUE VEREMOS

- A história até o Docker
- O que é virtualização e quais seus desafios?
- O que são containers?
- O que é Docker e como funciona?
- Vocabulário Docker
- O que é o DockerFile?
- O que é o DockerCompose?
- Exemplos de comandos Docker
- Criando um container na prática

A história até o Docker

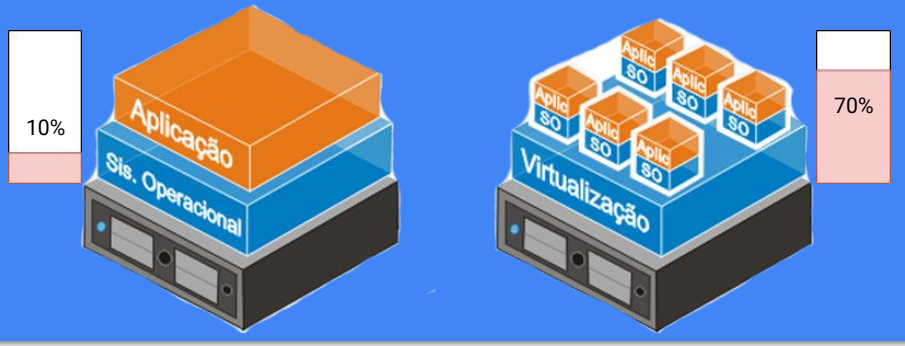


docker

Motivação e Conceitos

- **Problema:** dependências, portabilidade e escalabilidade
- **Solução:** Containers isolados e leves
- **Conceitos básicos:** Imagens, Containers, Volumes e Networks
- Docker vs Máquinas Virtuais

Virtualização



O que é:

“Tecnologia que permite o compartilhamento de recursos físicos de um servidor através do hypervisor, criando ambientes virtuais independentes executando um sistema operacional e aplicações instaladas nele.”

Ganhos:

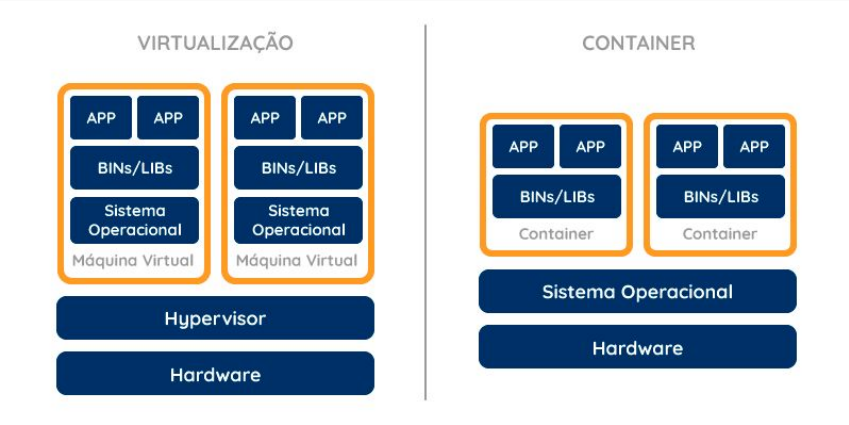
- Otimização de utilização de recursos físicos.
- Redução do consumo de energia.
- Redução de tempo para novos projetos produtivos.

Desafios:

- Carrega dependências de cada SO instalado.
- Altos custos com licenças de SO.
- Alto custo com recursos para gestão.

docker

O que são containers?



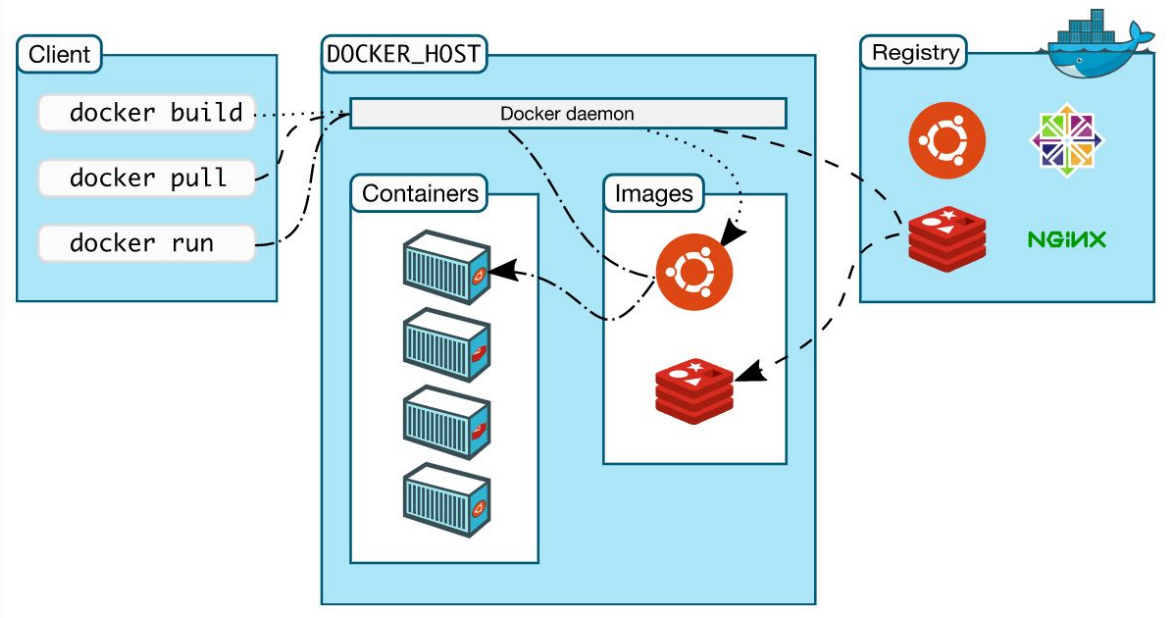
"É a tecnologia que permite o compartilhamento dos recursos físicos e do sistema operacional, empacotando a aplicação e suas dependências em uma camada independente"

docker

O que é o Docker e como funciona?

“Docker é uma plataforma aberta para desenvolvimento, transporte e execução de aplicações utilizando o conceito dos containers. ”

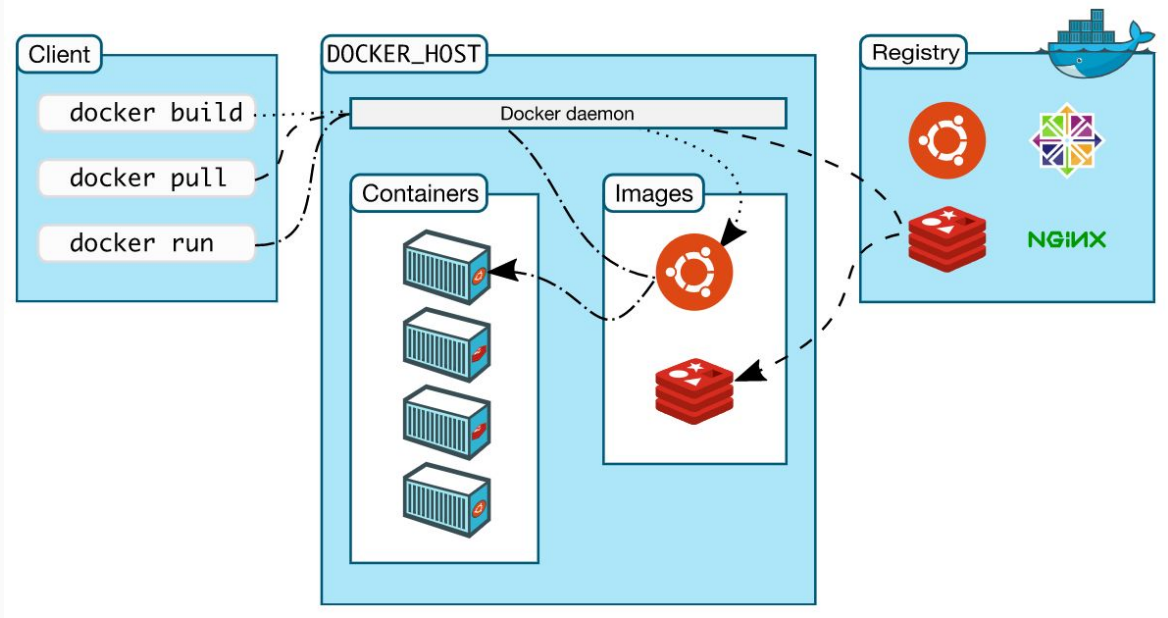
1. Build Image - Processo de construção da imagem através do Dockerfile.
2. Ship Image - Compartilhe sua imagens em um repositório público ou pivado.
3. Run Image - Execute os comandos de dentro de sua imagem



docker

Vocabulário Docker

- Container
- Build
- Images
- Run
- Docker Deamon
- Dockerfile
- Dockerignore
- Repository
- Registry
- Tag
- Volume
- Network
- Docker Compose
- Docker Swarm
- Task
- mais...

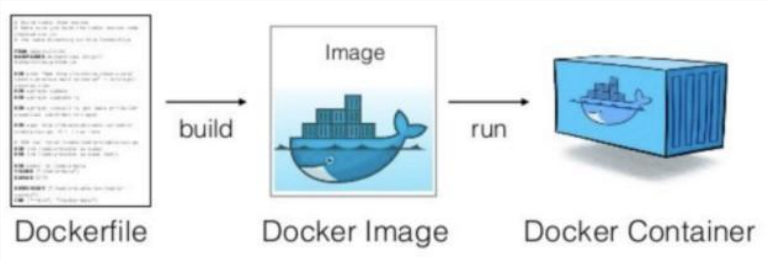


Arquitetura do Docker

Componentes principais:

- Docker Engine
- Docker Daemon
- Docker CLI
- Docker Hub

Fluxo: Imagem → Container → Execução

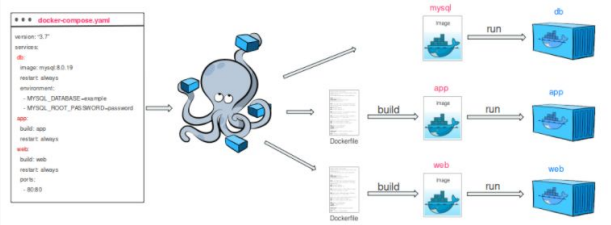


O que é Dockerfile?

“Dockerfile é um arquivo de texto que contém todos os comandos que o Docker tem que utilizar para construir sua imagem.”

Estrutura do Arquivo

```
FROM python:3.8-slim-buster

ENV FOO=/app

WORKDIR ${FOO}   # WORKDIR /app

COPY requirements.txt requirements.txt

RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```



O que é o DockerCompose?

"O Docker Compose é uma ferramenta para a criação e execução de múltiplos contêineres.

Com um único comando você criará e iniciará todos os serviços definidos"

Estrutura do Arquivo

```
version: "3.7"
services:
  db:
    image: mysql:8.0.19
    command:
'--default-authentication-plugin=mysql_native_password'
    restart: always
    environment:
      - MYSQL_DATABASE=example
      - MYSQL_ROOT_PASSWORD=password

  app:
    build: app
    restart: always

  web:
    build: web
    restart: always
    ports:
          - 80:80
```

docker

Instalação e Configuração

- Linux: apt install docker.io
- macOS: Docker Desktop
- Windows: WSL + Docker Desktop

Verifique a instalação:

- docker --version
- docker info

Exemplos de Comandos Docker

docker ps -a *#Listar containers*

docker images *#Lista suas imagens*

docker image rm <imageId>

docker run

docker tag python-docker:latest python-docker:v1.0.0 *#Cria tag em seu container*

docker exec -it <nomeContainer> bash -l

Comandos Básicos com Exemplos

docker pull nginx
docker run -d -p 8080:80 nginx
docker ps
docker stop <container>
docker rm <container>
docker images
docker rmi <imagem>

Criando suas Próprias Imagens

- Dockerfile define como a imagem é construída.
Exemplo:
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN npm install
CMD ["npm", "start"]

Comandos:
docker build -t minha-imagem .
docker run -p 3000:3000 minha-imagem

Volumes e Persistência

- Volumes armazenam dados fora do container.
- Permite persistir dados mesmo após parar o container.

Ex.: docker run -d -v meu_volume:/var/lib/mysql mysql

Redes no Docker

- Tipos: bridge, host, none
- Comunicação entre containers:

```
docker network create minha_rede
docker run -d --name db --network minha_rede postgres
docker run -d --name app --network minha_rede meu_app
```

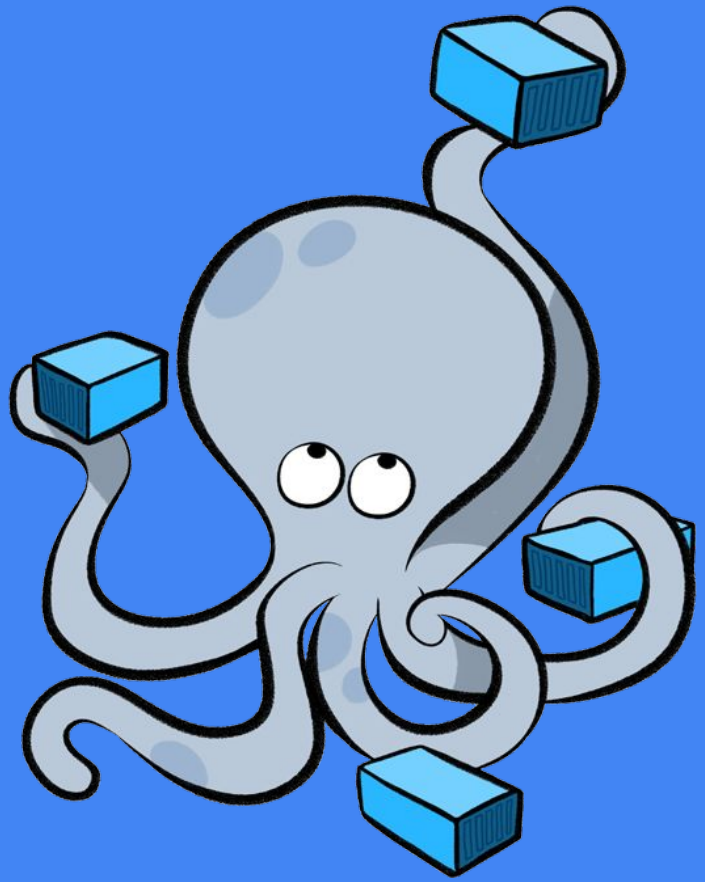
Docker Compose

Orquestra múltiplos containers com um único arquivo.

docker-compose.yml exemplo:

```
version: '3'
services:
  app:
    build: .
    ports:
      - '3000:3000'
  db:
    image: postgres
    volumes:
      - db_data:/var/lib/postgresql/data
volumes:
  db_data:
```

Vamos para a prática!



Boas Práticas e Dicas

- Utilize .dockerignore
- Evite imagens muito grandes
- Use camadas eficientes
- Atualize imagens base regularmente
- Não armazene segredos em imagens

docker

Exercícios Práticos

- Suba um container nginx e acesse via navegador.
- Crie uma imagem própria com Dockerfile simples.
- Monte um docker-compose com app e banco.

Obrigado!

Henrique Mota