

17/05/2024

Docker Basics

Como o docker pode facilitar a sua vida

João Pedro Barboza Lioneza

Sumário

- Container
 - Conceito
 - Container x VM
 - Exemplos
- Docker
 - Introdução
 - Docker Image
 - Dockerfile
- Utilização
 - docker build
 - docker run
 - docker-compose
 - docker network
- Docker Hub

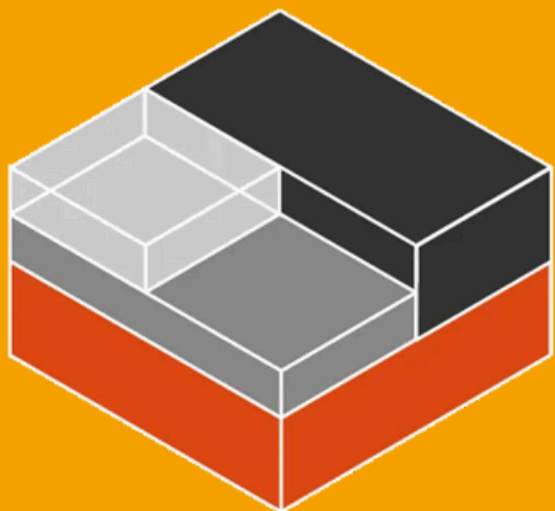
Container



docker



podman



LXD

Conceito

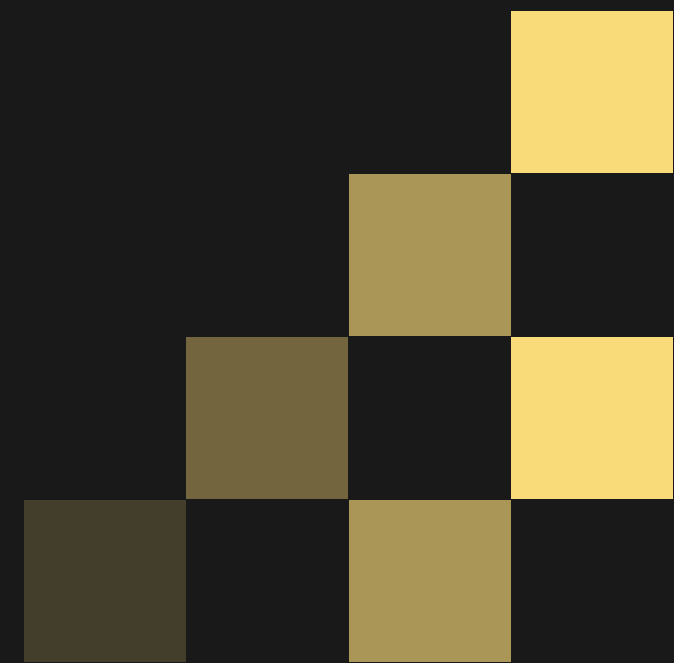
- Virtualização com ambiente isolado

Container X Máquina Virtual

- Diferenças
- Aplicações

Exemplos

- Docker
- LXD
- Podman

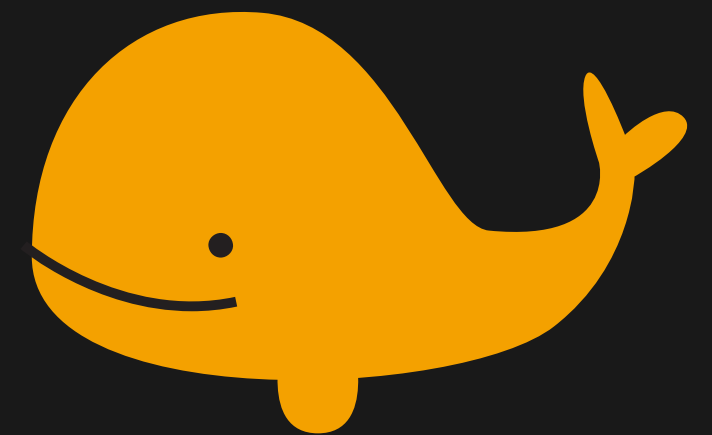


Docker

"Mas Na minha Máquina Funciona"

INTRODUÇÃO

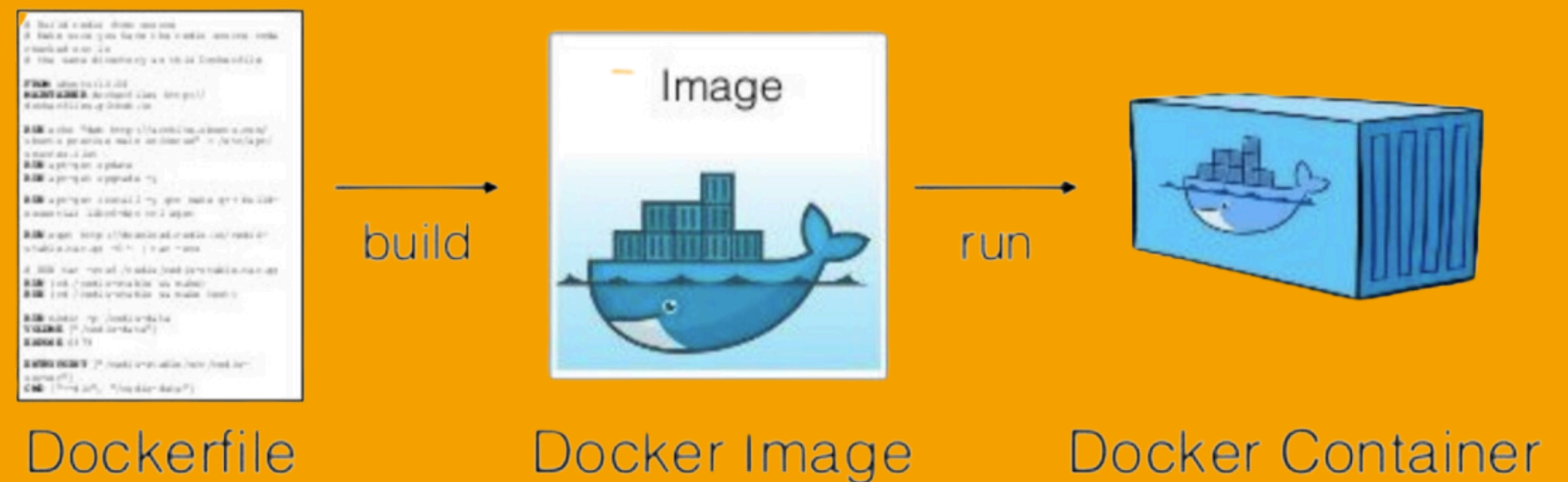
- Open source
- Escrito em Go
- Lançado em 2013
- Docker, Inc



Docker

Docker Image

- Arquivo executável usado para criação do container
- Lógica semelhante ao versionamento do git
- Versões diferentes sendo executadas ao mesmo tempo



Docker

Dockerfile


- Conjunto de instruções para criação da imagem

```
1 FROM python:3.11
2
3 WORKDIR /usr/src/app
4
5 RUN pip install pipenv
6
7 COPY Pipfile Pipfile.lock ./
8
9 RUN pipenv install
10
11 COPY . .
12
13 EXPOSE 8000
14
15 CMD ["pipenv", "run", "prod"]
```

Utilização


Comandos básicos

- docker build



```
1  docker build -t docker-example:latest .  
2
```

- docker run



```
1  docker run -d --name docker-example --env-file .env docker-example:latest
```

Utilização

Docker Compose

- Vantagens
 - Configurações mais complexos
 - Orquestrador de containers
 - Persistência de dados (volume)
 - Alocação de rede (docker network)
 - Configuração de múltiplos containers
 - Serviços de backend e frontend separados
 - Banco de dados
 - Dependência de serviços
 - Comandos mais simples
 - `docker-compose build`
 - `docker-compose down`

Utilização

Docker Compose

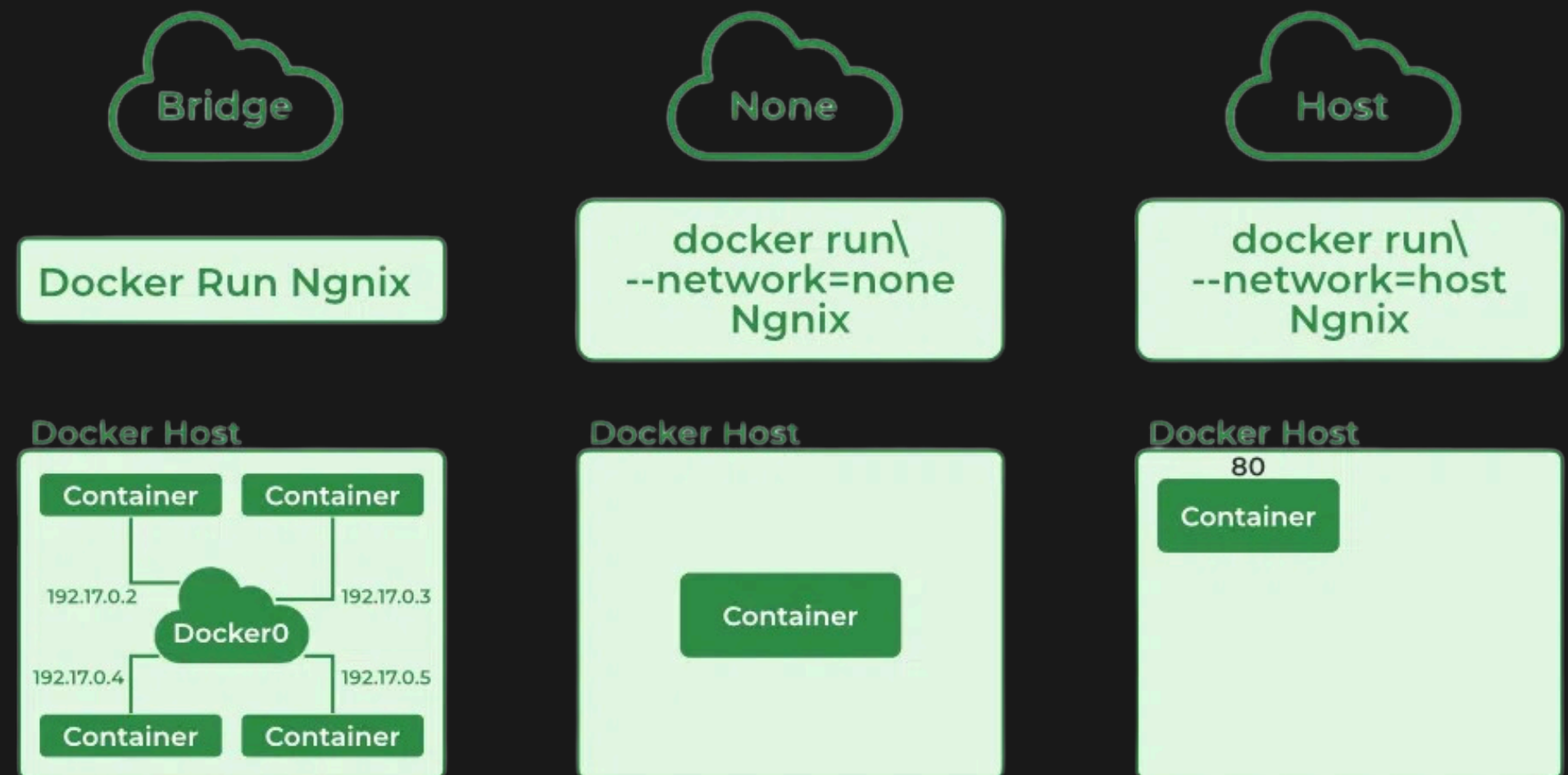
- Sintaxe

```
1  version: "3"
2  services:
3    prod:
4      image: docker-example:latest
5      ports:
6        - "8000:8000"
7      build:
8        context: .
9        dockerfile: compose.Dockerfile
10     env_file:
11       - .env.docker
12     networks:
13       - default
14     depends_on:
15       - mongo
16
17     mongo:
18       image: mongo:latest
19       ports:
20         - "27018:27017"
21
22       # Adicionando persistência de dados
23       volumes:
24         - mongo_data:/data/db
25
26 volumes:
27   mongo_data:
28
29 networks:
30   default:
31     driver: bridge
```

Utilização

Docker Network

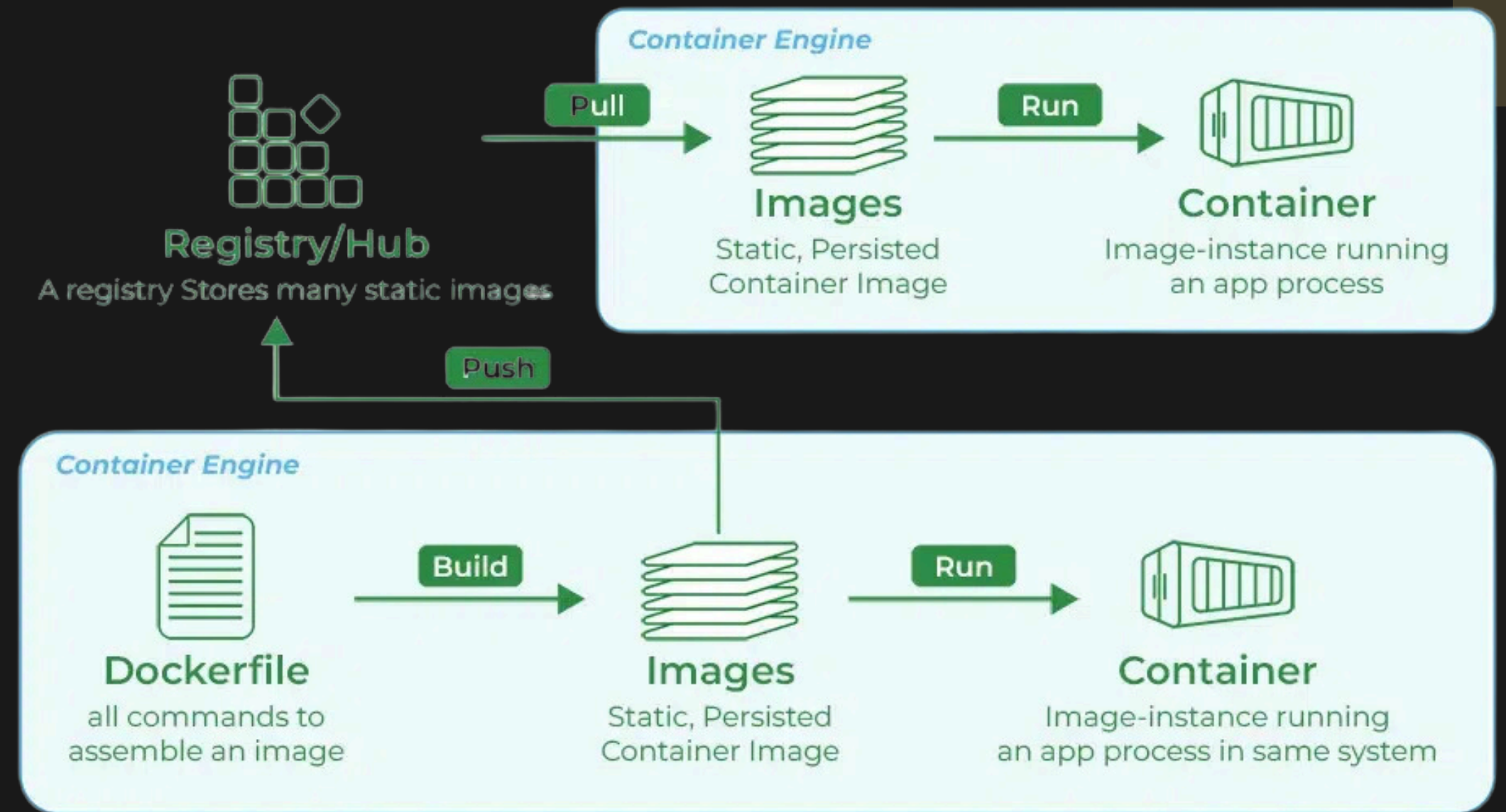
- Roteamento virtual
- Tipos de rede
 - host
 - bridge
 - none



Docker Hub

Afinal, o serviço não vai rodar apenas localmente

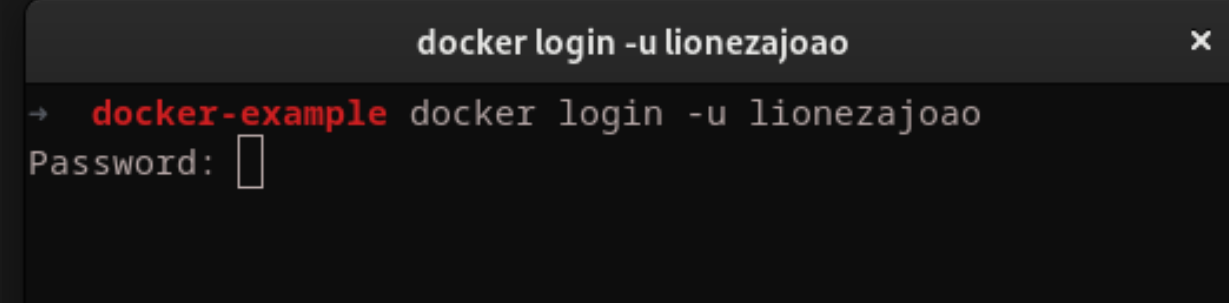
- Mantido pela Docker, Inc
 - Versão grátis x paga
- Não é a única solução
 - Container Registry (GCP)
 - Amazon ECR
 - Azure Container Registry



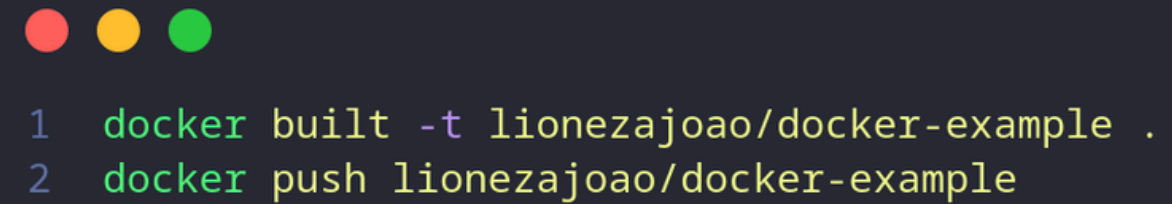
Docker Hub

Comandos

- Login
 - Shell com autenticação necessária (docker login)
- Build
 - Imagem com inficação para local de comit
 - <USERNAME>/<NOME DA IMAGEM>:<TAG>
- Push
 - Semelhante ao build
 - Só será executado caso o usuário esteja autenticado

A terminal window titled "docker login -u lionezajoao" with a close button. It shows a prompt "→ docker-example" followed by the command "docker login -u lionezajoao" and a "Password:" prompt with an empty input field.

```
docker login -u lionezajoao
→ docker-example docker login -u lionezajoao
Password: 
```

A terminal window with three colored window control buttons (red, yellow, green) at the top. It contains two numbered lines of code: "1 docker built -t lionezajoao/docker-example ." and "2 docker push lionezajoao/docker-example".

```
1 docker built -t lionezajoao/docker-example .
2 docker push lionezajoao/docker-example
```