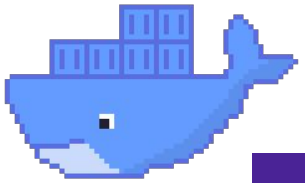
trybe

Orquestrando Containers com o Docker Compose



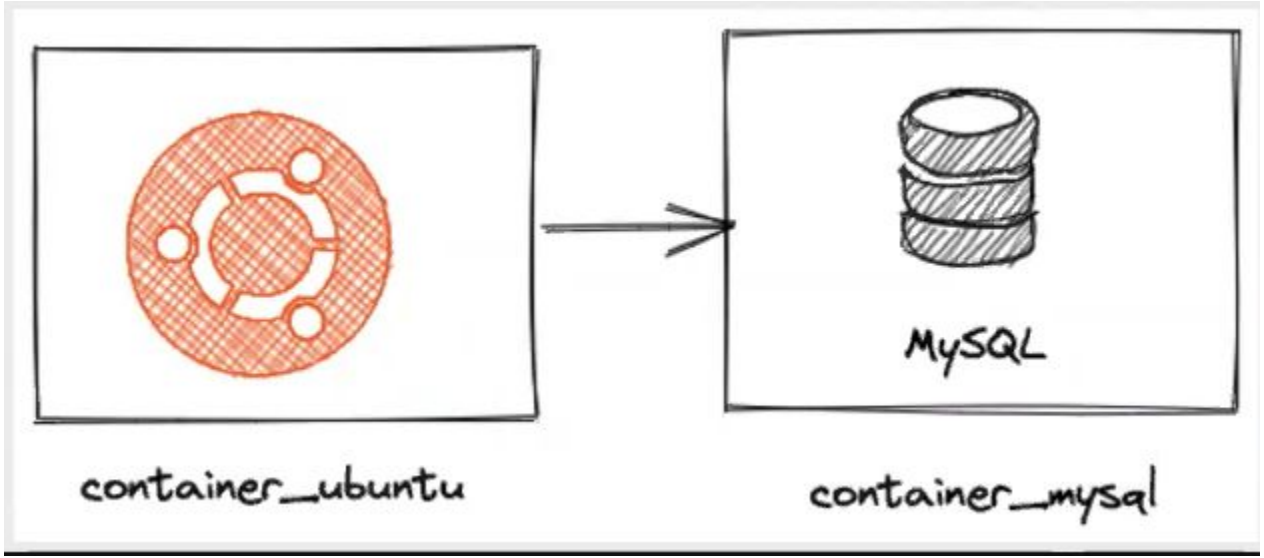
Bloco 19 - Aula 19.3

Roadmap da Aula

- Gerenciar redes Docker, utilizando-as para a comunicação e isolamento de containers;
- Persistir dados dos containers usando volumes;
- Criar arquivos Compose para gerenciar todo seu ambiente com Docker;
- Gerenciar Services, Network e Volumes a partir do Compose;

Case

- Uma API que precisa comunicar com um banco de dados MySQL;



Como conversar entre containers?

- Para que dois containers consigam "conversar" um com o outro, precisamos de um recurso chamado network;
- Basicamente, é criar uma rede virtual dentro do contexto do docker para que os containers tenham visibilidade da existência um do outro;

```
docker network ls

docker container run -dit --name container1 busybox

docker container run -dit --name container2 busybox

docker ps -a
```

Como conversar entre containers?

- Para que dois containers consigam “conversar” um com o outro, precisamos de um recurso chamado network;
- Basicamente, é criar uma rede virtual dentro do contexto do docker para que os containers tenham visibilidade da existência um do outro;

```
docker exec -it container1 /bin/sh

docker exec -it container2 /bin/sh

ping container1

ping container2
```

Como conversar entre containers?

- Precisamos criar uma rede entre nossos containers;

```
docker network ls

docker network create -d bridge tryber-network

docker network ls
```

```
docker network

docker network connect tryber-network container1

docker network inspect tryber-network

docker network connect tryber-network container2

docker network inspect tryber-network
```

Como conversar entre containers?

- Precisamos criar uma rede entre nossos containers;

```
docker run --name container3 -dit --network tryber-network busybox

docker network inspect tryber-network


docker exec -it container3 /bin/sh

ping container1

ping container2
```

Como conversar entre containers?

- Precisamos desconectar um container da rede;

```
docker network disconnect tryber-network container3

docker network inspect tryber-network

ping container3
```

```
docker network connect tryber-network container3

docker network inspect tryber-network

ping container3
```

Como conversar entre containers?

- MAS TODOS ESTES EXEMPLOS PARA FAZER PINGGGGGGGGGG?

Vamos instalar o MySQL?

- Precisamos instalar o MySQL e o Ubuntu para criar a conexão entre containers;

```
docker container run -dit --name container_ubuntu ubuntu

docker container run -dit --name container_mysql -e
MYSQL_ROOT_PASSWORD=root mysql
```

https://hub.docker.com/_/mysql

Vamos conectar no MySQL?

- Como me conecto?

```
docker exec -it ID bash

mysql -u root -proot


SHOW DATABASES;

CREATE DATABASE docker;
```

Vamos conectar no MySQL?

- Como me conecto entre máquinas?

```
docker exec -it ID bash

apt-get update -y

apt-get install mysql-client

mysql -u root -proot -h container_mysql
```

Vamos conectar no MySQL?

- Como me conecto entre máquinas?

```
docker network create mysql-network

docker network connect mysql-network ID do Ubuntu

docker network connect mysql-network ID do MySQL
```

```
docker exec -it ID do Ubuntu bash

mysql -u root -proot -h container_mysql

SHOW DATABASES;
```

Docker Compose - docker-compose.yml

- Criando um container do Ngnix e fazendo um biding de um diretório como volume

```
version: '3'

services:
  web:
    image: nginx:latest
    volumes:
      - ./public_html:/usr/share/nginx/html
    ports:
      - 8080:80
```

Docker Compose - docker-compose.yml

- Criando um container do Ngnix e fazendo um biding de um diretório como volume

```
docker-compose up

docker-compose up -d *alternativo


docker container ps

docker container inspect custom_nginx_web_1

http://localhost:8080/
```

Docker Compose - docker-compose.yml

- Criando um outro container com MySQL

```
docker-compose down
```

```
docker-compose up
```

```
version: '3'

services:
  web:
    image: nginx:latest
    volumes:
      - ./public_html:/usr/share/nginx/html
    ports:
      - 8080:80
  db:
    image: mysql:latest
    environment:
      - MYSQL_ROOT_PASSWORD=root
    volumes:
      - ./db:/var/lib/mysql
    ports:
      - 33060:3306
```