

# Introdução aos containers Docker

Igor Neves Faustino

- Não são mini-VMs
- São apenas processos
- Isolados do que está ao redor

# Imagens vs Containers

- Imagem é a aplicação que será rodada
- Container é uma instancia de uma imagem rodando como um processo
- Pode existir varios containers rodando de uma unica imagem

# Comandos Docker

## versão antiga

```
docker <command> (options)
```

## versão nova

```
docker <command> <sub-command> (options)
```

# Rodando nossos primeiros containers

Comando: `$ docker container run hello-world`

## Explicando o comando

- **container**: Comandos relacionados a containers
- **run**: subcomando utilizado para inicial um novo container

## Ações executadas

- Procura a imagem localmente (não acha)
- O cliente baixa a imagem do Docker Hub
- O docker inicia um novo container com a imagem baixada
- O docker replica a saída produzida no container para o terminal

# Tentando algo mais complexo

Comando: `$ docker container run -it ubuntu`

## Explicando o comando

- **-it**: opção utilizada para executar o comando de forma interativa

# Liberando uma porta externa

Comando: `$ docker container run -p 8080:80 nginx`

## Explicando o comando

- **-p** ou **-publish**: opção para liberar uma porta externa

## Ações executadas

- Baixa a imagem Nginx
- Inicia um container com a imagem baixada
- Abre a porta 8080 no host
- Liga o tráfego do host na porta 80 do container

# Outras opções disponíveis

- **-name**: dá um nome para o container
- **-d** ou **-detach**: Executa um container em background (Desacoplado do terminal)
- **-env** ou **-e**: define uma variável de ambiente



# Logs gerados por containers

Comando 1: `$ docker container run -d -p 8080:80 --name nginx`

Comando 2: `$ docker container logs nginx`

# Manipulando containers

Comando: `$ docker container ls`

Comando: `$ docker container rm`

comando `$ docker container stop`

comando `$ docker container start`

## Execute varios containers

- imagens: `nginx`, `mysql:5.7` e `httpd`
- todos os containers devem ser desacoplados
- todos devem possuir um nome
- portas: `nginx 80:80`, `httpd 8080:80` e `mysql 3306:3306`
- env: `mysql MYSQL_RANDOM_ROOT_PASSWORD=yes` (use o comando `log` para ver o password criado aleatoriamente na inicialização)
- delete todos os containers