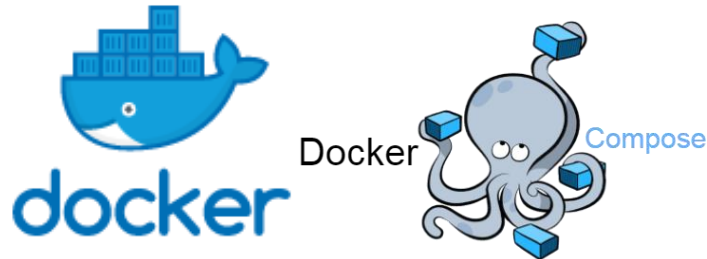


# Guia Básico de Docker e Docker Compose



*Obs!* Esse é um guia com o mínimo do que você precisa saber para usar o Docker e o Docker Compose. Aqui estão todos os comandos do Docker e do Docker Compose usados no curso e as explicações em detalhes de cada um deles. Sempre que necessário use-o como ponto de apoio e à medida que for surgindo a necessidade procure aprender mais sobre essas duas tecnologias.

## ➤ Comandos

`docker ps` → Lista os contêineres em execução. O parâmetro **-a** é opcional e pode ser usado para listar todos os contêineres inclusive os que estão parados.

`docker-compose up -d --build` → Inicializa a nossa aplicação. O parâmetro **-d** é pra que inicialize detachado e me permita usar o console e o parâmetro **--build** é pra que o Docker Compose recompile o projeto. Os dois parâmetros são opcionais

`docker-compose down` → Baixa todos os contêineres inicializados pelo comando anterior.

`docker-compose ps` → Similar ao **docker ps** lista os contêineres em execução vinculados ao arquivo **\*.yaml** que inicializou a execução do Docker Compose.

**wintpy** `docker exec -it CONTAINER_ID bash` → Acessa o terminal do container. O parâmetro **wintpy** só é necessário em ambiente Windows se estiver no Linux ele não precisa ser digitado.

`docker rmi $(docker images -q) -f` → A parte dentro dos parêntesis lista todas as imagens. E a parte externa remove todas as imagens retornadas na lista interna. O **-f** pode ser usado, de forma opcional, para forçar a remoção, mesmo que a imagem esteja sendo usada em algum contêiner em execução.

`docker rm $(docker ps -a -q) -f` → A parte dentro dos parêntesis lista todos os contêineres. E a parte externa remove todos os contêineres retornados na lista interna. O **-f** pode ser usado, de forma opcional, para forçar a remoção, mesmo que o contêiner esteja sendo usado.

`docker logs CONTAINER_ID` → Acessa o log de um contêiner.

`docker login docker.io` → Autentica no DockerHub via console.

`docker images` → Lista todas as imagens.

`docker tag <SRC_IMAGE>:<TAG> <DOCKER_HUB_USER>/<DESTINATION_IMAGE>:<TAG>` →  
Cria uma tag a partir de uma imagem existente.

`docker push IMAGE_NAME` → Envia uma imagem para o DockerHub. Normalmente o nome é  
**seu\_usuario\_docker\_hub/image\_name**.