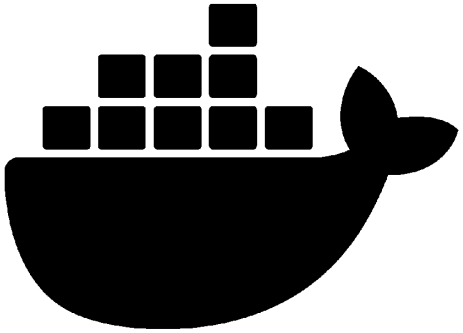

Curso de Docker

3. Comandos Docker

Mateus Schwede - UB Social

Parâmetros

idContainer pode também ser substituido pelo nome

-f: Especifica dockerfile

-d: Executar Container em background

--rm: Sobrescreve Container com novo, caso já exista antigo similar

-t: Tag / Apelido

-i: Execução de comando em modo interativo, como um shell

-ti: TTY, utilizar emulador de Terminal

-e: Environment/Entrypoint, especificar variáveis prévias de configurações

-v: Volume

Ponto no final: Especifica diretório atual

$(pwd): Especifica caminho do diretório atual

-p: Especifica porta no Container estará exposta para porta do host (pHostDocker:pContainer)

Container a partir de Imagem pronta

Criar Container

Criar Container: docker run -d --name nomeContainer nomeImagem (Também acessa e baixa, se não há na máquina)

Ex: docker run -d --name meuUbuntu ubuntu

Criar e entrar no container: docker run -ti nomeImagem (Ctrl+p+q sai do shell sem matar container)

Ver Containers em execução: docker ps (-a no final mostra todos containers)
Ver informações do container: docker inspect idContainer (Inserir para filtro: '| grep termoPesquisa')

Criar container em background, com tty e interativo: docker run -dti nomeImagem
Voltar para o container: docker attach idContainer

Criar container com variáveis de ambiente: docker run -d --name banco -e MYSQL_PASSWORD=abc -e MYSQL_USER=fulano mariadb

Dentro do Container

Executar comandos em Container em execução: docker exec -i nomeContainer comando
bash ao invés de comando, incluindo -t, para entrar no container)
Ex: docker exec -ti ubuntu bash

Sair do Container: exit

Iniciar container: docker start idContainer (stop para)
Pausar container: docker pause idContainer (unpause retoma)

Informações do Container

Ver status do container: docker stats idContainer

Ver processos do container: docker top idContainer

Ver logs do container: docker logs idContainer

Excluir Container

Excluir container (Parado): docker rm idContainer (Incluir -f para em execução, excluir manterá imagens)

Excluir todos containers: docker rm -f $(docker ps -qa) (-q é quit, mostra somente id's dos containers)

Container a partir do zero

Manipular Imagem

Construir Imagem: docker build -t nomeImagem:opcionalInfo -f caminho/nomeDockerFile .

Ex: docker build -t mysql-image -f api/db/Dockerfile .

Ver imagens: docker image ls

Somente baixar imagem: docker pull nomeImagem (push sobe imagem)

Exemplo nome de imagem: *docker.io/library/debian:buster-slim* (Ou usuario/nomeImagem:versaoImagem, versão padrão é latest)

Criar apelido para imagem: docker image tag nomeAtual novoAlternativo

Excluir imagem: docker image rm nomeImagem (-f para imagens com containers existentes)

Exemplo prático

1. **Criar Dockerfile:**
   FROM php:7.2-apache
   WORKDIR /var/www/html

2. **Construir imagem:** *docker build -t php-image -f Dockerfile .*
3. **Executar container com imagem:** *docker run -d -p 8888:80 --rm --name containerPhp php-image*
4. **Entrar no Container:** *docker exec -ti containerPhp bash*

Resumindo

docker run -d --name banco -e MYSQL_USER=fulano -v dados:/var/lib/mysql -p 3306:3306 mariadb

UB Social

ubsocial.github.io