



Comandos úteis

ADMINISTRANDO CONTAINER DOCKER

Download e execução de imagem

\$ docker run nomelimagem

Visualizar container em execução

\$ docker ps

Imagens na máquina

\$ docker images

Container que não estão em execução

\$ docker ps -a

Executar a imagem e interagir com ela

\$ docker run -ti nomelimagem

Sair, mas manter o container em execução

ctrl + p + q

Voltar para container

\$ docker attach containerID

Matar container

ctrl + d

Criar container sem executá lo

\$ docker creat nomeImagem

Encerrar container

\$ docker stop containerID

Startar container

\$ docker start containerID

Pausar container

\$ docker pause containerID

Voltar a executar container

\$ docker unpause containerID

Visualizar utilização recursos pelo container

\$ docker stats containerID

Remover container parado

\$ docker rm containerID

Remover container em execução

\$ docker rm -f containerID

LIMITANDO CPU E MEMÓRIA E O DOCKER UPDATE

Iniciando e nomeando um container

\$ run -ti --name nomeContainer nomeImagem

Mostrar informações de memória do container

\$ docker inspect containerID | grep -i mem

Alterar limite de memória do container

\$ docker run -ti --memory valorMem --name nomeContainer nomeImagem

Alterar limite de memória de um container em execução

\$ docker update -m valorMem nomeContainer

Alterar limite de CPU do container

\$ docker run -ti --cpu-shares valorCpu --name nomeContainer nomeImagem

Alterar limite de cpu de um container em execução

\$ docker update --cpu-share valorCpu nomeContainer

VOLUMES E CONTAINER DATA-ONLY (colocar diretório dentro do container, sem ele fazer parte do container)

Criando um volume

```
$ docker run -ti -v /nomeVolume nomeImagem /bin/bash
```

Montar partições

```
$ df -h
```

Visualizar volume criado no host docker

```
$ docker inspect -f {{.Mounts}} containerID
```

Criar um diretório no host docker

```
$ mkdir /root/nomeDiretorio
```

Criar volume apontando para o diretório

```
$ docker run -ti -v /root/nomeDiretorio:/nomeVolume nomeImagem
```

Criar arquivo

```
$ touch nomeArquivo
```

Importar volume de outro container

```
$ docker run -d --name nomeContainer --volumes-from nomeVolume
```

Instruções Dockerfile

Determinar imagem será usada como base

```
FROM nomeImagem
```

Definir quem escreve o container

```
MAINTAINER seuNome
```

Executar comandos no container

```
RUN apt-get update && apt-get install nomePacote && apt-get clean
```

Copiar uma arquivo, arquivo.tar ou diretório do host docker para dentro do container

```
ADD arquivo.txt /diretorio/
```

Parâmetro para o entrypoint

```
CMD ["sh", "-c", "echo", "$HOME"]
```

Adicionar metadata

```
LABEL description="bla bla bla bla"
```

Copiar um arquivo ou diretório para dentro do container

COPY arquivo.txt /diretorio/

Permitir que um processo seja o principal dentro do container(caso o processo morra, o container morre também)

ENTRYPOINT ["/usr/bin/nomeProcesso", "-D", "FOREGROUND"]

Determinar variáveis de ambiente para o container

ENV meuNome="seuNome"

Mostrar porta do container que está disponível

EXPOSE 80

Definir usuário default da imagem

USER seuNome

Definir um diretório do container como diretório raiz

WORKDIR /diretorio

Criar um volume no container

VOLUME /diretorio

Como construir a imagem a partir do Dockerfile

\$ docker build -t nomeNovoImagem:versão .