**Docker**

**Comandos:**

**Run:**

O comando “docker container run” serve para criar e executar um container, exemplo:

docker container run hello-world

ou

docker run hello-world

Dessa forma ele vai criar um container da imagem “hello-world”, e caso a imagem não exista no seu computador local, ele irá baixar a imagem, e após baixar executara o container, sempre que executar novamente o comando “run”, ele criará um novo container. Também é possível executar um container e já remove-lo da lista de containers, usando a flag “—rm”, exemplo:

docker container run --rm debian bash --version

Ou

docker run --rm debian bash --version

Dessa forma ele executara o container, e já o removera da lista. Também é possível executar o container e já acessar o terminal do container usando a flag “-it”, exemplo?

docker container run -it debian bash

Ou

docker run -it debian bash

Dessa forma ele executará um container da imagem do debian, e já acessará o terminal do container do debian.

Também é possível atribuir um nome para o container no momento de executa-lo usando a flag “—name”, exemplo:

docker container run -—name mydeb -it debian bash

Ou

docker run -—name mydeb -it debian bash

Dessa forma o container será criado com o nome “mydeb”, e não com um nome aleatório criado pelo docker, sendo que cada container deve ter um nome único.

Para expor um container para uma porta fora do container, basta passar a flag “-p” e passar a porta, sendo os parâmetros “portaExposta:portaDentroDoContainer”, exemplo:

docker container run -p 8080:80 nginx

Ou

docker run -p 8080:80 nginx

Dessa forma o container que será gerado da imagem “nginx” rodará dentro do container na porta “80”, e fora do container(no localhost do computador) rodará na porta “8080”.

Para mapear um volume, basta usar a flag “-v” e passar a pasta do host que vc quer mapear separado por dois pontos e a pasta do container que você deseja fazer esse mapeamento, exemplo:

docker container run -p 8080:80 -v $(pwd)/html:/usr/share/nginx/html nginx

Ou

docker run -p 8080:80 -v $(pwd)/html:/usr/share/nginx/html nginx

Neste caso ele vai pegar o diretório atual que você está no terminal através do método “$(pwd)” do unix, mais a pasta“html”, e vai fazer com que ele pare de apontar para a pasta padrão do nginx que seria “/usr/share/nginx/html”, e aponte para a pasta “html” do diretório atual da máquina host.

**container ls:**

O comando “container ls” serve para listar todos os containers do docker que estão em com o status “running”, exemplo:

docker container ls

Dessa forma será exibido todos os containers que estão em execução no momento, caso deseja visualizar também os containers que não estão mais em execução, basta usar a flag “-a”, que irá exibir todos os containers independente do status, exemplo:

docker container ls -a

Para iniciar um container já criado, basta usar o “start” e o nome do container, exemplo:

docker container start -ai mydeb

Ou

docker start -ai mydeb

Dessa forma ele vai iniciar o container nomeado como “mydeb” e a flag “-ai” ele vai anexar o terminal e o “i” é de modo interativo, ou seja, serve para acessar o terminal do container.

**ps:**

O comando “docker container ps” serve para listar todos os containers que estão em execução(processos) no docker no momento, exemplo:

docker container ps

Ou

docker ps

Dessa forma ele exibira no terminal todos os containers em execução, caso também queira ver os containers que foram executados e não estão mais em execução, basta passar a flag “-a”, exemplo:

docker container ps -a

Ou

docker ps -a

Dessa forma ele exibirá não apenas os containers em execução, mas também aqueles que já foram executados e não estão mais ativos no momento.

Executando um container

|  |  |
| --- | --- |
| **Parâmetro** | **Explicação** |
| -d | Execução do container em background |
| -i | Modo interativo. Mantém o STDIN aberto mesmo sem console anexado |
| -t | Aloca uma pseudo TTY |
| --rm | Automaticamente remove o container após finalização (**Não funciona com -d**) |
| --name | Nomear o container |
| -v | Mapeamento de volume |
| -p | Mapeamento de porta |
| -m | Limitar o uso de memória RAM |
| -c | Balancear o uso de CPU |

**Verificando a lista de containers**

|  |  |
| --- | --- |
| **Parâmetro** | **Explicação** |
| -a | Lista todos os containers, inclusive os desligados |
| -l | Lista os últimos containers, inclusive os desligados |
| -n | Lista os últimos N containers, inclusive os desligados |
| -q | Lista apenas os ids dos containers, ótimo para utilização em scripts |