Contêineres – Resolução dos Exercícios

1 - Emita o comando para exibir todas as imagens e depois os contêineres existentes no seu repositório local.

Resposta:

Exibir imagens : **docker images ou docker image ls**

Exibir contêineres : **docker container ps ou docker ps ou docker container ls**

2 - Emita um comando para exibir todos os contêineres em execução

Resposta:

docker container ps -a ou docker ps -a ou docker container ls -a

3 - Crie um contêiner com base na imagem do **alpine:3.8** no **modo iterativo(-it)** de forma que ele seja excluído assim que você sair do contêiner. **Resposta:**

Para que um contêiner seja excluído assim que ele for executado usamos a flag **--rm** no comando da criação do contêiner:

Docker container run --name teste --rm -it alpine:3.8

Após sair do contêiner se você digitar **docker container ls -a** não vai encontrar o contêiner pois ele foi excluído.

4 - Crie um contêiner com base na imagem oficial do **alpine:3.8** usando o comando: **docker container create.**

Resposta:

docker container create alpine:3.8

Nota: Se você não fornecer um nome ao contêiner o docker atribui um ID e um nome aleatório

5 - Crie um contêiner chamado **cont1** com base na imagem oficial do **alpine:3.8** usando o comando: **docker container run**

Resposta:

docker container run --name cont1 alpine:3.8

Nota: o argumento **--name** atribui um apelido ao contêiner

6 - Explique em detalhes o processo de execução do comando do item 4 e quais as diferenças em relação ao item comando usado no item 3.

Resposta:

docker - é o executor do comando

container - indica que o comando vai atuar em um contêiner

run - é a porta de entrada no Docker e agrupa as seguintes funcionalidades básicas :

Contêineres – Resolução dos Exercícios

Download automático das imagens não encontradas localmente: docker image pull
Criação do container: docker container create
Inicialização do container: docker container start
Execução do contêiner : docker container exec
alpine:3.8 - é a imagem existente usada para criar o contêiner

7 - Repita **os itens 1 e 2** para ver as imagens e contêineres existentes
Resposta:
docker images
docker container ps
docker container ps -a

8 - Remova o contêiner **cont1** usando o comando apropriado
Resposta:
docker container rm cont1 ou docker container rm <ID>

9 - Crie um contêiner no modo **iterativo** **(-it)** chamado **cont2** com base na imagem oficial do **alpine:3.8**. A seguir emita o comando **echo 'teste' > teste.txt** e a seguir o comando **ls -g**. A seguir sair do contêiner.
Resposta:
docker container -it --name cont2 alpine:3.8
#> echo 'teste' > teste.txt
#>ls -g
#>exit

10 - Crie um contêiner chamado **cont3** com base na imagem **nginx:alpine** mapeando a porta 80 do navegador para a **porta 80** do contêiner usando o comando: **docker container create. A** seguir verifique as imagens e os contêineres existentes e verifique se existe algum contêiner em execução.
Resposta:
docker container create --name cont3 -p 80:80 nginx:alpine
docker images
docker container ps

11 - Inicie o contêiner **cont3** criado no item anterior, a seguir abra e inicie o navegador no endereço **http://localhost**
Resposta:
docker container start cont3
http://localhost

12 - Obtenha informações do contêiner **cont3** como*: uso de cpu, de memória, I/O, etc.*

Resposta:
docker container stats cont3

CONTAINER - ID do Container
CPU % - uso de CPU em porcentagem
MEM USAGE / LIMIT - Memória usada/Limite que você pode ter setado
MEM - uso de memória em porcentagem
NET I/O - I/O de Internet
BLOCK IO - Outros processos de I/O.

13- Inspecione o contêiner **cont3** usando o comando apropriado.
Resposta:
docker container inspect cont3

14- Crie um contêiner chamado **cont4** com base na imagem do **nginx:alpine** usando o comando **docker container create**. A seguir inicie o contêiner. Depois pare o contêiner.
Resposta:
docker container create --name cont4 nginx:alpine
docker container start cont4
docker container stop cont4

15- Crie um contêiner chamado **cont5** com execução em segundo plano com base na imagem do **nginx:alpine**. Acesse o navegador e depois pare o contêiner e acesse o navegador em **localhost.**
Resposta:
docker container run -d --name cont5 nginx:alpine
http://localhost
docker container stop cont5
http://localhost

16- Execute um comando no contêiner **cont5** em execução para obter um **Shell(bash)**
Resposta:
docker exec -i -t cont5 /bin/bash

17 – Verifique os contêineres e imagens existentes e a seguir pare e remova todos os contêineres e imagens que foram criados para este exercício.
Resposta:
Verifica contêineres e imagens existentes
docker images ou docker image ls
docker container ps -a

Exclui imagens e contêineres

docker container prune
docker image prune

Exclui imagens e contêineres
docker container rm $(docker ps -qa)
docker image rm $(docker ps -q) ou docker rmi $(docker images -q)

Nota : ***Para entender o que o $(docker ps -qa) está fazendo, execute somente esse comando no Terminal e veja o retorno.***