

# SISTEMAS OPERACIONAIS

## Criando containers Linux com Docker

Nós já abordamos a questão dos containers. Atualmente a AWS refere-se como um sistema operacional. Vamos instalar e confinar uma aplicação.

Não fazem parte o Docker Compose que é o orquestrador.

Em modo terminal faça:

1. Atualize os pacotes ubuntu;

```
sudo apt update
```

2. Baixe e instalei o Docker para criarmos os containers;

```
sudo apt install docker.io
```

```
marise@marise-notebook:~$ sudo apt install docker.io  
[sudo] senha para marise:
```

Você quer continuar? S

3. Agora vamos usar o gerenciador de processos do linux o systemctl para iniciar o Docker

```
sudo systemctl start docker
```

```
marise@marise-notebook:~$ sudo systemctl start docker
```

```
sudo systemctl enable docker
```

```
marise@marise-notebook:~$ sudo systemctl enable docker  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
```

## SISTEMAS OPERACIONAIS

4. Vamos ver um pouco de serviços:

**systemctl -t service**

5. Voltando ao Docker;

**docker --version**

```
marise@marise-notebook:~$ docker --version
Docker version 19.03.6, build 369ce74a3c
```

6. Vamos listar os nossos containers;

**sudo docker ps**

```
marise@marise-notebook:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Não temos nenhum container criado...

7. Vamos baixar a imagem do mysql 5.7, seguida, vamos iniciar ela em um container;

**sudo docker pull mysql:5.7**

```
marise@marise-notebook:~$ sudo docker pull mysql:5.7
5.7: Pulling from library/mysql
8559a31e96f4: Pull complete
d51ce1c2e575: Pull complete
c2344adc4858: Pull complete
```

```
3dcd2a278b4a: Pull complete
Digest: sha256:32f9d9a069f7a735e28fd44ea944d53c61f990ba71460c5c183e610854ca4854
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
```

## SISTEMAS OPERACIONAIS

8. Vamos listar todas as imagens baixadas;

**sudo docker images**

```
marise@marise-notebook:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	5.7	9cfcce23593a	10 hours ago	448MB

Observe que agora temos a imagem do mysql versão 5.7, tamanho de 448MB.

9. Vamos criar um container, e confinar uma aplicação mysql nele;

**sudo docker run -d -p 3306:3306 --name ContainerBD -e "MYSQL\_DATABASE=banco1" -e "MYSQL\_ROOT\_PASSWORD=urubu100" mysql:5.7**

```
marise@marise-notebook:~$ sudo docker run -d -p 3306:3306 --name ContainerBD -e "MYSQL_DATABASE=banco1" -e "MYSQL_ROOT_PASSWORD=urubu100" mysql:5.7
bfb9d172522104e581074257f07720235e83b4afc976edcbd53448cb1c71c49b
```

10. Vamos tentar ver o que tem no contêiner;

**sudo docker ps**

```
marise@marise-notebook:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAME
bfb9d1725221	mysql:5.7	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	ContainerBD

<u>CONTAINER ID</u>	<u>IMAGE</u>	<u>COMMAND</u>	<u>CREATED</u>	<u>STATUS</u>	<u>PORTS</u>	<u>NAMES</u>
bfb9d1725221	mysql:5.7	"docker-entrypoint.s..."	5 m ago	Up 5 m	0.0.0.0:3306->3306/tcp, 33060/tcp	ContainerBD

## SISTEMAS OPERACIONAIS

11. Vamos ver novamente o que tem no container e a quanto tempo está rodando, veja o tempo

**sudo docker ps -a**

```
marise@marise-notebook:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bfb9d1725221	mysql:5.7	"docker-entrypoint.s..."	5 minutes ago	Up 5 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	ContainerBD

12. Vamos ver as estatísticas do container:

**sudo docker stats ContainerBD**

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
bfb9d1725221	ContainerBD	0.13%	188.6MiB / 7.691GiB	2.39%	8.21kB/OB

### **BLOCK I/O PIDS**

36.9kB / 291MB 27

```
marise@marise-notebook:~$ sudo docker stats ContainerBD
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O
PIDS						
bfb9d1725221	ContainerBD	0.13%	188.6MiB / 7.691GiB	2.39%	8.21kB / OB	36.9kB / 291MB
27						

13. Veja as mudanças, e dê ctrl C para parar;

14. Vamos executar o container que a gente criou com o mysql;

**sudo docker exec -it bfb9d1725221 bash**

No Sublinhado coloque o ID do seu container.

15. Agora já dentro do container, vamos iniciar o mysql;

**root@bfb9d1725221:/# mysql -u root -p**

## SISTEMAS OPERACIONAIS

Enter password: \*\*\*\*\*(senha criada para mysql)

Depois que entrar no Mysql pode dar todos os comandos que você conhece

**SHOW DATABASES;**

```
marise@marise-notebook:~$ sudo docker exec -it bfb9d1725221 bash
root@bfb9d1725221:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| banco1 |
| mysql |
| performance_schema |
| sys |
+-----+
```

16. saia do container:

**exit** // para terminar mysql

**exit** // para encerrar o bash

```
mysql> exit
Bye
root@bfb9d1725221:/# exit
exit
```

## SISTEMAS OPERACIONAIS

17. Você pode stopar o serviço, lembre-se que está no banco dentro do container;

**sudo docker stop (ID do Container)**

```
marise@marise-notebook:~$ sudo docker stop bfb9d1725221  
bfb9d1725221
```

Veja como a aplicação está apartada!

18. Você pode startar o container novamente;

**sudo docker start (ID do Container)**

Para obter o ID do Container, você pode usar o comando,

**sudo docker ps -a**

### AVALIAÇÃO

Você deve prover um serviço de container que acesse o banco e crie três tabelas que estão relacionadas com o projeto de PI (métricas CPU, métricas Memória, métricas disco), usando só e somente só o container.

Observação: os comandos de mysql funcionam todos via comand line

\*Atividade stand alone