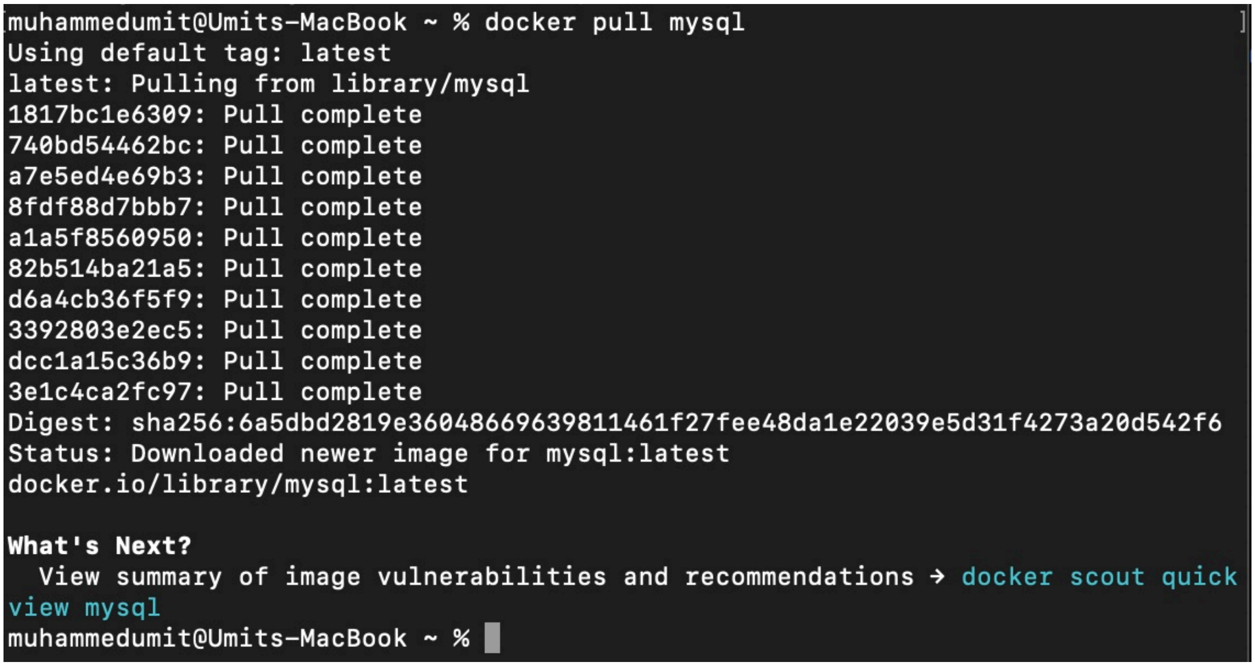
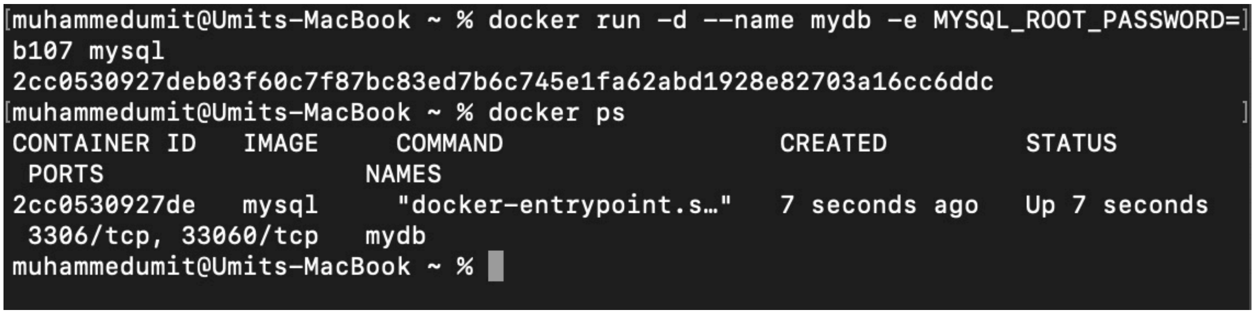
Docker Volume Practice

In this practice, I pull a Mysql Database and run it as a container.

In order to do that, the code will be ‘docker pull mysql:latest’ and I’ll run the container with the code ‘docker container run —detach —name mydb -e MYSQL_ROOT_PASSWORD=b107 mysql:latest”



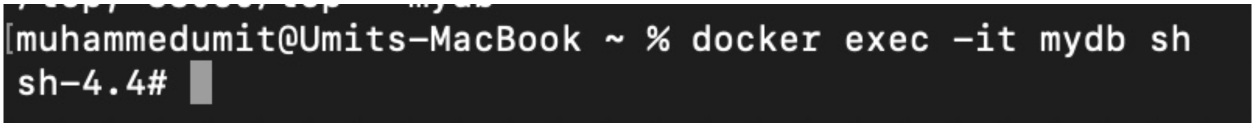
```
[muhammedumit@Umits-MacBook ~ % docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
1817bc1e6309: Pull complete
740bd54462bc: Pull complete
a7e5ed4e69b3: Pull complete
8fdf88d7bbb7: Pull complete
a1a5f8560950: Pull complete
82b514ba21a5: Pull complete
d6a4cb36f5f9: Pull complete
3392803e2ec5: Pull complete
dcc1a15c36b9: Pull complete
3e1c4ca2fc97: Pull complete
Digest: sha256:6a5dbd2819e36048669639811461f27fee48da1e22039e5d31f4273a20d542f6
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quick
view mysql
muhammedumit@Umits-MacBook ~ %
```



```
[muhammedumit@Umits-MacBook ~ % docker run -d --name mydb -e MYSQL_ROOT_PASSWORD=
b107 mysql
2cc0530927deb03f60c7f87bc83ed7b6c745e1fa62abd1928e82703a16cc6ddc
[muhammedumit@Umits-MacBook ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS
 PORTS                 NAMES
2cc0530927de   mysql     "docker-entrypoint.s…"   7 seconds ago   Up 7 seconds
 3306/tcp, 33060/tcp   mydb
muhammedumit@Umits-MacBook ~ %
```

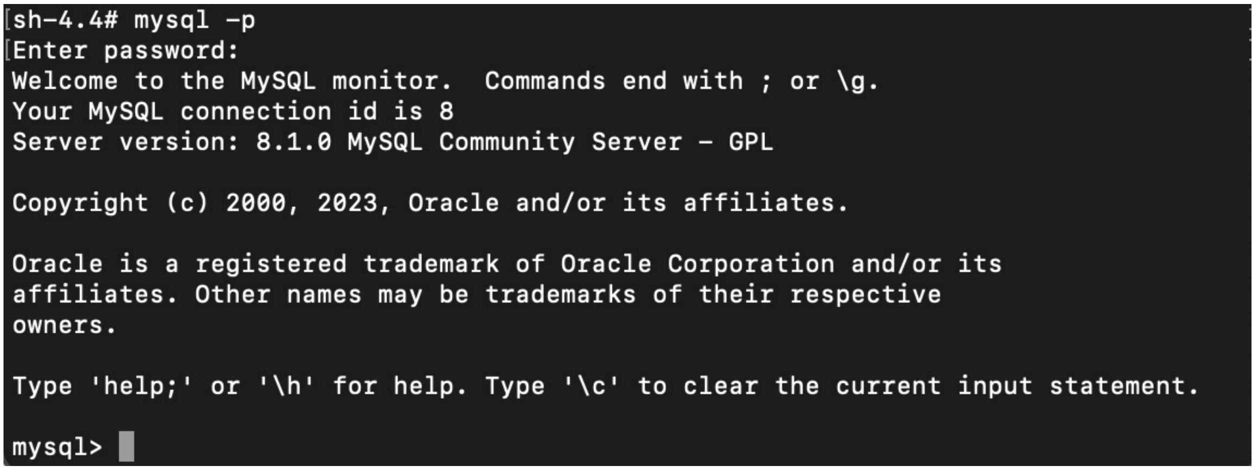
Modify the database

I enter the mysql container by using code “ docker exec -it {containerID} sh. Exec code allows you to execute commands within a running container without the need to create a new container instance.



```
[muhammedumit@Umits-MacBook ~ % docker exec -it mydb sh
sh-4.4#
```

Now we’re inside of the mysql database and enter the database client using the provided password then we’ll see the databases and tables by using code ‘mysql -p’.

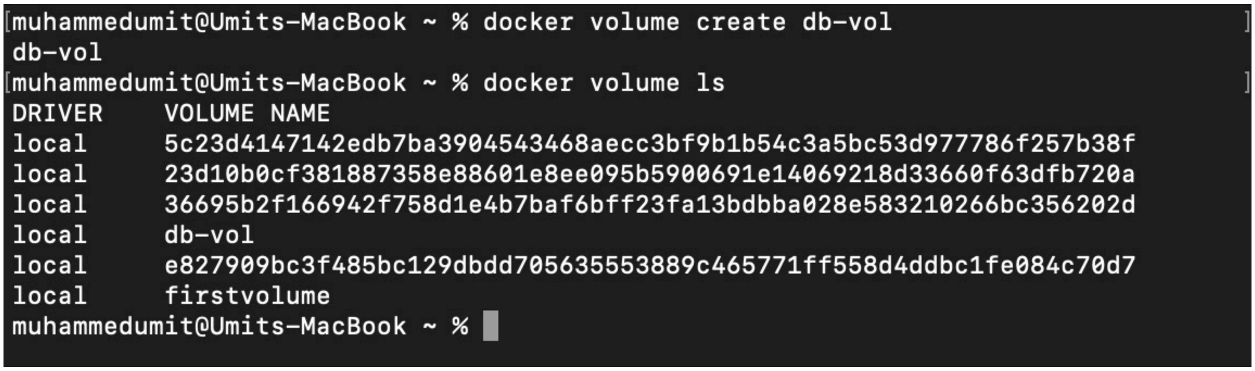


```
sh-4.4# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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> 
```

Containers are stateless, they do not store datas. The database and files are not there.

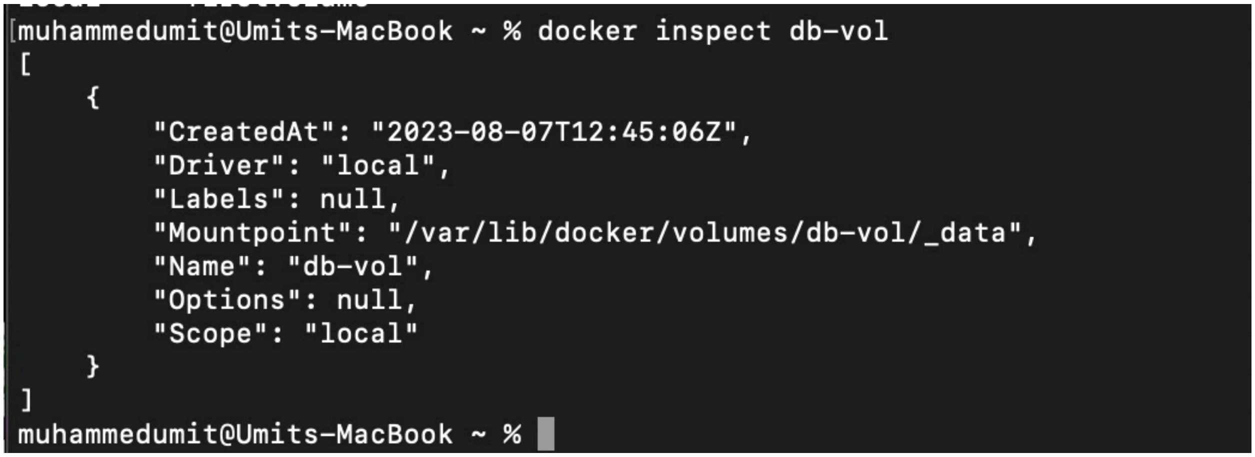
Run Mysql database using volumes

I create a volume by using code ‘docker volume create db-vol’.



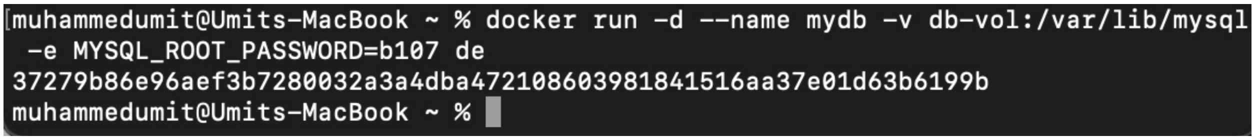
```
muhammedumit@Umits-MacBook ~ % docker volume create db-vol
db-vol
muhammedumit@Umits-MacBook ~ % docker volume ls
DRIVER    VOLUME NAME
local     5c23d4147142edb7ba3904543468aecc3bf9b1b54c3a5bc53d977786f257b38f
local     23d10b0cf381887358e88601e8ee095b5900691e14069218d33660f63dfb720a
local     36695b2f166942f758d1e4b7baf6bff23fa13bdbba028e583210266bc356202d
local     db-vol
local     e827909bc3f485bc129dbdd705635553889c465771ff558d4ddbc1fe084c70d7
local     firstvolume
muhammedumit@Umits-MacBook ~ % 
```

In order to inspect the volume ‘docker inspect volumename/volumeID’



```
muhammedumit@Umits-MacBook ~ % docker inspect db-vol
[
    {
        "CreatedAt": "2023-08-07T12:45:06Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/db-vol/_data",
        "Name": "db-vol",
        "Options": null,
        "Scope": "local"
    }
]
muhammedumit@Umits-MacBook ~ % 
```

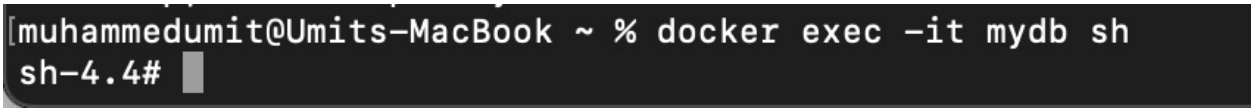
Now let’s run the database container, this time mapping it to a local volume we just created. The code ‘docker run -d —name mydb -v db-vol:var/lib/mysql -e MYSWL_ROOT_PASSWORD=b107 mysql’



```
[muhammedumit@Umits-MacBook ~ % docker run -d --name mydb -v db-vol:/var/lib/mysql
 -e MYSQL_ROOT_PASSWORD=b107 de
37279b86e96aef3b7280032a3a4dba472108603981841516aa37e01d63b6199b
muhammedumit@Umits-MacBook ~ %
```

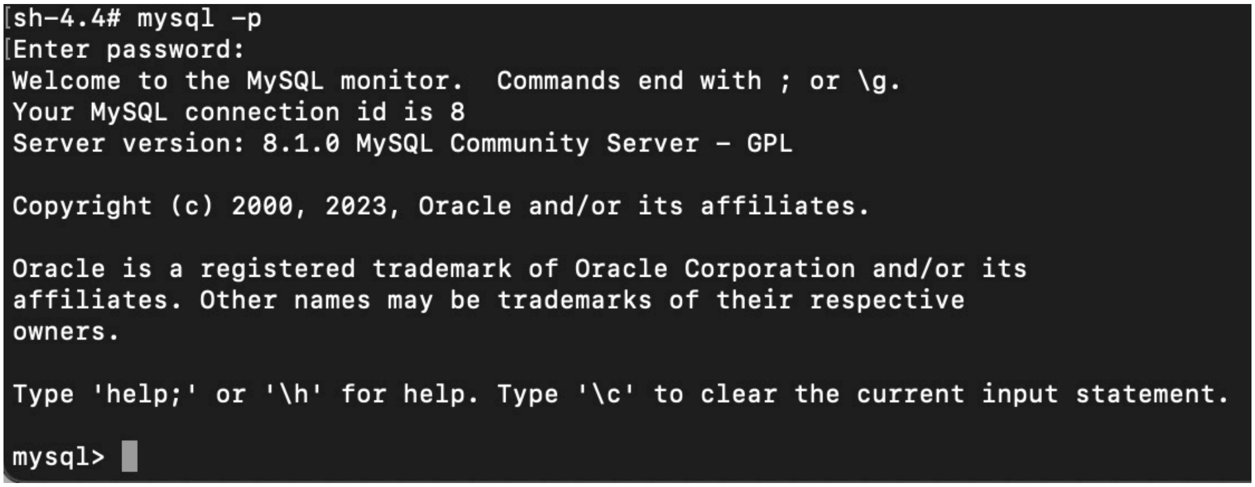
Mapping is local folder inside the container -v local:container. We connected the db-vol to the container.

Let’s connect to the container by using code ‘docker exec -it {containerID} sh’



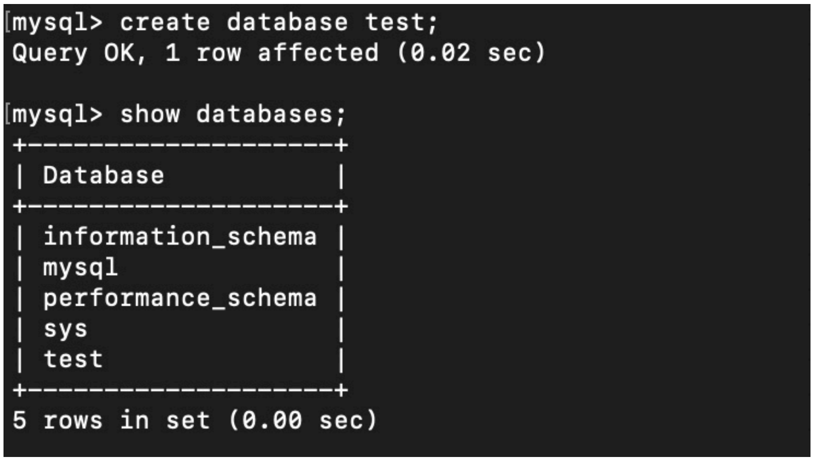
```
[muhammedumit@Umits-MacBook ~ % docker exec -it mydb sh
sh-4.4#
```

Now I enter the database client using the provided password then see the databases and tables by using code ‘mysql -p’.

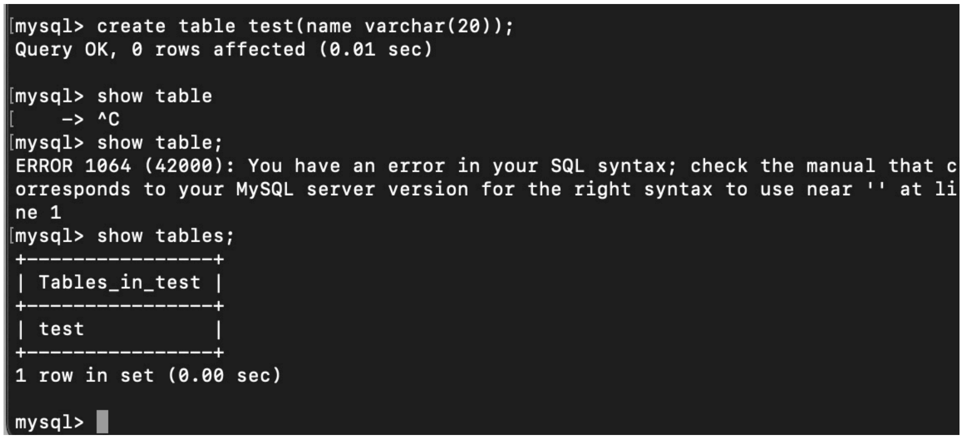


```
[sh-4.4# mysql -p
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

I create a database and a table by using code ‘ create database test;’ and ‘create table test(name varchar(20));’

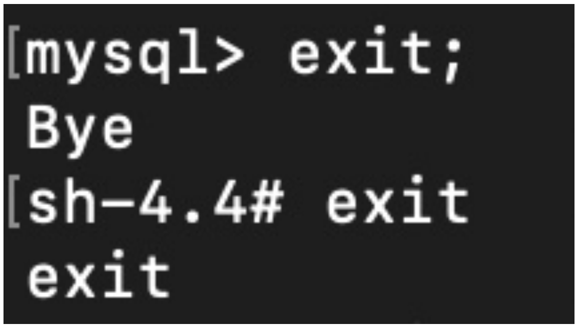


```
[mysql> create database test;
Query OK, 1 row affected (0.02 sec)

[mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.00 sec)
```



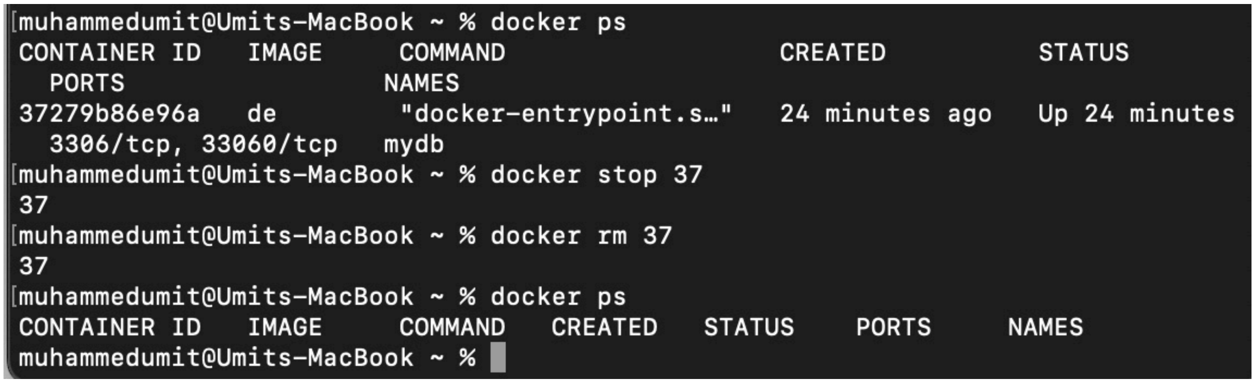
```
[mysql> create table test(name varchar(20));
 Query OK, 0 rows affected (0.01 sec)

[mysql> show table
[    -> ^C
[mysql> show table;
 ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that c
 orresponds to your MySQL server version for the right syntax to use near '' at li
 ne 1
[mysql> show tables;
 +----------------+
 | Tables_in_test |
 +----------------+
 | test           |
 +----------------+
 1 row in set (0.00 sec)

 mysql> 
```

Exit the database and exit mysql



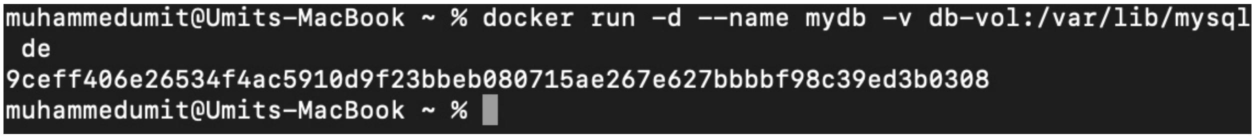
```
[mysql> exit;
 Bye
[sh-4.4# exit
 exit
```

Exit, stop and remove the container by using code ‘docker stop containername/containerID’ and ‘docker rm containername/containerID’



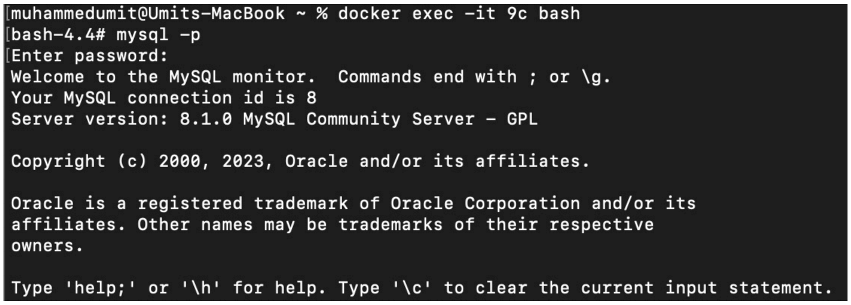
```
[muhammedumit@Umits-MacBook ~ % docker ps
 CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
   PORTS                 NAMES
 37279b86e96a   de        "docker-entrypoint.s…"   24 minutes ago   Up 24 minutes
   3306/tcp, 33060/tcp   mydb
[muhammedumit@Umits-MacBook ~ % docker stop 37
 37
[muhammedumit@Umits-MacBook ~ % docker rm 37
 37
[muhammedumit@Umits-MacBook ~ % docker ps
 CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
 muhammedumit@Umits-MacBook ~ % 
```

Now, I run another mysql container by using ‘ docker run -d —name mydb -v db-vol:/var/lib/mysql {mysql ID}’

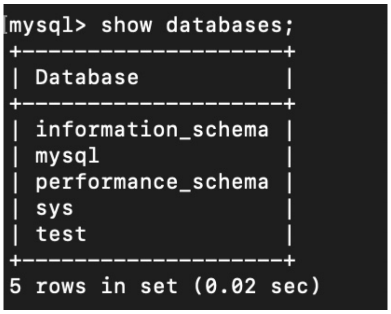


```
muhammedumit@Umits-MacBook ~ % docker run -d --name mydb -v db-vol:/var/lib/mysql
 de
9ceff406e26534f4ac5910d9f23bbeb080715ae267e627bbbbf98c39ed3b0308
muhammedumit@Umits-MacBook ~ % 
```

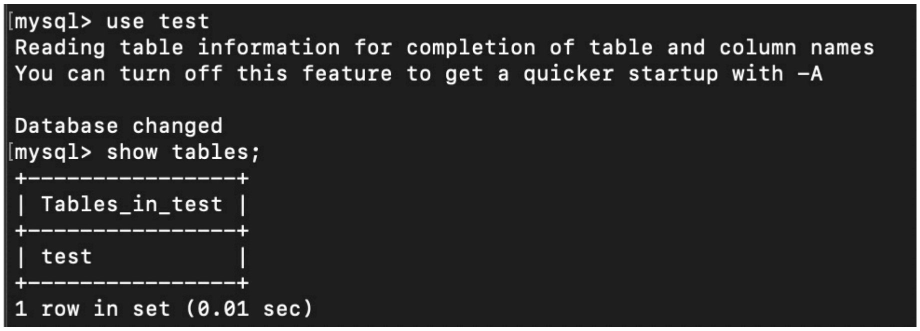
I enter the container and see the databases by using code ‘docker exec -it {containerID} sh or bash’ and ‘mysql -p’ and ‘show databases;’



```
[muhammedumit@Umits-MacBook ~ % docker exec -it 9c bash
[bash-4.4# mysql -p
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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 |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.02 sec)
```



```
[mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
[mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| test           |
+----------------+
1 row in set (0.01 sec)
```

My database exists, it is there and this is because we map the container’s folder to local folder, if you delete the volume locally, the data will be lost.