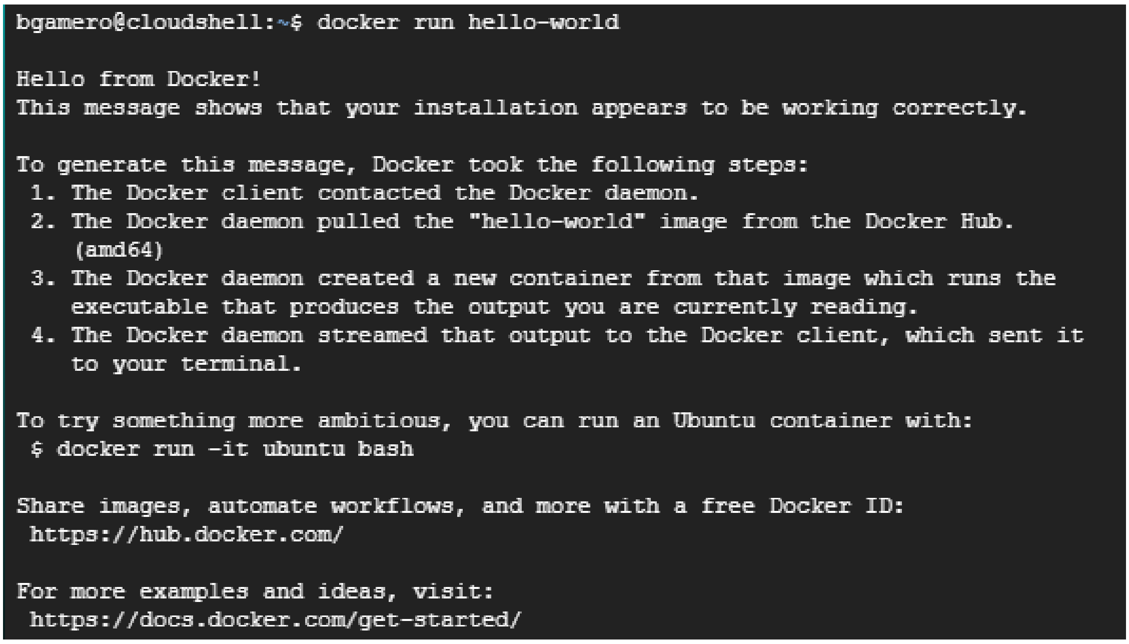
# ASSI Dockerized Database

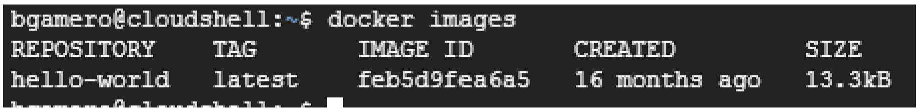
PARTE I: Docker

A. Use the Google cloud shell (https://console.cloud.google.com/cloudshell?pli=1) to launch a "hello-world" container.



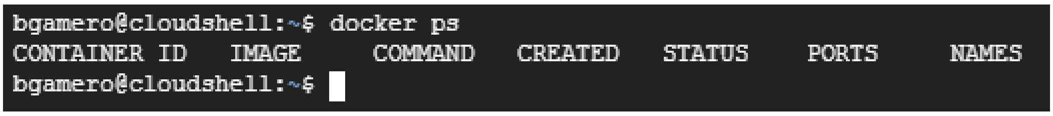
```
bgamero@cloudshell:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

B. Shows the list of images in the local repository.



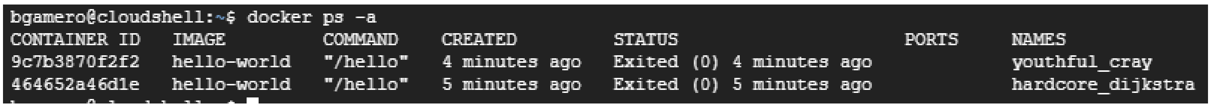
```
bgamero@cloudshell:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   16 months ago   13.3kB
```

C. Shows the list of running containers.



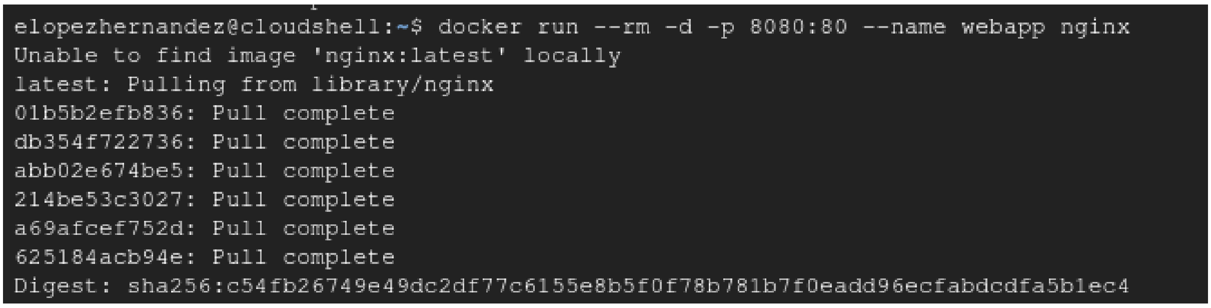
```
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
bgamero@cloudshell:~$
```

D. Shows the list of all containers, including those that are not running.

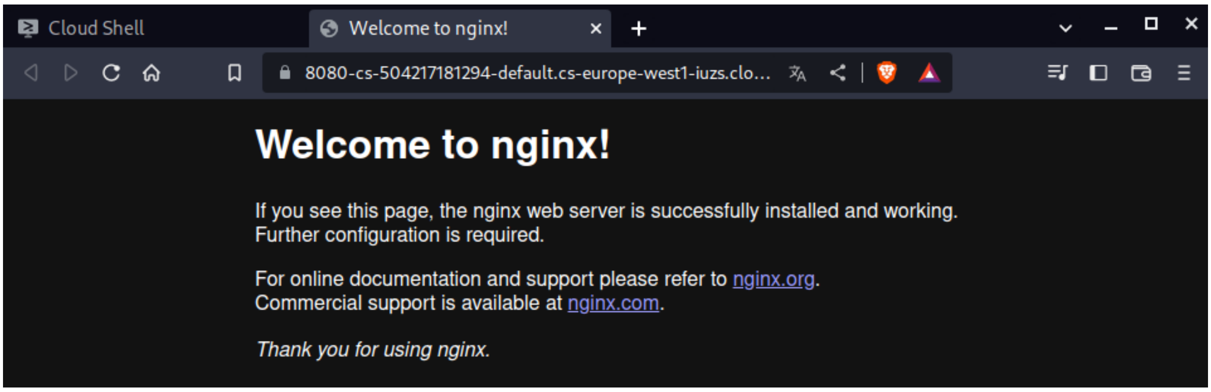


```
bgamero@cloudshell:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
9c7b3870f2f2   hello-world   "/hello"   4 minutes ago   Exited (0) 4 minutes ago             youthful_cray
464652a46d1e   hello-world   "/hello"   5 minutes ago   Exited (0) 5 minutes ago             hardcore_dijkstra
```

E. Run a dockerized nginx. Run it in "detached mode" and give it a name.



```
elopezhernandez@cloudshell:~$ docker run --rm -d -p 8080:80 --name webapp nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
01b5b2efb836: Pull complete
db354f722736: Pull complete
abb02e674be5: Pull complete
214be53c3027: Pull complete
a69afcef752d: Pull complete
625184acb94e: Pull complete
Digest: sha256:c54fb26749e49dc2df77c6155e8b5f0f78b781b7f0eadd96ecfabdcdfa5b1ec4
```



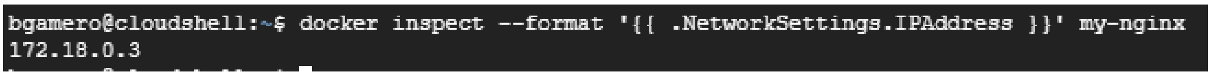
## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

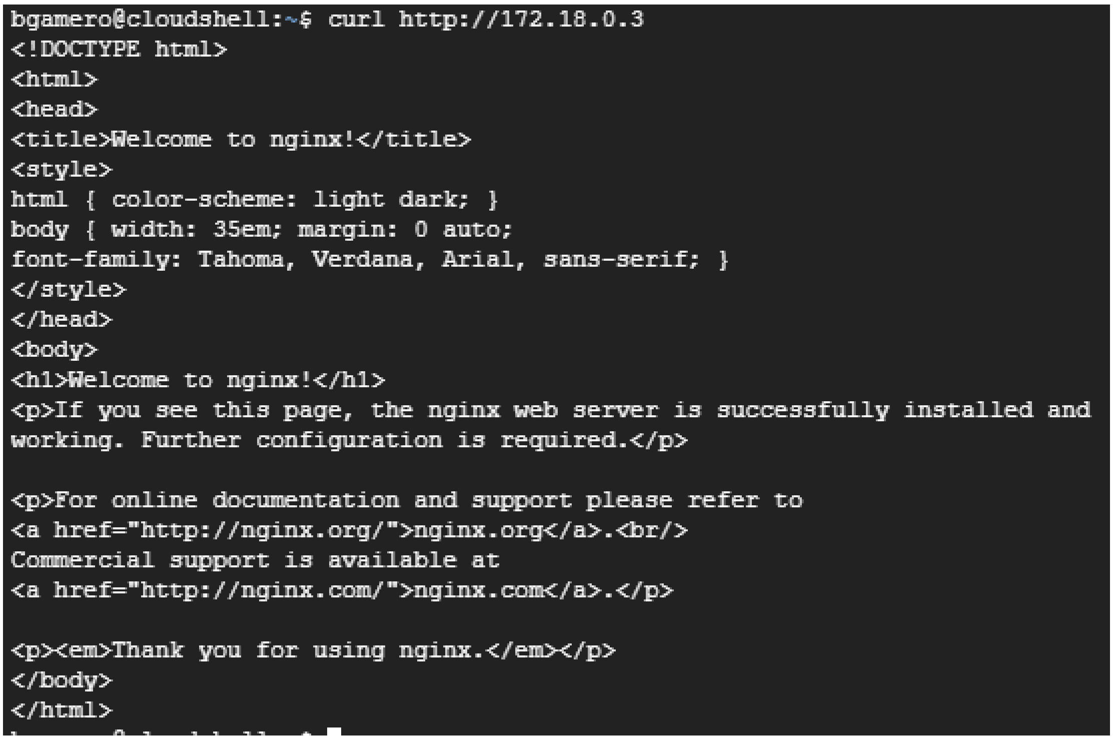
*Thank you for using nginx.*

F. Find out the address of the nginx server.



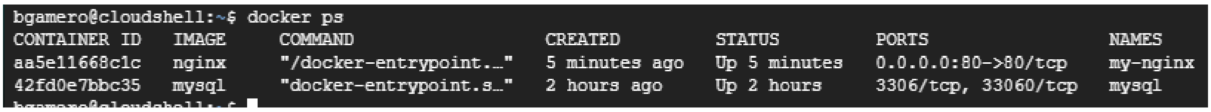
```
bgamero@cloudshell:~$ docker inspect --format '{{ .NetworkSettings.IPAddress }}' my-nginx
172.18.0.3
```

G. Use curl to connect to the server and verify that it is working.



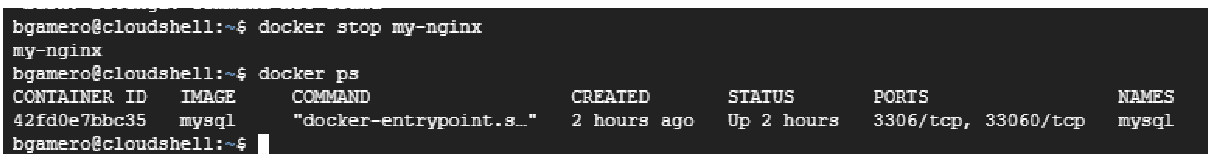
```
bgamero@cloudshell:~$ curl http://172.18.0.3
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

H. Look at the list of running containers.



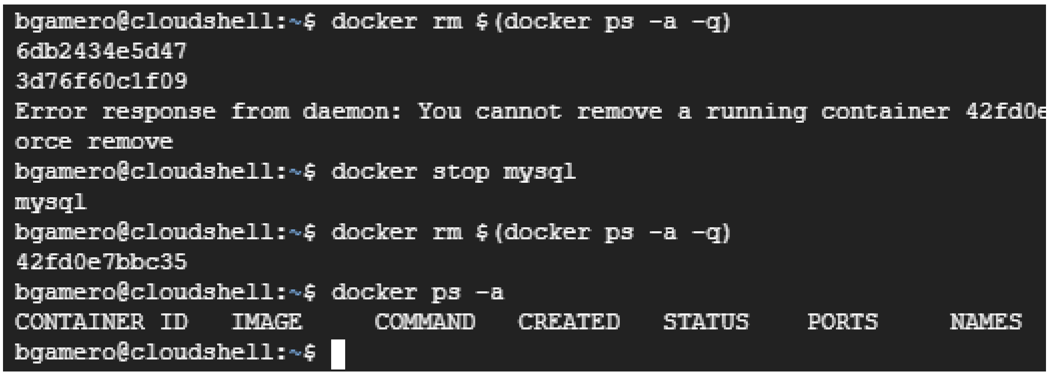
```
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                 NAMES
aa5e11668c1c   nginx     "/docker-entrypoint.…"   5 minutes ago   Up 5 minutes   0.0.0.0:80->80/tcp    my-nginx
42fd0e7bbc35   mysql     "docker-entrypoint.s…"   2 hours ago     Up 2 hours     3306/tcp, 33060/tcp   mysql
bgamero@cloudshell:~$
```

I. Stop the nginx container.



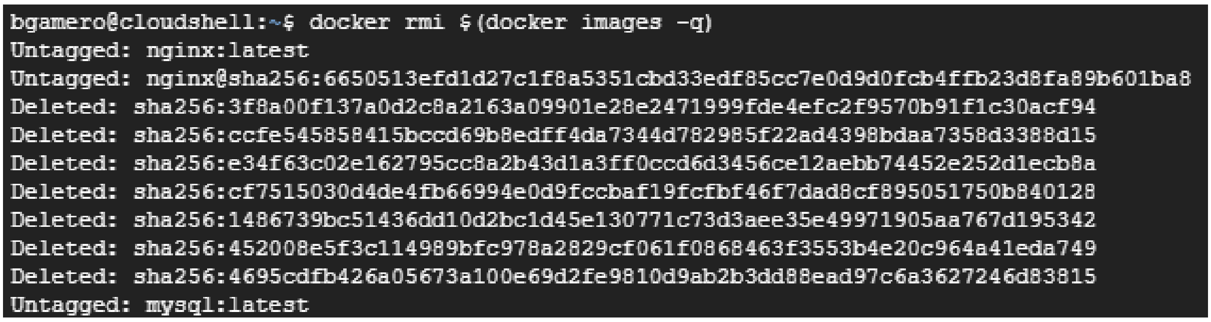
```
bgamero@cloudshell:~$ docker stop my-nginx
my-nginx
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED       STATUS       PORTS                 NAMES
42fd0e7bbc35   mysql     "docker-entrypoint.s…"   2 hours ago   Up 2 hours   3306/tcp, 33060/tcp   mysql
bgamero@cloudshell:~$
```

J. Delete all containers.

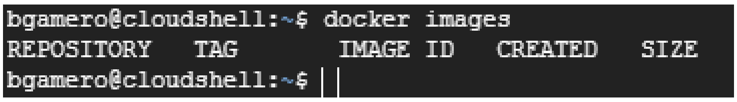


```
bgamero@cloudshell:~$ docker rm $(docker ps -a -q)
6db2434e5d47
3d76f60c1f09
Error response from daemon: You cannot remove a running container 42fd0e
orce remove
bgamero@cloudshell:~$ docker stop mysql
mysql
bgamero@cloudshell:~$ docker rm $(docker ps -a -q)
42fd0e7bbc35
bgamero@cloudshell:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
bgamero@cloudshell:~$
```

K. Delete all images.



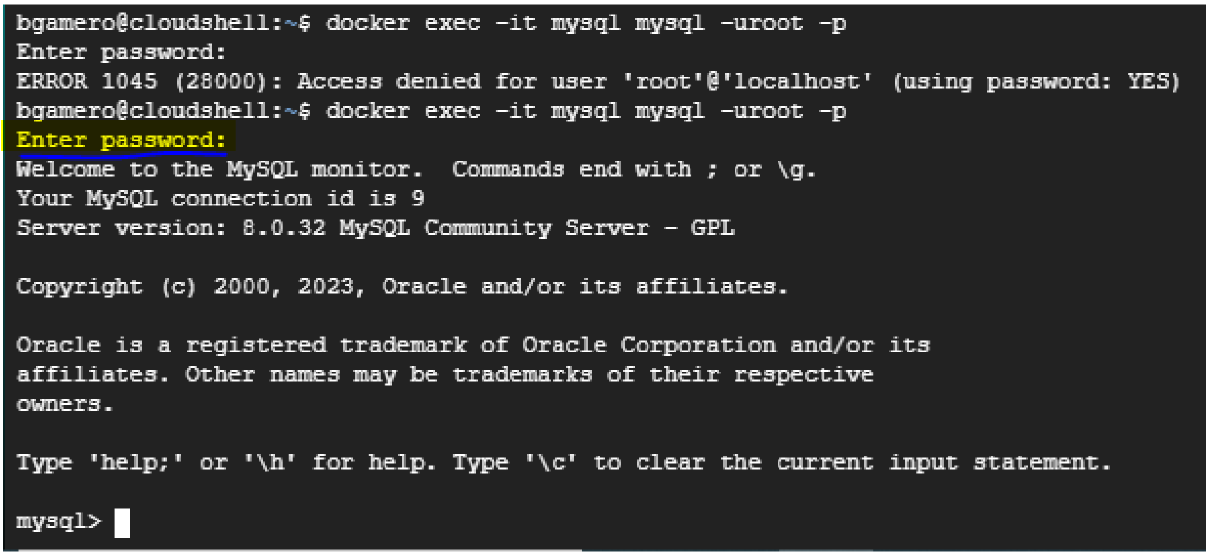
```
bgamero@cloudshell:~$ docker rmi $(docker images -q)
Untagged: nginx:latest
Untagged: nginx@sha256:6650513efd1d27c1f8a5351cbd33edf85cc7e0d9d0fcb4ffb23d8fa89b601ba8
Deleted: sha256:3f8a00f137a0d2c8a2163a09901e28e2471999fde4efc2f9570b91f1c30acf94
Deleted: sha256:ccfe545858415bccd69b8edff4da7344d782985f22ad4398bdaa7358d3388d15
Deleted: sha256:e34f63c02e162795cc8a2b43d1a3ff0ccd6d3456ce12aebb74452e252d1ecb8a
Deleted: sha256:cf7515030d4de4fb66994e0d9fccbaf19fcfbf46f7dad8cf895051750b840128
Deleted: sha256:1486739bc51436dd10d2bc1d45e130771c73d3aee35e49971905aa767d195342
Deleted: sha256:452008e5f3c114989bfc978a2829cf061f0868463f3553b4e20c964a41eda749
Deleted: sha256:4695cdfb426a05673a100e69d2fe9810d9ab2b3dd88ead97c6a3627246d83815
Untagged: mysql:latest
```



```
bgamero@cloudshell:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
bgamero@cloudshell:~$
```

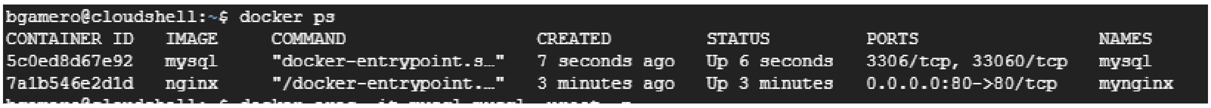
PART II: Docker database

L. Run a docker mysql database in detached mode. Use an environment variable to set the password. Connect to the DBMS.



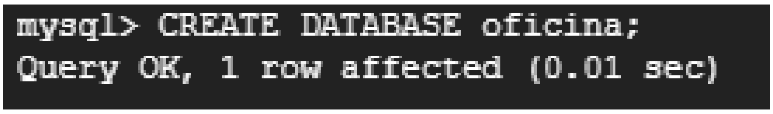
```
bgamero@cloudshell:~$ docker exec -it mysql mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
bgamero@cloudshell:~$ docker exec -it mysql mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32 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> 
```

M. Find the IP address of the database docker.



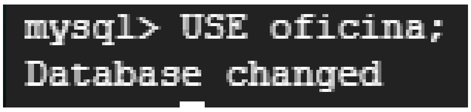
```
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                 NAMES
5c0ed8d67e92   mysql     "docker-entrypoint.s…"   7 seconds ago   Up 6 seconds   3306/tcp, 33060/tcp   mysql
7a1b546e2d1d   nginx     "/docker-entrypoint.…"   3 minutes ago   Up 3 minutes   0.0.0.0:80->80/tcp    mynginx
```

N. Create a database.



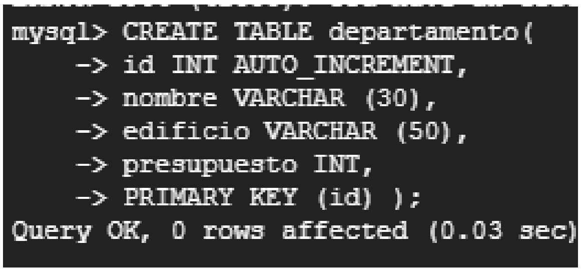
```
mysql> CREATE DATABASE oficina;
Query OK, 1 row affected (0.01 sec)
```

O. Use the database.



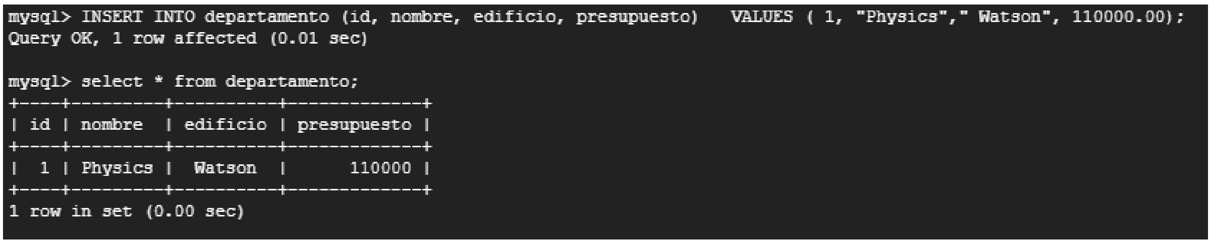
```
mysql> USE oficina;
Database changed
```

P. Create a table 'department' with the columns 'id', 'name', 'building' and 'budget'.



```
mysql> CREATE TABLE departamento(
    -> id INT AUTO_INCREMENT,
    -> nombre VARCHAR (30),
    -> edificio VARCHAR (50),
    -> presupuesto INT,
    -> PRIMARY KEY (id) );
Query OK, 0 rows affected (0.03 sec)
```

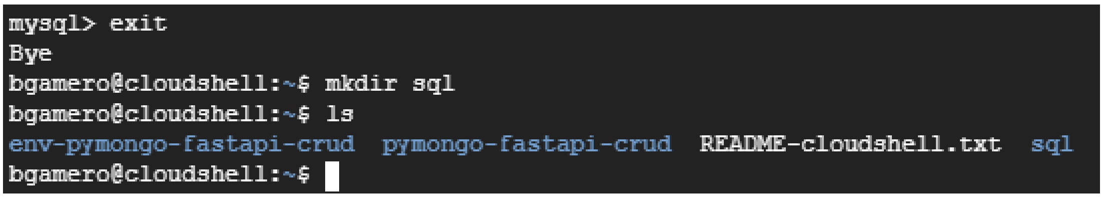
Q. Insert a row into the table with values 'Physical', 'Watson', '110000.00'.



```
mysql> INSERT INTO departamento (id, nombre, edificio, presupuesto)   VALUES ( 1, "Physics"," Watson", 110000.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from departamento;
+----+---------+----------+-------------+
| id | nombre  | edificio | presupuesto |
+----+---------+----------+-------------+
|  1 | Physics |  Watson  |      110000 |
+----+---------+----------+-------------+
1 row in set (0.00 sec)
```

R. Exit the mysql client.

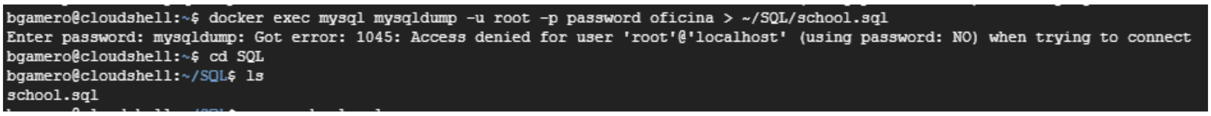
S. Create a directory called sql.



```
mysql> exit
Bye
bgamero@cloudshell:~$ mkdir sql
bgamero@cloudshell:~$ ls
env-pymongo-fastapi-crud  pymongo-fastapi-crud  README-cloudshell.txt  sql
bgamero@cloudshell:~$ 
```

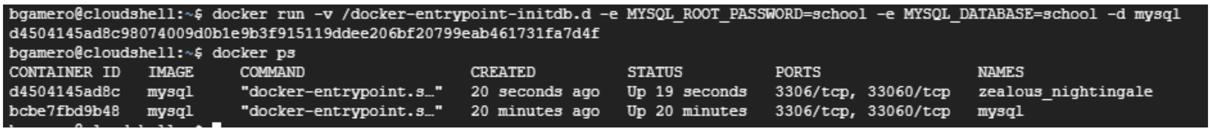
T. Use mysqldump to dump your database into a file called school.sql in the directory created in the previous step.

* Even though an error/alert came out, the copy was created correctly



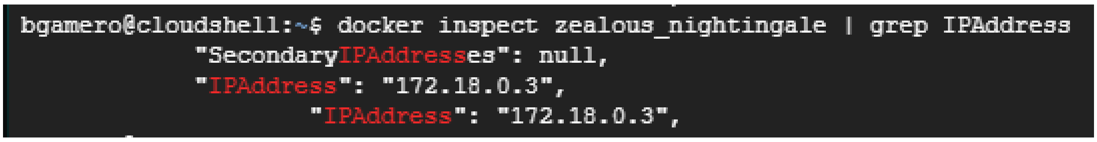
```
bgamero@cloudshell:~$ docker exec mysql mysqldump -u root -p password oficina > ~/SQL/school.sql
Enter password: mysqldump: Got error: 1045: Access denied for user 'root'@'localhost' (using password: NO) when trying to connect
bgamero@cloudshell:~$ cd SQL
bgamero@cloudshell:~/SQL$ ls
school.sql
```

U. Create a new container with a volume connecting to the directory you created with /docker-entrypoint-initdb.d .
This will automatically populate your new container's database. Run it in detached mode. Use the -v flag to map your local directory to the /docker-entrypoint-initdb.d directory. Example: docker run -v /home/jbarcelo/sql:/docker-entrypoint-initdb.d -e MYSQL_ROOT_PASSWORD=my-secret-pw -e MYSQL_DATABASE=school -d mysql



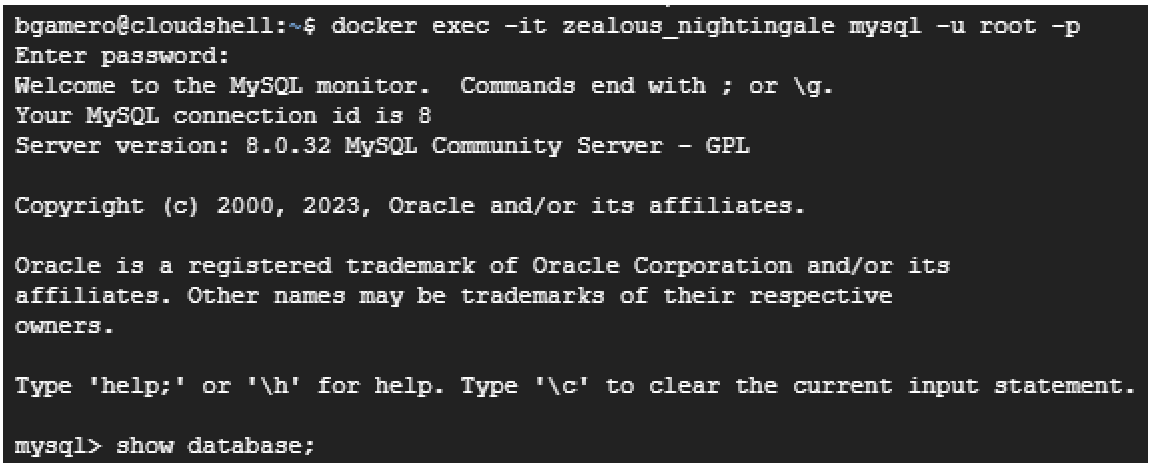
```
bgamero@cloudshell:~$ docker run -v /docker-entrypoint-initdb.d -e MYSQL_ROOT_PASSWORD=school -e MYSQL_DATABASE=school -d mysql
d4504145ad8c98074009d0b1e9b3f915119ddee206bf20799eab461731fa7d4f
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                 NAMES
d4504145ad8c   mysql     "docker-entrypoint.s…"   20 seconds ago   Up 19 seconds   3306/tcp, 33060/tcp   zealous_nightingale
bcbe7fbd9b48   mysql     "docker-entrypoint.s…"   20 minutes ago   Up 20 minutes   3306/tcp, 33060/tcp   mysql
```

V. Find the IP address of the new container.



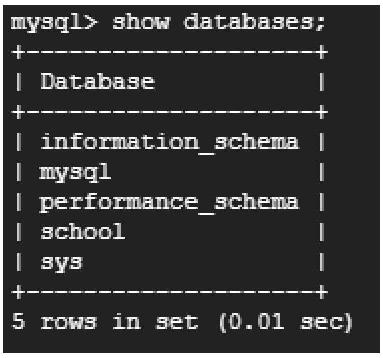
```
bgamero@cloudshell:~$ docker inspect zealous_nightingale | grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.18.0.3",
                    "IPAddress": "172.18.0.3",
```

W. Use the mysql client to connect to the new container.

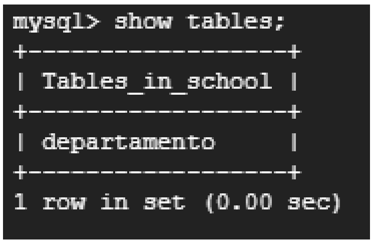


```
bgamero@cloudshell:~$ docker exec -it zealous_nightingale mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32 MySQL Community Server - GPL

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 database;
```

X. Verify that the table and its contents are available.



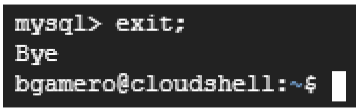
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| school             |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```



```
mysql> show tables;
+------------------+
| Tables_in_school |
+------------------+
| departamento     |
+------------------+
1 row in set (0.00 sec)
```

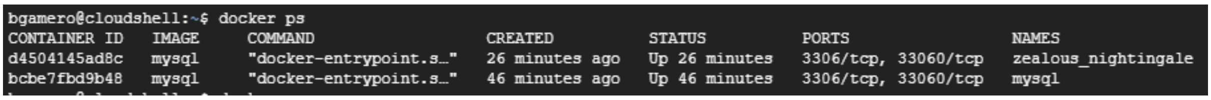
Edgar Lopez Hernande
Belen Gamero Garcia

Y. Exit the mysql client.



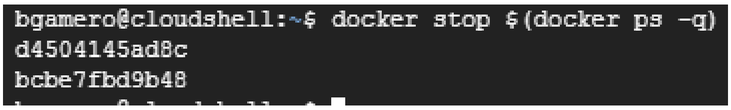
```
mysql> exit;
Bye
bgamero@cloudshell:~$
```

Z. List all running containers.



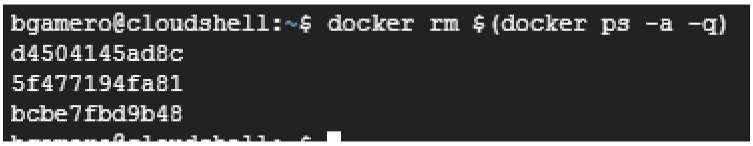
```
bgamero@cloudshell:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                 NAMES
d4504145ad8c   mysql     "docker-entrypoint.s…"   26 minutes ago   Up 26 minutes   3306/tcp, 33060/tcp   zealous_nightingale
bcbe7fbd9b48   mysql     "docker-entrypoint.s…"   46 minutes ago   Up 46 minutes   3306/tcp, 33060/tcp   mysql
```

AA. Stop all containers running.



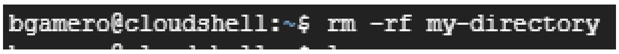
```
bgamero@cloudshell:~$ docker stop $(docker ps -q)
d4504145ad8c
bcbe7fbd9b48
```

BB. Delete the containers.



```
bgamero@cloudshell:~$ docker rm $(docker ps -a -q)
d4504145ad8c
5f477194fa81
bcbe7fbd9b48
```

CC. Delete the directory you created and its contents.



```
bgamero@cloudshell:~$ rm -rf my-directory
```

*Replace 'my-directory' with the name of the directory you want to remove.

Please provide feedback on the difficulties you encountered and the things you learned.

- The most notorious difficulty has been the lack of knowledge we have about docker.