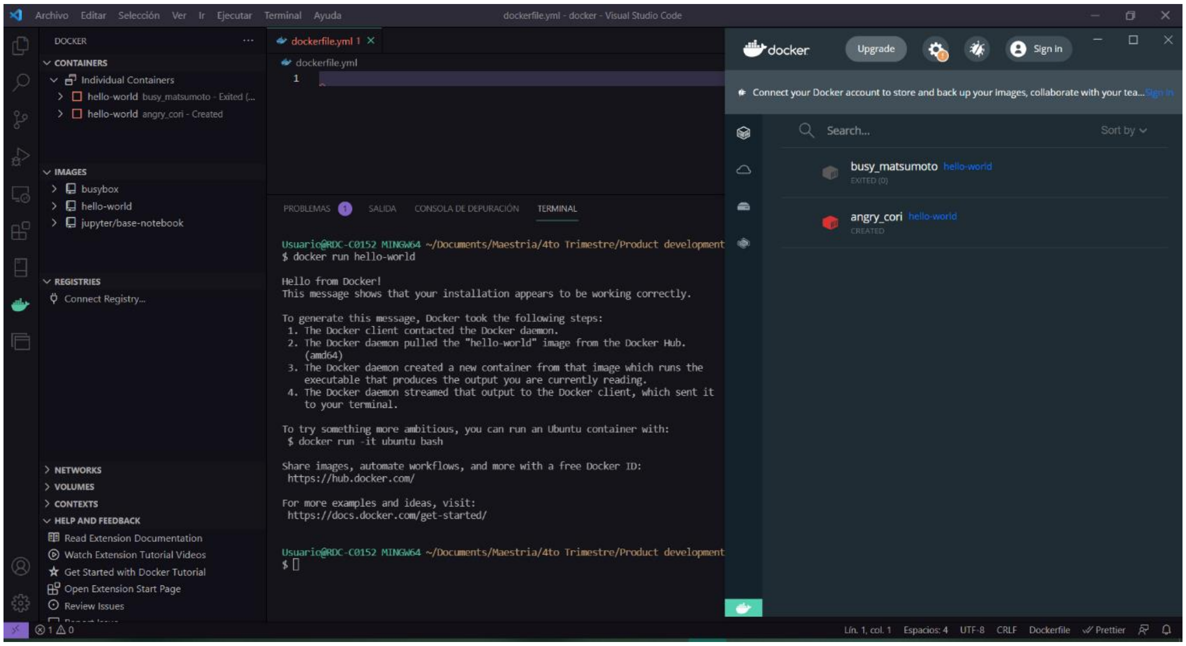
DOCKER

1. **docker run hello-world,** validando que el servidor de Docker está ejecutándose localmente.

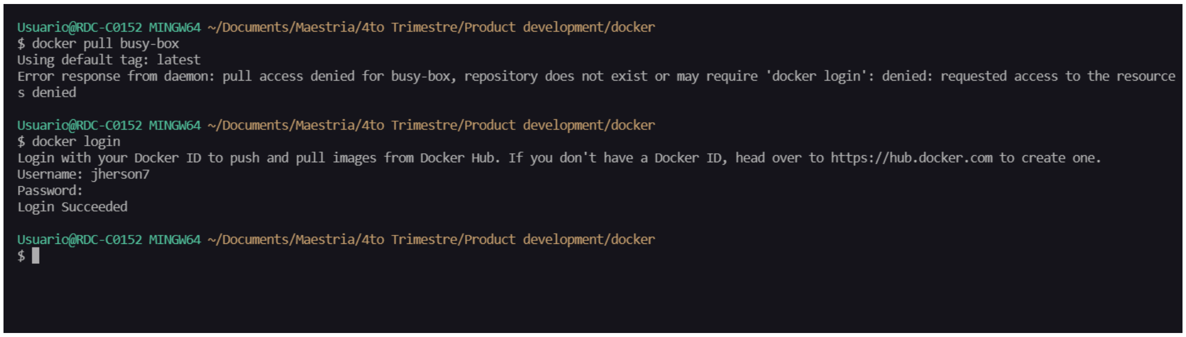


Connect your Docker account to store and back up your images, collaborate with your tea... Sign in

```
Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development
$ 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/


Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development
$ 
```

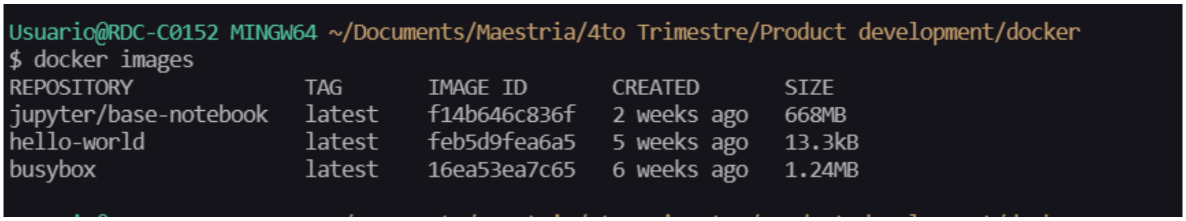
2. Docker pull busy-box

   En este caso no pude bajar la imagen ya que me pedía registro en Docker hub, una vez registrado pude trabajar.



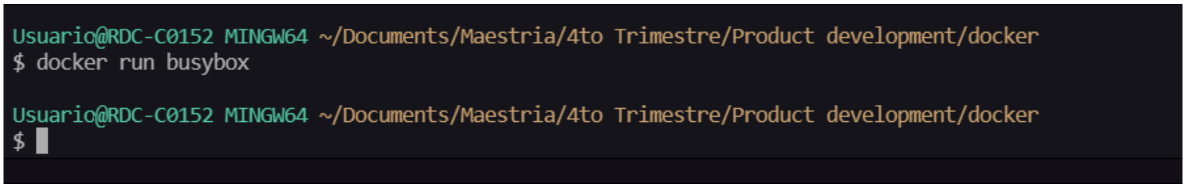
```
Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker pull busy-box
Using default tag: latest
Error response from daemon: pull access denied for busy-box, repository does not exist or may require 'docker login': denied: requested access to the resource
s denied

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: jherson7
Password:
Login Succeeded

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ 
```

3. Viendo las imágenes locales con docker images



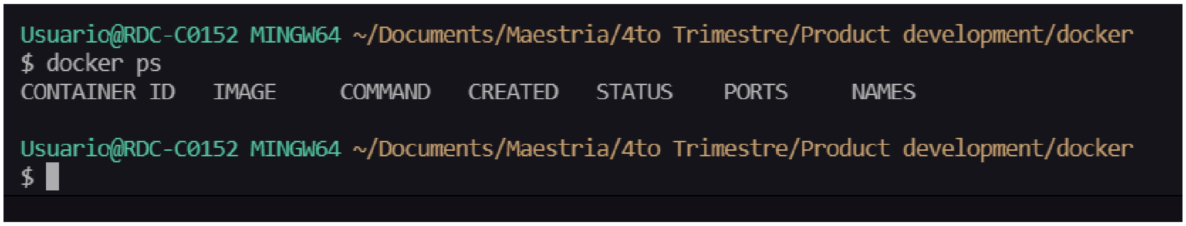
```
Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED       SIZE
jupyter/base-notebook   latest    f14b646c836f   2 weeks ago   668MB
hello-world             latest    feb5d9fea6a5   5 weeks ago   13.3kB
busybox                 latest    16ea53ea7c65   6 weeks ago   1.24MB
```

4. Docker run busybox



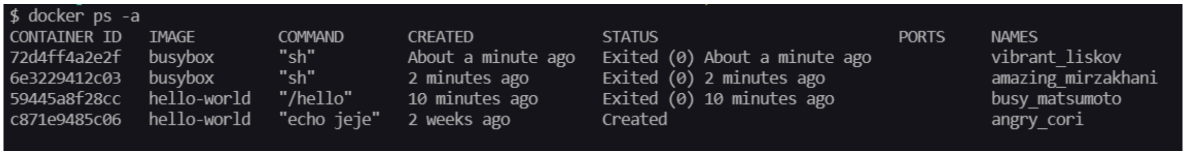
```
Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker run busybox

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$
```

5. Para ver los contendores activos usamos Docker ps



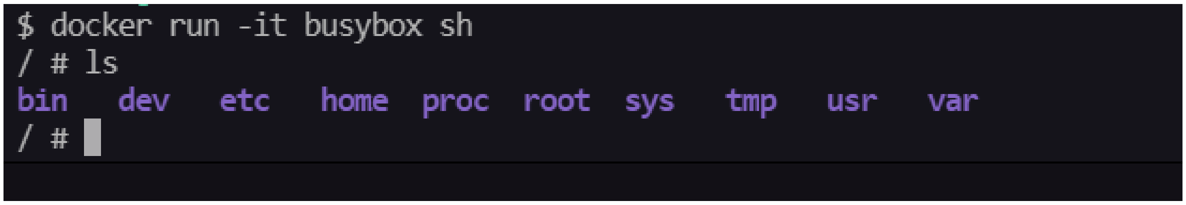
```
Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$
```

6. Para ver los contenedores ejecutados detenidos usamos Docker ps -a (a=all)



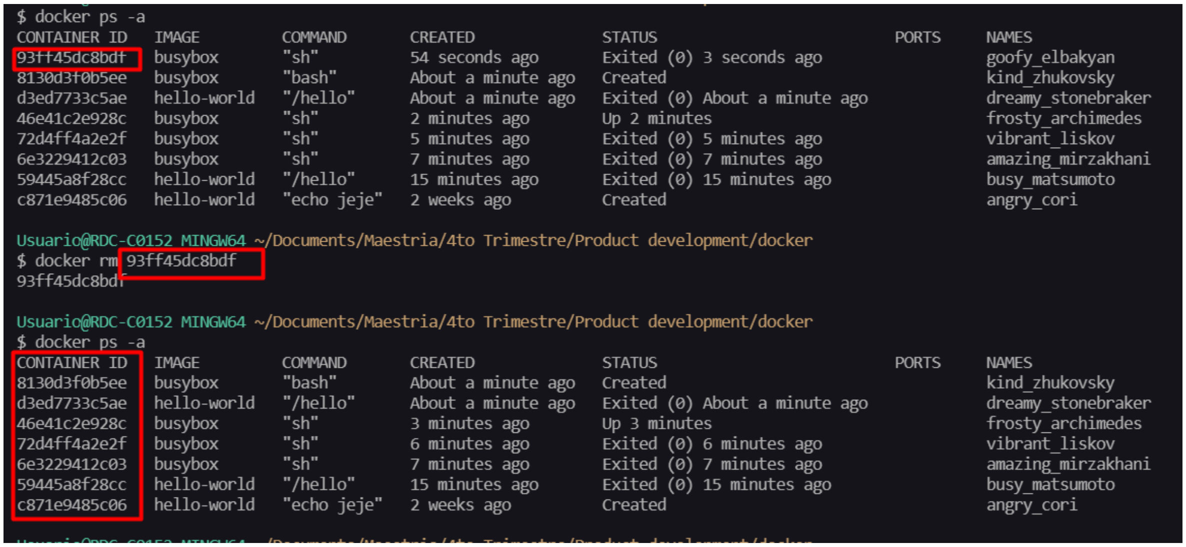
```
$ docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED              STATUS                          PORTS     NAMES
72d4ff4a2e2f   busybox       "sh"          About a minute ago   Exited (0) About a minute ago             vibrant_liskov
6e3229412c03   busybox       "sh"          2 minutes ago        Exited (0) 2 minutes ago                  amazing_mirzakhani
59445a8f28cc   hello-world   "/hello"      10 minutes ago       Exited (0) 10 minutes ago                 busy_matsumoto
c871e9485c06   hello-world   "echo jeje"   2 weeks ago          Created                                   angry_cori
```

7. Para entrar al modo iteractivo de un contenedor usamos Docker run -it <container-name> sh



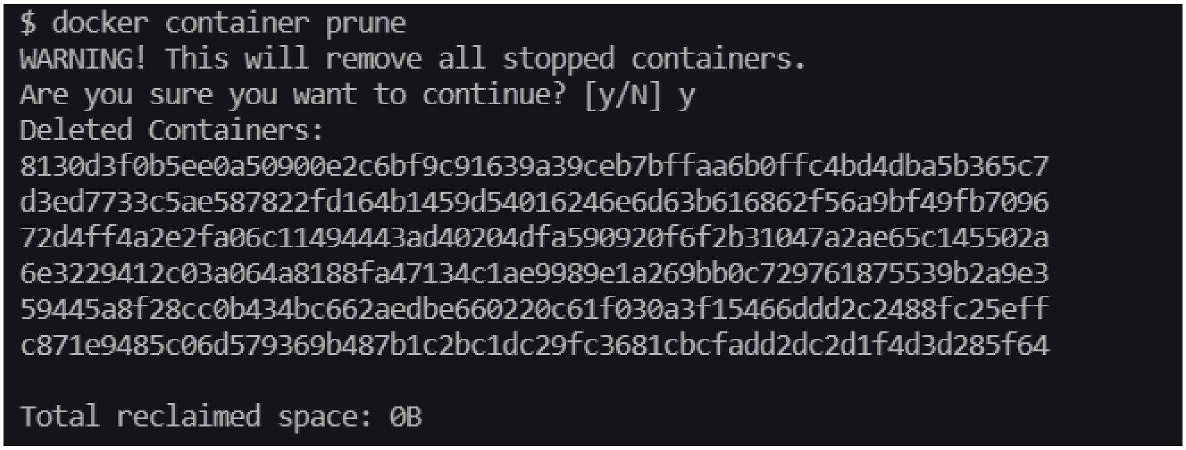
```
$ docker run -it busybox sh
/ # ls
bin   dev   etc   home  proc  root  sys   tmp   usr   var
/ #
```

8. Para eliminar contendores utilizamos docke rm <container-id|name>



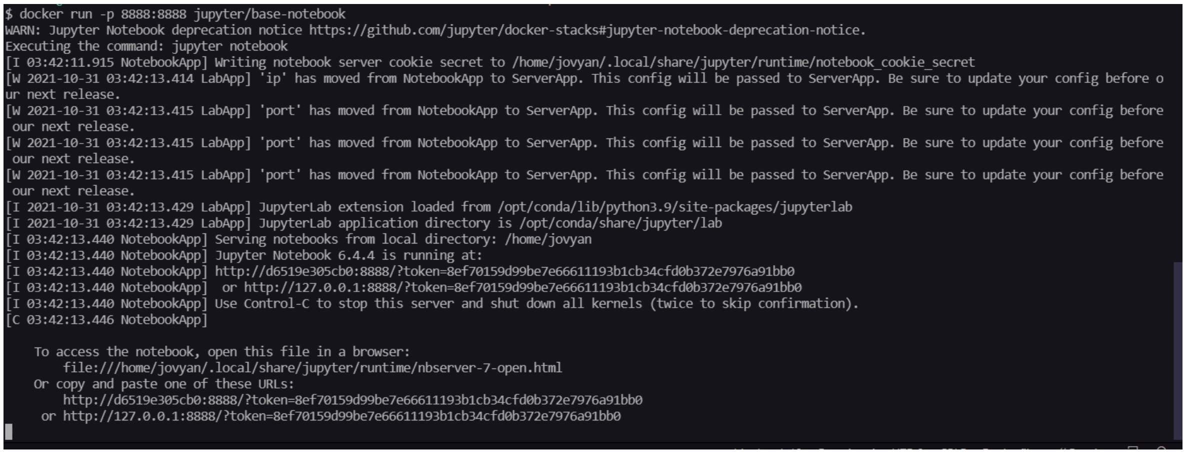
```
$ docker ps -a
CONTAINER ID   IMAGE         COMMAND        CREATED              STATUS                          PORTS     NAMES
93ff45dc8bdf   busybox       "sh"           54 seconds ago       Exited (0) 3 seconds ago                  goofy_elbakyan
8130d3f0b5ee   busybox       "bash"         About a minute ago   Created                                   kind_zhukovsky
d3ed7733c5ae   hello-world   "/hello"       About a minute ago   Exited (0) About a minute ago             dreamy_stonebraker
46e41c2e928c   busybox       "sh"           2 minutes ago        Up 2 minutes                              frosty_archimedes
72d4ff4a2e2f   busybox       "sh"           5 minutes ago        Exited (0) 5 minutes ago                  vibrant_liskov
6e3229412c03   busybox       "sh"           7 minutes ago        Exited (0) 7 minutes ago                  amazing_mirzakhani
59445a8f28cc   hello-world   "/hello"       15 minutes ago       Exited (0) 15 minutes ago                 busy_matsumoto
c871e9485c06   hello-world   "echo jeje"    2 weeks ago          Created                                   angry_cori

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker rm 93ff45dc8bdf
93ff45dc8bdf

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product development/docker
$ docker ps -a
CONTAINER ID   IMAGE         COMMAND        CREATED              STATUS                          PORTS     NAMES
8130d3f0b5ee   busybox       "bash"         About a minute ago   Created                                   kind_zhukovsky
d3ed7733c5ae   hello-world   "/hello"       About a minute ago   Exited (0) About a minute ago             dreamy_stonebraker
46e41c2e928c   busybox       "sh"           3 minutes ago        Up 3 minutes                              frosty_archimedes
72d4ff4a2e2f   busybox       "sh"           6 minutes ago        Exited (0) 6 minutes ago                  vibrant_liskov
6e3229412c03   busybox       "sh"           7 minutes ago        Exited (0) 7 minutes ago                  amazing_mirzakhani
59445a8f28cc   hello-world   "/hello"       15 minutes ago       Exited (0) 15 minutes ago                 busy_matsumoto
c871e9485c06   hello-world   "echo jeje"    2 weeks ago          Created                                   angry_cori
```

9. Para eliminar todos los contenedores usamos Docker container prune

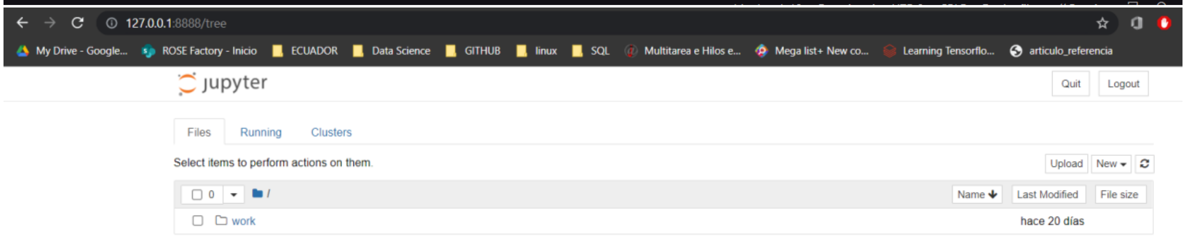


```
$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
8130d3f0b5ee0a50900e2c6bf9c91639a39ceb7bffaa6b0ffc4bd4dba5b365c7
d3ed7733c5ae587822fd164b1459d54016246e6d63b616862f56a9bf49fb7096
72d4ff4a2e2fa06c11494443ad40204dfa590920f6f2b31047a2ae65c145502a
6e3229412c03a064a8188fa47134c1ae9989e1a269bb0c729761875539b2a9e3
59445a8f28cc0b434bc662aedbe660220c61f030a3f15466ddd2c2488fc25eff
c871e9485c06d579369b487b1c2bc1dc29fc3681cbcfadd2dc2d1f4d3d285f64

Total reclaimed space: 0B
```

10. Ejecutando jupyter notebook desde un contenedor exponiendo el puerto 8888 local al 8888 de la interfaz de contenedor.



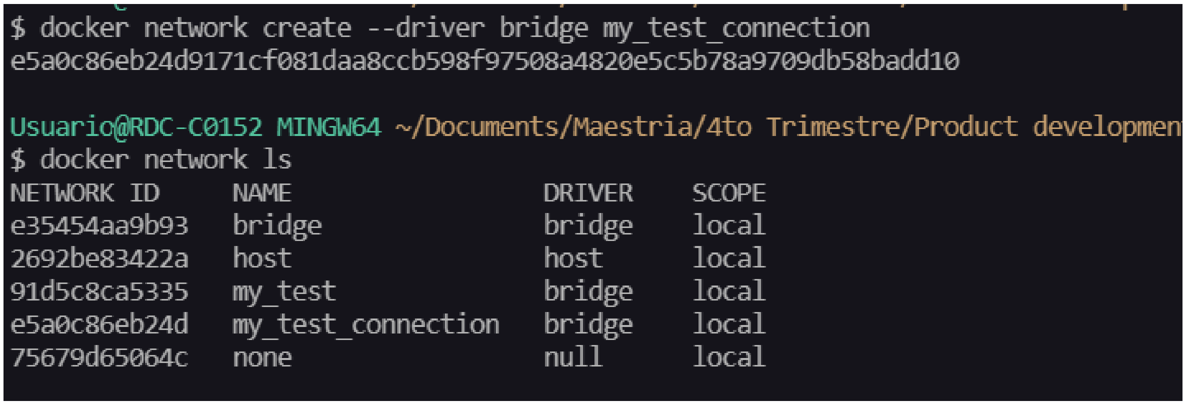
```
$ docker run -p 8888:8888 jupyter/base-notebook
WARN: Jupyter Notebook deprecation notice https://github.com/jupyter/docker-stacks#jupyter-notebook-deprecation-notice.
Executing the command: jupyter notebook
[I 03:42:11.915 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_secret
[W 2021-10-31 03:42:13.414 LabApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before o
ur next release.
[W 2021-10-31 03:42:13.415 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before
 our next release.
[W 2021-10-31 03:42:13.415 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before
 our next release.
[W 2021-10-31 03:42:13.415 LabApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before
 our next release.
[I 2021-10-31 03:42:13.429 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.9/site-packages/jupyterlab
[I 2021-10-31 03:42:13.429 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 03:42:13.440 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 03:42:13.440 NotebookApp] Jupyter Notebook 6.4.4 is running at:
[I 03:42:13.440 NotebookApp] http://d6519e305cb0:8888/?token=8ef70159d99be7e66611193b1cb34cfd0b372e7976a91bb0
[I 03:42:13.440 NotebookApp]  or http://127.0.0.1:8888/?token=8ef70159d99be7e66611193b1cb34cfd0b372e7976a91bb0
[I 03:42:13.440 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 03:42:13.446 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/nbserver-7-open.html
    Or copy and paste one of these URLs:
        http://d6519e305cb0:8888/?token=8ef70159d99be7e66611193b1cb34cfd0b372e7976a91bb0
     or http://127.0.0.1:8888/?token=8ef70159d99be7e66611193b1cb34cfd0b372e7976a91bb0
```



Select items to perform actions on them.

Name	Last Modified
work	hace 20 días

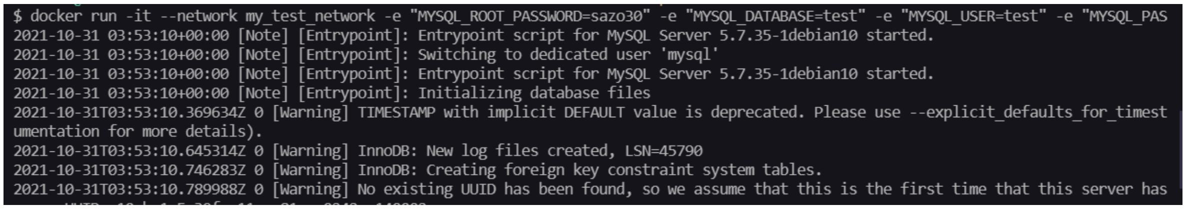
11. Creando una red para comunicar contenedores
Docker network créate –driver bridge my_test_network



```
$ docker network create --driver bridge my_test_connection
e5a0c86eb24d9171cf081daa8ccb598f97508a4820e5c5b78a9709db58badd10

Usuario@RDC-C0152 MINGW64 ~/Documents/Maestria/4to Trimestre/Product developmen
$ docker network ls
NETWORK ID     NAME                 DRIVER    SCOPE
e35454aa9b93   bridge               bridge    local
2692be83422a   host                 host      local
91d5c8ca5335   my_test              bridge    local
e5a0c86eb24d   my_test_connection   bridge    local
75679d65064c   none                 null      local
```

12. Corriendo un contenedor para base de datos con dbms mysql

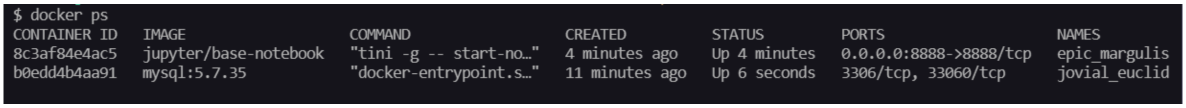
docker run -it --network my_test_network -e "MYSQL_ROOT_PASSWORD=sazo30" -e "MYSQL_DATABASE=test" -e "MYSQL_USER=test" -e "MYSQL_PASSWORD=sazo30"



```
$ docker run -it --network my_test_network -e "MYSQL_ROOT_PASSWORD=sazo30" -e "MYSQL_DATABASE=test" -e "MYSQL_USER=test" -e "MYSQL_PAS
2021-10-31 03:53:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.35-1debian10 started.
2021-10-31 03:53:10+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2021-10-31 03:53:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.35-1debian10 started.
2021-10-31 03:53:10+00:00 [Note] [Entrypoint]: Initializing database files
2021-10-31T03:53:10.369634Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timest
umentation for more details).
2021-10-31T03:53:10.645314Z 0 [Warning] InnoDB: New log files created, LSN=45790
2021-10-31T03:53:10.746283Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2021-10-31T03:53:10.789988Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has
```

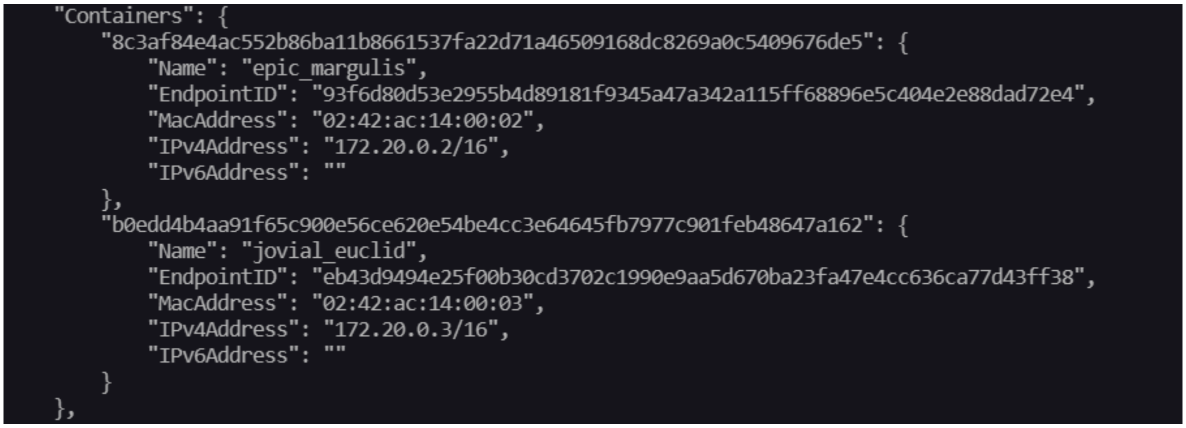
13. Ahora vamos a unir jupyter notebook con la base de datos mysql, para ello levantamos el notebook con la red recién creada.

    Tenemos levandandos los dos contenedores



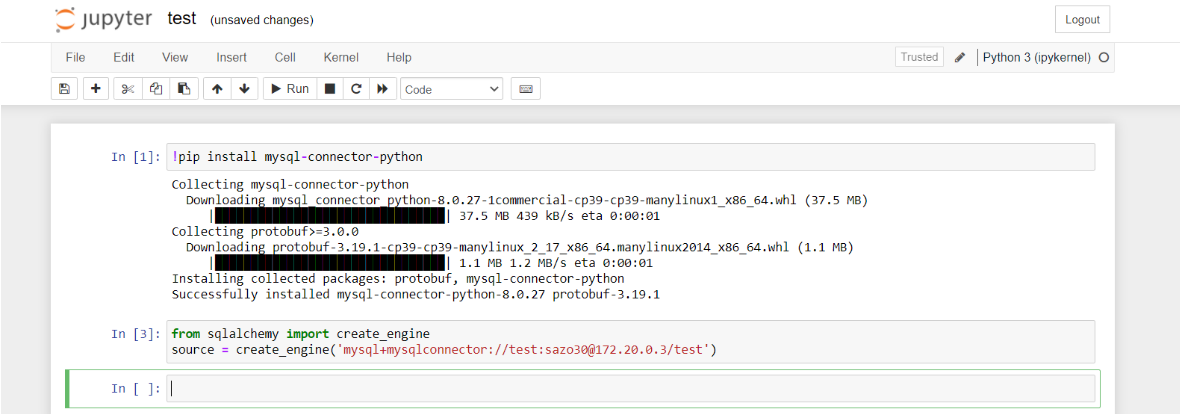
```
$ docker ps
CONTAINER ID   IMAGE                   COMMAND                  CREATED          STATUS         PORTS                    NAMES
8c3af84e4ac5   jupyter/base-notebook   "tini -g -- start-no…"   4 minutes ago    Up 4 minutes   0.0.0.0:8888->8888/tcp   epic_margulis
b0edd4b4aa91   mysql:5.7.35            "docker-entrypoint.s…"   11 minutes ago   Up 6 seconds   3306/tcp, 33060/tcp      jovial_euclid
```

14. Ahora para conectar necesitamos saber la ip que le asigno la red creada cada contenedor, para ello utilizamos Docker network inspect <network_name>



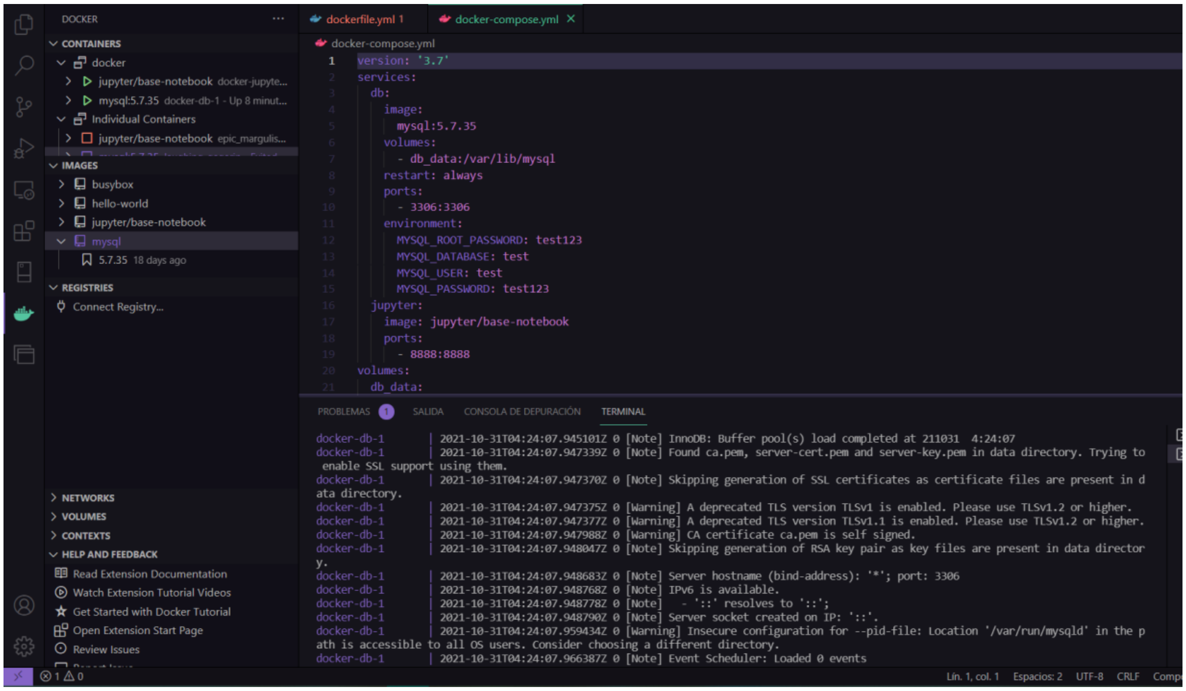
```
    "Containers": {
        "8c3af84e4ac552b86ba11b8661537fa22d71a46509168dc8269a0c5409676de5": {
            "Name": "epic_margulis",
            "EndpointID": "93f6d80d53e2955b4d89181f9345a47a342a115ff68896e5c404e2e88dad72e4",
            "MacAddress": "02:42:ac:14:00:02",
            "IPv4Address": "172.20.0.2/16",
            "IPv6Address": ""
        },
        "b0edd4b4aa91f65c900e56ce620e54be4cc3e64645fb7977c901feb48647a162": {
            "Name": "jovial_euclid",
            "EndpointID": "eb43d9494e25f00b30cd3702c1990e9aa5d670ba23fa47e4cc636ca77d43ff38",
            "MacAddress": "02:42:ac:14:00:03",
            "IPv4Address": "172.20.0.3/16",
            "IPv6Address": ""
        }
    },
```

15. Ahora necesitamos instalar las librerías en notebook, utilzamos las siguientes: pandas y mysql-connector-python.



```
In [1]: !pip install mysql-connector-python

        Collecting mysql-connector-python
          Downloading mysql_connector_python-8.0.27-1commercial-cp39-cp39-manylinux1_x86_64.whl (37.5 MB)
             |████████████████████████████████| 37.5 MB 439 kB/s eta 0:00:01
        Collecting protobuf>=3.0.0
          Downloading protobuf-3.19.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
             |████████████████████████████████| 1.1 MB 1.2 MB/s eta 0:00:01
        Installing collected packages: protobuf, mysql-connector-python
        Successfully installed mysql-connector-python-8.0.27 protobuf-3.19.1

In [3]: from sqlalchemy import create_engine
        source = create_engine('mysql+mysqlconnector://test:sazo30@172.20.0.3/test')

In [ ]: 
```

16. Ahora vamos a levantar lo mismo utilizando Docker-compose

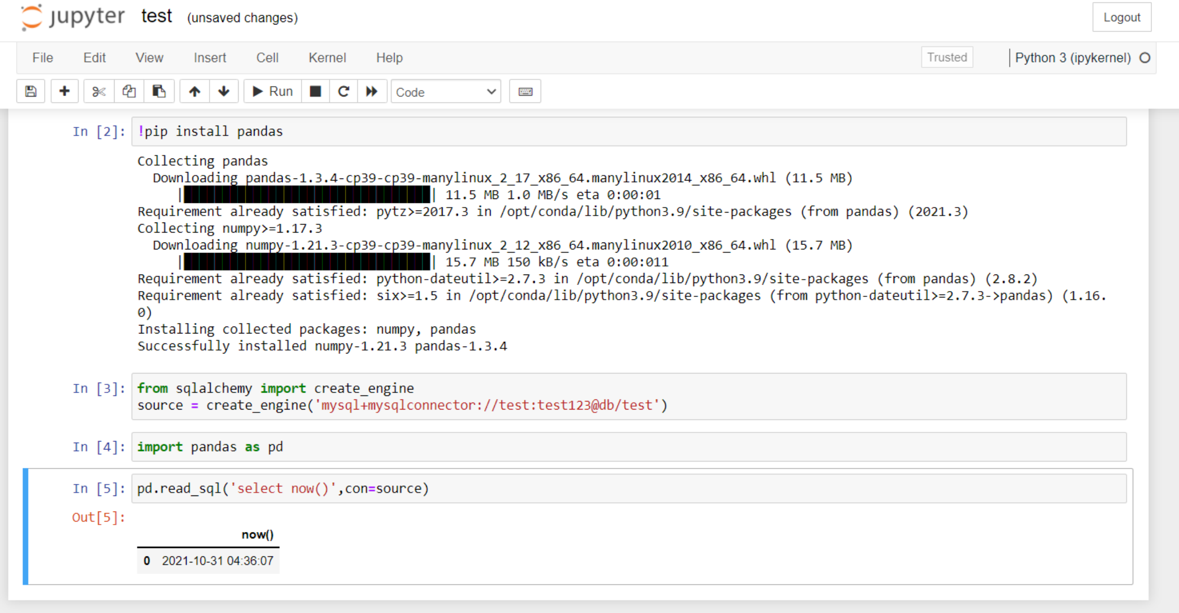


```yaml
version: '3.7'
services:
  db:
    image:
      mysql:5.7.35
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    ports:
      - 3306:3306
    environment:
      MYSQL_ROOT_PASSWORD: test123
      MYSQL_DATABASE: test
      MYSQL_USER: test
      MYSQL_PASSWORD: test123
  jupyter:
    image: jupyter/base-notebook
    ports:
      - 8888:8888
volumes:
  db_data:
```

```
docker-db-1       | 2021-10-31T04:24:07.945101Z 0 [Note] InnoDB: Buffer pool(s) load completed at 211031  4:24:07
docker-db-1       | 2021-10-31T04:24:07.947339Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to
 enable SSL support using them.
docker-db-1       | 2021-10-31T04:24:07.947370Z 0 [Note] Skipping generation of SSL certificates as certificate files are present in d
ata directory.
docker-db-1       | 2021-10-31T04:24:07.947375Z 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
docker-db-1       | 2021-10-31T04:24:07.947377Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
docker-db-1       | 2021-10-31T04:24:07.947988Z 0 [Warning] CA certificate ca.pem is self signed.
docker-db-1       | 2021-10-31T04:24:07.948047Z 0 [Note] Skipping generation of RSA key pair as key files are present in data director
y.
docker-db-1       | 2021-10-31T04:24:07.948683Z 0 [Note] Server hostname (bind-address): '*'; port: 3306
docker-db-1       | 2021-10-31T04:24:07.948768Z 0 [Note] IPv6 is available.
docker-db-1       | 2021-10-31T04:24:07.948778Z 0 [Note]   - '::' resolves to '::';
docker-db-1       | 2021-10-31T04:24:07.948790Z 0 [Note] Server socket created on IP: '::'.
docker-db-1       | 2021-10-31T04:24:07.959434Z 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the p
ath is accessible to all OS users. Consider choosing a different directory.
docker-db-1       | 2021-10-31T04:24:07.966387Z 0 [Note] Event Scheduler: Loaded 0 events
```

Podemos ver que tenemos levantados ambos contenedores perteneciendo a la misma red.

17. Y podemos validar que tenemos conexión a base de datos desde jupyter notebook



jupyter test (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel) ○
Run
Code

```
In [2]: !pip install pandas

        Collecting pandas
          Downloading pandas-1.3.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
             |████████████████████████████████| 11.5 MB 1.0 MB/s eta 0:00:01
        Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.9/site-packages (from pandas) (2021.3)
        Collecting numpy>=1.17.3
          Downloading numpy-1.21.3-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
             |████████████████████████████████| 15.7 MB 150 kB/s eta 0:00:011
        Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/lib/python3.9/site-packages (from pandas) (2.8.2)
        Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-packages (from python-dateutil>=2.7.3->pandas) (1.16.
        0)
        Installing collected packages: numpy, pandas
        Successfully installed numpy-1.21.3 pandas-1.3.4

In [3]: from sqlalchemy import create_engine
        source = create_engine('mysql+mysqlconnector://test:test123@db/test')

In [4]: import pandas as pd

In [5]: pd.read_sql('select now()',con=source)

Out[5]:
```

	now()
0	2021-10-31 04:36:07