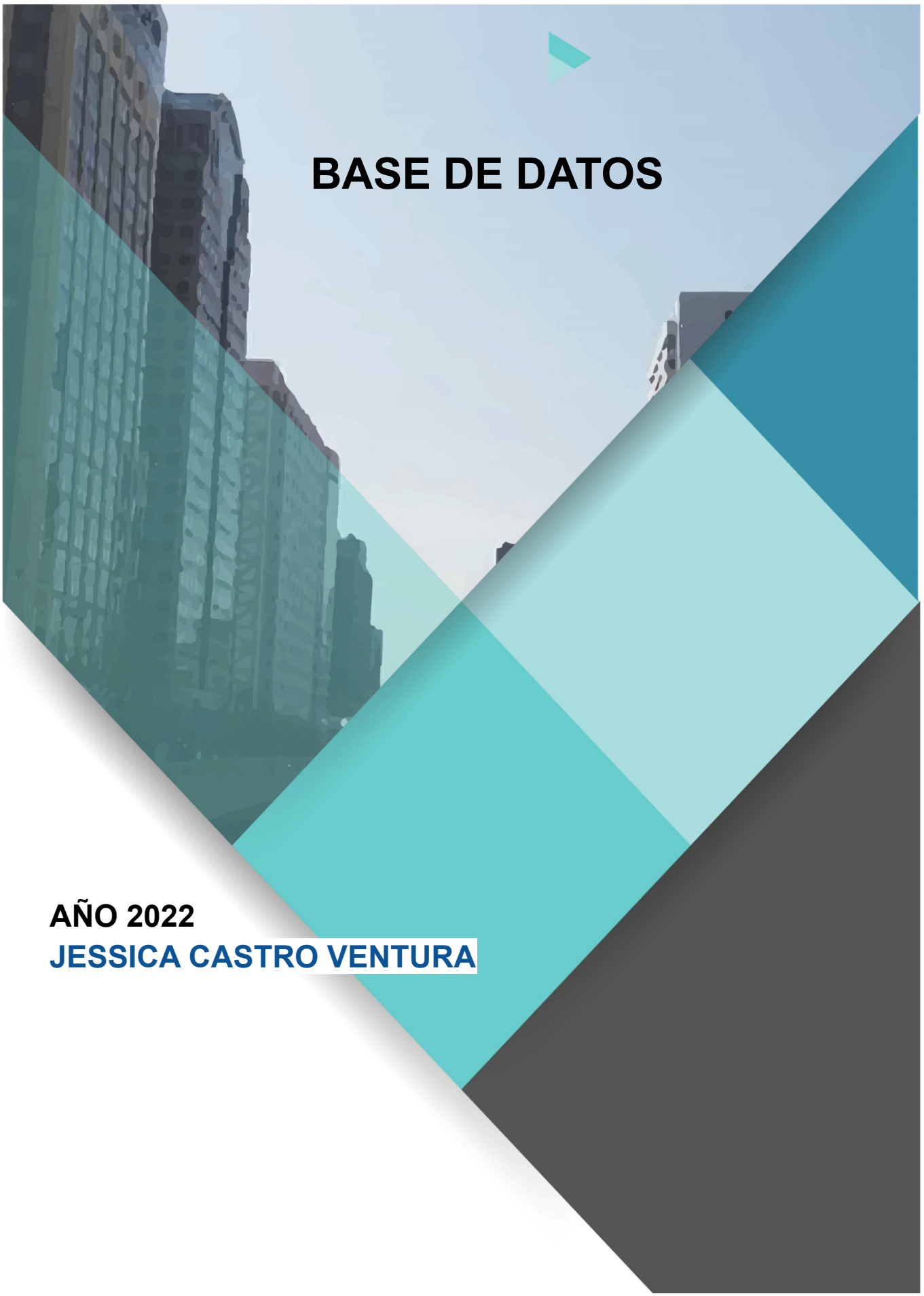


BASE DE DATOS

AÑO 2022

JESSICA CASTRO VENTURA

TABLA DE CONTENIDO 2º ENTREGA

INSTALACIÓN DOCKER

https://docs.docker.com/engine/install/ubuntu/

```
usuario@cjessica:~$ sudo su
[sudo] contraseña para usuario:
root@cjessica:/home/usuario#  sudo apt-get remove docker docker-en
gine docker.io containerd runc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete docker-engine
root@cjessica:/home/usuario#
```

```
usuario@cjessica:~$ sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
[114 kB]
Des:3 http://security.ubuntu.com/ubuntu focal-security InRelease [
114 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InReleas
e [108 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64
 Packages [1.674 kB]
Des:6 http://security.ubuntu.com/ubuntu focal-security/main i386 P
```

```
jessica@ubuntu:~$  sudo apt-get install \
>     ca-certificates \
>     curl \
>     gnupg \
>     lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20210119~20.04.2).
ca-certificates set to manually installed.
curl is already the newest version (7.68.0-1ubuntu2.7).
gnupg is already the newest version (2.2.19-3ubuntu2.1).
gnupg set to manually installed.
The following packages were automatically installed and are no lo
nger required:
  linux-headers-5.11.0-38-generic
  linux-hwe-5.11-headers-5.11.0-38
  linux-image-5.11.0-38-generic linux-modules-5.11.0-38-generic
  linux-modules-extra-5.11.0-38-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
```

```
jessica@ubuntu:~$ echo \
>   "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/
keyrings/docker-archive-keyring.gpg] https://download.docker.com/
linux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d
/docker.list > /dev/null
```



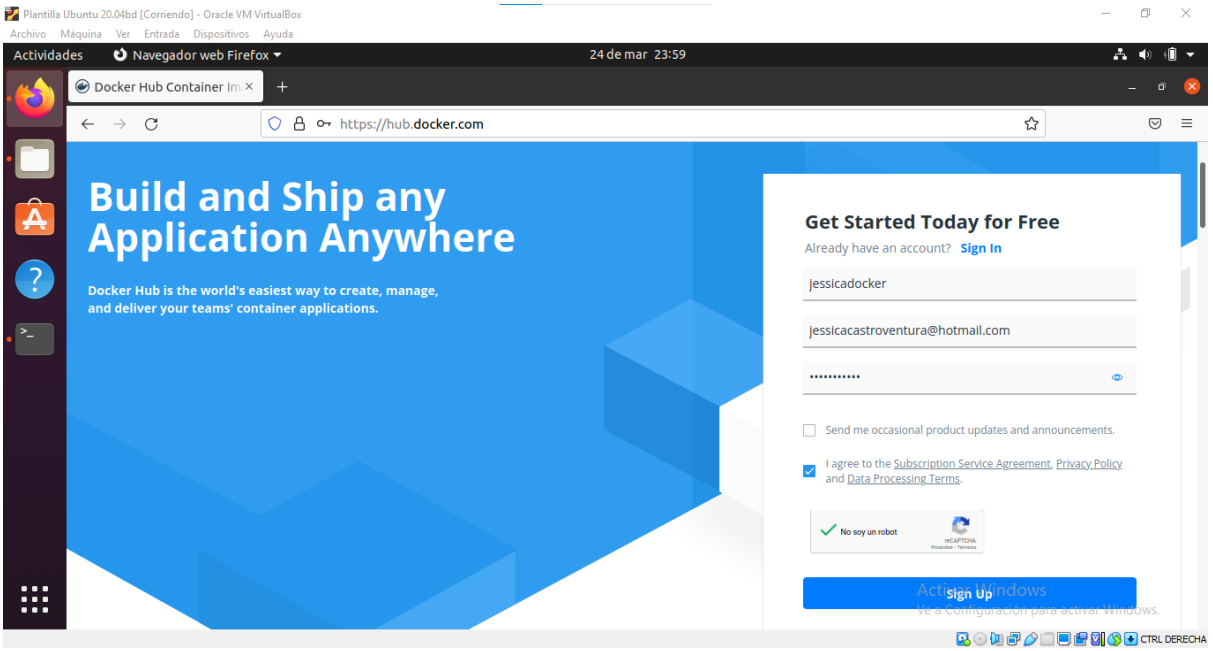
```
usuario@cjessica:~$  curl -fsSL https://download.docker.com/linux/
ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-arch
ive-keyring.gpg
```

```
usuario@cjessica:~$ echo \
>   "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/k
eyrings/docker-archive-keyring.gpg] https://download.docker.com/li
nux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/
docker.list > /dev/null
```

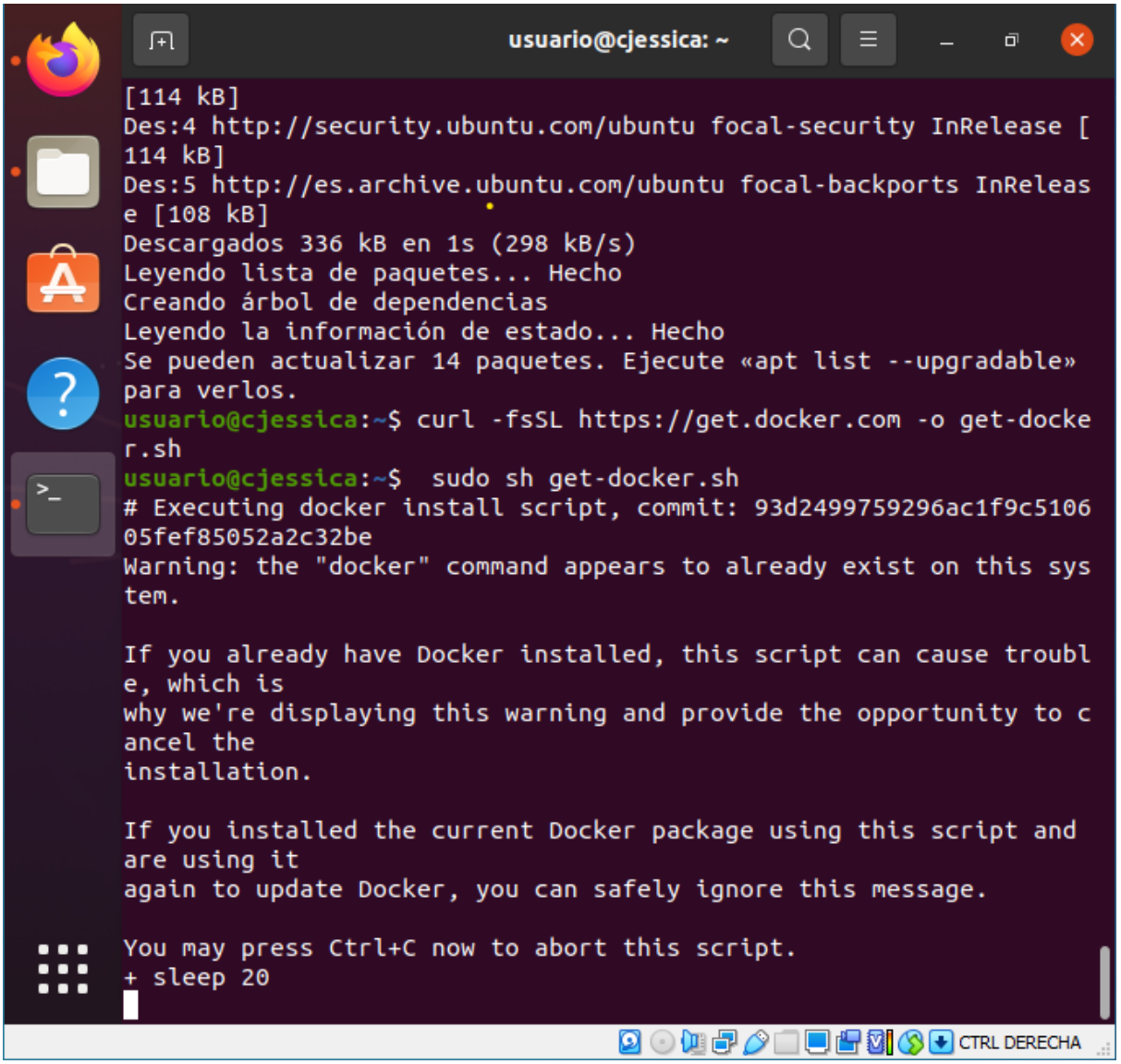
Instalar Docker Engine

```
usuario@cjessica:~$  sudo apt-get install docker-ce docker-ce-cli
containerd.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma autom
ática y ya no son necesarios.
```

```
usuario@cjessica:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41c553fdf5a8
3e95a41d40c38
Status: Downloaded newer image for hello-world:latest

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

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 Dock
er Hub.
    (amd64)
 3. The Docker daemon created a new container from that image whic
h runs the
    executable that produces the output you are currently reading.
```



en id pongo jessicaasirb


usuario@cjessica: ~
[114 kB]
Des:4 http://security.ubuntu.com/ubuntu focal-security InRelease [
114 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu focal-backports InReleas
e [108 kB]
Descargados 336 kB en 1s (298 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 14 paquetes. Ejecute «apt list --upgradable»
para verlos.
usuario@cjessica:~$ curl -fsSL https://get.docker.com -o get-docke
r.sh
usuario@cjessica:~$ sudo sh get-docker.sh
Executing docker install script, commit: 93d2499759296ac1f9c5106
05fef85052a2c32be
Warning: the "docker" command appears to already exist on this sys
tem.
If you already have Docker installed, this script can cause troubl
e, which is
why we're displaying this warning and provide the opportunity to c
ancel the
installation.
If you installed the current Docker package using this script and
are using it
again to update Docker, you can safely ignore this message.
You may press Ctrl+C now to abort this script.
+ sleep 20
CTRL DERECHA

```
Client: Docker Engine - Community
 Version:           20.10.14
 API version:       1.41
 Go version:        go1.16.15
 Git commit:        a224086
 Built:             Thu Mar 24 01:48:02 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.14
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.15
  Git commit:       87a90dc
  Built:            Thu Mar 24 01:45:53 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.5.11
  GitCommit:        3df54a852345ae127d1fa3092b95168e4a88e2f8
 runc:
  Version:          1.0.3
  GitCommit:        v1.0.3-0-gf46b6ba
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

mi usuario de la máquina no del docker

```
aso@sjessica:~$ sudo usermod -aG docker aso
aso@sjessica:~$ su - aso
Password:
aso@sjessica:~$
```

```
aso@sjessica:~$ id -nG
aso root docker
aso@sjessica:~$
```

CAMBIAR USUARIO Y PONERLO COMO ADMINISTRADOR

```
daniel@ubuntu:~$ sudo adduser jessica
Adding user `jessica' ...
Adding new group `jessica' (1001) ...
Adding new user `jessica' (1001) with group `jessica' ...
Creating home directory `/home/jessica' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jessica
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
daniel@ubuntu:~$ sudo usermod -aG sudo jessica
daniel@ubuntu:~$
```

DOCKER 1ER CONTACTO

```
aso@sjessica:~$ 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/
```

CARGA UBUNTU (DESCARGANDO SU IMAGEN) Y SE ABRE UNA SHELL

```
aso@sjessica:~$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
4d32b49e2995: Pull complete
Digest: sha256:bea6d19168bbfd6af8d77c2cc3c572114eb5d113e6f422573c93cb605a0e2ffb
Status: Downloaded newer image for ubuntu:latest
root@c8ccd124a355:/#
```

VER IMÁGENES DESCARGADAS

```
aso@sjessica:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    ff0fea8310f3   7 days ago      72.8MB
hello-world   latest    feb5d9fea6a5   6 months ago    13.3kB
```

VER CONTENEDORES QUE SE ESTÁN EJECUTANDO (NINGUNO)

```
aso@sjessica:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

CONTENEDORES QUE TENGO Y SU ESTADO AUNQUE ESTÉN EXITED O EN EJECUCIÓN (TODOS LOS QUE HAY)

```
aso@sjessica:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS
 NAMES
c8ccd124a355   ubuntu        "bash"     44 minutes ago   Exited (127) 37 minutes ago
 nifty_grothendieck
a7a7afe39cb4   hello-world   "/hello"   51 minutes ago   Exited (0) 51 minutes ago
 beautiful_joliot
3eb649fc212e   hello-world   "/hello"   2 hours ago      Exited (0) 2 hours ago
 optimistic_perlman
```

BORRAR

docker rm -f $(docker ps -qa) _ esto borra todos los contenedores

BORRAR UN CONTENEDOR POR ID (SI ESTÁ ACTIVO UTILIZO -F PARA FORZAR EL APAGADO O DOCKER STOP MYSQL-DB)

```
aso@sjessica:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS
               PORTS     NAMES
1c882453e162   ubuntu        "bash"     2 minutes ago    Exited (0) 2
minutes ago                tender_johnson
d108448b365c   hello-world   "/hello"   6 minutes ago    Exited (0) 6
minutes ago                quirky_gauss
8706ac817869   ubuntu        "bash"     12 minutes ago   Exited (1) 7
minutes ago                xenodochial_poincare
c8ccd124a355   ubuntu        "bash"     47 hours ago     Exited (127)
46 hours ago               nifty_grothendieck
a7a7afe39cb4   hello-world   "/hello"   47 hours ago     Exited (0) 47
 hours ago                 beautiful_joliot
3eb649fc212e   hello-world   "/hello"   2 days ago       Exited (0) 2
days ago                   optimistic_perlman
```

UN CONTENEDOR SE BORRA CON RM Y COMPROBAMOS NUEVAMENTE QUE YA NO APARECE EL ID1C882453E162

```
aso@sjessica:~$ docker rm 1c882453e162
1c882453e162
aso@sjessica:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS
               PORTS     NAMES
d108448b365c   hello-world   "/hello"   17 minutes ago   Exited (0) 17
 minutes ago              quirky_gauss
8706ac817869   ubuntu        "bash"     23 minutes ago   Exited (1) 18
 minutes ago              xenodochial_poincare
c8ccd124a355   ubuntu        "bash"     47 hours ago     Exited (127)
47 hours ago              nifty_grothendieck
a7a7afe39cb4   hello-world   "/hello"   47 hours ago     Exited (0) 47
 hours ago                beautiful_joliot
3eb649fc212e   hello-world   "/hello"   2 days ago       Exited (0) 2
days ago                  optimistic_perlman
```

BORRAR UN CONTENEDOR POR NOMBRE , VAMOS A ESCOGER QUIRKY GAUSS

```
aso@sjessica:~$ docker rm quirky_gauss
quirky_gauss
aso@sjessica:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS
               PORTS     NAMES
8706ac817869   ubuntu        "bash"     28 minutes ago   Exited (1) 24
 minutes ago              xenodochial_poincare
c8ccd124a355   ubuntu        "bash"     47 hours ago     Exited (127)
47 hours ago              nifty_grothendieck
a7a7afe39cb4   hello-world   "/hello"   47 hours ago     Exited (0) 47
 hours ago                beautiful_joliot
3eb649fc212e   hello-world   "/hello"   2 days ago       Exited (0) 2
days ago                  optimistic_perlman
```

PARA BORRAR IMÁGENES ES CON **RMI** DEL MISMO MODO QUE CON CONTENEDORES

A PARTIR DE LA IMAGEN—> SE CREA EL CONTENEDOR CON DOCKER RUN.

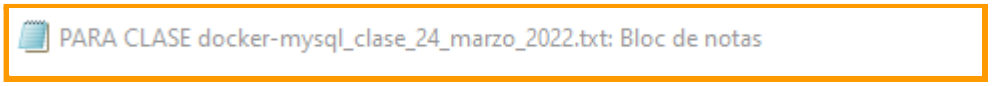
CON UNA SOLA IMAGEN PODRÍAMOS CREAR VARIOS CONTENEDORES.

EN ESTE CASO TENGO QUE FORZAR EL CIERRE CON -F PARA PODER BORRAR LA IMAGEN YA K NO ESTA STOPPED

```
aso@sjessica:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    ff0fea8310f3   9 days ago     72.8MB
hello-world   latest    feb5d9fea6a5   6 months ago   13.3kB
aso@sjessica:~$ docker rmi ff0fea8310f3
Error response from daemon: conflict: unable to delete ff0fea8310f3 (m
ust be forced) - image is being used by stopped container 8706ac817869
aso@sjessica:~$ docker rmi -f  ff0fea8310f3
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:bea6d19168bbfd6af8d77c2cc3c572114eb5d113e6f422
573c93cb605a0e2ffb
Deleted: sha256:ff0fea8310f3957d9b1e6ba494f3e4b63cb348c76160c6c15578e6
5995ffaa87
aso@sjessica:~$
```

AQUÍ LA BORRO POR EL NOMBRE Y COMPRUEBO QUE YA BORRÉ LAS DOS IMÁGENES QUE HABÍA EN LA CAPTURA DE ARRIBA.

```
aso@sjessica:~$ docker rmi -f hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:bfea6278a0a267fad2634554f4f0c6f31981eea41
c553fdf5a83e95a41d40c38
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef3
56dc6b3412
aso@sjessica:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
aso@sjessica:~$
```



INSTALAR IMAGEN DE MYSQL

tags = versiones

SI NO LE PONGO UN TAG NOS DA LA ÚLTIMA VERSIÓN

ES IMPORTANTE PONER EL –NAME QUE TÚ QUIERAS PARA QUE NO PONGA UNO ALEATORIO

(-D= DETTACH, SE EJECUTA EN BACKGROUND)

(-E: ENVIROMENT, VARAIBLE DE ENTORNO)

(EL ÚLTIMO “MYSQL” ES LA IMAGEN UTILIZADA)

```
aso@sjessica:~$ docker run -d -p 3306:3306 --name mysql-db -e MYSQL_RO
OT_PASSWORD=Abcd1234. mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
a4b007099961: Pull complete
e2b610d88fd9: Pull complete
38567843b438: Pull complete
5fc423bf9558: Pull complete
aa8241dfe828: Pull complete
cc662311610e: Pull complete
9832d1192cf2: Pull complete
f2aa1710465f: Pull complete
4a2d5722b8f3: Pull complete
3a246e8d7cac: Pull complete
2f834692d7cc: Pull complete
a37409568022: Pull complete
Digest: sha256:b2ae0f527005d99bacdf3a220958ed171e1eb0676377174f0323e0a
10912408a
Status: Downloaded newer image for mysql:latest
45a46687dbf707998bb69a3d59b2ee53672445833e3e879d0ae6f24858c0565c
```

```
aso@sjessica:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
mysql        latest    562c9bc24a08   9 days ago   521MB
aso@sjessica:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED
      STATUS                      PORTS
                    NAMES
45a46687dbf7   mysql          "docker-entrypoint.s…"   About a minute
ago   Up About a minute           0.0.0.0:3306->3306/tcp, :::3306->330
6/tcp, 33060/tcp   mysql-db
8706ac817869   ff0fea8310f3   "bash"                   About an hour a
go    Exited (1) 55 minutes ago
                    xenodochial_poincare
c8ccd124a355   ff0fea8310f3   "bash"                   47 hours ago
      Exited (127) 47 hours ago
                    nifty_grothendieck
a7a7afe39cb4   feb5d9fea6a5   "/hello"                 47 hours ago
      Exited (0) 47 hours ago
                    beautiful_joliot
3eb649fc212e   feb5d9fea6a5   "/hello"                 2 days ago
      Exited (0) 2 days ago
                    optimistic_perlman
```

DOCKER LOGS - NOS DA INFORMACIÓN DE LOGS, PODEMOS PONER EL ID, O EL NOMBRE DEL CONTENEDOR. (MYSQL-DB EN ESTE CASO)

```
aso@sjessica:~$ docker logs 45a46687dbf7
2022-03-27 17:44:25+00:00 [Note] [Entrypoint]: Entrypoint script for M
ySQL Server 8.0.28-1debian10 started.
2022-03-27 17:44:26+00:00 [Note] [Entrypoint]: Switching to dedicated
user 'mysql'
2022-03-27 17:44:27+00:00 [Note] [Entrypoint]: Entrypoint script for M
ySQL Server 8.0.28-1debian10 started.
2022-03-27 17:44:27+00:00 [Note] [Entrypoint]: Initializing database f
iles
2022-03-27T17:44:27.170803Z 0 [System] [MY-013169] [Server] /usr/sbin/
```

:~$ DOCKER INSPECT MYSQL-DB O DOCKER INSPECT FE39000EB625, PARA QUE NOS MUESTRE MÁS INFORMACIÓN DEL CONTENEDOR.

ENTRAR EN EL CONTENEDOR

(-IT = INTERACTIVE; -P = PASSWORD)

```
aso@sjessica:~$ docker exec -it mysql-db /bin/bash
root@45a46687dbf7:/# ls
bin   docker-entrypoint-initdb.d  home   media  proc  sbin  tmp
boot  entrypoint.sh               lib    mnt    root  srv   usr
dev   etc                         lib64  opt    run   sys   var
root@45a46687dbf7:/# mysql -uroot -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using p
assword: YES)
root@45a46687dbf7:/# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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 st
atement.
```

VERSIÓN PREFERIBLE

PASSWORD ABCD1234.

```
aso@sjessica:~$ docker exec -it mysql-db mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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.
```

APARECEN LAS BASES DEL SISTEMA

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

CREAMOS UNA BD PRUEBA

create database prueba;

```
mysql>  show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| prueba             |
| sys                |
+--------------------+
5 rows in set (0.03 sec)
```

mysql> exit
Bye
root@dcc0e40a5fd0:/# exit
exit

ahora nos cargamos el contenedor, lo volvemos a buscar y vemos que no está:
root@bdjessica:/home/usuario# docker rm -f mysql-db
ahora vemos que efectivamente se ha borrado:

```
root@bdjessica:/home/usuario# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS   PORTS          NAMES
root@bdjessica:/home/usuario# docker exec -it mysql-db mysql -p
Error: No such container: mysql-db
```

SE REALIZAN LOS PASOS HECHOS HASTA AHORA OTRA VEZ.

EN EL CASO ANTERIOR, SE HA CREADO UN VOLUME TEMPORAL, DONDE SE BORRAN LOS DATOS UNA VEZ ELIMINADO EL CONTENEDOR.

MONTAR UN VOLUMEN

PARA QUE LA BD CREADA NO SE BORRE Y SEPUEDA RECUPERAR USAMOS VOLÚMENES.

RUTA POR DEFECTO VOLUMES : **/VAR/LIB/DOCKER/VOLUMES**

OBJETOS PRINCIPALES:

-IMÁGENES

-CONTENEDORES

-VOLÚMENES = CARPETAS

-REDES

$ docker rm -f mysql-db

Eliminamos todos los volúmenes

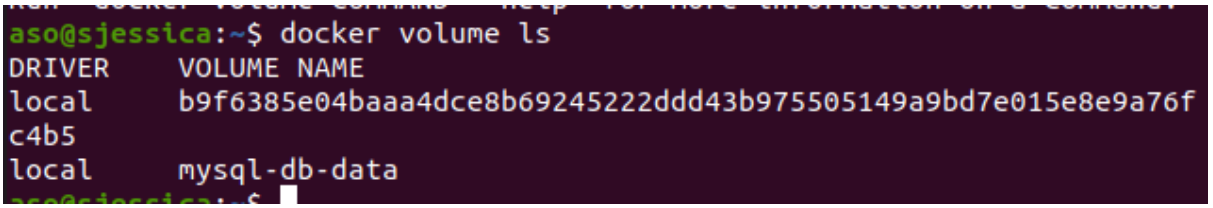
Docker crea volúmenes temporales sin pedirte permiso.

CREAMOS VOLUMEN

SI NO SE HACE EXPLÍCITAMENTE EL SISTEMA NOS CREARÁ UN VOLUMEN IMPLÍCITAMENTE.

```
aso@sjessica:~$ docker volume create mysql-db-data
mysql-db-data
```

Verificamos su creación



MONTAR CARPETAS PERSISTENTES

Docker run y agregamos el volumen con la opción --mount

root@bdjessica:/home/usuario# docker run -d -p 33060:3306 --name mysql-db -e MYSQL_ROOT_PASSWORD=Abcd1234. --mount src=mysql-db-data,dst=/var/lib/mysql mysql

1fdac1fa4e376886bd3ec9fea2a9d17bb6f810082007f25f121623e23c245768

--MOUNT SRC=MYSQL-DB-DATA = **VOLUMEN DEL HOST MAPEADO A LA CARPETA DE DIRECCIONAMIENTO DEL CONTENEDOR** DST=/VAR/LIB/MYSQL MYSQL

---

creamos otra bd demo y vamos a comprobar la persistencia
mysql> create database demo;
Query OK, 1 row affected (0.01 sec)

Entramos nuevamente al contenedor de forma interactiva y podemos ver que la base de datos que creamos se encuentra

$ docker exec -it mysql-db mysql -p

mysql> show databases;

```
+--------------------+
| Database           |
+--------------------+
| demo               |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> quit
Bye
root@bdjessica:/home/usuario# docker rm -f mysql-db
mysql-db
root@bdjessica:/home/usuario# docker run -d -p 33060:3306 --n
-db  -e MYSQL_ROOT_PASSWORD=Abcd1234. --mount src=mysql-db-da
ar/lib/mysql mysql
59b1b0fe822cd840f92735edcf3fc7ff101c6ec7681e9ba617d53f42b2870
root@bdjessica:/home/usuario#  docker exec -it mysql-db mysql
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28 MySQL Community Server - GPL
```

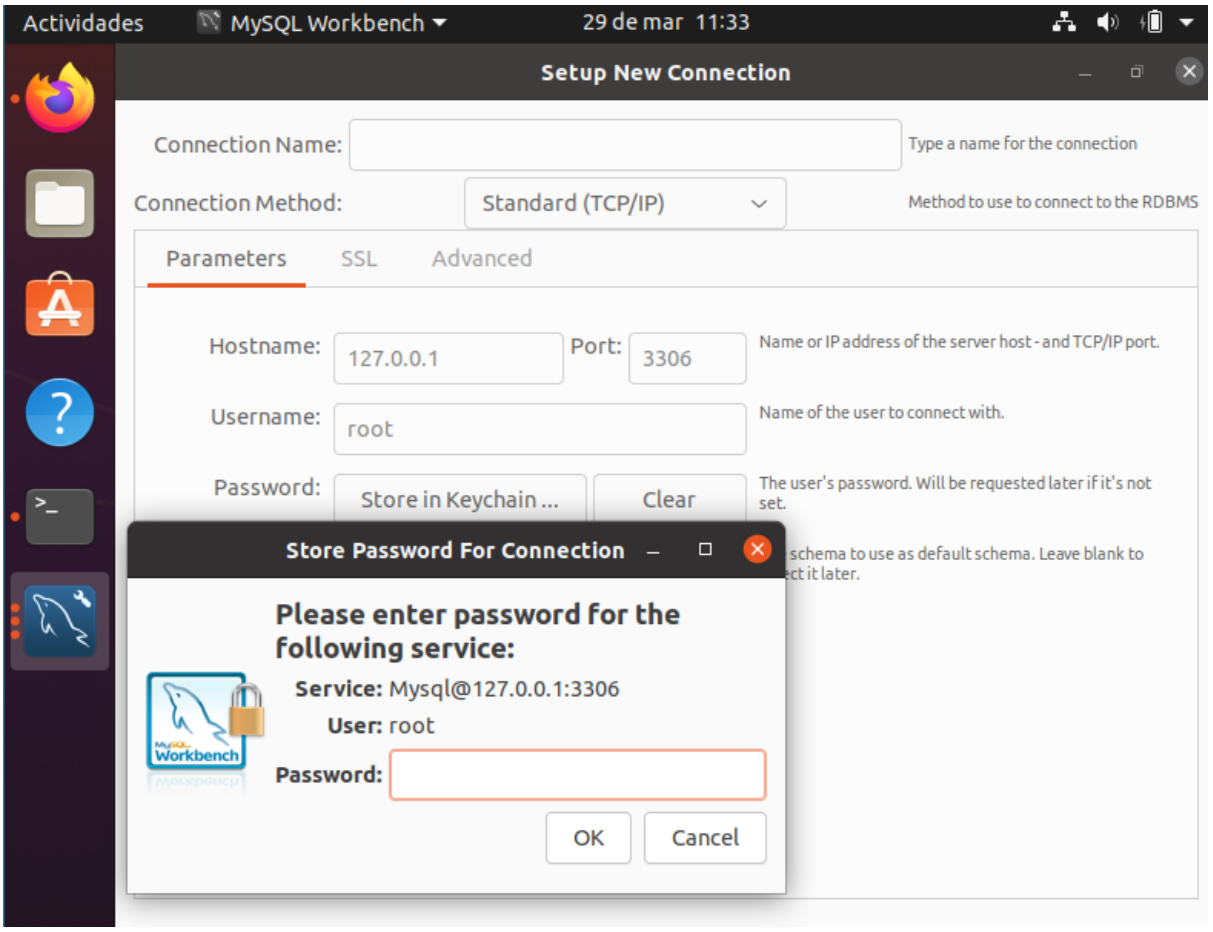
EFECTIVAMENTE LO SIGUE MOSTRANDO

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

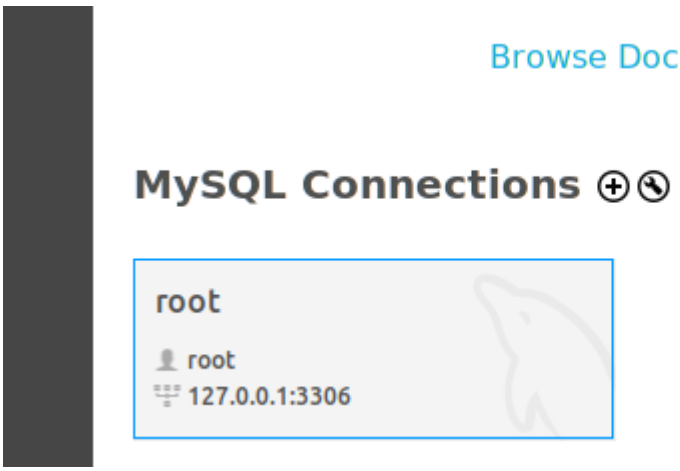
De esta forma ya estamos trabajando con volúmenes <u>donde persistimos los datos</u> en el Host , si queremos utilizar la base de datos solo hay que montar el volumen.

WORKBENCH

(DESCARGAR DESDE UBUNTU SOFTWARE)



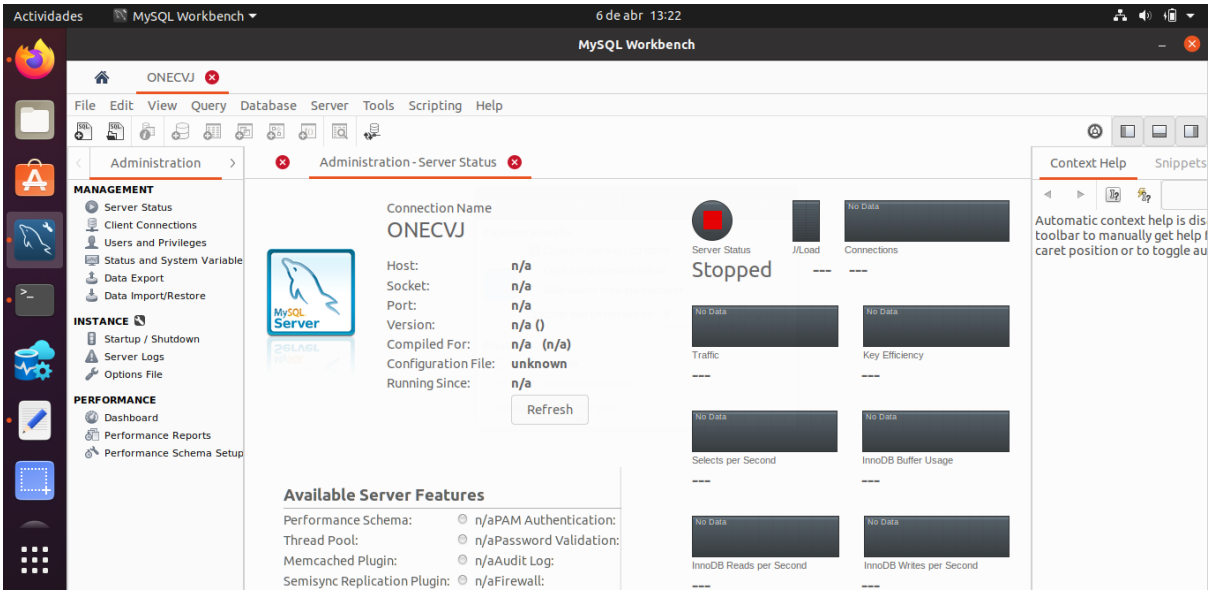
CONNECTION NAME : ROOT
PASSWORD : Abcd1234.



PARA QUE NO DE ERROR AL CLICAR EN LA CONEXIÓN HAY QUE AÑADIR ESTO EN LA TERMINAL Y ASÍ PROPORCIONARLE LOS PERMISOS:

snap connect mysql-workbench-community:password-manager-service
snap connect mysql-workbench-community:ssh-keys

AHORA CLIKEAS EN LA CONEXIÓN Y COMPRUEBAS



VAMOS A LA TERMINAL PARA COMPROBAR LA CONEXIÓN:

jessica@ubu20drc:~$ sudo su
[sudo] contraseña para jessica:
root@ubu20drc:/home/jessica# docker exec -it mysql-db mysql -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.29 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| demo               |
| football           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| tienda             |
+--------------------+
7 rows in set (0.02 sec)

 mysql> USE tienda;
Database changed
mysql> show tables;
+------------------+
| Tables_in_tienda |
+------------------+
| fabricante       |
| producto         |
+------------------+
2 rows in set (0.01 sec)

mysql> select * from fabricante;
+--------+-----------------+
| codigo | nombre          |
+--------+-----------------+
|      1 | Asus            |
|      2 | Lenovo          |
|      3 | Hewlett-Packard |
|      4 | Samsung         |
|      5 | Seagate         |
|      6 | Crucial         |
|      7 | Gigabyte        |
|      8 | Huawei          |
|      9 | Xiaomi          |
+--------+-----------------+
9 rows in set (0.00 sec)
```

CARGAR DATOS MSQL _"load_data.sql"

descargar archivo, luego lo muevo a home

root@bdjessica:/# cd home/
root@bdjessica:/home# ls
usuario
root@bdjessica:/home# cd usuario
root@bdjessica:/home/usuario# ls
Descargas get-docker.sh Música snap
Documentos Imágenes Plantillas test-docker.sh
Escritorio load_data.sql Público Vídeos
root@bdjessica:/home/usuario# cd Descargas/
(DESCARGO OTRO ARCHIVO Y LO METO EN HOME NUEVAMENTE)
root@bdjessica:/home/usuario/Descargas# ls
docker-compose.yml load_data.sql
root@bdjessica:/home/usuario/Descargas# cp docker-compose.yml /home/*
CP: NO SE SOBREESCRIBIRÁ EL FICHERO '/HOME/USUARIO/DOCKER-COMPOSE.YML' RECIÉN CREADO CON '/HOME/DOCKER-COMPOSE.YML'
root@bdjessica:/home/usuario/Descargas# cd ..
root@bdjessica:/home/usuario# ls
Descargas get-docker.sh Plantillas Vídeos
docker-compose.yml Imágenes Público
Documentos load_data.sql snap
Escritorio Música test-docker.sh

El primer enfoque será copiar los datos de la máquina local al contenedor docker usando el comando "docker cp".
mysql-db = CONTENEDOR

jessica@ubu20drc:~$ docker cp load_data.sql mysql-db:/tmp
jessica@ubu20drc:~$ docker exec -it mysql-db bash
root@db48ed82b5d7:/# ls -l /tmp/
total 4
-rw-rw-rw- 1 1000 1000 578 Feb 28 16:29 load_data.sql
Ahora puede conectarse al cliente mysql y ejecutar el comando fuente o redirigir el archivo al cliente mysql.
root@db48ed82b5d7:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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

```
mysql> source /tmp/load_data.sql
Query OK, 1 row affected, 1 warning (0.01 sec)

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
Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 1 row affected (0.02 sec)

Query OK, 1 row affected (0.02 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.02 sec)

Query OK, 1 row affected (0.03 sec)
```

otro modo de ejecutarlo : _ *(Or: mysql -u root -p < /tmp/load_data.sql)*

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| demo               |
| football           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| tienda             |
+--------------------+
7 rows in set (0.00 sec)

mysql> use football;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_football |
+--------------------+
| players            |
+--------------------+
1 row in set (0.00 sec)
```

```
mysql> select*from players;
+-------------+------------+-------------+----------------+
| player_name | player_age | player_club | player_country |
+-------------+------------+-------------+----------------+
| Messi       |         34 | PSG         | Argentina      |
| Ronaldo     |         36 | MANU        | Portugal       |
| Neymar      |         29 | PSG         | Brazil         |
| Kane        |         28 | SPURS       | England        |
| E Hazard    |         30 | MADRID      | Belgium        |
| Messi       |         34 | PSG         | Argentina      |
| Ronaldo     |         36 | MANU        | Portugal       |
| Neymar      |         29 | PSG         | Brazil         |
| Kane        |         28 | SPURS       | England        |
| E Hazard    |         30 | MADRID      | Belgium        |
+-------------+------------+-------------+----------------+
10 rows in set (0.00 sec)

mysql>
```

Para proyecto

MIGRAR DE SQL A MYSQL USANDO WORKBENCH

lo instalamos en windows (el workbench) y migramos
crear base de datos sql en nuestro contenedor docker

DOCKER COMPOSE CON MYSQL

https://docs.docker.com/samples/wordpress/

cuando editas el archivo la versión es la 3.3

```
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS
PORTS                                       NAMES
db48ed82b5d7   mysql     "docker-entrypoint.s…"   13 days ago   Exited (255) 12 days
ago   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql-db
jessica@ubu20drc:~$ docker rm $(docker ps -qa)---> esto para borrar todo de forma
recursiva
db48ed82b5d7
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS   PORTS          NAMES
jessica@ubu20drc:~$ ls
```

Descargas Escritorio load_data.sql Plantillas snap
Documentos Imágenes Música Público Vídeos
jessica@ubu20drc:~$ cd ..
jessica@ubu20drc:/home$ cd ..
jessica@ubu20drc:/$ ls estamos en raiz lo creo desde ahi corvelle
bin cdrom etc lib lib64 lost+found mnt proc run snap swapfile tmp var
boot dev home lib32 libx32 media opt root sbin srv sys usr
jessica@ubu20drc:/$ sudo mkdir compose-wordpress creo el directorio

[sudo] contraseña para jessica:
jessica@ubu20drc:/$ cd compose-wordpress/
jessica@ubu20drc:/compose-wordpress$ sudo nano docker-compose.yml
jessica@ubu20drc:/compose-wordpress$ docker-compose up -d
ERROR: Version in "./docker-compose.yml" is unsupported. You might be seeing this error because you're using the wrong Compose file version. Either specify a supported version (e.g "2.2" or "3.3") and place your service definitions under the `services` key, or omit the `version` key and place your service definitions at the root of the file to use version 1.
For more on the Compose file for versions,https://docs.docker.com/compose/compose-file/
UTILIZAMOS LA VERSIÓN: 3.3
jessica@ubu20drc:/compose-wordpress$ sudo nano docker-compose.yml
root@ubu20drc:/compose-wordpress# ls
docker-compose.yml
jessica@ubu20drc:/compose-wordpress$ **docker-compose up -d #TIENES QUE PONER EL COMANDO EN DONDE ESTÉ EL ARCHIVO #**
Creating volume "compose-wordpress_db_data" with default driver
Creating volume "compose-wordpress_wordpress_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
f003217c5aae: Already exists
ba61822c65c2: Pull complete
dec59acdf78a: Pull complete
0a05235a6981: Pull complete
c87d621d6916: Pull complete
Digest: sha256:1a73b6a8f507639a8f91ed01ace28965f4f74bb62a9d9b9e7378d5f07fab79dc
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
c229119241af: Pull complete
47e86af584f1: Pull complete
0f5086159710: Pull complete
232f558e90bc: Pull complete
a20f94b57db2: Pull completeDigest:
sha256:6ecc3e3bea4d3bfa8cc5e481bf93d2d8abc0dd3e93920e8a296dc0106d9f7696
Status: Downloaded newer image for wordpress:latest
Creating compose-wordpress_db_1 ... done
Creating compose-wordpress_wordpress_1 ... done

COMPROBAMOS LA EXISTENCIA

—--

jessica@ubu20drc:/$ **docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

11890aaf7eb6 wordpress:latest "docker-entrypoint.s…" 9 minutes ago Up 9 minutes 0.0.0.0:8000->80/tcp, :::8000->80/tcp compose-wordpress_wordpress_1

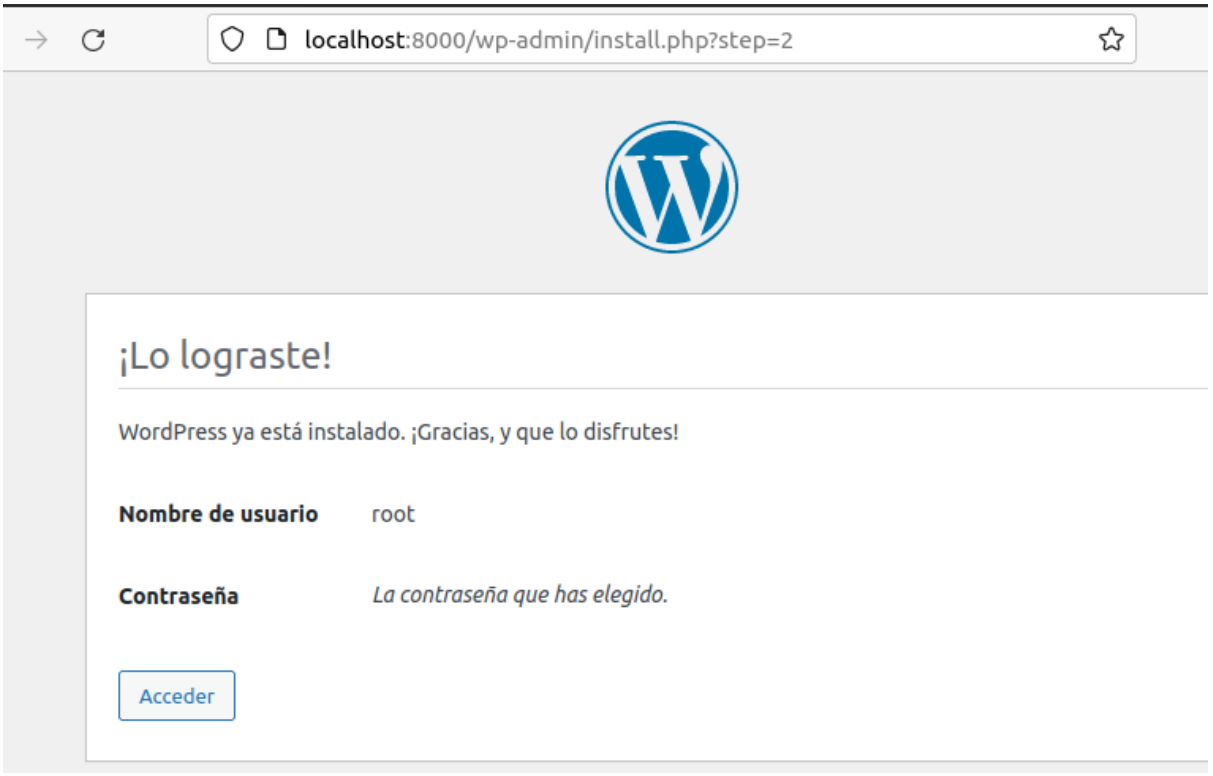
abd4a08e89cf mysql:5.7 "docker-entrypoint.s…" 9 minutes ago Up 9 minutes 3306/tcp, 33060/tcp compose-wordpress_db_1

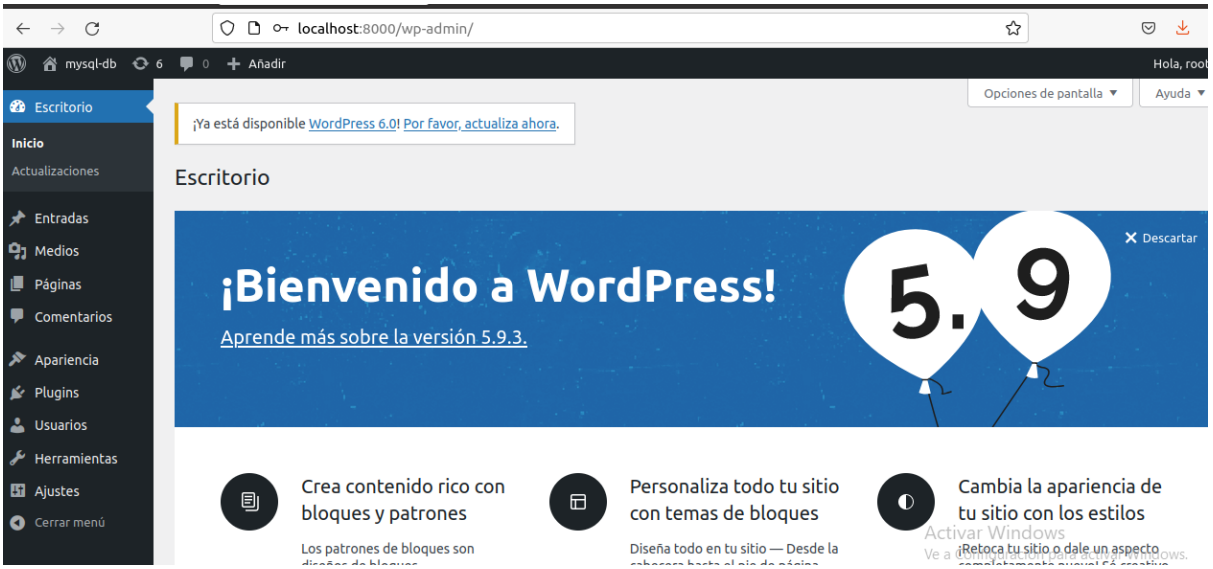
d3fc39727f6a mysql "docker-entrypoint.s…" 8 hours ago Up 8 hours 33060/tcp, 0.0.0.0:33060->3306/tcp, :::33060->3306/tcp mysql-db

—--

vamos a mozilla y comprobamos QUE CONECTA

http://localhost:8000 y enter


localhost:8000/wp-admin/install.php?step=2
¡Lo lograste!
WordPress ya está instalado. ¡Gracias, y que lo disfrutes!
Nombre de usuario
root
Contraseña
La contraseña que has elegido.
Acceder

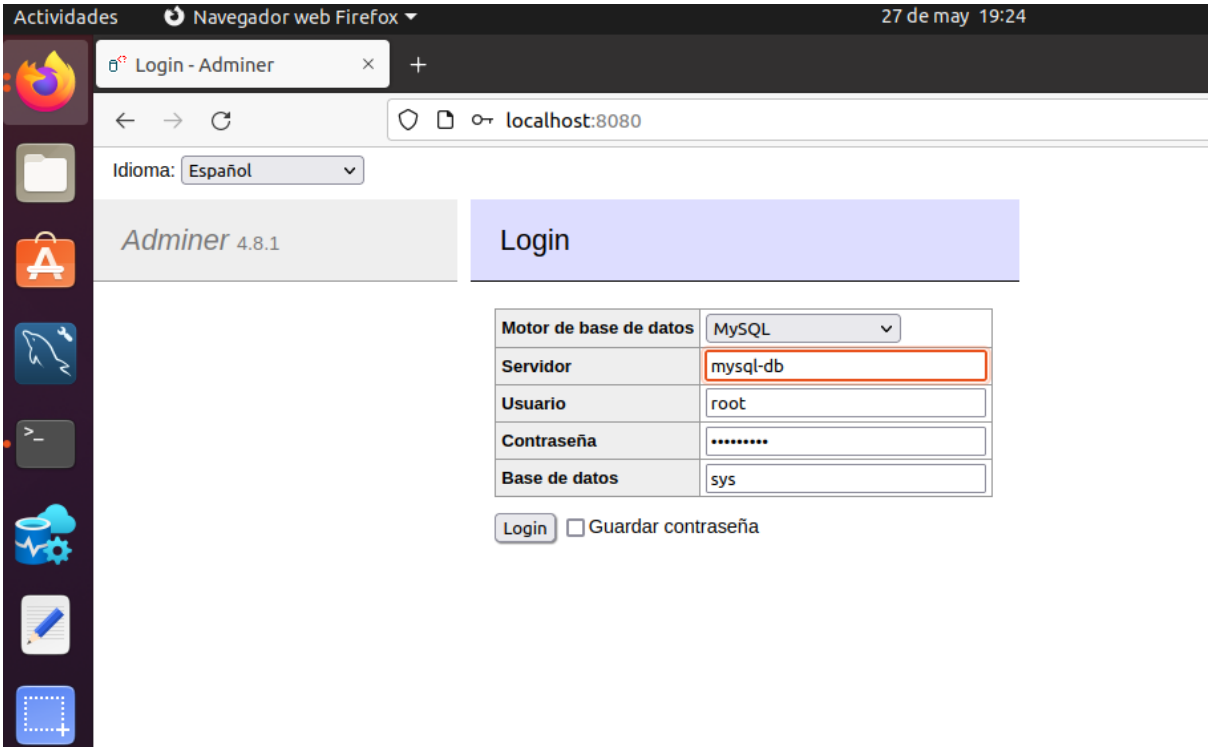

localhost:8000/wp-admin/
mysql-db
Añadir
Hola, root
Escritorio
Inicio
Actualizaciones
Entradas
Medios
Páginas
Comentarios
Apariencia
Plugins
Usuarios
Herramientas
Ajustes
Cerrar menú
Opciones de pantalla
Ayuda
¡Ya está disponible WordPress 6.0! Por favor, actualiza ahora.
Escritorio
¡Bienvenido a WordPress!
Aprende más sobre la versión 5.9.3.
Descartar
Crea contenido rico con bloques y patrones
Personaliza todo tu sitio con temas de bloques
Cambia la apariencia de tu sitio con los estilos
Activar Windows

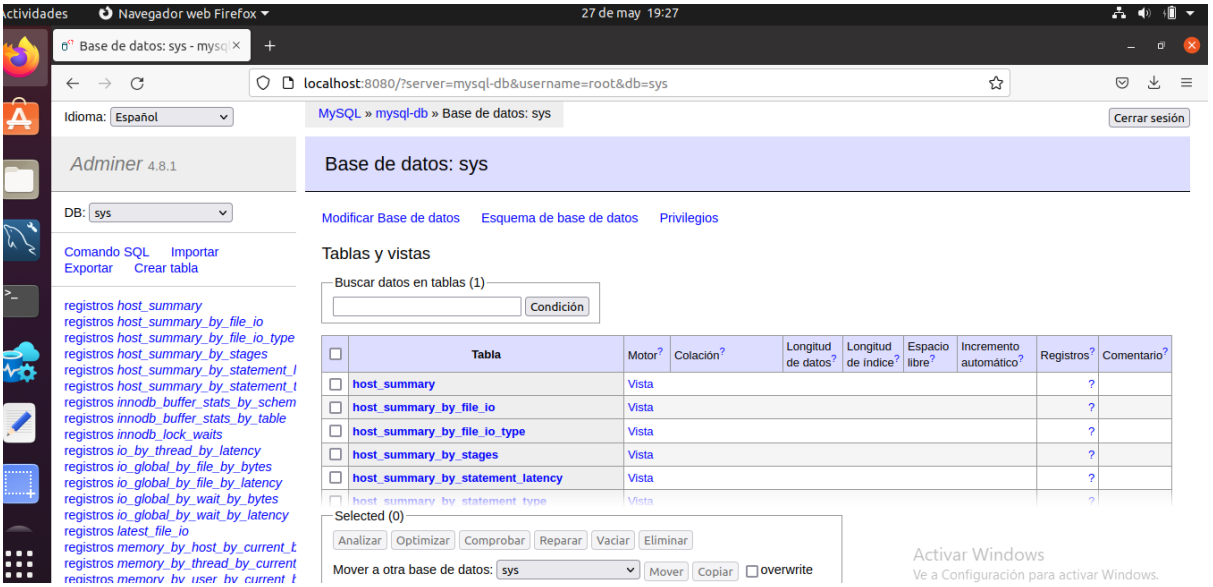
CONECTAR UN CONTENEDOR CON ADMINER CON MYSQL

PARA ESTE EJEMPLO UTILIZO EL CONTENEDOR DE **MYSQL-DB PARA APROVECHAR EL CONTENEDOR QUE YA CREE UN POCO MÁS ARRIBA Y PROBAR SI FUNCIONA EN VEZ DE HACER UNO NUEVO DE **Phpmyadmin.****

```
jessica@ubu20drc:~$ docker run -d \
> --rm \
> --link  mysql-db \
> -p 8080:8080 \
> adminer
Unable to find image 'adminer:latest' locally
latest: Pulling from library/adminer
2408cc74d12b: Pull complete
fde5ea1cb188: Pull complete
3935ba60366a: Pull complete
4712e34f1d29: Pull complete
38485f657c58: Pull complete
a95a057edf73: Pull complete
48767c9b5702: Pull complete
8cefbed5a861: Pull complete
2e1ccf753755: Pull complete
2e8f611b79fe: Pull complete
9eb1c0f3d65a: Pull complete
4f1e8aa8cfbe: Pull complete
418387a0394f: Pull complete
9b44cc5a1783: Pull complete
7ceff681fe6e: Pull complete
Digest: sha256:694911aafdb847a5753e437126e62be331ab3844ba1cbef5f3692da092f6f048
Status: Downloaded newer image for adminer:latest
80dd77a6e73ba3baa4fbba729b95398152f1bf0f3d60840c4fbd0f4803fdaeda
docker: Error response from daemon: Cannot link to a non running container: /mysql-db AS
/hardcore_almeida/mysql-db.
jessica@ubu20drc:~$
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS
PORTS                                        NAMES
11890aaf7eb6   wordpress:latest   "docker-entrypoint.s…"   17 hours ago   Up 9 minutes
0.0.0.0:8000->80/tcp, :::8000->80/tcp                    compose-wordpress_wordpress_1
abd4a08e89cf   mysql:5.7          "docker-entrypoint.s…"   17 hours ago   Up 9 minutes
3306/tcp, 33060/tcp                                      compose-wordpress_db_1
d3fc39727f6a   mysql              "docker-entrypoint.s…"   25 hours ago   Exited (255) 9
minutes ago   33060/tcp, 0.0.0.0:33060->3306/tcp, :::33060->3306/tcp   mysql-db
jessica@ubu20drc:~$ docker start mysql-db
```

```
mysql-db
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS
PORTS                                       NAMES
11890aaf7eb6   wordpress:latest   "docker-entrypoint.s…"   17 hours ago   Up 10 minutes
0.0.0.0:8000->80/tcp, :::8000->80/tcp       compose-wordpress_wordpress_1
abd4a08e89cf   mysql:5.7          "docker-entrypoint.s…"   17 hours ago   Up 10 minutes
3306/tcp, 33060/tcp                         compose-wordpress_db_1
d3fc39727f6a   mysql              "docker-entrypoint.s…"   25 hours ago   Up 5 seconds
33060/tcp, 0.0.0.0:33060->3306/tcp, :::33060->3306/tcp   mysql-db
jessica@ubu20drc:~$ docker run -d --rm --link  mysql-db -p 8080:8080 adminer
558c3353157dbcf55d7e906234fe862aa9d2029b95936e57e9a55fa702bd20fe
```





jessica@ubu20drc:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
558c3353157d adminer "entrypoint.sh docke…" 9 minutes ago Up 9 minutes 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp optimistic_cohen
11890aaf7eb6 wordpress:latest "docker-entrypoint.s…" 18 hours ago Up 20 minutes 0.0.0.0:8000->80/tcp, :::8000->80/tcp compose-wordpress_wordpress_1
abd4a08e89cf mysql:5.7 "docker-entrypoint.s…" 18 hours ago Up 20 minutes 3306/tcp, 33060/tcp compose-wordpress_db_1
d3fc39727f6a mysql "docker-entrypoint.s…" 25 hours ago Up 9 minutes 33060/tcp, 0.0.0.0:33060->3306/tcp, :::33060->3306/tcp mysql-db

ADMINER_PHPMYADMIN_WORPRESS:SCRIPT

CREAMOS UNA RED

PARA QUE SE COMUNIQUEN DOS CONTENEDORES

jessica@ubu20drc:~$ docker network create my-net
e977722ee943e539fc2d79d1ab3d31560d3256855be3e3b223876038d7d2c1a4

CREAMOS LOS CONTENEDORES PARA CONECTAR EN ESA RED

```
jessica@ubu20drc:~$ docker run -d \
> --rm \
> --name mysqlc \
> --network my-net \
> -p 3306:3306 \
> -e MYSQL_ROOT_PASSWORD=Abcd1234. \
> -v mysql_data:/var/lib/mysql \
> mysql:5.7.28
1ba6200d528b268a4a61913975e7b8900839c4b0810ce4f162ec6b4d1d9ac1fa
```

```
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
PORTS                                       NAMES
1ba6200d528b   mysql:5.7.28   "docker-entrypoint.s…"   34 seconds ago   Up 31 seconds
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysqlc
```

Creamos Un Contenedor Con Adminer Indicando Que Queremos Que Esté En La Red --Network My-Net.

```
jessica@ubu20drc:~$ docker run -d \
> --rm \
> --network my-net \
> -p 8080:8080 \
> adminer
33932c1350328ed0e64bd5f6c92acb6d9c27b8e5fc88a2e4d882d1941ec14b6f
```

Comprobamos que están los dos contenedores creados:

```
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS
PORTS                                       NAMES
33932c135032   adminer        "entrypoint.sh docke…"   About a minute ago   Up About a
minute   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp             affectionate_bardeen
1ba6200d528b   mysql:5.7.28   "docker-entrypoint.s…"   8 minutes ago       Up 8 minutes
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysqlc
```

Borramos :

```
jessica@ubu20drc:~$ docker rm -f $(docker ps -qa)
33932c135032
1ba6200d528b
jessica@ubu20drc:~$ docker network ls
NETWORK ID     NAME                    DRIVER    SCOPE
6936c4f5ec9c   bridge                  bridge    local
7a20f9761a23   compose-mongo_default   bridge    local
b10029cff76f   compose-wordpress_default   bridge    local
da1b0d090460   host                    host      local
e977722ee943   my-net                  bridge    local
11c0cab8076a   none
```

Para eliminar la red que hemos creado ejecutamos lo siguiente

```
jessica@ubu20drc:~$ docker network rm my-net
my-net
```

```
jessica@ubu20drc:~$ docker network ls
NETWORK ID     NAME                          DRIVER    SCOPE
6936c4f5ec9c   bridge                        bridge    local
7a20f9761a23   compose-mongo_default         bridge    local
b10029cff76f   compose-wordpress_default     bridge    local
da1b0d090460   host                          host      local
11c0cab8076a   none                          null      local
jessica@ubu20drc:~$
```


EJERCICIO CON MÚLTIPLES CONTENEDORES: WORDPRESS + MYSQL + PHPMYADMIN

1) Crea una user-defined bridge network para todos los contenedores. Por ejemplo, esta red se puede llamar wordpress-net.

```
jessica@ubu20drc:~$ docker network create wordpress-net
e3e8f9c3b3ecc244bed3d11fdffaa283b62929175887e80845900ce82a49add1
```

2) Crea un volumen nuevo para almacenar los datos de MySQL. Por ejemplo, este volumen se puede llamar wordpress mysql data.

1º controlamos la existencia

```
jessica@ubu20drc:~$ docker volume create wordpress_mysql_data
wordpress_mysql_data
```

3) Crea una instancia de un contenedor con MySQL con las siguientes características:

-Se ejecuta en modo detached (background).
-Está en la red wordpress-net.
-Redirige el puerto 3306 del host al puerto 3306 del contenedor.
-Crea la variable de entorno MYSQL_ROOT_PASSWORD y asígnale un valor.
-Crea la variable de entorno MYSQL_DATABASE y asígnale un valor.
-Crea la variable de entorno MYSQL_USER y asígnale un valor.
-Crea la variable de entorno MYSQL_PASSWORD y asígnale un valor.

-Usa el volumen que creaste en el paso 2 (wordpress_mysql_data) para montarlo en el directorio /var/lib/mysql del contenedor.

```
root@ubu20drc:/home/jessica# docker run -d \
--rm \
 --name mysqlc \
 --network wordpress-net \
 -p 3306:3306 \
 -e MYSQL_ROOT_PASSWORD=root \
 -e MYSQL_DATABASE=wp_database \
```

```
 -e MYSQL_USER=wp_user \
 -e MYSQL_PASSWORD=wp_password \
 -v wordpress_mysql_data:/var/lib/mysql \
 mysql
da2ab1c6ecc0301f7621e880ae030af765c7257df00401872969846e0f9a6e54
```

4) Crea una instancia de un contenedor con phpMyAdmin con las siguientes características:

Se ejecuta en modo detached (background).
Está en la red wordpress-net.
Redirige el puerto 8080 del host al puerto 80 del contenedor.
Crea la variable de entorno PMA_ARBITRARY y asígnale el valor 1, para que nos permita indicar el nombre del servidor de base de datos al que queremos conectarnos.

```
root@ubu20drc:/home/jessica# docker run -d \
 --rm \
 --network wordpress-net \
 -e PMA_ARBITRARY=1 \
 -p 8080:80 \
 phpmyadmin/phpmyadmin
Unable to find image 'phpmyadmin/phpmyadmin:latest' locally
latest: Pulling from phpmyadmin/phpmyadmin
214ca5fb9032: Already exists
cd813a1b2cb8: Already exists
63cf7574573d: Already exists
54c27146d16e: Already exists
078f4450f949: Already exists
(...)
Digest: sha256:ae6dadd9cf3c158e42937788f7255fa820ea3daef0349226d8d43f32e76535e1
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
9966a3744a7de90885b671c58618b9e7ab91fb9e9eaf437a91ce551716d14306
```

5) Crea una instancia de un contenedor con WordPress con las siguientes características:

Se ejecuta en modo detached (background).
Está en la red wordpress-net.
Redirige el puerto 80 del host al puerto 80 del contenedor.
Crea la variable de entorno WORDPRESS_DB_HOST y asígnale el nombre del contenedor que está ejecutando MySQL.
Crea la variable de entorno WORDPRESS_DB_NAME y asígnale el valor de la base de datos que has creado en la instancia de MySQL.
Crea la variable de entorno WORDPRESS_DB_USER y asígnale el valor del usuario de la base de datos que has creado en la instancia de MySQL.
Crea la variable de entorno WORDPRESS_DB_PASSWORD y asígnale el valor de la contraseña del usuario de la base de datos que has creado en la instancia de MySQL.

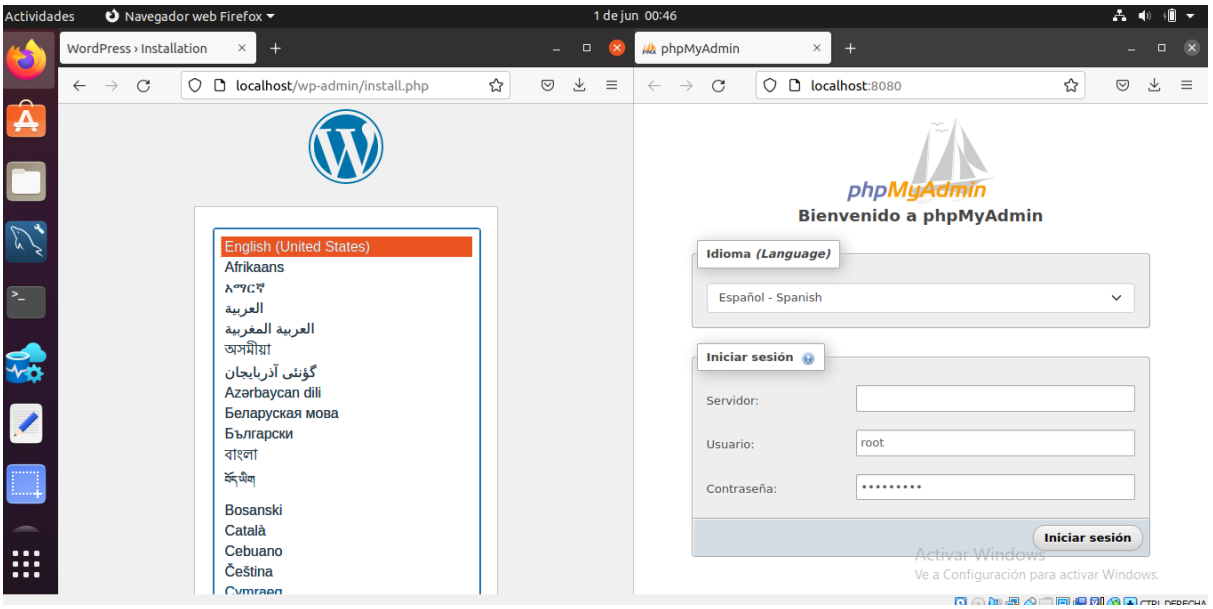
Necesitaremos crear un volumen (wordpress_data) para montarlo en el directorio /var/www/html del contenedor.

```
jessica@ubu20drc:~$  docker run -d \
>  --rm \
>  --name wordpressc \
>  --network wordpress-net \
>  -p 80:80 \
>  -e WORDPRESS_DB_HOST=mysqlc \
>  -e WORDPRESS_DB_NAME=wp_database \
>  -e WORDPRESS_DB_USER=wp_user \
>  -e WORDPRESS_DB_PASSWORD=wp_password \
>  -v wordpress_data:/var/www/html \
>  wordpress
7b2ac33f7d7767077afda95a4775c5b89c942a07dba678d4d321e4db782ddfd7
```

COMPROBAMOS SU EXISTENCIA

```
jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS
PORTS                                      NAMES
c1e6eb0951fd   phpmyadmin/phpmyadmin   "/docker-entrypoint.…"   23 seconds ago   Up
20 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp              mystifying_meitner
c9e64e7c5c25   mysql                  "docker-entrypoint.s…"   2 minutes ago    Up 2 minutes
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysqlc
7b2ac33f7d77   wordpress              "docker-entrypoint.s…"   26 minutes ago   Up 26
minutes   0.0.0.0:80->80/tcp, :::80->80/tcp                  wordpressc
```

PRUEBA NAVEGADOR:



1.usar en clase doccker sql server.txt

SQL SERVER

Descargamos el SQL Server:

```
root@ubu20drc:/home/jessica# docker run -e 'ACCEPT_EULA=Y' -e
'MSSQL_SA_PASSWORD=Abcd1234.' \
>    --name 'sql1' -p 1401:1433 \
>    -v sql1data:/var/opt/mssql \
>    -d mcr.microsoft.com/mssql/server:2019-latest
Unable to find image 'mcr.microsoft.com/mssql/server:2019-latest' locally
2019-latest: Pulling from mssql/server
5e51d371b4de: Pull complete
98f9a4e5d42f: Pull complete
239fcd477f4b: Pull complete
Digest:
sha256:49a57dc220b1ea9d5bbdb95a98a27b3ed2c50e164c643eefab2e985e798f0b6b
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2019-latest
8daeb34ae608f79f66c8e725c051c63d64e129a494cc2b49dd5266aba4f6b1a5
root@ubu20drc:/home/jessica# docker ps
CONTAINER ID   IMAGE                                  COMMAND                  CREATED
STATUS              PORTS                                       NAMES
8daeb34ae608   mcr.microsoft.com/mssql/server:2019-latest   "/opt/mssql/bin/perm…"   2
minutes ago   Up About a minute   0.0.0.0:1401->1433/tcp, :::1401->1433/tcp   sql1
c1e6eb0951fd   phpmyadmin/phpmyadmin
```

-Listamos el log de errores de mssql de dentro del contenedor:
-**Docker exec** vale para dar sentencias **dentro** del contenedor, ejecutar comandos dentro del contenedor.
-sql1 : el nombre del contenedor.

jessica@ubu20drc:~$ docker exec -t sql1 cat /var/opt/mssql/log/errorlog | grep connection
2022-04-28 16:30:04.36 Server The maximum number of dedicated administrator connections for this instance is '1'
2022-04-28 16:30:12.97 spid43s Always On: The availability replica manager is waiting for the instance of SQL Server to allow client connections. This is an informational message only. No user action is required.
2022-04-28 16:30:13.72 Server Dedicated admin connection support was established for listening locally on port 1434.
2022-04-28 16:30:13.80 spid42s SQL Server is now ready for client connections. This is an informational message; no user action is required.

-Ejecutamos el bash de dentro del contenedor:
jessica@ubu20drc:~$ docker exec -it sql1 "bash"
mssql@419fff2abe50:/$ ls
bin dev home lib32 libx32 mnt proc run srv tmp var
boot etc lib lib64 media opt root sbin sys usr
-Esto solo es para buscar la ruta, de ahí el asterisco que no se recuerda el nombre y nos devuelve la ruta correcta de sqlcmd (para encontrar el ejecutable):
mssql@419fff2abe50:/$**ls /opt/mssql-tools/bin/sqlcmd***
/opt/mssql-tools/bin/sqlcmd
mssql@419fff2abe50:/$ **/opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P "Abcd1234."**
1> ← *nos devuelve el prompt de sqlcmd*
-Comprobación de que funciona bien (Un ejemplo cualquiera para ver que funcione, si nos devuelve algo es que funciona:
1> select @@SERVERNAME
2> go
2810f76eab2d

(1 rows affected)

-Hacemos 2 exit, uno para salir de sqlcmd y otro para salir del contenedor:
2> exit
exit

-Volvemos a entrar y hacemos pruebas varias:
root@ubu20drc:/home/jessica# docker exec -it sql1 "bash"
mssql@8daeb34ae608:/$ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P "abcd1234."
1> create database agentesforestales;
2> select name from sys.databases;
3> go
name

```
------------------------------------------------------------------------------------------------------------------------------
master
tempdb
model
msdb
TestDB
agentesforestales


(6 rows affected)
1> usd agentesforestales
2> ATE TABLE Inventory (id INT, name NVARCHAR(50), quantity INT);
3> SELECT * FROM Inventory WHERE quantity > 152;
4> GO
Changed database context to 'agentesforestales'.
id          name                                               quantity
------------------------------------------------------------------------------------------------------------------------------


(0 rows affected)
1>
```

BACKUP

-docker exec para crear una carpeta de copia de seguridad.
El siguiente comando crea un directorio /var/opt/mssql/backup dentro del contenedor de SQL Server.

```
root@ubu20drc:/home/jessica# docker exec -it sql1 mkdir /var/opt/mssql/backup
```

- descargamos el archivo WideWorldImporters-Full.bak en el equipo host. Los siguientes comandos llevan al directorio home/user y descargan el archivo de copia de seguridad como wwi.bak.

```
jessica@ubu20drc:~$  curl -L -o wwi.bak
'https://github.com/Microsoft/sql-server-samples/releases/download/wide-world-importers-v1.
0/WideWorldImporters-Full.bak'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0     0    0     0    0     0      0      0 --:--:--
--:--:-- --:--:--     0
  0  121M    0 33373    0     0  44085      0  0:48:02 --:--:--  0  2  121M    2 3543k    0     0
2019k      0  0:01:01  0:00:01  0  5  121M    5 6663k    0     0  2417k      0  0:00:51  0:00:02  0
```

-docker cp para copiar el archivo de copia de seguridad en el contenedor, en el directorio /var/opt/mssql/backup.

docker cp wwi.bak sql1:/var/opt/mssql/backup

Notas en scripts:

Restauración de la base de datos

El archivo de copia de seguridad ahora se encuentra dentro del contenedor. Antes de restaurar la copia de seguridad, es importante conocer los nombres de archivo lógicos y los tipos de archivo que hay dentro de la copia de seguridad. Los siguientes comandos de Transact-SQL examinan la copia de seguridad y realizan la restauración con sqlcmd en el contenedor.

Sugerencia:

En este tutorial se usa sqlcmd dentro del contenedor, ya que este incluye esta herramienta preinstalada. Pero también puede ejecutar instrucciones de Transact-SQL con otras herramientas de cliente fuera del contenedor, como Visual Studio Code o SQL Server Management Studio. Para conectarte, usa el puerto de host que se ha asignado al puerto 1433 en el contenedor. En este ejemplo es localhost,1401 en el equipo host y Host_IP_Address,1401 de forma remota.

Ejecutamos sqlcmd dentro del contenedor para enumerar los nombres de archivo lógicos y las rutas de acceso que hay dentro de la copia de seguridad. Esto se hace con la instrucción de Transact-SQL RESTORE FILELISTONLY.

```
jessica@ubu20drc:~$ sudo docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd -S localhost \
>    -U SA -P "abcd1234." \
>    -Q 'RESTORE FILELISTONLY FROM DISK = "/var/opt/mssql/backup/wwi.bak"' \
>    | tr -s ' ' | cut -d ' ' -f 1-2
[sudo] contraseña para jessica:
LogicalName PhysicalName
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
WWI_Primary D:\Data\WideWorldImporters.mdf
WWI_UserData D:\Data\WideWorldImporters_UserData.ndf
WWI_Log E:\Log\WideWorldImporters.ldf
WWI_InMemory_Data_1 D:\Data\WideWorldImporters_InMemory_Data_1

(4 rows)
```

RESTORE DATABASE

para restaurar la base de datos dentro del contenedor.

LEVANTO A UP EL CONTENEDOR PREVIAMENTE

```
oot@ubu20drc:/home/jessica# docker ps -a
CONTAINER ID   IMAGE                                   COMMAND                  CREATED
STATUS         PORTS                                   NAMES
8daeb34ae608   mcr.microsoft.com/mssql/server:2019-latest   "/opt/mssql/bin/perm…"   8
hours ago   Up 7 seconds   0.0.0.0:1401->1433/tcp, :::1401->1433/tcp   sql1
root@ubu20drc:/home/jessica# docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd \
>    -S localhost -U SA -P "abcd1234." \
>    -Q 'RESTORE DATABASE WideWorldImporters FROM DISK =
"/var/opt/mssql/backup/wwi.bak" WITH MOVE "WWI_Primary" TO
"/var/opt/mssql/data/WideWorldImporters.mdf", MOVE "WWI_UserData" TO
"/var/opt/mssql/data/WideWorldImporters_userdata.ndf", MOVE "WWI_Log" TO
"/var/opt/mssql/data/WideWorldImporters.ldf", MOVE "WWI_InMemory_Data_1" TO
"/var/opt/mssql/data/WideWorldImporters_InMemory_Data_1"'
Processed 1464 pages for database 'WideWorldImporters', file 'WWI_Primary' on file 1.
Processed 53096 pages for database 'WideWorldImporters', file 'WWI_UserData' on file 1.
Processed 33 pages for database 'WideWorldImporters', file 'WWI_Log' on file 1.
Processed 3862 pages for database 'WideWorldImporters', file 'WWI_InMemory_Data_1' on
file 1.
Converting database 'WideWorldImporters' from version 852 to the current version 904.
Database 'WideWorldImporters' running the upgrade step from version 852 to version 853.
RESTORE DATABASE successfully processed 58455 pages in 32.442 seconds (14.076
MB/sec).
```

-Comprobar la base de datos restaurada
Ejecutamos la consulta siguiente para mostrar una lista de nombres de bases de datos del contenedor

```
root@ubu20drc:/home/jessica# docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd \
>    -S localhost -U SA -P "abcd1234." \
>    -Q 'SELECT Name FROM sys.Databases'
Name
--------------------------------------------------------------------------------------------------------------------------------
master
tempdb
model
msdb
TestDB
agentesforestales
WideWorldImporters

(7 rows affected)
```

-Consulta para ver los diez elementos principales de la tabla Warehouse.StockItems.

```
jessica@ubu20drc:~$ docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd    -S localhost -U SA
-P "abcd1234."    -Q 'SELECT TOP 10 StockItemID, StockItemName FROM
WideWorldImporters.Warehouse.StockItems ORDER BY StockItemID'
StockItemID StockItemName
----------- ----------------------------------------------------------------------------------------------------
          1 USB missile launcher (Green)
          2 USB rocket launcher (Gray)
          3 Office cube periscope (Black)
          4 USB food flash drive - sushi roll
          5 USB food flash drive - hamburger
          6 USB food flash drive - hot dog
          7 USB food flash drive - pizza slice
          8 USB food flash drive - dim sum 10 drive variety pack
          9 USB food flash drive - banana
         10 USB food flash drive - chocolate bar

(10 rows affected)
```

-Actualizamos la descripción del primer elemento con la siguiente instrucción UPDATE:

```
jessica@ubu20drc:~$ docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd \
>    -S localhost -U SA -P "abcd1234." \
>    -Q 'UPDATE WideWorldImporters.Warehouse.StockItems SET StockItemName="USB
missile launcher (Dark Green)" WHERE StockItemID=1; SELECT StockItemID,
StockItemName FROM WideWorldImporters.Warehouse.StockItems WHERE
StockItemID=1'

(1 rows affected)
StockItemID StockItemName
----------- ----------------------------------------------------------------------------------------------------
          1 USB missile launcher (Dark Green)

(1 rows affected)
```

-Crear una nueva copia de seguridad

-crea un nuevo archivo de copia de seguridad wwi_2.bak en el directorio /var/opt/mssql/backup creado previamente.

```
jessica@ubu20drc:~$ docker exec -it sql1 /opt/mssql-tools/bin/sqlcmd \
>    -S localhost -U SA -P "abcd1234." \
>    -Q "BACKUP DATABASE [WideWorldImporters] TO DISK =
N'/var/opt/mssql/backup/wwi_2.bak' WITH NOFORMAT, NOINIT, NAME =
'WideWorldImporters-full', SKIP, NOREWIND, NOUNLOAD, STATS = 10"
10 percent processed.
20 percent processed.
30 percent processed.
40 percent processed.
50 percent processed.
60 percent processed.
Processed 1560 pages for database 'WideWorldImporters', file 'WWI_Primary' on file 1.
Processed 53104 pages for database 'WideWorldImporters', file 'WWI_UserData' on file 1.
70 percent processed.
Processed 3865 pages for database 'WideWorldImporters', file 'WWI_InMemory_Data_1' on
file 1.
Processed 171 pages for database 'WideWorldImporters', file 'WWI_Log' on file 1.
100 percent processed.
BACKUP DATABASE successfully processed 58700 pages in 25.437 seconds (18.028
MB/sec).
```

-Copiamos el archivo de copia de seguridad del contenedor en el equipo host.
cd ~

```
jessica@ubu20drc:~$ cd ~
jessica@ubu20drc:~$ docker cp sql1:/var/opt/mssql/backup/wwi_2.bak wwi_2.bak

jessica@ubu20drc:~$ ls -l wwi*
-rw-r----- 1 jessica jessica 483926016 jun  1 21:53 wwi_2.bak
-rw-rw-r-- 1 jessica jessica 127056896 jun  1 10:28 wwi.bak
```

-Uso de los datos almacenados

Además de realizar copias de seguridad de las bases de datos para proteger los datos, puede usar contenedores de volúmenes de datos. Al principio de este tutorial se ha creado el contenedor sql1 con el parámetro -v sql1data:/var/opt/mssql.

El contenedor de volúmenes de datos sql1data almacena los datos de /var/opt/mssql incluso después de quitar el contenedor. **En los pasos siguientes se quita por completo el contenedor sql1 y, luego, se crea un nuevo contenedor, sql2, con los datos almacenados.**

jessica@ubu20drc:~$ docker stop sql1
sql1
jessica@ubu20drc:~$ docker rm sql1
sql1
YA NO APARECE EL CONTENEDOR SQL1
jessica@ubu20drc:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

-Crear un nuevo contenedor, sql2, y volver a usar el contenedor de volúmenes de datos sql1data.

jessica@ubu20drc:~$ docker run -e 'ACCEPT_EULA=Y' -e "MSSQL_SA_PASSWORD=abcd1234." \
> --name 'sql2' -e 'MSSQL_PID=Developer' -p 1401:1433 \
> -v sql1data:/var/opt/mssql -d mcr.microsoft.com/mssql/server:2019-latest
c60162e3f0c7bd060e600aab1361718ebcc551c3b7182d085cf6cf2c59875aaa
COMPROBAMOS QUE APARECE EL CONTENEDOR SQL2
jessica@ubu20drc:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c60162e3f0c7 mcr.microsoft.com/mssql/server:2019-latest "/opt/mssql/bin/perm…" 33 seconds ago Up 30 seconds 0.0.0.0:1401->1433/tcp, :::1401->1433/tcp sql2

jessica@ubu20drc:~$ docker inspect sql2
solo muestro la parte MOUNTS

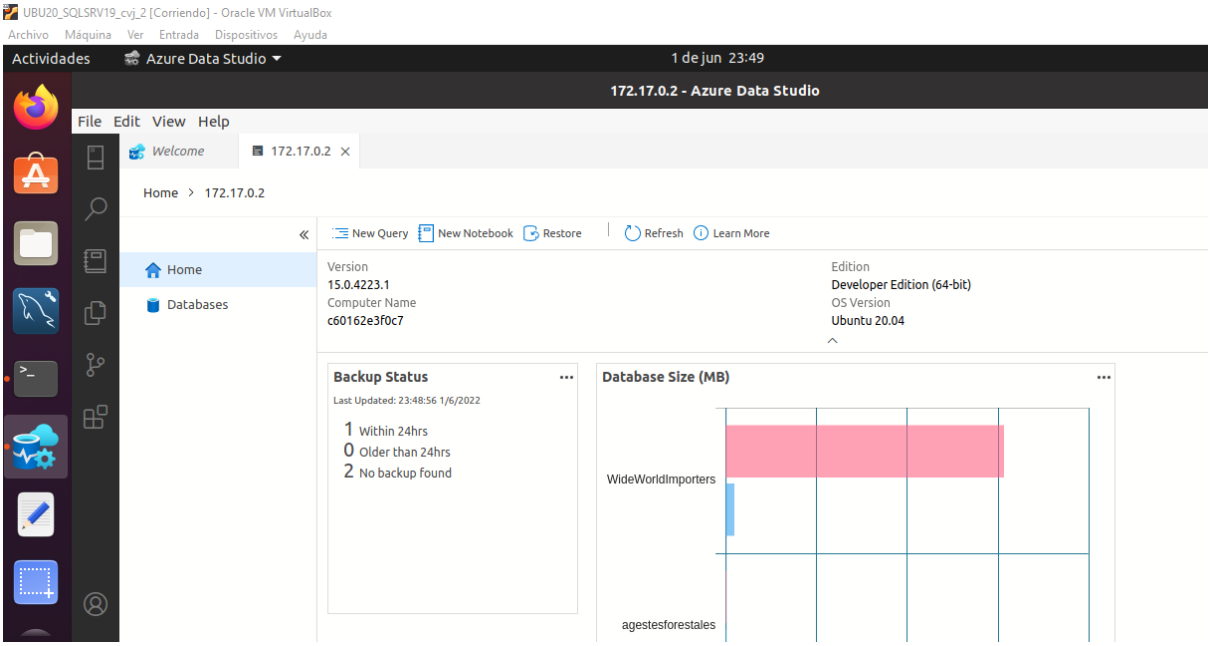
```
        "Mounts": [
            {
                "Type": "volume",
                "Name": "sql1data",
                "Source": "/var/lib/docker/volumes/sql1data/_data",
                "Destination": "/var/opt/mssql",
                "Driver": "local",
                "Mode": "z",
                "RW": true,
                "Propagation": ""
            }
```

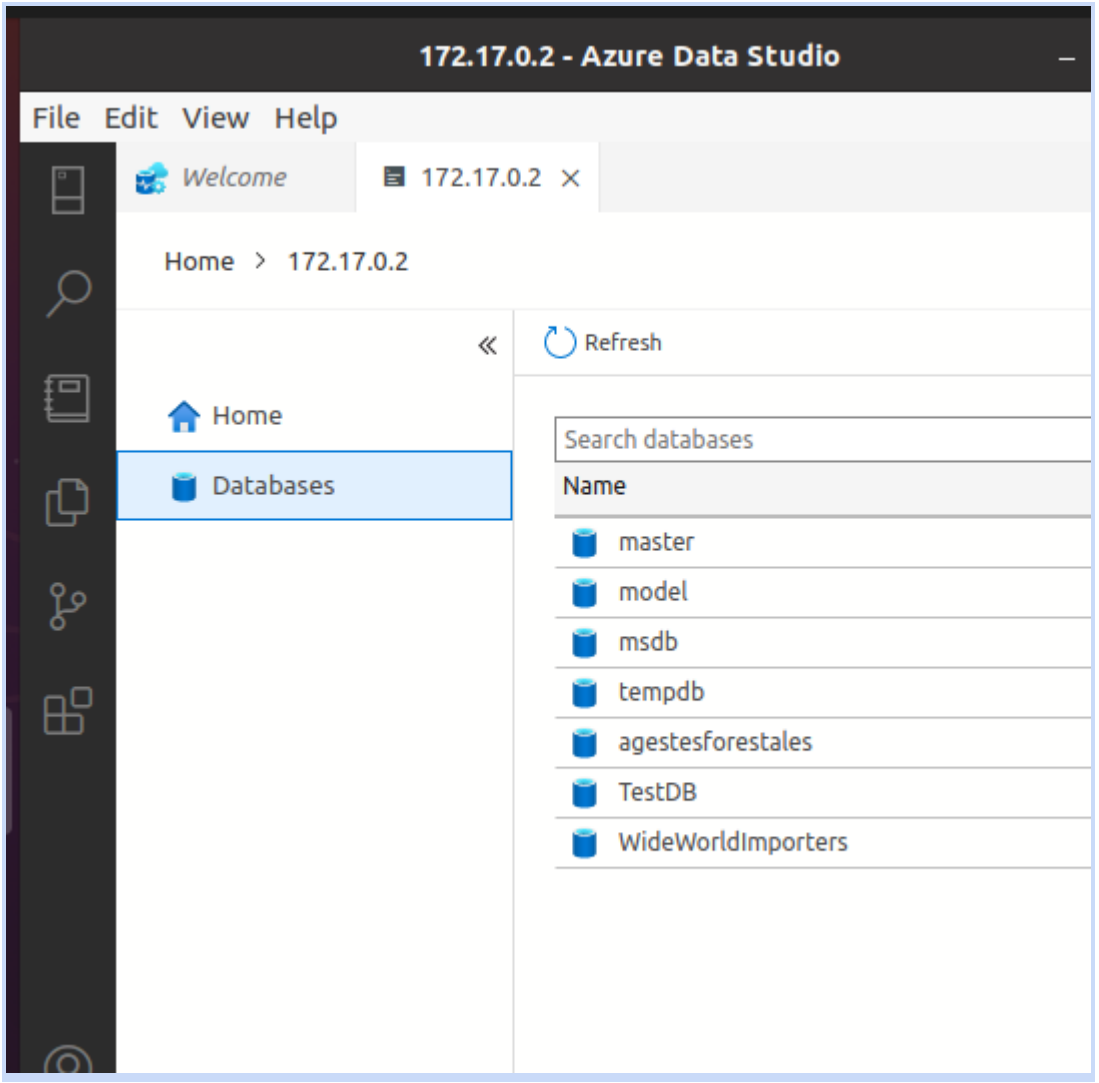
-La base de datos Wide World Importers está ahora en el nuevo contenedor.Comprobamos el cambio realizado.

```
jessica@ubu20drc:~$ docker exec -it sql2 /opt/mssql-tools/bin/sqlcmd    -S localhost -U SA
-P "abcd1234."    -Q 'SELECT StockItemID, StockItemName FROM
WideWorldImporters.Warehouse.StockItems WHERE StockItemID=1'
StockItemID StockItemName
---------- ----------------------------------------------------------------------------------------------------
         1 USB missile launcher (Dark Green)

(1 rows affected)

jessica@ubu20drc:~$ docker inspect sql2

            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID":
"e77e247923ab7d7b532d3603156b623ed3dd0bf28ce35ffdff92354fc6b2cb60",
                    "EndpointID":
"c620101bfe8f4d2304c8f487b7bb94fa1a080a9180d4216f1e9e892b609e7012",
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.2",
```

CONECTARSE A AZURE DATA STUDIO

IP 172.17.0.2
sa
abcd1234.





MONGODB

-d = (detach) se ejecuta en background de esta manera podemos seguir utilizando la misma terminal, de no ser así tendremos que abrir otra terminal para lanzar comandos.

```
jessica@ubu20drc:~$ docker run -d -p 27017:27017 -v mongodbdata:/data/db mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
d5fd17ec1767: Pull complete
568df027d825: Pull complete
Digest: sha256:50d8918de7b076feceb9ba1ee264afd5f67fb4baaff07949f3b9de92cdca79c2
Status: Downloaded newer image for mongo:latest
fba7df26761517d87ae210be2703b7906f78995c277e0eb3ba78d02e6901e0a3

jessica@ubu20drc:~$ docker ps -l
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS
NAMES
fba7df267615   mongo     "docker-entrypoint.s…"   7 minutes ago   Up 7 minutes
0.0.0.0:27017->27017/tcp, :::27017->27017/tcp   condescending_vaughan
```

EJECUTAMOS LA SESIÓN CON EL CLIENTE MONGO

essica@ubu20drc:~$ **docker exec -it fba7df267615 mongo**
MongoDB shell version v5.0.8
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("507a49b0-94a9-4f55-af06-24be5a1d6cea") }
MongoDB server version: 5.0.8

-CREAMOS UNA BD, INSERTAMOS ALGÚN DATO Y HACEMOS LA COMPROBACIÓN DE SU EXISTENCIA.

> db.version()
5.0.8
> **show dbs**
admin 0.000GB
config 0.000GB
local 0.000GB
películas 0.000GB
> **use agentesforestales**;
switched to db agentesforestales
> **db.agentes.save({ccaa:'Galicia'});**
WriteResult({ "nInserted" : 1 })
> db.agentes.find();
{ "_id" : ObjectId("6297f4c7664218ce609665a0"), "ccaa" : "Galicia" }
> **show dbs;**
admin 0.000GB
agentesforestales 0.000GB
config 0.000GB
local 0.000GB
películas 0.000GB
> exit
jessica@ubu20drc:~$ **docker rm -f fba7df267615**
fba7df267615

COMPROBAMOS SI PERSISTE:
jessica@ubu20drc:~$ docker run -d -p 27017:27017 -v mongodbdata:/data/db mongo
087ec937c88e5808d28e606098476df971e93c090bcfd65fe8159c75099465f4
jessica@ubu20drc:~$ docker ps -l
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
087ec937c88e mongo "docker-entrypoint.s…" 17 seconds ago Up 14 seconds 0.0.0.0:27017->27017/tcp, :::27017->27017/tcp upbeat_tesla
jessica@ubu20drc:~$ docker exec -it 087ec937c88e

"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
jessica@ubu20drc:~$ docker exec -it 087ec937c88e mongo
MongoDB shell version v5.0.8
	To enable free monitoring, run the following command:
db.enableFreeMonitoring()
	To permanently disable this reminder, run the following command:
db.disableFreeMonitoring()

> show dbs
admin	0.000GB
agentesforestales	0.000GB ←— AQUI SE VE LA BD CREADA ANTERIORMENTE
config	0.000GB
local	0.000GB
películas	0.000GB
>

—----------

COMPOSE-MONGO

Crear con persistencia

root@ubu20drc:/home/jessica# docker ps
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
root@ubu20drc:/home/jessica# docker run --name some-mongo -d mongo
f6b2abad862d4607e8cdc712e8fbb09b2aadebfdd366135a729e599f743f760c
root@ubu20drc:/home/jessica# docker images
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mongo	latest	7babdf608274	6 days ago	700MB
wordpress	latest	7b732547202e	2 weeks ago	606MB
mysql	5.7	f26e21ddd20d	5 weeks ago	450MB
mysql	latest	667ee8fb158e	5 weeks ago	521MB
mcr.microsoft.com/mssql/server	2019-latest	d78e982c2f2b	3 months ago	1.48GB
mysql	5.7.28	db39680b63ac	2 years ago	437MB
root@ubu20drc:/home/jessica# docker ps
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
f6b2abad862d   mongo   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   27017/tcp   some-mongo

root@ubu20drc:/home/jessica# **docker exec -it f6b2abad862d mongo**
MongoDB shell version v5.0.8
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("77dc9700-8577-4ba0-a3f2-76b24d043222") }
MongoDB server version: 5.0.8
    https://community.mongodb.com
(...)

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command:
db.disableFreeMonitoring()

>
> cd
[native code]
> ls
[native code]
> exit
bye
root@ubu20drc:/home/jessica# exit
exit

jessica@ubu20drc:~$ ls
Descargas Escritorio load_data.sql Plantillas snap
Documentos Imágenes Música Público Vídeos
jessica@ubu20drc:~$ cd ..
jessica@ubu20drc:/home$ cd ..
jessica@ubu20drc:/$ ls
bin cdrom dev home lib32 libx32 media opt root sbin srv sys usr
boot compose-wordpress etc lib lib64 lost+found mnt proc run snap swapfile tmp
var
CREAMOS UN DIRECTORIO
jessica@ubu20drc:/$ sudo mkdir compose-mongo
[sudo] contraseña para jessica:
jessica@ubu20drc:/$ ls
bin compose-mongo etc lib32 lost+found opt run srv tmp
boot compose-wordpress home lib64 media proc sbin swapfile usr
cdrom dev lib libx32 mnt root snap sys var
LE DAMOS PERMISO AL DIRECTORIO
jessica@ubu20drc:/$ sudo chmod 777 compose-mongo/
jessica@ubu20drc:/$ cd compose-mongo/
jessica@ubu20drc:/compose-mongo$ ls
jessica@ubu20drc:/compose-mongo$ ls /compose-wordpress/
docker-compose.yml
jessica@ubu20drc:/compose-mongo$ nano docker-compose.yml #COMPROBAMOS CONTENIDO#

```
jessica@ubu20drc:/compose-mongo$ docker-compose up #EJECUCIÓN#
Creating network "compose-mongo_default" with the default driver
Pulling mongo-express (mongo-express:)...
latest: Pulling from library/mongo-express
6a428f9f83b0: Pull complete
f2b1fb32259e: Pull complete
40888f2a0a1f: Pull complete
4e3cc9ce09be: Pull complete
eaa1898f3899: Pull complete
ab4078090382: Pull complete
ae780a42c79e: Pull complete
e60224d64a04: Pull complete
Digest: sha256:2a25aafdf23296823b06bc9a0a2af2656971262041b8dbf11b40444804fdc1
```

http://localhost:8081

DOCKERFILE

BORRAMOS IMÁGENES Y CONTENEDORES

root@ubu20drc:/home/jessica# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
CREAMOS DIRECTORIO CON PERMISOS
root@ubu20drc:/home/jessica# mkdir mongodb_dockerfile
root@ubu20drc:/home/jessica# chmod 777 mongodb_dockerfile
DESCARGAMOS EL DOCKERFILE QUE SE GUARDA DENTRO DEL DIRECTORIO
root@ubu20drc:/home/jessica/mongodb_dockerfile# ls
dockerfile
UBICADOS EN LA CARPETA DONDE ESTÁ EL DOCKEFILE, SE LANZA LA SIGUIENTE INSTANCIA

```
root@ubu20drc:/home/jessica/mongodb_dockerfile# docker build -t mongodbimagen .
Sending build context to Docker daemon  2.048kB
Step 1/6 : FROM mongo
 ---> 96c85f49715a
Step 2/6 : VOLUME ["/data/db"]
 ---> Running in 7e0cdefd1e84
Removing intermediate container 7e0cdefd1e84
 ---> a2b1981a298b
Step 3/6 : WORKDIR /data
 ---> Running in 0748b4e97f74
Removing intermediate container 0748b4e97f74
 ---> 835668a32623
Step 4/6 : CMD ["mongod"]
```

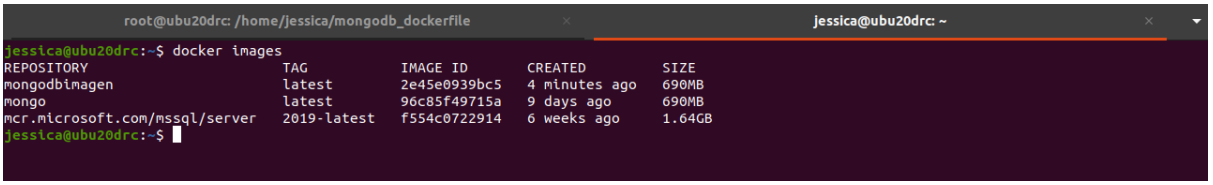
COMPROBACIÓN

```
root@ubu20drc:/home/jessica/mongodb_dockerfile# docker images
REPOSITORY                          TAG           IMAGE ID       CREATED          SIZE
mongodbimagen                       latest        2e45e0939bc5   45 seconds ago   690MB
mongo                               latest        96c85f49715a   9 days ago       690MB
mcr.microsoft.com/mssql/server      2019-latest   f554c0722914   6 weeks ago      1.64GB
```

EJECUCIÓN CONTENEDOR

```
root@ubu20drc:/home/jessica/mongodb_dockerfile# docker run --name dockerCVJ mongodbimagen
{"t":{"$date":"2022-06-02T08:43:15.154+00:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ct
x":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --ssl
DisabledProtocols 'none'"}
{"t":{"$date":"2022-06-02T08:43:15.155+00:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ct
x":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"
maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient"
:true}}}
```

TRABAJAMOS EN OTRA TERMINAL



EJECUCIÓN

```
jessica@ubu20drc:~$ docker exec -it dockerCVJ bash
root@bf690ab444ae:/data# mongo
MongoDB shell version v5.0.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gss
apiServiceName=mongodb
Implicit session: session { "id" : UUID("6554aa47-873c-477e-a01f-c
44e9386b066") }
MongoDB server version: 5.0.8
```

ENTRAMOS EN MONGO Y HACEMOS UNA PRUEBA SHOW DATABASES

```
---
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
> use mongodocker
switched to db mongodocker
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```


MONGOD

CADA SERVIDOR DE MONGODB EN UNA TERMINAL
Por ahora solamente hemos levantado tres instancias de mongodb por lo que todavía no hay sincronización entre las mismas. Para terminar de configuración nuestra réplica, entramos en un nodo
los nodos votan para elegir a un primario, que será quien reciba las escrituras. El resto de nodos se quedarán como secundarios. El proceso tarda en torno a 10 sg. Para ver el estado de las réplicas hacemos

TRAS LA INSTALACIÓN DE MONGOD CORREMOS LA SESIÓN CON MONGOSH

DIRECTORIO DE TRABAJO + PUERTO
mongod --replSet m101 --dbpath /data/rs1 --port 27017 1º INSTANCIA
mongod --replSet m101 --dbpath /data/rs2 --port 27018 2ºINSTANCIA
mongod --replSet m101 --dbpath /data/rs3 --port 27019 3ºINSTANCIA

```
jessica@ubu20drc:~$ mongo --port 27017
MongoDB shell version v5.0.8
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8f25b919-5c98-46a0-b61a-35eff03d78d9") }
MongoDB server version: 5.0.8
(...)

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command:
db.disableFreeMonitoring()
---------------------------------------------------------------------------------------------------------------------
> show dbs; Error
uncaught exception:: listDatabases failed:{
    "topologyVersion" : {
            "processId" : ObjectId("627d42b948864b114b323d2f"),
            "counter" : NumberLong(0)
    },
    "ok" : 0,
    "errmsg" : "not master and slaveOk=false",
    "code" : 13435,
    "codeName" : "NotPrimaryNoSecondaryOk"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:145:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:97:12
shellHelper.show@src/mongo/shell/utils.js:956:13
shellHelper@src/mongo/shell/utils.js:838:15
@(shellhelp2):1:1
---------------------------------------------------------------------------------------------------------------------
> config = { _id: "m101", members:[
...         { _id : 0, host : "localhost:27017"},
...         { _id : 1, host : "localhost:27018"},
...         { _id : 2, host : "localhost:27019"} ]
... };
{
    "_id" : "m101",
    "members" : [
            {
                    "_id" : 0,
                    "host" : "localhost:27017"
            },
            {
                    "_id" : 1,
                    "host" : "localhost:27018"
            },
            {
```

```
                "_id" : 2,
                "host" : "localhost:27019"
        }
    ]
}
> rs.initiate(config);
{
    "ok" : 1,
    "$clusterTime" : {
        "clusterTime" : Timestamp(1652376561, 1),
        "signature" : {
            "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
            "keyId" : NumberLong(0)
        }
    },
    "operationTime" : Timestamp(1652376561, 1)
}
m101:SECONDARY> rs.status(config);
{
    "set" : "m101",
    "date" : ISODate("2022-05-12T17:30:50.375Z"),
    "myState" : 1,
    "term" : NumberLong(1),
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "heartbeatIntervalMillis" : NumberLong(2000),
    "majorityVoteCount" : 2,
    "writeMajorityCount" : 2,
    "votingMembersCount" : 3,
    "writableVotingMembersCount" : 3,
    "optimes" : {
        "lastCommittedOpTime" : {
            "ts" : Timestamp(1652376643, 1),
            "t" : NumberLong(1)
        },
        "lastCommittedWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
        "readConcernMajorityOpTime" : {
            "ts" : Timestamp(1652376643, 1),
            "t" : NumberLong(1)
        },
        "appliedOpTime" : {
            "ts" : Timestamp(1652376643, 1),
            "t" : NumberLong(1)
        },
        "durableOpTime" : {
            "ts" : Timestamp(1652376643, 1),
            "t" : NumberLong(1)
        },
```

```
        "lastAppliedWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
        "lastDurableWallTime" : ISODate("2022-05-12T17:30:43.683Z")
},
"lastStableRecoveryTimestamp" : Timestamp(1652376613, 1),
"electionCandidateMetrics" : {
        "lastElectionReason" : "electionTimeout",
        "lastElectionDate" : ISODate("2022-05-12T17:29:33.368Z"),
        "electionTerm" : NumberLong(1),
        "lastCommittedOpTimeAtElection" : {
                "ts" : Timestamp(1652376561, 1),
                "t" : NumberLong(-1)
        },
        "lastSeenOpTimeAtElection" : {
                "ts" : Timestamp(1652376561, 1),
                "t" : NumberLong(-1)
        },
        "numVotesNeeded" : 2,
        "priorityAtElection" : 1,
        "electionTimeoutMillis" : NumberLong(10000),
        "numCatchUpOps" : NumberLong(0),
        "newTermStartDate" : ISODate("2022-05-12T17:29:33.466Z"),
        "wMajorityWriteAvailabilityDate" : ISODate("2022-05-12T17:29:34.805Z")
},
"members" : [
        {
                "_id" : 0,
                "name" : "localhost:27017",
                "health" : 1,
                "state" : 1,
                "stateStr" : "PRIMARY",
                "uptime" : 401,
                "optime" : {
                        "ts" : Timestamp(1652376643, 1),
                        "t" : NumberLong(1)
                },
                "optimeDate" : ISODate("2022-05-12T17:30:43Z"),
                "lastAppliedWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
                "lastDurableWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "Could not find member to sync from",
                "electionTime" : Timestamp(1652376573, 1),
                "electionDate" : ISODate("2022-05-12T17:29:33Z"),
                "configVersion" : 1,
                "configTerm" : 1,
                "self" : true,
                "lastHeartbeatMessage" : ""
        },
```

```
{
        "_id" : 1,
        "name" : "localhost:27018",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 88,
        "optime" : {
                "ts" : Timestamp(1652376643, 1),
                "t" : NumberLong(1)
        },
        "optimeDurable" : {
                "ts" : Timestamp(1652376643, 1),
                "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2022-05-12T17:30:43Z"),
        "optimeDurableDate" : ISODate("2022-05-12T17:30:43Z"),
        "lastAppliedWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
        "lastDurableWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
        "lastHeartbeat" : ISODate("2022-05-12T17:30:49.428Z"),
        "lastHeartbeatRecv" : ISODate("2022-05-12T17:30:48.934Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncSourceHost" : "localhost:27017",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 1,
        "configTerm" : 1
},
{
        "_id" : 2,
        "name" : "localhost:27019",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 88,
        "optime" : {
                "ts" : Timestamp(1652376643, 1),
                "t" : NumberLong(1)
        },
        "optimeDurable" : {
                "ts" : Timestamp(1652376643, 1),
                "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2022-05-12T17:30:43Z"),
        "optimeDurableDate" : ISODate("2022-05-12T17:30:43Z"),
        "lastAppliedWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
        "lastDurableWallTime" : ISODate("2022-05-12T17:30:43.683Z"),
```

```
                "lastHeartbeat" : ISODate("2022-05-12T17:30:49.428Z"),
                "lastHeartbeatRecv" : ISODate("2022-05-12T17:30:48.936Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "",
                "syncSourceHost" : "localhost:27017",
                "syncSourceId" : 0,
                "infoMessage" : "",
                "configVersion" : 1,
                "configTerm" : 1
        }
    ],
    "ok" : 1,
    "$clusterTime" : {
        "clusterTime" : Timestamp(1652376643, 1),
        "signature" : {
                "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                "keyId" : NumberLong(0)
        }
    },
    "operationTime" : Timestamp(1652376643, 1)
}
m101:PRIMARY> exit
bye

jessica@ubu20drc:~$ mongo --port 27017
MongoDB shell version v5.0.8
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fbacb4c2-7d72-4b28-920d-b43af43e2b6a") }
MongoDB server version: 5.0.8
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated
and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-05-12T19:24:09.041+02:00: Using the XFS filesystem is strongly
recommended with the WiredTiger storage engine. See
http://dochub.mongodb.org/core/prodnotes-filesystem
        2022-05-12T19:24:10.453+02:00: Access control is not enabled for the database.
Read and write access to data and configuration is unrestricted
        2022-05-12T19:24:10.453+02:00: You are running this process as the root user,
which is not recommended
```

2022-05-12T19:24:10.453+02:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2022-05-12T19:24:10.454+02:00: Soft rlimits for open file descriptors too low
2022-05-12T19:24:10.454+02:00: currentValue: 1024
2022-05-12T19:24:10.454+02:00: recommendedMinimum: 64000

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

m101:PRIMARY> use jcv;
switched to db jcv
m101:PRIMARY> db.usuario.insert({name: "jessica"});
WriteResult({ "nInserted" : 1 })
m101:PRIMARY> db.usuario.find()
{ "_id" : ObjectId("627d450d7888f318d6248658"), "name" : "jessica" }
m101:PRIMARY> exit;
jessica@ubu20drc:~$ mongo --port 27017
MongoDB shell version v5.0.8
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3d5eddda-f444-4fd4-bec2-4200f48575c3") }
MongoDB server version: 5.0.8
(...) To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
m101:PRIMARY>

---
m101:PRIMARY> show dbs;
admin 0.000GB
config 0.000GB

```
jcv      0.000GB
local   0.000GB
m101:PRIMARY> use jcv;
switched to db jcv
m101:PRIMARY> db.usuario.insert({name: "jessica"});
WriteResult({ "nInserted" : 1 })
m101:PRIMARY> db.usuario.find()
{ "_id" : ObjectId("627d450d7888f318d6248658"), "name" : "jessica" }
{ "_id" : ObjectId("627d46285187f6769b03e897"), "name" : "jessica" }
m101:PRIMARY> exit

m101:SECONDARY> show dbs;
uncaught exception: Error: listDatabases failed:{
    "topologyVersion" : {
            "processId" : ObjectId("627d42dad98927f2f01e1fb5"),
            "counter" : NumberLong(4)
    },
    "ok" : 0,
    "errmsg" : "not master and slaveOk=false",
    "code" : 13435,
    "codeName" : "NotPrimaryNoSecondaryOk",
    "$clusterTime" : {
            "clusterTime" : Timestamp(1652377213, 1),
            "signature" : {
                    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                    "keyId" : NumberLong(0)
            }
    },
    "operationTime" : Timestamp(1652377213, 1)
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:145:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:97:12
shellHelper.show@src/mongo/shell/utils.js:956:13
shellHelper@src/mongo/shell/utils.js:838:15
@(shellhelp2):1:1
```

NO ME REPLICA ….DA UN ERROR …COMPRUEBO EL SERVICIO:

```
● mongod.service - MongoDB Database Server
     Loaded: loaded (/lib/systemd/system/mongod.service; disabled>
     Active: failed (Result: exit-code) since Thu 2022-06-02 09:3>
       Docs: https://docs.mongodb.org/manual
    Process: 13211 ExecStart=/usr/bin/mongod --config /etc/mongod>
   Main PID: 13211 (code=exited, status=48)

jun 02 09:36:44 ubu20drc systemd[1]: Started MongoDB Database Ser>
jun 02 09:36:45 ubu20drc systemd[1]: mongod.service: Main process>
```

jun 02 09:36:45 ubu20drc systemd[1]: mongod.service: Failed with >

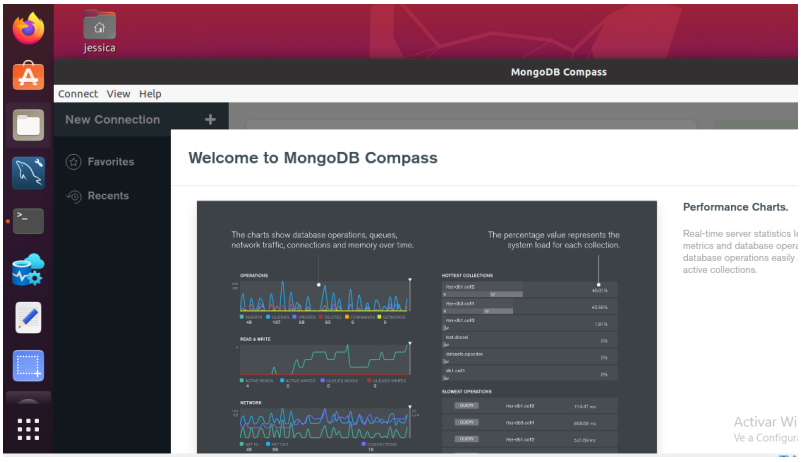
root@ubu20drc:/home/jessica# systemctl daemon-reload
root@ubu20drc:/home/jessica# systemctl status mongod
● mongod.service - MongoDB Database Server
Loaded: loaded (/lib/systemd/system/mongod.service; disabled>
Active: failed
ESTE FALLO APARECE TRÁS LAS PRUEBAS INICIALES CON LAS TRES PRIMERAS INSTANCIAS

PRUEBO A SOLUCIONARLO CON PERMISOS :

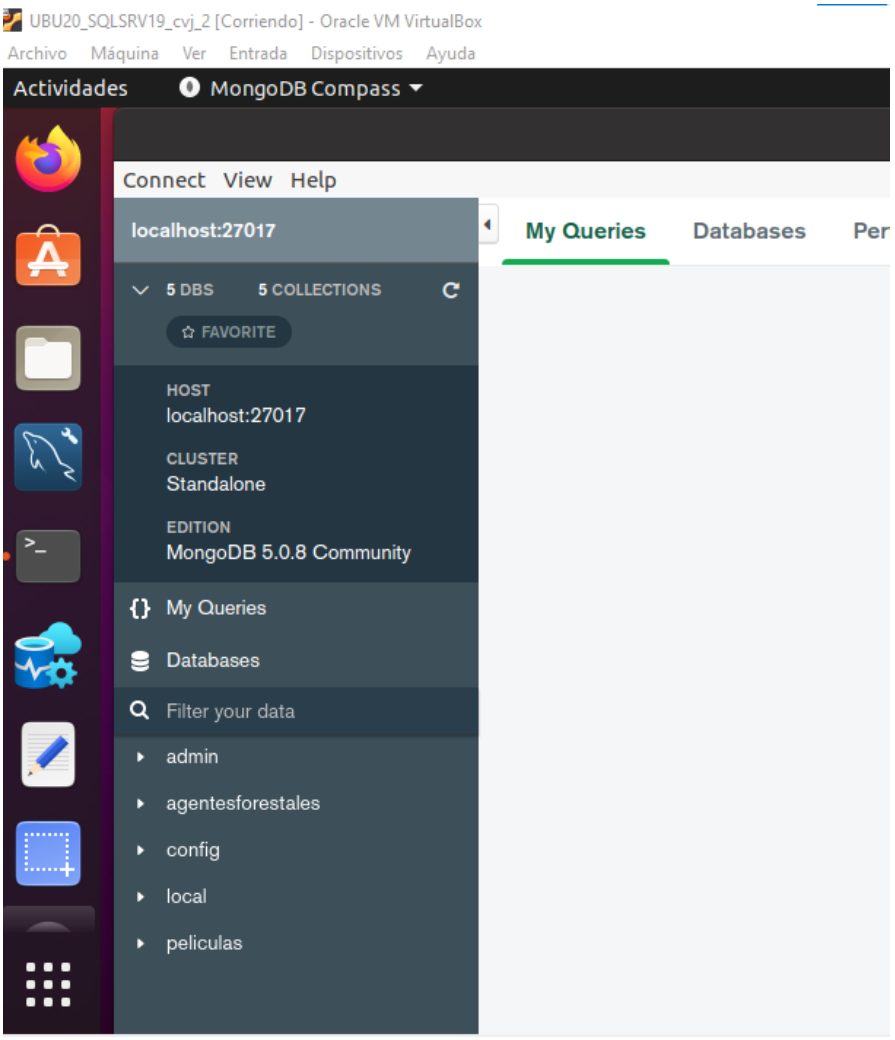
root@ubu20drc:/home/jessica# chown -R mongodb:mongodb /var/lib/mongodb
root@ubu20drc:/home/jessica# chown mongodb:mongodb /tmp/mongodb-27017.sock
—----------
root@ubu20drc:/home/jessica# sudo mkdir /var/lib/mongodb
mkdir: no se puede crear el directorio «/var/lib/mongodb»: El archivo ya existe
root@ubu20drc:/home/jessica# sudo mkdir /var/log/mongodb
mkdir: no se puede crear el directorio «/var/log/mongodb»: El archivo ya existe
root@ubu20drc:/home/jessica# sudo chown -R mongodb:mongodb /var/lib/mongodb
root@ubu20drc:/home/jessica# sudo chown -R mongodb:mongodb /var/log/mongodb
root@ubu20drc:/home/jessica# sudo service mongod start
root@ubu20drc:/home/jessica# sudo service mongod status
● mongod.service - MongoDB Database Server
Loaded: loaded (/lib/systemd/system/mongod.service; disabled>
Active: failed (Result: exit-code) since Thu 2022-06-02 09:3>

COMPASS

PREVIA INSTALACIÓN- (GUI)

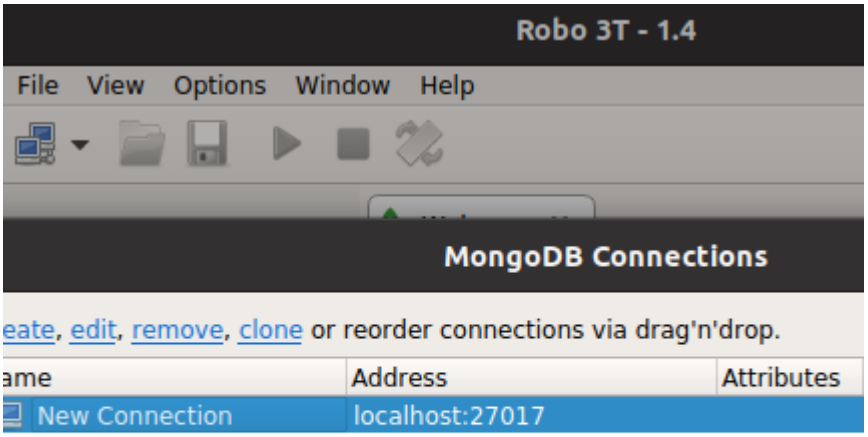


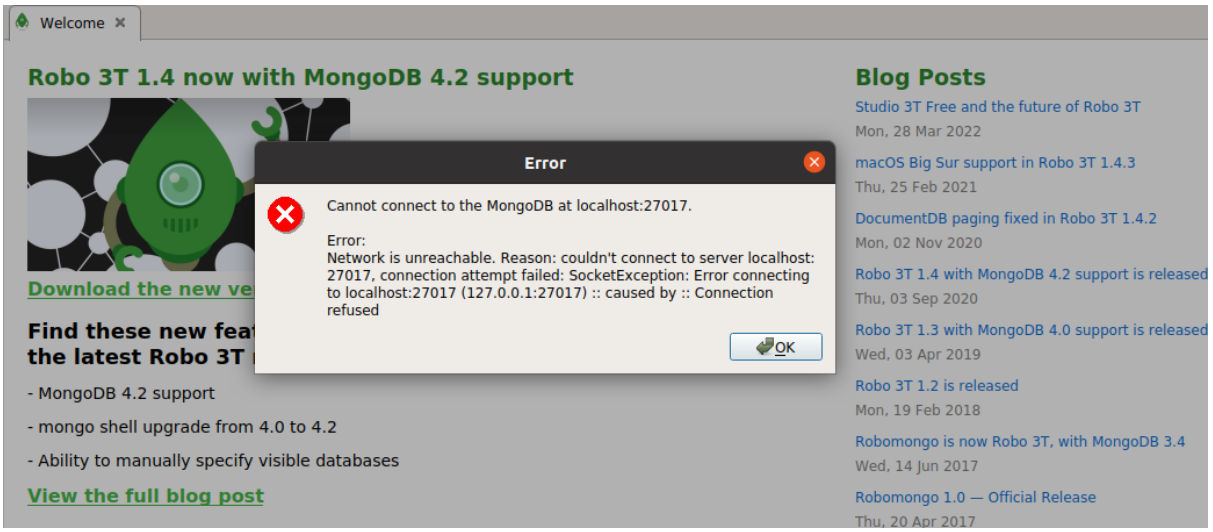
CONEXIÓN


UBU20_SQLSRV19_cvj_2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades
MongoDB Compass
Connect View Help
localhost:27017
5 DBS 5 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.8 Community
My Queries
Databases
Filter your data
admin
agentesforestales
config
local
peliculas
My Queries
Databases
Per

ROBO3T (ROBOMONGO)

ME DA ERROR DE CONEXIÓN





POSTGRESQL

GESTOR ORIENTADO A OBJETOS: https://hub.docker.com/_/postgres

jessica@ubu20drc:~$ docker stop $(docker ps -a -q)
bf690ab444ae
jessica@ubu20drc:~$ docker rm $(docker ps -a -q)
bf690ab444ae

INSTALACIÓN

```
jessica@ubu20drc:~$ docker run --name postgresql_cvj -e POSTGRES_U
SER=cvj -e POSTGRES_PASSWORD=123 -p 5432:5432 -v /data:/var/lib/po
stgresql/data -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
42c077c10790: Pull complete
3c2843bc3122: Pull complete
12e1d6a2dd60: Pull complete
9ae1101c4068: Pull complete
fb05d2fd4701: Pull complete
9785a964a677: Pull complete
16fc798b0e72: Pull complete
f1a0bfa2327a: Pull complete
fd2d68720749: Pull complete
83b23beac012: Pull complete
7962517582d4: Pull complete
6b4a569b8013: Pull complete
ad029fbc8984: Pull complete
Digest: sha256:2d1e636f07781d4799b3f2edbff78a0a5494f24c4512cb56a83
ebfd0e04ec074
Status: Downloaded newer image for postgres:latest
0f5661b91df5962d375b49cd16d99f7871f11e708979f6173f1ed6807590d7fa
```

jessica@ubu20drc:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
0f5661b91df5 postgres "docker-entrypoint.s…" About a minute ago Exited (1) About a minute ago postgresql_cvj

TRAS VARIOS INTENTOS NO CONSIGO EL UP DE ESTE CONTENEDOR UTILIZO OTRO COMANDO

```
jessica@ubu20drc:~$ docker run --name cvj-postgres -e
POSTGRES_PASSWORD=abc123. -d postgres
f914f3d272db36ad73cb96178458d5a44e4900d0404e8caf849c57302ebcecf2

jessica@ubu20drc:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS      NAMES
f914f3d272db   postgres   "docker-entrypoint.s…"   10 seconds ago   Up 7 seconds
5432/tcp   cvj-postgres
0f5661b91df5   postgres   "docker-entrypoint.s…"   7 minutes ago    Exited (1) 4 minutes ago
postgresql_cvj

ME CONECTO Y HAGO PRUEBAS
jessica@ubu20drc:~$ docker exec -it cvj-postgres bash
root@f914f3d272db:/# psql -U postgres
psql (14.3 (Debian 14.3-1.pgdg110+1))
Type "help" for help.
YA VEMOS LA TERMINAL CON LA QUE TRABAJAR
postgres=# create database agentes_forestales;
CREATE DATABASE
postgres=# create table agente (id int, name varchar (50) , ccaa varchar (100));
CREATE TABLE
postgres=# insert into agente values (1, 'juan', 'galicia');
INSERT 0 1
postgres=# insert into agente values (1, 'carlos', 'asturias');
INSERT 0 1
postgres=# select *from agente;
 id |  name  |   ccaa
----+--------+----------
  1 | juan   | galicia
  1 | carlos | asturias
(2 rows)
```

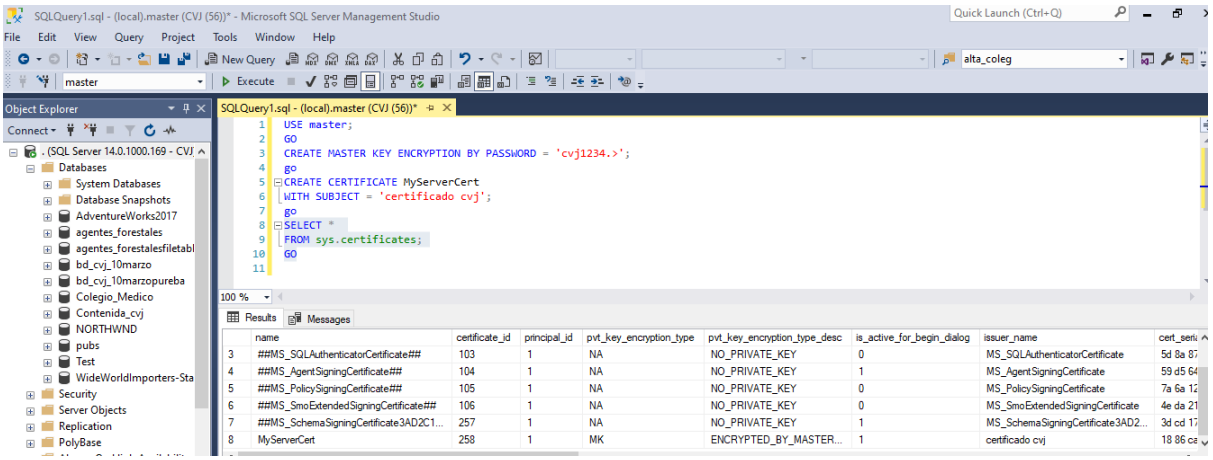
PGADMIN (GUI PARA POSTGRESQL):

```
jessica@ubu20drc:~$ docker run --name cvj-pgadmin -p 82:80 -e
'PGADMIN_DEFAULT_EMAIL=user@domain.org' -e
'PGADMIN_DEFAULT_PASSWORD=123'-d dpage/pgadmin4
Unable to find image 'dpage/pgadmin4:latest' locally
latest: Pulling from dpage/pgadmin4
df9b9388f04a: Pull complete
(...)1
d80b709014b3: Pull complete
Digest:
sha256:123ec096116d8e15875d7ed9d001eb1c755334acb248cfc4c48af154b23ab0da
Status: Downloaded newer image for dpage/pgadmin4:latest
NOTE: Configuring authentication for SERVER mode.
```

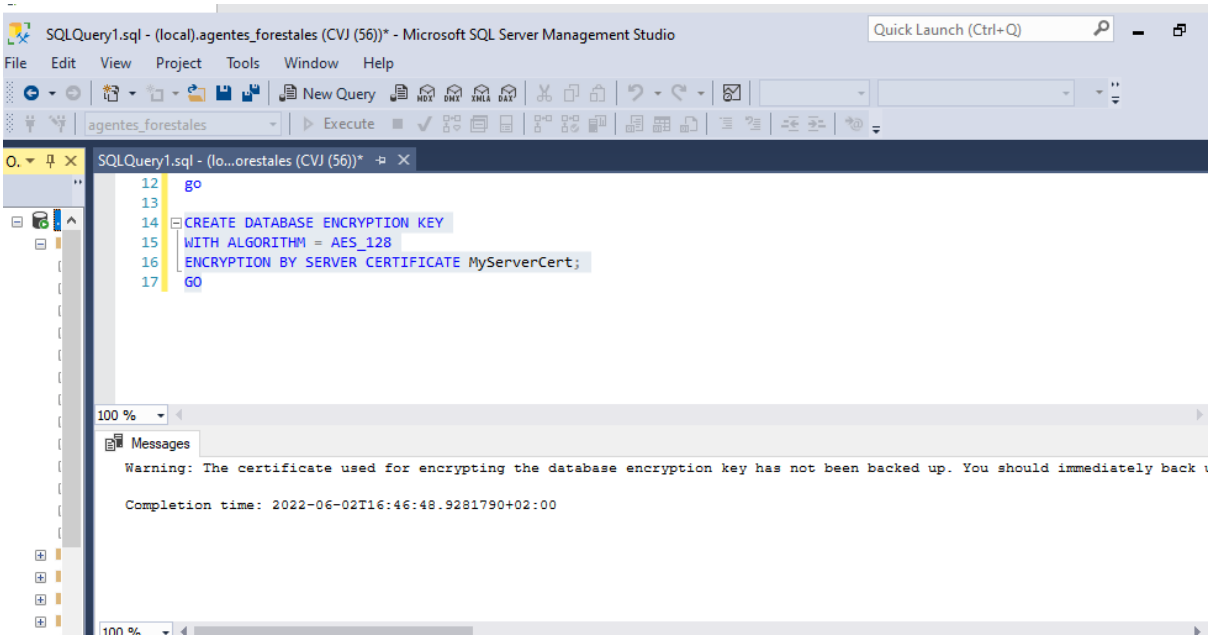
NO CONSIGO CONEXIÓN A TRAVÉS DEL NAVEGADOR

SEGURIDAD

CLAVE Y CERTIFICADO



ADVERTENCIA

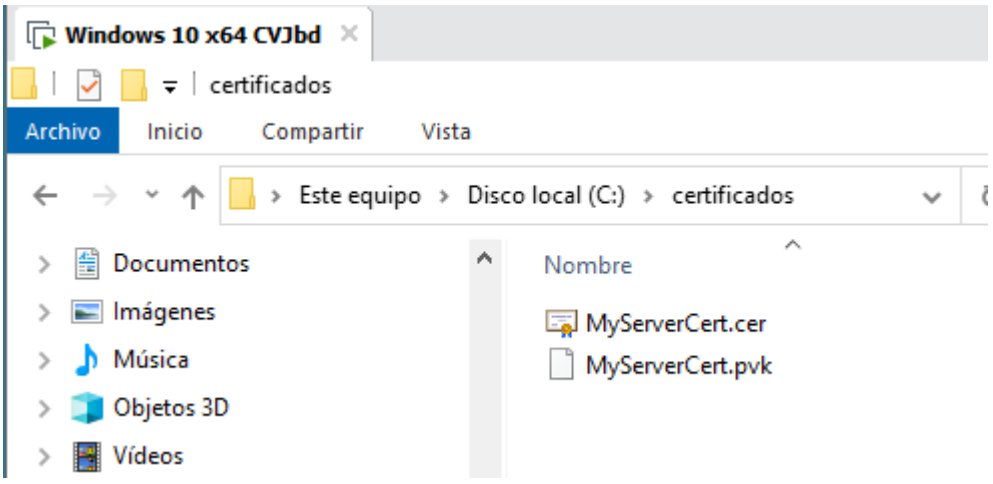


BACKUP CERTIFICADO

```
BACKUP CERTIFICATE MyServerCert
    TO FILE = 'C:\certificados\MyServerCert.cer'
    WITH PRIVATE KEY
        (
         FILE = 'C:\certificados\MyServerCert.pvk' ,
         ENCRYPTION BY PASSWORD = 'cvj1234.>'
        )
go
```

```
100 %
Messages
Commands completed successfully.
```

ARCHIVOS /IMPORTANTE REALIZAR UNA COPIA PARA GUARDAR EN OTRA UBICACIÓN FÍSICA.



COMPROBACIÓN BD

```
use master;
go

select * from sys.dm_database_encryption_keys
go
```

100 %

Results | Messages

	database_id	encryption_state	create_date	regenerate_date	modify_date	set_date	opened_date	key_algorithm	key_length	encryptor_thumbprint
1	12	1	2022-06-02 14:46:18.380	2022-06-02 14:46:18.380	2022-06-02 14:46:18.380	1900-01-01 00:00:00.000	2022-06-02 14:46:18.380	AES	128	0x8476A5651DAD6F

REALIZAR ENCRIPTADO