

No permite hacer pull pero la red esta bien

```
PS C:\Windows\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Windows\system32> docker pull mysql:latest
latest: Pulling from library/mysql
failed to copy: httpReadSeeker: failed open: failed to do request: Get "https://docker-images-prod.6aa30f8b08e16409b46e0173d6de2f56.r2
.cloudflarestorage.com/registry-v2/docker/registry/v2/blobs/sha256/f6/f6b0ca07d79d7d19c8da64558c3ccdd4ea635ac2193f551a1cb5370f33b494e8
/data?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=f1baa2dd9b876aeb89efebbf9c9e5d5f4%2F20251024%2Fauto%2Fs3%2Faws4_request&X-Amz-D
ate=20251024T192625Z&X-Amz-Expires=1200&X-Amz-SignedHeaders=host&X-Amz-Signature=e2c60a510644942d749afdfc176e08bf566f2b409e7cb0d9f9306
05164c1528e": dialing docker-images-prod.6aa30f8b08e16409b46e0173d6de2f56.r2.cloudflarestorage.com:443 container via direct connection
because static system has no HTTPS proxy: connecting to docker-images-prod.6aa30f8b08e16409b46e0173d6de2f56.r2.cloudflarestorage.com:
443: dial tcp: lookup docker-images-prod.6aa30f8b08e16409b46e0173d6de2f56.r2.cloudflarestorage.com: no such host
PS C:\Windows\system32> ping google.com

Haciendo ping a google.com [172.217.29.14] con 32 bytes de datos:
Respuesta desde 172.217.29.14: bytes=32 tiempo=10ms TTL=115
Respuesta desde 172.217.29.14: bytes=32 tiempo=10ms TTL=115
Respuesta desde 172.217.29.14: bytes=32 tiempo=10ms TTL=115
Respuesta desde 172.217.29.14: bytes=32 tiempo=10ms TTL=115

Estadísticas de ping para 172.217.29.14:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 10ms, Máximo = 10ms, Media = 10ms
PS C:\Windows\system32>
```

Configurar dns manualmente

Docker Engine

v28.4.0

Configure the Docker daemon by typing a json Docker daemon [configuration file](#).

This can prevent Docker from starting. Use at your own risk.

```
{
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "dns": [
    "8.8.8.8",
    "8.8.4.4",
    "1.1.1.1"
  ]
}
```

Pull exitoso

```
PS C:\Windows\system32> docker pull mysql:latest
latest: Pulling from library/mysql
21aa606d8d58: Pull complete
f5f78fcd9ccb: Pull complete
5a3f7744d0e7: Pull complete
834e15e3ed24: Pull complete
dcee80f7340c: Pull complete
c276de9b5571: Pull complete
0cd145fbb449: Pull complete
480d01bd7a6a: Pull complete
023a182c62a0: Pull complete
494c372d15c3: Pull complete
Digest: sha256:569c4128dfa625ac2ac62cdd8af588a3a6a60a049d1a8d8f0fac95880ecdbee5
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
PS C:\Windows\system32>
```

Instanciar la imagen mysql:latest

```
PS C:\Windows\system32> docker run --name mysql_feria -e MYSQL_ROOT_PASSWORD=tarea -e MYSQL_DATABASE=feria -p 3306:3306 -v mysql_data:/var/lib/mysql -d mysql:latest
8cacae54f6c0368d3ec921a01a4e0ad03c4374a02e27cca8ea8866bfff0dca985
```

Entrar a mysql

```
PS C:\Windows\system32> docker exec -it mysql_feria mysql -u root -p feria
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.5.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql>
```

creamos todas nuestras tablas

```
mysql> create table feria(  
-> id_feria int primary key auto_increment,  
-> nombre varchar(60),  
-> descripcion text  
-> );  
Query OK, 0 rows affected (0.065 sec)  
  
mysql>  
mysql> create table tematica(  
-> id_tematica int primary key auto_increment,  
-> tematica varchar(20)  
-> );  
Query OK, 0 rows affected (0.020 sec)  
  
mysql>  
mysql> create table pabellon(  
-> id_pabellon int primary key auto_increment,  
-> nombre varchar (60),  
-> id_tematica int,  
-> id_feria int,  
-> foreign key (id_tematica) references tematica(id_tematica),  
-> foreign key (id_feria) references feria(id_feria)  
-> );  
Query OK, 0 rows affected (0.080 sec)  
  
mysql>  
mysql> create table empresa(  
-> id_empresa int primary key auto_increment,  
-> nombre_empresa varchar (80),  
-> descripcion text  
-> );  
Query OK, 0 rows affected (0.022 sec)  
  
mysql>  
mysql> create table stand (  
-> id_stand int primary key auto_increment,  
-> nombre_stand varchar (80),  
-> id_pabellon int,  
-> id_empresa int,  
-> foreign key (id_pabellon) references pabellon(id_pabellon),  
-> foreign key (id_empresa) references empresa (id_empresa)  
-> );  
Query OK, 0 rows affected (0.073 sec)  
  
mysql>  
mysql> create table producto (  
-> id_producto int primary key auto_increment,  
-> nombre_producto varchar(60),  
-> id_stand int,  
-> foreign key (id_stand) references stand (id_stand)  
-> );  
Query OK, 0 rows affected (0.030 sec)
```

```
mysql> show tables
->
-> ;
+-----+
| Tables_in_feria |
+-----+
| charla          |
| demostracion    |
| empresa         |
| feria           |
| pabellon        |
| persona         |
| ponente         |
| producto        |
| registro        |
| responsable     |
| stand           |
| tematica        |
| tipo_visitante  |
| visitante       |
+-----+
14 rows in set (0.002 sec)

mysql>
```

verificar que todas las tablas se hayan creado correctamente

creamos nuestro sp para los 1000 inserts por tabla

```
mysql> delimiter $$
, 'nuevas tecnmysql>
mysql> create procedure insertar_datos_feria()
', 'transformac -> begin
-> declare i int default 1;
ncat('c -> declare j int;
->
->
-> set foreign_key_checks = 0;
-> del sector. sesión truncate table registro;
-> truncate table responsable;

-> truncate table producto;
-> truncate table stand;
-> truncate table pabellon;
-> truncate table visitante;
values
-> truncate table ponente;
': -> truncate table persona;
du -> truncate table demostracion;
-> truncate table charla;
-> vadora', truncate table empresa;
af -> truncate table tipo_visitante;
um')) -> truncate table tematica;
ci -> truncate table feria;
-> set foreign_key_checks = 1;
esentación interactiva d ->
el pr ->
-> while i <= 1000 do
-> insert into feria (nombre, descripcion) values
```

Seguido de esto lo ejecutamos

```
Query OK, 0 rows affected (0.018 sec)

mysql>
mysql> delimiter ;
mysql> call insertar_datos_feria();
+-----+
| resultado |
+-----+
| datos insertados exitosamente: 1000 registros en cada tabla |
+-----+
1 row in set (1 min 19.930 sec)

Query OK, 0 rows affected (1 min 19.930 sec)
```

Luego de esto lo primero que tenemos que hacer es crear los schemas y asignarle las tablas

```
-> create schema estructura;
Query OK, 1 row affected (0.016 sec)

mysql> create schema sector_empresarial;
Query OK, 1 row affected (0.047 sec)

mysql> create schema personas_actividades;
Query OK, 1 row affected (0.007 sec)

mysql>
mysql> rename table feria.feria to estructura.feria;
Query OK, 0 rows affected (0.024 sec)

mysql> rename table feria.tematica to estructura.tematica;
Query OK, 0 rows affected (0.019 sec)

mysql> rename table feria.pabellon to estructura.pabellon;
Query OK, 0 rows affected (0.060 sec)

mysql>
mysql> rename table feria.empresa to sector_empresarial.empresa;
Query OK, 0 rows affected (0.023 sec)

mysql> rename table feria.stand to sector_empresarial.stand;
Query OK, 0 rows affected (0.024 sec)

mysql> rename table feria.producto to sector_empresarial.producto;
Query OK, 0 rows affected (0.019 sec)

mysql> rename table feria.responsable to sector_empresarial.responsable;
Query OK, 0 rows affected (0.062 sec)

mysql>
mysql> rename table feria.persona to personas_actividades.persona;
Query OK, 0 rows affected (0.022 sec)

mysql> rename table feria.ponente to personas_actividades.ponente;
Query OK, 0 rows affected (0.024 sec)

mysql> rename table feria.tipo_visitante to personas_actividades.tipo_visitante;
Query OK, 0 rows affected (1.301 sec)

mysql> rename table feria.visitante to personas_actividades.visitante;
Query OK, 0 rows affected (0.029 sec)

mysql> rename table feria.charla to personas_actividades.charla;
Query OK, 0 rows affected (0.020 sec)

mysql> rename table feria.demonstracion to personas_actividades.demonstracion;
Query OK, 0 rows affected (1.770 sec)

mysql> rename table feria.registro to personas_actividades.registro;
Query OK, 0 rows affected (0.031 sec)

mysql>
```

Creamos los usuarios y asignamos privilegios

```
mysql> grant all privileges on estructura.* to 'user_adminferia'@'%';
Query OK, 0 rows affected (0.004 sec)

mysql> grant all privileges on sector_empresarial.* to 'user_adminferia'@'%';
Query OK, 0 rows affected (0.004 sec)

mysql> grant all privileges on personas_actividades.* to 'user_adminferia'@'%';
Query OK, 0 rows affected (0.004 sec)

mysql>
mysql> grant select, insert, update, delete on estructura.* to 'user_estructura'@'%';
Query OK, 0 rows affected (0.005 sec)

mysql>
mysql> grant select, insert, update, delete on sector_empresarial.* to 'user_empresarial'@'%';
Query OK, 0 rows affected (0.041 sec)

mysql>
mysql> grant select, insert, update, delete on personas_actividades.* to 'user_personaa'@'%';
Query OK, 0 rows affected (0.007 sec)

mysql>
```

Ejecutamos un contenedor de Ubuntu, así no tengamos la imagen la descargara y le hará la instancia, velamos para que el id del grupo de Linux sea 0. Actualizamos descargamos docker

```
PS C:\Windows\system32> docker run -it --name ubuntu_docker -v /var/run/docker.sock:/var/run/docker.sock --group-add 0 ubuntu
bash
root@886fb8085d3c:/# apt-get update && apt-get install -y docker.io
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1587 kB]
```

Duplicamos el volumen de nuestra anterior base de datos feria

```
root@886fb8085d3c:/# docker run --rm -v mysql_data:/source:ro -v mysql_data_ubuntu:/dest ubuntu bash -c "cp -av /source/. /dest/"
```

Luego ejecutamos el contenedor de mysql dentro de Ubuntu con nuestro volumen duplicado

```
root@886fb8085d3c:/# docker run --name mysql_feria_ubuntu -e MYSQL_ROOT_PASSWORD=tarea -p 3307:3306 -v mysql_data_ubuntu:/var/lib/mysql -d mysql:latest
2275be6f1b0ce5476411576fe463d3fb5caa9ffdc7d6d932828fc9b12970f254
```

Comprobamos que el volumen se haya duplicado y asignado correctamente y listo

```
root@886fb8085d3c:/# docker exec -it mysql_feria_ubuntu mysql -uroot -ptarea
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 9.5.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> select count(*) from estructura.feria;
+-----+
| count(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.034 sec)

mysql>
```

iniciamos el contenedor de mysql dentro de linux

```
root@886fb8085d3c:/# docker exec -it mysql_feria_ubuntu mysql -uroot -ptarea
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.5.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> select count(*) from esquema.feria;
ERROR 1049 (42000): Unknown database 'esquema'
mysql> exit
Bye
root@886fb8085d3c:/# docker exec -it mysql_feria_ubuntu mysql -uroot -ptarea
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 9.5.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> select count(*) from estructura.feria;
+-----+
| count(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.034 sec)

mysql> █
```

```
root@886fb8085d3c:/# apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@886fb8085d3c:/# apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

Salimos de linux y volvemos a entrar con docker attach

```
PS C:\Windows\system32> docker attach ubuntu_docker
```

Ejecutamos el demonio de mysql

```
root@886fb8085d3c:/# /usr/sbin/mysqld --daemonize &
[1] 13
root@886fb8085d3c:/# mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 15
```

Escribimos mysql

```
mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql>
```

Salimos de mysql para ejecutar la instalación segura a lo que todo le vamos a dar y

```
mysql> \q
Bye
[1]+  Done                  /usr/sbin/mysqld --daemonize
root@886fb8085d3c:/# mysql_secure_installation
```

Ejecutamos mysql de nuevo como somos el usuario root de Linux mysql asume que tenemos permiso por lo que en contraseña podemos digitar cualquier cosa

```
root@886fb8085d3c:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql>
```


Añadimos todos los contenedores a un network para que el proxysql pueda manejarlos

En PowerShell:

1. Crear una red de puente (bridge)

`docker network create db_network`

2. Conectar tu contenedor ubuntu_docker (donde está el cliente) a la nueva red

`docker network connect db_network ubuntu_docker`

3. Conectar tu contenedor mysql_feria (el servidor original) a la nueva red

`docker network connect db_network mysql_feria`

4. Conectar el otro servidor MySQL (mysql_feria_ubuntu) a la nueva red

`docker network connect db_network mysql_feria_ubuntu`

Instalamos el contenedor de el manejador de conexión proxy (obviamente añadiéndolo al bridge)

```
root@886fb8085d3c:/# docker run -d --name proxysql --network db_network -p 6033:6033 -p 6032:6032 -e MYSQL_HOSTGROUP="10" -e MYSQL_HOST="mysql_feria:3306,mysql_feria_ubuntu:3306" -e MYSQL_USER="root" -e MYSQL_PASSWORD="tarea" proxysql/proxysql
```

configuramos el proxy desde el contenedor de proxysql como admin para apoderarnos de los dos contenedores

```
root@886fb8085d3c:/# docker exec -it proxysql bash
root@19f8e9ad72fe:/# mysql -u admin -p -h 127.0.0.1 --port 6032
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.5.30 (ProxySQL Admin Module)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> select hostgroup_id, hostname, status, weight from runtime_mysql_servers;
Empty set (0.002 sec)

MySQL [(none)]> insert into mysql_servers (hostgroup_id, hostname, port) values (10, 'mysql_feria', 3306);
Query OK, 1 row affected (0.001 sec)

MySQL [(none)]> insert into mysql_servers (hostgroup_id, hostname, port) values (10, 'mysql_feria_ubuntu', 3306);
Query OK, 1 row affected (0.001 sec)

MySQL [(none)]> load mysql servers to runtime;
Query OK, 0 rows affected (0.005 sec)

MySQL [(none)]> save mysql servers to disk;
Query OK, 0 rows affected (0.073 sec)

MySQL [(none)]> select hostgroup_id, hostname, status, weight from runtime_mysql_servers;
+-----+-----+-----+-----+
| hostgroup_id | hostname           | status | weight |
+-----+-----+-----+-----+
| 10           | mysql_feria        | ONLINE | 1      |
| 10           | mysql_feria_ubuntu | ONLINE | 1      |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)

MySQL [(none)]>
```

salimos hasta el contenedor de Linux iniciamos una nueva conexión del usuario root utilizando proxy verificamos que todo sirva y listo

```
MySQL [(none)]> \q
Bye
root@19f8e9ad72fe:/# exit
exit
root@886fb8085d3c:/# mysql -u root -p -h proxysql --port 6033
Enter password:
ERROR 1045 (28000): ProxySQL Error: Access denied for user 'root'@'172.18.0.4' (using password: YES)
root@886fb8085d3c:/# mysql -u root -p -h proxysql --port 6033
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.5.30 (ProxySQL)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

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

Database changed
mysql> show tables;
+-----+
| Tables_in_estructura |
+-----+
| feria                 |
| pabellon              |
| tematica              |
+-----+
3 rows in set (0.00 sec)

mysql>
```

CONTINUAMOS CON SQLSERVER

Instalamos el contenedor de sqlserver con volumen y de una ves lo añadimos a el network

```
PS C:\Windows\system32> docker run -d --name sqlserver_feria --hostname sqlserver_feria --network db_network -e 'ACCEPT_EULA=Y' -e 'SA
_PASSWORD=Tarea_20258.' -e 'MSSQL_PID=Developer' -p 1433:1433 -v sqlserver_feria_data:/var/opt/mssql mcr.microsoft.com/mssql/server:20
19-latest
dce47aee7218977427c549dfa79438f54bdef686980df4b1d7c2d5839cfbd328
PS C:\Windows\system32>
```

Ingresamos al contenedor

```
PS C:\Windows\system32> docker exec -it sqlserver_feria bash
mssql@sqlserver_feria:/$
```

salimos de ahí y empezamos a instalar dependencias de sqlserver

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria bash -c "apt update && apt install -y curl apt-transport-https debconf-utils"

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 https://packages.microsoft.com/ubuntu/20.04/prod focal InRelease [3632 B]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:4 https://packages.microsoft.com/ubuntu/20.04/prod focal/main armhf Packages [28.5 kB]
Get:5 https://packages.microsoft.com/ubuntu/20.04/prod focal/main all Packages [2938 B]
Get:6 https://packages.microsoft.com/ubuntu/20.04/prod focal/main amd64 Packages [393 kB]
Get:7 https://packages.microsoft.com/ubuntu/20.04/prod focal/main arm64 Packages [106 kB]
Get:8 https://archive.ubuntu.com/ubuntu focal-updates InRelease [5120 B]
Get:9 https://archive.ubuntu.com/ubuntu focal-backports InRelease [5120 B]
Get:10 https://packages.microsoft.com/ubuntu/20.04/prod focal/main amd64 Packages [393 kB]
Get:11 https://packages.microsoft.com/ubuntu/20.04/prod focal/main arm64 Packages [106 kB]
Get:12 https://packages.microsoft.com/ubuntu/20.04/prod focal/main armhf Packages [28.5 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [49.8 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main arm64 Packages [49.8 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main armhf Packages [28.5 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/main all Packages [59.3 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [30.1 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-updates/restricted arm64 Packages [30.1 kB]
Get:19 http://archive.ubuntu.com/ubuntu focal-updates/restricted armhf Packages [28.5 kB]
Get:20 http://archive.ubuntu.com/ubuntu focal-updates/restricted all Packages [29.3 kB]
Get:21 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [72.6 kB]
Get:22 http://archive.ubuntu.com/ubuntu focal-backports/main arm64 Packages [72.6 kB]
Get:23 http://archive.ubuntu.com/ubuntu focal-backports/main armhf Packages [28.5 kB]
Get:24 http://archive.ubuntu.com/ubuntu focal-backports/main all Packages [29.3 kB]
Get:25 http://archive.ubuntu.com/ubuntu focal-backports/restricted amd64 Packages [30.1 kB]
Get:26 http://archive.ubuntu.com/ubuntu focal-backports/restricted arm64 Packages [30.1 kB]
Get:27 http://archive.ubuntu.com/ubuntu focal-backports/restricted armhf Packages [28.5 kB]
Get:28 http://archive.ubuntu.com/ubuntu focal-backports/restricted all Packages [29.3 kB]
Fetched 10.1 MB in 1s (10.1 MB/s)
Reading package lists...
Building dependency tree...
Reading state information...
curl is already the newest version (7.81.0-1ubuntu1.1).
debconf-utils is already the newest version (1.290ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria apt install -y gnupg

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
  gpgconf gpgsm libassuan0 libksba8 libnpth0 pinentry-curses
Suggested packages:
  dbus-user-session libpam-systemd pinentry-gnome3 tor parcimonie xloadimage
  scd daemon pinentry-doc
```

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria bash -c "curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--
100  975  100  975    0     0  6456    0  --:--:-- --:--:-- --:--:--  6456
Warning: apt-key output should not be parsed (stdout is not a terminal)
OK
PS C:\Windows\system32>
```

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria bash -c "curl https://packages.microsoft.com/config/ubuntu/20.04/prod.list | tee /etc/apt/sources.list.d/mssql-release.list"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total   Spent    Left   Speed
100   89   100   89    0     0   502    0  --:--:-- --:--:-- --:--:--   500
deb [arch=amd64,armhf,arm64] https://packages.microsoft.com/ubuntu/20.04/prod focal main
PS C:\Windows\system32>
```

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria apt update

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Hit:1 https://packages.microsoft.com/ubuntu/20.04/prod focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists...
Building dependency tree...
Reading state information...
1 package can be upgraded. Run 'apt list --upgradable' to see it.
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/mssql-release.list:1
```

```
PS C:\Windows\system32> docker exec -u 0 sqlserver_feria bash -c "ACCEPT_EULA=Y apt install -y mssql-tools unixodbc-dev"

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  autoconf automake autotools-dev binutils binutils-common
  binutils-x86-64-linux-gnu cpp cpp-9 gcc gcc-9 gcc-9-base libasan5
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libctf-nobfd0 libctf0
```

Después de instalar todas las dependencias y herramientas ejecutamos sqlserver de forma interactiva y ejecutamos sqlcmd de forma correcta (con la ruta)

```
PS C:\Windows\system32> docker exec -it sqlserver_feria bash
mssql@sqlserver_feria:/$ sqlcmd -S localhost -U SA -P 'Tarea_20258.'
bash: sqlcmd: command not found
mssql@sqlserver_feria:/$ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P 'Tarea_20258.'
1>
```

Lo siguiente es ingresar todo el código de nuestra base de datos

```
mssql@sqlserver_feria:/$ /opt/mssql-tools/bin/sqlcmd -S localhost -U SA -P 'Tarea_20258.'
1> create database feria;
2> go
```

Procedure con los 1000 inserts por tabla

[illegible]

Creamos los schemas, los logins y usuarios correctamente

```

1> create schema estructura authorization dbo;
2> go
1> create schema sector_empresarial authorization dbo;
2> go
1> create schema personas_actividades authorization dbo;
2> go
1> alter schema estructura transfer dbo.feria; alter schema estructura transfer dbo.tematica; alter schema estructura transfer dbo.pabellon;
2> go
1> alter transfer dbo.stand; alter schema sector_empresarial transfer dbo.producto; alter schema sector_empresarial transfer dbo.responsable;
2> go
1> alter transfer dbo.charla; alter schema personas_actividades transfer dbo.demonstracion; alter schema personas_actividades transfer dbo.registro;
2> go
1> or login login_estructura; create user user_empresarial for login login_empresarial; create user user_personaa for login login_personaa;
2> go
1> o admin_feria; grant control on schema::sector_empresarial to admin_feria; grant control on schema::personas_actividades to admin_feria;
2> go
1> grant select, delete, update, insert on schema::estructura to user_estructura;
2> go
1> grant select, delete, update, insert on schema::sector_empresarial to user_empresarial;
2> go
1> grant select, delete, update, insert on schema::personas_actividades to user_personaa;
2> go
1>

```

Ahora necesitamos duplicar el volumen de nuestro contenedor del host para tener todo lo que hemos hecho en el contenedor del motor que esta dentro de Ubuntu

Primeramente apagamos el contenedor al que le deseamos duplicar el volumen

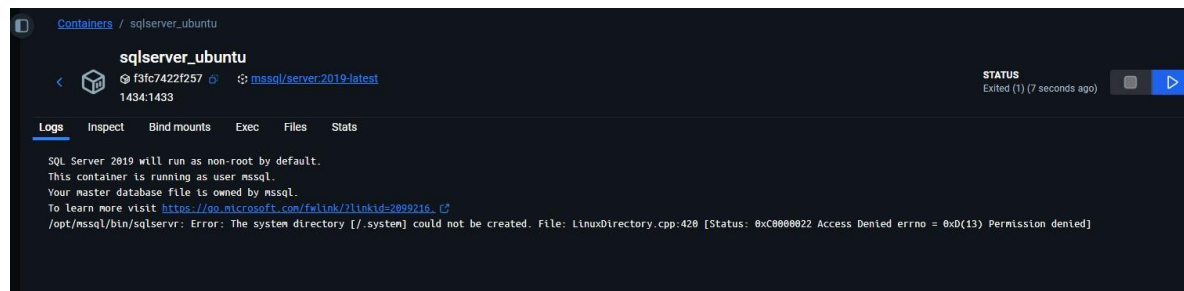
Luego duplicamos el volumen

```
PS C:\Windows\system32> docker stop sqlserver_feria
sqlserver_feria
PS C:\Windows\system32> docker run --rm -v sqlserver_feria_data:/data_origen -v sqlserver_feria_ubuntu_data:/data_destino alpine ash -c "cp /data_origen/* /data_destino"
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
3a85c1b5348c: Pulling fs object
3a85c1b5348c: Pull complete
Digest: sha256:3a85c1b5348c
```

Por ultimo encendemos Ubuntu y atacamos para así crear nuestro contenedor en Ubuntu

```
PS C:\Windows\system32> docker start ubuntu_docker
ubuntu_docker
PS C:\Windows\system32> docker attach ubuntu_docker
root@886fb8085d3c:/# docker run -d --name sqlserver_ubuntu --hostname sqlserver_ubuntu --network db_network -e 'accept_eula=Y' -e 'sa_password=tarea_20258.' -e 'mssql_pid=developer' -p 1434:1433 -v sqlserver_feria_ubuntu_data:/var/opt/mssql mcr.microsoft.com/mssql/server:2019-latest
a03c8b323a533a2eebd3eed3184ffe9c9ff470f678c2cf3edb5394c748a8587fc
root@886fb8085d3c:/#
```

Se nos va a presentar el siguiente error



Con este comando le damos permisos al volumen para forzar que la carpeta de datos del volumen (/var/opt/mssql) sea propiedad del usuario 10001 (el UID de mssql). Ahora nuestro contenedor debería permanecer activo

```
root@886fb8085d3c:/# docker run --rm -v sqlserver_feria_ubuntu_data:/mnt alpine chown -R 10001:0 /mnt
root@886fb8085d3c:/# docker start sqlserver_ubuntu
sqlserver_ubuntu
root@886fb8085d3c:/#
```

Instalar gnupg y wget

```
root@886fb8085d3c:/# apt update && apt install -y gnupg wget
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
```

Ahora vamos a ejecutar el contenedor de postgresql de windows

```
PS C:\Windows\system32> docker run -d --name postgresql_feria -e POSTGRES_USER=admin -e POSTGRES_PASSWORD=Tarea_20258. -e POSTGRES_DB=feria
p 5432:5432 -v pgdata_feria:/var/lib/postgresql/data postgres:16
883d40f90f046707b88cd0b2f4a1d47401386b7faf6374025146d51f030804d2
```

Ingresamos al contenedor

```
PS C:\Windows\system32> docker exec -it postgresql_feria psql -U admin -d feria
psql (16.10 (Debian 16.10-1.pgdg13+1))
Type "help" for help.

feria=#
```

Creamos nuestras tablas

```
PS C:\Windows\system32> docker exec -it postgresql_feria psql -U admin -d feria
psql (16.10 (Debian 16.10-1.pgdg13+1))
Type "help" for help.

feria=# create table feria(
feria(#      id_feria serial primary key,
feria(#      nombre varchar(60),
feria(#      descripcion text
feria(# );
CREATE TABLE
feria=#
feria=# create table tematica(
feria(#      id_tematica serial primary key,
```

Creamos la función

```
feria=# CREATE OR REPLACE FUNCTION insertar_datos_feria()
feria-# RETURNS void AS $$
feria$# DECLARE
feria$#      i INTEGER := 1;
feria$# BEGIN
feria$#      PERFORM pg_catalog.set_config('session_replication_role', 'replica', true);
feria$#
feria$#      TRUNCATE TABLE registro, responsable, producto, stand, pabellon,
feria$#                  visitante, ponente, persona, demostracion, charla,
feria$#                  empresa, tipo_visitante, tematica, feria RESTART IDENTITY CASCADE;
feria$#
feria$#      PERFORM pg_catalog.set_config('session_replication_role', 'origin', true);
feria$#
feria$#      FOR i IN 1..1000 LOOP
feria$#          INSERT INTO feria (nombre, descripcion)
```

Y la llamamos y verificamos

```
feria=# SELECT insertar_datos_feria();
insertar_datos_feria
-----
(1 row)

feria=# SELECT 'feria' AS tabla, COUNT(*) FROM feria
feria=# UNION ALL
feria=# SELECT 'tematica', COUNT(*) FROM tematica
feria=# UNION ALL
feria=# SELECT 'empresa', COUNT(*) FROM empresa
feria=# UNION ALL
feria=# SELECT 'pabellon', COUNT(*) FROM pabellon
feria=# UNION ALL
feria=# SELECT 'stand', COUNT(*) FROM stand
feria=# UNION ALL
feria=# SELECT 'producto', COUNT(*) FROM producto
feria=# UNION ALL
feria=# SELECT 'persona', COUNT(*) FROM persona
feria=# UNION ALL
feria=# SELECT 'responsable', COUNT(*) FROM responsable
feria=# UNION ALL
feria=# SELECT 'ponente', COUNT(*) FROM ponente
feria=# UNION ALL
feria=# SELECT 'tipo_visitante', COUNT(*) FROM tipo_visitante
feria=# UNION ALL
feria=# SELECT 'visitante', COUNT(*) FROM visitante
feria=# UNION ALL
feria=# SELECT 'charla', COUNT(*) FROM charla
feria=# UNION ALL
feria=# SELECT 'demostracion', COUNT(*) FROM demostracion
feria=# UNION ALL
feria=# SELECT 'registro', COUNT(*) FROM registro;
      tabla      | count
-----+-----
 feria           | 1000
 tematica        | 1000
 empresa         | 1000
 pabellon        | 1000
 stand           | 1000
 producto        | 1000
 persona         | 1000
 responsable     | 1000
 ponente         | 1000
 tipo_visitante  | 1000
 visitante       | 1000
 charla          | 1000
 demostracion    | 1000
 registro        | 1000
(14 rows)

feria=#
```


Creamos schemas pasamos tablas

```
feria=# create schema estructura;  
CREATE SCHEMA  
feria=# create schema sector_empresarial;  
CREATE SCHEMA  
feria=# create schema personas_actividades;  
CREATE SCHEMA  
feria=#  
feria=#  
feria=# alter table public.feria set schema estructura;  
ALTER TABLE  
feria=# alter table public.tematica set schema estructura;  
ALTER TABLE  
feria=# alter table public.pabellon set schema estructura;  
ALTER TABLE  
feria=#
```

Y creamos usuarios dándole los respectivos permisos

```
feria=# create user user_adminferia with password 'admin';  
CREATE ROLE  
feria=# create user user_estructura with password 'estructura';  
CREATE ROLE  
feria=# create user user_empresarial with password 'empresarial';  
CREATE ROLE  
feria=# create user user_personaa with password 'persona';  
CREATE ROLE  
feria=#
```

Creemos nuevo volumen donde vamos a duplicar este volumen

```
PS C:\Windows\system32> docker volume create pgdata_feria_ubuntu  
pgdata_feria_ubuntu  
PS C:\Windows\system32>
```

Pasamos los datos

```
PS C:\Windows\system32> docker run --rm -v pgdata_feria:/from -v pgdata_feria_ubuntu:/to alpine sh -c "cp -a /from/. /to/"  
PS C:\Windows\system32>
```

Creamos el contenedor postgresql dentro de Ubuntu con el volumen que duplicamos

```
root@886fb8085d3c:/# docker run -d --name postgresql_feria_ubuntu -e POSTGRES_USER=admin -e POSTGRES_PASSWORD=Tarea_20258. -e POSTGRES_DB=feria -p 5433:5432 -v pgdata_feria_ubuntu:/var/lib/postgresql/data postgres:16  
7e4fb21da42c695aa44f91d031e4eb8d9715ec6f7553e7d070d1cb17f6d13909  
root@886fb8085d3c:/#
```


Actualizamos las listas de paquetes

```
PS C:\Windows\system32> docker exec -it ubuntu_docker bash
root@886fb8085d3c:/# apt-get update
Hit:1 https://packages.microsoft.com/ubuntu/20.04/mssql-server-2019 focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:4 https://packages.microsoft.com/ubuntu/22.04/mssql-server-2022 jammy InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
root@886fb8085d3c:/# apt-get install -y postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree... Done
```

Instalamos postgresql

```
root@886fb8085d3c:/# apt-get install -y postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpopt0 libpq5
  libtypes-serialiser-perl libxslt1.1 locales logrotate mssql-server postgresql-1
libtypes-serialiser-perl libxslt1.1 locales logrotate mssql-server postgresql-1
```

Añadimos con éxito el usuario postgres

```
root@886fb8085d3c:/# useradd -m -s /bin/bash postgres
root@886fb8085d3c:/# useradd -m -s /bin/bash postgres
useradd: user 'postgres' already exists
```

Añadimos carpetas necesarias para archivos internos

```
root@886fb8085d3c:/# mkdir -p /var/lib/postgresql/data
root@886fb8085d3c:/# chown -R postgres:postgres /var/lib/postgresql
```

PostgreSQL no podía arrancar porque faltaba /var/run/postgresql/, así que la añadimos

```
root@886fb8085d3c:/# mkdir -p /var/run/postgresql
root@886fb8085d3c:/# chown postgres:postgres /var/run/postgresql
```

Ingresamos como usuario postgres y arrancamos el motor manualmente

```
root@886fb8085d3c:/# su - postgres
postgres@886fb8085d3c:~$ /usr/lib/postgresql/16/bin/pg_ctl -D /var/lib/postgresql/data -l logfile start
waiting for server to start.... done
server started
```

Entramos a postgresql

```
postgres@886fb8085d3c:~$ psql
psql (16.10 (Ubuntu 16.10-0ubuntu0.24.04.1))
Type "help" for help.

postgres=#
```

Añadimos la extensión para administrar los dos contenedores desde el contenedor Ubuntu (el ps que esta nativo)

```
postgres=# CREATE EXTENSION IF NOT EXISTS postgres_fdw;  
CREATE EXTENSION
```

Creamos una network y añadimos los contenedores a este

```
PS C:\Windows\system32> docker network create postgresql  
9e158499976f41ce112be571ff0910d12fd88cb36adf54522fa5aa6e5e37f2f9  
PS C:\Windows\system32> docker network ls  
NETWORK ID          NAME                DRIVER              SCOPE  
2b1257972f45        bridge              bridge              local  
d5d8c6054c88        db_network          bridge              local  
a3ec57b446ff        host                host                local  
a5a126cf29fa        none                null                local  
9e158499976f        postgresql          bridge              local  
PS C:\Windows\system32> docker network connect postgresql ubuntu_docker  
PS C:\Windows\system32> docker network connect postgresql ubuntu_postgres  
Error response from daemon: No such container: ubuntu_postgres  
PS C:\Windows\system32> docker network connect postgresql postgresql_feria  
Error response from daemon: No such container: postgresql_feria  
PS C:\Windows\system32> docker network connect postgresql postgresql_feria  
PS C:\Windows\system32> docker network connect postgresql postgresql_feria_ubuntu  
PS C:\Windows\system32>
```

Ingresamos de nuevo a postgres nativo de Linux

```
PS C:\Windows\system32> docker exec -it ubuntu_docker bash  
root@886fb8085d3c:/# su - postgres  
postgres@886fb8085d3c:~$ psql  
psql (16.10 (Ubuntu 16.10-0ubuntu0.24.04.1))  
Type "help" for help.  
  
postgres=#
```

Creamos los servidores y mapeamos los usuarios

```
postgres=# DROP SERVER IF EXISTS feria_win CASCADE;  
NOTICE: drop cascades to user mapping for postgres on server feria_win  
DROP SERVER  
postgres=# DROP SERVER IF EXISTS feria_ubuntu CASCADE;  
NOTICE: drop cascades to user mapping for postgres on server feria_ubuntu  
DROP SERVER  
postgres=#  
postgres=# CREATE SERVER feria_win FOREIGN DATA WRAPPER postgres_fdw  
postgres-# OPTIONS (host 'postgresql_feria', port '5432', dbname 'feria');  
CREATE SERVER  
postgres=#  
postgres=# CREATE SERVER feria_ubuntu FOREIGN DATA WRAPPER postgres_fdw  
postgres-# OPTIONS (host 'postgresql_feria_ubuntu', port '5432', dbname 'feria');  
CREATE SERVER  
postgres=# CREATE USER MAPPING FOR postgres SERVER feria_win  
postgres-# OPTIONS (user 'admin', password 'Tarea_20258.');
```

Hacemos una prueba para ver si tenemos control de nuestros servidores y.. perfecto

```
postgres=# CREATE FOREIGN TABLE feria_ubuntu_estructura_feria (
    id INT
) SERVER feria_ubuntu
OPTIONS (schema_name 'estructura', table_name 'feria');
CREATE FOREIGN TABLE
postgres=# SELECT COUNT(*) FROM feria_ubuntu_estructura_feria;
 count
-----
   1000
(1 row)

postgres=#
```

Seguimos con mongoddb

Creamos volumen y lo signamos al contenedor creado, mongo creara otro volumen pero es solo para su configuracion

```
PS C:\Windows\system32> docker volume create mongo_feria_data
mongo_feria_data
PS C:\Windows\system32> docker run -d --name mongo_feria --network mongodb -p 27017:27017 -v mongo_feria_data:/data/db mongo
86f6f83798b0b6424f3f5c8dba8430167ba305033049f240a706761a3b4d6a0f
PS C:\Windows\system32>
```

Ejecutamos contenedor temporal para usar el cliente mongo

```
PS C:\Windows\system32> docker run -it --rm --network mongodb mongo:5.0 mongo --host mongo_feria
Unable to find image 'mongo:5.0' locally
5.0: Pulling from library/mongo
3d7e6a7004dc: Pull complete
b076be86645a: Pull complete
```

Creamos las colecciones

```
> db.createCollection("feria"); db.createCollection("tematica"); db.createCollection("pabellon"); db.createCollection("empresa"); db.createCollection("stand"); db.createCollection("producto"); db.createCollection("persona"); db.createCollection("responsable"); db.createCollection("ponente"); db.createCollection("tipo_visitante"); db.createCollection("visitante"); db.createCollection("charla"); db.createCollection("demostracion"); db.createCollection("registro")
{ "ok" : 1 }
> show collections
charla
demostracion
empresa
feria
pabellon
persona
ponente
producto
registro
responsable
stand
tematica
tipo_visitante
visitante
>
```

Bucle para 1000 documentos

```
> for (let i = 1; i <= 1000; i++) {
...   db.feria.insertOne({
...     nombre: `feria ${i} - ${["tecnología","gastronomía","moda","es"][i % 10]}`,
...     descripcion: `descripción detallada de la feria número ${i}`,
...   });
...   db.tematica.insertOne({
...     tematica: `tema-${["tech","food","fashion","auto","building"][i % 15]}-${i}`
...   });
... }
```

Verificamos los documentos por conexión

```
> db.feria.count()
1000
> db.tematica.count()
1000
> db.pabellon.count()
1000
> db.empresa.count()
1000
> db.stand.count()
1000
> db.producto.count()
1000
> db.persona.count()
1000
> db.responsable.count()
1000
> db.ponente.count()
1000
> db.tipo_visitante.count()
1000
> db.visitante.count()
1000
> db.charla.count()
1000
> db.demonstracion.count()
1000
> db.registro.count()
1000
>
```

Creamos los “schemas”

```
> use estructura
switched to db estructura
> use sector_empresarial
switched to db sector_empresarial
> use personas_actividades
switched to db personas_actividades
>
```

Pasamos las colecciones a los “schemas”

```
switched to db feria
> // estructura
> db.getSiblingDB("feria").feria.find().forEach(doc => db.getSiblingDB("estructura").feria.insert(doc));
> db.getSiblingDB("feria").tematica.find().forEach(doc => db.getSiblingDB("estructura").tematica.insert(doc));
> db.getSiblingDB("feria").pabellon.find().forEach(doc => db.getSiblingDB("estructura").pabellon.insert(doc));
>
> // sector_empresarial
> db.getSiblingDB("feria").empresa.find().forEach(doc => db.getSiblingDB("sector_empresarial").empresa.insert(doc));
> db.getSiblingDB("feria").stand.find().forEach(doc => db.getSiblingDB("sector_empresarial").stand.insert(doc));
> db.getSiblingDB("feria").producto.find().forEach(doc => db.getSiblingDB("sector_empresarial").producto.insert(doc));
> db.getSiblingDB("feria").responsable.find().forEach(doc => db.getSiblingDB("sector_empresarial").responsable.insert(doc));
>
> // personas_actividades
> db.getSiblingDB("feria").persona.find().forEach(doc => db.getSiblingDB("personas_actividades").persona.insert(doc));
> db.getSiblingDB("feria").ponente.find().forEach(doc => db.getSiblingDB("personas_actividades").ponente.insert(doc));
> db.getSiblingDB("feria").tipo_visitante.find().forEach(doc => db.getSiblingDB("personas_actividades").tipo_visitante.insert(doc));
> db.getSiblingDB("feria").visitante.find().forEach(doc => db.getSiblingDB("personas_actividades").visitante.insert(doc));
> db.getSiblingDB("feria").charla.find().forEach(doc => db.getSiblingDB("personas_actividades").charla.insert(doc));
> db.getSiblingDB("feria").demonstracion.find().forEach(doc => db.getSiblingDB("personas_actividades").demonstracion.insert(doc));
> db.getSiblingDB("feria").registro.find().forEach(doc => db.getSiblingDB("personas_actividades").registro.insert(doc));
>
```

Creamos usuarios

```
> use admin
switched to db admin
>
> db.createUser({
...   user: "user_adminferia",
...   pwd: "admin",
...   roles: [
...     { role: "readWrite", db: "estructura" },
...     { role: "readWrite", db: "sector_empresarial" },
...     { role: "readWrite", db: "personas_actividades" }
...   ]
... })
Successfully added user: {
  "user" : "user_adminferia",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "estructura"
    },
    {
      "role" : "readWrite",
      "db" : "sector_empresarial"
    },
    {
      "role" : "readWrite",
      "db" : "personas_actividades"
    }
  ]
}
>
> db.createUser({
...   user: "user_estructura",
...   pwd: "estructura",
...   roles: [{ role: "readWrite", db: "estructura" }]
... })
```

Comprobamos que los usuarios se hayan creado

```
> db.getUsers()
[
  {
    "_id" : "admin.user_adminferia",
    "userId" : UUID("9fca83e3-e421-42e0-ba76-5d15c112469c"),
    "user" : "user_adminferia",
    "db" : "admin",
    "roles" : [
      {
        "role" : "readWrite",
        "db" : "estructura"
      },
      {

```

Duplicamos el volumen

```
PS C:\Windows\system32> docker run --rm -v mongo_feria_data:/from -v mongo_feria_data_ubuntu:/to alpine ash -c "cd /from && cp -a . /to"
PS C:\Windows\system32>
```

Creamos el contenedor utilizando el volumen que duplicamos añadiéndolo a la network de mongo

```
root@886fb8085d3c:/# docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
2b1257972f45    bridge   bridge  local
d5d8c6054c88    db_network bridge  local
a3ec57b446ff     host     host    local
a848979adc4f     mongodb  bridge  local
a5a126cf29fa     none     null    local
9e158499976f     postgresql bridge  local
root@886fb8085d3c:/# docker run -d --name mongo_feria_ubuntu --network mongodb -v mongo_feria_data_ubuntu:/data/db -p 27018:27017 mongo
aa0feca97b685afd6c5d677d6aa9f69ccba1c5392666887b7125ec6b8251a427
```

Dentramos al contenedor

```
root@886fb8085d3c:/# docker exec -it mongo_feria_ubuntu mongosh
Current Mongosh Log ID: 68fe81581465b775fcce5f46
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:      8.0.15
Using Mongosh:      2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
estructura>

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal-policy).
You can opt-out by running the disableTelemetry() command.
```

Verificamos que el volumen se haya aplicado correctamente

```
test> show dbs
admin                132.00 KiB
config               72.00 KiB
estructura           220.00 KiB
feria                 1.02 MiB
local                72.00 KiB
personas_actividades 528.00 KiB
sector_empresarial   292.00 KiB
test> use estructura
switched to db estructura
estructura> show collections
feria
pabellon
tematica
estructura> db.feria.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
1000
estructura>
```

La instalación de mongo en Linux se me complico un poco entro en un bucle y me toco cerrar la ventana pero por suerte funciona

Tenia el sistema dañado de tantos intentos para descargar sqlserver

```
dpkg --purge --force-all mssql-server
```

```
rm -rf /opt/mssql/var/opt/mssql
```

```
rm /etc/apt/sources.list.d/mssql-server.list
```

recupere paquetes dañados

```
echo -e "Package: base-files\nStatus: install ok installed\nPriority: required\nSection:
admin\nInstalled-Size: 0\nMaintainer: Ubuntu Developers <ubuntu-devel-
discuss@lists.ubuntu.com>\nArchitecture: amd64\nVersion: 1\nDescription: Minimal dpkg status
file to recover apt" > /var/lib/dpkg/status
```

```
apt-get update
```

configuración para el repositorio de mongo

```
curl -fsSL https://pgp.mongodb.com/server-6.0.asc | gpg --dearmor -o  
/usr/share/keyrings/mongodb-server-6.0.gpg
```

```
echo "deb [signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg]  
https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" >  
/etc/apt/sources.list.d/mongodb-org-6.0.list
```

```
apt-get update
```

arregle dependencias rotas

```
wget https://launchpad.net/ubuntu/+archive/primary/+files/libssl1.1_1.1.1f-  
1ubuntu2.24_amd64.deb
```

```
dpkg -i libssl1.1_1.1.1f-1ubuntu2.24_amd64.deb
```

arranque manual (aquí entro en bucle)

```
mkdir -p /data/db
```

```
mongod --dbpath /data/db
```

Y finalmente

```
PS C:\Windows\system32> docker exec -it ubuntu_docker bash
root@886fb8085d3c:/# which mongod
/usr/bin/mongod
root@886fb8085d3c:/# ps aux | grep mongod
root      8399  0.5  1.6 2585784 131072 pts/1    S1+   15:51   0:02 mongod --dbpath /data/db
root      8460  0.0  0.0   3528   1792 pts/2      S+    15:58   0:00 grep --color=auto mongod
root@886fb8085d3c:/# ls -la /data/db
total 240
drwxr-xr-x 4 root root 4096 Oct 26 15:57 .
drwxr-xr-x 3 root root 4096 Oct 26 15:54 ..
-rw-r----- 1 root root 50 Oct 26 15:51 WiredTiger
-rw-r----- 1 root root 21 Oct 26 15:51 WiredTiger.lock
-rw-r----- 1 root root 1465 Oct 26 15:57 WiredTiger.turtle
-rw-r----- 1 root root 69632 Oct 26 15:57 WiredTiger.wt
-rw-r----- 1 root root 4096 Oct 26 15:51 WiredTigerHS.wt
-rw-r----- 1 root root 20480 Oct 26 15:52 _mdb_catalog.wt
-rw-r----- 1 root root 20480 Oct 26 15:52 collection-0-7899193910552606137.wt
-rw-r----- 1 root root 20480 Oct 26 15:52 collection-2-7899193910552606137.wt
-rw-r----- 1 root root 4096 Oct 26 15:51 collection-4-7899193910552606137.wt
drwx-r----- 2 root root 4096 Oct 26 15:58 diagnostic.data
-rw-r----- 1 root root 20480 Oct 26 15:52 index-1-7899193910552606137.wt
-rw-r----- 1 root root 20480 Oct 26 15:52 index-3-7899193910552606137.wt
-rw-r----- 1 root root 4096 Oct 26 15:51 index-5-7899193910552606137.wt
-rw-r----- 1 root root 4096 Oct 26 15:51 index-6-7899193910552606137.wt
drwx-r----- 2 root root 4096 Oct 26 15:51 journal
-rw-r----- 1 root root 5 Oct 26 15:51 mongod.lock
-rw-r----- 1 root root 20480 Oct 26 15:53 sizeStorer.wt
-rw-r----- 1 root root 114 Oct 26 15:51 storage.bson
root@886fb8085d3c:/# mongosh
Current Mongosh Log ID: 68fe8b92a165d70765ce5f46
Connecting to:   mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:  6.0.26
Using Mongosh:  2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
  The server generated these startup warnings when booting
  2025-10-26T15:51:29.803-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2025-10-26T15:51:29.803-05:00: You are running this process as the root user, which is not recommended
  2025-10-26T15:51:29.803-05:00: 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
  2025-10-26T15:51:29.803-05:00: vm.max_map_count is too low
  -----
test>
```


COMANDOS UTILIZADOS

Docker images: visualizar las imágenes instaladas

docker pull: descargar imagen

docker run --name mysql-feria -e MYSQL_ROOT_PASSWORD=feria -e MYSQL_DATABASE=feria -p 3306:3306 -d mysql:latest: comando para instanciar la imagen y generar el contenedor de mysql

docker run: instanciar imagen, echar a correr el contenedor

--name mysql-feria: asignarle nombre del contenedor

-e MYSQL_ROOT_PASSWORD=feria: contraseña de usuario root

-e MYSQL_DATABASE=feria: creación de la base de datos feria

-p 3306:3306: asignar puerto

-d mysql:latest: imagen instanciada, y dando la orden para que se ejecute el contenedor en segundo plano "detached" con -d

show tables: muestra todas las tablas

docker attach Ubuntu_docker: entrar a la terminal de Linux obviamente primero hay que iniciarlo

docker start: inicializar contenedores.

-i (interactive): Mantiene la entrada estándar abierta, lo que permite interactuar con la terminal.

-t (tty): Asigna una pseudo-TTY (terminal), lo que hace que la experiencia de la *shell* sea funcional.

Comando para descargar sqlserver en el os de Ubuntu

apt update && apt install -y gnupg wget

apt update && apt install -y curl apt-transport-https && curl

<https://packages.microsoft.com/keys/microsoft.asc> | apt-key add - && add-apt-repository

"\$(wget -qO- <https://packages.microsoft.com/config/ubuntu/20.04/mssql-server-2019.list>)" &&

apt update && apt install -y mssql-server

actualizamos paquetes e instalamos postgres

apt update

apt install -y postgresql postgresql-contrib

añadir usuario postgresql

useradd -m -s /bin/bash postgres

preparamos directorios:

mkdir -p /var/lib/postgresql/data

chown -R postgres:postgres /var/lib/postgresql

mkdir -p /var/run/postgresql

chown postgres:postgres /var/run/postgresql

arrancamos motor manualmente(debemos conocer la version de nuestro postgresql

ls /usr/lib/postgresql/):

/usr/lib/postgresql/16/bin/pg_ctl -D /var/lib/postgresql/data -l logfile start

Ingresamos a postgres: Psql

Cremos nuevo volumen

```
PS C:\Windows\system32> docker volume create pgdata_feria_ubuntu
pgdata_feria_ubuntu
PS C:\Windows\system32>
```

Duplicamos volumen

docker run --rm -v pgdata_feria:/from -v \$(pwd)/pgdata_feria_ubuntu:/to alpine sh -c "cp -a /from/. /to/"

duplicamos volumen de mongo

docker run --rm -v mongo_feria_data:/from -v mongo_feria_data_ubuntu:/to alpine ash -c "cd /from && cp -a . /to"

descargar mongo en Linux ubuntu

```
apt-get update && apt-get install -y gnupg curl && curl -fsSL https://pgp.mongodb.com/server-6.0.asc | gpg --dearmor -o /usr/share/keyrings/mongodb-server-6.0.gpg && echo "deb [ signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0multiverse" | tee /etc/apt/sources.list.d/mongodb-org-6.0.list && apt-get update && apt-get install -y mongodb-org
```

GLOSARIO

DNS (Domain Name System) Imagina que es como la libreta de teléfonos de internet. Cuando tú quieres llamar a alguien, no memorizas su número completo, memorizas su nombre. El DNS hace exactamente eso: convierte nombres fáciles de recordar (como "google.com") en direcciones numéricas que las computadoras entienden (como "142.250.185.46"). Sin DNS, tendrías que memorizar números largos para cada sitio web.

Network Es simplemente una red de computadoras conectadas entre sí, como una telaraña. Así como las personas en una ciudad se comunican mediante teléfono, email o en persona, las computadoras en una red se comunican compartiendo información. Pueden ser desde 2 computadoras hasta millones en internet.

Bridge Es como un puente que conecta dos redes diferentes para que se comuniquen. Imagina dos pueblos separados por un río: el puente permite que la gente de un pueblo viaje al otro. Un bridge en networking hace lo mismo: permite que dispositivos de una red se conecten y compartan información con dispositivos de otra red, traduciendo entre ellos si es necesario.