

**Ejercicio:** Para evitar la saturación de su servidor MySQL han decidido replicarlo en otro servidor. La replicación de MySQL es un proceso en el que los datos de un servidor de base de datos MySQL (el maestro) se copian automáticamente en uno o más servidores de bases de datos MySQL (el esclavo).

1- Para poder practicar la replicación de un servidor MySQL en otro, obviamente necesitaremos dos servidores MySQL ejecutándose simultáneamente en dos ordenadores conectados. Quiero utilizar docker para la tarea. El servidor MySQL que actuará como maestro (maestro) y en el sistema operativo anfitrión se ejecutará el servidor MySQL que actuará como esclavo (esclavo). Hay que asegurarse que los dos servidores pueden conectar entre sí, para ello lo más sencillo es configurar ambos servidores (sus direcciones IP) para que estén en la misma red (configurar IP de los dos servidores y la prueba, haciendo ping, de que pueden verse entre sí).

2- Configuración del servidor maestro:

Hay que editar el archivo de configuración my.ini con permisos de administrador. (o lo que corresponda a docker)

3- Configuración del servidor esclavo

Para ello editamos el archivo de configuración MySQL “my.ini” en el servidor que hará de esclavo. Hay que configurar la directiva server-id y poner un identificador diferente al que se haya usado en el maestro, (o lo que corresponda a docker)

4- Comprobación de funcionamiento.

Una vez que tenemos a maestro y esclavo configurados y en funcionamiento, llega el momento de comprobar si funciona la replicación. Para ello podemos crear una base de datos en el maestro y comprobar si se replica en el esclavo:

**¡IMPORTANTE!** En tu tarea debes añadir una fila con tu nombre y apellidos (Ivana Sánchez Pérez) en la tabla del servidor maestro para verificar la autoría. Sin ese requisito la tarea no será válida.

5- Comprobación de no bidireccionalidad en las actualizaciones.

Ya tenemos al esclavo replicando todo los cambios que se producen en el maestro, pero... ¿qué pasará con los cambios que se produzcan en el esclavo? ¿se replicarán también en maestro? Vamos a comprobarlo

6- Comprobación de parada del esclavo y recuperación de actualizaciones en el siguiente arranque.

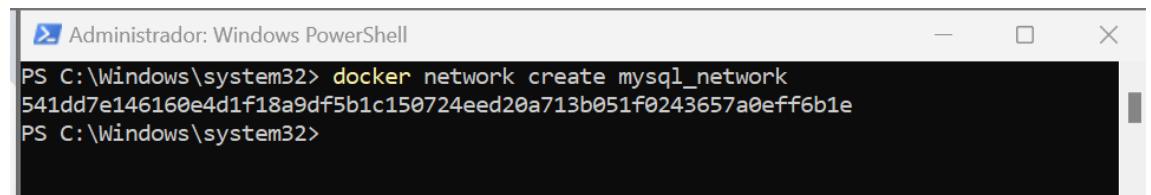
- a) Paramos el esclavo con STOP REPLICA.
- b) Insertamos una fila nueva en el maestro con el esclavo parado.
- c) Arrancamos el esclavo.

d) Comprobamos en el esclavo que recupera las actualizaciones hechas mientras estuvo parado.

## 1- CREAR UNA RED DOCKER

Docker utiliza redes internas para que los contenedores puedan comunicarse.

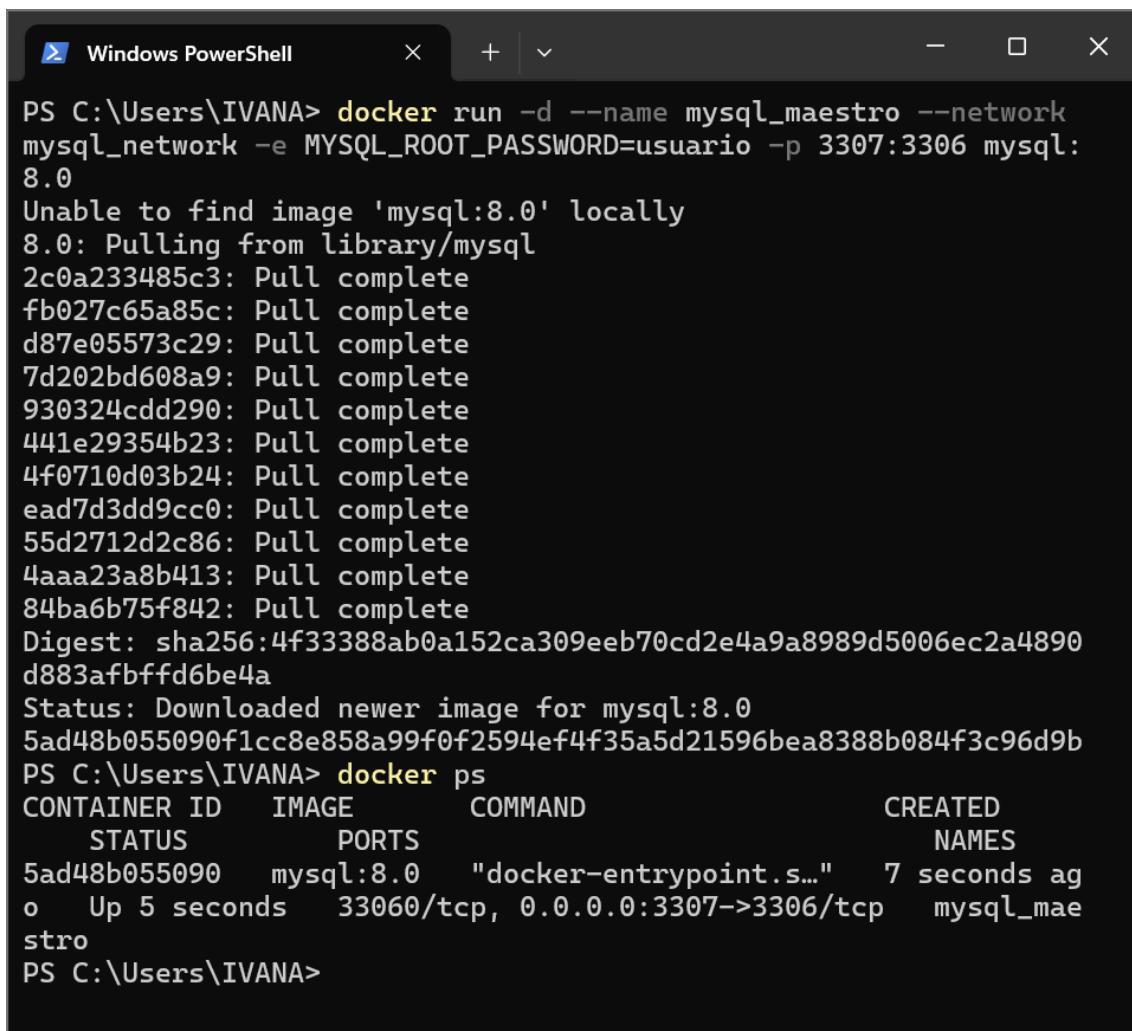
Para ello vamos a crear una llamada roja mysql\_network que permitirá que el maestro y el esclavo se vean entre sí.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker network create mysql_network
541dd7e146160e4d1f18a9df5b1c150724eed20a713b051f0243657a0eff6b1e
PS C:\Windows\system32>
```

### - CREAR EL CONTENEDOR MYSQL MAESTRO

Bajamos y crearemos el servidor maestro de MySQL

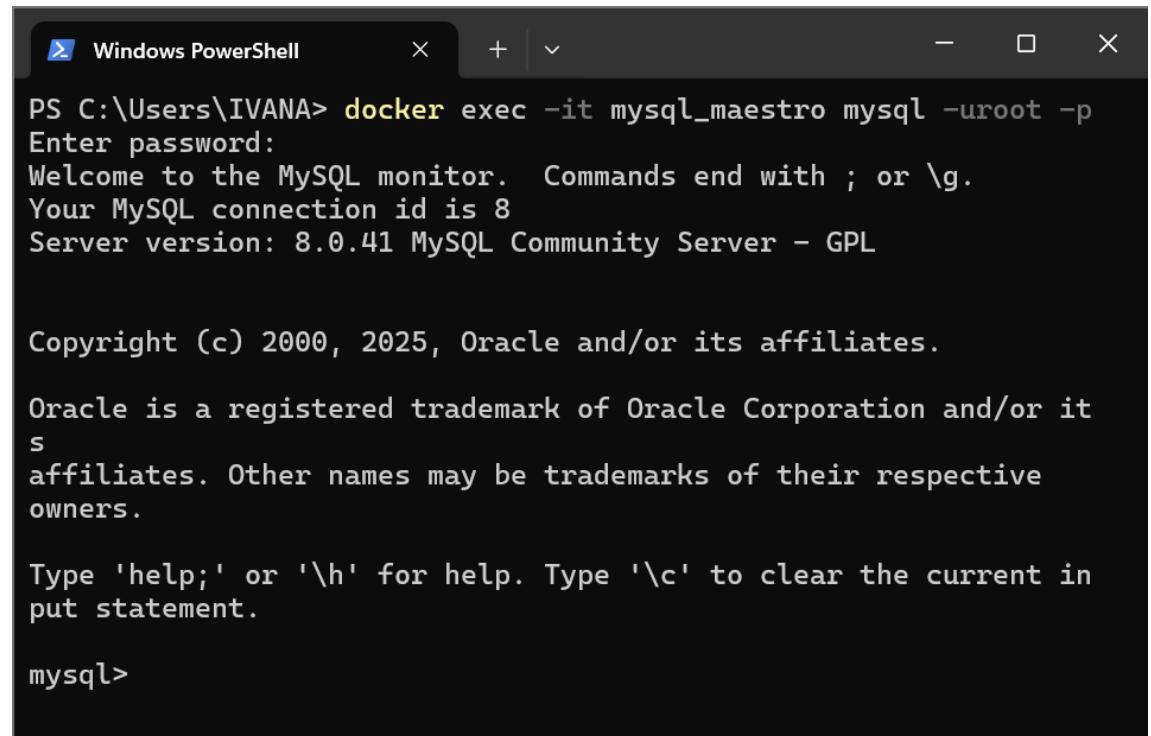


```
Windows PowerShell
PS C:\Users\IVANA> docker run -d --name mysql_maestro --network mysql_network -e MYSQL_ROOT_PASSWORD=usuario -p 3307:3306 mysql:8.0
Unable to find image 'mysql:8.0' locally
8.0: Pulling from library/mysql
2c0a233485c3: Pull complete
fb027c65a85c: Pull complete
d87e05573c29: Pull complete
7d202bd608a9: Pull complete
930324bdd290: Pull complete
441e29354b23: Pull complete
4f0710d03b24: Pull complete
ead7d3dd9cc0: Pull complete
55d2712d2c86: Pull complete
4aaa23a8b413: Pull complete
84ba6b75f842: Pull complete
Digest: sha256:4f33388ab0a152ca309eeb70cd2e4a9a8989d5006ec2a4890
d883afbffd6be4a
Status: Downloaded newer image for mysql:8.0
5ad48b055090f1cc8e858a99f0f2594ef4f35a5d21596bea8388b084f3c96d9b
PS C:\Users\IVANA> docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
5ad48b055090 mysql:8.0 "docker-entrypoint.s..." 7 seconds ago
Up 5 seconds 33060/tcp, 0.0.0.0:3307->3306/tcp mysql_maestro
PS C:\Users\IVANA>
```

- CONFIGURAR EL MAESTRO

Configuramos el contenedor maestro

- 1- Ingresamos al contenedor



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command entered is "PS C:\Users\IVANA> docker exec -it mysql\_maestro mysql -uroot -p". The response shows the MySQL monitor welcome screen, including the password prompt, server version (8.0.41), and copyright information. At the bottom, it shows the MySQL prompt "mysql>".

```
PS C:\Users\IVANA> docker exec -it mysql_maestro mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 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.

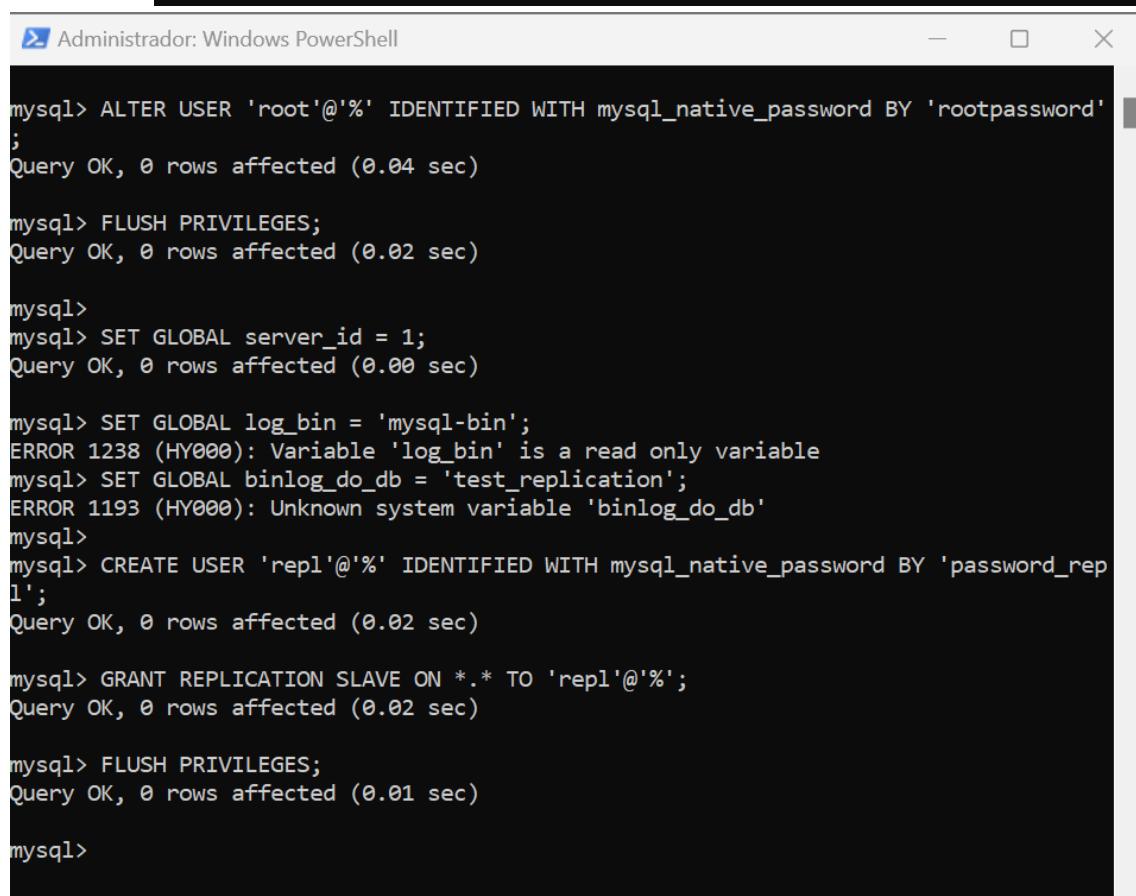
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- 2- Configuramos la replicación en el maestro

Antes tenemos que crear un archivo con los siguientes datos dentro de nuestro mysql\_maestro. Así que accedemos con exec -it mysql\_maestro bash para crearlo. Una vez hecho, salimos y guardamos los cambios y reiniciamos con el comando docker restart mysql\_maestro

```
PS C:\Users\IVANA> docker exec -it mysql_maestro bash
bash-5.1# mysql --verbose --help | grep -A 1 "Default options"
Default options are read from the following files in the given o
rder:
/etc/my.cnf /etc/mysql/my.cnf /usr/etc/my.cnf ~/.my.cnf
bash-5.1# nano /etc/my
my.cnf my.cnf.d/ mysql/
bash-5.1# nano /etc/my.cnf
bash: nano: command not found
bash-5.1# vi /etc/my.cnf
bash: vi: command not found
bash-5.1# echo "[mysqld]"
log_bin = mysql-bin
server-id = 1
binlog_do_db = test_replication" > /etc/mysql/my.cnf
bash-5.1# cat /etc/mysql/my.cnf
[mysqld]
log_bin = mysql-bin
server-id = 1
binlog_do_db = test_replication
bash-5.1#
```



The screenshot shows a Windows PowerShell window titled "Administrador: Windows PowerShell". The window contains a series of MySQL commands and their outputs:

```
mysql> ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'rootpassword';
Query OK, 0 rows affected (0.04 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> SET GLOBAL server_id = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL log_bin = 'mysql-bin';
ERROR 1238 (HY000): Variable 'log_bin' is a read only variable
mysql> SET GLOBAL binlog_do_db = 'test_replication';
ERROR 1193 (HY000): Unknown system variable 'binlog_do_db'
mysql>
mysql> CREATE USER 'repl'@'%' IDENTIFIED WITH mysql_native_password BY 'password_repl';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

3- Verificamos el estado del binlog

```
Administrator: Windows PowerShell
mysql>
mysql>
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| binlog.000002 | 1276 | | | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

- Creamos el contenedor MySQL Esclavo

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker run -d --name mysql_esclavo --network mysql_network -e MYSQL_ROOT_PASSWORD=usuario -p 3308:3306 mysql:8.0
30e6464ee8934d28b184ed722cc8dd162b836f69a49e904eb2c1402c3556c05d
PS C:\Windows\system32> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
30e6464ee893 mysql:8.0 "docker-entrypoint.s..." 16 seconds ago Up 15 seconds
33060/tcp, 0.0.0.0:3308->3306/tcp mysql_esclavo
6e47bd63c084 mysql:8.0 "docker-entrypoint.s..." 9 minutes ago Up 9 minutes
33060/tcp, 0.0.0.0:3307->3306/tcp mysql_master
PS C:\Windows\system32>
```

- Configuramos el esclavo

1- Ingresamos en el contenedor esclavo

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker exec -it mysql_esclavo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41 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>
```

2- Configuramos la conexión con el contenedor maestro

```

mysql> STOP REPLICA;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CHANGE MASTER TO
      -> MASTER_HOST='mysql_master',
      -> MASTER_USER='repl',
      -> MASTER_PASSWORD='usuario',
      -> MASTER_LOG_FILE='binlog.000002',
      -> MASTER_LOG_POS=1276;
Query OK, 0 rows affected, 8 warnings (0.13 sec)

```

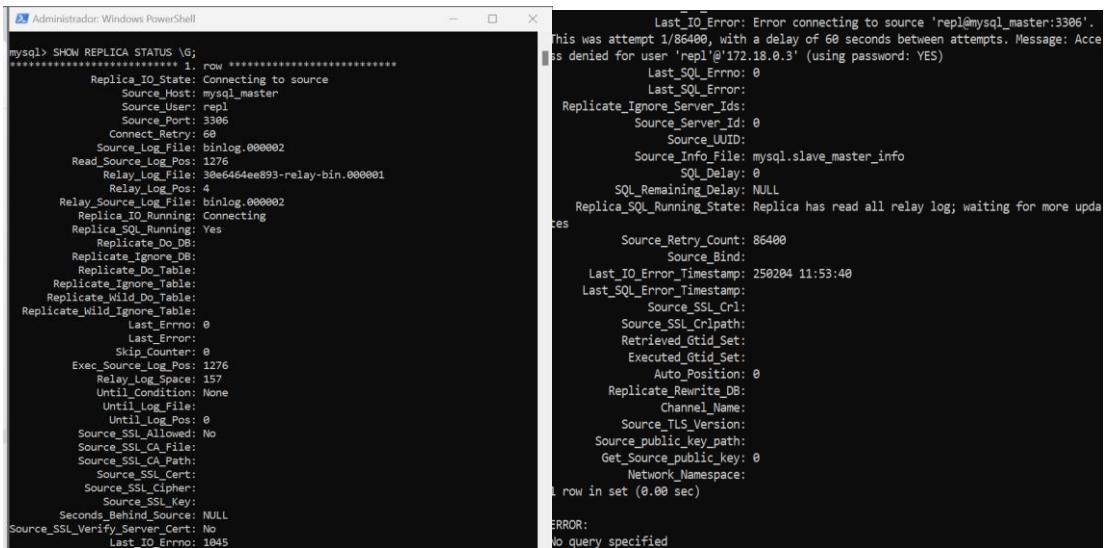
```

mysql>
mysql> START REPLICA;
Query OK, 0 rows affected (0.06 sec)

mysql>

```

### 3- Verificamos si la replicación funciona



```

Administrator: Windows PowerShell
mysql> SHOW REPLICATION STATUS \G
***** Row 1: *****
Replica_IO_State: Connecting to source
  Source_Host: mysql_master
  Source_User: repl
  Source_Port: 3306
  Connect_Retry: 60
  Source_Log_File: binlog.000002
  Read_Source_Log_Pos: 1276
  Relay_Log_File: 30e6464ee893-relay-bin.000001
  Relay_Log_Pos: 4
Relay_Source_Log_File: binlog.000002
  Replica_IO_Running: Connecting
  Replica_SQL_Running: Yes
  Replicate_Do_DB:
  Replicate_Ignore_DB:
  Replicate_Do_Table:
  Replicate_Ignore_Table:
  Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
    Last_Error: 0
    Last_Error:
    Skip_Counter: 0
  Exec_Source_Log_Pos: 1276
  Relay_Log_Skip: 157
  Until_Condition: None
  Until_Log_File:
  Until_Log_Pos: 0
  Source_SSL_Allowed: No
  Source_SSL_CA_File:
  Source_SSL_CA_Path:
  Source_SSL_Cert:
  Source_SSL_Cipher:
  Source_SSL_Key:
  Seconds_Behind_Source: NULL
  Source_SSL_Verify_Server_Cert: No
  Last_IO_Error: 1045
Last_IO_Error: Error connecting to source 'repl@mysql_master:3306'.
This was attempt 1/86400, with a delay of 60 seconds between attempts. Message: Access denied for user 'repl'@'172.18.0.3' (using password: YES)
  Last_SQL_Erno: 0
  Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
  Source_Server_Id: 0
  Source_UUID:
  Source_Info_File: mysql.slave_master_info
  SQL_Delay: 0
  SQL_Remaining_Delay: NULL
  Replica_SQL_Running_State: Replica has read all relay log; waiting for more updates
  Source_Retry_Count: 86400
  Source_Bind:
  Last_IO_Error_Timestamp: 250204 11:53:40
  Last_SQL_Error_Timestamp:
    Source_SSL_Crl:
    Source_SSL_Crlpath:
    Retrieved_Gtid_Set:
    Executed_Gtid_Set:
      Auto_Position: 0
    Replicate_Rewrite_DB:
      Channel_Name:
      Source_TLS_Version:
      Source_public_key_path:
      Get_Source_public_key: 0
      Network_Namespace:
1 row in set (0.00 sec)

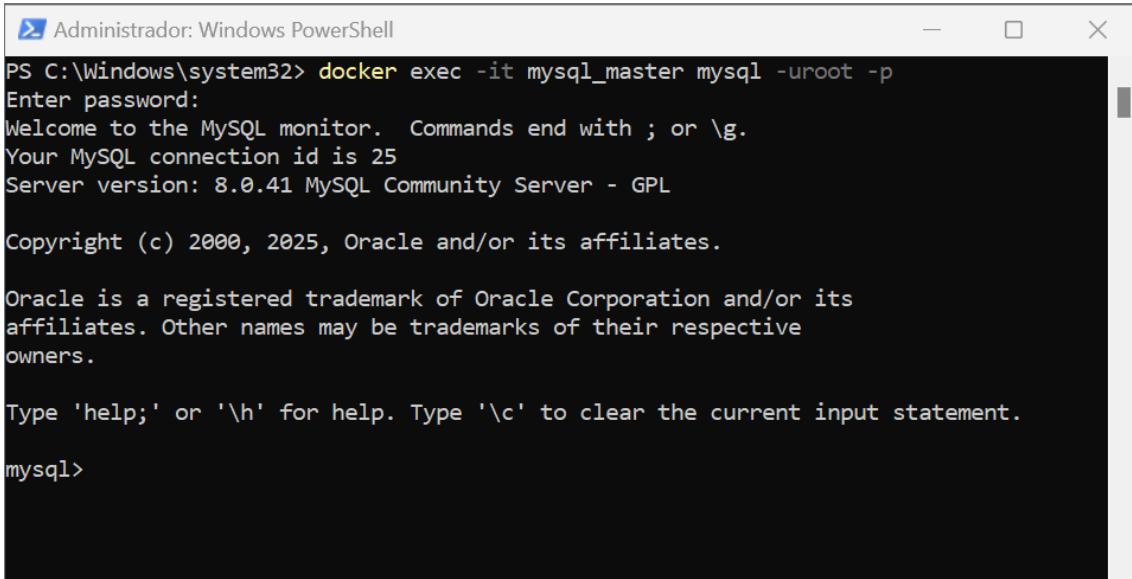
ERROR:
No query specified

```

Observo que me da error: `Last_IO_Error: Error connecting to source 'repl@mysql_master:3306'`. This was attempt 1/86400, with a delay of 60 seconds between attempts. Message: Access denied for user 'repl'@'172.18.0.3' (using password: YES).

La causa del problema es que el usuario `repl` no tiene permisos para conectarse desde la ip `172.18.0.3` (IP del contenedor esclavo). Para solucionarlo:

- 1- Accedemos al contenedor del maestro



```
PS C:\Windows\system32> docker exec -it mysql_master mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.41 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.

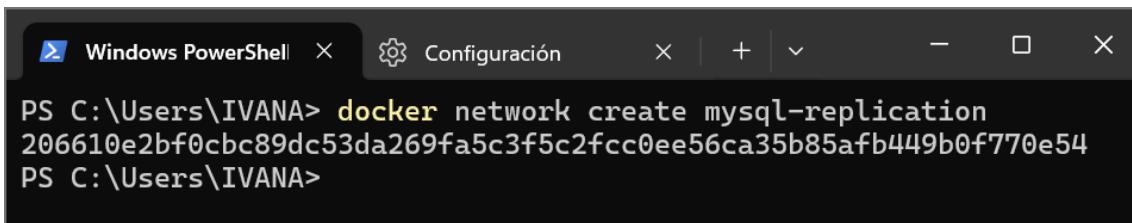
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

## Paso 1: Configuración de la red entre los servidores

Para que el maestro y el esclavo puedan comunicarse, los ejecutaremos en una misma red de Docker.

```
# Crear una red de Docker para la replicación
docker network create mysql-replication
```



```
PS C:\Users\IVANA> docker network create mysql-replication
206610e2bf0cbc89dc53da269fa5c3f5c2fcc0ee56ca35b85afb449b0f770e54
PS C:\Users\IVANA>
```

## Paso 2: Configuración del servidor maestro

### 2.1 Levantar el contenedor MySQL Maestro

Ejecutamos el siguiente comando para iniciar el contenedor del servidor maestro:

```
docker run --name mysql-master --network mysql-replication -e
MYSQL_ROOT_PASSWORD=usuario -e MYSQL_DATABASE=replication_db -e
```

```
MYSQL_USER=replication_user -e MYSQL_PASSWORD=usuario -p 3306:3306 -d  
mysql:latest
```

The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command entered is:

```
PS C:\Users\IVANA> docker run --name mysql-master --network mysql-replication -e MYSQL_ROOT_PASSWORD=usuario -e MYSQL_DATABASE=replication_db -e MYSQL_USER=replication_user -e MYSQL_PASSWORD=usuario -p 3306:3306 -d mysql:latest
```

The output indicates that the image 'mysql:latest' was not found locally and was pulled from the library. It lists several layers being pulled, their IDs, and their status (Pull complete). It also shows the image digest and the status of the download.

```
PS C:\Users\IVANA> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	NAME
STATUS	PORTS			
S	74fcfb4f3031	mysql:latest "docker-entrypoint.s..."	24 seconds ago	mysql-master
Up 22 seconds	0.0.0.0:3306->3306/tcp, 33060/tcp			

```
PS C:\Users\IVANA>
```

## 2.2 Configurar el maestro

Accedemos al contenedor del maestro:

```
docker exec -it mysql-master bash
```

The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command entered is:

```
PS C:\Users\IVANA> docker exec -it mysql-master bash
```

The response shows the user is now inside the mysql-master container, with the prompt "bash-5.1#".

```
bash-5.1#
```

Editamos el archivo de configuración my.cnf:

```
vi /etc/my.cnf
```

Añadimos las siguientes líneas:

```
[mysqld]
server-id=1
log-bin=mysql-bin
binlog-do-db=replication_db
```

The screenshot shows a Windows PowerShell window with the title "Windows PowerShell" and a tab labeled "Configuración". The main area displays a MySQL configuration file (my.cnf) with the following content:

```
host-cache-size=0
skip-name-resolve
datadir=/var/lib/mysql
socket=/var/run/mysqld/mysqld.sock
secure-file-priv=/var/lib/mysql-files
user=mysql

pid-file=/var/run/mysqld/mysqld.pid
[client]
socket=/var/run/mysqld/mysqld.sock

!includedir /etc/mysql/conf.d/

[mysqld]
server-id=1
log-bin=mysql-bin
binlog-do-db=replication_db
```

A red box highlights the section starting with "[mysqld]". Below the configuration file, the terminal prompt shows:

```
Ln 37, Col 28 | 1.067 caracteres. | 100% | Unix (LF) | UTF-8
bash-5.1# exit
exit
PS C:\Users\IVANA> docker cp mysql-master:/etc/my.cnf ./my.cnf
Successfully copied 2.56kB to C:\Users\IVANA\my.cnf
PS C:\Users\IVANA> notepad ./my.cnf
PS C:\Users\IVANA> docker cp ./my.cnf mysql-master:/etc/my.cnf
Successfully copied 3.07kB to mysql-master:/etc/my.cnf
PS C:\Users\IVANA>
```

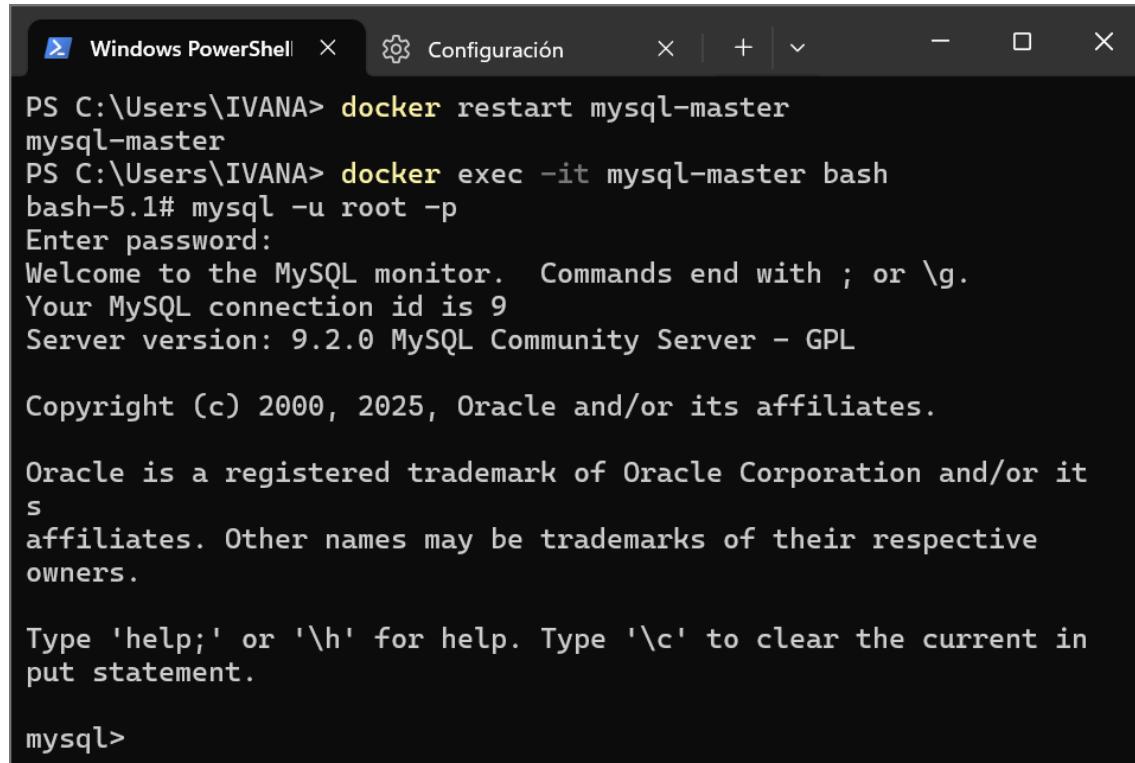
Reiniciamos el contenedor

The screenshot shows a Windows PowerShell window with the title "Windows PowerShell" and a tab labeled "Configuración". The terminal prompt shows:

```
PS C:\Users\IVANA> docker restart mysql-master
mysql-master
PS C:\Users\IVANA>
```

Accedemos a MySQL y creamos un usuario para la replicación:

```
mysql -u root -p
```



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command "docker restart mysql-master" is run, followed by "docker exec -it mysql-master bash". The MySQL monitor starts, displaying the welcome message, server version (9.2.0), and copyright information. It also mentions Oracle trademarks and a note about help. Finally, the prompt "mysql>" is shown.

```
PS C:\Users\IVANA> docker restart mysql-master
mysql-master
PS C:\Users\IVANA> docker exec -it mysql-master bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 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>
```

Ejecutamos:

```
CREATE USER 'replicator'@'%' IDENTIFIED WITH mysql_native_password BY
'replica_pass';
```

```
GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'%';
```

```
FLUSH PRIVILEGES;
```

```
SHOW MASTER STATUS;
```

```
mysql>
mysql> ALTER USER 'root'@'%' IDENTIFIED BY 'usuario';
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'root'@'%';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> |
```

Anotamos los valores de File y Position que nos servirán para configurar el esclavo.

### **Paso 3: Configuración del servidor esclavo**

#### **3.1 Levantar el contenedor MySQL Esclavo**

Ejecutamos el siguiente comando para iniciar el contenedor del esclavo:

```
docker run --name mysql-slave \
--network mysql-replication \
-e MYSQL_ROOT_PASSWORD=rootpassword \
-p 3307:3306 \
-d mysql:latest
```

#### **3.2 Configurar el esclavo**

Accedemos al contenedor del esclavo:

```
docker exec -it mysql-slave bash
```

Editamos el archivo my.cnf:

```
vi /etc/mysql/my.cnf
```

Añadimos las siguientes líneas:

```
[mysqld]
server-id=2
relay-log=mysql-relay-bin
```

Reiniciamos MySQL dentro del contenedor:

```
service mysql restart
```

Accedemos a MySQL y configuramos la replicación:

```
mysql -u root -p
```

Ejecutamos:

```
CHANGE MASTER TO
MASTER_HOST='mysql-master',
MASTER_USER='replicator',
MASTER_PASSWORD='replica_pass',
MASTER_LOG_FILE='[File]',
MASTER_LOG_POS=[Position];
```

```
START REPLICA;  
SHOW REPLICAS STATUS\G;
```

#### **Paso 4: Comprobación de funcionamiento**

Desde el maestro, creamos una base de datos y una tabla:

```
CREATE DATABASE test_db;  
USE test_db;
```

```
CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name  
VARCHAR(100));
```

```
INSERT INTO users (name) VALUES ('Ivana Sánchez Pérez');
```

Desde el esclavo, verificamos que la tabla se haya replicado:

```
USE test_db;  
SELECT * FROM users;
```

Si la fila con el nombre "Ivana Sánchez Pérez" aparece en el esclavo, la replicación funciona correctamente.

#### **Paso 5: Comprobación de no bidireccionalidad**

Intentamos insertar un dato en el esclavo:

```
INSERT INTO users (name) VALUES ('Test User');
```

Si MySQL arroja un error de solo lectura, significa que la replicación es unidireccional.

#### **Paso 6: Parada del esclavo y recuperación**

##### **6.1 Detener el esclavo**

```
STOP REPLICA;
```

##### **6.2 Insertar una nueva fila en el maestro**

```
INSERT INTO users (name) VALUES ('Nueva Entrada');
```

##### **6.3 Arrancar el esclavo**

```
START REPLICA;
```

##### **6.4 Verificar que el esclavo recupera las actualizaciones**

```
SELECT * FROM users;
```

Si la nueva fila aparece en el esclavo, la recuperación es correcta.

¡Con esto hemos completado la configuración y validación de la replicación en MySQL usando Docker!