



Replicación de bases de datos MySQL

17/05/2022

Adrián Acosa Sánchez

3º Grado en Ingeniería Informática (Tecnologías de la información)

Servidores Web de Altas Prestaciones

ÍNDICE

ÍNDICE	1
Crear una BD e insertar datos	2
Replicar una BD MySQL con mysqldump	4
Replicación de BD mediante una configuración maestro-esclavo	7
Configuración de la máquina líder	7
Configuración de la máquina seguidora	8
Configuración en MySQL en la máquina líder	8
Comprobación de funcionamiento	11
Bibliografía	12

Crear una BD e insertar datos

Lo primero que haremos en ésta práctica será crear e insertar datos en una base de datos MySQL en nuestra máquina 1. Para ello, en primer lugar instalamos el software de MySQL como aparece en la siguiente imagen:

```
adrianacosa@m1-adrianacosa:~$ sudo apt-get install mysql-client mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycat
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client
  mysql-client-core-8.0 mysql-client-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 26 newly installed, 0 to remove and 10 not upgraded.
Need to get 31.5 MB of archives.
After this operation, 262 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez lo tenemos instalado, ya podremos ejecutar el software. El comando que tenemos que ejecutar para abrir nuestro gestor de bases de datos es el siguiente:

```
adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql>
```

La contraseña por defecto no es ninguna, así que cuando nos pide la contraseña simplemente le damos a intro. Una vez dentro de MySQL, creamos la base de datos que se indica en el guión de la manera que sigue:

```
mysql> create database estudiante;
Query OK, 1 row affected (0.93 sec)

mysql> use estudiante;
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql> create table datos(nombre varchar(100), apellidos varchar(100), usuario varchar(100), email varchar(100));
Query OK, 0 rows affected (1.90 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_estudiante |
+-----+
| datos                 |
+-----+
1 row in set (0.00 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("Adrian", "Acosa Sanchez", "adrianacosa", "adrianacosa@correo.ugr.es");
Query OK, 1 row affected (0.48 sec)

mysql> select * from datos
-> ;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Adrian | Acosa Sanchez | adrianacosa | adrianacosa@correo.ugr.es |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Lo que hemos hecho con estas órdenes es crear una tabla en nuestra base de datos llamada "datos" que contendrá los datos de los usuarios en nuestro servidor. Con el comando *show tables* como podemos observar, podemos ver el contenido de las tablas de nuestra base de datos. Para insertar una fila de datos en una tabla, simplemente tenemos que hacer uso de la sintaxis de SQL para insertar una fila, que consiste en indicarle en la tabla donde queremos hacer la inserción y los propios datos de la fila que queremos insertar. Y como podemos ver en la última orden, al hacer un *select* de todas las columnas de la tabla *datos* nos muestra todo el contenido de dicha tabla.

Podemos ver una descripción de los tipos de datos que son cada una de las columnas de una tabla usando el comando *describe <tabla>*. En nuestro caso, vamos a verlo de la tabla *datos* como se muestra a continuación:

```
mysql> describe datos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombre     | varchar(100)  | YES  |     | NULL    |       |
| apellidos  | varchar(100)  | YES  |     | NULL    |       |
| usuario    | varchar(100)  | YES  |     | NULL    |       |
| email      | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> 
```

Donde se nos indica que todos los campos de la tabla *datos* son variables de tipo *varchar*, no tienen ningún valor por defecto y no son ningún tipo de clave.

Replicar una BD MySQL con mysqldump

En este apartado usaremos la herramienta *mysqldump* para volcar todos los datos contenidos en nuestra base de datos a un fichero, para así copiar estos en otra máquina y poder tener la misma información a la vez en dos servidores distintos.

Un ejemplo simple de uso de ésta herramienta sería el siguiente:

```
mysqldump datos -u root -p > /root/datos.sql
```

Pero hay que tener en cuenta que en un servidor puede ocurrir que no seamos nosotros los únicos que hagamos cambios en esa base de datos. Por lo que si mientras volcamos las tablas a un fichero alguien hace un cambio en éstas, tendremos una inconsistencia en los datos, cosa que no nos interesa en absoluto. Para poder evitar este tipo de situaciones, lo más adecuado es entrar en nuestra base de datos para bloquear las posibles modificaciones en las tablas por parte de cualquier otro usuario del sistema. De esta manera nos aseguramos de que los datos que copiaremos en otro servidor serán los mismos que los que había en el momento en el que hicimos el volcado en nuestro fichero.

Una vez aclarado esto, vamos a ver los pasos necesarios para llevar a cabo una operación del tipo explicado aquí en la siguiente captura de pantalla:

```

adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
[sudo] password for adrianacosa:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.01 sec)

mysql> quit
Bye
adrianacosa@m1-adrianacosa:~$ sudo mysqldump estudiante -u root -p > /tmp/estudiante.sql
Enter password:
adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
adrianacosa@m1-adrianacosa:~$

```

Llegados a este punto, ya tendríamos nuestro fichero con los datos de nuestra base de datos *estudiante* en el fichero *estudiante.sql*. Lo único que falta es copiarlo en nuestra máquina M2 como aparece en la imagen siguiente:

```

adrianacosa@m1-adrianacosa:~$ sudo scp /tmp/estudiante.sql adrianacosa@192.168.122.36:/tmp/estudiante.sql
adrianacosa@192.168.122.36's password:
estudiante.sql                                100% 2031   691.0KB/s   00:00
adrianacosa@m1-adrianacosa:~$

```

Y configurar nuestra base de datos en la máquina 2 para poder adquirir los datos recibidos desde la máquina 1 de la siguiente forma:

```

adrianacosa@m2-adrianacosa:~$ sudo mysql -u root -p
[sudo] password for adrianacosa:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> create database estudiante;
Query OK, 1 row affected (0.83 sec)

mysql> quit
Bye
adrianacosa@m2-adrianacosa:~$ sudo mysql -u root -p estudiante < /tmp/estudiante.sql
Enter password:
adrianacosa@m2-adrianacosa:~$

```

Y como podemos ver, adquirimos todos los datos que declaramos en la máquina 1:

```

adrianacosa@m2-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> use estudiante;
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_estudiante |
+-----+
| datos                 |
+-----+
1 row in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos   | usuario   | email                                     |
+-----+-----+-----+-----+
| Adrian | Acosa Sanchez | adrianacosa | adrianacosa@correo.ugr.es |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

Replicación de BD mediante una configuración líder-seguidor

Todo el proceso anterior sería totalmente válido y útil, pero es un proceso que se realiza completamente a mano por un administrador del sistema. Sin embargo, existe otra forma de hacerlo de manera automatizada por MySQL, que consiste en configurar el demonio de éste para que haga la replicación sobre una máquina seguidora a partir de los datos almacenados de la máquina líder.

Para poder llevar a cabo ésta labor, tendremos que hacer configuraciones en todos los servidores que participarán en éste proceso. Para ésta práctica veremos cómo hacerlo con un sólo servidor líder y un sólo servidor seguidor.

Empezaremos por la configuración de la máquina 1.

Configuración de la máquina líder

Empezamos la configuración editando el archivo de configuración de MySQL ubicado en la ruta `/etc/mysql/mysql.conf.d/mysqld.cnf` y tendremos que ajustar los siguientes parámetros:

- Comentamos la siguiente línea:

```
#bind-address            = 127.0.0.1
```

- Le indicamos el archivo donde deberá almacenar el log de errores para poder saber qué errores hemos cometido, por ejemplo, a la hora de ajustar el archivo de configuración:

```
log_error = /var/log/mysql/error.log
```

- Establecemos el id del servidor:

```
server-id                = 1
```

- Y por último indicamos el registro binario que contiene todo el registro de actualizaciones en un formato más eficiente y de una manera más segura para las transacciones:

```
log_bin                  = /var/log/mysql/mysql-bin.log
```

Una vez hemos indicado estas configuraciones, reiniciamos el sistema y si no nos da ningún fallo significa que hemos hecho bien todas las configuraciones:

```
adrianacosa@m1-adrianacosa:~$ sudo systemctl restart mysql.service
adrianacosa@m1-adrianacosa:~$
```

Y como vemos, todo ha salido bien ya que no nos ha saltado ningún mensaje de error.

Configuración de la máquina seguidora

En la máquina seguidora realizaremos exactamente las mismas configuraciones que acabamos de hacer en la máquina líder, con la única diferencia de que en la configuración de *server-id* pondremos 2 para diferenciarla de la máquina líder. Por lo tanto quedaría así:

```
server-id = 2
```

Y una vez tenemos las configuraciones bien hechas, reiniciamos el servicio de MySQL y podemos volver a la máquina 1 para continuar la configuración.

Configuración en MySQL en la máquina líder

Para poder crear un usuario seguidor en nuestra base de datos para realizar la replicación ejecutamos los siguientes comandos:

```
adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> CREATE USER 'seguidor_adrianacosa'@'%' IDENTIFIED BY 'seguidor_adrianacosa';
ERROR 1396 (HY000): Operation CREATE USER failed for 'seguidor_adrianacosa'@'%'
mysql> drop user seguidor_adrianacosa;
Query OK, 0 rows affected (0.44 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE USER seguidor_adrianacosa IDENTIFIED BY 'seguidor_adrianacosa';
Query OK, 0 rows affected (0.09 sec)

mysql>
```

Debido a un bug, no me dejaba crear el usuario seguidor. Por lo tanto la solución ha sido asumir que el usuario se encontraba creado y por lo tanto borrarlo, resetear los privilegios y tras eso ya me dejaba crear el usuario correctamente.

Tras esto, el siguiente paso consiste en darle los permisos necesarios para que pueda realizar la replicación de la máquina líder:

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'seguidor_adrianacosa'@'%';
Query OK, 0 rows affected (0.14 sec)

mysql> FLUSH PRIVILEGES;
FLQuery OK, 0 rows affected (0.82 sec)

mysql> FLUSH TABLES;
Query OK, 0 rows affected (1.42 sec)

mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

En estas órdenes lo que hacemos es darle privilegios de replicación de la base de datos en todas las tablas al usuario que acabamos de crear desde cualquier máquina (eso significa el %).

Para terminar con la máquina líder, comprobamos la configuración y obtenemos los datos de la base de datos que posteriormente vamos a usar en la configuración del seguidor:

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000002 |      1402 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Configuración en MySQL en la máquina seguidora

Ahora en la máquina seguidora lo que haremos será indicarle a la base de datos toda la información que hemos mostrado en la última captura para que conozca bien quién es la máquina líder:

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.122.43', MASTER_USER='seguidor_adrianacosa', MASTER_PASSWORD='seguidor_adrianacosa', MASTER_LOG_FILE='mysql-bin.000002', MASTER_LOG_POS=1402, MASTER_PORT=3306;
Query OK, 0 rows affected, 9 warnings (1.84 sec)

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (1.45 sec)

mysql>
```

Volvemos a la máquina 1 para volver a activar las tablas:

```
adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

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 statement.

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Y cuando comprobamos el valor de *"Seconds_Behind_Master"* en la máquina seguidora, nos damos cuenta de que el valor es *NULL* y por tanto hay un fallo. Viendo los mensajes de error mostrados al ver el estado del seguidor, nos damos cuenta de que nos salta un error debido a que intenta autenticar la clave del usuario recién creado mediante el uso de sha2 por un plugin que se usa por defecto en ésta versión. Para solucionarlo lo que he hecho ha sido un *ALTER USER* al usuario indicando que el método de autenticación sea el básico de mysql de la siguiente manera:

```
ALTER USER seguidor_adrianacosa IDENTIFIED WITH mysql_native_password BY
      'seguidor_adrianacosa';
```

Y tras reiniciar el servicio, comprobamos ejecutando la misma orden anterior si ha cambiado el estado de *"Seconds_Behind_Master"* de *NULL* a 0:

```
mysql> show slave status\G;
***** 1. row *****
Slave_IO_State: Waiting for source to send event
Master_Host: 192.168.122.43
Master_User: seguidor_adrianacosa
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000004
Read_Master_Log_Pos: 455
Relay_Log_File: m2-adrianacosa-relay-bin.000003
Relay_Log_Pos: 326
Relay_Master_Log_File: mysql-bin.000004
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 455
Relay_Log_Space: 725
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
```

Comprobación de funcionamiento

Para terminar de comprobar que hemos configurado todo correctamente, simplemente vamos a probar a introducir datos a la tabla de la máquina líder y ver si se replica la información a la máquina seguidora. Para ello vamos a la máquina 1 e introducimos un estudiante inventado y comprobamos en la otra máquina si se ha realizado la replicación de los datos:

```
mysql> insert into datos(nombre,apellidos,usuario,email) values ("Jose", "Rodriguez Rodriguez",
"joserr", "joserr@ugr.es");
Query OK, 1 row affected (0.87 sec)

mysql> select * from datos
-> ;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Adrian | Acosa Sanchez | adrianacosa | adrianacosa@correo.ugr.es |
| Jose | Rodriguez Rodriguez | joserr | joserr@ugr.es |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> use estudiante;
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> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Adrian | Acosa Sanchez | adrianacosa | adrianacosa@correo.ugr.es |
| Jose | Rodriguez Rodriguez | joserr | joserr@ugr.es |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Y como se puede ver a la izquierda en la máquina líder hemos introducido un alumno, y al entrar en la máquina seguidora (derecha) y comprobar los datos de la tabla datos vemos como se han actualizado con el nuevo alumno.

Reglas de IPTABLES para permitir comunicación entre las BD sólo entre M1 y M2

A la configuración del cortafuegos que hicimos en la práctica anterior, tendremos que añadir dos líneas más indicando la comunicación en el puerto 3306 y que sólo salgan comunicaciones en ese puerto si la conexión ha sido anteriormente establecida por la IP que hemos permitido. Esto que acabo de decir se traduce en lo siguiente:

```
sudo iptables -A INPUT -p tcp -s 192.168.122.103 --dport 3306 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 3306 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

Siento la IP que aparece tras la opción -s distinta en ambas máquinas, indicando la IP de la otra máquina.

También he añadido en ésta práctica dos reglas nuevas para poder conectarme por SSH únicamente desde el anfitrión de ambas máquinas virtuales de la misma manera en que he hecho las reglas anteriores:

```
iptables -A INPUT -p tcp -s 192.168.122.1 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

ACLARACIÓN: las IP cambian a partir de aquí debido a que he trabajado desde otro ordenador distinto al resto de la práctica. La máquina 1 pasa a ser **192.168.122.125** y la máquina 2 pasa a ser **192.168.122.103**.

Replicación de la BD mediante una configuración líder-líder

Lo primero que tendremos que configurar es la IP donde se van a estar escuchando las peticiones al servidor MySQL de la siguiente manera:

Para la máquina 1:

```
bind-address = 192.168.122.125
```

Y para la máquina 2:

```
bind-address = 192.168.122.103
```

A continuación vamos a crear en ambos servidores un usuario que será nuestro seguidor en la máquina contraria. Es decir, en la máquina 1 crearemos un seguidor de la máquina 2 y en la máquina 2 un seguidor de la máquina 1. Por lo tanto realizaremos los siguiente en la máquina 1:

```
adrianacosa@m1-adrianacosa:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)

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 statement.

mysql> CREATE USER 'seguidor1_adrianacosa'@'192.168.122.103' IDENTIFIED BY 'seguidor1_adrianacosa';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'seguidor1_adrianacosa'@'192.168.122.103';
Query OK, 0 rows affected (0.00 sec)

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

Y en la máquina 2:

```
adrianacosa@m2-adrianacosa:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.29-0ubuntu0.22.04.2 (Ubuntu)

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 statement.

mysql> CREATE USER 'seguidor2_adrianacosa'@'192.168.122.125' IDENTIFIED BY 'seguidor2_adrianacosa';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'seguidor2_adrianacosa'@'192.168.122.125';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Aquí lo que hemos hecho ha sido darle privilegios de replicación a ambos usuarios igual que hemos hecho en la configuración de líder-seguidor.

Una vez tenemos esto bien configurado, mostramos el estado de ambos líderes para poder configurar la replicación correctamente.

En la máquina 1:

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000006 |      891 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Y en la máquina 2:

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000006 |     1101 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Teniendo en cuenta estos datos, podemos configurar el maestro de cada una de las dos máquinas como sigue:

En la máquina 1:

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.122.103', MASTER_USER='seguidor2_adrianacosa', MASTER_PASSWORD='seguidor2_adrianacosa', MASTER_LOG_FILE='mysql-bin.000006', MASTER_LOG_POS=1101;
ERROR 3021 (HY000): This operation cannot be performed with a running slave io thread; run STOP SLAVE IO_THREAD FOR CHANNEL '' first.
mysql> STOP SLAVE IO_THREAD FOR CHANNEL '';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CHANGE MASTER TO MASTER_HOST='192.168.122.103', MASTER_USER='seguidor2_adrianacosa', MASTER_PASSWORD='seguidor2_adrianacosa', MASTER_LOG_FILE='mysql-bin.000006', MASTER_LOG_POS=1101;
Query OK, 0 rows affected, 8 warnings (0.00 sec)

mysql> show master status
-> ;
+-----+-----+-----+-----+-----+
| File           | Position | BinLog_Do_DB | BinLog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000006 |      891 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

Y en la máquina 2:

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.122.125', MASTER_USER='seguidor1_adrianacosa', MASTER_PASSWORD='seguidor1_adrianacosa', MASTER_LOG_FILE='mysql-bin.000006', MASTER_LOG_POS=891;
ERROR 3021 (HY000): This operation cannot be performed with a running slave io thread; run STOP SLAVE IO_THREAD FOR CHANNEL '' first.
mysql> STOP SLAVE IO_THREAD FOR CHANNEL '';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CHANGE MASTER TO MASTER_HOST='192.168.122.125', MASTER_USER='seguidor1_adrianacosa', MASTER_PASSWORD='seguidor1_adrianacosa', MASTER_LOG_FILE='mysql-bin.000006', MASTER_LOG_POS=891;
Query OK, 0 rows affected, 8 warnings (0.07 sec)

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> 
```

Como se puede apreciar, cometo el mismo error dos veces. Me salta un error de que no se puede cambiar el líder mientras hay un hilo de seguidor activo. Luego ejecuto el comando que me indica el error antes de ejecutar el cambio de líder y luego de ejecutar el cambio, inicio el seguidor.

Y con ésto ya estaría todo bien configurado. Para probarlo creo una BD llamada prueba que se replica directamente en el otro servidor y las pruebas que aparecen en la siguiente captura para comprobar que independientemente de dónde se haga la alteración de la base de datos se replica en el servidor contrario:

```
mysql> create database prueba;
Query OK, 1 row affected (0.03 sec)

mysql> use prueba
Database changed
mysql> CREATE TABLE tabla(id varchar(100));
Query OK, 0 rows affected (0.07 sec)

mysql> show tables;
+-----+
| Tables_in_prueba |
+-----+
| tabla             |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO tabla(id) VALUES ("1");
Query OK, 1 row affected (0.01 sec)

mysql> select * from tabla;
+----+
| id |
+----+
| 1  |
| 2  |
+----+
2 rows in set (0.00 sec)

mysql>

mysql> use prueba
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_prueba |
+-----+
| tabla             |
+-----+
1 row in set (0.01 sec)

mysql> select * from tabla;
Empty set (0.00 sec)

mysql> select * from tabla;
+----+
| id |
+----+
| 1  |
+----+
1 row in set (0.00 sec)

mysql> INSERT INTO tabla(id) VALUES ("2");
Query OK, 1 row affected (0.01 sec)

mysql>
```

Bibliografía

- 1.-<https://stackoverflow.com/questions/5555328/error-1396-hy000-operation-create-user-failed-for-jacklocalhost>
- 2.-<https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands>
- 3.-<https://unixcop.com/master-master-replication-mariadb/>