



SERVIDORES WEB DE ALTAS PRESTACIONES

PRÁCTICA 5: Replicación de bases de datos MySQL

Autor

Miguel Ángel Pérez Díaz



Escuela Técnica Superior de Ingenierías Informática y
de Telecomunicación

—
Granada, 2020

1. Crear una BD con al menos una tabla y algunos datos

Para esta primera tarea debemos crear un base de datos de MySQL con al menos una tabla y algunos datos. Esta tarea servirá de base para las siguientes, que como se puede observar se trata de tareas secuenciales por lo que debemos haber realizado la anterior antes de pasar a la siguiente.

Esta primera tarea es bastante sencilla, inicialmente debemos acceder a MySQL indicándole el usuario y la contraseña a usar cuando se conecta al servidor mediante los parámetros **-u** y **-p** respetivamente.

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

Usuario root y sin contraseña.

A continuación, crearemos la base de datos a la que llamaremos '*estudiante*' mediante el comando:

```
create database estudiante;
```

Inmediatamente después tras recibir el OK de MySQL le indicamos que vamos a trabajar con la base de datos anteriormente creada.

```
use estudiante;
```

Una vez recibida la confirmación procedemos a crear nuestra primera tabla en la base de datos indicándole el nombre de ésta (en nuestro caso se llamará 'datos') y los campos de cada registro, es decir, los atributos sobre los que guardaremos datos. En nuestro caso se va utilizar campos como nombre, apellidos, usuario y email, todos ellos con una longitud máxima de 100 caracteres.

```
create table datos(nombre varchar(100), apellidos  
    varchar(100), usuario varchar(100), email  
    varchar(100));
```

```
mysql> create database estudiante;
Query OK, 1 row affected (0.00 sec)

mysql> use estudiante;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table datos(nombre varchar(100), apellidos varchar(100), usuario varchar(100), email v
varchar(100));
Query OK, 0 rows affected (0.05 sec)
```

Vamos a comprobar que se ha creado la tabla correctamente.

```
show tables;
```

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

mysql>
```

Finalmente insertaremos datos a la tabla creada mediante la sentencia **INSERT** y trataremos de obtener todos los datos de la tabla mediante la sentencia **SELECT** para ver si se han añadido de la forma correcta.

```
insert into datos(nombre,apellidos,usuario,email)
values ("Miguel Angel", "Perez Diaz", "mapd0004",
mapd0004@correo.ugr.es);

select * from datos;
```

```
mysql> insert into datos(nombre,apellidos,usuario,email) values ("Miguel Angel", "Perez Diaz", "mapd0004", "mapd0004@correo.ugr.es");
Query OK, 1 row affected (0.04 sec)

mysql> select * from datos
-> ^C^C

^C
mysql> select * from datos;
+-----+-----+-----+-----+
| nombre      | apellidos | usuario | email                               |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

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)
```

Como se puede observar en el último paso utilizamos la sentencia **DESCRIBE DATOS** para describir la tabla de datos anteriormente creada. Y con esto tenemos ya finalmente nuestra base de datos creada con una tabla y algunos datos insertados para seguir con el resto de tareas propuestas.

2. Realizar la copia de seguridad de la BD completa usando mysqldump en la máquina principal y copiar el archivo de copia de seguridad a la máquina secundaria

En esta sección debemos realizar una copia de seguridad de la base de datos anteriormente creada y copiarla a la máquina secundaria (M2). Para ello utilizaremos el comando *mysqldump* para generar la copia de seguridad y la herramienta *scp* para realizar la copia, como ya hemos utilizado en otras prácticas.

Antes de realizar la copia de seguridad es necesario bloquear las tablas de la base de datos para evitar que se acceda para cambiar nada y así copiar la BD sin que se produzca una modificación simultáneamente y los datos de una máquina a otra no coincidan.

Para ello debemos acceder a MySQL en M1 y realizar este bloqueo con:

```
FLUSH TABLES WITH READ LOCK;
```

Tras esto ya podemos hacer el *mysqldump* para guardar los datos:

```
sudo mysqldump estudiante -u root -p >
/tmp/estudiante.sql
```

Como resultado se ha generado el fichero */tmp/estudiante.sql* el cuál vamos a copiar a M2. Después podemos volver a desbloquear las tablas de la BD, por lo que ejecutaremos *UNLOCK TABLES;* en MySQL.

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

mysql> quit
Bye
miguel444@m1:~$ sudo mysqldump estudiante -u root -p > /tmp/estudiante.sql
Enter password:
miguel444@m1:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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>
```

Una vez tenemos todo lo necesario vamos a tratar de copiar el archivo de seguridad con *scp* a M2, por lo que ejecutaremos:

```
sudo scp /tmp/estudiante.sql
miguel444@192.168.56.102:/tmp/estudiante.sql
```

Siendo *miguel444* el usuario y *192.168.56.102* la dirección IP de la máquina M2.

```
miguel444@m1:~$ sudo scp /tmp/estudiante.sql miguel444@192.168.56.102:/tmp/estudiante.sql
miguel444@192.168.56.102's password:
estudiante.sql                                100% 1994      2.7MB/s   00:00
miguel444@m1:~$ _
```

3. Restaurar dicha copia de seguridad en la segunda máquina (clonado manual de la BD), de forma que en ambas máquinas esté esa BD de forma idéntica.

Una vez realizada la tarea anterior podemos pensar que ya está todo preparado para restaurar la copia de seguridad transferida, pero no es del todo así. El fichero *.sql* copiado contiene las sentencias SQL en texto plano para restaurar los datos de la BD en otra máquina, pero no incluye la sentencia para crear la base de datos por lo que antes de restaurar la copia de seguridad debemos asegurarnos de haber realizado este paso previo.

Simplemente debemos crear la base de datos en M2, al igual que hicimos en M1, mediante la sentencia:

```
create database estudiante;
```

Una vez creada la base de datos ya podemos restaurar los datos contenidos en la BD indicándole la BD creada en el paso anterior y la copia de seguridad recibida con *scp*:

```
sudo mysql -u root -p estudiante <
/tmp/estudiante.sql
```

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

mysql> quit
Bye
miguel444@m2:~$ sudo mysql -u root -p estudiante < /tmp/estudiante.sql
Enter password:
```

Finalmente vamos a comprobar que se ha replicado correctamente la base de datos de M1 a M2. Para ello seleccionamos la base de datos con la que vamos a trabajar (*use estudiante*) y mostraremos las tablas de la base de datos (*show tables*) y los de datos de la tabla creada en la primera tarea llamada 'datos' (*select * from datos*).

```
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 |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

4. Realizar la configuración maestro-esclavo de los servidores MySQL para que la replicación de datos se realice automáticamente

Una vez tenemos todo lo necesario que hemos ido realizando en las anteriores tareas, vamos a tratar de replicar los datos de las BD pero en este caso de forma automática. En este caso vamos a realizar la configuración maestro-esclavo de servidores MySQL, siendo el maestro M1 (el que realiza cambios) y el esclavo M2 (el que replica los cambios de M1).

Para esta tarea empezamos trabajando con la máquina maestra M1 y configurar MySQL para que sea el maestro.

Inicialmente debemos modificar el fichero */etc/mysql/mysql.conf.d/mysqld.cnf* con las siguientes modificaciones:

- Comentar el parámetro *bind-address*.

```
#bind-address 127.0.0.1
```

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address            = 127.0.0.1
#
```

- Indicar el archivo donde almacenar el log de errores (en mi caso ya venía declarado).

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

- Establecer el identificador del servidor.

```
server-id = 1
```

- Indicar el registro binario que contiene toda la información que está disponible en el registro de actualizaciones (solo ha sido necesario descomentar la línea correspondiente).

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

Tras realizar las modificaciones pertinentes guardamos el documento y reiniciamos el servicio de MySQL.

```
#slow_query_log      = 1
#slow_query_log_file = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id            = 1
log_bin              = /var/log/mysql/mysql-bin.log
expire_logs_days     = 10
max_binlog_size      = 100M
#binlog_do_db        = include_database_name
#binlog_ignore_db     = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI "tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem

miguel444@m1:~$ sudo service mysql restart
miguel444@m1:~$
```

Una vez tenemos el fichero de configuración listo, necesitamos crear un usuario ‘esclavo’ para realizar la replicación. Para ello entraremos en MySQL y ejecutaremos las siguientes sentencias.

Creamos el usuario esclavo utilizando como contraseña: 'esclavo'.

```
CREATE USER esclavo IDENTIFIED BY 'esclavo';
```

Una vez creado el usuario le asignaremos permisos para que el usuario esclavo puede conectarse al maestro y recibir actualizaciones al registro binario del maestro.

```
GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%'  
IDENTIFIED BY 'esclavo';
```

Los asteriscos en este comando hacen referencia a la base de datos y la tabla (respectivamente) a la cual el nuevo usuario tendrá acceso.

Refrescamos todos los privilegios:

```
FLUSH PRIVILEGES;
```

Tras esto forzamos el cierre de todas las tablas. Esto es principalmente para asegurar que si alguien agrega una nueva tabla fuera de MySQL todos los hilos comenzarán a usar la nueva tabla.

```
FLUSH TABLES;
```

Finalmente cerramos todas las tablas abiertas y bloqueamos todas las tablas para todas las bases de datos con un bloqueo de lectura hasta que se ejecute.

```
FLUSH TABLES WITH READ LOCK;
```

Comprobamos la configuración y obtenemos los datos de la BD que vamos a replicar para después usarlos en la configuración del esclavo.

```
SHOW MASTER STATUS;
```

```
mysql> CREATE USER esclavo IDENTIFIED BY 'esclavo';  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%' IDENTIFIED BY 'esclavo';  
Query OK, 0 rows affected, 1 warning (0.02 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> FLUSH TABLES;  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> FLUSH TABLES WITH READ LOCK;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> SHOW MASTER STATUS;  
+-----+-----+-----+-----+-----+  
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |  
+-----+-----+-----+-----+-----+  
| mysql-bin.000001 | 980      |              |                  |                  |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```


Una vez finalizada la configuración de maestro en M1, vamos a pasar a configurar M2 para ser esclavo. Inicialmente se deberán realizar los mismos cambios sobre el fichero de configuración exceptuando el identificador del servidor (*server-id = 2, en este caso*).

```
#  
# Instead of skip-networking the default is now to listen only on  
# localhost which is more compatible and is not less secure.  
#bind-address          = 127.0.0.1  
#
```

```
# Both location gets rotated by the cronjob.  
# Be aware that this log type is a performance killer.  
# As of 5.1 you can enable the log at runtime!  
#general_log_file      = /var/log/mysql/mysql.log  
#general_log           = 1  
#  
# Error log - should be very few entries.  
#  
log_error = /var/log/mysql/error.log  
#  
# Here you can see queries with especially long duration  
#slow_query_log        = 1  
#slow_query_log_file   = /var/log/mysql/mysql-slow.log  
#long_query_time = 2  
#log-queries-not-using-indexes  
#  
# The following can be used as easy to replay backup logs or for replication.  
# note: if you are setting up a replication slave, see README.Debian about  
# other settings you may need to change.  
server-id             = 2  
log_bin              = /var/log/mysql/mysql-bin.log  
expire_logs_days     = 10  
max_binlog_size      = 100M  
#binlog_do_db        = include_database_name  
#binlog_ignore_db    = include_database_name  
#  
# * InnoDB  
#  
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.  
# Read the manual for more InnoDB related options. There are many!  
#  
# * Security Features  
  
miguel444@m2:~$ sudo service mysql restart  
miguel444@m2:~$
```

Finalmente, como antes en M1 reiniciamos el servicio de MySQL para guardar los cambios.

Una vez tenemos el servicio configurado, accedemos a MySQL y le indicamos los datos del maestro que previamente hemos obtenido tras realizar *SHOW MASTER STATUS*; en M1. Para ello utilizamos la siguiente sentencia:

```
CHANGE MASTER TO MASTER_HOST='192.168.56.12',  
MASTER_USER='esclavo', MASTER_PASSWORD='esclavo',  
MASTER_LOG_FILE='mysql-bin.000001',  
MASTER_LOG_POS=980, MASTER_PORT=3306;
```

Siendo *192.168.56.12* la dirección IP del maestro (M1) y el resto de datos: usuario y contraseña del usuario esclavo creado anteriormente y el resto obtenidos de la sentencia anterior en M1, además del puerto de escucha de MySQL = 3306.

Una vez recibida la confirmación, arrancamos el esclavo y ya estaría todo listo para que los datos se repliquen del maestro (M1) al esclavo (M2).

```
START SLAVE;
```

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.12', MASTER_USER='esclavo', MASTER_PASSWORD='esclavo', MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=980, MASTER_PORT=3306;
Query OK, 0 rows affected, 2 warnings (0.02 sec)

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

Como última tarea antes de probar el funcionamiento de la configuración maestro-esclavo, vamos a desbloquear las tablas que previamente habíamos bloqueado en M1.

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

Y comprobamos que la configuración se ha realizado correctamente en M2, para ello accedemos a MySQL y ejecutamos la sentencia *SHOW SLAVE STATUS\G*. Si la variable “*Seconds_Behind_Master*” no es nula y no aparece ningún error, todo estará funcionando correctamente.

```

      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
      Master_SSL_Verify_Server_Cert: No
      Last_IO_Errno: 0
      Last_IO_Error:
      Last_SQL_Errno: 0
      Last_SQL_Error:
      Replicate_Ignore_Server_Ids:
      Master_Server_Id: 1
      Master_UUID: 9a5e16cc-62f2-11ea-a4a9-0800277ad080
      Master_Info_File: /var/lib/mysql/master.info
      SQL_Delay: 0
      SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
      Master_Retry_Count: 86400
      Master_Bind:
      Last_IO_Error_Timestamp:
      Last_SQL_Error_Timestamp:
      Master_SSL_Crl:
      Master_SSL_Crlpath:
      Retrieved_Gtid_Set:
      Executed_Gtid_Set:
      Auto_Position: 0
      Replicate_Rewrite_DB:
      Channel_Name:
      Master_TLS_Version:
1 row in set (0.00 sec)

mysql>
```

Parece que no se obtiene ningún error y todo está funcionando correctamente, por último, ya solo nos quedaría realizar pruebas para observar si de verdad funciona. Para ello vamos a añadir datos a M1 y comprobar si estos datos se replican automáticamente en M2.

Añadimos nuevos datos a M1:

```
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> insert into datos(nombre,apellidos,usuario,email) values ("Juan", "Garcia Arroyo", "juan5", "
juan95@hotmail.com");
Query OK, 1 row affected (0.05 sec)
```

Comprobamos M2, para ello seleccionamos la tabla '*estudiante*' y mostraremos todos los datos de la tabla en la base de datos.

```
miguel444@m2:~$ sudo mysql -u root -p
[sudo] password for miguel444:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.29-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
| Juan | Garcia Arroyo | juan5 | juan95@hotmail.com |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Se puede observar que nuestra configuración maestro-esclavo funciona correctamente y obtenemos los datos replicados en M2 automáticamente.



Como tarea opcional se ha propuesto en esta práctica conseguir una configuración maestro – maestro entre nuestras dos máquinas, es decir, que ahora cualquier cambio realizado en M1 se replique en M2 y viceversa, y tengamos ambas máquinas con las BD sincronizadas.

Para ello debemos realizar los pasos en M1 que anteriormente hemos hecho en M2 y viceversa. No es necesario volver a modificar el fichero de configuración solo la parte de creación de usuario esclavo y asignación de los datos del maestro.

Empezaremos creando de nuevo el usuario esclavo, en este caso en M2 utilizando las sentencias anteriormente mencionadas:

- o CREATE USER esclavo IDENTIFIED BY 'esclavo';
- o GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%' IDENTIFIED BY 'esclavo';
- o FLUSH PRIVILEGES;
- o FLUSH TABLES;
- o FLUSH TABLES WITH READ LOCK;
- o SHOW MASTER STATUS

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.29-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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 esclavo IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%' IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected, 1 warning (0.02 sec)

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

mysql> FLUSH TABLES;
Query OK, 0 rows affected (0.03 sec)

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

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000002 |      980 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

A continuación, asignamos los datos del maestro (recibidos anteriormente) al esclavo, en este caso M1.

```
CHANGE MASTER TO MASTER_HOST='192.168.56.102',
MASTER_USER='esclavo', MASTER_PASSWORD='esclavo',
MASTER_LOG_FILE='mysql-bin.000002',
MASTER_LOG_POS=980, MASTER_PORT=3306;
```

Hemos cambiado la dirección indicándole la dirección IP de M2 y el nuevo *log_file* generado en M2.

Iniciamos el esclavo en M1 con: `START SLAVE;`

```
miguel444@m1:~$ sudo mysql -u root -p
[sudo] password for miguel444:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.29-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> CHANGE MASTER TO MASTER_HOST='192.168.56.102', MASTER_USER='esclavo', MASTER_PASSWORD='esclavo', MASTER_LOG_FILE='mysql-bin.000002', MASTER_LOG_POS=980, MASTER_PORT=3306;
Query OK, 0 rows affected, 2 warnings (0.07 sec)

mysql> START SLAVE;
Query OK, 0 rows affected (0.01 sec)

mysql> _
```

Finalmente desbloqueemos las tablas en M2 con: `UNLOCK TABLES;`

Y comprobamos que la configuración se ha realizado correctamente en M1 con: `SHOW SLAVE STATUS\G`

```
      Last_IO_Error:
      Last_SQL_Errno: 0
      Last_SQL_Error:
Replicate_Ignore_Server_Ids:
      Master_Server_Id: 2
      Master_UUID: cf540d4b-62f2-11ea-911c-080027847c72
      Master_Info_File: /var/lib/mysql/master.info
      SQL_Delay: 0
      SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
      Master_Retry_Count: 86400
      Master_Bind:
      Last_IO_Error_Timestamp:
      Last_SQL_Error_Timestamp:
      Master_SSL_Crl:
      Master_SSL_Crlpath:
      Retrieved_Gtid_Set:
      Executed_Gtid_Set:
      Auto_Position: 0
      Replicate_Rewrite_DB:
      Channel_Name:
      Master_TLS_Version:
1 row in set (0.00 sec)

mysql>
```

Parece que todo funciona correctamente, solo falta ponerlo en práctica.
Para ello añadiremos datos en M2 y comprobaremos que se añaden en M1.
Para inmediatamente después añadir datos en M1 y comprobar que de la misma forma se añaden en M2, por lo que ya tendríamos ambas máquinas sincronizadas.

MÁQUINA M2:

Mostramos los datos iniciales, añadimos nuevos datos, mostramos los nuevos datos añadidos y finalmente recibimos los datos añadidos por M1.

```
mysql> use estudiante;
Database changed
mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
| Juan | Garcia Arroyo | juan5 | juan95@hotmail.com |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("Pedro", "Gonzalez Soto", "pedrete", "pedrosoto@gmail.com");
Query OK, 1 row affected (0.07 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
| Juan | Garcia Arroyo | juan5 | juan95@hotmail.com |
| Pedro | Gonzalez Soto | pedrete | pedrosoto@gmail.com |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
| Juan | Garcia Arroyo | juan5 | juan95@hotmail.com |
| Pedro | Gonzalez Soto | pedrete | pedrosoto@gmail.com |
| Jose Manuel | Soto Hidalgo | jmsoto | jmsoto@ugr.es |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

MÁQUINA M1:

Tras realizar la configuración anteriormente comentada, mostramos los datos de la BD con los datos añadidos por M2 y añadimos nuevos datos (que se pueden apreciar en la foto anterior).

```
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0.00 sec)

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 |
+-----+-----+-----+-----+
| Miguel Angel | Perez Diaz | mapd0004 | mapd0004@correo.ugr.es |
| Juan | Garcia Arroyo | Juan5 | Juan95@hotmail.com |
| Pedro | Gonzalez Soto | pedrete | pedrosoto@gmail.com |
+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> create table datos(nombre,apellidos,usuario,email) values ("Jose Manuel", "Soto Hidalgo", "jmsoto", "jmsoto@ugr.es");
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'apellidos,usuario,email) values ("Jose Manuel', "Soto Hidalgo', "jmsoto', "jmsot' at line 1
mysql> insert into datos(nombre,apellidos,usuario,email) values ("Jose Manuel", "Soto Hidalgo", "jmsoto", "jmsoto@ugr.es");
Query OK, 1 row affected (0.01 sec)

mysql> _
```

Se puede apreciar que me he equivocado al principio a la hora de insertar datos, pero he preferido dejarlo en la captura para que se observe que el proceso ha sido seguido y no he podido añadir los datos a mano.

Se puede observar que todo funciona correctamente, por lo que nuestra configuración maestro – maestro estaría terminada y completamente funcional.



<https://dev.mysql.com/doc/internals/en/flush-tables.html>