

Práctica 5:

Replicación de bases de datos MySQL

Índice:

- 1) Introducción*
- 2) Crear BD e insertar datos por línea de comandos*
- 3) Replicar una BD MySQL con mysqldump*
- 4) Replicación de BD mediante una configuración maestro-esclavo*
- 5) Replicación de BD mediante una configuración maestro-maestro*

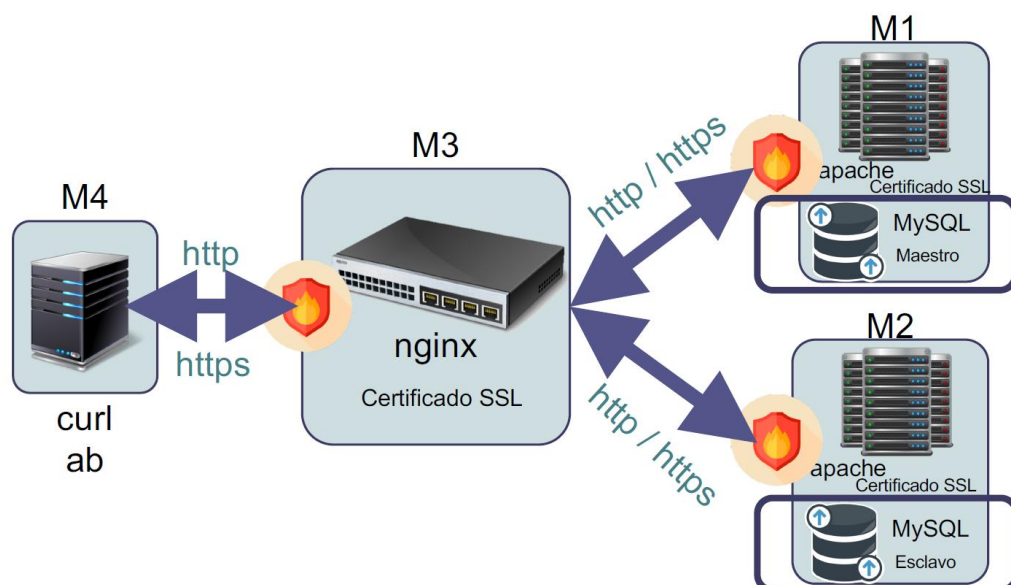
1) Introducción

A la hora de hacer copias de seguridad de nuestras bases de datos (BD) MySQL, una opción muy común suele ser la de usar una réplica maestro-esclavo, de manera que nuestro servidor en producción hace de maestro y otro servidor de backup hace de esclavo.

Tener una réplica en otro servidor también añade fiabilidad ante fallos totales del sistema en producción, los cuales, tarde o temprano, ocurrirán. Por ejemplo, podemos tener un pequeño servidor actuando como backup en nuestra oficina sincronizado mediante réplicas con nuestro sistema en producción.

Esta opción, además, también añade disponibilidad ante posibles interrupciones de servicio permanentes del servidor maestro por cualquier escenario catastrófico que nos podamos imaginar. En ese caso, tendremos posiblemente decenas de clientes y servicios parados sin posibilidad de recuperar sus datos si no hemos preparado un buen plan de contingencias. Tener un servidor de backup con MySQL actuando como esclavo de replicación es una solución asequible y no consume demasiado ancho de banda en un sitio web de tráfico normal, además de que no afecta al rendimiento del maestro en el sistema en producción.

Dado que es muy importante contar con un respaldo de los datos, en esa práctica seguiremos el siguiente esquema:



Puesto que las herramientas *tar* y *scp* que vimos en prácticas anteriores no son muy útiles para sincronizar grandes cantidades de datos, en el caso de BD, veremos formas mejores de hacerlo.

2) Crear BD e insertar datos por línea de comandos

Para el resto de la práctica debemos crearnos una BD en MySQL e insertar algunos datos. Así tendremos datos ejemplo con los cuales hacer las copias de seguridad.

Esto lo podemos llevar a cabo a través de la interfaz de línea de comandos de MySQL, conectando con el servidor a través del siguiente comando:

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

O bien con el siguiente script *createxampleDB.sql*:

```
/*Eliminamos la BD si existe*/
drop database estudiante;
/*Crear BD*/
create database estudiante;

/*Seleccionar base de datos estudiante*/
use estudiante;

/*Crear tabla datos*/
create table datos(nombre varchar(100),apellidos varchar(100),usuario varchar(100), email varchar(100));

/*Mostrar tabla*/
show tables;

/*Insertar tupla con mis datos*/
insert into datos(nombre,apellidos,usuario,email) values ("Jose Luis","Pedraza Roman","josele","joseperoman@correo.ugr.es");

/*Mostrar todos los datos de la tabla*/
select * from datos;

/*Para ver los tipos de datos de la base de datos*/
describe datos;
```

Ejecutamos el script en la máquina m1 y comprobamos que funciona correctamente como mostramos a continuación:

```
joselepedraza@m1:~$ ls
createxampleDB.sql  firewallresetconf.sh  iptablesfirewallconf.sh  tar.tgz  var
joselepedraza@m1:~$ sudo mysql -u root -p < createxampleDB.sql
Enter password:
Tables_in_estudiante
datos
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
+-----+-----+-----+-----+
Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+
nombre | varchar(100) | YES | | | |
apellidos | varchar(100) | YES | | | |
usuario | varchar(100) | YES | | | |
email | varchar(100) | YES | | | |
+-----+-----+-----+-----+
joselepedraza@m1:~$
```

3) Replicar una BD MySQL con mysqldump

MySQL ofrece una herramienta para clonar las BD que tenemos en nuestra máquina, llamada *mysqldump*.

mysqldump es parte de los programas de cliente de MySQL, que puede ser utilizado para generar copias de seguridad de la BD. Puede utilizarse para volcar una o varias BD para copia de seguridad o para transferir datos a otro servidor SQL (no necesariamente MySQL). El volcado contiene comandos SQL para crear la BD, las tablas y rellenarlas.

En el siguiente enlace se explican con detalle todas las opciones posibles: <https://dev.mysql.com/doc/refman/5.0/es/mysqldump.html> . Concretamente, las opciones *--quick* o *--opt* hacen que MySQL cargue el resultado entero en memoria antes de volcarlo al fichero, lo que puede ser un problema si se trata de una BD grande. Sin embargo, la opción *--opt* está activada por defecto.

La sintaxis básica de uso de *mysqldump* es:

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

Esto puede ser suficiente, pero tenemos que tener en cuenta que los datos pueden estar actualizándose constantemente en el servidor de BD principal. En este caso, antes de hacer la copia de seguridad del archivo *.sql* debemos evitar que se acceda a la BD para cambiar nada.

Por lo que, en el servidor de BD principal (en nuestro caso la máquina m1) hacemos lo siguiente antes de replicar la BD:

```
sudo mysql -u root -p  
  
FLUSH TABLES WITH READ LOCK;  
  
quit;
```

Es en este momento es cuando podemos hacer uso de *mysqldump* para guardar los datos (en m1) a replicar posteriormente en m2. Entonces en el servidor principal hacemos:

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

Como habíamos bloqueado las tablas, debemos desbloquearlas (quitar el *LOCK*):

```
sudo mysql -u root -p  
  
UNLOCK TABLES;  
  
quit;
```

```

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 (0.00 sec)

mysql> quit
Bye
joselepedraza@m1:~$ sudo mysqldump estudiante -u root -p > /tmp/estudiante.sql
Enter password:
joselepedraza@m1:~$ ls /tmp/
estudiante.sql
systemd-private-d2528e74e9364e01939f1f4b7342b026-apache2.service-17VKYj
systemd-private-d2528e74e9364e01939f1f4b7342b026-systemd-resolved.service-7NdfjL
systemd-private-d2528e74e9364e01939f1f4b7342b026-systemd-timesyncd.service-V29qsA
joselepedraza@m1:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
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> quit
Bye

```

Una vez hecho esto, ya podemos ir a la máquina m2 (configurada como secundaria) para copiar el archivo `.sql` con todos los datos salvados desde la máquina principal m1 (previamente cortamos los servicios de firewall tanto en m1 como en m2 para evitar posibles problemas de comunicación en los próximos puntos):

```
sudo scp /tmp/estudiante.sql usuario@ipm2:/tmp/estudiante.sql
```

```
joselepedraza@m2:~$ ls
firewallresetconf.sh iptablesfirewallconf.sh
joselepedraza@m2:~$ sudo ./firewallresetconf.sh
[sudo] password for joselepedraza:
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
joselepedraza@m2:~$ ls /tmp/
estudiante.sql
systemd-private-65a06b468818499a86496cd4c31d5eee-apache2.service-cFRh7A
systemd-private-65a06b468818499a86496cd4c31d5eee-systemd-resolved.service-1rRC20
systemd-private-65a06b468818499a86496cd4c31d5eee-systemd-timesyncd.service-hrsW00
joselepedraza@m2:~$
```

m1 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
joselepedraza@m1:~$ ls
createxampDB.sql firewallresetconf.sh iptablesfirewallconf.sh tar.tgz var
joselepedraza@m1:~$ sudo ./firewallresetconf.sh
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source                   destination
joselepedraza@m1:~$ sudo scp /tmp/estudiante.sql joselepedraza@192.168.56.102:/tmp/estudiante.sql
joselepedraza@192.168.56.102's password:
estudiante.sql                                100% 1995      1.2MB/s   00:00
joselepedraza@m1:~$
```

Es importante destacar que el archivo `.sql` de copia de seguridad tiene formato de texto plano, e incluye las sentencias SQL para restaurar los datos contenidos en la BD en otra máquina. Sin embargo, la orden `mysqldump` no incluye en ese archivo la sentencia para crear la BD (por lo que es necesario que nosotros la creamos en la máquina secundaria m2 en un primer paso, antes de restaurar las tablas de esa BD y los datos contenidos en éstas).

Con el archivo de copia de seguridad en el esclavo ya podemos importar la BD completa en el MySQL. Para ello, en un primer paso creamos la BD en m2:

```
joselepedraza@m2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
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> create database estudiante;
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
```

Y en un segundo paso restauramos los datos contenidos en la BD (creándose así las tablas en el proceso), entramos en MySQL y le indicamos que se use la BD

que acabamos de insertar (también mostramos los datos para comprobar que se ha insertado correctamente):

```
joselepedraza@m2:~$ sudo mysql -u root -p estudiante < /tmp/estudiante.sql
Enter password:
joselepedraza@m2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
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> 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> 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 (0.00 sec)

mysql> quit
Bye
joselepedraza@m2:~$
```

Por supuesto, también podemos hacer la orden de restaurar la BD directamente usando un “pipe” a un *ssh* para exportar los datos al mismo tiempo (siempre y cuando en la máquina m2 ya hubiésemos creado la BD), con el siguiente comando:

mysqldump estudiante -u root -p | ssh ipm2 mysql

4) Replicación de BD mediante una configuración maestro-esclavo

La opción anterior funciona perfectamente, pero es algo que realiza un operador a mano. Sin embargo, MySQL tiene la opción de configurar el demonio para hacer la replicación de las BD sobre un esclavo a partir de los datos que almacena el maestro.

Se trata de un proceso automático que resulta muy adecuado en un entorno de producción real. Implica realizar algunas configuraciones tanto en el servidor principal m1 como en el secundario m2.

Antes de empezar vamos a reconfigurar los cortafuegos de ambas máquinas para la apertura del puerto 3306 usado por MySQL y la comunicación entre ellas a través del mismo.

Es una buena práctica hacer la configuración maestro-esclavo (o cualquiera) con los cortafuegos desactivados para una vez que la tengamos operativa, los levantamos y comprobamos que sigue funcionando correctamente la configuración hecha, así nos curamos en salud por si algo falla en la configuración y nos aseguramos que no es problema del cortafuegos.

Cortafuegos máquina maestro m1:

```
# Eliminar todas las reglas (configuracion limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# Politica por defecto: denegar todo el trafico
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitir cualquier acceso desde localhost (interface lo)
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Abrir el puerto 3306 para permitir comunicaciones MySQL:
iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 3306 -j ACCEPT

# Abrir el puerto 22 para permitir el acceso por SSH:
iptables -A INPUT -p tcp --dport 22 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# Permitir el trafico por el puerto 80 (HTTP):
iptables -A INPUT -p tcp --dport 80 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT

# Permitir el tráfico por el puerto 443 (HTTPS):
iptables -A INPUT -p tcp --dport 443 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT

iptables -L -n -v
```


Cortafuegos máquina esclavo m2:

```
# Eliminar todas las reglas (configuracion limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# Política por defecto: denegar todo el trafico
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Permitir cualquier acceso desde localhost (interface lo)
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Abrir el puerto 3306 para permitir comunicaciones MySQL:
iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 3306 -j ACCEPT

# Abrir el puerto 22 para permitir el acceso por SSH:
iptables -A INPUT -p tcp --dport 22 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# Permitir el trafico por el puerto 80 (HTTP):
iptables -A INPUT -p tcp --dport 80 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT

# Permitir el tráfico por el puerto 443 (HTTPS):
iptables -A INPUT -p tcp --dport 443 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT

iptables -L -n -v
```

Hemos añadido la siguiente línea:

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Y modificado la línea:

```
iptables -P OUTPUT ACCEPT
```

Aceptando así todo el tráfico saliente de la máquina esclavo m2. Esto es algo usual, teniendo en cuenta que la idea de seguridad sobre una máquina es principalmente sobre el tráfico entrante ya que este tráfico proviene de máquinas externas que seguramente no sean conocidas, pero para el tráfico saliente, que al ser cosa de la propia máquina no tendremos problemas de seguridad con el tráfico que sale de ésta ya que si conoce hacia donde va.

A continuación, se detalla el proceso a realizar en ambas máquinas, para lo cual, supondremos que partimos teniendo clonadas las bases de datos en ambas máquinas.

Lo primero que debemos hacer es la configuración de MySQL del maestro. Entonces en la máquina m1:

1º) Editamos con *vim* como root el */etc/mysql/mysql.conf.d/mysqld.cnf* para realizar las siguientes configuraciones (previa copia de seguridad del archivo *.old* por si cometemos algún fallo y queremos volver a la configuración por defecto) :

```
joselepedraza@m1:~$ sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/mysql.conf.d/mysqld.cnf.old
[sudo] password for joselepedraza:
joselepedraza@m1:~$ ls /etc/mysql/mysql.conf.d/
mysqld.cnf  mysqld.cnf.old  mysqld_safe_syslog.cnf
joselepedraza@m1:~$
```

- 1.1) Comentamos el parámetro *bind-address* que sirve para que escuche a un servidor:

#bind-address 127.0.0.1

```
skip-external-locking
#
# 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
#
```

- 1.2) Le indicamos el archivo donde almacenará el *log* de errores. De esta forma, si por ejemplo al reiniciar el servicio cometemos algún error en el archivo de configuración, en el archivo de log nos mostrará con detalle lo sucedido:

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

- 1.3) Establecemos el identificador del servidor:

server-id=1

- 1.4) El registro binario contiene toda la información que está disponible en el registro de actualizaciones, en un formato más eficiente y de una manera que es segura para las transacciones:

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

```
# 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           = 1
log_bin             = /var/log/mysql/bin.log
```

2º) Reiniciamos el servicio y comprobamos que no hay errores:

```
sudo service mysql restart
```

```
sudo service mysql status
```

```
joselepedraza@m1:~$ sudo service mysql restart
joselepedraza@m1:~$ sudo service mysql status
• mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-05-07 13:15:11 UTC; 7s ago
     Process: 1947 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid (code=exite
     Process: 1928 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS
   Main PID: 1949 (mysqld)
     Tasks: 27 (limit: 1108)
   CGroup: /system.slice/mysql.service
           └─1949 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

may 07 13:15:11 m1 systemd[1]: Stopped MySQL Community Server.
may 07 13:15:11 m1 systemd[1]: Starting MySQL Community Server...
may 07 13:15:11 m1 systemd[1]: Started MySQL Community Server.
lines 1-13/13 (END)
```

Dado que todo ha funcionado correctamente hasta el momento en la configuración del maestro, podemos pasar a hacer la configuración de MySQL del esclavo, en este caso nuestra máquina m2:

3º) Editamos como root su archivo de configuración (previa copia *.old* como hemos hecho anteriormente en m1):

```
joselepedraza@m2:~$ sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/mysql.conf.d/mysqld.cnf.ol
d
[sudo] password for joselepedraza:
joselepedraza@m2:~$ ls /etc/mysql/mysql.conf.d/
mysqld.cnf  mysqld.cnf.old  mysqld_safe_syslog.cnf
joselepedraza@m2:~$
```

3.1) Comentamos el parámetro *bind-address* que sirve para que escuche a un servidor (como hicimos en m1):

```
#bind-address 127.0.0.1
```

3.2) Le indicamos el archivo donde almacenará el *log* de errores. De esta forma, si por ejemplo al reiniciar el servicio cometemos algún error en el archivo de configuración, en el archivo de log nos mostrará con detalle lo sucedido (como hicimos en m1):

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

3.3) Establecemos el identificador del servidor:

```
server-id=2
```

```
# 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/bin.log
```

3.4) El registro binario contiene toda la información que está disponible en el registro de actualizaciones, en un formato más eficiente y de una manera que es segura para las transacciones (como hicimos en m1):

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

4º) Reiniciamos el servicio y comprobamos que no hay errores:

sudo service mysql restart

sudo service mysql status

```
joselepedraza@m2:~$ sudo service mysql restart
joselepedraza@m2:~$ sudo service mysql status
• mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-05-07 13:33:33 UTC; 6s ago
     Process: 1802 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid (code=exited, status=0/SUCCESS)
     Process: 1783 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 1804 (mysqld)
      Tasks: 27 (limit: 1108)
   CGroup: /system.slice/mysql.service
           └─1804 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

may 07 13:33:33 m2 systemd[1]: Stopped MySQL Community Server.
may 07 13:33:33 m2 systemd[1]: Starting MySQL Community Server...
may 07 13:33:33 m2 systemd[1]: Started MySQL Community Server.
lines 1-13/13 (END)
```

Dado que todo ha funcionado correctamente hasta el momento en la configuración del esclavo, podemos volver al maestro para crear un usuario y darle permisos de acceso para la replicación.

Por tanto, en la máquina maestro m1 (configuración de MySQL del maestro):

5º) Entramos en MySQL y ejecutamos las siguientes sentencias para crear un usuario “esclavo” donde realizar la replicación, con los permisos necesarios:

sudo mysql -u root -p

CREATE USER esclavo IDENTIFIED BY 'esclavo';

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

FLUSH PRIVILEGES;

FLUSH TABLES;

FLUSH TABLES WITH READ LOCK;

6º) Después de la creación de usuario 'esclavo' en m1, mostramos la configuración del maestro:

SHOW MASTER STATUS;

```
mysql> CREATE USER esclavo IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo'@'%' IDENTIFIED BY 'esclavo';
Query OK, 0 rows affected, 1 warning (0.00 sec)

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

mysql> FLUSH TABLES;
Query OK, 0 rows affected (0.00 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 |
+-----+-----+-----+-----+-----+
| bin.000001    |      980 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

Comprobada la configuración del maestro, obtenemos los datos de la BD que vamos a replicar para posteriormente usarlos en la configuración del esclavo.

En la máquina m2 (configuración de MySQL del esclavo):

7º) Volvemos a la máquina esclava, entramos en MySQL y le damos los datos del maestro. Ejecutamos la siguiente sentencia (cuidado con la IP del maestro, con el valor de *master_log_file* y del *master_log_pos* que los obtuvimos del maestro con el comando *show master status* en la captura anterior):

sudo mysql -u root -p

```
CHANGE MASTER TO MASTER_HOST='dir_IP_maestro_m1',
MASTER_USER='esclavo', MASTER_PASSWORD='esclavo',
MASTER_LOG_FILE='bin.000001', MASTER_LOG_POS=980,
MASTER_PORT=3306;
```

8º) Por último, arrancamos el esclavo y ya está todo listo para que los demonios de MySQL de las dos máquinas repliquen automáticamente los datos que se introduzcan/modifiquen/borren en el servidor maestro:

START SLAVE;

```
joselepedraza@m2:~$ sudo mysql -u root -p
[sudo] password for joselepedraza:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
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='192.168.56.101',MASTER_USER='esclavo',MASTER_PASSWORD='esclavo',
MASTER_LOG_FILE='bin.000001',MASTER_LOG_POS=980,MASTER_PORT=3306;
Query OK, 0 rows affected, 2 warnings (0.05 sec)

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

mysql> _
```

9º) Antes de comenzar a probar, debemos activar las tablas en el maestro m1 con el siguiente comando como vemos a continuación:

UNLOCK TABLES;

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

mysql> _
```

10º) Para asegurarnos de que todo funciona perfectamente y que el esclavo no tiene ningún problema para replicar la información, nos vamos a la máquina esclavo m2 y ejecutamos la siguiente orden (recordar levantar los cortafuegos de ambas máquinas antes de hacer la prueba siguiente y después de haber comprobado que funciona correctamente la replicación sin los cortafuegos activos):

SHOW SLAVE STATUS\G

```
mysql> SHOW SLAVE STATUS\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 192.168.56.101
        Master_User: esclavo
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: bin.000001
      Read_Master_Log_Pos: 980
        Relay_Log_File: m2-relay-bin.000002
        Relay_Log_Pos: 314
      Relay_Master_Log_File: bin.000001
      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: 980
      Relay_Log_Space: 518
      Until_Condition: None
```

```
      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: f4941f51-5f10-11ea-acc4-080027b85ddd
        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> _
```

Una vez hecho esto, agregamos una tupla a la base de datos desde la máquina maestro 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 ("Paco","Pepe","pape","paco pepe@correo.es");
Query OK, 1 row affected (0.00 sec)

mysql> _
```

Y comprobamos de que se replica en la máquina esclavo m2 correctamente:

```
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 |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
| Paco | Pepe | pape | paco pepe@correo.es |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Como vemos se ha actualizado correctamente la base de datos *estudiante* en m2 después de la modificación en el maestro m1.

5) Replicación de BD mediante una configuración maestro-maestro

Después de configurar una arquitectura maestro-esclavo para que nuestro servidor esclavo m2 sea backup del maestro m1, ahora vamos a configurar para que cada uno de los servidores sea backup del otro, es decir, si realizamos una modificación en m1 se replicará en m2 y viceversa.

Una vez más tendremos que reconfigurar los cortafuegos de ambas máquinas como sigue (aunque los mantendremos desactivados hasta que realicemos y probemos la nueva configuración maestro-maestro para evitar problemas mayores).

En este caso, tanto en m1 como en m2 tendremos la siguiente configuración de los cortafuegos:

```
# Eliminar todas las reglas (configuracion limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# Política por defecto: denegar todo el trafico
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Permitir cualquier acceso desde localhost (interface lo)
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Abrir el puerto 3306 para permitir comunicaciones MySQL:
iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 3306 -j ACCEPT

# Abrir el puerto 22 para permitir el acceso por SSH:
iptables -A INPUT -p tcp --dport 22 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# Permitir el trafico por el puerto 80 (HTTP):
iptables -A INPUT -p tcp --dport 80 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT

# Permitir el tráfico por el puerto 443 (HTTPS):
iptables -A INPUT -p tcp --dport 443 -j ACCEPT --source 192.168.56.103
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT

iptables -L -n -v
```

Para la configuración maestro-maestro, debemos realizar los siguientes pasos:

1º) En la máquina “esclava” m2, debemos crear un usuario *maestro* (al igual que en la máquina “maestro” teníamos un usuario “esclavo”). Entonces, procedemos como en el punto 5º del apartado anterior con esta nueva configuración de m2:

```
joselepedraza@m2:~$ 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: 5.7.30-0ubuntu0.18.04.1-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 maestro IDENTIFIED BY 'maestro';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'maestro'@'%' IDENTIFIED BY 'maestro';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| bin.000005    |      688 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

2º) Una vez hecho esto debemos cambiar la configuración del maestro en m1 de la siguiente forma:

```
joselepedraza@m1:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
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='maestro', MASTER_PASSWORD='maestr
o', MASTER_LOG_FILE='bin.000005', MASTER_LOG_POS=688, MASTER_PORT=3306;
Query OK, 0 rows affected, 2 warnings (0.05 sec)
```

3º) Reiniciamos el servicio MySQL en ambas máquinas como sigue:

```
joselepedraza@m1:~$ sudo service mysql restart
joselepedraza@m1:~$ sudo service mysql status
• mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-05-08 12:37:53 UTC; 29s ago
  Process: 14587 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid (code=exit
  Process: 14565 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
  Main PID: 14589 (mysqld)
  Tasks: 30 (limit: 1108)
  CGroup: /system.slice/mysql.service
          └─14589 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

may 08 12:37:52 m1 systemd[1]: Stopped MySQL Community Server.
may 08 12:37:52 m1 systemd[1]: Starting MySQL Community Server...
may 08 12:37:53 m1 systemd[1]: Started MySQL Community Server.
joselepedraza@m1:~$ _
```

4º) Comprobamos el funcionamiento, añadiendo nuevas tuplas desde ambas máquinas y comprobando que se replican las BD modificadas en la otra máquina:

```
Database changed
mysql> select * from datos;
```

nombre	apellidos	usuario	email
Jose Luis	Pedraza Roman	josele	joseperoman@correo.ugr.es
Paco	Pepe	pape	pacopepe@correo.es
esclavo	clavo	eslave	esclavo@eslave.es

```
3 rows in set (0.00 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("insert","desdem1","insertdem1","inserm1@prueba.es");
Query OK, 1 row affected (0.01 sec)

mysql> select * from datos;
```

nombre	apellidos	usuario	email
Jose Luis	Pedraza Roman	josele	joseperoman@correo.ugr.es
Paco	Pepe	pape	pacopepe@correo.es
esclavo	clavo	eslave	esclavo@eslave.es
insert	desdem1	insertdem1	inserm1@prueba.es
insert	desdem2	insertdem2	inserm2@prueba.es

```
5 rows in set (0.00 sec)
```

```
mysql> select * from datos;
```

nombre	apellidos	usuario	email
Jose Luis	Pedraza Roman	josele	joseperoman@correo.ugr.es
Paco	Pepe	pape	pacopepe@correo.es
esclavo	clavo	eslave	esclavo@eslave.es
insert	desdem1	insertdem1	inserm1@prueba.es
insert	desdem2	insertdem2	inserm2@prueba.es

```
4 rows in set (0.00 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("insert","desdem2","insertdem2","inserm2@prueba.es");
Query OK, 1 row affected (0.01 sec)

mysql> select * from datos;
```

nombre	apellidos	usuario	email
Jose Luis	Pedraza Roman	josele	joseperoman@correo.ugr.es
Paco	Pepe	pape	pacopepe@correo.es
esclavo	clavo	eslave	esclavo@eslave.es
insert	desdem1	insertdem1	inserm1@prueba.es
insert	desdem2	insertdem2	inserm2@prueba.es

```
5 rows in set (0.00 sec)
```

5º) Por último solo nos queda levantar los cortafuegos anteriormente configurados en ambas máquinas y volver a probar que funciona correctamente:

```
m1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

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 |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
| Paco | Pepe | pape | pacopepe@correo.es |
| esclavo | clavo | esclave | esclavo@eslave.es |
| insert | desdem1 | insertdem1 | insertm1@prueba.es |
| insert | desdem2 | insertdem2 | insertm2@prueba.es |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("cortafuegosm1","cortafuegosm1","cortafuegosm1","insertm1@prueba.es");
Query OK, 1 row affected (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
| Paco | Pepe | pape | pacopepe@correo.es |
| esclavo | clavo | esclave | esclavo@eslave.es |
| insert | desdem1 | insertdem1 | insertm1@prueba.es |
| insert | desdem2 | insertdem2 | insertm2@prueba.es |
| cortafuegosm1 | cortafuegosm1 | cortafuegosm1 | insertm1@prueba.es |
| cortafuegos | cortafuegosm2 | cortafuegosm2 | insertm2@prueba.es |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
m2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

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 |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
| Paco | Pepe | pape | pacopepe@correo.es |
| esclavo | clavo | esclave | esclavo@eslave.es |
| insert | desdem1 | insertdem1 | insertm1@prueba.es |
| insert | desdem2 | insertdem2 | insertm2@prueba.es |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Jose Luis | Pedraza Roman | josele | joseperoman@correo.ugr.es |
| Paco | Pepe | pape | pacopepe@correo.es |
| esclavo | clavo | esclave | esclavo@eslave.es |
| insert | desdem1 | insertdem1 | insertm1@prueba.es |
| insert | desdem2 | insertdem2 | insertm2@prueba.es |
| cortafuegosm1 | cortafuegosm1 | cortafuegosm1 | insertm1@prueba.es |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> insert into datos(nombre,apellidos,usuario,email) values ("cortafuegos","cortafuegosm2","cortafuegosm2","insertm2@prueba.es");
Query OK, 1 row affected (0.02 sec)
```

Como vemos, la arquitectura maestro-maestro replica correctamente en ambas máquinas la BD “estudiante” modificadas en cada una de ellas.

Servidores Web de Altas Prestaciones