



SERVIDORES WEB DE ALTAS PRESTACIONES

Práctica 5 - Replicación de bases de datos MySQL

César Muñoz Reinoso

Curso 2022-2023

23 de mayo de 2023

Índice

1. Crear una BD e insertar datos	2
2. Replicar una BD MySQL con mysqldump	4
2.1. Copia de BD en máquina M1	4
2.2. Restauración de BD en máquina M2	6
3. Replicación de BD mediante una configuración maestro-esclavo	6
3.1. Configuración de Maestro	6
3.2. Configuración de Esclavo	8
3.3. Comprobación de replicado maestro-esclavo	10
4. Replicación de BD mediante una configuración maestro-maestro	11
5. Configuración IPTABLES para puerto 3306	13

1. Crear una BD e insertar datos

En la máquina M1 creamos una base de datos llamada estudiante y dentro de ella una tabla llamada datos con atributos nombre, apellidos, usuario y email. Introducimos valores correspondientes a nuestro usuario de la UGR (Cesar, Muñoz Reinoso, cesarmunoz, cesarmunoz@correo.ugr.es).

Además creamos otra tabla 'trabajo_final' con atributos nombre1, apellidos1, nombre2, apellidos2 y titulo_trabajo. Esta nos servirá para introducir los grupos de 2 personas que hagan el trabajo final junto con el título del proyecto. También insertamos los datos del grupo en el que estoy para el proyecto 'Configurar un hosting web'.

```
cesarmunoz@m1-cesarmunoz:~$ sudo mysql -u root -p
[sudo] password for cesarmunoz:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, 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 (0,02 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 (0,05 sec)

mysql> show tables;
+-----+
| Tables_in_estudiante |
+-----+
| datos                 |
+-----+
1 row in set (0,00 sec)
```

```
mysql> insert into datos(nombre,apellidos,usuario,email) values ("Cesar", "Muñoz Reinoso", "cesamunoz", "cesarmunoz@correo.ugr.es");
Query OK, 1 row affected (0,01 sec)

mysql> select * from datos;
+-----+-----+-----+-----+
| nombre | apellidos | usuario | email |
+-----+-----+-----+-----+
| Cesar  | Muñoz Reinoso | cesamunoz | cesarmunoz@correo.ugr.es |
+-----+-----+-----+-----+
1 row in set (0,00 sec)

mysql> create table trabajo_final (nombre1 varchar(100), apellidos1 varchar(100), nombre2 varchar(100), apellidos2 varchar(100), titulo_trabajo varchar(500));
Query OK, 0 rows affected (0,04 sec)

mysql> insert into trabajo_final(nombre1, apellidos1, nombre2, apellidos2, titulo_trabajo) values ("Cesar", "Muñoz Reinoso", "Santiago Miquel", "Garcia Santamaria", "Configurar un hosting web" );
Query OK, 1 row affected (0,01 sec)

mysql> select * from trabajo_final;
+-----+-----+-----+-----+-----+
| nombre1 | apellidos1 | nombre2 | apellidos2 | titulo_trabajo |
+-----+-----+-----+-----+-----+
| Cesar   | Muñoz Reinoso | Santiago Miquel | Garcia Santamaria | Configurar un hosting web |
+-----+-----+-----+-----+-----+
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)

mysql> describe trabajo_final;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombre1 | varchar(100) | YES | | NULL | |
| apellidos1 | varchar(100) | YES | | NULL | |
| nombre2 | varchar(100) | YES | | NULL | |
| apellidos2 | varchar(100) | YES | | NULL | |
| titulo_trabajo | varchar(500) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

2. Replicar una BD MySQL con mysqldump

2.1. Copia de BD en máquina M1

Realizaremos una copia de seguridad en frío de la base de datos mediante mysqldump, para posteriormente importarla a la máquina M2. Primero nos conectamos al servidor mysql y bloqueamos la tablas para que no puedan ser modificadas mientras creamos la copia de seguridad.

```
sudo mysql -u root
```

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

Posteriormente creamos la copia de seguridad en el archivo /tmp/estudiante.sql.

Utilizamos `-all-databases` para realizar una copia completa de todas las bases de datos del servidor, en este caso solo tendríamos la BD 'estudiante'; con `-complete-insert` nos aseguramos de que utilizamos sentencias INSERT completas que incluyen nombres de columnas y con `-verbose` imprime la información sobre de lo que está realizando el respaldo.

```

cesarmunoz@m1-cesarmunoz:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, 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
cesarmunoz@m1-cesarmunoz:~$ sudo mysqldump -u root --all-databases --complete-insert --verbose > /tmp/estudiante.sql
-- Connecting to localhost...
-- Retrieving table structure for table columns_priv...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table component...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table db...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table default_roles...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table engine_cost...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table func...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table global_grants...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table gtid_executed...
-- Retrieving table structure for table help_category...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table help_keyword...
-- Sending SELECT query...
-- Retrieving rows...

```

Una vez tenemos el archivo de volcado de copia de seguridad, podemos desbloquear las tablas. Con el comando scp estudiado, copiamos el archivo /tmp/estudiante.sql a la máquina M2.

```

sudo mysql -u root
UNLOCK TABLES;
quit;

```

```

sudo scp /tmp/estudiante.sql cesarmunoz@192.168.56.3:/tmp/estudiante.sql

```

```

cesarmunoz@m1-cesarmunoz:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, 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
cesarmunoz@m1-cesarmunoz:~$ sudo scp /tmp/estudiante.sql cesarmunoz@192.168.56.3:/tmp/estudiante.sql
cesarmunoz@192.168.56.3's password:
estudiante.sql
100% 1261KB 29.8MB/s 00:00

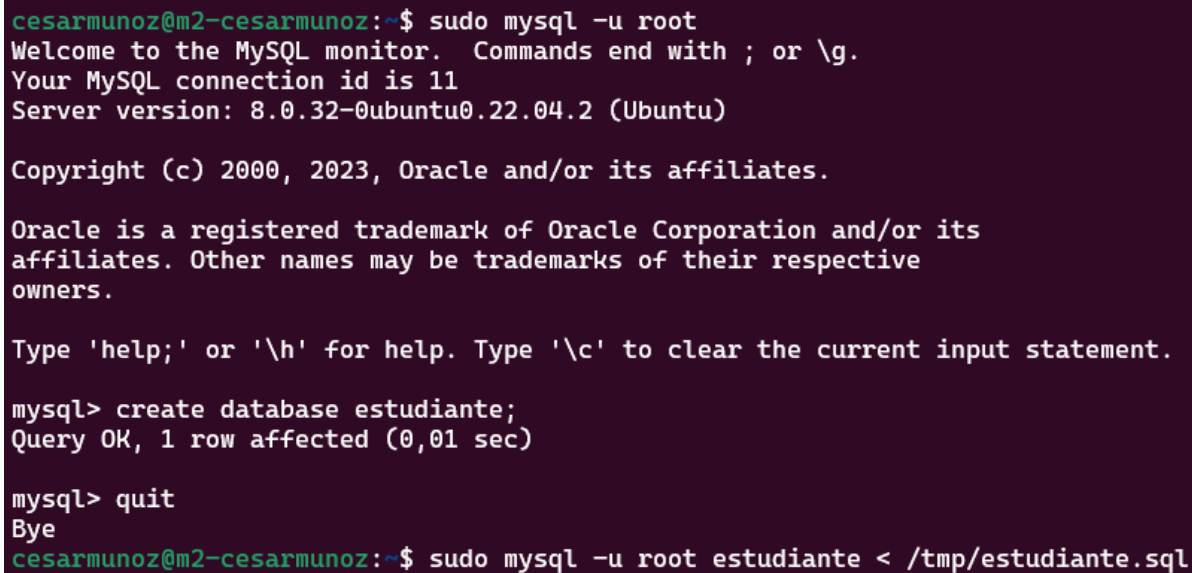
```

2.2. Restauración de BD en máquina M2

Una vez tenemos el archivo /tmp/estudiantes.sql en la máquina M2, la restauramos. Para ello es necesario crear las bases de datos de la que hemos creado la copia de seguridad y posteriormente restaurar la base de datos.

```
sudo mysql -u root
CREATE DATABASE estudiante;
quit;
```

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



```
cesarmunoz@m2-cesarmunoz:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, 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 (0,01 sec)

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

3. Replicación de BD mediante una configuración maestro-esclavo

3.1. Configuración de Maestro

Editamos el archivo /etc/mysql/mysql.conf.d/mysqld.cnf de la máquina maestro e introducimos la información relativa a el log de errores y binarios, así como asignarle el id 1. Comprobamos que funciona reiniciando el servicio mysql.

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
#bind-address 127.0.0.1
log_error = /var/log/mysql/error.log
server-id = 1
```

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

```
sudo service mysql restart
```

```
sudo service mysql status
```

```
GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf *
#bind-address            = 127.0.0.1
mysqlx-bind-address      = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size          = 16M
# max_allowed_packet     = 64M
# thread_stack           = 256K
#
# thread_cache_size      = -1
#
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options   = BACKUP
#
# max_connections        = 151
#
# table_open_cache       = 4000
#
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file       = /var/log/mysql/query.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               = 1
log_bin                 = /var/log/mysql/mysql-bin.log
```

Acto seguido creamos un usuario encargado de replicar la base de datos en el esclavo llamado 'esclavo_cesarmunoz' en la direccion ip de la máquina esclavo.

```
sudo mysql -u root
CREATE USER 'esclavo_cesarmunoz'@192.168.56.3 IDENTIFIED BY 'esclavo_cesarmunoz';
GRANT REPLICATION SLAVE ON *.* TO 'esclavo_cesarmunoz'@192.168.56.3;
FLUSH PRIVILEGES;
FLUSH TABLES;
FLUSH TABLES WITH READ LOCK;
SHOW MASTER STATUS;
```



```
mysql> CREATE USER 'esclavo_cesarmunoz'@192.168.56.3 IDENTIFIED BY 'esclavo_cesarmunoz';
Query OK, 0 rows affected (0,01 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo_cesarmunoz'@192.168.56.3;
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,01 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 |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 | 3720    |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```

3.2. Configuración de Esclavo

En la máquina que actuará como esclavo editamos también el archivo `/etc/mysql/mysql.conf.d/mysqld.cnf` y realizaremos la misma configuración, salvo que en esta estableceremos el `id` a 2.

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
#bind-address 127.0.0.1
log_error = /var/log/mysql/error.log
server-id = 2
log_bin = /var/log/mysql/mysqlbin.log
```

```
sudo service mysql restart
sudo service mysql status
```

También deberemos configurarlo con los datos del maestro, introduciremos la `ip`, el usuario, el archivo de log junto con la posición y el puerto que se utilizará para la replicación de la BD. El puerto utilizado es el 3306, veremos más adelante que será necesario abrirlo mediante `IPTABLES` para que funcione correctamente. Nos damos cuenta de que en el estado del esclavo, el valor `Seconds_Behind_Master` tiene valor null, lo que significa que no funciona la configuración maestro-esclavo, veremos como solucionarlo.

```
sudo mysql -u root
CHANGE MASTER TO MASTER_HOST='192.168.56.2', MASTER_USER='esclavo_cesarmunoz',
MASTER_PASSWORD='esclavo_cesarmunoz', MASTER_LOG_FILE='mysql-bin.000001',
MASTER_LOG_POS=3720, MASTER_PORT=3306;
```

```
START SLAVE;  
SHOW SLAVE STATUS\G;  
(VEMOS QUE Seconds_Behind_Master: NULL)
```

```
STOP SLAVE;  
CHANGE MASTER TO GET_MASTER_PUBLIC_KEY=1;  
START SLAVE;  
SHOW SLAVE STATUS\G;  
(VEMOS QUE Seconds_Behind_Master: 0)
```

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.2', MASTER_USER='esclavo_cesarmunoz', MASTER_PASSWORD='esclavo_cesarmunoz', MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=3720, MASTER_  
PORT=3306;  
Query OK, 0 rows affected, 9 warnings (0,11 sec)  
  
mysql> START SLAVE;  
Query OK, 0 rows affected, 1 warning (0,04 sec)
```

```

mysql> STOP SLAVE;
Query OK, 0 rows affected, 1 warning (0,02 sec)

mysql> CHANGE MASTER TO GET_MASTER_PUBLIC_KEY=1;
Query OK, 0 rows affected, 2 warnings (0,03 sec)

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0,05 sec)

mysql> SHOW SLAVE STATUS\G
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
        Master_Host: 192.168.56.2
        Master_User: esclavo_cesarmunoz
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 3720
        Relay_Log_File: m2-cesarmunoz-relay-bin.000002
        Relay_Log_Pos: 326
      Relay_Master_Log_File: mysql-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: 3720
      Relay_Log_Space: 544
      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

```

3.3. Comprobación de replicado maestro-esclavo

En la máquina que actúa como maestro introducimos nuevas tuplas en la tabla 'datos' y comprobamos que efectivamente se replican en la máquina esclavo.

```

mysql> insert into datos(nombre,apellidos,usuario,email) values ("Prueba", "Prueba Apellidos", "prueba", "prueba@correo.ugr.es");
Query OK, 1 row affected (0,01 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
Cesar	Muñoz Reinoso	cesamunoz	cesarmunoz@correo.ugr.es
Prueba	Prueba Apellidos	prueba	prueba@correo.ugr.es

```
2 rows in set (0,00 sec)
```

4. Replicación de BD mediante una configuración maestro-maestro

Para una configuración maestro-maestro replicamos la configuración M1 - M2 pero intercambiando los roles. Editamos el archivo /etc/mysql/mysql.conf.d/mysqld.cnf como hemos hecho anteriormente, creamos un usuario pero esta vez con la ip de la otra máquina y volvemos a ejecutar la sentencia para que Seconds_Behind_Master:0, comprobamos que realmente están conectadas.

```
mysql> CREATE USER 'esclavo_cesarmunoz'@192.168.56.2 IDENTIFIED BY 'esclavo_cesarmunoz';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'esclavo_cesarmunoz'@192.168.56.2;
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH TABLES;
Query OK, 0 rows affected (0,01 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	1417			

```
1 row in set (0,00 sec)
```

```
mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.3', MASTER_USER='esclavo_cesarmunoz', MASTER_PASSWORD='esclavo_cesarmunoz', MASTER_LOG_FILE='mysql-bin.000002', MASTER_LOG_POS=1417, MASTER_PORT=3306;  
Query OK, 0 rows affected, 9 warnings (0,06 sec)
```

```
mysql> START SLAVE;  
Query OK, 0 rows affected, 1 warning (0,03 sec)
```

```
mysql> STOP SLAVE;  
Query OK, 0 rows affected, 1 warning (0,02 sec)
```

```
mysql> CHANGE MASTER TO GET_MASTER_PUBLIC_KEY=1;  
Query OK, 0 rows affected, 2 warnings (0,03 sec)
```

```
mysql> START SLAVE;  
Query OK, 0 rows affected, 1 warning (0,03 sec)
```

```
mysql> SHOW SLAVE STATUS\G  
***** 1. row *****  
Slave_IO_State: Waiting for source to send event  
Master_Host: 192.168.56.3  
Master_User: esclavo_cesarmunoz  
Master_Port: 3306  
Connect_Retry: 60  
Master_Log_File: mysql-bin.000002  
Read_Master_Log_Pos: 1417  
Relay_Log_File: m1-cesarmunoz-relay-bin.000002  
Relay_Log_Pos: 326  
Relay_Master_Log_File: mysql-bin.000002  
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: 1417  
Relay_Log_Space: 544  
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:
```

5. Configuración IPTABLES para puerto 3306

Como en la práctica anterior hemos bloqueado las conexiones entrantes y salientes de M1 y M2 salvo las conexiones de SSH, HTTP/HTTPS desde la máquina M3, debemos añadir la sentencia para permitir la conexión entrante y saliente de una máquina a otra por el puerto 3306.

```
#Abrir el puerto para la replicación de la BD (3306)
```

```
iptables -A INPUT -s 192.168.56.3 -p tcp --dports 3306 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.3 -p tcp --sports 3306 -j ACCEPT
```

```
-----
#Abrir el puerto para la replicación de la BD (3306)
```

```
iptables -A INPUT -s 192.168.56.2 -p tcp --dports 3306 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.2 -p tcp --sports 3306 -j ACCEPT
```

```
GNU nano 6.2                                iptables_script.sh
#!/bin/bash

#Eliminar al mismo tiempo todas sus reglas

iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Denegar todo el trafico

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Abrir los puertos SSH y HTTP/HTTPS (22, 80 y 443)
iptables -A INPUT -s 192.168.56.4 -p tcp -m multiport --dports 22,80,443 -j ACCEPT
iptables -A OUTPUT -d 192.168.56.4 -p tcp -m multiport --sports 22,80,443 -j ACCEPT

#Abrir el puerto para la replicación de la BD (3306)

iptables -A INPUT -s 192.168.56.3 -p tcp --dports 3306 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.3 -p tcp --sports 3306 -j ACCEPT
```

```
GNU nano 6.2                                ssl/iptables_script.sh
#!/bin/bash

#Eliminar al mismo tiempo todas sus reglas

iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#Denegar todo el trafico

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#Abrir los puertos SSH y HTTP/HTTPS (22, 80 y 443)
iptables -A INPUT -s 192.168.56.4 -p tcp -m multiport --dports 22,80,443 -j ACCEPT
iptables -A OUTPUT -d 192.168.56.4 -p tcp -m multiport --sports 22,80,443 -j ACCEPT

#Abrir el puerto para la replicación de la BD (3306)

iptables -A INPUT -s 192.168.56.2 -p tcp --dports 3306 -j ACCEPT
iptables -A OUTPUT -s 192.168.56.2 -p tcp --sports 3306 -j ACCEPT
```