

E.T.S. de Ingenierías Informática y de
Telecomunicación



Servidores Web de Altas Prestaciones

Trabajo de Teoría- Configuración Base de Datos Maestro-Maestro en Mysql

Manuel Gutiérrez Delgado

Andrés Serrano Gómez

En este trabajo vamos a configurar una base de datos maestro-maestro en dos máquinas virtuales y a comprobar su correcto funcionamiento.

Como objetivos nos marcamos estos:

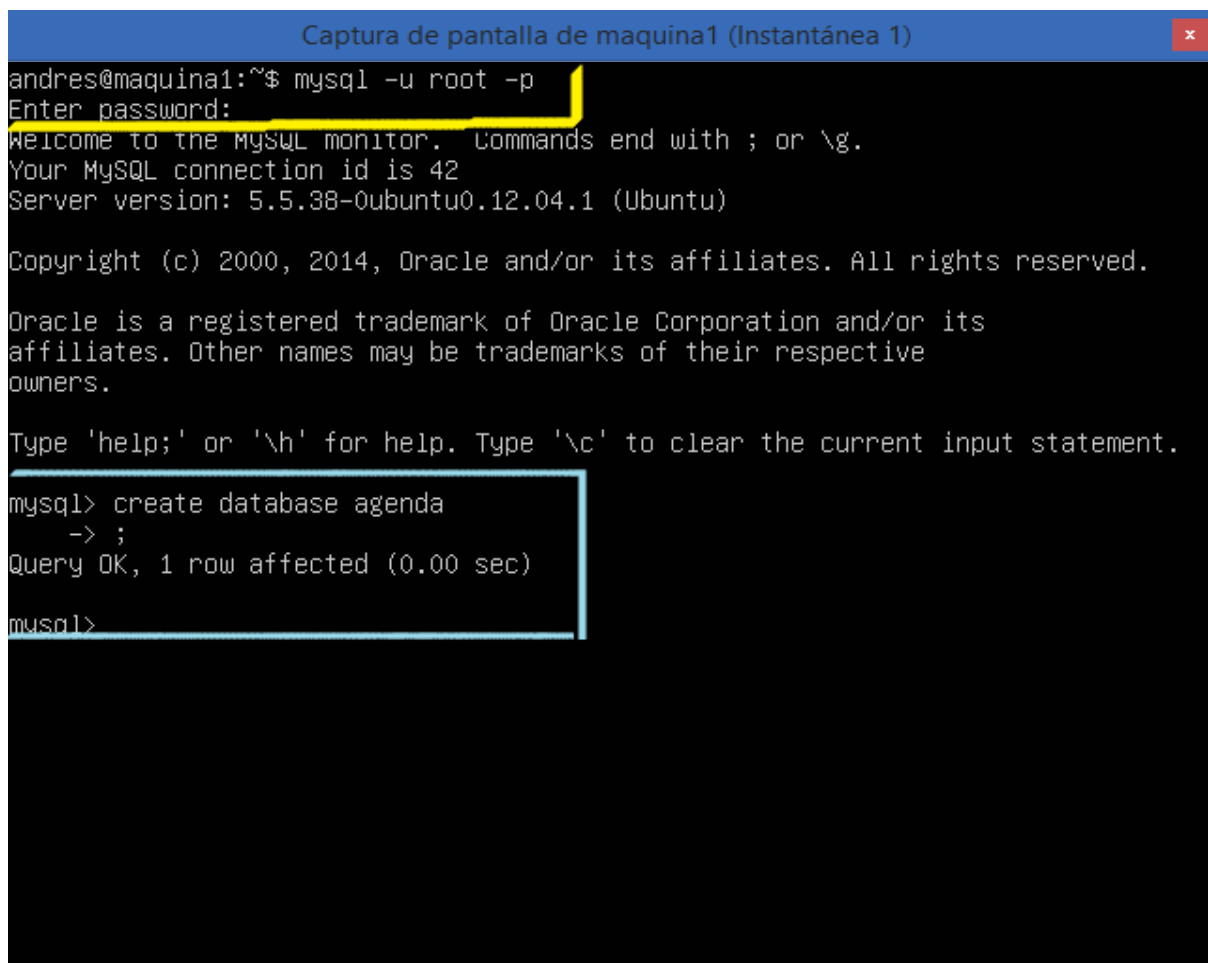
- ❖ Configurar correctamente las dos máquinas y la estructura necesaria para desarrollar el trabajo.
- ❖ Configurar correctamente toda la estructura de la base de datos y comenzar las pruebas.
- ❖ Comprobar que realmente lo que queríamos hacer se traslada a las máquinas y poder mostrarlo.

Y nuestra motivación fue esta:

- ❖ Hoy en día se utilizan Bases de datos en cualquier ámbito ya que cada vez la cantidad de información que se tiene es mayor y por ejemplo para cualquier empresa es muy necesario disponer de ella.
- ❖ La configuración de esta base de datos, maestro-maestro conlleva una gran ventaja entre otras y es que es mas fácil proporcionar alta disponibilidad porque si un maestro falla todavía se tiene el otro maestro y que todo lo que se escriba en uno o se modifique, se replicará en el otro.

Primero vamos a comenzar creando la misma base de datos en ambas máquinas. Para ello, entraremos en mysql ejecutando el comando **“mysql -u root -p”**.

Una vez dentro, crearemos la base de datos que se llamará “agenda”. Para ello, dentro de mysql, ejecutamos **“create database agenda;”**.



```
Captura de pantalla de maquina1 (Instantánea 1)
andres@maquina1:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.38-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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.

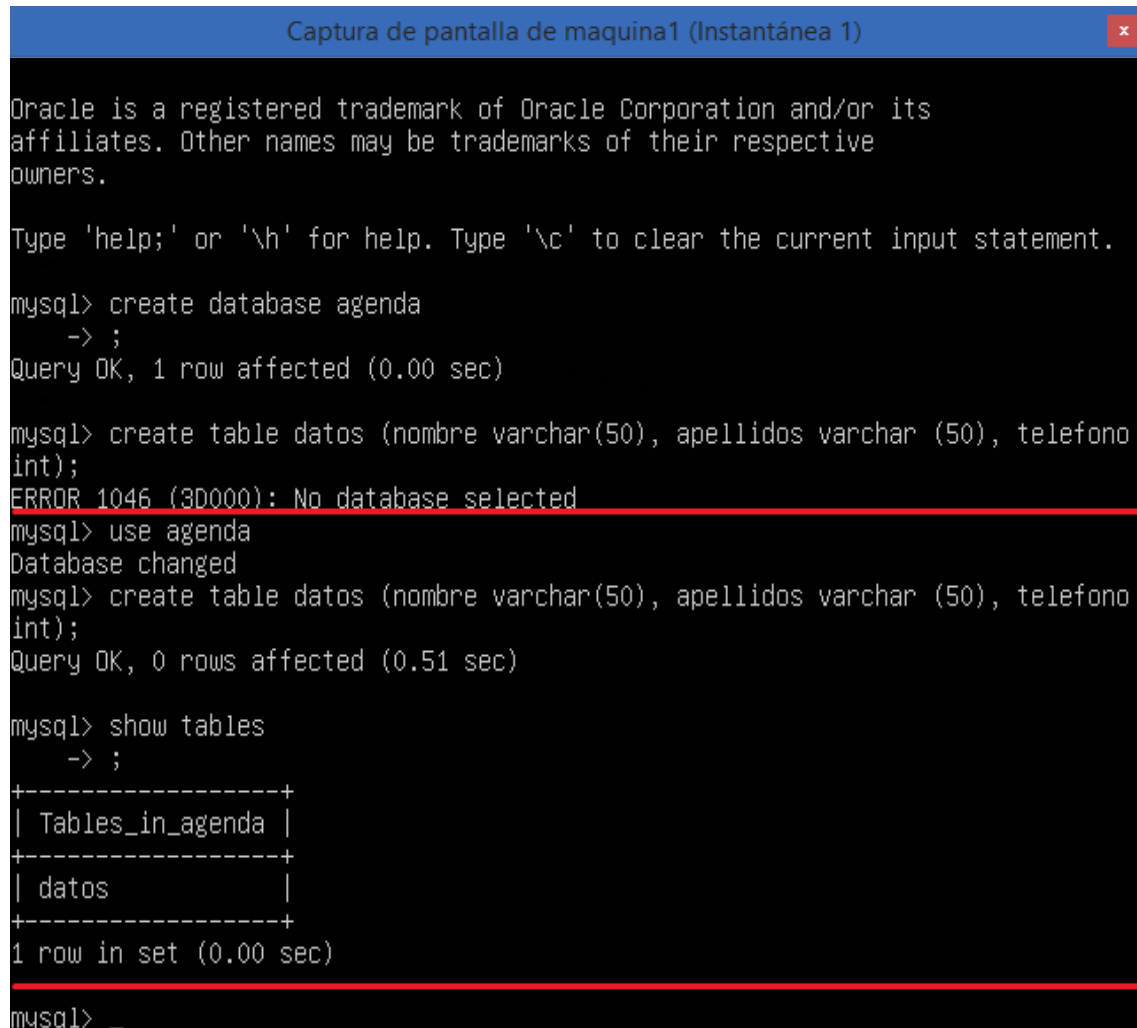
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database agenda
-> ;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Ahora crearemos una tabla llamada “datos” con “nombre, apellidos y teléfono”. Para ello, primero seleccionamos la base de datos que vamos a usar con el comando, “**use agenda**”, después ejecutamos “**create table datos (nombre varchar (50), apellidos varchar (50), teléfono int);**”

Y después comprobamos que se ha creado haciendo “**show tables;**”



```
Captura de pantalla de maquina1 (Instantánea 1)
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 agenda
-> ;
Query OK, 1 row affected (0.00 sec)

mysql> create table datos (nombre varchar(50), apellidos varchar (50), telefono
int);
ERROR 1046 (3D000): No database selected
mysql> use agenda
Database changed
mysql> create table datos (nombre varchar(50), apellidos varchar (50), telefono
int);
Query OK, 0 rows affected (0.51 sec)

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

mysql>
```

Ahora repetiremos los mismo pasos en la máquina 2. Crearemos la base de datos y a continuación la tabla de datos.

```
Captura de pantalla de maquina2 (Instantánea 1)
Your MySQL connection id is 42
Server version: 5.5.38-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2014, 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 agenda;
Query OK, 1 row affected (0.00 sec)

mysql> use agenda
Database changed
mysql> create table datos (nombre varchar(50), apellidos varchar(50), telefono i
nt);
Query OK, 0 rows affected (0.06 sec)

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

mysql>
```

Una vez que tenemos creada la base de datos en ambas máquinas, vamos a proceder a configurarlas.

Vamos a comenzar a configurar la máquina1. Para ello, editaremos el archivo “**/etc/mysql/my.cnf**”. Cuando estemos dentro lo primero que haremos será comentar la línea “**#bind-address 127.0.0.1**” y seguidamente añadiremos estas líneas:

server-id = 1

log_bin = /var/log/mysql/bin.log //(aquí se almacenará los cambios que se realicen en una base de datos o tabla)

log_error = /var/log/mysql/error.log //(almacena si ha ocurrido algún error crítico mientras el servidor se estaba ejecutando.)

MAESTRO-MAESTRO MYSQL

```
GNU nano 2.2.6      File: /etc/mysql/my.cnf      Modified
datadir            = /var/lib/mysql
tmpdir             = /tmp
lc-messages-dir    = /usr/share/mysql
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
#
server-id          =1
log_bin           = /var/log/mysql/bin.log
log_error         = /var/log/mysql/error.log

#
# * Fine Tuning
#
key_buffer         = 16M
max_allowed_packet = 16M
thread_stack       = 192K
thread_cache_size  = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover     = BACKUP
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Ahora guardamos el documento y reiniciamos el servicio con
“**sudo /etc/init.d/mysql restart**”.

```
#bind-address          = 127.0.0.1
#
server-id              =1
log_bin                = /var/log/mysql/bin.log
log_error              = /var/log/mysql/error.log
#
# * Fine Tuning
#
key_buffer              = 16M
max_allowed_packet      = 16M
thread_stack           = 192K
thread_cache_size       = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover          = BACKUP
[ Wrote 134 lines ]

andres@maquina1:~$ sudo /etc/init.d/mysql restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service mysql restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop mysql ; start mysql. The restart(8) utility is also available.
mysql stop/waiting
mysql start/running, process 1258
andres@maquina1:~$
```

Ahora realizaremos el mismo proceso en la máquina2. Entramos en el archivo “/etc/mysql/my.cnf”. Una vez dentro comentaremos la línea “#bind-address 127.0.0.1”, y una vez comentada añadiremos las siguientes líneas:

server-id = 2

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

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

MAESTRO-MAESTRO MYSQL

```
GNU nano 2.2.6      File: /etc/mysql/my.cnf      Modified
datadir            = /var/lib/mysql
tmpdir             = /tmp
lc-messages-dir    = /usr/share/mysql
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
#
server-id          = 2
log_bin           = /var/log/mysql/bin.log
log_error         = /var/log/mysql/error.log
#
# * Fine Tuning
#
key_buffer         = 16M
max_allowed_packet = 16M
thread_stack       = 192K
thread_cache_size  = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover     = BACKUP
#max_connections   = 100

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Y ahora reiniciaremos el servicio con “/etc/init.d/mysql restart”


```
#bind-address          = 127.0.0.1
#
server-id              = 2
log_bin                = /var/log/mysql/bin.log
log_error              = /var/log/mysql/error.log
#
# * Fine Tuning
#
key_buffer              = 16M
max_allowed_packet     = 16M
thread_stack           = 192K
thread_cache_size      = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover          = BACKUP
#max_connections       = 100
[ line 54/134 (40%), col 2/2 (100%), char 1506/3595 (41%) ]

andres@maquina2:~$ sudo /etc/init.d/mysql restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service mysql restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop mysql ; start mysql. The restart(8) utility is also available.
mysql stop/waiting
mysql start/running, process 1353
andres@maquina2:~$ _
```

Ahora vamos a proceder a crear un usuario en cada una de las máquinas (serán los mismos) y le daremos permisos. Vamos a comenzar con la máquina1. Entramos en mysql y ejecutamos las siguientes sentencias:

CREATE USER maestro IDENTIFIED BY 'usuario';

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

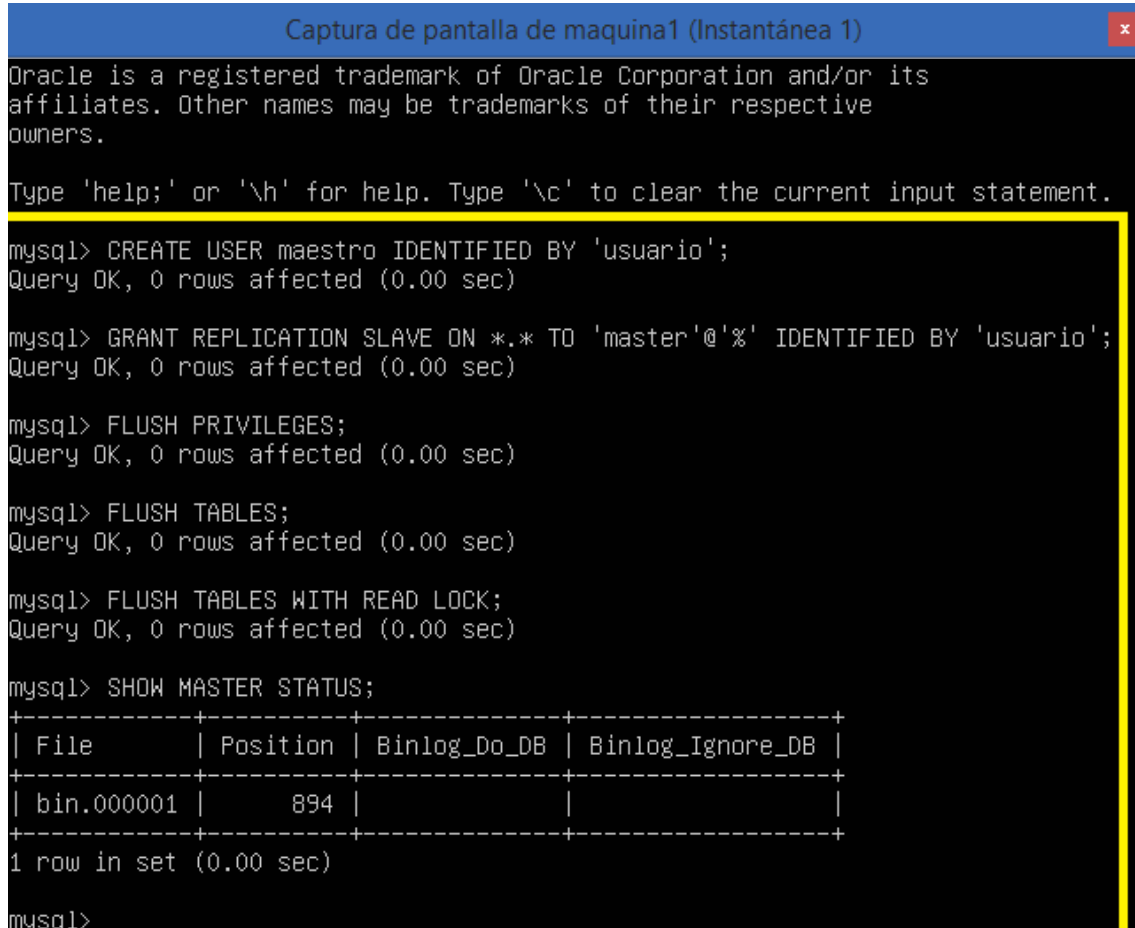
FLUSH PRIVILEGES; (**Recarga los permisos de las tablas de permisos en la base de datos mysql**).

FLUSH TABLES; (**cierra todas las tablas abiertas y fuerza a todas las tablas en uso a que se cierren**).

FLUSH TABLES WITH READ LOCK; (**Cierra todas las tablas abiertas y bloquea todas las tablas para todas las bases de datos con una bloqueo de lectura hasta que ejecute UNLOCK TABLES**).

Para acabar con la configuración en el máquina1 obtenemos los datos de la base de datos que vamos a replicar para posteriormente usarlos más adelante.

Ejecutamos en mysql “**SHOW MASTER STATUS**”.



```
Captura de pantalla de maquina1 (Instantánea 1)
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 'usuario';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'master'@'%' IDENTIFIED BY 'usuario';
Query OK, 0 rows affected (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 |
+-----+-----+-----+-----+
| bin.000001 |      894 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Ahora vamos a repetir este mismo proceso en Máquina2.

CREATE USER maestro IDENTIFIED BY 'usuario';

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

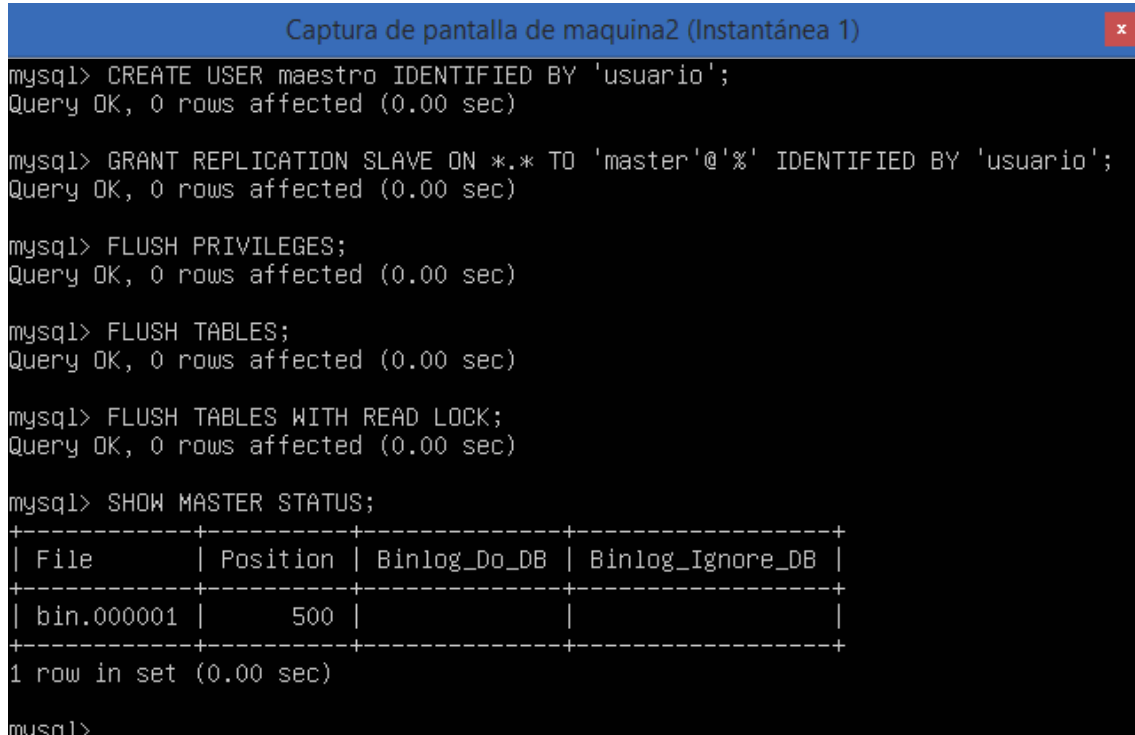
FLUSH PRIVILEGES;

FLUSH TABLES;

FLUSH TABLES WITH READ LOCK;

Para acabar con la configuración en el máquina2 obtenemos los datos de la base de datos que vamos a replicar para posteriormente usarlos más adelante.

Ejecutamos en mysql **“SHOW MASTER STATUS”**.



```
mysql> CREATE USER maestro IDENTIFIED BY 'usuario';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'master'@'%' IDENTIFIED BY 'usuario';
Query OK, 0 rows affected (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 |
+-----+-----+-----+-----+
| bin.000001 |      500 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Ahora volveremos a la máquina 1, entramos en mysql y le damos los datos del maestro. En el entorno de mysql ejecutamos la siguiente sentencia, donde HOST es la ip de la máquina2.

STOP SLAVE;

**CHANGE MASTER TO MASTER_HOST='192.168.56.106',
MASTER_USER='master', MASTER_PASSWORD='usuario',
MASTER_LOG_FILE='bin.000001', MASTER_LOG_POS=500,
MASTER_PORT=3306;**

START SLAVE;

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

mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.106',
    -> MASTER_USER='master',MASTER_PASSWORD='usuario',
    -> MASTER_LOG_FILE='bin.000001',MASTER_LOG_POS=500,
    -> MASTER_PORT=3306;
Query OK, 0 rows affected (0.10 sec)

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

mysql>
```

Y ahora repetiremos el mismo proceso para la **máquina 2**, pero esta vez en master-host pondremos la ip de la **máquina1**.

STOP SLAVE;

**CHANGE MASTER TO MASTER_HOST='192.168.56.105',
MASTER_USER='master', MASTER_PASSWORD='usuario',
MASTER_LOG_FILE='bin.000001', MASTER_LOG_POS=894,
MASTER_PORT=3306;**

START SLAVE;

MAESTRO-MAESTRO MYSQL

```
[ Read 133 lines ]
andres@maquina2:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.38-0ubuntu0.12.04.1-log (Ubuntu)

Copyright (c) 2000, 2014, 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> STOP SLAVE
-> ;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CHANGE MASTER TO MASTER_HOST='192.168.56.105',
-> MASTER_USER='master',MASTER_PASSWORD='usuario',
-> MASTER_LOG_FILE='bin.000001',MASTER_LOG_POS=894,
-> MASTER_PORT=3306;
Query OK, 0 rows affected (0.40 sec)

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

mysql> _
```

Por último, antes de comenzar a hacer las pruebas tendremos que desbloquear las tablas en la máquina1 y máquina2. Para ello, dentro de mysql, ejecutaremos “UNLOCK TABLES;”

```
Captura de pantalla de maquina1 (Instantánea 1)
mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> use agenda
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_agenda |
+-----+
| datos            |
+-----+
1 row in set (0.00 sec)

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

mysql> _
```

MAESTRO-MAESTRO MYSQL



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

mysql> use agenda
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_agenda |
+-----+
| datos              |
+-----+
1 row in set (0.00 sec)

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

mysql> _
```

Una vez hecho, ya podremos empezar a probar si al realizar cambios en cualquiera de las máquinas, también se producen en la otra. Comenzamos con las tablas vacías en ambas máquinas.

MAESTRO-MAESTRO MYSQL

Vemos que de primeras no hay nada en la tabla ejecutando, **“select * from datos;**

```
Captura de pantalla de maquina1 (Instantánea 1) x
mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> use agenda
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_agenda |
+-----+
| datos            |
+-----+
1 row in set (0.00 sec)

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

mysql>
```

```
Captura de pantalla de maquina2 (Instantánea 1) x
mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> use agenda
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_agenda |
+-----+
| datos            |
+-----+
1 row in set (0.00 sec)

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

mysql> _
```

MAESTRO-MAESTRO MYSQL

Hacemos un “**SHOW SLAVE STATUS\G**” en mysql en cada una de las máquinas y comprobamos que la variable “**Seconds_Behind_Master**” es distinta de NULL, ya que indica que "retraso" tiene un servidor respecto del otro.

Captura de pantalla de maquina1 (Instantánea 1)	Captura de pantalla de maquina2 (Instantánea 1)
<pre>Replicate_Wild_Do_Table: Replicate_Wild_Ignore_Table: Last_Errno: 0 Last_Error: Skip_Counter: 0 Exec_Master_Log_Pos: 500 Relay_Log_Space: 404 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: 2 1 row in set (0.00 sec) ERROR: No query specified</pre>	<pre>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: 894 Relay_Log_Space: 404 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 1 row in set (0.00 sec)</pre>

MAESTRO-MAESTRO MYSQL

Ahora vamos a añadir datos en la máquina1 y veremos que aparece en la máquina2.

Captura de pantalla de maquina1 (Instantánea 1)	Captura de pantalla de maquina2 (Instantánea 1)
<pre>Master_Server_Id: 2 1 row in set (0.00 sec) ERROR: No query specified mysql> show tables; +-----+ Tables_in_agenda +-----+ datos +-----+ 1 row in set (0.00 sec) mysql> select * from datos; Empty set (0.00 sec) mysql> insert into datos (nombre, apellidos, telefono) values ("Manuel", "Gutierrez", 958405548); Query OK, 1 row affected (0.37 sec) mysql> select * from datos; +-----+-----+-----+ nombre apellidos telefono +-----+-----+-----+ Manuel Gutierrez 958405548 +-----+-----+-----+ 1 row in set (0.00 sec)</pre>	<pre>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 1 row in set (0.00 sec) mysql> show tables; +-----+ Tables_in_agenda +-----+ datos +-----+ 1 row in set (0.00 sec) mysql> select * from datos; +-----+-----+-----+ nombre apellidos telefono +-----+-----+-----+ Manuel Gutierrez 958405548 +-----+-----+-----+ 1 row in set (0.00 sec)</pre>

MAESTRO-MAESTRO MYSQL

Vamos a incluir otra.

```
Captura de pantalla de maquina1 (Instantánea 1)
mysql> select * from datos;
Empty set (0.00 sec)

mysql> insert into datos (nombre, apellidos, telefono) values ("Manuel", "Gutierrez", 958405548);
Query OK, 1 row affected (0.37 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into datos (nombre, apellidos, telefono) values ("ANDRES", "SERRANO GOMEZ", 958207419);
Query OK, 1 row affected (0.06 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
Captura de pantalla de maquina2 (Instantánea 1)
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
1 row in set (0.00 sec)

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

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

MAESTRO-MAESTRO MYSQL

Ahora vamos a hacerlo al contrario. Introducimos desde la máquina 2 y veremos como también se replica en la máquina 1.

Captura de pantalla de maquina1 (Instantánea 1)

```
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into datos (nombre, apellidos, telefono) values ("ANDRES", "SERRANO GOMEZ", 958207419);
Query OK, 1 row affected (0.06 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
| JUAN DE DIOS | MARTINEZ MARTINEZ | 662145560 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Captura de pantalla de maquina2 (Instantánea 1)

```
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into datos (nombre, apellidos, telefono) values ("JUAN DE DIOS", "MARTINEZ MARTINEZ", 662145560);
Query OK, 1 row affected (0.06 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
| JUAN DE DIOS | MARTINEZ MARTINEZ | 662145560 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

MAESTRO-MAESTRO MYSQL

Ahora vamos a borrar 1 dato desde la máquina 1 y deberá actualizarse en la máquina 2.

Captura de pantalla de maquina1 (Instantánea 1)

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

nombre	apellidos	telefono
Manuel	Gutierrez	958405548
ANDRES	SERRANO GOMEZ	958207419

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

```
mysql> delete from datos where nombre="JUAN DE DIOS";
```

```
Query OK, 1 row affected (0.08 sec)
```

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

nombre	apellidos	telefono
Manuel	Gutierrez	958405548
ANDRES	SERRANO GOMEZ	958207419

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

Captura de pantalla de maquina2 (Instantánea 1)

```
mysql> insert into datos (nombre, apellidos, telefono) values ("JUAN DE DIOS", "MARTINEZ MARTINEZ", 662145560);
```

```
Query OK, 1 row affected (0.06 sec)
```

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

nombre	apellidos	telefono
Manuel	Gutierrez	958405548
ANDRES	SERRANO GOMEZ	958207419
JUAN DE DIOS	MARTINEZ MARTINEZ	662145560

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

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

nombre	apellidos	telefono
Manuel	Gutierrez	958405548
ANDRES	SERRANO GOMEZ	958207419

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

MAESTRO-MAESTRO MYSQL

Y ahora borraremos un dato desde máquina2 y deberá producirse el cambio en máquina1.

Captura de pantalla de maquina1 (Instantánea 1)

```
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
| JUAN DE DIOS | MARTINEZ MARTINEZ | 662145560 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> delete from datos where nombre="JUAN DE DIOS";
Query OK, 1 row affected (0.08 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Captura de pantalla de maquina2 (Instantánea 1)

```
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
| JUAN DE DIOS | MARTINEZ MARTINEZ | 662145560 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> delete from datos where nombre="Manuel";
Query OK, 1 row affected (0.09 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

MAESTRO-MAESTRO MYSQL

Por último, y para acabar con esto vamos a hacer una actualización desde cualquiera de las máquinas por ejemplo en la máquina 1 y cambiaremos el apellido y el teléfono.

Captura de pantalla de maquina1 (Instantánea 1)

```
mysql> select * from datos;
+-----+-----+-----+
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE datos SET apellidos="Lozano" WHERE apellidos="SERRANO GOMEZ";
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | Lozano    | 958207419 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> UPDATE datos SET telefono="610588744" WHERE telefono="958207419";
Query OK, 1 row affected (0.38 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | Lozano    | 610588744 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Captura de pantalla de maquina2 (Instantánea 1)

```
mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| Manuel | Gutierrez  | 958405548 |
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> delete from datos where nombre="Manuel";
Query OK, 1 row affected (0.09 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | SERRANO GOMEZ | 958207419 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from datos;
+-----+-----+-----+
| nombre | apellidos | telefono |
+-----+-----+-----+
| ANDRES | Lozano    | 610588744 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

BIBLIOGRAFÍA

- <http://dev.mysql.com/doc/refman/5.0/en/replication.html>
- <http://www.digitalvalley.com/blog/configurar-arquitectura-maestro-maestro-mysql-para-la-replicacion-de-datos/>
- <http://miguelangelnieto.net/?action=view&url=arquitectura-maestro-maestro-en-mysql>
- <http://luismido.wikidot.com/mysql5-replicacion-maestro-maestro>
- <http://panicoenelcpd.blogspot.com.es/2011/11/configurar-mysql-en-modo-maestro.html>