BRIEF NOTE ON THE MYSQL MASTER-SLAVE REPLICATION

What to do when you want to analyze data contained in database without overloading the server? The answer is simple as you can use MySQL Master-Slave replication to offload this task. I will describe it briefly here.

Default *my.cnf* configuration used in this example:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 1 - Master server

You need to define unique server id (*server-id* parameter) in *my.cnf* configuration file as it will identify this server.

```
[mysqld]
datadir=/var/lib/mysql
```

```
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0

server-id=1

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 2 - Master server

Define binary log (*log_bin* parameter) as it provides a record of the data changes to be sent to slave server. Set database to replicate (*binlog-do-db* parameter).

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0

server-id=1
log_bin = /var/lib/mysql/mysql-log-bin
binlog_do_db=my_database
```

```
[mysqld_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid
```

Restart MySQL server:

```
$ sudo service mysqld restart
```

Step 3 - Master server

Create MySQL user for replication purpose:

```
mysql> create user 'replication'@'%' identified by 'replication_password';

mysql> grant replication slave on *.* to 'replication'@'%';
```

Grant permission to read binary log events from the master:

```
mysql> show grants for 'replication'@'%';

+-------------------------------------------------------------------------------+

| Grants for replication@%                                                      |

+-------------------------------------------------------------------------------+

| GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%' IDENTIFIED BY

PASSWORD '*' |

+-------------------------------------------------------------------------------+

1 row in set (0.00 sec)
```

Reload the privileges from the grant tables:

```
mysql> flush privileges;
```

Step 4 - Slave Server

Define unique server id and relay log file (*relay-log* parameter) in configuration file:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0

server-id=2
relay-log=/var/lib/mysql/relay.log

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 5 - Master server

Close all open tables and lock them in desired database:

```
mysql> use my_database;

mysql> flush tables with read lock;
```

Dump desired database so you can move it to the slave server:

```
$ mysqldump -u root -p  my_database > my_database.sql
```

Step 6 - Slave server

Import database exported in the earlier step:

```
$ mysql -u root -p my_database < my_database.sql
```

Step 7 - Master server

Check master status and note *File* and *Position* values:

```
mysql> show master status;

+----------------------+----------+--------------+------------------+

| File                 | Position | Binlog_Do_DB | Binlog_Ignore_DB |

+----------------------+----------+--------------+------------------+

| mysql_bin_log.000001 |      106 |  my_database |                  |

+----------------------+----------+--------------+------------------+

1 row in set (0.00 sec)
```

Step 8 - Slave server

Enable replication on the slave server:

```
mysql> change master to
master_host='master_hostname',master_user='replication',master_password='replication_passwo
rd',master_log_file='mysql_bin_log.000001',master_log_pos=106;
```

Notice that *master_log_file* and *master_log_pos* parameters use values from previous step.

Enable replication:

```
mysql> start slave;
```

Step 9 - Master server

Unlock tables:

```
mysql> use my_database;
mysql> unlock tables;
```

MySQL Master-Slave configuration is done.

Source: https://blog.sleeplessbeastie.eu/2013/04/24/brief-note-on-the-mysql-master-slave-replication/