By Oliver Meyer
Published: 2008-02-17 19:06

Master-Master Replication With MySQL 5 On Fedora 8

Version 1.0
Author: Oliver Meyer <o [dot] meyer [at] projektfarm [dot] de>
Last edited 02/12/2008

This document describes how to set up master-master replication with MySQL 5 on Fedora 8. Since version 5, MySQL comes with built-in support for master-master replication, solving the problem that can happen with self-generated keys. In former MySQL versions, the problem with master-master replication was that conflicts arose immediately if node A and node B both inserted an auto-incrementing key on the same table. The advantages of master-master replication over the traditional master-slave replication are that you don't have to modify your applications to make write accesses only to the master, and that it is easier to provide high-availability because if the master fails, you still have the other master.

This howto is a practical guide without any warranty - it doesn't cover the theoretical backgrounds. There are many ways to set up such a system - this is the way I chose.

1 Preparation

For this howto I set up two Fedora 8 systems (minimal installation without gui etc.) with the following configuration.

1.1 System 1

Hostname: *`server1.example.com`*
IP: *`192.168.0.100`*

1.2 System 2

Hostname: *`server2.example.com`*
IP: *`192.168.0.200`*

2 MySQL2.1 Needed Packages On Both Systems

If you haven't installed MySQL on both systems you can install it (client & server) via:

```
yum -y install mysql mysql-server
```

2.2 MySQL Server Initial Start On Both Systems

Start the MySQL server.

```
/etc/init.d/mysqld start
```

2.3 MySQL Root Password2.3.1 Both Systems

Set a password for the MySQL root-user on localhost.

```
mysqladmin -u root password %sql_root_password%
```

2.3.2 System 1

Set a password for the MySQL root-user on server1.example.com.

```
mysqladmin -u root -h server1.example.com password %mysql_root_password%
```

2.3.3 System 2

Set a password for the MySQL root-user on server2.example.com.

```
mysqladmin -u root -h server2.example.com password %mysql_root_password%
```

2.4 MySQL Replication User2.4.1 System 1

Create the replication user that System 2 will use to access the MySQL database on System 1.

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

```
GRANT REPLICATION SLAVE ON *.* TO 'slave2_user'@'%' IDENTIFIED BY '%mysql_slaveuser_password%';

FLUSH PRIVILEGES;

quit;
```

2.4.2 System 2

Create the replication user that System 1 will use to access the MySQL database on System 2.

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

```
GRANT REPLICATION SLAVE ON *.* TO 'slave1_user'@'%' IDENTIFIED BY '%mysql_slaveuser_password%';

FLUSH PRIVILEGES;

quit;
```

2.5 Database On System 2

I proceed on the assumption that the database `exampledb` is already existing on System 1 - containing tables with records. So we have to create an empty database with the same name as the existing database on System 1.

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

```
CREATE DATABASE exampledb;

quit;
```

3 Replication

3.1 Firewall Configuration On Both Systems

Versions of `system-config-firewall-tui` before 1.0.12-4.x had a bug in conjunction with custom rules (they were not aquired) - so check which version is installed on your system.

```
yum list installed | grep firewall
```

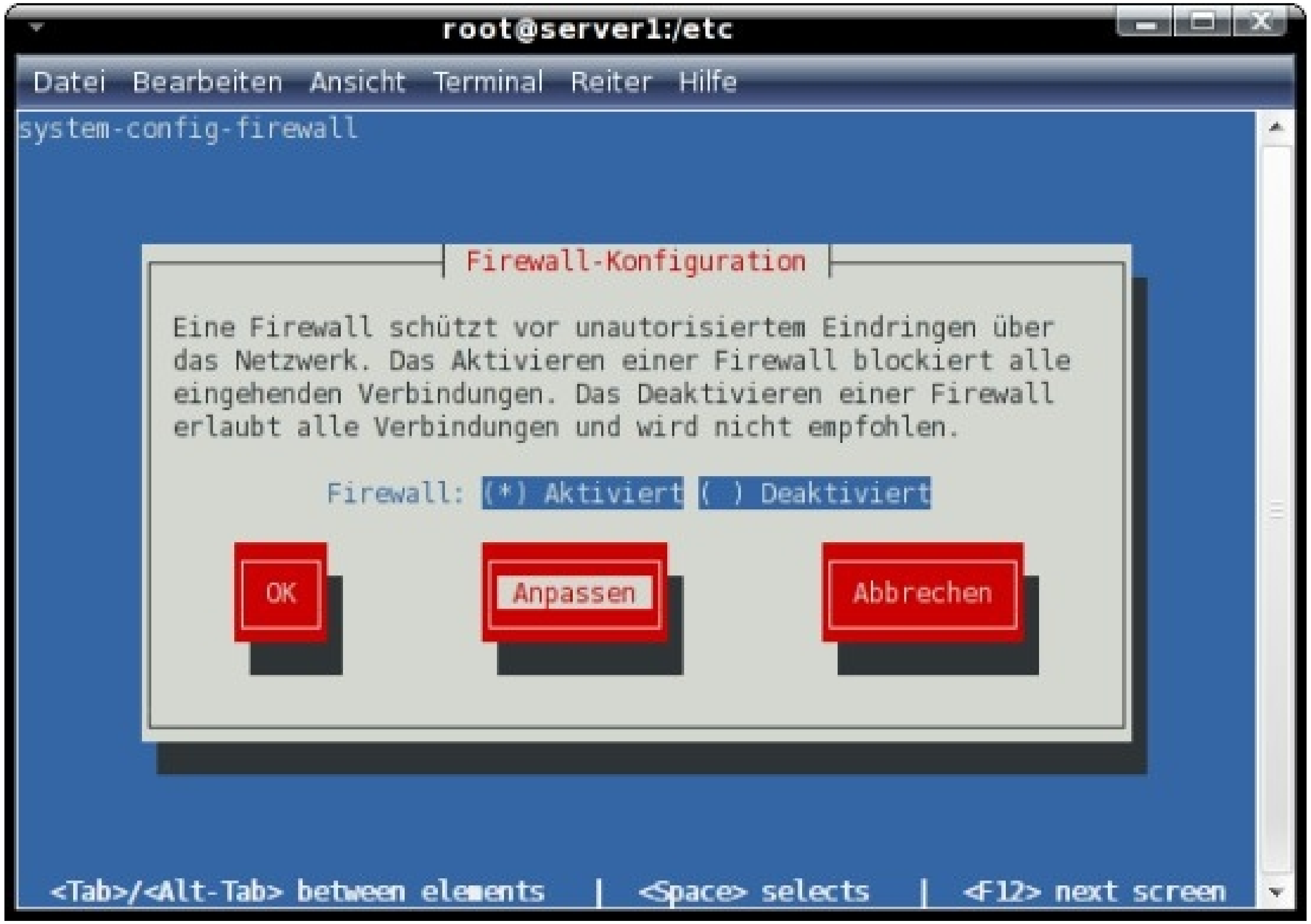
If the installed version is lower than 1.0.12-4.x you have to update to the new version. While I was writing this howto, the new version was only available in the updates-testing repository.

```
yum --enablerepo=updates-testing update system-config-firewall-tui
```

In order that the mysql servers are able to connect each other you have to open the port 3306 (tcp) on both systems.

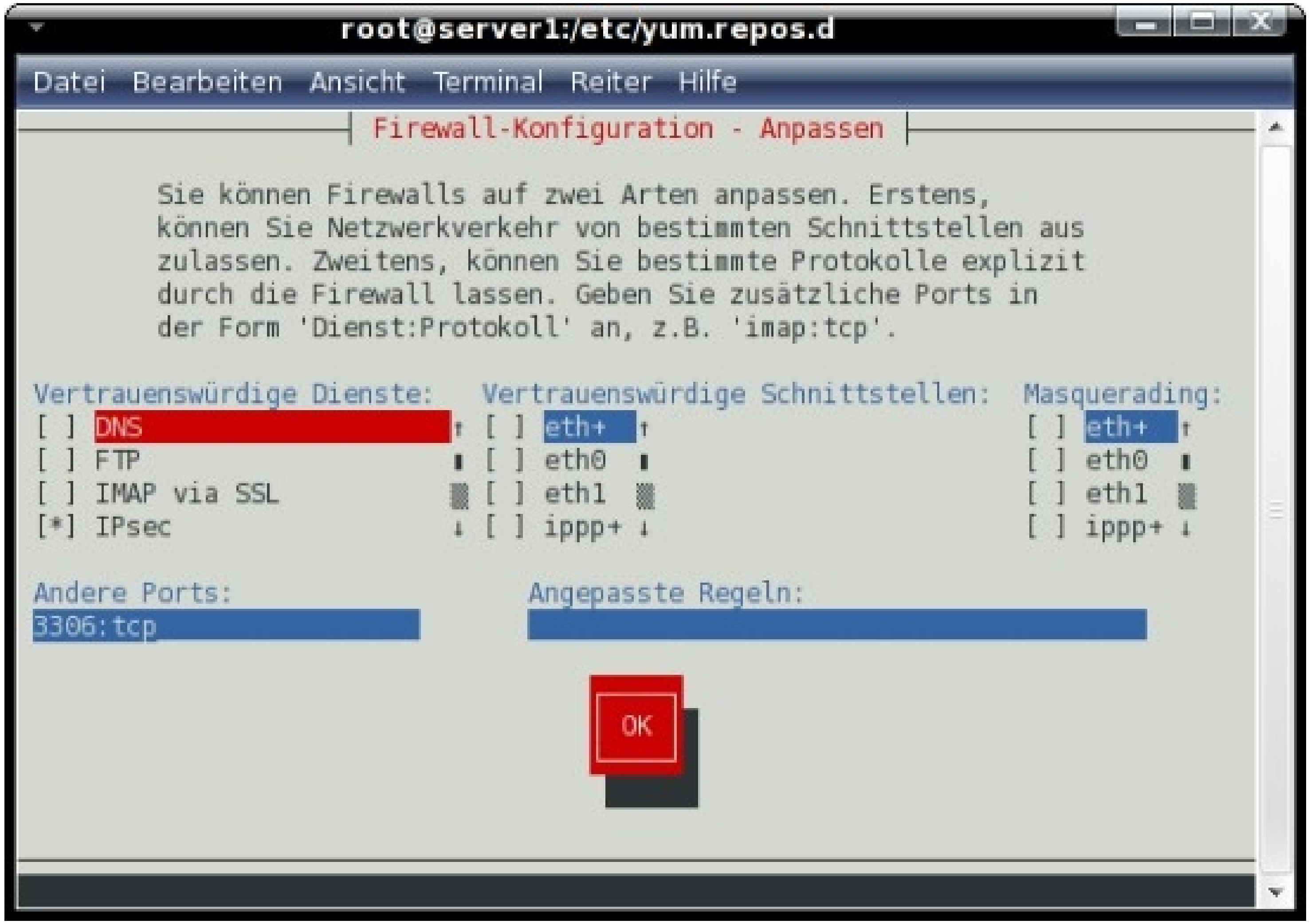
```
system-config-firewall
```

Click on "`Customize`".



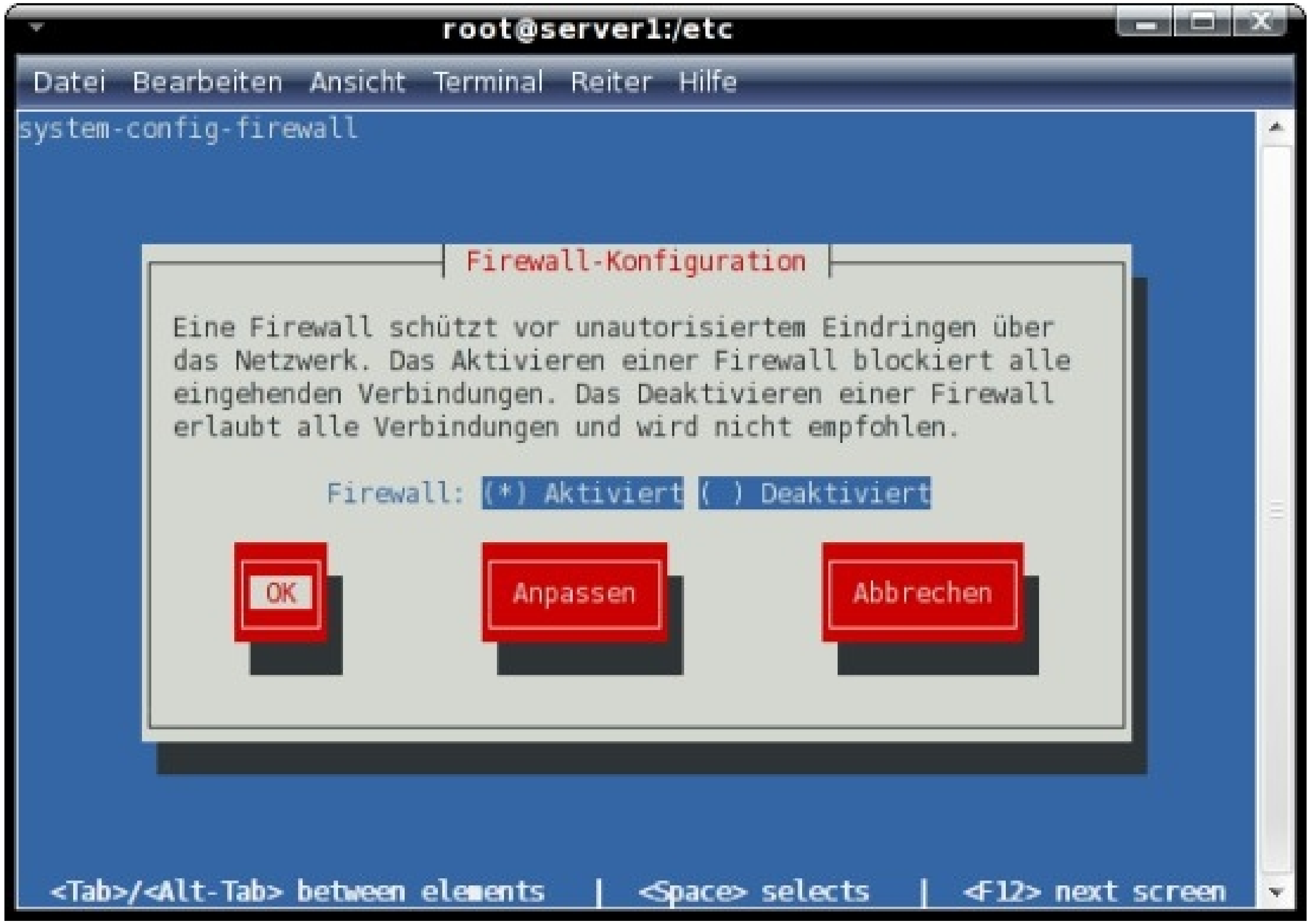
```
system-config-firewall

Firewall-Konfiguration

Eine Firewall schützt vor unautorisiertem Eindringen über
das Netzwerk. Das Aktivieren einer Firewall blockiert alle
eingehenden Verbindungen. Das Deaktivieren einer Firewall
erlaubt alle Verbindungen und wird nicht empfohlen.

Firewall: (*) Aktiviert ( ) Deaktiviert

OK    Anpassen    Abbrechen

<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen
```

Insert the MySQL-port into the section "`Other Ports`" as shown on the screenshot below and click on "`OK`" to save the settings.


root@server1:/etc/yum.repos.d
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
OK

```
Firewall-Konfiguration - Anpassen

        Sie können Firewalls auf zwei Arten anpassen. Erstens,
        können Sie Netzwerkverkehr von bestimmten Schnittstellen aus
        zulassen. Zweitens, können Sie bestimmte Protokolle explizit
        durch die Firewall lassen. Geben Sie zusätzliche Ports in
        der Form 'Dienst:Protokoll' an, z.B. 'imap:tcp'.

Vertrauenswürdige Dienste:  Vertrauenswürdige Schnittstellen:  Masquerading:
[ ] DNS                     [ ] eth+                           [ ] eth+
[ ] FTP                     [ ] eth0                           [ ] eth0
[ ] IMAP via SSL            [ ] eth1                           [ ] eth1
[*] IPsec                   [ ] ippp+                          [ ] ippp+

Andere Ports:                 Angepasste Regeln:
3306:tcp
```

Click on "*OK*".



```
system-config-firewall

Firewall-Konfiguration

Eine Firewall schützt vor unautorisiertem Eindringen über
das Netzwerk. Das Aktivieren einer Firewall blockiert alle
eingehenden Verbindungen. Das Deaktivieren einer Firewall
erlaubt alle Verbindungen und wird nicht empfohlen.

Firewall: (*) Aktiviert ( ) Deaktiviert

OK    Anpassen    Abbrechen

<Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

3.2 Log Directory On Both Systems

In order that the MySQL server is able to create log-files we have to create a directory and pass the ownership to MySQL.

```
mkdir /var/log/mysql/

chown mysql:mysql /var/log/mysql/
```

3.3 MySQL Configuration

In the next two steps we adjust the MySQL configuration on both systems for master-master replication.

3.3.1 System 1

```
vi /etc/my.cnf
```

Add the following lines to the section [mysqld]:

```
server-id = 1
replicate-same-server-id = 0
auto-increment-increment = 2
auto-increment-offset = 1

master-host = 192.168.0.200
master-user = slave1_user
master-password = %mysql_slaveuser_password%
master-connect-retry = 60
replicate-do-db = exampledb

log-bin = /var/log/mysql/mysql-bin.log
binlog-do-db = exampledb

relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index
```

```
expire_logs_days        = 10
max_binlog_size         = 500M
```

Afterwards restart the MySQL server.

```
/etc/init.d/mysqld restart
```

3.3.2 System 2

```
vi /etc/my.cnf
```

Add the following lines to the section [mysqld]:

```
server-id = 2
replicate-same-server-id = 0
auto-increment-increment = 2
auto-increment-offset = 2

master-host = 192.168.0.100
master-user = slave2_user
master-password = %mysql_slaveuser_password%
master-connect-retry = 60
replicate-do-db = exampledb

log-bin= /var/log/mysql/mysql-bin.log
binlog-do-db = exampledb

relay-log = /var/lib/mysql/slave-relay.log
relay-log-index = /var/lib/mysql/slave-relay-log.index

expire_logs_days        = 10
```

```
max_binlog_size         = 500M
```

Afterwards restart the MySQL server.

```
/etc/init.d/mysqld restart
```

3.4 Export MySQL Dump On System 1

Now we create a dump of the existing database and transfer it to system 2.

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

```
USE exampledb;

FLUSH TABLES WITH READ LOCK;

SHOW MASTER STATUS;
```

The output should look like this. Note down the file and the position - you'll need both later.

```
+------------------+----------+--------------------+------------------+
| File             | Position | Binlog_Do_DB       | Binlog_Ignore_DB |
+------------------+----------+--------------------+------------------+
| mysql-bin.000004 |       98 | exampledb,exampledb |                  |
+------------------+----------+--------------------+------------------+
1 row in set (0.00 sec)
```

Open a second terminal for system 1, create the dump and transfer it to system 2. Don't leave the MySQL-shell at this point - otherwise you'll loose the read-lock.

```
cd /tmp/
```

```
mysqldump -u root -p%mysql_root_password% --opt exampledb > sqldump.sql

scp sqldump.sql root@192.168.0.200:/tmp/
```

Afterwards close the second terminal and switch back to the first. Remove the read-lock and leave the MySQL-shell.

```
UNLOCK TABLES;

quit;
```

3.5 Import MySQL Dump On System 2

Time to import the database dump on system 2.

```
mysqladmin --user=root --password=%mysql_root_password% stop-slave

cd /tmp/

mysql -u root -p%mysql_root_password% exampledb < sqldump.sql
```

3.6 System 2 As Master

Now we need information about the master status on system 2.

```
mysql -u root -p

USE exampledb;

FLUSH TABLES WITH READ LOCK;
```

```
SHOW MASTER STATUS;
```

The output should look like this. Note down the file and the position - you'll need both later.

```
+------------------+----------+--------------------+------------------+
| File             | Position | Binlog_Do_DB       | Binlog_Ignore_DB |
+------------------+----------+--------------------+------------------+
| mysql-bin.000003 |      958 | exampledb,exampledb |                  |
+------------------+----------+--------------------+------------------+
1 row in set (0.00 sec)
```

Afterwards remove the read-lock.

```
UNLOCK TABLES;
```

At this point we're ready to become the master for system 1. Replace %mysql_slaveuser_password% with the password you choose and be sure that you replace the values for MASTER_LOG_FILE and MASTER_LOG_POS with the values that you noted down at step 3.4!

```
CHANGE MASTER TO MASTER_HOST='192.168.0.100', MASTER_USER='slave2_user', MASTER_PASSWORD='%mysql_slaveuser_password%', MASTER_LOG_FILE='mysql-bin.000004', MASTER_LOG_POS=98;
```

Now start the slave ...

```
START SLAVE;
```

... and take a look at the slave status. It's very important that both, `Slave_IO_Running` and `Slave_SQL_Running` are set to `Yes`. If they're not, something went wrong and you should take a look at the logs.

```
SHOW SLAVE STATUS;
```

```
+----------------------------------+---------------+-------------+-------------+---------------+------------------+---------------------+--------------------+---------------+----------------------+-----------------+------------------+--------------
-------+---------------------+--------------------+-----------------------+------------------------+----------------------------+------------+------------+-------------+--------------------+-----------------+-----------------+----------------+---
------------+--------------------+--------------------+--------------------+----------------+-------------------+----------------+----------------------+
| Slave_IO_State                   | Master_Host   | Master_User | Master_Port | Connect_Retry | Master_Log_File  | Read_Master_Log_Pos | Relay_Log_File     | Relay_Log_Pos | Relay_Master_Log_File |
Slave_IO_Running | Slave_SQL_Running | Replicate_Do_DB       | Replicate_Ignore_DB | Replicate_Do_Table | Replicate_Ignore_Table | Replicate_Wild_Do_Table | Replicate_Wild_Ignore_Table | Last_Errno |
Last_Error | Skip_Counter | Exec_Master_Log_Pos | Relay_Log_Space | Until_Condition | Until_Log_File | Until_Log_Pos | Master_SSL_Allowed | Master_SSL_CA_File | Master_SSL_CA_Path | Master_SSL_Cert
| Master_SSL_Cipher | Master_SSL_Key | Seconds_Behind_Master |
+----------------------------------+---------------+-------------+-------------+---------------+------------------+---------------------+--------------------+---------------+----------------------+-----------------+------------------+--------------
-------+---------------------+--------------------+-----------------------+------------------------+----------------------------+------------+------------+-------------+--------------------+-----------------+-----------------+----------------+---
------------+--------------------+--------------------+--------------------+----------------+-------------------+----------------+----------------------+
| Waiting for master to send event | 192.168.0.100 | slave2_user |        3306 |            60 | mysql-bin.000004 |                  98 | slave-relay.000002 |           235 | mysql-bin.000004      | Yes              | Yes               |
exampledb,exampledb |                     |                    |                        |                        |                             |          0 |            |            0 |                  98 |             235 | None            |                |             0 | No                 |
|                    |                    |                    |                |                   0 |
+----------------------------------+---------------+-------------+-------------+---------------+------------------+---------------------+--------------------+---------------+----------------------+-----------------+------------------+--------------
-------+---------------------+--------------------+-----------------------+------------------------+----------------------------+------------+------------+-------------+--------------------+-----------------+-----------------+----------------+---
------------+--------------------+--------------------+--------------------+----------------+-------------------+----------------+----------------------+
1 row in set (0.00 sec)
```

Afterwards leave the MySQL-shell.

```
quit;
```

3.7 System 1 As Master

Open a MySQL-shell on system 1 ...

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

... and stop the slave.

```
STOP SLAVE;
```

At this point we're ready to become the master for system 2. Replace %mysql_slaveuser_password% with the password you choose and be sure that you replace the values for MASTER_LOG_FILE and MASTER_LOG_POS with the values that you noted down at step 3.6!

```
CHANGE MASTER TO MASTER_HOST='192.168.0.200', MASTER_USER='slave1_user', MASTER_PASSWORD='%mysql_slaveuser_password%',
MASTER_LOG_FILE='mysql-bin.000003', MASTER_LOG_POS=958;
```

Now start the slave ...

```
START SLAVE;
```

... and take a look at the slave status. It's very important that both, `Slave_IO_Running` and `Slave_SQL_Running` are set to `Yes`. If they're not, something went wrong and you should take a look at the logs.

```
SHOW SLAVE STATUS;
```

```
+----------------------------------+---------------+-------------+-------------+---------------+-----------------+--------------------+-------------------+--------------+----------------------+-----------------+------------------+-------------
-------+--------------------+-------------------+-----------------------+-----------------------+----------------------------+------------+-----------+-------------+--------------------+----------------+----------------+----------------+---
-----------+--------------------+-------------------+-------------------+----------------+------------------+----------------+----------------------+
| Slave_IO_State                   | Master_Host   | Master_User | Master_Port | Connect_Retry | Master_Log_File | Read_Master_Log_Pos | Relay_Log_File     | Relay_Log_Pos | Relay_Master_Log_File |
Slave_IO_Running | Slave_SQL_Running | Replicate_Do_DB     | Replicate_Ignore_DB | Replicate_Do_Table | Replicate_Ignore_Table | Replicate_Wild_Do_Table | Replicate_Wild_Ignore_Table | Last_Errno |
Last_Error | Skip_Counter | Exec_Master_Log_Pos | Relay_Log_Space | Until_Condition | Until_Log_File | Until_Log_Pos | Master_SSL_Allowed | Master_SSL_CA_File | Master_SSL_CA_Path | Master_SSL_Cert
| Master_SSL_Cipher | Master_SSL_Key | Seconds_Behind_Master |
+----------------------------------+---------------+-------------+-------------+---------------+-----------------+--------------------+-------------------+--------------+----------------------+-----------------+------------------+-------------
-------+--------------------+-------------------+-----------------------+-----------------------+----------------------------+------------+-----------+-------------+--------------------+----------------+----------------+----------------+---
-----------+--------------------+-------------------+-------------------+----------------+------------------+----------------+----------------------+
| Waiting for master to send event | 192.168.0.200 | slave1_user |        3306 |           60 | mysql-bin.000003 |                958 | slave-relay.000002 |           235 | mysql-bin.000003      | Yes              | Yes              |
exampledb,exampledb |                  |                 |                    |                      |                        |          0 |           |          0 |              958 |          235 | None           |              |          0 | No                   |
 |                  |               |                 |              |                 0 |
+----------------------------------+---------------+-------------+-------------+---------------+------------------+---------------------+--------------------+---------------+----------------------+------------------+-------------------+--------------
-------+---------------------+--------------------+------------------------+------------------------+----------------------------+------------+------------+-------------+---------------------+-----------------+-----------------+-----------------+---
------------+--------------------+--------------------+--------------------+-----------------+-------------------+----------------+-----------------------+
1 row in set (0.00 sec)
```

Afterwards leave the MySQL shell.

```
quit;
```

If all went ok, the master-master replication is working now. Check your logs on both systems if you encounter problems.

4 Links

- Fedora: **http://fedoraproject.org/**
- MySQL: **http://www.mysql.com/**