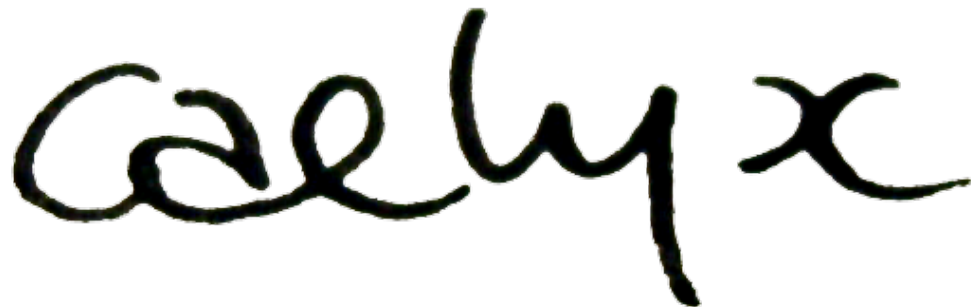


MYSQL REPLICATION

caelyx.networks quick notes

A large, handwritten signature in black ink that reads "caelyx". The letters are cursive and fluid, with the 'c' and 'a' being particularly prominent.

Simon Brown
projects at caelyx dot net
22 February, 2005

INTRODUCTION

These notes are derived from personal experience and section 6.4 of the MySQL Technical Reference¹. Neither Simon Brown, nor any other party will be liable for any use of this information. It is provided as is, and should be considered non-prescriptive friendly advice; your mileage may vary. Consequently, while all reasonable care an effort has been taken to ensure the correctness, completeness, and reliability of this information, no warranty is provided nor assurances given.

You should always carefully test any advice in a suitable non-production or staging environment prior to using it in production or critical environments. You should always backup everything (preferably a drive image, but separate system and data backups may be appropriate) before attempting any significant change to a database server.

INITIAL CONDITIONS

The initial conditions for my implementation were:

- A master database server running an instance of MySQL version 4.1.x with pre-existing user accounts and data;
- A secondary server, with a fresh MySQL 4.1.x installed & stopped, and a hardened OS; and
- I wanted to replicate all databases and tables.

If your assumptions or conditions vary, you may need to proceed differently. In particular, you can exclude some databases, specifically include databases, and choose which tables to replicate.

Note that I'm not going to explain why or when to replicate, merely the base elements of how. I also don't explain how to failover (although I hear good things about Linux-HA), nor how to recover from a failed master. All of these topic are covered elsewhere, by more experienced people.

STEPS

1. Turn on binary logging in your master server; the option lives in `/etc/my.cnf`. Add the following lines if they don't already exist:

```
log-bin
server-id=1
```

¹ Available on the web at <http://dev.mysql.com/doc/>

2. Add a replication account to the master server;


```
GRANT REPLICATION SLAVE ON *.* TO
G_REPL_01@slave.example.com IDENTIFIED BY 'password' ;
```
3. If the master server already had binary logging on, save the output of:


```
SHOW MASTER STATUS ;
```
4. Shut down the master server:


```
mysqldadmin [-u <dba>] shutdown
```
5. Take a snapshot of the master's database directory. The MySQL guide recommends you do this by snarfing the files off the disk:

```
cd /
tar jcvf /tmp/mysql-snapshot.tar.bz2 /var/lib/mysql2
```

6. Restart the master database server:


```
master# service mysqld start
```
7. Copy the snapshot to the slave host. I used scp with user keys:


```
scp /tmp/mysql-snapshot.tar.bz2 slave.example.com:/tmp/
```
8. On the slave, unpack the snapshot, overwriting MySQL's datastore:

```
cd /
tar jxvf /tmp/mysql-snapshot.tar.bz2
```

9. Ensure that `/etc/my.cnf` on the slave has a value set for `server-id`, and that it is different to the master's value. In my case, this involved adding the following line:


```
server-id=2
```
10. Start the slave;


```
slave# service mysqld start
```

11. Start replication on the slave:

Replace `<logfile_name>` and `<logfile_position>` with the details you got from the `SHOW MASTER STATUS` command. If the MySQL master wasn't binary logging, use the values `"` and `4`, respectively.

```
CHANGE MASTER TO
  MASTER_HOST='master.example.com' ,
  MASTER_USER='G_REPL_01' ,
  MASTER_PASSWORD='password' ,
  MASTER_LOG_FILE='<logfile_name>' ,
  MASTER_LOG_POS=<logfile_position>;
```

² This is the default for the RedHat MySQL binaries, AFAIK. Modify this path as appropriate.

12. Start the slave replicating:

```
START SLAVE;
```

You should now be done and replicating; you can check this by logging into the master server, and listing the connections, looking for the slave user. Alternatively, log into the slave, and check to see that updates are being made -- I added a a new user to the master, and checked that it appeared and worked on the slave.