

# High Availability: Can I replicate my own MySQL database using the failover mechanism supplied by Bright?

Yes.

All the MySQL databases can be replicated between the active and passive, not just CMDaemon and Slurm. (Have a look in the Admin Manual for how BCM does replication/cloning of the CMDaemon+Slurm databases with the dbreclone utility).

The following procedure replicates all the databases:

What you will get:

- All database changes done on the active master node will find themselves on the passive
- All database changes done on the passive will corrupt the database (including cmdaemon database)

The procedure currently works for Bright Cluster Manager 5.2 and 6.0.

## Procedure

This requires manual labour (mysqld and cmd will be stopped and started on both head-nodes, all data in your database on the passive will be lost!):

- Create the XX database.

(XX will be used as the name of your database in the remainder of the text, and belongs to a new mysql user, with a new password (stored in userXX, passXX in the changed script below)).

- Make sure the database and new mysql user exists on both headnodes

- Edit /etc/my.cnf (on both head nodes, do not scp!!!)

- Add the lines: (below other similar lines)

# High Availability: Can I replicate my own MySQL database using the failover mechanism supplied by Bright?

```
binlog-do-db=<your-db-name>  
replicate-do-db=<your-db-name>
```

- Edit /cm/local/apps/cmd/scripts/clonemysqloffailover.sh (on active head node, scp to passive after edit)

Add the following lines (below if [ -f \$conf ]; then)

```
tmpXX=/tmp/.dump.$$$.sql  
userXX=...  
passXX=...
```

Next, dump the database to the temporary file (place these lines below similar code for cmdaemon database):

```
mysqldump -u$userXX -p$passXX XX > $tmpXX  
scp $tmpXX $failover:$tmpXX
```

Then, import the database into mysql on the passive (insert the XX related code before the echo \"RESET MASTER... command):

```
ssh $failover "mysql -u$user -p$pass cmdaemon < $tmp; rm $tmp;  
*mysql -u$userXX -p$passXX XX < $tmpXX; rm $tmpXX*; echo \"RESET MASTER;  
STOP SLAVE; RESET SLAVE; CHANGE MASTER TO MASTER_HOST=\" | mysql -u$user  
-p$pass cmdaemon"
```

Finally, insert statement to remove the temporary file:

```
rm $tmp *$tmpXX*
```

- run cmha dbreclone

All done. Your database will now be synced and be kept in sync on both head-nodes, even after a failover.

Unique solution ID: #1060

Author: Frank Furter

Last update: 2013-08-07 16:01