

Linuxha.net
Step-by-step Guide:
MySQL

Introduction

The purpose of this document is to suggest procedures for creating a clustered MySQL configuration with Linuxha.net. These procedures were tested using the following configuration:

- Fedora Core 5
- DRBD version 0.7.20
- Linuxha.net version 1.2
- MySQL version 5.0.24-standard-log

The following conventions are used throughout this document:

# ls	Single-line command entered as root user.
# useradd \ > --home-dir /home2/xyz \ > --gid abc \ > xyz	Multi-line command entered as root user.
\$ ls	Single-line command entered as non-root user
\$ cut \ > --delimiter=":" \ > --fields=1,5 \ / etc/passwd	Multi-line command entered as non-root user.
127.0.0.1 localhost 192.168.1.32 fc5s1 192.168.1.33 fc5s2 192.168.100.32 fc5s1b 192.168.100.33 fc5s2b	Command output or file contents

User Account Creation

The MySQL processes will run with the **mysql** user credentials. The **mysql** user and group must exist on both nodes and have the same user and groups ids.

All commands in this section are executed as **root**.

1. Check whether the **mysql** user exists and that the user and group ids are the same on both nodes.

```
# grep mysql /etc/passwd
```

If there is no result then continue to step 2. Otherwise, compare the user and group ids on both nodes. These are the third and fourth fields respectively of the user account record as shown below:

```
mysql:x:502:502:MySQL server:/var/lib/mysql:/bin/bash
```

If the user and group ids are not the same on both nodes, proceed to step 6. Otherwise continue to the next section.

2. Identify largest available group id.

```
# expr `cut --delimiter=":" --fields=3 /etc/group | \  
> grep --regexp="^[5-9][0-9][0-9]$" | sort --numeric-sort|tail -1` + 1
```

3. Create **mysql** group and assing the larger of the group ids identified in step 2, e.g. 502.

```
# groupadd --gid 502 mysql
```

4. Identify largest available user id

```
# expr `cut --delimiter=":" --fields=3 /etc/passwd | \
> grep --regexp="^[5-9][0-9][0-9]$" | sort --numeric-sort|tail -1` + 1
```

5. Create **mysql** user and assign the larger of the user ids identified in step 4, e.g. 502.

```
# useradd --home-dir /var/lib/mysql -M \
> --comment "MySQL server" --gid mysql --uid 502 \
> mysql
```

Proceed to the next section.

6. If the **mysql** group ids are different, change the group id on **one of the nodes** to be the same as the other, ensure that the new group id is not already in use.

```
# grep 504 /etc/group          # check that group id is not in use
# groupmod -g 504 mysql       # change group id to 504
```

7. If the **mysql** user ids are different, change the user id on **one of the nodes** to be the same as the other, ensure that the new user id is not already in use.

```
# grep 504 /etc/passwd        # check that user id is not in use
# usermod --uid 504 --gid mysql mysql # change user id to 504
```

8. On the node where the user and group ids were changed, change the user and group ownership of files owned by **mysql**.

The following commands will find all files whose user / group ownership is 502 and change their user / group ownership to **mysql**.

```
# find /usr -gid 502 -exec chgrp mysql {} \;
# find /usr -uid 502 -exec chown mysql {} \;
```

Installation

This section provides basic instructions for downloading and installing MySQL version 5.0 from binary RPM or source. Instructions for installing or upgrading using other package formats are beyond the scope of this guide.

All commands are executed on both nodes as **root**, unless otherwise indicated.

1. Check whether MySQL is installed, by executing

```
# find /usr -name mysqld -print
```

If a result similar to the following is displayed, then MySQL is installed and you may proceed to step 4.

```
/usr/sbin/mysqld
```

2. Download MySQL server and client i386 RPM's as shown, these are **single-line** commands:

```
# wget http://www.mysql.org/get/Downloads/MySQL-5.0/MySQL-server-5.0.24-0.i386.rpm/from/http://www.mirrorservice.org/sites/ftp.mysql.com/
# wget http://www.mysql.org/get/Downloads/MySQL-5.0/MySQL-client-5.0.24-0.i386.rpm/from/http://www.mirrorservice.org/sites/ftp.mysql.com/
```

3. Install MySQL server and client.

```
# rpm --hash --install MySQL-server-5.0.24-0.i386.rpm
# rpm --hash --install MySQL-client-5.0.24-0.i386.rpm
```

4. Configure firewall

If the firewall is running, then it should be configured to accept connections on port 3306, the default MySQL port.

Insert the following line into **/etc/sysconfig/iptables**.

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
```

Restart the firewall

```
# /etc/init.d/iptables/restart
```

5. Delete the MySQL automatic startup and shutdown scripts.

```
# rm /etc/rc3.d/*mysql /etc/rc0.d/*mysql
```

6. Start MySQL database server

```
# /etc/init.d/mysql start
```

7. On the **primary node**, configure MySQL to accept root user connections from both cluster nodes (*fc5s1* and *fc5s2*) and the virtual IP address (*192.168.1.45*).

```
# mysql
mysql> use mysql;
mysql> grant all privileges on *.* to 'root'@'fc5s1';
mysql> grant all privileges on *.* to 'root'@'fc5s2';
mysql> grant all privileges on *.* to 'root'@'192.168.1.45';
mysql> exit
```

Directories / Files for Replication

The directory where MySQL stores its database files will be replicated, this directory can be determined by executing:

```
# mysqladmin variables | grep datadir
```

which should display a result similar to the following:

```
| datadir | /var/lib/mysql/ |
```

In this example, the MySQL data directory is **/var/lib/mysql**.

Create Replicated File System

All commands in this section are executed on both nodes as **root**. Steps 5 to 6 may be omitted on the secondary node.

1. Use **fdisk** to create a partition (*/dev/sda6*) large enough to accommodate all databases.

2. Initialize partition

```
# pvcreate /dev/sda6
```

3. Create application volume group (*mysqlvg*)

```
# vgcreate mysqlvg /dev/sda6
```

4. Create mount point for replicated file system

```
# mkdir /mysql
```

5. Create logical volume (*data1v*) to store databases.

```
# lvcreate --size 512M --name data1v mysqlvg
```

6. Create file system on *data1v*.

```
# mkfs -t ext3 /dev/mysqlvg/data1v
```

Populate Replicated File System

All instructions are to be carried out as **root** on the primary node, unless otherwise indicated.

1. Shut down MySQL database server (**both nodes**).

```
# /etc/init.d/mysql stop
```

2. Mount logical volume

```
# mount -t ext3 /dev/mysqlvg/data1v /mysql
```

3. Create directory structure

```
# mkdir /mysql/data
# chown mysql:mysql /mysql/data
# mkdir /mysql/scripts
# touch /mysql/scripts/startapp /mysql/scripts/stopapp
# chmod u+x /mysql/scripts/*
```

4. Copy contents of data directory to logical volume.

```
# cd /var/lib/mysql
# find . -print | cpio --pass-through /mysql/data
```

Configuration

All commands are executed as **root** on the primary node, except where indicated otherwise.

1. Create directory to store process id file (**both nodes**).

```
# mkdir /var/run/mysql
# chown mysql:mysql /var/run/mysql
```

2. Insert the following into **/mysql/scripts/startapp**:

```
mysqld_safe \
  --datadir=/mysql/data \
  --pid-file=/var/run/mysql/mysql.pid \
  --log-error=/var/log/mysql.log > /dev/null 2>&1 &
```

3. Insert the following into **/mysql/scripts/stopapp**:

```
if [ -f /var/run/mysql/mysql.pid ]
then kill `cat /var/run/mysql/mysql.pid`
fi
```

Build Application

The commands in this section are executed as **root** on the primary node.

1. Create directory **/etc/cluster/mysql**.

```
# mkdir /etc/cluster/mysql
```

2. Create **/etc/cluster/mysql/appconf.xml** as shown:

```
<?xml version="1.0"?>
<appconf>
  <global>
    <version>0.1</version>
    <name>mysql</name>
    <takeover>normal</takeover>
    <syncrate>2000</syncrate>
    <preferred_node>fc5s1</preferred_node>
  </global>

  <networks>
    <network net="main" ip="192.168.1.45" netmask="255.255.255.0" />
  </networks>
  <vg>
    <name>mysqlvg</name>
    <type>filesystems</type>
  </vg>
  <application>
    <startscript>/mysql/scripts/startapp</startscript>
    <stopscript>/mysql/scripts/stopapp</stopscript>
    <maxstoptime>10</maxstoptime>
    <maxstarttime>20</maxstarttime>
  </application>
</appconf>
```

The bolded values are configuration-specific, as described in the following table:

Entry	Value
global/preferred_node	Host name of one of the nodes in the cluster, or LEAST_CPU_LOAD
networks/network.net	Same as one of <i>node/network.name</i> in /etc/cluster/clconf.xml
networks/network.ip	Virtual IP address of application
networks/network.netmask	Netmask corresponding to virtual IP address
vg/name	Name of application volume group

3. Create **/etc/cluster/mysql/lems.local.xml** as shown:

```
<?xml version="1.0"?>
<lems_config>
  <globals modules="/sbin/cluster/lems/modules"
    programs="/sbin/cluster/lems/programs"
    logs="/var/log/cluster/lems"
  />
  <check>
    <name>flag_check</name>
    <type>internal</type>
    <module>flag_check mysql</module>
    <interval>5</interval>
    <action_list>
      <action rc="0" action="NOP"/>
    </action_list>
  </check>
</lems_config>
```

```

        <action rc="1" action="%RCDATA%"/>
        <action rc="2" action="ABORT"/>
    </action_list>
</check>
<check>
    <name>mysqld</name>
    <type>internal</type>
    <module>procmon /etc/cluster/mysql/mysqld.xml</module>
    <interval>10</interval>
    <action_list>
        <action rc="0" action="NOP"/>
        <action rc="1" action="STOP"/>
        <action rc="2" action="FAILOVER"/>
    </action_list>
</check>
<check>
    <name>ip</name>
    <type>internal</type>
    <module>ip_module mysql</module>
    <interval>10</interval>
    <action_list>
        <action rc="0" action="NOP"/>
        <action rc="1" action="RUN move_ip"/>
        <action rc="2" action="STOP"/>
    </action_list>
</check>
<check>
    <name>fsmonitor</name>
    <type>internal</type>
    <module>fsmon mysql</module>
    <interval>10</interval>
    <action_list>
        <action rc="0" action="NOP"/>
        <action rc="1" action="PAUSE 30"/>
        <action rc="2" action="STOP"/>
        <action rc="3" action="FAILOVER"/>
        <action rc="10" action="PAUSE 60"/>
    </action_list>
</check>
</lems_config>

```

4. Create **/etc/cluster/mysql/mysqld.xml** as shown:

```

<?xml version="1.0"?>
<procmon>
    <global>
        <logdir>/var/log/cluster</logdir>
        <restarts>5</restarts>
        <resetwindow>3600</resetwindow>
        <restartcmd>
            /mysql/scripts/stopapp ; /mysql/scripts/startapp
        </restartcmd>
    </global>
    <process>
        <label>MySQL Parent Process</label>
        <user>root</user>
        <process_string>/bin/sh /usr/bin/mysqld_safe*</process_string>
        <min_count>1</min_count>
        <max_count>1</max_count>
    </process>
</process>

```

```

<label>MySQL Database Server</label>
<user>mysql</user>
<process_string>/usr/sbin/mysqld*</process_string>
<min_count>2</min_count>
<max_count>30</max_count>
</process>
</procmon>

```

5. Verify LEMS configuration by executing:

```

# lems.pl --config /etc/cluster/mysql/lems.local.xml \
> --application mysql --verbose --check --file /dev/tty

```

If successful, a result similar to the following will be displayed.

```

INFO 22/08/2006 16:00:40 Using modules from : /sbin/cluster/lems/modules
INFO 22/08/2006 16:00:40 Using programs from : /sbin/cluster/lems/programs
INFO 22/08/2006 16:00:40 Writing logs to : /var/log/cluster/lems
INFO 22/08/2006 16:00:40 Listening on port : 9904
INFO 22/08/2006 16:00:40 Global initialisation complete.
INFO 22/08/2006 16:00:40 Started local server on port 9904
INFO 22/08/2006 16:00:40 Validating monitor entry ip...
INFO 22/08/2006 16:00:40 Validated monitor entry ip successfully.
INFO 22/08/2006 16:00:40 Validating monitor entry fsmonitor...
INFO 22/08/2006 16:00:40 Validated monitor entry fsmonitor successfully.
INFO 22/08/2006 16:00:40 Validating monitor entry flag_check...
INFO 22/08/2006 16:00:40 Validated monitor entry flag_check successfully.
INFO 22/08/2006 16:00:40 Validating monitor entry mysqld...
INFO 22/08/2006 16:00:40 Validated monitor entry mysqld successfully.
INFO 22/08/2006 16:00:40 Check mode - transferring validated config to remote node.
INFO 22/08/2006 16:00:40 Configuration transferred successfully.
INFO 22/08/2006 16:00:40 Calculated a check interval of 2.5 seconds.

```

6. Build the application.

```

# clbuildapp --application mysql --sync

```

The output of a successful build is shown below:

```

INFO 22/08/2006 16:03:57 Backups directory defaulted to /clbackup
INFO 22/08/2006 16:03:57
INFO 22/08/2006 16:03:57 Validation of Application 'mysql' started.
INFO 22/08/2006 16:03:57 ['var/log/cluster/build/mysql-check-300608221603.log']
INFO 22/08/2006 16:03:58 Initial Validation of Application successful.
INFO 22/08/2006 16:03:58
INFO 22/08/2006 16:03:58 NOTE: Build of new application is being performed.
INFO 22/08/2006 16:03:58
INFO 22/08/2006 16:03:58 Host Environment Validation started.
INFO 22/08/2006 16:03:58 ['var/log/cluster/build/mysql-envcheck-300608221603.log']
INFO 22/08/2006 16:04:03 Host Environment Validation successful.
INFO 22/08/2006 16:04:03
INFO 22/08/2006 16:04:03 Cluster state : RUNNING
INFO 22/08/2006 16:04:03 Application state: UNDEFINED
INFO 22/08/2006 16:04:03
INFO 22/08/2006 16:04:03 Volume Group Configuration started.
INFO 22/08/2006 16:04:03 ['var/log/cluster/build/mysql-lvm-300608221603.log']
INFO 22/08/2006 16:04:08 Volume Group Configuration successful.
INFO 22/08/2006 16:04:08
INFO 22/08/2006 16:04:08 Application Resource Allocation started.
INFO 22/08/2006 16:04:08 ['var/log/cluster/build/mysql-build-300608221603.log']
INFO 22/08/2006 16:04:15 Application Resource Allocation successful.
INFO 22/08/2006 16:04:15
INFO 22/08/2006 16:04:15 Application Data Synchronisation started.
INFO 22/08/2006 16:04:15 ['var/log/cluster/build/mysql-syncdata-300608221603.log']
Storage Syncing: 256Mb/ 0Mb [0.3 % Complete]
Storage Syncing: 0Mb/ 0Mb [100 % Complete]
INFO 22/08/2006 16:06:37 Application Data Synchronisation successful.
INFO 22/08/2006 16:06:37

```

Run Application

The commands in this section are executed as **root** on either node except where indicated.

1. If necessary, form the cluster

```
# clform
```

2. Start **mysql** on the primary node (*fc5s1*).

```
# clrunapp --application mysql --node fc5s1
```

3. Verify state of **mysql** using **clstat**.

```
# clstat --application mysql
```

A result similar to the following will be returned if **mysql** is running:

```
Cluster: cluster1 - UP

Application      Node      State  Runnig  Monitor  Stale  Fail-over?
mysql           fc5s1    STARTED  0:00:01  Running    0      Yes

File Systems

Mount Point      Valid  Type    State  % Complete  Completion
/mysql           both   drbd    Sync

Process Monitors

Name    Status  Restarts  Current  Reset at
mysqld  Running    5         0        N/A

General Monitors

Type      Name      Status
Flag Check  flag_check  Running
FS Monitor  fsmonitor  Running
IP Monitor  ip         Running
```

4. On the node where **mysql** is running, list the **mysqld** processes by executing:

```
# ps -u root -f | grep mysqld | grep -v grep; \
# ps -u mysql -f | tail -1
```

A process list similar to the following should be returned:

```
root  4755    1  0  22:09 pts/1  00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir
=/mysql/data --pid-file=/var/run/mysql/mysql.pid --log-error=/var/log/mysql.log
mysql 4801 4792  0  22:09 pts/1  00:00:00 /usr/sbin/mysqld --basedir=/ --datadir
=/mysql/data --user=mysql --pid-file=/var/run/mysql/mysql.pid --skip-locking --port
=3306 --socket=/var/lib/mysql/mysql.sock
```

5. Verify connection to MySQL server on the cluster IP address (*192.168.1.45*) from either node as follows:

```
# mysql --host 192.168.1.45 --user root << EOF
> use mysql;
> show tables;
> exit
> EOF
```

If the MySQL database server is running a result similar to the following will be returned:

```
Tables_in_mysql
columns_priv
db
func
help_category
help_keyword
help_relation
help_topic
host
proc
procs_priv
tables_priv
time_zone
time_zone_leap_second
time_zone_name
time_zone_transition
time_zone_transition_type
user
```

6. On the node where **mysql** is running, stop it by executing:

```
# clhaltapp --application mysql
```

7. Start **mysql** on the secondary node (*fc5s2*).

```
# clrunapp --application mysql --node fc5s2
```

8. Repeat steps 3 to 6 to test **mysql** on the secondary node.