
OKLAHOMA CITY • OPEN SOURCE DATABASES •

OKC Open Source Database User Group

Sponsored By...

Meetup Mission

What is this even about?!

- Champion Open Source database usage in OKC
- Discuss various open source RDBMS and NoSQL technologies
- Help to grow the local database ecosystem

Highly Available MySQL

Base Installation

- By default, MySQL is a single node deployment
- Some flavors include:
  - MySQL Community (Free / Open Source)
  - MySQL Enterprise ($)
  - Percona Server (Free / Open Source)
  - MariaDB (Free / Open Source)
- Easiest installation is RPM/apt-get package installation

Note: For this demo, we'll be using base CentOS 7 and Percona Server 5.7

Base Installation - Steps

Let's install Percona Server...

1. Install Percona repo
    a. `yum install http://www.percona.com/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm`
2. Install Percona Server packages
    a. `yum install Percona-Server-server-57 Percona-Server-client-57 Percona-Server-shared-57 Percona-Server-shared-compat-57 Percona-Server-devel-57`
3. Start MySQL and secure the installation
    a. `service mysql start`
    b. `grep password /var/log/mysqld.log`
    c. `mysql_secure_installation`

So... what about that HA part?!

Hopefully, you noticed we only set up one instance of MySQL...

There is a very important saying popularized by the Navy Seals:

“Two is one and one is none.”

Setup a second server

Follow the same Base Installation steps on another server...

Voila!

Now we have two servers!!

So... what next?

Setup Master/Slave Replication

There are a few types of replication, for simplicity, we are just going to use good ole' fashioned asynchronous replication:

1. Enable binlog on master and set unique server ID (edit /etc/my.cnf)

```
[mysqld]

server-id = 100
log-bin = mysql-bin
log-slave-updates
```

2. Restart master with new updated config
   a. `service mysql restart`
3. Repeat config/restart on slave, but with a DIFFERENT server-id

Start Replication

1. Add a replication user on the master node
   a. `GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%' IDENTIFIED BY "a352qINva23asda##";`
   b. `FLUSH PRIVILEGES;`
2. Start replication on the slave
   a. `CHANGE MASTER TO MASTER_HOST = "192.168.56.120", MASTER_USER = "repl", MASTER_PASSWORD = "a352qINva23asda##", MASTER_LOG_FILE = "mysql-bin.000001", MASTER_LOG_POS = 4;`
   b. `START SLAVE;`
   c. `SHOW SLAVE STATUS\G`

Verify Replication

At this point, the last command shows if replication is working, but I (and you) should always validate things...

1. Create a test table on the master (run on master)
   a. `CREATE DATABASE mydb;`
   b. `USE mydb;`
   c. `CREATE TABLE t1 (id int unsigned auto_increment primary key, val varchar(255));`
2. Verify on the slave
   a. `USE mydb;`
   b. `SHOW TABLES;`
3. ... and look for the table you just created on the master

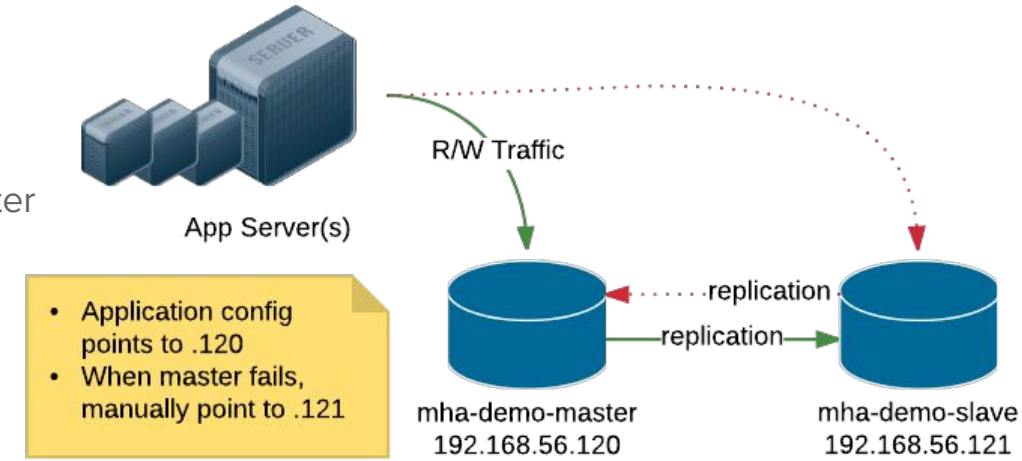
Huzzah!

Replication is working!

Let’s see where we are now...

Current Architecture

- Standard Master / Slave
- Application config
  - Database IP is master IP
  - Reads/Writes against the master
- If the master dies?
  - Update application config
  - Point to slave server



Still very manual, but at least when (not if) the master dies, you have redundancy...

Let's up the anti...

Automation and tools can greatly simplify our lives as DBAs (or sysadmins or whatever firefighting hat you are wearing today)

- Master High Availability Tools for MySQL (MHA)
  - https://code.google.com/archive/p/mysql-master-ha/
  - https://drive.google.com/drive/folders/0B1lu97m8-haWeHdGWXp0YVVUSlk
  - We'll download the most recent EL7 build from ^
- Basic process
  - Add a new, monitor node to the mix
  - Install mha-node on all boxes (mysql servers and monitor)
  - Install mha-manager on monitor host
  - Set up access between all the nodes (linux / MySQL)
  - Start the monitor with a newly configured virtual IP

Installing MHA Packages

Start by installing the mha-node package on all nodes and the mha-manager on the new monitor host:

- On all nodes, install the mha-node package
  - yum localinstall mha4mysql-node-0.57-0.el7.noarch.rpm
- On the manager node, install EPEL repo and mha-manager package
  - yum -y install epel-release
  - yum localinstall mha4mysql-manager-0.57-0.el7.noarch.rpm

Setup MHA User: Manager Node

1. On the manager node, add the mha user (as root)
   a. `adduser mha`
   b. `passwd -f -u mha`
2. Create public/private key for mha user
   a. `su -l mha`
   b. `ssh-keygen -t rsa`
   c. Save contents of **.ssh/id_rsa**, **.ssh/id_rsa.pub**
   d. Update **.ssh/authorized_keys** with new pubkey
3. Return to root user, create mysql group, add mha user
   a. `exit`
   b. `groupadd mysql`
   c. `usermod -a -G mysql mha`

Setup MHA User: Other Nodes

- Add the mha user and add to mysql group (already on nodes)
  - `adduser mha`
  - `passwd -f -u mha`
  - `usermod -a -G mysql mha`
- Setup the VIP sudo privileges for mha user
  - `vi /etc/sudoers.d/mha_sudo (paste contents below)`
  - `chmod 440 /etc/sudoers.d/mha_sudo`

```
Defaults:mha !requiretty

Cmnd_Alias VIP_MGMT = /sbin/ip
mha     ALL=(root)      NOPASSWD: VIP_MGMT
```

Setup MHA User: Other Nodes (cont)

- Setup the passwordless SSH for mha user
  - `su -l mha`
  - `mkdir .ssh`
  - `chmod 700 .ssh/`
  - `cd .ssh/`
  - `vi id_rsa (paste private key from earlier)`
  - `vi id_rsa.pub (paste public key from earlier)`
  - `vi authorized_keys (paste public key from earlier)`
  - `chmod 600 authorized_keys`
  - `chmod 400 id_rsa`

Setup Log Directory on ALL nodes

- Create working directory on mysql nodes and monitor
  - `mkdir /var/log/masterha`
  - `chown mha:mysql /var/log/masterha`
  - `chmod 755 /var/log/masterha`

Setup Configuration File

- Create initial config in /home/mha/confs/db.conf

```
[server default]
# mysql user and password
user=mha
password=as234fnRD32++
ssh_user=mha

repl_user = repl
repl_password = a352qINva23asda##

# working directories
manager_workdir=/var/log/masterha/db
remote_workdir=/var/log/masterha/db

[server1]
hostname=mha-demo-master
ip=192.168.56.120
candidate_master=1
[server2]
hostname=mha-demo-slave
ip=192.168.56.121
candidate_master=1
```

Grant permission in MySQL to mha

The last step... we need to give permission to an MHA user in mysql. (This is the user/password specified in the config we just made).

NOTE: This requires **ALL privileges** as mha *may need to* apply changes from any schema...

```
GRANT ALL ON *.* TO "mha"@"%" IDENTIFIED BY "as234fnRD32++";
```

Did you follow instructions?

Time to smoke test the installation!!

1. Check that SSH is all configured properly
    a. masterha_check_ssh --conf=/home/mha/confs/db.conf
    b. Look for output: **[info] All SSH connection tests passed successfully.**
2. Check that replication / mysql is all set up properly
    a. masterha_check_repl --conf=/home/mha/confs/db.conf
    b. Look for output: **MySQL Replication Health is OK.**

#winning

We are almost there, I promise!!

The last step to HA (I promise!)

Setup the failover scripts in /home/mha/scripts/

- Locate the templates here:
  - https://github.com/jayjanssen/mha_env/tree/master/mha_failover
  - Note: you will need to manually hardcode your new VIP (bummer, I know)
  - Store them (with updated VIP) in /home/mha/scripts
  - `chmod +x master_ip_*`
- Update the config to include the failover scripts

```
# Failover scripts
master_ip_failover_script = /home/mha/scripts/master_ip_failover
master_ip_online_change_script = /home/mha/scripts/master_ip_online_change
```

Testing once more...

```
masterha_check_repl --conf=/home/mha/confs/db.conf

Wed Jul 26 17:15:46 2017 - [info]
mha-demo-master(192.168.56.120:3306) (current master)
 +--mha-demo-slave(192.168.56.121:3306)

Wed Jul 26 17:15:46 2017 - [info] Checking replication health on mha-demo-slave..
Wed Jul 26 17:15:46 2017 - [info]  ok.
Wed Jul 26 17:15:46 2017 - [info] Checking master_ip_failover_script status:
Wed Jul 26 17:15:46 2017 - [info]   /home/mha/scripts/master_ip_failover --command=status --ssh_user=mha
--orig_master_host=mha-demo-master --orig_master_ip=192.168.56.120 --orig_master_port=3306
    inet 192.168.56.123/32 scope global enp0s8
INFO: VIP 192.168.56.123 found on Master
Wed Jul 26 17:15:46 2017 - [info]  OK.
Wed Jul 26 17:15:46 2017 - [warning] shutdown_script is not defined.
Wed Jul 26 17:15:46 2017 - [info] Got exit code 0 (Not master dead).

MySQL Replication Health is OK.
```

Verify Connections

- Confirm (either via CLI client or GUI) that all the IP addresses point to the correct servers
  - For this, I'm going to use MySQL Workbench to simulate my "application"
- Here is our current setup:
  - mha-demo-master: 192.168.56.120
  - mha-demo-slave: 192.168.56.121
  - mha-demo-vip: 192.168.56.123

Test a planned failover

- This is great for planned maintenance operations that require a restart
- Manually move the VIP from the slave to master

```
masterha_master_switch --conf=/home/mha/confs/db.conf --master_state=alive
--new_master_host=mha-demo-slave --orig_master_is_new_slave
```

If everything worked correctly, try to hit the VIP again from your “application” and check the server host name...

Now, let's put it all together!

The final step is to turn on the monitor...

```
masterha_manager --conf=/home/mha/confs/db.conf

…

Wed Jul 26 19:32:15 2017 - [info] Starting ping health check on
mha-demo-slave(192.168.56.121:3306)..
Wed Jul 26 19:32:15 2017 - [info] Ping(SELECT) succeeded, waiting until
MySQL doesn't respond..
```

Ooops... somebody unplugged the wrong server in the DC!!

Success!!

MHA detected the failure, re-assigned the VIP, and told you how to rebuild your slave when it comes back!

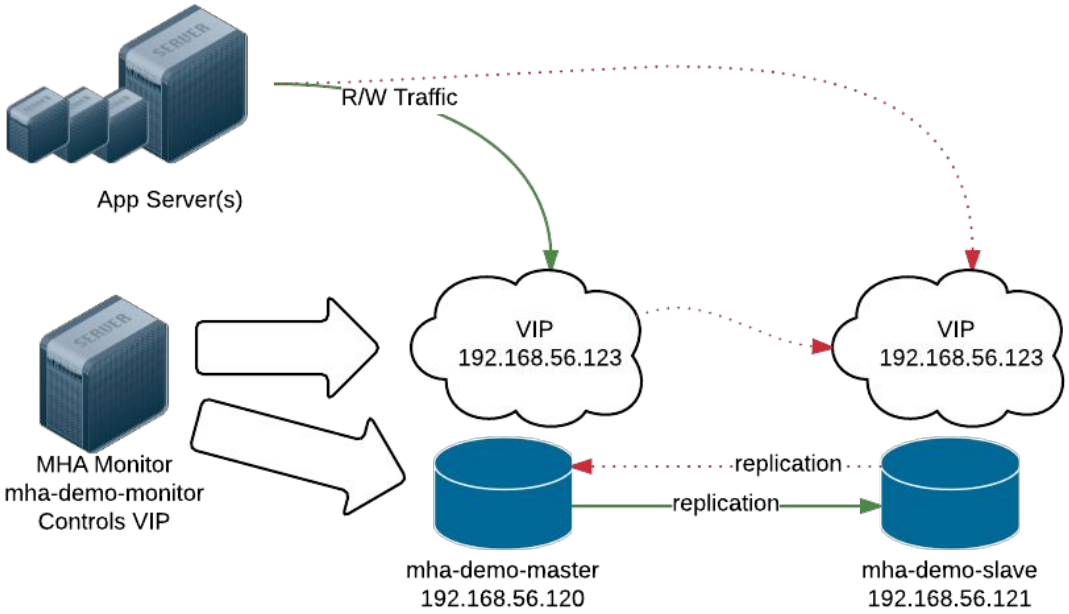
```
Wed Jul 26 19:35:07 2017 - [info]  All other slaves should start replication from here.
Statement should be: CHANGE MASTER TO MASTER_HOST='mha-demo-master or 192.168.56.120',
MASTER_PORT=3306, MASTER_LOG_FILE='mysql-bin.000003', MASTER_LOG_POS=1474,
MASTER_USER='repl', MASTER_PASSWORD='xxx';

...

Wed Jul 26 19:35:07 2017 - [info] Master failover to mha-demo-master(192.168.56.120:3306)
completed successfully.
```

Final Architecture

- Standard Master / Slave
- Application config
  - Database IP is VIP
  - R/W against **current master**
- If the master dies?
  - MHA shifts the IP
  - No application changes
  - No pager at 3:15AM!



We Made It!

Any questions or comments (nothing negative allowed)?

Get In Touch!

Website: http://okc-osdb.org

Email: mike@okc-osdb.org

Twitter: @okcopensourcedb