

# MySQL MHA 文档总结

## 目录

MySQL MHA 介绍.....	3
操作流程步骤.....	4
拓扑图演变.....	4
MHA 软件包说明.....	6
Manager 工具包.....	6
Node 工具包.....	6
实验环境.....	6
建立 ssh 无密码登录环境.....	7
manager  公约操作.....	7
主 mysql  公约操作.....	8
从 mysql1  公约操作.....	8
从 mysql2  公约操作.....	8
主机名.....	8
修改 hosts.....	9
测试 ssh 登录.....	9
安装 mysql 和配置主从关系.....	10
在线安装 mysql5.5.....	10
编辑 mysql 配置文件.....	10
启动 mysql 和查询启动状态.....	11
数据库一致性.....	12
半同步复制开启.....	12
配置 mysql 主从.....	13
测试 mysql 主从.....	14
部署 MHA.....	15
安装 MHA Node.....	15
安装 MHA manager.....	16
检查 SSH 配置.....	18
检查复制情况.....	19
启动 MHA manager.....	21
停止 MHA manager.....	22
任务计划.....	22
配置 vip.....	22
测试 MHA.....	24
停止主 mysql.....	24
查看从 mysql 情况.....	24
资料（源码包/配置文件）.....	26
参考文章.....	26
FAQ.....	26

# MySQL MHA

## ◆ MySQL MHA 介绍

实现原理：MHA 是由日本 Mysql 专家用 Perl 写的一套 Mysql 故障切换方案以保障数据库的高可用性，它的功能是在 0-30s 之内实现主 Mysql 故障转移（failover），MHA 故障转移可以很好的帮我们解决从库数据的一致性问题，同时最大化挽回故障发生后的数据。MHA 里有两个角色一个是 node 节点 一个是 manager 节点，要实现这个 MHA，必须最少要三台数据库服务器，一主多备，即一台充当 master，一台充当 master 的备份机，另外一台是从属机，这里实验为了实现更好的效果使用四台机器，需要说明的是一旦主服务器宕机，备份机即开始充当 master 提供服务，如果主服务器上线也不会再成为 master 了，因为如果这样数据库的一致性就被改变了。

该软件由两部分组成：MHA Manager（管理节点）和 MHA Node（数据节点）。MHA Manager 可以单独部署在一台独立的机器上管理多个 master-slave 集群，也可以部署在一台 slave 节点上。MHA Node 运行在每台 MySQL 服务器上，MHA Manager 会定时探测集群中的 master 节点，当 master 出现故障时，它可以自动将最新数据的 slave 提升为新的 master，然后将所有其他的 slave 重新指向新的 master。整个故障转移过程对应用程序完全透明。

在 MHA 自动故障切换过程中，MHA 试图从宕机的主服务器上保存二进制日志，最大程度的保证数据的不丢失，但这并不总是可行的。例如，如果主服务器硬件故障或无法通过 ssh 访问，MHA 没法保存二进制日志，只进行故障转移而丢失了最新的数据。使用 MySQL 5.5 的半同步复制，可以大大降低数据丢失的风险。MHA 可以与半同步复制结合起来。如果只有一个 slave 已经收到了最新的二进制日志，MHA 可以将最新的二进制日志应用于其他所有的 slave 服务器上，因此可以保证所有节点的数据一致性。

目前 MHA 主要支持一主多从的架构，要搭建 MHA，要求一个复制集群中必须最少有三台数据库服务器，一主二从，即一台充当 master，一台充当备用 master，另外一台充当从库，因为至少需要三台服务器，出于机器成本的考虑，淘宝也在该基础上进行了改造，目前淘宝 TMHA 已经支持一主一从。

官方介绍：<https://code.google.com/p/mysql-master-ha/>

注意：不可以访问 google 的时候，请使用翻墙或者修改 hosts 文件以下我提供一个 hosts 文件。



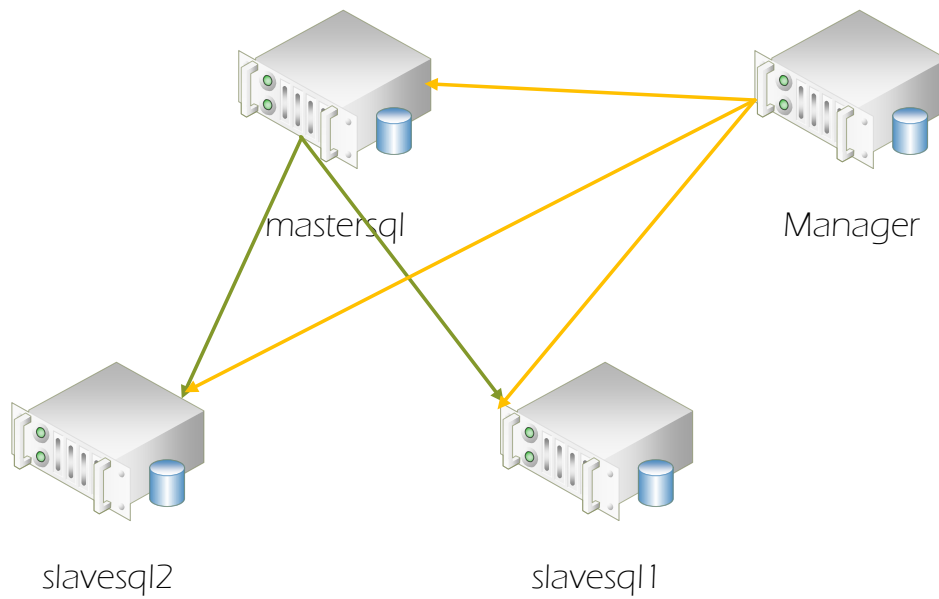
hosts文件.txt

将 hosts 文件内容拷贝到 C:\Windows\System32\drivers\etc\hosts 中粘贴到文件最后面保存即可，正常访问 google。

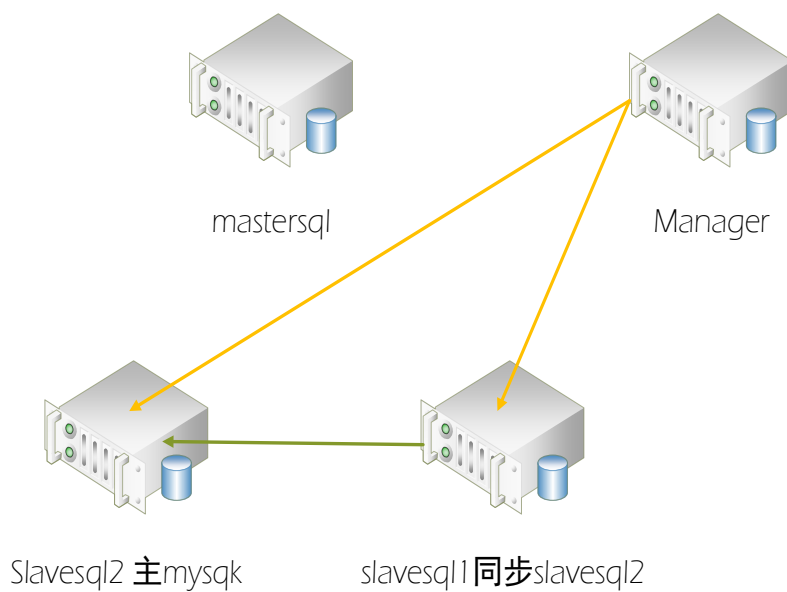
## ◆ 操作流程步骤

1. 首先要保证虚拟机能够上网，这里我使用公司电信云平台的四台 vmware 的虚拟机
2. 关闭 selinux 和配置 IP 地址和本地 source 源
3. 配置 epel 源（在线安装需要）
4. 配置 ssh 公钥免登录环境
5. 修改 hostname
6. 配置 hosts 文件
7. 配置 Mysql 的主从同步关系并通过 grant 命令赋权
8. 安装 node 包
9. 在管理机安装 manager 包
10. 编辑主配置文件
11. 测试及排错
12. 启动

## ◆ 拓扑图演变



主 mysql 宕机以后架构变成一主一从，



1. 从宕机崩溃的 master 保存二进制日志事件 (binlog events) ;
2. 识别含有最新更新的 slave;
3. 应用差异的中继日志 (relay log) 到其他的 slave;
4. 应用从 master 保存的二进制日志事件 (binlog events) ;
5. 提升一个 slave 为新的 master;
6. 使其他的 slave 连接新的 master 进行复制;

## ◆ MHA 软件包说明

MHA 软件由两部分组成，Manager 工具包和 Node 工具包，具体的说明如下。

### ❖ Manager 工具包

主要包括以下几个工具：

- |                            |                    |
|----------------------------|--------------------|
| 1. masterha_check_ssh      | 检查 MHA 的 SSH 配置状况  |
| 2. masterha_check_repl     | 检查 MySQL 复制状况      |
| 3. masterha_manger         | 启动 MHA             |
| 4. masterha_check_status   | 检测当前 MHA 运行状态      |
| 5. masterha_master_monitor | 检测 master 是否宕机     |
| 6. masterha_master_switch  | 控制故障转移（自动或者手动）     |
| 7. masterha_conf_host      | 添加或删除配置的 server 信息 |

### ❖ Node 工具包

这些工具通常由 MHA Manager 的脚本触发，无需人为操作）主要包括以下几个工具：

- |                       |                                   |
|-----------------------|-----------------------------------|
| save_binary_logs      | 保存和复制 master 的二进制日志               |
| apply_diff_relay_logs | 识别差异的中继日志事件并将其差异的事件应用于其他的 slave   |
| filter_mysqlbinlog    | 去除不必要的 ROLLBACK 事件（MHA 已不再使用这个工具） |
| purge_relay_logs      | 清除中继日志（不会阻塞 SQL 线程）               |

注意：

为了尽可能的减少主库硬件损坏宕机造成的数据丢失，因此在配置 MHA 的同时建议配置成 MySQL 5.5 的半同步复制。关于半同步复制原理各位自己进行查阅。（不是必须）

## ◆ 实验环境

使用电信云主机。详细情况：

- 1、操作系统版本  
Ubuntu 12.04.5 LTS \n \l

2、操作系统位数

x86\_64，64 位操作系统

3、操作系统内核

Linux mastersql 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86\_64 x86\_64 x86\_64 GNU/Linux

4、云主机情况

角色	Ip 地址	主机名	Server_id	类型
Monitor host	192.168.62.37	manager	37	监控复制组
Master	192.168.62.42	mastersql	42	写入
Candidate master	192.168.62.36	slavesql2	36	读
Slave	192.168.62.41	slavesql1	41	读

其中 master 对外提供写服务，备选 master（实际的 slave，主机名 slavesql2）提供读服务，slave 也提供相关的读服务，一旦 master 宕机，将会把备选 master 提升为新的 master，slave 指向新的 master。上面和下面所有的命令最好都使用 root 用户执行，我曾经使用非 root 用户，最后发现很烦，另 ubuntu 默认 root 是不可以 ssh 登陆的，要先：passwd root 给 root 添加密码，这样 root 就可以 ssh 登陆了。

◆ 建立 ssh 无密码登录环境

❖ manager 公约操作

```
root@manager: ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
20:a3:4a:e2:1f:c4:4c:9a:90:8f:00:c4:26:36:aa:81 root@iZ28i9mza1uZ
The key's randomart image is:
+--[ RSA 2048 ]-----+
|+.
|o*
|X .. o .
|Eo*. o .
|+=o+ S
|=. .
|.. .
| . .
| .
+-----+
```

然后在/root/.ssh/id\_rsa.pub.下面会新建两个文件 id\_rsa（私钥）和 id\_rsa.pub（私钥）。  
拷贝公钥到其他三台机器：

```
root@manager: ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.37
```

#为什么要在本机也要设置呢，因为 manager 节点安装在这上面，如不设置在下面 ssh 检查时会通不过。

```
root@manager: ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.37
root@manager: ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.36
root@manager: ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.41
```

过程示意图（因其过程都一样，故只示范 192.168.62.37）

## ❖ 主 mysql 公约操作

```
ssh-keygen -t rsa
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.37
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.41
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.42
```

## ❖ 从 mysql1 公约操作

```
ssh-keygen -t rsa
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.42
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.36
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.37
```

## ❖ 从 mysql2 公约操作

从 mysql2 也就是主 mysql 宕机以后要充当主 mysql 的备用主 mysql。

```
ssh-keygen -t rsa
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.37
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.41
ssh-copy-id -i .ssh/id_rsa.pub root@192.168.62.42
```

## ❖ 主机名

```
root@manager:~# cat /etc/hostname
```

```
manager
```

在不重启的情况下在手动执行一次。

```
root@localhost:~# hostname manager
```

```
root@manager:~# bash
```



我只列出 manager 机器操作步骤，其他机器一样操作设置 mastersql, slavesql1, slavesql2

## ❖ 修改 hosts

```
root@manager:~# cat /etc/host
```

```
host.conf    hostname    hosts        hosts.allow  hosts.deny
```

```
root@slavesql1:~# cat /etc/hosts
```

```
127.0.0.1 localhost
```

```
192.168.56.98 ubuntu1
```

```
# The following lines are desirable for IPv6 capable hosts
```

```
::1        ip6-localhost ip6-loopback
```

```
fe00::0 ip6-localnet
```

```
ff00::0 ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
```

```
ff02::2 ip6-allrouters
```

```
192.168.62.42 mastersql
```

```
192.168.62.41 slavesql1
```

```
192.168.62.36 slavesql2
```

```
192.168.62.37 manager
```

添加到 hosts 文件中，对应各自的 ip 和主机名



我只列出了 manager 的主机 hosts 文件，其他三台机器相同操作修改 hosts。

## ❖ 测试 ssh 登录

```
root@manager:~# ssh mastersql
```

```
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-23-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com/
```

```
System information as of Wed Aug 10 10:38:20 CST 2016
```

```
System load:  0.27                Processes:            83
```

```
Usage of /:   4.2% of 98.85GB      Users logged in:      1
```

```
Memory usage: 4%                IP address for eth0:   192.168.62.42
```

```
Swap usage:   0%                 IP address for eth0:0: 192.168.62.200
```

```
Graph this data and manage this system at https://landscape.canonical.com/
```

```
9 packages can be updated.
```

```
6 updates are security updates.
```

New release '14.04.4 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Aug 10 10:22:36 2016 from 192.168.62.33

root@mastersql:~#

说明已经无密码登陆了。

```
root@manager:~# ssh slavesql1
```

```
root@manager:~# ssh slavesql2
```



我只测试了 manger 无密码登陆到其他三台机器。其他类似。

## 安装 MySQL5.5

### ◆ 安装 mysql 和配置主从关系

#### ❖ 在线安装 mysql5.5

```
root@mastersql:~# apt-get install -y mysql-server
```

The following NEW packages will be installed:

libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16

libnet-daemon-perl libplrpc-perl mysql-client-5.5 mysql-client-core-5.5

mysql-common mysql-server mysql-server-5.5 mysql-server-core-5.5

0 upgraded, 12 newly installed, 0 to remove and 1 not upgraded.

Need to get 23.8 MB of archives.

After this operation, 61.2 MB of additional disk space will be used.

Do you want to continue [Y/n]? 此时按下 Y

.....

【mysql root 密码设为: root】

.....

#### ❖ 编辑 mysql 配置文件

```
root@mastersql:~# cat /etc/mysql/my.cnf | grep bind-address
```

```
bind-address                = 0.0.0.0 #这样就可以 root 远程连接 mysql
```

## ❖ 启动 mysql 和查询启动状态

## ➤ 启动 mysql

```
root@mastersql:~# /etc/init.d/mysql start
```

## ➤ 查看 mysql 的启动情况

```
root@mastersql:~# ps aux|grep -v grep | grep mysql
```

```
mysql      29437  0.0  0.6 634448 53868 ?        Ssl  Aug09   0:13 /usr/sbin/mysqld
```

可以看出 mysql 进程已经启动，且进 pid 号为 29347。接着查看 mysql 的端口号。

## ➤ 查看 mysql 的端口号

```
root@mastersql:~# netstat -lnp|grep mysql
```

```
tcp        0      0 0.0.0.0:3306          0.0.0.0:*        LISTEN      29437/mysqld
unix       2      0      [ ACC ]              STREAM          LISTENING   2662765     29437/mysqld
/var/run/mysqld/mysqld.sock
```

从上面的列表可以看出 mysql 的默认端口号已经启动。

## ➤ 登陆 mysql

```
root@mastersql:~# mysql -uroot -proot
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 39

Server version: 5.5.49-0ubuntu0.12.04.1-log (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| hive       |
| mysql     |
| networkd  |
| performance_schema |
+-----+
```

5 rows in set (0.01 sec)

以上单台 mysql 测试 OK。



此时我只在 mastersql 主机上在线安装了 mysql5.5，其他两个主机 mysqlsql1 和 mysqlsql2 相同 操作安装 mysql5.5。Manager 主机不需要安装 mysql。

## ❖ 数据库一致性

在主数据库里导出所有的数据库。然后导入到两个从数据库，保证做主从之前数据一致。主数据库之前一直在使用，两个从数据库都是新安装的。在导出数据库之前一定要锁表，或者保证数据库没有使用。

```
root@mastersql:mysql> mysqldump -uroot -proot --all-databases --lock-tables=false -- > /root/all.sql
```

把从主数据中的数据拷贝到两到两台从 mysql，且将该 all.sql 导入到这两个从 mysql 数据库。

```
root@slavesql1:mysql> mysql -uroot -proot </root/all.sql
```

```
root@slavesql2:mysql> mysql -uroot -proot </root/all.sql
```

## ❖ 半同步复制开启

➤ mastersql 上：

```
mysql> install plugin rpl_semi_sync_master soname 'semisync_master.so';
mysql> set global rpl_semi_sync_master_enabled=1;
mysql> set global rpl_semi_sync_master_timeout=1000;
mysql> show global status like 'rpl%';
```

为了让 mysql 在重启时自动加载该功能，在/etc/mysql/my.cnf 加入：

```
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000
```

➤ 备选 master (slavesql2) 上：

```
mysql> install plugin rpl_semi_sync_master soname 'semisync_master.so';
mysql> set global rpl_semi_sync_master_enabled=1;
mysql> set global rpl_semi_sync_master_timeout=1000;
mysql> install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
mysql> set global rpl_semi_sync_slave_enabled=1;
```

在/etc/mysql/my.cnf 中加入：

```
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000
rpl_semi_sync_slave_enabled=1
```

➤ slavesql1 上：

```
mysql> install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
mysql> set global rpl_semi_sync_slave_enabled=1;
```

在/etc/mysql/my.cnf 中加入：

```
rpl_semi_sync_slave_enabled=1
```

在备用节点和从节点的/etc/mysql/my.cnf 中加入选项:

read\_only=1

relay\_log\_purge=0

## ❖ 配置 mysql 主从

### ➤ 在 master 上

```
root@mastersql:~# cat /etc/mysql/my.cnf | grep log_bin
```

```
#log_bin            = /var/log/mysql/mysql-bin.log
```

```
log_bin            = realcloud
```

修改 mysql 的主的 log\_bin 日志名字, 用于主宕机以后从备主机接管后从同步的 log\_bin。名字不通用于区别。



此时, mysql 的 logbin 的位置路径是路径: /var/lib/mysql, 而不再是/var/log/mysql。

```
mysql> grant replication slave on *.* to realcloud@'192.168.62.%' identified by 'realcloud';
```

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
realcloud.000010	107		

1 row in set (0.00 sec)

记录下 “File” 和 “Position” 即当前主库使用的二进制日志名称和位置。

### ➤ 在备选 master (mysqlsql2) 和 slave (mysqlsql1) 上

```
mysql> slave stop;
```

```
mysql> change master to
```

```
master_host="192.168.62.42",master_user="realcloud",master_password="realcloud",master_log_file="realcloud.00007",master_log_pos=107;
```

master\_log\_file 和 master\_log\_pos 是上面记下的东西。

### ➤ 在备选 master (slavesql2) 上, 授权其网段内其他机器登陆的账号和密码

```
mysql> grant replication slave on *.* to repl@'192.168.62.%' identified by repl;
```

### ➤ 然后在备选 master (slavesql2) 和 slave (slavesql1) 上

```
mysql> slave start;
```

```
mysql> show slave status\G | egrep 'Slave_IO|Slave_SQL'
```

Slave\_IO\_State: Waiting for master to send event

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

# 如果 Slave\_IO\_Running: Yes 和 Slave\_SQL\_Running: Yes 则说明主从配置成功

# 还可以到 master 上执行 Mysql>show global status like "rpl%";

```
mysql> show global status like "rpl%";
```

Variable_name	Value
Rpl_semi_sync_master_clients	2
Rpl_semi_sync_master_net_avg_wait_time	0
Rpl_semi_sync_master_net_wait_time	0
Rpl_semi_sync_master_net_waits	0
Rpl_semi_sync_master_no_times	0
Rpl_semi_sync_master_no_tx	0
Rpl_semi_sync_master_status	ON
Rpl_semi_sync_master_timefunc_failures	0
Rpl_semi_sync_master_tx_avg_wait_time	0
Rpl_semi_sync_master_tx_wait_time	0
Rpl_semi_sync_master_tx_waits	0
Rpl_semi_sync_master_wait_pos_backtraverse	0
Rpl_semi_sync_master_wait_sessions	0
Rpl_semi_sync_master_yes_tx	0
Rpl_semi_sync_slave_status	OFF
Rpl_status	AUTH_MASTER

如果 Rpl\_semi\_sync\_master\_clients 是 2.说明半同步复制正常



在做主从的时候，我设置了两组账号和密码，分别如下，列其作用

1. account: root      password: root  
用于登陆远程 mysql 账号和密码
2. account: realcloud      password: realcloud  
用于主从复制的时候，主开发给从的账号和密码
3. account: repl      password: repl  
Manager 管理时候使用的账号和密码

## ❖ 测试 mysql 主从

### ➤ 主 mysql 创建数据库

```
root@mastersql:~# mysql -uroot -proot -e "create database ReadlCloudDatabase;"
```

```
root@mastersql:~# mysql -uroot -proot -e "show databases;"
```

Database
information_schema
ReadlCloudDatabase
hive
mysql
networkd

```
| performance_schema |
+-----+
|
```

主 mysql 创建一个新的数据库为 ReadlCloudDatabase。

- 被主 mysql 查看同步情况

```
root@slavesql2:~# mysql -uroot -proot -e "show databases;"
```

```
+-----+
| Database |
+-----+
| information_schema |
| ReadlCloudDatabase |
| hive |
| mysql |
| networkd |
| performance_schema |
+-----+
```

- 从 mysql 查看同步情况

```
root@slavesql1:~# mysql -uroot -proot -e "show databases;"
```

```
+-----+
| Database |
+-----+
| information_schema |
| ReadlCloudDatabase |
| hive |
| mysql |
| networkd |
| performance_schema |
+-----+
```

结合之前的 status 状态和 mysql 主从同步操作查看可见测试 mysql 主从成功。

## 安装 MHA

### ◆ 部署 MHA

### ❖ 安装 MHA Node

- 先在 4 台机器上安装 MHA Node:

```
root@manager:~# apt-get install libdbd-mysql-perl
root@manager:~# dpkg -i mha4mysql-node_0.54-0_all.deb
```

```
root@manager:~# ll /usr/local/bin
```

```
total 40
```

```
-r-xr-xr-x 1 root root 15498 Aug 10 14:05 apply_diff_relay_logs
-r-xr-xr-x 1 root root 4807 Aug 10 14:05 filter_mysqlbinlog
-r-xr-xr-x 1 root root 7401 Aug 10 14:05 purge_relay_logs
-r-xr-xr-x 1 root root 7263 Aug 10 14:05 save_binary_logs
```



其他三台机器按照此方式一样安装。

## ❖ 安装 MHA manager

➤ 在 manager 上安装 MHA Manager:

```
root@manager:~# apt-get install libdbd-mysql-perl
root@manager:~# apt-get install libconfig-tiny-perl
root@manager:~# apt-get install liblog-dispatch-perl
root@manager:~# apt-get install libparallel-forkmanager-perl
root@manager:~# mha4mysql-manager-0.56.tar.gz
root@manager:~# cd mha4mysql-manager-0.56
root@manager:~/mha4mysql-manager-0.56# perl Makefile
root@manager:~/mha4mysql-manager-0.56# make && make install
```

安装完毕后会会在/usr/local/bin 目录下生成以下脚本文件:

```
root@manager:~# ll /usr/local/bin
```

```
total 26052
```

```
drwxr-xr-x 2 root root 4096 Aug 9 17:07 ./
drwxr-xr-x 12 root root 4096 Aug 9 16:50 ../
-r-xr-xr-x 1 root root 15977 Aug 8 20:03 apply_diff_relay_logs*
-rwxr-xr-x 1 root root 8023561 Aug 8 11:58 cmake*
-rwxr-xr-x 1 root root 8587942 Aug 8 11:58 cpack*
-rwxr-xr-x 1 root root 9939826 Aug 8 11:58 ctest*
-r-xr-xr-x 1 root root 4807 Aug 8 20:03 filter_mysqlbinlog*
-rwxr-xr-x 1 root root 679 Jul 8 16:44 KernelUpdateChecker*
-rwxr-xr-x 1 root root 245 Jul 8 16:44 KernelUpdate.desktop*
-rwxr-xr-x 1 root root 9627 Jul 8 16:44 KernelUpdateScriptGenerator*
-r-xr-xr-x 1 root root 1995 Aug 8 20:20 masterha_check_repl*
-r-xr-xr-x 1 root root 1779 Aug 8 20:20 masterha_check_ssh*
-r-xr-xr-x 1 root root 1865 Aug 8 20:20 masterha_check_status*
-r-xr-xr-x 1 root root 3201 Aug 8 20:20 masterha_conf_host*
-r-xr-xr-x 1 root root 2517 Aug 8 20:20 masterha_manager*
-r-xr-xr-x 1 root root 2165 Aug 8 20:20 masterha_master_monitor*
-r-xr-xr-x 1 root root 2373 Aug 8 20:20 masterha_master_switch*
-r-xr-xr-x 1 root root 5171 Aug 9 10:22 masterha_secondary_check*
-r-xr-xr-x 1 root root 1739 Aug 8 20:20 masterha_stop*
```

```
-rwx----- 1 root root    2744 Aug  9 17:07 master_ip_failover*
-rwxrwxrwx  1 root root    2744 Aug  9 12:55 master_ip_online_change_script*
-rw-----  1 root root      291 Aug  9 15:42 nohup.out
-r-xr-xr-x  1 root root   7401 Aug  8 20:03 purge_relay_logs*
-r-xr-xr-x  1 root root   7263 Aug  8 20:03 save_binary_logs*
```

[root@192.168.0.50 bin]#

➤ 查看 mha 的自带的脚本

```
root@manager:~/mha4mysql-manager-0.56/samples/scripts# ll /root/mha4mysql-manager-0.56/samples/scripts
total 40
```

```
drwxr-xr-x 2 4984 users  4096 Apr  1  2014 ./
drwxr-xr-x 4 4984 users  4096 Apr  1  2014 ../
-rwxr-xr-x 1 4984 users  3648 Apr  1  2014 master_ip_failover*
```

#自动切换时 vip 管理的脚本，不是必须，如果我们使用 keepalived 的，我们可以自己编写脚本完成对 vip 的管理，比如监控 mysql，如果 mysql 异常，我们停止 keepalived 就行，这样 vip 就会自动漂移

```
-rwxr-xr-x 1 4984 users  9870 Apr  1  2014 master_ip_online_change*
```

#在线切换时 vip 的管理，不是必须，同样可以自行编写简单的 shell 完成

```
-rwxr-xr-x 1 4984 users 11867 Apr  1  2014 power_manager*
```

#故障发生后关闭主机的脚本，不是必须

```
-rwxr-xr-x 1 4984 users  1360 Apr  1  2014 send_report*
```

#因故障切换后发送报警的脚本，不是必须，可自行编写简单的 shell 完成。

复制相关脚本到/usr/local/bin 目录(软件包解压缩后就有了，不是必须，因为这些脚本不完整，需要自己修改，这是软件开发着留给我们自己发挥的,如果开启下面的任何一个脚本对应的参数，而对应这里的脚本又没有修改，则会抛错，自己被坑的很惨)

➤ 编辑配置文件在 manager 上创建配置文件/etc/app1.cnf，内容如下：

```
root@manager:~# cat /etc/app1.cnf
```

```
[server default]
```

```
manager_log=/masterha/app1/manager.log
```

#manager 的 log 日志

```
manager_workdir=/masterha/app1
```

#manager 的工作目录

```
master_ip_failover_script=/usr/local/bin/master_ip_failover
```

#vip 漂移的脚本和路径

```
master_ip_online_change_script=/usr/local/bin/master_ip_online_change_script
```

#设置手动切换时候脚本

```
password=root
```

#设置 mysql 中的 root 用户的密码，这个，IMA 是前面中创建监控用户的那个密码

```
user=root
```

#设置监控用户为 root

```
ping_interval=1
```

#设置监控主库，发送 ping 包的时间间隔，默认是 3 秒，

#尝试三次没有回应的时候自动进行 railover

```
repl_password=repl
```

#设置复制用户的密码

```
repl_user=repl
```

#设置复制环境中的复制用户名

```
ssh_user=root
```

#设置 ssh 的登陆用户名

```
user=root
```

```
#shutdown_script=""
```

# 设置故障发生后关闭故障主机脚本（该脚本的主要作用是关闭主机放在发生脑裂,这里没有使用）

```
[server1]
```

```
hostname=mastersql
```

```
master_binlog_dir=/var/lib/mysql
```

```
[server2]
```

candidate\_master=1 #设置为候选 master，如果设置该参数以后，发生主从切换以后将会将此从库提升为主库，即使这个主库不是集群中事件最新的 slave

check\_repl\_delay=0 #默认情况下如果一个 slave 落后 master 100M 的 relay logs 的话，MHA 将不会选择该 slave 作为一个新的 master，因为对于这个 slave 的恢复需要花费很长时间，通过设置 check\_repl\_delay=0,MHA 触发切换在选择一个新的 master 的时候将会忽略复制延时，这个参数对于设置了 candidate\_master=1 的主机非常有用，因为这个候选主在切换的过程中一定是新的 master

```
hostname=slavesql2
```

```
master_binlog_dir=/var/log/mysql
```

```
[server3]
```

```
hostname=slavesql1
```

```
master_binlog_dir=/var/log/mysql
```

```
no_master=1
```

## ❖ 检查 SSH 配置

➤ 检查 MHA manager 到所有 MHA Node 的 SSH 连接状态

```
root@manager:~# masterha_check_ssh --conf=/etc/app1.cnf
```

```
Wed Aug 10 14:46:55 2016 - [info] Reading default configuration from /etc/masterha_default.cnf..
```

```
Wed Aug 10 14:46:55 2016 - [info] Reading application default configuration from /etc/app1.cnf..
```

```
Wed Aug 10 14:46:55 2016 - [info] Reading server configuration from /etc/app1.cnf..
```

```
Wed Aug 10 14:46:55 2016 - [info] Starting SSH connection tests..
```

```
Wed Aug 10 14:46:57 2016 - [debug]
```

```
Wed Aug 10 14:46:55 2016 - [debug] Connecting via SSH from root@mastersql(192.168.62.42:22) to root@slavesql2(192.168.62.36:22)..
```

```
Wed Aug 10 14:46:56 2016 - [debug] ok.
```

```
Wed Aug 10 14:46:56 2016 - [debug] Connecting via SSH from root@mastersql(192.168.62.42:22) to root@slavesql1(192.168.62.41:22)..
```

```
Wed Aug 10 14:46:57 2016 - [debug] ok.
```

```
Wed Aug 10 14:46:57 2016 - [debug]
```

```
Wed Aug 10 14:46:56 2016 - [debug] Connecting via SSH from root@slavesql2(192.168.62.36:22) to root@mastersql(192.168.62.42:22)..
```

```
Wed Aug 10 14:46:56 2016 - [debug] ok.
```

```
Wed Aug 10 14:46:56 2016 - [debug] Connecting via SSH from root@slavesql2(192.168.62.36:22) to root@slavesql1(192.168.62.41:22)..
```

```
Wed Aug 10 14:46:57 2016 - [debug] ok.
```

```
Wed Aug 10 14:46:58 2016 - [debug]
```

```
Wed Aug 10 14:46:56 2016 - [debug] Connecting via SSH from root@slavesql1(192.168.62.41:22) to root@mastersql(192.168.62.42:22)..
```

```
Wed Aug 10 14:46:57 2016 - [debug] ok.
```

```
Wed Aug 10 14:46:57 2016 - [debug] Connecting via SSH from root@slavesql1(192.168.62.41:22) to
```

```
root@slavesql2(192.168.62.36:22)..  
Wed Aug 10 14:46:58 2016 - [debug]    ok.  
Wed Aug 10 14:46:58 2016 - [info] All SSH connection tests passed successfully.  
SSH 连接测试 OK。
```

## ❖ 检查复制情况

```
root@manager:~# masterha_check_repl --conf=/etc/app1.cnf  
Wed Aug 10 17:06:09 2016 - [info] Reading default configuration from /etc/masterha_default.cnf..  
Wed Aug 10 17:06:09 2016 - [info] Reading application default configuration from /etc/app1.cnf..  
Wed Aug 10 17:06:09 2016 - [info] Reading server configuration from /etc/app1.cnf..  
Wed Aug 10 17:06:09 2016 - [info] MHA::MasterMonitor version 0.56.  
Wed Aug 10 17:06:09 2016 - [info] GTID failover mode = 0  
Wed Aug 10 17:06:09 2016 - [info] Dead Servers:  
Wed Aug 10 17:06:09 2016 - [info] Alive Servers:  
Wed Aug 10 17:06:09 2016 - [info]    mastersql(192.168.62.42:3306)  
Wed Aug 10 17:06:09 2016 - [info]    slavesql2(192.168.62.36:3306)  
Wed Aug 10 17:06:09 2016 - [info]    slavesql1(192.168.62.41:3306)  
Wed Aug 10 17:06:09 2016 - [info] Alive Slaves:  
Wed Aug 10 17:06:09 2016 - [info]    slavesql2(192.168.62.36:3306) Version=5.5.50-0ubuntu0.12.04.1-log (oldest  
major version between slaves) log-bin:enabled  
Wed Aug 10 17:06:09 2016 - [info]    Replicating from 192.168.62.42(192.168.62.42:3306)  
Wed Aug 10 17:06:09 2016 - [info]    Primary candidate for the new Master (candidate_master is set)  
Wed Aug 10 17:06:09 2016 - [info]    slavesql1(192.168.62.41:3306) Version=5.5.49-0ubuntu0.12.04.1-log (oldest  
major version between slaves) log-bin:enabled  
Wed Aug 10 17:06:09 2016 - [info]    Replicating from 192.168.62.42(192.168.62.42:3306)  
Wed Aug 10 17:06:09 2016 - [info]    Not candidate for the new Master (no_master is set)  
Wed Aug 10 17:06:09 2016 - [info] Current Alive Master: mastersql(192.168.62.42:3306)  
Wed Aug 10 17:06:09 2016 - [info] Checking slave configurations..  
Wed Aug 10 17:06:09 2016 - [info]    read_only=1 is not set on slave slavesql2(192.168.62.36:3306).  
Wed Aug 10 17:06:09 2016 - [warning]    relay_log_purge=0 is not set on slave slavesql2(192.168.62.36:3306).  
Wed Aug 10 17:06:09 2016 - [warning]    relay_log_purge=0 is not set on slave slavesql1(192.168.62.41:3306).  
Wed Aug 10 17:06:09 2016 - [info] Checking replication filtering settings..  
Wed Aug 10 17:06:09 2016 - [info]    binlog_do_db= , binlog_ignore_db=  
Wed Aug 10 17:06:09 2016 - [info]    Replication filtering check ok.  
Wed Aug 10 17:06:09 2016 - [info] GTID (with auto-pos) is not supported  
Wed Aug 10 17:06:09 2016 - [info] Starting SSH connection tests..  
Wed Aug 10 17:06:12 2016 - [info] All SSH connection tests passed successfully.  
Wed Aug 10 17:06:12 2016 - [info] Checking MHA Node version..  
Wed Aug 10 17:06:12 2016 - [info]    Version check ok.  
Wed Aug 10 17:06:12 2016 - [info] Checking SSH publickey authentication settings on the current master..  
Wed Aug 10 17:06:13 2016 - [info] HealthCheck: SSH to mastersql is reachable.  
Wed Aug 10 17:06:13 2016 - [info] Master MHA Node version is 0.54.  
Wed Aug 10 17:06:13 2016 - [info] Checking recovery script configurations on mastersql(192.168.62.42:3306)..  
Wed Aug 10 17:06:13 2016 - [info]    Executing command: save_binary_logs --command=test --start_pos=4
```

```
--binlog_dir=/var/lib/mysql          --output_file=/var/tmp/save_binary_logs_test          --manager_version=0.56
--start_file=realcloud.000012
Wed Aug 10 17:06:13 2016 - [info]    Connecting to root@192.168.62.42(mastersql:22)..
    Creating /var/tmp if not exists..    ok.
    Checking output directory is accessible or not..
        ok.
    Binlog found at /var/lib/mysql, up to realcloud.000012
Wed Aug 10 17:06:14 2016 - [info] Binlog setting check done.
Wed Aug 10 17:06:14 2016 - [info] Checking SSH publickey authentication and checking recovery script configurations
on all alive slave servers..
Wed Aug 10 17:06:14 2016 - [info]    Executing command : apply_diff_relay_logs --command=test --slave_user='root'
--slave_host=slavesql2          --slave_ip=192.168.62.36          --slave_port=3306          --workdir=/var/tmp
--target_version=5.5.50-0ubuntu0.12.04.1-log --manager_version=0.56--relay_log_info=/var/lib/mysql/relay-log.info
--relay_dir=/var/lib/mysql/ --slave_pass=xxx
Wed Aug 10 17:06:14 2016 - [info]    Connecting to root@192.168.62.36(slavesql2:22)..
    Checking slave recovery environment settings..
        Opening /var/lib/mysql/relay-log.info ... ok.
        Relay log found at /var/lib/mysql, up to mysqld-relay-bin.000006
        Temporary relay log file is /var/lib/mysql/mysqld-relay-bin.000006
        Testing mysql connection and privileges.. done.
        Testing mysqlbinlog output.. done.
        Cleaning up test file(s).. done.
Wed Aug 10 17:06:14 2016 - [info]    Executing command : apply_diff_relay_logs --command=test --slave_user='root'
--slave_host=slavesql1          --slave_ip=192.168.62.41          --slave_port=3306          --workdir=/var/tmp
--target_version=5.5.49-0ubuntu0.12.04.1-log --manager_version=0.56--relay_log_info=/var/lib/mysql/relay-log.info
--relay_dir=/var/lib/mysql/ --slave_pass=xxx
Wed Aug 10 17:06:14 2016 - [info]    Connecting to root@192.168.62.41(slavesql1:22)..
    Checking slave recovery environment settings..
        Opening /var/lib/mysql/relay-log.info ... ok.
        Relay log found at /var/lib/mysql, up to mysqld-relay-bin.000007
        Temporary relay log file is /var/lib/mysql/mysqld-relay-bin.000007
        Testing mysql connection and privileges.. done.
        Testing mysqlbinlog output.. done.
        Cleaning up test file(s).. done.
Wed Aug 10 17:06:15 2016 - [info] Slaves settings check done.
Wed Aug 10 17:06:15 2016 - [info]
mastersql(192.168.62.42:3306) (current master)
+--slavesql2(192.168.62.36:3306)
+--slavesql1(192.168.62.41:3306)

Wed Aug 10 17:06:15 2016 - [info] Checking replication health on slavesql2..
Wed Aug 10 17:06:15 2016 - [info]    ok.
Wed Aug 10 17:06:15 2016 - [info] Checking replication health on slavesql1..
Wed Aug 10 17:06:15 2016 - [info]    ok.
Wed Aug 10 17:06:15 2016 - [info] Checking master_ip_failover_script status:
Wed Aug 10 17:06:15 2016 - [info]    /usr/local/bin/master_ip_failover --command=status --ssh_user=root
```

```
--orig_master_host=mastersql --orig_master_ip=192.168.62.42 --orig_master_port=3306
```

```
IN SCRIPT TEST====/sbin/ifconfig eth0:1 down==/sbin/ifconfig eth0:1 192.168.62.200/24===
```

Checking the Status of the script.. OK

ssh: Could not resolve hostname cluster1: Name or service not known

Wed Aug 10 17:06:20 2016 - [info] OK.

Wed Aug 10 17:06:20 2016 - [warning] shutdown\_script is not defined.

Wed Aug 10 17:06:20 2016 - [info] Got exit code 0 (Not master dead).

s

MySQL Replication Health is OK.

此时可以看出 mysql 的主从检查 OK。

此时查看 vip 在 master 已经启动。

```
root@mastersql:~# ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 00:50:56:a8:6b:35
          inet addr:192.168.62.42  Bcast:192.168.62.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea8:6b35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7406297 errors:0 dropped:2772 overruns:0 frame:0
          TX packets:4406496 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:572214031 (572.2 MB)  TX bytes:261846085 (261.8 MB)
```

```
eth0:0    Link encap:Ethernet  HWaddr 00:50:56:a8:6b:35
          inet addr:192.168.62.200  Bcast:192.168.62.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:31209 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2803948 (2.8 MB)  TX bytes:2803948 (2.8 MB)
```

## ❖ 启动 MHA manager

```
root@manager:~# nohup masterha_manager --conf=/etc/app1.cnf < /dev/null > /masterha/app1/manager.log 2>&1 &
[2] 14422
```

```
root@manager:~# masterha_check_status --conf=/etc/app1.cnf
app1 (pid:14393) is running(0:PING_OK), master:mastersql
```

## ❖ 停止 MHA manager

```
root@manager:~# masterha_stop --conf=/etc/app1.cnf
# 如果不能停止， 加 --abort 选项
```

## ❖ 任务计划

在备选 maste (slavesql2) 和 slave(slavesql1)节点 crontab -e 添加计划任务

```
00 00 * * * /usr/local/bin/purge_relay_logs -user=root -password=root - disable_relay_log_purge >>
/var/log/purge_relay_logs.log 2>&1
```

Purge\_relay\_logs 脚本中继日志不会阻塞 SQL 线程，在每台从节点上设置计划任务定期清除中继日志

## ❖ 配置 vip

➤ 有个简单的方法添加虚拟 ip, 用 ifconfig 命令，添加 VIP:

```
/sbin/ifconfig eth0:1 192.168.62.200/24
```

➤ 删除 VIP:

```
/sbin/ifconfig eth0:1 down
```

➤ 网上找了一个 master\_ip\_failover 脚本就是用此方法更改 VIP:

```
#!/usr/bin/env perl
use strict;
use warnings FATAL => 'all';
use Getopt::Long;
my (
    $command,          $ssh_user,          $orig_master_host, $orig_master_ip,
    $orig_master_port, $new_master_host, $new_master_ip,    $new_master_port);
my $vip = '192.168.62.200'; # Virtual IP
my $key = "1"; my $ssh_start_vip = "/sbin/ifconfig eth0:$key $vip"; my
$ssh_stop_vip = "/sbin/ifconfig eth0:$key down";
GetOptions(
    'command=s'          => \$command,
    'ssh_user=s'         => \$ssh_user,
    'orig_master_host=s' => \$orig_master_host,
    'orig_master_ip=s'   => \$orig_master_ip,
    'orig_master_port=i' => \$orig_master_port,
    'new_master_host=s'  => \$new_master_host,
    'new_master_ip=s'    => \$new_master_ip,
    'new_master_port=i'  => \$new_master_port,);
exit &main();
```

```

sub main {

    print "\n\nIN SCRIPT TEST====$ssh_stop_vip==$ssh_start_vip==\n\n";

    if ( $command eq "stop" || $command eq "stopssh" ) {

        # $orig_master_host, $orig_master_ip, $orig_master_port are passed.
        # If you manage master ip address at global catalog database,
        # invalidate orig_master_ip here.
        my $exit_code = 1;
        eval {
            print "Disabling the VIP on old master: $orig_master_host \n";
            &stop_vip();
            $exit_code = 0;
        };
        if ($?) {
            warn "Got Error: $?\n";
            exit $exit_code;
        }
        exit $exit_code;
    }
    elsif ( $command eq "start" ) {

        # all arguments are passed.
        # If you manage master ip address at global catalog database,
        # activate new_master_ip here.
        # You can also grant write access (create user, set read_only=0, etc) here.
        my $exit_code = 10;
        eval {
            print "Enabling the VIP - $vip on the new master - $new_master_host \n";
            &start_vip();
            $exit_code = 0;
        };
        if ($?) {
            warn $?;
            exit $exit_code;
        }
        exit $exit_code;
    }
    elsif ( $command eq "status" ) {
        print "Checking the Status of the script.. OK \n";
        `ssh $ssh_user\@cluster1 \" $ssh_start_vip \"`;
        exit 0;
    }
    else {
        &usage();
    }
}

```

```

        exit 1;
    }}
# A simple system call that enable the VIP on the new mastersub start_vip() {
    `ssh $ssh_user@$new_master_host \" $ssh_start_vip \";`}# A simple system call that disable the VIP on the
old_mastersub stop_vip() {
    `ssh $ssh_user@$orig_master_host \" $ssh_stop_vip \";`}
sub usage {
    print
    "Usage: master_ip_failover --command=start|stop|stopssh|status --orig_master_host=host --orig_master_ip=ip
--orig_master_port=port --new_master_host=host --new_master_ip=ip --new_master_port=port\n";}

```

将此文档复制两次到/usr/local/bin, 分别命名为 master\_ip\_failover 和 master\_ip\_online\_change\_script  
然后将 manager 主机中的/etc/app1.cnf 中下面两行注释去掉（使用该脚本）,注意添加可执行权限。

```

master_ip_failover_script=/usr/local/bin/master_ip_failover
master_ip_online_change_script=/usr/local/bin/master_ip_online_change_script

```

## ◆ 测试 MHA

### ❖ 停止主 mysql

在主 mysql 的主机上行停止 mysql 服务。

```
root@mastersql:~# /etc/init.d/mysql stop
```

### ❖ 查看从 mysql 情况

```
root@slavesql2:~# mysql -uroot -proot
```

```
mysql> show slave status\G;
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 192.168.62.36 #此时 slave 的已经切换到备主 mysql (slavesql2)
```

```
Master_User: repl
```

```
Master_Port: 3306
```

```
Connect_Retry: 60
```

```
Master_Log_File: mysql-bin.000002
```

```
Read_Master_Log_Pos: 107
```

```
Relay_Log_File: mysqld-relay-bin.000002
```

```
Relay_Log_Pos: 253
```

```
Relay_Master_Log_File: mysql-bin.000002
```

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```

```
Replicate_Do_DB:
```

```
Replicate_Ignore_DB:
```

```
Replicate_Do_Table:
```

```
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
    Last_Errno: 0
    Last_Error:
    Skip_Counter: 0
Exec_Master_Log_Pos: 107
Relay_Log_Space: 410
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
    Last_IO_Errno: 0
    Last_IO_Error:
    Last_SQL_Errno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:
    Master_Server_Id: 36
1 row in set (0.00 sec)
```

ERROR:

No query specified

可以看出 mysql 的从已经切换到新的 master mysql 机器上。此时查看 vip 已经切换到备主 mysql (slavesql2) 上。  
root@slavesql2:~# ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:50:56:a8:6b:35
          inet addr:192.168.62.42  Bcast:192.168.62.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea8:6b35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7406297 errors:0 dropped:2772 overruns:0 frame:0
          TX packets:4406496 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:572214031 (572.2 MB)  TX bytes:261846085 (261.8 MB)

eth0:0    Link encap:Ethernet  HWaddr 00:50:56:a8:6b:35
          inet addr:192.168.62.200  Bcast:192.168.62.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
```

RX packets:31209 errors:0 dropped:0 overruns:0 frame:0  
TX packets:31209 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:2803948 (2.8 MB) TX bytes:2803948 (2.8 MB)

## ◆ 资料（源码包/配置文件）



data.tar.gz

## ◆ 参考文章

1. MySQL 做热备和高可用的方法有很多种， 比如：  
mmm: <http://mysql-mmm.org/>  
mha: <https://code.google.com/p/mysql-master-ha/>  
heartbeat+brdb: <http://lin128.blog.51cto.com/407924/279411> <http://www.centos.bz/2012/03/achieve-drbd-high-availability-with-heartbeat/>
2. cluster(使用 ndb 引擎):<http://database.51cto.com/art/201008/218326.htm>
3. 双 master+keepived: <http://database.51cto.com/art/201012/237204.htm>
4. 双 master: <http://yunnick.iteye.com/blog/1845301>
5. 官方 wiki: <https://code.google.com/p/mysql-master-ha/wiki/Tutorial>
6. 使用 MHA 做 mysql 的高可用: <http://qiufengy.blog.51cto.com/391990/848468>
7. Mysql5.5 部署 MHA: <http://ylw6006.blog.51cto.com/470441/890360>
8. mysql High Availability -MHA: <http://www.vmcd.org/2012/04/mysql-high-availability-mha/>
9. MySQL 高可用性大杀器之 MHA: <http://huoding.com/2011/12/18/139>
10. mysql-mha 高可用 : <http://blog.chinaunix.net/uid-28437434-id-3476641.html>
11. 另外有个 slide 讲 mha 的，可以看看: <http://www.slideshare.net/ylouis83/mysqlmha>

## ◆ FAQ

### 1、主从健康检查出错问题 1

```
root@manager:~# masterha_check_repl --conf=/etc/app1.cnf
Wed Aug 10 15:02:09 2016 - [info] Reading default configuration from /etc/masterha_default.cnf..
Wed Aug 10 15:02:09 2016 - [info] Reading application default configuration from /etc/app1.cnf..
Wed Aug 10 15:02:09 2016 - [info] Reading server configuration from /etc/app1.cnf..
Wed Aug 10 15:02:09 2016 - [info] MHA::MasterMonitor version 0.56.
Wed Aug 10 15:02:09 2016 - [warning] SQL Thread is stopped(no error) on mastersql(192.168.62.42:3306)
Wed Aug 10 15:02:09 2016 - [info] GTID failover mode = 0
```

Wed Aug 10 15:02:09 2016 - [info] Dead Servers:  
Wed Aug 10 15:02:09 2016 - [info] Alive Servers:  
Wed Aug 10 15:02:09 2016 - [info] mastersql(192.168.62.42:3306)  
Wed Aug 10 15:02:09 2016 - [info] slavesql2(192.168.62.36:3306)  
Wed Aug 10 15:02:09 2016 - [info] slavesql1(192.168.62.41:3306)  
Wed Aug 10 15:02:09 2016 - [info] Alive Slaves:  
Wed Aug 10 15:02:09 2016 - [info] mastersql(192.168.62.42:3306) Version=5.5.49-0ubuntu0.12.04.1-log (oldest major version between slaves) log-bin:enabled  
Wed Aug 10 15:02:09 2016 - [info] Replicating from 192.168.62.42(192.168.62.42:3306)  
Wed Aug 10 15:02:09 2016 - [info] slavesql2(192.168.62.36:3306) Version=5.5.50-0ubuntu0.12.04.1-log (oldest major version between slaves) log-bin:enabled  
Wed Aug 10 15:02:09 2016 - [info] Replicating from 192.168.62.42(192.168.62.42:3306)  
Wed Aug 10 15:02:09 2016 - [info] Primary candidate for the new Master (candidate\_master is set)  
Wed Aug 10 15:02:09 2016 - [info] slavesql1(192.168.62.41:3306) Version=5.5.49-0ubuntu0.12.04.1-log (oldest major version between slaves) log-bin:enabled  
Wed Aug 10 15:02:09 2016 - [info] Replicating from 192.168.62.42(192.168.62.42:3306)  
Wed Aug 10 15:02:09 2016 - [info] Not candidate for the new Master (no\_master is set)  
Wed Aug 10 15:02:09 2016 - [info] Current Alive Master: mastersql(192.168.62.42:3306)  
Wed Aug 10 15:02:09 2016 - [info] Checking slave configurations..  
Wed Aug 10 15:02:09 2016 - [info] read\_only=1 is not set on slave mastersql(192.168.62.42:3306).  
Wed Aug 10 15:02:09 2016 - [warning] relay\_log\_purge=0 is not set on slave mastersql(192.168.62.42:3306).  
Wed Aug 10 15:02:09 2016 - [info] read\_only=1 is not set on slave slavesql2(192.168.62.36:3306).  
Wed Aug 10 15:02:09 2016 - [warning] relay\_log\_purge=0 is not set on slave slavesql2(192.168.62.36:3306).  
Wed Aug 10 15:02:09 2016 - [warning] relay\_log\_purge=0 is not set on slave slavesql1(192.168.62.41:3306).  
Wed Aug 10 15:02:09 2016 - [info] Checking replication filtering settings..  
Wed Aug 10 15:02:09 2016 - [info] binlog\_do\_db= , binlog\_ignore\_db=  
Wed Aug 10 15:02:09 2016 - [info] Replication filtering check ok.  
Wed Aug 10 15:02:09 2016 - [info] GTID (with auto-pos) is not supported  
Wed Aug 10 15:02:09 2016 - [info] Starting SSH connection tests..  
Wed Aug 10 15:02:12 2016 - [info] All SSH connection tests passed successfully.  
Wed Aug 10 15:02:12 2016 - [info] Checking MHA Node version..  
Wed Aug 10 15:02:13 2016 - [info] Version check ok.  
Wed Aug 10 15:02:13 2016 - [info] Checking SSH publickey authentication settings on the current master..  
Wed Aug 10 15:02:13 2016 - [info] HealthCheck: SSH to mastersql is reachable.  
Wed Aug 10 15:02:14 2016 - [info] Master MHA Node version is 0.54.  
Wed Aug 10 15:02:14 2016 - [info] Checking recovery script configurations on mastersql(192.168.62.42:3306)..  
Wed Aug 10 15:02:14 2016 - [info] Executing command: save\_binary\_logs --command=test --start\_pos=4  
--binlog\_dir=/var/lib/mysql --output\_file=/var/tmp/save\_binary\_logs\_test --manager\_version=0.56  
--start\_file=realcloud.000010  
Wed Aug 10 15:02:14 2016 - [info] Connecting to root@192.168.62.42(mastersql:22)..  
Creating /var/tmp if not exists.. ok.  
Checking output directory is accessible or not..  
ok.  
Binlog found at /var/lib/mysql, up to realcloud.000010  
Wed Aug 10 15:02:14 2016 - [info] Binlog setting check done.  
Wed Aug 10 15:02:14 2016 - [info] Checking SSH publickey authentication and checking recovery script configurations

on all alive slave servers..

```
Wed Aug 10 15:02:14 2016 - [info]   Executing command : apply_diff_relay_logs --command=test --slave_user='root'
--slave_host=mastersql          --slave_ip=192.168.62.42          --slave_port=3306          --workdir=/var/tmp
--target_version=5.5.49-0ubuntu0.12.04.1-log --manager_version=0.56--relay_log_info=/var/lib/mysql/relay-log.info
--relay_dir=/var/lib/mysql/    --slave_pass=xxx
```

Wed Aug 10 15:02:14 2016 - [info] Connecting to root@192.168.62.42(mastersql:22)..

mysqlbinlog: unknown variable 'default-character-set=utf8'

mysqlbinlog version not found!

at /usr/bin/apply\_diff\_relay\_logs line 482

Wed Aug 10 15:02:15 2016 - [error][/usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln205] Slaves settings check failed!

Wed Aug 10 15:02:15 2016 - [error][/usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln413] Slave configuration failed.

Wed Aug 10 15:02:15 2016 - [error][/usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln424] Error happened on checking configurations. at /usr/local/bin/masterha\_check\_repl line 48

Wed Aug 10 15:02:15 2016 - [error][/usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln523] Error happened on monitoring servers.

Wed Aug 10 15:02:15 2016 - [info] Got exit code 1 (Not master dead).

MySQL Replication Health is NOT OK!

解决方法:

编码格式问题导致

在 master mysql 的配置文件中注释掉 default-character-set = utf8

```
root@mastersql:~# vim /etc/mysql/my.cn
```

```
[client]
```

```
port                = 3306
```

```
socket              = /var/run/mysqld/mysqld.sock
```

```
#default-character-set = utf8
```

## 2、主从健康检查出错问题 2

```
root@manager:/etc/mha# masterha_check_repl --conf=/etc/app1.cnf
```

```
Wed Aug 10 16:15:51 2016 - [info] Reading default configuration from /etc/masterha_default.cnf..
```

```
Wed Aug 10 16:15:51 2016 - [info] Reading application default configuration from /etc/app1.cnf..
```

```
Wed Aug 10 16:15:51 2016 - [info] Reading server configuration from /etc/app1.cnf..
```

```
Wed Aug 10 16:15:51 2016 - [info] MHA::MasterMonitor version 0.56.
```

```
Wed Aug 10 16:15:52 2016 - [info] GTID failover mode = 0
```

```
Wed Aug 10 16:15:52 2016 - [info] Dead Servers:
```

```
Wed Aug 10 16:15:52 2016 - [info] Alive Servers:
```

```
Wed Aug 10 16:15:52 2016 - [info]   mastersql(192.168.62.42:3306)
```

```
Wed Aug 10 16:15:52 2016 - [info]   slavesql2(192.168.62.36:3306)
```

```
Wed Aug 10 16:15:52 2016 - [info]   slavesql1(192.168.62.41:3306)
```

```
Wed Aug 10 16:15:52 2016 - [info] Alive Slaves:
```

```
Wed Aug 10 16:15:52 2016 - [info]   mastersql(192.168.62.42:3306) Version=5.5.49-0ubuntu0.12.04.1-log (oldest
major version betweenslaves) log-bin:enabled
```

```
Wed Aug 10 16:15:52 2016 - [info]       Replicating from 192.168.62.42(192.168.62.42:3306)
```

```
Wed Aug 10 16:15:52 2016 - [info]   slavesql2(192.168.62.36:3306) Version=5.5.50-0ubuntu0.12.04.1-log (oldest
major version betweenslaves) log-bin:enabled
```

```
Wed Aug 10 16:15:52 2016 - [info]       Replicating from 192.168.62.42(192.168.62.42:3306)
```

Wed Aug 10 16:15:52 2016 - [info] Primary candidate for the new Master (candidate\_master is set)

Wed Aug 10 16:15:52 2016 - [info] slavesql1(192.168.62.41:3306) Version=5.5.49-0ubuntu0.12.04.1-log (oldest major version between slaves) log-bin:enabled

Wed Aug 10 16:15:52 2016 - [info] Replicating from 192.168.62.42(192.168.62.42:3306)

Wed Aug 10 16:15:52 2016 - [info] Not candidate for the new Master (no\_master is set)

Wed Aug 10 16:15:52 2016 - [info] Current Alive Master: mastersql(192.168.62.42:3306)

Wed Aug 10 16:15:52 2016 - [info] Checking slave configurations..

Wed Aug 10 16:15:52 2016 - [info] read\_only=1 is not set on slave mastersql(192.168.62.42:3306).

Wed Aug 10 16:15:52 2016 - [warning] relay\_log\_purge=0 is not set on slave mastersql(192.168.62.42:3306).

Wed Aug 10 16:15:52 2016 - [info] read\_only=1 is not set on slave slavesql2(192.168.62.36:3306).

Wed Aug 10 16:15:52 2016 - [warning] relay\_log\_purge=0 is not set on slave slavesql2(192.168.62.36:3306).

Wed Aug 10 16:15:52 2016 - [warning] relay\_log\_purge=0 is not set on slave slavesql1(192.168.62.41:3306).

Wed Aug 10 16:15:52 2016 - [info] Checking replication filtering settings..

Wed Aug 10 16:15:52 2016 - [info] binlog\_do\_db= , binlog\_ignore\_db=

Wed Aug 10 16:15:52 2016 - [info] Replication filtering check ok.

Wed Aug 10 16:15:52 2016 - [info] GTID (with auto-pos) is not supported

Wed Aug 10 16:15:52 2016 - [info] Starting SSH connection tests..

Wed Aug 10 16:15:54 2016 - [info] All SSH connection tests passed successfully.

Wed Aug 10 16:15:54 2016 - [info] Checking MHA Node version..

Wed Aug 10 16:15:55 2016 - [info] Version check ok.

Wed Aug 10 16:15:55 2016 - [info] Checking SSH publickey authentication settings on the current master..

Wed Aug 10 16:15:56 2016 - [info] HealthCheck: SSH to mastersql is reachable.

Wed Aug 10 16:15:56 2016 - [info] Master MHA Node version is 0.54.

Wed Aug 10 16:15:56 2016 - [info] Checking recovery script configurations on mastersql(192.168.62.42:3306)..

Wed Aug 10 16:15:56 2016 - [info] Executing command: save\_binary\_logs --command=test --start\_pos=4 --binlog\_dir=/var/lib/mysql --output\_file=/var/tmp/save\_binary\_logs\_test --manager\_version=0.56 --start\_file=realcloud.000011

Wed Aug 10 16:15:56 2016 - [info] Connecting to root@192.168.62.42(mastersql:22)..

Creating /var/tmp if not exists.. ok.

Checking output directory is accessible or not..

ok.

Binlog found at /var/lib/mysql, up to realcloud.000011

Wed Aug 10 16:15:57 2016 - [info] Binlog setting check done.

Wed Aug 10 16:15:57 2016 - [info] Checking SSH publickey authentication and checking recovery script configurations on all alive slave servers..

Wed Aug 10 16:15:57 2016 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=mastersql --slave\_ip=192.168.62.42 --slave\_port=3306 --workdir=/var/tmp --target\_version=5.5.49-0ubuntu0.12.04.1-log --manager\_version=0.56--relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxx

Wed Aug 10 16:15:57 2016 - [info] Connecting to root@192.168.62.42(mastersql:22)..

Checking slave recovery environment settings..

Opening /var/lib/mysql/relay-log.info ... ok.

Relay log found at /var/lib/mysql, up to mysqld-relay-bin.000002

Temporary relay log file is /var/lib/mysql/mysqld-relay-bin.000002

Testing mysql connection and privileges.. done.

Testing mysqlbinlog output.. done.

Cleaning up test file(s).. done.

```
Wed Aug 10 16:15:57 2016 - [info]   Executing command : apply_diff_relay_logs --command=test --slave_user='root'
--slave_host=slavesql2           --slave_ip=192.168.62.36           --slave_port=3306           --workdir=/var/tmp
--target_version=5.5.50-0ubuntu0.12.04.1-log --manager_version=0.56--relay_log_info=/var/lib/mysql/relay-log.info
--relay_dir=/var/lib/mysql/ --slave_pass=xxx
```

Wed Aug 10 16:15:57 2016 - [info] Connecting to root@192.168.62.36(slavesql2:22)..

Checking slave recovery environment settings..

Opening /var/lib/mysql/relay-log.info ... ok.

Relay log found at /var/lib/mysql, up to mysqld-relay-bin.000004

Temporary relay log file is /var/lib/mysql/mysqld-relay-bin.000004

Testing mysql connection and privileges.. done.

Testing mysqlbinlog output.. done.

Cleaning up test file(s).. done.

```
Wed Aug 10 16:15:57 2016 - [info]   Executing command : apply_diff_relay_logs --command=test --slave_user='root'
--slave_host=slavesql1           --slave_ip=192.168.62.41           --slave_port=3306           --workdir=/var/tmp
--target_version=5.5.49-0ubuntu0.12.04.1-log --manager_version=0.56--relay_log_info=/var/lib/mysql/relay-log.info
--relay_dir=/var/lib/mysql/ --slave_pass=xxx
```

Wed Aug 10 16:15:57 2016 - [info] Connecting to root@192.168.62.41(slavesql1:22)..

Checking slave recovery environment settings..

Opening /var/lib/mysql/relay-log.info ... ok.

Relay log found at /var/lib/mysql, up to mysqld-relay-bin.000005

Temporary relay log file is /var/lib/mysql/mysqld-relay-bin.000005

Testing mysql connection and privileges.. done.

Testing mysqlbinlog output.. done.

Cleaning up test file(s).. done.

Wed Aug 10 16:15:58 2016 - [info] Slaves settings check done.

Wed Aug 10 16:15:58 2016 - [info]

mastersql(192.168.62.42:3306) (current master)

+--mastersql(192.168.62.42:3306)

+--slavesql2(192.168.62.36:3306)

+--slavesql1(192.168.62.41:3306)

Wed Aug 10 16:15:58 2016 - [info] Checking replication health on mastersql..

Wed Aug 10 16:15:58 2016 - [error][usr/local/share/perl/5.14.2/MHA/Server.pm, ln485] **Slave IO thread is not running on mastersql(192.168.62.42:3306)**

Wed Aug 10 16:15:58 2016 - [error][usr/local/share/perl/5.14.2/MHA/ServerManager.pm, ln1526] failed!

Wed Aug 10 16:15:58 2016 - [error][usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln424] Error happened on checking configurations. at /usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm line 417

Wed Aug 10 16:15:58 2016 - [error][usr/local/share/perl/5.14.2/MHA/MasterMonitor.pm, ln523] Error happened on monitoring servers.

Wed Aug 10 16:15:58 2016 - [info] Got exit code 1 (Not master dead).

MySQL Replication Health is NOT OK!

解决方式:

清除 master 的从设置（之前不小心设置从模式且同步自己，真是自己坑自己）

mysql> stop slave;

Query OK, 0 rows affected, 1 warning (0.00 sec)

```
mysql> slave reset;  
mysql> reset slave;  
Query OK, 0 rows affected (0.02 sec)
```

主 MySQL 上操作:

```
root@mastersql:~# /etc/init.d/mysql restart
```

Rather than invoking init scripts through /etc/init.d, use the service(8) utility, e.g. service mysql restart

Since the script you are attempting to invoke has been converted to an Upstart job, you may also use the stop(8) and then start(8) utilities, e.g. stop mysql ; start mysql. The restart(8) utility is also available.

mysql stop/waiting

mysql start/running, process 10979

```
mysql> show slave status\G;
```

Empty set (0.00 sec)

ERROR:

No query specified