**基于MHA的MySQL高可用方案**

# 一、MHA简介

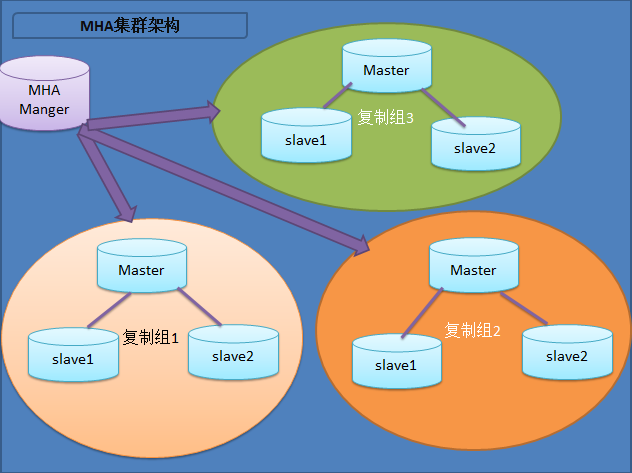
**MHA（Master High Availability）**目前在MySQL高可用方面是一个相对成熟的解决方案，是一套优秀的作为MySQL高可用性环境下故障切换和主从角色提升的高可用软件。在MySQL故障切换过程中，MHA能做到在0~30秒之内自动完成数据库的主从故障切换操作，并且在进行故障切换的过程中，MHA能在最大程度上保证数据的一致性，以达到真正意义上的高可用。

**MHA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服务器及Manager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，它可以自动将拥有最新数据的slave提升为新的master，然后将所有其他的slave重新指向新提升的master。整个故障转移过程对应用程序层面完全透明。

在MHA自动故障切换过程中，MHA会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不丢失，但这种操作是有概率性的。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，只进行故障转移从而丢失了最新的数据。使用MySQL 5.5的半同步复制，可以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MySQL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.

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



## 1.1、工作流程

（1）从宕机崩溃的master上尝试保存二进制日志事件（binlog events）;

（2）识别含有最新更新的slave服务器；

（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服务器；

（6）将其他的slave连接指向新的master进行主从复制；

## 1.2、MHA工具介绍

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：

* masterha\_check\_ssh 检查MHA的SSH配置状况
* masterha\_check\_repl 检查MySQL复制状况
* masterha\_manger 启动MHA
* masterha\_check\_status 检测当前MHA运行状态
* masterha\_master\_monitor 检测master是否宕机
* masterha\_master\_switch 控制故障转移（自动或者手动）
* 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）**

## 1.3、MHA环境说明

所有操作系统均为centos 7.x 64bit，涉及到主从复制环境搭建后面会简单演示步骤，但是相关的安全复制不会详细说明，[MySQL Replication需要注意的问题](http://www.cnblogs.com/gomysql/p/3662492.html)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 角色 | IP地址 | 主机名 | ServerID | 数据库类型 |
| Primary Master | 192.168.200.111 | server01 | 1 | 写入 |
| Secondary Master | 192.168.200.112 | server02 | 2 | 写入 |
| Slave1 | 192.168.200.113 | server03 | 3 | 读 |
| Slave2 | 192.168.200.114 | server04 | 4 | 读 |
| Manager | 192.168.200.115 | server05 | - | 监控复制组 |

其中Primary Master对外提供写服务，备选Secondary Master实际的slave提供读服务，slave1和slave2也提供相关的读服务，一旦Primary Master宕机，将会把备选Secondary Master提升为新的Primary Master，slave1和slave2指向新的master。

# 二、前期环境部署

## 2.1、配置所有主机名称

**master1主机:**

**hostname server01**

**bash**

**cat <<END>/etc/hostname #配置主机名永久更改**

**server01**

**END**

**rm -rf /root/.ssh/ #删除ssh目录，重新配置ssh以免冲突**

**master2主机:**

**hostname server02**

**bash**

**cat <<END>/etc/hostname**

**server02**

**END**

**rm -rf /root/.ssh/**

**slave1 主机:**

**hostname server03**

**bash**

**cat <<END>/etc/hostname**

**server03**

**END**

**rm -rf /root/.ssh/**

**slave2 主机:**

**hostname server04**

**bash**

**cat <<END>/etc/hostname**

**server04**

**END**

**rm -rf /root/.ssh/**

**manager主机:**

**hostname server05**

**bash**

**cat <<END>/etc/hostname**

**server05**

**END**

**rm -rf /root/.ssh/**

## 2.2、配置所有主机名映射

**[root@server05 ~]# cat << END >> /etc/hosts**

**192.168.200.111 server01**

**192.168.200.112 server02**

**192.168.200.113 server03**

**192.168.200.114 server04**

**192.168.200.115 server05**

**END**

## 2.3、所有主机关闭防火墙和安全机制

**systemctl stop iptables**

**systemctl stop firewalld**

**systemctl disable firewalld**

**setenforce 0**

**iptables -F**

# 三、配置SSH密钥对验证

服务器之间需要实现密钥对验证。关于配置密钥对验证可看下面步骤。但是有一点需要注意：不能禁止 password 登陆，否则会出现错误。

1. 服务器先生成一个密钥对
2. 把自己的公钥传给对方

为了防止以前有过同地址的主机冲突，建议将所有主机的/root/.ssh/\*中文件先删除

**3.1、Server05(192.168.200.115)上:**

**ssh-keygen**

**ssh-copy-id** [**root@192.168.200.111**](mailto:root@192.168.200.202)

**ssh-copy-id** [**root@192.168.200.112**](mailto:root@192.168.200.60)

**ssh-copy-id** [**root@192.168.200.113**](mailto:root@192.168.200.70)

**ssh-copy-id** [**root@192.168.200.114**](mailto:root@192.168.200.80)

**注意：Server05需要连接每个主机名，因为第一次连接的时候需要输入yes，影响后期故障切换时，对于每个主机的SSH控制，所以需要预测试一下，以保存各个server的信息。**

**ssh server01**

**exit**

**ssh server02**

**exit**

**ssh server03**

**exit**

**ssh server04**

**exit**

**[root@server05 ~]# cat /root/.ssh/known\_hosts**

**192.168.200.111 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEZKxgMVIxQRASe/xXYqdu1MEZKDvFxx77XmoEBtMSQ93adDk+4iPu2JeKQzm2J/QK/YEU18TvN3QBt5uDZmHAI=**

**192.168.200.112 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLeZezLr3su+2dvq6enTWi46nfNmJoeZeKRWzvUS4MrLcUqzsWzPnxdHdR8/VteCY+MB0lfFQ6fhnckGhi5u0b0=**

**192.168.200.113 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIvuNVBCVFMzu4kIe2Dfs5IrGyIRsp+EDLj0M29unejbgcAo2w1S/k12mdRoDjBBzVtM1EtbN53foH0fzUgzcEc=**

**192.168.200.114 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIvuNVBCVFMzu4kIe2Dfs5IrGyIRsp+EDLj0M29unejbgcAo2w1S/k12mdRoDjBBzVtM1EtbN53foH0fzUgzcEc=**

**server01 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEZKxgMVIxQRASe/xXYqdu1MEZKDvFxx77XmoEBtMSQ93adDk+4iPu2JeKQzm2J/QK/YEU18TvN3QBt5uDZmHAI=**

**server02 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLeZezLr3su+2dvq6enTWi46nfNmJoeZeKRWzvUS4MrLcUqzsWzPnxdHdR8/VteCY+MB0lfFQ6fhnckGhi5u0b0=**

**server03 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIvuNVBCVFMzu4kIe2Dfs5IrGyIRsp+EDLj0M29unejbgcAo2w1S/k12mdRoDjBBzVtM1EtbN53foH0fzUgzcEc=**

**server04 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIvuNVBCVFMzu4kIe2Dfs5IrGyIRsp+EDLj0M29unejbgcAo2w1S/k12mdRoDjBBzVtM1EtbN53foH0fzUgzcEc=**

**3.2、Primary Master(192.168.200.111):**

**ssh-keygen**

**ssh-copy-id** [**root@192.168.200.112**](mailto:root@192.168.200.202)

**ssh-copy-id** [**root@192.168.200.113**](mailto:root@192.168.200.202)

**ssh-copy-id** [**root@192.168.200.**](mailto:root@192.168.200.)**114**

**3.3、Secondary Master(192.168.200.112):**

**ssh-keygen**

**ssh-copy-id** [**root@192.168.200.111**](mailto:root@192.168.200.50)

**ssh-copy-id** [**root@192.168.200.113**](mailto:root@192.168.200.202)

**ssh-copy-id** [**root@192.168.200.**](mailto:root@192.168.200.)**114**

**3.4、slave1(192.168.200.113):**

**ssh-keygen**

**ssh-copy-id** [**root@192.168.200.111**](mailto:root@192.168.200.50)

**ssh-copy-id** [**root@192.168.200.112**](mailto:root@192.168.200.202)

**ssh-copy-id** [**root@192.168.200.114**](mailto:root@192.168.200.80)

**3.5、slave2(192.168.200.114):**

**ssh-keygen**

**ssh-copy-id** [**root@192.168.200.111**](mailto:root@192.168.200.50)

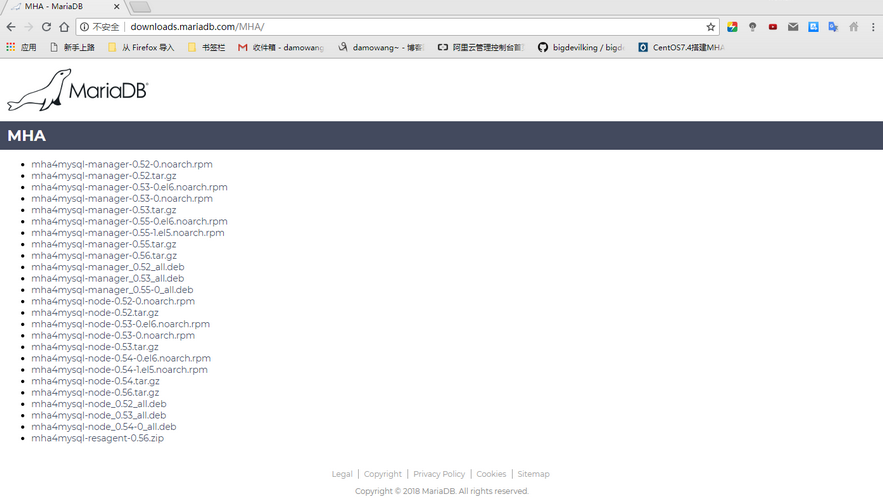
**ssh-copy-id** [**root@192.168.200.112**](mailto:root@192.168.200.60)

**ssh-copy-id** [**root@192.168.200.**](mailto:root@192.168.200.)**113**

# 四、安装MHA node

# 下载mha-manager和mha-node(已提前下载好)

[**http://downloads.mariadb.com/MHA/**](http://downloads.mariadb.com/MHA/)



**4.1、所有主机安装MHA node相关perl依赖包**

本实验所有软件包都已经提前下载好，配置好本地yum文件，指向软件目录即可以安装

**[root@server01 ~]# cd /etc/yum.repos.d/**

**rm -rf \***

**cat <<END>y.repo**

**[y]**

**name=y**

**baseurl=file:///usr/src/mha**

**gpgcheck=0**

**END**

**#为了方便管理，创建一个/usr/src/mha目录（与yum配置文件对应），将所有文件下载到/usr/src/mha目录中**

**[root@server01 ~]# mkdir /usr/src/mha**

**cd /usr/src/mha**

**rz**

**#生成yum仓库数据库文件，将在目录下生成repodata目录**

**[root@server01 mha]# createrepo .**

**Spawning worker 0 with 90 pkgs**

**Workers Finished**

**Saving Primary metadata**

**Saving file lists metadata**

**Saving other metadata**

**Generating sqlite DBs**

**Sqlite DBs complete**

**[root@server05 mha]# ls repodata/**

**4d60d2749096ada857bdbd4e0cb9296f1d46eab79da5cea0cb5952f2774eb38c-primary.xml.gz**

**5a1c9cbf244c613a0dc463d175674beed45ee6256a79f32268e41b6199330503-other.sqlite.bz2**

**ac44671e66836484a384ac1f74c7e2377c328de69fa773ddb1473fb0fbf81107-filelists.xml.gz**

**cb460a0e225aa92df99ae030802c46d348c226d0bda71399eafbb365d8c520dd-filelists.sqlite.bz2**

**dbf3d46c3ed22b9c6cf1698502253d9db84fb40ffb5effb505ed4999c3f46509-other.xml.gz**

**de4b3a24ce28eed6cf0a5475eaf08750822b42c0bf6706a78183969eef9d80ef-primary.sqlite.bz2**

**repomd.xml**

**[root@localhost mha]# yum -y localinstall \*.rpm （安装了所有的包，包括mariadb）**

**[root@server02 mha]# rpm -ivh --force --nodeps \*.rpm （强制安装所有rpm包）**

检查所需软件包是否全部安装

**[root@localhost mha]# rpm -q perl-DBD-MySQL.x86\_64 perl-DBI.x86\_64 perl-CPAN perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker**

**perl-DBD-MySQL-4.023-6.el7.x86\_64**

**perl-DBI-1.627-4.el7.x86\_64**

**perl-CPAN-1.9800-292.el7.noarch**

**perl-ExtUtils-CBuilder-0.28.2.6-292.el7.noarch**

**perl-ExtUtils-MakeMaker-6.68-3.el7.noarch**

**4.2、所有主机上安装MHA Node**

**[root@server05 mha]# cd /usr/src/mha**

**[root@localhost mha]#tar xf mha4mysql-node-0.56.tar.gz**

**[root@localhost mha]#cd mha4mysql-node-0.56/**

**[root@localhost mha4mysql-node-0.56]# perl Makefile.PL &&make && make install**

**4.3、MHA Node安装完后会在 /usr/local/bin生成以下脚本**

**ls -l /usr/local/bin/**

总用量 40

-r-xr-xr-x 1 root root 16346 12月 29 15:07 apply\_diff\_relay\_logs

-r-xr-xr-x 1 root root 4807 12月 29 15:07 filter\_mysqlbinlog

-r-xr-xr-x 1 root root 7401 12月 29 15:07 purge\_relay\_logs

-r-xr-xr-x 1 root root 7395 12月 29 15:07 save\_binary\_logs

# 五、安装MHA Manger（115）

注意：安装MHA Manger之前也需要安装MHA Node

**5.1、首先安装MHA Manger依赖的perl模块**

**[root@localhost mha]# yum -y localinstall \*.rpm #安装node时候已经安装了，无需再做**

**[root@localhost mha]#rpm -q perl perl-CPAN perl-Log-Dispatch perl-Parallel-ForkManager perl-DBD-MySQL perl-DBI perl-Time-HiRes perl-Config-Tiny**

**perl-5.16.3-294.el7\_6.x86\_64**

**perl-CPAN-1.9800-292.el7.noarch**

**perl-Log-Dispatch-2.41-1.el7.1.noarch**

**perl-Parallel-ForkManager-1.18-2.el7.noarch**

**perl-DBD-MySQL-4.023-6.el7.x86\_64**

**perl-DBI-1.627-4.el7.x86\_64**

**perl-Time-HiRes-1.9725-3.el7.x86\_64**

**perl-Config-Tiny-2.14-7.el7.noarch**

**5.2、安装MHA Manger软件包**

**[root@server05 mha]# cd /usr/src/mha**

**[root@server05 mha]#****tar xf mha4mysql-manager-0.56.tar.gz**

**cd mha4mysql-manager-0.56/**

**perl Makefile.PL**

**make && make install**

**5.3、安装完成后会有以下脚本文件**

**ls -l /usr/local/bin/**

**总用量 76**

**-r-xr-xr-x 1 root root 16346 12月 29 15:08 apply\_diff\_relay\_logs**

**-r-xr-xr-x 1 root root 4807 12月 29 15:08 filter\_mysqlbinlog**

**-r-xr-xr-x 1 root root 1995 12月 29 15:37 masterha\_check\_repl**

**-r-xr-xr-x 1 root root 1779 12月 29 15:37 masterha\_check\_ssh**

**-r-xr-xr-x 1 root root 1865 12月 29 15:37 masterha\_check\_status**

**-r-xr-xr-x 1 root root 3201 12月 29 15:37 masterha\_conf\_host**

**-r-xr-xr-x 1 root root 2517 12月 29 15:37 masterha\_manager**

**-r-xr-xr-x 1 root root 2165 12月 29 15:37 masterha\_master\_monitor**

**-r-xr-xr-x 1 root root 2373 12月 29 15:37 masterha\_master\_switch**

**-r-xr-xr-x 1 root root 3879 12月 29 15:37 masterha\_secondary\_check**

**-r-xr-xr-x 1 root root 1739 12月 29 15:37 masterha\_stop**

**-r-xr-xr-x 1 root root 7401 12月 29 15:08 purge\_relay\_logs**

**-r-xr-xr-x 1 root root 7395 12月 29 15:08 save\_binary\_logs**

# 六、安装mysql

**111-115主机上的操作（安装node那步已经全部安装了,115上安装只是为了有一个mysql指令做一个客户端测试，不需要启动）：**

**yum -y install mariadb mariadb-server mariadb-devel （此步已做，省略）**

**systemctl start mariadb**

**netstat -lnpt | grep :3306**

**mysqladmin -u root password 123456 #设置数据库初始密码（后续操作中使用）**

# 七、搭建主从复制环境

注意：binlog-do-db 和 replicate-ignore-db 设置必须相同(默认未设置)。 MHA 在启动时候会检测过滤规则，如果过滤规则不同，MHA 将不启动监控和故障转移功能。

## 7.1、修改mysql主机的配置文件

**Primary Master(192.168.200.111):**

**vim /etc/my.cnf**

**[mysqld]**

**server-id = 1**

**log-bin=master-bin**

**log-slave-updates=true**

**relay\_log\_purge=0** #禁止 SQL 线程在执行完一个 relay log 后自动将其删除

**systemctl restart mariadb**

**Secondary Master(192.168.200.112):**

**vim /etc/my.cnf**

**[mysqld]**

**server-id = 2**

**log-bin=master-bin**

**log-slave-updates=true**

**relay\_log\_purge=0**

**systemctl restart mariadb**

**slave1(192.168.200.113):**

**vim /etc/my.cnf**

**[mysqld]**

**server-id=3**

**log-bin=mysql-bin**

**relay-log=slave-relay-bin**

**log-slave-updates=true**

**relay\_log\_purge=0**

**systemctl restart mariadb**

**slave2(192.168.200.114):**

**vim /etc/my.cnf**

**[mysqld]**

**server-id=4**

**log-bin=mysql-bin**

**relay-log=slave-relay-bin**

**log-slave-updates=true**

**relay\_log\_purge=0**

**systemctl restart mariadb**

## 7.2、mysql服务器创建复制授权用户

所有数据库服务器

[root@server01 mariadb]# mysql -p123456

**grant replication slave on \*.\* to 'repl'@'192.168.200.%' identified by '123456';**

**flush privileges;**

## 7.3、查看主库备份时的binlog名称和位置

**MariaDB [(none)]> show master status;**

**+-------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+-------------------+----------+--------------+------------------+**

**| master-bin.000001 | 474 | | |**

**+-------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

## 7.4、112-114主机上执行复制相关命令

**stop slave;**

**CHANGE MASTER TO MASTER\_HOST='192.168.200.111', MASTER\_USER='repl', MASTER\_PASSWORD='123456', MASTER\_LOG\_FILE='master-bin.000001', MASTER\_LOG\_POS=474;**

**start slave;**

**show slave status\G**

**# 检查IO和SQL线程是否为：yes**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

## 7.5、主从同步故障处理（无故障忽略此步）

**Slave\_IO\_Running: No**

**Slave\_SQL\_Running: Yes**

**-----------------------------------忽略部分信息----------------------------------- Last\_IO\_Errno: 1236**

**Last\_IO\_Error: Got fatal error 1236 from master when reading data from binary log: 'Could not find first log file name in binary log index file' -----------------------------------忽略部分信息-----------------------------------**

**处理方式：**

**stop slave;**

**reset slave;**

**set global sql\_slave\_skip\_counter =1 ;**

**start slave;**

## 7.6、三台slave服务器设置read\_only状态

**从库对外只提供读服务，只所以没有写进mysql配置文件，是因为随时server02会提升为master**

**[root@server02 ~]# mysql -uroot -p123456 -e 'set global read\_only=1;'**

**或者直接在数据库里执行：MariaDB [(none)]> set global read\_only=1;**

**[root@server03 ~]# mysql -uroot -p123456 -e 'set global read\_only=1;'**

**[root@server04 ~]# my****sql -uroot -p123456 -e 'set global read\_only=1;'**

## 7.7、创建监控用户（111-114主机上的操作）：

**grant all privileges on \*.\* to 'root'@'192.168.200.%' identified by '123456';**

**grant all privileges on \*.\* to 'root'@'server0x' identified by '123456';**

**flush privileges;**

#注意：每个数据库服务器要为自己的主机名和server05授权，其中x代表每台服务器主机名编号

到这里整个mysql主从集群环境已经搭建完毕。

# 八、配置MHA环境

## 8.1、创建MHA的工作目录及相关配置文件

Server05(192.168.200.115)：**在软件包解压后的目录里面有样例配置文件**

**mkdir /etc/masterha**

**cd /usr/src/mha/**

**cp mha4mysql-manager-0.56/samples/conf/app1.cnf /etc/masterha**

## 8.2、修改app1.cnf配置文件

已经创建好，可以直接从mha目录复制

**vim /etc/masterha/app1.cnf**

**[server default]**

**#设置manager的工作日志**

**manager\_workdir=/var/log/masterha/app1**

**#设置manager的日志,这两条都是默认存在的**

**manager\_log=/var/log/masterha/app1/manager.log**

**#设置master默认保存binlog的位置,以便MHA可以找到master日志**

**master\_binlog\_dir=/var/lib/mysql**

**#设置自动failover时候的切换脚本**

**master\_ip\_failover\_script= /usr/local/bin/master\_ip\_failover**

**#设置mysql中root用户的密码**

**password=123456**

**user=root**

**#ping包的时间间隔**

**ping\_interval=1**

**#设置远端mysql在发生切换时保存binlog的具体位置**

**remote\_workdir=/tmp**

**#设置复制用户的密码和用户名**

**repl\_password=123456**

**repl\_user=repl**

**[server1]**

**hostname=server01**

**port=3306**

**[server2]**

**hostname=server02**

**#优先到新主人，即设置为候选master**

**candidate\_master=1**

**port=3306**

#忽略复制延迟（如果延迟过多不能成为新主）

**check\_repl\_delay=0**

**[server3]**

**hostname=server03**

**port=3306**

**[server4]**

**hostname=server04**

**port=3306**

## 8.3、配置故障转移脚本

**/usr/local/bin/master\_ip\_failover**脚本需要根据自己环境修改ip和网卡名称等。（已创建好，直接从mha目录复制即可）

**[root@server05 ~]# vim /usr/local/bin/master\_ip\_failover**

**#!/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 = '192.168.200.100'; # 写入VIP**

**my $key = "1"; #非keepalived方式切换脚本使用的**

**my $ssh\_start\_vip = "/sbin/ifconfig ens33:$key $vip";**

**my $ssh\_stop\_vip = "/sbin/ifconfig ens33:$key down"; #那么这里写服务的开关命令**

**$ssh\_user = "root";**

**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;**

**#};**

**eval {**

**print "Disabling the VIP on old master: $orig\_master\_host \n";**

**#my $ping=`ping -c 1 10.0.0.13 | grep "packet loss" | awk -F',' '{print $3}' | awk '{print $1}'`;**

**#if ( $ping le "90.0%"&& $ping gt "0.0%" ){**

**#$exit\_code = 0;**

**#}**

**#else {**

**&stop\_vip();**

**# updating global catalog, etc**

**$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 $ssh\_user\@$orig\_master\_ip \" $ssh\_start\_vip \"`;**

**exit 0;**

**}**

**else {**

**&usage();**

**exit 1;**

**}**

**}**

**# A simple system call that enable the VIP on the new master**

**sub start\_vip() {**

**`ssh $ssh\_user\@$new\_master\_host \" $ssh\_start\_vip \"`;**

**}**

**# A simple system call that disable the VIP on the old\_master**

**sub 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"; }**

**[root@server05 ~]# chmod +x /usr/local/bin/master\_ip\_failover**

## 8.4、设置从库relay log的清除方式(112-114)：

**[root@server02 mha]# mysql -uroot -p123456 -e 'set global relay\_log\_purge=0;'**

**或者在数据库里直接设置：MariaDB [(none)]> set global relay\_log\_purge=0;**

注意：

MHA在故障切换的过程中，从库的恢复过程依赖于relay log的相关信息，所以这里要将relay log的自动清除设置为OFF，采用手动清除relay log的方式。在默认情况下，从服务器上的中继日志会在SQL线程执行完毕后被自动删除。但是在MHA环境中，这些中继日志在恢复其他从服务器时可能会被用到，因此需要禁用中继日志的自动清除功能。定期清除中继日志需要考虑到复制延时的问题。在ext3的文件系统下，删除大的文件需要一定的时间，会导致严重的复制延时。为了避免复制延时，需要暂时为中继日志创建硬链接，因为在linux系统中通过硬链接删除大文件速度会很快。（在mysql数据库中，删除大表时，通常也采用建立硬链接的方式）

## 8.5、配置从库（112-114）relay\_log清除脚本加入计划任务（实验时可省略）

MHA节点中包含了pure\_relay\_logs命令工具，它可以为中继日志创建硬链接，执行SET GLOBAL relay\_log\_purge=1,等待几秒钟以便SQL线程切换到新的中继日志，再执行SET GLOBAL relay\_log\_purge=0。

**vim purge\_relay\_log.sh**

**#!/bin/bash**

**user=root**

**passwd=123456 #注意：数据库要有密码,填自己所设置的密码就可以，前面设置过**

**port=3306**

**log\_dir='/tmp'**

**work\_dir='/tmp'**

**purge='/usr/local/bin/purge\_relay\_logs'**

**if [ ! -d $log\_dir ]**

**then**

**mkdir $log\_dir -p**

**fi**

**$purge --user=$user --password=$passwd --disable\_relay\_log\_purge --port=$port --workdir=$work\_dir >> $log\_dir/purge\_relay\_logs.log 2>&1**

**chmod +x purge\_relay\_log.sh**

**crontab -e**

**0 4 \* \* \* /bin/bash /root/purge\_relay\_log.sh**

**pure\_relay\_logs脚本参数如下所示：**

--user mysql 用户名

--password mysql 密码

--port 端口号

--workdir 指定创建relay log的硬链接的位置，默认是/var/tmp，由于系统不同分区创建硬链接文件会失败，故需要执行硬链接具体位置，成功执行脚本后，硬链接的中继日志文件被删除

--disable\_relay\_log\_purge 默认情况下，如果relay\_log\_purge=1，脚本会什么都不清理，自动退出，通过设定这个参数，当relay\_log\_purge=1的情况下会将relay\_log\_purge设置为0。清理relay log之后，最后将参数设置为OFF。

## 8.6、手动清除中继日志（实验时可省略）

**purge\_relay\_logs --user=root --password=123456 --disable\_relay\_log\_purge --port=3306 --workdir=/tmp**

2017-08-31 21:33:52: purge\_relay\_logs script started.

Found relay\_log.info: /usr/local/mysql/data/relay-log.info

Removing hard linked relay log files slave-relay-bin\* under /tmp.. done.

Current relay log file: /usr/local/mysql/data/slave-relay-bin.000002

Archiving unused relay log files (up to /usr/local/mysql/data/slave-relay-bin.000001) ...

Creating hard link for /usr/local/mysql/data/slave-relay-bin.000001 under /tmp/slave-relay-bin.000001 .. ok.

Creating hard links for unused relay log files completed.

Executing SET GLOBAL relay\_log\_purge=1; FLUSH LOGS; sleeping a few seconds so that SQL thread can delete older relay log files (if i

t keeps up); SET GLOBAL relay\_log\_purge=0; .. ok. Removing hard linked relay log files slave-relay-bin\* under /tmp.. done.

2017-08-31 21:33:56: All relay log purging operations succeeded.

## 8.7、检查MHA ssh通信状态

**[root@server05 ~]# masterha\_check\_ssh --conf=/etc/masterha/app1.cnf**

**Sat Dec 29 16:03:57 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.**

**Sat Dec 29 16:03:57 2018 - [info] Reading application default configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:03:57 2018 - [info] Reading server configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:03:57 2018 - [info] Starting SSH connection tests..**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:58 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:03:59 2018 - [debug] ok.**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:58 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:01 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:57 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:03:59 2018 - [debug] ok.**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:01 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [info] All SSH connection tests passed successfully.**

最后会返回 successfully表示没有问题

## 8.9、检查整个集群的状态

**[root@server05 ~]# masterha\_check\_repl --conf=/etc/masterha/app1.cnf**

**Sat Dec 29 16:03:57 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.**

**Sat Dec 29 16:03:57 2018 - [info] Reading application default configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:03:57 2018 - [info] Reading server configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:03:57 2018 - [info] Starting SSH connection tests..**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:58 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:03:59 2018 - [debug] ok.**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server02(192.168.200.112:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:58 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:01 2018 - [debug] Connecting via SSH from root@server03(192.168.200.113:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:57 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:03:59 2018 - [debug] ok.**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server01(192.168.200.111:22) to root@server04(192.168.200.114:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [debug]**

**Sat Dec 29 16:03:59 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server01(192.168.200.111:22)..**

**Sat Dec 29 16:04:00 2018 - [debug] ok.**

**Sat Dec 29 16:04:00 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server02(192.168.200.112:22)..**

**Sat Dec 29 16:04:01 2018 - [debug] ok.**

**Sat Dec 29 16:04:01 2018 - [debug] Connecting via SSH from root@server04(192.168.200.114:22) to root@server03(192.168.200.113:22)..**

**Sat Dec 29 16:04:02 2018 - [debug] ok.**

**Sat Dec 29 16:04:02 2018 - [info] All SSH connection tests passed successfully.**

**[root@server05 ~]# masterha\_check\_repl --conf=/etc/masterha/app1.cnf**

**[root@server05 ~]# masterha\_check\_repl --conf=/etc/masterha/app1.cnf**

**Sat Dec 29 16:04:53 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.**

**Sat Dec 29 16:04:53 2018 - [info] Reading application default configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:04:53 2018 - [info] Reading server configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:04:53 2018 - [info] MHA::MasterMonitor version 0.56.**

**Creating directory /var/log/masterha/app1.. done.**

**Sat Dec 29 16:04:55 2018 - [info] Dead Servers:**

**Sat Dec 29 16:04:55 2018 - [info] Alive Servers:**

**Sat Dec 29 16:04:55 2018 - [info] server01(192.168.200.111:3306)**

**Sat Dec 29 16:04:55 2018 - [info] server02(192.168.200.112:3306)**

**Sat Dec 29 16:04:55 2018 - [info] server03(192.168.200.113:3306)**

**Sat Dec 29 16:04:55 2018 - [info] server04(192.168.200.114:3306)**

**Sat Dec 29 16:04:55 2018 - [info] Alive Slaves:**

**Sat Dec 29 16:04:55 2018 - [info] server02(192.168.200.112:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:04:55 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:04:55 2018 - [info] Primary candidate for the new Master (candidate\_master is set)**

**Sat Dec 29 16:04:55 2018 - [info] server03(192.168.200.113:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:04:55 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:04:55 2018 - [info] server04(192.168.200.114:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:04:55 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:04:55 2018 - [info] Current Alive Master: server01(192.168.200.111:3306)**

**Sat Dec 29 16:04:55 2018 - [info] Checking slave configurations..**

**Sat Dec 29 16:04:55 2018 - [info] Checking replication filtering settings..**

**Sat Dec 29 16:04:55 2018 - [info] binlog\_do\_db= , binlog\_ignore\_db=**

**Sat Dec 29 16:04:55 2018 - [info] Replication filtering check ok.**

**Sat Dec 29 16:04:55 2018 - [info] Starting SSH connection tests..**

**Sat Dec 29 16:05:00 2018 - [info] All SSH connection tests passed successfully.**

**Sat Dec 29 16:05:00 2018 - [info] Checking MHA Node version..**

**Sat Dec 29 16:05:02 2018 - [info] Version check ok.**

**Sat Dec 29 16:05:02 2018 - [info] Checking SSH publickey authentication settings on the current master..**

**Sat Dec 29 16:05:03 2018 - [info] HealthCheck: SSH to server01 is reachable.**

**Sat Dec 29 16:05:04 2018 - [info] Master MHA Node version is 0.56.**

**Sat Dec 29 16:05:04 2018 - [info] Checking recovery script configurations on the current master..**

**Sat Dec 29 16:05:04 2018 - [info] Executing command: save\_binary\_logs --command=test --start\_pos=4 --binlog\_dir=/var/lib/mysql --output\_file=/tmp/save\_binary\_logs\_test --**

**manager\_version=0.56 --start\_file=master-bin.000001 Sat Dec 29 16:05:04 2018 - [info] Connecting to root@server01(server01)..**

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

**Checking output directory is accessible or not..**

**ok.**

**Binlog found at /var/lib/mysql, up to master-bin.000001**

**Sat Dec 29 16:05:04 2018 - [info] Master setting check done.**

**Sat Dec 29 16:05:04 2018 - [info] Checking SSH publickey authentication and checking recovery script configurations on all alive slave servers..**

**Sat Dec 29 16:05:04 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server02 --slave\_ip=192.168.200.112 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:05:04 2018 - [info] Connecting to root@192.168.200.112(server02:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:05:05 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server03 --slave\_ip=192.168.200.113 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:05:05 2018 - [info] Connecting to root@192.168.200.113(server03:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:05:07 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server04 --slave\_ip=192.168.200.114 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:05:07 2018 - [info] Connecting to root@192.168.200.114(server04:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:05:08 2018 - [info] Slaves settings check done.**

**Sat Dec 29 16:05:09 2018 - [info]**

**server01 (current master)**

**+--server02**

**+--server03**

**+--server04**

**Sat Dec 29 16:05:09 2018 - [info] Checking replication health on server02..**

**Sat Dec 29 16:05:09 2018 - [info] ok.**

**Sat Dec 29 16:05:09 2018 - [info] Checking replication health on server03..**

**Sat Dec 29 16:05:09 2018 - [info] ok.**

**Sat Dec 29 16:05:09 2018 - [info] Checking replication health on server04..**

**Sat Dec 29 16:05:09 2018 - [info] ok.**

**Sat Dec 29 16:05:09 2018 - [info] Checking master\_ip\_failover\_script status:**

**Sat Dec 29 16:05:09 2018 - [info] /usr/local/bin/master\_ip\_failover --command=status --ssh\_user=root --orig\_master\_host=server01 --orig\_master\_ip=192.168.200.111 --orig\_m**

**aster\_port=3306**

**IN SCRIPT TEST====/etc/init.d/keepalived stop==/etc/init.d/keepalived start===**

**Checking the Status of the script.. OK**

**bash: /etc/init.d/keepalived: 没有那个文件或目录**

**Sat Dec 29 16:05:09 2018 - [info] OK.**

**Sat Dec 29 16:05:09 2018 - [warning] shutdown\_script is not defined.**

**Sat Dec 29 16:05:09 2018 - [info] Got exit code 0 (Not master dead).**

MySQL Replication Health is OK.

返回OK表示没有问题

# 九、VIP配置管理

Master vip配置有两种方式,一种是通过keepalived或者heartbeat类似的软件的方式管理VIP的浮动,另一种为通过ifconfig命令方式管理。

通过ifconfig命令方式管理VIP地址：

打开在前面编辑过的文件**/etc/masterha/app1.cnf**，检查如下行是否正确，再检查集群状态。

**[root@server05 ~]# grep -n 'master\_ip\_failover\_script' /etc/masterha/app1.cnf**

**9:master\_ip\_failover\_script= /usr/local/bin/master\_ip\_failover**

Server05(192.168.200.115)修改故障转移脚本

**[root@server05 ~]# head -13 /usr/local/bin/master\_ip\_failover**

**#!/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 = '192.168.200.100'; # 写入VIP**

**my $key = "1"; #非keepalived方式切换脚本使用的**

**my $ssh\_start\_vip = "/sbin/ifconfig ens33:$key $vip"; #若是使用keepalived**

**my $ssh\_stop\_vip = "/sbin/ifconfig ens33:$key down"; #那么这里写服务的开关命令**

  /usr/local/bin/master\_ip\_failover文件的内容意思是当主库发生故障时，会触发MHA切换，MHA manager会停掉主库上的ens33:1接口，触发虚拟ip漂移到备选从库，从而完成切换。

Server05(192.168.200.115) 检查manager状态

**masterha\_check\_status --conf=/etc/masterha/app1.cnf**

app1 is stopped(2:NOT\_RUNNING).

注意：如果正常会显示"PING\_OK"，否则会显示"NOT\_RUNNING"，代表MHA监控没有开启。

Server05(192.168.200.115) 开启manager监控

**[root@server05 ~]# nohup masterha\_manager --conf=/etc/masterha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover< /dev/null >/var/log/masterha/app1/manager.log 2>&1 &**

**启动参数介绍：**

--remove\_dead\_master\_conf      该参数代表当发生主从切换后，老的主库的ip将会从配置文件中移除。

--manger\_log                            日志存放位置

--ignore\_last\_failover                  在缺省情况下，如果MHA检测到连续发生宕机，且两次宕机间隔不足8小时的话，则不会进行Failover，之所以这样限制是为了避免ping-pong效应。该参数代表忽略上次MHA触发切换产生的文件，默认情况下，MHA发生切换后会在日志目录，也就是上面我设置的/data产生app1.failover.complete文件，下次再次切换的时候如果发现该目录下存在该文件将不允许触发切换，除非在第一次切换后收到删除该文件，为了方便，这里设置为--ignore\_last\_failover。

Server05(192.168.200.115)查看Server05监控是否正常：

**[root@monitor ~]# masterha\_check\_status --conf=/etc/masterha/app1.cnf**

**app1 (pid:65837) is running(0:PING\_OK), master:server01**

可以看见已经在监控了

Server05(192.168.200.115)查看启动日志

**[root@server05 ~]# cat /var/log/masterha/app1/manager.log**

**Sat Dec 29 16:09:50 2018 - [warning] Global configuration file /etc/masterha\_default.cnf not found. Skipping.**

**Sat Dec 29 16:09:50 2018 - [info] Reading application default configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:09:50 2018 - [info] Reading server configurations from /etc/masterha/app1.cnf..**

**Sat Dec 29 16:09:50 2018 - [info] MHA::MasterMonitor version 0.56.**

**Sat Dec 29 16:09:51 2018 - [info] Dead Servers:**

**Sat Dec 29 16:09:51 2018 - [info] Alive Servers:**

**Sat Dec 29 16:09:51 2018 - [info] server01(192.168.200.111:3306)**

**Sat Dec 29 16:09:51 2018 - [info] server02(192.168.200.112:3306)**

**Sat Dec 29 16:09:51 2018 - [info] server03(192.168.200.113:3306)**

**Sat Dec 29 16:09:51 2018 - [info] server04(192.168.200.114:3306)**

**Sat Dec 29 16:09:51 2018 - [info] Alive Slaves:**

**Sat Dec 29 16:09:51 2018 - [info] server02(192.168.200.112:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:09:51 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:09:51 2018 - [info] Primary candidate for the new Master (candidate\_master is set)**

**Sat Dec 29 16:09:51 2018 - [info] server03(192.168.200.113:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:09:51 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:09:51 2018 - [info] server04(192.168.200.114:3306) Version=5.5.56-MariaDB (oldest major version between slaves) log-bin:enabled**

**Sat Dec 29 16:09:51 2018 - [info] Replicating from 192.168.200.111(192.168.200.111:3306)**

**Sat Dec 29 16:09:51 2018 - [info] Current Alive Master: server01(192.168.200.111:3306)**

**Sat Dec 29 16:09:51 2018 - [info] Checking slave configurations..**

**Sat Dec 29 16:09:51 2018 - [info] Checking replication filtering settings..**

**Sat Dec 29 16:09:51 2018 - [info] binlog\_do\_db= , binlog\_ignore\_db=**

**Sat Dec 29 16:09:51 2018 - [info] Replication filtering check ok.**

**Sat Dec 29 16:09:51 2018 - [info] Starting SSH connection tests..**

**Sat Dec 29 16:09:58 2018 - [info] All SSH connection tests passed successfully.**

**Sat Dec 29 16:09:58 2018 - [info] Checking MHA Node version..**

**Sat Dec 29 16:10:01 2018 - [info] Version check ok.**

**Sat Dec 29 16:10:01 2018 - [info] Checking SSH publickey authentication settings on the current master..**

**Sat Dec 29 16:10:02 2018 - [info] HealthCheck: SSH to server01 is reachable.**

**Sat Dec 29 16:10:03 2018 - [info] Master MHA Node version is 0.56.**

**Sat Dec 29 16:10:03 2018 - [info] Checking recovery script configurations on the current master..**

**Sat Dec 29 16:10:03 2018 - [info] Executing command: save\_binary\_logs --command=test --start\_pos=4 --binlog\_dir=/var/lib/mysql --output\_file=/tmp/save\_binary\_logs\_test --**

**manager\_version=0.56 --start\_file=master-bin.000001 Sat Dec 29 16:10:03 2018 - [info] Connecting to root@server01(server01)..**

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

**Checking output directory is accessible or not..**

**ok.**

**Binlog found at /var/lib/mysql, up to master-bin.000001**

**Sat Dec 29 16:10:03 2018 - [info] Master setting check done.**

**Sat Dec 29 16:10:03 2018 - [info] Checking SSH publickey authentication and checking recovery script configurations on all alive slave servers..**

**Sat Dec 29 16:10:03 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server02 --slave\_ip=192.168.200.112 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:10:03 2018 - [info] Connecting to root@192.168.200.112(server02:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:10:04 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server03 --slave\_ip=192.168.200.113 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:10:04 2018 - [info] Connecting to root@192.168.200.113(server03:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:10:05 2018 - [info] Executing command : apply\_diff\_relay\_logs --command=test --slave\_user='root' --slave\_host=server04 --slave\_ip=192.168.200.114 --slave\_po**

**rt=3306 --workdir=/tmp --target\_version=5.5.56-MariaDB --manager\_version=0.56 --relay\_log\_info=/var/lib/mysql/relay-log.info --relay\_dir=/var/lib/mysql/ --slave\_pass=xxxSat Dec 29 16:10:05 2018 - [info] Connecting to root@192.168.200.114(server04:22)..**

**Checking slave recovery environment settings..**

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

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

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

**Testing mysql connection and privileges.. done.**

**Testing mysqlbinlog output.. done.**

**Cleaning up test file(s).. done.**

**Sat Dec 29 16:10:06 2018 - [info] Slaves settings check done.**

**Sat Dec 29 16:10:06 2018 - [info]**

**server01 (current master)**

**+--server02**

**+--server03**

**+--server04**

**Sat Dec 29 16:10:06 2018 - [info] Checking master\_ip\_failover\_script status:**

**Sat Dec 29 16:10:06 2018 - [info] /usr/local/bin/master\_ip\_failover --command=status --ssh\_user=root --orig\_master\_host=server01 --orig\_master\_ip=192.168.200.111 --orig\_m**

**aster\_port=3306**

**IN SCRIPT TEST====/etc/init.d/keepalived stop==/etc/init.d/keepalived start===**

**Checking the Status of the script.. OK**

**bash: /etc/init.d/keepalived: 没有那个文件或目录**

**Sat Dec 29 16:10:07 2018 - [info] OK.**

**Sat Dec 29 16:10:07 2018 - [warning] shutdown\_script is not defined.**

**Sat Dec 29 16:10:07 2018 - [info] Set master ping interval 1 seconds.**

**Sat Dec 29 16:10:07 2018 - [warning] secondary\_check\_script is not defined. It is highly recommended setting it to check master reachability from two or more routes.**

**Sat Dec 29 16:10:07 2018 - [info] Starting ping health check on server01(192.168.200.111:3306)..**

**Thu Aug 31 21:55:23 2017 - [info] Ping(SELECT) succeeded, waiting until MySQL doesn't respond..**

注意：其中"Ping(SELECT) succeeded, waiting until MySQL doesn't respond.."说明整个系统已经开始监控了。

关闭MHA manager监控，忽略操作

**masterha\_stop --conf=/etc/masterha/app1.cnf**

发现已经将VIP：192.168.200.100绑定在网卡ens33。

**[root@server01 ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.111/24 brd 192.168.200.255 scope global ens33**

**inet 192.168.200.100/24 brd 192.168.200.255 scope global secondary ens33:1**

**可以从server05模拟客户端测试连接到群集ip地址**

**[root@server05 mha]# mysql -uroot -p123456 -h 192.168.200.100**

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

**Your MariaDB connection id is 19**

**Server version: 5.5.68-MariaDB MariaDB Server**

**Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.**

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

**MariaDB [(none)]>**

### Primary Master(192.168.200.111) 模拟主库故障

**[root@server01 ~]# systemctl stop mariadb**

**[root@server01 ~]# netstat -lnpt | grep :3306**

**[root@server01 ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.111/24 brd 192.168.200.255 scope global ens33**

slave1(192.168.200.113)状态：

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.112**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000001**

**Read\_Master\_Log\_Pos: 1372**

**Relay\_Log\_File: slave-relay-bin.000002**

**Relay\_Log\_Pos: 530**

**Relay\_Master\_Log\_File: master-bin.000001**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

slave2(192.168.200.114)状态：

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.112**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000001**

**Read\_Master\_Log\_Pos: 1372**

**Relay\_Log\_File: slave-relay-bin.000002**

**Relay\_Log\_Pos: 530**

**Relay\_Master\_Log\_File: master-bin.000001**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

Server05(192.168.200.115) 监控已经自动关闭：

**[root@server05 ~]#**

**[1]+ 完成 nohup masterha\_manager --conf=/etc/masterha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover < /dev/null > /var/log/masterha/app1/manager.log 2>&1**

Server05(192.168.200.115) 查看监控配置文件已经发生了变化（server01的配置已被删除）：

**[root@server05 ~]# cat /etc/masterha/app1.cnf**

**[server default]**

**manager\_log=/var/log/masterha/app1/manager.log**

**manager\_workdir=/var/log/masterha/app1**

**master\_binlog\_dir=/var/lib/mysql**

**master\_ip\_failover\_script=/usr/local/bin/master\_ip\_failover**

**password=123456**

**ping\_interval=1**

**remote\_workdir=/tmp**

**repl\_password=123456**

**repl\_user=repl**

**user=root**

**[server2]**

**candidate\_master=1**

**check\_repl\_delay=0**

**hostname=server02**

**port=3306**

**[server3]**

**hostname=server03**

**port=3306**

**[server4]**

**hostname=server04**

**port=3306**

Server05(192.168.200.115) 故障切换过程中的日志文件内容如下：

**[root@server05 ~]# tail -f /var/log/masterha/app1/manager.log**

**Selected server02 as a new master.**

**server02: OK: Applying all logs succeeded.**

**server02: OK: Activated master IP address.**

**server04: This host has the latest relay log events.**

**server03: This host has the latest relay log events.**

**Generating relay diff files from the latest slave succeeded.**

**server04: OK: Applying all logs succeeded. Slave started, replicating from server02.**

**server03: OK: Applying all logs succeeded. Slave started, replicating from server02.**

**server02: Resetting slave info succeeded.**

**Master failover to server02(192.168.200.112:3306) completed successfully.**

Server02已经是新的master，查看server02的ip地址发现VIP地址已经转移过来

**[root@server02 mha]# ip a|grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.112/24 brd 192.168.200.255 scope global noprefixroute ens33**

**inet 192.168.200.100/24 brd 192.168.200.255 scope global secondary ens33:1**

从server05模拟客户端测试访问vip地址仍然可以访问到

**[root@server05 mha]# mysql -uroot -p123456 -h 192.168.200.100**

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

**Your MariaDB connection id is 20**

**Server version: 5.5.68-MariaDB MariaDB Server**

**Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.**

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

**MariaDB [(none)]>**

### 故障主库修复及VIP切回测试

**Primary Master(192.168.200.111)：**

**[root@server01 ~]# systemctl start mariadb**

**[root@server01 ~]# netstat -lnpt | grep :3306**

**tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 6131/mysqld**

**Primary Master(192.168.200.111) 指向新的主库**

**[root@server01 ~]# mysql -u root -p123456**

**stop slave;**

**CHANGE MASTER TO**

**MASTER\_HOST='192.168.200.112',**

**MASTER\_USER='repl',**

**MASTER\_PASSWORD='123456';**

**start slave;**

**show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.112**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000001**

**Read\_Master\_Log\_Pos: 1372**

**Relay\_Log\_File: mariadb-relay-bin.000002**

**Relay\_Log\_Pos: 1208**

**Relay\_Master\_Log\_File: master-bin.000001**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

Server05(192.168.200.115) 修改监控配置文件添加server1配置：

**[root@server05 ~]# vim /etc/masterha/app1.cnf**

**[server1]**

**hostname=server01**

**port=3306**

Server05(192.168.200.115) 检查集群状态：

**[root@server05 ~]# masterha\_check\_repl --conf=/etc/masterha/app1.cnf**

**-----------------------------------忽略部分信息-----------------------------------**

**Thu Aug 31 22:20:30 2017 - [info] Alive Servers:**

**Thu Aug 31 22:20:30 2017 - [info] server01(192.168.200.111:3306)**

**Thu Aug 31 22:20:30 2017 - [info] server02(192.168.200.112:3306)**

**Thu Aug 31 22:20:30 2017 - [info] server03(192.168.200.113:3306)**

**Thu Aug 31 22:20:30 2017 - [info] server04(192.168.200.114:3306)**

**-----------------------------------忽略部分信息-----------------------------------**

**server02 (current master)**

**+--server01**

**+--server03**

**+--server04**

**-----------------------------------忽略部分信息-----------------------------------**

**MySQL Replication Health is OK.**

Server05(192.168.200.115) 开启监控

**[root@server05 ~]# nohup masterha\_manager --conf=/etc/masterha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover< /dev/null >/var/log/masterha/app1/manager.log 2>&1 &**

Secondary Master(192.168.200.112) 关闭现有主库mysql

**[root@server02 ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.112/24 brd 192.168.200.255 scope global ens33**

**inet 192.168.200.100/24 brd 192.168.200.255 scope global secondary ens33:1**

**[root@server02 ~]# systemctl stop mariadb**

**[root@server02 ~]# netstat -lnpt | grep :3306**

Primary Master(192.168.200.111)

**[root@server01 ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.111/24 brd 192.168.200.255 scope global ens33**

**inet 192.168.200.100/24 brd 192.168.200.255 scope global secondary ens33:1**

slave1(192.168.200.113)状态：

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.111**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000002**

**Read\_Master\_Log\_Pos: 923**

**Relay\_Log\_File: slave-relay-bin.000002**

**Relay\_Log\_Pos: 530**

**Relay\_Master\_Log\_File: master-bin.000002**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

slave2(192.168.200.114)状态：

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.111**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000002**

**Read\_Master\_Log\_Pos: 923**

**Relay\_Log\_File: slave-relay-bin.000002**

**Relay\_Log\_Pos: 530**

**Relay\_Master\_Log\_File: master-bin.000002**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

Server05(192.168.200.115) 配置文件变化（已经移除故障机server2配置）：

**[root@server05 ~]# cat /etc/masterha/app1.cnf**

**[server default]**

**manager\_log=/var/log/masterha/app1/manager.log**

**manager\_workdir=/var/log/masterha/app1**

**master\_binlog\_dir=/var/lib/mysql**

**master\_ip\_failover\_script=/usr/local/bin/master\_ip\_failover**

**password=123456**

**ping\_interval=1**

**remote\_workdir=/tmp**

**repl\_password=123456**

**repl\_user=repl**

**user=root**

**[server1]**

**hostname=server01**

**port=3306**

**[server3]**

**hostname=server03**

**port=3306**

**[server4]**

**hostname=server04**

**port=3306**

Server05(192.168.200.115) 监控日志：

**Selected server01 as a new master.**

**server01: OK: Applying all logs succeeded.**

**server01: OK: Activated master IP address.**

**server03: This host has the latest relay log events.**

**server04: This host has the latest relay log events.**

**Generating relay diff files from the latest slave succeeded.**

**server04: OK: Applying all logs succeeded. Slave started, replicating from server01.**

**server03: OK: Applying all logs succeeded. Slave started, replicating from server01.**

**server01: Resetting slave info succeeded.**

**Master failover to server01(192.168.200.111:3306) completed successfully.**

修复Secondary Master(192.168.200.112)主机：

**[root@server02 ~]# systemctl start mariadb**

**[root@server02 ~]# netstat -lnpt | grep :3306**

**tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 5982/mysqld**

Secondary Master(192.168.200.112) 指向新的主库

**[root@server02 ~]# mysql -u root -p123456**

**stop slave; CHANGE MASTER TO MASTER\_HOST='192.168.200.111', MASTER\_USER='repl', MASTER\_PASSWORD='123456';**

**start slave;**

**show slave status\G**

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**Master\_Host: 192.168.200.111**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: master-bin.000002**

**Read\_Master\_Log\_Pos: 923**

**Relay\_Log\_File: mariadb-relay-bin.000003**

**Relay\_Log\_Pos: 530**

**Relay\_Master\_Log\_File: master-bin.000002**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

Server05(192.168.200.115)修改监控配置文件添加server2配置：

**[root@server05 ~]# vim /etc/masterha/app1.cnf**

**[server2]**

**hostname=server02**

**candidate\_master=1**

**port=3306**

**check\_repl\_delay=0**

Server05(192.168.200.115)检查集群状态：

**[root@server05 ~]# masterha\_check\_repl --conf=/etc/masterha/app1.cnf**

**server01 (current master)**

**+--server02**

**+--server03**

**+--server04**

**-----------------------------------忽略部分信息-----------------------------------**

**MySQL Replication Health is OK.**

如果server01，server02的mariadb都stop了，master将往server03转移，只要还有一台没有stop，就会有master，自行测试验证，注意必须开启监控才能转移

[root@server04 mha4mysql-node-0.56]# mysql -p123456 -e 'show slave status\G'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

Master\_Host: 192.168.200.113

Master\_User: repl

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000001

……略

[root@server05 mha]# masterha\_check\_status --conf=/etc/masterha/app1.cnf

app1 (pid:63698) is running(0:PING\_OK), master:server03