一.环境

master1	192.168.18.10	主（写入）
master3	192.168.18.20	从（读）
slave3	192.168.18.30	从（读）
manager	192.168.18.100	管理节点

二.操作

1.安装epel源

```
wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo
yum makecache
```

2.安装依赖

```
yum -y install perl-DBD-MySQL perl-Config-Tiny perl-Log-Dispatch perl-Parallel-ForkManager perl-Config-IniFiles ncftp perl-Params-Validate perl-CPAN perl-Test-Mock-LWP.noarch perl-LWP-Authen-Negotiate.noarch perl-devel perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker
```

3.更改防火墙规则

```
setenforce 0
sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config
systemctl disable firewalld
```

4.更改主机名

```
vim /etc/hosts
192.168.18.10 master1
192.168.18.20 master3
192.168.18.30 slave3
192.168.18.100 manager
```

5.删除UUID

```
rm -rf /usr/local/mysql/data/auto.cnf
```

6.在所有主机上安装mha4mysql-node-0.56.tar.gz

```
tar zxf mha4mysql-node-0.58.tar.gz
cd mha4mysql-node-0.58
perl Makefile.PL
make && make install
```

7.在管理节点上安装mha4mysql-manager-0.56.tar.gz

```
tar zxf mha4mysql-manager-0.58.tar.gz
cd mha4mysql-manager-0.58/
perl Makefile.PL
make && make install
```

8.设置公钥并分发

```
[root@slave3 ~]# ssh-keygen -t rsa
[root@slave3 ~]# for i in master1 master3 slave3 manager;do ssh-copy-id -i $i;done
```

9.配置半同步复制

1）查看plugin目录

```
mysql> show variables like '%plugin_dir%';
```

2）查看是否支持安装插件

```
mysql> show variables like '%have_dynamic%';
```

3）安装半同步插件

```
mysql> install plugin rpl_semi_sync_master SONAME 'semisync_master.so';
mysql> install plugin rpl_semi_sync_slave SONAME 'semisync_slave.so';
```

4）查看是否安装成功

```
mysql> show plugins;
```

5）查看半同步相关信息

```
mysql> show variables like '%rpl_semi_sync%';
```

10.修改配置文件

master1

```
vim /etc/my.cnf
server-id = 1
log-bin=mysql-bin
binlog_format=mixed
log-bin-index=mysql-bin.index
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000
rpl_semi_sync_slave_enabled=1
relay_log_purge=0
 relay-log = relay-bin
 relay-log-index = slave-relay-bin.index
```

rpl_semi_sync_master_enabled=1 1表是启用，0表示关闭

rpl_semi_sync_master_timeout=10000：毫秒单位 ，该参数主服务器等待确认消息10秒后，不再等待，变为异步方式。

relay_log_purge=0，禁止 SQL 线程在执行完一个 relay log 后自动将其删除

master3

```
vim /etc/my.cnf
server-id=2
```

```
log-bin=mysql-bin
binlog_format=mixed
relay_log_purge=0
relay-log=relay-bin
log-bin-index=mysql-bin.index
relay-log-index=slave-relay-bin.index
rpl_semi_sync_master_enabled=1
 rpl_semi_sync_master_timeout=10000
 rpl_semi_sync_slave_enabled=1

```

slave3

```
vim /etc/my.cnf
server-id=3
log-bin=mysql-bin
relay-log=relay-bin
relay-log-index=slave-relay-bin.index
rpl_semi_sync_slave_enabled=1
read_only = 1
```

11.重启后查看半同步相关信息和状态

```
mysql>show variables like '%rpl_semi_sync%';
mysql>show status like '%rpl_semi_sync%';
```

rpl_semi_sync_master_status ： 显示主服务是异步复制模式还是半同步复制模式

rpl_semi_sync_master_clients ： 显示有多少个从服务器配置为半同步复制模式

rpl_semi_sync_master_yes_tx ： 显示从服务器确认成功提交的数量

rpl_semi_sync_master_no_tx ： 显示从服务器确认不成功提交的数量

rpl_semi_sync_master_tx_avg_wait_time ： 事务因开启 semi_sync ， 平均需要额外等待的时间

rpl_semi_sync_master_net_avg_wait_time ： 事务进入等待队列后，到网络平均等待时间

12.在两台master上创建用于主从复制的帐号

```
mysql> grant replication slave on *.* to mharep@'192.168.18.%' identified by '123';
```

13.所有mysql服务器上创建mha管理账户

```
mysql> grant all privileges on *.* to manager@'192.168.18.%' identified by '123';
```

14.配置主从复制

```
mysql> show master status;
change master to
```

```
master_host='192.168.43.10',
master_user='mharep',
master_password='123',
master_log_file='mysql-bin.000001',
master_log_pos=154;
mysql> start slave;
mysql> show slave status\G
```

15.再次在master1上查看半同步状态

```
mysql>show status like '%rpl_semi_sync%';
```

16.在manager上创建目录

```
mkdir /etc/masterha
mkdir -p /masterha/app1
mkdir /scripts
cp samples/conf/* /etc/masterha/
[root@manager masterha]# >masterha_default.cnf 清空数据
cp samples/scripts/* /scripts/
```

17.修改配置文件

```
vim /etc/masterha/app1.cnf
[server default]
manager_workdir=/masterha/app1
manager_log=/masterha/app1/manager.log
user=manager
password=123
ssh_user=root
repl_user=mharep
repl_password=123
 ping_interval=1
 [server1]
 hostname=192.168.18.10
 port=3306
 master_binlog_dir=/usr/local/mysql/data
 candidate_master=1
 [server2]
 hostname=192.168.18.20
 port=3306
 master_binlog_dir=/usr/local/mysql/data
 candidate_master=1
 [server3]
 hostname=192.168.18.30
```

```
port=3306
master_binlog_dir=/usr/local/mysql/data
no_master=1
```

18.使用ssh和repl测试是否成功

```
masterha_check_ssh --global_conf=/etc/masterha/masterha_default.cnf --conf=/etc/masterha/app1.cnf
masterha_check_repl --global_conf=/etc/masterha/masterha_default.cnf --conf=/etc/masterha/app1.cnf
--global_conf=/etc/masterha/masterha_defa
```

19.启动manager

```
nohup masterha_manager --conf=/etc/masterha/app1.cnf &>/tmp/mha_manager.log &
```

20.查看是否启动成功

```
masterha_check_status --conf=/etc/masterha/app1.cnf
```

21.测试一台宕机，是否自动切换

宕机后mha会自动关闭

22.将宕机的在添加回阵列中

1）指认副主为主

```
[root@manager ~]# grep "CHANGE" /masterha/app1/manager.log
mysql> CHANGE MASTER TO MASTER_HOST='192.168.18.20', MASTER_PORT=3306, MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=744, MASTER_USER='mharep', MASTER_PASSWORD='123';
mysql> start slave;
mysql> show slave status\G
```

2）删除宕机所产生的错误文件

```
[root@manager ~]# cd /masterha/app1/
[root@manager app1]# rm -f app1.failover.complete
```

3）启动

```
[root@manager ~]# nohup masterha_manager --conf=/etc/masterha/app1.cnf &>/tmp/mha_manager.log &
[root@manager ~]# masterha_check_status --conf=/etc/masterha/app1.cnf
```

三.配置vip

1）通过keepalived的方式管理虚拟ip（在主和备主上设置）

```
yum -y install kernel-devel openssl-devel popt-devel
[root@localhost ~]# wget https://www.keepalived.org/software/keepalived-2.1.2.tar.gz
root@localhost keepalived-2.1.2]# yum -y install gcc gcc-c++
[root@localhost keepalived-2.1.2]# ./configure --prefix=/ && make && make install
[root@master1 keepalived]# systemctl enable keepalived.service
```

执行make install操作之后，会自动生成/etc/init.d/keepalived脚本文件，但还需要手动添加为系统服务，这样就可以使用service、chkconfig工具来对keepalived服务程序进行管理了。

2）创建防火墙规则

```
[root@localhost ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --in-interface ens33 --destination 224.0.0.18 --protocol vrrp -j ACCEPT
[root@localhost ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ens33 --destination 224.0.0.18 --protocol vrrp -j ACCEPT
firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 --in-interface ens33 --destination 224.0.0.18 --protocol vrrp -j ACCEPT
```

--in-interface 接口

--destination 组播地址

--protocol 协议

3）修改配置文件

global_defs：主要是配置故障发生时的通知对象以及机器标识。

vrrp_instance：用来定义对外提供服务的VIP区域及其相关属性。

virtual_server：虚拟服务器定义

```
[root@localhost ~]# vim /etc/keepalived/keepalived.conf 主1
! Configuration File for keepalived
global_defs {
 router_id master-1
}
vrrp_instance VI_1 {
 state BACKUP
 interface ens33
 virtual_router_id 51
  priority 100
  nopreempt
  advert_int 1
  authentication {
  auth_type PASS
  auth_pass 1111
  }
  virtual_ipaddress {
  192.168.18.200
  }
 }
 service keepalived start
```

```
ip a
```

同步复制到另一台

4.把keepalived服务引入MHA

```
[root@manager ~]# vim /scripts/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 = '192.168.18.200';
 my $ssh_start_vip = "systemctl start keepalived.service";
 my $ssh_stop_vip = "systemctl stop keepalived.service";
 GetOptions(
 'command=s' => \$command,
 'ssh_user=s' => \$ssh_user,
 'orig_master_host=s' => \$orig_master_host,
 'orig_master_ip=s' => \$orig_master_ip,
 'orig_master_port=i' => \$orig_master_port,
 'new_master_host=s' => \$new_master_host,
 'new_master_ip=s' => \$new_master_ip,
 'new_master_port=i' => \$new_master_port,
 );
 exit &main();
 sub main {
 print "\n\nIN SCRIPT TEST====$ssh_stop_vip==$ssh_start_vip===\n\n";
 if ( $command eq "stop" || $command eq "stopssh" ) {
 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" ) {
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 master
sub start_vip() {
 `ssh $ssh_user\@$new_master_host \" $ssh_start_vip \"`;
}
# A simple system call that disable the VIP on the old_master
sub stop_vip() {
return 0 unless ($ssh_user);
`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@manager scripts]# vim /etc/masterha/app1.cnf
[server default]
manager_workdir=/masterha/app1
manager_log=/masterha/app1/manager.log
master_ip_failover_script=/scripts/master_ip_failover
[root@manager ~]# masterha_stop --conf=/etc/masterha/app1.cnf
[root@manager ~]# nohup masterha_manager --conf=/etc/masterha/app1.cnf &
>/tmp/mha_manager.log &
[root@manager ~]# masterha_check_status --conf=/etc/masterha/app1.cnf
[root@manager ~]# masterha_check_repl --conf=/etc/masterha/app1.cnf
```

宕机master1查看状态

```
[root@master1 ~]# systemctl stop mysqld
```

在启动添加进去

```
[root@master1 ~]# systemctl start mysqld
[root@master1 ~]# mysql -uroot -p123
[root@manager ~]# grep "CHANGE MASTER TO MASTER" /masterha/app1/manager.l
og
mysql> CHANGE MASTER TO MASTER_HOST='192.168.18.20', MASTER_PORT=3306, MA
STER_LOG_FILE='mysql-bin.000003', MASTER_LOG_POS=154, MASTER_USER='mharep',
MASTER_PASSWORD='123';
mysql> start slave;
mysql> show slave status\G;
[root@master1 ~]# systemctl start keepalived
[root@master1 ~]# systemctl status keepalived
[root@manager ~]# rm -rf /masterha/app1/app1.failover.complete
[root@manager ~]# nohup masterha_manager --conf=/etc/masterha/app1.cnf &
>/tmp/mha_manager.log &
[root@manager ~]# masterha_check_status --conf=/etc/masterha/app1.cnf
```

2）通过脚本实现VIP切换

手动在master服务器上绑定一个vip

```
[root@master3 ~]# ifconfig ens33:0 192.168.18.150/24
```

```
[root@manager ~]# vim /scripts/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 = '192.168.18.150';
my $key = '0';
my $ssh_start_vip = "/sbin/ifconfig ens33:$key $vip";
my $ssh_stop_vip = "/sbin/ifconfig ens33:$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" ) {
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" ) {
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 master
sub start_vip() {
`ssh $ssh_user\@$new_master_host \" $ssh_start_vip \"`;
}
# A simple system call that disable the VIP on the old_master
sub stop_vip() {
return 0 unless ($ssh_user);
`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@manager scripts]# vim /etc/masterha/app1.cnf
[server default]
manager_workdir=/masterha/app1
manager_log=/masterha/app1/manager.log
master_ip_failover_script=/scripts/master_ip_failover
[root@manager ~]# masterha_stop --conf=/etc/masterha/app1.cnf
[root@manager ~]# nohup masterha_manager --conf=/etc/masterha/app1.cnf &
>/tmp/mha_manager.log &
[root@manager ~]# masterha_check_status --conf=/etc/masterha/app1.cnf
[root@manager ~]# masterha_check_repl --conf=/etc/masterha/app1.cnf
```

宕机master1查看状态

```
[root@master1 ~]# systemctl stop mysqld
```