一、MHA概述

1. 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，它由日本DeNA公司youshimaton（现就职于Facebook公司）开发，是一套优秀的作为MySQL高可用性环境下故障切换和主从提升的高可用软件。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故障切换操作，并且在进行故障切换的过程中，MHA能在较大程度上保证数据的一致性，以达到真正意义上的高可用。

2. MHA组件

2.1 MHA相关组件

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。

2.2 MHA组件介绍

- MHA Manager

运行一些工具，比如masterha_manager工具实现自动监控MySQL Master和实现master故障切换，其它工具手动实现master故障切换、在线mater转移、连接检查等等。一个Manager可以管理多 个master-slave集群

- MHA Node

部署在所有运行MySQL的服务器上，无论是master还是slave。主要有三个作用：

1）保存二进制日志

如果能够访问故障master，会拷贝master的二进制日志

2）应用差异中继日志

从拥有最新数据的slave上生成差异中继日志，然后应用差异日志。

3）清除中继日志

在不停止SQL线程的情况下删除中继日志

2.3 相关工具

1.Manager工具:

- masterha_check_ssh : 检查MHA的SSH配置。
- masterha_check_repl : 检查MySQL复制。
- masterha_manager : 启动MHA。
- masterha_check_status : 检测当前MHA运行状态。
- masterha_master_monitor : 监测master是否宕机。
- masterha_master_switch : 控制故障转移(自动或手动)。

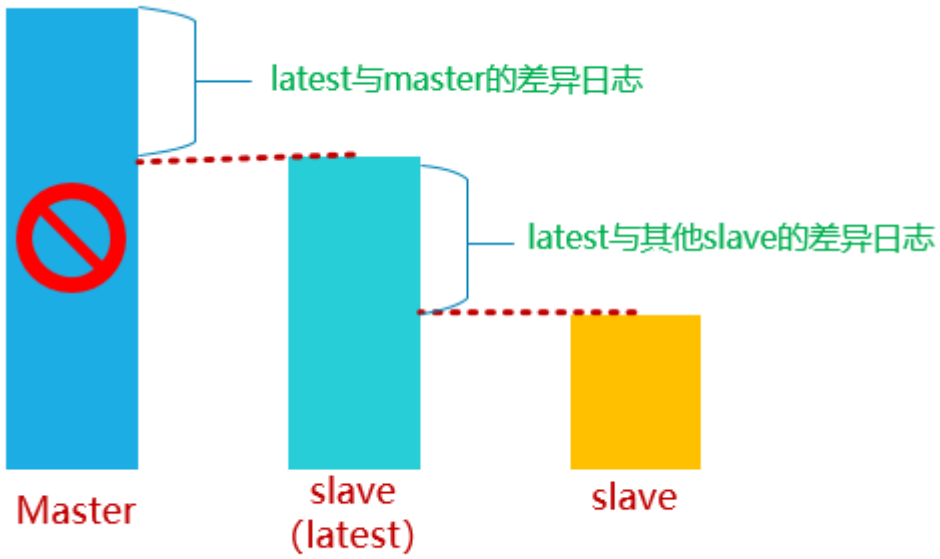
- masterha_conf_host : 添加或删除配置的server信息。

2. Node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应用于其它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：Node这些工具通常由MHA Manager的脚本触发,无需人手操作。

二、MHA工作原理



1. 当master出现故障时，通过对比slave之间I/O线程读取master上binlog的位置，选取最接近的slave做为最新的slave（latest slave）。
2. 其它slave通过与latest slave对比生成差异中继日志，并应用。
3. 在latest slave上应用从master保存的binlog，同时将latest slave提升为master。
4. 最后在其它slave上应用相应的差异中继日志并开始从新的master开始复制.

三、MHA部署

1. 部署规划

角色	IP	主机名	server-id	功能	备注
MHA-Manager	10.1.1.8	mha-mgr.misshou.cc	—	管理节点	
MHA-Node（Master）	10.1.1.5	master.misshou.cc	10	数据节点	写
MHA-Node（Slave1）	10.1.1.6	slave1.misshou.cc	20	数据节点	读
MHA-Node（Slave2）	10.1.1.7	slave2.misshou.cc	30	数据节点	读

2. 环境准备

系统版本	MySQL版本	MHA版本
CentOS Linux release 7.5.1804	MySQL-5.6.35	mha4mysql-manager-0.57
		mha4mysql-node-0.57

3. 系统环境初始化

3.1 修改主机名和hosts

```
# hostnamectl set-hostname master.misshou.cc
# hostnamectl set-hostname slave1.misshou.cc
# hostnamectl set-hostname slave2.misshou.cc
# hostnamectl set-hostname mha-mgr.misshou.cc
# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.1.5    master.misshou.cc master
10.1.1.6    slave1.misshou.cc slave1
10.1.1.7    slave2.misshou.cc slave2
10.1.1.8    mha-mgr.misshou.cc   mgr
```

3.2 关闭防火墙和selinux

```
# systemctl stop firewalld
# systemctl disable firewalld
# setenforce 0
# sed -i '/SELINUX=enforcing/cSELINUX=disabled' /etc/selinux/config
```

3.3 关闭NetworkManager服务

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
```

3.4 配置yum源

说明：分别配置 aliyun、epel 和本地源

```
# rpm -ivh /soft/mha/epel-release-latest-7.noarch.rpm
[root@db01 yum.repos.d]# cat server.repo
[local]
name=local yum
baseurl=file:///mnt
enabled=1
gpgcheck=0
[aliyun]
name=this is aliyun yum
baseurl=http://mirrors.aliyun.com/centos/7/os/x86_64/
enabled=1
gpgcheck=0
```

3.5 安装依赖包

注意：所有服务器均需要安装

```
yum -y install perl-DBD-MySQL \
perl-Config-Tiny \
perl-Time-HiRes \
perl-Mail-Sender \
perl-Mail-Sendmail \
perl-MIME-Base32 \
perl-MIME-Charset \
perl-MIME-EncWords \
perl-Params-Classify \
perl-Params-Validate.x86_64 \
perl-Log-Dispatch \
perl-Parallel-ForkManager

yum -y install net-tools
```

4. 部署MySQL复制环境

4.1 MySQL部署规划

安装目录	数据库目录	配置文件	套接字文件	端口
/usr/local/mysql	/usr/local/mysql/data	/etc/my.cnf	/usr/local/mysql/mysql.sock	3307

4.2 搭建主从复制

```
1. 创建配置文件
[root@master ~]# cat /etc/my.cnf
[mysqld]
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
port=3307
socket=/usr/local/mysql/mysql.sock
gtid-mode=on
log-bin
log-slave-updates
```

```
enforce-gtid-consistency
server-id=10
[client]
socket=/usr/local/mysql/mysql.sock


[root@slave1 ~]# cat /etc/my.cnf
[mysqld]
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
port=3307
socket=/usr/local/mysql/mysql.sock
gtid-mode=on
log-bin
relay-log=/usr/local/mysql/relay-bin.log
log-slave-updates
enforce-gtid-consistency
server-id=20
[client]
socket=/usr/local/mysql/mysql.sock


[root@slave2 ~]# cat /etc/my.cnf
[mysqld]
basedir=/usr/local/mysql
datadir=/usr/local/mysql/data
port=3307
socket=/usr/local/mysql/mysql.sock
gtid-mode=on
log-bin
relay-log=/usr/local/mysql/relay-bin.log
log-slave-updates
enforce-gtid-consistency
server-id=30
[client]
socket=/usr/local/mysql/mysql.sock

2. 将master原有数据分别同步到slave1和slave2上
1)slave1、slave2、mha-mgr上分别创建mysql用户
# useradd  mysql
2)master同步数据到所有slave上
[root@master ~]# rsync -av /usr/local/mysql 10.1.1.6:/usr/local/
[root@master ~]# rsync -av /usr/local/mysql 10.1.1.7:/usr/local/

3) 删除三台服务器上的auto.cnf文件
[root@master data]# rm -f auto.cnf
[root@slave1 data]# rm -f auto.cnf
[root@slave2 data]# rm -f auto.cnf

3. 启动三台数据库服务器
[root@master data]# service mysql start
Starting MySQL.Logging to '/usr/local/mysql/data/master.misshou.cc.err'.
.. SUCCESS!
```

```
[root@slave1 data]# service mysql start
Starting MySQL.Logging to '/usr/local/mysql/data/slave1.misshou.cc.err'.
. SUCCESS!
[root@slave2 data]# service mysql start
Starting MySQL.Logging to '/usr/local/mysql/data/slave2.misshou.cc.err'.
.. SUCCESS!

4. 在master上分别创建复制和mha用户
mysql> grant replication slave, replication client on *.* to repl@'10.1.1.%' identified by
'123';
mysql> grant all privileges on *.* to mha@'10.1.1.8' identified by '123';
mysql> flush privileges;

5. 在slave上设置同步信息(slave1和slave2都要设置)
mysql> change master to
master_host='10.1.1.5',master_port=3307,master_user='repl',master_password='123',master_auto_pos
ition=1;

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

mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.1.1.5
                  Master_User: repl
                  Master_Port: 3307
                Connect_Retry: 60
              Master_Log_File: master-bin.000001
          Read_Master_Log_Pos: 812
               Relay_Log_File: relay-bin.000002
                Relay_Log_Pos: 1024
        Relay_Master_Log_File: master-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes

6. 测试主从同步
略
```

5. MHA软件安装

5.1 不同节点安装软件

说明：在所有节点安装 mha-node 软件包，在 mha 管理端再安装 mha-manager 软件包

```
[root@mha-mgr ~]# yum -y install mha4mysql-node-0.57-0.el7.noarch.rpm
[root@mha-mgr ~]# yum -y install mha4mysql-manager-0.57-0.el7.noarch.rpm
[root@master ~]# yum -y install mha4mysql-node-0.57-0.el7.noarch.rpm
[root@slave1 ~]# yum -y install mha4mysql-node-0.57-0.el7.noarch.rpm
[root@slave2 ~]# yum -y install mha4mysql-node-0.57-0.el7.noarch.rpm
```

5.2 配置ssh互信

说明：在生产环境中几乎都是禁止root远程登陆服务器的，所以ssh免密码登陆要在mysql用户下进行配置，这是处于安全角度考虑出发。

```
master端:
[mysql@master ~]$ ssh-keygen -P "" -f ~/.ssh/id_rsa
[mysql@master ~]$ rm -rf .ssh/*

slave1和slave2端:
[mysql@slave1 ~]$ ssh-keygen -P "" -f ~/.ssh/id_rsa
[mysql@slave1 ~]$ rm -rf .ssh/*

[mysql@slave2 ~]$ ssh-keygen -P "" -f ~/.ssh/id_rsa
[mysql@slave2 ~]$ rm -rf .ssh/*

mha-mgr端:
[root@mha-mgr ~]# su - mysql
[mysql@mha-mgr ~]$ ssh-keygen -P "" -f ~/.ssh/id_rsa
[mysql@mha-mgr ~]$ cd .ssh/
[mysql@mha-mgr .ssh]$ ls
id_rsa  id_rsa.pub
[mysql@mha-mgr .ssh]$ mv id_rsa.pub authorized_keys
[mysql@mha-mgr .ssh]$ for i in 5 6 7;do scp * 10.1.1.$i:~/.ssh/;done

测试免密登录:
[mysql@mha-mgr .ssh]$ ssh 10.1.1.5
[mysql@mha-mgr .ssh]$ ssh 10.1.1.6
[mysql@mha-mgr .ssh]$ ssh 10.1.1.7
```

5.3 配置mysql用户的sudo权限

- 配置mysql用户执行sudo命令权限

```
[root@master ~]# vim /etc/sudoers.d/mysql
#User_Alias  表示具有sudo权限的用 户 列表; Host_Alias表示主机的列表
User_Alias MYSQL_USERS = mysql
#Runas_Alias  表示用户以什么身份登录
Runas_Alias MYSQL_RUNAS = root
#Cmnd_Alias  表示允许执行命令的列表
Cmnd_Alias MYSQL_CMNDS = /sbin/ifconfig,/sbin/arping
MYSQL_USERS ALL = (MYSQL_RUNAS) NOPASSWD: MYSQL_CMNDS

[root@master ~]# for i in 6 7;do scp /etc/sudoers.d/mysql 10.1.1.$i:/etc/sudoers.d/
```

- 测试mysql用户是否可以挂载VIP

```
[mysql@master ~]$ sudo /sbin/ifconfig ens33:1 10.1.1.100 broadcast 10.1.1.255 netmask
255.255.255.0
[mysql@master ~]$ sudo /sbin/arping -f -q -c 5 -w 5 -I ens33 -s 10.1.1.100 -U 10.1.1.5
[mysql@master ~]$ ifconfig
```

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.5  netmask 255.255.255.0  broadcast 10.1.1.255
        inet6 fe80::20c:29ff:fe4c:a304  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:4c:a3:04  txqueuelen 1000  (Ethernet)
        RX packets 87558  bytes 7043682 (6.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 981178  bytes 2419766959 (2.2 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.100  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 00:0c:29:4c:a3:04  txqueuelen 1000  (Ethernet)
```

补充:
arping:用来向局域网内的其它主机发送ARP请求的指令，可以用来测试局域网内的某个IP是否已被使用。
Usage: arping [-fqbDUAV] [-c count] [-w timeout] [-I device] [-s source] destination
-f:收到第一个响应包后退出。
-q:quite模式，不显示输出。
-c:发送指定的count个ARP REQUEST包后停止。如果指定了-w参数，则会等待相同数量的ARP REPLY包，直到超时为止。
-w:指定一个超时时间，单位为秒，arping在到达指定时间后退出，无论期间发送或接收了多少包。在这种情况下，arping在发送完指定的count（-c）个包后并不会停止，而是等待到超时或发送的count个包都进行了回应后才会退出。
-I:指定设备名，用来发送ARP REQUEST包的网络设备的名称。
-D:重复地址探测模式，用来检测有没有IP地址冲突，如果没有IP冲突则返回0。
-s:设置发送ARP包的IP资源地址
-U:无理由的（强制的）ARP模式去更新别的主机上的ARP CACHE列表中的本机的信息，不需要响应。
-h:显示帮助页。

5.4 创建mha相关配置文件

- 创建 mha 相关的工作目录

```
[root@mha-mgr ~]# mkdir /etc/mha/
[root@mha-mgr ~]# mkdir -p /data/mha/masterha/app1
[root@mha-mgr ~]# chown -R mysql. /data/mha
```

- 创建mha局部配置文件

```
[root@mha-mgr ~]# cat /etc/mha/app1.conf
[server default]
# 设置监控用户和密码
user=mha
password=123
# 设置复制环境中的复制用户和密码
repl_user=repl
repl_password=123
# 设置ssh的登录用户名
ssh_user=mysql
# 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failover
ping_interval=3
# 设置mgr的工作目录
```

```
manager_workdir=/data/mha/masterha/app1
# 设置mysql master 保存 binlog 的目录,以便 MHA 可以找到 master 的二进制日志
master_binlog_dir=/usr/local/mysql/data
# 设置 master 的 pid 文件
master_pid_file=/usr/local/mysql/data/master.misshou.cc.pid
# 设置 mysql master 在发生切换时保存 binlog 的目录（在mysql master上创建这个目录）
remote_workdir=/data/mysql/mha
# 设置 mgr 日志文件
manager_log=/data/mha/masterha/app1/app1-3307.log
# MHA 到 master 的监控之间出现问题,MHA Manager 将会尝试从slave1和slave2登录到master上
secondary_check_script=/usr/bin/masterha_secondary_check -s 10.1.1.6 -s 10.1.1.7 --user=mysql --
port=22 --master_host=10.1.1.5 --master_port=3307
# 设置自动 failover 时候的切换脚本
master_ip_failover_script="/etc/mha/master_ip_failover.sh 10.1.1.100 1"
# 设置手动切换时候的切换脚本
#master_ip_online_change_script="/etc/mha/master_ip_online_change.sh 10.1.1.100 1"
# 设置故障发生后关闭故障主机脚本
# shutdown_script="/etc/mha/power_manager"
[server1]
hostname=10.1.1.5
port= 3307
candidate_master=1
[server2]
hostname=10.1.1.6
port= 3307
candidate_master=1
[server3]
hostname=10.1.1.7
port= 3307
no_master=1
```

5.5 上传相应脚本

```
[root@mha-mgr ~]# ls /etc/mha/
app1.conf  master_ip_failover.sh  master_ip_online_change.sh  power_manager
注意：脚本内容中要修改网卡名字
my $vip  = shift;
my $interface = 'ens33';
my $key = shift;

[root@mha-mgr ~]# chmod 755 /etc/mha/master_ip_*
[root@mha-mgr ~]# chmod 755 /etc/mha/power_manager
```

6. 检查ssh互信和集群状态

- 检查ssh互信

```
[mysql@mha-mgr ~]$ masterha_check_ssh --conf=/etc/mha/app1.conf
Tue Oct 16 00:05:26 2018 - [warning] Global configuration file /etc/masterha_default.cnf not
found. Skipping.
```

```
Tue Oct 16 00:05:26 2018 - [info] Reading application default configuration from
/etc/mha/app1.conf..
Tue Oct 16 00:05:26 2018 - [info] Reading server configuration from /etc/mha/app1.conf..
Tue Oct 16 00:05:26 2018 - [info] Starting SSH connection tests..
Tue Oct 16 00:05:27 2018 - [debug]
Tue Oct 16 00:05:26 2018 - [debug]  Connecting via SSH from mysql@10.1.1.5(10.1.1.5:22) to
mysql@10.1.1.6(10.1.1.6:22)..
Tue Oct 16 00:05:26 2018 - [debug]   ok.
Tue Oct 16 00:05:26 2018 - [debug]  Connecting via SSH from mysql@10.1.1.5(10.1.1.5:22) to
mysql@10.1.1.7(10.1.1.7:22)..
Warning: Permanently added '10.1.1.7' (ECDSA) to the list of known hosts.
Tue Oct 16 00:05:27 2018 - [debug]   ok.
Tue Oct 16 00:05:28 2018 - [debug]
Tue Oct 16 00:05:27 2018 - [debug]  Connecting via SSH from mysql@10.1.1.7(10.1.1.7:22) to
mysql@10.1.1.5(10.1.1.5:22)..
Tue Oct 16 00:05:27 2018 - [debug]   ok.
Tue Oct 16 00:05:27 2018 - [debug]  Connecting via SSH from mysql@10.1.1.7(10.1.1.7:22) to
mysql@10.1.1.6(10.1.1.6:22)..
Warning: Permanently added '10.1.1.6' (ECDSA) to the list of known hosts.
Tue Oct 16 00:05:28 2018 - [debug]   ok.
Tue Oct 16 00:05:28 2018 - [debug]
Tue Oct 16 00:05:26 2018 - [debug]  Connecting via SSH from mysql@10.1.1.6(10.1.1.6:22) to
mysql@10.1.1.5(10.1.1.5:22)..
Tue Oct 16 00:05:27 2018 - [debug]   ok.
Tue Oct 16 00:05:27 2018 - [debug]  Connecting via SSH from mysql@10.1.1.6(10.1.1.6:22) to
mysql@10.1.1.7(10.1.1.7:22)..
Warning: Permanently added '10.1.1.7' (ECDSA) to the list of known hosts.
Tue Oct 16 00:05:27 2018 - [debug]   ok.
Tue Oct 16 00:05:28 2018 - [info] All SSH connection tests passed successfully.  //说明ok
```

- 检查集群状态

```
[mysql@mha-mgr ~]$ masterha_check_repl --conf=/etc/mha/app1.conf
Tue Oct 16 00:07:00 2018 - [warning] Global configuration file /etc/masterha_default.cnf not
found. Skipping.
Tue Oct 16 00:07:00 2018 - [info] Reading application default configuration from
/etc/mha/app1.conf..
Tue Oct 16 00:07:00 2018 - [info] Reading server configuration from /etc/mha/app1.conf..
Tue Oct 16 00:07:00 2018 - [info] MHA::MasterMonitor version 0.57.
Tue Oct 16 00:07:03 2018 - [info] GTID failover mode = 1
Tue Oct 16 00:07:03 2018 - [info] Dead Servers:
Tue Oct 16 00:07:03 2018 - [info] Alive Servers:
Tue Oct 16 00:07:03 2018 - [info]   10.1.1.5(10.1.1.5:3307)
Tue Oct 16 00:07:03 2018 - [info]   10.1.1.6(10.1.1.6:3307)
Tue Oct 16 00:07:03 2018 - [info]   10.1.1.7(10.1.1.7:3307)
Tue Oct 16 00:07:03 2018 - [info] Alive Slaves:
Tue Oct 16 00:07:03 2018 - [info]   10.1.1.6(10.1.1.6:3307)  Version=5.6.35-log (oldest major
version between slaves) log-bin:enabled
Tue Oct 16 00:07:03 2018 - [info]     GTID ON
Tue Oct 16 00:07:03 2018 - [info]     Replicating from 10.1.1.5(10.1.1.5:3307)
Tue Oct 16 00:07:03 2018 - [info]     Primary candidate for the new Master (candidate_master is
set)
```

```
Tue Oct 16 00:07:03 2018 - [info]   10.1.1.7(10.1.1.7:3307)  Version=5.6.35-log (oldest major
version between slaves) log-bin:enabled
Tue Oct 16 00:07:03 2018 - [info]     GTID ON
Tue Oct 16 00:07:03 2018 - [info]     Replicating from 10.1.1.5(10.1.1.5:3307)
Tue Oct 16 00:07:03 2018 - [info]     Not candidate for the new Master (no_master is set)
Tue Oct 16 00:07:03 2018 - [info] Current Alive Master: 10.1.1.5(10.1.1.5:3307)
Tue Oct 16 00:07:03 2018 - [info] Checking slave configurations..
Tue Oct 16 00:07:03 2018 - [info]  read_only=1 is not set on slave 10.1.1.6(10.1.1.6:3307).
Tue Oct 16 00:07:03 2018 - [info]  read_only=1 is not set on slave 10.1.1.7(10.1.1.7:3307).
Tue Oct 16 00:07:03 2018 - [info] Checking replication filtering settings..
Tue Oct 16 00:07:03 2018 - [info]  binlog_do_db= , binlog_ignore_db=
Tue Oct 16 00:07:03 2018 - [info]  Replication filtering check ok.
Tue Oct 16 00:07:03 2018 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node
package checking.
Tue Oct 16 00:07:03 2018 - [info] Checking SSH publickey authentication settings on the current
master..
Tue Oct 16 00:07:03 2018 - [info] HealthCheck: SSH to 10.1.1.5 is reachable.
Tue Oct 16 00:07:03 2018 - [info]
10.1.1.5(10.1.1.5:3307) (current master)
 +--10.1.1.6(10.1.1.6:3307)
 +--10.1.1.7(10.1.1.7:3307)

Tue Oct 16 00:07:03 2018 - [info] Checking replication health on 10.1.1.6..
Tue Oct 16 00:07:03 2018 - [info]  ok.
Tue Oct 16 00:07:03 2018 - [info] Checking replication health on 10.1.1.7..
Tue Oct 16 00:07:03 2018 - [info]  ok.
Tue Oct 16 00:07:03 2018 - [info] Checking master_ip_failover_script status:
Tue Oct 16 00:07:03 2018 - [info]   /etc/mha/master_ip_failover.sh 10.1.1.100 1 --command=status
--ssh_user=mysql --orig_master_host=10.1.1.5 --orig_master_ip=10.1.1.5 --orig_master_port=3307
Checking the Status of the script.. OK
Tue Oct 16 00:07:04 2018 - [info]  OK.
Tue Oct 16 00:07:04 2018 - [warning] shutdown_script is not defined.
Tue Oct 16 00:07:04 2018 - [info] Got exit code 0 (Not master dead).

MySQL Replication Health is OK.     //说明ok
```

7. 检查MHA-Mgr状态

```
[mysql@mha-mgr ~]$ masterha_check_status --conf=/etc/mha/app1.conf
app1 is stopped(2:NOT_RUNNING).
开启MHA Manager监控:
[mysql@mha-mgr ~]$ nohup masterha_manager --conf=/etc/mha/app1.conf --remove_dead_master_conf --
ignore_last_failover &
再次查看监控状态:
[mysql@mha-mgr ~]$ masterha_check_status --conf=/etc/mha/app1.conf
app1 (pid:16338) is running(0:PING_OK), master:10.1.1.5

注意:
1. 如果正常，会显示"PING_OK "，否则会显示"NOT_RUNNING "，说明 MHA  监控没有开启
2. 使用mysql用户启动监控，否则会报权限拒绝
3. 手动停止监控命令:masterha_stop --conf=/etc/mha/app1.conf
```

四、自动Failover测试

1. 安装测试工具

```
[root@master ~]# yum -y install sysbench
```

2. 创建测试数据

```
mysql> create database autotest charset utf8;
Query OK, 1 row affected (0.17 sec)

mysql> grant all on *.* to 'mha'@'localhost' identified by '123';
Query OK, 0 rows affected (0.14 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.11 sec)

mysql> exit
Bye


[root@master ~]# sysbench /usr/share/sysbench/oltp_read_only.lua \
 --mysql-host=10.1.1.5  --mysql-port=330  --mysql-user=mha \
 --mysql-password=123  --mysql-socket=/usr/local/mysql/mysql.sock \
 --mysql-db=autotest  --db-driver=mysql  --tables=1 \
 --table-size=100000 --report-interval=10 --threads=128 --time=120 prepare
```

3. 模拟故障

```
[root@master ~]# service mysql stop
Shutting down MySQL.............. [ OK ]
```

4. 查看切换过程

```
[root@mha-mgr ~]# tail -f /data/mha/masterha/app1/app1-3307.log
....

----- Failover Report -----

app1: MySQL Master failover 10.1.1.5(10.1.1.5:3307) to 10.1.1.6(10.1.1.6:3307) succeeded

Master 10.1.1.5(10.1.1.5:3307) is down!

Check MHA Manager logs at mha-mgr.misshou.cc:/data/mha/masterha/app1/app1-3307.log for details.

Started automated(non-interactive) failover.
Invalidated master IP address on 10.1.1.5(10.1.1.5:3307)
Selected 10.1.1.6(10.1.1.6:3307) as a new master.
10.1.1.6(10.1.1.6:3307): OK: Applying all logs succeeded.
10.1.1.6(10.1.1.6:3307): OK: Activated master IP address.
```

```
10.1.1.7(10.1.1.7:3307): OK: Slave started, replicating from 10.1.1.6(10.1.1.6:3307)
10.1.1.6(10.1.1.6:3307): Resetting slave info succeeded.
Master failover to 10.1.1.6(10.1.1.6:3307) completed successfully.
```