# MHA高可用及Atlas读写分离方案

**课程大纲**

* **MHA作者及软件介绍**
* **MHA工作流程介绍**
* **MHA架构简单讲解**
* **MHA工具介绍**
* **MHA安装部署**
* **MHA的failover测试**
* **MHA的VIP漂移**
* **MHA的binlog-server讲解**
* **Atlas介绍、部署及安装**
* **Atlas配置文件详解**
* **Atlas管理接口讲解**

# 简介

### **1.1作者简介**

**松信嘉範**：

MySQL/Linux专家

2001年索尼公司入职

2001年开始使用oracle

2004年开始使用MySQL

2006年9月-2010年8月MySQL从事顾问

2010年-2012年 DeNA

2012年~至今 Facebook

### **1.2软件简介**

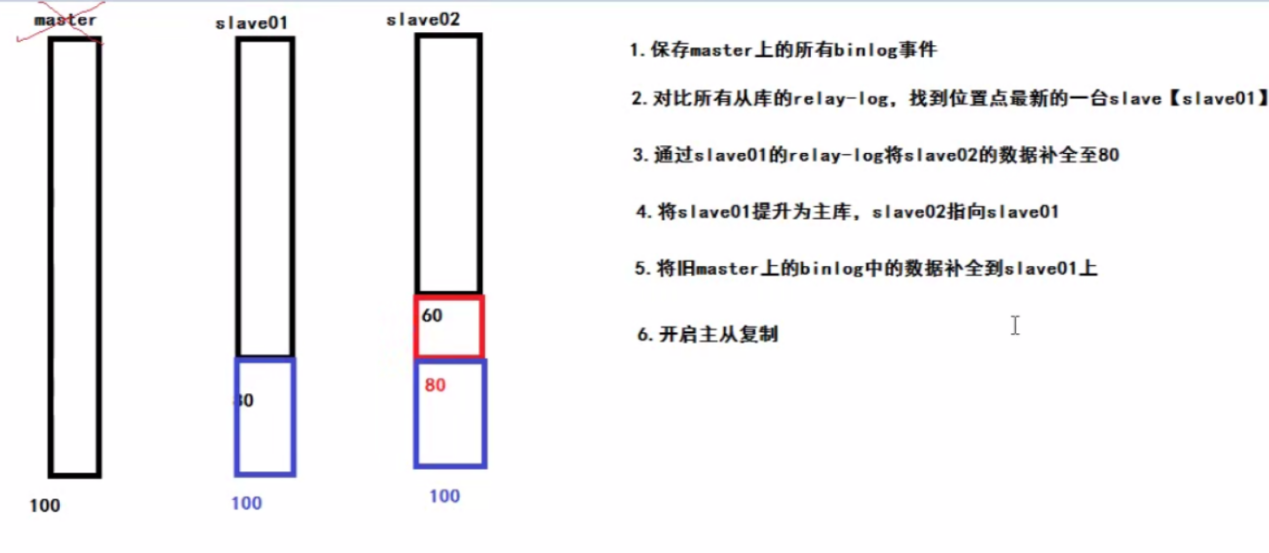
MHA能够在较短的时间内实现自动故障检测和故障转移，通常在10-30秒以内;在复制框架中，MHA能够很好地解决复制过程中的数据一致性问题，由于不需要在现有的replication中添加额外的服务器，仅需要一个manager节点，而一个Manager能管理多套复制，所以能大大地节约服务器的数量;另外，安装简单，无性能损耗，以及不需要修改现有的复制部署也是它的优势之处。

MHA还提供在线主库切换的功能，能够安全地切换当前运行的主库到一个新的主库中(通过将从库提升为主库),大概0.5-2秒内即可完成。

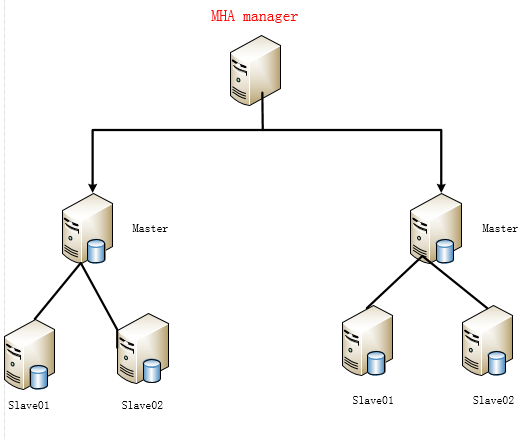
MHA由两部分组成：MHA Manager（管理节点）和MHA Node（数据节点）。MHA Manager可以独立部署在一台独立的机器上管理多个Master-Slave集群，也可以部署在一台Slave上。**当Master出现故障时，它可以自动将最新数据的Slave提升为新的Master,然后将所有其他的Slave重新指向新的Master**。整个故障转移过程对应用程序是完全透明的。

### **1.3工作流程**

* 把宕机的master二进制日志保存下来。
* 找到binlog位置点最新的slave。
* 在binlog位置点最新的slave上用relay log（差异日志）修复其它slave。
* 将宕机的master上保存下来的二进制日志恢复到含有最新位置点的slave上。
* 将含有最新位置点binlog所在的slave提升为master。
* 将其它slave重新指向新提升的master，并开启主从复制。



### **1.4MHA架构图**



### **1.5MHA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-key

masterha\_check\_repl #检查主从复制情况

masterha\_manger #启动MHA

masterha\_check\_status #检测MHA的运行状态

masterha\_master\_monitor #检测master是否宕机

masterha\_master\_switch #手动故障转移

masterha\_conf\_host #手动添加server信息

masterha\_secondary\_check #建立TCP连接从远程服务器

masterha\_stop #停止MHA

* Node工具包主要包括以下几个工具：

save\_binary\_logs #保存宕机的master的binlog

apply\_diff\_relay\_logs #识别relay log的差异

filter\_mysqlbinlog #防止回滚事件

purge\_relay\_logs #清除中继日志

### **1.6 MHA优点总结**

* Masterfailover and slave promotion can be done very quickly  
  自动故障转移快
* Mastercrash does not result in data inconsistency  
  主库崩溃不存在数据一致性问题
* Noneed to modify current MySQL settings (MHA works with regular MySQL)  
  不需要对当前mysql环境做重大修改
* Noneed to increase lots of servers  
  不需要添加额外的服务器(仅一台manager就可管理上百个replication)
* Noperformance penalty  
  性能优秀，可工作在半同步复制和异步复制，当监控mysql状态时，仅需要每隔N秒向master发送ping包(默认3秒)，所以对性能无影响。你可以理解为MHA的性能和简单的主从复制框架性能一样。
* Works with any storage engine  
  只要replication支持的存储引擎，MHA都支持，不会局限于innodb

# mysql环境准备

### **2.1环境检查**

* **mysql-db01**

#系统版本

[root@mysql-db01 ~]# cat /etc/redhat-release

CentOS release 6.7 (Final)

#内核版本

[root@mysql-db01 ~]# uname -r

2.6.32-573.el6.x86\_64

#IP地址

[root@mysql-db01 ~]# hostname -I

10.0.0.51

* **mysql-db02**

#系统版本

[root@mysql-db02 ~]# cat /etc/redhat-release

CentOS release 6.7 (Final)

#内核版本

[root@mysql-db02 ~]# uname -r

2.6.32-573.el6.x86\_64

#IP地址

[root@mysql-db02 ~]# hostname -I

10.0.0.52

* **mysql-db03**

#系统版本

[root@mysql-db03 ~]# cat /etc/redhat-release

CentOS release 6.7 (Final)

#内核版本

[root@mysql-db03 ~]# uname -r

2.6.32-573.el6.x86\_64

#IP地址

[root@mysql-db03 ~]# hostname -I

10.0.0.53

### **2.2安装mysql**

#### 2.2.1安装包准备

#创建安装包存放目录

[root@mysql-db01 ~]# mkdir /home/oldboy/tools -p

#进入目录

[root@mysql-db01 ~]# cd /home/oldboy/tools/

#上传mysql安装包（mysql-5.6.16-linux-glibc2.5-x86\_64.tar.gz）

[root@mysql-db01 tools]# rz -be

#### 2.2.2安装

#创建安装目录

[root@mysql-db01 tools]# mkdir /application

#解压mysql二进制包

[root@mysql-db01 tools]# tar xf mysql-5.6.16-linux-glibc2.5-x86\_64.tar.gz

#移动安装包

[root@mysql-db01 tools]# mv mysql-5.6.16-linux-glibc2.5-x86\_64 /application/mysql-5.6.16

#做软链接

[root@mysql-db01 tools]# ln -s /application/mysql-5.6.16/ /application/mysql

#创建mysql用户

[root@mysql-db01 tools]# useradd mysql -s /sbin/nologin -M

#进入mysql初始化目录

[root@mysql-db01 tools]# cd /application/mysql/scripts/

#初始化mysql

[root@mysql-db01 scripts]# ./mysql\_install\_db \

--user=mysql \

--datadir=/application/mysql/data/ \

--basedir=/application/mysql/

#注解

--user： 指定mysql用户

--datadir：指定mysql数据存放目录

--basedir：指定mysql base目录

#拷贝mysql配置文件

[root@mysql-db01 ~]# \cp /application/mysql/support-files/my-default.cnf /etc/my.cnf

#拷贝mysql启动脚本

[root@mysql-db01 ~]# cp /application/mysql/support-files/mysql.server /etc/init.d/mysqld

#修改mysql默认安装目录（否则无法启动）

[root@mysql-db01 ~]# sed -i 's#/usr/local#/application#g' /etc/init.d/mysqld

[root@mysql-db01 ~]# sed -i 's#/usr/local#/application#g' /application/mysql/bin/mysqld\_safe

#配置mysql环境变量

[root@mysql-db01 ~]# echo 'export PATH="/application/mysql/bin:$PATH"' >> /etc/profile.d/mysql.sh

#刷新环境变量

[root@mysql-db01 ~]# source /etc/profile

#### 2.2.3启动

#加入开机自启

[root@mysql-db01 ~]# chkconfig mysqld on

#启动mysql

[root@mysql-db01 ~]# /etc/init.d/mysqld start

Starting MySQL........... SUCCESS! #启动成功

#### 2.2.4配置密码

#配置mysql密码为oldboy123

[root@mysql-db01 ~]# mysqladmin -uroot password oldboy123

# 配置主从复制

### **3.1先决条件**

* 主库和从库都要开启binlog
* 主库和从库server-id不同
* 要有主从复制用户

### **3.2主库操作**

#### 3.2.1修改配置文件

#编辑mysql配置文件

[root@mysql-db01 ~]# vim /etc/my.cnf

#在mysqld标签下配置

[mysqld]

#主库server-id为1，从库不等于1

server\_id =1

#开启binlog日志

log\_bin=mysql-bin

#### 3.2.2创建主从复制用户

#登录数据库

[root@mysql-db01 ~]# mysql -uroot -poldboy123

#创建rep用户

mysql> grant replication slave on \*.\* to rep@'10.0.0.%' identified by 'oldboy123';

### **3.3从库操作**

#### 3.3.1修改配置文件

#修改mysql-db02配置文件

[root@mysql-db02 ~]# vim /etc/my.cnf

#在mysqld标签下配置

[mysqld]

#主库server-id为1，从库必须大于1

server\_id =5

#开启binlog日志

log\_bin=mysql-bin

#重启mysql

[root@mysql-db02 ~]# /etc/init.d/mysqld restart

#修改mysql-db03配置文件

[root@mysql-db03 ~]# vim /etc/my.cnf

#在mysqld标签下配置

[mysqld]

#主库server-id为1，从库必须大于1

server\_id =10

#开启binlog日志

log\_bin=mysql-bin

#重启mysql

[root@mysql-db03 ~]# /etc/init.d/mysqld restart

注：在以往如果是基于binlog日志的主从复制，则必须要记住主库的master状态信息。

mysql> show master status;

+------------------+----------+

| File | Position |

+------------------+----------+

| mysql-bin.000002 | 120 |

+------------------+----------+

### **3.4**[开启GTID](MYSQL开启GTID.txt)

#没开启之前先看一下GTID的状态

mysql> show global variables like '%gtid%';

+--------------------------+-------+

| Variable\_name | Value |

+--------------------------+-------+

| enforce\_gtid\_consistency | OFF |

| gtid\_executed | |

| gtid\_mode | OFF |

| gtid\_owned | |

| gtid\_purged | |

+--------------------------+-------+

#编辑mysql配置文件（主库从库都需要修改）

[root@mysql-db01 ~]# vim /etc/my.cnf

#在[mysqld]标签下添加

[mysqld]

gtid\_mode=ON

log\_slave\_updates

enforce\_gtid\_consistency

#重启数据库

[root@mysql-db01 ~]# /etc/init.d/mysqld restart

#检查GTID状态

mysql> show global variables like '%gtid%';

+--------------------------+-------+

| Variable\_name | Value |

+--------------------------+-------+

| enforce\_gtid\_consistency | ON | #执行GTID一致

| gtid\_executed | |

| gtid\_mode | ON | #开启GTID模块

| gtid\_owned | |

| gtid\_purged | |

+--------------------------+-------+

注：主库从库都需要开启GTID否则在做主从复制的时候就会报错：

[root@mysql-db02 ~]# mysql -uroot -poldboy123

mysql> change master to

-> master\_host='10.0.0.51',

-> master\_user='rep',

-> master\_password='oldboy123',

-> master\_auto\_position=1;

ERROR 1777 (HY000): CHANGE MASTER TO MASTER\_AUTO\_POSITION = 1 can only be executed when @@GLOBAL.GTID\_MODE = ON.

### **3.5配置主从复制**

#登录数据库

[root@mysql-db02 ~]# mysql -uroot -poldboy123

#配置复制主机信息

mysql> change master to

#主库IP

-> master\_host='10.0.0.51',

#主库复制用户

-> master\_user='rep',

#主库复制用户的密码

-> master\_password='oldboy123',

#GTID位置点

-> master\_auto\_position=1;

#开启slave

mysql> start slave;

#查看slave状态

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.51

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000003

Read\_Master\_Log\_Pos: 403

Relay\_Log\_File: mysql-db02-relay-bin.000002

Relay\_Log\_Pos: 613

Relay\_Master\_Log\_File: mysql-bin.000003

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: 403

Relay\_Log\_Space: 822

Until\_Condition: None

### **3.6从库设置**

#登录从库

[root@mysql-db02 ~]# mysql -uroot -poldboy123

#禁用自动删除relay log 功能

mysql> set global relay\_log\_purge = 0;

#设置只读

mysql> set global read\_only=1;

#编辑配置文件

[root@mysql-db02 ~]# vim /etc/my.cnf

#在mysqld标签下添加

[mysqld]

#禁用自动删除relay log 永久生效

relay\_log\_purge = 0

# 部署MHA

### **4.1环境准备（所有节点）**

#安装依赖包

[root@mysql-db01 ~]# yum install perl-DBD-MySQL -y

#进入安装包存放目录

[root@mysql-db01 ~]# cd /home/oldboy/tools/

#上传mha安装包

[root@mysql-db01 tools]# rz -be

mha4mysql-manager-0.56-0.el6.noarch.rpm

mha4mysql-manager-0.56.tar.gz

mha4mysql-node-0.56-0.el6.noarch.rpm

mha4mysql-node-0.56.tar.gz

#安装node包

[root@mysql-db01 tools]# rpm -ivh mha4mysql-node-0.56-0.el6.noarch.rpm

Preparing... ########################################### [100%]

1:mha4mysql-node ########################################### [100%]

#登录数据库

[root@mysql-db01 tools]# mysql -uroot -poldboy123

#添加mha管理账号

mysql> grant all privileges on \*.\* to mha@'10.0.0.%' identified by 'mha';

#查看是否添加成功

mysql> select user,host from mysql.user;

#主库上创建，从库会自动复制（在从库上查看）

mysql> select user,host from mysql.user;

### **4.2命令软连接（所有节点）**

#如果不创建命令软连接，检测mha复制情况的时候会报错

[root@mysql-db01 ~]# ln -s /application/mysql/bin/mysqlbinlog /usr/bin/mysqlbinlog

[root@mysql-db01 ~]# ln -s /application/mysql/bin/mysql /usr/bin/mysql

### **4.3部署管理节点（mha-manager）**

#### 4.3.1在mysql-db03上部署管理节点

#使用epel源

[root@mysql-db03 ~]# wget -O /etc/yum.repos.d/epel.repo <http://mirrors.aliyun.com/repo/epel-6.repo>

#安装manager依赖包

[root@mysql-db03 ~]# yum install -y perl-Config-Tiny epel-release perl-Log-Dispatch perl-Parallel-ForkManager perl-Time-HiRes

#安装manager包

[root@mysql-db03 tools]# rpm -ivh mha4mysql-manager-0.56-0.el6.noarch.rpm

Preparing... ########################################### [100%]

1:mha4mysql-manager ########################################### [100%]

#### 4.3.2编辑配置文件

#创建配置文件目录

[root@mysql-db03 ~]# mkdir -p /etc/mha

#创建日志目录

[root@mysql-db03 ~]# mkdir -p /var/log/mha/app1

#编辑mha配置文件

[root@mysql-db03 ~]# vim /etc/mha/app1.cnf

[server default]

manager\_log=/var/log/mha/app1/manager

manager\_workdir=/var/log/mha/app1

master\_binlog\_dir=/application/mysql/data

user=mha

password=mha

ping\_interval=2

repl\_password=oldboy123

repl\_user=rep

ssh\_user=root

[server1]

hostname=10.0.0.51

port=3306

[server2]

candidate\_master=1

check\_repl\_delay=0

hostname=10.0.0.52

port=3306

[server3]

hostname=10.0.0.53

port=3306

【配置文件详解】

[server default]

#设置manager的工作目录

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

#设置manager的日志

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

#设置master 保存binlog的位置，以便MHA可以找到master的日志，我这里的也就是mysql的数据目录

master\_binlog\_dir=/data/mysql

#设置自动failover时候的切换脚本

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

#设置手动切换时候的切换脚本

master\_ip\_online\_change\_script= /usr/local/bin/master\_ip\_online\_change

#设置mysql中root用户的密码，这个密码是前文中创建监控用户的那个密码

password=123456

#设置监控用户root

user=root

#设置监控主库，发送ping包的时间间隔，尝试三次没有回应的时候自动进行failover

ping\_interval=1

#设置远端mysql在发生切换时binlog的保存位置

remote\_workdir=/tmp

#设置复制用户的密码

repl\_password=123456

#设置复制环境中的复制用户名

repl\_user=rep

#设置发生切换后发送的报警的脚本

report\_script=/usr/local/send\_report

#一旦MHA到server02的监控之间出现问题，MHA Manager将会尝试从server03登录到server02

secondary\_check\_script= /usr/local/bin/masterha\_secondary\_check -s server03 -s server02 --user=root --master\_host=server02 --master\_ip=192.168.0.50 --master\_port=3306

#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）

shutdown\_script=""

#设置ssh的登录用户名

ssh\_user=root

[server1]

hostname=10.0.0.51

port=3306

[server2]

hostname=10.0.0.52

port=3306

#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

candidate\_master=1

#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

check\_repl\_delay=0

### **4.4配置ssh信任（所有节点）**

#创建秘钥对

[root@mysql-db01 ~]# ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa >/dev/null 2>&1

#发送公钥，包括自己

[root@mysql-db01 ~]# ssh-copy-id -i /root/.ssh/id\_dsa.pub [root@10.0.0.51](mailto:root@10.0.0.51)

[root@mysql-db01 ~]# ssh-copy-id -i /root/.ssh/id\_dsa.pub [root@10.0.0.52](mailto:root@10.0.0.52)

[root@mysql-db01 ~]# ssh-copy-id -i /root/.ssh/id\_dsa.pub [root@10.0.0.53](mailto:root@10.0.0.53)

### **4.5启动测试**

#测试ssh

[root@mysql-db03 ~]# masterha\_check\_ssh --conf=/etc/mha/app1.cnf

#看到如下字样，则测试成功

Tue Mar 7 01:03:33 2017 - [info] All SSH connection tests passed successfully.

#测试复制

[root@mysql-db03 ~]# masterha\_check\_repl --conf=/etc/mha/app1.cnf

#看到如下字样，则测试成功

MySQL Replication Health is OK.

### **4.6启动MHA**

#启动

[root@mysql-db03 ~]# nohup masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover < /dev/null > /var/log/mha/app1/manager.log 2>&1 &

### **4.7切换master测试**

#登录数据库（db02）

[root@mysql-db02 ~]# mysql -uroot -poldboy123

#检查复制情况

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.51

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000006

Read\_Master\_Log\_Pos: 191

Relay\_Log\_File: mysql-db02-relay-bin.000002

Relay\_Log\_Pos: 361

Relay\_Master\_Log\_File: mysql-bin.000006

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#登录数据库（db03）

[root@mysql-db03 ~]# mysql -uroot -poldboy123

#检查复制情况

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.51

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000006

Read\_Master\_Log\_Pos: 191

Relay\_Log\_File: mysql-db03-relay-bin.000002

Relay\_Log\_Pos: 361

Relay\_Master\_Log\_File: mysql-bin.000006

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#停掉主库

[root@mysql-db01 ~]# /etc/init.d/mysqld stop

Shutting down MySQL..... SUCCESS!

#登录数据库（db02）

[root@mysql-db02 ~]# mysql -uroot -poldboy123

#查看slave状态

mysql> show slave status\G

#db02的slave已经为空

Empty set (0.00 sec)

#登录数据库（db03）

[root@mysql-db03 ~]# mysql -uroot -poldboy123

#查看slave状态

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.52

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000006

Read\_Master\_Log\_Pos: 191

Relay\_Log\_File: mysql-db03-relay-bin.000002

Relay\_Log\_Pos: 361

Relay\_Master\_Log\_File: mysql-bin.000006

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

|  |
| --- |
| #grep -i 'change master' /var/log/mha/app1/manager 恢复旧master时操作，使得数据同步。 |

# 配置VIP漂移

### **5.1IP漂移的两种方式**

* 通过keepalived的方式，管理虚拟IP的漂移
* 通过MHA自带脚本方式，管理虚拟IP的漂移

### **5.2MHA脚本方式**

#### 5.2.1修改配置文件

#编辑配置文件

[root@mysql-db03 ~]# vim /etc/mha/app1.cnf

#在[server default]标签下添加

[server default]

#使用MHA自带脚本

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

#### 5.2.2[编辑脚本](master_ip_failover)

#根据配置文件中脚本路径编辑

[root@mysql-db03 ~]# vim /etc/mha/master\_ip\_failover

#修改以下几行内容

my $vip = '10.0.0.55/24';

my $key = '0';

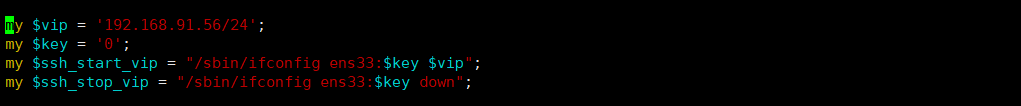
my $ssh\_start\_vip = "/sbin/ifconfig eth0:$key $vip";

my $ssh\_stop\_vip = "/sbin/ifconfig eth0:$key down";

#添加执行权限，否则mha无法启动

[root@mysql-db03 ~]# chmod +x /etc/mha/master\_ip\_failover

Centos7



#### 5.2.3手动绑定VIP

#绑定vip

[root@mysql-db01 ~]# ifconfig eth0:0 10.0.0.55/24

[root@mysql-db01 ~]#**ifconfig ens33:0 192.168.91.56/24**

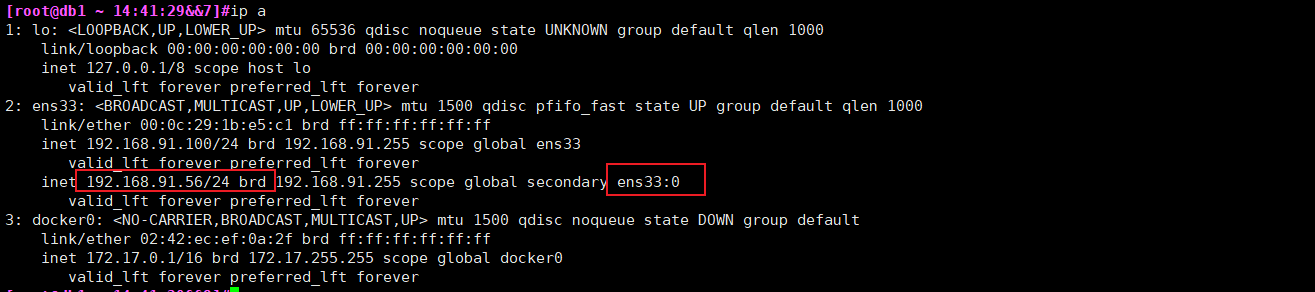
#查看vip

[root@mysql-db01 ~]# ip a |grep eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

inet 10.0.0.51/24 brd 10.0.0.255 scope global eth0

inet 10.0.0.55/24 brd 10.0.0.255 scope global secondary eth0:0



### **5.3 启动**

|  |
| --- |
| [root@mha1 /etc/mha ]#nohup masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover < /dev/null > /var/log/mha/app1/manager.log 2>&1 &报错 **解决**  **#dos2unix master\_ip\_failover** |

### **5.4测试ip漂移**

#登录db02

[root@mysql-db02 ~]# mysql -uroot -poldboy123

#查看slave信息

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.51

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000007

Read\_Master\_Log\_Pos: 191

Relay\_Log\_File: mysql-db02-relay-bin.000002

Relay\_Log\_Pos: 361

Relay\_Master\_Log\_File: mysql-bin.000007

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#停掉主库

[root@mysql-db01 ~]# /etc/init.d/mysqld stop

Shutting down MySQL..... SUCCESS!

#在db03上查看从库slave信息

mysql> show slave status\G

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

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

Master\_Host: 10.0.0.52

Master\_User: rep

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000006

Read\_Master\_Log\_Pos: 191

Relay\_Log\_File: mysql-db03-relay-bin.000002

Relay\_Log\_Pos: 361

Relay\_Master\_Log\_File: mysql-bin.000006

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#在db01上查看vip信息

[root@mysql-db01 ~]# ip a |grep eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

inet 10.0.0.51/24 brd 10.0.0.255 scope global eth0

#在db02上查看vip信息

[root@mysql-db02 ~]# ip a |grep eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

inet 10.0.0.52/24 brd 10.0.0.255 scope global eth0

inet 10.0.0.55/24 brd 10.0.0.255 scope global secondary eth0:0

# 六．配置binlog-server

### **6.1修改mha配置文件**

[root@mysql-db03 ~]# vim /etc/mha/app1.cnf

[binlog1]

no\_master=1

hostname=10.0.0.53

master\_binlog\_dir=/data/mysql/binlog/

### **6.2备份binlog**

#创建备份binlog目录

[root@mysql-db03 ~]# mkdir -p /data/mysql/binlog/

#进入该目录

[root@mysql-db03 ~]# cd /data/mysql/binlog/

#备份binlog

[root@mysql-db03 binlog]# mysqlbinlog -R --host=10.0.0.51 --user=mha --password=mha --raw --stop-never mysql-bin.000001 &

#启动mha

[root@mysql-db03 binlog]# nohup masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover < /dev/null > /var/log/mha/app1/manager.log 2>&1 &

### **6.3测试binlog备份**

#查看binlog目录中的binlog

[root@mysql-db03 binlog]# ll

total 44

-rw-r--r-- 1 root root 285 Mar 8 03:11 mysql-bin.000001

#登录主库

[root@mysql-db01 ~]# mysql -uroot -poldboy123

#刷新binlog

mysql> flush logs;

#再次查看binlog目录

[root@mysql-db03 binlog]# ll

total 48

-rw-r--r-- 1 root root 285 Mar 8 03:11 mysql-bin.000001

-rw-r--r-- 1 root root 143 Mar 8 04:00 mysql-bin.000002

# 七．mysql中间件Atlas

### **7.1Atlas简介**

Atlas是由 Qihoo 360公司Web平台部基础架构团队开发维护的一个基于MySQL协议的数据中间层项目。它在MySQL官方推出的MySQL-Proxy 0.8.2版本的基础上，修改了大量bug，添加了很多功能特性。它在MySQL官方推出的MySQL-Proxy 0.8.2版本的基础上，修改了大量bug，添加了很多功能特性。

### **7.2Atlas主要功能**

1.读写分离

2.从库负载均衡

3.IP过滤

4.自动分表

5.DBA可平滑上下线DB

6.自动摘除宕机的DB

### **7.3Atlas相对于官方MySQL-Proxy的优势**

1.将主流程中所有Lua代码用C重写，Lua仅用于管理接口

2.重写网络模型、线程模型

3.实现了真正意义上的连接池

4.优化了锁机制，性能提高数十倍

### **7.4安装Atlas**

同学们有福了，安装Atlas真的是炒鸡简单，官方提供的Atlas有两种：

Atlas (普通) : **[Atlas-2.2.1.el6.x86\_64.rpm](https://github.com/Qihoo360/Atlas/releases/download/2.2.1/Atlas-2.2.1.el6.x86_64.rpm)**

Atlas (分表) : **[Atlas-sharding\_1.0.1-el6.x86\_64.rpm](https://github.com/Qihoo360/Atlas/releases/download/sharding-1.0.1/Atlas-sharding_1.0.1-el6.x86_64.rpm)**

这里我们只需要下载普通的即可。

#在主库安装，进入安装包目录

[root@mysql-db01 ~]# cd /home/oldboy/tools/

#下载Atlas

[root@mysql-db01 tools]#

wget <https://github.com/Qihoo360/Atlas/releases/download/2.2.1/Atlas-2.2.1.el6.x86_64.rpm>

#安装

[root@mysql-db01 tools]# rpm -ivh Atlas-2.2.1.el6.x86\_64.rpm

Preparing... ########################################### [100%]

1:Atlas ########################################### [100%]

### **7.5编辑配置文件**

#进入Atlas工具目录

[root@mysql-db01 ~]# cd /usr/local/mysql-proxy/bin/

#生成密码

[root@mysql-db01 bin]# ./encrypt oldboy123

#修改Atlas配置文件

[root@mysql-db01 ~]# vim /usr/local/mysql-proxy/conf/test.cnf

#Atlas后端连接的MySQL主库的IP和端口，可设置多项，用逗号分隔

proxy-backend-addresses = 10.0.0.51:3306

#Atlas后端连接的MySQL从库的IP和端口

proxy-read-only-backend-addresses = 10.0.0.52:3306,10.0.0.53:3306

#用户名与其对应的加密过的MySQL密码

pwds = root:1N/CNLSgqXuTZ6zxvGQr9A==

#SQL日志的开关

sql-log = ON

#Atlas监听的工作接口IP和端口

proxy-address = 0.0.0.0:3307

#默认字符集，设置该项后客户端不再需要执行SET NAMES语句

charset = utf8

### **7.6启动Atlas**

[root@mysql-db01 ~]# /usr/local/mysql-proxy/bin/mysql-proxyd test start

OK: MySQL-Proxy of test is started

### **7.7Atlas管理操作**

#用atlas管理用户登录

[root@mysql-db01 ~]# mysql -uuser -ppwd -h127.0.0.1 -P2345

#查看可用命令帮助

mysql> select \* from help;

#查看后端代理的库

mysql> SELECT \* FROM backends;

+-------------+----------------+-------+------+

| backend\_ndx | address | state | type |

+-------------+----------------+-------+------+

| 1 | 10.0.0.51:3307 | up | rw |

| 2 | 10.0.0.53:3307 | up | ro |

| 3 | 10.0.0.52:3307 | up | ro |

+-------------+----------------+-------+------+

#平滑摘除mysql

mysql> REMOVE BACKEND 2;

Empty set (0.00 sec)

#检查是否摘除成功

mysql> SELECT \* FROM backends;

+-------------+----------------+-------+------+

| backend\_ndx | address | state | type |

+-------------+----------------+-------+------+

| 1 | 10.0.0.51:3307 | up | rw |

| 2 | 10.0.0.52:3307 | up | ro |

+-------------+----------------+-------+------+

#保存到配置文件中

mysql> SAVE CONFIG;

Empty set (0.06 sec)

