# 配置MYSQL集群

## MHA 集群概述

### MHA 简介

• MHA ( Master High Availability )

– 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。

– 目前在 MySQL 高可用方面是一个相对成熟的解决方案。

– 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 组成

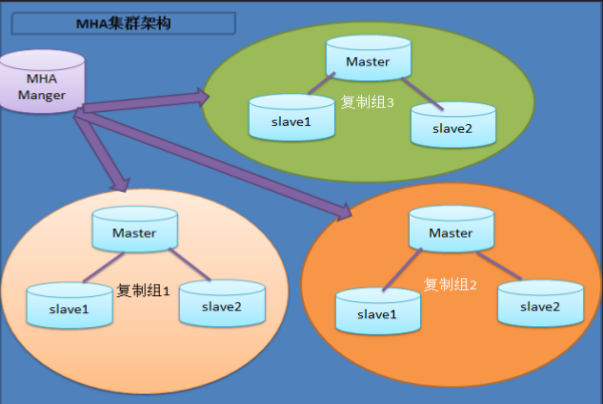
• MHA Manager (管理节点)

– 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 (数据节点)

– 运行在每台 MySQL 服务器上。

### MHA 集群架构



### MHA 工作过程

**• MHA Manager 会定时探测集群中的 master 节点,当 master 出现故障时,它可以自动将最新数据的 slave 提升为新的 master ,然后将所有其他的 slave重新指向新的 master 。整个故障转移过程对应用程序完全透明。**

– ( 1 )从宕机崩溃的 master 保存二进制日志事件( binlog events)

– ( 2 )识别含有最新更新的 slave

– ( 3 )应用差异的中继日志( relay log )到其他的 slave

– ( 4 )应用从 master 保存的二进制日志事件( binlog events )

– ( 5 )提升一个 slave 为新的 master ;

– ( 6 )使其他的 slave 连接新的 master 进行复制;

## 部署MHA集群

### 集群拓扑结构

### 配置MYSQL主从同步结构 一主多从

配置方法参考文档：<day3/MHA-master-slave.txt>

myslq51主库

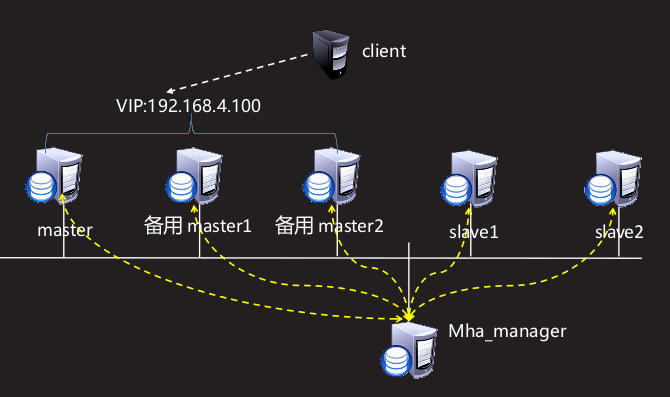
mysql52从库 备用主

mysql53 从库 备用主

mysql54 从库

mysql55 从库

mysql56 管理集群



### IP规划

|  |  |  |
| --- | --- | --- |
| 角色 | IP地址 | 主机名 |
| Master主节点服务器 | 192.168.4.51 | master51 |
| 备用 1 master 数据库服务器 | 192.168.4.52 | master52 |
| 备用 2 master 数据库服务器 | 192.168.4.53 | master53 |
| 第 1 台 slave 服务器 | 192.168.4.54 | slave54 |
| 第 2 台 slave 服务器 | 192.168.4.55 | slave55 |
| Mha\_manager 服务器 | 192.168.4.56 | mgm56 |
| VIP 地址 | 192.168.4.100 |  |

## 配置主从同步，要求如下：

51 主库 开半同步复制

52 从库（备用主库） 开半同步复制

53 从库（备用主库） 开半同步复制

54 从库 不做备用主库所以不用开半同步复制

55 从库 不做备用主库所以不用开半同步复制

### 配置master主节点

[root@master51 ~]# vim /etc/my.cnf

[mysqld]

server\_id=51

log-bin=master51

binlog\_format="mixed"

plugin-load = "rpl\_semi\_sync\_master=semisync\_master.so;

rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled = 1

rpl-semi-sync-slave-enabled = 1

[root@master51 ~]# systemctl restart mysqld

[root@master51 ~]# mysql -uroot -p123456

mysql> grant replication slave on \*.\*

-> to repluser@"%"

-> identified by "123456";

mysql> set global relay\_log\_purge=off; //不允许自动删除中继日志文件

mysql> show variables like "relay\_log\_purge";

mysql> show master status;

### 配置备用1主节点

[root@master52 ~]# vim /etc/my.cnf

[mysqld]

server\_id=52

log-bin=master52

binlog\_format="mixed"

plugin-load = "rpl\_semi\_sync\_master=semisync\_master.so;

rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled = 1

rpl-semi-sync-slave-enabled = 1

[root@master52 ~]# systemctl restart mysqld

[root@master52 ~]# mysql -uroot -p123456

mysql> grant replication slave on \*.\*

-> to repluser@"%"

-> identified by "123456";

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

mysql> show variables like "relay\_log\_purge";

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master51.000001",

-> master\_log\_pos=441;

mysql> start slave;

mysql> show slave status\G;

### 配置备用2主节点

[root@master53 ~]# vim /etc/my.cnf

[mysqld]

server\_id=53

log-bin=master53

binlog\_format="mixed"

plugin-load = "rpl\_semi\_sync\_master=semisync\_master.so;

rpl\_semi\_sync\_slave=semisync\_slave.so"

rpl-semi-sync-master-enabled = 1

rpl-semi-sync-slave-enabled = 1

[root@master53 ~]# systemctl restart mysqld

[root@master53 ~]# mysql -uroot -p123456

mysql> grant replication slave on \*.\*

-> to repluser@"%"

-> identified by "123456";

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

mysql> show variables like "relay\_log\_purge";

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master51.000001",

-> master\_log\_pos=441;

mysql> start slave;

mysql> show slave status\G;

### 配置2台slave服务器

**配置slave 1服务器**

[root@slave54 ~]# vim /etc/my.cnf

[mysqld]

server\_id=54

[root@slave54 ~]# systemctl restart mysqld

[root@slave54 ~]# mysql -uroot -p123456

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master51.000001",

-> master\_log\_pos=441;

mysql> start slave;

mysql> show slave status\G;

**配置slave 2服务器**

[root@slave55 ~]# vim /etc/my.cnf

[mysqld]

server\_id=55

[root@slave55 ~]# systemctl restart mysqld

[root@slave55 ~]# mysql -uroot -p123456

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123456",

-> master\_log\_file="master51.000001",

-> master\_log\_pos=441;

mysql> start slave;

mysql> show slave status\G;

### 准备集群环境

1. **在所有主机上安装perl软件包（51-56）**

]# cd mha-soft-student

]# ls perl-\*.rpm

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

perl-Mail-Sender-0.8.23-1.el7.noarch.rpm

perl-MIME-Types-1.38-2.el7.noarch.rpm

perl-Email-Date-Format-1.002-15.el7.noarch.rpm

perl-Mail-Sendmail-0.79-21.el7.art.noarch.rpm

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

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

perl-MIME-Lite-3.030-1.el7.noarch.rpm

]# yum -y install perl-\*.rpm

**2.** **在所有数据库服务器51-55节点上授权监控用户**

mysql> grant all on \*.\* to root@"%" identified by '123asd...A';

**3. 在所有服务器51-56上安装mha-node软件包**

# yum -y install perl-DBD-mysql

# rpm -ivh /root/mha-soft-student/mha4mysql-node-0.56-0.el6.noarch.rpm

# rpm -q mha4mysql-node

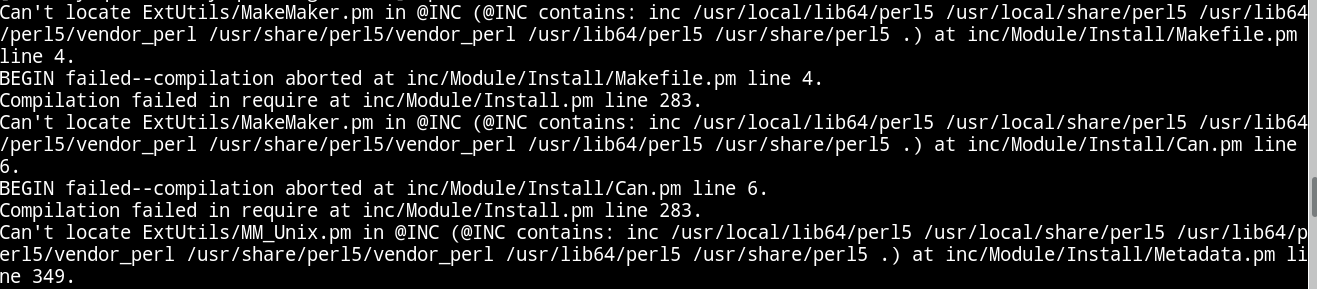
**4. 在管理主机上安装mha\_node 和 mha-manager包**

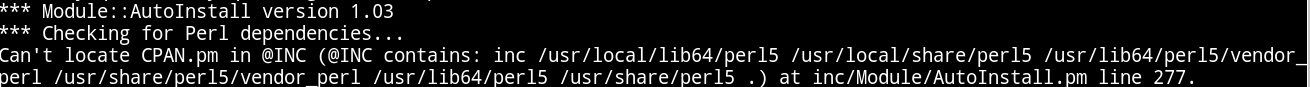
]# tar -xvf ./mha-soft-student/mha4mysql-manager-0.56.tar.gz

[root@mysql56 ~]# cd /root/mha4mysql-manager-0.56/

# perl Makefile.PL

根据提示解决依赖关系

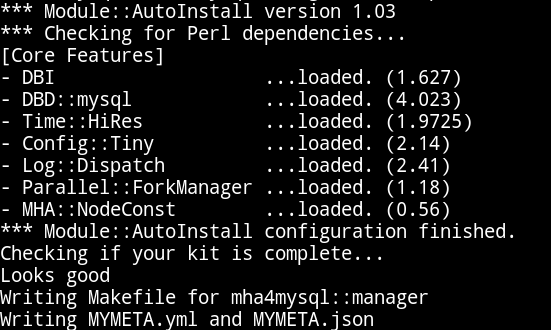




# yum -y install perl-ExtUtils\*

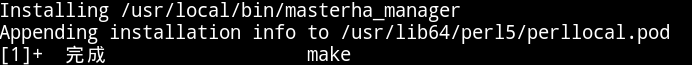
# yum -y install perl-CPAN\*

# perl Makefile.PL



# make & make install

结尾显示以下信息即成功



**5. 配置ssh密钥对认证登陆**

5.1 管理主机56可以无密码ssh连接所有数据库服务器51-55

5.2 数据库服务器主机之间可以互相无密码ssh连接

# ssh-keygen -f /root/.ssh/id\_rsa -N ''

# for i in 50 51 52 53 54 55 56 ; do ssh-copy-id 192.168.4.$i; done

### 安装 manager 软件包 后产生的命令

|  |  |
| --- | --- |
| 命令 | 作用 |
| 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 MHA 运行状态 |
| masterha\_master\_monitor | 检测 master 是否宕机 |

## 配置MJHA集群

### 1. 在管理主机56上创建管理命令

# cd mha4mysql-manager-0.56/bin/

# echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

# mkdir /root/bin

# cp \* /root/bin

### 2. 创建主配置文件并编辑

# mkdir /etc/mha

[root@mysql56 conf]# pwd

/root/mha4mysql-manager-0.56/samples/conf

# cp app1.cnf /etc/mha/

# vim /etc/mha/app1.cnf

[server default] //服务默认配置

manager\_workdir=/etc/mha //工作目录

manager\_log=/etc/mha/manager.log //日志文件

master\_ip\_failover\_script=/etc/mha/master\_ip\_failover //故障切换脚本

ssh\_user=root //ssh连接用户名

ssh\_port=22 //ssh连接端口号

repl\_user=repluser //主从同步用户名

repl\_password=123asd...A //主从同步密码

#监控数据库时，连接服务器的用户名及密码

user=root

password=123asd...A

#指定数据库服务器ip及角色

[server1]

hostname=192.168.4.51

candidate\_master=1

[server2]

hostname=192.168.4.52

candidate\_master=1

[server3]

hostname=192.168.4.53

candidate\_master=1

[server4]

hostname=192.168.4.54

no\_master=1

[server5]

hostname=192.168.4.55

no\_master=1

创建故障切换脚本并编辑

# pwd

/root/mha4mysql-manager-0.56/samples/scripts

# cp master\_ip\_failover /etc/mha/

【# cp /root/master\_ip\_failover /etc/mha/】

# chmod +x /etc/mha/master\_ip\_failover

# vim /etc/mha/master\_ip\_failover

35 my $vip = '192.168.4.100/24'; # Virtual IP

36 my $key = "1";

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

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

把脚本中指定的ip 地址手动部署在当前主库51上

[root@mysql51 ~]# ifconfig eth0:1 192.168.4.100/24

[root@mysql51 ~]# ifconfig eth0:1

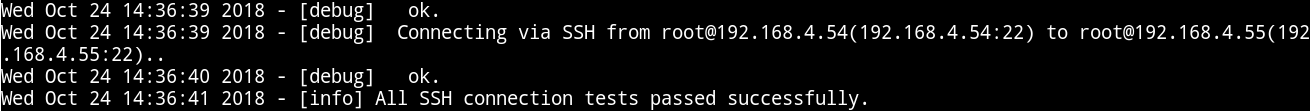
### 3. 启动服务

**3.1 检查ssh连接**

— masterha\_check\_ssh --conf=管理节点主配置文件

[root@mysql56 ~]# masterha\_check\_ssh --conf=/etc/mha/app1.cnf

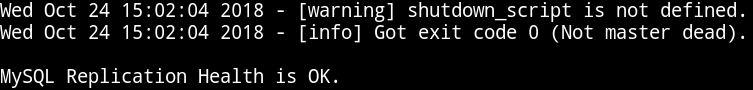
输出结束显示successfully字样即为成功



**3.2 检查主从同步配置**

—masterha\_check\_repl --conf=管理节点主配置文件

[root@mysql56 ~]# masterha\_check\_repl --conf=/etc/mha/app1.cnf



**3.3 启动管理服务**

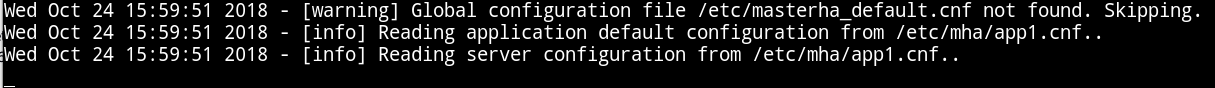
使用masterha\_manager工具

— --remove\_dead\_master\_conf //删除宕机主库配置

— --ignore\_last\_failover //忽略xxx.health文件

[root@mysql56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf

--remove\_dead\_master\_conf --ignore\_last\_failover



### 4. 查看状态：masterha\_check\_status

新开终端连接管理主机

[root@mysql56 ~]# masterha\_check\_status --conf=/etc/mha/app1.cnf

app1 (pid:6438) is running(0:PING\_OK), master:192.168.4.51

### 停止服务:masterha\_stop

[root@mysql56 ~]# masterha\_stop=/etc/mha/app1.cnf

### 5. 测试配置

**5.1 在客户端50主机上连接vip 192.168.4.100访问数据库服务**

主库做用户授权

[root@mysql51 ~]# mysql -uroot -p123asd...A

mysql> create database db3;

mysql> create table db3.a(id int);

mysql> grant select,insert on db3.\* to asd@'%' identified by '123asd...A';

在客户端使用授权用户连接

[root@client50 ~]# mysql -uasd -p123asd...A -h192.168.4.100

mysql> insert into db3.a values(123);

mysql> select \* from db3.a;（在52-55上都能查看到插入的数据）

**5.2 测试高可用配置**

## 测试集群VIP的故障切换功能

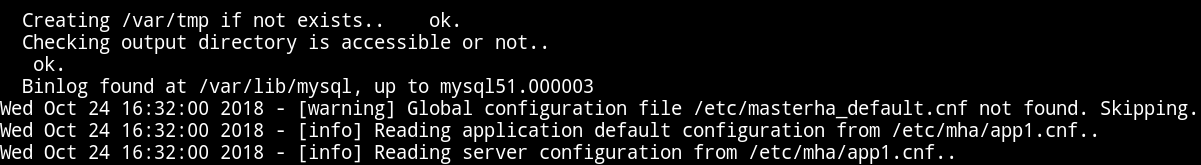
把主库51 的数据库服务停止

客户端仍然可以连接vip地址192.168.4.100 访问数据库服务

### 将主库51的数据库服务停止

[root@mysql51 ~]# systemctl stop mysqld

56主机:之前启动的管理服务会退出



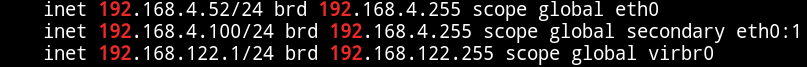
[root@client50 ~]# ping -c 2 192.168.4.100

[root@client50 ~]# mysql -uasd -p123asd...A -h192.168.4.100 （此时连接的是52或者53，通过mysql> select @@hostname;查看）

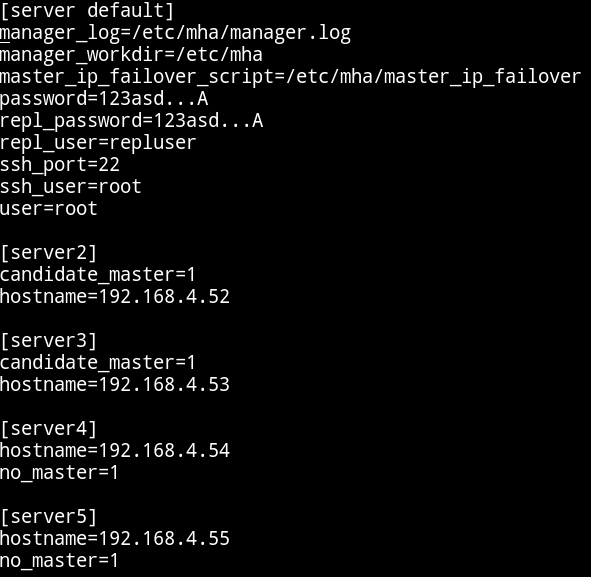
mysql> insert into db3.a values(222);

mysql> select \* from db3.a;（数路会同步到其他主机）

[root@mysql52 ~]# ip addr show | grep 192（VIP会自动切换到52上）



在/etc/mha/app1.cnf文件中主库51的配置被删除



**数据库53--55 把主库地址指向新选举出的主库地址，root管理员连接53-55数据库服务，通过**mysql> show slave status\G;**查看**

**5.3 恢复：把坏掉的数据库服务器51再添加到集群里**

**配置51主机：**

[root@mysql51 ~]# systemctl start mysqld

[root@mysql51 ~]# mysql -uroot -p123asd...A

mysql> change master to master\_host="192.168.4.52",

-> master\_user="repluser",

-> master\_password="123asd...A",

-> master\_log\_file="mysql52.000003",

-> master\_log\_pos=958;

mysql> start slave;

mysql> show slave status\G;

**配置56主机：将51主机的配置加入到管理主机的配置文件中**

[root@mysql56 ~]# vim /etc/mha/app1.cnf

[server1]

candidate\_master=1

hostname=192.168.4.51

**启动MHA\_Manager**

[root@mysql56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf

--remove\_dead\_master\_conf --ignore\_last\_failover

**新开终端连接到管理主机56上，查看状态显示当前监听到的运行master为52**

[root@mysql56 ~]# masterha\_check\_status --conf=/etc/mha/app1.cnf

app1 (pid:8406) is running(0:PING\_OK), master:192.168.4.52

将52上的binlog日志拷贝到主机51上进行手动数据同步

**把主库52的数据库服务停止后，客户端仍然可以连接**

**vip地址192.168.4.100访问MYSQL数据库服务**

[root@mysql52 ~]# mysql -uroot -p123asd...A

mysql> change master to master\_host='192.168.4.51',

-> master\_user='repluser',

-> master\_password='123asd...A',

-> master\_log\_file='mysql51.000004',

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

[root@mysql56 ~]# vim /etc/mha/app1.cnf

[server2]

candidate\_master=1

hostname=192.168.4.52

[root@mysql56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf

--remove\_dead\_master\_conf --ignore\_last\_failover

使用MAH软件+mysql主从同步 实现MYSQL高可用集群

perl

shell