**Morning：**

1. MHA概述

**-What’s MHA?**

MHA是一套优秀的实现Mysql高可用集群的解决方案；

数据库的自动故障切换操作能够做到30秒以内；在故障切换过程中保证数据的一致性，实现真正意义上的高可用；

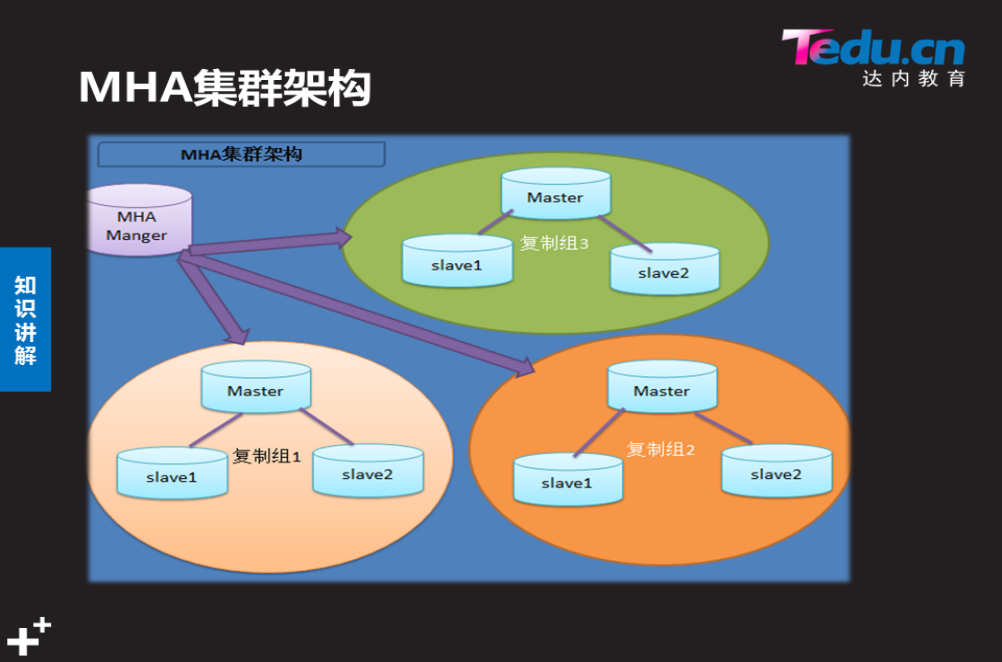
MHA要求至少一主两从，防止主库、从库1同时故障。

**-MHA的组成**

MHA Manager(管理节点)：可以部署在独立机器，也可以部署在一台slave

MHA Node(数据节点)：运行在每台Mysql服务器

**-MHA工作过程**

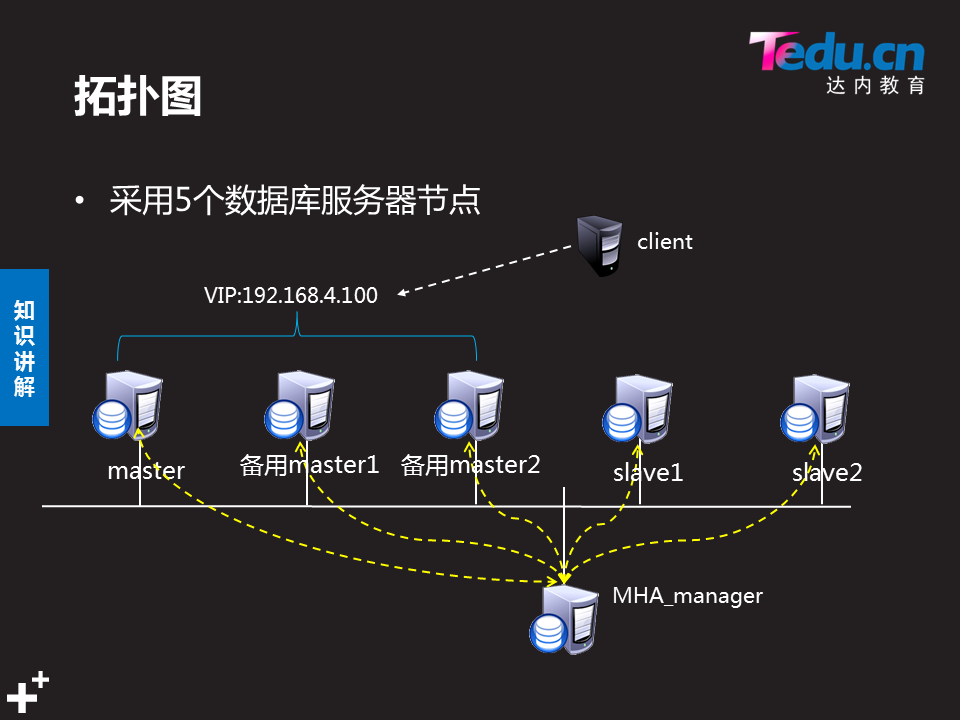


1. 主库宕机时，保存binlog日志
2. MHA自动识别具有最新更新数据的备用主库
3. 将原主库的binlog日志应用到该备用主库
4. 提升备用主库为新的主库
5. 所有slave把主库地址自动切换为新主库的IP，通过新主库同步数据。

**Afternoon：**

1、部署MHA集群

**-拓扑结构**



|  |  |  |
| --- | --- | --- |
| IP地址 | 主机名 | 角色 |
| 192.168.4.51 | master51 | 主节点 |
| 192.168.4.52 | master52 | 备用1主节点 |
| 192.168.4.53 | master53 | 备用2主节点 |
| 192.168.4.54 | slave54 | 第1台slave |
| 192.168.4.55 | slave55 | 第2台slave |
| 192.168.4.56 | host56 | MHA\_manager |
| 192.168.4.100 |  | VIP地址 |
| 192.168.4.50 | client50 | 客户端 |

**步骤1：配置主节点/备用主节点**

###以下为master51~master53的相同配置###

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

log\_bin=masterXX

server\_id=XX

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

#启动binlog日志，开启主、从半同步复制功能

[root@masterXX ~]# systemctl restart mysqld

#重启mysql服务

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

#设置不自动删除本机的中继日志文件

myslq> grant replication slave on \*.\* to repluser@’%’

> identified by ‘123456’;

#授权同步用户

###master52~master53配置为51的从库###

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;

###验证主从同步、半同步复制###

show variables like 'rpl\_semi\_sync\_%\_enabled'; #检查半同步复制

mysql> show master status; #检查主从同步

mysql> show slave status \G;

**步骤2：配置从服务器**

###以下为slave54~55的相同配置###

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

server\_id=XX

plugin-load=rpl\_semi\_sync\_slave=semisync\_slave.so

rpl\_semi\_sync\_slave\_enabled=1

#设置从服务器server\_id，开启从服务器的半同步复制功能

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;

**步骤3：测试一主多从结构**

###在master51建表、添加授权用户yaya200###

mysql> create database db8;

mysql> create table db8.a (id int);

mysql> grant select,insert on db8.\* to yaya200@'%'

-> identified by '123456';

###客户端client50插入记录，测试是否同步成功###

[root@client50 ~]# mysql -u'yaya200' -p123456 -h'192.168.4.51'

mysql> insert into db8.a values (111),(222);

mysql> select \* from db8.a;

**步骤4：集群环境准备（主机51~56）**

###执行以下指令，实现免密登陆###

[root@xx ~]# yum -y install expect

[root@xx ~]# ssh-keygen -N '' -f /root/.ssh/id\_rsa

[root@xx ~]# for i in {51..55}; do

> expect << EOF

> spawn ssh-copy-id -o StrictHostKeyChecking=no 192.168.4.$i

> expect "password" {send "123456\n"}

> expect

> EOF

> done

#51~55可以互相免密登陆，56可以免密登陆所有服务器

###主机51~56安装perl软件包###

[root@xx ~]# yum -y install perl-\*

#由于MHA是由Perl语言编写，需要安装perl依赖包

###主机51~56安装MHA依赖包###

[root@xx ~]# cd /root/mha-soft-student/

[root@xx mha-soft-student]# rpm -ivh perl-\*.rpm

###主机51~56安装mha\_node软件包###

[root@xx ~]# cd /root/mha-soft-student/

[root@xx mha-soft-student]# rpm -ivh mha4mysql-node-...rpm

###主机56安装mha\_manager软件包###

[root@host56 mha-soft-student]# tar -zxvf mha4mysql-mana...tar.gz

[root@host56 mha-soft-student]# cd mha4mysql-manager-0.56/

[root@host56 mha4mysql-manager-0.56]# perl Makefile.PL

#perl配置(检查依赖关系)

[root@host56 mha4mysql-manager-0.56]# make && make install

#编译、安装

[root@host56 ~]# ls /usr/local/bin/

masterha\_check\_repl masterha\_master\_monitor

masterha\_check\_ssh masterha\_master\_switch

masterha\_check\_status masterha\_secondary\_check

masterha\_conf\_host masterha\_stop

masterha\_manager

#查看manager节点命令

**步骤5：配置集群**

###创建集群配置文件###

[root@host56 ~]# mkdir /etc/mha/

[root@host56 conf]# pwd

/root/mha-soft-student/mha4mysql-manager-0.56/samples/conf/

[root@host56 conf]# cp ./app1.cnf /etc/mha/

###编辑配置文件###

[root@host56 ~]# vim /etc/mha/app1.cnf

[server default] #集群服务器的默认设置

manager\_workdir=/etc/mha #集群管理器的工作目录

manager\_log=/etc/mha/manager.log #工作日志

master\_ip\_failover\_script=/etc/mha/master\_ip\_failover

#故障切换脚本：监视主库，当主机宕机通过脚本将**VIP地址切换**到备用主库。

ssh\_user=root #无密码登陆的用户名、端口号

ssh\_port=22

repl\_user=repluser #主从同步用户名、密码

repl\_password=123456

user=root #用于监视master主库状态的用户

password=123456

[server1]

hostname=192.168.4.51

port=3306 #默认为3306，可以不写

candidate\_master=1 #竞选主库

[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 #不竞选主库

[server5]

hostname=192.168.4.55

no\_master=1

###创建故障切换脚本###

[root@host56 ~]# cp /root/mha-soft-student/master\_ip\_failover /etc/mha/ && chmod +x /etc/mha/master\_ip\_failover

#拷贝脚本模板，给执行权限

[root@host56 ~]# vim /etc/mha/master\_ip\_failover

...

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

my $key = "1"; #eth0:1编号

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

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

...

###在竞选主库服务器上添加授权用户###

mysql> grant all on \*.\* to root@'%' identified by '123456';

#在主机51执行，其他数据库服务器会自动同步

###把VIP地址绑定在主库服务器的eth0接口###

[root@master51 ~]# ifconfig eth0:1 192.168.4.100

#为了VIP可切换，必须**临时配置**

**步骤6：测试配置并启动MHA\_Manager**

[root@host56 ~]# masterha\_check\_ssh --conf=/etc/mha/app1.cnf

[root@host56 ~]# masterha\_check\_repl --conf=/etc/mha/app1.cnf

#测试ssh连接与主从同步健康性

[root@host56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover

#--remove\_dead..：当主库宕机，会把当前主库的配置从app1.cnf中删掉

#--ignore...：忽略xxx.health，当8小时内连续发生宕机，照样执行VIP切换

[root@host56 ~]# masterha\_check\_status --conf=/etc/mha/app1.cnf

#在**新终端**查看状态信息

**步骤7：验证高可用集群**

当主库宕机，manager的管理服务会自杀，自动触发VIP切换脚本。

###模拟主库master51宕机###

[root@master51 ~]# systemctl stop mysqld

###验证VIP是否自动切换###

[root@master51 ~]# ifconfig eth0:1 #无IP

[root@master52 ~]# ifconfig eth0:1 #IP变为192.168.4.100

###验证客户机访问的服务器地址###

[root@client50 ~]# mysql -uroot -p123456 -h'192.168.4.100'

mysql> select @@hostname;

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

| @@hostname |

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

| master52 |

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

###验证所有从库的IO线程指向###

[root@master53 ~]# mysql -uroot -p123456 -e 'show slave status\G' | grep -i Master\_Host

Master\_Host: 192.168.4.52

[root@master53 ~]# mysql -uroot -p123456 -e "show slave status\G" | grep -i running:

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

###验证客户插入记录的数据同步情况###

mysql> insert into db8.a values(333),(333);

#在client50插入记录

[root@xx ~]# mysql -uroot -p123456 -e 'select \* from db8.a;'

#在数据库服务器验证数据同步情况

2、MHA的宕机恢复

本实验模拟master51宕机后，修复服务并作为备用主节点加入到MHA。

**步骤1：手动启动mysqld服务**

[root@master51 ~]# systemctl start mysqld

**步骤2：手动配置为master52的从库**

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

-> master\_user='repluser',master\_password='123456',

-> master\_log\_file='master52.000095',master\_log\_pos=428;

mysql> start slave;

mysql> show slave status \G;

**步骤3：手动还原宕机过程的数据**

方法1：完全备份/完全恢复

[root@master52 ~]# mysqldump -uroot -p123456 db8 > /root/db8.sql

[root@master52 ~]# scp /root/db8.sql root@192.168.4.51:/root/

[root@master51 ~]# mysql -uroot -p123456 db8 < /root/db8.sql

方法2：在配置成从库时，自定义日志偏移量

[root@master52 ~]# mysqlbinlog /var/lib/mysql/master52.000095

#查看新增数据的起始偏移量

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

-> master\_user='repluser',master\_password='123456',

-> master\_log\_file='master52.000095',master\_log\_pos=293;

**步骤4：修改MHA配置文件**

[root@host56 ~]# vim /etc/mha/app1.cnf

[server1]

candidate\_master=1

hostname=192.168.4.51

**步骤5：启动管理服务**

[root@host56 ~]# masterha\_manager --conf=/etc/mha/app1.cnf --remove\_dead\_master\_conf --ignore\_last\_failover