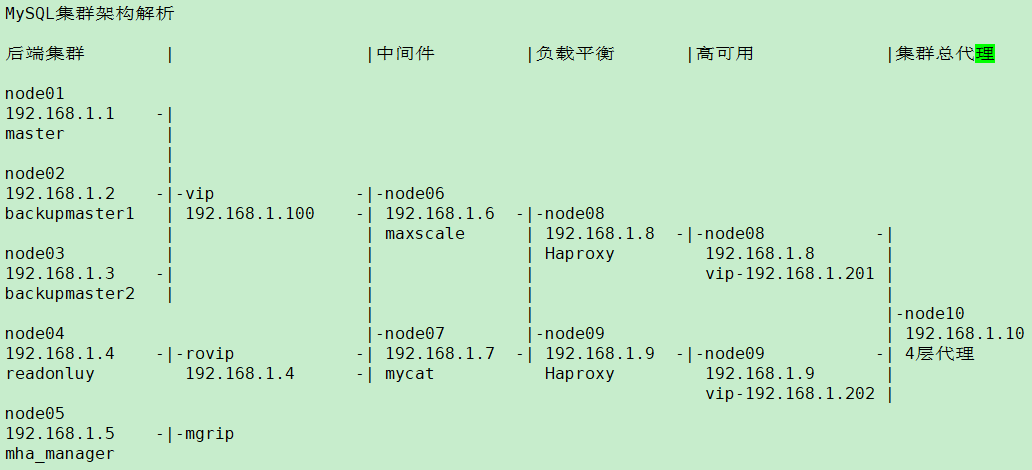
MySQL集群架构

总规划图



1、mha集群

环境：centos7.4虚拟机5台，关闭防火墙、SELinux，清空iptables规则，配置好yum源

规划： node01 192.168.1.1 主

Node02 192.168.1.2 备1

Node03 192.168.1.3 备2

Node04 192.168.1.4 从

Node05 192.168.1.5 管理节点

环境准备：所有节点可以相互免密登录，修改好hosts文件，可以通过主机名通信

1、安装MySQL服务

[root@hostos mysql]# for i in node0{1..5}

> do

> scp mha\_soft.tar.gz mysql-5.7.17.tar $i:/root

> done

[root@hostos ~]# vim install\_mysql.sh

[root@hostos ~]# cat install\_mysql.sh

#!/bin/bash

tar -xf mysql-5.7.17.tar

yum -y install ./mysql-community-\*.rpm

rm -rf mysql-community-\*.rpm

rm -rf mysql-5.7.17.tar

rm -rf $0

[root@hostos ~]# for i in node0{1..4}

> do

> scp install\_mysql.sh $i:/root

> ssh $i "/bin/bash /root/install\_mysql.sh"

> done

#修改各数据库节点密码策略

[root@node01 ~]# systemctl start mysqld

[root@node01 ~]# cat /var/log/mysqld.log | grep password

2020-01-03T05:40:36.255488Z 1 [Note] A temporary password is generated for root@localhost: z(kmWULc4Eep

[root@node01 ~]# vim /etc/my.cnf

[root@node01 ~]# sed -rn '4,7p' /etc/my.cnf

[mysqld]

validate\_password\_policy = 0

validate\_password\_length = 6

character\_set\_server = "utf8"

[root@node01 ~]# systemctl restart mysqld

[root@node01 ~]# mysql -hlocalhost -uroot -p'z(kmWULc4Eep'

mysql> alter user root@'localhost' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

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

Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

[root@node01 ~]#

[root@node02 ~]# systemctl start mysqld

[root@node02 ~]# cat /var/log/mysqld.log | grep password

2020-01-03T05:48:15.293554Z 1 [Note] A temporary password is generated for root@localhost: EIKAHhcLt9-t

[root@node02 ~]# vim /etc/my.cnf

[root@node02 ~]# sed -rn '4,7p' /etc/my.cnf

[mysqld]

validate\_password\_policy = 0

validate\_password\_length = 6

character\_set\_server = "utf8"

[root@node02 ~]# systemctl restart mysqld

[root@node02 ~]# mysql -hlocalhost -uroot -p'EIKAHhcLt9-t'

mysql> alter user root@'localhost' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

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

Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.01 sec)

mysql> exit

Bye

[root@node02 ~]#

[root@node03 ~]# systemctl start mysqld

[root@node03 ~]# cat /var/log/mysqld.log | grep password

2020-01-03T05:53:26.295886Z 1 [Note] A temporary password is generated for root@localhost: Nrns:\*#:Y7Au

[root@node03 ~]# vim /etc/my.cnf

[root@node03 ~]# sed -rn '4,7p' /etc/my.cnf

[mysqld]

validate\_password\_policy = 0

validate\_password\_length = 6

character\_set\_server = "utf8"

[root@node03 ~]# systemctl restart mysqld

[root@node03 ~]# mysql -hlocalhost -uroot -p'Nrns:\*#:Y7Au'

mysql> alter user root@'localhost' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

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

Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

[root@node03 ~]#

[root@node04 ~]# systemctl start mysqld

[root@node04 ~]# cat /var/log/mysqld.log | grep password

2020-01-03T05:59:26.118595Z 1 [Note] A temporary password is generated for root@localhost: liCf!6t08u8T

[root@node04 ~]# vim /etc/my.cnf

[root@node04 ~]# sed -rn '4,7p' /etc/my.cnf

[mysqld]

validate\_password\_policy = 0

validate\_password\_length = 6

character\_set\_server = "utf8"

[root@node04 ~]# systemctl restart mysqld

[root@node04 ~]# mysql -hlocalhost -uroot -p'liCf!6t08u8T'

mysql> alter user root@'localhost' identified by '123456';

Query OK, 0 rows affected (0.00 sec)

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

Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

[root@node04 ~]#

2、设置半同步复制，开启binlog日志

[root@node01 ~]# vim /etc/my.cnf

[root@node01 ~]# sed -rn '9,16p' /etc/my.cnf

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

relay\_log\_purge = 0

server\_id = 1

log\_bin = 'node01'

binlog\_format = 'mixed'

[root@node01 ~]# systemctl restart mysqld

[root@node02 ~]# vim /etc/my.cnf

[root@node02 ~]# sed -rn '9,16p' /etc/my.cnf

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

relay\_log\_purge = 0

server\_id = 2

log\_bin = 'node02'

binlog\_format = 'mixed'

[root@node02 ~]# systemctl restart mysqld

[root@node02 ~]#

[root@node03 ~]# vim /etc/my.cnf

[root@node03 ~]# sed -rn '9,16p' /etc/my.cnf

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

relay\_log\_purge = 0

server\_id = 3

log\_bin = 'node03'

binlog\_format = 'mixed'

[root@node03 ~]# systemctl restart mysqld

[root@node03 ~]#

[root@node04 ~]# vim /etc/my.cnf

[root@node04 ~]# sed -rn '9,16p' /etc/my.cnf

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

relay\_log\_purge = 0

server\_id = 4

log\_bin = 'node04'

binlog\_format = 'mixed'

[root@node04 ~]# systemctl restart mysqld

[root@node04 ~]#

3、构建主从同步架构，4台机器设置为一主三从结构

[root@node01 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

> grant replication slave on \*.\* to repluser@'%' identified by '123456';

> reset master;

> show master status;

> EOF

File Position Binlog\_Do\_DB Binlog\_Ignore\_DB Executed\_Gtid\_Set

node01.000001 154

[root@node01 ~]#

[root@node02 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

> grant replication slave on \*.\* to repluser@'%' identified by '123456';

> reset slave;

> change master to master\_host="192.168.1.1", master\_user="repluser", master\_password="123456", master\_log\_file="node01.000001", master\_log\_pos=154;

> start slave;

> EOF

[root@node02 ~]# mysql -hlocalhost -uroot -p123456 -e "show slave status\G" 2> /dev/null | grep -i yes

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

[root@node02 ~]#

[root@node03 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

> grant replication slave on \*.\* to repluser@'%' identified by '123456';

> reset slave;

> change master to master\_host="192.168.1.1", master\_user="repluser", master\_password="123456", master\_log\_file="node01.000001", master\_log\_pos=154;

> start slave;

> EOF

[root@node03 ~]# mysql -hlocalhost -uroot -p123456 -e "show slave status\G" 2> /dev/null | grep -i "yes"

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

[root@node03 ~]#

[root@node04 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

> reset slave;

> change master to master\_host="192.168.1.1", master\_user="repluser", master\_password="123456", master\_log\_file="node01.000001", master\_log\_pos=154;

> start slave;

> EOF

[root@node04 ~]# mysql -hlocalhost -uroot -p123456 -e "show slave status\G" 2> /dev/null | grep -i "yes"

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

[root@node04 ~]#

##注意，主和两个备主都需要授权同一个主从同步用户

#测试主从同步

[root@node01 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

> create database mydb;

> EOF

[root@hostos ~]# for i in node0{1..4}

> do

> ssh $i "mysql -hlocalhost -uroot -p123456 -e 'show databases' 2> /dev/null | grep -i 'mydb'"

> done

Warning: Permanently added 'node01,192.168.1.1' (ECDSA) to the list of known hosts.

mydb

Warning: Permanently added 'node02,192.168.1.2' (ECDSA) to the list of known hosts.

mydb

Warning: Permanently added 'node03,192.168.1.3' (ECDSA) to the list of known hosts.

mydb

Warning: Permanently added 'node04,192.168.1.4' (ECDSA) to the list of known hosts.

mydb

[root@hostos ~]#

#主从结构构建成功

4、部署MHA

[root@hostos ~]# vim install\_mha.sh

[root@hostos ~]# cat install\_mha.sh

#!/bin/bash

tar -xf mha\_soft.tar.gz

cd mha-soft-student

yum -y install ./\*.rpm

rm –rf \*.rpm

cd ..

rm -rf mha\_soft.tar.gz

rm -rf $0

[root@hostos ~]# for i in node0{1..5}

> do

> scp install\_mha.sh $i:/root

> ssh $i "/bin/bash /root/install\_mha.sh"

> done

[root@node05 ~]# yum -y install perl-ExtUtils-MakeMaker perl-CPAN

[root@node05 ~]# cd mha-soft-student/

[root@node05 mha-soft-student]# ls

master\_ip\_failover mha4mysql-manager-0.56.tar.gz

[root@node05 mha-soft-student]# tar -xf mha4mysql-manager-0.56.tar.gz

[root@node05 mha-soft-student]# cd mha4mysql-manager-0.56/

[root@node05 mha4mysql-manager-0.56]# perl Makefile.PL

\*\*\* Module::AutoInstall version 1.03

\*\*\* Checking for Perl dependencies...

[Core Features]

- DBI ...loaded. (1.627)

- DBD::mysql ...loaded. (4.023)

- Time::HiRes ...loaded. (1.9725)

- Config::Tiny ...loaded. (2.14)

- Log::Dispatch ...loaded. (2.41)

- Parallel::ForkManager ...loaded. (1.18)

- MHA::NodeConst ...loaded. (0.56)

\*\*\* Module::AutoInstall configuration finished.

Checking if your kit is complete...

Looks good

Writing Makefile for mha4mysql::manager

[root@node05 mha4mysql-manager-0.56]# make && make install

[root@node05 ~]# mkdir /etc/mha

[root@node05 ~]# touch /etc/mha/mha\_master.cnf

[root@node05 ~]# vim /etc/mha/mha\_master.cnf

[root@node05 ~]# cat /etc/mha/mha\_master.cnf

[server default]

manager\_workdir = /etc/mha/

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

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

ssh\_user = root

ssh\_port = 22

repl\_user = repluser

repl\_password = 123456

user = root

password = 123456

[server1]

hostname = 192.168.1.1

port = 3306

candidate\_master = 1

[server2]

hostname = 192.168.1.2

port = 3306

candidate\_master = 1

[server3]

hostname = 192.168.1.3

port = 3306

candidate\_master = 1

[server4]

hostname = 192.168.1.4

port = 3306

no\_master = 1

[root@node05 ~]#

[root@node05 ~]# cp mha-soft-student/master\_ip\_failover /etc/mha/

[root@node05 ~]# vim /etc/mha/master\_ip\_failover

[root@node05 ~]# sed -rn '35p' /etc/mha/master\_ip\_failover

my $vip = '192.168.1.100/24'; # Virtual IP

[root@node05 ~]#

#指定主服务器配置虚拟IP地址

[root@node01 ~]# ip addr add 192.168.1.100/24 brd 192.168.1.255 dev eth0 label eth0:1

[root@node01 ~]# ifconfig eth0:1

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255

ether 52:54:00:52:f4:ca txqueuelen 1000 (Ethernet)

##测试配置

[root@node05 ~]# touch /etc/masterha\_default.cnf

[root@node05 ~]# masterha\_check\_ssh --conf=/etc/mha/mha\_master.cnf

Fri Jan 3 15:30:05 2020 - [info] All SSH connection tests passed successfully.

[root@node05 ~]#

#ssh免密通信没有问题

[root@node05 ~]# chmod 777 /etc/mha/master\_ip\_failover

[root@node05 ~]# masterha\_check\_repl --conf=/etc/mha/mha\_master.cnf

MySQL Replication Health is OK.

[root@node05 ~]#

#MySQL节点主从同步没有问题

5、测试集群

#启动监听进程

[root@node05 ~]# masterha\_manager --conf=/etc/mha/mha\_master.cnf --ignore\_last\_failover

Fri Jan 3 15:35:12 2020 - [info] Reading default configuration from /etc/masterha\_default.cnf..

Fri Jan 3 15:35:12 2020 - [info] Reading application default configuration from /etc/mha/mha\_master.cnf..

Fri Jan 3 15:35:12 2020 - [info] Reading server configuration from /etc/mha/mha\_master.cnf..

#测试集群可用性

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456

mysql> use mydb;

Database changed

mysql> create table mytb(

-> id int(2),

-> name char(20)

-> );

Query OK, 0 rows affected (0.64 sec)

mysql> exit

Bye

[root@hostos ~]#

[root@hostos ~]# for i in node0{1..4}

> do

> echo $i

> ssh $i "mysql -hlocalhost -uroot -p123456 -e 'select count(\*) from mydb.mytb' 2> /dev/null" 2> /dev/null

> done

node01

count(\*)

0

node02

count(\*)

0

node03

count(\*)

0

node04

count(\*)

0

#集群读写没有问题

#测试故障切换

[root@node01 ~]# systemctl stop mysqld

[root@node02 ~]# ifconfig eth0:1

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255

ether 52:54:00:cc:30:a4 txqueuelen 1000 (Ethernet)

#vip漂移到备1

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 -e "show databases" 2> /dev/null | grep -i "mydb"

Mydb

#集群高可用

#修复宕机节点

[root@node02 ~]# mysql -hlocalhost -uroot -p123456 -e "show master status\G" 2> /dev/null

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

File: node02.000001

Position: 441

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

[root@node01 ~]# systemctl start mysqld

[root@node01 ~]# mysql -hlocalhost -uroot -p123456 2> /dev/null << EOF

reset slave;

change master to master\_host="192.168.1.2", master\_user="repluser", master\_password="123456", master\_log\_file="node02.000001", master\_log\_pos=441;

start slave;

EOF

[root@node01 ~]# mysql -hlocalhost -uroot -p123456 -e "show slave status\G" 2> /dev/null | grep -i yes

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

[root@node01 ~]#

[root@node05 ~]# masterha\_check\_repl --conf=/etc/mha/mha\_master.cnf

MySQL Replication Health is OK.

[root@node05 ~]#

#主从结构没问题，从新启动监控进程

[root@node05 ~]# masterha\_manager --conf=/etc/mha/mha\_master.cnf --ignore\_last\_failover

#客户端测试

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 -e "show databases" 2> /dev/null | grep -i "mydb"

Mydb

MySQL中间件maxscale和mycat

环境：centos7.4虚拟机两台，关闭防火墙、SELinux，配置好yum源，清空iptables规则

后端：mha高可用集群，vip地址192.168.1.100，只读服务器192.168.1.4

规划： node06 192.168.1.6 maxscale

Node07 192.168.1.7 mycat

##部署node06

[root@node06 ~]# ls

maxscale-2.1.2-1.rhel.7.x86\_64.rpm

[root@node06 ~]# yum -y install ./maxscale-2.1.2-1.rhel.7.x86\_64.rpm

[root@node06 ~]# vim /etc/maxscale.cnf

[root@node06 ~]# cat /etc/maxscale.cnf | grep -v ^$ | grep -v ^#

[maxscale]

threads=auto

[server1]

type=server

address=192.168.1.100

port=3306

protocol=MySQLBackend

[server2]

type=server

address=192.168.1.4

port=3306

protocol=MySQLBackend

[MySQL Monitor]

type=monitor

module=mysqlmon

servers=server1,server2

user=maxscalemon

passwd=123456

monitor\_interval=10000

[Read-Write Service]

type=service

router=readwritesplit

servers=server1,server2

user=maxscalerouter

passwd=123456

max\_slave\_connections=100%

[MaxAdmin Service]

type=service

router=cli

[Read-Write Listener]

type=listener

service=Read-Write Service

protocol=MySQLClient

port=4006

[MaxAdmin Listener]

type=listener

service=MaxAdmin Service

protocol=maxscaled

socket=default

port=4009

[root@node06 ~]#

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 2> /dev/null << EOF

grant replication slave, replication client on \*.\* to maxscalemon@'%' identified by '123456';

grant select on mysql.\* to maxscalerouter@'%' identified by '123456';

EOF

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 -e "select user from mysql.user" 2> /dev/null | grep -i 'maxscale'

maxscalemon

maxscalerouter

[root@hostos ~]#

#测试授权用户

[root@node06 ~]# yum -y install maraidb

[root@node06 ~]# mysql -h192.168.1.100 -umaxscalemon -p123456 -e "show databases" &> /dev/null

[root@node06 ~]# echo $?

0

[root@node06 ~]# mysql -h192.168.1.4 -umaxscalemon -p123456 -e "show databases" &> /dev/null

[root@node06 ~]# echo $?

0

[root@node06 ~]# mysql -h192.168.1.100 -umaxscalerouter -p123456 -e "show databases" &> /dev/null

[root@node06 ~]# echo $?

0

[root@node06 ~]# mysql -h192.168.1.4 -umaxscalerouter -p123456 -e "show databases" &> /dev/null

[root@node06 ~]# echo $?

0

[root@node06 ~]#

#启动服务

[root@node06 ~]# maxscale -f /etc/maxscale.cnf

[root@node06 ~]# netstat -antpu | grep maxscale

tcp 0 0 192.168.1.6:43642 192.168.1.100:3306 ESTABLISHED 1277/maxscale

tcp 0 0 192.168.1.6:41968 192.168.1.4:3306 ESTABLISHED 1277/maxscale

tcp6 0 0 :::4006 :::\* LISTEN 1277/maxscale

tcp6 0 0 :::4009 :::\* LISTEN 1277/maxscale

[root@node06 ~]#

#测试配置

[root@node06 ~]# maxadmin -uadmin -pmariadb -P4009

MaxScale> list servers

Servers.

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

Server | Address | Port | Connections | Status

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

server1 | 192.168.1.100 | 3306 | 0 | Master, Running

server2 | 192.168.1.4 | 3306 | 0 | Slave, Running

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

MaxScale> list services

Services.

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

Service Name | Router Module | #Users | Total Sessions | Backend databases

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

Read-Write Service | readwritesplit | 1 | 1 | server1, server2

MaxAdmin Service | cli | 3 | 4 |

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

MaxScale> exit

[root@node06 ~]#

#授权maxscale用户，针对于mydb库

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 2> /dev/null << EOF

> grant all on mydb.\* to maxuser@'%' identified by '123456';

> EOF

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456 -e "select user from mysql.user" 2> /dev/null | grep "maxuser"

maxuser

[root@hostos ~]#

#测试用户

[root@hostos ~]# mysql -h192.168.1.6 -P 4006 -umaxuser -p123456 -e "select count(\*) from mydb.mytb;" 2> /dev/null

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

| count(\*) |

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

| 0 |

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

[root@hostos ~]#

##部署node07

[root@node07 ~]# ls

Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz

[root@node07 ~]# yum -y install java-1.8.0-openjdk-devel

[root@node07 ~]# tar -xf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz

[root@node07 ~]# mv mycat/ /usr/local/

[root@node07 ~]# cd /usr/local/mycat/

[root@node07 mycat]# ls

bin catlet conf lib logs version.txt

[root@node07 mycat]# vim conf/server.xml

[root@node07 mycat]# cp conf/server.xml conf/server.xml.bak

[root@node07 mycat]# vim conf/server.xml

[root@node07 mycat]# sed -rn '80,89p' conf/server.xml

<user name="puser">

<property name="password">123456</property>

<property name="schemas">mydb</property>

</user>

<user name="mycatreader">

<property name="password">123456</property>

<property name="schemas">mydb</property>

<property name="readOnly">true</property>

</user>

[root@node07 mycat]#

[root@node07 mycat]# yum -y install mariadb

[root@node07 mycat]# mysql -h192.168.1.100 -uroot -p123456 2> /dev/null << EOF

> grant select on mydb.\* to mycatreader@'%' identified by '123456';

> EOF

[root@node07 mycat]# mysql -h192.168.1.100 -uroot -p123456 -e "select user from mysql.user" 2> /dev/null | grep "mycat"

mycatreader

[root@node07 mycat]#

[root@node07 mycat]# cp conf/schema.xml conf/schema.xml.bak

[root@node07 mycat]# vim conf/schema.xml

[root@node07 mycat]# cat conf/schema.xml

<?xml version="1.0"?>

<!DOCTYPE mycat:schema SYSTEM "schema.dtd">

<mycat:schema xmlns:mycat="http://io.mycat/">

<schema name="mydb" checkSQLschema="false" sqlMaxLimit="100" dataNode="mha1"></schema>

<dataNode name="mha1" dataHost="localhost1" database="mydb" />

<dataHost name="localhost1" maxCon="1000" minCon="10" balance="3" writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">

<heartbeat>select user()</heartbeat>

<writeHost host="master" url="192.168.1.100:3306" user="root" password="123456">

<readHost host="slavereadonly" url="192.168.1.4:3306" user="mycatreader" password="123456" />

</writeHost>

</dataHost>

</mycat:schema>

[root@node07 mycat]# ./bin/mycat start

Starting Mycat-server...

[root@node07 mycat]# jps

1546 WrapperSimpleApp

1580 Jps

[root@node07 mycat]# netstat -antpu | grep java

tcp 0 0 127.0.0.1:32000 0.0.0.0:\* LISTEN 1546/java

tcp6 0 0 :::45103 :::\* LISTEN 1546/java

tcp6 0 0 :::41529 :::\* LISTEN 1546/java

tcp6 0 0 :::1984 :::\* LISTEN 1546/java

tcp6 0 0 :::8066 :::\* LISTEN 1546/java

tcp6 0 0 :::9066 :::\* LISTEN 1546/java

tcp6 0 0 192.168.1.7:45666 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 127.0.0.1:31000 127.0.0.1:32000 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:55072 192.168.1.4:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:55070 192.168.1.4:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45650 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45660 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45664 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45648 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45652 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:55066 192.168.1.4:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45654 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45662 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:55068 192.168.1.4:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45656 192.168.1.100:3306 ESTABLISHED 1546/java

tcp6 0 0 192.168.1.7:45658 192.168.1.100:3306 ESTABLISHED 1546/java

[root@node07 mycat]#

#重新授权用户puser

[root@hostos ~]# mysql -h192.168.1.100 -uroot -p123456

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

Query OK, 0 rows affected, 1 warning (0.13 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.13 sec)

mysql> exit

Bye

[root@hostos ~]#

#测试puser用户读写分离

[root@hostos ~]# mysql -h192.168.1.6 -P4006 -upuser -p123456 -e "select @@hostname" 2> /dev/null

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

| @@hostname |

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

| node04 |

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

[root@hostos ~]# mysql -h192.168.1.7 -P8066 -upuser -p123456 -e "select @@hostname" 2> /dev/null

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

| @@hostname |

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

| node04 |

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

[root@hostos ~]# mysql -h192.168.1.6 -P4006 -upuser -p123456 -e "insert into mydb.mytb values(1, 'zhangsan');" 2> /dev/null

[root@hostos ~]# echo $?

0

[root@hostos ~]# mysql -h192.168.1.7 -P8066 -upuser -p123456 -e "insert into mydb.mytb values(2, 'lisi');" 2> /dev/null

[root@hostos ~]# echo $?

0

[root@hostos ~]# mysql -h192.168.1.7 -P8066 -upuser -p123456 -e "select count(\*) from mydb.mytb;" 2> /dev/null

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

| count(\*) |

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

| 2 |

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

[root@hostos ~]# mysql -h192.168.1.6 -P4006 -upuser -p123456 -e "select count(\*) from mydb.mytb;" 2> /dev/null

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

| count(\*) |

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

| 2 |

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

[root@hostos ~]#

MySQL中间件负载平衡

环境：centos7.4虚拟机两台，关闭防火墙、SELinux，清空iptables规则，搭建好yum源

规划： node08 192.168.1.8 Haproxy1

Node09 192.168.1.9 Haproxy2

##部署node08

[root@node08 ~]# yum -y install haproxy

[root@node08 ~]# vim /etc/haproxy/haproxy.cfg

[root@node08 ~]# sed -rn '87,$p' /etc/haproxy/haproxy.cfg

#---------------------------------------------------------------------

# healthy check

#---------------------------------------------------------------------

listen stats 0.0.0.0:1080

stats refresh 30s

stats uri /stats

stats realm Haproxy Manager

stats auth admin:admin

#---------------------------------------------------------------------

# round robin balancing between the mysql proxy maxscale and mycat

#---------------------------------------------------------------------

listen proxy 0.0.0.0:3306

mode tcp # mysql 得使用 tcp 协议

option tcpka # 使用长连接

balance leastconn # 最小连接调度算法

server maxscale 192.168.1.6:4006 check inter 3000 rise 1 maxconn 1000 fall 3

server mycat 192.168.1.7:8066 check inter 3000 rise 1 maxconn 1000 fall 3

[root@node08 ~]# systemctl restart haproxy

[root@node08 ~]# netstat -antpu | grep haproxy

tcp 0 0 0.0.0.0:1080 0.0.0.0:\* LISTEN 1356/haproxy

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 1356/haproxy

udp 0 0 0.0.0.0:45010 0.0.0.0:\* 1355/haproxy

[root@node08 ~]#

[root@node09 ~]# yum -y install haproxy

[root@node09 ~]# scp 192.168.1.8:/etc/haproxy/haproxy.cfg /etc/haproxy/

haproxy.cfg 100% 3987 2.5MB/s 00:00

[root@node09 ~]# systemctl start haproxy

[root@node09 ~]# netstat -antpu | grep haproxy

tcp 0 0 0.0.0.0:1080 0.0.0.0:\* LISTEN 1208/haproxy

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 1208/haproxy

udp 0 0 0.0.0.0:58233 0.0.0.0:\* 1207/haproxy

[root@node09 ~]#

#测试负载平衡

[root@hostos ~]# mysql -h192.168.1.8 -upuser -p123456 -e "select @@hostname" 2> /dev/null

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

| @@hostname |

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

| node04 |

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

[root@hostos ~]# mysql -h192.168.1.9 -upuser -p123456 -e "select @@hostname" 2> /dev/null

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

| @@hostname |

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

| node04 |

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

[root@hostos ~]# mysql -h192.168.1.8 -upuser -p123456 -e "insert into mydb.mytb values (3, 'wangwu')" 2> /dev/null

[root@hostos ~]# mysql -h192.168.1.9 -upuser -p123456 -e "insert into mydb.mytb values (4, 'zhaoliu')" 2> /dev/null

[root@hostos ~]# mysql -h192.168.1.9 -upuser -p123456 -e "select count(\*) from mydb.mytb" 2> /dev/null

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

| count(\*) |

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

| 4 |

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

[root@hostos ~]# mysql -h192.168.1.8 -upuser -p123456 -e "select count(\*) from mydb.mytb" 2> /dev/null

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

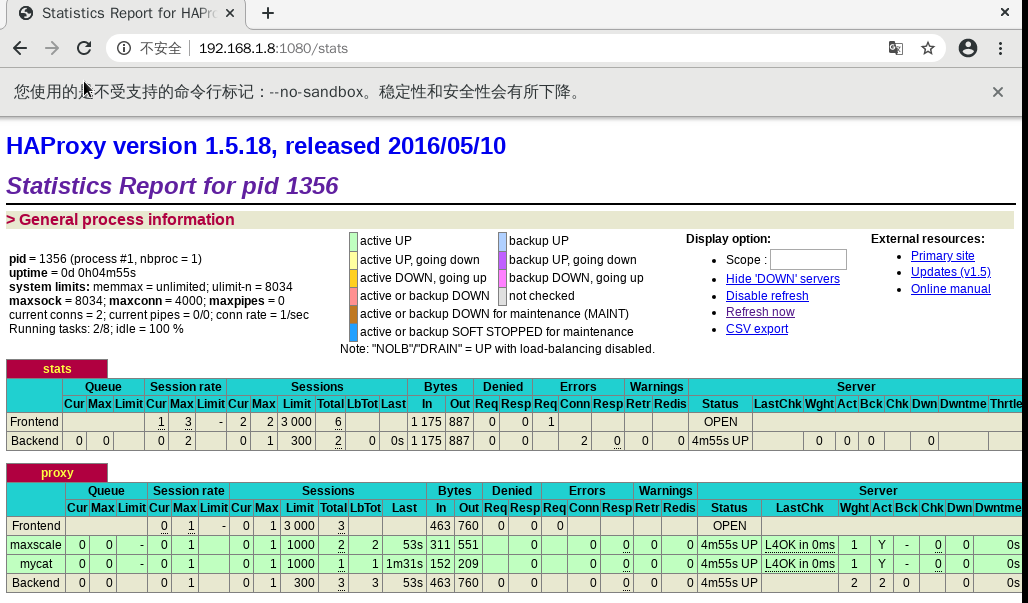
| count(\*) |

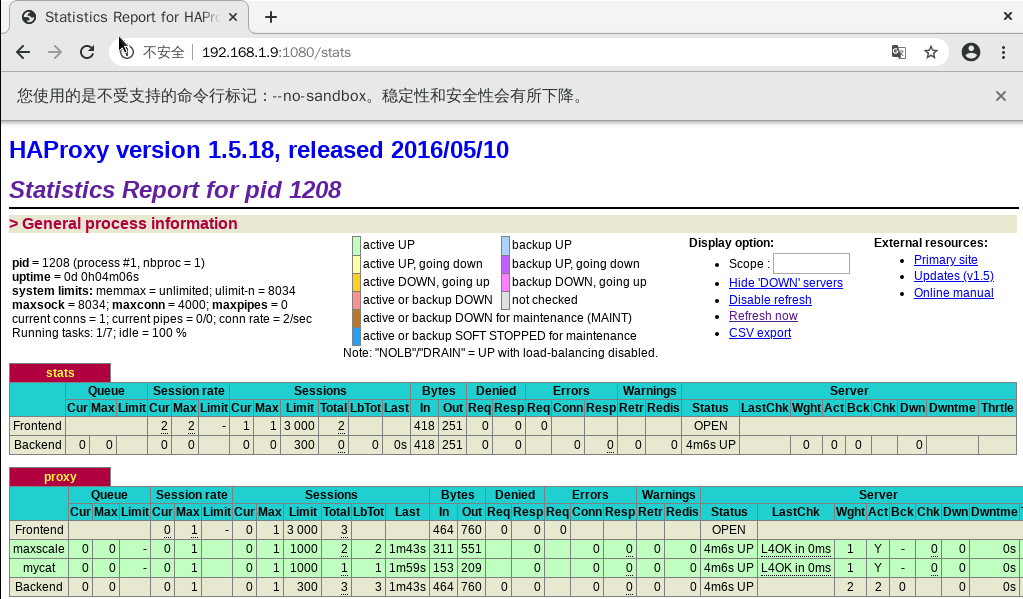
+----------+

| 4 |

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

[root@hostos ~]#





#中间件负载平衡实现

##验证负载平衡

[root@hostos ~]# mysql -h192.168.1.8 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.8 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]# mysql -h192.168.1.9 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.9 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]#

#因为中间件一个是maxscale，一个是mycat，所以能看到5个数据库的是通过maxscale，只能看到mydb库的是mycat

负载平衡中间件高可用

环境：centos7.4虚拟机两台（安装了Haproxy的两个机器）

规划：部署Keepalived软件实现高可用，提供VIP

Node08 192.168.1.8 Keepalived

Node09 192.168.1.9 Keepalived

[root@node08 ~]# yum -y install keepalived

[root@node08 ~]# vim /etc/keepalived/keepalived.conf

[root@node08 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id mysql\_proxy

}

vrrp\_script chk\_ha {

script "killall -0 haproxy"

interval 2

}

vrrp\_instance proxy\_1 {

state MASTER

interface eth0

virtual\_router\_id 51

priority 200

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.201/24 brd 192.168.1.255 dev eth0 label eth0:1

}

track\_script {

chk\_ha weight=0

}

}

vrrp\_instance proxy\_2 {

state BACKUP

interface eth0

virtual\_router\_id 52

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.202/24 brd 192.168.1.255 dev eth0 label eth0:2

}

track\_script {

chk\_ha weight=0

}

}

[root@node08 ~]# systemctl restart keepalived.service

[root@node08 ~]# ip a s | grep 192.168.1.

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node08 ~]#

[root@node09 ~]# yum -y install keepalived

[root@node09 ~]# scp 192.168.1.8:/etc/keepalived/keepalived.conf /etc/keepalived/

[root@node09 ~]# vim /etc/keepalived/keepalived.conf

[root@node09 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id mysql\_proxy

}

vrrp\_script chk\_ha {

script "killall -0 haproxy"

interval 2

}

vrrp\_instance proxy\_1 {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.201/24 brd 192.168.1.255 dev eth0 label eth0:1

}

track\_script {

chk\_ha weight=0

}

}

vrrp\_instance proxy\_2 {

state MASTER

interface eth0

virtual\_router\_id 52

priority 200

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.202/24 brd 192.168.1.255 dev eth0 label eth0:2

}

track\_script {

chk\_ha weight=0

}

}

[root@node09 ~]# systemctl restart keepalived.service

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node09 ~]#

#测试高可用

[root@hostos ~]# mysql -h192.168.1.201 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.201 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]# mysql -h192.168.1.202 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.202 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]#

#测试vip漂移

[root@node08 ~]# systemctl stop keepalived.service

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

[root@node08 ~]#

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node09 ~]#

[root@node08 ~]# systemctl start keepalived.service

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node09 ~]# systemctl stop keepalived.service

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

[root@node09 ~]#

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node08 ~]#

[root@node09 ~]# systemctl start keepalived.service

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node09 ~]#

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node08 ~]#

#测试Keepalived关联Haproxy

[root@node08 ~]# systemctl stop haproxy.service

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

[root@node08 ~]#

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node09 ~]#

[root@node08 ~]# systemctl start haproxy.service

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node08 ~]#

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node09 ~]# systemctl stop haproxy.service

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

[root@node09 ~]#

[root@node08 ~]# ip a s | grep 192.168.1

inet 192.168.1.8/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary eth0:1

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node08 ~]#

[root@node09 ~]# systemctl start haproxy.service

[root@node09 ~]# ip a s | grep 192.168.1

inet 192.168.1.9/24 brd 192.168.1.255 scope global eth0

inet 192.168.1.202/24 brd 192.168.1.255 scope global secondary eth0:2

[root@node09 ~]#

#Keepalived关联Haproxy配置中interval设置为2s，所以由于Haproxy造成的vip漂移抢占时间会有2s的时间差

Nginx的4层代理

环境：centos7.4虚拟机1台，关闭防火墙、SELinux，清空iptables规则，搭建好yum源

规划：node10 192.168.1.10 nginx

[root@node10 ~]# yum -y install gcc make zlib-devel pcre-devel openssl-devel

[root@node10 ~]# useradd -s /sbin/nologin nginx

[root@node10 ~]# tar -xf nginx-1.12.2.tar.gz

[root@node10 ~]# cd nginx-1.12.2/

[root@node10 nginx-1.12.2]# cp -r contrib/vim/\* /usr/share/vim/vimfiles/

[root@node10 nginx-1.12.2]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module --with-stream --with-http\_stub\_status\_module --without-http\_autoindex\_module --without-http\_ssi\_module

[root@node10 nginx-1.12.2]# make && make install

[root@node10 nginx-1.12.2]# cd ..

[root@node10 ~]# rm -rf nginx-1.12.2 nginx-1.12.2.tar.gz

[root@node10 ~]# cat >> /lib/systemd/system/nginx.service << EOF

> [Unit]

> Description=nginx

> After=network.target

>

> [Service]

> Type=forking

> PIDFile=/usr/local/nginx/logs/nginx.pid

> ExecStartPre=/usr/local/nginx/sbin/nginx -tc /usr/local/nginx/conf/nginx.conf

> ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

> ExecReload=/usr/local/nginx/sbin/nginx -s reload

> ExecStop=/usr/local/nginx/sbin/nginx/-s quit

> PrivateTmp=true

>

> [Install]

> WantedBy=multi-user.target

> EOF

[root@node10 ~]# ls /lib/systemd/system/nginx.service

/lib/systemd/system/nginx.service

[root@node10 ~]#

[root@node10 ~]# cd /usr/local/nginx/

[root@node10 nginx]# vim conf/nginx.conf

[root@node10 nginx]# cat conf/nginx.conf

user nginx;

worker\_processes auto;

events {

worker\_connections 1024;

}

stream {

upstream mysql\_proxys {

server 192.168.1.201:3306;

server 192.168.1.202:3306;

}

server {

listen 3306;

proxy\_pass mysql\_proxys;

}

}

[root@node10 nginx]# ./sbin/nginx -t

nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful

[root@node10 nginx]# ./sbin/nginx

[root@node10 nginx]# netstat -antpu | grep nginx

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 4131/nginx: master

[root@node10 nginx]#

#测试代理

[root@hostos ~]# mysql -h192.168.1.10 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]# mysql -h192.168.1.10 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]# mysql -h192.168.1.10 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.10 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| Database |

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

| information\_schema |

| mydb |

| mysql |

| performance\_schema |

| sys |

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

[root@hostos ~]# mysql -h192.168.1.10 -upuser -p123456 -e "show databases;" 2> /dev/null

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

| DATABASE |

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

| mydb |

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

[root@hostos ~]#

4层代理实现