## HA 高可用集群

提高服务的在线时间

可用性==在线时间/(在线时间+故障修复时间)

## keepalived

基于vrrp(虚拟路由冗余协议)协议工作的软件

优先级选举主备的角色

0---255

主服务器周期发送心跳信息

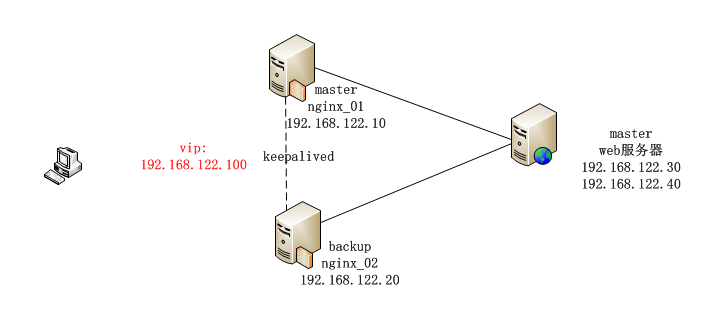
组播 224.0.0.18

虚拟MAC地址

在VRRP协议实现中，虚拟路由器使用00-00-5E-00-01-XX作为虚拟MAC地址。

xx：把组号转换为十六进制数

## 示例01：keepalived实现nginx高可用



1、web服务器部署基于IP地址的虚拟主机

2、安装配置nginx

3、安装keepalived软件

[root@nginx\_01 ~]# yum install -y keepalived

[root@nginx\_02 ~]# yum install -y keepalived

4、编辑nginx\_01的keepalived

[root@nginx\_01 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id nginx\_01

}

vrrp\_instance nginx\_service {

state MASTER

interface eth0

virtual\_router\_id 51

priority 200

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

5、将主服务器的keepalived配置文件拷贝，做适当修改

[root@nginx\_02 ~]# rsync -av 192.168.122.10:/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf

[root@nginx\_02 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id nginx\_02

}

vrrp\_instance nginx\_service {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

6、分别启动keepalived服务

[root@nginx\_01 ~]# systemctl start keepalived

[root@nginx\_01 ~]# systemctl enable keepalived.service

7、验证keepalived原理

1) 查看心跳信息

[root@nginx\_01 ~]# tcpdump -i eth0 -nn vrrp

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

12:16:44.272277 IP 192.168.122.10 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 200, authtype simple, intvl 1s, length 20

12:16:45.273478 IP 192.168.122.10 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 200, authtype simple, intvl 1s, length 20

12:16:46.274678 IP 192.168.122.10 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 200, authtype simple, intvl 1s, length 20

keepalived默认使用组播的方式传递心跳信息，有的云平台是禁止组播数据传递的，这时，需要手动将keepalived的心跳改为单播

vrrp\_instance nginx\_server {

……

unicast\_src\_ip 192.168.122.10 //指定源IP

unicast\_peer {

192.168.122.20 //指定对端IP

}

}

验证：

[root@nginx\_02 ~]# tcpdump -i eth0 -nn vrrp

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

12:27:07.965927 IP 192.168.122.10 > 192.168.122.20: VRRPv2, Advertisement, vrid 51, prio 200, authtype simple, intvl 1s, length 20

12:27:08.966761 IP 192.168.122.10 > 192.168.122.20: VRRPv2, Advertisement, vrid 51, prio 200, authtype simple, intvl 1s, length 20

**方案一：**

定义外部脚本检测服务状态实现VIP的故障转移

[root@nginx\_01 ~]# cat /etc/keepalived/check\_nginx.sh

#!/bin/bash

#

curl 127.0.0.1 &> /dev/null

if [ $? -ne 0 ]; then

systemctl stop keepalived

fi

[root@nginx\_01 ~]# chmod a+x /etc/keepalived/check\_nginx.sh

编辑keepalived配置文件

1) 定义外部脚本

vrrp\_script check\_nginx\_service {

script "/etc/keepalived/check\_nginx.sh"

interval 1

}

2、在vrrp实例中调用外部脚本

vrrp\_instance nginx\_service {

……

track\_script {

check\_nginx\_service

}

}

**方案二：**

1) 通过killall发送0信号检测nginx服务的状态

vrrp\_script check\_nginx\_service {

script "killall -0 nginx"

interval 1

rise 2

fall 2

weight -180

}

2) 使用外部脚本

vrrp\_instance nginx\_service {

……

track\_script {

check\_nginx\_service

}

}

keepalived默认情况，master服务器会自动抢占VIP

配置master不抢占VIP，仅需要在高优先级调度器上配置

[root@nginx01 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id nginx01

}

vrrp\_script check\_nginx {

script "killall -0 nginx"

interval 3

rise 2

fall 2

weight -80

}

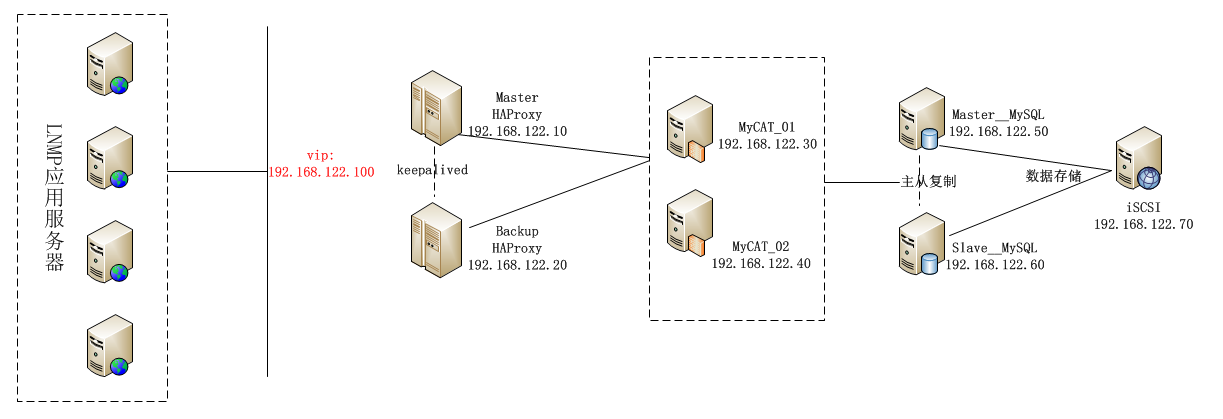
vrrp\_instance VI\_1 {

state BACKUP

nopreempt

interface eth0

## 示例02：keepalived + Haproxy + MyCAT + MySQL



1、安装配置iSCSI存储

[root@iscsi ~]# yum install -y targetcli

[root@iscsi ~]# targetcli

/> /backstores/block create bk\_01 /dev/vdb

/> /iscsi create iqn.2018-09.com.linux:jf1-jg1-IBM-disk

/> /iscsi/iqn.2018-09.com.linux:jf1-jg1-ibm-disk/tpg1/luns create /backstores/block/bk\_01

/> /iscsi/iqn.2018-09.com.linux:jf1-jg1-ibm-disk/tpg1/acls create iqn.2018-09.com.linux:client

同样的方法共享另一块硬盘

[root@iscsi ~]# systemctl start target

[root@iscsi ~]# systemctl enable target

Created symlink from /etc/systemd/system/multi-user.target.wants/target.service to /usr/lib/systemd/system/target.service.

[root@iscsi ~]# netstat -antp | grep :3260

tcp 0 0 0.0.0.0:3260 0.0.0.0:\* LISTEN -

2、安装MySQL服务器，连接iSCSI存储

[root@master\_mysql ~]# yum install -y mariadb-server iscsi-initiator-utils

[root@master\_mysql ~]# cat /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-09.com.linux:client

[root@master\_mysql ~]#

[root@master\_mysql ~]# iscsiadm -m discovery -t st -p 192.168.122.70

192.168.122.70:3260,1 iqn.2018-09.com.linux:jf1-jg1-ibm-disk

[root@master\_mysql ~]#

[root@master\_mysql ~]# iscsiadm -m node -T iqn.2018-09.com.linux:jf1-jg1-ibm-disk -p 192.168.122.70 -l

[root@master\_mysql ~]# mkfs.ext4 /dev/sda

[root@master\_mysql ~]# cat /etc/fstab

/dev/sda /var/lib/mysql ext4 defaults,\_netdev 0 0

[root@master\_mysql ~]# mount -a

启动mysql服务

[root@master\_mysql ~]# systemctl start mariadb

[root@master\_mysql ~]# systemctl enable mariadb

MySQL从服务器连接第二块共享硬盘

在数据库服务器创建允许mycat连接的用户

MariaDB [(none)]> grant all on \*.\* to 'admin'@'%' identified by 'redhat';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;

Query OK, 0 rows affected (0.01 sec)

3、配置MySQL主从复制环境，测试主从复制正常

4、安装配置MyCAT实现读写分离

1) 配置主机名解析

[root@mycat\_01 ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.122.30 mycat\_01

2) 安装jdk

[root@mycat\_01 ~]# mkdir /app

[root@mycat\_01 ~]# tar xf jdk-8u91-linux-x64.tar.gz -C /app/

[root@mycat\_01 ~]# tail -n 2 /etc/profile

export JAVA\_HOME=/app/jdk1.8.0\_91

export PATH=$PATH:$JAVA\_HOME/bin

[root@mycat\_01 ~]# source /etc/profile

[root@mycat\_01 ~]# java -version

java version "1.8.0\_91"

Java(TM) SE Runtime Environment (build 1.8.0\_91-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)

3) 安装mycat

[root@mycat\_01 ~]# tar xf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz -C /app/

[root@mycat\_01 ~]# tail -n 2 /etc/profile

export MYCAT\_HOME=/app/mycat

[root@mycat\_01 ~]# source /etc/profile

4) 配置mycat

[root@mycat\_01 conf]# vim /app/mycat/conf/schema.xml

<?xml version="1.0"?>

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

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

<schema name="game" checkSQLschema="false" sqlMaxLimit="100" dataNode="dn1">

</schema>

<dataNode name="dn1" dataHost="localhost1" database="game" />

<dataHost name="localhost1" maxCon="1000" minCon="10" balance="1"

writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">

<heartbeat>select user()</heartbeat>

<writeHost host="hostM1" url="192.168.122.50:3306" user="admin"

password="redhat">

</writeHost>

<writeHost host="hostS1" url="192.168.122.60:3306" user="admin"

password="redhat" />

</dataHost>

</mycat:schema>

[root@mycat\_01 conf]# vim /app/mycat/conf/server.xml

81 <user name="mycatuser">

82 <property name="password">redhat</property>

83 <property name="schemas">game</property>

84 </user>

5) 启动mycat

[root@mycat\_01 conf]# /app/mycat/bin/mycat start

Starting Mycat-server...

[root@mycat\_01 conf]# netstat -antp | grep java

tcp 0 0 127.0.0.1:32000 0.0.0.0:\* LISTEN 11047/java

tcp6 0 0 :::39980 :::\* LISTEN 11047/java

tcp6 0 0 :::40920 :::\* LISTEN 11047/java

tcp6 0 0 :::1984 :::\* LISTEN 11047/java

tcp6 0 0 :::8066 :::\* LISTEN 11047/java

tcp6 0 0 :::9066 :::\* LISTEN 11047/java

6) 测试mycat连接

[root@localhost ~]# mysql -u mycatuser -p -h 192.168.122.30 -P 8066

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MySQL connection id is 1

Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloundDB)

将安装好的第一个mycat所有配置复制第二台mycat服务器

[root@mycat\_01 ~]# scp -r /app/\* 192.168.122.40:/app/

[root@mycat\_01 ~]# scp /etc/profile 192.168.122.40:/etc/profile

[root@mycat\_02 ~]# source /etc/profile

[root@mycat\_02 ~]# /app/mycat/bin/mycat start

5、安装配置HAProxy实现MyCAT的调度

[root@ha\_master ~]# yum install -y haproxy

[root@ha\_master ~]# cat /etc/haproxy/haproxy.cfg

global

maxconn 2000

nbproc 1

user nobody

group nobody

log 127.0.0.1 local0 info

daemon

pidfile /var/run/haproxy.pid

defaults

mode tcp

retries 3

timeout connect 5s

timeout client 10s

timeout server 30s

timeout check 2s

frontend mycat-proxy

bind \*:8066

mode tcp

log global

use\_backend server\_mycat

backend server\_mycat

mode tcp

balance roundrobin

server mycat01 192.168.122.30:8066 weight 3 check inter 2000 rise 1 fall 2

server mycat02 192.168.122.40:8066 weight 3 check inter 2000 rise 1 fall 2

[root@ha\_master ~]# systemctl start haproxy

[root@ha\_master ~]# systemctl enable haproxy

Created symlink from /etc/systemd/system/multi-user.target.wants/haproxy.service to /usr/lib/systemd/system/haproxy.service.

[root@ha\_master ~]# netstat -antp | grep haproxy

tcp 0 0 0.0.0.0:8066 0.0.0.0:\* LISTEN 11600/haproxy

测试连接haproxy连接到mycat

[root@localhost ~]# mysql -u mycatuser -p -h 192.168.122.10 -P 8066

Enter password:

将haproxy配置拷贝到另一台机器

[root@ha\_master ~]# rsync -av /etc/haproxy/haproxy.cfg 192.168.122.20:/etc/haproxy/haproxy.cfg

[root@ha\_backup ~]# systemctl start haproxy

[root@ha\_backup ~]# systemctl enable haproxy

Created symlink from /etc/systemd/system/multi-user.target.wants/haproxy.service to /usr/lib/systemd/system/haproxy.service.

[root@ha\_backup ~]# netstat -antp | grep haproxy

tcp 0 0 0.0.0.0:8066 0.0.0.0:\* LISTEN 10814/haproxy

[root@localhost ~]# mysql -u mycatuser -p -h 192.168.122.20 -P 8066

Enter password:

6、安装配置keepalived实现Haproxy的高可用

[root@ha\_master ~]# yum install -y keepalived

[root@ha\_master ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id haproxy01

}

vrrp\_script check\_mycat\_service {

script "killall -0 haproxy"

interval 1

rise 2

fall 2

weight -90

}

vrrp\_instance mycat\_service {

state MASTER

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

track\_script {

check\_mycat\_service

}

virtual\_ipaddress {

192.168.122.100

}

}

[root@ha\_master ~]# rsync -av /etc/keepalived/keepalived.conf 192.168.122.20:/etc/keepalived/keepalived.conf

备调度器的keepalived配置文件：

[root@ha\_backup ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id haproxy02

}

vrrp\_script check\_mycat\_service {

script "killall -0 haproxy"

interval 1

rise 2

fall 2

weight -40

}

vrrp\_instance mycat\_service {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

track\_script {

check\_mycat\_service

}

virtual\_ipaddress {

192.168.122.100

}

}

分别启动keepalived服务

[root@ha\_master ~]# systemctl start keepalived

[root@ha\_master ~]# systemctl enable keepalived

测试：

1) 通过vip连接mycat

[root@localhost ~]# mysql -u mycatuser -p -h 192.168.122.100 -P 8066

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloundDB)

2) 测试vip的故障转移