**工作层次：**

Layer3：工作在三层（网络层）时，keepalived会定期向热备组中的服务器发送一个ICMP数据包，来判断某台服务器是否故障，如果故障则将这台服务器从热备组移除。

Layer4：工作在四层（传输层）时，keepalived以TCP端口的状态判断服务器是否故障，比如检测mysql 3306端口，如果故障则将这台服务器从热备组移除。

**示例：**

配置文件一：

global\_defs {

notification\_email {

example@163.com

}

notification\_email\_from example@example.com

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id MYSQL\_HA

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth1

virtual\_router\_id 50

nopreempt #当主down时，备接管，主恢复，不自动接管

priority 100

advert\_int 1

authentication {

auth\_type PASS

ahth\_pass 123

}

virtual\_ipaddress {

192.168.1.200 #虚拟IP地址

}

}

virtual\_server 192.168.1.200 3306 {

delay\_loop 6

# lb\_algo rr

# lb\_kind NAT

persistence\_timeout 50

protocol TCP

real\_server 192.168.1.201 3306 { #监控本机3306端口

weight 1

notify\_down /etc/keepalived/kill\_keepalived.sh #检测3306端口为down状态就执行此脚本（只有keepalived关闭，VIP才漂移 ）

TCP\_CHECK { #健康状态检测方式，可针对业务需求调整（TTP\_GET|SSL\_GET|TCP\_CHECK|SMTP\_CHECK|MISC\_CHECK）

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

Layer7：工作在七层时，keepalived根据用户设定的策略判断服务器上的程序是否正常运行，如果故障则将这台服务器从热备组移除。

示例：

! Configuration File for keepalived

global\_defs {

notification\_email {

example@163.com

}

notification\_email\_from example@example.com

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id MYSQL\_HA

}

vrrp\_script check\_nginx {

script /etc/keepalived/check\_nginx.sh #检测脚本

interval 2 #执行间隔时间

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth1

virtual\_router\_id 50

nopreempt #当主down时，备接管，主恢复，不自动接管

priority 100

advert\_int 1

authentication {

auth\_type PASS

ahth\_pass 123

}

virtual\_ipaddress {

192.168.1.200 #虚拟IP地址

}

track\_script { #在实例中引用脚本

check\_nginx

}

}

脚本内容如下：

# cat /etc/keepalived/check\_nginx.sh

Count1=`netstat -antp |grep -v grep |grep nginx |wc -l`

if [ $Count1 -eq 0 ]; then

/usr/local/nginx/sbin/nginx

sleep 2

Count2=`netstat -antp |grep -v grep |grep nginx |wc -l`

if [ $Count2 -eq 0 ]; then

service keepalived stop

else

exit 0

fi

else

exit 0

fi

4、健康状态检测方式

4.1 HTTP服务状态检测

HTTP\_GET或SSL\_GET {

url {

path /index.html #检测url，可写多个

digest 24326582a86bee478bac72d5af25089e #检测效验码

#digest效验码获取方法：genhash -s IP -p 80 -u http://IP/index.html

status\_code 200 #检测返回http状态码

}

connect\_port 80 #连接端口

connect\_timeout 3 #连接超时时间

nb\_get\_retry 3 #重试次数

delay\_before\_retry 2 #连接间隔时间

}

4.2 TCP端口状态检测（使用TCP端口服务基本上都可以使用）

TCP\_CHECK {

connect\_port 80 #健康检测端口，默认为real\_server后跟端口

connect\_timeout 5

nb\_get\_retry 3

delay\_before\_retry 3

}

4.3 邮件服务器SMTP检测

SMTP\_CHECK { #健康检测邮件服务器smtp

host {

connect\_ip

connect\_port

}

connect\_timeout 5

retry 2

delay\_before\_retry 3

hello\_name "mail.domain.com"

}

4.4 用户自定义脚本检测real\_server服务状态

MISC\_CHECK {

misc\_path /script.sh #指定外部程序或脚本位置

misc\_timeout 3 #执行脚本超时时间

!misc\_dynamic #不动态调整服务器权重（weight），如果启用将通过退出状态码动态调整real\_server权重值

}

5、状态转换通知功能

keepalived主配置邮件通知功能，默认当real\_server宕机或者恢复时才会发出邮件。

有时我们更想知道keepalived的主服务器故障切换后，VIP是否顺利漂移到备服务器，MySQL服务器是否正常？

那写个监控脚本吧，可以，但没必要，因为keepalived具备状态检测功能，所以我们直接使用就行了。

主配置默认邮件通知配置模板如下：

global\_defs # Block id

{

notification\_email # To:

{

admin@example1.com

...

}

# From: from address that will be in header

notification\_email\_from admin@example.com

smtp\_server 127.0.0.1 # IP

smtp\_connect\_timeout 30 # integer, seconds

router\_id my\_hostname # string identifying the machine,

# (doesn't have to be hostname).

enable\_traps # enable SNMP traps

}

5.1 实例状态通知

a) notify\_master ：节点变为master时执行

b) notify\_backup ： 节点变为backup时执行

c) notify\_fault ： 节点变为故障时执行

5.2 虚拟服务器检测通知

a) notify\_up ： 虚拟服务器up时执行

b) notify\_down ： 虚拟服务器down时执行

示例：

! Configuration File for keepalived

global\_defs {

notification\_email {

example@163.com

}

notification\_email\_from example@example.com

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id MYSQL\_HA

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth1

virtual\_router\_id 50

nopreempt #当主down时，备接管，主恢复，不自动接管

priority 100

advert\_int 1

authentication {

auth\_type PASS

ahth\_pass 123

}

virtual\_ipaddress {

192.168.1.200

}

notify\_master /etc/keepalived/to\_master.sh

notify\_backup /etc/keepalived/to\_backup.sh

notify\_fault /etc/keepalived/to\_fault.sh

}

virtual\_server 192.168.1.200 3306 {

delay\_loop 6

persistence\_timeout 50

protocol TCP

real\_server 192.168.1.201 3306 {

weight 1

notify\_up /etc/keepalived/mysql\_up.sh

notify\_down /etc/keepalived/mysql\_down.sh

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

状态参数后可以是bash命令，也可以是shell脚本，内容根据自己需求定义，以上示例中所涉及状态脚本如下：

1) 当服务器改变为主时执行此脚本

# cat to\_master.sh

#!/bin/bash

Date=$(date +%F" "%T)

IP=$(ifconfig eth0 |grep "inet addr" |cut -d":" -f2 |awk '{print $1}')

Mail="baojingtongzhi@163.com"

echo "$Date $IP change to master." |mail -s "Master-Backup Change Status" $Mail

2) 当服务器改变为备时执行此脚本

# cat to\_backup.sh

#!/bin/bash

Date=$(date +%F" "%T)

IP=$(ifconfig eth0 |grep "inet addr" |cut -d":" -f2 |awk '{print $1}')

Mail="baojingtongzhi@163.com"

echo "$Date $IP change to backup." |mail -s "Master-Backup Change Status" $Mail

3) 当服务器改变为故障时执行此脚本

# cat to\_fault.sh

#!/bin/bash

Date=$(date +%F" "%T)

IP=$(ifconfig eth0 |grep "inet addr" |cut -d":" -f2 |awk '{print $1}')

Mail="baojingtongzhi@163.com"

echo "$Date $IP change to fault." |mail -s "Master-Backup Change Status" $Mail

4) 当检测TCP端口3306为不可用时，执行此脚本，杀死keepalived，实现切换

# cat mysql\_down.sh

#!/bin/bash

Date=$(date +%F" "%T)

IP=$(ifconfig eth0 |grep "inet addr" |cut -d":" -f2 |awk '{print $1}')

Mail="baojingtongzhi@163.com"

pkill keepalived

echo "$Date $IP The mysql service failure,kill keepalived." |mail -s "Master-Backup MySQL Monitor" $Mail

5) 当检测TCP端口3306可用时，执行此脚本

# cat mysql\_up.sh

#!/bin/bash

Date=$(date +%F" "%T)

IP=$(ifconfig eth0 |grep "inet addr" |cut -d":" -f2 |awk '{print $1}')

Mail="baojingtongzhi@163.com"

echo "$Date $IP The mysql service is recovery." |mail -s "Master-Backup MySQL Monitor"