# Keepalived 安装

* 目的  
  通过Keepalived监控Mysql进程状态,实现Mysql cluster 自动切换功能,并通过修改相关配置开启keepalived的日志功能。

## 安装配置步骤

1. 安装  
   通过yum源安装   
   yum install keepalived -y

确保keepalived服务关闭

service keepalived stop

配置  
**修改/etc/keepalived/Keepalived.conf**

**Keepalived.conf**  Expand source

#! Configuration File for keepalived

global\_defs {

notification\_email {

dbagroup@yooli.com

}

notification\_email\_from dba@yooli.com

smtp\_server smtp.mail.yooli.com

smtp\_connect\_timeout 5

router\_id KP\_CORE\_HA #集群内相同的一个名称 #需要修改#

}

vrrp\_script check\_run {

script "/etc/keepalived/check\_mysql\_cluster.sh --vip=192.168.1.120"

interval 15 #15秒执行一次

timeout 10 #脚本执行超过10秒即为超时,触发keepalived ip切换动作

}

vrrp\_sync\_group VG1 {

group {

VI\_1

}

}

vrrp\_instance VI\_1 {

state backup

interface bond0

virtual\_router\_id 120 #需要修改#

priority 100 #主服务器值大于备服务器 #需要修改#

nopreempt #不主动重新取得IP地址

advert\_int 1

authentication { #验证类型与密码 集群内相同

auth\_type PASS

auth\_pass 1111

}

track\_script {

check\_run

}

virtual\_ipaddress {

192.168.1.120 #需要修改#

}

notify\_master "/etc/keepalived/keepalive\_notify.sh --stateinfo=Master" #状态触发后执行的脚本

notify\_backup "/etc/keepalived/keepalive\_notify.sh --stateinfo=Backup"

notify\_fault "/etc/keepalived/keepalive\_notify.sh --stateinfo=fault"

}

virtual\_server 192.168.1.120 3306{ #根据实际分配的IP进行设置 #需要修改#

delay\_loop 6

lb\_algo wcl

lb\_kind DR

persistence\_timeout 30

protocol TCP

real\_server 192.168.1.30 3306 { #根据实际IP和端口进行修改 #需要修改#

weight 200

TCP\_CHECK {

connect\_timeout 10 #(10秒无响应超时)

connect\_port 3306

}

}

}

**创建/etc/keepalived/check\_mysql\_cluster.sh**

**check\_mysql\_cluster.sh**  Expand source

#!/bin/bash

clear

date=`date "+%Y-%m-%d %H:%M:%S"`

hostname=`hostname|sed 's/\./\\_/g'`

myconf="/etc/my.cnf"

MYSQL\_HOST=localhost

MYSQL\_USER=sstuser

MYSQL\_PASSWORD=sstuser

emailsuject="Keepalived check"

emailbody="/tmp/Keepalived\_check\_"`date "+%Y%m%d%H%M%S"`".log"

CHECK\_TIMES=5

log\_info\_msg()

{

local logtime=`date "+%Y-%m-%d %H:%M:%S"`

echo -e "\\033[44;37m$logtime\t[INFO]\t$@\033[0m"

echo -e "$logtime\t[INFO]\t$@" >> $emailbody

}

log\_success\_msg()

{

local logtime=`date "+%Y-%m-%d %H:%M:%S"`

echo -e "\033[46;37m$logtime\t[SUCCESS]\t$@\033[0m"

echo -e "$logtime\t[SUCCESS]\t$@" >> $emailbody

}

log\_failure\_msg()

{

local logtime=`date "+%Y-%m-%d %H:%M:%S"`

echo -e "\033[41;37m$logtime\t[ERROR]\t$@\033[0m"

echo -e "$logtime\t[ERROR]\t$@" >> $emailbody

}

script\_quit()

{

local logtime=`date "+%Y-%m-%d %H:%M:%S"`

echo -e "\033[41;37m$logtime\t[script\_quit]\tError num:${@:1} \033[0m"

echo -e "$logtime\t[script\_quit]\tError num:${@:1}" >> $emailbody

# Send Email

if [ ${@:1} -ne 0 ]

then

keepalive\_stop

log\_info\_msg "(cat $emailbody | mail -s "${emailsuject}" dbagroup@yooli.com)."

cat $emailbody | mail -s "Error!! ${emailsuject} ${Vip}" dbagroup@yooli.com

fi

# Directory cleaning

rm $emailbody -f

exit ${@:1}

}

# Get the MySQL configuration information

get\_mysql\_config() {

local farg=

for farg

do

case "$farg" in

--instance=\*) instanceID=`echo "$farg" | sed -e 's/^[^=]\*=//'`

;;

--defaults-file=\*) myconf=`echo "$farg" | sed -e 's/^[^=]\*=//'`

;;

esac

done

which "my\_print\_defaults" >/dev/null 2>&1

if [ $? -ne 0 ]

then

log\_failure\_msg "The program 'my\_print\_defaults' not found !"

script\_quit 1

fi

local mysqlarg=

for mysqlarg in `my\_print\_defaults $myconf $instanceID`

do

case "$mysqlarg" in

--datadir=\*) mysqld\_datadir=`echo "$mysqlarg" | sed -e 's/^[^=]\*=//'`

if [ ! -d "$mysqld\_datadir" ]

then

log\_failure\_msg "Unable to find the MySQL data directory."

script\_quit 1

fi

log\_info\_msg "The MySQL data directory path is $mysqld\_datadir."

;;

--port=\*) MYSQL\_PORT=`echo "$mysqlarg" | sed -e 's/^[^=]\*=//'`

if [ -z "$MYSQL\_PORT" ]

then

log\_failure\_msg "Unable to find the MySQL port."

script\_quit 1

fi

log\_info\_msg "The MySQL port is $MYSQL\_PORT."

;;

--socket=\*) MYSQL\_SOCKET=`echo "$mysqlarg" | sed -e 's/^[^=]\*=//'`

;;

esac

done

if [ -z "$mysqlarg" ]

then

log\_failure\_msg "Get MySQL configuration is failure !\n(my\_print\_defaults $myconf $instanceID)"

script\_quit 1

fi

if [ ! -S "$MYSQL\_SOCKET" ]

then

MYSQL\_SOCKET=$mysqld\_datadir/`hostname`.sock

else

case "$MYSQL\_SOCKET" in

/\* ) ;;

\* ) MYSQL\_SOCKET="$mysqld\_datadir/$MYSQL\_SOCKET" ;;

esac

fi

if [ -S "$MYSQL\_SOCKET" ]

then

log\_info\_msg "The MySQL socket is $MYSQL\_SOCKET."

else

log\_failure\_msg "Unable to find the MySQL Socket"

script\_quit 1

fi

return 0

}

function keepalive\_stop() {

local keepalivedpath=`find /etc/init.d/ -regex '.\*keepalive[d]?'`

log\_info\_msg "Get keepalived Service file. $keepalivedpath"

if [ -z "$keepalivedpath" ]

then

log\_failure\_msg "Could not find the keepalived Service file !"

script\_quit 1

fi

local num=0

while [ ${num} -le ${CHECK\_TIMES} ]

do

${keepalivedpath} stop >/dev/null 2>&1

if [ $? -eq 0 ]

then

log\_success\_msg "Keepalived service stop successfully."

return

fi

sleep 5

num=`expr $num + 1`

log\_info\_msg "Delay check to wait for $fdelaynum seconds."

done

return

}

function check\_mysql\_health (){

local fmysqladminpath=`which "mysqladmin"`

if [ -z "$fmysqladminpath" ]

then

log\_failure\_msg "The program 'mysqladmin' not found !"

script\_quit 1

fi

local num=1

while [ ${num} -le ${CHECK\_TIMES} ]

do

if [ -z "$MYSQL\_PASSWORD" ]

then

local mysqlstatus=`${fmysqladminpath} --host=${MYSQL\_HOST} --socket=${MYSQL\_SOCKET} --user=${MYSQL\_USER} extended-status | awk -F "|" '/wsrep\_cluster\_status|wsrep\_connected|wsrep\_ready/{gsub(/[[:blank:]]\*/,"",$0);print "--"$2"="$3}'`

else

local mysqlstatus=`${fmysqladminpath} --host=${MYSQL\_HOST} --socket=${MYSQL\_SOCKET} --user=${MYSQL\_USER} --password=${MYSQL\_PASSWORD} extended-status | awk -F "|" '/wsrep\_cluster\_status|wsrep\_connected|wsrep\_ready/{gsub(/[[:blank:]]\*/,"",$0);print "--"$2"="$3}'`

fi

local s=

for s in ${mysqlstatus}

do

case "$s" in

--wsrep\_cluster\_status=\*) local wsrep\_cluster\_status=`echo "$s" | sed -e 's/^[^=]\*=//'`

;;

--wsrep\_connected=\*) local wsrep\_connected=`echo "$s" | sed -e 's/^[^=]\*=//'`

;;

--wsrep\_ready=\*) local wsrep\_ready=`echo "$s" | sed -e 's/^[^=]\*=//'`

;;

esac

done

if [ \( "${wsrep\_cluster\_status}" = "Primary" \) -a \( "${wsrep\_connected}" = "ON" \) -a \( "${wsrep\_connected}" = "ON" \) ]

then

log\_success\_msg "Mysql cluster service is health."

return

else

log\_info\_msg "MySQL Cluster health check for abnormal, began to try again.num is ${num}"

num=`expr $num + 1`

sleep ${CHECK\_TIMES}

fi

done

if [ ! ${num} -lt ${CHECK\_TIMES} ]

then

log\_failure\_msg "Mysql cluster service is not health."

keepalive\_stop

fi

return

}

clear

#Main

for mainarg in "$@"

do

case "$mainarg" in

--vip=\*) Vip=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

log\_info\_msg "The Virtual IP is ($Vip)."

;;

--defaults-file=\*) myconf=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

-host=\*) MYSQL\_HOST=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

-user=\*) MYSQL\_USER=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

-password=\*) MYSQL\_PASSWORD=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

mysqld\*) instanceID=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

log\_info\_msg "Set Mysql instance to ($instanceID)."

;;

esac

done

if [ -z "$Vip" ]

then

log\_failure\_msg "Virtual IP is not set !"

script\_quit 1

else

log\_info\_msg "Virtual IP is $Vip !"

fi

if [ ! -r "$myconf" ]

then

log\_failure\_msg "Unable to find the MySQL configuration file.\n(${myconf}) !"

script\_quit 1

fi

if [ -z "$instanceID" ]

then

instanceID=`cat $myconf | grep -P '\[mysqld[0-9]?\]' | head -1 | sed -r 's/\[|\]//g'`

log\_info\_msg "The instance name is not set.So Mysql instance to default value (${instanceID})."

fi

log\_info\_msg "Get MySQL configuration parameters."

get\_mysql\_config --instance=$instanceID --defaults-file=$myconf

log\_info\_msg "Check MySQL health."

check\_mysql\_health --instance=$instanceID --defaults-file=$myconf

script\_quit 0

**创建keepalived/keepalive\_notify.sh**

**keepalive\_notify.sh**  Expand source

#!/bin/bash

clear

date=`date "+%Y-%m-%d %H:%M:%S"`

hostname=`hostname|sed 's/\./\\_/g'`

ipaddress=`ifconfig | grep "inet addr"|cut -d ":" -f 2|cut -d " " -f 1|head -1`

emailsuject="Keepalived notify "

emaillog="/var/log/keepalived\_notify\_"`date "+%Y%m%d%H%M%S"`".log"

clear

#Main

for mainarg in "$@"

do

case "$mainarg" in

--mailto=\*) mailto=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

--stateinfo=\*) stateinfo=`echo "$mainarg" | sed -e 's/^[^=]\*=//'`

;;

esac

done

[ "${mailto}" = '' ] && mailto="dbagroup@yooli.com"

keepstate=`pidof keepalived`

if [ -z "$keepstate" ]

then

keepresult=`/etc/init.d/keepalived start`

else

keepresult=`/etc/init.d/keepalived status`

fi

echo -e "HostName:\t${hostname}\ndatetime:\t${date}\nIP Address:\t${ipaddress}\nKeepalived State:\t${stateinfo}\nKeepalievd Run Status:\t${keepresult}" | mail -s "${emailsuject}\_${hostname}\_${stateinfo}" $mailto

配置smtp邮箱  
**YUM install mailx -y**  
**修改/etc/mail.rc**

**mail.rc**  Expand source

cat << EOF >> /etc/mail.rc

set nss-config-dir=/etc/pki/nssdb

set from="dba@yooli.com"

set smtp=smtps://smtp.exmail.qq.com:465

set smtp-auth=login

set smtp-auth-user=dba@yooli.com

set smtp-auth-password="urpass"

set ssl-verify=ignore

EOF

1. 配置keepalived 日志

**修改 /etc/sysconfig/keepalived**

**sysconfig/keepalived**  Expand source

sed '/KEEPALIVED\_OPTIONS="-D"/s/"-D"/"-D -d -S 0"/' /etc/sysconfig/keepalived

**修改/etc/rsyslog.conf**

**rsyslog.conf**  Expand source

cat << EOF >> /etc/rsyslog.conf

# keepalived -S 0

local0.\* /var/log/keepalived.log

EOF

1. 启动

Service rsyslog restart

Service keepalived start

## 相关的文章

主页:

[DBA 团队](/pages/viewpage.action?pageId=1540353)

页面:

[Keepalived 安装](/pages/viewpage.action?pageId=1540575)

Icon