# redis主备部署方案

**Redis部署方式采用主备的方式，通过keepalived来对外提供虚IP，并实现主备自动切换功能。**

**主实例A**：192.168.20.30

**备实例B**：192.168.20.232

**虚IP**：192.168.20.110

正常工作时，虚IP在主实例A上，主实例A上的数据自动同步到备实例B上，当主实例A挂掉之后，备实例B将自动接管虚IP，并将redis转换为主模式，待原主实例A恢复后，A将自动切换成备模式，从B上同步数据，主备角色互换，实现融灾备份。

**安装部署步骤如下：**

1. **安装keepalived**

wget http://www.keepalived.org/software/keepalived-1.2.6.tar.gz

tar zxvf keepalived-1.2.6.tar.gz

cd keepalived-1.2.6

./configure

如果报错

configure: error:

!!! OpenSSL is not properly installed on your system. !!!

!!! Can not include OpenSSL headers files.

解决办法：

yum -y install openssl-devel

yum -y install popt-devel

ln -s /usr/src/kernels/2.6.32-220.el6.x86\_64/ /usr/src/linux

./configure

make

make install

cp /usr/local/etc/rc.d/init.d/keepalived /etc/rc.d/init.d/

cp /usr/local/etc/sysconfig/keepalived /etc/sysconfig/

cp /usr/local/sbin/keepalived /usr/sbin/

mkdir /etc/keepalived

**添加keepalived的配置项：**

vi /etc/keepalived/keepalived.conf

global\_defs {

router\_id LVS\_DEVEL

}

vrrp\_script Monitor\_Redis {

script "/home/bbcv/redis/redis\_keepalive.sh"

interval 3 #每3秒执行一次

weight 2

}

# VIP1

vrrp\_instance VI\_1 {

state BACKUP #主备服务器都设置成BACKUP

interface eth0

virtual\_router\_id 55

priority 100 # 备份服务上将100改为90

advert\_int 1 #检查间隔

nopreempt #设置为不抢占，注意这个配置只能设置在state为BACKUP的主机上，而且这个主机的priority必须比另外一台高

authentication {

auth\_type PASS

auth\_pass 1111

}

notify\_master /home/bbcv/redis/redis\_master.sh

notify\_backup /home/bbcv/redis/redis\_backup.sh

track\_script {

Monitor\_Redis #(调用redis进程检测脚本)

}

virtual\_ipaddress {

192.168.20.110

#(如果有多个VIP，继续换行填写.)

}

}

**启动keepalived**

service keepalived start

1. **redis\_keepalive.sh脚本**

该脚本主要实现对redis进程进行监控，当检测到redis进程挂掉时，自动停止keepalived进程，使虚IP进行漂移

* 注意添加该脚本的可执行权限！

chmod +x redis\_keepalive.sh

#!/bin/bash

time=$(date '+%Y-%m-%d %H:%M:%S' )

#redis部署路径

redispath=/home/bbcv/redis

#redis-cli部署路径

rediscli=$redispath/redis-2.0.0-rc4/redis-cli

if [ ! -d "logs" ]; then

mkdir $redispath/logs

fi

logfile=$redispath/logs/redis-state.log

oldfile=$logfile'.'$(date +%Y-%m-%d --date='30 days ago')

yesterdayfile=$logfile'.'$(date +%Y-%m-%d --date='1 days ago')

#把昨天的日志重命名

if [ -f $yesterdayfile ]

then

echo "OK"

else

mv $logfile $yesterdayfile >> $logfile 2>&1

echo "$time [$yesterdayfile] Move yesterdayfile Success!" >> $logfile

fi

#删除老的日志文件

if [ -f $oldfile ]

then

rm -f $oldfile >> $logfile 2>&1

echo "$time [$oldfile] Delete Old File Success!" >> $logfile

else

echo "no old file"

fi

#检测redis端口是否正常

status=`$rediscli -h 127.0.0.1 -p 6379 info|grep role|awk -F ":" '{print $2}'`

$rediscli -h 127.0.0.1 -p 6379 info > /dev/null

if [ $? -eq 0 ]; then

echo "$time redis server is OK" >>$logfile

echo 目前状态为:$status >>$logfile

else

echo "$time no redis service found!" >>$logfile

sleep 2

# try to found it again

$rediscli -h 127.0.0.1 -p 6379 info > /dev/null

if [ $? -eq 0 ]; then

echo "$time redis server is OK" >>$logfile

echo 目前状态为:$status >>$logfile

#exit 0

else

echo "$time redis server error" >>$logfile

#stop keepalived

/etc/init.d/keepalived stop

echo "$time stop keepalived" >>$logfile

fi

fi

1. **redis\_master.sh脚本**

该脚本主要实现当keepalived切换到master时，执行slaveof no one语句把redis的端口转换为主模式

* 注意添加该脚本的可执行权限！

chmod +x redis\_master.sh

#!/bin/sh

#本机IP地址

localip=127.0.0.1

#另一台服务器的地址

backip=192.168.20.232

#redis端口号

port="6379 9998 11111 14948"

time=$(date '+%Y-%m-%d %H:%M:%S' )

#redis部署路径

redispath=/home/bbcv/redis

#redis-cli部署路径

rediscli=$redispath/redis-2.0.0-rc4/redis-cli

#日志文件目录

logfile=$redispath/logs/redis-state.log

for p in $port

do

{

echo "$time redis $p端口切换成主端口" >>$logfile

$rediscli -h $localip -p $p slaveof no one >>$logfile 2>&1

sleep 1

}

done

1. **redis\_backup.sh**

该脚本主要实现当keepalived切换到master时，执行slaveof 主redisIP 端口 语句，把redis的端口转换为备模式，并同步主redis的数据

* 注意添加该脚本的可执行权限！

chmod +x redis\_backup.sh

#!/bin/sh

#本机IP地址

localip=127.0.0.1

#另一台服务器的地址

backip=192.168.20.232

#redis端口号

port="6379 9998 11111 14948"

time=$(date '+%Y-%m-%d %H:%M:%S' )

#redis部署路径

redispath=/home/bbcv/redis

#redis-cli部署路径

rediscli=$redispath/redis-2.0.0-rc4/redis-cli

#日志文件目录

logfile=$redispath/logs/redis-state.log

for p in $port

do

{

echo "$time redis $p端口切换成$backip的备端口" >>$logfile

$rediscli -h $localip -p $p slaveof $backip $p >>$logfile 2>&1

sleep 1

}

done

1. **安装redis**

1.新建文件夹，如mkdir redis

2.将redis-2.2.13.tar.gz拷入其中

3.解压并进入主目录

4.make

5.make install（Ubuntu上需要sudo执行）

6.修改配置文件redis.conf

修改如下：

daemonize no 改为 daemonize yes

取消注释：

syslog-enabled no （改为syslog-enabled yes ）

syslog-facility local0

已将修改后的redis.conf文件放入附件，不同需求还需再修改配置文件（如修改端口等）

7.在执行redis-server redis.conf启动时，可能会报如下错误：

Can't chdir to '/var/db/redis': No such file or directory

需要手动创建'/var/db/redis'文件夹

# WARNING overcommit\_memory is set to 0! Background save may fail under low memory condition

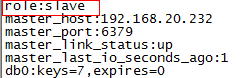
需要执行 sysctl vm.overcommit\_memory=1 修改内核参数

8.再次执行redis-server redis.conf可成功启动redis

1. **测试**

1.查看当前redis是主还是备

/home/bbcv/redis/redis-2.0.0-rc4/redis-cli -h 127.0.0.1 -p 6379 info



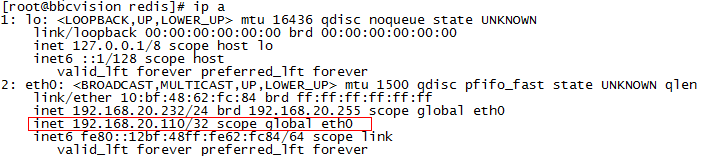
2.在主redisA上新增一条数据，查看备redisB是否同步该数据

/home/bbcv/redis/redis-2.0.0-rc4/redis-cli -h 127.0.0.1 -p 6379



3.停止主redisA，查看虚IP是否切换到备redisB服务器，并查看备redisB服务器是否切换成主状态

ip a查看虚IP是否在本机上



4.恢复redisA，查看redisA是否从redisB中同步数据