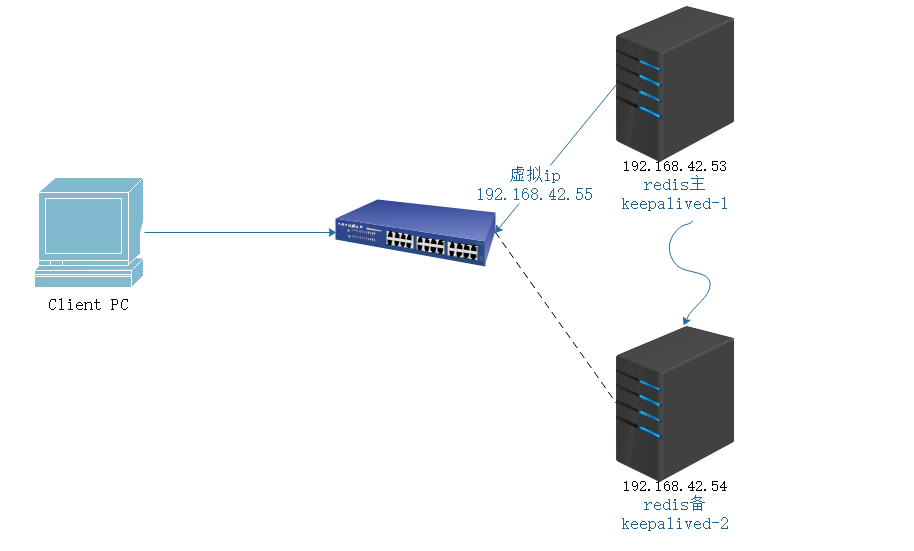
## Redis主从配置及keepalived实现Redis高可用

### 架构图：



### 环境说明

**系统环境和软件版本号：**

|  |  |
| --- | --- |
| 项目 | 参数 |
| 系统版本 | Redhat 7.2 64位 |
| 内核版本 | jdk 1.8.0\_111 64位 |
| Redis版本 | redis-4.0.0 |
| Keepalived版本 | Keepalived v1.3.5 |

### 主机环境配置

#### 主机名解析

* 在各主机/etc/hosts中配置主机名解析，两台均执行：

#vim /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.42.53 osb53

192.168.42.54 osb54

#### 安全策略

* 两台主机均selinux/iptables设置:

sed -i 's/SELINUX=.\*/SELINUX=disabled/' /etc/selinux/config

setenforce 0

systemctl stop firewalld

systemctl disable firewalld

### 安装redis keepalived

#### yum 安装

* 本地有配置redis源两台主机直接yum安装即可

yum -y install redis keepalived

#### 二进制包安装

* 两台主机下载目前redis最稳定版本

wget http://download.redis.io/releases/redis-4.0.0.tar.gz

* 两台主机下载keepalived二进制安装包

wget http://www.keepalived.org/software/keepalived-1.3.5.tar.gz

#### 编译安装redis

* **编译安装redis：**

#两台主机执行：

tar axf /opt/redis-4.0.0.tar.gz

cd redis-4.0.0

make && make install

* **编译安装keepalived**

##两台主机执行：

tar zxvf keepalived-1.3.5.tar.gz

./configure --prefix=/usr/local/keepalived

make && make install

### 配置redis主从

#### 配置redis

* 复制redis配置文件到/etc目录：

cp /opt/redis-4.0.0/redis.conf /etc/redis

* 修改配置文件使redis-server可以在后台启动

sed -i s/daemonize no/daemonize yes/ /etc/redis.conf

* Init.d 启动脚本：

vim /etc/init.d/redis-server

#!/usr/bin/env bash

#redis start up the redis server daemon

#chkconfig: 345 99 99

#description: redis service in /etc/init.d/redis

#chkconfig --add redis or chkconfig --list redis

# service redis start or service redis stop

#processname: redis-server

#config: /etc/redis.conf

PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379

EXEC=/usr/local/bin/redis-server

REDIS\_CLI=/usr/local/bin/redis-cli

PIDFILE=/var/run/redis.pid

CONF="/etc/redis.conf"

#make sure some dir exist

if [ ! -d /var/lib/redis ] ;then

mkdir -p /var/lib/redis

mkdir -p /var/log/redis

fi

case "$1" in

status)

ps -A|grep redis

;;

start)

if [ -f $PIDFILE ]

then

echo "$PIDFILE exists, process is already running or crashed"

else

echo "Starting Redis server..."

${EXEC} ${CONF}

fi

if [ "$?"="0" ]

then

echo "Redis is running..."

fi

;;

stop)

if [ ! -f $PIDFILE ]

then

echo "$PIDFILE does not exist, process is not running"

else

PID=$(cat $PIDFILE)

echo "Stopping ..."

$REDIS\_CLI -p $REDISPORT SHUTDOWN

while [ -x ${PIDFILE} ]

do

echo "Waiting for Redis to shutdown ..."

sleep 1

done

echo "Redis stopped"

fi

;;

restart|force-reload)

${0} stop

${0} start

;;

\*)

echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

exit 1

esac

* 赋权启动：

chmod 755 /etc/init.d/redis-server

chkconfig --add redis-server

service redis-server start/stop/restart

#### 从服务器配置Redis

* 从服务器配置一样，只不过更改/etx/redis.cof中slaveof修改为：

slaveof 192.168.42.54 6379

* 重启配置生效

/etc/init.d/redis-server restart

#### 测试Redis主从复制

#主服务器

redis-cli -p 6379 set hello word

OK

#从服务器

redis-cli -p 6379 get hello

"word"

#主服务器

redis-cli -p 6379 set hello2 word2

OK

#从服务器

redis-cli -p 6379 get hello2

"word2"

redis-cli -p 6379 set hello world

(error) READONLY You can't write against a read only slave.

#成功配置主从redi服务器，又要配置中有一条从服务器是只读的，所以从服务器没法设置数据，只可以读取数据。

### 配置keepalived高可用

#### 配置keepalived

* 复制相应配置文件

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

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

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

mkdir -p /etc/keepalived

cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived

#### 修改配置文件

* **Master192.168.42.53上配置文件：**

#vim /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id redis1

}

vrrp\_script chk\_redis {

script "/etc/keepalived/scripts/redis\_check.sh"

interval 2

}

vrrp\_instance Redis {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

nopreempt

unicast\_src\_ip 192.168.42.53

unicast\_peer {

192.168.42.54

}

authentication {

auth\_type PASS

auth\_pass 112233

}

track\_script {

chk\_redis

}

virtual\_ipaddress {

192.168.42.55

}

notify\_master /etc/keepalived/scripts/redis\_master.sh

notify\_backup /etc/keepalived/scripts/redis\_backup.sh

notify\_fault /etc/keepalived/scripts/redis\_fault.sh

notify\_stop /etc/keepalived/scripts/redis\_stop.sh

}

* **Slave 192.168.42.54上配置文件：**

#vim /etc/keepalived/kepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id redis2

}

vrrp\_script chk\_redis {

script "/etc/keepalived/scripts/redis\_check.sh"

interval 2

}

vrrp\_instance Redis {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

# nopreempt

unicast\_src\_ip 192.168.42.54

unicast\_peer {

192.168.42.53

}

authentication {

auth\_type PASS

auth\_pass 112233

}

track\_script {

chk\_redis

}

virtual\_ipaddress {

192.168.42.55

}

notify\_master /etc/keepalived/scripts/redis\_master.sh

notify\_backup /etc/keepalived/scripts/redis\_backup.sh

notify\_fault /etc/keepalived/scripts/redis\_fault.sh

notify\_stop /etc/keepalived/scripts/redis\_stop.sh

}

### 创建redis监控脚本

* 创建存放脚本目录：

mkdir -pv /etc/keepalived/scripts

#### 编写状态监控脚本

* 在master/slave创建redis状态监控脚本

vim /etc/keepalived/scripts/redis\_check.sh

#!/bin/bash

ALIVE=`/usr/local/bin/redis-cli PING`

if [ "$ALIVE" == "PONG" ]

then

echo $ALIVE

exit 0

else

echo $ALIVE

exit 1

fi

#### 编写关键脚本

ll /etc/keepalived/scripts/

-rwxr-xr-x 1 root root 891 10月 18 16:07 keepalived-redis-state.log

-rwxr-xr-x 1 755 root 457 10月 17 16:59 redis\_backup.sh

-rwxr-xr-x 1 755 root 182 10月 17 16:41 redis\_fault.sh

-rwxr-xr-x 1 755 root 580 10月 18 09:19 redis\_master.sh

-rwxr-xr-x 1 755 root 162 10月 17 16:46 redis\_stop.sh

* Keepalived在转换状态时会依照状态来执行脚本：
* 当进入master状态时会执行redis\_master.sh
* 当进入backup状态时会执行redis\_backup.sh
* 当发现异常情况时会执行redis\_fault.sh
* 当进入当程序终止时则会执行redis\_stop.sh
* redis\_master.sh脚本

vim redis\_master.sh

#!/bin/bash

#当进入master状态会呼叫notify\_master

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[master]" >> $log\_file

date >> $log\_file

echo "Being master...." >>$log\_file 2>&1

echo "Run SLAVEOF cmd....." >>$log\_file

######设置本机53为54的从机

$redis\_cli SLAVEOF 192.168.42.54 6379 >> $log\_file 2>&1q

#延迟10秒以后待数据同步完成后再取消同步状态

sleep 10

echo "Run SLAVEOF NO ONE cmd...." >>$log\_file

######设置为本地为Redis的主机

$redis\_cli SLAVEOF NO ONE >> $log\_file 2>&1

* redis\_backup.sh

vim redis\_backup.sh

#!/bin/bash

#当进入BACKUP状态会呼叫notify\_backup

redis\_cli="/usr/local/bin/redis-cli"

log\_file="/etc/keepalived/scripts/keepalived-redis-state.log"

echo "[backup]" >>$log\_file

date >> $log\_file

echo "Being slave......" >> $log\_file 2>&1

#延迟15秒待数据被对方同步完成之后再切换主从角色

sleep 15

echo "Run SLAVEOF cmd...." >>$log\_file

######设置本机53为54的从机

$redis\_cli SLAVEOF 192.168.42.54 6379 >>log\_file 2>&1

* redis\_fault.sh

vim redis\_fault.sh

#!/bin/bash

#当发现异常情况时进入Fault状态呼叫notify\_fault

log\_file=/etc/keepalived/scripts/keepalived-redis-state.log

echo "[fault]" >>$log\_file

date >> $log\_file

* redis\_stop.sh

vim redis\_stop.sh

#!/bin/bash

#当程序终止时，则呼叫notify\_stop

log\_file=/etc/keepalived/scripts/keepalived-redis-state.log

echo "[stop]" >>$log\_file

date >>$log\_file