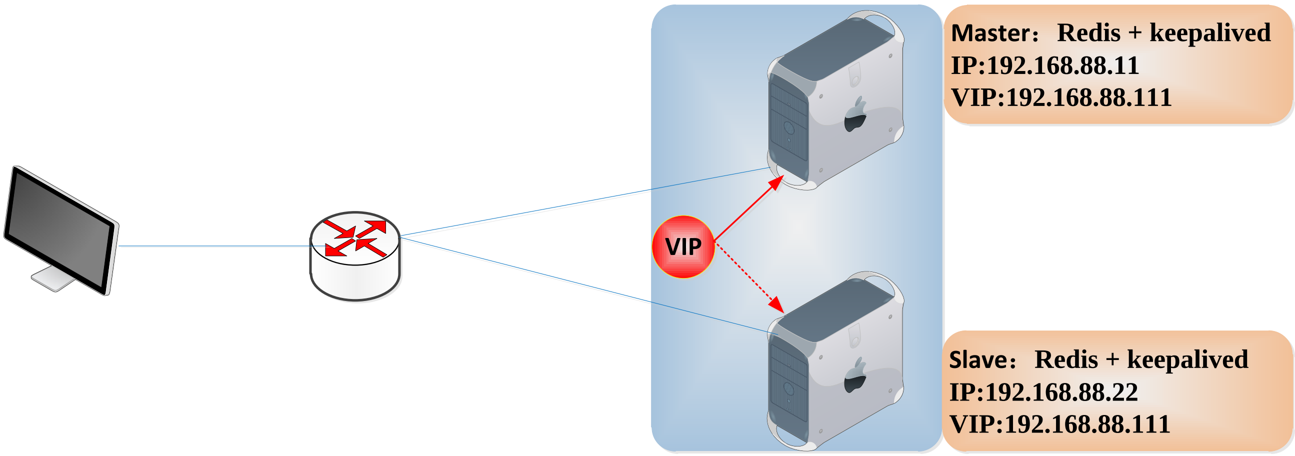
Redis + Keepalived—高可用

1、环境简介：

VS1：CentOS 6.8+Keepalived v1.2.13(**Master**)　+　Redis server v=3.0.6　+　eth0:192.168.88.11

VS2：CentOS 6.8+Keepalived v1.2.13(**Slave**)　+　Redis server v=3.0.6　+　eth0:192.168.88.22

2、网络拓扑：



3、实验前准备：

注：需配置好本地 yum 源，以下操作需在各主机上进行

(1)修改各主机的/etc/hosts 文件，实现主机名解析：

```
[root@vs1 ~]#cat /etc/hosts
192.168.88.11    vs1.maochen.com  vs1
192.168.88.22    vs2.maochen.com  vs2
```

(2)添加主机秘钥，实现各主机无秘钥登录：

```
[root@vs1 ~]#cat key.sh
#/bin/bash
HOST=(‘vs1’ ‘vs2’)
for i in {0..1}
do
        NAME=${HOST[$i]}
        [ -e /root/.ssh/id_rsa.pub ] || ssh-keygen -f /root/.ssh/id_rsa -P “”
        ssh-copy-id -i /root/.ssh/id_rsa.pub root@${NAME}
        scp /etc/hosts ${NAME}:/etc/
done
[root@vs1 ~]#sh key.sh
```

(3)确保各个主机的时间同步(不一定准时，但必须相同)：

```
[root@vs1 ~]#ntpdate time.nist.gov
[root@vs2 ~]#ntpdate time.nist.gov
 [root@vs1 ~]#date;ssh vs2 “date”          #检测时间是否同步
```

(4)确保 iptables 和 selinux 关闭：

```
[root@vs1 ~]#service iptables start
[root@vs1 ~]#sed -i ‘s/SELINUX=.*/SELINUX=disabled/g’ /etc/sysconfig/selinux
[root@vs1 ~]#setenforce 0 ##临时关闭 selinux，上一条为永久关闭
```

(5)安装 redis、keepalived 软件(一下步骤需在两个节点上安装)

```
[root@vs1 ~]#yum install vim gcc telnet wget lrzsz openssl openssl-devel openssl-clients ntp
date  -y
#-------------------------------源码编译安装 keepalived----------------------------#
[root@vs1 ~]# wget http://www.keepalived.org/software/keepalived-1.2.20.tar.gz
[root@vs1 ~]# tar -xf keepalived-1.2.20.tar.gz
[root@vs1 ~]# cd keepalived-1.2.20
[root@vs1 keepalived-1.2.20]# ./configure --prefix=/usr/local/keepalived
[root@vs1 keepalived-1.2.20]# make && make install
[root@vs1 ~]# cp /usr/local/keepalived/sbin/keepalived  /usr/sbin/
[root@vs1 ~]# cp /usr/local/keepalived/etc/sysconfig/keepalived  /etc/sysconfig/
[root@vs1 ~]# cp  /usr/local/keepalived/etc/rc.d/init.d/keepalived  /etc/init.d/
[root@vs1  ~]#  cp /usr/local/keepalived/etc/keepalived/keepalived.conf  /etc/keepalived/
[root@vs1 ~]# mkdir -p /etc/keepalived/log/
[root@vs1 ~]# mkdir -p /etc/keepalived/scripts/
添加日志
[root@vs1 ~]# vim /etc/sysconfig/keepalived
KEEPALIVED_OPTIONS="-D"
修改为
KEEPALIVED_OPTIONS="-D -d -S 0"
[root@vs1 ~]# vi /etc/rsyslog.conf          #加入如下配置
#keepalived -S 0
local0.* /var/log/keepalived.log
[root@vs1 ~]# /etc/init.d/rsyslog restart  #重启日志服务
[root@vs1 ~]# ll /var/log/keepalived.log
-rw-------. 1 root root 0 Jun 21 19:46 /var/log/keepalived.log
```

#-------------------------------源码编译安装 redis----------------------------------#

```
[root@vs1 ~]# wget http://download.redis.io/releases/redis-3.2.0.tar.gz
[root@vs1 ~]# tar xf redis-3.2.0.tar.gz
[root@vs1 ~]# mkdir -p /usr/local/redis
[root@vs1 ~]# mv redis-3.2.0/*  /usr/local/redis
[root@vs1 redis]# cd /usr/local/redis
[root@vs1 redis]# make
[root@vs1 src]# cd src && make install
[root@vs1 src]# cd /usr/local/redis
[root@vs1 redis]# cp redis.conf   /etc/
[root@vs1 redis]# mkdir  -p  /redis/log      #日志目录
[root@vs1 redis]# mkdir  -p   /redis/run     #pid 文件目录
[root@vs1 redis]# mkdir  -p   /redis/data    #本地快照数据库存放目录
[root@vs1 redis]# vi /etc/redis.conf         #编辑
daemonize yes  #设置后台启动 redis
[root@vs1 redis]# sysctl vm.overcommit_memory=1
[root@vs1 redis]# echo never > /sys/kernel/mm/transparent_hugepage/enabled
[root@vs1 redis]# vi /etc/sysctl.conf        #编辑，在最后一行添加下面代码
vm.overcommit_memory = 1
[root@vs1 redis]# sysctl -p #使设置立即生效
```

#----------------------------设置开机自启动 redis----------------------------------#

```
[root@vs1 redis]# vim /etc/init.d/redis   #编辑，添加以下代码
#!/bin/sh
# chkconfig:   2345 90 10
# description:  Redis is a persistent key-value database
# redis    Startup script for redis processe
# processname: redis
redis_path="/usr/local/bin/redis-server"
redis_conf="/etc/redis.conf"
redis_pid="/redis/run/redis.pid"
# Source function library.
. /etc/rc.d/init.d/functions
[ -x $redis_path ] || exit 0
RETVAL=0
prog="redis"
# Start daemons.
start() {
```

```
if [ -e $redis_pid -a ! -z $redis_pid ];then
echo $prog" already running...."
exit 1
fi
echo -n $"Starting $prog "
# Single instance for all caches
$redis_path $redis_conf
RETVAL=$?
[ $RETVAL -eq 0 ] && {
touch /var/lock/subsys/$prog
success $"$prog"
}
echo
return $RETVAL
}
# Stop daemons.
stop() {
echo -n $"Stopping $prog "
killproc -d 10 $redis_path
echo
[ $RETVAL = 0 ] && rm -f $redis_pid /var/lock/subsys/$prog
RETVAL=$?
return $RETVAL
}
# See how we were called.
case "$1" in
start)
start;;
stop)
stop;;
status)
status $prog
RETVAL=$?;;
restart)
stop
start;;
condrestart)
if test "x`pidof redis`" != x; then
stop
```

```
start
fi;;
*)
echo $"Usage: $0 {start|stop|status|restart|condrestart}"
exit 1
esac
exit $RETVAL
[root@vs1 redis]# chmod 755 /etc/init.d/redis                 #添加脚本执行权限
[root@vs1 redis]# chkconfig --add redis                       #添加开启启动
[root@vs1  redis]# chkconfig --level 2345 redis on            #设置启动级别
[root@vs1 redis]# chkconfig --list redis                      #查看启动级别
[root@vs1 redis]# service redis restart                       #重新启动 redis
```

然后重启 redis，现在 redis 都配置了双主，但是他们现在都不可以写入，必须要配置 keepalived，有 redis 主，才可以写入。

4、修改配置文件：

(1)Redis—Master 上的配置文件，红线地方必须在 Redis—Slave 做相应的修改：

```
#--------------------------------修改/etc/redis.conf 文件--------------------------#
protected-mode no    #必须要加的参数，在 3.2 版本
daemonize yes
pidfile /redis/run/redis.pid
port 6379
tcp-backlog 511
timeout 1800
tcp-keepalive 0
loglevel verbose
logfile "/redis/log/redis.log"
databases 16
save 900 1
save 300 10
save 60 10000
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
dir ./
slaveof 192.168.88.55 6379      #在 slave 上应修改为 master 的 IP 即为：192.168.88.11 6379
```

```
slave-read-only yes
slave-serve-stale-data yes
slave-read-only yes
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-disable-tcp-nodelay no
slave-priority 100
appendonly yes
appendfilename "appendonly.aof"
appendfsync everysec
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
aof-load-truncated yes
lua-time-limit 5000
slowlog-log-slower-than 10000
slowlog-max-len 128
latency-monitor-threshold 0
notify-keyspace-events ""
hash-max-ziplist-entries 512
hash-max-ziplist-value 64
list-max-ziplist-entries 512
list-max-ziplist-value 64
set-max-intset-entries 512
zset-max-ziplist-entries 128
zset-max-ziplist-value 64
hll-sparse-max-bytes 3000
activerehashing yes
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit slave 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60
hz 10
aof-rewrite-incremental-fsync yes
#-------------修改 Master  /etc/keepalived/keepalived.conf 文件---------------#
global_defs {
          lvs_id LVS_redis 80
         smtp_connect_timeout 30
       }
```

```
vrrp_script chk_redis {
        script "sh /etc/keepalived/scripts/redis_check.sh"
        interval 1
        weight 2
}
vrrp_instance VI_1 {
        state MASTER
        interface eth1
        virtual_router_id 60
        unicast_src_ip 192.168.88.11   #在 slave 上应修改为 slave 的 IP 即为：192.168.88.55
        unicast_peer {
            192.168.88.55              #在 slave 上应修改为 slave 的 IP 即为：192.168.88.11
        }
        priority 200
        advert_int 1
        track_script {
            chk_redis
        }
        virtual_ipaddress {
             192.168.88.111
        }
        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
}
##注：一下的检测脚本的 IP 都应该修改为对应主机的 IP 地址，即在从上则为 192.168.88.11。
#------------------------------添加 redis_check.sh 脚本--------------------------------#
#!/bin/bash
SERV=keepalived
CHECK_TIME=2
check() {
    /usr/local/bin/redis-cli  ping > /dev/null 2>&1
    ret=$?
    if [ $ret -ne 0 ];then
        return $ret;
    fi
}
```

```
while [ $CHECK_TIME -ne 0 ];do
    let "CHECK_TIME -= 1"
    check
    REDIS_OK=$?
    if [ $REDIS_OK -eq 0 ];then
        exit $REDIS_OK
    else
        if [ $CHECK_TIME -eq 0 ];then
            /etc/init.d/$SERV stop
            exit $REDIS_OK
        fi
    fi
done
#-----------------------------添加 redis_stop.sh 脚本------------------------------#
#!/bin/bash
###/etc/keepalived/scripts/redis_stop.sh
REDISCLI="/usr/local/bin/redis-cli "
LOGFILE="/etc/keepalived/log/redis-state.log"
pid=$$
echo "Run redis_stop.sh" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master]" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Being slave state..." >>$LOGFILE 2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 192.168.88.55 6379'" >>
$LOGFILE

$REDISCLI SLAVEOF  192.168.88.55 6379 >> $LOGFILE  2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] slave connect to 192.168.88.55  ok...
" >> $LOGFILE
#-----------------------------添加 redis_fault.sh 脚本------------------------------#
#!/bin/bash
###/etc/keepalived/scripts/redis_fault.sh
REDISCLI="/usr/local/bin/redis-cli "
LOGFILE="/etc/keepalived/log/redis-state.log"
pid=$$
echo "Run redis_fault.sh" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master]" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Being slave state..." >> $LOGFILE 2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 192.168.88.55 6379'" >> $LOGFILE
$REDISCLI SLAVEOF  192.168.88.55 6379 >> $LOGFILE  2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] slave connect to 192.168.88.55  ok..." >> $LOGFILE
#----------------------------添加 redis_backup.sh 脚本----------------------------#
#!/bin/bash
###/etc/keepalived/scripts/redis_backup.sh
REDISCLI="/usr/local/bin/redis-cli "
LOGFILE="/etc/keepalived/log/redis-state.log"
pid=$$
echo "Run redis_backup.sh" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master]" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Being slave state..." >> $LOGFILE 2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 192.168.88.55 6379'" >> $LOGFILE
$REDISCLI SLAVEOF  192.168.88.55 6379 >> $LOGFILE  2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] slave connect to 192.168.88.55  ok..." >> $LOGFILE
#------------------------------添加 redis_master.sh 脚本----------------------------#
redis_master.sh 文件如下：
#!/bin/bash
###/etc/keepalived/scripts/redis_master.sh
REDISCLI="/usr/local/bin/redis-cli "
LOGFILE="/etc/keepalived/log/redis-state.log"
pid=$$
echo "Run redis_master.sh" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver]" >> $LOGFILE
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 192.168.88.55 6379'" >> $LOGFILE
$REDISCLI SLAVEOF 192.168.88.55  6379 >> $LOGFILE  2>&1
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Run slaveof no one,close master/slave" >> $LOGFILE
$REDISCLI SLAVEOF NO ONE >> $LOGFILE 2>&1
```

```
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] wait other slave connect...." >> $LOGFILE
```

7、启动服务，模拟故障，检测 IP 可用

```
## redis 主从都启动 keepalived、redis 服务：
[root@vs1 ~]#service keepalived start
[root@vs1 ~]#ifconfig
[root@vs2 ~]#service keepalived start
[root@vs2 ~]#ifconfig
##测试主从同步：
[root@test ~]#redis-cli -h 192.168.88.11 -p 6379
192.168.88.11:6379>set ms “hello world”
[root@test ~]#redis-cli -h 192.168.88.22 -p 6379
192.168.88.22:6379>get ms
##模拟故障，关闭 vs1 的 redis：
[root@vs1 ~]#service keepalived stop（或者 service nginx stop）
[root@vs1 ~]#ip a
##检测 web 服务是否可用
[root@test ~]# curl http://192.168.88.111
```

参考链接：http://www.178linux.com/56546