**高可用Keepalived切换redis主从**

V1.0.0

杜标 @ Enlink 数据中心

**目录**

[1. 概述 3](#_Toc32355)

[1.1. 说明 3](#_Toc2194)

[2. 安装redis 3](#_Toc29742)

[2.1. 安装步骤 4](#_Toc21715)

[2.2. 配置redis 4](#_Toc17066)

[3. 安装keepalived 4](#_Toc6978)

[3.1. 安装步骤 4](#_Toc30413)

[3.2. 配置keepalived 5](#_Toc4389)

[3.2.1. master侧的配置 5](#_Toc25700)

[3.2.2. backup侧的配置 6](#_Toc18500)

[3.2.3. 测试 7](#_Toc30334)

[4. 三个脚本 7](#_Toc17972)

[4.1. 升级为master触发执行的脚本 7](#_Toc20499)

[4.2. 降级为backup触发执行的脚本 7](#_Toc28382)

[4.3. cron定时任务 8](#_Toc9978)

# 概述

本文档简述keepalived和redis在linux系统下的安装以及通过keepalived配置redis主从节点切换以实现redis高可用。

## 说明

本文档使用的keepalived版本是2.0.11，linux发行版是centos7，redis版本是5.0.3。本文档实现高可用的方式是用keepalived控制主从节点的相互切换，但系统承载力没有提高。在高并发、存在三台主机可用的情况下，可选用哨兵式、集群式来实现高可用目标。

keepalived实现redis高可用时，系统对外只暴露了虚拟ip（VIP），因此对客户端无感知。初始时，A机的redis是主节点，B机的redis是从节点。当主机A的redis死掉（或者A机宕机）之后，主机A的优先级降低，B机夺得了VIP，同时B机的redis由从节点升级成为主节点继续提供服务。当A机redis恢复后，B机仍持有VIP，A机的redis将成为B机redis的从节点。如图1-1所示：

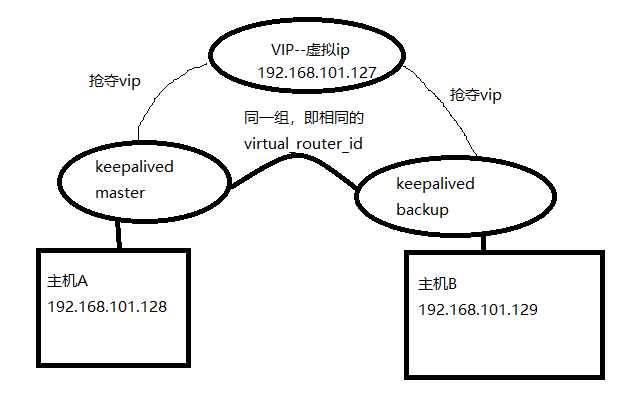


图 1-1 总体逻辑图

# 安装redis

本章节描述redis的安装步骤及redis的配置。

## 安装步骤

* 下载、编译和安装

|  |
| --- |
| wget http://download.redis.io/releases/redis-5.0.3.tar.gz  tar xzf redis-5.0.3.tar.gz  cd redis-5.0.3  make && make install |

* 启动

|  |
| --- |
| src/redis-server redis.conf |

## 配置redis

新建配置文件——redis.conf

|  |
| --- |
| daemonize yes  logfile /var/log/redis.log  bind 0.0.0.0 |

另一台redis也是同样配置，不同点是redis绑定的ip要根据实际配置。

# 安装keepalived

本章节描述keepalived的安装和配置。

## 安装步骤

* 前置条件

您的系统应该已经安装gcc、openssl-devel等工具，在下面步骤中你可以检查到缺失了哪些编译依赖的工具。

* 下载并解压

|  |
| --- |
| wget [http://www.keepalived.org/software/keepalived-2.0.11.tar.gz](http://www.keepalived.org/software/keepalived-2.0.11.tar.gz" \t "_blank)  tar -zxvf keepalived-2.0.11.tar.gz  cd keepalived-2.0.11 |

* 检查环境

|  |
| --- |
| ./configure |

* 编译以及编译安装

|  |
| --- |
| make && make install |

* 加入到可执行路径下

|  |
| --- |
| cp keepalived/keepalived /usr/local/sbin |

* 加入到系统服务中

|  |
| --- |
| cp keepalived/keepalived.service /usr/lib/systemd/system/ |

* 开机启动

|  |
| --- |
| systemctl enable keepalived |

另一台主机也是这样安装。

## 配置keepalived

本小节描述master和backup的配置。

### master侧的配置

* master具体配置

|  |
| --- |
| global\_defs {  router\_id redis  enable\_script\_security  }  vrrp\_instance VI\_REDIS {  state BACKUP  interface ens33  virtual\_router\_id 51  priority 100  advert\_int 1  nopreempt  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  192.168.101.127  }  notify\_master "/opt/keepalived/redis\_master.sh"  notify\_backup "/opt/keepalived/redis\_backup.sh"  } |

注意：interface要写您实际机器上的网卡名称。相同的virtual\_router\_id代表同一个组，所以backup端配置virtual\_router\_id要跟这个一致。nopreempt表明其它系统恢复后即使优先权高也不能抢夺我的VIP。

* 创建用户（keepalived执行脚本需要一个默认用户，叫作keepalived\_script）

|  |
| --- |
| useradd keepalived\_script |

现在master侧配置完毕。

### backup侧的配置

* backup具体配置

|  |
| --- |
| global\_defs {  router\_id redis  enable\_script\_security  }  vrrp\_instance VI\_REDIS {  state BACKUP  interface ens32  virtual\_router\_id 51  priority 50  advert\_int 1  nopreempt  authentication {  auth\_type PASS  auth\_pass 1111  }  virtual\_ipaddress {  192.168.101.127  }  notify\_master "/opt/keepalived/redis\_master.sh"  notify\_backup "/opt/keepalived/redis\_backup.sh"  } |

注意：backup侧的priority 是50，而master侧的priority 是100。state的状态不能决定真正的状态，真正的状态是由优先级决定的。指定为nopreempt的时候，state必须是BACKUP。

* 创建用户（keepalived执行脚本需要一个默认用户，叫作keepalived\_script）

|  |
| --- |
| useradd keepalived\_script |

现在backup侧配置完毕。

### 测试

* 分别关闭master机和backup机的防火墙
* 启动master机的keepalived，查看当前网卡IP信息，VIP被当前机器持有。
* 启动backup机的keepalived，查看当前网卡IP信息，没有VIP。
* 关闭master机，查看backup机器的网卡IP信息，发现backup机持有VIP。
* 启动master机，先关闭防火墙，后启动了keepalived，查看当前网卡IP信息，无VIP。查看backup机器的网卡IP信息，VIP仍被backup机持有。

# 三个脚本

本章节描述涉及到的所有脚本。

## 升级为master触发执行的脚本

|  |
| --- |
| #!/bin/bash  REDISCLI="/opt/redis-5.0.3/src/redis-cli"  LOGFILE="/var/log/keepalived-redis-state.log"  echo "[master]" >> $LOGFILE  date >> $LOGFILE  echo "Being master...." >> $LOGFILE 2>&1  $REDISCLI SLAVEOF NO ONE >> $LOGFILE 2>&1 |

## 降级为backup触发执行的脚本

|  |
| --- |
| #!/bin/bash  REDISCLI="/opt/redis-5.0.3/src/redis-cli"  LOGFILE="/var/log/keepalived-redis-state.log"  echo "[backup]" >> $LOGFILE  date >> $LOGFILE  echo "Being slave...." >> $LOGFILE 2>&1  $REDISCLI SLAVEOF 192.168.101.128 6379 >> $LOGFILE 2>&1 |

注意：IP要根据实际编写，master（A机）侧填写backup（B机）侧的IP，backup（B机）侧填写master（A机）侧的IP。

## cron定时任务

此定时任务，每分钟监视一次redis。若redis崩溃，拉起redis进程。

|  |
| --- |
| #!/bin/bash  CHECK=`/opt/redis-5.0.3/src/redis-cli PING`  if [ "$CHECK" != "PONG" ] ;then  echo “start redis......” >> /var/log/cron-redis.log  /opt/redis-5.0.3/src/redis-cli redis.conf  fi |