关于本文档

文档名称：《高可用集群 KEEPALIVED》

使用协议：《知识共享公共许可协议(CCPL)》

贡献者

贡献者名称	贡献度	文档变更记录	个人主页
马哥（马永亮）	主编		http://github.com/iKubernetes/
王晓春	作者	80 页	http://www.wangxiaochun.com

文档协议

致谢

本文档中，部分素材参考了相关项目的文档，以及通过搜索引擎获得的内容，这里先一并向相关的贡献者表示感谢。

高可用集群 KEEPALIVED

内容概述

- **高可用集群技术**
- **VRRP 协议**
- **KeepAlived 组成和安装**

- **KeepAlived 实现 VRRP**
- **KeepAlived 实现 LVS 的高可用**
- **KeepAlived 实现其它应用的高可用**

1 高可用集群

1.1 集群类型 Cluster

- LB：Load Balance 负载均衡

  LVS/HAProxy/nginx（http/upstream, stream/upstream）
- HA：High Availability 高可用集群

  MySQL、Redis、Zookeeper、Kafka 有状态的服务

  KeepAlived 通用的高可用集群,更适合无状态的服务

  SPoF: Single Point of Failure，解决单点故障
- HPC：High Performance Computing 高性能集群

  https://www.top500.org

1.2 系统可用性

SLA：Service-Level Agreement 服务等级协议（提供服务的企业与客户之间就服务的品质、水准、性能等方面所达成的双方共同认可的协议或契约）

A = MTBF / (MTBF+MTTR)

MTBF:Mean Time Between Failure 平均无故障时间，正常时间

MTTR:Mean Time To Restoration（ repair）平均恢复前时间，故障时间

```
99.95%:(60*24*30)*(1-0.9995)=21.6分钟   #一般按一个月或一年非计划内停机时间统计
```

指标： 99%,99.9%, 99.99%, 99.999%,99.9999%

1.3 系统故障

硬件故障：设计缺陷、wear out（损耗）、自然灾害.....

软件故障：设计缺陷 bug

人为故障：故意或无意

1.4 实现高可用

提升系统高用性的解决方案：降低MTTR- Mean Time To Repair(平均故障时间)

解决方案：建立多节点的冗余机制

- active/passive 主/备
- active/active 双主
- active --> HEARTBEAT --> passive

- active <--> HEARTBEAT <--> active

1.5 高可用相关技术

1.5.1 HA Service

资源：组成一个高可用服务的“组件”，比如：vip，service process，shared storage

- passive node的数量
- 资源切换

1.5.2 Shared Storage

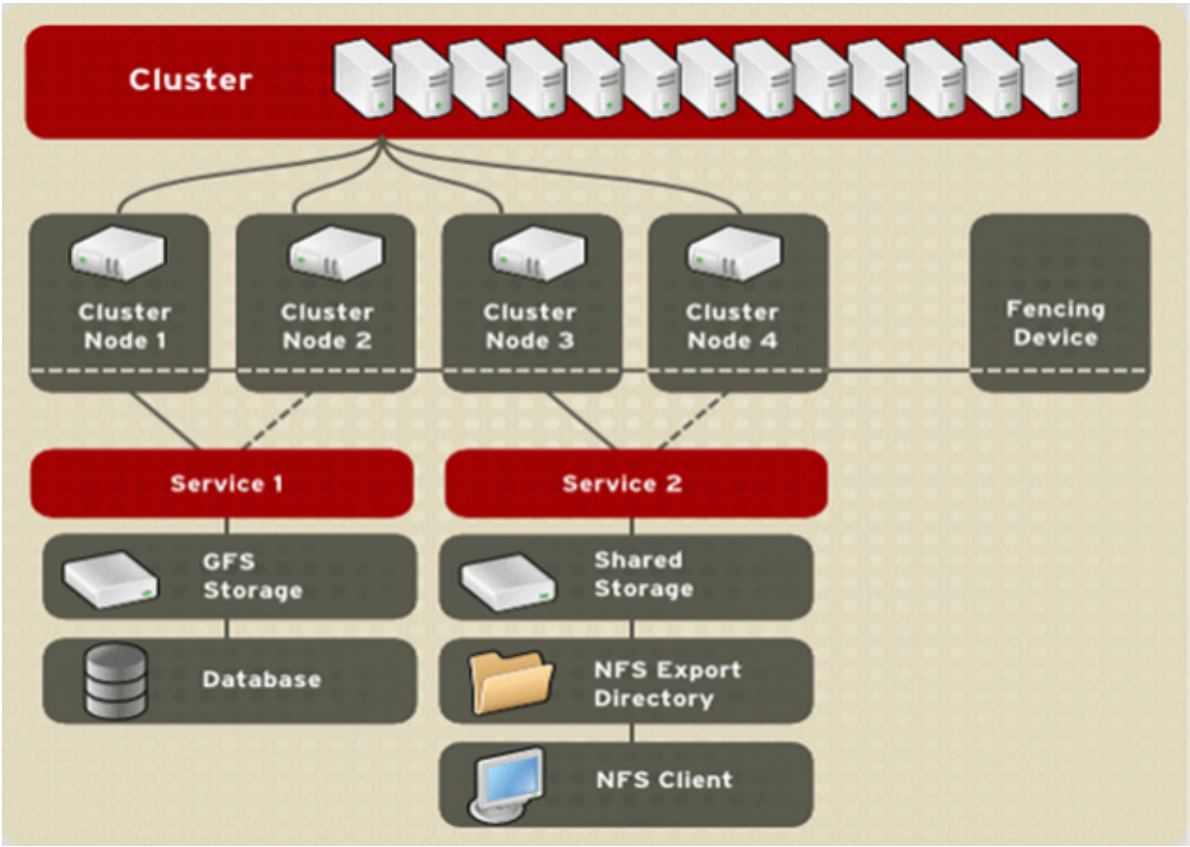
- NAS(Network Attached Storage)：网络附加存储，基于网络的共享文件系统。
- SAN(Storage Area Network)：存储区域网络，基于网络的块级别的共享
- 分布式存储: Ceph、GlusterFS、HDFS、GFS、DFS、moosefs，MinIO (OSS 对象存储)等

1.5.3 HA Cluster 实现方案

1.5.3.1 AIS：Applicaiton Interface Specification 应用程序接口规范

- RHCS：Red Hat Cluster Suite 红帽集群套件

  参考资料：https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/5/html/cluster_suite_overview/ch.gfscs.cluster-overview-cso



- heartbeat：基于心跳监测实现服务高可用
- pacemaker+corosync：资源管理与故障转移

1.5.3.2 VRRP：Virtual Router Redundancy Protocol

虚拟路由冗余协议,解决静态网关单点风险

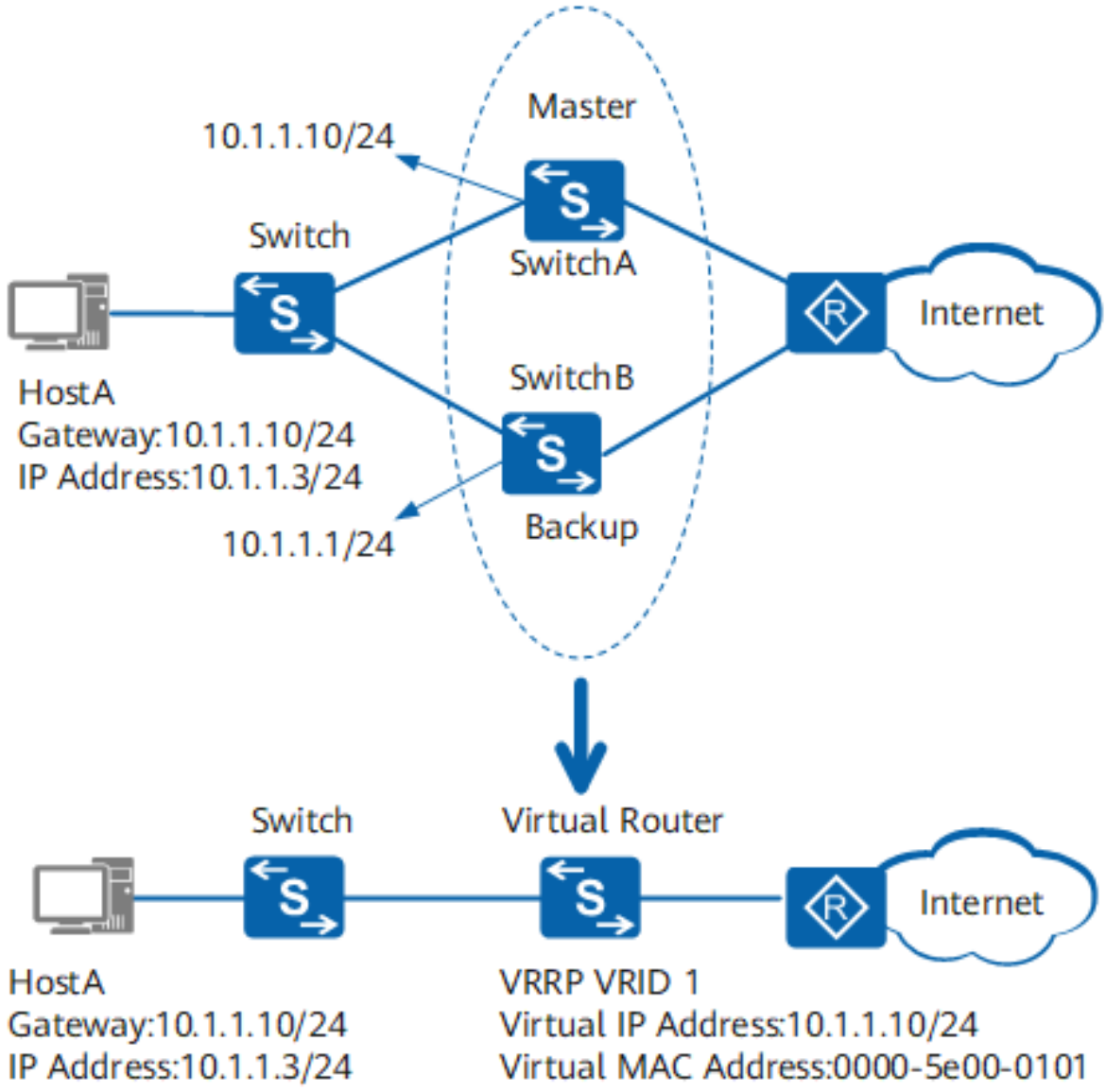
- 物理层:路由器、三层交换机
- 软件层:keepalived

1.5.4 VRRP

1.5.4.1 VRRP 网络层硬件实现

参考链接:

```
https://app.huawei.com/weiknow#/weiknow?
lang=zh&question=VRRP%E6%98%AF%E4%BB%8O%E4%B9%88
https://support.huawei.com/enterprise/zh/doc/EDOC1000141382/19258d72/basic-
concepts-of-vrrp
https://wenku.baidu.com/view/dc0afaa6f524ccbff1218416.html
https://wenku.baidu.com/view/281ae109ba1aa8114431d9d0.html
```



1.5.4.2 VRRP 相关术语

- 虚拟路由器：Virtual Router
- 虚拟路由器标识：VRID(0-255)，唯一标识虚拟路由器
- VIP：Virtual IP
- VMAC：Virutal MAC (00-00-5e-00-01-VRID)
- 物理路由器：

  master：主设备

  backup：备用设备

  priority：优先级

1.5.4.3 VRRP 相关技术

通告：心跳，优先级等；周期性

工作方式：抢占式，非抢占式

安全认证：

- 无认证
- 简单字符认证：预共享密钥
- MD5

工作模式：

- 主/备：单个虚拟路由器
- 主/主：主/备（虚拟路由器1），备/主（虚拟路由器2）

2 Keepalived 架构和安装

2.1 Keepalived 介绍

vrrp 协议的软件实现，原生设计目的为了高可用 ipvs服务

keepalived 是高可用集群的通用无状态应用解决方案

官网：http://keepalived.org/

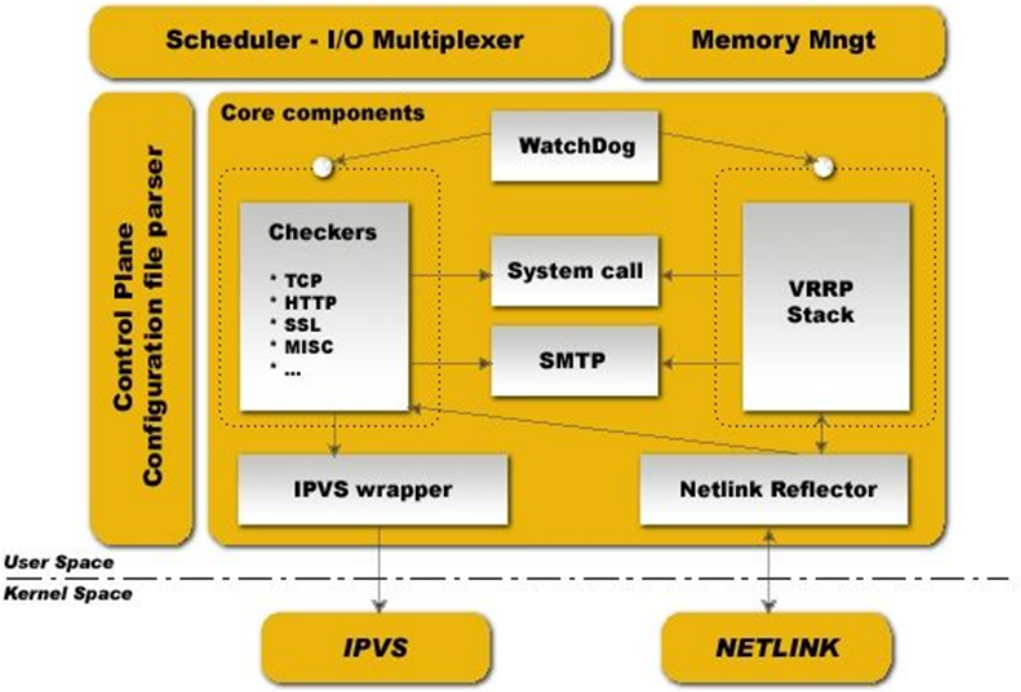
功能：

- 基于vrrp协议完成IP地址流动
- 为vip地址所在的节点生成ipvs规则(在配置文件中预先定义)
- 为ipvs集群的各RS做健康状态检测
- 基于脚本调用接口完成脚本中定义的功能，进而影响集群事务，以此支持实现nginx、haproxy等服务的高可用

2.2 Keepalived 架构

官方文档：

```
https://keepalived.org/doc/
http://keepalived.org/documentation.html
```



- 用户空间核心组件：

  vrrp stack：VIP消息通告

  checkers：监测 Real Server

  system call：实现 vrrp 协议状态转换时调用脚本的功能

  SMTP：邮件组件

  IPVS wrapper：生成 IPVS 规则

  Netlink Reflector：网络接口

  WatchDog：监控进程

- 控制组件：提供keepalived.conf 的解析器，完成Keepalived配置
- IO复用器：针对网络目的而优化的自己的线程抽象
- 内存管理组件：为某些通用的内存管理功能（例如分配，重新分配，发布等）提供访问权限

Keepalived进程树

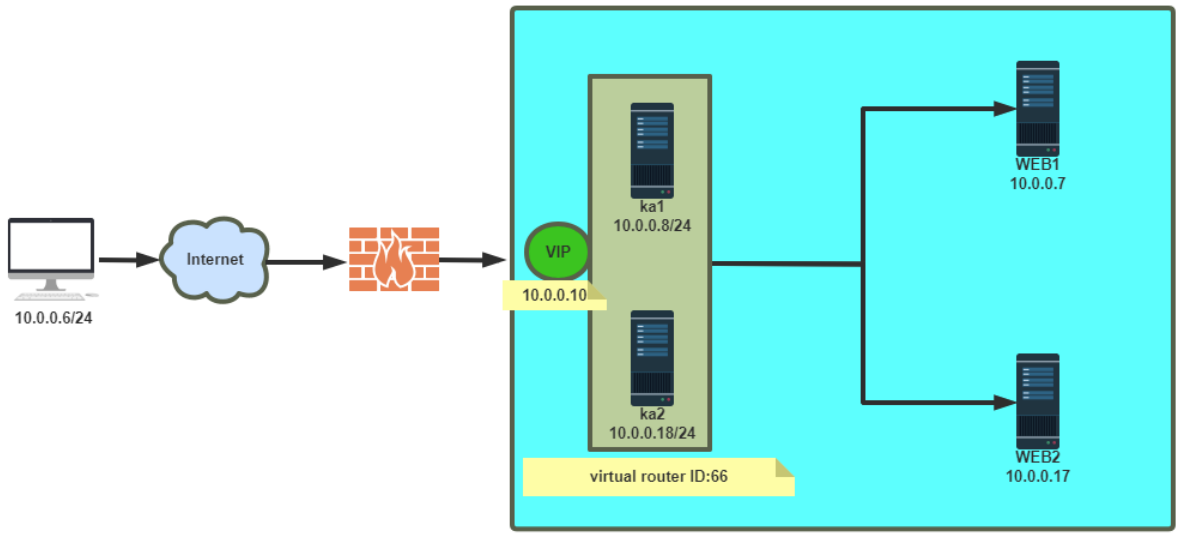
```
Keepalived  <-- Parent process monitoring children
\_ Keepalived   <-- VRRP child
\_ Keepalived   <-- Healthchecking child
```

VRRP 协议

```
> Frame 10: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{82A4737E-2875-456A-94C6-ABF7B3BE4CB8}, id 0
> Ethernet II, Src: VMware_be:a5:f0 (00:0c:29:be:a5:f0), Dst: IPv4mcast_64:64:64 (01:00:5e:64:64:64)
v Internet Protocol Version 4, Src: 10.0.0.201, Dst: 224.100.100.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 48
    Identification: 0x000b (11)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 255
    Protocol: VRRP (112)
    Header Checksum: 0x6b01 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.0.201
    Destination Address: 224.100.100.100
v Virtual Router Redundancy Protocol
  v Version 2, Packet type 1 (Advertisement)
      0010 .... = VRRP protocol version: 2
      .... 0001 = VRRP packet type: Advertisement (1)
    Virtual Rtr ID: 66
    Priority: 100 (Default priority for a backup VRRP router)
    Addr Count: 3
    Auth Type: Simple Text Authentication [RFC 2338] / Reserved [RFC 3768] (1)
    Adver Int: 1
    Checksum: 0x45ed [correct]
    [Checksum Status: Good]
    IP Address: 192.168.200.16
    IP Address: 192.168.200.17
    IP Address: 192.168.200.18
    Authentication String: 123456
```

```
0000  01 00 5e 64 64 64 00 0c  29 be a5 f0 08 00 45 c0   ··^ddd·· )·····E·
0010  00 30 00 0b 00 00 ff 70  6b 01 0a 00 00 c9 e0 64   ·0·····p k······d
0020  64 64 21 42 64 03 01 01  45 ed c0 a8 c8 10 c0 a8   ddlBd··· E·······
0030  c8 11 c0 a8 c8 12 31 32  33 34 35 36 00 00        ·····12 3456··
```

2.3 Keepalived 环境准备



- 各节点时间必须同步：ntp, chrony
- 关闭防火墙及SELinux
- 各节点之间可通过主机名互相通信：非必须
- 建议使用/etc/hosts文件实现：非必须
- 各节点之间的root用户可以基于密钥认证的ssh服务完成互相通信：非必须

2.4 Keepalived 相关文件

- 软件包名：keepalived
- 主程序文件：/usr/sbin/keepalived
- 主配置文件：/etc/keepalived/keepalived.conf
- 配置文件示例：/usr/share/doc/keepalived/
- Unit File：/lib/systemd/system/keepalived.service
- Unit File的环境配置文件：
  - /etc/sysconfig/keepalived CentOS
  - /etc/default/keepalived Ubuntu

注意：CentOS 7 上有 bug，可能有下面情况出现

```
systemctl restart keepalived                                    #新配置可能无法生效
systemctl stop keepalived;systemctl start keepalived            #无法停止进程，需要 kill
停止
```

2.5 Keepalived 安装

安装方法

- 包安装
- 编译安装
- 容器运行

2.5.1 包安装

```
#CentOS
[root@centos ~]#yum -y install keepalived

#ubuntu
[root@ubuntu1804 ~]#apt update && apt -y install keepalived
```

2.5.1.1 CentOS 安装 keepalived

```
[root@rocky8 ~]#yum list keepalived
Last metadata expiration check: 0:00:40 ago on Sun 25 Jun 2023 10:22:33 AM CST.
Available Packages
keepalived.x86_64                                  2.1.5-9.el8
                           AppStrea

[root@centos8 ~]#dnf -y install keepalived
[root@centos8 ~]#dnf info keepalived
Last metadata expiration check: 0:00:24 ago on Thu 26 Mar 2020 07:28:36 PM CST.
Installed Packages
Name         : keepalived
Version      : 2.0.10
Release      : 4.el8_0.2
Architecture : x86_64
Size         : 1.4 M
Source       : keepalived-2.0.10-4.el8_0.2.src.rpm
Repository   : @System
From repo    : AppStream
Summary      : High Availability monitor built upon LVS, VRRP and service
pollers
URL          : http://www.keepalived.org/
License      : GPLv2+
Description  : Keepalived provides simple and robust facilities for load
balancing
             : and high availability to Linux system and Linux based
infrastructures.
             : The load balancing framework relies on well-known and widely used
             : Linux Virtual Server (IPVS) kernel module providing Layer4 load
             : balancing. Keepalived implements a set of checkers to dynamically
and
```

```
             : adaptively maintain and manage load-balanced server pool
according
             : their health. High availability is achieved by VRRP protocol. VRRP
is
             : a fundamental brick for router failover. In addition, keepalived
             : implements a set of hooks to the VRRP finite state machine
providing
             : low-level and high-speed protocol interactions. Keepalived
frameworks
             : can be used independently or all together to provide resilient
             : infrastructures.

[root@centos8 ~]#systemctl start keepalived.service
[root@centos8 ~]#ps auxf |grep keepalived
root       12864  0.0  0.1  12108  1100 pts/0    S+   19:25   0:00          |
\_ grep --color=auto keepalive
root       12835  0.0  0.3  91444  2484 ?        Ss   19:24   0:00
/usr/sbin/keepalived -D
root       12836  0.0  0.5  91576  4212 ?        S    19:24   0:00  \_
/usr/sbin/keepalived -D
root       12837  0.0  0.5  91444  4620 ?        S    19:24   0:00  \_
/usr/sbin/keepalived -D


[root@centos8 ~]#pstree -p
......
           ├─keepalived(12835)─┬─keepalived(12836)
           |                   └─keepalived(12837)
......

[root@rocky8 ~]#keepalived -v |grep mark
Keepalived v2.1.5 (07/13,2020)

Copyright(C) 2001-2020 Alexandre Cassen, <acassen@gmail.com>

Built with kernel headers for Linux 4.18.0
Running on Linux 4.18.0-348.el8.0.2.x86_64 #1 SMP Sun Nov 14 00:51:12 UTC 2021

configure options: --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-
gnu --program-prefix= --disable-dependency-tracking --prefix=/usr --exec-
prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --sysconfdir=/etc --
datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64 --
libexecdir=/usr/libexec --localstatedir=/var --sharedstatedir=/var/lib --
mandir=/usr/share/man --infodir=/usr/share/info --disable-libiptc --disable-
ipset --enable-snmp --enable-snmp-rfc --enable-sha1 --with-init=systemd
build_alias=x86_64-redhat-linux-gnu host_alias=x86_64-redhat-linux-gnu
PKG_CONFIG_PATH=:/usr/lib64/pkgconfig:/usr/share/pkgconfig CFLAGS=-O2 -g -pipe -
Wall -Werror=format-security -Wp,-D_FORTIFY_SOURCE=2 -Wp,-D_GLIBCXX_ASSERTIONS -
fexceptions -fstack-protector-strong -grecord-gcc-switches -
specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -specs=/usr/lib/rpm/redhat/redhat-
annobin-cc1 -m64 -mtune=generic -fasynchronous-unwind-tables -fstack-clash-
protection -fcf-protection LDFLAGS=-Wl,-z,relro -Wl,-z,now -
specs=/usr/lib/rpm/redhat/redhat-hardened-ld

Config options:  LVS VRRP VRRP_AUTH OLD_CHKSUM_COMPAT FIB_ROUTING SNMP_V3_FOR_V2
SNMP_VRRP SNMP_CHECKER SNMP_RFCV2 SNMP_RFCV3
```

```
System options:  PIPE2 SIGNALFD INOTIFY_INIT1 VSYSLOG EPOLL_CREATE1 IPV4_DEVCONF
IPV6_ADVANCED_API LIBNL3 RTA_ENCAP RTA_EXPIRES RTA_NEWDST RTA_PREF
FRA_SUPPRESS_PREFIXLEN FRA_SUPPRESS_IFGROUP FRA_TUN_ID RTAX_CC_ALGO RTAX_QUICKACK
RTEXT_FILTER_SKIP_STATS FRA_L3MDEV FRA_UID_RANGE RTAX_FASTOPEN_NO_COOKIE RTA_VIA
FRA_OIFNAME FRA_PROTOCOL FRA_IP_PROTO FRA_SPORT_RANGE FRA_DPORT_RANGE
RTA_TTL_PROPAGATE IFA_FLAGS IP_MULTICAST_ALL LWTUNNEL_ENCAP_MPLS
LWTUNNEL_ENCAP_ILA NET_LINUX_IF_H_COLLISION LIBIPTC_LINUX_NET_IF_H_COLLISION
LIBIPVS_NETLINK IPVS_DEST_ATTR_ADDR_FAMILY IPVS_SYNCD_ATTRIBUTES IPVS_64BIT_STATS
VRRP_VMAC VRRP_IPVLAN IFLA_LINK_NETNSID CN_PROC SOCK_NONBLOCK SOCK_CLOEXEC O_PATH
GLOB_BRACE INET6_ADDR_GEN_MODE VRF SO_MARK SCHED_RESET_ON_FORK
```

2.5.1.2 Ubuntu 安装 keepalived

范例：Ubuntu22.04

```
[root@ubuntu2204 ~]#apt list keepalived
正在列表... 完成
keepalived/jammy 1:2.2.4-0.2build1 amd64
[root@ubuntu2204 ~]#apt update;apt -y install keepalived

#默认缺少配置，服务无法启动，提示/etc/keepalived/keepalived.conf 不存在
[root@ubuntu2204 ~]#cat /lib/systemd/system/keepalived.service
[Unit]
Description=Keepalive Daemon (LVS and VRRP)
After=network-online.target
Wants=network-online.target
# Only start if there is a configuration file
ConditionFileNotEmpty=/etc/keepalived/keepalived.conf  #此配置文件不存在，导致无法启动


[root@ubuntu2204 ~]#systemctl status keepalived.service
○ keepalived.service - Keepalive Daemon (LVS and VRRP)
     Loaded: loaded (/lib/systemd/system/keepalived.service; enabled; vendor
preset: enabled)
     Active: inactive (dead)
  Condition: start condition failed at Thu 2023-01-12 15:28:42 CST; 4s ago
             └─ ConditionFileNotEmpty=/etc/keepalived/keepalived.conf was not
met

1月 12 15:26:47 ubuntu2204.wang.org systemd[1]: Condition check resulted in
Keepalive Daemon (LVS and VRRP) being skipped.
1月 12 15:28:42 ubuntu2204.wang.org systemd[1]: Condition check resulted in
Keepalive Daemon (LVS and VRRP) being skipped.

#利用模板生成生配置文件
[root@ubuntu2204 ~]#cp /usr/share/doc/keepalived/samples/keepalived.conf.sample
/etc/keepalived/keepalived.conf
[root@ubuntu2204 ~]#systemctl start keepalived.service
[root@ubuntu2204 ~]#systemctl status keepalived.service
● keepalived.service - Keepalive Daemon (LVS and VRRP)
     Loaded: loaded (/lib/systemd/system/keepalived.service; enabled; vendor
preset: enabled)
     Active: active (running) since Thu 2023-01-12 15:29:54 CST; 1s ago
   Main PID: 2174 (keepalived)
      Tasks: 3 (limit: 2196)
     Memory: 4.5M
        CPU: 40ms
```

```
     CGroup: /system.slice/keepalived.service
             ├─2174 /usr/sbin/keepalived --dont-fork
             ├─2175 /usr/sbin/keepalived --dont-fork
             └─2176 /usr/sbin/keepalived --dont-fork


[root@ubuntu2204 ~]#keepalived -v
Keepalived v2.2.4 (08/21,2021)

Copyright(C) 2001-2021 Alexandre Cassen, <acassen@gmail.com>

Built with kernel headers for Linux 5.15.27
Running on Linux 5.15.0-75-generic #82-Ubuntu SMP Tue Jun 6 23:10:23 UTC 2023
Distro: Ubuntu 22.04.2 LTS

configure options: --build=x86_64-linux-gnu --prefix=/usr --
includedir=${prefix}/include --mandir=${prefix}/share/man --
infodir=${prefix}/share/info --sysconfdir=/etc --localstatedir=/var --disable-
option-checking --disable-silent-rules --libdir=${prefix}/lib/x86_64-linux-gnu -
-runstatedir=/run --disable-maintainer-mode --disable-dependency-tracking --
enable-snmp --enable-sha1 --enable-snmp-rfcv2 --enable-snmp-rfcv3 --enable-dbus
--enable-json --enable-bfd --enable-regex --with-init=systemd
build_alias=x86_64-linux-gnu CFLAGS=-g -O2 -ffile-prefix-map=/build/keepalived-
NeItXh/keepalived-2.2.4=. -flto=auto -ffat-lto-objects -flto=auto -ffat-lto-
objects -fstack-protector-strong -Wformat -Werror=format-security LDFLAGS=-Wl,-
Bsymbolic-functions -flto=auto -ffat-lto-objects -flto=auto -Wl,-z,relro
CPPFLAGS=-Wdate-time -D_FORTIFY_SOURCE=2

Config options:  NFTABLES LVS REGEX VRRP VRRP_AUTH VRRP_VMAC JSON BFD
OLD_CHKSUM_COMPAT SNMP_V3_FOR_V2 SNMP_VRRP SNMP_CHECKER SNMP_RFCV2 SNMP_RFCV3
DBUS INIT=systemd SYSTEMD_NOTIFY

System options:  VSYSLOG MEMFD_CREATE IPV4_DEVCONF LIBNL3 RTA_ENCAP RTA_EXPIRES
RTA_NEWDST RTA_PREF FRA_SUPPRESS_PREFIXLEN FRA_SUPPRESS_IFGROUP FRA_TUN_ID
RTAX_CC_ALGO RTAX_QUICKACK RTEXT_FILTER_SKIP_STATS FRA_L3MDEV FRA_UID_RANGE
RTAX_FASTOPEN_NO_COOKIE RTA_VIA FRA_PROTOCOL FRA_IP_PROTO FRA_SPORT_RANGE
FRA_DPORT_RANGE RTA_TTL_PROPAGATE IFA_FLAGS LWTUNNEL_ENCAP_MPLS
LWTUNNEL_ENCAP_ILA NET_LINUX_IF_H_COLLISION LIBIPVS_NETLINK
IPVS_DEST_ATTR_ADDR_FAMILY IPVS_SYNCD_ATTRIBUTES IPVS_64BIT_STATS IPVS_TUN_TYPE
IPVS_TUN_CSUM IPVS_TUN_GRE VRRP_IPVLAN IFLA_LINK_NETNSID GLOB_BRACE
GLOB_ALTDIRFUNC INET6_ADDR_GEN_MODE VRF SO_MARK
```

范例：Ubuntu20.04

```
[root@ubuntu2004 ~]#apt list keepalived
Listing... Done
keepalived/focal-updates,now 1:2.0.19-2ubuntu0.2 amd64 [installed]
N: There are 2 additional versions. Please use the '-a' switch to see them.

[root@ubuntu2004 ~]#apt update && apt install -y keepalived

#默认没有配置文件无法启动
[root@ubuntu2004 ~]#systemctl status keepalived

#利用范例生成配置文件
[root@ubuntu2004 ~]#cp /usr/share/doc/keepalived/samples/keepalived.conf.sample
/etc/keepalived/keepalived.conf
```

```
[root@ubuntu2004 ~]#systemctl start keepalived

[root@ubuntu2004 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:0c:29:0a:b2:20 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.100/24 brd 10.0.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.11/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.12/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.13/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe0a:b220/64 scope link
       valid_lft forever preferred_lft forever

[root@ubuntu2004 ~]#pstree -p |grep keepalived
           |-keepalived(6260)-+-keepalived(6274)
           |                  `-keepalived(6275)

[root@ubuntu2004 ~]#ps auxf |grep keepalived
root        6460  0.0  0.0   6432   720 pts/1    S+   10:35   0:00  |       \_
grep --color=auto keepalived
root        6260  0.0  0.4  25372  8132 ?        Ss   10:29   0:00
/usr/sbin/keepalived --dont-fork
root        6274  0.0  0.1  25372  2812 ?        S    10:29   0:00  \_
/usr/sbin/keepalived --dont-fork
root        6275  0.0  0.1  25372  2916 ?        S    10:29   0:00  \_
/usr/sbin/keepalived --dont-fork
```

范例： Ubuntu18.04

```
[root@ubuntu1804 ~]#apt install keepalived -y
[root@ubuntu1804 ~]#dpkg -s keepalived
Package: keepalived
Status: install ok installed
Priority: extra
Section: admin
Installed-Size: 824
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Version: 1:1.3.9-1ubuntu0.18.04.2
Depends: iproute2, libc6 (>= 2.27), libglib2.0-0 (>= 2.26.0), libip4tc0 (>=
1.6.0+snapshot20161117), libip6tc0 (>= 1.6.0+snapshot20161117), libnl-3-200 (>=
3.2.27), libnl-genl-3-200 (>= 3.2.7), libnl-route-3-200 (>= 3.2.7), libsnmp30
(>= 5.7.3+dfsg-1.8ubuntu3.1~dfsg), libssl1.1 (>= 1.1.0), libxtables12 (>=
1.6.0+snapshot20161117)
```

```
Recommends: ipvsadm
Conffiles:
 /etc/dbus-1/system.d/org.keepalived.Vrrp1.conf 6b020ff46c6425d3a9cfa179814d7253
 /etc/default/keepalived 6b2e3432e4ae31b444058ba2b0d1f06a
 /etc/init.d/keepalived 0312972e0718331b4c90b3b98e623624
Description: Failover and monitoring daemon for LVS clusters
 keepalived is used for monitoring real servers within a Linux
 Virtual Server (LVS) cluster.  keepalived can be configured to
 remove real servers from the cluster pool if it stops responding,
 as well as send a notification email to make the admin aware of
 the service failure.
 .
 In addition, keepalived implements an independent Virtual Router
 Redundancy Protocol (VRRPv2; see rfc2338 for additional info)
 framework for director failover.
 .
 You need a kernel >= 2.4.28 or >= 2.6.11 for keepalived.
 See README.Debian for more information.
Homepage: http://keepalived.org
Original-Maintainer: Alexander Wirt <formorer@debian.org>


[root@ubuntu1804 ~]#dpkg -L keepalived
/.
/etc
/etc/dbus-1
/etc/dbus-1/system.d
/etc/dbus-1/system.d/org.keepalived.Vrrp1.conf
/etc/default
/etc/default/keepalived
/etc/init.d
/etc/init.d/keepalived
/etc/keepalived
/lib
/lib/systemd
/lib/systemd/system
/lib/systemd/system/keepalived.service
/usr
/usr/bin
/usr/bin/genhash
/usr/sbin
/usr/sbin/keepalived
/usr/share
/usr/share/dbus-1
/usr/share/dbus-1/interfaces
/usr/share/dbus-1/interfaces/org.keepalived.Vrrp1.Instance.xml
/usr/share/dbus-1/interfaces/org.keepalived.Vrrp1.Vrrp.xml
/usr/share/doc
/usr/share/doc/keepalived
/usr/share/doc/keepalived/AUTHOR
/usr/share/doc/keepalived/CONTRIBUTORS
/usr/share/doc/keepalived/README
/usr/share/doc/keepalived/TODO
/usr/share/doc/keepalived/changelog.Debian.gz
/usr/share/doc/keepalived/copyright
/usr/share/doc/keepalived/keepalived.conf.SYNOPSIS.gz
/usr/share/doc/keepalived/samples
/usr/share/doc/keepalived/samples/client.pem
```

```
/usr/share/doc/keepalived/samples/dh1024.pem
/usr/share/doc/keepalived/samples/keepalived.conf.HTTP_GET.port
/usr/share/doc/keepalived/samples/keepalived.conf.IPv6
/usr/share/doc/keepalived/samples/keepalived.conf.SMTP_CHECK
/usr/share/doc/keepalived/samples/keepalived.conf.SSL_GET
/usr/share/doc/keepalived/samples/keepalived.conf.fwmark
/usr/share/doc/keepalived/samples/keepalived.conf.inhibit
/usr/share/doc/keepalived/samples/keepalived.conf.misc_check
/usr/share/doc/keepalived/samples/keepalived.conf.misc_check_arg
/usr/share/doc/keepalived/samples/keepalived.conf.quorum
/usr/share/doc/keepalived/samples/keepalived.conf.sample
/usr/share/doc/keepalived/samples/keepalived.conf.status_code
/usr/share/doc/keepalived/samples/keepalived.conf.track_interface
/usr/share/doc/keepalived/samples/keepalived.conf.virtual_server_group
/usr/share/doc/keepalived/samples/keepalived.conf.virtualhost
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.localcheck
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.lvs_syncd
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.routes
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.rules
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.scripts
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.static_ipaddress
/usr/share/doc/keepalived/samples/keepalived.conf.vrrp.sync
/usr/share/doc/keepalived/samples/root.pem
/usr/share/doc/keepalived/samples/sample.misccheck.smbcheck.sh
/usr/share/doc/keepalived/samples/sample_notify_fifo.sh
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/genhash.1.gz
/usr/share/man/man5
/usr/share/man/man5/keepalived.conf.5.gz
/usr/share/man/man8
/usr/share/man/man8/keepalived.8.gz
/usr/share/snmp
/usr/share/snmp/mibs
/usr/share/snmp/mibs/KEEPALIVED-MIB.txt
/usr/share/snmp/mibs/VRRP-MIB.txt
/usr/share/snmp/mibs/VRRPv3-MIB.txt
[root@ubuntu1804 ~]#cp /usr/share/doc/keepalived/samples/keepalived.conf.sample
/etc/keepalived/keepalived.conf
[root@ubuntu1804 ~]#systemctl start keepalived.service
[root@ubuntu1804 ~]#systemctl status keepalived.service
● keepalived.service - Keepalive Daemon (LVS and VRRP)
   Loaded: loaded (/lib/systemd/system/keepalived.service; enabled; vendor
preset: enabled)
   Active: active (running) since Thu 2020-03-26 19:33:48 CST; 1min 9s ago
  Process: 3208 ExecStart=/usr/sbin/keepalived $DAEMON_ARGS (code=exited,
status=0/SUCCESS)
 Main PID: 3209 (keepalived)
    Tasks: 3 (limit: 1084)
   CGroup: /system.slice/keepalived.service
           ├─3209 /usr/sbin/keepalived
           ├─3210 /usr/sbin/keepalived
           └─3211 /usr/sbin/keepalived

Mar 26 19:34:04 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Timeout
connecting server [192.168.200.2]:tcp:1358.
```

```
Mar 26 19:34:10 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Timeout
connecting server [192.168.200.2]:tcp:1358.
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Timeout
connecting server [192.168.200.2]:tcp:1358.
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Check on
service [192.168.200.2]:tcp:1358 failed after 3 retry.
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Removing
service [192.168.200.2]:tcp:1358 to VS [10.10.10.2]:tc
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Lost quorum
1-0=1 > 0 for VS [10.10.10.2]:tcp:1358
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Adding
sorry server [192.168.200.200]:tcp:1358 to VS [10.10.10.
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Removing
alive servers from the pool for VS [10.10.10.2]:tcp:13
Mar 26 19:34:16 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Remote SMTP
server [192.168.200.1]:25 connected.
Mar 26 19:34:37 ubuntu1804.wang.org Keepalived_healthcheckers[3210]: Error
reading data from remote SMTP server [192.168.200.1]:25.
[root@ubuntu1804 ~]#ps auxf |grep keepalived
root       3224  0.0  0.1  14428  1040 pts/0    S+   19:34   0:00      |   \_
grep --color=auto keepalived
root       3209  0.0  0.3  91812  2996 ?        Ss   19:33   0:00
/usr/sbin/keepalived
root       3210  0.0  0.5  96100  5276 ?        S    19:33   0:00  \_
/usr/sbin/keepalived
root       3211  0.0  0.5  96152  5420 ?        S    19:33   0:00  \_
/usr/sbin/keepalived
```

2.5.2 编译安装

```
#Ubuntu24.04,20.04和22.04安装相关包
[root@ubuntu2004 ~]#apt update  && apt -y install make gcc ipvsadm build-
essential pkg-config automake autoconf libipset-dev libnl-3-dev libnl-genl-3-dev
libssl-dev libxtables-dev libip4tc-dev libip6tc-dev libmagic-dev libsnmp-dev
libglib2.0-dev libpcre2-dev libnftnl-dev libmnl-dev libsystemd-dev

#Ubuntu18.04安装相关包
[root@ubuntu1804 ~]#apt update
[root@ubuntu1804 ~]#apt -y install gcc curl openssl libssl-dev libpopt-dev
daemon build-essential

#红帽系统安装相关包
[root@centos7 ~]#yum install  gcc curl openssl-devel libnl3-devel net-snmp-devel


#下载解压
[root@centos7 ~]#wget https://keepalived.org/software/keepalived-2.0.20.tar.gz

[root@centos7 ~]#tar xvf keepalived-2.0.20.tar.gz  -C /usr/local/src

[root@centos7 ~]#cd /usr/local/src/keepalived-2.0.20/

#选项--disable-fwmark 可用于禁用iptables规则,可防止VIP无法访问,无此选项默认会启用iptables
规则，注意：新版--disable-fwmark仍然会有iptables规则
[root@centos7 keepalived-2.0.20]#./configure --prefix=/usr/local/keepalived  #--
disable-fwmark仍然会有iptables规则
[root@centos7 keepalived-2.0.20]#make && make install
```

```
[root@centos7 keepalived-2.0.20]#cd
[root@centos7 ~]#/usr/local/keepalived/sbin/keepalived -v
Keepalived v2.0.20 (01/22,2020)

Copyright(C) 2001-2020 Alexandre Cassen, <acassen@gmail.com>

Built with kernel headers for Linux 3.10.0
Running on Linux 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC 2019

configure options: --prefix=/usr/local/keepalived

Config options:  LVS VRRP VRRP_AUTH OLD_CHKSUM_COMPAT FIB_ROUTING

System options:  PIPE2 SIGNALFD INOTIFY_INIT1 VSYSLOG EPOLL_CREATE1
IPV6_ADVANCED_API LIBNL3 RTA_ENCAP RTA_EXPIRES RTA_PREF FRA_SUPPRESS_PREFIXLEN
FRA_TUN_ID RTAX_CC_ALGO RTAX_QUICKACK FRA_OIFNAME IFA_FLAGS IP_MULTICAST_ALL
NET_LINUX_IF_H_COLLISION LIBIPTC_LINUX_NET_IF_H_COLLISION LIBIPVS_NETLINK
VRRP_VMAC IFLA_LINK_NETNSID CN_PROC SOCK_NONBLOCK SOCK_CLOEXEC O_PATH GLOB_BRACE
INET6_ADDR_GEN_MODE SO_MARK SCHED_RESET_ON_FORK

#下默认源码目录会自动生成unit文件
[root@centos7 keepalived-2.0.20]#cp ./keepalived/keepalived.service
/lib/systemd/system/
[root@centos7 ~]#cat /usr/lib/systemd/system/keepalived.service
[Unit]
Description=LVS and VRRP High Availability Monitor
After=network-online.target syslog.target
Wants=network-online.target

[Service]
Type=forking
PIDFile=/run/keepalived.pid
KillMode=process
EnvironmentFile=-/usr/local/keepalived/etc/sysconfig/keepalived
ExecStart=/usr/local/keepalived/sbin/keepalived $KEEPALIVED_OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target

[root@centos7 ~]#cat /usr/local/keepalived/etc/sysconfig/keepalived
# Options for keepalived. See `keepalived --help' output and keepalived(8) and
# keepalived.conf(5) man pages for a list of all options. Here are the most
# common ones :
#
# --vrrp               -P    Only run with VRRP subsystem.
# --check              -C    Only run with Health-checker subsystem.
# --dont-release-vrrp  -V    Dont remove VRRP VIPs & VROUTEs on daemon stop.
# --dont-release-ipvs  -I    Dont remove IPVS topology on daemon stop.
# --dump-conf          -d    Dump the configuration data.
# --log-detail         -D    Detailed log messages.
# --log-facility       -S    0-7 Set local syslog facility (default=LOG_DAEMON)
#

KEEPALIVED_OPTIONS="-D"

#默认无法启动
```

```
[root@centos7 ~]#systemctl start keepalived.service
Job for keepalived.service failed because the control process exited with error
code. See "systemctl status keepalived.service" and "journalctl -xe" for
details.

#查看日志，可以看到是因为缺少配置文件导致无法启动
[root@centos7 ~]#journalctl -xe
-- Subject: Unit keepalived.service has begun start-up
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit keepalived.service has begun starting up.
Mar 29 00:38:17 centos7.wang.org Keepalived[1123]: Starting Keepalived v2.0.20
(01/22,2020)
Mar 29 00:38:17 centos7.wang.org Keepalived[1123]: Running on Linux 3.10.0-
1062.el7.x86_64 #1 SMP Wed Aug 7
Mar 29 00:38:17 centos7.wang.org Keepalived[1123]: Command line:
'/usr/local/keepalived/sbin/keepalived' '-D
Mar 29 00:38:17 centos7.wang.org Keepalived[1123]: Unable to find configuration
file /etc/keepalived/keepali   #默认配置文件路径
Mar 29 00:38:17 centos7.wang.org Keepalived[1123]: Stopped Keepalived v2.0.20
(01/22,2020)
Mar 29 00:38:17 centos7.wang.org systemd[1]: keepalived.service: control process
exited, code=exited status=
Mar 29 00:38:17 centos7.wang.org systemd[1]: Failed to start LVS and VRRP High
Availability Monitor.
-- Subject: Unit keepalived.service has failed
-- Defined-By: systemd
-- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
--
-- Unit keepalived.service has failed.
--
-- The result is failed.
Mar 29 00:38:17 centos7.wang.org systemd[1]: Unit keepalived.service entered
failed state.
Mar 29 00:38:17 centos7.wang.org systemd[1]: keepalived.service failed.
Mar 29 00:38:17 centos7.wang.org polkitd[565]: Unregistered Authentication Agent
for unix-process:1117:11546

#创建配置文件，默认配置路径可以
是/usr/local/keepalived/etc/keepalived/keepalived.conf（优先高）或
者/etc/keepalived/keepalived.conf（优先级低）
[root@centos7 ~]#mkdir /etc/keepalived
[root@centos7 ~]#cp /usr/local/keepalived/etc/keepalived/keepalived.conf.sample
/etc/keepalived/keepalived.conf

#再次启动成功
[root@centos7 ~]#systemctl enable --now  keepalived.service
Created symlink from /etc/systemd/system/multi-
user.target.wants/keepalived.service to
/usr/lib/systemd/system/keepalived.service.

[root@centos7 ~]#systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; disabled; vendor
preset: disabled)
   Active: active (running) since Sun 2020-03-29 00:44:33 CST; 4s ago
```

```
  Process: 1191 ExecStart=/usr/local/keepalived/sbin/keepalived
$KEEPALIVED_OPTIONS (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/keepalived.service
           ├─1192 /usr/local/keepalived/sbin/keepalived -D
           ├─1193 /usr/local/keepalived/sbin/keepalived -D
           └─1194 /usr/local/keepalived/sbin/keepalived -D

Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.18
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.16
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.17
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.18
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.16
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.17
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.18
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.16
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.17
Mar 29 00:44:37 centos7.wang.org Keepalived_vrrp[1194]: Sending gratuitous ARP
on eth0 for 192.168.200.18
[root@centos7 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:32:80:38 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.7/24 brd 10.0.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.16/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.17/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.18/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe32:8038/64 scope link
       valid_lft forever preferred_lft forever

[root@centos7 ~]#hostname -I
10.0.0.7 192.168.200.16 192.168.200.17 192.168.200.18
[root@centos7 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 192.168.200.16 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms
```

```
#新版：nftables规则替代iptables规则
[root@ubuntu2204 ~]#apt update &&& apt -y install nftables
[root@ubuntu2204 ~]#nft list ruleset
table ip keepalived {
    set vips {
        type ipv4_addr
        elements = { 192.168.200.16, 192.168.200.17,
                 192.168.200.18 }
    }

    chain out {
        type filter hook output priority filter - 1; policy accept;
        ip saddr @vips drop
    }

    chain in {
        type filter hook input priority filter - 1; policy accept;
        ip daddr @vips drop
    }
}

#清空nft规则
[root@ubuntu2204 ~]#nft flush  ruleset
[root@ubuntu2204 ~]#nft list ruleset

#可以访问vip
[root@ubuntu2204 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
64 bytes from 192.168.200.16: icmp_seq=1 ttl=64 time=0.019 ms


#旧版：默认生成iptables规则,无法访问VIP,编译时可以加--disable-fwmark禁用生成iptables规则
[root@centos7 ~]#iptables -vnL
Chain INPUT (policy ACCEPT 860 packets, 46129 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.18
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.17
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.16

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 1737 packets, 1188K bytes)
 pkts bytes target     prot opt in     out     source               destination

    4   336 DROP       all  --  *      *       192.168.200.18       0.0.0.0/0

    0     0 DROP       all  --  *      *       192.168.200.17       0.0.0.0/0

    0     0 DROP       all  --  *      *       192.168.200.16       0.0.0.0/0

[root@centos7 ~]#vim /etc/keepalived/keepalived.conf
```

```
#注释下面一行
#vrrp_strict

#重启动不生效，有bug
[root@centos7 ~]#systemctl restart keepalived.service
[root@centos7 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 192.168.200.16 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms

[root@centos7 ~]#iptables -vnL
Chain INPUT (policy ACCEPT 1219 packets, 67647 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.18
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.17
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.16

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 2282 packets, 1233K bytes)
 pkts bytes target     prot opt in     out     source               destination

    4   336 DROP       all  --  *      *       192.168.200.18       0.0.0.0/0

    0     0 DROP       all  --  *      *       192.168.200.17       0.0.0.0/0

    4   336 DROP       all  --  *      *       192.168.200.16       0.0.0.0/0


#无法关闭进程
[root@centos7 ~]#systemctl stop keepalived.service
[root@centos7 ~]#ps aux|grep keepalived
root       1383  0.0  0.1  69672  1020 ?        Ss   00:57   0:00
/usr/local/keepalived/sbin/keepalived -D
root       1384  0.0  0.2  69804  2308 ?        S    00:57   0:00
/usr/local/keepalived/sbin/keepalived -D
root       1385  0.0  0.1  69672  1308 ?        S    00:57   0:00
/usr/local/keepalived/sbin/keepalived -D
root       1392  0.0  0.0 112712   964 pts/0    R+   00:59   0:00 grep --
color=auto keepalived

[root@centos7 ~]#killall  keepalived
[root@centos7 ~]#systemctl start keepalived.service
[root@centos7 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
64 bytes from 192.168.200.16: icmp_seq=1 ttl=64 time=0.093 ms
^C
--- 192.168.200.16 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

```
rtt min/avg/max/mdev = 0.093/0.093/0.093/0.000 ms
[root@centos7 ~]#iptables -vnL
Chain INPUT (policy ACCEPT 125 packets, 8493 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 135 packets, 20190 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

2.5.3 容器 Docker 安装

Docker的官方仓库上没有提供Keepalived的容器镜像，可以利用在源码目录中提供的Dockerfile自行制作

范例：定制镜像并启动

```
[root@ubuntu2204 ~]#mkdir -p /root/keepalived/docker
[root@ubuntu2204 ~]#wget https://keepalived.org/software/keepalived-2.2.8.tar.gz
[root@ubuntu2204 ~]#tar xf keepalived-2.2.8.tar.gz

[root@ubuntu2204 ~]#cat keepalived-2.2.8/Dockerfile
FROM alpine:latest
ARG GIT_VER=
ENV VER=2.2.8
LABEL version=2.2.8${GIT_VER}
LABEL author="Alexandre Cassen <acassen@gmail.com>"
LABEL project="https://github.com/acassen/keepalived"
LABEL homepage="https://www.keepalived.org"

# add keepalived sources to /tmp/keepalived-2.2.8
ADD keepalived-2.2.8.tar.gz /tmp

# Add keepalived default script user to make sure their IDs get assigned
consistently,
# regardless of whatever dependencies get added
RUN addgroup -S keepalived_script && adduser -D -S -G keepalived_script
keepalived_script

# 1. install required libraries and tools
# 2. compile and install keepalived
# 3. remove keepalived sources and unnecessary libraries and tools
RUN apk --no-cache add \
    binutils \
    file \
    file-dev \
    gcc \
#   glib \
#   glib-dev \
    ipset \
    ipset-dev \
    iptables \
    iptables-dev \
```

```
    libmnl-dev \
    libnftnl-dev \
    libnl3 \
    libnl3-dev \
    make \
    musl-dev \
#   net-snmp-dev \
    openssl \
    openssl-dev \
#   pcre2 \
#   pcre2-dev \
    autoconf \
    automake \
    && cd /tmp/keepalived-2.2.8/ \
    && ./autogen.sh \
    && ./configure \
        --disable-dynamic-linking \
        --prefix=/usr \
        --exec-prefix=/usr \
        --bindir=/usr/bin \
        --sbindir=/usr/sbin \
        --sysconfdir=/etc \
        --datadir=/usr/share \
        --localstatedir=/var \
        --mandir=/usr/share/man \
#       --with-dbus-data-dir=/usr/share \
        --enable-bfd \
#       --enable-dbus \
#       --enable-regex \
#       --enable-snmp \
#       --enable-snmp-rfc \
        --enable-nftables \
#       --disable-iptables \
#       --disable-libipset \
#       --enable-json \
    && make && make install \
    && strip /usr/sbin/keepalived \
    && cd - \
    && rm -rf /tmp/keepalived-2.2.8 \
    && apk --no-cache del \
    binutils \
    file-dev \
    gcc \
#   glib-dev \
    ipset-dev \
    iptables-dev \
    libmnl-dev \
    libnl3-dev \
    libnftnl-dev \
    make \
    musl-dev \
    openssl-dev \
#   pcre2-dev \
    autoconf \
    automake

ADD docker/keepalived.conf /etc/keepalived/keepalived.conf
```

```
# set keepalived as image entrypoint with --dont-fork and --log-console (to make it docker friendly)
# define /etc/keepalived/keepalived.conf as the configuration file to use
ENTRYPOINT ["/usr/sbin/keepalived","--dont-fork","--log-console", "-f","/etc/keepalived/keepalived.conf"]

# example command to customise keepalived daemon:
# CMD ["--log-detail","--dump-conf"]


[root@ubuntu2204 ~]#cp keepalived-2.2.8/doc/samples/keepalived.conf.sample /root/keepalived/docker/keepalived.conf
[root@ubuntu2204 ~]#cp keepalived-2.2.8.tar.gz  /root/keepalived/
[root@ubuntu2204 ~]#cp keepalived-2.2.8/Dockerfile /root/keepalived/
[root@ubuntu2204 ~]#cd /root/keepalived/

[root@ubuntu2204 keepalived]#pwd
/root/keepalived

[root@ubuntu2204 keepalived]#tree
.
├── docker
│   └── keepalived.conf
├── Dockerfile
└── keepalived-2.2.8.tar.gz

1 directory, 3 files

#制作镜像
[root@ubuntu2204 keepalived]#docker build -t wangxiaochun/keepalived:2.2.8-alpine .

#上传镜像
[root@ubuntu2204 keepalived]#docker login
[root@ubuntu2204 keepalived]#docker push wangxiaochun/keepalived:2.2.8-alpine

#创建和定制配置文件（可选）
[root@ubuntu2204 ~]#vim ./keepalived.conf

#启动容器
[root@ubuntu2204 ~]#docker run -d --name mykeepalived  --restart=always  -v ./keepalived.conf:/etc/keepalived/keepalived.conf --privileged --network host wangxiaochun/keepalived:2.2.8-alpine

[root@ubuntu2204 ~]#docker logs mykeepalived
Wed Nov 22 03:11:07 2023: Starting Keepalived v2.2.8 (04/04,2023), git commit v2.2.7-154-g292b299e+
Wed Nov 22 03:11:07 2023: WARNING - keepalived was built for newer Linux 6.3.0, running on Linux 5.15.0-52-generic #58-Ubuntu SMP Thu Oct 13 08:03:55 UTC 2022
Wed Nov 22 03:11:07 2023: Command line: '/usr/sbin/keepalived' '--dont-fork' '--log-console' '-f'
Wed Nov 22 03:11:07 2023:                '/etc/keepalived/keepalived.conf'
Wed Nov 22 03:11:07 2023: Configuration file /etc/keepalived/keepalived.conf
Wed Nov 22 03:11:07 2023: NOTICE: setting config option max_auto_priority should result in better keepalived performance
Wed Nov 22 03:11:07 2023: Starting Healthcheck child process, pid=8
Wed Nov 22 03:11:07 2023: Starting VRRP child process, pid=9
Wed Nov 22 03:11:07 2023: Initializing ipvs
```

```
Wed Nov 22 03:11:07 2023: Gained quorum 1+0=1 <= 1 for VS [10.10.10.2]:tcp:1358
Wed Nov 22 03:11:07 2023: Activating healthchecker for service
[192.168.200.2]:tcp:1358 for VS [10.10.10.2]:tcp:1358
Wed Nov 22 03:11:07 2023: Activating BFD healthchecker
Wed Nov 22 03:11:07 2023: Startup complete
Wed Nov 22 03:11:07 2023: (VI_1) Entering BACKUP STATE (init)
Wed Nov 22 03:11:11 2023: (VI_1) Entering MASTER STATE
Wed Nov 22 03:11:13 2023: HTTP_CHECK on service [192.168.200.2]:tcp:1358 failed
after 3 retries.
Wed Nov 22 03:11:13 2023: Removing service [192.168.200.2]:tcp:1358 from VS
[10.10.10.2]:tcp:1358
Wed Nov 22 03:11:13 2023: Lost quorum 1-0=1 > 0 for VS [10.10.10.2]:tcp:1358
Wed Nov 22 03:11:13 2023: Adding sorry server [192.168.200.200]:tcp:1358 to VS
[10.10.10.2]:tcp:1358
Wed Nov 22 03:11:13 2023: Removing alive servers from the pool for VS
[10.10.10.2]:tcp:1358
Wed Nov 22 03:11:34 2023: smtp fd 10 returned write error


[root@ubuntu2204 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:0c:29:b4:43:e9 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname ens33
    inet 10.0.0.200/24 brd 10.0.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.11/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.12/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.13/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb4:43e9/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group default
    link/ether 02:42:e1:61:9c:e3 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:e1ff:fe61:9ce3/64 scope link
       valid_lft forever preferred_lft forever

[root@ubuntu2204 ~]#ping 192.168.200.11 -c1
PING 192.168.200.11 (192.168.200.11) 56(84) bytes of data.
64 bytes from 192.168.200.11: icmp_seq=1 ttl=64 time=0.022 ms

--- 192.168.200.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.022/0.022/0.022/0.000 ms
```

2.6 KeepAlived 配置说明

2.6.1 配置文件组成部分

配置文件

```
/etc/keepalived/keepalived.conf
```

配置文件组成

- GLOBAL CONFIGURATION

  Global definitions：定义邮件配置，route_id，vrrp配置，多播地址等

- VRRP CONFIGURATION

  VRRP instance(s)：定义每个vrrp虚拟路由器

- LVS CONFIGURATION

  Virtual server group(s)

  Virtual server(s)：LVS集群的VS和RS

2.6.2 配置语法说明

帮助

```
#包安装
man keepalived.conf
#编译安装
man /apps/keepalived/share/man/man5/keepalived.conf.5
```

2.6.2.1 全局配置

```
#/etc/keepalived/keepalived.conf
global_defs {
  notification_email {
  root@localhost              #keepalived 发生故障切换时邮件发送的目标邮箱，可以按行区分写多个
  root@wangxiaochun.com
  29308620@qq.com
  }
  notification_email_from keepalived@localhost  #发邮件的地址
  smtp_server 127.0.0.1      #邮件服务器地址
  smtp_connect_timeout 30    #邮件服务器连接timeout
  router_id ka1.example.com #每个keepalived主机唯一标识，建议使用当前主机名，如果多节点重名可能会影响切换脚本执行
  vrrp_skip_check_adv_addr  #默认会对所有通告报文都检查，会比较消耗性能，启用此配置后，如果收到的通告报文和上一个报文是同一个路由器，则跳过检查
  vrrp_strict                #严格遵守VRRP协议,启用此项后以下状况将无法启动服务或工作异常:1.无VIP地址 2.配置了单播邻居 3.在VRRP版本2中有IPv6地址，开启动此项并且没有配置vrrp_iptables时会自动开启iptables(旧内核)或者nft(新内核)的防火墙规则，默认导致VIP无法访问，建议不加此项配置
  vrrp_garp_interval 0       #gratuitous 免费 ARP messages 报文发送延迟，0表示不延迟
  vrrp_gna_interval 0        #unsolicited NA messages （不请自来）消息发送延迟
  vrrp_mcast_group4 224.0.0.18 #指定组播IP地址范围：224.0.0.0到239.255.255.255,默认值：224.0.0.18，如果配置了单播，此项失效
```

```
  vrrp_iptables    #此项和vrrp_strict同时开启时，则不会添加防火墙规则,如果无配置
vrrp_strict项,则无需启用此项配置，注意：新版加此项仍有iptables(旧内核)或者nft(新内核)规则
}
```

2.6.2.2 配置虚拟路由器

```
vrrp_instance <STRING> { #<String>为vrrp的实例名,一般为业务名称
    配置参数
    ......
 }

#配置参数：
state MASTER|BACKUP#当前节点在此虚拟路由器上的初始状态，状态为MASTER或者BACKUP，当
priority相同时，先启动的节点优先获取VIP
interface IFACE_NAME #绑定为当前VRRP虚拟路由器使用的物理接口，如：eth0,bond0,br0,可以和
VIP不在一个网卡，实现心跳功能

virtual_router_id VRID #每个虚拟路由器唯一标识，范围：0-255，每个虚拟路由器此值必须唯一，否
则服务无法启动，同属一个虚拟路由器的多个keepalived节点必须相同,务必要确认在同一网络中此值必须唯
一

priority 100     #当前物理节点在此虚拟路由器的优先级，范围：1-254，每个keepalived主机节点此
值不同，如果多节点此值相同，则先来后到原理获取VIP
advert_int 1     #vrrp通告的时间间隔，默认1s，注意：集群内多节点此值必须相同
authentication { #认证机制
    auth_type AH|PASS    #AH为IPSEC认证(不推荐),PASS为简单密码(建议使用)
    auth_pass <PASSWORD> #预共享密钥，仅前8位有效，同一个虚拟路由器的多个keepalived节点必
须一样
}
virtual_ipaddress { #虚拟IP,生产环境可能指定几十上百个VIP地址
    <IPADDR>/<MASK> brd <IPADDR> dev <STRING> scope <SCOPE> label <LABEL>
    192.168.200.100          #指定VIP，不指定网卡，默认为eth0,注意：不指定/prefix,默认
为/32
    192.168.200.101/24 dev eth1    #指定VIP的网卡，建议和interface指令指定的网卡不在一个
网卡
    192.168.200.102/24 dev eth2 label eth2:1     #指定VIP的网卡label
}
track_interface { #配置监控网络接口，一旦出现故障，则转为FAULT状态实现地址转移
    eth0
    eth1
    …
}
```

范例：

```
[root@centos7 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
```

```
   smtp_connect_timeout 30
   router_id LVS_DEVEL
   vrrp_skip_check_adv_addr
   vrrp_strict                  #开启限制，会自动生效防火墙设置，导致无访问VIP
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 80 #修改此行
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        192.168.200.16
        192.168.200.17
        192.168.200.18
    }
}
[root@centos7 ~]#systemctl start keepalived.service
[root@centos7 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:33:b4:1a brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.17/24 brd 10.0.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.16/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.17/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet 192.168.200.18/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe33:b41a/64 scope link
       valid_lft forever preferred_lft forever


[root@centos7 ~]#iptables -vnL
Chain INPUT (policy ACCEPT 59 packets, 3372 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.16
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.17
    0     0 DROP       all  --  *      *       0.0.0.0/0
 192.168.200.18
```

```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 33 packets, 6940 bytes)
 pkts bytes target     prot opt in     out     source               destination


[root@centos7 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
^C
--- 192.168.200.16 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5002ms

[root@centos7 ~]#


# 如果是CentOS 8 ，会显示以下warning
[root@centos8 ~]#iptables -vnL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

# Warning: iptables-legacy tables present, use iptables-legacy to see them

#无法访问VIP
[root@centos8 ~]#ping 192.168.200.16
PING 192.168.200.16 (192.168.200.16) 56(84) bytes of data.
^C
--- 192.168.200.16 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 143ms
```

2.7 启用 Keepalived 日志功能

默认 keepalived的日志记录在LOG_DAEMON中，记录在/var/log/syslog或messages, 也支持自定义日志配置

范例：实现日志功能,

注意: 编译安装方式如果实现有问题，可以重启主机可以解决

```
#红帽系统包安装修改文件路径
[root@ka1 ~]#vim /etc/sysconfig/keepalived
KEEPALIVED_OPTIONS="-D -S 6"

#编译安装修改文件路径
[root@ka1 ~]#grep ExecStart /lib/systemd/system/keepalived.service
ExecStart=/apps/keepalived/sbin/keepalived --dont-fork  $KEEPALIVED_OPTIONS
```

```
#编译安装修改文件路径
[root@ka1 ~]#vim /apps/keepalived/etc/sysconfig/keepalived
# Options for keepalived. See `keepalived --help' output and keepalived(8) and
# keepalived.conf(5) man pages for a list of all options. Here are the most
# common ones :
#
# --vrrp               -P    Only run with VRRP subsystem.
# --check              -C    Only run with Health-checker subsystem.
# --dont-release-vrrp  -V    Dont remove VRRP VIPs & VROUTEs on daemon stop.
# --dont-release-ipvs  -I    Dont remove IPVS topology on daemon stop.
# --dump-conf          -d    Dump the configuration data.
# --log-detail         -D    Detailed log messages.
# --log-facility       -S    0-7 Set local syslog facility (default=LOG_DAEMON)
#
#修改下面行
KEEPALIVED_OPTIONS="-D -S 6"

#修改日志服务配置
[root@ka1 ~]#vim /etc/rsyslog.conf
local6.*             /var/log/keepalived.log

#重启服务生效
[root@ka1 ~]#systemctl restart keepalived.service rsyslog.service

#如果不行，就重启主机
[root@ka1 ~]#reboot

[root@ka1 ~]#tail -f /var/log/keepalived.log
Apr 14 09:25:51 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:51 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:51 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:51 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: (VI_1) Sending/queueing gratuitous
ARPs on eth0 for 10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
Apr 14 09:25:56 ka1 Keepalived_vrrp[1263]: Sending gratuitous ARP on eth0 for
10.0.0.10
```

2.8 实现 Keepalived 独立子配置文件

当生产环境复杂时，`/etc/keepalived/keepalived.conf`文件中保存所有集群的配置会导致内容过多，不易管理

可以将不同集群的配置，比如：不同集群的VIP配置放在独立的子配置文件中

利用include 指令可以实现包含子配置文件

格式:

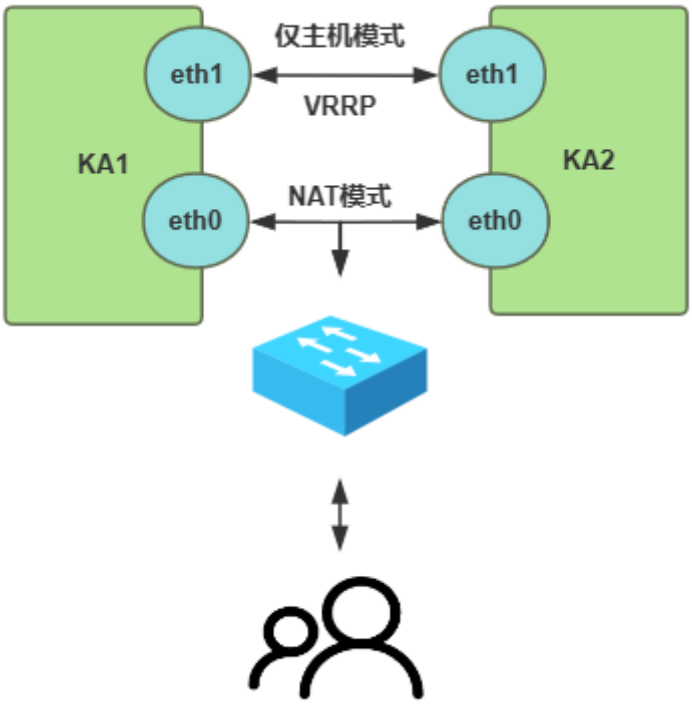
```
include /path/file
```

范例:

```
[root@ka1 ~]#mkdir /etc/keepalived/conf.d/
[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
global_defs {
   notification_email {
        29308620@qq.com
   }
   notification_email_from 29308620@qq.com
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   router_id ka1.wang.org
   vrrp_skip_check_adv_addr
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}
include /etc/keepalived/conf.d/*.conf    #将VRRP相关配置放在子配置文件中

[root@ka1 ~]#vim /etc/keepalived/conf.d/cluster1.conf
```

3 Keepalived 实现 VRRP

3.1 实现Master/Backup的 Keepalived 单主架构



3.1.1 MASTER配置

```
[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
global_defs {
  notification_email {
  root@localhost #keepalived 发生故障切换时邮件发送的对象，可以按行区分写多个
  }
  notification_email_from keepalived@localhost
  smtp_server 127.0.0.1
  smtp_connect_timeout 30
  router_id ka1.example.com
  vrrp_skip_check_adv_addr #所有报文都检查比较消耗性能，此配置为如果收到的报文和上一个报文是同一个路由器则跳过检查报文中的源地址
  #vrrp_strict #严格遵守VRRP协议,禁止状况:1.无VIP地址,2.配置了单播邻居,3.在VRRP版本2中有IPv6地址
  vrrp_garp_interval 0 #ARP报文发送延迟
  vrrp_gna_interval 0 #消息发送延迟
  vrrp_mcast_group4 224.0.0.18 #默认组播IP地址，可指定组播范围：224.0.0.0到239.255.255.255
 }
vrrp_instance VI_1 {
  state MASTER            #在另一个节点上为BACKUP,如果当priority相同时,先启动的节点优先获取VIP
  interface eth0
  virtual_router_id 66    #每个虚拟路由器必须唯一，同属一个虚拟路由器的多个keepalived节点必须相同
  priority 100            #在另一个结点上为80
  advert_int 1
  authentication {
    auth_type PASS        #预共享密钥认证，同一个虚拟路由器的keepalived节点必须一样
    auth_pass 12345678
  }
  virtual_ipaddress {
    10.0.0.10 dev eth0 label eth0:0
  }
}
```

3.1.2 BACKUP配置

```
#配置文件和master基本一致，只需修改三行
[root@ka2 ~]#vim /etc/keepalived/keepalived.conf
global_defs {
  notification_email {
  root@localhost
  }
  notification_email_from keepalived@localhost
  smtp_server 127.0.0.1
  smtp_connect_timeout 30
  router_id ka2.example.com          #修改此行
  vrrp_skip_check_adv_addr
  #vrrp_strict
  vrrp_garp_interval 0
  vrrp_gna_interval 0
  vrrp_mcast_group4 224.0.0.18
}
vrrp_instance VI_1 {
```

```
    state BACKUP                #修改此行
    interface eth0
    virtual_router_id 66
    priority 80                 #修改此行
    advert_int 1
    authentication {
      auth_type PASS
      auth_pass 12345678
    }
    virtual_ipaddress {
      10.0.0.10 dev eth0 label eth0:0
    }
}
```

抓包观察VRRP协议

```
tcpdump -i eth0 -nn host 224.0.0.18
```

VRRP 协议包构成

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)
2	1.000337	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)
3	2.001403	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)
4	3.002394	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)
5	4.002749	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)
6	5.003789	192.168.10.202	224.1.1.1	VRRP	60	Announcement (v2)

```
Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{4DD740AC-3567-41F3-A635-7F7ADAAADB47}, id 0
Ethernet II, Src: VMware_a6:5e:49 (00:0c:29:a6:5e:49), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01)
Internet Protocol Version 4, Src: 192.168.10.202, Dst: 224.1.1.1
Virtual Router Redundancy Protocol
  Version 2, Packet type 1 (Advertisement)
      0010 .... = VRRP protocol version: 2
      .... 0001 = VRRP packet type: Advertisement (1)
    Virtual Rtr ID: 66
    Priority: 80 (Non-default backup priority)
    Addr Count: 1
    Auth Type: Simple Text Authentication [RFC 2338] / Reserved [RFC 3768] (1)
    Adver Int: 1
    Checksum: 0xe956 [correct]
    [Checksum Status: Good]
    IP Address: 10.0.0.200
    Authentication String: 123456
```

```
000  01 00 5e 01 01 01 00 0c  29 a6 5e 49 08 00 45 c0   ··^····· )·^I··E·
010  00 28 00 df 00 00 ff 70  0d 52 c0 a8 0a ca e0 01   ·(·····p ·R······
020  01 01 21 42 50 01 01 01  e9 56 0a 00 00 c8 31 32   ··!BP··· ·V····12
030  33 34 35 36 00 00 00 00  00 00 00 00               3456···· ····
```

3.2 抢占模式和非抢占模式

3.2.1 非抢占模式 nopreempt

默认为抢占模式 preempt，即当高优先级的主机恢复在线后，会抢占低先级的主机的master角色，造成网络抖动，建议设置为非抢占模式 nopreempt ，即高优先级主机恢复后，并不会抢占低优先级主机的 master 角色

注意: 非抢占模式下,如果原主机down机, VIP迁移至的新主机, 后续新主机也发生down （（keepalived服务down））时,VIP还会迁移回修复好的原主机

但如果新主机的服务down掉（keepalived服务正常），原主机也不会接管VIP，仍会由新主机拥有VIP

即非抢占式模式，只是适合当主节点宕机，切换到从节点的一次性的高可用性，后续即使当原主节点修复好，仍无法再次起到高用功能

注意：要关闭 VIP抢占，必须将各 Keepalived 服务器 state 配置为 BACKUP

```
#ha1主机配置
vrrp_instance VI_1 {
  state BACKUP                #都为BACKUP
  interface eth0
  virtual_router_id 66
  priority 100                #优先级高
  advert_int 1
  nopreempt                   #添加此行，设为nopreempt

#ha2主机配置
vrrp_instance VI_1 {
  state BACKUP                #都为BACKUP
  interface eth0
  virtual_router_id 66
  priority 80                 #优先级低
  advert_int 1
  #nopreempt                  #注意：如果ka2主机也是非抢占式，会导致ka1即使优先级降低于
ka2，VIP也不会切换至ka2
```

3.2.2 抢占延迟模式 preempt_delay

抢占延迟模式，即优先级高的主机恢复后，不会立即抢回VIP，而是延迟一段时间（默认300s）再抢回VIP

但是如果低优先级的主机down机，则立即抢占VIP地址，而不再延迟

```
preempt_delay    #       #指定抢占延迟时间为#s，默认延迟300s，在优先级高的节点配置
```

注意：需要各keepalived服务器state为BACKUP,并且不要启用 vrrp_strict

范例：

```
#ka1主机配置
vrrp_instance VI_1 {
  state BACKUP                        #都为BACKUP
  interface eth0
  virtual_router_id 66
  priority 100                        #优先级高
  advert_int 1
  preempt_delay 60                    #抢占延迟模式，默认延迟300s

#ka2主机配置
vrrp_instance VI_1 {
  state BACKUP                        #都为BACKUP
  interface eth0
  virtual_router_id 66
  priority 80                         #优先级低
  advert_int 1
```

3.3 VIP 单播配置

默认keepalived主机之间利用多播相互通告消息，会造成网络拥塞，可以设置为单播，减少网络流量

另外：有些公有云不支持多播，可以利用单播实现

单播优先与多播，即同时配置，单播生效

注意：启用 vrrp_strict 时，不能启用单播

```
#在所有节点vrrp_instance语句块中设置对方主机的IP，建议设置为专用于对应心跳线网络的地址，而非使用业务网络
unicast_src_ip <IPADDR>  #指定发送单播的源IP
unicast_peer {
    <IPADDR>             #指定接收单播的对方目标主机IP
    ......
}
```

范例：

```
#master 主机配置
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id ka1.wang.org
   vrrp_skip_check_adv_addr
   #vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
   vrrp_mcast_group4 239.0.0.0  #单播优先于多播,即配置了单播后，多播将失效
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 66
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.8     #本机IP
    unicast_peer{
        10.0.0.18               #指向对方主机IP
        10.0.0.28               #如果有多个keepalived,再加其它节点的IP
    }
}

[root@ha1-centos8 ~]#hostname -I
10.0.0.8 10.0.0.10
```

```
#slave 主机配置
[root@ka2 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id ka2.wang.org
   vrrp_skip_check_adv_addr
   #vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
   vrrp_mcast_group4 239.0.0.0  #单播优先于多播
}

vrrp_instance VI_1 {
    state SLAVE
    interface eth0
    virtual_router_id 66
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.18        #本机IP
    unicast_peer {
        10.0.0.8                    #指向对方主机IP
    }
}
[root@ka2 ~]#hostname -I
10.0.0.18
```

范例: 抓包观察

```
root@centos6 ~]#tcpdump  -i eth0 -nn src host 10.0.0.8 and dst host 10.0.0.18
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:37:48.069158 IP 10.0.0.8 > 10.0.0.18: VRRPv2, Advertisement, vrid 66, prio
100, authtype simple, intvl 1s, length 20
23:37:49.070013 IP 10.0.0.8 > 10.0.0.18: VRRPv2, Advertisement, vrid 66, prio
100, authtype simple, intvl 1s, length 20
23:37:50.071144 IP 10.0.0.8 > 10.0.0.18: VRRPv2, Advertisement, vrid 66, prio
100, authtype simple, intvl 1s, length 20
```

3.4 Keepalived 通知脚本配置

当keepalived的状态变化时，可以自动触发脚本的执行，比如：发邮件通知用户

默认以用户keepalived_script身份执行脚本，如果此用户不存在，以root执行脚本

可以用下面指令指定脚本执行用户的身份

```
global_defs {
    ......
    script_user <USER>
    ......
}
```

3.4.1 通知脚本类型

- 当前节点成为主节点时触发的脚本

```
notify_master <STRING>|<QUOTED-STRING>
```

- 当前节点转为备节点时触发的脚本

```
notify_backup <STRING>|<QUOTED-STRING>
```

- 当前节点转为“失败”状态时触发的脚本

```
notify_fault <STRING>|<QUOTED-STRING>
```

- 通用格式的通知触发机制，一个脚本可完成以上三种状态的转换时的通知

```
notify <STRING>|<QUOTED-STRING>
```

- 当停止VRRP时触发的脚本

```
notify_stop <STRING>|<QUOTED-STRING>
```

3.4.2 脚本的调用方法

在 vrrp_instance VI_1 语句块的末尾加下面行

```
notify_master "/etc/keepalived/notify.sh master"
notify_backup "/etc/keepalived/notify.sh backup"
notify_fault "/etc/keepalived/notify.sh fault"
```

3.4.3 实战案例1：实现 Keepalived 状态切换的通知脚本

以下脚本支持RHEL和Ubuntu系统

```
#在所有 keepalived节点配置如下
[root@ka1 ~]#cat /etc/keepalived/notify.sh
#!/bin/bash
#
#**********************************************************************
#Author:            wangxiaochun
```

```
#QQ:                29308620
#Date:              2020-02-31
#FileName:          notify.sh
#URL:               http://www.wangxiaochun.com
#Description:       The test script
#Copyright (C):     2020 All rights reserved
#**********************************************************************

contact='root@wangxiaochun.com'
email_send='29308620@qq.com'
email_passwd='dgezyimkdswwbhea'
email_smtp_server='smtp.qq.com'

. /etc/os-release

msg_error() {
  echo -e "\033[1;31m$1\033[0m"
}

msg_info() {
  echo -e "\033[1;32m$1\033[0m"
}

msg_warn() {
  echo -e "\033[1;33m$1\033[0m"
}

color () {
    RES_COL=60
    MOVE_TO_COL="echo -en \\033[${RES_COL}G"
    SETCOLOR_SUCCESS="echo -en \\033[1;32m"
    SETCOLOR_FAILURE="echo -en \\033[1;31m"
    SETCOLOR_WARNING="echo -en \\033[1;33m"
    SETCOLOR_NORMAL="echo -en \E[0m"
    echo -n "$1" && $MOVE_TO_COL
    echo -n "["
    if [ $2 = "success" -o $2 = "0" ] ;then
        ${SETCOLOR_SUCCESS}
        echo -n $"  OK  "
    elif [ $2 = "failure" -o $2 = "1"  ] ;then
        ${SETCOLOR_FAILURE}
        echo -n $"FAILED"
    else
        ${SETCOLOR_WARNING}
        echo -n $"WARNING"
    fi
    ${SETCOLOR_NORMAL}
    echo -n "]"
    echo
}


install_sendemail () {
    if [[ $ID =~ rhel|centos|rocky ]];then
        rpm -q sendemail &> /dev/null ||  yum install -y sendemail
    elif [ $ID = 'ubuntu' ];then
        dpkg -l |grep -q sendemail  || { apt update; apt install -y libio-
socket-ssl-perl libnet-ssleay-perl sendemail ; }
```

```
    else
        color "不支持此操作系统，退出!" 1
        exit
    fi
}

send_email () {
    local email_receive="$1"
    local email_subject="$2"
    local email_message="$3"
    sendemail -f $email_send -t $email_receive -u $email_subject -m
$email_message -s $email_smtp_server -o message-charset=utf-8 -o tls=yes -xu
$email_send -xp $email_passwd
    [ $? -eq 0 ] && color "邮件发送成功!" 0 || color "邮件发送失败!" 1
}

notify() {
    if [[ $1 =~ ^(master|backup|fault)$ ]];then
        mailsubject="$(hostname) to be $1, vip floating"
        mailbody="$(date +'%F %T'): vrrp transition, $(hostname) changed to be
$1"
        send_email "$contact" "$mailsubject" "$mailbody"
   else
        echo "Usage: $(basename $0) {master|backup|fault}"
        exit 1
   fi
}

install_sendemail
notify $1


[root@ka1 ~]#chmod a+x /etc/keepalived/notify.sh

[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
vrrp_instance VI_1 {
    ......
    virtual_ipaddress {
        10.0.0.10 dev eth0 label eth0:1
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}

#模拟master故障
[root@ka1 ~]#killall keepalived
```

3.4.4 实战案例2：实现 Keepalived 状态切换的通知脚本

下面仅支持RHEL系统

3.4.4.1 邮件配置

案例：QQ邮箱配置

```
[root@centos8 ~]# vim  /etc/mail.rc
#在最后面添加下面行
set from=29308620@qq.com
set smtp=smtp.qq.com
set smtp-auth-user=29308620@qq.com
set smtp-auth-password=esvnhbnqocirbicf
set smtp-auth=login
set ssl-verify=ignore
```

范例：163 邮箱配置

```
[root@centos8 ~]#vi /etc/mail.rc
set from=xxx@163.com #之前设置好的邮箱地址
set smtp=smtp.163.com #邮件服务器
set smtp-auth-user=xxx@163.com #之前设置好的邮箱地址
set smtp-auth-password=QXFIOQXEJNSVSDM #授权码
set smtp-auth=login #默认login即可
```

范例：发送测试邮件

```
[root@centos8 ~]# yum -y install mailx
[root@centos8 ~]# echo "Test Mail"| mail  -s Warning  root@wangxiaochun.com
```

3.4.4.2 创建通知脚本

```
#在所有 keepalived节点配置如下
[root@ka1 ~]#cat /etc/keepalived/notify.sh
#!/bin/bash
#
contact='root@wangxiaochun.com'
notify() {
    mailsubject="$(hostname) to be $1, vip floating"
    mailbody="$(date +'%F %T'): vrrp transition, $(hostname) changed to be $1"
    echo "$mailbody" | mail -s "$mailsubject" $contact
}
case $1 in
master)
    notify master
    ;;
backup)
    notify backup
    ;;
fault)
    notify fault
    ;;
*)
    echo "Usage: $(basename $0) {master|backup|fault}"
    exit 1
    ;;
esac
```

```
[root@ka1 ~]#chmod a+x /etc/keepalived/notify.sh

[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
vrrp_instance VI_1 {
    ......
    virtual_ipaddress {
        10.0.0.10 dev eth0 label eth0:1
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}

#模拟master故障
[root@ka1 ~]#killall keepalived
```

查看邮箱收到邮件如下：

ka2-centos8.magedu.org to be master, vip floating

20 29308620

2020-03-28 12:15 邮件参与人>

发件人为企业外人员

2020-03-28 12:15:19: vrrp transition, ka2-centos8.magedu.org changed to be master

3.5 脑裂现象

主备节点同时拥有同一个VIP，此时为脑裂现象

注意：脑裂现象原因

- 心跳线故障： 注意:在虚拟机环境中测试可以通过修改网卡的工作模式实现模拟，断开网卡方式无法模拟
- 防火墙错误配置：在从节点服务器执行iptables -A INPUT -s 主服务心跳网卡IP -j DROP 进行模拟
- Keepalived 配置错误：多播或单播地址不同，interface错误，virtual_router_id不一致，密码不一致

范例: 发现脑裂

```
[root@centos7 ~]#arping  -I eth1 -c1 10.0.0.100
ARPING 10.0.0.100 from 192.168.10.130 eth1
Unicast reply from 10.0.0.100 [00:0C:29:7E:DA:E6]  0.801ms
Unicast reply from 10.0.0.100 [00:0C:29:97:AF:4F]  0.814ms
Sent 1 probes (1 broadcast(s))
Received 2 response(s)
```

3.6 实现 Master/Master 的 Keepalived 双主架构

master/slave的单主架构，同一时间只有一个Keepalived对外提供服务，此主机繁忙，而另一台主机却很空闲，利用率低下，可以使用master/master的双主架构，解决此问题。

Master/Master 的双主架构：

即将两个或以上VIP分别运行在不同的keepalived服务器，以实现服务器并行提供web访问的目的，提高服务器资源利用率

```
#ha1主机配置
[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
    notification_email {
        root@wangxiaochun.com
    }
    notification_email_from keepalived@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org
    vrrp_mcast_group4 224.0.100.100
}

vrrp_instance VI_1 {
    state MASTER            #在另一个主机上为BACKUP
    interface eth0
    virtual_router_id 66    #每个vrrp_instance唯一
    priority 100            #在另一个主机上为80
```

```
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 12345678
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1 #指定vrrp_instance各自的VIP
    }
}
vrrp_instance VI_2 {            #添加 VI_2 实例
    state BACKUP                #在另一个主机上为MASTER
    interface eth0
    virtual_router_id 88        #每个vrrp_instance唯一
    priority 80                 #在另一个主机上为100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 12345678
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:1  #指定vrrp_instance各自的VIP
    }
}

#ka2主机配置,和ka1配置只需五行不同
[root@ka2 ~]#vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
global_defs {
    notification_email {
        root@wangxiaochun.com
    }
    notification_email_from keepalived@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka2.wang.org        #修改此行
    vrrp_mcast_group4 224.0.100.100
}

vrrp_instance VI_1 {
    state BACKUP                    #此修改行为BACKUP
    interface eth0
    virtual_router_id 66
    priority 80                     #此修改行为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 12345678
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
}
vrrp_instance VI_2 {
    state MASTER                    #修改此行为MASTER
    interface eth0
    virtual_router_id 88
    priority 100                    #修改此行为100
    advert_int 1
```

```
    authentication {
        auth_type PASS
        auth_pass 12345678
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:1
    }
}
```

实战案例：利用子配置文件实现master/master的Keepalived双主架构

```
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id ha1.wang.org
   vrrp_skip_check_adv_addr
   #vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0

}

include /etc/keepalived/conf.d/*.conf

[root@ka1 ~]#mkdir /etc/keepalived/conf.d/
[root@ka1 ~]#cat /etc/keepalived/conf.d/cluster1.conf
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 66
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.8
    unicast_peer{
        10.0.0.18
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}
```

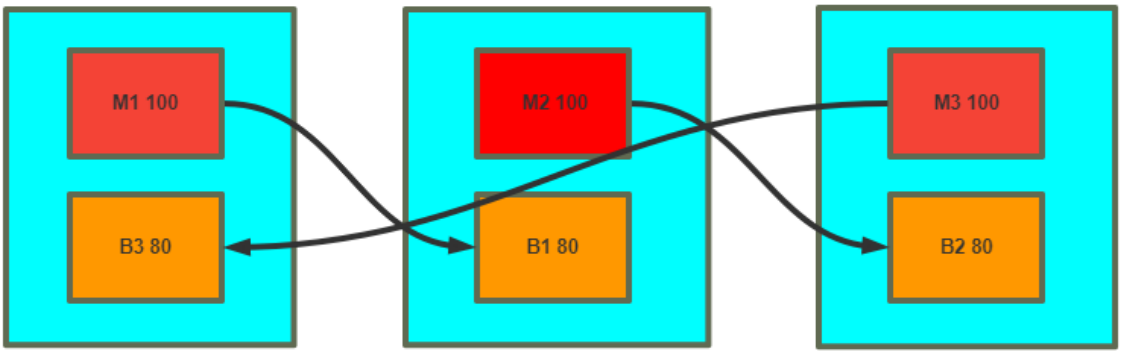
```
[root@ka1 ~]#cat /etc/keepalived/conf.d/cluster2.conf
vrrp_instance VI_2 {
    state BACKUP
    interface eth0
    virtual_router_id 88
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.8
    unicast_peer{
        10.0.0.18
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}

[root@ka1 ~]#tree /etc/keepalived/
/etc/keepalived/
├── conf.d
|   ├── cluster1.conf
|   └── cluster2.conf
├── keepalived.conf
├── keepalived.conf.bak
└── notify.sh

1 directory, 5 files
[root@ka1 ~]#


#ka2主机的配置
[root@ka2 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id ha2.wang.org
   vrrp_skip_check_adv_addr
   #vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}

include /etc/keepalived/conf.d/*.conf
```

```
[root@ka2 ~]#cat /etc/keepalived/conf.d/cluster1.conf
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 66
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.18
    unicast_peer {
    10.0.0.8
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}
[root@ka2 ~]#cat /etc/keepalived/conf.d/cluster2.conf
vrrp_instance VI_2 {
    state MASTER
    interface eth0
    virtual_router_id 88
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:1
    }
    unicast_src_ip 10.0.0.18
    unicast_peer{
        10.0.0.8
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}

[root@ka2 ~]#


#查看IP
[root@ka1 ~]#hostname -I
10.0.0.8 10.0.0.10
[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.20

#ka1主机故障，测试VIP漂移至ka2主机
[root@ka1 ~]#killall keepalived
[root@ka1 ~]#hostname -I
10.0.0.8
```

```
[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.20 10.0.0.10

#恢复ka1主机
[root@ka1 ~]#systemctl start keepalived.service
[root@ka1 ~]#hostname -I
10.0.0.8 10.0.0.10
[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.20

#脑裂现象
[root@ka1 ~]#hostname -I
10.0.0.8 10.0.0.10
[root@ka1 ~]#iptables -A INPUT -s10.0.0.18 -j DROP
[root@ka1 ~]#hostname -I
10.0.0.8 10.0.0.10 10.0.0.20

[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.20
[root@ka2 ~]#iptables -A INPUT -s 10.0.0.8 -j REJECT
[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.20 10.0.0.10
```

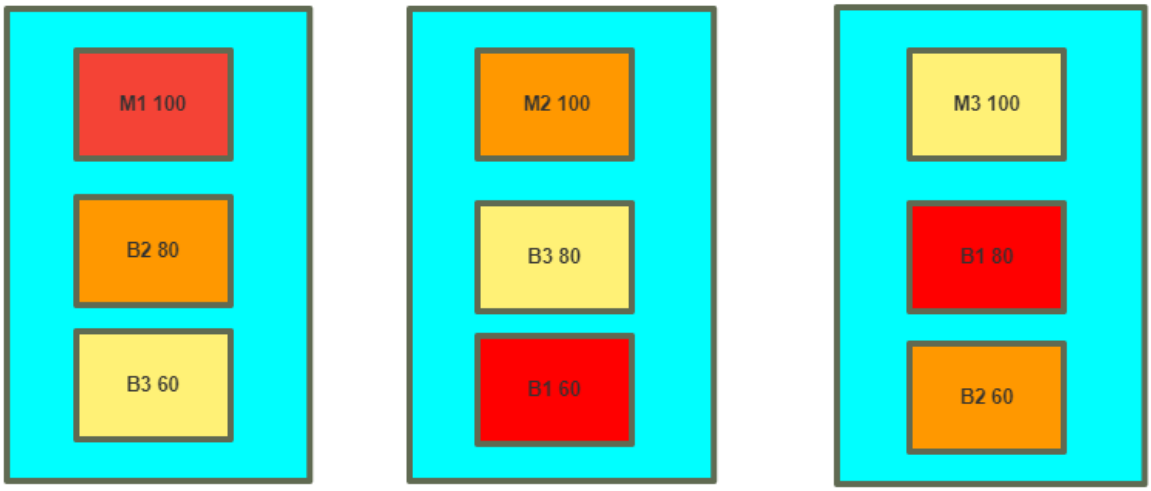
3.7 实现多主模架构

3.7.1 案例：三个节点的三主三从架构实现




```
#第一个节点ka1配置：
virtual_router_id 1 , Vrrp instance 1 , MASTER，优先级 100
virtual_router_id 3 , Vrrp instance 2 , BACKUP，优先级 80

#第二个节点ka2配置：
virtual_router_id 2 , Vrrp instance 1 , MASTER，优先级 100
virtual_router_id 1 , Vrrp instance 2 , BACKUP，优先级 80

#第三个节点ka3配置：
virtual_router_id 3 , Vrrp instance 1 , MASTER，优先级 100
virtual_router_id 2 , Vrrp instance 2 , BACKUP，优先级 80
```

3.7.2 案例：三个节点的三主六从架构实现




```
#第一个节点ka1配置：
virtual_router_id 1 , Vrrp instance 1 , MASTER，优先级100
virtual_router_id 2 , Vrrp instance 2 , BACKUP，优先级80
virtual_router_id 3 , Vrrp instance 3 , BACKUP，优先级60

#第二个节点ka2配置：
virtual_router_id 1 , Vrrp instance 1 , BACKUP，优先级60
virtual_router_id 2 , Vrrp instance 2 , MASTER，优先级100
virtual_router_id 3 , Vrrp instance 3 , BACKUP，优先级80

#第三个节点ka3配置：
virtual_router_id 1 , Vrrp instance 1 , BACKUP，优先级80
virtual_router_id 2 , Vrrp instance 2 , BACKUP，优先级60
virtual_router_id 3 , Vrrp instance 3 , MASTER，优先级100
```

3.8 同步组

LVS NAT 模型VIP和DIP需要同步，需要同步组

```
vrrp_sync_group VG_1 {
 group {
    VI_1  # name of vrrp_instance (below)
    VI_2  # One for each moveable IP
   }
  }
vrrp_instance VI_1 {
    eth0
    vip
  }
vrrp_instance VI_2 {
    eth1
    dip
  }
```

4 实现 IPVS 的高可用性

4.1 IPVS 相关配置

4.1.1 虚拟服务器配置结构

每一个虚拟服务器即一个IPVS集群

可以通过下面语法实现

```
 virtual_server IP port  {
    ...
    real_server {
        ...
    }
    real_server {
        ...
    }
  …
}
```

4.1.2 Virtual Server （虚拟服务器）的定义格式

```
virtual_server IP port       #定义虚拟主机IP地址及其端口
virtual_server fwmark int    #ipvs的防火墙打标，实现基于防火墙的负载均衡集群
virtual_server group string #使用虚拟服务器组
```

4.1.3 虚拟服务器组

将多个虚拟服务器定义成一个组，统一对外服务，如：http和https定义成一个虚拟服务器组

```
#参考文档：/usr/share/doc/keepalived/keepalived.conf.virtual_server_group
virtual_server_group <STRING> {
           # Virtual IP Address and Port
           <IPADDR> <PORT>
           <IPADDR> <PORT>
           ...
           # <IPADDR RANGE> has the form
           # XXX.YYY.ZZZ.WWW-VVV eg 192.168.200.1-10
           # range includes both .1 and .10 address
           <IPADDR RANGE> <PORT># VIP range VPORT
           <IPADDR RANGE> <PORT>
           ...
           # Firewall Mark (fwmark)
           fwmark <INTEGER>
           fwmark <INTEGER>
           ...
}
```

4.1.4 虚拟服务器配置

```
virtual_server IP port  {                    #VIP和PORT
    delay_loop <INT>                         #检查后端服务器的时间间隔
    lb_algo rr|wrr|lc|wlc|lblc|sh|dh         #定义调度方法
    lb_kind NAT|DR|TUN                       #集群的类型,注意要大写
    persistence_timeout <INT>                #持久连接时长
    protocol TCP|UDP|SCTP                    #指定服务协议,一般为TCP
    sorry_server <IPADDR> <PORT>             #所有RS故障时，备用服务器地址
```

```
    real_server <IPADDR> <PORT>  {          #RS的IP和PORT
        weight <INT>                        #RS权重
        notify_up <STRING>|<QUOTED-STRING>  #RS上线通知脚本
        notify_down <STRING>|<QUOTED-STRING> #RS下线通知脚本
        HTTP_GET|SSL_GET|TCP_CHECK|SMTP_CHECK|MISC_CHECK { ... } #定义当前主机健康状
态检测方法
    }
}

#注意:括号必须分行写,两个括号写在同一行,如: }} 会出错
```

4.1.5 应用层监测

应用层检测：HTTP_GET|SSL_GET

```
HTTP_GET|SSL_GET {
    url {
        path <URL_PATH>          #定义要监控的URL
        status_code <INT>        #判断上述检测机制为健康状态的响应码，一般为 200
    }
    connect_timeout <INTEGER>    #客户端请求的超时时长, 相当于haproxy的timeout server
    nb_get_retry <INT>           #重试次数
    delay_before_retry <INT>     #重试之前的延迟时长
    connect_ip <IP ADDRESS>      #向当前RS哪个IP地址发起健康状态检测请求
    connect_port <PORT>          #向当前RS的哪个PORT发起健康状态检测请求
    bindto <IP ADDRESS>          #向当前RS发出健康状态检测请求时使用的源地址
    bind_port <PORT>             #向当前RS发出健康状态检测请求时使用的源端口
}
```

范例：

```
virtual_server 10.0.0.10 80 {
        delay_loop 3
        lb_algo wrr
        lb_kind DR
        protocol TCP
        sorry_server 127.0.0.1 80
        real_server 10.0.0.7 80 {
            weight 1
            HTTP_GET {
                url {
                    path /monitor.html
                    status_code 200
                }
                connect_timeout 1
                nb_get_retry 3
                delay_before_retry 1
            }
        }
        real_server 10.0.0.17 80 {
            weight 1
            HTTP_GET {
                url {
                    path /
                    status_code 200
                }
```

```
                connect_timeout 1
                nb_get_retry 3
                delay_before_retry 1
            }
        }
}

#在后端服务器可以观察到健康检测日志
[root@web01 ~]#tail /var/log/nginx/access.log
10.0.0.201 - - [13/Jan/2023:11:27:01 +0800] "GET / HTTP/1.0" 200 24 "-"
"KeepAliveClient"
10.0.0.201 - - [13/Jan/2023:11:27:04 +0800] "GET / HTTP/1.0" 200 24 "-"
"KeepAliveClient"
10.0.0.201 - - [13/Jan/2023:11:27:07 +0800] "GET / HTTP/1.0" 200 24 "-"
"KeepAliveClient"
```

4.1.6 TCP监测

传输层检测：TCP_CHECK

```
TCP_CHECK {
     connect_ip <IP ADDRESS>     #向当前RS的哪个IP地址发起健康状态检测请求
     connect_port <PORT>         #向当前RS的哪个PORT发起健康状态检测请求
     bindto <IP ADDRESS>         #发出健康状态检测请求时使用的源地址
     bind_port <PORT>            #发出健康状态检测请求时使用的源端口
     connect_timeout <INTEGER>   #客户端请求的超时时长, 等于haproxy的timeout server
}
```

范例：

```
virtual_server 10.0.0.10 80 {
    delay_loop 6
    lb_algo wrr
    lb_kind DR
    #persistence_timeout 120    #会话保持时间
    protocol TCP
    sorry_server 127.0.0.1 80
    real_server 10.0.0.7 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 5
            nb_get_retry 3
            delay_before_retry 3
            connect_port 80
        }
    }
    real_server 10.0.0.17 80 {
        weight 1
        TCP_CHECK {
            connect_timeout 5
            nb_get_retry 3
            delay_before_retry 3
            connect_port 80
        }
    }
}
```

4.2 实战案例

4.2.1 实战案例：实现单主的 LVS-DR 模式

准备web服务器并使用脚本绑定VIP至web服务器lo网卡

```
#准备两台后端RS主机
[root@rs1 ~]#cat lvs_dr_rs.sh
#!/bin/bash
#Author:wangxiaochun
#Date:2017-08-13
vip=10.0.0.10
mask='255.255.255.255'
dev=lo:1
rpm -q httpd &> /dev/null || yum -y install httpd &>/dev/null
service httpd start &> /dev/null && echo "The httpd Server is Ready!"
echo "<h1>`hostname`</h1>" > /var/www/html/index.html

case $1 in
start)
    echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
    echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
    echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
    echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
    ifconfig $dev $vip netmask $mask #broadcast $vip up
    #route add -host $vip dev $dev
    echo "The RS Server is Ready!"
    ;;
stop)
    ifconfig $dev down
    echo 0 > /proc/sys/net/ipv4/conf/all/arp_ignore
    echo 0 > /proc/sys/net/ipv4/conf/lo/arp_ignore
    echo 0 > /proc/sys/net/ipv4/conf/all/arp_announce
    echo 0 > /proc/sys/net/ipv4/conf/lo/arp_announce
    echo "The RS Server is Canceled!"
    ;;
*)
    echo "Usage: $(basename $0) start|stop"
    exit 1
    ;;
esac

[root@rs1 ~]#bash lvs_dr_rs.sh start
The httpd Server is Ready!
The RS Server is Ready!
[root@rs1 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.10/32 scope global lo:1
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
```

```
    link/ether 00:0c:29:32:80:38 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.7/24 brd 10.0.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe32:8038/64 scope link
       valid_lft forever preferred_lft forever


[root@rs2 ~]#bash lvs_dr_rs.sh start
The httpd Server is Ready!
The RS Server is Ready!

[root@rs2 ~]#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.10/32 scope global lo:1
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:33:b4:1a brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.17/24 brd 10.0.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe33:b41a/64 scope link
       valid_lft forever preferred_lft forever

#测试直接访问两台RS
[root@centos6 ~]#curl 10.0.0.7
<h1>rs1.wang.org</h1>
[root@centos6 ~]#curl 10.0.0.17
<h1>rs2.wang.org</h1>
```

配置keepalived

```
#ka1节点的配置
[root@ka1 ~]#cat  /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
            root@localhost
        }
        notification_email_from keepalived@localhost
        smtp_server 127.0.0.1
        smtp_connect_timeout 30
        router_id ka1.wang.org
        vrrp_mcast_group4 224.0.100.10
    }
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 66
    priority 100
    advert_int 1
    authentication {
```

```
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    }
virtual_server 10.0.0.10 80 {
        delay_loop 3
        lb_algo wrr
        lb_kind DR
        protocol TCP
        sorry_server 127.0.0.1 80
        real_server 10.0.0.7 80 {
            weight 1
            HTTP_GET {                #应用层检测
                url {
                    path /
                    status_code 200
                }
                connect_timeout 1
                nb_get_retry 3
                delay_before_retry 1
            }
        }
        real_server 10.0.0.17 80 {
            weight 2
            TCP_CHECK {               #另一台主机使用TCP检测
                connect_timeout 5
                nb_get_retry 3
                delay_before_retry 3
                connect_port 80
            }
        }
}

#ka2节点的配置，配置和ka1基本相同，只需修改三行
[root@ka2 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
            root@localhost
        }
        notification_email_from keepalived@localhost
        smtp_server 127.0.0.1
        smtp_connect_timeout 30
        router_id ka1.wang.org             #修改此行
        vrrp_mcast_group4 224.0.100.10
    }
vrrp_instance VI_1 {
    state BACKUP                            #修改此行
    interface eth0
    virtual_router_id 66
    priority 80                             #修改此行
    advert_int 1
```

```
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    }
virtual_server 10.0.0.10 80 {
        delay_loop 3
        lb_algo rr
        lb_kind DR
        protocol TCP
        sorry_server 127.0.0.1 80
        real_server 10.0.0.7 80 {
            weight 1
            HTTP_GET {
                url {
                    path /
                    status_code 200
                }
                connect_timeout 1
                nb_get_retry 3
                delay_before_retry 1
            }
        }
        real_server 10.0.0.17 80 {
            weight 1
            TCP_CHECK {
                connect_timeout 5
                nb_get_retry 3
                delay_before_retry 3
                connect_port 80
            }
        }
}
```

访问测试结果

```
[root@centos6 ~]#curl 10.0.0.10
<h1>rs1.wang.org</h1>
[root@centos6 ~]#curl 10.0.0.10
<h1>rs2.wang.org</h1>

[root@ka1 ~]#dnf -y install ipvsadm
[root@ka1 ~]#ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.0.10:80 rr
  -> 10.0.0.7:80                  Route   1      0          0
  -> 10.0.0.17:80                 Route   1      0          0
```

```
#后端服务器可以看到七层监测的访问日志生成
[root@web01 ~]#tail -f  /var/log/httpd/access_log
10.0.0.202 - - [22/Nov/2022:18:23:48 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:23:50 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:23:52 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:23:54 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:23:56 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:23:58 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:24:00 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:24:02 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:24:04 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
10.0.0.202 - - [22/Nov/2022:18:24:06 +0800] "GET /index.html HTTP/1.0" 200 26 "-
" "KeepAliveClient"
```

模拟故障

```
#第一台RS1故障，自动切换至RS2
[root@rs1 ~]#chmod 0 /var/www/html/index.html

[root@centos6 ~]#curl 10.0.0.10
<h1>rs2.wang.org</h1>
[root@centos6 ~]#curl 10.0.0.10
<h1>rs2.wang.org</h1>

[root@ka1 ~]#dnf -y install ipvsadm
[root@ka1 ~]#ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.0.10:80 rr
  -> 10.0.0.17:80                 Route   1      0          3

#后端RS服务器都故障，启动Sorry Server
[root@rs2 ~]#systemctl stop httpd
[root@centos6 ~]#curl 10.0.0.10
Sorry Server on ka1
[root@ka1 ~]#ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.0.10:80 rr
  -> 127.0.0.1:80                 Route   1      0          0

#ka1故障，自动切换至ka2
[root@ka1 ~]#killall keepalived
[root@centos6 ~]#curl 10.0.0.10
Sorry Server on ka2
```

```
#恢复都有后端 RS
[root@rs1 ~]#chmod 644 /var/www/html/index.html
[root@rs2 ~]#systemctl start httpd
[root@centos6 ~]#curl 10.0.0.10
<h1>rs1.wang.org</h1>
[root@centos6 ~]#curl 10.0.0.10
<h1>rs2.wang.org</h1>
[root@ka1 ~]#hostname -I
10.0.0.8
[root@ka2 ~]#hostname -I
10.0.0.18 10.0.0.10

#恢复ka1服务器，又抢占回原来的VIP
[root@ka1 ~]#systemctl start keepalived.service
[root@ka1 ~]#hostname -I
10.0.0.8 10.0.0.10
[root@ka2 ~]#hostname -I
10.0.0.18
[root@centos6 ~]#curl 10.0.0.10
<h1>rs1.wang.org</h1>
[root@centos6 ~]#curl 10.0.0.10
<h1>rs2.wang.org</h1>
```

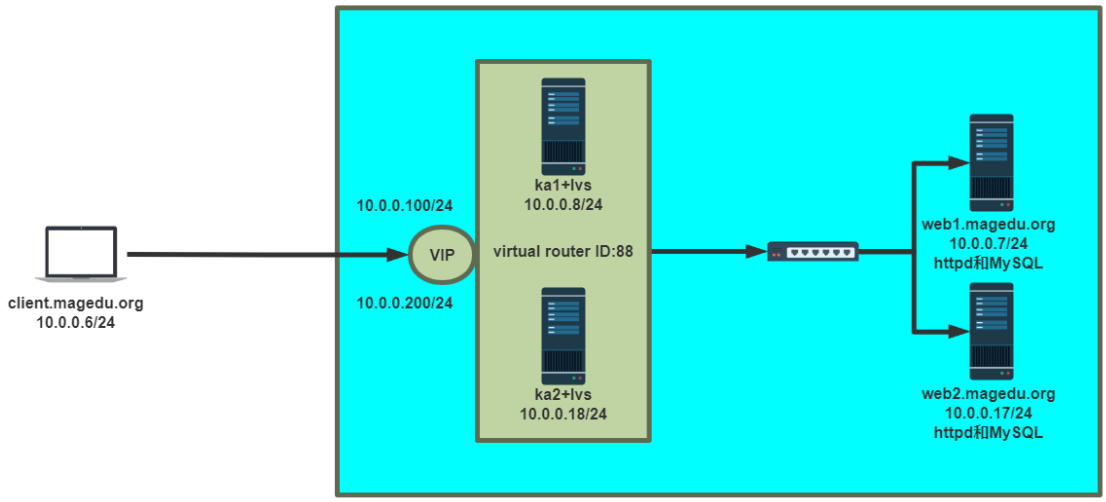
4.2.2 实战案例：实现双主的 LVS-DR 模式

```
[root@ka1 ~]#vim /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
            root@localhost
        }
        notification_email_from keepalived@localhost
        smtp_server 127.0.0.1
        smtp_connect_timeout 30
        router_id ka1.wang.org              #另一个节点为ka2.wang.org
        vrrp_mcast_group4 224.0.100.10
    }

vrrp_instance VI_1 {
    state MASTER                            #在另一个结点上为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                            #在另一个结点上为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1      #指定VIP
    }
}

vrrp_instance VI_2 {
    state BACKUP                            #在另一个结点上为MASTER
    interface eth0
    virtual_router_id  88
```

```
    priority 80                                     #在另一个结点上为100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:2      #指定VIP2
    }
}
virtual_server 10.0.0.10 80 {
    delay_loop 6
    lb_algo rr
    lb_kind DR
    protocol TCP
    sorry_server 127.0.0.1 80
    real_server 10.0.0.7 80 {  #指定RS1地址
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }

    }
    real_server 10.0.0.17 80 {                  #指定RS2地址
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }

    }

}

virtual_server 10.0.0.20 80 {                       #指定VIP2
    delay_loop 6
    lb_algo rr
    lb_kind DR
    protocol TCP
    sorry_server 127.0.0.1 80
    real_server 10.0.0.27 80 {                      #指定RS3地址
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
```

```
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }

    }
    real_server 10.0.0.37 80 {                          #指定RS4地址
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}
```

范例: 双主分别实现httpd和mysql服务的调度




```
[root@ka1 conf.d]#cat web1.conf
vrrp_instance web1 {
    state MASTER
    interface eth0
    virtual_router_id 66
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.100/24 dev eth0 label eth0:100
    }
    notify_master "/etc/keepalived/notify.sh master"

    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}
```

```
[root@ka1 conf.d]#cat lvs_.conf
lvs_mysql.conf  lvs_web1.conf
[root@ka1 conf.d]#cat lvs_web1.conf
virtual_server 10.0.0.100 80 {
        delay_loop 3
        lb_algo rr
        lb_kind DR
        protocol TCP
        sorry_server 127.0.0.1 80
        real_server 10.0.0.7 80 {
            weight 1
            HTTP_GET {
                url {
                    path /monitor.html
                    status_code 200
                }
                connect_timeout 1
                nb_get_retry 3
                delay_before_retry 1
            }
        }
        real_server 10.0.0.17 80 {
            weight 1
            TCP_CHECK {
                connect_timeout 5
                nb_get_retry 3
                delay_before_retry 3
                connect_port 80
            }
        }
}
[root@ka1 conf.d]#cat mysql_vip.conf
vrrp_instance mysql{
    state BACKUP
    interface eth0
    virtual_router_id 88
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.200/24 dev eth0 label eth0:200
    }
}

[root@ka1 conf.d]#cat lvs_mysql.conf
virtual_server 10.0.0.200 3306 {
        delay_loop 3
        lb_algo rr
        lb_kind DR
        protocol TCP
        real_server 10.0.0.7 3306 {
            weight 1
            TCP_CHECK {
                connect_timeout 5
                nb_get_retry 3
```

```
                delay_before_retry 3
                connect_port 3306
            }
        }
        real_server 10.0.0.17 3306 {
            weight 1
            TCP_CHECK {
                connect_timeout 5
                nb_get_retry 3
                delay_before_retry 3
                connect_port 3306
            }
        }
}


#注意:在后端服务器要实现两个VIP的配置
[root@web1 ~]#ip a show lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.100/32 scope global lo:1
       valid_lft forever preferred_lft forever
    inet 10.0.0.200/32 scope global lo:2
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever


[root@ka1 ~]#ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.0.0.100:80 rr
  -> 10.0.0.7:80                  Route   1      0          0
  -> 10.0.0.17:80                 Route   1      0          0
TCP  10.0.0.200:3306 rr
  -> 10.0.0.7:3306                Route   1      0          0
  -> 10.0.0.17:3306               Route   1      0          0


#测试
[root@client ~]#while true;do mysql -utest -p123456 -h10.0.0.200 -e 'show
variables like "%hostname%"';curl 10.0.0.100;sleep 0.5;done
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| hostname      | web1.wang.org |
+---------------+-----------------+
10.0.0.17
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| hostname      | web2.wang.org |
+---------------+-----------------+
10.0.0.7
```

4.2.3 实战案例：实现单主的 LVS-DR 模式，利用FWM绑定成多个服务为一个集群服务

参考文档： 注意有bug

```
#源码
/usr/local/src/keepalived-2.2.8/doc/samples/keepalived.conf.fwmark

#包安装
/usr/share/doc/keepalived/keepalived.conf.vrrp.localcheck
```

范例：

```
#准备后端https服务
[root@web01 ~]#apt update && apt -y install nginx

#生成证书文件
[root@web01 conf.d]#cat certificate.sh
#!/bin/bash
#
#**********************************************************************
#Author:        wangxiaochun
#QQ:            29308620
#Date:          2020-02-07
#FileName:       certificate.sh
#URL:           http://www.liwenliang.org
#Description:         本脚本纪念武汉疫情鸣笛人李文亮医生
#Copyright (C):     2020 All rights reserved
#**********************************************************************

SITE_NAME=www.wang.org
CA_SUBJECT="/O=wang/CN=ca.wang.org"
SUBJECT="/C=CN/ST=henan/L=zhengzhou/O=wang/CN=$SITE_NAME"
SERIAL=34
EXPIRE=202002
FILE=$SITE_NAME

openssl req  -x509 -newkey rsa:2048 -subj $CA_SUBJECT -keyout ca.key -nodes -
days 202002 -out ca.crt

openssl req -newkey rsa:2048 -nodes -keyout ${FILE}.key  -subj $SUBJECT -out
${FILE}.csr

openssl x509 -req -in ${FILE}.csr  -CA ca.crt -CAkey ca.key -set_serial $SERIAL
 -days $EXPIRE -out ${FILE}.crt

chmod 600 ${FILE}.key ca.key

[root@web01 conf.d]#bash certificate.sh
[root@web01 conf.d]#cat ca.crt www.wang.org.crt >> www.wang.org.pem
[root@web01 ~]#ls /etc/nginx/conf.d/
ca.crt  ca.key  certificate.sh  www.wang.org.crt  www.wang.org.csr
www.wang.org.key  www.wang.org.pem

[root@web01 ~]#vim /etc/nginx/sites-enabled/default
```

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    listen 443 ssl default_server;
    ssl_certificate  /etc/nginx/conf.d/www.wang.org.pem;
    ssl_certificate_key /etc/nginx/conf.d/www.wang.org.key;
    ......

[root@web01 ~]#systemctl restart nginx

#配置第二台后端web服务器
[root@web02 ~]#apt update && apt -y install nginx
[root@web01 conf.d]#scp www.wang.org.key www.wang.org.pem
10.0.0.102:/etc/nginx/conf.d/
[root@web01 conf.d]#scp /etc/nginx/sites-available/default
10.0.0.102:/etc/nginx/sites-available/default
[root@web02 ~]#systemctl restart nginx

#两个节点都执行以下操作
[root@ka1 ~]#iptables -t mangle -A PREROUTING -d 10.0.0.10   -p tcp  -m
multiport --dports 80,443 -j MARK --set-mark 6

[root@ka1 ~]#vim  /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org          #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.100.100.100
    }

vrrp_instance VI_1 {
    state MASTER                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                #在另一个节点为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
}

virtual_server fwmark 6 {   #指定FWM为6
    delay_loop 2
```

```
    lb_algo rr
    lb_kind DR
    sorry_server 127.0.0.1 80  #注意端口必须指定
    real_server 10.0.0.7 80 {  #注意端口必须指定
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
            connect_timeout 2
            nb_get_retry 3
            delay_before_retry 3
            connect_port    80  #如果此行不写，real_server 10.0.0.7 80 就必须指定端口
        }
    }
    real_server 10.0.0.17 80 { #注意端口必须指定,只用于健康性检查,而非通信
        weight 1
        HTTP_GET {
            url {
                path /
                status_code 200
            }
            connect_timeout 2
            nb_get_retry 3
            delay_before_retry 3
            connect_port    80  #如果此行不写，real_server 10.0.0.17 80 就必须指定端
口
        }
    }
}


#在RS1和RS2运行下面脚本
[root@rs1 ~]#cat lvs_dr_rs.sh
#!/bin/bash
#Author:wangxiaochun
#Date:2017-08-13
vip=10.0.0.10
vip2=10.0.0.20
mask='255.255.255.255'
dev=lo:1
dev2=lo:2
rpm -q httpd &> /dev/null || yum -y install httpd &>/dev/null
service httpd start &> /dev/null && echo "The httpd Server is Ready!"
echo "<h1>`hostname`</h1>" > /var/www/html/index.html

case $1 in
start)
    echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
    echo 1 > /proc/sys/net/ipv4/conf/lo/arp_ignore
    echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
    echo 2 > /proc/sys/net/ipv4/conf/lo/arp_announce
    ifconfig $dev $vip netmask $mask #broadcast $vip up
    ifconfig $dev2 $vip2 netmask $mask #broadcast $vip up
    #route add -host $vip dev $dev
    echo "The RS Server is Ready!"
    ;;
```

```
stop)
    ifconfig $dev down
    ifconfig $dev2 down
    echo 0 > /proc/sys/net/ipv4/conf/all/arp_ignore
    echo 0 > /proc/sys/net/ipv4/conf/lo/arp_ignore
    echo 0 > /proc/sys/net/ipv4/conf/all/arp_announce
    echo 0 > /proc/sys/net/ipv4/conf/lo/arp_announce
    echo "The RS Server is Canceled!"
    ;;
*)
    echo "Usage: $(basename $0) start|stop"
    exit 1
    ;;
esac
[root@rs1 ~]#bash lvs_dr_rs.sh  start
[root@rs2 ~]#bash lvs_dr_rs.sh  start

#访问测试
[root@centos6 ~]#curl 10.0.0.10;curl  -k https://10.0.0.20
<h1>rs1.wang.org</h1>
<h1>rs2.wang.org</h1>
```

5 基于 VRRP Script 实现其它应用的高可用性

keepalived利用 VRRP Script 技术，可以调用外部的辅助脚本进行资源监控，并根据监控的结果实现优先动态调整，从而实现其它应用的高可用性功能

参考配置文件：

```
#源码
/usr/local/src/keepalived-2.2.8/doc/samples/keepalived.conf.vrrp.localcheck
[root@ubuntu2204 ~]#cat /usr/local/src/keepalived-
2.2.8/doc/samples/keepalived.conf.vrrp.localcheck
! Configuration File for keepalived

# Note: The "</dev/tcp/127.0.0.1/80" syntax for a script
# requires that /bin/sh is Bash.
# On systems where /bin/sh is not Bash (e.g. Debian, Ubuntu),
# then the following alternatives should be used:
#   script "bash -c '</dev/tcp/127.0.0.1/80'"
# or
#   script "nc -z 127.0.0.1 80"

vrrp_script chk_sshd {
       script "killall -0 sshd"         # cheaper than pidof
       interval 2                       # check every 2 seconds
       weight -4                        # default prio: -4 if KO
       fall 2                           # require 2 failures for KO
       rise 2                           # require 2 successes for OK
}

vrrp_script chk_haproxy {
       script "killall -0 haproxy"      # cheaper than pidof
       interval 2                       # check every 2 seconds
}
```

```
vrrp_script chk_http_port {
       script "</dev/tcp/127.0.0.1/80" # connects and exits
       interval 1                      # check every second
       weight -2                       # default prio: -2 if connect fails
}

vrrp_script chk_https_port {
       script "</dev/tcp/127.0.0.1/443"
       interval 1
       weight -2
}

vrrp_script chk_smtp_port {
       script "</dev/tcp/127.0.0.1/25"
       interval 1
       weight -2
}

vrrp_instance VI_1 {
    interface eth0
    state MASTER
    virtual_router_id 51
    priority 100
    virtual_ipaddress {
        192.168.200.18/25
    }
    track_interface {
       eth1 weight 2   # prio = +2 if UP
       eth2 weight -2  # prio = -2 if DOWN
       eth3            # no weight, fault if down
    }
    track_script {
       chk_sshd                # use default weight from the script
       chk_haproxy weight 2    # +2 if process is present
       chk_http_port
       chk_https_port
       chk_smtp_port
    }
}

vrrp_instance VI_2 {
    interface eth1
    state MASTER
    virtual_router_id 52
    priority 100
    virtual_ipaddress {
        192.168.201.18/26
    }
    track_interface {
       eth0 weight 2   # prio = +2 if UP
       eth2 weight -2  # prio = -2 if DOWN
       eth3            # no weight, fault if down
    }
    track_script {
       chk_haproxy weight 2
       chk_http_port
       chk_https_port
       chk_smtp_port
```

```
    }
}

vrrp_instance VI_3 {
    interface eth0
    virtual_router_id 53
    priority 100
    virtual_ipaddress {
        192.168.200.19/27
    }
}

vrrp_instance VI_4 {
    interface eth1
    virtual_router_id 54
    priority 100
    virtual_ipaddress {
        192.168.201.19/28
    }
}

vrrp_instance VI_5 {
    state MASTER
    interface eth0
    virtual_router_id 55
    priority 100
    virtual_ipaddress {
        192.168.200.20/27
    }
}


vrrp_instance VI_6 {
    state MASTER
    interface eth0
    virtual_router_id 56
    priority 100
    virtual_ipaddress {
        192.168.200.21/27
    }
}

vrrp_instance VI_7 {
    state MASTER
    interface eth0
    virtual_router_id 57
    priority 100
    virtual_ipaddress {
        192.168.200.22/27
    }
}

vrrp_instance VI_8 {
    state MASTER
    interface eth0
    virtual_router_id 58
    priority 100
    virtual_ipaddress {
```

```
        192.168.200.23/27
    }
}

vrrp_instance VI_9 {
    state MASTER
    interface eth0
    virtual_router_id 59
    priority 100
    virtual_ipaddress {
        192.168.200.24/27
    }
}


#包安装
/usr/share/doc/keepalived/keepalived.conf.vrrp.localcheck
```

5.1 VRRP Script 配置

分两步实现：

- 定义脚本

  vrrp_script：自定义资源监控脚本，vrrp实例根据脚本返回值，公共定义，可被多个实例调用，定义在vrrp实例之外的独立配置块，一般放在global_defs设置块之后,是和global_defs平级的语句块

  通常此脚本用于监控指定应用的状态。一旦发现应用的状态异常，则触发对MASTER节点的权重减至低于SLAVE节点，从而实现 VIP 切换到 SLAVE 节点

  当`keepalived_script`用户存在时,会以此用户身份运行脚本,否则默认以root运行脚本

  注意: 此定义脚本的语句块一定要放在下面调用此语句vrrp_instance语句块的前面

  ```
  vrrp_script <SCRIPT_NAME> {
      script <STRING>|<QUOTED-STRING>    #此脚本返回值为非0时，会触发下面OPTIONS执行
      OPTIONS
  }
  ```

- 调用脚本

  track_script：调用vrrp_script定义的脚本去监控资源，定义在VRRP实例之内，调用事先定义的vrrp_script

  ```
  track_script {
      SCRIPT_NAME_1
      SCRIPT_NAME_2
  }
  ```

5.1.1 定义 VRRP script

```
vrrp_script <SCRIPT_NAME> {                    #定义一个检测脚本，在global_defs 之外配置
      script <STRING>|<QUOTED-STRING>          #shell命令或脚本路径
      interval <INTEGER>                       #间隔时间，单位为秒，默认1秒
      timeout <INTEGER>                        #超时时间
      weight <INTEGER:-254..254>               #默认为0,如果设置此值为负数，当上面脚本返回值为非0时，会将此值与本节点权重相加可以降低本节点权重，即表示fall. 如果是正数，当脚本返回值为0，会将此值与本节点权重相加可以提高本节点权重，即表示 rise.通常使用负值
      fall <INTEGER>                           #执行脚本连续几次都失败,则转换为失败，建议设为2以上
      rise <INTEGER>                           #执行脚本连续几次都成功，把服务器从失败标记为成功
      user USERNAME [GROUPNAME]                #执行监测脚本的用户或组
      init_fail                                #设置默认标记为失败状态，监测成功之后再转换为成功状态
}
```

5.1.2 调用 VRRP script

```
vrrp_instance VI_1 {
    …
    track_script {
        <SCRIPT_NAME>
  }
}
```

5.2 实战案例：利用脚本实现主从角色切换

```
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org                          #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.0.100.100
}

vrrp_script check_down {
    script "[ ! -f /etc/keepalived/down ]"          #/etc/keepalived/down存在时返回非0，触发权重-30
    interval 1
    weight -30
    fall 3
    rise 2
    timeout 2
}

vrrp_instance VI_1 {
    state MASTER                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                #在另一个节点为80
```

```
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
       check_down           #调用前面定义的脚本
   }
}

[root@ka1 ~]#touch /etc/keepalived/down

[root@ka1 ~]#tail -f /var/log/messages
Mar 28 19:47:03 ka1-centos8 Keepalived_vrrp[7200]: Script `check_down` now
returning 1
Mar 28 19:47:05 ka1-centos8 Keepalived_vrrp[7200]: VRRP_Script(chk_down) failed
(exited with status 1)
Mar 28 19:47:05 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Changing effective
priority from 100 to 70
Mar 28 19:47:07 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Master received advert
from 10.0.0.18 with higher priority 80, ours 70
Mar 28 19:47:07 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Entering BACKUP STATE
Mar 28 19:47:07 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) removing VIPs.

[root@rs1 ~]#tcpdump -i eth0 -nn 224.0.100.100
19:42:09.578203 IP 10.0.0.8 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 100, authtype simple, intvl 1s, length 20
19:42:10.579304 IP 10.0.0.8 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 70, authtype simple, intvl 1s, length 20


[root@ka1 ~]#rm -f  /etc/keepalived/down
[root@ka1 ~]#tail -f /var/log/messages
Mar 28 19:47:45 ka1-centos8 Keepalived_vrrp[7200]: Script `check_down` now
returning 0
Mar 28 19:47:46 ka1-centos8 Keepalived_vrrp[7200]: VRRP_Script(check_down)
succeeded
Mar 28 19:47:46 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Changing effective
priority from 70 to 100
Mar 28 19:47:46 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) received lower
priority (80) advert from 10.0.0.18 - discarding
Mar 28 19:47:47 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) received lower
priority (80) advert from 10.0.0.18 - discarding
Mar 28 19:47:48 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) received lower
priority (80) advert from 10.0.0.18 - discarding
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Receive advertisement
timeout
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Entering MASTER STATE
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) setting VIPs.
```

```
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: Sending gratuitous ARP on
eth0 for 10.0.0.10
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: (VI_1) Sending/queueing
gratuitous ARPs on eth0 for 10.0.0.10
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: Sending gratuitous ARP on
eth0 for 10.0.0.10
Mar 28 19:47:49 ka1-centos8 Keepalived_vrrp[7200]: Sending gratuitous ARP on
eth0 for 10.0.0.10

[root@rs1 ~]#tcpdump -i eth0 -nn 224.0.100.100
19:49:16.199462 IP 10.0.0.18 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 80, authtype simple, intvl 1s, length 20
19:49:17.199897 IP 10.0.0.18 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 80, authtype simple, intvl 1s, length 20
19:49:17.810376 IP 10.0.0.8 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 100, authtype simple, intvl 1s, length 20
19:49:18.811048 IP 10.0.0.8 > 224.0.100.100: VRRPv2, Advertisement, vrid 66,
prio 100, authtype simple, intvl 1s, length 20
```

5.3 实战案例：实现单主模式的 Nginx 反向代理的高可用

```
#在两个节点都配置nginx反向代理
[root@ka1 ~]#vim /etc/nginx/nginx.conf
http {
    upstream websrvs {
        server 10.0.0.7:80 weight=1;
        server 10.0.0.17:80 weight=1;
    }
    server {
        listen 80;
        location /{
            proxy_pass http://websrvs/;
        }
    }
}

#在两个节点都配置实现nginx反向代理高可用
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org          #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.0.100.100
}
vrrp_script check_nginx {
    script "/etc/keepalived/check_nginx.sh"
    #script "/usr/bin/killall -0 nginx"   此写法支持
    #script "/usr/bin/killall -0 nginx &>/dev/null" 不支持&>此写法
    interval 1
    weight -30
    fall 3
    rise 5
```

```
    timeout 2
}

vrrp_instance VI_1 {
    state MASTER                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                #在另一个节点为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
        check_nginx
    }
}

[root@ka1 ~]# yum install psmisc -y
[root@ka1 ~]# cat /etc/keepalived/check_nginx.sh
#!/bin/bash
/usr/bin/killall -0 nginx || systemctl restart nginx
[root@ka1 ~]# chmod a+x /etc/keepalived/check_nginx.sh
```

范例: 利用通知脚本,实现切换时，自动重启服务

```
vim /etc/keepalived/notify.sh
#!/bin/bash
contact='root@localhost'
notify() {
    mailsubject="$(hostname) to be  $1:vip floating"
    mailbody="$(date +'%F %T'):vrrp transition,$(hostname) change to be $1"
    echo $mailbody | mail -s "$mailsubject" $contract
}
case $1 in
master)
    notify master
    systemctl start nginx
    ;;
backup)
    notify backup
    systemctl restart nginx
    ;;
fault)
    notify fault
    ;;
*)
    echo "Usage: $(basename $0) {master|backup|fault}"
```

```
esac
```

5.4 实战案例：实现双主模式 Nginx 反向代理的高可用

```
#在两个节点都配置nginx反向代理
[root@ka1 ~]vim /etc/nginx/nginx.conf
http {
    upstream websrvs {
        server 10.0.0.7:80 weight=1;
        server 10.0.0.17:80 weight-1;
    }
    upstream websrvs2 {
        server 10.0.0.27:80 weight=1;
        server 10.0.0.37:80 weight-1;
    }

    server {
        listen 80;
        server_name www.a.com;
        location /{
            proxy_pass http://webservs/;
        }
    }
    server {
        listen 80;
        server_name www.b.com;
        location /{
            proxy_pass http://webservs2/;
        }
    }

}


#在两个节点都配置实现双主模式的nginx反向代理高可用
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org          #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.100.100.100
}
vrrp_script check_nginx {
    script "/etc/keepalived/check_nginx.sh"
    #script "/usr/bin/killall -0 nginx"
    interval 1
    weight -30
    fall 3
    rise 5
    timeout 2
```

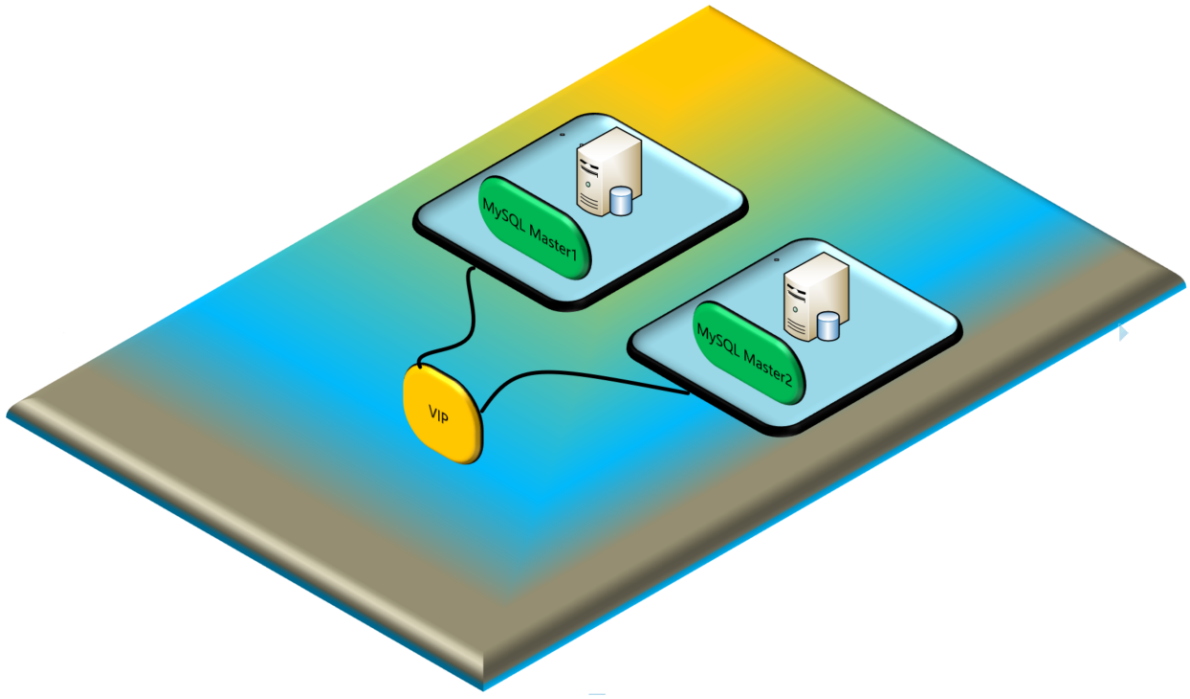
```
}

vrrp_instance VI_1 {
    state MASTER                                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                                #在另一个节点为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
        check_nginx
    }
}

vrrp_instance VI_2 {
    state BACKUP                            #在另一个节点为MASTER
    interface eth0
    virtual_router_id 88
    priority 80                             #在另一个节点为100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.20/24 dev eth0 label eth0:2
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
        check_nginx
    }
}
[root@ka1 ~]# yum install psmisc -y
[root@ka1 ~]# cat /etc/keepalived/check_nginx.sh
#!/bin/bash
/usr/bin/killall -0 nginx
[root@ka1 ~]# chmod a+x /etc/keepalived/check_nginx.sh
```

5.5 实战案例：实现 HAProxy 高可用

```
#在两个ka1和ka2先实现haproxy的配置
[root@ka1 ~]#apt update && apt -y install haproxy
[root@ka1 ~]#cat /etc/haproxy/haproxy.cfg
listen  web_http
    bind 10.0.0.10:80
    server web1 10.0.0.7:80 check
    server web2 10.0.0.17:80 check

listen stats
    mode http
    bind 10.0.0.8:9999
    stats enable
    log global
    stats uri     /haproxy-status
    stats auth    haadmin:123456

#在两个ka1和ka2两个节点启用内核参数
[root@ka1，2 ~]#vim /etc/sysctl.conf
net.ipv4.ip_nonlocal_bind = 1
[root@ka1，2 ~]#sysctl -p

#创建脚本
[root@ka1 ~]# yum install psmisc -y
[root@ka1 ~]# cat /etc/keepalived/check_haproxy.sh
#!/bin/bash
/usr/bin/killall -0 haproxy  || systemctl restart haproxy

#Ubuntu22.04包安装HAProxy-v2.4.22上面写法不支持，换成下面写法
[root@ka1 ~]#cat /etc/keepalived/check_haproxy.sh
if /usr/bin/killall -0 haproxy  ;then
    exit 0
else
    systemctl restart haproxy
fi

[root@ka1 ~]# chmod a+x /etc/keepalived/check_haproxy.sh

#配置keepalived调用脚本
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org              #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.0.100.100
}
vrrp_script check_haproxy {             #定义脚本
    script "/etc/keepalived/check_haproxy.sh"
    interval 1
    weight -30
    fall 3
    rise 2
    timeout 2
}
```

```
vrrp_instance VI_1 {
    state MASTER                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                #在另一个节点为80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
        check_haproxy                       #调用上面定义的脚本
    }
}
```

5.6 实战案例：实现 MySQL 双主模式的高可用

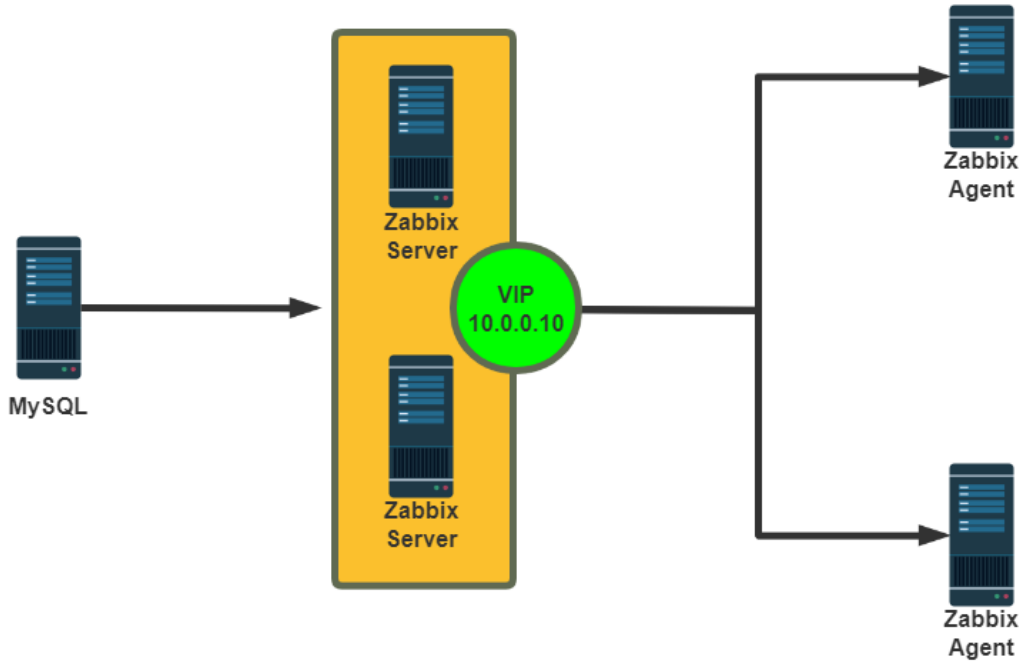



```
#先实现MySQL的双主架构
[root@ka1 ~]#vim /etc/my.cnf.d/mariadb-server.cnf
[mysqld]
server-id=8
log-bin
auto_increment_offset=1          #开始点
auto_increment_increment=2       #增长幅度
```

```
#在ka2第二个节点创建连接MySQL查看同步状态的授权用户
[root@ka2 ~]#mysql -uroot -p123456
MariaDB [(none)]> grant replication slave on *.* to repluser@'10.0.0.%'
identified by '123456';


#实现MySQL的健康性检测脚本1
[root@ka1 ~]#vi /etc/keepalived/check_mysql.sh
#!/bin/bash
slave_is=( $(mysql -uroot -p123456 -h10.0.0.18 -e "show slave status\G" | grep
"Slave_.*_Running:" | awk '{print $2}') )
if [ "${slave_is[0]}" = "Yes" -a "${slave_is[1]}" = "Yes" ];then
    exit 0
else
    exit 1
fi

#实现MySQL的健康性检测脚本2
[root@ka1 ~]#vi /etc/keepalived/check_mysql.sh
mysqladmin -uroot -p123456  ping &> /dev/null

#实现MySQL的健康性检测脚本3
[root@ka1 ~]#vi /etc/keepalived/check_mysql.sh
mysql  -uroot -p123456 -e 'status' &> /dev/null

#实现MySQL的健康性检测脚本4
[root@ka1 ~]#vi /etc/keepalived/check_mysql.sh
systemctl is-active mariadb &> /dev/null

#配置keepalived调用上面脚本
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived
    global_defs {
        notification_email {
        root@localhost
        }
    notification_email_from kaadmin@localhost
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id ka1.wang.org          #在另一个节点为ka2.wang.org
    vrrp_mcast_group4 224.0.100.100
}
vrrp_script check_mysql {                   #只需在第一个节点上实现脚本
    script "/etc/keepalived/check_mysql.sh"
    interval 1
    weight -30
    fall 3
    rise 2
    timeout 2
}

vrrp_instance VI_1 {
    state MASTER                #在另一个节点为BACKUP
    interface eth0
    virtual_router_id 66
    priority 100                #在另一个节点为80
    advert_int 1
    authentication {
```

```
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    notify_master "/etc/keepalived/notify.sh master"
    notify_backup "/etc/keepalived/notify.sh backup"
    notify_fault "/etc/keepalived/notify.sh fault"
    track_script {
       check_mysql                          #只需在第一个节点上实现脚本
   }
}
```

5.7 实战案例：实现 Zabbix Server 的高可用

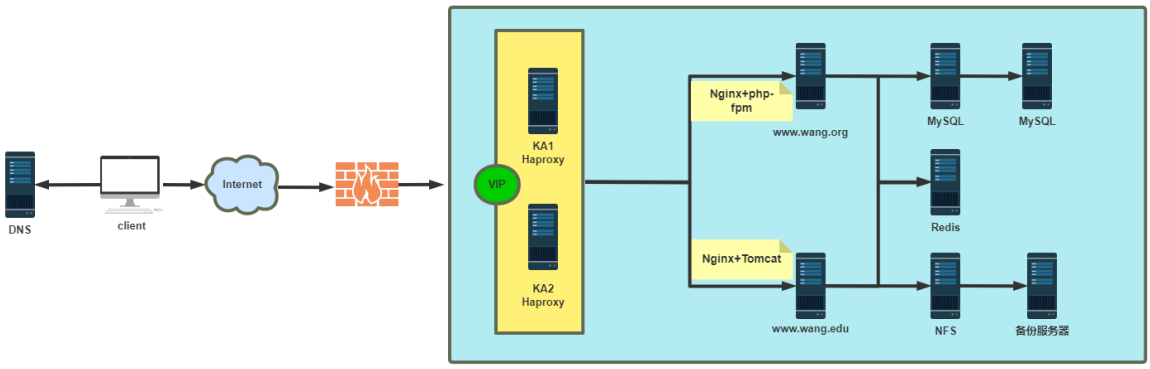



```
#在两个Zabbix Server 使用下面相同的配置
[root@ka1 ~]#grep -i  SourceIP=  /etc/zabbix/zabbix_server.conf
SourceIP=10.0.0.10
[root@ka1 ~]#grep -i '^server=' /etc/zabbix/zabbix_agentd.conf
Server=127.0.0.1,10.0.0.10

[root@ka1 ~]#systemctl enable zabbix-server.service
[root@ka2 ~]#systemctl disable zabbix-server.service

#keepalived相关配置
#ka1节点配置
[root@ka1 ~]#cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   notification_email {
       root@wangxiaochun.com
```

```
   }
   notification_email_from 29308620@qq.com
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   router_id ka1.wang.org
   vrrp_skip_check_adv_addr
   #vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
   vrrp_mcast_group4 230.6.6.6
}
#指定检测脚本
vrrp_script check_zabbix_server{
    script "/usr/bin/killall -0 zabbix_server"
    interval 1
    weight -30
    fall 2
    rise 2
    timeout 2
}
include  /etc/keepalived/conf.d/*.conf

[root@ka1 ~]#cat /etc/keepalived/conf.d/vip_zabbix.conf
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 66
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
    #notify_master "systemctl start zabbix-server"
    #notify_backup "systemctl stop zabbix-server"
    track_script {
        check_zabbix_server
    }
}

#ka2节点框配置
[root@ka2 ~]#cat /etc/keepalived/conf.d/vip_zabbix.conf
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 66
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 123456
    }
    virtual_ipaddress {
        10.0.0.10/24 dev eth0 label eth0:1
    }
```

```
    notify_master "systemctl start zabbix-server"
    notify_backup "systemctl stop zabbix-server"
    track_script {
    #    check_zabbix_server  #在ka2节点不能启用脚本,否则会导致ka2节点也降低优先级,从而切换失败
    }
}
```

6 综合实战案例



- 编译安装 HAProxy 新版 LTS 版本，编译安装 Keepalived
- 开启HAProxy多线程，线程数与CPU核心数保持一致,并绑定CPU核心
- 因业务较多避免配置文件误操作，需要按每业务一个配置文件并统一保存至/etc/haproxy/conf.d目录中
- 基于ACL实现单IP多域名负载功能,两个域名的业务: www.wang.org 和 www.wang.net
- 实现MySQL主从复制，并通过HAProxy对MySQL进行四层反向代理
- 对 www.wang.net 域名基于HAProxy+Nginx+PHP+MySQL+Redis，实现phpMyadmin的PHP应用,并实现Session会话保持统一保存到Redis
- 对 www.wang.org 域名基于HAProxy+Nginx+Tomcat+MySQL，并实现Jpress的JAVA应用