

Keepalived部署与配置详解 - 昀溪 - 博客园

cnblogs.com/rexcheny/p/10778567.html

Keepalive详解

工作原理

Keepalived本质就是为ipvs服务的，它也不需要共享存储。IPVS其实就是一些规则，Keepalived主要的任务就是去调用ipvsadm命令，来生成规则，并自动实现将用户需要访问的地址转移到可用LVS节点实现。所以keepalived的高可用是属于具有很强针对性的高可用，它和corosync这种通用性HA方案不同。

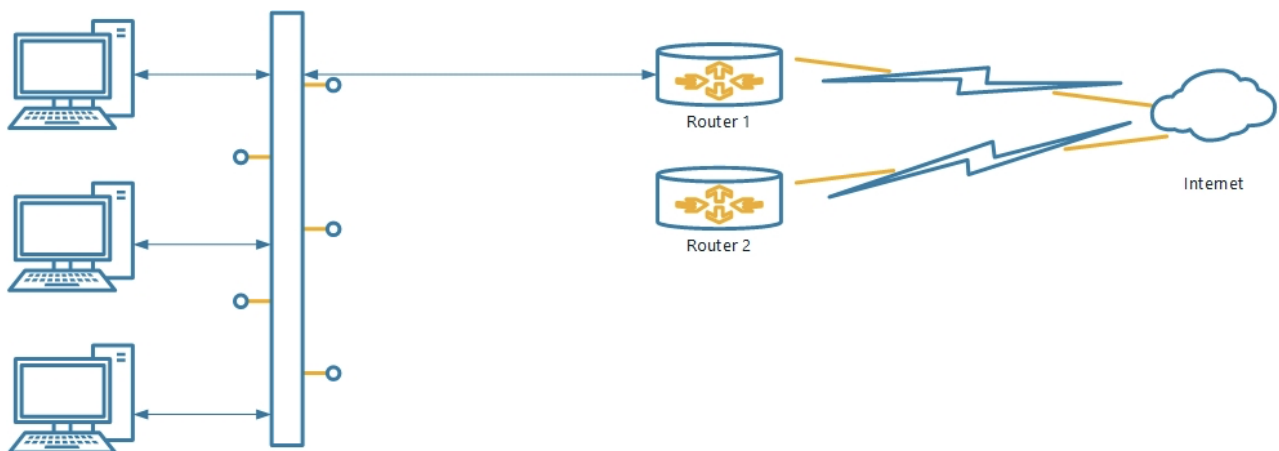
Keepalived的主要目的就是它自身启动为一个服务，它工作在多个LVS主机节点上，当前活动的节点叫做Master备用节点叫做Backup，Master会不停的向Backup节点通告自己的心跳，这种通告是基于VRRP协议的。Backup节点一旦接收不到Master的通告信息，它就会把LVS的VIP拿过来，并且把ipvs的规则也拿过来，在自己身上生效，从而替代Master节点。

Keepalived除了可以监控和转移LVS资源之外，它还可以直接配置LVS而不需要直接使用ipvsadm命令，因为它可以调用，也就是说在LVS+KEEPALIVED模型中，你所有的工作在Keepalived中配置就可以了，而且它还有对后端应用服务器健康检查的功能。

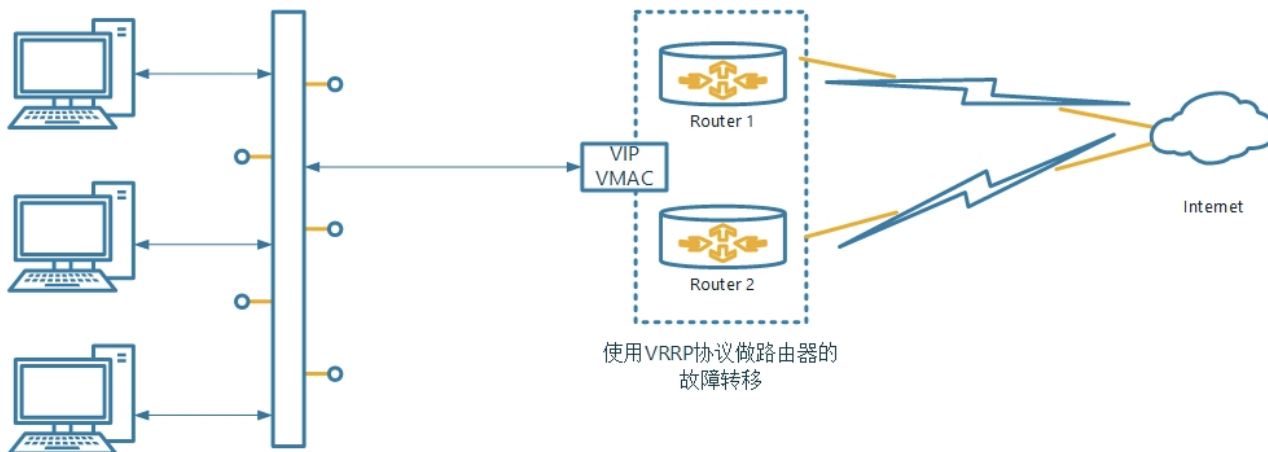
直接一句话Keepalived就是VRRP协议的实现，该协议是虚拟冗余路由协议。

VRRP工作原理简述

那么这个VRRP协议是干嘛用呢？传统上来说我们通过一个路由器上网，如果故障那就不能用了，如果使用2个路由器，有一个故障你就需要手动的设置客户端切换到另外的路由器上，或者使用ARP客户端也可以实现，但总之部署比较麻烦不利于管理，就像下图：



有没有一种办法可以自动转移而省去手动配置呢？我们就可以通过VRRP协议来实现路由器的故障转移。如下图：



这里有个问题，VRRP提供一个VIP，它可以来设定那个路由器是活动节点，然后出现故障进行切换，VIP也随之对应到新的路由器上，但是内网是用过MAC地址来寻址的，虽然VIP对应到了新的路由器上，可是MAC变了，客户端的ARP表也没有更新，所以还是用不了，为了解决这个问题VRRP不但提供VIP还提供VMAC地址，这个VMAC地址是VRRP单独申请的，大家都可以正常使用。

故障切换的时候虽然改变了后端路由器，但是由于客户端使用的是VIP和VMAC地址，这样就不会有任何影响了。

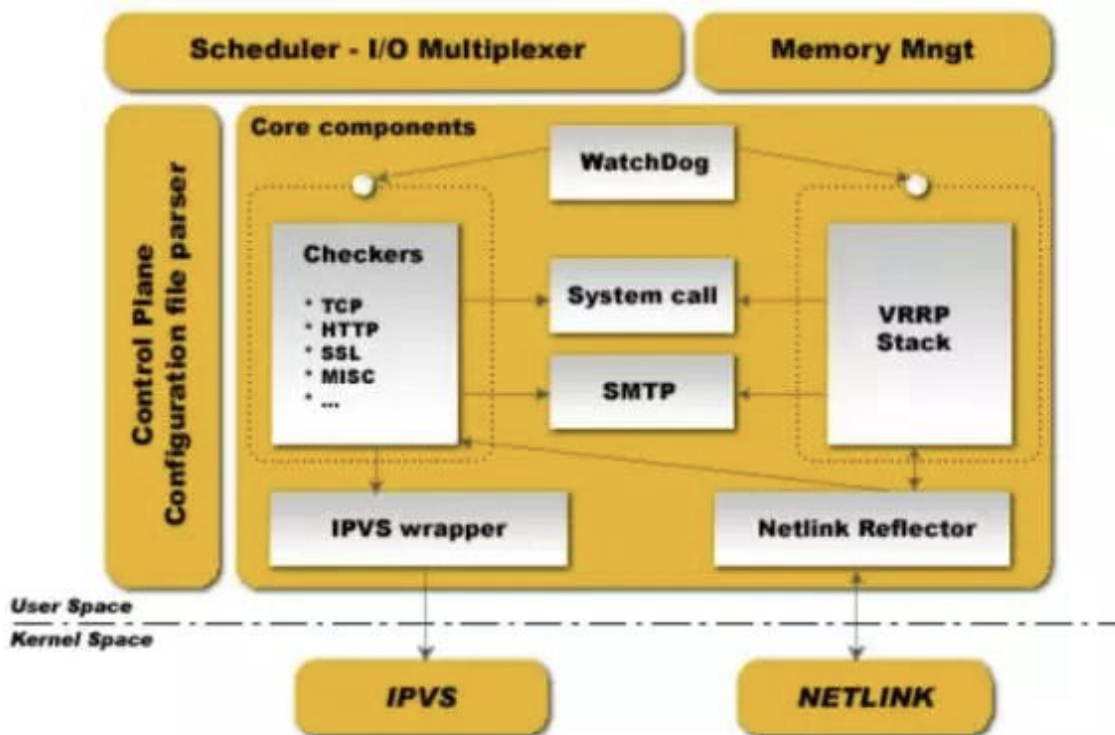
所以Keepalived就是在Linux系统上提供了VRRP功能，当然还提供了服务监控功能，比如监控后端服务器的健康检查、LVS服务可用性检查。

VRRP的工作过程是这样的：

1. 虚拟路由器中的路由器根据优先级选举出Master，Master路由器通过发送免费ARP报文，将自己的虚拟MAC地址通告给与它连接的设备。
2. Master路由器周期性发送VRRP报文，以公布自己的配置信息（优先级等）和工作状态
3. 如果Master故障，虚拟路由器中的Backup路由器将根据优先级重新选举新的Master
4. 虚拟路由器状态切换时，Master路由器由一台设备切换会另外一台设备，新的Master路由器只是简单的发送一个携带虚拟MAC地址和虚拟IP的免费ARP报文，这样就可以更新其他设备中缓存的ARP信息
5. Backup路由器的优先级高于Master时，由Backup的工作方式（抢占式或者非抢占式）决定是否重新选举Master。

VRRP还支持认证，就是为了防止随意一个VRRP设备加入到当前的虚拟路由组离来，它提供无认证、简单8位字符串认证和MD5认证（该认证方式Keepalive不支持）。

Keepalive软件结构



Keepalived启动后以后会有一个主进程Master，它会生成还有2个子进程，一个是VRRP Stack负责VRRP（也就是VRRP协议的实现）、一个是Checkers负责IPVS的后端的应用服务器的健康检查，当检测失败就会调用IPVS规则删除后端服务器的IP地址，检测成功了再加回来。当检测后端有失败的情况可以使用SMTP通知管理员。另外VRRP如果检测到另外一个Keepalive失败也可以通过SMTP通知管理员。

Control Plane：这个就是主进程，主进程的功能是分析配置文件，读取、配置和生效配置文件，指挥那2个子进程工作。

WatchDog：看门狗，这个是Linux系统内核的一个模块，它的作用是帮助主进程盯着那2个子进程，因为主进程并不负责具体工作，具体工作都是子进程完成的。如果子进程挂了，那Keepalived就不完整了，所以那2个子进程会定期的向主进程打开的一个内部Unix Socket文件写心跳信息。如果有某个子进程不写信息了，它就会重启子进程，主进程就是让WatchDog来监控子进程的。下面我们就使用Keepalive来做LVS的高可用讲解。关于后端服务器上的设置我这里就不说了请看另外一篇博文。

Keepalive安装和配置

| 服务器 | IP地址 | 角色 |
|-------|-------------------------------------|---------------|
| Srv01 | 172.16.42.100 VIP: 172.16.42.111 | LVS+Keepalive |
| Srv02 | 172.16.42.101 VIP: 192.168.100.1 | LVS+Keepalive |
| Srv03 | 172.16.42.102 VIP: 172.16.42.111 | Nginx |

| 服务器 | IP地址 | 角色 |
|-------|-------------------------------------|-------|
| Srv04 | 172.16.42.103 VIP: 172.16.42.111 | Nginx |

先决条件

1. 禁用SELinux、清除iptables规则、关闭防火墙。就算因某种原因不能清除iptables规则，那么你需要增加一条规则放行多播
2. 各个节点时间同步，启用时间同步服务 `systemctl start chronyd`
3. 确保Keepalive使用的网卡开启了多播，如下图：

```
[root@Srv01 ~]#
[root@Srv01 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.42.100 netmask 255.255.255.0 broadcast 172.16.42.255
    inet6 fe80::30f1:5120:28b7:6843 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::2e95:4b05:1e10:f7f prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:37:ee:90 txqueuelen 1000 (Ethernet)
    RX packets 6201 bytes 4081651 (3.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3622 bytes 387453 (378.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

如果没有开启，可以使用该命令打开 `ip link set multicast on dev ens33`，ens33是网卡名称。

安装keepalive

之间通过yum安装即可 `yum install -y keepalived`。我这里使用的是阿里云的源，它默认就在里面，如下图：

```
[root@Srv01 ~]# yum search keepalive
Failed to set locale, defaulting to C
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com

===== N/S matched: keepalive =====
keepalived.x86_64 : Load balancer and high availability service

Name and summary matches only, use "search all" for everything.
[root@Srv01 ~]#
[root@Srv01 ~]#
[root@Srv01 ~]#
```

在2个节点都安装。

| 文件 | 说明 |
|--------------------------------------------|-------|
| /usr/sbin/keepalived | 二进制程序 |
| /etc/keepalived/keepalived.conf | 配置文件 |
| /usr/lib/systemd/system/keepalived.service | 服务文件 |


```

global_defs {

    notification_email {

        acassen@firewall.loc
    }

    notification_email_from Alexandre.Cassen@firewall.loc

    smtp_server 192.168.200.1
    smtp_connect_timeout 30

    router_id LVS_DEVEL

    vrrp_mcast_group4 224.0.0.18
    vrrp_mcast_group6 ff02::12
}

vrrp_script SCRIPT_NAME {

}

vrrp_instance VI_1 {

    state MASTER

    interface ens33

    virtual_router_id 51

    use_vmac XX:XX:XX:XX:XX

    track_interface {
        eth0
        ens33
    }

    priority 100

    advert_int 1

    authentication {
        auth_type PASS
        auth_pass 1111
    }

    virtual_ipaddress {

        192.168.200.16/24 dev eth1

        192.168.200.17/24 dev label eth1:1
    }

    virtual_routes {

```

```

    192.168.110.0/24 dev eth2
}

nopreempt|preempt

preempt delay 300

track_script {

}

notify_master ""
notify_backup ""
notify_fault ""
}

virtual_server 10.10.10.2 1358 {
    delay_loop 6

    lb_algo rr|wrr|lc|wlc|lblc|sh|dh

    lb_kind NAT|DR|TUN

    nat_mask 255.255.255.0

    persistence_timeout 50

    protocol TCP

    sorry_server 192.168.200.200 1358

    real_server 192.168.200.2 1358 {

        weight 1

        MISC_CHECK {}

        SMTP_CHECK {}

        TCP_CHECK {
            connect_port <PORT>

            bindto <IP>

            connect_timeout 3
        }

        SSL_GET {}

        notify_up "这里写的是路径，如果脚本后有参数，整体路径+参数引起来"
        notify_down "/PATH/SCRIPTS.sh 参数"
    }
}

```

```

HTTP_GET {

    url {

        path /testurl/test.jsp

        digest 640205b7b0fc66c1ea91c463fac6334d

        status_code 200
    }
    url {
        path /testurl2/test.jsp
        digest 640205b7b0fc66c1ea91c463fac6334d
    }
    url {
        path /testurl3/test.jsp
        digest 640205b7b0fc66c1ea91c463fac6334d
    }

    connect_port <PORT>

    bindto <IP>

    connect_timeout 3

    nb_get_retry 3

    delay_before_retry 3
}

}

real_server 192.168.200.3 1358 {
    weight 1
    HTTP_GET {
        url {
            path /testurl/test.jsp
            digest 640205b7b0fc66c1ea91c463fac6334c
        }
        url {
            path /testurl2/test.jsp
            digest 640205b7b0fc66c1ea91c463fac6334c
        }
        connect_timeout 3
        nb_get_retry 3
        delay_before_retry 3
    }
}
}

```

配置Srv01和Srv02

配置VRRP部分

Srv01上的keepalived.conf


```

global_defs {
    notification_email {
        acassen@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id srv01
}

vrrp_instance VI_1 {
    state MASTER
    interface ens33
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    virtual_ipaddress {
        172.16.42.111/24 brd 172.16.42.111 dev ens33 label ens33:0
    }
    preempt delay 60
}

```

Srv02上的keepalived.conf，唯一不同的就是state、priority以及router_id。

```

global_defs {
    notification_email {
        acassen@firewall.loc
    }
    notification_email_from Alexandre.Cassen@firewall.loc
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id srv02
}

vrrp_instance VI_1 {
    state BACKUP
    interface ens33
    virtual_router_id 51
    priority 90
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    virtual_ipaddress {
        172.16.42.111/24 brd 172.16.42.111 dev ens33 label ens33:0
    }
    preempt delay 60
}

```

启动2个节点，启动后会自动配置ens33:0这个子接口的虚拟IP

```

[root@Srv01 ~]#
[root@Srv01 ~]# systemctl start keepalived
[root@Srv01 ~]#
[root@Srv01 ~]#
[root@Srv01 ~]#
[root@Srv01 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.42.100 netmask 255.255.255.0 broadcast 172.16.42.255
    inet6 fe80::30f1:5120:28b7:6843 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::2e95:4b05:1e10:f7f prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:37:ee:90 txqueuelen 1000 (Ethernet)
    RX packets 8691 bytes 6121924 (5.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4774 bytes 537997 (525.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.42.111 netmask 255.255.255.0 broadcast 172.16.42.111
    ether 00:50:56:37:ee:90 txqueuelen 1000 (Ethernet)

```

在主节点上你通过 `systemctl status keepalived` 看不到它到底是什么角色，不过在 BACKUP节点上你可以看到，但是你在主节点日志中 `cat /var/log/message` 里可以看到 Srv01进入到MASTER状态，如下图：

```

Apr 21 19:20:31 Srv01 chronyd[6461]: System clock wrong by 8.391473 seconds, adjustment started
Apr 21 19:21:24 Srv01 systemd: Starting LVS and VRRP High Availability Monitor...
Apr 21 19:21:24 Srv01 Keepalived[28173]: Starting Keepalived v1.3.5 (03/19,2017), git commit v1.3.5-6-g6fa32f2
Apr 21 19:21:24 Srv01 Keepalived[28173]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 21 19:21:24 Srv01 systemd: PID file /var/run/keepalived.pid not readable (yet?) after start.
Apr 21 19:21:24 Srv01 Keepalived[28174]: Starting Healthcheck child process, pid=28175
Apr 21 19:21:24 Srv01 Keepalived[28174]: Starting VRRP child process, pid=28176
Apr 21 19:21:24 Srv01 systemd: Started LVS and VRRP High Availability Monitor.
Apr 21 19:21:24 Srv01 Keepalived_healthcheckers[28175]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: Registering Kernel netlink reflector
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: Registering Kernel netlink command channel
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: Registering gratuitous ARP shared channel
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) removing protocol VIPs.
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: Using LinkWatch kernel netlink reflector...
Apr 21 19:21:24 Srv01 Keepalived_vrrp[28176]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]
Apr 21 19:21:25 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) Transition to MASTER STATE
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) Entering MASTER STATE
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) setting protocol VIPs.
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:26 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:31 Srv01 Keepalived_vrrp[28176]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 172.16.42.111
Apr 21 19:21:31 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:31 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:31 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:31 Srv01 Keepalived_vrrp[28176]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 19:21:37 Srv01 chronyd[6461]: Can't synchronise: no selectable sources
Apr 21 19:21:48 Srv01 chronyd[6461]: Source 51.75.17.219 replaced with 78.46.102.180
[root@Srv01 ~]#

```

查看Srv02的状态

```

[root@Srv02 ~]# systemctl status keepalived
● 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 2019-04-21 19:21:28 CST; 2min 6s ago
     Process: 7583 ExecStart=/usr/sbin/keepalived $KEEPALIVED_OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 7584 (keepalived)
      CGroup: /system.slice/keepalived.service
              └─7584 /usr/sbin/keepalived -D
                  └─7585 /usr/sbin/keepalived -D
                      └─7586 /usr/sbin/keepalived -D

Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: Registering Kernel netlink reflector
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: Registering Kernel netlink command channel
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: Registering gratuitous ARP shared channel
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: Opening file '/etc/keepalived/keepalived.conf'.
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: VRRP_Instance(VI_1) removing protocol VIPs.
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: Using LinkWatch kernel netlink reflector...
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: VRRP_Instance(VI_1) Entering BACKUP STATE
Apr 21 19:21:28 Srv02 Keepalived_vrrp[7586]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]
Apr 21 19:21:28 Srv02 Keepalived_healthcheckers[7585]: Initializing ipvs
Apr 21 19:21:28 Srv02 Keepalived_healthcheckers[7585]: Opening file '/etc/keepalived/keepalived.conf'.
[root@Srv02 ~]#

```

那么你通过停止Srv01上的keepalived服务就看到MASTER会被转移到Srv02上。

使用该命令查看VRRP通告 `tcpdump -i ens33 -nn host 224.0.0.18`，你在2台主机都会看到相同的信息。

```

[root@Srv01 ~]#
[root@Srv01 ~]# tcpdump -i ens33 -nn host 224.0.0.18
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
19:41:40.400440 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:41.403018 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:42.405026 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:43.406920 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:44.409507 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:45.411415 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:46.414147 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:47.417426 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
19:41:48.420715 IP 172.16.42.100 > 224.0.0.18: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20
^C

```

Srv01使用真实物理IP对该地址进行发送通告，那么Srv02也会收到，如果Srv01宕机，那么Srv02就会使用自己的物理IP向该地址发送通告，由于Srv01已经宕机那么此时Srv02的优先级就是最高的，所以Srv02就变成了MASTER。

配置LVS部分

这里只是用了LVS来说明如何配置Keepalived，如果要看完整内容请移步[使用Keepalived构建LVS高可用集群](#)

在keepalived.conf文件中增加下面的内容，2台服务器增加的内容一致，所以这里就写一份。

```

virtual_server 172.16.42.111 80 {
    delay_loop 6
    lb_algo rr
    lb_kind DR
    nat_mask 255.255.255.0
    persistence_timeout 0
    protocol TCP

    sorry_server 192.168.200.200 1358

    real_server 172.16.42.102 80 {
        weight 1

        notify_up "/usr/local/notify.sh up"
        notify_down "/usr/local/notify.sh down"
        HTTP_GET {

            url {
                path /index.html

                status_code 200
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }

    real_server 172.16.42.103 80 {
        weight 1

        notify_up "/usr/local/notify.sh up"
        notify_down "/usr/local/notify.sh down"
        HTTP_GET {

            url {
                path /index.html

                status_code 200
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}

```

这里的notify_up|down脚本我写的很简单就是为了使用一下这个功能，内容如下：

```

#!/bin/bash

if [ $1 == "up" ]; then
    echo "Srv02 is UP" > /tmp/notify.txt
elif [ $1 == "down" ]; then
    echo "Srv02 is DOWN" > /tmp/notify.txt
fi

```

重启Keepalived服务之后你就可以通过 `ipvsadm -Ln` 查看ipvs规则了，这些规则在2台服务器上都会有，如下图：

```
[root@Srv01 src]# ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  172.16.42.111:80 rr persistent 50
  -> 172.16.42.102:80              Route    1      0          0
  -> 172.16.42.103:80              Route    1      0          0
[root@Srv01 src]#
```

测试访问

使用下面的命令快速访问 `for i in {1..20}; do curl http://172.16.42.111/ | grep "Srv0" --color ; done`

```
rex.chen@yuns-MBP:~/Downloads$
rex.chen@yuns-MBP:~/Downloads$ for i in {1..20}; do curl http://172.16.42.111/ | grep "Srv0" --color ; done
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  626  100  626    0     0  18411      0 --:--:-- --:--:-- --:--:-- 18411
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  626  100  626    0     0  18411      0 --:--:-- --:--:-- --:--:-- 18411
<h1>Welcome to nginx! This is Srv03</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  626  100  626    0     0  17388      0 --:--:-- --:--:-- --:--:-- 17388
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  626  100  626    0     0  18969      0 --:--:-- --:--:-- --:--:-- 18969
<h1>Welcome to nginx! This is Srv03</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100  626  100  626    0     0  41733      0 --:--:-- --:--:-- --:--:-- 44714
```

可以看到2台服务器交替，因为我们使用的rr调度算法。

故障转移测试

连续访问VIP，然后停止Srv01上面的keepalived服务，这就意味着Srv01也就是失去了VIP，然后观察请求情况以及是否触发之前设定的脚本。

```

rex.chen@yuns-MBP:~/Downloads$
rex.chen@yuns-MBP:~/Downloads$ for i in {1..20}; do curl http://172.16.42.111/ | grep "Srv0" --color && sleep 0.5; done
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   36823      0  --:--:-- --:--:-- --:--:-- 39125
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   41733      0  --:--:-- --:--:-- --:--:-- 41733
<h1>Welcome to nginx! This is Srv03</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   17388      0  --:--:-- --:--:-- --:--:-- 17388
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   41733      0  --:--:-- --:--:-- --:--:-- 41733
<h1>Welcome to nginx! This is Srv03</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0    320      0  0:00:01 0:00:01 --:--:-- 320
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   36823      0  --:--:-- --:--:-- --:--:-- 39125
<h1>Welcome to nginx! This is Srv03</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100  626  100    626    0     0   34777      0  --:--:~ --:~:~ --:~:~ 36823
<h1>Welcome to nginx! This is Srv04</h1>
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed

```

这里就是我停止Srv01
上keepalived服务的
时候，所以出现了一个
耗时，所以
keepalived切换
MASTER了

在Srv01上查看脚本执行情况

```

[root@Srv01 src]#
[root@Srv01 src]# systemctl stop keepalived
[root@Srv01 src]#
[root@Srv01 src]# cat /tmp/notify.txt
Srv01 is DOWN
[root@Srv01 src]#
[root@Srv01 src]#

```

查看Srv02上面的日志

```

Apr 21 20:05:25 Srv02 chronyd[6452]: Can't synchronise: no majority
Apr 21 20:06:29 Srv02 chronyd[6452]: Selected source 51.75.17.219
Apr 21 20:06:29 Srv02 chronyd[6452]: System clock wrong by 12.616301 seconds, adjustment started
Apr 21 20:07:34 Srv02 chronyd[6452]: System clock wrong by 4.062832 seconds, adjustment started
Apr 21 20:08:26 Srv02 Keepalived_vrrp[28067]: VRRP_Instance(VI_1) Transition to MASTER STATE
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: VRRP_Instance(VI_1) Entering MASTER STATE
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: VRRP_Instance(VI_1) setting protocol VIPs.
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 172.16.42.111
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:27 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:32 Srv02 Keepalived_vrrp[28067]: Sending gratuitous ARP on ens33 for 172.16.42.111
Apr 21 20:08:32 Srv02 Keepalived_vrrp[28067]: VRRP_Instance(VI_1) Sending/queueing gratuitous ARPs on ens33 for 172.16.42.111

```