

KeepAlived+LVS_NAT模式

环境准备：

虚拟机4台： 2台lvs主机， 2台web服务器

IP地址规划：

lvs主节点：

ens33 (外网) : 192.168.189.171

ens37 (内网) : 192.168.1.10

VIP: 192.168.189.191

DIP: 192.168.1.100

lvs备节点：

ens33 (外网) : 192.168.189.172

ens37 (内网) : 192.168.1.12

VIP: 192.168.189.191

DIP: 192.168.1.100

注意： VIP和DIP不用配置，是由keepalived软件自动生成的

RS1:

ens33 (内网) : 192.168.1.21 网关: 192.168.1.100

RS2:

ens33 (内网) : 192.168.1.22 网关: 192.168.1.100

部署web服务

所有节点安装httpd软件

并创建不同的页面，用来区分两台web服务器

```
1 # yum -y install httpd
2 # systemctl start httpd
```

创建页面

```
1 web1:
2 # vim /var/www/html/index.html
3 web1
4
5 web2:
6 # vim /var/www/html/index.html
7 web2
```

部署lvs主机

lvs节点操作:

所有的lvs主机都执行如下操作

1. 关闭防火墙

```
1 # systemctl stop firewalld
2 # setenforce 0
```

开启ipvs转发

```
1 # vim /etc/sysctl.conf
2 net.ipv4.ip_forward = 1
3 [root@node-172 ~]# sysctl -p
```

2. 安装keepalived和ipvsadm

```
1 # yum -y install keepalived ipvsadm
```

3. 配置keepalived

lvs主节点和备份节点配置基本相同，只是在角色和优先级上不同

主节点配置

```
1 [root@node-171 ~]# vim /etc/keepalived/keepalived.conf
```

```
1 ! Configuration File for keepalived
2
3 global_defs {
4     notification_email {
5         acassen@firewall.loc
6         failover@firewall.loc
7         sysadmin@firewall.loc
8     }
9     notification_email_from Alexandre.Cassen@firewall.loc
10    smtp_server 192.168.200.1
11    smtp_connect_timeout 30
12    router_id LVS_DEVEL
13    vrrp_skip_check_adv_addr
```

```

14 ! vrrp_strict # 注释掉该参数
15 vrrp_garp_interval 0
16 vrrp_gna_interval 0
17 }
18
19 vrrp_instance VI_1 {
20     state MASTER # 初始的角色, MASTER为主节点, BACKUP为备节点
21     interface ens33 # 默认生成VIP的网卡
22     virtual_router_id 51 # 虚拟路由广播的ID
23     priority 100 # 优先级, 值越高, 优先级越高, 备节点优先级要低于主节点
24     advert_int 1
25     authentication {
26         auth_type PASS
27         auth_pass 1111
28     }
29     virtual_ipaddress {
30         192.168.189.191 dev ens33 # 要生成的VIP
31         192.168.1.100 dev ens37 # 要生成的DIP
32     }
33 }
34
35 virtual_server 192.168.189.191 80 { # 创建虚拟服务器, 定义的虚拟服务器的IP和端口
36     delay_loop 6 # 健康检查的时间间隔 单位 秒
37     lb_algo rr # 调度算法
38     lb_kind NAT # 工作模式
39     ! persistence_timeout 50 # 保持会话的时长, 为测试负载均衡效果需注释该项配置
40     protocol TCP # tcp协议
41
42     real_server 192.168.1.21 80 { # 为虚拟服务器添加的真实服务器, 配置的IP和端口
43         weight 1 # 权重值
44         HTTP_GET { # 健康检查策略, 有TCP_CHECK\HTTP_GET\SSL_GET
45             url {
46                 path / # 使用http获取的文件的路径
47                 status_code 200 # 判断状态码是否为200, 若不是200, 则判定为不健康
48             }
49             connect_timeout 3 # 连接超时时间 单位 秒
50             nb_get_retry 3 # 超时重试次数
51             delay_before_retry 3 # 重试的间隔时间, 单位 秒
52         }
53     }
54     real_server 192.168.1.22 80 { # 另一台真实服务器
55         weight 1
56         HTTP_GET {
57             url {
58                 path /
59                 status_code 200
60             }
61             connect_timeout 3
62             nb_get_retry 3
63             delay_before_retry 3
64         }
65     }
66 }
67 }
68

```

备节点不同的配置, 主要是“state”角色和“priority”优先级,其他的配置都一样

```

1 vrrp_instance VI_1 {
2     state BACKUP      # 初始的角色为BACKUP
3     interface ens33   # 默认生成VIP的网卡
4     virtual_router_id 51 # 虚拟路由广播的ID
5     priority 90        # 备节点优先级要低于主节点
6     advert_int 1
7     authentication {
8         auth_type PASS
9         auth_pass 1111
10    }
11    virtual_ipaddress {
12        192.168.189.191 dev ens33    # 要生成的VIP
13        192.168.1.100   dev ens37    # 要生成的DIP
14    }
15 }

```

4. 启动所有节点的keepalived

```

1 # systemctl start keepalived
2 # systemctl enable keepalived

```

5. 验证效果:

查看VIP和DIP是否正确生成

lvs主节点VIP和DIP成功生成

```

[root@node-171 ~]# ip a s
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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:f3:61:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.189.171/24 brd 192.168.189.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet 192.168.189.191/32 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::d409:7c08:a1c6:8e30/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:f3:61:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global noprefixroute ens37
        valid_lft forever preferred_lft forever
    inet 192.168.1.100/32 scope global ens37
        valid_lft forever preferred_lft forever
    inet6 fe80::aaac:7a84:8f45:4e42/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@node-171 ~]#

```

ens33上的VIP

ens37上的DIP

测试高可用性 (ip漂移)

停止主节点keepalived

```

1 [root@node-171 ~]# systemctl stop keepalived

```

备节点查看IP是否漂移，可以成功看到VIP漂移到了备节点

```
[root@node-172 ~]# ip a s
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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default ql
    link/ether 00:0c:29:ec:31:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.189.172/24 brd 192.168.189.255 scope global noprefixroute ens33
        valid_lft forever preferred_lft forever
    inet 192.168.189.191/32 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::1626:de47:3622:659e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default ql
    link/ether 00:0c:29:ec:31:fb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.1.255 scope global noprefixroute ens37
        valid_lft forever preferred_lft forever
    inet 192.168.1.100/32 scope global ens37
        valid_lft forever preferred_lft forever
    inet6 fe80::aaac:7a84:8f45:4e42/64 scope link tentative noprefixroute dadfailed
        valid_lft forever preferred_lft forever
    inet6 fe80::3225:c969:a2bd:a9d2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@node-172 ~]#
```

如此变实现了高可用，主节点宕机，VIP依然存在，用户依然可以访问，不会收到影响

测试负载均衡

查看调度表,可以看到所有的真实服务器说明节点都是正常的

```
1 [root@node-172 ~]# ipvsadm -L -n
2 IP Virtual Server version 1.2.1 (size=4096)
3 Prot LocalAddress:Port Scheduler Flags
4   -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
5 TCP 192.168.189.191:80 rr
6   -> 192.168.1.21:80              Masq    1      0      0
7   -> 192.168.1.22:80              Masq    1      0      0
```

访问VIP测试负载均衡效果，

```
C:\Users\liuzh>curl 192.168.189.191
web1

C:\Users\liuzh>curl 192.168.189.191
web2

C:\Users\liuzh>curl 192.168.189.191
web1

C:\Users\liuzh>curl 192.168.189.191
web2

C:\Users\liuzh>curl 192.168.189.191
web1

C:\Users\liuzh>_
```

可以看到负载均衡效果说明成功

keepalived+lvs最终验证效果主要就是两点的验证：

1. 高可用
2. 负载均衡

这两点效果都实现那么就配置成功了