一　keepalived

作用：

给电脑配VIP，自动配置LVS,自动后端主机健康检查

1，部署Keepalived服务

[root@web1 ~]# yum install -y keepalived

[root@web2 ~]# yum install -y keepalived

1）修改web1服务器Keepalived配置文件

[root@web1 ~]# vim /etc/keepalived/keepalived.conf

global\_defs {

notification\_email {

admin@tarena.com.cn //设置报警收件人邮箱

}

notification\_email\_from ka@localhost //设置发件人

smtp\_server 127.0.0.1 //定义邮件服务器

smtp\_connect\_timeout 30

router\_id web1 //设置路由ID号，必须是唯一的

}

vrrp\_instance VI\_1 {

state MASTER //主服务器为MASTER｜备份为BACKUP

interface eth0 //定义网络接口

virtual\_router\_id 50 //主辅VRID号必须一致(主辅路由组必须一致)分组组号

priority 100 //服务器优先级,优先级高优先获取VIP（默认100）

advert\_int 1　　　//间隔时间（单位秒），主辅对比priproty优先权，决定主服务器

authentication {

auth\_type pass

auth\_pass forlvs //主辅服务器密码必须一致（生产环境建议修改）

}

virtual\_ipaddress { 192.168.4.80 } //谁是主服务器谁获得该VIP（可以写多个vip）

}

2）启动服务

[root@web1 ~]# systemctl start keepalived

[root@web2 ~]# systemctl start keepalived

1. 配置防火墙和SELinux

启动keepalived会自动添加一个drop的防火墙（iptables）规则，需要清空！

[root@web1 ~]#iptables -nL　　　　＃查看iptables防火墙规则

[root@web1 ~]# iptables -F　　　　　#清空iptables规则

[root@web1 ~]# setenforce 0　　　　＃设置selinux为宽松模式

[root@web2 ~]# iptables -F

[root@web1 ~]# setenforce 0

1. 测试

--查看

[root@web1 ~]# ip addr show

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:e2:e7:84 brd ff:ff:ff:ff:ff:ff

inet 192.168.4.100/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet 192.168.4.80/32 scope global eth0　　　//vip

valid\_lft forever preferred\_lft forever

inet6 fe80::5e12:447e:8fba:9aa2/64 scope link

valid\_lft forever preferred\_lft forever

[root@web2 ~]# ip addr show

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:00:31:ac:a3 brd ff:ff:ff:ff:ff:ff

inet 192.168.4.200/24 brd 192.168.4.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet6 fe80::cf79:170b:4ae0:5705/64 scope link

valid\_lft forever preferred\_lft forever

－－客户端访问

客户端使用curl命令连接http://192.168.4.80，查看Web页面；关闭Web1服务器的网卡，客户端再次访问http://192.168.4.80，验证是否可以正常访问服务。

2，Keepalived+LVS服务器，

1. LVS1调度器安装与配置Keepalived，并启动服务

[root@proxy1 ~]# yum -y install keepalived

[root@proxy1 ~]# vim /etc/keepalived/keepalived.conf

global\_defs {

notification\_email {

admin@tarena.com.cn //设置报警收件人邮箱

}

notification\_email\_from ka@localhost //设置发件人

smtp\_server 127.0.0.1 //定义邮件服务器

smtp\_connect\_timeout 30

router\_id lvs1 //设置路由ID号(实验需要修改)

}

vrrp\_instance VI\_1 {

state MASTER //主服务器为MASTER（实验需要修改）

interface eth0 //定义网络接口（实验需要修改）

virtual\_router\_id 50 //主辅VRID号必须一致（实验需要修改）

priority 100 //服务器优先级（实验需要修改）

advert\_int 1

authentication {

auth\_type pass

auth\_pass forlvs //主辅服务器密码必须一致（实验需要修改）

}

virtual\_ipaddress { 192.168.4.15 } //配置VIP（实验需要修改）

}

virtual\_server 192.168.4.15 80 { //设置ipvsadm的VIP规则（实验需要修改）

delay\_loop 6

lb\_algo wrr //设置LVS调度算法为WRR，常用(RR,WRR,LC,WLC)

lb\_kind DR //设置LVS的模式为DR，常用(NAT,DR,TUN)

#persistence\_timeout 50

#注意这样的作用是保持连接，开启后，客户端在一定时间内始终访问相同服务器

protocol TCP

real\_server 192.168.4.100 80 { //设置后端web服务器真实IP

weight 1 //设置权重为1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.4.200 80 { //设置后端web服务器真实IP

weight 2 //设置权重为2

TCP\_CHECK {

connect\_timeout 3　　　//连接超时，时间秒

nb\_get\_retry 3　　　　　//重连次数，重连3次

delay\_before\_retry 3　　//延时3秒重连一次

}

}

}

[root@proxy1 ~]# systemctl start keepalived

[root@proxy2 ~]# yum -y install ipvsadm

[root@proxy1 ~]# ipvsadm -Ln #查看LVS规则

[root@proxy1 ~]# ip a s #查看VIP配置

1. LVS2调度器设置Keepalived

[root@proxy1 ~]# yum -y install keepalived

[root@proxy2 ~]# vim /etc/keepalived/keepalived.conf

global\_defs {

notification\_email {

root@localhost //设置报警收件人邮箱

}

notification\_email\_from ka@localhost //设置发件人

smtp\_server 127.0.0.1 //定义邮件服务器

smtp\_connect\_timeout 30

router\_id lvs2 //设置路由ID号（实验需要修改）

}

vrrp\_instance VI\_1 {

state BACKUP //从服务器为BACKUP（实验需要修改）

interface eth0 //定义网络接口（实验需要修改）

virtual\_router\_id 50 //主辅VRID号必须一致（实验需要修改）

priority 50 //服务器优先级（实验需要修改）

advert\_int 1

authentication {

auth\_type pass

auth\_pass forlvs //主辅服务器密码必须一致（实验需要修改）

}

virtual\_ipaddress { 192.168.4.15 } //设置VIP（实验需要修改）

}

virtual\_server 192.168.4.15 80 { //自动设置LVS规则（实验需要修改）

delay\_loop 6　　　　　//检查周期6秒

lb\_algo wrr //设置LVS调度算法为WRR

lb\_kind DR //设置LVS的模式为DR

# persistence\_timeout 50

#注意这样的作用是保持连接，开启后，客户端在一定时间内始终访问相同服务器

protocol TCP

#sorry\_server 127.0.0.1:80 //当后台主机服务器挂掉后，该调度器为MASTER时，显示本机的页面给客户机

real\_server 192.168.4.100 80 { //设置后端web服务器的真实IP

weight 1 //设置权重为1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.4.200 80 { //设置后端web服务器的真实IP

weight 2 //设置权重为2

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

[root@proxy2 ~]# systemctl start keepalived

[root@proxy2 ~]# ipvsadm -Ln #查看LVS规则

[root@proxy2 ~]# ip a s #查看VIP设置

1. 测试

客户端使用curl命令反复连接http://192.168.4.15，查看访问的页面是否会轮询到不同的后端真实服务器。

３　keepalived配置文件详解

! Configuration File for keepalived

global\_defs {

notification\_email {

root@localhost　　//管理员邮箱

}

notification\_email\_from [Alexandre.Cassen@firewall.loc](mailto:Alexandre.Cassen@firewall.loc) //设置发件人邮箱

smtp\_server 127.0.0.1 //邮箱服务器地址

smtp\_connect\_timeout 30　　//连接邮箱服务器超时时间

router\_id LVS\_DEVEL　　　　//路由id，每组的路由id是唯一的

vrrp\_skip\_check\_adv\_addr

vrrp\_strict

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

vrrp\_instance VI\_1 {

state MASTER　　//主服务器MASTER,从服务器BACKUP

interface eth0 //定义网络接口，网卡名称

virtual\_router\_id 51　//定义主辅VRID号必须一致（生产环境建议修改）

priority 100　　//定义服务器优先级(默认为100)

advert\_int 1　　//主辅服务器每个1秒互相对比priority值

authentication {

auth\_type PASS　　//keepalived只支持俩中认证，这里使用第二种简单密码认证，也就是预认证

auth\_pass 1111　　　//主辅认证密码（生产环境介意修改），最长支持八位

}

virtual\_ipaddress {　　　　//vip虚拟ip地址，可以配置多个

192.168.200.16

　　 192.168.200.17

}

}

virtual\_server 192.168.200.16 443 {　　　//vip,端口号

delay\_loop 6

lb\_algo wrr //常用算法(rr | wrr | lc | wlc)

lb\_kind DR //模式(NAT | DR | TUN),常用DR,NAT

#persistence\_timeout 50 //保持连接，开启后，客户端在一定时间内始终访问相同服务器,测试环境可注释掉，生产环境开启

protocol TCP　　　//协议类型TCP,只支持TCP，所以想想就好

#sorry\_server 127.0.0.1:80 //用于集群挂掉了，并且该调度器为MASTER时,提示的友好界面给客户机，比如提示：服务器正在维护等等友好信息...

real\_server 192.168.201.100 443 { //实际后端主机ip和端口号

weight 1　　//权重值（默认为1，最小值为1），权重越大，该后端主机承受的压力越大，提示：仅仅lb\_algo　选择wrr | wlc，有效

SSL\_GET {　　＃加密的https健康检查

url {

path /　　　　　　　　//检查根下的文件

digest ff20ad2481f97b1754ef3e12ecd3a9cc　　//对比码，如加密的md5

#status\_code 200 #返回状态码为200表示OK,上一行可用这个代替，具体情况看自己的需求

}

url {

path /mrtg/　　　　　//检查/mrtg/下的文件

digest 9b3a0c85a887a256d6939da88aabd8cd

}

connect\_timeout 3　　//连接超时时间，单位秒

nb\_get\_retry 3　　　　//重复连接次数

delay\_before\_retry 3　//延时3秒重连一次

}

　 }

real\_server 192.168.201.101 443 { //实际后端主机ip和端口号

...........

}

　．．．．．．

}

virtual\_server 192.168.200.17 1358 {

delay\_loop 6

lb\_algo wrr

lb\_kind DR

#persistence\_timeout 50

protocol TCP

sorry\_server 192.168.200.200 1358

real\_server 192.168.200.2 1358 {

weight 1

HTTP\_GET {　　　　＃普通的http健康检查

url {

path /testurl/test.jsp

digest 640205b7b0fc66c1ea91c463fac6334d

}

url {

path /testurl2/test.jsp

digest 640205b7b0fc66c1ea91c463fac6334d

}

url {

path /testurl3/test.jsp

digest 640205b7b0fc66c1ea91c463fac6334d

}

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

．．．．．．

｝

二　配置HAProxy负载平衡集群

做集群与nginx相似，HAProxy性能比nginx好一点，LVS性能是最高的

HAProxy支持正则没有nginx好，HAProxy与LVS不支持web

F5 BIGIP也是做负载均衡，集成成硬件，比较贵

1，部署HAProxy服务器

1）配置网络，安装软件

[root@haproxy ~]# echo 'net.ipv4.ip\_forward = 1' >> sysctl.conf //开启路由转发

[root@haproxy ~]# sysctl -p

[root@haproxy ~]# yum -y install haproxy

２）修改配置文件

[root@haproxy ~]# vim /etc/haproxy/haproxy.cfg

global

log 127.0.0.1 local2 ###[err warning info debug]

chroot /usr/local/haproxy　　＃定义工作目录

pidfile /var/run/haproxy.pid ###haproxy的pid存放路径

maxconn 4000 ###最大连接数，默认4000

user haproxy

group haproxy

daemon ###创建1个进程进入deamon模式运行，放后台

defaults

mode http ###默认的模式mode { tcp|http|health }，调度模式health基本不用

log global ###采用全局定义的日志

option dontlognull ###不记录健康检查的日志信息（默认值）

option httpclose ###每次请求完毕后主动关闭http通道

option httplog ###日志类别http日志格式

option forwardfor ###后端服务器可以从Http Header中获得客户端ip

option redispatch ###serverid服务器挂掉后强制定向到其他健康服务器

timeout connect 10000 #如果backend没有指定，默认为10s

timeout client 300000 ###客户端连接超时

timeout server 300000 ###服务器连接超时

maxconn 60000 ###最大并发连接数

retries 3 ###3次连接失败就认为服务不可用，也可以通过后面设置

listen stats

bind 0.0.0.0:1080 #监听端口

stats refresh 30s #统计页面自动刷新时间

stats uri /stats #统计页面url

stats realm Haproxy Manager #统计页面密码框上提示文本

stats auth admin:admin #统计页面用户名和密码设置

#stats hide-version #隐藏统计页面上HAProxy的版本信息

listen websrv-rewrite 0.0.0.0:80

cookie SERVERID rewrite　　＃在原有的cookie上加个标记，下面的cookie后跟的标记，

balance roundrobin　　　　　＃算法

server web1 192.168.2.100:80 cookie app1inst1 check inter 2000 rise 2 fall 5

server web2 192.168.2.200:80 cookie app1inst2 check inter 2000 rise 2 fall 5

＃check inter 时间 单位毫秒，间隔时间多久做健康检查

＃cookie 标记(自己手打字符串)　对cookie加上标记

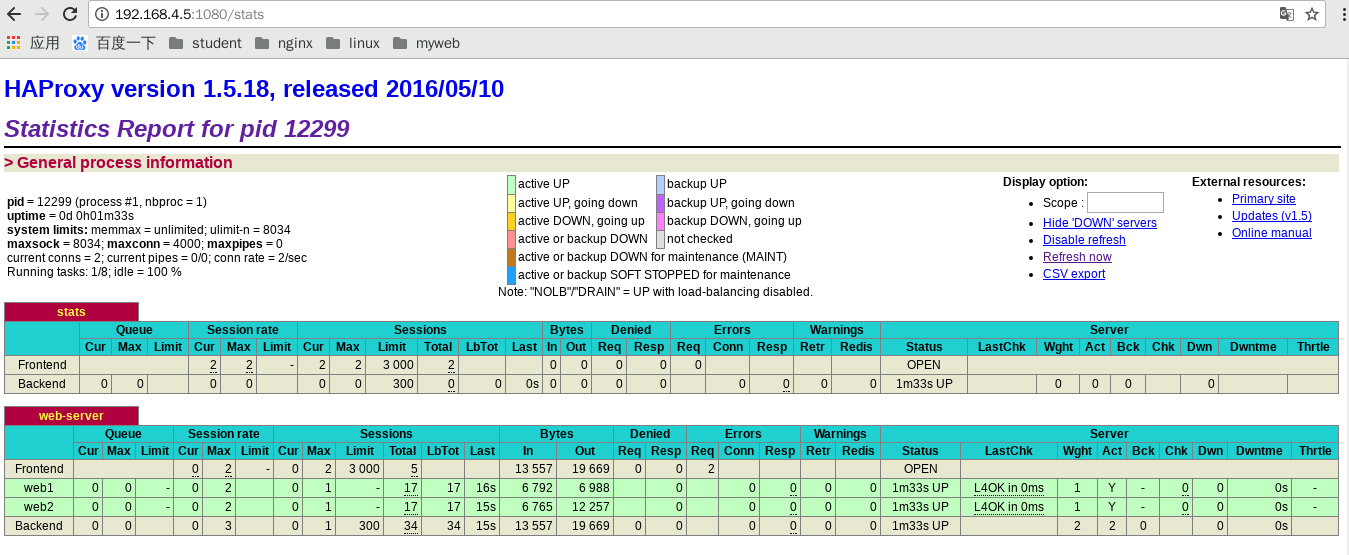
3）启动服务器并设置开机启动

[root@haproxy ~]# systemctl start haproxy

[root@haproxy ~]# systemctl enable haproxy

1. 测试

客户端配置与HAProxy相同网络的IP地址，并使用火狐浏览器访问http://192.168.4.5，测试调度器是否正常工作，客户端访问http://192.168.4.5:1080/stats测试状态监控页面是否正常。访问状态监控页的内容



备注：

Queue队列数据的信息（当前队列数量，最大值，队列限制数量）；

Session rate每秒会话率（当前值，最大值，限制数量）；

Sessions总会话量（当前值，最大值，总量，Lbtot: total number of times a server was selected选中一台服务器所用的总时间）；

Bytes（入站、出站流量）；

Denied（拒绝请求、拒绝回应）；

Errors（错误请求、错误连接、错误回应）；

Warnings（重新尝试警告retry、重新连接redispatches）；

Server(状态、最后检查的时间（多久前执行的最后一次检查）、权重、备份服务器数量、down机服务器数量、down机时长)。