keepalived+LVS

keepalived:三个功能(模块)

1. VRRP功能
2. 自动配置LVS(ipvsadm)

-A

-a

3.健康检查

案例1:keepalived高可用服务器

1.1问题

准备三台Linux服务器，两台做Web服务器，并部署Keepalived高可用软件，一台作为客户端主机，实现如下功能：

\*使用Keepalived实现web服务器的高可用

\*Web服务器IP地址分别为192.168.4.100和192.168.4.200

\*Web服务器的浮动VIP地址为192.168.4.80

\*客户端通过访问VIP地址访问Web页面

1.2方案

使用3台虚拟机,2台作为web服务器,并部署Keepalived,1台作为客户端,拓扑如图所示:

1.3步骤

步骤一:配置网络环境(如果在前面课程已经完成该配置,可以忽略)

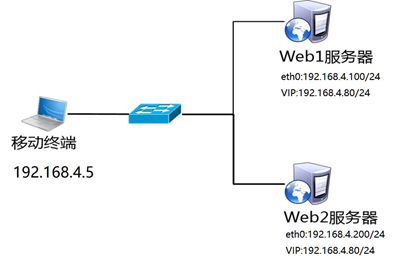
[1]设置web1服务器网络参数,配置web服务

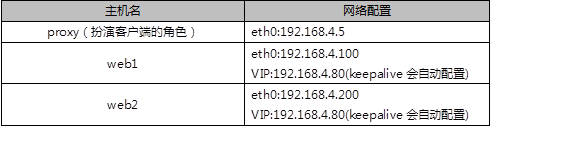
[root@web1 ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.100/24 connection.autoconnect yes

[root@web1 ~]# nmcli connection up eth0

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

[root@web1 ~]# echo "192.168.4.100" > /var/www/html/index.html

[root@web1 ~]# systemctl restart httpd



[2]设置web2服务器网络参数,配置web服务

[root@web2 ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.200/24 connection.autoconnect yes

[root@web2 ~]# nmcli connection up eth0

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

[root@web2 ~]# echo "192.168.4.200" > /var/www/html/index.html

[root@web2 ~]# systemctl restart httpd

[3]配置proxy主机网络参数(如果已经配置可以忽略)

[root@proxy ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.5/24 connection.autoconnect yes

[root@proxy ~]# nmcli connection up eth0

步骤二:安装keepaliced

注意:两台web做相同的配置

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

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

步骤三:部署keepalived服务

[1]修改web1服务器keepalive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（实验需要修改）

}

vrrp\_instance VI\_1 {

state MASTER //主服务器为MASTER（备服务器需要修改为BACKUP）

interface eth0 //定义网络接口

virtual\_router\_id 51 //主备服务器VRID号必须一致

priority 100 //服务器优先级,优先级高优先获取VIP

advert\_int 1

authentication {

auth\_type pass

auth\_pass 1111 //主备服务器密码必须一致

}

virtual\_ipaddress { //谁是主服务器谁获得该VIP（实验需要修改）

192.168.4.80

}

}

[2]修改web2服务器keepalived配置文件

[root@web2 ~]# 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 web2 //设置路由ID号（实验需要修改）

}

vrrp\_instance VI\_1 {

state BACKUP //备服务器为BACKUP（实验需要修改）

interface eth0 //定义网络接口

virtual\_router\_id 51 //主辅VRID号必须一致

priority 50 //服务器优先级（实验需要修改）

advert\_int 1

authentication {

auth\_type pass

auth\_pass 1111 //主辅服务器密码必须一致

}

virtual\_ipaddress { //谁是主服务器谁配置VIP（实验需要修改）

192.168.4.80

}

}

[3]启动服务

[root@web1 ~]# systemctl start keepalived

[root@web2 ~]# systemctl start keepalived

[4]配置防火墙和Selinux

启动keepalived会自动添加一个drop的防火墙规则,需要清空

[root@web1 ~]# iptables -F

[root@web1 ~]# setenforce 0

[root@web2 ~]# iptables -F

[root@web1 ~]# setenforce 0

步骤四:测试

[1]登录两台web服务器查看VIP信息

[root@web1 ~]# ip addr show eth0

[root@web2 ~]# ip addr show eth0

[2]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。

案例2:Keepalived+LVS服务器

2.1问题

使用Keepalived为LVS调度器提供高可用功能，防止调度器单点故障，为用户提供Web服务：

LVS1调度器真实IP地址为192.168.4.5

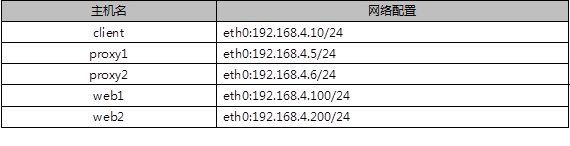
LVS2调度器真实IP地址为192.168.4.6

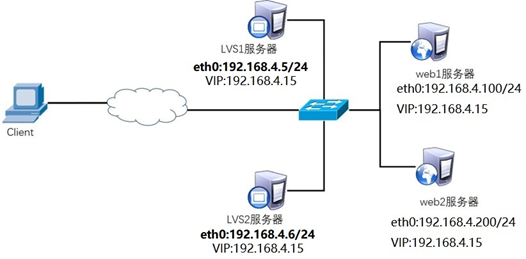
服务器VIP地址设置为192.168.4.15

真实Web服务器地址分别为192.168.4.100、192.168.4.200

使用加权轮询调度算法，真实web服务器权重不同

2.2方案

使用5台虚拟机，1台作为客户端主机、2台作为LVS调度器、2台作为Real Server，实验拓扑环境结构如图-2所示，基础环境配置如表-2所示



注意:所有主机需需要配置IP地址与有效的YUM源.

2.3步骤

步骤一:配置网络环境

[1]设置web1服务器的网路参数

[root@web1 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.100/24 connection.autoconnect yes

[root@web1 ~]# nmcli connection up eth0

接下来给web1配置VIP地址

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

[root@web1 ~]# cd /etc/sysconfig/network-scripts/

[root@web1 ~]# cp ifcfg-lo{,:0}

[root@web1 ~]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

注意：这里因为web1也配置与调度器一样的VIP地址，默认肯定会出现地址冲突。

写入这四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应。

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

#手动写入如下4行内容

net.ipv4.conf.all.arp\_ignore = 1

net.ipv4.conf.lo.arp\_ignore = 1

net.ipv4.conf.lo.arp\_announce = 2

net.ipv4.conf.all.arp\_announce = 2

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

重启网络服务，设置防火墙与SELinux

[root@web1 ~]# systemctl stop NetworkManager

[root@web1 ~]# systemctl disable NetworkManager

[root@web1 ~]# systemctl restart network

[root@web1 ~]# ifconfig

[root@web1 ~]# systemctl stop firewalld

[root@web1 ~]# setenforce 0

[2]设置Web2服务器的网络参数

[root@web2 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.200/24 connection.autoconnect yes

[root@web2 ~]# nmcli connection up eth0

接下来给web2配置VIP地址

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

[root@web2 ~]# cd /etc/sysconfig/network-scripts/

[root@web2 ~]# cp ifcfg-lo{,:0}

[root@web2 ~]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

注意：这里因为web2也配置与代理一样的VIP地址，默认肯定会出现地址冲突。

写入这四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应。

[root@web2 ~]# vim /etc/sysctl.conf

#手动写入如下4行内容

net.ipv4.conf.all.arp\_ignore = 1

net.ipv4.conf.lo.arp\_ignore = 1

net.ipv4.conf.lo.arp\_announce = 2

net.ipv4.conf.all.arp\_announce = 2

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

重启网络服务，设置防火墙与SELinux

[root@web2 ~]# systemctl stop NetworkManager

[root@web1 ~]# systemctl disable NetworkManager

[root@web2 ~]# systemctl restart network

[root@web2 ~]# ifconfig

[root@web2 ~]# systemctl stop firewalld

[root@web2 ~]# setenforce 0

[3]配置proxy1主机的网络参数(不配置VIP，由keepalvied自动配置)

[root@proxy1 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.5/24 connection.autoconnect yes

[root@proxy1 ~]# nmcli connection up eth0

[4]配置proxy2主机的网络参数(不配置VIP，由keepalvied自动配置)

注意：按照前面的课程环境，默认没有该虚拟机，需要重新建一台虚拟机proxy2。

[root@proxy2 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.6/24 connection.autoconnect yes

[root@proxy2 ~]# nmcli connection up eth0

步骤二:配置后台web服务

[1]安装软件,自定义web页面,(web1和web2)

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

[root@web1 ~]# echo "192.168.4.100" > /var/www/html/index.html

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

[root@web2 ~]# echo "192.168.4.200" > /var/www/html/index.html

[2]启动web服务器软件

[root@web1 ~]# systemctl start httpd ; systemctl enable httpd

[root@web2 ~]# systemctl start httpd ; systemctl enable httpd

步骤三:调度器安装keepalived与ipvsadm软件

注意:两台LVS调度执行相同的操作.

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

[root@proxy1 ~]# systemctl enable keepalived

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

[root@proxy1 ~]# ipvsadm -C

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

[root@proxy2 ~]# systemctl enable keepalived

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

[root@proxy2 ~]# ipvsadm -C

步骤四:部署keepalived实现LVS-DR模式调度器的高可用

[1]LVS1调度器设置Keepalived，并启动服务

[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

interface eth0 //定义网络接口

virtual\_router\_id 51 //主辅VRID号必须一致

priority 100 //服务器优先级

advert\_int 1

authentication {

auth\_type pass

auth\_pass 1111 //主辅服务器密码必须一致

}

virtual\_ipaddress { //配置VIP（实验需要修改）

192.168.4.15

}

}

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 rr //设置LVS调度算法为RR

lb\_kind DR //设置LVS的模式为DR（实验需要修改）

#persistence\_timeout 50 //（实验需要注释）

#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（实验需要修改）

weight 1 //设置权重为1

TCP\_CHECK { //对后台real\_server做健康检查（实验需要修改）

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（实验需要修改）

weight 2 //设置权重为1

TCP\_CHECK { //对后台real\_server做健康检查（实验需要修改）

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@proxy1 ~]# systemctl start keepalived

[root@proxy1 ~]# ipvsadm -Ln #查看LVS规则

[root@proxy1 ~]# ip a s #查看VIP配置

[root@proxy1 ~]# iptables -F

[2]LVS2调度器设调度设置keeplined

[root@proxy2 ~]# 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 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

virtual\_router\_id 51 //主辅VRID号必须一致

priority 50 //服务器优先级（实验需要修改）

advert\_int 1

authentication {

auth\_type pass

auth\_pass 1111 //主辅服务器密码必须一致

}

virtual\_ipaddress { //设置VIP（实验需要修改）

192.168.4.15

}

}

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

delay\_loop 6

lb\_algo rr //设置LVS调度算法为RR

lb\_kind DR //设置LVS的模式为DR（实验需要修改）

# persistence\_timeout 50 //（实验需要注释）

#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服务器的真实IP（实验需要修改）

weight 1 //设置权重为1

TCP\_CHECK { //对后台real\_server做健康检查（实验需要修改）

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（实验需要修改）

weight 2 //设置权重为1

TCP\_CHECK { //对后台real\_server做健康检查（实验需要修改）

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

[root@proxy2 ~]# iptables -F

步骤五:客户端测试

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。