LVS+Keepalived高可用负载平衡集群

1、DR模式

环境：centos7.4虚拟机4台，关闭防火墙、SELinux，清空iptables规则，搭建好yum

规划： node01 192.168.1.1

Node02 192.168.1.2

Node03 192.168.1.3

Node04 192.168.1.4

VIP 192.168.1.100

[root@node03 ~]# cd /etc/sysconfig/network-scripts/

[root@node03 network-scripts]# cp ifcfg-lo ifcfg-lo:0

[root@node03 network-scripts]# vim ifcfg-lo:0

[root@node03 network-scripts]# cat ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.1.100

NETMASK=255.255.255.255

NETWORK=192.168.1.100

# If you're having problems with gated making 127.0.0.0/8 a martian,

# you can change this to something else (255.255.255.255, for example)

BROADCAST=192.168.1.100

ONBOOT=yes

NAME=lo:0

[root@node03 network-scripts]# cd

[root@node03 ~]# vim /etc/sysctl.conf

[root@node03 ~]# sysctl -p

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

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

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

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

[root@node03 ~]# tail -4 /etc/sysctl.conf

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

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

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

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

[root@node03 ~]# systemctl restart network

[root@node03 ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.1.3 netmask 255.255.255.0 broadcast 192.168.1.255

inet6 fe80::5054:ff:fe4e:6abf prefixlen 64 scopeid 0x20<link>

ether 52:54:00:4e:6a:bf txqueuelen 1000 (Ethernet)

RX packets 1446 bytes 118379 (115.6 KiB)

RX errors 0 dropped 106 overruns 0 frame 0

TX packets 726 bytes 89445 (87.3 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1 (Local Loopback)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo:0: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 192.168.1.100 netmask 255.255.255.255

loop txqueuelen 1 (Local Loopback)

[root@node03 ~]# yum -y install httpd

[root@node03 ~]# echo "I'm NODE03" >> /var/www/html/index.html

[root@node03 ~]# systemctl start httpd

[root@node03 ~]# curl localhost

I'm NODE03

[root@node03 ~]#

[root@node04 ~]# vim /etc/sysctl.conf

[root@node04 ~]# tail -4 /etc/sysctl.conf

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

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

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

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

[root@node04 ~]# sysctl -p

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

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

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

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

[root@node04 ~]# cd /etc/sysconfig/network-scripts/

[root@node04 network-scripts]# cp ifcfg-lo ifcfg-lo:0

[root@node04 network-scripts]# vim ifcfg-lo:0

[root@node04 network-scripts]# cat ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.1.100

NETMASK=255.255.255.255

NETWORK=192.168.1.100

# If you're having problems with gated making 127.0.0.0/8 a martian,

# you can change this to something else (255.255.255.255, for example)

BROADCAST=192.168.1.100

ONBOOT=yes

NAME=lo:0

[root@node04 network-scripts]# systemctl restart network

[root@node04 network-scripts]# cd

[root@node04 ~]# ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.1.4 netmask 255.255.255.0 broadcast 192.168.1.255

inet6 fe80::5054:ff:fe53:be67 prefixlen 64 scopeid 0x20<link>

ether 52:54:00:53:be:67 txqueuelen 1000 (Ethernet)

RX packets 1174 bytes 94540 (92.3 KiB)

RX errors 0 dropped 218 overruns 0 frame 0

TX packets 548 bytes 70078 (68.4 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1 (Local Loopback)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo:0: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 192.168.1.100 netmask 255.255.255.255

loop txqueuelen 1 (Local Loopback)

[root@node04 ~]# yum -y install httpd

[root@node04 ~]# echo "I'm NODE04" >> /var/www/html/index.html

[root@node04 ~]# systemctl restart httpd

[root@node04 ~]# curl localhost

I'm NODE04

[root@node04 ~]#

#配置调度器

[root@node01 ~]# yum -y install keepalived.x86\_64 ipvsadm.x86\_64

[root@node01 ~]# vim /etc/keepalived/keepalived.conf

[root@node01 ~]# cat /etc/keepalived/keepalived.conf

global\_defs {

router\_id LVS

}

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.100/24 brd 192.168.1.255 dev eth0 label eth0:1

}

}

virtual\_server 192.168.1.100 80 {

delay\_loop 6

lb\_algo wrr

lb\_kind DR

protocol TCP

real\_server 192.168.1.3 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.1.4 80 {

weight 2

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@node01 ~]# systemctl restart keepalived.service

[root@node01 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.1.100:80 rr

-> 192.168.1.3:80 Route 1 0 0

-> 192.168.1.4:80 Route 2 0 0

[root@node01 ~]# ip a s | grep 192.168.1.100

inet 192.168.1.100/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node02 ~]# yum -y install keepalived.x86\_64 ipvsadm.x86\_64

[root@node02 ~]# mv keepalived.conf /etc/keepalived/keepalived.conf

mv：是否覆盖"/etc/keepalived/keepalived.conf"？ y

[root@node02 ~]#

[root@node02 ~]# vim /etc/keepalived/keepalived.conf

[root@node02 ~]# cat /etc/keepalived/keepalived.conf

global\_defs {

router\_id LVS

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.100/24 brd 192.168.1.255 dev eth0 label eth0:1

}

}

virtual\_server 192.168.1.100 80 {

delay\_loop 6

lb\_algo wrr

lb\_kind DR

protocol TCP

real\_server 192.168.1.3 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.1.4 80 {

weight 2

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@node02 ~]# systemctl restart keepalived.service

[root@node02 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.1.100:80 wrr

-> 192.168.1.3:80 Route 1 0 0

-> 192.168.1.4:80 Route 2 0 0

[root@node02 ~]# ip a s | grep 192.168.1.100

[root@node02 ~]#

##测试负载平衡

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

[root@hostos ~]# curl 192.168.1.100

I'm NODE03

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

[root@hostos ~]#

#测试高可用

[root@node01 ~]# systemctl stop keepalived.service

[root@node01 ~]# ip a s | grep 192.168.1.100

[root@node01 ~]#

[root@node02 ~]# ip a s | grep 192.168.1.100

inet 192.168.1.100/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node02 ~]#

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

[root@hostos ~]# curl 192.168.1.100

I'm NODE03

[root@hostos ~]# curl 192.168.1.100

I'm NODE04

#测试修复占先权

[root@node01 ~]# systemctl start keepalived.service

[root@node01 ~]# ip a s | grep 192.168.1.100

inet 192.168.1.100/24 brd 192.168.1.255 scope global secondary eth0:1

[root@node01 ~]#

[root@node02 ~]# ip a s | grep 192.168.1.100

[root@node02 ~]#

#测试完成

2、NAT模式

#核心，vip和vgw同时漂移

环境：centos7.4虚拟机4台，关闭防火墙、SELinux，搭建好yum源，清空iptables规则

规划： node01 eth0 192.168.4.1 eth1 201.0.1.1

Node02 eth0 192.168.4.2 eth1 201.0.1.2

Node03 eth0 192.168.4.3

Node04 eth0 192.168.4.4.

Vip eth1:1 201.0.1.10

Vgw eth0:1 192.168.4.10

#部署后台

[root@web1 ~]# yum -y install httpd &> /dev/null

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

[root@web1 ~]# systemctl start httpd

[root@web1 ~]#

[root@web2 ~]# yum -y install httpd &> /dev/null

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

[root@web2 ~]# systemctl start httpd

[root@web2 ~]#

[root@hostos ~]# curl 192.168.4.3

web1

[root@hostos ~]# curl 192.168.4.4

web2

[root@hostos ~]#

#部署调度器

[root@proxy1 ~]# yum -y install keepalived.x86\_64 ipvsadm.x86\_64

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

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

global\_defs {

router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 101

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.4.10/24 brd 192.168.4.255 dev eth0 label eth0:1

}

}

vrrp\_instance VI\_2 {

state MASTER

interface eth1

virtual\_router\_id 102

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 2222

}

virtual\_ipaddress {

201.0.1.10/24 brd 201.0.1.255 dev eth1 label eth1:1

}

}

virtual\_server 201.0.1.10 80 {

delay\_loop 6

lb\_algo wrr

lb\_kind NAT

protocol TCP

real\_server 192.168.4.3 80 {

weight 1

TCP\_CHECK {

connect\_port 80

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.4.4 80 {

weight 2

TCP\_CHECK {

connect\_port 80

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@proxy1 ~]# systemctl restart keepalived.service

[root@proxy1 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 201.0.1.10:80 wrr

-> 192.168.4.3:80 Masq 1 0 2

-> 192.168.4.4:80 Masq 2 0 7

[root@proxy1 ~]#

[root@proxy1 ~]# cat /proc/sys/net/ipv4/ip\_forward

0

[root@proxy1 ~]# echo 1 > /proc/sys/net/ipv4/ip\_forward

[root@proxy1 ~]# cat /proc/sys/net/ipv4/ip\_forward

1

[root@proxy1 ~]#

[root@proxy1 ~]# echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

[root@proxy1 ~]# sysctl -p

net.ipv4.ip\_forward = 1

[root@proxy1 ~]# ip a s | grep '\.10'

inet 192.168.4.10/24 brd 192.168.4.255 scope global secondary eth0:1

inet 201.0.1.10/24 brd 201.0.1.255 scope global secondary eth1:1

[root@proxy2 ~]# yum -y install keepalived.x86\_64 ipvsadm.x86\_64

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

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

global\_defs {

router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth0

virtual\_router\_id 101

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.4.10/24 brd 192.168.4.255 dev eth0 label eth0:1

}

}

vrrp\_instance VI\_2 {

state BACKUP

interface eth1

virtual\_router\_id 102

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 2222

}

virtual\_ipaddress {

201.0.1.10/24 brd 201.0.1.255 dev eth1 label eth1:1

}

}

virtual\_server 201.0.1.10 80 {

delay\_loop 6

lb\_algo wrr

lb\_kind NAT

protocol TCP

real\_server 192.168.4.3 80 {

weight 1

TCP\_CHECK {

connect\_port 80

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.4.4 80 {

weight 2

TCP\_CHECK {

connect\_port 80

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@proxy2 ~]# systemctl restart keepalived.service

[root@proxy2 ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 201.0.1.10:80 wrr

-> 192.168.4.3:80 Masq 1 0 0

-> 192.168.4.4:80 Masq 2 0 0

[root@proxy2 ~]# echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

[root@proxy2 ~]# cat /proc/sys/net/ipv4/ip\_forward

0

[root@proxy2 ~]# sysctl -p

net.ipv4.ip\_forward = 1

[root@proxy2 ~]# cat /proc/sys/net/ipv4/ip\_forward

1

[root@proxy2 ~]# ip a s | grep '\.10'

[root@proxy2 ~]#

#配置后台网关

[root@web1 ~]# route add default gw 192.168.4.10

[root@web1 ~]# route -n | grep 4.10

0.0.0.0 192.168.4.10 0.0.0.0 UG 0 0 0 eth0

[root@web1 ~]#

[root@web2 ~]# route add default gw 192.168.4.10

[root@web2 ~]# route -n | grep 4.10

0.0.0.0 192.168.4.10 0.0.0.0 UG 0 0 0 eth0

[root@web2 ~]#

#测试负载平衡

[root@hostos ~]# curl 201.0.1.10

web2

[root@hostos ~]# curl 201.0.1.10

web2

[root@hostos ~]# curl 201.0.1.10

web1

[root@hostos ~]# curl 201.0.1.10

web2

[root@hostos ~]#

#测试高可用

[root@proxy1 ~]# systemctl stop keepalived.service

[root@proxy1 ~]# ip a s | grep '\.10'

[root@proxy1 ~]#

[root@proxy2 ~]# ip a s | grep '\.10'

inet 192.168.4.10/24 brd 192.168.4.255 scope global secondary eth0:1

inet 201.0.1.10/24 brd 201.0.1.255 scope global secondary eth1:1

[root@proxy2 ~]#

[root@hostos ~]# curl 201.0.1.10

web1

[root@hostos ~]# curl 201.0.1.10

web2

[root@hostos ~]# curl 201.0.1.10

web2

[root@hostos ~]# curl 201.0.1.10

web1

[root@hostos ~]#

#测试占先权

[root@proxy1 ~]# systemctl start keepalived.service

[root@proxy1 ~]# ip a s | grep '\.10'

inet 192.168.4.10/24 brd 192.168.4.255 scope global secondary eth0:1

inet 201.0.1.10/24 brd 201.0.1.255 scope global secondary eth1:1

[root@proxy2 ~]# ip a s | grep '\.10'

[root@proxy2 ~]#