LVS Linux Virtual Server

默认写入内核程序默认开启

LVS 原理与 Nginx 完全不同

LVS 为调度功能 及 不修改数据包的IP层 [ SIP(源ip) DIP(目标ip) ] 而是修改数据包包头的MAC地址来达到转发功能 (与交换机原理相同)

Nginx 为代理功能 及 解封数据包后 重新打包封装 发给真实服务器

集群分类

HPC 高性能计算集群

LB 负载均衡

HA 高可用

LVS工作模式

NAT模式 小规模 调度服务器作为一个软路由 负载存在上线

DR模式 大规模 服务器反馈给客户端不通过调度器 通过其他手段实现

TUN模式 不常用

\*\* DR与NAT不同点在于 数据包返回的方式不同 不通过调度服务器

负载均衡常用调度算法

目前有10种

常用

--轮询(Round Robin)

--加权轮询(Weighted Round Robin)

--最少连接(Least Connections)

--加权最少连接(Weighted Least Connections)

最少连接 优于 轮询

\*\* 轮询必定为 1人 1次 若存在客户端持续连接 则会导致 服务器负载不均衡情况

###################################################################################

LVS 也是通过ipvsadm软件实现设置

虚拟集群服务器:

ipvsadm -[A|E|D|C] -[t|u] 192.168.4.5:80(调度器ip及端口) -[s] [rr|wrr|lc|wlc|sh|dh]

-A 添加虚拟集群服务器

-E 编辑虚拟集群服务器

-D 删除虚拟集群服务器(单一)

-C 清空所有虚拟集群服务器

-t TCP协议

-u UDP协议

-s 算法

-rr 轮询 wrr 加权轮询 lc 最少连接次数 wlc 加权最少连接数 sh dh 与ip\_hash作用相同

虚拟集群服务器中真实服务器:

ipvsadm -[a|e|d|] -[t|u] 192.168.4.5:80(虚拟集群服务器ip及端口) -r 192.168.4.10 (真实服务器ip) -[g|m|i] (默认为g) -w (权重)

-a 添加真实服务器

-e 编辑真实服务器

-d 删除真实服务器

-g DR模式

-m NAT模式

-i 隧道模式

ipvsadm -Ln 查看规则列表

####################################################################################

部署LVS-NAT集群

\*\* 由于NAT模式需要将真实服务器地址返回的数据包再次通过调度服务器发送 并且 内部通讯与 客户端与调度通讯 跨网段

\*\* 所以优先要给真是服务器配置网关 及调度ip作为网关 (ip route 查看)

1)部署LVS-NAT模式调度器

确认调度器的路由转发功能

# echo 1 > /proc/sys/net/ipv4/ip\_forward //开启路由转发功能

2)创建集群服务器

# yum -y install ipvsadm

# ipvsadm -A -t 192.168.4.5:80 -s wrr

3)添加真实服务器

# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.100 -w 1 -m

# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -w 1 -m

4）查看规则列表，并保存规则

# ipvsadm -Ln

# ipvsadm-save -n > /etc/sysconfig/ipvsadm //存入配置文档 开启服务自动加载. 若不存入 重启及消失

#########################################################################################

部署LVS-DR集群

配置实验网络环境

1）设置Proxy代理服务器的VIP和DIP

注意：为了防止冲突，VIP必须要配置在网卡的虚拟接口！！！

# cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0:0 //虚拟网卡文档

TYPE=Ethernet

DEFROUTE=yes

NAME=eth0:0

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.4.15

PREFIX=24

# systemctl restart network //若重启后未生效 则因为网卡是有2个服务共同管理 可关闭 NetworkManager 后再次重启

2）设置Web1 web2服务器网络参数 (假装自己为调度通过vip发送)

cp /etc/sysconfig/network-scripts/ifcfg-lo /etc/sysconfig/network-scripts/ifcfg-lo:0 //虚拟网卡文档

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255 //必须为4个255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

3)防止地址冲突的问题

这里因为web1 web2也配置与代理一样的VIP地址，默认肯定会出现地址冲突；

sysctl.conf文件写入这下面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，这样防止地址冲突的问题。

# vim /etc/sysctl.conf

#手动写入如下4行内容

net.ipv4.conf.all.arp\_ignore = 1 //设置所有网卡(包括lo)该回答的回答 不该回答不回答

net.ipv4.conf.lo.arp\_ignore = 1 //单独设置lo 该回答回答 不该回答不回答

net.ipv4.conf.lo.arp\_announce = 2 //设置所有网卡(包括lo)什么都不申明

net.ipv4.conf.all.arp\_announce = 2 //单独设置lo 什么都不申明

#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

# sysctl -p //刷新配置

4)重启网络服务，关闭防火墙与SELinux

# systemctl restart network

proxy调度器安装软件并部署LVS-DR模式调度器

创建新的集群服务器规则

# ipvsadm -A -t 192.168.4.15:80 -s wrr

添加真实服务器(-g参数设置LVS工作模式为DR模式，-w设置权重)

# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.100 -g -w 1

# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.200 -g -w 1

查看规则列表，并保存规则

# ipvsadm -Ln

# ipvsadm-save -n > /etc/sysconfig/ipvsadm //存入配置文档 开启服务自动加载. 若不存入 重启及消失

##########################################################################################################################

Keepalived

Keepalived运行原理

\* 检测每个服务器节点状态

\* 具有安全检测 服务器节点出现异常或故障 keepalived将故障节点从集训中剔除

\* 故障节点恢复后 再将其加入到集群中

\* 所有工作自动完成 无需人工干预

##########################################################################

Keepalived高可用服务器

\*原理与VRRP一致 一台作为主服务器 一台作为备用服务器

\*主 备关系由优先级大小决定(priority) 若一致则通过ip大小比较 ip大优先

安装Keepalived软件

# yum install -y keepalived

修改web1服务器Keepalived配置文件

# 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 \*\*\*与VRID效果一致 若存在同一id则会影响//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

}

}

修改web2服务器Keepalived配置文件

# 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

}

}

启动服务

# systemctl start keepalived

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！

# iptables -F //必须要 并且不是马上生效可能存在1s 2s 延迟

# setenforce 0

登录两台Web服务器查看VIP信息

# ip addr show eth0 与 ip a s eth0d

#########################################################################################

Keepalived+LVS服务器

\*\*本次实验目的为设置一台主调度服务器 一台备用调度 达到高可用效果

\*\*本次实验lvs模式为DR模式 需要在真实服务器web1 web2 上设置回环地址与vip一致 及lo:0 (与vip一致)

1）LVS1调度器设置Keepalived，并启动服务

# yum install -y keepalived

# 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 \*\*\*与VRID效果一致 若存在同一id则会影响//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）

192.168.4.15

}

}

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

}

}

}

# systemctl start keepalived

# ipvsadm -Ln #查看LVS规则

# ip a s #查看VIP配置

# iptables -F

##########################################################################################

配置HAProxy负载平衡集群

功能 lvs<HAProxy<Nginx

性能 lvs>HAProxy>Nginx

\*\* 实用性最高

# yum -y install haproxy

# vim /etc/haproxy/haproxy.cfg

global

log 127.0.0.1 local2 ##[err warning info debug]

chroot /usr/local/haproxy

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

defaults

mode http ##默认的模式mode { tcp|http|health }

option dontlognull ##不记录健康检查的日志信息

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 3000 ##最大连接数

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 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 #进入管理解面查看状态信息

stats auth admin:admin #统计页面用户名和密码设置

\*\* 以下需要手写

listen websrv-rewrite 0.0.0.0:80 // 与listen XX \*:80 效果相同

balance roundrobin // 计算方式必须全写 lestconn

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

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

\*\* check 为安全检查参数

\*\* inter 安全检查间隔 默认毫秒为单位

\*\* rise 开启测试2次成功 则加入集群

\*\* fall 失败测试5次 则从集群删除