一　LVS

不带健康检查，需自己写脚本检测

１，基础

１）集群目的

提高性能

降低成本

提高可扩展性

增强可靠性

1. 集群分类

－－高性能计算集群HPC

　　通过以集群开发的并行应用程序，解决复杂的科学问题

－－负载均衡（LB）集群

　　客户端负载在计算集群中尽可能平均分摊

－－高可用（HA）集群

避免单点故障，当一个系统发生故障时，可以快速迁移

1. LVS集群组成

－－前端：负载均衡层

由一台或多台负载调度器构成

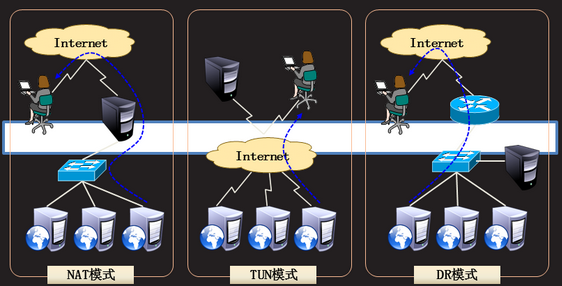
－－中间：服务器群组层

由一组实际运行应用服务的服务器组成

－－底端：数据库共享存储层

提供共享存储空间的存储区域

1. LVS工作模式



－－VS/NAT

　　通过网络地址转换实现的虚拟服务器，大并发访问时，调度器的性能成为瓶劲

－－VS/DR

直接使用路由技术实现虚拟服务器，节点服务器需要配置VIP,注意MAC地址传播

－－VS/TUN

通过隧道方式实现虚拟服务器

1. ipvsadm用法

ipvsadm -Ａ｜-E |-D |-C

2　部署LVS-NAT集群

１）环境

使用LVS实现NAT模式的集群调度服务器，为用户提供Web服务：

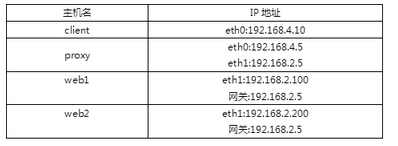
集群对外公网IP地址为192.168.4.5

调度器内网IP地址为192.168.2.5

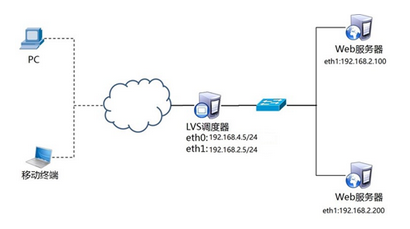
真实Web服务器地址分别为192.168.2.100、192.168.2.200

使用加权轮询调度算法，真实服务器权重分别为1和2

主机环境配置：



拓扑图：



２）配置web基础环境

３）部署LVS-NAT模式调度器

－－确认调度器的路由转发功能(如果已经开启，可以忽略)

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

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

1

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

#修改配置文件，设置永久规则

－－创建集群服务器

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

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:80 -s wrr

－－添加真实服务器

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.100 -w 1 -m

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -w 1 -m

－－查看规则列表，并保存规则

[root@proxy ~]# ipvsadm -Ln

[root@proxy ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm

1. 客户端测试

客户端使用curl命令反复连接http://192.168.4.5，查看访问的页面是否会轮询到不同的后端真实服务器。

[root@client ~]# curl http://192.168.4.5

<h2>ip is 192.168.2.100</h2>

[root@client ~]# curl http://192.168.4.5

<h2>ip is 192.168.2.200</h2>

[root@proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 wrr

-> 192.168.2.100:80 Masq 1 0 5

-> 192.168.2.200:80 Masq 1 0 4

3　部署LVS-DR集群

同一个网卡可以配置多个ip，如(eth0),(eth0:0),(eth0:1)

Linux:拷贝ifcfg-eth0 -> ifcfg-eth0:0

lo网卡:回环地址

１）需求

使用LVS实现DR模式的集群调度服务器，为用户提供Web服务：

客户端IP地址为192.168.4.10

LVS调度器VIP地址为192.168.4.15

LVS调度器DIP地址设置为192.168.4.5

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，web1的权重为1，web2的权重为2

说明：

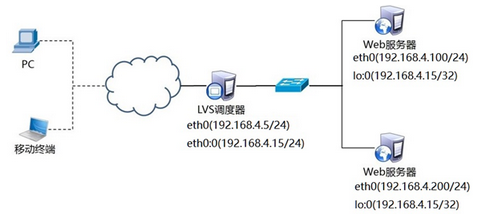
CIP是客户端的IP地址；

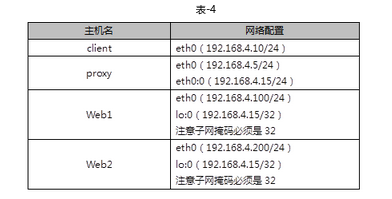
VIP是对客户端提供服务的IP地址；

RIP是后端服务器的真实IP地址；

DIP是调度器与后端服务器通信的IP地址（VIP必须配置在虚拟接口）。

使用4台虚拟机，1台作为客户端、1台作为Director调度器、2台作为Real Server，拓扑结构如图-2所示。实验拓扑结构主机配置细节如表-4所示。





1. 设置Proxy代理服务器的VIP和DIP

－－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！！！

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

[root@proxy ~]# cp ifcfg-eth0{,:0}

[root@proxy ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0:0

TYPE=Ethernet

BOOTPROTO=none

DEFROUTE=yes

NAME=eth0:0

UUID=4f5f8b5d-4b51-4e1c-98c7-40f0eb90f241

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.4.15

PREFIX=24

－－设置Web1服务器网络参数

[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

防止地址冲突的问题：

这里因为web1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。

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

#手动写入如下4行内容

net.ipv4.conf.all.arp\_ignore = 1　　　　＃all所有网卡忽略

net.ipv4.conf.lo.arp\_ignore = 1　　　　　＃回环网卡忽略

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

＃仅仅影响lo回环地址

[root@web1 ~]# sysctl -p

重启网络服务，设置防火墙与SELinux

[root@web1 ~]# systemctl restart network

[root@web1 ~]# ifconfig

[root@web1 ~]# systemctl stop firewalld

[root@web1 ~]# setenforce 0

－－设置Web2服务器网络参数

[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

防止地址冲突的问题：

这里因为web1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。

[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

[root@web2 ~]# sysctl -p

重启网络服务，设置防火墙与SELinux

[root@web2 ~]# systemctl restart network

[root@web2 ~]# ifconfig

[root@web2 ~]# systemctl stop firewalld

[root@web2 ~]# setenforce 0

1. 配置后端Web服务器

－－自定义Web页面

[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

－－启动Web服务器软件

[root@web1 ~]# systemctl restart httpd

[root@web2 ~]# systemctl restart httpd

３）proxy调度器安装软件并部署LVS-DR模式调度器

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

[root@proxy ~]# ipvsadm -C #清空所有规则

[root@proxy ~]# ipvsadm -A -t 192.168.4.15:80 -s wrr

[root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.100 -g -w 1

root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.200 -g -w 1

[root@proxy ~]# ipvsadm -Ln　　　＃查看规则列表

TCP 192.168.4.15:80 wrr

-> 192.168.4.100:80 Route 1 0 0

-> 192.168.4.200:80 Route 2 0 0 [root@proxy ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm　　＃保存规则

1. 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。

扩展知识：默认LVS不带健康检查功能，需要自己手动编写动态检测脚本，实现该功能：(参考脚本如下，仅供参考)

[root@proxy ~]# vim check.sh

#!/bin/bash

VIP=192.168.4.15:80

RIP1=192.168.4.100

RIP2=192.168.4.200

while :

do

for IP in $RIP1 $RIP2

do

curl -s http://$IP &>/dev/vnull

web\_stat=$?

ipvsadm -Ln | grep -q $IP

web\_in\_lvs=$?

if [ $web\_stat -ne 0 -a $web\_in\_lvs -eq 0 ];then

ipvsadm -d -t $VIP -r $IP

elif [ $web\_stat -eq 0 -a $web\_in\_lvs -ne 0 ];then

ipvsadm -a -t $VIP -r $IP

fi

done

sleep 1

done