1. [案例1：ipvsadm命令用法](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY01/CASE/01/index.html" \l "case1)
2. [案例2：部署LVS-NAT集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY01/CASE/01/index.html#case2)
3. [案例3：部署LVS-DR集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY01/CASE/01/index.html#case3)

**1 案例1：ipvsadm命令用法**

**1.1 问题**

准备一台Linux服务器，安装ipvsadm软件包，练习使用ipvsadm命令，实现如下功能：

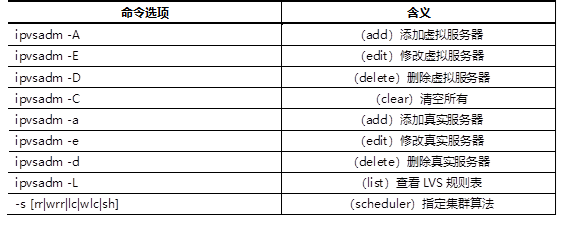
* 使用命令添加基于TCP一些的集群服务
* 在集群中添加若干台后端真实服务器
* 实现同一客户端访问，调度器分配固定服务器
* 会使用ipvsadm实现规则的增、删、改
* 保存ipvsadm规则

**1.2 方案**

安装ipvsadm软件包，关于ipvsadm的用法可以参考man ipvsadm资料。

常用ipvsadm命令语法格式如表-1及表-2所示。

表－1 ipvsadm命令选项



表－2 ipvsadm语法案例



**1.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：使用命令增、删、改LVS集群规则**

1）创建LVS虚拟集群服务器（算法为加权轮询：wrr）

1. **[**root@proxy **~]**# yum **-**y install ipvsadm
2. **[**root@proxy **~]**# ipvsadm **-**A **-**t **192.168.4.5:80** **-**s wrr
3. **[**root@proxy **~]**# ipvsadm **-**Ln
4. IP Virtual Server version **1.2.1** **(**size**=4096)**
5. Prot LocalAddress**:**Port Scheduler Flags
6. **->** RemoteAddress**:**Port Forward Weight ActiveConn InActConn
7. TCP **192.168.4.5:80** wrr

2）为集群添加若干real server

1. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.100**
2. **[**root@proxy **~]**# ipvsadm **-**Ln
3. IP Virtual Server version **1.2.1** **(**size**=4096)**
4. Prot LocalAddress**:**Port Scheduler Flags
5. **->** RemoteAddress**:**Port Forward Weight ActiveConn InActConn
6. TCP **192.168.4.5:80** wrr
7. **->** **192.168.2.100:80** router **1** **0** **0**
8. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.200** **-**m **-**w **2**
9. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.201** **-**m **-**w **3**
10. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.202** **-**m **-**w **4**

3）修改集群服务器设置(修改调度器算法，将加权轮询修改为轮询)

1. **[**root@proxy **~]**# ipvsadm **-**E **-**t **192.168.4.5:80** **-**s rr
2. **[**root@proxy **~]**# ipvsadm **-**Ln
3. IP Virtual Server version **1.2.1** **(**size**=4096)**
4. Prot LocalAddress**:**Port Scheduler Flags
5. **->** RemoteAddress**:**Port Forward Weight ActiveConn InActConn
6. TCP **192.168.4.5:80** rr
7. **->** **192.168.2.100:80** router **1** **0** **0**
8. **->** **192.168.2.200:80** masq **2** **0** **0**
9. **->** **192.168.2.201:80** masq **2** **0** **0**
10. **->** **192.168.2.202:80** masq **1** **0** **0**

4）修改read server（使用-g选项，将模式改为DR模式）

1. **[**root@proxy **~]**# ipvsadm **-**e **-**t **192.168.4.5:80** **-**r **192.168.2.202** **-**g

5）查看LVS状态

1. **[**root@proxy **~]**# ipvsadm **-**Ln

6）创建另一个集群（算法为最少连接算法；使用-m选项，设置工作模式为NAT模式）

1. **[**root@proxy **~]**# ipvsadm **-**A **-**t **192.168.4.5:3306** **-**s lc
2. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:3306** **-**r **192.168.2.100** **-**m
3. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:3306** **-**r **192.168.2.200** **-**m

7）永久保存所有规则（非必须的操作）

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

注意：永久规则需要确保ipvsadm服务为开机启动服务才可以。

（systemctl enable ipvsadm）。

8）清空所有规则

1. **[**root@proxy **~]**# ipvsadm **-**C

**2 案例2：部署LVS-NAT集群**

**2.1 问题**

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，真实服务器权重任意

**2.2 方案**

实验拓扑结构主机配置细节如表-3所示，注意下面的网卡名称仅为参考，不能照抄。

表-3



使用4台虚拟机，1台作为Director调度器、2台作为Real Server、1台客户端，拓扑结构如图-1所示，注意：web1和web2必须配置网关地址。

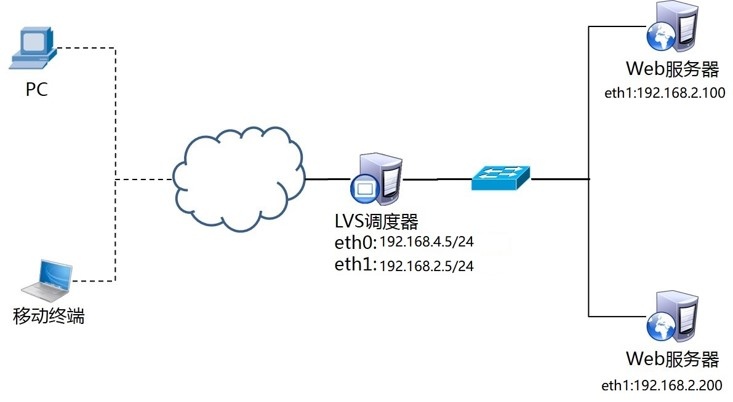


图-1

**2.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：配置基础环境**

1）设置Web服务器

1. **[**root@web1 **~]**# yum **-**y install httpd        #安装软件
2. **[**root@web1 **~]**# echo "192.168.2.100" **>** /var/www**/**html**/**index**.**html    #创建网页文件
3. **[**root@web1 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted            #设置防火墙
4. **[**root@web1 **~]**# setenforce **0**
5. **[**root@web1 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
6. **[**root@web2 **~]**# yum **-**y install httpd        #安装软件
7. **[**root@web2 **~]**# echo "192.168.2.200" **>** /var/www**/**html**/**index**.**html    #创建网页文件
8. **[**root@web2 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted            #设置防火墙
9. **[**root@web2 **~]**# setenforce **0**
10. **[**root@web2 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config

2）启动Web服务器软件

1. **[**root@web1 **~]**# systemctl restart httpd
2. **[**root@web2 **~]**# systemctl restart httpd

如何验证？

完成后可以使用proxy主机测试下是否可以访问web1和web2

1. **[**root@proxy **~]**# curl http**:**//192.168.2.100
2. **[**root@proxy **~]**# curl http**:**//192.168.2.200

3）配置网关，将web1和web2的网关设置为192.168.2.5（不能照抄网卡名称）

1. **[**root@web1 **~]**# nmcli connection modify ens33 **\**
2. ipv4**.**method manual ipv4**.**gateway **192.168.2.5**
3. #备注**:**网卡名称不能照抄**,**需要自己查看下**2.100**的网卡名称
4. **[**root@web1 **~]**# nmcli connection up ens33
5. **[**root@web1 **~]**# ip route show         #查看默认网关
6. **default** via **192.168.2.5** dev ens33 #提示：这里**default**后面的IP就是默认网关
7. #英语词汇：**default**（默认，预设值）
8. … …
9. **[**root@web2 **~]**# nmcli connection modify ens33 **\**
10. ipv4**.**method manual ipv4**.**gateway **192.168.2.5**
11. #备注**:**网卡名称不能照抄**,**需要自己查看下**2.200**的网卡名称
12. **[**root@web2 **~]**# nmcli connection up ens33
13. **[**root@web2 **~]**# ip route show        #查看默认网关，**default**后面的IP就是默认网关

**为什么需要配置网关？**

实验拓扑如图-2所示。

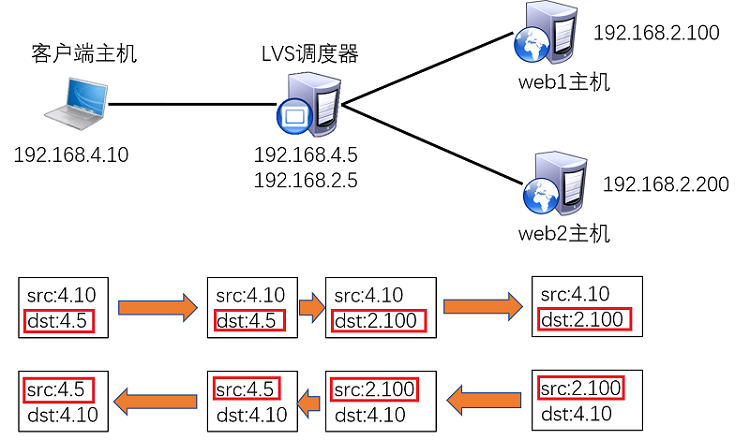


图-2

为了方便下面所有的IP都采用简写,如4.10代表192.168.4.10，2.100代表192.168.2.100。

英语词汇：source（src）代表源地址，destination（dest或dst）代表目标地址。

LVS采用的是路由器的NAT通讯原理！通讯流程如下：

1.客户端发送请求数据包(src:4.10,dst:4.5)

2.数据包被发送给LVS调度器，调度器做NAT地址转换（外网转内网，内网转外网）

a)数据包被修改为src:4.10,dst:2.100(dst也有可能被修改为2.200，随机的）

b)LVS调度器把数据包转发给后端真正的web服务器（2.100）

3.web1收到数据包开始回应数据（rsc:2.100，dst:4.10）

备注：谁访问就给谁回复数据，因为src是4.10，所以应该给4.10回应数据！

但是，自己是2.100，对方是4.10，跨网段默认无法通讯，如何解决？？？

Web1和web2都需要设置默认网关（也就是192.168.2.5）

4）web1想发送数据给4.10但是又无法与其通讯，所以数据包被交给默认网关

5）LVS调度器（软路由）收到后端web发送过来的数据后，再次做NAT地址转换

a)数据包被修改为src:4.5,dst:4.10

b)LVS调度器把数据包转发给客户端主机

6）客户端接收网页数据内容

注意：客户端访问的是4.5，最后是4.5给客户端回复的网页数据！！！！

**步骤二：部署LVS-NAT模式调度器**

1)确认调度器的路由转发功能(如果已经开启，可以忽略)

1. **[**root@proxy **~]**# echo **1** **>** /proc/sys**/**net**/**ipv4**/**ip\_forward #开启路由转发，临时有效
2. **[**root@proxy **~]**# cat **/**proc**/**sys**/**net**/**ipv4**/**ip\_forward #查看效果
3. **1**
4. **[**root@proxy **~]**# echo "net.ipv4.ip\_forward = 1" **>>** /etc/sysctl**.**conf
5. #修改配置文件，设置永久规则，英语词汇：forward（转寄，转发，发送，向前）

2）创建集群服务器

1. **[**root@proxy **~]**# yum **-**y install ipvsadm
2. **[**root@proxy **~]**# ipvsadm **-**A **-**t **192.168.4.5:80** **-**s wrr
3. # **-**A**(**add**)**是创建添加虚拟服务器集群
4. # **-**t**(**tcp**)**后面指定集群VIP的地址和端口，协议是tcp协议
5. # **-**s后面指定调度算法，如rr（轮询）、wrr（加权轮询）、lc（最少连接）、wlc（加权最少连接）等等

3）添加真实服务器

1. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.100** **-**w **1** **-**m
2. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.5:80** **-**r **192.168.2.200** **-**w **1** **-**m
3. #**-**a**(**add**)**往虚拟服务器集群中添加后端真实服务器IP**,**指定往**-**t **192.168.4.5:80**这个集群中添加
4. #**-**r**(**real**)**后面跟后端真实服务器的IP和端口，这里不写端口默认是**80**端口
5. #**-**w**(**weight**)**指定服务器的权重，权重越大被访问的次数越多，英语词汇：weight（重量，分量）
6. #**-**m指定集群工作模式为NAT模式，如果是**-**g则代表使用DR模式，**-**i代表TUN模式

4）查看规则列表（L是list查看，n是number数字格式显示）

1. **[**root@proxy **~]**# ipvsadm **-**Ln

5)设置防火墙，SELinux

1. **[**root@proxy **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
2. **[**root@proxy **~]**# setenforce **0**
3. **[**root@proxy **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config

**步骤三：客户端测试**

客户端client主机使用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。

**3 案例3：部署LVS-DR集群**

**3.1 问题**

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，权重可以任意

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）。

**3.2 方案**

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-3所示。实验拓扑结构主机配置细节如表-4所示。

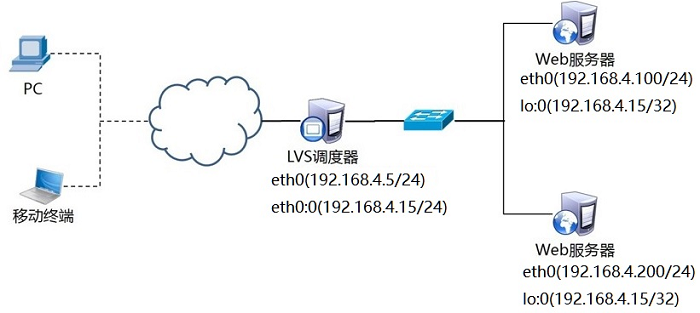
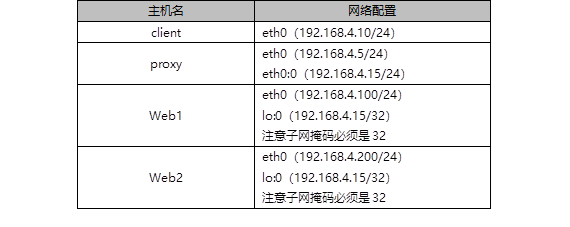


图-3

表-4



为什么本实验中web1和web2要采用4网段IP？如图-4所示。

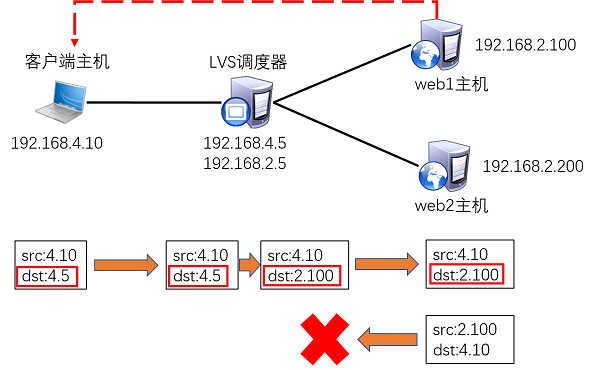


图-4

LVS NAT实验请求数据包从LVS调度器进，web的相应数据包也从LVS调度器出，那么LVS调度器就需要承载所有数据的压力，会成为整个集群的瓶颈！！

本实验LVS DR模式的核心需求是希望web1和web2可以不走调度器返回数据！

但是如如-3所示，如果web1和web2采用2.100和2.200这样2网段的IP，又不希望给4.10回复数据走LVS调度器（也就是不给web1和web2配置默认网关为2.5），最后是无法跨网段通讯的！！！

怎么办？核心需求是希望web1和web2可以直接返回数据给客户端！！！

想让web1和web2可以直接返回数据给客户端，可以给web1和web2配置4网段IP，如图-5所示。

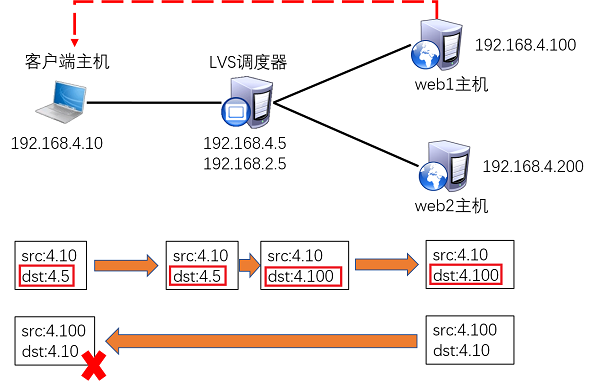


图-5

这样就可以了吗？答案是否定的！！！网络中的基本原则是A访问B，必须是B返回数据给A，现在4.10访问4.5，最终4.100给4.10返回网页数据，所有数据包都会被丢弃！！！

那怎么办呢？地址欺骗！

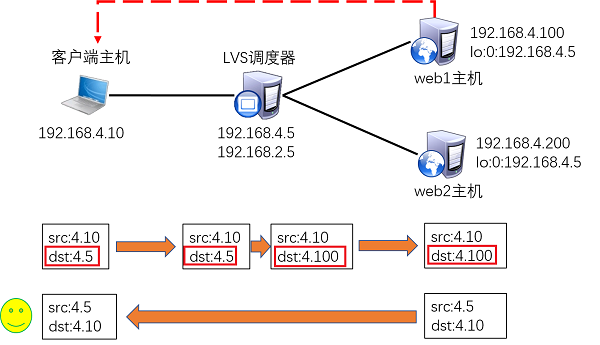


图-6

如图-6所示，我们给web1和web2再额外添加一个伪装的IP地址，这个IP地址因为是用来做地址欺骗用的，假的就是假的，不能暴露（必须配置在lo本地回环网卡上面）。

lo网卡上面默认配置的IP是127.0.0.1。

如果你家里有非法的1000W，你会天天出去跟别人说你有1000W吗?

**3.3 步骤**

实现此案例需要按照如下步骤进行。

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（本案例为192.168.4.15）；

VIP必须配置在虚拟接口（目的是防止地址冲突）；

RIP是后端服务器的真实IP地址（本案例为192.168.4.100和192.168.4.200）；

DIP是调度器与后端服务器通信的IP地址（本案例为192.168.4.5）。

**步骤一：配置实验网络环境**

**1）设置Proxy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，网卡名称不能照抄！！！**

1. **[**root@proxy **~]**# cd **/**etc**/**sysconfig**/**network**-**scripts**/**
2. **[**root@proxy **~]**# cp ifcfg**-**eth0 ifcfg**-**eth0**:0**
3. **[**root@proxy **~]**# vim ifcfg**-**eth0**:0**
4. TYPE**=**Ethernet
5. #网卡类型为：以太网卡
6. BOOTPROTO**=**none
7. #none手动配置IP，或者dhcp自动配置IP
8. NAME**=**eth0**:0**
9. #网卡名称
10. DEVICE**=**eth0**:0**
11. #设备名称
12. ONBOOT**=**yes
13. #开机时是否自动激活该网卡
14. IPADDR**=192.168.4.15**
15. #IP地址
16. PREFIX**=24**
17. #子网掩码
18. **[**root@proxy **~]**# systemctl restart network        #重启网络服务
19. **[**root@proxy **~]**# ip a s #会看到一个网卡下面有两个IP地址

常见问题：RHEL7和Centos7系统中有两个管理网络的服务，有可能冲突？

解决方法：关闭NetworkManager服务后重启network即可。

2）设置Web1服务器网络参数（不能照抄网卡名称）

1. **[**root@web1 **~]**# nmcli connection modify eth0 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.4.100/24** connection**.**autoconnect yes
3. **[**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。

1. **[**root@web1 **~]**# cd **/**etc**/**sysconfig**/**network**-**scripts**/**
2. **[**root@web1 **~]**# cp ifcfg**-**lo ifcfg**-**lo**:0**
3. **[**root@web1 **~]**# vim ifcfg**-**lo**:0**
4. DEVICE**=**lo**:0**
5. #设备名称
6. IPADDR**=192.168.4.15**
7. #IP地址
8. NETMASK**=255.255.255.255**
9. #子网掩码
10. NETWORK**=192.168.4.15**
11. #网络地址
12. BROADCAST**=192.168.4.15**
13. #广播地址
14. ONBOOT**=**yes
15. #开机是否激活本网卡
16. NAME**=**lo**:0**
17. #网卡名称

防止地址冲突的问题：

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；

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。

1. **[**root@web1 **~]**# vim **/**etc**/**sysctl**.**conf
2. #文件末尾手动写入如下**4**行内容**,**英语词汇：ignore（忽略、忽视），announce（宣告、广播通知）
3. net**.**ipv4**.**conf**.**all**.**arp\_ignore **=** **1**
4. net**.**ipv4**.**conf**.**lo**.**arp\_ignore **=** **1**
5. net**.**ipv4**.**conf**.**lo**.**arp\_announce **=** **2**
6. net**.**ipv4**.**conf**.**all**.**arp\_announce **=** **2**
7. #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（防止进站冲突）
8. #本机不要向外宣告自己的lo回环地址是**192.168.4.15**（防止出站冲突）
9. **[**root@web1 **~]**# sysctl **-**p

重启网络服务

1. **[**root@web1 **~]**# systemctl restart network        #重启网络服务
2. **[**root@web1 **~]**# ip a s #会看到一个网卡下面有两个IP地址

常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。（非必须的操作）

1. **[**root@web1 **~]**# systemctl stop NetworkManager
2. **[**root@web1 **~]**# systemctl restart network

3）设置Web2服务器网络参数（不能照抄网卡名称）

1. **[**root@web2 **~]**# nmcli connection modify eth0 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.4.200/24** connection**.**autoconnect yes
3. **[**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。

1. **[**root@web2 **~]**# cd **/**etc**/**sysconfig**/**network**-**scripts**/**
2. **[**root@web2 **~]**# cp ifcfg**-**lo ifcfg**-**lo**:0**
3. **[**root@web2 **~]**# vim ifcfg**-**lo**:0**
4. DEVICE**=**lo**:0**
5. #设备名称
6. IPADDR**=192.168.4.15**
7. #IP地址
8. NETMASK**=255.255.255.255**
9. #子网掩码
10. NETWORK**=192.168.4.15**
11. #网络地址
12. BROADCAST**=192.168.4.15**
13. #广播地址
14. ONBOOT**=**yes
15. #开机是否激活该网卡
16. NAME**=**lo**:0**
17. #网卡名称

防止地址冲突的问题：

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；

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。

1. **[**root@web2 **~]**# vim **/**etc**/**sysctl**.**conf
2. #手动写入如下**4**行内容，英语词汇：ignore（忽略、忽视），announce（宣告、广播通知）
3. net**.**ipv4**.**conf**.**all**.**arp\_ignore **=** **1**
4. net**.**ipv4**.**conf**.**lo**.**arp\_ignore **=** **1**
5. net**.**ipv4**.**conf**.**lo**.**arp\_announce **=** **2**
6. net**.**ipv4**.**conf**.**all**.**arp\_announce **=** **2**
7. #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（防止进站冲突）
8. #本机不要向外宣告自己的lo回环地址是**192.168.4.15**（防止出站冲突）
9. **[**root@web2 **~]**# sysctl **-**p

重启网络服务

1. **[**root@web2 **~]**# systemctl restart network        #重启网络服务
2. **[**root@web2 **~]**# ip a s #会看到一个网卡下面有两个IP地址

常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。（非必须的操作）

1. **[**root@web1 **~]**# systemctl stop NetworkManager
2. **[**root@web1 **~]**# systemctl restart network

**步骤二：proxy调度器安装软件并部署LVS-DR模式调度器**

1）安装软件（如果已经安装，此步骤可以忽略）

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

2）清理之前实验的规则，创建新的集群服务器规则

1. **[**root@proxy **~]**# ipvsadm **-**C #清空所有规则
2. **[**root@proxy **~]**# ipvsadm **-**A **-**t **192.168.4.15:80** **-**s wrr
3. ## **-**A**(**add**)**是创建添加虚拟服务器集群
4. # **-**t**(**tcp**)**后面指定集群VIP的地址和端口，协议是tcp协议
5. # **-**s后面指定调度算法，如rr（轮询）、wrr（加权轮询）、lc（最少连接）、wlc（加权最少连接）等等

3）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)

1. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.15:80** **-**r **192.168.4.100** **-**g **-**w **1**
2. **[**root@proxy **~]**# ipvsadm **-**a **-**t **192.168.4.15:80** **-**r **192.168.4.200** **-**g **-**w **1**
3. #**-**a**(**add**)**往虚拟服务器集群中添加后端真实服务器IP**,**指定往**-**t **192.168.4.15:80**这个集群中添加
4. #**-**r**(**real**)**后面跟后端真实服务器的IP和端口，这里不写端口默认是**80**端口
5. #**-**w**(**weight**)**指定服务器的权重，权重越大被访问的次数越多，英语词汇：weight（重量，分量）
6. #**-**m指定集群工作模式为NAT模式，如果是**-**g则代表使用DR模式，**-**i代表TUN模式

4）查看规则列表（L代表list查看规则，n代表number数字格式显示）

1. **[**root@proxy **~]**# ipvsadm **-**Ln
2. TCP **192.168.4.15:80** wrr
3. **->** **192.168.4.100:80** Route **1** **0** **0**
4. **->** **192.168.4.200:80** Route **1** **0** **0**

**步骤三：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。

注意：本实验不可以在proxy主机（LVS调度器）使用curl访问网页验证！！！

为什么？请思考图-7示意图。

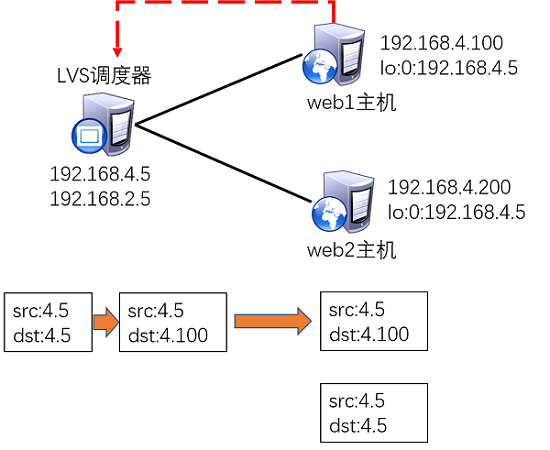


图-7

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)

1. **[**root@proxy **~]**# vim check**.**sh
2. #**!**/bin/bash
3. VIP**=192.168.4.15:80**
4. RIP1**=192.168.4.100**
5. RIP2**=192.168.4.200**
6. **while** **:**
7. **do**
8. **for** IP **in** $RIP1 $RIP2
9. **do**
10. curl **-**s http**:**//$IP &>/dev/null
11. **if** **[** $**?** **-**eq **0** **];**then
12. ipvsadm **-**Ln **|**grep **-**q $IP **||** ipvsadm **-**a **-**t $VIP **-**r $IP
13. **else**
14. ipvsadm **-**Ln **|**grep **-**q $IP **&&** ipvsadm **-**d **-**t $VIP **-**r $IP
15. fi
16. done
17. sleep **1**
18. done