第三十二章：Lvs负载均衡群集

**一、概述：**

**二、LVS详解：**

**三、案例一：搭建LVS 的NAT模式负载均衡集群；**

**四、案例二：搭建LVS 的DR模式负载均衡集群；**

**五、总结：Lvs负载均衡群集与nginx负载均衡群集对比（自主学习）；**

**一、概述：**

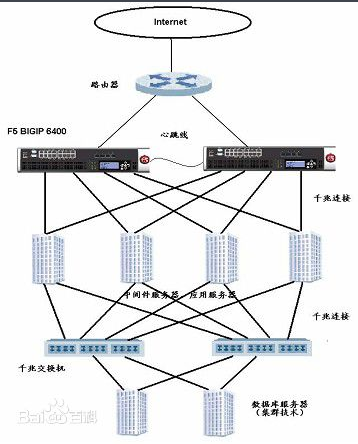
**概述：**Linux Virtual Server的简写，意即Linux虚拟服务器，是一个虚拟的服务器集群系统。我国章文嵩博士在1998年五月创建，默认编译为ip\_vs内核模块，而在linux kernel的2.6版本之后kernel是直接支持ipvs的，优势：LVS承受负载能力高、稳定、占用服务器资源小，缺点：适配场景、配置较麻烦、不支持节点的健康检查机制；

**官网：**http://www.linuxvirtualserver.org/zh/lvs1.html

**集群技术概述：**至少包含两个节点服务器，对外表示为一个整体，只提供一个访问入口；

负载均衡（load balance cluster）：将整个平台的负载均衡到多台单位；

高可用（high availablity cluster）：使整个应用平台拥有容错能力；



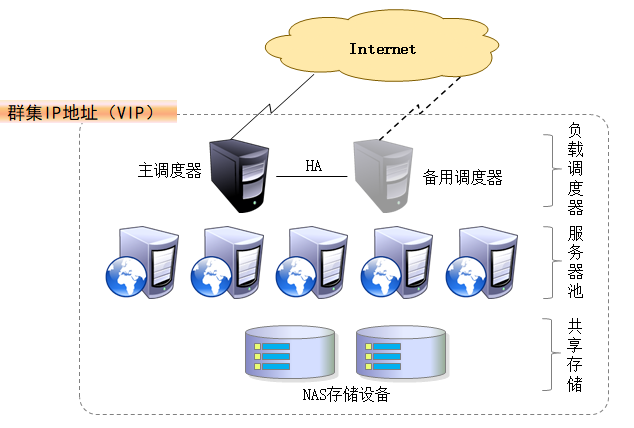
可伸缩性（Scalability）：当服务的负载增长时，系统能被扩展来满足需求，且不降低服务质量；

高可用性（Availability）：尽管部分硬件和软件会发生故障，整个系统的服务必须是每天24小时每星期7天可用的；

可管理性（Manageability）：整个系统可能在物理上很大，但应该容易管理；

价格有效性（Cost-effectiveness）：整个系统实现是经济的、易支付的；

**集群的分层结构：**



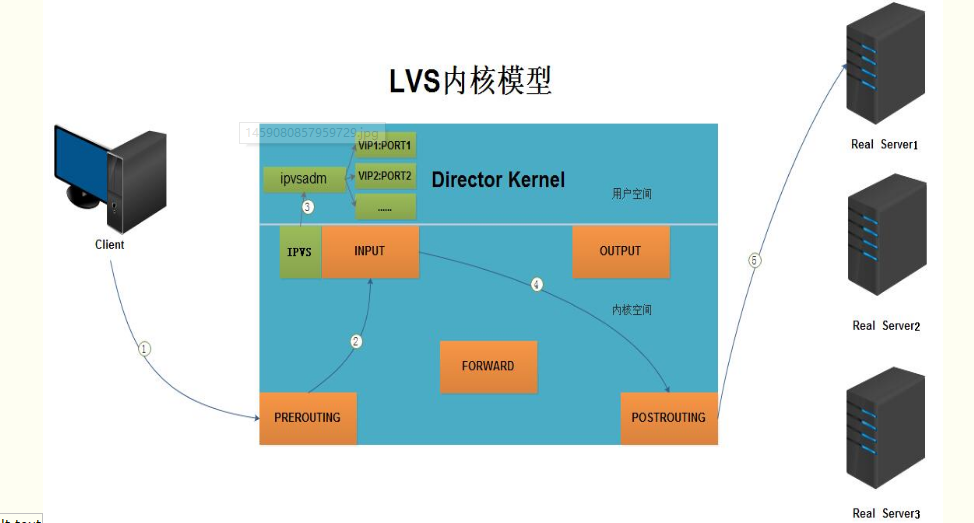
第一层：负载调度器（load balancer或director），访问群集的唯一入口，对外使用所有服务器共有的VIP（virtual ip）地址，也称为群集IP地址。

第二层：节点层（real server pool），服务器池群集所提供的应用服务由服务器池承担，其中的每个节点具有独立的RIP（real IP真实地址），只处理调度服务器分发过来的客户机请求。当某个节点暂时失效时，负载调度器的容错机制会将其隔离，等待错误排除以后再重新纳入服务器池。

第三层：共享存储层或数据库层，共享存储为服务器池中的所有节点提供稳定，一致的文件存取服务，确保整个群集的统一性。

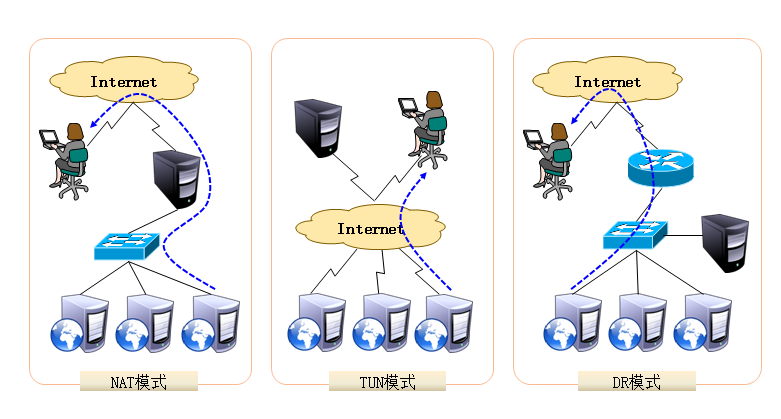
**二、LVS详解：**

**LVS内核模型：**



**调度条件：**基于IP（效率最高）、基于端口、基于内容

**LVS的三种工作模式：**



1.地址转换：NAT模式，负载调度器作为所有服务器的网关，服务器节点使用私有IP地址（隐蔽性高），与负载调度器位于同一个物理网络，安全性高于其他两种方式。但是负载调度器是整个架构的瓶颈；

b.IP隧道：TUN模式，采用开放是的网络结构，负载调度器仅作为客户机的访问入口，个节点通过各自的Internet连接回应客户机，而不再经过负载调度器。服务器节点分散在互联网的不同位置，具有独立的公网IP地址，通过专用的IP隧道与负载调度器互通。安全性低，性能高；

c.直接路由：DR模式，采用半开放式的网络结构，与TUN模式结构类似，但各节点并不是分散在各地，而是与调度服务器位于同一个物理网络。负载调度器与各节点服务器通过本地网络连接，不需要建立专用的IP隧道。性能最高，安全性较高；

**Lvs负载调度算法：**

**静态：**

1.轮询rr：将请求按顺序依次分配给各个web节点；

2.加权轮询wrr：根据web服务器的处理能力，动态调整权重，将请求按照各节点的负载情况进行顺序分配；

3.目标地址散列调度dh：根据请求的目标IP地址，作为散列键（HashKey）从静态分配的散列表找出对应的服务器，若该服务器是可用的且未超载，将请求发送到该服务器，否则返回空。

4.源地址 hash：源地址散列”调度算法根据请求的源IP地址，作为散列键（HashKey）从静态分配的散列表找出对应的服务器，若该服务器是可用的且未超载，将请求发送到该服务器，否则返回空｡

**动态：**

1.最小连接Lc：根据web服务器的连接数量，将请求分配给web节点服务器；

2.加权最小连接WLC：根据web服务器的处理能力，动态调整权重，将请求按照各节点的连接数量进行分配，默认；

3.最短延迟调度SED：在WLC基础上改进，Overhead = （ACTIVE+1）\*256/加权，不再考虑非活动状态，把当前处于活动状态的数目+1来实现，数目最小的，接受下次请求，+1的目的是为了考虑加权的时候，非活动连接过多缺陷：当权限过大的时候，会倒置空闲服务器一直处于无连接状态。

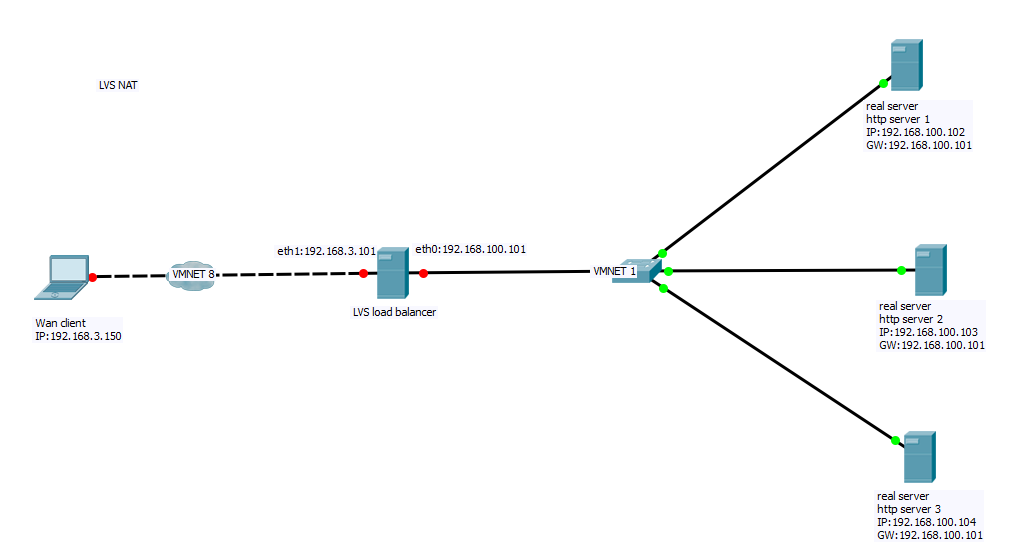
4.永不排队/最少队列调度NQ：无需队列。如果有台 realserver的连接数＝0就直接分配过去，不需要再进行sed运算，保证不会有一个主机很空间。在SED基础上无论+几，第二次一定给下一个，保证不会有一个主机不会很空闲着，不考虑非活动连接，才用NQ，SED要考虑活动状态连接，对于DNS的UDP不需要考虑非活动连接，而httpd的处于保持状态的服务就需要考虑非活动连接给服务器的压力。

5.基于局部性的最少链接LBLC：基于局部性的最少链接”调度算法是针对目标IP地址的负载均衡，目前主要用于Cache集群系统｡该算法根据请求的目标IP地址找出该目标IP地址最近使用的服务器，若该服务器是可用的且没有超载，将请求发送到该服务器;若服务器不存在，或者该服务器超载且有服务器处于一半的工作负载，则用“最少链接”的原则选出一个可用的服务器，将请求发送到该服务器｡

6.带复制的基于局部性最少连接LBLCR：带复制的基于局部性最少链接”调度算法也是针对目标IP地址的负载均衡，目前主要用于Cache集群系统｡它与LBLC算法的不同之处是它要维护从一个目标IP地址到一组服务器的映射，而LBLC算法维护从一个目标IP地址到一台服务器的映射｡该算法根据请求的目标IP地址找出该目标IP地址对应的服务器组，按”最小连接”原则从服务器组中选出一台服务器，若服务器没有超载，将请求发送到该服务器；若服务器超载，则按“最小连接”原则从这个集群中选出一台服务器，将该服务器加入到服务器组中，将请求发送到该服务器｡同时，当该服务器组有一段时间没有被修改，将最忙的服务器从服务器组中删除，以降低复制的程度。

**三、案例一：搭建LVS 的NAT模式负载均衡集群；**

**案例拓扑：**



**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 7.4 1708 64bit | 192.168.100.101  192.168.3.101 | ld.linuxfan.cn | ipvsadm、内核模块ip\_vs |
| Centos 7.4 1708 64bit | 192.168.100.102 | real1.linuxfan.cn | httpd |
| Centos 7.4 1708 64bit | 192.168.100.103 | real2.linuxfan.cn | httpd |
| Centos 7.4 1708 64bit | 192.168.100.104 | real3.linuxfan.cn | httpd |
| win7 | 192.168.3.150 | win7-1 |  |

**案例步骤：**

* 搭建real server 节点池中的三台http服务节点（在此只列出一台配置，其他两台相同）；
* 配置LD负载调度器的环境以及安装软件程序；
* 配置LD负载调度服务器的调度服务；
* 公网客户端测试访问集群；
* **搭建real server 节点池中的三台http服务节点（在此只列出一台配置，其他两台相同）；**

[root@real1 ~]# ip a |grep 192.168.100.102 ##设置ip地址

inet 192.168.100.102/24 brd 192.168.100.255 scope global eth0

[root@real1 ~]# ip r |grep 192.168.100.101 ##设置默认网关地址

default via 192.168.100.101 dev eth0 proto static metric 100

[root@real1 ~]# yum -y install httpd

[root@real1 ~]# cat <<END >/var/www/html/index.html

I am real1.linuxfan.cn

END

[root@real1 ~]# systemctl start httpd

[root@real1 ~]# systemctl enable httpd ##设置开机自动启动

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

* **配置LD负载调度器的环境以及安装软件程序；**

**[root@ld ~]# ip a |grep 101**

**inet 192.168.100.101/24 brd 192.168.100.255 scope global eth0**

**inet 192.168.3.101/24 brd 192.168.3.255 scope global eth1**

**[root@ld ~]# echo "net.ipv4.ip\_forward = 1" >>/etc/sysctl.conf ##开启路由转发功能，因为调度器同时承载着网关的角色**

**[root@ld ~]# sysctl -p**

**net.ipv4.ip\_forward = 1**

**[root@ld ~]# modprobe ip\_vs ##加载内核模块ip\_vs**

**[root@ld ~]# lsmod |grep ip\_vs**

**ip\_vs 141092 0**

**nf\_conntrack 133387 1 ip\_vs**

**libcrc32c 12644 3 xfs,ip\_vs,nf\_conntrack**

* **配置LD负载调度服务器的调度服务；**

[root@ld ~]# yum -y install ipvsadm ##安装ip\_vs模块的规则编写工具

[root@ld ~]# systemctl stop ipvsadm ##清空内部规则

[root@ld ~]# ipvsadm -A -t 192.168.3.101:80 -s rr ##指定集群的VIP地址（Virtual IP），rr指定轮询调度算法

[root@ld ~]# ipvsadm -a -t 192.168.3.101:80 -r 192.168.100.102:80 -m -w 1 ##-m表示NAT模式，-w指定权重值

[root@ld ~]# ipvsadm -a -t 192.168.3.101:80 -r 192.168.100.103:80 -m -w 1

[root@ld ~]# ipvsadm -a -t 192.168.3.101:80 -r 192.168.100.104:80 -m -w 1

**[root@ld ~]# ipvsadm-save ##保存规则**

**-A -t ld.linuxfan.cn:http -s rr**

**-a -t ld.linuxfan.cn:http -r 192.168.100.102:http -m -w 1**

**-a -t ld.linuxfan.cn:http -r 192.168.100.103:http -m -w 1**

**-a -t ld.linuxfan.cn:http -r 192.168.100.104:http -m -w 1**

**[root@ld ~]# ipvsadm -L ##查看规则**

**IP Virtual Server version 1.2.1 (size=4096)**

**Prot LocalAddress:Port Scheduler Flags**

**-> RemoteAddress:Port Forward Weight ActiveConn InActConn**

**TCP ld.linuxfan.cn:http rr**

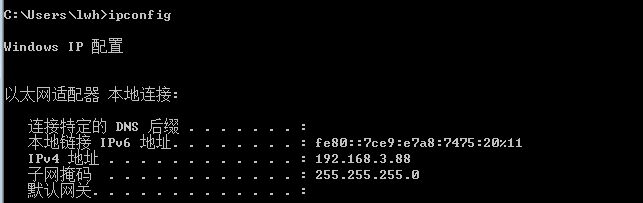
**-> 192.168.100.102:http Masq 1 0 0**

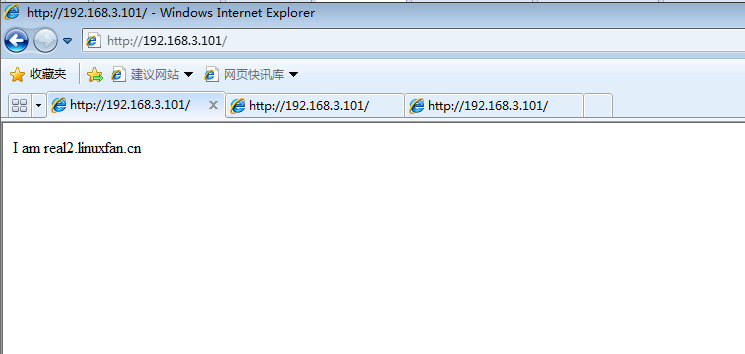
**-> 192.168.100.103:http Masq 1 0 0**

**-> 192.168.100.104:http Masq 1 0 0**

**注解：LVS的负载调度算法中，无论**哪种算法，在一段时间内（360s），都能够实现将来自同一个地址的请求始终发往同一个real server，这样就避免了重复请求的次数，可以使用如下命令更改lvs的默认保持时间：ipvsadm -A -t 192.168.3.101:80 -s rr -p 300

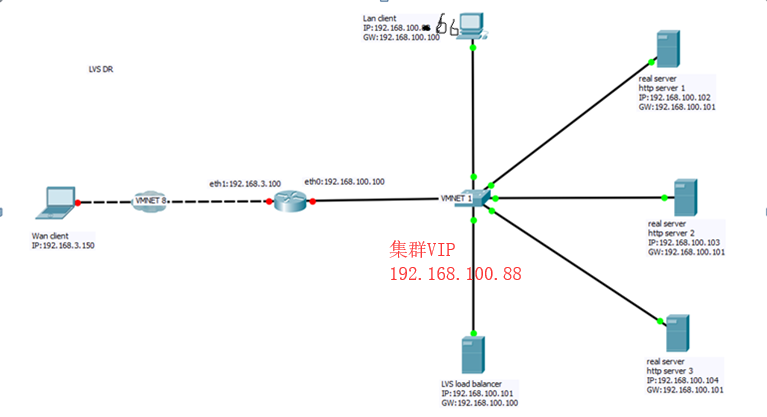
* **公网客户端测试访问集群；**





**四、案例二：搭建LVS 的DR模式负载均衡集群；**

**案例拓扑：**



**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 6.5 64bit | 192.168.100.100  192.168.3.100 | GW | iptables |
| Centos 7.4 1708 64bit | 192.168.100.101 | ld.linuxfan.cn | ipvsadm、内核模块ip\_vs |
| Centos 7.4 1708 64bit | 192.168.100.102 | real1.linuxfan.cn | httpd |
| Centos 7.4 1708 64bit | 192.168.100.103 | real2.linuxfan.cn | httpd |
| Centos 7.4 1708 64bit | 192.168.100.104 | real3.linuxfan.cn | httpd |
| win7-1 | 192.168.100.66 | lan-client | 浏览器 |
| win7-2 | 192.168.3.150 | wan-client | 浏览器 |
| VIP（virtual-ip） | 192.168.100.88 |  |  |

**案例步骤：**

* 搭建real server 节点池中的三台http服务节点（在此只列出一台配置，其他两台相同）；
* 设置负载调度器上的VIP地址；
* 调整负载调度器的响应参数；
* 配置负载调度器负载分配策略；
* 配置real server节点池内节点服务器的网络参数（在此只列出一台配置，其他两台相同）；
* 内网客户端访问测试集群；
* 在GW服务器上设置iptables的DNAT规则；
* 外网客户端访问测试集群；
* 测试LVS是否支持节点的健康检查功能；
* 调整LVS的负载调度算法（自主学习）；
* **搭建real server 节点池中的三台http服务节点（在此只列出一台配置，其他两台相同）；**

[root@real1 ~]# ip a |grep 192.168.100.102 ##设置ip地址

inet 192.168.100.102/24 brd 192.168.100.255 scope global eth0

[root@real1 ~]# ip r |grep 192.168.100.100 ##设置默认网关地址

default via 192.168.100.100 dev eth0 proto static metric 100

[root@real1 ~]# yum -y install httpd

[root@real1 ~]# cat <<END >/var/www/html/index.html

I am real1.linuxfan.cn

END

[root@real1 ~]# systemctl start httpd

[root@real1 ~]# systemctl enable httpd

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

* **设置负载调度器上的VIP地址；**

[root@ld ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

TYPE=Ethernet

BOOTPROTO=static

DEFROUTE=yes

PEERDNS=yes

PEERROUTES=yes

IPV4\_FAILURE\_FATAL=no

NAME=eth0

DEVICE=eth0

ONBOOT=yes

IPADDR=192.168.100.101

PREFIX=24

GATEWAY=192.168.100.100

DNS1=192.168.100.100

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

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

TYPE=Ethernet

BOOTPROTO=static

DEFROUTE=yes

PEERDNS=yes

PEERROUTES=yes

IPV4\_FAILURE\_FATAL=no

NAME=eth0:0

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.100.88

PREFIX=24

GATEWAY=192.168.100.100

DNS1=192.168.100.100

[root@ld ~]# systemctl restart network

[root@ld ~]# ip a |grep 192.168.100

inet 192.168.100.101/24 brd 192.168.100.255 scope global eth0

inet 192.168.100.88/24 brd 192.168.100.255 scope global secondary eth0:0

* **调整负载调度器的响应参数；**

[root@ld ~]# vi /etc/sysctl.conf ##最后添加

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.eth0.send\_redirects = 0

:wq

注解：

禁止所有网卡转发重定向报文

禁止默认转发重定向报文

禁止eth0网卡转发重定向报文

[root@ld ~]# sysctl -p

* **配置负载调度器负载分配策略；**

[root@ld ~]# modprobe ip\_vs

[root@ld ~]# lsmod |grep ip\_vs

ip\_vs\_rr 12600 1

ip\_vs 141092 3 ip\_vs\_rr

nf\_conntrack 133387 1 ip\_vs

libcrc32c 12644 3 xfs,ip\_vs,nf\_conntrack

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

[root@ld ~]# ipvsadm --clear ##清空ipvs的调度规则

[root@ld ~]# systemctl stop ipvsadm

[root@ld ~]# ipvsadm -L

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

[root@ld ~]# ipvsadm -A -t 192.168.100.88:80 -s rr

[root@ld ~]# ipvsadm -a -t 192.168.100.88:80 -r 192.168.100.102:80 -g -w 1 ##表示DR模式

[root@ld ~]# ipvsadm -a -t 192.168.100.88:80 -r 192.168.100.103:80 -g -w 1

[root@ld ~]# ipvsadm -a -t 192.168.100.88:80 -r 192.168.100.104:80 -g -w 1

[root@ld ~]# ipvsadm-save

-A -t ld.linuxfan.cn:http -s rr

-a -t ld.linuxfan.cn:http -r 192.168.100.102:http -g -w 1

-a -t ld.linuxfan.cn:http -r 192.168.100.103:http -g -w 1

-a -t ld.linuxfan.cn:http -r 192.168.100.104:http -g -w 1

[root@ld ~]# ipvsadm -L

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP ld.linuxfan.cn:http rr

-> 192.168.100.102:http Route 1 0 0

-> 192.168.100.103:http Route 1 0 0

-> 192.168.100.104:http Route 1 0 0

* **配置real server节点池内节点服务器的网络参数（在此只列出一台配置，其他两台相同）；**

[root@real1 ~]# cat <<END >/etc/sysconfig/network-scripts/ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.100.88

NETMASK=255.255.255.255

ONBOOT=yes

NAME=lo:0

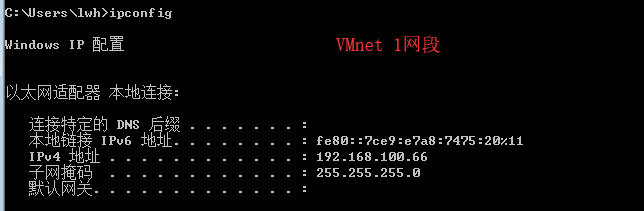
END

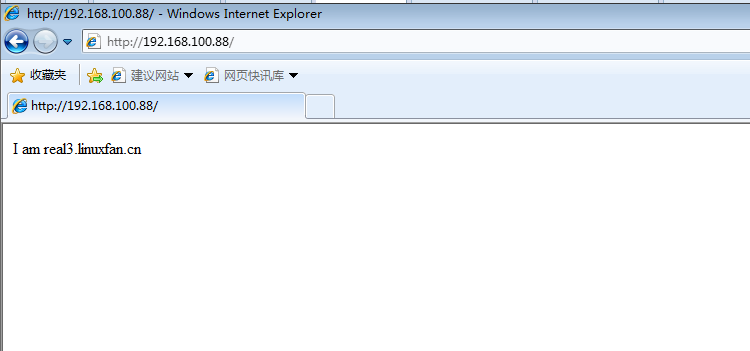
[root@ real1 ~]# systemctl restart network

[root@real1 ~]# ip a |grep 88

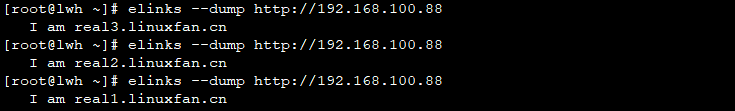
inet 192.168.100.88/32 brd 192.168.100.88 scope global lo:0

* **内网客户端访问测试集群；**









* **在GW服务器上设置iptables的DNAT规则；**

[root@GW ~]# iptables -t nat -A PREROUTING -i eth1 -d 192.168.3.100 -p tcp --dport 80 -j DNAT --to-destination 192.168.100.88

[root@ GW ~]# iptables -t nat -L -n

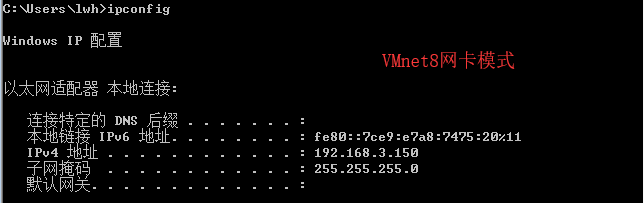
Chain PREROUTING (policy ACCEPT)

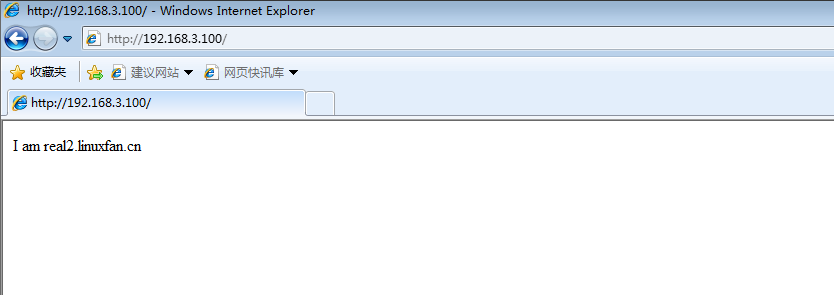
target prot opt source destination

DNAT tcp -- 0.0.0.0/0 192.168.3.100 tcp dpt:80 to:192.168.100.88

...

* **外网客户端访问测试集群；**





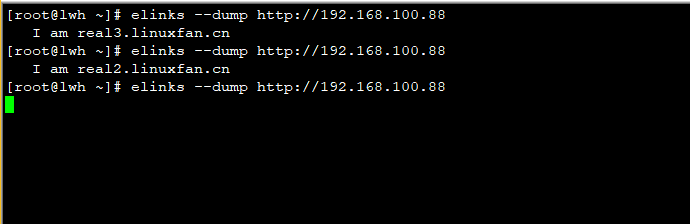
* **测试LVS是否支持节点的健康检查功能；**

[root@real1 ~]# ifconfig lo:0 down ##断开real1的lo：0的网卡，使其脱离集群

[root@real1 ~]# ip a |grep 192.168.100.

inet 192.168.100.102/24 brd 192.168.100.255 scope global eth0

客户端访问集群测试：



* **调整LVS的负载调度算法（自主学习）；**

**五、总结：Lvs负载均衡群集与nginx负载均衡群集对比（自主学习）**

