

老男孩教育LVS四层负载均衡企业级实践

- 2022年LVS四层大并发负载均衡，带你深度掌握LVS集群。
- 该PDF文档会持续更新，有新技术会补充进去。
- 如获取干货资料，可关注公众号，如咨询\技术交流，可加群\微信，老男孩教育陪你进步。



关注公众号 领取更多干货



学习交流，加V联系

Linux技术交流QQ群784850820

附：往期干货文档笔记下载

PS：点击标题链接直接跳转

- [网络安全大厂面试真题.pdf下载](#)
- [公司网站架构图.png下载](#)
- [Linux三剑客-sed详解.pdf下载](#)
- [Git命令手册.pdf下载](#)
- [Linux速查备忘手册+知识点总结](#)

老男孩教育LVS四层负载均衡企业级实践

笔者的话:

- 1.1 LVS介绍
- 1.2 IPVS发展史
- 1.3 IPVS工作层次
- 1.4 LVS技术点小结
- 1.5 LVS的4种工作模式原理
 - 1.5.1 LVS工具术语命名约定
 - 1.5.2 LVS四种负载工作模式名称
 - 1.5.3 NAT模式原理
 2. NAT模式原理图
 - 1.5.4 TUN模式原理
 - 1.5.4 DR模式原理
 - 1.5.4 FNAT模式原理
- 1.6 LVS的不同模式优缺点比较
- 1.7 LVS调度算法
- 1.8 LVS负载均衡实践
 - 1.8.1 环境准备
 - 1.8.2 项目架构图
 - 1.8.3 LVS负载均衡安装
 - 1.8.4 场景1：实现MySQL负载均衡

- 1.配置LVS虚拟IP (VIP)
- 2.手工执行配置添加LVS服务并增加两台RS ipvsadm
- 3.手工在RS端绑定lo网卡及抑制ARP
- 4.手工在RS端抑制ARP响应
- 5.抑制ARP脚本实现
- 6.arp抑制技术参数说明
- 1.8.5 场景2: 实现LVS+keepalived db高可用
- 1.8.6 场景3: 实现Web4层负载及后端7层反代+Web节点
- 1.9 LVS视频教程
- 2.0 总结

笔者的话:

随着时代发展，LVS在中小企业作为单独Web集群负载均衡用的已经极少了，更多被更简单的Nginx替代。在今天（2022年）的运维工作中，LVS更多的是放在Nginx7层负载前面做4层负载，目的是支持大并发，另外，TCP负载依然有用武之地，例如MySQL集群负载、K8S集群负载，因此LVS是运维人员的一个重要的技术栈内容。——老男孩

1.1 LVS介绍

LVS是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，可以在Unix/Linux平台下实现负载均衡集群功能。该项目是中国国内最早出现的自由软件项目之一。官方：<http://www.linuxvirtualserver.org/zh>

1.2 IPVS发展史

LVS的具体实现是IPVS内核模块，因此，可简单将LVS和IPVS作为一个东西称呼。早在Linux2.2内核时，IPVS就已经以内核补丁的形式出现了。从2.4.23版本开始，IPVS软件就是合并到Linux内核的常用版本的内核补丁的集合。从2.4.24以后IPVS已经成为Linux官方标准内核的一部分。

1.3 IPVS工作层次



从上图我们看出，LVS负载均衡调度技术是在Linux内核中实现的，因此，被称之为Linux虚拟服务器（Linux Virtual Server）。使用该软件配置LVS时候，不能直接配置内核中的ipvs，而需要使用ipvs的管理工具ipvsadm进行管理，当然还可通过Keepalived软件直接管理ipvs，ipvs的基础配置工具是ipvsadm。

1.4 LVS技术点小结

- 1、真正实现调度的工具是IPVS，工作在Linux内核层面。
- 2、LVS自带的IPVS命令行管理工具是ipvsadm。
- 3、Keepalived可更好实现管理IPVS（配置文件）及负载均衡器的高可用。
- 4、Red hat工具Piranha WEB管理实现调度的工具IPVS。

1.5 LVS的4种工作模式原理

1.5.1 LVS工具术语命名约定

名称	缩写	说明
虚拟IP地址	VIP	VIP (Virtual Ip Address), 域名解析的IP地址。比如： www.etiantian.org 解析到vip上。
真实IP地址	RIP	RIP(Real Server Ip Address)在集群下面节点上使用的IP地址，物理IP地址。
Director IP地址	DIP	(Director Ip Address)是 Director用于连接内外网络即物理网卡的IP地址。
客户端IP地址	CIP	(Client Ip Address) 是客户端请求集群服务器的IP地址。

1.5.2 LVS四种负载工作模式名称

LVS负载均衡器支持下面四种负载工作模式。

- 网络地址转换模式，即**NAT**模式（Network Address Translation）
- IP隧道模式，即**TUN**模式（IP Tunneling）
- 直接路由模式，即**DR**模式（Direct Routing）

完全网络地址转换模式，即FULLNAT模式（FULL Network Address Translation）

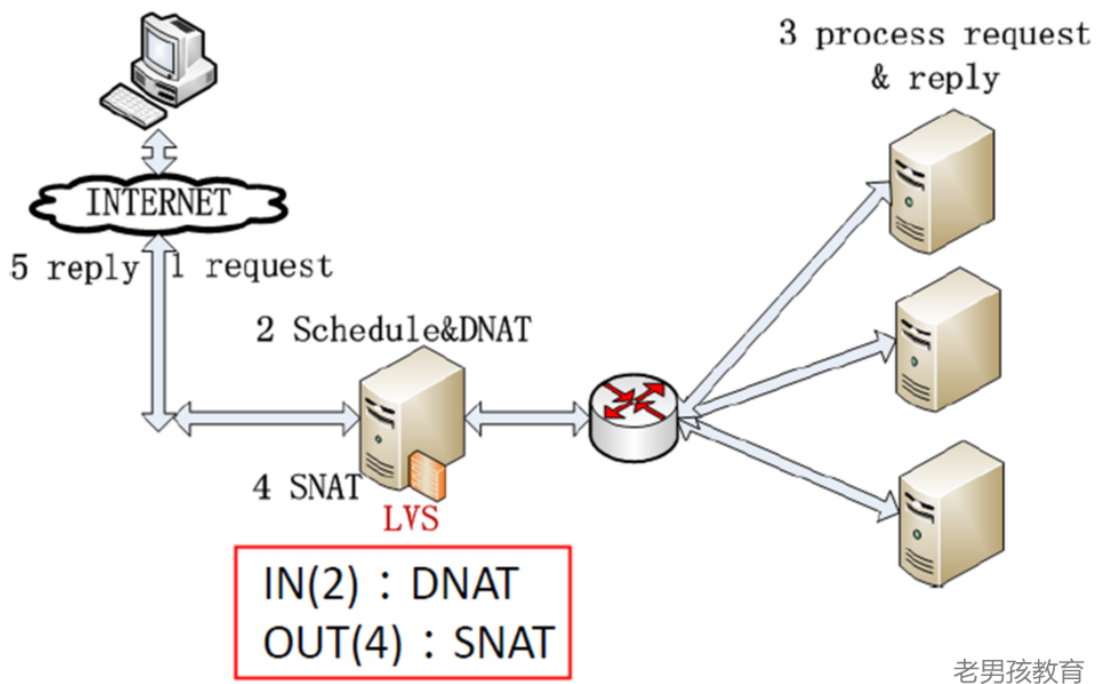
1.5.3 NAT模式原理

1. NAT模式介绍

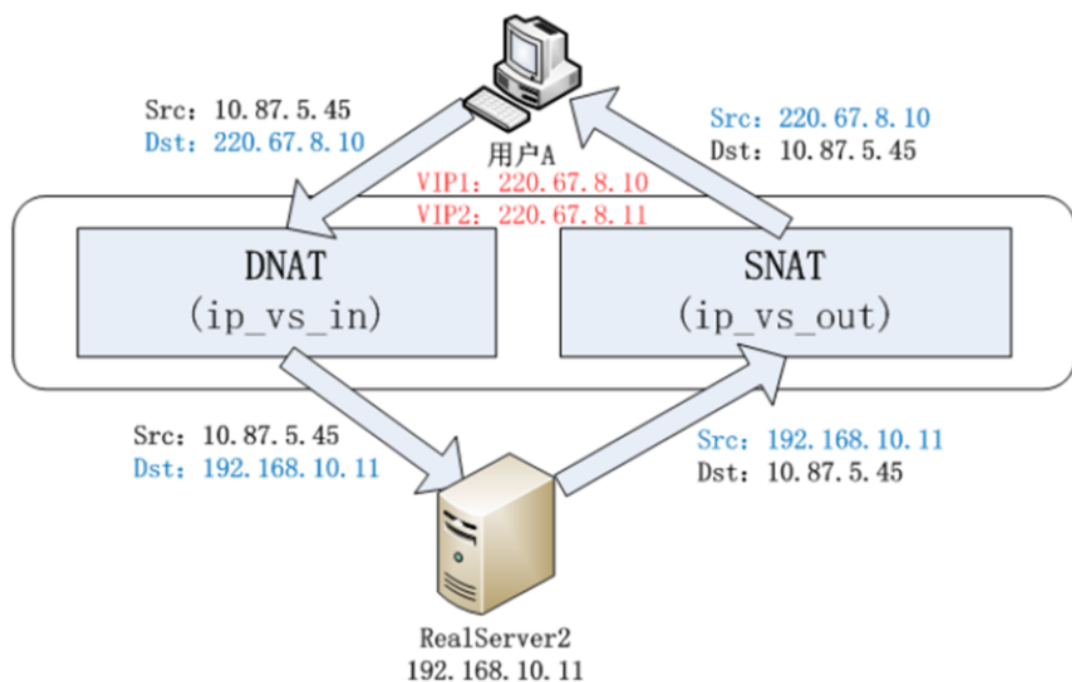
简单说就是生活中收费站，大多数商业化IP负载均衡调度器产品都是使用NAT方法，如F5、Netscaler。NAT模式是通过网络地址转换，由LB重写请求报文目标地址，根据预设调度算法，将请求分派给后端真实服务器；服务器响应并处理，报文返回时必须要通过调度器，经过调度器时报文的源地址被重写，再返回给客户完成整个负载调度过程。

2. NAT模式原理图

• NAT



• NAT实现原理



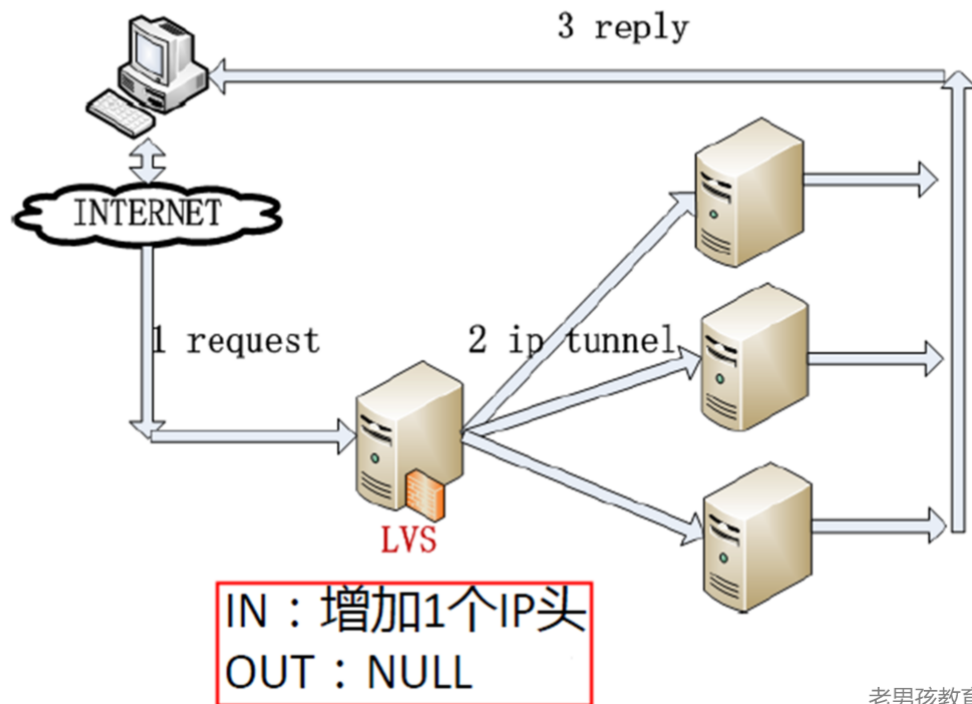
在这里需要指出，根据缺省的TCP/IP协议栈处理，请求报文的目标地址为VIP，响应报文的源地址肯定也为VIP，所以响应报文不需要作任何修改，可以直接返回给客户，客户认为得到正常的服务，而不会知道究竟是哪一台服务器处理的，所以LVS才叫虚拟服务器。

1.5.4 TUN模式原理

1. TUN模式介绍

2. TUN模式原理图

• TUNNEL



老男孩教育

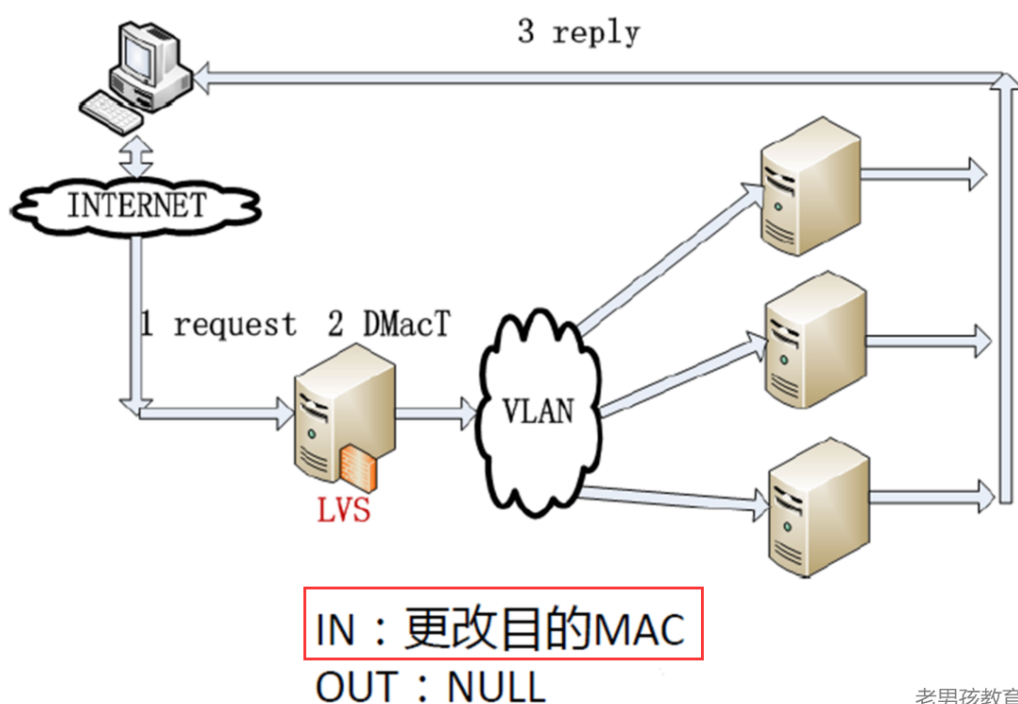
1.5.4 DR模式原理

1. DR模式介绍

DR模式是通过改写请求报文的目标MAC地址，将请求发给真实服务器的，而真实服务器将响应后的处理结果直接返回给客户端用户。同TUN技术，DR技术可极大地提高集群系统伸缩性。但是这种DR模式没有IP隧道的开销，对集群中的真实服务器也没有必须支持IP隧道协议的要求，但是要求调度器LB与真实服务器RS都有一块网卡连在同一物理网段上，即必须在同一个局域网环境。

2. DR模式原理图

• DR



老男孩教育

1.5.4 FNAT模式原理

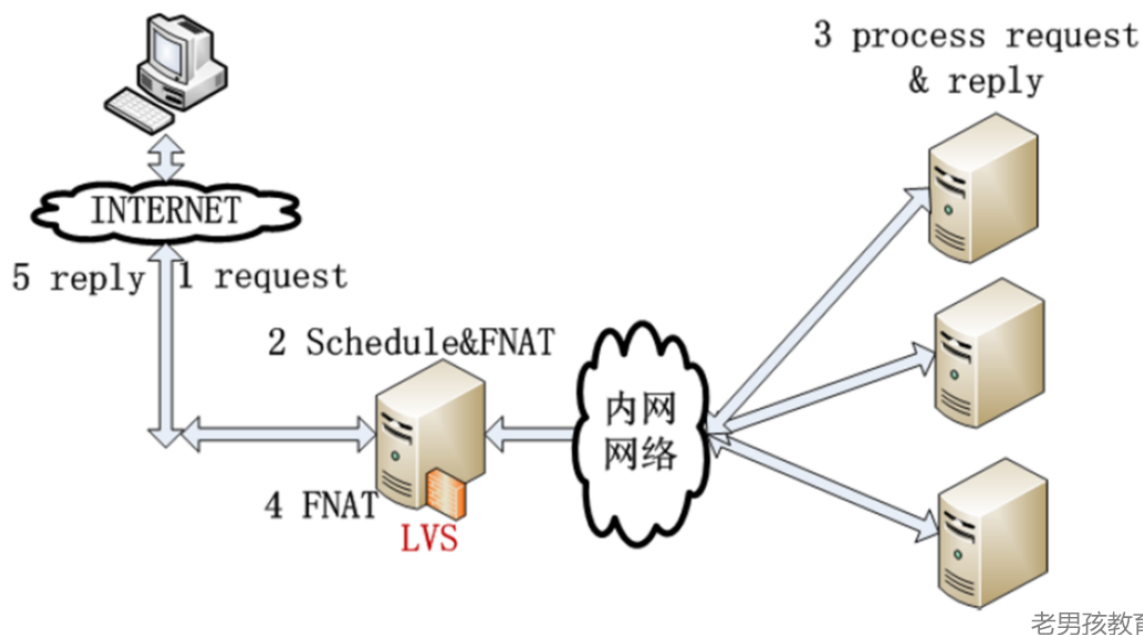
1. FNAT模式介绍

FULLNAT模式是后期淘宝网开源的模式，LVS应用主要采用DR和NAT模式，但这2种模式要求RealServer和LVS在同一个vlan中，导致部署成本过高；TUNNEL模式虽然可以跨vlan，但RealServer上需要部署ipip隧道模块等，网络拓扑上需要连通外网，较复杂，不易运维。

为了解决上述问题，FULLNAT诞生，该模式和NAT模式区别是：数据包进入时，除了做DNAT，还做SNAT（用户ip->内网ip），数据包出时，除了做SNAT，还做DNAT（内网ip->用户ip）从而实现LVS-RealServer间可以跨vlan通讯，RealServer只需要连接到内网。和NAT比，正常转发性能下降<10%，但是LB可以多台同时负载均衡工作；

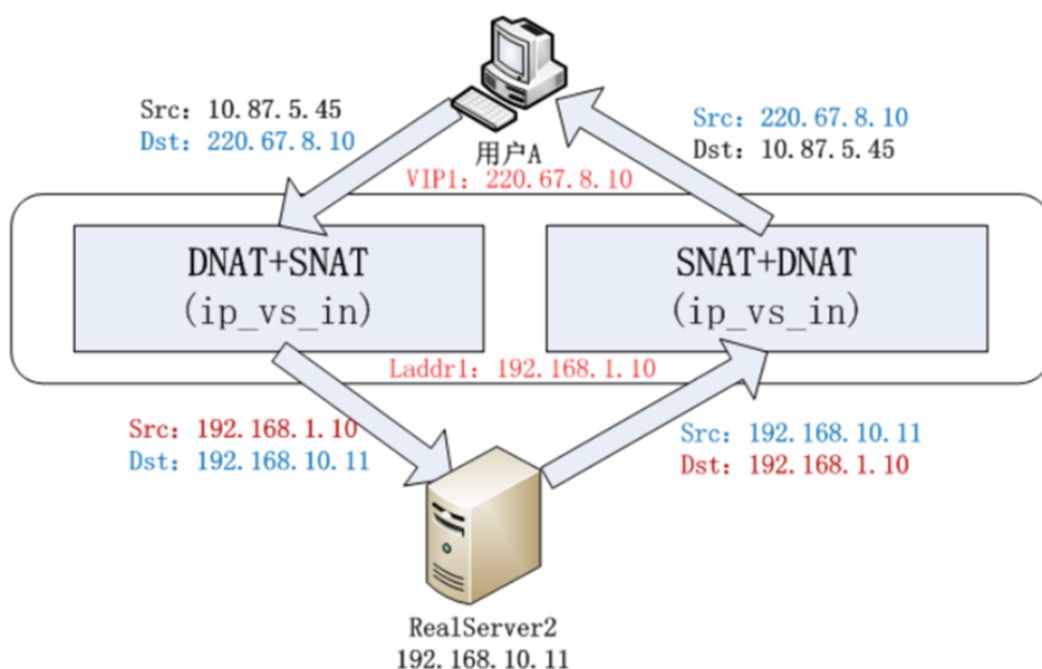
2. FNAT模式原理图

• FULLNAT转发模式



• FULLNAT实现原理

老男孩linux高薪实战培训



1.6 LVS的不同模式优缺点比较

-	NAT	/TUN	DR	FNAT
节点服务器	config dr gw	Tunneling	Non-arp device/tie vip	config dr gw
网络	private	LAN/WAN	LAN	LAN/WAN
节点数量	low (10~20)	High (100)	High (100)	High (1W)
节点服务器网关	load balancer	Own router(能出网)	Own router(能出网)	任意LB
优点	地址和端口转换	WAN环境	性能最高	支持大并发
缺点	瓶颈大效率低	系统需要支持隧道协议	不能跨出LAN	复杂

1.7 LVS调度算法

LVS的调度算法决定了如何在集群节点之间分布工作负荷。当Director调度器收到来自客户端计算机访问它的VIP上的集群服务的入站请求时，Director调度器必须决定哪个集群节点应该处理请求。Director调度器可用于做出该决定的调度方法分成两个基本类别：

- 固定调度算法：rr,wrr,dh,sh
- 动态调度算法：wlc,lc,lblc,lbicr,SED,NQ

常用算法为rr,wrr,wlc

10种调度算法见如下表格：

算法	说明
rr	轮循调度(Round-Robin)，它将请求依次分配不同的RS节点，也就是在RS节点中均摊请求。
wrr	加权轮循调度(Weighted Round-Robin)，它将依据不同RS节点的权值分配任务。
dh	目的地址哈希调度 (Destination Hashing) 以目的地址为关键字查找一个静态hash表来获得需要的RS。
sh	源地址哈希调度(Source Hashing) 以源地址为关键字查找一个静态hash表来获得需要的RS。
wlc	加权最小连接数调度(Weighted Least-Connection)。
lc	最小连接数调度(Least-Connection), IPVS表存储了所有的活动的连接。
lblc	基于地址最小连接数调度，将来自同一目的地址请求分配给同一台RS。
lbicr	基于地址带重复最小连接数调度。
SED	最短的期望的延迟 (Shortest Expected Delay Scheduling SED) (SED) 基于wlc算法。
NQ	最少队列调度 (Never Queue Scheduling NQ) (NQ) 无需队列。

1.8 LVS负载均衡实践

1.8.1 环境准备

1.MySQL负载均衡

外部IP地址 内部IP地址 角色 备注

L4:

10.0.0.15	172.16.1.15	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17
10.0.0.16	172.16.1.16	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17

10.0.0.7	172.16.1.7	RS1 (真实服务器)	MySQL
10.0.0.51	172.16.1.51	RS2 (真实服务器)	MySQL

2.L4+L7+WEB 大规模web负载均衡

L4:

10.0.0.15	172.16.1.15	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17
10.0.0.16	172.16.1.16	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17

L7:

10.0.0.5	172.16.1.5	LVS调度器 (Director)	
10.0.0.6	172.16.1.6	LVS调度器 (Director)	
10.0.0.7	172.16.1.7	RS1 (真实服务器)	web01
10.0.0.8	172.16.1.8	RS2 (真实服务器)	web02

1.8.2 项目架构图

1.8.3 LVS负载均衡安装

```
1 yum install ipvsadm -y
2 rpm -qa ipvsadm #ipvsadm-1.27-8.el7.x86_64
3 modprobe ip_vs #把ipvs加入到内核
4 lsmod|grep ip_vs #检查内核是否有ip_vs
5 uname -r #3.10.0-1160.31.1.el7.x86_64
6 ln -s /usr/src/kernels/3.10.0-1160.31.1.el7.x86_64 /usr/src/linux #yum
install kernel-devel -y
7 ls -l /usr/src/
8 #lrwxrwxrwx 1 root root 44 8月 1 18:04 linux -> /usr/src/kernels/3.10.0-
1160.31.1.el7.x86_6
9 特别注意:
10 1)ln命令链接路径要和uname -r输出结果内核版本对应。
11 2)如果没有/usr/src/kernels/xx路径,可通过yum install kernel-devel -y安装。
```

1.8.4 场景1：实现MySQL负载均衡

1.配置LVS虚拟IP (VIP)


```

1 ifconfig eth1:18 172.16.1.18/24 up      #==>简便写法
2 #route add -host 172.16.1.18 dev eth1  #==添加主机路由，也可不加此行。
3
4 ◎配置后的检查结果：
5 ifconfig eth1:0
6
7 C:\Users\oldboy>ping 172.16.1.18
8 正在 Ping 172.16.1.18 具有 32 字节的数据：
9 来自 172.16.1.18 的回复: 字节=32 时间<1ms TTL=64
10 #提示：到这里说明VIP地址已经配好，并可以使用了。

```

2.手工执行配置添加LVS服务并增加两台RS ipvsadm

```

1 ipvsadm -C                                #<== -C          clear the whole table
2 ipvsadm --set 30 5 60                    #<== --set tcp tcpfin udp      set
   connection timeout values
3 ipvsadm -A -t 172.16.1.18:3306 -s wrr    #--add-service -A    add virtual
   service with options
4 #ipvsadm -A -t 172.16.1.18:3306 -s wrr -p 20
5 ipvsadm -a -t 172.16.1.18:3306 -r 172.16.1.51:3306 -g -w 1 #dr模式
6 ipvsadm -a -t 172.16.1.18:3306 -r 172.16.1.7:3306 -g -w 1
7
8 # ipvsadm -a|e -t|u|f service-address -r server-address [options]
9
10 [root@lb4-01 ~]# ipvsadm -Ln
11 IP Virtual Server version 1.2.1 (size=4096)
12 Prot LocalAddress:Port Scheduler Flags
13   -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
14 TCP  172.16.1.18:3306 wrr
15   -> 172.16.1.7:3306             Route    1         0         0
16   -> 172.16.1.51:3306           Route    1         0         0
17
18 [root@lb4-01 ~]# ipvsadm -Ln --stats
19 IP Virtual Server version 1.2.1 (size=4096)
20 Prot LocalAddress:Port          Conns    InPkts  OutPkts  InBytes
   OutBytes
21   -> RemoteAddress:Port
22 TCP  172.16.1.18:3306              0         0         0         0
   0
23   -> 172.16.1.7:3306              0         0         0         0
   0
24   -> 172.16.1.51:3306              0         0         0         0
   0

```

[删除方法]

```

1 #ipvsadm -D -t 172.16.1.18:3306 -s wrr
2 #ipvsadm -D -t 172.16.1.18:3306
3 #ipvsadm -d -t 172.16.1.18:3306 -r 172.16.1.18:3306 <==正确
4 #ipvsadm -d -t 172.16.1.18:3306 -r 172.16.1.18:3306 -g -w 1 <==不好用

```

[相关参数说明]

```

1 [root@oldboy ~]# ipvsadm -help
2 # --clear          -C          clear the whole table

```

```

3
4 # --add-service -A add virtual service with options
5
6 # --tcp-service -t service-address service-address is host[:port]
7
8 # --scheduler -s scheduler one of
  rr|wrr|lc|wlc|lb|lcr|dh|sh|sed|nq,
9
10 # --add-server -a add real server with options
11
12 # --real-server -r server-address server-address is host (and port)
13
14 # --masquerading -m masquerading (NAT)
15
16 # --gatewaying -g gatewaying (direct routing) (default)
17
18 # --delete-server -d delete real server
19
20 # --persistent -p [timeout] persistent service (会话保持功能)
21
22 # --set tcp tcpfin udp set connection timeout values
23
24 # --weight -w weight capacity of real server
25
26 # --ipip -i ipip encapsulation (tunneling)
27
28 提示: 更多参数请ipvsadm -help自行查看
29 ⑥命令执行过程及检查配置的执行结果
30 ipvsadm -C
31 ipvsadm --set 30 5 60
32 ipvsadm -A -t 172.16.1.18:3306 -s wrr -p 20
33 ipvsadm -a -t 172.16.1.18:3306 -r 172.16.1.51:3306 -g -w 1
34 ipvsadm -a -t 172.16.1.18:3306 -r 172.16.1.7:3306 -g -w 1
35 ipvsadm -L -n --sort
36
37
38
39 ipvsadm -d -t 172.16.1.18:3306 -r 172.16.1.51:80 #==>删除测试
40 ipvsadm -L -n --sort
41
42 ipvsadm -a -t 172.16.1.18:3306 -r 172.16.1.51:80
43 ipvsadm -L -n --sort
44
45 此时, 可以打开浏览器访问http://172.16.1.18体验结果, 如果没意外, 是无法访问的。

```

3.手工在RS端绑定lo网卡及抑制ARP

```

1  @每台real server端执行
2  命令:
3  ifconfig lo:18 172.16.1.18/32 up
4  #route add -host 172.16.1.18 dev lo
5
6  #工作中写到配置文件见后文
7
8  #centos7
9  ip addr add 172.16.1.18/32 dev lo label lo:50
10 #route add -host 172.16.1.18 dev lo
11
12 每个集群节点上的环回接口（lo）设备上被绑定VIP地址（其广播地址是其本身，子网掩码是
    255.255.255.255，采取可变量掩码方式把网段划分成只含一个主机地址的目的是避免ip地址冲突）
    允许LVS-DR集群中的集群节点接受发向该VIP地址的数据包，这会有一个非常严重的问题发生，集群内
    部的真实服务器将尝试回复来自正在请求VIP客户端的ARP广播，这样所有的真实服务器都将声称自己拥
    有该VIP地址，这时客户端将有可能直接发送请求数据包到某台真实服务器上，从而破坏了DR集群的负
    载均衡策略。因此，必须要抑制所有真实服务器响应目标地址为VIP的ARP广播，而把客户端ARP广播响
    应交给负载均衡器。

```

4.手工在RS端抑制ARP响应

```

1  @抑制ARP响应方法如下:
2  echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
3  echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
4  echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
5  echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
6
7  以RS1为例演示执行过程如下:
8  ifconfig lo:18 172.16.1.18 netmask 255.255.255.255 up
9  #route add -host 172.16.1.18 dev lo
10 cat /proc/sys/net/ipv4/conf/lo/arp_ignore
11 cat /proc/sys/net/ipv4/conf/all/arp_ignore
12 cat /proc/sys/net/ipv4/conf/all/arp_announce
13 cat /proc/sys/net/ipv4/conf/lo/arp_announce
14 echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
15 echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
16 echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
17 echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
18
19 测试最终LVS对数据库负载是否成功:
20 采用web02测试:
21     phpmyadmin
22     mysql 测试成功
23
24  windows:navicat...需要vmware 映射
25
26  实际教学使用mysql客户端测试，在web02上
27  yum install mariadb -y
28  mysql -ublog -poldboy123 -h172.16.1.18 #18是VIP，用户密码为前面课程提前设置。
29
30  #####停掉了51，发现连不了
31  lvs没有健康检查功能，51停掉了，仍然请求发给51，需要手工清理51
32  [root@lb4-01 ~]# ipvsadm -Ln
33  IP Virtual Server version 1.2.1 (size=4096)
34  Prot LocalAddress:Port Scheduler Flags
35      -> RemoteAddress:Port          Forward Weight ActiveConn InActConn

```

```

36 TCP 172.16.1.18:3306 wrr persistent 20
37   -> 172.16.1.7:3306           Route 1 0 0
38   -> 172.16.1.51:3306          Route 1 0 4
39 [root@lb4-01 ~]# ipvsadm -d -t 172.16.1.18:3306 -r 172.16.1.51:3306
40 [root@lb4-01 ~]# ipvsadm -Ln
41 IP Virtual Server version 1.2.1 (size=4096)
42 Prot LocalAddress:Port Scheduler Flags
43   -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
44 TCP 172.16.1.18:3306 wrr persistent 20
45   -> 172.16.1.7:3306           Route 1 0 0
46 重连成功，LVS的健康检查要靠keepalived配合实现，后文讲

```

5.抑制ARP脚本实现

```

1  开发脚本配置LVS RS真实服务器端
2  #!/bin/bash
3  # Written by oldboy (31333741@qq.com)
4  # QQ:31333741
5  # description: Config realserver lo and apply noarp
6  VIP=(
7      172.16.1.18
8  )
9  . /etc/rc.d/init.d/functions
10 case "$1" in
11     start)
12         for ((i=0; i<`echo ${#VIP[*]}`; i++))
13         do
14             interface="lo:`echo ${VIP[$i]}|awk -F . '{print $4}'`"
15             /sbin/ifconfig $interface ${VIP[$i]} broadcast ${VIP[$i]} netmask
16             255.255.255.255 up
17             done
18             echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
19             echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
20             echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
21             echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
22             action "Start LVS of RearServer.by oldlboy"
23             ;;
24         stop)
25             for ((i=0; i<`echo ${#VIP[*]}`; i++))
26             do
27                 interface="lo:`echo ${VIP[$i]}|awk -F . '{print $4}'`"
28                 /sbin/ifconfig $interface ${VIP[$i]} broadcast ${VIP[$i]}
29                 netmask 255.255.255.255 down
30             done
31             echo "close LVS Directorserver"
32             if [ ${#VIP[*]} -eq 1];then
33                 echo "0" >/proc/sys/net/ipv4/conf/lo/arp_ignore
34                 echo "0" >/proc/sys/net/ipv4/conf/lo/arp_announce
35                 echo "0" >/proc/sys/net/ipv4/conf/all/arp_ignore
36                 echo "0" >/proc/sys/net/ipv4/conf/all/arp_announce
37             fi
38             action "Close LVS of RearServer.by old2boy"
39             ;;
40     *)
41         echo "Usage: $0 {start|stop}"
42         exit 1
43     esac

```

```

42
43 #工作中可以将lo网卡绑定写到配置文件：
44 cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-
scripts/ifcfg-lo:18
45 [root@db01 network-scripts]# cat ifcfg-lo:18
46 TYPE="Ethernet"
47 PROXY_METHOD="none"
48 BROWSER_ONLY="no"
49 BOOTPROTO="none"
50 DEFROUTE="yes"
51 NAME="lo:18"
52 DEVICE="lo:18"
53 ONBOOT="yes"
54 IPADDR="172.16.1.18"
55 PREFIX="32

```

6.arp抑制技术参数说明

```

1  arp_ignore- INTEGER
2  定义对目标地址为本地IP的ARP询问不同的应答模式
3  0 -(默认值)： 回应任何网络接口上对任何本地IP地址的arp查询请求。
4  1 -只回答目标IP地址是来访网络接口本地地址的ARP查询请求。
5  2 -只回答目标IP地址是来访网络接口本地地址的ARP查询请求,且来访IP必须在该网络接口的子网段
   内。
6  3 -不回应该网络界面的arp请求,而只对设置的唯一和连接地址做出回应。
7  4-7 -保留未使用。
8  8 -不回应所有(本地地址)的arp查询。
9  arp_announce - INTEGER
10  对网络接口上,本地IP地址的发出的,ARP回应,作出相应级别的限制:
11  确定不同程度的限制,宣布对来自本地源IP地址发出Arp请求的接口
12  0 -(默认) 在任意网络接口(eth0,eth1,lo)上的任何本地地址
13  1 -尽量避免不在该网络接口子网段的本地地址做出arp回应.当发起ARP请求的源IP地址是被设置应
   该经由路由达到此网络接口的时候很有用.此时会检查来访IP是否为所有接口上的子网段内ip之一。如
   果该来访IP不属于各个网络接口上的子网段内,那么将采用级别2的方式来进行处理。
14  2 -对查询目标使用最适当的本地地址,在此模式下将忽略这个IP数据包的源地址并尝试选择能与该地
   址通信的本地地址,首要选择所有的网络接口的子网中外出访问子网中包含该目标IP地址的本地地
   址。如果没有合适的地址被发现,将选择当前的发送网络接口或其他的有可能接受到该ARP回应的网络接
   口来进行发送.限制了使用本地的vip地址作为优先的网络接口

```

1.8.5 场景2：实现LVS+keepalived db高可用

```

1  lb4-01配置:
2  [root@lb4-01 keepalived]# cat keepalived.conf
3  global_defs {
4      router_id lb01
5  }
6  vrrp_instance VI_2 {
7      state MASTER
8      interface eth1
9      virtual_router_id 52
10     priority 150
11     advert_int 1
12     authentication {
13         auth_type PASS
14         auth_pass 1111
15     }

```

```

16     virtual_ipaddress {
17         172.16.1.18/24 dev eth1 label eth1:18
18     }
19 }
20
21 #port 3306
22 ##oldboy service virtual_server
23 virtual_server 172.16.1.18 3306 {
24     delay_loop 6
25     lb_algo wrr
26     lb_kind DR
27     persistence_timeout 20
28     protocol TCP
29     real_server 172.16.1.7 3306 {
30         weight 1
31         TCP_CHECK {
32             connect_timeout 5
33             #nb_get_retry 3
34             delay_before_retry 3
35             connect_port 3306
36         }
37     }
38
39     real_server 172.16.1.51 3306 {
40         weight 1
41         TCP_CHECK {
42             connect_timeout 5
43             # nb_get_retry 3
44             delay_before_retry 3
45             connect_port 3306
46         }
47     }
48 }

```

```

1  lb4-02配置:
2  [root@lb4-02 keepalived]# cat keepalived.conf
3  global_defs {
4      router_id lb4-02
5  }
6  vrrp_instance VI_2 {
7      state BACKUP
8      interface eth1
9      virtual_router_id 52
10     priority 50
11     advert_int 1
12     authentication {
13         auth_type PASS
14         auth_pass 1111
15     }
16     virtual_ipaddress {
17         172.16.1.18/24 dev eth1 label eth1:18
18     }
19 }
20
21
22
23 #port 3306

```

```

24  ##oldboy service  virtual_server
25  virtual_server 172.16.1.18 3306 {
26      delay_loop 6
27      lb_algo wrr
28      lb_kind DR
29      persistence_timeout 20
30      protocol TCP
31  real_server 172.16.1.7 3306 {
32      weight 1
33      TCP_CHECK {
34          connect_timeout 5
35          #nb_get_retry 3
36          delay_before_retry 3
37          connect_port 3306
38      }
39  }
40
41  real_server 172.16.1.51 3306 {
42      weight 1
43      TCP_CHECK {
44          connect_timeout 5
45          # nb_get_retry 3
46          delay_before_retry 3
47          connect_port 3306
48      }
49  }
50  }

```

重启keep完成

1.8.6 场景3：实现Web4层负载及后端7层反代+Web节点

1) 环境说明

2.L4+L7+WEB 大规模web负载均衡

L4:

10.0.0.15	172.16.1.15	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17
10.0.0.16	172.16.1.16	LVS调度器 (Director)	对外提供服务的VIP为10.0.0.17

L7:

10.0.0.5	172.16.1.5	nginx lb01	测好了。
10.0.0.6	172.16.1.6	nginx lb02	
10.0.0.7	172.16.1.7	RS1 (真实服务器)	web01
10.0.0.8	172.16.1.8	RS2 (真实服务器)	web02

2)测试节点可用性

```

1  [root@lb4-02 keepalived]# curl -H"host:www.etiantian.org" 172.16.1.7
2  web01
3  [root@lb4-02 keepalived]# curl -H"host:www.etiantian.org" 172.16.1.8
4  web02, 老男孩Linux77最优班级
5  remote_addr: 172.16.1.16

```

3) 配置keepalived lb4-01

```

1  [root@lb4-01 keepalived]# cat keepalived.conf

```



```

2 global_defs {
3     router_id lb01
4 }
5 vrrp_instance VI_1 {
6     state BACKUP
7     interface eth0
8     virtual_router_id 53
9     priority 50
10    advert_int 1
11    authentication {
12        auth_type PASS
13        auth_pass 1111
14    }
15    virtual_ipaddress {
16        10.0.0.17/24 dev eth0 label eth0:17
17    }
18 }
19 #web
20 ##oldboy service virtual_server
21 virtual_server 10.0.0.17 80 {
22     delay_loop 6
23     lb_algo wrr
24     lb_kind DR
25     persistence_timeout 20
26     protocol TCP
27 }
28 real_server 10.0.0.5 80 {
29     weight 1
30     TCP_CHECK {
31         connect_timeout 5
32         #nb_get_retry 3
33         delay_before_retry 3
34         connect_port 80
35     }
36 }
37
38 real_server 10.0.0.6 80 {
39     weight 1
40     TCP_CHECK {
41         connect_timeout 5
42         # nb_get_retry 3
43         delay_before_retry 3
44         connect_port 80
45     }
46 }
47 }
48
49 ;=====
50 vrrp_instance VI_2 {
51     state MASTER
52     interface eth1
53     virtual_router_id 52
54     priority 150
55     advert_int 1
56     authentication {
57         auth_type PASS
58         auth_pass 1111
59     }

```

```

60     virtual_ipaddress {
61         172.16.1.18/24 dev eth1 label eth1:18
62     }
63 }
64
65
66
67 #port 3306
68 ##oldboy service virtual_server
69 virtual_server 172.16.1.18 3306 {
70     delay_loop 6
71     lb_algo wrr
72     lb_kind DR
73     persistence_timeout 20
74     protocol TCP
75     real_server 172.16.1.7 3306 {
76         weight 1
77         TCP_CHECK {
78             connect_timeout 5
79             #nb_get_retry 3
80             delay_before_retry 3
81             connect_port 3306
82         }
83     }
84
85     real_server 172.16.1.51 3306 {
86         weight 1
87         TCP_CHECK {
88             connect_timeout 5
89             # nb_get_retry 3
90             delay_before_retry 3
91             connect_port 3306
92         }
93     }
94 }

```

4) 配置keepalived lb4-02

```

1 [root@lb4-02 keepalived]# cat keepalived.conf
2 global_defs {
3     router_id lb4-02
4 }
5
6 vrrp_instance VI_1 {
7     state MASTER
8     interface eth0
9     virtual_router_id 53
10    priority 150
11    advert_int 1
12    authentication {
13        auth_type PASS
14        auth_pass 1111
15    }
16    virtual_ipaddress {
17        10.0.0.17/24 dev eth0 label eth0:17
18    }
19 }

```

```
20 #web
21 ##oldboy service virtual_server
22 virtual_server 10.0.0.17 80 {
23     delay_loop 6
24     lb_algo wrr
25     lb_kind DR
26     persistence_timeout 20
27     protocol TCP
28     real_server 10.0.0.5 80 {
29         weight 1
30         TCP_CHECK {
31             connect_timeout 5
32             #nb_get_retry 3
33             delay_before_retry 3
34             connect_port 80
35         }
36     }
37
38     real_server 10.0.0.6 80 {
39         weight 1
40         TCP_CHECK {
41             connect_timeout 5
42             # nb_get_retry 3
43             delay_before_retry 3
44             connect_port 80
45         }
46     }
47 }
48 vrrp_instance VI_2 {
49     state BACKUP
50     interface eth1
51     virtual_router_id 52
52     priority 50
53     advert_int 1
54     authentication {
55         auth_type PASS
56         auth_pass 1111
57     }
58     virtual_ipaddress {
59         172.16.1.18/24 dev eth1 label eth1:18
60     }
61 }
62
63
64
65 #port 3306
66 ##oldboy service virtual_server
67 virtual_server 172.16.1.18 3306 {
68     delay_loop 6
69     lb_algo wrr
70     lb_kind DR
71     persistence_timeout 20
72     protocol TCP
73     real_server 172.16.1.7 3306 {
74         weight 1
75         TCP_CHECK {
76             connect_timeout 5
77             #nb_get_retry 3
```

```

78     delay_before_retry 3
79     connect_port 3306
80 }
81 }
82
83 real_server 172.16.1.51 3306 {
84     weight 1
85     TCP_CHECK {
86         connect_timeout 5
87         # nb_get_retry 3
88         delay_before_retry 3
89         connect_port 3306
90     }
91 }
92 }

```

5) web节点分别 (web01,web02) 配置VIP绑定, 及抑制ARP (脚本实现)

```

1  [root@lb01 conf.d]# mkdir /server/scripts -p
2  [root@lb01 conf.d]# cd /server/scripts/
3  [root@lb01 scripts]# vim ipvs.sh
4  #!/bin/bash
5
6  # Written by oldboy (31333741@qq.com)
7
8  # QQ:31333741
9
10 # description: Config realserver lo and apply noarp
11
12 VIP=(
13     10.0.0.17
14 )
15 . /etc/rc.d/init.d/functions
16 case "$1" in
17 start)
18     for ((i=0; i<`echo ${#VIP[*]}`; i++))
19     do
20         interface="lo:`echo ${VIP[$i]}|awk -F . '{print $4}'`"
21         /sbin/ifconfig $interface ${VIP[$i]} broadcast ${VIP[$i]} netmask
22 255.255.255.255 up
23     done
24     echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
25     echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
26     echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
27     echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
28     action "Start LVS of RearServer.by oldboy"
29     ;;
30 stop)
31     for ((i=0; i<`echo ${#VIP[*]}`; i++))
32     do
33         interface="lo:`echo ${VIP[$i]}|awk -F . '{print $4}'`"
34         /sbin/ifconfig $interface ${VIP[$i]} broadcast ${VIP[$i]}
35  netmask 255.255.255.255 down
36     done
37     echo "close LVS Directorserver"
38     if [ ${#VIP[*]} -eq 1];then
39         echo "0" >/proc/sys/net/ipv4/conf/lo/arp_ignore

```

```

38     echo "0" >/proc/sys/net/ipv4/conf/lo/arp_announce
39     echo "0" >/proc/sys/net/ipv4/conf/all/arp_ignore
40     echo "0" >/proc/sys/net/ipv4/conf/all/arp_announce
41 fi
42     action "Close LVS of RearServer.by old2boy"
43 ;;
44 *)
45     echo "Usage: $0 {start|stop}"
46     exit 1
47 esac
48
49 [root@lb01 scripts]# sh ipvs.sh
50 Usage: ipvs.sh {start|stop}
51 [root@lb01 scripts]# sh ipvs.sh start
52 Start LVS of RearServer.by old1boy [ 确定 ]
53 ifconfig查看

```

6)获取lvs+nginx+web前端用户真实IP

```

1 关于FullNat模式的 Toa 实现原理
2  https://blog.csdn.net/liwei0526vip/article/details/106108844
3
4  server {
5             listen 80;          #添加proxy_protocol
6             #listen 80 proxy_protocol;      #添加proxy_protocol
7             server_name www.etiantian.org;
8             set_real_ip_from 172.16.1.0/24; #添加七层负载前经过的代理
9             #IP地址
10            real_ip_header proxy_protocol; #将proxy_protocol获取的
11            #IP赋值给$remote_addr
12            location / {
13                proxy_pass http://www;
14                proxy_set_header Host $http_host;
15                proxy_set_header X-Forwarded-For
16                $proxy_protocol_addr;
17                #将proxy_protocol真实客户端的IP地址赋值给X-Forwarded-
18                #For变量携带至后端
19                proxy_set_header    X-Forwarded-For
20                $proxy_add_x_forwarded_for;
21            }
22        }
23
24 找到原因了额
25  #proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for; 这个开启。
26
27 客户端访问LVS是携带源IP的，LVS仅仅是转发，DR修改目的MAC，并没有改变源IP。所以客户源IP可
28 以被nginx lb01获取到，然后通过x-forward-for传给web节点
29
30 [root@web01 conf.d]# tail /var/log/nginx/access.log
31 ---,--172.16.1.5--,---10.0.0.1---,----10.0.0.1, 172.16.1.5---
32 -, -,01/Aug/2021:19:33:24 +0800,www.etiantian.org,GET /
33 HTTP/1.0,200,-,-,0.000,Mozilla/5.0 (Windows NT 10.0; Win64; x64)
34 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36 -
35 - -

```

```
25 ---,--172.16.1.5--,---10.0.0.1---,----10.0.0.1, 172.16.1.5---  
-,-,01/Aug/2021:19:33:24 +0800,www.etiantian.org,GET /  
HTTP/1.0,200,-,-,0.000,Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36 -  
- -
```

1.9 LVS视频教程

如果视频不能免费观看，可加群入群领取免费视频。

2022LVS负载均衡集群企业级实践（老男孩创始人亲授）

[点击进入学习](#)

](https://ke.oldboyedu.com/detail/term_61e549b853e90_W6Y0I7/25?fromH5=true)



视频目录结构：

- oldboy-01-LVS集群介绍.mp4
- oldboy-02-LVS集群名词介绍.mp4
- oldboy-03-LVS集群NAT模式原理精讲1.mp4
- oldboy-04-LVS集群NAT模式原理精讲2.mp4
- oldboy-05-LVS集群TUN模式原理精讲1.mp4
- oldboy-06-LVS集群TUN模式原理精讲2.mp4
- oldboy-07-LVS集群DR模式原理精讲1.mp4
- oldboy-08-LVS集群DR模式原理精讲2.mp4
- oldboy-09-LVS集群FULLNAT模式原理精讲1.mp4
- oldboy-10-LVS集群不同模式区别及调度算法介绍i.mp4
- oldboy-11-LVS集群场景实践准备及安装LVS.mp4
- oldboy-12-LVS集群场景实践准备及安装LVS2.mp4
- oldboy-13-LVS集群数据库负载场景实践及故障排错.mp4
- oldboy-14-LVS集群绑定VIP及抑制ARP说明.mp4
- oldboy-15-配置keepalived管理LVS集群并实现数据库负载实践.mp4
- oldboy-16-配置keepalived管理LVS集群并实现Web四七层负载实践.mp4

oldboy-17-LVS四七层负载均衡获取真实用户IP (未完) .mp4
oldboy-18-获取lvs+nginx+web前端用户真实IP说明.mp4
oldboy-19-梳理昨天lvs+nginx4-层负载均衡部署流程.mp4

2.0 总结

由于文档整理较费时费力，难免会有疏漏和不妥当地方，敬请谅解。

本文档已经发布到www.yunweibase.com/www.oldboyedu.com网站，里面还会更新更多的教程，敬请期待。

如需获取干货资料，可关注公众号获取，如咨询技术交流，可加群以及加微信，老男孩教育陪你进步！



关注公众号 领取更多干货



学习交流，加V联系

Linux技术交流QQ群784850820

附：往期干货文档笔记下载

PS：点击标题链接直接跳转

- [网络安全大厂面试真题.pdf下载](#)
- [公司网站架构图.png下载](#)
- [Linux三剑客-sed详解.pdf下载](#)
- [Git命令手册.pdf下载](#)
- [Linux速查备忘手册+知识点总结](#)

成功绝非偶然，而是正确选择和持续努力后的必然！——老男孩老师 2022