1. [案例1：Keepalived高可用服务器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY02/CASE/01/index.html#case1)
2. [案例2：Keepalived+LVS服务器](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY02/CASE/01/index.html#case2)
3. [案例3：配置HAProxy负载平衡集群](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/CLUSTER/DAY02/CASE/01/index.html#case3)

**1 案例1：Keepalived高可用服务器**

**1.1 问题**

准备三台Linux服务器，两台做Web服务器，并部署Keepalived高可用软件，一台作为客户端主机，实现如下功能：

* 使用Keepalived实现web服务器的高可用
* Web服务器IP地址分别为192.168.4.100和192.168.4.200
* Web服务器的浮动VIP地址为192.168.4.80
* 客户端通过访问VIP地址访问Web页面

**1.2 方案**

使用3台虚拟机，2台作为Web服务器，并部署Keepalived、1台作为客户端，拓扑结构如图-1所示，主机配置如表-1所示。

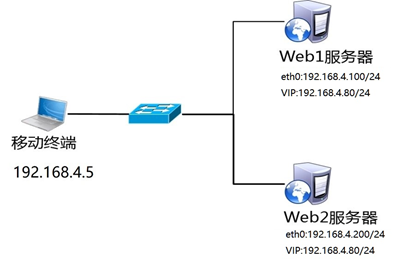
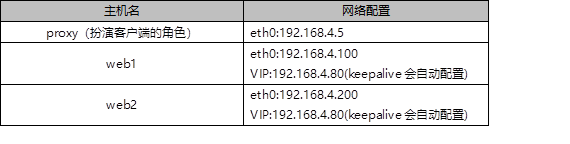


图-1

表-1



**1.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：配置网络环境（如果在前面课程已经完成该配置，可以忽略此步骤）**

1）设置Web1服务器网络参数、配置Web服务（不能照抄网卡名称）

1. **[**root@web1 **~]**# nmcli connection modify eth0 ipv4**.**method manual ipv4**.**addresses **192.168.4.100/24** connection**.**autoconnect yes
2. **[**root@web1 **~]**# nmcli connection up eth0
3. **[**root@web1 **~]**# yum **-**y install httpd        #安装软件
4. **[**root@web1 **~]**# echo "192.168.4.100" **>** /var/www**/**html**/**index**.**html    #创建网页文件
5. **[**root@web1 **~]**# systemctl restart httpd        #启动服务器

2）设置Web2服务器网络参数、配置Web服务（不能照抄网卡名称）

1. **[**root@web2 **~]**# nmcli connection modify eth0 ipv4**.**method manual ipv4**.**addresses **192.168.4.200/24** connection**.**autoconnect yes
2. **[**root@web2 **~]**# nmcli connection up eth0
3. **[**root@web2 **~]**# yum **-**y install httpd        #安装软件
4. **[**root@web2 **~]**# echo "192.168.4.200" **>** /var/www**/**html**/**index**.**html    #创建网页文件
5. **[**root@web2 **~]**# systemctl restart httpd        #启动服务器

3）配置proxy主机的网络参数（如果已经设置，可以忽略此步骤）

备注：这个实验，我们使用proxy当作客户端主机，网卡名称不能照抄。

1. **[**root@proxy **~]**# nmcli connection modify eth0 ipv4**.**method manual ipv4**.**addresses **192.168.4.5/24** connection**.**autoconnect yes
2. **[**root@proxy **~]**# nmcli connection up eth0

**步骤二：安装Keepalived软件**

***注意：两台Web服务器做相同的操作。***

1. **[**root@web1 **~]**# yum install **-**y keepalived
2. **[**root@web2 **~]**# yum install **-**y keepalived

**步骤三：部署Keepalived服务**

1）修改web1服务器Keepalived配置文件

1. **[**root@web1 **~]**# vim **/**etc**/**keepalived**/**keepalived**.**conf
2. global\_defs **{**
3. router\_id web1        #**12**行，设置路由ID号（实验需要修改）
4. vrrp\_iptables            #**13**行，清除防火墙的拦截规则（实验需要修改，手动添加该行）
5. **}**
6. vrrp\_instance VI\_1 **{**
7. state MASTER            #**21**行，主服务器为MASTER（备服务器需要修改为BACKUP）
8. **interface** eth0            #**22**行，VIP配在哪个网卡（实验需要修改，不能照抄网卡名）
9. virtual\_router\_id **51**        #**23**行，主备服务器VRID号必须一致
10. priority **100**            #**24**行，服务器优先级**,**优先级高优先获取VIP
11. advert\_int **1**
12. authentication **{**
13. auth\_type pass
14. auth\_pass **1111**
15. **}**
16. virtual\_ipaddress **{**        #**30~32**行，谁是主服务器谁获得该VIP（实验需要修改）
17. **192.168.4.80**
18. **}**
19. **}**

2）修改web2服务器Keepalived配置文件

1. **[**root@web2 **~]**# vim **/**etc**/**keepalived**/**keepalived**.**conf
2. global\_defs **{**
3. router\_id web2        #**12**行，设置路由ID号（实验需要修改）
4. vrrp\_iptables            #**13**行，清除防火墙的拦截规则（实验需要修改，手动添加该行）
5. **}**
6. vrrp\_instance VI\_1 **{**
7. state BACKUP            #**21**行，备服务器为BACKUP（实验需要修改）
8. **interface** eth0            #**22**行，VIP配在哪个网卡（实验需要修改，不能照抄网卡名）
9. virtual\_router\_id **51**        #**23**行，主辅VRID号必须一致
10. priority **50**                #**24**行，服务器优先级（实验需要修改）
11. advert\_int **1**
12. authentication **{**
13. auth\_type pass
14. auth\_pass **1111**
15. **}**
16. virtual\_ipaddress **{**        #**30~32**行，谁是主服务器谁配置VIP（实验需要修改）
17. **192.168.4.80**
18. **}**
19. **}**

3）启动服务

1. **[**root@web1 **~]**# systemctl start keepalived
2. **[**root@web2 **~]**# systemctl start keepalived

4）配置防火墙和SELinux

1. **[**root@web1 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
2. **[**root@web1 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
3. **[**root@web1 **~]**# setenforce **0**
4. **[**root@web2 **~]**# firewall**-**cmd **--**set**-default-**zone**=**trusted
5. **[**root@web2 **~]**# sed **-**i '/SELINUX/s/enforcing/permissive/' /etc/selinux**/**config
6. **[**root@web2 **~]**# setenforce **0**

**步骤四：测试**

1）登录两台Web服务器查看VIP信息

1. **[**root@web1 **~]**# ip addr show
2. **[**root@web2 **~]**# ip addr show

2) 客户端访问

客户端使用curl命令连接http://192.168.4.80，查看Web页面；给Web1关机，客户端再次访问http://192.168.4.80，验证是否可以正常访问服务。

**2 案例2：Keepalived+LVS服务器**

**2.1 问题**

使用Keepalived为LVS调度器提供高可用功能，防止调度器单点故障，为用户提供Web服务：

* LVS1调度器真实IP地址为192.168.4.5
* LVS2调度器真实IP地址为192.168.4.6
* 服务器VIP地址设置为192.168.4.15
* 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，真实web服务器权重不同

**2.2 方案**

使用5台虚拟机，1台作为客户端主机、2台作为LVS调度器、2台作为Real Server，实验拓扑环境结构如图-2所示，基础环境配置如表-2所示。

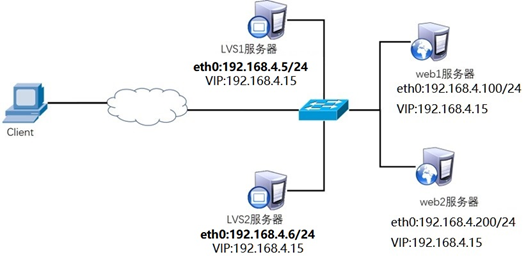
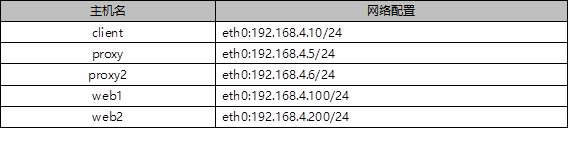


图-2

表-2



注意：所有主机都需要配置IP地址与有效的YUM源，网卡名称仅供参考。

**2.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：配置网络环境**

1）关闭服务（把案例1中给web1和web2安装的keepalived关闭）

***警告：请先将案例1中web1和web2的keepalived关闭！！！***

1. **[**root@web1 **~]**# systemctl stop keepalived
2. **[**root@web2 **~]**# systemctl stop keepalived

2）设置Web1服务器的网络参数（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. #开机是否激活该网卡
16. NAME**=**lo**:0**
17. #网卡名称

注意：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。

写入下面这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。

1. **[**root@web1 **~]**# 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. #arp\_ignore**(**防止进站冲突**)**
8. #arp\_announce**(**防出站冲突**)**
9. **[**root@web1 **~]**# sysctl **-**p #刷新，让配置文件立刻生效

重启网络服务

1. **[**root@web1 **~]**# systemctl restart network        #重启网络服务
2. **[**root@web1 **~]**# ip a s                        #查看IP地址

3）设置Web2服务器的网络参数（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. IPADDR**=192.168.4.15**
6. NETMASK**=255.255.255.255**
7. NETWORK**=192.168.4.15**
8. BROADCAST**=192.168.4.15**
9. ONBOOT**=**yes
10. NAME**=**lo**:0**

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

写入这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。

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. #arp\_ignore**(**防止进站冲突**)**
8. #arp\_announce**(**防出站冲突**)**
9. **[**root@web2 **~]**# sysctl **-**p #刷新，让配置文件立刻生效

重启网络服务

1. **[**root@web2 **~]**# systemctl restart network        #重启网络服务
2. **[**root@web2 **~]**# ip a s                            #查看IP地址

4）配置proxy主机的网络参数(不配置VIP，VIP由keepalvied自动配置)

备注：不能照抄网卡名称。

1. **[**root@proxy **~]**# nmcli connection modify eth0 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.4.5/24** connection**.**autoconnect yes
3. **[**root@proxy **~]**# nmcli connection up eth0

5）配置proxy2主机的网络参数(不配置VIP，VIP由keepalvied自动配置)

注意：按照前面的课程环境，默认没有该虚拟机，需要重新建一台虚拟机proxy2。

备注：不能照抄网卡名称。

1. **[**root@proxy2 **~]**# nmcli connection modify eth0 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.4.6/24** connection**.**autoconnect yes
3. **[**root@proxy2 **~]**# nmcli connection up eth0

**步骤二：配置后台web服务**

1）安装软件，自定义Web页面（web1和web2主机）

1. **[**root@web1 **~]**# yum **-**y install httpd
2. **[**root@web1 **~]**# echo "192.168.4.100" **>** /var/www**/**html**/**index**.**html
3. **[**root@web2 **~]**# yum **-**y install httpd
4. **[**root@web2 **~]**# echo "192.168.4.200" **>** /var/www**/**html**/**index**.**html

2）启动Web服务器软件(web1和web2主机)

1. **[**root@web1 **~]**# systemctl start httpd **;** systemctl enable httpd
2. **[**root@web2 **~]**# systemctl start httpd **;** systemctl enable httpd

**步骤三：调度器安装Keepalived与ipvsadm软件**

***注意：两台LVS调度器执行相同的操作（如何已经安装软件，可忽略此步骤）。***

安装软件

1. **[**root@proxy **~]**# yum install **-**y keepalived
2. **[**root@proxy **~]**# systemctl enable keepalived
3. **[**root@proxy **~]**# yum install **-**y ipvsadm
4. **[**root@proxy **~]**# ipvsadm **-**C
5. **[**root@proxy2 **~]**# yum install **-**y keepalived
6. **[**root@proxy2 **~]**# systemctl enable keepalived
7. **[**root@proxy2 **~]**# yum install **-**y ipvsadm
8. **[**root@proxy2 **~]**# ipvsadm **-**C

**步骤四：部署Keepalived实现LVS-DR模式调度器的高可用**

1）LVS1调度器设置Keepalived，并启动服务（在192.168.4.5主机操作）

1. **[**root@proxy **~]**# vim **/**etc**/**keepalived**/**keepalived**.**conf
2. global\_defs **{**
3. router\_id lvs1        #**12**行，设置路由ID号**(**实验需要修改**)**
4. vrrp\_iptables            #**13**行，清除防火墙的拦截规则（实验需要修改，手动添加）
5. **}**
6. vrrp\_instance VI\_1 **{**
7. state MASTER            #**21**行，主服务器为MASTER
8. **interface** eth0            #**22**行，定义网络接口（不能照抄网卡名）
9. virtual\_router\_id **51**        #**23**行，主辅VRID号必须一致
10. priority **100**            #**24**行，服务器优先级
11. advert\_int **1**
12. authentication **{**
13. auth\_type pass
14. auth\_pass **1111**
15. **}**
16. virtual\_ipaddress **{**        #**30~32**行，配置VIP（实验需要修改）
17. **192.168.4.15**
18. **}**
19. **}**
20. virtual\_server **192.168.4.15** **80** **{**        #设置ipvsadm的VIP规则（实验需要修改）
21. delay\_loop **6**                        #默认健康检查延迟**6**秒
22. lb\_algo rr                            #设置LVS调度算法为RR
23. lb\_kind DR                            #设置LVS的模式为DR（实验需要修改）
24. #persistence\_timeout **50**                #（实验需要删除）
25. #注意persistence\_timeout的作用是保持连接
26. #开启后，客户端在一定时间内**(50**秒**)**始终访问相同服务器
27. protocol TCP                        #TCP协议
28. real\_server **192.168.4.100** **80** **{**        #设置后端web服务器真实IP（实验需要修改）
29. weight **1**                            #设置权重为**1**
30. TCP\_CHECK **{**                        #对后台real\_server做健康检查（实验需要修改）
31. connect\_timeout **3**                #健康检查的超时时间**3**秒
32. nb\_get\_retry **3**                    #健康检查的重试次数**3**次
33. delay\_before\_retry **3**                #健康检查的间隔时间**3**秒
34. **}**
35. **}**
36. real\_server **192.168.4.200** **80** **{**        #设置后端web服务器真实IP（实验需要修改）
37. weight **2**                        #设置权重为**2**
38. TCP\_CHECK **{**                    #对后台real\_server做健康检查（实验需要修改）
39. connect\_timeout **3**            #健康检查的超时时间**3**秒
40. nb\_get\_retry **3**                #健康检查的重试次数**3**次
41. delay\_before\_retry **3**            #健康检查的间隔时间**3**秒
42. **}**
43. **}**
44. **}**
45. **[**root@proxy1 **~]**# systemctl start keepalived
46. **[**root@proxy1 **~]**# ipvsadm **-**Ln        #查看LVS规则
47. **[**root@proxy1 **~]**# ip a s            #查看VIP配置

2）LVS2调度器设置Keepalived（在192.168.4.6主机操作）

1. **[**root@proxy2 **~]**# vim **/**etc**/**keepalived**/**keepalived**.**conf
2. global\_defs **{**
3. router\_id lvs2                     #**12**行，设置路由ID号（实验需要修改）
4. vrrp\_iptables #**13**行，清除防火墙的拦截规则（实验需要修改，手动添加）
5. **}**
6. vrrp\_instance VI\_1 **{**
7. state BACKUP                          #**21**行，从服务器为BACKUP（实验需要修改）
8. **interface** eth0                        #**22**行，定义网络接口（不能照抄网卡名）
9. virtual\_router\_id **51**                 #**23**行，主辅VRID号必须一致
10. priority **50**                      #**24**行，服务器优先级（实验需要修改）
11. advert\_int **1**
12. authentication **{**
13. auth\_type pass
14. auth\_pass **1111**
15. **}**
16. virtual\_ipaddress **{** #**30~32**行，设置VIP（实验需要修改）
17. **192.168.4.15**
18. **}**
19. **}**
20. virtual\_server **192.168.4.15** **80** **{**      #自动设置LVS规则（实验需要修改）
21. delay\_loop **6**
22. lb\_algo rr                  #设置LVS调度算法为RR
23. lb\_kind DR                          #设置LVS的模式为DR（实验需要修改）
24. # persistence\_timeout **50** #（实验需要删除该行）
25. #注意persistence\_timeout的作用是保持连接
26. #开启后，客户端在一定时间内**(50**秒**)**始终访问相同服务器
27. protocol TCP                        #TCP协议
28. real\_server **192.168.4.100** **80** **{** #设置后端web服务器的真实IP（实验需要修改）
29. weight **1**                  #设置权重为**1**
30. TCP\_CHECK **{**                     #对后台real\_server做健康检查（实验需要修改）
31. connect\_timeout **3** #健康检查的超时时间**3**秒
32. nb\_get\_retry **3** #健康检查的重试次数**3**次
33. delay\_before\_retry **3** #健康检查的间隔时间**3**秒
34. **}**
35. **}**
36. real\_server **192.168.4.200** **80** **{** #设置后端web服务器的真实IP（实验需要修改）
37. weight **2**                  #设置权重为**2**，权重可以根据需要修改
38. TCP\_CHECK **{**                        #对后台real\_server做健康检查（实验需要修改）
39. connect\_timeout **3** #健康检查的超时时间**3**秒
40. nb\_get\_retry **3** #健康检查的重试次数**3**次
41. delay\_before\_retry **3** #健康检查的间隔时间**3**秒
42. **}**
43. **}**
44. **[**root@proxy2 **~]**# systemctl start keepalived
45. **[**root@proxy2 **~]**# ipvsadm **-**Ln #查看LVS规则
46. **[**root@proxy2 **~]**# ip a s #查看VIP设置

**步骤五：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。

**3 案例3：配置HAProxy负载平衡集群**

**3.1 问题**

准备4台Linux服务器，两台做Web服务器，1台安装HAProxy，1台做客户端，实现如下功能：

* 客户端访问HAProxy，HAProxy分发请求到后端Real Server
* 开启HAProxy监控页面，及时查看调度器状态
* 设置HAProxy为开机启动

**3.2 方案**

使用4台虚拟机，1台作为HAProxy调度器、2台作为Real Server、1台作为客户端，拓扑结构如图-3所示，具体配置如表-3所示。

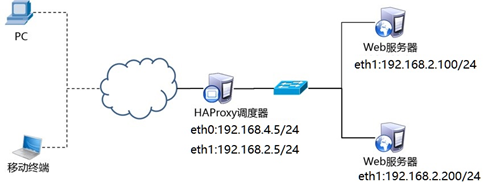
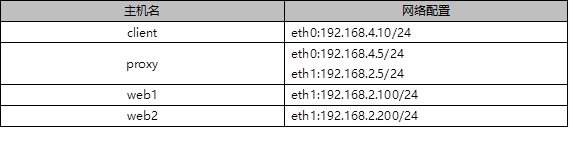


图-3

表-3



为什么Haproxy的实验不需要开启路由，不需要给web服务器配置网关？

Hapoxy是代理服务器（帮你干活的人或物就是你的代理），通讯流程如图-4所示。

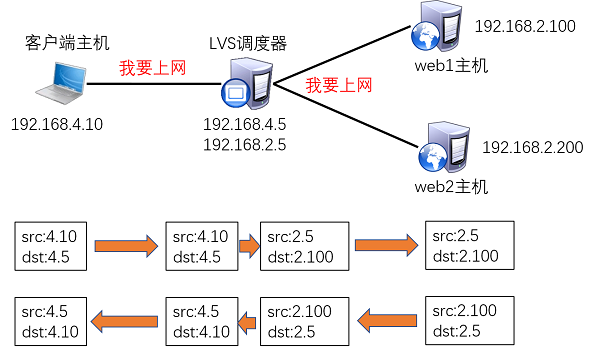


图-4

**3.3 步骤**

实现此案例需要按照如下步骤进行。

web1配置本地真实IP地址（不能照抄网卡名）。

1. **[**root@web1 **~]**# nmcli connection modify eth1 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.2.100/24** connection**.**autoconnect yes
3. **[**root@web1 **~]**# nmcli connection up eth1

Web2配置本地真实IP地址（不能照抄网卡名）。

1. **[**root@web2 **~]**# nmcli connection modify eth1 ipv4**.**method manual **\**
2. ipv4**.**addresses **192.168.2.200/24** connection**.**autoconnect yes
3. **[**root@web2 **~]**# nmcli connection up eth1

proxy关闭keepalived服务，清理LVS规则，不能照抄网卡名。

1. **[**root@proxy **~]**# systemctl stop keepalived
2. **[**root@proxy **~]**# systemctl disable keepalived
3. **[**root@proxy **~]**# ipvsadm **-**C
4. **[**root@proxy **~]**# nmcli connection modify eth0 ipv4**.**method manual **\**
5. ipv4**.**addresses **192.168.4.5/24** connection**.**autoconnect yes
6. **[**root@proxy **~]**# nmcli connection up eth0
7. **[**root@proxy **~]**# nmcli connection modify eth1 ipv4**.**method manual **\**
8. ipv4**.**addresses **192.168.2.5/24** connection**.**autoconnect yes
9. **[**root@proxy **~]**# nmcli connection up eth1

**步骤一：配置后端Web服务器**

设置两台后端Web服务（如果已经配置完成，可忽略此步骤）

1. **[**root@web1 **~]**# yum **-**y install httpd
2. **[**root@web1 **~]**# systemctl start httpd
3. **[**root@web1 **~]**# echo "192.168.2.100" **>** /var/www**/**html**/**index**.**html
4. **[**root@web2 **~]**# yum **-**y install httpd
5. **[**root@web2 **~]**# systemctl start httpd
6. **[**root@web2 **~]**# echo "192.168.2.200" **>** /var/www**/**html**/**index**.**html

**步骤二：部署HAProxy服务器**

1）配置网络，安装软件

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

2）修改配置文件

1. **[**root@proxy **~]**# vim **/**etc**/**haproxy**/**haproxy**.**cfg
2. global
3. log **127.0.0.1** local2 ##**[**err warning info debug**]**
4. pidfile **/var**/run/haproxy**.**pid ##haproxy的pid存放路径
5. user haproxy
6. group haproxy
7. daemon ##以后台进程的方式启动服务
8. defaults
9. mode http ##默认的模式mode **{** tcp**|**http**|**health **}**
10. option dontlognull ##不记录健康检查的日志信息
11. option httpclose ##每次请求完毕后主动关闭http通道
12. option httplog ##日志类别http日志格式
13. option redispatch ##当某个服务器挂掉后强制定向到其他健康服务器
14. timeout client **300000** ##客户端连接超时，默认毫秒，也可以加时间单位
15. timeout server **300000** ##服务器连接超时
16. maxconn **3000** ##最大连接数
17. 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
19. listen websrv**-**rewrite **0.0.0.0:80**
20. balance roundrobin
21. server web1 **192.168.2.100:80** check inter **2000** rise **2** fall **5**
22. server web2 **192.168.2.200:80** check inter **2000** rise **2** fall **5**
23. #定义集群**,**listen后面的名称任意，端口为**80**
24. #balance指定调度算法为轮询（不能用简写的rr）
25. #server指定后端真实服务器，web1和web2的名称可以任意
26. #check代表健康检查，inter设定健康检查的时间间隔，rise定义成功次数，fall定义失败次数
27. listen stats **\*:1080** #监听端口
28. stats refresh 30s #统计页面自动刷新时间
29. stats uri **/**stats #统计页面url
30. stats realm Haproxy Manager #进入管理解面查看状态信息
31. stats auth admin**:**admin #统计页面用户名和密码设置

3）启动服务器并设置开机启动

1. **[**root@proxy **~]**# systemctl restart haproxy
2. **[**root@proxy **~]**# systemctl enable haproxy

**步骤三：客户端验证**

客户端配置与HAProxy相同网络的IP地址，并使用火狐浏览器访问http://192.168.4.5，测试调度器是否正常工作，客户端访问http://192.168.4.5:1080/stats测试状态监控页面是否正常。访问状态监控页的内容，参考图-5所示。

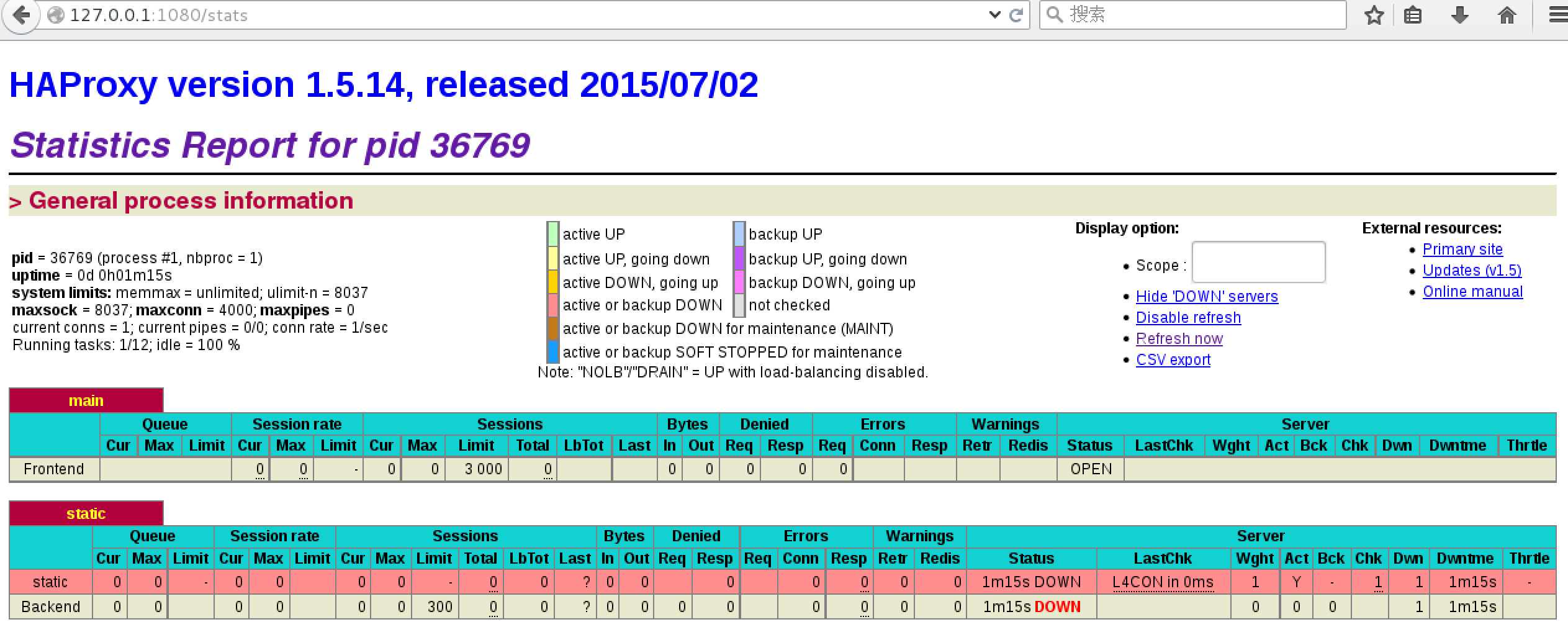


图-5

备注：

Queue队列数据的信息（当前队列数量，最大值，队列限制数量）；

Session rate每秒会话率（当前值，最大值，限制数量）；

Sessions总会话量（当前值，最大值，总量，Lbtot: total number of times a server was selected选中一台服务器所用的总时间）；

Bytes（入站、出站流量）；

Denied（拒绝请求、拒绝回应）；

Errors（错误请求、错误连接、错误回应）；

Warnings（重新尝试警告retry、重新连接redispatches）；

Server(状态、最后检查的时间（多久前执行的最后一次检查）、权重、备份服务器数量、down机服务器数量、down机时长)。