

# NSD CLUSTER DAY03

1. [案例1：Keepalived高可用服务器](#)
2. [案例3：Keepalived+LVS服务器](#)
3. [案例1：配置HAProxy负载均衡集群](#)

## 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。

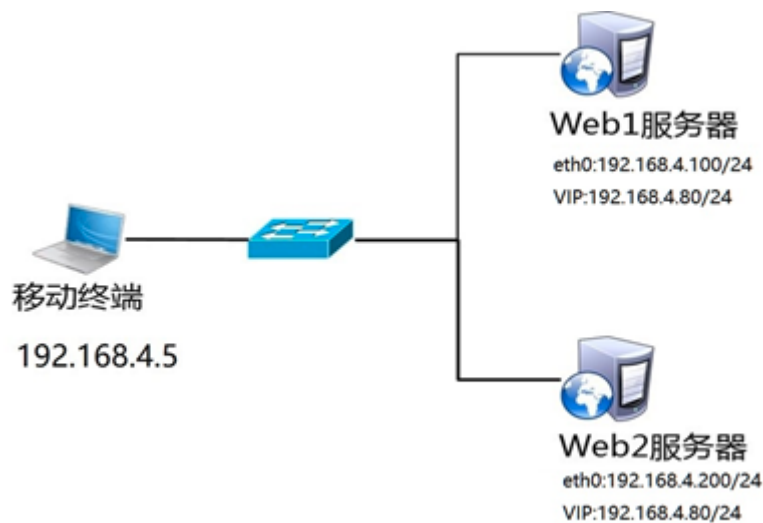


图-1

表-1

主机名	网络配置
proxy (扮演客户端的角色)	eth0:192.168.4.5
web1	eth0:192.168.4.100 VIP:192.168.4.80(keepalive 会自动配置)
web2	eth0:192.168.4.200 VIP:192.168.4.80(keepalive 会自动配置)

[Top](#)

### 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

```
01. [ root@web1 ~] # nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.100
02. [ root@web1 ~] # nmcli connection up eth0
03. [ root@web1 ~] # yum -y install httpd
04. [ root@web1 ~] # echo "192.168.4.100" > /var/www/html/index.html
05. [ root@web1 ~] # systemctl restart httpd
```

### 2) 设置Web2服务器网络参数、配置Web服务

```
01. [ root@web2 ~] # nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.200
02. [ root@web2 ~] # nmcli connection up eth0
03. [ root@web2 ~] # yum -y install httpd
04. [ root@web2 ~] # echo "192.168.4.200" > /var/www/html/index.html
05. [ root@web2 ~] # systemctl restart httpd
```

### 3) 配置proxy主机的网络参数（如果已经设置，可以忽略此步骤）

```
01. [ root@proxy ~] # nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.1
02. [ 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。**

```
01. [ root@web1 ~] # yum install -y keepalived
02. [ 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

```
01. [ root@web1 ~] # vim /etc/keepalived/keepalived.conf
```

[Top](#)

```

02. global_defs {
03.     notification_email {
04.         admin@tarena.com.cn           //设置报警收件人邮箱
05.     }
06.     notification_email_from ka@localhost //设置发件人
07.     smtp_server 127.0.0.1             //定义邮件服务器
08.     smtp_connect_timeout 30
09.     router_id web1                   //设置路由ID号 (实验需要修改)
10. }
11. vrrp_instance VI_1 {
12.     state MASTER                     //主服务器为MASTER (备服务器需要修改为BACKUP)
13.     interface eth0                  //定义网络接口
14.     virtual_router_id 50             //主备服务器VRID号必须一致
15.     priority 100                     //服务器优先级,优先级高优先获取VIP (实验需要修改)
16.     advert_int 1
17.     authentication {
18.         auth_type pass
19.         auth_pass 1111               //主备服务器密码必须一致
20.     }
21.     virtual_ipaddress { 192.168.4.80 } //谁是主服务器谁获得该VIP (实验需要修改)
22. }

```

## 2 ) 修改web2服务器Keepalived配置文件

```

01. [ root@web2 ~] # vim /etc/keepalived/keepalived.conf
02. global_defs {
03.     notification_email {
04.         admin@tarena.com.cn           //设置报警收件人邮箱
05.     }
06.     notification_email_from ka@localhost //设置发件人
07.     smtp_server 127.0.0.1             //定义邮件服务器
08.     smtp_connect_timeout 30
09.     router_id web2                   //设置路由ID号 (实验需要修改)
10. }
11. vrrp_instance VI_1 {
12.     state BACKUP                     //备服务器为BACKUP (实验需要修改)
13.     interface eth0                  //定义网络接口
14.     virtual_router_id 50             //主辅VRID号必须一致
15.     priority 50                     //服务器优先级 (实验需要修改)
16.     advert_int 1

```

[Top](#)

```

17. authentication {
18.     auth_type pass
19.     auth_pass 1111 //主辅服务器密码必须一致
20. }
21. virtual_ipaddress { 192.168.4.80 } //谁是主服务器谁配置VIP (实验需要修改)
22. }

```

### 3 ) 启动服务

```

01. [ root@web1 ~] # systemctl start keepalived
02. [ root@web2 ~] # systemctl start keepalived

```

### 4 ) 配置防火墙和SELinux

启动keepalived会自动添加一个drop的防火墙规则，需要清空！

```

01. [ root@web1 ~] # iptables - F
02. [ root@web1 ~] # setenforce 0
03. [ root@web2 ~] # iptables - F
04. [ root@web1 ~] # setenforce 0

```

## 步骤四：测试

### 1 ) 登录两台Web服务器查看VIP信息

```

01. [ root@web1 ~] # ip addr show
02. [ 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；关闭Web1服务器的网卡，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3：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

[Top](#)

- 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。

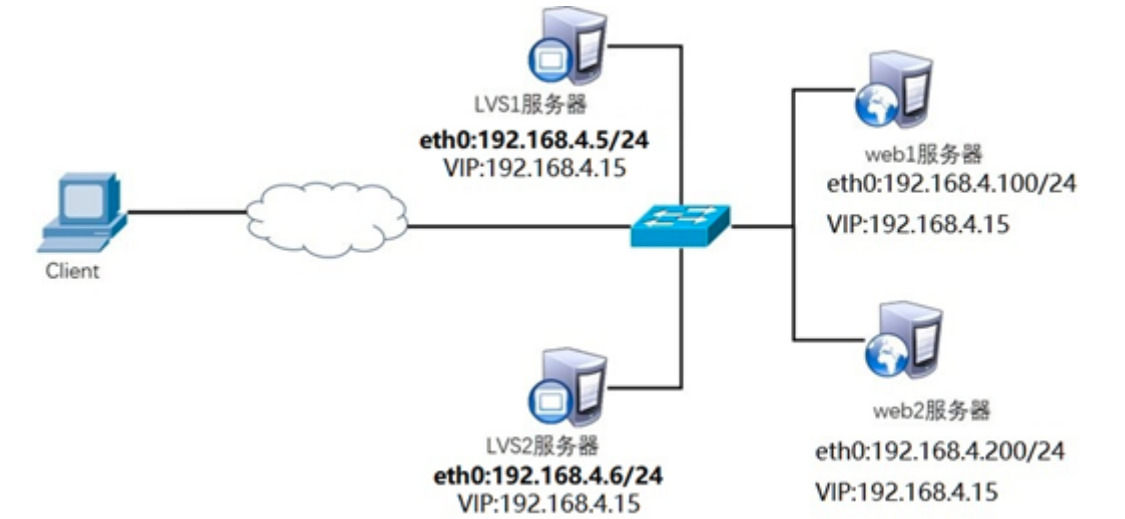


图-3

表-2

主机名	网络配置
client	eth0:192.168.4.10/24
proxy1	eth0:192.168.4.5/24
proxy2	eth0:192.168.4.6/24
web1	eth0:192.168.4.100/24
web2	eth0:192.168.4.200/24

注意：所有主机都需要配置IP地址与有效的YUM源。

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 ) 设置Web1服务器的网络参数

```
01. [ root@web1 ~] # nmcli connection modify eth0 ipv4.method manual \
02.   ipv4.addresses 192.168.4.100/24 connection.autoconnect yes
03. [ 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。

[Top](#)

```

01. [ root@web1 ~] # cd /etc/sysconfig/network-scripts/
02. [ root@web1 ~] # cp ifcfg-lo{, :0}
03. [ root@web1 ~] # vim ifcfg-lo:0
04. DEVICE=lo:0
05. IPADDR=192.168.4.15
06. NETMASK=255.255.255.255
07. NETWORK=192.168.4.15
08. BROADCAST=192.168.4.15
09. ONBOOT=yes
10. NAME=lo:0

```

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。

```

01. [ root@web1 ~] # vim /etc/sysctl.conf
02. #手动写入如下4行内容
03. net.ipv4.conf.all.arp_ignore = 1
04. net.ipv4.conf.lo.arp_ignore = 1
05. net.ipv4.conf.lo.arp_announce = 2
06. net.ipv4.conf.all.arp_announce = 2
07. #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
08. #本机不要向外宣告自己的lo回环地址是192.168.4.15

```

重启网络服务，设置防火墙与SELinux

```

01. [ root@web1 ~] # systemctl restart network
02. [ root@web1 ~] # ifconfig
03. [ root@web1 ~] # systemctl stop firewalld
04. [ root@web1 ~] # setenforce 0

```

## 2 ) 设置Web2服务器的网络参数

```

01. [ root@web2 ~] # nmcli connection modify eth0 ipv4.method manual \
02. ipv4.addresses 192.168.4.200/24 connection.autoconnect yes
03. [ root@web2 ~] # nmcli connection up eth0

```

[Top](#)

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。

```
01. [ root@web2 ~] # cd /etc/sysconfig/network-scripts/
02. [ root@web2 ~] # cp ifcfg-lo{, :0}
03. [ root@web2 ~] # vim ifcfg-lo:0
04. DEVICE=lo:0
05. IPADDR=192.168.4.15
06. NETMASK=255.255.255.255
07. NETWORK=192.168.4.15
08. BROADCAST=192.168.4.15
09. 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。

```
01. [ root@web2 ~] # vim /etc/sysctl.conf
02. #手动写入如下4行内容
03. net.ipv4.conf.all.arp_ignore = 1
04. net.ipv4.conf.lo.arp_ignore = 1
05. net.ipv4.conf.lo.arp_announce = 2
06. net.ipv4.conf.all.arp_announce = 2
07. #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
08. #本机不要向外宣告自己的lo回环地址是192.168.4.15
```

重启网络服务，设置防火墙与SELinux

```
01. [ root@web2 ~] # systemctl restart network
02. [ root@web2 ~] # ifconfig
03. [ root@web2 ~] # systemctl stop firewalld
04. [ root@web2 ~] # setenforce 0
```

3 ) 配置proxy1主机的网络参数(不配置VIP，由keepalived自动配置)

[Top](#)

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

#### 4) 配置proxy2主机的网络参数(不配置VIP, 由keepalived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。

01. [ root@proxy 2 ~] # nmcli connection modify eth0 ipv4.method manual \
02. ipv4.addresses 192.168.4.6/24 connection.autoconnect yes
03. [ root@proxy 2 ~] # 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 )

01. [ root@web1 ~] # yum -y install httpd
02. [ root@web1 ~] # echo "192.168.4.100" > /var/www/html/index.html
03. [ root@web2 ~] # yum -y install httpd
04. [ root@web2 ~] # echo "192.168.4.200" > /var/www/html/index.html

#### 3) 启动Web服务器软件(web1和web2主机)

01. [ root@web1 ~] # systemctl start httpd ; systemctl enable httpd
02. [ 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, 可用忽略此步骤 )。**

安装软件

01. [ root@proxy 1 ~] # yum install -y keepalived
02. [ root@proxy 1 ~] # systemctl enable keepalived
03. [ root@proxy 1 ~] # yum install -y ipvsadm
04. [ root@proxy 1 ~] # ipvsadm -C
- 05.
- 06.

[Top](#)



07. [ root@proxy 2 ~] # yum install - y keepalived
08. [ root@proxy 2 ~] # systemctl enable keepalived
09. [ root@proxy 2 ~] # yum install - y ipvsadm
10. [ root@proxy 2 ~] # 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

```

01. [ root@proxy 1 ~] # vim /etc/keepalived/keepalived.conf
02. global_defs {
03.     notification_email {
04.         admin@tarena.com.cn           //设置报警收件人邮箱
05.     }
06.     notification_email_from ka@localhost //设置发件人
07.     smtp_server 127.0.0.1             //定义邮件服务器
08.     smtp_connect_timeout 30
09.     router_id lvs1                   //设置路由ID号( 实验需要修改)
10. }
11. vrrp_instance VI_1 {
12.     state MASTER                     //主服务器为MASTER
13.     interface eth0                  //定义网络接口
14.     virtual_router_id 50            //主辅VRID号必须一致
15.     priority 100                    //服务器优先级
16.     advert_int 1
17.     authentication {
18.         auth_type pass
19.         auth_pass 1111              //主辅服务器密码必须一致
20.     }
21.     virtual_ipaddress { 192.168.4.15 } //配置VIP (实验需要修改)
22. }
23. virtual_server 192.168.4.15 80 {    //设置ipvsadm的VIP规则 (实验需要修改)
24.     delay_loop 6
25.     lb_algo wrr                      //设置LVS调度算法为WRR
26.     lb_kind DR                      //设置LVS的模式为DR
27.     #persistence_timeout 50
28.     #注意这样的作用是保持连接，开启后，客户端在一定时间内始终访问相同服务器
29.     protocol TCP
30.     real_server 192.168.4.100 80 {    //设置后端web服务器真实IP (实验需要修改)
31.         weight 1                    //设置权重为1
32.         TCP_CHECK {                 //对后台real_server做健康检查

```

```

33.     connect_timeout 3
34.     nb_get_retry 3
35.     delay_before_retry 3
36. }
37. }
38. real_server 192.168.4.200 80 {    //设置后端web服务器真实IP (实验需要修改)
39.     weight 2                    //设置权重为2
40.     TCP_CHECK {
41.         connect_timeout 3
42.         nb_get_retry 3
43.         delay_before_retry 3
44.     }
45. }
46. }
47. [ root@proxy 1 ~] # systemctl start keepalived
48. [ root@proxy 1 ~] # ipvsadm -Ln          #查看LVS规则
49. [ root@proxy 1 ~] # ip a s              #查看VIP配置

```

## 2 ) LVS2调度器设置Keepalived

```

01. [ root@proxy 2 ~] # vim /etc/keepalived/keepalived.conf
02. global_defs {
03.     notification_email {
04.         admin@tarena.com.cn    //设置报警收件人邮箱
05.     }
06.     notification_email_from ka@localhost //设置发件人
07.     smtp_server 127.0.0.1      //定义邮件服务器
08.     smtp_connect_timeout 30
09.     router_id lvs2            //设置路由ID号 (实验需要修改)
10. }
11. vrrp_instance VI_1 {
12.     state BACKUP              //从服务器为BACKUP (实验需要修改)
13.     interface eth0           //定义网络接口
14.     virtual_router_id 50      //主辅VRID号必须一致
15.     priority 50              //服务器优先级 (实验需要修改)
16.     advert_int 1
17.     authentication {
18.         auth_type pass
19.         auth_pass 1111        //主辅服务器密码必须一致
20.     }

```

[Top](#)

```

21.     virtual_ipaddress { 192.168.4.15 } //设置VIP (实验需要修改)
22. }
23.     virtual_server 192.168.4.15 80 { //自动设置LVS规则 (实验需要修改)
24.         delay_loop 6
25.         lb_algo wrr //设置LVS调度算法为WRR
26.         lb_kind DR //设置LVS的模式为DR
27.         # persistence_timeout 50
28.         #注意这样的作用是保持连接，开启后，客户端在一定时间内始终访问相同服务器
29.         protocol TCP
30.         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)
31.             weight 1 //设置权重为1
32.             TCP_CHECK { //对后台real_server做健康检查
33.                 connect_timeout 3
34.                 nb_get_retry 3
35.                 delay_before_retry 3
36.             }
37.         }
38.         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)
39.             weight 2 //设置权重为2
40.             TCP_CHECK {
41.                 connect_timeout 3
42.                 nb_get_retry 3
43.                 delay_before_retry 3
44.             }
45.         }
46.     [ root@proxy2 ~ ] # systemctl start keepalived
47.     [ root@proxy2 ~ ] # ipvsadm - Ln //查看LVS规则
48.     [ 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1：配置HAProxy负载均衡集群

### 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

[Top](#)

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

主机名	网络配置
client	eth0:192.168.4.10/24
proxy	eth0:192.168.4.5/24 eth1:192.168.2.5/24
web1	eth1:192.168.2.100/24
web2	eth1:192.168.2.200/24

3.3 步骤

实现此案例需要按照如下步骤进行。

注意事项：

将前面实验VIP、LVS等实验的内容清理干净！！！！！！

删除所有设备的VIP，清空所有LVS设置，关闭keepalived！！

web1关闭多余的网卡与VIP，配置本地真实IP地址。

```
01. [ root@web1 ~] # if down eth0
02. [ root@web1 ~] # if down lo:0
03. [ root@web1 ~] # nmcli connection modify eth1 ipv4.method manual \
04.   ipv4.addresses 192.168.2.100/24 connection.autoconnect yes
05. [ root@web1 ~] # nmcli connection up eth1
```

Web2关闭多余的网卡与VIP，配置本地真实IP地址。

```
01. [ root@web2 ~] # if down eth0
02. [ root@web2 ~] # if down lo:0
03. [ root@web2 ~] # nmcli connection modify eth1 ipv4.method manual \
```

[Top](#)

04. `ipv4.addresses 192.168.2.200/24 connection.autoconnect yes`
05. `[ root@web2 ~] # nmcli connection up eth1`

proxy关闭keepalived服务，清理LVS规则。

01. `[ root@proxy ~] # systemctl stop keepalived`
02. `[ root@proxy ~] # systemctl disable keepalived`
03. `[ root@proxy ~] # ipvsadm - C`
- 04.
05. `[ root@proxy ~] # nmcli connection modify eth0 ipv4.method manual \`
06. `ipv4.addresses 192.168.4.5/24 connection.autoconnect yes`
07. `[ root@proxy ~] # nmcli connection up eth0`
- 08.
09. `[ root@proxy ~] # nmcli connection modify eth1 ipv4.method manual \`
10. `ipv4.addresses 192.168.2.5/24 connection.autoconnect yes`
11. `[ 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，可用忽略此步骤）

01. `[ root@web1 ~] # yum -y install httpd`
02. `[ root@web1 ~] # systemctl start httpd`
03. `[ root@web1 ~] # echo "192.168.2.100" > /var/www/html/index.html`
- 04.
05. `[ root@web2 ~] # yum -y install httpd`
06. `[ root@web2 ~] # systemctl start httpd`
07. `[ 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

01. `[ root@haproxy ~] # echo 'net.ipv4.ip_forward = 1' >> sysctl.conf //开启路由转发`
02. `[ root@haproxy ~] # sysctl - p`
03. `[ root@haproxy ~] # yum -y install haproxy`

[Top](#)

2) 修改配置文件

```

01. [ root@haproxy ~] # vim /etc/haproxy/haproxy.cfg
02. global
03.     log 127.0.0.1 local2 ##### [ err warning info debug]
04.     chroot /usr/local/haproxy
05.     pidfile /var/run/haproxy.pid #####haproxy的pid存放路径
06.     maxconn 4000 #####最大连接数，默认4000
07.     user haproxy
08.     group haproxy
09.     daemon #####创建1个进程进入daemon模式运行
10. defaults
11.     mode http #####默认的模式mode { tcp| http| health } log global #####采用全局定义的日志
12.     option dontlognull #####不记录健康检查的日志信息
13.     option httpclose #####每次请求完毕后主动关闭http通道
14.     option httplog #####日志类别http日志格式
15.     option forwardfor #####后端服务器可以从Http Header中获得客户端ip
16.     option redispatch #####serverid服务器挂掉后强制定向到其他健康服务器
17.     timeout connect 10000 #如果backend没有指定，默认为10s
18.     timeout client 300000 #####客户端连接超时
19.     timeout server 300000 #####服务器连接超时
20.     maxconn 60000 #####最大连接数
21.     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
22. listen stats
23.     bind 0.0.0.0:1080 #监听端口
24.     stats refresh 30s #统计页面自动刷新时间
25.     stats uri /stats #统计页面url
26.     stats realm Haproxy Manager #统计页面密码框上提示文本
27.     stats auth admin:admin #统计页面用户名和密码设置
28.     #stats hide-version #隐藏统计页面上HAProxy的版本信息
29. listen webserv - rewrite 0.0.0.0:80
30.     balance roundrobin
31.     server web1 192.168.2.100:80 check inter 2000 rise 2 fall 5
32.     server web2 192.168.2.200:80 check inter 2000 rise 2 fall 5

```

### 3) 启动服务器并设置开机启动

```

01. [ root@haproxy ~] # systemctl start haproxy
02. [ root@haproxy ~] # systemctl enable haproxy

```

[Top](#)

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-4所示。

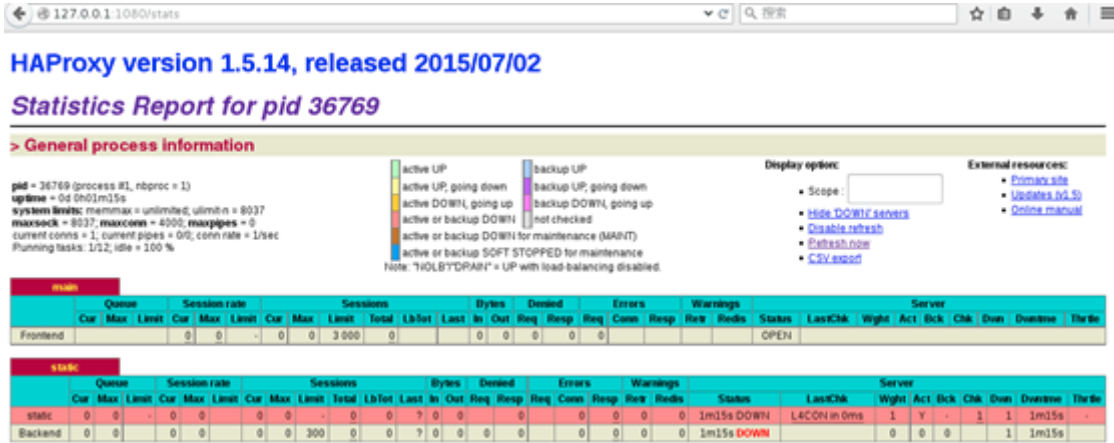


图-4

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redispaches）；

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)。

Top