

Diagram illustrating a network setup for a Client (R1) connecting to a Load Balancer (R2) and a Backend (R3, R4, R5, R6, R7).

The Client (R1) is connected to the Load Balancer (R2) via IP 192.168.12.x. The Load Balancer (R2) is connected to the Backend (R3, R4, R5, R6, R7) via IP 192.168.23.x, 192.168.34.x, and 192.168.35.x. The Backend (R3, R4, R5, R6, R7) is connected to the Client (R1) via IP 192.168.50.x.

The diagram is divided into two sections by a vertical line: "front" (Client and Load Balancer) and "back" (Backend servers). The Backend servers are labeled "http nginx".

Server1	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.12.1 PREFIX=24 ONBOOT=yes
<pre> 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:32:d3:6d brd ff:ff:ff:ff:ff:ff inet 192.168.12.1/24 brd 192.168.12.255 scope global noprefixroute enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe32:d36d/64 scope link valid_lft forever preferred_lft forever </pre>	
Server2	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.12.2 PREFIX=24 ONBOOT=yes
	enp0s9
	DEVICE=enp0s9 IPADDR=192.168.23.2 PREFIX=24 ONBOOT=yes

```

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:00:cc:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.12.2/24 brd 192.168.12.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe00:cclc/64 scope link
        valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:03:45:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.23.2/24 brd 192.168.23.255 scope global noprefixroute enp0s9
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe03:4504/64 scope link
        valid_lft forever preferred_lft forever

```

Server3	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.34.3 PREFIX=24 ONBOOT=yes
	enp0s9
	DEVICE=enp0s9 IPADDR=192.168.23.3 PREFIX=24 ONBOOT=yes
	enp0s10
	DEVICE=enp0s10 IPADDR=192.168.35.3 PREFIX=24 ONBOOT=yes
<pre> 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:e5:3d:ba brd ff:ff:ff:ff:ff:ff inet 192.168.34.3/24 brd 192.168.34.255 scope global noprefixroute enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fee5:3dba/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:64:7e:37 brd ff:ff:ff:ff:ff:ff inet 192.168.23.3/24 brd 192.168.23.255 scope global noprefixroute enp0s9 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe64:7e37/64 scope link valid_lft forever preferred_lft forever 5: enp0s10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:6c:ec:6d brd ff:ff:ff:ff:ff:ff inet 192.168.35.3/24 brd 192.168.35.255 scope global noprefixroute enp0s10 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe6c:ec6d/64 scope link valid_lft forever preferred_lft forever </pre>	
Server4	enp0s8

ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.34.4 PREFIX=24 ONBOOT=yes
	enp0s9
	DEVICE=enp0s9 IPADDR=192.168.50.4 PREFIX=24 ONBOOT=yes
<pre> 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:10:f6:f8 brd ff:ff:ff:ff:ff:ff inet 192.168.34.4/24 brd 192.168.34.255 scope global noprefixroute enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe10:f6f8/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:a2:52:ac brd ff:ff:ff:ff:ff:ff inet 192.168.50.4/24 brd 192.168.50.255 scope global noprefixroute enp0s9 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fea2:52ac/64 scope link tentative valid_lft forever preferred_lft forever </pre>	
Server5	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.35.5 PREFIX=24 ONBOOT=yes
	enp0s9

	DEVICE=enp0s9 IPADDR=192.168.50.5 PREFIX=24 ONBOOT=yes
<pre> 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:f8:54:05 brd ff:ff:ff:ff:ff:ff inet 192.168.35.5/24 brd 192.168.35.255 scope global noprefixroute enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe80:5405/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:dd:00:3b brd ff:ff:ff:ff:ff:ff inet 192.168.50.5/24 brd 192.168.50.255 scope global noprefixroute enp0s9 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fedd:3b/64 scope link valid_lft forever preferred_lft forever </pre>	

Server6	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8 UUID=1fab61aa-c80a-4c33-bd28-efd19ea5844b DEVICE=enp0s8 ONBOOT=no	DEVICE=enp0s8 IPADDR=192.168.50.6 PREFIX=24 ONBOOT=yes
<pre> 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:ed:b8:37 brd ff:ff:ff:ff:ff:ff inet 192.168.50.6/24 brd 192.168.50.255 scope global noprefixroute enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:feed:b837/64 scope link tentative valid_lft forever preferred_lft forever </pre>	

Server7	enp0s8
ДО	ПОСЛЕ
TYPE=Ethernet PROXY_METHOD=none BROWSER_ONLY=no BOOTPROTO=dhcp DEFROUTE=yes IPV4_FAILURE_FATAL=no IPV6INIT=yes IPV6_AUTOCONF=yes IPV6_DEFROUTE=yes IPV6_FAILURE_FATAL=no IPV6_ADDR_GEN_MODE=stable-privacy NAME=enp0s8	DEVICE=enp0s8 IPADDR=192.168.50.7 PREFIX=24 ONBOOT=yes

UUID=1fab61aa-c80a-4c33-bd28-efdf19ea5844b DEVICE=enp0s8 ONBOOT=no	
--	--

```

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:64:82:f6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.7/24 brd 192.168.50.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe64:82f6/64 scope link
        valid_lft forever preferred_lft forever

```

2. Создал dummy3 интерфейс на машинах Server2, Server4 и Server5

#Выполнил настройку всех трёх серверов абсолютно одинаково, как внизу (Ctrl+C, Ctrl+D)

DUMMY

Создал dummy интерфейс

```
ip link add dummy3 type dummy
```

Присвоил ip

```
ip addr add 10.1.1.1/32 dev dummy3
```

Включил dummy

```
ip link set up dev dummy3
```

```

6: dummy3: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether 9e:b6:73:be:9f:8a brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.1/32 scope global dummy3
        valid_lft forever preferred_lft forever
    inet6 fe80::9cb6:73ff:febe:9f8a/64 scope link
        valid_lft forever preferred_lft forever

```

Настроил dummy3

```
vim /etc/sysconfig/network-scripts/ifcfg-dummy3
```

```

DEVICE=dummy3
ONBOOT=yes
IPADDR=10.1.1.1
PREFIX=32
TYPE=dummy
NM_CONTROLLED=no

```

Перезагрузил dummy

```
ifdown dummy3
```

```
ifup dummy3
```

3. Настроил связность компонентов сети

Server1

FRR

Установил FRR в соответствии с документацией

```

# possible values for FRRVER: frr-6 frr-7 frr-8 frr-stable
# frr-stable will be the latest official stable release

```

```
FRRVER="frr-stable"
```

```
# add RPM repository on CentOS 7
```

```
curl -O https://rpm.frrouting.org/repo/\$FRRVER-repo-1-0.el7.noarch.rpm
```

```
sudo yum install ./$FRRVER*
```

```
# install FRR
```

```
sudo yum install frr frr-pythontools
```

Активирую демона ospfd

```
vim /etc/frr/daemons
```

В значении ospfd заменил значение с no на yes

Для активации новых настроек перезагрузил frr

```
systemctl restart frr
```

Проверил статус frr

```
systemctl status frr
```

```
● frr.service - FRRouting
   Loaded: loaded (/usr/lib/systemd/system/frr.service; disabled; vendor preset: disabled)
   Active: active (running) since Пт 2022-09-02 16:59:32 MSK; 6s ago
     Docs: https://frrouting.readthedocs.io/en/latest/setup.html
```

Добавил сервис frr в автозагрузку, чтобы оно работало после перезагрузки

```
systemctl enable frr
```

Настроил маршруты ospf:

Переключился в консоль vtysh

```
vttysh
```

Сделал вывод сетей на экран, чтобы видеть сети и мог копировать в дальнейшей настройке

```
sh ip route
```

```
configure terminal
```

```
router ospf
```

Анонсирую сеть

```
network 192.168.12.0/24 area 0
```

ех из уровня config terminal и далее из config

Записал в файл

```
w
```

Проверил сети, которые анонсирую

```
sh running-config
```

```
router ospf
 network 192.168.12.0/24 area 0
exit
```

Firewalld

Отключил firewall, проверил статус и удалил из автозагрузки, чтобы он уже никогда не включился сам. Статья: <https://losst.ru/kak-otklyuchit-firewall-centos-7>
systemctl stop firewalld
systemctl status firewalld
systemctl disable firewalld

Server2

FRR

Установил и настроил frr по аналогии с Server1

Анонсирую сеть

network 10.1.1.1/32 area 0

network 192.168.12.0/24 area 0

network 192.168.23.0 area 0

ex из уровня config terminal и далее из config

Записал в файл

w

Проверил сети, которые анонсирую

sh running-config

```
router ospf
 network 10.1.1.1/32 area 0
 network 192.168.12.0/24 area 0
 network 192.168.23.0/24 area 0
 exit
```

Firewalld

Отключил firewalld по аналогии с Server1

Включил forward

sysctl -w net.ipv4.ip_forward=1 *#-w позволяет записать значение, чтобы оно не изменилось после перезагрузки*

Server3

FRR

Установил и настроил frr по аналогии с Server1

Анонсирую сеть

network 192.168.23.0/24 area 0

network 192.168.34.0/24 area 0

network 192.168.35.0/24 area 0

ex из уровня config terminal и далее из config

Записал в файл

w

Проверил сети, которые анонсирую
sh running-config

```
router ospf
 network 192.168.23.0/24 area 0
 network 192.168.34.0/24 area 0
 network 192.168.35.0/24 area 0
```

Firewalld

Отключил firewalld по аналогии с Server1

Включил forward

sysctl -w net.ipv4.ip_forward=1 *#-w позволяет записать значение, чтобы оно не изменилось после перезагрузки*

Server4

FRR

Установил и настроил frr по аналогии с Server1

Анонсирую сеть

network 192.168.34.0/24 area 0

ex из уровня config terminal и далее из config

Записал в файл

w

Проверил сети, которые анонсирую

sh running-config

network 192.168.34.0/24 area 0

```
router ospf
 network 192.168.34.0/24 area 0
```

Firewalld

Отключил firewalld по аналогии с Server1

Включил forward

sysctl -w net.ipv4.ip_forward=1 *#-w позволяет записать значение, чтобы оно не изменилось после перезагрузки*

Server5

FRR

Установил и настроил frr по аналогии с Server1

Анонсирую сеть

network 192.168.35.0/24 area 0

ех из уровня config terminal и далее из config

Записал в файл
w

Проверил сети, которые анонсирую
sh running-config

network 192.168.34.0/24 area 0

```
router ospf
 network 192.168.35.0/24 area 0
exit
```

Firewalld

Отключил firewalld по аналогии с Server1

Включил forward

sysctl -w net.ipv4.ip_forward=1 *#-w позволяет записать значение, чтобы оно не изменилось после перезагрузки*

Server6

Firewalld

Отключил firewall, проверил статус и удалил из автозагрузки, чтобы он уже никогда не включился сам. Статья: <https://losst.ru/kak-otklyuchit-firewall-centos-7>
systemctl stop firewalld
systemctl status firewalld
systemctl disable firewalld

Server7

Firewalld

Отключил firewall, проверил статус и удалил из автозагрузки, чтобы он уже никогда не включился сам. Статья: <https://losst.ru/kak-otklyuchit-firewall-centos-7>
systemctl stop firewalld
systemctl status firewalld
systemctl disable firewalld

Чтобы выполнить задачу, нужно установить компоненты. Рис.2

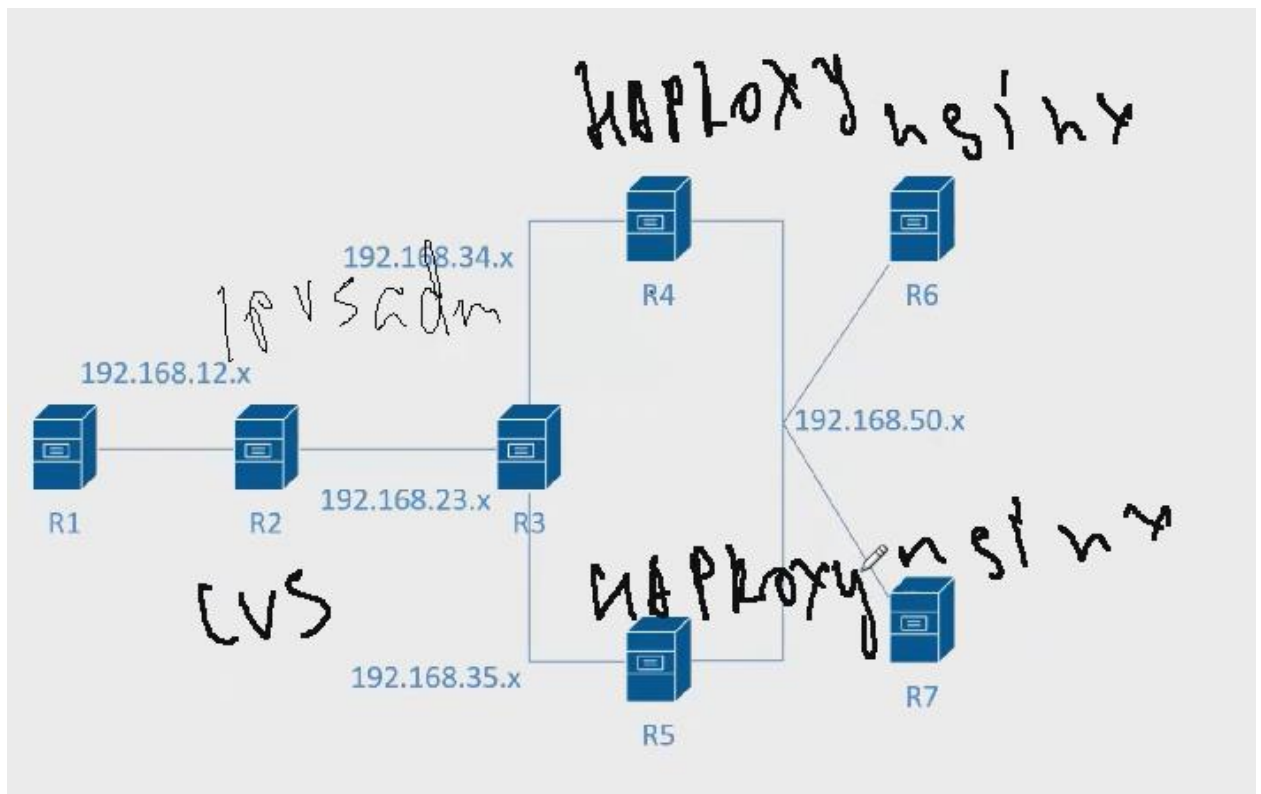


Рис.2 Утилиты на серверах

3. Установка и настройка компонентов

Server2

IPVSADM

Установил ipvsadm
yum install ipvsadm -y

Открываю мануал ipvsadm, чтобы найти примеры и скопировать их
man ipvsadm

EXAMPLE

```
ipvsadm -A -t 207.175.44.110: 80 -s rr
ipvsadm -a -t 207.175.44.110: 80 -r 192.168.10.1: 80 -m
ipvsadm -a -t 207.175.44.110: 80 -r 192.168.10.1: 80 -m
ipvsadm -a -t 207.175.44.110: 80 -r 192.168.10.3: 80 -m
ipvsadm -a -t 207.175.44.110:80 -r 192.168.10.4:80 -m
ipvsadm -a -t 207.175.44.110:80 -r 192.168.10.5:80 -m
```

Вышел из man. Прописал следующие правила

ДО ПРАВКИ

ПОСЛЕ

```
ipvsadm -A -t 207.175.44.110:80 -s rr
ipvsadm -a -t 207.175.44.110:80 -r
192.168.10.1:80 -m
ipvsadm -a -t 207.175.44.110:80 -r
192.168.10.1:80 -m
```

```
ipvsadm -A -t 10.1.1.1:80 -s rr
ipvsadm -a -t 10.1.1.1:80 -r
192.168.34.4:80 -i
#ip dummy, потом ip Server4
# Ключ -m – это работа по NAT.
Заменил на туннель -i
ipvsadm -a -t 10.1.1.1:80 -r
192.168.35.5:80 -i
```

Проверил настройки

ipvsadm -Ln

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  10.1.1.1:80 rr
  -> 192.168.34.4:80              Tunnel  1      0          0
  -> 192.168.35.5:80              Tunnel  1      0          0
```

Далее надо настроить интерфейсы на Server4 и Server5.

URPF

Исключительно в учебных целях надо отключить службу URPF. Она сверяет с какого интерфейса пришел трафик и куда его надо отправить. Защита от DDOS.

Проверил состояние URPF. В Linux она называется rp_filter

sysctl -a | grep rp_filter

Включен на нужных интерфейсах

Отключил URPF

sysctl -w net.ipv4.conf.all.rp_filter=0

sysctl -w net.ipv4.conf.dummy0.rp_filter=0

sysctl -w net.ipv4.conf.dummy3.rp_filter=0

sysctl -w net.ipv4.conf.enp0s8.rp_filter=0

sysctl -w net.ipv4.conf.enp0s9.rp_filter=0

Server3

URPF

Исключительно в учебных целях надо отключить службу URPF. Она сверяет с какого интерфейса пришел трафик и куда его надо отправить. Защита от DDOS.

Проверил состояние URPF. В Linux она называется rp_filter

```
sysctl -a | grep rp_filter
```

Включен на нужных интерфейсах

Отключил URPF

```
sysctl -w net.ipv4.conf.all.rp_filter=0
```

```
sysctl -w net.ipv4.conf.enp0s8.rp_filter=0
```

```
sysctl -w net.ipv4.conf.enp0s9.rp_filter=0
```

```
sysctl -w net.ipv4.conf.enp0s10.rp_filter=0
```

Server4

IPVSADM

Чтобы получить туннель от Server2 загрузил модуль ipip

```
modprobe ipip
```

```
valid_lif forever preferred_lif forever
7: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
```

#Если хочу, чтобы интерфейс пережил перезагрузку системы, то надо создать файл ifsfgr-tun10 и записать в нём: 'ONBOOT=yes'. Остальные настройки не надо, т.к. туннельный интерфейс сам все настроит.

Включил интерфейс tun10

```
ip link set up tun10
```

```
valid_lif forever preferred_lif forever
7: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1480 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
```

Нароку

Проверил версию, доступную напрямую через CentOS

```
yum info haproxy
```

```

* updates: ftp.nsc.1
Доступные пакеты
Название: haproxy
Архитектура: x86_64
Версия: 1.5.18
Выпуск: 9.el7_9.1
Объем: 835 k
Источник: updates/7/

```

Для учебы подойдет, но в продакшен необходимо ставить свежую версию.
 На сайте <https://www.haproxy.com/> лежит последняя платная версия для уровня Enterprise
 На сайте <https://www.haproxy.org/> лежит последняя бесплатная версия

Установил haproxy
 yum install haproxy -y

Пример конфига HAproxу находится в файле
 less /usr/share/doc/haproxy-1.5.18/examples/examplex.cfg

Настроил haproxy.cfg
 vim /etc/haproxy/haproxy.cfg

Задал сервера и порты между которыми надо балансировать нагрузку

ДО	ПОСЛЕ
<pre> frontend main *:5000 acl url_static path_beg acl url_static path_end use_backend static if default_backend app backend app balance roundrobin server app1 127.0.0.1:5001 check server app2 127.0.0.1:5002 check server app3 127.0.0.1:5003 check server app4 127.0.0.1:5004 check </pre>	<pre> frontend main 10.1.1.1:80 acl url_static pa acl url_static pa use_backend static default_backend backend app balance roundrobin server app1 192.168.50.6:80 check server app2 192.168.50.7:80 check </pre>

Перезагрузил HAproxу
 systemctl restart haproxy
 systemctl status haproxy

```

● haproxy.service - HAProxy Load Bal
   Loaded: loaded (/usr/lib/systemd
   Active: active (running) since П
   Main PID: 17861 (haproxy-systemd)
   CGroup: /system.slice/haproxy.se

```

Проверил, балансировку нагрузки
 curl <http://10.1.1.1>

```
[root@localhost ~]# curl http://10.1.1.1
'Hello from Server7'
[root@localhost ~]# curl http://10.1.1.1
'Hello from Server6'
[root@localhost ~]# curl http://10.1.1.1
'Hello from Server7'
[root@localhost ~]# curl http://10.1.1.1
'Hello from Server6'
```

Запросы доходят. Все балансится.

Контрольно проверил логи Server6 и Server7

less /var/log/nginx/access.log

```
192.168.50.4 - - [02/Sep/2022:19:05:35 +0300] "GET / HTTP/1.1" 200 25 "-" "curl/7.29.0" "10.1.1.1"
192.168.50.4 - - [02/Sep/2022:19:05:36 +0300] "GET / HTTP/1.1" 200 25 "-" "curl/7.29.0" "10.1.1.1"
192.168.50.4 - - [02/Sep/2022:19:05:41 +0300] "GET / HTTP/1.1" 200 25 "-" "curl/7.29.0" "10.1.1.1"
192.168.50.4 - - [02/Sep/2022:19:05:43 +0300] "GET / HTTP/1.1" 200 25 "-" "curl/7.29.0" "10.1.1.1"
/var/log/nginx/access.log (END)
```

Все ок

URPF

Исключительно в учебных целях надо отключить службу URPF. Она сверяет с какого интерфейса пришел трафик и куда его надо отправить. Защита от DDOS.

Проверил состояние URPF. В Linux она называется rp_filter

sysctl -a | grep rp_filter

```
[root@localhost ~]# sysctl -a | grep
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.dummy0.rp_filter = 0
net.ipv4.conf.dummy0.rp_filter = 1
net.ipv4.conf.dummy3.rp_filter = 0
net.ipv4.conf.dummy3.rp_filter = 1
net.ipv4.conf.enp0s3.rp_filter = 0
net.ipv4.conf.enp0s3.rp_filter = 1
net.ipv4.conf.enp0s8.rp_filter = 0
net.ipv4.conf.enp0s8.rp_filter = 1
net.ipv4.conf.enp0s9.rp_filter = 0
net.ipv4.conf.enp0s9.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 0
sysctl: reading key "net.ipv6.conf.all.rp_filter"
sysctl: reading key "net.ipv6.conf.default.rp_filter"
sysctl: reading key "net.ipv6.conf.default.rp_filter"
sysctl: reading key "net.ipv6.conf.default.rp_filter"
sysctl: reading key "net.ipv6.conf.default.rp_filter"
net.ipv4.conf.lo.rp_filter = 0
net.ipv4.conf.tunl0.rp_filter = 0
net.ipv4.conf.tunl0.rp_filter = 1
sysctl: reading key "net.ipv6.conf.default.rp_filter"
```

Включен на нужных интерфейсах

```
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.dummy0.rp_filter = 1
net.ipv4.conf.dummy3.rp_filter = 1
```

```
net.ipv4.conf.enp0s8.rp_filter = 1
net.ipv4.conf.tunl0.rp_filter = 1
```

Отключил URPF

```
sysctl -w net.ipv4.conf.all.rp_filter=0
sysctl -w net.ipv4.conf.dummy0.rp_filter=0
sysctl -w net.ipv4.conf.dummy3.rp_filter=0
sysctl -w net.ipv4.conf.enp0s8.rp_filter=0
sysctl -w net.ipv4.conf.tunl0.rp_filter=0
```

Server5

IPVSADM

Чтобы получить туннель от Server2 загрузил модуль ipip
modprobe ipip

```
valid_irt forever preferred_irt forever
7: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
```

#Если хочу, чтобы интерфейс пережил перезагрузку системы, то надо создать файл ifsg-tun10 и записать в нём: 'ONBOOT=yes'. Остальные настройки не надо, т.к. туннельный интерфейс сам все настроит.

Включил интерфейс tun10

```
ip link set up tun10
```

```
valid_irt forever preferred_irt forever
7: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1480 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
```

HAPROXY

Установил, настроил haproxy.cfg также, как и на Server4

URPF

Исключительно в учебных целях надо отключить службу URPF. Она сверяет с какого интерфейса пришел трафик и куда его надо отправить. Защита от DDOS.

Проверил состояние URPF. В Linux она называется rp_filter
sysctl -a | grep rp_filter

Включен на нужных интерфейсах

Отключил URPF

```
sysctl -w net.ipv4.conf.all.rp_filter=0
```

```
sysctl -w net.ipv4.conf.dummy0.rp_filter=0
```

```
sysctl -w net.ipv4.conf.dummy3.rp_filter=0
```

```
sysctl -w net.ipv4.conf.enp0s8.rp_filter=0
```

```
sysctl -w net.ipv4.conf.enp0s9.rp_filter=0
```

```
sysctl -w net.ipv4.conf.tunl0.rp_filter=0
```

Server6

NGINX

Установил epel-release
yum install epel-release -y

Установил nginx
yum install nginx -y

Открыл настройки nginx, чтобы посмотреть, где лежит файл, в котором заменяю вывод файла index.html на 'Hello from Server6'
vim /etc/nginx/nginx.conf

```
server {  
    listen      80;  
    listen      [::]:80;  
    server_name _;  
    root        /usr/share/nginx/html;  
  
    # Load configuration files for the default server block.  
    include     /etc/nginx/default.d/*.conf;  
}
```

Сменил директорию
cd /usr/share/nginx/html

ll -al

```
drwxr-xr-x. 2 root root   27 сен  2 16:14 icons  
lrwxrwxrwx. 1 root root   18 сен  2 16:14 img -> ../../doc/HTML/img  
lrwxrwxrwx. 1 root root   25 сен  2 16:14 index.html -> ../../doc/HTML/index.html  
-rw-r--r--. 1 root root 368 окт 19 2021 nginx-logo.png  
lrwxrwxrwx. 1 root root   14 сен  2 16:14 poweredby.png -> nginx-logo.png
```

echo 'Hello from Server6'>index.html
cat index.html

```
[root@localhost html]# cat index.html  
'Hello from Server6'
```

Перезапустил nginx

```
[root@localhost html]# systemctl restart nginx  
[root@localhost html]# systemctl sstatus nginx  
Unknown operation 'sstatus'.  
[root@localhost html]# systemctl status nginx  
● nginx.service - The nginx HTTP and reverse proxy server  
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)  
   Active: active (running) since Пт 2022-09-02 16:30:36 MSK; 12s ago  
 Process: 16847 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)  
 Process: 16845 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)  
 Process: 16844 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)  
 Main PID: 16849 (nginx)  
   CGroup: /system.slice/nginx.service  
           └─16849 nginx: master process /usr/sbin/nginx  
             └─16851 nginx: worker process
```

systemctl enable nginx

Server7

NGINX

Установил по аналогии с Server6

4. ГОТОВО

mRemoteNG - conf(Cons.xml) - Server1

Подключения:

- Server1
- Server2
- Server3
- Server4
- Server5
- Server6
- Server7

Подключение:

Process: 844 ExecStart=/usr/sbin/firewallld --nofork --nopid \$FIREWALLD_ARGS (code=exited, status=0/SUCCESS)

Main PID: 844 (code=exited, status=0/SUCCESS)

```

сен 02 15:12:22 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
сен 02 15:12:24 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
сен 02 15:12:25 localhost.localdomain firewallld[844]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. ... it now.
сен 02 17:35:08 localhost.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
сен 02 17:35:09 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@localhost ~]# curl http://10.1.1.1
curl: (6) Could not resolve host: http; Неизвестная ошибка
[root@localhost ~]# curl http://10.1.1.1
^C
[root@localhost ~]# curl http://10.1.1.1
^C
[root@localhost ~]# curl http://10.1.1.1
Hello from Server6'
[root@localhost ~]# curl http://10.1.1.1
Hello from Server7'
[[Acurl: (6) Could not resolve host: http; Неизвестная ошибка
[root@localhost ~]# curl http://10.1.1.1
curl: (6) Could not resolve host: http; Неизвестная ошибка
[root@localhost ~]# curl http://10.1.1.1
curl: (6) Could not resolve host: http; Неизвестная ошибка
[root@localhost ~]# curl http://10.1.1.1
curl: (6) Could not resolve host: http; Неизвестная ошибка
[root@localhost ~]# curl http://10.1.1.1
Hello from Server7'
[root@localhost ~]# curl http://10.1.1.1
Hello from Server6'
[root@localhost ~]# curl http://10.1.1.1
Protocol
Port 22
SSH version 2
Session RUTTY Default Settings
Разное
Внеш. Инстр. Д
Внеш. Инстр. П
Протокол
Выберите протокол, который должен использоваться для подключения.
```

Конфигурация

Имя	Описание	Значок
Server5		
mRemoteNG General		
Имя вкладки		
Имя хоста / IP	192.168.1.70	
Пользователь		
Пароль		
Протокол	SSH version 2	
Порт	22	
Сессия RUTTY	Default Settings	
Разное		
Внеш. Инстр. Д		
Внеш. Инстр. П		
Протокол		
Выберите протокол, который должен использоваться для подключения.		

Уведомления

21:48
02.09.2022