

Выполнил студент группы 2345 Романенко Кирилл

1. На сервере Server1 установить ipvsadm.

Установил

```
yum install -y ipvsadm
```

Проверил работоспособность

```
ipvsadm -l
```

```
[root@server1 kdromanenko]# ipvsadm -l
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
```

2. На сервере Server1 установить docker и запустить 2 контейнера с Nginx.

Установил docker

```
yum install docker
```

```
systemctl start docker
```

Запустил два контейнера с Nginx

```
mkdir /srv/A /srv/B
```

Контейнер 1

```
echo "This is A" > /srv/A/index.html
```

```
docker run --rm -d -v "/srv/A:/usr/share/nginx/html" --name nginx-A nginx
```

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' nginx-A 172.17.0.2
```

Контейнер 2

```
echo "This is B" > /srv/B/index.html
```

```
docker run --rm -d -v "/srv/B:/usr/share/nginx/html" --name nginx-B nginx
```

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' nginx-B 172.17.0.3
```

Проверил, что могу запросить index.html у каждого из контейнеров

```
curl 172.17.0.2
```

В выводе есть forbidden. Надо поправить selinux-контекст

```
ls -laZ /srv/*/index.html
```

```
semanage fcontext -a -t httpd_sys_content_t /srv/A/index.html
```

```
semanage fcontext -a -t httpd_sys_content_t /srv/B/index.html
```

```
restorecon -v /srv/A/index.html
```

```
restorecon -v /srv/B/index.html
```

```
ls -laZ /srv/*/index.html
```

Проверил еще раз

```
curl 172.17.0.2
```

```
This is A
```

```
curl 172.17.0.3
```

```
This is B
```

3. Создать интерфейс dummy2 и назначить ему IP-адрес 111.111.111.111/32.
4. Проанонсировать этот IP в сеть из трёх серверов.

Создал dummy

```
ip link add dummy2 type dummy
```

Присвоил ip

```
ip addr add 111.111.111.111/32 dev dummy2
```

Включил dummy

```
ip link set up dev dummy2
```

Настроил автоматическое поднятие модуля при запуске машины

```
vim /etc/modules dummy
```

Настроил dummy2

```
vim /etc/sysconfig/network-scripts/ifcfg-dummy2
```

```
DEVICE=dummy2
NAME=dummy2
ONBOOT=yes
IPADDR=111.111.111.111
PREFIX=32
NETMASK=255.255.255.255
MACADDR=02:22:22:ff:ff:ff
TYPE=dummy
NM_CONTROLLED=no
```

Перезагрузил dummy

```
ifdown dummy2
```

Проанонсировал маршрут в сеть из трех серверов

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
```

```
network 111.111.111.111/32 area 0
```

5. Настроить ipvs для передачи http запросов с сервера server3 в оба контейнера в Nginx используя механизм балансировки round-robin.

6. Удостовериться, что запросы балансируются между контейнерами путем проверки access.log внутри каждого из них.

Узнал ip того интерфейса, который смотрит во внутреннюю сеть, где nginx.

```
ip a
```

```
enp0s8:      192.168.23.3/24
```

Создал новый IPVS-сервис

```
ipvsadm -A -t 192.168.23.3:80 -s rr
```

```
ipvsadm -l -n
```

```
ipvsadm -a -t 192.168.23.3:80 -r 172.17.0.2 -m
```

```
ipvsadm -a -t 192.168.23.3:80 -r 172.17.0.3 -m
```

```
ipvsadm -l -n
```

Открыл порт 80 на firewall

```
firewall-cmd --zone=public --add-service=http --permanent success
```

```
firewall-cmd --zone=public --add-service=http success
```

Проверил работоспособность на 1000 запросов

```
for i in `seq 1 1000`; do curl http://192.168.1.41 -s; done | sort | uniq -c
```

Вывод:

```
500 This is A
```

```
500 This is B
```