

Устанавливаем утилиту ipvsadm

```
[root@R1 ~]# yum install -y ipvsadm Dependencies Resolved
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink
 * base: mirror.docker.ru
 * epel: ftp.lysator.liu.se
 * extras: mirror.axelname.ru
 * updates: mirror.axelname.ru
base
epel
extras
frr
frr-extras
updates
(1/3): epel/x86_64/group_gz
(2/3): epel/x86_64/updateinfo
(3/3): epel/x86_64/primary_db
No package Dependencies available.
No package Resolved available.
Resolving Dependencies
--> Running transaction check
---> Package ipvsadm.x86_64 0:1.27-8.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

Устанавливаем докер

```
[root@R1 ~]# yum install docker
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.docker.ru
 * epel: ftp.lysator.liu.se
 * extras: mirror.axelname.ru
 * updates: mirror.axelname.ru
Resolving Dependencies
--> Running transaction check
---> Package docker.x86_64 2:1.13.1-209.git7d71120.el7.centos will be installed
--> Processing Dependency: docker-common = 2:1.13.1-209.git7d71120.el7.centos for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: docker-client = 2:1.13.1-209.git7d71120.el7.centos for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: subscription-manager-rhsm-certificates for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: libseccomp.so.2()(64bit) for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Running transaction check
---> Package docker-client.x86_64 2:1.13.1-209.git7d71120.el7.centos will be installed
```

Запускаем докер, включаем в автозагрузку, проверяем статус

```
Completed.  
[root@R1 ~]# systemctl start docker  
[root@R1 ~]# systemctl enable docker  
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.  
[root@R1 ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)  
   Active: active (running) since Wed 2022-08-31 17:14:50 MSK; 14s ago  
     Docs: http://docs.docker.com  
   Main PID: 4693 (dockerd-current)  
   CGroup: /system.slice/docker.service  
           └─4693 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libexec/docker/docker-  
             └─4700 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/doc  
  
Aug 31 17:14:48 R1 dockerd-current[4693]: time="2022-08-31T17:14:48.601033092+03:00" level=info  
Aug 31 17:14:49 R1 dockerd-current[4693]: time="2022-08-31T17:14:49.684857807+03:00" level=info  
Aug 31 17:14:49 R1 dockerd-current[4693]: time="2022-08-31T17:14:49.685677377+03:00" level=info  
Aug 31 17:14:49 R1 dockerd-current[4693]: time="2022-08-31T17:14:49.705355946+03:00" level=info  
Aug 31 17:14:49 R1 dockerd-current[4693]: time="2022-08-31T17:14:49.862707365+03:00" level=info  
Aug 31 17:14:50 R1 dockerd-current[4693]: time="2022-08-31T17:14:49.996975662+03:00" level=info  
Aug 31 17:14:50 R1 dockerd-current[4693]: time="2022-08-31T17:14:50.011521691+03:00" level=info  
Aug 31 17:14:50 R1 dockerd-current[4693]: time="2022-08-31T17:14:50.011994015+03:00" level=info  
Aug 31 17:14:50 R1 dockerd-current[4693]: time="2022-08-31T17:14:50.015793031+03:00" level=info  
Aug 31 17:14:50 R1 systemd[1]: Started Docker Application Container Engine.  
Hint: Some lines were ellipsized, use -l to show in full.  
[root@R1 ~]#
```

Создадим папки /srv/A и /srv/B

```
[root@R1 ~]# cd /srv/  
[root@R1 srv]# ll  
total 0  
[root@R1 srv]#  
[root@R1 srv]#  
[root@R1 srv]#  
[root@R1 srv]# mkdir ./A/ ./B/  
[root@R1 srv]# ll  
total 0  
drwxr-xr-x. 2 root root 6 Aug 31 17:16 A  
drwxr-xr-x. 2 root root 6 Aug 31 17:16 B  
[root@R1 srv]#
```

Создадим примитивный html файл. Скачаем докер образы

```

[root@R1 srv]# echo "This is A" >/srv/A/index.html
[root@R1 srv]# docker run --rm -d -v "/srv/A:/usr/share/nginx/html" --name
flag needs an argument: --name
See 'docker run --help'.
[root@R1 srv]# nginx-A nginx
-bash: nginx-A: command not found
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]#
[root@R1 srv]# docker run --rm -d -v "/srv/A:/usr/share/nginx/html" --name nginx-A nginx
Unable to find image 'nginx:latest' locally
Trying to pull repository docker.io/library/nginx ...
latest: Pulling from docker.io/library/nginx
7a6db449b51b: Pull complete
ca1981974b58: Pull complete
d4019c921e20: Pull complete
7cb804d746d4: Pull complete
e7a561826262: Pull complete
7247f6e5c182: Pull complete
Digest: sha256:b95a99feebf7797479e0c5eb5ec0bdafa5d9f504bc94da550c2f58e839ea6914f
Status: Downloaded newer image for docker.io/nginx:latest
bl07af628766583ea5ad2ced9cd2b361816362e86358f41a02878afc2b1f595f
[root@R1 srv]# cd a
-bash: cd: a: No such file or directory
[root@R1 srv]# cd ./A
[root@R1 A]# ll
total 4
-rw-r--r--. 1 root root 10 Aug 31 17:18 index.html
[root@R1 A]# cat index.html
This is A
[root@R1 A]#

```

```

[root@R1 srv]#
[root@R1 srv]# docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' nginx-A
172.17.0.2
[root@R1 srv]#

```

Повторим тоже самое для nginx-B

```
[root@R1 srv]# echo "This is B" > /srv/B/index.html
[root@R1 srv]# docker run --rm -d -v "/srv/B:/usr/share/nginx/html" --name nginx-B nginx
7a0f17d200e4c8735343957403e5d471d777704b7292aa7e56bf0f0d98e1817c
[root@R1 srv]# docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' nginx-B
172.17.0.3
[root@R1 srv]#
```

Изменим selinux контекст

```
[root@R1 srv]# semanage fcontext -a -t httpd_sys_content_t /srv/A/index.html
[root@R1 srv]# semanage fcontext -a -t httpd_sys_content_t /srv/B/index.html
[root@R1 srv]# restorecon -v /srv/A/index.html
restorecon reset /srv/A/index.html context unconfined_u:object_r:var_t:s0->unconfined_u:object_r:httpd_sys_content_t:s0
[root@R1 srv]# restorecon -v /srv/B/index.html
restorecon reset /srv/B/index.html context unconfined_u:object_r:var_t:s0->unconfined_u:object_r:httpd_sys_content_t:s0
[root@R1 srv]#
```

Создадим dummy1 интерфейс

```
ip link add dummy1 type dummy
```

Сделаем так что бы он появлялся при загрузке ОС

```
vim /etc/modprobe.d/dummy.conf
```

```
options dummy numdummies=2
~
~
~
```

Зададим настройки интерфейсу

```
vim /etc/sysconfig/network-scripts/ifcfg-dummy1
```

```
DEVICE=dummy1
TYPE=dummy
IPADDR=111.111.111.111
PREFIX=32
ONBOOT=yes
~
~
```

```
ifdown dummy1
```

```
ifup dummy1
```

Проверяем

```
13: dummy1: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether 22:23:ff:31:1a:af brd ff:ff:ff:ff:ff:ff
    inet 111.111.111.111/32 brd 111.111.111.111 scope global dummy1
        valid_lft forever preferred_lft forever
    inet6 fe80::2023:ffff:fe31:1aaf/64 scope link
        valid_lft forever preferred_lft forever
[root@R1 srv]#
```

Анонсируем подсеть через ospf

```
[root@R1 srv]# vtysh

Hello, this is FRRouting (version 8.3).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R1# conf t
R1(config)# router ospf
R1(config-router)# network
    A.B.C.D/M OSPF network prefix
R1(config-router)# network 111.111.111.111/32
    area Set the OSPF area ID
R1(config-router)# network 111.111.111.111/32 area 0
R1(config-router)#
```

Проверим на R2

```

R2# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, A - Babel, F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* 0.0.0.0/0 [0/100] via 192.168.0.1, ens33, 08:43:12
O>* 1.1.1.1/32 [110/110] via 192.168.12.15, team0, weight 1, 08:41:57
O   2.2.2.2/32 [110/10] via 0.0.0.0, dummy0 onlink, weight 1, 08:43:11
C>* 2.2.2.2/32 is directly connected, dummy0, 08:43:11
O>* 3.3.3.3/32 [110/110] via 192.168.23.2, ens35, weight 1, 08:42:14
O>* 111.111.111.111/32 [110/110] via 192.168.12.15, team0, weight 1, 00:00:50
K>* 169.254.0.0/16 [0/1006] is directly connected, dummy0, 08:43:11
C>* 192.168.0.0/24 is directly connected, ens33, 08:43:12
O   192.168.12.0/24 [110/100] is directly connected, team0, weight 1, 08:42:59
C>* 192.168.12.0/24 is directly connected, team0, 08:43:12
O   192.168.23.0/24 [110/100] is directly connected, ens35, weight 1, 08:43:12
C>* 192.168.23.0/24 is directly connected, ens35, 08:43:12
R2#

```

Маршрут получен.

Итак, у нас сейчас в ipvs пусто

```

[root@R1 srv]# ipvsadm -l -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
[root@R1 srv]#
[root@R1 srv]#

```

Нужно добавить сервис. Добавим локальный ip адрес и порт, с которых будет идти балансировка.

```
ipvsadm -A -t 111.111.111.111:80 -s rr
```

```
[root@R1 srv]# ipvsadm -l -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
[root@R1 srv]#
[root@R1 srv]# ipvsadm -A -t 111.111.111.111:80 -s rr
[root@R1 srv]# ipvsadm -l -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP 111.111.111.111:80 rr
[root@R1 srv]#
```

Добавим на какие адреса будет идти балансировка

```
[root@R1 srv]# ipvsadm -a -t 111.111.111.111:80 -r 172.17.0.2 -m
[root@R1 srv]# ipvsadm -a -t 111.111.111.111:80 -r 172.17.0.3 -m
[root@R1 srv]# ipvsadm -l -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP 111.111.111.111:80 rr
  -> 172.17.0.2:80              Masq    1        0          0
  -> 172.17.0.3:80              Masq    1        0          0
[root@R1 srv]#
```

Добавим правило в фаервол

```
[root@R1 srv]# firewall-cmd --zone=public --add-service=http --permanent
success
[root@R1 srv]#
[root@R1 srv]#
```

Рестарт, чтобы применились новые правила

```
systemctl restart firewalld
```

Отправим с R3 1000 запросов на адрес 111.111.111.111 порт 80

```
[root@R3 mailcow-dockerized]# for i in `seq 1 1000`; do curl http://111.111.111.111 -s; done | sort | uniq -c
500 This is A
500 This is B
[root@R3 mailcow-dockerized]#
```

Запросы распределились ровно пополам.

Посмотрим логи наших nginx'ов

Узнаем id наших контейнеров

```
[root@R1 dev]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7a0f17d200e4	nginx	"/docker-entrypoin..."	53 minutes ago	Up 53 minutes	80/tcp	nginx-B
b107af628766	nginx	"/docker-entrypoin..."	About an hour ago	Up About an hour	80/tcp	nginx-A

```
[root@R1 dev]#
```

Перейдём в директорию с контейнерами.
По ID контейнера найдем необходимую нам директорию

```
[root@R1 dev]# cd /var/lib/docker/containers/
[root@R1 containers]# ll
total 0
drwx-----. 5 root root 168 Aug 31 17:27 7a0f17d200e4c8735343957403e5d471d777704b7292aa7e56bf0f0d98e1817c
drwx-----. 5 root root 168 Aug 31 17:21 b107af628766583ea5ad2ced9cd2b361816362e86358f41a02878afc2b1f595f
drwx-----. 5 root root 168 Aug 31 18:15 c24d277011b0150eaaac78593a4bbd2f5142af9b34975d317ddab320deca3238
[root@R1 containers]#
[root@R1 containers]#
[root@R1 containers]# ls | grep b107af628766
b107af628766583ea5ad2ced9cd2b361816362e86358f41a02878afc2b1f595f
[root@R1 containers]#
```

Ничего похожего на лог файл не нашли

```
[root@R1 b107af628766583ea5ad2ced9cd2b361816362e86358f41a02878afc2b1f595f]# ll
total 24
drwx-----. 2 root root    6 Aug 31 17:21 checkpoints
-rw-r--r--. 1 root root 3018 Aug 31 17:21 config.v2.json
-rw-r--r--. 1 root root 1152 Aug 31 17:21 hostconfig.json
-rw-r--r--. 1 root root   13 Aug 31 17:21 hostname
-rw-r--r--. 1 root root  174 Aug 31 17:21 hosts
-rw-r--r--. 1 root root   72 Aug 31 17:21 resolv.conf
-rw-r--r--. 1 root root   71 Aug 31 17:21 resolv.conf.hash
drwxr-xr-x. 3 root root   18 Aug 31 17:21 secrets
drwxrwxrwt. 2 root root   40 Aug 31 17:21 shm
```

Похоже, что лог файл куда не пишется


```
[root@R1 containers]# docker inspect nginx-B | grep Log
    "LogPath": "",
    "LogConfig": {
[root@R1 containers]# docker inspect nginx-A | grep Log
    "LogPath": "",
    "LogConfig": {
[root@R1 containers]#
```

Логи уходят в стандартный поток вывода /dev/stdout и не сохраняются.

```
[root@R1 log]# docker run nginx ls -l /var/log/nginx
total 0
lrwxrwxrwx. 1 root root 11 Aug 23 03:59 access.log -> /dev/stdout
lrwxrwxrwx. 1 root root 11 Aug 23 03:59 error.log -> /dev/stderr
[root@R1 log]#
```