

Лабораторная работа №13

Настройка NFS

Сахно Никита НФИбд-02-23

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	2
3.1	Настройка сервера NFSv4.....	2
3.2	Монтирование NFS на клиенте	5
3.3	Подключение каталогов к дереву NFS.....	7
3.4	Подключение каталогов для работы пользователей	8
3.5	Внесение изменений в настройки внутреннего окружения виртуальных машин 9	
4	Выводы.....	11

1 Цель работы

Приобрести навыки настройки сервера NFS для удалённого доступа к ресурсам.

2 Задание

1. Установить и настроить сервер NFSv4.
2. Подмонтировать удалённый ресурс на клиенте.
3. Подключить каталог с контентом веб-сервера к дереву NFS.
4. Подключить каталог для удалённой работы вашего пользователя к дереву NFS.
5. Написать скрипты для Vagrant, фиксирующие действия по установке и настройке сервера NFSv4 во внутреннем окружении виртуальных машин server и client. Соответствующим образом внести изменения в Vagrantfile.

3 Выполнение лабораторной работы

3.1 Настройка сервера NFSv4

На сервере установим необходимое программное обеспечение: `dnf -y install nfs-utils`

```
Установка      : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/
Запуск скрипта : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/
Проверка       : gssproxy-0.8.4-7.el9.x86_64                1/
Проверка       : libev-4.33-6.el9.x86_64                    2/
Проверка       : libnfsidmap-1:2.5.4-38.el9.x86_64           3/
Проверка       : libverto-libev-0.3.2-3.el9.x86_64           4/
Проверка       : nfs-utils-1:2.5.4-38.el9.x86_64             5/
Проверка       : rpcbind-1.2.6-7.el9.x86_64                  6/
Проверка       : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/

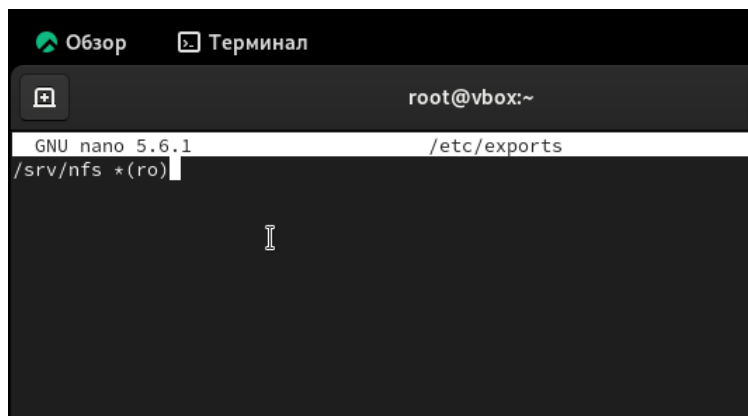
Установлен:
gssproxy-0.8.4-7.el9.x86_64      libev-4.33-6.el9.x86_64
libnfsidmap-1:2.5.4-38.el9.x86_64  libverto-libev-0.3.2-3.el9.x86_64
nfs-utils-1:2.5.4-38.el9.x86_64  rpcbind-1.2.6-7.el9.x86_64
sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64

Выполнено!
```

Установка пакетов

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS): `mkdir -p /srv/nfs`

В файле `/etc/exports` пропишем подключаемый через NFS общий каталог с доступом только на чтение: `/srv/nfs *(ro)`



Редактирование файла

Для общего каталога зададим контекст безопасности NFS: `semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"`

Применим изменённую настройку SELinux к файловой системе: `restorecon -vR /srv/nfs`

Запустим сервер NFS:

```
systemctl start nfs-server.service
systemctl enable nfs-server.service
```

Настроим межсетевой экран для работы сервера NFS:

```
firewall-cmd --add-service=nfs
firewall-cmd --add-service=nfs --permanent
firewall-cmd --reload
```

```
[root@vbox ~]# systemctl start nfs-server.service
[root@vbox ~]# systemctl enable nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/usr/lib/systemd/system/nfs-server.service.
[root@vbox ~]# firewall-cmd --add-service=nfs
success
[root@vbox ~]# firewall-cmd --add-service=nfs --permanent
success
[root@vbox ~]# firewall-cmd --reload
success
[root@vbox ~]#
```

Настройка межсетевого экрана

На клиенте установим необходимое для работы NFS программное обеспечение: `dnf -y install nfs-utils`

```
Установка      : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/
Запуск скрипта: sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/
Проверка       : gssproxy-0.8.4-7.el9.x86_64                1/
Проверка       : libev-4.33-6.el9.x86_64                    2/
Проверка       : libnfsidmap-1:2.5.4-38.el9.x86_64          3/
Проверка       : libverto-libev-0.3.2-3.el9.x86_64          4/
Проверка       : nfs-utils-1:2.5.4-38.el9.x86_64            5/
Проверка       : rpcbind-1.2.6-7.el9.x86_64                 6/
Проверка       : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/

Установлен:
gssproxy-0.8.4-7.el9.x86_64      libev-4.33-6.el9.x86_64
libnfsidmap-1:2.5.4-38.el9.x86_64  libverto-libev-0.3.2-3.el9.x86_64
nfs-utils-1:2.5.4-38.el9.x86_64  rpcbind-1.2.6-7.el9.x86_64
sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64

Выполнено!
```

Установка пакетов

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы (вместо user укажите свой логин):

```
showmount -e server.nvsakhno.net
```

```
[root@vbox ~]# showmount -e server.nvsakhno.net
mnt_create: RPC: Unknown host
[root@vbox ~]#
```

Просмотр подмонтированных удаленных ресурсов

Попробуем на сервере остановить сервис межсетевого экрана: `systemctl stop firewalld.service`

Затем на клиенте вновь попробуем подключиться к удалённо смонтированному ресурсу: `showmount -e server.nvsakhno.net`

На сервере запустим сервис межсетевого экрана `systemctl start firewallld`

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

```
lsof | grep TCP
lsof | grep UDP
```

```
root@vbox:~
>93.243.107.34.bc.googleusercontent.com:https (ESTABLISHED)
dovecot 4762 root 21u IPv4 44674 0t0 TCP *:pop3 (LIS
TEN)
dovecot 4762 root 22u IPv6 44675 0t0 TCP *:pop3 (LIS
TEN)
dovecot 4762 root 23u IPv4 44676 0t0 TCP *:pop3s (LI
TEN)
dovecot 4762 root 24u IPv6 44677 0t0 TCP *:pop3s (LI
TEN)
dovecot 4762 root 40u IPv4 44711 0t0 TCP *:imap (LIS
TEN)
dovecot 4762 root 41u IPv6 44712 0t0 TCP *:imap (LIS
TEN)
dovecot 4762 root 42u IPv4 44713 0t0 TCP *:imaps (LI
TEN)
dovecot 4762 root 43u IPv6 44714 0t0 TCP *:imaps (LI
TEN)
master 5049 root 13u IPv4 46559 0t0 TCP *:smtp (LIS
TEN)
sshd 5895 root 3u IPv4 115396 0t0 TCP *:down (LIS
TEN)
sshd 5895 root 4u IPv6 115398 0t0 TCP *:down (LIS
TEN)
sshd 5895 root 5u IPv4 115400 0t0 TCP *:ssh (LIST
EN)
sshd 5895 root 6u IPv6 115402 0t0 TCP *:ssh (LIST
EN)
rpcbind 7684 rpc 4u IPv4 120866 0t0 TCP *:sunrpc (L
ISTEN)
rpcbind 7684 rpc 6u IPv6 120311 0t0 TCP *:sunrpc (L
ISTEN)
rpc.statd 7686 rpcuser 8u IPv4 121516 0t0 TCP *:37697 (LI
STEN)
rpc.statd 7686 rpcuser 10u IPv6 121524 0t0 TCP *:36357 (LI
STEN)
rpc.mount 7690 root 5u IPv4 126452 0t0 TCP *:mountd (L
ISTEN)
rpc.mount 7690 root 7u IPv6 126460 0t0 TCP *:mountd (L
ISTEN)
[root@vbox ~]#
```

Задействованные службы при удаленном монтировании по протоколу TCP

```
root@vbox:~
0318 0t0 UDP *:sunrpc
avahi-dae 801 avahi 12u IPv4 1
7041 0t0 UDP *:mdns
avahi-dae 801 avahi 13u IPv6 1
7042 0t0 UDP *:mdns
NetworkMa 978 root 27u IPv4 11
9234 0t0 UDP vbox:bootpc->_gateway:bootps
NetworkMa 978 981 gmain root 27u IPv4 11
9234 0t0 UDP vbox:bootpc->_gateway:bootps
NetworkMa 978 982 gdbus root 27u IPv4 11
9234 0t0 UDP vbox:bootpc->_gateway:bootps
chronyd 6280 chrony 5u IPv4 11
8276 0t0 UDP localhost:323
chronyd 6280 chrony 6u IPv6 11
8277 0t0 UDP localhost:323
chronyd 6280 chrony 7u IPv4 11
8278 0t0 UDP *:ntp
rpcbind 7684 rpc 5u IPv4 12
0304 0t0 UDP *:sunrpc
rpcbind 7684 rpc 7u IPv6 12
0318 0t0 UDP *:sunrpc
rpc.statd 7686 rpcuser 7u IPv4 12
1512 0t0 UDP *:54477
rpc.statd 7686 rpcuser 9u IPv6 12
1520 0t0 UDP *:56124
rpc.statd 7686 rpcuser 11u IPv4 12
1505 0t0 UDP localhost:659
rpc.mount 7690 root 4u IPv4 12
6448 0t0 UDP *:mountd
rpc.mount 7690 root 6u IPv6 12
6456 0t0 UDP *:mountd
[root@vbox ~]#
```

Задействованные службы при удаленном монтировании по протоколу UDP


```

mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
/var/lib/napd/snaps/octave_306.snap on /var/lib/napd/snap/octave/306 type squashfs (ro,nodev,relatime,context=system_u:object_r:snappy_snap_t:s0,errors=continue,x-gdu.hide)
/var/lib/napd/snaps/snapd_25935.snap on /var/lib/napd/snap/snapd/25935 type squashfs (ro,nodev,relatime,context=system_u:object_r:snappy_snap_t:s0,errors=continue,x-gdu.hide)
/var/lib/napd/snaps/core18_2979.snap on /var/lib/napd/snap/core18/2979 type squashfs (ro,nodev,relatime,context=system_u:object_r:snappy_snap_t:s0,errors=continue,x-gdu.hide)
/dev/sdal on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=174760k,nr_inodes=43690,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)

```

Монтирование NFS на клиенте

На клиенте в конце файла `/etc/fstab` добавим следующую запись:
`server.nvsakhno.net:/srv/nfs /mnt/nfs nfs _netdev 0 0`

```

root@vbox:~
GNU nano 5.6.1 /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Jan 29 08:39:33 2026
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rl_vbox-root / xfs defaults >
UUID=c56db9f5-7c89-4468-a10b-643149818586 /boot xfs >
/dev/mapper/rl_vbox-swap none swap defaults >
server.nvsakhno.net:/srv/nfs /mnt/nfs nfs _netdev 0 0

```

Редактирование файла

На клиенте проверим наличие автоматического монтирования удалённых ресурсов при запуске операционной системы: `systemctl status remote-fs.target`

```

root@vbox ~]# systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled; p
   Active: active since Thu 2026-02-12 13:02:27 MSK; 2h 28min ago
   Until: Thu 2026-02-12 13:02:27 MSK; 2h 28min ago
   Docs: man:systemd.special(7)
lines 1-5/5 (END)

```

Проверка наличия автоматического монтирования удалённых ресурсов

Перезапустим клиент и убедимся, что удалённый ресурс подключается автоматически.

3.3 Подключение каталогов к дереву NFS

На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера: `mkdir -p /srv/nfs/www`

Подмонтируем каталог web-сервера: `mount -o bind /var/www/ /srv/nfs/www/`

На сервере проверим, что отображается в каталоге `/srv/nfs`.

```

root@vbox ~]# mkdir -p /srv/nfs/www
root@vbox ~]# mount -o bind /var/www/ /srv/nfs/www/
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
root@vbox ~]#

```

Содержимое каталога

На клиенте посмотрим, что отображается в каталоге `/mnt/nfs`.

```

[root@vbox ~]# cd /srv/nfs
[root@vbox nfs]# ls
www
[root@vbox nfs]#

```

Содержимое каталога

На сервере в файле `/etc/exports` добавим экспорт каталога веб-сервера с удалённого ресурса: `/srv/nfs/www 192.168.0.0/16(rw)`

```

root@vbox:/mnt/nfs
GNU nano 5.6.1 /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Jan 29 08:39:33 2026
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more
#
# After editing this file, run 'systemctl daemon-reload' to update syst
# units generated from this file.
#
/dev/mapper/r1_vbox-root / xfs defaults xfs
UUID=c36db9f5-7c89-4468-a10b-643149818586 /boot xfs defaults xfs
/dev/mapper/r1_vbox-swap none swap defaults
server.nvsakhno.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
/var/www /srv/nfs/www none bind 0 0

```

Редактирование файла

Экспортируем все каталоги, упомянутые в файле /etc/exports: `exportfs -r`

Проверим на клиенте каталог /mnt/nfs.

На сервере в конце файла /etc/fstab добавим следующую запись: `/var/www /srv/nfs/www none bind 0 0`

3.4 Подключение каталогов для работы пользователей

На сервере под пользователем nvsakhno в его домашнем каталоге создадим каталог common с полными правами доступа только для этого пользователя, а в нём файл nvsakhno@server.txt:

```
mkdir -p -m 700 ~/common
cd ~/common
touch nvsakhno@server.txt
```

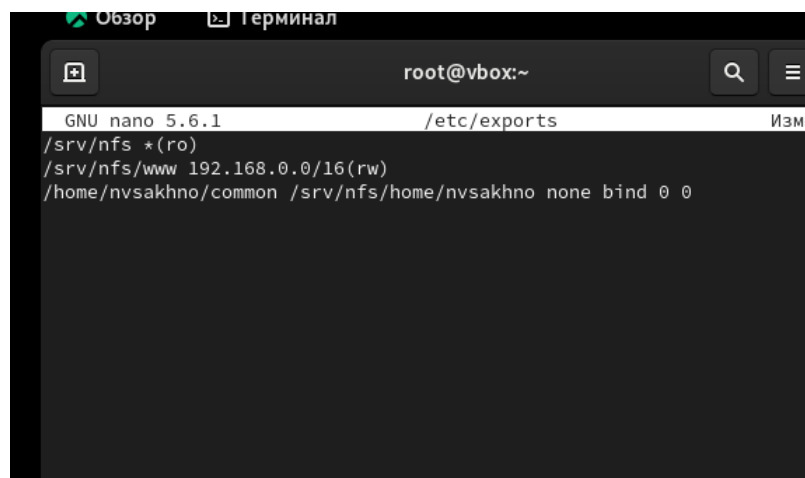
На сервере создадим общий каталог для работы пользователя nvsakhno по сети: `mkdir -p /srv/nfs/home/user`

Подмонтируем каталог common пользователя nvsakhno в NFS: `mount -o bind /home/user/common /srv/nfs/home/user`

```
root@vbox ~]# mkdir -p -m 700 ~/common
root@vbox ~]# cd ~/common
root@vbox common]# touch nvsakhno@server.txt
root@vbox common]#
```

Подключение каталогов для работы пользователей

Подключим каталог пользователя в файле /etc/exports, прописав в нём (вместо user укажите свой логин): `/srv/nfs/home/user 192.168.0.0/16(rw)`



```
Обзор Терминал
root@vbox:~
GNU nano 5.6.1 /etc/exports
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/home/nvsakhno/common /srv/nfs/home/nvsakhno none bind 0 0
```

Редактирование файла

Внесем изменения в файл /etc/fstab (вместо user укажите свой логин): `/home/user/common /srv/nfs/home/user none bind 0 0`

Повторно экспортируем каталоги: `exportfs -r`

На клиенте проверим каталог `/mnt/nfs`.

На клиенте под пользователем `user` перейдем в каталог `/mnt/nfs/home/user` и попробуем создать в нём файл `user@client.txt` и внести в него какие-либо изменения:

```
cd /mnt/nfs/home/user
touch user@client.txt
```

Безуспешно.

Попробуем это проделать под пользователем `root`.

Безуспешно.

На сервере посмотрим, появились ли изменения в каталоге пользователя `/home/user/common`.

Не появились, все тщетно.

3.5 Внесение изменений в настройки внутреннего окружения виртуальных машин

На виртуальной машине `server` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `nfs`, в который поместим в соответствующие подкаталоги конфигурационные файлы:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/nfs/etc
cp -R /etc/exports /vagrant/provision/server/nfs/etc/
```

В каталоге `/vagrant/provision/server` создадим исполняемый файл `nfs.sh`:

```
cd /vagrant/provision/server
touch nfs.sh
chmod +x nfs.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
root@server:/vagrant/provision/server/
GNU nano 5.6.1                                nfs.sh
#!/bin/bash

echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/nfs/etc/* /etc
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service nfs --permanent
firewall-cmd --add-service mountd --add-service rpc-bind --permanent
firewall-cmd --reload

echo "Tuning SELinux"
mkdir -p /srv/nfs
semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
restorecon -vR /srv/nfs

echo "Mounting dirs"
mkdir -p /srv/nfs/www
mount -o bind /var/www /srv/nfs/www
echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
mkdir -p /srv/nfs/home/user
mkdir -p -m 700 /home/user/common
chown user:user /home/user/common
mount -o bind /home/user/common /srv/nfs/home/user
echo "/home/user/common /srv/nfs/home/user none bind 0 0" >> /etc/fstab

echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server

systemctl restart firewalld
```

Редактирование файла

На виртуальной машине client перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/: `cd /vagrant/provision/client`

В каталоге /vagrant/provision/client создадим исполняемый файл nfs.sh:

```
cd /vagrant/provision/client
touch nfs.sh
chmod +x nfs.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
GNU nano 5.6.1                                nfs.sh
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install nfs-utils

echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.user.net:/srv/nfs /mnt/nfs
echo "server.user.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Редактирование файла

Для отработки созданных скриптов во время загрузки виртуальных машин server и client в конфигурационном файле Vagrantfile необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

```
server.vm.provision "server nfs",  
  type: "shell",  
  preserve_order: true,  
  path: "provision/server/nfs.sh"  
  
client.vm.provision "client nfs",  
  type: "shell",  
  preserve_order: true,  
  path: "provision/client/nfs.sh"
```

4 Выводы

В процессе выполнения данной лабораторной работы я приобрел навыки настройки сервера NFS для удалённого доступа к ресурсам.