

# **Лабораторная работа №13**

**Настройка NFS**

Шубина София Антоновна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Настройка сервера NFSv4 . . . . .	7
3.2	Монтирование NFS на клиенте . . . . .	12
3.3	Подключение каталогов к дереву NFS . . . . .	15
3.4	Подключение каталогов для работы пользователей . . . . .	18
3.5	Внесение изменений в настройки внутреннего окружения виртуальных машин . . . . .	22
<b>4</b>	<b>Выводы</b>	<b>25</b>

## Список иллюстраций

3.1	Установка пакетов . . . . .	7
3.2	создание каталога . . . . .	7
3.3	Редактирование файла . . . . .	8
3.4	зададим контекст безопасности и применим измененную настройку	8
3.5	Настройка межсетевого экрана . . . . .	8
3.6	Установка пакетов . . . . .	9
3.7	Просмотр подмонтированных удаленных ресурсов . . . . .	9
3.8	остановка сервиса межсетевого экрана . . . . .	9
3.9	Подключение к удаленно смонтированному ресурсу . . . . .	10
3.10	запустим сервис межсетевого экрана . . . . .	10
3.11	Задействованные службы при удаленном монтировании по прото- колу TCP . . . . .	10
3.12	Задействованные службы при удаленном монтировании по прото- колу UDP . . . . .	11
3.13	Настройка межсетевого экрана на сервере . . . . .	12
3.14	Подключение к удаленно смонтированному ресурсу . . . . .	12
3.15	создание каталога . . . . .	12
3.16	Монтирование NFS на клиенте . . . . .	13
3.17	Редактирование файла . . . . .	14
3.18	Проверка наличия автоматического монтирования удалённых ре- сурсов . . . . .	14
3.19	Проверка . . . . .	15
3.20	создание и подмонтирование каталога . . . . .	15
3.21	Содержимое каталога . . . . .	16
3.22	Содержимое каталога . . . . .	16
3.23	Редактирование файла . . . . .	16
3.24	Содержимое каталога . . . . .	17
3.25	Редактирование файла . . . . .	17
3.26	экспорт каталогов . . . . .	17
3.27	Содержимое каталога . . . . .	18
3.28	Создание каталога . . . . .	18
3.29	Подключение каталогов для работы пользователей . . . . .	19
3.30	Редактирование файла . . . . .	19
3.31	Редактирование файла . . . . .	20
3.32	права доступа . . . . .	20
3.33	Проверка содержимого каталога . . . . .	20
3.34	Создание файла . . . . .	21

3.35 Добавила описание . . . . .	21
3.36 попробуем через root . . . . .	21
3.37 изменения в каталоге . . . . .	21
3.38 создание файла . . . . .	22
3.39 Редактирование файла . . . . .	23
3.40 создание файла . . . . .	23
3.41 Редактирование файла . . . . .	24
3.42 Редактирование файла . . . . .	24
3.43 Редактирование файла . . . . .	24

# 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`

```
[root@server.sashubina.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64      26 kB/s | 48 kB  00:01
Extra Packages for Enterprise Linux 10 - x86_64      156 kB/s | 4.8 MB 00:31
Rocky Linux 10 - BaseOS                             10 kB/s | 4.3 kB 00:00
Rocky Linux 10 - AppStream                          17 kB/s | 4.3 kB 00:00
Rocky Linux 10 - CRB                               16 kB/s | 4.3 kB 00:00
Rocky Linux 10 - Extras                             12 kB/s | 3.1 kB 00:00
Dependencies resolved.
=====
Package            Architecture Version      Repository    Size
=====
Installing:
nfs-utils           x86_64      1:2.8.2-3.el10 baseos        473 k
Installing dependencies:
gssproxy            x86_64      0.9.2-10.el10 baseos        111 k
libev               x86_64      4.33-14.el10 baseos        52 k
libnfsidmap         x86_64      1:2.8.2-3.el10 baseos        61 k
libverto-libev      x86_64      0.3.2-10.el10 baseos        13 k
rpcbind             x86_64      1.2.7-3.el10 baseos        57 k
sssd-nfs-idmap      x86_64      2.10.2-3.el10_0.2 baseos        36 k
Transaction Summary
=====
Install 7 Packages

Total download size: 884 k
Installed size: 2.0 M
Downloading Packages:
(1/7): gssproxy-0.9.2-10.el10.x86_64.rpm             1.1 MB/s | 111 kB  00:00
(2/7): libnfsidmap-2.8.2-3.el10.x86_64.rpm           480 kB/s | 61 kB  00:00
(3/7): libverto-libev-0.3.2-10.el10.x86_64.rpm       168 kB/s | 13 kB  00:00
(4/7): libev-4.33-14.el10.x86_64.rpm                 282 kB/s | 52 kB  00:00
(5/7): sssd-nfs-idmap-2.10.2-3.el10_0.2.x86_64.rpm    1.2 MB/s | 36 kB  00:00
(6/7): nfs-utils-2.8.2-3.el10.x86_64.rpm             3.6 MB/s | 473 kB 00:00
(7/7): rpcbind-1.2.7-3.el10.x86_64.rpm               166 kB/s | 57 kB  00:00
=====
```

Рис. 3.1: Установка пакетов

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

```
[root@server.sashubina.net ~]# mkdir -p /srv/nfs
[root@server.sashubina.net ~]#
```

Рис. 3.2: создание каталога

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

```
GNU nano 8.1 /etc/exports
/srv/nfs *(ro)
```

Рис. 3.3: 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`

```
[root@server.sashubina.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.sashubina.net ~]# restorecon -vR /srv/nfs
Relabeled /srv/nfs from unconfined_u:object_r:var_t:s0 to unconfined_u:object_r:nfs_t:s0
[root@server.sashubina.net ~]#
```

Рис. 3.4: зададим контекст безопасности и применим изменённую настройку

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@server.sashubina.net ~]# systemctl start nfs-server.service
[root@server.sashubina.net ~]# 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@server.sashubina.net ~]# firewall-cmd --add-service=nfs
success
[root@server.sashubina.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.sashubina.net ~]# firewall-cmd --reload
success
[root@server.sashubina.net ~]#
```

Рис. 3.5: Настройка межсетевого экрана

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



```
[sashubina@client.sashubina.net ~]$ sudo -i
[sudo] password for sashubina:
[root@client.sashubina.net ~]# dnf -y install nfs-utils

Extra Packages for Enterprise Linux 10 2.6 kB/s | 40 kB 00:15
Extra Packages for Enterprise Linux 10 8.8 MB/s | 4.8 MB 00:00
Rocky Linux 10 - BaseOS 7.4 kB/s | 4.3 kB 00:00
Rocky Linux 10 - AppStream 11 kB/s | 4.3 kB 00:00
Rocky Linux 10 - CRB 12 kB/s | 4.3 kB 00:00
Rocky Linux 10 - Extras 8.8 kB/s | 3.1 kB 00:00
Dependencies resolved.

=====
Package Arch Version Repository Size
=====
Installing:
nfs-utils x86_64 1:2.8.2-3.el10 baseos 473 k
Installing dependencies:
gssproxy x86_64 0.9.2-10.el10 baseos 111 k
libev x86_64 4.33-14.el10 baseos 52 k
libnfsidmap x86_64 1:2.8.2-3.el10 baseos 61 k
libverto-libev x86_64 0.3.2-10.el10 baseos 13 k
rpcbind x86_64 1.2.7-3.el10 baseos 57 k
sssd-nfs-idmap x86_64 2.10.2-3.el10_0.2 baseos 36 k
=====

Transaction Summary
=====
Install 7 Packages

Total download size: 804 k
Installed size: 2.0 M
Downloading Packages:
(1/7): gssproxy-0.9.2-10.el10.x86_64.rpm 1.4 MB/s | 111 kB 00:00
(2/7): libverto-libev-0.3.2-10.el10.x86_64.rpm 455 kB/s | 13 kB 00:00
(3/7): libev-4.33-14.el10.x86_64.rpm 455 kB/s | 52 kB 00:00
(4/7): libnfsidmap-2.8.2-3.el10.x86_64.rpm 500 kB/s | 61 kB 00:00
(5/7): nfs-utils-2.8.2-3.el10.x86_64.rpm 8.1 MB/s | 473 kB 00:00
(6/7): rpcbind-1.2.7-3.el10.x86_64.rpm 973 kB/s | 57 kB 00:00
(7/7): sssd-nfs-idmap-2.10.2-3.el10_0.2.x86_64.rpm 36 kB 00:00
=====
```

Рис. 3.6: Установка пакетов

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы:

```
showmount -e server.sashubina.net
```

```
[root@client.sashubina.net ~]# showmount -e server.sashubina.net
clnt_create: RPC: Unable to receive
[root@client.sashubina.net ~]#
```

Рис. 3.7: Просмотр подмонтированных удаленных ресурсов

Мы не получили доступ и не смогли посмотреть подмонтированные ресурсы. Команда завершилась ошибкой RPC: Unable to receive. Это означает, что клиент не смог подключиться к серверу NFS (server.sashubina.net)

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

```
[root@server.sashubina.net ~]# systemctl stop firewalld.service
[root@server.sashubina.net ~]#
```

Рис. 3.8: остановка сервиса межсетевого экрана

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

```
[root@client.sashubina.net ~]# showmount -e server.sashubina.net
Export list for server.sashubina.net:
/srv/nfs *
[root@client.sashubina.net ~]#
```

Рис. 3.9: Подключение к удаленно смонтированному ресурсу

Команда выполнена успешно. Видно, что сервер server.sashubina.net экспортирует директорию /srv/nfs и разрешает доступ всем хостам (\*).

На сервере запустим сервис межсетевого экрана `systemctl start firewalld`

```
[root@server.sashubina.net ~]# systemctl start firewalld
[root@server.sashubina.net ~]#
```

Рис. 3.10: запустим сервис межсетевого экрана

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

```
lsof | grep TCP
```

```
lsof | grep UDP
```

```
[root@server.sashubina.net ~]# lsof | grep TCP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
Output information may be incomplete.
systemd 1 root 426u IPv6 8854 0t0 TCP *:websock (LISTEN)
systemd 1 root 422u IPv4 42421 0t0 TCP *:sunrpc (LISTEN)
systemd 1 root 445u IPv6 42435 0t0 TCP *:sunrpc (LISTEN)
cupsd 1237 root 7u IPv6 11445 0t0 TCP localhost:ipp (LISTEN)
cupsd 1237 root 8u IPv4 11446 0t0 TCP localhost:ipp (LISTEN)
sshd 1253 root 7u IPv4 11463 0t0 TCP *:down (LISTEN)
sshd 1253 root 8u IPv6 11465 0t0 TCP *:down (LISTEN)
sshd 1253 root 9u IPv4 11467 0t0 TCP *:ssh (LISTEN)
sshd 1253 root 18u IPv6 11469 0t0 TCP *:ssh (LISTEN)
named 1341 named 27u IPv4 10793 0t0 TCP localhost:domain (LISTEN)
named 1341 named 29u IPv4 10794 0t0 TCP localhost:domain (LISTEN)
named 1341 named 33u IPv6 10797 0t0 TCP localhost:domain (LISTEN)
named 1341 named 34u IPv6 10798 0t0 TCP localhost:domain (LISTEN)
named 1341 named 35u IPv4 11530 0t0 TCP localhost:rndc (LISTEN)
named 1341 named 36u IPv4 11531 0t0 TCP localhost:rndc (LISTEN)
named 1341 named 37u IPv6 11533 0t0 TCP localhost:rndc (LISTEN)
named 1341 named 38u IPv6 11534 0t0 TCP localhost:rndc (LISTEN)
named 1341 named 41u IPv4 11744 0t0 TCP server.sashubina.net:domain (LISTEN)
named 1341 named 42u IPv4 11745 0t0 TCP server.sashubina.net:domain (LISTEN)
named 1341 named 45u IPv4 24297 0t0 TCP dhcp.sashubina.net:domain (LISTEN)
named 1341 named 46u IPv4 24298 0t0 TCP dhcp.sashubina.net:domain (LISTEN)
named 1341 1343 isc-net-0 named 27u IPv4 10793 0t0 TCP localhost:domain (LISTEN)
named 1341 1343 isc-net-0 named 29u IPv4 10794 0t0 TCP localhost:domain (LISTEN)
named 1341 1343 isc-net-0 named 33u IPv6 10797 0t0 TCP localhost:domain (LISTEN)
named 1341 1343 isc-net-0 named 34u IPv6 10798 0t0 TCP localhost:domain (LISTEN)
named 1341 1343 isc-net-0 named 35u IPv4 11530 0t0 TCP localhost:rndc (LISTEN)
named 1341 1343 isc-net-0 named 36u IPv4 11531 0t0 TCP localhost:rndc (LISTEN)
named 1341 1343 isc-net-0 named 37u IPv6 11533 0t0 TCP localhost:rndc (LISTEN)
named 1341 1343 isc-net-0 named 38u IPv6 11534 0t0 TCP localhost:rndc (LISTEN)
named 1341 1343 isc-net-0 named 41u IPv4 11744 0t0 TCP server.sashubina.net:domain (LISTEN)
named 1341 1343 isc-net-0 named 42u IPv4 11745 0t0 TCP server.sashubina.net:domain (LISTEN)
named 1341 1343 isc-net-0 named 45u IPv4 24297 0t0 TCP dhcp.sashubina.net:domain (LISTEN)
named 1341 1343 isc-net-0 named 46u IPv4 24298 0t0 TCP dhcp.sashubina.net:domain (LISTEN)
```

Рис. 3.11: Задействованные службы при удаленном монтировании по протоколу TCP

Команда `lsof | grep TCP` показывает все открытые TCP-сокеты на сервере. В выводе видно, что сервер прослушивает несколько критически важных служб:

sshd на порту 22 (ssh) для удалённого управления, sunrpc (порт 111) для работы NFS, domain (порт 53) для DNS-сервера named, а также cupsd для печати и websn для веб-службы.

```
[root@sasubina.net ~]# lsuf | grep UDP
lsuf: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
lsuf: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
Output information may be incomplete.
systemd 1 root 436u IPv4 42428 0t0 UDP *:sunrpc
systemd 1 root 449u IPv6 42442 0t0 UDP *:sunrpc
avahi-daemon 885 avahi 12u IPv4 7795 0t0 UDP *:mdns
avahi-daemon 885 avahi 13u IPv6 7796 0t0 UDP *:mdns
chronyd 970 chrony 5u IPv4 9419 0t0 UDP localhost:323
chronyd 970 chrony 6u IPv6 9420 0t0 UDP localhost:323
chronyd 970 chrony 7u IPv4 9421 0t0 UDP *:ntp
named 1341 named 6u IPv4 11742 0t0 UDP server.sashubina.net:domain
named 1341 named 25u IPv4 10791 0t0 UDP localhost:domain
named 1341 named 26u IPv4 10792 0t0 UDP localhost:domain
named 1341 named 31u IPv6 10795 0t0 UDP localhost:domain
named 1341 named 32u IPv6 10796 0t0 UDP localhost:domain
named 1341 named 40u IPv4 11743 0t0 UDP server.sashubina.net:domain
named 1341 named 43u IPv4 24295 0t0 UDP www.sashubina.net:domain
named 1341 named 44u IPv4 24296 0t0 UDP www.sashubina.net:domain
named 1341 1343 isc-net-0 named 6u IPv4 11742 0t0 UDP server.sashubina.net:domain
named 1341 1343 isc-net-0 named 25u IPv4 10791 0t0 UDP localhost:domain
named 1341 1343 isc-net-0 named 26u IPv4 10792 0t0 UDP localhost:domain
named 1341 1343 isc-net-0 named 31u IPv6 10795 0t0 UDP localhost:domain
named 1341 1343 isc-net-0 named 32u IPv6 10796 0t0 UDP localhost:domain
named 1341 1343 isc-net-0 named 40u IPv4 11743 0t0 UDP server.sashubina.net:domain
named 1341 1343 isc-net-0 named 43u IPv4 24295 0t0 UDP www.sashubina.net:domain
named 1341 1343 isc-net-0 named 44u IPv4 24296 0t0 UDP www.sashubina.net:domain
named 1341 1344 isc-net-0 named 6u IPv4 11742 0t0 UDP server.sashubina.net:domain
named 1341 1344 isc-net-0 named 25u IPv4 10791 0t0 UDP localhost:domain
named 1341 1344 isc-net-0 named 26u IPv4 10792 0t0 UDP localhost:domain
named 1341 1344 isc-net-0 named 31u IPv6 10795 0t0 UDP localhost:domain
named 1341 1344 isc-net-0 named 32u IPv6 10796 0t0 UDP localhost:domain
named 1341 1344 isc-net-0 named 40u IPv4 11743 0t0 UDP server.sashubina.net:domain
named 1341 1344 isc-net-0 named 43u IPv4 24295 0t0 UDP www.sashubina.net:domain
named 1341 1344 isc-net-0 named 44u IPv4 24296 0t0 UDP www.sashubina.net:domain
named 1341 1346 isc-net-0 named 6u IPv4 11742 0t0 UDP server.sashubina.net:domain
named 1341 1346 isc-net-0 named 25u IPv4 10791 0t0 UDP localhost:domain
```

Рис. 3.12: Задействованные службы при удаленном монтировании по протоколу UDP

вывод команды показывает активные UDP-сервисы на сервере, где ключевыми являются DNS-сервер (named), прослушивающий порт 53 для обработки DNS-запросов как от внешних клиентов, так и от локальных приложений; NTP-сервер (chronyd), работающий на порту 123 для синхронизации времени в сети; Avahi-daemon на порту 5353 для автоматического обнаружения сетевых сервисов; и критически важные RPC-сервисы (systemd) на порту 111, обеспечивающие работу NFS-системы. Предупреждения о недоступности FUSE-файловых систем не влияют на сетевую функциональность и являются стандартными для данной среды.

Добавим службы rpc-bind и mountd в настройки межсетевого экрана на сервере:

```
firewall-cmd --get-services
firewall-cmd --add-service=mountd --add-service=rpc-bind
firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
firewall-cmd --reload
```

[illegible]

Рис. 3.13: Настройка межсетевого экрана на сервере

На клиенте проверим подключение удалённого ресурса: `showmount -e server.sashubina.net`

```
[root@client.sashubina.net ~]# showmount -e server.sashubina.net
Export list for server.sashubina.net:
/srv/nfs *
```

Рис. 3.14: Подключение к удаленно смонтированному ресурсу

Подключение успешно. Теперь, когда все необходимые порты открыты (`grs-bind` (порт 111) позволяет клиенту узнать, на каких портах висят службы NFS, а `mountd` позволяет управлять запросами на монтирование), запрос клиента доходит до службы `grs.mountd` на сервере. Сервер обрабатывает запрос, формирует список экспортов и отправляет его обратно клиенту.

## 3.2 Монтирование NFS на клиенте

На клиенте создадим каталог, в который будет монтироваться удалённый ресурс, и подмонтируем дерево NFS:

```
mkdir -p /mnt/nfs
```

```
mount server.sashubina.net:/srv/nfs /mnt/nfs
```

```
[root@client.sashubina.net ~]# mkdir -p /mnt/nfs
[root@client.sashubina.net ~]# mount server.sashubina.net:/srv/nfs /mnt/
nfs
[root@client.sashubina.net ~]#
```

Рис. 3.15: создание каталога

## Проверим, что общий ресурс NFS подключён правильно: mount

```
[root@client.sashubina.net ~]# mount
/dev/mapper/r1_vbox-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4896k,nr_inodes=211279,mode=755,inode64)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=345560k,nr_inodes=819200,mode=755,inode64)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=36,pgpr=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=5657)
queue on /dev/queue type queue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymlfollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
/dev/sda2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=ascii,shortname=winnt,errors=remount-ro)
/dev/mapper/r1_vbox-home on /home type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000,_netdev)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=172776k,nr_inodes=43194,mode=700,uid=1001,gid=1001,inode64)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
portal on /run/user/1001/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=172776k,nr_inodes=43194,mode=700,inode64)
server.sashubina.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
[root@client.sashubina.net ~]#
```

Рис. 3.16: Монтирование NFS на клиенте

Видно, что удаленный каталог `server.sashubina.net:/srv/nfs` смонтирован в локальную директорию `/mnt/nfs` с использованием NFS версии 4.2. Параметр `rw` указывает на доступность как чтения, так и записи, а `proto=tcp` подтверждает работу через TCP-протокол. Настройки `hard` и `timeo=600` означают жесткое монтирование с 10-минутными таймаутами. IP-адреса 192.168.1.1 (сервер) и 192.168.1.30 (клиент) показывают established-соединение. Это доказывает полную работоспособность NFS-сервера, корректную настройку фаервола и успешное сетевое взаимодействие между машинами.

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

```
GNU nano 8.1 /etc/fstab Modified
#
# /etc/fstab
# Created by anaconda on Mon Sep  8 16:08:06 2025
#
# 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.
#
UUID=3b63f118-3ab0-4ef9-924b-0f58bc98b6f7 / xfs >
UUID=c2609db3-859d-41c9-a1a4-f66785f24f62 /boot xfs >
UUID=F61C-8765 /boot/efi vfat umask=0077,shor >
UUID=e05cd838-7f05-42f5-bdf8-498e29d736d5 /home xfs >
UUID=10d4a7c4-f6ca-46ac-ae28-80797fcce7c4 none swap >
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not mod>
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
server.sashubina.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рис. 3.17: Редактирование файла

Эта запись в `fstab` обеспечивает автоматическое монтирование NFS-шары при загрузке системы. Синтаксис: `server.sashubina.net:/srv/nfs` - удаленный ресурс, `/mnt/nfs` - точка монтирования, `nfs` - тип ФС. Ключевая опция `_netdev` гарантирует, что монтирование произойдет только после полной инициализации сети, предотвращая зависание системы при недоступности сервера. Последние два нуля отключают резервное копирование и проверку ФС для сетевых ресурсов.

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

```
[root@client.sashubina.net ~]# systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled;
   Active: active since Wed 2025-11-05 18:56:14 UTC; 29min ago
   Invocation: 2d12462253f2442d920d0208d0248f4d
   Docs: man:systemd.special(7)

Nov 05 18:56:14 client.sashubina.net systemd[1]: Reached target remote-
lines 1-7/7 (END)
```

Рис. 3.18: Проверка наличия автоматического монтирования удалённых ресурсов

Все работает корректно.

Перезапустим клиент и убедимся, что удалённый ресурс подключается автоматически.

```
[root@client.sashubina.net ~]# systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled;
   Active: active since Wed 2025-11-05 19:27:13 UTC; 19min ago
   Invocation: 7fc04932032446deaa78023d83dff520
   Docs: man:systemd.special(7)

Nov 05 19:27:13 client.sashubina.net systemd[1]: Reached target remote-

[1]+  Stopped                  systemctl status remote-fs.target
[root@client.sashubina.net ~]# mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.sashubina.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.
2,rsize=262144,wsizе=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=
2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1,_netc
su\
```

Рис. 3.19: Проверка

После перезагрузки, мы видим, что все работает корректно.

### 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 веб-сервера: `mount -o bind /var/www/ /srv/nfs/www/`

```
[root@server.sashubina.net ~]# mkdir -p /srv/nfs/www
[root@server.sashubina.net ~]# mount -o bind /var/www/ /srv/nfs/www/
[root@server.sashubina.net ~]#
```

Рис. 3.20: создание и подмонтирование каталога

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

```
[root@server.sashubina.net srv]# ls -la /srv/nfs
total 0
drwxr-xr-x. 3 root root 17 Nov 5 19:49 .
drwxr-xr-x. 3 root root 17 Nov 5 19:03 ..
drwxr-xr-x. 4 apache apache 33 Sep 23 17:40 www
[root@server.sashubina.net srv]#
```

Рис. 3.21: Содержимое каталога

На сервере в каталоге /srv/nfs находится директория www, принадлежащая пользователю и группе apache, что указывает на связь с веб-сервером.

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

```
[root@client.sashubina.net ~]# ls -la /mnt/nfs
total 0
drwxr-xr-x. 3 root root 17 Nov 5 19:49 .
drwxr-xr-x. 3 root root 17 Nov 5 19:18 ..
drwxr-xr-x. 2 root root 6 Nov 5 19:49 www
[root@client.sashubina.net ~]#
```

Рис. 3.22: Содержимое каталога

Первоначальный просмотр точки монтирования на клиенте показывает пустую директорию www с правами root, что указывает на необходимость настройки монтирования или обновления экспортов.

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

```
GNU nano 8.1 /etc/exports
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
```

Рис. 3.23: Редактирование файла

В конфигурации NFS экспорта прописан доступ для подсети 192.168.0.0/16 с правами чтения и записи (rw) к каталогу /srv/nfs/www, что обеспечивает сетевое взаимодействие между сервером и клиентами.

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



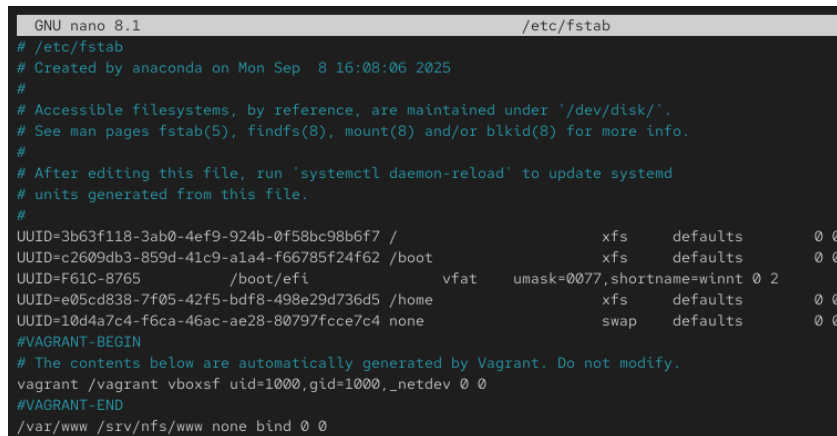
Проверим на клиенте каталог /mnt/nfs.

```
[root@client.sashubina.net ~]# ls -la /mnt/nfs
total 0
drwxr-xr-x. 3 root root 17 Nov  5 19:49 .
drwxr-xr-x. 3 root root 17 Nov  5 19:18 ..
drwxr-xr-x. 4  48   48 33 Sep 23 17:40 www
```

Рис. 3.24: Содержимое каталога

На клиенте в точке монтирования /mnt/nfs видна директория www, но с правами root вместо ожидаемого владельца apache, что может свидетельствовать о проблемах с отображением идентификаторов пользователей.

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

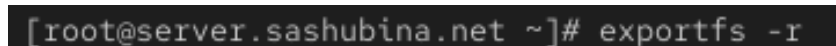


```
GNU nano 8.1 /etc/fstab
# /etc/fstab
# Created by anaconda on Mon Sep  8 16:08:06 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=3b63f118-3ab0-4ef9-924b-0f58bc98b6f7 / xfs defaults 0 0
UUID=c2609db3-859d-41c9-a1a4-f66785f24f62 /boot xfs defaults 0 0
UUID=F61C-8765 /boot/efi vfat umask=0077,shortname=winnt 0 2
UUID=e05cd838-7f05-42f5-bdf8-498e29d736d5 /home xfs defaults 0 0
UUID=10d4a7c4-f6ca-46ac-ae28-80797fccc7c4 none swap defaults 0 0
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
/var/www /srv/nfs/www none bind 0 0
```

Рис. 3.25: Редактирование файла

В файле fstab сервера добавлена bind-запись, которая связывает содержимое /var/www с /srv/nfs/www, позволяя экспортировать через NFS данные веб-сервера без их физического перемещения.

Повторно экспортируем каталоги, указанные в файле /etc/exports: exportfs -r



```
[root@server.sashubina.net ~]# exportfs -r
```

Рис. 3.26: экспорт каталогов

На клиенте проверим каталог /mnt/nfs. Перед этим подмонтируем каталог

```
[root@client.sashubina.net ~]# mount -t nfs -o vers=3 server.sashubina.net:/srv/nfs/www /mnt/nfs
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.
[root@client.sashubina.net ~]# ls -la /mnt/nfs
total 0
drwxr-xr-x. 4 48 48 33 Sep 23 17:40 .
drwxr-xr-x. 3 root root 17 Nov 5 19:18 ..
drwxr-xr-x. 2 48 48 6 Jul 15 00:00 cgi-bin
drwxr-xr-x. 5 48 48 100 Oct 27 17:07 html
[root@client.sashubina.net ~]#
```

Рис. 3.27: Содержимое каталога

На клиенте выполнено монтирование NFS версии 3, и в точке монтирования /mnt/nfs отображается содержимое удаленного каталога с идентификаторами пользователя и группы 48 (соответствует apache на сервере), включая подкаталоги cgi-bin и html.

### 3.4 Подключение каталогов для работы пользователей

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

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

```
[root@server.sashubina.net ~]# mkdir -p -m 700 ~/common
[root@server.sashubina.net ~]# cd ~/common
[root@server.sashubina.net common]# touch sashubina@server.txt
[root@server.sashubina.net common]#
```

Рис. 3.28: Создание каталога

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

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

```
[root@server.sashubina.net common]# mkdir -p /srv/nfs/home/sashubina
[root@server.sashubina.net common]# mount -o bind /home/sashubina/common /srv/nfs/home/sashubina
mount: /srv/nfs/home/sashubina: special device /home/sashubina/common does not exist.
dmesg(1) may have more information after failed mount system call.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@server.sashubina.net common]# systemctl daemon-reload
[root@server.sashubina.net common]#
```

Рис. 3.29: Подключение каталогов для работы пользователей

После выполнения команды `mount -o bind` для монтирования каталога `/home/user/common` в `/srv/nfs/home/user` устанавливаются точно такие же права доступа, как и у исходного каталога, поскольку bind-монтирование не изменяет атрибуты файловой системы, а лишь предоставляет доступ к содержимому одного каталога через другую точку монтирования. Сохраняются все исходные характеристики: владелец и группа остаются прежними, права доступа `chmod` полностью соответствуют оригинальным настройкам, а контекст SELinux наследуется от исходного каталога без изменений.

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)`

```
GNU nano 8.1 /etc/exports
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/sashubina 192.168.0.0/16(rw)
```

Рис. 3.30: Редактирование файла

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

```

GNU nano 8.1 /etc/fstab
# /etc/fstab
# Created by anaconda on Mon Sep  8 16:08:06 2025
#
# 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.
#
UUID=3b63f118-3ab0-4ef9-924b-0f58bc98b6f7 / xfs defaults
UUID=c2609db3-859d-41c9-a1a4-f66785f24f62 /boot xfs defaults
UUID=F61C-8765 /boot/efi vfat umask=0077,shortname=winnt 0
UUID=e05cd838-7f05-42f5-bdf8-498e29d736d5 /home xfs defaults
UUID=10d4a7c4-f6ca-46ac-ae28-80797fcce7c4 none swap defaults
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
/var/www /srv/nfs/www none bind 0 0
/home/sashubina/common /srv/nfs/home/sashubina none bind 0 0

```

Рис. 3.31: Редактирование файла

Повторно экспортируем каталоги: `exportfs -r`, а также я поменяла права доступа

```

[root@server.sashubina.net common]# chown -R sashubina:sashubina /home/sashubina/common/
[root@server.sashubina.net common]# chown -R sashubina:sashubina /srv/nfs/home/sashubina/

```

Рис. 3.32: права доступа

На клиенте проверим каталог `/mnt/nfs`.

```

[sashubina@client.sashubina.net ~]$ ls -la /mnt/nfs
total 0
drwx-----. 2 sashubina sashubina 34 Nov  5 20:45 .
drwxr-xr-x. 3 root      root      17 Nov  5 19:18 ..
-rw-r--r--. 1 sashubina sashubina  0 Nov  5 20:47 sashubina@server.txt
[sashubina@client.sashubina.net ~]$

```

Рис. 3.33: Проверка содержимого каталога

в точке монтирования `/mnt/nfs` отображается содержимое удаленного каталога с файлом `sashubina@server.txt`, причем владелец и группа правильно отображаются как `sashubina`, что подтверждает корректную работу NFS с сохранением прав доступа.

На клиенте под пользователем `user` перейдем в каталог `/mnt/nfs/home/sashubina` и попробуем создать в нём файл `sashubina@client.txt` и внести в него какие-либо изменения, перед этим в `root` я размонтировала каталог, добавила

/mnt/nfs/home/sashubina и вышла из root

```
cd /mnt/nfs/home/sashubina
```

```
touch sashubina@client.txt
```

```
[sashubina@client.sashubina.net ~]$ cd /mnt/nfs/home/sashubina
[sashubina@client.sashubina.net sashubina]$ touch sashubina@client.txt
[sashubina@client.sashubina.net sashubina]$ █
```

Рис. 3.34: Создание файла

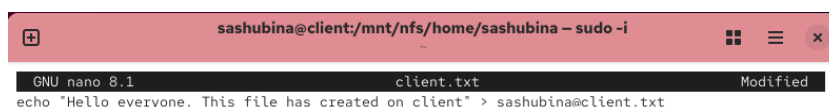


Рис. 3.35: Добавила описание

Попробуем проделать это под пользователем root.

```
[root@client.sashubina.net ~]# cd /mnt/nfs/home/sashubina
-bash: cd: /mnt/nfs/home/sashubina: Permission denied
[root@client.sashubina.net ~]# █
```

Рис. 3.36: попробуем через root

Каталог /home/sashubina/common на сервере, который был примонтирован через bind, имеет строгие права доступа 700 (только владелец). Как видно из предыдущих скринов, владельцем является пользователь sashubina, а права drwx— означают, что только сам пользователь sashubina имеет доступ к этому каталогу

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

```
[root@server.sashubina.net ~]# ls -la /home/sashubina/common
total 8
drwx-----. 2 sashubina sashubina 80 Nov 5 21:06 .
drwx-----. 17 sashubina sashubina 4096 Nov 5 20:44 ..
-rw-r--r--. 1 sashubina sashubina 78 Nov 5 21:06 client.txt
-rw-r--r--. 1 sashubina sashubina 0 Nov 5 21:03 sashubina@client.txt
-rw-r--r--. 1 sashubina sashubina 0 Nov 5 20:47 sashubina@server.txt
[root@server.sashubina.net ~]# █
```

Рис. 3.37: изменения в каталоге

вид на сервере: в исходном каталоге /home/sashubina/common видно три файла, включая client.txt и sashubina@server.txt, все файлы принадлежат пользователю sashubina, причем директория имеет строгие права 700 (только владелец), что объясняет почему в клиентском представлении тоже показываются права 700.

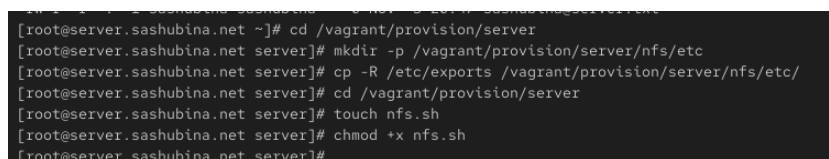
## 3.5 Внесение изменений в настройки внутреннего окружения виртуальных машин

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:

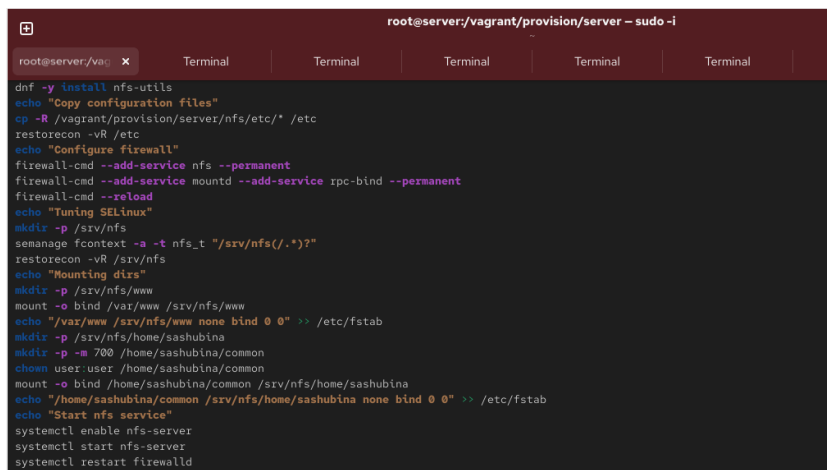
```
cd /vagrant/provision/server
touch nfs.sh
chmod +x nfs.sh
```



```
[root@server.sashubina.net ~]# cd /vagrant/provision/server
[root@server.sashubina.net server]# mkdir -p /vagrant/provision/server/nfs/etc
[root@server.sashubina.net server]# cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[root@server.sashubina.net server]# cd /vagrant/provision/server
[root@server.sashubina.net server]# touch nfs.sh
[root@server.sashubina.net server]# chmod +x nfs.sh
[root@server.sashubina.net server]#
```

Рис. 3.38: создание файла

Открыв его на редактирование, пропишем в нём следующий скрипт:



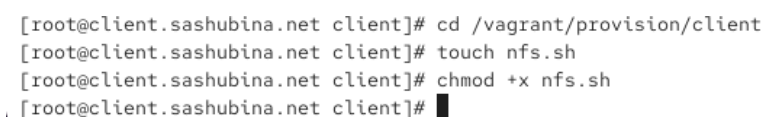
```
root@server:/vagrant/provision/server - sudo -i
dnf -y install nfs-utils
echo "Copy configuration files"
cp -R /vagrant/provision/server/nfs/etc/* /etc
restorecon -vR /etc
echo "Configure firewall"
firewall-cmd --add-service nfs --permanent
firewall-cmd --add-service mountd --add-service rpc-bind --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/sashubina
mkdir -p -m 700 /home/sashubina/common
chown user:user /home/sashubina/common
mount -o bind /home/sashubina/common /srv/nfs/home/sashubina
echo "/home/sashubina/common /srv/nfs/home/sashubina none bind 0 0" >> /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld
```

Рис. 3.39: 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
```



```
[root@client.sashubina.net client]# cd /vagrant/provision/client
[root@client.sashubina.net client]# touch nfs.sh
[root@client.sashubina.net client]# chmod +x nfs.sh
[root@client.sashubina.net client]#
```

Рис. 3.40: создание файла

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
GNU nano 8.1                                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.sashubina.net:/srv/nfs /mnt/nfs
echo "server.sashubina.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Рис. 3.41: 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"
```

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

Рис. 3.42: Редактирование файла

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

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

Рис. 3.43: Редактирование файла



## 4 Выводы

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