# Лабораторная работа №13

## Настройка NFS

Сахно Никита НФИбд-02-23

## Содержание



## 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

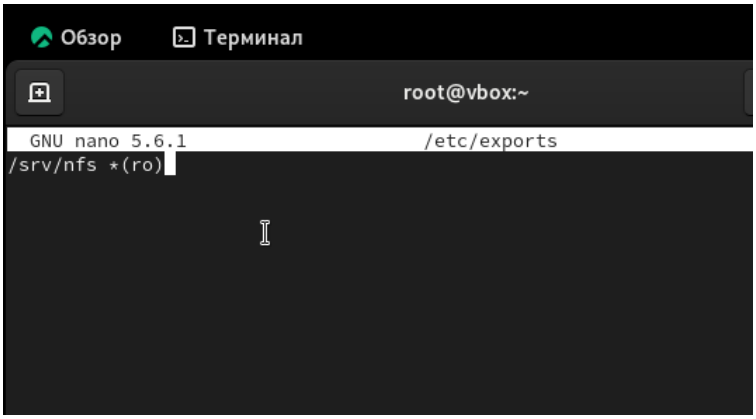
На сервере установим необходимое программное обеспечение: `dnf -y install nfs-utils`

```
  Установка        : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64                7/
  Запуск скриптлета: 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(/.*)?"`

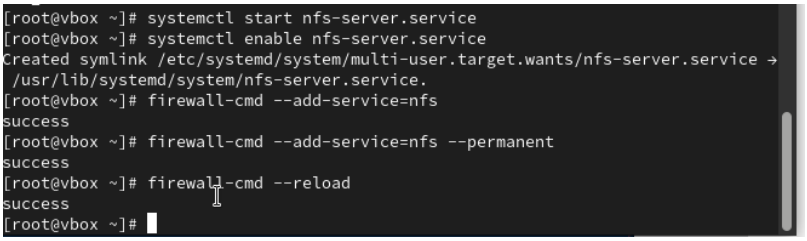
Применим изменённую настройку SELinux к файловой системе: `restorecon -vR /srv/nfs`

Запустим сервер NFS:

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

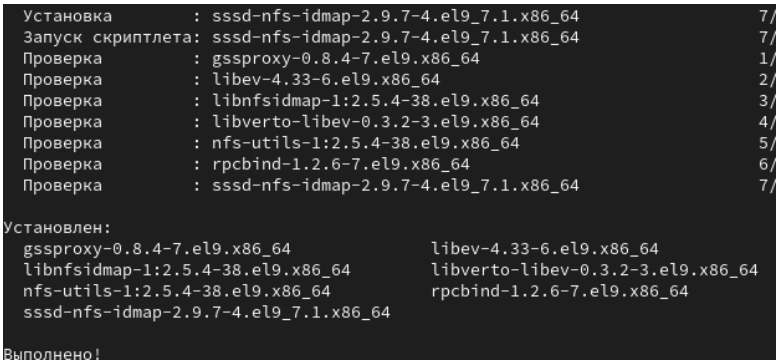
Настроим межсетевой экран для работы сервера NFS:

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




*Настройка межсетевого экрана*

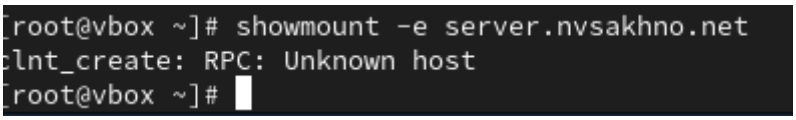
На клиенте установим необходимое для работы NFS программное обеспечение: `dnf -y install nfs-utils`




*Установка пакетов*

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

`showmount -e server.nvsakhno.net`




*Просмотр подмонтированных удаленных ресурсов*

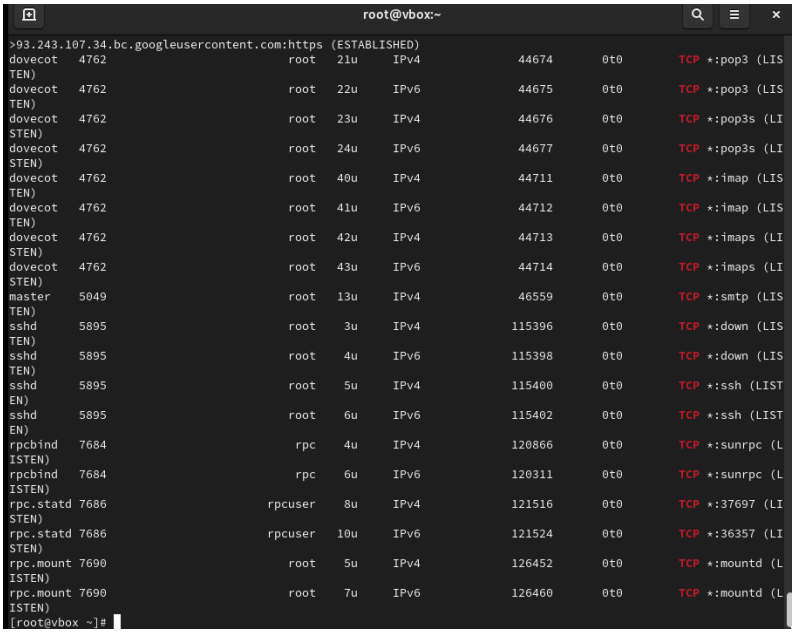
Попробуем на сервере остановить сервис межсетевого экрана: `systemctl stop firewalld.service`

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

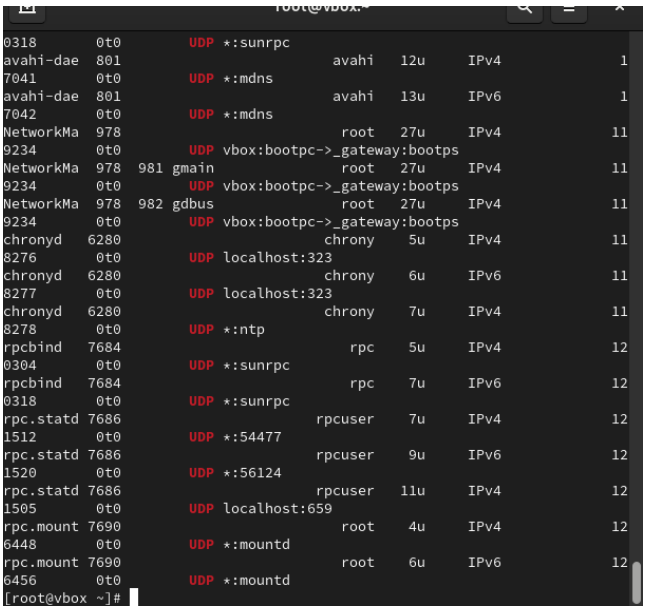
На сервере запустим сервис межсетевого экрана `systemctl start firewalld`

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

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




*Задействованные службы при удаленном монтировании по протоколу TCP*




*Задействованные службы при удаленном монтировании по протоколу UDP*

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
```


*Настройка межсетевого экрана на сервере*

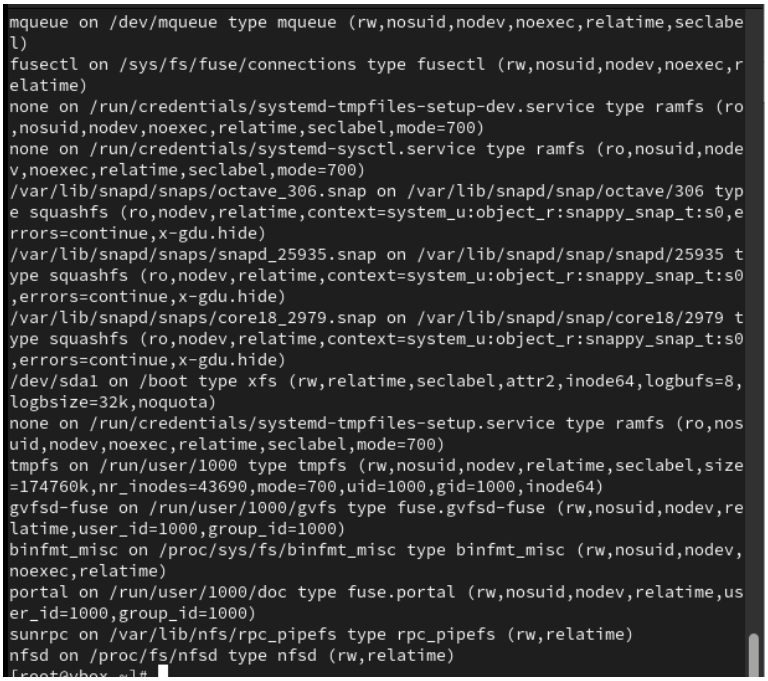
На клиенте проверим подключение удалённого ресурса: `showmount -e server.nvsakhno.net`

## 3.2 Монтирование NFS на клиенте

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

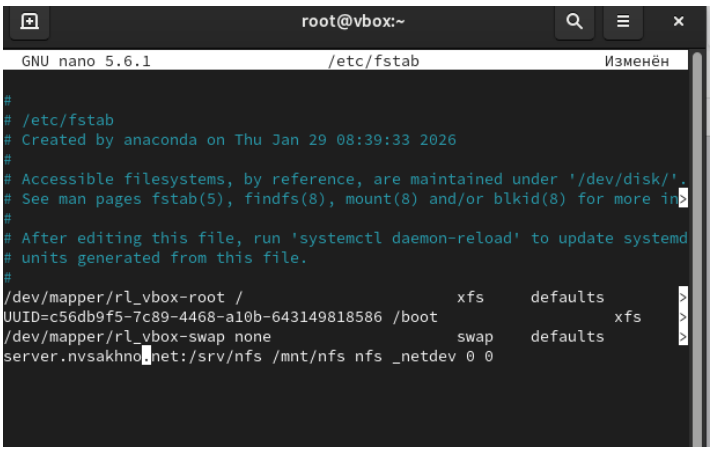
```
mkdir -p /mnt/nfs
mount server.nvsakhno.net:/srv/nfs /mnt/nfs
```

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



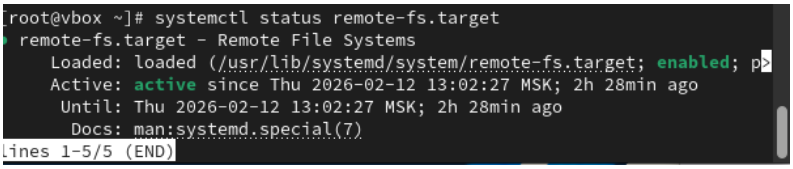

*Монтирование NFS на клиенте*

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




*Редактирование файла*

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




*Проверка наличия автоматического монтирования удалённых ресурсов*

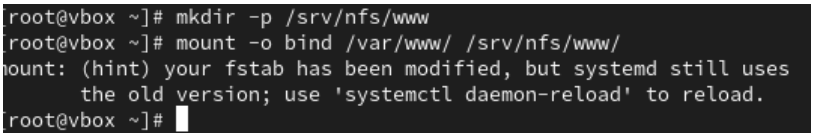
Перезапустим клиент и убедимся, что удалённый ресурс подключается автоматически.

## 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`

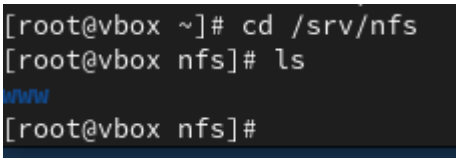
Подмонтируем каталог web-сервера: `mount -o bind /var/www/ /srv/nfs/www/`

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



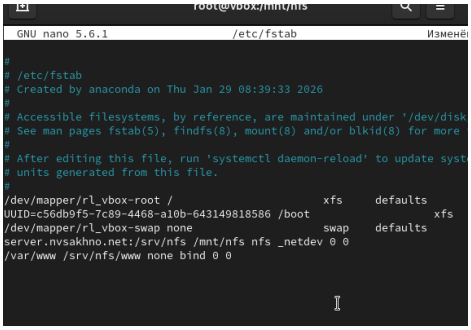

*Содержимое каталога*

На клиенте посмотрим, что отображается в каталоге /mnt/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)`




*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
```

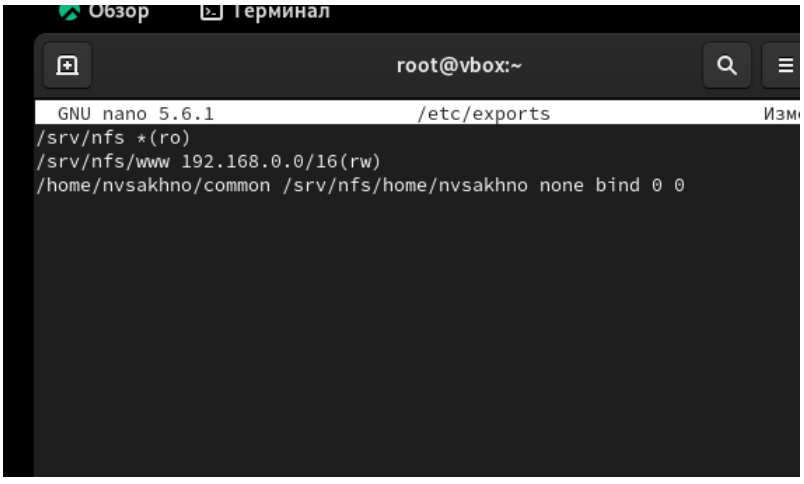
На сервере создадим общий каталог для работы пользователя nvsakhno по сети: `mkdir -p /srv/nfs/home/user`

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)`




*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

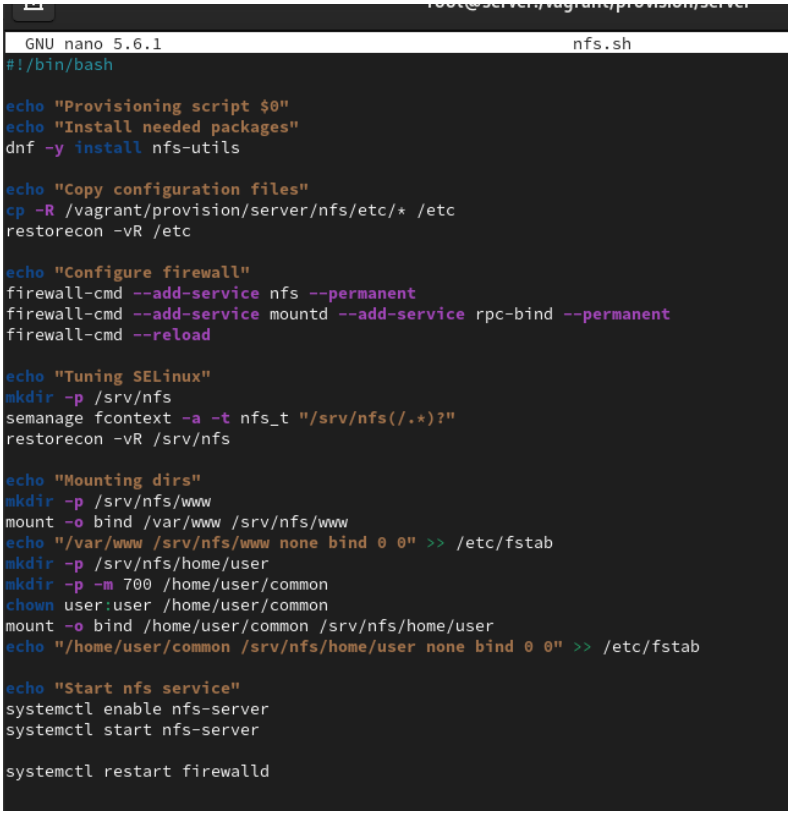
На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог nfs, в который поместим в соответствующие подкаталоги конфигурационные файлы:

```
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:




*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:


*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.