

# Цель работы

Приобрести навыки настройки сервера NFS для удалённого доступа к ресурсам.

## 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.

## Выполнение лабораторной работы

### Настройка сервера NFSv4

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

```
[root@server.tbmanturov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
[          ]      ===
[          ]      130 kB/s | 39 kB   00:00
[          ]      3.3 MB/s | 5.5 MB  00:01
[          ]      7.3 kB/s | 4.3 kB  00:00
[          ]      10 kB/s | 4.3 kB  00:00
[          ]      8.1 kB/s | 4.3 kB  00:00
[          ]      --- B/s | 0 B    --- ETA
```

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

## Настройка межсетевого экрана

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

```
[root@client.tbmanturov.net ~]# dnf -y install n
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
Rocky Linux 10 - BaseOS
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
100% [==
```

## Установка пакетов

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

```
showmount -e server.tbmanturov.net
```

```
Complete!
[root@client.tbmanturov.net ~]# showmount -e server.tbmanturov.net
clnt_create: RPC: Unknown host
```

## 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.tbmanturov.net`

```
Complete!
[root@client.tbmanturov.net ~]# showmount -e server.tbmanturov.net
clnt_create: RPC: Unknown host
```

## Подключение к удаленно смонтированному ресурсу

На сервере запустим сервис межсетевого экрана `systemctl start firewalld`

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

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

```
[root@server.tbmanturov.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.
```

## Задействованные службы при удаленном монтировании по протоколу TCP

root@server:~ - sudo -i							
named	1339	1344	isc-net-0	named	40u	IPv6	10977
named	1339	1345	isc-timer	named	25u	IPv4	10964
named	1339	1345	isc-timer	named	26u	IPv4	10965
named	1339	1345	isc-timer	named	31u	IPv4	10968
.net:domain							
named	1339	1345	isc-timer	named	32u	IPv4	10969
.net:domain							
named	1339	1345	isc-timer	named	35u	IPv4	10972
.net:domain							
named	1339	1345	isc-timer	named	36u	IPv4	10973
.net:domain							
named	1339	1345	isc-timer	named	39u	IPv6	10976
named	1339	1345	isc-timer	named	40u	IPv6	10977
named	1339	2596	libuv-wor	named	25u	IPv4	10964
named	1339	2596	libuv-wor	named	26u	IPv4	10965
named	1339	2596	libuv-wor	named	31u	IPv4	10968
.net:domain							
named	1339	2596	libuv-wor	named	32u	IPv4	10969
.net:domain							
named	1339	2596	libuv-wor	named	35u	IPv4	10972
.net:domain							
named	1339	2596	libuv-wor	named	36u	IPv4	10973
.net:domain							
named	1339	2596	libuv-wor	named	39u	IPv6	10976
named	1339	2596	libuv-wor	named	40u	IPv6	10977
named	1339	2597	libuv-wor	named	25u	IPv4	10964
named	1339	2597	libuv-wor	named	26u	IPv4	10965
named	1339	2597	libuv-wor	named	31u	IPv4	10968
.net:domain							
named	1339	2597	libuv-wor	named	32u	IPv4	10969
.net:domain							
named	1339	2597	libuv-wor	named	35u	IPv4	10972
.net:domain							
named	1339	2597	libuv-wor	named	36u	IPv4	10973
.net:domain							
named	1339	2597	libuv-wor	named	39u	IPv6	10976

Задействованные службы при удаленном мониторинге по протоколу UDP

Добавим службы `grpc-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
```

```
[root@server.tbmanturov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind  
success  
[root@server.tbmanturov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent  
success  
[root@server.tbmanturov.net ~]# firewall-cmd --reload  
success
```

## Настройка межсетевого экрана на сервере

## Монтирование NFS на клиенте

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

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

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

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

## Монтирование NFS на клиенте

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

```

root@client:~ - sudo -
GNU nano 8.1          /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 11 15:45:56 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=6c6ef144-7f77-44a1-a012-195e8a596169 /           xfs    defaults      0 0
UUID=26c7eaf8-7027-4054-bbc8-fcc5361426d9 /boot        xfs    defaults      0 0
UUID=A1BD-EDF5   /boot/efi      vfat   umask=077,shortname=winnt 0 2
UUID=b57e0f91-3701-4e3f-9512-a73bc4f9b286 /home        xfs    defaults      0 0
UUID=c5a712df-efb8-4a76-b590-c624c0f0787c 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
server.tbmanturov.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@client.tbmanturov.net ~]# systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
  Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled; preset: enabled)
  Active: active since Sat 2025-11-29 10:36:26 UTC; 31min ago
  Invocation: 925adaf8893645b7889ab98ee5302759
  Docs: man:systemd.special(7)
```

### Проверка наличия автоматического монтирования удалённых ресурсов

Перезапустим клиент и убедимся, что удалённый ресурс подключается автоматически.

```
[root@client.tbmanturov.net ~]# showmount -e 192.168.1.1
Export list for 192.168.1.1:
/srv/nfs *
```

### Проверка

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

### Содержимое каталога

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

```
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,noexec)
[root@client.tbmanturov.net ~]# ls /mnt/nfs
www
```

### 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)`

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

Редактирование файла

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

Проверим на клиенте каталог /mnt/nfs.

```
[root@client.tbmanturov.net ~]# ls /mnt/nfs
www
[root@client.tbmanturov.net ~]# ls /mnt/nfs
www
```

Содержимое каталога

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 Tue Nov 11 15:45:56 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=6c6ef144-7f77-44a1-a012-195e8a596169   xfs    defaults      0 0
UUID=26c7eff8-7027-4054-bbc8-fcc5361426d9 /boot   xfs    defaults      0 0
UUID=A1BD-EDF5     /boot/efi   vfat   umask=0077,shortname=winnt 0 2
UUID=b97e0f91-3701-4e3f-9512-a73bc4f9b286 /home   xfs    defaults      0 0
UUID=c5a712df-efb8-4a76-b690-c624c0f0787c 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
```

Редактирование файла

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

На клиенте проверим каталог /mnt/nfs.

```
www
[root@client.tbmanturov.net ~]# ls /mnt/nfs
www
[root@client.tbmanturov.net ~]# ls /mnt/nfs
www
```

Содержимое каталога

## Подключение каталогов для работы пользователей

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

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

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

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

```
[tbmanturov@server.tbmanturov.net ~]$ mkdir -p -m 700 ~/common
[tbmanturov@server.tbmanturov.net ~]$ cd ~/common
[tbmanturov@server.tbmanturov.net common]$ touch tbmanturov@server.txt
[tbmanturov@server.tbmanturov.net 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)`

```
root@server:~ - sudo -i
GNU nano 8.1
/etc/exports
Modified
/srv/nfs *(rw)
/srv/nfs/www 192.168.0.0/16(rw)
/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`.

```
[root@client.tbmanturov.net ~]# ls /mnt/nfs
home  www
[root@client.tbmanturov.net ~]#
```

Проверка содержимого каталога

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
```

```
[tbmanturov@client.tbmanturov.net ~]$ cd /mnt/nfs/home/tbmanturov
[tbmanturov@client.tbmanturov.net tbmanturov]$ sudo touch tbmanturov@client.txt
[sudo] password for tbmanturov:
touch: cannot touch 'tbmanturov@client.txt': Permission denied
[tbmanturov@client.tbmanturov.net tbmanturov]$ touch tbmanturov@client.txt
[tbmanturov@client.tbmanturov.net tbmanturov]$
```

Создание файла

Безуспешно.

Попробуем это проделать под пользователем root.

Безуспешно.

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

Не появились, все тщетно.

## 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:

```
GNU nano 8.1
#!/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 mounted --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 8.1
#!/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.tbmanturov.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"
```

## Выводы

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