

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №13**

*дисциплина: Администрирование сетевых подсистем*

Студент: Бансимба Клодели Дьегра

Студ. билет № 1032215651

Группа: НПИбд-02-22

**МОСКВА**

2024 г.

## Цель работы:

Целью данной работы является приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

## Выполнение работы:

На сервере установим необходимое программное обеспечение (Рис. 1.1):

```
dnf -y install nfs-utils
```

```
[claudely@server.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@server.claudely.net ~]# dnf -y install nfs-utils
Last metadata expiration check: 0:00:47 ago on Sun 22 Dec 2024 12:56:08 PM UTC.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
nfs-utils                              x86_64            1:2.5.4-27.el9     baseos              431 k
Upgrading:
libipa_hbac                            x86_64            2.9.5-4.el9_5.4    baseos              35 k
libsmbclient                           x86_64            4.20.2-2.el9_5     baseos              73 k
libsss_certmap                         x86_64            2.9.5-4.el9_5.4    baseos              90 k
libsss_idmap                           x86_64            2.9.5-4.el9_5.4    baseos              41 k
libsss_nss_idmap                       x86_64            2.9.5-4.el9_5.4    baseos              45 k
libsss_sudo                            x86_64            2.9.5-4.el9_5.4    baseos              35 k
libwbclient                            x86_64            4.20.2-2.el9_5     baseos              42 k
samba-client-libs                      x86_64            4.20.2-2.el9_5     baseos              5.2 M
samba-common                           noarch            4.20.2-2.el9_5     baseos              168 k
samba-common-libs                      x86_64            4.20.2-2.el9_5     baseos              100 k
sssd                                    x86_64            2.9.5-4.el9_5.4    baseos              27 k
sssd-ad                                x86_64            2.9.5-4.el9_5.4    baseos              215 k
sssd-client                            x86_64            2.9.5-4.el9_5.4    baseos              161 k
sssd-common                            x86_64            2.9.5-4.el9_5.4    baseos              1.6 M
sssd-common-pac                        x86_64            2.9.5-4.el9_5.4    baseos              96 k
=====
```

**Рис. 1.1.** Установка на сервере программного обеспечения nfs-utils.

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS) (Рис. 1.2):

```
mkdir -p /srv/nfs
```

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

**Рис. 1.2.** Создание на сервере каталога, который предполагается сделать доступным всем пользователям сети (корень дерева NFS).

В файле `/etc/exports` пропишем подключаемый через NFS общий каталог с доступом только на чтение (Рис. 1.3):

```
/srv/nfs *(ro)
```



**Рис. 1.3.** Прописывание в файле `/etc/exports` подключаемого через NFS общего каталога с доступом только на чтение.

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 (Рис. 1.4):

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

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

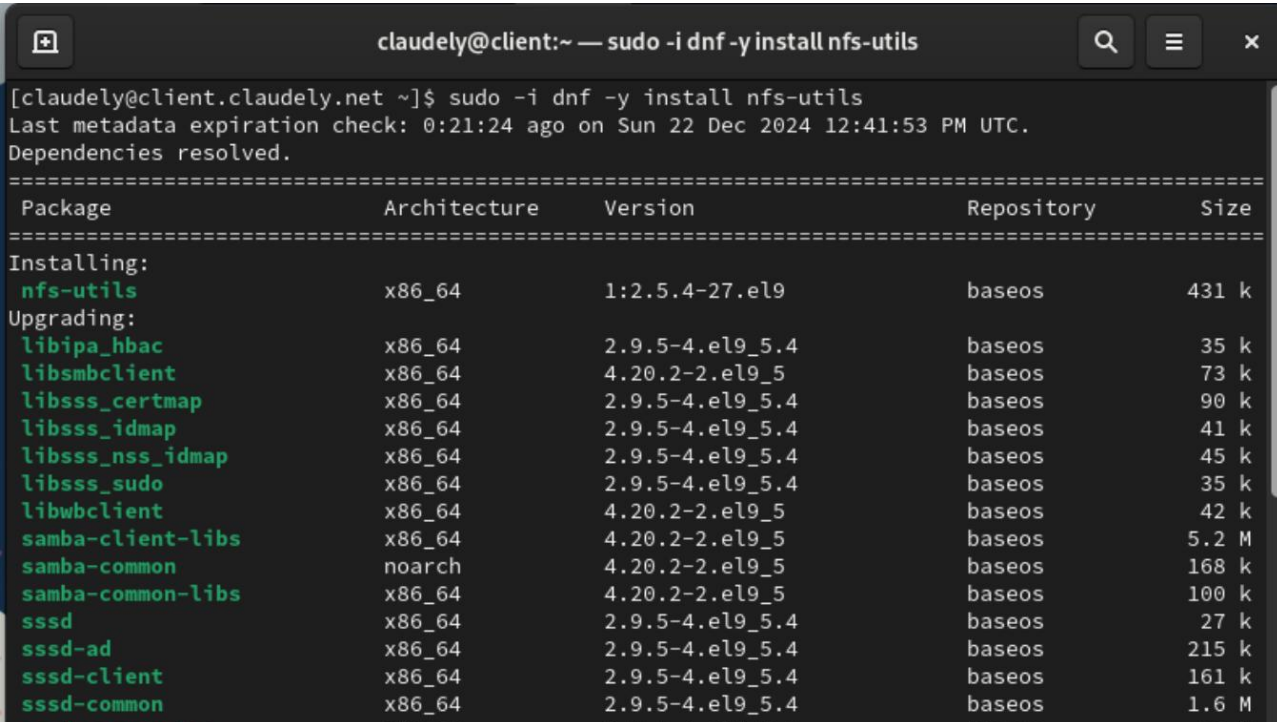
```
firewall-cmd --reload
```

```
[root@server.claudely.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.claudely.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.claudely.net ~]# systemctl start nfs-server.service
[root@server.claudely.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.claudely.net ~]#
[root@server.claudely.net ~]# firewall-cmd --add-service=nfs
bash: firewall-cmd: command not found...
[root@server.claudely.net ~]# firewall-cmd --add-service=nfs
success
[root@server.claudely.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.claudely.net ~]# firewall-cmd --reload
success
[root@server.claudely.net ~]#
```

**Рис. 1.4.** Настройка для общего каталога контекста безопасности NFS, применение изменённой настройки SELinux к файловой системе, запуск сервера NFS и настройка межсетевого экрана для работы сервера NFS.

На клиенте установим необходимое для работы NFS программное обеспечение (Рис. 1.5):

```
dnf -y install nfs-utils
```

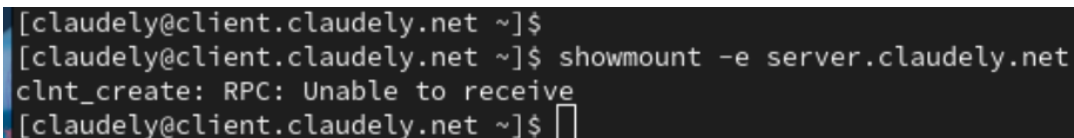


```
claudey@client:~ — sudo -i dnf -y install nfs-utils
[claudey@client.claudely.net ~]$ sudo -i dnf -y install nfs-utils
Last metadata expiration check: 0:21:24 ago on Sun 22 Dec 2024 12:41:53 PM UTC.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
nfs-utils                              x86_64            1:2.5.4-27.el9   baseos            431 k
Upgrading:
libipa_hbac                            x86_64            2.9.5-4.el9_5.4  baseos            35 k
libsmbclient                          x86_64            4.20.2-2.el9_5   baseos            73 k
libsss_certmap                        x86_64            2.9.5-4.el9_5.4  baseos            90 k
libsss_idmap                          x86_64            2.9.5-4.el9_5.4  baseos            41 k
libsss_nss_idmap                      x86_64            2.9.5-4.el9_5.4  baseos            45 k
libsss_sudo                          x86_64            2.9.5-4.el9_5.4  baseos            35 k
libwbclient                          x86_64            4.20.2-2.el9_5   baseos            42 k
samba-client-libs                     x86_64            4.20.2-2.el9_5   baseos            5.2 M
samba-common                          noarch            4.20.2-2.el9_5   baseos            168 k
samba-common-libs                     x86_64            4.20.2-2.el9_5   baseos            100 k
sssd                                   x86_64            2.9.5-4.el9_5.4  baseos            27 k
sssd-ad                              x86_64            2.9.5-4.el9_5.4  baseos            215 k
sssd-client                          x86_64            2.9.5-4.el9_5.4  baseos            161 k
sssd-common                          x86_64            2.9.5-4.el9_5.4  baseos            1.6 M
```

**Рис. 1.5.** Установка на клиенте программного обеспечения для работы NFS.

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы (Рис. 1.6):

```
showmount -e server.claudely.net
```

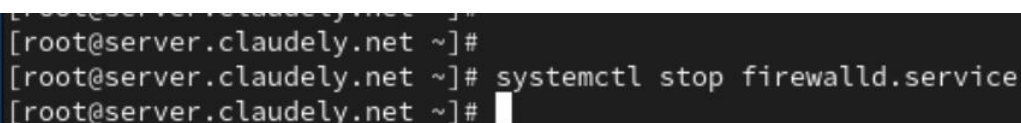


```
[claudely@client.claudely.net ~]$  
[claudely@client.claudely.net ~]$ showmount -e server.claudely.net  
clnt_create: RPC: Unable to receive  
[claudely@client.claudely.net ~]$
```

**Рис. 1.6.** Попытка посмотреть на клиенте имеющихся подмонтированных удалённых ресурсов.

Попробуем на сервере остановить сервис межсетевого экрана (Рис. 1.7):

```
systemctl stop firewalld.service
```



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

**Рис. 1.7.** Попытка остановить на сервере сервис межсетевого экрана.

На сервере запустим сервис межсетевого экрана (Рис. 1.9):

```
systemctl start firewalld
```



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

**Рис. 1.9.** Запуск на сервере сервиса межсетевого экрана.

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

```
lsof | grep TCP (Рис. 1.10):
```

```
lsof | grep UDP (Рис. 1.11):
```

```
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# systemctl start firewalld
[root@server.claudely.net ~]# lsof | grep TCP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1002/gvfs
Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1002/doc
Output information may be incomplete.
systemd      1          root    250u      IPv4        117244      0t0      TCP *:sunrpc
(LISTEN)
systemd      1          root    252u      IPv6        117262      0t0      TCP *:sunrpc
(LISTEN)
cupsd        783        root     6u       IPv6        20625       0t0      TCP localhost
:ipp (LISTEN)
cupsd        783        root     7u       IPv4        20626       0t0      TCP localhost
:ipp (LISTEN)
sshd         798        root     3u       IPv4        20713       0t0      TCP *:down (L
ISTEN)
sshd         798        root     4u       IPv6        20733       0t0      TCP *:down (L
```

**Рис. 1.10.** Просмотр на сервере служб, задействованных при удалённом монтировании (TCP).

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
```

```
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 b
acula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet
bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-r
egistry docker-docker-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa
-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gp
sd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-tar
get isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-con
trol-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-servi
ces kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt
libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna
mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 n
mea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmpoxy pmwebapi pmw
ebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulse
audio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba s
amba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-
lansync spotify-sync squid sssd ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui syncthing-rela
y synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsd vnc
-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discover
y-udp wsmann wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.claudely.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.claudely.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.claudely.net ~]# firewall-cmd --reload
success
[root@server.claudely.net ~]#
```



**Рис. 1.12.** Добавление службы `rpc-bind` и `mountd` в настройки межсетевого экрана на сервере.

На клиенте создадим каталог, в который будем монтировать удалённый ресурс, и подмонтируем дерево NFS. После чего проверим, что общий ресурс NFS подключён правильно (Рис. 2.1):

```
[claudely@client.claudely.net ~]$ mkdir -p /mnt/nfs
mkdir: cannot create directory '/mnt/nfs': Permission denied
[claudely@client.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@client.claudely.net ~]# mkdir -p /mnt/nfs
```

**Рис. 2.1.** Создание на клиенте каталога, в который будет монтироваться удалённый ресурс, и монтироваться дерево NFS. Проверка, что общий ресурс NFS подключён правильно.

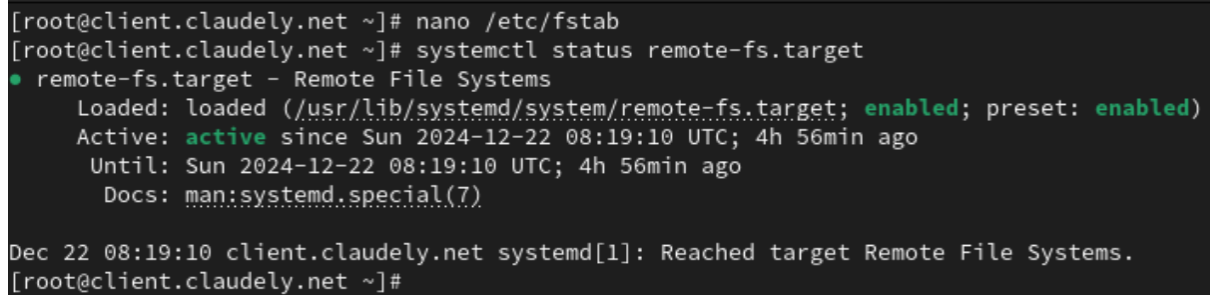
На клиенте в конце файла `/etc/fstab` добавим следующую запись (Рис. 2.2):  
`server.claudely.net:/srv/nfs /mnt/nfs nfs _netdev 0 0`

```
GNU nano 5.6.1 /etc/fstab Modified
#
# /etc/fstab
# Created by anaconda on Tue Sep 10 20:22:55 2024
#
# 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=56c1387a-1ca9-4840-80e7-dc1160461c4d / xfs defaults 0 0
/swapfile 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.user.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

**Рис. 2.2.** Добавление на клиенте в конце файла `/etc/fstab` записи.

На клиенте проверим наличие автоматического монтирования удалённых ресурсов при запуске операционной системы (Рис. 2.3):

```
systemctl status remote-fs.target
```



```
[root@client.claudely.net ~]# nano /etc/fstab
[root@client.claudely.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 Sun 2024-12-22 08:19:10 UTC; 4h 56min ago
     Until: Sun 2024-12-22 08:19:10 UTC; 4h 56min ago
    Docs: man:systemd.special(7)

Dec 22 08:19:10 client.claudely.net systemd[1]: Reached target Remote File Systems.
[root@client.claudely.net ~]#
```

**Рис. 2.3.** Проверка на клиенте наличия автоматического монтирования удалённых ресурсов при запуске операционной системы.

На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера:

```
mkdir -p /srv/nfs/www
```

После чего подмонтируем каталог web-сервера:

```
mount -o bind /var/www/ /srv/nfs/www/
```

И на сервере проверим, что отображается в каталоге /srv/nfs (Рис. 3.1):



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

**Рис. 3.1.** Создание на сервере общего каталога, в который затем будет подмонтирован каталог с контентом веб-сервера. Монтирование каталога web-сервера и проверка, что отображается в каталоге /srv/nfs.

На сервере в файле /etc/exports добавим экспорт каталога веб-сервера с удалённого ресурса (Рис. 3.3):

```
/srv/nfs/www 192.168.0.0/16(rw)
```



```
root@server:~  
GNU nano 5.6.1 /etc/exports  
/srv/nfs *(ro)  
/srv/nfs/www 192.168.0.0/16(rw)
```

**Рис. 3.3.** Добавление на сервере в файле `/etc/exports` экспорта каталога веб-сервера с удалённого ресурса.

На сервере в конце файла `/etc/fstab` добавим следующую запись (Рис. 3.6):

```
/var/www /srv/nfs/www none bind 0 0
```

```
GNU nano 5.6.1 /etc/fstab  
#  
# /etc/fstab  
# Created by anaconda on Tue Sep 10 20:22:55 2024  
#  
# 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=56c1387a-1ca9-4840-80e7-dc1160461c4d / xfs defaults 0 0  
/swapfile 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. 3.6.** Добавление на сервере в конце файла `/etc/fstab` записи.

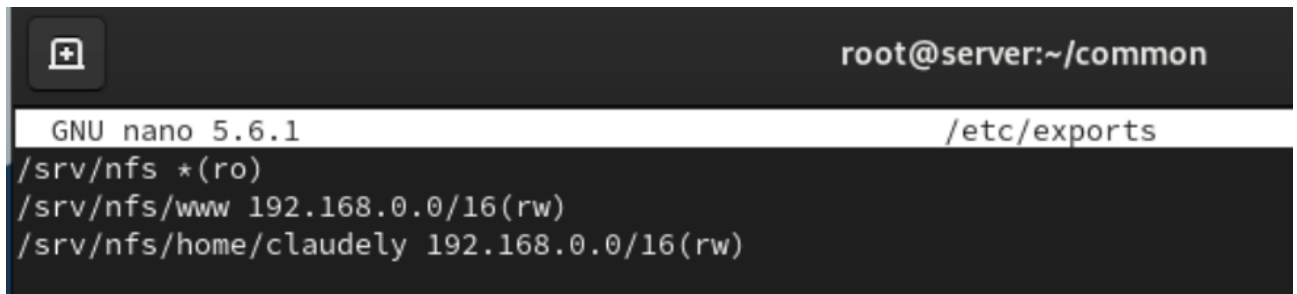
На сервере под пользователем `claudely` в домашнем каталоге создадим каталог `common` с полными правами доступа только для этого пользователя, а в нём файл `claudely@server.txt`. Далее создадим общий каталог для работы пользователя `claudely` по сети и подмонтируем каталог `common` пользователя `claudely` в NFS (Рис. 4.1):

```
[root@server.claudely.net ~]# mkdir -p -m 700 ~/common  
[root@server.claudely.net ~]# cd ~/common  
[root@server.claudely.net common]# touch claudely@server.txt  
[root@server.claudely.net common]# mkdir -p /srv/nfs/home/claudely  
[root@server.claudely.net common]# mount -o bind /home/user/common /srv/nfs/home/claudely
```

**Рис. 4.1.** Создание на сервере под пользователем claudely в домашнем каталоге каталога common с полными правами доступа только для этого пользователя, а в нём файл claudely@server.txt. Создание общего каталога для работы пользователя claudely по сети и монтирование каталога common пользователя claudely в NFS.

Подключим каталог пользователя в файле /etc/exports, прописав в нём (Рис. 4.2):

```
/srv/nfs/home/claudely 192.168.0.0/16(rw)
```

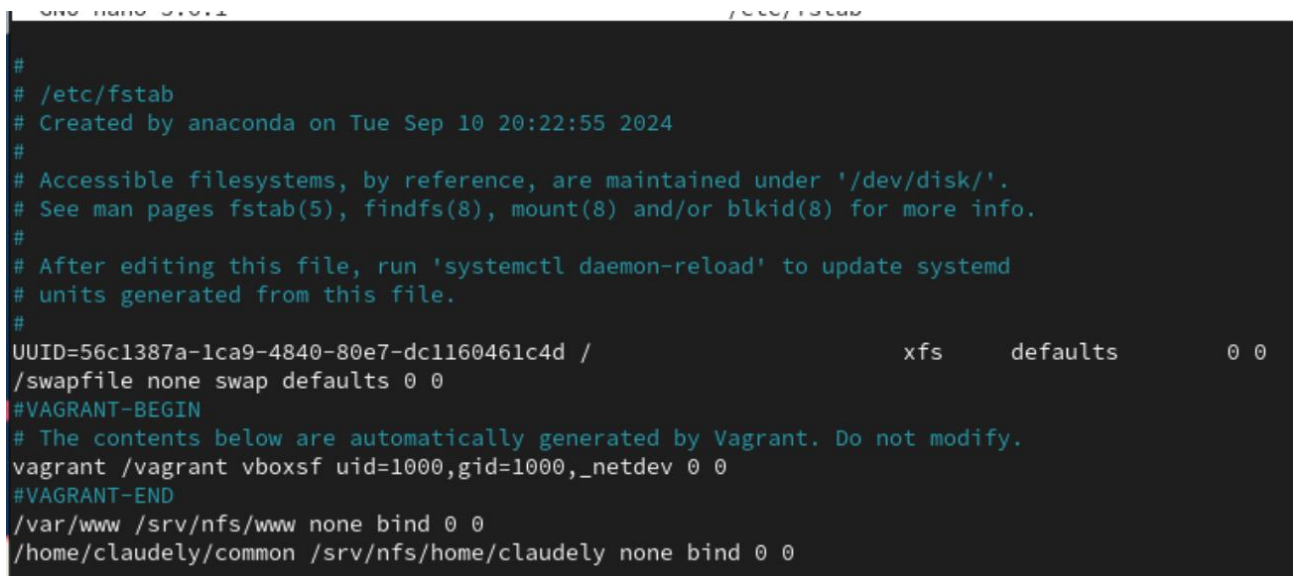


```
root@server:~/common
GNU nano 5.6.1 /etc/exports
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/claudely 192.168.0.0/16(rw)
```

**Рис. 4.2.** Подключение каталога пользователя в файле /etc/exports.

Внесём изменения в файл /etc/fstab (Рис. 4.3):

```
/home/claudely/common /srv/nfs/home/claudely none bind 0 0
```



```
#
# /etc/fstab
# Created by anaconda on Tue Sep 10 20:22:55 2024
#
# 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=56c1387a-1ca9-4840-80e7-dc1160461c4d / xfs defaults 0 0
/swapfile 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
/home/claudely/common /srv/nfs/home/claudely none bind 0 0
```

**Рис. 4.3.** Внесение изменения в файл /etc/fstab.

На клиенте под пользователем claudely перейдём в каталог /mnt/nfs/home/claudey и попробуем создать в нём файл claudely@client.txt и внести в него какие-либо изменения (Рис. 4.6). Попробуем также это проделать под пользователем root (Рис. 4.7):

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

**Рис. 4.6.** Переход на клиенте под пользователем claudely в каталог /mnt/nfs/home/claudey и попытка создать в нём файл claudely@client.txt.

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

**Рис. 5.1.** Переход на виртуальной машине server в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога nfs, в который помещаем в соответствующие подкаталоги конфигурационные файлы. Создание в каталоге /vagrant/provision/server исполняемого файла nfs.sh.

Открываем его на редактирование и пропишем в нём следующий скрипт из лабораторной работы (Рис. 5.2):

```
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
```

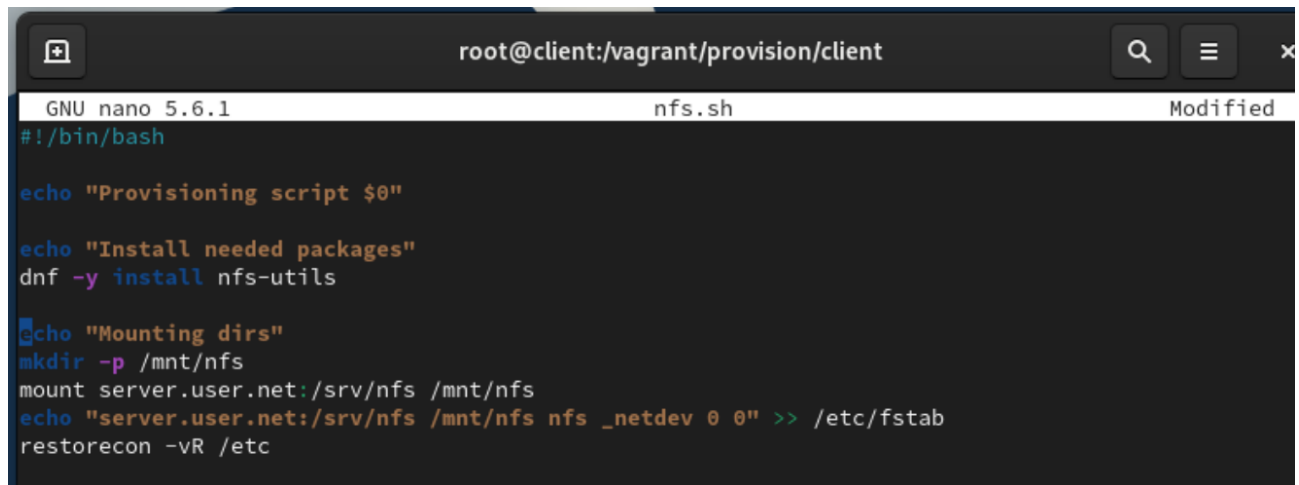
**Рис. 5.2.** Открытие файла на редактирование и добавление скрипта.

После чего на виртуальной машине client перейдём в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`. В каталоге `/vagrant/provision/client` создадим исполняемый файл `nfs.sh` (Рис. 5.3):

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

**Рис. 5.3.** Переход на виртуальной машине client в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`.  
Создание в каталоге `/vagrant/provision/client` исполняемого файла `nfs.sh`.

Открываем его на редактирование и пропишем в нём следующий скрипт (Рис. 5.4):



```
root@client:/vagrant/provision/client
GNU nano 5.6.1      nfs.sh      Modified
#!/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
```

**Рис. 5.4.** Открытие файла на редактирование и добавление скрипта.

Для отработки созданных скриптов во время загрузки виртуальных машин `server` и `client` в конфигурационном файле `Vagrantfile` добавим в соответствующих разделах конфигураций для сервера (Рис. 5.5) и клиента (Рис. 5.6):

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

**Рис. 5.5.** Добавление записи в конфигурационном файле `Vagrantfile` для сервера.

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

**Рис. 5.6.** Добавление записи в конфигурационном файле Vagrantfile для клиента.

### **Вывод:**

В ходе выполнения лабораторной работы были приобретены навыки настройки сервера NFS для удалённого доступа к ресурсам.

### **Ответы на контрольные вопросы:**

1. Как называется файл конфигурации, содержащий общие ресурсы NFS?  
- **Файл конфигурации, содержащий общие ресурсы NFS, называется /etc/exports. В этом файле определяются каталоги, которые будут доступны для общего использования через NFS.**
2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS? - **Для обеспечения полного доступа к серверу NFS, обычно открываются следующие порты:**

**TCP и UDP порт 2049: Основной порт для NFS.**

**TCP и UDP порт 111: Порт для службы rpcbind (или портmap), которая используется для регистрации служб RPC.**

**Порты для динамически выделяемых портов (обычно в диапазоне 32768-32779), используемых NFS для передачи данных.**



3. Какую опцию следует использовать в `/etc/fstab`, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке? - Для автоматической установки общих ресурсов NFS при загрузке системы, в файле `/etc/fstab` следует использовать опцию `auto`. Пример строки в `/etc/fstab`:

```
server:/remote/export /mnt/nfs_share nfs auto 0 0
```

Здесь `server:/remote/export` - адрес и путь к общему ресурсу NFS, `/mnt/nfs_share` - точка монтирования, `nfs` - тип файловой системы, `auto` - опция для автоматического монтирования при загрузке системы, `0 0` - опции для процесса проверки файловой системы при загрузке.