

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: Жаворонков Кирилл Александрович

Студ. билет № 1132231844

Группа: НПИбд-01-23

МОСКВА

2025 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
```

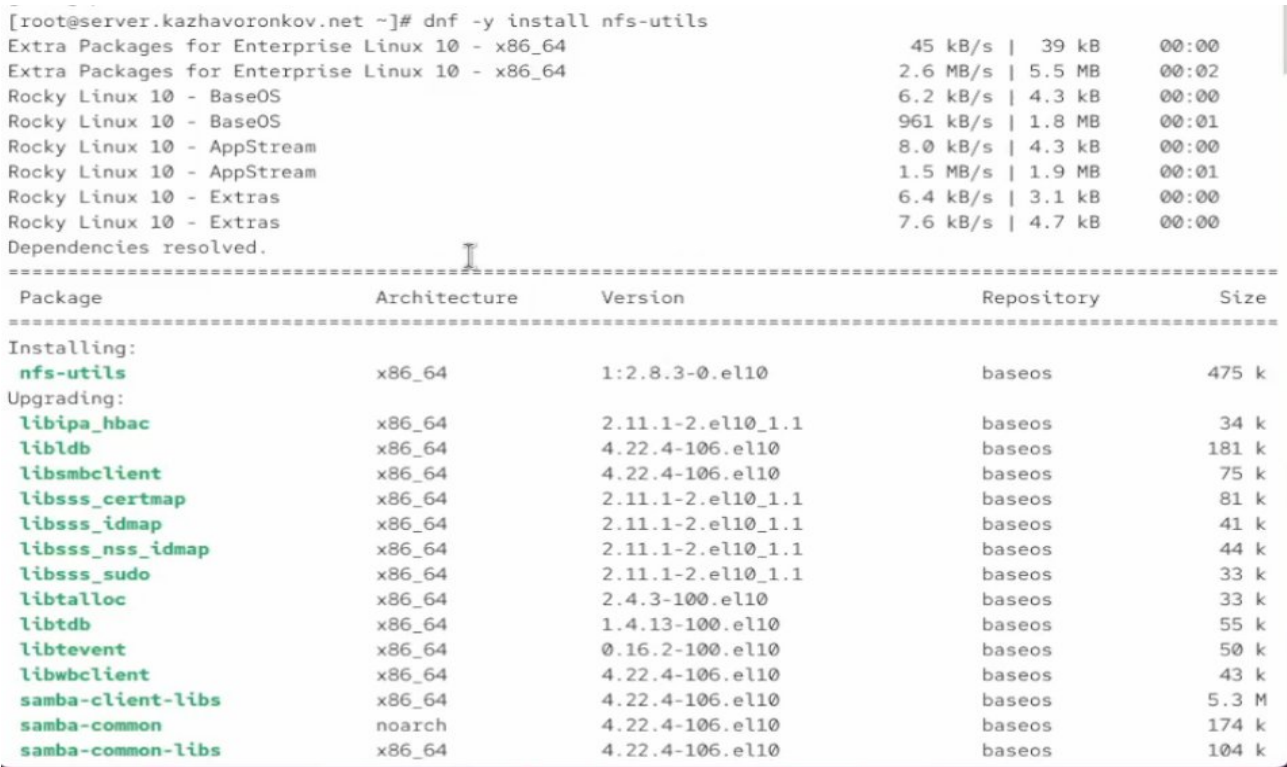


Рис. 1.1. Установка на сервере программного обеспечения nfs-utils.

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

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

```
Complete!  
[root@server.kazhavoronkov.net ~]# mkdir -p /srv/nfs  
[root@server.kazhavoronkov.net ~]# nan
```

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.kazhavoronkov.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.kazhavoronkov.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.kazhavoronkov.net ~]# systemctl start nfs-server.service
[root@server.kazhavoronkov.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.kazhavoronkov.net ~]# firewall-cmd --add-service=nfs
success
[root@server.kazhavoronkov.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.kazhavoronkov.net ~]# firewall-cmd --reload
success
[root@server.kazhavoronkov.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
```

```
root@client:~ - sudo -i
[kazhavoronkov@client.kazhavoronkov.net ~]$ sudo -i
[sudo] password for kazhavoronkov:
Sorry, try again.
[sudo] password for kazhavoronkov:
[root@client.kazhavoronkov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64 48 kB/s | 39 kB 00:00
Extra Packages for Enterprise Linux 10 - x86_64 1.1 MB/s | 5.5 MB 00:05
Rocky Linux 10 - BaseOS 7.4 kB/s | 4.3 kB 00:00
Rocky Linux 10 - BaseOS 859 kB/s | 1.8 MB 00:02
Rocky Linux 10 - AppStream 6.8 kB/s | 4.3 kB 00:00
Rocky Linux 10 - AppStream 1.1 MB/s | 1.9 MB 00:01
Rocky Linux 10 - Extras 5.3 kB/s | 3.1 kB 00:00
Rocky Linux 10 - Extras 5.0 kB/s | 4.7 kB 00:00
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
nfs-utils x86_64 1:2.8.3-0.el10 baseos 475 k
Upgrading:
libipa_hbac x86_64 2.11.1-2.el10_1.1 baseos 34 k
libldb x86_64 4.22.4-106.el10 baseos 181 k
libsmbclient x86_64 4.22.4-106.el10 baseos 75 k
libsss_certmap x86_64 2.11.1-2.el10_1.1 baseos 81 k
libsss_idmap x86_64 2.11.1-2.el10_1.1 baseos 41 k
libsss_nss_idmap x86_64 2.11.1-2.el10_1.1 baseos 44 k
```

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.kazhavoronkov.net
```

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

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

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

Затем на клиенте вновь попробуем подключиться к удалённо смонтированному ресурсу (Рис. 1.8):

```
showmount -e server.kazhavoronkov.net
```

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

Рис. 1.8. Повторная попытка подключения к удалённо смонтированному ресурсу.

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

```
systemctl start firewalld
```

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

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.kazhavoronkov.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  265u    IPv6        7787      0t0      TCP *:web
sm (LISTEN)
systemd      1                root  305u    IPv4       66390      0t0      TCP *:sun
rpc (LISTEN)
systemd      1                root  307u    IPv6       64678      0t0      TCP *:sun
rpc (LISTEN)
cupsd       1088             root    7u     IPv6        8909      0t0      TCP local
host:ipp (LISTEN)
cupsd       1088             root    8u     IPv4        8910      0t0      TCP local
host:ipp (LISTEN)
sshd        1093             root    7u     IPv4        8886      0t0      TCP *:dow
n (LISTEN)
sshd        1093             root    8u     IPv6        8888      0t0      TCP *:dow
n (LISTEN)
sshd        1093             root    9u     IPv4        8890      0t0      TCP *:ssh
(LISTEN)
sshd        1093             root   10u     IPv6        8892      0t0      TCP *:ssh
(LISTEN)
^C
[root@server.kazhavoronkov.net ~]# lsof | grep
```

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

```
[root@server.kazhavoronkov.net ~]# lsof | grep UDP
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  306u    IPv4       64671      0t0      UDP *:sun
rpc
systemd      1                root  308u    IPv6       64685      0t0      UDP *:sun
rpc
avahi-dae   698             avahi   12u    IPv4        8301      0t0      UDP *:mdn
s
avahi-dae   698             avahi   13u    IPv6        8302      0t0      UDP *:mdn
s
chronyd     740             chrony    5u    IPv4        8135      0t0      UDP local
host:323
chronyd     740             chrony    6u    IPv6        8136      0t0      UDP local
host:323
chronyd     740             chrony    7u    IPv4        8137      0t0      UDP *:ntp
^C
[root@server.kazhavoronkov.net ~]#
```


Рис. 1.11. Просмотр на сервере служб, задействованных при удалённом монтировании (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
```

```
~
[root@server.kazhavoronkov.net ~]# firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 ann
o-1800 apcupsd aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-stor
age bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-m
on cfengine checkmk-agent civilization-iv civilization-v cockpit collectd condor-collector cratedb ctdb dds
dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-quic dns-over-tls docker-registry
docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server factorio finger foreman foreman-proxy fr
eeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-mast
er git gpsd grafana gre high-availability http http3 https ident imap imaps iperf2 iperf3 ipfs ipp ipp-clie
nt ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell k
ube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller
-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-rea
donly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-ud
p managesieve matrix mdns memcache minecraft minidlna mndp mongodb mosh mountd mpd mqtt mqtt-tls ms-wbt mss
ql murmur mysql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe
ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmpoxy pmwebapi
pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv p
tp pulseaudio puppetmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd
rtsp salt-master samba samba-client samba-dc sane settlers-history-collection sip sips slimevr slp smtp sm
tp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-cu
stom statsrv steam-lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submission supertu
xkart svdrp svn syncthing syncthing-gui syncthing-relay synergy syscomlan syslog syslog-tls telnet tentacle
terraria tftp tile38 tinc tor-socks transmission-client turn turns upnp-client vdsms vnc-server vrrp warpin
ator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-host ws-discovery-tcp ws-
discovery-udp wsdd wsdd-http wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent z
abbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.kazhavoronkov.net ~]# firewall-cmd --add-serv
```

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

На клиенте проверим подключение удалённого ресурса (Рис. 1.13):

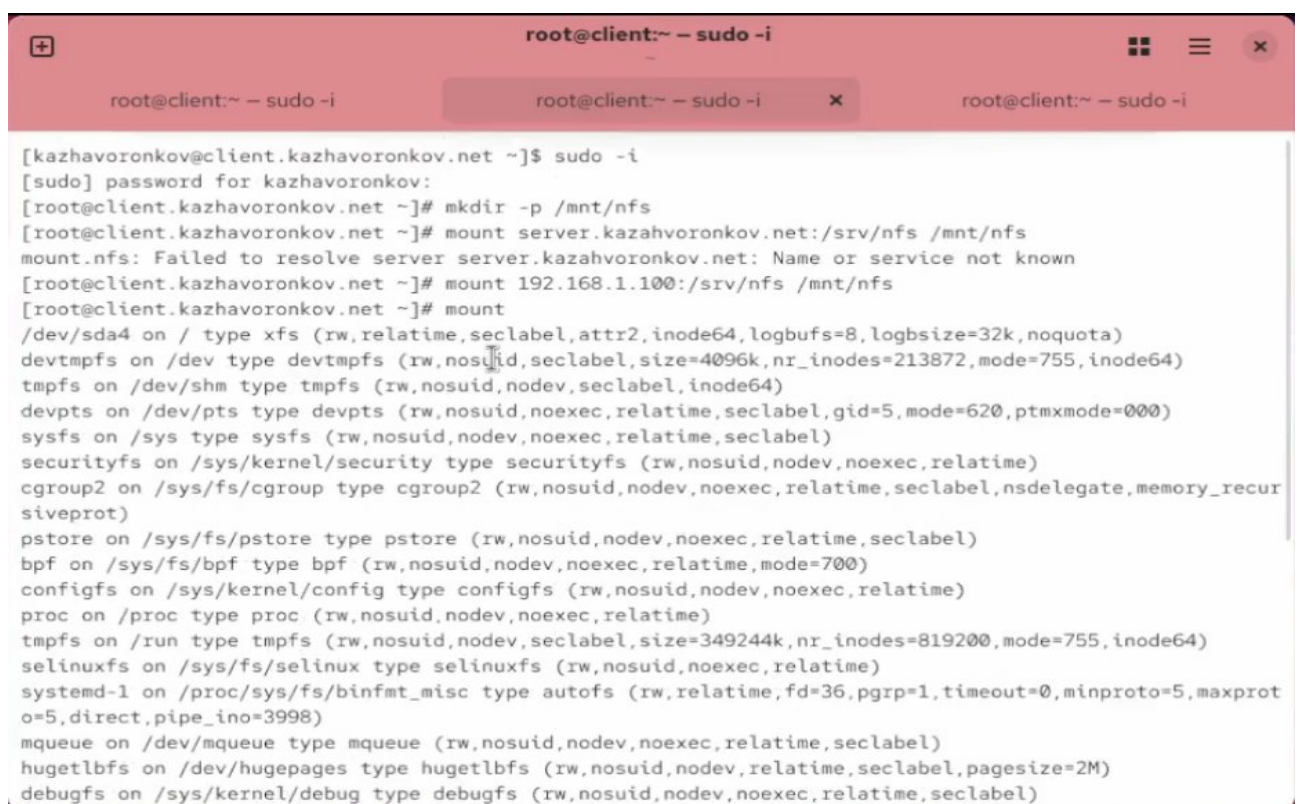
```
showmount -e server.kazhavoronkov.net
```



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

Рис. 1.13. Проверка на клиенте подключения удалённого ресурса.

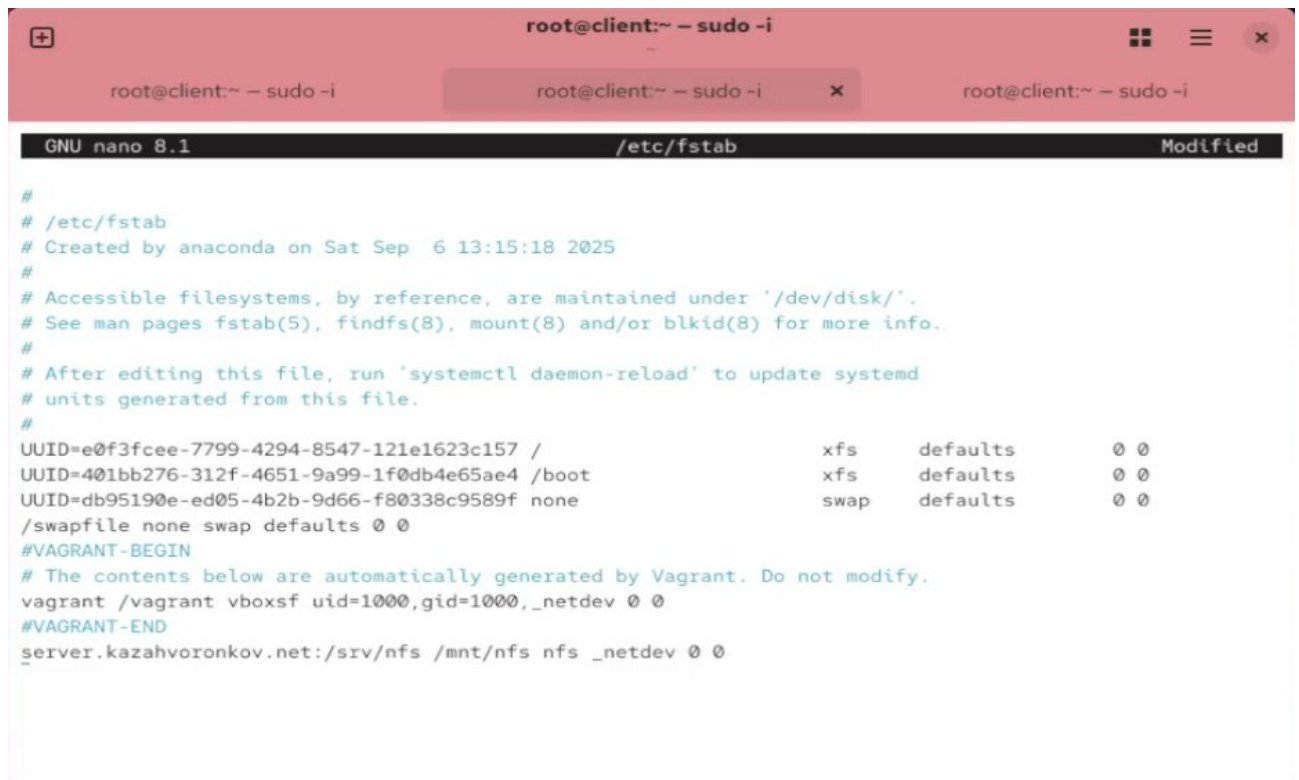
На клиенте создадим каталог, в который будем монтировать удалённый ресурс, и подмонтируем дерево NFS. После чего проверим, что общий ресурс NFS подключён правильно (Рис. 2.1):



```
root@client:~ -- sudo -i
[kazhavoronkov@client.kazhavoronkov.net ~]$ sudo -i
[sudo] password for kazhavoronkov:
[root@client.kazhavoronkov.net ~]# mkdir -p /mnt/nfs
[root@client.kazhavoronkov.net ~]# mount server.kazhavoronkov.net:/srv/nfs /mnt/nfs
mount.nfs: Failed to resolve server server.kazhavoronkov.net: Name or service not known
[root@client.kazhavoronkov.net ~]# mount 192.168.1.100:/srv/nfs /mnt/nfs
[root@client.kazhavoronkov.net ~]# mount
/dev/sda4 on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=213872,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)
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=349244k,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,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=3998)
mqueue on /dev/mqueue type mqueue (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)
```

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.kazhavoronkov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0



```
root@client:~ - sudo -i
GNU nano 8.1 /etc/fstab Modified
#
# /etc/fstab
# Created by anaconda on Sat Sep  6 13:15:18 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=e0f3fcee-7799-4294-8547-121e1623c157 / xfs defaults 0 0
UUID=401bb276-312f-4651-9a99-1f0db4e65ae4 /boot xfs defaults 0 0
UUID=db95190e-ed05-4b2b-9d66-f80338c9589f none swap 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.kazhavoronkov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

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

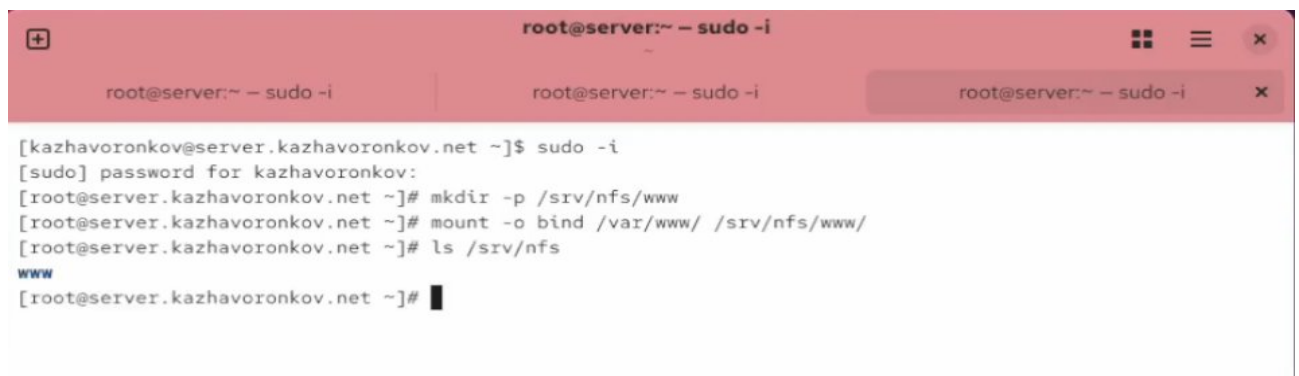
На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера:

```
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:~ - sudo -i
[kazhavoronkov@server.kazhavoronkov.net ~]$ sudo -i
[sudo] password for kazhavoronkov:
[root@server.kazhavoronkov.net ~]# mkdir -p /srv/nfs/www
[root@server.kazhavoronkov.net ~]# mount -o bind /var/www/ /srv/nfs/www/
[root@server.kazhavoronkov.net ~]# ls /srv/nfs
www
[root@server.kazhavoronkov.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 клиенте посмотрим, что отображается в каталоге /mnt/nfs (Рис. 3.2):

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

Рис. 3.2. Просмотр на клиенте содержимого каталога /mnt/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:~ - sudo -i
GNU nano 8.1 /etc/exports Modified
/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.

Экспортируем все каталоги, упомянутые в файле /etc/exports (Рис. 3.4):

exportfs -r

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

Рис. 3.4. Экспорт всех каталогов, упомянутых в файле /etc/exports.

Проверим на клиенте каталог /mnt/nfs (Рис. 3.5):

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

Рис. 3.5. Проверка на клиенте каталога /mnt/nfs.

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

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



Рис. 3.6. Добавление на сервере в конце файла /etc/fstab записи.

Повторно экспортируем каталоги, указанные в файле /etc/exports (Рис. 3.7):

exportfs -r

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

Рис. 3.7. Повторный экспорт каталогов, указанных в файле /etc/exports.

На клиенте проверим каталог /mnt/nfs (Рис. 3.8):

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

Рис. 3.8. Проверка на клиенте каталога /mnt/nfs.

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

```
[kazhavoronkov@server.kazhavoronkov.net ~]$ mkdir -p -m 700 ~/common
[kazhavoronkov@server.kazhavoronkov.net ~]$ cd ~/common
[kazhavoronkov@server.kazhavoronkov.net common]$ touch kazhavoronkov@server.txt
[kazhavoronkov@server.kazhavoronkov.net common]$ mkdir -p /srv/nfs/home/kazhavoronkov
mkdir: cannot create directory '/srv/nfs/home': Permission denied
[kazhavoronkov@server.kazhavoronkov.net common]$ sudo -i
[sudo] password for kazhavoronkov:
[root@server.kazhavoronkov.net ~]# mkdir -p /srv/nfs/home/kazhavoronkov
[root@server.kazhavoronkov.net ~]# mount -o bind /home/kazhavoronkov/common /srv/nfs/home/kazhavoronkov
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@server.kazhavoronkov.net ~]#
```

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

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

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

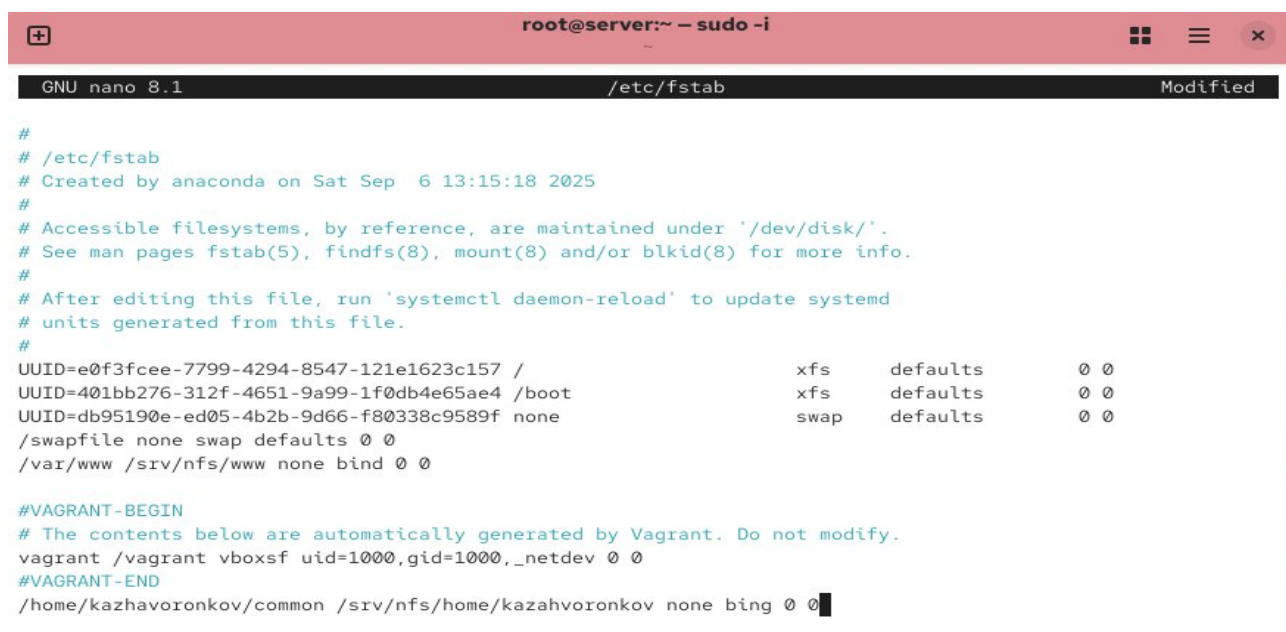


```
root@server:~ - sudo -i
GNU nano 8.1 /etc/exports Modified
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/kazhavoronkov 192.168.0.0/16(rw)
```

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

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

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



```
root@server:~ - sudo -i
GNU nano 8.1 /etc/fstab Modified
#
# /etc/fstab
# Created by anaconda on Sat Sep  6 13:15:18 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=e0f3fcee-7799-4294-8547-121e1623c157 / xfs defaults 0 0
UUID=401bb276-312f-4651-9a99-1f0db4e65ae4 /boot xfs defaults 0 0
UUID=db95190e-ed05-4b2b-9d66-f80338c9589f none swap defaults 0 0
/swapfile none swap defaults 0 0
/var/www /srv/nfs/www none bind 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
/home/kazhavoronkov/common /srv/nfs/home/kazhavoronkov none bind 0 0
```

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

Повторно экспортируем каталоги (Рис. 4.4):

```
exportfs -r
```



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

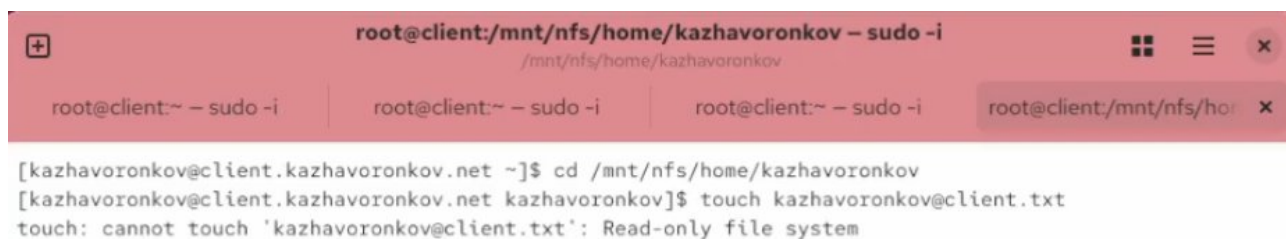
Рис. 4.4. Повторный экспорт каталогов.

На клиенте проверим каталог /mnt/nfs (Рис. 4.5):

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

Рис. 4.5. Проверка на клиенте каталога /mnt/nfs.

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



The screenshot shows a terminal window with the title bar "root@client:/mnt/nfs/home/kazhavoronkov – sudo -i". The terminal content is as follows:

```
root@client:~ – sudo -i
[kazhavoronkov@client.kazhavoronkov.net ~]$ cd /mnt/nfs/home/kazhavoronkov
[kazhavoronkov@client.kazhavoronkov.net kazhavoronkov]$ touch kazhavoronkov@client.txt
touch: cannot touch 'kazhavoronkov@client.txt': Read-only file system
```

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

```
[root@client.kazhavoronkov.net ~]# cd /mnt/nfs/home/kazhavoronkov
[root@client.kazhavoronkov.net kazhavoronkov]# touch kazhavoronkov@client.txt
touch: cannot touch 'kazhavoronkov@client.txt': Read-only file system
[root@client.kazhavoronkov.net kazhavoronkov]#
```

Рис. 4.7. Переход на клиенте под пользователем root в каталог /mnt/nfs/home/kazhavoronkov и попытка создать в нём файл kazhavoronkov@client.txt.

На сервере посмотрим, появились ли изменения в каталоге пользователя /home/kazhavoronkov/common (Рис. 4.8):


```
[root@server.kazhavoronkov.net ~]# ls /home/kazhavoronkov/common
kazhavoronkov@server.txt
[root@server.kazhavoronkov.net ~]# █
```

Рис. 4.8. Просмотр наличия изменений на сервере в каталоге пользователя /home/kazhavoronkov/common.

На виртуальной машине server перейдём 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. В каталоге /vagrant/provision/server создадим исполняемый файл nfs.sh (Рис. 5.1):

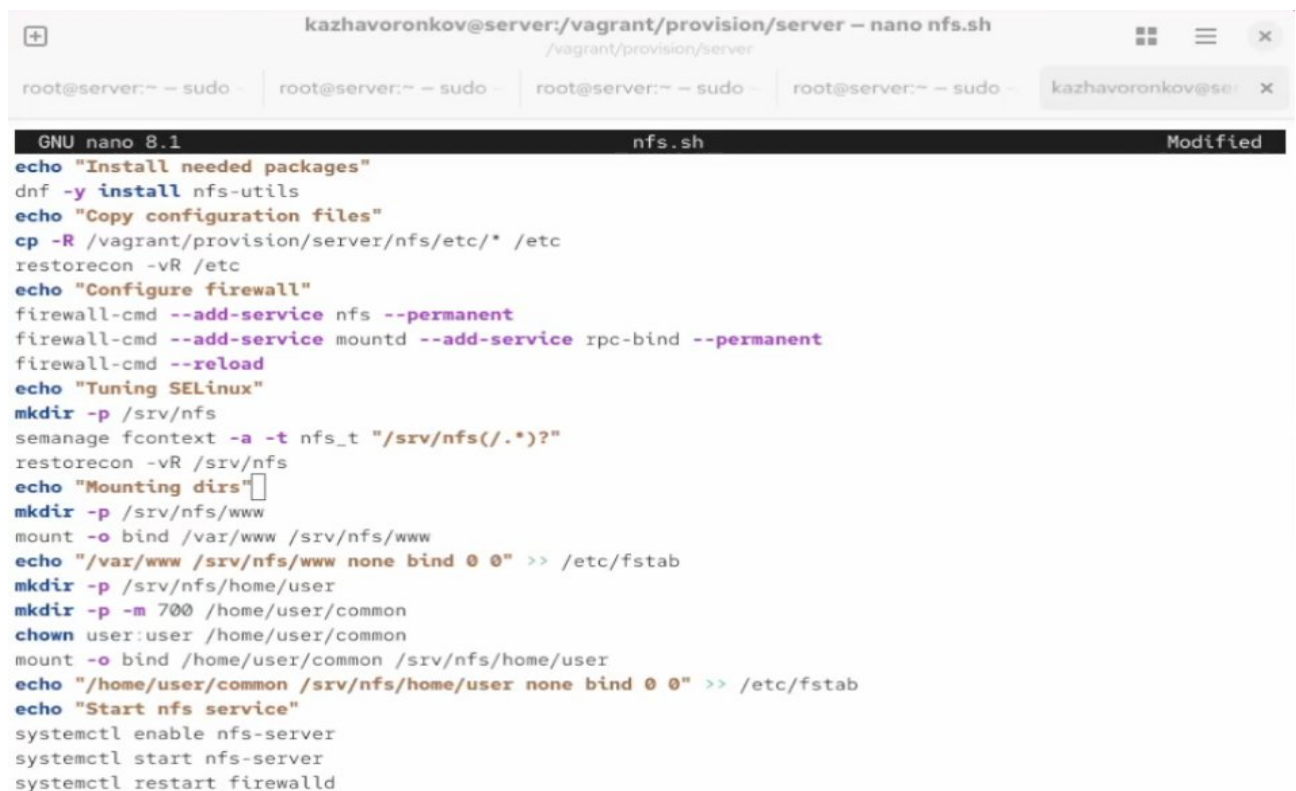


```
kazhavoronkov@server:/vagrant/provision/server
/vagrant/provision/server
root@server:~ - sudo - | root@server:~ - sudo - | root@server:~ - sudo - | root@server:~ - sudo - | kazhavoronkov@ser x

[kazhavoronkov@server.kazhavoronkov.net common]$ cd /vagrant/provision/server
[kazhavoronkov@server.kazhavoronkov.net server]$ mkdir -p /vagrant/provision/server/nfs/etc
[kazhavoronkov@server.kazhavoronkov.net server]$ cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[kazhavoronkov@server.kazhavoronkov.net server]$ touch nfs.sh
[kazhavoronkov@server.kazhavoronkov.net server]$ chmod +x nfs.sh
[kazhavoronkov@server.kazhavoronkov.net server]$ nano nfs █
```

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):

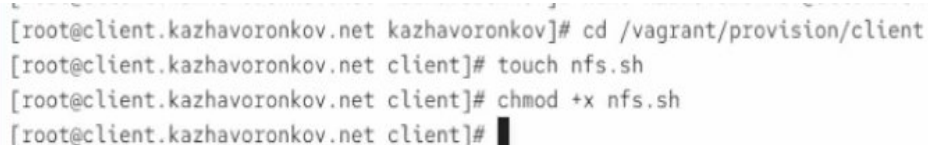


```
kazhavoronkov@server:/vagrant/provision/server - nano nfs.sh
/vagrant/provision/server
root@server:~ - sudo - | root@server:~ - sudo - | root@server:~ - sudo - | root@server:~ - sudo - | kazhavoronkov@server:~ - sudo - x

GNU nano 8.1                                nfs.sh                                Modified
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.

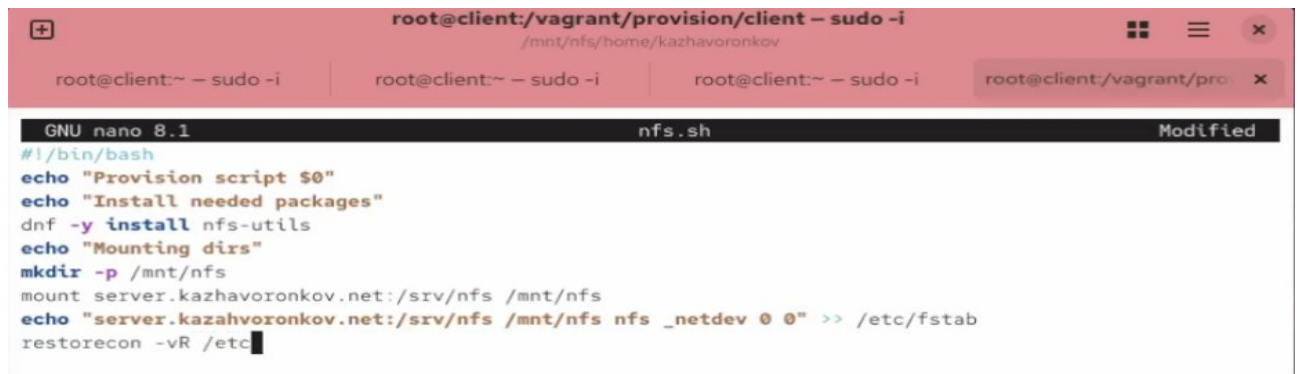
После чего на виртуальной машине client перейдём в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/. В каталоге /vagrant/provision/client создадим исполняемый файл nfs.sh (Рис. 5.3):



```
[root@client.kazhavoronkov.net kazhavoronkov]# cd /vagrant/provision/client
[root@client.kazhavoronkov.net client]# touch nfs.sh
[root@client.kazhavoronkov.net client]# chmod +x nfs.sh
[root@client.kazhavoronkov.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 - sudo -i
/mnt/nfs/home/kazhavoronkov

root@client:~ - sudo -i | root@client:~ - sudo -i | root@client:~ - sudo -i | root@client:/vagrant/provision/client - sudo -i

GNU nano 8.1                                nfs.sh                                Modified
#!/bin/bash
echo "Provision script $0"
echo "Install needed packages"
dnf -y install nfs-utils
echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.kazhavoronkov.net:/srv/nfs /mnt/nfs
echo "server.kazhavoronkov.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 nfs",
  type: "shell",
  preserve_order: true,
  path: "provision/server/nfs.sh"
```

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

```
client.vm.provision "client nfs",
  path: "provision/client/ntp.sh"
  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 си-

стемы, 0 0 - опции для процесса проверки файловой системы при загрузке.