

Отчёт по лабораторной работе 13

Настройка NFS

Седохин Даниил Алексеевич

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
4	Контрольные вопросы	23
5	Выводы	24

Список иллюстраций

3.1	Установка необходимого программного обеспечения	8
3.2	Подключаемый через NFS общий каталог с доступом только на чтение	9
3.3	Задаем контекст безопасности NFS, применение измененных настроек SELinux, запуск сервера NFS, настройка межсетевого экрана	9
3.4	Установка необходимого ПО на клиенте	10
3.5	Попытка просмотра имеющихся подмонтированных удаленных ресурсов	10
3.6	Остановка сервиса межсетевого экрана	11
3.7	Повторная попытка подключиться к удаленно смонтированному ресурсу	11
3.8	Просмотр службы TCP	11
3.9	Просмотр службы UDP	12
3.10	Добавление службы rpc-bind и mountd в настройки межсетевого экрана на сервере	13
3.11	Проверка подключения удаленного ресурса	13
3.12	Создание каталога для монтирования удаленного ресурса и проверка корректного подключения ресурса NFS	14
3.13	Редактирование файла /etc/fstab	14
3.14	Проверка наличия автоматического монтирования удаленных ресурсов при запуске ОС	15
3.15	Проверка автоматического подключения удаленного ресурса	15
3.16	Создание общего каталога, подмонтирование каталога web-сервера и просмотр каталога /srv/nfs	15
3.17	Просмотр каталога /mnt/nfs	16
3.18	Редактирование файла /etc/exports	16
3.19	Экспорт всех каталогов указанных в файле /etc/exportfs	16
3.20	Редактирование файла /etc/fstab	17
3.21	Повторный экспорт каталогов, указанных в файле /etc/exportfs	17
3.22	Проверка каталога /mnt/nfs	17
3.23	Создание каталогов, создание файла и монтирование каталога common	18
3.24	Редактирование файла /etc/exports	18
3.25	Редактирование файла /etc/fstab	19
3.26	Экспорт каталогов, проверка /mnt/nfs и попытка создать файл и отредактировать его	19
3.27	Просмотр каталога /home/dasedokhin/common на сервере	20

3.28	Создание необходимых каталогов для конфигурационных файлов, создание конфиг файлов	20
3.29	Скрипт файла nfs.sh	21
3.30	Создание каталога для внесения изменений и создание исполняемо- го файла nfs.sh на клиенте	21
3.31	Скрипт файла nfs.sh на клиенте	22
3.32	Скрипт в Vagrantfile в разделе конфигураций сервера	22
3.33	Скрипт в Vagrantfile в разделе конфигураций клиента	22

Список таблиц

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

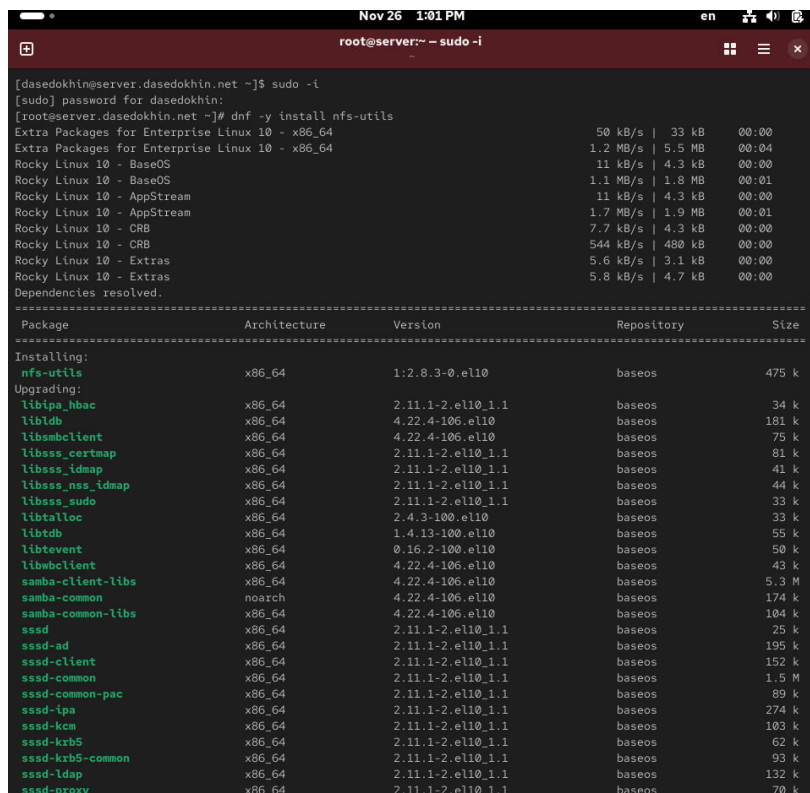
2 Задание

1. Установите и настройте сервер NFSv4 (см. раздел 13.4.1).
2. Подмонтируйте удалённый ресурс на клиенте (см. раздел 13.4.2).
3. Подключите каталог с контентом веб-сервера к дереву NFS (см. раздел 13.4.3).
4. Подключите каталог для удалённой работы вашего пользователя к дереву NFS (см. раздел 13.4.4).
5. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сервера NFSv4 во внутреннем окружении виртуальных машин `server` и `client`. Соответствующим образом внесите изменения в Vagrantfile (см. раздел 13.4.5).

3 Выполнение лабораторной работы

На сервере установим необходимое программное обеспечение с помощью команды:

`dnf -y install nfs-utils` (рис. 3.1).



```
[dasedokhin@server.dasedokhin.net ~]$ sudo -i
[sudo] password for dasedokhin:
[root@server.dasedokhin.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64                    50 kB/s | 33 kB    00:00
Extra Packages for Enterprise Linux 10 - x86_64                    1.2 MB/s | 5.5 MB    00:04
Rocky Linux 10 - BaseOS                                           11 kB/s | 4.3 kB    00:00
Rocky Linux 10 - BaseOS                                           1.1 MB/s | 1.8 MB    00:01
Rocky Linux 10 - AppStream                                         11 kB/s | 4.3 kB    00:00
Rocky Linux 10 - AppStream                                         1.7 MB/s | 1.9 MB    00:01
Rocky Linux 10 - CRB                                               7.7 kB/s | 4.3 kB    00:00
Rocky Linux 10 - CRB                                              544 kB/s | 480 kB    00:00
Rocky Linux 10 - Extras                                           5.6 kB/s | 3.1 kB    00:00
Rocky Linux 10 - Extras                                           5.8 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
libmbclient                           x86_64            4.22.4-106.el10  baseos            75 k
libss_certmap                         x86_64            2.11.1-2.el10_1.1 baseos            81 k
libss_idmap                           x86_64            2.11.1-2.el10_1.1 baseos            41 k
libss_nss_idmap                       x86_64            2.11.1-2.el10_1.1 baseos            44 k
libss_sudo                            x86_64            2.11.1-2.el10_1.1 baseos            33 k
libtalloc                             x86_64            2.4.3-100.el10   baseos            33 k
libtdb                                 x86_64            1.4.13-100.el10  baseos            55 k
libtevent                             x86_64            0.16.2-100.el10  baseos            50 k
libwbclient                           x86_64            4.22.4-106.el10  baseos            43 k
samba-client-libs                     x86_64            4.22.4-106.el10  baseos            5.3 M
samba-common                          noarch            4.22.4-106.el10  baseos            174 k
samba-common-libs                     x86_64            4.22.4-106.el10  baseos            104 k
sssd                                   x86_64            2.11.1-2.el10_1.1 baseos            25 k
sssd-ad                               x86_64            2.11.1-2.el10_1.1 baseos            195 k
sssd-client                           x86_64            2.11.1-2.el10_1.1 baseos            152 k
sssd-common                           x86_64            2.11.1-2.el10_1.1 baseos            1.5 M
sssd-common-pac                       x86_64            2.11.1-2.el10_1.1 baseos            89 k
sssd-ipa                              x86_64            2.11.1-2.el10_1.1 baseos            274 k
sssd-kcm                              x86_64            2.11.1-2.el10_1.1 baseos            103 k
sssd-krb5                             x86_64            2.11.1-2.el10_1.1 baseos            62 k
sssd-krb5-common                     x86_64            2.11.1-2.el10_1.1 baseos            93 k
sssd-ldap                             x86_64            2.11.1-2.el10_1.1 baseos            132 k
sssd-proxy                            x86_64            2.11.1-2.el10_1.1 baseos            70 k
```

Рисунок 3.1: Установка необходимого программного обеспечения

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети с помощью команды:

`mkdir -p /srv/nfs`

После этого в файле `/etc/exports` пропишем подключаемый через NFS общий каталог с доступом только на чтение:

`/srv/nfs *(ro)` (рис. 3.2).

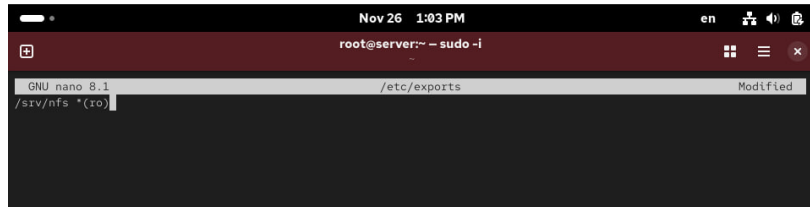


Рисунок 3.2: Подключаемый через NFS общий каталог с доступом только на чтение

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:

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

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

```
firewall-cmd --reload (рис. 3.3).
```



Рисунок 3.3: Задаем контекст безопасности NFS, применение измененных настроек SELinux, запуск сервера NFS, настройка межсетевого экрана

Установка необходимого для работы NFS программного обеспечения на клиенте (рис. 3.4).

dnf -y install nfs-utils

```
root@client:~ - sudo -i
[dasedokhin@client.dasedokhin.net ~]$ sudo -i
[sudo] password for dasedokhin:
[root@client.dasedokhin.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 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
Rocky Linux 10 - Extras
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
libnmbclient x86_64 4.22.4-106.el10 baseos 75 k
libnss_certmap x86_64 2.11.1-2.el10 1.1 baseos 81 k
libnss_idmap x86_64 2.11.1-2.el10 1.1 baseos 41 k
libnss_nss_idmap x86_64 2.11.1-2.el10 1.1 baseos 44 k
libnss_sudo x86_64 2.11.1-2.el10 1.1 baseos 33 k
libnss_ldap x86_64 2.4.3-100.el10 baseos 33 k
libtdb x86_64 1.4.13-100.el10 baseos 55 k
libtevent x86_64 0.16.2-100.el10 baseos 50 k
libwbclient x86_64 4.22.4-106.el10 baseos 43 k
samba-client-libs x86_64 4.22.4-106.el10 baseos 5.3 M
samba-common noarch 4.22.4-106.el10 baseos 174 k
samba-common-libs x86_64 4.22.4-106.el10 baseos 104 k
sssd x86_64 2.11.1-2.el10 1.1 baseos 25 k
sssd-ad x86_64 2.11.1-2.el10 1.1 baseos 195 k
sssd-client x86_64 2.11.1-2.el10 1.1 baseos 152 k
sssd-common x86_64 2.11.1-2.el10 1.1 baseos 1.5 M
sssd-common-pac x86_64 2.11.1-2.el10 1.1 baseos 89 k
sssd-tpa x86_64 2.11.1-2.el10 1.1 baseos 274 k
sssd-krb5 x86_64 2.11.1-2.el10 1.1 baseos 103 k
sssd-krb5-common x86_64 2.11.1-2.el10 1.1 baseos 62 k
sssd-ldap x86_64 2.11.1-2.el10 1.1 baseos 93 k
sssd-proxy x86_64 2.11.1-2.el10 1.1 baseos 132 k
```

Рисунок 3.4: Установка необходимого ПО на клиенте

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы:

showmount -e server.dasedokhin.net

(рис. 3.5).

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

Рисунок 3.5: Попытка просмотра имеющихся подмонтированных удаленных ресурсов

Теперь на сервере останавливаем сервис межсетевого экрана:

systemctl stop firewalld.service

(рис. 3.6).

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

Рисунок 3.6: Остановка сервиса межсетевого экрана

После этого на клиенте вновь пробуем подключиться к удаленно смонтированному ресурсу:

showmount -e server.dasedokhin.net (рис. 3.7).

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

Рисунок 3.7: Повторная попытка подключиться к удаленно смонтированному ресурсу

На сервере запустим сервис межсетевого экрана:

```
systemctl start firewalld
```

После этого посмотрим какие службы задействованы при удаленном монтировании:

```
lsof | grep TCP
```

(рис. 3.8).

```
[root@server.dasedokhin.net ~]# systemctl start firewalld
[root@server.dasedokhin.net ~]# lsof | grep TCP
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 318u IPv6 8116 0t0 TCP *:websock (LISTEN)
systemd 1 root 429u IPv4 27545 0t0 TCP *:sunrpc (LISTEN)
systemd 1 root 433u IPv6 27559 0t0 TCP *:sunrpc (LISTEN)
cupsd 1240 root 7u IPv6 11374 0t0 TCP localhost:ipp (LISTEN)
cupsd 1240 root 8u IPv4 11375 0t0 TCP localhost:ipp (LISTEN)
sshd 1250 root 7u IPv4 11382 0t0 TCP *:down (LISTEN)
sshd 1250 root 8u IPv6 11384 0t0 TCP *:down (LISTEN)
sshd 1250 root 9u IPv4 11386 0t0 TCP *:ssh (LISTEN)
sshd 1250 root 10u IPv6 11388 0t0 TCP *:ssh (LISTEN)
named 1300 named 27u IPv4 11446 0t0 TCP localhost:domain (LISTEN)
named 1300 named 29u IPv4 11447 0t0 TCP localhost:domain (LISTEN)
named 1300 named 33u IPv6 11450 0t0 TCP localhost:domain (LISTEN)
named 1300 named 34u IPv6 11451 0t0 TCP localhost:domain (LISTEN)
named 1300 named 35u IPv4 11462 0t0 TCP localhost:rndc (LISTEN)
named 1300 named 36u IPv4 11463 0t0 TCP localhost:rndc (LISTEN)
named 1300 named 37u IPv6 11464 0t0 TCP localhost:rndc (LISTEN)
named 1300 named 38u IPv6 11465 0t0 TCP localhost:rndc (LISTEN)
named 1300 named 43u IPv4 11683 0t0 TCP server.dasedokhin.net:domain (LISTEN)
```

Рисунок 3.8: Просмотр службы TCP

lsof | grep UDP (рис. 3.9).

```
[root@server.dasedokhin.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    430u  IPv4        27552      0t0      UDP *:sunrpc
systemd      1      root    434u  IPv6        27566      0t0      UDP *:sunrpc
avahi-daemon 884     avahi    12u   IPv4        9457       0t0      UDP *:mdns
avahi-daemon 884     avahi    13u   IPv6        9458       0t0      UDP *:mdns
chronyd      937     chrony    5u   IPv4        8737       0t0      UDP localhost:323
chronyd      937     chrony    6u   IPv6        8738       0t0      UDP localhost:323
chronyd      937     chrony    7u   IPv4        8739       0t0      UDP *:ntp
NetworkManager 1215    root     29u   IPv4        11681      0t0      UDP server.dasedokhin.net
NetworkManager 1215 1226 gmain    root     29u   IPv4        11681      0t0      UDP server.dasedokhin.net
NetworkManager 1215 1227 pool-spawn root     29u   IPv4        11681      0t0      UDP server.dasedokhin.net
NetworkManager 1215 1228 gdbus    root     29u   IPv4        11681      0t0      UDP server.dasedokhin.net
named        1300    named    25u   IPv4        11444      0t0      UDP localhost:domain
named        1300    named    26u   IPv4        11445      0t0      UDP localhost:domain
named        1300    named    31u   IPv6        11448      0t0      UDP localhost:domain
named        1300    named    32u   IPv6        11449      0t0      UDP localhost:domain
named        1300    named    41u   IPv4        11680      0t0      UDP server.dasedokhin.net
named        1300    named    42u   IPv4        11682      0t0      UDP server.dasedokhin.net
named        1300    named    45u   IPv4        11691      0t0      UDP server.dasedokhin.net
named        1300    named    46u   IPv4        11692      0t0      UDP server.dasedokhin.net
```

Рисунок 3.9: Просмотр службы 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
```

(рис. 3.10).

```
[root@server.dasedokhin.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 anno-1800 apc
upsd aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin b
itcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civi
lization-iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 d
hcpv6-client distcc dns dns-over-quit dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-cl
ient etcd-server factorio finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freei
pa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability http http3 https ident imap
imaps iperf2 iperf3 ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana kl
ogin kpasswd kprop kshell kube-api kube-api-server kube-control-plane kube-control-plane-secure kube-controller-manage
r kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kube
let-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp m
anagesieve matrix mdns memcache minecraft minidlna mndp mongodb mosh mountd mpd mqtt mqtt-tls ms-wbt mssql murmur mys
ql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry o
penvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql
privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius r
adsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane se
tillers-history-collection stp sips slimevr slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spidero
k-lansync spotify-sync squid ssdp ssh ssh-custom statrsv steam-lan-transfer steam-streaming stellaris stronghold-crus
ader stun stuns submission supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay synergy syscomlan syslog sy
slog-tls telnet tentacle terraria tftp tile38 tinc tor-socks transmission-client turn turns upnp-client vdsu vnc-serv
er vrrp warpinator wben-http wben-https wreguard ws-discovery ws-discovery-client ws-discovery-host ws-discovery-tcp
ws-discovery-udp wssd wssd-http wsmn wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-
java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.dasedokhin.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.dasedokhin.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.dasedokhin.net ~]# firewall-cmd --reload
success
[root@server.dasedokhin.net ~]#
```

Рисунок 3.10: Добавление службы rpc-bind и mountd в настройки межсетевого экрана на сервере

На клиенте проверим подключение удалённого ресурса:

showmount -e server.dasedokhin.net (рис. 3.11).

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

Рисунок 3.11: Проверка подключения удаленного ресурса

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

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

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

Проверим, что общий ресурс NFS подключён правильно:

```
mount (рис. 3.12).
```

```
root@client:~# sudo -i
[roo@client.dasedokhin.net ~]# mkdir -p /mnt/nfs
[roo@client.dasedokhin.net ~]# mount server.dasedokhin.net:/srv/nfs /mnt/nfs
[roo@client.dasedokhin.net ~]# mount
/dev/mapper/rl-root 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=211166,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=345392k,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,pi
pe_ino=6287)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
nfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymlinks,seclabel,s
ize=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,erro
rs=remount-ro)
/dev/mapper/rl-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,_netdev)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=172692k,nr_inodes=43173,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=172692k,nr_inodes=43173,mode=700,inode64)
server.dasedokhin.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)
[roo@client.dasedokhin.net ~]#
```

Рисунок 3.12: Создание каталога для монтирования удаленного ресурса и проверка корректного подключения ресурса NFS

На клиенте в конце файла `/etc/fstab` добавим следующую запись:
`server.dasedokhin.net:/srv/nfs /mnt/nfs nfs _netdev 0 0` (рис. 3.13).

```
GNU nano 8.1 /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Sep 18 18:06:51 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=3533f347-0c0c-4e45-8f1a-dab6fc99e8fa / xfs defaults 0 0
UUID=4c96d706-e6a8-410b-8436-e385d9062b7e /boot xfs defaults 0 0
UUID=4665-6c76 /boot/efi vfat umask=0077,shortname=winnt 0 2
UUID=bb1ce926-ba7d-43b6-b3fe-4c7bf383a00f /home xfs defaults 0 0
UUID=d7514780-f487-4286-a4e3-3505c485f873 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.dasedokhin.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рисунок 3.13: Редактирование файла `/etc/fstab`

На клиенте проверим наличие автоматического монтирования удалённых ресур-
сов при запуске операционной системы:

systemctl status remote-fs.target (рис. 3.14).

```
[root@client.dasedokhin.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 Wed 2025-11-26 12:56:56 UTC; 16min ago
   Invocation: b6ffb7409c7c44cb8bca32be60681c07
   Docs: man:systemd.special(7)

Nov 26 12:56:56 client.dasedokhin.net systemd[1]: Reached target remote-fs.target - Remote File Systems.
[root@client.dasedokhin.net ~]#
```

Рисунок 3.14: Проверка наличия автоматического монтирования удаленных ресурсов при запуске ОС

Перезапустим клиента и убедитесь, что удалённый ресурс подключается автоматически. (рис. 3.15).

```
[root@client.dasedokhin.net ~]# mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.dasedokhin.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsz=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,_netdev)
[root@client.dasedokhin.net ~]#
```

Рисунок 3.15: Проверка автоматического подключения удаленного ресурса

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

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

Подмонтируйте каталог web-сервера:

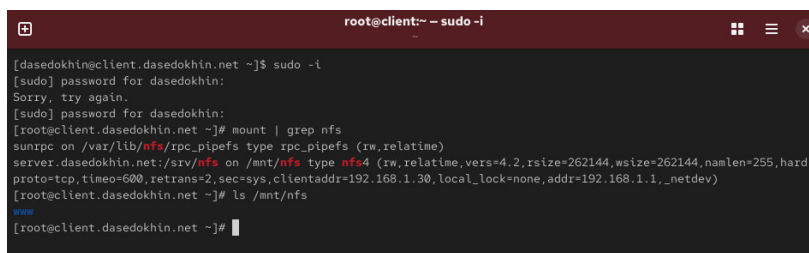
```
mount -o bind /var/www/ /srv/nfs/www/
```

На сервере проверьте, что отображается в каталоге /srv/nfs (рис. 3.16).

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

Рисунок 3.16: Создание общего каталога, подмонтирование каталога web-сервера и просмотр каталога /srv/nfs

На клиенте посмотрим, что отображается в каталоге /mnt/nfs (рис. 3.17).

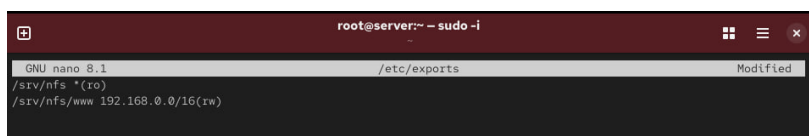


```
root@client:~# sudo -i
[dasedokhin@client.dasedokhin.net ~]$ sudo -i
[sudo] password for dasedokhin:
Sorry, try again.
[sudo] password for dasedokhin:
[root@client.dasedokhin.net ~]# mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.dasedokhin.net:/srv/nfs on /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,_netdev)
[root@client.dasedokhin.net ~]# ls /mnt/nfs
www
[root@client.dasedokhin.net ~]#
```

Рисунок 3.17: Просмотр каталога /mnt/nfs

На сервере в файле /etc/exports добавьте экспорт каталога веб-сервера с удалённого ресурса:

/srv/nfs/www 192.168.0.0/16(rw) (рис. 3.18).




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

Рисунок 3.18: Редактирование файла /etc/exports

Экспортируем все каталоги, упомянутые в файле /etc/exports:

exportfs -r (рис. 3.19).

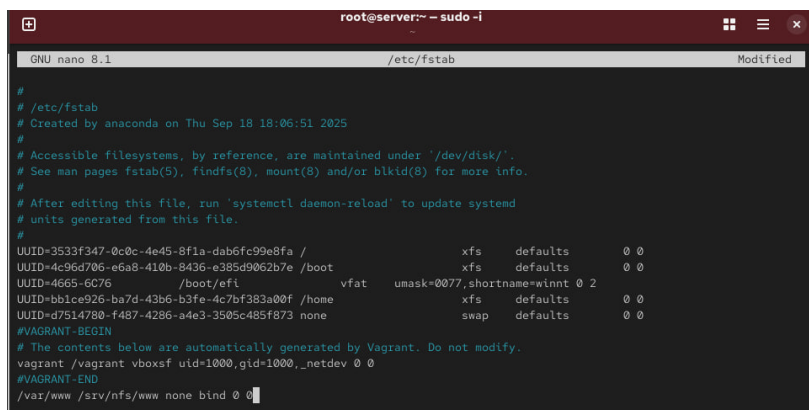


```
[root@server.dasedokhin.net ~]# exportfs -r
[root@server.dasedokhin.net ~]#
```

Рисунок 3.19: Экспорт всех каталогов указанных в файле /etc/exports

На сервере в конце файла /etc/fstab добавим следующую запись:

/var/www /srv/nfs/www none bind 0 0 (рис. 3.20).

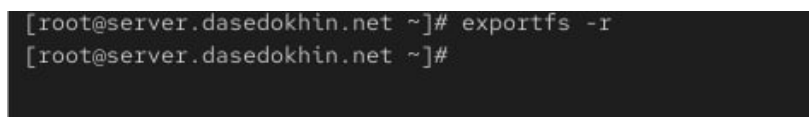


```
root@server:~ - sudo -i
GNU nano 8.1 /etc/fstab Modified
#
# /etc/fstab
# Created by anaconda on Thu Sep 18 18:06:51 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=3533f347-0c0c-4e45-8f1a-dab6fc99e8fa / xfs defaults 0 0
UUID=4c96d706-e6a8-410b-8436-e385d9062b7e /boot xfs defaults 0 0
UUID=4665-6C76 /boot/efi vfat umask=0077,shortname=winnt 0 2
UUID=bb1ce926-ba7d-43b6-b3fe-4c7bf383a00f /home xfs defaults 0 0
UUID=d7514780-f487-4286-a4e3-3505c485f873 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
```

Рисунок 3.20: Редактирование файла /etc/fstab

Повторно экспортируем каталоги, указанные в файле /etc/exports:

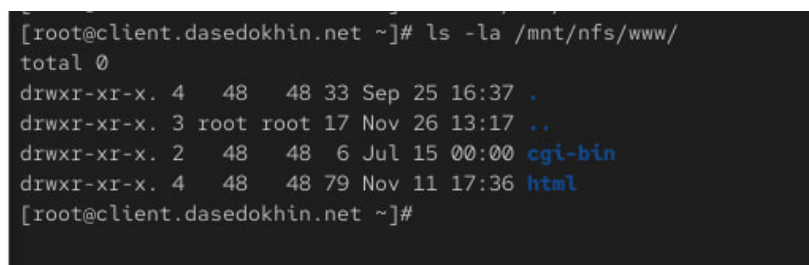
exportfs -r (рис. 3.21).



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

Рисунок 3.21: Повторный экспорт каталогов, указанных в файле /etc/exports

На клиенте проверим каталог /mnt/nfs. (рис. 3.22).



```
[root@client.dasedokhin.net ~]# ls -la /mnt/nfs/www/
total 0
drwxr-xr-x. 4 48 48 33 Sep 25 16:37 .
drwxr-xr-x. 3 root root 17 Nov 26 13:17 ..
drwxr-xr-x. 2 48 48 6 Jul 15 00:00 cgi-bin
drwxr-xr-x. 4 48 48 79 Nov 11 17:36 html
[root@client.dasedokhin.net ~]#
```

Рисунок 3.22: Проверка каталога /mnt/nfs

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

```
mkdir -p -m 700 ~/common
cd ~/common
```

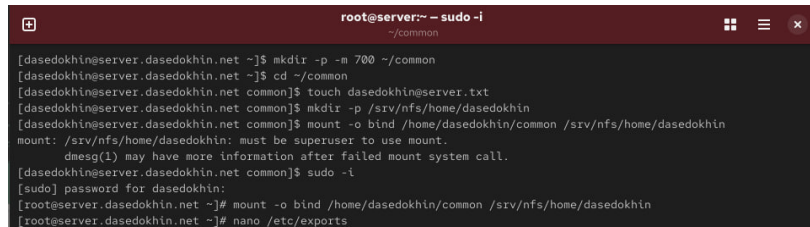
touch dasedokhin@server.txt

На сервере создадим общий каталог для работы пользователя user по сети:

mkdir -p /srv/nfs/home/dasedokhin

Подмонтируем каталог common пользователя user в NFS:

mount -o bind /home/dasedokhin/common /srv/nfs/home/dasedokhin (рис. 3.23).

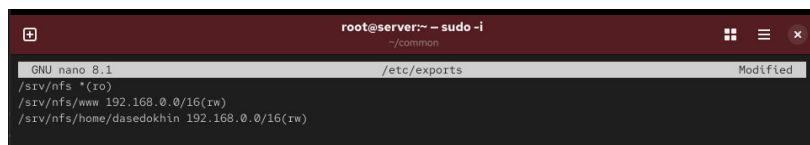


```
root@server:~# sudo -i
~/common
[dasedokhin@server.dasedokhin.net ~]$ mkdir -p -m 700 ~/common
[dasedokhin@server.dasedokhin.net ~]$ cd ~/common
[dasedokhin@server.dasedokhin.net common]$ touch dasedokhin@server.txt
[dasedokhin@server.dasedokhin.net common]$ mkdir -p /srv/nfs/home/dasedokhin
[dasedokhin@server.dasedokhin.net common]$ mount -o bind /home/dasedokhin/common /srv/nfs/home/dasedokhin
mount: /srv/nfs/home/dasedokhin: must be superuser to use mount.
dmesg(1) may have more information after failed mount system call.
[dasedokhin@server.dasedokhin.net common]$ sudo -i
[sudo] password for dasedokhin:
[root@server.dasedokhin.net ~]# mount -o bind /home/dasedokhin/common /srv/nfs/home/dasedokhin
[root@server.dasedokhin.net ~]# nano /etc/exports
```

Рисунок 3.23: Создание каталогов, создание файла и монтирование каталога common

Подключим каталог пользователя в файле /etc/exports, прописав в нём:

/srv/nfs/home/dasedokhin 192.168.0.0/16(rw) (рис. 3.24).



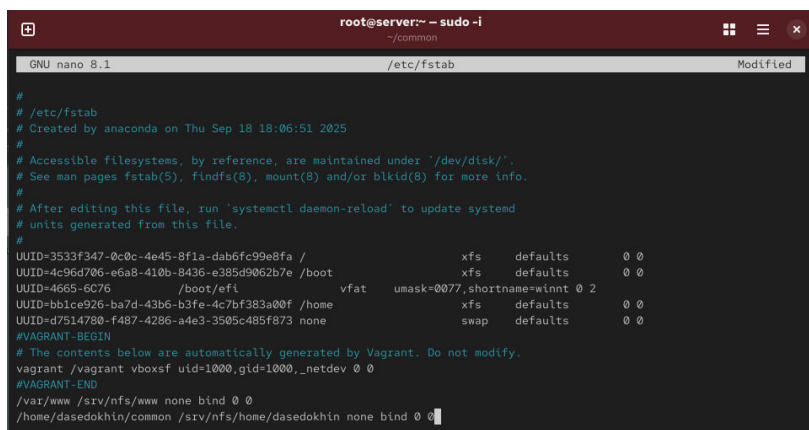
```
GNU nano 8.1 /etc/exports Modified
/srv/nfs /*(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/dasedokhin 192.168.0.0/16(rw)
```

Рисунок 3.24: Редактирование файла /etc/exports

Внесем изменения в файл /etc/fstab:

/home/dasedokhin/common /srv/nfs/home/dasedokhin none bind 0 0

(рис. 3.25).



```
root@server:~# sudo -i
GNU nano 8.1 /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Sep 18 18:06:51 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=3533f347-0c0c-4e45-8f1a-dab6fc99e8fa / xfs defaults 0 0
UUID=4c96d786-e6a8-410b-8436-e385d9062b7e /boot xfs defaults 0 0
UUID=4665-6676 /boot/efi vfat umask=0077,shortname=winnt 0 2
UUID=b11ce926-ba7d-43b6-b3fe-4c7bf383a00f /home xfs defaults 0 0
UUID=d7514780-f487-4286-a4e3-3505c485f873 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
/vai/www /srv/nfs/www none bind 0 0
/home/dasedokhin/common /srv/nfs/home/dasedokhin none bind 0 0
```

Рисунок 3.25: Редактирование файла /etc/fstab

Повторно экспортируем каталоги:

```
exportfs -r
```

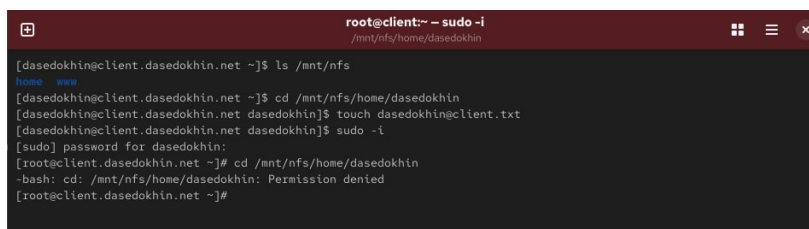
На клиенте проверим каталог /mnt/nfs.

На клиенте под пользователем user перейдем в каталог /mnt/nfs/home/user и попробуйте создать в нём файл dasedokhin@client.txt и внести в него какие-либо изменения:

```
cd /mnt/nfs/home/user
```

```
touch dasedokhin@client.txt
```

Попробуем проделать это под пользователем root (рис. 3.26).



```
root@client:~# sudo -i
[dasedokhin@client.dasedokhin.net ~]$ ls /mnt/nfs
home  www
[dasedokhin@client.dasedokhin.net ~]$ cd /mnt/nfs/home/dasedokhin
[dasedokhin@client.dasedokhin.net dasedokhin]$ touch dasedokhin@client.txt
[dasedokhin@client.dasedokhin.net dasedokhin]$ sudo -i
[sudo] password for dasedokhin:
[root@client.dasedokhin.net ~]# cd /mnt/nfs/home/dasedokhin
-bash: cd: /mnt/nfs/home/dasedokhin: Permission denied
[root@client.dasedokhin.net ~]#
```

Рисунок 3.26: Экспорт каталогов, проверка /mnt/nfs и попытка создать файл и отредактировать его

На сервере посмотрим, появились ли изменения в каталоге пользователя /home/dasedokhin/common. (рис. 3.27).

```
[root@server.dasedokhin.net ~]# exportfs -r
[root@server.dasedokhin.net ~]# ls /home/dasedokhin/common
dasedokhin@client.txt  dasedokhin@server.txt
[root@server.dasedokhin.net ~]#
```

Рисунок 3.27: Просмотр каталога /home/dasedokhin/common на сервере

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создайте 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 создайте исполняемый файл nfs.sh:

```
cd /vagrant/provision/server
```

```
touch nfs.sh
```

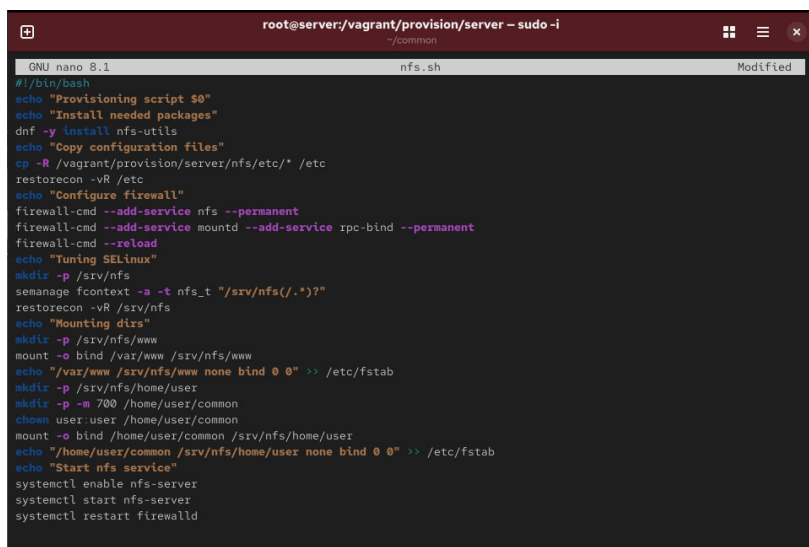
```
chmod +x nfs.sh
```

(рис. 3.28).

```
root@server:/vagrant/provision/server - sudo -i
~/common
[root@server.dasedokhin.net ~]# cd /vagrant/provision/server
[root@server.dasedokhin.net server]# mkdir -p /vagrant/provision/server/nfs/etc
[root@server.dasedokhin.net server]# cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[root@server.dasedokhin.net server]# touch /nfs.sh
[root@server.dasedokhin.net server]# chmod +x nfs.sh
chmod: cannot access 'nfs.sh': No such file or directory
[root@server.dasedokhin.net server]# touch nfs.sh
[root@server.dasedokhin.net server]# chmod +x nfs.sh
[root@server.dasedokhin.net server]#
```

Рисунок 3.28: Создание необходимых каталогов для конфигурационных файлов, создание конфиг файлов

Теперь пропишем скрипт в созданном файле nfs.sh (рис. 3.29).



```
root@server:/vagrant/provision/server - sudo -i
GNU nano 8.1                                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
```

Рисунок 3.29: Скрипт файла nfs.sh

На виртуальной машине client перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/:

```
cd /vagrant/provision/client
```

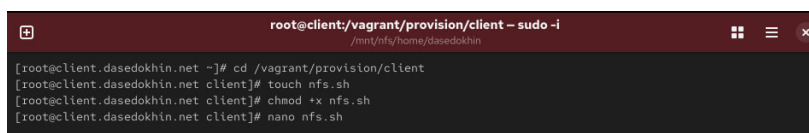
В каталоге /vagrant/provision/client создадим исполняемый файл nfs.sh:

```
cd /vagrant/provision/client
```

```
touch nfs.sh
```

```
chmod +x nfs.sh
```

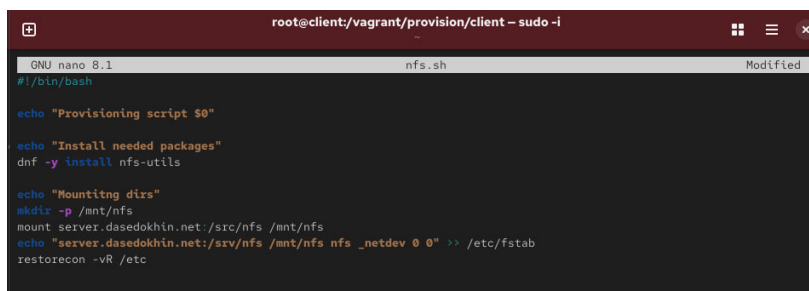
(рис. 3.30).



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

Рисунок 3.30: Создание каталога для внесения изменений и создание исполняемого файла nfs.sh на клиенте

Пропишем скрипт файла nfs.sh на клиенте (рис. 3.31).

A terminal window titled 'root@client:/vagrant/provision/client - sudo -i' showing the contents of a file named 'nfs.sh' in nano 8.1 editor. The script contains the following commands:

```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install nfs-utils

echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.dasedokhin.net /src/nfs /mnt/nfs
echo "server.dasedokhin.net:/src/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Рисунок 3.31: Скрипт файла nfs.sh на клиенте

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: (рис. 3.32).

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

Рисунок 3.32: Скрипт в Vagrantfile в разделе конфигураций сервера

(рис. 3.33).

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

Рисунок 3.33: Скрипт в Vagrantfile в разделе конфигураций клиента

4 Контрольные вопросы

1. Как называется файл конфигурации, содержащие общие ресурсы NFS? -
/etc/exports
2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS? - 2049 (nfs), 111 (rpcbind/sunrpc), а также порты для mountd, nlockmgr и status
3. Какую опцию следует использовать в /etc/fstab, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке? -
_netdev

5 Выводы

Я приобрел навыки настройки сервера NFS для удаленного доступа к ресурсам.