

# Отчёт о лабораторной работе

Лабораторная работа 13

Мошаров Денис Максимович

## Содержание

## Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам

## Выполнение лабораторной работы

Установим на сервере пакет nfs-utils, необходимый для работы NFS (рис. [-@fig:001]).

```
[root@server.dmmosharov.net ~]# dnf -y install nfs-utils
Rocky Linux 10 - BaseOS              722 B/s | 4.3 kB      00:06
Rocky Linux 10 - BaseOS              22 MB/s | 9.9 MB      00:00
Rocky Linux 10 - AppStream           400 B/s | 4.3 kB      00:11
Rocky Linux 10 - AppStream           6.3 MB/s | 2.1 MB      00:00
Rocky Linux 10 - CRB                 826 B/s | 4.3 kB      00:05
Rocky Linux 10 - CRB                 1.6 MB/s | 499 kB      00:00
Rocky Linux 10 - Extras              6.6 kB/s | 3.1 kB      00:00
Rocky Linux 10 - Extras              21 kB/s | 5.9 kB      00:00
Dependencies resolved.
=====
Package           Arch      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
libsss_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
```

Установка пакета nfs-utils на сервере

Создим каталог /srv/nfs, который будет использоваться как общий ресурс, и открываем файл конфигурации экспорта /etc/exports (рис. [-@fig:002]).

```
[root@server.dmmosharov.net ~]# mkdir -p /srv/nfs
[root@server.dmmosharov.net ~]# nano /etc/exports
```

Создание каталога и открытие файла конфигурации

В файле /etc/exports прописываем настройки для созданного каталога, разрешая доступ к нему всем узлам (\*) только на чтение (ro) (рис. [-@fig:003]).

```
GNU nano 8.1 /etc/exports
/srv/nfs *(ro)
```

Настройка экспорта каталога

Применим контекст безопасности SELinux nfs\_t к каталогу, запускаем и добавляем в автозагрузку службу NFS-сервера, а также открываем порт для службы nfs в межсетевом экране (рис. [-@fig:004]).

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

Настройка SELinux, запуск служб и настройка firewall

На клиентской виртуальной машине также устанавливаем пакет nfs-utils, необходимый для монтирования удаленных ресурсов (рис. [-@fig:005]).

```
[root@client.dmmosharov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64 157 kB/s | 39 kB 00:00
Extra Packages for Enterprise Linux 10 - x86_64 8.4 MB/s | 5.6 MB 00:00
Rocky Linux 10 - BaseOS 8.3 kB/s | 4.3 kB 00:00
Rocky Linux 10 - BaseOS 20 MB/s | 9.9 MB 00:00
Rocky Linux 10 - AppStream 14 kB/s | 4.3 kB 00:00
Rocky Linux 10 - CRB 14 kB/s | 4.3 kB 00:00
Rocky Linux 10 - Extras 2.1 kB/s | 3.1 kB 00:01
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
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
libsss_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
```

Установка nfs-utils на клиенте

На клиенте была выполнена попытка просмотра доступных удаленных ресурсов с помощью команды showmount -e. В результате было получено сообщение об ошибке cInt\_create: RPC: Unable to receive, что свидетельствует о невозможности установить соединение со службой RPC на сервере, вероятно, из-за блокировки сетевым экраном (рис. [-@fig:006]).

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

Ошибка подключения RPC при просмотре ресурсов на клиенте

Для диагностики проблемы и проверки влияния межсетевого экрана на сервере была временно остановлена служба `firewalld` (рис. [-@fig:007]).

```
[dmmosharov@server.dmmosharov.net ~]$ systemctl stop firewalld.service
[dmmosharov@server.dmmosharov.net ~]$
```

Остановка межсетевого экрана на сервере

После отключения фаервола на сервере, повторная проверка доступных ресурсов на клиенте прошла успешно. Команда отобразила экспортируемый каталог `/srv/nfs`, что подтверждает блокировку необходимых портов межсетевым экраном при его активной работе (рис. [-@fig:008]).

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

Успешный просмотр списка экспорта при отключенном фаерволе

Далее на сервере работа межсетевого экрана была восстановлена. Для определения служб и портов, задействованных при работе NFS, выполним команду `lsof` с фильтрацией по протоколу TCP (рис. [-@fig:009]).

```
[dmmosharov@server.dmmosharov.net ~]$ lsof | grep TCP
firefox    9490      dm-mosharov  102u      IPv4
49366      0t0        TCP server.dmmosharov.net:59120->34.107.243.9
3:https (ESTABLISHED)
firefox    9490      9510 AsyncSi~l  dm-mosharov  102u      IPv4
49366      0t0        TCP server.dmmosharov.net:59120->34.107.243.9
3:https (ESTABLISHED)
firefox    9490      9511 pool-spaw  dm-mosharov  102u      IPv4
49366      0t0        TCP server.dmmosharov.net:59120->34.107.243.9
3:https (ESTABLISHED)
firefox    9490      9512 gmain       dm-mosharov  102u      IPv4
49366      0t0        TCP server.dmmosharov.net:59120->34.107.243.9
3:https (ESTABLISHED)
```

Запуск `firewalld` и просмотр открытых портов

Вывод команды показывает наличие активных служб `rpcbind` и `rpc.mountd`. Это указывает на то, что для корректной работы NFS через фаервол необходимо разрешить доступ не только к службе `nfs`, но и к `mountd` и `rpc-bind` (рис. [-@fig:010]).

```
systemd    1         root      89u      IPv6      9794      0t0      TCP *:websock (LISTEN)
rpcbind    883       rpc       5u       IPv4      5864      0t0      TCP *:sunrpc (LISTEN)
rpcbind    883       rpc       7u       IPv6      5870      0t0      TCP *:sunrpc (LISTEN)
cupsd     1335      root      7u       IPv6      9184      0t0      TCP localhost:ipp (LISTEN)
cupsd     1335      root      8u       IPv4      9185      0t0      TCP localhost:ipp (LISTEN)
cupsd     1348      root      3u       IPv4     11115     0t0      TCP localhost:ipp (LISTEN)
```

Службы `rpcbind` и `mountd` в выводе `lsof`

На сервере выполняется команда `sudo lsof | grep UDP`, позволяющая определить, какие процессы используют UDP-сокеты; в выводе видны службы `systemd`, `avahi-daemon`, `chronyd`, `named` и другие, что используется для анализа сетевых сервисов, задействованных при работе NFS (рис. [-@fig:011]).

```

[dmmosharov@server.dmmosharov.net ~]$ sudo lsof | grep UDP
[sudo] password for dmmosharov:
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    43u   IPv4        6247      0t0      UDP *:sunrpc
systemd      1          root    45u   IPv6        6253      0t0      UDP *:sunrpc
rpcbind     883          rpc      6u   IPv4        6247      0t0      UDP *:sunrpc
rpcbind     883          rpc      8u   IPv6        6253      0t0      UDP *:sunrpc
avahi-daemon 937        avahi    12u   IPv4        9975      0t0      UDP *:mdns
avahi-daemon 937        avahi    13u   IPv6        9976      0t0      UDP *:mdns
chronyd     986        chrony    5u   IPv4        8041      0t0      UDP localhost:323
chronyd     986        chrony    6u   IPv6        8042      0t0      UDP localhost:323
chronyd     986        chrony    7u   IPv4        8043      0t0      UDP *:ntp
named       1414        named    41u   IPv4       11536      0t0      UDP localhost:domain
named       1414        named    42u   IPv4       11537      0t0      UDP localhost:domain
named       1414        named    43u   IPv4       11538      0t0      UDP localhost:domain
named       1414        named    44u   IPv4       11539      0t0      UDP localhost:domain
named       1414        named    53u   IPv4       11544      0t0      UDP server.dmmosharov.net:bootpc->_gateway
named       1414        named    54u   IPv4       11545      0t0      UDP server.dmmosharov.net:bootps
named       1414        named    55u   IPv4       11546      0t0      UDP server.dmmosharov.net:bootpc->_gateway

```

Просмотр UDP-соединений с помощью lsof (начало вывода)

Продолжение вывода команды `sudo lsof | grep UDP` на сервере; среди задействованных UDP-служб отображаются процессы `rpcbind`, `rpc.statd` и `rpc.mountd`, относящиеся к подсистеме RPC и участвующие в работе NFS-сервера, поэтому именно их порты необходимо открыть в межсетевом экране (рис. [-@fig:012]).

```

rpc.statd    1658          rpcuser    5u   IPv4
14376        0t0      UDP localhost:986
rpc.statd    1658          rpcuser    8u   IPv4
14381        0t0      UDP *:58507
rpc.statd    1658          rpcuser   10u   IPv6
13843        0t0      UDP *:34024
rpc.mountd   1659          root      4u   IPv4
14340        0t0      UDP *:mountd
rpc.mountd   1659          root      6u   IPv6
14346        0t0      UDP *:mountd
NetworkManager 5101        root      27u   IPv4
22874        0t0      UDP server.dmmosharov.net:bootpc->_gateway
:bootps
NetworkManager 5101        gmain     27u   IPv4
22874        0t0      UDP server.dmmosharov.net:bootpc->_gateway
:bootps

```

Просмотр UDP-соединений с помощью lsof (окончание вывода)

На сервере с помощью `firewall-cmd --get-services` просматриваем список предопределённых сетевых служб в `firewalld`, после чего выполняем команды `firewall-cmd --add-service=mountd --add-service=rpc-bind` и аналогичную команду с опцией `--permanent`, открывающие доступ к службам `mountd` и `rpc-bind` как в текущей, так и в постоянной конфигурации брандмауэра, затем конфигурация межсетевого экрана перезагружается командой `firewall-cmd --reload` (рис. [-@fig:013]).

```

7.34.0c.googleusercontent.com.https (ESTABLISHED)
[dmmosharov@server.dmmosharov.net ~]$ firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client aman
da-k5-client amqp amqps anno-1602 anno-1800 apcupsd aseqnet audit ausw
eisapp2 bacula 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 civ
ilization-iv civilization-v cockpit collectd condor-collector cratedb
ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dn
s dns-over-quic dns-over-tls docker-registry docker-swarm dropbox-lans
ync elasticsearch etcd-client etcd-server factorio finger foreman fore
man-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication fre
eipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana g
re high-availability http http3 https ident imap imaps iperf2 iperf3 i
pfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kde
connect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apis
erver kube-control-plane kube-control-plane-secure kube-controller-man
ager kube-controller-manager-secure kube-nodeport-services kube-schedu
ler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet
-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-c
lient llmnr-tcp llmnr-udp managesieve matrix mdns memcache minecraft m
inidlna mndp mongodb mosh mountd mpd mqtt mqtt-tls ms-wbt mssql murmur
mysql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashbo
ard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imagei
o ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwe
bapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporte
r proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel rad
ius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd
rtsp salt-master samba samba-client samba-dc sane settlers-history-col
lection sip sips slimevr slp smtp smtp-submission smtps snmp snmptls s
nmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh
-custom statsrv steam-lan-transfer steam-streaming stellaris stronghol
d-crusader stun stuns submission supertuxkart svdrp svn syncthing sync
thing-gui syncthing-relay synergy syscomlan syslog syslog-tls telnet t
entacle terraria tftp tile38 tinc tor-socks transmission-client turn t
urns upnp-client vdsm vnc-server vrrp warpinator wbem-http wbem-https
wireguard ws-discovery ws-discovery-client ws-discovery-host ws-discov
ery-tcp ws-discovery-udp wsdd wsdd-http wsman wsmans xdmcp xmpp-bosh x
mpp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zab
bix-server zabbix-trapper zabbix-web-service zero-k zerotier
[dmmosharov@server.dmmosharov.net ~]$ firewall-cmd --add-service=mountd
--add-service=rpc-bind
success
[dmmosharov@server.dmmosharov.net ~]$ sudo -i
[root@server.dmmosharov.net ~]# firewall-cmd --add-service=mountd --ad
d-service=rpc-bind --permanent
success
[root@server.dmmosharov.net ~]# firewall-cmd --reload
success
[root@server.dmmosharov.net ~]#

```

Настройка firewalld для работы служб mountd и rpc-bind

На клиентской машине выполняем команду `sudo showmount -e server.dmmosharov.net`, запрашивающую у NFS-сервера список экспортируемых каталогов; в ответ отображается экспорт `/srv/nfs` для всех клиентов (\*), что подтверждает доступность NFS-сервера через настроенный межсетевой экран (рис. [-@fig:014]).

```

[dmmosharov@client.dmmosharov.net ~]$ showmount -e server.dmmosharov.net
Export list for server.dmmosharov.net:
/srv/nfs *
[dmmosharov@client.dmmosharov.net ~]$

```

Проверка экспорта NFS-ресурса командой showmount

На клиенте создаём каталог `/mnt/nfs`, после чего выполняем монтирование экспортируемого с сервера каталога `server.dmmosharov.net/srv/nfs` в эту точку командой `sudo mount`; затем команда `mount` без параметров выводит список всех смонтированных файловых систем, в котором присутствует запись о подключённом ресурсе типа `nfs4` с указанными параметрами монтирования, что подтверждает успешное подключение удалённого раздела (рис. [-@fig:015]).

```
[root@client.dmmosharov.net ~]# mkdir -p /mnt/nfs
[root@client.dmmosharov.net ~]# mount server.dmmosharov.net:/srv/nfs /mnt/nfs
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@client.dmmosharov.net ~]# mount
/dev/mapper/r1_l8-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=408474,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,nodelegate,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=742200k,nr_inodes=819200,mode=755,inode64)
sellinuxfs on /sys/fs/sellinux type sellinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=30,pgpr=1,timeout=0,pipeproto=5,maxproto=5,direct,pipe_ino=3978)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (rw,nosuid,nodev,noexec,relatime,nosyfollow,seclabel,size=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,errors=remount-ro)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
vagrant on /vagrant type vboxsf (rw,nosuid,relatime,iocharset=utf8,uid=1000,gid=1000)
vagrant on /vagrant type vboxsf (rw,nosuid,relatime,iocharset=utf8,uid=1000,gid=1000,_netdev)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=372136k,nr_inodes=93034,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=372136k,nr_inodes=93034,mode=700,uid=1001,gid=1001,inode64)
server.dmmosharov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=524288,wsize=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
[root@client.dmmosharov.net ~]#
```

Монтирование каталога NFS на клиенте и проверка через mount

Для настройки автоматического монтирования удаленного ресурса при запуске операционной системы отредактируем файл /etc/fstab на клиенте. В конец файла была добавлена запись server.dmmosharov.net:/srv/nfs /mnt/nfs nfs \_netdev 0 0. Опция \_netdev здесь указывает системе, что монтирование зависит от наличия сети, и предотвращает попытки монтирования до поднятия сетевых интерфейсов (рис. [-@fig:016]).

```
GNU nano 8.1 /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 25 17:14:02 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=d3e7b147-c1d6-424e-819d-7a2628a132ec / xfs defaults 0 0
UUID=86c00fee-6d70-4d3e-9d8f-62dca7832f5 /boot xfs defaults 0 0
UUID=3D80-501E /boot/efi vfat umask=0077,shortname=winnt 0 2
UUID=e7ad08e1-c163-42a6-8723-3d23736bc2c9 /home xfs defaults 0 0
UUID=84be6eba-1981-4517-8c07-547bd3127c94 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.dmmosharov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Настройка автоматического монтирования в /etc/fstab

После сохранения изменений проверим статус службы, отвечающей за монтирование удаленных файловых систем. Команда systemctl status remote-fs.target показала, что цель (target) активна и загружена, что означает готовность системы к автоматическому монтированию сетевых ресурсов (рис. [-@fig:017]).

```
[root@client.dmmosharov.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 2026-02-01 20:15:30 UTC; 1min 39s ago
   Invocation: dce231fid0914acba09d279b9f267c3e
             Docs: man:systemd.special(7)

Feb 01 20:15:30 client.dmmosharov.net systemd[1]: Reached target remote-fs.target - Remote File Systems.
[root@client.dmmosharov.net ~]#
```

Проверка статуса remote-fs.target

После перезагрузки клиента убедимся, что ресурс действительно подключен. Команда mount выводит список всех точек монтирования. В выводе присутствует строка,

подтверждающая, что ресурс `server.dmmosharov.net:/srv/nfs` смонтирован в `/mnt/nfs` с типом файловой системы `nfs4` (версия 4.2) и правами `rw` (чтение и запись), что соответствует успешному выполнению настройки (рис. [-@fig:018]).

```

[root@client.dmmosharov.net ~]# mount
/dev/mapper/l1_10-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
devpts on /dev/pts type devpts (rw,nosuid,nodev,seclabel,inode64)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,exec,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,nfsdelegate,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,nodev=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=744284k,nr_inodes=819200,mode=755,inode64)
sellinuxfs on /sys/fs/sellinux type sellinuxfs (rw,nosuid,nodev,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=3904)
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)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
tracefs on /sys/kernel/trace type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (rw,nosuid,nodev,noexec,relatime,nosymlink,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
/dev/sd2 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,errors=remount-ro)
/dev/mapper/l1_10-home on /home type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
server.dmmosharov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsiz=524288,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)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=772140k,nr_inodes=93835,mode=700,uid=1000,gid=1000,inode64)
gravid-fuse on /run/user/1000/grafs type fuse.gravid-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=772140k,nr_inodes=93835,mode=700,uid=1001,gid=1001,inode64)
[root@client.dmmosharov.net ~]#

```

Проверка подключенных ресурсов командой `mount`

Далее перейдём к выполнению задания по подключению каталога с контентом веб-сервера к дереву NFS. На сервере создадим каталог `/srv/nfs/www` и выполним `bind`-монтирование реального каталога веб-сервера `/var/www/` в созданную директорию с помощью команды `mount -o bind`. Это позволяет сделать содержимое `/var/www` доступным внутри структуры NFS, не перемещая сами файлы (рис. [-@fig:019]).

```

[root@server.dmmosharov.net ~]# mkdir -p /srv/nfs/www
[root@server.dmmosharov.net ~]# mount -o bind /var/www/ /srv/nfs/www/
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@server.dmmosharov.net ~]# ls /srv/fs
ls: cannot access '/srv/fs': No such file or directory
[root@server.dmmosharov.net ~]# ls /srv/nfs
www
[root@server.dmmosharov.net ~]# █

```

`Bind`-монтирование каталога веб-сервера на сервере NFS

После выполнения монтирования на сервере, вернёмся на клиентскую машину и проверим содержимое смонтированного каталога `/mnt/nfs`. Команда `ls` показала наличие директории `www`, что подтверждает корректное отображение структуры каталогов сервера на клиенте через NFS (рис. [-@fig:020]).

```

[root@client.dmmosharov.net ~]# ls /mnt/nfs
www
[root@client.dmmosharov.net ~]# █

```

Проверка отображения нового каталога на клиенте

На сервере в файл `/etc/exports` добавляем запись для экспорта каталога веб-сервера. Указываем путь `/srv/nfs/www` и разрешаем доступ для подсети `192.168.0.0/16` с правами на чтение и запись (`rw`) (рис. [-@fig:021]).

```

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

```

### Редактирование файла /etc/exports на сервере

Применяем изменения и экспортируем указанные каталоги с помощью утилиты `exportfs` с флагом `-r` (рис. [-@fig:022]).

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

Экспорт каталогов командой `exportfs`

На клиентской машине проверяем содержимое смонтированного ресурса `/mnt/nfs`. Убеждаемся, что каталог `www` теперь отображается и доступен (рис. [-@fig:023]).

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

Проверка наличия каталога `www` на клиенте

Для того чтобы привязка (`bind`) каталога `/var/www` к `/srv/nfs/www` сохранялась после перезагрузки, добавляем соответствующую строку конфигурации в конец файла `/etc/fstab` на сервере (рис. [-@fig:024]).

```
GNU nano 8.1 /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 25 17:14:02 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/di>
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for mor>
#
# After editing this file, run 'systemctl daemon-reload' to update sy>
# units generated from this file.
#
UUID=d3e7b147-c1d6-424e-819d-7a2628a132ec / xfs
UUID=86c00fec-6d70-4d3e-9d8f-62dca78372f5 /boot xfs
UUID=3D80-501E /boot/efi vfat umask=0077,sho
UUID=e7ad08e1-c163-42a6-8723-3d23736bc2c9 /home xfs
UUID=84ebe6ba-1981-4517-8c07-547bd3127c94 none swap
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not m>
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
var/www /srv/nfs/www none bind 0 0
```

Добавление `bind`-монтирования в `/etc/fstab`

После настройки автоматического монтирования повторно выполняем команду `exportfs -r` на сервере для обновления таблицы экспорта (рис. [-@fig:025]).

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

Повторный запуск `exportfs` на сервере

Проверяем содержимое примонтированных каталогов на клиентской машине. В каталоге `/mnt/nfs/` успешно отображается директория `www`, содержащая контент веб-сервера (рис. [-@fig:026]).



```

Last login: Sun Feb 1 20:16:32 UTC 2026 on pts/0
[dmmosharov@client.dmmosharov.net ~]$ ls /mnt/nfs
www
[dmmosharov@client.dmmosharov.net ~]$ ls /mnt/nfs/www
cgi-bin  html
[dmmosharov@client.dmmosharov.net ~]$

```

Проверка доступа к каталогу www на клиенте

На сервере создаём рабочий каталог common в домашней директории пользователя и создаём в нём тестовый файл. Затем создаём соответствующую директорию в дереве NFS (/srv/nfs/home/dmmosharov) и выполняем монтирование методом bind для объединения этих каталогов (рис. [-@fig:027]).

```

[dmmosharov@server.dmmosharov.net ~]$ mkdir -p -m 700 ~/common
[dmmosharov@server.dmmosharov.net ~]$ cd ~/common
[dmmosharov@server.dmmosharov.net common]$ touch dmmosharov@server.txt
[dmmosharov@server.dmmosharov.net common]$ sudo mkdir -p /srv/nfs/home/
/dmmosharov
[sudo] password for dmmosharov:
[dmmosharov@server.dmmosharov.net common]$ mount -o bind /home/dmmosha
rov/common /srv/nfs/home/dmmosharov
mount: /srv/nfs/home/dmmosharov: must be superuser to use mount.
dmesg(1) may have more information after failed mount system ca
ll.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[dmmosharov@server.dmmosharov.net common]$ sudo mount -o bind /home/dm
mosharov/common /srv/nfs/home/dmmosharov
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[dmmosharov@server.dmmosharov.net common]$

```

Подготовка каталога пользователя и bind-монтирование на сервере

Редктируем файл конфигурации /etc/exports на сервере. Добавляем запись для экспорта домашнего каталога пользователя с правами на чтение и запись для локальной сети (рис. [-@fig:028]).

```

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

```

Настройка экспорта каталога пользователя

Для сохранения настроек монтирования после перезагрузки сервера вношу изменения в файл /etc/fstab. Добавляю записи для bind-монтирования каталогов /var/www и /home/dmmosharov/common в соответствующие точки монтирования дерева NFS (рис. [-@fig:029]).

```

GNU nano 8.1 /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 25 17:14:02 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/di>
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for mo>
#
# After editing this file, run 'systemctl daemon-reload' to update sy>
# units generated from this file.
#
UUID=d3e7b147-c1d6-424e-819d-7a2628a132ec / xfs>
UUID=86c00fec-6d70-4d3e-9d8f-62dca78372f5 /boot xfs>
UUID=3D80-501E /boot/efi vfat umask=0077,sh>
UUID=e7ad08e1-c163-42a6-8723-3d23736bc2c9 /home xfs>
UUID=84ebe6ba-1981-4517-8c07-547bd3127c94 none swa>
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not m>
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
var/www /srv/nfs/www none bind 0 0
/home/dmmosharov /common /srv/nfs/home/dmmosharov none bind 0 0

```

Настройка автоматического bind-монтирования в fstab

После обновления таблицы экспорта на сервере (exportfs -r) перехожу на клиентскую машину и проверяю содержимое /mnt/nfs/. Вижу, что появилась директория home, а в ней — каталог пользователя dmmosharov (рис. [-@fig:030]).

```

[dm-mosharov@client.dmmosharov.net ~]$ ls /mnt/nfs
home www
[dm-mosharov@client.dmmosharov.net ~]$ ls /mnt/nfs/home
dmmosharov
[dm-mosharov@client.dmmosharov.net ~]$

```

Проверка доступа к каталогу home на клиенте

На клиенте в каталоге /mnt/nfs/home/dmmosharov создадим файл dmmosharov@client.txt с произвольным содержимым. При попытке выполнить действия в этом каталоге от имени суперпользователя получен отказ в доступе (Permission denied), что свидетельствует о корректной работе настройки безопасности NFS, которая по умолчанию запрещает удаленному root-пользователю иметь привилегии root на файловой системе сервера (рис. [-@fig:031]).

```

[dm-mosharov@client.dmmosharov.net ~]$ cd /mnt/nfs/home/dmmosharov
[dm-mosharov@client.dmmosharov.net dmmosharov]$ touch dmmosharov@client.txt
[dm-mosharov@client.dmmosharov.net dmmosharov]$ echo '123123' >> dmmosharov@client.txt
[dm-mosharov@client.dmmosharov.net dmmosharov]$ sudo -i
[sudo] password for dmmosharov:
[root@client.dmmosharov.net ~]# cd /mnt/nfs/home/dmmosharov
-bash: cd: /mnt/nfs/home/dmmosharov: Permission denied
[root@client.dmmosharov.net ~]#

```

Проверка доступа к смонтированному каталогу пользователя на клиенте

На сервере проверим наличие созданного на клиенте файла. Файл dmmosharov@client.txt успешно отображается в каталоге /home/dmmosharov/common, содержимое файла совпадает с тем, что было записано на клиенте (рис. [-@fig:032]).

```

[root@server.dmmosharov.net ~]# logout
[dm-mosharov@server.dmmosharov.net common]$ ls
dmmosharov@client.txt dmmosharov@server.txt
[dm-mosharov@server.dmmosharov.net common]$ cat dmmosharov@client.txt
123123
[dm-mosharov@server.dmmosharov.net common]$

```

Проверка изменений в каталоге common на сервере

Далее выполним настройку provision. На сервере в каталоге /vagrant/provision/server создадим необходимую структуру каталогов, скопируем файл конфигурации /etc/exports и создадим исполняемый файл скрипта nfs.sh (рис. [-@fig:033]).

```
[dmmosharov@server.dmmosharov.net server]$ sudo mkdir -p /vagrant/provision/server/nfs/etc
[sudo] password for dmmosharov:
Sorry, try again.
[sudo] password for dmmosharov:
[dmmosharov@server.dmmosharov.net server]$ sudo cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[dmmosharov@server.dmmosharov.net server]$ cd /vagrant/provision/server
[dmmosharov@server.dmmosharov.net server]$ touch nfs.sh
[dmmosharov@server.dmmosharov.net server]$ chmod +x nfs.sh
[dmmosharov@server.dmmosharov.net server]$ sudo nano nfs.sh
```

Подготовка каталогов и создание скрипта nfs.sh на сервере

В файл nfs.sh на сервере запишем скрипт, автоматизирующий все ранее выполненные действия: установку пакетов, копирование конфигурационных файлов, настройку межсетевого экрана и SELinux, создание каталогов, bind-монтирование и запуск служб (рис. [-@fig:034]).

```
dmmosharov@server:/vagrant/provision/server - s...
/vagrant/provision/server
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/dmmosharov
mkdir -p -m 700 /home/dmmosharov/common
chown dmmosharov:dmmosharov /home/dmmosharov/common
mount -o bind /home/dmmosharov/common /srv/nfs/home/dmmosharov
echo "/home/dmmosharov/common /srv/nfs/home/dmmosharov none bind 0 0" >> /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld
```

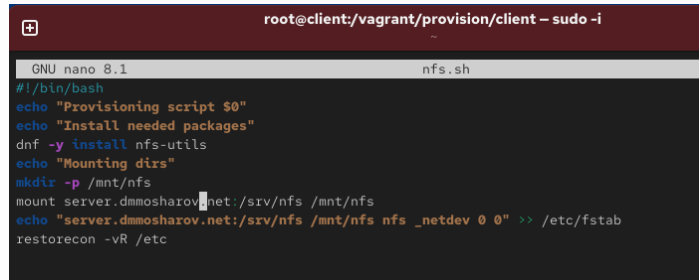
Содержимое скрипта автоматической настройки сервера

Аналогичные действия выполним для клиентской машины. В каталоге /vagrant/provision/client создадим и сделаем исполняемым файл скрипта настройки nfs.sh (рис. [-@fig:035]).

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

### Создание скрипта nfs.sh на клиенте

Открываем его на редактирование и прописываем скрипт, который устанавливает необходимый пакет nfs-utils, создает каталог /mnt/nfs, монтирует удаленный ресурс и добавляет соответствующую запись в /etc/fstab для автоматического монтирования (рис. [-@fig:036]).

A terminal window titled 'root@client:/vagrant/provision/client - sudo -i' showing the GNU nano 8.1 editor editing 'nfs.sh'. The script content is as follows:

```
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils
echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.dmmosharov.net:/srv/nfs /mnt/nfs
echo "server.dmmosharov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

### Скрипт настройки NFS на клиенте

Для автоматического запуска созданных скриптов во время загрузки виртуальных машин необходимо внести изменения в основной конфигурационный файл Vagrantfile. В соответствующие разделы конфигураций для сервера и клиента добавляем инструкции vm.provision для запуска файлов nfs.sh (рис. [-@fig:037]).

```

122
123     server.vm.provision "server nfs",
124         type: "shell",
125         preserve_order: true,
126         path: "provision/server/nfs.sh"
127     end
128
129
130 ## Client configuration
131 config.vm.define "client", autostart: false do |client|
132     client.vm.box = "rockylinux10"
133     client.vm.hostname = 'client'
134
135     client.vm.boot_timeout = 1440
136
137     client.ssh.insert_key = false
138     client.ssh.username = 'vagrant'
139     client.ssh.password = 'vagrant'
140
141     client.vm.network :private_network,
142         type: "dhcp",
143         virtualbox____intnet: true
144
145     client.vm.provider :virtualbox do |virtualbox|
146         virtualbox.customize ["modifyvm", :id, "--vrde", "on"]
147         virtualbox.customize ["modifyvm", :id, "--vrdeport", "3392"]
148     end
149
150     client.vm.provision "client dummy",
151         type: "shell",
152         preserve_order: true,
153         path: "provision/client/01-dummy.sh"
154
155     client.vm.provision "client routing",
156         type: "shell",
157         preserve_order: true,
158         run: "always",
159         path: "provision/client/01-routing.sh"
160
161     client.vm.provision "client mail",
162         type: "shell",
163         preserve_order: true,
164         path: "provision/client/mail.sh"
165
166     client.vm.provision "client ntp",
167         type: "shell",
168         preserve_order: true,
169         path: "provision/client/ntp.sh"
170
171     client.vm.provision "client nfs",
172         type: "shell",
173         preserve_order: true,
174         path: "provision/client/nfs.sh"
175     end
176 end

```

Добавление запуска скриптов в Vagrantfile

## Выводы

В результате выполнения лабораторной работы были получены навыки работы с nfs и сетевыми хранилищами, а так же их настройка