

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: Махорин Иван Сергеевич

Студ. билет № 1032211221

Группа: НПИбд-02-21

МОСКВА

2023 г.

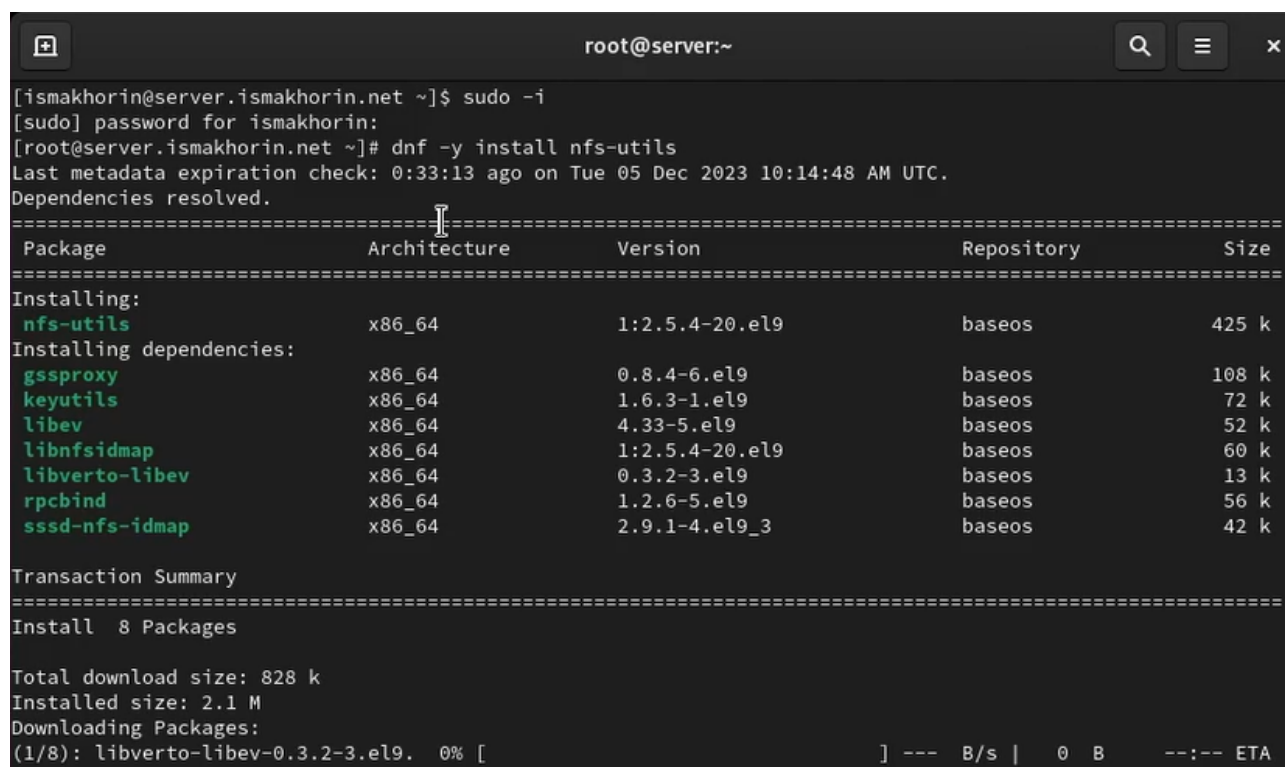
Цель работы:

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
```



```
root@server:~  
[ismakhorin@server.ismakhorin.net ~]$ sudo -i  
[sudo] password for ismakhorin:  
[root@server.ismakhorin.net ~]# dnf -y install nfs-utils  
Last metadata expiration check: 0:33:13 ago on Tue 05 Dec 2023 10:14:48 AM UTC.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
nfs-utils	x86_64	1:2.5.4-20.el9	baseos	425 k
Installing dependencies:				
gssproxy	x86_64	0.8.4-6.el9	baseos	108 k
keyutils	x86_64	1.6.3-1.el9	baseos	72 k
libev	x86_64	4.33-5.el9	baseos	52 k
libnfsidmap	x86_64	1:2.5.4-20.el9	baseos	60 k
libverto-libev	x86_64	0.3.2-3.el9	baseos	13 k
rpcbind	x86_64	1.2.6-5.el9	baseos	56 k
sssd-nfs-idmap	x86_64	2.9.1-4.el9_3	baseos	42 k

```
Transaction Summary  
=====
```

Transaction Summary	
Install	8 Packages
Total download size:	828 k
Installed size:	2.1 M
Downloading Packages:	
(1/8): libverto-libev-0.3.2-3.el9.	0% [] --- B/s 0 B --- ETA

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
```

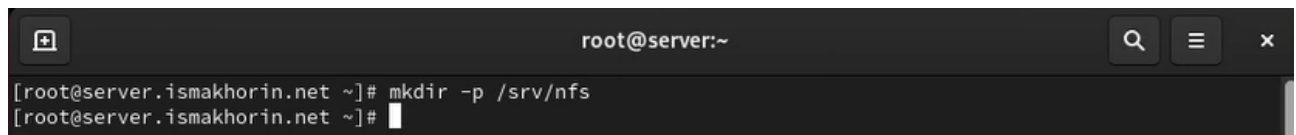


Рис. 1.2. Создание на сервере каталога, который предполагается сделать доступным всем пользователям сети (корень дерева NFS).

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

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

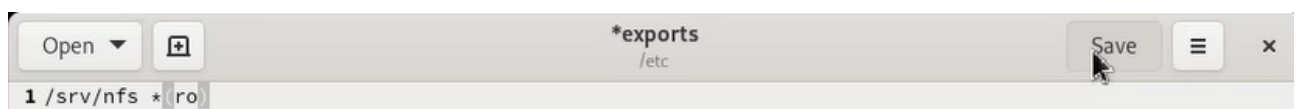


Рис. 1.3. Прописывание в файле `/etc/exports` подключаемого через NFS общего каталога с доступом только на чтение.

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

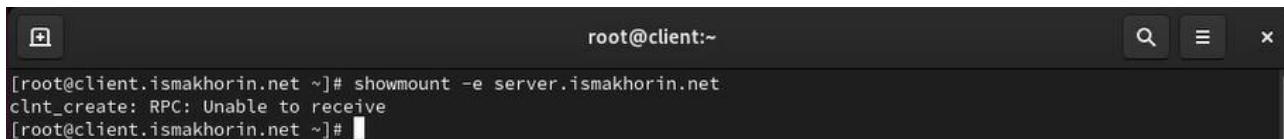
```
root@client:~
(2/8): rpcbind-1.2.6-5.el9.x86_64.rpm                69 kB/s | 56 kB    00:00
(3/8): libev-4.33-5.el9.x86_64.rpm                   58 kB/s | 52 kB    00:00
(4/8): sssd-nfs-idmap-2.9.1-4.el9_3.x86_64.rpm        339 kB/s | 42 kB    00:00
(5/8): gssproxy-0.8.4-6.el9.x86_64.rpm               877 kB/s | 108 kB   00:00
(6/8): libnfsidmap-2.5.4-20.el9.x86_64.rpm            742 kB/s | 60 kB    00:00
(7/8): keyutils-1.6.3-1.el9.x86_64.rpm               808 kB/s | 72 kB    00:00
(8/8): nfs-utils-2.5.4-20.el9.x86_64.rpm             2.1 MB/s | 425 kB   00:00
-----
Total                                           413 kB/s | 828 kB   00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Installing               : libnfsidmap-1:2.5.4-20.el9.x86_64      1/8
  Installing               : keyutils-1.6.3-1.el9.x86_64          2/8
  Installing               : libev-4.33-5.el9.x86_64               3/8
  Installing               : libverto-libev-0.3.2-3.el9.x86_64     4/8
  Installing               : gssproxy-0.8.4-6.el9.x86_64           5/8
  Running scriptlet: gssproxy-0.8.4-6.el9.x86_64                   5/8
  Running scriptlet: rpcbind-1.2.6-5.el9.x86_64                    6/8
  Installing               : rpcbind-1.2.6-5.el9.x86_64            6/8
  Running scriptlet: rpcbind-1.2.6-5.el9.x86_64                    6/8
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.

  Running scriptlet: nfs-utils-1:2.5.4-20.el9.x86_64              7/8
  Installing               : nfs-utils-1:2.5.4-20.el9.x86_64       7/8
  Running scriptlet: nfs-utils-1:2.5.4-20.el9.x86_64              7/8
  Installing               : sssd-nfs-idmap-2.9.1-4.el9_3.x86_64    8/8
  Running scriptlet: sssd-nfs-idmap-2.9.1-4.el9_3.x86_64          8/8
```

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.ismakhorin.net
```

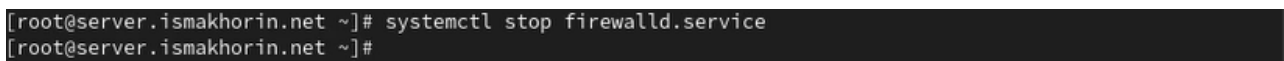
A terminal window titled 'root@client:~' with search, menu, and close icons in the top right. The command '[root@client.ismakhorin.net ~]# showmount -e server.ismakhorin.net' has been executed, resulting in the output 'clnt_create: RPC: Unable to receive'. The prompt '[root@client.ismakhorin.net ~]#' is visible on the next line.

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

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

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

```
systemctl stop firewalld.service
```

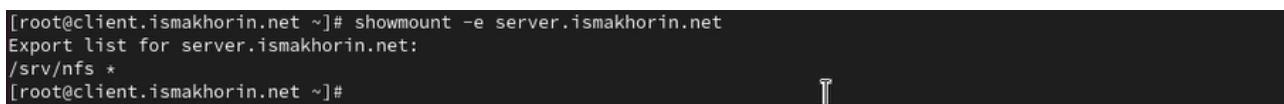
A terminal window showing the command '[root@server.ismakhorin.net ~]# systemctl stop firewalld.service' being executed. The output is empty, and the prompt '[root@server.ismakhorin.net ~]#' is shown on the next line.

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

A terminal window showing the command '[root@client.ismakhorin.net ~]# showmount -e server.ismakhorin.net' being executed. The output is 'Export list for server.ismakhorin.net:' followed by '/srv/nfs *' on the next line. The prompt '[root@client.ismakhorin.net ~]#' is visible at the bottom.

```
[root@client.ismakhorin.net ~]# showmount -e server.ismakhorin.net
Export list for server.ismakhorin.net:
/srv/nfs *
[root@client.ismakhorin.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.ismakhorin.net ~]# systemctl start firewalld
[root@server.ismakhorin.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.ismakhorin.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    78u    IPv4        241971      0t0      TCP *:sun
rpc (LISTEN)
systemd      1      root    80u    IPv6        241989      0t0      TCP *:sun
rpc (LISTEN)
cupsd        797     root     6u    IPv6        20396       0t0      TCP local
host:ipp (LISTEN)
cupsd        797     root     7u    IPv4        20397       0t0      TCP local
host:ipp (LISTEN)
httpd        15299   root     3u    sock        0,8        0t0      95966 proto
col: TCP
httpd        15299   root     4u    IPv6        95967       0t0      TCP *:htt
p (LISTEN)
httpd        15299   root     5u    sock        0,8        0t0      95978 proto
col: TCP
httpd        15299   root     6u    IPv6        95979       0t0      TCP *:htt
ps (LISTEN)
mariadb      20116   mysql    18u    IPv6       116407      0t0      TCP *:mys
ql (LISTEN)
mariadb      20116 20117 mariadb    mysql    18u    IPv6       116407      0t0      TCP *:mys
ql (LISTEN)
mariadb      20116 20118 mariadb    mysql    18u    IPv6       116407      0t0      TCP *:mys
ql (LISTEN)
mariadb      20116 20119 mariadb    mysql    18u    IPv6       116407      0t0      TCP *:mys
ql (LISTEN)
mariadb      20116 20120 mariadb    mysql    18u    IPv6       116407      0t0      TCP *:mys
ql (LISTEN)

```

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

```
root@server:~  
[root@server.ismakhorin.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    79u    IPv4        241980      0t0      UDP *:sun  
rpc  
systemd      1          root    81u    IPv6        241998      0t0      UDP *:sun  
rpc  
avahi-daemon 495        avahi    12u    IPv4        19204       0t0      UDP *:mdn  
s  
avahi-daemon 495        avahi    13u    IPv6        19205       0t0      UDP *:mdn  
s  
avahi-daemon 495        avahi    14u    IPv4        19206       0t0      UDP *:600  
26  
avahi-daemon 495        avahi    15u    IPv6        19207       0t0      UDP *:503  
18  
NetworkManager 10196      root    30u    IPv4        247377      0t0      UDP serve  
r.ismakhorin.net:bootpc->_gateway:bootps  
NetworkManager 10196 10207 gmain     root    30u    IPv4        247377      0t0      UDP serve  
r.ismakhorin.net:bootpc->_gateway:bootps  
NetworkManager 10196 10208 gdbus     root    30u    IPv4        247377      0t0      UDP serve  
r.ismakhorin.net:bootpc->_gateway:bootps  
dhcpcd       12670      dhcpcd    7u    IPv4        75054       0t0      UDP *:boo  
tps  
dhcpcd       12670      dhcpcd   20u    IPv4        75040       0t0      UDP *:382  
95  
dhcpcd       12670      dhcpcd   21u    IPv6        75041       0t0      UDP *:408  
32  
named        23082      named     6u    IPv4        247374      0t0      UDP serve
```

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

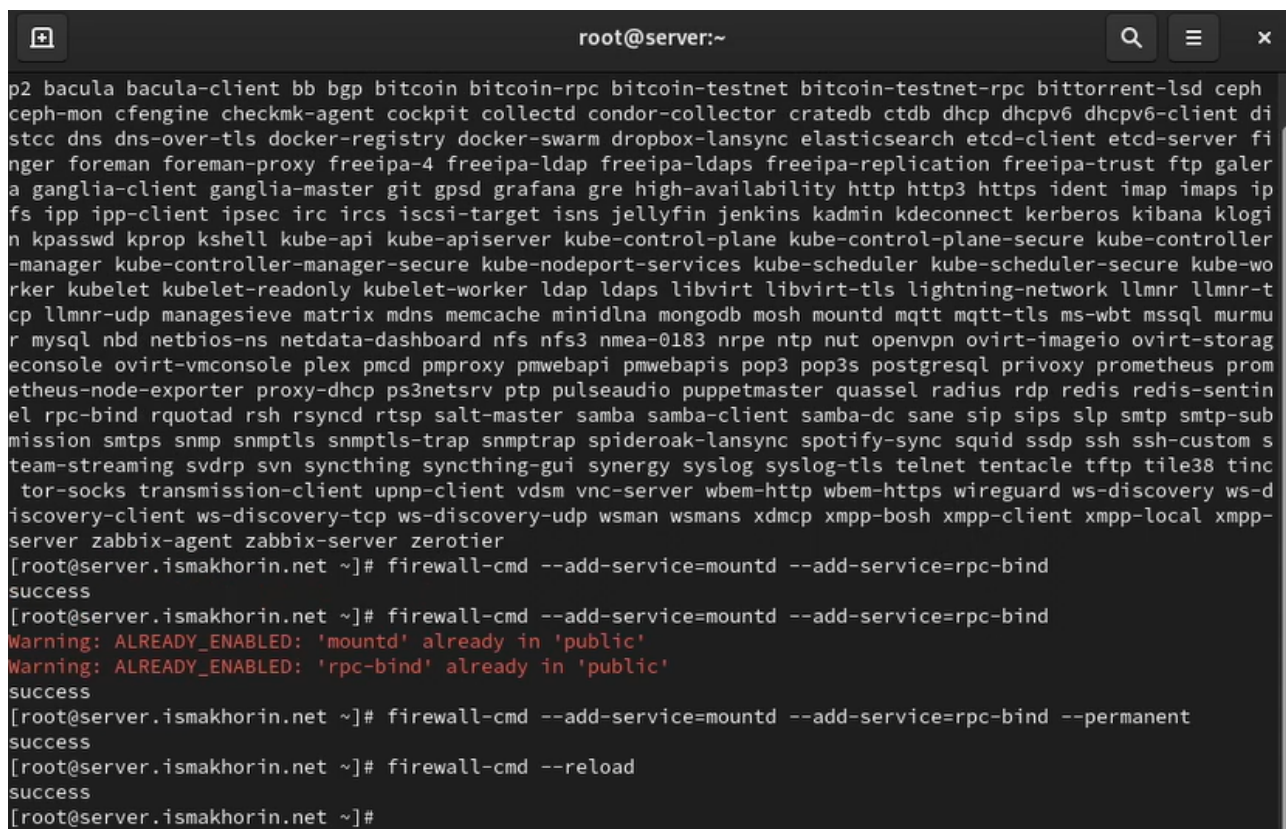
Добавим службы `rpc-bind` и `mountd` в настройки межсетевого экрана на сервере:

```
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
```

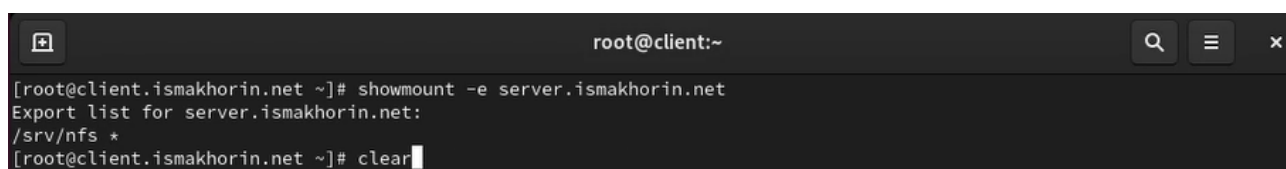



```
root@server:~
p2 bacula bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph
ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dhcp dhcpv6 dhcpv6-client di
stcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server fi
nger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galer
a ganglia-client ganglia-master git gpsd grafana gre high-availability http http3 https ident imap imaps ip
fs ipp ipp-client ipsec irc ircs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogi
n kpasswd kprop kshell kube-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-wo
rker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-t
cp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmu
r mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storag
econsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prom
etheus-node-exporter proxy-dhcp ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentin
el rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-sub
mission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom s
team-streaming svdrp svn syncthing syncthing-gui synergy syslog-tls telnet tentacle tftp tile38 tinc
tor-socks transmission-client upnp-client vdsms vnc-server wbem-http wbem-https wireguard ws-discovery ws-d
iscovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-
server zabbix-agent zabbix-server zerotier
[root@server.ismakhorin.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.ismakhorin.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
Warning: ALREADY_ENABLED: 'mountd' already in 'public'
Warning: ALREADY_ENABLED: 'rpc-bind' already in 'public'
success
[root@server.ismakhorin.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.ismakhorin.net ~]# firewall-cmd --reload
success
[root@server.ismakhorin.net ~]#
```

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

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

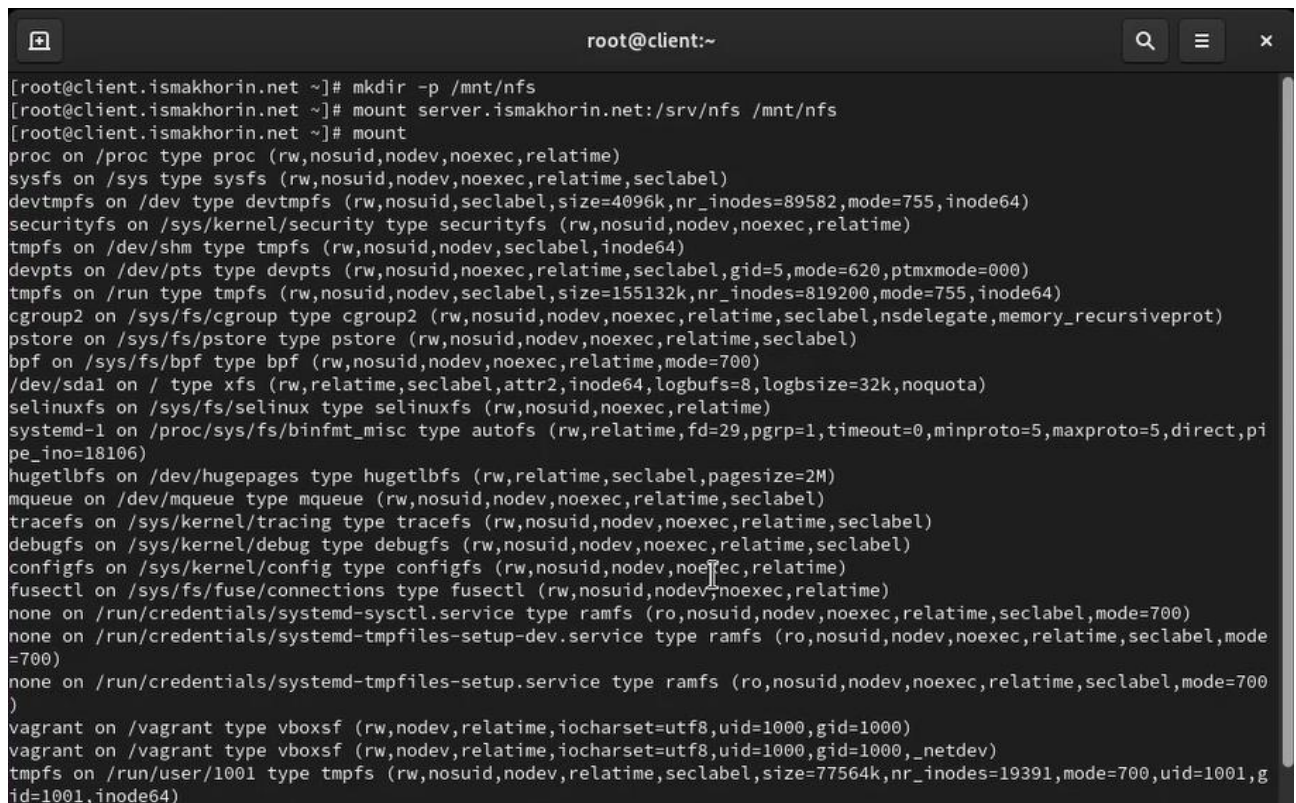
`showmount -e server.ismakhorin.net`



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

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:~  
[root@client.ismakhorin.net ~]# mkdir -p /mnt/nfs  
[root@client.ismakhorin.net ~]# mount server.ismakhorin.net:/srv/nfs /mnt/nfs  
[root@client.ismakhorin.net ~]# mount  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)  
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=89582,mode=755,inode64)  
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)  
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)  
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=155132k,nr_inodes=819200,mode=755,inode64)  
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)  
/dev/sdal on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)  
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)  
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18106)  
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)  
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)  
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)  
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)  
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)  
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)  
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)  
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)  
vagrant on /vagrant type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=1000,gid=1000)  
vagrant on /vagrant type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=1000,gid=1000,_netdev)  
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=77564k,nr_inodes=19391,mode=700,uid=1001,gid=1001,inode64)
```

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.ismakhorin.net:/srv/nfs /mnt/nfs nfs _netdev 0 0`

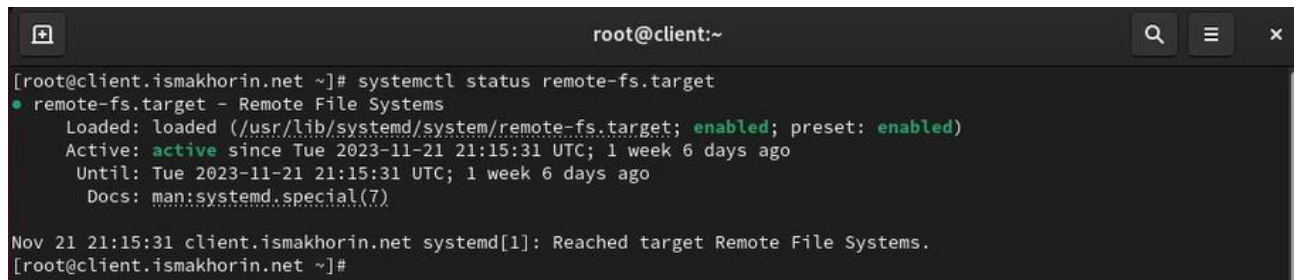


```
1  
2 #  
3 # /etc/fstab  
4 # Created by anaconda on Tue Nov 21 11:09:30 2023  
5 #  
6 # Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
8 #  
9 # After editing this file, run 'systemctl daemon-reload' to update systemd  
10 # units generated from this file.  
11 #  
12 UUID=6da01b9e-c688-4a88-83e1-03aaf1ca21fa / xfs defaults 0 0  
13 /swapfile none swap defaults 0 0  
14 #VAGRANT-BEGIN  
15 # The contents below are automatically generated by Vagrant. Do not modify.  
16 vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0  
17 #VAGRANT-END  
18 server.ismakhorin.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

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

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

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

A terminal window titled 'root@client:~' showing the command 'systemctl status remote-fs.target'. The output indicates that the 'remote-fs.target' is loaded and active. It was loaded from '/usr/lib/systemd/system/remote-fs.target' and is enabled. It became active on 'Tue 2023-11-21 21:15:31 UTC' and will remain active until 'Tue 2023-11-21 21:15:31 UTC'. The documentation path is 'man:systemd.special(7)'. At the bottom, a log message states: 'Nov 21 21:15:31 client.ismakhorin.net systemd[1]: Reached target Remote File Systems.'

```
root@client:~  
[root@client.ismakhorin.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 Tue 2023-11-21 21:15:31 UTC; 1 week 6 days ago  
     Until: Tue 2023-11-21 21:15:31 UTC; 1 week 6 days ago  
    Docs: man:systemd.special(7)  
  
Nov 21 21:15:31 client.ismakhorin.net systemd[1]: Reached target Remote File Systems.  
[root@client.ismakhorin.net ~]#
```

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

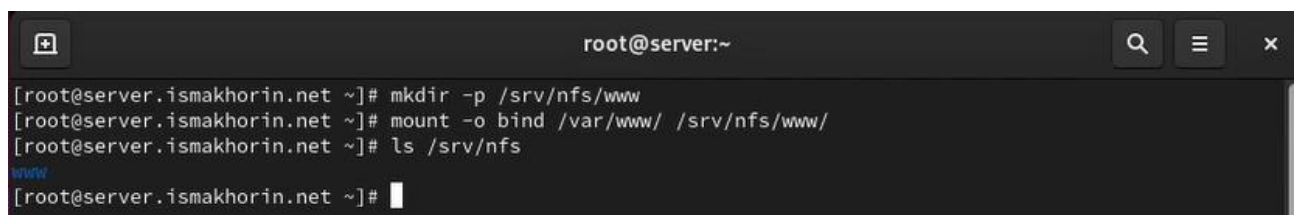
На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера:

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

После чего подмонтируем каталог web-сервера:

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

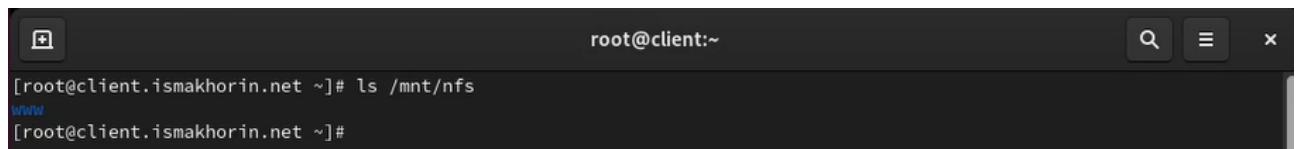
И на сервере проверим, что отображается в каталоге /srv/nfs (Рис. 3.1):

A terminal window titled 'root@server:~' showing three commands: 'mkdir -p /srv/nfs/www', 'mount -o bind /var/www/ /srv/nfs/www/', and 'ls /srv/nfs'. The output of 'ls' shows the 'www' directory.

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

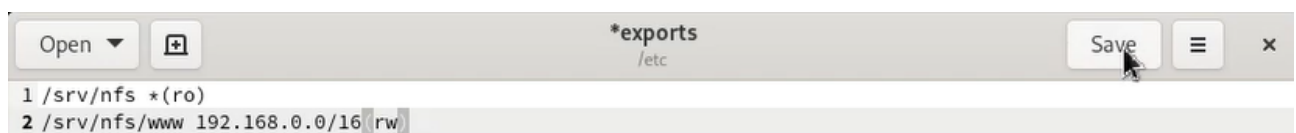


```
root@client:~  
[root@client.ismakhorin.net ~]# ls /mnt/nfs  
www  
[root@client.ismakhorin.net ~]#
```

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)
```

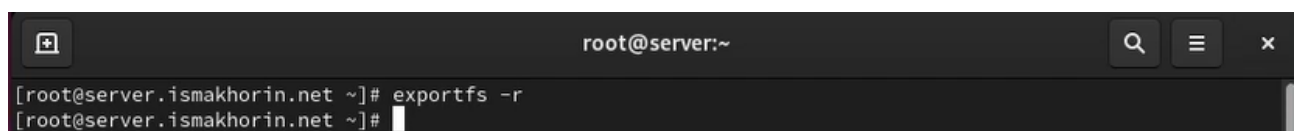


```
*exports  
/etc  
1 /srv/nfs *(ro)  
2 /srv/nfs/www 192.168.0.0/16(rw)
```

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

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

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

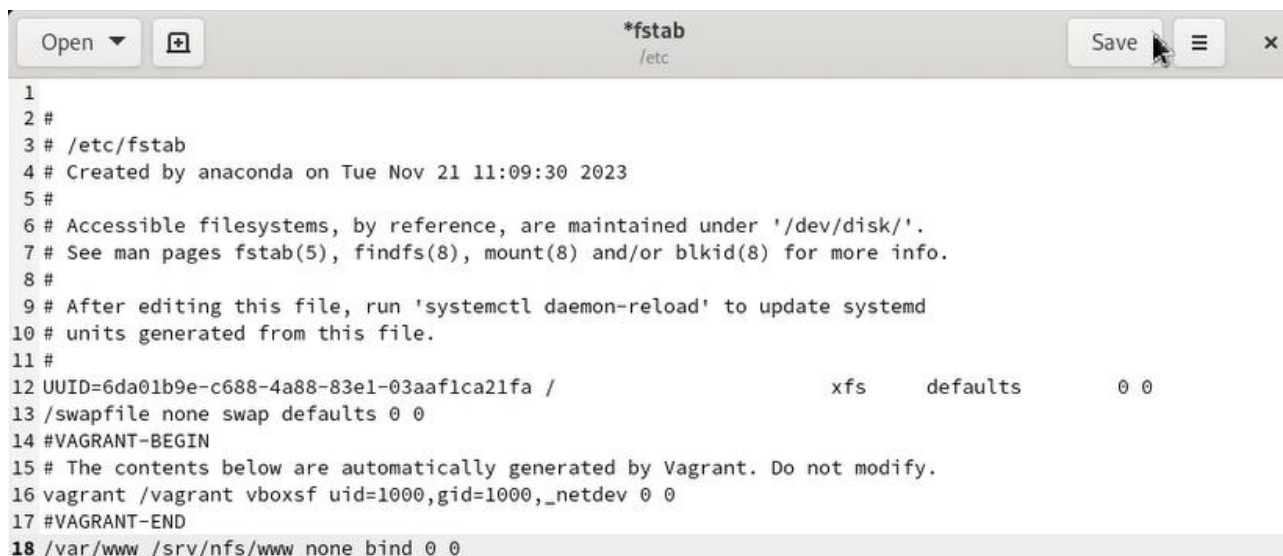


```
[root@client.ismakhorin.net ~]# ls /mnt/nfs  
www  
[root@client.ismakhorin.net ~]#
```

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
```



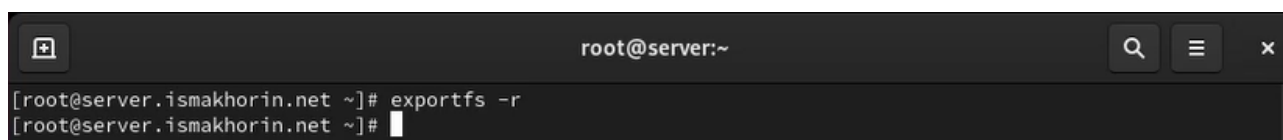
```
*fstab
/etc

1
2 #
3 # /etc/fstab
4 # Created by anaconda on Tue Nov 21 11:09:30 2023
5 #
6 # Accessible filesystems, by reference, are maintained under '/dev/disk/'.
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
8 #
9 # After editing this file, run 'systemctl daemon-reload' to update systemd
10 # units generated from this file.
11 #
12 UUID=6da01b9e-c688-4a88-83e1-03aaf1ca21fa /                xfs      defaults    0 0
13 /swapfile none swap defaults 0 0
14 #VAGRANT-BEGIN
15 # The contents below are automatically generated by Vagrant. Do not modify.
16 vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
17 #VAGRANT-END
18 /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):

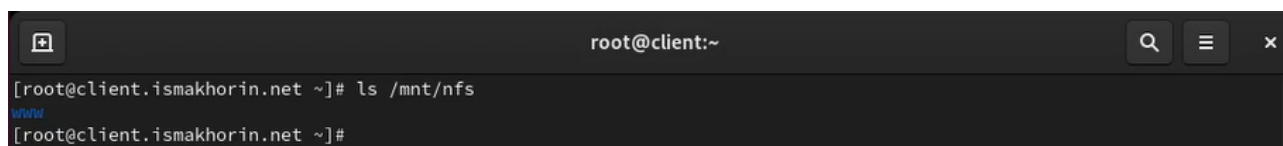
```
exportfs -r
```



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

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

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

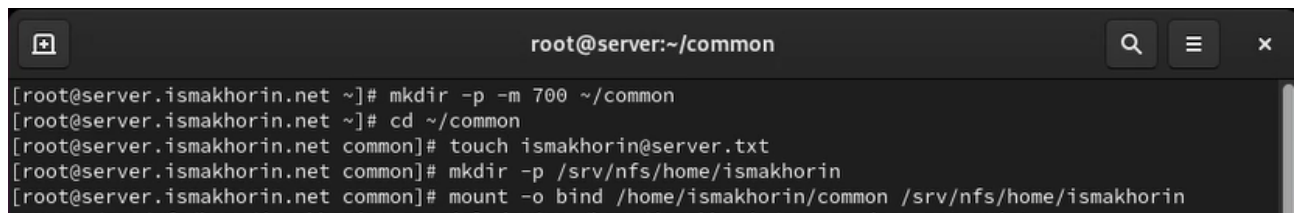


```
root@client:~
[root@client.ismakhorin.net ~]# ls /mnt/nfs
www
[root@client.ismakhorin.net ~]#
```

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

На сервере под пользователем ismakhorin в домашнем каталоге создадим каталог common с полными правами доступа только для этого пользователя, а в нём файл ismakhorin@server.txt. Далее создадим общий каталог для работы

пользователя ismakhorin по сети и подмонтируем каталог common пользователя ismakhorin в NFS (Рис. 4.1):

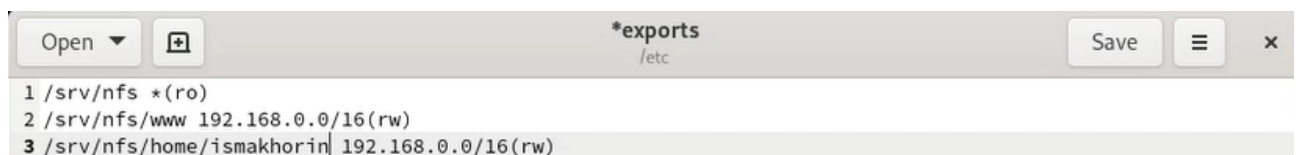


```
root@server:~/common
[root@server.ismakhorin.net ~]# mkdir -p -m 700 ~/common
[root@server.ismakhorin.net ~]# cd ~/common
[root@server.ismakhorin.net common]# touch ismakhorin@server.txt
[root@server.ismakhorin.net common]# mkdir -p /srv/nfs/home/ismakhorin
[root@server.ismakhorin.net common]# mount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

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

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

/srv/nfs/home/ismakhorin 192.168.0.0/16(rw)



```
*exports
/etc
1 /srv/nfs *(ro)
2 /srv/nfs/www 192.168.0.0/16(rw)
3 /srv/nfs/home/ismakhorin 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/ismakhorin/common /srv/nfs/home/ismakhorin none bind 0 0

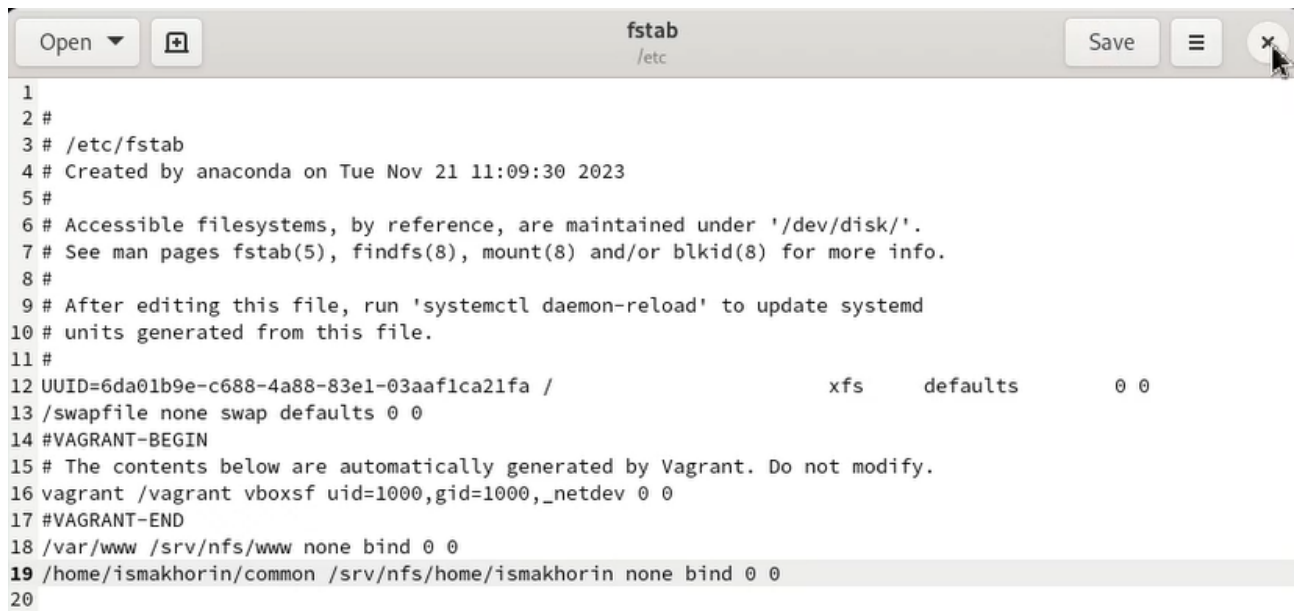


Рис. 4.3. Внесение изменения в файл /etc/fstab.

Повторно экспортируем каталоги (Рис. 4.4):

`exportfs -r`

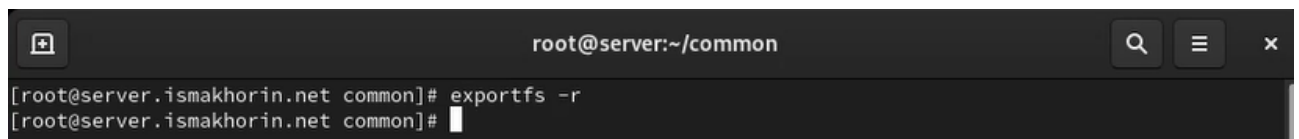


Рис. 4.4. Повторный экспорт каталогов.

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

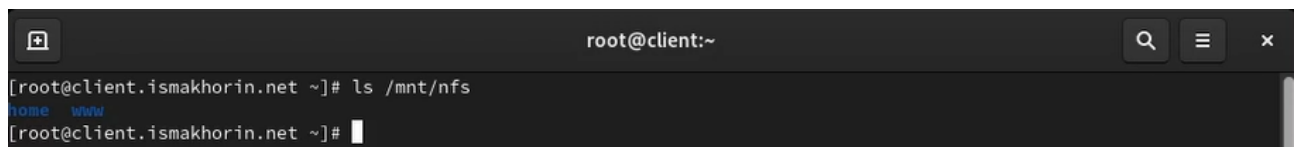
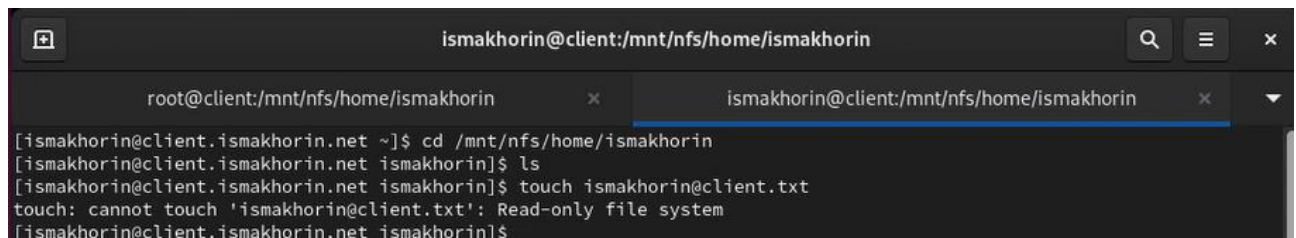


Рис. 4.5. Проверка на клиенте каталога /mnt/nfs.

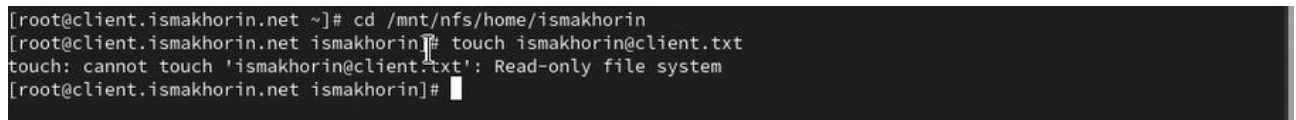
На клиенте под пользователем ismakhorin перейдём в каталог /mnt/nfs/home/ismakhorin и попробуем создать в нём файл ismakhorin@client.txt и

внести в него какие-либо изменения (Рис. 4.6). Попробуем также это проделать под пользователем root (Рис. 4.7):



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

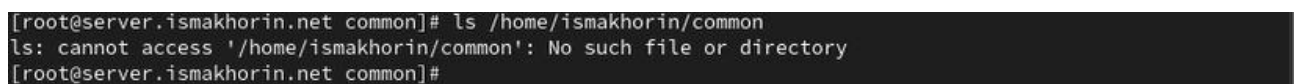
Рис. 4.6. Переход на клиенте под пользователем ismakhorin в каталог /mnt/nfs/home/ismakhorin и попытка создать в нём файл ismakhorin@client.txt.



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

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

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



```
[root@server.ismakhorin.net common]# ls /home/ismakhorin/common
ls: cannot access '/home/ismakhorin/common': No such file or directory
[root@server.ismakhorin.net common]#
```

Рис. 4.8. Просмотр наличия изменений на сервере в каталоге пользователя /home/ismakhorin/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):



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

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


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



```
nfs.sh
/vagrant/provision/server
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 echo "Install needed packages"
6 dnf -y install nfs-utils
7
8 echo "Copy configuration files"
9 cp -R /vagrant/provision/server/nfs/etc/* /etc
10
11 restorecon -vR /etc
12
13 echo "Configure firewall"
14 firewall-cmd --add-service nfs --permanent
15 firewall-cmd --add-service mountd --add-service rpc-bind --permanent
16 firewall-cmd --reload
17
18 echo "Tuning SELinux"
19 mkdir -p /srv/nfs
20 semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
21 restorecon -vR /srv/nfs
22
23 echo "Mounting dirs"
24 mkdir -p /srv/nfs/www
25 mount -o bind /var/www /srv/nfs/www
26 echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
27 mkdir -p /srv/nfs/home/user
28 mkdir -p /srv/nfs/home/user/common
```

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:/vagrant/provision/client
[root@client.ismakhorin.net ismakhorin]# cd /vagrant/provision/client
[root@client.ismakhorin.net client]# cd /vagrant/provision/client
[root@client.ismakhorin.net client]# touch nfs.sh
[root@client.ismakhorin.net client]# chmod +x nfs.sh
[root@client.ismakhorin.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):



```
Open  [icon]  nfs.sh  Save  [icon]
/vagrant/provision/client

1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 echo "Install needed packages"
6 dnf -y install nfs-utils
7
8 echo "Mounting dirs"
9 mkdir -p /mnt/nfs
10 mount server.user.net:/srv/nfs /mnt/nfs
11 echo "server.user.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
12 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):

```

82
83     server.vm.provision "server nfs",
84                         type: "shell",
85                         preserve_order: true,
86                         path: "provision/server/nfs.sh"
87

```

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

```

140
141     client.vm.provision "client nfs",
142                        type: "shell",
143                        preserve_order: true,
144                        path: "provision/client/nfs.sh"
145

```

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

Вывод:

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

Ответы на контрольные вопросы:

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

TCP и UDP порт 2049: Основной порт для NFS.

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

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

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

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

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