

Лабораторная работа №13

Настройка NFS

Беличева Дарья Михайловна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Настройка сервера NFSv4	6
3.2	Монтирование NFS на клиенте	10
3.3	Подключение каталогов к дереву NFS	12
3.4	Подключение каталогов для работы пользователей	14
3.5	Внесение изменений в настройки внутреннего окружения виртуальных машин	16
4	Выводы	19

Список иллюстраций

3.1	Установка пакетов	6
3.2	Редактирование файла	7
3.3	Настройка межсетевого экрана	7
3.4	Установка пакетов	8
3.5	Просмотр подмонтированных удаленных ресурсов	8
3.6	Подключение к удаленно смонтированному ресурсу	8
3.7	Задействованные службы при удаленном монтировании по протоколу TCP	9
3.8	Задействованные службы при удаленном монтировании по протоколу UDP	9
3.9	Настройка межсетевого экрана на сервере	9
3.10	Подключение к удаленно смонтированному ресурсу	10
3.11	Монтирование NFS на клиенте	10
3.12	Редактирование файла	11
3.13	Проверка наличия автоматического монтирования удалённых ресурсов	11
3.14	Проверка	12
3.15	Содержимое каталога	12
3.16	Содержимое каталога	12
3.17	Редактирование файла	13
3.18	Содержимое каталога	13
3.19	Редактирование файла	13
3.20	Содержимое каталога	14
3.21	Подключение каталогов для работы пользователей	14
3.22	Редактирование файла	15
3.23	Проверка содержимого каталога	15
3.24	Создание файла	15
3.25	Редактирование файла	17
3.26	Редактирование файла	18

1 Цель работы

Приобрести навыки настройки сервера NFS для удалённого доступа к ресурсам.

2 Задание

1. Установить и настроить сервер NFSv4.
2. Подмонтировать удалённый ресурс на клиенте.
3. Подключить каталог с контентом веб-сервера к дереву NFS.
4. Подключить каталог для удалённой работы вашего пользователя к дереву NFS.
5. Написать скрипты для Vagrant, фиксирующие действия по установке и настройке сервера NFSv4 во внутреннем окружении виртуальных машин `server` и `client`. Соответствующим образом внести изменения в `Vagrantfile`.

3 Выполнение лабораторной работы

3.1 Настройка сервера NFSv4

На сервере установим необходимое программное обеспечение: `dnf -y install nfs-utils`

```
[dmbelicheva@server.dmbelicheva.net ~]$ sudo -i
[sudo] password for dmbelicheva:
[root@server.dmbelicheva.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 9 - x86_64 22 kB/s | 36 kB 00:01
Rocky Linux 9 - BaseOS 2.3 kB/s | 4.1 kB 00:01
Rocky Linux 9 - BaseOS 2.2 MB/s | 2.2 MB 00:01
Rocky Linux 9 - AppStream 12 kB/s | 4.5 kB 00:00
Rocky Linux 9 - AppStream 6.5 MB/s | 7.4 MB 00:01
Rocky Linux 9 - Extras 8.4 kB/s | 2.9 kB 00:00
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
nfs-utils x86_64 1:2.5.4-20.el9 baseos 425 k
Upgrading:
libipa_hbac x86_64 2.9.1-4.el9_3.1 baseos 38 k
libldb x86_64 2.7.2-2.el9 baseos 180 k
libsmbclient x86_64 4.18.6-101.el9_3 baseos 74 k
libsss_certmap x86_64 2.9.1-4.el9_3.1 baseos 93 k
```

Рис. 3.1: Установка пакетов

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS): `mkdir -p /srv/nfs`

В файле `/etc/exports` пропишем подключаемый через NFS общий каталог с доступом только на чтение: `/srv/nfs *(ro)`



Рис. 3.2: Редактирование файла

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
```

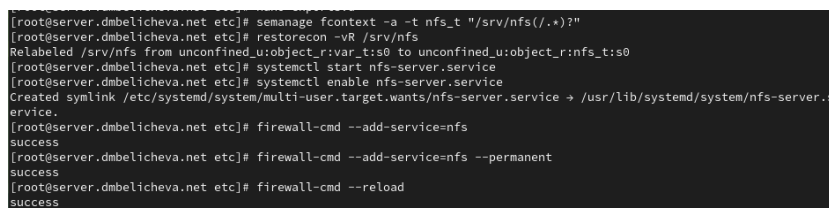


Рис. 3.3: Настройка межсетевого экрана

На клиенте установим необходимое для работы NFS программное обеспечение: `dnf -y install nfs-utils`

```
[dmbelicheva@client.dmbelicheva.net ~]$ sudo -i
[sudo] password for dmbelicheva:
[root@client.dmbelicheva.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 9 - x86_64 17 kB/s |
Extra Packages for Enterprise Linux 9 - x86_64 4.5 MB/s |
Rocky Linux 9 - BaseOS 1.4 kB/s |
Rocky Linux 9 - BaseOS 3.0 MB/s |
Rocky Linux 9 - AppStream 10 kB/s |
Rocky Linux 9 - AppStream 3.4 MB/s |
Rocky Linux 9 - Extras 7.8 kB/s |
Dependencies resolved.
=====
Package Architecture Version Repository
=====
Installing:
nfs-utils x86_64 1:2.5.4-20.el9 baseos
Upgrading:
libnfs  x86_64 2.9.1-4.el9_3.1 baseos
```

Рис. 3.4: Установка пакетов

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы (вместо user укажите свой логин):

`showmount -e server.dmbelicheva.net`

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

Рис. 3.5: Просмотр подмонтированных удаленных ресурсов

Попробуем на сервере остановить сервис межсетевого экрана: `systemctl stop firewalld.service`

Затем на клиенте вновь попробуем подключиться к удалённо смонтированному ресурсу: `showmount -e server.dmbelicheva.net`

```
clnt_create: RPC: Unable to receive
[root@client.dmbelicheva.net ~]# showmount -e server.dmbelicheva.net
Export list for server.dmbelicheva.net:
/srv/nfs *
```

Рис. 3.6: Подключение к удаленно смонтированному ресурсу

На сервере запустим сервис межсетевого экрана `systemctl start firewalld`

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

`lsof | grep TCP`

`lsof | grep UDP`


```

[root@server.dmbelicheva.net etc]# lsuf | grep TCP
lsuf: WARNING: can't stat() fuse.gvfs-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
systemd      1          root    92u    IPv4          38904      0t0      TCP *:sunrpc (LISTEN)
systemd      1          root    94u    IPv6          38922      0t0      TCP *:sunrpc (LISTEN)
cupsd        861        root    6u     IPv6          19573      0t0      TCP localhost:ipp (LISTEN)
cupsd        861        root    7u     IPv4          19574      0t0      TCP localhost:ipp (LISTEN)
sshd         871        root    3u     IPv4          19727      0t0      TCP *:down (LISTEN)
sshd         871        root    4u     IPv6          19734      0t0      TCP *:down (LISTEN)
sshd         871        root    5u     IPv4          19738      0t0      TCP *:ssh (LISTEN)

```

Рис. 3.7: Задействованные службы при удаленном монтировании по протоколу TCP

```
[root@server.dmbelicheva.net etc]# lsaf | grep UDP
lsaf: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
```

systemd	1	root	93u	IPv4	38913	0t0	UDP	*sunrpc
systemd	1	root	95u	IPv6	38921	0t0	UDP	*sunrpc
avahi-dae	563	avahi	12u	IPv4	18794	0t0	UDP	*mdns
avahi-dae	563	avahi	13u	IPv6	18795	0t0	UDP	*mdns
avahi-dae	563	avahi	14u	IPv4	18796	0t0	UDP	*34106
avahi-dae	563	avahi	15u	IPv6	18797	0t0	UDP	*34135
chronyd	590	chrony	5u	IPv4	18692	0t0	UDP	localhost:323
chronyd	590	chrony	6u	IPv6	18693	0t0	UDP	localhost:323
chronyd	590	chrony	7u	IPv4	18694	0t0	UDP	*ntp
named	901	named	6u	IPv4	21448	0t0	UDP	server.dmbelich

```
eva.net:domain
```

Рис. 3.8: Задействованные службы при удаленном монтировании по протоколу UDP

Добавим службы `grc-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.dbmlchiva.net]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capable amp amanda-client amanda-KS-client amqp ampps apcupsd audit ausweisapp2 bacula
bacula-client bdb bgp bitcoin bitcoind-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd cepb cepb-nom cephengine
ckmkm-agent cockpit collectd collector-collector cratedbd ctddb dhcp dnchp6 dnchpvc-client distcc dns-dns-over-tls docker-
registry docker-swarm drophox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4
freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gssdp grafana-9
rh-high-availability httpd http3 https ident imap imaps ipfs ipfs ipip-pclient ipsec irc ircs iscisi-target insn jellyfin
jenkins kadmin kdeconnect kerberos kibana klogon kpasword kprop kshell kube-api kube-api-server kube-control-plane kube-
control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kub-
be-scheduler-secure kube-worker kubelet kubelet-readyonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-netw-
ork mmrm lmnrr-tcp lmnrr-udp managesieve matrix mdms memcache minidna mongodb mos mountd mqtt mqtt-tls ms-wbt msssl
l murmur mysql nbd nbdtios-nb netdata-dashboard nfs nfsv3 nmca-0183 nrpe ntp npv openvpn ovirt-imgio ovirt-storageco-
nsole ovirt-vmsconsole pbs pmcd pmpoxy pmwebapi pmwebapis pop3 pop3s postgresql proxy prometheus prometheus-node-e-
xporter proxynet-cpu psnsetrps tcp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rsyncd rsh
rsyncd rtp salt-master samba samba-client samba-dc sane sip sipd sip smp smtp smtp-submission snmp snmpd snmptls smpls
s-trap snmptrap spideroak-lansync spotify-sync squid ssh sshd custom steam-streaming svdrp svn synching synthpro
g-ui synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client unpp-client vdsim vdsim
nserver wben-http wben-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsman-
xmldb-xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zoriot
```

```
[root@server.dbmlchiva.net]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.dbmlchiva.net]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.dbmlchiva.net]# firewall-cmd --reload
success
```

Рис. 3.9: Настройка межсетевого экрана на сервере

На клиенте проверим подключение удалённого ресурса: `showmount -e server.dmbelicheva.net`

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

Рис. 3.10: Подключение к удаленно смонтированному ресурсу

3.2 Монтирование NFS на клиенте

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

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

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

Проверим, что общий ресурс NFS подключён правильно: `mount`

```
[root@client.dmbelicheva.net ~]# mkdir -p /mnt/nfs
[root@client.dmbelicheva.net ~]# mount server.dmbelicheva.net:/srv/nfs /mnt/nfs
[root@client.dmbelicheva.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=114507,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=194916k,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,inode=17475)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,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)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (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=97456k,nr_inodes=24364,mode=700,uid=1001,inode64)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.dmbelicheva.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsz=131072,namlen=255,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
```

Рис. 3.11: Монтирование NFS на клиенте

На клиенте в конце файла `/etc/fstab` добавим следующую запись: `server.dmbelicheva.net:/srv/nfs nfs _netdev 0 0`

```
#
UUID=d648795f-bd46-4875-b95d-2593e1087e22 /                                xfs
/swapfile none swap defaults 0 0
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END

server.dmbelicheva.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рис. 3.12: Редактирование файла

На клиенте проверим наличие автоматического монтирования удалённых ресурсов при запуске операционной системы: `systemctl status remote-fs.target`

```
[root@client.dmbelicheva.net etc]# 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 Sat 2023-12-23 01:51:16 MSK; 26min ago
     Until: Sat 2023-12-23 01:51:16 MSK; 26min ago
    Docs: man:systemd.special(7)

Dec 23 01:51:16 client.dmbelicheva.net systemd[1]: Reached target Remote File Systems.
[root@client.dmbelicheva.net etc]# nano fstab
```

Рис. 3.13: Проверка наличия автоматического монтирования удалённых ресурсов

Перезапустим клиент и убедимся, что удалённый ресурс подключается автоматически.

```
[dmbelicheva@client.dmbelicheva.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=114507,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=194916k,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/sda1 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,inode=17478)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,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)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (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)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=1000,gid=1000)
server.dmbelicheva.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsz=131072,namlen=255,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1,_netdev)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=97456k,nr_inodes=24364,mode=700,uid=1000,gid=1001,inode64)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=1000,gid=1000,_netdev)
[dmbelicheva@client.dmbelicheva.net ~]$
```

Рис. 3.14: Проверка

3.3 Подключение каталогов к дереву NFS

На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера: `mkdir -p /srv/nfs/www`

Подмонтируем каталог веб-сервера: `mount -o bind /var/www/ /srv/nfs/www/`

На сервере проверим, что отображается в каталоге /srv/nfs.

```
[root@server.dmbelicheva.net etc]# mount -o bind /var/www/ /srv/nfs/www/
[root@server.dmbelicheva.net etc]# cd /srv/nfs
[root@server.dmbelicheva.net nfs]# ls
www
```

Рис. 3.15: Содержимое каталога

На клиенте посмотрим, что отображается в каталоге /mnt/nfs.

```
[dmbelicheva@client.dmbelicheva.net ~]$ cd /mnt/nfs
[dmbelicheva@client.dmbelicheva.net nfs]$ ls
www
[dmbelicheva@client.dmbelicheva.net nfs]$ cd www
```

Рис. 3.16: Содержимое каталога

На сервере в файле `/etc/exports` добавим экспорт каталога веб-сервера с удалённого ресурса: `/srv/nfs/www 192.168.0.0/16(rw)`

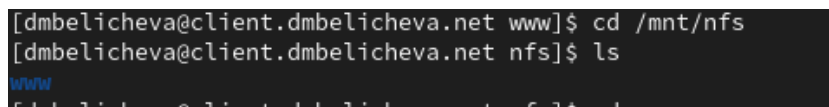


```
GNU nano 5.6.1
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
```

Рис. 3.17: Редактирование файла

Экспортируем все каталоги, упомянутые в файле `/etc/exports`: `exportfs -r`

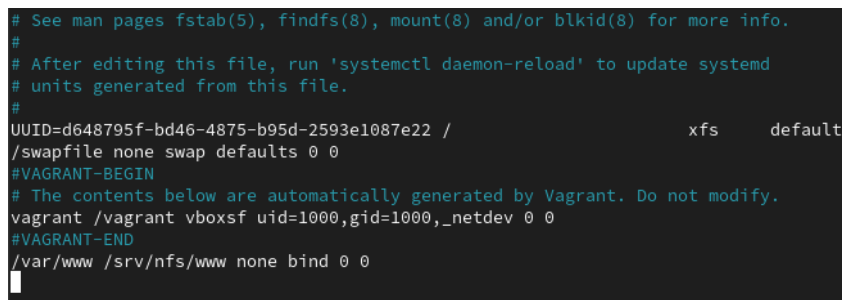
Проверим на клиенте каталог `/mnt/nfs`.



```
[dmbelicheva@client.dmbelicheva.net www]$ cd /mnt/nfs
[dmbelicheva@client.dmbelicheva.net nfs]$ ls
www
```

Рис. 3.18: Содержимое каталога

На сервере в конце файла `/etc/fstab` добавим следующую запись: `/var/www /srv/nfs/www none bind 0 0`



```
# 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=d648795f-bd46-4875-b95d-2593e1087e22 / xfs default
/swapfile none swap defaults 0 0
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
/var/www /srv/nfs/www none bind 0 0
```

Рис. 3.19: Редактирование файла

Повторно экспортируем каталоги, указанные в файле `/etc/exports`: `exportfs -r`

На клиенте проверим каталог `/mnt/nfs`.

```
[dmbelicheva@client.dmbelicheva.net nfs]$ ls
www
[dmbelicheva@client.dmbelicheva.net nfs]$
```

Рис. 3.20: Содержимое каталога

3.4 Подключение каталогов для работы пользователей

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

```
mkdir -p -m 700 ~/common
cd ~/common
touch dmbelicheva@server.txt
```

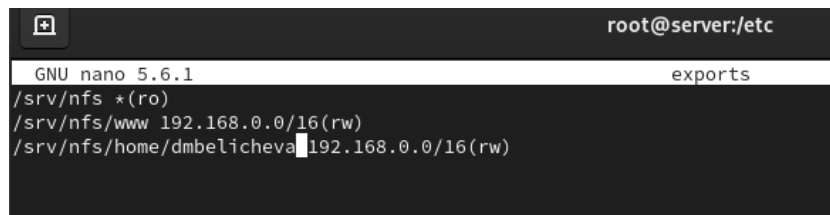
На сервере создадим общий каталог для работы пользователя dmbelicheva по сети: mkdir -p /srv/nfs/home/user

Подмонтируем каталог common пользователя dmbelicheva в NFS: mount -o bind /home/user/common /srv/nfs/home/user

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

Рис. 3.21: Подключение каталогов для работы пользователей

Подключим каталог пользователя в файле /etc/exports, прописав в нём (вместо user укажите свой логин): /srv/nfs/home/user 192.168.0.0/16(rw)



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

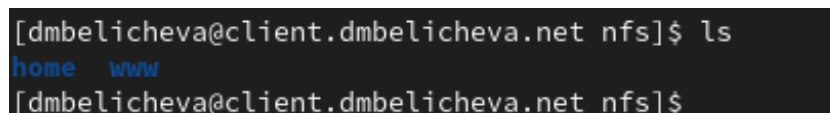
Рис. 3.22: Редактирование файла

Внесем изменения в файл /etc/fstab (вместо user укажите свой логин):

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

Повторно экспортируем каталоги: `exportfs -r`

На клиенте проверим каталог /mnt/nfs.



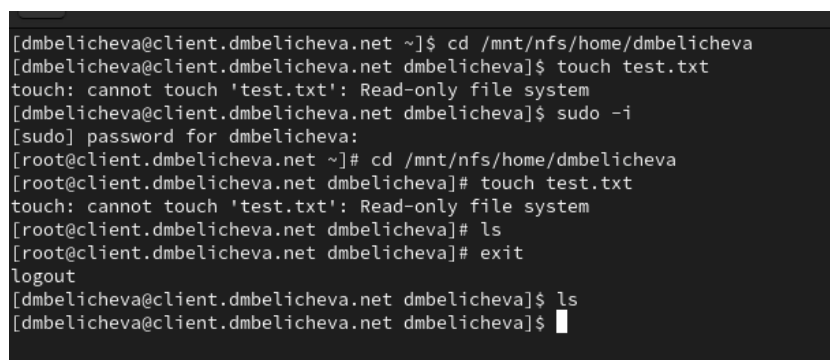
```
[dmbelicheva@client.dmbelicheva.net nfs]$ ls
home www
[dmbelicheva@client.dmbelicheva.net nfs]$
```

Рис. 3.23: Проверка содержимого каталога

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

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

```
touch user@client.txt
```



```
[dmbelicheva@client.dmbelicheva.net ~]$ cd /mnt/nfs/home/dmbelicheva
[dmbelicheva@client.dmbelicheva.net dmbelicheva]$ touch test.txt
touch: cannot touch 'test.txt': Read-only file system
[dmbelicheva@client.dmbelicheva.net dmbelicheva]$ sudo -i
[sudo] password for dmbelicheva:
[root@client.dmbelicheva.net ~]# cd /mnt/nfs/home/dmbelicheva
[root@client.dmbelicheva.net dmbelicheva]# touch test.txt
touch: cannot touch 'test.txt': Read-only file system
[root@client.dmbelicheva.net dmbelicheva]# ls
[root@client.dmbelicheva.net dmbelicheva]# exit
logout
[dmbelicheva@client.dmbelicheva.net dmbelicheva]$ ls
[dmbelicheva@client.dmbelicheva.net dmbelicheva]$
```

Рис. 3.24: Создание файла

Безуспешно.

Попробуем это проделать под пользователем root.

Безуспешно.

На сервере посмотрим, появились ли изменения в каталоге пользователя /home/user/common.

Не появились, все тщетно.

3.5 Внесение изменений в настройки внутреннего окружения виртуальных машин

На виртуальной машине server перейдем 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:

```
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 создадим исполняемый файл nfs.sh:

```
cd /vagrant/provision/server
touch nfs.sh
chmod +x nfs.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт:


```
GNU nano 5.6.1                                nfs.sh
#!/bin/bash

echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/nfs/etc/* /etc
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service nfs --permanent
firewall-cmd --add-service mountd --add-service rpc-bind --permanent
firewall-cmd --reload

echo "Tuning SELinux"
mkdir -p /srv/nfs
semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
restorecon -vR /srv/nfs

echo "Mounting dirs"
mkdir -p /srv/nfs/www
mount -o bind /var/www /srv/nfs/www
echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
mkdir -p /srv/nfs/home/user
mkdir -p -m 700 /home/user/common
chown user:user /home/user/common
mount -o bind /home/user/common /srv/nfs/home/user
echo "/home/user/common /srv/nfs/home/user none bind 0 0" >> /etc/fstab

echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server

systemctl restart firewalld
```

Рис. 3.25: Редактирование файла

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
```

Открыв его на редактирование, пропишем в нём следующий скрипт:



```
GNU nano 5.6.1                                nfs.sh
#!/bin/bash

echo "Provisioning script $0"

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

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

Рис. 3.26: Редактирование файла

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:

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

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

4 Выводы

В процессе выполнения данной лабораторной работы я приобрела навыки настройки сервера NFS для удалённого доступа к ресурсам.