

Отчет по лабораторной работе № 13. Настройка NFS

Данила Стариков
НПИбд-02-22

2024

Содержание

1	Цель работы	3
2	Выполнение работы	4
2.1	Настройки сервера NFSv4	4
2.2	Монтирование NFS на клиенте	7
2.3	Подключение каталогов к дереву NFS	9
2.4	Подключение каталогов для работы пользователей	12
2.5	Внесение изменений в настройки внутреннего окружения виртуальных машин	14
3	Выводы	16

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Выполнение работы

2.1 Настройки сервера NFSv4

1. На сервере установили необходимое программное обеспечение:

```
dnf -y install nfs-utils
```

2. На сервере создали каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS):

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

3. В файле `/etc/exports` прописали подключаемый через NFS общий каталог с доступом только на чтение:

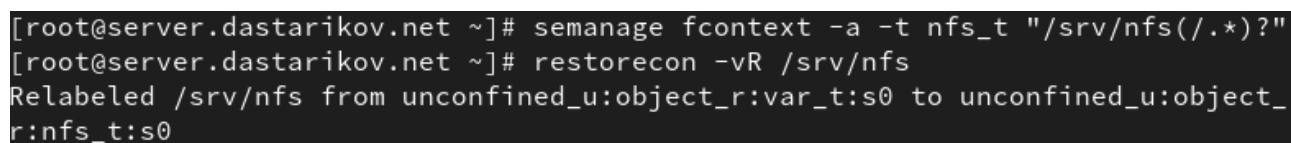
```
/srv/nfs *(ro)
```

4. Для общего каталога задали контекст безопасности NFS (Рис. 1):

```
semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
```

5. Применили изменённую настройку SELinux к файловой системе (Рис. 1):

```
restorecon -vR /srv/nfs
```

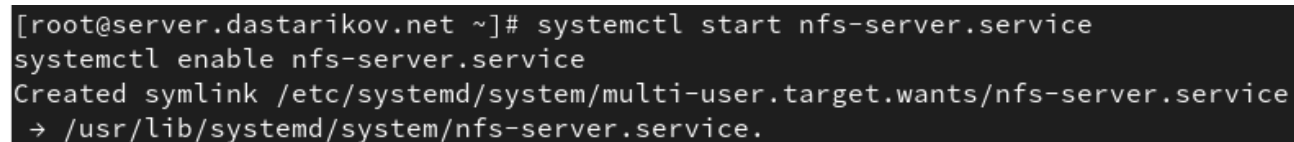


```
[root@server.dastarikov.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.dastarikov.net ~]# restorecon -vR /srv/nfs
Relabeled /srv/nfs from unconfined_u:object_r:var_t:s0 to unconfined_u:object_r:nfs_t:s0
```

Рис. 1: Создание контекста безопасности для каталога `/srv/nfs`.

6. Запустили сервер NFS (Рис. 2):

```
systemctl start nfs-server.service
systemctl enable nfs-server.service
```



```
[root@server.dastarikov.net ~]# systemctl start nfs-server.service
systemctl enable nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service
→ /usr/lib/systemd/system/nfs-server.service.
```

Рис. 2: Запуск сервера NFS.

7. Настроили межсетевой экран для работы сервера NFS (Рис. 3):

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

```
[root@server.dastarikov.net ~]# firewall-cmd --add-service=nfs
firewall-cmd --add-service=nfs --permanent
firewall-cmd --reload
success
success
success
```

Рис. 3: Настройка межсетевого экрана.

8. На клиенте установили необходимое для работы NFS программное обеспечение:

```
dnf -y install nfs-utils
```

9. На клиенте попробовали посмотреть имеющиеся подмонтированные удалённые ресурсы (вместо dastarikov укажите свой логин) (Рис. 4):

```
showmount -e server.dastarikov.net
```

В отчёте поясните, что при этом происходит.

10. Попробовали на сервере остановить сервис межсетевого экрана:

```
systemctl stop firewalld.service
```

Затем на клиенте вновь попробовали подключиться к удалённо смонтированному ресурсу (Рис. 4):

```
showmount -e server.dastarikov.net
```

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

Рис. 4: Попытки подключения к удаленно смонтированному ресурсу.

11. На сервере запустили сервис межсетевого экрана

```
systemctl start firewalld
```

12. На сервере посмотрели, какие службы задействованы при удалённом монтировании (Рис. 5 и 6):

```
lsof | grep TCP
```

```
lsof | grep UDP
```



```
[root@server.dastarikov.net ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula
bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bittor
itcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-col
lector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-regis
try docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 fr
eeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd gra
fana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isn
s jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-pla
ne kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-
scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-t
ls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mo
sh mountd mqttt mqttt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrp
e ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 p
op3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsh ptp pulseaudio puppetma
ster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client sam
ba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-
sync squid sssd ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog
syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsms vnc-server warpinator
wbem-http wbem-https wireguard ws-discovery ws-discovery-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.dastarikov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.dastarikov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.dastarikov.net ~]# firewall-cmd --reload
success
[root@server.dastarikov.net ~]#
```

Рис. 7: Добавление служб rpc-bind и mountd в настройки межсетевого экрана.

14. На клиенте проверили подключение удалённого ресурса (Рис. 8):

```
showmount -e server.dastarikov.net
```

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

Рис. 8: Проверка подключения удаленного ресурса.

2.2 Монтирование NFS на клиенте

1. На клиенте создали каталог, в который будет монтироваться удалённый ресурс, и подмонтировали дерево NFS (Рис. 9):

```
mkdir -p /mnt/nfs
mount server.dastarikov.net:/srv/nfs /mnt/nfs
```

```
[root@client.dastarikov.net ~]# mkdir -p /mnt/nfs
[root@client.dastarikov.net ~]# mount server.dastarikov.net:/srv/nfs /mnt/nfs
```

Рис. 9: Создание каталога на клиенте, в который будет монтироваться удаленный ресурс.

2. Проверили, что общий ресурс NFS подключён правильно (Рис. 10):

```
mount
```

```
server.dastarikov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsi
ze=131072,wsiz=131072,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,c
lientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
```

Рис. 10: Проверка правильности подключения ресурса NFS.

3. На клиенте в конце файла `/etc/fstab` добавили следующую запись (Рис. 11):

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

```
#
# /etc/fstab
# Created by anaconda on Wed Oct  2 05:47:10 2024
#
# 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=4abc3784-4e81-4c61-a127-ab85a2e1cdf8 /                xfs      defa
ults          0 0
/swapfile none swap defaults 0 0
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
server.dastarikov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рис. 11: Изменение файла `/etc/fstab`.

4. На клиенте проверили наличие автоматического монтирования удалённых ресурсов при запуске операционной системы (Рис. 12):

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

```
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled; prese>
   Active: active since Sat 2024-11-30 18:08:01 UTC; 15min ago
     Until: Sat 2024-11-30 18:08:01 UTC; 15min ago
    Docs: man:systemd.special(7)

Nov 30 18:08:01 client.dastarikov.net systemd[1]: Reached target Remote File >
```

Рис. 12: Проверка автоматического монтирования удаленных ресурсов на клиенте.

5. Перезапустили клиента и убедились, что удалённый ресурс подключается автомати-чески (Рис. 13).


```
server.dastarikov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsi
ze=131072,wsizе=131072,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,c
lientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1,_netdev)
```

Рис. 13: Проверка автоматического монтирования после перезагрузки клиента.

2.3 Подключение каталогов к дереву NFS

1. На сервере создали общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера (Рис. 14):

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

2. Подмонтировали каталог web-сервера:

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

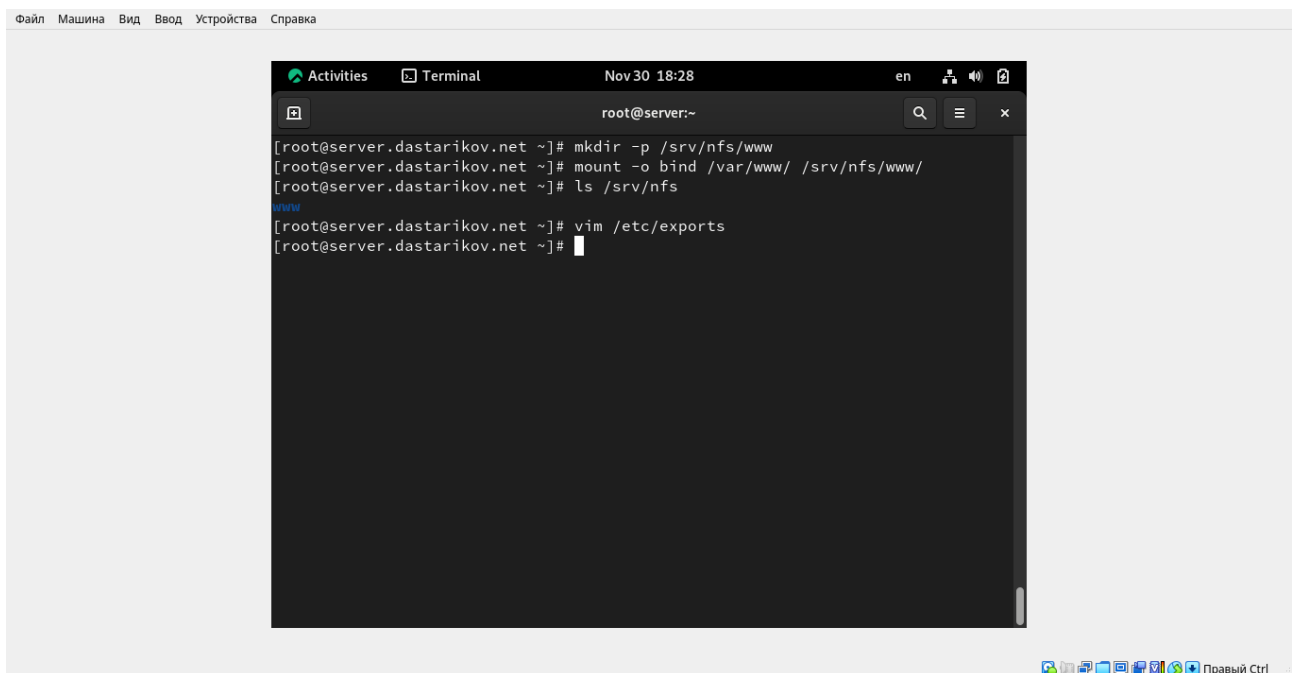


Рис. 14: Создание общего каталога с контентом веб-сервера.

3. На сервере проверили, что отображается в каталоге /srv/nfs (Рис. 15).

```
[root@server.dastarikov.net ~]# ls /srv/nfs  
www
```

Рис. 15: Проверка содержимого /srv/nfs.

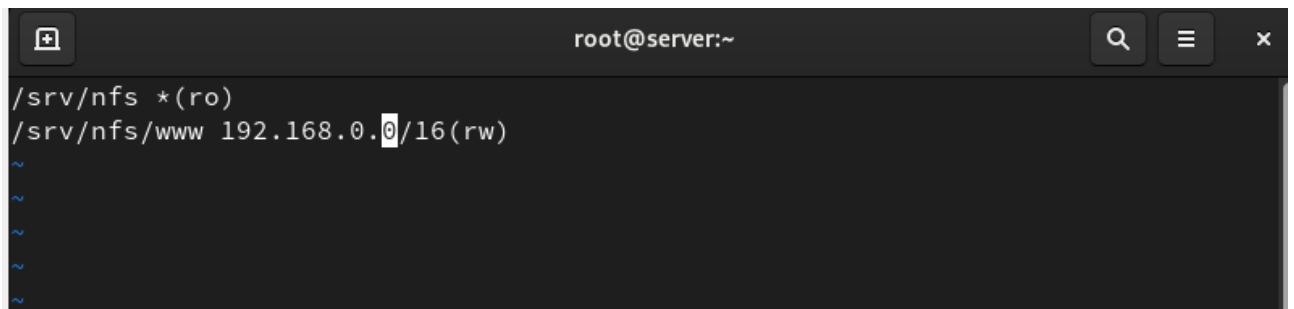
4. На клиенте посмотрели, что отображается в каталоге /mnt/nfs (Рис. 16).

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

Рис. 16: Проверка содержимого /mnt/nfs.

5. На сервере в файле /etc/exports добавили экспорт каталога веб-сервера с удалённого ресурса (Рис. 17):

```
/srv/nfs/www 192.168.0.0/16(rw)
```



The screenshot shows a terminal window titled 'root@server:~'. The terminal displays the content of the /etc/exports file, which contains two lines: '/srv/nfs *(ro)' and '/srv/nfs/www 192.168.0.0/16(rw)'. Below these lines, there are several tilde (~) characters, likely representing the end of the file or a prompt.

Рис. 17: Изменение файла /etc/exports.

6. Экспортировали все каталоги, упомянутые в файле /etc/exports:

```
exportfs -r
```

7. Проверили на клиенте каталог /mnt/nfs.

8. На сервере в конце файла /etc/fstab добавили следующую запись (Рис. 18):

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

```
#
# /etc/fstab
# Created by anaconda on Wed Oct  2 05:47:10 2024
#
# 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=4abc3784-4e81-4c61-a127-ab85a2e1cdf8 /          xfs      defa
ults          0 0
/swapfile none swap defaults 0 0
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
/var/www /srv/nfs/www none bind 0 0
```

Рис. 18: Изменение файла /etc/fstab.

9. Повторно экспортировали каталоги, указанные в файле /etc/exports:
exportfs -r
10. На клиенте проверили каталог /mnt/nfs (Рис. 19).

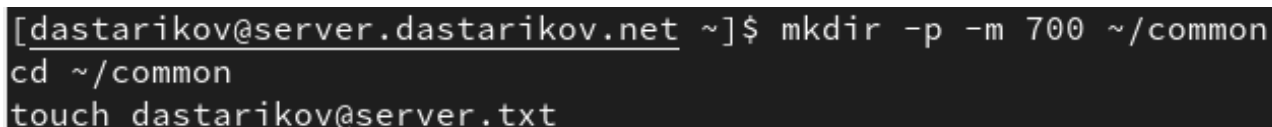
```
[root@client.dastarikov.net www]# tree .
.
├── cgi-bin
├── html
│   ├── server.dastarikov.net
│   │   └── index.html
│   └── www.dastarikov.net
│       └── index.php
```

Рис. 19: Содержимое каталога /mnt/nfs/www.

2.4 Подключение каталогов для работы пользователей

1. На сервере под пользователем dastarikov в его домашнем каталоге создали каталог common с полными правами доступа только для этого пользователя, а в нём файл dastarikov@server.txt (Рис. 20):

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



```
[dastarikov@server.dastarikov.net ~]$ mkdir -p -m 700 ~/common
cd ~/common
touch dastarikov@server.txt
```

Рис. 20: Создание личного каталога common для пользователя dastarikov.

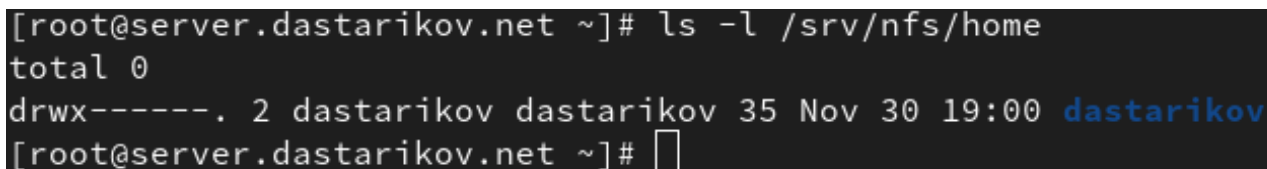
2. На сервере создали общий каталог для работы пользователя dastarikov по сети:

```
mkdir -p /srv/nfs/home/dastarikov
```

3. Подмонтировали каталог common пользователя dastarikov в NFS:

```
mount -o bind /home/dastarikov/common /srv/nfs/home/dastarikov
```

Этот каталог имеет те же права доступа, что и локальный каталог common (Рис. 21):

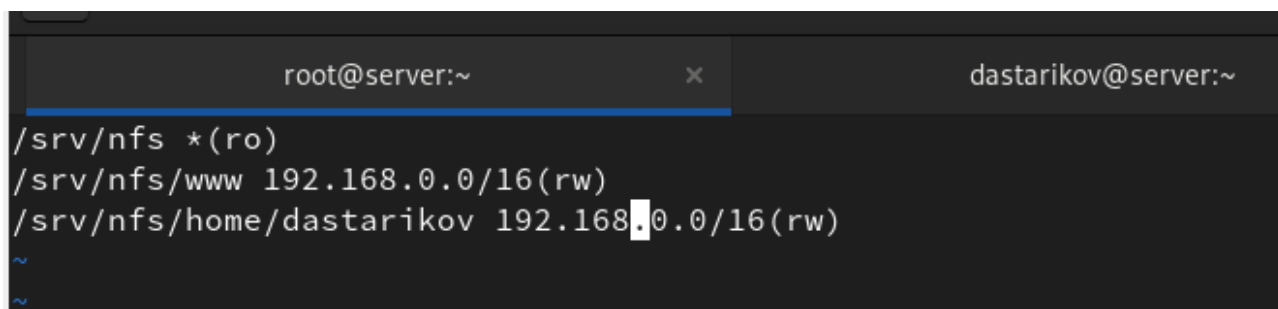


```
[root@server.dastarikov.net ~]# ls -l /srv/nfs/home
total 0
drwx-----. 2 dastarikov dastarikov 35 Nov 30 19:00 dastarikov
[root@server.dastarikov.net ~]#
```

Рис. 21: Права каталога /srv/nfs/home/dastarikov.

4. Подключили каталог пользователя в файле /etc/exports, прописав в нём (Рис. 22):

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

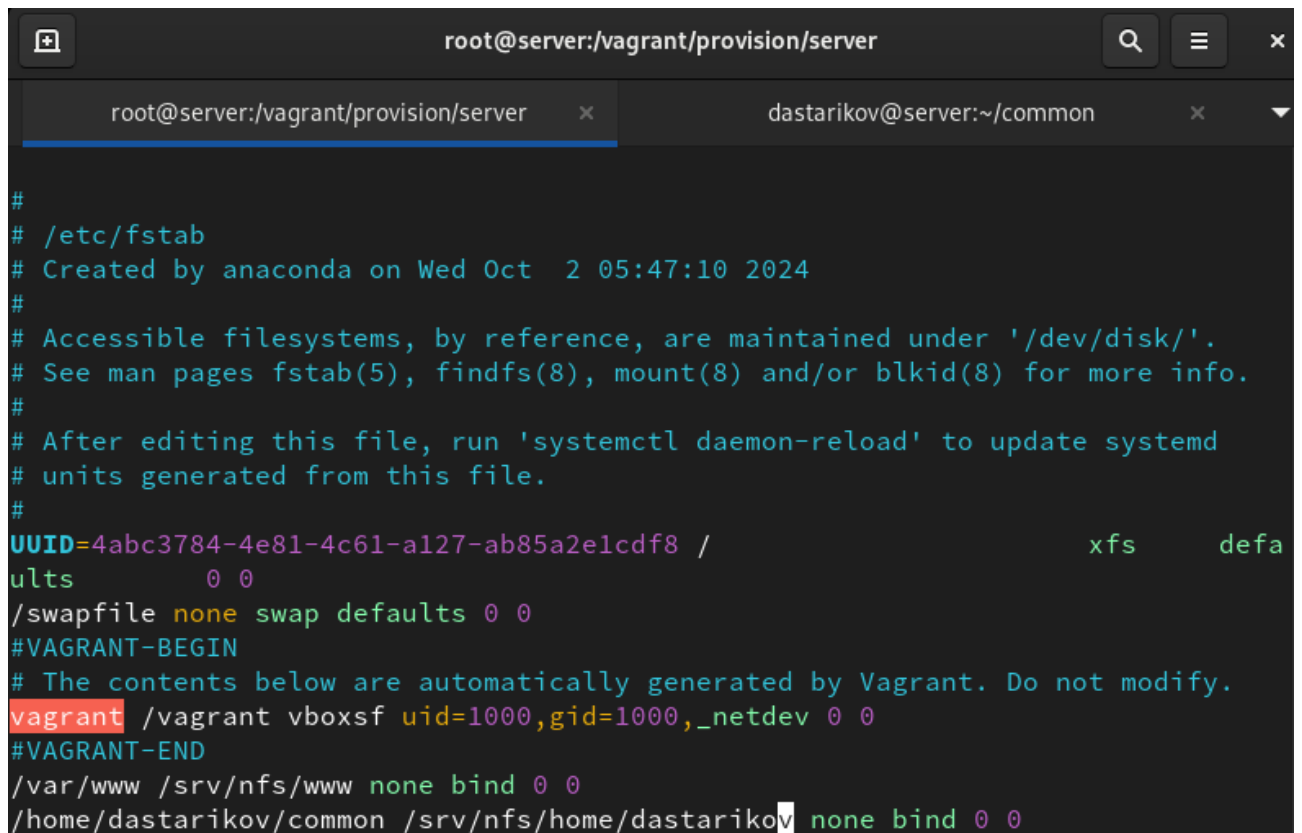


```
root@server:~ x dastarikov@server:~
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/dastarikov 192.168.0.0/16(rw)
~
~
```

Рис. 22: Изменение файла /etc/exports.

5. Внесли изменения в файл `/etc/fstab` (Рис. 23):

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



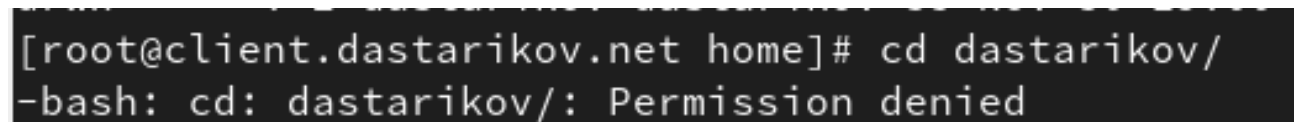
```
#  
# /etc/fstab  
# Created by anaconda on Wed Oct 2 05:47:10 2024  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd  
# units generated from this file.  
#  
UUID=4abc3784-4e81-4c61-a127-ab85a2e1cdf8 / xfs defa  
ults 0 0  
/swapfile none swap defaults 0 0  
#VAGRANT-BEGIN  
# The contents below are automatically generated by Vagrant. Do not modify.  
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0  
#VAGRANT-END  
/var/www /srv/nfs/www none bind 0 0  
/home/dastarikov/common /srv/nfs/home/dastarikov none bind 0 0
```

Рис. 23: Изменение `/etc/fstab`.

6. Повторно экспортировали каталоги:

```
exportfs -r
```

7. На клиенте проверили каталог `/mnt/nfs` (Рис. 24).



```
[root@client.dastarikov.net home]# cd dastarikov/  
-bash: cd: dastarikov/: Permission denied
```

Рис. 24: Попытка зайти в каталог `dastarikov` на клиенте пользователем `root`.

8. На клиенте под пользователем `dastarikov` перешли в каталог `/mnt/nfs/home/dastarikov` и попробовали создать в нём файл `dastarikov@client.txt` и внести в него какие-либо изменения (Рис. 25):

```
cd /mnt/nfs/home/dastarikov  
touch dastarikov@client.txt
```

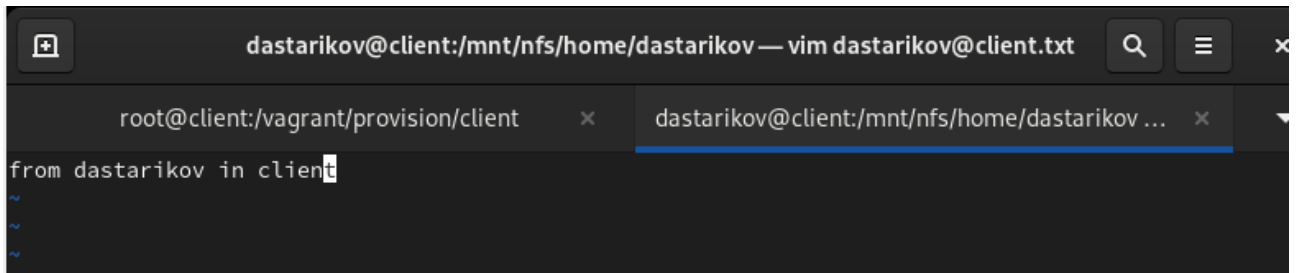


Рис. 25: Создание файла на клиенте.

Попробовали проделать это под пользователем root (Рис. 24).

9. На сервере посмотрели, появились ли изменения в каталоге пользователя `/home/dastarikov/` (Рис. 26).

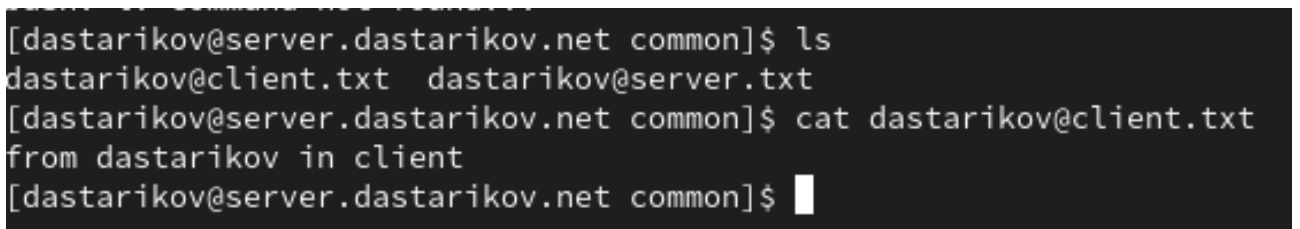


Рис. 26: Проверка изменений в локальном каталоге пользователя dastarikov.

2.5 Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине server перешли в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создали в нём каталог `nfs`, в который поместите в соответствующие подкаталоги конфигурационные файлы (Рис. 27):

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/nfs/etc
cp -R /etc/exports /vagrant/provision/server/nfs/etc/
```

2. В каталоге `/vagrant/provision/server` создали исполняемый файл `nfs.sh` (Рис. 27):

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

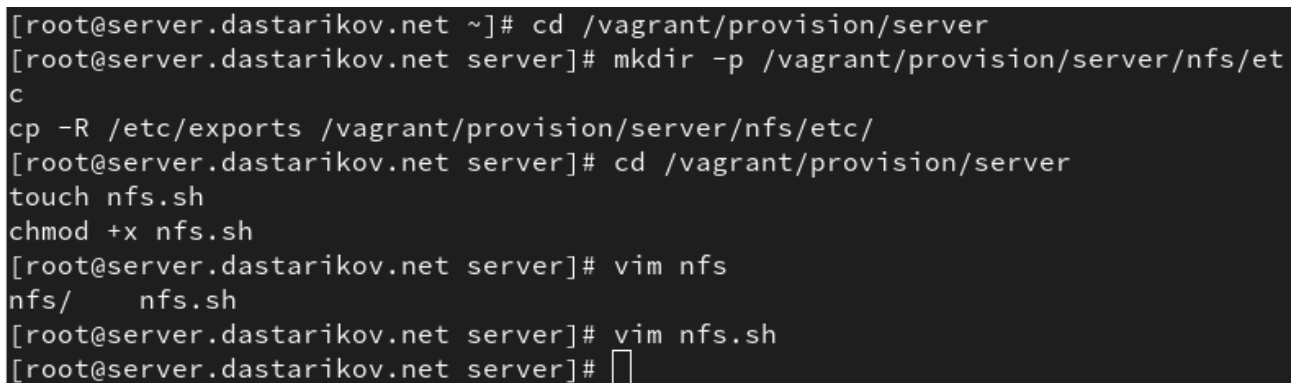
Открыв его на редактирование, прописали в нём следующий скрипт (вместо `dastarikov` укажите свой логин) (Рис. 27):

```
#!/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/dastarikov
mkdir -p -m 700 /home/dastarikov/common
chown dastarikov:dastarikov /home/dastarikov/common
mount -o bind /home/dastarikov/common /srv/nfs/home/dastarikov
echo "/home/dastarikov/common /srv/nfs/home/dastarikov none bind 0 0" >>
↵ /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld

```



```

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

```

Рис. 27: Изменение настроек внутреннего окружения на виртуальной машине server.

3. На виртуальной машине client перейдите в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/ (Рис. 28):

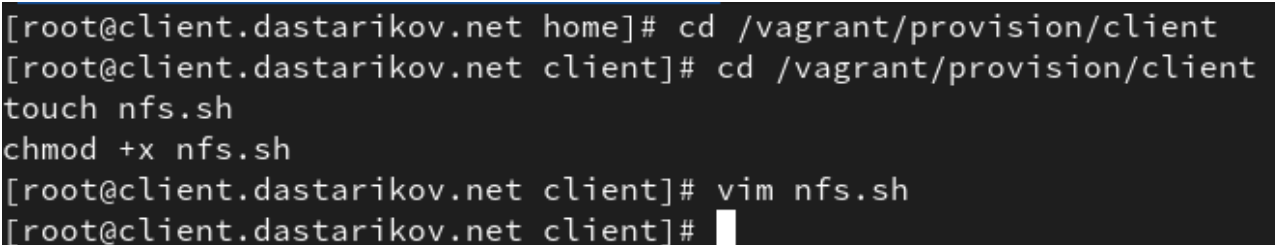
```
cd /vagrant/provision/client
```

4. В каталоге /vagrant/provision/client создайте исполняемый файл nfs.sh (Рис. 28):

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

Открыв его на редактирование, пропишите в нём следующий скрипт (Рис. 28):

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



```
[root@client.dastarikov.net home]# cd /vagrant/provision/client
[root@client.dastarikov.net client]# cd /vagrant/provision/client
touch nfs.sh
chmod +x nfs.sh
[root@client.dastarikov.net client]# vim nfs.sh
[root@client.dastarikov.net client]#
```

Рис. 28: Изменение настроек внутреннего окружения на виртуальной машине client.

5. 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"
```

3 Выводы

В результате лабораторной работы познакомились с настройкой сервера NFS на примере создания каталога веб-сервера и каталога для удаленной работы конкретного пользователя.