

Отчет по лабораторной работе №13

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Настройка сервера NFSv4	7
3.2	Монтирование NFS на клиенте	11
3.3	Подключение каталогов к дереву NFS	13
3.4	Подключение каталогов для работы пользователей	15
3.5	Внесение изменений в настройки внутреннего окружения виртуальных машин	17
4	Ответы на контрольные вопросы	21
5	Выводы	22

Список иллюстраций

3.1	Установка необходимого ПО	7
3.2	Создание каталога	7
3.3	Редактирование exports	8
3.4	Задание контекста SELinux	8
3.5	Запуск сервера NFS	8
3.6	Настройка firewall	9
3.7	Установка ПО на клиенте	9
3.8	Просмотр подмонтированных ресурсов	9
3.9	Остановка firewall	9
3.10	Повторная попытка подключения	10
3.11	Запуск firewall	10
3.12	Службы с TCP	10
3.13	Службы с UDP	10
3.14	Настройка firewall	11
3.15	Проверка подключения удалённого ресурса	11
3.16	Создание каталога и подмонтирование NFS	11
3.17	Проверка подключения NFS	12
3.18	Редактирование fstab	12
3.19	Проверка службы автоматического монтирования	12
3.20	Перезапуск клиента	13
3.21	Проверка мониторинга	13
3.22	Создание и подмонтирование каталога	13
3.23	Проверка каталога /srv/nfs на сервере	13
3.24	Проверка каталога /mnt/nfs на клиенте	14
3.25	Редактирование exports	14
3.26	Экспортирование каталогов в exports	14
3.27	Проверка /mnt/www на клиенте	14
3.28	Редактирование fstab	15
3.29	Проверка /mnt/www на клиенте	15
3.30	Создание каталога и файла	15
3.31	Создание каталога для работы пользователя по сети	15
3.32	Монтирование каталога common	16
3.33	Поделочение каталога пользователя	16
3.34	Редактирование fstab	16
3.35	Проверка каталога /mnt/nfs	16
3.36	Создание файла и внесение изменений под пользователем	17
3.37	Создание файла и внесение изменений под root	17

3.38	Просмотр /home/user/common на сервере	17
3.39	Создание каталогов и копирование конф.файлов	18
3.40	Скрипт nfs.sh	18
3.41	Создание скрипта	19
3.42	Скрипт nfs.sh	19
3.43	Редактирование Vagrantfile	19
3.44	Редактирование Vagrantfile	20

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Задание

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

3 Выполнение лабораторной работы

3.1 Настройка сервера NFSv4

На сервере установим необходимое программное обеспечение: `dnf -y install nfs-utils`. (рис. 1)

```
[root@server.svivanov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64           1.0 kB/s | 19 kB    00:18
Extra Packages for Enterprise Linux 10 - x86_64           3.5 MB/s | 5.5 MB    00:01
Rocky Linux 10 - BaseOS                                   9.7 kB/s | 4.3 kB    00:00
Rocky Linux 10 - AppStream                                13 kB/s | 4.3 kB    00:00
Rocky Linux 10 - Extras                                   8.7 kB/s | 3.1 kB    00:00
Dependencies resolved.
=====
Package                Architecture      Version           Repository        Size
=====
Installing:
nfs-utils              x86_64            1:2.8.2-3.el10   baseos            473 k
Installing dependencies:
gssproxy               x86_64            0.9.2-10.el10   baseos            111 k
libev                  x86_64            4.33-14.el10    baseos             52 k
=====
```

Рис. 3.1: Установка необходимого ПО

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS): `mkdir -p /srv/nfs` (рис. 2)

```
[root@server.svivanov.net ~]# mkdir -p /srv/nfs
[root@server.svivanov.net ~]#
```

Рис. 3.2: Создание каталога

В файле `/etc/exports` пропишем подключаемый через NFS общий каталог с доступом только на чтение: `/srv/nfs *(ro)` (рис. 3)

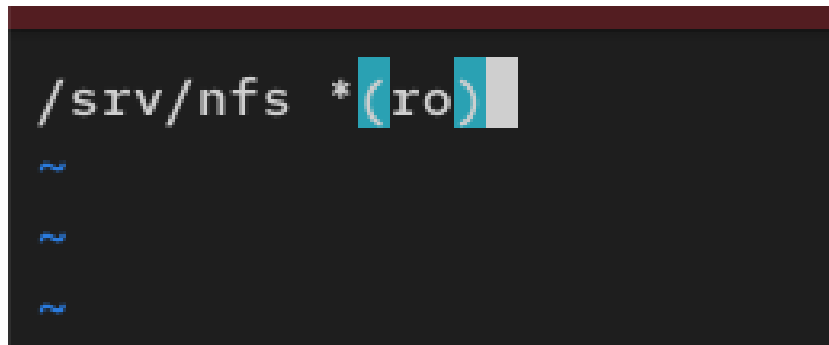


Рис. 3.3: Редактирование exports

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 (рис. 4)
```

```
[root@server.svivanov.net etc]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.svivanov.net etc]# 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.svivanov.net etc]#
```

Рис. 3.4: Задание контекста SELinux

Запустим сервер NFS:

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

```
systemctl enable nfs-server.service. (рис. 5)
```

```
[root@server.svivanov.net etc]# systemctl start nfs-server.service
[root@server.svivanov.net etc]# 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.svivanov.net etc]#
```

Рис. 3.5: Запуск сервера NFS

Настроим межсетевой экран для работы сервера NFS:

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

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

```
firewall-cmd --reload. (рис. 6)
```

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

Рис. 3.6: Настройка firewall

На клиенте установим необходимое для работы NFS программное обеспечение: `dnf -y install nfs-utils` (рис. 7)

```
[svivanov@client.svivanov.net ~]$ sudo -i
[sudo] пароль для svivanov:
[root@client.svivanov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86_64                3.5 kB/s | 19 kB      00:05
Extra Packages for Enterprise Linux 10 - x86_64                546 kB/s | 5.5 MB    00:10
Rocky Linux 10 - BaseOS                                         828 B/s | 4.3 kB     00:05
Rocky Linux 10 - AppStream                                       564 B/s | 4.3 kB     00:07
Rocky Linux 10 - Extras                                         599 B/s | 3.1 kB     00:05
Dependencies resolved.
=====
Package                Architecture    Version           Repository         Size
=====
Installing:
nfs-utils               x86_64         1:2.8.2-3.el10    baseos             473 k
Installing dependencies:
gssproxy                x86_64         0.9.2-10.el10    baseos             111 k
libev                   x86_64         4.33-14.el10     baseos              52 k
libtirpc                x86_64         1:2.8.2-3.el10    baseos              61 k
=====
```

Рис. 3.7: Установка ПО на клиенте

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы: `showmount -e server.user.net`.

Ошибка `Unable to receive` говорит о том, что клиент не может связаться с NFS сервером. (рис. 8)

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

Рис. 3.8: Просмотр подмонтированных ресурсов

Попробуем на сервере остановить сервис межсетевого экрана: `systemctl stop firewalld.service` (рис. 9)

```
[root@server.svivanov.net etc]# systemctl stop firewalld.service
[root@server.svivanov.net etc]#
```

Рис. 3.9: Остановка firewall

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

В этот раз мы смогли подключиться. Похоже проблема была в том, что firewall мешал нам это сделать, но отключив его, все получилось. (рис. 10)

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

Рис. 3.10: Повторная попытка подключения

На сервере запустите сервис межсетевого экрана: `systemctl start firewalld` (рис. 11)

```
[root@server.svivanov.net etc]# systemctl start firewalld
```

Рис. 3.11: Запуск firewall

На сервере посмотрим, какие службы задействованы при удалённом монтировании:

`lsof | grep TCP`

`lsof | grep UDP` (рис. 12, 13)

```
[root@server.svivanov.net etc]# 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 504u IPv4 45878 0t0 TCP *:sunrpc (LISTEN)
systemd 1 root 506u IPv6 46857 0t0 TCP *:sunrpc (LISTEN)
systemd 1 root 532u IPv6 6866 0t0 TCP *:websockets (LISTEN)
cupsd 1215 root 7u IPv6 10530 0t0 TCP localhost:ipp (LISTEN)
cupsd 1215 root 8u IPv4 10531 0t0 TCP localhost:ipp (LISTEN)
sshd 1230 root 7u IPv4 11733 0t0 TCP *:down (LISTEN)
sshd 1230 root 8u IPv6 11735 0t0 TCP *:down (LISTEN)
sshd 1230 root 9u IPv4 11737 0t0 TCP *:ssh (LISTEN)
sshd 1230 root 10u IPv6 11739 0t0 TCP *:ssh (LISTEN)
named 1293 named 45u IPv4 11813 0t0 TCP localhost:domain (LISTEN)
named 1293 named 46u IPv4 11814 0t0 TCP localhost:domain (LISTEN)
```

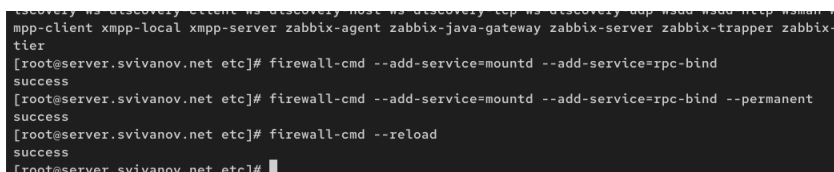
Рис. 3.12: Службы с TCP

```
[root@server.svivanov.net etc]# 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 505u IPv4 45885 0t0 UDP *:sunrpc
systemd 1 root 508u IPv6 47211 0t0 UDP *:sunrpc
avahi-daemon 864 avahi 12u IPv4 8706 0t0 UDP *:mdns
avahi-daemon 864 avahi 13u IPv6 8707 0t0 UDP *:mdns
chronyd 927 chrony 5u IPv4 9544 0t0 UDP localhost:323
chronyd 927 chrony 6u IPv6 9545 0t0 UDP localhost:323
chronyd 927 chrony 7u IPv4 9546 0t0 UDP *:ntp
named 1293 named 6u IPv4 47860 0t0 UDP server.svivanov.net:domain
named 1293 named 41u IPv4 11809 0t0 UDP localhost:domain
named 1293 named 42u IPv4 11810 0t0 UDP localhost:domain
```

Рис. 3.13: Службы с UDP

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

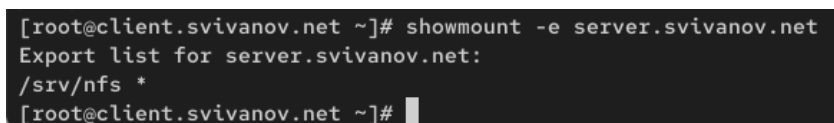
```
firewall-cmd --get-services
firewall-cmd --add-service=mountd --add-service=rpc-bind
firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
firewall-cmd --reload. (рис. 14)
```



```
mp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zabbix-server zabbix-trapper zabbix-
tier
[root@server.svivanov.net etc]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.svivanov.net etc]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.svivanov.net etc]# firewall-cmd --reload
success
[root@server.svivanov.net etc]#
```

Рис. 3.14: Настройка firewall

На клиенте проверим подключение удалённого ресурса: `showmount -e server.user.net`. (рис. 15)



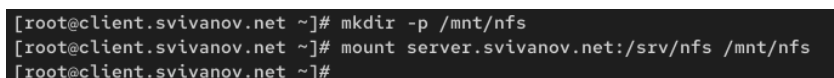
```
[root@client.svivanov.net ~]# showmount -e server.svivanov.net
Export list for server.svivanov.net:
/srv/nfs *
[root@client.svivanov.net ~]#
```

Рис. 3.15: Проверка подключения удалённого ресурса

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.user.net:/srv/nfs /mnt/nfs. (рис. 16)
```



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

Рис. 3.16: Создание каталога и подмонтирование NFS

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

Тут мы видим что удаленный ресурс NFS успешно подключен. /srv/nfs - это удаленный ресурс, /mnt/nfs - точка монтирования на клиенте. (рис. 17)

```
server.svivanov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=524288,wsize=524288,nam
len=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.16
8.1.1)
[root@client.svivanov.net ~]#
```

Рис. 3.17: Проверка подключения NFS

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

server.svivanov.net:/srv/nfs - ресурс монтирования.

/mnt/nfs - точка монтирования на клиенте.

nfs - тип файловой системы.

_netdev - указывает на то, что ресурс должен монтироваться только после загрузки сети. 0 - указывает, нужно ли делать резервную копию файловой системы (0 = не нужно), 0 - порядок проверки файловой системы (0 = не проверять). (рис. 18)

```
#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.svivanov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
-- РЕЖИМ ВСТАВКИ --
```

Рис. 3.18: Редактирование fstab

На клиенте проверим наличие автоматического монтирования удалённых ресурсов при запуске операционной системы: systemctl status remote-fs.target. Сервис запущен и работает. (рис. 19)

```
[root@client.svivanov.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 Wed 2025-11-12 10:19:46 UTC; 16min ago
   Invocation: 5882b089355c47e0a4df4eb6b7b92ae4
   Docs: man:systemd.special(7)

ноя 12 10:19:46 client.svivanov.net systemd[1]: Reached target remote-fs.target - Remote File S
[root@client.svivanov.net etc]#
```

Рис. 3.19: Проверка службы автмонтирования

Перезапустим клиента и убедимся, что удалённый ресурс подключается автоматически. Убеждаемся, что все работает корректно. (рис. 20, 21)

```
C:\work_asp\svivanov\vagrant>vagrant halt client
==> client: Attempting graceful shutdown of VM...

C:\work_asp\svivanov\vagrant>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Clearing any previously set forwarded ports...
```

Рис. 3.20: Перезапуск клиента

```
[svivanov@client.svivanov.net ~]$ sudo -i
[sudo] пароль для svivanov:
[root@client.svivanov.net ~]# showmount -e server.svivanov.net
Export list for server.svivanov.net:
/srv/nfs *
[root@client.svivanov.net ~]#
```

Рис. 3.21: Проверка мониторинга

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/` (рис. 22)

```
[root@server.svivanov.net etc]# mkdir -p /srv/nfs/www
[root@server.svivanov.net etc]# mount -o bind /var/www/ /srv/nfs/www/
[root@server.svivanov.net etc]#
```

Рис. 3.22: Создание и подмонтирования каталога

На сервере проверим, что отображается в каталоге `/srv/nfs`. Отображается каталог `www`. (рис. 23)

```
[root@server.svivanov.net etc]# ls /srv/nfs
www
[root@server.svivanov.net etc]#
```

Рис. 3.23: Проверка каталога `/srv/nfs` на сервере

На клиенте посмотрим, что отображается в каталоге `/mnt/nfs`. Отображается каталог `www`. (рис. 24)

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

Рис. 3.24: Проверка каталога /mnt/nfs на клиенте

На сервере в файле /etc/exports добавим экспорт каталога веб-сервера с удалённого ресурса: /srv/nfs/www 192.168.0.0/16(rw) (рис. 25)

```
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
~
```

Рис. 3.25: Редактирование exports

Экспортируем все каталоги, упомянутые в файле /etc/exports: exportfs -r (рис. 26)

```
[root@server.svivanov.net etc]# exportfs -r
[root@server.svivanov.net etc]#
```

Рис. 3.26: Экспортирование каталогов в exports

Проверим на клиенте каталог /mnt/nfs. Там у нас отображается каталог www. (рис. 27)

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

Рис. 3.27: Проверка /mnt/www на клиенте

На сервере в конце файла /etc/fstab добавим следующую запись: /var/www /srv/nfs/www none bind 0 0 (рис. 28)

```
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
var/www /srv/nfs/www none bind 0 0
#VAGRANT-END
~
```

Рис. 3.28: Редактирование fstab

Повторно экспортируем каталоги, указанные в файле /etc/exports и на клиенте проверим каталог /mnt/nfs. (рис. 29)

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

Рис. 3.29: Проверка /mnt/www на клиенте

3.4 Подключение каталогов для работы пользователей

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

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

```
cd ~/common
```

```
touch user@server.txt (рис. 30)
```

```
[root@server.svivanov.net etc]# logout
[svivanov@server.svivanov.net ~]$ mkdir -p -m 700 ~/common
[svivanov@server.svivanov.net ~]$ cd ~/common
[svivanov@server.svivanov.net common]$ touch svivanov@server.txt
[svivanov@server.svivanov.net common]$
```

Рис. 3.30: Создание каталога и файла

На сервере создадим общий каталог для работы пользователя по сети: mkdir -p /srv/nfs/home/user (рис. 31)

```
[root@server.svivanov.net ~]# mkdir -p /srv/nfs/home/svivanov
```

Рис. 3.31: Создание каталога для работы пользователя по сети

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

Мы устанавливали права 700 - это все права (чтение, запись, выполнение) для пользователя (svivanov). Остальные пользователи, в том числе root не имеют прав. (рис. 32)

```
[root@server.svivanov.net ~]# mount -o bind /home/svivanov/common /srv/nfs/home/svivanov
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@server.svivanov.net ~]# systemctl daemon-reload
[root@server.svivanov.net ~]# mount -o bind /home/svivanov/common /srv/nfs/home/svivanov
[root@server.svivanov.net ~]#
```

Рис. 3.32: Монтирование каталога common

Подключим каталог пользователя в файле `/etc/exports`, прописав в нём: `/srv/nfs/home/user 192.168.0.0/16(rw)` (рис. 33)

```
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/svivanov 192.168.0.0/16(rw)
~
```

Рис. 3.33: Подключение каталога пользователя

Внесем изменения в файл `/etc/fstab`: `/home/user/common /srv/nfs/home/user none bind 0 0` (рис. 34)

```
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
var/www /srv/nfs/www none bind 0 0
/home/svivanov/common /srv/nfs/home/svivanov none bind 0 0
#VAGRANT-END
~
```

Рис. 3.34: Редактирование fstab

Повторно экспортируем каталоги. На клиенте проверим каталог `/mnt/nfs`. Теперь тут появился каталог home. (рис. 35)

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

Рис. 3.35: Проверка каталога /mnt/nfs

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 (рис. 36)
```

```
[svivanov@client.svivanov.net ~]$ cd /mnt/nfs/home/svivanov
[svivanov@client.svivanov.net svivanov]$ touch svivanov@client.txt
[svivanov@client.svivanov.net svivanov]$ vim svivanov@client.txt
[svivanov@client.svivanov.net svivanov]$ cat svivanov@client.txt
ksfjdfnsdjfnsdfjfdskj
[svivanov@client.svivanov.net svivanov]$
```

Рис. 3.36: Создание файла и внесение изменений под пользователем

Попробуем проделать это под пользователем root. Ничего не получилось, т.к у root нет прав на это. (рис. 37)

```
[root@client.svivanov.net ~]# cd /mnt/nfs/home/svivanov
-bash: cd: /mnt/nfs/home/svivanov: Отказано в доступе
[root@client.svivanov.net ~]#
```

Рис. 3.37: Создание файла и внесение изменений под root

На сервере посмотрим, появились ли изменения в каталоге пользователя /home/user/common. Да, теперь тут файлы и с клиента, и с сервера. (рис. 38)

```
[svivanov@server.svivanov.net common]$ ls /home/svivanov/common
svivanov@client.txt  svivanov@server.txt
[svivanov@server.svivanov.net common]$
```

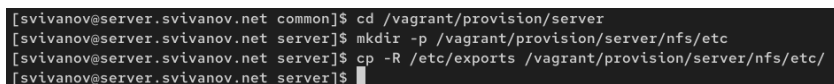
Рис. 3.38: Просмотр /home/user/common на сервере

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/ (рис. 39)
```



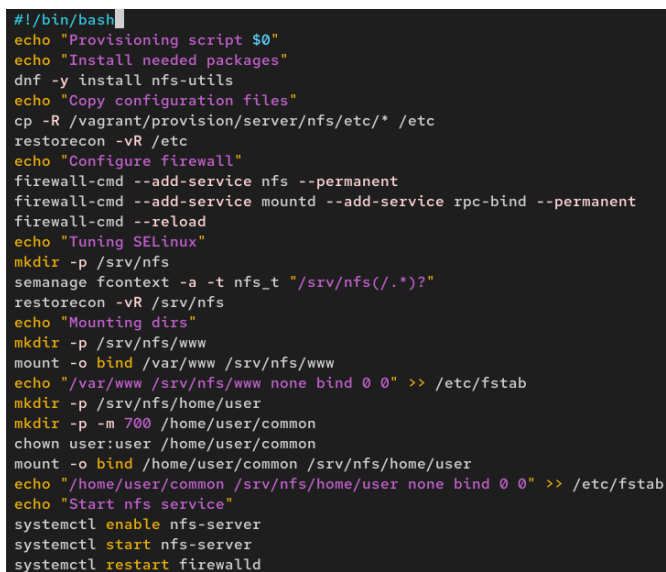
```
[svivanov@server.svivanov.net common]$ cd /vagrant/provision/server
[svivanov@server.svivanov.net server]$ mkdir -p /vagrant/provision/server/nfs/etc
[svivanov@server.svivanov.net server]$ cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[svivanov@server.svivanov.net server]$
```

Рис. 3.39: Создание каталогов и копирование конф.файлов

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 (рис. 40)



```
#!/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. 3.40: Скрипт `nfs.sh`

На виртуальной машине `client` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`: `cd /vagrant/provision/client`. В каталоге `/vagrant/provision/client` создадим исполняемый файл `nfs.sh`:

```
cd /vagrant/provision/client
touch nfs.sh
chmod +x nfs.sh (рис. 41)
```

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

Рис. 3.41: Создание скрипта

Открыв его на редактирование, пропишем в нём следующий скрипт: (рис. 42)

```
#!/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. 3.42: Скрипт nfs.sh

Для отработки созданных скриптов во время загрузки виртуальных машин server и client в конфигурационном файле Vagrantfile необходимо добавить в соответствующих разделах конфигураций для сервера и клиента: (рис. 43, 44)

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

Рис. 3.43: Редактирование Vagrantfile

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

Рис. 3.44: Редактирование Vagrantfile

4 Ответы на контрольные вопросы

1. Как называется файл конфигурации, содержащий общие ресурсы NFS?

Файл конфигурации, содержащий общие ресурсы NFS, называется `/etc/exports`.

2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS?

Для доступа к серверу NFSv4 должны быть открыты:

`nfs` - основной сервис NFS (порт 2049)

`rpc-bind` (`portmapper`) - служба RPC-привязки (порт 111)

`mountd` - служба монтирования NFS

3. Какую опцию следует использовать в `/etc/fstab`, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке?

В файле `/etc/fstab` для NFS-ресурсов следует использовать опцию `_netdev`.

5 Выводы

В ходе выполнения лабораторной работы мы приобрели навыки настройки сервера NFS для удалённого доступа к ресурсам.