

Отчёт по лабораторной работе 13

Настройка NFS

Сейдалиев Тагиретдин Ровшенович

Содержание

1	Цель работы	5
2	Выполнение	6
2.1	Настройка NFSv4 сервера и клиента	6
2.1.1	Развёртывание и подготовка NFS-сервера	6
2.1.2	Настройка SELinux и запуск NFS	7
2.2	Проверка работы с клиента	7
2.3	Монтирование NFS на клиенте	9
2.4	Добавление веб-контента в экспортируемое дерево	11
2.5	Подключение каталогов для работы пользователей	13
2.5.1	Создание пользовательского каталога и файла на сервере	13
2.5.2	Экспорт пользовательского каталога по NFS	14
2.5.3	Проверка работы на клиенте	15
2.6	Внесение изменений в настройки внутреннего окружения VM	17
3	Заключение	20
4	Контрольные вопросы	21
4.1	1. Как называется файл конфигурации, содержащий общие ресурсы NFS?	21
4.2	2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS?	21
4.3	3. Какую опцию следует использовать в /etc/fstab, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке?	22

Список иллюстраций

2.1	Установка nfs-utils и создание каталога /srv/nfs	6
2.2	Настройка экспорта каталога /srv/nfs	7
2.3	Настройка SELinux и запуск NFS-сервера	7
2.4	Ошибка RPC при выполнении showmount	8
2.5	Успешный showmount при отключённом firewall	8
2.6	Просмотр задействованных служб и добавление mountd/rpc-bind в firewall	9
2.7	Проверка монтирования /mnt/nfs	10
2.8	Запись NFS в /etc/fstab на клиенте	11
2.9	Экспорт каталога /srv/nfs/www	12
2.10	Запись bind-монтирования в /etc/fstab на сервере	12
2.11	Содержимое /srv/nfs и повторный экспорт каталогов	13
2.12	Появление каталога www на клиенте в /mnt/nfs	13
2.13	Создание каталога common и файла trseidaliev@server.txt	14
2.14	Добавление экспорта пользовательского каталога в /etc/exports	14
2.15	Запись bind-монтирования пользовательского каталога в /etc/fstab	15
2.16	Создание каталога /srv/nfs/home/trseidaliev, bind-монтирование и повторный exportfs	15
2.17	Создание файла на клиенте, отказ в доступе для root и итоговый список файлов	16
2.18	Просмотр файлов в каталоге common на сервере	17
2.19	Фрагмент скрипта nfs.sh на сервере	18
2.20	Фрагмент скрипта nfs.sh на клиенте	19

Список таблиц

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Выполнение

2.1 Настройка NFSv4 сервера и клиента

2.1.1 Развёртывание и подготовка NFS-сервера

На сервере установлено необходимое ПО для работы NFS, после чего создан корневой каталог экспортируемого дерева. В файл `/etc/exports` добавлена запись, предоставляющая доступ всем клиентам сети в режиме только чтение.

```
Upgraded:
libipa_hbac-2.11.1-2.el10_1.1.x86_64
libsmclient-4.22.4-106.el10.x86_64
libsss_idmap-2.11.1-2.el10_1.1.x86_64
libsss_sudo-2.11.1-2.el10_1.1.x86_64
libtdb-1.4.13-100.el10.x86_64
libwbclient-4.22.4-106.el10.x86_64
samba-common-4.22.4-106.el10.noarch
sssd-2.11.1-2.el10_1.1.x86_64
sssd-client-2.11.1-2.el10_1.1.x86_64
sssd-common-pac-2.11.1-2.el10_1.1.x86_64
sssd-kcm-2.11.1-2.el10_1.1.x86_64
sssd-krb5-common-2.11.1-2.el10_1.1.x86_64
sssd-proxy-2.11.1-2.el10_1.1.x86_64
libldb-4.22.4-106.el10.x86_64
libsss_certmap-2.11.1-2.el10_1.1.x86_64
libsss_nss_idmap-2.11.1-2.el10_1.1.x86_64
libtalloc-2.4.3-100.el10.x86_64
libtevent-0.16.2-100.el10.x86_64
samba-client-libs-4.22.4-106.el10.x86_64
samba-common-libs-4.22.4-106.el10.x86_64
sssd-ad-2.11.1-2.el10_1.1.x86_64
sssd-common-2.11.1-2.el10_1.1.x86_64
sssd-ipa-2.11.1-2.el10_1.1.x86_64
sssd-krb5-2.11.1-2.el10_1.1.x86_64
sssd-ldap-2.11.1-2.el10_1.1.x86_64

Installed:
gssproxy-0.9.2-10.el10.x86_64
libev-4.33-14.el10.x86_64
libnfsidmap-1:2.8.3-0.el10.x86_64
libverto-libev-0.3.2-10.el10.x86_64
nfs-utils-1:2.8.3-0.el10.x86_64
rpcbind-1.2.7-3.el10.x86_64
sssd-nfs-idmap-2.11.1-2.el10_1.1.x86_64

Complete!
[root@server.trseidaliev.net ~]# mkdir -p /srv/nfs
[root@server.trseidaliev.net ~]# █
```

Рис. 2.1: Установка nfs-utils и создание каталога `/srv/nfs`

Файл `/etc/exports` после настройки содержит строку с экспортируемым каталогом:

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

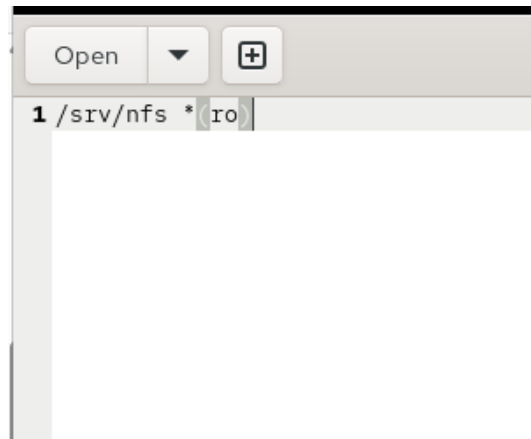


Рис. 2.2: Настройка экспорта каталога /srv/nfs

2.1.2 Настройка SELinux и запуск NFS

Для каталога /srv/nfs назначен контекст безопасности nfs_t, после чего изменения применены к файловой системе. Далее служба NFS была запущена, добавлена в автозагрузку, а в firewall разрешён сервис NFS.

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

Рис. 2.3: Настройка SELinux и запуск NFS-сервера

2.2 Проверка работы с клиента

На клиенте установлено ПО NFS, после чего выполнена команда для просмотра экспортированных ресурсов. Запрос завершается ошибкой из-за того, что часть RPC-служб была заблокирована межсетевым экраном.

```

Upgraded:
libipa_hbac-2.11.1-2.el10_1.1.x86_64
libsmclient-4.22.4-106.el10.x86_64
libsss_idmap-2.11.1-2.el10_1.1.x86_64
libsss_sudo-2.11.1-2.el10_1.1.x86_64
libtdb-1.4.13-100.el10.x86_64
libwbclient-4.22.4-106.el10.x86_64
samba-common-4.22.4-106.el10.noarch
sssd-2.11.1-2.el10_1.1.x86_64
sssd-client-2.11.1-2.el10_1.1.x86_64
sssd-common-pac-2.11.1-2.el10_1.1.x86_64
sssd-kcm-2.11.1-2.el10_1.1.x86_64
sssd-krb5-common-2.11.1-2.el10_1.1.x86_64
sssd-proxy-2.11.1-2.el10_1.1.x86_64
libldb-4.22.4-106.el10.x86_64
libsss_certmap-2.11.1-2.el10_1.1.x86_64
libsss_nss_idmap-2.11.1-2.el10_1.1.x86_64
libtalloc-2.4.3-100.el10.x86_64
libtevent-0.16.2-100.el10.x86_64
samba-client-libs-4.22.4-106.el10.x86_64
samba-common-libs-4.22.4-106.el10.x86_64
sssd-ad-2.11.1-2.el10_1.1.x86_64
sssd-common-2.11.1-2.el10_1.1.x86_64
sssd-ipa-2.11.1-2.el10_1.1.x86_64
sssd-krb5-2.11.1-2.el10_1.1.x86_64
sssd-ldap-2.11.1-2.el10_1.1.x86_64

Installed:
gssproxy-0.9.2-10.el10.x86_64
libnfsidmap-1:2.8.3-0.el10.x86_64
nfs-utils-1:2.8.3-0.el10.x86_64
sssd-nfs-idmap-2.11.1-2.el10_1.1.x86_64
libev-4.33-14.el10.x86_64
libverto-libev-0.3.2-10.el10.x86_64
rpcbind-1.2.7-3.el10.x86_64

Complete!
[root@client.trseidaliev.net ~]# showmount -e server.trseidaliev.net
clnt_create: RPC: Unable to receive
[root@client.trseidaliev.net ~]#

```

Рис. 2.4: Ошибка RPC при выполнении showmount

Команда обращается к RPC-службам mountd и rpcbind. Пока соответствующие порты не разрешены firewall, клиент не может получить ответ — возникает ошибка RPC.

После остановки firewall повторная попытка получить список экспортов проходит успешно. Это подтверждает, что причиной ошибки действительно были заблокированные порты RPC.

```

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

```

Рис. 2.5: Успешный showmount при отключённом firewall

Межсетевой экран затем был снова включён.

Для идентификации используемых служб были просмотрены активные TCP- и UDP-порты. В выводе обнаружены сервисы rpcbind, rpc.statd, mountd. Эти службы были добавлены в firewall, что позволило корректно выполнять запросы NFS-клиента.


```

ev.net:bootpc->_gateway:bootps
NetworkMa 8403 8411 pool-spaw          root 26u IPv4 46674 0t0 UDP server.trseidali
ev.net:bootpc->_gateway:bootps
NetworkMa 8403 8412 gdbus             root 26u IPv4 46674 0t0 UDP server.trseidali
ev.net:bootpc->_gateway:bootps
rpcbind 15647                          rpc 6u IPv4 61648 0t0 UDP *:sunrpc
rpcbind 15647                          rpc 8u IPv6 61662 0t0 UDP *:sunrpc
rpc.statd 15653                        rpcuser 7u IPv4 70271 0t0 UDP *:58695
rpc.statd 15653                        rpcuser 9u IPv6 70277 0t0 UDP *:37946
rpc.statd 15653                        rpcuser 11u IPv4 70268 0t0 UDP localhost:994
rpc.mount 15673                        root 4u IPv4 70405 0t0 UDP *:mountd
rpc.mount 15673                        root 6u IPv6 70411 0t0 UDP *:mountd
[root@server.trseidali.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.trseidali.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.trseidali.net ~]# firewall-cmd --reload
success
[root@server.trseidali.net ~]#

```

Рис. 2.6: Просмотр задействованных служб и добавление mountd/rpc-bind в firewall

После добавления служб в правила firewall запрос showmount снова выполняется успешно:

2.3 Монтирование NFS на клиенте

На клиенте создан каталог для точки монтирования. После подключения ресурса проверка показывает, что файловая система смонтирована корректно и работает через протокол NFS версии 4.2.

```

[root@client.trseidaliev.net ~]#
[root@client.trseidaliev.net ~]# mkdir -p /mnt/nfs
[root@client.trseidaliev.net ~]# mount server.trseidaliev.net:/srv/nfs /mnt/nfs
[root@client.trseidaliev.net ~]# mount | grep mnt
server.trseidaliev.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
[root@client.trseidaliev.net ~]# gedit /etc/fstab
[root@client.trseidaliev.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 Mon 2025-12-08 11:30:51 MSK; 11min ago
 Invocation: 3f0ea87b278d42a4b328d33e4166dd65
    Docs: man:systemd.special(7)

Dec 08 11:30:51 client.trseidaliev.net systemd[1]: Reached target remote-fs.target - Remote File
[root@client.trseidaliev.net ~]# mount -a
[root@client.trseidaliev.net ~]# mount | grep mnt
server.trseidaliev.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1)
[root@client.trseidaliev.net ~]#

```

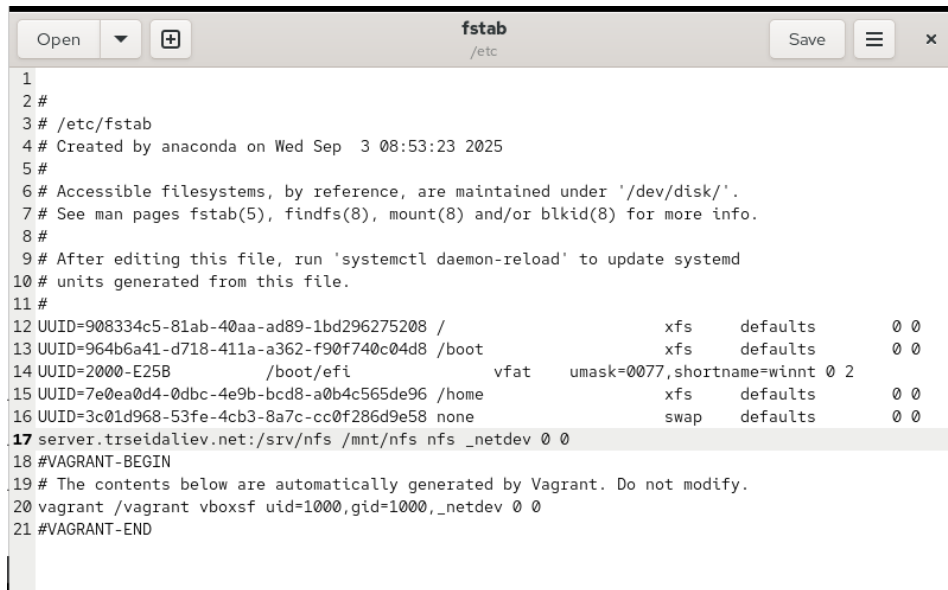
Рис. 2.7: Проверка монтирования /mnt/nfs

Пояснение параметров монтирования:

- используется протокол NFSv4,
- транспорт — TCP,
- указаны размеры блоков, параметры таймаутов,
- отображён IP-адрес клиента,
- ресурс успешно подмонтирован в /mnt/nfs.

В /etc/fstab добавлена строка для автоматического подключения ресурса при загрузке системы:

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



```
1
2 #
3 # /etc/fstab
4 # Created by anaconda on Wed Sep 3 08:53:23 2025
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=908334c5-81ab-40aa-ad89-1bd296275208 / xfs defaults 0 0
13 UUID=964b6a41-d718-411a-a362-f90f740c04d8 /boot xfs defaults 0 0
14 UUID=2000-E258 /boot/efi vfat umask=0077,shortname=winnt 0 2
15 UUID=7e0ea0d4-0dbc-4e9b-bcd8-a0b4c565de96 /home xfs defaults 0 0
16 UUID=3c01d968-53fe-4cb3-8a7c-cc0f286d9e58 none swap defaults 0 0
17 server.trseidaliev.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
18 #VAGRANT-BEGIN
19 # The contents below are automatically generated by Vagrant. Do not modify.
20 vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
21 #VAGRANT-END
```

Рис. 2.8: Запись NFS в /etc/fstab на клиенте

Пояснение синтаксиса:

- первая часть — путь к экспортируемому каталогу сервера,
- вторая — локальная точка монтирования,
- тип — NFS,
- `_netdev` — отложенное монтирование до поднятия сети,
- `0 0` — отключены `dump` и `fsck`.

Проверка состояния `remote-fs.target` подтверждает, что автоматическое монтирование работает корректно. После перезапуска клиент автоматически подключает удалённый ресурс.

2.4 Добавление веб-контента в экспортируемое дерево

На сервере создан каталог `/srv/nfs/www`, в который через `bind` было подключено содержимое `/var/www`. Теперь веб-контент стал частью экспортируемой структуры.

В файл `/etc/exports` добавлена строка:

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

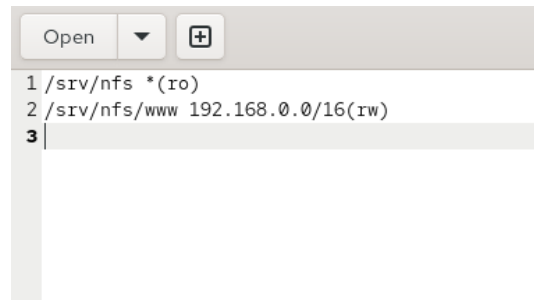


Рис. 2.9: Экспорт каталога `/srv/nfs/www`

После обновления экспорта клиенты получают доступ к содержимому, доступному в каталоге `/srv/nfs/www`.

В файл `/etc/fstab` на сервере внесена строка:

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

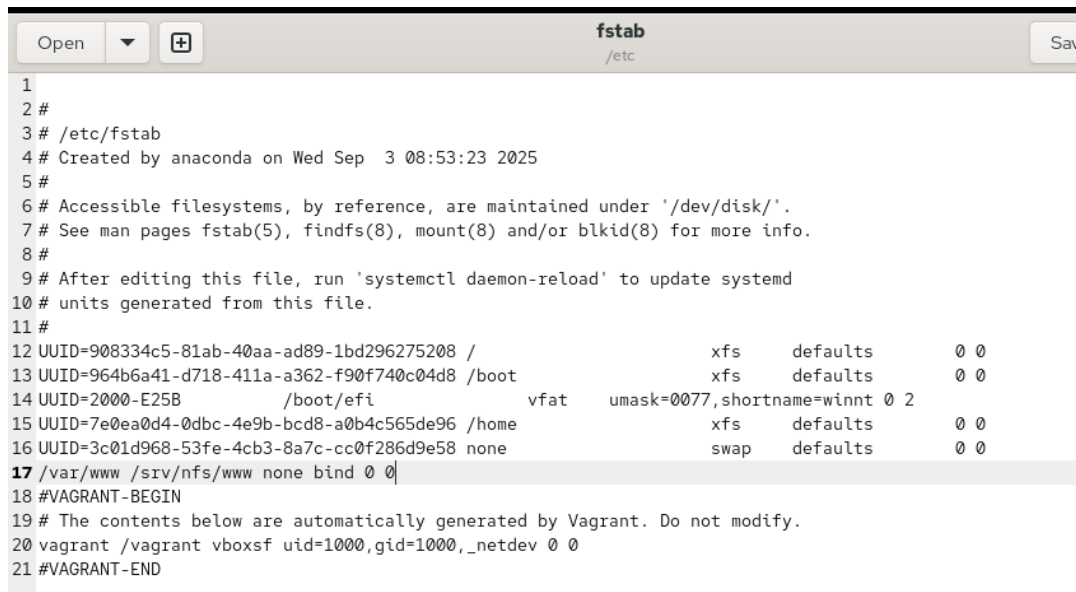


Рис. 2.10: Запись `bind`-монтирования в `/etc/fstab` на сервере

Эта запись обеспечивает автоматическое подключение каталога веб-сервера в дерево NFS при каждом запуске системы.

После добавления экспорта каталога веб-сервера и записи bind-монтирования в `/etc/fstab` на сервере были повторно экспортированы каталоги, указанные в `/etc/exports`.

Каталог `/srv/nfs` теперь содержит подкаталог `www`, соответствующий привязанному (bind) каталогу `/var/www`.

```
[root@server.trseidaliev.net ~]#  
[root@server.trseidaliev.net ~]# mkdir -p /srv/nfs/www  
[root@server.trseidaliev.net ~]# mount -o bind /var/www /srv/nfs/www/  
[root@server.trseidaliev.net ~]# ls /srv/nfs/  
www  
[root@server.trseidaliev.net ~]#  
[root@server.trseidaliev.net ~]# gedit /etc/exports  
[root@server.trseidaliev.net ~]# exportfs -r  
[root@server.trseidaliev.net ~]# gedit /etc/fstab  
[root@server.trseidaliev.net ~]# exportfs -r  
[root@server.trseidaliev.net ~]# █
```

Рис. 2.11: Содержимое `/srv/nfs` и повторный экспорт каталогов

На клиенте при проверке каталога `/mnt/nfs` отображается подкаталог `www`, что подтверждает доступность экспортированного дерева NFS.

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

Рис. 2.12: Появление каталога `www` на клиенте в `/mnt/nfs`

2.5 Подключение каталогов для работы пользователей

2.5.1 Создание пользовательского каталога и файла на сервере

Под пользователем `trseidaliev` в домашнем каталоге создан каталог `common` с правами доступа `700`, то есть полный доступ только у владельца. Внутри каталога создан файл `trseidaliev@server.txt`.

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

Рис. 2.13: Создание каталога common и файла trseidaliev@server.txt

Таким образом:

- только пользователь trseidaliev имеет доступ к содержимому ~/common;
- остальные пользователи, включая root при обращении по NFS (с учётом механизма root_squash), не могут читать или изменять этот каталог.

2.5.2 Экспорт пользовательского каталога по NFS

На сервере создан каталог /srv/nfs/home/trseidaliev, в который затем будет подмонтирован каталог common.

В файле /etc/exports добавлен экспорт пользовательского каталога:

- строка /srv/nfs/home/trseidaliev 192.168.0.0/16(rw) разрешает доступ на чтение и запись всем хостам из подсети 192.168.0.0/16.

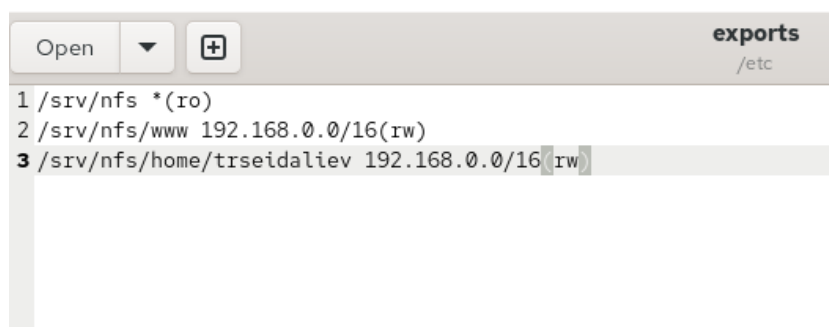


Рис. 2.14: Добавление экспорта пользовательского каталога в /etc/exports

В том же файле /etc/fstab настроено bind-монтирование:

- строка `/home/trseidaliev/common /srv/nfs/home/trseidaliev none bind 0` обеспечивает автоматическое подключение каталога `common` в дерево `/srv/nfs/home`.

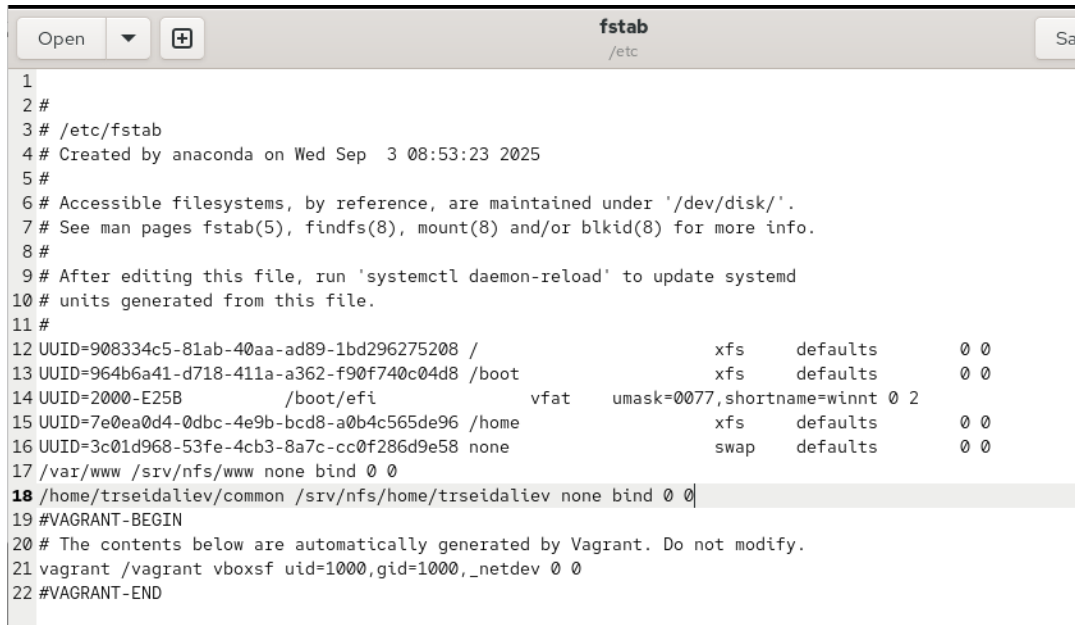


Рис. 2.15: Запись bind-монтирования пользовательского каталога в `/etc/fstab`

Далее каталоги были повторно экспортированы, после чего пользовательский каталог стал доступен по NFS:

```

[root@server.trseidaliev.net ~]#
[root@server.trseidaliev.net ~]# mkdir -p /srv/nfs/home/trseidaliev
[root@server.trseidaliev.net ~]# mount -o bind /home/trseidaliev/common /srv/nfs/home/trseidaliev/
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@server.trseidaliev.net ~]# gedit /etc/exports
[root@server.trseidaliev.net ~]# gedit /etc/fstab
[root@server.trseidaliev.net ~]# exportfs -r
[root@server.trseidaliev.net ~]# █

```

Рис. 2.16: Создание каталога `/srv/nfs/home/trseidaliev`, bind-монтирование и повторный `exportfs`

2.5.3 Проверка работы на клиенте

На клиенте в каталоге `/mnt/nfs` теперь видна структура `home/www`. Пользователь `trseidaliev` переходит в каталог `/mnt/nfs/home/trseidaliev` и со-

здаёт файл `trseidaliev@client.txt`. В результате в каталоге отображаются два файла: `trseidaliev@server.txt` и `trseidaliev@client.txt`.

Попытка зайти в тот же каталог с клиента под пользователем `root` завершается ошибкой `Permission denied`. Это связано с тем, что:

- каталог `/home/trseidaliev/common` на сервере имеет права `700`;
- на стороне NFS включён механизм `root_squash`, который подменяет пользователя `root` на клиентах на непривилегированного пользователя, не имеющего прав доступа к каталогу.

```
[trseidaliev@client.trseidaliev.net ~]#  
[trseidaliev@client.trseidaliev.net ~]$ cd /mnt/nfs/  
[trseidaliev@client.trseidaliev.net nfs]$ ls  
home  www  
[trseidaliev@client.trseidaliev.net nfs]$ cd home/trseidaliev/  
[trseidaliev@client.trseidaliev.net trseidaliev]$ ls  
trseidaliev@server.txt  
[trseidaliev@client.trseidaliev.net trseidaliev]$ touch trseidaliev@client.txt  
[trseidaliev@client.trseidaliev.net trseidaliev]$ ls  
trseidaliev@client.txt  trseidaliev@server.txt  
[trseidaliev@client.trseidaliev.net trseidaliev]$ sudo -i  
[sudo] password for trseidaliev:  
[root@client.trseidaliev.net ~]# cd /mnt/nfs/home/trseidaliev/  
-bash: cd: /mnt/nfs/home/trseidaliev/: Permission denied  
[root@client.trseidaliev.net ~]#  
logout  
[trseidaliev@client.trseidaliev.net trseidaliev]$ ls  
trseidaliev@client.txt  trseidaliev@server.txt  
[trseidaliev@client.trseidaliev.net trseidaliev]$ ls -l  
total 0  
-rw-r--r--. 1 trseidaliev trseidaliev 0 Dec  8 11:51 trseidaliev@client.txt  
-rw-r--r--. 1 trseidaliev trseidaliev 0 Dec  8 11:48 trseidaliev@server.txt  
[trseidaliev@client.trseidaliev.net trseidaliev]$ █
```

Рис. 2.17: Создание файла на клиенте, отказ в доступе для `root` и итоговый список файлов

На сервере в каталоге `~/common` видны оба файла — и созданный на сервере, и созданный на клиенте. Это подтверждает корректную синхронизацию содержимого каталога через NFS.

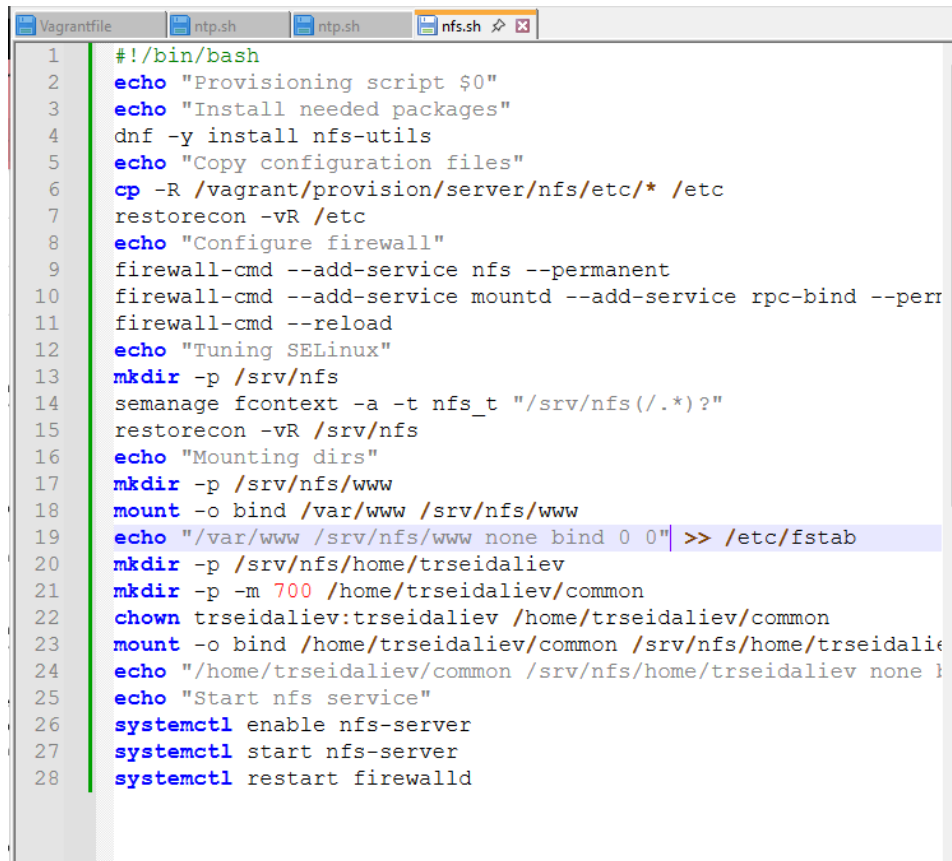

```
[trseidaliev@server.trseidaliev.net common]$ ls
trseidaliev@server.txt
[trseidaliev@server.trseidaliev.net common]$
[trseidaliev@server.trseidaliev.net common]$ ls -l
total 0
-rw-r--r--. 1 trseidaliev trseidaliev 0 Dec  8 11:51 trseidaliev@client.txt
-rw-r--r--. 1 trseidaliev trseidaliev 0 Dec  8 11:48 trseidaliev@server.txt
[trseidaliev@server.trseidaliev.net common]$ █
```

Рис. 2.18: Просмотр файлов в каталоге common на сервере

2.6 Внесение изменений в настройки внутреннего окружения VM

На виртуальной машине `server` в каталоге `/vagrant/provision/server` создан подкаталог `nfs/etc`. В него скопирован конфигурационный файл `/etc/exports`. Это позволяет хранить «шаблон» настроек экспорта внутри каталога, доступного Vagrant, и использовать его при автоматическом развертывании стенда.

В каталоге `/vagrant/provision/server` создан исполняемый файл `nfs.sh`.

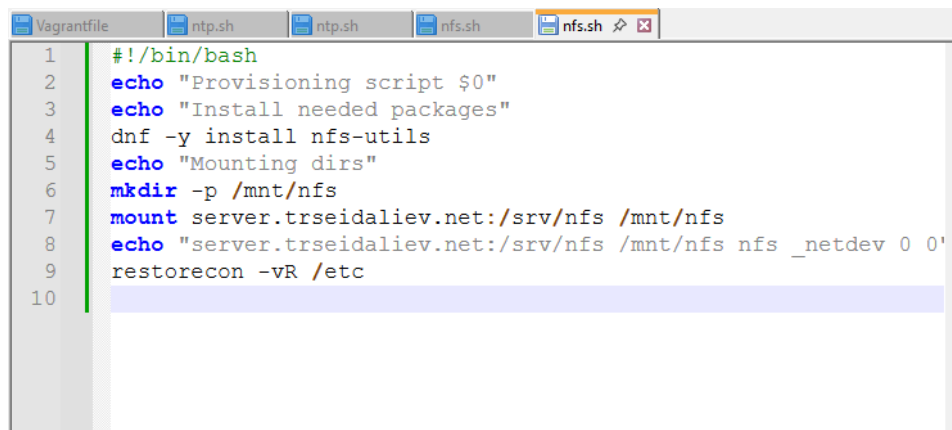


```
1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Install needed packages"
4  dnf -y install nfs-utils
5  echo "Copy configuration files"
6  cp -R /vagrant/provision/server/nfs/etc/* /etc
7  restorecon -vR /etc
8  echo "Configure firewall"
9  firewall-cmd --add-service nfs --permanent
10 firewall-cmd --add-service mountd --add-service rpc-bind --per
11 firewall-cmd --reload
12 echo "Tuning SELinux"
13 mkdir -p /srv/nfs
14 semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
15 restorecon -vR /srv/nfs
16 echo "Mounting dirs"
17 mkdir -p /srv/nfs/www
18 mount -o bind /var/www /srv/nfs/www
19 echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
20 mkdir -p /srv/nfs/home/trseidaliev
21 mkdir -p -m 700 /home/trseidaliev/common
22 chown trseidaliev:trseidaliev /home/trseidaliev/common
23 mount -o bind /home/trseidaliev/common /srv/nfs/home/trseidaliev
24 echo "/home/trseidaliev/common /srv/nfs/home/trseidaliev none bind 0 0" >> /etc/fstab
25 echo "Start nfs service"
26 systemctl enable nfs-server
27 systemctl start nfs-server
28 systemctl restart firewalld
```

Рис. 2.19: Фрагмент скрипта nfs.sh на сервере

В результате при автоматическом провижининге VM server весь набор настроек NFS (экспорты, firewall, SELinux, bind-монтирования и пользовательские каталоги) разворачивается автоматически.

На виртуальной машине client в каталоге /vagrant/provision/client создан исполняемый файл nfs.sh.



The image shows a Vagrant terminal window with four tabs: 'Vagrantfile', 'ntp.sh', 'ntp.sh', and 'nfs.sh'. The 'nfs.sh' tab is active, displaying a shell script. The script starts with a shebang line, followed by two echo statements for provisioning and installation. It then uses 'dnf' to install 'nfs-utils', echoes a message about mounting, creates the '/mnt/nfs' directory with 'mkdir', mounts the remote NFS share from 'server.trseidaliev.net' to '/mnt/nfs', echoes the mount command, and finally runs 'restorecon' on the '/etc' directory. Line numbers 1 through 10 are visible on the left side of the script.

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

Рис. 2.20: Фрагмент скрипта nfs.sh на клиенте

3 Заключение

В ходе выполнения работы:

- настроен сервер NFSv4 с учётом SELinux и межсетевого экрана;
- подготовлены каталоги для экспорта, включая пользовательские и системные (`www`, `home/user/common`);
- выполнено `bind`-монтирование каталогов в дерево NFS;
- настроены автоматические подключения через `/etc/fstab` на сервере и клиенте;
- корректно организована работа пользовательских каталогов с учётом прав доступа и механизма `root_squash`;
- созданы скрипты автоматического провижининга для серверной и клиентской виртуальной машины.

4 Контрольные вопросы

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

Файл конфигурации, в котором перечислены экспортируемые сервером каталоги NFS, называется:

`/etc/exports`

В нём задаются пути, параметры доступа и список клиентов, которым разрешено подключение.

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

Для корректной работы NFS необходимо разрешить доступ к следующим службам:

- **nfs** — основной сервис NFS;
- **rpc-bind** — назначение динамических RPC-портов;
- **mountd** — обслуживание запросов `mount` и `showmount`.

Эти службы автоматически открывают необходимые порты, включая:

- 2049/tcp — порт NFSv4;
- 111/tcp и 111/udp — порт rpcbind;
- динамические порты для mountd и rpc.statd.

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

Для автоматического монтирования сетевых ресурсов NFS при загрузке используется опция:

`_netdev`

Она указывает системе, что устройство зависит от сети, и его следует монтировать только после поднятия сетевых служб. Это предотвращает ошибки при загрузке и обеспечивает корректное подключение NFS-ресурсов.