

Лабораторная работа №13

Презентация

Мосолов А.Д.

29 ноября 2025

Российский университет дружбы народов, Москва, Россия

- Мосолов Александр Денисович
- Студент, НПИбд02-23
- Российский университет дружбы народов
- 1132236128@pfur.ru

Приобретение навыков настройки сервера NFS (Network File System) для организации удалённого доступа к файловым ресурсам в локальной сети.

1. Установить и настроить сервер NFSv4.
2. Подмонтировать удалённый ресурс на клиенте.
3. Подключить каталог с контентом веб-сервера к дереву NFS.
4. Подключить домашний каталог пользователя к дереву NFS.
5. Написать скрипты автоматизации настройки (provision) для Vagrant.

Установка nfs-utils на сервере

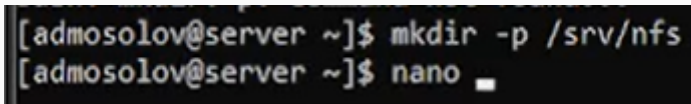
Первым шагом я устанавливаю необходимый пакет `nfs-utils` на сервере. Этот пакет содержит утилиты для работы как сервера, так и клиента NFS. Как видно из вывода, пакет уже был установлен в системе.

```
[admosolov@server ~]$ sudo dnf -y install nfs-utils
Extra Packages for Enterprise Linux 9 - x86_64          24 kB/s | 38 kB    00:01
Extra Packages for Enterprise Linux 9 - x86_64          3.7 MB/s | 20 MB    00:05
Rocky Linux 9 - BaseOS                                  1.9 kB/s | 4.1 kB    00:02
Rocky Linux 9 - AppStream                                5.7 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                                  7.3 kB/s | 2.9 kB    00:00
Package nfs-utils-1:2.5.4-34.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
```

Рис. 1: Установка `nfs-utils` на сервере

Создание корневого каталога NFS

Далее я создаю основной каталог, который будет служить корнем для экспортируемой файловой системы NFS (/srv/nfs). После этого я открываю текстовый редактор для настройки конфигурационного файла экспорта.



```
[admosolov@server ~]$ mkdir -p /srv/nfs  
[admosolov@server ~]$ nano _
```

Рис. 2: Создание корневого каталога NFS

Настройка /etc/exports

В файл /etc/exports я добавляю запись, разрешающую доступ к каталогу /srv/nfs для всех хостов (*) в режиме только для чтения (ro). Это базовая настройка безопасности.

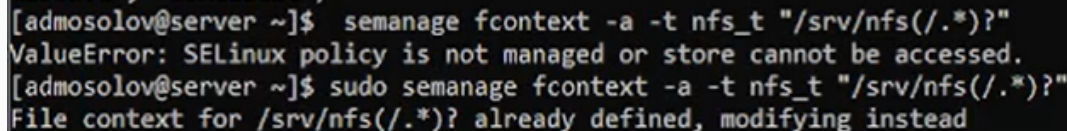


```
admosolov@server:~  
GNU nano 5.6.1 /etc/exports  
/srv/nfs *(ro)
```

Рис. 3: Настройка /etc/exports

Настройка контекста SELinux

Для корректной работы NFS с включенным SELinux необходимо настроить контекст безопасности. Я выполняю команду `semanage fcontext` (исправив опечатку с `sudo`), чтобы назначить тип `nfs_t` для каталога `/srv/nfs` и его содержимого.

A terminal window with a black background and white text. The first command is `[admosolov@server ~]$ semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"`, which results in an error: `ValueError: SELinux policy is not managed or store cannot be accessed.` The second command is `[admosolov@server ~]$ sudo semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"`, which results in the message: `File context for /srv/nfs(/.*)? already defined, modifying instead`.

```
[admosolov@server ~]$ semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
ValueError: SELinux policy is not managed or store cannot be accessed.
[admosolov@server ~]$ sudo semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
File context for /srv/nfs(/.*)? already defined, modifying instead
```

Рис. 4: Настройка контекста SELinux

Запуск службы NFS

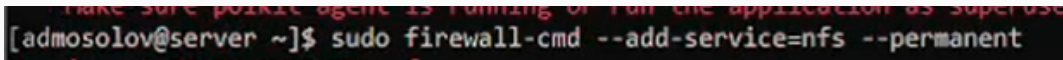
После определения правил SELinux я применяю их командой `restorecon`. Затем я запускаю службу NFS-сервера и добавляю её в автозагрузку, чтобы сервис поднимался после перезагрузки системы.

```
[admosolov@server ~]$ sudo restorecon -vR /srv/nfs
[admosolov@server ~]$ systemctl start nfs-server.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'nfs-server.service'.
Multiple identities can be used for authentication:
 1. avlisovskaya
 2. admosolov
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ====
[admosolov@server ~]$ systemctl enable nfs-server.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Multiple identities can be used for authentication:
 1. avlisovskaya
 2. admosolov
Choose identity to authenticate as (1-2): 2
Password: █
```

Рис. 5: Запуск службы NFS

Настройка firewall для NFS

Для доступа клиентов к серверу необходимо открыть соответствующие порты в брандмауэре. Я добавляю службу `nfs` в постоянные правила `firewall-cmd`.

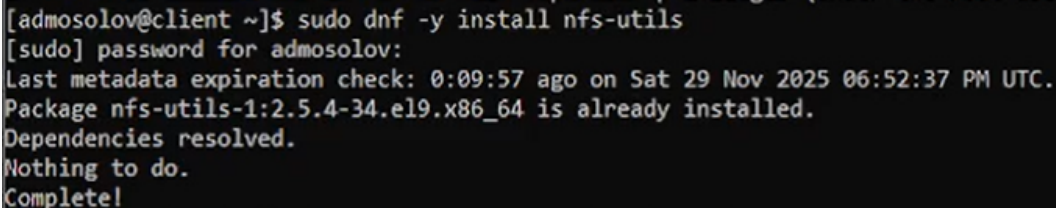
A terminal window with a dark background and red text. The prompt is [admosolov@server ~]\$. The command entered is sudo firewall-cmd --add-service=nfs --permanent. The output is not visible.

```
[admosolov@server ~]$ sudo firewall-cmd --add-service=nfs --permanent
```

Рис. 6: Настройка firewall для NFS

Установка nfs-utils на клиенте

Перехожу к настройке клиента. На клиентской машине также необходимо установить пакет `nfs-utils`, чтобы система могла монтировать удаленные ресурсы.

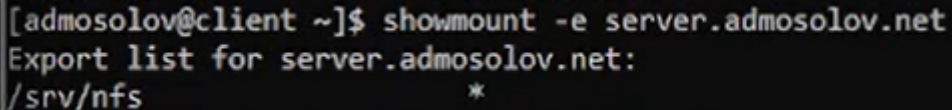
A terminal window with a black background and white text. The prompt is [admosolov@client ~]\$. The command sudo dnf -y install nfs-utils is entered. The output shows a password prompt, a metadata expiration check timestamp, a message that the package is already installed, and a completion message.

```
[admosolov@client ~]$ sudo dnf -y install nfs-utils
[sudo] password for admosolov:
Last metadata expiration check: 0:09:57 ago on Sat 29 Nov 2025 06:52:37 PM UTC.
Package nfs-utils-1:2.5.4-34.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

Рис. 7: Установка `nfs-utils` на клиенте

Проверка экспорта с клиента

С клиентской машины я проверяю доступность экспортируемых ресурсов на сервере с помощью команды `showmount -e`. Вывод показывает, что каталог `/srv/nfs` доступен.



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

Рис. 8: Проверка экспорта с клиента

Добавление служб RPC в firewall

Возвращаюсь на сервер для донастройки брандмауэра. Поскольку NFS использует RPC, необходимо также разрешить службы mountd и rpc-bind. Сначала я проверяю открытые порты, а затем добавляю необходимые службы.

```
[admosolov@server ~]$ lsof | grep TCP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/990/gvfs
Output information may be incomplete.
[admosolov@server ~]$ sudo firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
Warning: ALREADY_ENABLED: mountd
Warning: ALREADY_ENABLED: rpc-bind
success

[admosolov@server ~]$ sudo firewall-cmd --reload

success
[admosolov@server ~]$
[admosolov@server ~]$ lsof | grep TCP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/990/gvfs
Output information may be incomplete.
[admosolov@server ~]$
```

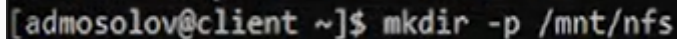
Список разрешенных сервисов

Убеждаюсь, что все необходимые службы (nfs, mountd, rpc-bind) добавлены в список разрешенных сервисов брандмауэра.

```
[admosolov@server ~]$ sudo firewall-cmd --list-services  
cockpit dhcp dhcpv6-client dns http https imap imaps mountd mysql nfs ntp pop3 pop3s rpc-bind samba smtp  
smtp-submission ssh  
[admosolov@server ~]$
```

Рис. 10: Список разрешенных сервисов

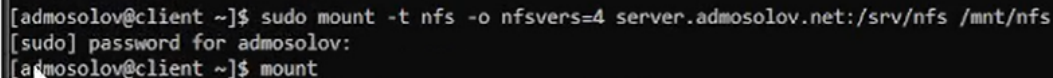
На клиентской машине создаю директорию `/mnt/nfs`, которая будет использоваться как точка монтирования для удаленного ресурса.

A terminal window with a black background and white text. The prompt is `[admosolov@client ~]` and the command entered is `$ mkdir -p /mnt/nfs`.

```
[admosolov@client ~]$ mkdir -p /mnt/nfs
```

Рис. 11: Создание точки монтирования

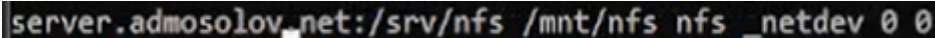
Выполняю монтирование удаленного каталога `/srv/nfs` с сервера в локальную директорию `/mnt/nfs`, явно указывая использование протокола NFS версии 4.

A terminal window with a black background and white text. The prompt is [admosolov@client ~]. The command entered is sudo mount -t nfs -o nfsvers=4 server.admosolov.net:/srv/nfs /mnt/nfs. The next line shows [sudo] password for admosolov: followed by a cursor. The final line shows [admosolov@client ~]\$ mount.

```
[admosolov@client ~]$ sudo mount -t nfs -o nfsvers=4 server.admosolov.net:/srv/nfs /mnt/nfs
[sudo] password for admosolov:
[admosolov@client ~]$ mount
```

Рис. 12: Монтирование ресурса

Проверяю успешность операции командой `mount`. Видно, что ресурс `server.admosolov.net:/srv/nfs` успешно смонтирован. Параметр `_netdev` указывает, что это сетевое устройство.

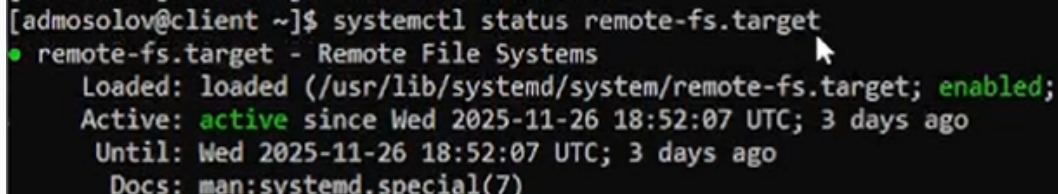
A terminal window with a dark background showing the command `server.admosolov_net:/srv/nfs /mnt/nfs nfs _netdev 0 0` being executed. The text is in a light-colored monospace font.

```
server.admosolov_net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рис. 13: Проверка монтирования

Статус remote-fs.target

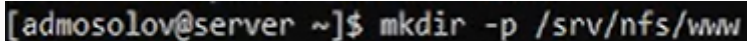
Также проверяю статус системного юнита `remote-fs.target`, который отвечает за работу с удаленными файловыми системами. Он находится в активном состоянии.

A terminal window with a black background and light green text. The prompt is [admosolov@client ~]\$. The command systemctl status remote-fs.target has been entered. The output shows that the unit is loaded and enabled, and has been active since Wednesday, November 26, 2025, at 18:52:07 UTC, 3 days ago. The documentation path is man:systemd.special(7).

```
[admosolov@client ~]$ systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled;
   Active: active since Wed 2025-11-26 18:52:07 UTC; 3 days ago
   Until: Wed 2025-11-26 18:52:07 UTC; 3 days ago
   Docs: man:systemd.special(7)
```

Рис. 14: Статус remote-fs.target

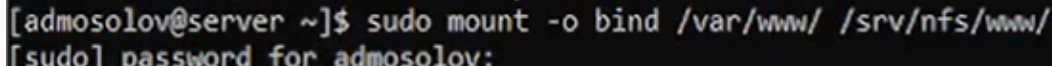
Теперь я перехожу к настройке экспорта каталога веб-сервера. На сервере создаю подкаталог `/srv/nfs/www` внутри дерева NFS.

A terminal window with a black background and white text. The prompt is [admosolov@server ~]\$. The command entered is mkdir -p /srv/nfs/www.

```
[admosolov@server ~]$ mkdir -p /srv/nfs/www
```

Рис. 15: Создание каталога для www

Монтирую реальный каталог веб-сервера /var/www в созданный подкаталог NFS с использованием опции --bind. Это позволяет “пробросить” существующую директорию в дерево экспорта.

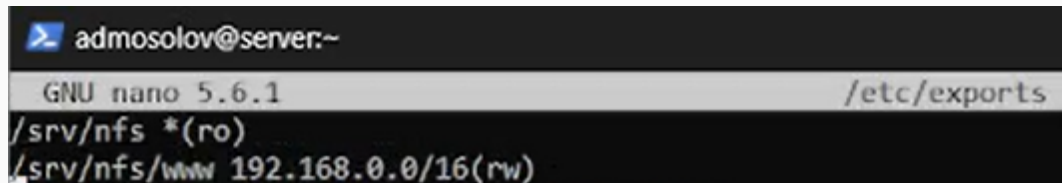


```
[admosolov@server ~]$ sudo mount -o bind /var/www/ /srv/nfs/www/  
[sudo] password for admosolov:
```

Рис. 16: Bind-монтирование www

Обновление exports для www

Редактирую файл `/etc/exports` для настройки прав доступа к новому ресурсу. Я разрешаю доступ к `/srv/nfs/www` для подсети `192.168.0.0/16` с правами на чтение и запись (`rw`).

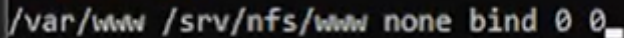


```
> admosolov@server:~  
GNU nano 5.6.1 /etc/exports  
/srv/nfs *(ro)  
/srv/nfs/www 192.168.0.0/16(rw)
```

Рис. 17: Обновление exports для www

Запись bind-mount в fstab

Чтобы bind-монтирование сохранялось после перезагрузки, я добавляю соответствующую запись в файл `/etc/fstab` на сервере.

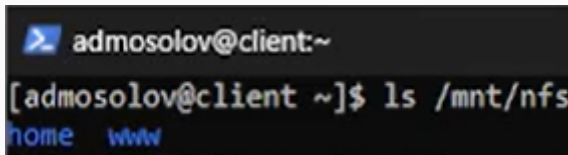


```
/var/www /srv/nfs/www none bind 0 0_
```

Рис. 18: Запись bind-mount в fstab

Проверка содержимого www на клиенте

На клиенте проверяю содержимое смонтированного ресурса `ls /mnt/nfs`.
Теперь там виден каталог `www` (внутри которого `home` и `html`), что подтверждает успешный проброс директории.

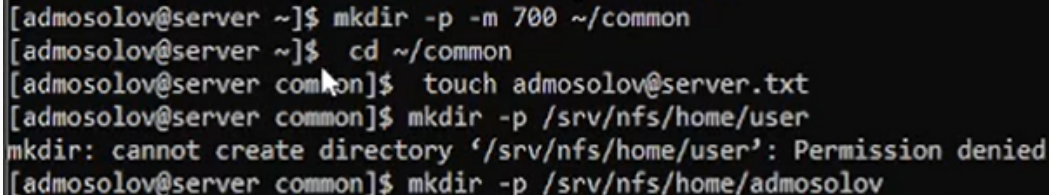


```
admosolov@client:~  
[admosolov@client ~]$ ls /mnt/nfs  
home  www
```

Рис. 19: Проверка содержимого `www` на клиенте

Подготовка каталога пользователя

Далее я настраиваю экспорт домашнего каталога пользователя. Создаю директорию `~/common`, тестовый файл в ней, а также соответствующую структуру каталогов в `/srv/nfs`, исправляя ошибки доступа при необходимости.

A terminal window with a black background and white text. The prompt is [admosolov@server ~]. The user enters 'mkdir -p -m 700 ~/common'. The prompt changes to [admosolov@server ~]. The user enters 'cd ~/common'. The prompt changes to [admosolov@server common]. The user enters 'touch admosolov@server.txt'. The prompt changes to [admosolov@server common]. The user enters 'mkdir -p /srv/nfs/home/user'. The terminal shows an error: 'mkdir: cannot create directory '/srv/nfs/home/user': Permission denied'. The user then enters 'mkdir -p /srv/nfs/home/admosolov'.

```
[admosolov@server ~]$ mkdir -p -m 700 ~/common
[admosolov@server ~]$ cd ~/common
[admosolov@server common]$ touch admosolov@server.txt
[admosolov@server common]$ mkdir -p /srv/nfs/home/user
mkdir: cannot create directory '/srv/nfs/home/user': Permission denied
[admosolov@server common]$ mkdir -p /srv/nfs/home/admosolov
```

Рис. 20: Подготовка каталога пользователя

Bind-монтирование каталога пользователя

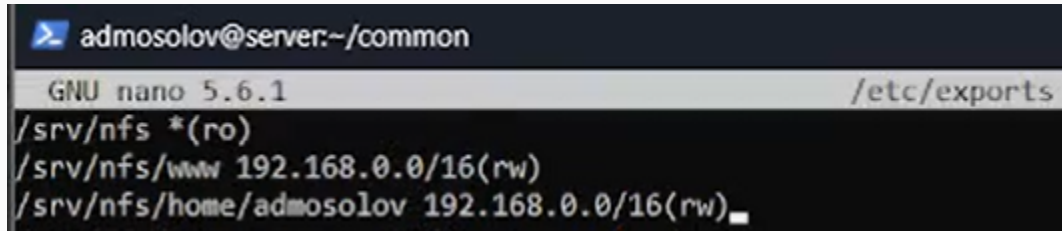
Выполняю bind-монтирование личного каталога пользователя в дерево NFS и открываю файл экспорта для настройки.

```
[admosolov@server common]$ sudo mount -o bind /home/admosolov/common /srv/nfs/home/admosolov  
[admosolov@server common]$ sudo nano /etc/exports
```

Рис. 21: Bind-монтирование каталога пользователя

Обновление exports для пользователя

Добавляю в `/etc/exports` запись для `/srv/nfs/home/admosolov`, разрешая доступ на чтение и запись для локальной сети.

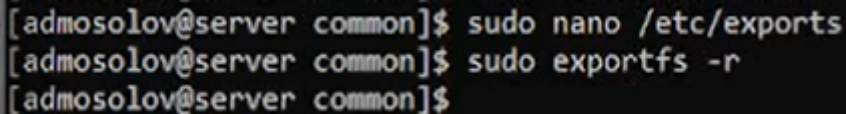


```
> admosolov@server:~/common
GNU nano 5.6.1 /etc/exports
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/admosolov 192.168.0.0/16(rw)_
```

Рис. 22: Обновление exports для пользователя

Применение изменений экспорта

Применяю изменения конфигурации NFS без перезагрузки службы, используя команду `exportfs -r`.

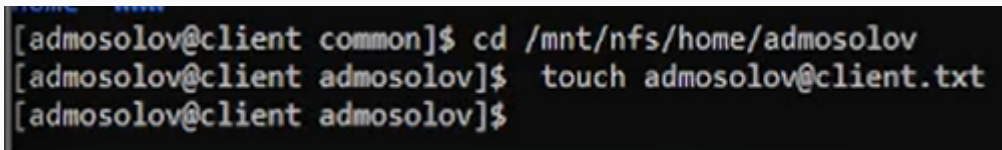
A terminal window with a black background and light blue text. It shows three lines of commands being executed from the user 'admosolov' on the host 'server' in the 'common' directory. The first line is 'sudo nano /etc/exports', the second is 'sudo exportfs -r', and the third is an empty prompt.

```
[admosolov@server common]$ sudo nano /etc/exports  
[admosolov@server common]$ sudo exportfs -r  
[admosolov@server common]$
```

Рис. 23: Применение изменений экспорта

Создание файла на клиенте

Для проверки прав на запись перехожу на клиенте в личный каталог пользователя и создаю там тестовый файл `admosolov@client.txt`.

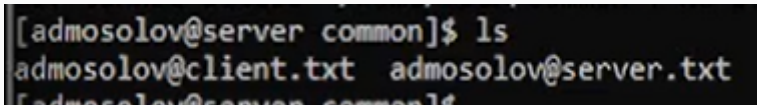
A terminal window with a black background and yellow text. It shows three lines of commands and their outputs. The first line shows a directory change to /mnt/nfs/home/admosolov. The second line shows the creation of a file named admosolov@client.txt using the touch command. The third line shows the current directory after the command.

```
[admosolov@client common]$ cd /mnt/nfs/home/admosolov
[admosolov@client admosolov]$ touch admosolov@client.txt
[admosolov@client admosolov]$
```

Рис. 24: Создание файла на клиенте

Проверка файла на сервере

Возвращаюсь на сервер и проверяю содержимое исходного каталога ~/common. Файл, созданный на клиенте, успешно появился, что подтверждает корректную работу rw-доступа.

A terminal window with a black background and yellow text. The prompt is [admosolov@server common]\$ and the command is ls. The output shows two files: admosolov@client.txt and admosolov@server.txt.

```
[admosolov@server common]$ ls
admosolov@client.txt  admosolov@server.txt
[admosolov@server common]$
```

Рис. 25: Проверка файла на сервере

Подготовка скрипта сервера

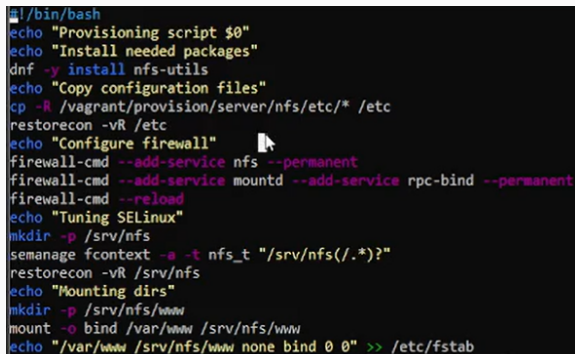
Начинаю этап автоматизации. Создаю на сервере каталог для хранения конфигурационных файлов provisioning-скрипта, копирую туда текущий `/etc/exports` и создаю скрипт `nfs.sh`.

```
[admosolov@server common]$ cd /vagrant/provision/server
[admosolov@server server]$ mkdir-p /vagrant/provision/server/nfs/etc
bash: mkdir-p: command not found...
[admosolov@server server]$ mkdir -p /vagrant/provision/server/nfs/etc
[admosolov@server server]$ cp -R /etc/exports /vagrant/provision/server/nfs/etc/
[admosolov@server server]$ cd /vagrant/provision/server
[admosolov@server server]$ touch nfs.sh
[admosolov@server server]$ chmod +x nfs.sh
```

Рис. 26: Подготовка скрипта сервера

Скрипт nfs.sh для сервера

Наполняю скрипт `nfs.sh` командами, которые я выполнял вручную: установка пакетов, копирование конфигов, настройка `firewall`, `SELinux`, создание каталогов, монтирование и обновление `fstab`.

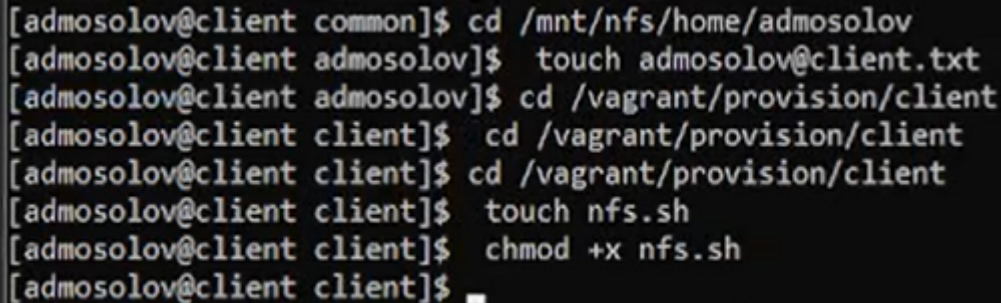
A screenshot of a terminal window with a black background and white text. The terminal shows the execution of a script named `nfs.sh`. The script starts with a prompt `#!/bin/bash` and then prints several status messages in yellow: `Provisioning script $0`, `Install needed packages`, `Copy configuration files`, `Configure firewall`, `Tuning SELinux`, and `Mounting dirs`. The actual commands being run are in white: `dnf -y install nfs-utils`, `cp -R /vagrant/provision/server/nfs/etc/* /etc`, `restorecon -vR /etc`, `firewall-cmd --add-service nfs --permanent`, `firewall-cmd --add-service mountd --add-service rpc-bind --permanent`, `firewall-cmd --reload`, `mkdir -p /srv/nfs`, `semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"`, `restorecon -vR /srv/nfs`, `mkdir -p /srv/nfs/www`, `mount -o bind /var/www /srv/nfs/www`, and `echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab`. A mouse cursor is visible over the `firewall-cmd` command line.

```
#!/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
```

Рис. 27: Скрипт `nfs.sh` для сервера

Подготовка скрипта клиента

Аналогичные действия выполняю для клиента: создаю структуру каталогов в `/vagrant/provision/client` и создаю исполняемый скрипт настройки.

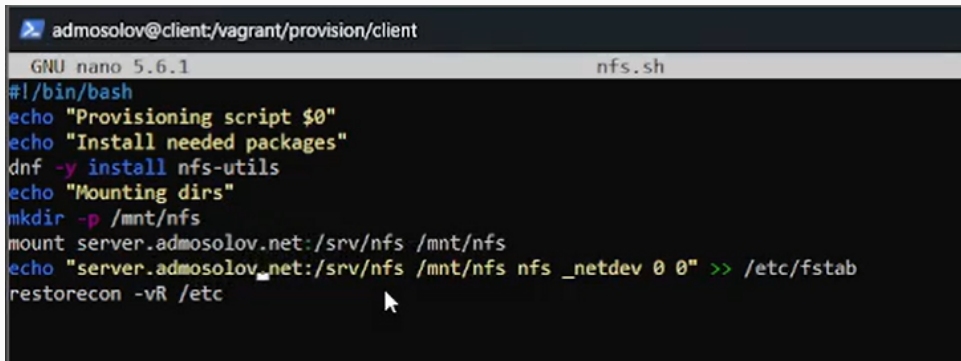
A terminal window with a black background and yellow text. The prompt is [admosolov@client common]. The user enters 'cd /mnt/nfs/home/admosolov', then 'touch admosolov@client.txt'. The prompt changes to [admosolov@client admosolov]. The user enters 'cd /vagrant/provision/client', and the prompt changes to [admosolov@client client]. The user then enters 'cd /vagrant/provision/client' again, 'touch nfs.sh', 'chmod +x nfs.sh', and finally a blank line, resulting in a prompt of [admosolov@client client]\$ followed by a cursor.

```
[admosolov@client common]$ cd /mnt/nfs/home/admosolov
[admosolov@client admosolov]$ touch admosolov@client.txt
[admosolov@client admosolov]$ cd /vagrant/provision/client
[admosolov@client client]$ cd /vagrant/provision/client
[admosolov@client client]$ touch nfs.sh
[admosolov@client client]$ chmod +x nfs.sh
[admosolov@client client]$
```

Рис. 28: Подготовка скрипта клиента

Скрипт nfs.sh для клиента

В скрипт клиента добавляю команды установки `nfs-utils`, создания точки монтирования, монтирования ресурса и добавления записи в `/etc/fstab` для авто-монтирования.



```
admosolov@client:/vagrant/provision/client
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.admosolov.net:/srv/nfs /mnt/nfs
echo "server.admosolov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Рис. 29: Скрипт `nfs.sh` для клиента

Конфигурация Vagrantfile для сервера

В завершение я настраиваю Vagrantfile. Добавляю блок provisioning для сервера, указывая путь к созданному shell-скрипту.

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

Рис. 30: Конфигурация Vagrantfile для сервера

Конфигурация Vagrantfile для клиента

Также добавляю блок provisioning для клиента, чтобы он автоматически настраивался при развертывании виртуальной машины.

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

Рис. 31: Конфигурация Vagrantfile для клиента

Во время выполнения данной лабораторной работы я приобрел практические навыки настройки сетевой файловой системы NFS. Я научился конфигурировать сервер NFSv4, управлять экспортом каталогов, настраивать брандмауэр и SELinux для корректной работы сервиса. Также я освоил методы монтирования удаленных ресурсов на клиенте, использование `bind-mount` для проброса директорий и автоматизацию процесса настройки с помощью `shell`-скриптов в среде Vagrant.