Лабораторная работа №13

Настройка NFS

Лисовская Арина Валерьевна

Содержание

Список иллюстраций

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Задание

1. Установить и настроить сервер NFSv4.
2. Подмонтировать удалённый ресурс на клиенте.
3. Подключить каталог с контентом веб-сервера к дереву NFS.
4. Подключить каталог для удалённой работы пользователя к дереву NFS.
5. Написать скрипты для Vagrant, фиксирующие действия по установке и настройке.

3 Выполнение лабораторной работы

Начинаю работу с установки необходимого программного обеспечения на клиентской машине. Использую команду `dnf -y install nfs-utils` для установки утилит NFS, чтобы иметь возможность монтировать удалённые ресурсы (рис. 3.1).

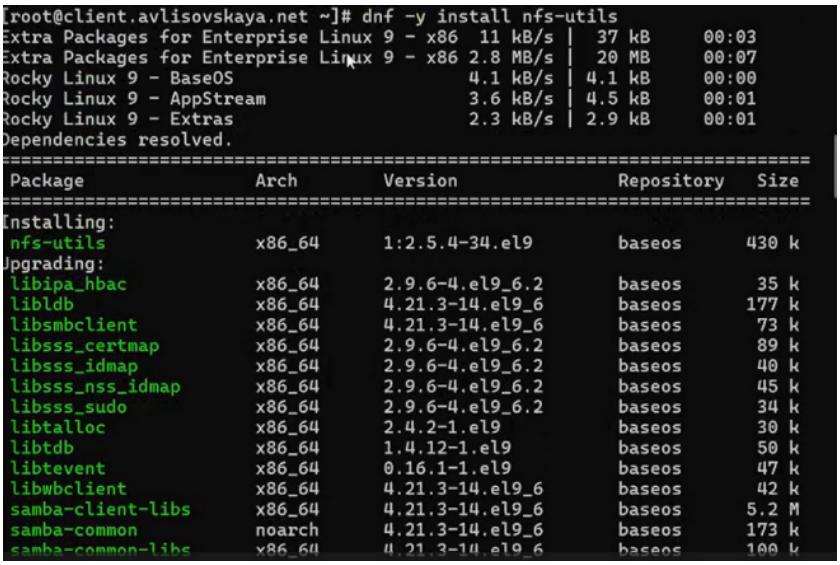

Рис. 3.1: Установка nfs-utils на клиенте

Перехожу к настройке сервера. Создаю основной каталог, который будет служить корнем дерева NFS, с помощью команды `mkdir -p /srv/nfs` (рис. 3.2).

Рис. 3.2: Создание корневого каталога NFS на сервере

Для экспорта созданного каталога редактирую конфигурационный файл

/etc/exports. Добавляю строку /srv/nfs *(ro), разрешающую всем клиентам монтировать этот ресурс в режиме «только чтение» (рис. 3.3).

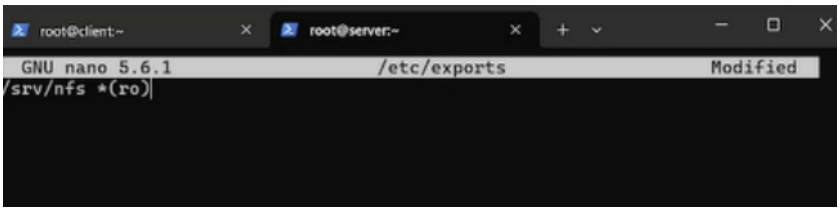

Рис. 3.3: Редактирование /etc/exports

Настраиваю контекст безопасности SELinux, чтобы сервер мог корректно предоставлять доступ к файлам. Выполняю команды semanage fcontext и restorecon для каталога /srv/nfs (рис. 3.4).

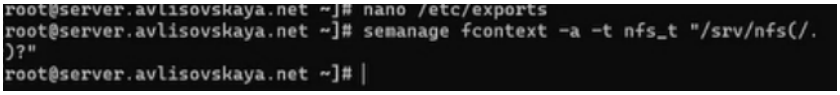

Рис. 3.4: Настройка SELinux

Запускаю службу NFS-сервера и добавляю её в автозагрузку. Также настраиваю межсетевой экран (firewall), разрешая работу службы nfs (рис. 3.5).

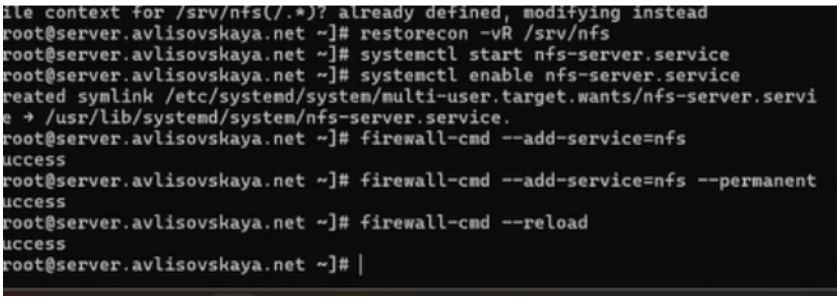

Рис. 3.5: Запуск служб и настройка Firewall

Убеждаюсь, что на клиенте установлены утилиты NFS (повторная проверка/установка для корректности выполнения шагов) (рис. 3.6).

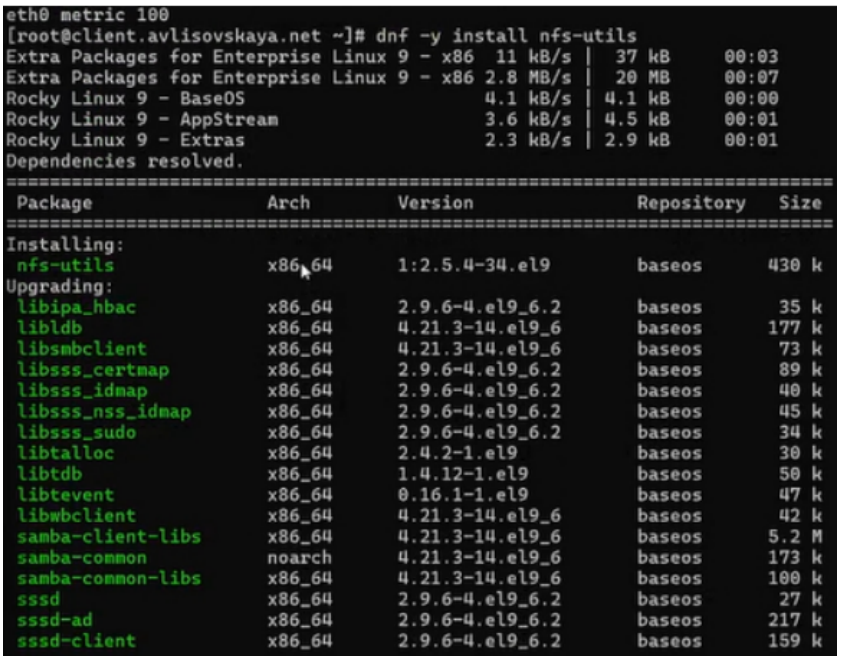

Рис. 3.6: Повторная проверка пакетов на клиенте

Пробую просмотреть список экспортируемых ресурсов с клиента командой `showmount -e`. Получаю ошибку `clnt_create: RPC: Unable to receive`, что свидетельствует о блокировке соединений межсетевым экраном сервера (рис. 3.7).

Рис. 3.7: Ошибка при выполнении showmount

Для диагностики проблемы временно останавливаю службу `firewalld` на сервере (рис. 3.8).

Рис. 3.8: Остановка Firewall на сервере

Повторяю команду `showmount -e` на клиенте. Теперь список экспортируемых

ресурсов отображается корректно, что подтверждает проблему в настройках фаервола (рис. 3.9).

Рис. 3.9: Успешный вывод showmount

Включаю фаервол обратно и анализирую используемые порты. С помощью команды `lsof | grep TCP` просматриваю запущенные TCP-службы (рис. 3.10).

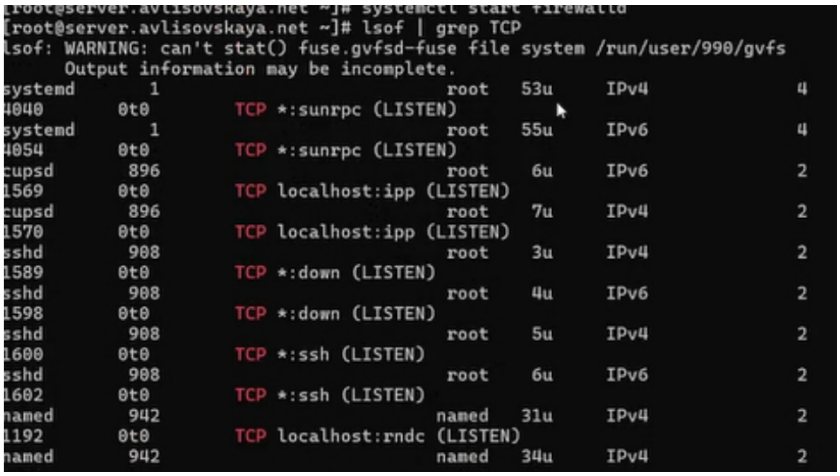

Рис. 3.10: Проверка TCP сокетов

Аналогично проверяю UDP-сокеты командой `lsof | grep UDP`, чтобы понять, какие еще службы требуют разрешения в межсетевом экране (рис. 3.11).

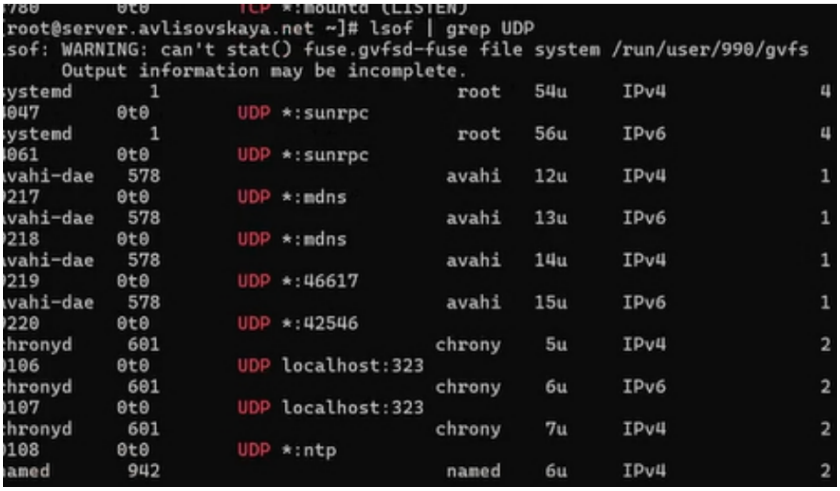

Рис. 3.11: Проверка UDP сокетов

Добавляю разрешающие правила в `firewalld` для служб `mountd` и `rpc-bind`, которые необходимы для корректной работы NFS и RPC-запросов (рис. 3.12).

Рис. 3.12: Дополнительная настройка Firewall

После корректной настройки фаервола снова проверяю доступность ресурсов

с клиента. Команда showmount отрабатывает успешно (рис. 3.13).

Рис. 3.13: Проверка доступа после настройки Firewall

На клиенте создаю точку монтирования /mnt/nfs и монтирую удаленный ресурс сервера командой mount (рис. 3.14).

Рис. 3.14: Монтирование ресурса на клиенте

Проверяю успешность операции с помощью команды mount. Вижу, что ресурс сервера успешно смонтирован как тип nfs4 (рис. 3.15).

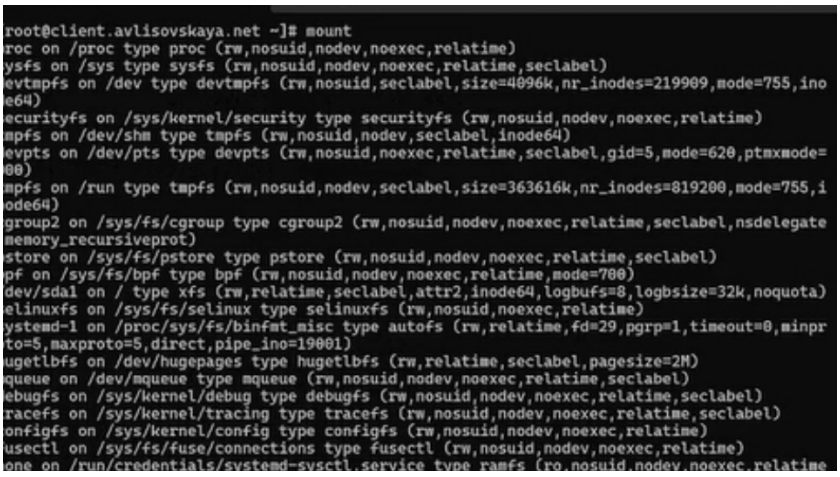

Рис. 3.15: Проверка смонтированных файловых систем

Для автоматического монтирования ресурса при загрузке системы добавляю соответствующую запись в файл /etc/fstab на клиенте, используя опцию _netdev (рис. 3.16).

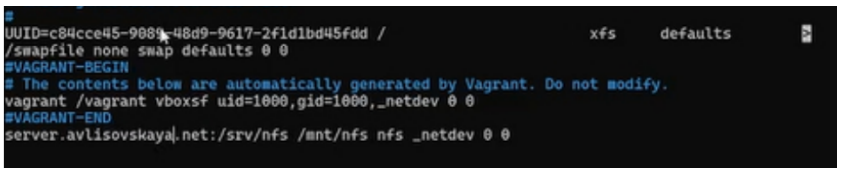

Рис. 3.16: Редактирование /etc/fstab на клиенте

Проверяю статус службы `remote-fs.target`, отвечающей за монтирование удаленных файловых систем (рис. 3.17).

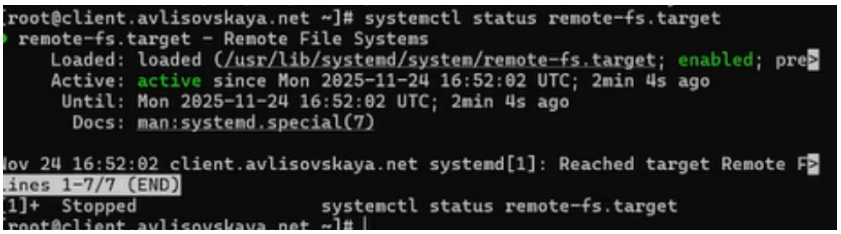

Рис. 3.17: Проверка статуса remote-fs.target

Перехожу к настройке экспорта каталога веб-сервера. На сервере создаю каталог `/srv/nfs/www` и выполняю bind-монтирование реального каталога `/var/www` в дерево NFS (рис. 3.18).

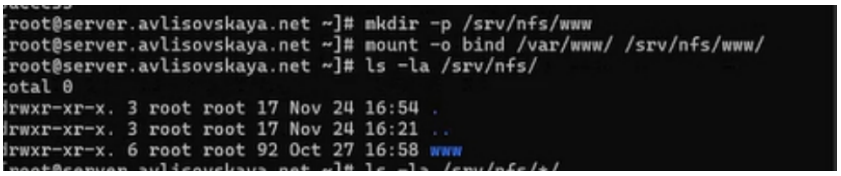

Рис. 3.18: Монтирование каталога www на сервере

Проверяю на клиенте содержимое смонтированного ресурса `/mnt/nfs`. Появился каталог `www`, что означает успешное отображение изменений (рис. 3.19).

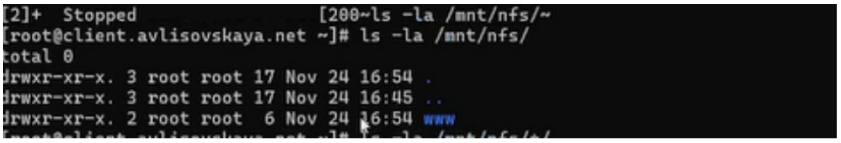

Рис. 3.19: Проверка видимости каталога www на клиенте

Редактирую `/etc/exports` на сервере, добавляя экспорт каталога `/srv/nfs/www` для локальной сети с правами на чтение и запись (рис. 3.20).

Рис. 3.20: Экспорт каталога www

Чтобы bind-монтирование сохранялось после перезагрузки, добавляю соответствующую запись в файл `/etc/fstab` на сервере (рис. 3.21).

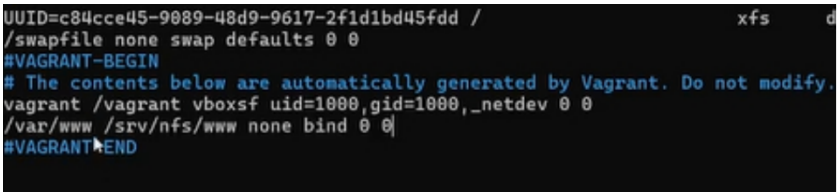

Рис. 3.21: Фиксация bind-монтирования в fstab

Перехожу к настройке пользовательских каталогов. Создаю директорию пользователя и тестовый файл в ней (рис. 3.22).

Рис. 3.22: Создание каталога пользователя

При попытке выполнить действия с пользовательскими каталогами возникла ошибка, связанная с отсутствием точки монтирования или применением настроек (рис. 3.23).

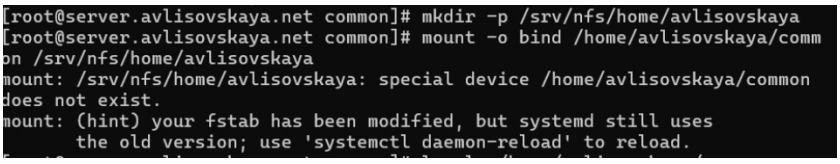

Рис. 3.23: Ошибка монтирования пользовательского каталога

Добавляю экспорт домашнего каталога пользователя в `/etc/exports`, предоставляя права доступа для локальной сети (рис. 3.24).

Рис. 3.24: Экспорт домашнего каталога

Проверяю на клиенте содержимое `/mnt/nfs`. Теперь там отображаются и `www`, и `home` (рис. 3.25).

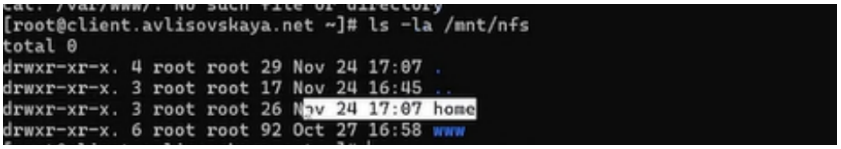

Рис. 3.25: Проверка ресурсов на клиенте

Захожу в смонтированный каталог пользователя на клиенте, создаю файл и проверяю права доступа. Файлы успешно создаются и читаются (рис. 3.26).

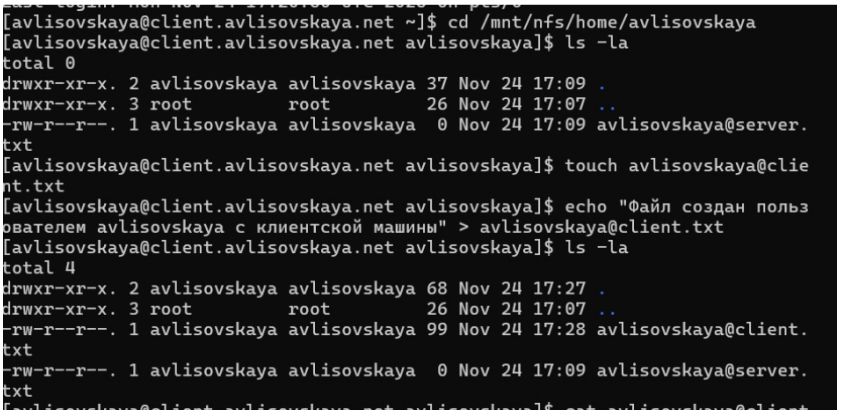

Рис. 3.26: Проверка прав доступа пользователя

Приступаю к автоматизации. На сервере подготавливаю структуру каталогов в `/vagrant` для создания скрипта провижининга (рис. 3.27).

```
[avlisovskaya@server.avlisovskaya.net ~]$ cd /vagrant/provision/server
[avlisovskaya@server.avlisovskaya.net server]$ mkdir -p /vagrant/provision/s
erver/nfs/etc
[avlisovskaya@server.avlisovskaya.net server]$ cp -R /etc/exports /vagrant/p
rovision/server/nfs/etc/
[avlisovskaya@server.avlisovskaya.net server]$ cd /vagrant/provision/server
[avlisovskaya@server.avlisovskaya.net server]$ touch nfs.sh
[avlisovskaya@server.avlisovskaya.net server]$ chmod +x nfs.sh
[avlisovskaya@server.avlisovskaya.net server]$ nano  nfs.sh
```

Рис. 3.27: Подготовка каталогов для скриптов

Создаю скрипт `nfs.sh` для сервера, который устанавливает пакеты, копирует конфиги, настраивает фаервол, SELinux и монтирует каталоги (рис. 3.28).

```
 GNU nano 5.6.1                        nfs.sh                              Modified
#!/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. 3.28: Скрипт настройки сервера nfs.sh

Аналогично создаю структуру и файл скрипта для клиентской машины (рис. 3.29).

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

Рис. 3.29: Подготовка скрипта для клиента

Наполняю скрипт `nfs.sh` для клиента командами установки пакетов и авто-

матического монтирования ресурса при развертывании (рис. 3.30).

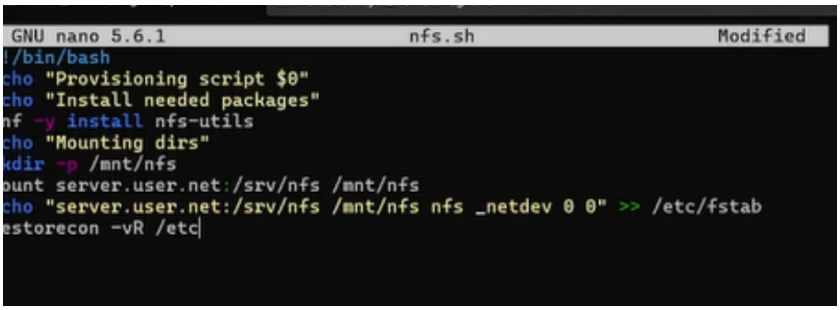

Рис. 3.30: Скрипт настройки клиента nfs.sh

Вношу изменения в `Vagrantfile`, добавляя секцию provision для запуска скрипта настройки на сервере (рис. 3.31).

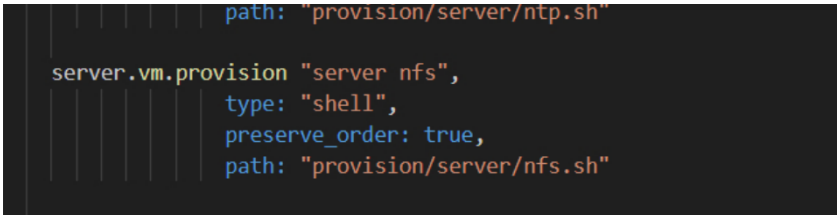

Рис. 3.31: Настройка provision для сервера в Vagrantfile

Добавляю аналогичную секцию provision для клиента в `Vagrantfile` (рис. 3.32).

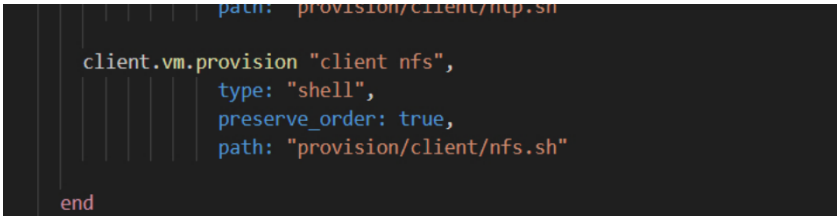

Рис. 3.32: Настройка provision для клиента в Vagrantfile

4 Выводы

В ходе выполнения лабораторной работы я приобрела навыки настройки NFS-сервера в Linux. Я научилась экспортировать файловые системы, управлять правами доступа через `/etc/exports` и настройки firewall, а также монтировать удаленные ресурсы на клиенте. Кроме того, была реализована автоматизация настройки стенда с использованием Vagrant и bash-скриптов.

5 Ответы на контрольные вопросы

1. **Как называется файл конфигурации, содержащий общие ресурсы NFS?** Файл конфигурации называется `/etc/exports`. В нем описываются экспортируемые каталоги, разрешенные клиенты и параметры доступа.

2. **Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS?** Для полноценной работы NFSv4 и сопутствующих служб необходимо открыть порты для служб:

   - `nfs` (TCP/2049)
   - `mountd` (обычно TCP/UDP 20048)
   - `rpc-bind` (TCP/UDP 111)

3. **Какую опцию следует использовать в `/etc/fstab`, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке?** Следует использовать опцию `_netdev`. Она указывает системе, что данное устройство требует сети, и предотвращает попытки монтирования до того, как сетевые интерфейсы будут подняты, что позволяет избежать зависания системы при загрузке.