

Отчёт по лабораторной работе №13

Дисциплина: Администрирование сетевых подсистем

Ибрахим Мохсейн Алькамаль

Содержание

1 Цель работы	6
2 Выполнение лабораторной работы	7
2.1 Настройка сервера NFSv4	7
2.2 Монтирование NFS на клиенте	12
2.3 Подключение каталогов к дереву NFS	14
2.4 Подключение каталогов для работы пользователей	17
2.5 Внесение изменений в настройки внутреннего	20
3 Выводы	23
4 Контрольные вопросы	24

Список иллюстраций

2.1	Процесс установки пакета nfs-utils с помощью dnf	8
2.2	Настройка файла /etc(exports с экспортом каталога /srv/nfs	8
2.3	Назначение SELinux-контекста nfs_t для каталога /srv/nfs	9
2.4	Применение restorecon и запуск службы nfs-server	9
2.5	Настройка firewalld для разрешения сервиса NFS	9
2.6	Установка пакета nfs-utils на клиенте	10
2.7	Ошибка RPC при выполнении showmount на клиенте	10
2.8	Остановка службы firewalld на сервере	10
2.9	Успешный вывод showmount с отображением экспорта /srv/nfs	11
2.10	Повторный запуск службы firewalld на сервере	11
2.11	Вывод lsof для TCP-соединений служб RPC и NFS	11
2.12	Вывод lsof для UDP-соединений служб RPC и NFS	11
2.13	Добавление служб mountd и rpc-bind в firewalld	12
2.14	Проверка доступа к экспортируемому ресурсу через showmount	12
2.15	Создание каталога и монтирование NFS-ресурса на клиенте	12
2.16	Вывод mount с отображением подключённого NFSv4-ресурса	13
2.17	Добавление записи автоматического монтирования в /etc/fstab	13
2.18	Проверка состояния remote-fs.target	14
2.19	Подтверждение автоматического монтирования NFS после перезагрузки	14
2.20	Создание каталога /srv/nfs/www и bind-монтирование /var/www	15
2.21	Просмотр каталога /mnt/nfs на клиенте	15
2.22	Добавление экспорта каталога /srv/nfs/www в /etc(exports	15
2.23	Повторная инициализация экспорта NFS с помощью exportfs -r	16
2.24	Просмотр содержимого /mnt/nfs/www на клиенте	16
2.25	Добавление bind-монтирования в /etc/fstab	16
2.26	Повторный экспорт каталогов после изменения /etc/fstab	17
2.27	Итоговая проверка содержимого /mnt/nfs/www на клиенте	17
2.28	Создание каталога common с правами 700 и файла пользователя	17
2.29	Добавление каталога пользователя в /etc(exports	18
2.30	Добавление bind-монтирования каталога пользователя в /etc/fstab	18
2.31	Повторный экспорт каталогов NFS	19
2.32	Проверка каталога /mnt/nfs на клиенте	19
2.33	Создание файла пользователем alkamal на клиенте через NFS	19
2.34	Проверка появления файла клиента на сервере	19
2.35	Отказ в доступе под пользователем root на клиенте	20

2.36	Копирование файла /etc(exports в каталог provision/server/nfs/etc	20
2.37	Содержимое скрипта nfs.sh для виртуальной машины server	21
2.38	Создание файла nfs.sh в каталоге provision/client	21
2.39	Содержимое скрипта nfs.sh для виртуальной машины client	22

Список таблиц

1 Цель работы

Приобретение практических навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Выполнение лабораторной работы

2.1 Настройка сервера NFSv4

Выполнена установка пакета `nfs-utils` с использованием менеджера пакетов `dnf`, что обеспечивает наличие необходимых утилит и служб для функционирования сервера NFSv4 (рис. 2.1).

```

server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 11 01:53
root@server:/vagrant/provision/server

total download size: 757 k
Installed size: 1.9 M
Downloading Packages:
(1/7): libnfsidmap-2.5.4-38.el9.x86_64.rpm           261 kB/s | 60 kB   00:00
(2/7): libev-4.33-6.el9.x86_64.rpm                   182 kB/s | 51 kB   00:00
(3/7): gssproxy-0.8.4-7.el9.x86_64.rpm             355 kB/s | 108 kB  00:00
(4/7): libverto-libev-0.3.2-3.el9.x86_64.rpm       52 kB/s | 13 kB   00:00
(5/7): rpcbind-1.2.6-7.el9.x86_64.rpm              207 kB/s | 56 kB   00:00
(6/7): nfs-utils-1.2.5.4-38.el9.x86_64.rpm        681 kB/s | 432 kB  00:00
(7/7): sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64.rpm   81 kB/s | 37 kB   00:00
Total                                         628 kB/s | 757 kB  00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Installing : libnfsidmap-1:2.5.4-38.el9.x86_64          1/7
  Running scriptlet: rpcbind-1.2.6-7.el9.x86_64          2/7
  Installing : rpcbind-1.2.6-7.el9.x86_64               2/7
  Running scriptlet: rpcbind-1.2.6-7.el9.x86_64          2/7
  created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
  created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.

  Installing : libev-4.33-6.el9.x86_64                  3/7
  Installing : libverto-libev-0.3.2-3.el9.x86_64         4/7
  Installing : gssproxy-0.8.4-7.el9.x86_64              5/7
  Running scriptlet: gssproxy-0.8.4-7.el9.x86_64          5/7
  Running scriptlet: nfs-utils-1:2.5.4-38.el9.x86_64      6/7
  Installing : nfs-utils-1:2.5.4-38.el9.x86_64            6/7
  Running scriptlet: nfs-utils-1:2.5.4-38.el9.x86_64      6/7
  Installing : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64     7/7
  Running scriptlet: sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64 7/7
  Verifying   : gssproxy-0.8.4-7.el9.x86_64              1/7
  Verifying   : libev-4.33-6.el9.x86_64                  2/7
  Verifying   : libnfsidmap-1:2.5.4-38.el9.x86_64          3/7
  Verifying   : libverto-libev-0.3.2-3.el9.x86_64          4/7
  Verifying   : nfs-utils-1:2.5.4-38.el9.x86_64            5/7
  Verifying   : rpcbind-1.2.6-7.el9.x86_64                6/7
  Verifying   : sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64      7/7

Installed:
  gssproxy-0.8.4-7.el9.x86_64                         libev-4.33-6.el9.x86_64
  libnfsidmap-1:2.5.4-38.el9.x86_64                   libverto-libev-0.3.2-3.el9.x86_64
  nfs-utils-1:2.5.4-38.el9.x86_64                      rpcbind-1.2.6-7.el9.x86_64
  sssd-nfs-idmap-2.9.7-4.el9_7.1.x86_64

complete!

```

Рисунок 2.1: Процесс установки пакета nfs-utils с помощью dnf

Создан каталог `/srv/nfs`, предназначенный для использования в качестве корневого каталога экспортируемого дерева NFS. В файле `/etc/exports` добавлена строка `/srv/nfs * (ro)`, определяющая экспорт каталога с правами только на чтение для всех узлов сети (рис. 2.2).

```

server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 11 01:56
root@server:/vagrant/provision/server

GNU nano 5.6.1
/etc/exports
/srv/nfs *(ro)

```

Рисунок 2.2: Настройка файла `/etc/exports` с экспортом каталога `/srv/nfs`

Для каталога `/srv/nfs` задан контекст безопасности SELinux типа `nfs_t` с помощью команды `semanage fcontext`, что обеспечивает корректную работу

службы NFS в рамках политики безопасности (рис. 2.3).

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

Рисунок 2.3: Назначение SELinux-контекста nfs_t для каталога /srv/nfs

Изменённый контекст SELinux применён к файловой системе командой `restorecon -vR /srv/nfs`, после чего выполнен запуск и добавление службы `nfs-server.service` в автозагрузку средствами `systemctl` (рис. 2.4).

```
[root@server.alkamal.net server]# 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.alkamal.net server]# systemctl start nfs-server.service  
[root@server.alkamal.net server]# systemctl enable nfs-server.service
```

Рисунок 2.4: Применение restorecon и запуск службы nfs-server

Настроен межсетевой экран: добавлено разрешение сервиса `nfs` и выполнена перезагрузка конфигурации `firewalld`, что обеспечивает доступ к серверу NFS через соответствующие сетевые порты (рис. 2.5).

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

Рисунок 2.5: Настройка firewalld для разрешения сервиса NFS

На клиенте выполнена установка пакета `nfs-utils` с использованием `dnf`, что обеспечивает наличие утилит для работы с NFS-ресурсами, включая `showmount` (рис. 2.6).

```

[root@client.alkamal.net client]# dnf -y install nfs-utils
Last metadata expiration check: 0:04:03 ago on Wed 11 Feb 2026 01:53:28 AM UTC.
Dependencies resolved.
=====
Package           Architecture   Version      Repository  Size
=====
Installing:
  nfs-utils       x86_64        1:2.5.4-38.el9    baseos     432 k
Installing dependencies:
  gssproxy        x86_64        0.8.4-7.el9      baseos     108 k
  libev           x86_64        4.33-6.el9      baseos      51 k
  libnfsidmap     x86_64        1:2.5.4-38.el9    baseos      60 k
  libertto-libev  x86_64        0.3.2-3.el9      baseos      13 k
  rpcbind         x86_64        1.2.6-7.el9      baseos      56 k
  sssd-nfs-idmap  x86_64        2.9.7-4.el9_7.1  baseos      37 k

Transaction Summary
Install 7 Packages

Total download size: 757 k
Installed size: 1.9 M
Downloading Packages:

```

Рисунок 2.6: Установка пакета nfs-utils на клиенте

После установки выполнена команда `showmount -e server.alkamal.net`. В результате получено сообщение `clnt_create: RPC: Unable to receive`, что указывает на невозможность получения ответа от сервера по протоколу RPC при активном межсетевом экране (рис. 2.7).

```

complete!
[root@client.alkamal.net client]# showmount -e server.alkamal.net
clnt_create: RPC: Unable to receive
[root@client.alkamal.net client]#

```

Рисунок 2.7: Ошибка RPC при выполнении showmount на клиенте

На сервере выполнена остановка службы межсетевого экрана командой `systemctl stop firewalld.service`, что временно снимает сетевые ограничения (рис. 2.8).

```

[root@server.alkamal.net server]# systemctl stop firewalld.service
[root@server.alkamal.net server]#

```

Рисунок 2.8: Остановка службы firewalld на сервере

После отключения `firewalld` повторный запуск команды `showmount -e server.alkamal.net` на клиенте приводит к успешному получению списка экспортируемых ресурсов. Отображается каталог `/srv/nfs *`, что подтверждает доступность NFS-экспорта (рис. 2.9).

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

Рисунок 2.9: Успешный вывод showmount с отображением экспорта /srv/nfs

На сервере служба межсетевого экрана вновь запущена командой `systemctl start firewalld`, что восстанавливает стандартные правила фильтрации сетевого трафика (рис. 2.10).

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

Рисунок 2.10: Повторный запуск службы firewalld на сервере

На сервере выполнен анализ активных служб, задействованных при удалённом мониторинге, с использованием команд `lsof | grep TCP` и `lsof | grep UDP`. В выводе отображаются процессы `rpcbind`, `rpc.statd`, `rpc.mountd`, работающие по протоколам TCP и UDP в режимах IPv4 и IPv6, что подтверждает использование RPC-служб при обслуживании NFS-запросов (рис. 2.11, рис. 2.12).

```
[root@server.alkamal.net server]# lsof -i :11877
sunrpc (LISTEN)
rpc.statd 11877
50491 (LISTEN)
rpc.statd 11877
47419 (LISTEN)
rpc.mount 11885
mountd (LISTEN)
rpc.mount 11885
mountd (LISTEN)
[root@server.alkamal.net server]#
```

Рисунок 2.11: Вывод lsof для TCP-соединений служб RPC и NFS

```
[root@server.alkamal.net server]# lsof -i :11876
rpcbind 11876
sunrpc
rpcbind 11876
sunrpc
rpc.bind 11877
rpc.statd 11877
34159
rpc.statd 11877
48036
rpc.statd 11877
calhost:659
rpc.mount 11885
mountd
rpc.mount 11885
mountd
[root@server.alkamal.net server]#
```

Рисунок 2.12: Вывод lsof для UDP-соединений служб RPC и NFS

В конфигурацию межсетевого экрана добавлены службы `mountd` и `rpc-bind` командами `firewall-cmd --add-service=mountd --add-service=rpc-bind`

и с параметром `--permanent`, после чего выполнена перезагрузка конфигурации `firewalld`. Это обеспечивает разрешение необходимых RPC-портов для корректной работы NFS через межсетевой экран (рис. 2.13).

```
[root@server.alkamal.net server]# firewall-cmd --add-service=mountd --add-service=rpc-bind  
success  
[root@server.alkamal.net server]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent  
success  
[root@server.alkamal.net server]# firewall-cmd --reload  
success  
[root@server.alkamal.net server]#
```

Рисунок 2.13: Добавление служб mountd и rpc-bind в firewalld

На клиенте повторно выполнена команда `showmount -e server.alkamal.net`. В результате успешно получен список экспортируемых ресурсов, включая каталог `/srv/nfs *`, что подтверждает корректную настройку служб и правил межсетевого экрана (рис. 2.14).

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

Рисунок 2.14: Проверка доступа к экспортируемому ресурсу через showmount

2.2 Монтирование NFS на клиенте

На клиенте создан каталог `/mnt/nfs`, после чего выполнено монтирование удалённого ресурса командой `mount server.alkamal.net:/srv/nfs /mnt/nfs`. Это приводит к подключению экспортируемого каталога сервера к локальной точке монтирования клиента (рис. 2.15).

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

Рисунок 2.15: Создание каталога и монтирование NFS-ресурса на клиенте

Проверка командой `mount` подтверждает наличие записи вида `server.alkamal.net:/srv/nfs /mnt/nfs type nfs4`. Это означает, что ресурс подключён по протоколу NFSv4, используется транспорт TCP (`proto=tcp`), режим жёсткого монтирования

(hard), а также параметры обмена данными (`rsize`, `wsize`) и механизм аутентификации `sec=sys` (рис. 2.16).

```
[root@client.alkamal.net client]# mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=89962,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=156656k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recur
siveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/dev/sdal on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxprot
o=5,direct,pipe_ino=20161)
hugeglbfs on /dev/hugepages type hugeglbfs (rw,relatime,seclabel,pagesize=2M)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,ngdev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,ngdev,noexec,relatime,seclabel)
mqqueue on /dev/mqueue type mqqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,
seclabel,mode=700)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,secl
abel,mode=700)
none on /vagrant type vboxsf (rw,ngdev,relatime,iocharset=utf8,uid=1000,gid=1000)
vagrant on /vagrant type vboxsf (rw,ngdev,relatime,iocharset=utf8,uid=1000,gid=1000)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=7828k,nr_inodes=19582,mode=7
00,uid=1001,gid=1001,inode64)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_id=1
001)
portal on /run/user/1001/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.alkamal.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen
=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,addr=192.168.1.1
)
```

Рисунок 2.16: Вывод `mount` с отображением подключённого NFSv4-ресурса

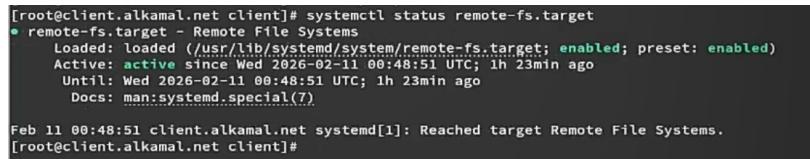
В файл `/etc/fstab` добавлена строка `server.alkamal.net:/srv/nfs /mnt/nfs nfs _netdev 0 0`. Первое поле — удалённый ресурс, второе — точка монтирования, третье — тип файловой системы (`nfs`), параметр `_netdev` указывает на зависимость монтирования от сетевой подсистемы, последние два нуля определяют отсутствие резервного копирования (`dump`) и проверки `fsck` (рис. 2.17).

```
GNU nano 5.6.1                               /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Feb  6 01:52:40 2026
#
# 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=f93095b3-9d30-43f4-b9e7-d0ab51fd6bc3 /xfs      defaults
/swappfile 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.alkamal.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рисунок 2.17: Добавление записи автоматического монтирования в `/etc/fstab`

Проверка состояния цели `remote-fs.target` командой `systemctl status remote-fs.target` показывает статус `active`, что свидетельствует о готовности системы автоматически монтировать удалённые файловые системы при запуске (рис. 2.18).



```
[root@client.alkamal.net client]# 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 2026-02-11 00:48:51 UTC; 1h 23min ago)
     Until: Wed 2026-02-11 00:48:51 UTC; 1h 23min ago
   Docs: man:systemd.special(7)

Feb 11 00:48:51 client.alkamal.net systemd[1]: Reached target Remote File Systems.
[root@client.alkamal.net client]#
```

Рисунок 2.18: Проверка состояния `remote-fs.target`

После перезапуска клиента выполнена проверка содержимого `/mnt/nfs` и повторный вывод `mount | grep nfs`. Отображение строки с типом `nfs4` подтверждает, что удалённый ресурс автоматически подключён при старте операционной системы (рис. 2.19).

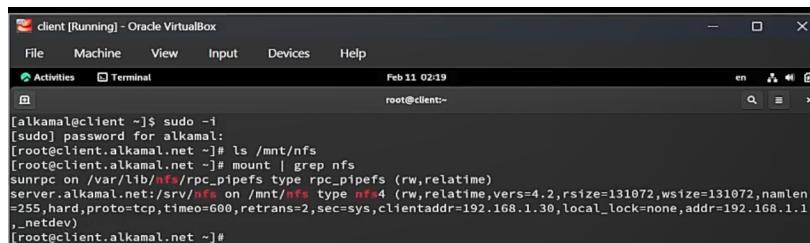


Рисунок 2.19: Подтверждение автоматического монтирования NFS после перезагрузки

2.3 Подключение каталогов к дереву NFS

На сервере создан каталог `/srv/nfs/www`, предназначенный для включения в дерево NFS, после чего выполнено bind-монтирование каталога веб-сервера `/var/www` в созданный каталог командой `mount -o bind /var/www/ /srv/nfs/www/`. Проверка содержимого подтверждает наличие каталогов `cgi-bin` и `html` (рис. 2.20).

```

server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 11 03:06
root@server:~
[alkamal@server.alkamal.net ~]$ sudo -i
[sudo] password for alkamal:
[root@server.alkamal.net ~]# mkdir -p /srv/nfs/www
[root@server.alkamal.net ~]# mount -o bind /var/www/ /srv/nfs/www/
mount: (hint) your fstab has been modified, but systemd still uses
      the old version; use 'systemctl daemon-reload' to reload.
[root@server.alkamal.net ~]# ls /srv/nfs/www/
cgi-bin html
[root@server.alkamal.net ~]#

```

Рисунок 2.20: Создание каталога /srv/nfs/www и bind-монтирование /var/www

На клиенте при просмотре каталога /mnt/nfs отображается подкаталог www, что подтверждает его наличие в экспортируемом дереве NFS (рис. 2.21).

```

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

```

Рисунок 2.21: Просмотр каталога /mnt/nfs на клиенте

В файл /etc(exports на сервере добавлена строка /srv/nfs/www 192.168.0.0/16(rw), что разрешает доступ на чтение и запись к каталогу www для узлов сети 192.168.0.0/16 (рис. 2.22).

```

server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 11 03:07
root@server:~
root@server:/vagrant/provision/server
GNU nano 5.6.1
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)

```

Рисунок 2.22: Добавление экспорта каталога /srv/nfs/www в /etc/exports

Выполнена команда `exportfs -r`, обеспечивающая повторное применение конфигурации экспорта без перезапуска службы NFS (рис. 2.23).

```
[root@server.alkamal.net ~]# nano /etc(exports  
[root@server.alkamal.net ~]# exportfs -r  
[root@server.alkamal.net ~]#
```

Рисунок 2.23: Повторная инициализация экспорта NFS с помощью exportfs -r

На клиенте выполнена проверка каталога `/mnt/nfs/www`, где отображаются каталоги `cgi-bin` и `html`, что подтверждает корректность экспорта и доступность веб-контента через NFS (рис. 2.24).

```
[root@client.alkamal.net ~]# ls /mnt/nfs/www  
cgi-bin  html  
[root@client.alkamal.net ~]#
```

Рисунок 2.24: Просмотр содержимого `/mnt/nfs/www` на клиенте

Для обеспечения постоянного bind-монтирования на сервере в файл `/etc/fstab` добавлена строка `/var/www /srv/nfs/www none bind 0 0`, где первое поле — источник, второе — точка монтирования, тип `none` используется для bind-монтирования, параметр `bind` задаёт тип операции, последние два нуля отключают `dump` и проверку `fsck` (рис. 2.25).

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

Рисунок 2.25: Добавление bind-монтирования в `/etc/fstab`

После повторного выполнения `exportfs -r` конфигурация экспорта актуализирована с учётом постоянного bind-монтирования (рис. 2.26).

```
[root@server.alkamal.net ~]# nano /etc/fstab
[root@server.alkamal.net ~]# exportfs -r
[root@server.alkamal.net ~]#
```

Рисунок 2.26: Повторный экспорт каталогов после изменения /etc/fstab

На клиенте повторная проверка каталога /mnt/nfs/www подтверждает сохранение доступа к каталогам cgi-bin и html, что свидетельствует о корректной настройке подключения каталогов к дереву NFS (рис. 2.27).

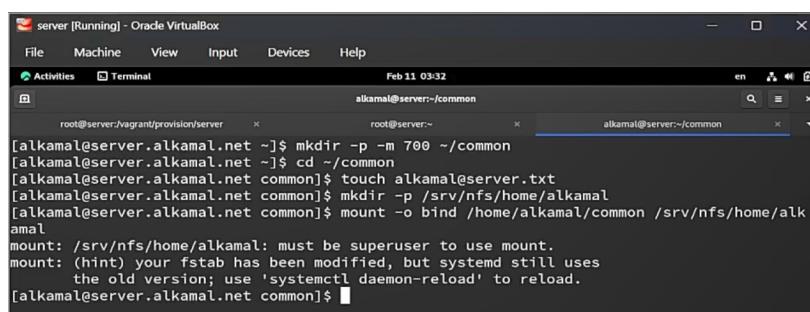
```
[root@client.alkamal.net ~]# ls /mnt/nfs/www
cgi-bin  html
[root@client.alkamal.net ~]#
```

Рисунок 2.27: Итоговая проверка содержимого /mnt/nfs/www на клиенте

2.4 Подключение каталогов для работы

пользователей

На сервере под пользователем alkamal создан каталог ~/common с правами 700, что обеспечивает полный доступ только владельцу. В каталоге создан файл alkamal@server.txt. Права 700 означают rwx-----, то есть доступ запрещён для группы и остальных пользователей (рис. 2.28).



```
server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 11 03:32
root@server:vagrant/provision/server alkamal@server:~/common
[alkamal@server.alkamal.net ~]$ mkdir -p -m 700 ~/common
[alkamal@server.alkamal.net ~]$ cd ~/common
[alkamal@server.alkamal.net common]$ touch alkamal@server.txt
[alkamal@server.alkamal.net common]$ mkdir -p /srv/nfs/home/alkamal
[alkamal@server.alkamal.net common]$ mount -o bind /home/alkamal/common /srv/nfs/home/alkamal
mount: /srv/nfs/home/alkamal: must be superuser to use mount.
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[alkamal@server.alkamal.net common]$
```

Рисунок 2.28: Создание каталога common с правами 700 и файла пользователя

В файл /etc(exports добавлена строка /srv/nfs/home/alkamal

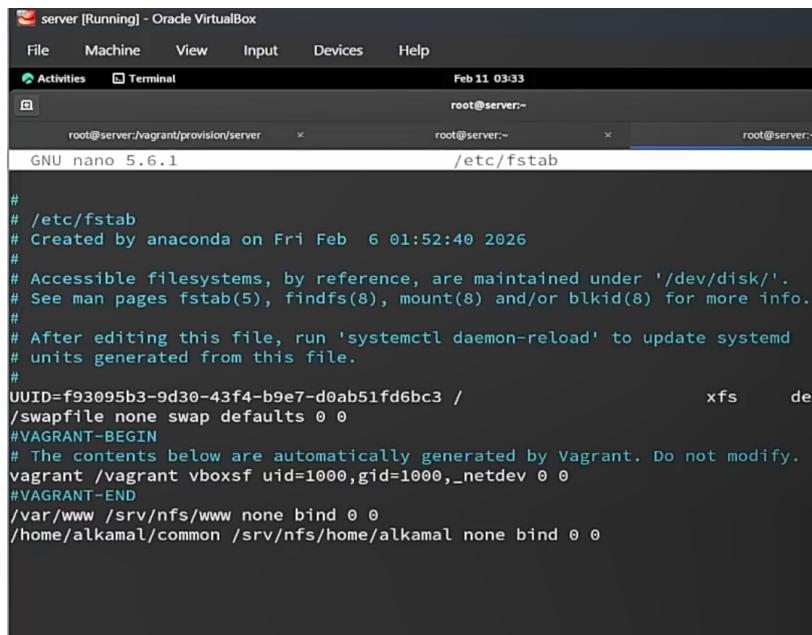
192.168.0.0/16 (rw), что разрешает доступ к каталогу по сети с правами чтения и записи для указанного диапазона адресов (рис. 2.29).



```
GNU nano 5.6.1
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/alkamal 192.168.0.0/16(rw)
```

Рисунок 2.29: Добавление каталога пользователя в /etc/exports

В файл /etc/fstab добавлена запись /home/alkamal/common /srv/nfs/home/alkamal none bind 0 0, где первое поле – источник, второе – точка монтирования в дереве NFS, none – тип для bind-монтирования, bind – параметр операции, последние два нуля отключают dump и проверку fsck (рис. 2.30).



```
# /etc/fstab
# Created by anaconda on Fri Feb  6 01:52:40 2026
#
# 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=f93095b3-9d30-43f4-b9e7-d0ab51fd6bc3 /          xfs      de
/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/alkamal/common /srv/nfs/home/alkamal none bind 0 0
```

Рисунок 2.30: Добавление bind-монтирования каталога пользователя в /etc/fstab

После выполнения `exportfs -r` конфигурация экспорта обновлена без перезапуска службы NFS (рис. 2.31).

```
[root@server.alkamal.net ~]# nano /etc/exports  
[root@server.alkamal.net ~]# exportfs -r  
[root@server.alkamal.net ~]#
```

Рисунок 2.31: Повторный экспорт каталогов NFS

На клиенте при просмотре `/mnt/nfs` отображаются каталоги `home` и `www`, что подтверждает экспорт пользовательского каталога (рис. 2.32).

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

Рисунок 2.32: Проверка каталога `/mnt/nfs` на клиенте

Под пользователем `alkamal` на клиенте выполнен переход в каталог `/mnt/nfs/home/alkamal` и создан файл `alkamal@client.txt`, что подтверждает наличие прав записи через NFS (рис. 2.33).

```
[alkamal@client ~]$ cd /mnt/nfs/home/alkamal  
[alkamal@client alkamal]$ touch alkamal@client.txt  
[alkamal@client alkamal]$
```

Рисунок 2.33: Создание файла пользователем `alkamal` на клиенте через NFS

На сервере в каталоге `/home/alkamal/common` отображаются файлы `alkamal@server.txt` и `alkamal@client.txt`, что подтверждает синхронность данных через bind-мониторинг и экспорт NFS (рис. 2.34).

```
[root@server.alkamal.net ~]# logout  
[alkamal@server.alkamal.net common]$ ls /home/alkamal/common  
alkamal@client.txt alkamal@server.txt
```

Рисунок 2.34: Проверка появления файла клиента на сервере

При попытке доступа к каталогу `/mnt/nfs/home/alkamal` под пользователем `root` на клиенте получено сообщение `Permission denied`, что обусловлено

правами 700, установленными на исходный каталог пользователя на сервере (рис. 2.35).

```
[alkamal@client alkamal]$ sudo -i  
[sudo] password for alkamal:  
[root@client.alkamal.net ~]# cd /mnt/nfs/home/alkamal  
-bash: cd: /mnt/nfs/home/alkamal: Permission denied  
[root@client.alkamal.net ~]#
```

Рисунок 2.35: Отказ в доступе под пользователем root на клиенте

2.5 Внесение изменений в настройки внутреннего

На виртуальной машине **server** выполнен переход в каталог `/vagrant/provision/server`, создан каталог `nfs/etc`, после чего файл `/etc(exports` скопирован во внутреннюю структуру provisioning. Это обеспечивает сохранение конфигурации экспорта для автоматического развёртывания (рис. 2.36).

```
[alkamal@server.alkamal.net common]$ cd /vagrant/provision/server  
[alkamal@server.alkamal.net server]$ mkdir -p /vagrant/provision/server/nfs/etc  
[alkamal@server.alkamal.net server]$ cp -R /etc(exports /vagrant/provision/server/nfs/etc  
/  
[alkamal@server.alkamal.net server]$ cd /vagrant/provision/server  
[alkamal@server.alkamal.net server]$ touch nfs.sh  
[alkamal@server.alkamal.net server]$ chmod +x nfs.sh  
[alkamal@server.alkamal.net server]$ nano nfs.sh
```

Рисунок 2.36: Копирование файла `/etc(exports` в каталог `provision/server/nfs/etc`

В каталоге `/vagrant/provision/server` создан исполняемый файл `nfs.sh`, в котором реализован сценарий автоматической настройки: установка `nfs-utils`, копирование конфигурации в `/etc`, настройка `firewalld`, назначение контекста SELinux `nfs_t`, bind-монтирование каталогов `/var/www` и `/home/alkamal/common`, добавление записей в `/etc/fstab`, а также запуск и включение службы `nfs-server` (рис. 2.37).

```

root@server:/vagrant/provision/server      x          root@server:~          x          alkamal@server:/vagrant/provision/server —n
GNU nano 5.6.1
#!/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 -v /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/alkamal
mkdir -p -m 700 /home/alkamal/common
chown alkamal:alkamal /home/alkamal/common
mount -o bind /home/alkamal/common /srv/nfs/home/alkamal
echo "/home/alkamal/common /srv/nfs/home/alkamal none bind 0 0" >> /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld

```

Рисунок 2.37: Содержимое скрипта nfs.sh для виртуальной машины server

На виртуальной машине **client** выполнен переход в каталог /vagrant/provision/client создан исполняемый файл nfs.sh, предназначенный для автоматической настройки клиента NFS (рис. 2.38).

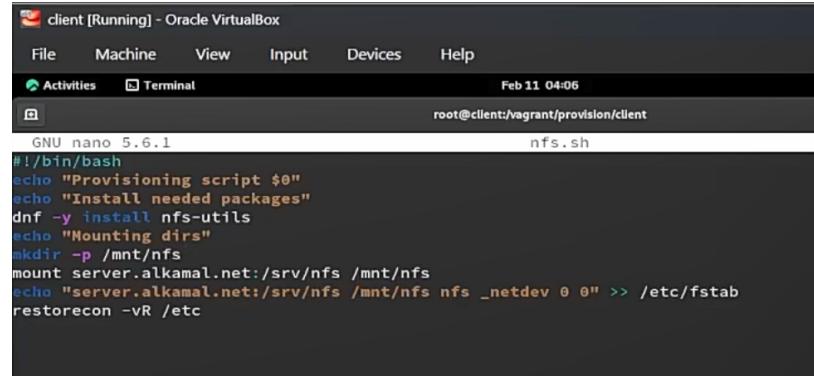
```

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

```

Рисунок 2.38: Создание файла nfs.sh в каталоге provision/client

В скрипте клиента реализована установка пакета nfs-utils, создание точки монтирования /mnt/nfs, подключение ресурса server.alkamal.net:/srv/nfs, добавление строки автоматического монтирования в /etc/fstab с параметром _netdev, а также восстановление контекстов SELinux (рис. 2.39).



```
GNU nano 5.6.1
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils
echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.alkamal.net:/srv/nfs /mnt/nfs
echo "server.alkamal.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Рисунок 2.39: Содержимое скрипта nfs.sh для виртуальной машины client

Таким образом, подготовлены сценарии автоматического развёртывания NFS-инфраструктуры для сервера и клиента, предназначенные для выполнения при загрузке виртуальных машин через механизм provisioning в Vagrantfile.

3 Выводы

В ходе работы реализована конфигурация сервера и клиента NFSv4 с последующей автоматизацией развёртывания средствами Vagrant provisioning.

На сервере выполнена настройка экспорта каталогов, конфигурирование SELinux-контекстов типа `nfs_t`, настройка `firewalld`, организация bind-монтирования каталогов веб-сервера и пользовательского каталога, а также обеспечение их постоянного подключения через `/etc/fstab`.

На клиенте выполнено подключение удалённого ресурса, проверена корректность монтирования, реализовано автоматическое подключение при загрузке системы и подтверждена работоспособность механизма разграничения доступа на основе прав файловой системы.

Создание скриптов `nfs.sh` для сервера и клиента позволило формализовать и автоматизировать процесс настройки, обеспечив воспроизводимость конфигурации виртуальной инфраструктуры.

4 Контрольные вопросы

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

`/etc(exports`

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

Следует открыть TCP и UDP порты 2049 в брандмауэре.

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

Для автоматической установки общих ресурсов NFS при перезагрузке следует использовать опцию `auto` в `/etc/fstab`.