

Отчёт по лабораторной работе 13

Настройка NFS

Суннатилло Махмудов

Содержание

1 Цель работы	5
2 Теоретические сведения	6
2.1 Принцип работы NFS	6
2.2 Версии протокола NFS	7
3 Выполнение лабораторной работы	8
3.1 Настройка сервера NFSv4	8
3.2 Монтирование NFS на клиенте	11
3.3 Подключение каталогов к дереву NFS	13
3.4 Подключение каталогов для работы пользователей	15
3.5 Внесение изменений в настройки внутреннего окружения виртуальных машин	18
4 Вывод	20
5 Контрольные вопросы	21
6 Список литературы	22

Список иллюстраций

3.1 Установка и перезапуск демонов systemd	8
3.2 Создание каталога и редактирование файла /etc/exports	9
3.3 Настройка контекста безопасности SELinux для каталога /srv/nfs .	9
3.4 Запуск сервиса NFS и настройка правил межсетевого экрана	10
3.5 Проверка доступных экспортов на клиенте	10
3.6 Добавление служб mountd и rpc-bind и повторная проверка доступности экспорта	11
3.7 Монтирование и проверка NFS-ресурса на клиенте	12
3.8 Добавление записи для автоматического монтирования в /etc/fstab	12
3.9 Проверка состояния remote-fs.target	13
3.10 Редактирование файла /etc/fstab для монтирования каталога веб-сервера	13
3.11 Проверка доступности каталога www на клиенте	14
3.12 Добавление записи экспорта каталога веб-сервера в /etc/exports .	14
3.13 Создание каталога common и файла на сервере	15
3.14 Добавление записи в /etc/exports для домашнего каталога пользователя	16
3.15 Настройка постоянного bind-монтирования в /etc/fstab	16
3.16 Создание файла на клиенте в NFS-каталоге	17
3.17 Попытка доступа root к каталогу пользователя	17
3.18 Проверка файлов на сервере	18
3.19 Содержимое скрипта nfs.sh на сервере	19
3.20 Содержимое скрипта nfs.sh на клиенте	19

Список таблиц

1 Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Теоретические сведения

Network File System (NFS) — это сетевая файловая система, разработанная компанией **Sun Microsystems**, которая позволяет различным узлам в сети обмениваться файлами и каталогами так, как если бы они находились в локальной файловой системе.

Основное назначение NFS — организация централизованного хранения данных с возможностью их совместного использования.

2.1 Принцип работы NFS

Работа NFS основана на архитектуре **клиент–сервер**.

Сервер экспортирует (предоставляет общий доступ) к определённым каталогам, а клиенты монтируют эти каталоги в свою локальную файловую систему. Таким образом, пользователь может работать с удалёнными файлами как с локальными.

Для взаимодействия NFS использует набор RPC-служб (Remote Procedure Call). Основные компоненты: - **rpcbind** — служба, регистрирующая RPC-процессы и сопоставляющая их номера портов;

- **nfsd** — демон, обеспечивающий обработку запросов от клиентов;
- **mountd** — служба, отвечающая за экспорт и монтирование каталогов;
- **statd** и **lockd** — обеспечивают блокировку и согласованность доступа к файлам.

2.2 Версии протокола NFS

Наиболее распространённые версии – **NFSv3** и **NFSv4**:

- **NFSv3** – использует UDP/TCP и не требует авторизации, проста в настройке, но менее безопасна;
- **NFSv4** – поддерживает работу через один порт (2049), интегрирована с механизмами **Kerberos**, поддерживает ACL и улучшенную производительность.

3 Выполнение лабораторной работы

3.1 Настройка сервера NFSv4

1. На виртуальной машине **server** был установлен необходимый пакет nfs-utils и его зависимости. После завершения установки выполнен перезапуск системных демонов для применения новых конфигураций.

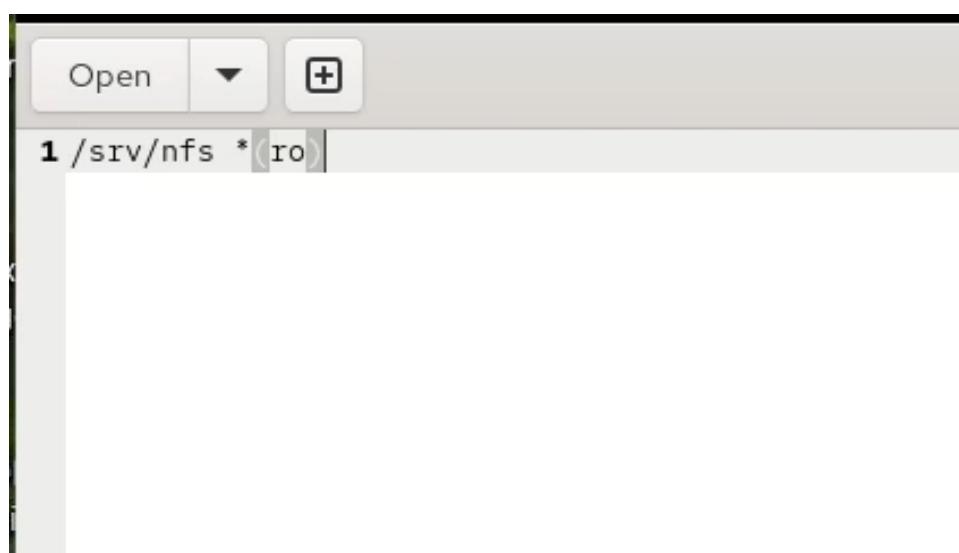


Рис. 3.1: Установка и перезапуск демонов systemd

2. Был создан каталог, предназначенный для экспорта через NFS, и отредактирован файл **/etc(exports**, где задан общий ресурс с доступом только на чтение:

/srv/nfs *(ro)

```
Installed:
gssproxy-0.9.2-10.el10.x86_64          libev-4.33-14.el10.x86_64          libnfsidmap-1:2.
libverto-libev-0.3.2-10.el10.x86_64      nfs-utils-1:2.8.2-3.el10.x86_64      rpcbind-1.2.7-3.
sssd-nfs-idmap-2.10.2-3.el10_0.2.x86_64

Complete!
[root@server.smahmudov.net ~]# systemctl daemon-reload
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# mkdir -p /srv/nfs
[root@server.smahmudov.net ~]# gedit /etc/exports
[root@server.smahmudov.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.smahmudov.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.smahmudov.net ~]# systemctl start nfs-server.service
[root@server.smahmudov.net ~]# systemctl enable nfs-server.service
Created symlink '/etc/systemd/system/multi-user.target.wants/nfs-server.service' → '/usr/lib/systemd/system/
[root@server.smahmudov.net ~]# firewall-cmd --add-service=nfs
success
[root@server.smahmudov.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.smahmudov.net ~]# firewall-cmd --reload
success
[root@server.smahmudov.net ~]#
```

Рис. 3.2: Создание каталога и редактирование файла /etc/exports

3. Для каталога /srv/nfs был установлен контекст безопасности **nfs_t**, после чего были применены изменения SELinux.

```
Installed:
gssproxy-0.9.2-10.el10.x86_64          libev-4.33-14.el10.x86_64          libnfsidmap-1:2.8.2-3.el10.x86_64
libnfsidmap-1:2.8.2-3.el10.x86_64      nfs-utils-1:2.8.2-3.el10.x86_64      rpcbind-1.2.7-3.el10.x86_64
sssd-nfs-idmap-2.10.2-3.el10_0.2.x86_64

Complete!
[root@client.smahmudov.net ~]# systemctl daemon-reload
[root@client.smahmudov.net ~]#
[root@client.smahmudov.net ~]# showmount -e server.smahmudov.net
clnt_create: RPC: Unable to receive
[root@client.smahmudov.net ~]# showmount -e server.smahmudov.net
Export list for server.smahmudov.net:
/srv/nfs
[root@client.smahmudov.net ~]#
```

Рис. 3.3: Настройка контекста безопасности SELinux для каталога /srv/nfs

4. Был запущен и добавлен в автозагрузку сервис NFS. Для корректной работы сервера также были добавлены необходимые службы в брандмауэр и произведена перезагрузка его конфигурации.

```

rpcbind 17460          rpc     8u    IPv6          61100      0t0      UDP *:sunrpc
rpc.statd 17461         rpcuser  7u    IPv4          74994      0t0      UDP *:37354
rpc.statd 17461         rpcuser  9u    IPv6          74999      0t0      UDP *:34883
rpc.statd 17461         rpcuser 29u    IPv4          74990      0t0      UDP localhost:700
rpc.mount 17485          root    4u    IPv4          74661      0t0      UDP *:mountd
rpc.mount 17485          root    6u    IPv6          74665      0t0      UDP *:mountd
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# firewall-cmd --get-services
0:AdRH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1600 anno-1800 apcu ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgn bitcoin bitcoin-rpc bitcoin-estnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civilization-iv civilization-v cockpit collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpcv6 dhcpcv6-client distcc dns dns-over-quic dns-over-tls dockr-swarm dropbox-lansync elasticsearch etcd-client etcd-server factorio finger foreman foreman-proxy freeipa-4 freeipa-s freeipa-replication freeipa-trust ftp galeria ganglia-client ganglia-master git gpd grafana gre high-availability htent imap imaps iperf2 ipfs ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kswd kprop kshell kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-connecure kube-nodeport-services kube-scheduler kube-worker kubelet kubelet-readonly kubelet-worker t libvirt-tls lightning-network llmntr llmntr-client llmntr-tcp llmntr-udp managesieve matrix mdns memcache minecraft minio mosh mountd mpd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashle-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pop3s postgresql privoxy prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptcp pulseaudio puppetmaster adsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane setlection sip sips slimervr slp smtp smtp-submission smtpts snmp snmpfsl snmpfsls-snmptrap spiderOak-lansync spotify-sh ssh-custom statsrv steam-lan-transfer steam-streaming stellaris stronghold-crusader stun stun submission supertuxka:thing syncthing-gui syncthing-relay synergy syscomlan syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-n-client turn turns upnp-client vdsman vnc-server vrpp warpinator wbem-wbem https wireguard ws-discovery ws-discovery:every-host ws-discovery-tcp ws-discovery-udp wsdd wsdd-https wsmans wsman xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-srt zabbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]#
[root@server.smahmudov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind
success
[root@server.smahmudov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.smahmudov.net ~]# firewall-cmd --reload
success
[root@server.smahmudov.net ~]#

```

Рис. 3.4: Запуск сервиса NFS и настройка правил межсетевого экрана

- На виртуальной машине **client** был установлен пакет **nfs-utils** и выполнена проверка доступных экспортируемых каталогов на сервере. Команда вывела список экспортируемых директорий, доступных клиентам сети.

```

[root@client.smahmudov.net ~]# systemctl daemon-reload
[root@client.smahmudov.net ~]#
[root@client.smahmudov.net ~]# showmount -e server.smahmudov.net
clnt_create: RPC: Unable to receive
[root@client.smahmudov.net ~]# showmount -e server.smahmudov.net
Export list for server.smahmudov.net:
/srv/nfs *
[root@client.smahmudov.net ~]# showmount -e server.smahmudov.net
Export list for server.smahmudov.net:
/srv/nfs *
[root@client.smahmudov.net ~]#

```

Рис. 3.5: Проверка доступных экспортов на клиенте

- После добавления дополнительных служб **rpc-bind** и **mountd** в настройки брандмауэра на сервере была выполнена повторная проверка, показавшая корректный список экспортов.

```
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=345560k,nr_inodes=819200,mode=755,inode64)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=36,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=4915)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugegetlbfs on /dev/hugepages type hugegetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymf
ollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
/dev/sda2 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
/dev/sda1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=ascii,shortname=winnt,errors=remount-ro)
/dev/mapper/r1_vbox-home on /home type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k
,noquota)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000,_netdev)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=172776k,nr_inodes=43194,mode
=700,uid=1001,gid=1001,inode64)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_i
d=1001)
portal on /run/user/1001/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=172776k,nr_inodes=43194,mode
=700,inode64)
server.smahmudov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,n
amlen=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.smahmudov.net ~]#
[root@client.smahmudov.net ~]#
```

Рис. 3.6: Добавление служб mountd и rpc-bind и повторная проверка доступности экспорта

3.2 Мониторинг NFS на клиенте

- На клиентской машине был создан каталог **/mnt/nfs**, после чего удалённый ресурс был смонтирован в него. Проверка подключённых файловых систем показала, что NFS-ресурс успешно смонтирован.

```

*fstab
/etc
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-E25B /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.smahmudov.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. 3.7: Монтирование и проверка NFS-ресурса на клиенте

- Для обеспечения автоматического монтирования при загрузке системы в конец файла **/etc/fstab** была добавлена строка:

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

Здесь параметр **“_netdev”** указывает, что устройство зависит от сети, и его монтирование должно выполняться после активации сетевых служб.

```

[root@client smahmudov.net ~]#
[root@client smahmudov.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 2025-10-27 06:40:32 UTC; 20min ago
        Docs: man:systemd.special(7)

Oct 27 06:40:32 client.smahmudov.net systemd[1]: Reached target remote-fs.target - Remote File System
[root@client smahmudov.net ~]#

```

Рис. 3.8: Добавление записи для автоматического монтирования в /etc/fstab

- Проверка статуса цели **remote-fs.target**, отвечающей за автоматическое подключение удалённых файловых систем, показала активное состояние сервиса.

```
[smahmudov@client.smahmudov.net ~]$ sudo -i
[sudo] password for smahmudov:
[root@client.smahmudov.net ~]#
[root@client.smahmudov.net ~]# mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.smahmudov.net:/srv/nfs on /mnt/nfs type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262
amlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.30,local_lock=none,add
.168.1.1,_netdev)
[root@client.smahmudov.net ~]#
```

Рис. 3.9: Проверка состояния remote-fs.target

4. После перезапуска клиента было подтверждено, что каталог NFS автоматически подключается при старте системы.

3.3 Подключение каталогов к дереву NFS

1. На сервере был создан каталог **/srv/nfs/www**, предназначенный для размещения контента веб-сервера, экспортируемого через NFS.
2. Каталог **/var/www** был подмонтирован в созданный каталог **/srv/nfs/www** с использованием опции **bind**, что позволило отразить содержимое веб-каталога в дереве NFS.

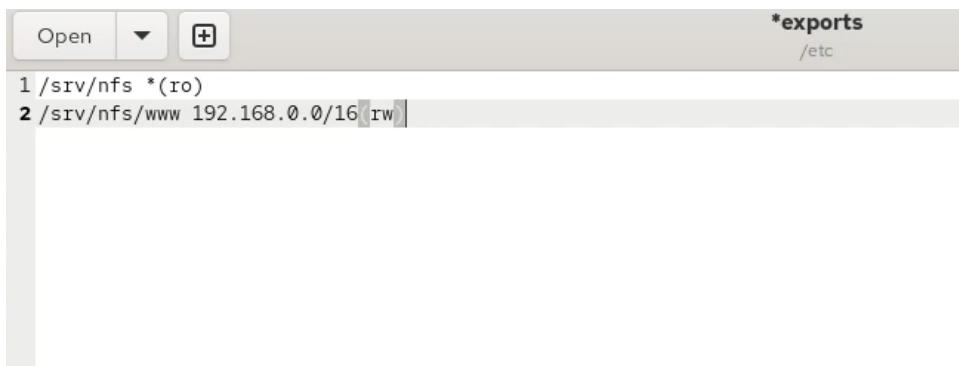
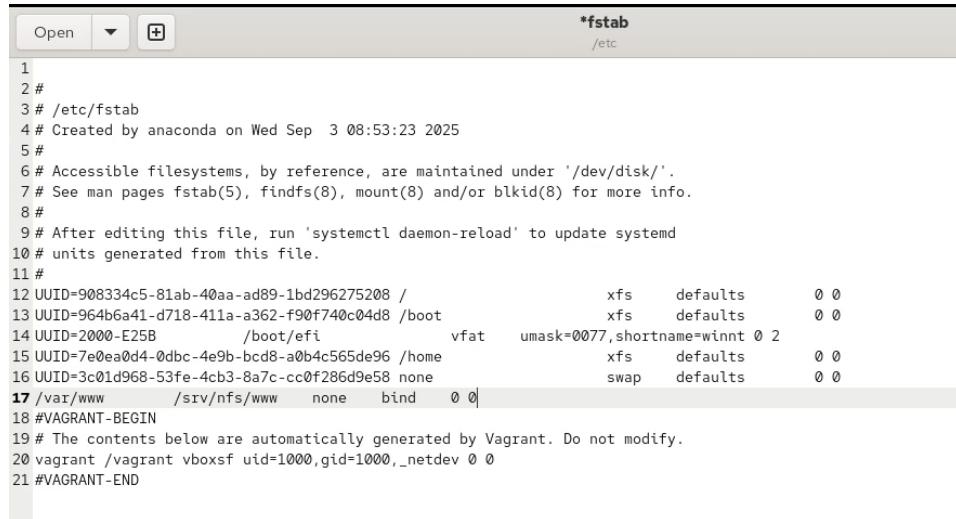


Рис. 3.10: Редактирование файла /etc/fstab для монтирования каталога веб-сервера

3. Проверка содержимого каталога **/srv/nfs** показала наличие подкаталога **www**, что подтверждает успешное монтирование каталога веб-сервера.

4. На клиентской машине в каталоге **/mnt/nfs** также был отображён подкаталог **www**, что свидетельствует о корректной работе экспортируемого ресурса.

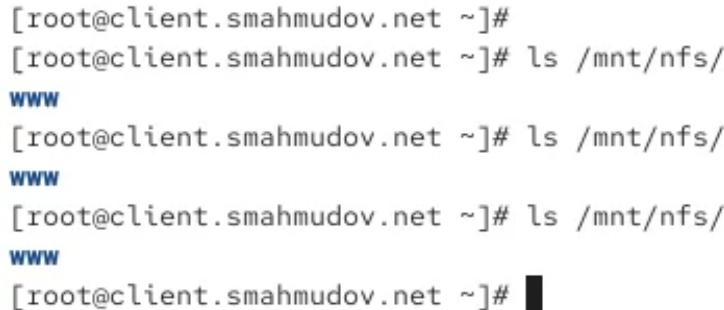


```
*fstab
/etc
1
2 #
3 # /etc/fstab
4 # Created by anaconda on Wed Sep 3 08:53:23 2025
5 #
6 # Accessible filesystems, by reference, are maintained under '/dev/disk/'.
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
8 #
9 # After editing this file, run 'systemctl daemon-reload' 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-E25B /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 /var/www /srv/nfs/www none bind 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. 3.11: Проверка доступности каталога www на клиенте

5. В файл **/etc(exports** на сервере была добавлена строка, разрешающая запись в каталог **/srv/nfs/www** для узлов из подсети **192.168.0.0/16**:

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



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

Рис. 3.12: Добавление записи экспорта каталога веб-сервера в /etc/exports

6. После обновления конфигурации экспортов команда `exportfs -r` применила изменения, и каталог **/srv/nfs/www** стал доступен клиентам сети с правами записи.

7. На клиенте была повторно проведена проверка содержимого каталога **/mnt/nfs**, подтвердившая успешный экспорт и доступность каталога веб-сервера.

3.4 Подключение каталогов для работы пользователей

1. На сервере под пользователем **smahmudov** был создан каталог **common** с правами доступа только для владельца. Внутри каталога был создан файл **smahmudov@server.txt**. Каталог защищён правами **700**, что обеспечивает доступ лишь владельцу.

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

Рис. 3.13: Создание каталога **common** и файла на сервере

2. В директории NFS на сервере был подготовлен каталог **/srv/nfs/home/smahmudov** для сетевого доступа пользователя.
3. Локальный каталог пользователя был подмонтирован в структуру NFS. В результате каталог **/srv/nfs/home/smahmudov** стал отражать содержимое **~/common**. Права доступа на каталог сохранились — доступ разрешён только владельцу.

```
*exports  
/etc  
1 /srv/nfs *(ro)  
2 /srv/nfs/www 192.168.0.0/16(rw)  
3 /srv/nfs/home/smahmudov 192.168.0.0/16(rw)
```

Рис. 3.14: Добавление записи в /etc/exports для домашнего каталога пользователя

4. В файл **/etc(exports** была добавлена строка, разрешающая rw-доступ клиентам в сети

После этого конфигурация экспортов была обновлена.

5. Для постоянного монтирования в **/etc/fstab** была внесена запись, обеспечивающая привязку локального каталога пользователя к каталогу экспортируемому через NFS:

```
*fstab  
/etc  
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-E25B /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 /var/www /srv/nfs/www none bind 0 0  
18 /home/smahmudov/common /srv/nfs/home/smahmudov none bind 0 0|  
19 #VAGRANT-BEGIN  
20 # The contents below are automatically generated by Vagrant. Do not modify.  
21 vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0  
22 #VAGRANT-END
```

Рис. 3.15: Настройка постоянного bind-монтирования в /etc/fstab

6. После повторного экспорта каталогов на клиенте в структуре `/mnt/nfs` появился каталог домашнего пространства пользователя.
7. На клиенте пользователь перешёл в каталог `/mnt/nfs/home/smahmudov` и создал файл `smahmudov@client.txt`. Файл успешно появился, что подтверждает корректность записи и синхронизации.

```
[smahmudov@client.smahmudov.net ~]$ cd /mnt/nfs/home/smahmudov/  
[smahmudov@client.smahmudov.net smahmudov]$ ls  
smahmudov@server.txt  
[smahmudov@client.smahmudov.net smahmudov]$ touch smahmudov@client.txt  
[smahmudov@client.smahmudov.net smahmudov]$ ls  
smahmudov@client.txt smahmudov@server.txt  
[smahmudov@client.smahmudov.net smahmudov]$ █
```

Рис. 3.16: Создание файла на клиенте в NFS-каталоге

8. При попытке выполнить те же действия от имени `root` на клиенте доступ был запрещён, что подтверждает корректное разграничение прав доступа.

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

Рис. 3.17: Попытка доступа `root` к каталогу пользователя

9. На сервере в каталоге `~/common` были видны оба файла – созданный на сервере и созданный на клиенте, что подтверждает корректность двусторонней синхронизации через NFS.

```
[smahmudov@client.smahmudov.net smahmudov]$ sudo -i  
[sudo] password for smahmudov:  
[root@client.smahmudov.net ~]# ls  
anaconda-ks.cfg original-ks.cfg  
[root@client.smahmudov.net ~]# cd /mnt/nfs/home/smahmudov/  
-bash: cd: /mnt/nfs/home/smahmudov/: Permission denied  
[root@client.smahmudov.net ~]#  
logout  
[smahmudov@client.smahmudov.net smahmudov]$
```

Рис. 3.18: Проверка файлов на сервере

3.5 Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине **server** был выполнен переход в каталог **/vagrant/provision/server/**, где создана структура каталогов для хранения конфигурационных файлов NFS:

`/vagrant/provision/server/nfs/etc`

В созданный каталог были скопированы файлы конфигурации сервера NFS, включая **/etc(exports**.

2. В каталоге **/vagrant/provision/server/** был создан исполняемый скрипт **nfs.sh**, содержащий команды для автоматической настройки и запуска NFS-сервера. Скрипт выполняет установку пакетов, настройку SELinux, конфигурацию межсетевого экрана, монтирование каталогов и добавление записей в **/etc/fstab**.

```

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 --permanent
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/smahmudov
21 mkdir -p -m 700 /home/smahmudov/common
22 chown smahmudov:smahmudov /home/smahmudov/common
23 mount -o bind /home/smahmudov/common /srv/nfs/home/smahmudov
24 echo "/home/smahmudov/common /srv/nfs/home/smahmudov 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

```

I

Рис. 3.19: Содержимое скрипта nfs.sh на сервере

3. Аналогичный скрипт был создан на виртуальной машине **client**, обеспечивающий автоматическое подключение удалённого NFS-ресурса при инициализации системы. В нём реализованы установка клиента NFS, создание точки монтирования и добавление соответствующей записи в файл **/etc/fstab**.

```

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.smahmudov.net:/srv/nfs /mnt/nfs
8  echo "server.smahmudov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
9  restorecon -vR /etc
10

```

Рис. 3.20: Содержимое скрипта nfs.sh на клиенте

4. После запуска скриптов сервер и клиент корректно настроили взаимодействие по протоколу NFS, обеспечив автоматическое монтирование каталогов и доступность экспортируемых ресурсов.

4 Вывод

В ходе лабораторной работы был развернут и настроен сервер **NFSv4**, обеспечивающий совместный доступ к каталогам между сервером и клиентом.

На стороне сервера выполнена настройка экспорта директорий, конфигурация SELinux и правил брандмауэра, а также организация автоматического мониторинга через **/etc/fstab**.

На клиентской машине реализовано подключение удалённых ресурсов с автоматической инициализацией при запуске системы.

Проверка показала корректную синхронизацию данных и разграничение прав доступа пользователей.

5 Контрольные вопросы

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

Конфигурация экспортируемых ресурсов NFS хранится в файле **/etc(exports**, где указываются пути к каталогам, параметры доступа и список сетей или хостов, которым разрешено подключение.

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

Для корректной работы NFS необходимо открыть следующие порты:

- **2049/tcp** и **2049/udp** – основной порт сервиса NFS;
- **111/tcp** и **111/udp** – служба **rpcbind**;
- порты, используемые сервисами **mountd**, **nlockmgr** и **statd** (динамические, задаются в конфигурации при необходимости).

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 при запуске системы используется опция ****_netdev****, указывающая, что устройство зависит от сети и должно подключаться только после инициализации сетевых служб.

6 Список литературы

- 1. Red Hat Documentation – Network File System (NFS)**

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_file_systems/nfs-to-share-directories_managing-file-systems

- 2. The Linux NFS HOWTO**

<https://tldp.org/HOWTO/NFS-HOWTO/index.html>