

Лабораторная работа №13

Настройка NFS

Спелов Андрей Николаевич

Содержание

1 Цель работы	6
2 Выполнение лабораторной работы	7
3 Выводы	20
4 Ответы на контрольные вопросы:	21
Список литературы	22

Список иллюстраций

2.1 Установка на сервере программного обеспечения nfs-utils.	7
2.2 Создание на сервере каталога, который предполагается сделать доступным всем пользователям сети (корень дерева NFS).	8
2.3 Прописывание в файле /etc(exports подключаемого через NFS общего каталога с доступом только на чтение.	8
2.4 Настройка для общего каталога контекста безопасности NFS, применение изменённой настройки SELinux к файловой системе, запуск сервера NFS и настройка межсетевого экрана для работы сервера NFS.	8
2.5 Установка на клиенте программного обеспечения для работы NFS.	9
2.6 Попытка просмотреть на клиенте имеющихся подмонтированных удалённых ресурсов.	9
2.7 Попытка остановить на сервере сервис межсетевого экрана.	9
2.8 Повторная попытка подключения к удалённо смонтированному ресурсу.	9
2.9 Запуск на сервере сервиса межсетевого экрана.	10
2.10 Просмотр на сервере служб, задействованных при удалённом мониторинге (TCP).	10
2.11 Просмотр на сервере служб, задействованных при удалённом мониторинге (UDP).	11
2.12 Добавление службы rpc-bind и mountd в настройки межсетевого экрана на сервере.	11
2.13 Проверка на клиенте подключения удалённого ресурса.	12
2.14 Создание на клиенте каталога, в который будет монтироваться удалённый ресурс, и монтироваться дерево NFS. Проверка, что общий ресурс NFS подключён правильно.	12
2.15 Добавление на клиенте в конце файла /etc/fstab записи.	12
2.16 Проверка на клиенте наличия автоматического мониторинга удалённых ресурсов при запуске операционной системы.	13
2.17 Создание на сервере общего каталога, в который затем будет подмонтирован каталог с контентом веб-сервера. Монтирование каталога web-сервера и проверка, что отображается в каталоге /srv/nfs.	13
2.18 Просмотр на клиенте содержимого каталога /mnt/nfs.	13
2.19 Добавление на сервере в файле /etc(exports экспорта каталога веб-сервера с удалённого ресурса.	14
2.20 Экспорт всех каталогов, упомянутых в файле /etc(exports.	14
2.21 Проверка на клиенте каталога /mnt/nfs.	14

2.22 Добавление на сервере в конце файла /etc/fstab записи.	14
2.23 Повторный экспорт каталогов, указанных в файле /etc(exports.	14
2.24 Проверка на клиенте каталога /mnt/nfs.	15
2.25 Создание на сервере под пользователем anspelov в домашнем ка- талоге каталога common с полными правами доступа только для этого пользователя, а в нём файл anspelov@server.txt. Создание общего каталога для работы пользователя anspelov по сети и мон- тирование каталога common пользователя anspelov в NFS.	15
2.26 Подключение каталога пользователя в файле /etc(exports	15
2.27 Внесение изменения в файл /etc/fstab.	15
2.28 Повторный экспорт каталогов.	16
2.29 Проверка на клиенте каталога /mnt/nfs.	16
2.30 Переход на клиенте под пользователем anspelov в ката- лог /mnt/nfs/home/anspelov и попытка создать в нём файл anspelov@client.txt.	16
2.31 Переход на клиенте под пользователем root в каталог /mnt/nfs/home/anspelov и попытка создать в нём файл anspelov@client.txt.	16
2.32 Просмотр наличия изменений на сервере в каталоге пользователя /home/anspelov/common.	17
2.33 Переход на виртуальной машине server в каталог для внесения из- менений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога nfs, в который помещаем в соответствую- щие подкаталоги конфигурационные файлы. Создание в каталоге /vagrant/provision/server исполняемого файла nfs.sh.	17
2.34 Открытие файла на редактирование и добавление скрипта.	18
2.35 Переход на виртуальной машине client в каталог для внесения из- менений в настройки внутреннего окружения /vagrant/provision/client/. Создание в каталоге /vagrant/provision/client исполняемого файла nfs.sh.	18
2.36 Открытие файла на редактирование и добавление скрипта.	19
2.37 Добавление записи в конфигурационном файле Vagrantfile для сервера и для клиента.	19

Список таблиц

1 Цель работы

Целью данной работы является приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

2 Выполнение лабораторной работы

На сервере установим необходимое программное обеспечение(рис. 2.1).

The screenshot shows a terminal window with a pink header bar containing the text "root@server:~ - sudo -i". The main area of the terminal displays the output of a package manager (likely yum or dnf) during an installation process. The output is as follows:

```
=====
Installing:
 nfs-utils           x86_64    1:2.8.3-0.el10      baseos      475 k
Upgrading:
 libipa_hbac         x86_64    2.11.1-2.el10_1.1   baseos      34 k
 libldb              x86_64    4.22.4-106.el10     baseos     181 k
 libsmclient          x86_64    4.22.4-106.el10     baseos      75 k
 libsses_certmap     x86_64    2.11.1-2.el10_1.1   baseos      81 k
 libsses_idmap        x86_64    2.11.1-2.el10_1.1   baseos      41 k
 libsses_nss_idmap   x86_64    2.11.1-2.el10_1.1   baseos      44 k
 libsses_sudo         x86_64    2.11.1-2.el10_1.1   baseos      33 k
 libtalloc             x86_64    2.4.3-100.el10      baseos      33 k
 libtcrdb             x86_64    1.4.13-100.el10     baseos      55 k
 libtevent             x86_64    0.16.2-100.el10     baseos      50 k
 libwbclient           x86_64    4.22.4-106.el10     baseos      43 k
 samba-client-libs    x86_64    4.22.4-106.el10     baseos      5.3 M
 samba-common          noarch   4.22.4-106.el10     baseos     174 k
 samba-common-libs    x86_64    4.22.4-106.el10     baseos     104 k
 sssd                 x86_64    2.11.1-2.el10_1.1   baseos      25 k
 sssd-ad               x86_64    2.11.1-2.el10_1.1   baseos     195 k
 sssd-client            x86_64    2.11.1-2.el10_1.1   baseos     152 k
 sssd-common            x86_64    2.11.1-2.el10_1.1   baseos      1.5 M
 sssd-common-pac       x86_64    2.11.1-2.el10_1.1   baseos      89 k
 sssd-ipa              x86_64    2.11.1-2.el10_1.1   baseos     274 k
 sssd-kcm               x86_64    2.11.1-2.el10_1.1   baseos     103 k
 sssd-krb5              x86_64    2.11.1-2.el10_1.1   baseos      62 k
 sssd-krb5-common       x86_64    2.11.1-2.el10_1.1   baseos      93 k
 sssd-ldap              x86_64    2.11.1-2.el10_1.1   baseos     132 k
 sssd-proxy             x86_64    2.11.1-2.el10_1.1   baseos      70 k
Installing dependencies:
 gssproxy              x86_64    0.9.2-10.el10      baseos      111 k
 libev                 x86_64    4.33-14.el10      baseos      52 k
 libnfsidmap            x86_64    1:2.8.3-0.el10     baseos      61 k
 libverto-libev          x86_64    0.3.2-10.el10     baseos      13 k
 rpcbind                x86_64    1.2.7-3.el10      baseos      57 k
 sssd-nfs-idmap          x86_64    2.11.1-2.el10_1.1   baseos      36 k
=====
Transaction Summary
=====
Install  7 Packages
Upgrade  25 Packages

Total download size: 9.7 M
Downloading Packages:
 [ 1/32]: gssproxy-0.9.2-10.el10.x86_64.x 0% [
```

At the bottom of the terminal window, there is a progress bar indicating the download of the gssproxy package, showing 0% completion.

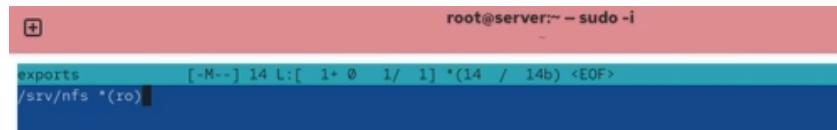
Рис. 2.1: Установка на сервере программного обеспечения nfs-utils.

На сервере создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS) (рис. 2.2).

```
[root@server.an spelov.net ~]# mkdir -p /srv/nfs  
[root@server.an spelov.net ~]# █
```

Рис. 2.2: Создание на сервере каталога, который предполагается сделать доступным всем пользователям сети (корень дерева NFS).

В файле /etc(exports пропишем подключаемый через NFS общий каталог с доступом только на чтение(рис. 2.3).



```
root@server:~ - sudo -i  
root@server:~ - nano /etc/exports  
exports /srv/nfs *(ro)
```

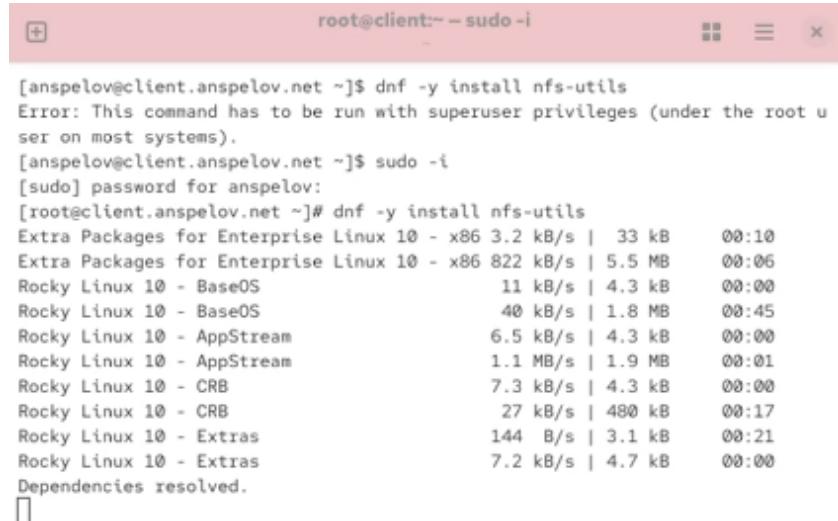
Рис. 2.3: Прописывание в файле /etc/exports подключаемого через NFS общего каталога с доступом только на чтение.

Для общего каталога зададим контекст безопасности NFS: semanage fcontext -a -t nfs_t “/srv/nfs(/.*)?” Применим изменённую настройку SELinux к файловой системе: restorecon -vR /srv/nfs Запустим сервер NFS: systemctl start nfs-server.service systemctl enable nfs-server.service Настроим межсетевой экран для работы сервера NFS(рис. 2.4).

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

Рис. 2.4: Настройка для общего каталога контекста безопасности NFS, применение изменённой настройки SELinux к файловой системе, запуск сервера NFS и настройка межсетевого экрана для работы сервера NFS.

На клиенте установим необходимое для работы NFS программное обеспечение (рис. 2.5).



```
[root@client:~ - sudo -i
[anspelov@client.an spelov.net ~]$ dnf -y install nfs-utils
Error: This command has to be run with superuser privileges (under the root user on most systems).
[anspelov@client.an spelov.net ~]$ sudo -i
[sudo] password for an spelov:
[root@client.an spelov.net ~]# dnf -y install nfs-utils
Extra Packages for Enterprise Linux 10 - x86 3.2 kB/s | 33 kB 00:10
Extra Packages for Enterprise Linux 10 - x86 822 kB/s | 5.5 MB 00:06
Rocky Linux 10 - BaseOS 11 kB/s | 4.3 kB 00:00
Rocky Linux 10 - BaseOS 40 kB/s | 1.8 MB 00:45
Rocky Linux 10 - AppStream 6.5 kB/s | 4.3 kB 00:00
Rocky Linux 10 - AppStream 1.1 MB/s | 1.9 MB 00:01
Rocky Linux 10 - CRB 7.3 kB/s | 4.3 kB 00:00
Rocky Linux 10 - CRB 27 kB/s | 480 kB 00:17
Rocky Linux 10 - Extras 144 B/s | 3.1 kB 00:21
Rocky Linux 10 - Extras 7.2 kB/s | 4.7 kB 00:00
Dependencies resolved.
[
```

Рис. 2.5: Установка на клиенте программного обеспечения для работы NFS.

На клиенте попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы (рис. 2.6).

```
[root@client.an spelov.net ~]# showmount -e server.an spelov.net
clnt_create: RPC: Unable to receive
[root@client.an spelov.net ~]# [
```

Рис. 2.6: Попытка просмотреть на клиенте имеющихся подмонтированных удалённых ресурсов.

Попробуем на сервере остановить сервис межсетевого экрана (рис. 2.7).

```
[root@server.an spelov.net ~]# systemctl stop firewalld.service
[root@server.an spelov.net ~]# [
```

Рис. 2.7: Попытка остановить на сервере сервис межсетевого экрана.

Затем на клиенте вновь попробуем подключиться к удалённо смонтированному ресурсу(рис. 2.8).

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

Рис. 2.8: Повторная попытка подключения к удалённо смонтированному ресурсу.

На сервере запустим сервис межсетевого экрана (рис. 2.9).

```
[root@server.an spelov.net ~]# systemctl start firewalld  
[root@server.an spelov.net ~]# █
```

Рис. 2.9: Запуск на сервере сервиса межсетевого экрана.

На сервере посмотрим, какие службы задействованы при удалённом мониторинге: lsof | grep TCP(рис. 2.10).

```
[root@server.an spelov.net ~]# lsof | grep TCP  
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs  
Output information may be incomplete.  
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc  
Output information may be incomplete.  
systemd      1          root  209u   IPv4          47072    0t0      TCP *:sunrpc (LISTEN  
N)  
systemd      1          root  211u   IPv6          47083    0t0      TCP *:sunrpc (LISTEN  
N)  
systemd      1          root  226u   IPv6          8688     0t0      TCP *:websm (LISTEN  
)  
cupsd     1268          root    7u   IPv6          11446    0t0      TCP localhost:ipp (LISTEN  
cupsd     1268          root    8u   IPv4          11447    0t0      TCP localhost:ipp (LISTEN  
sshd     1279          root    7u   IPv4          11437    0t0      TCP *:down (LISTEN)  
sshd     1279          root    8u   IPv6          10698    0t0      TCP *:down (LISTEN)  
sshd     1279          root    9u   IPv4          10700    0t0      TCP *:ssh (LISTEN)  
sshd     1279          root   10u   IPv6          10702    0t0      TCP *:ssh (LISTEN)  
named    1335          named   27u   IPv4          11598    0t0      TCP localhost:domain (LISTEN)  
named    1335          named   29u   IPv4          11599    0t0      TCP localhost:domain (LISTEN)  
named    1335          named   33u   IPv4          11602    0t0      TCP server.an spelov (LISTEN  
.net:domain (LISTEN)  
named    1335          named   34u   IPv4          11603    0t0      TCP server.an spelov (LISTEN  
.net:domain (LISTEN)  
named    1335          named   41u   IPv6          11610    0t0      TCP localhost:domain (LISTEN)  
named    1335          named   42u   IPv6          11611    0t0      TCP localhost:domain (LISTEN)  
named    1335          named   43u   IPv4          10750    0t0      TCP localhost:rndc (LISTEN)  
named    1335          named   44u   IPv4          10751    0t0      TCP localhost:rndc (LISTEN)  
named    1335          named   45u   IPv6          10753    0t0      TCP localhost:rndc (LISTEN)  
named    1335          named   46u   IPv6          10754    0t0      TCP localhost:rndc (LISTEN)  
named    1335          named   58u   IPv4          26648    0t0      TCP www.an spelov.net (LISTEN)  
named    1335          named   59u   IPv4          26649    0t0      TCP www.an spelov.net (LISTEN)
```

Рис. 2.10: Просмотр на сервере служб, задействованных при удалённом мониторинге (TCP).

На сервере посмотрим, какие службы задействованы при удалённом мониторинге: lsof | grep UDP(рис. 2.11).

```
[root@server.anspelov.net ~]# lsof | grep UDP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
      Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
      Output information may be incomplete.

systemd      1          root    210u   IPv4          48226    0t0    UDP *:sunrpc
systemd      1          root    212u   IPv6          47090    0t0    UDP *:sunrpc
avahi-dae  881          avahi    12u   IPv4          8038    0t0    UDP *:mdns
avahi-dae  881          avahi    13u   IPv6          8039    0t0    UDP *:mdns
chronynd  954          chrony    5u   IPv4          9110    0t0    UDP localhost:323
chronynd  954          chrony    6u   IPv6          9111    0t0    UDP localhost:323
chronynd  954          chrony    7u   IPv4          9112    0t0    UDP *:ntp
named    1335          named   25u   IPv4          11596    0t0    UDP localhost:domai
n
named    1335          named   26u   IPv4          11597    0t0    UDP localhost:domai
n
named    1335          named   31u   IPv4          11600    0t0    UDP server.anspelov
.named:domain
named    1335          named   32u   IPv4          11601    0t0    UDP server.anspelov
.named:domain
named    1335          named   39u   IPv6          11608    0t0    UDP localhost:domai
n
named    1335          named   40u   IPv6          11609    0t0    UDP localhost:domai
n
named    1335          named   56u   IPv4          26646    0t0    UDP server.anspelov
.named:domain
named    1335          named   57u   IPv4          26647    0t0    UDP server.anspelov
.named:domain
named    1335  1337 isc-net-0          named   25u   IPv4          11596    0t0    UDP localhost:domai
n
named    1335  1337 isc-net-0          named   26u   IPv4          11597    0t0    UDP localhost:domai
n
named    1335  1337 isc-net-0          named   31u   IPv4          11600    0t0    UDP server.anspelov
.named:domain
named    1335  1337 isc-net-0          named   32u   IPv4          11601    0t0    UDP server.anspelov
.named:domain
named    1335  1337 isc-net-0          named   39u   IPv6          11608    0t0    UDP localhost:domai
n
named    1335  1337 isc-net-0          named   40u   IPv6          11609    0t0    UDP localhost:domai
n
named    1335  1337 isc-net-0          named   56u   IPv4          26646    0t0    UDP server.anspelov
.named:domain
named    1335  1337 isc-net-0          named   57u   IPv4          26647    0t0    UDP server.anspelov
.named:domain
named    1335  1338 isc-net-0          named   25u   IPv4          11596    0t0    UDP localhost:domai
-

```

Рис. 2.11: Просмотр на сервере служб, задействованных при удалённом монтировании (UDP).

Добавим службы rpc-bind и mountd в настройки межсетевого экрана на сервере(рис. 2.12).

```
[root@server.anspelov.net ~]# firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 anno-1800 apc
upsd aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bpg bitcoin b
itcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civi
lization-iv civilization-v cockpit collected condor-collector cratedb ctdb dds dds-unicast dhcp dhcpcv6 d
hcpv6-client distcc dns dns-over-quic dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-cl
ient etcd-server factorio finger foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freei
pa-trust ftp galera ganglia-client ganglia-master git gpad grafana gre high-availability http http3 https ident imap
imaps iperf2 iperf3 ipfs ipp ipp-client ipsec irc icsi-target isns jenkins kadmin kdeconnect kerberos kibana kl
ogin kpsswd kprox kshell kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manage
r kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kube
let-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp m
anagesieve matrix mdns memcache minecraft minidlna mndo mongodb mosh mounted mpd mqtt mqtt-tls ms-wbt mssql murmur mys
ql nbd nebulia need-for-speed-most-wanted netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry o
penvpn ovirt-imageio ovirt-storagedconsole ovirt-vncconsole plex pncd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql
proxvxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius r
adsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsysnd rtsp salt-master samba samba-client samba-dc sane se
ttlers-history-collection sip sips slimerv slp smtp smtp-submission snmp snmplts snmplts-trap snnptrap spidero
k-lansync spotify-sync squid ssdp ssh ssh-custom statsrv steam-steam-transfer steam-streaming stellaris stronghold-crus
ader stun stuns submission supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay synergy syscomian syslog sy
slog-tls telnet tentacle terraria tftp title38 time tor-socks transmission-client turn turns upnp-client vds vnc-serv
er vrpp warpinator wbem-https wbem-https wsman wsman xdmcx xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-
java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.anspelov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.anspelov.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent
success
[root@server.anspelov.net ~]# firewall-cmd --reload
success
[root@server.anspelov.net ~]#
```

Рис. 2.12: Добавление службы rpc-bind и mountd в настройки межсетевого экрана на сервере.

На клиенте проверим подключение удалённого ресурса (рис. 2.13).

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

Рис. 2.13: Проверка на клиенте подключения удалённого ресурса.

На клиенте создадим каталог, в который будем монтировать удалённый ресурс, и подмонтируем дерево NFS. После чего проверим, что общий ресурс NFS подключён правильно(рис. 2.14).

```
[root@client.an spelov.net ~]# mkdir -p /mnt/nfs
[root@client.an spelov.net ~]# mount server.an spelov.net:/srv/nfs /mnt/nfs
[root@client.an spelov.net ~]# mount
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=3
2k,noquota)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=211173,mode=755,in
ode64)
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)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegat
e,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
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)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=345392k,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,minp
rroto=5,maxproto=5,direct,pipe_ino=5507)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
hugepages on /dev/hugepages type hugepages (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,rela
time,nosymfollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,nosmap)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
/dev/sda? on /boot type vfat (rw,relatime,seclabel,attr2,inode64,loopbufs=8,loopsize=32k,noa
```

Рис. 2.14: Создание на клиенте каталога, в который будет монтироваться удалённый ресурс, и монтируется дерево NFS. Проверка, что общий ресурс NFS подключён правильно.

На клиенте в конце файла /etc/fstab добавим следующую запись (рис. 2.15).

```
#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.an spelov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```

Рис. 2.15: Добавление на клиенте в конце файла /etc/fstab записи.

На клиенте проверим наличие автоматического монтирования удалённых ресурсов при запуске операционной системы (рис. 2.16).

```
[root@client.an spelov.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 Wed 2025-11-26 11:36:29 UTC; 14min ago
    Invocation: a8e46f3b1ccl4d7bb7333b7faf395b37
      Docs: man:systemd.special(7)

Nov 26 11:36:29 client.an spelov.net systemd[1]: Reached target remote-fs.target - Remote
lines 1-7/7 (END)
```

Рис. 2.16: Проверка на клиенте наличия автоматического монтирования удалённых ресурсов при запуске операционной системы.

На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера: `mkdir -p /srv/nfs/www` После чего подмонтируем каталог web-сервера: `mount -o bind /var/www/ /srv/nfs/www/` И на сервере проверим, что отображается в каталоге `/srv/nfs`(рис. 2.17).

```
-----
[root@server.an spelov.net ~]# mkdir -p /srv/nfs/www
[root@server.an spelov.net ~]# mount -o bind /var/www/ /srv/nfs/www/
[root@server.an spelov.net ~]# ls /srv/nfs
www
[root@server.an spelov.net ~]#
```

Рис. 2.17: Создание на сервере общего каталога, в который затем будет подмонтирован каталог с контентом веб-сервера. Монтируем каталог web-сервера и проверка, что отображается в каталоге `/srv/nfs`.

На клиенте посмотрим, что отображается в каталоге `/mnt/nfs` (рис. 2.18).

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

Рис. 2.18: Просмотр на клиенте содержимого каталога `/mnt/nfs`.

На сервере в файле `/etc/exports` добавим экспорт каталога веб-сервера с удалённого ресурса(рис. 2.19).

```
exports [-M--] 31 L:[ 1+ 1 2/ 2] *(46 / 46b) <EOF>
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
```

Рис. 2.19: Добавление на сервере в файле /etc/exports экспорта каталога веб-сервера с удалённого ресурса.

Экспортируем все каталоги, упомянутые в файле /etc/exports(рис. 2.20).

```
[root@server.an spelov.net ~]# exportfs -r
[root@server.an spelov.net ~]#
```

Рис. 2.20: Экспорт всех каталогов, упомянутых в файле /etc/exports.

Проверим на клиенте каталог /mnt/nfs(рис. 2.21).

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

Рис. 2.21: Проверка на клиенте каталога /mnt/nfs.

На сервере в конце файла /etc/fstab добавим следующую запись(рис. 2.22).

```
# 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
```

Рис. 2.22: Добавление на сервере в конце файла /etc/fstab записи.

Повторно экспортируем каталоги, указанные в файле /etc/exports(рис. 2.23).

```
[root@server.an spelov.net ~]# exportfs -r
[root@server.an spelov.net ~]#
```

Рис. 2.23: Повторный экспорт каталогов, указанных в файле /etc/exports.

На клиенте проверим каталог /mnt/nfs(рис. 2.24).

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

Рис. 2.24: Проверка на клиенте каталога /mnt/nfs.

На сервере под пользователем anspelov в домашнем каталоге создадим каталог common с полными правами доступа только для этого пользователя, а в нём файл anspelov@server.txt. Далее создадим общий каталог для работы пользователя anspelov по сети и подмонтируем каталог common пользователя anspelov в NFS(рис. 2.25).

```
[anspelov@server anspelov.net ~]$ mkdir -p -m 700 ~/common  
[anspelov@server anspelov.net ~]$ cd ~/common  
[anspelov@server anspelov.net common]$ touch anspelov@server.txt  
[anspelov@server anspelov.net common]$ mkdir -p /srv/nfs/home/anspelov  
mkdir: cannot create directory '/srv/nfs/home': Permission denied  
[anspelov@server anspelov.net common]$ sudo mkdir -p /srv/nfs/home/user  
[sudo] password for anspelov:  
[anspelov@server anspelov.net common]$ mount -o bind /home/anspelov/common /srv/nfs/home/anspelov
```

Рис. 2.25: Создание на сервере под пользователем anspelov в домашнем каталоге каталога common с полными правами доступа только для этого пользователя, а в нём файл anspelov@server.txt. Создание общего каталога для работы пользователя anspelov по сети и монтирование каталога common пользователя anspelov в NFS.

Подключим каталог пользователя в файле /etc/exports, прописав в нём(рис. 2.26).

```
exports [-M--] 22 L:[ 1+ 2 3/ 3] *(69 / 88b) 0032 0x020  
/srv/nfs *(ro)  
/srv/nfs/www 192.168.0.0/16(rw)  
/srv/nfs/home/anspelov 192.168.0.0/16(rw)
```

Рис. 2.26: Подключение каталога пользователя в файле /etc/exports

Внесём изменения в файл /etc/fstab(рис. 2.27).

```
# 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/anspelov/common /srv/nfs/home/anspelov none bind 0 0
```

Рис. 2.27: Внесение изменения в файл /etc/fstab.

Повторно экспортируем каталоги(рис. 2.28).

```
[root@server.an spelov.net common]# exportfs -r  
[root@server.an spelov.net common]#
```

Рис. 2.28: Повторный экспорт каталогов.

На клиенте проверим каталог /mnt/nfs(рис. 2.29).

```
[root@client.an spelov.net ~]# ls /mnt/nfs  
home www  
[root@client.an spelov.net ~]# █
```

Рис. 2.29: Проверка на клиенте каталога /mnt/nfs.

На клиенте под пользователем an spelov перейдём в каталог /mnt/nfs/home/an spelov и попробуем создать в нём файл an spelov@client.txt и внести в него какие-либо изменения(рис. 2.30).

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

Рис. 2.30: Переход на клиенте под пользователем an spelov в каталог /mnt/nfs/home/an spelov и попытка создать в нём файл an spelov@client.txt.

Попробуем также это проделать под пользователем root(рис. 2.31).

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

Рис. 2.31: Переход на клиенте под пользователем root в каталог /mnt/nfs/home/an spelov и попытка создать в нём файл an spelov@client.txt.

На сервере посмотрим, появились ли изменения в каталоге пользователя /home/anspelov/common(рис. 2.32).

```
[anspelov@server.anspelov.net common]$ ls /home/anspelov/common
anspelov@client.txt  anspelov@server.txt
[anspelov@server.anspelov.net common]$ █
```

Рис. 2.32: Просмотр наличия изменений на сервере в каталоге пользователя /home/anspelov/common.

На виртуальной машине server перейдём в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог nfs, в который поместим в соответствующие подкаталоги конфигурационные файлы. В каталоге /vagrant/provision/server создадим исполняемый файл nfs.sh(рис. 2.33).

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

Рис. 2.33: Переход на виртуальной машине server в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создание в нём каталога nfs, в который помещаем в соответствующие подкаталоги конфигурационные файлы. Создание в каталоге /vagrant/provision/server исполняемого файла nfs.sh.

Открываем его на редактирование и пропишем в нём следующий скрипт из лабораторной работы(рис. 2.34).

```

nfs.sh          [-M--] 27 L:[ 1+27 28/ 28] *(966 / 966b) <EOF>
#!/bin/bash
echo "Provisioning script $@"
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/anspelov
mkdir -p -m 700 /home/anspelov/common
chown anspelov:anspelov /home/anspelov/common
mount -o bind /home/anspelov/common /srv/nfs/home/anspelov
echo "/home/anspelov/common /srv/nfs/home/anspelov none bind 0 0" >> /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld

```

Рис. 2.34: Открытие файла на редактирование и добавление скрипта.

После чего на виртуальной машине client перейдём в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/. В каталоге /vagrant/provision/client создадим исполняемый файл nfs.sh(рис. 2.35).

```

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

```

Рис. 2.35: Переход на виртуальной машине client в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/. Создание в каталоге /vagrant/provision/client исполняемого файла nfs.sh.

Открываем его на редактирование и пропишем в нём следующий скрипт (рис. 2.36).

```

nfs.sh [-M--] 19 L:[ 1* 8 9/ 9] *(275 / 275b) <EOF>
#!/bin/bash
echo "Provisioning script $@"
echo "Install needed packages"
dnf -y install nfs-utils
echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.an spelov.net:/srv/nfs /mnt/nfs
echo "server.an spelov.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc

```

Рис. 2.36: Открытие файла на редактирование и добавление скрипта.

Для отработки созданных скриптов во время загрузки виртуальных машин server и client в конфигурационном файле Vagrantfile добавим в соответствующих разделах конфигураций для сервера и клиента(рис. 2.37).

```

    preserve_order: true
    path: "provision/server/ntp.sh"

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

end

## Client configuration
config.vm.define "client", autostart: false do |client|
  client.vm.box = "rockylinux10"
  client.vm.hostname = 'client'

  client.vm.boot_timeout = 1440

  client.ssh.insert_key = false
  client.ssh.username = 'vagrant'
  client.ssh.password = 'vagrant'

  client.vm.network :private_network,
    ip: "192.168.1.2",
    virtualbox_intnet: true

  client.vm.provider :virtualbox do |virtualbox|
    virtualbox.customize ["modifyvm", :id, "--vnde", "on"]
    virtualbox.customize ["modifyvm", :id, "--vdeport", "3392"]
  end

  client.vm.provision "client dummy",
    type: "shell",
    preserve_order: true,
    path: "provision/client/01-dummy.sh"

  client.vm.provision "client routing",
    type: "shell",
    preserve_order: true,
    run: "always",
    path: "provision/client/01-routing.sh"

  client.vm.provision "client routing",
    type: "shell",
    preserve_order: true,
    run: "always",
    path: "provision/client/01-routing.sh"

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

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

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

end
end

```

Рис. 2.37: Добавление записи в конфигурационном файле Vagrantfile для сервера и для клиента.

3 Выводы

В ходе выполнения лабораторной работы были приобретены навыки настройки сервера NFS для удалённого доступа к ресурсам.

4 Ответы на контрольные вопросы:

1. Как называется файл конфигурации, содержащий общие ресурсы NFS?
- Файл конфигурации, содержащий общие ресурсы NFS, называется /etc(exports. В этом файле определяются каталоги, которые будут доступны для общего использования через NFS.
2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS? - Для обеспечения полного доступа к серверу NFS, обычно открываются следующие порты: TCP и UDP порт 2049: Основной порт для NFS. TCP и UDP порт 111: Порт для службы rpcbind (или портmap), которая используется для регистрации служб RPC. Порты для динамически выделяемых портов (обычно в диапазоне 32768-32779), используемых NFS для передачи данных.
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 загрузке системы, в файле /etc/fstab следует использовать опцию auto. Пример строки в /etc/fstab: server:/remote/export /mnt/nfs_share nfs auto 0 0 Здесь server:/remote/export - адрес и путь к общему ресурсу NFS, /mnt/nfs_share - точка монтирования, nfs - тип файловой системы, auto - опция для автоматического монтирования при загрузке системы, 0 0 - опции для процесса проверки файловой системы при загрузке.

Список литературы