

Отчёт по лабораторной работе №13

Дисциплина: Администрирование сетевых подсистем

Мишина Анастасия Алексеевна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
2.1 Настройка сервера NFSv4	6
2.2 Монтирование NFS на клиенте	9
2.3 Подключение каталогов к дереву NFS	12
2.4 Подключение каталогов для работы пользователей	15
2.5 Внесение изменений в настройки внутреннего окружения виртуальных машины	18
3 Контрольные вопросы	21
4 Выводы	22

Список иллюстраций

2.1	Подключение через NFS каталога только на чтение	6
2.2	Запуск NFS-сервера	7
2.3	Просмотр подмонтированных удалённых ресурсов на клиенте	7
2.4	Просмотр подмонтированных удалённых ресурсов на клиенте	8
2.5	Просмотр задействованных при удалённом монтировании служб	8
2.6	Просмотр задействованных при удалённом монтировании служб	8
2.7	Добавление служб rpc-bind и mountd в настройки межсетевого экрана	9
2.8	Проверка подключения удалённого ресурса на клиенте . .	9
2.9	Проверка правильности подключения общего ресурса NFS	10
2.10	Добавление записи в файл /etc/fstab на клиенте	11
2.11	Проверка наличия автоматического монтирования удалённых ресурсов при запуске ОС	11
2.12	Проверка автоматического подключения удалённого ресурса	12
2.13	Проверка содержимого /srv/nfs	13
2.14	Проверка содержимого /mnt/nfs	13
2.15	Добавление в файл /etc(exports экспорт каталога веб-сервера	13
2.16	Проверка содержимого /mnt/nfs	14
2.17	Добавление записи в файл /etc/fstab	14
2.18	Проверка содержимого /mnt/nfs	15
2.19	Проверка прав доступа на каталог	16
2.20	Подключение каталога пользователя в файле /etc(exports .	16
2.21	Добавление записи в файл /etc/fstab	17
2.22	Проверка содержимого /mnt/nfs	17
2.23	Создание файла в каталоге /mnt/nfs/home/aamishina . .	18
2.24	Изменения на сервере	18
2.25	Скрипт файла /vagrant/provision/server/nfs.sh	19
2.26	Скрипт файла /vagrant/provision/client/nfs.sh	20

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

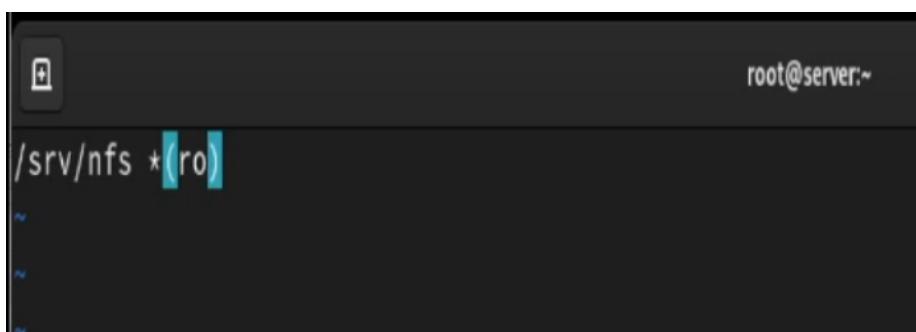
Запустим виртуальную машину server. На сервере установим необходимое программное обеспечение:

```
dnf -y install nfs-utils
```

Затем создадим каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS):

```
mkdir -p /srv/nfs
```

В файле /etc(exports пропишем подключаемый через NFS общий каталог с доступом только на чтение(рис. fig. 2.1):



```
/srv/nfs *(ro)
```

Рис. 2.1: Подключение через NFS каталога только на чтение

Для общего каталога зададим контекст безопасности NFS и применим изменённую настройку SELinux к файловой системе, затем запустим сервер NFS и настроим межсетевой экран для работы сервера NFS(рис. fig. 2.2):

```
[root@server.aamishina.net ~]# mkdir -p /srv/nfs
[root@server.aamishina.net ~]# vim /etc/exports
[root@server.aamishina.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.aamishina.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.aamishina.net ~]# systemctl start nfs-server.service
[root@server.aamishina.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.aamishina.net ~]# firewall-cmd --add-service=nfs
success
[root@server.aamishina.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.aamishina.net ~]# firewall-cmd --reload
success
[root@server.aamishina.net ~]#
```

Рис. 2.2: Запуск NFS-сервера

На клиенте установим необходимое для работы NFS программное обеспечение:

```
dnf -y install nfs-utils
```

Попробуем посмотреть имеющиеся подмонтированные удалённые ресурсы(fig. 2.3):

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

Рис. 2.3: Просмотр подмонтированных удалённых ресурсов на клиенте

Увидим, что протокол удалённого вызова процедур (RPC) не может получить доступ к запрашиваемым ресурсам.

На сервере остановим сервис межсетевого экрана с помощью команды `systemctl stop firewalld.service` и на клиенте вновь попробуем подключиться к удалённо смонтированному ресурсу(fig. 2.4):

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

Рис. 2.4: Просмотр подмонтированных удалённых ресурсов на клиенте

Теперь подключение к удаленно смонтированному ресурсу успешно произведено.

На сервере запустим сервис межсетевого экрана и посмотрим, какие службы задействованы при удалённом монтировании (fig. 2.5, fig. 2.6):

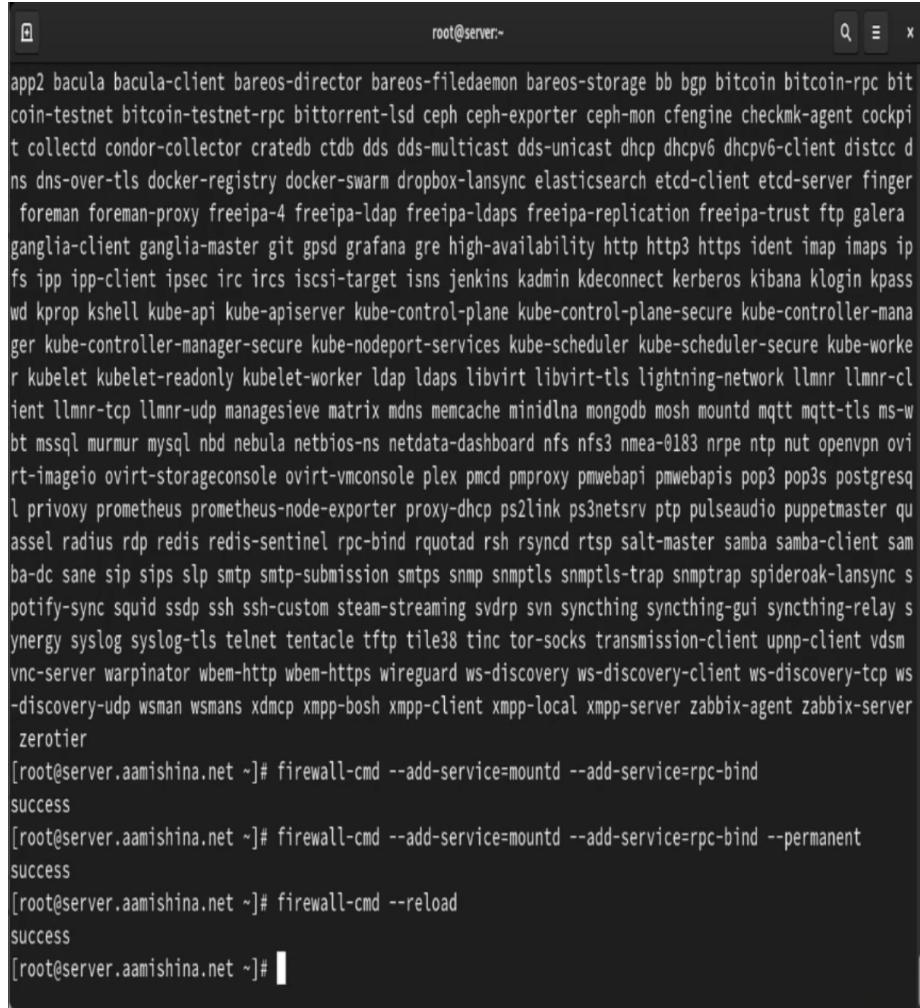
```
httpd 1251 1456 httpd apache 4u IPv6 22825 0t0 TCP *:http (LISTEN)
httpd 1251 1456 httpd apache 5u sock 0,8 0t0 22836 protocol: TCP
httpd 1251 1456 httpd apache 6u IPv6 22837 0t0 TCP *:https (LISTEN)
httpd 1251 1456 httpd apache 23u sock 0,8 0t0 23100 protocol: TCP
rpcbind 12895 rpc 4u IPv4 48625 0t0 TCP *:sunrpc (LISTEN)
rpcbind 12895 rpc 6u IPv6 48643 0t0 TCP *:sunrpc (LISTEN)
rpc.statd 12896 rpcuser 9u IPv4 54348 0t0 TCP *:58275 (LISTEN)
rpc.statd 12896 rpcuser 11u IPv6 54358 0t0 TCP *:59917 (LISTEN)
rpc.mount 12904 root 5u IPv4 55291 0t0 TCP *:mountd (LISTEN)
rpc.mount 12904 root 7u IPv6 55297 0t0 TCP *:mountd (LISTEN)
```

Рис. 2.5: Просмотр задействованных при удалённом монтировании служб

```
tcp->_gateway:bootps NetworkMa 7373 7397 gmain root 27u IPv4 39184 0t0 UDP server.aamishina.net:boo
tcp->_gateway:bootps NetworkMa 7373 7398 gibus root 27u IPv4 39184 0t0 UDP server.aamishina.net:boo
tcp->_gateway:bootps rpcbind 12895 rpc 5u IPv4 48634 0t0 UDP *:sunrpc
rpcbind 12895 rpc 7u IPv6 48652 0t0 UDP *:sunrpc
rpc.statd 12896 rpcuser 7u IPv4 54339 0t0 UDP *:41319
rpc.statd 12896 rpcuser 8u IPv4 54324 0t0 UDP localhost:778
rpc.statd 12896 rpcuser 10u IPv6 54352 0t0 UDP *:50818
rpc.mount 12904 root 4u IPv4 55288 0t0 UDP *:mountd
rpc.mount 12904 root 6u IPv6 55294 0t0 UDP *:mountd
```

Рис. 2.6: Просмотр задействованных при удалённом монтировании служб

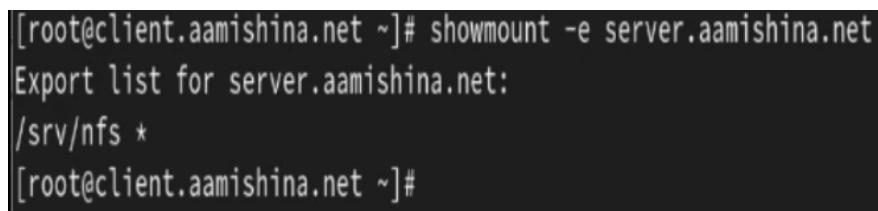
Затем добавим службы rpc-bind и mountd в настройки межсетевого экрана на сервере (fig. 2.7):



```
root@server:~ [ ]  
app2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bit  
coin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpcv6 dhcpcv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpgsql grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpass kwd kprop kshell kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmppts snmptrap spiderOak-lansync spotify-sync squid ssdp ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdserv vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier  
[root@server.aamishina.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind  
success  
[root@server.aamishina.net ~]# firewall-cmd --add-service=mountd --add-service=rpc-bind --permanent  
success  
[root@server.aamishina.net ~]# firewall-cmd --reload  
success  
[root@server.aamishina.net ~]#
```

Рис. 2.7: Добавление служб rpc-bind и mountd в настройки межсетевого экрана

Теперь на клиенте проверим подключение удалённого ресурса(fig. 2.8):



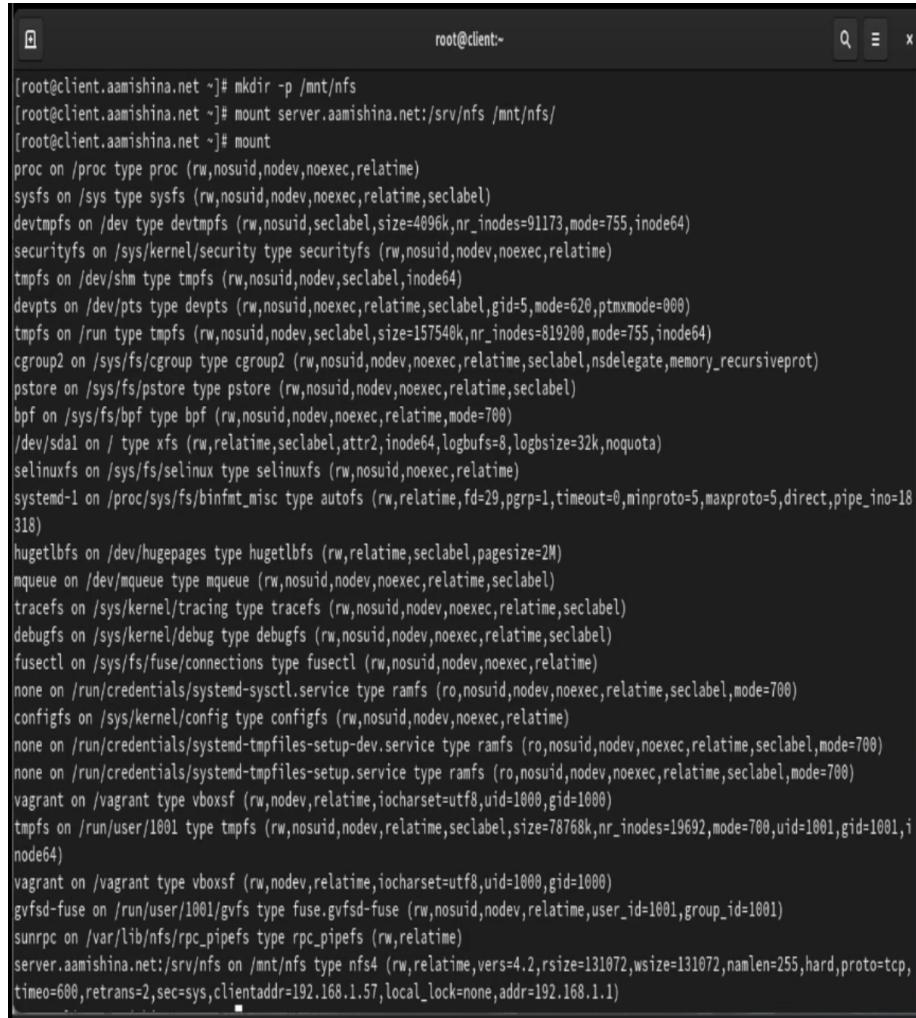
```
[root@client.aamishina.net ~]# showmount -e server.aamishina.net  
Export list for server.aamishina.net:  
/srv/nfs *
```

Рис. 2.8: Проверка подключения удалённого ресурса на клиенте

2.2 Монтирование NFS на клиенте

На клиенте создадим каталог, в который будет монтироваться удалённый ресурс, и подмонтируем дерево NFS. Затем проверим, что общий

ресурс NFS подключён правильно(fig. 2.9):



```
[root@client.aamishina.net ~]# mkdir -p /mnt/nfs
[root@client.aamishina.net ~]# mount server.aamishina.net:/srv/nfs /mnt/nfs/
[root@client.aamishina.net ~]# 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=91173,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=157540k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
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)
/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,maxproto=5,direct,pipe_ino=18318)
hugepages on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
configs on /sys/kernel/config type configs (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,seclabel,mode=700)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=78768k,nr_inodes=19692,mode=700,uid=1001,gid=1001,inode64)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1001/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1001,group_id=1001)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
server.aamishina.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=192.168.1.57,local_lock=none,addr=192.168.1.1)
```

Рис. 2.9: Проверка правильности подключения общего ресурса NFS

На клиенте в конце файла `/etc/fstab` добавьте следующую запись(fig. 2.10):

```

#
# /etc/fstab
# Created by anaconda on Wed Sep 18 16:20:42 2024
#
# 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=707fdfff6-f8d1-4415-8394-9da35f1e5962 /           xfs    defaults    0 0
/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.aamishina.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
-

```

Рис. 2.10: Добавление записи в файл /etc/fstab на клиенте

Мы подключаем каталоги сервера для автоматического монтирования при загрузке через nfs, для этого указываются каталоги и `_netdev` – опция, указывающая, что данный ресурс является сетевым(то есть монтируться каталоги будут после загрузки сетевой подсистемы).

На клиенте проверьте наличие автоматического монтирования удалённых ресурсов при запуске операционной системы(рис. fig. 2.11):

```

[root@client.aamishina.net ~]# vim /etc/fstab
[root@client.aamishina.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 Sun 2024-10-27 14:58:08 UTC; 29min ago
      Until: Sun 2024-10-27 14:58:08 UTC; 29min ago
        Docs: man:systemd.special(7)

Oct 27 14:58:08 client.aamishina.net systemd[1]: Reached target Remote File Systems.
[root@client.aamishina.net ~]#

```

Рис. 2.11: Проверка наличия автоматического монтирования удалённых ресурсов при запуске ОС

Перезапустим клиента и убедимся, что удалённый ресурс подключается автоматически(рис. fig. 2.12):

```
[aamishina@client.aamishina.net ~]$ 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=91173,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=157540k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstree on /sys/fs/pstore type pstree (rw,nosuid,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/sda1 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,pgroup=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18314)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
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)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-svcctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
configfs on /sys/kernel/config type configfs (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,seclabel,mode=700)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
server.aamishina.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=192.168.1.57,local_lock=none,addr=192.168.1.1,_netdev)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=78768k,nr_inodes=19692,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_id=1001)
vagrant on /vagrant type vboxsf (rw,nodev,relatime,iocharset=utf8,uid=1000,gid=1000)
[aamishina@client.aamishina.net ~]$ systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
```

Рис. 2.12: Проверка автоматического подключения удалённого ресурса

2.3 Подключение каталогов к дереву NFS

На сервере создадим общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера и подмонтируем каталог web-сервера:

```
mkdir -p /srv/nfs/www
mount -o bind /var/www/ /srv/nfs/www/
```

На сервере проверим, что отображается в каталоге /srv/nfs (fig. 2.13)

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

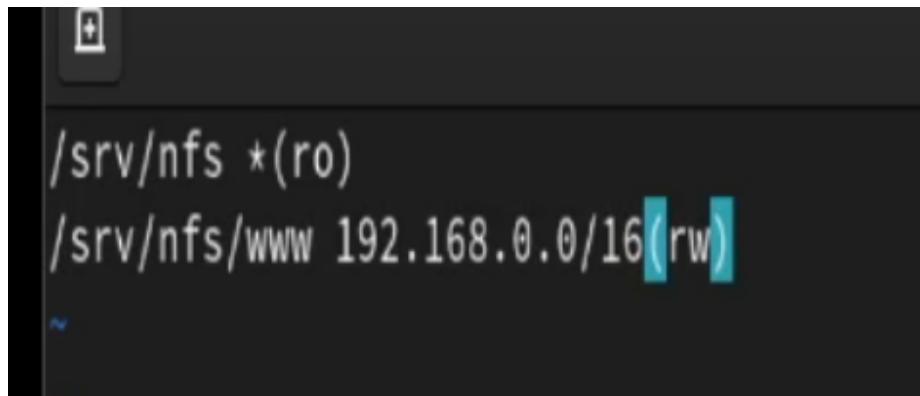
Рис. 2.13: Проверка содержимого /srv/nfs

На сервере проверим, что отображается в каталоге /mnt/nfs(fig. 2.14)

```
[aamishina@client.aamishina.net ~]$ ls /mnt/nfs  
www  
[aamishina@client.aamishina.net ~]$
```

Рис. 2.14: Проверка содержимого /mnt/nfs

На сервере в файле /etc/exports добавьте экспорт каталога веб-сервера с удалённого ресурса(fig. 2.15)



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

Рис. 2.15: Добавление в файл /etc/exports экспорт каталога веб-сервера

Экспортируем все каталоги, упомянутые в файле /etc/exports командной `exportfs -r`. Проверим на клиенте каталог /mnt/nfs(fig. 2.16)

```
[aamishina@client.aamishina.net ~]$ ls /mnt/nfs
```

```
www
```

```
[aamishina@client.aamishina.net ~]$
```

Рис. 2.16: Проверка содержимого /mnt/nfs

На сервере в конце файла /etc/fstab добавим следующую запись(fig. 2.17)

```
#  
# /etc/fstab  
# Created by anaconda on Wed Sep 18 16:20:42 2024  
#  
# 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=707fdff6-f8d1-4415-8394-9da35fle5962 / xfs defaults 0 0  
/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
```

Рис. 2.17: Добавление записи в файл /etc/fstab

Повторно экспортируем каталоги, указанные в файле /etc/exports и на клиенте проверим каталог /mnt/nfs(fig. 2.18)

```
[aamishina@client.aamishina.net ~]$ ls /mnt/nfs
```

```
www
```

```
[aamishina@client.aamishina.net ~]$ █
```

Рис. 2.18: Проверка содержимого /mnt/nfs

2.4 Подключение каталогов для работы пользователей

На сервере под пользователем aamishina в домашнем каталоге создадим каталог common с полными правами доступа только для этого пользователя, а в нём файл aamishina@server.txt:

```
mkdir -p -m 700 ~/common  
cd ~/common  
touch aamishina@server.txt
```

На сервере создадим общий каталог для работы пользователя aamishina по сети, подмонтируем каталог common пользователя aamishina в NFS. Затем посмотрим какие права доступа установлены на этот каталог (fig. 2.19):

```
[aamishina@server.aamishina.net ~]$ mkdir -p -m 700 ~/common
[aamishina@server.aamishina.net ~]$ cd ~/common
[aamishina@server.aamishina.net common]$ touch aamishina@server.txt
[aamishina@server.aamishina.net common]$ mkdir -p /srv/nfs/home/aamishina
mkdir: cannot create directory '/srv/nfs/home': Permission denied
[aamishina@server.aamishina.net common]$ sudo -i
[sudo] password for aamishina:
[root@server.aamishina.net ~]# mkdir -p /srv/nfs/home/aamishina
[root@server.aamishina.net ~]# mount -o bind /home/aamishina/common /srv/nfs/home/aamishina
mount: (hint) your fstab has been modified, but systemd still uses
      the old version; use 'systemctl daemon-reload' to reload.
[root@server.aamishina.net ~]# ls -l /srv/nfs/home/
total 0
drwx----- 2 aamishina aamishina 34 Oct 27 15:38 aamishina
[root@server.aamishina.net ~]#
```

Рис. 2.19: Проверка прав доступа на каталог

Подключим каталог пользователя в файле `/etc/exports`, прописав в нём(fig. 2.20):

```
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/aamishina 192.168.0.0/16(rw)
```

Рис. 2.20: Подключение каталога пользователя в файле `/etc/exports`

Внесем изменения в файл `/etc/fstab`(fig. 2.21):

```
#  
# /etc/fstab  
# Created by anaconda on Wed Sep 18 16:20:42 2024  
#  
# 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=707fdff6-f8d1-4415-8394-9da35f1e5962 / xfs defaults 0 0  
/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/aamishina/common /srv/nfs/home/aamishina none bind 0 0  
~
```

Рис. 2.21: Добавление записи в файл /etc/fstab

Затем повторно экспортируем каталоги и проверим на клиенте каталог mnt/nfs(fig. 2.22):

```
[aamishina@client.aamishina.net ~]$ ls /mnt/nfs  
home www  
[aamishina@client.aamishina.net ~]$
```

Рис. 2.22: Проверка содержимого /mnt/nfs

На клиенте попробуем создать файл в каталоге /mnt/nfs/home/aamishina под пользователями aamishina и root (fig. 2.23):

```
[aamishina@client.aamishina.net ~]$ cd /mnt/nfs/home/aamishina
[aamishina@client.aamishina.net aamishina]$ touch aamishina@client.txt
[aamishina@client.aamishina.net aamishina]$ ls
aamishina@client.txt aamishina@server.txt
[aamishina@client.aamishina.net aamishina]$ cd
[aamishina@client.aamishina.net ~]$ sudo -i
[sudo] password for aamishina:
[root@client.aamishina.net ~]# cd /mnt/nfs/home/aamishina
-bash: cd: /mnt/nfs/home/aamishina: Permission denied
[root@client.aamishina.net ~]# exit
logout
[aamishina@client.aamishina.net ~]$
```

Рис. 2.23: Создание файла в каталоге /mnt/nfs/home/aamishina

У пользователя root нет прав на переход в эту директорию, пользователь aamishina может и перейти, и создать файл.

Проверим появились ли изменения на сервере(fig. 2.24):

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

Рис. 2.24: Изменения на сервере

Действительно, созданные файлы отразились на сервере.

2.5 Внесение изменений в настройки внутреннего окружения виртуальных машины

На виртуальной машине server перейдем 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 и внесем скрипт(fig. 2.25):

```
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

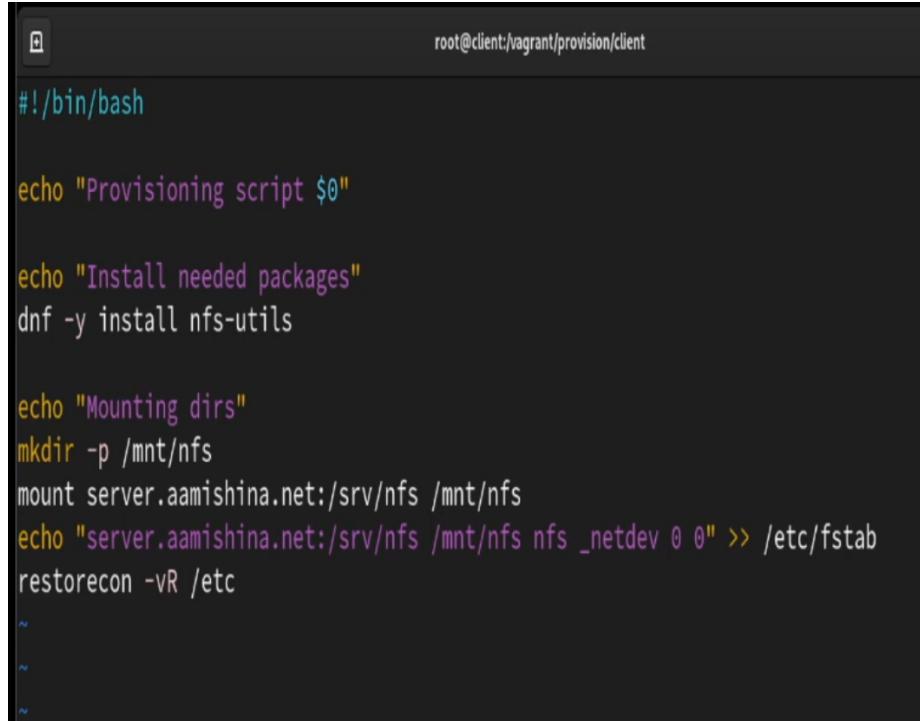
echo "Mounting dirs"
mkdir -p /srv/nfs/www
mount -o bind /var/www /srv/nfs/www
echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
mkdir -p /srv/nfs/home/aamishina
mkdir -p -m 700 /home/aamishina/common
chown aamishina:aamishina /home/aamishina/common
mount -o bind /home/aamishina/common /srv/nfs/home/aamishina
echo "/home/aamishina/common /srv/nfs/home/aamishina none bind 0 0" >> /etc/fstab

echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server

systemctl restart firewalld
:wq
```

Рис. 2.25: Скрипт файла /vagrant/provision/server/nfs.sh

В каталоге /vagrant/provision/client создадим исполняемый файл nfs.sh и внесем скрипт(fig. 2.26):



```
root@client:/vagrant/provision/client
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install nfs-utils

echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.aamishina.net:/srv/nfs /mnt/nfs
echo "server.aamishina.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >> /etc/fstab
restorecon -vR /etc
```

Рис. 2.26: Скрипт файла /vagrant/provision/client/nfs.sh

Затем для отработки созданных скриптов в конфигурационном файле Vagrantfile необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

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

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

3 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`.

4 Выводы

В результате выполнения данной работы были приобретены практические навыки настройки сервера NFS для удалённого доступа к ресурсам.