

Лабораторная работа №13

Настройка NFS

Ромицына Анастасия Романовна

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 6 |
| 2 | Выполнение лабораторной работы | 7 |
| 3 | Выводы | 22 |
| 4 | Ответы на контрольные вопросы: | 23 |
| | Список литературы | 24 |

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. | 10 |
| 2.8 | Повторная попытка подключения к удалённо смонтированному ресурсу. | 10 |
| 2.9 | Запуск на сервере сервиса межсетевого экрана. | 10 |
| 2.10 | Просмотр на сервере служб, задействованных при удалённом монтировании (TCP). | 11 |
| 2.11 | Просмотр на сервере служб, задействованных при удалённом монтировании (UDP). | 12 |
| 2.12 | Добавление службы rpc-bind и mountd в настройки межсетевого экрана на сервере. | 13 |
| 2.13 | Проверка на клиенте подключения удалённого ресурса. | 13 |
| 2.14 | Создание на клиенте каталога, в который будет монтироваться удалённый ресурс, и монтироваться дерево NFS. Проверка, что общий ресурс NFS подключён правильно. | 14 |
| 2.15 | Добавление на клиенте в конце файла /etc/fstab записи. | 14 |
| 2.16 | Проверка на клиенте наличия автоматического монтирования удалённых ресурсов при запуске операционной системы. | 14 |
| 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. | 15 |
| 2.18 | Просмотр на клиенте содержимого каталога /mnt/nfs. | 15 |
| 2.19 | Добавление на сервере в файле /etc/exports экспорта каталога веб-сервера с удалённого ресурса. | 15 |
| 2.20 | Экспорт всех каталогов, упомянутых в файле /etc/exports. | 15 |
| 2.21 | Проверка на клиенте каталога /mnt/nfs. | 16 |

| | | |
|------|--|----|
| 2.22 | Добавление на сервере в конце файла <code>/etc/fstab</code> записи. | 16 |
| 2.23 | Повторный экспорт каталогов, указанных в файле <code>/etc/exports</code> . . . | 16 |
| 2.24 | Проверка на клиенте каталога <code>/mnt/nfs</code> | 16 |
| 2.25 | Создание на сервере под пользователем <code>arromichina</code> в домашнем каталоге каталога <code>common</code> с полными правами доступа только для этого пользователя, а в нём файл <code>arromichina@server.txt</code> . Создание общего каталога для работы пользователя <code>arromichina</code> по сети и монтирование каталога <code>common</code> пользователя <code>arromichina</code> в NFS. | 17 |
| 2.26 | Подключение каталога пользователя в файле <code>/etc/exports</code> | 17 |
| 2.27 | Внесение изменения в файл <code>/etc/fstab</code> | 17 |
| 2.28 | Повторный экспорт каталогов. | 17 |
| 2.29 | Проверка на клиенте каталога <code>/mnt/nfs</code> | 18 |
| 2.30 | Переход на клиенте под пользователем <code>arromichina</code> в каталог <code>/mnt/nfs/home/arromichina</code> и попытка создать в нём файл <code>arromichina@client.txt</code> | 18 |
| 2.31 | Переход на клиенте под пользователем <code>root</code> в каталог <code>/mnt/nfs/home/arromichina</code> и попытка создать в нём файл <code>arromichina@client.txt</code> | 18 |
| 2.32 | Просмотр наличия изменений на сервере в каталоге пользователя <code>/home/arromichina/common</code> | 18 |
| 2.33 | Переход на виртуальной машине <code>server</code> в каталог для внесения изменений в настройки внутреннего окружения <code>/vagrant/provision/server/</code> , создание в нём каталога <code>nfs</code> , в который помещаем в соответствующие подкаталоги конфигурационные файлы. Создание в каталоге <code>/vagrant/provision/server</code> исполняемого файла <code>nfs.sh</code> | 19 |
| 2.34 | Открытие файла на редактирование и добавление скрипта. | 20 |
| 2.35 | Переход на виртуальной машине <code>client</code> в каталог для внесения изменений в настройки внутреннего окружения <code>/vagrant/provision/client/</code> . Создание в каталоге <code>/vagrant/provision/client</code> исполняемого файла <code>nfs.sh</code> | 20 |
| 2.36 | Открытие файла на редактирование и добавление скрипта. | 21 |
| 2.37 | Добавление записи в конфигурационном файле <code>Vagrantfile</code> для сервера и для клиента. | 21 |

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).

```
root@server:~# sudo -i

libsss_idmap      x86_64      2.11.1-2.el10_1.1  baseos
libsss_nss_idmap  x86_64      2.11.1-2.el10_1.1  baseos
libsss_sudo       x86_64      2.11.1-2.el10_1.1  baseos
libtalloc         x86_64      2.4.3-100.el10     baseos
libtdb            x86_64      1.4.13-100.el10    baseos
libtevent         x86_64      0.16.2-100.el10    baseos
libwbclient       x86_64      4.22.4-106.el10    baseos
samba-client-libs x86_64      4.22.4-106.el10    baseos
samba-common      noarch      4.22.4-106.el10    baseos
samba-common-libs x86_64      4.22.4-106.el10    baseos
sssd              x86_64      2.11.1-2.el10_1.1  baseos
sssd-ad           x86_64      2.11.1-2.el10_1.1  baseos
sssd-client       x86_64      2.11.1-2.el10_1.1  baseos
sssd-common       x86_64      2.11.1-2.el10_1.1  baseos
sssd-common-pac   x86_64      2.11.1-2.el10_1.1  baseos
sssd-ipa          x86_64      2.11.1-2.el10_1.1  baseos
sssd-kcm          x86_64      2.11.1-2.el10_1.1  baseos
sssd-krb5         x86_64      2.11.1-2.el10_1.1  baseos
sssd-krb5-common  x86_64      2.11.1-2.el10_1.1  baseos
sssd-ldap         x86_64      2.11.1-2.el10_1.1  baseos
sssd-proxy        x86_64      2.11.1-2.el10_1.1  baseos

Installing dependencies:
gssproxy          x86_64      0.9.2-10.el10      baseos
libev             x86_64      4.33-14.el10       baseos
libnfsidmap       x86_64      1:2.8.3-0.el10     baseos
libverto-libev    x86_64      0.3.2-10.el10      baseos
rpcbind           x86_64      1.2.7-3.el10       baseos
sssd-nfs-idmap    x86_64      2.11.1-2.el10_1.1  baseos

Transaction Summary
-----
Install  7 Packages
Upgrade 25 Packages

Total download size: 9.7 M
Downloading Packages:
[1/32]: gssproxy-0 0% [          ] --- B/s | 0 B  ---
```

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.arromichina.net ~]# mkdir -p /srv/nfs
[root@server.arromichina.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).

```
exports      [-M--] 14 L:[ 1+ 0 1/ 1] *(14 / 14b) <EOF>
/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.arromichina.net ~]# semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
[root@server.arromichina.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.arromichina.net ~]# systemctl start nfs-server.service
[root@server.arromichina.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.arromichina.net ~]# firewall-cmd --add-service=nfs
success
[root@server.arromichina.net ~]# firewall-cmd --add-service=nfs --permanent
success
[root@server.arromichina.net ~]# firewall-cmd --reload
```

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

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): libev-4.33-14.el10.x86_64.rpm      204 kB/s | 52 kB      00:00
(2/32): libnfsidmap-2.8.3-0.el10.x86_64.rpm 221 kB/s | 61 kB      00:00
(3/32): gssproxy-0.9.2-10.el10.x86_64.rpm  354 kB/s | 111 kB     00:00
(4/32): libverto-libev-0.3.2-10.el10.x86_64 196 kB/s | 13 kB      00:00
(5/32): sssd-nfs-idmap-2.11.1-2.el10_1.1.x8 505 kB/s | 36 kB      00:00
(6/32): rpcbind-1.2.7-3.el10.x86_64.rpm     645 kB/s | 57 kB      00:00
(7/32): libipa_hbac-2.11.1-2.el10_1.1.x86_6 427 kB/s | 34 kB      00:00
(8/32): nfs-utils-2.8.3-0.el10.x86_64.rpm   1.7 MB/s | 475 kB     00:00
(9/32): libsmbclient-4.22.4-106.el10.x86_64 858 kB/s | 75 kB      00:00
(10/32): libldb-4.22.4-106.el10.x86_64.rpm  1.1 MB/s | 181 kB     00:00
(11/32): libsss_idmap-2.11.1-2.el10_1.1.x86 626 kB/s | 41 kB      00:00
(12/32): libsss_certmap-2.11.1-2.el10_1.1.x 945 kB/s | 81 kB      00:00
(13/32): libsss_nss_idmap-2.11.1-2.el10_1.1 594 kB/s | 44 kB      00:00
(14/32): libsss_sudo-2.11.1-2.el10_1.1.x86_ 478 kB/s | 33 kB      00:00
(15/32): libtalloc-2.4.3-100.el10.x86_64.rp 496 kB/s | 33 kB      00:00
(16/32): libtdb-1.4.13-100.el10.x86_64.rpm  758 kB/s | 55 kB      00:00
(17/32): libtevent-0.16.2-100.el10.x86_64.r 753 kB/s | 50 kB      00:00
(18/32): libwbclient-4.22.4-106.el10.x86_64 622 kB/s | 43 kB      00:00
(19/32): samba-common-libs-4.22.4-106.el10.  1.0 MB/s | 104 kB     00:00
(20/32): samba-common-4.22.4-106.el10.noarc  1.2 MB/s | 174 kB     00:00
(21/32): sssd-2.11.1-2.el10_1.1.x86_64.rpm  231 kB/s | 25 kB      00:00
(22/32): sssd-ad-2.11.1-2.el10_1.1.x86_64.r 1.5 MB/s | 195 kB     00:00
(23-24/32): samba- 34% [*****] 1.1 MB/s | 3.4 MB 00:05 ETA
```

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

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

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

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

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

```
[root@client.arromichina.net ~]# systemctl stop firewalld.service
[root@client.arromichina.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.arromichina.net ~]# showmount -e server.arromichina.net
Export list for server.arromichina.net:
/srv/nfs *
[root@client.arromichina.net ~]#
```

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

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

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

Рис. 2.9: Запуск на сервере сервиса межсетевого экрана.

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

| root@server:~ - sudo -i | | | | | | |
|-------------------------|------------|---------|-----------------------|-----|------|---|
| 2116 | 0t0 | TCP | *:mysql (LISTEN) | | | |
| mariadb | 1369 10696 | mariadb | mysql | 22u | IPv6 | 1 |
| 2117 | 0t0 | TCP | *:mysql (LISTEN) | | | |
| dovecot | 1489 | | root | 22u | IPv4 | 1 |
| 2706 | 0t0 | TCP | *:pop3 (LISTEN) | | | |
| dovecot | 1489 | | root | 23u | IPv6 | 1 |
| 2707 | 0t0 | TCP | *:pop3 (LISTEN) | | | |
| dovecot | 1489 | | root | 24u | IPv4 | 1 |
| 2708 | 0t0 | TCP | *:pop3s (LISTEN) | | | |
| dovecot | 1489 | | root | 25u | IPv6 | 1 |
| 2709 | 0t0 | TCP | *:pop3s (LISTEN) | | | |
| dovecot | 1489 | | root | 41u | IPv4 | 1 |
| 2725 | 0t0 | TCP | *:imap (LISTEN) | | | |
| dovecot | 1489 | | root | 42u | IPv6 | 1 |
| 2726 | 0t0 | TCP | *:imap (LISTEN) | | | |
| dovecot | 1489 | | root | 43u | IPv4 | 1 |
| 2727 | 0t0 | TCP | *:imaps (LISTEN) | | | |
| dovecot | 1489 | | root | 44u | IPv6 | 1 |
| 2728 | 0t0 | TCP | *:imaps (LISTEN) | | | |
| master | 1520 | | root | 13u | IPv4 | 1 |
| 2509 | 0t0 | TCP | *:smtp (LISTEN) | | | |
| master | 1520 | | root | 17u | IPv4 | 1 |
| 2512 | 0t0 | TCP | *:submission (LISTEN) | | | |
| nginx | 1616 | | root | 6u | IPv4 | 1 |
| 3020 | 0t0 | TCP | *:http (LISTEN) | | | |
| nginx | 1616 | | root | 7u | IPv6 | 1 |
| 3021 | 0t0 | TCP | *:http (LISTEN) | | | |
| nginx | 1617 | | nginx | 6u | IPv4 | 1 |
| 3020 | 0t0 | TCP | *:http (LISTEN) | | | |
| nginx | 1617 | | nginx | 7u | IPv6 | 1 |
| 3021 | 0t0 | TCP | *:http (LISTEN) | | | |
| nginx | 1620 | | nginx | 6u | IPv4 | 1 |
| 3020 | 0t0 | TCP | *:http (LISTEN) | | | |
| nginx | 1620 | | nginx | 7u | IPv6 | 1 |
| 3021 | 0t0 | TCP | *:http (LISTEN) | | | |

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

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

| root@server:~ - sudo -i | | | | | | |
|-------------------------|------|------|--|------|-----|--------|
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1606 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1606 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1610 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1610 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1611 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1611 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1612 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1612 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1658 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1658 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| kea-dhcp4 | 1492 | 1659 | kea-dhcp4 | kea | 18u | IPv4 1 |
| 1980 | 0t0 | UDP | *:52542 | | | |
| kea-dhcp4 | 1492 | 1659 | kea-dhcp4 | kea | 21u | IPv4 1 |
| 1983 | 0t0 | UDP | mail.arromichina.net:bootps | | | |
| NetworkMa | 4808 | | | root | 31u | IPv4 2 |
| 1306 | 0t0 | UDP | server.arromichina.net:bootpc->_gateway:bootps | | | |
| NetworkMa | 4808 | 4820 | gmain | root | 31u | IPv4 2 |
| 1306 | 0t0 | UDP | server.arromichina.net:bootpc->_gateway:bootps | | | |
| NetworkMa | 4808 | 4821 | pool-spaw | root | 31u | IPv4 2 |
| 1306 | 0t0 | UDP | server.arromichina.net:bootpc->_gateway:bootps | | | |
| NetworkMa | 4808 | 4822 | gdbus | root | 31u | IPv4 2 |
| 1306 | 0t0 | UDP | server.arromichina.net:bootpc->_gateway:bootps | | | |

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

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

```
root@server:~ - sudo -i
ry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server factor
io finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa
-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd
grafana gre high-availability http http3 https ident imap imaps iperf2 iper
f3 ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdec
nnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver ku
be-control-plane kube-control-plane-secure kube-controller-manager kube-cont
roller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-s
ecure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt
libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp manage
sieve matrix mdns memcache minecraft minidlna mndp mongodb mosh mountd mpd m
qtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula need-for-speed-most-wanted
netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry
openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pncd pmproxy
pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node
-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel r
adius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp
salt-master samba samba-client samba-dc sane settlers-history-collection si
p sips slimevr slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmp
trap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom statsrv steam
lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submis
sion supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay synergy
syscomlan syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-so
cks transmission-client turn turns upnp-client vdsu vnc-server vrrp warpinat
or wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discov
ery-host ws-discovery-tcp ws-discovery-udp wsdd wsdd-http wsman wsmans xdmcp
xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gatew
ay zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.arromichina.net ~]# firewall-cmd --add-service=mountd --add-ser
vice=rpc-bind
success
[root@server.arromichina.net ~]# firewall-cmd --add-service=mountd --add-ser
vice=rpc-bind --permanent
success
[root@server.arromichina.net ~]# firewall-cmd --reload
success
[root@server.arromichina.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.arromichina.net ~]# showmount -e server.arromichina.net
Export list for server.arromichina.net:
/srv/nfs *
[root@client.arromichina.net ~]#
```

Рис. 2.13: Проверка на клиенте подключения удалённого ресурса.

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

```
[root@client.arromichina.net ~]# mkdir -p /mnt/nfs
[root@client.arromichina.net ~]# mount server.arromichina.net:/srv/nfs /mnt/
nfs
[root@client.arromichina.net ~]# mount
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbuf
s=8,logbsize=32k,noquota)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=2111
73,mode=755,inode64)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mod
e=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,r
elative)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,secl
abel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclab
el)
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,relatim
e)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=345392k,nr_inodes=81
9200,mode=755,inode64)
```

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

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

```
# The contents below are automatically generated by Vagrant. Do not modify.
vagrant /vagrant vboxsf uid=1000,gid=1000,_netdev 0 0
#VAGRANT-END
server.arromichina.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.arromichina.net ~]# systemctl status remote-fs.target
● remote-fs.target - Remote File Systems
   Loaded: loaded (/usr/lib/systemd/system/remote-fs.target; enabled; pre
   Active: active since Wed 2025-11-26 20:13:23 UTC; 11min ago
   Invocation: d68d8dff3cd040509cb23c21bcd097b
   Docs: man:systemd.special(7)

Nov 26 20:13:23 client.arromichina.net systemd[1]: Reached target remote-fs
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 веб-сервера: `mount -o bind /var/www/ /srv/nfs/www/` И на сервере проверим, что отображается в каталоге `/srv/nfs`(рис. 2.17).

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

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

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

```
[root@client.arromichina.net ~]# ls /mnt/nfs
www
[root@client.arromichina.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.arromichina.net ~]# exportfs -r
[root@server.arromichina.net ~]#
```

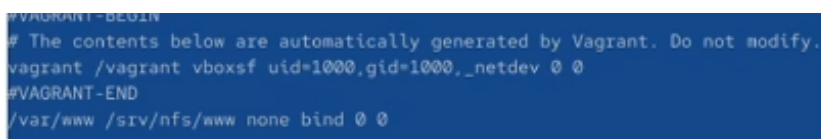
Рис. 2.20: Экспорт всех каталогов, упомянутых в файле `/etc/exports`.

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

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

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

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



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

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

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

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

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

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

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

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


```

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

```

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

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

```

exports      [-M--] 25 L:[ 1+ 2 3/ 3] *(72 / 91b) 0032 0x0[*][X]
/srv/nfs *(ro)
/srv/nfs/www 192.168.0.0/16(rw)
/srv/nfs/home/arromichina 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).

```

#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/arromichina/common /srv/nfs/home/arromichina none bind 0 0

```

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

Повторно экспортируем каталоги(рис. 2.28).

```

[root@server.arromichina.net ~]# exportfs -r
[root@server.arromichina.net ~]# █

```

Рис. 2.28: Повторный экспорт каталогов.

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

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

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

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

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

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

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

```
[root@client.arromichina.net ~]# cd /mnt/nfs/home/arromichina
-bash: cd: /mnt/nfs/home/arromichina: Permission denied
[root@client.arromichina.net ~]#
```

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

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

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

Рис. 2.32: Просмотр наличия изменений на сервере в каталоге пользователя /home/arromichina/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).

```
[root@server.arromichina.net ~]# cd /vagrant/provision/server
[root@server.arromichina.net server]# mkdir -p /vagrant/provision/server/nfs
/etc
[root@server.arromichina.net server]# cp -R /etc/exports /vagrant/provision/
server/nfs/etc/
[root@server.arromichina.net server]# cd /vagrant/provision/server
[root@server.arromichina.net server]# touch nfs.sh
[root@server.arromichina.net server]# chmod +x nfs.sh
[root@server.arromichina.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+28 29/ 29] *(931 / 931b) <EOF>  [*][X]
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/nfs/etc/* /etc
restorecon -vR /etc
echo "Configure firewall"
firewall-cmd --add-service nfs --permanent
firewall-cmd --add-service mountd --add-service rpc-bind --permanent
firewall-cmd --reload
echo "Tuning SELinux"
mkdir -p /srv/nfs
semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
restorecon -vR /srv/nfs
echo "Mounting dirs"
mkdir -p /srv/nfs/www
mount -o bind /var/www /srv/nfs/www
echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
mkdir -p /srv/nfs/home/user
mkdir -p -m 700 /home/user/common
chown user:user /home/user/common
mount -o bind /home/user/common /srv/nfs/home/user
echo "/home/user/common /srv/nfs/home/user none bind 0 0" >> /etc/fstab
echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
systemctl restart firewalld

```

Рис. 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.arromichina.net client]# cd /vagrant/provision/client
[root@client.arromichina.net client]# touch nfs.sh
[root@client.arromichina.net client]# chmod +x nfs.sh
[root@client.arromichina.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--] 24 L: [ 1* 7 8/ 9] *(208 / 281b) 0046 0x02E
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install nfs-utils
echo "Mounting dirs"
mkdir -p /mnt/nfs
mount server.arrowichina.net:/srv/nfs /mnt/nfs
echo "server.arrowichina.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).

```
server.vm.provision :shell,
  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, "--urde", "on"]
    virtualbox.customize ["modifyvm", :id, "--urdeport", "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 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"
```

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 `portmap`), 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