

Лабораторная работа №4

Базовая настройка HTTP-сервера Apache

Беличева Дарья Михайловна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Контрольные вопросы	16
5	Выводы	18

Список иллюстраций

3.1	Вывод списка групп	6
3.2	Установка веб-сервера	7
3.3	Команда firewall	7
3.4	Расширенный лог системных сообщений	8
3.5	Активация и запуск HTTP-сервер	8
3.6	Лог ошибок и мониторинг доступа	8
3.7	Тестовая страница	9
3.8	Результат мониторинга	9
3.9	Файл прямой зоны	10
3.10	Файл обратной зоны	10
3.11	Удаление файлов журналов DNS	11
3.12	Редактирование файл server.dmbelicheva.net.conf	11
3.13	Редактирование файл www.dmbelicheva.net.conf	12
3.14	Редактирование файла	12
3.15	Редактирование файла	13
3.16	Редактирование файла	13
3.17	Содержимое сайта	13
3.18	Содержимое сайта	14
3.19	Окно терминала	14
3.20	Редактирование файла	15

1 Цель работы

Приобрести практические навыки по установке и базовому конфигурированию HTTP-сервера Apache.

2 Задание

1. Установить необходимые для работы HTTP-сервера пакеты.
2. Запустить HTTP-сервер с базовой конфигурацией и проанализируйте его работу.
3. Настроить виртуальный хостинг.
4. Написать скрипт для Vagrant, фиксирующий действия по установке и настройке HTTP-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile`

3 Выполнение лабораторной работы

Установка HTTP-сервера

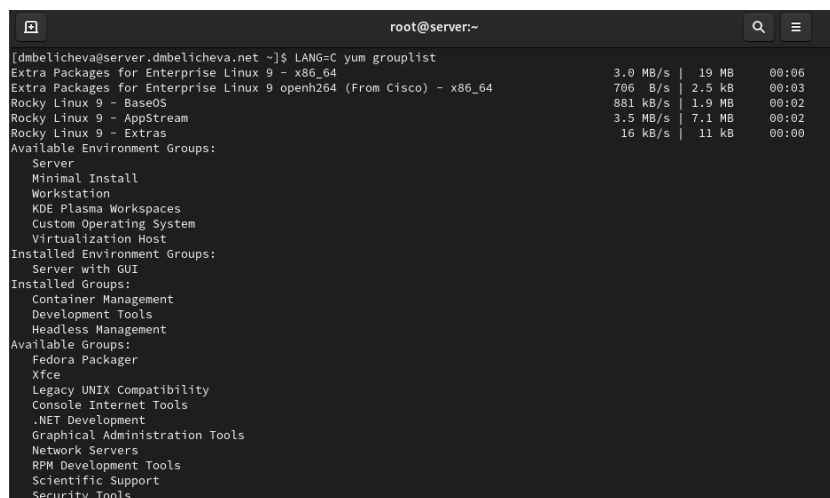
Загрузим операционную систему и перейдем в рабочий каталог с проектом:

```
cd C:\Users\dasha\work\study\dmbelicheva\vagrant
```

Запустим виртуальную машину `server: make server-up`.

На виртуальной машине `server` войдем под своим пользователем и откроем терминал. Перейдем в режим суперпользователя.

Установим из репозитория стандартный веб-сервер (HTTP-сервер и утилиты `httpd`, криптоутилиты и пр.):



```
root@server:~  
[dmbelicheva@server.dmbelicheva.net ~]$ LANG=C yum grouplist  
Extra Packages for Enterprise Linux 9 - x86_64          3.0 MB/s | 19 MB   00:06  
Extra Packages for Enterprise Linux 9 openh264 (From Cisco) - x86_64 706 B/s | 2.5 kB  00:03  
Rocky Linux 9 - BaseOS          881 kB/s | 1.9 MB  00:02  
Rocky Linux 9 - AppStream       3.5 MB/s | 7.1 MB  00:02  
Rocky Linux 9 - Extras          16 kB/s | 11 kB   00:00  
Available Environment Groups:  
  Server  
  Minimal Install  
  Workstation  
  KDE Plasma Workspaces  
  Custom Operating System  
  Virtualization Host  
Installed Environment Groups:  
  Server with GUI  
Installed Groups:  
  Container Management  
  Development Tools  
  Headless Management  
Available Groups:  
  Fedora Packager  
  Xfce  
  Legacy UNIX Compatibility  
  Console Internet Tools  
  .NET Development  
  Graphical Administration Tools  
  Network Servers  
  RPM Development Tools  
  Scientific Support  
  Security Tools
```

Рис. 3.1: Вывод списка групп

```
[root@server.dmbelicheva.net ~]# dnf -y groupinstall "Basic Web Server"
Rocky Linux 9 - BaseOS                               2.4 kB/s | 4.1 kB  00:01
Rocky Linux 9 - AppStream                             12 kB/s | 4.5 kB  00:00
Rocky Linux 9 - Extras                               852 B/s | 2.9 kB  00:03
Dependencies resolved.
=====
Package                Architecture      Version              Repository           Size
=====
Installing group/module packages:
httpd                  x86_64            2.4.53-11.el9_2.5    appstream             47 k
httpd-manual            noarch            2.4.53-11.el9_2.5    appstream             2.2 M
mod_fcgid               x86_64            2.3.9-28.el9         appstream             74 k
mod_ssl                x86_64            1:2.4.53-11.el9_2.5  appstream            110 k
Installing dependencies:
apr                    x86_64            1.7.0-11.el9         appstream             123 k
apr-util              x86_64            1.6.1-20.el9_2.1     appstream             94 k
apr-util-bdb          x86_64            1.6.1-20.el9_2.1     appstream             12 k
httpd-core            x86_64            2.4.53-11.el9_2.5    appstream            1.4 M
httpd-filesystem       noarch            2.4.53-11.el9_2.5    appstream             14 k
httpd-tools            x86_64            2.4.53-11.el9_2.5    appstream             81 k
rocky-logos-httpd      noarch            90.14-1.el9          appstream             24 k
Installing weak dependencies:
apr-util-openssl       x86_64            1.6.1-20.el9_2.1     appstream             14 k
mod_http2              x86_64            1.15.19-4.el9_2.4    appstream            149 k
mod_lua                x86_64            2.4.53-11.el9_2.5    appstream             61 k
Installing Groups:
```

Рис. 3.2: Установка веб-сервера

Базовое конфигурирование HTTP-сервера

Внесем изменения в настройки межсетевого экрана узла server, разрешив работу с http:

```
[root@server.dmbelicheva.net ~]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns ssh
[root@server.dmbelicheva.net ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula
bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine ch
eckmk-agent cockpit collectd condor-collector cratedb ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-r
egistry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 fre
eipa-ldap freeipa-ltdaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana g
re high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jellyfin
jenkins kadmin kdeconnect kerberos kibana klogin kpasswd 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 ku
be-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-netw
ork llmnr llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssq
l murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageco
nsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-e
xporter proxy-dhcp ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquoted rsh
rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission snmp snmp-tls snmptl
s-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syner
gy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsa vnc-server wbem-
http wbm-http https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsmn wsmans xdmcp xmpp-
bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.dmbelicheva.net ~]# firewall-cmd --add-service=http
success
[root@server.dmbelicheva.net ~]# firewall-cmd --add-service=http --permanent
success
[root@server.dmbelicheva.net ~]#
```

Рис. 3.3: Команда firewall

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:

```
[root@server.dmbelicheva.net ~]# journalctl -x -f
Nov 13 14:18:28 server.dmbelicheva.net systemd[1]: Starting Cleanup of Temporary Directories...
Subject: A start job for unit systemd-tmpfiles-clean.service has begun execution
Defined-By: systemd
Support: https://access.redhat.com/support

A start job for unit systemd-tmpfiles-clean.service has begun execution.

The job identifier is 2727.
Nov 13 14:18:28 server.dmbelicheva.net systemd[1]: systemd-tmpfiles-clean.service: Deactivated successfully.
Subject: Unit succeeded
Defined-By: systemd
Support: https://access.redhat.com/support

The unit systemd-tmpfiles-clean.service has successfully entered the 'dead' state.
Nov 13 14:18:28 server.dmbelicheva.net systemd[1]: Finished Cleanup of Temporary Directories.
Subject: A start job for unit systemd-tmpfiles-clean.service has finished successfully
Defined-By: systemd
Support: https://access.redhat.com/support

A start job for unit systemd-tmpfiles-clean.service has finished successfully.

The job identifier is 2727.
Nov 13 14:18:28 server.dmbelicheva.net systemd[1]: run-credentials-systemd\x2dtmpfiles\x2dclean.service.mount: Deactivated successfully.
Subject: Unit succeeded
Defined-By: systemd
Support: https://access.redhat.com/support

The unit run-credentials-systemd\x2dtmpfiles\x2dclean.service.mount has successfully entered the 'dead' state.
Nov 13 14:28:01 server.dmbelicheva.net systemd[5590]: Started Application launched by gnome-shell.
Subject: A start job for unit UNIT has finished successfully
```

Рис. 3.4: Расширенный лог системных сообщений

В первом терминале активируем и запустим HTTP-сервер:

```
[root@server.dmbelicheva.net ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@server.dmbelicheva.net ~]# systemctl start httpd
```

Рис. 3.5: Активация и запуск HTTP-сервер

Анализ работы HTTP-сервера

Запустим виртуальную машину client: make client-up.

На виртуальной машине server посмотрим лог ошибок работы веб-сервера и мониторинг доступа к веб-серверу:

```
[root@server.dmbelicheva.net ~]# tail -f /var/log/httpd/error_log
[Mon Nov 13 14:32:12.663636 2023] [core:notice] [pid 7252:tid 7252] SELinux policy enabled; httpd running as context
system_u:system_r:httpd_t:s0
[Mon Nov 13 14:32:12.667070 2023] [suexec:notice] [pid 7252:tid 7252] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Mon Nov 13 14:32:12.703442 2023] [lbmethod_heartbeat:notice] [pid 7252:tid 7252] AH02282: No slotmem from mod_heartm
onitor
[Mon Nov 13 14:32:12.728280 2023] [mpm_event:notice] [pid 7252:tid 7252] AH00489: Apache/2.4.53 (Rocky Linux) OpenSSL
/3.0.7 mod_fcgid/2.3.9 configured -- resuming normal operations
[Mon Nov 13 14:32:12.728331 2023] [core:notice] [pid 7252:tid 7252] AH00094: Command line: '/usr/sbin/httpd -D FOREGR
OUND'
^C
[root@server.dmbelicheva.net ~]# tail -f /var/log/httpd/access_log
```

Рис. 3.6: Лог ошибок и мониторинг доступа

На виртуальной машине client запустили браузер и в адресной строке ввели 192.168.1.1.


```
root@server:/var/named/master/fz
GNU nano 5.6.1 dmbelicheva.net
$ORIGIN .
$TTL 86400 ; 1 day
dmbelicheva.net IN SOA dmbelicheva.net. server.dmbelicheva.net. (
    2023111304 ; serial
    86400 ; refresh (1 day)
    3600 ; retry (1 hour)
    604800 ; expire (1 week)
    10800 ; minimum (3 hours)
)
NS dmbelicheva.net.
A 192.168.1.1
$ORIGIN dmbelicheva.net.
$TTL 300 ; 5 minutes
client A 192.168.1.30
TXT "3197659e2c40e26a0e15932b7c018329f7"
$TTL 86400 ; 1 day
dhcp A 192.168.1.1
ns A 192.168.1.1
server A 192.168.1.1
www A 192.168.1.1
```

Рис. 3.9: Файл прямой зоны

и в конце файла обратной зоны /var/named/master/rz/192.168.1:

```
root@server:/var/named/master/rz
GNU nano 5.6.1 192.168.1
$ORIGIN .
$TTL 86400 ; 1 day
1.168.192.in-addr.arpa IN SOA 1.168.192.in-addr.arpa. server.dmbelicheva.net. (
    2023111303 ; serial
    86400 ; refresh (1 day)
    3600 ; retry (1 hour)
    604800 ; expire (1 week)
    10800 ; minimum (3 hours)
)
NS 1.168.192.in-addr.arpa.
A 192.168.1.1
PTR server.dmbelicheva.net.
$ORIGIN 1.168.192.in-addr.arpa.
1 PTR server.dmbelicheva.net.
PTR ns.dmbelicheva.net.
PTR dhcp.dmbelicheva.net.
$TTL 300 ; 5 minutes
30 PTR client.dmbelicheva.net.
1 PTR server.dmbelicheva.net.
1 PTR www.dmbelicheva.net.
```

Рис. 3.10: Файл обратной зоны

В обоих файлах изменим серийный номер файла зоны, указав текущую дату в нотации ГГТТММДДВВ. Также из соответствующих каталогов удалим файлы журналов DNS: dmbelicheva.net.jnl и 192.168.1.jnl.

```

[root@server.dmbelicheva.net fz]# ls
dmbelicheva.net  dmbelicheva.net.jnl
[root@server.dmbelicheva.net fz]# rm -r dmbelicheva.net.jnl
rm: remove regular file 'dmbelicheva.net.jnl'? y
[root@server.dmbelicheva.net fz]# ls
dmbelicheva.net
[root@server.dmbelicheva.net fz]# cd ..
[root@server.dmbelicheva.net master]# rz
bash: rz: command not found...
Install package 'lrzsz' to provide command 'rz'? [N/y] n

[root@server.dmbelicheva.net master]# cd rz
[root@server.dmbelicheva.net rz]# ls
192.168.1  192.168.1.jnl
[root@server.dmbelicheva.net rz]# rm -r 192.168.1.jnl
rm: remove regular file '192.168.1.jnl'? y
[root@server.dmbelicheva.net rz]# ls
192.168.1
[root@server.dmbelicheva.net rz]# █

```

Рис. 3.11: Удаление файлов журналов DNS

Перезапустим DNS-сервер: `systemctl start named`

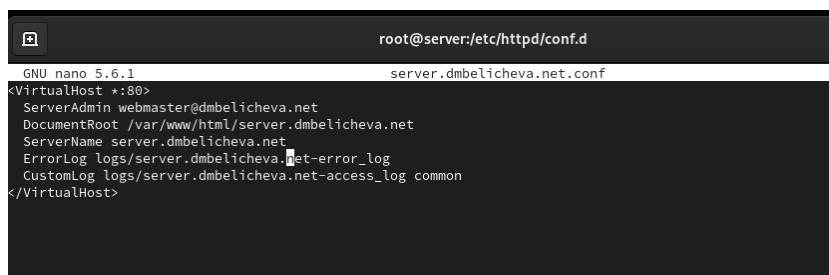
В каталоге `/etc/httpd/conf.d` создайте файлы `server.dmbelicheva.net.conf` и `www.dmbelicheva.net.conf`:

```

cd /etc/httpd/conf.d
touch server.dmbelicheva.net.conf
touch www.dmbelicheva.net.conf

```

Откроем на редактирование файл `server.dmbelicheva.net.conf` и внесем следующее содержание:



```

GNU nano 5.6.1 server.dmbelicheva.net.conf
<VirtualHost *:80>
  ServerAdmin webmaster@dmbelicheva.net
  DocumentRoot /var/www/html/server.dmbelicheva.net
  ServerName server.dmbelicheva.net
  ErrorLog logs/server.dmbelicheva.net-error_log
  CustomLog logs/server.dmbelicheva.net-access_log common
</VirtualHost>

```

Рис. 3.12: Редактирование файл `server.dmbelicheva.net.conf`

Откроем на редактирование файл `www.dmbelicheva.net.conf` и внесем следующее содержание:

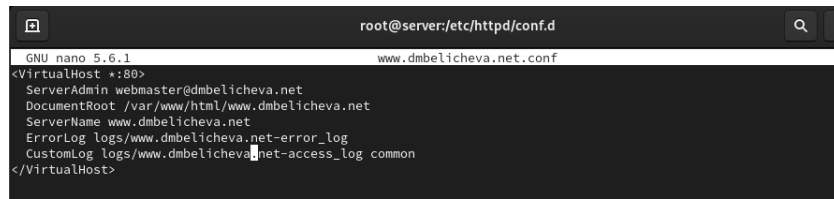


Рис. 3.13: Редактирование файл `www.dmbelicheva.net.conf`

Перейдем в каталог `/var/www/html`, в котором должны находиться файлы с содержимым (контентом) веб-серверов, и создадим тестовые страницы для виртуальных веб-серверов `server.dmbelicheva.net` и `www.dmbelicheva.net`. Для виртуального веб-сервера `server.dmbelicheva.net`:

```
cd /var/www/html
mkdir server.dmbelicheva.net
cd /var/www/html/server.dmbelicheva.net
touch index.html
```

Откроем на редактирование файл `index.html` и внесем следующее содержание: `Welcome to the server.dmbelicheva.net server.`

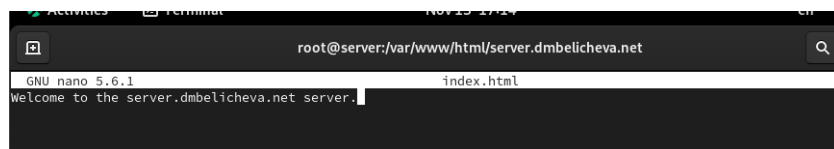


Рис. 3.14: Редактирование файла

Для виртуального веб-сервера `www.dmbelicheva.net`:

```
cd /var/www/html
mkdir www.dmbelicheva.net
cd /var/www/html/www.dmbelicheva.net
touch index.html
```

Откроем на редактирование файл `index.html` и внесем следующее содержание:
`Welcome to the www.dmbelicheva.net server.`

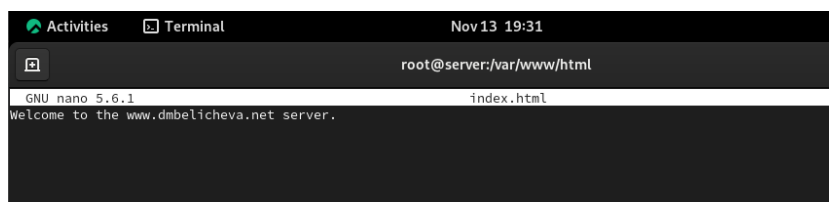


Рис. 3.15: Редактирование файла

Скорректируем права доступа в каталог с веб-контентом: `chown -R apache:apache /var/www`

Восстановим контекст безопасности в SELinux и перезапустим HTTP-сервер.

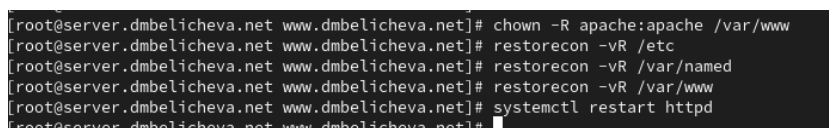


Рис. 3.16: Редактирование файла

На виртуальной машине client убедимся в корректном доступе к веб-серверу по адресам `server.dmbelicheva.net` и `www.dmbelicheva.net` в адресной строке веб-браузера.

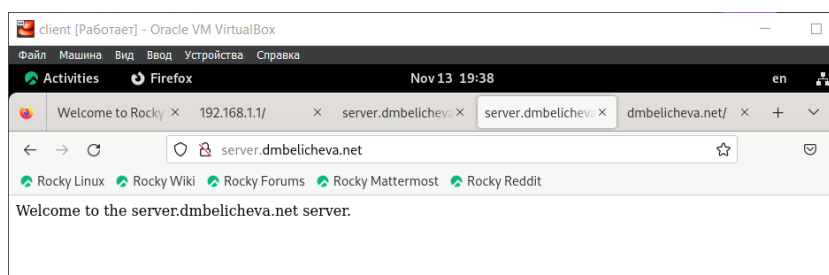


Рис. 3.17: Содержимое сайта

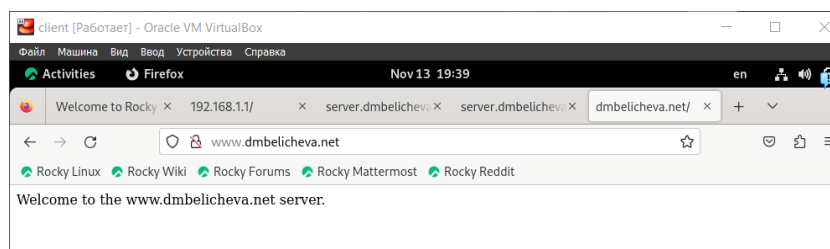


Рис. 3.18: Содержимое сайта

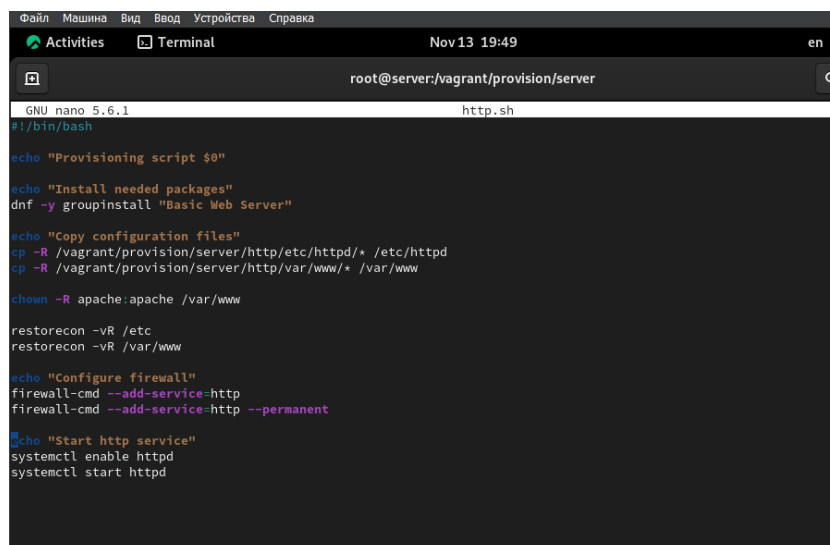
Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине `server` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `http`, в который поместим в соответствующие подкаталоги конфигурационные файлы HTTP-сервера. Заменяем конфигурационные файлы DNS-сервера. В каталоге `/vagrant/provision/server` создадим исполняемый файл `http.sh`.

```
[root@server.dmbelicheva.net www.dmbelicheva.net]# cd /vagrant/provision/server
[root@server.dmbelicheva.net server]# mkdir -p /vagrant/provision/server/http/etc/httpd/conf.d
[root@server.dmbelicheva.net server]# mkdir -p /vagrant/provision/server/http/var/www/html
[root@server.dmbelicheva.net server]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d/
[root@server.dmbelicheva.net server]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
[root@server.dmbelicheva.net server]# cd /vagrant/provision/server/dns/
[root@server.dmbelicheva.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/
cp: overwrite '/vagrant/provision/server/dns/var/named/data/named.run'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/dynamic/managed-keys.bind.jnl'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/dynamic/managed-keys.bind'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/dmbelicheva.net.jnl'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/dmbelicheva.net'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1.jnl'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/named.ca'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/named.empty'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/named.localhost'? y
cp: overwrite '/vagrant/provision/server/dns/var/named/named.loopback'? y
[root@server.dmbelicheva.net dns]# cd /vagrant/provision/server
[root@server.dmbelicheva.net server]# touch http.sh
[root@server.dmbelicheva.net server]# chmod +x http.sh
[root@server.dmbelicheva.net server]# nano http.sh
```

Рис. 3.19: Окно терминала

Открыв его на редактирование, пропишем в нём следующий скрипт:



```
GNU nano 5.6.1 http.sh
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y groupinstall "Basic Web Server"

echo "Copy configuration files"
cp -R /vagrant/provision/server/http/etc/httpd/* /etc/httpd
cp -R /vagrant/provision/server/http/var/www/* /var/www

chown -R apache:apache /var/www

restorecon -vR /etc
restorecon -vR /var/www

echo "Configure firewall"
firewall-cmd --add-service=http
firewall-cmd --add-service=http --permanent

echo "Start http service"
systemctl enable httpd
systemctl start httpd
```

Рис. 3.20: Редактирование файла

Этот скрипт повторяет произведённые вами действия по установке и настройке HTTP-сервера.

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile необходимо добавить в конфигурации сервера следующую запись:

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

4 Контрольные вопросы

1. Через какой порт по умолчанию работает Apache?

Сервер Apache по умолчанию настроен на ожидание входящих соединений через порт 80.

2. Под каким пользователем запускается Apache и к какой группе относится этот пользователь?

По умолчанию, Apache запускается от пользователя, называемого apache или www-data, в зависимости от операционной системы. Этот пользователь обладает минимальными привилегиями, что делает его безопасным для запуска веб-сервера. Группа www-data.

3. Где располагаются лог-файлы веб-сервера? Что можно по ним отслеживать?

Логи сервера, в том числе логи apache хранятся в каталоге /var/log/. Лог-файлы (файлы регистрации, журнальные файлы) на Linux - это текстовые файлы о событиях, произошедших на сайте: информация о параметрах посещений сайта и ошибках, которые возникали на нем.

4. Где по умолчанию содержится контент веб-серверов?

В каталоге /var/www/html

5. Каким образом реализуется виртуальный хостинг? Что он даёт?

Виртуальный хостинг реализуется путем размещения нескольких сайтов на одном физическом сервере. Веб-сервер использует виртуальные хосты (обычно на основе доменных имен) для определения, какой сайт обслуживать при запросе. Это позволяет разделить ресурсы сервера между разными сайтами и обеспечивает эффективное использование аппаратных ресурсов. Виртуальный хостинг позволяет размещать сайты с базовыми потребностями.

5 Выводы

В процессе выполнения лабораторной работы я приобрела практические навыки по установке и базовому конфигурированию HTTP-сервера Apache.