

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

дисциплина: Администрирование сетевых подсистем

Студент: Бансимба Клодели Дьегра

Студ. билет № 1032215651

Группа: НПИбд-02-22

**МОСКВА**

2024 г.

## Цель работы:

Целью данной работы является приобретение практических навыков по установке и базовому конфигурированию HTTP-сервера Apache.

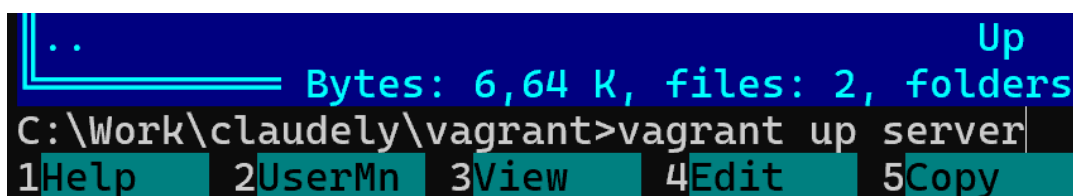
## Выполнение работы:

Загрузим нашу операционную систему и перейдём в рабочий каталог с проектом:

```
cd /var/tmp/claudey/vagrant
```

Далее запустим виртуальную машину server (Рис. 1.1):

```
Vagrant up server
```



**Рис. 1.1.** Открытие рабочего каталога с проектом и запуск виртуальной машины server.

На виртуальной машине server войдём под нашим пользователем и откроем терминал. Далее перейдём в режим суперпользователя и установим из репозитория стандартный веб-сервер (HTTP-сервер и утилиты httpd, криптоутилиты и пр.) (Рис. 1.2):

```
LANG=C yum grouplist
```

```
dnf -y groupinstall "Basic Web Server"
```

```

[claudely@server.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# LANG=C yum grouplist
Last metadata expiration check: 3:28:45 ago on Fri Sep 27 17:17:53 2024.
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
  VideoLAN Client
  Xfce
  Legacy UNIX Compatibility
  Console Internet Tools
  .NET Development
  Graphical Administration Tools
  Network Servers
  RPM Development Tools
  Scientific Support
  Security Tools
  Smart Card Support
  System Tools
[root@server.claudely.net ~]# dnf -y groupinstall "Basic Web Server"
Last metadata expiration check: 3:29:01 ago on Fri 27 Sep 2024 05:17:53 PM UTC.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing group/module packages:
httpd                          x86_64            2.4.57-11.el9_4.1  appstream         44 k
httpd-manual                   noarch            2.4.57-11.el9_4.1  appstream         2.2 M
mod_fcgid                      x86_64            2.3.9-28.el9       appstream          74 k
mod_ssl                        x86_64            1:2.4.57-11.el9_4.1 appstream        108 k
Installing dependencies:
apr                            x86_64            1.7.0-12.el9_3     appstream         122 k
apr-util                      x86_64            1.6.1-23.el9       appstream          94 k
apr-util-bdb                  x86_64            1.6.1-23.el9       appstream          12 k
httpd-core                     x86_64            2.4.57-11.el9_4.1  appstream         1.4 M
httpd-filesystem              noarch            2.4.57-11.el9_4.1  appstream          11 k
httpd-tools                   x86_64            2.4.57-11.el9_4.1  appstream          79 k
rocky-logos-httpd             noarch            00.15-2.el9        appstream          24 k

```

**Рис. 1.2.** Переход в режим суперпользователя и установка из репозитория стандартного веб-сервера.

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

```
firewall-cmd --list-services
```

```
firewall-cmd --get-services
```

```
firewall-cmd --add-service=http
```

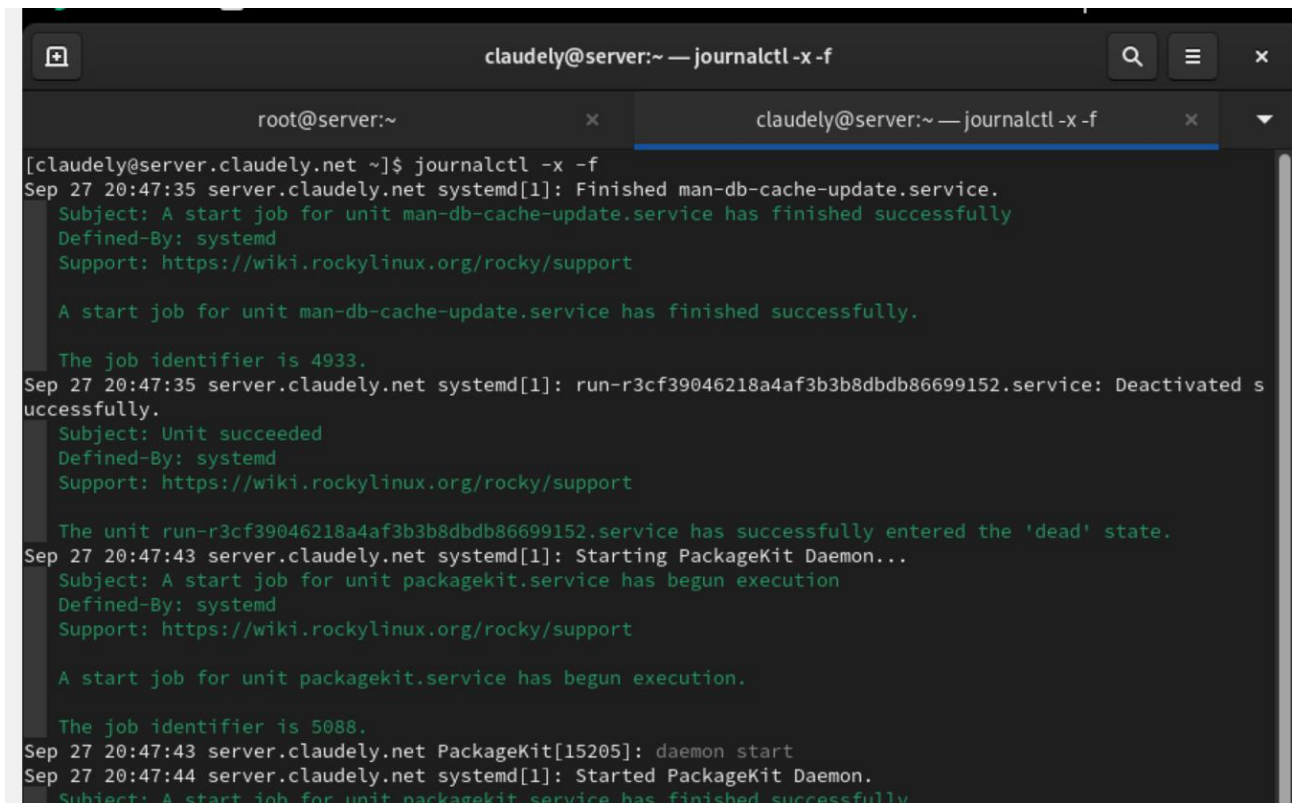
firewall-cmd --add-service=http --permanent (Рис. 2.2).

```
[root@server.claudely.net ~]#  
[root@server.claudely.net ~]# firewall-cmd --list-services  
cockpit dhcp dhcpv6-client dns ssh  
[root@server.claudely.net ~]# firewall-cmd --get-services  
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp  
2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-t  
estnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collect  
d condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-  
tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman forema  
n-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client gan  
glia-master git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ips  
ec irc ircs iscsi-target isns 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 kube-scheduler-secure kube-worker kubelet kubelet-readonly kub  
elet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesi  
eve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula net  
bios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-v  
mconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-expor  
ter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bin  
d rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smt  
ps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp sv  
n syncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-soc  
ks transmission-client upnp-client vds vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws  
-discovery-client ws-discovery-tcp ws-discovery-udp wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp  
-server zabbix-agent zabbix-server zerotier  
[root@server.claudely.net ~]# firewall-cmd --add-service=http  
success  
[root@server.claudely.net ~]# firewall-cmd --add-service=http --permanent  
success  
[root@server.claudely.net ~]#
```

**Рис. 2.2.** Внесение изменений в настройки межсетевого экрана узла  
server, разрешив работу с http.

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы (Рис. 2.3):

```
journalctl -x -f
```



```
claudely@server:~ — journalctl -x -f
root@server:~
[claudely@server.claudely.net ~]$ journalctl -x -f
Sep 27 20:47:35 server.claudely.net systemd[1]: Finished man-db-cache-update.service.
  Subject: A start job for unit man-db-cache-update.service has finished successfully
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  A start job for unit man-db-cache-update.service has finished successfully.

  The job identifier is 4933.
Sep 27 20:47:35 server.claudely.net systemd[1]: run-r3cf39046218a4af3b3b8dbdb86699152.service: Deactivated s
successfully.
  Subject: Unit succeeded
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  The unit run-r3cf39046218a4af3b3b8dbdb86699152.service has successfully entered the 'dead' state.
Sep 27 20:47:43 server.claudely.net systemd[1]: Starting PackageKit Daemon...
  Subject: A start job for unit packagekit.service has begun execution
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  A start job for unit packagekit.service has begun execution.

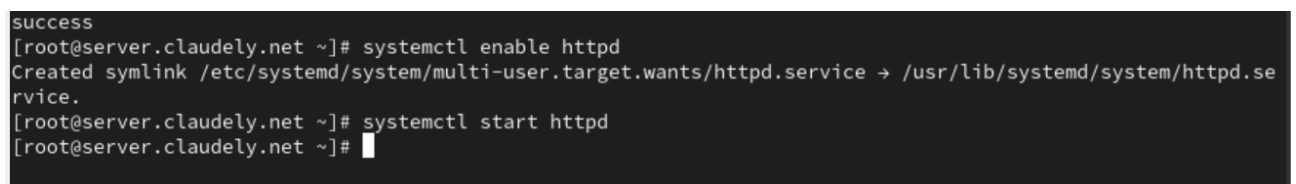
  The job identifier is 5088.
Sep 27 20:47:43 server.claudely.net PackageKit[15205]: daemon start
Sep 27 20:47:44 server.claudely.net systemd[1]: Started PackageKit Daemon.
  Subject: A start job for unit packagekit.service has finished successfully
```

**Рис. 2.3.** Запуск в дополнительном терминале в режиме реального времени расширенного лога системных сообщений для проверки корректности работы системы.

В первом терминале активируем и запустим HTTP-сервер (Рис. 2.4):

```
systemctl enable httpd
```

```
systemctl start httpd
```



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

**Рис. 2.4.** Активация и запуск HTTP-сервера.

Просмотрим расширенный лог системных сообщений, убедимся, что веб-сервер успешно запустился (Рис. 2.5):

```

[vice.
[root@server.claudely.net ~]# systemctl start httpd
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# tail -f /var/log/httpd/error_log
[Fri Sep 27 20:53:41.484118 2024] [core:notice] [pid 15335:tid 15335] SELinux policy enabled; httpd running
as context system_u:system_r:httpd_t:s0
[Fri Sep 27 20:53:41.489060 2024] [suexec:notice] [pid 15335:tid 15335] AH01232: suEXEC mechanism enabled (w
rapper: /usr/sbin/suexec)
[Fri Sep 27 20:53:41.507166 2024] [lbmethod_heartbeat:notice] [pid 15335:tid 15335] AH02282: No slotmem from
mod_heartbeat
[Fri Sep 27 20:53:41.539055 2024] [mpm_event:notice] [pid 15335:tid 15335] AH00489: Apache/2.4.57 (Rocky Lin
ux) OpenSSL/3.0.7 mod_fcgid/2.3.9 configured -- resuming normal operations
[Fri Sep 27 20:53:41.539090 2024] [core:notice] [pid 15335:tid 15335] AH00094: Command line: '/usr/sbin/http
d -D FOREGROUND'

```

**Рис. 2.5.** Просмотр расширенного лога системных сообщений.

Запустим виртуальную машину client (Рис. 3.1):

make client-up

```

C:\work\claudely\vagrant>vagrant up client --provision
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Running provisioner: common dummy (shell)...
    client: Running: C:/Users/bansi/AppData/Local/Temp/vagrant-shell20240928-9880-iwoate.sh
    client: Provisioning script /tmp/vagrant-shell
==> client: Running provisioner: common hostname (shell)...
    client: Running: C:/Users/bansi/AppData/Local/Temp/vagrant-shell20240928-9880-zv5xzi.sh
==> client: Running provisioner: common user (shell)...

```

**Рис. 3.1.** Запуск виртуальной машины client.

На виртуальной машине client запустим браузер и в адресной строке введём 192.168.1.1 (Рис. 3.3):

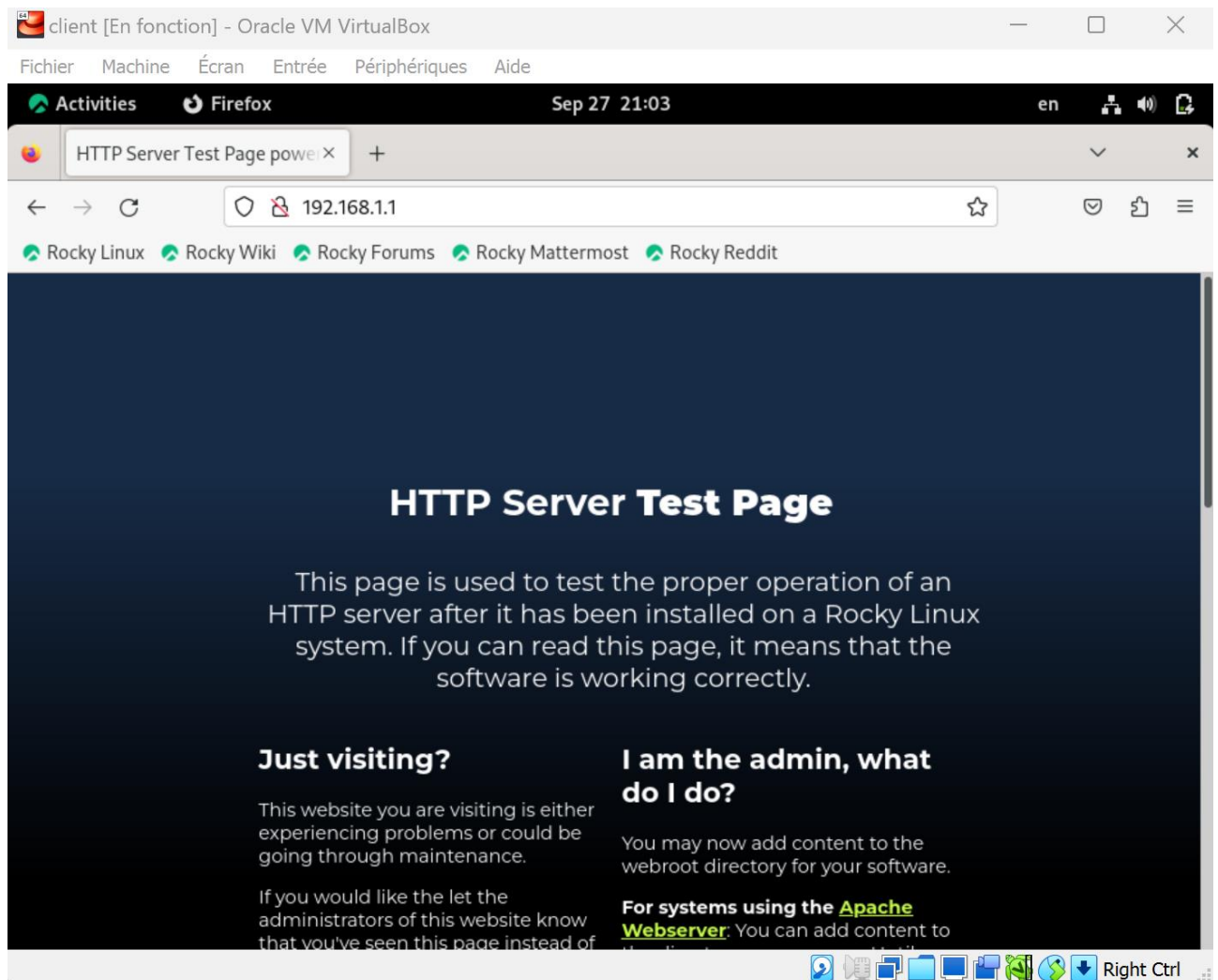
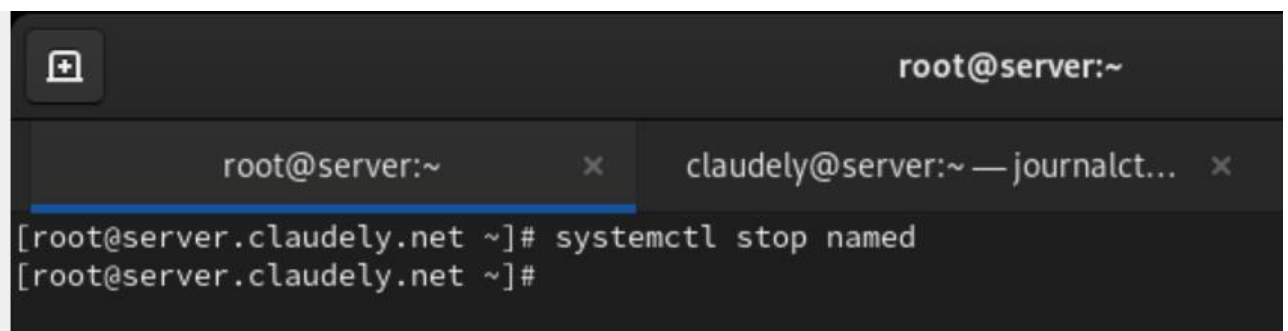


Рис. 3.3. Запуск браузера на виртуальной машине client и ввод в адресной строке 192.168.1.1.

Остановим работу DNS-сервера для внесения изменений в файлы описания DNS-зон (рис. 4.1):

```
systemctl stop named
```



**Рис. 4.1.** Остановка работы DNS-сервера для внесения изменений в файлы описания DNS-зон.

Теперь добавим запись для HTTP-сервера в конце файла прямой DNS-зоны /var/named/master/fz/claudey.net (рис. 4.2):

Server	A	192.168.1.1
www	A	192.168.1.1



```
root@server:~ x claudey@server:~ — journalct... x
$ORIGIN .
$TTL 86400 ; 1 day
claudey.net IN SOA claudey.net. server.claudey.net. (
    2024072704 ; serial
    86400 ; refresh (1 day)
    3600 ; retry (1 hour)
    604800 ; expire (1 week)
    10800 ; minimum (3 hours)
)
                NS    claudey.net.
                A      192.168.1.1
$ORIGIN claudey.net.
$TTL 300 ; 5 minutes
client A 192.168.1.30
                TXT    "318e4a6424df107a131a96573d72148283"
$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. 4.2.** Добавление записи для HTTP-сервера в конце файла прямой DNS-зоны /var/named/master/fz/claudey.net.

Также в конце файла обратной зоны /var/named/master/rz/192.168.1 (рис. 4.3):



1	PTR	server.claudely.net.
1	PTR	www.claudely.net.

```

$ORIGIN .
$TTL 86400          ; 1 day
1.168.192.in-addr.arpa IN SOA 1.168.192.in-addr.arpa. server.claudely.net. (
    2024072703 ; 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.claudely.net.
$ORIGIN 1.168.192.in-addr.arpa.
1          PTR     server.claudely.net.
          PTR     ns.claudely.net.
          PTR     dhcp.claudely.net.
$TTL 300      ; 5 minutes
30         PTR     client.claudely.net.
1          PTR     www.claudely.net.
~
~
~
~

```

**Рис. 4.3.** Добавление записи для HTTP-сервера в конце файла обратной DNS-зоны /var/named/master/rz/192.168.1.

Нам требуется из соответствующих каталогов удалить файлы журналов DNS (рис. 4.4):

claudely.net.jnl и 192.168.1.jnl. (рис. 4.4):

```

[root@server.claudely.net ~]# ls /var/named/master/fz/
claudely.net  claudely.net.jnl
[root@server.claudely.net ~]# ls /var/named/master/rz
192.168.1  192.168.1.jnl
[root@server.claudely.net ~]#

```

**Рис. 4.4.** Удаление файлов журналов DNS.

Перезапустим DNS-сервер:

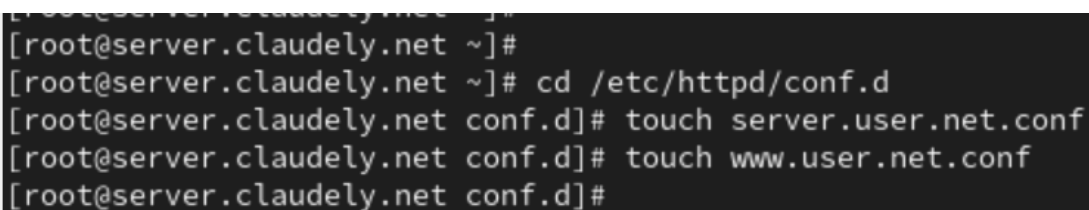
```
systemctl start named
```

В каталоге /etc/httpd/conf.d создадим файлы server.claudely.net.conf и www.claudely.net.conf (рис. 4.5):

```
cd /etc/httpd/conf.d
```

```
touch server.user.net.conf
```

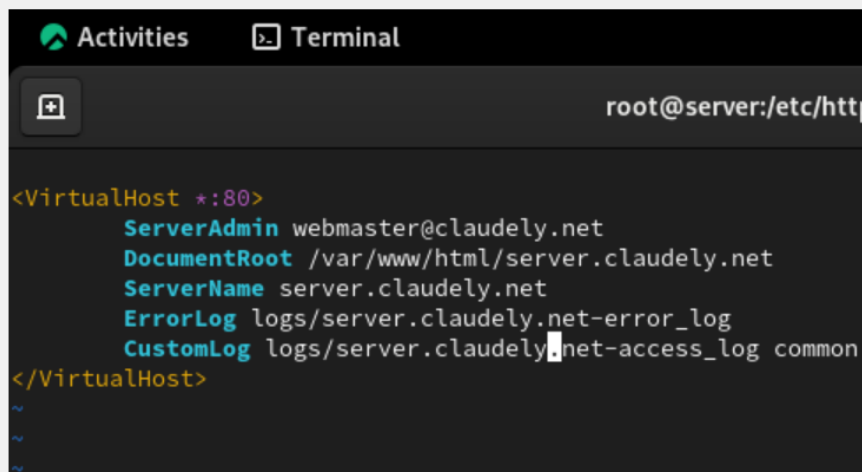
```
touch www.user.net.conf
```



```
[root@server.claudely.net ~]#  
[root@server.claudely.net ~]# cd /etc/httpd/conf.d  
[root@server.claudely.net conf.d]# touch server.user.net.conf  
[root@server.claudely.net conf.d]# touch www.user.net.conf  
[root@server.claudely.net conf.d]#
```

**Рис. 4.5.** Перезапуск DNS-сервера и создание в каталоге /etc/httpd/conf.d файлов server.claudely.net.conf и www.claudely.net.conf.

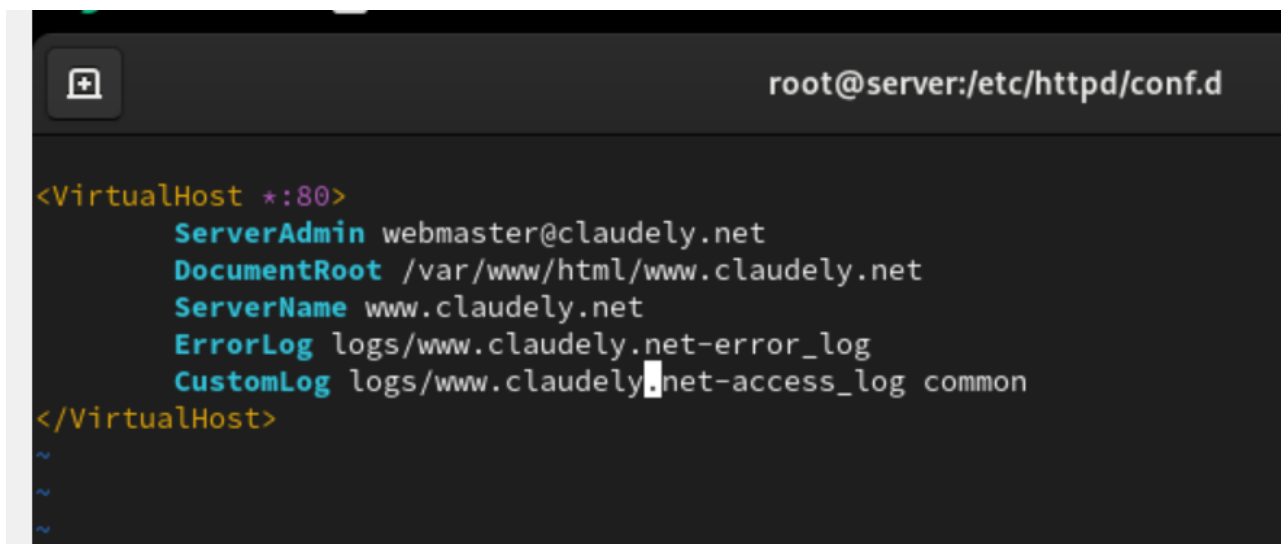
Откроем на редактирование файл server.claudely.net.conf и внесём туда следующее содержание (рис. 4.6):



```
Activities Terminal  
root@server:/etc/httpd/conf.d  
  
<VirtualHost *:80>  
    ServerAdmin webmaster@ Claudely.net  
    DocumentRoot /var/www/html/server.claudely.net  
    ServerName server.claudely.net  
    ErrorLog logs/server.claudely.net-error_log  
    CustomLog logs/server.claudely.net-access_log common  
</VirtualHost>  
~  
~  
~
```

**Рис. 4.6.** Открытие на редактирование файла `server.claudely.net.conf` и добавление содержания из лабораторной работы.

Откроем на редактирование файл `www.claudely.net.conf` и внесём туда следующее содержание (рис. 4.7):



```
root@server:/etc/httpd/conf.d

<VirtualHost *:80>
    ServerAdmin webmaster@claudely.net
    DocumentRoot /var/www/html/www.claudely.net
    ServerName www.claudely.net
    ErrorLog logs/www.claudely.net-error_log
    CustomLog logs/www.claudely.net-access_log common
</VirtualHost>
~
~
~
```

**Рис. 4.7.** Открытие на редактирование файла `www.claudely.net.conf` и добавление содержания из лабораторной работы.

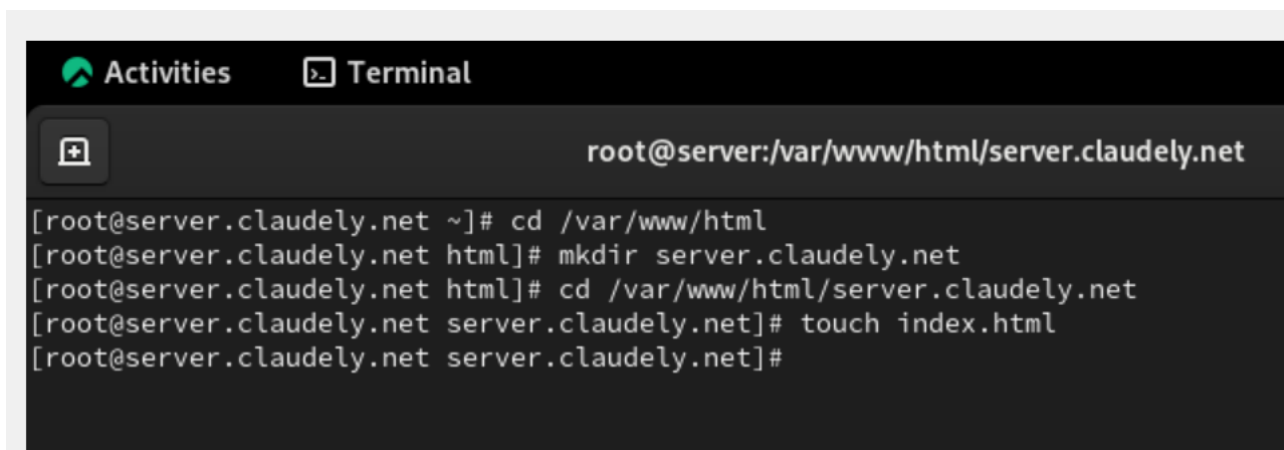
Перейдём в каталог `/var/www/html`, в котором находятся файлы с содержимым (контентом) веб-серверов, и создадим тестовые страницы для виртуальных веб-серверов `server.claudely.net` и `www.claudely.net`. Для виртуального веб-сервера `server.claudely.net`:

```
cd /var/www/html

mkdir server.claudely.net

cd /var/www/html/server.claudely.net

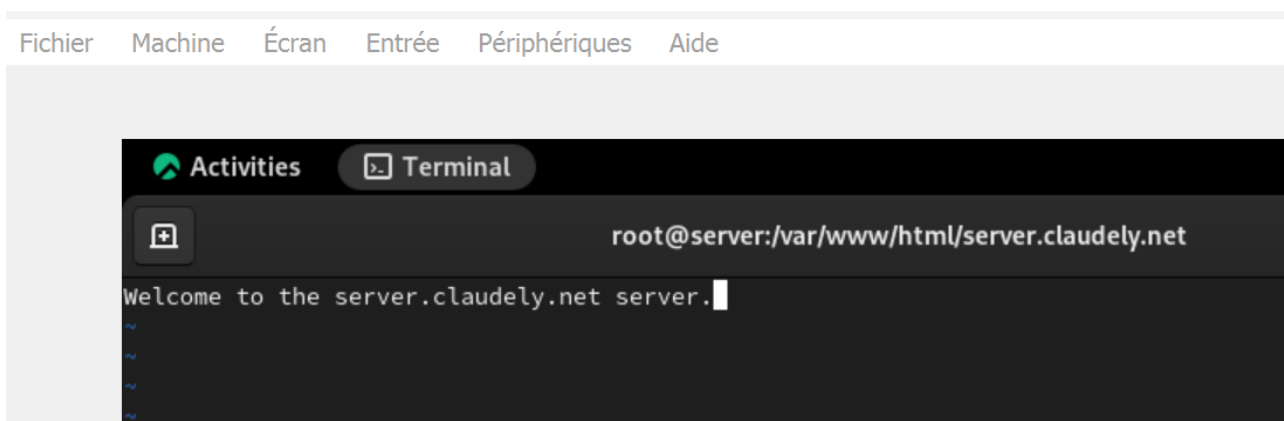
touch index.html (рис. 4.8):
```



**Рис. 4.8.** Открытие каталога `/var/www/html` и создание тестовой страницы для виртуального веб-сервера `server.claudely.net`.

Откроем на редактирование файл `index.html` и внесём следующее содержание (рис. 4.9):

Welcome to the server.claudely.net server.



**Рис. 4.9.** Открытие на редактирование файла `index.html` и внесение содержания.

Для виртуального веб-сервера `www.claudely.net` (рис. 4.10):

`cd /var/www/html`

`mkdir www.claudely.net`

`cd /var/www/html/www.claudely.net`

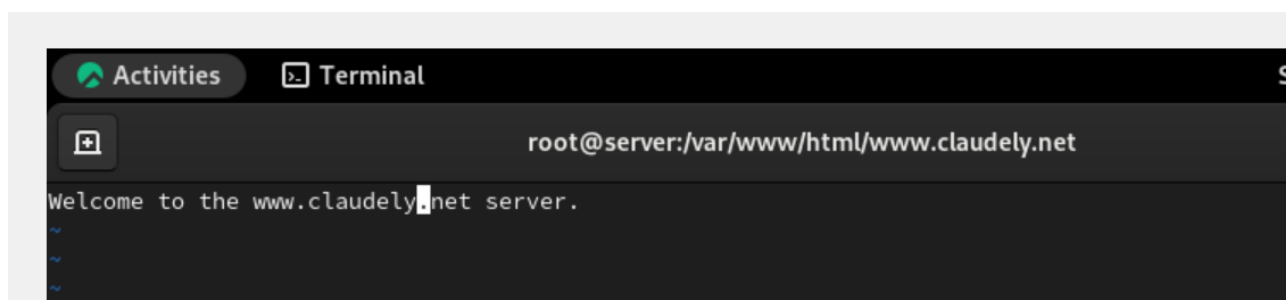
```
touch index.html
```

```
[root@server.claudely.net server.claudely.net]#  
[root@server.claudely.net server.claudely.net]# cd /var/www/html  
[root@server.claudely.net html]# mkdir www.claudely.net  
[root@server.claudely.net html]# cd /var/www/html/www.claudely.net  
[root@server.claudely.net www.claudely.net]# touch index.html  
[root@server.claudely.net www.claudely.net]#
```

**Рис. 4.10.** Открытие каталога /var/www/html и создание тестовой страницы для виртуального веб-сервера www.claudely.net.

Откроем на редактирование файл index.html и внесём следующее содержание (рис. 4.11):

```
Welcome to the www.claudely.net server.
```



**Рис. 4.11.** Открытие на редактирование файла index.html и внесение содержания.

Скорректируем права доступа в каталог с веб-контентом:

```
chown -R apache:apache /var/www
```

Далее восстановим контекст безопасности в SELinux:

```
restorecon -vR /etc/named
```

```
restorecon -vR /var/named
```

```
restorecon -vR /var/www
```

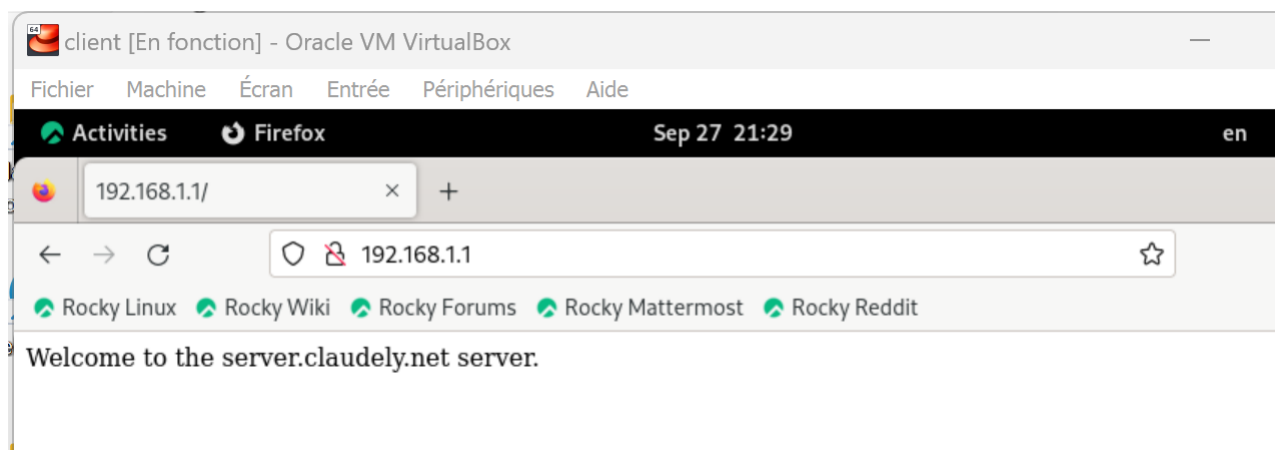
Теперь перезапустим HTTP-сервер (рис. 4.12):

```
systemctl restart httpd
```

```
[root@server.claudely.net www.claudely.net]#  
[root@server.claudely.net www.claudely.net]# chown -R apache:apache /var/www  
[root@server.claudely.net www.claudely.net]# restorecon -vR /etc  
[root@server.claudely.net www.claudely.net]# restorecon -vR /var/named  
[root@server.claudely.net www.claudely.net]# restorecon -vR /var/www  
[root@server.claudely.net www.claudely.net]#  
[root@server.claudely.net www.claudely.net]# systemctl restart httpd  
[root@server.claudely.net www.claudely.net]#
```

**Рис. 4.12.** Исправление прав доступа в каталог с веб-контентом, восстановление контекста безопасности в SELinux и перезапуск HTTP-сервера.

На виртуальной машине client убедимся в корректном доступе к веб-серверу по адресам `server.claudely.net` (рис. 4.13) и `www.claudely.net` (рис. 4.14) в адресной строке веб-браузера.



**Рис. 4.13.** Проверка корректного доступа на виртуальной машине client к веб-серверу по адресу `server.claudely.net`.

На виртуальной машине server перейдём в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём

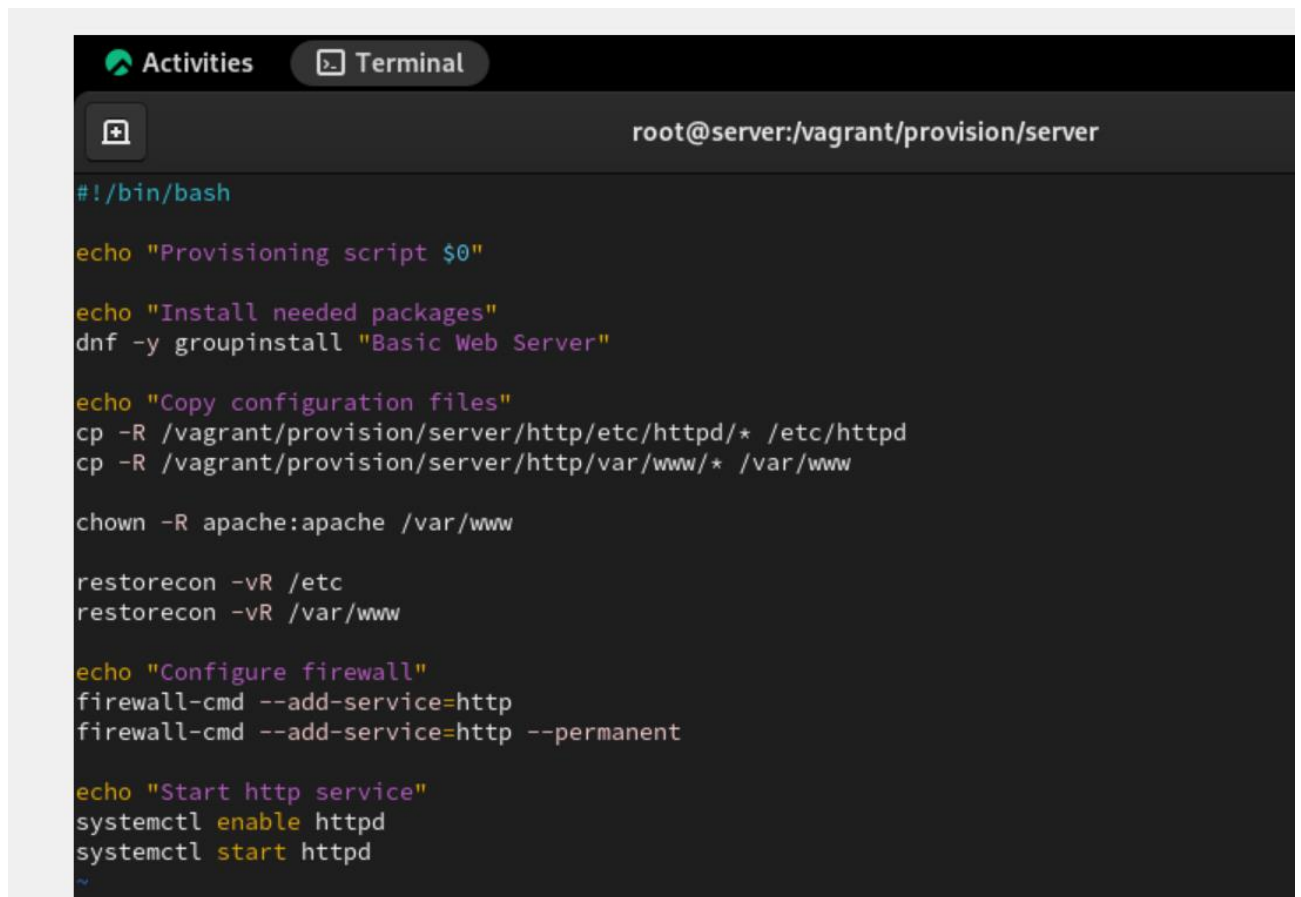
каталог `http`, в который поместим в соответствующие подкаталоги конфигурационные файлы HTTP-сервера. Теперь заменим конфигурационные файлы DNS-сервера. В каталоге `/vagrant/provision/server` создадим исполняемый файл `http.sh` (рис. 5.1):

```
[root@server.claudely.net server]#  
[root@server.claudely.net server]# cd /vagrant/provision/server/dns/  
[root@server.claudely.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/  
cp: overwrite '/vagrant/provision/server/dns/var/named/data/named.run'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/dynamic/managed-keys.bind'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/dynamic/managed-keys.bind.jnl'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/claudey.net.jnl'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/claudey.net'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1.jnl'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/named.ca'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/named.empty'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/named.localhost'? yes  
cp: overwrite '/vagrant/provision/server/dns/var/named/named.loopback'? yes  
[root@server.claudely.net dns]#
```

**Рис. 5.1.** Открытие на виртуальной машине `server` каталога для внесения изменений в настройки внутреннего окружения, создание в нём каталога `http`.

Замена конфигурационных файлов DNS-сервера и создание исполняемого файла `http.sh`.

Откроем созданный файл на редактирование и пропишем в нём скрипт (рис. 5.2):



The image shows a terminal window titled "Terminal" with the prompt "root@server:/vagrant/provision/server". The terminal displays a series of commands and their outputs, which are part of a provisioning script. The commands include installing packages, copying configuration files, setting permissions, restoring permissions, configuring the firewall, and starting the http service.

```
#!/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

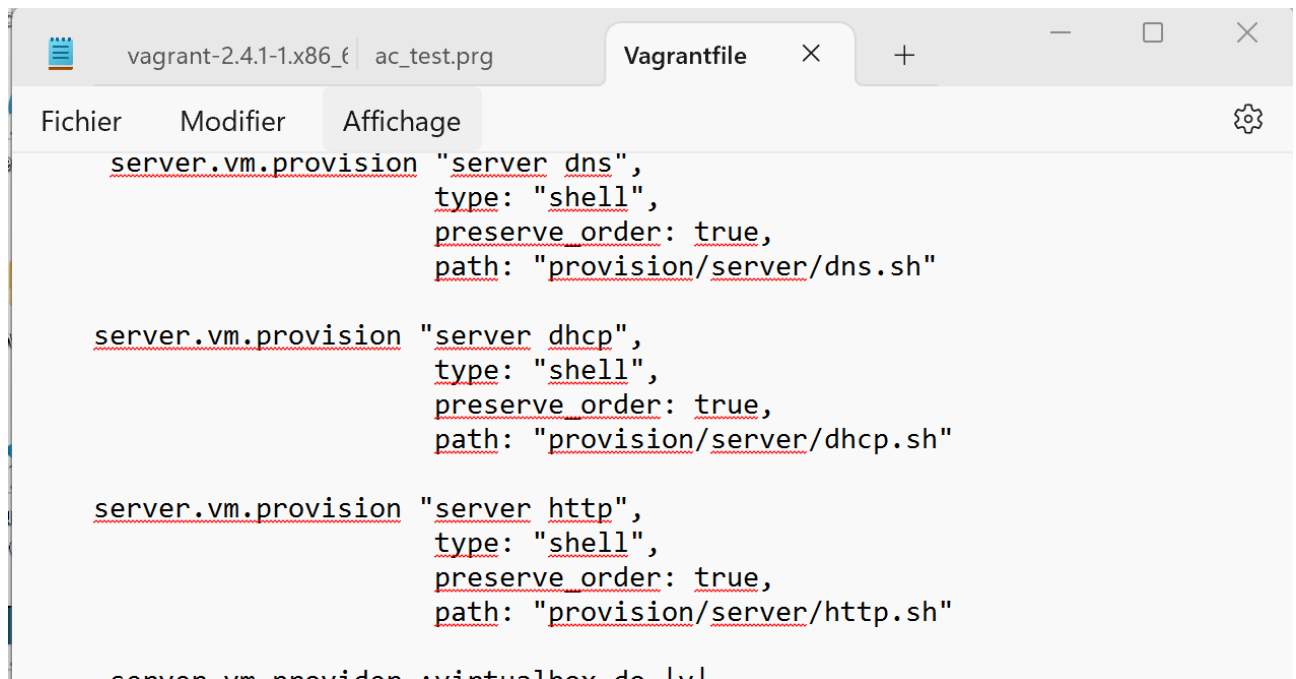
echo "Configure firewall"
firewall-cmd --add-service=http
firewall-cmd --add-service=http --permanent

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

**Рис. 5.2.** Открытие созданного файла на редактирование и прописывание скрипта.

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile добавим в конфигурации сервера следующую запись (рис. 5.3):





**Рис. 5.3.** Добавление записи для отработки созданного скрипта во время загрузки виртуальных машин.

### **Вывод:**

В ходе выполнения лабораторной работы были приобретены практические навыки по установке и базовому конфигурированию HTTP-сервера Apache.

### **Ответы на контрольные вопросы:**

1. Через какой порт по умолчанию работает Apache? - **По умолчанию, Apache работает через порт 80 для HTTP и порт 443 для HTTPS.**
2. Под каким пользователем запускается Apache и к какой группе относится этот пользователь? - **Apache обычно запускается от имени пользователя www-data (или apache, в зависимости от дистрибутива) и относится к группе с тем же именем.**
3. Где располагаются лог-файлы веб-сервера? Что можно по ним отслеживать? - **Лог-файлы веб-сервера обычно располагаются в директории логов. Например, в Ubuntu логи Apache хранятся в /var/log/apache2/, а в CentOS - в /etc/httpd/logs/. Лог-файлы**

содержат информацию о запросах к серверу, ошибки, статусы запросов и другие события, что позволяет администраторам отслеживать активность и выявлять проблемы.

4. Где по умолчанию содержится контент веб-серверов? - Контент веб-серверов по умолчанию обычно находится в директории, называемой "DocumentRoot". Например, в Apache на Linux DocumentRoot по умолчанию установлен в /var/www/html/. В этой директории содержатся файлы, которые веб-сервер отдает при запросах.
5. Каким образом реализуется виртуальный хостинг? Что он даёт? - Виртуальный хостинг в Apache позволяет hostить несколько сайтов на одном сервере. Это можно сделать двумя способами:

**Базовый виртуальный хостинг:** Разные сайты обслуживаются на одном сервере с использованием разных директорий DocumentRoot.

**Name-based (или Host-based) виртуальный хостинг:** Разные сайты обслуживаются на одном IP-адресе, но на разных доменных именах. Это основывается на значении заголовка "Host" в HTTP-запросе, который используется для определения, какой виртуальный хост должен обработать запрос.

Виртуальный хостинг позволяет хозяину сервера размещать несколько сайтов на одном физическом сервере, управлять ими независимо, и предоставлять услуги хостинга для различных клиентов или проектов.