

Отчет по лабораторной работе №4

Базовая настройка HTTP-сервера Apache

Галацан Николай, НПИбд-01-22

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Установка HTTP-сервера	5
2.2	Базовое конфигурирование HTTP-сервера	5
2.3	Анализ работы HTTP-сервера	7
2.4	Настройка виртуального хостинга для HTTP-сервера	9
2.5	Внесение изменений в настройки внутреннего окружения виртуальной машины	12
3	Выводы	14
4	Ответы на контрольные вопросы	15

Список иллюстраций

2.1	Конфигурационный файл <code>/etc/httpd/conf/httpd.conf</code>	6
2.2	Внесение изменений в настройки межсетевого экрана	6
2.3	Расширенный лог системных сообщений	7
2.4	Тестовая страницы HTTP-сервера	8
2.5	Запись в мониторинге доступа	8
2.6	Запись в логе ошибок	9
2.7	Добавление записи в конец файла прямой DNS-зоны	9
2.8	Добавление записи в конец файла обратной DNS-зоны	10
2.9	Удаление файлов журналов DNS	10
2.10	Редактирование <code>server.ngalacan.net.conf</code>	11
2.11	Редактирование <code>www.ngalacan.net.conf</code>	11
2.12	Создание каталога и файла главной страницы	11
2.13	Корректирование прав доступа, восстановление контекста безопасности, перезагрузка <code>httpd</code>	11
2.14	Доступ к <code>server.ngalacan.net</code>	12
2.15	Доступ к <code>www.ngalacan.net</code>	12
2.16	Создание скрипта <code>http.sh</code>	13

1 Цель работы

Приобретение практических навыков по установке и базовому конфигурированию HTTP-сервера Apache.

2 Выполнение лабораторной работы

2.1 Установка HTTP-сервера

Запускаю ВМ через рабочий каталог. На ВМ `server` захожу под собственным пользователем и перехожу в режим суперпользователя. Устанавливаю стандартный веб-сервер:

```
LANG=C yum grouplist  
dnf -y groupinstall "Basic Web Server"
```

2.2 Базовое конфигурирование HTTP-сервера

Просматриваю содержание конфигурационных файлов в каталогах `/etc/httpd/conf` и `/etc/httpd/conf.d` (рис. 2.1).

```
root@server:/etc/httpd/conf
root@server conf]# cat /etc/httpd/conf/httpd.conf

#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# See the httpd.conf(5) man page for more information on this configuration,
# and httpd.service(8) on using and configuring the httpd service.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.
```

Рис. 2.1: Конфигурационный файл /etc/httpd/conf/httpd.conf

Вношу изменения в настройки межсетевого экрана узла server, разрешив работу с http (рис. 2.2)

```
root@server:~
t imap imaps ipfs ipp ipp-client ipsec 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-contro
ller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure
kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-t
ls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix m
dns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysq
l nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvp
n ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi
pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter prox
y-dhcp ps2link ps3netrvr ptp pulseaudio puppetmaster quassel radius rdp redis re
dis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client sam
ba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snm
ptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn sy
ncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle
tftp tile38 tinc tor-socks transmission-client upnp-client vdsms vnc-server warpi
nator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discove
ry-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp
-server zabbix-agent zabbix-server zerotier
[root@server ~]# firewall-cmd --add-service=http
success
[root@server ~]# firewall-cmd --add-service=http --permanent
success
[root@server ~]#
```

Рис. 2.2: Внесение изменений в настройки межсетевого экрана

Во втором терминале открываю расширенный лог системных сообщений. В первом терминале активирую и запускаю HTTP-сервер:

```
systemctl enable httpd
```

```
systemctl start httpd
```

В лог системных сообщений вижу, что веб-сервер запущен (рис. 2.3)

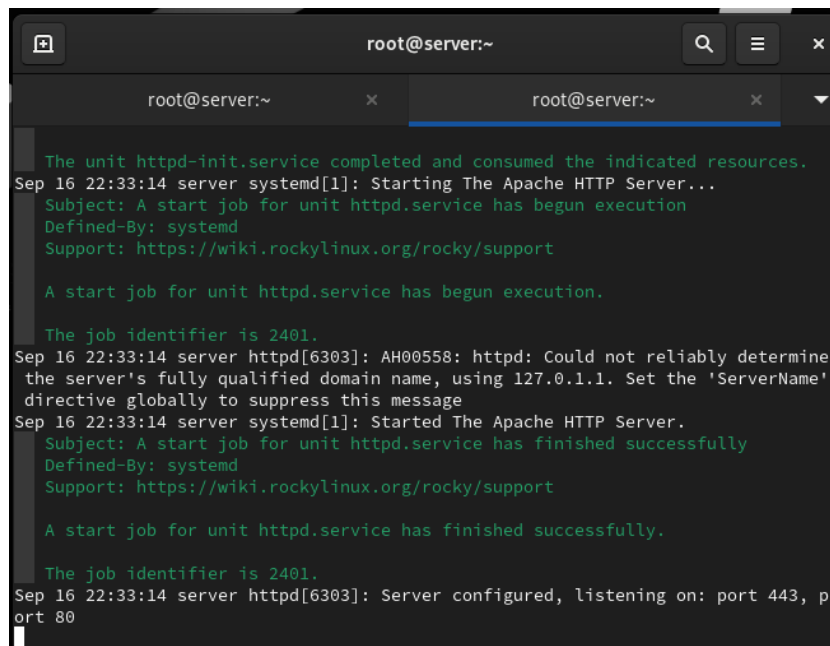


Рис. 2.3: Расширенный лог системных сообщений

2.3 Анализ работы HTTP-сервера

На виртуальной машине server открываю лог ошибок работы веб-сервера: `tail -f /var/log/httpd/error_log`; мониторинг доступа к веб-серверу: `tail -f /var/log/httpd/access_log`.

Запустив VM client, открываю браузер и в адресной строке ввожу 192.168.1.1 (рис. 2.4)

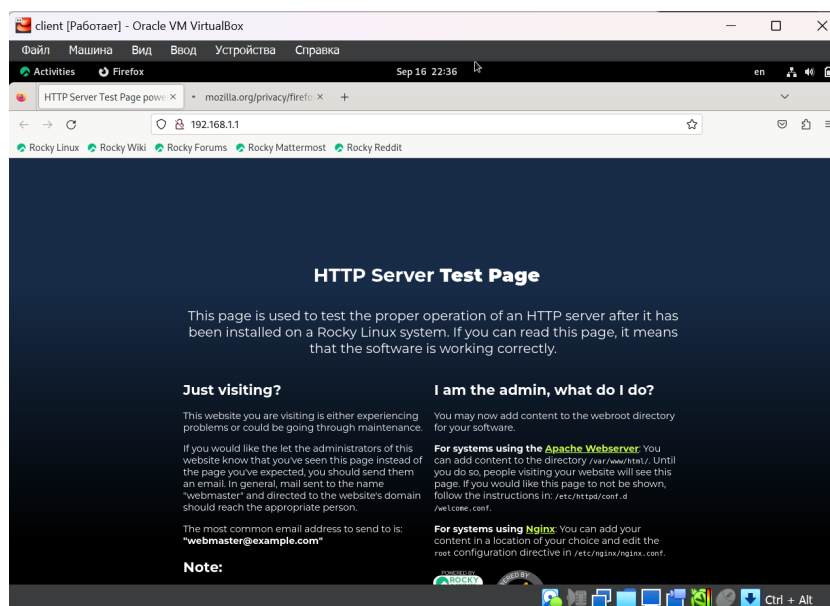


Рис. 2.4: Тестовая страницы HTTP-сервера

Вижу записи в мониторинге доступа (рис. 2.5) и в логе ошибок о подключении к серверу (рис. 2.6).

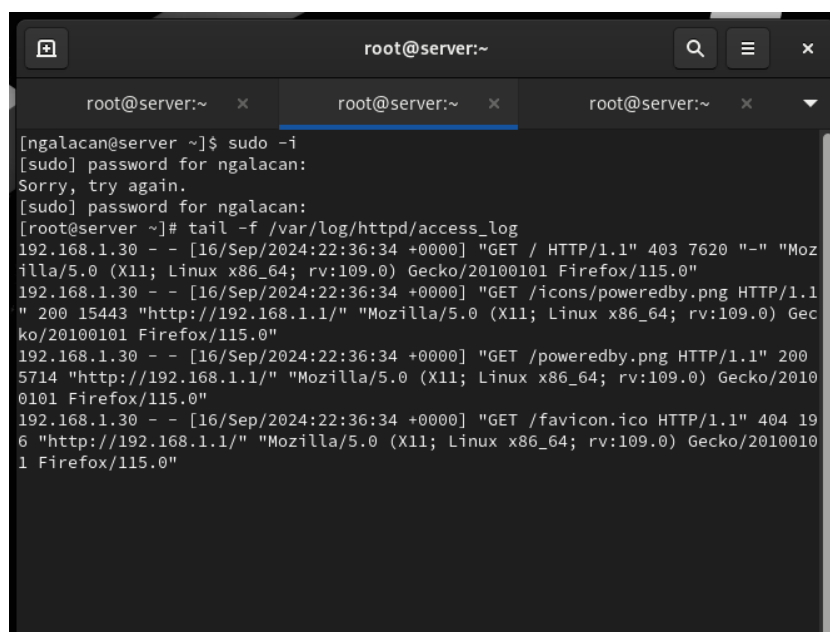
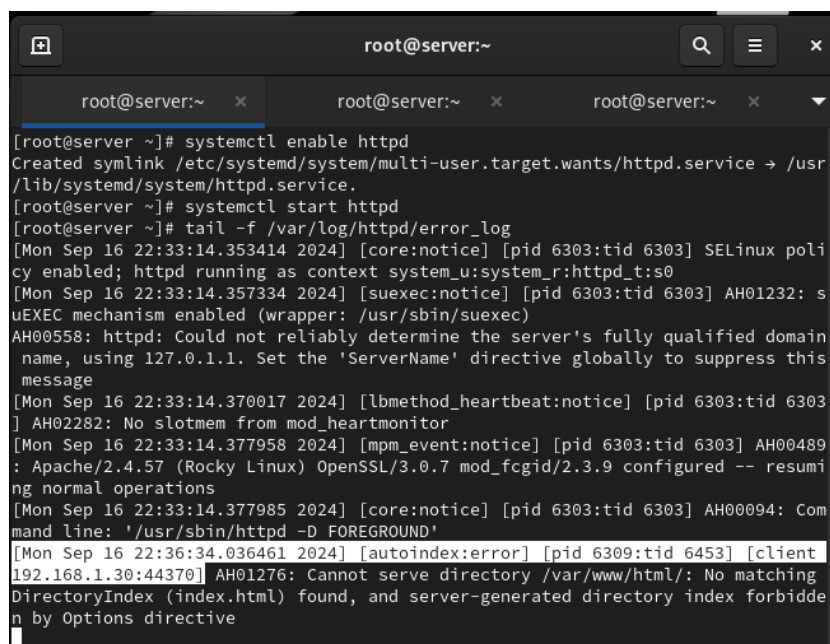


Рис. 2.5: Запись в мониторинге доступа

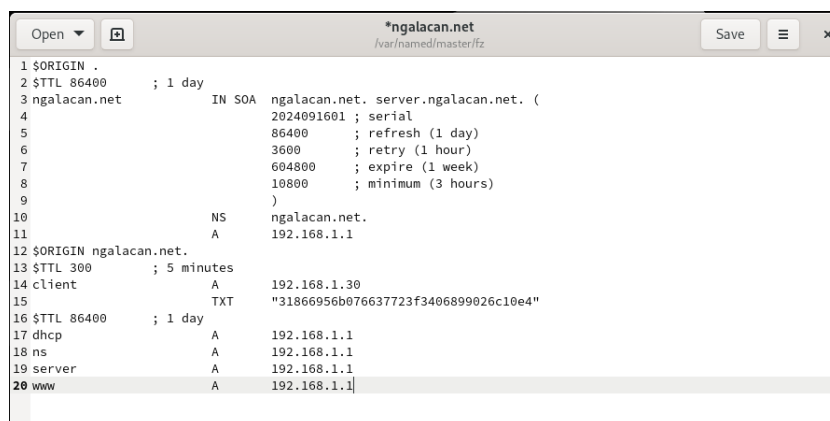


```
root@server:~  
[root@server ~]# systemctl enable httpd  
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.  
[root@server ~]# systemctl start httpd  
[root@server ~]# tail -f /var/log/httpd/error_log  
[Mon Sep 16 22:33:14.353414 2024] [core:notice] [pid 6303:tid 6303] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0  
[Mon Sep 16 22:33:14.357334 2024] [suexec:notice] [pid 6303:tid 6303] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)  
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message  
[Mon Sep 16 22:33:14.370017 2024] [lbmethod_heartbeat:notice] [pid 6303:tid 6303] AH02282: No slotmem from mod_heartbeat  
[Mon Sep 16 22:33:14.377958 2024] [mpm_event:notice] [pid 6303:tid 6303] AH00489: Apache/2.4.57 (Rocky Linux) OpenSSL/3.0.7 mod_fcgid/2.3.9 configured -- resuming normal operations  
[Mon Sep 16 22:33:14.377985 2024] [core:notice] [pid 6303:tid 6303] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'  
[Mon Sep 16 22:36:34.036461 2024] [autoindex:error] [pid 6309:tid 6453] [client 192.168.1.30:44370] AH01276: Cannot serve directory /var/www/html/: No matching DirectoryIndex (index.html) found, and server-generated directory index forbidden by Options directive
```

Рис. 2.6: Запись в логе ошибок

2.4 Настройка виртуального хостинга для HTTP-сервера

Останавливаю работу DNS-сервера. Добавляю запись для HTTP-сервера в конце файла прямой DNS-зоны (рис. 2.7).



```
*ngalacan.net  
/var/named/master/fz  
1 $ORIGIN .  
2 $TTL 86400 ; 1 day  
3 ngalacan.net IN SOA ngalacan.net. server.ngalacan.net. (  
4 2024091601 ; serial  
5 86400 ; refresh (1 day)  
6 3600 ; retry (1 hour)  
7 604800 ; expire (1 week)  
8 10800 ; minimum (3 hours)  
9 )  
10 NS ngalacan.net.  
11 A 192.168.1.1  
12 $ORIGIN ngalacan.net.  
13 $TTL 300 ; 5 minutes  
14 client A 192.168.1.30  
15 TXT "31866956b076637723f3406899026c10e4"  
16 $TTL 86400 ; 1 day  
17 dhcp A 192.168.1.1  
18 ns A 192.168.1.1  
19 server A 192.168.1.1  
20 www A 192.168.1.1
```

Рис. 2.7: Добавление записи в конец файла прямой DNS-зоны

Добавляю также запись в конце файла обратной зоны (рис. 2.8)

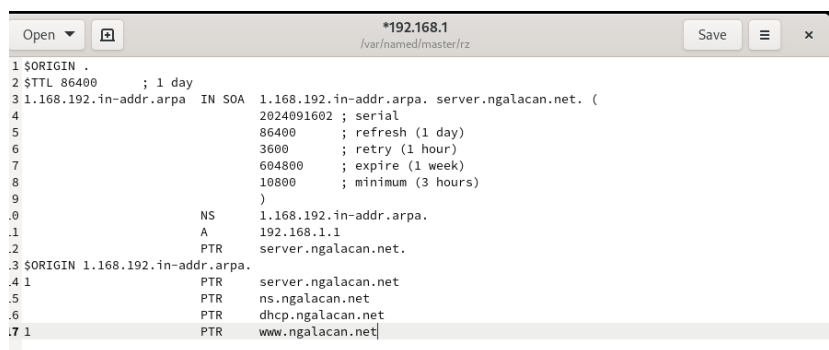


Рис. 2.8: Добавление записи в конец файла обратной DNS-зоны

Удаляю из соответствующих каталогов файлы журналов DNS (рис. 2.9).

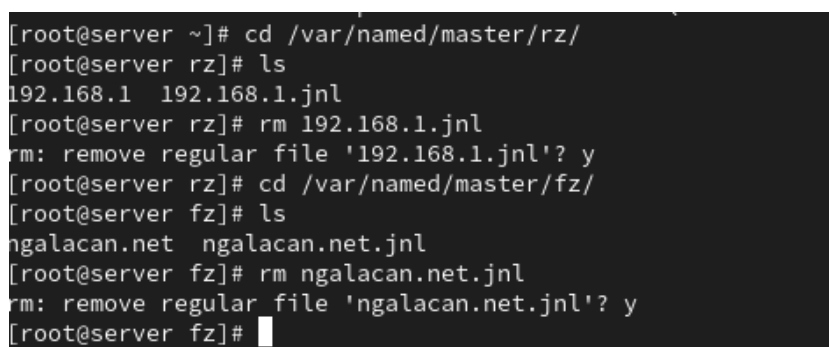


Рис. 2.9: Удаление файлов журналов DNS

Перезапускаю DNS-сервер. В каталоге /etc/httpd/conf.d создаю файлы server.ngalacan.net.conf и www.ngalacan.net.conf:

```

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

```

Редактирую server.ngalacan.net.conf (рис. 2.10).



Рис. 2.10: Редактирование server.ngalacan.net.conf

Редактирую www.ngalacan.net.conf (рис. 2.11).



Рис. 2.11: Редактирование www.ngalacan.net.conf

Перехожу в /var/www/html создаю каталог server.ngalacan.net и в нем файл index.html (рис. 2.12). Ввожу в файл сообщение *Welcome to the server.ngalacan.net server*. Аналогично создаю каталог и файл для www.ngalacan.net.

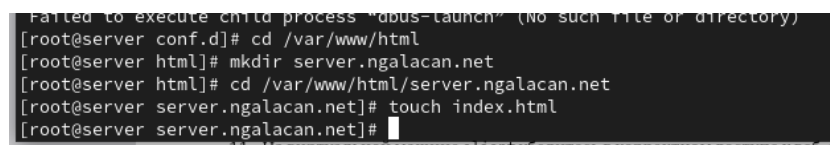


Рис. 2.12: Создание каталога и файла главной страницы

Корректирую права доступа в каталог с веб-контентом, восстанавливаю контекст безопасности в SELinux, перезагружаю HTTP-сервер (рис. 2.13)

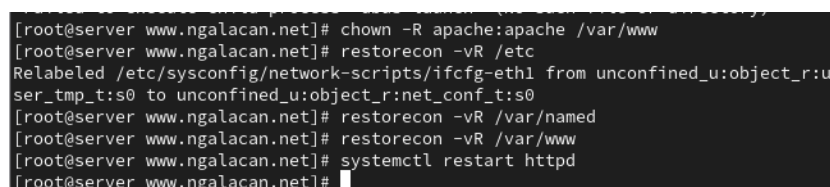


Рис. 2.13: Корректирование прав доступа, восстановление контекста безопасности, перезагрузка httpd

На VM client ввожу в адресную строку браузера `server.ngalacan.net` (рис. 2.14) и `www.ngalacan.net` (рис. 2.15).

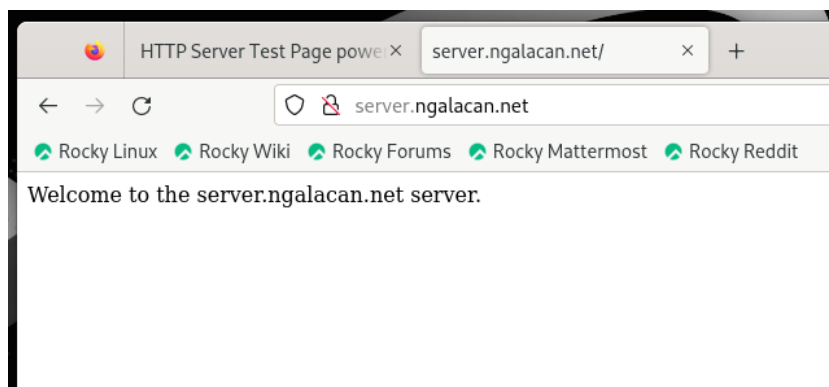


Рис. 2.14: Доступ к `server.ngalacan.net`

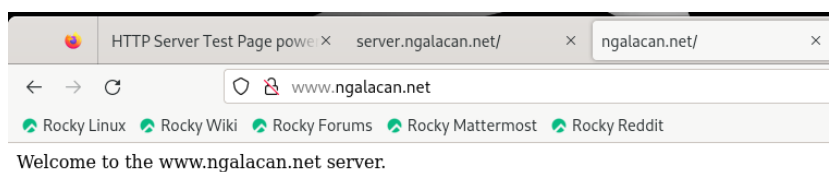


Рис. 2.15: Доступ к `www.ngalacan.net`

2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

На VM server перехожу в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создаю в нём каталог `http`, в который помещаю в соответствующие подкаталоги конфигурационные файлы HTTP-сервера:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/http/etc/httpd/conf.d
```

```
mkdir -p /vagrant/provision/server/http/var/www/html
cp -R /etc/httpd/conf.d/*
↪ /vagrant/provision/server/http/etc/httpd/conf.d/
cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
```

Заменяю конфигурационные файлы DNS-сервера:

```
cd /vagrant/provision/server/dns/
cp -R /var/named/* /vagrant/provision/server/dns/var/named/
```

В каталоге /vagrant/provision/server создаю исполняемый файл http.sh (рис. 2.16).

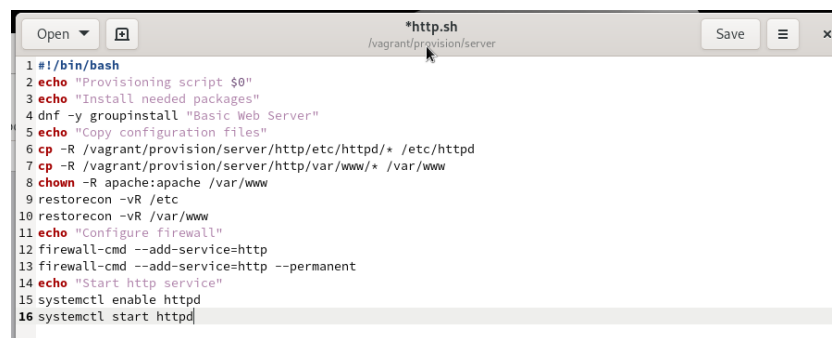


Рис. 2.16: Создание скрипта http.sh

Для отработки скрипта во время запуска добавляю в Vagrantfile в разделе конфигурации для сервера

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

После этого выключаю ВМ:

```
vagrant halt client
vagrant halt server
```

3 Выводы

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

4 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 хостить несколько сайтов на одном сервере. 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.