

# **Лабораторная работа №4**

**Базовая настройка HTTP-сервера Apache**

Андреева Софья Владимировна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Установка HTTP-сервера . . . . .	6
3.2	Базовое конфигурирование HTTP-сервера . . . . .	6
3.3	Анализ работы HTTP-сервера . . . . .	9
3.4	Настройка виртуального хостинга для HTTP-сервера . . . . .	10
3.5	Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	14
<b>4</b>	<b>Контрольные вопросы</b>	<b>16</b>
<b>5</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

3.1	Установка стандартного веб-сервера . . . . .	6
3.2	conf . . . . .	7
3.3	conf.d . . . . .	7
3.4	Разрешение работы с http . . . . .	8
3.5	Запуск лога системных сообщений . . . . .	8
3.6	Запуск веб-сервера . . . . .	9
3.7	Тестовая страница веб-сервера . . . . .	9
3.8	Лог ошибок . . . . .	9
3.9	Информация мониторинга . . . . .	10
3.10	Внесение содержания файла server.svandreeva.net.conf . . . . .	11
3.11	Внесение содержания файла www.svandreeva.net.conf . . . . .	11
3.12	Внесение содержания файла index.html для server.svandreeva.net .	12
3.13	Внесение содержания файла index.html для www.svandreeva.net .	12
3.14	server.svandreeva.net . . . . .	13
3.15	www.svandreeva.net . . . . .	13
3.16	Создание окружения для внесения изменений в настройки окру- жающей среды . . . . .	14
3.17	Содержание http.sh . . . . .	14
3.18	Изменение файла Vagrantfile . . . . .	15

# 1 Цель работы

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

## 2 Задание

1. Установите необходимые для работы HTTP-сервера пакеты.
2. Запустите HTTP-сервер с базовой конфигурацией и проанализируйте его работу.
3. Настройте виртуальный хостинг.
4. Напишите 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

### 3.1 Установка HTTP-сервера

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

```
cd /var/tmp/svandreeva/vagran
```

Затем запустим виртуальную машину server:

```
make server-up
```

На виртуальной машине server войдем под созданным в предыдущей работе пользователем и откроем терминал. Перейдем в режим суперпользователя и установим стандартный веб-сервер(рис. fig. 3.1):



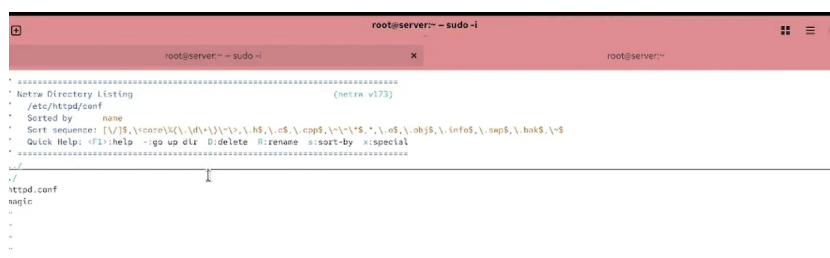
```
[root@server.svandreeva.net ~]# LANG=C yum groupinstall
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
[ ] 34 kB/s | 44 kB 00:03
] --- B/s | 0 B --- ETA
```

Рис. 3.1: Установка стандартного веб-сервера

### 3.2 Базовое конфигурирование HTTP-сервера

Просмотрим содержание конфигурационных файлов в каталогах `/etc/httpd/conf` и `/etc/httpd/conf.d`. В каталоге `/etc/httpd/conf` лежат файлы `httpd.conf` и `magic`. Первый – это основной файл конфигурации HTTP-сервера Apache. Он содержит директивы конфигурации, которые дают серверу инструкции. Второй – данные

для модуля `mod_mime_magic`, этот модуль определяет тип MIME файлов так же, как работает команда Unix `file(1)`: она просматривает первые несколько байтов файла. Он задуман как «вторая линия защиты» в случаях, которые `mod_mime` не может разрешить. Этот модуль создан на основе бесплатной версии команды `file(1)` для Unix, которая использует «магические числа» и другие подсказки по содержимому файла, чтобы выяснить, что это за содержимое. В каталоге `/etc/httpd/conf.d` лежат файлы `autoindex.conf`(настраивает листинг директорий по http, средствами веб-сервера), `fcgid.conf`(настраивает клиент-серверный протокол взаимодействия веб-сервера и приложения), `manual.conf`(позволяет получить доступ к руководству по адресу `http://localhost/manual/`), `ssl.conf`(SSL-конфигурация, SSL – это протокол для безопасной передачи кодированных данных между веб-браузером и веб-сервером.), `userdir.conf`(конфигурация `userdir` - позволяет пользователям размещать материалы на сайте, без предоставления доступа к директориям Web-сервера), `welcome.conf`(включает страницу «Добро пожаловать» по умолчанию, если она есть).

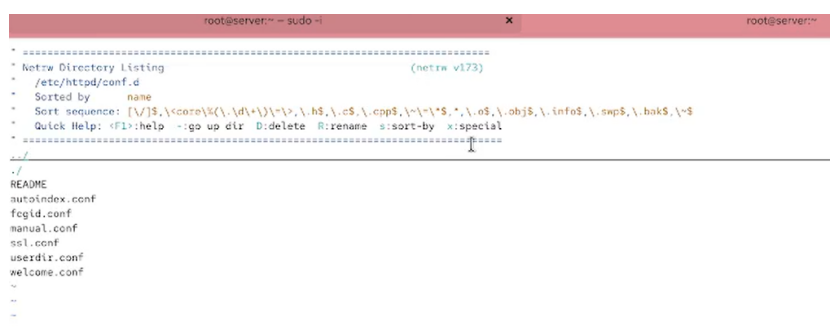


```

root@server:~# sudo -i
root@server:~# ls -la /etc/httpd/conf
total 12
drwxr-xr-x 2 root root 4096 Jun 10 12:12 .
drwxr-xr-x 1 root root 4096 Jun 10 12:12 ..
-rw-r--r-- 1 root root 1219 Jun 10 12:12 httpd.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 magic
-rw-r--r-- 1 root root 1219 Jun 10 12:12 ssl.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 userdir.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 welcome.conf

```

Рис. 3.2: conf



```

root@server:~# sudo -i
root@server:~# ls -la /etc/httpd/conf.d
total 12
drwxr-xr-x 2 root root 4096 Jun 10 12:12 .
drwxr-xr-x 1 root root 4096 Jun 10 12:12 ..
-rw-r--r-- 1 root root 1219 Jun 10 12:12 autoindex.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 fcgid.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 manual.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 ssl.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 userdir.conf
-rw-r--r-- 1 root root 1219 Jun 10 12:12 welcome.conf

```

Рис. 3.3: conf.d

[illegible]

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы(рис. fig. 3.5):

```
#1 0x000000000041dc94 n/a (n/a + 0x0)
#2 0x00000000000045841c n/a (n/a + 0x0)
#3 0x000000000000435d08 n/a (n/a + 0x0)
#4 0x00007f9285c54b68 start_thread (libc.so.6 + 0x94b68)
#5 0x00007f9285c596bc __clone2 (libc.so.6 + 0x1056bc)

Stack trace of thread 6587:
#0 0x00007f9285c34bd syscall (libc.so.6 + 0x1034bd)
#1 0x000000000000434c30 n/a (n/a + 0x0)
#2 0x000000000000450bf8 n/a (n/a + 0x0)
#3 0x00000000000043566a n/a (n/a + 0x0)
#4 0x00000000000045841c n/a (n/a + 0x0)
#5 0x000000000000435d08 n/a (n/a + 0x0)
#6 0x00007f9285c54b68 start_thread (libc.so.6 + 0x94b68)
#7 0x00007f9285c596bc __clone2 (libc.so.6 + 0x1056bc)

Stack trace of thread 6586:
#0 0x00007f9285c34bd syscall (libc.so.6 + 0x1034bd)
#1 0x0000000000004344e2 n/a (n/a + 0x0)
#2 0x000000000000450066 n/a (n/a + 0x0)
#3 0x000000000000405123 n/a (n/a + 0x0)
#4 0x00007f9285bea30e __libc_start_call_main (libc.so.6 + 0x2a30e)
#5 0x00007f9285bea3c9 __libc_start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)
#6 0x0000000000004044aa n/a (n/a + 0x0)
ELF object binary architecture: AMD x86-64
```

Subject: Процесс 6586 (YBoxClient) сбросил данные памяти  
 Defined By: systemd  
 Support: <https://wiki.rockylinux.org/rocky/support>  
 Documentation: man:core(5)

Процесс 6586 (YBoxClient) завершился из-за критической ошибки.  
 Записи данных памяти.

Вероятно, это произошло из-за ошибки, допущенной в коде программы.

В первом терминале активируем и запустим НТТР-сервер следующими командами:

Просмотрим расширенный лог системных сообщений, чтобы убедиться, что веб-сервер успешно запустился(рис. fig. 3.6):



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

Рис. 3.6: Запуск веб-сервера

### 3.3 Анализ работы HTTP-сервера

Запустим виртуальную машину client:

make client-up

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

Затем виртуальной машине client запустим браузер и в адресной строке введем 192.168.1.1.(рис. fig. 3.7):

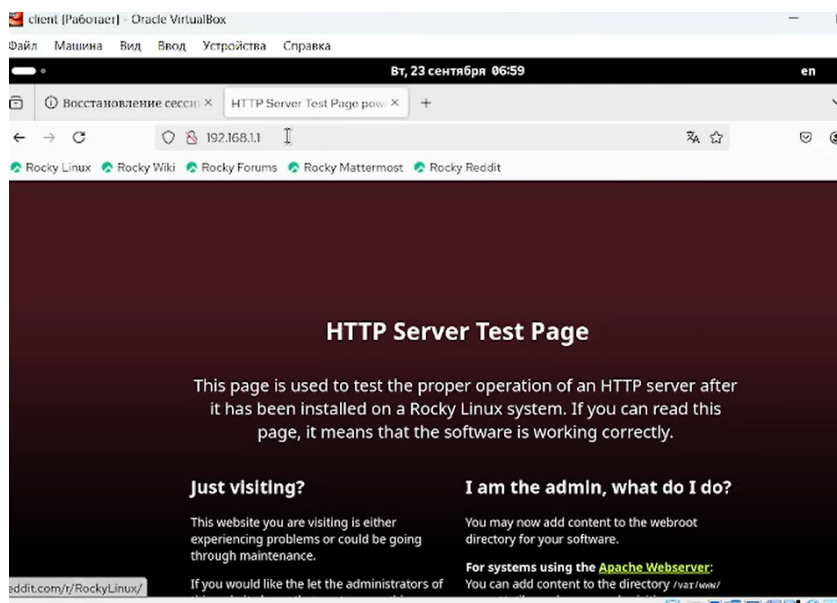


Рис. 3.7: Тестовая страница веб-сервера

Посмотрим информацию, отразившуюся при мониторинге(fig. 3.8, fig. 3.9):

```
[root@server.svandreeva.net ~]# tail -f /var/log/httpd/error_log
[Tue Sep 23 05:38:58.679259 2025] [suexec:notice] [pid 6754:tid 6754] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Tue Sep 23 05:38:58.907301 2025] [lbmethod:heartbeat:notice] [pid 6754:tid 6754] AH02282: No slotmem from mod_heartbeat
[Tue Sep 23 05:38:58.911072 2025] [systemd:notice] [pid 6754:tid 6754] SELinux policy enabled: httpd running as config
[Tue Sep 23 05:38:58.916667 2025] [mpm_event:notice] [pid 6754:tid 6754] AH00489: Apache/2.4.63 (Rocky Linux) OpenSSL/3.0.15
al operations
[Tue Sep 23 05:38:58.916707 2025] [core:notice] [pid 6754:tid 6754] AH00094: Command line: '/usr/sbin/httpd -D FOREGR
^Z
```

Рис. 3.8: Лог ошибок

```

[root@server.svandreeva.net ~]# tail -f /var/log/httpd/access_log
192.168.1.2 - - [23/Sep/2025:05:59:33 +0000] "GET / HTTP/1.1" 403 2620 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
192.168.1.2 - - [23/Sep/2025:05:59:33 +0000] "GET /icons/poweredby.png HTTP/1.1" 200 15443 "http://192.168.1.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
192.168.1.2 - - [23/Sep/2025:05:59:33 +0000] "GET /poweredby.png HTTP/1.1" 200 5714 "http://192.168.1.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
192.168.1.2 - - [23/Sep/2025:05:59:33 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://192.168.1.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"

```

Рис. 3.9: Информация мониторинга

Можно увидеть ip-адрес устройства, зашедшего на веб-сервер, дату доступа, версию браузера, информацию об устройстве(его ОС и архитектура).

### 3.4 Настройка виртуального хостинга для HTTP-сервера

Остановим работу DNS-сервера для внесения изменений в файлы описания DNS-зон:

```
systemctl stop named
```

Добавим запись для HTTP-сервера в конце файла прямой DNS-зоны /var/named/master/fz/svandreeva.net:

```
server A 192.168.1.1
```

```
www A 192.168.1.1
```

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

```
1 PTR server.svandreeva.net.
```

```
1 PTR www.svandreeva.net.
```

Также удалим из этих каталогов файлы журналов DNS: user.net.jnl и 192.168.1.jnl.

Затем перезапустим DNS-сервер командой:

```
systemctl start named
```

В каталоге /etc/httpd/conf.d создадим файлы server.svandreeva.net.conf и www.svandreeva.net.conf командами:

```
cd /etc/httpd/conf.d
touch server.svandreeva.net.conf
touch www.svandreeva.net.conf
```

Откроем на редактирование файл `server.svandreeva.net.conf` и внесем следующее содержание(fig. 3.10):



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

Рис. 3.10: Внесение содержания файла `server.svandreeva.net.conf`

Откроем на редактирование файл `www.svandreeva.net.conf` и внесем следующее содержание(рис. fig. 3.11):



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

Рис. 3.11: Внесение содержания файла `www.svandreeva.net.conf`

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

```
cd /var/www/html
mkdir server.svandreeva.net
```

```
cd /var/www/html/server.svandreeva.net
touch index.htm
```

Откроем на редактирование файл index.html и внесем следующее содержание (рис. fig. 3.12):



Рис. 3.12: Внесение содержания файла index.html для server.svandreeva.net

Для виртуального веб-сервера www.svandreeva.net:

```
cd /var/www/html
mkdir www.svandreeva.net
cd /var/www/html/www.svandreeva.net
touch index.htm
```

Откроем на редактирование файл index.html и внесем следующее содержание (рис. fig. 3.13):



Рис. 3.13: Внесение содержания файла index.html для www.svandreeva.net

Теперь скопируем права доступа в каталог с веб-контентом командой:

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

Затем восстановим контекст безопасности:

```
restorecon -vR /etc
```

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

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

И теперь перезапустим HTTP-сервер командой `systemctl restart httpd`.

На виртуальной машине client убедимся в корректном доступе к веб-серверу по адресам `server.svandreeva.net` и `www.svandreeva.net` в адресной строке веб-браузера (рис. fig. 3.14, fig. 3.15):

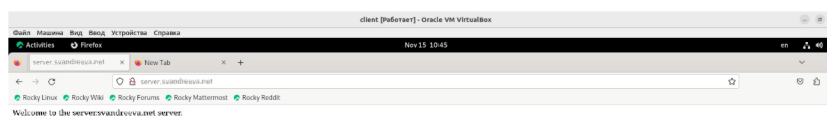


Рис. 3.14: server.svandreeva.net



Рис. 3.15: www.svandreeva.net

## 3.5 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, затем заменим конфигурационные файлы DNS-сервера и в каталоге `/vagrant/provision/server` создадим исполняемый файл `http.sh`(рис. fig. 3.16)

```
[root@server.svandreeva.net ~]# cd /vagrant/provision/server
[root@server.svandreeva.net server]# ls
01-dummy.sh 02-forward.sh 03-dns.sh 04-dhcp.sh 05-dns.sh 06-http.sh
[root@server.svandreeva.net server]# chmod +x http.sh
[root@server.svandreeva.net server]# http.sh
```

Рис. 3.16: Создание окружения для внесения изменений в настройки окружающей среды

Открыв `http.sh` на редактирование, пропишем в нём следующий скрипт(fig. 3.17):

```
root@server:/vagrant/provision/server -- sudo -i
#!/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. 3.17: Содержание `http.sh`

Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` добавим в разделе конфигурации для сервера(fig. 3.18):

```

C: > work > svandreeva > vagrant > Vagrantfile
4   Vagrant.configure("2") do |config|
24   config.vm.define "server", autostart: false do |server|
25
30     server.ssh.insert_key = false
31     server.ssh.username = 'vagrant'
32     server.ssh.password = 'vagrant'
33
34     server.vm.network :private_network,
35                       ip: "192.168.1.1",
36                       virtualbox____intnet: true
37
38     server.vm.provision "server dummy",
39                       type: "shell",
40                       preserve_order: true,
41                       path: "provision/server/01-dummy.sh"
42
43     server.vm.provision "server dns",
44                       type: "shell",
45                       preserve_order: true,
46                       path: "provision/server/dns.sh"
47
48     server.vm.provision "server dhcp",
49                       type: "shell",
50                       preserve_order: true,
51                       path: "provision/server/dhcp.sh"
52     server.vm.provision "server http",
53                       type: "shell",
54                       preserve_order: true,
55                       path: "provision/server/http.sh"
56

```

Рис. 3.18: Изменение файла Vagrantfile

## 4 Контрольные вопросы

1. Через какой порт по умолчанию работает Apache?

По умолчанию Apache работает через порт 80.

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

Apache обычно запускается под пользователем “www-data” и относится к группе “www-data”.

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

Лог-файлы веб-сервера обычно располагаются в каталоге /var/log/apache2. Можно отслеживать доступ, ошибки, запросы и другую информацию.

4. Где по умолчанию содержится контент веб-серверов?

Контент веб-серверов по умолчанию содержится в каталоге /var/www/html.

5. Каким образом реализуется виртуальный хостинг? Что он даёт

Виртуальный хостинг реализуется через конфигурацию веб-сервера, позволяя одному серверу обслуживать несколько доменов. Он дает возможность размещать несколько веб-сайтов на одном сервере.



## **5 Выводы**

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