

# Лабораторная работа №4

## Базовая настройка HTTP-сервера Apache

Сахно Никита НФИбд-02-23

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Выполнение лабораторной работы.....	1
4	Контрольные вопросы.....	7
5	Выводы.....	8

### 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 HTTPсервера 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\nikita\work\study\nvsakhno\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, криptoутилиты и пр.):

```

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

```

## Вызов списка групп

```

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:

```

## Установка веб-сервера

### Базовое конфигурирование HTTP-сервера

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

```

RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula
bacula-client bb bpg bitcoin bitcoin-pool bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine ch
ckmk-agent cockpit collected condor-collector cratedb ctdb dhcp dhcpcv6 dhcpcv6-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
ipa-dap freeipa-daps freeipa-replication freeipa-ldap galera ganglia-master git gssd grafana &
re high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target ism jellyfin
jenkins admin kademaint kerberos Kibana klogin kpasswd kprop kshell kube-api kube-control-plane kube
-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler ku
be-scheduler-secure kubelet-worker kubelet-worker kubelet-worker ldap ldaps libvirt libvirt-tls lightning-netw
ork llmnr llmnr-tcp llmnr-udp managementv4 matrix mds memcache minidlna mongodb mosh mountd mqtt ms-wbt msqd
l murmur mysql nbd netbios-ns netdata-dashboard nfs nfsv3 nmea-0183 ntp ntp nut openvpn ovirt-imageo ovirt-storageco
nsole ovirt-vmconsole plesk pmproxy pmwebapi pmwebapi-pop3 pop3s postgresql privoxy prometheus-prometheus-node-e
xporter proxy-dhcp ps3netsrv ptb pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh
rsyncd rtsp salt-master samba samba-client samba-dc same sftp sftp-sasl smtp smtp-submission smtsp smptl smptl-s-trap smpttrap 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 vdm vnc-server wbem-
http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsmans xmppc xmpp-
bosk xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier

```

### Команда firewall

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

```
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.
```

## Расширенный лог системных сообщений

### Анализ работы HTTP-сервера

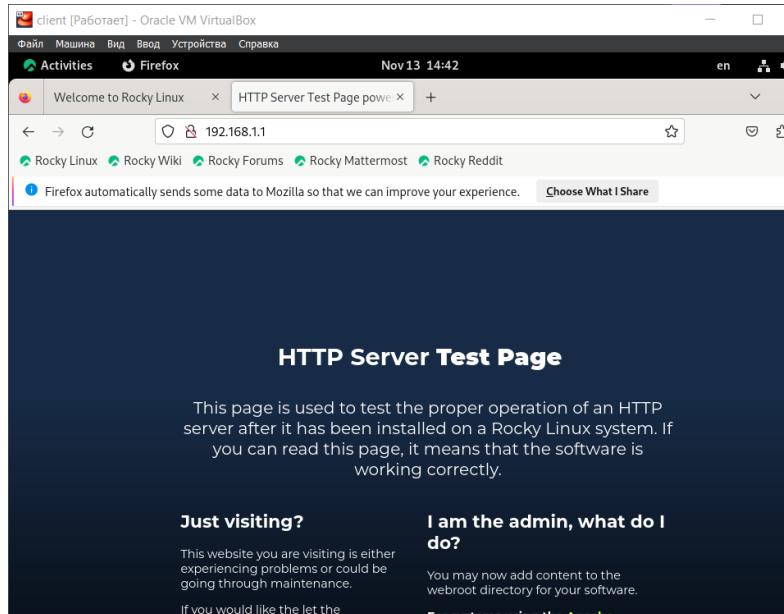
Запустим виртуальную машину client: `make client-up`.

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

```
tail -f /var/log/httpd/error_log
[core:notice] [pid 7252:tid 7252] SELinux policy enabled; httpd running as context
[suexec:notice] [pid 7252:tid 7252] AH01232: suEXEC mechanism enabled (wrapper: /us
[lbmethode_heartbeat:notice] [pid 7252:tid 7252] AH02282: No slotmem from mod_heartm
[mpm_event:notice] [pid 7252:tid 7252] AH00489: Apache/2.4.53 (Rocky Linux) OpenSSL
-- resuming normal operations
[core:notice] [pid 7252:tid 7252] AH00094: Command line: '/usr/sbin/httpd -D FOREGR
tail -f /var/log/httpd/access_log
```

### Лог ошибок и мониторинг доступа

На виртуальной машине client запустим браузер и в адресной строке введите `192.168.1.1`.



### Тестовая страница

В браузере открылась тестовая страница HTTP-сервера, на которой написано сообщение: “Если вы можете это читать, то ПО работает корректно”. Следовательно, базовое конфигурирование HTTP-сервера выполнено правильно.

```
192.168.1.30 - [13/Nov/2023:14:48:15 +0000] "GET / HTTP/1.1" 403 7620 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.1.30 - [13/Nov/2023:14:48:15 +0000] "GET /favicons/favicon.jpg HTTP/1.1" 200 579 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.1.30 - [13/Nov/2023:14:48:15 +0000] "GET /favicon.ico HTTP/1.1" 200 1556 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.1.30 - [13/Nov/2023:14:48:16 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "HTTP://192.168.1.1/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
```

*Результат мониторинга*

## **Настройка виртуального хостинга для HTTP-сервера**

Настроим виртуальный хостинг по двум DNS-адресам: server.nvsakhno.net и www.nvsakhno.net.

Для этого сначала 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/nvsakhno.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
```

*Файл прямой зоны*

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

```
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
```

*Файл обратной зоны*

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

```
.net master]# cd rz  
.net rz]# ls  
  
.net rz]# rm -r 192.168.1.jnl  
'192.168.1.jnl'? y  
.net rz]# ls  
  
.net rz]#
```

### *Удаление файлов журналов DNS*

Перезапустим DNS-сервер: `systemctl start named`

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

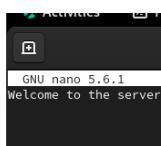
```
cd /etc/httpd/conf.d  
touch server.nvsakhno.net.conf  
touch www.nvsakhno.net.conf
```

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

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.nvsakhno.net` и `www.nvsakhno.net`. Для виртуального веб-сервера `server.nvsakhno.net`:

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

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



### *Редактирование файла*

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

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

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

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

Восстановим контекст безопасности в SELinux и перезапустим HTTP-сервер.

```
chown -R apache:apache /var/www  
restorecon -vR /etc  
restorecon -vR /var/named  
restorecon -vR /var/www  
systemctl restart httpd
```

### *Редактирование файла*

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

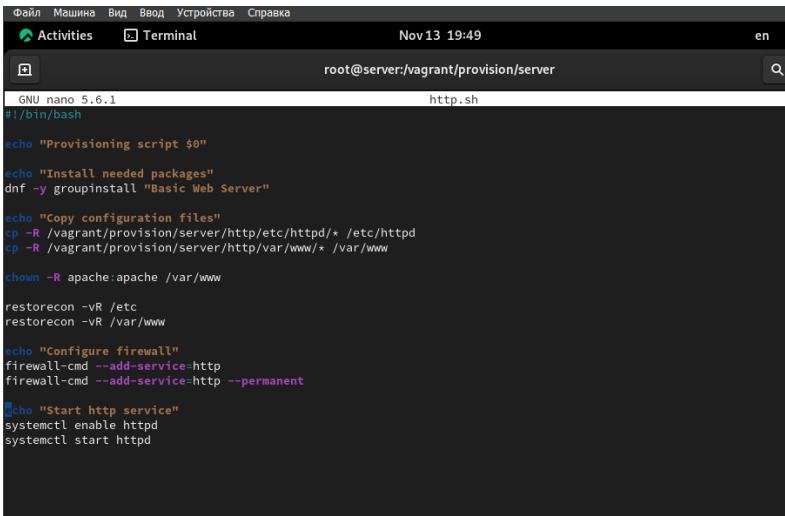
## **Внесение изменений в настройки внутреннего окружения виртуальной машины**

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. Заменим конфигурационные файлы DNS-сервера. В каталоге /vagrant/provision/server создадим исполняемый файл http.sh.

```
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
```

### *Окно терминала*

Открыв его на редактирование, пропишем в нём следующий скрипт:



```
Файл Машина Вид Ввод Устройства Справка
Activities Terminal Nov 13 19:49 en
root@server:/vagrant/provision/server
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
```

### 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

- Через какой порт по умолчанию работает Apache?

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

- 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.

- 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.

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