

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ
НАРОДОВ**

**Факультет физико-математических и естественных
наук**

Кафедра теории вероятностей и кибербезопасности

Отчет лабораторной работы 4

Дисциплина: Администрирование сетевых подсистем

Студент: Астахова Марина

Группа: НПИбд-02-23

Тема: Базовая настройка HTTP-сервера Apache

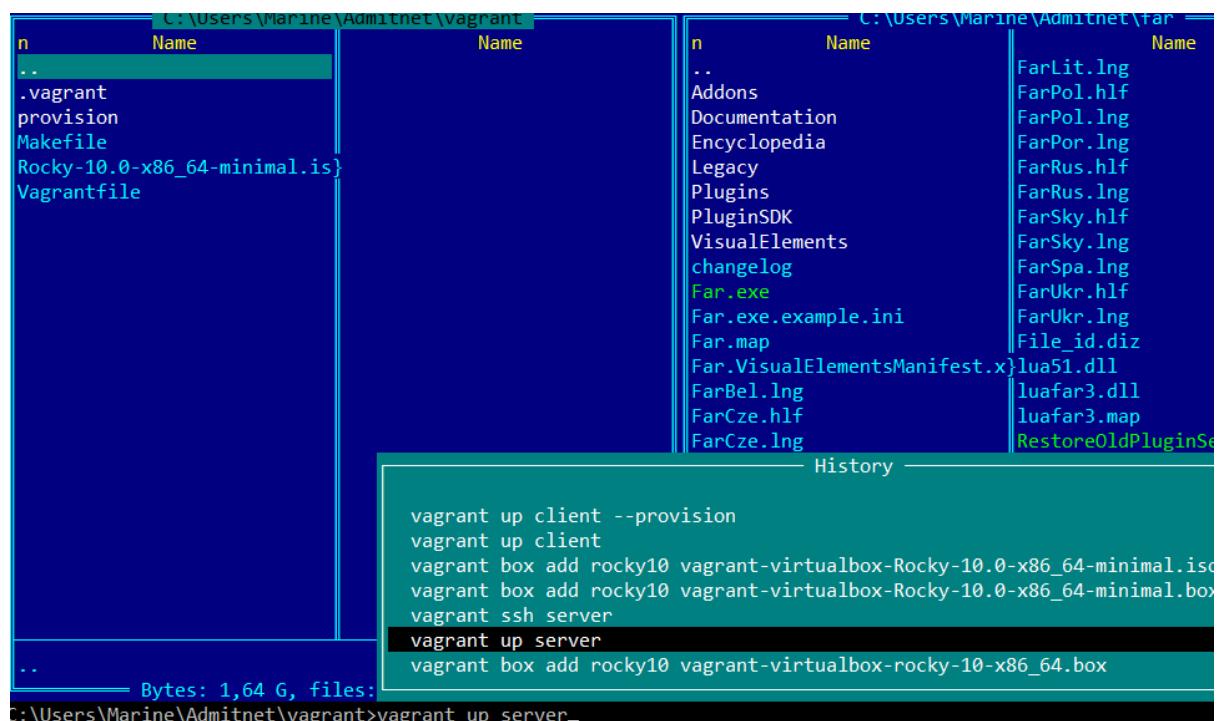
4.1. Цель работы.

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

4.2. Выполнение работы

1. Установите необходимые для работы HTTP-сервера пакеты. Установка HTTP-сервера.

Сначала запускаем виртуальную машину с помощью команды vagrant up server. Так как операционная система Windows, то работаем через far.



The screenshot shows the Far Manager interface with two main panes and a bottom command history window.

- Left Pane:** Shows the directory structure of a Vagrant environment:

```
C:\Users\Marine\Adminitnet\vagrant
n      Name
..
.vagrant
provision
Makefile
Rocky-10.0-x86_64-minimal.iso
Vagrantfile
```
- Right Pane:** Shows the contents of the Far Manager application folder:

```
C:\Users\Marine\Adminitnet\far
n      Name
..
Addons
Documentation
Encyclopedia
Legacy
Plugins
PluginSDK
VisualElements
changelog
Far.exe
Far.exe.example.ini
Far.map
Far.VisualElementsManifest.xml
FarBel.lng
FarCze.hlf
FarCze.lng
FarLit.lng
FarPol.hlf
FarPol.lng
FarPor.lng
FarRus.hlf
FarRus.lng
FarSky.hlf
FarSky.lng
FarSpa.lng
FarUkr.hlf
FarUkr.lng
File_id.diz
luafar3.dll
luafar3.map
RestoreOldPluginSe
```
- Bottom Window:** Command history window showing the execution of Vagrant commands:

```
vagrant up client --provision
vagrant up client
vagrant box add rocky10 vagrant-virtualbox-Rocky-10.0-x86_64-minimal.iso
vagrant box add rocky10 vagrant-virtualbox-Rocky-10.0-x86_64-minimal.box
vagrant ssh server
vagrant up server
vagrant box add rocky10 vagrant-virtualbox-rocky-10-x86_64.box
```

Команда vagrant up server в far

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

```
[astakhovam@localhost.user.net ~]$ sudo -i
[root@localhost.user.net ~]# LANG=C yum grouplist
Extra Packages for Enterprise Linux 9 - x86_64 6.2 kB/s | 35 kB     00:05
Extra Packages for Enterprise Linux 9 - x86_64          691 kB/s | 20 MB    00:29
Rocky Linux 9 - BaseOS                         400 B/s | 4.1 kB   00:10
Rocky Linux 9 - AppStream                      442 B/s | 4.5 kB   00:10
Rocky Linux 9 - Extras                        287 B/s | 2.9 kB   00:10
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

[root@localhost.user.net ~]# dnf -y groupinstall "Basic Web Server"
Last metadata expiration check: 0:00:36 ago on Mon 17 Nov 2025 08:53:35 PM UTC.
Dependencies resolved.
=====


| Package                           | Architecture | Version            | Repository | Size  |
|-----------------------------------|--------------|--------------------|------------|-------|
| <hr/>                             |              |                    |            |       |
| Installing group/module packages: |              |                    |            |       |
| <code>httpd</code>                | x86_64       | 2.4.62-4.el9_6.4   | appstream  | 44 k  |
| <code>httpd-manual</code>         | noarch       | 2.4.62-4.el9_6.4   | appstream  | 2.2 M |
| <code>mod_fcgid</code>            | x86_64       | 2.3.9-28.el9       | appstream  | 74 k  |
| <code>mod_ssl</code>              | x86_64       | 1:2.4.62-4.el9_6.4 | appstream  | 108 k |
| <hr/>                             |              |                    |            |       |
| Installing dependencies:          |              |                    |            |       |
| <code>apr</code>                  | x86_64       | 1.7.0-12.el9_3     | appstream  | 122 k |
| <code>apr-util</code>             | x86_64       | 1.6.1-23.el9       | appstream  | 94 k  |
| <code>apr-util-bdb</code>         | x86_64       | 1.6.1-23.el9       | appstream  | 12 k  |
| <code>httpd-core</code>           | x86_64       | 2.4.62-4.el9_6.4   | appstream  | 1.4 M |
| <code>httpd-filesystem</code>     | noarch       | 2.4.62-4.el9_6.4   | appstream  | 11 k  |
| <code>httpd-tools</code>          | x86_64       | 2.4.62-4.el9_6.4   | appstream  | 78 k  |
| <code>rocky-logos-httpd</code>    | noarch       | 90.16-1.el9        | appstream  | 24 k  |


```

2. Запустите HTTP-сервер с базовой конфигурацией и проанализируйте его работу. Базовое конфигурирование HTTP-сервера.

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

```

Complete!
[root@localhost.user.net ~]# firewall-cmd --list-services
cockpit dhcp dhcpcv6-client dns ssh
[root@localhost.user.net ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit aus
weisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin
-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-
agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpcv6 dhcpcv6
-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-clien
t etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication f
reeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability http htt
p3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconn
ect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-contr
ol-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-sc
heduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt
libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache
minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-d
ashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovi
rt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus
-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redi
s-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip si
ps slp smtp smtp-submission smtsp snmp snmppts snmptrap spideroak-lansync spotify-sync s

-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-clien
t etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication f
reeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability http htt
p3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconn
ect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-contr
ol-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-sc
heduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt
libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache
minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-d
ashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovi
rt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus
-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redi
s-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip si
ps slp smtp smtp-submission smtsp snmp snmppts snmptrap spideroak-lansync spotify-sync s
quid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog syslog
-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsm vnc-server warpi
nator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-u
dp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotie
r
[root@localhost.user.net ~]# firewall-cmd --add-service=http
success
[root@localhost.user.net ~]# firewall-cmd --add-service=http --permanent
success

```

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

The screenshot shows a dual-terminal window in a desktop environment. The left terminal window is titled 'root@localhost:~' and displays the output of the command 'journalctl -x -f'. It logs the deactivation of the 'man-db-cache-update.service' at 20:54:26 on Nov 17, followed by its successful start and the creation of a start job. The right terminal window is also titled 'root@localhost:~' and shows the successful entry of the 'run-r0436a244ce0041d1a18b51da4ce1961d.service' into the 'dead' state at 20:54:26 on Nov 17. The desktop interface includes a top bar with 'Activities', 'Terminal', date/time, and system status icons.

```
[astakhovam@localhost.user.net ~]$ sudo -i
[sudo] password for astakhovam:
[root@localhost.user.net ~]# journalctl -x -f
Nov 17 20:54:26 localhost.user.net systemd[1]: man-db-cache-update.service: Deactivated successfully.
  Subject: Unit succeeded
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  The unit man-db-cache-update.service has successfully entered the 'dead' state.
Nov 17 20:54:26 localhost.user.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 1996.
Nov 17 20:54:26 localhost.user.net systemd[1]: run-r0436a244ce0041d1a18b51da4ce1961d.service: Deactivated successfully.
  Subject: Unit succeeded
  Defined-By: systemd
  Support: https://wiki.rockylinux.org/rocky/support

  The unit run-r0436a244ce0041d1a18b51da4ce1961d.service has successfully entered the 'dead' state.
```

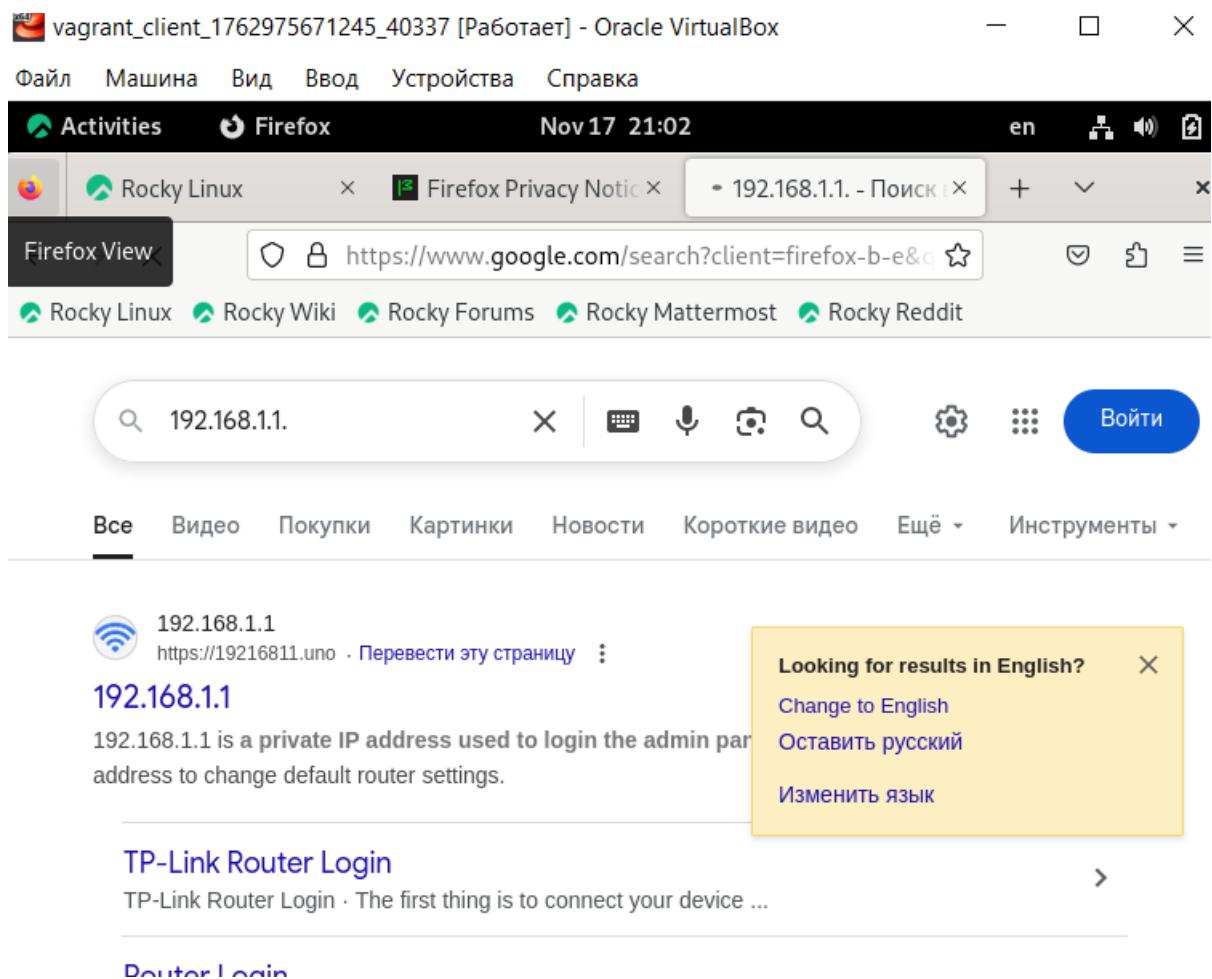
В первом терминале активировали и запустили HTTP-сервер:

The screenshot shows a terminal window with the following command history:

```
[ws-discovery-dap wsman wsmanis xmppc xmpp bosn xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier]
[root@localhost.user.net ~]# firewall-cmd --add-service=http
success
[root@localhost.user.net ~]# firewall-cmd --add-service=http --permanent
success
[root@localhost.user.net ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@localhost.user.net ~]# systemctl start httpd
[root@localhost.user.net ~]#
[root@localhost.user.net ~]#
```

3. Анализ работы HTTP-сервера

Результат запроса в client



4. Настройте виртуальный хостинг. Настройка виртуального хостинга для HTTP-сервера.

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

```
[astakhovamd@localhost.user.net ~]$ sudo -i
[sudo] password for astakhovamd:
[root@localhost.user.net ~]# systemctl stop named
[root@localhost.user.net ~]# cd /var/named/master/fz
[root@localhost.user.net fz]#
```

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

```
root@localhost:/var/named/master/fz
root@localhost:~          x      root@localhost:~          x
GNU nano 5.6.1
$TTL 1D
@ IN SOA      @ server.astakhovamd.net. (
    2024072700      ; serial
    1D              ; refresh
    1H              ; retry
    1W              ; expire
    3H )            ; minimum
    NS      @
    A      192.168.1.1
$ORIGIN astakhovamd.net.
server      A      192.168.1.1
ns          A      192.168.1.1
dhc         A      192.168.1.1
www         A      192.168.1.1
```

И в файле 192.168.1

```
root@localhost:/var/named/master/fz
root@localhost:~          x      root@localhost:~          x
GNU nano 5.6.1
192.168.1
$TTL 1D
@ IN SOA      @ server.astakhovamd.net. (
    2024072700      ; serial
    1D              ; refresh
    1H              ; retry
    1W              ; expire
    3H )            ; minimum
    NS      @
    A      192.168.1.1
    PTR     server.astakhovamd.net.
$ORIGIN 1.168.192.in-addr.arpa.
1      PTR     server.astakhovamd.net.
1      PTR     ns.astakhovamd.net.
1      PTR     www.astakhovamd.net.
```

В каталоге /etc/httpd/conf.d создали файлы server.astakhovamd.net.conf и www.astakhovamd.net.conf:

```
[root@localhost.user.net fz]# cd /etc/httpd/conf.d
[root@localhost.user.net conf.d]# sudo nano server.astakhovamd.net.conf
[root@localhost.user.net conf.d]# touch server.astakhovamd.net.conf
[root@localhost.user.net conf.d]# touch www.astakhovamd.net.conf
[root@localhost.user.net conf.d]#
```

Открыли на редактирование файл server.astakhovamd.net.conf и внесли следующее содержание:

```
root@localhost:/etc/httpd/conf.d
root@localhost:~          x      root@localhost:~          x
GNU nano 5.6.1
<VirtualHost *:80>
ServerAdmin webmaster@astakhovamd.net
DocumentRoot /var/www/html/server.astakhovamd.net
ServerName server.astakhovamd.net
ErrorLog logs/server.astakhovamd.net-error_log
CustomLog logs/server.astakhovamd.net-access_log common
</VirtualHost>
```

Открыли на редактирование файл www.astakhovamd.net.conf и внесли следующее содержание:

```
root@localhost:/etc/httpd/conf.d
root@localhost:~          x      root@localhost:/etc/httpd/conf.d
GNU nano 5.6.1
<VirtualHost *:80>
ServerAdmin webmaster@astakhovamd.net
DocumentRoot /var/www/html/www.astakhovamd.net
ServerName www.astakhovamd.net
ServerAlias www.astakhovamd.net
ErrorLog logs/www.astakhovamd.net-error_log
CustomLog logs/www.astakhovamd.net-access_log common
RewriteEngine on
RewriteRule ^(.*)$ https://%(HTTP_HOST)$1 [R=301,L]
</VirtualHost>

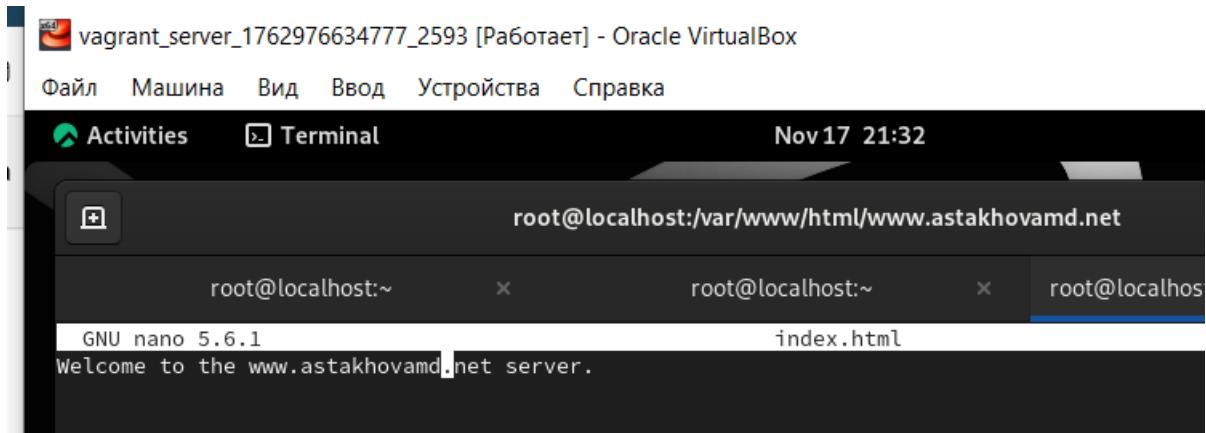
<IfModule mod_ssl.c>
<VirtualHost *:443>
SSLEngine on
ServerAdmin webmaster@astakhovamd.net
DocumentRoot /var/www/html/www.astakhovamd.net
ServerName www.astakhovamd.net
ServerAlias www.astakhovamd.net
ErrorLog logs/www.astakhovamd.net-error_log
CustomLog logs/www.astakhovamd.net-access_log common
```

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.astakhovamd.net и www.astakhovamd.net.

Для виртуального веб-сервера server.astakhovamd.net:

```
[root@localhost.user.net html]# mkdir server.astakhovamd.net
[root@localhost.user.net html]# cd /var/www/html/server.astakhovamd.net
[root@localhost.user.net server.astakhovamd.net]# touch index.html
[root@localhost.user.net server.astakhovamd.net]# sudo nano index.html
[root@localhost.user.net server.astakhovamd.net]# cd /var/www/html
[root@localhost.user.net html]# mkdir www.astakhovamd.net
[root@localhost.user.net html]# cd /var/www/html/www.astakhovamd.net
[root@localhost.user.net www.astakhovamd.net]# touch index.html
[root@localhost.user.net www.astakhovamd.net]#
```

Открыли на редактирование файл index.html и внесли следующее содержание: Welcome to the server.astakhovamd.net.server.



Скорректировали права доступа в каталог с веб-контентом и восстановили контекст безопасности, и перезапустили сервер:

```
[root@localhost.user.net www.astakhovam.net]# sudo nano index.html
[root@localhost.user.net www.astakhovam.net]# chown -R apache:apache /var/www
[root@localhost.user.net www.astakhovam.net]# restorecon -VR /etc
[root@localhost.user.net www.astakhovam.net]# restorecon -VR /var/named
[root@localhost.user.net www.astakhovam.net]# restorecon -VR /var/www
[root@localhost.user.net www.astakhovam.net]# systemctl restart httpd
```

5. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке HTTP-сервера во внутреннем окружении виртуальной машины server.

Соответствующим образом внесите изменения в Vagrantfile. Внесение изменений в настройки внутреннего окружения виртуальной машины.

На виртуальной машине server перешли в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создали в нём каталог http, в который поместили в соответствующие подкаталоги конфигурационные файлы HTTP-сервера:

```
[root@localhost.user.net www.astakhovam.net]# cd /vagrant/provision/server
[root@localhost.user.net server]# mkdir -p /vagrant/provision/server/http/etc/httpd/conf.d
[root@localhost.user.net server]# mkdir -p /vagrant/provision/server/http/var/www/html
[root@localhost.user.net server]# cp -R /etc/httpd/conf.d/*
↳ /vagrant/provision/server/http/etc/httpd/conf.d/
cp: target '/etc/httpd/conf.d/www.astakhovam.net.conf' is not a directory
bash: ↳: command not found...
[root@localhost.user.net server]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d/
[root@localhost.user.net server]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
[root@localhost.user.net server]#
```

Заменили конфигурационные файлы DNS-сервера:

```
[root@localhost.user.net server]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d
[root@localhost.user.net server]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
[root@localhost.user.net server]# cd /vagrant/provision/server/dns/
[root@localhost.user.net dns]# cp -R /var/named/* /vagrant/provision/server/dns/var/named/
cp: overwrite '/vagrant/provision/server/dns/var/named/data/named.run'? cp -R /var/named/* /vagrant/provision/server/
[astakhovamd@localhost.user.net ~]$ sudo -i
[sudo] password for astakhovamd:
[root@localhost.user ~]# 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/astakhovamd.net'? Y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/192.168.1'? Y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/fz/user.net'? Y
cp: overwrite '/vagrant/provision/server/dns/var/named/master/rz/192.168.1'? Y
cp: overwrite '/vagrant/provision/server/dns/var/named/named.astakhovamd.net.zone'? 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
cp: overwrite '/vagrant/provision/server/dns/var/named/named.user.net.zone'? Y
[root@localhost.user.net ~]# █

[root@localhost.user.net www.astakhovamd.net]# cd /vagrant/provision/server
[root@localhost.user.net server]# mkdir -p /vagrant/provision/server/http/etc/httpd/conf.d
[root@localhost.user.net server]# mkdir -p /vagrant/provision/server/http/var/www/html
[root@localhost.user.net server]# cp -R /etc/httpd/conf.d/*
↳ /vagrant/provision/server/http/etc/httpd/conf.d/
cp: target '/etc/httpd/conf.d/www.astakhovamd.net.conf' is not a directory
bash: ↳: command not found...
[root@localhost.user.net server]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d/
[root@localhost.user.net server]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
[root@localhost.user.net server]#
```

В каталоге `/vagrant/provision/server` создайте исполняемый файл `http.sh`:

```
[root@localhost ~]# cd /vagrant/provision/server
[root@localhost server]# touch http.sh
[root@localhost server]# chmod +x http.sh
[root@localhost server]#
```

```
root@localhost:/vagrant/provision/server
root@localhost:~          x      root@localhost:~          x      ro

GNU nano 5.6.1           http.sh

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
```

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

```
end

server.vm.provision "server dhcp",
  type: "shell",
  preserve_order: true,
  path: "provision/server/dhcp.sh"
end

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

4.3. Итог работы

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

4.4. Контрольные вопросы

Через какой порт по умолчанию работает Apache?

- A. ***Порт 80 (HTTP):*** Это стандартный порт для незашифрованного веб-трафика (Hypertext Transfer Protocol). Apache слушает этот порт по умолчанию для обработки обычных HTTP-запросов.
- B. ***Порт 443 (HTTPS):*** Если Apache настроен для работы с SSL/TLS (то есть для защищенного соединения HTTPS - Hypertext Transfer Protocol Secure), то он будет слушать порт 443.

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

Apache разработан для работы с минимальными привилегиями из соображений безопасности. Он запускается как root только для того, чтобы привязаться к портам ниже 1024 (например, 80 и 443), а затем переключается на непrivилегированного пользователя для обработки запросов.

Имя пользователя и группы зависят от дистрибутива Linux:

1. На Debian/Ubuntu:
2. Пользователь: www-data
3. Группа: www-data
4. На RHEL/CentOS/Fedora:
5. Пользователь: apache
6. Группа: apache

Зачем это нужно?

Если злоумышленник скомпрометирует веб-сервер, работающий под непrivилегированным пользователем, его возможности будут сильно ограничены. Он не сможет получить полный контроль над системой или получить доступ к файлам, принадлежащим root или другим пользователям.

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

Лог-файлы Apache очень важны для мониторинга, отладки и анализа работы веб-сервера.

Расположение по умолчанию:

1. На Debian/Ubuntu: /var/log/apache2/
2. На RHEL/CentOS/Fedora: /var/log/httpd/

Примечание: для каждого VirtualHost могут быть настроены свои собственные файлы логов.

Основные типы лог-файлов и что по ним можно отслеживать:

1. access.log (или access_log):

1. Что отслеживает:

1. Все входящие запросы к веб-серверу.

2. Содержит информацию о:

1. IP-адрес клиента, сделавшего запрос.
2. Дата и время запроса.
3. Метод запроса (GET, POST и т.д.) и запрашиваемый URL.
4. Код состояния HTTP (200 OK, 404 Not Found, 500 Internal Server Error и т.д.).
5. Размер ответа в байтах.
6. Referer (URL, с которого пришел пользователь).
7. User-Agent (информация о браузере/ОС клиента).
8. Время обработки запроса.
 - 2.8.1. Для чего используется:
 - 2.8.1.1. Анализ трафика
 - 2.8.1.2. Поведенческий анализ пользователей
 - 2.8.1.3. Выявление проблем с доступностью контента
 - 2.8.1.4. Мониторинг производительности.

3. *error.log (или error_log):*

1. **Что отслеживает:** Ошибки, предупреждения и диагностические сообщения, возникающие на стороне сервера.
2. **Содержит информацию о:**
 - 3.2.1. Внутренние ошибки сервера (например, 500 Internal Server Error).
 - 3.2.2. Проблемы с конфигурацией Apache.
 - 3.2.3. Ошибки при обработке скриптов (например, PHP-ошибки, если используется mod_php).
 - 3.2.4. Проблемы с правами доступа к файлам.
 - 3.2.5. Несуществующие файлы, к которым был запрос.
3. **Для чего используется:**
 - 3.3.1. Отладка проблем сервера
 - 3.3.2. Выявление уязвимостей
 - 3.3.3. Мониторинг стабильности работы.
4. **Другие логи:** Могут быть специфические логи для SSL, перезаписи URL (mod_rewrite) и т.д, если они настроены.

