

## **Тема и цель работы**

Тема лабораторной работы: «Настройка веб-сервера на примере Apache».

Цель работы: Научиться устанавливать, проводить базовые настройки и проверять работоспособность apache.

## Оборудование, ПО

Устройство	Операционная система	IP адрес/Маска	Шлюз	DNS
CLI_A1	Astra Linux SE 1.8.x	10.0.0.1/24	-	au-1.au.team.lab
CLI_A2	Astra Linux SE 1.8.x	10.0.0.2/24	-	au-2.au.team.lab
CLI_A3	Astra Linux SE 1.8.x	10.0.0.3/24	-	au-3.au.team.lab

## Выполнение

1. Перед началом работы необходимо прописать следующие команды:

**apt-get update apt-get**

**upgrade apt-get dist-**

**upgrade -f**

2. Затем установить веб-сервер apache2

```
root@astra:/media/cdrom# apt install apache2
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0
Предлагаемые пакеты:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Следующие НОВЫЕ пакеты будут установлены:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0
Обновлено 0 пакетов, установлено 9 новых пакетов, для удаления отмечено 0 пакетов, и 556 пакетов не обновлен
Необходимо скачать 2 315 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 8 300 kB.
Хотите продолжить? [Д/н] Д
Пол:1 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 libapr1 a
64 1.7.2-3+b1 [102 kB]
Пол:2 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 libapruti
amd64 1.6.3-1+b2 [88,9 kB]
Пол:3 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 libapruti
-dbd-sqlite3 amd64 1.6.3-1+b2 [14,1 kB]
Пол:4 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 libapruti
-ldap amd64 1.6.3-1+b2 [12,3 kB]
Пол:5 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 liblua5.3
```

Рис.1 – Установка apache2

3. Выключим режим AstraMode в файле /etc/apache2/apache2.conf.

```
#ServerRoot "/etc/apache2"

#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#Mutex file:${APACHE_LOCK_DIR} default

#
# The directory where shm and other runtime files will be stored.
#

DefaultRuntimeDir ${APACHE_RUN_DIR}

# Astra security mode.
#
[AstraMode off

#
# Including realm to user name for astra mode.
#
# IncludeRealm off

#
# Controls which parsec capabilities are allowed for child processes
-- ВСТАВКА --
```

Рис.2 – Выключение AstraMode

4. Создать файл /etc/apache2/conf-available/fqdn.conf с содержимым:

**ServerName localhost**

```
ServerName localhost
```

Рис.3 – Создание файла /etc/apache2/conf-available/fqdn.conf с нужным содержимым

5. Активировать конфигурационный файл fqdn, перечитать конфигурацию вебсервера и проверить его состояние, но перед этим запустим apache2:

```
sudo start apache2 sudo
```

```
a2enconf fqdn sudo systemctl
```

```
reload apache2 sudo
```

```
systemctl status apache2
```

```
root@astra:/media/cdrom# a2enconf fqdn
Conf fqdn already enabled
root@astra:/media/cdrom# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-11-24 12:44:00 MSK; 13min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 5758 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
    Main PID: 4484 (apache2)
      Tasks: 6 (limit: 5067)
     Memory: 17.3M
        CPU: 298ms
    CGroup: /system.slice/apache2.service
            └─4484 /usr/sbin/apache2 -k start
              └─5762 /usr/sbin/apache2 -k start
                └─5763 /usr/sbin/apache2 -k start
                  └─5764 /usr/sbin/apache2 -k start
                    └─5765 /usr/sbin/apache2 -k start
                      └─5766 /usr/sbin/apache2 -k start

ноя 24 12:44:00 astra systemd[1]: Starting apache2.service - The Apache HTTP Server...
ноя 24 12:44:00 astra apachectl[4483]: AH00558: apache2: Could not reliably determine the server's fully qual
ноя 24 12:44:00 astra systemd[1]: Started apache2.service - The Apache HTTP Server.
ноя 24 12:56:37 astra systemd[1]: Reloading apache2.service - The Apache HTTP Server...
ноя 24 12:56:37 astra systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
lines 1-22/22 (END)
```

Рис.4 – Запуск apache2, активация fqdn, перечитывание конфигурации сервера и проверка его состояния

6. Отредактировать файл конфигурации виртуального хоста /etc/apache2/sitesavailable/000-default.conf, указав доменное имя сервера в параметре ServerName и путь к каталогу с веб-страницами

```
ServerName www.au-team.lab

ServerAdmin webmaster@localhost
DocumentRoot /var/www/astra
```


Рис.5 – Указание доменного имени сервера и пути к каталогу с веб-страницами

7. Создать каталог для размещения веб-страниц:

**mkdir /var/www/astra**

8. Разместить в созданном каталоге веб-страницу index.html, например, со следующим содержимым:

```
<html>
<head>
<title>Welcome to astra</title>
</head>
<body>
<h1>Welcome to astra</h1>
<p>Hello!</p>
</body>
</html>
```

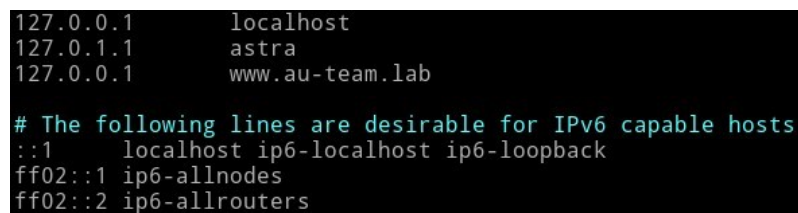


```
root@astra:/var/www/astra# cat /var/www/astra/index.html
<html>
<head>
<title>Welcome to astra</title>
</head/>
<body>
<h1>Welcome to astra</h1>
<p>Hello!</p>
</body>
</html>
```

Рис.6 – Создание веб-страницы index.html

9. Дописать в файле /etc/hosts следующую строку:

**127.0.0.1 www.au-team.lab**



```
127.0.0.1    localhost
127.0.1.1    astra
127.0.0.1    www.au-team.lab

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Рис.7 – Редактирование файла /etc/hosts

10. Добавить в /etc/apache2/sites-available файл astra.conf со следующим содержанием:

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/astra
```

**ServerName astra.local**

**ErrorLog \${APACHE\_LOG\_DIR}/error.log**

**CustomLog \${APACHE\_LOG\_DIR}/access.log combined </VirtualHost>**

```
<VirtualHost *:80>
    ServerName www.au-team.lab
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/astra/info.php
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Рис.8 – Добавление файла astra.conf с нужным содержимым

11. Перезагрузить apache при помощи команды **systemctl reload apache2**
12. Активировать сайт, описанный в конфигурации виртуального хоста **a2ensite astra**
13. Проверим созданную страницу в браузере.

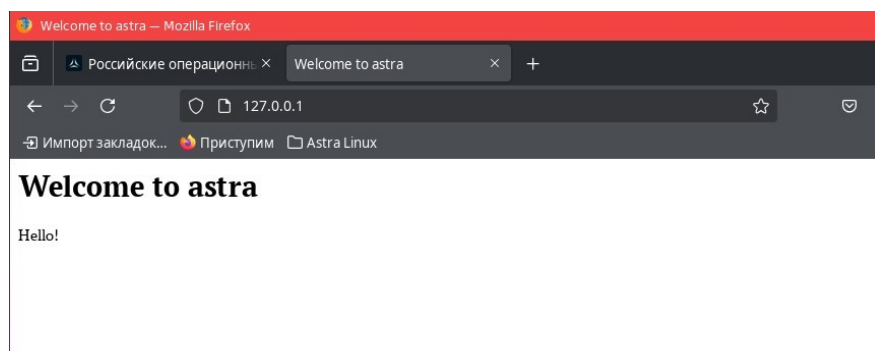


Рис.9 – Созданная страница в браузере

Сайт работает.

14. Установить php8.1 в качестве модуля Apache2, прописав команду:  
**sudo apt install php**

```

Ошибка: Site astra does not exist.
root@astra:/media/cdrom# apt install php
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libapache2-mod-php8.2 php-common php8.2 php8.2-cli php8.2-common php8.2-opcache php8.2-readline
Предлагаемые пакеты:
  php-pear
Следующие НОВЫЕ пакеты будут установлены:
  libapache2-mod-php8.2 php php-common php8.2 php8.2-cli php8.2-common php8.2-opcache php8.2-readline
Обновлено 0 пакетов, установлено 8 новых пакетов, для удаления отмечено 0 пакетов, и 556 пакетов не обновлено.
Необходимо скачать 4 506 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 21,2 MB.
Хотите продолжить? [Д/н] д
Пол:1 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 php-common
all 2:93+b1 [13,4 kB]
Пол:2 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 php8.2-comm
on amd64 8.2.18-1~deb12u1+b1 [680 kB]
Пол:3 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 php8.2-opca
che amd64 8.2.18-1~deb12u1+b1 [345 kB]
Пол:4 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 php8.2-read
line amd64 8.2.18-1~deb12u1+b1 [12,7 kB]
Пол:5 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 php8.2-cli
amd64 8.2.18-1~deb12u1+b1 [1 735 kB]
Пол:6 https://download.astralinux.ru/astra/stable/1.8_x86-64/repository-main 1.8_x86-64/main amd64 libapache2-

```

Рис.10 – Установка php

## 15. Проверить версию php: **php -v**

```

root@astra:/var/www/astra# php -v
PHP 8.2.18 (cli) (built: Jul 19 2024 05:14:00) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.18, Copyright (c) Zend Technologies
with Zend OPcache v8.2.18, Copyright (c), by Zend Technologies
root@astra:/var/www/astra# █

```

Рис.11 – Проверка версии php

## 16. Создать в каталоге виртуального хоста файл info.php с содержимым: **<?php phpinfo(); ?>**

```

root@astra:/var/www/astra# cat /var/www/astra/info.php
<?php
phpinfo();
?>

root@astra:/var/www/astra# █

```

Рис.12 – Создание файла info.php с таким содержимым

## 17. Отредактировать файл конфигурации виртуального хоста /etc/apache2/sitesavailable/000-default.conf, указав в пути к каталогу с веб-страницами файл info.php.

```

<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName www.au-team.lab

ServerAdmin webmaster@localhost
DocumentRoot /var/www/astra/info.php

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

Рис.13 – Редактирование конфигурационного файла виртуального хоста

## 18. Проверить созданную страницу в браузере

PHP Version 8.2.18	
System	Linux astra1 6.1.90-1-generic #astra2+015 SMP PREEMPT_DYNAMIC Tue Jul 23 09:49:19 MSK 2024 x86_64
Build Date	Jul 19 2024 05:14:00
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.2/apache2
Loaded Configuration File	/etc/php/8.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.2/apache2/conf.d
Additional .ini files parsed	/etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-curl.ini, /etc/php/8.2/apache2/conf.d/20-dom.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-filter.ini, /etc/php/8.2/apache2/conf.d/20-gd.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-ldap.ini, /etc/php/8.2/apache2/conf.d/20-ldap_sasl.ini, /etc/php/8.2/apache2/conf.d/20-mbstring.ini, /etc/php/8.2/apache2/conf.d/20-mcrypt.ini, /etc/php/8.2/apache2/conf.d/20-mysqlnd.ini, /etc/php/8.2/apache2/conf.d/20-openssl.ini, /etc/php/8.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.2/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/8.2/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/8.2/apache2/conf.d/20-pgsql.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	40202029
Zend Extension Build	API40202029.NTS
PHP Extension Build	API20220829.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tls_v1, tls_v1.2, tls_v1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

Рис.14 – Проверка созданной страницы в браузере

## 19. Модуль php работает, но сайт является незащищённым. Чтобы сайт считался безопасным, надо воспользоваться сертификатом.

Для тестовой среды можно сгенерировать самоподписанный сертификат. Чтобы сделать это, необходимо перейти в рабочую папку и сгенерировать сертификат:

```
cd /var/www/astra
```

```
openssl req -new -x509 -days 1461 -nodes -out cert.pem
```



[illegible]

20. Прежде чем установить модуль SSL для Apache, надо прописать команду, проверяющую данный модуль на наличие: **apachectl -M | grep ssl**

```
root@astra-1:/var/www/astra# apachectl -M | grep ssl
AH00112: Warning: DocumentRoot [/var/www/astra/info.php] does not exist
AH00112: Warning: DocumentRoot [/var/www/astra/info.php] does not exist
AH00112: Warning: DocumentRoot [/var/www/apache/data] does not exist
ssl_module (shared)
root@astra-1:/var/www/astra#
```

Видим строку `ssl module (shared)`, означающую, что модуль `ssl` установлен

```
cd ... vim /etc/apache2/sites-enabled/astra.conf
```

<VirtualHost \*:443>

**SSLCertificateKeyFile** /var/www/astra/cert.key

```
<VirtualHost *:443>
    ServerName astra
    DocumentRoot /var/www/apache/data
    SSLEngine on
    SSLCertificateFile /var/www/astra/cert.pem
    SSLCertificateKeyFile /var/www/astra/cert.key
</VirtualHost>
```

9

23. Проверить настройки apache на корректность: **apachectl configtest**

```
root@astra:/# apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/astra/info.php] does not exist
AH00112: Warning: DocumentRoot [/var/www/apache/data] does not exist
Syntax OK
root@astra:/#
```

Рис.18 – Проверка настройки apache на корректность

24. Видим надпись «Syntax OK», значит, можно перечитать конфигурацию apache:  
**apachectl graceful**

```
Syntax OK
root@astra:/# apachectl graceful
AH00112: Warning: DocumentRoot [/var/www/astra/info.php] does not exist
AH00112: Warning: DocumentRoot [/var/www/apache/data] does not exist
root@astra:/#
```

Рис.19 – Перечитывание конфигурации apache

25. Чтобы все запросы по http автоматически перенаправлялись на https, необходимо настроить перенаправление (redirect).

Для этого в конфигурационном файле sites-enabled/astra.conf надо дописать следующие строки:

**<VirtualHost \*:80>**

**ServerName astra**

**RewriteEngine On**

**RewriteCond %{HTTPS} off**

**RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI} [R=301,L]**

**</VirtualHost>**

```
<VirtualHost *:80>
    ServerName astra
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
```

Рис. 20 – Настройка перенаправления с http на https

26. Чтобы перенаправление работало в Apache, необходимо установить модуль rewrite:

**a2enmod rewrite systemctl**

**restart apache2**

```
root@astra:/etc/apache2# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@astra:/etc/apache2# systemctl restart apache2
root@astra:/etc/apache2# █
```

Рис.21 – Установка модуля rewrite и перезапуск apache

27. Далее зайдём на сайт и проверим сертификат на наличие.

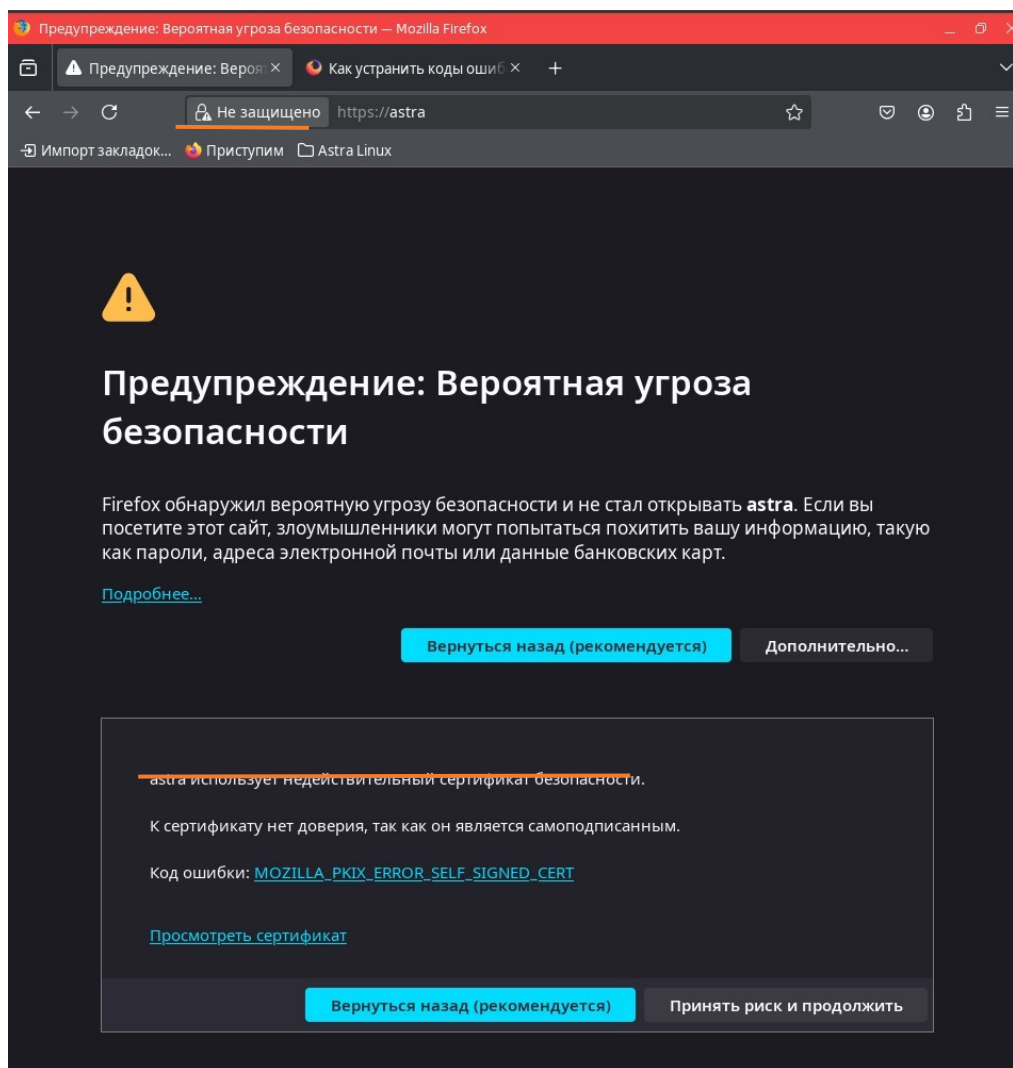


Рис. 22 – Проверка сертификата на наличие

Сертификат привязан к сайту.

## Вывод

В ходе лабораторной работы мы научились устанавливать, проводить базовые настройки и проверять работоспособность арасче.

## **Контрольные вопросы**

1) Для чего используется apache?

Apache, или Apache HTTP Server, — это один из самых популярных веб-серверов в мире. Он используется для следующих целей:

1. Хостинг веб-сайтов: Apache позволяет размещать статические и динамические веб-страницы, обеспечивая доступ к ним через интернет.
2. Обработка запросов: Сервер принимает запросы от клиентов (обычно веббраузеров) и отвечает на них, отправляя запрашиваемые ресурсы.
3. Поддержка различных языков программирования: Apache может работать с различными языками, такими как PHP, Python, Ruby и другими, позволяя создавать динамические веб-приложения.

4. Настройка виртуальных хостов: С помощью Apache можно настроить несколько веб-сайтов на одном сервере, используя виртуальные хосты.
5. Безопасность: Apache предоставляет множество функций для обеспечения безопасности, таких как аутентификация пользователей, управление доступом и поддержка HTTPS.
6. Модули расширения: Apache поддерживает модули, которые позволяют добавлять дополнительные функции, такие как URL-переписывание, кэширование и сжатие контента.
7. Статистика и логирование: Apache может вести журналы посещений и ошибок, что помогает администраторам отслеживать использование сайта и диагностировать проблемы.

## 2) Зачем нужен php модуль?

PHP модуль для Apache (или других веб-серверов) необходим для выполнения скриптов на языке PHP.

## 3) Для чего используются сертификаты?

Сертификаты используются для различных целей, связанных с обеспечением безопасности и доверия в цифровом мире. Вот основные из них:

1. Шифрование данных.
2. Аутентификация.
3. Целостность данных.
4. Электронная подпись.
5. Управление доступом.
6. Системы управления идентификацией.
7. Поддержка стандартов и протоколов.

