

Apache

Apache - популярный бесплатный opensource веб-сервер. Он является частью стека **LAMP** (Linux, Apache, MySQL, PHP), который обеспечивает большую часть Интернета.

Установка

CentOS: `dnf update && dnf -y install httpd`

Debian: `apt update && apt -y install apache2`

Файлы

Файлы конфигураций

- `/etc/httpd/` - Расположение всех файлов конфигурации
- `/etc/httpd/conf/httpd.conf` - основной файл конфигурации Apache
- `/etc/httpd/conf.d/` - Все конфигурационные файлы в этом каталоге включены в основной файл настроек
- `/etc/httpd/conf.modules.d/` - Расположение конфигурационных файлов модуля Apache

Примечание. При внесении изменений в файлы конфигурации не забывайте всегда перезапускать службу Apache, чтобы применить новую конфигурацию.

Логи

- `/var/log/httpd/` - расположение файлов логов Apache
- `/var/log/httpd/access_log` - показывает журнал систем, которые обращались к серверу
- `var/log/httpd/error_log` - показывает список любых ошибок, с которыми сталкивается Apache

Типичные места для хранения сайтов

- `/home/username/my_website`
- `/var/www/my_website`
- `/var/www/html/my_website`
- `/opt/my_website`

Документация

- `/usr/share/doc/httpd`

Управление

```
1 systemctl [start/stop] [status] [enable/disable] [restart/reload]
   httpd/apache2
```

- `reload` - перечитывание конфигурации
- `restart` - перезапуск сервиса

apachectl

[apachectl](#) - интерфейс управления HTTP-сервером Apache

- `start`

Запускает `httpd` демон **Apache**. Выдает ошибку, если он уже запущен. Это эквивалентно `apachectl -k start`.

- `stop`

Останавливает `httpd` демон **Apache**. Это эквивалентно `apachectl -k stop`.

- `restart`

Перезапускает `httpd` демон **Apache**. Если демон не запущен, он запускается. Эта команда автоматически проверяет файлы конфигурации `configtest` перед началом перезапуска, чтобы убедиться, что демон не умер. Это эквивалентно `apachectl -k restart`.

- `fullstatus`

Отображает полный отчет о состоянии из `mod_status`. Чтобы это работало, вам необходимо иметь включенным на своем сервере `mod_status` и текстовый браузер, например, `Tlynx` доступный в вашей системе. URL-адрес, используемый для доступа к отчету о состоянии, можно установить, отредактировав `STATUSURL` переменную в скрипте.

- `status`

Отображает краткий отчет о состоянии. Аналогично этой `fullstatus` опции, за исключением того, что список обслуживаемых в данный момент запросов опущен.

- `graceful`

Изящно перезапускает `httpd` демон **Apache**. Если демон не запущен, он запускается. Это отличается от обычного перезапуска тем, что открытые в данный момент соединения не прерываются. Побочным эффектом является то, что старые файлы журналов не закрываются немедленно. Это означает, что при использовании в сценарии ротации журналов может потребоваться значительная задержка, чтобы гарантировать закрытие старых файлов журналов перед их обработкой. Эта команда автоматически проверяет файлы конфигурации `configtest` перед запуском перезапуска, чтобы убедиться, что **Apache** не умирает. Это эквивалентно `apachectl -k graceful`.

- `graceful-stop`

Изящно останавливает `httpd` демон **Apache**. Это отличается от обычной остановки тем, что открытые в данный момент соединения не прерываются. Побочным эффектом является то, что старые файлы журналов не закрываются немедленно. Это эквивалентно `apachectl -k graceful-stop`.

- `configtest`

Запустите проверку синтаксиса файла конфигурации. Он анализирует файлы конфигурации и сообщает `Syntax OK` либо подробную информацию о конкретной синтаксической ошибке. Это эквивалентно `apachectl -t`.

Конфигурация

Сервер настраивается путём размещения [директив конфигурации](#) в этих файлах конфигурации.

Директива — это ключевое слово с одним или несколькими аргументами, устанавливающими её значение.

директива {аргумент} {аргумент\ через\ пробелы}

На вопрос: «Где я должен прописать эту директиву?» – обычно отвечают, там где ты хочешь использовать её. Если это глобальная настройка, она должна располагаться в конфигурационном файле вне разделов `<Directory>`, `<Location>`, `<VirtualHost>` или других разделов. Если настройка относится только к конкретному каталогу, значит она должна быть внутри секции `<Directory>`, которая описывает этот каталог, и так далее. Смотри документ [Разделы конфигурации](#) с подробным описанием вышеуказанных разделов.

httpd.conf

Совет: Вы можете проверить корректность настроек Apache без запуска сервера командой `apachectl configtest`.

Директивы:

- `User http`

По соображениям безопасности при запуске сервера Apache от имени суперпользователя (напрямую или через скрипт инициализации) происходит смена идентификатора пользователя (UID), от имени которого выполняется процесс сервера. По умолчанию используется пользователь `http`, который не имеет привилегированных полномочий в системе.

- `Listen 80` | [Документация](#)

Это порт, через который Apache принимает входящие соединения. Если сервер имеет выход в интернет через маршрутизатор, необходимо будет настроить перенаправление этого порта.

Если вы используете Apache для разработки и тестирования, вы можете разрешить только локальный доступ к нему. Для этого укажите `Listen 127.0.0.1:80`.

- `ServerAdmin you@example.com`

Адрес электронной почты администратора, который будет выводиться, например, на странице ошибки **Apache**.

- `DocumentRoot "/srv/http"`

Это корневая директория **Apache**, в которой можно разместить ваши веб-страницы. Измените ее, если нужно, но не забудьте также поменять путь в директиве `<Directory "/srv/http">` на новое расположение `DocumentRoot`, иначе вы, скорее всего, получите сообщение об ошибке 403 Error (недостаточно полномочий) при попытке получить доступ к новому корневому каталогу Apache. Также не забудьте изменить строку `Require all denied` на `Require all granted`, иначе снова получите ошибку 403 Error. Помните, что директория `DocumentRoot` и ее родительские папки должны иметь разрешения на запуск для всех (можно установить командой `chmod o+x /path/to/DocumentRoot`), в противном случае вы получите ошибку **403 Error**.

Обычно, когда запрашивается каталог, без указания имени файла, то будет отдан документ с именем `index.html`. Например, если для директивы `DocumentRoot` установлено значение `/var/www/html` и приходит запрос на адрес `http://www.example.com/work/`, то файл расположенный по пути `/var/www/html/work/index.html` будет отдан клиенту.

- `AllowOverride None`

Запрещает переопределение настроек. Если в секции `<Directory>` указана эта директива, Apache будет полностью игнорировать настройки в файле `.htaccess`. Обратите внимание, что теперь такая настройка для Apache 2.4 является настройкой по умолчанию, поэтому если вы планируете использовать `.htaccess`, вам необходимо дать соответствующие разрешения.

Если вы собираетесь включить модуль `mod_rewrite` или использовать настройки в `.htaccess`, вы можете определить какие из директив, объявленных в этих файлах, могут перезаписывать конфигурацию сервера. Для получения дополнительной информации обратитесь к [документации Apache](#).

Виртуальные хосты (VirtualHost)

Термин виртуальный хост относится к практике запуска более одного веб-сайта (например, `company1.example.com` и `company2.example.com`) на одной машине. Виртуальные хосты могут быть "[IP-based](#)", что означает, что у вас есть разные IP-адреса для каждого веб-сайта, или "[name-based](#)", что означает, что у вас есть несколько имен, работающих на каждом IP-адресе. Тот факт, что они работают на одном физическом сервере, не очевиден для конечного пользователя.

[Примеры конфигураций](#)

Apache был одним из первых серверов, который прямо из коробки поддерживал виртуальные хосты на основе IP. Версии 1.1 и более поздние версии Apache поддерживают виртуальные хосты на основе IP и имен (vhosts). Последний вариант виртуальных хостов иногда также называют виртуальными хостами на *основе хоста* или *без IP*.

Reverse-proxy

http://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

Перенаправление HTTP на HTTPS

<https://www.dmosk.ru/miniinstruktions.php?mini=apache-ssl>

Инструкция

1. Установить **Apache**

```
1 dnf install httpd
2 apt install apache2
```

2. Запустить и добавить его в автозагрузку

```
1 systemctl start httpd
2 systemctl enable httpd
```

3. Добавить исключение в фаервол (**http** - 80 port: **https** - 443 port)

4. Создать директорию для нового сайта с каталогами **html** (файлы сайта) и **log** (для хранения журналов)

```
1 mkdir -p /var/www/www.skill39.wsr/html
2 mkdir /var/www/www.skill39.wsr/log
```

5. Назначить права доступа для пользователя указанного либо глобально в файле *httpd.conf* (**apache/apache**) или для кастомного пользователя, объявленного в конфигурации определенного *виртуального хоста*

```
1 chown -R [user]:[group] /var/www/www.skill39.wsr/html
```

6. Убедитесь, что ваша корневая директория имеет набор разрешений по умолчанию:

```
1 chmod -R 755 /var/www
```

7. Создать файл `index.html` (данная страница будет показывать по умолчанию при входе на сайт)

```
1 nano /var/www/www.skill39.wsr/html/index.html
2
3 <html>
4   <head>
5     <title>welcome to WSR!</title>
6   </head>
7   <body>
8     <h1>Success! The www.skill39.wsr virtual host is working!</h1>
9   </body>
10 </html>
```

8. Затем необходимо создать директории `sites-available` (конфигурации виртуальных хостов) и `sites-enabled` (символьные ссылки на конфигурации виртуальных хостов, которые готовы к обслуживанию)

```
1 | cd /etc/httpd/  
2 | mkdir sites-available sites-enabled
```

9. Указать глобально, что поиск хостов будет выполняться в директории `sites-enabled`

```
1 | nano conf/httpd.conf  
2 |  
3 | IncludeOptional sites-enabled/*.conf
```

10. Теперь можно создать новый виртуальный хост. Шаблон можно найти в директории `/usr/share/doc/httpd/`

```
1 | cd /usr/share/doc/httpd/  
2 | cp httpd-vhost.conf /etc/httpd/site-available/www.skill139.wsr.conf
```

11. Редактирование файла конфигурации виртуального хоста

```
1 | nano /etc/httpd/site-available/www.skill139.wsr.conf  
2 |  
3 | <VirtualHost *:80> #Виртуальный хост доступный при запросе на любой  
   адрес сервера и 80 порт  
4 |     ServerName www.skill139.wsr # имя, которое будет сравниваться, если  
   несколько хостов будет иметь одинаковые настройки  
5 |     ServerAlias skill139.wsr     # Другие имена, по которым будет доступен  
   данный хост  
6 |     DocumentRoot /var/www/www.skill139.wsr/html      # Точка начала сайта  
7 |     ErrorLog /var/www/www.skill139.wsr/log/error.log   # журнал ошибок  
8 |     CustomLog /var/www/www.skill139.wsr/log/requests.log combined #  
   журнал запросов  
9 | </VirtualHost>
```

12. Создать символическую ссылку (**абсолютный путь**) в директорию `sites-enabled`, чтобы показать, что хост готов к работе

```
1 | ln -s /etc/httpd/sites-available/www.skill139.wsr.conf /etc/httpd/sites-  
   enabled/www.skill139.wsr.conf
```

13. Перезагрузить сервер

```
1 | systemctl restart httpd
```

Источники

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-centos-7-ru>

<http://httpd.apache.org/>

<http://httpd.apache.org/docs/2.4/>

<http://httpd.apache.org/docs/2.4/configuring.html>

<http://httpd.apache.org/docs/2.4/sections.html>

[https://wiki.archlinux.org/index.php/Apache_HTTP_Server_\(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9\)](https://wiki.archlinux.org/index.php/Apache_HTTP_Server_(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9))

<https://unixhost.pro/clientarea/knowledgebase/12/ustanovka-i-nastroika-apache-php-mysql-na-centos.html>

<https://www.digitalocean.com/community/tutorials/apache-ubuntu-14-04-lts-ru>