

# Apache HTTP Server (Русский)

Ссылки по теме

- [PHP](#)
- [MySQL \(Русский\)](#)
- [PhpMyAdmin \(Русский\)](#)
- [Adminer](#)
- [Xampp \(Русский\)](#)
- [mod\\_perl](#)



Эта статья или раздел нуждается в [переводе](#)



Примечания: в английской версии накопилось много изменений (обсуждение: [Talk:Apache HTTP Server \(Русский\)#](#))



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

[Apache HTTP Server](#), или сокращенно Apache — популярный веб-сервер, разработанный Apache Software Foundation.

Apache часто используется вместе с языком сценариев PHP и СУБД MySQL. Такую комбинацию обычно называют [LAMP](#) (Linux, Apache, MySQL, PHP). Эта статья объясняет, как настроить Apache и как интегрировать с ним [PHP](#) и [MySQL](#).

Если вам нужно быстро создать окружение для разработки и тестирования, то можете просто установить [Xampp](#).

## Contents

[hide]

- [1Установка](#)
- [2Настройка](#)
  - [2.1Дополнительные опции](#)
  - [2.2Пользовательские каталоги](#)
  - [2.3TLS/SSL](#)
  - [2.4Виртуальные хосты](#)
    - [2.4.1Управление большим количеством виртуальных хостов](#)
- [3Расширения](#)
  - [3.1PHP](#)
    - [3.1.1Дополнительные параметры](#)
    - [3.1.2Использование php5 с php-fpm и mod\\_proxy\\_fcgi](#)
    - [3.1.3Использование php5 с apache2-mpm-worker и mod\\_fcgid](#)
    - [3.1.4MySQL/MariaDB](#)
- [4Решение проблем](#)
  - [4.1Просмотр журнала и текущего состояния Apache](#)
  - [4.2PID file /run/httpd/httpd.pid not readable \(yet?\) after start](#)
  - [4.3Обновление с Apache 2.2 до 2.4](#)
  - [4.4Apache is running a threaded MPM, but your PHP Module is not compiled to be threadsafe](#)

## Установка

Установите пакет [apache](#), доступный в [официальных репозиториях](#).

## Настройка

Файлы настроек Apache находятся в `/etc/httpd/conf`. Основным файлом является `/etc/httpd/conf/httpd.conf`, который может по ссылкам включать в себя дополнительные файлы с настройками. В большинстве случаев будет достаточно стандартных настроек из этого файла. По умолчанию корневым каталогом веб-сервера является `/srv/http`.

Для старта Apache [запустите](#) службу `httpd.service`.

После этих действий Apache должен запуститься. Проверьте работает ли он, набрав в адресной строке браузера <http://localhost/>. Веб-сервер должен отправить вам простую тестовую страничку.

При необходимости дальнейшей настройки сервера смотрите следующие разделы.

### Дополнительные опции

Следующие опции (*директивы*) в `/etc/httpd/conf/httpd.conf` могут быть вам интересны:

```
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
```

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**.

```
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](#).

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

Дополнительные настройки можно найти в `/etc/httpd/conf/extra/httpd-default.conf`.

Чтобы полностью отключить вывод версии Apache в генерируемых сервером страницах, добавьте:

```
ServerSignature Off
```

Чтобы подавить вывод такой информации, как версии Apache и PHP, добавьте:

```
ServerTokens Prod
```

## Пользовательские каталоги

По умолчанию доступ к каталогам пользователей возможен по адресу `http://localhost/~'user'/'`, который показывает содержимое каталога `~/public_html` (его имя и расположение задается в файле `/etc/httpd/conf/extra/httpd-userdir.conf`).

Если вы не хотите, чтобы пользовательские каталоги были доступны через web, закомментируйте следующую строку в `/etc/httpd/conf/httpd.conf`:

```
Include conf/extra/httpd-userdir.conf
```

Убедитесь, что права доступа к вашему домашнему каталогу и `~/public_html` позволяют получать доступ к файлам в них **всем** пользователям:

```
$ chmod o+x ~
$ chmod o+x ~/public_html
$ chmod -R o+r ~/public_html
```

Однако с точки зрения безопасности вышеприведенное решение слишком фривольно. Правильнее поступить по-другому. Сначала добавьте пользователя **http** в группу, которой принадлежит ваша домашняя папка. Например, если ваша домашняя папка и все ее подкаталоги принадлежат группе **piter**, можно проделать следующее:

```
# usermod -aG piter http
```

или

```
# gpasswd -a http piter
```

После этого назначьте права на *чтение и исполнение* для каталогов `~/`, `~/public_html` и, рекурсивно, на остальные подкаталоги для `~/public_html` для членов группы (в нашем

примере для членов группы **piter**). Опираясь на нижеприведенный шаблон, осуществите эти мероприятия:

```
$ chmod g+rx-w /home/yourusername
$ chmod -R g+rx-w /home/yourusername/public_html
```

**Примечание:** Таким образом, только пользователь **http** и все потенциальные пользователи группы **piter** будут иметь разделяемый доступ к вашему домашнему каталогу.

[Перезапустите](#) службу `httpd.service`, чтобы изменения вступили в силу. Смотрите также [Umask#Set the mask value](#).

## TLS/SSL

Для использования TLS/SSL необходимо установить [openssl](#).

Создайте закрытый ключ и запрос на получение сертификата (CSR). Также вы можете [самозаверить](#) CSR (который создаст сертификат):

**Примечание:** Вы можете настроить длину ключа в битах (`rsa_keygen_bits:2048`). Также вы можете убрать опцию `-sha256` для использования SHA-1 вместо SHA-2 или изменить время его действия в днях (`-days 365`).

```
# cd /etc/httpd/conf
# openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out
server.key
# chmod 600 server.key
# openssl req -new -sha256 -key server.key -out server.csr
# openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt
```

Теперь раскомментируйте следующие строки в `/etc/httpd/conf/httpd.conf`:

```
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
Include conf/extra/httpd-ssl.conf
```

[Перезапустите](#) службу `httpd.service`, чтобы изменения вступили в силу.

## Виртуальные хосты

**Примечание:** Вам нужно будет добавить отдельную секцию `<VirtualHost dommainame:443>` для поддержки SSL на виртуальном хосте. Пример файла можно посмотреть ниже: [#Управление большим количеством виртуальных хостов](#).

Если вы хотите, чтобы Apache обслуживал не один, а несколько хостов, раскомментируйте следующую строку в файле `/etc/httpd/conf/httpd.conf`:

```
Include conf/extra/httpd-vhosts.conf
```

Укажите свои виртуальные хосты в `/etc/httpd/conf/extra/httpd-vhosts.conf`. Файл уже содержит пример полностью рабочих настроек, что поможет вам быстро выполнить настройки под ваши нужды.

Для проверки виртуальных хостов на локальной машине добавьте их виртуальные имена в ваш файл `/etc/hosts`:

```
127.0.0.1 domainname1.dom
127.0.0.1 domainname2.dom
```

**Перезапустите** `httpd.service`, чтобы изменения вступили в силу.

### Управление большим количеством виртуальных хостов

Если Apache используется для обслуживания очень большого количества виртуальных хостов, вам может быть полезна возможность их легко включать и отключать. Для этого рекомендуется создавать собственный файл настроек на каждый хост и хранить все эти файлы в одном каталоге, например `/etc/httpd/conf/vhosts`.

Сначала создайте каталог:

```
# mkdir /etc/httpd/conf/vhosts
```

Теперь создайте в нем отдельные конфигурационные файлы:

```
# nano /etc/httpd/conf/vhosts/domainname1.dom
# nano /etc/httpd/conf/vhosts/domainname2.dom
...
```

И включите эти файлы в основной файл настроек `/etc/httpd/conf/httpd.conf`:

```
#Enabled Vhosts:
Include conf/vhosts/domainname1.dom
Include conf/vhosts/domainname2.dom
...
```

Теперь можно быстро включать/отключать требуемые виртуальные хосты, просто закомментировав или раскомментировав соответствующие директивы `Include` в основном файле настроек.

Очень простой файл виртуального хоста будет выглядеть следующим образом:

```
/etc/httpd/conf/vhosts/domainname1.dom

<VirtualHost domainname1.dom:80>
    ServerAdmin webmaster@domainname1.dom
    DocumentRoot "/home/user/http/domainname1.dom"
    ServerName domainname1.dom
    ServerAlias domainname1.dom
    ErrorLog "/var/log/httpd/domainname1.dom-error_log"
    CustomLog "/var/log/httpd/domainname1.dom-access_log" common

    <Directory "/home/user/http/domainname1.dom">
        Require all granted
    </Directory>
</VirtualHost>
```

```
<VirtualHost domainname1.dom:443>
    ServerAdmin webmaster@domainname1.dom
    DocumentRoot "/home/user/http/domainname1.dom"
    ServerName domainname1.dom:443
    ServerAlias domainname1.dom:443
    ErrorLog "/var/log/httpd/domainname1.dom-error_log"
    CustomLog "/var/log/httpd/domainname1.dom-access_log" common

    <Directory "/home/user/http/domainname1.dom">
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile "/etc/httpd/conf/server.crt"
    SSLCertificateKeyFile "/etc/httpd/conf/server.key"
</VirtualHost>
```

## Расширения

### PHP

[Установите PHP](#) с пакетами [php](#) и [php-apache](#), доступными в [официальных репозиториях](#).

**Примечание:** `libphp7.so` входящий в [php-apache](#), не работает с `mod_mpm_event` ([FS#39218](#)). Вместо него следует использовать `mod_mpm_prefork`. В противном случае, вы получите следующее сообщение об ошибке:

```
Apache is running a threaded MPM, but your PHP Module is not compiled to be
threadsafe. You need to recompile PHP.
AH00013: Pre-configuration failed
httpd.service: control process exited, code=exited status=1
```

Для использования `mod_mpm_prefork`, откройте `/etc/httpd/conf/httpd.conf` и поменяйте строку

```
LoadModule mpm_event_module modules/mod_mpm_event.so
```

на

```
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

Также вы можете просто использовать `mod_proxy_fcgi` (смотрите [#Использование php7 с php-fpm и mod\\_proxy\\_fcgi](#) [\[broken link: invalid section\]](#)).

Чтобы включить модуль PHP, добавьте следующие строки в `/etc/httpd/conf/httpd.conf`:

- Поместите эту строку в любом месте после строки `LoadModule dir_module modules/mod_dir.so`:

```
LoadModule php7_module modules/libphp7.so
```

- Разместите эту строку в конце списка `Include`:

```
Include conf/extra/php7_module.conf
```

**Примечание:** Если вы не обнаружите библиотеку `libphp7.so` в каталоге (`/etc/httpd/modules`), то вероятнее всего, что вы не установили [php-apache](#).

Если ваш корневой каталог `DocumentRoot` не `/srv/http`, добавьте его в список `open_basedir` в `/etc/php/php.ini`:

```
open_basedir=/srv/http:/home:/tmp:/usr/share/pear/:/путь/до/корневого/ката  
лога
```

[Перезапустите](#) службу `httpd.service`, чтобы изменения вступили в силу.

Чтобы убедиться в том, что PHP настроен корректно, создайте файл `test.php` в каталоге `DocumentRoot` (то есть в `/srv/http/` или `~/public_html`) и поместите в него следующий код:

```
<?php phpinfo(); ?>
```

По адресу `http://localhost/test.php` или `http://localhost/~пользователь/test.php` вы должны увидеть информационную страницу PHP.

Если PHP-код не выполняется, а на странице браузера вы увидите содержимое `test.php`, проверьте добавили ли вы `Includes` в строку `Options` для вашего корневого каталога в `/etc/httpd/conf/httpd.conf`. Кроме того, убедитесь, что `TypesConfig conf/mime.types` раскомментирован в секции `<IfModule mime_module>`. Также можно попробовать добавить нижеследующую строку в секцию `<IfModule mime_module>` файла `httpd.conf`:

```
AddHandler application/x-httpd-php .php
```

Дополнительную информацию вы можете получить на странице [PHP](#).

### Дополнительные параметры

Рекомендуется правильно настроить вашу временную зону ([список временных зон](#)) в `/etc/php/php.ini` по примеру:

```
date.timezone = Europe/Moscow
```

По желанию включите режим показа ошибок при отладке PHP-кода, для этого измените значение опции `display_errors` на `On` в файле `/etc/php/php.ini`:

```
display_errors=On
```

Если вы хотите использовать модуль `libGD`, установите [php-gd](#) и раскомментируйте `extension=gd.so` в `/etc/php/php.ini`:

**Примечание:** Пакет [php-gd](#) требует [libpng](#), [libjpeg-turbo](#) и [freetype2](#).

```
extension=gd.so
```

**Примечание:** Проверьте, какие расширения PHP вы раскомментируете. Подключайте только те расширения, которые необходимы и достаточны для работы.

Для использования модуля `mcrypt` установите [php-mcrypt](#) [ссылка недействительна](#): package not found и раскомментируйте `extension=mcrypt.so` в `/etc/php/php.ini`:

```
extension=mcrypt.so
```

Не забудьте добавить индексные файлы `/etc/httpd/conf/extra/php7_module.conf`, если это необходимо:

```
DirectoryIndex index.php index.phtml index.html
```

Для дополнительной настройки, пожалуйста прочтите [PHP](#).

### Использование php5 с php-fpm и mod\_proxy\_fcgi

**Примечание:** В отличие от широко распространенной установки с ProxyPass, конфигурация прокси с SetHandler использует директивы DirectoryIndex. Это обеспечивает лучшую совместимость с программным обеспечением, разработанным для libphp5, mod\_fastcgi и mod\_fcgid. Если вы хотите попробовать ProxyPass, используйте такую строку:

```
ProxyPassMatch ^/(.*\.php(/.*)?)$ unix:/run/php-fpm/php-fpm.sock|fcgi://localhost/srv/http/$1
```

- [Установите php-fpm](#)
- Задайте `listen` в `/etc/php/php-fpm.conf` следующим образом:

```
;listen = 127.0.0.1:9000
listen = /run/php-fpm/php-fpm.sock
listen.owner = http
listen.group = http
```

- Добавьте следующие строки в `/etc/httpd/conf/httpd.conf`:

```
<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php-fpm/php-fpm.sock|fcgi://localhost/"
</FilesMatch>
<IfModule dir_module>
    DirectoryIndex index.php index.html
```



```
</IfModule>
```

- Если у вас добавлен модуль php, уберите его, так как он больше не нужен:

```
#LoadModule php5_module modules/libphp5.so
```

- [Перезапустите](#) демон apache php-fpm:

```
# systemctl restart httpd.service php-fpm.service
```

### Использование php5 с apache2-mpm-worker и mod\_fcgid

- Раскомментируйте следующую строку в `/etc/conf.d/apache`:

```
HTTPD=/usr/sbin/httpd.worker
```

- Раскомментируйте следующую строку в `/etc/httpd/conf/httpd.conf`:

```
Include conf/extra/httpd-mpm.conf
```

- [Установите](#) пакеты [mod\\_fcgid](#) и [php-cgi](#), доступные в [официальных репозиториях](#).

- Создайте файл `/etc/httpd/conf/extra/php5_fcgid.conf` со следующим содержимым:

```
/etc/httpd/conf/extra/php5_fcgid.conf
-----
# Required modules: fcgid_module

<IfModule fcgid_module>
    AddHandler php-fcgid .php
    AddType application/x-httpd-php .php
    Action php-fcgid /fcgid-bin/php-fcgid-wrapper
    ScriptAlias /fcgid-bin/ /srv/http/fcgid-bin/
    SocketPath /var/run/httpd/fcgidsock
    SharememPath /var/run/httpd/fcgid_shm
    # If you don't allow bigger requests many applications may fail (such
as WordPress login)
    FcgidMaxRequestLen 536870912
    PHP_Fix_Pathinfo_Enable 1
    # Path to php.ini - defaults to /etc/phpX/cgi
    DefaultInitEnv PHPRC=/etc/php/
    # Number of PHP childs that will be launched. Leave undefined to let
PHP decide.
```

```
#DefaultInitEnv PHP_FCGI_CHILDREN 3
# Maximum requests before a process is stopped and a new one is
launched
#DefaultInitEnv PHP_FCGI_MAX_REQUESTS 5000
<Location /fcgid-bin/>
    SetHandler fcgid-script
    Options +ExecCGI
</Location>
</IfModule>
```

- Создайте каталог и символическую ссылку в нем на *php-cgi*:

```
# mkdir /srv/http/fcgid-bin
# ln -s /usr/bin/php-cgi /srv/http/fcgid-bin/php-fcgid-wrapper
```

- Отредактируйте `/etc/httpd/conf/httpd.conf`:

```
#LoadModule php5_module modules/libphp5.so
LoadModule fcgid_module modules/mod_fcgid.so
Include conf/extra/php5_fcgid.conf
```

и [перезапустите](#) службу `httpd.service`.

**Примечание:** Так же как и в Apache 2.4 вы можете использовать [mod proxy fcgi](#) вместе с PHP-FPM. Пример конфигурации вы можете найти на странице [\[1\]](#).

## MySQL/MariaDB

Следуйте инструкциям на странице [PHP \(Русский\)#MySQL/MariaDB](#).

После выполнения настройки, [перезапустите](#) службу `mysqld`, чтобы изменения вступили в силу.

## Решение проблем

### Просмотр журнала и текущего состояния Apache

Текущее состояние службы `httpd` вы можете вывести командой `systemctl status httpd`.

Лог-файлы Apache вы найдете в каталоге `/var/log/httpd`.

### PID file `/run/httpd/httpd.pid` not readable (yet?) after start

Если вы получаете такую ошибку, прокомментируйте строку:

```
LoadModule unique_id_module modules/mod_unique_id.so
```

в файле настроек Apache.

### Обновление с Apache 2.2 до 2.4

Если вы используете `php-apache`, посмотрите инструкции к [Apache с PHP](#) выше.

Управление доступом было изменено. Приведите все директивы `Order`, `Allow`, `Deny` и `Satisfy` к новому синтаксису

с Require. [mod\\_access\\_compat](#) позволит использовать устаревший формат на время этапа перехода.

Подробную информацию вы найдете на странице [Upgrading to 2.4 from 2.2](#).

## Apache is running a threaded MPM, but your PHP Module is not compiled to be threadsafe

Если не удалось запустить `php5_module` при старте `httpd.service` и вы получаете следующее сообщение об ошибке:

```
Apache is running a threaded MPM, but your PHP Module is not compiled to be threadsafe. You need to recompile PHP.
```

Это значит, что Apache работает с поточным MPM, но используется не потокобезопасная версия PHP. В этом случае, следует заменить `mpm_event_module` на `mpm_prefork_module`:

```
/etc/httpd/conf/httpd.conf

# LoadModule mpm_event_module modules/mod_mpm_event.so
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so
```

и перезапустить `httpd.service`.

- [Официальный сайт веб-сервера Apache](#)
- [Руководство по созданию самоподписанных сертификатов](#)
- [Поиск и устранение неисправностей при настройке Apache](#)
- [Официальный сайт MariaDB](#)
- [Официальный сайт PHP](#)

[Categories:](#)

- [Web server \(Русский\)](#)
- [Apache \(Русский\)](#)
- [Русский](#)