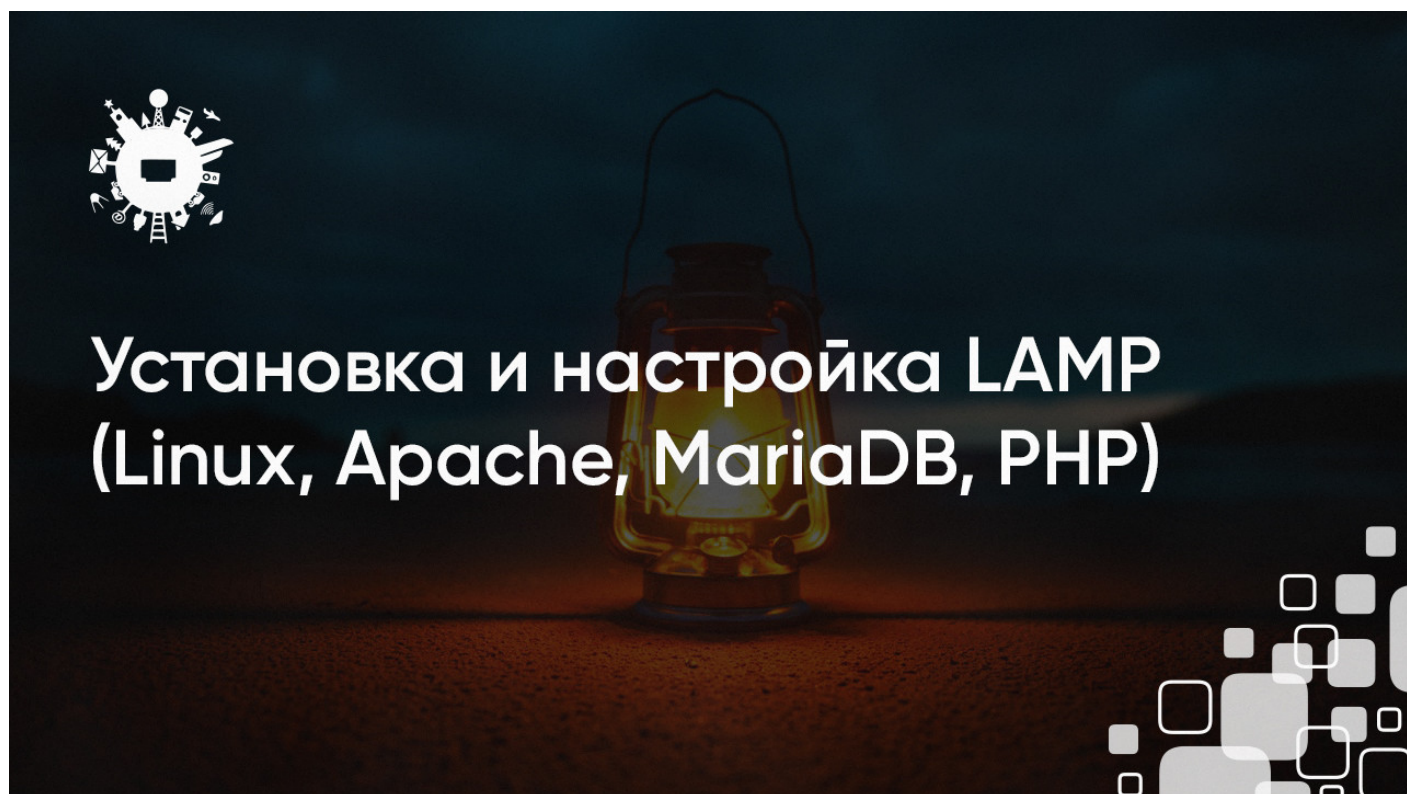


# Установка LAMP



## Введение

Стек «LAMP» — это группа программного обеспечения с открытым исходным кодом, которое обычно устанавливается вместе, чтобы позволить серверу размещать динамические веб-сайты и веб-приложения. Этот термин представляет собой аббревиатуру, обозначающую операционную систему Linux с веб-сервером Apache. Данные сайта хранятся в базе данных MariaDB, а динамический контент обрабатывается PHP.

Хотя этот программный стек обычно включает MySQL в качестве системы управления базами данных, некоторые дистрибутивы Linux, включая Debian, используют MariaDB в качестве замены MySQL.

В этом руководстве вы установите стек LAMP на сервер Debian 11, используя MariaDB в качестве системы управления базой данных.

## Шаг 1 — Установка Apache

Веб-сервер Apache является одним из самых популярных веб-серверов в мире. Он хорошо задокументирован, имеет активное сообщество пользователей и широко использовался на протяжении большей части истории Интернета, что делает его отличным выбором для размещения веб-сайта.

Начните с обновления кеша менеджера пакетов. Если вы впервые используете `sudo` в этом сеансе, вам будет предложено ввести пароль пользователя, чтобы подтвердить, что у вас есть права на управление системными пакетами с помощью `apt`:

```
sudo apt update
```

Затем установите Apache такой командой:

```
sudo apt install apache2
```

Эта команда предложит вам подтвердить установку Apache. Подтвердите, написав `Y`, затем `ENTER`.

Вы можете убедиться, что все прошло по плану, посетив общедоступный IP-адрес вашего сервера в веб-браузере:

```
http://ip_вашего_сервера
```

После перехода по адресу у Вас должна открыться тестовая страница:



## Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in** [`/usr/share/doc/apache2/README.Debian.gz`](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

## Шаг 2 — Установка MariaDB

Теперь, когда у вас есть запущенный и работающий веб-сервер, вам необходимо установить систему базы данных, чтобы иметь возможность хранить и управлять данными для вашего сайта.

Чтобы установить программное обеспечение MariaDB, запустите команду:

```
sudo apt install mariadb-server
```

После завершения установки рекомендуется запустить сценарий безопасности, который предварительно устанавливается вместе с MariaDB. Этот сценарий удалит некоторые небезопасные настройки по умолчанию и заблокирует доступ к вашей системе баз данных. Запустите интерактивный скрипт, выполнив:

```
sudo mysql_secure_installation
```

Этот сценарий проведет вас через ряд подсказок, где вы можете внести некоторые изменения в настройку MariaDB. Первое приглашение попросит вас ввести текущий пароль root базы данных. Не следует путать с системным паролем. Пользователь root базы данных является администратором с полными привилегиями в системе базы данных. Поскольку вы только что установили MariaDB и еще не вносили никаких изменений в конфигурацию, этот пароль будет пустым, поэтому нажмите ENTER в командной строке.

Следующее приглашение спрашивает, хотите ли вы установить пароль root для базы данных. Поскольку MariaDB использует специальный метод аутентификации для пользователя root, который обычно более безопасен, чем использование пароля, вам не нужно устанавливать его сейчас. Нажмите N, а затем ENTER.

Оттуда вы можете нажать Y, а затем ENTER, чтобы принять значения по умолчанию для всех последующих вопросов. Это удалит анонимных пользователей и тестовую базу данных, отключит удаленный вход в систему root и загрузит эти новые правила, чтобы MariaDB сразу учитывала внесенные Вами изменения.

Когда вы закончите, войдите в консоль MariaDB:

```
sudo mariadb
```

Вы должны получить такой ответ:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Обратите внимание, что вам не нужно было вводить пароль для подключения в качестве пользователя root. Это связано с тем, что методом аутентификации по умолчанию для

административного пользователя MariaDB является `unix_socket` вместо пароля. Хотя на первый взгляд это может показаться проблемой безопасности, это делает сервер базы данных более безопасным, поскольку единственными пользователями, которым разрешено входить в систему в качестве пользователя `root` MariaDB, являются системные пользователи с привилегиями `sudo`, подключающиеся из консоли или через приложение, работающее с теми же привилегиями. На практике это означает, что вы не сможете использовать администратора базы данных `root` для подключения из вашего PHP-приложения.

Для повышения безопасности лучше иметь выделенные учетные записи пользователей с менее широкими привилегиями, настроенными для каждой базы данных, особенно если вы планируете разместить несколько баз данных на своем сервере.

Выйти из консоли MariaDB можно следующим образом:

```
exit
```

## Шаг 3 — Установка PHP

У вас установлен Apache для обслуживания вашего контента и установлена MariaDB для хранения и управления вашими данными. PHP — это компонент вашей установки, который будет обрабатывать код для отображения динамического содержимого конечному пользователю. Он может запускать сценарии, подключаться к вашим базам данных MariaDB для получения информации и передавать обработанный контент на ваш веб-сервер для отображения.

В дополнение к пакету `php` вам понадобится `php-mysql`, модуль PHP, который позволяет PHP взаимодействовать с базой данных на основе MySQL, такой как MariaDB. Вам также понадобится `libapache2-mod-php`, чтобы Apache мог обрабатывать файлы PHP. Базовые пакеты PHP будут автоматически установлены как зависимости.

Чтобы установить эти пакеты, выполните следующую команду:

```
sudo apt install php libapache2-mod-php php-mysql
```

После завершения установки вы можете проверить свою версию PHP с помощью следующей команды:

```
php -v
```

Вы должны получить такой вывод:

```
PHP 7.4.30 (cli) (built: Jul 7 2022 15:51:43) ( NTS )  
Copyright (c) The PHP Group
```

На этом этапе ваш стек LAMP полностью готов к работе, но перед тестированием вашей установки с помощью PHP-скрипта лучше всего настроить правильный виртуальный хост Apache для хранения файлов и папок вашего веб-сайта.

## Шаг 4 — Создание виртуального хоста Apache

При использовании веб-сервера Apache вы можете создавать виртуальные хосты (аналогично блокам сервера в Nginx), чтобы инкапсулировать детали конфигурации и размещать более одного домена на одном сервере. В этом руководстве мы настроим домен с именем `vash_domain`, но вы должны заменить его своим собственным доменным именем.

По умолчанию Apache предоставляет свое содержимое из каталога, расположенного в `/var/www/html`, используя конфигурацию, содержащуюся в `/etc/apache2/sites-available/000-default.conf`. Вместо изменения файла конфигурации веб-сайта по умолчанию `/var/www/html` вы создадите новый виртуальный хост для тестирования среды PHP. Виртуальные хосты позволяют размещать несколько веб-сайтов на одном сервере Apache. Вы также создадите структуру каталогов в `/var/www` для сайта `vash_domain`, оставив `/var/www/html` в качестве каталога по умолчанию, который будет обслуживаться, если запрос клиента не соответствует каким-либо другим сайтам.

Начните с создания корневого веб-каталога для `vash_domain` следующим образом:

```
sudo mkdir /var/www/vash_domain
```

Затем назначьте владельца каталога с помощью переменной окружения `$USER`, которая будет ссылаться на вашего текущего системного пользователя:

```
sudo chown -R $USER:$USER /var/www/vash_domain
```

Затем откройте новый файл конфигурации в каталоге, доступном для сайтов Apache, с помощью предпочитаемого вами редактора командной строки. Здесь мы будем использовать `nano`:

```
sudo nano /etc/apache2/sites-available/vash_domain.conf
```

Команда выше создаст новый пустой файл. Добавьте следующую базовую конфигурацию с вашим собственным доменным именем:

```
<VirtualHost *:80>
    ServerName vash_domain
    ServerAlias www.vash_domain
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/vash_domain
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Сохраните и закройте файл, когда закончите. Если вы используете nano, вы можете сделать это, нажав CTRL + X, затем Y и ENTER.

С помощью данной конфигурации VirtualHost вы говорите Apache обслуживать vash\_domain, используя /var/www/vash\_domain в качестве корневого веб-каталога. Если вы хотите протестировать Apache без доменного имени, вы можете удалить или закомментировать параметры ServerName и ServerAlias, добавив знак решетки (#) в начало строк каждого параметра.

Теперь используйте команду a2ensite, чтобы включить этот виртуальный хост:

```
sudo a2ensite vash_domain
```

Возможно, вы захотите отключить веб-сайт по умолчанию, который устанавливается вместе с Apache. Это необходимо, если вы не используете собственное доменное имя, потому что в этом случае конфигурация Apache по умолчанию переопределяет ваш виртуальный хост. Чтобы отключить веб-сайт Apache по умолчанию, запустите:

```
sudo a2dissite 000-default
```

Чтобы убедиться, что ваш файл конфигурации не содержит синтаксических ошибок, выполните следующую команду:

```
sudo apache2ctl configtest
```

Наконец, перезагрузите Apache, чтобы эти изменения вступили в силу:

```
sudo systemctl reload apache2
```

Далее вы создадите PHP-скрипт для проверки правильности установки и настройки PHP на вашем сервере.

## Шаг 5 — Тестирование работы PHP

Теперь, когда у вас есть специальное место для размещения файлов и папок вашего веб-сайта, создайте тестовый скрипт PHP, чтобы убедиться, что Apache может обрабатывать и обрабатывать запросы на файлы PHP.

```
nano /var/www/vash_domain/info.php
```

Команда откроет пустой файл. Добавьте в файл следующий текст, который является кодом PHP:

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

Сохраните и закройте файл, когда закончите. Если вы используете nano, вы можете сделать это, нажав CTRL + X, затем Y и ENTER.

Чтобы протестировать скрипт, откройте свой веб-браузер и перейдите по доменному имени или IP-адресу вашего сервера, а затем к имени скрипта, в данном случае это info.php:

```
http://vash_domain/info.php
```

Вы должны увидеть такую страницу:



System	Linux lamp 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64
Build Date	Jul 7 2022 15:51:43
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	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, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
 Zend Engine v3.4.0, Copyright (c) Zend Technologies  
 with Zend OPcache v7.4.30, Copyright (c), by Zend Technologies



На этой странице представлена информация о вашем сервере с точки зрения PHP. Это полезно для отладки и проверки правильности применения ваших настроек.

Если вы видите эту страницу в своем браузере, ваша установка PHP работает должным образом.

После проверки соответствующей информации о вашем PHP-сервере на этой странице лучше всего удалить созданный вами файл, поскольку он содержит конфиденциальную информацию о вашей среде PHP и вашем сервере Debian. Для этого используйте команду rm:

```
sudo rm /var/www/vash_domain/info.php
```

На этом установка и тестирование LAMP завершено.

Версия #15

Artem создал 13 августа 2023 16:34:19

Artem обновил 21 августа 2023 06:59:25