**[УСТАНОВКА WORDPRESS НА СТЕК LEMP](https://www.8host.com/blog/ustanovka-wordpress-na-stek-lemp/" \o "Установка WordPress на стек LEMP)**

22 марта, 2023 12:34 пп 22 views | Комментариев нет

[LEMP Stack](https://www.8host.com/blog/category/lemp-stack/), [Ubuntu](https://www.8host.com/blog/category/ubuntu/) | [Amber](https://www.8host.com/blog/author/amber/) | [Комментировать запись](https://www.8host.com/blog/ustanovka-wordpress-na-stek-lemp/#respond)

[WordPress](https://wordpress.com/ru/support/getting-started-with-wordpress-com/) — это популярная система управления контентом (CMS) для создания сайтов и блогов с помощью бэкенда MySQL с обработкой PHP. Это отличный выбор для быстрого запуска сайта. После начальной настройки администрирование практически всех WordPress-сайтов можно выполнять через графический интерфейс.

В этом мануале мы рассмотрим настройку экземпляра WordPress через LEMP (Linux, Nginx, MySQL и PHP) на сервере Ubuntu.

Требования

Для выполнения мануала требуется доступ к серверу Ubuntu и вам нужно будет пройти следующие этапы:

* Создайте пользователя sudo: все этапы этого туториала выполняются через пользователя non-root с привилегиями sudo. Вы можете создать пользователя sudo c помощью [мануала по начальной настройке сервера Ubuntu](https://www.8host.com/blog/nachalnaya-nastrojka-ubuntu-22-04/).
* Установите стек LEMP: для корректной работы WordPress нужен сервер, база данных и PHP. Следуйте [этому гайду](https://www.8host.com/blog/ustanovka-steka-lemp-v-ubuntu-20-04/), чтобы установить и настроить это программное обеспечение.
* Обеспечьте безопасность сайта с помощью SSL: WordPress обслуживает динамический контент и обрабатывает аутентификацию и авторизацию пользователей. TLS/SSL — это технология, которая позволяет шифровать трафик на сайте, чтобы ваше соединение с пользователями было безопасным. Способ настройки SSL зависит от того, есть ли у вас домен:
  + Если у вас есть домен, вы можете защитить свой сайт с помощью Let’s Encrypt, который предоставляет бесплатные доверенные сертификаты. Чтобы его настроить, выполните [этот туториал](https://www.8host.com/blog/sozdanie-sertifikata-lets-encrypt-dlya-nginx-v-ubuntu-20-04/).
  + Если у вас нет домена и вы используете эту конфигурацию только для тестирования или практики, вы можете использовать самоподписанный сертификат. Это обеспечит тот же тип шифрования, но без проверки домена. Следуйте [этому мануалу](https://www.8host.com/blog/sozdanie-samopodpisannogo-ssl-sertifikata-dlya-nginx-v-ubuntu-18-04/), чтобы настроить его.

Выполнив все предварительные условия, зайдите на свой сервер как пользователь sudo.

1: Создание базы данных MySQL и пользователя для WordPress

WordPress использует MySQL для управления и хранения информации о сайте и пользователях. MySQL у нас уже установлен, но нам нужно создать базу данных и пользователя для работы с WordPress.

Для начала войдите в root (административную) учетную запись MySQL. Если MySQL настроен на использование плагина аутентификации auth\_socket (по умолчанию), вы можете войти в учетную запись администратора MySQL с помощью sudo:

sudo mysql

Если вы изменили метод аутентификации, чтобы использовать пароль для учетной записи MySQL root, выполните следующую команду:

mysql -u root -p

Вам будет предложено ввести пароль, который вы установили для root пользователя MySQL при инсталляции программы.

После входа в систему создайте отдельную базу данных для управления WordPress. У нас она будет условно называться wordpress, но вы можете выбрать любое название. Создайте БД для WordPress с помощью следующей команды:

CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8\_unicode\_ci;

**Примечание**: каждый оператор MySQL должен заканчиваться точкой с запятой (;). Если у вас возникнут какие-либо проблемы, проверьте наличие этого знака.

Затем мы создадим отдельную учетную запись MySQL, которую будем использовать исключительно для работы с нашей новой базой данных. Создавая индивидуальные БД и пользователей, вы повышаете безопасность и упрощаете управление установкой. В этом гайде аккаунт будет условно называться wordpressuser, но вы можете выбрать любое имя на свое усмотрение.

Мы создадим эту учетную запись, установим пароль и предоставим доступ к созданной нами БД. Чтобы сделать это, выполните следующую команду. Не забудьте указать надежный пароль:

CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL ON wordpress.\* TO 'wordpressuser'@'localhost';

Теперь у вас есть база данных и учетная запись пользователя, созданные специально для WordPress.

Выйдите из MySQL, выполнив следующую команду:

EXIT;

Сеанс MySQL завершится и вы вернетесь в обычную оболочку Linux.

2: Установка дополнительных расширений PHP

При настройке стека LEMP нам требовался минимальный набор расширений, чтобы PHP мог взаимодействовать с MySQL. Однако WordPress и многие из его плагинов используют дополнительные расширения PHP.

Теперь мы загрузим и установим несколько популярных расширений PHP для работы с WordPress, выполнив следующие команды:

sudo apt update  
sudo apt install php-curl php-gd php-intl php-mbstring php-soap php-xml php-xmlrpc php-zip

**Примечание**: у каждого плагина WordPress свой набор требований. Некоторым необходимы дополнительные пакеты PHP. Проверьте документацию плагина, чтобы узнать его требования к PHP. Если PHP соответствует этим требованиям, то плагин можно установить с помощью apt, как показано выше.

После установки расширений необходимо перезапустить процесс PHP-FPM, чтобы запущенный процессор PHP мог использовать только что установленные функции:

sudo systemctl restart php8.1-fpm

Теперь у вас установлены все необходимые расширения PHP.

3: Настройка Nginx

Далее давайте внесем несколько изменений в server блоки Nginx. Исходя из предыдущих туториалов, у вас должен быть файл конфигурации в директории /etc/nginx/sites-available/, отвечающий на домен или IP-адрес вашего сервера и защищенный сертификатом TLS/SSL. В этом гайде мы используем путь /etc/nginx/sites-available/wordpress в качестве примера, но вы должны заменить его вашим путем, если он отличается.

Кроме того, мы будем использовать /var/www/wordpress в качестве root каталога для установки WordPress. Вы должны использовать root директорию, указанную в вашей конфигурации.

**Примечание**: возможно, вы используете конфигурацию по умолчанию /etc/nginx/sites-available/default (/var/www/html в качестве root каталога сайта). Это подходит для размещения только одного сайта на этом сервере. Если вы планируете размещать несколько сайтов, то лучше разбить конфигурацию на логические блоки, по одному файлу на сайт.

Откройте файл server блока вашего сайта с привилегиями sudo:

sudo nano /etc/nginx/sites-available/wordpress

В главном server блоке разместите несколько блоков location.

Начните с создания блоков location для запросов к /favicon.ico и /robots.txt, для которых не нужно вести лог запросов.

Используйте location по регулярному выражению, чтобы сопоставлять любые запросы на статические файлы. Мы снова отключим логирование для этих запросов и пометим их как высоко-кэшируемые, поскольку обычно они требуют значительных затрат для обслуживания. Вы можете настроить список статических файлов, чтобы включать любые другие расширения файлов, которые использует ваш сайт:

server {

   . . .

   location = /favicon.ico { log\_not\_found off; access\_log off; }

   location = /robots.txt { log\_not\_found off; access\_log off; allow all; }

   location ~\* \.(css|gif|ico|jpeg|jpg|js|png)$ {

       expires max;

       log\_not\_found off;

   }

    . . .

}

Внутри существующего location block отредактируем список try\_files. Закомментируйте настройку по умолчанию, поставив перед строкой знак решетки (#), а затем добавьте выделенную строку. Таким образом, вместо возврата кода ошибки 404 в качестве стандартной опции, управление передается файлу index.php с аргументами запроса.

Это должно выглядеть примерно так:

server {

   . . .

   location / {

       #try\_files $uri $uri/ =404;

       try\_files $uri $uri/ /index.php$is\_args$args;

   }

   . . .

}

Когда закончите, сохраните и закройте файл.

Теперь давайте проверим конфигурацию на наличие синтаксических ошибок:

sudo nginx -t

Если ошибок не обнаружено, перезагрузите Nginx, выполнив следующую команду:

sudo systemctl reload nginx

Далее загрузим и настроим WordPress.

4: Загрузка WordPress

После настройки серверного программного обеспечения мы можем загрузить и настроить WordPress. Для обеспечения безопасности следует загружать последнюю версию с официального сайта.

Перейдите в записываемый каталог и загрузите сжатый релиз:

cd /tmp

Команда изменит вашу текущую директорию на временную папку. Затем введите следующую команду, чтобы загрузить последнюю версию WordPress в сжатом файле:

curl -LO https://wordpress.org/latest.tar.gz

**Примечание**: флаг -LO используется для прямого доступа к источнику сжатого файла. Флаг -L обеспечивает успешное получение файла в случае перенаправления, а -O записывает вывод удаленного файла в локальный файл с тем же названием.

**Читайте также**: [Загрузка файлов с помощью cURL](https://www.8host.com/blog/zagruzka-fajlov-s-pomoshhyu-curl/)

Распакуйте сжатый файл, чтобы создать структуру каталогов WordPress:

tar xzvf latest.tar.gz

Сейчас мы переместим эти файлы в корневой каталог, но перед этим давайте скопируем образец файла конфигурации в файл, который будет читать WordPress:

cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php

Теперь мы можем скопировать все содержимое каталога в корневой каталог. Мы используем флаг -a, чтобы сохранить привилегии, и точку в конце исходной директории, чтобы указать, что нужно скопировать все содержимое, включая скрытые файлы:

sudo cp -a /tmp/wordpress/. /var/www/<^>your\_domain/wordpress

Теперь, когда файлы на месте, мы назначим владельца для пользователя и группы www-data. Это пользователь и группа, под которыми работает Nginx, и он должен иметь возможность читать и записывать файлы WordPress, чтобы обслуживать сайт и выполнять автоматические обновления:

sudo chown -R www-data:www-data /var/www/your\_domain/wordpress

Теперь файлы находятся в корневом каталоге сервера и имеют правильные права доступа, но ещё нужно выполнить дополнительную настройку.

5: Настройка WordPress

Теперь нам необходимо внести изменения в основной файл конфигурации WordPress.

Когда мы откроем файл, первой задачей будет настройка секретных ключей для безопасности установки. WordPress предоставляет генератор ключей, так что вам не придется придумывать сложные значения самостоятельно. Они используются только внутри системы, поэтому никак не повлияют на вашу работу.

Чтобы получить безопасные значения из генератора секретных ключей WordPress, выполните команду:

curl -s https://api.wordpress.org/secret-key/1.1/salt/

Вы получите уникальные значения, похожие на приведенный ниже блок.

**Внимание!** Для безопасности вашей установки важно каждый раз запрашивать уникальные значения. НЕ копируйте приведенный ниже блок в свой файл!

define('AUTH\_KEY',         '1jl/vqfs<XhdXoAPz9 DO NOT COPY THESE VALUES c\_j{iwqD^<+c9.k<J@4H');

define('SECURE\_AUTH\_KEY',  'E2N-h2]Dcvp+aS/p7X DO NOT COPY THESE VALUES {Ka(f;rv?Pxf})CgLi-3');

define('LOGGED\_IN\_KEY',    'W(50,{W^,OPB%PB<JF DO NOT COPY THESE VALUES 2;y&,2m%3]R6DUth[;88');

define('NONCE\_KEY',        'll,4UC)7ua+8<!4VM+ DO NOT COPY THESE VALUES #`DXF+[$atzM7 o^-C7g');

define('AUTH\_SALT',        'koMrurzOA+|L\_lG}kf DO NOT COPY THESE VALUES  07VC\*Lj\*lD&?3w!BT#-');

define('SECURE\_AUTH\_SALT', 'p32\*p,]z%LZ+pAu:VY DO NOT COPY THESE VALUES C-?y+K0DK\_+F|0h{!\_xY');

define('LOGGED\_IN\_SALT',   'i^/G2W7!-1H2OQ+t$3 DO NOT COPY THESE VALUES t6\*\*bRVFSD[Hi])-qS`|');

define('NONCE\_SALT',       'Q6]U:K?j4L%Z]}h^q7 DO NOT COPY THESE VALUES 1% ^qUswWgn+6&xqHN&%');

Это строки, которые мы можем вставить непосредственно в наш файл конфигурации для установки ключей. Скопируйте полученный вывод.

Затем откройте файл WordPress:

sudo nano /var/www/your\_domain/wordpress/wp-config.php

Найдите раздел, в котором содержатся примеры значений для этих параметров.

. . .

define('AUTH\_KEY',         'put your unique phrase here');

define('SECURE\_AUTH\_KEY',  'put your unique phrase here');

define('LOGGED\_IN\_KEY',    'put your unique phrase here');

define('NONCE\_KEY',        'put your unique phrase here');

define('AUTH\_SALT',        'put your unique phrase here');

define('SECURE\_AUTH\_SALT', 'put your unique phrase here');

define('LOGGED\_IN\_SALT',   'put your unique phrase here');

define('NONCE\_SALT',       'put your unique phrase here');

. . .

Удалите эти строки и вставьте значения, скопированные из командной строки:

. . .

define('AUTH\_KEY',         'VALUES COPIED FROM THE COMMAND LINE');

define('SECURE\_AUTH\_KEY',  'VALUES COPIED FROM THE COMMAND LINE');

define('LOGGED\_IN\_KEY',    'VALUES COPIED FROM THE COMMAND LINE');

define('NONCE\_KEY',        'VALUES COPIED FROM THE COMMAND LINE');

define('AUTH\_SALT',        'VALUES COPIED FROM THE COMMAND LINE');

define('SECURE\_AUTH\_SALT', 'VALUES COPIED FROM THE COMMAND LINE');

define('LOGGED\_IN\_SALT',   'VALUES COPIED FROM THE COMMAND LINE');

define('NONCE\_SALT',       'VALUES COPIED FROM THE COMMAND LINE');

. . .

Далее мы изменим несколько настроек соединения с базой данных, которые находятся в начале файла. Вам нужно настроить название БД, имя пользователя и связанный с ним пароль.

Ещё одно изменение, которое нам нужно внести, — это установить метод, который WordPress должен использовать для записи в файловую систему. Поскольку мы дали разрешение серверу записывать туда, куда ему нужно, мы можем установить метод файловой системы на “direct” (прямой). Если не установить этот параметр при нашей текущей настройке, WordPress будет запрашивать учетные данные FTP при выполнении некоторых действий. Этот параметр можно добавить ниже настроек соединения с базой данных или в любое другое место в файле:

. . .

define( 'DB\_NAME', 'wordpress' );

/\*\* MySQL database username \*/

define( 'DB\_USER', 'wordpressuser' );

/\*\* MySQL database password \*/

define( 'DB\_PASSWORD', 'password' );

. . .

define( 'FS\_METHOD', 'direct' );

Сохраните и закройте файл, когда закончите.

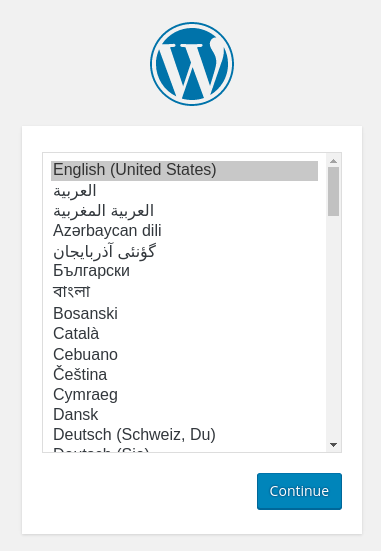
6: Завершение установки через веб-интерфейс

После настройки сервера завершим установку через веб-интерфейс.

В браузере перейдите к домену или внешнему IP-адресу вашего сервера:

http://server\_domain\_or\_IP/wordpress

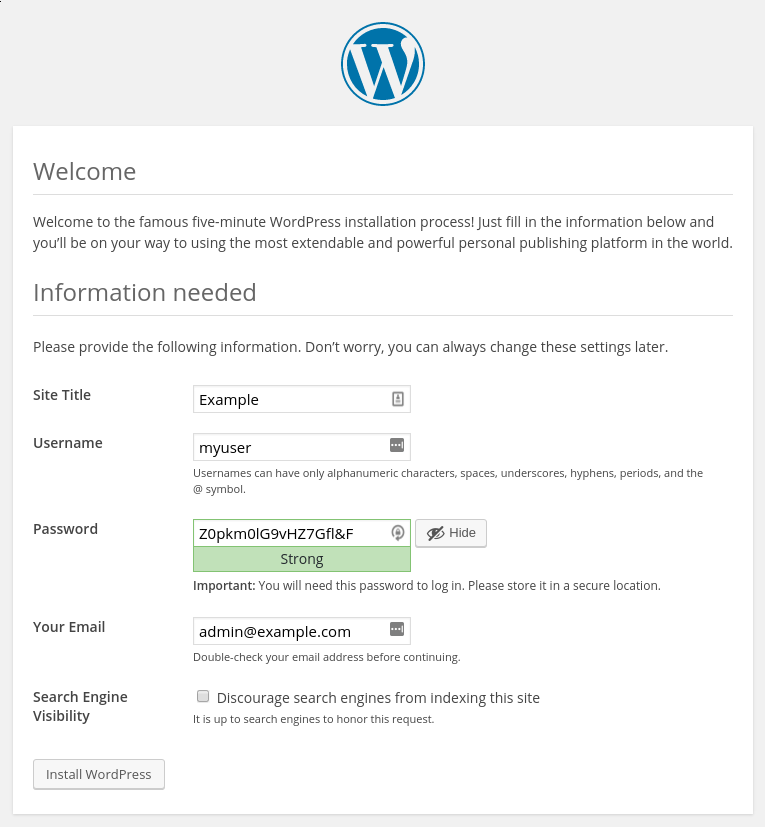
Выберите язык, который вы хотите использовать:



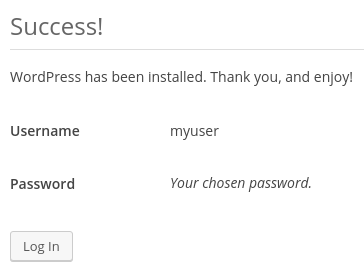
Далее вы попадете на главную страницу настройки.

Выберите название для вашего сайта WordPress и имя пользователя. В целях безопасности рекомендуется выбрать что-то уникальное и избегать распространенных имен, таких как “admin”. Надежный пароль генерируется автоматически. Сохраните этот или выберите другой пароль.

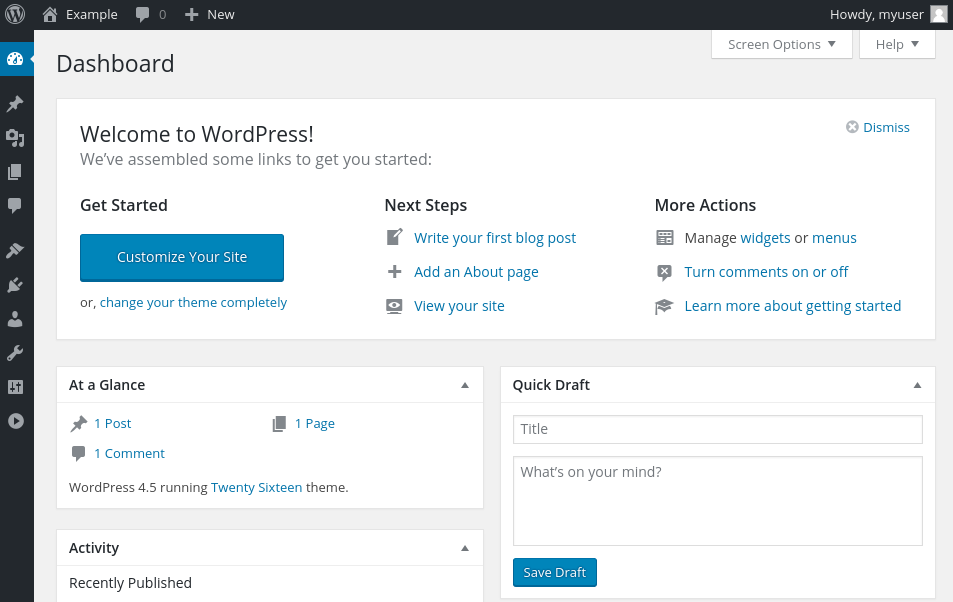
Введите свой имейл и укажите, хотите ли вы запретить поисковым системам индексировать ваш сайт:



Далее вы попадете на страницу, которая предложит вам войти в систему:



После входа в систему вы попадете в панель администратора WordPress:



Заключение

WordPress установлен и готов к работе! На этом этапе вы можете сделать следующее: выбрать настройки пермалинков для постов WordPress (их можно найти в разделе Settings > Permalinks) или изменить тему (Appearance > Themes). Если вы используете WordPress впервые, немного изучите интерфейс, чтобы ознакомиться с новой CMS.