

Устанавливаем веб-сервис

LAMP — это акроним, который обозначает определённый стек технологий, используемых для создания веб-приложений:

- серверная ОС **Linux**;
- веб-сервер **Apache**, для которого создано множество дополнительных модулей, которые решают вопрос совместной работы веб-сервера и сценариев, написанных на самых разных языках программирования;
- серверный язык динамических сценариев **PHP** (либо Perl, либо Python);
- СУБД **MySQL**. MySQL работает на Unix и Windows, отличается особым удобством в работе.

Взаимодействие элементов можно проследить в общем представлении порядка выполнения стека LAMP.

Процесс начинается с того, что веб-сервер получает от браузера пользователя запросы на веб-страницы. Если запрос касается файла PHP, то Apache передаёт запрос PHP, который загружает файл и выполняет содержащийся в нём код. Если в коде упоминаются данные, которые требуется извлечь, то PHP взаимодействует с MySQL.

Затем на основе кода в файле и данных, извлечённых из базы данных, PHP создаёт разметку HTML, согласно которой браузеры отображают веб-страницы.

После выполнения кода PHP передаёт полученные данные обратно на веб-сервер Apache для отправки в браузер. Также эти новые данные могут сохраняться в MySQL. Все эти операции поддерживаются операционной системой Linux, лежащей в основе всего стека.

Мы используем для нашего веб-сервиса **WordPress** — систему менеджмента контента. Веб-приложение, которое позволяет, например, создать блог, сайт-визитку, интернет-магазин и предоставляет удобные инструменты для настройки самого сайта и для манипуляции контентом.

Нам потребуется сервер приложений для того, чтобы развернуть на нём наш бэкэнд — систему менеджмента контента WordPress.

Создание логической базы данных в установленном сервере MySQL

Выполните вход в учётную запись MySQL с правами root:

```
mysql -u root -p
```

Вам будет предложено ввести пароль, который вы задали для учётной записи MySQL с правами root.

Сначала нам нужно создать отдельную базу данных, которую сможет контролировать WordPress. Создайте базу данных для WordPress, введя следующую команду:

```
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Затем мы создадим отдельную учётную запись пользователя MySQL, которую будем использовать исключительно для работы с нашей новой базы данных. В этом руководстве мы будем использовать имя wordpressuser. Вы можете использовать любое имя по вашему желанию.

Мы создадим учётную запись, установим пароль и предоставим доступ к созданной нами базе данных. Для этого мы воспользуемся следующей командой:

```
CREATE USER 'wordpressuser'@'localhost' IDENTIFIED with mysql_native_password BY PASSWORD 'qwerty';  
GRANT ALL ON wordpress.* TO 'wordpressuser'@'localhost';
```

Теперь у вас есть база данных и учётная запись пользователя, которые созданы специально для WordPress.

Нам нужно установить права, чтобы текущий экземпляр MySQL узнал о последних внесённых нами изменениях:

```
FLUSH PRIVILEGES;
```

Выйдите из MySQL, введя следующую команду:

```
EXIT;
```

Сессия MySQL будет закрыта, а вы вернётесь к стандартной командной строке Linux.

Установка дополнительных расширений PHP

При настройке нашего стека LEMP нам требуется только минимальный набор расширений, чтобы реализовать коммуникацию PHP с MySQL.

Мы можем загрузить и установить некоторые из самых популярных расширений PHP, которые будет использовать WordPress, введя следующую команду:

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

Примечание. Каждый плагин WordPress имеет собственный набор требований.

Завершив установку расширений, перезапустите процесс PHP-FPM, чтобы запущенный процессор PHP мог использовать новые установленные функции:

```
sudo systemctl restart php7.2-fpm
```

Настройка Nginx

Далее мы внесём несколько незначительных изменений в наши файлы серверного блока Nginx.

```
sudo vim /etc/nginx/sites-available/default
```

У нас уже есть имя сайта, его я оставила тем же. Для него есть и самоподписной сертификат, и запись в `/etc/hosts`.

В качестве корневой директории мы будем использовать `/var/www/wordpress`.

Внутри основного блока `server` нам нужно добавить несколько блоков `location`. Начнём с создания блоков с точным совпадением расположения для запросов к `/favicon.ico` и `/robots.txt`, для которых мы не хотим регистрировать запросы.

Можно изменить статичный список файлов и включить любые другие расширения файла, которые вы можете использовать на вашем сайте:

```
/etc/nginx/sites-available/wordpress
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 блока `location /` нам нужно изменить список `try_files`, чтобы вместо возвращения ошибки 404 по умолчанию управление передавалось файлу `index.php` с аргументами запроса.

Вы должны получить примерно следующий вывод:

```
/etc/nginx/sites-available/wordpress
server {
    . . .
    location / {
        #try_files $uri $uri/ =404;
        try_files $uri $uri/ /index.php$is_args$args;
```

```
}  
.  
.  
.  
}
```

Кроме того, добавим index.php в строчку с индексными файлами и раскомментируем строку для работы с PHP-FPM.

После завершения редактирования сохраните и закройте файл.

Теперь мы можем проверить нашу конфигурацию на наличие ошибок в синтаксисе, введя следующую команду:

```
sudo nginx -t
```

Если не будет выявлено ошибок, перезагрузите Nginx, введя следующую команду:

```
sudo systemctl reload nginx
```

Теперь мы загрузим и настроим непосредственно WordPress.

Загрузка WordPress

Перейдите в директорию с возможностью записи и загрузите сжатый релиз, введя следующую команду:

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

Извлеките сжатый файл для создания структуры директории WordPress:

```
tar xzvf latest.tar.gz
```

Мы сразу же перенесём эти файлы в корневую директорию документов. Перед этим мы можем скопировать пример файла конфигурации и присвоить имя файла, который считывается WordPress:

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

Теперь мы можем скопировать всё содержимое директории в корневую директорию. Мы будем использовать флаг -a для гарантии того, что наши разрешения поддерживаются. Мы используем точку в конце директории источника, чтобы указать, что всё содержимое директории должно быть скопировано, включая любые скрытые файлы.

```
sudo mkdir /var/www/wordpress  
sudo cp -a /tmp/wordpress/. /var/www/wordpress
```

Теперь, когда наши файлы на месте, мы укажем принадлежность к пользователю и группе www-data.

```
sudo chown -R www-data:www-data /var/www/wordpress
```

Теперь наши файлы находятся в корневой директории нашего сервера, мы указали принадлежность, но нам нужно выполнить дополнительную конфигурацию.

Настройка файла конфигурации WordPress

При первом открытии файла нам в первую очередь нужно будет изменить некоторые секретные ключи для обеспечения безопасности нашей установки. Чтобы получить безопасные значения из генератора секретных ключей WordPress, введите:

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

Вы получите уникальные значения, которые будут выглядеть следующим образом. Предупреждение. Важно каждый раз запрашивать уникальные значения. Не копируйте значения, приведённые ниже.

Output

```
define('AUTH_KEY', '1jl/vqfs<XhdXoAPz9 DO NOT COPY THESE VALUES  
...
```

Это строки конфигурации, которые мы можем вставить прямо в наш файл конфигурации для установки защищённых ключей. Скопируйте вывод, который вы получили.

Теперь откройте файл конфигурации WordPress:

```
sudo nano /var/www/wordpress/wp-config.php
```

Найдите раздел, содержащий фиктивные значения для этих настроек. Он будет выглядеть примерно так:

```
/var/www/wordpress/wp-config.php  
. . .  
  
define('AUTH_KEY',          'put your unique phrase here');
```

Удалите эти строки и добавьте значения, которые вы скопировали из командной строки:

```
/var/www/wordpress/wp-config.php  
. . .  
  
define('AUTH_KEY',          'VALUES COPIED FROM THE COMMAND LINE');  
. . .
```

Теперь нам нужно изменить некоторые настройки подключения базы данных в начале файла. Вам нужно изменить имя базы данных, пользователя базы данных и соответствующий пароль, которые мы настроили в MySQL.

Ещё одно изменение, которое нам нужно внести, — это настройка метода, который WordPress должен использовать для записи данных в файловой системе. Эта настройка может быть добавлена под настройками подключения базы данных или в любом другом месте в файле:

```
/var/www/wordpress/wp-config.php
. . .

define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'wordpressuser');

/** MySQL database password */
define('DB_PASSWORD', 'password');

. . .

define('FS_METHOD', 'direct');
```

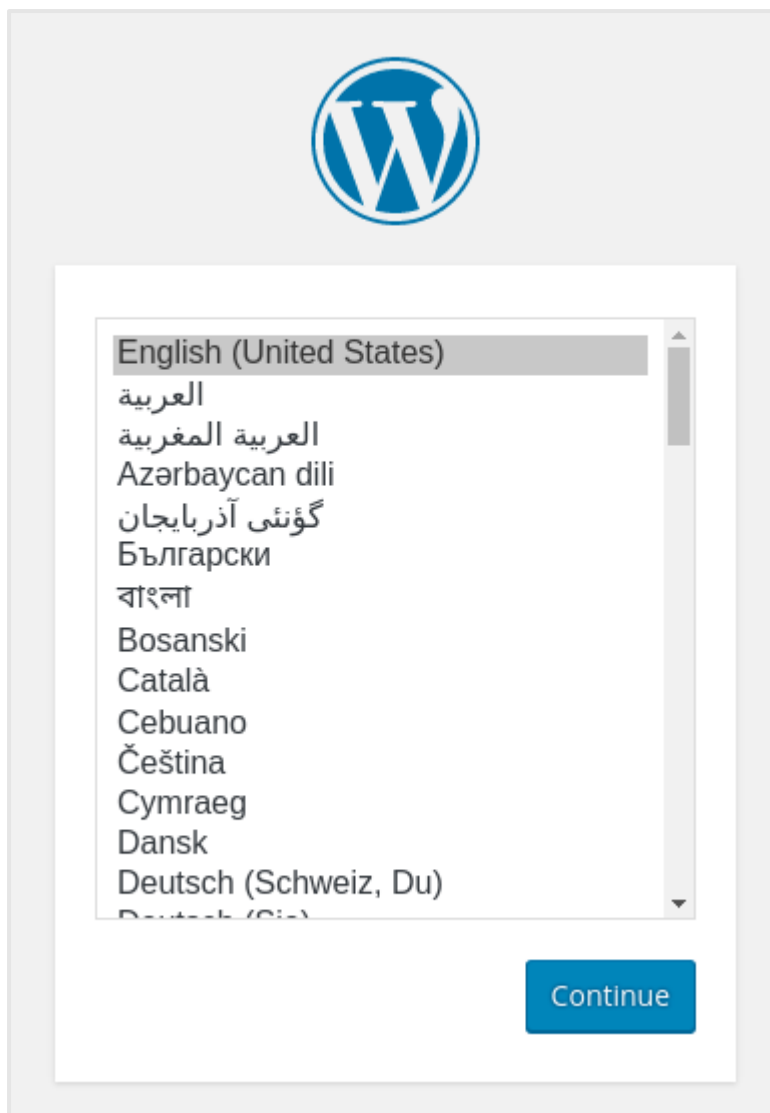
Сохраните файл и закройте его после завершения.

Завершение установки через веб-интерфейс

В адресной строке браузера введите доменное имя или открытый IP-адрес вашего сервера:

`http://server_domain_or_IP`

Выберите язык, который вы хотите использовать:



После этого вы должны попасть на главную страницу настройки.

Выберите имя для вашего сайта WordPress и имя пользователя (не рекомендуется использовать имя admin или что-то подобное в целях безопасности). Надёжный пароль создаётся автоматически. Сохраните этот пароль или выберите другой надёжный пароль.

Введите адрес электронной почты и укажите, хотите ли вы, чтобы поисковые машины индексировали ваш сайт:



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	<input type="text" value="Example"/>
Username	<input type="text" value="myuser"/> <small>Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.</small>
Password	<input type="password" value="Z0pkm0lG9vHZ7GfI&F"/> <div>Strong</div> <div><input type="button" value="Hide"/></div> <small>Important: You will need this password to log in. Please store it in a secure location.</small>
Your Email	<input type="text" value="admin@example.com"/> <small>Double-check your email address before continuing.</small>
Search Engine Visibility	<input type="checkbox"/> Discourage search engines from indexing this site <small>It is up to search engines to honor this request.</small>

При нажатии «Далее» вы перейдёте на страницу входа:

Success!

WordPress has been installed. Thank you, and enjoy!

Username	myuser
Password	<i>Your chosen password.</i>

Log In

После входа вы перейдёте в панель управления WordPress:

W Example 0 + New Howdy, myuser

Screen Options Help

Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

Get Started

Customize Your Site
or, [change your theme completely](#)

Next Steps

[Write your first blog post](#)
[Add an About page](#)
[View your site](#)

More Actions

[Manage widgets or menus](#)
[Turn comments on or off](#)
[Learn more about getting started](#)

At a Glance

1 Post

1 Page

1 Comment

WordPress 4.5 running [Twenty Sixteen](#) theme.

Activity

Recently Published

Quick Draft

Title

What's on your mind?

Save Draft