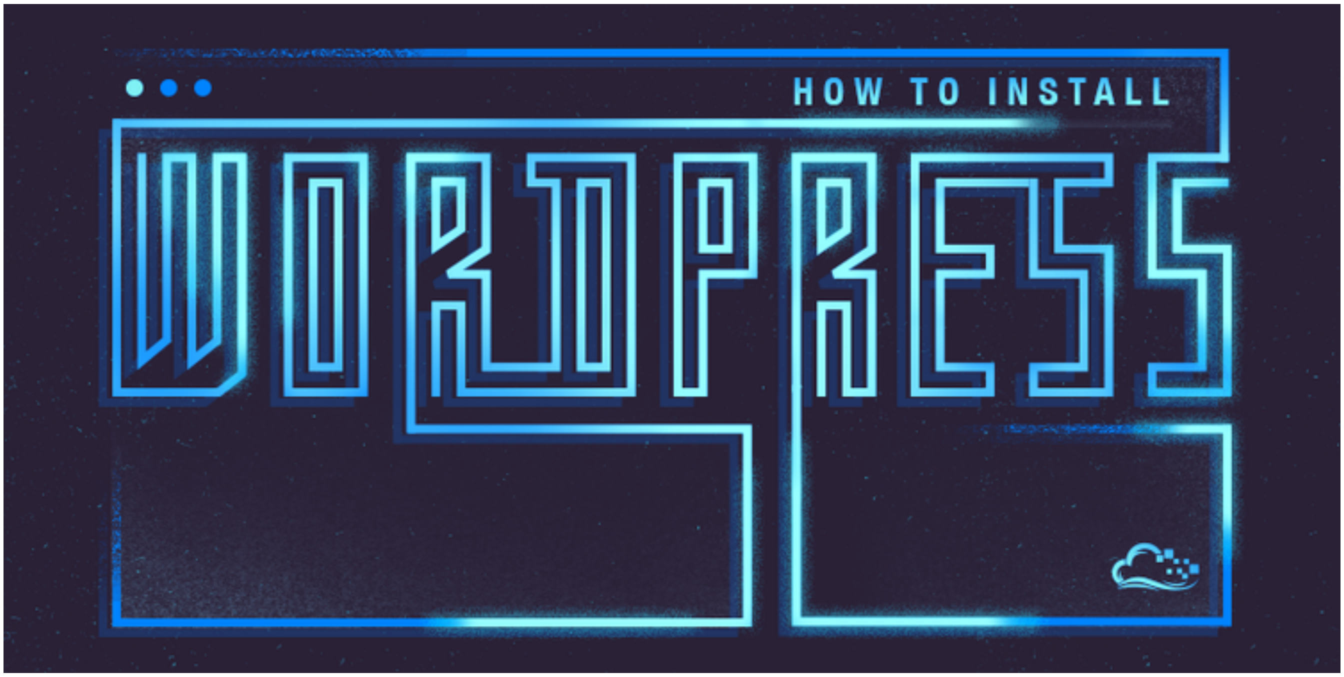
DigitalOcean | Community   Tutorials   Questions   Projects   Meetups
Log In   Sign Up

By: Justin Ellingwood   Subscribe   Share   Contents



How To Install WordPress with LAMP on Ubuntu 16.04

21

Posted April 22, 2016   141.6k   WORDPRESS   APACHE   LAMP STACK   UBUNTU   UBUNTU 16.04

Intr

Word ily
set u een
incre ost
all ad

In this guide, we'll focus on getting a WordPress instance set up on a LAMP stack (Lin
and PHP) on an Ubuntu 16.04 server.

SCROLL TO TOP

Prerequisites

In order to complete this tutorial, you will need access to an Ubuntu 16.04 server.

You will need to perform the following tasks before you can start this guide:

- **Create a `sudo` user on your server**: We will be completing the steps in this guide using a non-root user with `sudo` privileges. You can create a user with `sudo` privileges by following our Ubuntu 16.04 initial server setup guide.
- **Install a LAMP stack**: WordPress will need a web server, a database, and PHP in order to correctly function. Setting up a LAMP stack (Linux, Apache, MySQL, and PHP) fulfills all of these requirements. Follow this guide to install and configure this software.
- **Secure your site with SSL**: WordPress serves dynamic content and handles user authentication and authorization. TLS/SSL is the technology that allows you to encrypt the traffic from your site so that your connection is secure. The way you set up SSL will depend on whether you have a domain name for your site.
    - **If you have a domain name...** the easiest way to secure your site is with Let's Encrypt, which provides free, trusted certificates. Follow our Let's Encrypt guide for Apache to set this up.
    - **If you do not have a domain...** and you are just using this configuration for testing or personal use, you can use a self-signed certificate instead. This provides the same type of encryption, but without the domain validation. Follow our self-signed SSL guide for Apache to get set up.

When you are finished the setup steps, log into your server as your `sudo` user and continue below.

Step 1: Create a MySQL Database and User for WordPress

The first step that we will take is a preparatory one. WordPress uses MySQL to manage and store site and user information. We have MySQL installed already, but we need to make a database and a user for WordPress to use.

To ge

```
$
```

You will be prompted for the password you set for the MySQL root account when you installed the software.

SCROLL TO TOP

First, we can create a separate database that WordPress can control. You can call this whatever you would like, but we will be using `wordpress` in this guide to keep it simple. You can create the database for WordPress by typing:

```
mysql> CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci
```

> **Note:** Every MySQL statement must end in a semi-colon (;). Check to make sure this is present if you are running into any issues.

Next, we are going to create a separate MySQL user account that we will use exclusively to operate on our new database. Creating one-function databases and accounts is a good idea from a management and security standpoint. We will use the name `wordpressuser` in this guide. Feel free to change this if you'd like.

We are going to create this account, set a password, and grant access to the database we created. We can do this by typing the following command. Remember to choose a strong password here for your database user:

```
mysql> GRANT ALL ON wordpress.* TO 'wordpressuser'@'localhost' IDENTIFIED BY 'passw
```

You now have a database and user account, each made specifically for WordPress. We need to flush the privileges so that the current instance of MySQL knows about the recent changes we've made:

```
mysql> FLUSH PRIVILEGES;
```

Exit out of MySQL by typing:

```
my
```

Ste

When setting up our LAMP stack, we only required a very minimal set of extensions ir 'o communicate with MySQL. WordPress and many of its plugins leverage additional PHF extensions.

SCROLL TO TOP

We can download and install some of the most popular PHP extensions for use with WordPress by typing:

```
$ sudo apt-get update
$ sudo apt-get install php-curl php-gd php-mbstring php-mcrypt php-xml php-xmlrpc
```

Note

Each WordPress plugin has its own set of requirements. Some may require additional PHP packages to be installed. Check your plugin documentation to discover its PHP requirements. If they are available, they can be installed with `apt-get` as demonstrated above.

We will restart Apache to leverage these new extensions in the next section. If you are returning here to install additional plugins, you can restart Apache now by typing:

```
$ sudo systemctl restart apache2
```

Step 3: Adjust Apache's Configuration to Allow for .htaccess Overrides and Rewrites

Next, we will be making a few minor adjustments to our Apache configuration. Currently, the use of `.htaccess` files is disabled. WordPress and many WordPress plugins use these files extensively for in-directory tweaks to the web server's behavior.

Additionally, we will enable `mod_rewrite`, which will be needed in order to get WordPress permalinks to function correctly.

Enable .htaccess Overrides

Oper

```
$
```

To allow `.htaccess` files, we need to set the `AllowOverride` directive within a `Directory` block pointing to our document root. Towards the bottom of the file, add the following blocl

SCROLL TO TOP

/etc/apache2/apache2.conf

```
. . .

<Directory /var/www/html/>
    AllowOverride All
</Directory>

. . .
```

When you are finished, save and close the file.

Enable the Rewrite Module

Next, we can enable `mod_rewrite` so that we can utilize the WordPress permalink feature:

```
$ sudo a2enmod rewrite
```

Enable the Changes

Before we implement the changes we've made, check to make sure we haven't made any syntax errors:

```
$ sudo apache2ctl configtest
```

The output might have a message that looks like this:

```
Output
AH00558: apache2: Could not reliably determine the server's fully qualified domain
Syntax OK
```

If you

/apa d

does

conti

Restart Apache to implement the changes:

SCROLL TO TOP

```
$ sudo systemctl restart apache2
```

Step 4: Download WordPress

Now that our server software is configured, we can download and set up WordPress. For security reasons in particular, it is always recommended to get the latest version of WordPress from their site.

Change into a writable directory and then download the compressed release by typing:

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

Extract the compressed file to create the WordPress directory structure:

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

We will be moving these files into our document root momentarily. Before we do, we can add a dummy `.htaccess` file and set its permissions so that this will be available for WordPress to use later.

Create the file and set the permissions by typing:

```
$ touch /tmp/wordpress/.htaccess
$ chmod 660 /tmp/wordpress/.htaccess
```

We'll also copy over the sample configuration file to the filename that WordPress actually reads:

```
$
```

We c

trying

```
$ mkdir /tmp/wordpress/wp-content/upgrade
```

SCROLL TO TOP

Now, we can copy the entire contents of the directory into our document root. We are using the `-a` flag to make sure our permissions are maintained. We are using a dot at the end of our source directory to indicate that everything within the directory should be copied, including hidden files (like the `.htaccess` file we created):

```
$ sudo cp -a /tmp/wordpress/. /var/www/html
```

Step 5: Configure the WordPress Directory

Before we do the web-based WordPress setup, we need to adjust some items in our WordPress directory.

Adjusting the Ownership and Permissions

One of the big things we need to accomplish is setting up reasonable file permissions and ownership. We need to be able to write to these files as a regular user, and we need the web server to also be able to access and adjust certain files and directories in order to function correctly.

We'll start by assigning ownership over all of the files in our document root to our username. We will use `sammy` as our username in this guide, but you should change this to match whatever your `sudo` user is called. We will assign group ownership to the `www-data` group:

```
$ sudo chown -R sammy:www-data /var/www/html
```

Next, we will set the `setgid` bit on each of the directories within the document root. This causes new files created within these directories to inherit the group of the parent directory (which we just set to `www-data`) instead of the creating user's primary group. This just makes sure that whenever we create a file in the directory on the command line, the web server will still have group ownership over it.

We c

```
$
```

There

`wp-content` directory so that the web interface can make theme and plugin change

SCROLL TO TOP

```
$ sudo chmod g+w /var/www/html/wp-content
```

As part of this process, we will give the web server write access to all of the content in these two directories:

```
$ sudo chmod -R g+w /var/www/html/wp-content/themes
$ sudo chmod -R g+w /var/www/html/wp-content/plugins
```

This should be a reasonable permissions set to start with. Some plugins and procedures might require additional tweaks.

Setting up the WordPress Configuration File

Now, we need to make some changes to the main WordPress configuration file.

When we open the file, our first order of business will be to adjust some secret keys to provide some security for our installation. WordPress provides a secure generator for these values so that you do not have to try to come up with good values on your own. These are only used internally, so it won't hurt usability to have complex, secure values here.

To grab secure values from the WordPress secret key generator, type:

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

You will get back unique values that look something like this:

> **Warning!** It is important that you request unique values each time. Do **NOT** copy the values shown below!

```
Ou
de                                                                      D^<+
de                                                                      v?Px
de                                                                      %3]R
define('NONCE_KEY',        'tt,4UC)/ua+8<:4vM+ DO NOT COPY THESE VALUES # DXT+[$atz
define('AUTH_SALT',        'koMrurzOA+|L_lG}kf DO NOT COPY THESE V                *lD
define('SECURE_AUTH_SALT', 'p32*p,]z%LZ+pAu:VY DO NOT COPY THESE V                K_+
```

SCROLL TO TOP

```
define('LOGGED_IN_SALT',   'i^/G2W7!-1H2OQ+t$3 DO NOT COPY THESE VALUES t6**bRVFSD[
define('NONCE_SALT',       'Q6]U:K?j4L%Z]}h^q7 DO NOT COPY THESE VALUES 1% ^qUswWgn
```

These are configuration lines that we can paste directly in our configuration file to set secure keys. Copy the output you received now.

Now, open the WordPress configuration file:

```
$ nano /var/www/html/wp-config.php
```

Find the section that contains the dummy values for those settings. It will look something like this:

/var/www/html/wp-config.php

```
. . .

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');

. . .
```

Delete those lines and paste in the values you copied from the command line:

/var/www/html/wp-config.php

```
.

de
de
de
de
define('AUTH_SALT',        'VALUES COPIED FROM THE COMMAND LINE');
define('SECURE_AUTH_SALT', 'VALUES COPIED FROM THE COMMAND LINE')
```

SCROLL TO TOP

```
define('LOGGED_IN_SALT',   'VALUES COPIED FROM THE COMMAND LINE');
define('NONCE_SALT',       'VALUES COPIED FROM THE COMMAND LINE');

. . .
```

Next, we need to modify some of the database connection settings at the beginning of the file. You need to adjust the database name, the database user, and the associated password that we configured within MySQL.

The other change we need to make is to set the method that WordPress should use to write to the filesystem. Since we've given the web server permission to write where it needs to, we can explicitly set the filesystem method to "direct". Failure to set this with our current settings would result in WordPress prompting for FTP credentials when we perform some actions.

This setting can be added below the database connection settings, or anywhere else in the file:

/var/www/html/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');
```

Save

Ste ce

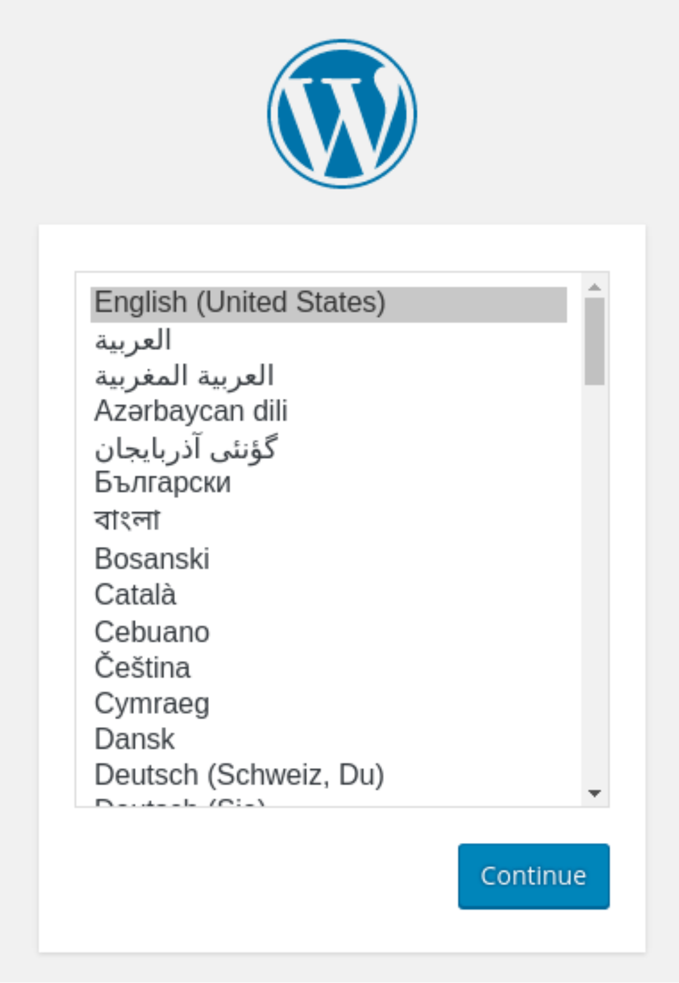
Now ace.

In your web browser, navigate to your server's domain name or public IP address:

SCROLL TO TOP

```
http://server_domain_or_IP
```

Select the language you would like to use:



Next, you will come to the main setup page.

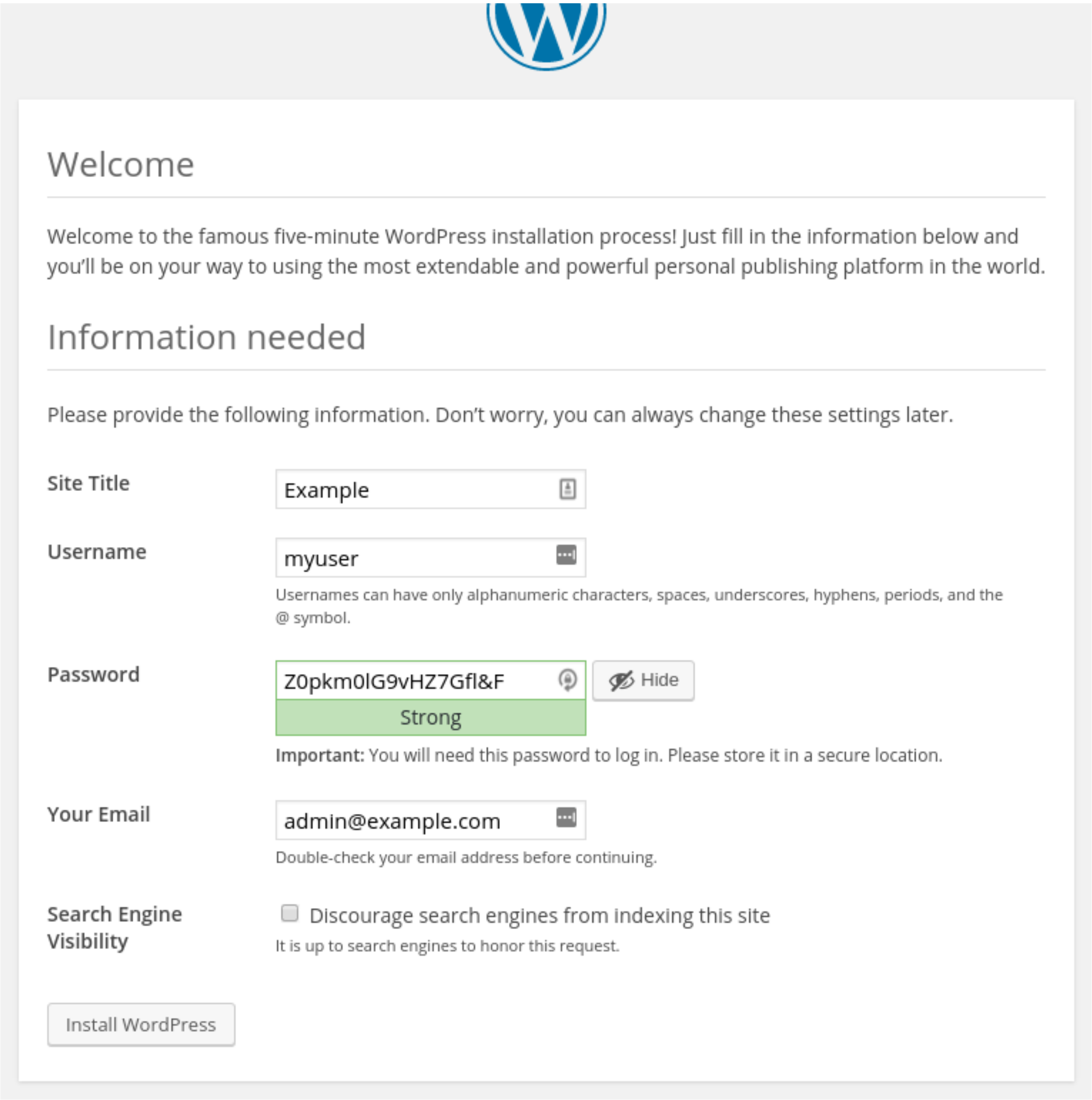
Select a name for your WordPress site and choose a username (it is recommended not to choose

some

passw

Enter ur

site:



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: Example

Username: myuser

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: Z0pkm0lG9vHZ7Gfl&F

Important: You will need this password to log in. Please store it in a secure location.

Your Email: admin@example.com

Double-check your email address before continuing.

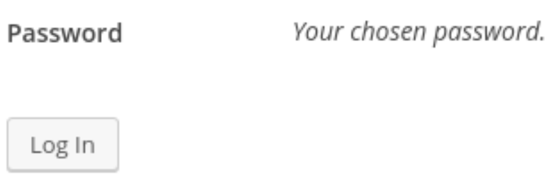
Search Engine Visibility: ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Wher

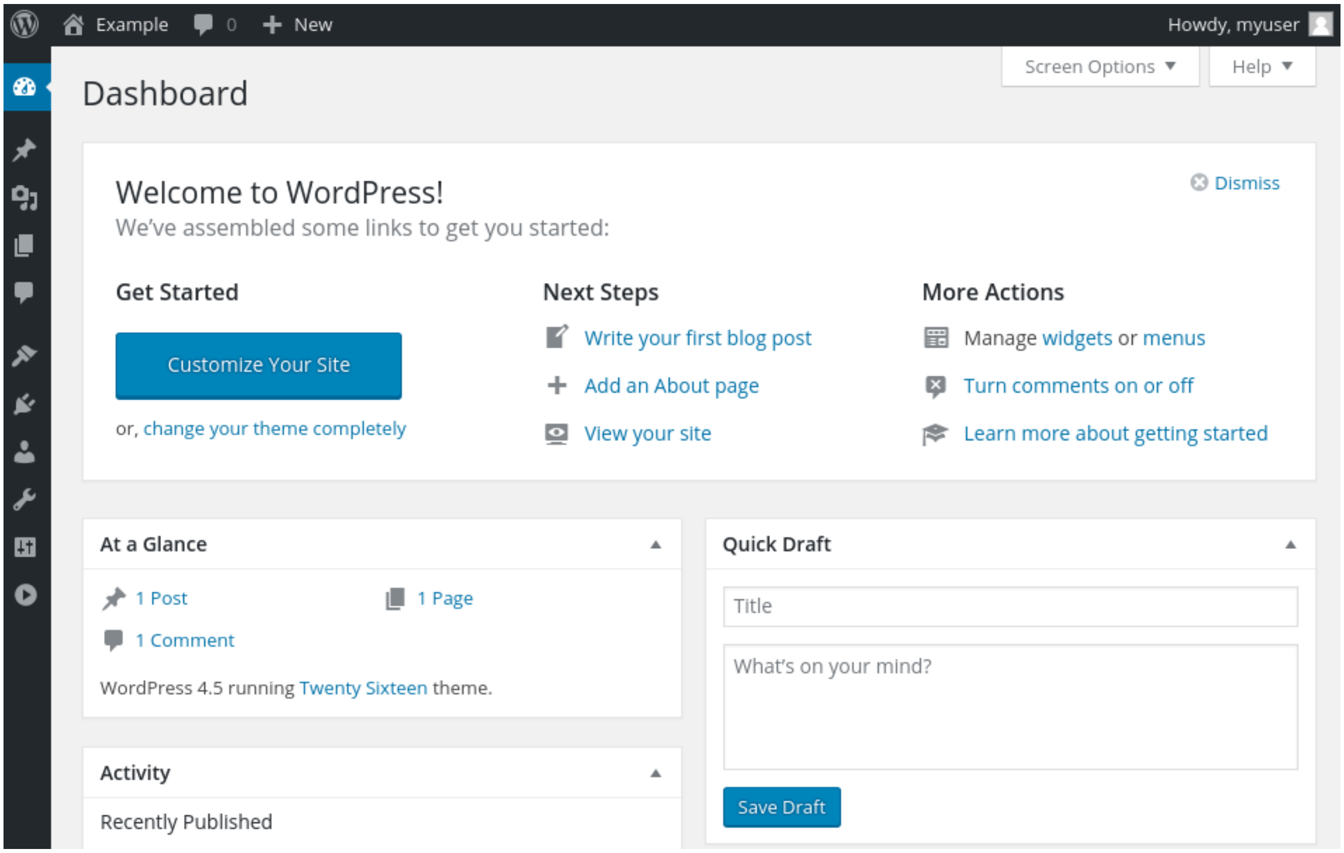
Username myuser

SCROLL TO TOP



Password *Your chosen password.*

Once you log in, you will be taken to the WordPress administration dashboard:



Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

Get Started

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?

Upgrading WordPress

As WordPress upgrades become available, you will be unable in install them through the interface with the current permissions.

The p y for

the 9

autor

When an update becomes available, log back into your server as your `sudo` user. Temporarily give the web server process access to the whole document root:

SCROLL TO TOP

```
$ sudo chown -R www-data /var/www/html
```

Now, go back the WordPress administration panel and apply the update.

When you are finished, lock the permissions down again for security:

```
$ sudo chown -R sammy /var/www/html
```

This should only be necessary when applying upgrades to WordPress itself.

Conclusion

WordPress should be installed and ready to use! Some common next steps are to choose the permalinks setting for your posts (can be found in `Settings > Permalinks`) or to select a new theme (in `Appearance > Themes`). If this is your first time using WordPress, explore the interface a bit to get acquainted with your new CMS.

By: Justin Ellingwood

Upvote (21) Subscribe Share

SCROLL TO TOP

Related Tutorials

How To Set Up Multiple WordPress Sites Using Multisite

How To Install WordPress with LEMP on Ubuntu 16.04

How To Protect WordPress from XML-RPC Attacks on Ubuntu 14.04

How To Install Wordpress and PhpMyAdmin with Docker Compose on Ubuntu 14.04

How To Protect WordPress with Fail2Ban on Ubuntu 14.04

47 Comments

Leave a comment...

Log In to Comment

dordar *May 6, 2016*

2 Is there a way to automate this so that I can run a script with user-init (user data) when I setup a droplet?

ioanmoldovan199 *August 3, 2016*

0

Unpacking the update...

SCROLL TO TOP

The update cannot be installed because we will be unable to copy some files. This is usually due to inconsistent file permissions.: wp-admin/includes/update-core.php

Installation Failed

jellingwood MOD *May 9, 2016*

0 @SummonD: Hey, sorry about that. The tutorial was updated recently to account for that.

Basically, the permissions in this guide offer better security for the majority of your time, between updates. They're a bit too restrictive to use the WordPress update button as-is. For those purposes, you'll need to temporarily change ownership to loosen permissions a bit, update, and then tighten them again.

Since you initially followed this tutorial before it was updated, the first time you do this, you might need to create and adjust the permissions of the `upgrade` directory:

```
$ sudo mkdir -p /var/www/html/wp-content/upgrade
$ sudo chmod g+ws /var/www/html/wp-content/upgrade
$ sudo chown sammy:www-data /var/www/html/wp-content/upgrade
```

Now, each time you upgrade WordPress itself, log into the server and time:

```
$ sudo chown -R www-data /var/www/html
```

In the WordPress interface, apply the pending update.

Once the upgrade is complete, lock down the permissions again for security purposes:

```
$ sudo chown -R sammy /var/www/html
```

That should allow you to apply any updates securely and relatively easily.

may

ay

```
    sudo mkdir -p /var/www/html/wp-content/upgrade
```

SCROLL TO TOP

Results in an error

```
mkdir: cannot create directory ‘/tmp/wordpress/wp-content/upgrade’: File exis
```

If you unpack the .tar file (took a screenshot of the FTP client and Console), you'll see a file is already there named 'upgrade' with no file extension. Users will need to remove the file before they can proceed.

```
sudo rm -r /tmp/wordpress/wp-content/upgrade
mkdir /tmp/wordpress/wp-content/upgrade
```

If they have already installed Wordpress, they'll need to change the path to wherever they installed it.

```
sudo rm -r /var/www/html/upgrade
```

SummonD *May 10, 2016*

0 @jellingwood Now I can update my Wordpress already. Thank you so much.

clreed87 *May 13, 2016*

0 Do SSH updates (secure updates) work on Ubuntu 16? I set them up in a way that should work and the WordPress update connect but I get the error:

```
An error occurred while updating Akismet: Unable to locate WordPress Content di
```

thanks!

1 ' nt

charliec27c2333 *June 26, 2016*

SCROLL TO TOP

This tutorial is great. It installs Wordpress in the web root. If I want to have several instances of WP in the same

Ubuntu install would I just change all the directories to the virtual directories?

0

L0cutus *June 26, 2016*

0 is also OK that wordpress ask me for FTP/SFTP/SSH credential only for delete a plugin ?

kymgard *June 27, 2016*

2 Justin, Thank you for your tutorial, and especially for the clear explanations you provide along the way. I think I am close to having everything set up; but there is a hitch. Having (I think) successfully set everything up, I try to access the Wordpress through the web interface, and I get the Apache 2 Ubuntu Default Page. Any suggestions to what I might have or have not done? Thanks again, Kym

devinsrose *September 28, 2016*

1 That is the same thing that happened to me. Any solution?

taylancorbaci *November 7, 2016*

0 Same here. Any solution?

IvanKutsarovPho *December 8, 2016*

0 Guys, just make sure you deleted default index.html file maded by initial Apache install in /var/www/html/ directory.

tomisin *July 1, 2016*

0 I got this error message, how do I solve it?

SCROLL TO TOP

tomisin *July 1, 2016*

I did this instead, but the last command still gives that error.

0

```
sudo service apache2 restart
```

helgatheviking *July 2, 2016*

0 Any chance that updating the do*user*scripts at github for 16.04 is on the horizon? I'm looking at the 15.10 one (https://github.com/digitalocean/do_user_scripts/blob/master/Ubuntu-15.10/cms/wordpress.sh) and not sure how much needs to be modified.

ryanpq **MOD** *July 5, 2016*

1 The update was pretty basic but is now available here. The changes made include:

- Updated package names from php5 to php7.0
- Modified the password generation section to prevent sporadic errors due to non-alphanumerics

We have also added updated LEMP and LAMP stack scripts for Ubuntu 16.04.

helgatheviking *July 9, 2016*

0 Great, thanks for that! My WordPress is all set up, but that'll be handy for future droplets. Now, onto your postfix tutorials (wouldn't mind seeing that script updated for 16.04 as well ;)

Vardaloupas *July 23, 2016*

0 I prepare my WordPress blog but the logo which is appearing in the top of the front page is broken. I made a check for for permissions in upload folder ant it is 755. I think is not the problem. The theme I use is the "Gadget". Any idea?

1 thing

Anyways, I got all the files on the server and configured everything according to the tu

There is no wordpress setup page when I try to access my IP through web browsers. L a

SCROLL TO TOP

what would cause this?

probalicos *September 19, 2016*

0 Try

http://your-ip-address/wp-login.php

devinsrose *September 28, 2016*

1 I also didn't get the wordpress setup page going to the ip address, only the apache info page. Using /wp-login.php gave me the error page that said:
Error establishing a database connection

Which is at least a wordpress error I recall. Not sure what I have done wrong. Any ideas would be helpful.

yeya *September 24, 2016*

1 Just follow your guid, But can not install any theme or plugin. "Installation failed: Could not create directory."
I chown www-data:www-data and it works,
What did I miss?

cconnoisseur *September 26, 2016*

0 How do you copy and past the secure keys into the wp-config.php file?

brydavismpa *September 27, 2016*

1 If using a one-click LAMP image, be sure to remove `/var/www/html/index.html` file (or change the configuration) because the server defaults to `.html` files before `.php` files.

ivangayton *September 27, 2016*

0 Turns out on MySQL 5.7 the line:

SCROLL TO TOP

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

did not not work for me; this command seems to be deprecated in MySQL. I used:

CREATE USER 'wordpressuser';

Then:

ALTER USER 'wordpressuser'@'localhost' IDENTIFIED BY 'password';

(I know this can be done as a one-liner, but it worked when I did it in two steps and rather than speculating on the exact code of the one-liner I thought I'd post what I'm certain already worked. I'm sure someone far cleverer than I will soon post the one-line solution).

EdgarEstrada *September 29, 2016*

0 I must have made something wrong, Im getting an error 500, or a DNS error for my computer name :(

What do I have to delete to start over again?
Do I have to reinstall all ubuntu?

Thanks!

---

krishnag *October 6, 2016*

0 I am unable to install the WordPress...
To grab secure values from the WordPress secret key generator, type:

Unable to understand what I really need to do after generating secret keys.

---

jellingwood **MOD** *October 6, 2016*

1 @krishnag: You should copy the values you generate on your own machine into the configuration file to replace the lines that have `'put your unique phrase here'` in them. You can also visit the salt generator directly in your web browser if that's easier.

1 I got this error when I try to access the wordpress on a broswer
Forbidden

SCROLL TO TOP

You don't have permission to access / on this server.
Server unable to read htaccess file, denying access to be safe
any idea? I follow all the steps one by one

Load More Comments

SCROLL TO TOP