
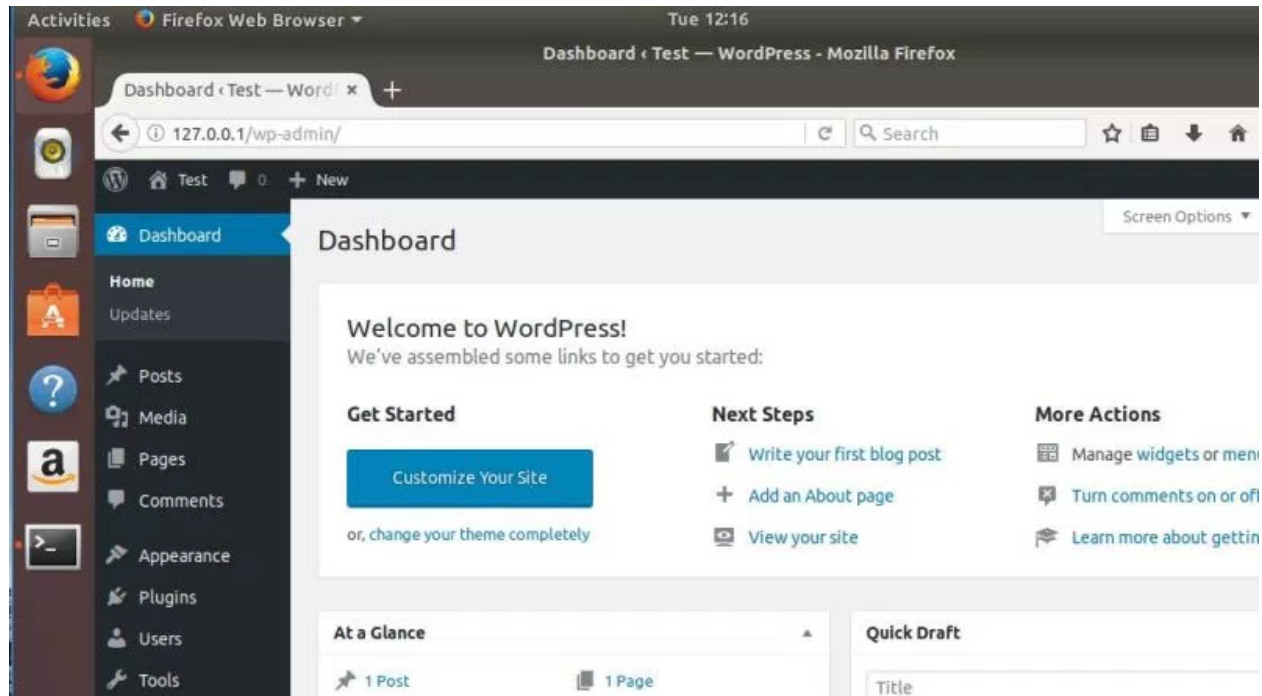


Install And Configure WordPress On Ubuntu 17.10 Artful Aardvark

 [computerbeginnersguides.com/blog/2017/10/31/install-and-configure-wordpress-on-ubuntu-17-10-artful-aardvark](https://www.computerbeginnersguides.com/blog/2017/10/31/install-and-configure-wordpress-on-ubuntu-17-10-artful-aardvark)

Jonathan Moeller

October 31, 2017



WordPress, first developed in 2003, is a content management system that runs quite a variety of websites. In WordPress's early days, it tended to run only personal blogs and smaller sites. However, the software improved considerably with every version, and it's adaptable enough to be used for a variety of different sites. WordPress's big break came in 2009 when popular blogging platform Movable Type made unpopular changes to its licensing terms. Enough Movable Type users were annoyed with the change to migrate to WordPress, and WordPress is the dominant open-source content management system on the Internet as of this writing.

It's quite possible to install WordPress on a machine running either the desktop or the server version of Ubuntu. This makes an excellent environment for testing and experimentation, since you can tweak and experiment to your heart's desire without rendering your production blog inoperable. If you need to run an internal corporate blog of some kind, hosting WordPress on an Ubuntu box or virtual machine is a cheaper solution than more expensive blogging or collaboration software. Or, if you're really ambitious (and you can afford the bandwidth), you can host an Internet-accessible WordPress blog out of your apartment.

Let's begin!

(Note that the Ubuntu repositories now contain a fully automated WordPress installation. However, I believe that installing it by hand yourself has significant value – you'll not only learn how to set up a LAMP server, but installing step-by-step gives you valuable troubleshooting knowledge later on in case something breaks.)

First, install Ubuntu into your machine of choice. Once Ubuntu is installed and updated, you will need to install five pieces of software: the Apache web server, the MySQL database server version 5.7, PHP version 7, the MySQL module for PHP, and finally the WordPress software itself.

First, install the Apache web server. To install Apache, go to a Terminal window or a command prompt and type this command:

sudo apt-get install apache2

(Technically, you'll be installing Apache 2, the latest version.)

Enter your password to authenticate, follow the default prompts, and apt will download and install the Apache web server for you.

And that's it! Apache should now be working. To test it from the web server itself, go to a web browser and visit the address `http://127.0.0.1`. You'll remember that this is the "local loopback" address, basically the address the computer uses to refer to itself. Alternatively, you could test it from another computer on the same subnet. For instance, if you installed Apache on a computer with an IP address of `192.168.1.100`, you could test it by going to another computer on the same subnet and visiting `http://192.168.100` from the web browser.

Regardless, if Apache is working properly, you should see the default Apache/Ubuntu web page with the Ubuntu logo in the upper left-hand corner.

Apache is now operational.

The next step to install WordPress is to install the MySQL database server. Return to the Terminal and type this command:

sudo apt-get install mysql-server-5.7

(As of this writing, MySQL Server version 5.7 is the latest version available in the Ubuntu repositories, though future versions of Ubuntu may receive the newer versions of MySQL.)

Enter your password to authenticate, and apt will download the MySQL files and install them for you. It's a big set of files, so depending on the speed of your Internet connection, it might take a while to download. After the files are downloaded and are installing, the installer will ask you for a password for MySQL's root user. Just as the root user in Linux has complete control over the system, the root user in MySQL has absolute control over all databases, tables, permissions, and users. For obvious security reasons, you'll want to create an extremely strong password (a mixture of uppercase, lowercase, numbers, and punctuation, the longer the better) for your MySQL root user.

(Note that in the password dialog box, if you can't get the <OK> field selected, you can use the tab key to jump from the text input line to the <OK> field.)

After you enter the root password, the installer will finish working with the MySQL files, and return you to the command line.

Next, you'll want to run the `mysql_secure_installation` script to tighten up security on your new MySQL server. Run this command from the prompt:

sudo mysql_secure_installation

First, the `mysql_secure_installation` script will ask you to enter the current password for the MySQL root user. After you do that, it will ask if you want to change the root password. Since you already set a root password, you can hit "n" (unless you want to change it again for some

Next, the script will ask you to setup the VALIDATE PASSWORD plugin, which checks passwords for appropriate strength. Hit Y to accept, and then set the password strength level by hitting 0 for Low, 1, for Medium, and 2 for Strong.

Next, the script will ask if you want to remove the anonymous user. The anonymous user, like anonymous access in FTP, lets someone log into MySQL without having a proper user account. For security reasons, it's always best to remove the anonymous user, so hit "y" to continue.

After that, the script will ask if you want to prevent the MySQL root user from logging in remotely to the MySQL server. Always hit "y" to forbid root remote access, since if an attacker guesses your root password, he can destroy your databases or steal the information they contain.

After this, the script will ask if you want to remove the test database. MySQL includes a test database that anyone can access. Again, this is a security hole, so you'll want to hit "y" to remove the test database.

The script will then ask to reload the privilege tables so the changes take effect. Hit "y", and the mysql_secure_installation script will conclude and return you to the command line.

MySQL server is now installed on your Ubuntu system.

The next step to installing WordPress is to install PHP version 7:

sudo apt-get install php

Then install the MySQL module for PHP:

sudo apt-get install php-mysql

You may also want to install the GD library for PHP, since many WordPress plugins rely on it for graphical manipulation (note that this is an optional step):

sudo apt-get install php-gd

Finally, download the WordPress software to your computer. You can obtain it from this address:

<http://wordpress.org/latest.tar.gz>

If you're working exclusively from the command line, you can use this command to download the WordPress software:

wget <http://wordpress.org/latest.tar.gz>

Use this command to unpack the WordPress files:

tar -xzf latest.tar.gz

Move the WordPress files over to the /var/www/html directory. (In this command, I'll assume you unpacked the WordPress files to your home directory; you will have to adjust the command if you unpacked them in a different directory.)

sudo cp -r ~/wordpress/* /var/www/html/

Now that we've got our software installed and downloaded, we'll need to configure it. First, we'll need to prepare MySQL for use with WordPress. WordPress requires a database and a database user, and full permission for its database user to access its database. To start the MySQL command-line client, use this command:

mysql -u root -p

Enter the password for the MySQL root user, and you'll find yourself at the MySQL command prompt, which will look like this:

mysql>

Our next steps are to create a WordPress database, create a user to access that database, and grant our new user all rights to the WordPress database.

(An important note before we continue, though. All commands made from the MySQL prompt must end with a semicolon to denote the end of the statement. Any commands that do not end with a semicolon will not work. With that in mind, let's first create a database.)

To create a database for WordPress, use this command at the mysql> prompt:

CREATE DATABASE wordpress;

The MySQL client will respond with a message that should say "Query OK, 1 row affect (0.00 sec)." This means the command was successful, and a new database named "newdatabase" has been created.

Next, you'll create a user who will access that database and assign a password to that user. In this example, we'll assign a password of "1234". However, in real life, just as with the root user, you'll want to assign a strong password. (This database user won't have full control over the MySQL user as the root user does, but we will give it full control over the database we just created, and a malicious user who guesses the password could cause all kinds of trouble.):

CREATE USER wordpressuser IDENTIFIED BY '1234';

You should get the "Query OK" message again. The final step is to assign all privileges on the "wordpress" database to the "wordpressuser" user. Use this command to assign the permissions:

GRANT ALL PRIVILEGES ON wordpress.* TO wordpressuser IDENTIFIED BY '1234';

That isn't a typo – those are single quote marks (') instead of the usual double quote marks (") that were used in the command to set the password. MySQL's internal syntax, alas, isn't always consistent. Anyway, if you typed the command correctly, you should get the "Query OK" confirmation again.

Once you are done, use this command to quit the MySQL command-line interface:

exit

Now that we have MySQL prepared, we can now configure WordPress itself. Specifically, we'll have to configure WordPress to talk to the database we just created. To do so, you must create a wp-config.php file in the WordPress directory. Fortunately, WordPress includes a handy wp-config-sample.php you can use as a template.

Type this command to create a wp-config file (assuming you installed WordPress in the /var/www/html directory):

```
sudo cp /var/www/html/wordpress/wp-config-sample.php /var/www/html/wp-config.php
```

Next, use vi to edit the newly created wp-config file:

```
sudo vi /var/www/html/wp-config.php
```

You'll see that the wp-config.php file contains a number of variables. Change the following three variables:

-Change putyourdbnamehere to wordpress .

-Change usernamehere to wordpressuser .

-Change yourpassword here to 1234 .

Once you have your changes made (after double-checking the spelling, of course), switch vi to Command mode, and save your changed file.

Once this is done, rename Apache's default index.html file to index.html.old. Otherwise Apache will access the default sample page instead of starting WordPress:

```
sudo mv index.html index.html.old
```

Everything should now be ready. Open up a web browser, and navigate to http://127.0.0.1/. (If you want to access it from a different machine, of course, use the IP address of the WordPress server, for instance – http://192.168.1.100.) If you configured everything correctly, you should then be greeted by the WordPress configuration page. You'll pick a username, and WordPress will assign you a password. Follow the prompts, and you will have a functional WordPress blog installed on a Ubuntu machine. Congratulations!

One final note: if you configure your blog from a local browser, it will probably set your blog's address as http://127.0.0.1. This is fine if you only want to view it from the host machine, but if you want to view it over the network, you'll need to change it. Fortunately, you can do so quickly by going Options in the WordPress admin interface, and then to General, and changing the blog's URL.

Was this post helpful? These books might be useful:

