



How to Install WordPress with Apache in Ubuntu 20.04

Aaron Kili June 22, 2020 May 22, 2020 Categories [Apache](#), [Ubuntu](#), [WordPress](#) 15 Comments

WordPress is the most popular platform in the world for building websites, be it a blog, an e-commerce website, a business website, a portfolio website, an online business directory, and so on. It is free and open-source, easy to install, learn and use, highly pluggable and customizable too.

Read Also: [How to Install WordPress with Nginx in Ubuntu 20.04](#)

This guide shows how to install the latest version of **WordPress** with **Apache** in **Ubuntu 20.04**. It assumes that you have the **LAMP** stack installed and well configured for hosting websites, otherwise, see our guide:

- [How to Install LAMP Stack with PhpMyAdmin in Ubuntu 20.04](#)

Installing WordPress in Ubuntu 20.04

1. Once the **LAMP** stack (**Apache**, **MariaDB**, and **PHP**) installed and configured on the Ubuntu 20.04 server, you can proceed further to download the latest version of WordPress using the following [wget command](#).

```
$ wget -c http://wordpress.org/latest.tar.gz
```

```

tecmint@tecmint-ubuntu:~$ wget -c http://wordpress.org/latest.tar.gz
--2020-05-22 05:23:40-- http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2020-05-22 05:23:40-- https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12234700 (12M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz          100%[=====] 11.67M  354KB/s   in 28s

2020-05-22 05:24:09 (423 KB/s) - 'latest.tar.gz' saved [12234700/12234700]

tecmint@tecmint-ubuntu:~$ _

```

Download WordPress in Ubuntu 20.04

2. Once the download is complete, extract the archived file using the [tar command](#) as shown.

```
$ tar -xzf latest.tar.gz
```

```

wordpress/wp-admin/includes/class-wp-list-table-compat.php
wordpress/wp-admin/includes/post.php
wordpress/wp-admin/includes/class-core-upgrader.php
wordpress/wp-admin/includes/import.php
wordpress/wp-admin/includes/class-ftp-sockets.php
wordpress/wp-admin/includes/class-wp-automatic-updater.php
wordpress/wp-admin/includes/list-table.php
wordpress/wp-admin/includes/class-wp-filesystem-ssh2.php
wordpress/wp-admin/includes/class-automatic-upgrader-skin.php
wordpress/wp-admin/includes/screen.php
wordpress/wp-admin/includes/class-wp-privacy-data-export-requests-list-table.php
wordpress/wp-admin/includes/translation-install.php
wordpress/wp-admin/includes/admin.php
wordpress/wp-admin/includes/class-wp-privacy-policy-content.php
wordpress/wp-admin/includes/file.php
wordpress/wp-admin/includes/bookmark.php
wordpress/wp-admin/includes/class-wp-upgrader.php
wordpress/wp-admin/includes/class-plugin-upgrader-skin.php
wordpress/wp-admin/includes/ms.php
wordpress/wp-admin/includes/widgets.php
wordpress/wp-admin/includes/class-wp-ms-themes-list-table.php
wordpress/wp-admin/includes/class-wp-filesystem-ftpsockets.php
wordpress/wp-admin/includes/class-wp-theme-install-list-table.php
wordpress/wp-admin/includes/theme.php
wordpress/wp-admin/includes/template.php
wordpress/wp-admin/includes/class-walker-category-checklist.php
wordpress/wp-admin/includes/class-wp-users-list-table.php
wordpress/wp-admin/menu-header.php
wordpress/wp-admin/options-head.php
wordpress/wp-admin/theme-install.php
wordpress/wp-admin/install-helper.php
wordpress/wp-admin/credits.php
wordpress/wp-admin/maint/
wordpress/wp-admin/maint/repair.php
wordpress/wp-admin/ms-upgrade-network.php
wordpress/wp-admin/ms-themes.php
wordpress/wp-admin/post.php

```

Extract WordPress Files

3. Next, move the extracted WordPress directory into your document root i.e. `/var/www/html/` and under your website as shown (replace **mysite.com** with your website's name or domain name). The following command will create a **mysite.com** directory and move WordPress files under it.

```
$ ls -l
```

```
$ sudo cp -R wordpress /var/www/html/mysite.com
```

```
$ ls -l /var/www/html/
```

```
tecmint@tecmint-ubuntu:~$ ls -l
total 11952
-rw-rw-r-- 1 tecmint tecmint 12234700 Apr 29 18:59 latest.tar.gz
drwxr-xr-x 5 tecmint tecmint    4096 Apr 29 18:58 wordpress
tecmint@tecmint-ubuntu:~$ sudo cp -R wordpress/ /var/www/html/mysite.com
tecmint@tecmint-ubuntu:~$ ls -l /var/www/html/
total 16
-rw-r--r-- 1 root root 10918 May 22 05:36 index.html
drwxr-xr-x 5 root root  4096 May 22 05:54 mysite.com
tecmint@tecmint-ubuntu:~$
```

Setup WordPress on Website

4. Now set appropriate permissions on the website (`/var/www/html/mysite.com`) directory. It should be owned by the **Apache2** user and group called **www-data**.

```
$ sudo chown -R www-data:www-data /var/www/html/mysite.com
```

```
$ sudo chmod -R 775 /var/www/html/mysite.com
```

```
tecmint@tecmint-ubuntu:~$ sudo chown -R www-data:www-data /var/www/html/mysite.com/
tecmint@tecmint-ubuntu:~$ sudo chmod -R 775 /var/www/html/mysite.com/
tecmint@tecmint-ubuntu:~$ ls -l /var/www/html/
total 16
-rw-r--r-- 1 root root 10918 May 22 05:36 index.html
drwxrwxr-x 5 www-data www-data 4096 May 22 05:54 mysite.com
tecmint@tecmint-ubuntu:~$
```

Creating a WordPress Database for Website

5. To get started, log into your MariaDB database shell using the following **mysql** command with -u flag to supply the username which should be **root** and -p to enter a password that you set for the MySQL root account when you installed the MariaDB software.

```
$ sudo mysql -u root -p
```

6. After login, run the following commands to create your site's database and a database user with privileges as shown. Remember to replace “**mysite**”, “**mysiteadmin**” and “**SecureP@ssHere!**” with your database name, database username, and the user's password.

```
MariaDB [(none)]> CREATE DATABASE mysite;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON mysite.* TO 'mysiteadmin'@'localhost'  
IDENTIFIED BY 'SecureP@ssHere!';
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> EXIT
```

```
tecmint@tecmint-ubuntu:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 55
Server version: 10.3.22-MariaDB-1ubuntu1 Ubuntu 20.04

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)]> create database mysite;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> Grant all privileges on mysite.* to 'mysiteadmin@localhost' identified by 'tecmin
123';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
tecmint@tecmint-ubuntu:~$ _
```

Create a WordPress Database

7. Next, move into your website's document root, create a **wp-config.php** file from the sample configuration file provided as shown.

```
$ cd /var/www/html/mysite.com
```

```
$ sudo mv wp-config-sample.php wp-config.php
```

8. Then open the **wp-config.php** configuration file for editing.

```
$ sudo vim wp-config.php
```

and update the database connection parameters (database name, database user, and the user's password created above) as shown in the following screenshot.

```

<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'mysite' ); ← Database name

/** MySQL database username */
define( 'DB_USER', 'mysiteadmin' ); ← Database user

/** MySQL database password */
define( 'DB_PASSWORD', 'tecmint123_' ); ← Database user password

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

```

Configure WordPress Database Settings

Creating Apache VirtualHost for WordPress Website

9. Next, you need to configure Apache webserver to serve your **WordPress** site using your fully qualified domain name, by creating a Virtual Host for it under the Apache configuration.

To create and activate a new Virtual Host, create a new file under the **/etc/apache2/sites-available/** directory. In this example, we will call the file **mysite.com.conf** (it should end with the .conf extension).

```
$ sudo vim /etc/apache2/sites-available/mysite.com.conf
```

Then copy and paste the following configuration in it (replacing the **ServerName** and **ServerAdmin**

emails with your values).

```
<VirtualHost *:80>
```

```
ServerName mysite.com
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/html/mysite.com
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
</VirtualHost>
```

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

Create WordPress Apache Virtual Host

Save the file and close it.

10. Then check the Apache configuration for syntax correctness. If the syntax is OK, enable the new site and reload the apache2 service to apply the new changes.

```
$ apache2ctl -t
```

```
$ sudo a2ensite mysite.com.conf
```

```
$ sudo systemctl reload apache2
```

```
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$ sudo a2ensite mysite.com.conf
Enabling site mysite.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$ sudo systemctl reload apache2.service
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$
```

Enable Website in Apache

11. Also, disable the default virtual host to allow your new site to load properly from a web browser.

```
$ sudo a2dissite 000-default.conf
```

```
$ sudo systemctl reload apache2
```

```
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$ sudo systemctl reload apache2.service
tecmint@tecmint-ubuntu:/var/www/html/mysite.com$
```

Disable Website in Apache

Completing the WordPress Installation via Web Interface

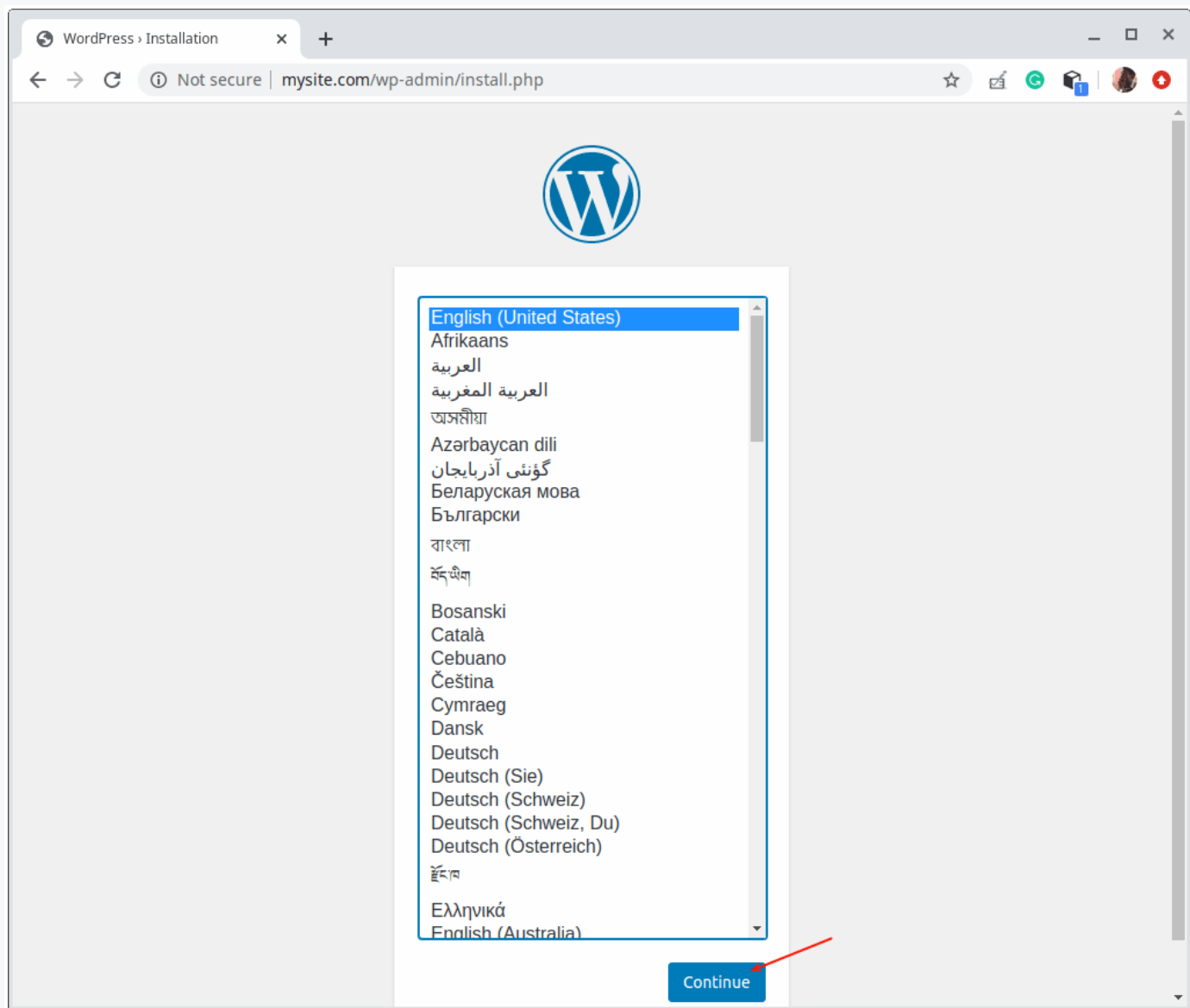
12. The final section demonstrates how to complete the WordPress installation using the web installer.

So open your browser and navigate using your site's domain name:

```
http://mysite.com.
```

Once the WordPress web installer loads, pick the language you wish to use for installation and click

Continue.



Select WordPress Installation Language

13. Next, set your site's title, administrative username, and password and an email for managing your site content. Then click **Install WordPress**.

WordPress › Installation

Not secure | mysite.com/wp-admin/install.php?step=1

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

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

Password [Hide](#)
Strong

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

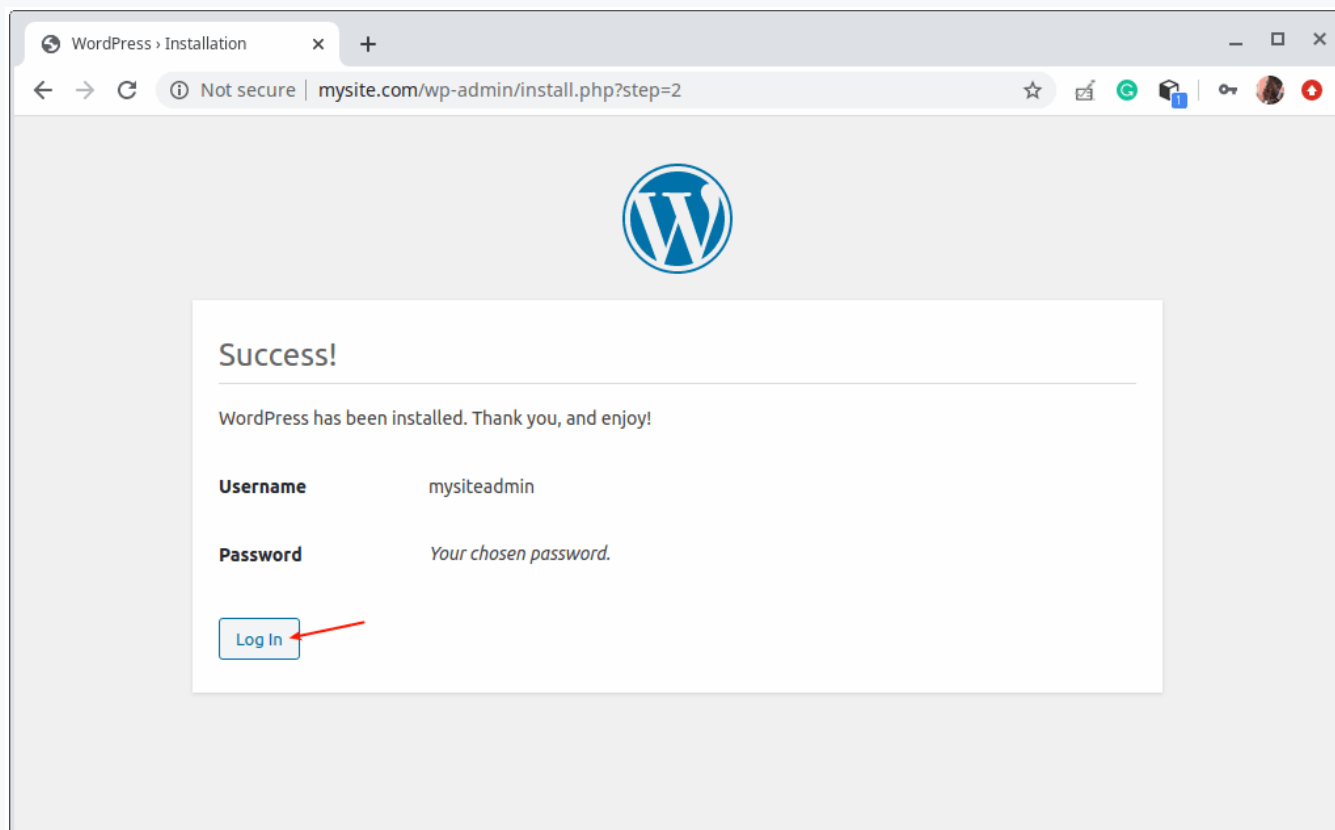
Your Email
Double-check your email address before continuing.

Search Engine Visibility ☐ Discourage search engines from indexing this site
It is up to search engines to honor this request.

[Install WordPress](#)

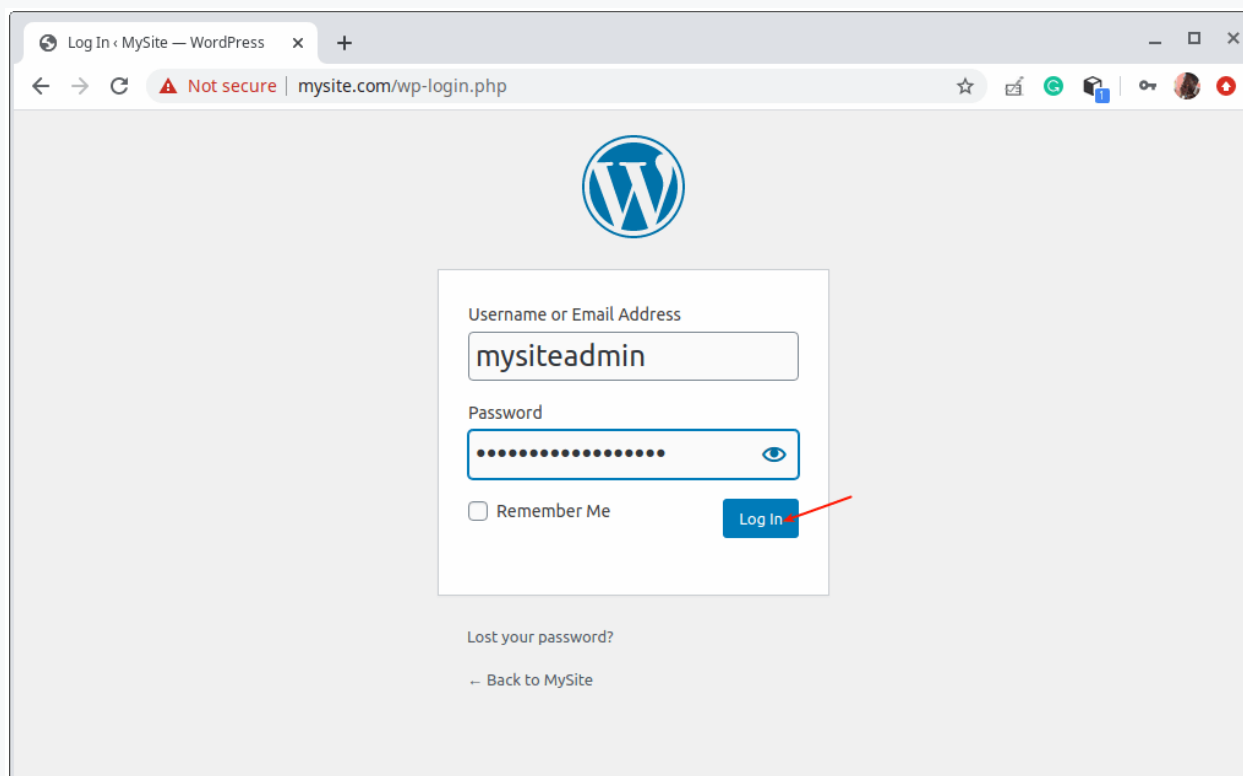
Add WordPress Site Details

14. Once the WordPress installation is complete, click on **Log in** to access your site's administrative login page.

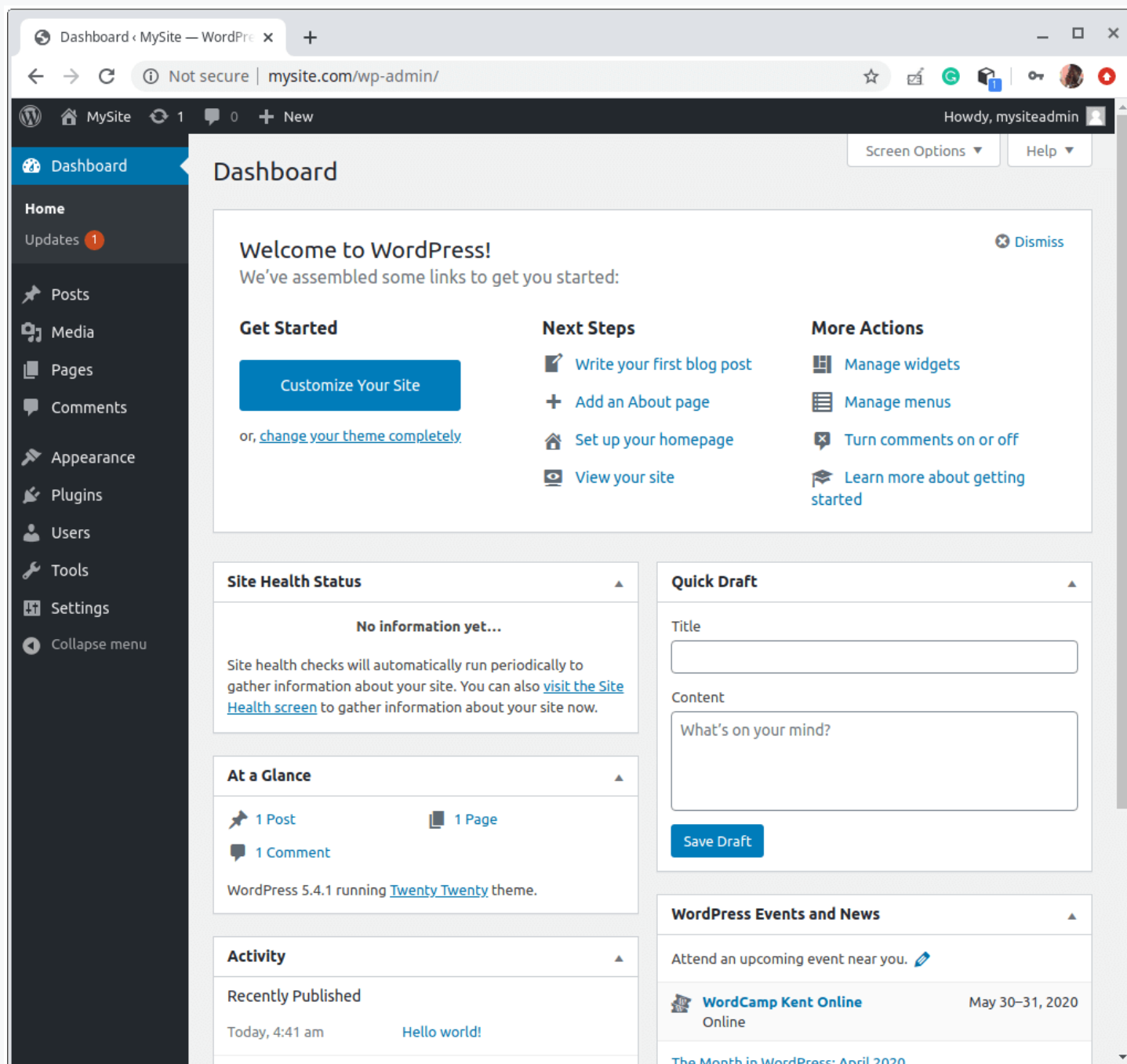


WordPress Installation Complete

15. Now log into your new **WordPress** website using your administrative credentials (username and password created above) and start to customize your site from **Dashboard**.



WordPress Login



WordPress Dashboard

In this article, we have explained how to install **WordPress** using **Apache** as a web server and **MySQL** as a database system for serving PHP websites.

Next, the crucial step is to secure your WordPress site with SSL. If you have deployed WordPress on a real domain, you can secure the site with **Free Let's Encrypt** certificate. If you have deployed WordPress locally on a dummy website for testing or personal use, I suggest you to use a **self-signed certificate** instead.

If you liked this article, then do [subscribe to email alerts](#) for Linux tutorials. If you have any questions or doubts? do [ask for help in the comments](#) section.