# **How to Install WordPress with Apache on Ubuntu 18.04**

****

WordPress is by far the most popular open source blogging and CMS platform that powers over a quarter of the world’s websites. It is based on PHP and MySQL and packs a ton of features that can be extended with free and premium plugins and themes. WordPress enables you to easily build your eCommerce store, website, portfolio or blog.

In this tutorial, we will show you how to install WordPress on an Ubuntu 18.04 machine. It is a fairly straightforward process that takes less than ten minutes to complete. At the time of writing this article, the latest version of **WordPress is version 5.0.2**.

We’ll be using a [LAMP stack](https://linuxize.com/series/how-to-install-lamp-stack-on-ubuntu-18-04/) with Apache as a web server, SSL certificate, the latest **PHP 7.2** and MySQL/MariaDB as a database server.

## **[Prerequisites](https://linuxize.com/post/how-to-install-wordpress-with-apache-on-ubuntu-18-04/#prerequisites)**

Ensure the following prerequisites are met before continuing with this tutorial:

* Logged in as a [user with sudo privileges](https://linuxize.com/post/how-to-create-a-sudo-user-on-ubuntu/).
* Apache installed by following [these instructions](https://linuxize.com/post/how-to-install-apache-on-ubuntu-18-04/).
* You have an SSL certificate installed for your domain. You can install a free Let’s Encrypt SSL certificate by following [these instructions](https://linuxize.com/post/secure-apache-with-let-s-encrypt-on-ubuntu-18-04/).

Update the packages list and upgrade the installed packages to the latest versions:

sudo apt update

sudo apt upgrade

## **[Creating a MySQL database and user account for Wordpress](https://linuxize.com/post/how-to-install-wordpress-with-apache-on-ubuntu-18-04/#creating-a-mysql-database)**

WordPress uses MySQL database to store all its data like posts, pages, users, plugins and themes settings. We’ll start by creating a MySQL database, MySQL user account and grant access to the database.

If you don’t have MySQL or MariaDB installed on your Ubuntu server you can do that by following one of the guides below:

* [Install MySQL on Ubuntu 18.04](https://docs.google.com/document/d/1-sMpMldLtis-27cY5IBGyYuyGtWck9h0IvDoTB4RoAs/edit?usp=sharing)
* Start the database server.

**sudo systemctl start mysql\_server**

* Log in to the database server as the root user. Enter your database root password when prompted; this may be different than your root system password, or it may even be empty if you have not secured your database server.

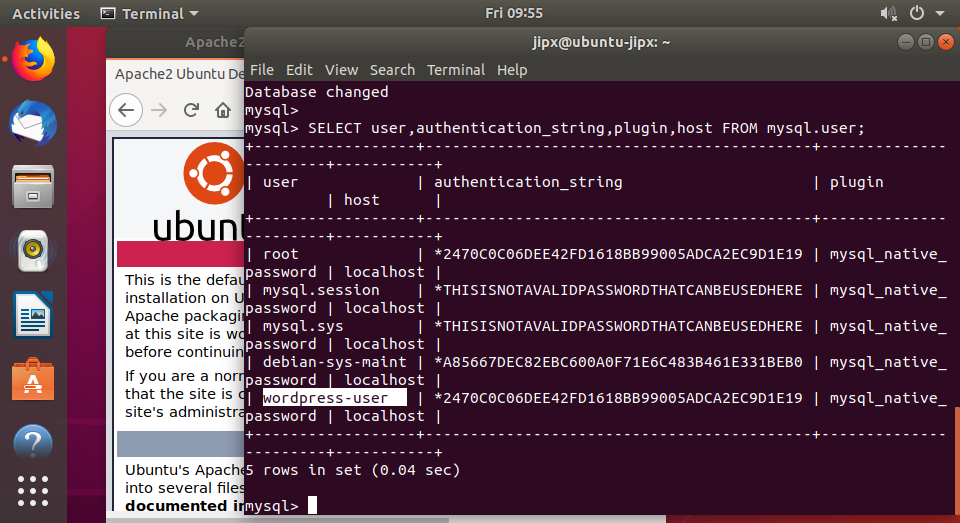
If you have not secured your database server yet, it is important that you do so. For more information, see [To secure the database server](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html#SecuringMySQLProcedure).

**mysql -u root -p**

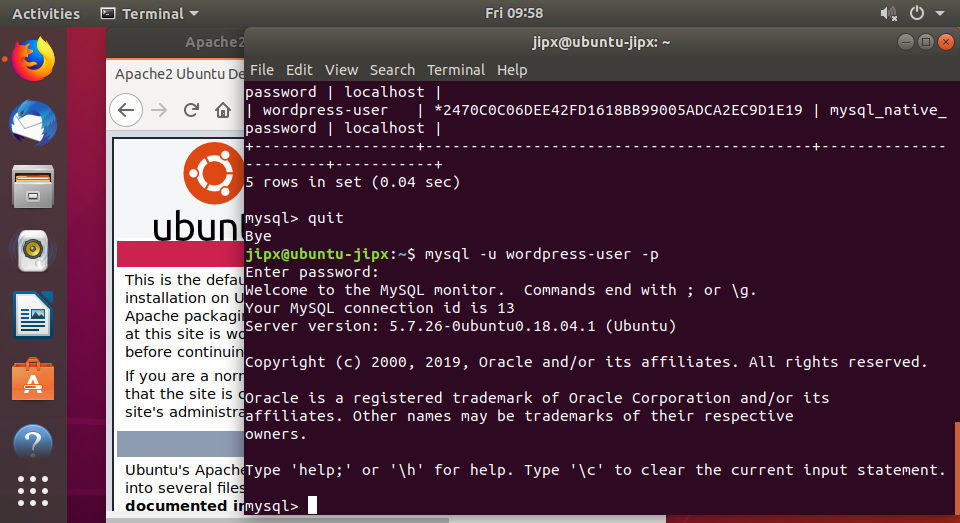
* Create a user ***wordpress-user*** and password for your MySQL database. Your WordPress installation uses these values to communicate with your MySQL database. Enter the following command, substituting a unique user name and password.

**CREATE USER '*wordpress-user*'@'localhost' IDENTIFIED BY '*your\_strong\_password*';**

Make sure that you create a strong password for your user. Do not use the single quote character ( ' ) in your password, because this will break the preceding command. For more information about creating a secure password, go to <http://www.pctools.com/guides/password/>. Do not reuse an existing password, and make sure to store this password in a safe place.

Verify the user “wordpress-user” is created:  


Alternatively you can log in to mysql using “wordpress-user”



* Create your database. Give your database a descriptive, meaningful name, such as wordpress-db.

**Note**

The punctuation marks surrounding the database name in the command below are called backticks. The backtick (`) key is usually located above the Tab key on a standard keyboard. Backticks are not always required, but they allow you to use otherwise illegal characters, such as hyphens, in database names.

**CREATE DATABASE `*wordpress-db*`;**

* Grant full privileges for your database to the WordPress user that you created earlier.

**GRANT ALL PRIVILEGES ON `*wordpress-db*`.\* TO "*wordpress-user*"@"localhost";**

* Flush the database privileges to pick up all of your changes.

**FLUSH PRIVILEGES;**

* Exit the mysql client.

**exit**

Document the above:

|  |  |  |  |
| --- | --- | --- | --- |
| DB\_NAME | DB\_USER | DB\_PASSWORD |  |
|  |  |  |  |

**Install WordPress**

Connect to your instance, and download the WordPress installation package.

**To download and unzip the WordPress installation package**

1. Download the latest WordPress installation package with the **wget** command. The following command should always download the latest release.

$ **wget https://wordpress.org/latest.tar.gz**

1. Unzip and unarchive the installation package. The installation folder is unzipped to a folder called wordpress.

$ **tar -xzf latest.tar.gz**

**To create and edit the wp-config.php file**

The WordPress installation folder contains a sample configuration file called wp-config-sample.php. In this procedure, you copy this file and edit it to fit your specific configuration.

1. Copy the wp-config-sample.php file to a file called wp-config.php. This creates a new configuration file and keeps the original sample file intact as a backup.

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

1. Edit the wp-config.php file with your favorite text editor (such as **nano** or **vim**) and enter values for your installation. If you do not have a favorite text editor, nano is suitable for beginners.

vi **wordpress/wp-config.php**

* 1. Find the line that defines DB\_NAME and change database\_name\_here to the database name that you created in [Step 4](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_database) of [To create a database user and database for your WordPress installation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_user_and_database).

define('DB\_NAME', '*wordpress-db*');

* 1. Find the line that defines DB\_USER and change username\_here to the database user that you created in [Step 3](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_database_user) of [To create a database user and database for your WordPress installation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_user_and_database).

define('DB\_USER', '*wordpress-user*');

* 1. Find the line that defines DB\_PASSWORD and change password\_here to the strong password that you created in [Step 3](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_database_user) of [To create a database user and database for your WordPress installation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html#create_user_and_database).

define('DB\_PASSWORD', '*your\_strong\_password*');

* 1. Find the section called Authentication Unique Keys and Salts. These KEY and SALT values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure. Visit <https://api.wordpress.org/secret-key/1.1/salt/> to randomly generate a set of key values that you can copy and paste into your wp-config.php file. To paste text into a PuTTY terminal, place the cursor where you want to paste the text and right-click your mouse inside the PuTTY terminal.

For more information about security keys, go to <http://codex.wordpress.org/Editing_wp-config.php#Security_Keys>.

**Note**

The values below are for example purposes only; do not use these values for your installation.

\* 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://codex.wordpress.org/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', 'wordpress-db' );

/\*\* MySQL database username \*/

define( 'DB\_USER', 'wordpress-user' );

/\*\* MySQL database password \*/

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

/\*\* MySQL hostname \*/

define( 'DB\_HOST', 'localhost' );

/\*\* Database Charset to use in creating database tables. \*/

define( 'DB\_CHARSET', 'utf8' );

/\*\* The Database Collate type. Don't change this if in doubt. \*/

define( 'DB\_COLLATE', '' );

/\*\*#@+

\* Authentication Unique Keys and Salts.

\*

\* Change these to different unique phrases!

\* You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}

* 1. Save the file and exit your text editor.

|  |
| --- |
| **How to save and quit the vi or vim text editor** To save and quit the vi or vim editor with saving any changes you have made:   1. If you are currently in insert or append mode, press Esc key. 2. Press : (colon). The cursor should reappear at the lower left corner of the screen beside a colon prompt. 3. Enter the following command (type :x and press Enter key): or :wq! |

**To install your WordPress files under the Apache document root**

1. Now that you've unzipped the installation folder, created a MySQL database and user, and customized the WordPress configuration file, you are ready to copy your installation files to your **web server document root** so you can run the installation script that completes your installation. The location of these files depends on whether you want your WordPress blog to be available at the actual root of your web server (for example,*my.public.dns.amazonaws.com*) or in a subdirectory or folder under the root (for example,*my.public.dns.amazonaws.com/blog*).
2. If you want WordPress to run at your document root, copy the contents of the wordpress installation directory (but not the directory itself) as follows:

**cp -r wordpress/\* /var/www/html/**

1. If you want WordPress to run in an alternative directory under the document root, first create that directory, and then copy the files to it. In this example, WordPress will run from the directory blog:

**mkdir /var/www/html/blog**

**cp -r wordpress/\* /var/www/html/blog/**

**Important**

For security purposes, if you are not moving on to the next procedure immediately, stop the Apache web server (apache2) now. After you move your installation under the Apache document root, the WordPress installation script is unprotected and an attacker could gain access to your blog if the Apache web server were running. To stop the Apache web server, enter the command **sudo service apache2 stop**. If you are moving on to the next procedure, you do not need to stop the Apache web server.

**To fix file permissions for the Apache web server**

Some of the available features in WordPress require write access to the Apache document root (such as uploading media though the Administration screens). If you have not already done so, apply the following group memberships and permissions (as described in greater detail in the [LAMP web server tutorial](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html)).

1. Grant file ownership of /var/www and its contents to the www-data user.  
   (Set your web server’s user, www-data, as the owner of your site’s home director)

**sudo chown -R www-data /var/www**

1. Check the owner of the directory /var/www

jipx@ubuntu-jipx:~$ ls -l /var/www

|  |
| --- |
|  |

1. Grant group ownership of /var/www and its contents to the www-data group.

**sudo chgrp -R www-data /var/www**

1. Change the directory permissions of /var/www and its subdirectories to add group write permissions and to set the group ID on future subdirectories.
2. **sudo chmod 2775 /var/www**

**find /var/www -type d -exec sudo chmod 2775 {} \;**

|  |
| --- |
| Command Explained: chmod 2775 /var/www   *2=set group id, 7=rwx for owner , 7=rwx for group , 5=rx for others* |

1. Recursively change the file permissions of /var/www and its subdirectories to add group write permissions.

**find /var/www -type f -exec sudo chmod 0664 {} \;**

|  |
| --- |
| Command Explained: chmod 0664 /var/www   *6=rw- for owner , 7=rw- for group , 5=r-- for others* |

1. Restart the Apache web server to pick up the new group and permissions.

[ **sudo systemctl restart apache2**

**To run the WordPress installation script**

You are ready to install WordPress. The commands that you use depend on the operating system. The commands in this procedure are for use with Amazon Linux 2. Use the procedure that follows this one with Amazon Linux AMI.

1. Use the **chkconfig** command to ensure that the httpd and database services start at every system boot.

**sudo systemctl enable apache2 && sudo systemctl enable mysql**

1. Verify that the database server is running.

**sudo systemctl status mysql**

If the database service is not running, start it.

**sudo systemctl start mysql**

1. Verify that your Apache web server (httpd) is running.

**sudo systemctl status apache2**

If the httpd service is not running, start it.

**sudo systemctl start apache2**

1. In a web browser, type the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the blog folder). You should see the WordPress installation script. Provide the information required by the WordPress installation. Choose **Install WordPress** to complete the installation. For more information, see [Run the Install Script](https://codex.wordpress.org/Installing_WordPress#Step_5:_Run_the_Install_Script) on the WordPress website.

## **[Completing the WordPress Installation](https://linuxize.com/post/how-to-install-wordpress-with-apache-on-ubuntu-18-04/#completing-the-wordpress-installation)**

Now that Wordpress is downloaded and the server configuration is complete, it is time to finalize the WordPress installation through the web interface.

<http://localhost/blog>

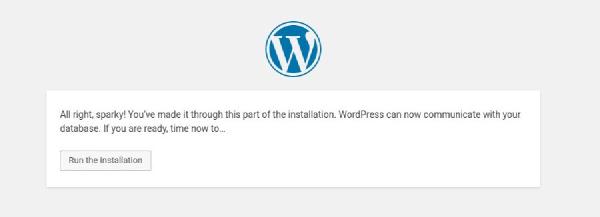
Open your browser, type your domain and a screen similar to the following will appear:



Select the language you would like to use and click on the Continue button.

Next, you will see the following information page, click on the Let's go! button.

Start the installation by clicking on the Run the Installation button.



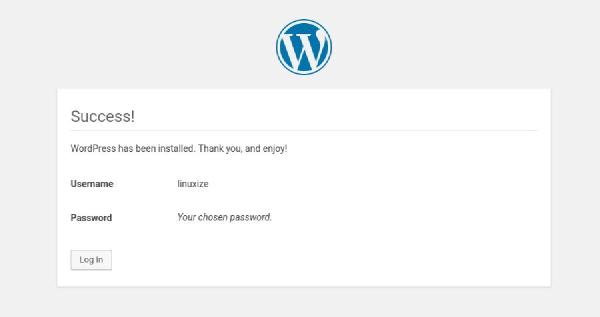
In the next step, you’ll need to enter a name for your WordPress site and choose a username (for security purposes do not enter “admin” ).

The installer will automatically generate a strong password for you. Do not forget to save this password. You can also set the password by yourself.

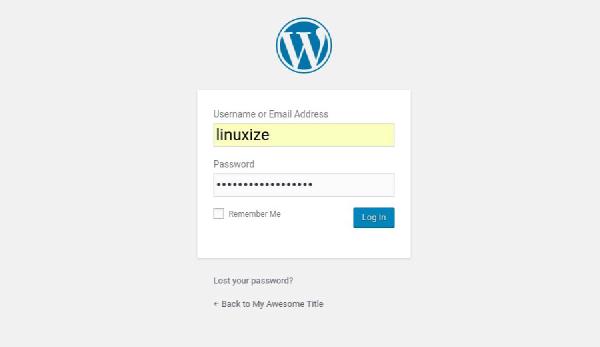
Enter your email address and select whether you want to discourage search engines from indexing the site (not recommended).



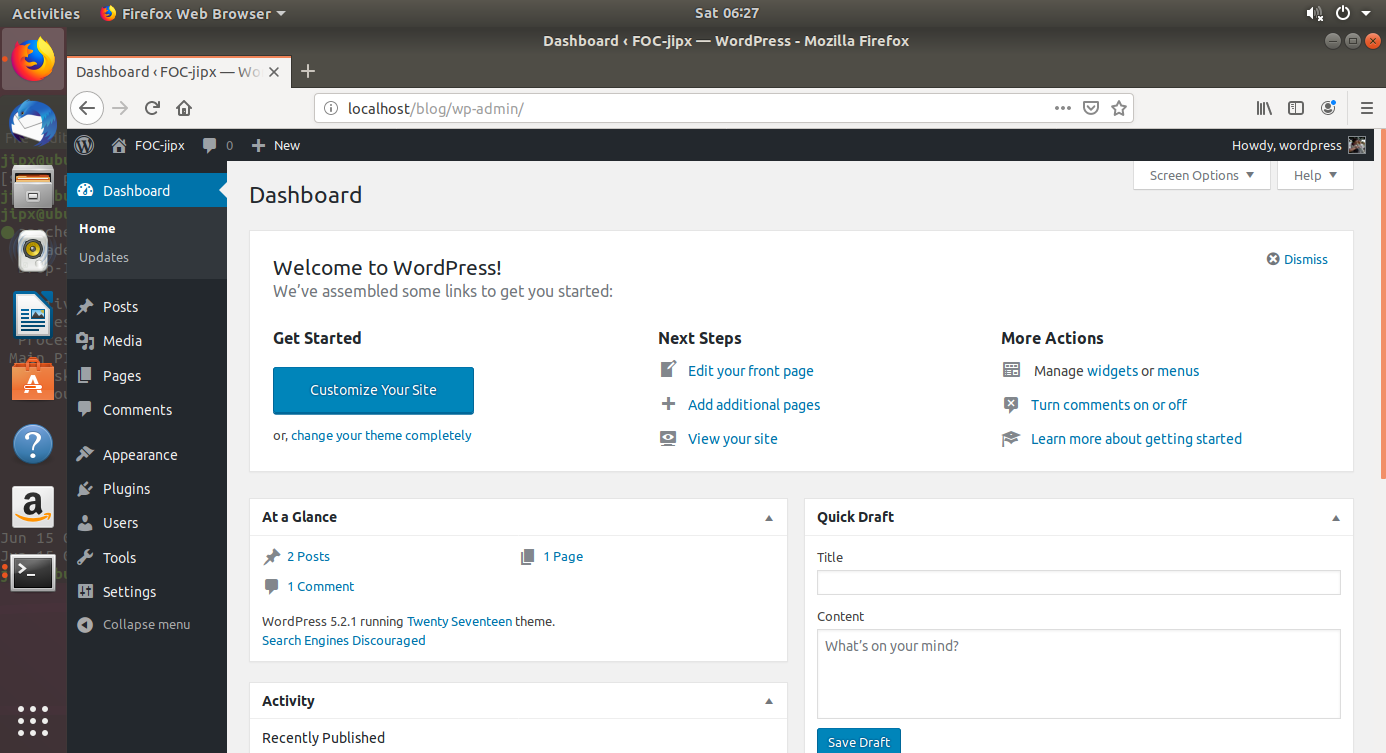
Click Install WordPress and once the installation is completed you will be taken to a page informing you that WordPress has been installed. To access your WordPress login form click on the Log in button.



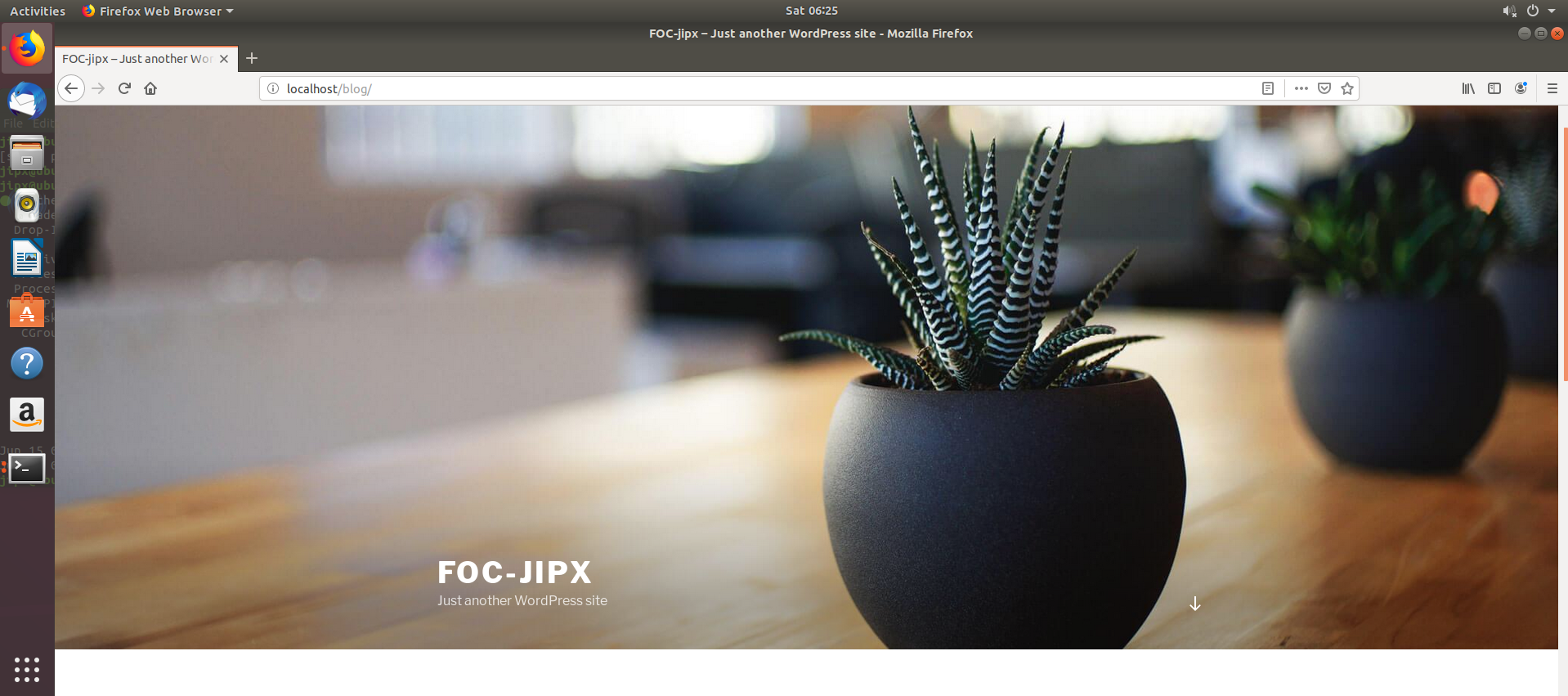
Enter your username and password and click on the Log in button.



Once you log in, you will be redirected to the WordPress administration dashboard.  
Dashboard URI: <http://localhost/blog/wp-admin/>



From here you can start customizing your WordPress installation by installing new themes and plugins.

When your users come to your site, this is what they can see:  
  


**Congratulations**, you have successfully installed WordPress with Apache on your Ubuntu 18.04 server. [First Steps With WordPress](https://codex.wordpress.org/Getting_Started_with_WordPress) is a good starting place to learn more about how to get started with WordPress.

If you have questions feel

Reference:

1. <https://linuxize.com/post/how-to-install-wordpress-with-apache-on-ubuntu-18-04/>
2. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html>
3. [How To Install WordPress with LAMP on Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04)