

# Installing WordPress on Ubuntu 20.04

## **Goal:**

The goal of this project is to install WordPress on Ubuntu 20.04. To complete this project you will install NGINX, PHP, MySQL, and finally WordPress.

## **Instructions:**

This lesson requires that you have access to a system running Ubuntu 20.04.

### **Install the NGINX Web Server**

Before you install any software on an Ubuntu system, you first need to update the local list of available packages and package versions.

```
sudo apt update
```

NOTE: If you are logged in as the root user, you do not need to use sudo. In that case, you can simply run `apt update`.

Now install NGINX.

```
sudo apt install nginx
```

### **Install and Configure the MySQL Database Server**

WordPress needs a place to store its data such as the text of a blog post, the name of the author, and the date it was posted. All of this type of data is stored in a database and we're going to use a MySQL database to store that data.

```
sudo apt install -y mysql-server
```

Use the `mysqladmin` command to create a database named "wordpress".

```
sudo mysqladmin create wordpress
```

Connect to the MySQL server using the `mysql` client. Next, use the `CREATE USER` command to create a database user named "wordpress". Use the `GRANT ALL` command to allow full permissions to the "wordpress" database.

```
sudo mysql
mysql> CREATE USER wordpress@localhost identified by 'wordpress123';
mysql> GRANT ALL on wordpress.* to wordpress@localhost;
mysql> exit
```

## Install PHP

WordPress is written in PHP, so we need to install PHP. Specifically, we're going to install PHP FPM.

```
sudo apt install -y php-fpm
```

WordPress requires several PHP modules to function correctly. They are:

- **MySQL** for connecting to the MySQL database.
- **cURL** for making remote requests.
- **Mbstring** to handle multibyte strings.
- **ImageMagick** to perform actions such as image resizing.
- **XML** to provide XML support.
- **Zip** to unzip plugins, themes, and WordPress update packages.

Install the required PHP modules.

```
sudo apt install -y php-mysql php-curl php-mbstring php-imagick php-xml php-zip
```

## Configure NGINX

We need to tell NGINX to send all PHP requests to PHP FPM for processing. To do this, update the default NGINX configuration.

```
cd /etc/nginx/sites-available/
sudo nano default
```

Change this line from:

```
index index.html index.htm index.nginx-debian.html;
```

to:

```
index index.php index.html index.htm index.nginx-debian.html;
```

Next, change this line from:

```
try_files $uri $uri/ =404;
```

to:

```
try_files $uri $uri/ /index.php$is_args$args;
```

Now change these lines from:

```
#location ~ /\.php$ {  
#     include snippets/fastcgi-php.conf;  
#  
#     # With php-fpm (or other unix sockets):  
#     fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
#     # With php-cgi (or other tcp sockets):  
#     fastcgi_pass 127.0.0.1:9000;  
#}
```

to:

```
location ~ /\.php$ {  
    include snippets/fastcgi-php.conf;  
  
    # With php-fpm (or other unix sockets):  
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
#    # With php-cgi (or other tcp sockets):  
#    fastcgi_pass 127.0.0.1:9000;  
}
```

For reference, here are the contents of the `/etc/nginx/sites-available/default` file with the comments removed.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;

    index index.php index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    }
}
```

Because we made a configuration change to NGINX, we need to tell nginx to reload its configuration.

```
sudo systemctl reload nginx
```

## Download WordPress

Now, download WordPress. You can do that directly from your Linux system by using the `curl` command. Curl is mostly used to transfer data over a network such as downloading a file. The `-O` option causes the file to be saved locally with the same name that was used on the remote system.

```
curl -O https://wordpress.org/wordpress-5.6.tar.gz
```

## Extract WordPress and Move It Into the DocumentRoot

Now that you've downloaded WordPress, it's time to extract its contents and place them into the DocumentRoot. By default, the DocumentRoot is `/var/www/html`, so that's where you'll place the files.

```
tar xvf wordpress-5.6.tar.gz
sudo mv wordpress/* /var/www/html
```

## Assign File Permissions for WordPress

In a later step, you will use the WordPress web installer. You will answer some questions and WordPress will write a configuration file based on those answers. In order to allow it to write to the file, it needs the proper permissions. The web server runs as the "www-data" user, so one simple way to give the web application the proper permissions is to change the ownership of the configuration file to "www-data". Do that with this command:

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

## Determine Your IP Address

Determine the IP address of your server by examining the output of the following command:

```
ip a
```

The first network interface is named "lo" and it's the loopback interface. Since that is an internal-only address, this is not the IP address you're looking for.

Typically, the second network interface listed in the output is the one that has the system's primary IP address assigned to it.

NOTE: If you are using a provider to host your Linux server, you may need to look in the provider's web interface to get the public IP address of your server.

## Complete the Web Application Install

Now that the server configuration is complete, we can complete the installation through the web interface. Open up a web browser on your local machine and type in the IP address of your server.

On the screen that appears select your language and click "Continue."

On the next screen, simply click "Let's Go!"

Next, fill in the database connection details as follows:

Database Name: **wordpress**

Username: **wordpress**

Password: **wordpress123**

Database Host: **localhost**

Table Prefix: **wp\_**

Click the "Submit" button.

On the next screen, click "Run the installation".

Next, fill in the blog details as follows:

Site Title: **My Fun Blog**

Username: **jason**

Password: **jason123**

Confirm Password: Check "Confirm use of weak password"

Your Email: **your\_email@your\_domain.com**

Click the "Install WordPress" button.

You should now be on a screen that says "Success!" Click the "Log In" button.

Log into the WordPress administration dashboard with your credentials.

Username: **jason**

Password: **jason123**

Click "Log In".

## **Congratulations**

You've successfully installed and configured WordPress on Ubuntu!