

# How to Install WordPress on Ubuntu 24.04

Learn how to install WordPress on Ubuntu 24.04 with our step-by-step guide. Set up a LAMP stack, configure MySQL, and get your website running quickly.

---

# Contents

01	Introduction	3
02	Prerequisites	3
03	Install Nginx Web Server	3
04	Configure the Firewall	4
05	Install MySQL Database	5
06	Create a WordPress Database and User	7
07	Install PHP and Extensions	9
08	Download and Install WordPress	10
09	Create an Nginx Server Block for WordPress	12
010	Access the WordPress Dashboard	15
011	Conclusion	18

# Introduction

WordPress is a free, open-source content management system (CMS) built with PHP and backed by MySQL or MariaDB. It began as a blogging tool and has since grown into one of the most widely used platforms for building websites, powering over 22% of the top one million websites as of December 2024.

This article explains how to install and configure WordPress on a server running Ubuntu 24.04 with the LEMP stack (Linux, Nginx, MySQL, PHP). You'll set up the necessary software, secure the environment, and deploy WordPress to serve dynamic websites from your own server.

## Prerequisites

Before you begin, ensure you:

- Have access to an [Ubuntu 24.04 instance](#) as a non-root sudo user.
- [Configure a domain A record](#) pointing to your server IP (e.g., `www.example.com`) if using a custom domain.

## Install Nginx Web Server

1. Install Nginx.

CONSOLE

```
$ sudo apt install nginx -y
```

2. Start the Nginx service.

CONSOLE

```
$ sudo systemctl start nginx
```

3. Enable Nginx to start automatically on boot.

CONSOLE

```
$ sudo systemctl enable nginx
```

4. Check the service status to verify Nginx is running.

CONSOLE

```
$ sudo systemctl status nginx
```

Output.

```
● nginx.service - nginx - high performance web server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled)
   Active: active (running) since Sat 2025-04-26 13:18:09 WAT; 9min ago
     Docs: https://nginx.org/en/docs/
   Main PID: 1629 (nginx)
   ...
```

## Configure the Firewall

Use UFW (Uncomplicated Firewall) to allow HTTP and HTTPS traffic.

1. Check the current firewall status.

CONSOLE

```
$ sudo ufw status
```

2. Allow both HTTP and HTTPS through the firewall using the `Nginx Full` profile.

CONSOLE

```
$ sudo ufw allow "Nginx Full"
```

3. Reload UFW to apply the rule.

CONSOLE

```
$ sudo ufw reload
```

4. Verify that the rule has been applied.

CONSOLE

```
$ sudo ufw status
```

Output:

To	Action	From
--	-----	----
Nginx Full	ALLOW	Anywhere
Nginx Full (v6)	ALLOW	Anywhere (v6)

## Install MySQL Database

1. Install the MySQL server package.

CONSOLE

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

2. Verify that MySQL was installed successfully.

CONSOLE

```
$ mysql --version
```

Output:

```
mysql Ver 8.0.36 for Linux on x86_64 (MySQL Community Server - GPL)
```

3. Start the MySQL service to activate the database server.

CONSOLE

```
$ sudo systemctl start mysql
```

4. Enable MySQL to start on boot.

CONSOLE

```
$ sudo systemctl enable mysql
```

5. Verify that MySQL is running correctly by checking its service status.

CONSOLE

```
$ sudo systemctl status mysql
```

Output:

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled)
   Active: active (running) since Sat 2025-04-26 13:42:56 WAT; 2min ago
     Main PID: 43442 (mysqld)
   ...
```

6. Harden your MySQL setup by running the built-in security script.

CONSOLE

```
$ sudo mysql_secure_installation
```

# Create a WordPress Database and User

WordPress requires a dedicated MySQL database to store its content, settings, and user information. Follow these steps to set up a database and user:

1. Log in to the MySQL shell as the root user.

CONSOLE

```
$ sudo mysql
```

2. Create the WordPress database.

SQL

```
mysql> CREATE DATABASE wordpressdb;
```

3. Create a dedicated user for WordPress with a strong password.

SQL

```
mysql> CREATE USER 'wordpressdbuser'@'localhost' IDENTIFIED BY 'strongpassword';
```

4. Grant full privileges to the new user on the WordPress database.

SQL

```
mysql> GRANT ALL PRIVILEGES ON wordpressdb.* TO 'wordpressdbuser'@'localhost';
```

5. Apply the changes by reloading the privileges.

SQL

```
mysql> FLUSH PRIVILEGES;
```

6. Verify that the database was created successfully.

SQL

```
mysql> SHOW DATABASES;
```

Output:

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
| wordpressdb       |
+-----+
```

7. Check that the user was created.

SQL

```
mysql> SELECT User, Host FROM mysql.user WHERE User =
'wordpressdbuser';
```

Output:

```
+-----+-----+
| User          | Host    |
+-----+-----+
| wordpressdbuser | localhost |
+-----+-----+
```

8. Confirm the privileges granted to the user.

SQL

```
mysql> SHOW GRANTS FOR 'wordpressdbuser'@'localhost';
```

Output:

```
+-----+
| Grants for wordpressdbuser@localhost          |
+-----+
| GRANT USAGE ON *.* TO `wordpressdbuser`@`localhost` |
| GRANT ALL PRIVILEGES ON `wordpressdb`.`*` TO `wordpressdbuser`@`localhost` |
+-----+
```

9. Exit the MySQL shell.

```
SQL
```

```
mysql> EXIT;
```

## Install PHP and Extensions

WordPress uses PHP to process dynamic content and connect to the MySQL database. Install PHP and the required extensions:

1. Install PHP and all required modules.

```
CONSOLE
```

```
$ sudo apt install php php-cli php-common php-imap php-fpm
php-snmp php-xml php-zip php-mbstring php-curl php-mysqli
php-gd php-intl -y
```

2. Verify the installed PHP version.

```
CONSOLE
```

```
$ php -v
```

Output:

```
PHP 8.3.X (cli) (built: ...)
```

3. Confirm that `php-fpm` is running.

```
CONSOLE
```

```
$ sudo systemctl status php8.3-fpm
```

The output should show `active (running)`.

## Download and Install WordPress

1. Download the latest WordPress release.

```
CONSOLE
```

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

2. Extract the archive.

```
CONSOLE
```

```
$ sudo tar -xvzf latest.tar.gz
```

3. Move the contents to the Nginx web root.

```
CONSOLE
```

```
$ sudo mv wordpress/* /var/www/html/
```

4. Set ownership to the Nginx user.

```
CONSOLE
```

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

5. Change to the web root.

```
CONSOLE
```

```
$ cd /var/www/html/
```

6. Remove default Nginx placeholders.

```
CONSOLE
```

```
$ sudo rm index.html index.nginx-debian.html
```

7. Rename the sample config.

```
CONSOLE
```

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

8. Verify the WordPress files are in place.

```
CONSOLE
```

```
$ ls -l
```

Ensure files like `wp-config.php`, `wp-login.php`, and the `wp-admin` folder exist.

## Configure the WordPress `wp-config.php` File

To allow WordPress to connect to your MySQL database, update the `wp-config.php` file with the database credentials created earlier.

1. Open the configuration file in a text editor.

```
CONSOLE
```

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

2. Find and update the following lines with your database information:

## PHP

```
// The name of the database for WordPress
define( 'DB_NAME', 'wordpressdb' );

// MySQL database username
define( 'DB_USER', 'wordpressdbuser' );

// MySQL database password
define( 'DB_PASSWORD', 'strongpassword' );

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

Save and close the file.

- Ensure the values for `DB_NAME`, `DB_USER`, and `DB_PASSWORD` match what you used during the **Create a WordPress Database and User** step.
- Leave `DB_HOST` as `localhost` if the database is hosted on the same server.

## Create an Nginx Server Block for WordPress

To serve your WordPress site, configure Nginx with a server block that handles PHP requests and routes domain traffic to your WordPress directory.

1. Identify the active PHP-FPM socket on your system.

## CONSOLE

```
$ ls /var/run/php
```

Output:

```
php8.3-fpm.pid  php8.3-fpm.sock  php-fpm.sock
```

Note the exact `.sock` filename, such as `php8.3-fpm.sock`. You'll need this for the `fastcgi_pass` directive in the next step.

2. Open the default Nginx server block configuration file.

## CONSOLE

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

3. Replace the contents of the file with the following configuration. Replace `www.example.com` with your domain name or server IP, and update the PHP socket path if needed.

## INI

```
server {
    listen 80;
    server_name www.example.com;

    root /var/www/html;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?q=$uri&$args;
    }

    location ~ /\.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.(php|php5|php7|php8|php9|html|htm))(/.+)$;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock; #
adjust if your PHP version differs
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location = /favicon.ico {
        access_log off;
        log_not_found off;
        expires max;
    }

    location = /robots.txt {
        access_log off;
        log_not_found off;
    }
}
```

```
error_page 404 /404.html;

error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/html;
}

location ~ /\. {
    deny all;
    access_log off;
    log_not_found off;
}

location /wp-content/uploads/ {
    location ~ \.php$ {
        deny all;
    }
}
}
```

#### 4. Test the Nginx configuration for syntax correctness.

CONSOLE

```
$ sudo nginx -t
```

Output:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

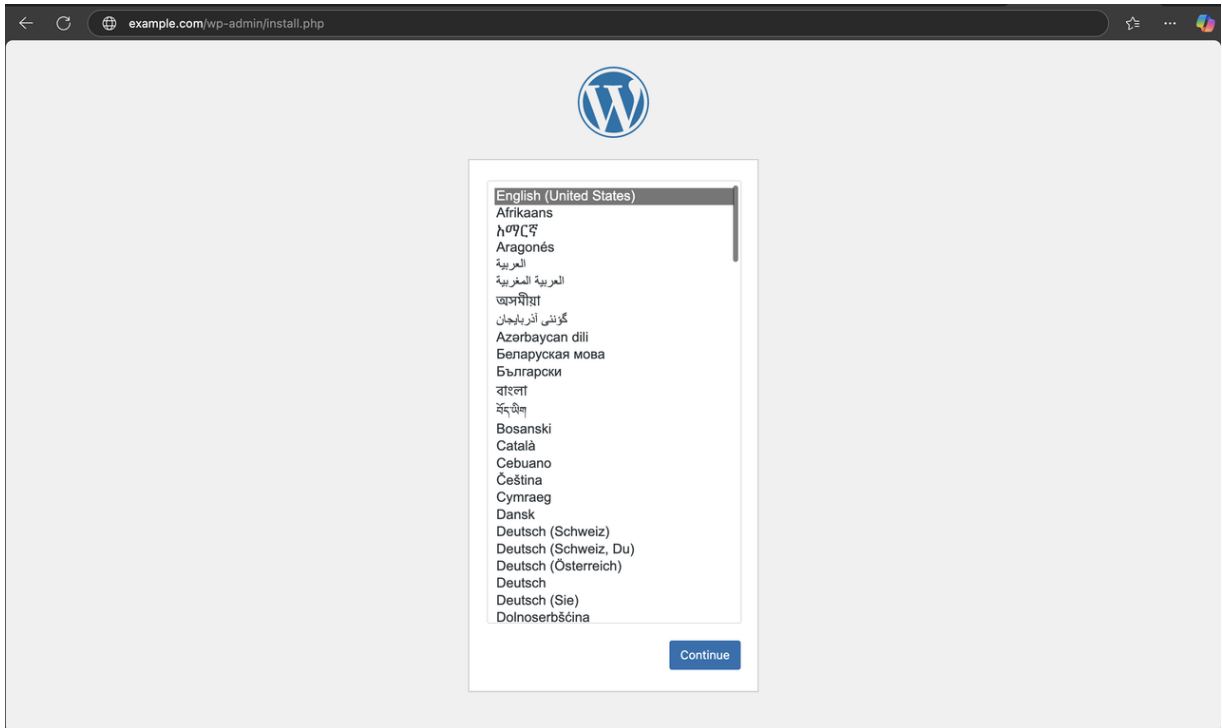
#### 5. Restart Nginx to apply the updated configuration.

CONSOLE

```
$ sudo systemctl restart nginx
```

# Access the WordPress Dashboard

1. Open your web browser and visit `http://www.example.com`. You should see the WordPress installation screen.



Select your preferred language and click **Continue**.

2. Fill in the required details, including your **Site Title**, **Username**, **Password**, and **Email**, then click **Install WordPress**.

## 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.

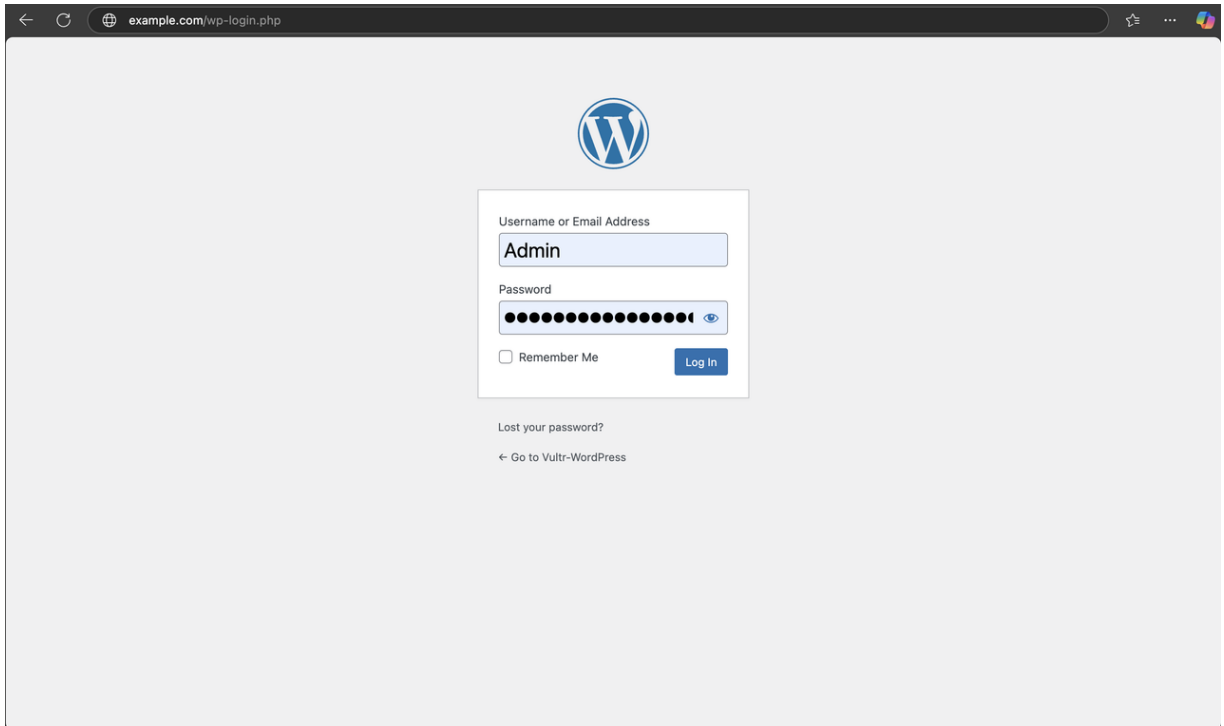
**Password**    
Strong

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

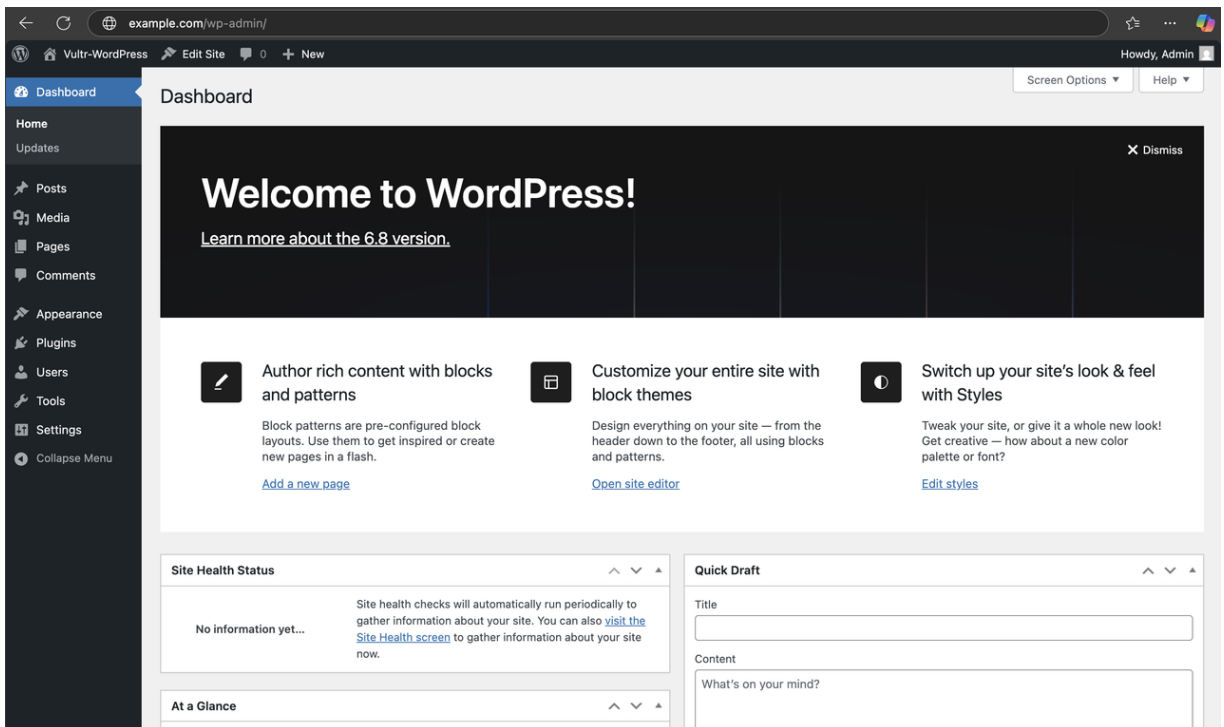
**Your Email**   
Double-check your email address before continuing.

**Search engine visibility**  Discourage search engines from indexing this site  
It is up to search engines to honor this request.

3. Once installation completes, log in with the credentials you just set.



4. You'll be redirected to the WordPress dashboard, where you can begin customizing your website.



## Conclusion

---

In this article, you installed and configured WordPress on Ubuntu 24.04 using the LEMP stack (Linux, Nginx, MySQL, PHP). You set up Nginx to serve your site, configured a secure MySQL database, tuned PHP with necessary extensions, and deployed WordPress for production use.



VULTR

