

# Wordpress paso a paso con docker



**Index**

Instalación ubuntu con todo todito.....3

    Configuración wordpress.....3

    .....4

    Preparación de mariadb.....5

Acceder al wordpress.....7

# Instalación ubuntu con todo todito

Comando para pillar el contenedor y ejecutarlo directo:

```
docker run -it -p 8080:80 --name LAMP ubuntu:22.04 /bin/bash
```

En caso de que salgamos y nos haga falta volver el siguiente comando:

```
docker start -ai LAMP
```

Ahora actualizamos el sistema con un:

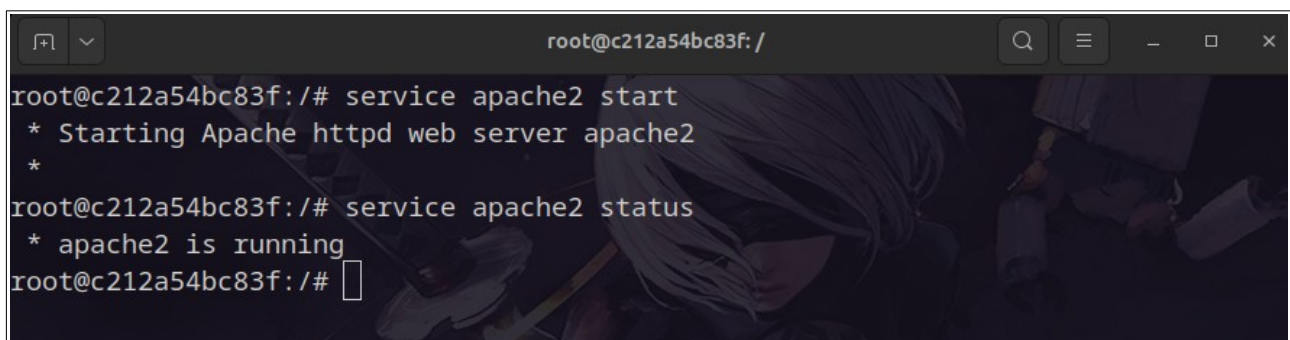
```
apt update && apt upgrade
```

Ahorita con el siguiente comando instalamos php, wordpress, y cosillas para la conexión con la base de datos mariadb:

```
apt install wordpress php libapache2-mod-php mariadb-server php-mysql
```

Ahorita iniciamos apache2:

```
service apache2 start
```

A terminal window with a dark background and a subtle image of a person. The prompt is 'root@c212a54bc83f: /'. The user enters 'service apache2 start', and the output is '\* Starting Apache httpd web server apache2 \*'. Then the user enters 'service apache2 status', and the output is '\* apache2 is running'. The prompt returns to 'root@c212a54bc83f: /#'.

```
root@c212a54bc83f: /  
root@c212a54bc83f:/# service apache2 start  
* Starting Apache httpd web server apache2  
*  
root@c212a54bc83f:/# service apache2 status  
* apache2 is running  
root@c212a54bc83f:/#
```

Acá instalamos un editor para crear un archivo de configuración, en mi caso instalaré nano:

```
apt install nano
```

## Configuración wordpress

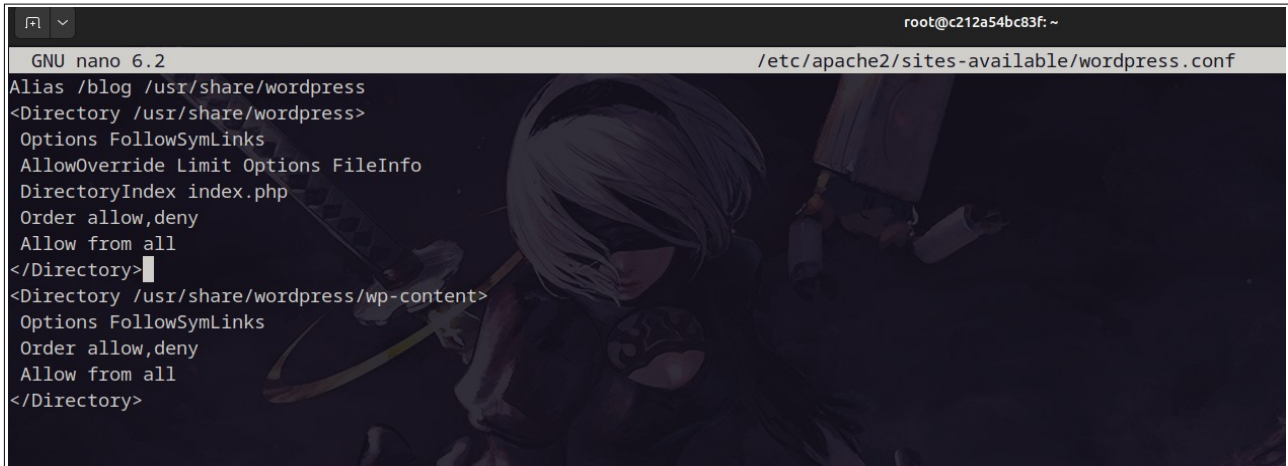
Y ahora creamos un archivo conf para la configuración de nuestra web en sites-available:

```
nano /etc/apache2/sites-available/wordpress.conf
```

Content:

```
Alias /blog /usr/share/wordpress  
<Directory /usr/share/wordpress>  
Options FollowSymLinks  
AllowOverride Limit Options FileInfo  
DirectoryIndex index.php  
Order allow,deny  
Allow from all
```

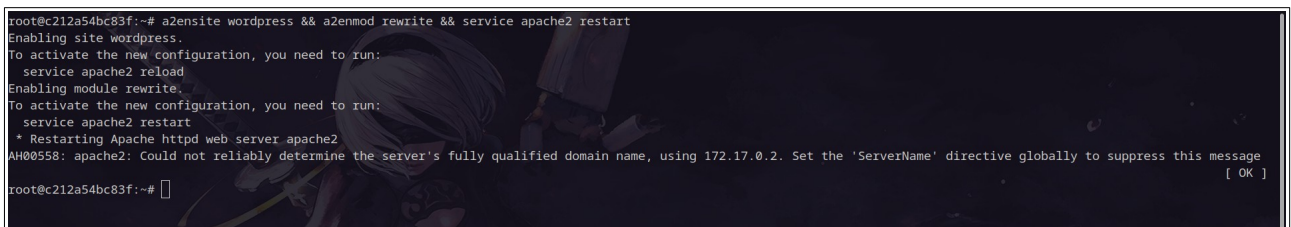
```
</Directory>
<Directory /usr/share/wordpress/wp-content>
Options FollowSymLinks
Order allow,deny
Allow from all
</Directory>
```



```
root@c212a54bc83f: ~
GNU nano 6.2 /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
Options FollowSymLinks
AllowOverride Limit Options FileInfo
DirectoryIndex index.php
Order allow,deny
Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
Options FollowSymLinks
Order allow,deny
Allow from all
</Directory>
```

Y ahorita con apache haremos en una línea unos comandos para recargar sites-avaliable con el wordpress que acabamos de configurar en el wordpress.conf:

a2ensite wordpress && a2enmod rewrite && service apache2 restart



```
root@c212a54bc83f:~# a2ensite wordpress && a2enmod rewrite && service apache2 restart
Enabling site wordpress.
To activate the new configuration, you need to run:
  service apache2 reload
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
* Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[ OK ]
root@c212a54bc83f:~#
```

## Preparación de mariadb

Iniciamos el servicio de mariadb

service mariadb start

```
root@c212a54bc83f:~# a2ensite wordpress && a2enmod rewrite && service apache2 restart
Enabling site wordpress.
To activate the new configuration, you need to run:
    service apache2 reload
Enabling module rewrite.
To activate the new config, you need to run:
    service apache2 restart
* Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[ OK ]

root@c212a54bc83f:~# service mariadb start
* Starting MariaDB database server mariadb
root@c212a54bc83f:~# service mariadb status
[ OK ]
/usr/bin/mysqldadmin Ver 10.0 Distrib 10.6.18-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version      10.6.18-MariaDB-0ubuntu0.22.04.1
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /run/mysqld/mysqld.sock
Uptime:             4 sec

Threads: 1 Questions: 61 Slow queries: 0 Opens: 33 Open tables: 26 Queries per second avg: 15.250
root@c212a54bc83f:~#
```

Una vez iniciado, vamos a preparar la contraseña del server:

mysql\_secure\_installation

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@c212a54bc83f:~#
```

Y ya con la contraseña de root creada, nos metemos dentro del root de mariadb:

mysql -u root -p

```
root@c212a54bc83f:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]> █
```

Y ya dentro, vamos a crear la base de datos de wordpress:

CREATE DATABASE wordpress;

Ahorita dentro, creamos un usuario con contraseña para que wordpress pueda acceder a la base de datos y le damos full permisos a la base de datos de wordpress:

CREATE USER 'wordpress'@'%' IDENTIFIED BY 'MiPass-2025';

GRANT ALL PRIVILEGES ON wordpress.\* TO 'wordpress'@'%' WITH GRANT OPTION;

FLUSH PRIVILEGES;

Y ya para acabar, para el server de wordpress vamos a meterle el usuario, la contraseña y todo de la base de datos para que pueda acceder.

Creamos un archivo en:

nano /etc/wordpress/config-localhost.php

Y le metemos el siguiente contenido:

```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'MiPass-2025');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```



# Acceder al wordpress

Y ahora accedemos a <http://localhost:8080/blog/wp-admin/install.php> y aquí metemos las cosillas que hemos configurado y tiramos como de normal:

The image shows two screenshots of a web browser. The top screenshot displays the WordPress installation 'Welcome' screen. It includes a form for 'Information needed' with fields for Site Title, Username, Password (with a strength indicator), Your Email, and a checkbox for Search engine visibility. An 'Install WordPress' button is at the bottom. The bottom screenshot shows the WordPress Dashboard. It features a 'Welcome to WordPress!' message, a 'Get Started' section with a 'Customize Your Site' button, and a 'Next Steps' list. The dashboard also includes a sidebar menu with options like Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main content area contains widgets for 'Site Health Status', 'Quick Draft', 'At a Glance', 'WordPress Events and News', and 'Activity'.