

# Instalación de WordPress

AWS

Yeray Gutiérrez Mullor

## Contenido

Wordpress: .....	2
Instalar nginx: .....	2
Instalar php y mysql: .....	2
Crear archivo de configuración en Nginx: .....	2
Modificar el archivo de configuración:.....	2
1. Escucha de tráfico (Puerto 80) .....	4
2. Directorio raíz y archivos índice.....	4
3. Manejo de URLs (Enlaces permanentes).....	4
4. Procesamiento de PHP (El motor).....	4
Instalar WordPress: .....	5
Resultado: .....	6

## Wordpress:

Para crear nuestro sitio web, en este proyecto debemos hacerlo utilizando WordPress, para ello, debemos instalarlo de la siguiente manera:

## Instalar nginx:

Instalar servidor web nginx para configurar el acceso web

```
ubuntu@ip-10-0-128-249:~$ sudo apt install nginx[]
```

## Instalar php y mysql:

Instalar estas herramientas para su posterior configuración y funcionamiento

```
ubuntu@ip-10-0-128-249:/etc/nginx/sites-available$ apt install php-fpm php-mysql[]
```

## Crear archivo de configuración en Nginx:

Crear el archivo de configuración del sitio web del WordPress

```
ubuntu@ip-10-0-128-249:~$ cd /etc/nginx/sites-available/[]
```

Crear una copia del archivo por defecto

```
ubuntu@ip-10-0-128-249:/etc/nginx/sites-available$ cp default wordpress[]
```

## Modificar el archivo de configuración:

```
ubuntu@ip-10-0-128-249:/etc/nginx/sites-available$ sudo nano wordpress []
```

```
GNU nano 7.2                               wordpress
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.php index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ /index.php?$args;
    }

    # pass PHP scripts to FastCGI server
    #
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;

        # With php-fpm (or other unix sockets):
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        # With php-cgi (or other tcp sockets):
        #fastcgi_pass 127.0.0.1:9000;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny all;
    }
}
```

## 1. Escucha de tráfico (Puerto 80)

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

Configura el servidor para escuchar peticiones en el puerto 80

default\_server significa que este es el sitio que responderá si alguien entra a la IP del servidor sin un dominio específico configurado.

## 2. Directorio raíz y archivos índice

```
root /var/www/html;  
  
# Add index.php to the list if you are using PHP  
index index.php index.html index.htm index.nginx-debian.html;
```

root: Define que los archivos del WordPress están físicamente en la carpeta /var/www/html.

index: Le dice a Nginx que, si alguien entra a la raíz, busque primero el archivo index.php.

## 3. Manejo de URLs (Enlaces permanentes)

```
server_name _;  
  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    try_files $uri $uri/ /index.php?$args;  
}
```

Esta es la pieza clave para WordPress. Intenta buscar el archivo solicitado (como una imagen o un CSS).

Si el archivo no existe, envía la petición al index.php con todos los argumentos. Sin esto, se verían errores 404 al navegar por el sitio

## 4. Procesamiento de PHP (El motor)

```
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
  
    # With php-fpm (or other unix sockets):  
    fastcgi_pass unix:/run/php/php8.0-fpm.sock;  
    # With php-cgi (or other tcp sockets):  
    #fastcgi_pass 127.0.0.1:9000;  
}
```

Nginx por sí solo no sabe leer PHP. Cuando ve un archivo .php, lo envía a un programa externo llamado PHP-FPM mediante un "socket" y PHP procesa la base de datos y genera el HTML que el usuario finalmente ve.

## Instalar WordPress:

Para instala WordPress hay que descargar el archivo comprimido con todo lo necesario utilizando este comando:

```
● ubuntu@ip-10-0-128-249:~$ sudo wget https://es.wordpress.org/latest-es_ES.zip
++-2026-02-05 17:58:52-- https://es.wordpress.org/latest-es_ES.zip
Resolving es.wordpress.org (es.wordpress.org).... 198.143.164.252, 2607:f978:5:8002::c68f:a4fc
Connecting to es.wordpress.org (es.wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 36112982 (34M) [application/zip]
Saving to: 'latest-es_ES.zip.1'

latest-es_ES.zip.1      100%[=====] 34.44M  96.9MB/s   in 0.4s

2026-02-05 17:58:53 (96.9 MB/s) - 'latest-es_ES.zip.1' saved [36112982/36112982]
```

Lo descomprimimos con este otro comando:

```
● ubuntu@ip-10-0-128-249:~$ sudo unzip latest-es_ES.zip []
```

```
● ubuntu@ip-10-0-128-249:/var/www/html$ ls
index.php          wordpress        wp-config.php    wp-login.php
info.php           wp-activate.php  wp-content      wp-mail.php
license.txt        wp-admin         wp-cron.php    wp-settings.php
phpMyAdmin-5.2.1-all-languages.zip  wp-blog-header.php  wp-includes  wp-signup.php
phpmyadmin        wp-comments-post.php  wp-links-opml.php  wp-trackback.php
readme.html        wp-config-sample.php  wp-load.php    xmlrpc.php
● ubuntu@ip-10-0-128-249:/var/www/html$ []
```

Configuramos este archivo:

```
● ubuntu@ip-10-0-128-249:/var/www/html$ sudo nano wp-config.php[]
```

En el apartado “DB\_HOST” ponemos elEndPoint del Amazon RDS:

```
GNU nano 7.2                                     wp-config.php
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 website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

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

/** Database username */
define( 'DB_USER', 'root' );

/** Database password */
define( 'DB_PASSWORD', 'usuario2' );

/** Database hostname */
define( 'DB_HOST', 'fatgram.cqg5ve3796m7.us-east-1.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );</pre
```

## Resultado:

Si accedemos desde el navegador con la IP del proxy configurado anteriormente ya tendremos la interfaz web del WordPress

