

# Instalación de WordPress en Ubuntu 24.04

La instalación utiliza el *stack* **LAMP** (Linux, Apache, MySQL y PHP) en un entorno local, ideal para desarrollo y pruebas.

## Paso 1: Actualizar el Sistema

Es crucial asegurar que todos los paquetes estén actualizados para garantizar la estabilidad y seguridad.

Comando	Función
<code>sudo apt update</code>	Actualiza la lista de paquetes disponibles
<code>sudo apt upgrade -y</code>	Instala las actualizaciones pendientes

## Paso 2: Instalar Apache (Servidor Web)

Apache es el servidor que se encargará de mostrar tu sitio web.

### 1. Instalar Apache:

```
sudo apt install apache2 -y
```

### 2. Verificar la instalación:

Accede a `http://localhost/` en tu navegador. Deberías ver la página de bienvenida de Apache

## Paso 3: Instalar MySQL (Base de Datos)

MySQL se utiliza para almacenar todo el contenido de WordPress.

### 1. Instalar MySQL:

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

### 2. Ejecutar el script de seguridad:

Se recomienda ejecutar este *script* para mejorar la seguridad de MySQL y establecer opciones básicas.

```
sudo mysql_secure_installation
```

Sigue las instrucciones en pantalla para establecer la contraseña del usuario root y otras opciones de seguridad.

```
jose@jose-VirtualBox:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: █
```

```
There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
        file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: █
```

12345678

Remove anonymous users? (Press y|Y for Yes, any other key for No) : **y**

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : **n**

Con esa respuesta le has indicado a la configuración de SQL que **no quieres deshabilitar el inicio de sesión remoto para el usuario root**.

Al presionar **n**, le indicas al sistema de gestión de bases de datos que **permitirá** que el usuario root se conecte a la base de datos **desde cualquier dirección IP** de la red, no solo desde la máquina local (localhost).

### Implicaciones de Seguridad

La configuración por defecto y la **recomendación de seguridad** es siempre responder **y (Sí)** a esta pregunta.

Configuración	Implicación de Seguridad
<code>Permitir root remoto</code>	<b>Riesgo ALTO.</b> Si alguien adivina o descifra la contraseña de root, puede acceder a toda la base de datos desde cualquier lugar de la red, sin necesidad de acceder primero al servidor.
<code>Deshabilitar root remoto</code>	<b>Riesgo BAJO.</b> El usuario root solo puede conectarse desde el servidor (localhost). Para acceder como root, el atacante necesitaría primero obtener acceso al servidor mismo, lo que añade una capa de defensa.

Si estás en un entorno de máquina virtual aislada y necesitas acceder como root desde otra máquina virtual o desde tu máquina anfitriona, esta opción te lo permite. Sin embargo, en un entorno de producción, esta configuración es peligrosa.

Si el proceso de `mysql_secure_installation` no te pidió ingresar una nueva contraseña para el usuario root de MySQL, a pesar de que es lo que el *script* suele hacer:

Esto es un comportamiento **esperado y normal** en las instalaciones modernas de MySQL en Ubuntu. La razón principal es la siguiente:

### El Método de Autenticación por Socket (`auth_socket`)

La instalación predeterminada de MySQL en sistemas como Ubuntu 24.04 utiliza un mecanismo de seguridad llamado **autenticación por socket de Unix** (a través del *plugin* `auth_socket` o similar).

1. **¿Qué es?** En lugar de usar una contraseña, este método permite que el usuario del sistema operativo (feval, cuando usa `sudo`) inicie sesión automáticamente como el usuario root de MySQL, **siempre y cuando se conecte desde la misma máquina.**
2. **La Omisión de Contraseña:** Como el acceso a root de MySQL ahora está vinculado al acceso al root del sistema operativo (al usar `sudo`), el *script* de instalación segura **omite el paso de establecer la contraseña** porque ya no es el método principal de autenticación.
3. **El Mensaje Clave:** El propio *script* te lo indicó con el mensaje:

**Skipping password set for root as authentication with auth\_socket is used by default.**

En resumen, no es un error. Simplemente, **el acceso de root de MySQL está protegido por tu contraseña de sudo y no por una contraseña separada de MySQL.**

## **Paso 4: Crear la Base de Datos y el Usuario de WordPress**

Debes crear una base de datos específica y un usuario dedicado para WordPress.

### **1. Acceder al *shell* de MySQL:**

```
sudo mysql
```

### **2. Ejecutar los comandos SQL (reemplaza **12345678** con una contraseña fuerte):**

SQL

```
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;  
CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY '12345678';  
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

*Estos comandos crean la base de datos wordpress y el usuario **wordpressuser**, otorgándole todos los privilegios sobre esa base de datos.*

## **Paso 5: Instalar PHP y Extensiones Necesarias**

PHP es el lenguaje de programación que utiliza WordPress, y requiere extensiones específicas para manejar diversas funcionalidades (como imágenes y datos XML).

### **1. Instalar PHP y extensiones:**

```
sudo apt install php libapache2-mod-php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip -y
```

## Paso 6: Configurar Apache para WordPress

Necesitas configurar Apache para que sepa dónde encontrar y cómo servir el sitio de WordPress.

### 1. Crear el archivo de configuración del sitio:

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

### 2. Añadir la configuración (copia y pega):

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/wordpress

    <Directory /var/www/wordpress/>

        AllowOverride All

        Require all granted

    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

*Esta configuración establece /var/www/wordpress como el directorio raíz y permite el uso de archivos .htaccess (gracias a AllowOverride All) para las configuraciones de WordPress, como las URLs amigables.*

3. **Guardar y cerrar** el archivo (en nano: Ctrl+O, Enter, Ctrl+X).
4. **Habilitar el sitio y el módulo rewrite** (necesario para las URLs amigables):

```
sudo a2ensite wordpress.conf
sudo a2enmod rewrite
```

```
jose@jose-VirtualBox:~$ sudo a2ensite wordpress.conf
Enabling site wordpress.
To activate the new configuration, you need to run:
    systemctl reload apache2
jose@jose-VirtualBox:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    systemctl restart apache2
jose@jose-VirtualBox:~$
```

##### 5. Recargar Apache para aplicar los cambios:

```
sudo systemctl reload apache2
```

La diferencia principal entre **systemctl restart apache2** y **systemctl reload apache2** es el método que usan para aplicar los cambios en la configuración de Apache y el impacto que tienen sobre las conexiones activas:

Característica	<b>systemctl reload apache2 (Recargar)</b>	<b>systemctl restart apache2 (Reiniciar)</b>
Mecanismo	Envía una señal al proceso principal de Apache para que relea su archivo de configuración.	Detiene completamente el servicio y luego lo vuelve a iniciar.
Tiempo de Inactividad	<b>Casi nulo (0 segundos).</b> Las conexiones activas terminan sus tareas con el proceso antiguo.	<b>Breve interrupción.</b> Hay un momento de inactividad mientras el servicio se detiene y se inicia.
Impacto en Conexiones	<b>Mantiene las conexiones activas.</b> Las conexiones se sirven hasta que terminan, luego se usan los nuevos <i>workers</i> .	<b>Corta todas las conexiones activas.</b> Las peticiones en curso se pierden.

Característica	<code>systemctl reload apache2</code> (Recargar)	<code>systemctl restart apache2</code> (Reiniciar)
Cuándo usarlo	Para <b>cambios menores</b> en la configuración (ej. agregar un Virtual Host, cambiar una directiva de registro).	Para <b>cambios mayores</b> (ej. cambiar puertos, cambiar módulos, errores críticos) o cuando una recarga falla.

### `systemctl reload apache2` (Recargar)

Esta es la forma **preferida y más segura** de aplicar cambios en la configuración en un entorno de producción, ya que minimiza el impacto en los usuarios finales.

Simplemente le dice a Apache que:

1. Cargue la nueva configuración.
2. Inicie nuevos procesos *worker* con esa configuración.
3. Permita que los procesos *worker* antiguos terminen de servir sus peticiones pendientes antes de cerrarse.

### `systemctl restart apache2` (Reiniciar)

Esto es más drástico. Es necesario cuando la configuración es tan crítica (como cambiar un módulo, un puerto principal o la configuración SSL/TLS) que **Apache necesita una detención completa** para aplicar los cambios de manera segura. Si una recarga falla o los cambios no se aplican, un reinicio forzado suele ser la solución.

## Paso 7: Descargar y Configurar WordPress

Descarga los archivos de WordPress y configúralos para que se conecten a la base de datos.

### 1. Descargar y extraer:

```
wget https://wordpress.org/latest.tar.gz
tar -xvzf latest.tar.gz
sudo mv wordpress /var/www/
```

2. **Asignar permisos:** Esto permite a Apache (www-data) acceder y modificar los archivos.

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

3. **Configurar la base de datos:**

```
cd /var/www/wordpress
sudo cp wp-config-sample.php wp-config.php
sudo nano wp-config.php
```

4. **Editar el archivo wp-config.php** y reemplazar los valores de la base de datos con los que creaste en el Paso 4.

El bloque modificado completo en tu wp-config.php debe quedar **exactamente** así:

```
define( 'DB_NAME', 'wordpress' );

define( 'DB_USER', 'wordpressuser' );

define( 'DB_PASSWORD', '12345678' );

define( 'DB_HOST', 'localhost' );

define( 'DB_CHARSET', 'utf8' );

define( 'DB_COLLATE', '' );
```

```
// ** 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', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', '12345678' );

/** Database 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', '' );
```



## 5. Guardar y cerrar el archivo.

```
* @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', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', '56781812' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

Si recargamos localhost en el navegador seguiremos viendo la página de Apache.

La razón por la que sigues viendo la página por defecto de Apache ("It works!") es que tu servidor está cargando la configuración predeterminada (000-default.conf) antes que la de WordPress, ya que ambas están escuchando en el puerto 80.

Para que WordPress se muestre al acceder a http://localhost/, debemos **deshabilitar el sitio web predeterminado** de Apache.

## Solución: Deshabilitar el Sitio por Defecto

### 1. Deshabilitar el sitio predeterminado

Utiliza la herramienta a2dissite para deshabilitar el archivo de configuración por defecto (000-default.conf).

```
sudo a2dissite 000-default.conf
```

### 2. Verificar que WordPress esté habilitado (Solo si es necesario)

Puedes confirmarlo con a2ensite:

```
sudo a2ensite wordpress.conf
```

(Si ya está habilitado, te dirá: "Site wordpress already enabled").

### 3. Recargar el Servicio Apache

Ahora, con el sitio predeterminado deshabilitado y el sitio de WordPress habilitado, recarga el servicio para que los cambios surtan efecto.

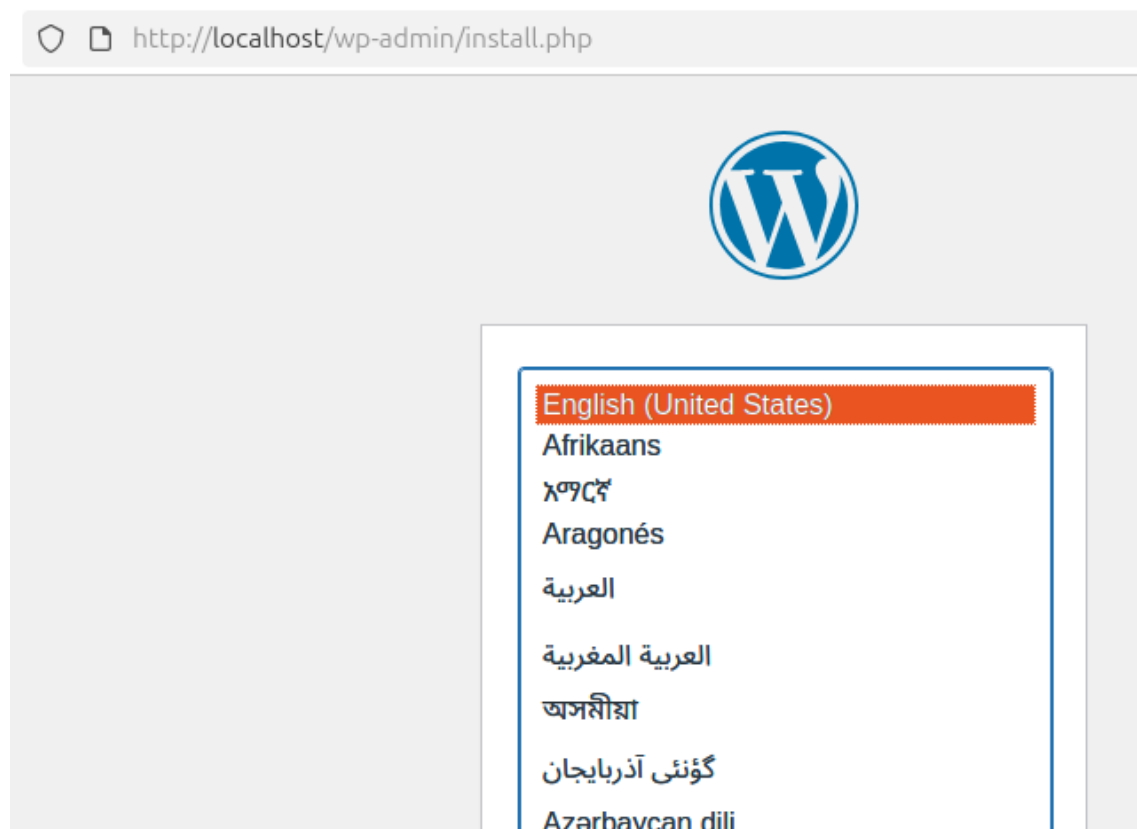
```
sudo systemctl reload apache2
```

## Verificación Final

Después de ejecutar estos comandos, vuelve a tu navegador y accede a la dirección:

<http://localhost/>

Deberías ver la pantalla de bienvenida y el asistente de instalación de WordPress solicitándote que selecciones el idioma. Si es así, ¡la configuración de Apache estará completa!



## Paso 8: Completar la Instalación a Través del Navegador

El último paso es el asistente de configuración web de WordPress.

1. Accede a `http://localhost/` en tu navegador.

### ¿Qué sigue?

- Ahora que tienes WordPress instalado localmente, puedes comenzar a:
  - Explorar la interfaz de administración (`http://localhost/wp-admin/`).
  - Instalar temas y plugins para personalizar y mejorar tu sitio.
  - Crear contenido como páginas y entradas de blog.
2. Deberías ver la pantalla de instalación de WordPress.
  3. Sigue las instrucciones en pantalla para **elegir el idioma, el nombre del sitio y establecer el nombre de usuario y contraseña del administrador.**

¡Con esto, tu instalación local de WordPress en Ubuntu 24.04 estará completa!

Título del sitio: `diversionconbanderas`

Nombre de usuario: `sheldon`

Contraseña: `penny`

Tu correo electrónico: [sheldon@diversionconbanderas.local](mailto:sheldon@diversionconbanderas.local)

Título del sitio	<input type="text" value="diversionconbanderas"/>
Nombre de usuario	<input type="text" value="sheldon"/> <small>Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.</small>
Contraseña	<div><input type="password" value="penny"/><div><div>Muy débil</div></div><div>Ocultar</div></div> <b>Importante:</b> Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.
Confirma la contraseña	<input checked="" type="checkbox"/> Confirma el uso de una contraseña débil.
Tu correo electrónico	<input type="text" value="sheldon@diversionconbanderas.local"/> <small>Comprueba bien tu dirección de correo electrónico antes de continuar.</small>
Visibilidad en los motores de búsqueda	<input type="checkbox"/> Pedir a los motores de búsqueda que no indexen este sitio <small>Depende de los motores de búsqueda atender esta petición o no.</small>
<div>Instalar WordPress</div>	

## ¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

**Nombre de usuario** sheldon

**Contraseña** *La contraseña que has elegido.*


[Acceder](#)

Nombre de usuario o correo electrónico

sheldon

Contraseña

penny



☐ Recuérdame

Acceder

