

Manual de Instalación de Entorno PHP con MongoDB y Ratchet

Introducción

Este manual proporciona instrucciones detalladas para instalar y configurar un entorno PHP en un servidor Ubuntu que permita el uso de MongoDB y Ratchet para WebSockets.

Requisitos Previos

- Un servidor Ubuntu (ej. Ubuntu 24.04)
- Acceso SSH al servidor
- Permisos de administrador (root)

Paso 1: Actualizar el Sistema

Antes de instalar nuevos paquetes, es importante asegurarse de que el sistema esté actualizado.

```
```bash
sudo apt update
sudo apt upgrade
```
```

Paso 2: Instalar PHP y Extensiones Necesarias

Instala PHP y las extensiones necesarias, incluyendo `php-mongodb`.

```
```bash
sudo apt install php php-cli php-mbstring php-xml php-zip php-curl php-mongodb
```
```

Verificar la Instalación de PHP

Asegúrate de que PHP está instalado correctamente.

```
```bash
php -v
```
```

Paso 3: Instalar Composer

Composer es un gestor de dependencias para PHP. Instálalo con los siguientes comandos:

```
```bash
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'abcdef...') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
sudo mv composer.phar /usr/local/bin/composer
```
```

Reemplaza `abcdef...` con el hash actual de la verificación de Composer que puedes encontrar en la página de [instalación de Composer](https://getcomposer.org/download/).

Verificar la Instalación de Composer

```
```bash
composer --version
```
```

Paso 4: Crear un Proyecto y Agregar Dependencias

Crea un nuevo directorio para tu proyecto y navega hacia él:

```
```bash
mkdir /var/www/html/mi_proyecto
cd /var/www/html/mi_proyecto
```
```

Inicializar Composer

Ejecuta el siguiente comando para crear un archivo `composer.json`:

```
```bash
composer init
```
```

Sigue las instrucciones para configurar tu proyecto.

Instalar Dependencias de MongoDB y Ratchet

Instala las dependencias de MongoDB y Ratchet con:

```
```bash
composer require mongodb/mongodb
composer require cboden/ratchet
```
```

Paso 5: Configuración de MongoDB

Instalar MongoDB

Sigue estos comandos para instalar MongoDB:

```
```bash
wget -qO - https://www.mongodb.org/static/pgp/server-8.0.asc | sudo apt-key add -
echo "deb [arch=amd64,arm64]
https://repo.mongodb.org/apt/ubuntu/dists/noble/mongodb-org/8.0/multiverse/
binary-amd64/ ." | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
sudo apt update
sudo apt install -y mongodb-org
```
```

Iniciar MongoDB

```
```bash
sudo systemctl start mongod
sudo systemctl enable mongod
```
```

Verificar la Instalación de MongoDB

Verifica que MongoDB esté funcionando:

```
```bash
sudo systemctl status mongod
```
```

Paso 6: Configuración del Servidor Apache

Instalar Apache

Si aún no lo tienes instalado, instala Apache:

```
```bash
sudo apt install apache2
```
```

Habilitar Módulos de Apache

Habilita los módulos necesarios:

```
```bash
sudo a2enmod rewrite
sudo systemctl restart apache2
```
```

Configurar los Permisos

Asegúrate de que Apache tiene acceso a tu proyecto:

```
```bash
sudo chown -R www-data:www-data /var/www/html/mi_proyecto
sudo chmod -R 755 /var/www/html/mi_proyecto
```
```

Paso 7: Crear el Script PHP

Crea un archivo PHP en tu directorio de proyecto para probar la conexión a MongoDB:

```
```php
<?php
require '../vendor/autoload.php'; // Cambia la ruta si es necesario

use MongoDB\Client;

try {
 $client = new Client("mongodb+srv://username:password@cluster-url/db-name");
 // Reemplaza con tus credenciales
 $db = $client->libreria; // Cambia 'libreria' por tu base de datos
 echo "Conexión establecida con MongoDB";
} catch (Exception $e) {
 echo "Error: " . $e->getMessage();
}
?>
```
```

Paso 8: Probar el Script

Accede al script desde tu navegador:

```
```
http://<tu_ip>/mi_proyecto/test_mongo.php
```
```

Si todo está bien configurado, deberías ver un mensaje indicando que la conexión se ha establecido.

Solución de Problemas

1. ****Error 500 en el Navegador****: Revisa los registros de errores de Apache:

```
```bash
sudo tail -f /var/log/apache2/error.log
```
```

2. ****Clase `MongoDB\Client` No Encontrada****: Asegúrate de que la extensión de MongoDB esté habilitada y que `vendor/autoload.php` esté correctamente incluido.

3. ****Conexión a MongoDB Fallida****: Verifica las credenciales y que tu IP esté permitida en MongoDB Atlas.

Paso 9: Ejecutar el Servidor Ratchet en Segundo Plano

Para ejecutar un script PHP que actúe como un servidor Ratchet en segundo plano, puedes usar `nohup` o crear un servicio `systemd`.

Usando nohup

```
```bash
nohup php /var/www/html/mi_proyecto/server.php > /dev/null 2>&1 &
```
```

Creando un Servicio systemd

Crea un archivo de servicio:

```
```bash
sudo nano /etc/systemd/system/ratchet.service
```
```

Agrega lo siguiente:

```
```ini
[Unit]
Description=Ratchet WebSocket Server

[Service]
ExecStart=/usr/bin/php /var/www/html/mi_proyecto/server.php
Restart=always
User=www-data

[Install]
WantedBy=multi-user.target
```
```

Habilita y ejecuta el servicio:

```
```bash
sudo systemctl enable ratchet
sudo systemctl start ratchet
```
```

Conclusión

Siguiendo este manual, deberías haber configurado un entorno PHP en Ubuntu con soporte para MongoDB y Ratchet. Si encuentras problemas, revisa los registros de errores y asegúrate de que todas las dependencias estén correctamente instaladas.