# **WordPress Deployment Guide**

## **📌 Project Overview**

This guide provides step-by-step instructions for manually setting up and deploying a WordPress website using **GitHub Actions**, **Nginx**, and a **custom domain name**.

# Step 1: Setting Up the Repository

* **Create a new GitHub repository** named Wordpress-Assignment.
* **Initialize the repository** and add necessary files.

| git clone https://github.com/Raghava1201/Wordpress-Assignment.git  cd Wordpress-Assignment |
| --- |

# Step 2: Setting Up WordPress Manually

# **Download WordPress** from the official website: https://wordpress.org/download/

# **Extract WordPress files** and move them to the project directory.

# **Configure the database**:

# Create a MySQL database.

# Create a database user and grant permissions.

# Update wp-config.php with database details.

# **Start a local server** and place WordPress files in the web directory.

# **Access WordPress in a browser** and complete the installation at:

| http://localhost |
| --- |

# 🌐 Step 3: Setting Up Nginx and Domain Name

## **Install Nginx**:

| sudo apt update  sudo apt install nginx -y |
| --- |

## **Configure Nginx for WordPress**:

## Create a new Nginx configuration file:

| sudo nano /etc/nginx/sites-available/wordpress |
| --- |

## Add the following configuration:

| server {  listen 80;  server\_name wordpress.raghava.site (public.ip);  root /var/www/html;  index index.php index.html index.htm;  location / {  try\_files $uri $uri/ /index.php?$args;  }  location ~ \.php$ {  include snippets/fastcgi-php.conf;  fastcgi\_pass unix:/run/php/php7.4-fpm.sock;  fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  include fastcgi\_params;  }  } |
| --- |

## Save the file and create a symlink:

| sudo ln -s /etc/nginx/sites-available/wordpress /etc/nginx/sites-enabled/ |
| --- |

## Restart Nginx:

| sudo systemctl restart nginx |
| --- |

## **Verify the domain name is working**:

| https://wordpress.raghava.site |
| --- |

# ⚙ Step 4: Creating the GitHub Actions Workflow

## **Create a `` file** to automate the deployment process.

## **Configure the workflow** to:

## Trigger on push to the main branch.

## Deploy WordPress automatically.

## Restart Nginx after deployment.

## **Add the following GitHub Actions workflow**:

| name: Deploy WordPress  on:  push:  branches:  - main  jobs:  deploy:  runs-on: ubuntu-latest  steps:  - name: Checkout Repository  uses: actions/checkout@v3  - name: Deploy to Server  uses: appleboy/ssh-action@v0.1.10  with:  host: ${{ secrets.SERVER\_IP }}  username: ${{ secrets.SERVER\_USER }}  key: ${{ secrets.SSH\_PRIVATE\_KEY }}  script: |  cd /var/www/html  git pull origin main  sudo systemctl restart nginx  sudo systemctl restart php8.1-fpm |
| --- |

## **Commit and push the workflow file**:

| git add .github/workflows/deploy.yml  git commit -m "Added deploy.yml file for WordPress"  git push origin main |
| --- |

## 

## **Verify the workflow run in GitHub Actions** (Success ✅).