5-Minute CI/CD Setup Guide for VPS

Automate deployment from GitHub to any Linux VPS (Hostinger, DigitalOcean, AWS, etc.)

This guide sets up a pipeline where pushing code to a specific branch (e.g., main or home) automatically deploys it to your server.

Step 1: VPS Setup (Create the "Key")

Time: 1 Minute

Log into your VPS via SSH and run these commands to generate a dedicated deployment key.

1. **Generate the Key Pair:**
   ```
   ssh-keygen -t ed25519 -C "github-actions"
   # Press ENTER for all prompts (no passphrase)
   ```

2. Authorize the Key:
   This tells the server to accept connections using this key.
   ```
   cat ~/.ssh/id_ed25519.pub >> ~/.ssh/authorized_keys
   ```

3. Copy the Private Key:
   You will need this specific block of text for the next step.
   ```
   cat ~/.ssh/id_ed25519
   ```

   Action: Copy the **entire** output (from -----BEGIN... to -----END...).

Step 2: GitHub Setup (Add the Secrets)

Time: 1 Minute

Tell GitHub how to log into your server securely without exposing passwords.

1. Go to your Repository on GitHub.
2. Navigate to **Settings** > **Secrets and variables** > **Actions**.
3. Click **New repository secret** and add these three:

Name	Value
VPS_HOST	Your VPS IP Address (e.g., 123.45.67.89)
VPS_USERNAME	Usually root (or your specific user)
VPS_SSH_KEY	Paste the **Private Key** you copied in Step 1

Step 3: Create the Automation File

Time: 2 Minutes

Create a file in your project repository to control the robot.

File Path: .github/workflows/deploy.yml
(Create these folders if they don't exist)
Content (Copy & Paste):

```
name: Deploy to VPS

on:
  push:
    branches:
      - main  # CHANGE THIS: The branch you want to deploy (e.g., 'home', 'master')

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Deploy via SSH
        uses: appleboy/ssh-action@master
        with:
          host: ${{ secrets.VPS_HOST }}
          username: ${{ secrets.VPS_USERNAME }}
          key: ${{ secrets.VPS_SSH_KEY }}
          script: |
            # 1. Navigate to your project folder
            cd /var/www/your-project-folder

            # 2. Pull the latest code
            git fetch --all
            git checkout main
            git pull origin main

            # 3. Build commands (Edit based on your stack)
            # Example for Next.js/Node:
            npm install
            npm run build
            pm2 restart all
```

```
# Example for Python/Django:
# source venv/bin/activate
# pip install -r requirements.txt
# python manage.py migrate
# sudo systemctl restart gunicorn
```

Step 4: Finalize

Time: 1 Minute

1. **Edit the script:** inside deploy.yml, change /var/www/your-project-folder to your actual folder path.
2. **Push:** Commit and push the .github/workflows/deploy.yml file to GitHub.
3. **Verify:** Go to the **Actions** tab in your GitHub repo. You should see the workflow running.

Success! Your project now auto-deploys whenever you push to that branch.