Configure GitHub → Jenkins Webhook (step by step)

Step 0: Basic requirement (must-have)

- Jenkins server is **running**
- Jenkins is **reachable from GitHub**

(public IP, DNS, or exposed via tunnel)

Step 1: Create or open a Jenkins job

- Open **Jenkins dashboard**
- Click **New Item** (or open existing job)
- Choose **Freestyle project** (easy for interviews)
- Click **OK**

Step 2: Configure source code (GitHub)

- Go to **Source Code Management**
- Select **Git**
- Paste your **GitHub repository URL**
- Add GitHub credentials (if private repo)

Term explained

- **Repository:** Place where code is stored

Step 3: Enable webhook trigger in Jenkins

- Scroll to **Build Triggers**
- Check ✅ **GitHub hook trigger for GITScm polling**

Meaning

- Jenkins is now ready to listen to GitHub events

Step 4: Get Jenkins webhook URL

- Jenkins webhook URL is:

`http://<jenkins-ip>:8080/github-webhook/`

Term explained

- **Webhook URL / Endpoint:** Address where GitHub sends notifications

Step 5: Add webhook in GitHub

- Open **GitHub repository**
- Go to **Settings**
- Click **Webhooks**
- Click **Add webhook**

Fill details:

- **Payload URL:** Jenkins webhook URL
- **Content type:** application/json
- **Events:** Just the **Push event**
- Click **Add webhook**

Step 6: Test the webhook

1. Make a small code change
2. Push code to GitHub
3. Jenkins job starts automatically

Common mistakes (interview gold)

- Jenkins not publicly accessible ❌
- Wrong webhook URL ❌

- Forgot to enable trigger in Jenkins ❌

Real-world example

- Like enabling **delivery notifications**:
    - Order placed → notification sent → action happens

Interview one-liner

“To configure a GitHub webhook, enable the GitHub hook trigger in Jenkins and add Jenkins’ webhook URL in the GitHub repository settings.”