**Pulling Code Automatically from GitHub**

**Step 1: Create a New Jenkins Job**

1. Open Jenkins Dashboard.
2. Click **“New Item”**.
3. Enter a name (e.g., GitHub-AutoBuild).
4. Choose **Freestyle project**, then click **OK**.

**Step 2: Configure Source Code Management (GitHub Repo)**

1. In the project configuration page, scroll to **Source Code Management**.
2. Select **Git**.
3. In **Repository URL**, enter your GitHub repo URL (e.g., https://github.com/yourusername/your-repo.git).
4. Under **Credentials**, click **Add**, and provide:
   * GitHub username & password (or token), or
   * SSH key (for private repos).
5. Select the correct **branch** (e.g., main or master).

**Step 3: Add Build Trigger for GitHub Webhook**

1. Scroll down to **Build Triggers**.
2. Check the box: **“GitHub hook trigger for GITScm polling”**.

This allows GitHub to notify Jenkins when a change is pushed, instead of Jenkins polling constantly.

**Step 4: Add a Simple Build Step (Optional)**

1. Scroll to **Build** section.
2. Click **Add build step** → **Execute shell** (on Linux/macOS) or **Execute Windows batch command**.
3. Enter a command like:

echo "Code pulled successfully!"

ls -l

**Step 5: Save the Job**

Click **Save** at the bottom of the page.

**🌐 Step 6: Set Up GitHub Webhook**

1. Go to your GitHub repository.
2. Click **Settings** → **Webhooks** → **Add webhook**.
3. In the **Payload URL**, enter:

http://<your-jenkins-url>/github-webhook/

(e.g., http://yourdomain.com:8080/github-webhook/)

1. Set **Content type** to application/json.
2. Choose **Just the push event**.
3. Click **Add webhook**.

*Make sure your Jenkins instance is accessible from the internet (or GitHub’s IP ranges) for the webhook to reach it.*

**Test the Integration**

1. Make a commit and push it to the GitHub repository.
2. Jenkins should automatically trigger the job.
3. Check the build logs to confirm that the code was pulled and the job ran successfully.