Setting Up Jenkins Pipeline with GitHub Webhooks

1. *Prerequisites*

- *Jenkins installed and running (accessible via http://localhost:8080 or your server URL).*
- *A GitHub account and a repository.*
- *Basic knowledge of Git, Jenkins, and pipeline scripting.*

2. *Install Jenkins and Required Plugins*

   a. *Install Jenkins:*

- *Download Jenkins from Jenkins.io and complete the installation.*
- *Access Jenkins in your browser.*

   b. *Install Necessary Plugins:*

- *Navigate to Manage Jenkins > Manage Plugins.*
- *Under the "Available" tab, search for and install the following plugins:*
- *GitHub Integration Plugin.*
- *Pipeline Plugin*
- *GitHub Branch Source Plugin(optional but useful).*

3. Set Up a Git Hub Repository

a. Create a Repository:

- *Go to GitHub and create a new repository.*

b. Push Your Code:

- *Add your project files to the repository.*
- *Ensure you include a Jenkinsfile in the root directory. This file defines the pipeline steps.*

4. Configure Jenkins for GitHub

a. Add GitHub Credentials to Jenkins:

- *In Jenkins, go to Manage Jenkins > Credentials.*
- *Under the appropriate domain, add a new credentials:*
- *Kind: Secret Text or Personal Access Token.*
- *Token: Generate a personal access token in GitHub with the following permissions:*

- *Repo*
- *Admin:repo_hook*

- *Save the credentials.*

b. Connect GitHub to Jenkins:

- *Navigate to Manage Jenkins > Configure System.*
- *Scroll to the GitHub section.*

- *Add your GitHub server and paste the token into the required field.*

5. Create a Jenkins Job

a. New Job:

- *Go to Jenkins and click New Item.*
- *Select Pipeline, and give it a name.*

b. Job Configuration:

- *Under Pipeline, choose Pipeline script from SCM.*
- *Select Git as the SCM.*
- *Provide the repository URL (e.g., https://github.com/username/repository.git).*
- *Specify the branch to monitor(e.g., main or master).*
- *Add the credentials you configured earlier.*
- *Enter the Jenkinsfile path(default is Jenkinsfile).*

6. Write a Jenkinsfile:

Create a file named Jenkinsfile in the root of your repository. Below is an example:

```
pipeline {
    agent any
```

```
    stages {
        stage('Checkout Code') {
            steps {
                git branch: 'main', url:
'https://github.com/username/repository.git'
            }
        }
        stage('Build') {
            steps {
                echo 'Building the application...'
                // Add build commands, e.g., mvn clean
install
            }
        }
        stage('Test') {
            steps {
                echo 'Running tests...'
                // Add test commands, e.g., npm test or
mvn test
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying the application...'
                // Add deployment steps
            }
        }
    }
}
```

7. Set Up GitHub Webhook

a. Go to Your Repository:

- *In GitHub, navigate to your repository's Settings > Webhook.*

b. Add a Webhook:

- *Click Add Webhook.*
- *Enter the Payload URL as your Jenkins server URL followed by /GitHub-webhook/ (e.g., http://your-jenkins-url/github-webhook/).*
- *Set the content type to application/JSON.*
- *Choose the events to trigger the webhook(e.g., Push events).*
- *Save the webhook.*

8. Test the Pipeline:

a. Push Code to GitHub:

- *Make a change in your repository and push it to the monitored branch.*

b. Monitor Jenkins:

- *Open Jenkins and check if the pipeline is triggered automatically.*
- *The pipeline should execute the steps defined in the Jenkinsfile.*