



Jenkins

Using shared
libraries in
Jenkins

- Aman Joshi

Why do we need shared
libraries?

Why do we need shared libraries?

- ▶ Avoid writing the same code from scratch for multiple projects
- ▶ Simplifies the process of pushing source code updates for a project
- ▶ We might want to customize implementation of external plugins
- ▶ Implement custom code

How to use shared libraries?

How to use shared libraries?

- ▶ Open the Jenkins dashboard in your web browser
- ▶ Click the **Manage Jenkins** link on the left-hand side of the Jenkins dashboard
- ▶ Under *System Configuration*, click the **Configure System** button
- ▶ Scroll down to the *Global Pipeline Libraries* section and click the **Add** button
- ▶ Fill in the details to configure the library
- ▶ Once you are done configuring the new library, click the **Save** button to save the changes to Jenkins
- ▶ Use @Library annotation, followed by library name to load the library

How did I do it?

- ▶ Install Jenkins on local machine
<https://www.jenkins.io/download/weekly/macOS/>
- ▶ Expose the Jenkins service running on local machine to internet
<https://ngrok.com/docs/getting-started>
- ▶ Create public GitHub repositories:
<https://github.com/amanjoshicodes/devops-test>
<https://github.com/amanjoshicodes/devops-driver>
- ▶ Configure shared library on Jenkins instance
- ▶ Create pipeline job to test code



Is this how things are done in industry?



What did we miss?

What did we miss?

- ▶ Webhooks :
<https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project#:~:text=Step 1%3A go to your,add %2Fgithub-webhook%2F>.
- ▶ Better UI :
<https://www.jenkins.io/doc/book/blueocean/getting-started/>
- ▶ GitHub App :
<https://docs.cloudbees.com/docs/cloudbees-ci/latest/cloud-admin-guide/github-app-auth>





Thank you!