**Lab: Maven Project**

**Pre-Requisites: -**

1. Git Repository to be used in this lab: <https://github.com/LovesCloud/java-tomcat-demo-app> Please fork the repo in your GitHub account.
2. [Configure GitHub Webhook for Jenkins](#_Configure_GitHub_Webhook) on your forked repo. follow the steps mentioned in [**Configure GitHub Webhook for Jenkins**](#_Configure_GitHub_Webhook_1)

***Steps To Follow:***

1. Click on **New Item**
2. Enter **Name** such as <yourname>\_mavenproject
3. Select **Maven project**
4. Click **OK**
5. Under Source Code Management section, Select **Git** radio button
   1. Enter Repository URL - <Git repo>

*(Note: Git Repository URL is the one forked as part of prerequisites)*

1. Under Build Triggers section, Select **GitHub hook trigger for GITScm polling** checkbox
2. Under Build section
   1. Select the **Maven Version** as maven 3.5.4
   2. Enter **Root POM**-pom.xml
   3. Enter **Goals and options**-clean package
3. Click **Save**
4. Click on **Build Now**
5. Click on the Build #
6. Click on **Console Output**
7. It should display the Finished status as **Success**
8. You can Scroll and Down and Read the Console Output to see what all actions took place.
9. Open the Build Job’s Details Page again.
10. On GitHub end, as you are the owner of the new forked Git repo, edit any file and commit the changes.
11. Open Jenkins again, Observe the new triggered build On Left Navigation Panel in Build Job details page.
12. Follow the same steps to see the log as we did above.
13. View the Console Output.
14. It should display the Finished status as Success

# **Configure GitHub Webhook for Jenkins**

1. Open GitHub
2. Navigate to Git Repo;
3. Navigate to Settings of repository
4. Click Webhook
5. Click Add Webhook
6. Enter Payload URL-http://<Public IP of Jenkins Server>:8080/github-webhook/
7. Click Save Webhook