Module 6: Automated and Continuous Deployment

Demo Document - 3

edureka!

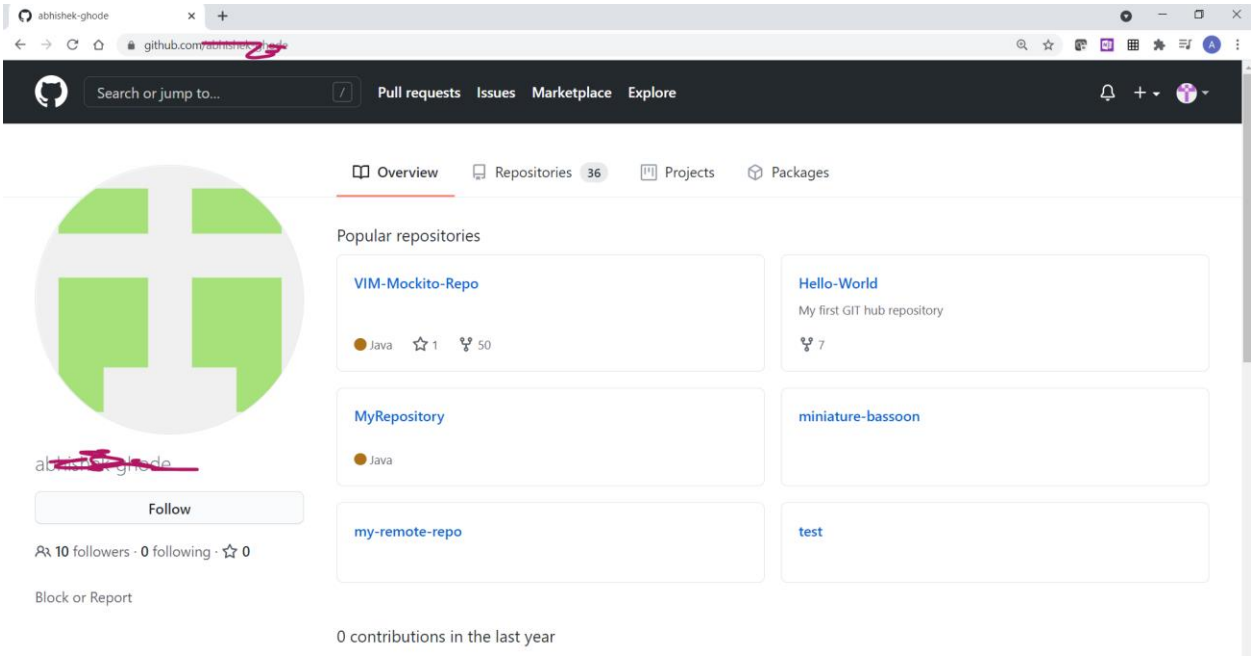
Demo: Deployment of Java app with Jenkins pipeline and Github

Problem statement: Execute commands to do show integrations with github

Solution:

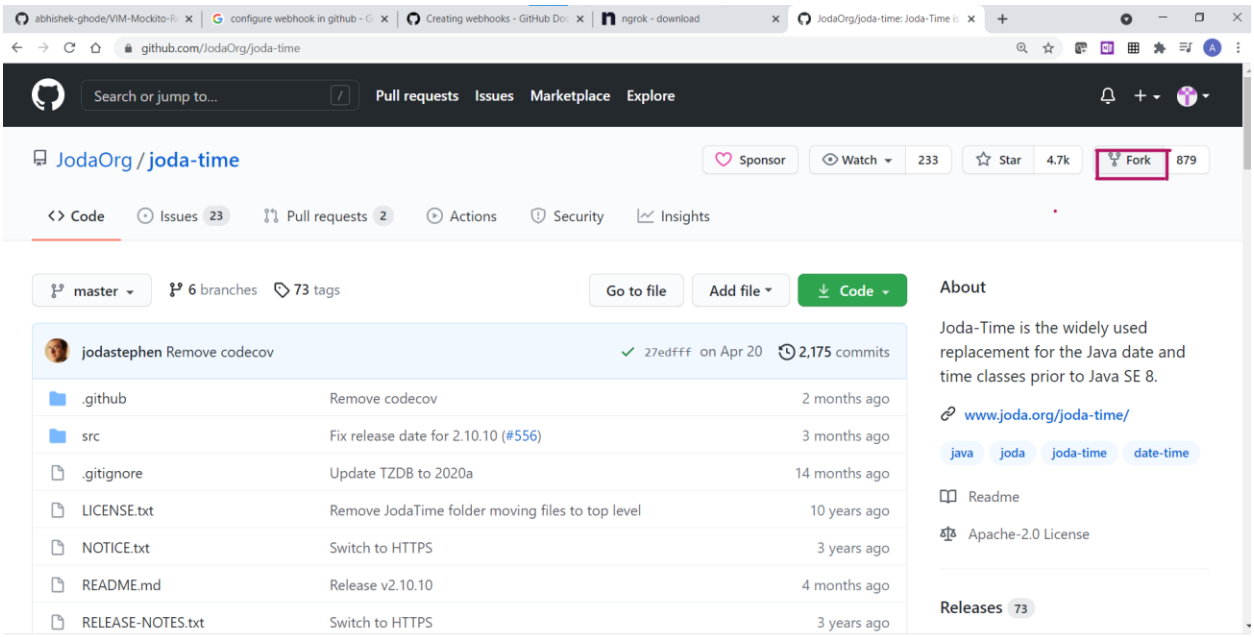
Step 1: There is a prerequisite that we need to complete this LAB. Please login to gihub.com.

URL to access: github.com



Step 2: We will fork a repository.

URL to access: https://github.com/JodaOrg/joda-time



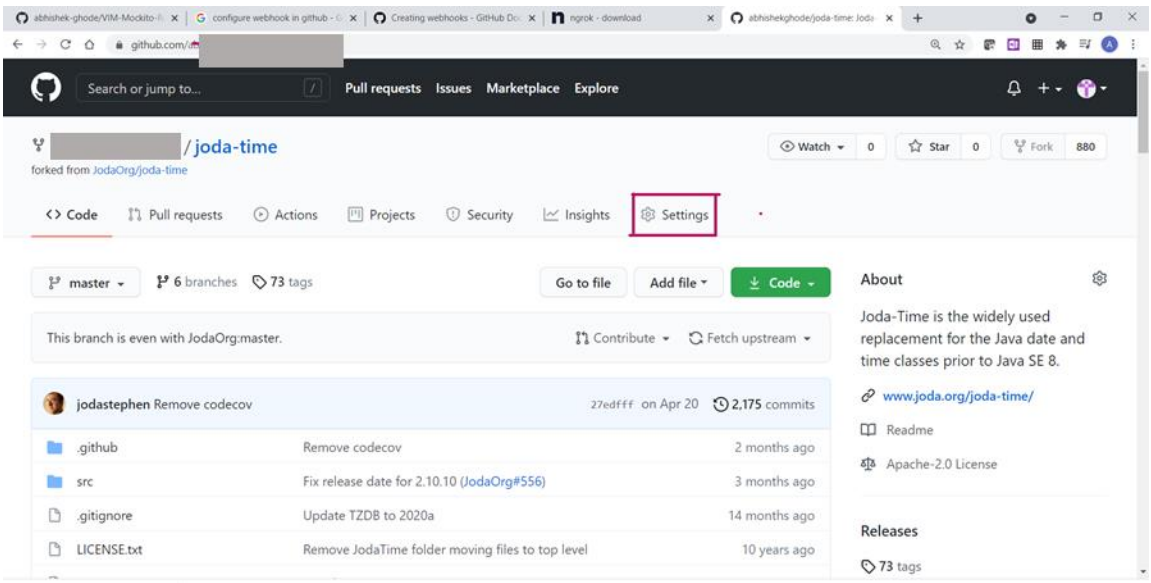
So here we are working with a repository which we used in previous LABS. This time we are forking this repository. The purpose is to make changes to the codebase. We will do it in the last step of this LAB.

Step 3: We will configure a webhook here for the repository that we forked.

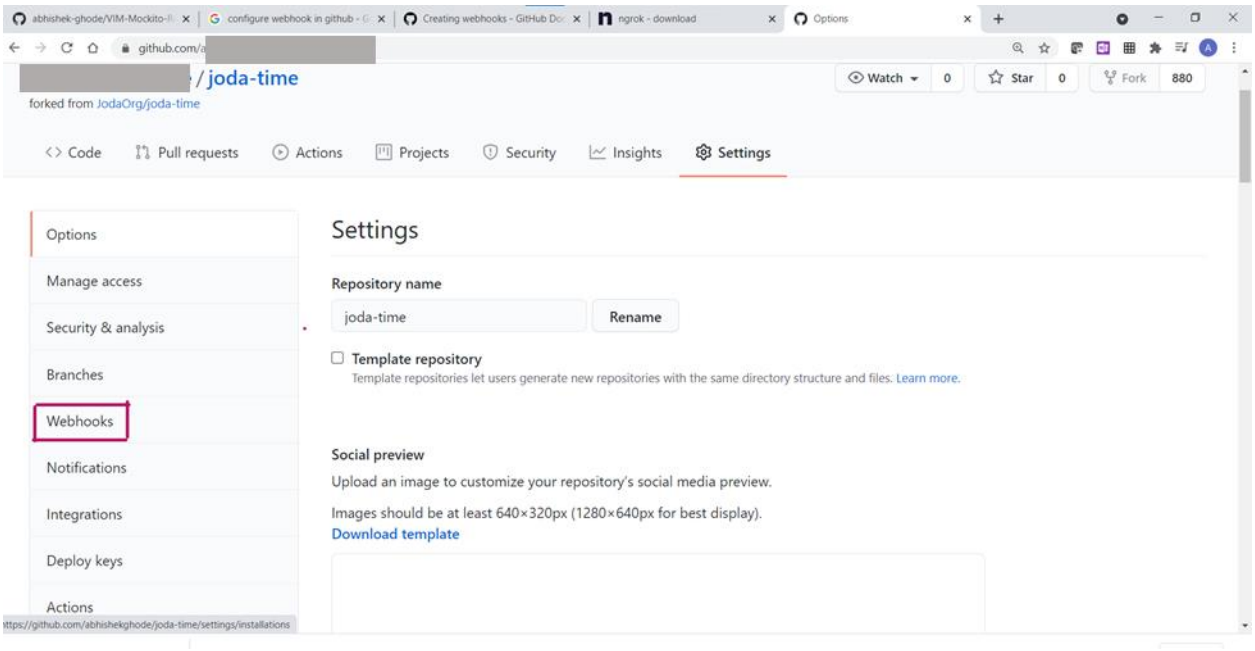
Prior to that, please follow below link to map your machine exposing localhost to the internet.

https://ngrok.com/download

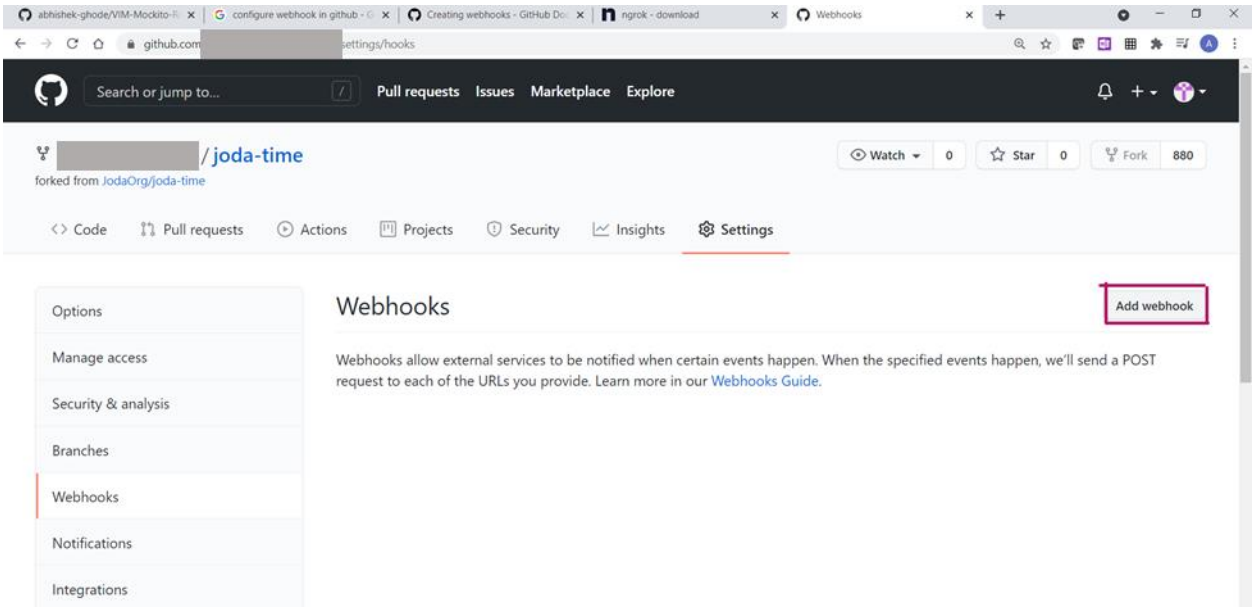
Now please click on settings of your forked repository. We will be adding a webhook for this repository.



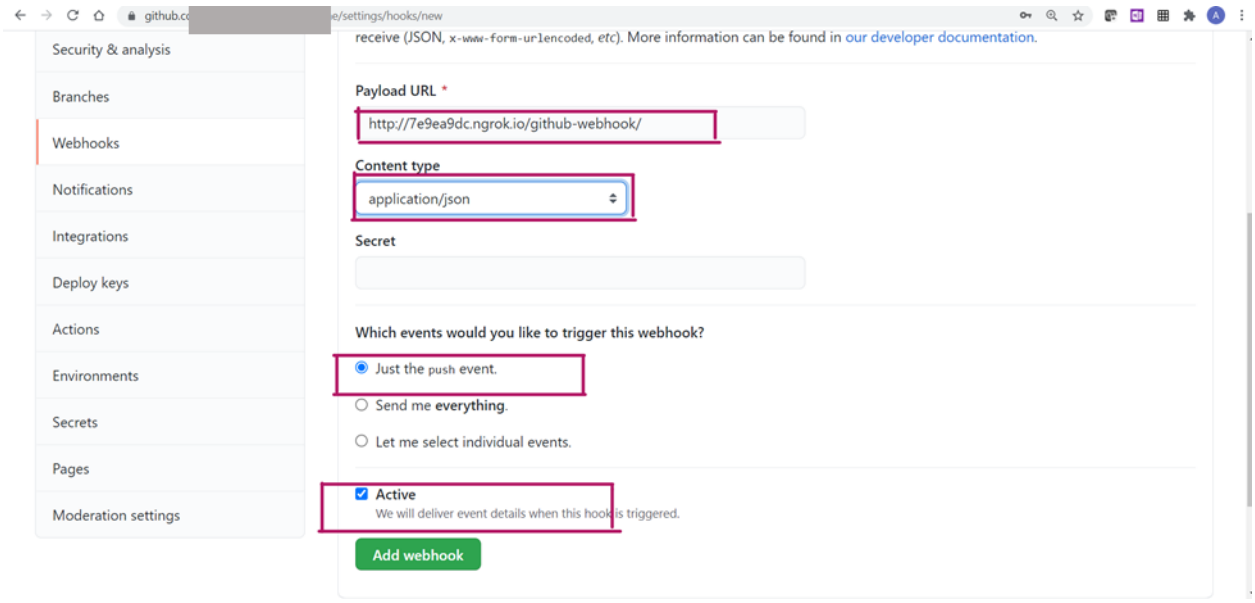
Click on ‘Webhooks’



Click on ‘Add webhook’



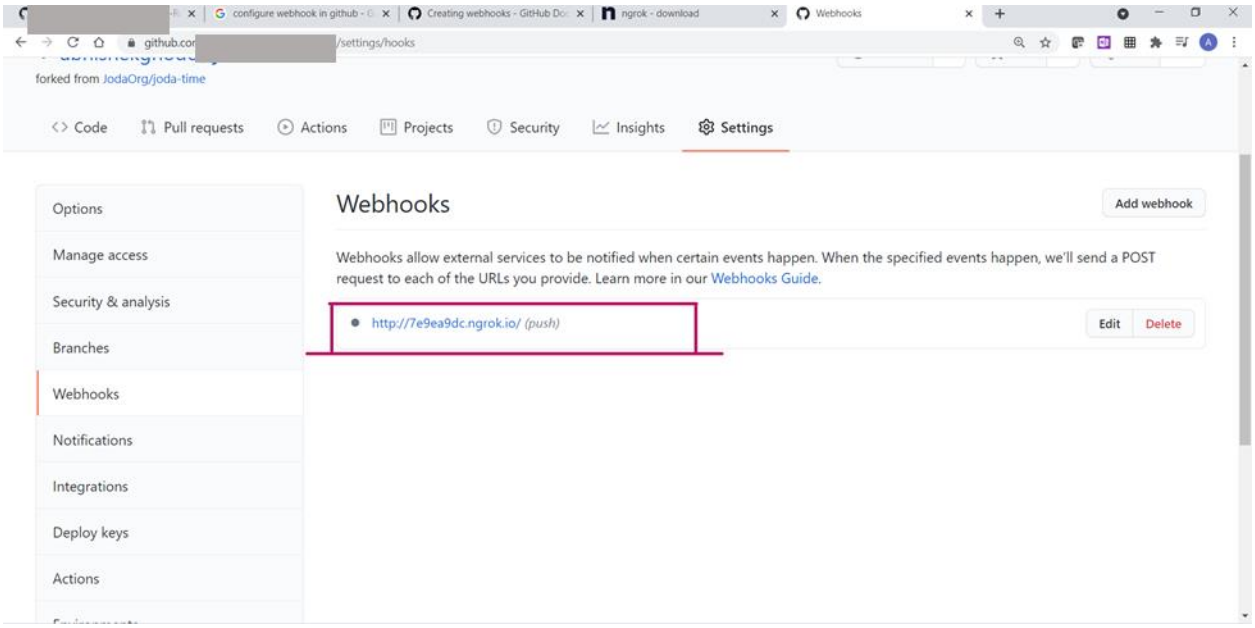
Please add below details for adding a webhook.



Payload URL should be `*.ngrok.io`. Do not forget to append `github-webhook/` in the Payload URL. The `application/json` content type will POST the body of the request as JSON payload.

Please do ensure to keep webhook deliveries are 'Active'.

Observe the webhook which is recently added.

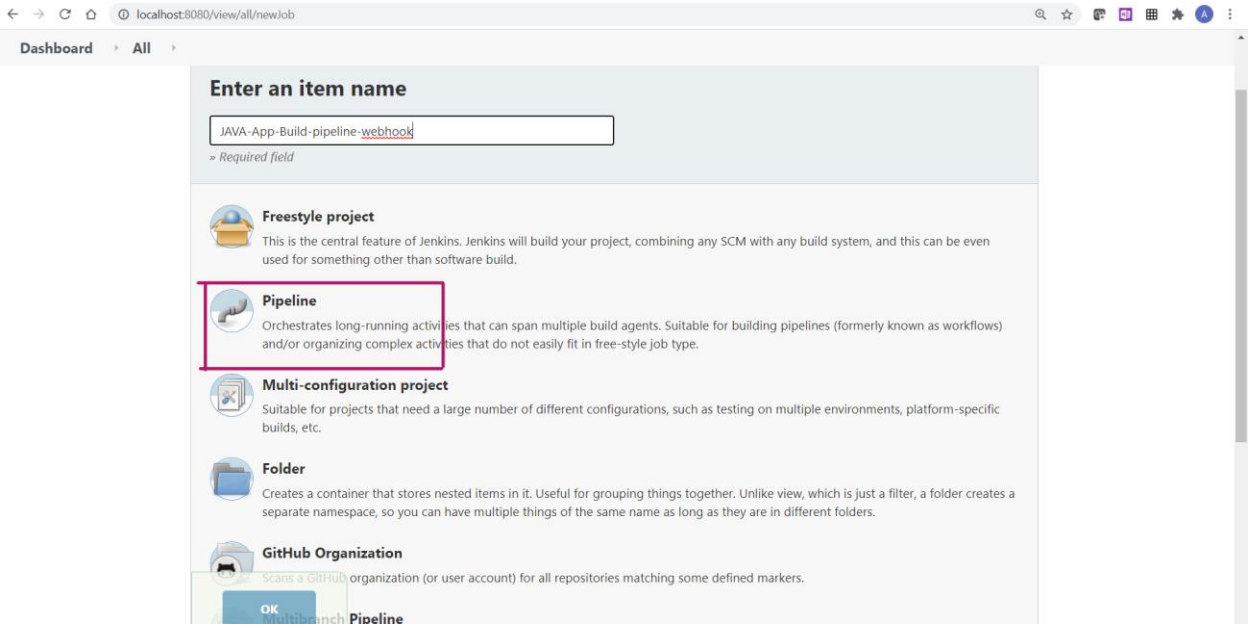


Step 4: Now we will create a Jenkins pipeline project.

URL to access: http://localhost:8080/view/all/newJob

Module 6 – Automated and Continuous Deployment

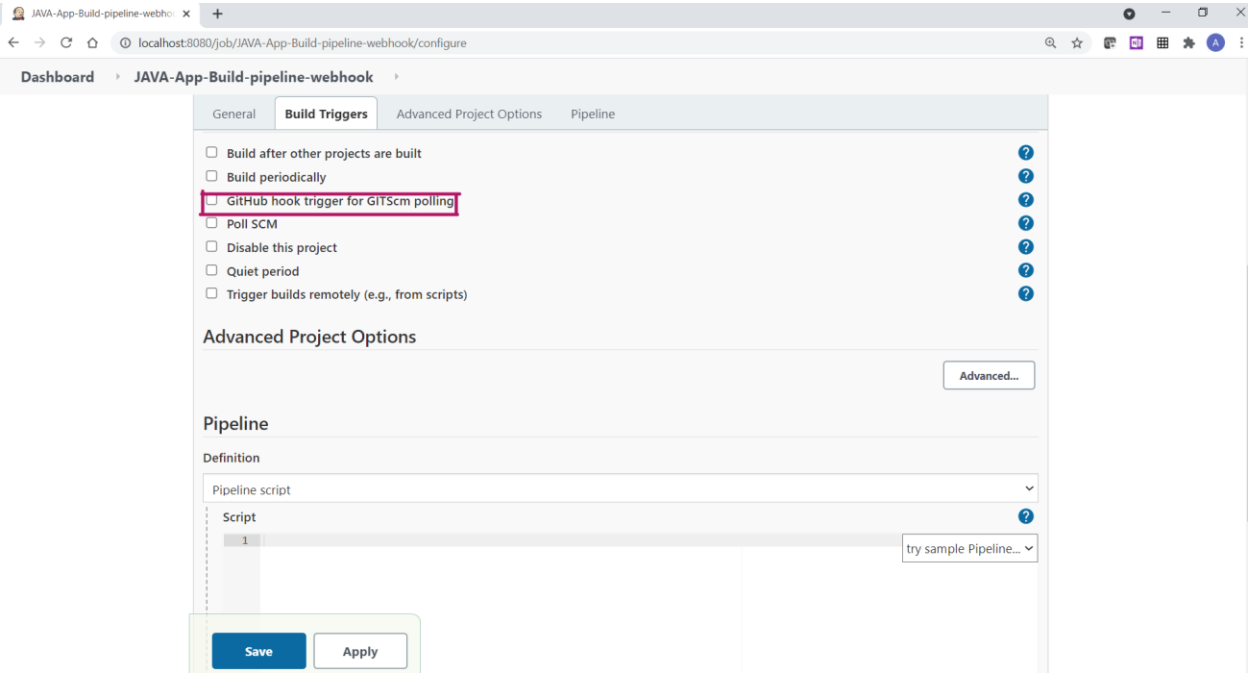
Please create a pipeline project named as ‘JAVA-App-Build-pipeline-webhook’.



Step 5: Now we can configure the pipeline project.

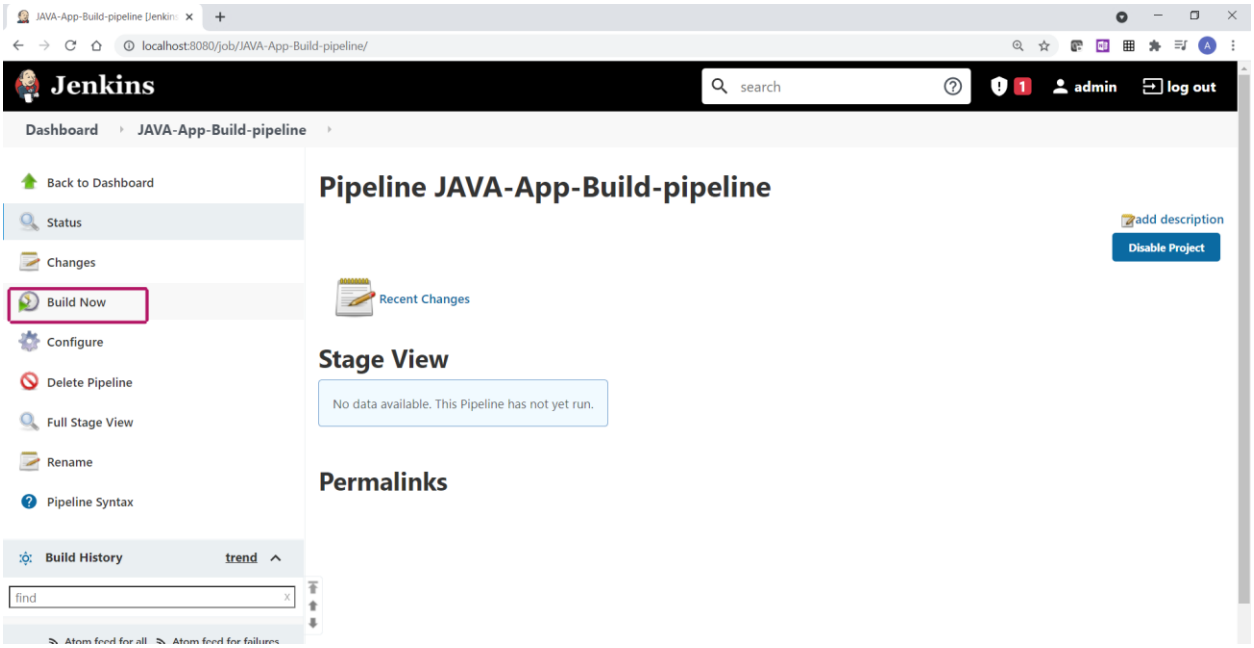
URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-webhook/configure

Please select ‘Build Trigger’ and check the option ‘github hook trigger for GIT scm polling’



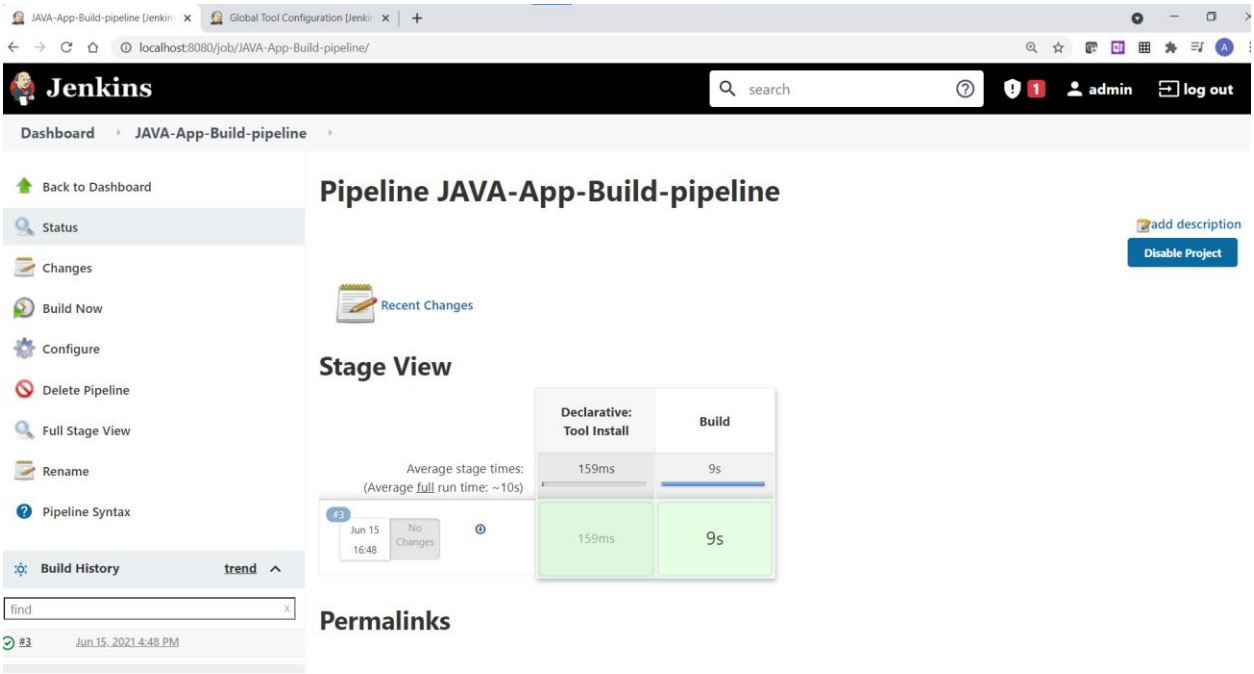
Step 6: You can now build the project. We did this step in the previous LAB.

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-webhook/



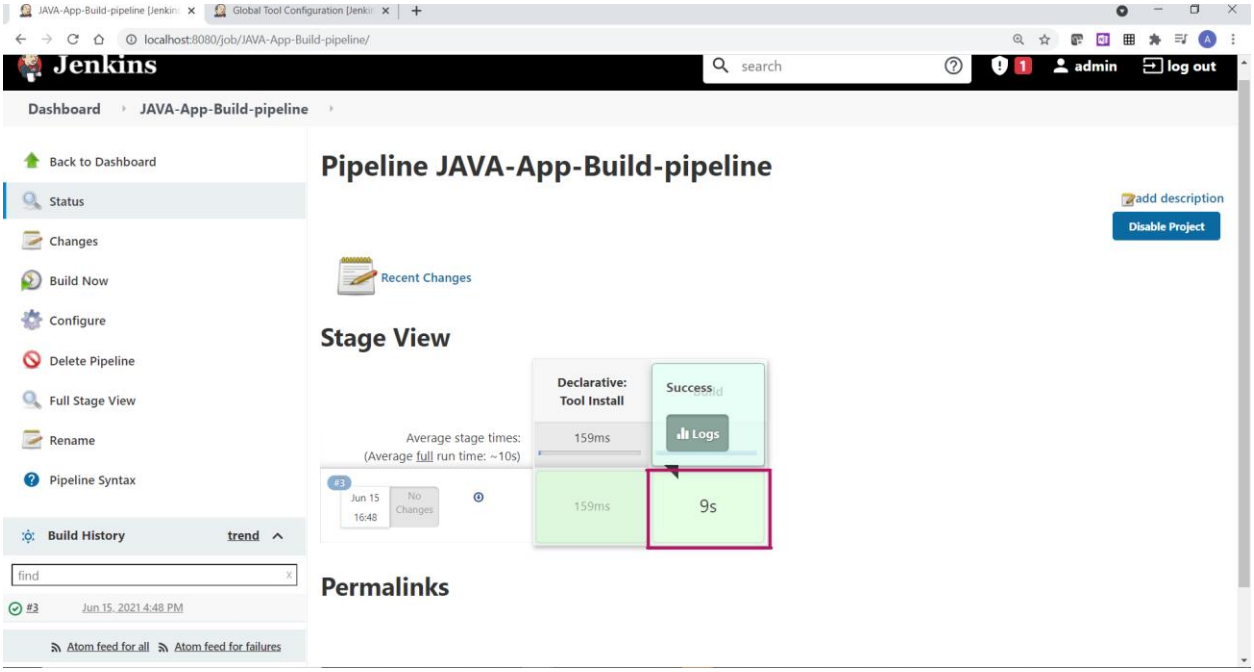
Step 7: You can also view whether build is successful or failed

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-webhook/



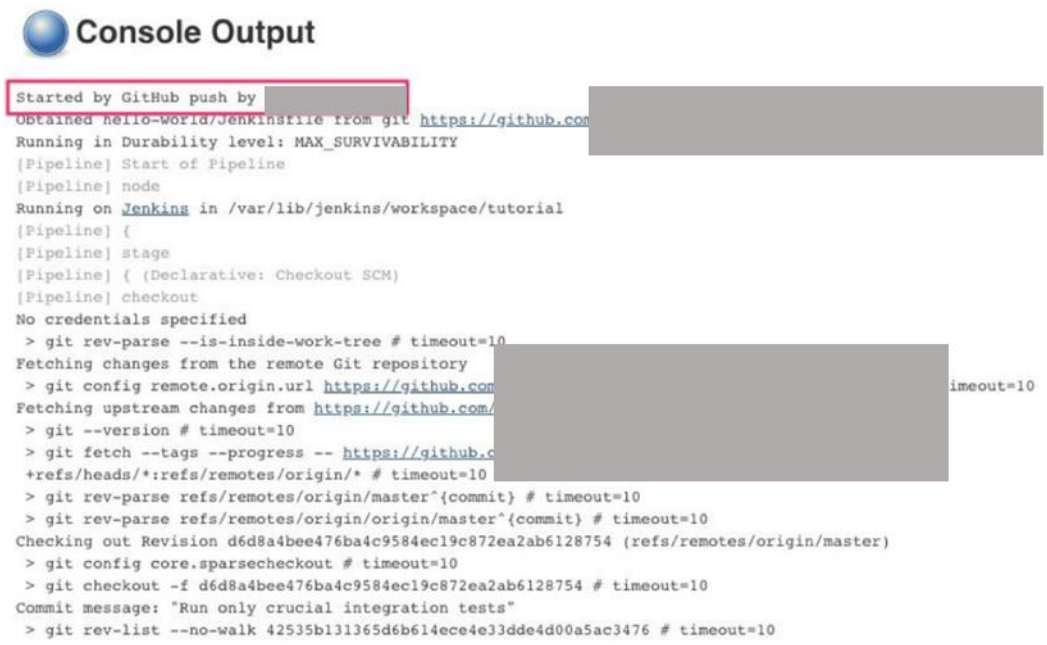
Step 5: View logs generated for the build.

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-webhook/



This build will have access to the artifact generated, test reports and useful information about the build execution in the form of logs.

Step 8: This step needs to be done on the repository that we forked earlier. Go to repository, you can edit readme.md for this demo and push the changes, commit and push. If you wait for few seconds, you would observe that there is a new run (build) of pipeline job.



You can observe the first statement in the log saying that build was started by github. It will also show your github user name. Now whenever someone pushes a change in the repository, the Jenkins job will be triggered automatically.