

Module 6: Automated and Continuous Deployment

Demo Document - 3

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

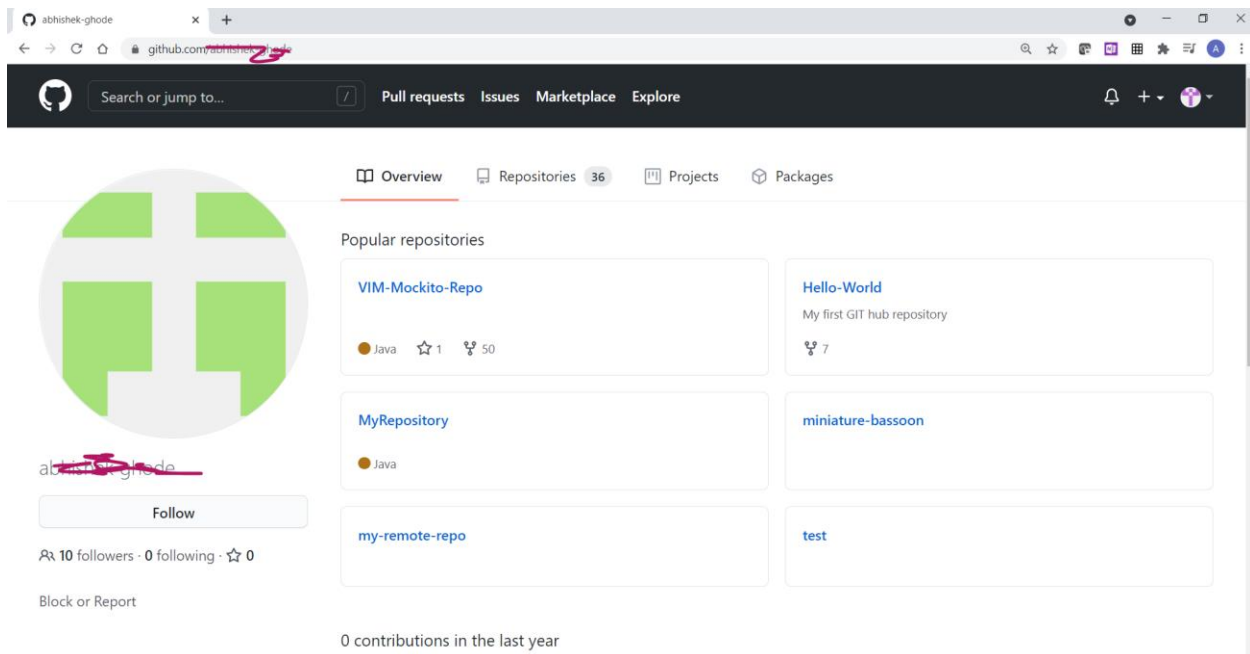
Demo: Deployment of Java app with Jenkins pipeline and Github

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 github.com.

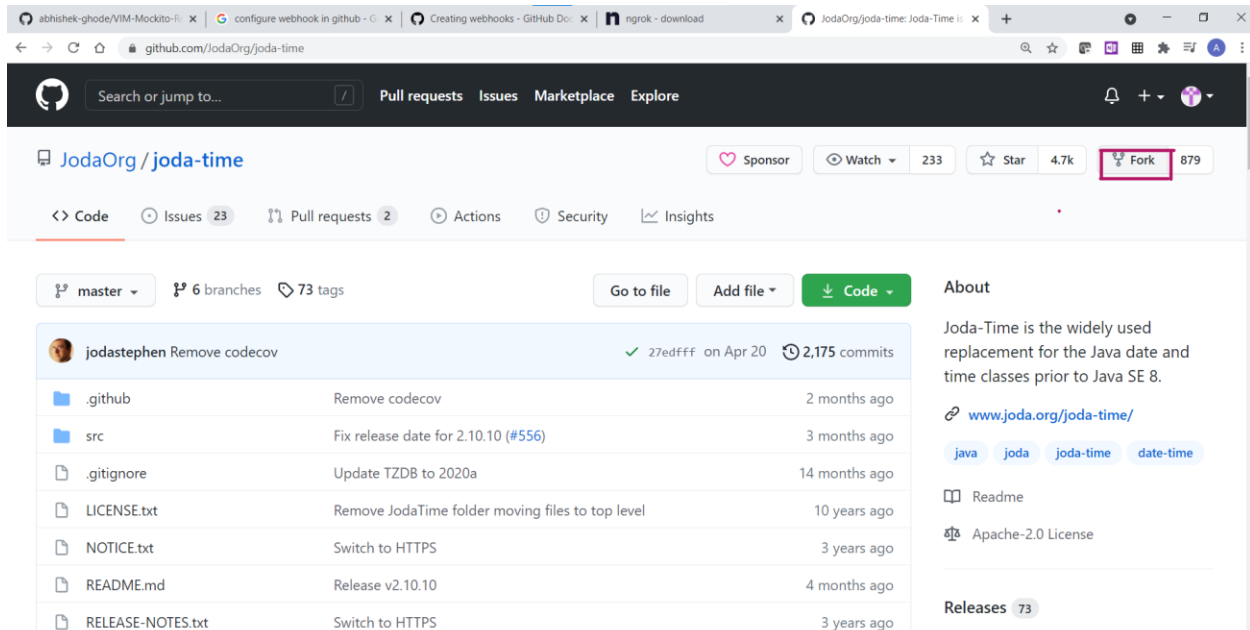
URL to access: github.com



Step 2: We will fork a repository.

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

Module 6 – Automated and Continuous Deployment



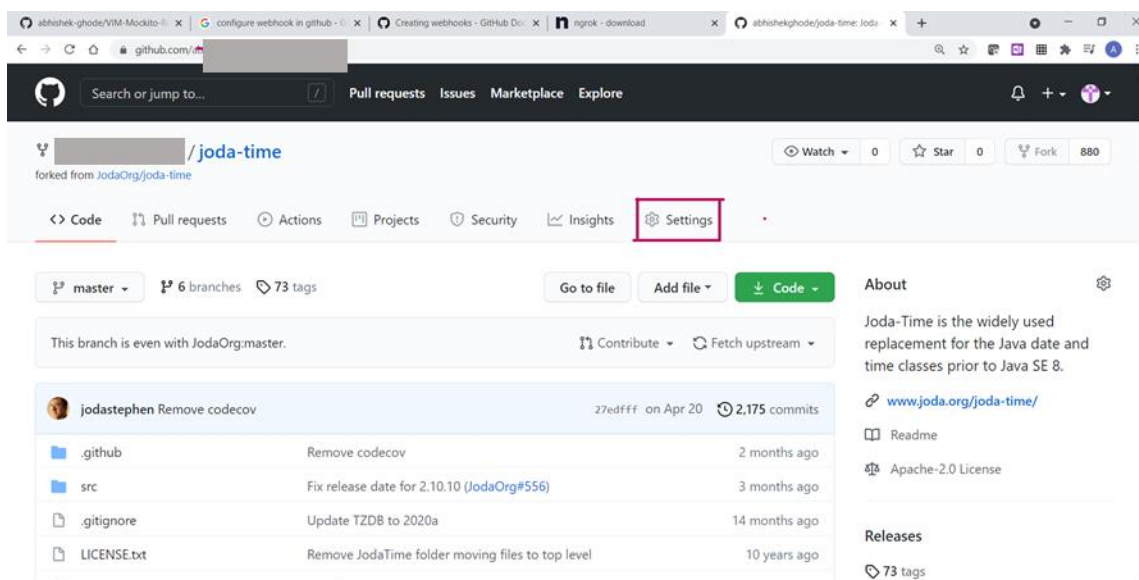
So here we are working with a repository which we used in previous LABS. This time we are forking this repository. The purpose is to make changes to the codebase. We will do it in the last step of this LAB.

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.

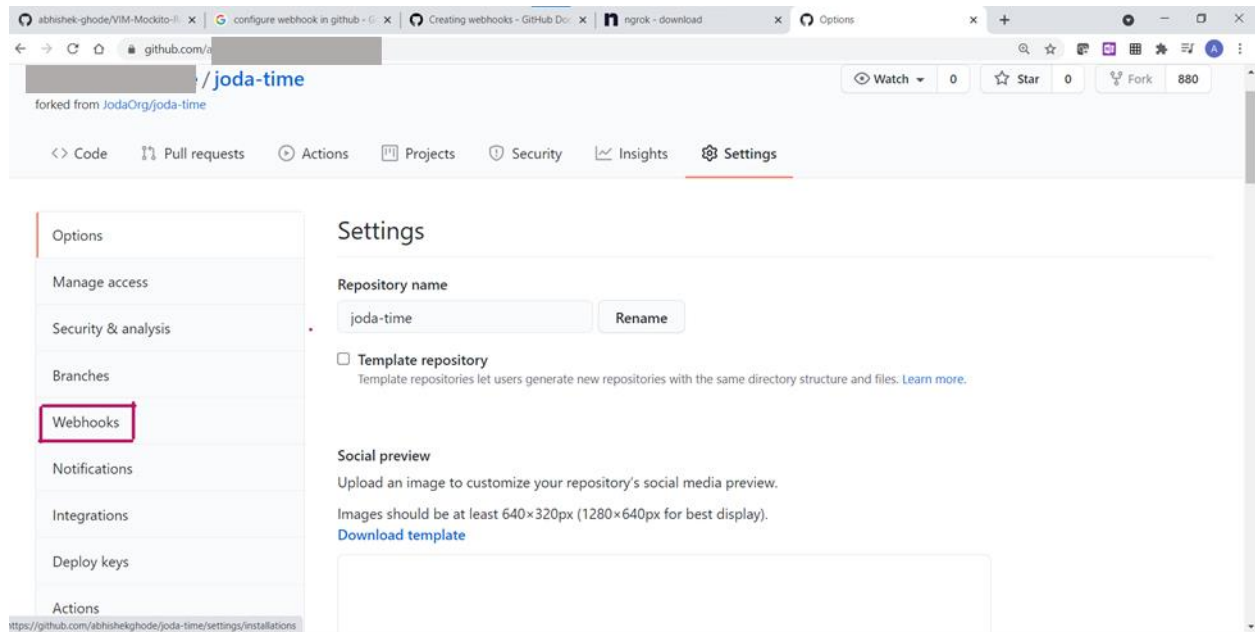
<https://ngrok.com/download>

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



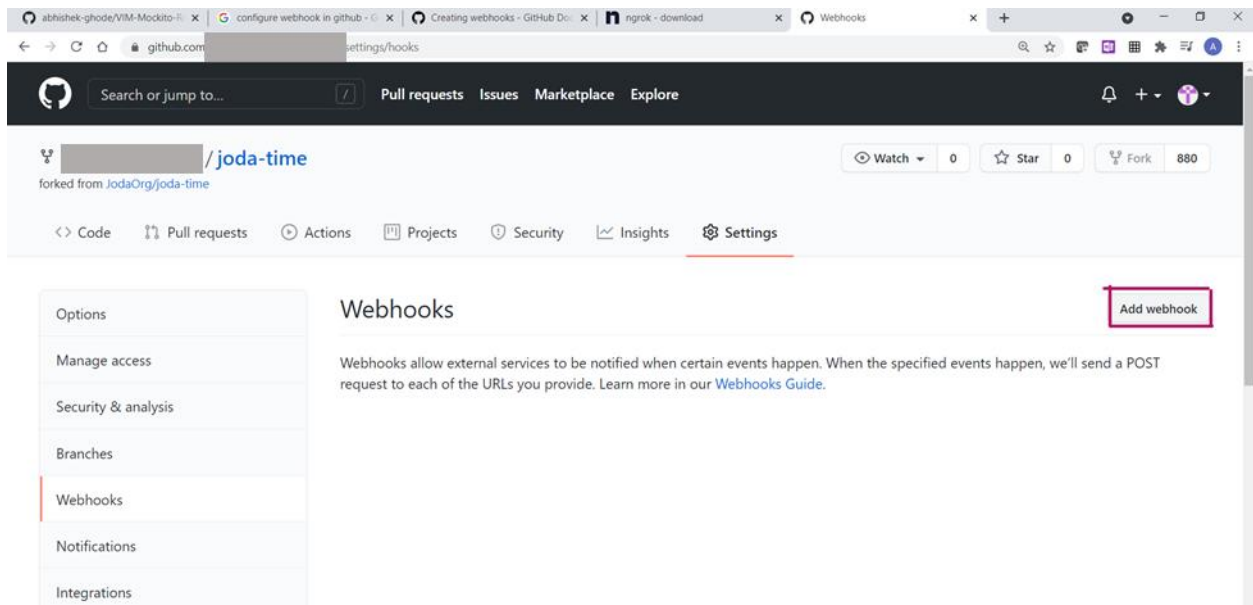
Module 6 – Automated and Continuous Deployment

Click on 'Webhooks'



Module 6 – Automated and Continuous Deployment

Click on 'Add webhook'



Please add below details for adding a webhook.

Module 6 – Automated and Continuous Deployment

receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *
http://7e9ea9dc.ngrok.io/github-webhook/

Content type
application/json

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.
☐ Send me everything.
☐ Let me select individual events.

☒ **Active**
We will deliver event details when this hook is triggered.

Add 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.

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Actions

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

• http://7e9ea9dc.ngrok.io/ (push) Edit Delete

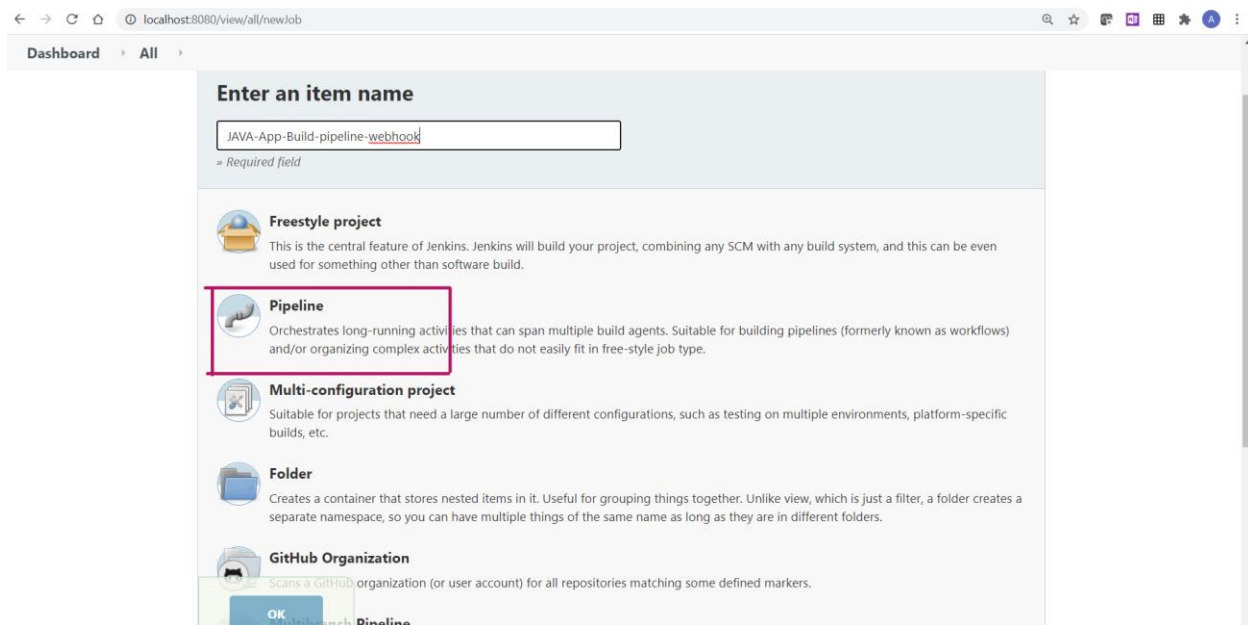
Add webhook

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

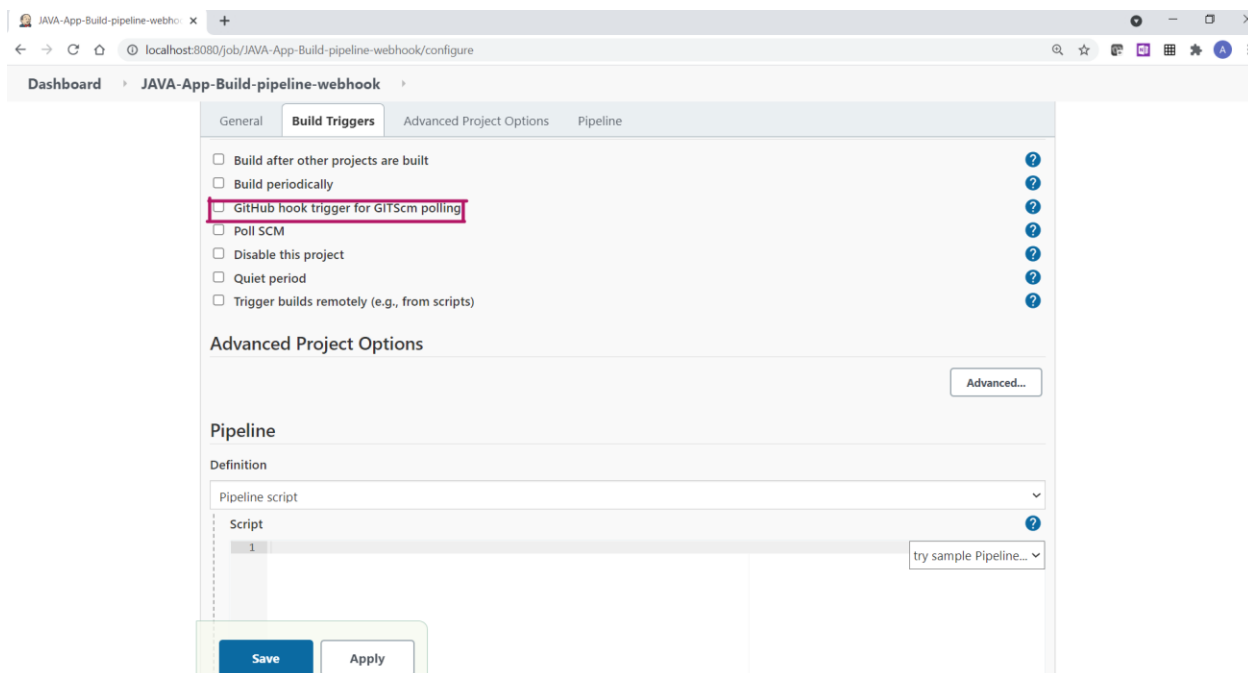
Please create a pipeline project named as 'JAVA-App-Build-pipeline-webhook'.



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'



Module 6 – Automated and Continuous Deployment

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/>

The screenshot shows the Jenkins dashboard for the 'JAVA-App-Build-pipeline' job. The left sidebar contains a list of actions: 'Back to Dashboard', 'Status', 'Changes', 'Build Now' (highlighted with a red box), 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. The main content area displays the 'Pipeline JAVA-App-Build-pipeline' title, a 'Recent Changes' section, a 'Stage View' section with a message 'No data available. This Pipeline has not yet run.', and a 'Permalinks' section. The top navigation bar includes a search bar, a help icon, a notification icon, a user profile icon labeled 'admin', and a 'log out' button.

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

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

The screenshot shows the Jenkins dashboard for the 'JAVA-App-Build-pipeline' job, displaying build history and stage view. The left sidebar contains a list of actions: 'Back to Dashboard', 'Status', 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. The main content area displays the 'Pipeline JAVA-App-Build-pipeline' title, a 'Recent Changes' section, a 'Stage View' section with a table showing average stage times, and a 'Permalinks' section. The top navigation bar includes a search bar, a help icon, a notification icon, a user profile icon labeled 'admin', and a 'log out' button.

Average stage times:	
Declarative: Tool Install	Build
159ms	9s

Average full run time: ~10s

Build #	Time	Status	Changes
#3	Jun 15 16:48	Success	No Changes

Module 6 – Automated and Continuous Deployment

Step 5: View logs generated for the build.

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

The screenshot shows the Jenkins web interface for a pipeline named 'JAVA-App-Build-pipeline'. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. Below these is the 'Build History' section, which shows a table of builds. The main content area displays the 'Stage View' of the pipeline. It shows a 'Declarative: Tool Install' stage with a 'Success' status and a 'Logs' button. Below this, a table shows the stage times for the last build: 'Jun 15 16:48' with 'No Changes' and a duration of '159ms'. The 'Average stage times' are also shown: 'Average stage times: (Average full run time: ~10s)'. The 'Permalinks' section is visible at the bottom.

Build	Status	Changes	Duration
#3	Success	No Changes	159ms

Stage	Time
Declarative: Tool Install	159ms

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.

Module 6 – Automated and Continuous Deployment

Console Output

```
Started by GitHub push by [REDACTED]
Obtained hello-world/Jenkinsfile from git https://github.com/[REDACTED]
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/tutorial
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/[REDACTED] # timeout=10
Fetching upstream changes from https://github.com/[REDACTED]
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/[REDACTED]
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision d6d8a4bee476ba4c9584ec19c872ea2ab6128754 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f d6d8a4bee476ba4c9584ec19c872ea2ab6128754 # timeout=10
Commit message: "Run only crucial integration tests"
> git rev-list --no-walk 42535b131365d6b614ece4e33dde4d00a5ac3476 # timeout=10
```

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.