



Name: Vanshika

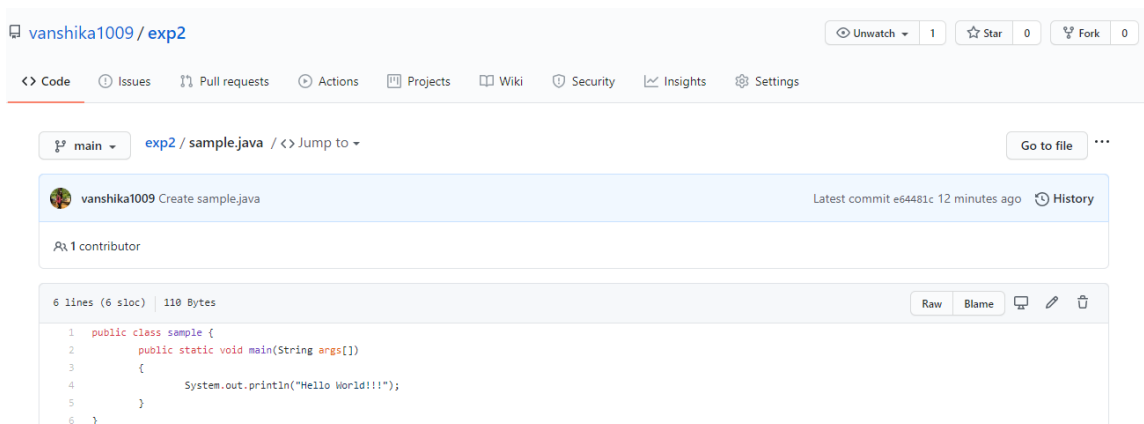
SAP: 500069061

Roll number: R171218110

CICD LAB EXPERIMENT 2

INTEGRATING GITHUB TO JENKINS

Step 1) We will start by writing and pushing our code to github.com.

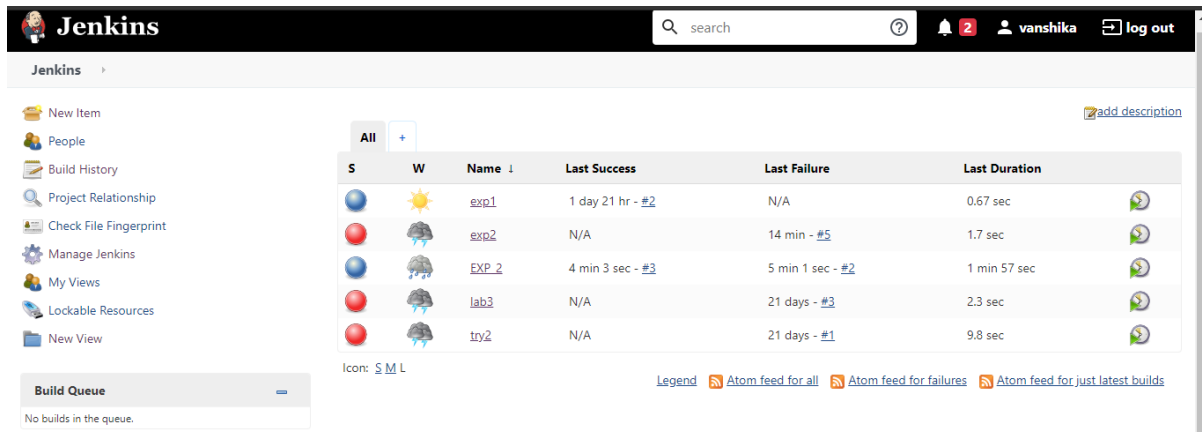


Step2) Start your jenkins with the command:

```
java -jar jenkins.war -httpPort = 8383
```

Go to your localhost to get to Jenkins.

Select “new item” to create your project.

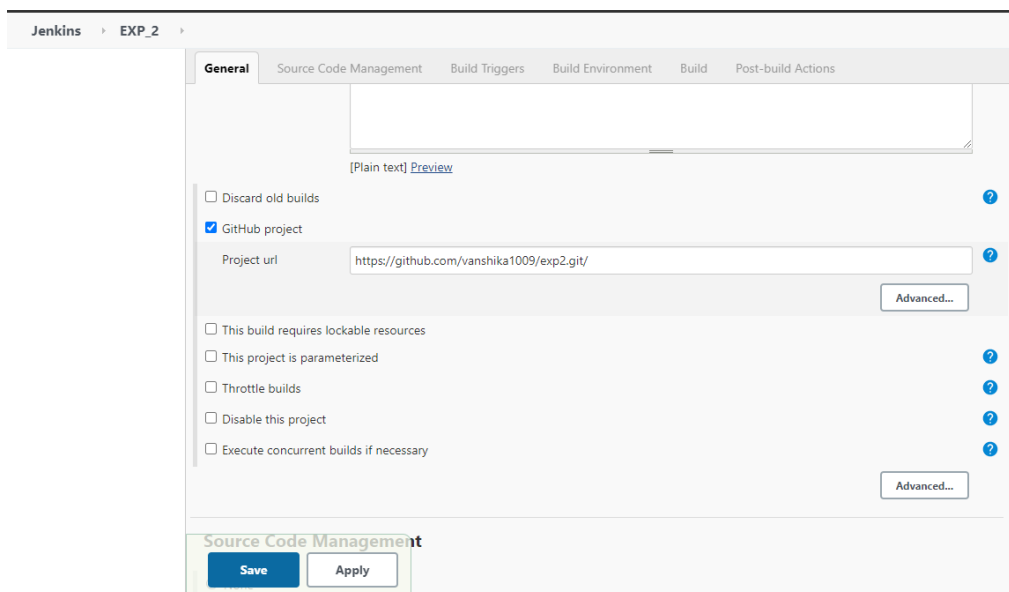


The screenshot shows the Jenkins dashboard. On the left is a sidebar with navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. The main area displays a table of projects under the 'All' filter. The table has columns for status (S), icon (W), name, last success, last failure, and last duration. Below the table are links for 'add description', 'Icon: S M L', and a 'Legend' section with Atom feed links for all, failures, and latest builds. A 'Build Queue' box at the bottom left shows 'No builds in the queue.'

S	W	Name	Last Success	Last Failure	Last Duration
		exp1	1 day 21 hr - #2	N/A	0.67 sec
		exp2	N/A	14 min - #5	1.7 sec
		EXP_2	4 min 3 sec - #3	5 min 1 sec - #2	1 min 57 sec
		lab3	N/A	21 days - #3	2.3 sec
		try2	N/A	21 days - #1	9.8 sec

Step3) Now give a name to your project and select “freestyle” as the type of project.

For the specifications, refer to the below pictures.



The screenshot shows the Jenkins configuration page for a project named 'EXP_2'. The 'General' tab is selected. It features a text area for the project name, a 'Discard old builds' checkbox, and a checked 'GitHub project' checkbox. The 'Project url' is set to 'https://github.com/vanshika1009/exp2.git/'. There are several checkboxes for build options like 'This build requires lockable resources', 'This project is parameterized', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. At the bottom, there are 'Save' and 'Apply' buttons under the 'Source Code Management' section.

Jenkins > EXP_2 >

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

☐ None
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser

Jenkins > EXP_2 >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Build

Execute Windows batch command

Command

See [the list of available environment variables](#)

Post-build Actions

Click the **“Save”** button to save the changes for the current project.

Step4) After you are done with the specification for your project, open your project and go to **“build now”** to start building the project.

- Back to Project
- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete build '#3'
- Git Build Data
- No Tags
- Previous Build

Console Output

```
Started by user vanshika
Running as SYSTEM
Building in workspace C:\Users\Dell\.jenkins\workspace\EXP_2
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\cmd\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\cmd\git.exe config remote.origin.url https://github.com/vanshika1009/exp2.git # timeout=10
Fetching upstream changes from https://github.com/vanshika1009/exp2.git
> C:\Program Files\Git\cmd\git.exe --version # timeout=10
> git --version # 'git version 2.25.1.windows.1'
> C:\Program Files\Git\cmd\git.exe fetch --tags --force --progress -- https://github.com/vanshika1009/exp2.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\cmd\git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
> C:\Program Files\Git\cmd\git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision e64481c720514db90c173099579cd55ff7175a65 (refs/remotes/origin/main)
> C:\Program Files\Git\cmd\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\cmd\git.exe checkout -f e64481c720514db90c173099579cd55ff7175a65 # timeout=10
Commit message: "Create sample.java"
First time build. Skipping changelog.
[EXP_2] $ cmd /c call C:\Users\Dell\AppData\Local\Temp\jenkins2948253877028069207.bat

C:\Users\Dell\.jenkins\workspace\EXP_2>javac sample.java

C:\Users\Dell\.jenkins\workspace\EXP_2>java sample
Hello World!!!
```

Voila! And you have finally integrated your Github with Jenkins.