Module 6: Jenkins Assignment - 2

Tasks To Be Performed:

1. Add 2 nodes to Jenkins master

2. Create 2 jobs with the following jobs:

a. Push to test

b. Push to prod

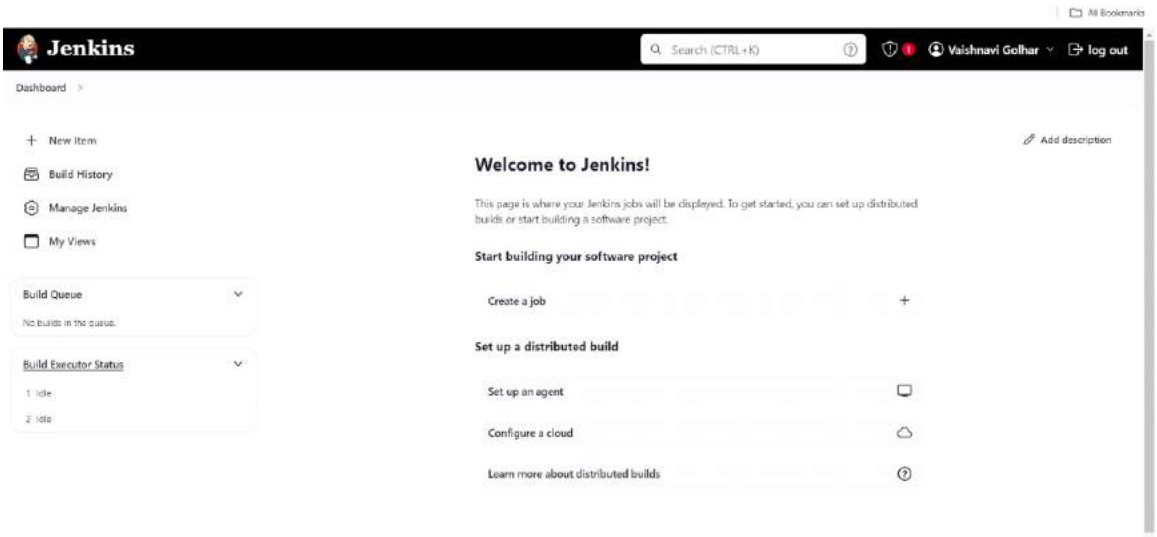
3. Once a push is made to test branch, copy Git files to test server

4. Once a push is made to master branch, copy Git files to Prod server

SOLUTION:

A) 3 instances have been successfully saved as “Jenkins-Master”, “Test” & “Prod”.

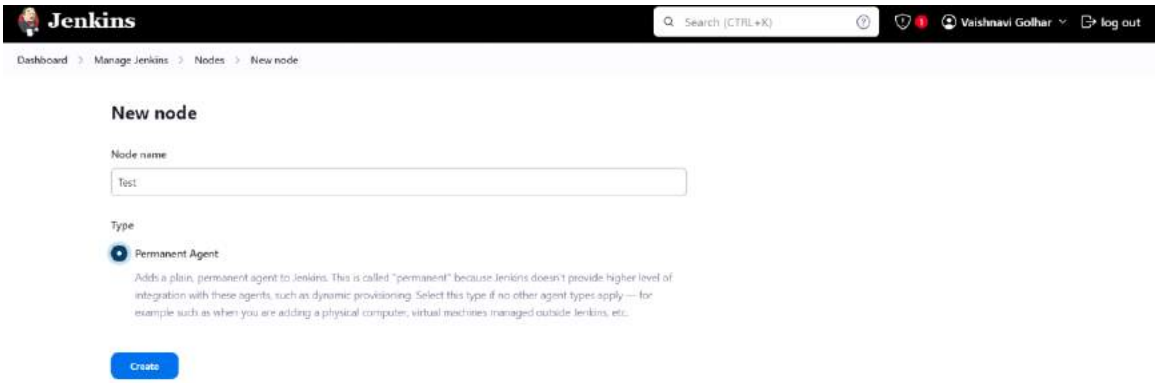
a} Installed Jenkins and Java on Jenkins-Master

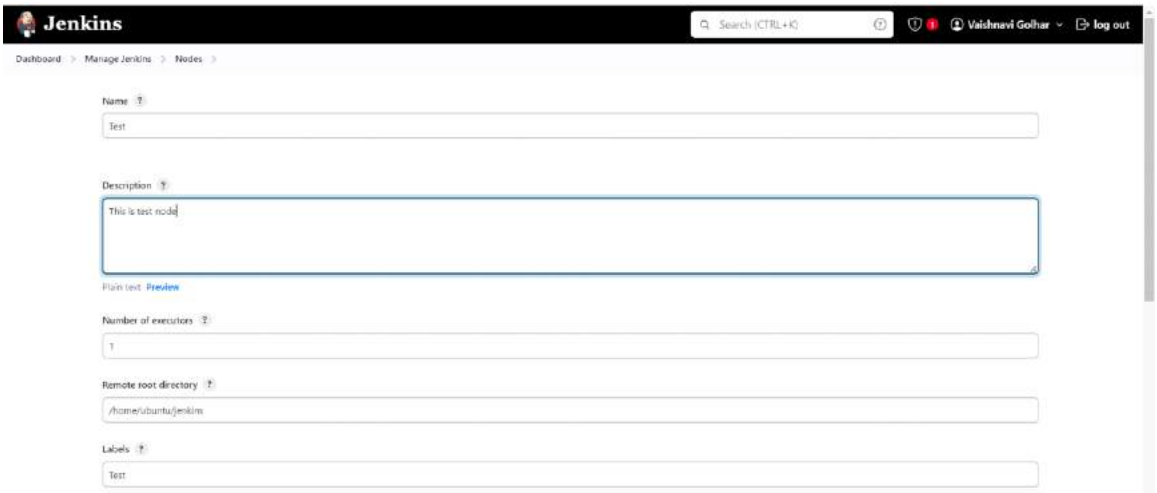


b} Install “Java” on “Test Server”.

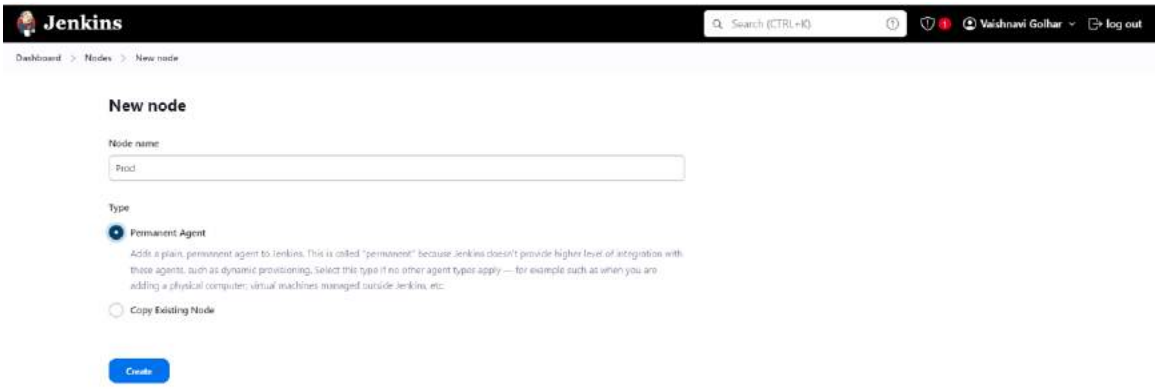
c} Install “Java” on “Prod Server”.

d} Add “Test Server” to Jenkins Master





e} Add “Prod Server” to Jenkins Master

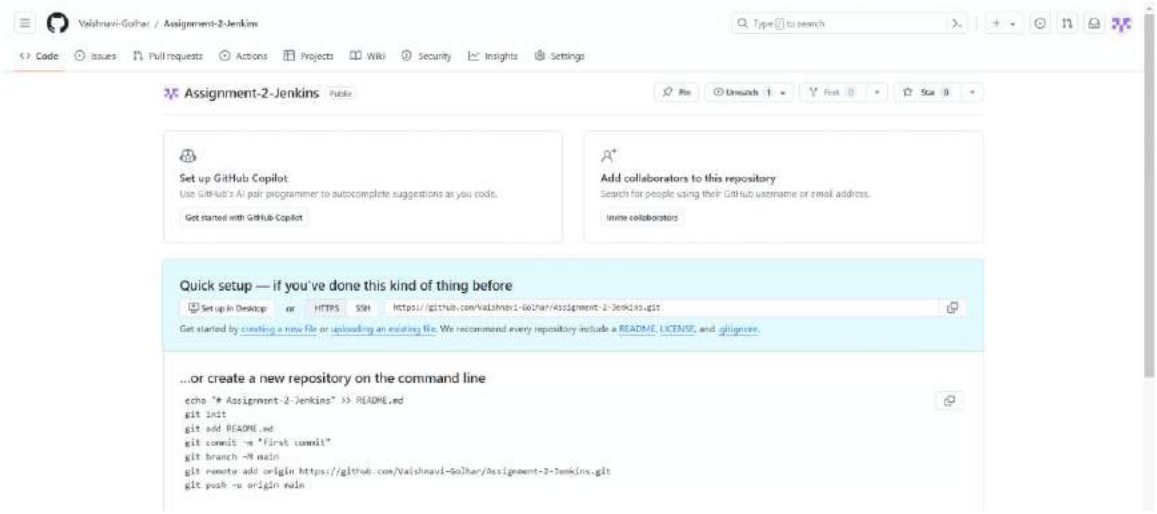


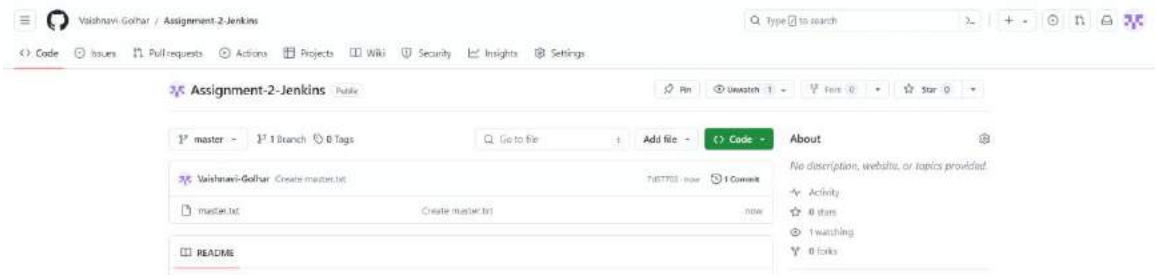
Problem (2) Solution: Create 2 jobs with the following jobs:

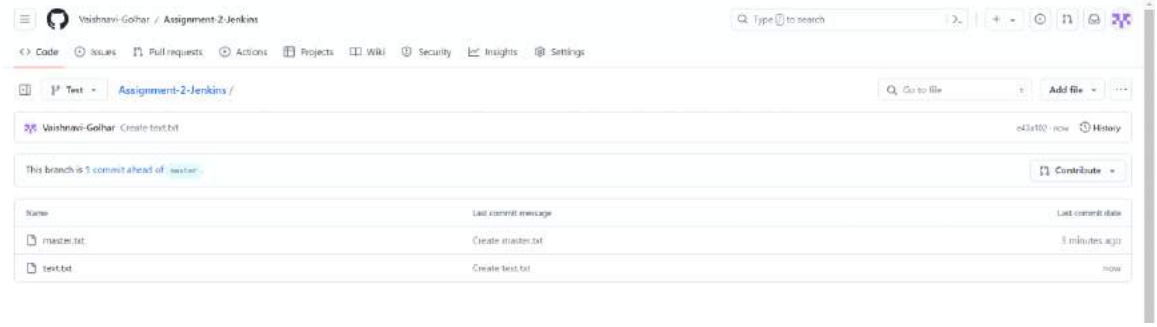
a. Push to test

b. Push to prod

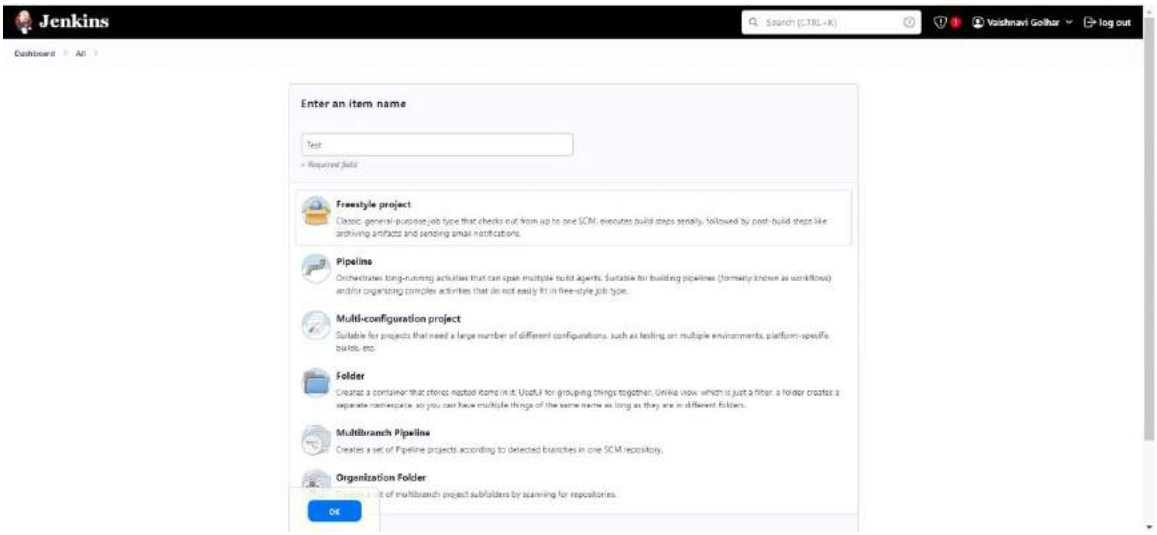
A. Create a Repository in Git Hub as “Assignment-2-Jenkins”

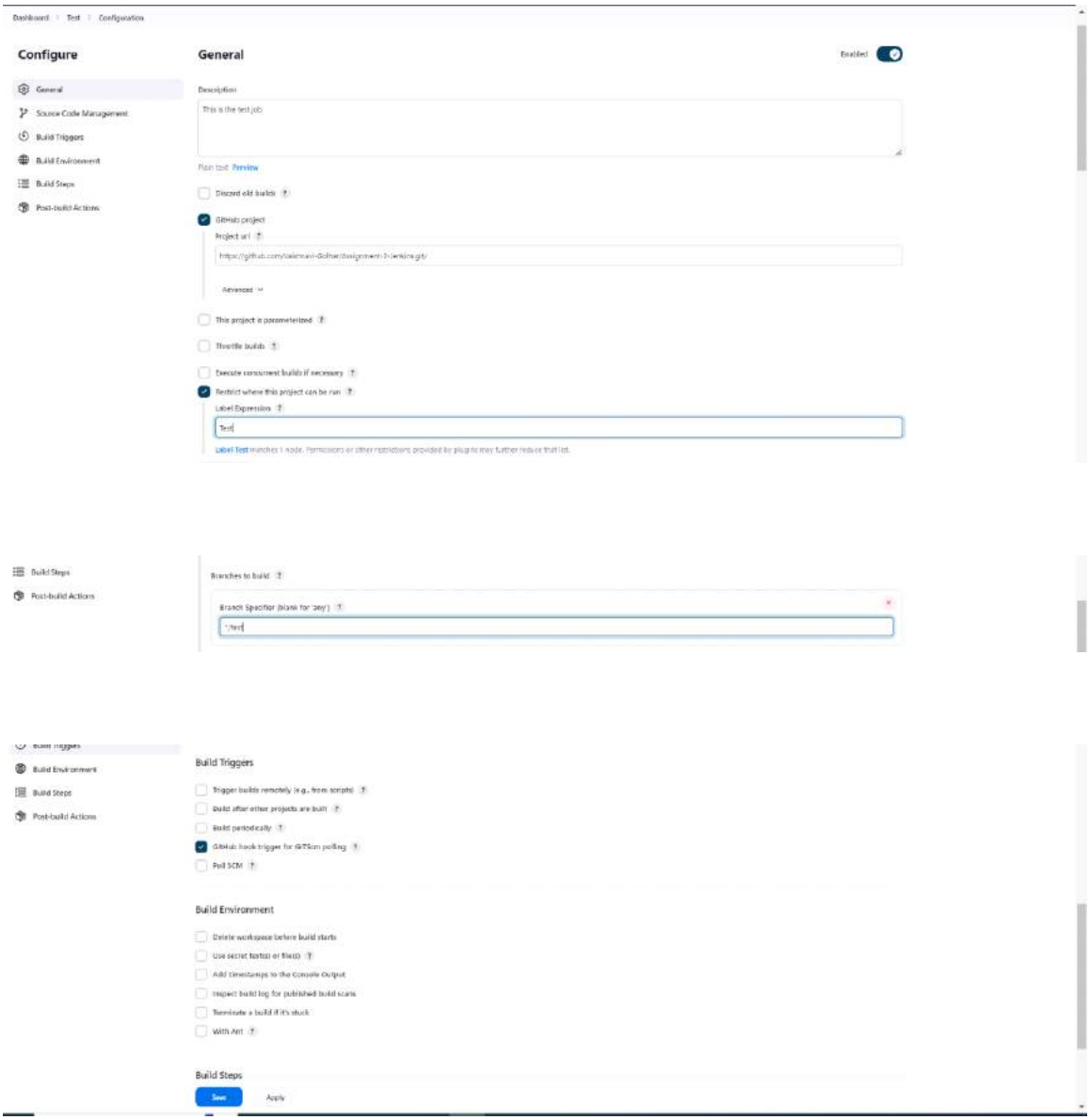


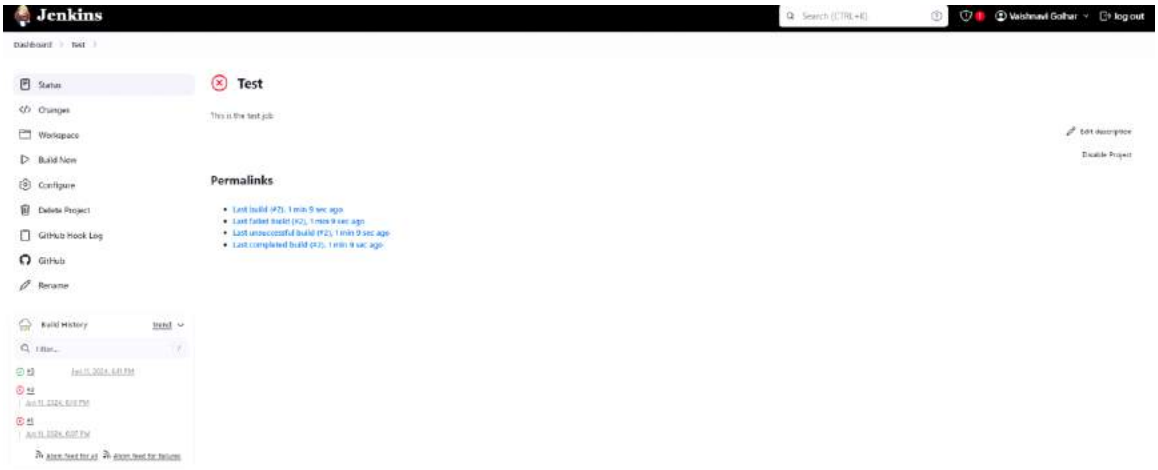


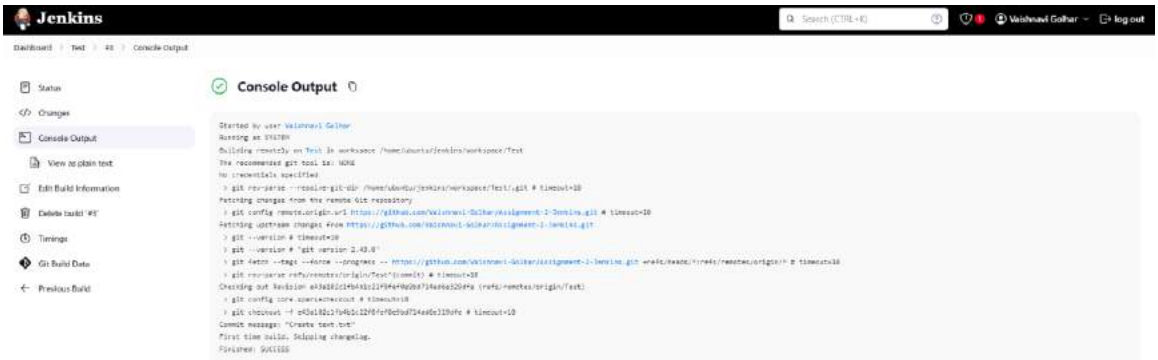


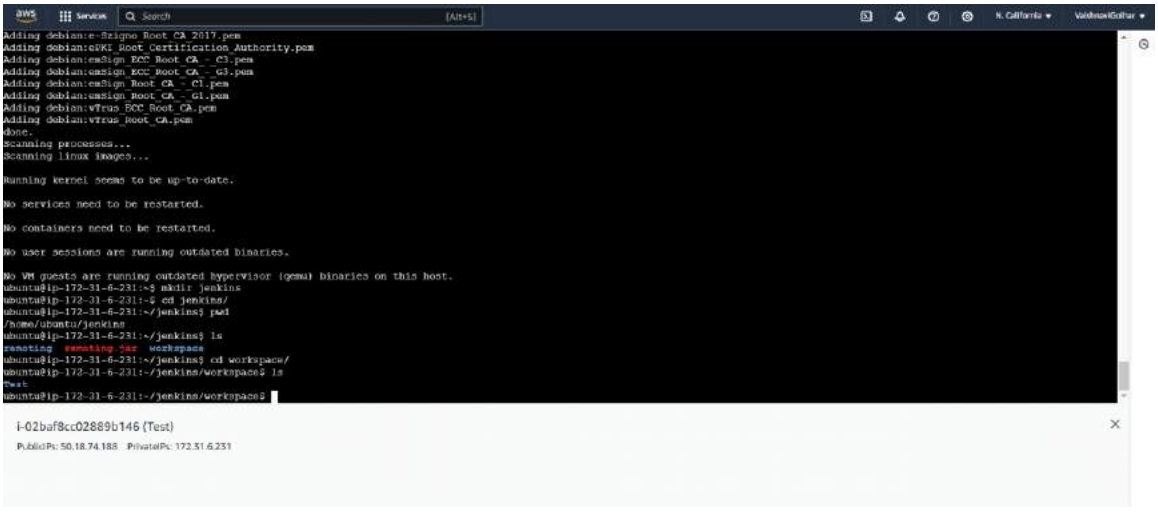
B. Create a Job for Pushing to Test Repo



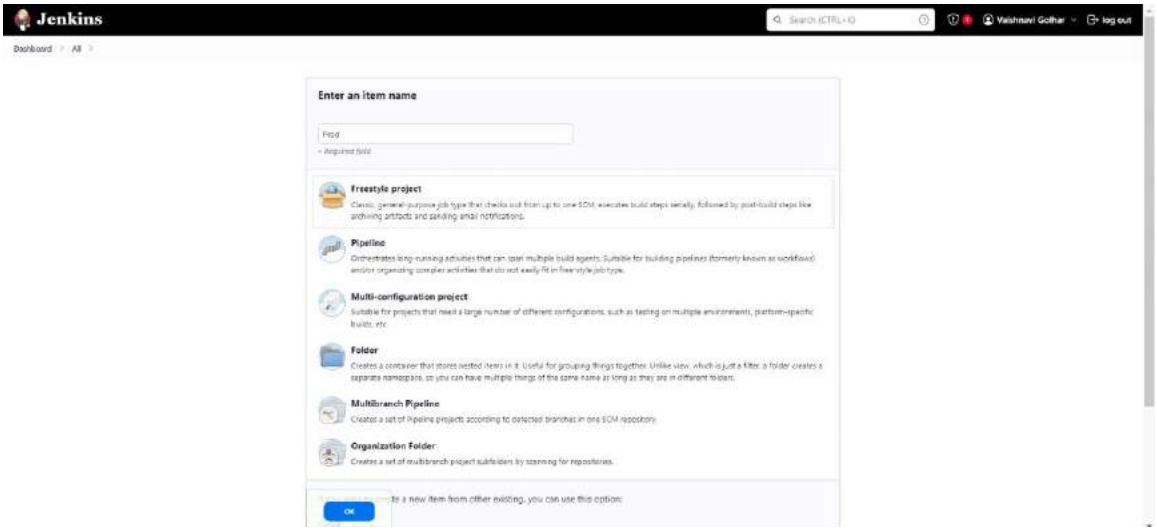

Dashboard > Test > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
This is the test job
Plain text Preview
Discard old builds
GitHub project
Project url
Advanced
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression
Test
Label Test matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.
Build Steps
Post-build Actions
Branches to build
Branch Specifier (blank for 'any')
*/test
Build Environment
Build Steps
Post-build Actions
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Build Environment
Delete workspace before build starts
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Save
Apply

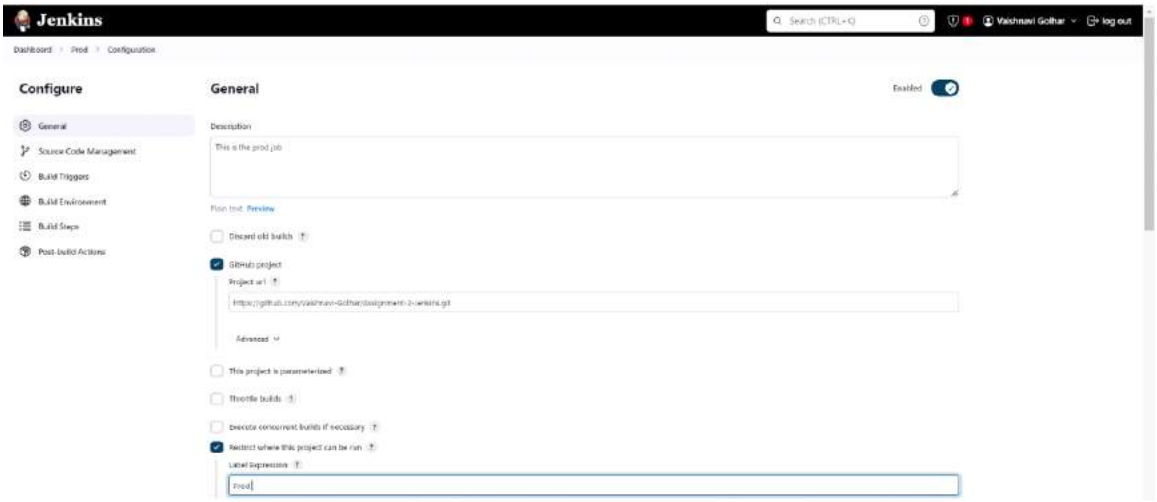


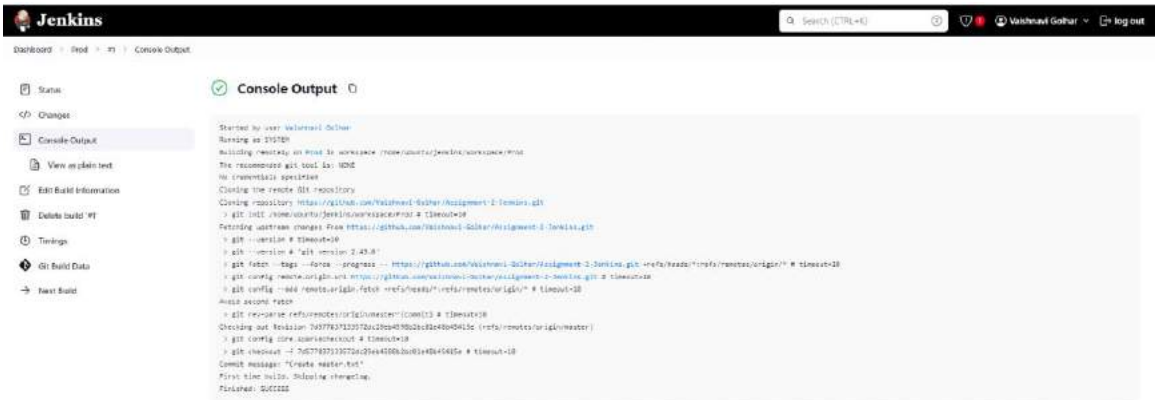




C) Create a Job for Pushing to Prod Repo


Jenkins
Search (CTRL+K)
Vaishnavi Golhar
log out
Dashboard > All >
Enter an item name
Prod
» Required field
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.
OK


Jenkins
Search (CTRL+K)
Vaishnavi Golhar
log out
Dashboard > Prod > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
This is the prod job
Plain text Preview
Discard old builds
GitHub project
Project url
https://github.com/Vaishnavi-Golhar/assignment-2-jenkins.git
Advanced
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression
Prod


Jenkins
Search (CTRL+K)
Vaishnavi Golhar
log out
Dashboard > Prod > #1 > Console Output
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'
Timings
Git Build Data
Next Build
Console Output
Started by user Vaishnavi Golhar
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Prod
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Vaishnavi-Golhar/Assignment-2-Jenkins.git
> git init /home/ubuntu/jenkins/workspace/Prod # timeout=10
Fetching upstream changes from https://github.com/Vaishnavi-Golhar/Assignment-2-Jenkins.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/Vaishnavi-Golhar/Assignment-2-Jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/Vaishnavi-Golhar/Assignment-2-Jenkins.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 7e377f637233572dc29e4598b2bc80e40d45415c (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7e377f637233572dc29e4598b2bc80e40d45415c # timeout=10
Commit message: "Create master.txt"
First time build. Skipping changelog.
Finished: SUCCESS

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-9-19:~$ mkdir jenkins
ubuntu@ip-172-31-9-19:~$ cd jenkins
ubuntu@ip-172-31-9-19:~/jenkins$ pwd
/home/ubuntu/jenkins
ubuntu@ip-172-31-9-19:~/jenkins$ ls
remoting remoting.jar workspace
ubuntu@ip-172-31-9-19:~/jenkins$ cd workspace/
ubuntu@ip-172-31-9-19:~/jenkins/workspace$ ls
Prod
ubuntu@ip-172-31-9-19:~/jenkins/workspace$ ls Prod
master.txt
ubuntu@ip-172-31-9-19:~/jenkins/workspace$
i-051a8011967b67e94 (Prod)
PublicIPs: 54.153.32.228 PrivateIPs: 172.31.9.19