Academic Year: 2025-26 **Semester: V**

Class / Branch: TEIT

Subject: DevOps Lab

Name of Instructor: Prof. Sujata Oak

Experiment No. 4

Aim: To install and configure Jenkins to test and deploy an application with Maven.

Theory:

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible." In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

Jenkins is a self-contained, open-source automation server which can be used to automate all sortsof tasks related to building, testing, and delivering or deploying software.

To install Jenkins following software packages are required
1) GIT (git-scm.com)
2) Notepad++ (https://notepad-plus-plus.org/downloads/)
3) Latest Java development kit (JDK)
4) Jenkins (https://www.jenkins.io/)
5) Apache Maven (Optional)

Installation Steps for Jenkins on Ubuntu 20.04

Step 1-: Update ubuntu repository

```
sujata@Ubuntu:~$ su root
Password:
```

#apt-get update

```
root@Ubuntu:/home/sujata# apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
```

```
Fetched 20.8 MB in 7s (3,072 kB/s)
Reading package lists... Done
```

Step 2-: Install Java development kit

#apt-get install openjdk-11-jdk

```
root@Ubuntu:/home/sujata# apt-get install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-6
  libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless
  x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
```

```
done.
Processing triggers for sgml-base (1.29.1) ...
Setting up x11proto-dev (2019.2-1ubuntu1) ...
Setting up libxau-dev:amd64 (1:1.0.9-0ubuntu1) ...
Setting up libice-dev:amd64 (2:1.0.10-0ubuntu1) ...
Setting up libsm-dev:amd64 (2:1.2.3-1) ...
Setting up libxdmcp-dev:amd64 (1:1.1.3-0ubuntu1) ...
Setting up x11proto-core-dev (2019.2-1ubuntu1) ...
Setting up libxcb1-dev:amd64 (1.14-2) ...
Setting up libx11-dev:amd64 (2:1.6.9-2ubuntu1.6) ...
Setting up libxt-dev:amd64 (1:1.1.5-1) ...
```

Step 3-: To test if Java has been installed successfully, run this command:

#java -version

```
root@Ubuntu:/home/sujata# java -version
openjdk version "11.0.23" 2024-04-16
OpenJDK Runtime Environment (build 11.0.23+9-post-Ubuntu-1ubuntu120.04.2)
OpenJDK 64-Bit Server VM (build 11.0.23+9-post-Ubuntu-1ubuntu120.04.2, mixed mod
e, sharing)
```

Step 4-: Install Jenkins

Now, we will install Jenkins itself. Issue the following four commands in sequence to initiate the installation from the Jenkins repository:

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null

```
root@Ubuntu:/home/sujata# curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
>   /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

```
root@Ubuntu:/home/sujata# echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
>   https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
>   /etc/apt/sources.list.d/jenkins.list > /dev/null
```

sudo apt-get update

```
root@Ubuntu:/home/sujata# sudo apt-get update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:4 https://pkg.jenkins.io/debian-stable binary/ Packages [27.3 kB]
Hit:5 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:8 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 30.1 kB in 1s (34.5 kB/s)
Reading package lists... Done
```

sudo apt-get install jenkins

```
root@Ubuntu:/home/sujata# sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
```

Step 5-: To check Jenkins installed or not:
jenkins --version

```
root@Ubuntu:/home/sujata# jenkins --version
2.452.3
```

Step 6-: Once that's done, start the Jenkins service with the following command:
sudo systemctl start jenkins.service

```
root@Ubuntu:/home/sujata# sudo systemctl start jenkins.service
```

To confirm its status, use:

#sudo systemctl status jenkins

If its WORKING it will show as active (running)

```
root@Ubuntu:/home/sujata# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-07-28 11:03:05 IST; 6min ago
   Main PID: 6542 (java)
      Tasks: 37 (limit: 4542)
     Memory: 411.9M
```

Press ctrl+z to exit

Step 7:- Adjust Firewall and Configuring Jenkins

With Jenkins installed, we can proceed with adjusting the firewall settings. By default, Jenkins will run on port **8080**.
In order to ensure that this port is accessible, we will need to configure the built-in **Ubuntu firewall** (ufw). To open the 8080 port and enable the firewall, use the following commands:

```
root@Ubuntu:/home/sujata# sudo ufw status
Status: inactive
```

```
root@Ubuntu:/home/sujata# sudo ufw allow 8080
Rules updated
Rules updated (v6)
```

```
root@Ubuntu:/home/sujata# sudo ufw enable
Firewall is active and enabled on system startup
```

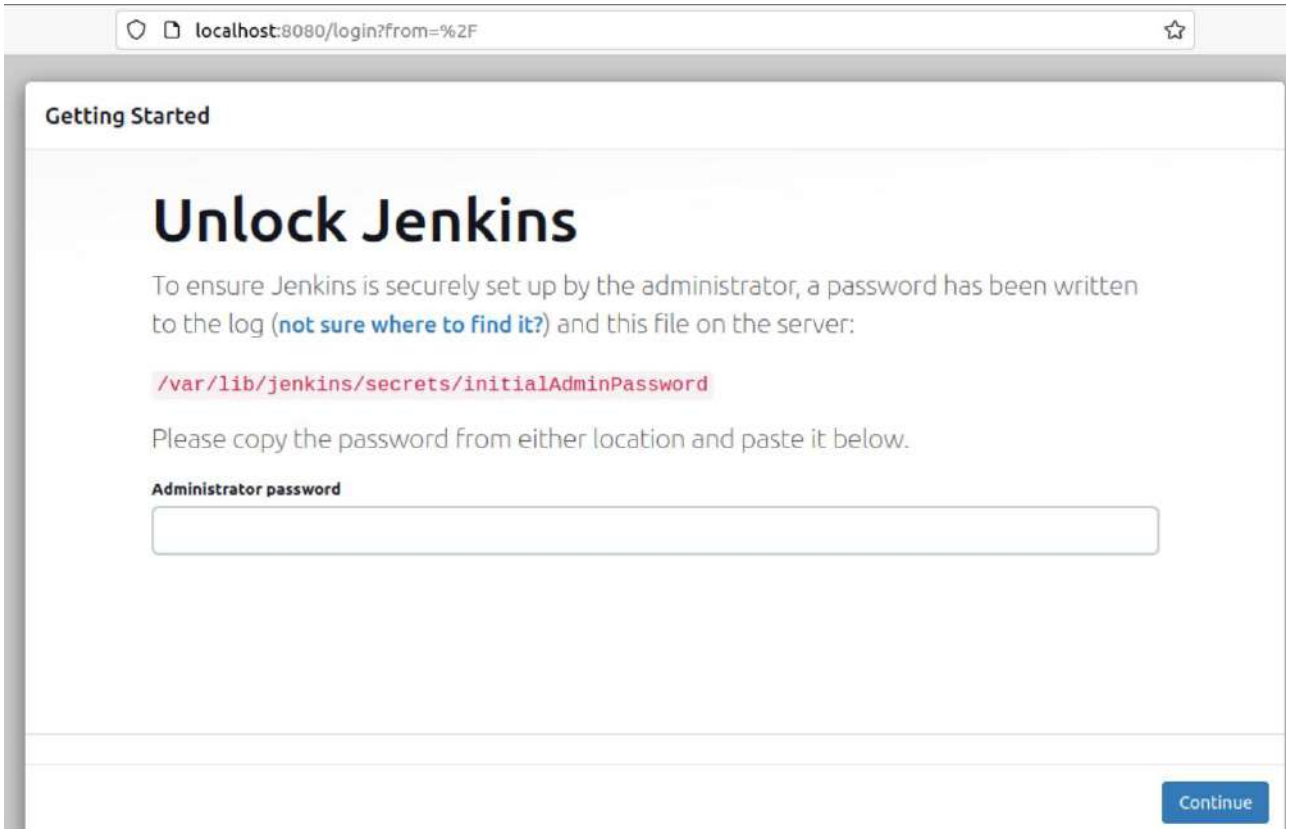
```
root@Ubuntu:/home/sujata# sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8080                       ALLOW       Anywhere
8080 (v6)                  ALLOW       Anywhere (v6)
```

Step 8-: With the firewall configured, it's time to set up Jenkins itself.

Once installation is done, you can test the application on http://localhost:8080 in browser **OR** **http://127.0.0.1:8080**

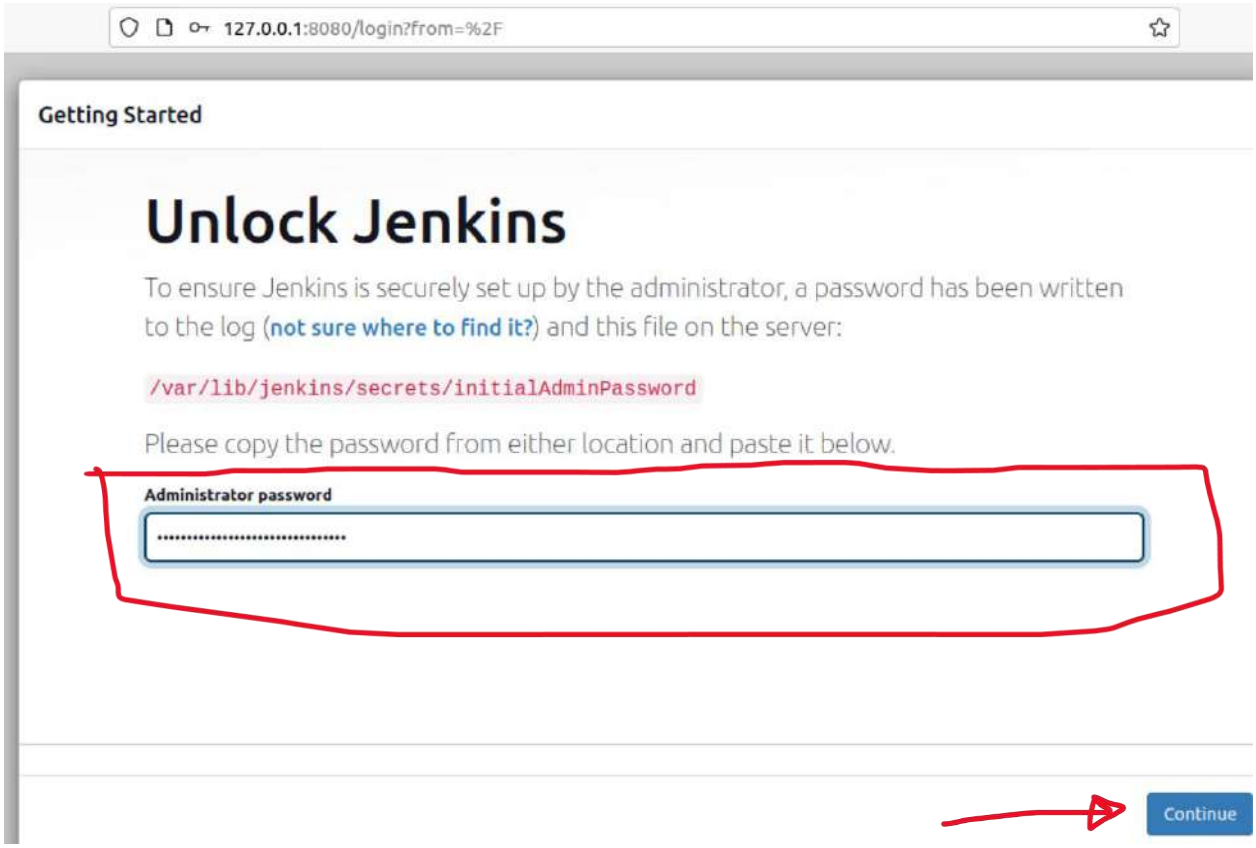
Type in the IP of your VPS along with the port number. The Jenkins setup wizard will be shown below:



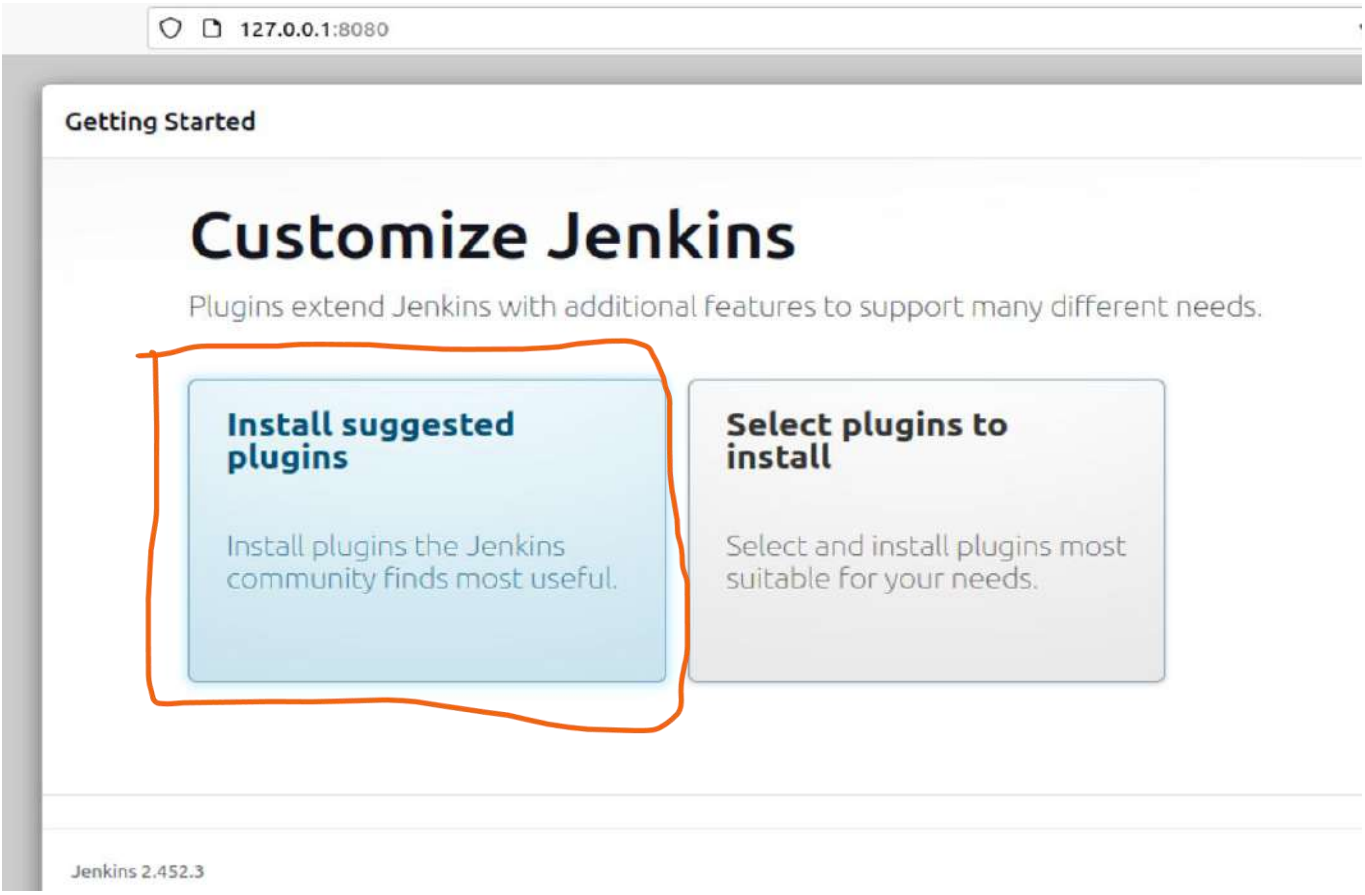
An administrator password will be needed to proceed with the configuration. It can be easily found inside the **/var/lib/jenkins/secrets/initialAdminPassword** file. To check the initial password, use the **cat command** as indicated below:

```
root@Ubuntu:/home/sujata# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
69e2d2c9995344768c72bfb4e28eba79
```

Copy the password, go back to the setup wizard, paste it and click **Continue**.

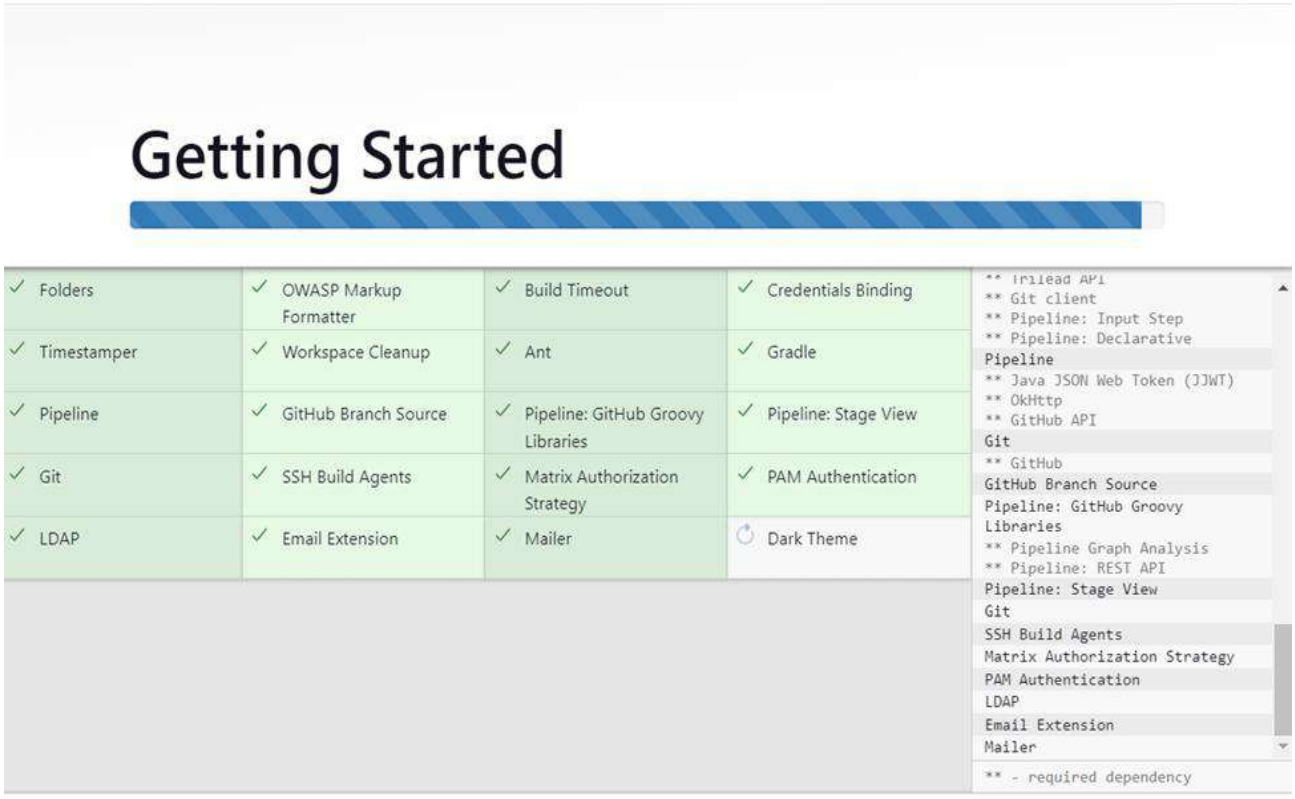


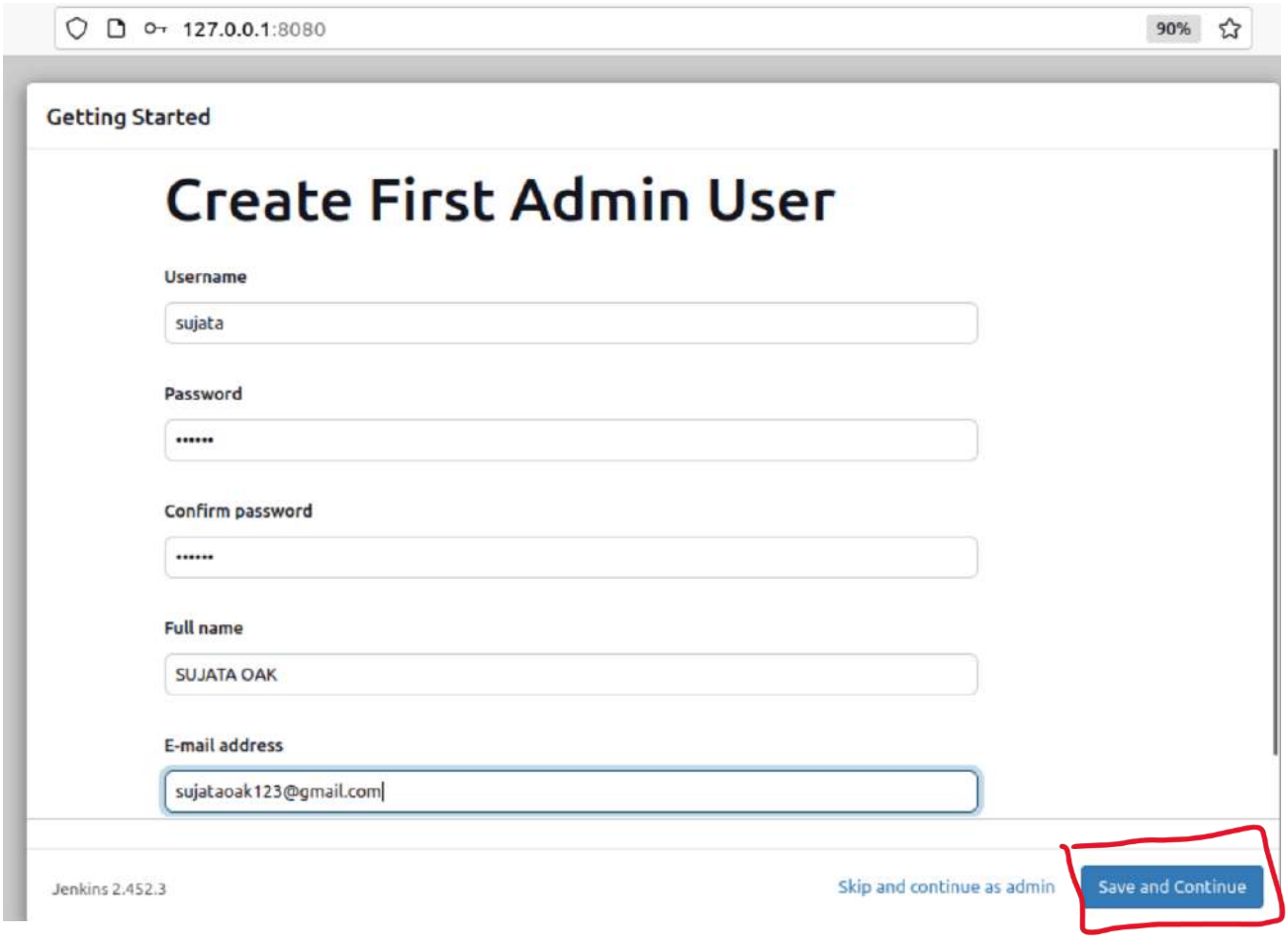
Step 9-: Customize Jenkins window will appear. We recommend simply selecting the **Install suggested plugins** option for this step.



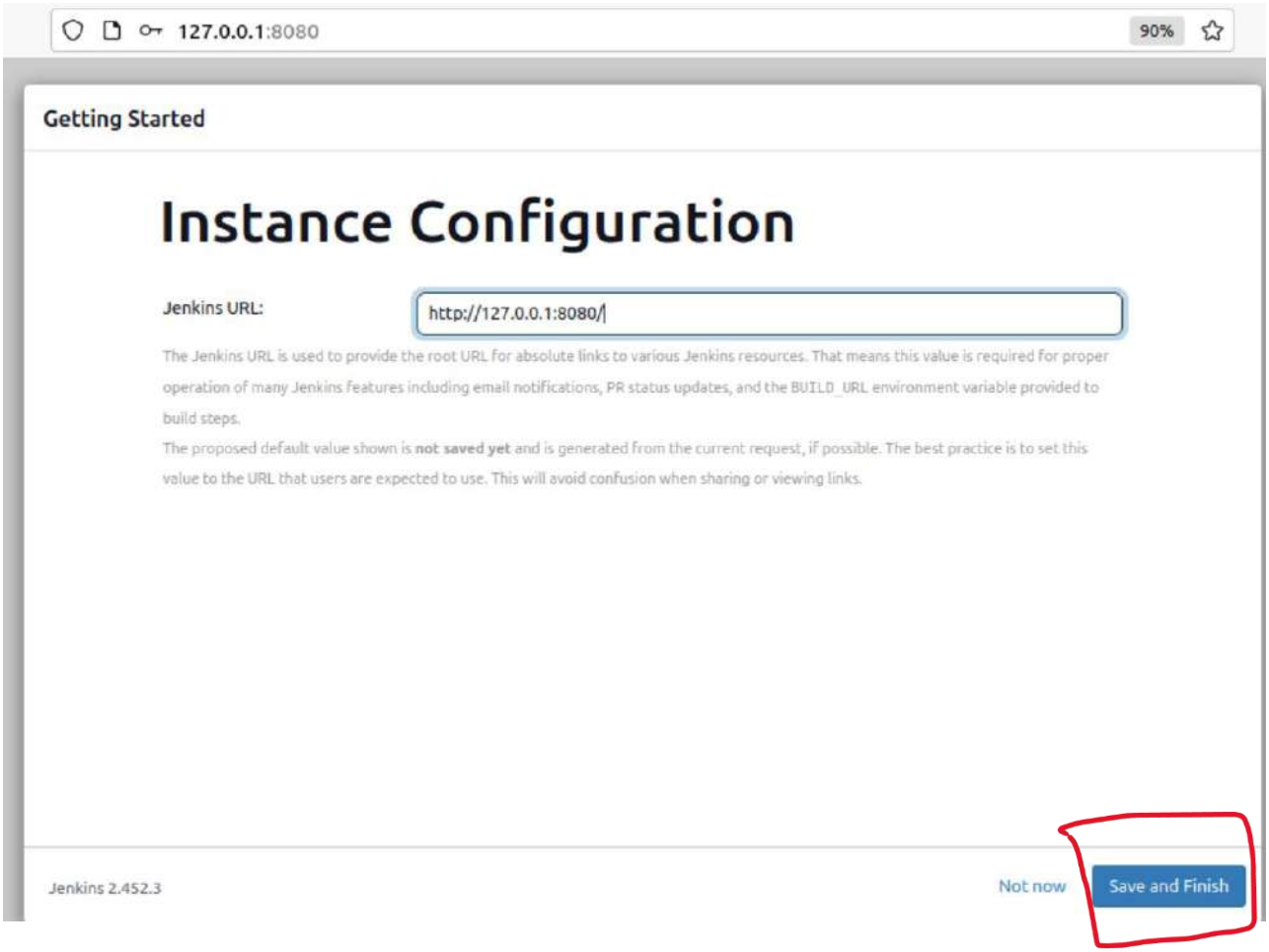
Step 10-: Give it a couple of minutes for the installation process to complete. Once it's done, specify your username, password, full name, and email address, and click on **Save and Continue** to create an admin user.

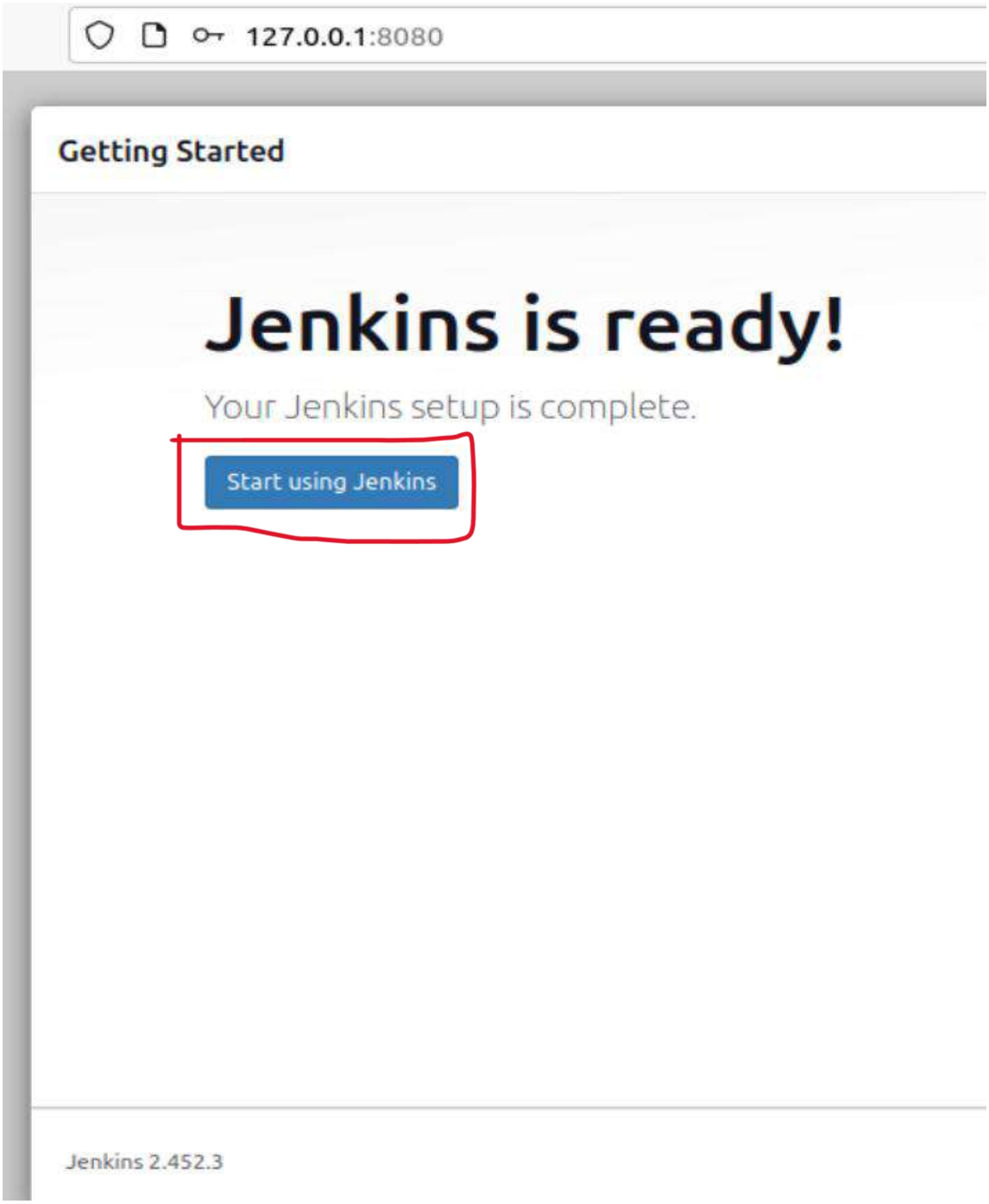
Getting Started



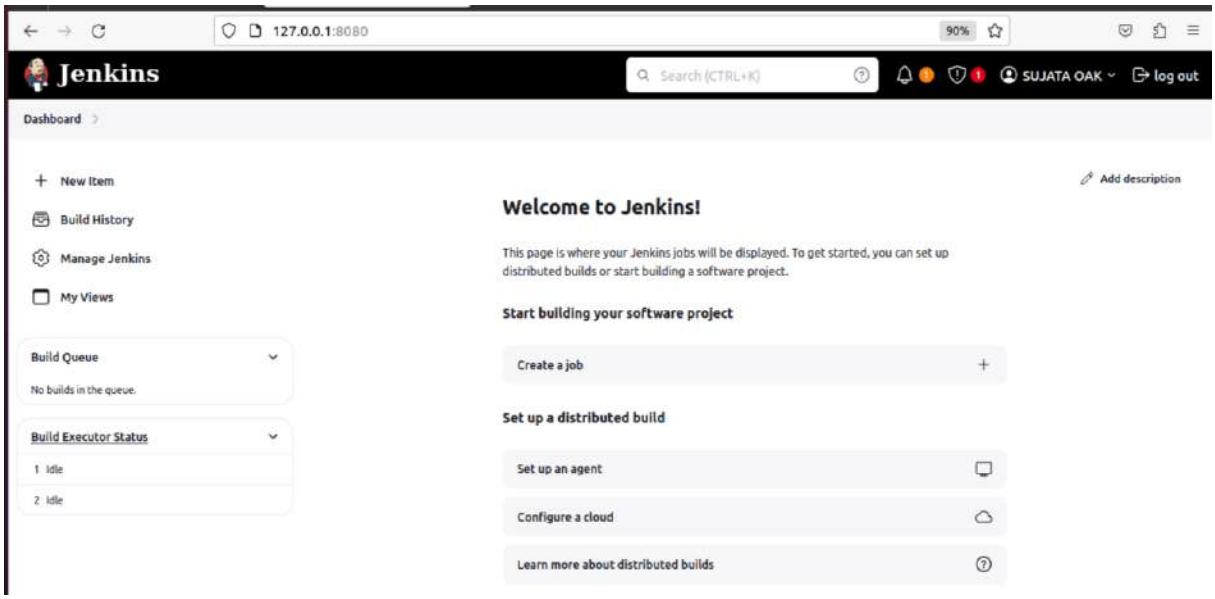


Then specify the preferred **Jenkins URL** and finish the configuration process.


127.0.0.1:8080
90%
Getting Started
Instance Configuration
Jenkins URL:
http://127.0.0.1:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.
The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
Jenkins 2.452.3
Not now
Save and Finish



Step 11-: After configuration, the Jenkins dashboard will appear, meaning the Jenkins server installation and initial setup were successful.

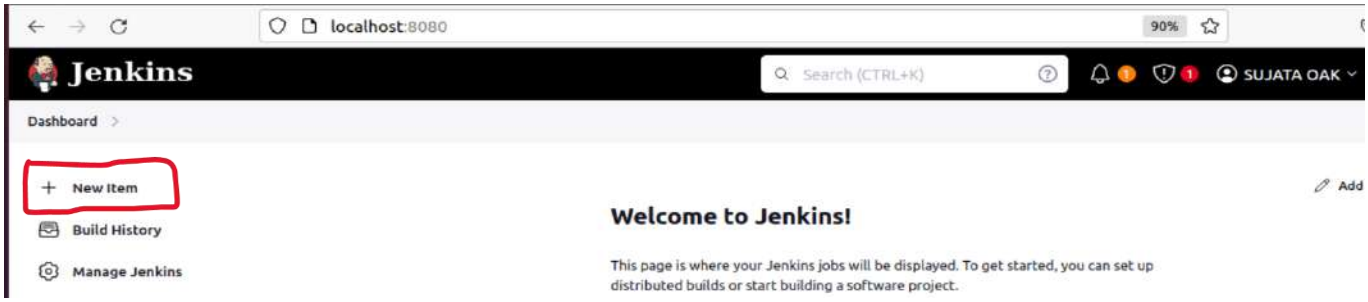

127.0.0.1:8080
90%
Jenkins
Search (CTRL+K)
SUJATA OAK
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds

Task to be performed as a part of this experiment:

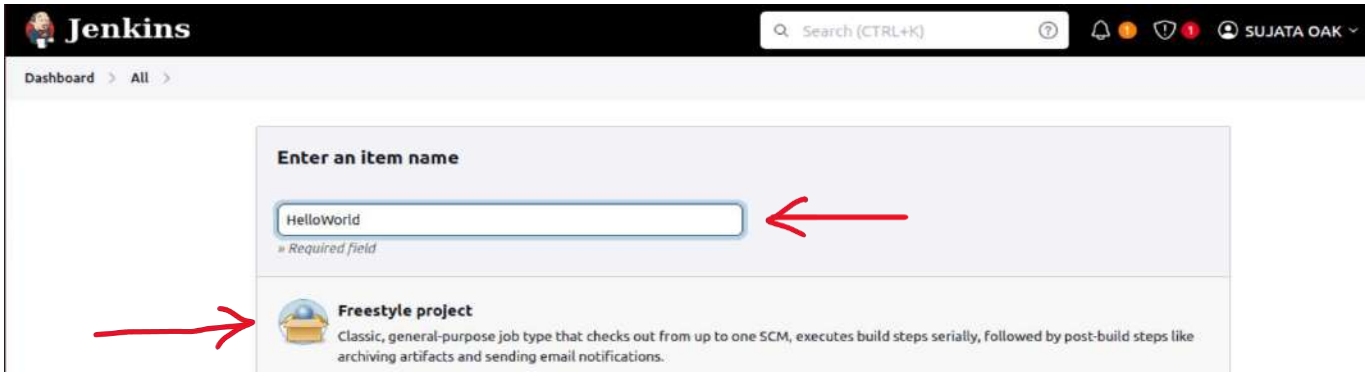
Jenkins Programming in Java using Version Control Git

Create and run a job in Jenkins for simple **HelloWorld** in Java

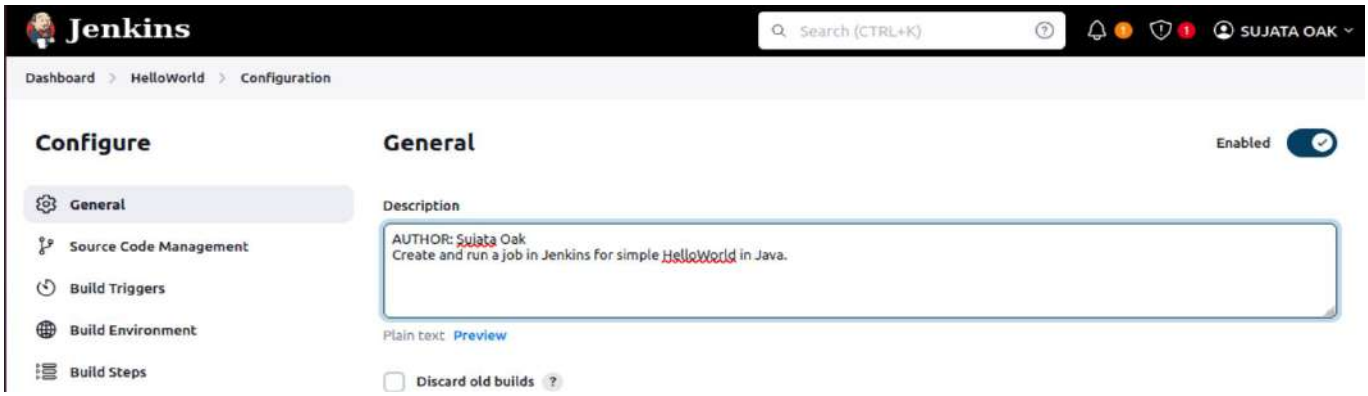
Step 1: Go to the Jenkins dashboard and click on the New Item.



Step 2: In the next page, enter the item name, and select the 'Freestyle project' option. And click OK. Here, my item name is HelloWorld.



Step 3: When you enter the OK, you will get a configuration page. Enter the details of the project in the Description section.



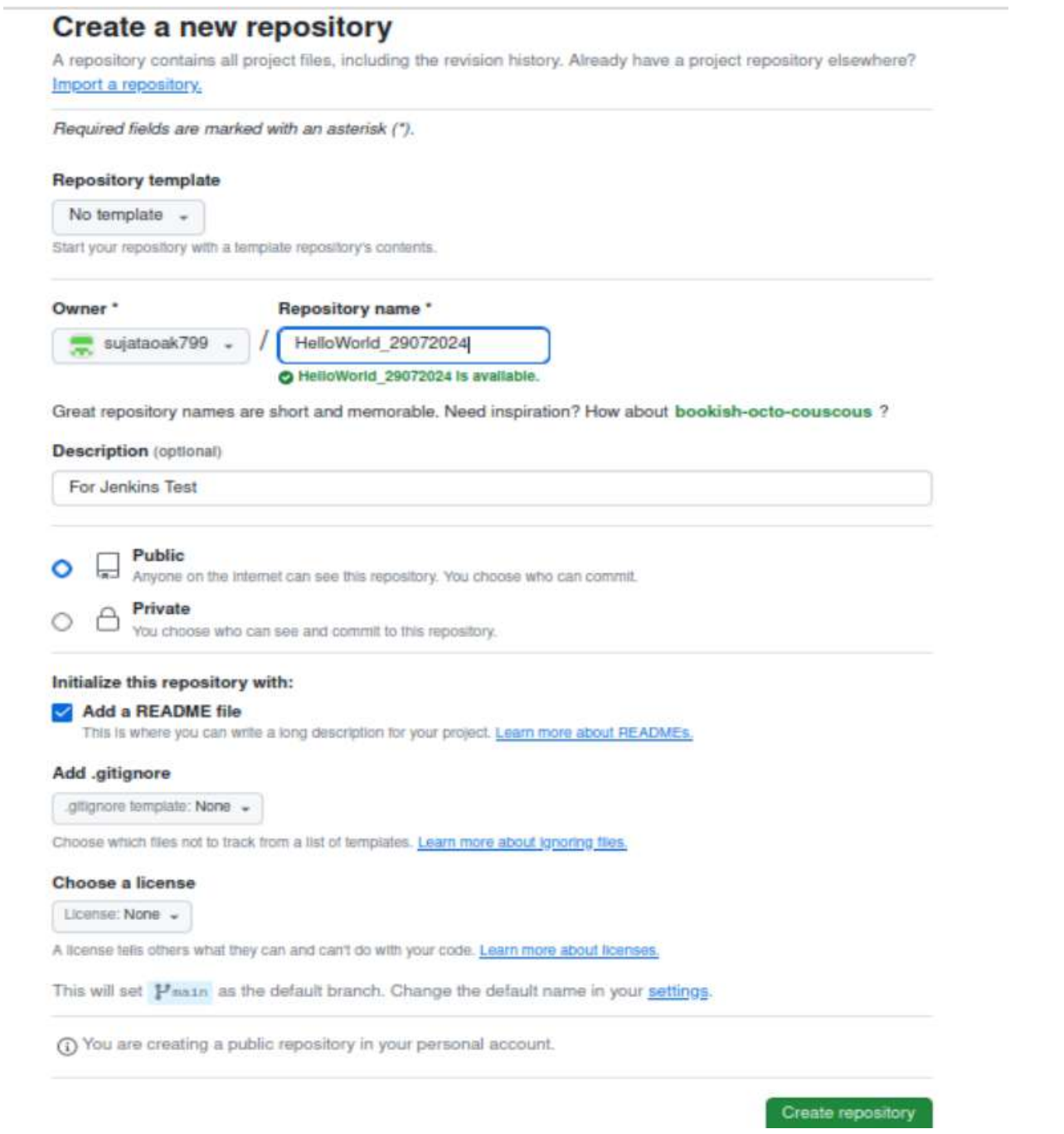
Step 4: On the Source Code Management section, select the **Git** option, and specify the Repository URL.

To do that you should have proper github setup on your system. To do the github setup:

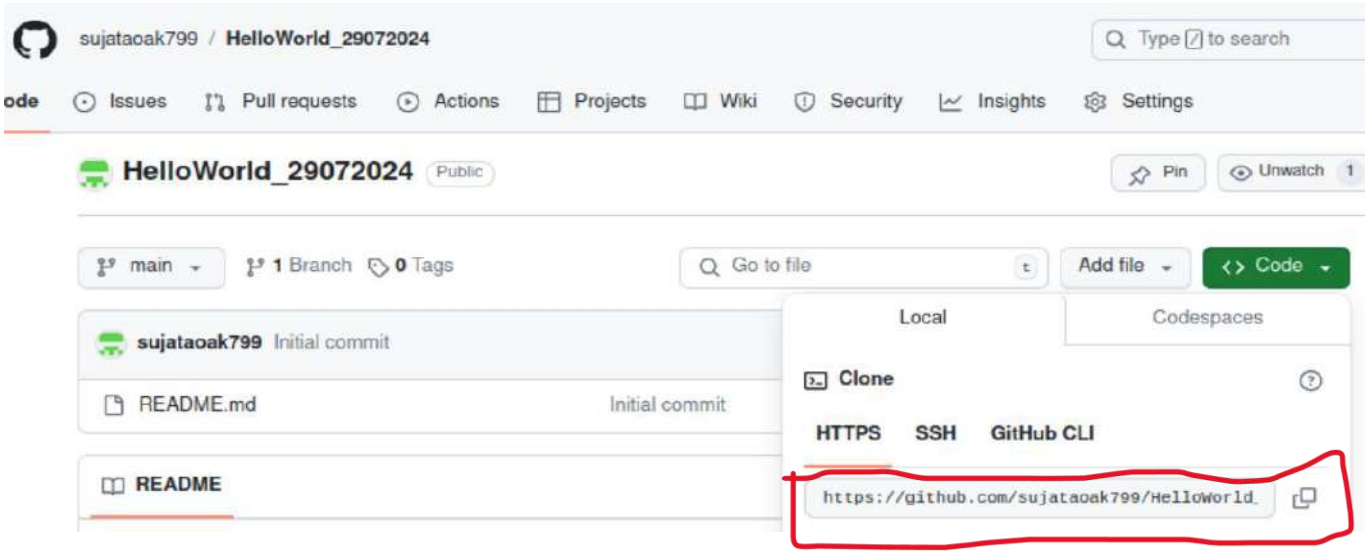
- First, you have to create a project in java. Here, I created a simple **HelloWorld** program and saved it to one folder i.e. **Desktop/JENKINS_LAB.** Compile the HelloWorld.java file.

```
sujata@Ubuntu:~/Desktop$ cd JENKINS_LAB/
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ gedit Simple.java
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ javac Simple.java
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ java Simple
Hello Java
```

- Now create a project in your GitHub account and give the Repository name. Here my repository name is HelloWorld_29072024.



- Click on **Create repository**.



Your repository is created. Copy the repository URL. My repository URL is: https://github.com/sujataoak799/HelloWorld_29072024.git

- Open the command prompt in your Ubuntu and go to the path where your java file is created.
- Then run the following command.

  git init

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git init
Initialized empty Git repository in /home/sujata/Desktop/JENKINS_LAB/.git/
```

  git status

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Jenkins-installation-link.odt
        Simple.class
        Simple.java
```

git add .

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git add .
```

git status

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Jenkins-installation-link.odt
        new file:   Simple.class
        new file:   Simple.java
```

- Configure your GitHub account in your system.

1. git config --global user.email "your@email"
2. git config --global user.name "username"

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git config --global user.name "sujataoak799"
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git config --global user.email "sujataoak2021@gmail.com"
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git config --list
user.name=sujataoak799
user.email=sujataoak2021@gmail.com
```

- Commit it and add the repository URL.

1. git commit -m "Added HelloWorld Java Program"

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git commit -m "Added HelloWorld Java Program"
[master (root-commit) b38b097] Added HelloWorld Java Program
 3 files changed, 5 insertions(+)
 create mode 100644 Jenkins-installation-link.odt
 create mode 100644 Simple.class
 create mode 100644 Simple.java
```

2. git remote add origin https://github.com/sujataoak799/HelloWorld_29072024.git

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git remote add origin https://github.com/sujataoak799/HelloWorld_2907
2024.git
```

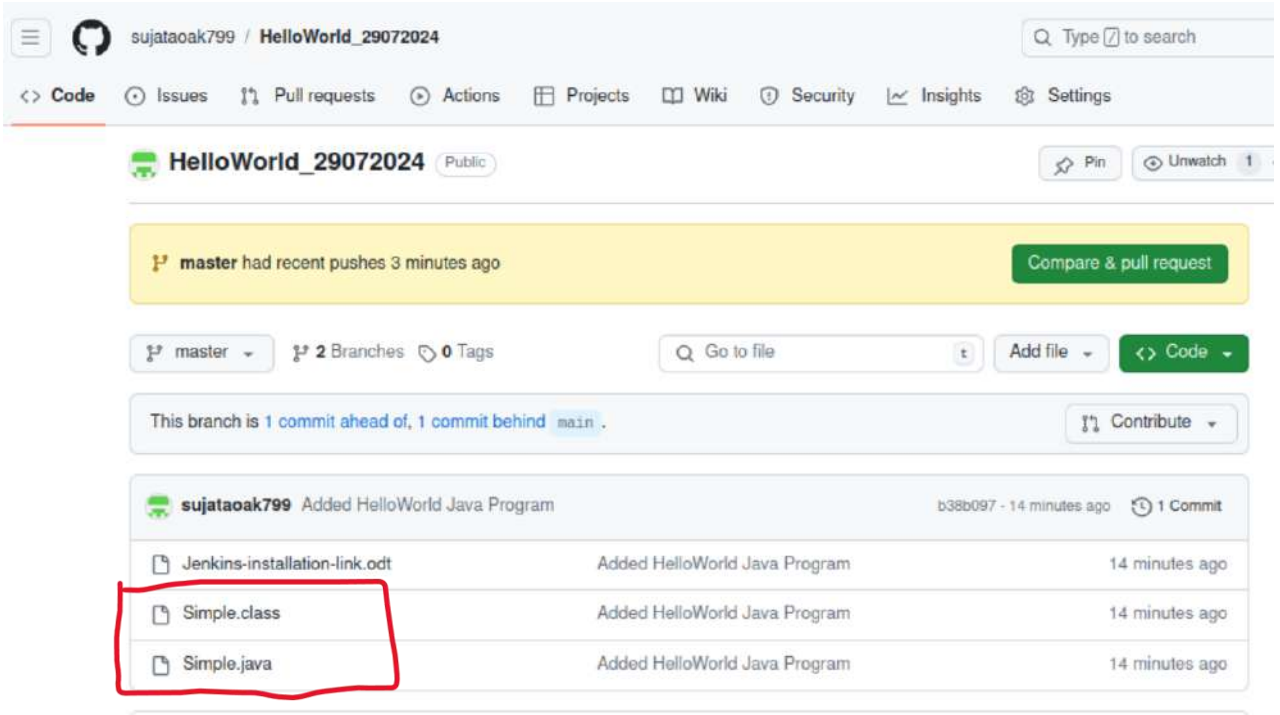
3. git push -u origin master

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git push -u origin master
Username for 'https://github.com': sujataoak799
Password for 'https://sujataoak799@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#
cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/sujataoak799/HelloWorld_29072024.git/'
```

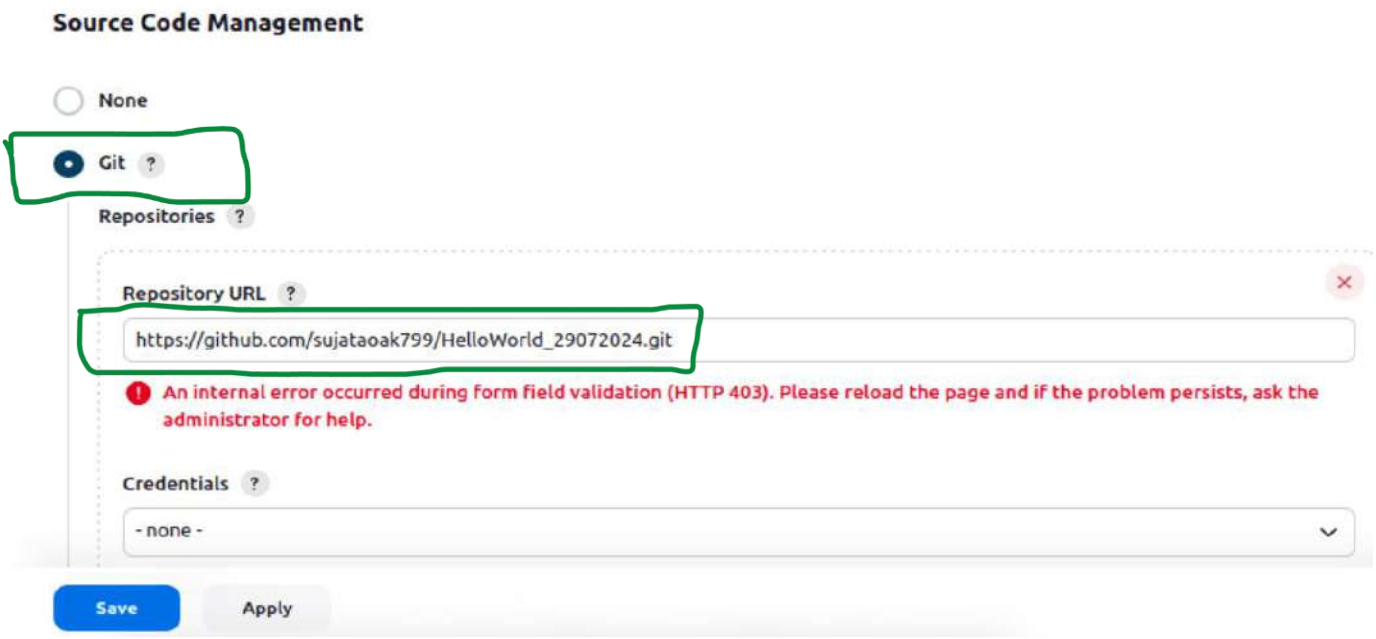
```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git remote set-url origin https://ghp_ZaLvsbTPgxILguomA7MJ2Wwa52NpCX3
iwWij@github.com/sujataoak799/HelloWorld_29072024.git
```

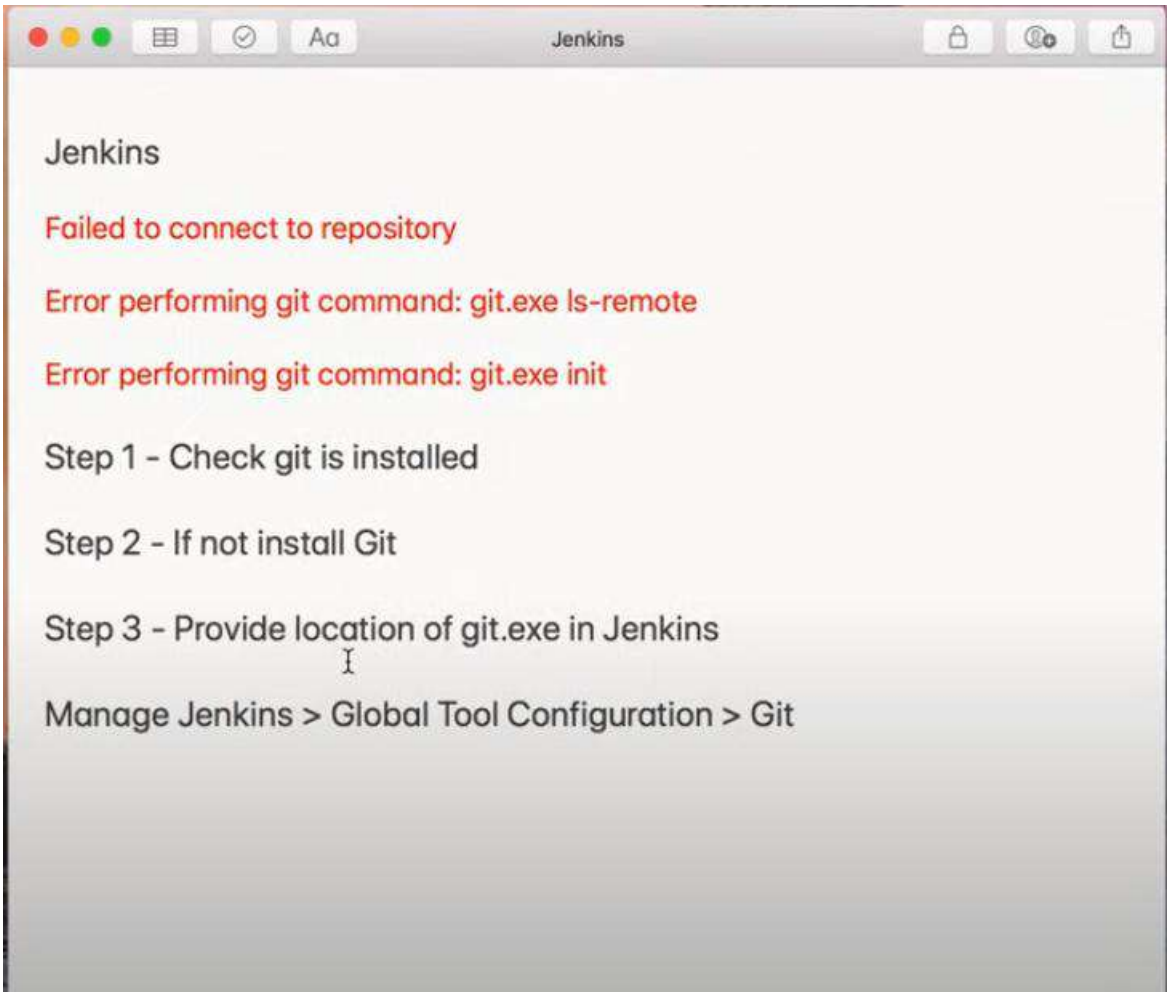
```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 8.19 KiB | 8.19 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/sujataoak799/HelloWorld_29072024/pull/new/master
remote:
To https://github.com/sujataoak799/HelloWorld_29072024.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

- Now, when you refresh your GitHub account, the HelloWorld file will be added in your repository.

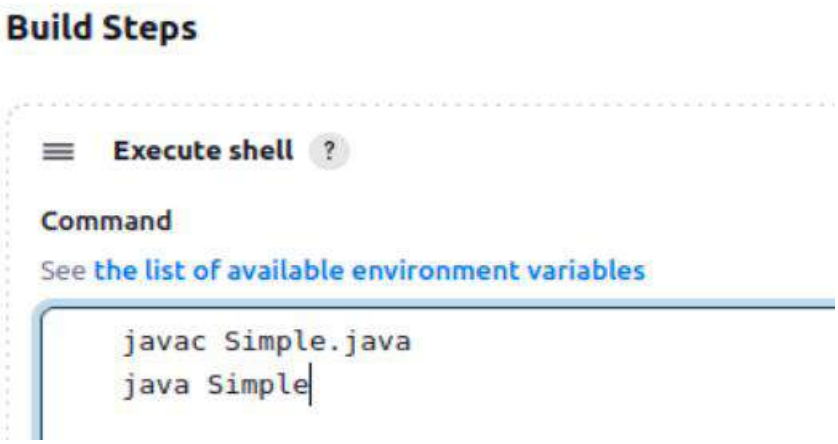


Step 5: Add the Repository URL in the **Source Code Management** section.


Source Code Management
None
Git
Repositories
Repository URL
https://github.com/sujataoak799/HelloWorld_29072024.git
An internal error occurred during form field validation (HTTP 403). Please reload the page and if the problem persists, ask the administrator for help.
Credentials
- none -
Save
Apply



Step 6: Now, it is time to build the code. Click on "**Add build step**" and select the "**Execute Shell**".



Step 7: Enter the following command to compile the java code.

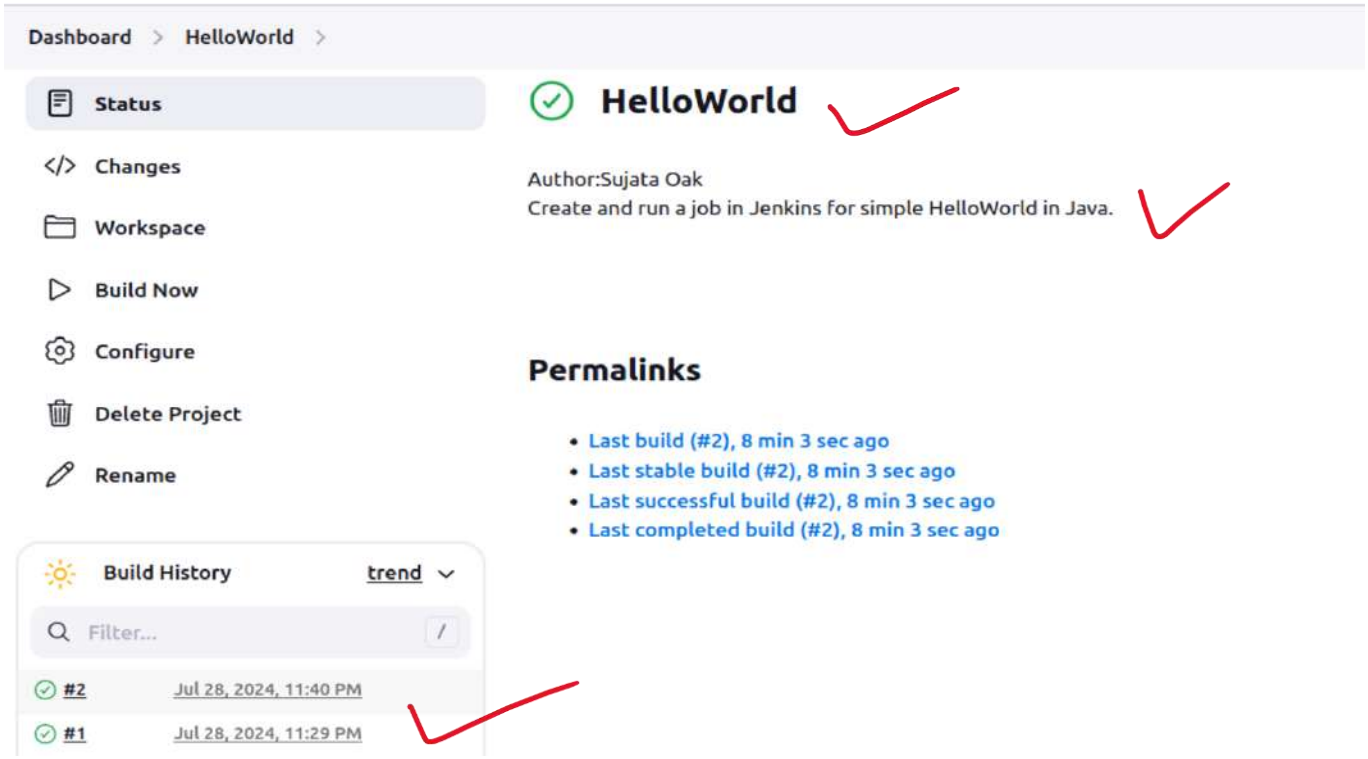
```
javac Simple.java
java Simple
```

Step 8: Click Apply and then Save button.

Step 9: Once you saved the configuration, then now can click on **Build Now** option.

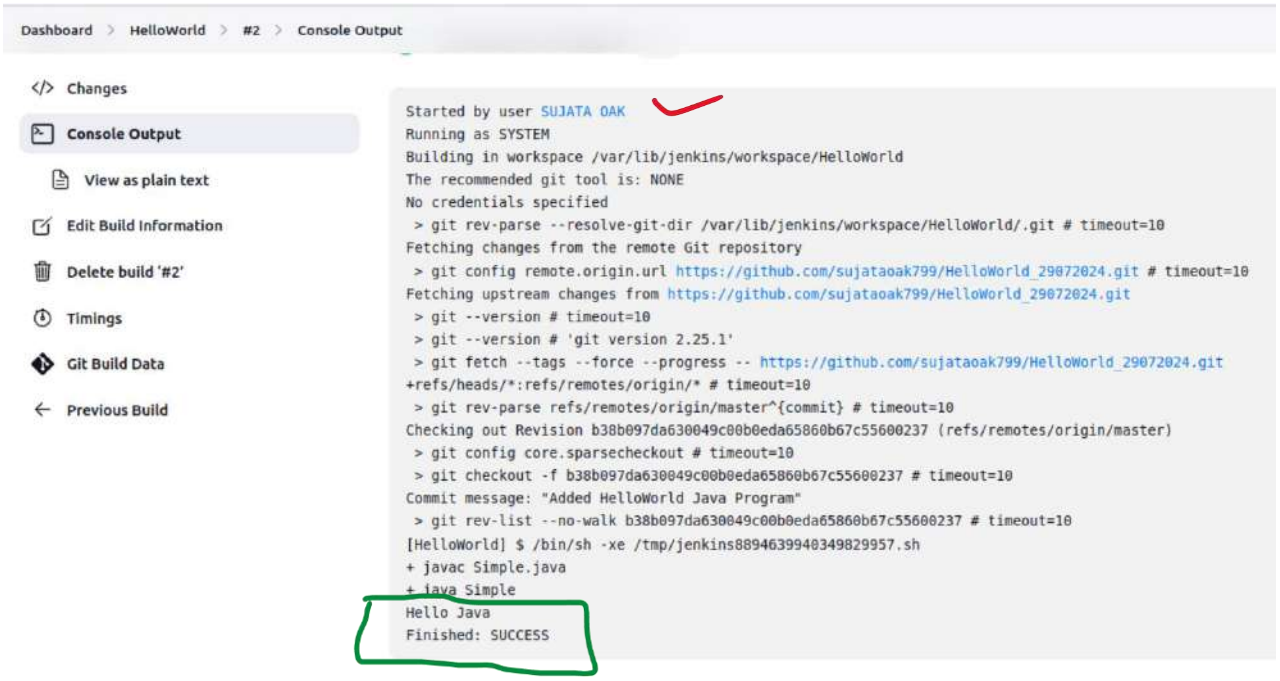
Step 10: After clicking on **Build Now**, you can see the status of the build on the Build History section.

Once the build is completed, a status of the build will show if the build was successful or not. If the build is failed then it will show in red color. Blue symbol is for success.



Click on the build number **#2** in the **Build History section** to see the details of the build.

Step 11: Click on **Console Output** from the left side of the screen to see the status of the build you run. It should show the success message.



Conclusion: In this experiment we understood the installation and implemented the use case of Jenkins in version control system.