**120A3051**

**Shreya Idate**

**Batch: E3**

**Experiment No: 5**

**AIM**:

1. To set up and build a Java, Maven /Ant and Python jobs in Jenkins.
2. To build the pipeline of jobs using Maven / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.

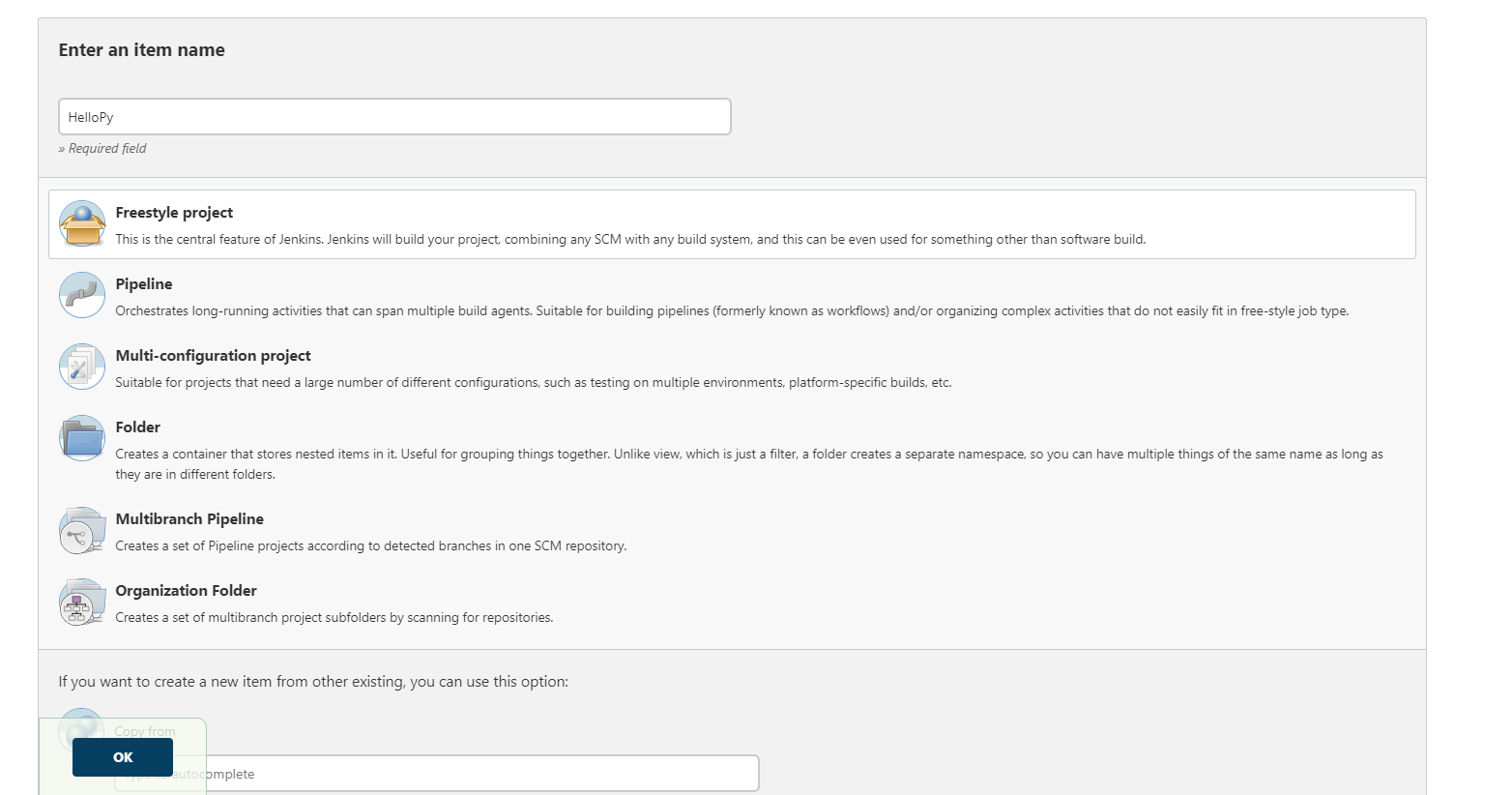
**THEORY**:

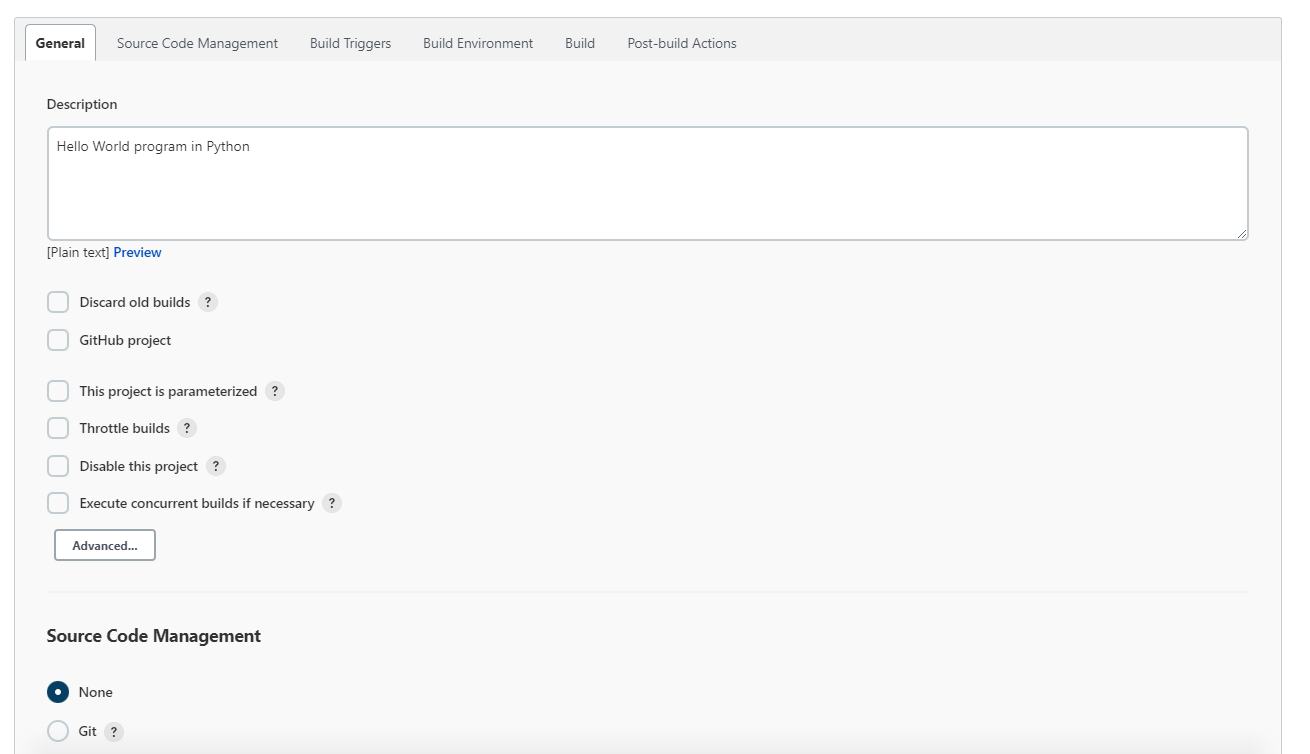
Jenkins is a free and open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

1. To set up and build a Java, Maven /Ant and Python jobs in Jenkins.

**Steps for Creating and Testing Python job in Jenkins**:

* 1. Install python and set environment settings on our machine.
  2. Install Python plugins in Jenkins i.e. Manage Jenkins-> Manage Plugins -> Available-> Python Plugin -> select the plugin and click on install without restart.
  3. Create a new job as freestyle project in Jenkins. New Item - > Freestyle Project. Add some optional description.





* 1. Configure -> Build section, add build steps-> commands to run the py script. Implicit =

>select Execute python script => write some implicit python script here -> save and apply. Go to step 5. Explicit => write the following Build steps, assuming your python script is stored in this location D:\python scripts

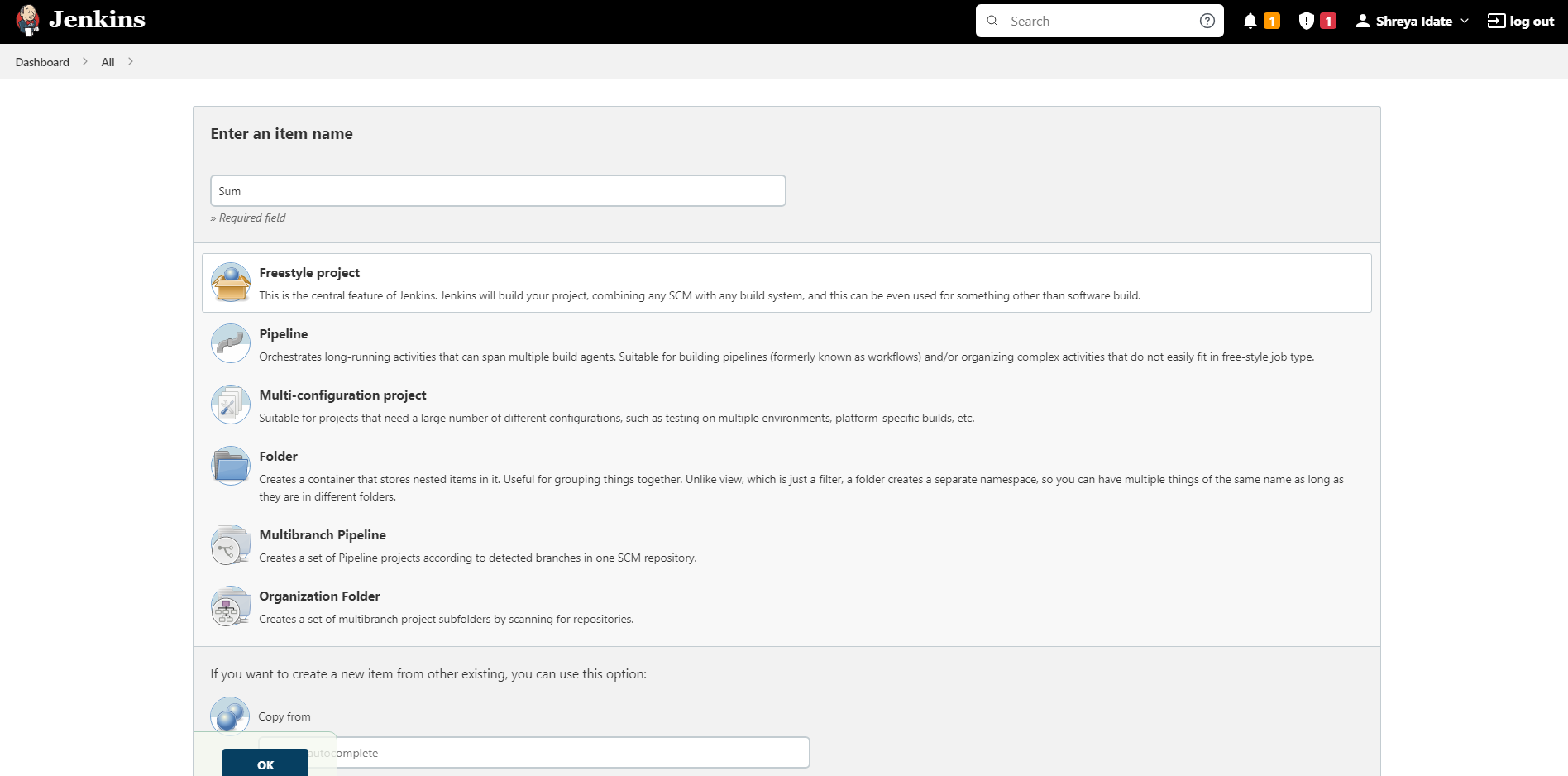


* 1. Build Now - > Console Output



**Steps for Testing Java job in Jenkins**:

1. Install Java and set environment settings.
2. Create a new job as freestyle project.



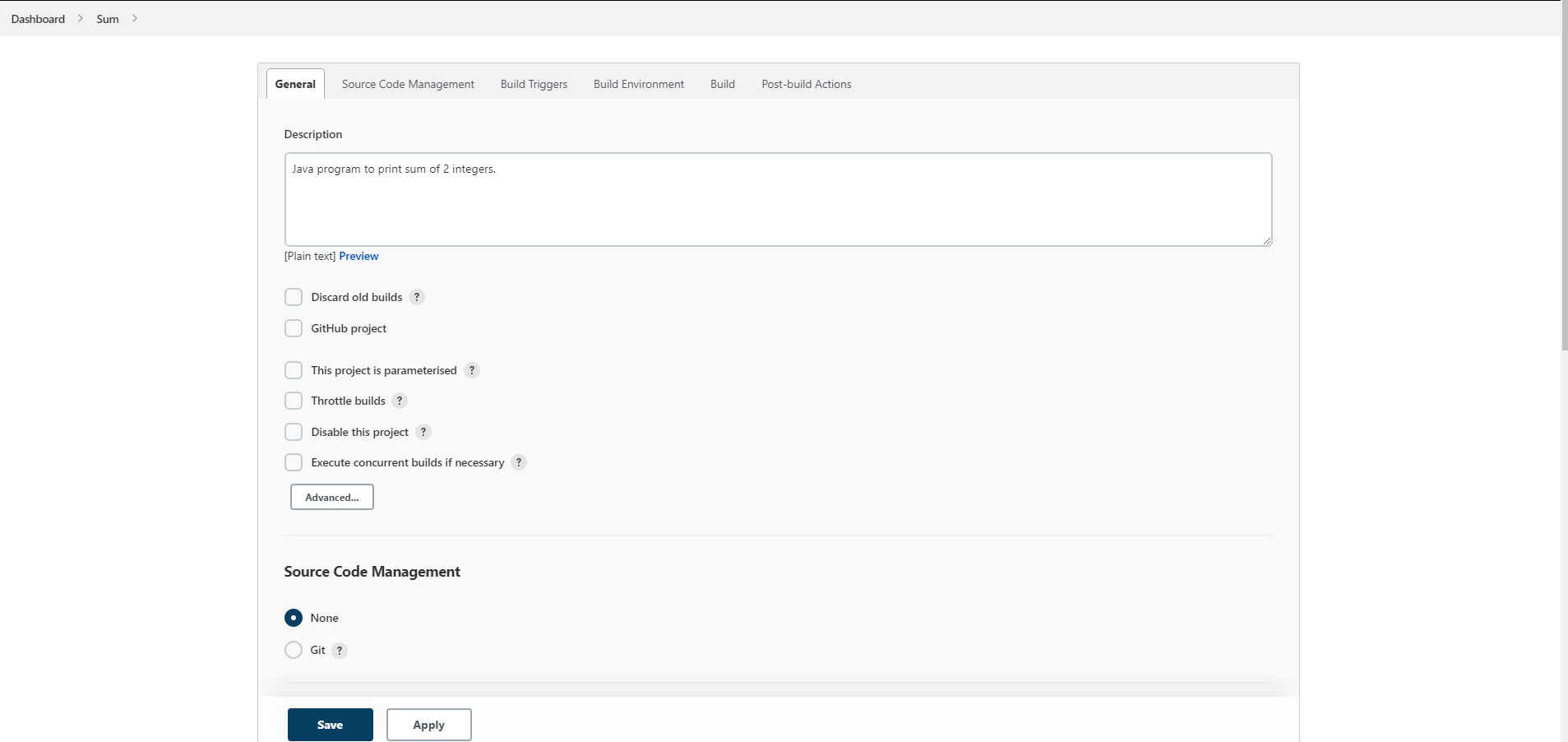
1. Configure ->Build section, commands to run the Java program. Add build steps-> select Windows batch commands. Go to Configure -> Build section, write the following Build steps, assuming your java program is stored in this location D:\ java program

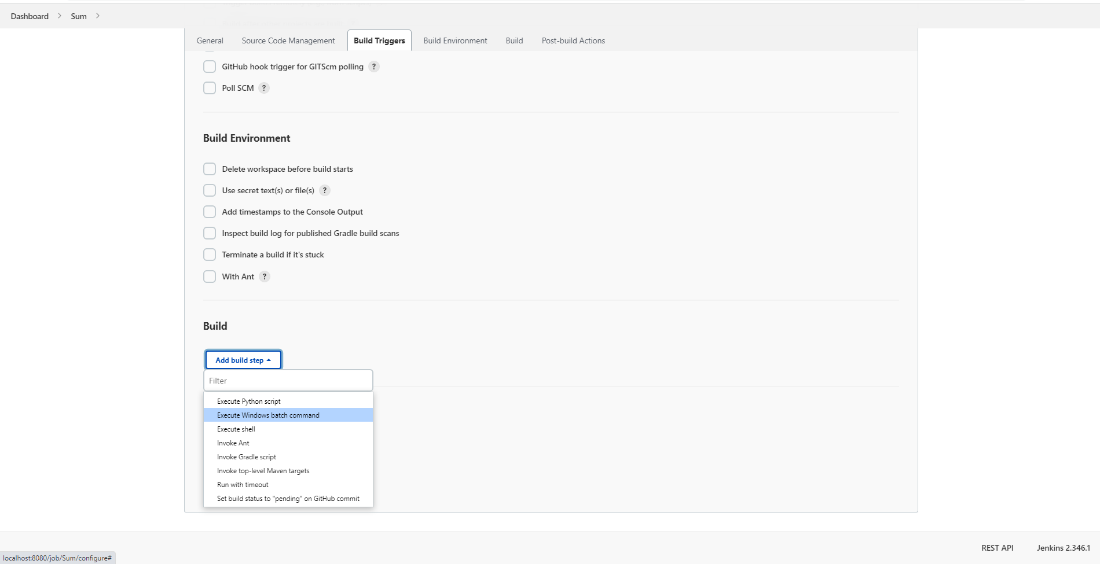
###### E:

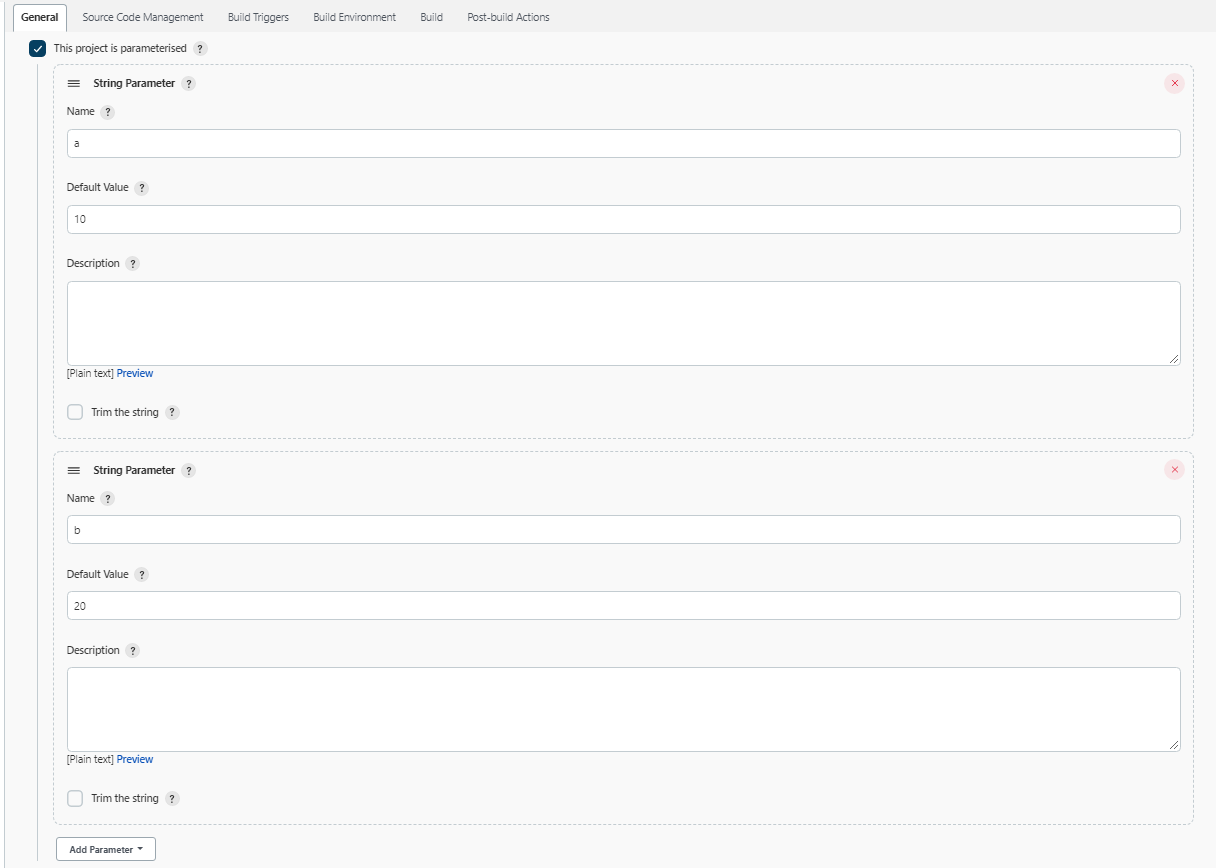
**javac sum.java**

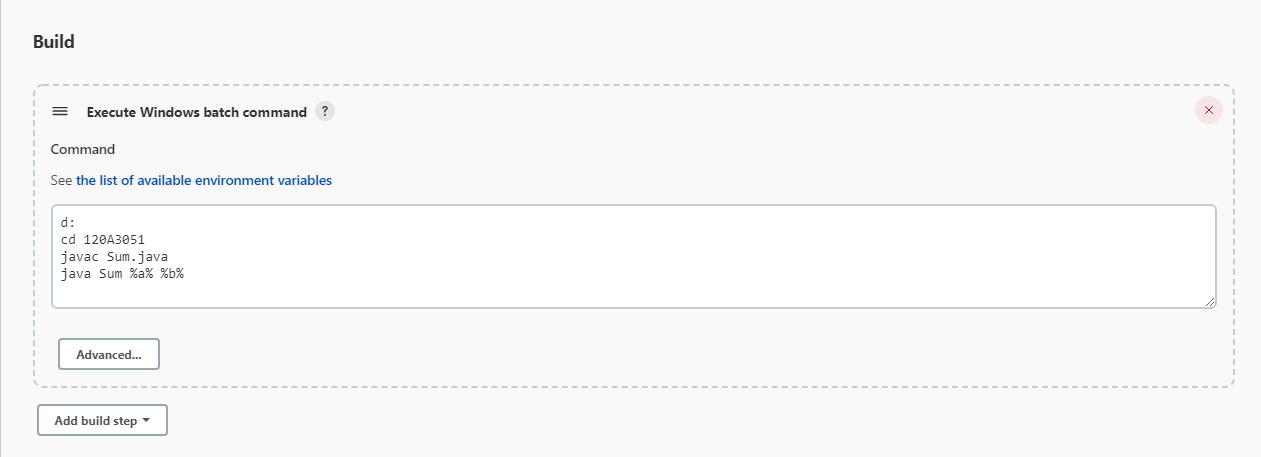
###### java sum %a% %b%

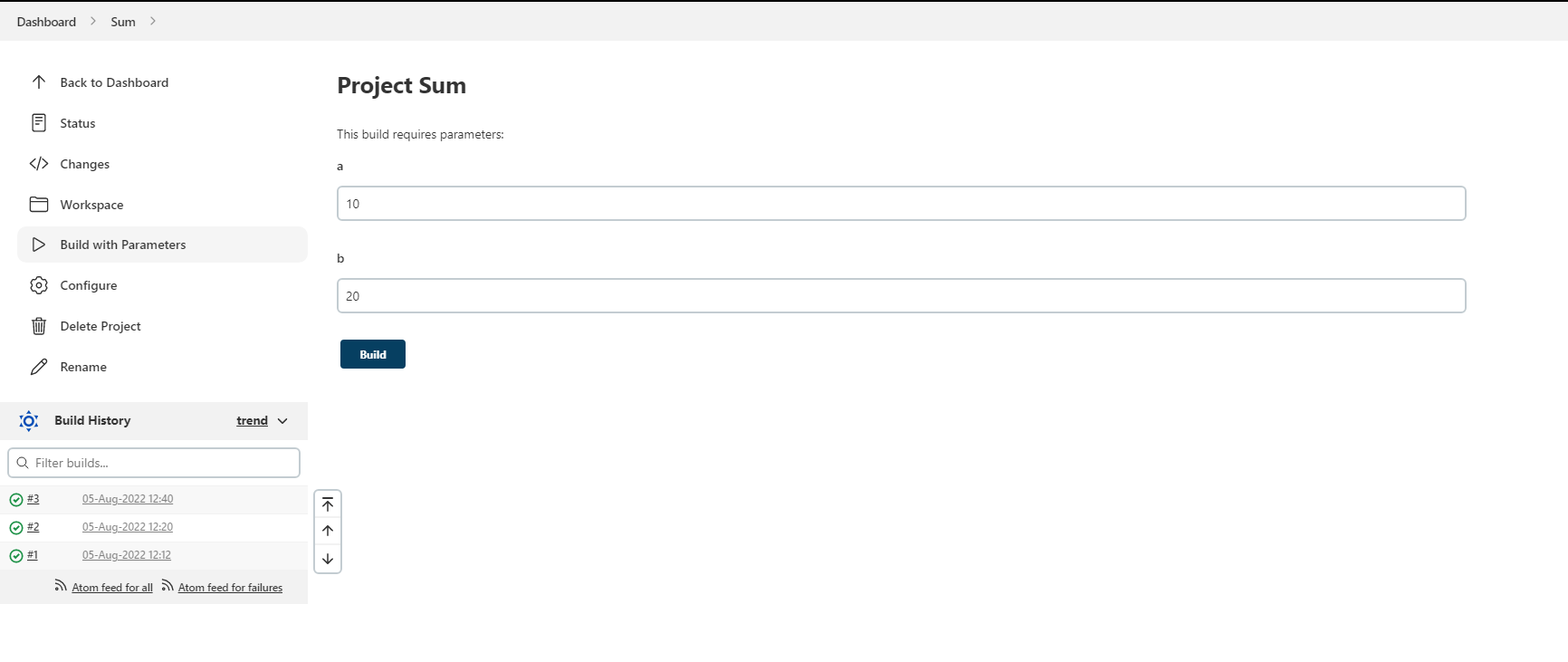
-> save and apply

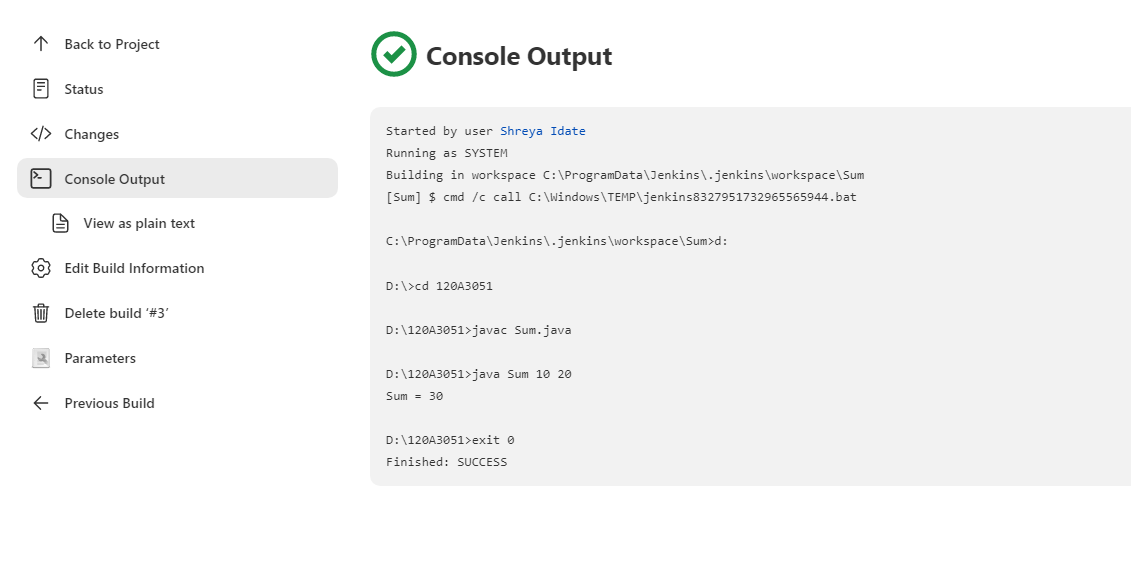








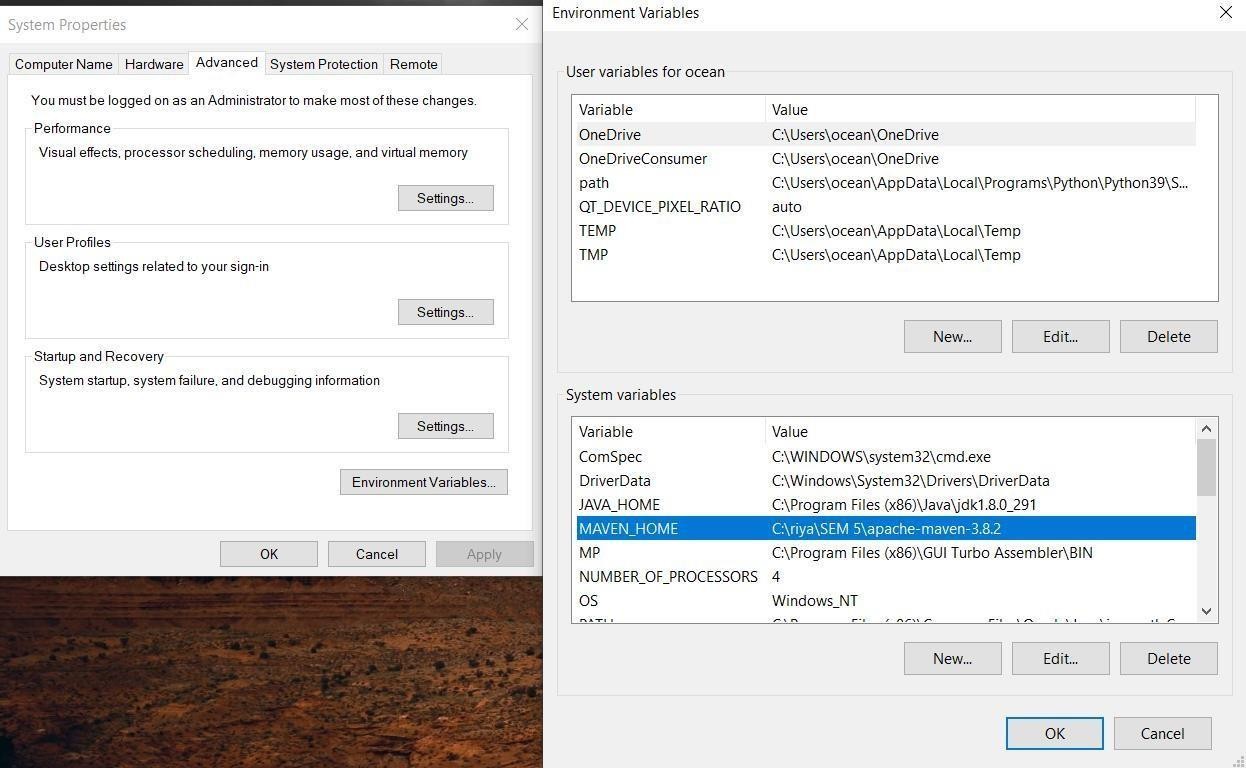




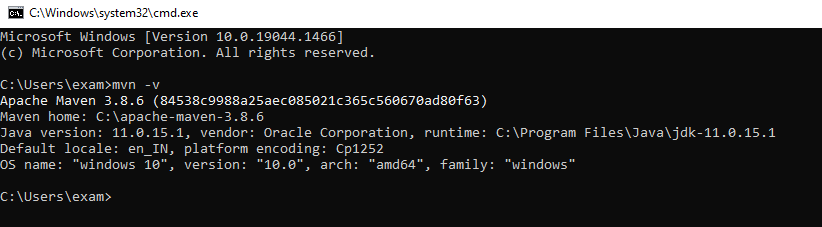
***Steps for Building Maven job from the GitHub remote repository into Jenkins*:**

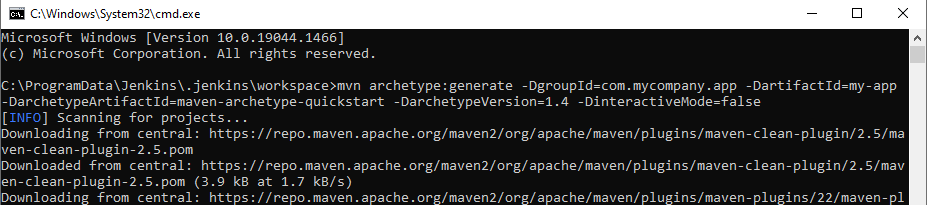
1. Install Maven and set environment settings. Make sure you've previously set java settings properly.

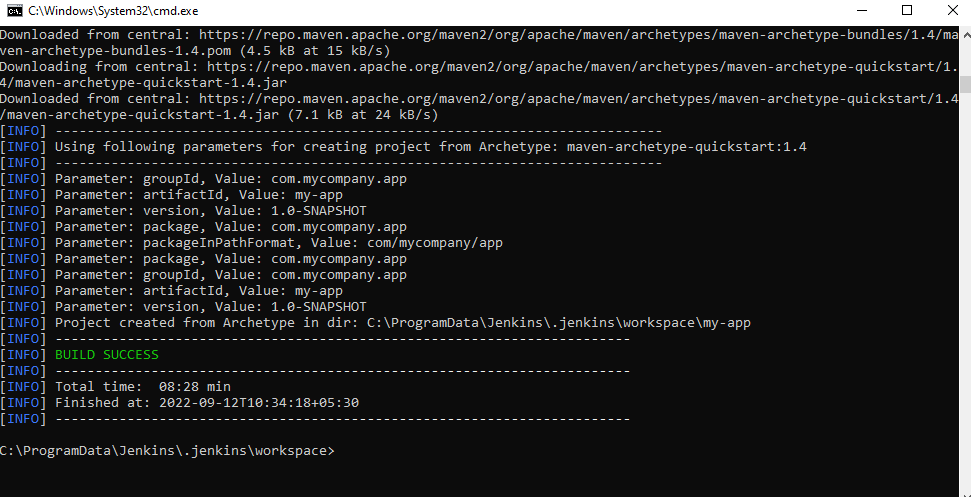
Install Maven Integration plugin if not installed earlier and got to Manage Jenkins -> Global tool configurations -> Set Name => Maven <version> and Maven Home => path value of MAVEN\_HOME.

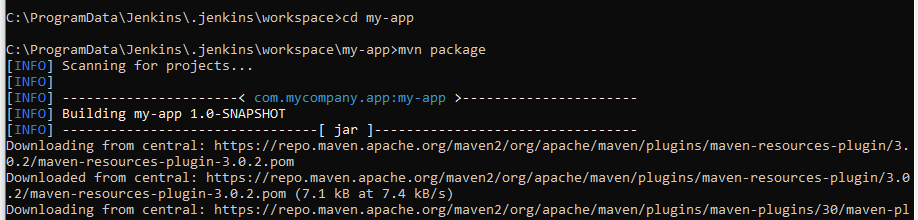


1. Build maven project in your machine using mvn commands to learn how to create and build any maven project as shown here: [https://maven.apache.org/guides/getting-started/maven-in- five-minutes.html](https://maven.apache.org/guides/getting-started/maven-in-%20five-minutes.html)

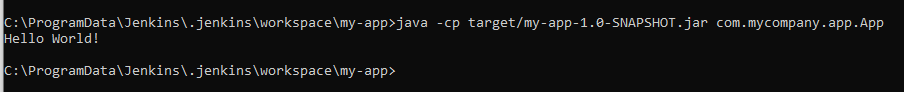




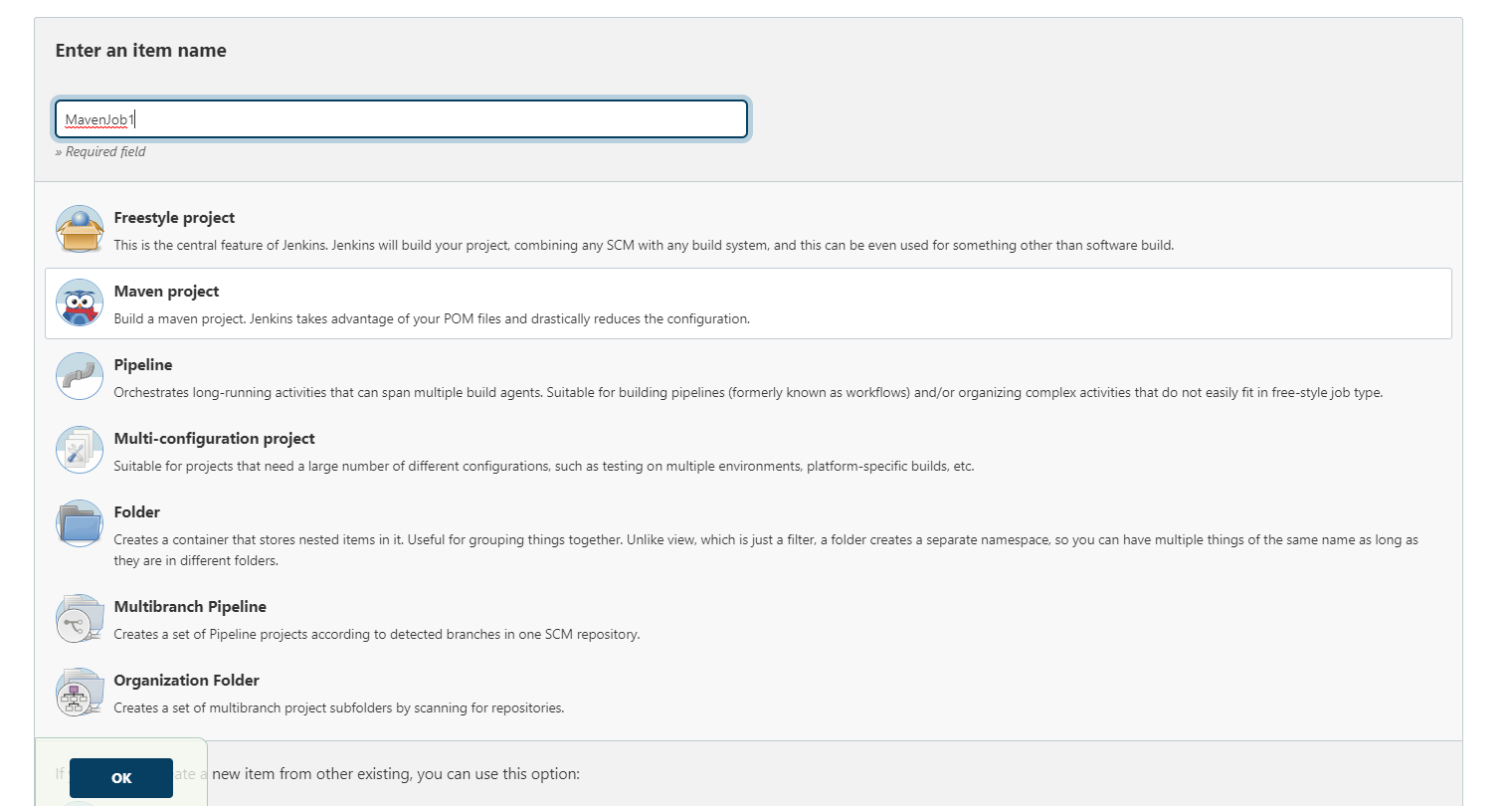






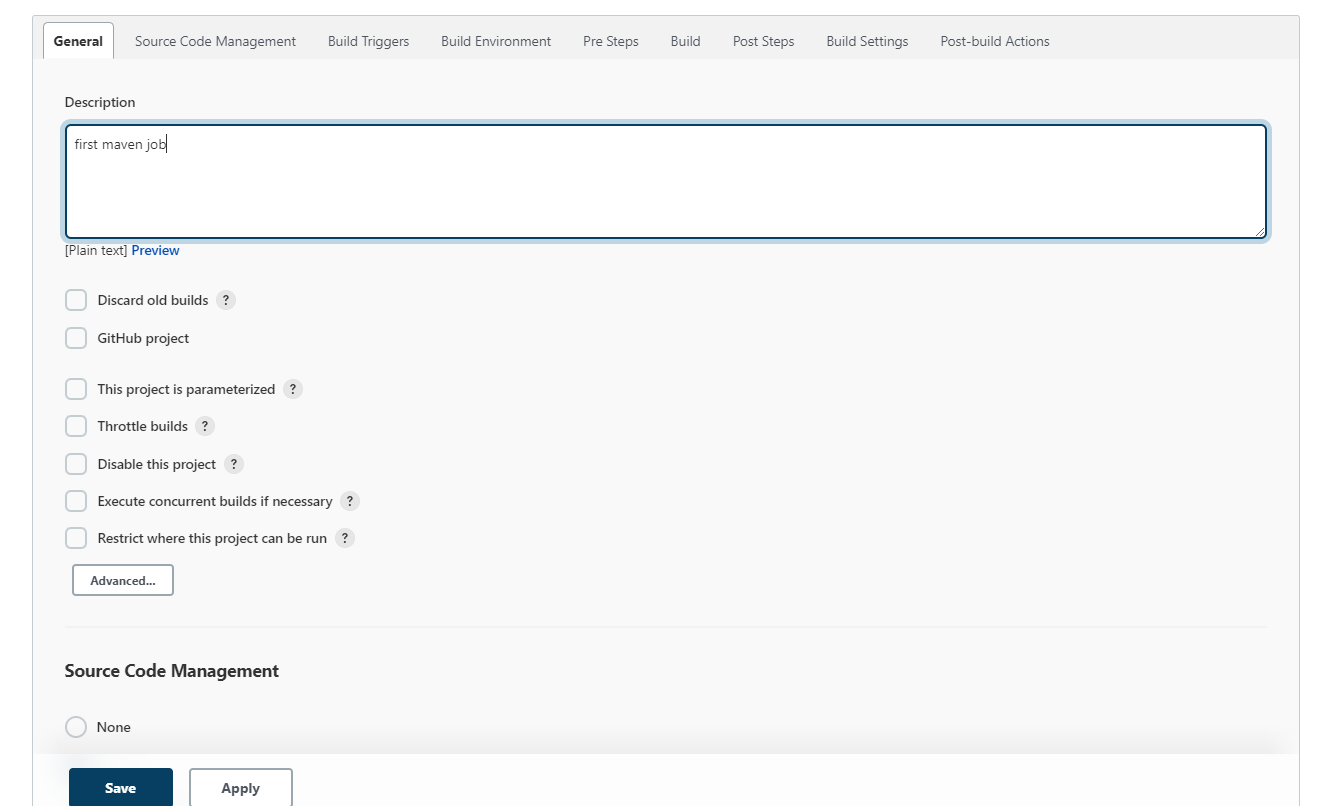


1. Create a maven job as Maven Project.



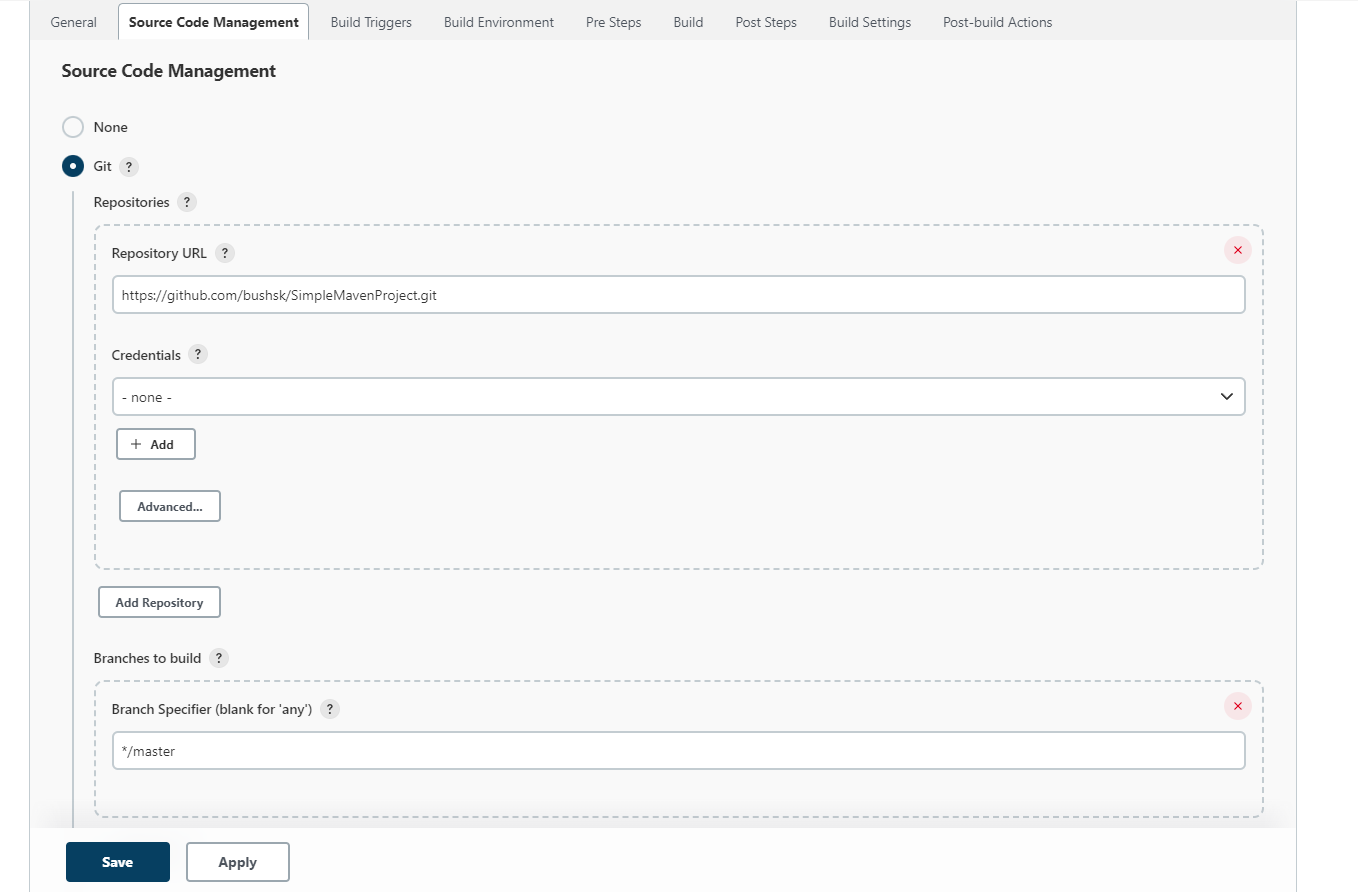
4. In Configure section - >go to Source Code Management -> Git - > paste this repository URL

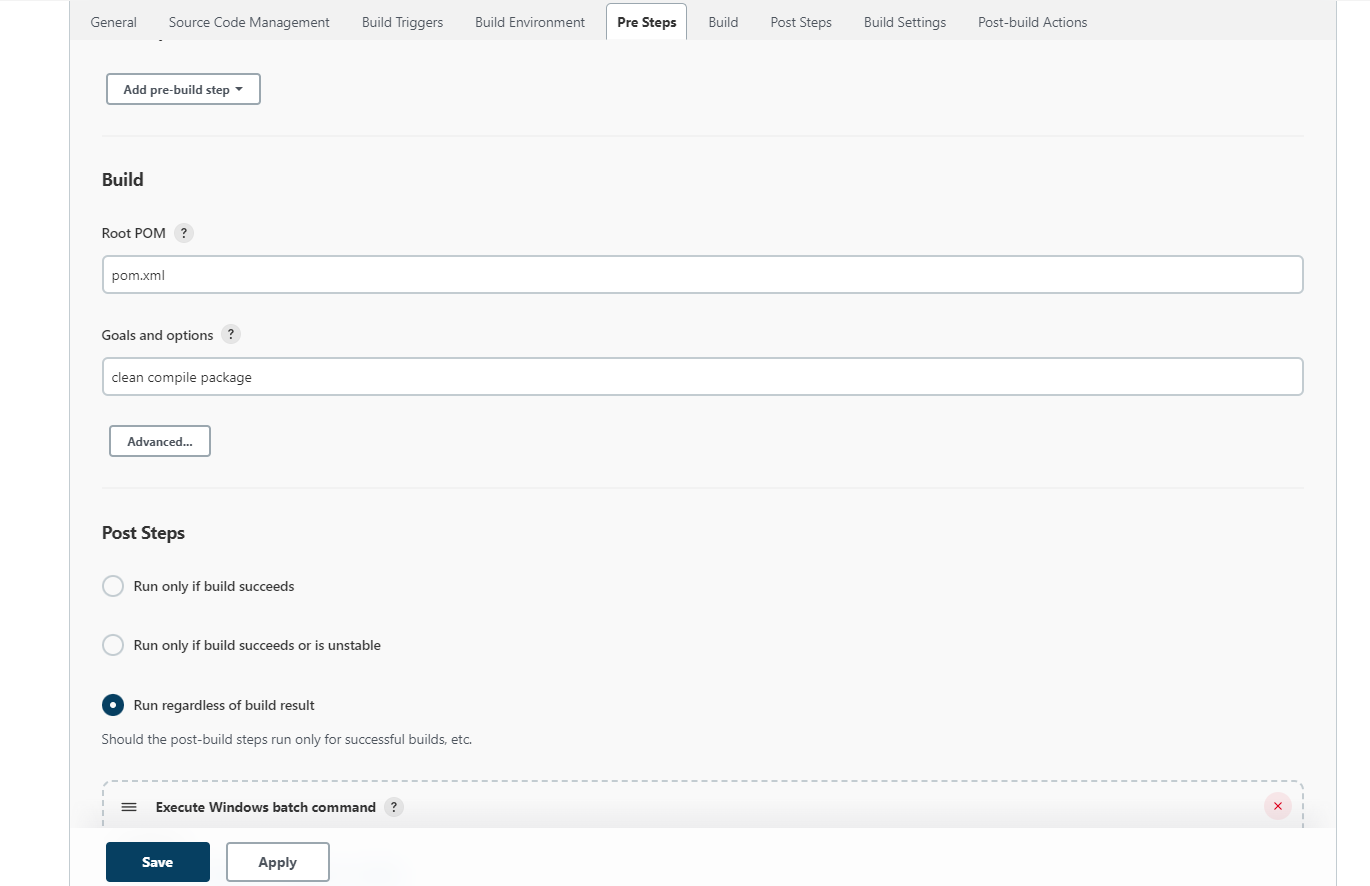
=> https://github.com/bushsk/SimpleMavenProject.git (Assuming our maven project is on GitHub repository).

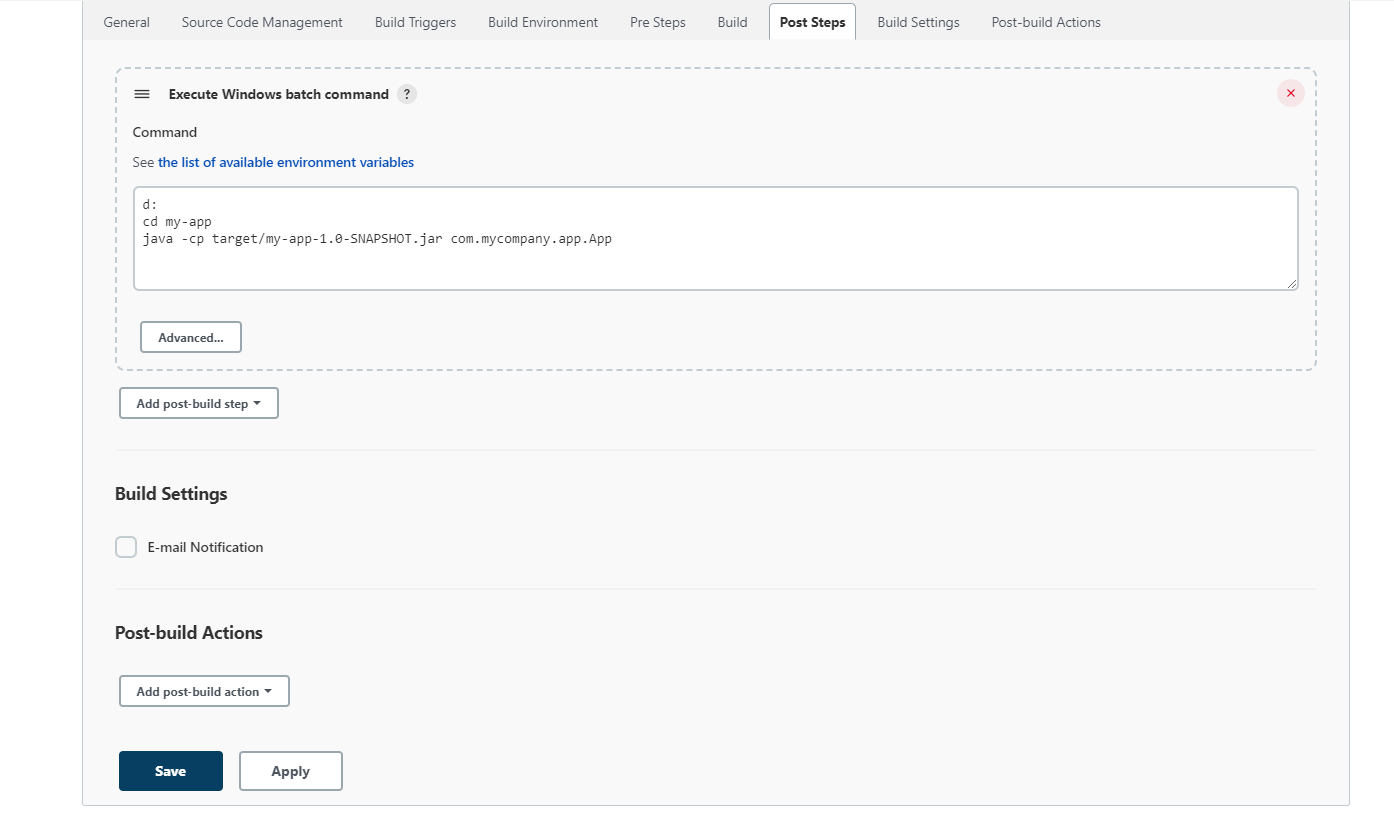


Note: Ensure that the GitHub Plugin is installed at this point. If it has not been installed, then do install it from Manage Jenkins-> Manage Plugins

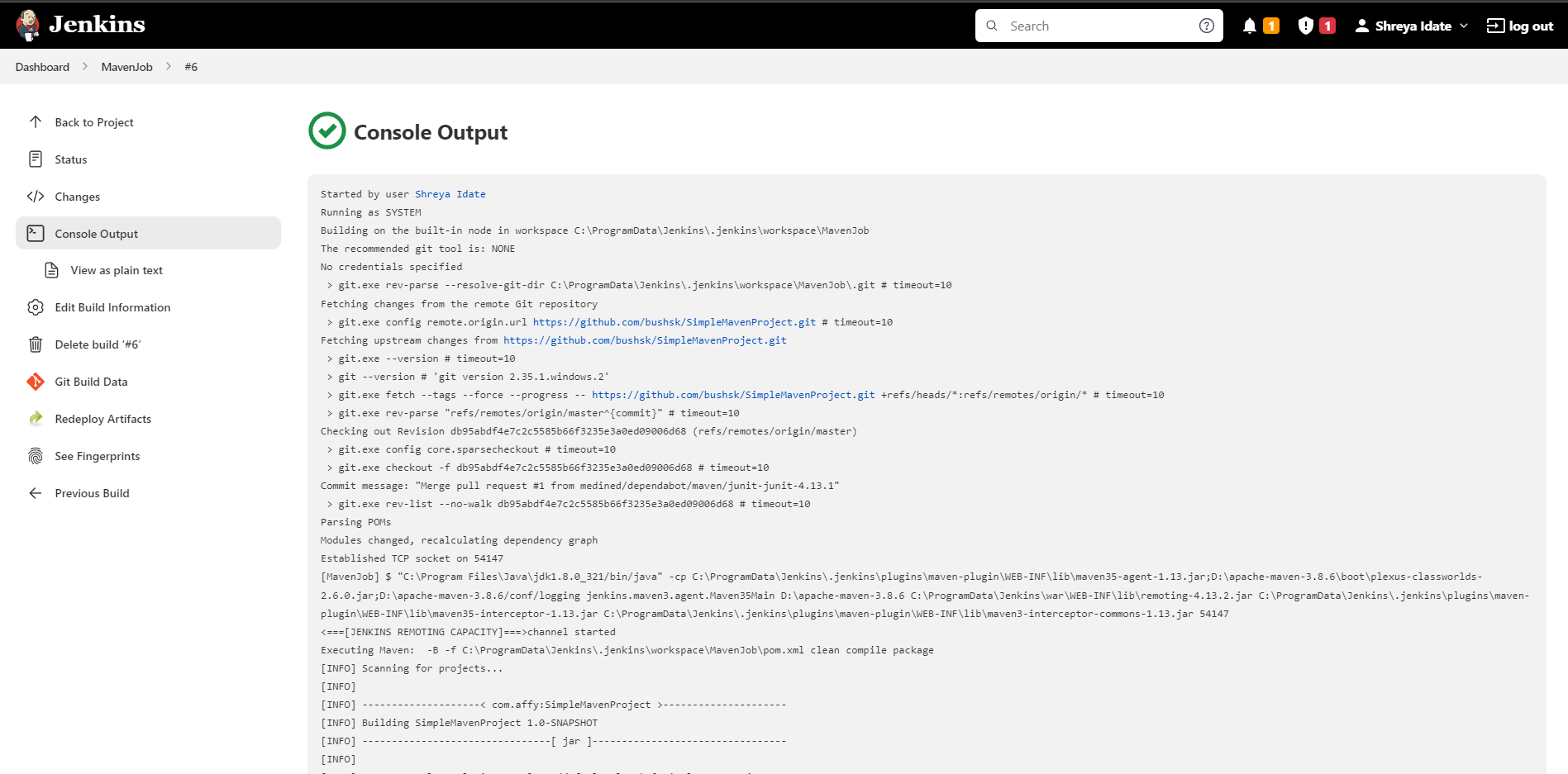
5. In Build section -> Write the goals and options as => clean compile package. Make sure Root POM is set to pom.xml. -> save and apply

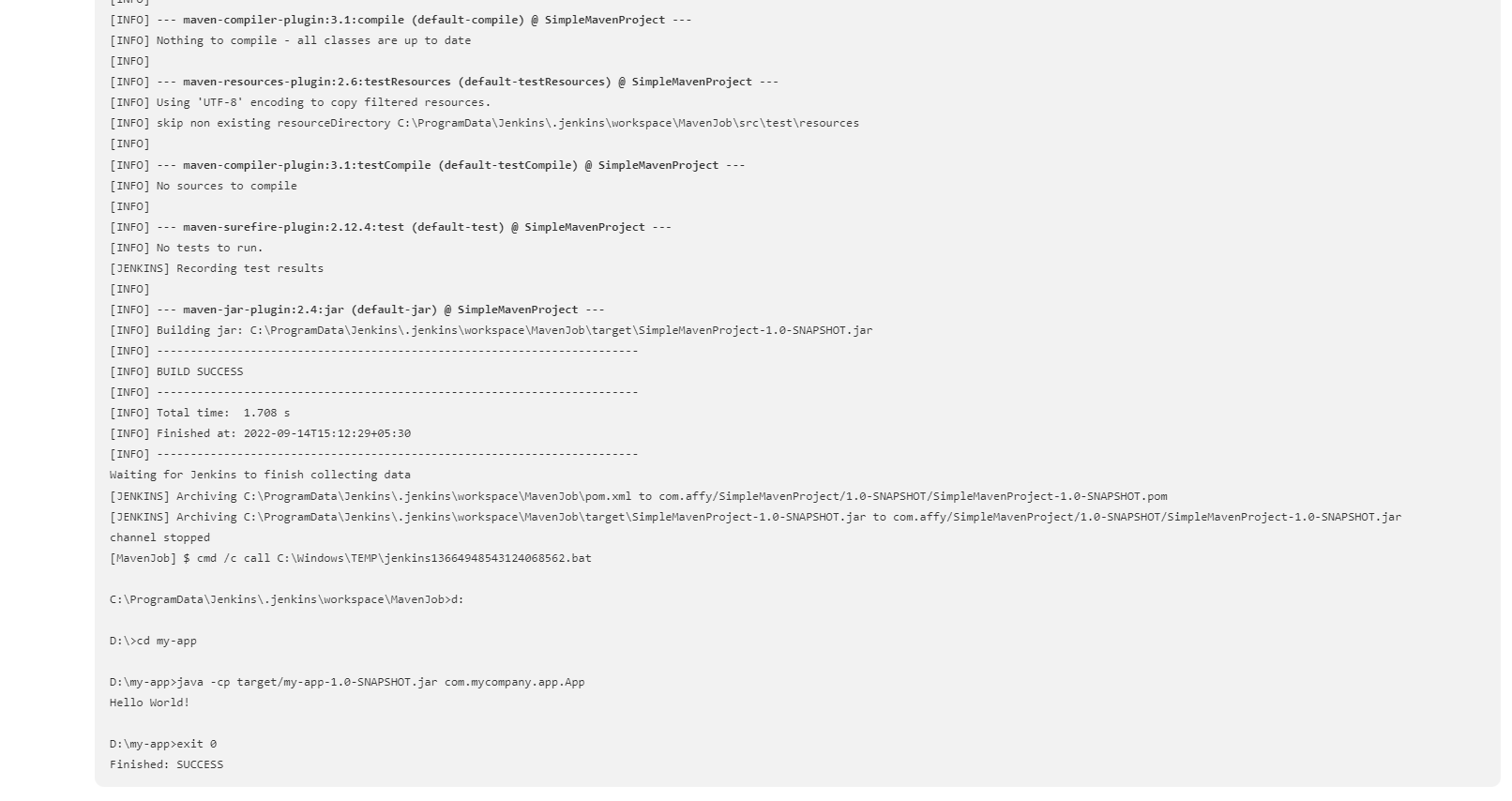






* 1. Go to build now -> console output and verify whether the .jar file with the given artefact has been created or not inside the given path shown in the console output.



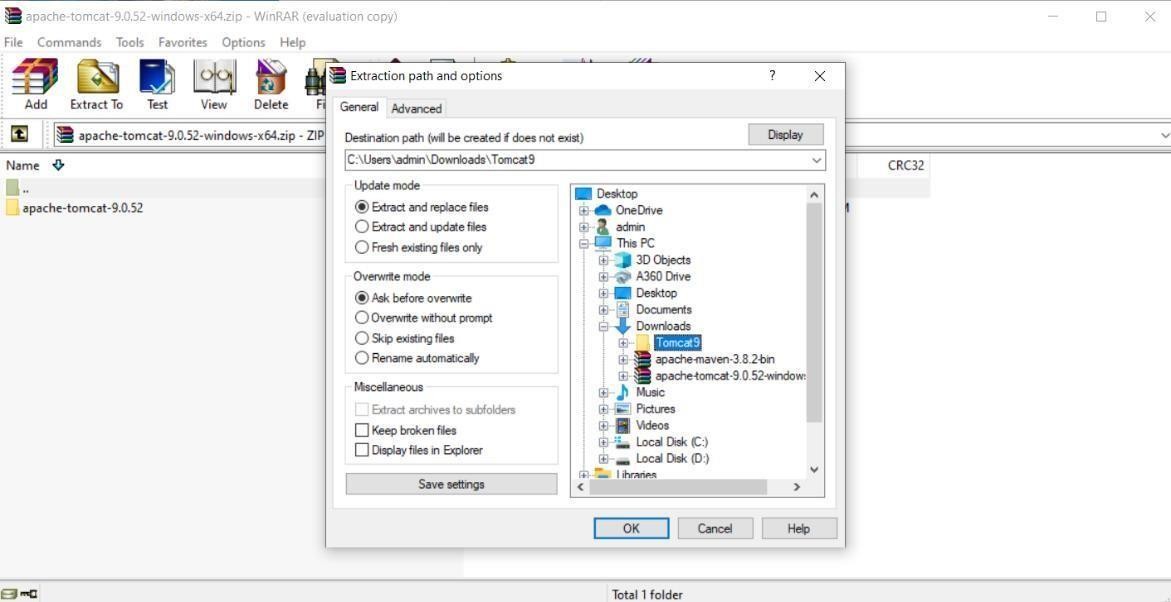


1. **To build the pipeline of jobs using Maven / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.**

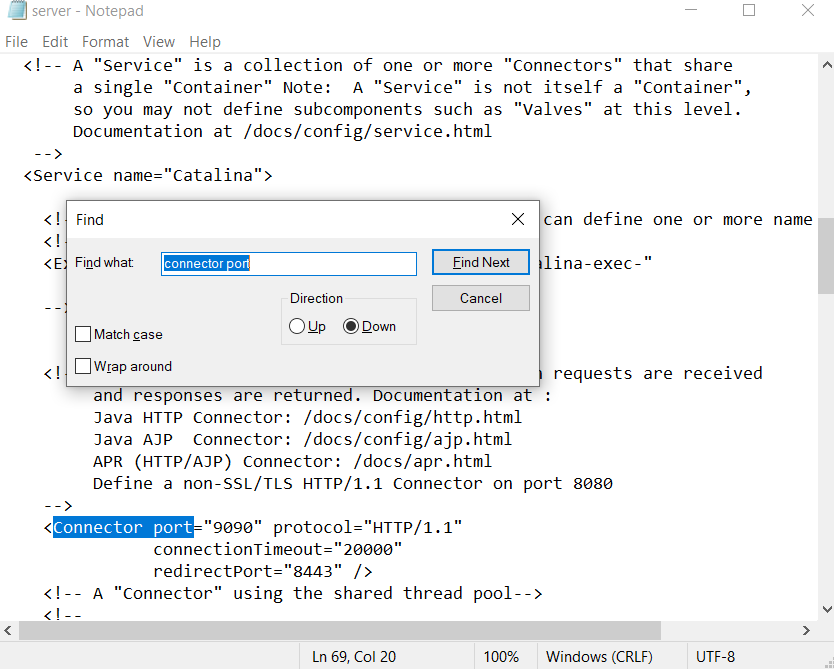
Steps for Building Maven job from the GitHub remote repository into Jenkins using Jenkins pipeline: Setup and run apache tomcat server:

1.

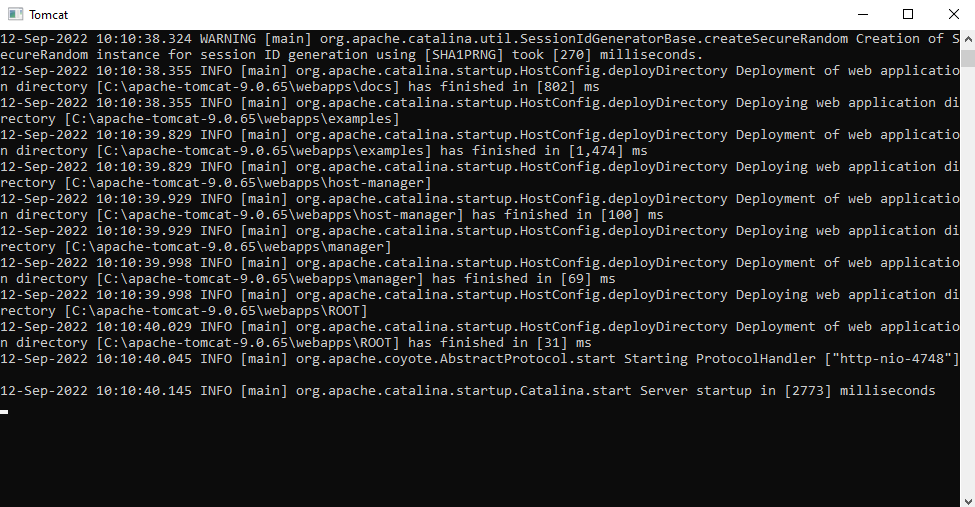
1. Download apache tomcat server from official apache‘s tomcat website and Install it by extracting and set it to run on port 9090 or any other port since Jenkins is already running on 8080 port and by default tomcat also runs on 8080.

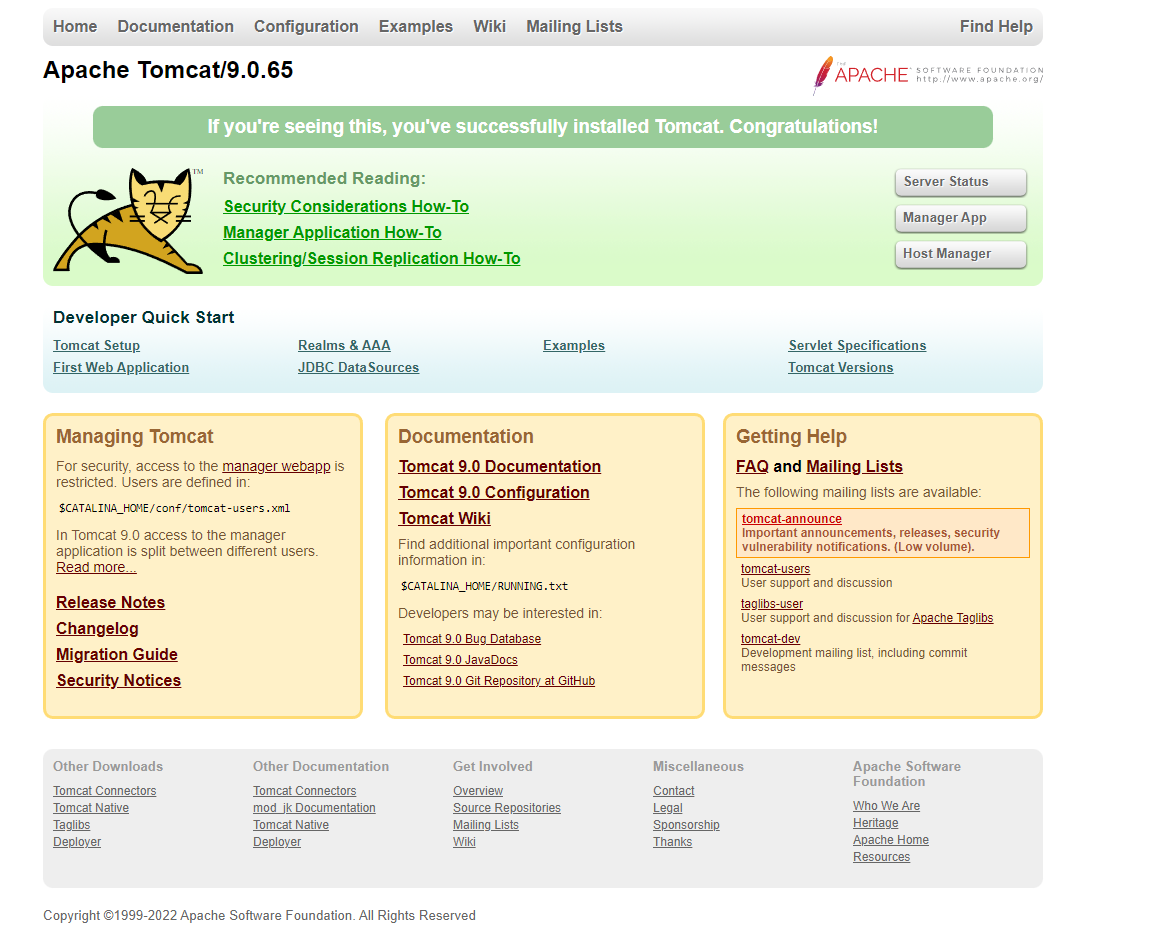


1. To do the above, go to the apache tomcat directory and find server.xml inside config folder. Find this line by Ctrl+f ―<Connector port="8080"‖ and replace it with

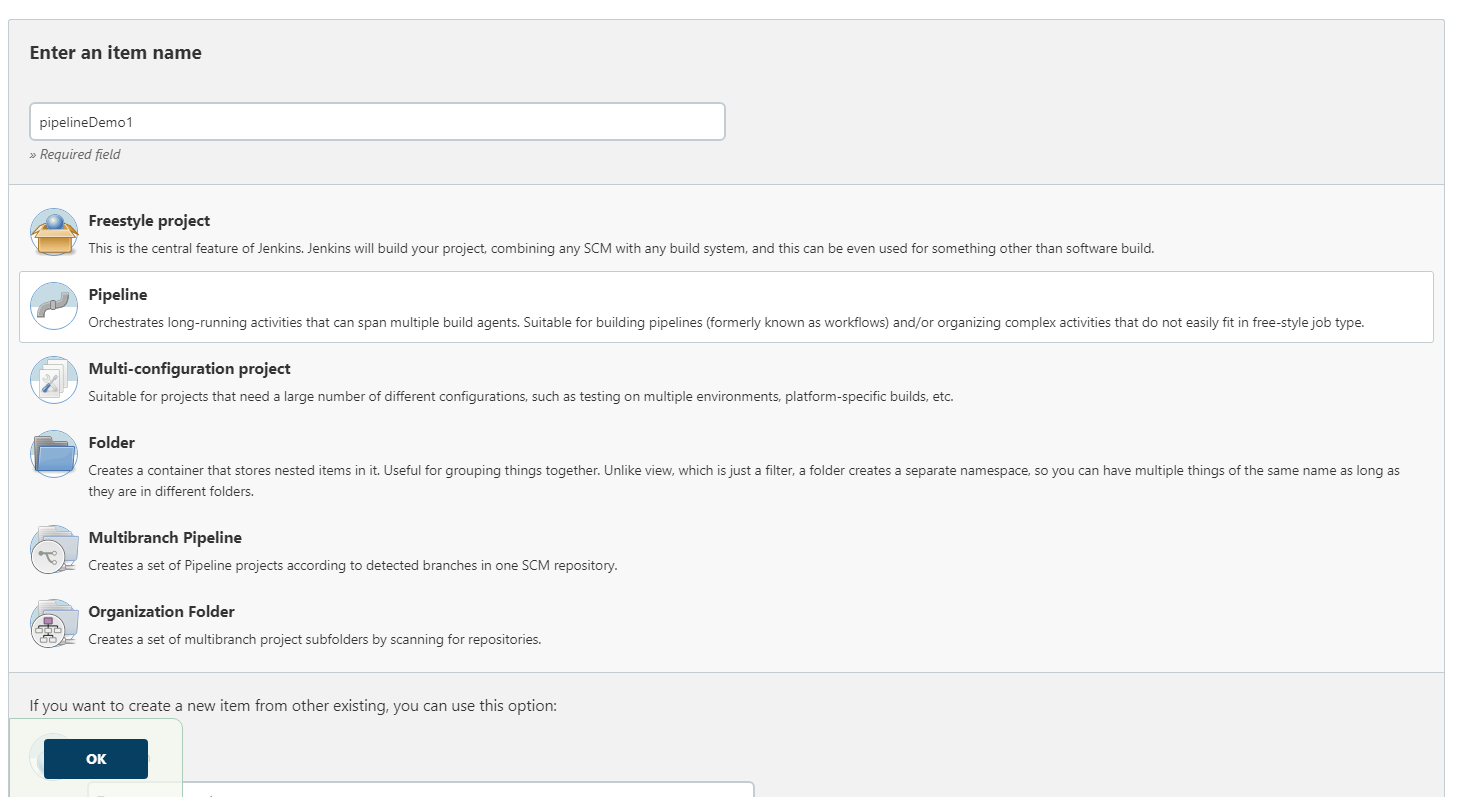
―<Connector port="9090" and save it.

1. Start the server by running the startup.bat file found inside apache-tomcat-9.0.52\bin folder and go to browser and test if the home page of server is up or not on localhost:9090.





2. Go to Jenkins Dashboard and create a new pipeline project.



3. Go to Pipeline Section and choose Pipeline Script and write the following script here:

pipeline

{

agent any

tools

{

maven 'Maven 3.8.2'

}

stages

{

stage("Git Clone")

{

steps

{

git 'https://github.com/bushsk/hello-world.git'

}

}

stage("Build")

{

steps

{

bat 'mvn clean install'

}

}

stage("Deploy")

{

steps

{

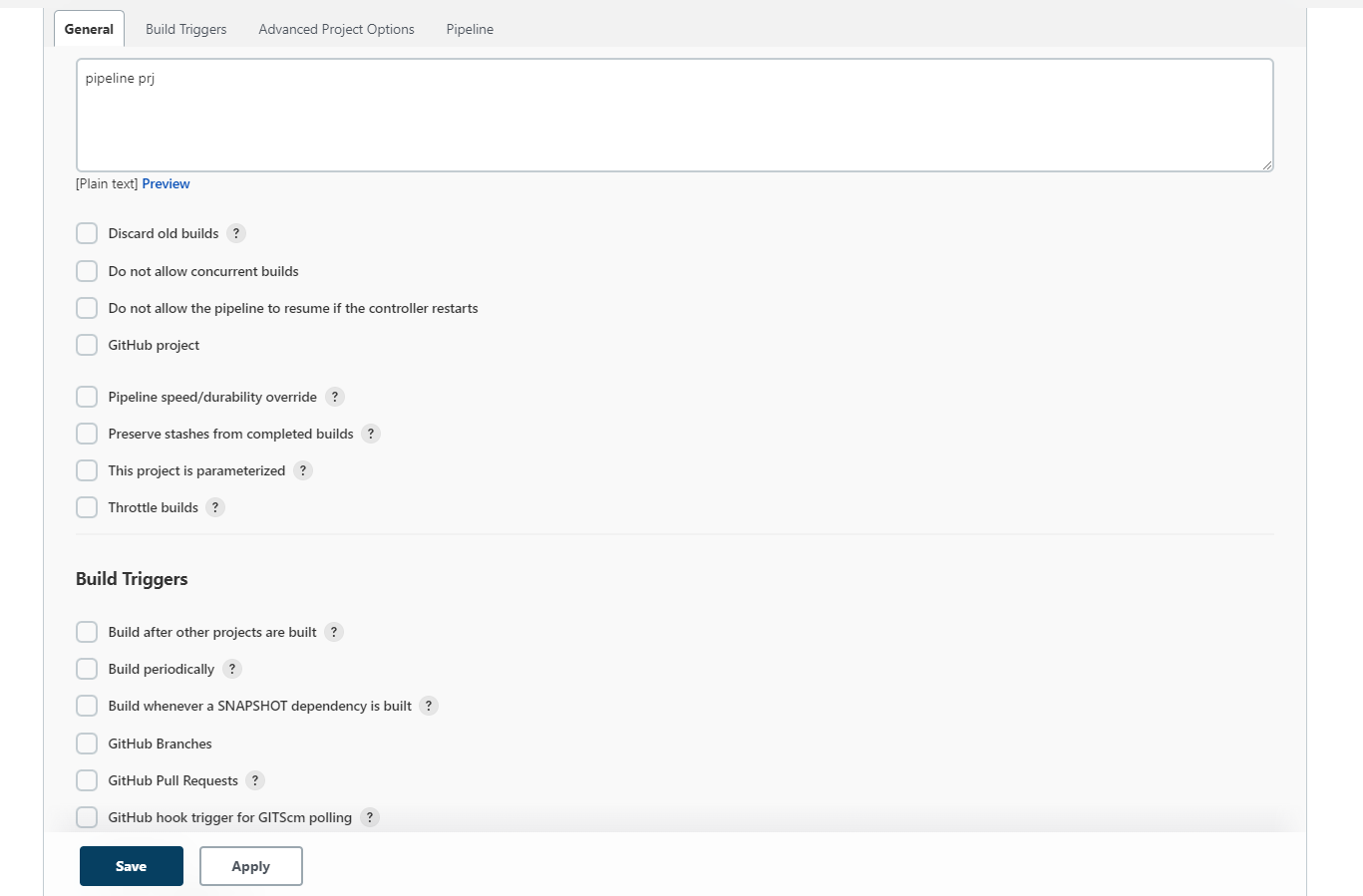
bat     'copy C:\\ProgramData\\Jenkins\\.jenkins\\workspace\\Pipeline\\webapp\\target\\webapp.war D:\\apache-tomcat-9.0.65\\webapps'

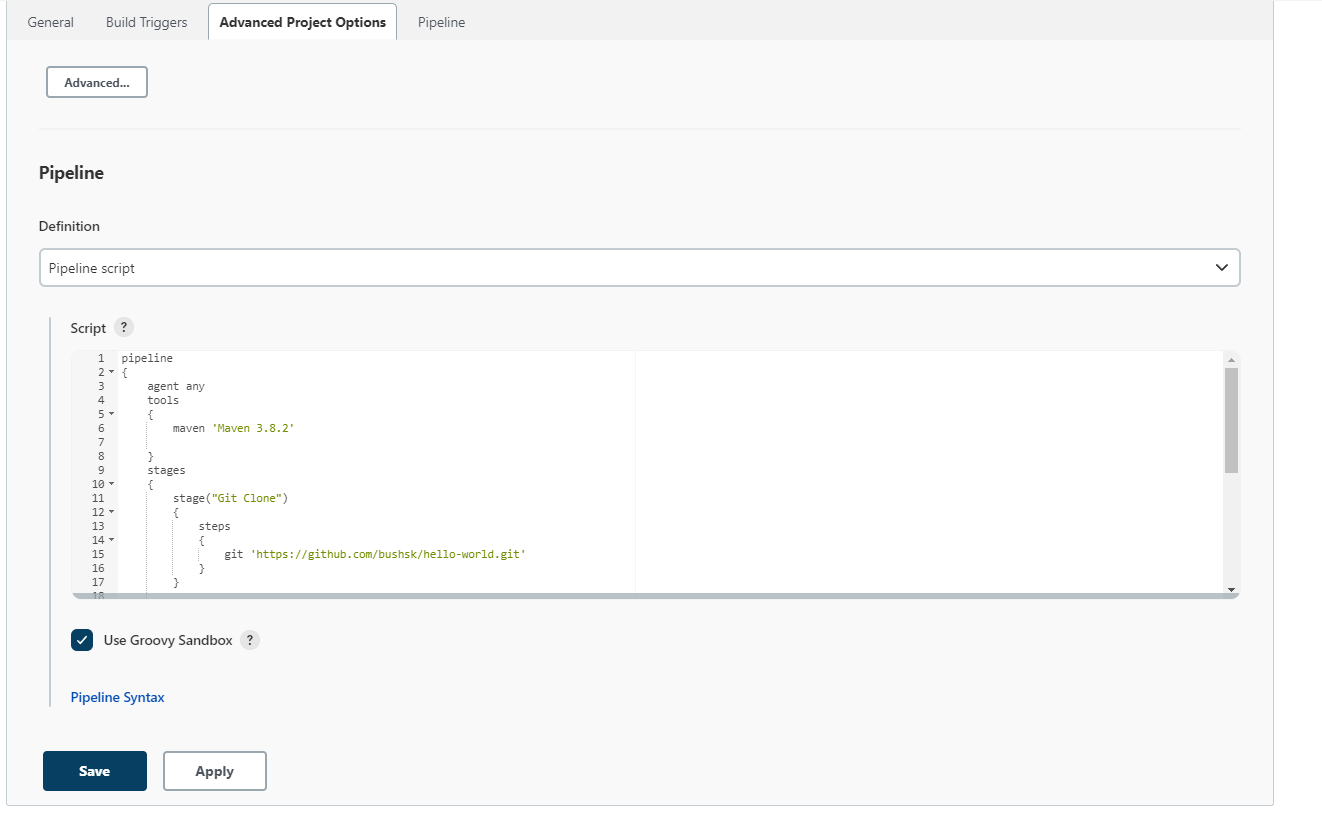
}

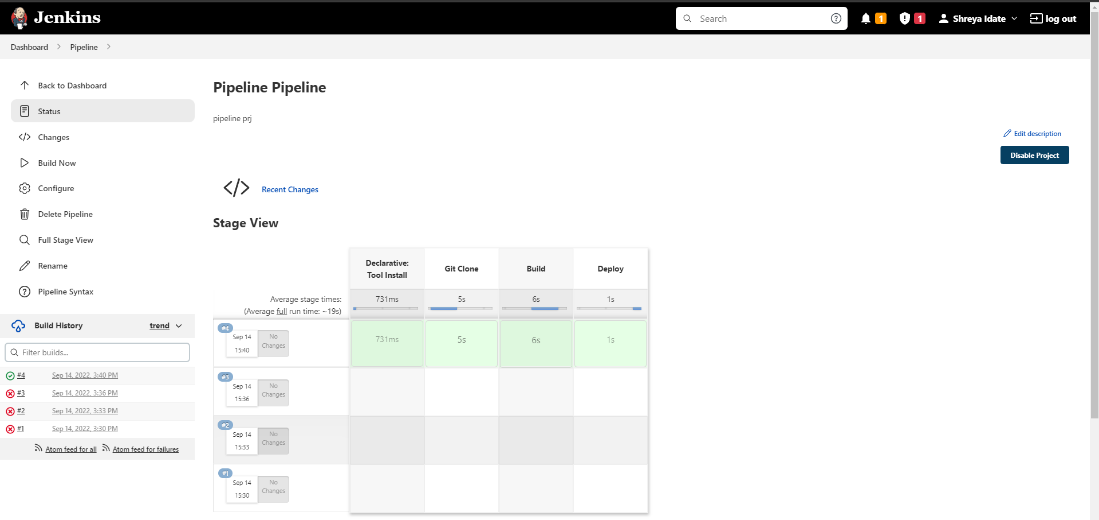
}

}

}

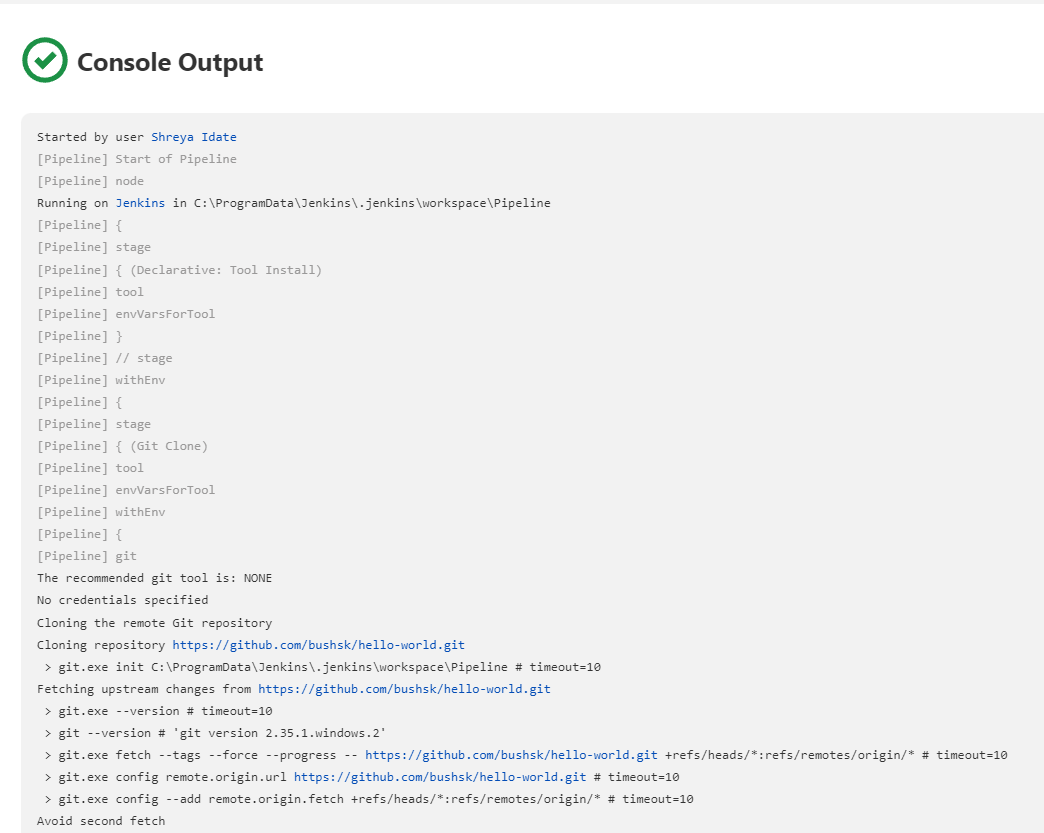


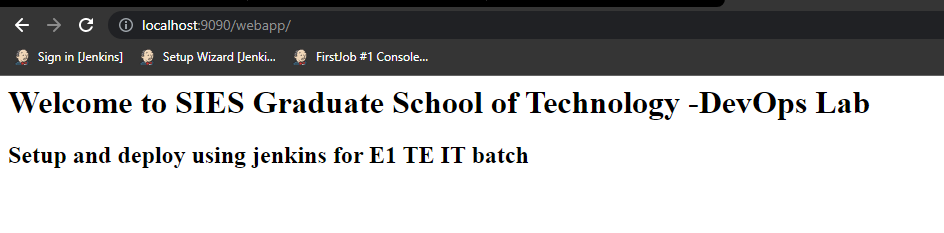




4. Got to Build Now and verify if the pipeline build is successful.

Go to http://localhost:9090/webapp/ and verify if the home page of your application is up or not.





**CONCLUSION:** Successfully installed Jenkins and configured Jenkins with Maven/Ant/Gradle, GitHub and Python Plugins to set up a build Job.