**PROJECT1: Jenkins Freestyle Project**

**Jenkins Freestyle Projects are used to execute some scripts to gather some information or to execute some task.**

1. Pull image from docker in power shell: [jenkins/jenkins Tags | Docker Hub](https://hub.docker.com/r/jenkins/jenkins/tags)

PS C:\Users\Shilpi\_Harnal> docker pull jenkins/jenkins:latest

2. Check image name:

PS C:\Users\Shilpi\_Harnal> docker images

-jenkins/jenkins

3. PS C:\Users\Shilpi\_Harnal> docker run -dp 8080:8080 jenkins/jenkins:latest

<-some id—->

4. Open browser and run === localhost:8080

It will run the URL: 127.0.0.1/8080

It will prompt for password from: /var/jenkins\_home/secrets/initialAdminPassword

To access the above file we need to execute a container having jenkins/jenkins image.



5. Check the container id:

PS C:\Users\Shilpi\_Harnal> docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d400148e1240 jenkins/jenkins:latest "/usr/bin/tini -- /u…" About an hour ago Up About an hour 0.0.0.0:8080->8080/tcp, 50000/tcp upbeat\_darwin

6. Execute container with id:

PS C:\Users\Shilpi\_Harnal> docker exec -it d400148e1240 bash

7. New prompt. Check list of files

jenkins@d400148e1240:/$ ls

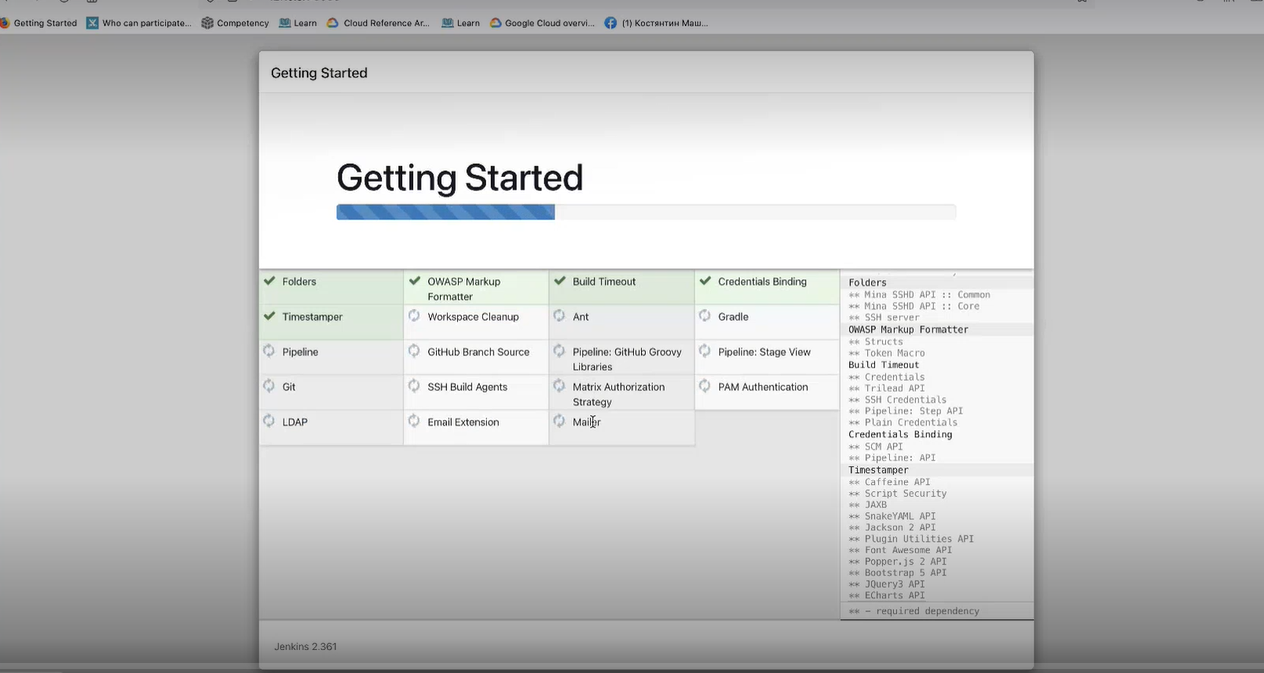
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

8. Reference to point No. 4, extract password from: /var/jenkins\_home/secrets/initialAdminPassword

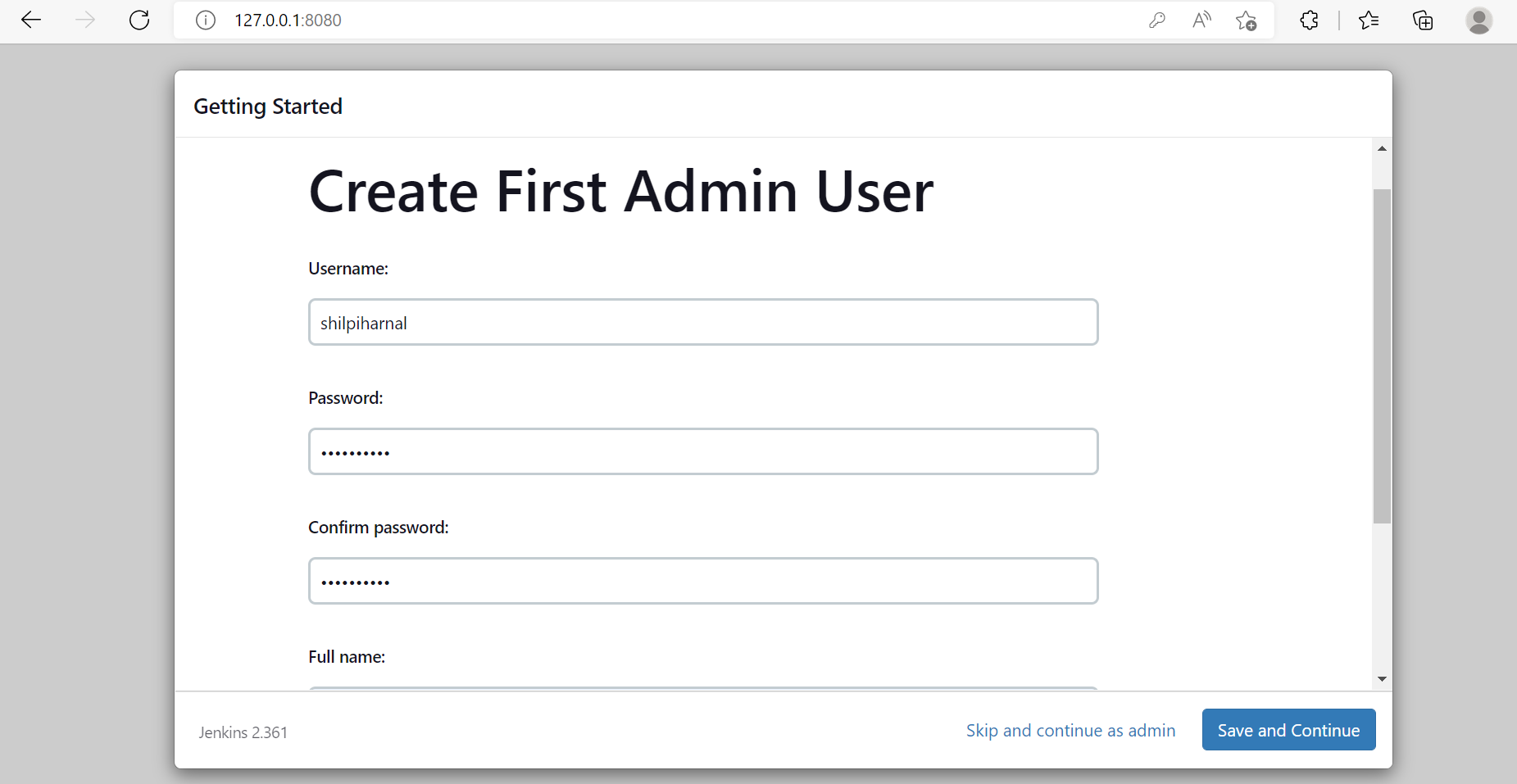
jenkins@d400148e1240:/$ cat /var/jenkins\_home/secrets/initialAdminPassword

4c58a23ad96441ae800b3eb78583f3b3

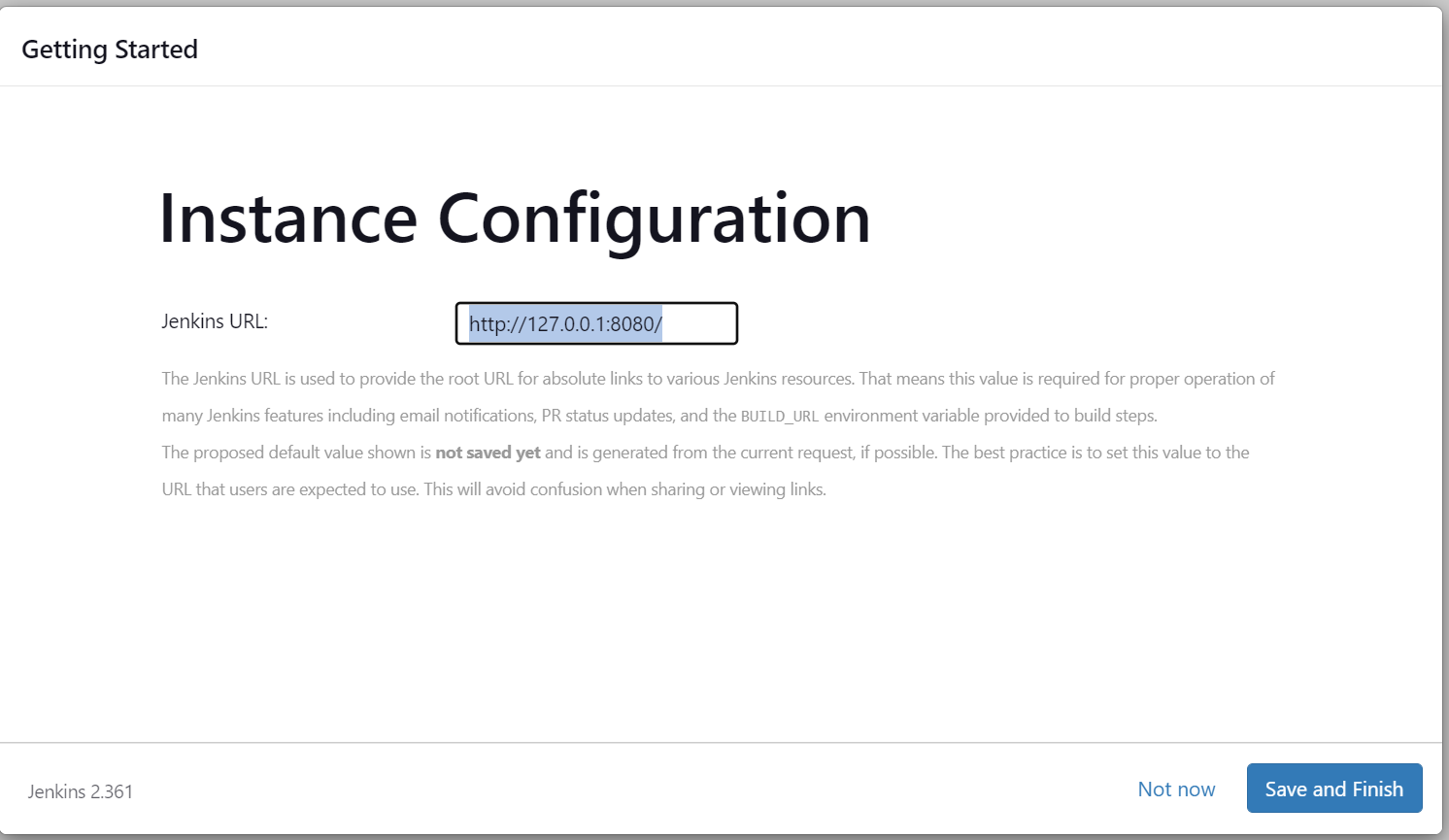
9. Reference to point No. 4, Go back to browser and enter password and click continue. It will start building setup that would take some time:



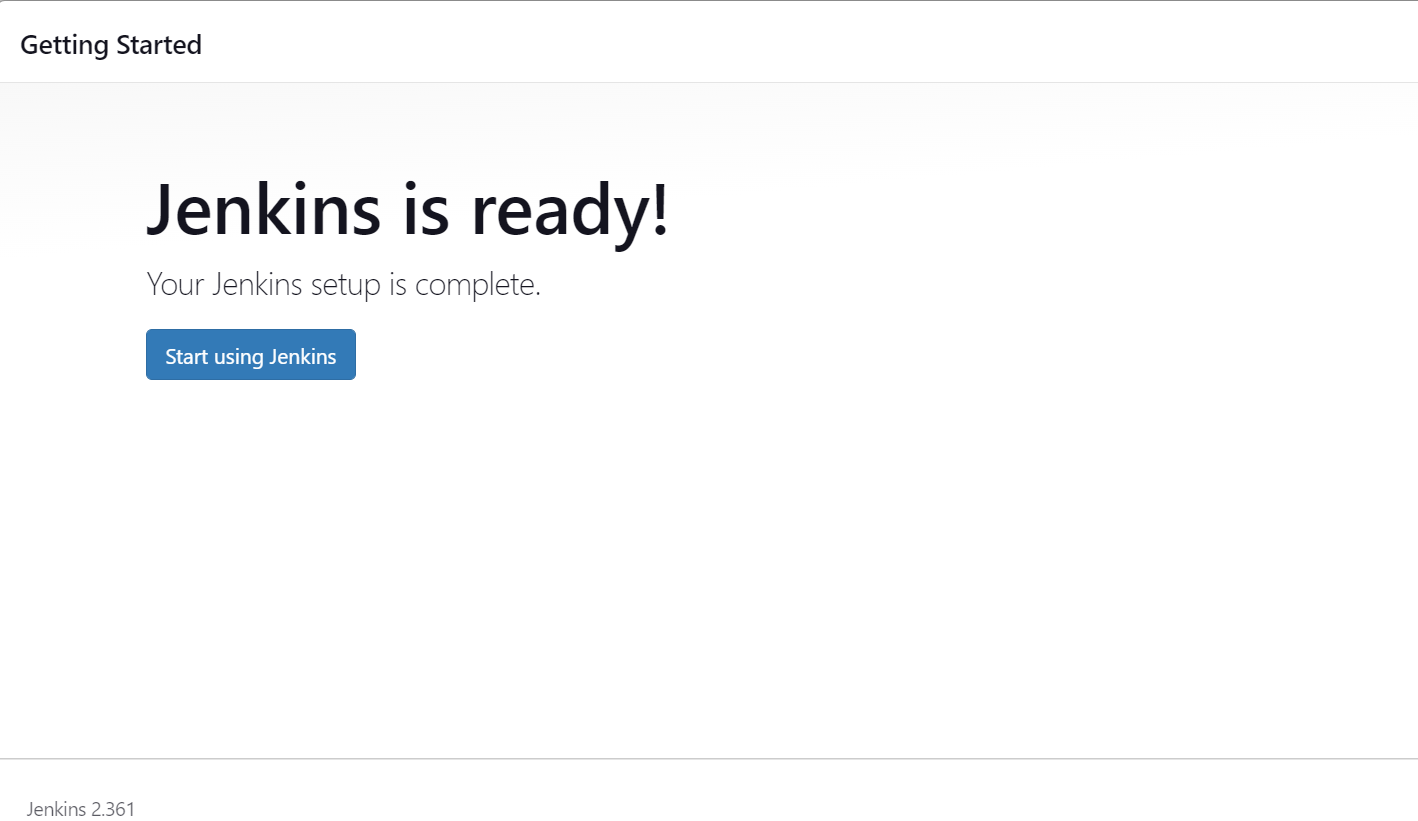
10. Create Admin User:



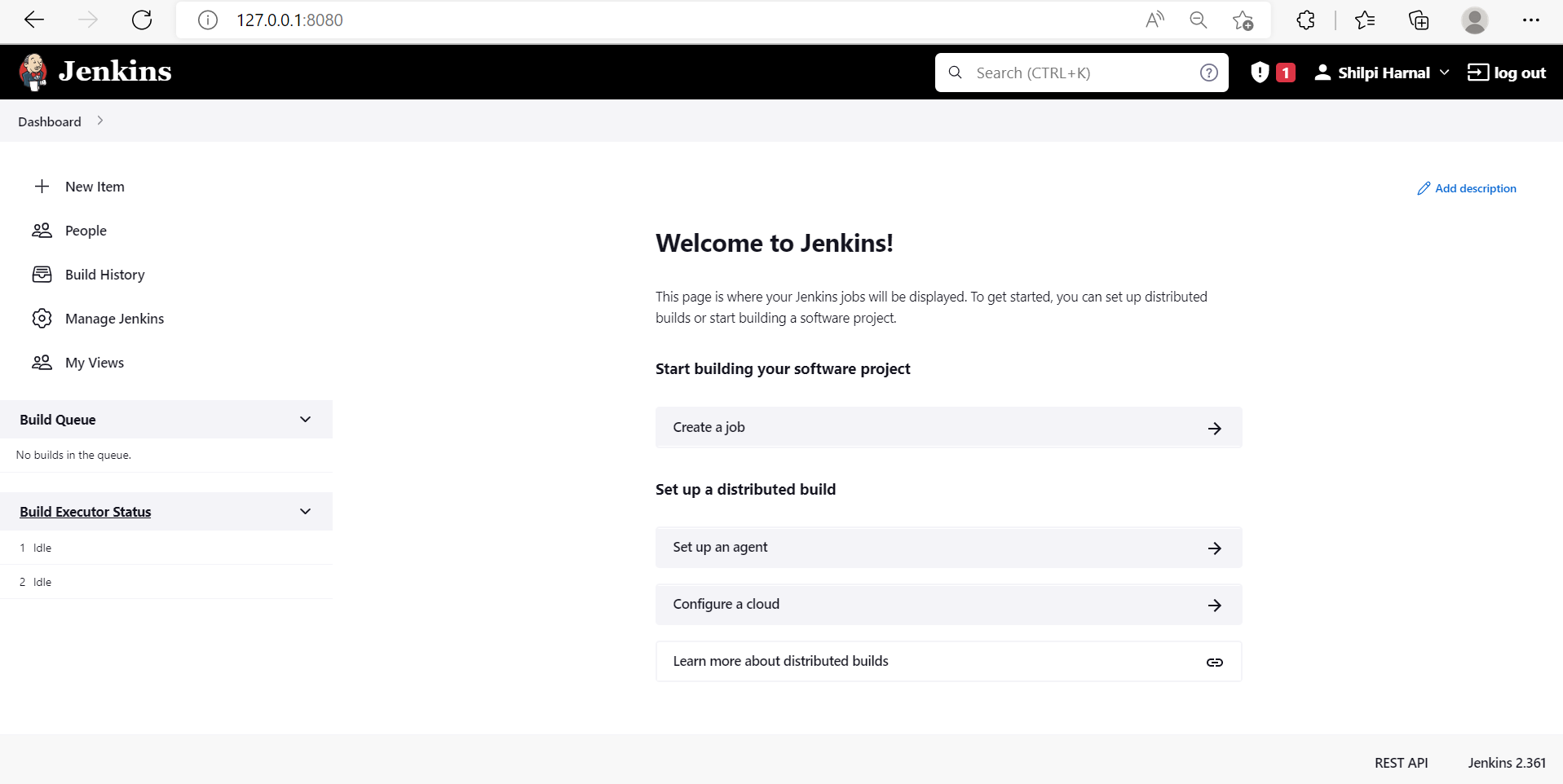
11. Click Save and finish. Or you can also give any other url :



12. Click - start using Jenkins

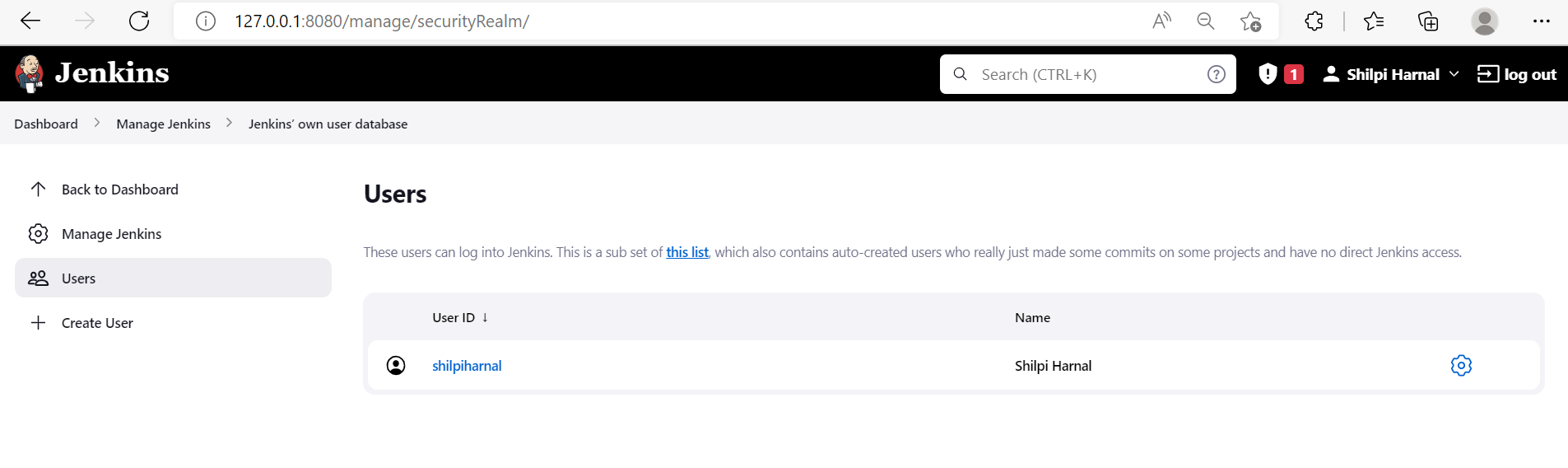


13. Jenkins Dashboard:



14. To check existing Users:

Manage Jenkins -> Manage Users



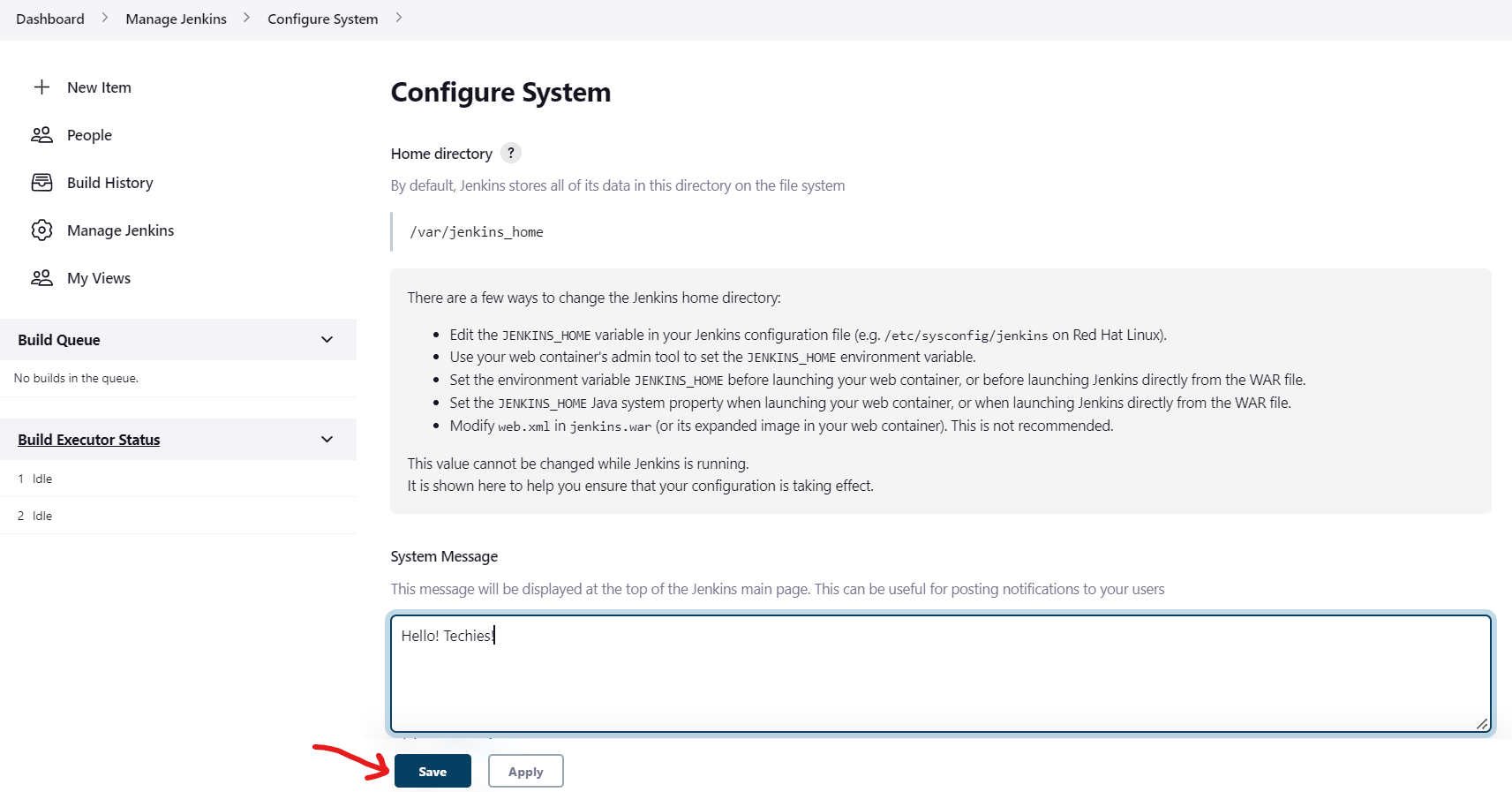
15. Manage Jenkins -> Configure Global Security

Enable following and click SAVE to save settings:

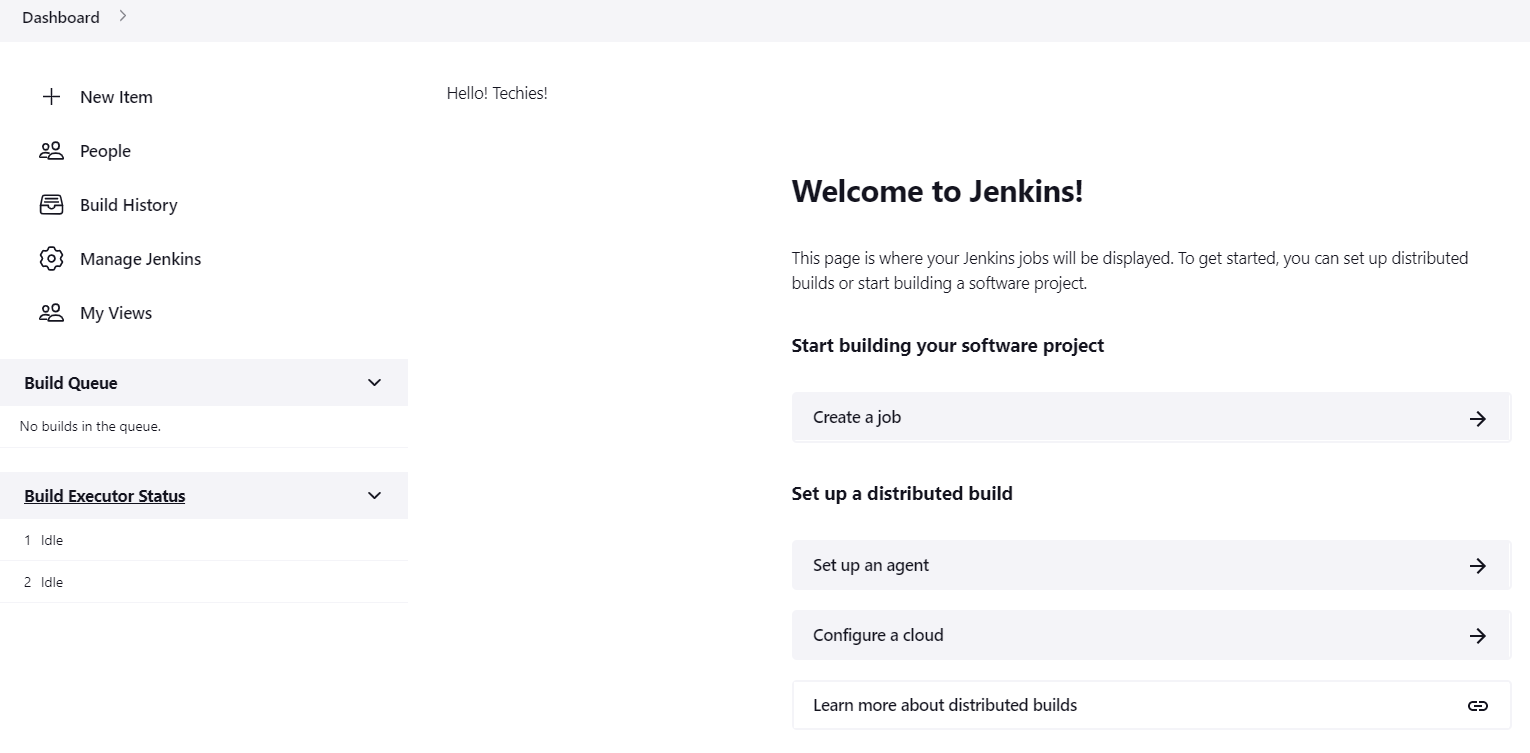


16. Manage Jenkins -> Configure System

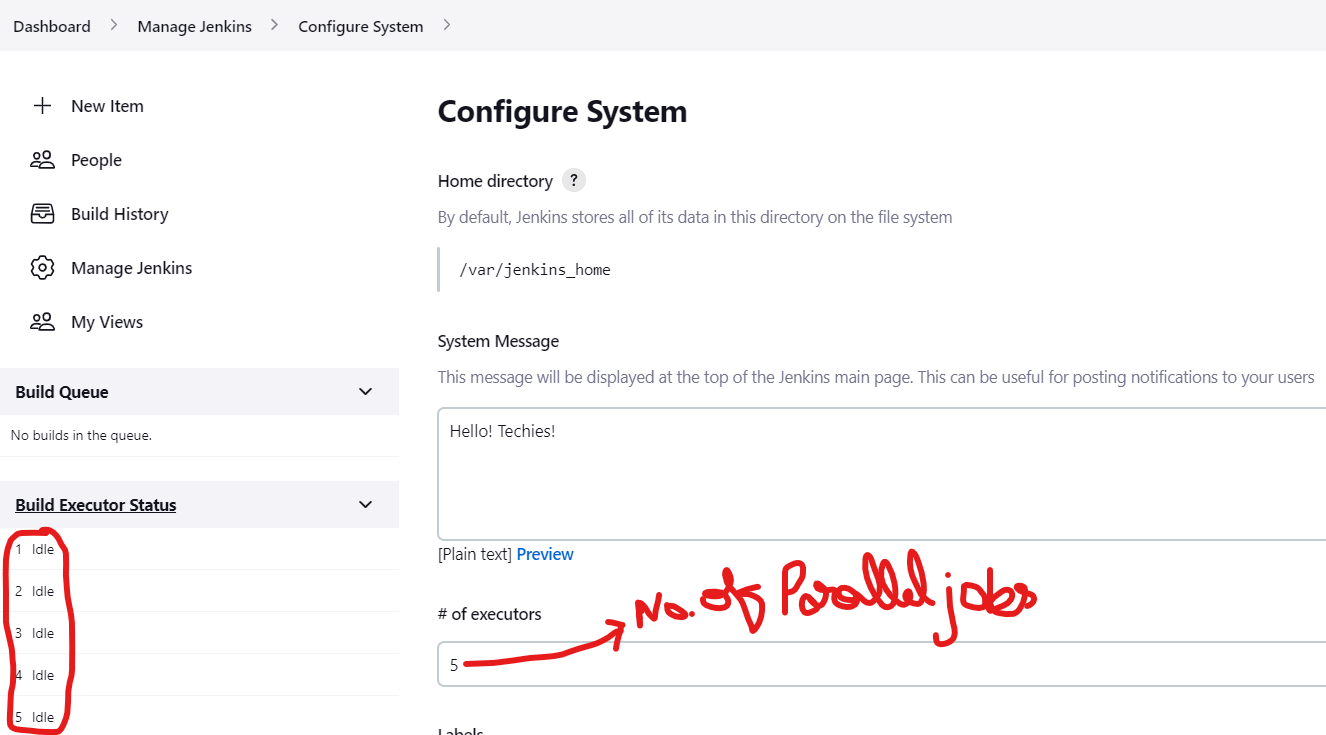
Write a message and save.



17. Next Screen:



18. Manage Jenkins -> Configure System

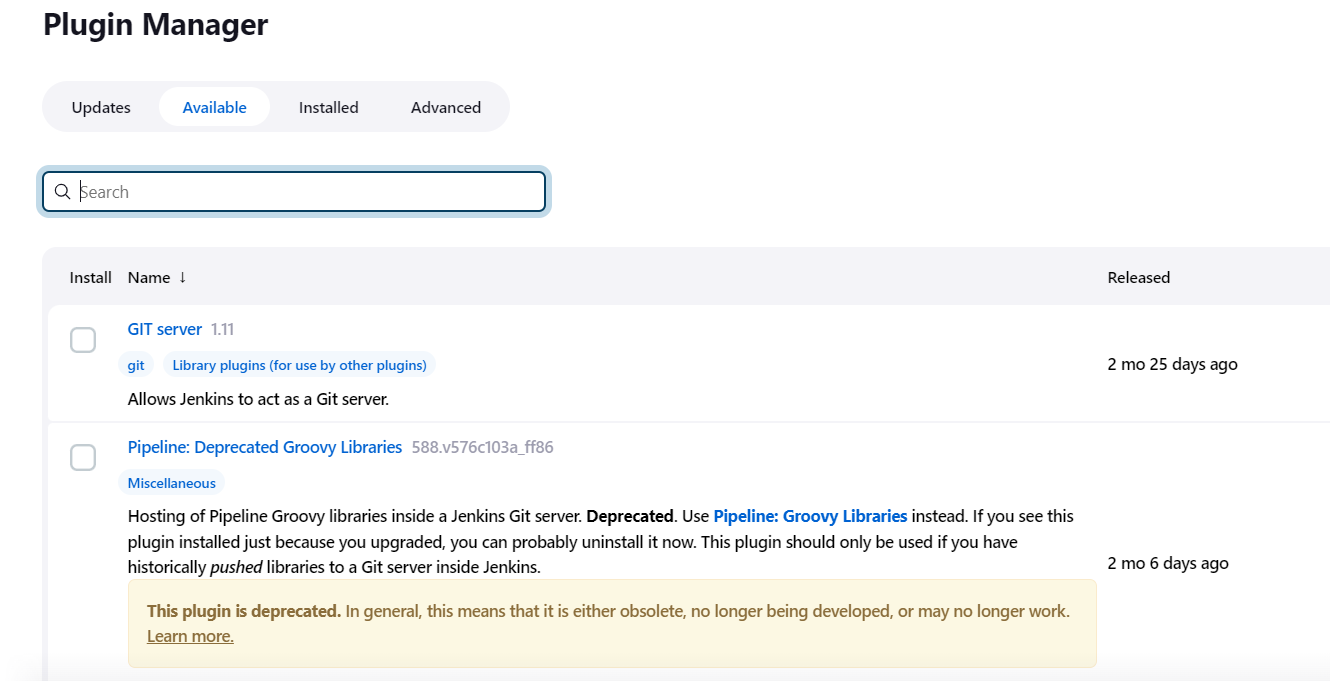


19. Manage Jenkins -> Global Tool Configuration

You can make changes and update settings there.

20. Manage Jenkins -> Plugin Manager

You can install, enable, disable and update plugins from here,

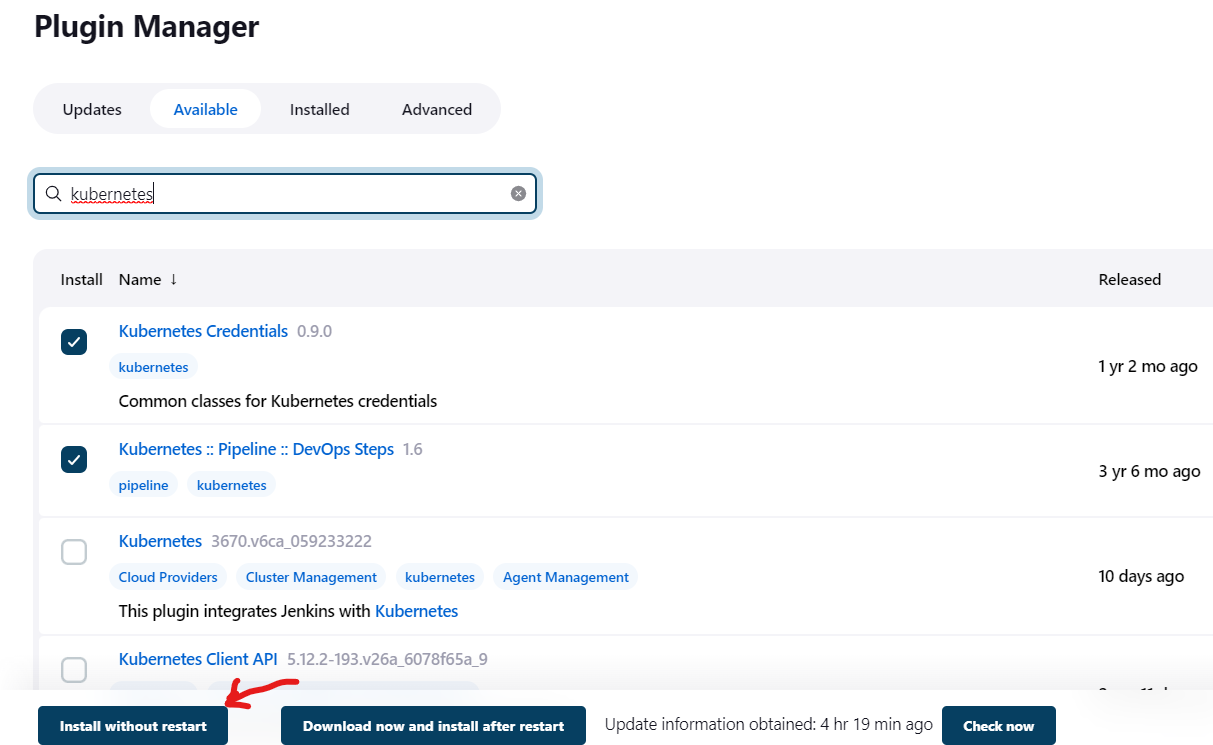


21. From available plugins search for BLUE OCEAN plugin and click INSTALL WITHOUT DOWNLOAD. It will install the same. You can also install other plugins like kubernets.

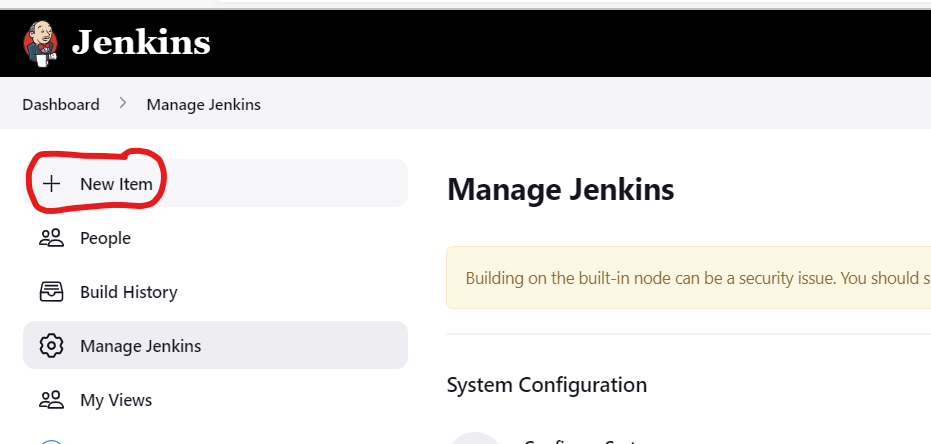
**Blue ocean: to check only the output of project.**



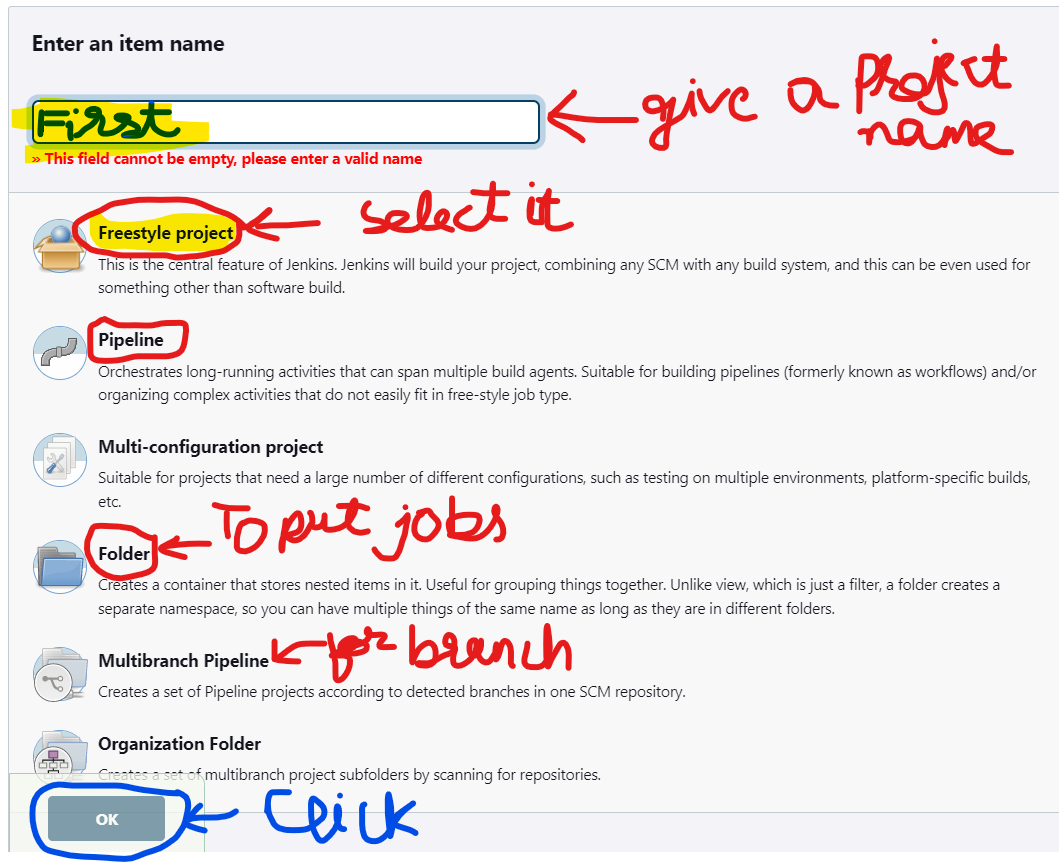
22. Also search for kubernetes credentials and kubernetes::pipeline::Devops Steps for installation.



23. To create a pipeline, click NEW ITEM :



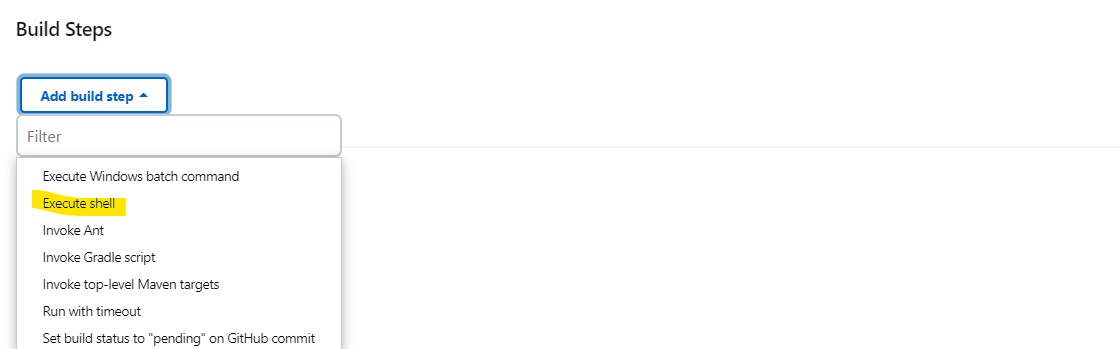
24. Give a NAME, select FREESTYLE PROJECT and click OK.



25. Add some description



26. Select EXECUTE SHELL and add some script to run as:



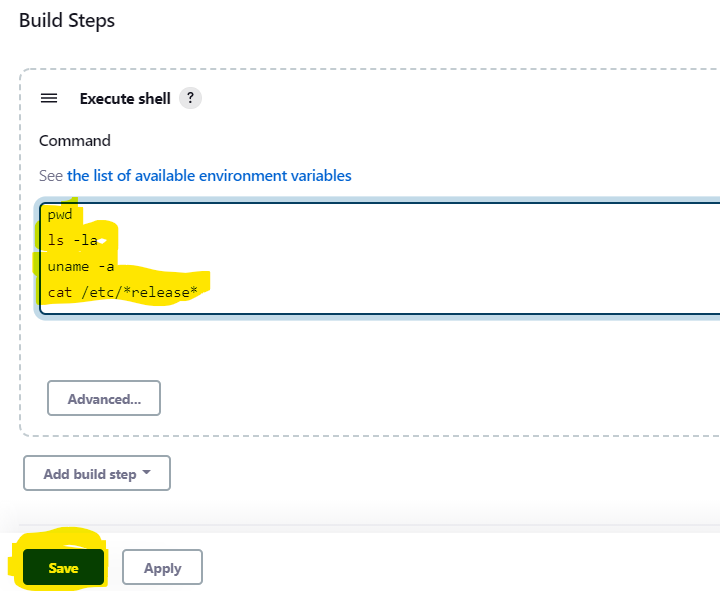
27. add some script to run as:

pwd

ls -la

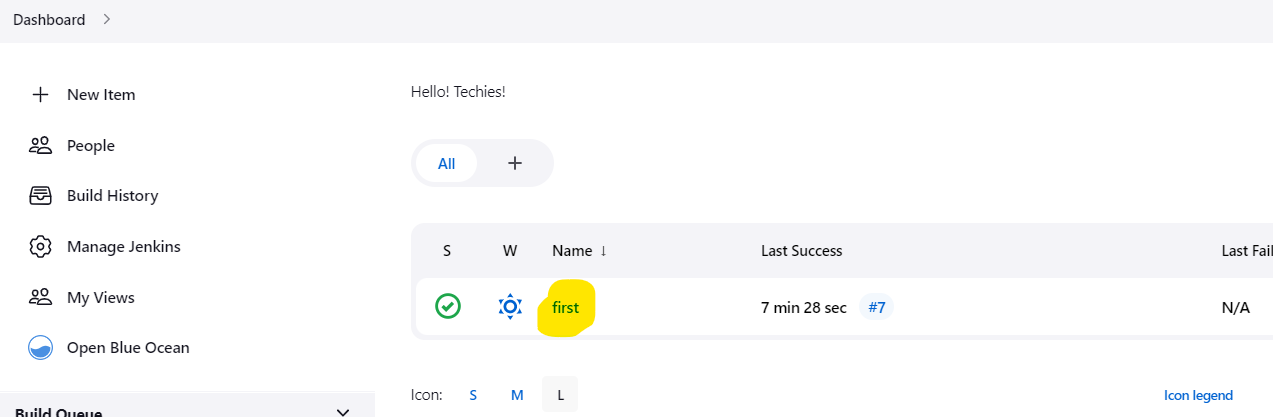
uname -a

cat /etc/\*release\*

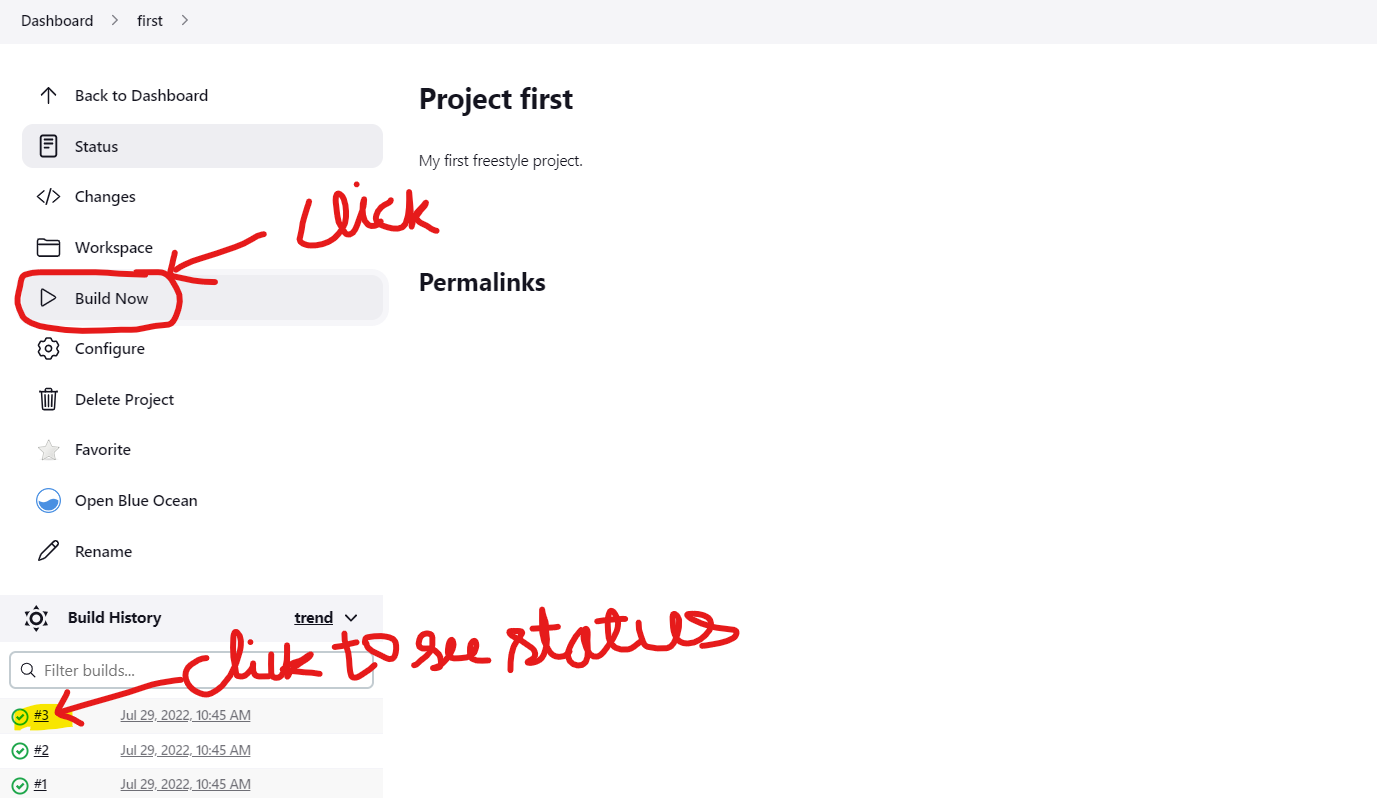


28. Back to DASHBOARD to see your created Project named as first.

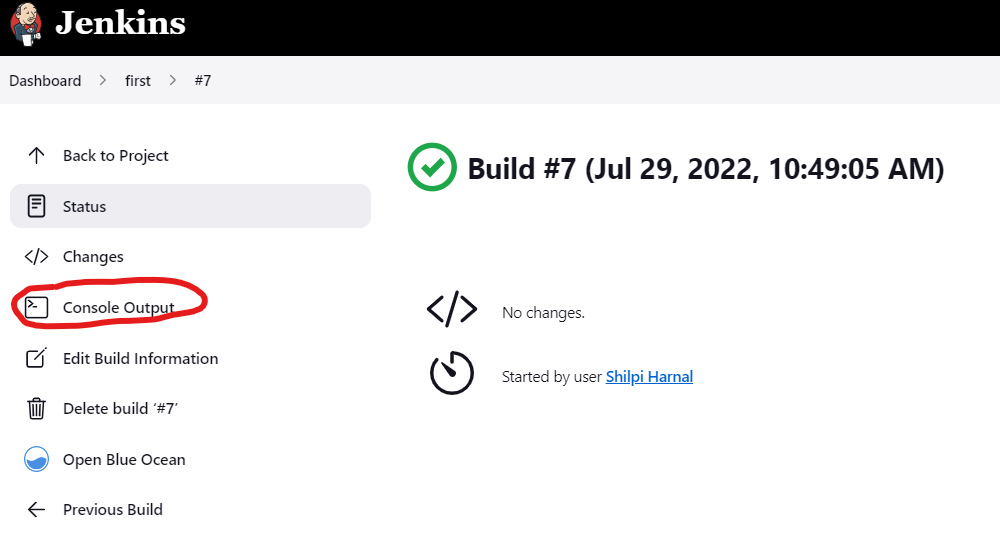
Click on project name: first



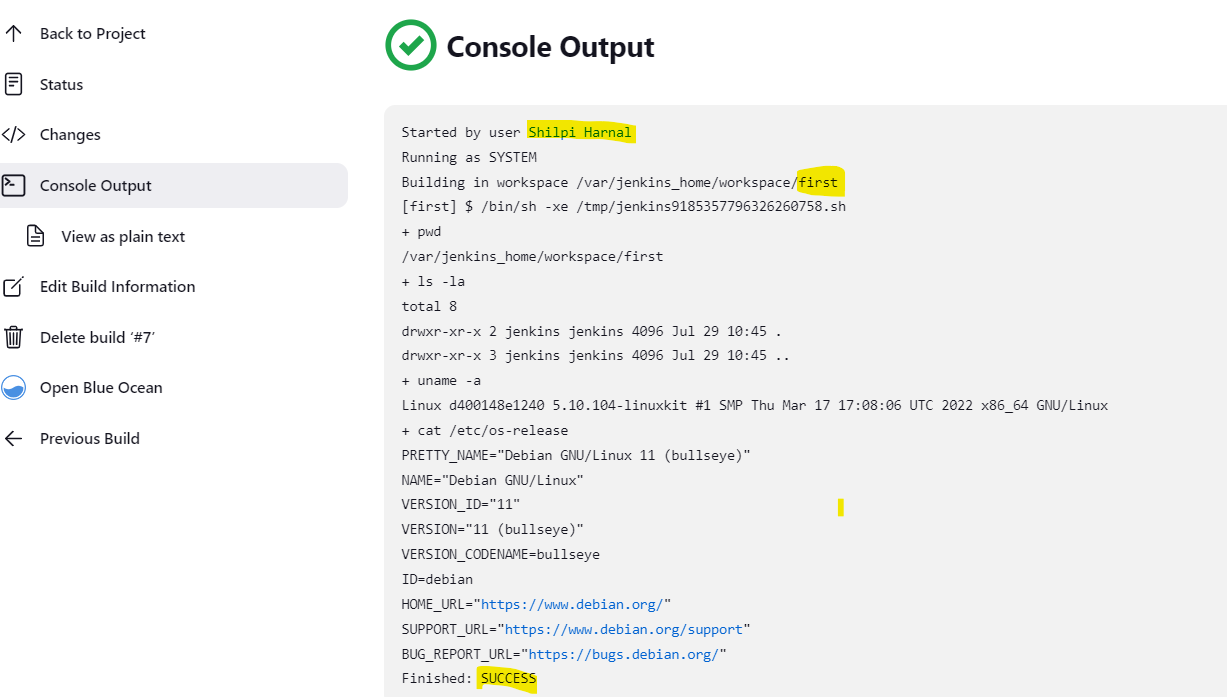
Click : BUILD NOW



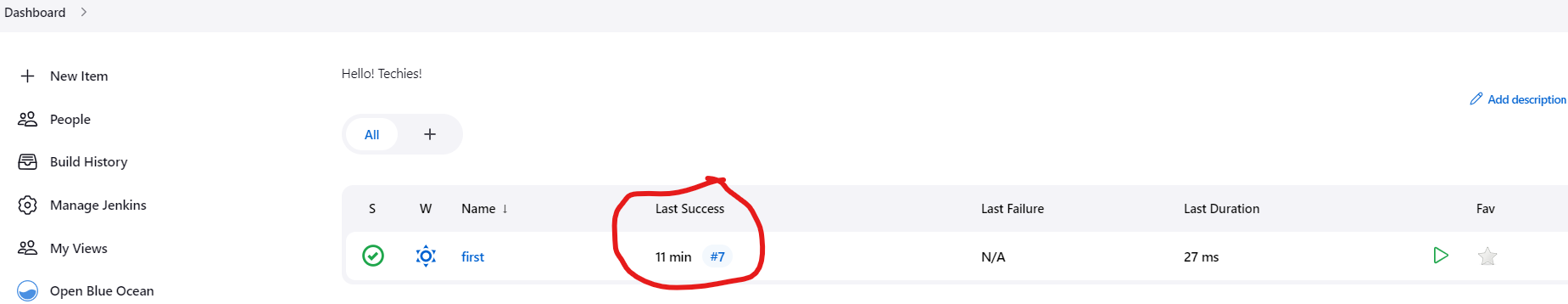
29. Click on any task. Here Build #7 indicates the number of times you have build the project.



30. Click console output:

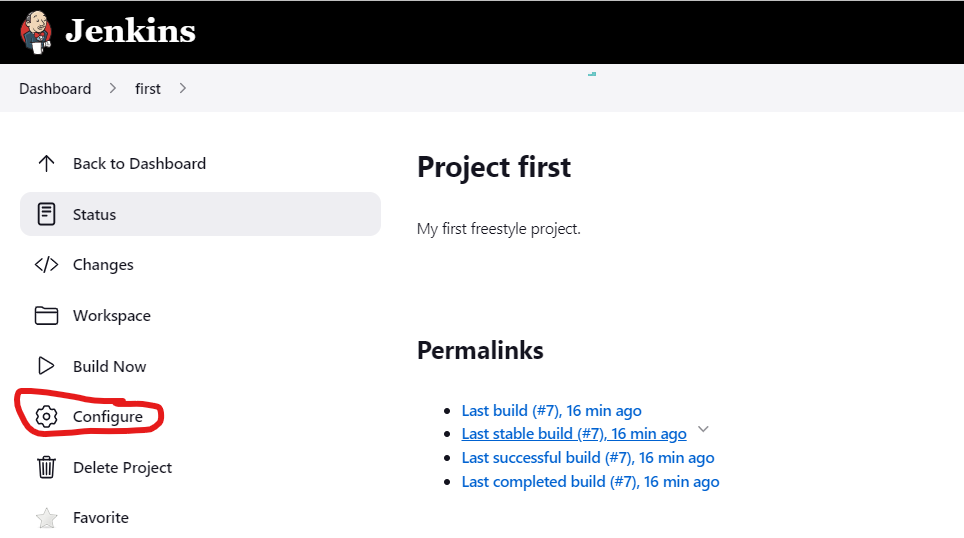


31. Back to DASHBOARD and check status as NO FAILURE

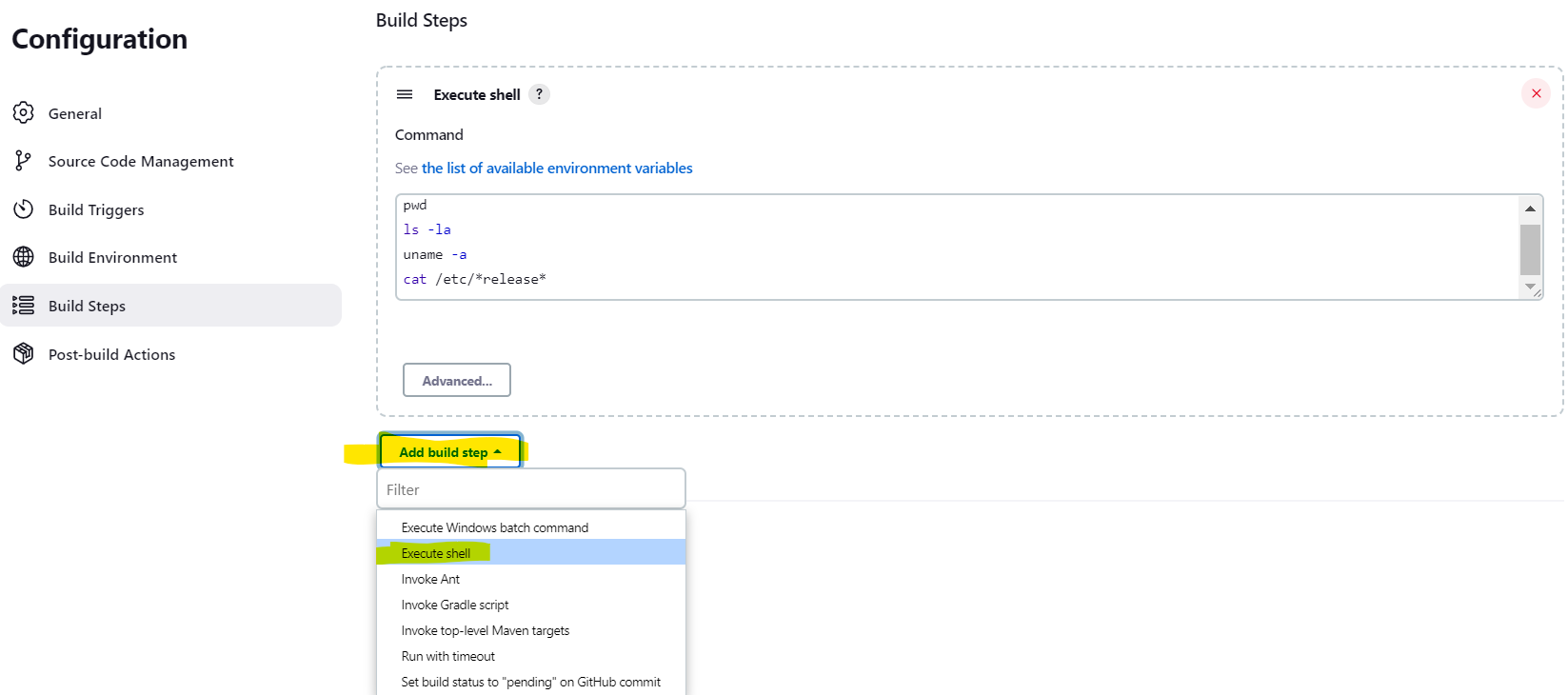


32. Lets modify the project to add some more shell script and rebuild it

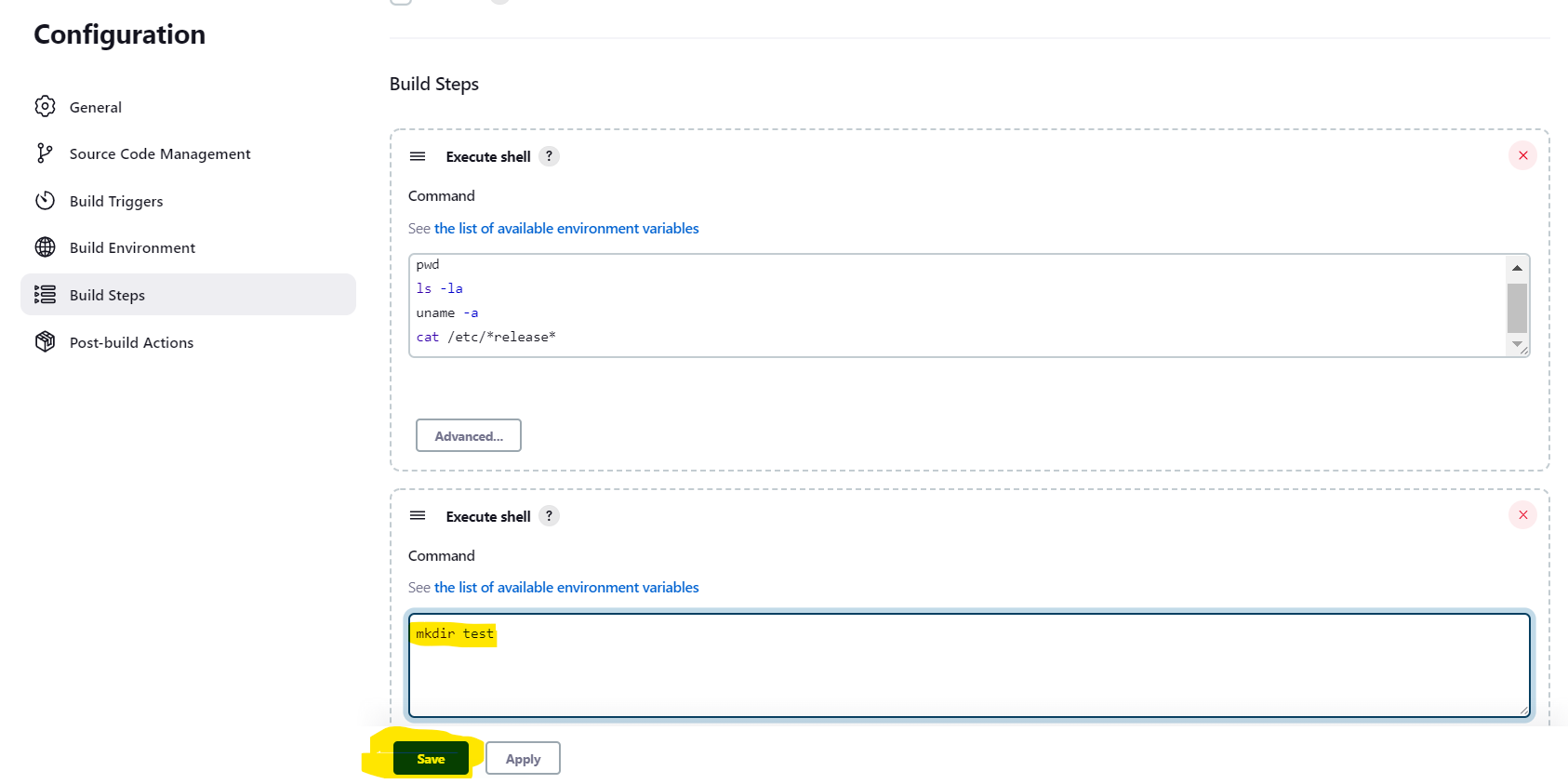
Click CONFIGURE:



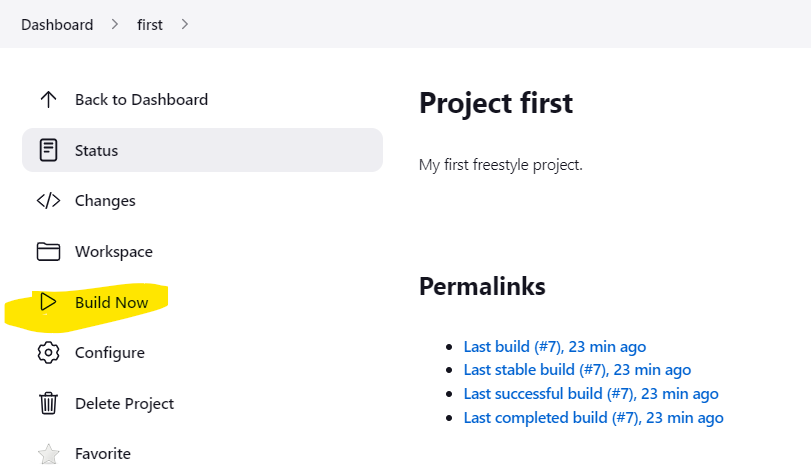
33. Click ADD BUILD STEP -> EXECUTE SHELL



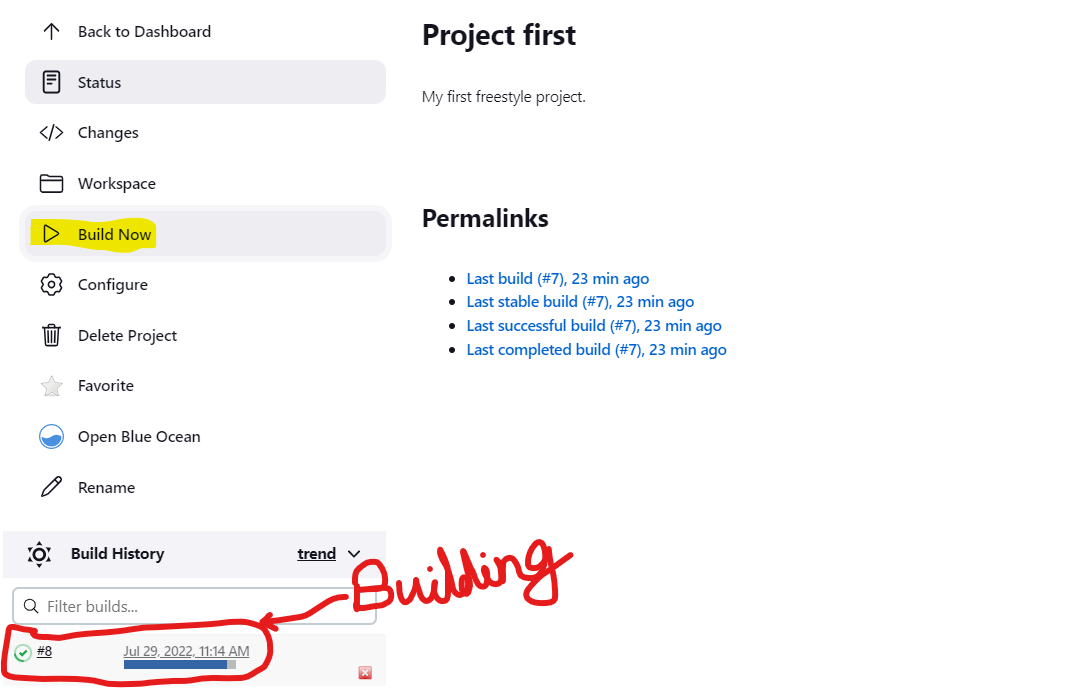
34. Add a cmd to create a directory: mkdir test



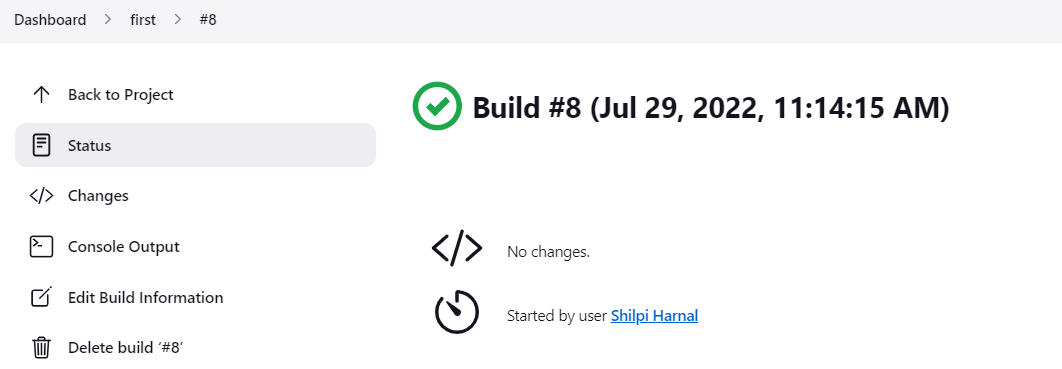
35. Build it again



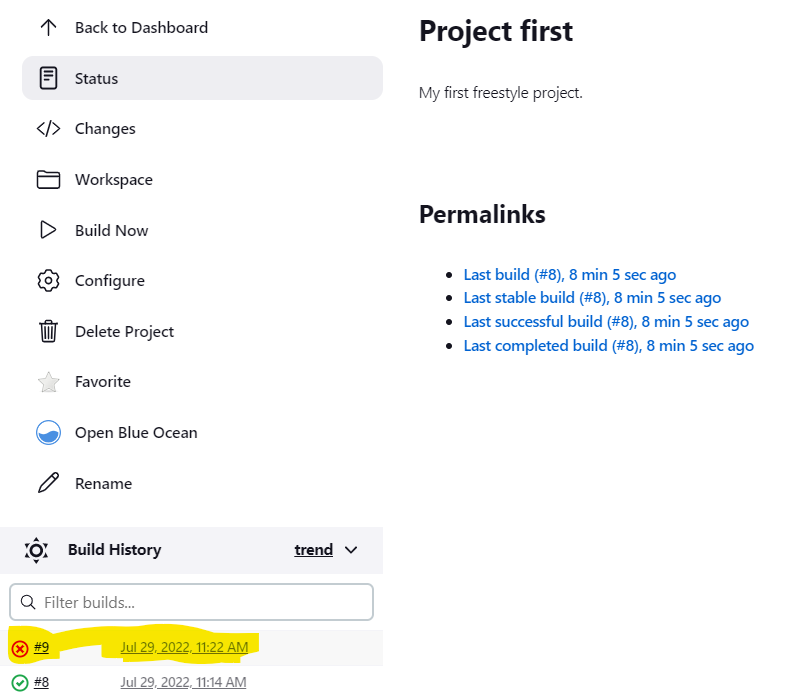
36. New version of project is building



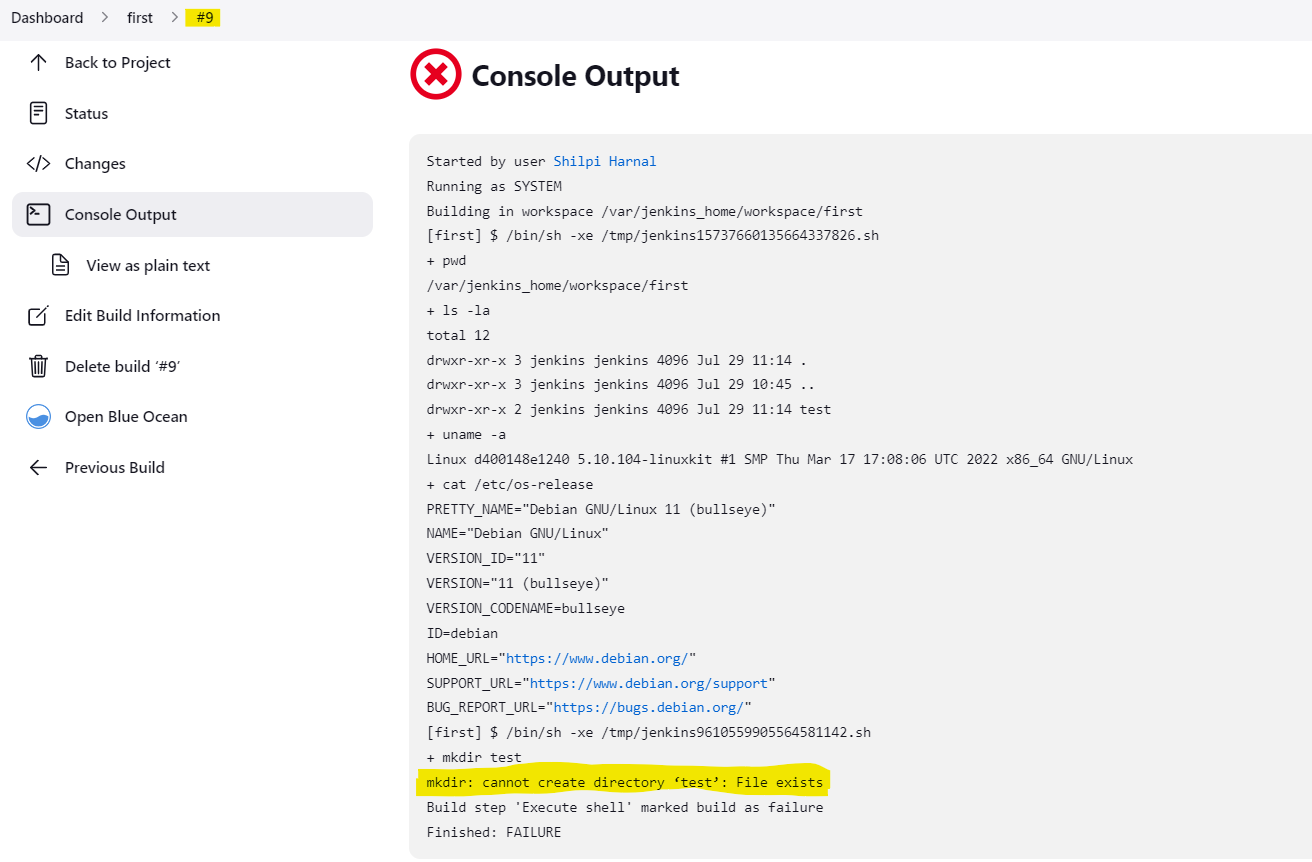
37. Here The project is being built for the 8th time. So click #8 to see status. You can also see console output for the same.



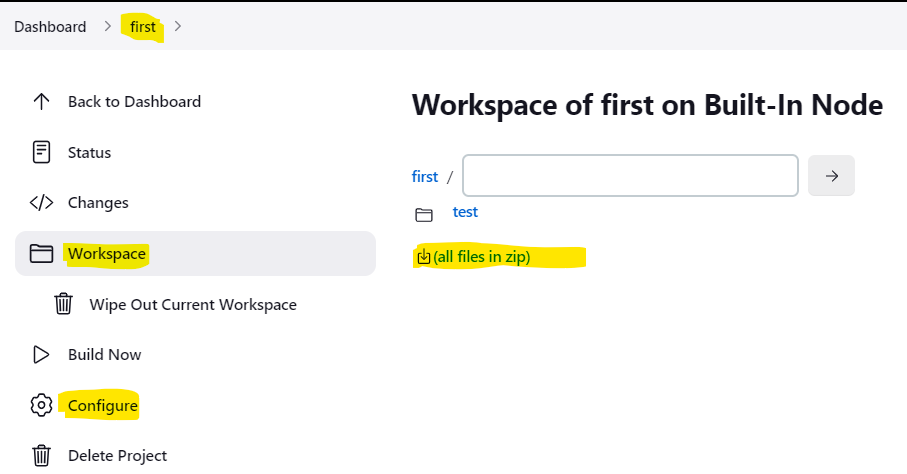
38. If you try to build the same project again, you will see failure as it will execute the **mkdir test** command again. And you cannot create the same directory again.



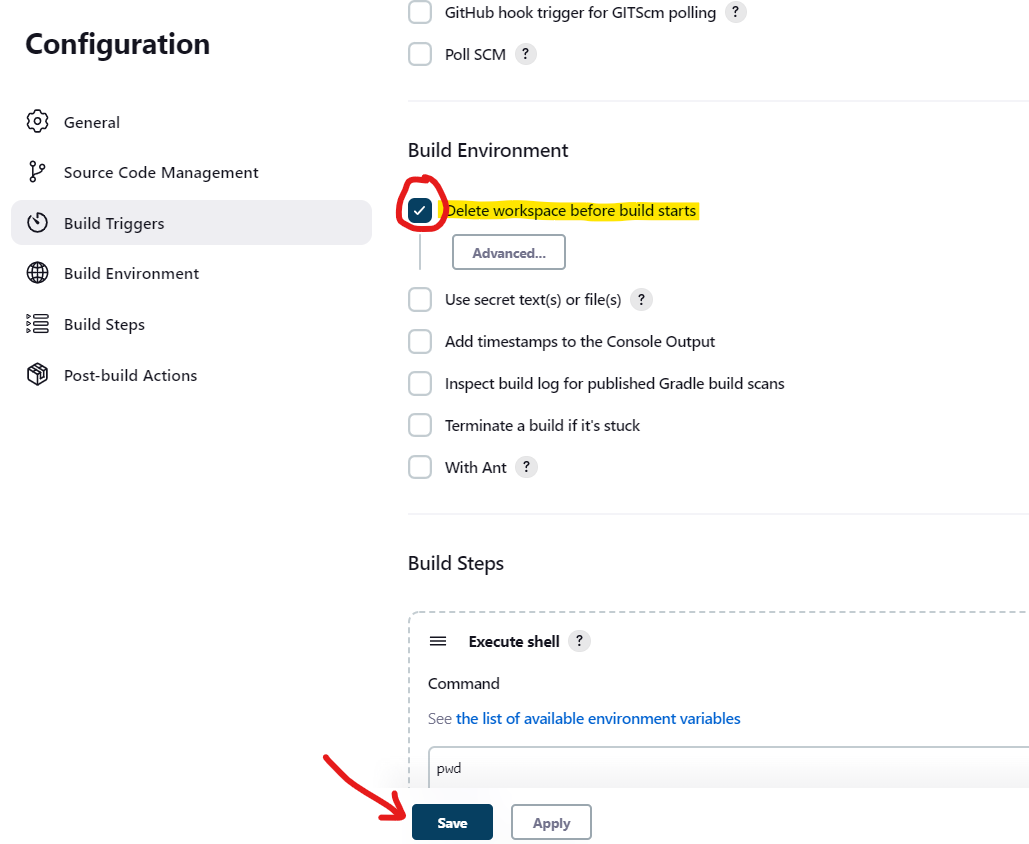
39. Click on #9 and then on CONSOLE OUTPUT to see status



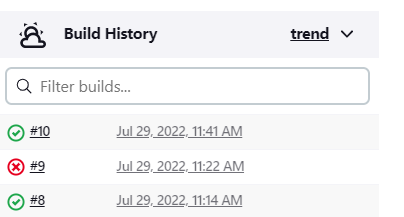
40. Click on the project name and then on WORKSPACE. Here you can get the option to delete all project files. But here we are not having any files in project.



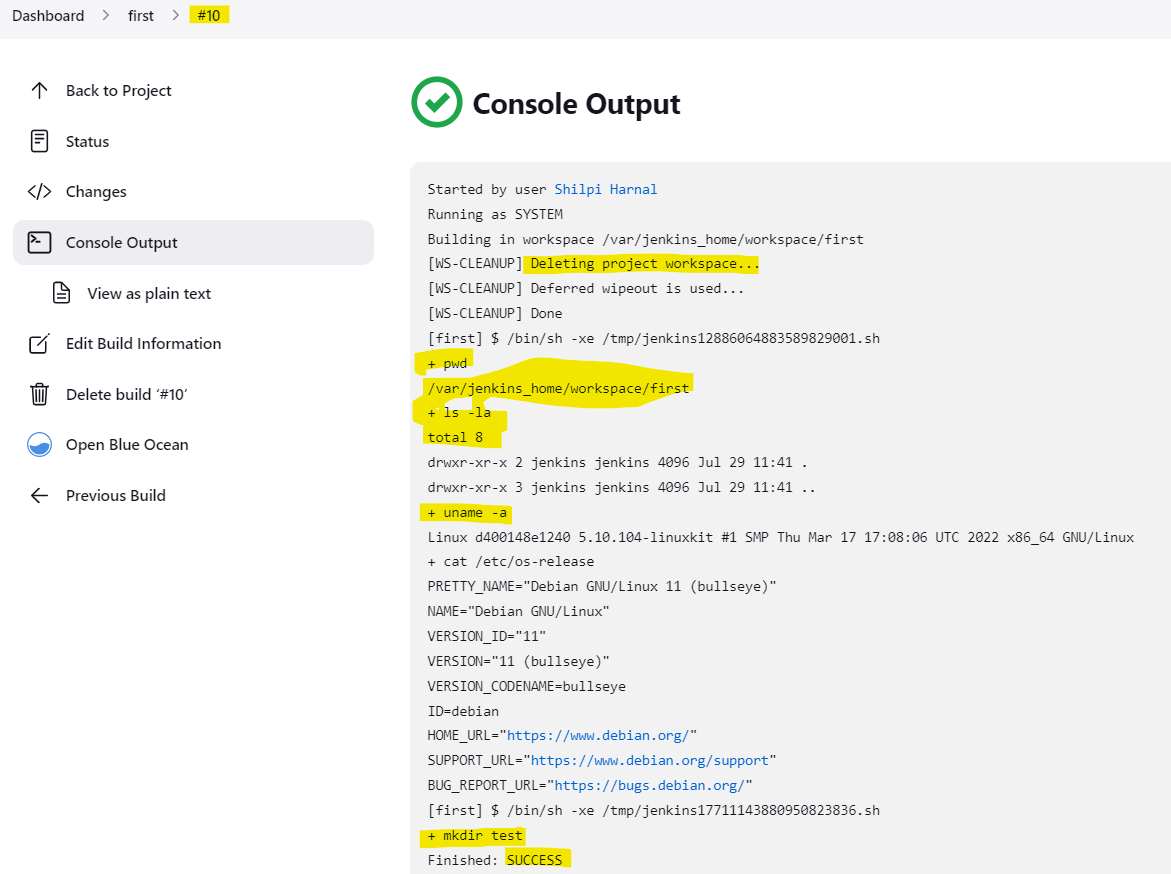
41. To fix the failure here. **We can choose to delete the existing workspace before building it again. This option is available in Configure as shown in figure above.**

****

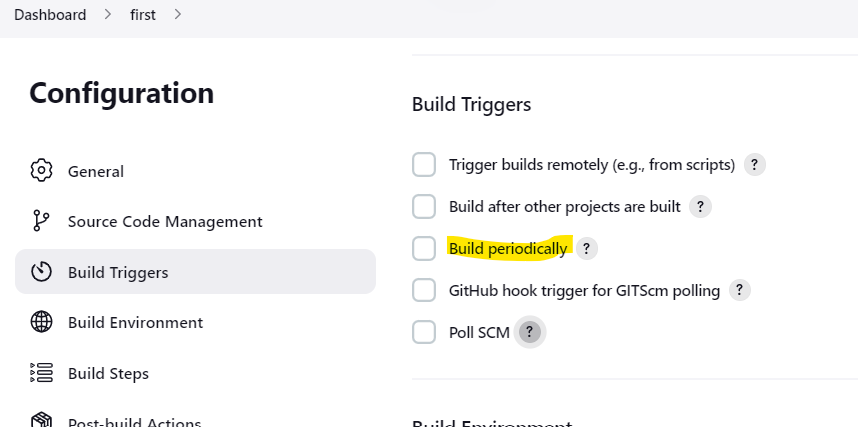
42. Click save and then BUILD AGAIN. This time it will run successfully as it will delete the earlier created test directory first and then rebuild it along with executing another shell script.



43. Click #10 and check status:



44. We can also use triggers from configuration to set triggers to run project.

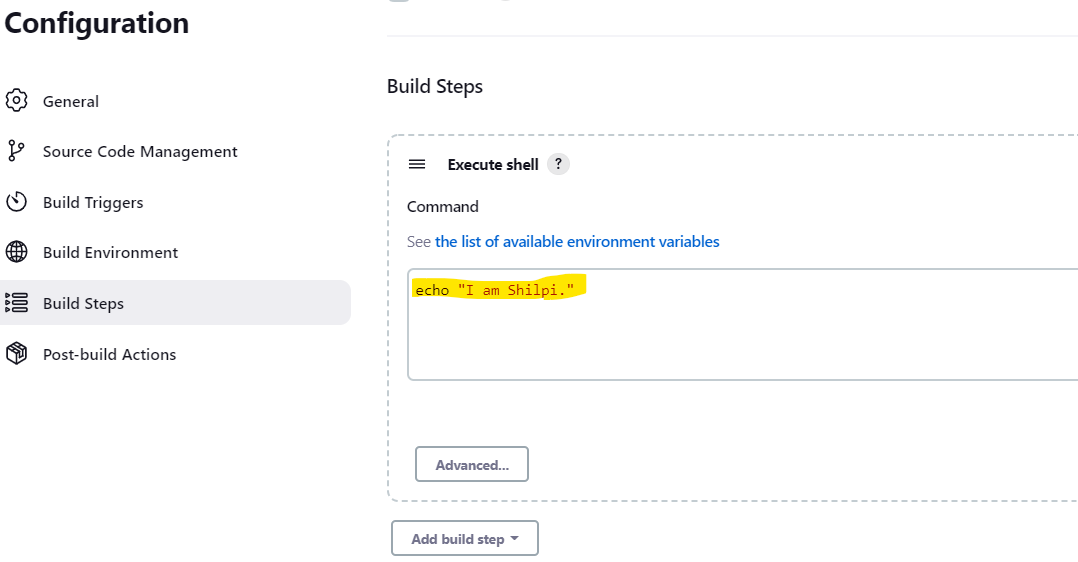


**Lets create another Jenkins Freestyle Project as second that will be executed as a trigger after the earlier created project: first is executed.**

45. Go to DASHBOARD, click NEW ITEM, and add a new project as **second**:



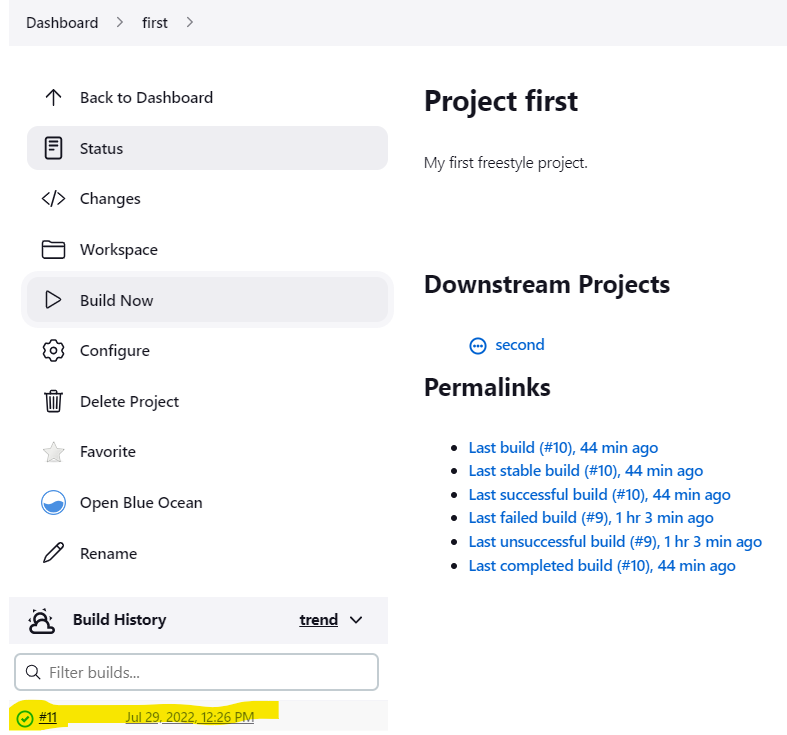
46. Add some script:



47. Add a trigger to execute it after **first** project and SAVE.



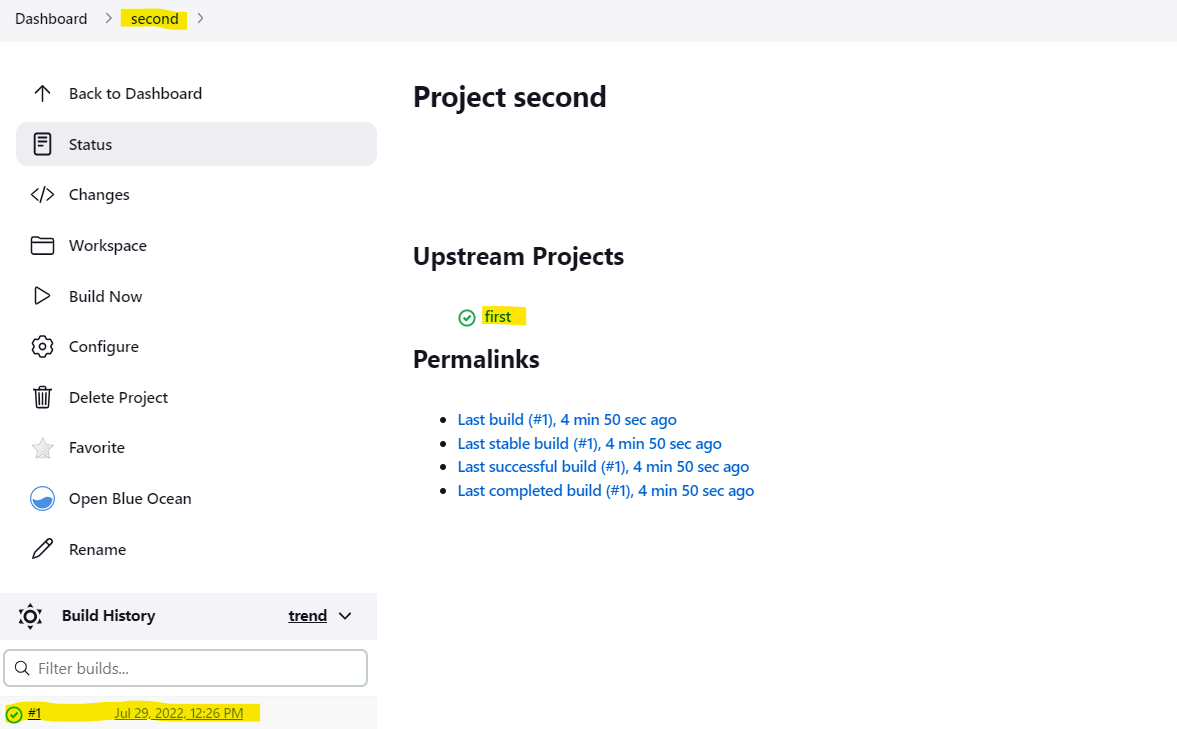
48. Now go back to the dashboard and build the first project again and it will automatically trigger the second project.



49. Check the dashboard for the status of both projects as first has triggered the second project. Earlier second was unchecked.



50. Click on second, and check status as already built by first.



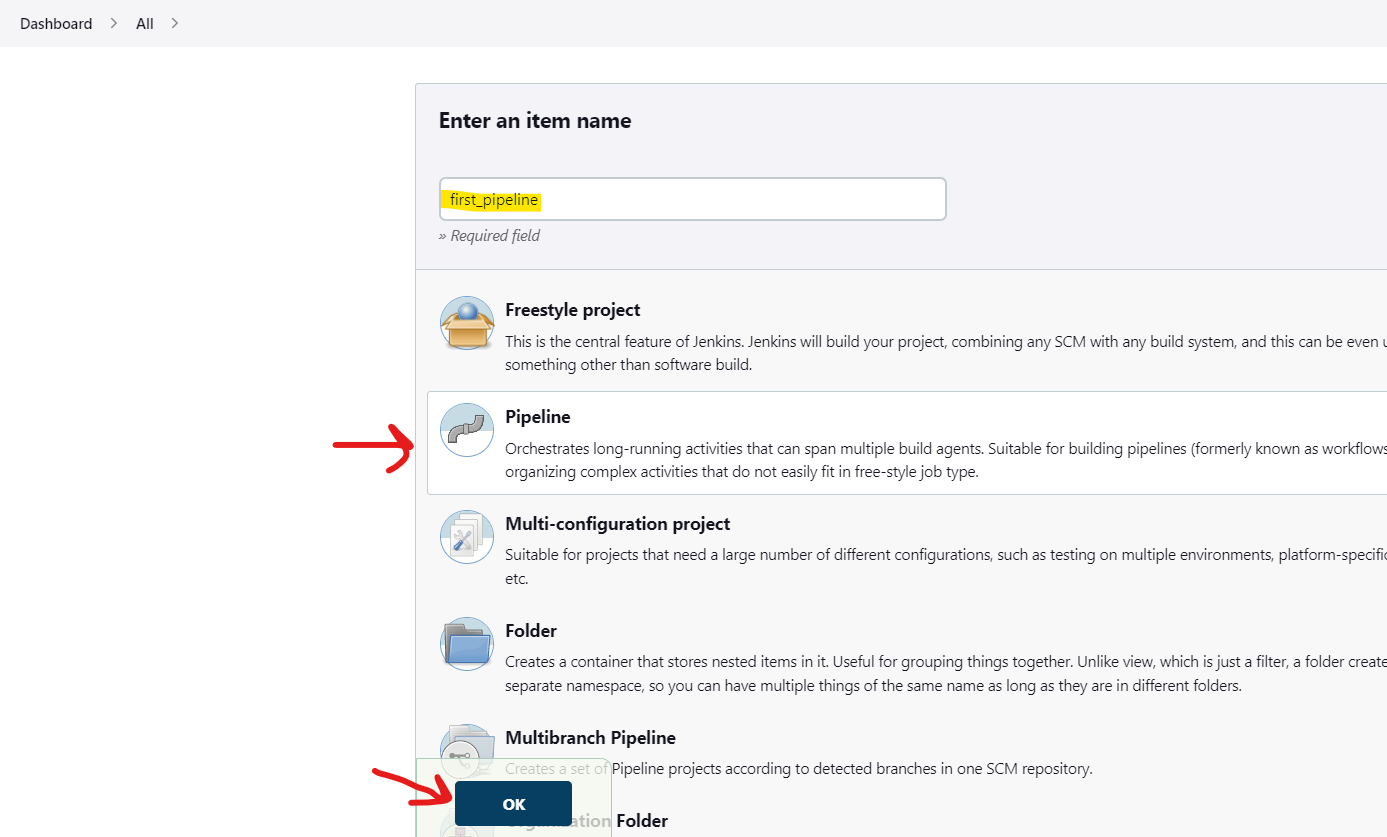
51. To check console output of second click on #1:



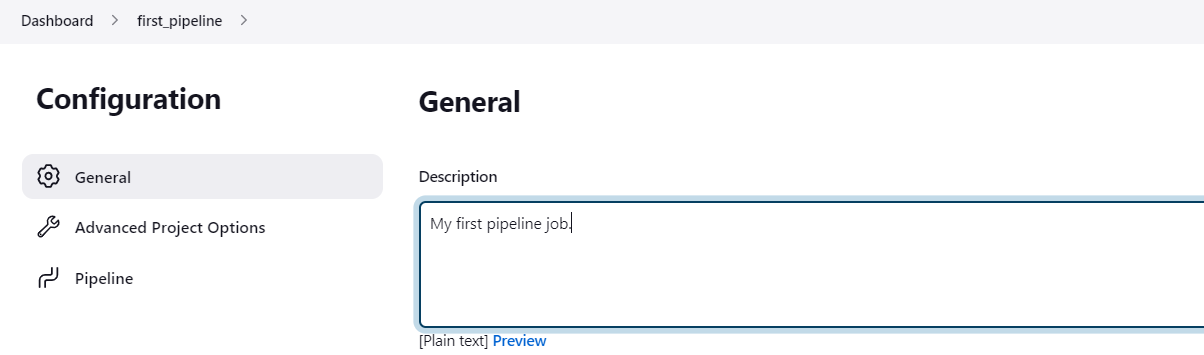
**PROJECT2: Jenkins Pipeline**

**Jenkins Pipeline Projects are used for CI/CD.**

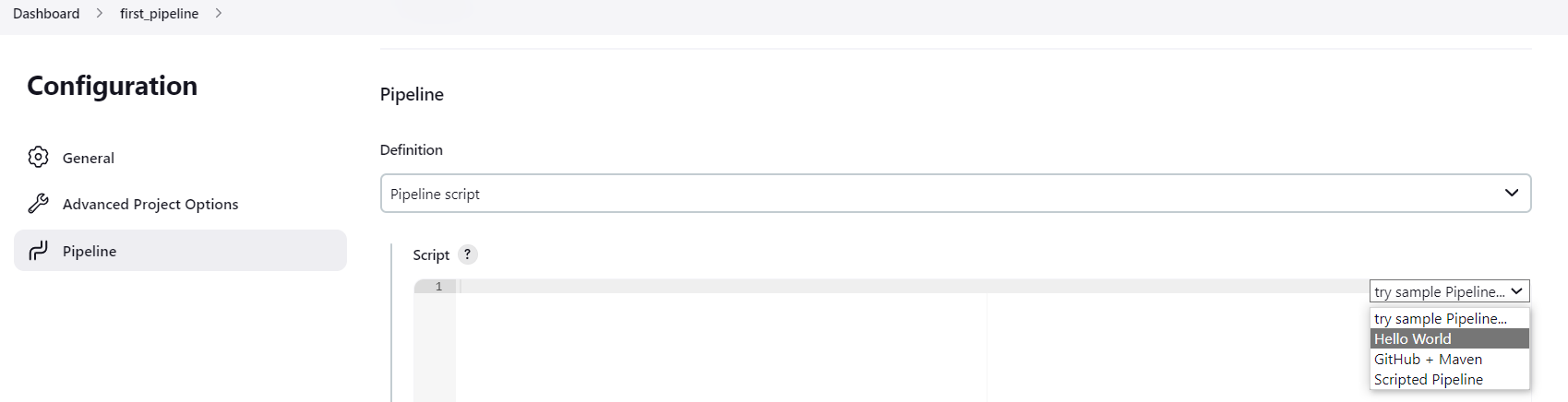
1. Add a new pipeline project as first\_pipeline. Click save.



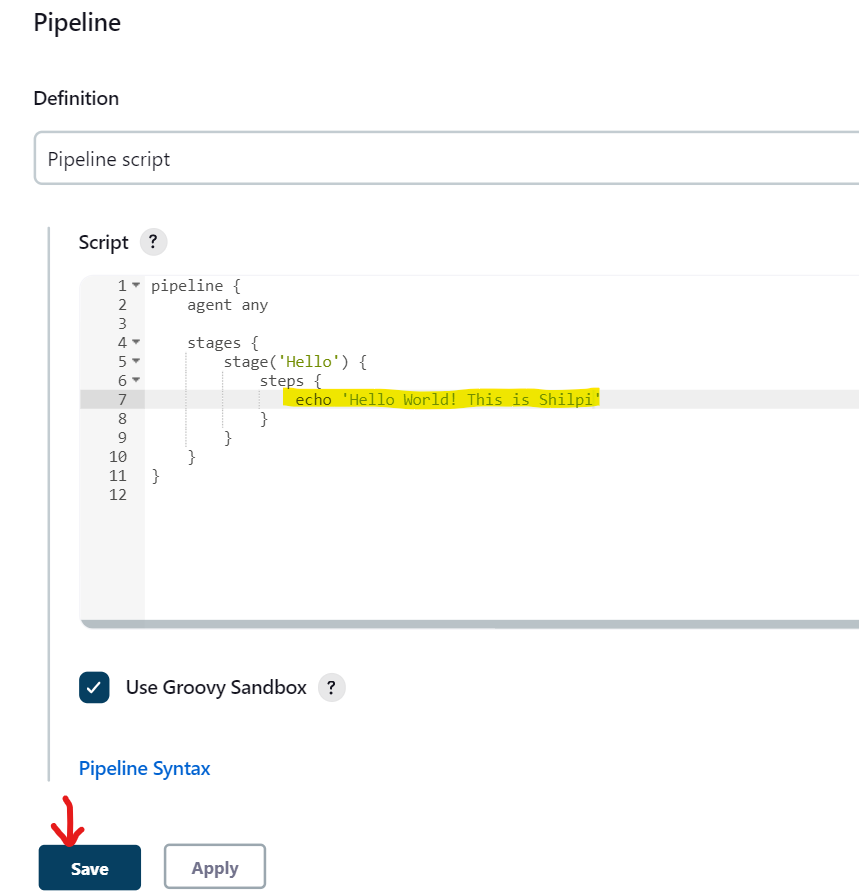
2. Add some description.



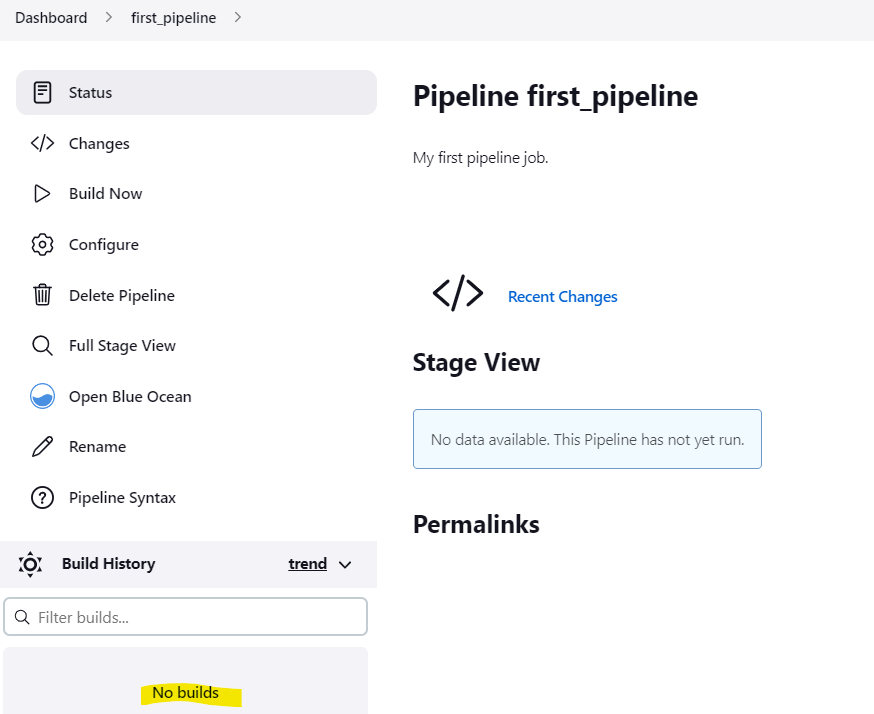
3. Add sample pipeline script.



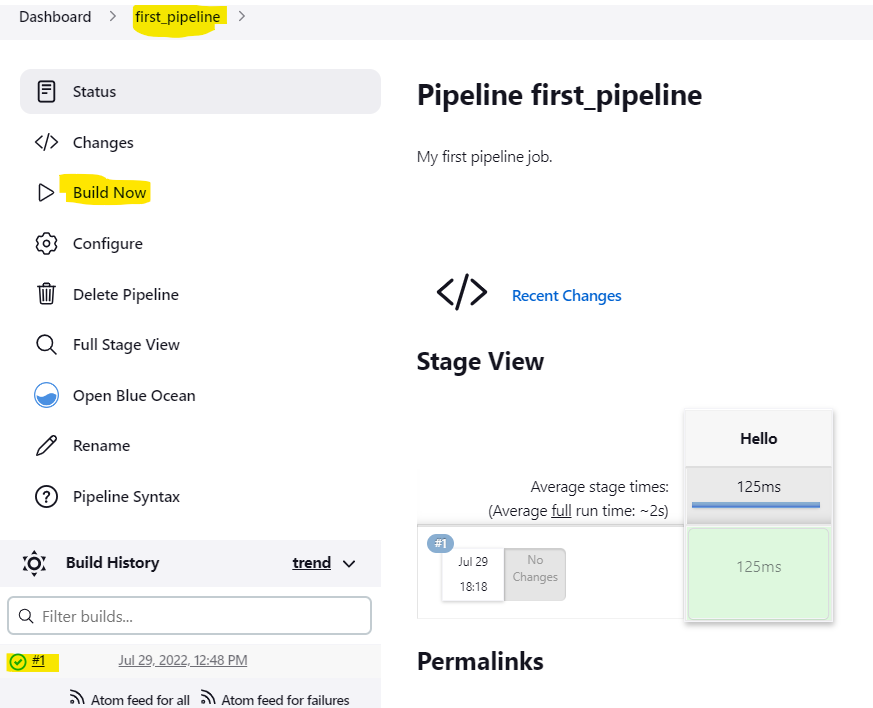
4. Sample pipeline script:



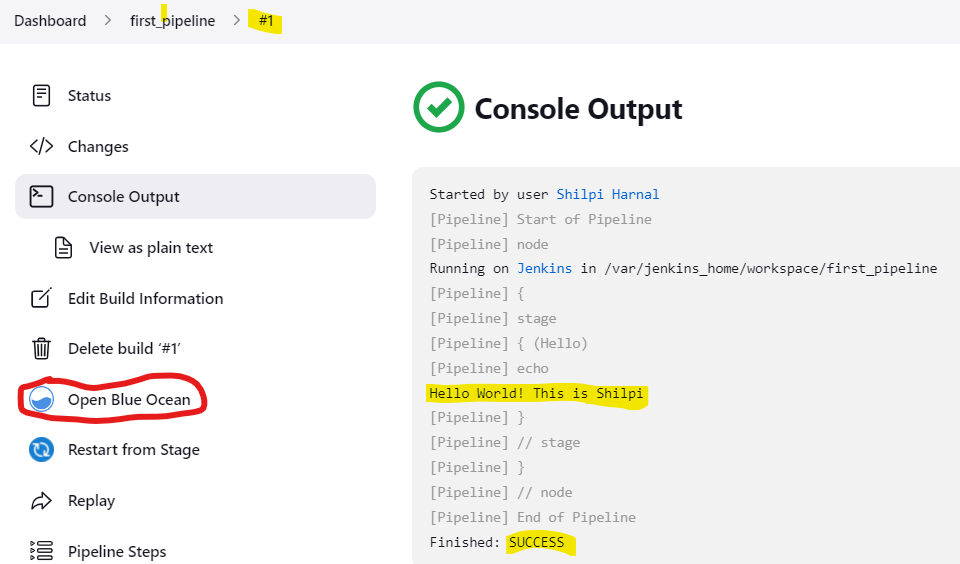
5. Initial pipeline is created, with no build yet.



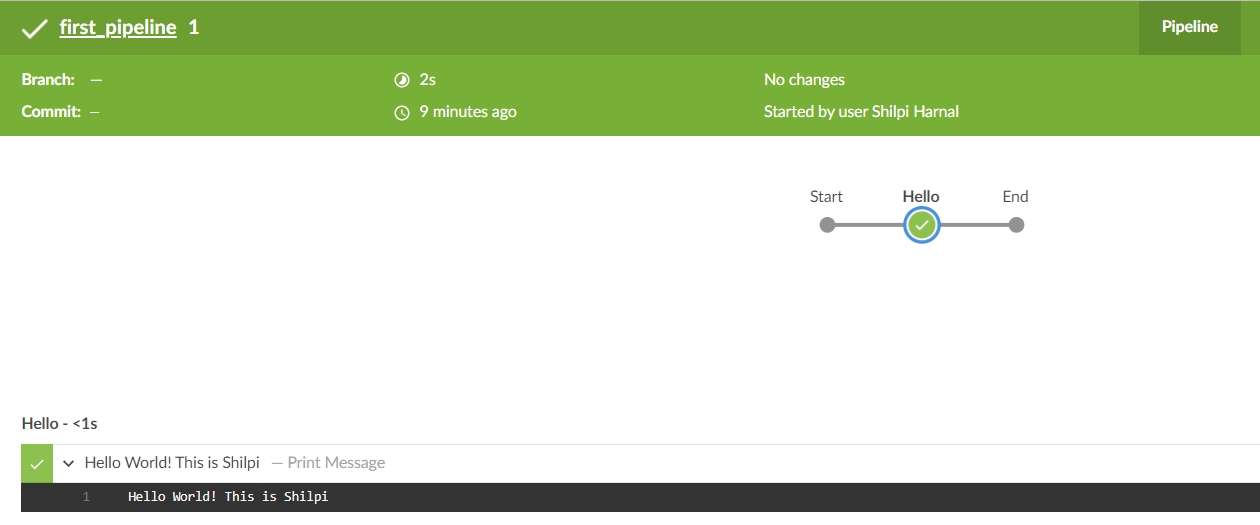
6. Build it Now and rest is same:



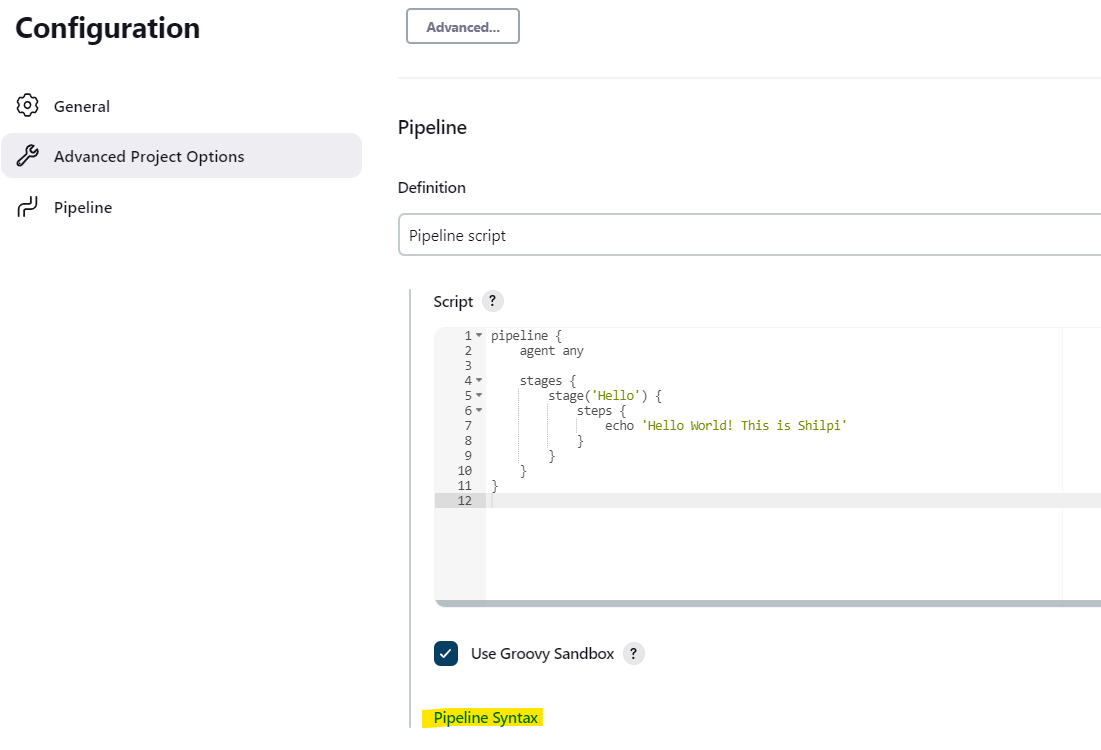
7. Click #1 to check console output.



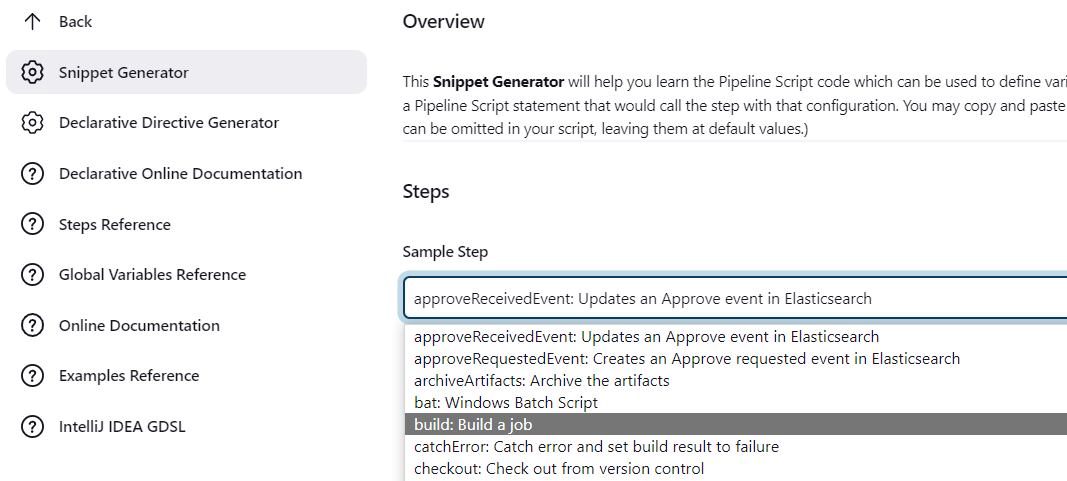
8. Click Open blue Ocean to check the output:



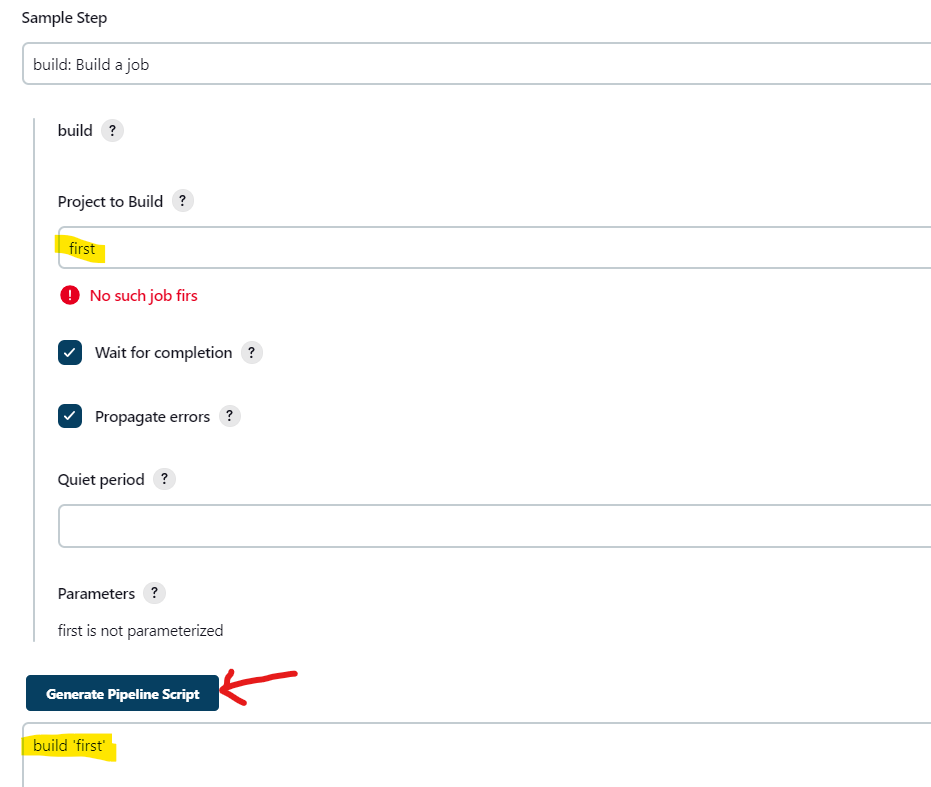
9. Go back and Lets configure pipeline to run earlier created freestyle project first that will automatically triggers project second. Click on Pipeline Syntax.



10. From sample step: Build a job.



11. Select project name to build, that is first in this case. Now click Generate Pipeline Script to check and copy the command to be added to build first project, i.e. **build ‘first’**:



12. Go to previous tab where we had first\_pipeline script and add a new stage to build first project as:

stage('Invoke and build: first'){

steps {

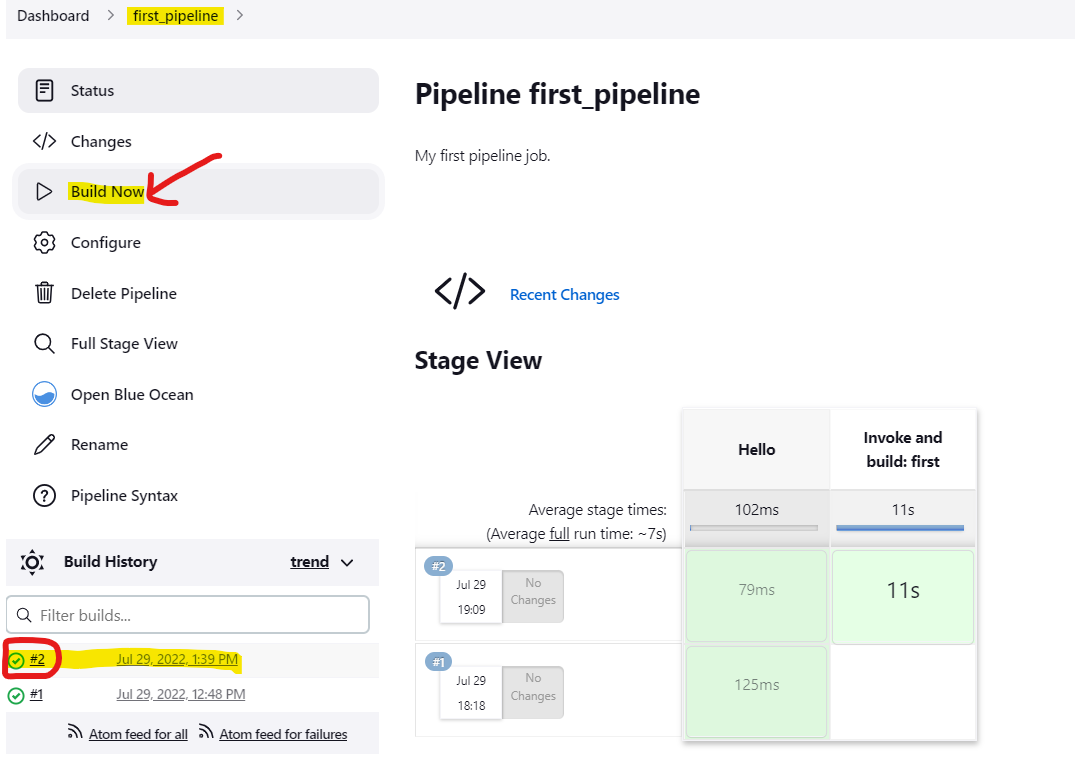
build 'first'

}

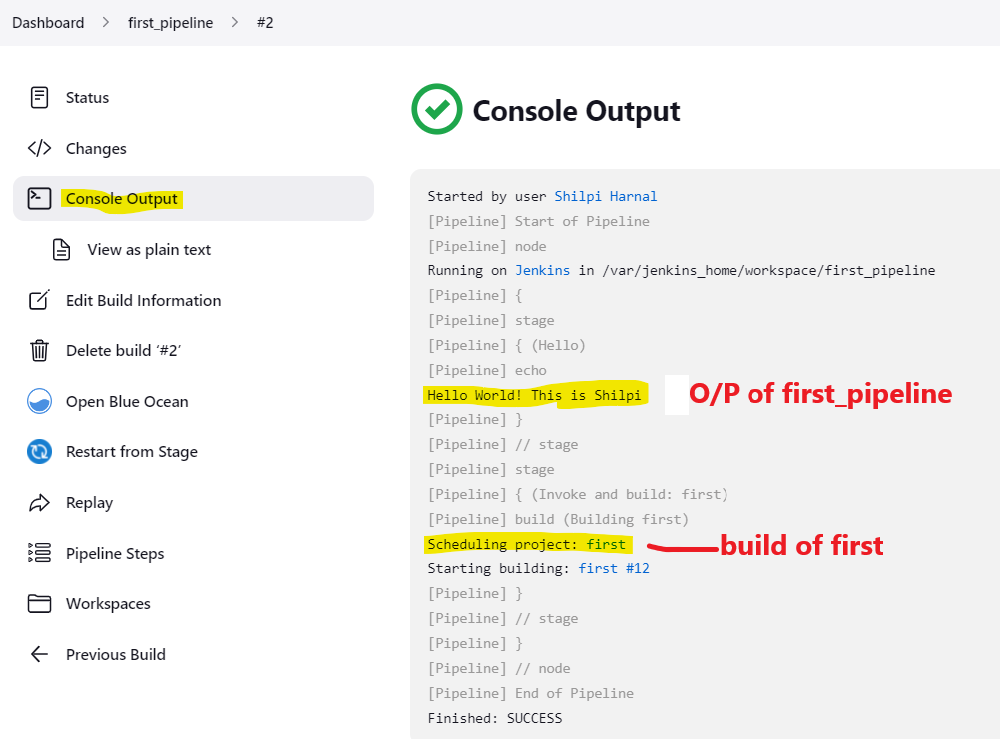
}



13. Click Save and build the first\_pipeline project again. That in turn builds first and that will build second.



14. Click #2 to check console output.

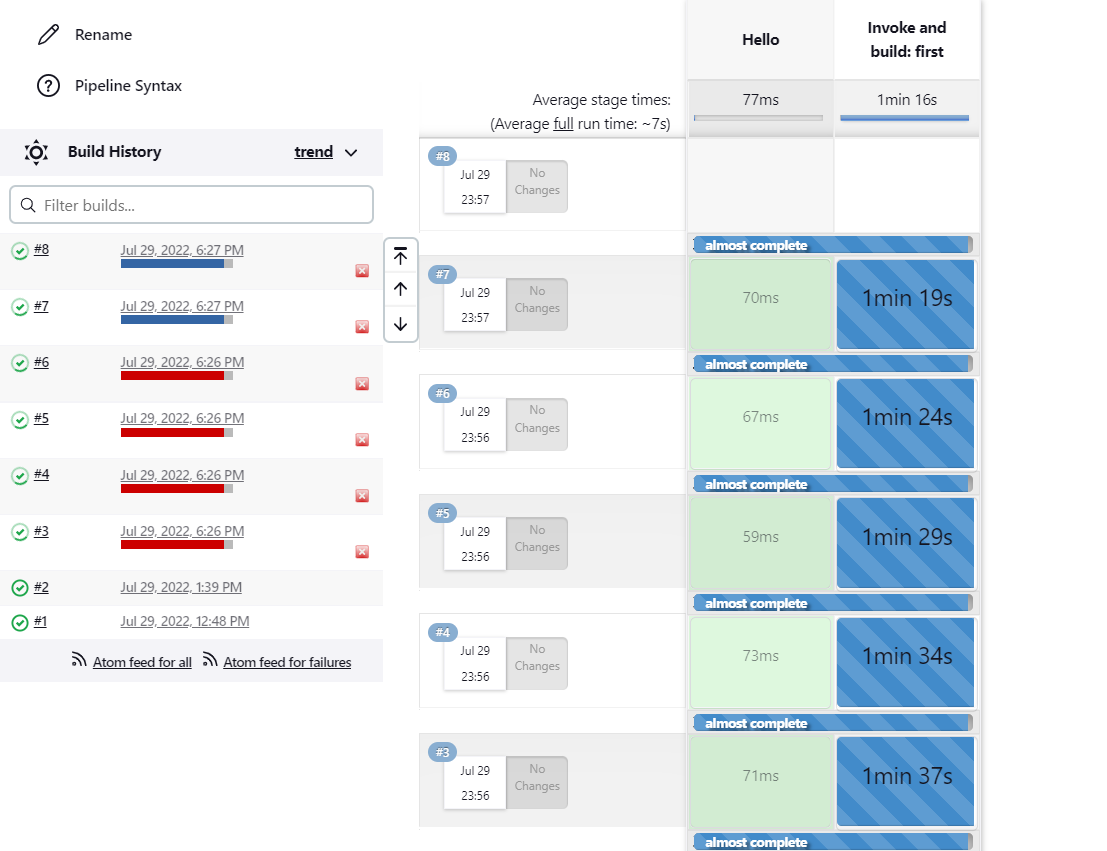


15. You can go back to Dashboard and check the status of the first project pipeline that has also been executed and the second project pipeline triggered by the first.

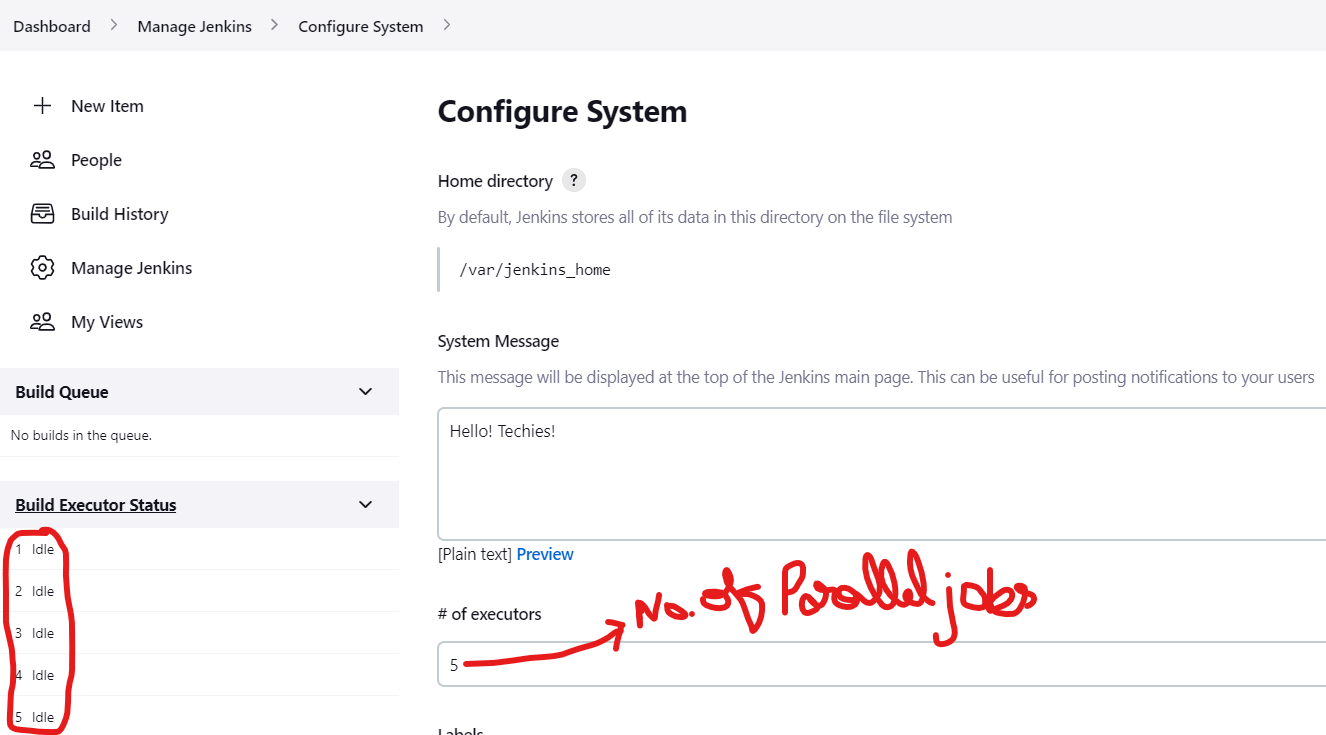
16. Go back and Lets configure pipeline again to add some delay with sleep command as:



17. After SAVE build the pipeline again. This time it will be running with delay. Try to build multiple pipelines with multiple clicks parallely as:

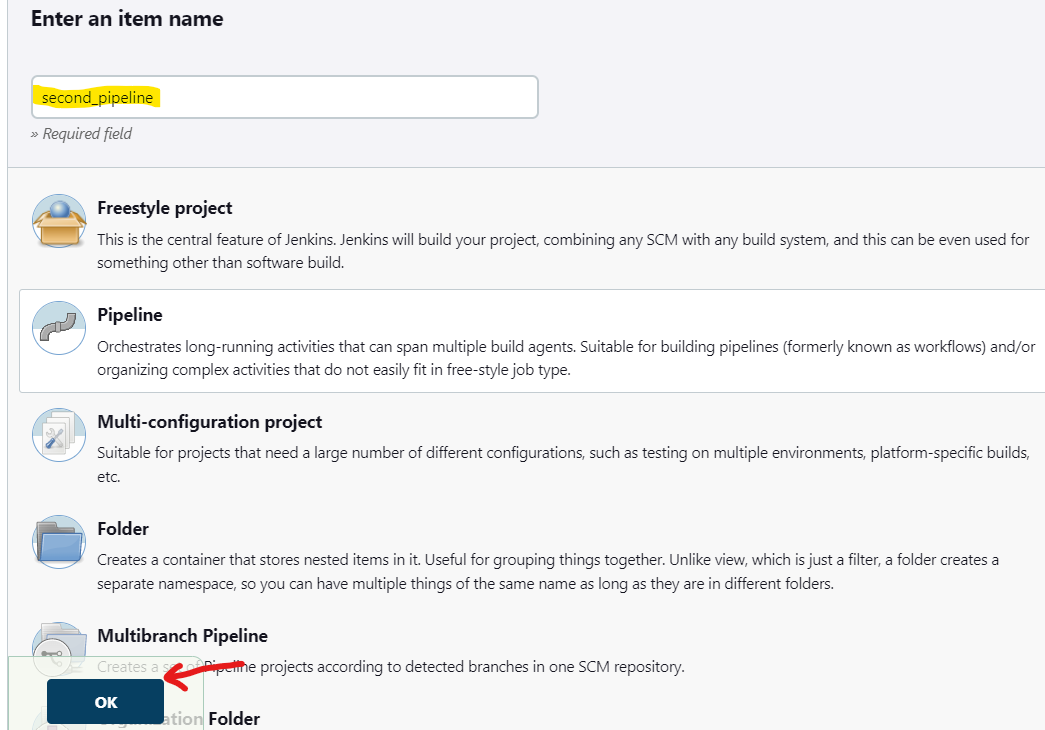


18. But it will only execute 5 pipelines at a time because of jenkins general settings as done earlier from MANAGE JENKINS. If required, you can change the settings to a higher number.

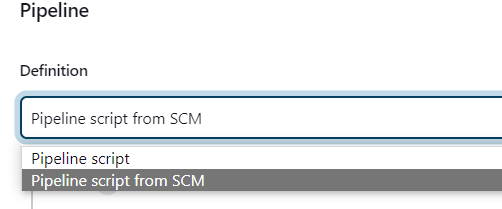


**PROJECT3: Jenkins Pipeline**

1. Add a new pipeline project as second\_pipeline. Click save.



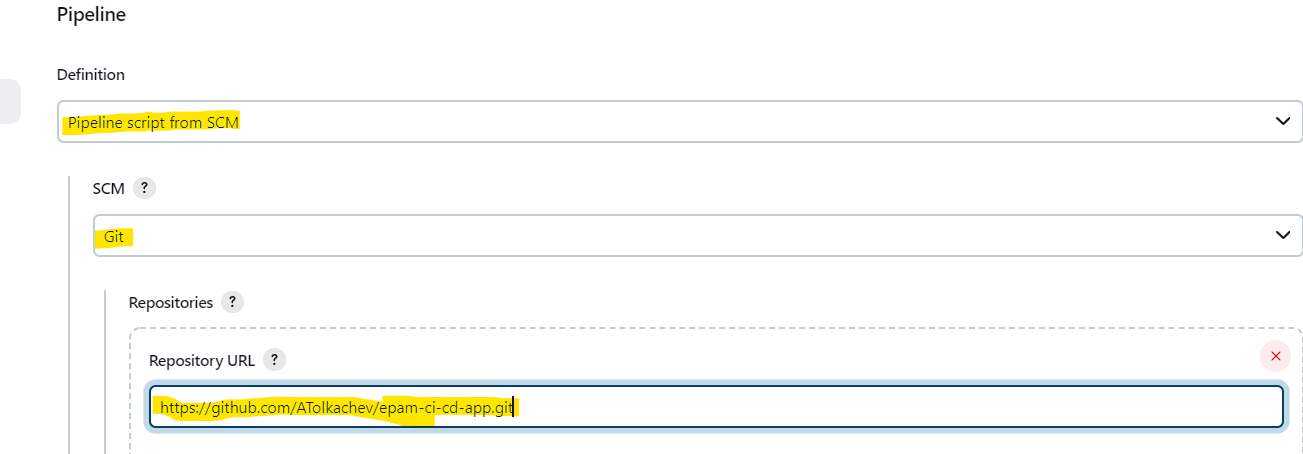
2. This time, build pipeline from SCM instead of script by selecting **Pipeline script from SCM**. This means building a pipeline from git-link.



3. Select Definition: **Pipeline script from SCM** and SCM: **Git**

Paste Repository URL: <https://github.com/ATolkachev/epam-ci-cd-app.git>

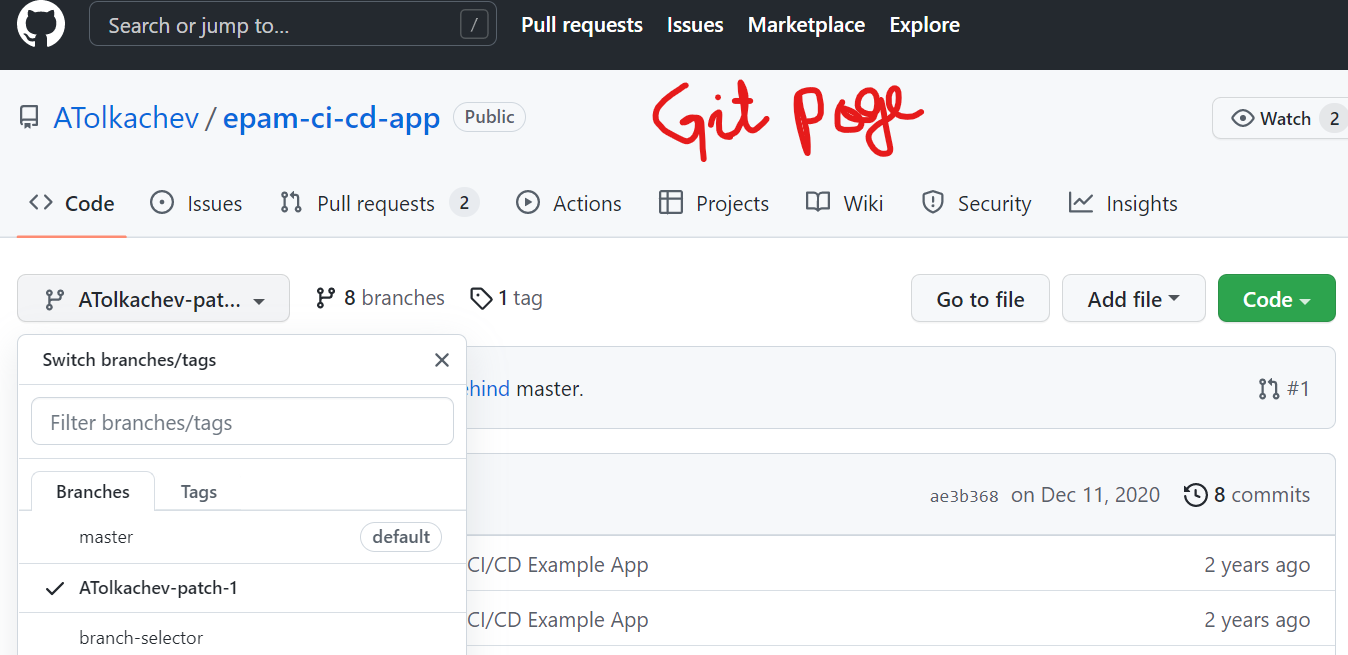
And **SAVE**



4. Build the pipeline and you will get an error because of some required plugin i.e. Kubernetives. Kubernetives plugin is required as it is mentioned in Git - Jenkinsfile but missing with Jenkins pipeline.

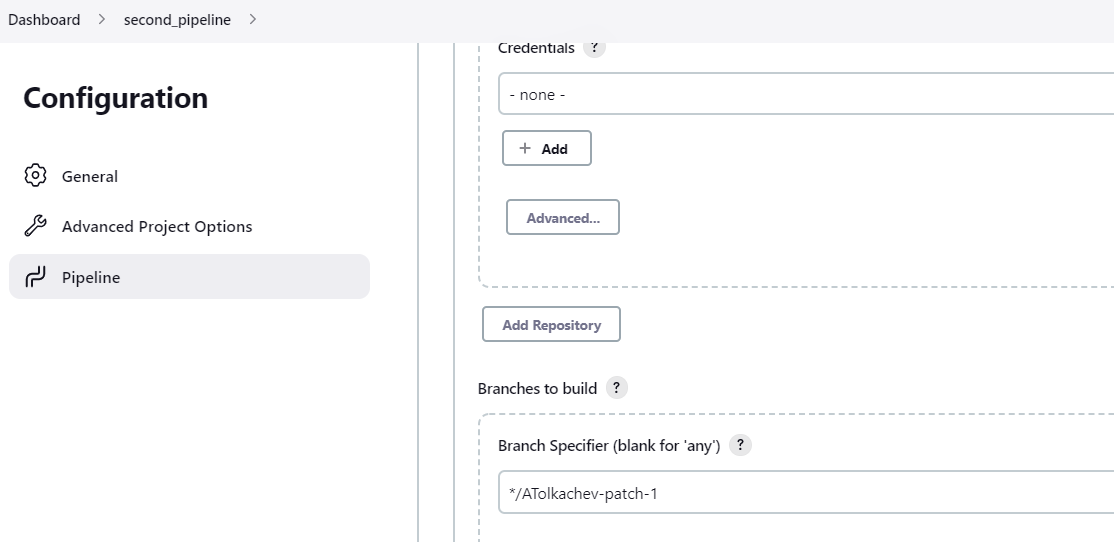


5. Check Jenkinsfile path from Git as it is not having that plugin and reconfigure pipeline:

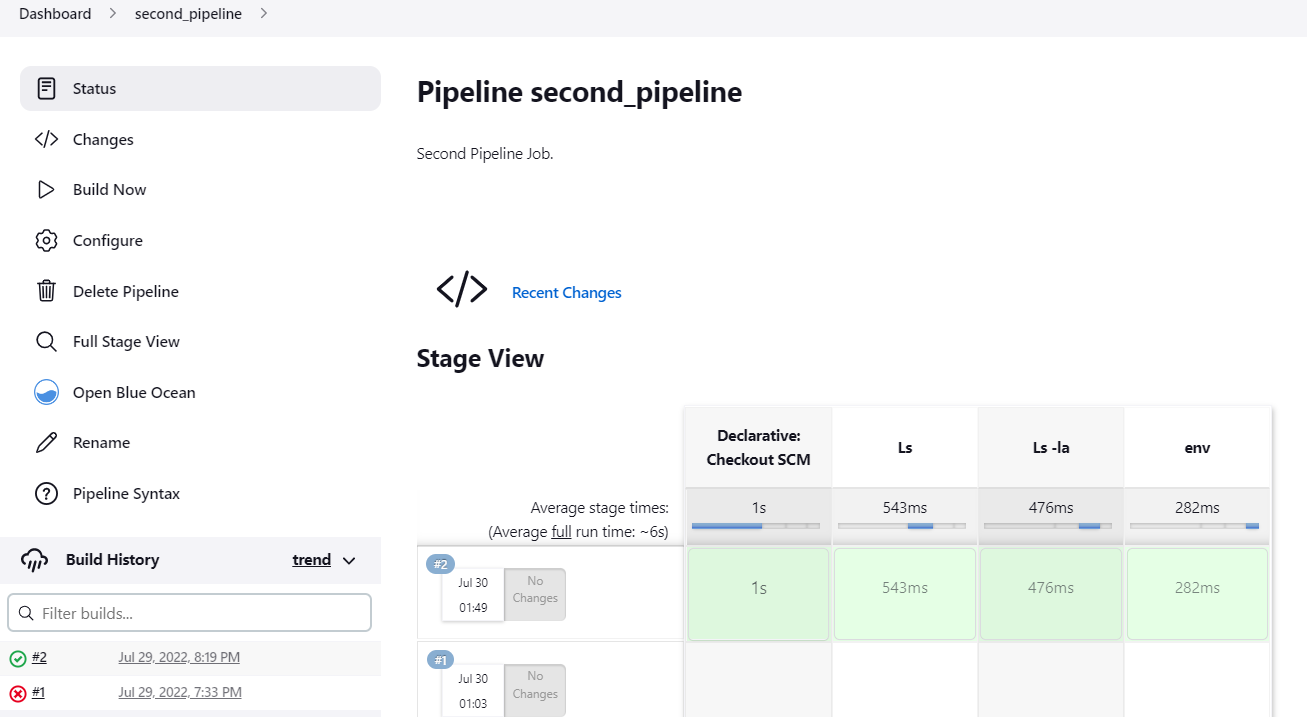


6. Click second\_pipeline -> Configure

Change Branch Specifier from “\*/master” to “\*/ATolkachev-patch-1” and SAVE



7. Rebuild the second\_pipeline



8. So the pipeline is now up and running from github repository.