## School of Computer Science

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES DEHRADUN, UTTARAKHAND**



Continuous Integration and Continuous Delivery Lab

## Lab File (2023-2024)

**for**

**5th Semester**

**Submitted To:**

Dr. Hitesh Kumar Sharma

**Submitted By:**

Arman Bisht

B. Tech. CSE [DevOps] 500093031

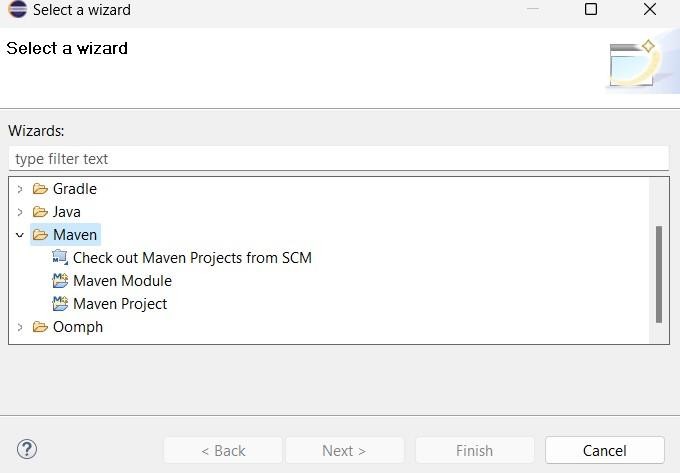
R2142210141

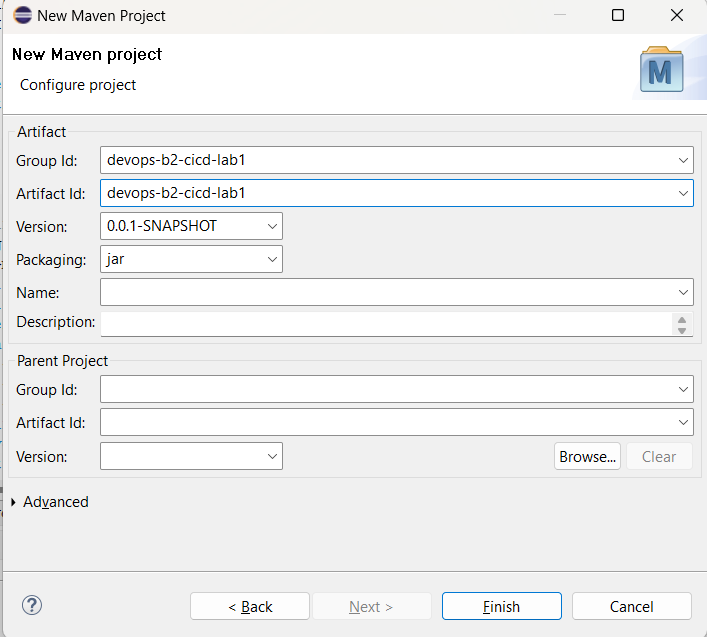
B2- Devops (Non-Hons)

## LAB EXERCISE - 1

**Aim:**

1. **Create a Maven Project in eclipse.**

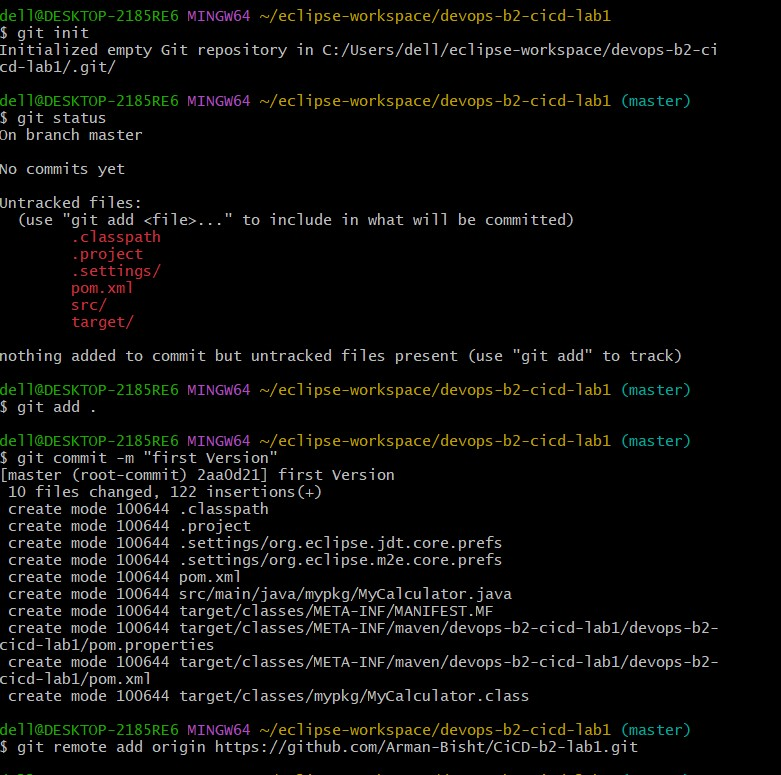




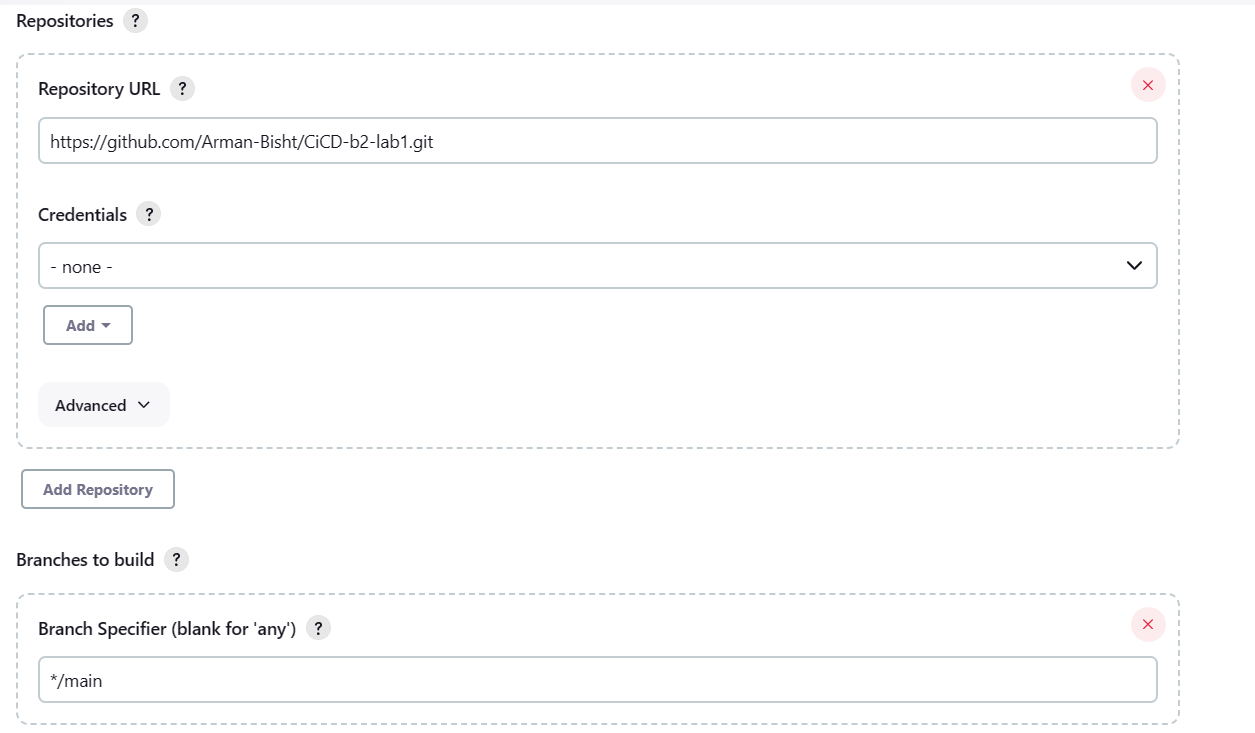
****

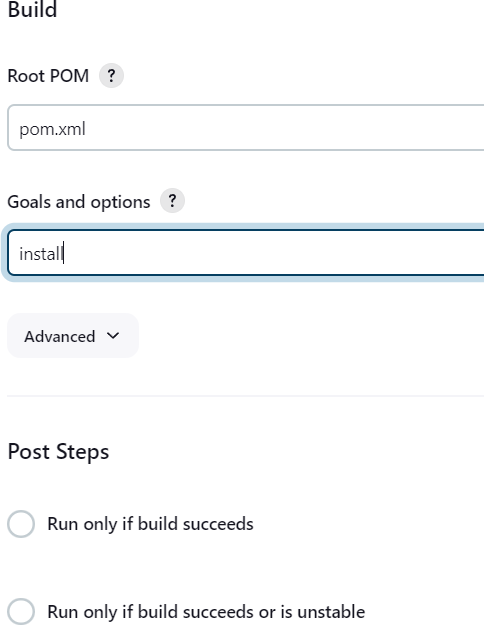


1. **Push project on github.**



1. **Start Jenkins and add maven Integration Plugin and create a maven project name as “Cicd-lab1” and add URL of our repo in github and configure it.**

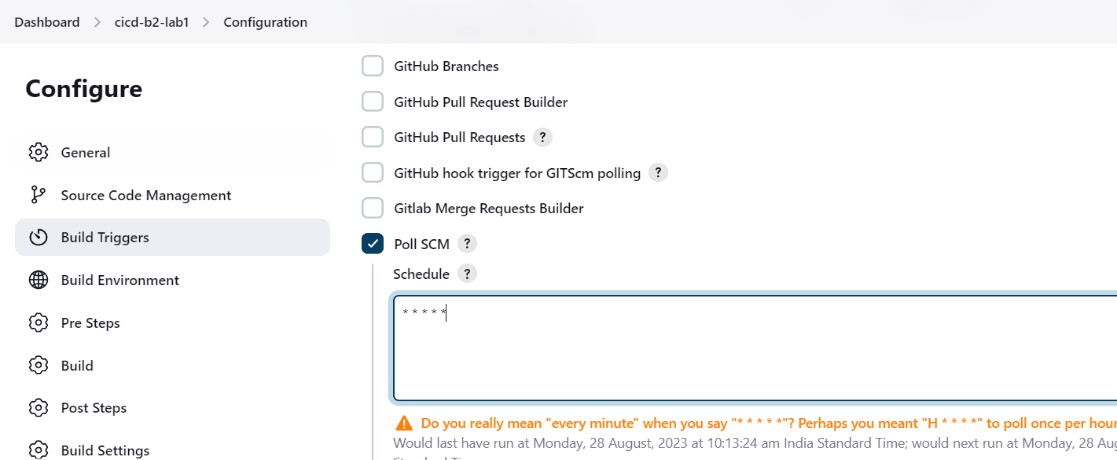


****

1. **Now Build it.**

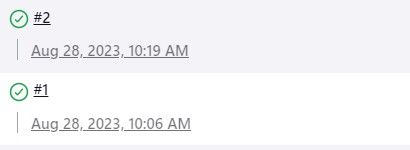


1. **edit the code and add poll scm in jenkins configuration then again commit and push it.**



1. **After push we can see it will make automatic build.**



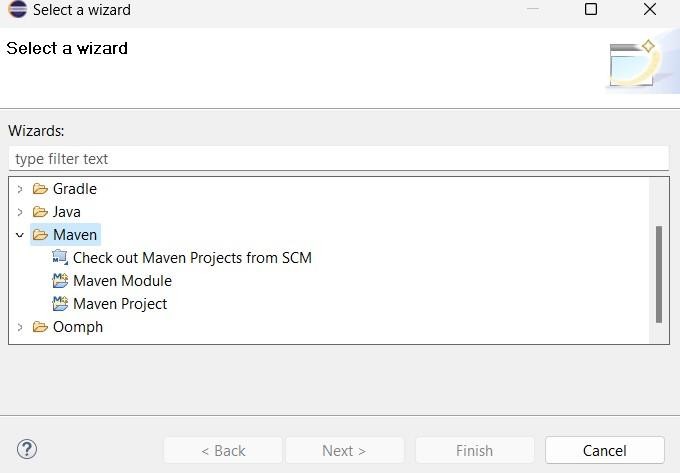


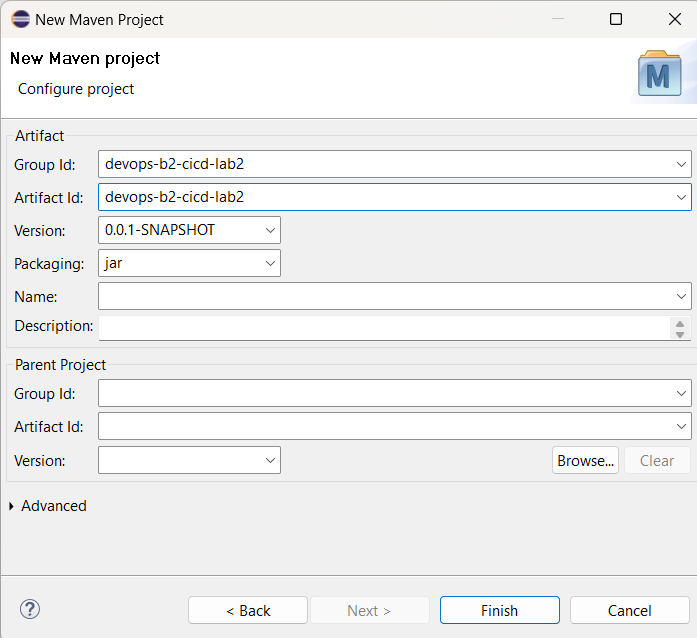
# EXPERIMENT – 2

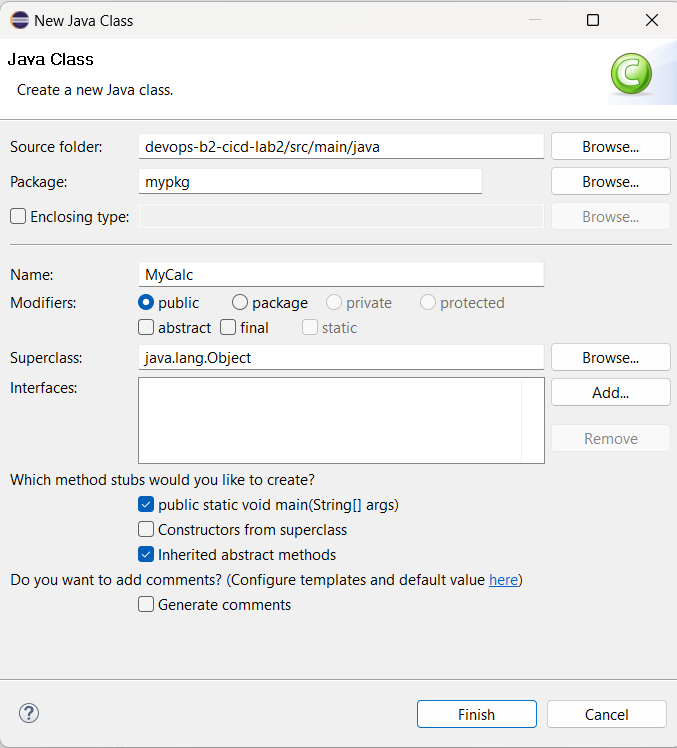
|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 50009031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

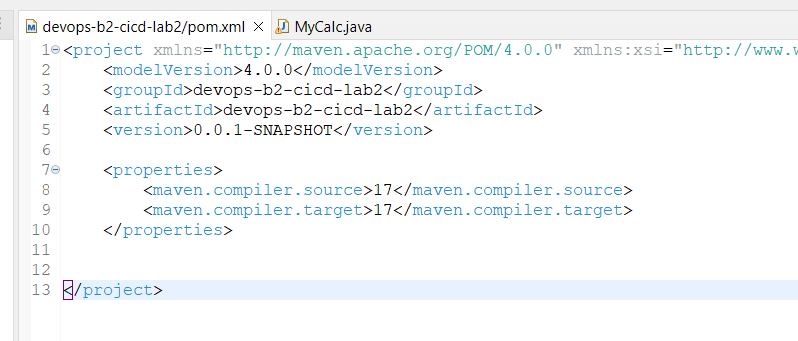
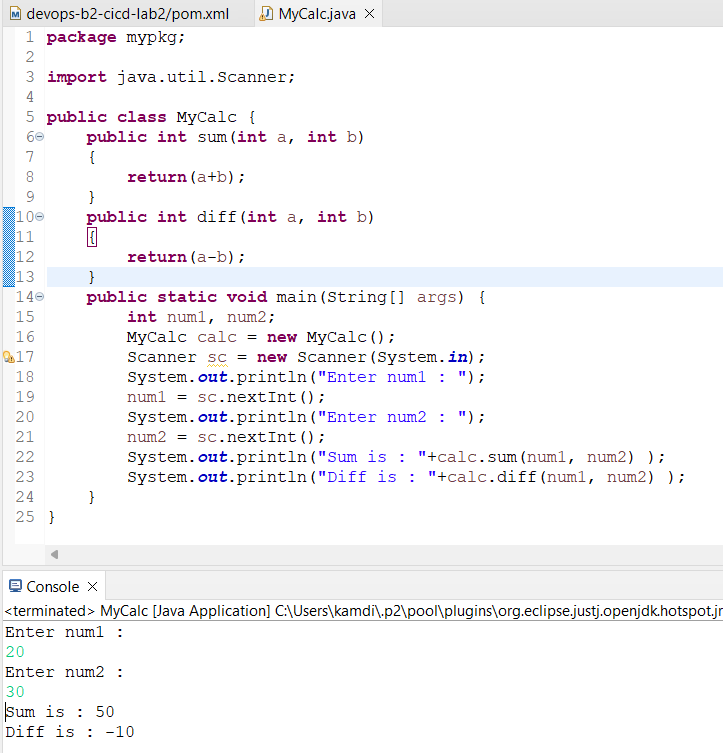
**Aim: Creating a Jenkins Pipeline with a Jenkins file.**

1. **Create a Maven Project in eclipse.**

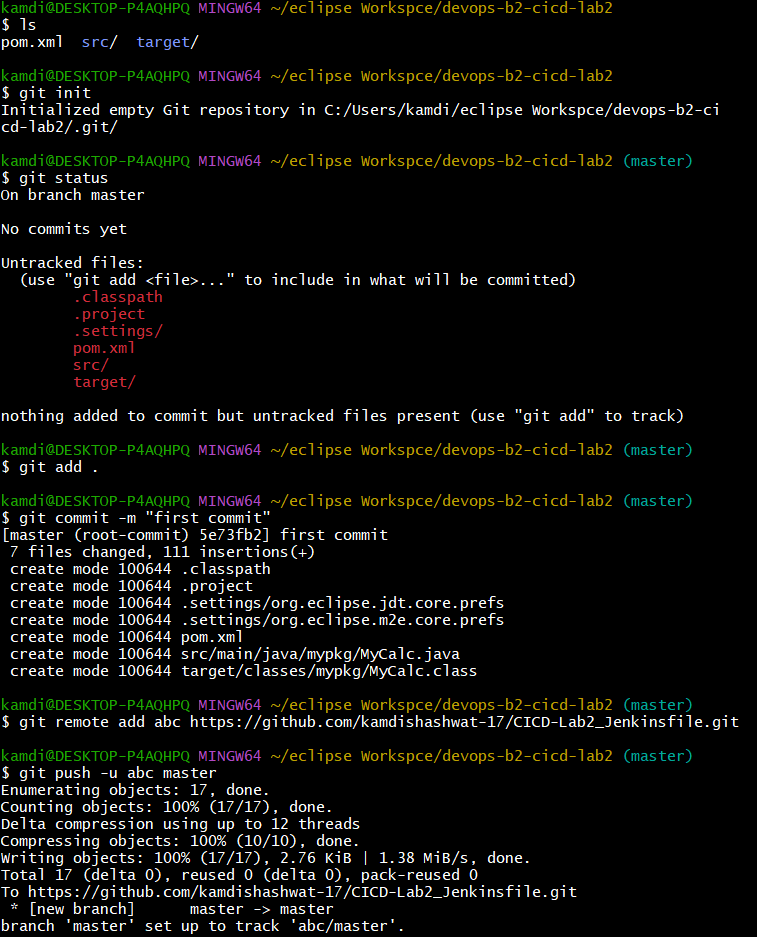
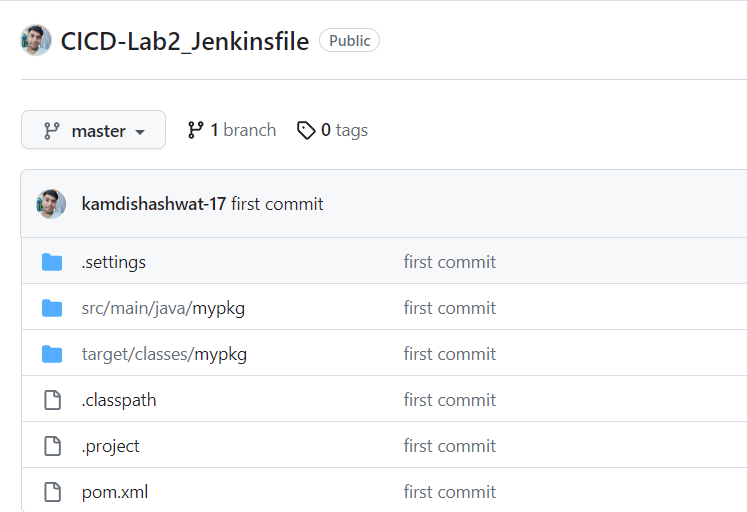




****



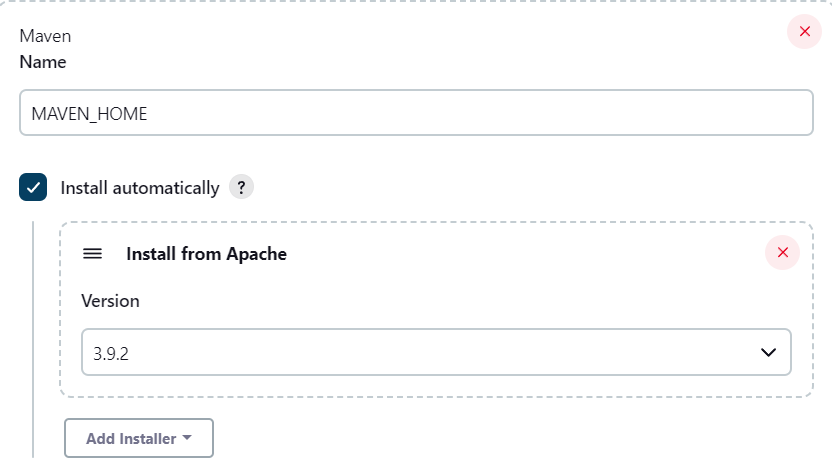
1. **Make a repo name as “CICD-Lab2\_Jenkinsfile” and Push project on github.**



1. **Add Maven integration and Git plugin in Jenkins , configure repo and make a build**

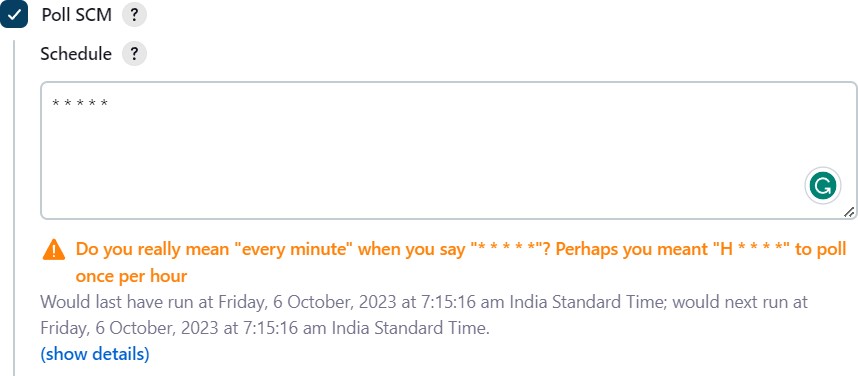


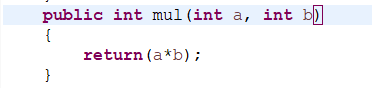
****

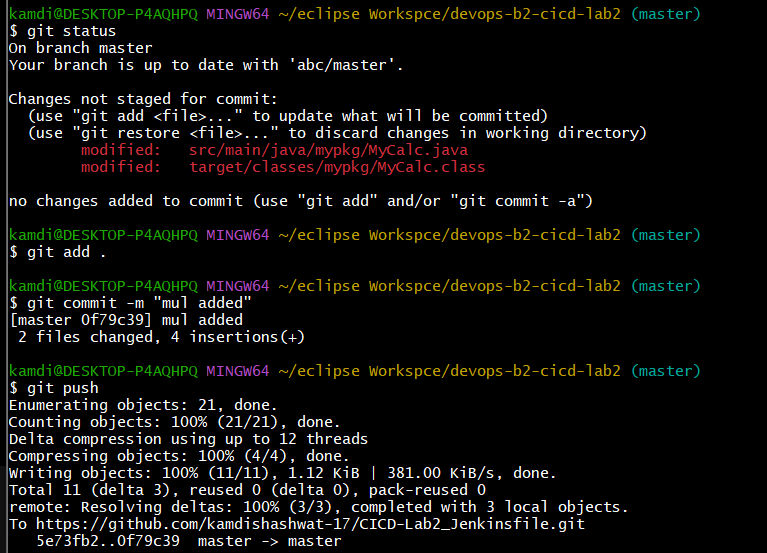
****

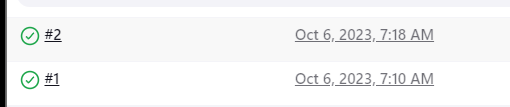


1. **edit the code ,add mul funcion and add poll scm in jenkins configuration then again commit and push it.**

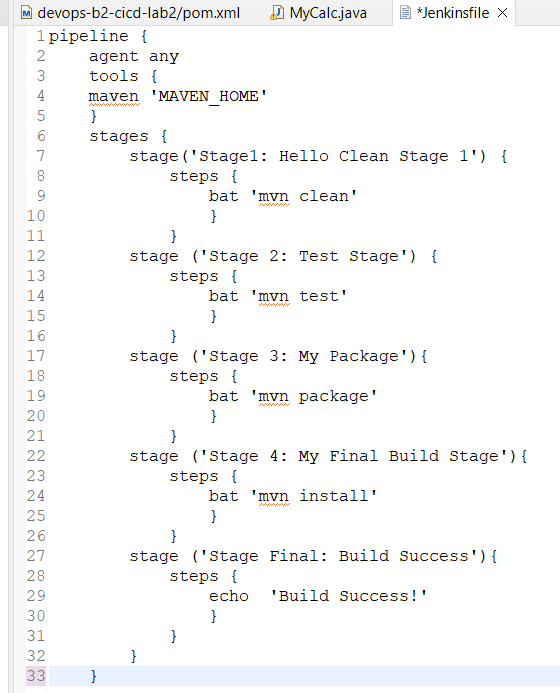


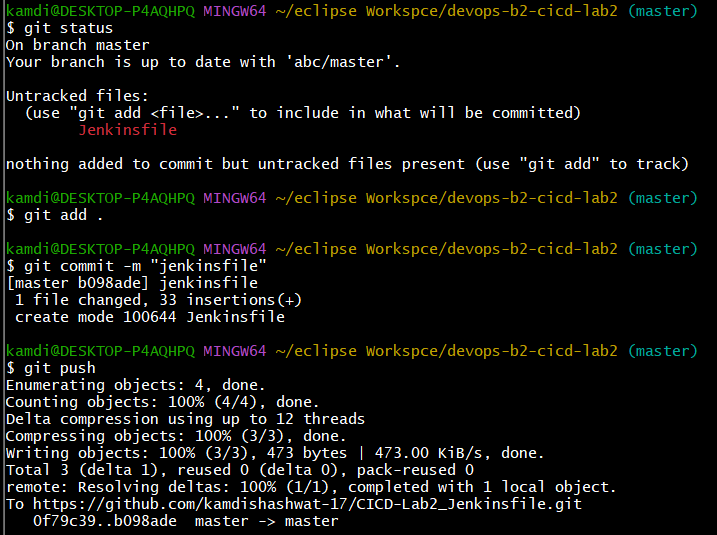


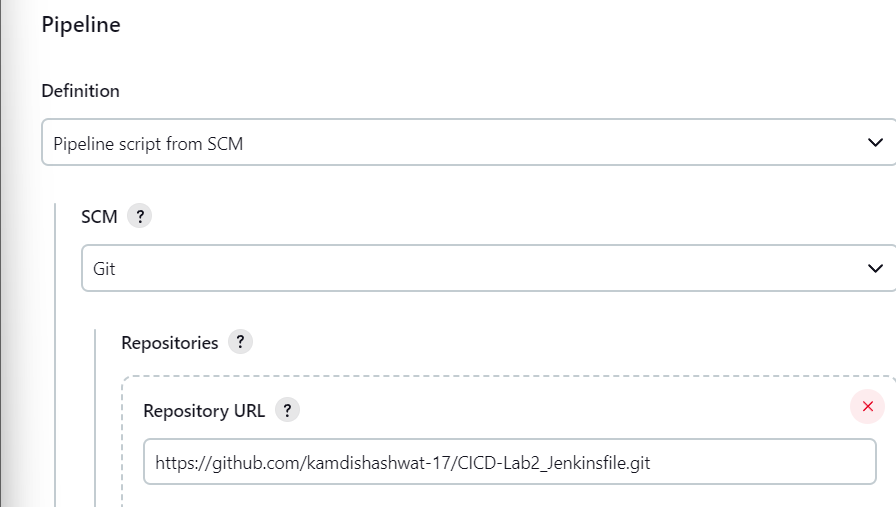
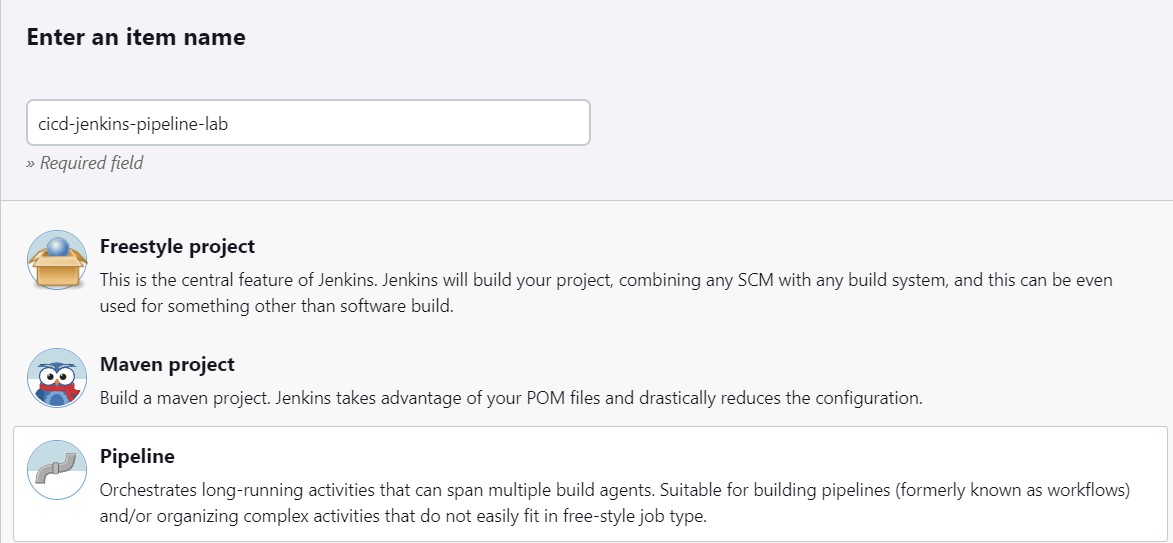
****

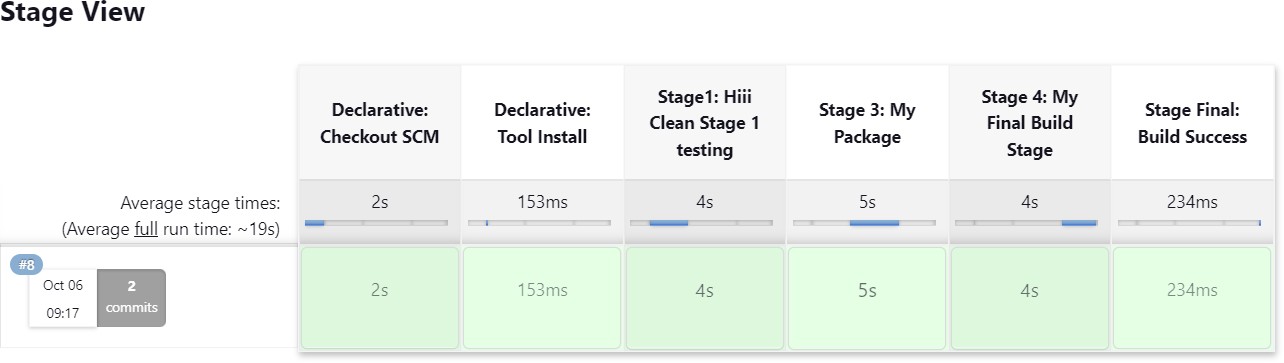


1. **Add Jenkins file then commit and push it , In Jenkins make pipeline add configure it. When you create a pipeline. The pipeline will start.**



****



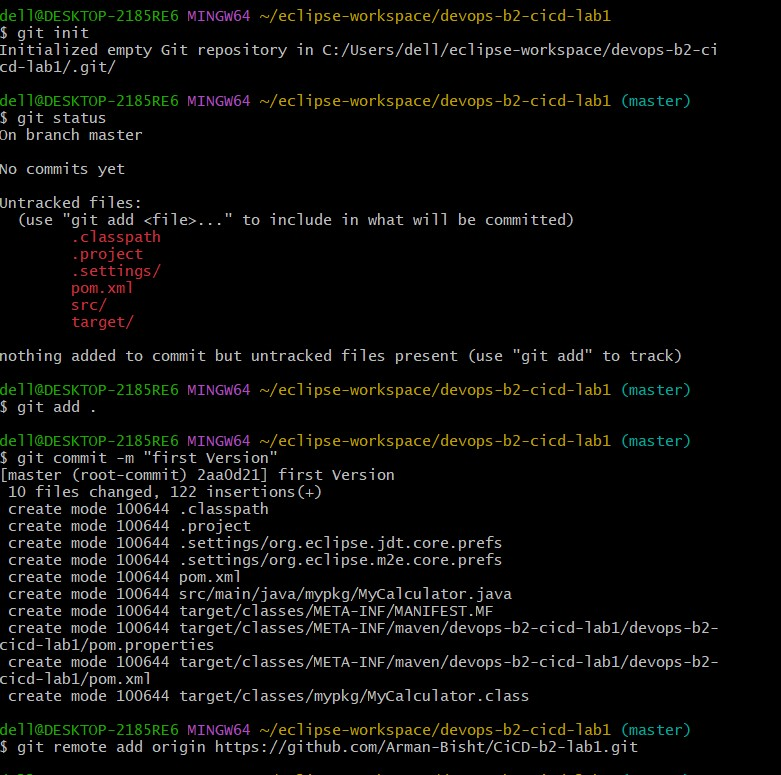


# EXPERIMENT – 3 and 4

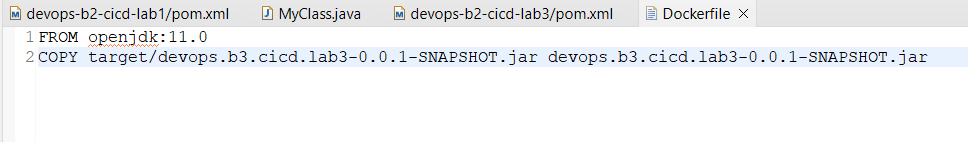
|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 500093031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

**Aim: Maven Build using GitHub Actions**

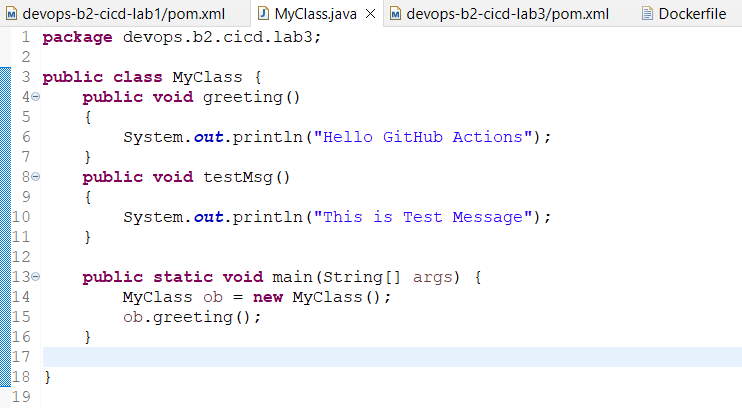
1. Push our file on git repo name as CICD-lab3 which contain class file,dockerfile,pom.xml file and so on.



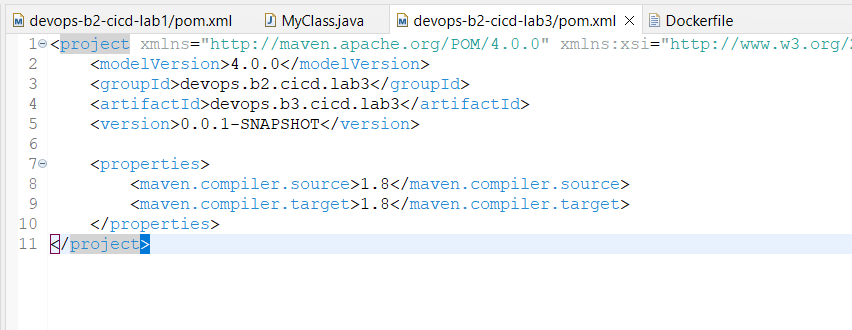
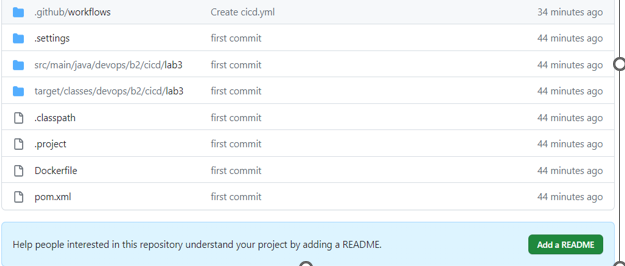
Docker file



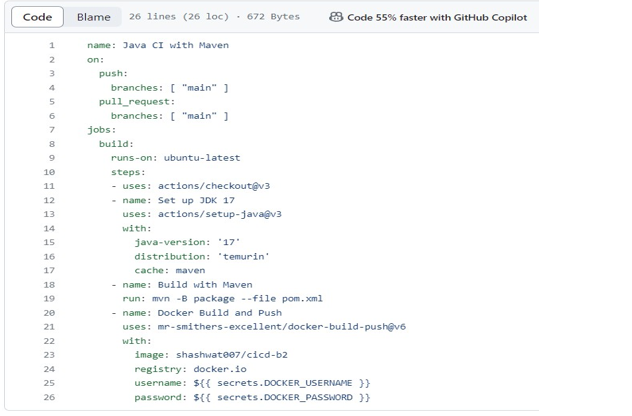
MyClass.java file



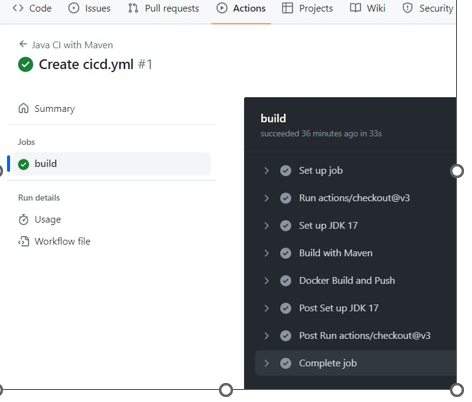
Pom.xml file

****

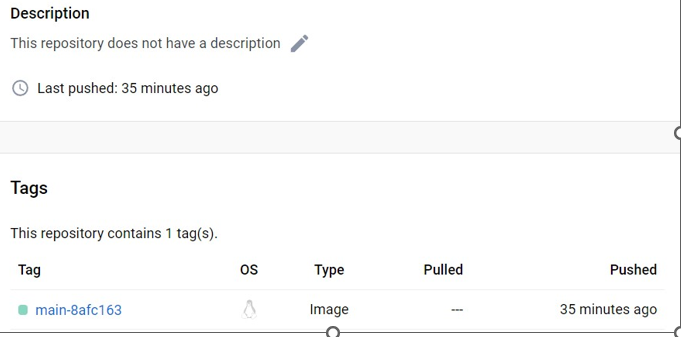
1. **Create a GitHub Actions Workflow name as cicd.yml and click on commit and push changes.**

****

1. **Check the Workflow Status in that you will notices that our build has been completed and build images has been visible in our docker hub .**



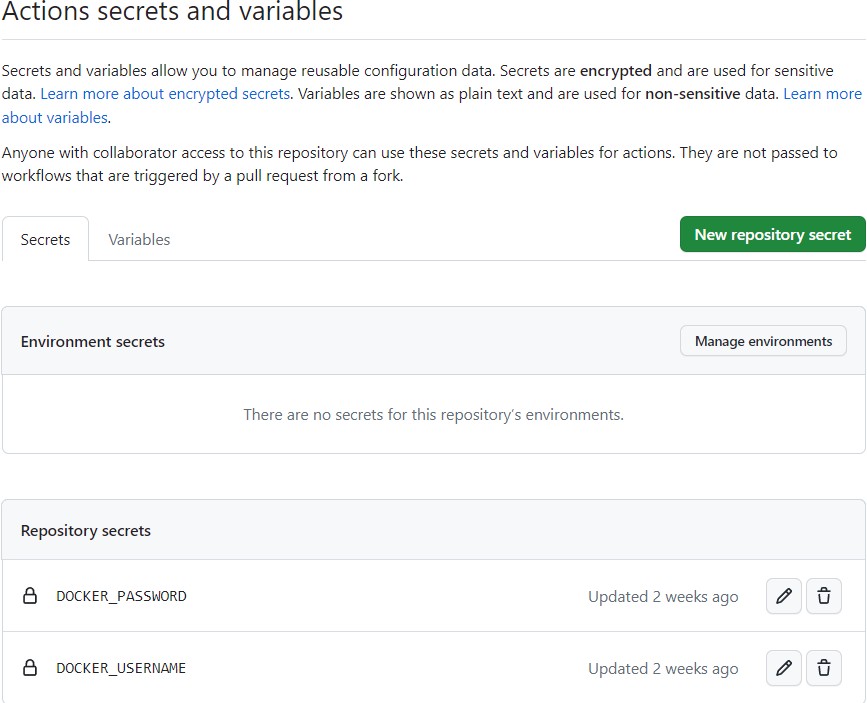
1. **Build image in docker hub**

****

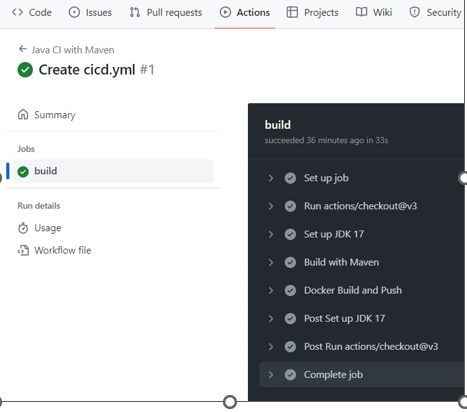
# EXPERIMENT – 4

**Aim : - Docker build and push using Github actions.**

1. **Add Docker Hub Credentials to GitHub Secrets**



1. **Commit and Push Changes**
2. **Check the Workflow Status in that you will notices that our build has been completed and build images has been visible in our docker hub .**

****

4] Build image in docker hub and docker command in a public view.



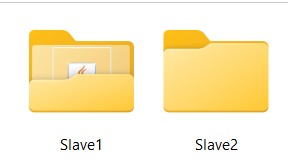
Docker Arman-Bisht cicd-b2:tagname

EXPERIMENT – 5

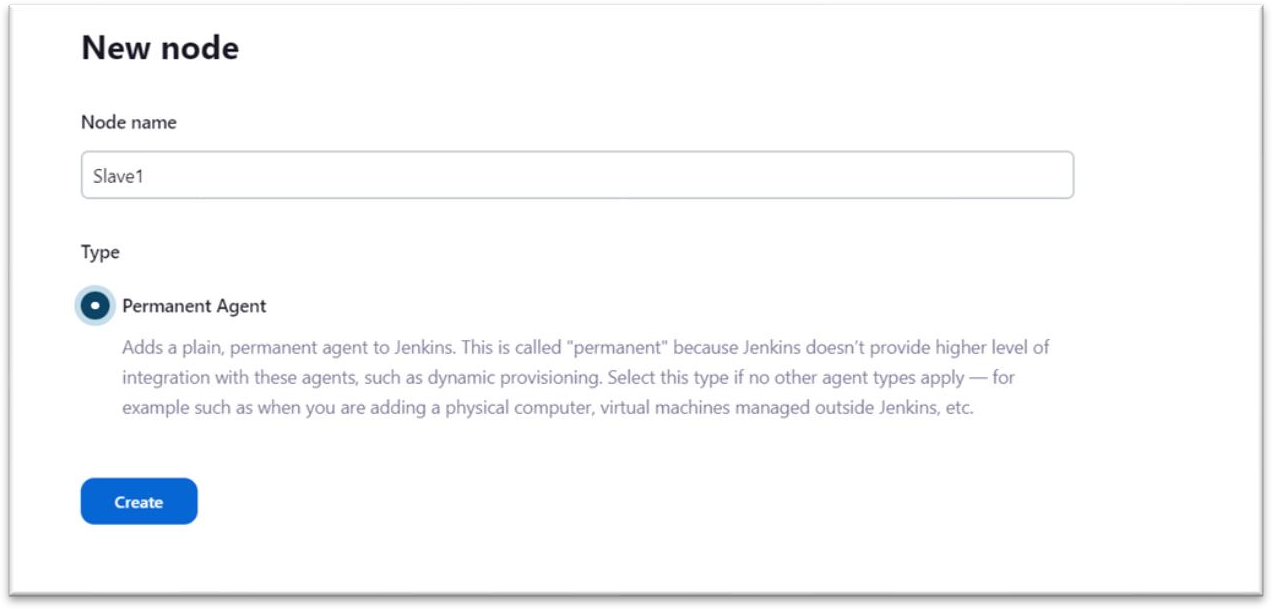
|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 500093031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

**Aim: Master/Slave concept in Jenkins**

1. Create Salve1 and Slave2 folder in your Local drive.

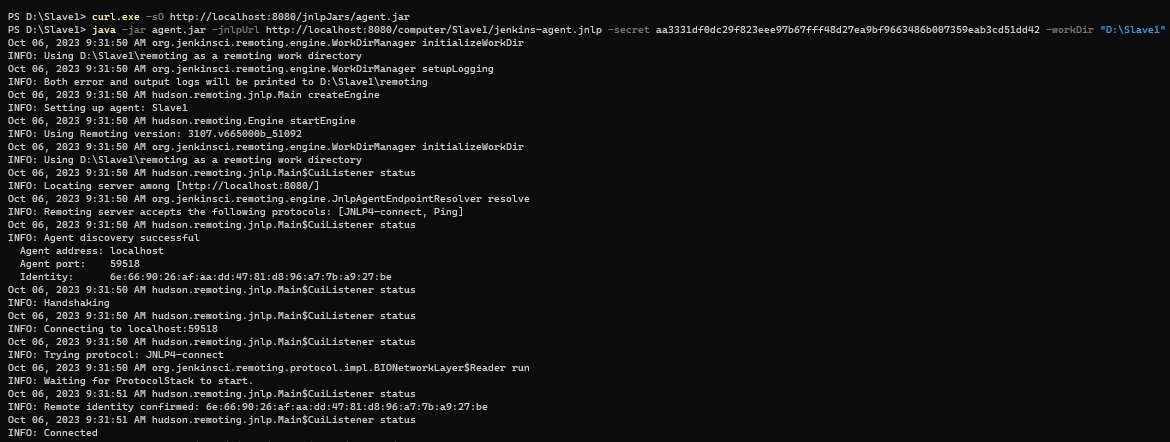


1. Create a New Node and assign node name as Slave1.

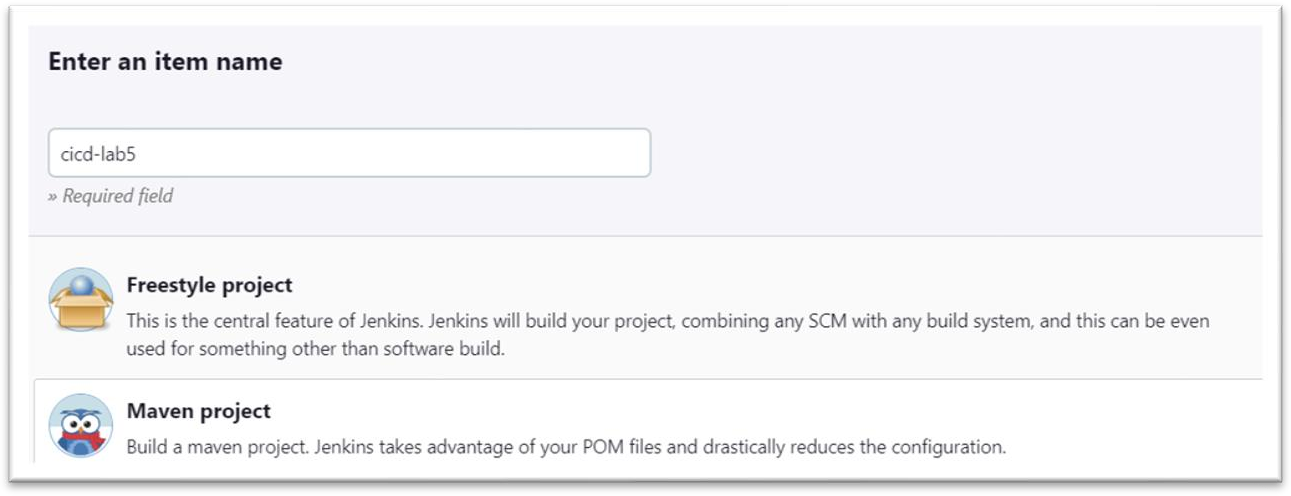


1. Run Agent slave1 command on our command line windows.

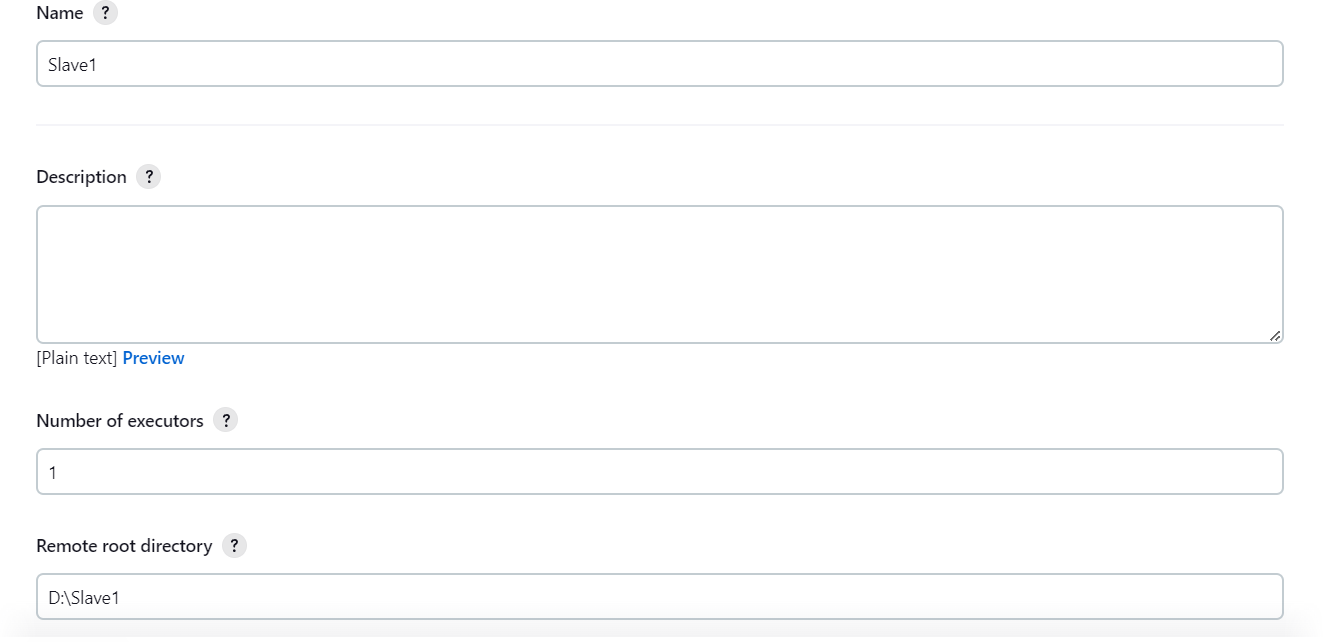
.

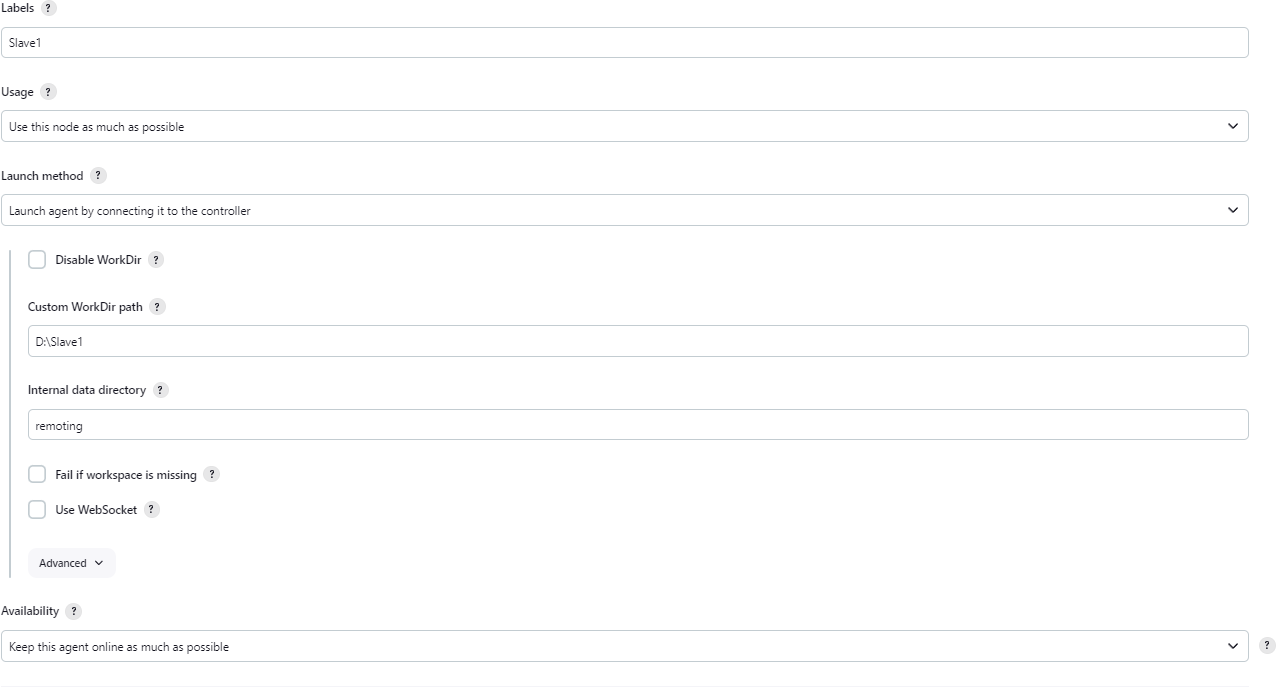


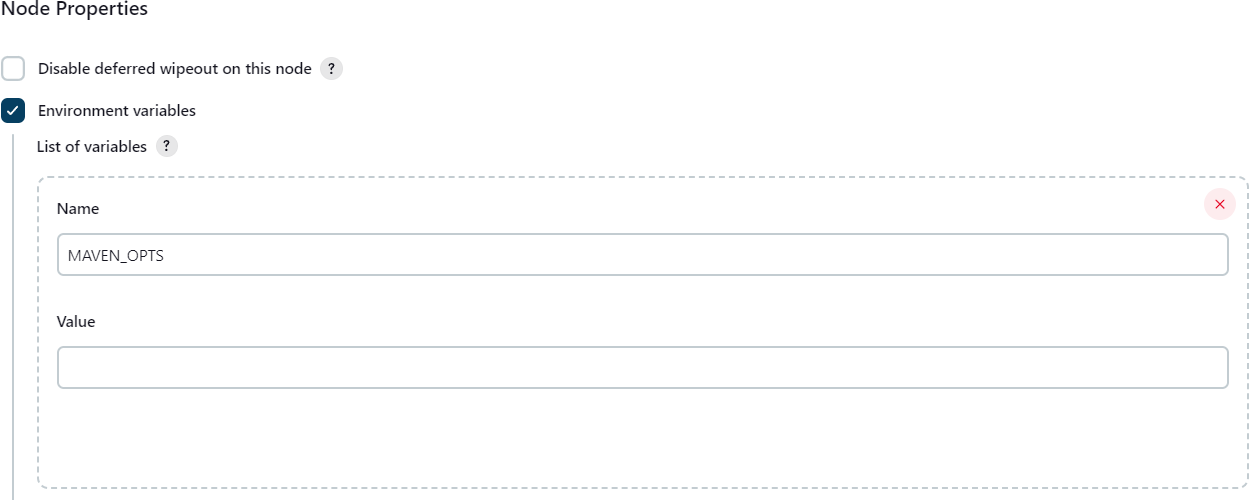
1. On your Jenkins create a new maven project name as cicd-lab5.



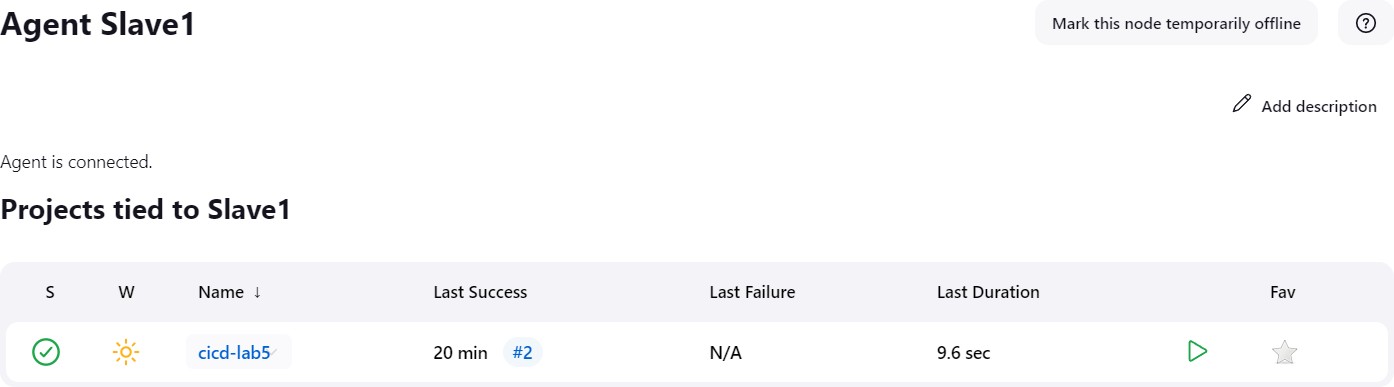
1. Configure it.

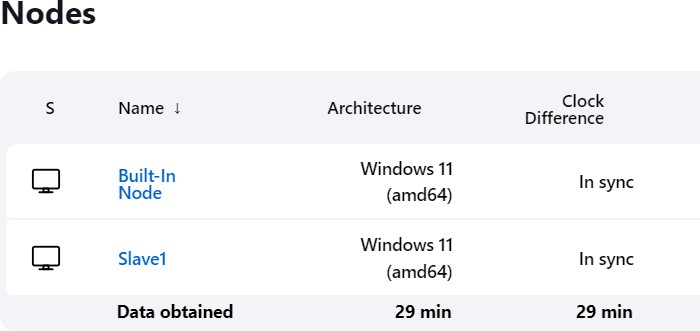






1. Make a build and check the output and Node.



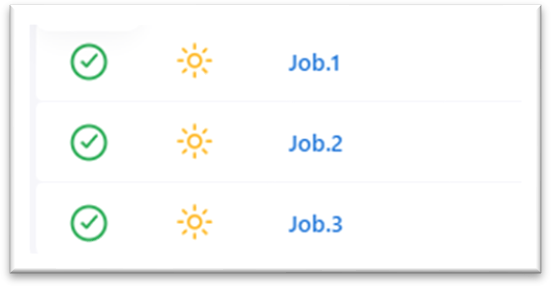


# EXPERIMENT – 6

|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 500093031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

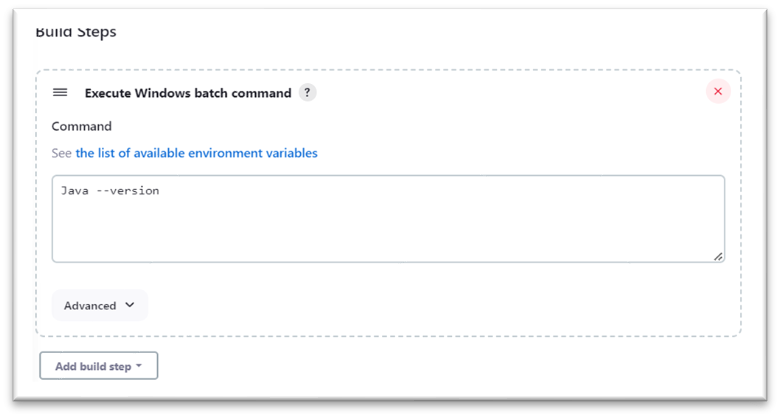
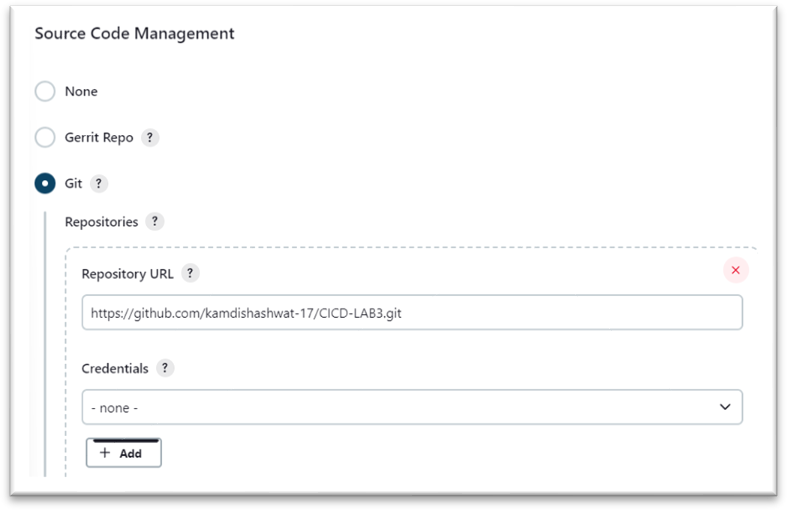
### Aim: Job Chaining in Jenkins

1. Creating Jobs, such as Job1, Job2, Job3 to stimulate job chaining and sequential execution.

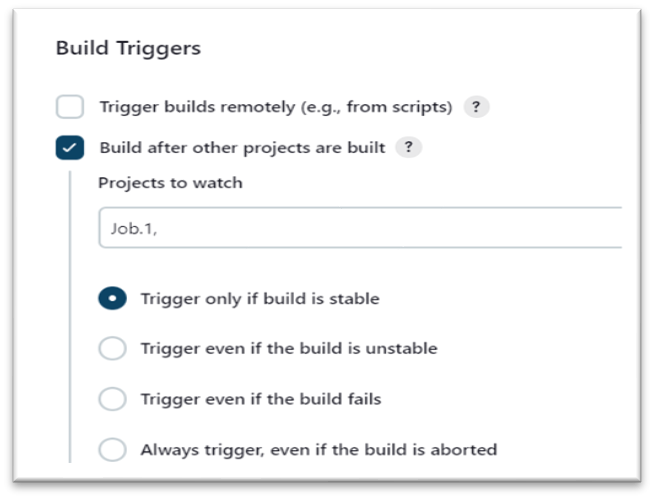
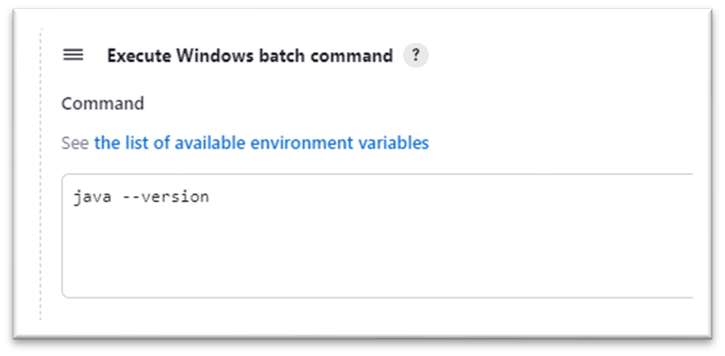
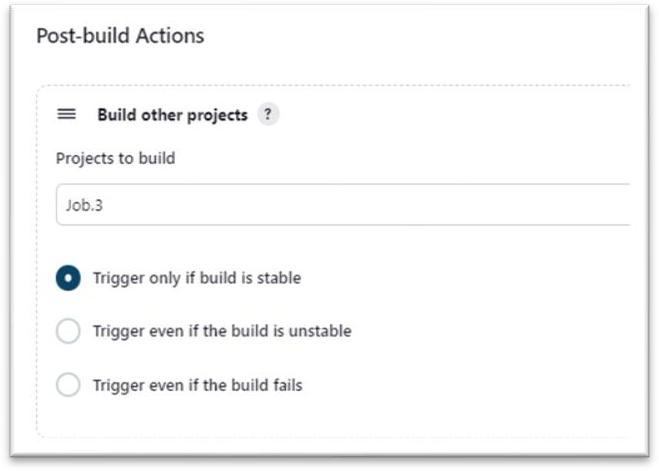


1. Configure Job Dependencies.

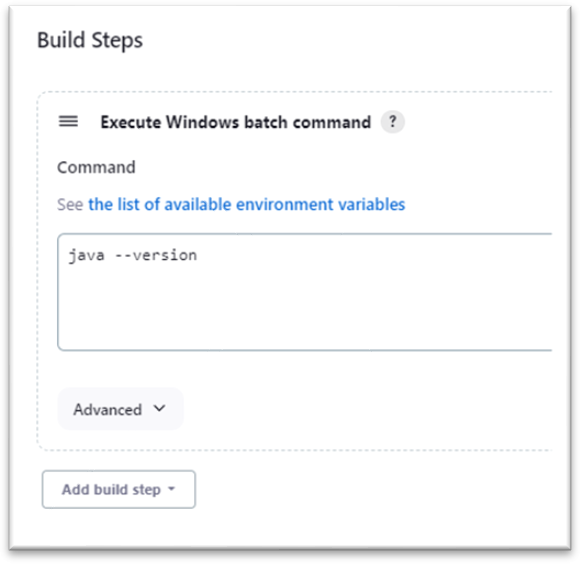
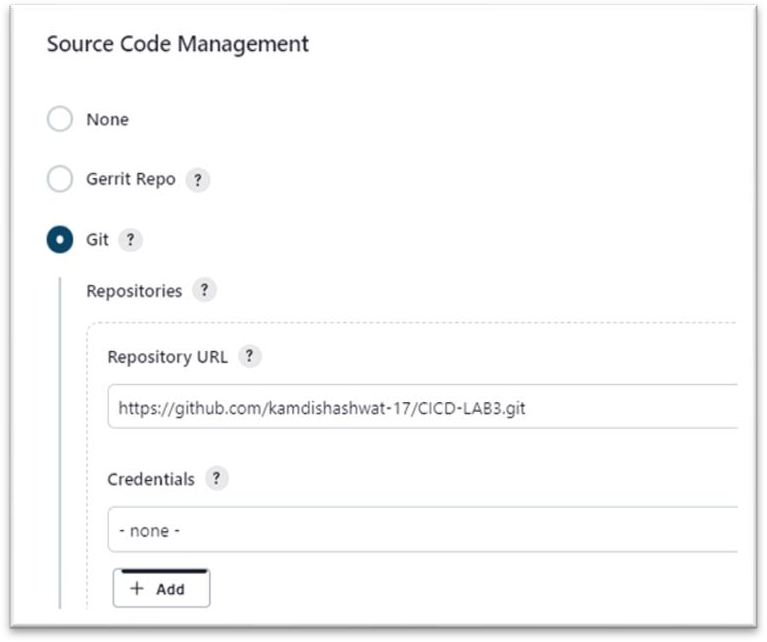
**Job.1 Configuration:**



**Job.2 Configuration:**

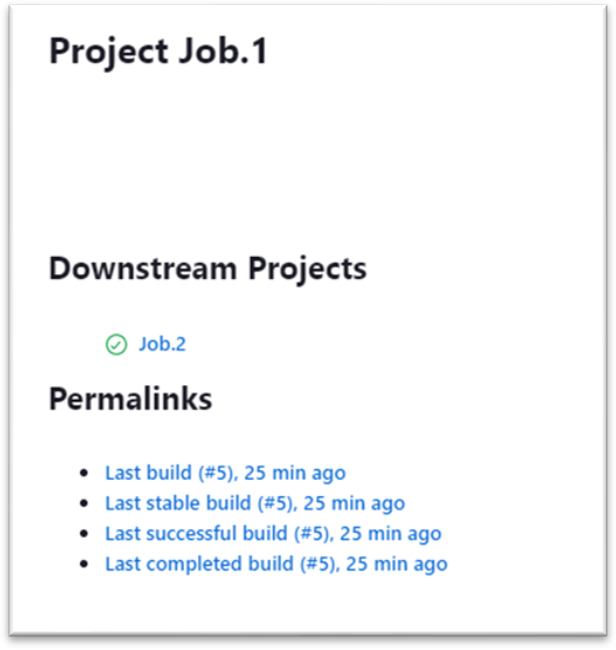


**Job.3 Configuration:**



1. Setting Up Build Triggers and make the Build.

**Job.1:**



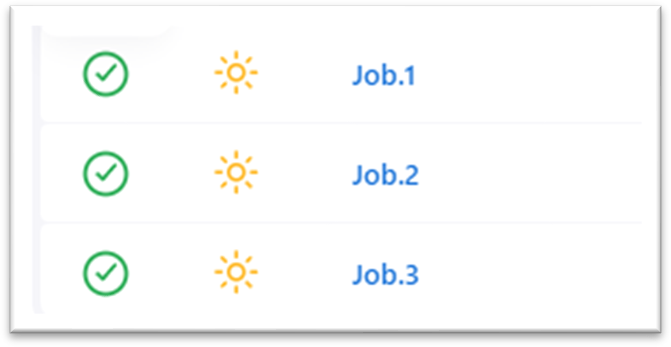
**Job.2 :**



**Job.3 :**



1. Viewing Console Output and Logs.

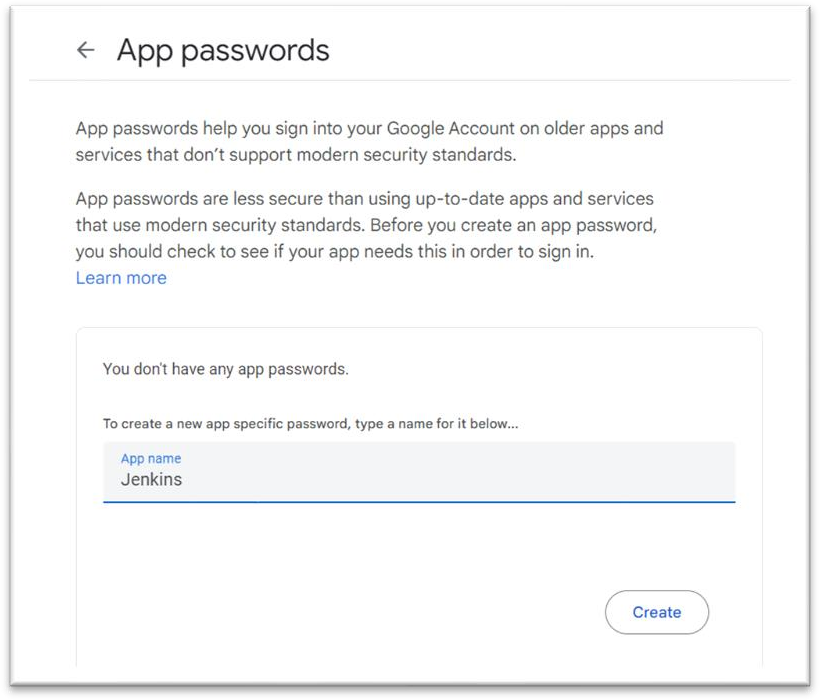


# EXPERIMENT – 7

|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 500093031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

### Aim: Email Notification in Jenkins

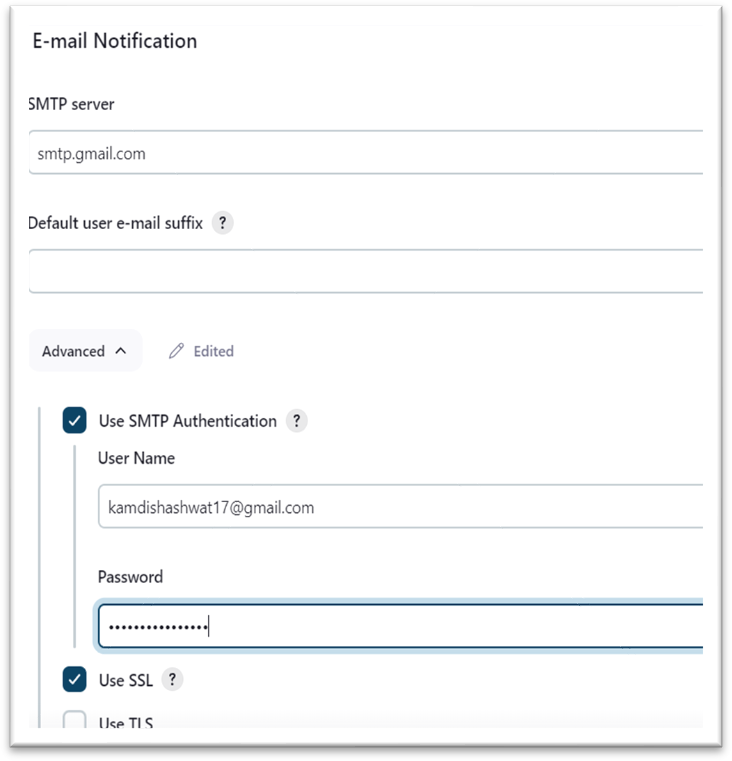
1. Create a “App Password” from our google account setting.



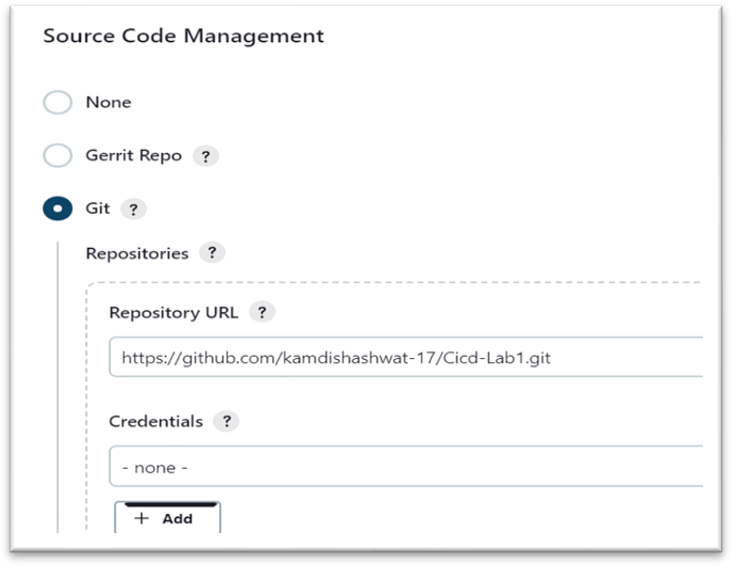
1. Installing Email Notification plugin.

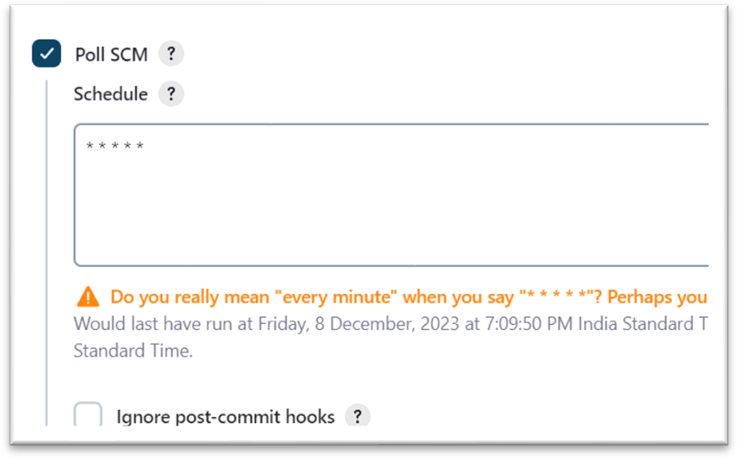


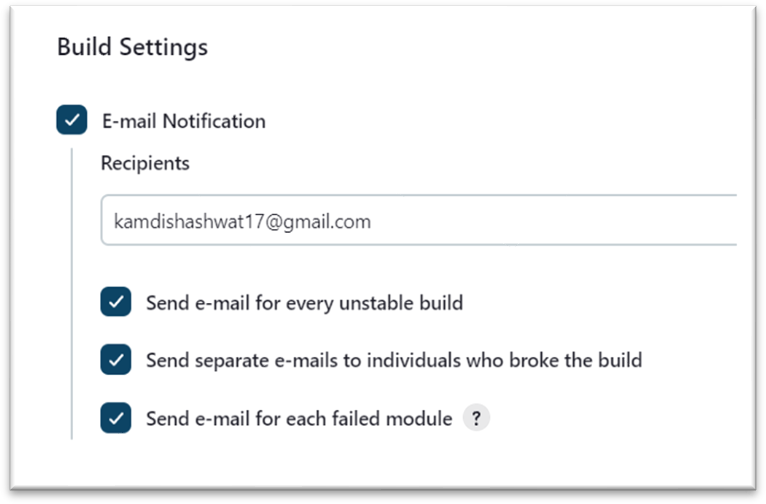
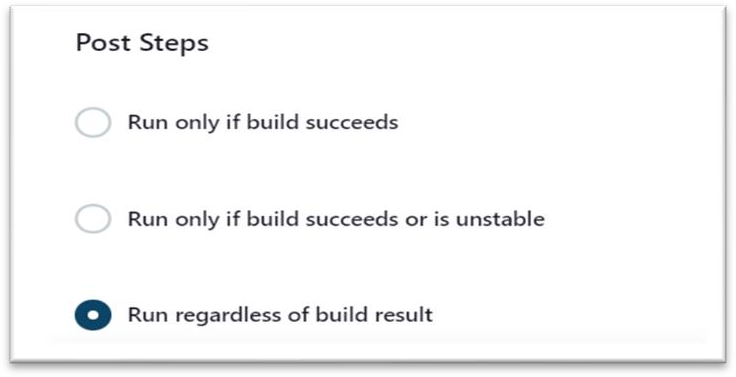
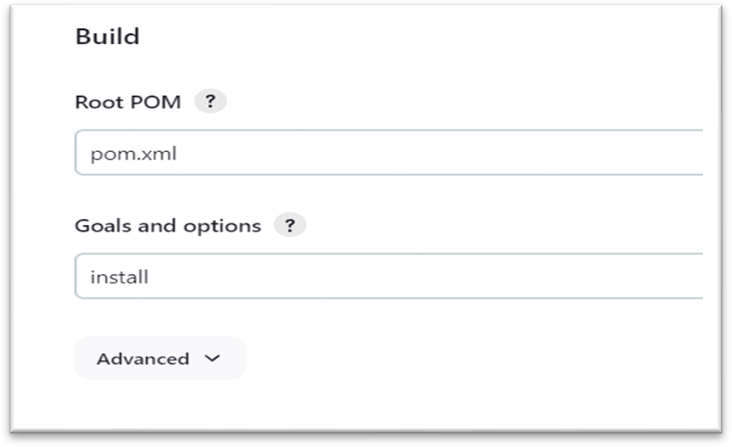
1. Configure SMTP Server by going to Dashboard > manage Jenkins > System > Email notification.



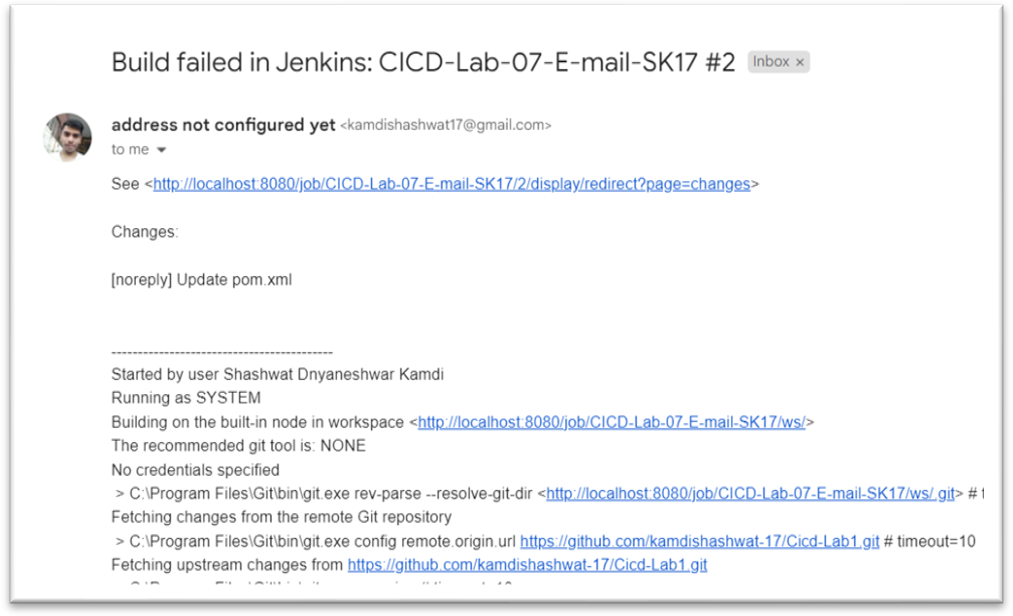
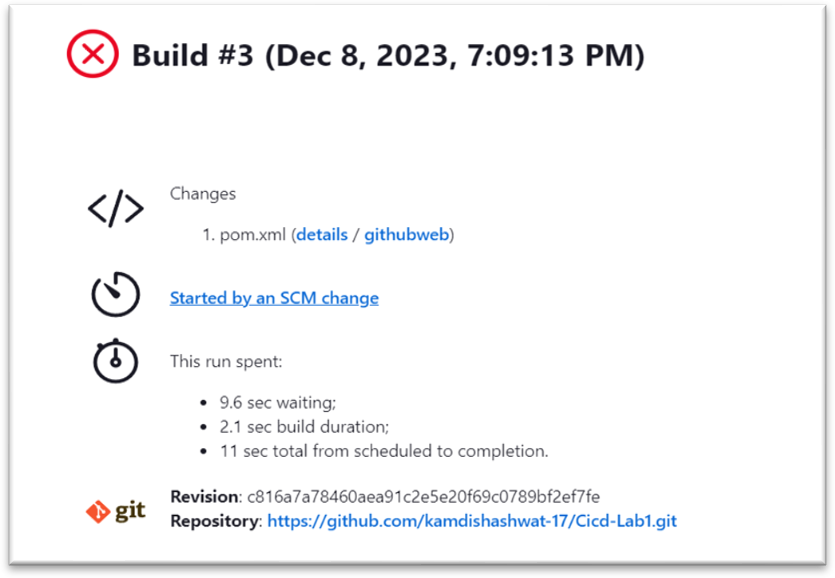
1. Create a maven project, configure it and make a build.







4] Check the Build and test the mail by intentionally failing a build.



# EXPERIMENT – 8

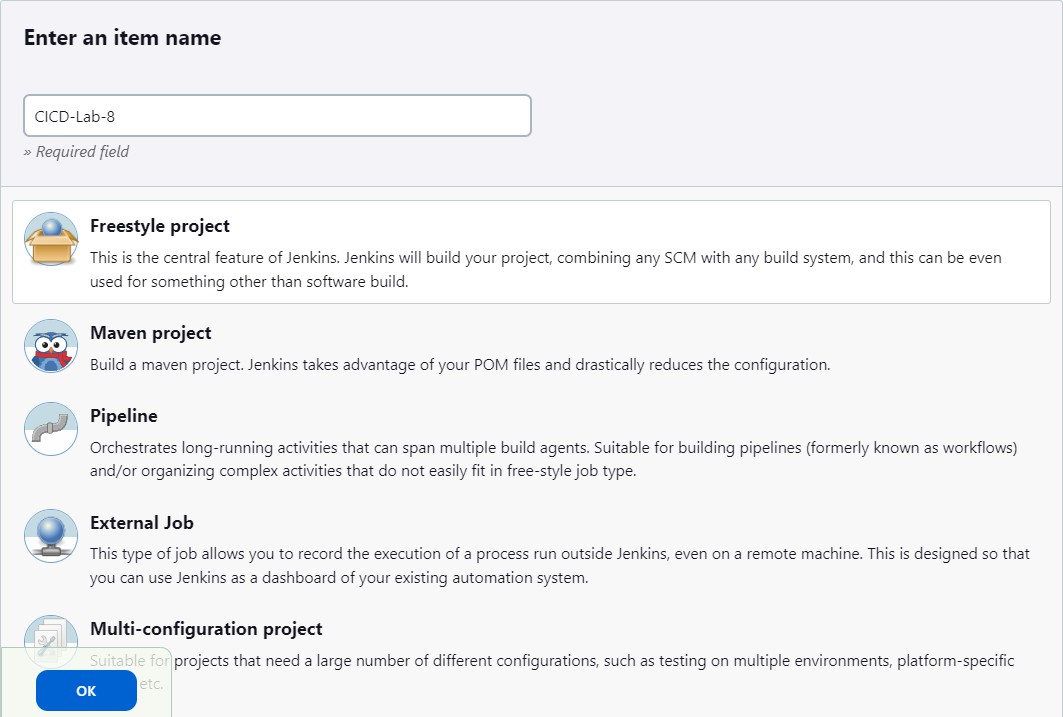
|  |
| --- |
| Name: - Arman Bisht |
| Batch – 2 [DevOps Non-Hons] |
| SAP ID- 500093031 |
| Subject – Continuous Integration and Continuous Delivery Lab |

### Aim: Remote Triggering of a Parameterized Build

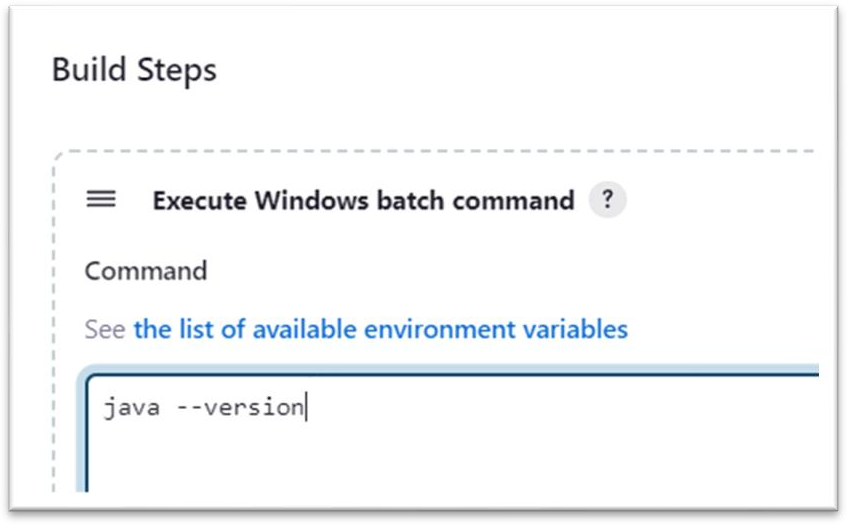
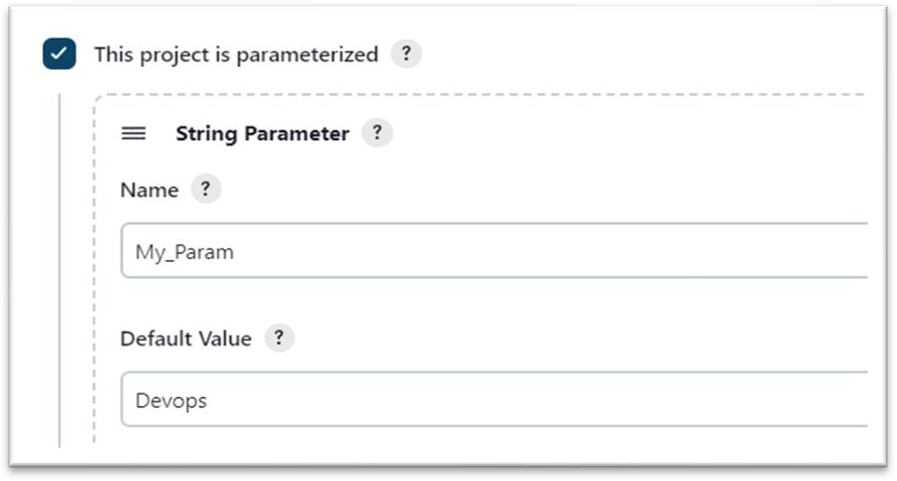
1. Open manage Jenkins and install “Build with parameters” Plugin.

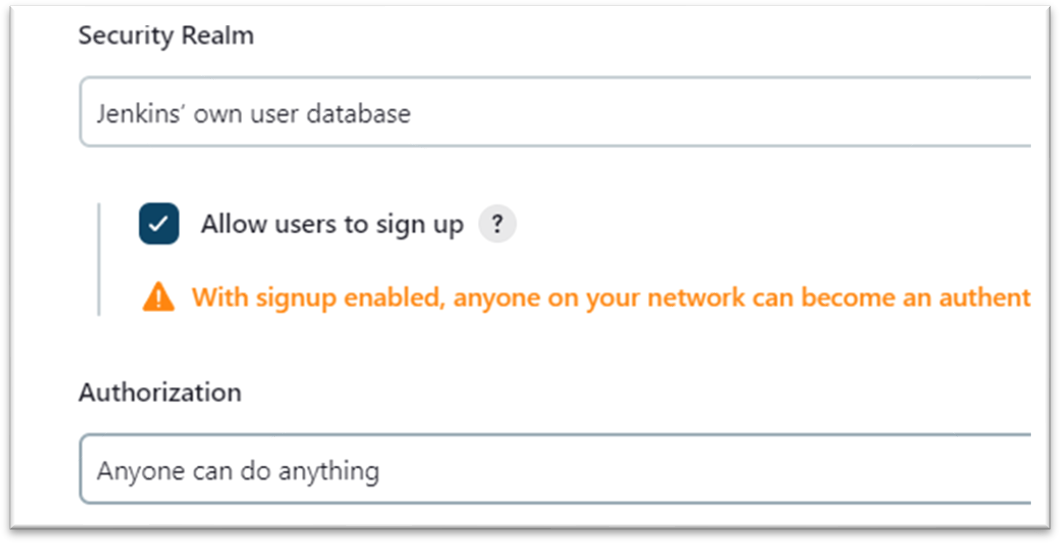


1. Select the **Freestyle project** as the build job type.

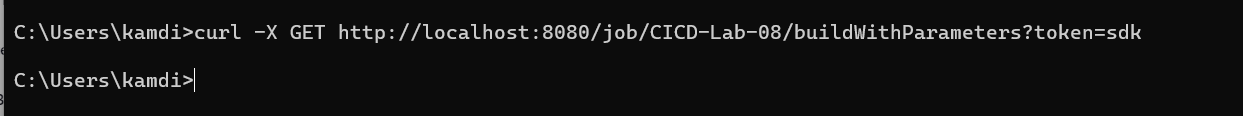


1. Configure as per Remote Triggering of a Parameterized Build.
2. Configure Global Security.





1. “curl -X GET <YourJenkinsJobUrl>/buildWithParameters?token=Sam”.Configure as per your URL and run it on Command Line.



1. Check the Console Output.

