Lab Experiment 2: Creating a Jenkins Pipeline with a Jenkinsfile

Objective: Create a Jenkins pipeline using a Jenkinsfile that builds a simple project, runs tests, and deploys the project to a designated environment.

Prerequisites:

1. Jenkins server up and running.
2. A sample project hosted in a version control repository (e.g., Git).

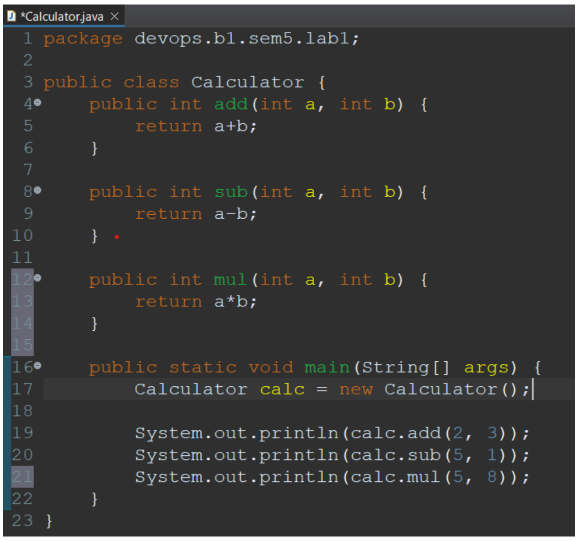
Steps:

Jenkins Configuration:

- Ensure that Jenkins is installed and accessible.
- Install necessary plugins: Pipeline and any plugins specific to your version control system (e.g., Git Plugin).

Setting Up the Project:

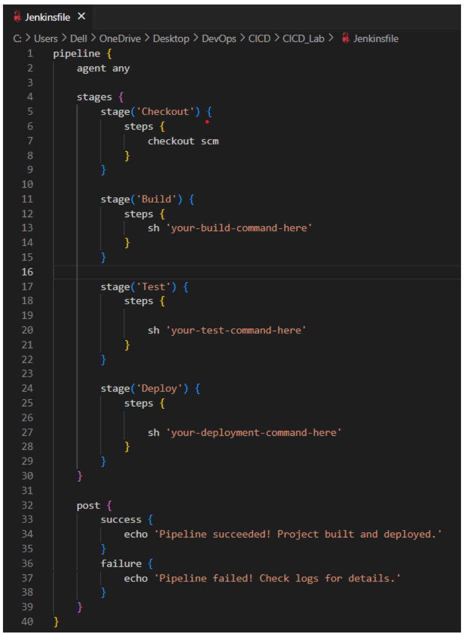
- Create a sample project (e.g., a simple web application) and host it on a version control repository (e.g., GitHub).



```java
package devops.b1.sem5.lab1;

public class Calculator {
    public int add(int a, int b) {
        return a+b;
    }

    public int sub(int a, int b) {
        return a-b;
    }

    public int mul(int a, int b) {
        return a*b;
    }

    public static void main(String[] args) {
        Calculator calc = new Calculator();

        System.out.println(calc.add(2, 3));
        System.out.println(calc.sub(5, 1));
        System.out.println(calc.mul(5, 8));
    }
}
```

Creating a Jenkinsfile:

In the root of your project repository, create a file named Jenkinsfile.



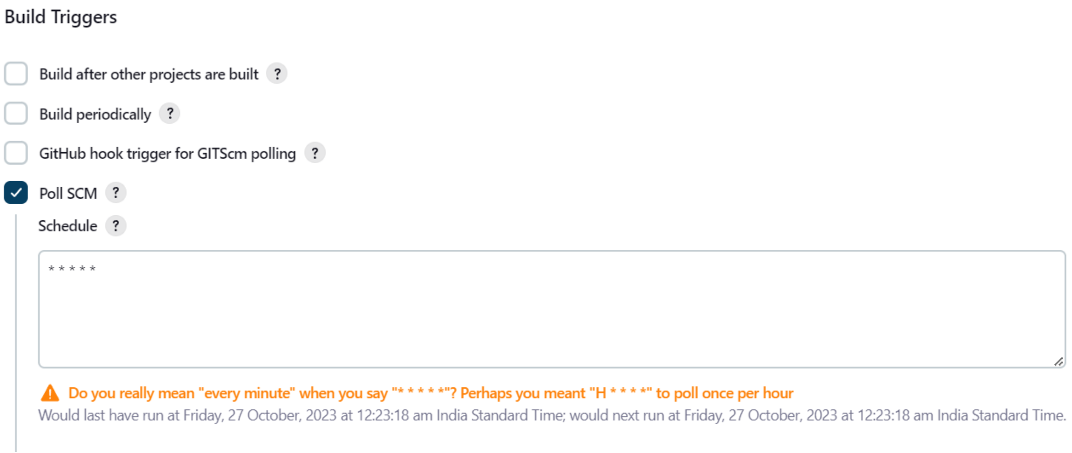
Jenkinsfile ✕

C: > Users > Dell > OneDrive > Desktop > DevOps > CICD > CICD_Lab > Jenkinsfile

```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Build') {
            steps {
                sh 'your-build-command-here'
            }
        }

        stage('Test') {
            steps {

                sh 'your-test-command-here'
            }
        }

        stage('Deploy') {
            steps {

                sh 'your-deployment-command-here'
            }
        }
    }

    post {
        success {
            echo 'Pipeline succeeded! Project built and deployed.'
        }
        failure {
            echo 'Pipeline failed! Check logs for details.'
        }
    }
}
```

Configuring the Pipeline in Jenkins:

- In Jenkins, create a new pipeline job.
- Link the job to your version control repository (e.g., provide the repository URL).
- Choose the option to use a Jenkinsfile from the repository and specify the path to your Jenkinsfile (usually the root directory).



Build Triggers

□ Build after other projects are built ?

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

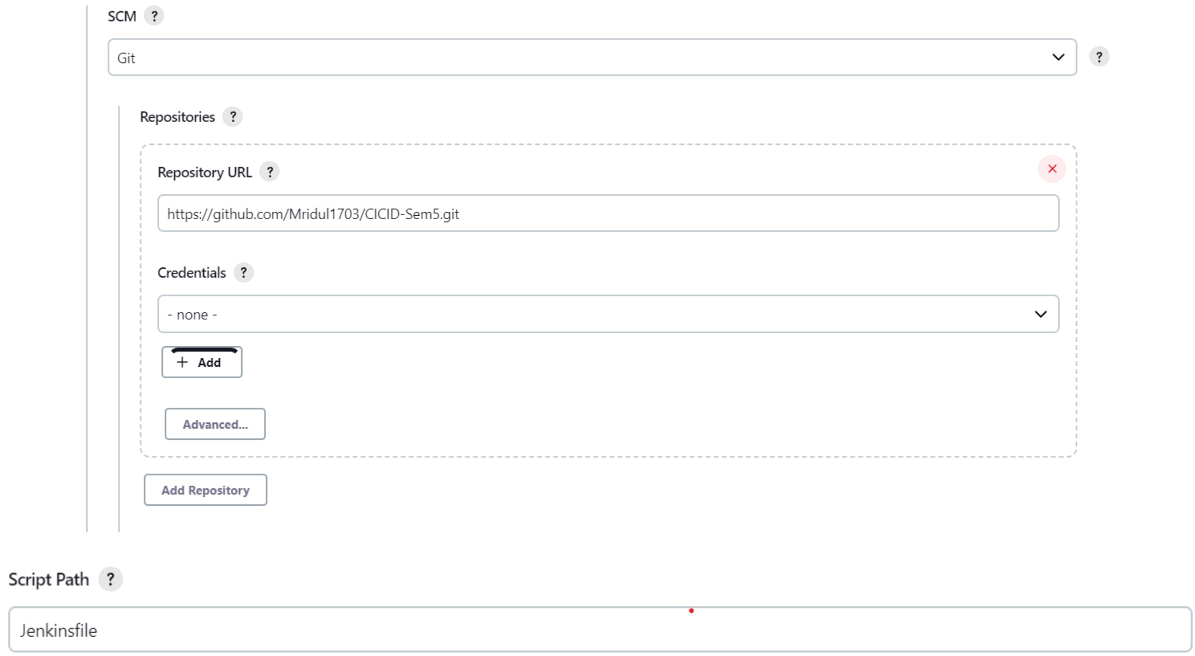
☑ Poll SCM ?

Schedule ?

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Friday, 27 October, 2023 at 12:23:18 am India Standard Time; would next run at Friday, 27 October, 2023 at 12:23:18 am India Standard Time.



SCM: Git

Repositories

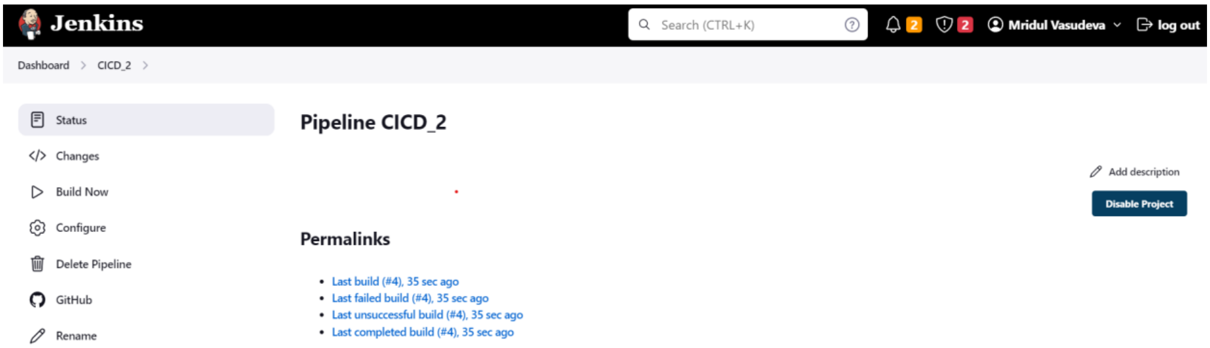
Repository URL: https://github.com/Mridul1703/CICID-Sem5.git

Credentials: - none -

Script Path: Jenkinsfile

Running the Pipeline:

- Trigger the pipeline manually or set up a webhook to trigger it automatically on repository changes.



Pipeline CICD_2

Permalinks

- Last build (#4), 35 sec ago
- Last failed build (#4), 35 sec ago
- Last unsuccessful build (#4), 35 sec ago
- Last completed build (#4), 35 sec ago