SIMPLE CALCULATOR PROJECT USING DEVOPS PRINCIPLES

NAME: Shivankar Piligundla
Roll NO: IMT2020016

Github: https://github.com/shivankar-p/Calculator

INTRODUCTION:

Create a simple calculator program to demonstrate devops principles using INTELLIJ IDEA, Java, Maven as the build tool, and Git as the Version control system. We will be using Jenkins for CI/CD, Ansible & ngrok for IAC and deployment

Creating the Project:

Creating new project in Intellij using maven as the build tool. Then we are going to run the following commands:

mvn clean
mvn compile
mvn install

```
~/IdeaProjects/Calculator master > mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< org.example:Calculator >----------------------
[INFO] Building Calculator 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ Calculator ---
[INFO] Deleting /Users/shivankar/IdeaProjects/Calculator/target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.140 s
[INFO] Finished at: 2023-10-30T20:56:49+05:30
[INFO] ------------------------------------------------------------------------
```

```
~/IdeaProjects/Calculator master > mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----------------------< org.example:Calculator >-----------------------
[INFO] Building Calculator 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ Calculator ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ Calculator ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 1 source file with javac [debug target 11] to target/classes
[WARNING] system modules path not set in conjunction with -source 11
[INFO] Annotation processing is enabled because one or more processors were found
  on the class path. A future release of javac may disable annotation processing
  unless at least one processor is specified by name (-processor), or a search
  path is specified (--processor-path, --processor-module-path), or annotation
  processing is enabled explicitly (-proc:only, -proc:full).
  Use -Xlint:-options to suppress this message.
  Use -proc:none to disable annotation processing.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.610 s
[INFO] Finished at: 2023-10-30T20:57:28+05:30
[INFO] ------------------------------------------------------------------------
~/IdeaProjects/Calculator master >
```

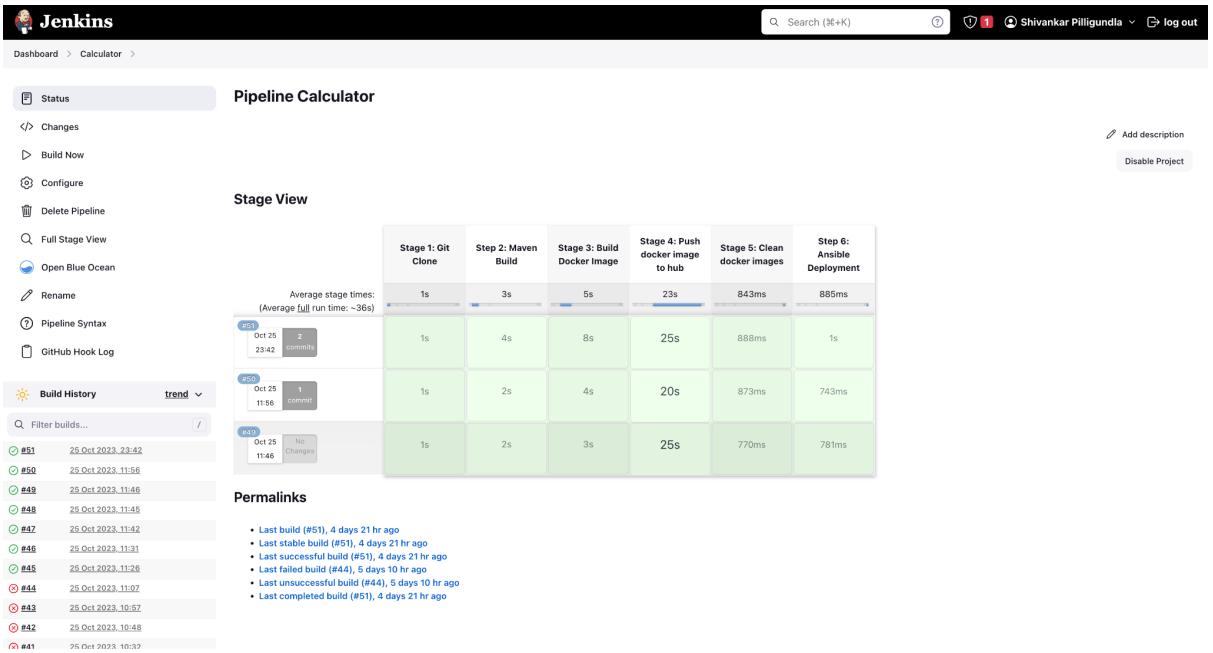
```
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ Calculator ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running CalculatorTest
ERROR StatusConsoleListener Unrecognized format specifier []
ERROR StatusConsoleListener Empty conversion specifier starting at position 25 in conversion pattern.
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.283 s -- in CalculatorTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ Calculator ---
[INFO] Building jar: /Users/shivankar/IdeaProjects/Calculator/target/Calculator-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- assembly:3.3.0:single (default) @ Calculator ---
[INFO] Building jar: /Users/shivankar/IdeaProjects/Calculator/target/Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ Calculator ---
[INFO] Installing /Users/shivankar/IdeaProjects/Calculator/pom.xml to /Users/shivankar/.m2/repository/org/example/Calculator/1.0-SNAPSHOT/Calculator-1.0-SNAPSHOT.pom
[INFO] Installing /Users/shivankar/IdeaProjects/Calculator/target/Calculator-1.0-SNAPSHOT.jar to /Users/shivankar/.m2/repository/org/example/Calculator/1.0-SNAPSHOT/Calculator-1.0-SNAPSHOT.jar
[INFO] Installing /Users/shivankar/IdeaProjects/Calculator/target/Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar to /Users/shivankar/.m2/repository/org/example/Calculator/1.0-SNAPSHOT/Calculator-1.0-S
NAPSHOT-jar-with-dependencies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.159 s
[INFO] Finished at: 2023-10-30T20:58:22+05:30
[INFO] ------------------------------------------------------------------------
~/IdeaProjects/Calculator master >                                                3s
```

Then Git version control is used and a remote repo is created in github. To build a CI/CD pipeline Jenkins is used.

We will create a Jenkins pipeline with the following script:

```
pipeline{
    environment{
        docker_image = ""
    }
    agent any
    stages{
        stage('Stage 1: Git Clone'){
            steps{
                git branch: 'master',
                url:'https://github.com/shivankar-p/Calculator.git'
            }
        }
        stage('Step 2: Maven Build'){
            steps{
                sh 'mvn clean install'
            }
        }
        stage('Stage 3: Build Docker Image'){
            steps{
                script{
                    docker_image = docker.build "shivankarp/calculator:latest"
                }
            }
        }
        stage('Stage 4: Push docker image to hub') {
            steps{
                script{
                    docker.withRegistry('https://index.docker.io/v1/',
"DockerhubCred"){
                    docker_image.push()
```

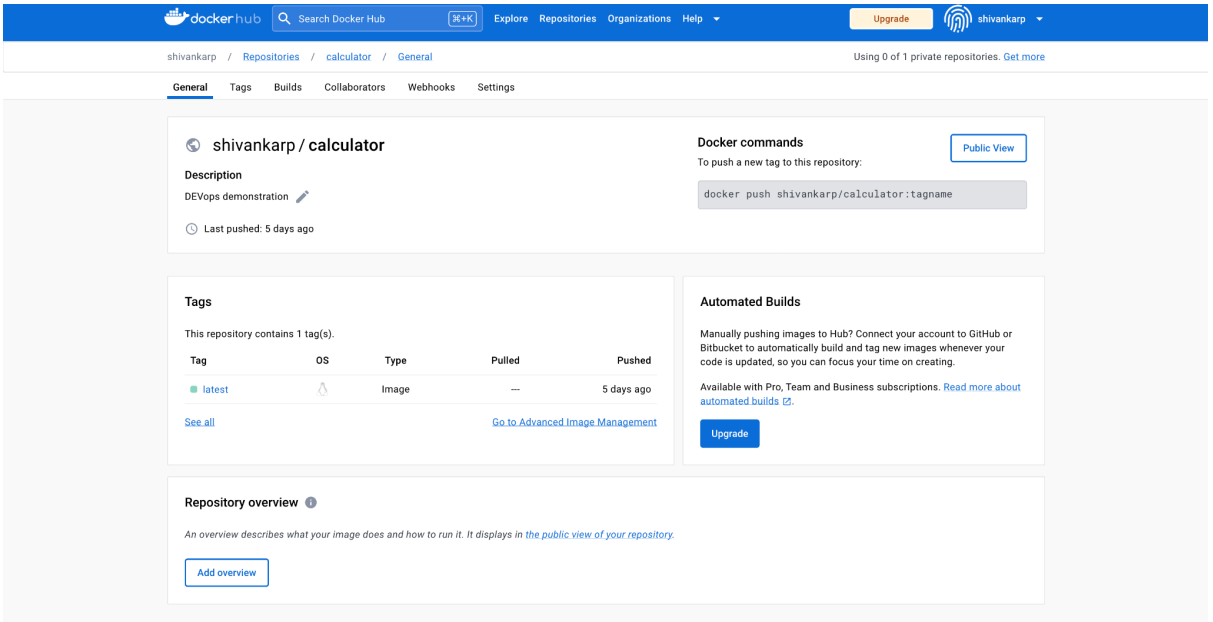
```
                }
            }
        }
    }
    stage('Stage 5: Clean docker images'){
        steps{
            script{
                sh 'docker container prune -f'
                sh 'docker image prune -f'
            }
        }
    }
    stage('Step 6: Ansible Deployment'){
        steps{
            ansiblePlaybook becomeUser: null,
            colorized: true,
            credentialsId: 'localhost',
            disableHostKeyChecking: true,
            installation: 'Ansible',
            inventory: 'Deployment/inventory',
            playbook: 'Deployment/deploy.yml',
            sudoUser: null
        }
    }
  }
}
```



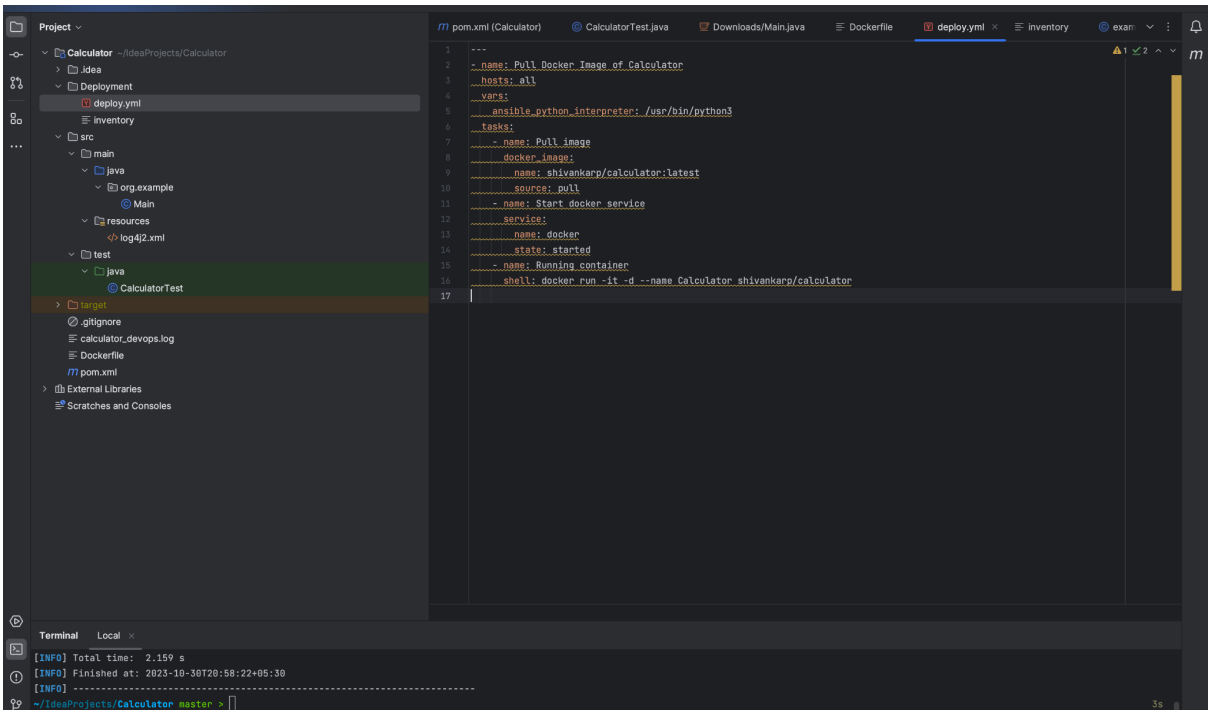
This has 6 stages:

- Git clone
- Maven Build
- Build docker image
- Push image to docker hub
- Clean docker image
- Ansible deployment

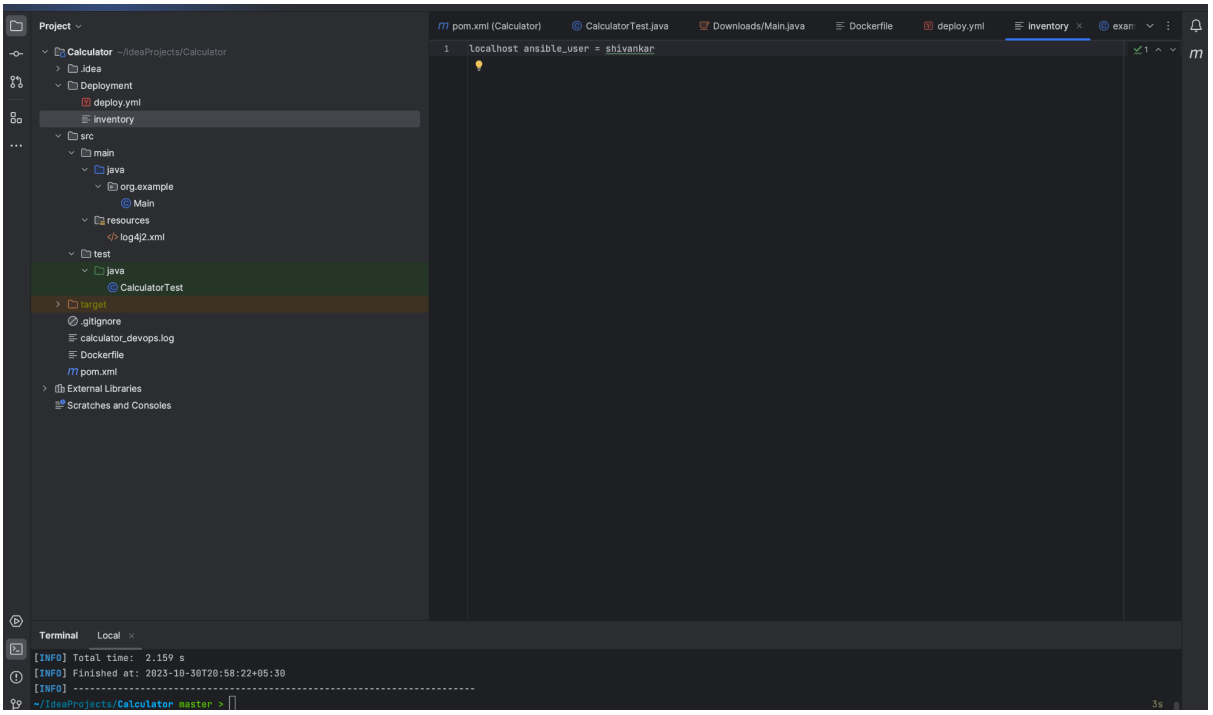
This is the docker image in docker hub



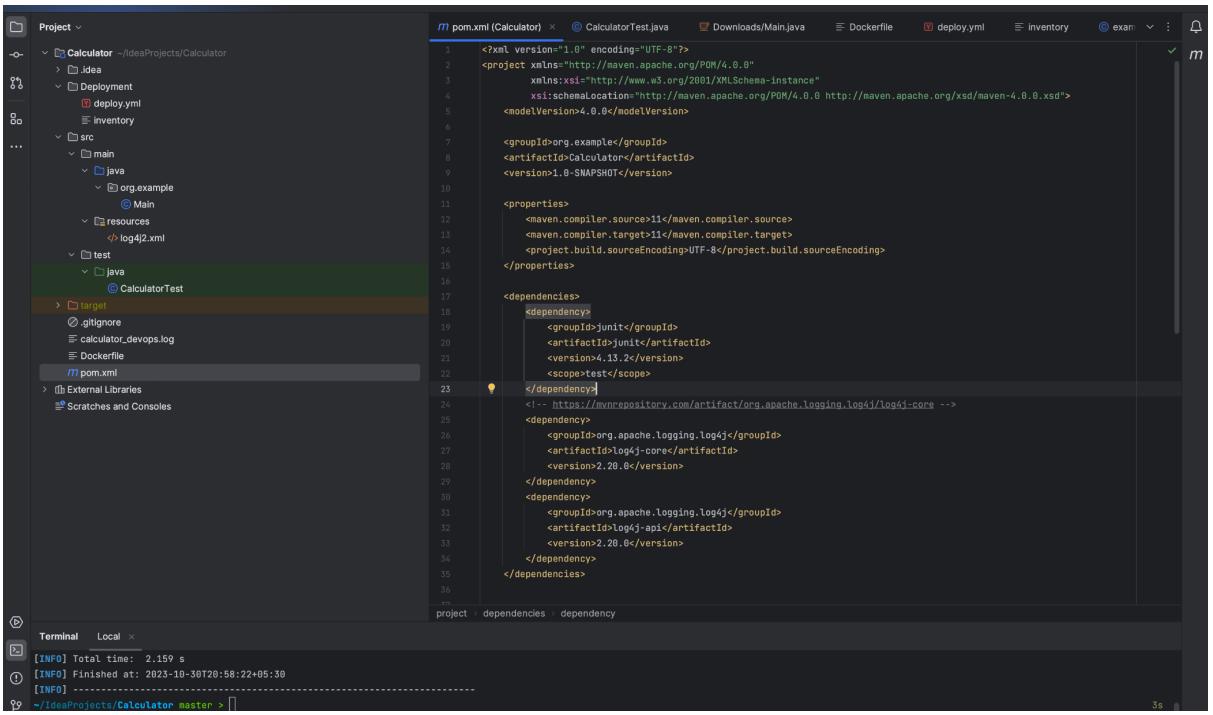
To configure Ansible:
deploy.yml



Inventory

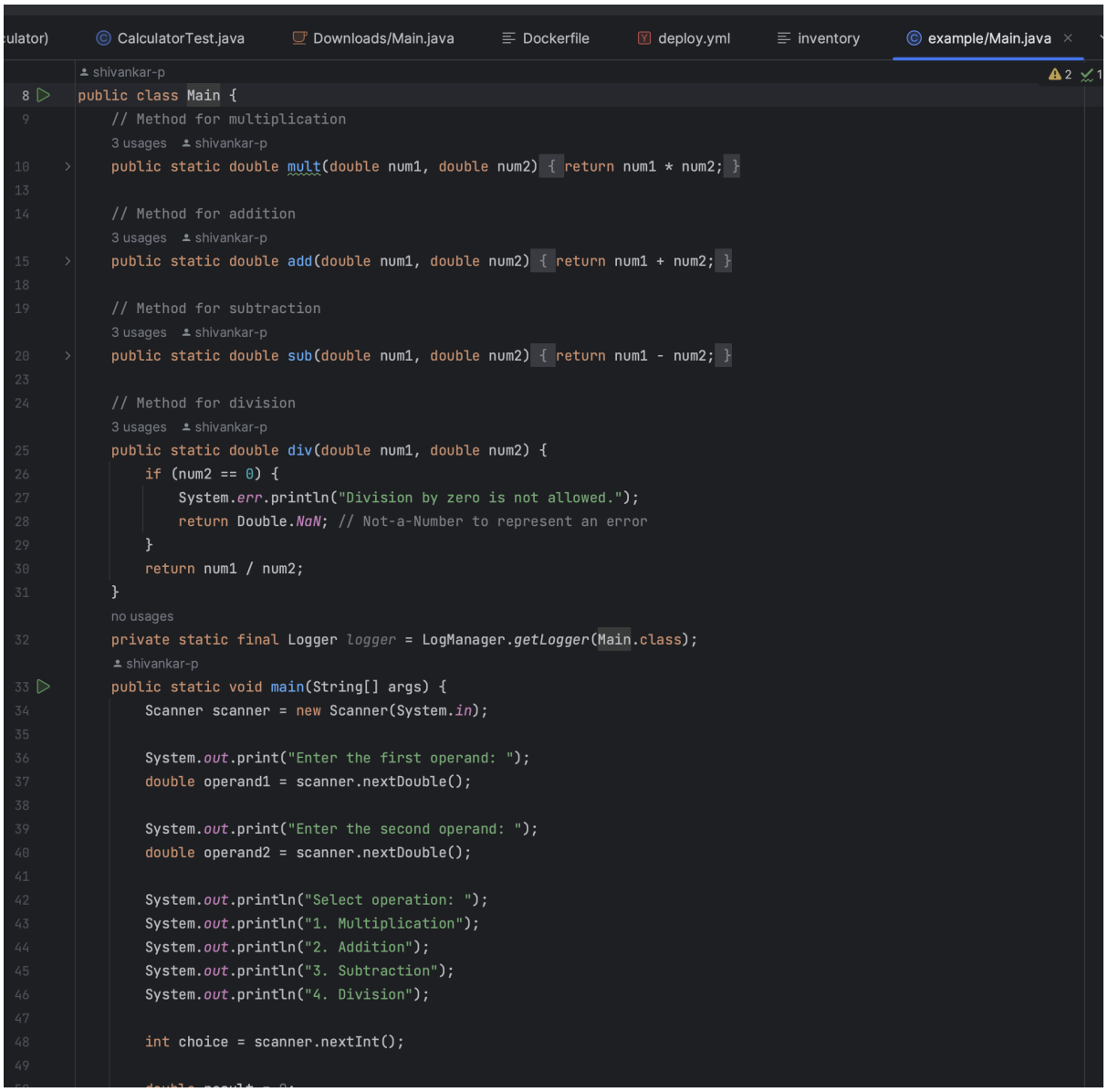


Pom.xml is also configured appropriately to generate logging using log4j2 and junit packages for testing

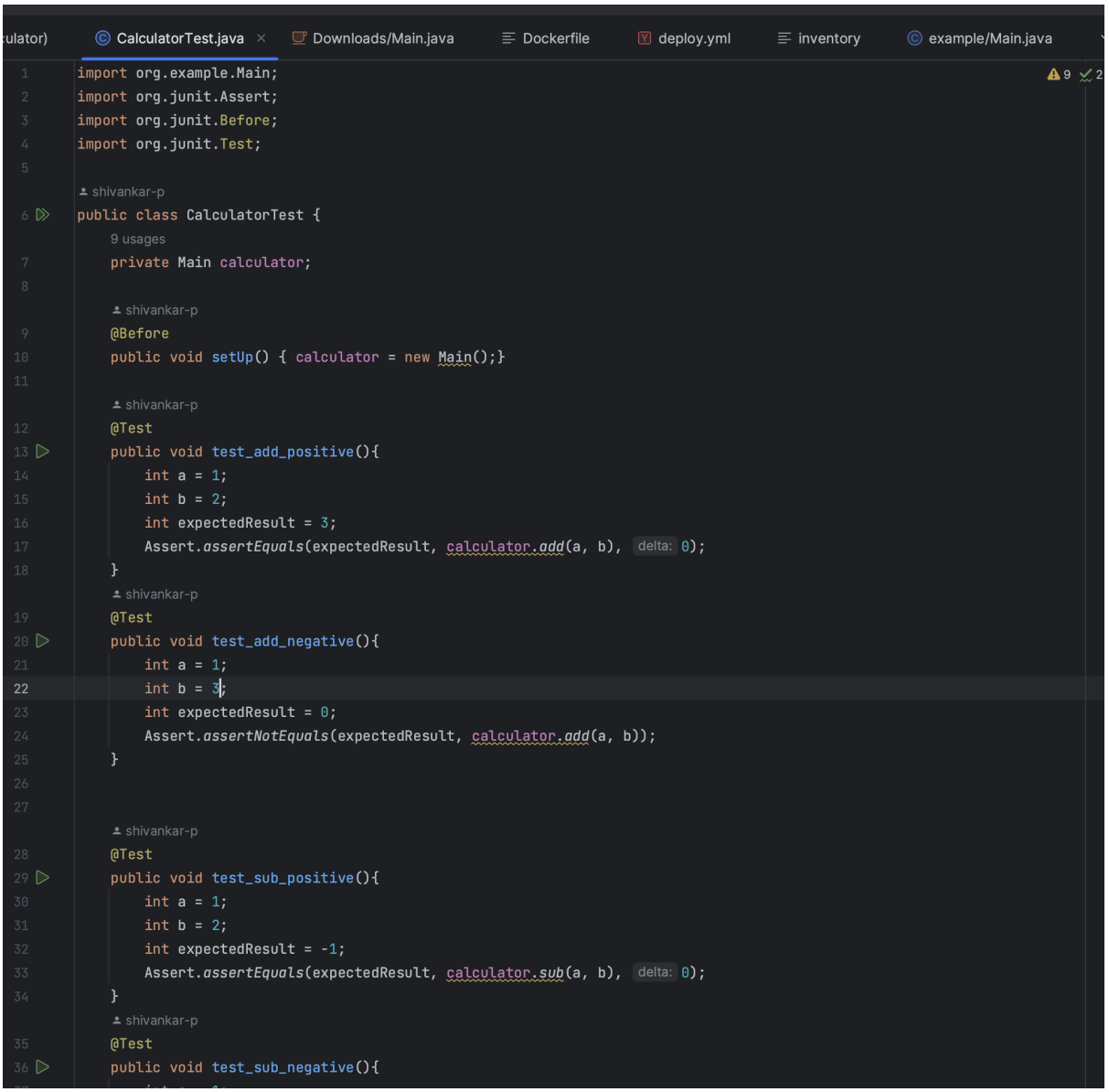


The java calculator program handle the following operations:

- Add
- Subtract
- Multiply
- Divide


ulator)
CalculatorTest.java
Downloads/Main.java
Dockerfile
deploy.yml
inventory
example/Main.java
shivankar-p
2
1
8 public class Main {
9 // Method for multiplication
3 usages shivankar-p
10 public static double mult(double num1, double num2) { return num1 * num2; }
13
14 // Method for addition
3 usages shivankar-p
15 public static double add(double num1, double num2) { return num1 + num2; }
18
19 // Method for subtraction
3 usages shivankar-p
20 public static double sub(double num1, double num2) { return num1 - num2; }
23
24 // Method for division
3 usages shivankar-p
25 public static double div(double num1, double num2) {
26 if (num2 == 0) {
27 System.err.println("Division by zero is not allowed.");
28 return Double.NaN; // Not-a-Number to represent an error
29 }
30 return num1 / num2;
31 }
no usages
32 private static final Logger logger = LogManager.getLogger(Main.class);
shivankar-p
33 public static void main(String[] args) {
34 Scanner scanner = new Scanner(System.in);
35
36 System.out.print("Enter the first operand: ");
37 double operand1 = scanner.nextDouble();
38
39 System.out.print("Enter the second operand: ");
40 double operand2 = scanner.nextDouble();
41
42 System.out.println("Select operation: ");
43 System.out.println("1. Multiplication");
44 System.out.println("2. Addition");
45 System.out.println("3. Subtraction");
46 System.out.println("4. Division");
47
48 int choice = scanner.nextInt();
49

This is the Junit test file



Running the container: