DevOps Project Report: Scientific Calculator

Name: Ayyan Pasha
Roll Number: MT2024029

1. Introduction to DevOps

This project focuses on designing and deploying a **Scientific Calculator** application using modern DevOps practices. The application is built using Java and supports the following operations:

- **Square Root (√x)**
- **Factorial (x!)**
- **Natural Logarithm (ln(x))**
- **Power Function (x^b)**

The project involves creating a Java-based application, testing it with JUnit, building it using Maven, containerizing it with Docker, and deploying it using Ansible. The entire workflow is automated using Jenkins for continuous integration and deployment.

2. Tools Used

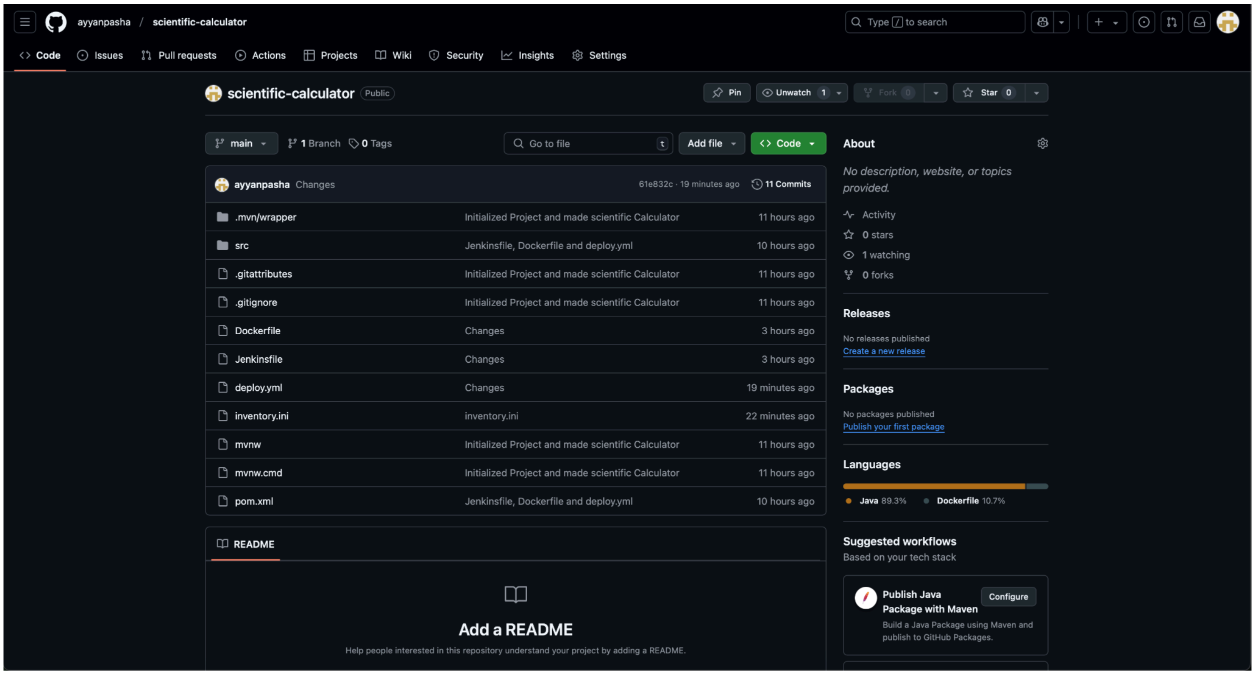
Tool	Purpose
GitHub	Source Code Management
JUnit	Unit Testing
Maven	Build Automation
Jenkins	Continuous Integration & Delivery (CI/CD)
Docker	Containerization
Ansible	Configuration Management & Deployment

3. Project Setup

Git Commands

git clone https://github.com/ayyanpasha/scientific-calculator.git
git add .
git commit -m "Changes"
git push origin main

Screenshot 1: GitHub Repository



ayyanpasha Changes 61e832c · 19 minutes ago 11 Commits

.mvn/wrapper	Initialized Project and made scientific Calculator	11 hours ago
src	Jenkinsfile, Dockerfile and deploy.yml	10 hours ago
.gitattributes	Initialized Project and made scientific Calculator	11 hours ago
.gitignore	Initialized Project and made scientific Calculator	11 hours ago
Dockerfile	Changes	3 hours ago
Jenkinsfile	Changes	3 hours ago
deploy.yml	Changes	19 minutes ago
inventory.ini	inventory.ini	22 minutes ago
mvnw	Initialized Project and made scientific Calculator	11 hours ago
mvnw.cmd	Initialized Project and made scientific Calculator	11 hours ago
pom.xml	Jenkinsfile, Dockerfile and deploy.yml	10 hours ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Java 89.3% Dockerfile 10.7%

Suggested workflows
Based on your tech stack

Publish Java Package with Maven
Build a Java Package using Maven and publish to GitHub Packages.

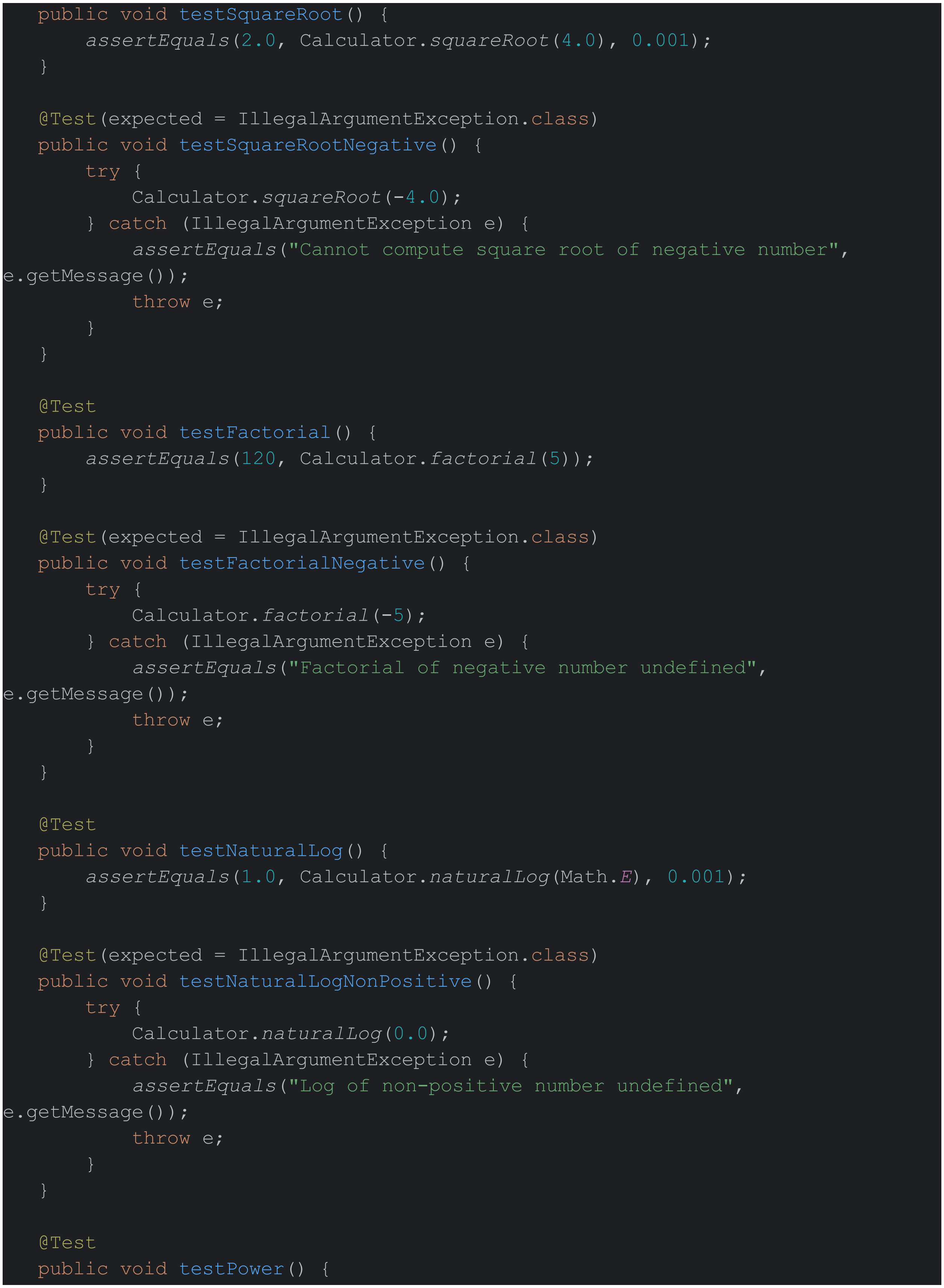
4. Testing with JUnit

Test Cases

- Validated edge cases (e.g., factorial of 0, square root of negative numbers).
- Parameterized tests for multiple inputs.

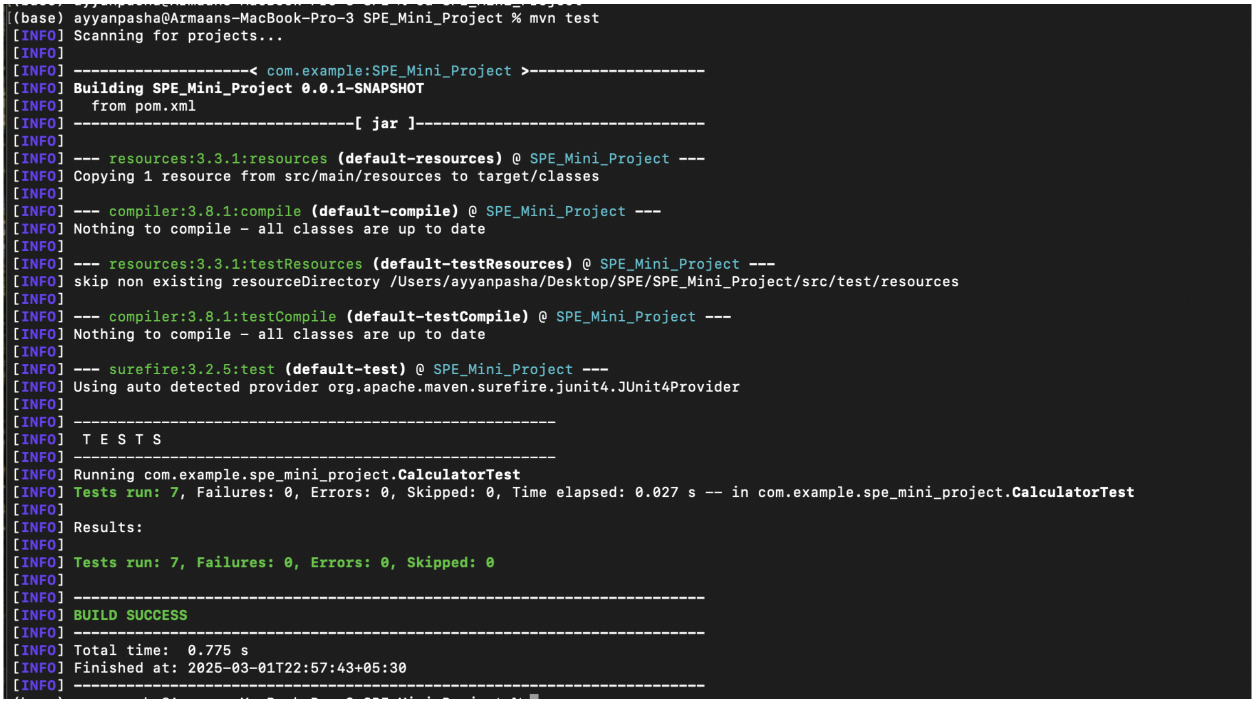
Code Snippet: `CalculatorTest.java`

```
package com.example.spe_mini_project;

import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {
   @Test
```



```
    public void testSquareRoot() {
        assertEquals(2.0, Calculator.squareRoot(4.0), 0.001);
    }

    @Test(expected = IllegalArgumentException.class)
    public void testSquareRootNegative() {
        try {
            Calculator.squareRoot(-4.0);
        } catch (IllegalArgumentException e) {
            assertEquals("Cannot compute square root of negative number",
e.getMessage());
            throw e;
        }
    }

    @Test
    public void testFactorial() {
        assertEquals(120, Calculator.factorial(5));
    }

    @Test(expected = IllegalArgumentException.class)
    public void testFactorialNegative() {
        try {
            Calculator.factorial(-5);
        } catch (IllegalArgumentException e) {
            assertEquals("Factorial of negative number undefined",
e.getMessage());
            throw e;
        }
    }

    @Test
    public void testNaturalLog() {
        assertEquals(1.0, Calculator.naturalLog(Math.E), 0.001);
    }

    @Test(expected = IllegalArgumentException.class)
    public void testNaturalLogNonPositive() {
        try {
            Calculator.naturalLog(0.0);
        } catch (IllegalArgumentException e) {
            assertEquals("Log of non-positive number undefined",
e.getMessage());
            throw e;
        }
    }

    @Test
    public void testPower() {
        assertEquals(8.0, Calculator.power(2.0, 3.0), 0.001);
    }
}
```

Screenshot 2: JUnit Test Results



```
(base) ayyanpasha@Armaans-MacBook-Pro-3 SPE_Mini_Project % mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< com.example:SPE_Mini_Project >--------------------
[INFO] Building SPE_Mini_Project 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ SPE_Mini_Project ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ SPE_Mini_Project ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ SPE_Mini_Project ---
[INFO] skip non existing resourceDirectory /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/src/test/resources
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ SPE_Mini_Project ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ SPE_Mini_Project ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.example.spe_mini_project.CalculatorTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 s -- in com.example.spe_mini_project.CalculatorTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.775 s
[INFO] Finished at: 2025-03-01T22:57:43+05:30
[INFO] ------------------------------------------------------------------------
```

---

5. Build with Maven

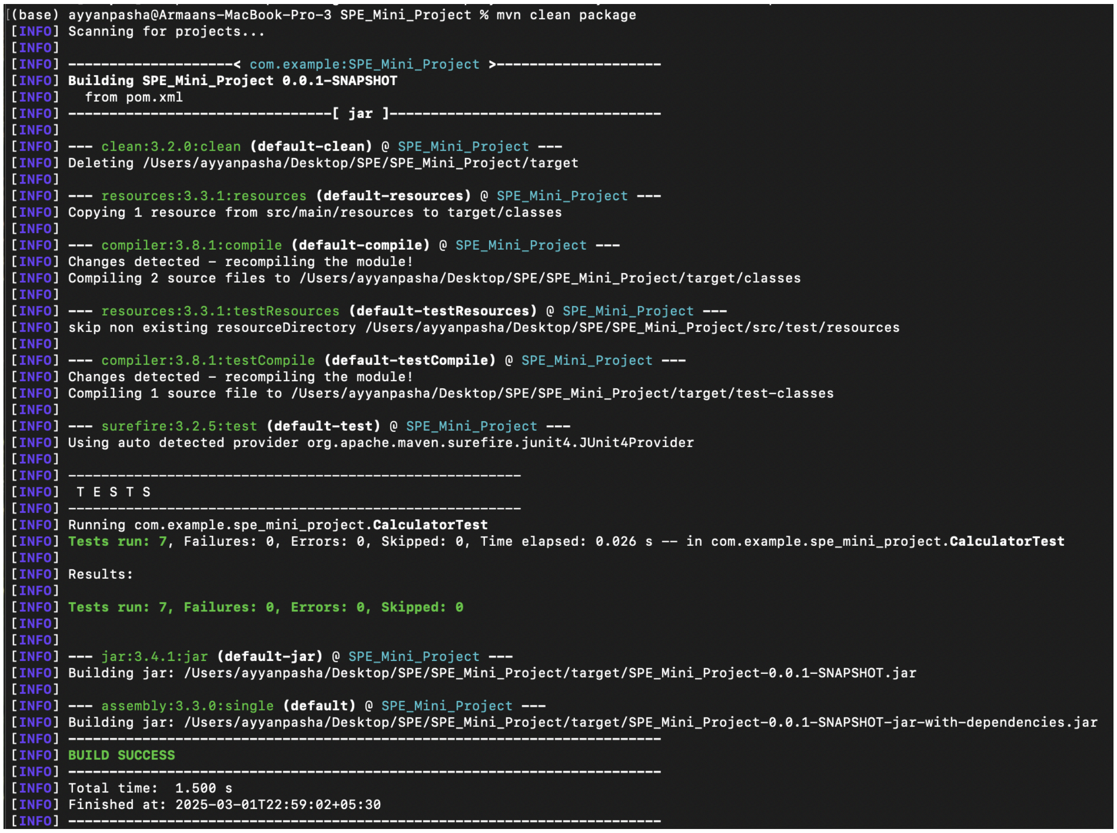
pom.xml Configuration

- Added dependencies for JUnit and Maven Assembly Plugin.
- Configured `<mainClass>` for executable JAR.

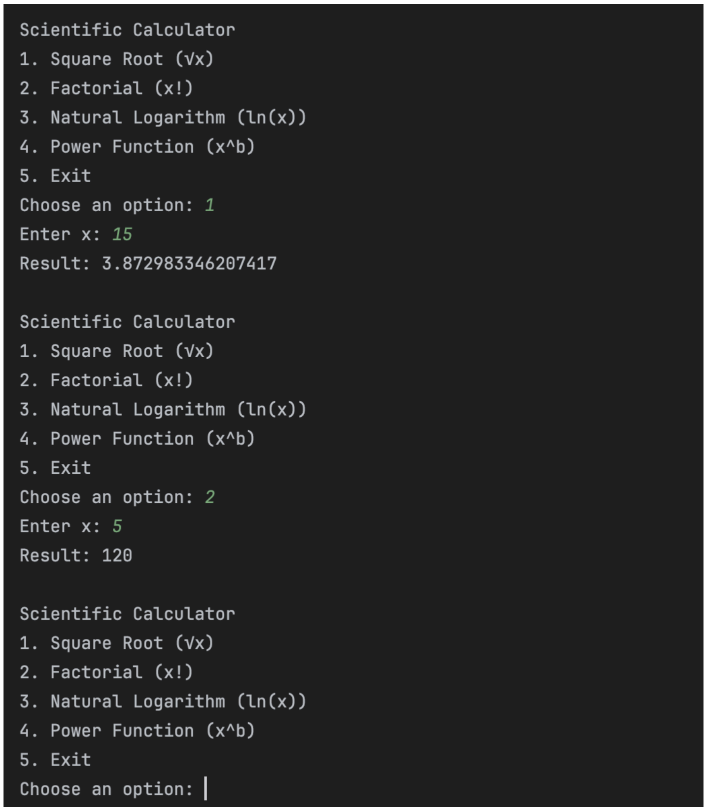
Build Command:

mvn clean package

Screenshot 3: Maven Build Output



```
(base) ayyanpasha@Armaans-MacBook-Pro-3 SPE_Mini_Project % mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< com.example:SPE_Mini_Project >--------------------
[INFO] Building SPE_Mini_Project 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ SPE_Mini_Project ---
[INFO] Deleting /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ SPE_Mini_Project ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ SPE_Mini_Project ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/target/classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ SPE_Mini_Project ---
[INFO] skip non existing resourceDirectory /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/src/test/resources
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ SPE_Mini_Project ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/target/test-classes
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ SPE_Mini_Project ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.example.spe_mini_project.CalculatorTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.026 s -- in com.example.spe_mini_project.CalculatorTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ SPE_Mini_Project ---
[INFO] Building jar: /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/target/SPE_Mini_Project-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- assembly:3.3.0:single (default) @ SPE_Mini_Project ---
[INFO] Building jar: /Users/ayyanpasha/Desktop/SPE/SPE_Mini_Project/target/SPE_Mini_Project-0.0.1-SNAPSHOT-jar-with-dependencies.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.500 s
[INFO] Finished at: 2025-03-01T22:59:02+05:30
[INFO] ------------------------------------------------------------------------
```



```
Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Logarithm (ln(x))
4. Power Function (x^b)
5. Exit
Choose an option: 1
Enter x: 15
Result: 3.872983346207417

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Logarithm (ln(x))
4. Power Function (x^b)
5. Exit
Choose an option: 2
Enter x: 5
Result: 120

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Logarithm (ln(x))
4. Power Function (x^b)
5. Exit
Choose an option: 
```

6. Continuous Integration with Jenkins

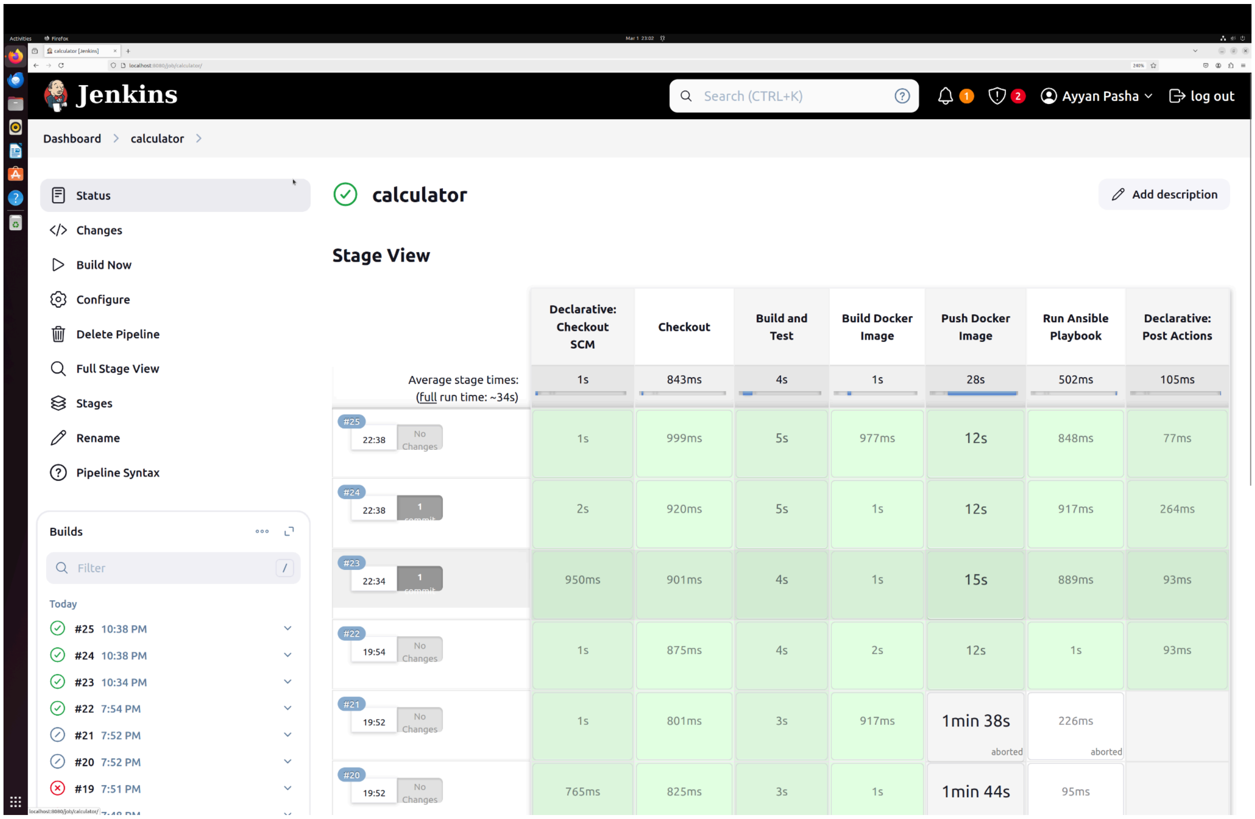
Jenkins Pipeline

- **Stages:** Checkout → Build → Test → Docker Build/Push → Ansible Deployment.

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE_NAME = 'scientific-calculator'
        GITHUB_REPO_URL = 'https://github.com/ayyanpasha/scientific-calculator.git'
        DOCKER_REGISTRY = 'docker.io'
        IMAGE_TAG = '0.0.1'
    }
    stages {
        stage('Checkout') {
            steps {
                script {
                    git branch: 'main', url: "${GITHUB_REPO_URL}"
                }
            }
        }
        stage('Build and Test') {
            steps {
                script {
                    sh 'mvn clean install'
                }
            }
        }
        stage('Build Docker Image') {
            steps {
                script {
                    docker.build("${DOCKER_IMAGE_NAME}:${IMAGE_TAG}", '.')
                    // Optionally, also tag the image as "latest" for consistency
                    docker.build("${DOCKER_IMAGE_NAME}:latest", '.')
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                script {
                    docker.withRegistry('', 'DockerHubCred') {
                        // Tag the image with the appropriate version and latest
                        sh "docker tag ${DOCKER_IMAGE_NAME}:${IMAGE_TAG} ${DOCKER_REGISTRY}/ayyanpasha/scientific-calculator:${IMAGE_TAG}"
                        sh "docker tag ${DOCKER_IMAGE_NAME}:${IMAGE_TAG} ${DOCKER_REGISTRY}/ayyanpasha/scientific-calculator:latest"

                        // Push both the versioned tag and latest tag
                        sh "docker push ${DOCKER_REGISTRY}/ayyanpasha/scientific-calculator:${IMAGE_TAG}"
                        sh "docker push ${DOCKER_REGISTRY}/ayyanpasha/scientific-calculator:latest"
                    }
                }
            }
        }
        stage('Run Ansible Playbook') {
            steps {
                script {
                    ansiblePlaybook(
                        playbook: 'deploy.yml',
                        inventory: 'inventory'
                    )
                }
            }
        }
    }
    post {
        success {
            echo 'Pipeline successfully completed!'
        }
        failure {
            echo 'Pipeline failed!'
            // Add failure notifications here (e.g., Slack, email)
```

```
●               }
●           }
●       }
```

Screenshot 4: Jenkins Pipeline



calculator

Stage View

	Declarative: Checkout SCM	Checkout	Build and Test	Build Docker Image	Push Docker Image	Run Ansible Playbook	Declarative: Post Actions
Average stage times: (full run time: ~34s)	1s	843ms	4s	1s	28s	502ms	105ms
#25 22:38 No Changes	1s	999ms	5s	977ms	12s	848ms	77ms
#24 22:38 1 commit	2s	920ms	5s	1s	12s	917ms	264ms
#23 22:34 1 commit	950ms	901ms	4s	1s	15s	889ms	93ms
#22 19:54 No Changes	1s	875ms	4s	2s	12s	1s	93ms
#21 19:52 No Changes	1s	801ms	3s	917ms	1min 38s aborted	226ms aborted	
#20 19:52 No Changes	765ms	825ms	3s	1s	1min 44s	95ms	

Today

- #25 10:38 PM
- #24 10:38 PM
- #23 10:34 PM
- #22 7:54 PM
- #21 7:52 PM
- #20 7:52 PM
- #19 7:51 PM

7. Containerization with Docker

Dockerfile

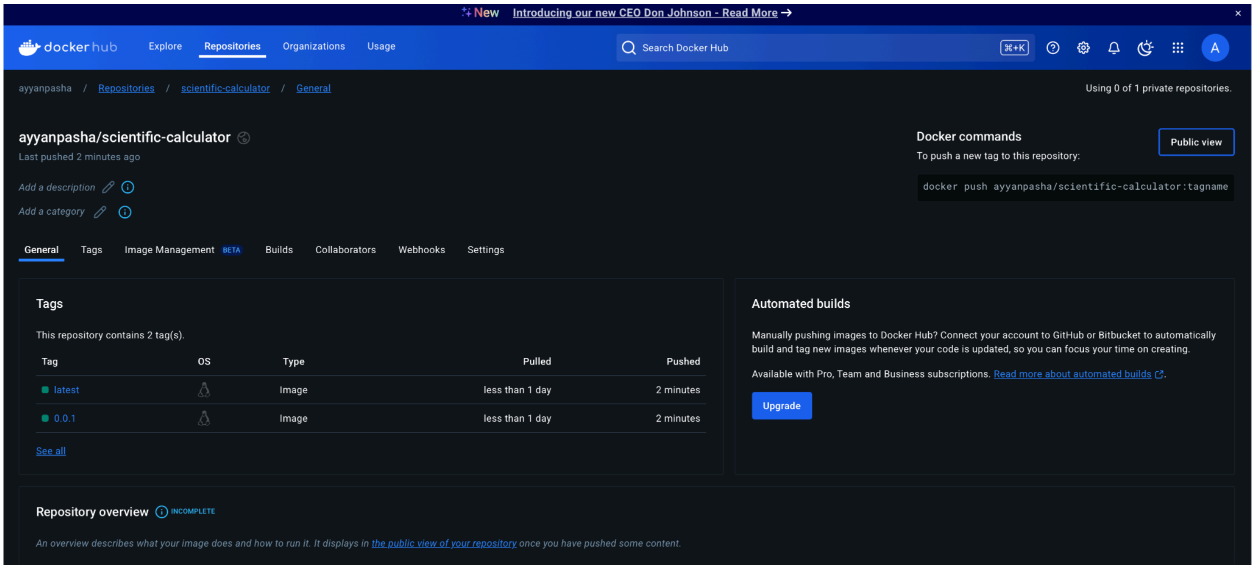
```
# Stage 1: Build the project using Maven
FROM maven:3.9.9-amazoncorretto-17 AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
RUN mvn clean package -DskipTests

# Stage 2: Run the application with Amazon Corretto (OpenJDK 17)
FROM amazoncorretto:17
WORKDIR /app
```

```
# Copy the generated JAR file from the builder stage
COPY --from=build /app/target/*-jar-with-dependencies.jar
/app/SPE_Mini_Project-0.0.1-SNAPSHOT.jar

# Make sure the entry point uses the correct JAR name
ENTRYPOINT ["java", "-jar", "SPE_Mini_Project-0.0.1-SNAPSHOT.jar"]
```

Docker Commands

docker build -t ayyanpasha/scientific-calculator:0.0.1 .
docker push ayyanpasha/scientific-calculator:0.0.1

Screenshot 5: Docker Image on Docker Hub



Using 0 of 1 private repositories.

ayyanpasha/scientific-calculator

Last pushed 2 minutes ago

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push ayyanpasha/scientific-calculator:tagname
```

Tags

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	2 minutes
0.0.1		Image	less than 1 day	2 minutes

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Repository overview

An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

---

8. Deployment with Ansible

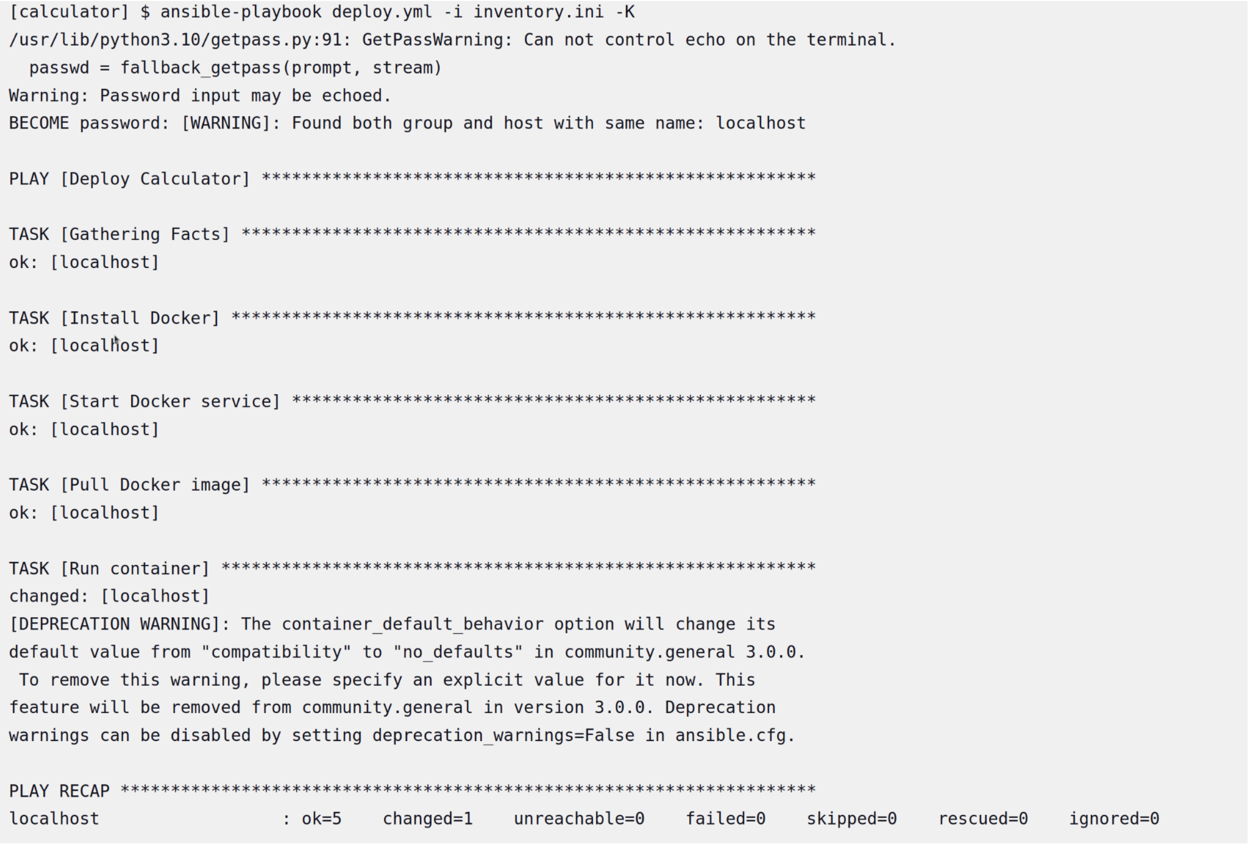
Ansible Playbook (`deploy.yml`)

```
- hosts: all
 become: yes
 tasks:
   - name: Install Docker
     apt:
       name: docker.io
       state: present
   - name: Start Docker service
     service:
       name: docker
       state: started
   - name: Pull Docker image
     docker_image:
       name: ayyanpasha/scientific-calculator
       source: pull
   - name: Run container
     docker_container:
       name: calculator
       image: ayyanpasha/scientific-calculator
       state: started
       ports:
         - "8081:8080"
```

Inventory File (`inventory.ini`)

```
[localhost]
localhost ansible_connection=local
```

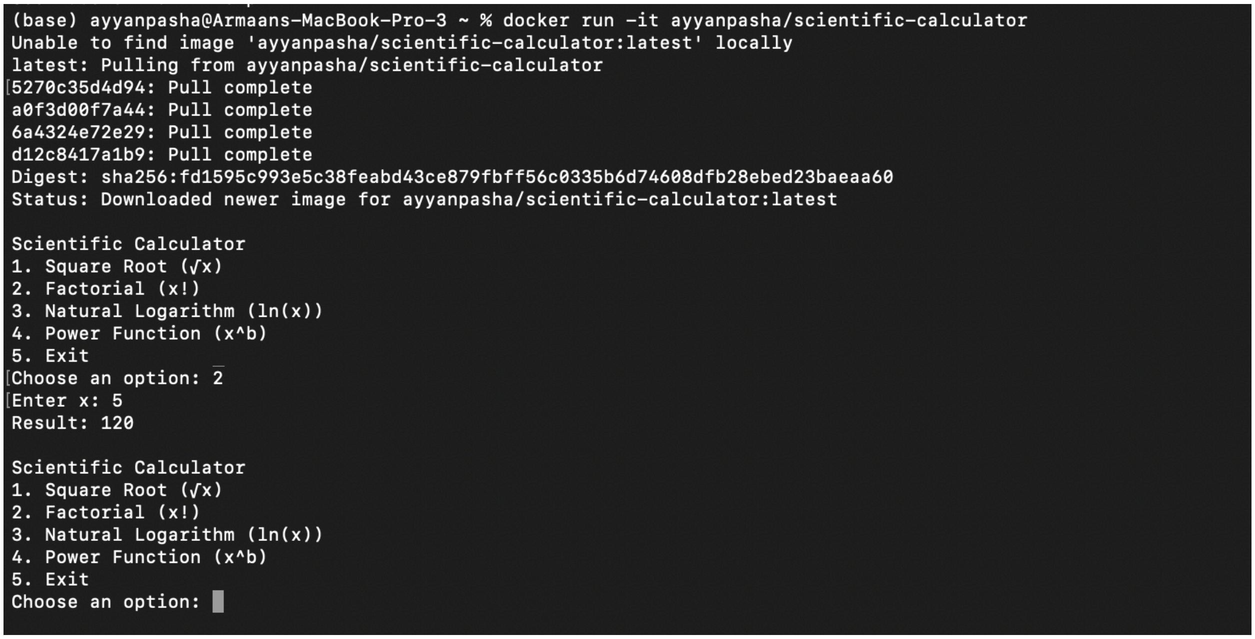
Screenshot 6: Ansible Deployment Output



```
[calculator] $ ansible-playbook deploy.yml -i inventory.ini -K
/usr/lib/python3.10/getpass.py:91: GetPassWarning: Can not control echo on the terminal.
  passwd = fallback_getpass(prompt, stream)
Warning: Password input may be echoed.
BECOME password: [WARNING]: Found both group and host with same name: localhost

PLAY [Deploy Calculator] *******************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Install Docker] **********************************************************
ok: [localhost]

TASK [Start Docker service] ****************************************************
ok: [localhost]

TASK [Pull Docker image] *******************************************************
ok: [localhost]

TASK [Run container] ***********************************************************
changed: [localhost]
[DEPRECATION WARNING]: The container_default_behavior option will change its
default value from "compatibility" to "no_defaults" in community.general 3.0.0.
 To remove this warning, please specify an explicit value for it now. This
feature will be removed from community.general in version 3.0.0. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY RECAP *********************************************************************
localhost                  : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

9. Application Screenshots

Screenshot 7: CLI Interface & Sample Calculation (Factorial of 5)



```
(base) ayyanpasha@Armaans-MacBook-Pro-3 ~ % docker run -it ayyanpasha/scientific-calculator
Unable to find image 'ayyanpasha/scientific-calculator:latest' locally
latest: Pulling from ayyanpasha/scientific-calculator
5270c35d4d94: Pull complete
a0f3d00f7a44: Pull complete
6a4324e72e29: Pull complete
d12c8417a1b9: Pull complete
Digest: sha256:fd1595c993e5c38feabd43ce879fbff56c0335b6d74608dfb28ebed23baeaa60
Status: Downloaded newer image for ayyanpasha/scientific-calculator:latest

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Logarithm (ln(x))
4. Power Function (x^b)
5. Exit
Choose an option: 2
Enter x: 5
Result: 120

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Logarithm (ln(x))
4. Power Function (x^b)
5. Exit
Choose an option: 
```

10. Troubleshooting

1. **Maven Command Not Found:**
   - Installed Maven and added it to `PATH`.
2. **Docker Permission Denied:**
   - sudo usermod -aG docker jenkins
3. **Jenkins Docker Plugin Error:**
   - Installed Docker Pipeline plugin.

11. Conclusion

This project demonstrates a complete DevOps pipeline from code commit to deployment. The integration of GitHub, Maven, Jenkins, Docker, and Ansible ensures efficient and automated software delivery.

Links:

- **GitHub Repo:** github.com/ayyanpasha/scientific-calculator
- **Docker Image:** hub.docker.com/r/ayyanpasha/scientific-calculator

Submitted By:
Ayyan Pasha
Roll Number: MT2024029