Scientific Calculator with DevOps - Project Report

Akash Upadhyay

MT2024013

1. Introduction

This report outlines the implementation of a Scientific Calculator using Java and DevOps principles. The project integrates various DevOps tools and follows a CI/CD pipeline to automate testing, building, containerization, and deployment.

2. Problem Statement

The goal of this project is to develop a command-line-based scientific calculator that performs the following operations:

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

The project is implemented using Java and follows DevOps practices to automate its development lifecycle.

3. Tools Used

The following tools were used in the project:

- **Java**: Programming language for the scientific calculator.
- **JUnit 5**: For unit testing the calculator functions.
- **Maven**: For dependency management and build automation.
- **GitHub**: Source control management.
- **Jenkins**: Continuous Integration (CI) and Continuous Deployment (CD).
- **Docker**: Containerization of the application.
- **Docker Hub**: To store and share Docker images.
- **Ansible**: For automated deployment.

4. Implementation Steps

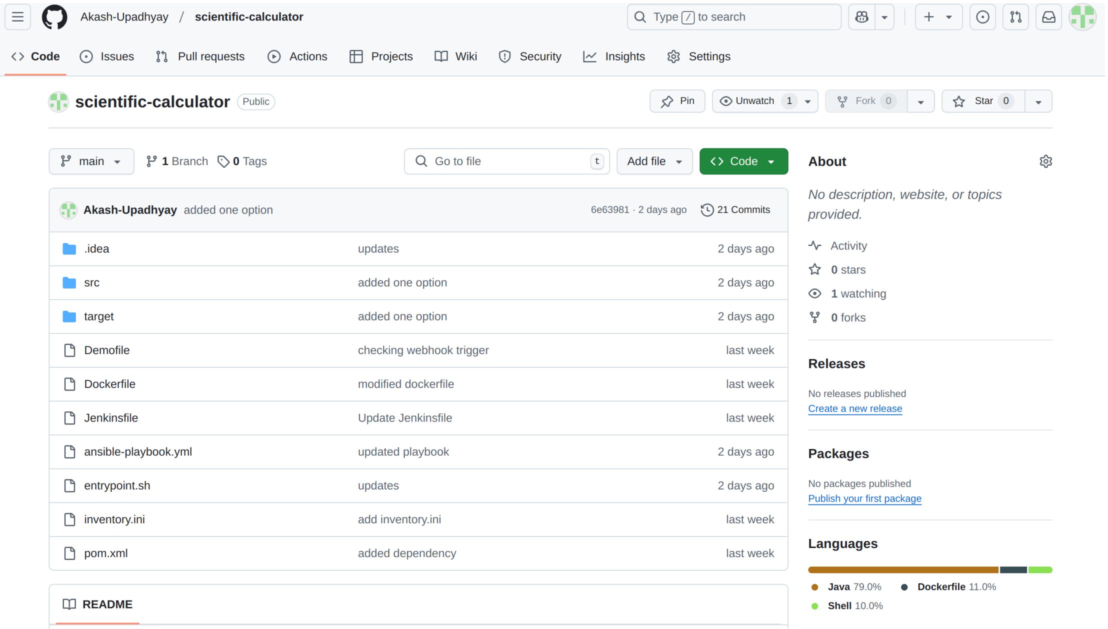
4.1 Source Control Management with GitHub

- A GitHub repository was created to store the source code.
- The repository is cloned locally using:

```
git clone <repository-url>
```

- Changes are committed and pushed using:

```
git add .
git commit -m "Initial commit"
git push origin main
```



Akash-Upadhyay added one option 6e63981 · 2 days ago 21 Commits

.idea	updates	2 days ago
src	added one option	2 days ago
target	added one option	2 days ago
Demofile	checking webhook trigger	last week
Dockerfile	modified dockerfile	last week
Jenkinsfile	Update Jenkinsfile	last week
ansible-playbook.yml	updated playbook	2 days ago
entrypoint.sh	updates	2 days ago
inventory.ini	add inventory.ini	last week
pom.xml	added dependency	last week

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** 79.0% ● **Dockerfile** 11.0% ● **Shell** 10.0%

4.2 Testing with JUnit 5

- JUnit test cases were written to verify the correctness of each mathematical function.
- The test cases are executed using Maven:

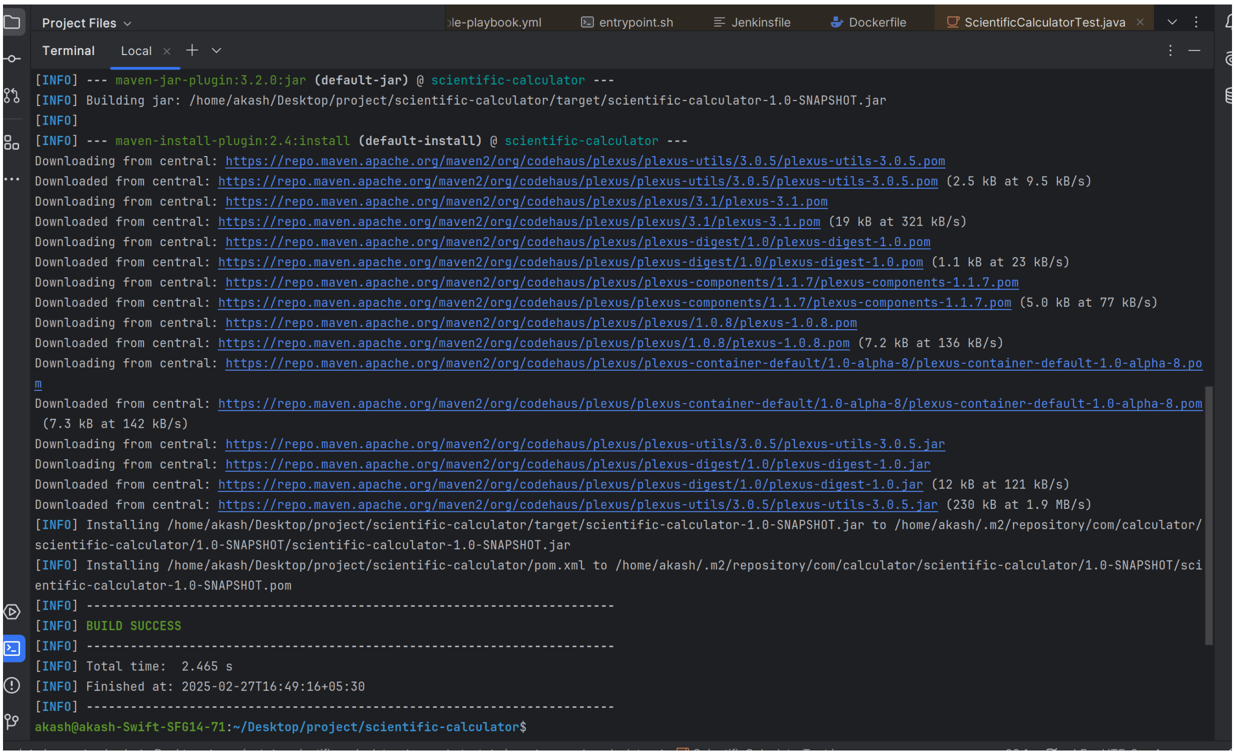
```
mvn test
```

4.3 Building the Project with Maven

- Maven is used to compile the Java code and package it into a JAR file.

```
mvn package
```

- The final JAR file is located in the `target` directory.



```
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ scientific-calculator ---
[INFO] Building jar: /home/akash/Desktop/project/scientific-calculator/target/scientific-calculator-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ scientific-calculator ---
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom (2.5 kB at 9.5 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom (19 kB at 321 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom (1.1 kB at 23 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom (5.0 kB at 77 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom (7.2 kB at 136 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.po
m
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/plexus-container-default-1.0-alpha-8.pom
 (7.3 kB at 142 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (12 kB at 121 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (230 kB at 1.9 MB/s)
[INFO] Installing /home/akash/Desktop/project/scientific-calculator/target/scientific-calculator-1.0-SNAPSHOT.jar to /home/akash/.m2/repository/com/calculator/
scientific-calculator/1.0-SNAPSHOT/scientific-calculator-1.0-SNAPSHOT.jar
[INFO] Installing /home/akash/Desktop/project/scientific-calculator/pom.xml to /home/akash/.m2/repository/com/calculator/scientific-calculator/1.0-SNAPSHOT/sci
entific-calculator-1.0-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.465 s
[INFO] Finished at: 2025-02-27T16:49:16+05:30
[INFO] ------------------------------------------------------------------------
akash@akash-Swift-SFG14-71:~/Desktop/project/scientific-calculator$
```

4.4 Continuous Integration with Jenkins

- A Jenkins pipeline is created to automate build, test, and deployment processes.
- The `Jenkinsfile` contains:

```
pipeline {
    agent any
    stages {
        stage('Clone Repository') {
            steps {
                git 'https://github.com/Akash-Upadhyay/scientific-
calculator.git'
            }
        }
        stage('Build with Maven') {
            steps {
                sh 'mvn package'
            }
        }
        stage('Run Tests') {
            steps {
                sh 'mvn test'
            }
        }
        stage('Build Docker Image') {
            steps {
                sh 'docker build -t mt2024013/scientific-calculator .'
            }
        }
        stage('Push to Docker Hub') {
            steps {
                withDockerRegistry([credentialsId: 'docker-hub-
credentials']) {
                    sh 'docker push mt2024013/scientific-calculator'
                }
            }
        }
        stage('Deploy Using Ansible') {
            steps {
                sh 'ansible-playbook -i inventory.ini ansible-
playbook.yml'
            }
        }
    }
}
```

4.5 Containerization with Docker

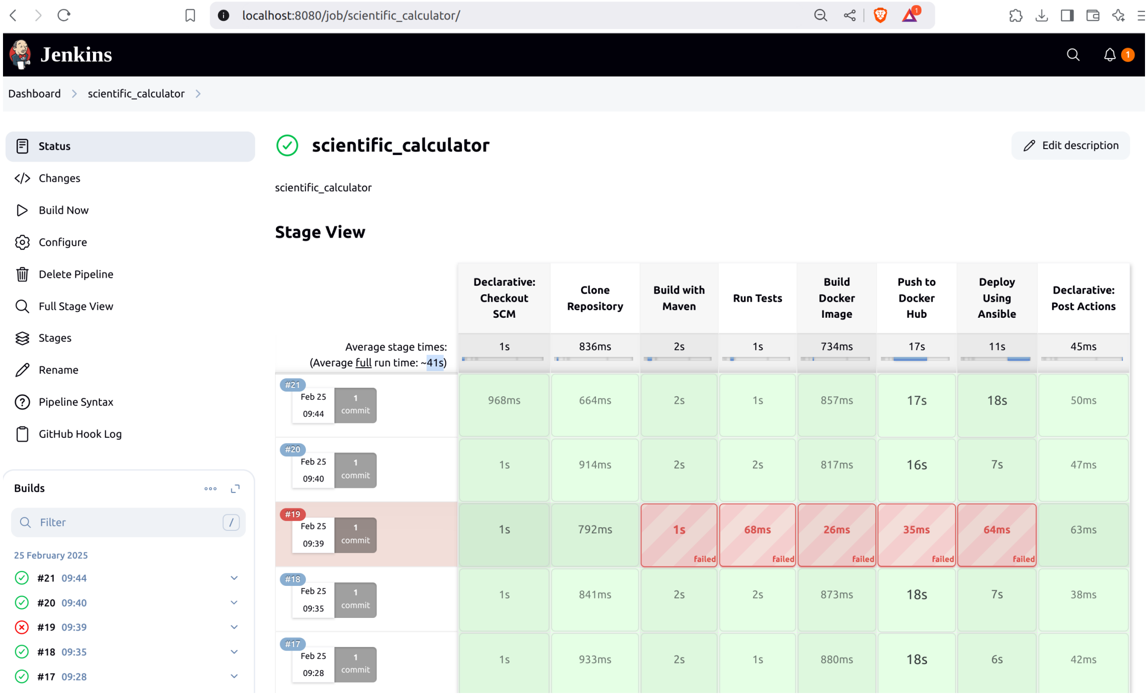
- A `Dockerfile` is created to containerize the application:

  ```
  FROM openjdk:17-jdk-slim

  WORKDIR /app
  COPY target/scientific-calculator-1.0-SNAPSHOT.jar /app/scientific-
  calculator.jar
  COPY entrypoint.sh /entrypoint.sh
  RUN chmod +x /entrypoint.sh
  CMD ["/entrypoint.sh"]
  ```

- The Docker image is built and pushed to Docker Hub:

  ```
  docker build -t mt2024013/scientific-calculator .
  docker push mt2024013/scientific-calculator
  ```



scientific_calculator

scientific_calculator

Stage View

	Declarative: Checkout SCM	Clone Repository	Build with Maven	Run Tests	Build Docker Image	Push to Docker Hub	Deploy Using Ansible	Declarative: Post Actions
Average stage times: (Average full run time: ~41s)	1s	836ms	2s	1s	734ms	17s	11s	45ms
#21 Feb 25 09:44 1 commit	968ms	664ms	2s	1s	857ms	17s	18s	50ms
#20 Feb 25 09:40 1 commit	1s	914ms	2s	2s	817ms	16s	7s	47ms
#19 Feb 25 09:39 1 commit	1s	792ms	1s failed	68ms failed	26ms failed	35ms failed	64ms failed	63ms
#18 Feb 25 09:35 1 commit	1s	841ms	2s	2s	873ms	18s	7s	38ms
#17 Feb 25 09:28 1 commit	1s	933ms	2s	1s	880ms	18s	6s	42ms

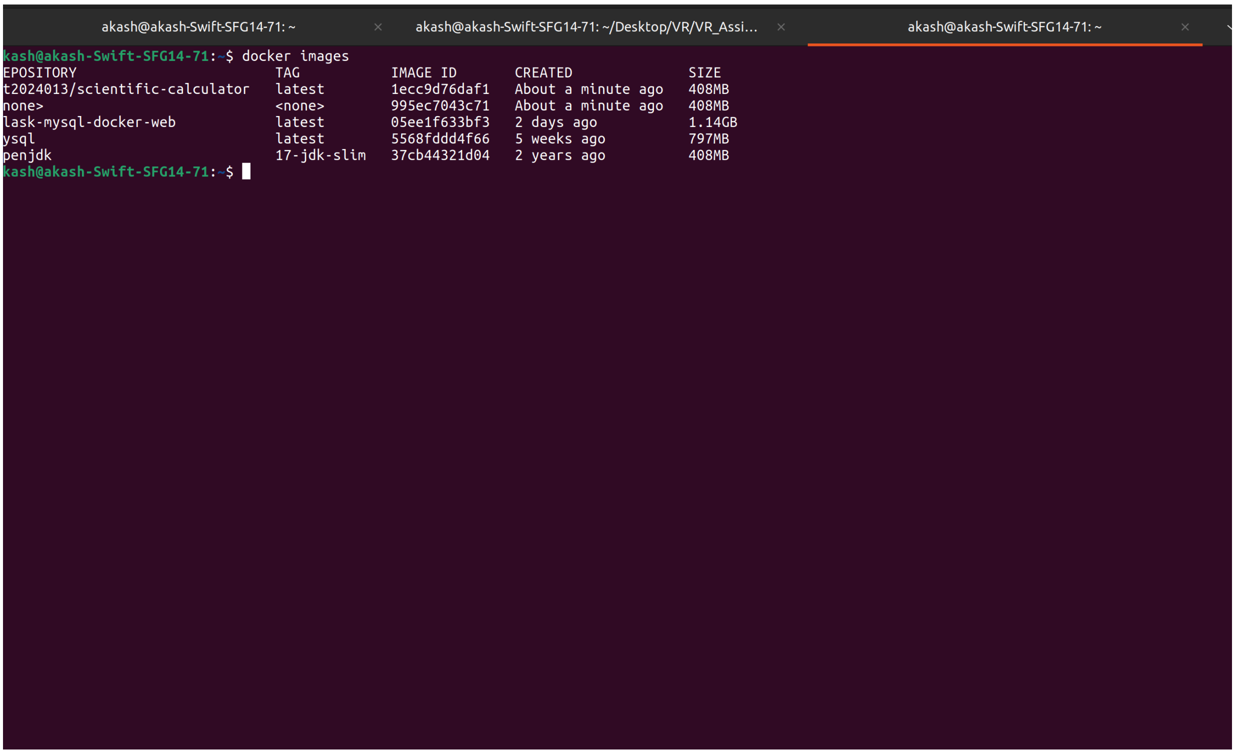
4.6 Deployment with Ansible

- An Ansible playbook is created for automated deployment:

```
- name: Deploy Scientific Calculator Container
  hosts: localhost
  become: true
  tasks:
    - name: Pull the latest Docker image
      command: docker pull mt2024013/scientific-calculator
    - name: Stop existing container (if running)
      command: docker stop calculator_container
      ignore_errors: yes
    - name: Remove old container (if exists)
      command: docker rm calculator_container
      ignore_errors: yes
    - name: Run new container
      command: docker run -d --name calculator_container
mt2024013/scientific-calculator
```

- The deployment is executed using:

```
ansible-playbook -i inventory.ini ansible-playbook.yml
```



akash@akash-Swift-SFG14-71: ~
akash@akash-Swift-SFG14-71: ~/Desktop/VR/VR_Assi...
akash@akash-Swift-SFG14-71: ~

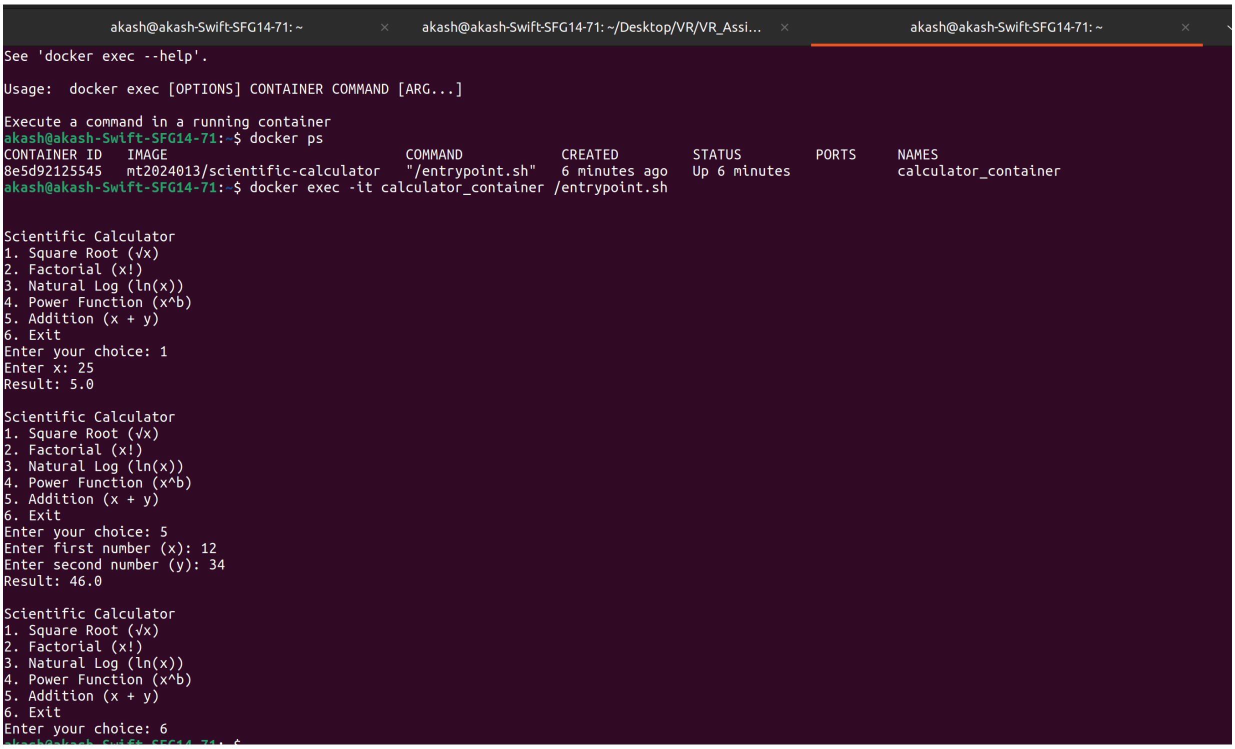
```
kash@akash-Swift-SFG14-71:~$ docker images
EPOSITORY                       TAG           IMAGE ID       CREATED              SIZE
t2024013/scientific-calculator   latest        1ecc9d76daf1   About a minute ago   408MB
none>                           <none>        995ec7043c71   About a minute ago   408MB
lask-mysql-docker-web           latest        05ee1f633bf3   2 days ago           1.14GB
ysql                            latest        5568fddd4f66   5 weeks ago          797MB
penjdk                          17-jdk-slim   37cb44321d04   2 years ago          408MB
kash@akash-Swift-SFG14-71:~$ 
```

5. Final Execution

- After deployment, the container keeps running.
- Users can attach to it and perform calculations using:

```
docker attach calculator_container

docker exec -it calculator_container /entrypoint.sh
```


akash@akash-Swift-SFG14-71: ~
akash@akash-Swift-SFG14-71: ~/Desktop/VR/VR_Assi...
akash@akash-Swift-SFG14-71: ~

```
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Execute a command in a running container
akash@akash-Swift-SFG14-71:~$ docker ps
CONTAINER ID   IMAGE                              COMMAND              CREATED         STATUS         PORTS     NAMES
8e5d92125545   mt2024013/scientific-calculator   "/entrypoint.sh"   6 minutes ago   Up 6 minutes             calculator_container
akash@akash-Swift-SFG14-71:~$ docker exec -it calculator_container /entrypoint.sh


Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Log (ln(x))
4. Power Function (x^b)
5. Addition (x + y)
6. Exit
Enter your choice: 1
Enter x: 25
Result: 5.0

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Log (ln(x))
4. Power Function (x^b)
5. Addition (x + y)
6. Exit
Enter your choice: 5
Enter first number (x): 12
Enter second number (y): 34
Result: 46.0

Scientific Calculator
1. Square Root (√x)
2. Factorial (x!)
3. Natural Log (ln(x))
4. Power Function (x^b)
5. Addition (x + y)
6. Exit
Enter your choice: 6
akash@akash-Swift-SFG14-71:~$
```

6. Conclusion

This project successfully integrates a Java-based scientific calculator with DevOps tools for continuous integration, testing, containerization, and automated deployment. The setup ensures a fully automated CI/CD pipeline, allowing seamless updates and deployments.