Mini Project Report
Scientific Calculator with DevOps Pipeline

Student Name: Areen Vaghasiya

Roll Number: IMT2022048

Project Title: Scientific Calculator with DevOps Pipeline

October 10, 2025

Contents

1 Introduction

1.1 Problem Statement

This project develops a scientific calculator implementing:

- Square root function: $\sqrt{x}$
- Factorial function: $x!$
- Natural logarithm (base e): $\ln(x)$
- Power function: x^b

1.2 Project Goal

The project implements a complete DevOps pipeline demonstrating CI/CD practices, automated testing, containerization, and configuration management.

DockerHub Repository: `https://hub.docker.com/r/areen9295/calc-app2`
Github Repository: `https://github.com/AV-AKIHIRO/Scientific-Calculator-SPE.git`

2 What and Why of DevOps?

2.1 What is DevOps?

DevOps unifies software development (Dev) and IT operations (Ops) to deliver applications faster and more reliably. It emphasizes:

- **Collaboration:** Breaking down barriers between development and operations teams
- **Automation:** Automating repetitive tasks throughout the software lifecycle
- **Continuous Feedback:** Establishing feedback loops for rapid iteration
- **Monitoring:** Continuous observation of application performance

Key components include Continuous Integration (CI), Continuous Delivery/Deployment (CD), Infrastructure as Code (IaC), automated testing, and monitoring.

2.2 Why DevOps?

DevOps adoption provides:

1. **Faster Release Cycles:** Automation enables rapid deployment
2. **Improved Quality:** Automated testing catches bugs early
3. **Enhanced Scalability:** Infrastructure as code facilitates scaling
4. **Reduced Deployment Risk:** Automated deployments with rollback capabilities
5. **Better Collaboration:** Unified tooling improves team communication
6. **Cost Efficiency:** Automation reduces manual effort

2.3 DevOps Lifecycle

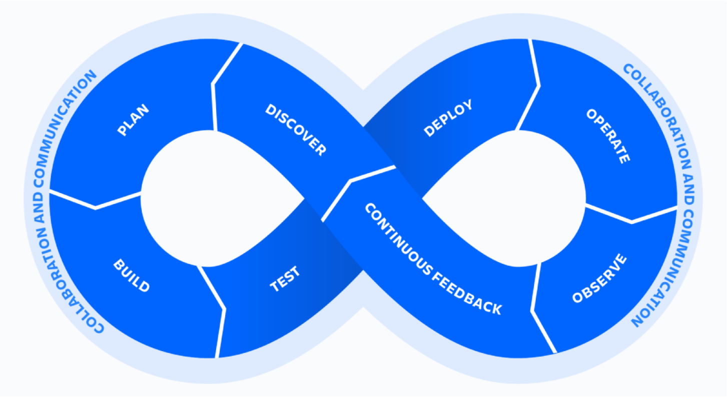

COLLABORATION AND COMMUNICATION
PLAN
DISCOVER
DEPLOY
OPERATE
COLLABORATION AND COMMUNICATION
BUILD
TEST
CONTINUOUS FEEDBACK
OBSERVE

Figure 1: DevOps Lifecycle

3 Tools Used

Table 1: Tools Used in the DevOps Pipeline

Pipeline Stage	**Tool**	**Purpose**
Source Control	Git + GitHub	Version control and collaboration
Automated Testing	JUnit 5	Unit testing framework
Build Management	Maven	Build automation and dependency management
Continuous Integration	Jenkins	CI/CD pipeline orchestration
Webhook Tunneling	ngrok	GitHub-Jenkins webhook connectivity
Containerization	Docker	Application containerization
Container Registry	Docker Hub	Image storage and distribution
Configuration Management	Ansible	Automated deployment

4 Pipeline Architecture

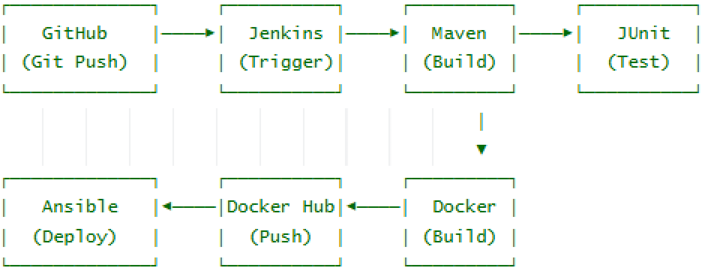


Figure 2: Pipeline Architecture Overview

Pipeline Workflow:

1. Developer pushes code to GitHub repository
2. GitHub webhook triggers Jenkins via ngrok tunnel

3. Jenkins pipeline executes via Jenkinsfile (provides stage view)
4. Jenkins pulls latest code from GitHub repository
5. Maven runs automated test cases using JUnit
6. Docker builds container image from Dockerfile
7. Jenkins logs into Docker Hub using credentials
8. Docker image is pushed to Docker Hub registry
9. Ansible deploys container on local system
10. Email notification sent about pipeline success/failure

5 Implementation Details

5.1 Source Control Management (Git + GitHub)

5.1.1 Setup

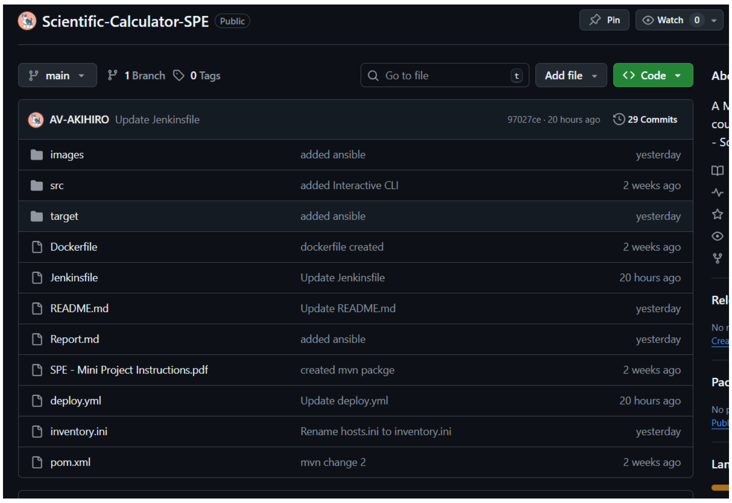
```
# Initialize local Git repository
git init

# Add remote repository
git remote add origin https://github.com/AV-AKIHIRO/Scientific-Calculator-SPE.git

# Stage and commit files
git add .
git commit -m "Initial commit: Calculator implementation"

# Push to remote
git push -u origin main
```

Listing 1: Git Setup Commands



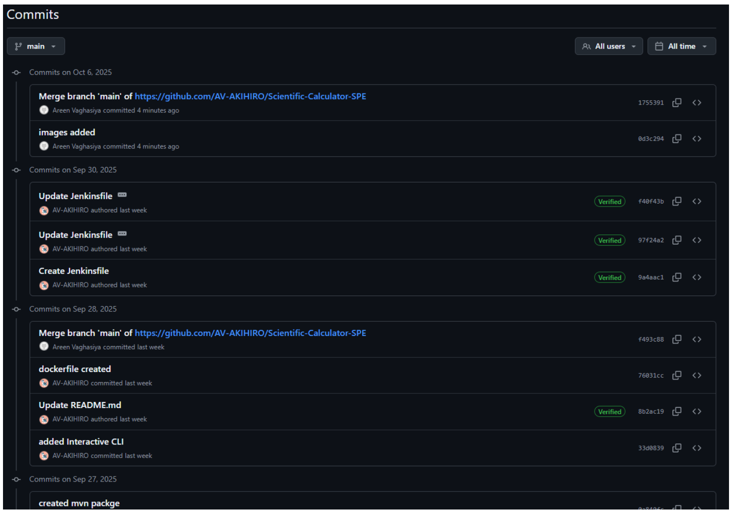
AV-AKIHIRO Update Jenkinsfile 97027ce · 20 hours ago 29 Commits

images	added ansible	yesterday
src	added Interactive CLI	2 weeks ago
target	added ansible	yesterday
Dockerfile	dockerfile created	2 weeks ago
Jenkinsfile	Update Jenkinsfile	20 hours ago
README.md	Update README.md	yesterday
Report.md	added ansible	yesterday
SPE - Mini Project Instructions.pdf	created mvn packge	2 weeks ago
deploy.yml	Update deploy.yml	20 hours ago
inventory.ini	Rename hosts.ini to inventory.ini	yesterday
pom.xml	mvn change 2	2 weeks ago

Figure 3: Project structure in GitHub repository

5.1.2 Commit History

Regular commits were made throughout the project development to maintain a clear history of changes.



Commits on Oct 6, 2025

Merge branch 'main' of https://github.com/AV-AKIHIRO/Scientific-Calculator-SPE
Areen Vaghasiya committed 4 minutes ago 1755391

images added
Areen Vaghasiya committed 4 minutes ago 0d3c294

Commits on Sep 30, 2025

Update Jenkinsfile
AV-AKIHIRO authored last week Verified f40f43b

Update Jenkinsfile
AV-AKIHIRO authored last week Verified 97f24a2

Create Jenkinsfile
AV-AKIHIRO authored last week Verified 9a4aac1

Commits on Sep 28, 2025

Merge branch 'main' of https://github.com/AV-AKIHIRO/Scientific-Calculator-SPE
Areen Vaghasiya committed last week f493c88

dockerfile created
AV-AKIHIRO committed last week 76031cc

Update README.md
AV-AKIHIRO authored last week Verified 8b2ac19

added Interactive CLI
AV-AKIHIRO committed last week 33d0839

Commits on Sep 27, 2025

created mvn packge

Figure 4: Commit history showing project progression

5.2 Testing (JUnit 5)

5.2.1 Configuration

JUnit 5 dependency in `pom.xml`:

```xml
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.9.2</version>
    <scope>test</scope>
</dependency>
```

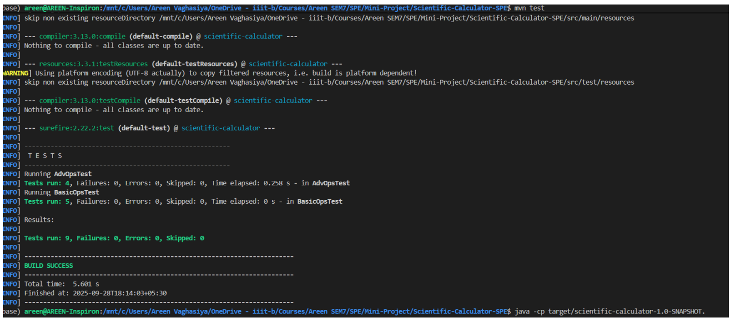
Listing 2: JUnit 5 Dependency in `pom.xml`

Test classes were created in the `src/test/java` directory to verify calculator operations including square root, factorial, logarithm, and power functions.

5.2.2 Test Execution

```
mvn test
```

Listing 3: Maven Test Command



```
ase) areen@AREEN-Inspiron:/mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE$ mvn test
NFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/main/resources
NFO]
NFO] --- compiler:3.13.0:compile (default-compile) @ scientific-calculator ---
NFO] Nothing to compile - all classes are up to date.
NFO]
NFO] --- resources:3.3.1:testResources (default-testResources) @ scientific-calculator ---
ARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
NFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/test/resources
NFO]
NFO] --- compiler:3.13.0:testCompile (default-testCompile) @ scientific-calculator ---
NFO] Nothing to compile - all classes are up to date.
NFO]
NFO] --- surefire:2.22.2:test (default-test) @ scientific-calculator ---
NFO]
NFO] -------------------------------------------------------
NFO]  T E S T S
NFO] -------------------------------------------------------
NFO] Running AdvOpsTest
NFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.258 s - in AdvOpsTest
NFO] Running BasicOpsTest
NFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in BasicOpsTest
NFO]
NFO] Results:
NFO]
NFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
NFO]
NFO] ------------------------------------------------------------------------
NFO] BUILD SUCCESS
NFO] ------------------------------------------------------------------------
NFO] Total time:  5.601 s
NFO] Finished at: 2025-09-28T18:14:03+05:30
NFO] ------------------------------------------------------------------------
ase) areen@AREEN-Inspiron:/mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE$ java -cp target/scientific-calculator-1.0-SNAPSHOT.
```

Figure 5: Successful test execution

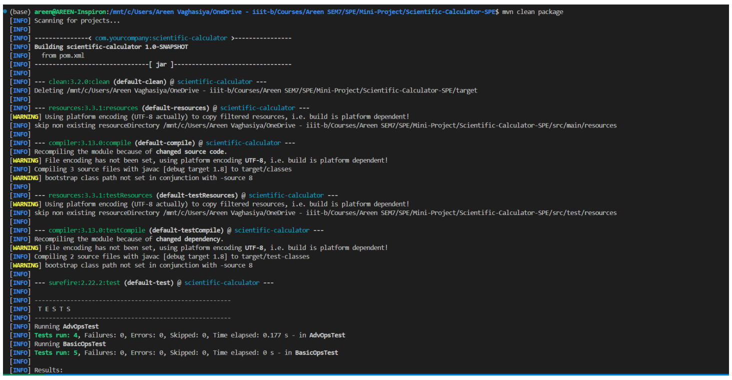
5.3 Build Tool (Maven)

Maven follows a standard directory layout with `src/main/java` for source code, `src/test/java` for tests, and `target/` for build output. The `pom.xml` file defines project metadata, dependencies, and build configurations.

5.3.1 Build Process

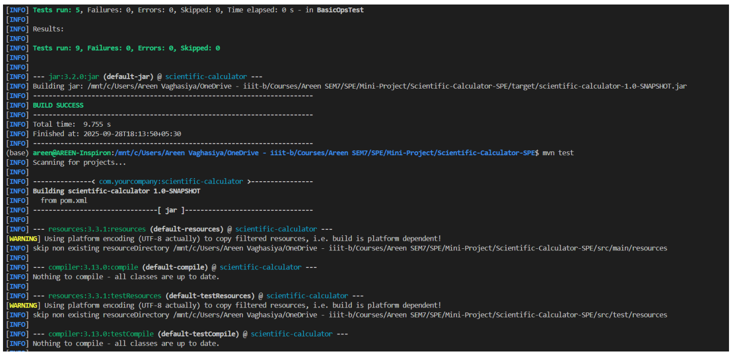
```
mvn clean package
```

Listing 4: Maven Clean and Package Command



```
(base) areen@AREEN-Inspiron:/mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE$ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] --------------< com.yourcompany:scientific-calculator >---------------
[INFO] Building scientific-calculator 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ scientific-calculator ---
[INFO] Deleting /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ scientific-calculator ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ scientific-calculator ---
[INFO] Recompiling the module because of changed source code.
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 3 source files with javac [debug target 1.8] to target/classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ scientific-calculator ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ scientific-calculator ---
[INFO] Recompiling the module because of changed dependency.
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 2 source files with javac [debug target 1.8] to target/test-classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[INFO]
[INFO] --- surefire:2.22.2:test (default-test) @ scientific-calculator ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running AdvOpsTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.177 s - in AdvOpsTest
[INFO] Running BasicOpsTest
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in BasicOpsTest
[INFO]
[INFO] Results:
```

Figure 6: Maven build process - compilation and testing



```
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in BasicOpsTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.2.0:jar (default-jar) @ scientific-calculator ---
[INFO] Building jar: /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/target/scientific-calculator-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.755 s
[INFO] Finished at: 2025-09-28T18:13:50+05:30
[INFO] ------------------------------------------------------------------------
(base) areen@AREEN-Inspiron:/mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE$ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] --------------< com.yourcompany:scientific-calculator >---------------
[INFO] Building scientific-calculator 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ scientific-calculator ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ scientific-calculator ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ scientific-calculator ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /mnt/c/Users/Areen Vaghasiya/OneDrive - iiit-b/Courses/Areen SEM7/SPE/Mini-Project/Scientific-Calculator-SPE/src/test/resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ scientific-calculator ---
[INFO] Nothing to compile - all classes are up to date.
```

Figure 7: Maven build process - packaging and artifact creation

5.4 Continuous Integration (Jenkins)

Jenkins automates the entire build, test, and deployment pipeline, triggered by code changes. It provides automation, extensibility, flexibility, scalability, and strong community support.

5.4.1 Required Plugins

The following Jenkins plugins were installed:

- Git Plugin
- Maven Integration Plugin
- Docker Pipeline Plugin
- Ansible Plugin
- GitHub Integration Plugin
- Email Extension Plugin

5.4.2 Global Tool Configuration

Tools configured in Jenkins:

- **JDK:** JDK 17
- **Maven:** Maven 3.9.3
- **Git:** Git executable
- **Docker:** Docker installation
- **Ansible:** Ansible executable

5.4.3 Credentials Configuration

Required credentials added:

- GitHub credentials (personal access token)
- Docker Hub credentials (ID: `dockerhub-credentials`)
- Email credentials (for notifications)

5.4.4 Jenkins URL Configuration

Jenkins URL set to: `https://katabolically-recrudescent-gigi.ngrok-free.dev/`

This ngrok URL enables GitHub to communicate with the locally hosted Jenkins instance through a secure tunnel.

5.4.5 GitHub Webhook Configuration

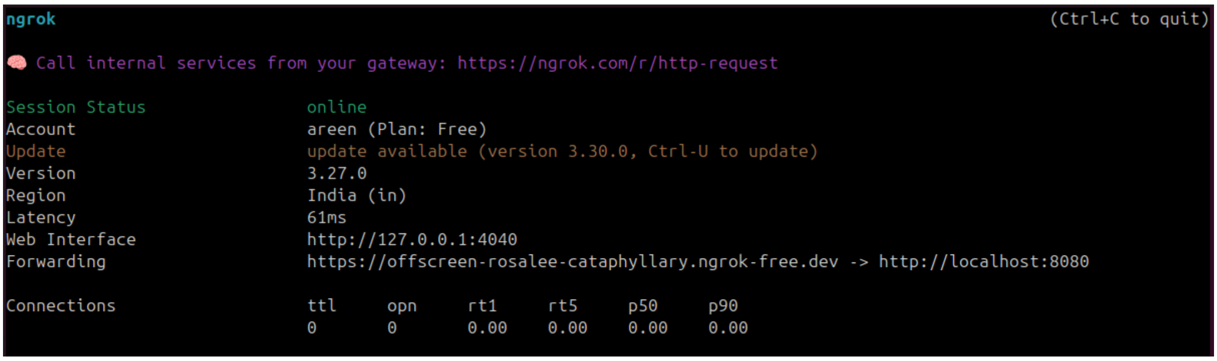
Setting up ngrok Tunnel:

```
# Start ngrok tunnel on port 8080 (Jenkins default port)
ngrok http --url="https://katabolically-recrudescent-gigi.ngrok-free.dev" 8080
```

Listing 5: Starting ngrok Tunnel

Configuring GitHub Webhook:

1. Navigate to GitHub repository: `SciCalc`
2. Go to **Settings** → **Webhooks** → **Add webhook**
3. Configure:
   - **Payload URL:** `https://katabolically-...dev/github-webhook/`
   - **Content type:** `application/json`
   - **Events:** Just the push event
   - **Secret:** the key from personal access token that was used to connect the webhook
   - **Active:** Checked



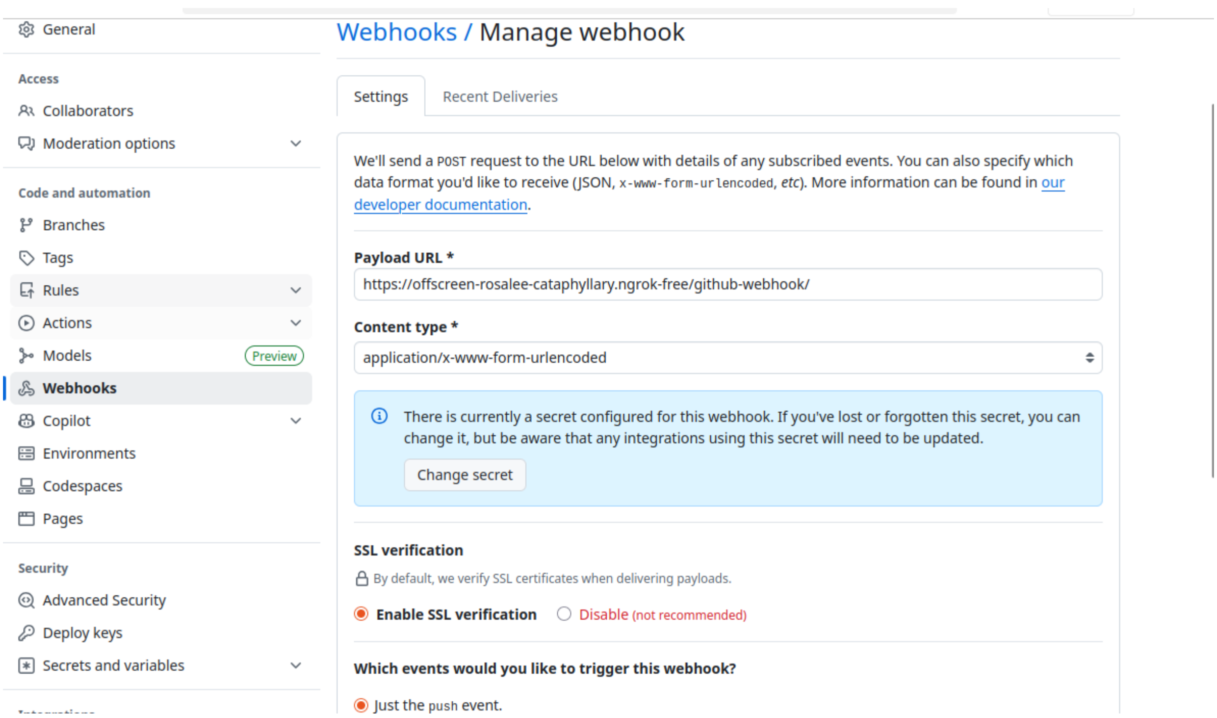
```
ngrok                                                                                  (Ctrl+C to quit)

🧠 Call internal services from your gateway: https://ngrok.com/r/http-request

Session Status                online
Account                       areen (Plan: Free)
Update                        update available (version 3.30.0, Ctrl-U to update)
Version                       3.27.0
Region                        India (in)
Latency                       61ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://offscreen-rosalee-cataphyllary.ngrok-free.dev -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 8: Ngrok static URL for this project



We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *
https://offscreen-rosalee-cataphyllary.ngrok-free/github-webhook/

Content type *
application/x-www-form-urlencoded

There is currently a secret configured for this webhook. If you've lost or forgotten this secret, you can change it, but be aware that any integrations using this secret will need to be updated.

SSL verification
By default, we verify SSL certificates when delivering payloads.
● Enable SSL verification ○ Disable (not recommended)

Which events would you like to trigger this webhook?
● Just the push event.

Figure 9: Webhook setup in GitHub repository settings

Workflow:

1. Developer pushes code to GitHub
2. GitHub sends POST request to ngrok webhook URL
3. ngrok tunnels request to Jenkins (port 8080)
4. Jenkins triggers pipeline from Jenkinsfile
5. Pipeline stages execute sequentially
6. Email notification sent on completion

5.4.6 Pipeline Job Configuration

1. Create new Pipeline job: `SciCalc Pipeline`
2. Configure:
   - Check **GitHub project** with repository URL
   - Enable **GitHub hook trigger for GITScm polling**
   - Set **Pipeline script from SCM**
   - Select **Git** as SCM with repository URL and credentials
   - Branch: `*/main`
   - Script Path: `Jenkinsfile`

5.4.7 Jenkinsfile

You can view the jenkinsfile from the github repo.

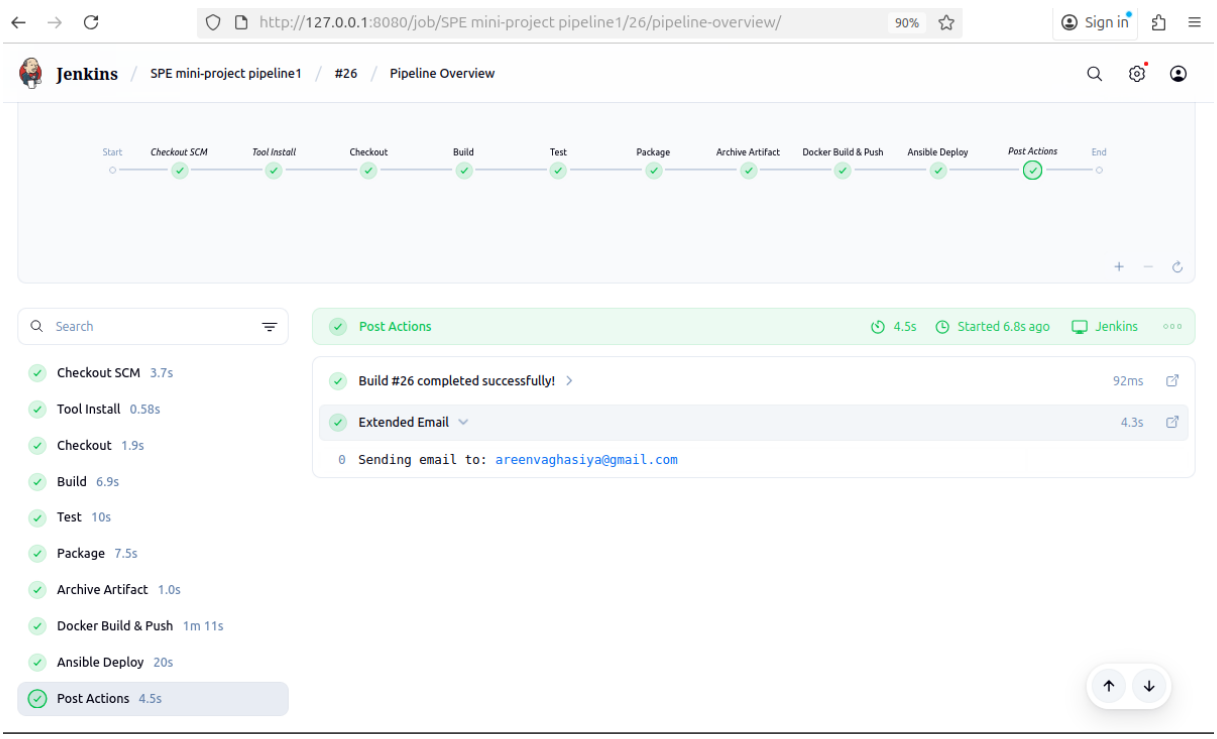
Explanation

The Jenkinsfile defines a declarative CI/CD pipeline that automates the build, test, packaging, containerization, and deployment of the `SciCalc` application. The pipeline is structured into sequential stages:

- **Checkout:** Retrieves the latest source code from the GitHub repository (`main` branch).
- **Build:** Compiles the Java project using Maven to ensure the codebase is syntactically correct and ready for testing.
- **Test:** Executes unit test cases with Maven to verify the functionality and correctness of the source code.
- **Package:** Packages the compiled code into a distributable `.jar` file.
- **Archive Artifact:** Archives the generated artifacts within Jenkins for traceability and reuse in future stages.
- **Build Docker Image:** Creates a Docker image for the application tagged with both the current build number and the `latest` tag.
- **Push to Docker Hub:** Authenticates with Docker Hub using stored credentials and pushes both image tags to the repository.
- **Deploy with Ansible:** Uses an Ansible playbook to deploy the latest Docker image to the target environment specified in the inventory file.

Post-build actions are handled using the `post` block:

- On **success**, a notification email is sent summarizing the build details, image tags, and completion message.
- On **failure**, an alert email is sent to notify the developer about the failure and prompt log inspection.



http://127.0.0.1:8080/job/SPE mini-project pipeline1/26/pipeline-overview/
90%
Sign in
Jenkins / SPE mini-project pipeline1 / #26 / Pipeline Overview
Start
Checkout SCM
Tool Install
Checkout
Build
Test
Package
Archive Artifact
Docker Build & Push
Ansible Deploy
Post Actions
End
Search

Checkout SCM 3.7s
Tool Install 0.58s
Checkout 1.9s
Build 6.9s
Test 10s
Package 7.5s
Archive Artifact 1.0s
Docker Build & Push 1m 11s
Ansible Deploy 20s
Post Actions 4.5s

Post Actions 4.5s Started 6.8s ago Jenkins

Build #26 completed successfully! 92ms

Extended Email 4.3s

```
0 Sending email to: areenvaghasiya@gmail.com
```

Figure 10: Jenkins pipeline execution showing all stages in Stage View

The Jenkinsfile creates a visual Stage View in Jenkins UI, with each stage color-coded (green for success, red for failure) and displaying execution time.

5.5 Containerization (Docker)

Docker packages the application with all dependencies, ensuring consistent behavior across environments. Benefits include portability, isolation, efficiency, scalability, version control, and consistency.

5.5.1 Dockerfile

```
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY target/scientific-calculator-1.0-SNAPSHOT.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]
```

Listing 6: Dockerfile for Scientific Calculator

5.5.2 Build and Run

```
# Build image
docker build -t areen9295/calc-app2:latest .

# Run container interactively
docker run -it --rm areen9295/calc-app2:latest

# Run in detached mode
```

```
docker run -d --name calc-container areen9295/calc-app2:latest
```

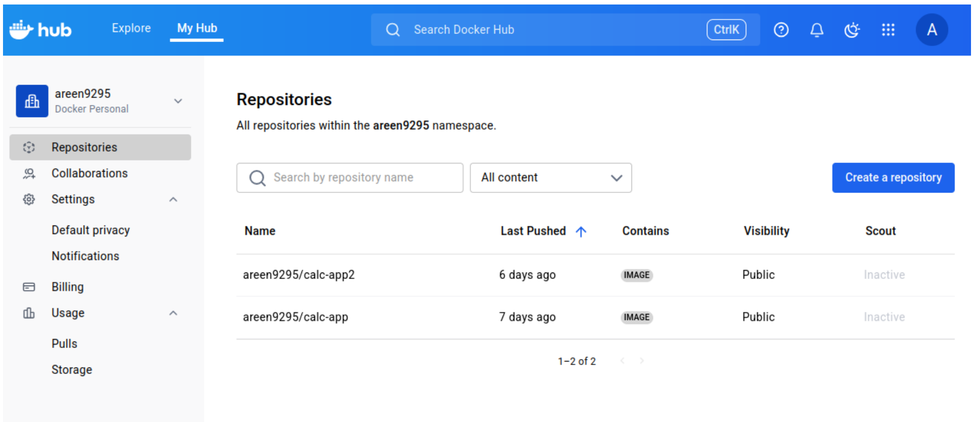
Listing 7: Docker Build and Run Commands

5.6 Docker Hub Repository

5.6.1 Push to Docker Hub

```
docker login
docker push areen9295/calc-app2:latest
```

Listing 8: Docker Login and Push



Repositories

All repositories within the **areen9295** namespace.

Name	Last Pushed ↑	Contains	Visibility	Scout
areen9295/calc-app2	6 days ago	IMAGE	Public	Inactive
areen9295/calc-app	7 days ago	IMAGE	Public	Inactive

Figure 11: Docker Hub repository showing multiple image versions

Repository: https://hub.docker.com/r/areen9295/calc-app2

5.7 Continuous Deployment (Ansible)

Ansible automates application deployment using YAML-based playbooks and operates in an agentless manner.

5.7.1 Project Structure

```
Scientific-Calculator-SPE/
|-- inventory.ini        (Ansible inventory)
|-- deploy.yml           (Ansible playbook)
```

Listing 9: Ansible Files

5.7.2 Inventory File

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

Listing 10: inventory.ini

5.7.3 Deployment Playbook

You can view the deploy.yml file on github repo.

5.7.4 Explanation

This Ansible playbook automates the deployment of the SciCalc application as a Docker container on the local host. It is executed automatically from the Jenkins pipeline during the Deploy with Ansible stage.

- **Play Definition:** The play is configured to run on the localhost using a local connection without privilege escalation (become: no), as the deployment occurs on the same system where Jenkins runs.
- **Check Docker Python Module:** Before performing Docker operations, the playbook verifies whether the python3-docker module is available. This is essential because Ansible's community.docker collection relies on this module to interact with Docker.
- **Fail-Safe Check:** If the Python Docker module is missing, the playbook fails gracefully with a clear error message instructing the user to install the dependency manually. This ensures early detection of missing runtime requirements.
- **Deploy Container:** The main task uses the community.docker.docker_container module to deploy and manage the scientific-calculator container:
  - Pulls the latest image (areen9295/calc-app2:latest) from Docker Hub.
  - Ensures the container is always running (state: started) and recreates it if configuration changes occur.
  - Maps container port 80 to host port 9000 for web access.
  - Sets the restart_policy to always to ensure automatic recovery if the container stops unexpectedly.

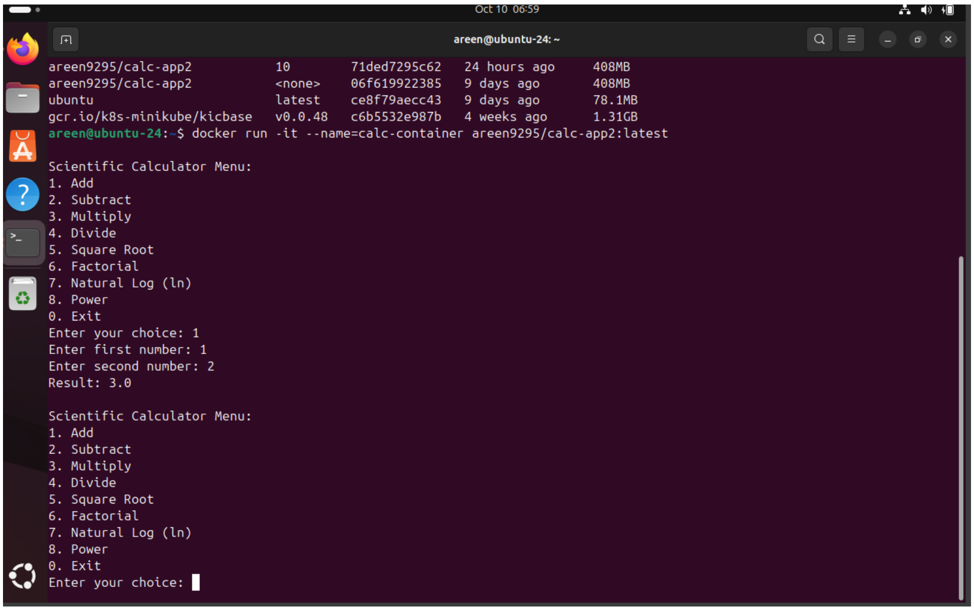
5.7.5 Execution

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

Listing 11: Ansible Playbook Execution

6 Application Output

The Scientific Calculator is a terminal based application written in java. Below are the attached images:



```
areen9295/calc-app2           10        71ded7295c62   24 hours ago     408MB
areen9295/calc-app2           <none>    06f619922385   9 days ago       408MB
ubuntu                        latest    ce8f79aecc43   9 days ago       78.1MB
gcr.io/k8s-minikube/kicbase   v0.0.48   c6b5532e987b   4 weeks ago      1.31GB
areen@ubuntu-24:~$ docker run -it --name=calc-container areen9295/calc-app2:latest

Scientific Calculator Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Square Root
6. Factorial
7. Natural Log (ln)
8. Power
0. Exit
Enter your choice: 1
Enter first number: 1
Enter second number: 2
Result: 3.0

Scientific Calculator Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Square Root
6. Factorial
7. Natural Log (ln)
8. Power
0. Exit
Enter your choice: █
```

Figure 12: How app looks like in terminal

7 Challenges and Solutions

7.1 Docker Permission Issues

Problem: Jenkins user lacked Docker daemon permissions.

Solution:

```
sudo usermod -aG docker jenkins
sudo systemctl restart jenkins
```

Listing 12: Fix Docker Permissions

7.2 Ansible SSH Connection

Problem: SSH authentication failures for localhost.

Solution: Used `ansible_connection=local` in inventory file.

7.3 Maven Dependency Resolution

Problem: Corrupted repository cache.

Solution:

```
mvn clean install -U
```

Listing 13: Force Dependency Update

7.4 ngrok URL Changes

Problem: Free tier ngrok generates new URLs on restart.

Solution: Consider using the one static subdomain on free tier.