

Software Production Engineering

Mini Project 1

Name: Sparsh Salodkar

Roll Number: IMT2022113

Scientific Calculator

CI/CD Pipeline using Docker, Jenkins, and Ansible

Introduction

This report walks through a simple, reproducible CI/CD setup for a **Scientific Calculator** (CLI) application. The pipeline uses GitHub for source control, Jenkins for automation, Docker for packaging, and Ansible for local deployment. The idea is straightforward: *push code → Jenkins builds & tests → image pushed to Docker Hub → Ansible pulls the latest image and deploys.*

What and Why of DevOps

DevOps combines development and operations to shorten feedback loops and make releases swift. In this project:

- Every push runs the same build and tests on Jenkins.
- The app is containerized, so it runs the same everywhere.
- Deployment is scripted with Ansible, removing manual steps.

Tools Used

GitHub (SCM & webhooks), **Jenkins** (CI/CD), **Docker** (container image), **Ansible** (pull & run latest image), **Ngrok** (expose local Jenkins to GitHub for webhooks).

1. Source Control: Git & GitHub

The repository holds the Java source, tests, Dockerfile, Jenkinsfile, ansible/ play-book, and a small `RUN.sh` helper to run the image locally.

Webhook (GitHub → Jenkins)

In the repository settings, a webhook points to Jenkins' GitHub endpoint (via your active Ngrok URL). On every push, GitHub pings Jenkins, which triggers the pipeline on the latest commit.

Screenshots

```
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   thescicalc/pom.xml
    new file:   thescicalc/src/main/java/scicalc/App/App.java
    new file:   thescicalc/src/main/java/scicalc/App/Calculator.java
    new file:   thescicalc/src/main/java/scicalc/Main.java
    new file:   thescicalc/src/main/java/scicalc/operations/BasicOperations.java
    new file:   thescicalc/src/main/java/scicalc/operations/ScientificOperations.java
    new file:   thescicalc/src/test/java/BasicOperationsTests.java
    new file:   thescicalc/src/test/java/ScientificOperationsTests.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ../Report/
    thescicalc/target/

(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git commit -m "Created the App for calculator and the tests to handle basic and scientific operations"
[main d9d4dbf] Created the App for calculator and the tests to handle basic and scientific operations
 8 files changed, 501 insertions(+)
 create mode 100644 thescicalc/pom.xml
 create mode 100644 thescicalc/src/main/java/scicalc/App/App.java
 create mode 100644 thescicalc/src/main/java/scicalc/App/Calculator.java
 create mode 100644 thescicalc/src/main/java/scicalc/Main.java
 create mode 100644 thescicalc/src/main/java/scicalc/operations/BasicOperations.java
 create mode 100644 thescicalc/src/main/java/scicalc/operations/ScientificOperations.java
 create mode 100644 thescicalc/src/test/java/BasicOperationsTests.java
 create mode 100644 thescicalc/src/test/java/ScientificOperationsTests.java
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git push -u origin main
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 4.43 KiB | 1.48 MiB/s, done.
Total 20 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/SparshGHUB/Sc.git
   0e29376..d9d4dbf  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ../Report/
    thescicalc/Dockerfile
    thescicalc/README.md
    thescicalc/target/

nothing added to commit but untracked files present (use "git add" to track)
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git add thescicalc/Dockerfile
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/SciCalC$ git commit -m "Created the Dockerfile for the project"
[main 77bf931] Created the Dockerfile for the project
```

Figure 1: Repository creation and initial push.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/SparshGHub/Scientific-Calculator/

Credentials ?

sparshGHub/*****

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Figure 2: Jenkins job connected to the GitHub repo (credentials and URL).

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

Session Status      online
Account             sparsh salodkar (Plan: Free)
Update              update available (version 3.30.0, Ctrl-U to update)
Version             3.29.0
Region              India (in)
Latency              41ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://nonopinionated-unreducible-alica.ngrok-free.dev -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
                    88     0      0.00   0.00   0.65   42.32

HTTP Requests
-----
02:32:27.536 IST GET /job/Scientific Calculator/40/statusIcon 200 OK
02:32:26.205 IST GET /job/Scientific Calculator/childrenContextMenu 200 OK
02:32:26.205 IST GET /job/Scientific Calculator/contextMenu 200 OK
02:32:22.507 IST GET /$stapler/bound/script/$stapler/bound/ed9f07d6-23bd-4dfa-b062-ce57e56698ab 200 OK
02:32:22.097 IST GET /job/Scientific Calculator/40/pipeline-overview 302 Found
02:32:22.736 IST GET /plugin/pipeline-graph-view/js/bundles/assets/PipelineConsole-BUubu70L.js 304 Not Modified
02:32:22.324 IST GET /github-webhook/theme-dark/theme.css 404 Not Found
02:32:22.577 IST GET /plugin/pipeline-graph-view/js/build.js 304 Not Modified
02:32:22.828 IST GET /i18n/resourceBundle 200 OK
02:32:22.737 IST GET /plugin/pipeline-graph-view/js/bundles/assets/PipelineGraph-DWQowfQ4.js 304 Not Modified
```

Figure 3: Ngrok exposing local Jenkins for GitHub webhooks.

```

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:
9

Exiting...

Thank you for using SciCalc!
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git add Jenkinsfile
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git commit -m "Changed the jenkins file to include testing also"
[main 3a2aadc] changed the jenkins file to include testing also
1 file changed, 0 insertions(+), 0 deletions(-)
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 470 bytes | 470.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote: This repository moved. Please use the new location:
remote: https://github.com/SparshHub/scientific-calculator.git
To https://github.com/SparshHub/Sc.git
09fe2df..3a2aadc main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ echo "Another Trigger After fixing the Ansible deploy file" >> README.md
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git add README.md
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git commit -m "Test Trigger"
[main 7e87c9f] Test Trigger
1 file changed, 1 insertion(+)
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 335 bytes | 335.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote: https://github.com/SparshHub/Scientific-Calculator.git
To https://github.com/SparshHub/Sc.git
3a2aadc..7e87c9f main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) sparsh@garrison:~/Desktop/CDL/SEM 7/Software Production Engineering/Mini Project $

```

Figure 4: Test Trigger

2. Jenkins CI Pipeline (Stage-by-stage)

The pipeline below lives at SciCalc/thescicalc/Jenkinsfile. It runs on any agent with Java, Maven, Docker, and Ansible available.

Pipeline (final version)

```

pipeline {
    agent any
    options { timestamps(); timeout(time: 60, unit: 'MINUTES') }
    environment {
        DOCKER_IMAGE = "sparshdockerman/scicalc"
        CREDENTIALS_ID = "dockerHubCreds"
    }
    stages {
        stage('Checkout') { steps { checkout scm } }
        stage('Verify Tools') {
            steps {
                sh 'java -version || true'
                sh 'mvn -v || true'
                sh 'docker --version || true'
                sh 'ansible --version || true'
            }
        }
        stage('Test') {

```

```

    steps { dir('SciCalC/thescicalc') { sh 'mvn -B test' } }
    post { always { junit 'SciCalC/thescicalc/target/surefire-reports
        /*.xml' } }
}
stage('Package') {
    steps { dir('SciCalC/thescicalc') { sh 'mvn -B package -
        DSkipTests' } }
    post { success { archiveArtifacts artifacts: 'SciCalC/thescicalc/
        target/*.jar', fingerprint: true } }
}
stage('Docker Build & Push') {
    steps {
        script {
            def sha = env.GIT_COMMIT ?: sh(script: "git rev-parse --short
                HEAD", returnStdout: true).trim()
            def imgLatest = "${DOCKER_IMAGE}:latest"
            def imgSha     = "${DOCKER_IMAGE}:${sha}"
            sh "docker build -f SciCalC/thescicalc/Dockerfile -t ${
                imgLatest} -t ${imgSha} SciCalC/thescicalc"
            withCredentials([usernamePassword(credentialsId: env.
                CREDENTIALS_ID,
                                usernameVariable: '
                                DOCKERHUB_USER',
                                passwordVariable: '
                                DOCKERHUB_TOKEN')]) {
                sh '''
                    echo "$DOCKERHUB_TOKEN" | docker login -u "
                        $DOCKERHUB_USER" --password-stdin
                    docker push ${DOCKER_IMAGE}:latest
                    docker logout
                '''
            }
        }
    }
}
stage('Deploy with Ansible') {
    steps {
        dir('SciCalC/thescicalc/ansible') {
            withCredentials([usernamePassword(credentialsId: env.
                CREDENTIALS_ID,
                                usernameVariable: '
                                DOCKERHUB_USER',
                                passwordVariable: '
                                DOCKERHUB_TOKEN')]) {
                sh '''
                    ansible-playbook -i inventory deploy.yml \

```

```

--extra-vars "docker_image=${DOCKER_IMAGE}:latest
              docker_user=${DOCKERHUB_USER} docker_token=
              ${DOCKERHUB_TOKEN}"
    },
  }
}
}
}
}
post {
  success { echo 'Build and deployment successful.' }
  failure { echo 'Build or deployment failed. Check console output.' }
  always { echo 'Pipeline completed.' }
}
}

```

What each stage does

Checkout pulls the exact commit that triggered the job.

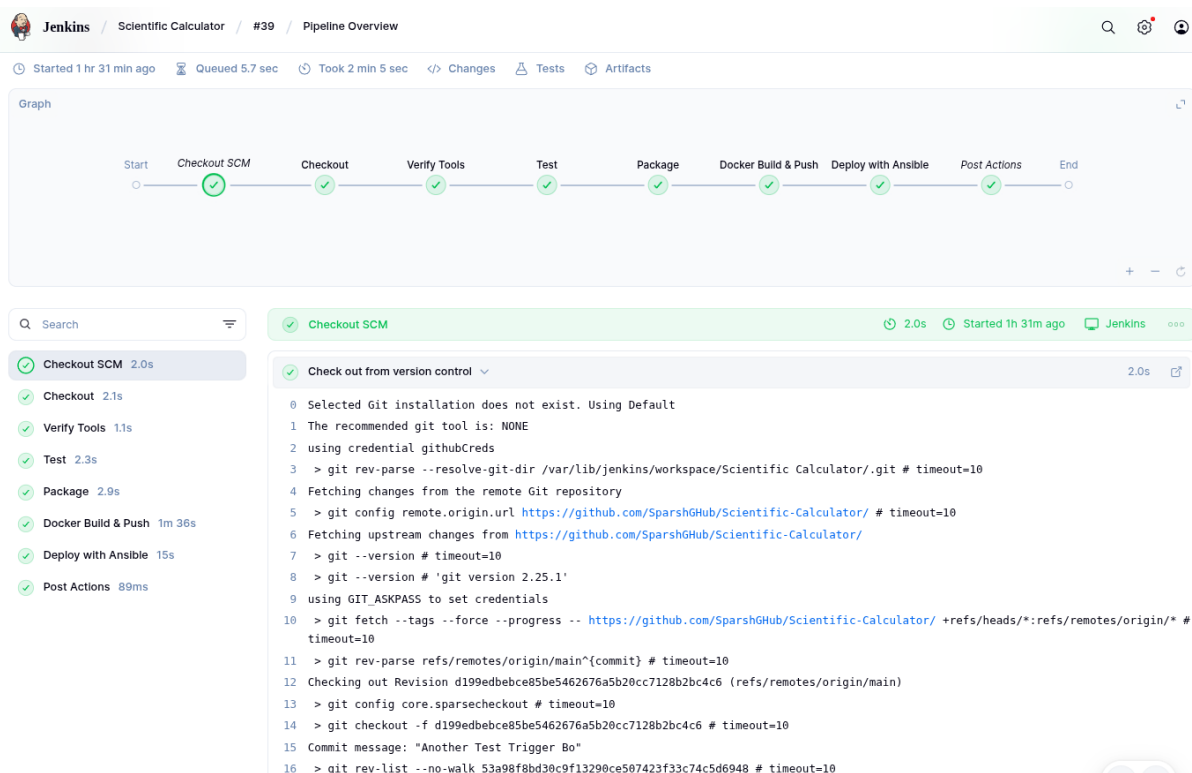


Figure 5: Checkout SCM

Verify Tools prints versions so you immediately see if a tool is missing or misconfig-

ured.

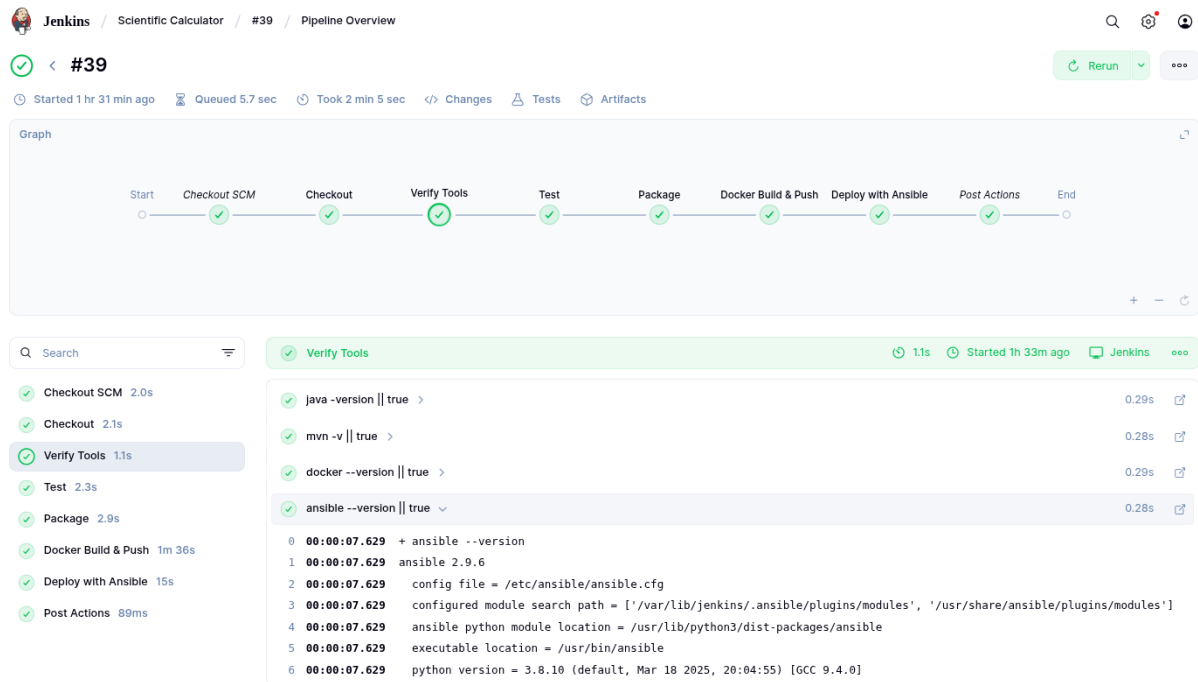


Figure 6: Verify Tools

Test runs all JUnit tests (Jenkins always publishes results from target/surefire-reports).

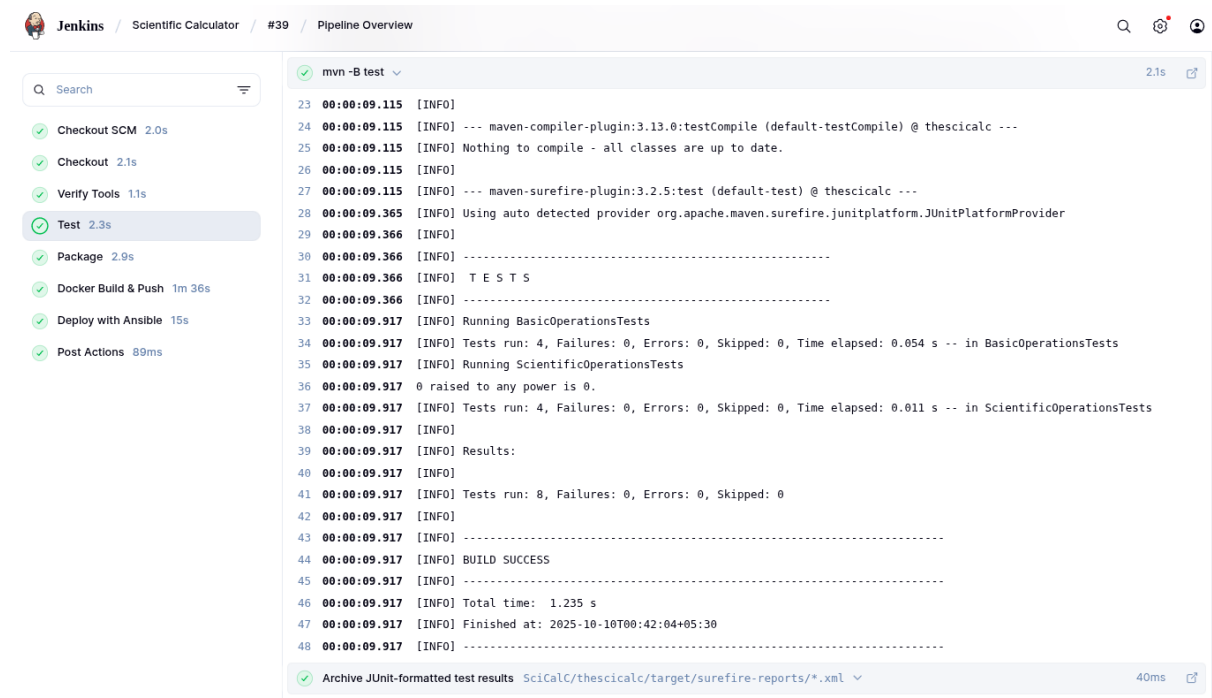


Figure 7: Test

Package builds the JAR (tests skipped here), and archives it as a build artifact.

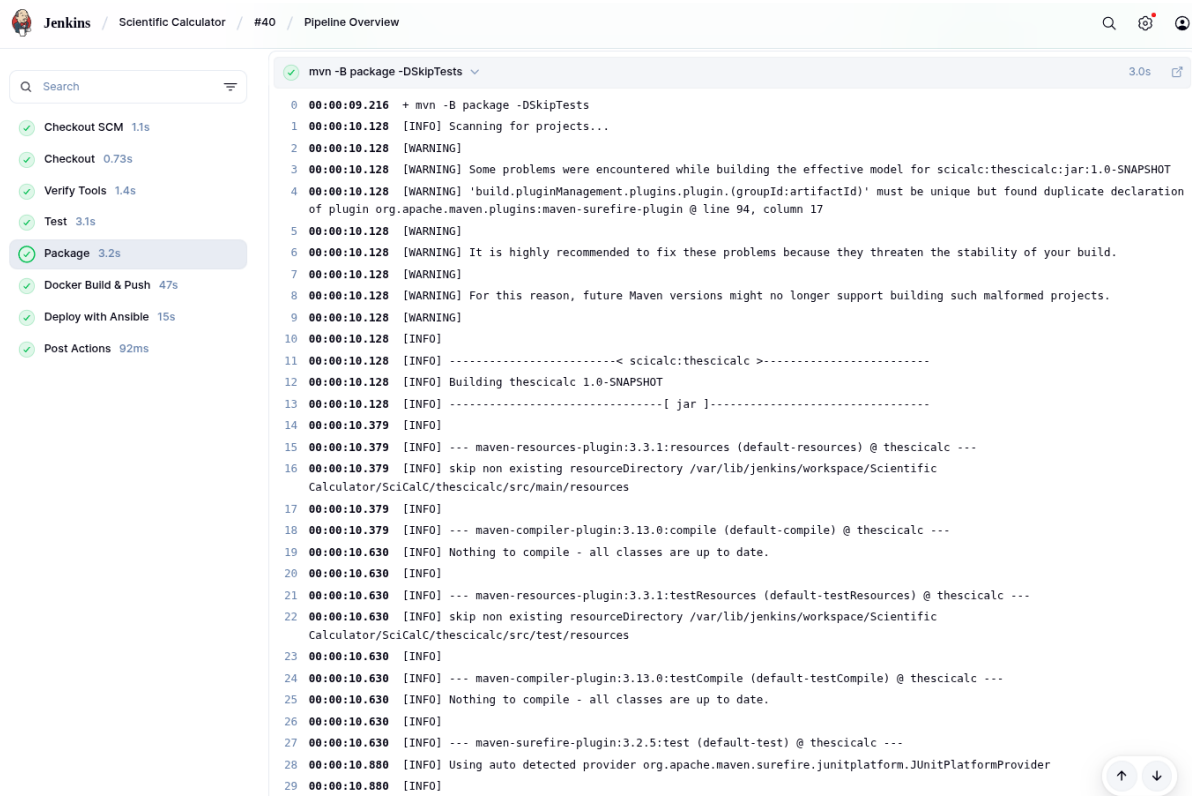


Figure 8: Package

Docker Build & Push builds a multi-tagged image (:latest and the commit SHA) and pushes :latest to Docker Hub using Jenkins credentials.

The screenshot shows the Jenkins Pipeline Overview for a job named 'Scientific Calculator' (ID #40). The pipeline consists of several stages: Checkout SCM (1.1s), Checkout (0.73s), Verify Tools (1.4s), Test (3.1s), Package (3.2s), Docker Build & Push (47s), Deploy with Ansible (15s), and Post Actions (92ms). The 'Docker Build & Push' stage is currently selected and expanded, showing a list of build steps with their durations and status. The steps include: 'docker build -f SciCalC/thescicalc/Dockerfile -t sparshdockerman/scicalc:latest -t sparshdockerman/scicalc:e1c3af03022b...' (22s), 'echo "\$DOCKERHUB_TOKEN" | docker login -u "\$DOCKERHUB_USER" --password-stdin docker push \$(DOCKER_IMAGE):lat...' (24s), and a series of 'Preparing' and 'Pushed' messages for various layers. The 'Docker Build & Push' stage is marked as successful with a green checkmark.

Jenkins / Scientific Calculator / #40 / Pipeline Overview

Search

Checkout SCM 1.1s

Checkout 0.73s

Verify Tools 1.4s

Test 3.1s

Package 3.2s

Docker Build & Push 47s

Deploy with Ansible 15s

Post Actions 92ms

docker build -f SciCalC/thescicalc/Dockerfile -t sparshdockerman/scicalc:latest -t sparshdockerman/scicalc:e1c3af03022b... 22s

echo "\$DOCKERHUB_TOKEN" | docker login -u "\$DOCKERHUB_USER" --password-stdin docker push \$(DOCKER_IMAGE):lat... 24s

00:00:35.008 + echo ****

00:00:35.008 + docker login -u sparshdockerman --password-stdin

00:00:43.034

00:00:43.034 WARNING! Your credentials are stored unencrypted in '/var/lib/jenkins/.docker/config.json'.
Configure a credential helper to remove this warning. See
<https://docs.docker.com/go/credential-store/>

00:00:43.034 Login Succeeded

00:00:43.034 + docker push sparshdockerman/scicalc:latest

00:00:43.034 The push refers to repository [docker.io/sparshdockerman/scicalc]

00:00:43.034 597b0bc9bd56: Preparing

00:00:43.034 0a6ba769bad2: Preparing

00:00:43.034 404b7f04e63d: Preparing

00:00:43.034 f3247191fe92: Preparing

00:00:43.034 6e771b1c41af: Preparing

00:00:43.034 2b35671596c3: Preparing

00:00:43.034 65b08cd99c60: Preparing

00:00:43.034 2b35671596c3: Waiting

00:00:44.378 404b7f04e63d: Layer already exists

00:00:44.378 6e771b1c41af: Layer already exists

00:00:49.574 65b08cd99c60: Layer already exists

00:00:49.574 0a6ba769bad2: Layer already exists

00:00:49.574 f3247191fe92: Layer already exists

00:00:49.574 2b35671596c3: Layer already exists

00:00:49.574 597b0bc9bd56: Pushed

00:00:59.451 latest: digest: sha256:8873cd4835d7211f66911014f02d7055ac47973b54d7a0225dd61cc42e8c38b7 size: 1782

00:00:59.451 + docker logout

00:00:59.451 Removing login credentials for <https://index.docker.io/v1/>

Figure 9: Docker Build and Push

Deploy with Ansible switches into the `ansible/` folder and runs the playbook that pulls `:latest` and restarts the local container.

The screenshot shows the Jenkins Pipeline Overview for a project named 'Scientific Calculator'. The pipeline consists of several steps: Checkout SCM (1.1s), Checkout (0.73s), Verify Tools (1.4s), Test (3.1s), Package (3.2s), Docker Build & Push (47s), and Deploy with Ansible (15s). The 'Deploy with Ansible' step is currently selected and expanded, showing its execution details. The step is titled 'ansible-playbook -i inventory deploy.yml --extra-vars "docker_image=\${DOCKER_IMAGE}:latest docker_user=\${DOCKERHUB...}' and has a duration of 14s. The output shows the Ansible playbook running on localhost, gathering facts, checking if Docker is installed, installing Docker if not installed, ensuring Docker service is running, pulling the latest Docker image, and stopping and removing any container using host port.

```

0 00:01:00.041 + ansible-playbook -i inventory deploy.yml --extra-vars "docker_image=${DOCKER_IMAGE}:latest docker_user=${DOCKERHUB...
1 00:01:00.291
2 00:01:00.291 PLAY [Deploy Scientific Calculator Container] *****
3 00:01:00.291
4 00:01:00.291 TASK [Gathering Facts] *****
5 00:01:04.415 [WARNING]: Platform linux on host localhost is using the discovered Python
6 00:01:04.415 interpreter at /usr/bin/python, but future installation of another Python
7 00:01:04.415 interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
8 00:01:04.415 ce_appendices/interpreter_discovery.html for more information.
9 00:01:04.415 ok: [localhost]
10 00:01:04.415
11 00:01:04.415 TASK [Check if Docker is already installed] *****
12 00:01:04.415 changed: [localhost]
13 00:01:04.415
14 00:01:04.415 TASK [Install Docker if not installed] *****
15 00:01:04.415 skipping: [localhost]
16 00:01:04.665
17 00:01:04.665 TASK [Ensure Docker service is running] *****
18 00:01:05.216 ok: [localhost]
19 00:01:05.216
20 00:01:05.216 TASK [Pull latest Docker image] *****
21 00:01:13.242 changed: [localhost]
22 00:01:13.242
23 00:01:13.242 TASK [Stop and remove any container using host port] *****
24 00:01:13.493 changed: [localhost]
25 00:01:13.493
26 00:01:13.493 TASK [Stop and remove existing container if running (by name)] *****
27 00:01:13.743 changed: [localhost]

```

Figure 10: Deploy With Ansible

3. Docker Image (multi-stage)

Dockerfile (final)

```

# Stage 1: Build the JAR using Maven (Java 17)
FROM maven:3.9.6-eclipse-temurin-17 AS build
WORKDIR /app
COPY . .
RUN mvn clean package -DskipTests

# Stage 2: Run the application using OpenJDK (Java 17)
FROM eclipse-temurin:17-jre
WORKDIR /app
COPY --from=build /app/target/thescicalc-*.jar app.jar
ENTRYPOINT ["java", "-jar", "app.jar"]

```

```
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering$ cd Mini/ Project/ 1/
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1$ ls
docker.txt  Report  'SPE - Mini Project Instructions.pdf'
README.md  Scicalc
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1$ docker build -t sparshdockerman/scicalc:latest .
[*] Building 0.1s (1/1) FINISHED                                docker:default
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 2B
0.0s
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1$ ls
docker.txt  Report  'SPE - Mini Project Instructions.pdf'
README.md  Scicalc
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1$ cd Scicalc/thescicalc/
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/Scicalc/thescicalc$ docker build -t sparshdockerman/scicalc:latest .
[*] Building 17.0s (15/15) FINISHED                                docker:default
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 38B
-> [internal] load metadata for docker.io/library/eclipse-temurin:17-jre
-> [internal] load metadata for docker.io/library/maven:3.9.6-eclipse-temurin-17
-> [auth] library/eclipse-temurin:pull token for registry-1.docker.io
-> [auth] library/maven:pull token for registry-1.docker.io
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [build 1/4] FROM docker.io/library/maven:3.9.6-eclipse-temurin-17@sha256:29a1658b1f5078e07c2b1777b519b45eb477650928e807aac43abcc8f929d4
-> [internal] load build context
-> => transferring context: 3.82kB
-> [stage-1 1/4] FROM docker.io/library/eclipse-temurin:17-jre@sha256:d321dc467a2649472123e94f91829ab3e168aa76f88f6d1cc6e8a1f02afbd
-> => resolve docker.io/library/eclipse-temurin:17-jre@sha256:d321dc467a2649472123e94f91829ab3e168aa76f88f6d1cc6e8a1f02afbd
-> => sha256:d321dc467a2649472123e94f91829ab3e168aa76f88f6d1cc6e8a1f02afbd 8.48kB / 8.48kB
-> => sha256:c7f6d453a1228e71e80b05918b0f9a2024426283f4a36024c09513f0e9f6 159B / 159B
-> => sha256:80e8480445c7036f7e0d0970b1c115308521b154a39727cc3b0e0f0c 5.92kB / 5.92kB
-> => sha256:ala21c9bc1c121509d9370cd1c7643a081629b3b08aa4446020e091e1044 29.72MB / 29.72MB
-> => sha256:c218f8e12eacac59c3bfc362e3c0d25fdae298944f8f8f8c941d10051d16 19.58MB / 19.58MB
-> => sha256:5afe7cfaf1d33027190a433aa99e985f0bf30d59eeaa0585c912d080f 44.99MB / 44.99MB
-> => extracting sha256:ala21c9bc1c121509d9370cd1c7643a081629b3b08aa4446020e091e1044
1.7s
-> => sha256:c7f6d453a1228e71e80b05918b0f9a2024426283f4a36024c09513f0e9f6 159B / 159B
-> => sha256:c7f6d453a1228e71e80b05918b0f9a2024426283f4a36024c09513f0e9f6 2.20MB / 2.20MB
-> => extracting sha256:c218f8e12eacac59c3bfc362e3c0d25fdae298944f8f8f8c941d10051d16
1.1s
-> => extracting sha256:5afe7cfaf1d33027190a433aa99e985f0bf30d59eeaa0585c912d080f
2.8s
-> => extracting sha256:c7f6d453a1228e71e80b05918b0f9a2024426283f4a36024c09513f0e9f6
0.9s
-> => extracting sha256:c7f6d453a1228e71e80b05918b0f9a2024426283f4a36024c09513f0e9f6
0.9s
-> CACHED [build 2/4] WORKDIR /app
0.0s
-> CACHED [build 3/4] COPY . .
0.0s
-> CACHED [build 4/4] RUN rm -rf clean package -DskipTests
0.0s
-> [stage-1 2/5] WORKDIR /app
0.0s
-> [stage-1 3/3] COPY --from=build /app/target/thescicalc-*.jar app.jar
0.1s
-> exporting to image
0.0s
-> => writing image sha256:302b0f77f0d0a9888a29295a3c050f0eb03a3ff3c44ab04ee147b0442044
0.0s
-> naming to docker.io/sparshdockerman/scicalc:latest
0.0s
(base) sparsh@garrison:~/Desktop/COL/SEN 7/Software Production Engineering/Mini Project 1/Scicalc/thescicalc$
```

Figure 11: Docker Build

4. Ansible Deployment (local host)

The playbook below runs on localhost, pulls the latest image, stops any old container by name or port, and (for the CLI case) demonstrates an interactive run step in the pipeline context.

deploy.yml

```
---
- name: Deploy Scientific Calculator Container
  hosts: localhost
  connection: local
  become: yes

  vars:
    app_name: scicalc
    image_name: sparshdockerman/scicalc:latest
    container_port: 8080
    host_port: 9090

  tasks:
    - name: Check if Docker is already installed
      shell: "docker --version"
      register: docker_check
      ignore_errors: yes

    - name: Ensure Docker service is running
```

```
service:
  name: docker
  state: started
  enabled: true

- name: Pull latest Docker image
  shell: |
    docker pull {{ image_name }}

- name: Stop and remove any container using host port
  shell: |
    ids=$(docker ps --filter "publish={{ host_port }}" -q)
    if [ -n "$ids" ]; then
      docker stop $ids
      docker rm $ids
    fi

- name: Stop and remove existing container if running (by name)
  shell: |
    if [ "$(docker ps -q -f name={{ app_name }})" ]; then
      docker stop {{ app_name }}
    fi
    if [ "$(docker ps -aq -f name={{ app_name }})" ]; then
      docker rm {{ app_name }}
    fi
    fi

# For CLI demonstration (non-daemon)
- name: Run new CLI container interactively (pipeline demo)
  shell: |
    docker run -it --name {{ app_name }} --rm {{ image_name }} <<'
      EOF'
    echo "Scientific Calculator CLI started"
    exit
    EOF
  register: run_output
  ignore_errors: yes

- name: Verify running containers
  shell: "docker ps --filter name={{ app_name }}"
  register: container_status

- name: Display container status and CLI output
  debug:
    msg:
      - "Container status:"
      - "{{ container_status.stdout_lines }}"
```

```
- "CLI output:"  
- "{{ run_output.stdout_lines }}"
```

5. RUN.sh (local helper)

This is just to shorten the command for checking if any local container is already running and then running a new one. Make it executable and run:

```
chmod +x RUN.sh  
./RUN.sh
```

6. Application Demonstration

Images demonstrating the working of the required functions.

```
1. Addition  
2. Subtraction  
3. Multiplication  
4. Division  
5. Logarithm  
6. Exponentiation  
7. SquareRoot  
8. Factorial  
9. Exit  
  
Enter your choice:  
  
7  
Enter the number:  
  
8  
  
RESULT: 2.8284271247461903
```

Figure 12: Square root function - \sqrt{x}

```
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:

8
Enter the number :

5

RESULT: 120
```

Figure 13: Factorial function - $x!$

```
(base) sparsh@garrison:~/Desktop/COL/SEM 7/Software Production Engineering/Mini Project 1$ ./RUN.sh
Starting Scientific Calculator container...
Welcome to SciCalc - Scientific Calculator

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:

5
Enter the number and the base:

16
4

RESULT: 2.0
```

Figure 14: Logarithm function - $\log_b a$


```
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:

6
Enter the base and the exponent:

3
5

RESULT: 243.0
```

Figure 15: Power function - a^b

```
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:

7
Enter the number:

-23

SQUARE ROOT OF NEGATIVE NUMBERS IS NOT DEFINED.
```

Figure 16: Exception Handling 1

```
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Logarithm
6. Exponentiation
7. SquareRoot
8. Factorial
9. Exit

Enter your choice:

5
Enter the number and the base:

-12 2

LOGARITHM OF NON-POSITIVE NUMBERS IS NOT DEFINED.
```

Figure 17: Exception Handling 2

7. Repository Links

GitHub: <https://github.com/SparshGHub/Scientific-Calculator>

Docker Hub: <https://hub.docker.com/r/sparshdockerman/scicalc>