International Institute of Information Technology, Bengaluru

Software Production Engineering Mini Project Report

Scientific Calculator

Submitted by

Mohit Gupta (MT2024049)

In the guidance of Prof. B. Thangaraju

Problem Statement:

The goal of the project is to create a Scientific Calculator with the following user menu driven options :

1. Square Root Function - ($\sqrt{x}$)

2. Factorial Function - ($x!$)

3. Natural Logarithm (base e) - ($\ln x$)

4. Power Function - (ab)

What is DevOps:

DevOps is a software development approach that unites development and operations teams, enabling them to collaborate throughout the entire application lifecycle. The primary objective of DevOps is to accelerate the delivery of applications and services.

Why DevOps:

DevOps enhances software delivery by automating and optimizing workflows, enabling businesses to deliver software more quickly and effectively. It also fosters improved collaboration between teams.

Benefits of DevOps:

- **Faster delivery**: DevOps allows businesses to release software updates more frequently and rapidly.
- **Improved quality**: By reducing errors, DevOps contributes to enhanced product quality.
- **Enhanced collaboration**: DevOps encourages breaking down silos between development and operations teams, promoting better teamwork.
- **Increased security:** DevOps integrates security practices into the development lifecycle, improving overall security.
- **Scalability:** DevOps supports the management of scalable solutions through automation and optimized workflows.
- **Better customer experience:** By delivering high-quality software, DevOps ensures that customer needs are met effectively.
- **Cost savings:** DevOps reduces the costs associated with slow and inefficient software delivery processes.

Tools and Technologies Used:

Programming Language: Java

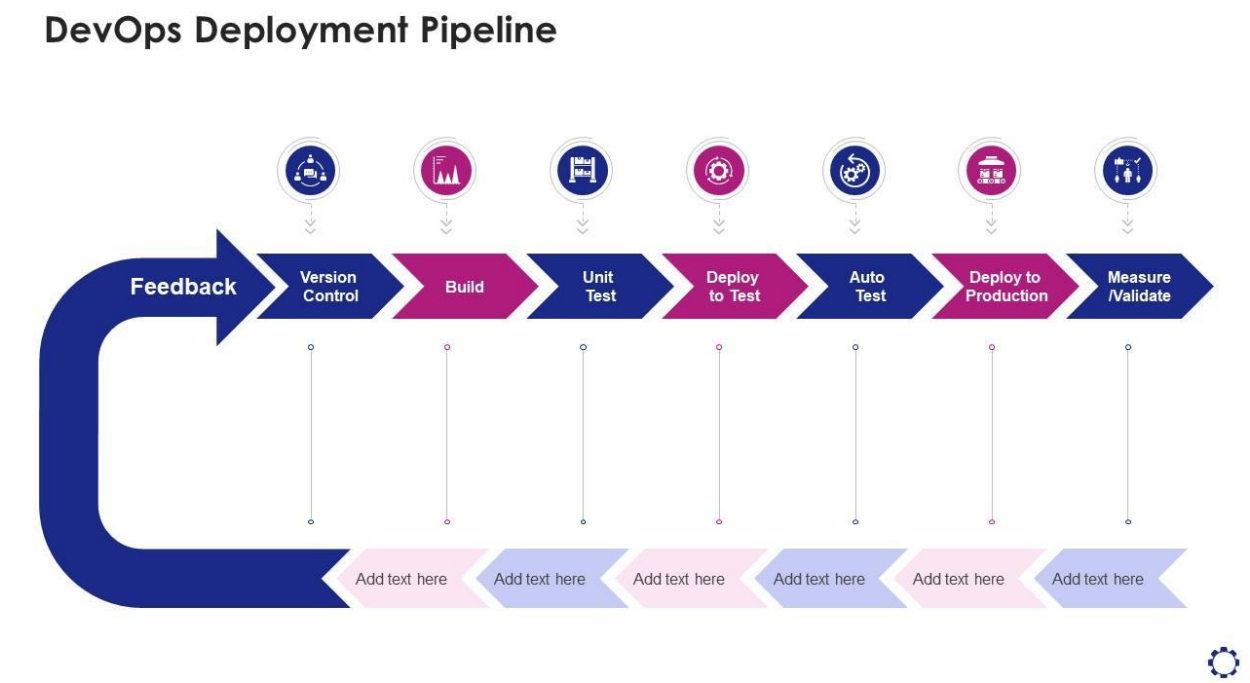
Version Control: Git & GitHub

Build Automation: Maven

CI/CD Pipeline: GitHub Actions / Jenkins

Containerization: Docker

CI/CD Pipeline:

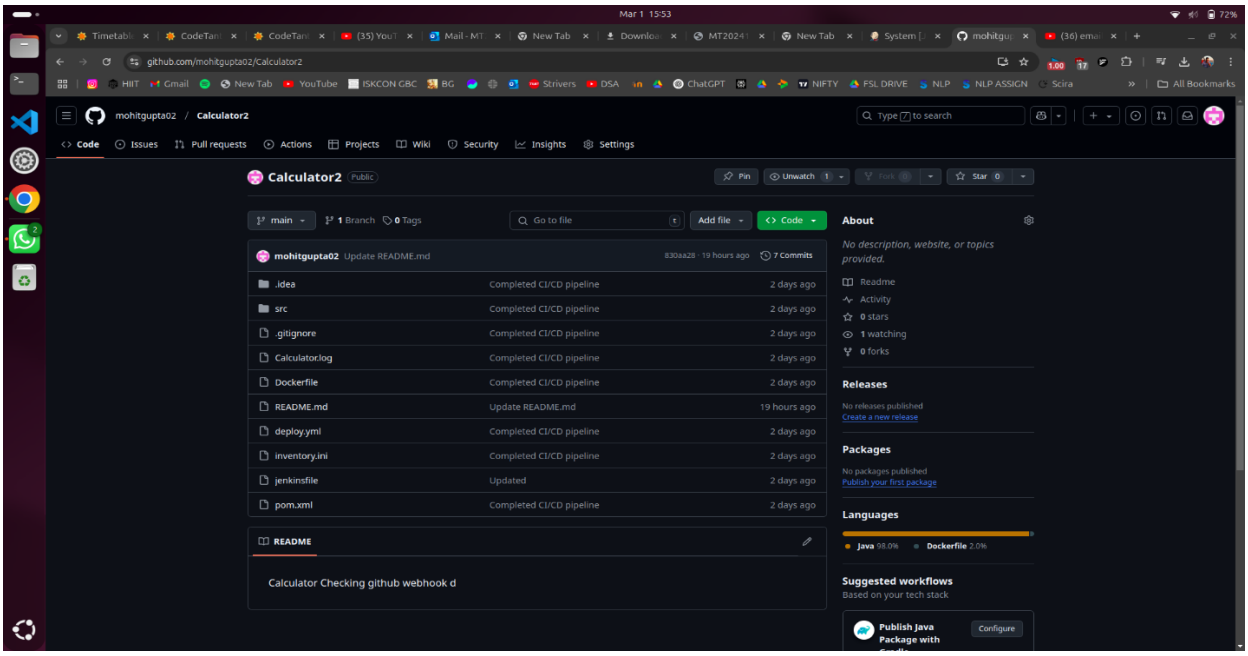


Source Code Management

SCM (Source Code Management) is a system used to track and manage changes made to a source code repository. It monitors modifications to the codebase and assists in resolving conflicts when merging updates from multiple contributors. SCM is often used interchangeably with Version Control.

The project is initially created on a local machine (local repository), and later pushed to a remote repository on platforms like GitHub. To push the local repository to the remote repository, a series of commands are executed, as outlined below:

- git init — Initializes a new Git repository in the local project directory.
- git status — Shows the current status of changes in the working directory.
- git add — Stages the changes to be committed.
- git commit -m “Commit Message” — Commits the staged changes with a descriptive message.
- git remote add origin https://github.com/mohitgupta02/Calculator2.git — Links the local repository to the remote GitHub repository.
- git push -u origin main — Pushes the committed changes to the remote repository's master branch.



Source Code Management

Docker:

- Docker is a tool for containerization that allows packaging software into a standardized unit for development, shipment, and deployment.
- It leverages kernel-level virtualization to create isolated environments known as containers.
- In this case, Docker is used to generate an image that runs the JAR file produced by the Maven build process.
- This image can then be uploaded to Docker Hub, where other machines can pull the image and instantiate containers to run the JAR file.

To build a Docker image, you can use the following command:

$ docker build -t <USERNAME>/<IMAGE_NAME>:<TAG>

To push the Docker image to Docker Hub, use:

$ docker push <USERNAME>/<REPOSITORY_NAME>:<TAG>

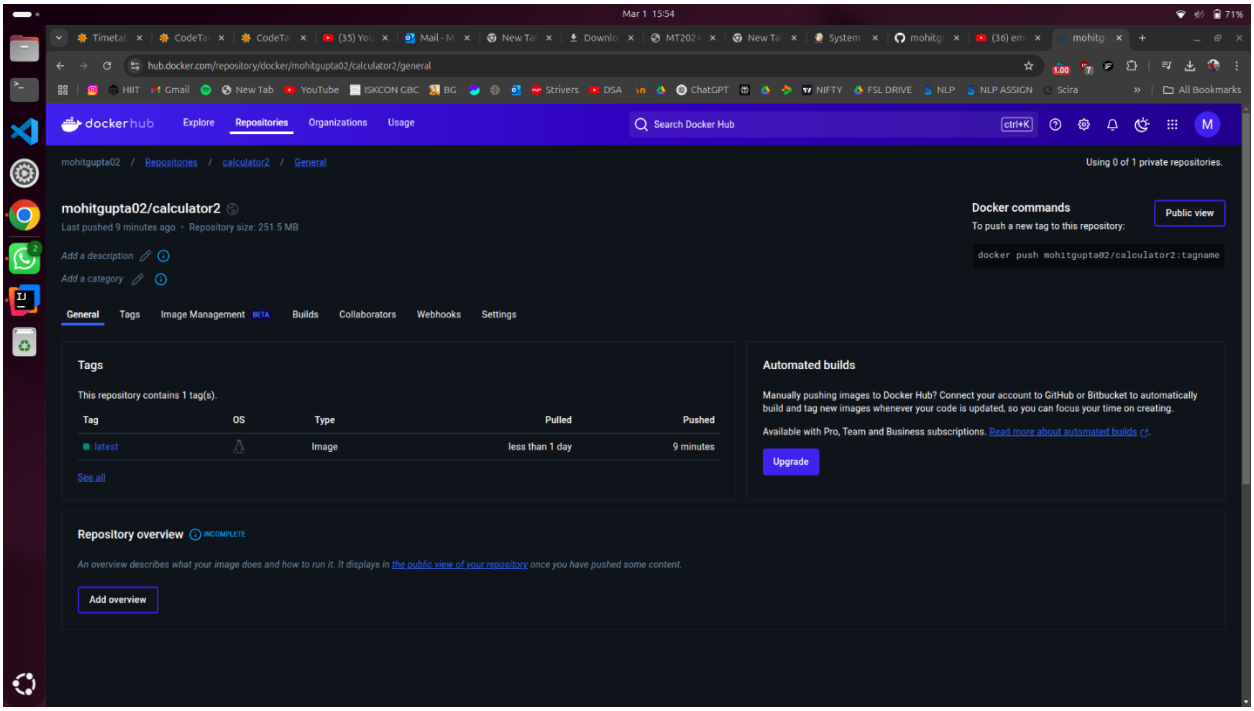
The image is pushed from Jenkins after the build process and pulled during the Ansible job. You can run the Docker image with this command:

$ Docker run -i -t <IMAGE_NAME>

Once the Docker image is pushed to Docker Hub, you can verify it by checking your Docker Hub profile.

DockerFile

- To create the Docker image and upload it to a remote repository on Docker Hub, we will define the steps in a Jenkins pipeline script.
- The Docker image is generated and uploaded to the user's Docker Hub repository.
- After the image is successfully uploaded to Docker Hub, the local image on the machine is removed.



Docker Hub Repository

Jenkins:

- Jenkins is a robust tool that facilitates continuous integration and continuous delivery (CI/CD) for projects, regardless of the platform in use.
- It is an open-source application capable of handling all kinds of builds and CI processes.
- Jenkins can be integrated with a variety of testing and deployment tools.
- The pipeline is set up so that whenever a new commit is made to the GitHub repository, the pipeline is triggered, executing all steps in sequence.
- A Jenkins pipeline is an automated workflow that outlines the process of retrieving software from version control.
- Each change to the software undergoes a series of intricate processes before being released.
- This includes developing the software in a repeatable, reliable manner, and progressing through multiple stages of testing and deployment.
- All tasks are interconnected through the Jenkins pipeline.
- The pipeline pulls changes from GitHub, builds the project using Maven, creates the Docker image, pushes the image to Docker Hub, and then runs the Ansible playbook.

Create CI Pipeline

- A Jenkins pipeline is an automated representation of the steps involved in obtaining software from version control.
- Every change made to the software goes through a series of detailed processes before being released.
- This includes developing the software in a consistent and dependable way, as well as advancing the software through several stages of testing and deployment.

Jenkins Pipeline

There are various stages in Jenkins Pipeline including Checkout, Build Docker Image, Push Docker Image and Run ansible playbook then there is post action to send email for acknowledge the status of pipeline.

```
    stages{
        stage('Checkout'){
            steps{
                script{
                    git branch: 'main', url: "${GITHUB_REPO_URL}"
                }
            }
        }

        stage('Build and test') {
            steps {
                script {
                    sh 'mvn clean package'
                }
            }
        }

        stage('Build Docker Image'){
            steps{
                script{
                    docker.build("${DOCKER_IMAGE_NAME}",'.')
                }
            }
        }

        stage('Push Docker Images'){
            steps{
                script{
                    docker.withRegistry('', 'DockerHub') {
                        sh 'docker tag calculator mohitgupta02/calculator:latest'
                        sh 'docker push mohitgupta02/calculator'
                    }
                }
            }
        }
```

```
            stage('Run Ansible playbook') {
                steps {
                script {
                    ansiblePlaybook(
                        playbook: 'deploy.yaml'
                    )
                }
            }
        }
    }
}
```

Stages of Pipeline

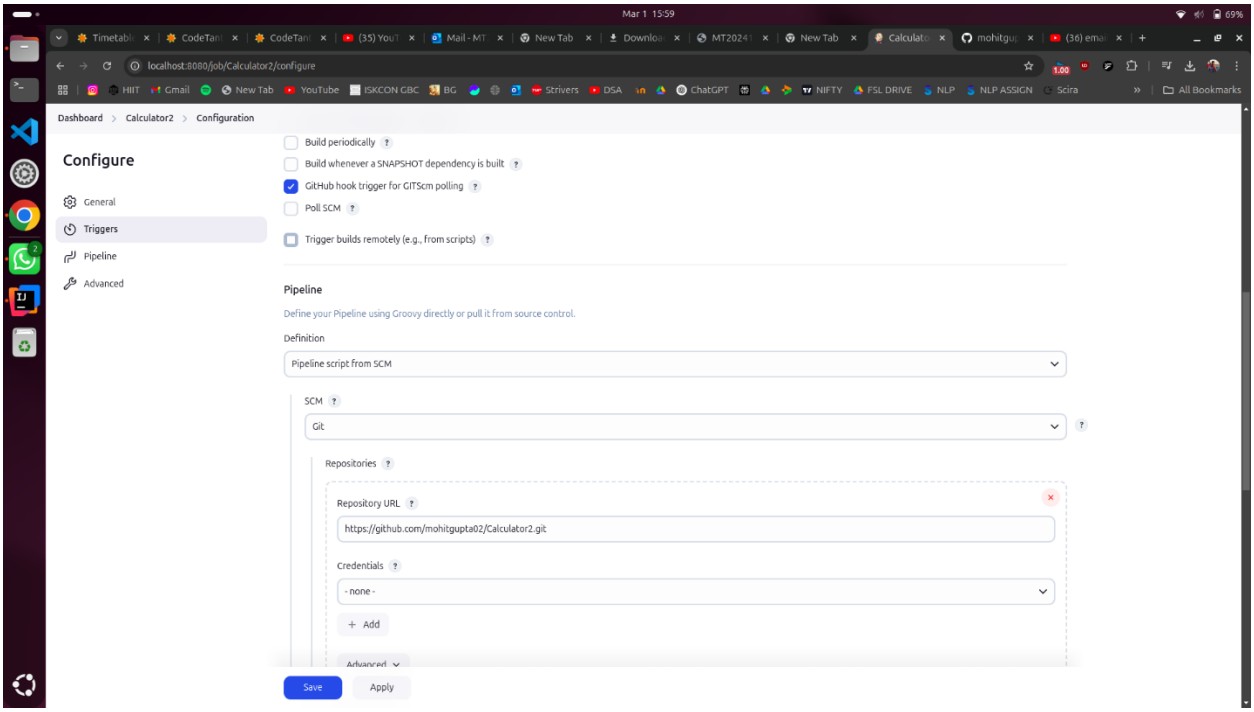
Configuration of Project in Jenkins

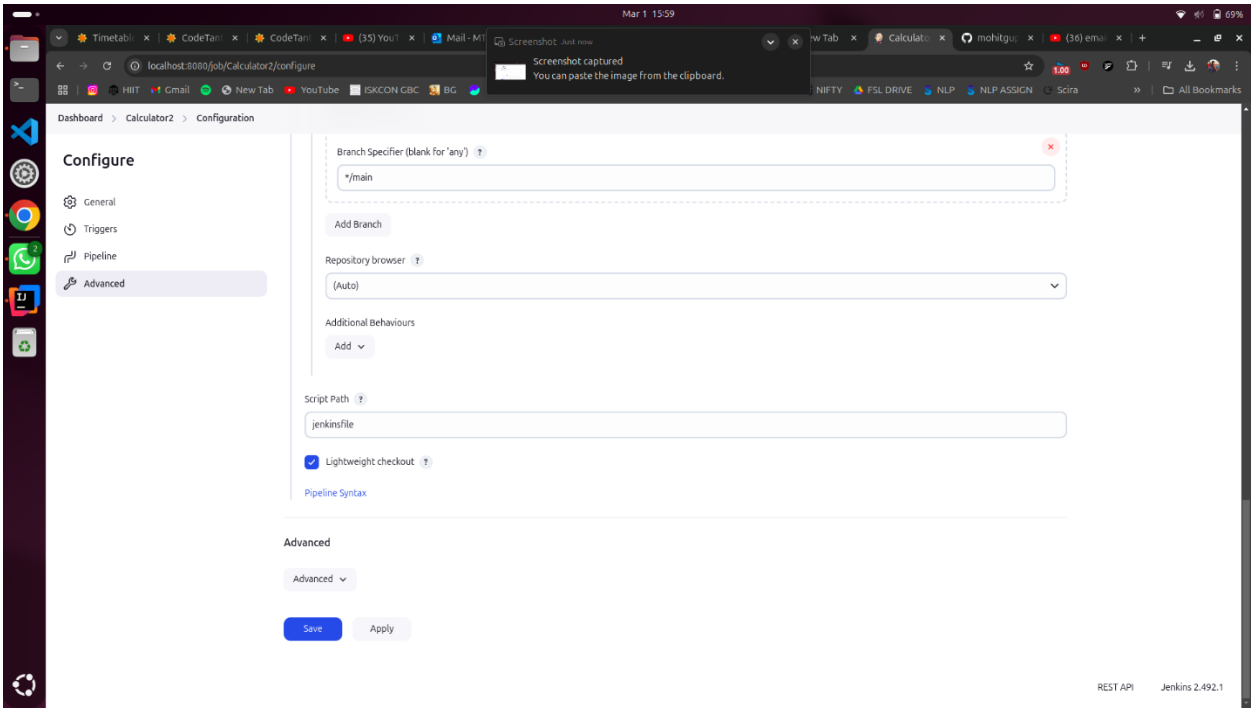
To run Jenkins in local machine there is a command to run it

$ sudo systemctl start jenkins

To check status of Jenkins

$ sudo systemctl status jenkins

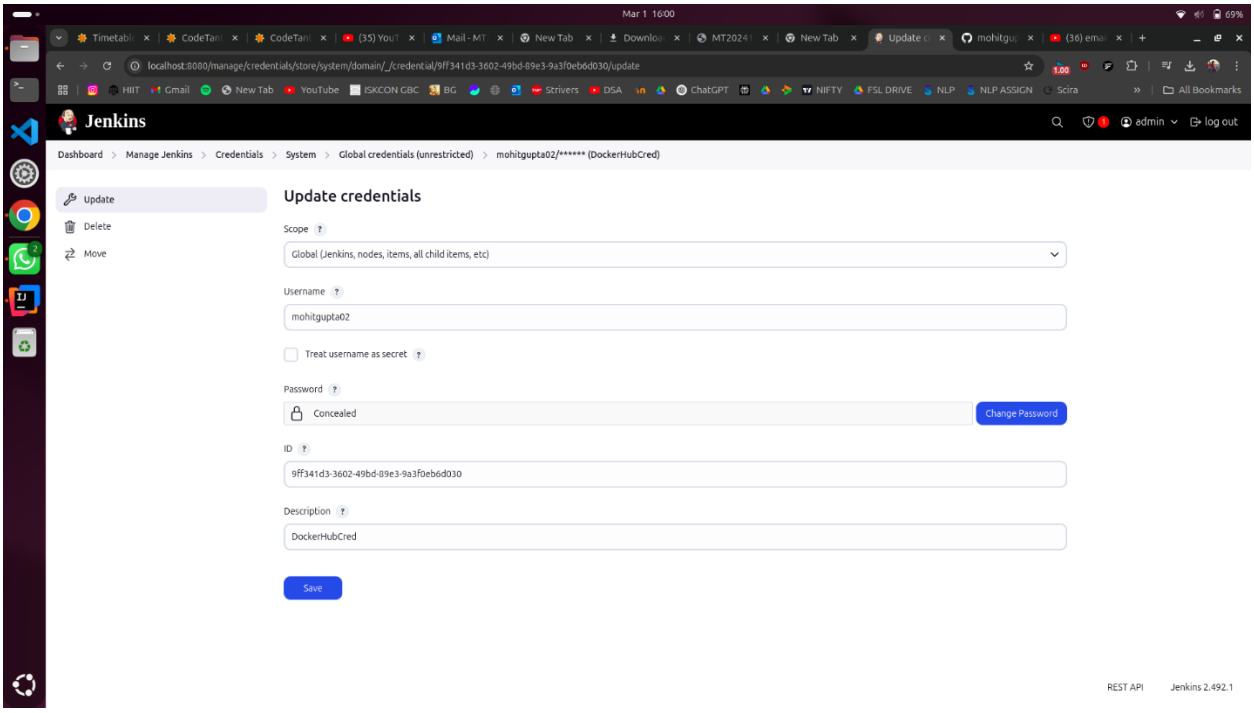




Configuration of Project

- In this step, we configure a GitHub project in Jenkins by providing the URL of our GitHub repository and specifying the path to the Jenkins file in the configuration.
- We then set up system credentials for both Docker Hub and the local machine for the Ansible playbook.
- Since the project runs on the local machine and pulls the image from Docker Hub, it’s necessary to define these system configurations in Jenkins' system credentials.
- Two credentials need to be added: one for the Docker Hub repository and another for the local machine.

To configure Docker Hub credentials, we must specify the Docker Hub username from which the image will be pulled.



Adding DockerHub Credentials

Ansible:

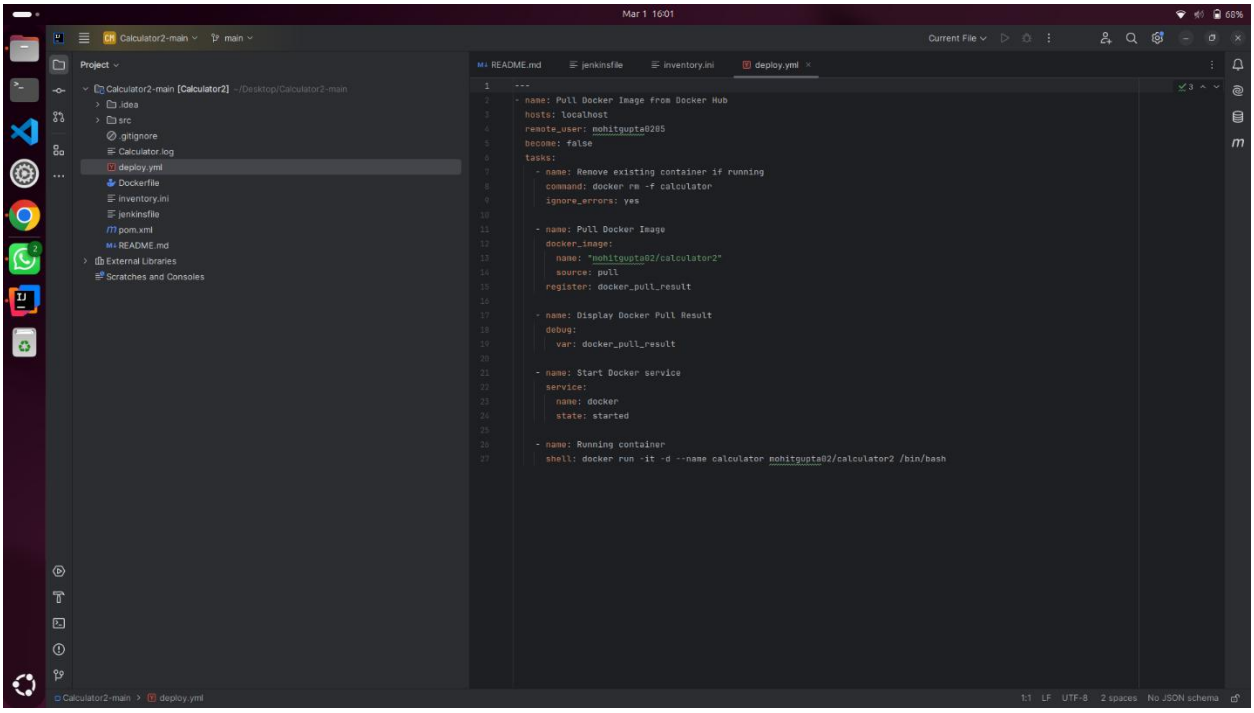
- Ansible is a continuous deployment tool. In which there is one control node and the nodes on which the deliverables are deployed are called managed nodes.
- Ansible need to be installed only on the control node.
- Ansible connects with the managed nodes using ssh.
- In Ansible we provide a set of commands that need to be executed on the managed node in the form of yml file.
- It is called a playbook. Along with the playbook we also provide information about the managed nodes. Which called inventory.

To execute this playbook we run the following command:

$ ansible-playbook <playbook> -i <inventory>

Inventory File

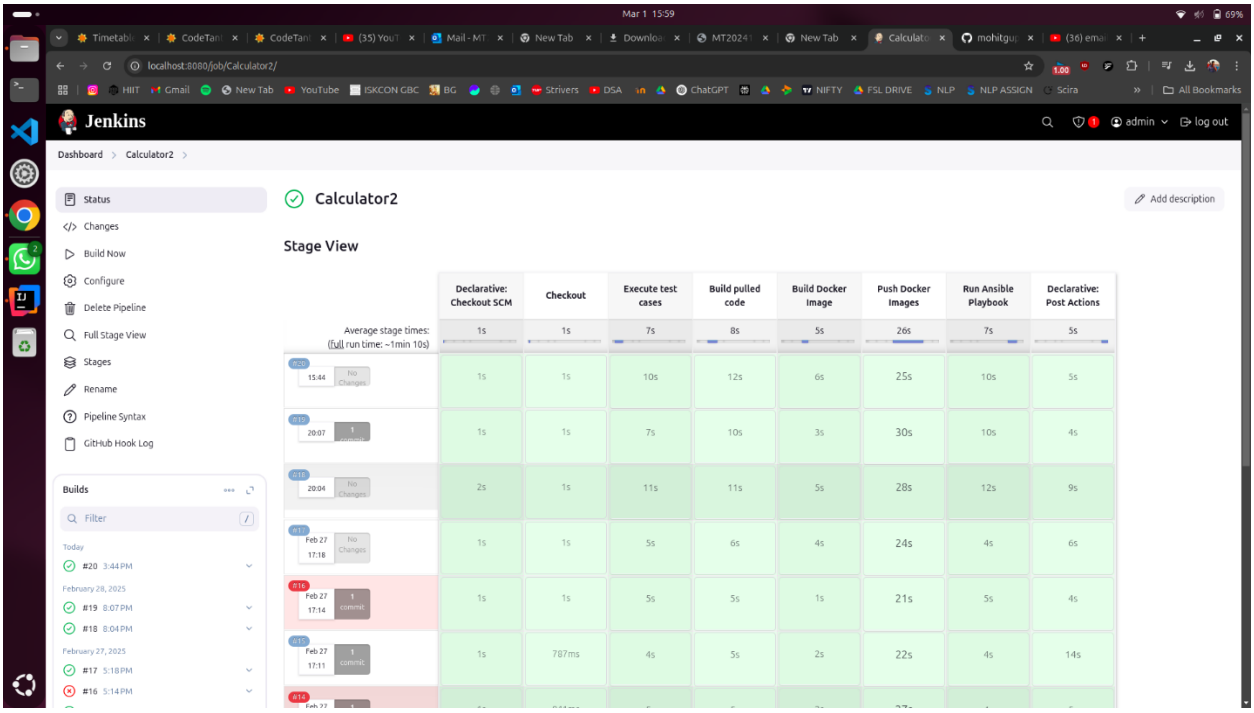
- The Ansible inventory file specifies the hosts and host groups on which tasks, modules, and commands within a playbook are executed.
- The format of the inventory file can vary based on the Ansible environment and the plugins in use.
- Ansible playbooks are used to document and carry out configuration, deployment, and orchestration tasks.
- To run the application’s Docker image on the host, Python's pip and Docker must be installed on the host machine.
- The Ansible playbook pulls the Docker image from Docker Hub and launches the container on the specified hosts.
- This entire process is automated through the playbook.



Playbook file

Run Project as Pipeline in Jenkins

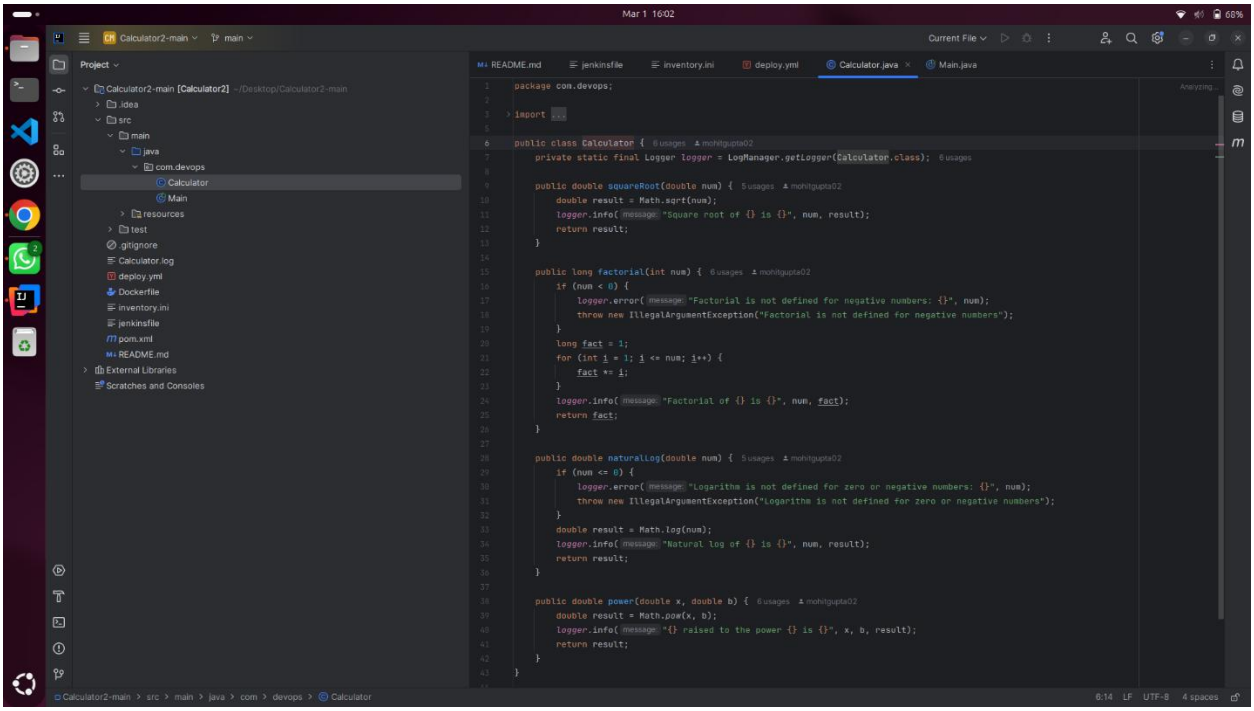
To run the project as pipeline in jenkins we have to click build now to see each stages of pipeline and show the output of each stages.

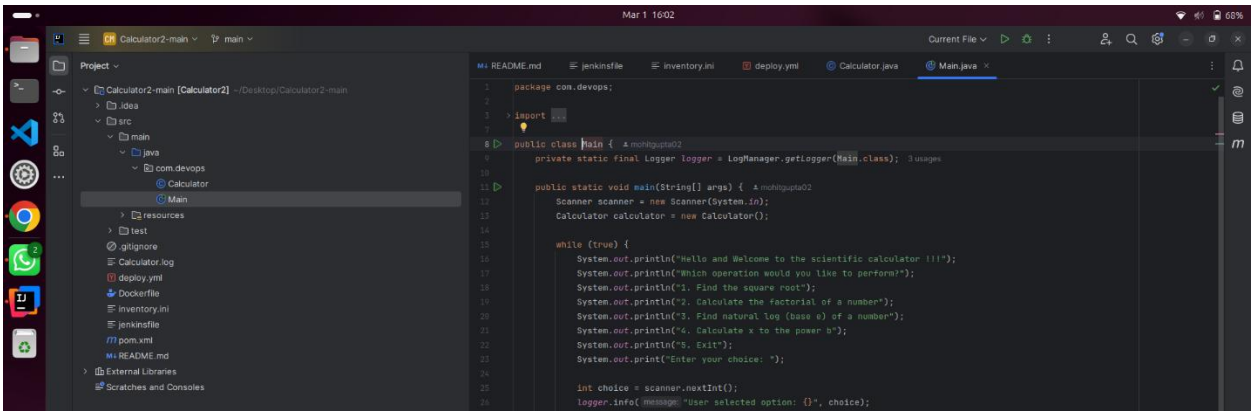


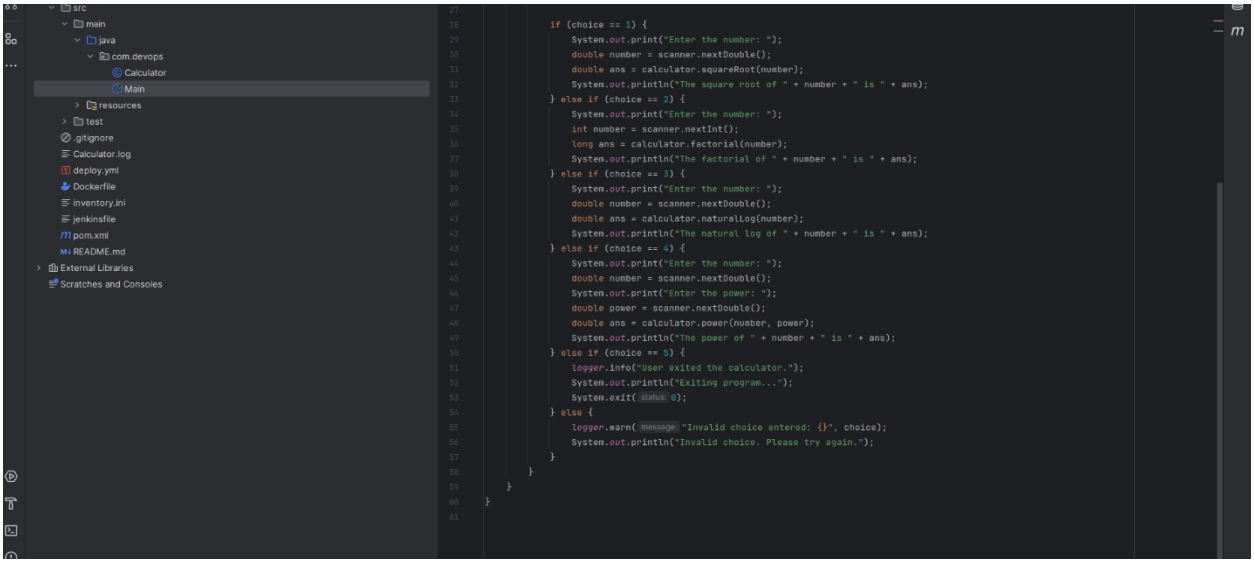
View of Pipeline Stages

Scientific Calculator Source Code:

Below attached the source code of scientific calculator in java






src
main
java
com.devops
Calculator
Main
resources
test
.gitignore
Calculator.log
deploy.yml
Dockerfile
inventory.ini
jenkinsfile
pom.xml
README.md
External Libraries
Scratches and Consoles
if (choice == 1) {
System.out.print("Enter the number: ");
double number = scanner.nextDouble();
double ans = calculator.squareRoot(number);
System.out.println("The square root of " + number + " is " + ans);
} else if (choice == 2) {
System.out.print("Enter the number: ");
int number = scanner.nextInt();
long ans = calculator.factorial(number);
System.out.println("The factorial of " + number + " is " + ans);
} else if (choice == 3) {
System.out.print("Enter the number: ");
double number = scanner.nextDouble();
double ans = calculator.naturalLog(number);
System.out.println("The natural log of " + number + " is " + ans);
} else if (choice == 4) {
System.out.print("Enter the number: ");
double number = scanner.nextDouble();
System.out.print("Enter the power: ");
double power = scanner.nextDouble();
double ans = calculator.power(number, power);
System.out.println("The power of " + number + " is " + ans);
} else if (choice == 5) {
logger.info("User exited the calculator.");
System.out.println("Exiting program...");
System.exit(status: 0);
} else {
logger.warn(message: "Invalid choice entered: {}", choice);
System.out.println("Invalid choice. Please try again.");
}
}
}
}

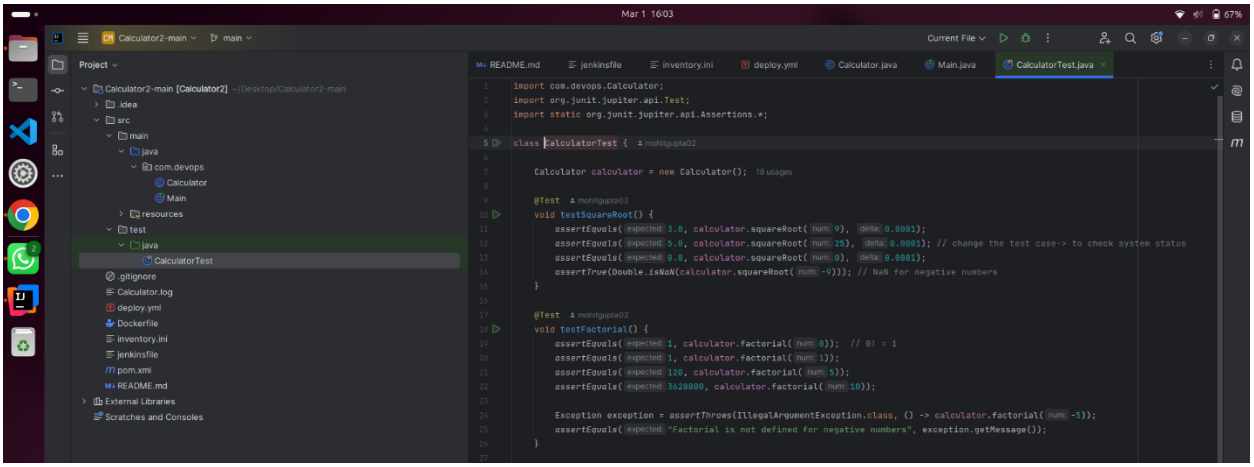
Adding Test Cases:

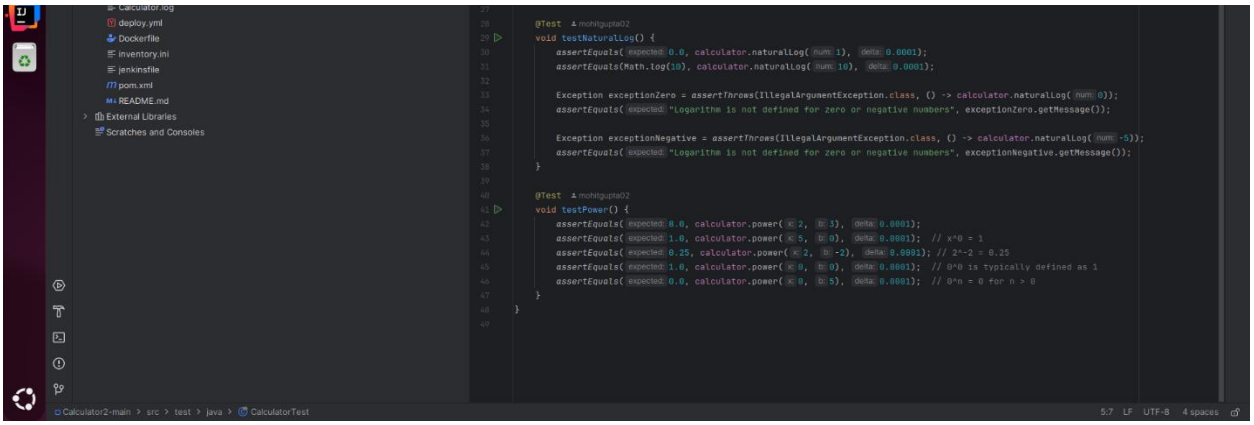
What is Unit Testing?

- Unit Testing is the process of testing individual components of code in isolation.
- A "unit" refers to the smallest functional part of a system, which could be a single line of code, a method, or an entire class.
- The smaller the unit being tested, the faster and more efficient the process becomes.
- This helps in gaining deeper insights into the code's functionality and overall performance.

JUnit: A Unit Testing Framework for Java

- JUnit is an open-source framework designed for unit testing in Java.
- It allows developers to write and execute automated test cases efficiently.
- In Java development, test cases must be rerun whenever new code is introduced to ensure that existing functionality remains intact.
- JUnit also provides visual feedback on test execution through graphical indicators.
- A successful test is shown in green, while a failed test appears in red.
- By leveraging JUnit, developers can create robust, error-free code with greater reliability.

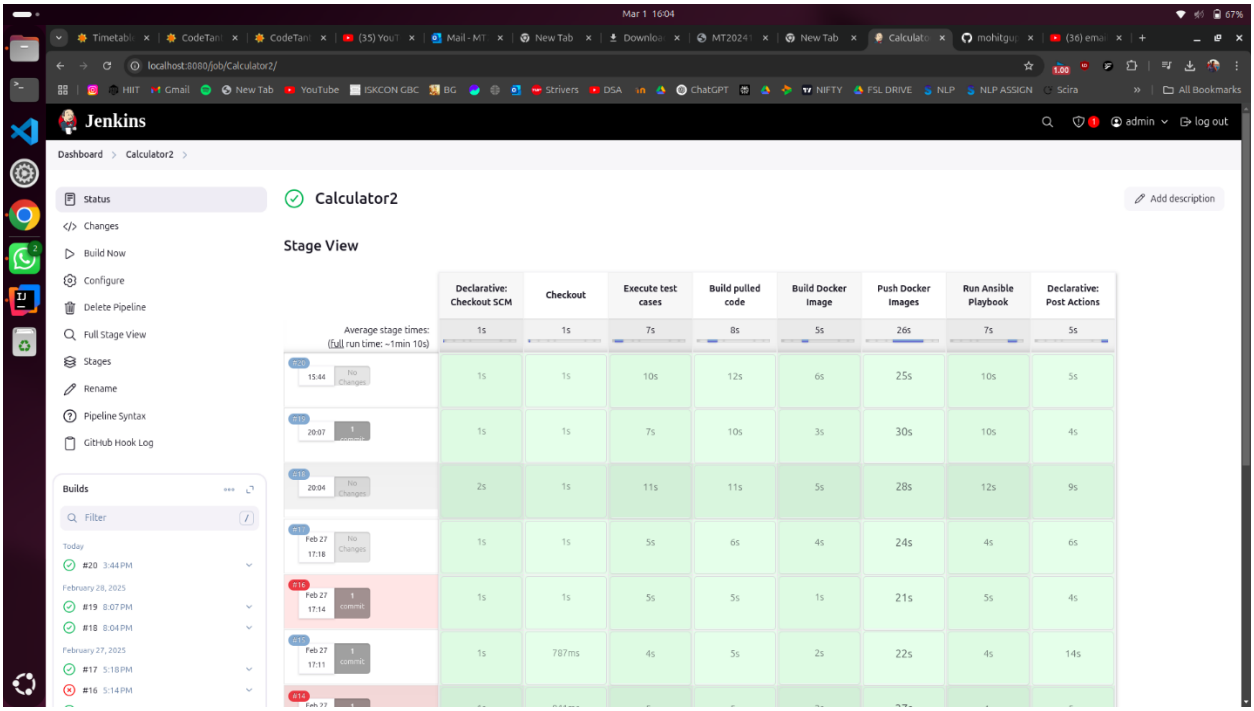




Adding Test Cases Java File

Run Project:

To run project first run pipeline in Jenkins



Pipeline View

Webhooks:

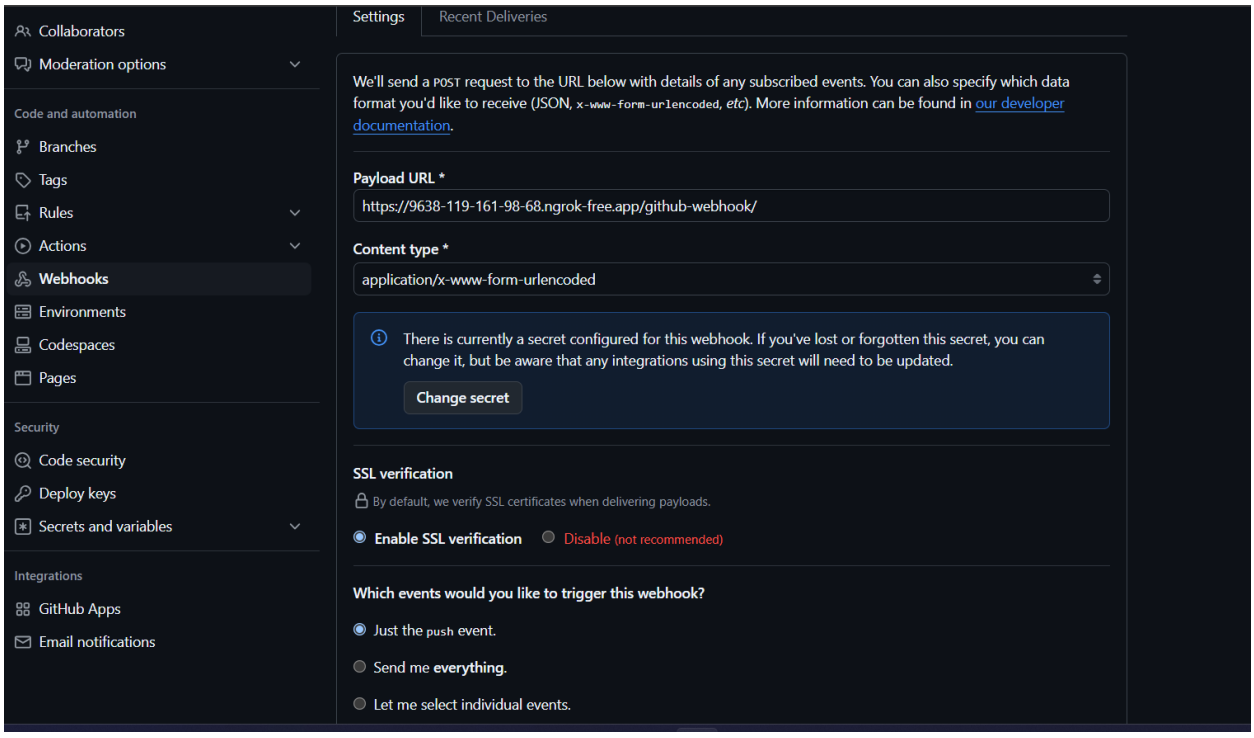
Webhooks are automated HTTP callbacks that trigger actions when an event occurs, such as a code push in a repository.

Use of Webhooks in Jenkins with Ngrok

Ngrok is used to expose Jenkins running on a local machine to the internet, enabling webhooks to communicate with it. This allows GitHub/GitLab webhooks to trigger Jenkins builds even when running on a local server.

Key Benefits:

- **Automatic Build Triggers:** Webhooks notify Jenkins of code changes.
- **Remote Accessibility:** Ngrok provides a public URL for Jenkins, making it accessible for webhook events.
- **Faster CI/CD:** Automates build and deployment without manual intervention.



Webhooks Using Through the Git-hub

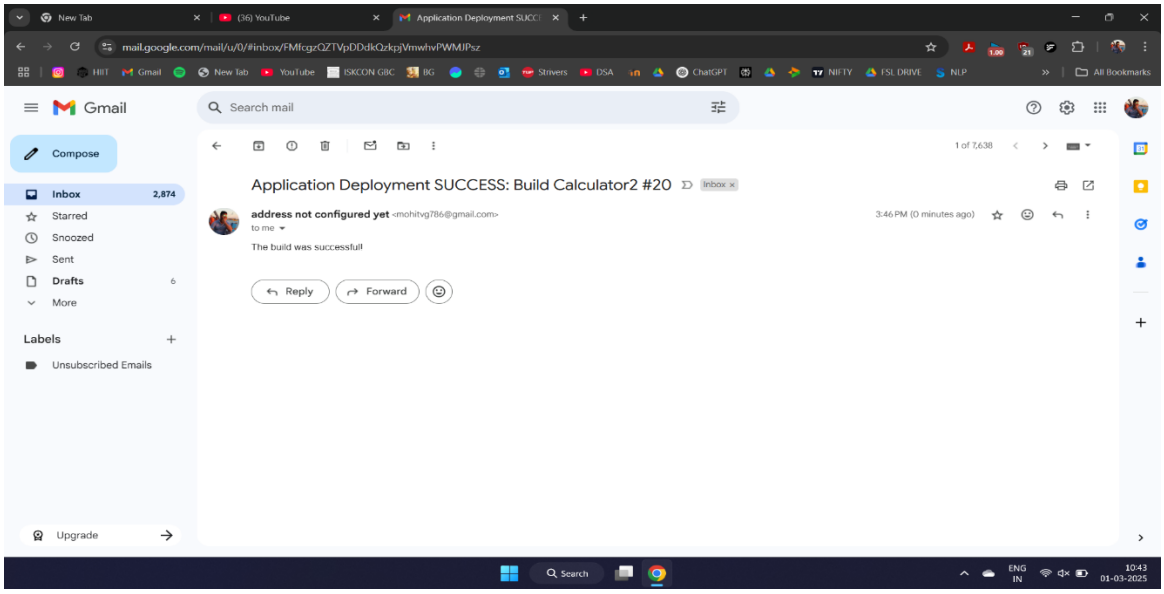
Email Notification In Jenkins Pipeline project:

What is SMTP in Jenkins?

SMTP (Simple Mail Transfer Protocol) allows Jenkins to send email notifications about build status, failures, or successes to developers and stakeholders. This ensures timely updates on project progress.

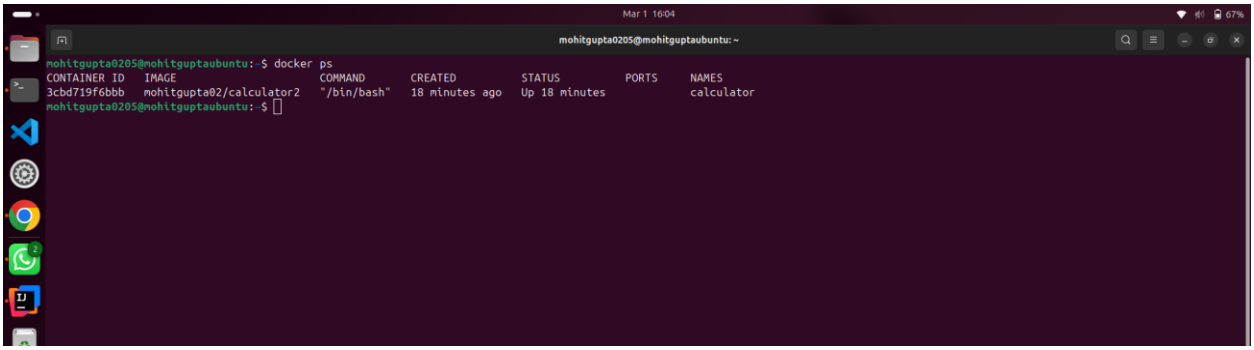
Use of SMTP in Jenkins

- Automated Email Alerts: Sends build status updates after each pipeline run.
- Failure Notifications: Alerts developers immediately if a build fails.
- Custom Messages: Can include logs, reports, and issue details in emails.

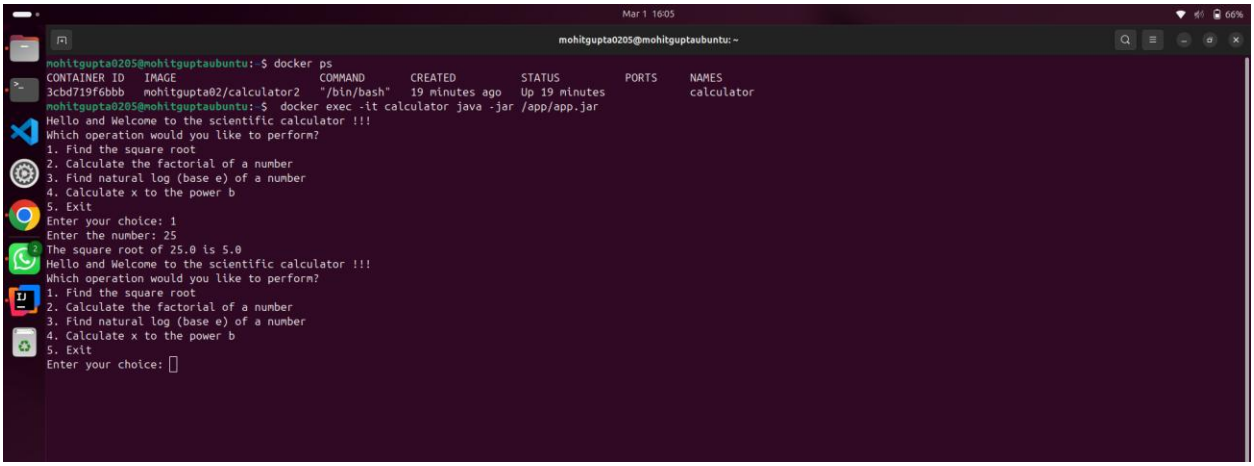


Email Photo Send by Jenkins after Deployment

Project Runs successfully and now our docker image is also pulled from docker hub. We Can run in local machine to check whether it runs or not by using below command.

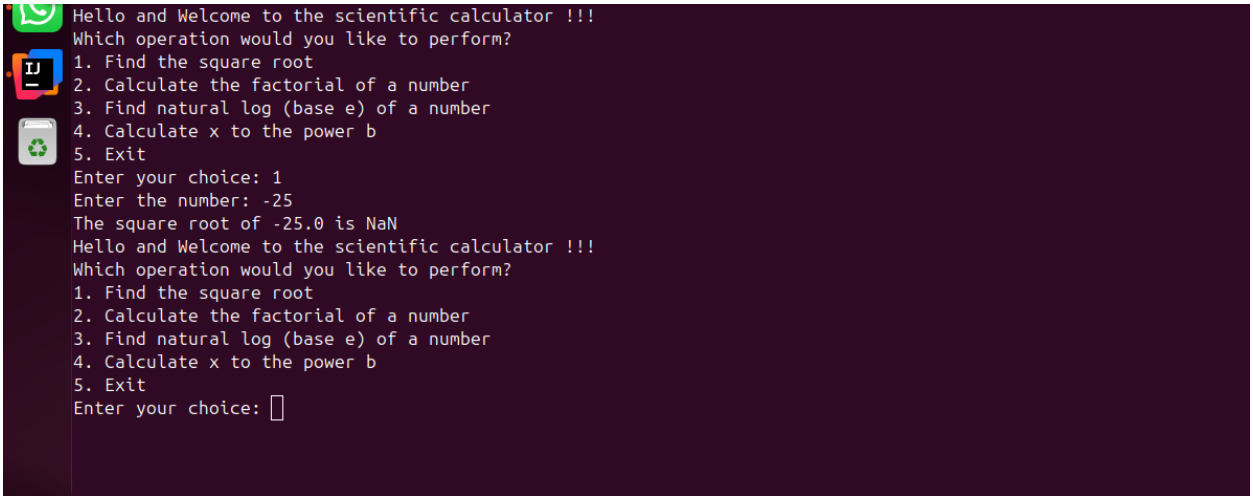


So, right now our container is running and to execute this code we are using below command to enter the container and check whether our code is working or not.



Execute Container

Check for Some Test Cases:



Checking for the Execution

GitHub Repo : https://github.com/mohitgupta02/Calculator2

Dockerhub Repo:
https://hub.docker.com/repository/docker/mohitgupta02/calculator2/general