Mini Project Report

CS 836: Software Production Engineering

March 22, 2023

Calculator Using DevOps Toolchain

Manas Gupta (IMT2019050)

1.Introduction

A calculator is a simple computer program that conducts arithmetic operations such as addition, subtraction, multiplication, and division. We examine the functions square root, factorial, natural logarithm, and power in our implementation. It can be created with a DevOps toolchain that includes the following stages.

1) Version Control: Use a version control system like Git and GitHub to keep track of changes in the codebase.
2) Build: Use a build tool like Maven to automate the build process. The build tool can compile the source code, create a binary file, and run unit tests.
3) Test: Use a testing framework like JUnit to automate the testing process. The testing framework can run unit tests and generate reports on test coverage and test results.
4) Continuous Integration: Use a CI/CD server like Jenkins to continuously integrate your code.
5) Containerize: Use a containerization tool like Docker to create a container and push it to the Docker Hub.
6) Webhooks: To use webhooks, the private IP address of the local machine must be changed to a public IP address using Ngrok.
7) Deploy: Use a deployment tool like Ansible to perform the configuration management and deployment.
8) Logging: Use a Java-based logging utility like log4j. It is a Java-based logging utility that provides a flexible logging infrastructure for Java applications. It is used to log messages from applications, allowing developers to track the flow of their code and diagnose problems that may occur during runtime.
9) Monitoring: Use a monitoring tool like the ELK Stack to monitor the application and generate alerts on errors and failures.

We can automate every step of software development, from code changes to deployment and tracking, by utilizing a DevOps toolchain. This could facilitate the quicker and more accurate delivery of high-quality software.

2. What is DevOps?

DevOps is a software development methodology that encourages team cooperation, communication, and automation in order to dissolve the divide between development (Dev) and operations (Ops) teams. By ensuring that development and operations teams collaborate effectively, DevOps aims to help organizations produce software products and services more quickly, reliably, and with higher quality.

A combination of techniques, resources, and societal standards are used in DevOps, including:

1) **Continuous Delivery/Deployment and Continuous Integration (CI/CD**): This entails automating the development, testing, and distribution of software in order to guarantee the prompt and reliable delivery of changes.
2) Treating infrastructure as software code, or **infrastructure as code (IaC),** allows organizations to manage their infrastructure with the same instruments and procedures they use to manage application code.
3) Breaking down team silos and fostering cooperation and communication between development, operations, and other stakeholders are part of this process.
4) **Agile and Lean methodologies**: These methods put a strong emphasis on iterative, incremental development and continuous growth, which is in line with the DevOps way of thinking.

3. Why DevOps?

Organizations are embracing DevOps as a software development methodology for a number of reasons. Here are a few of the key advantages:

1) **Time-to-market** is shortened because of DevOps, which allows businesses to release software solutions more frequently and with shorter lead times. By doing this, businesses can remain one step ahead of the competition and react quickly to shifting market conditions.
2) **Collaboration and communication** between teams are encouraged by DevOps, which can result in better alignment, increased effectiveness, and quicker issue solving.
3) **Enhanced quality and dependability**: DevOps can increase the general quality and dependability of software goods and services by

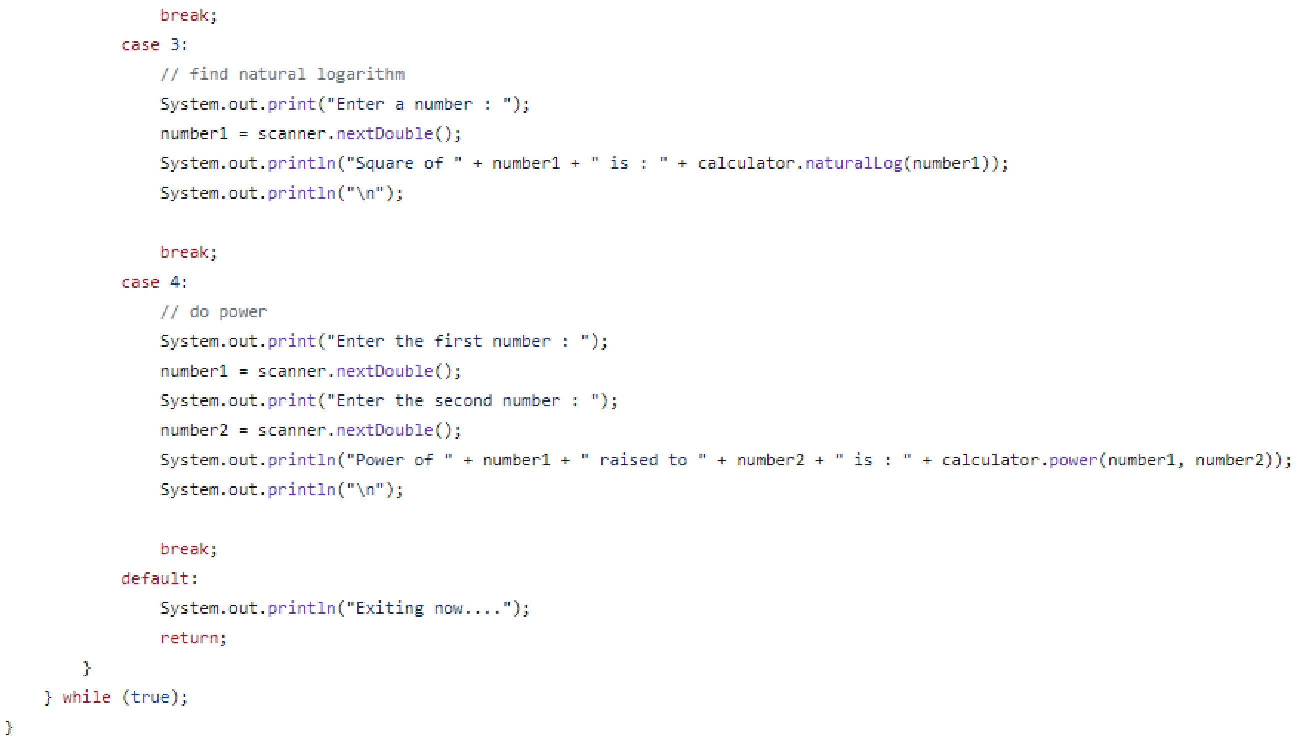
lowering the risk of errors and outages by automating testing and deployment.

4) **Better customer experience**: DevOps can enhance the customer experience and increase customer happiness by providing software products and services more frequently and with higher quality.
5) **Greater agility and innovation**: DevOps allows businesses to react quickly to shifting customer demands and market dynamics, which promotes innovation.

In general, DevOps can assist businesses in streamlining their software development and delivery procedures, allowing them to innovate more rapidly, lower risks, and provide customers with value more reliably and swiftly.

4. Development

The Java 8 software was created. Based on the input choice, a switch case is used to implement the calculator's various functionalities, with distinct functions for each feature.

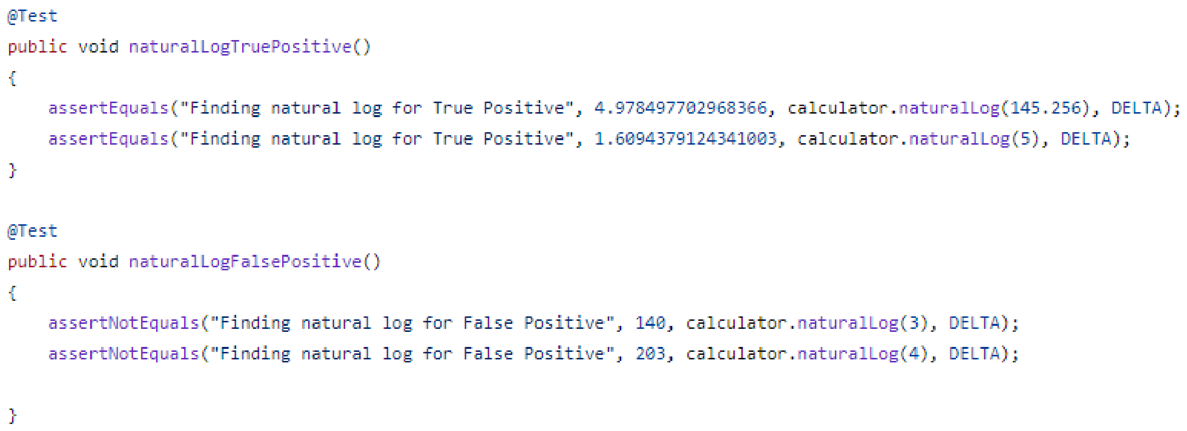


```java
                break;
            case 3:
                // find natural logarithm
                System.out.print("Enter a number : ");
                number1 = scanner.nextDouble();
                System.out.println("Square of " + number1 + " is : " + calculator.naturalLog(number1));
                System.out.println("\n");

                break;
            case 4:
                // do power
                System.out.print("Enter the first number : ");
                number1 = scanner.nextDouble();
                System.out.print("Enter the second number : ");
                number2 = scanner.nextDouble();
                System.out.println("Power of " + number1 + " raised to " + number2 + " is : " + calculator.power(number1, number2));
                System.out.println("\n");

                break;
            default:
                System.out.println("Exiting now....");
                return;
        }
    } while (true);
}
```

Figure 1. A switch statement

5. Testing

JUnit test cases are created to unit test every aspect of the calculator's functionality. Java's JUnit testing framework is used to create and execute unit tests for Java programs. A software testing method known as unit testing involves testing individual software application modules or components to make sure they are functioning properly. While the project is being created, the code is checked using test cases in the test file that are both true and false positive.

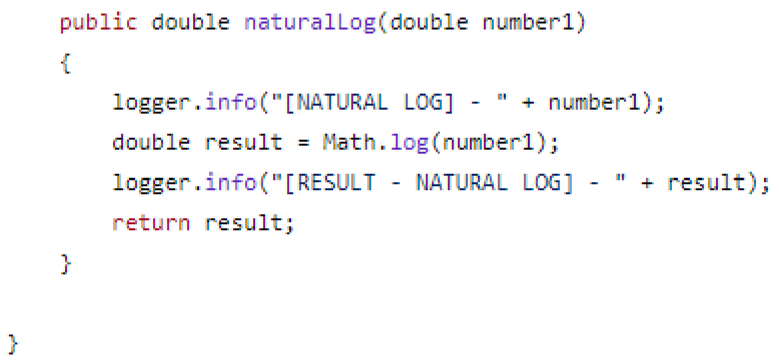


```
@Test
public void naturalLogTruePositive()
{
    assertEquals("Finding natural log for True Positive", 4.978497702968366, calculator.naturalLog(145.256), DELTA);
    assertEquals("Finding natural log for True Positive", 1.6094379124341003, calculator.naturalLog(5), DELTA);
}

@Test
public void naturalLogFalsePositive()
{
    assertNotEquals("Finding natural log for False Positive", 140, calculator.naturalLog(3), DELTA);
    assertNotEquals("Finding natural log for False Positive", 203, calculator.naturalLog(4), DELTA);

}
```

Figure 2: Test code

6. Logging

Popular Java-based logging tool Log4j (short for Log for Java) offers a flexible and configurable logging framework for Java apps. It enables programmers to produce log lines from their code to record information about the behavior, efficiency, and errors of the application during runtime. Developers can use Log4j to set log statements to be written to a variety of locations, including the console, files, emails, or other unique locations. Additionally, Log4j supports log levels, which give developers control over the degree of detail in the data recorded in the logs. DEBUG, INFO, WARN, ERROR, and FATAL are some of the log categories.



```
    public double naturalLog(double number1)
    {
        logger.info("[NATURAL LOG] - " + number1);
        double result = Math.log(number1);
        logger.info("[RESULT - NATURAL LOG] - " + result);
        return result;
    }

}
```

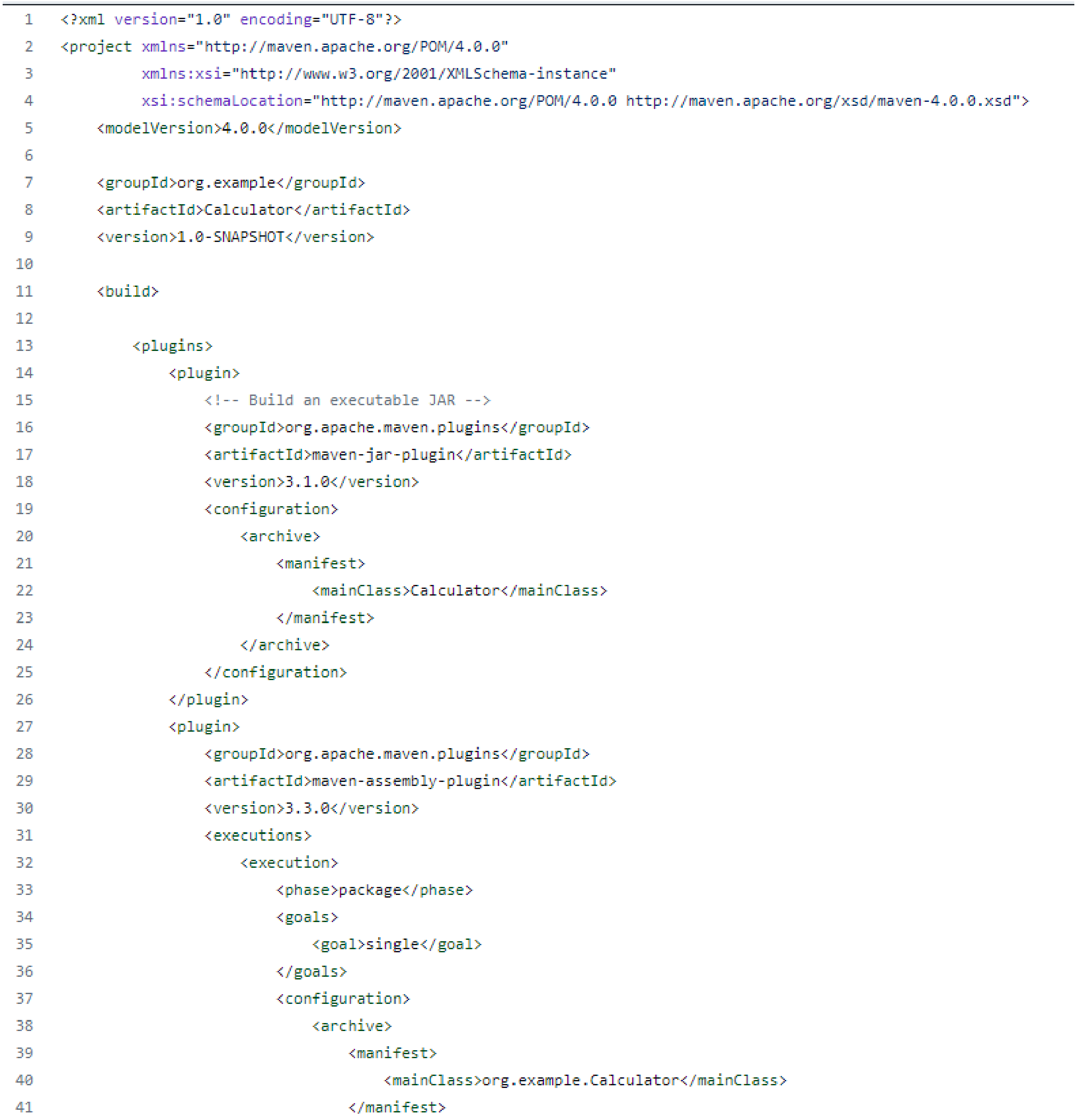
Figure 3: Logging task

7. Building Project

The build automation tool Maven is mostly utilized for Java applications. By specifying the project structure, dependencies, and modules, it aids in build management. Some of Maven's primary applications:

1) **Dependency management**: Maven assists with dependency management by obtaining necessary libraries and frameworks from remote sources.
2) **Construction of the project**: Maven constructs the project by assembling the source code, putting it in an executable package (like a JAR file), and executing tests.
3) **Organizing** code into logical components and providing a consistent build process, Maven specifies a standard project structure that aids developers.

Overall, Maven enhances code quality, streamlines the build process for Java applications, and facilitates team collaboration. We specify the pom.xml file, a component of the Maven build, to use specific dependencies like JUnit in our project.

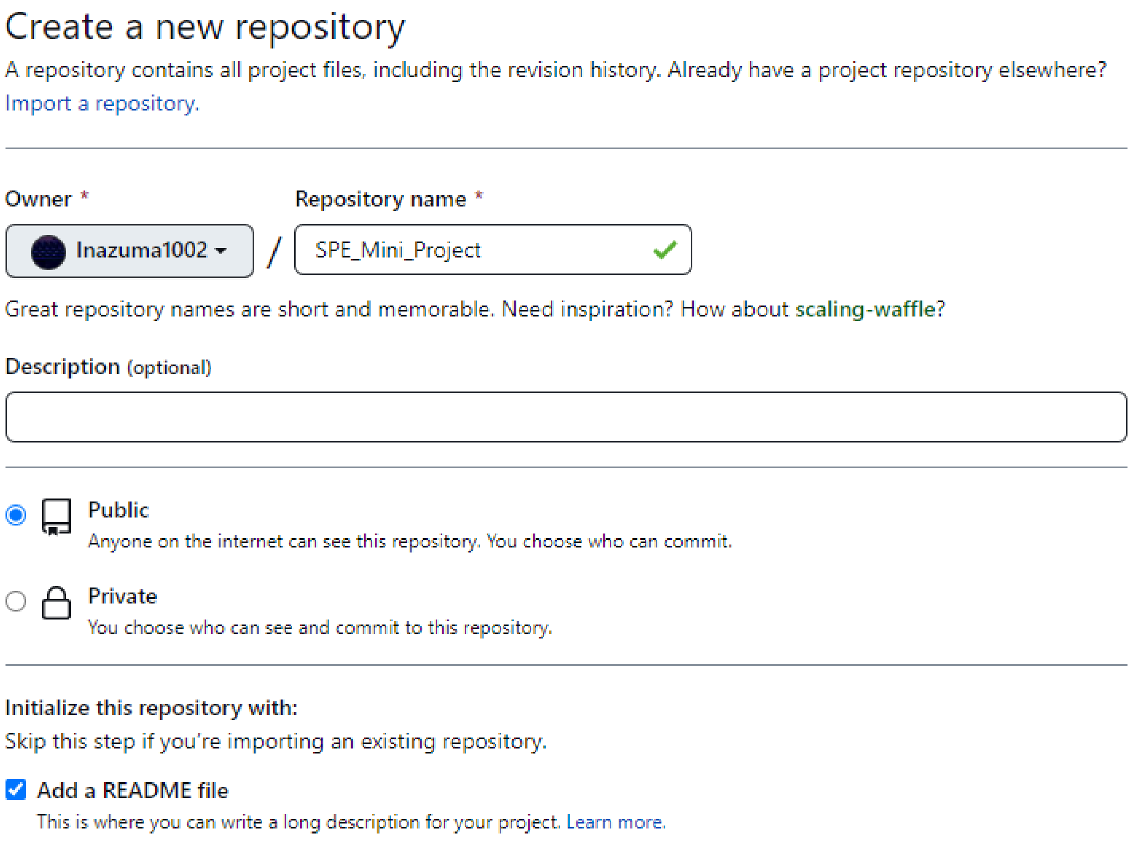


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>Calculator</artifactId>
    <version>1.0-SNAPSHOT</version>

    <build>

        <plugins>
            <plugin>
                <!-- Build an executable JAR -->
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jar-plugin</artifactId>
                <version>3.1.0</version>
                <configuration>
                    <archive>
                        <manifest>
                            <mainClass>Calculator</mainClass>
                        </manifest>
                    </archive>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-assembly-plugin</artifactId>
                <version>3.3.0</version>
                <executions>
                    <execution>
                        <phase>package</phase>
                        <goals>
                            <goal>single</goal>
                        </goals>
                        <configuration>
                            <archive>
                                <manifest>
                                    <mainClass>org.example.Calculator</mainClass>
                                </manifest>
```

The codebase will be built, dependencies will be fetched, test cases will be performed, and results will be reported by running the command “mvn clean install”. The executable “.jar” files will be located in a new directory called “target” that will be created in the current directory.

8. Source Code Management Using Git and GitHub

Tracking and managing modifications to software code over time is called **source code management (SCM)**, also known as version control. SCM enables developers to work together on a project, keep track of past modifications, and manage access to various code versions. **Git** is a well-liked distributed version control system (DVCS) for monitoring updates to software code. It was developed by Linus Torvalds in 2005 and is extensively used by businesses and software developers worldwide.
Software development projects using the Git version management system are hosted on the web platform **GitHub**. It is a well-liked tool for managing private projects inside of companies as well as for working together on open source software projects.
Creation of GitHub repository:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

Inazuma1002 / SPE_Mini_Project ✓

Great repository names are short and memorable. Need inspiration? How about scaling-waffle?

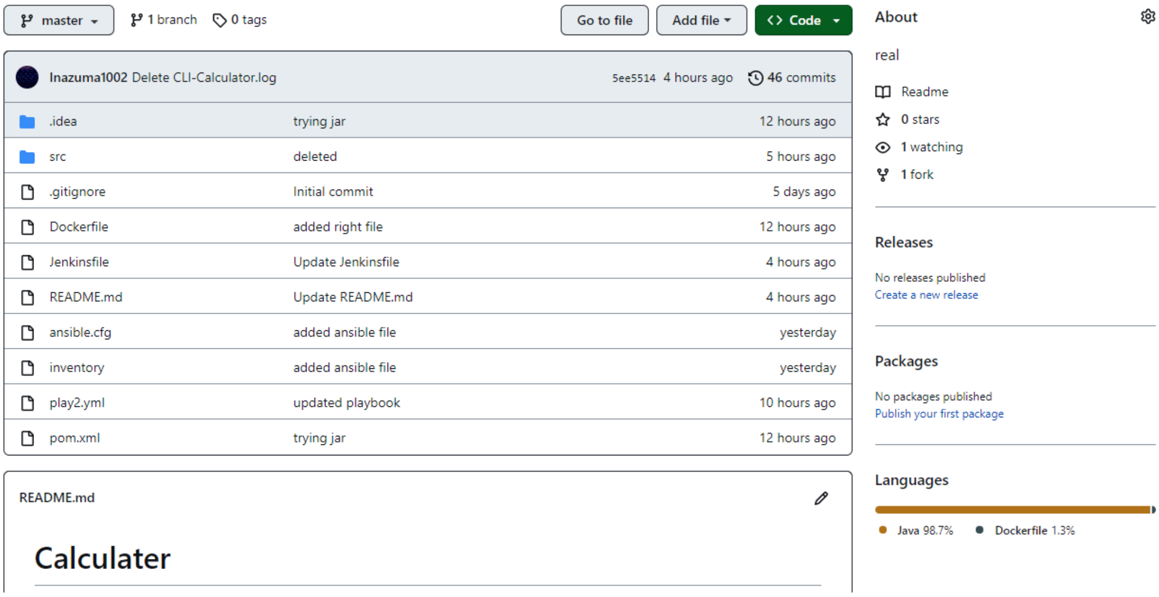
Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ Add a README file
This is where you can write a long description for your project. Learn more.



Inazuma1002 Delete CLI-Calculator.log 5ee5514 4 hours ago 46 commits

.idea	trying jar	12 hours ago
src	deleted	5 hours ago
.gitignore	Initial commit	5 days ago
Dockerfile	added right file	12 hours ago
Jenkinsfile	Update Jenkinsfile	4 hours ago
README.md	Update README.md	4 hours ago
ansible.cfg	added ansible file	yesterday
inventory	added ansible file	yesterday
play2.yml	updated playbook	10 hours ago
pom.xml	trying jar	12 hours ago

README.md

Calculater

About

real

Readme

0 stars

1 watching

1 fork

Releases

No releases published
Create a new release

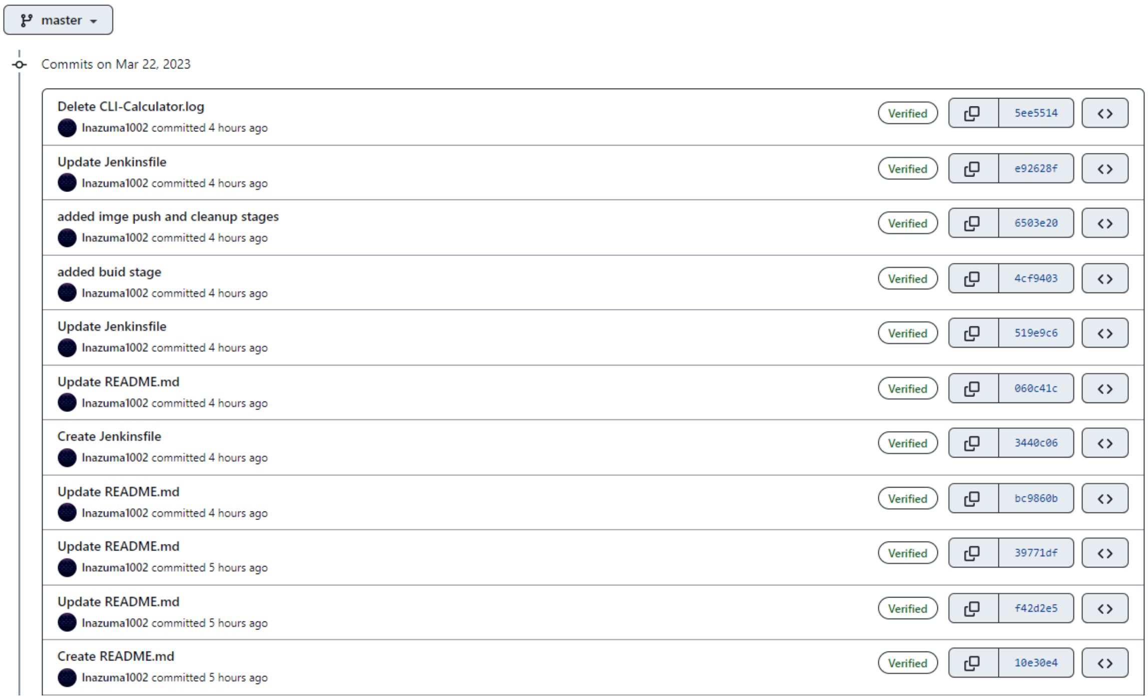
Packages

No packages published
Publish your first package

Languages

● Java 98.7% ● Dockerfile 1.3%

You can access the code at “**https://github.com/Inazuma1002/Calculater.git**”.
Using the command “**git remote add origin <repo URL>**”, the GitHub URL can be added to our local git repository. “**Git commit -m "a meaningful commit message"** “can be used to create a commit, and git push origin main can be used to submit local filesystem changes to GitHub.



Commits on Mar 22, 2023

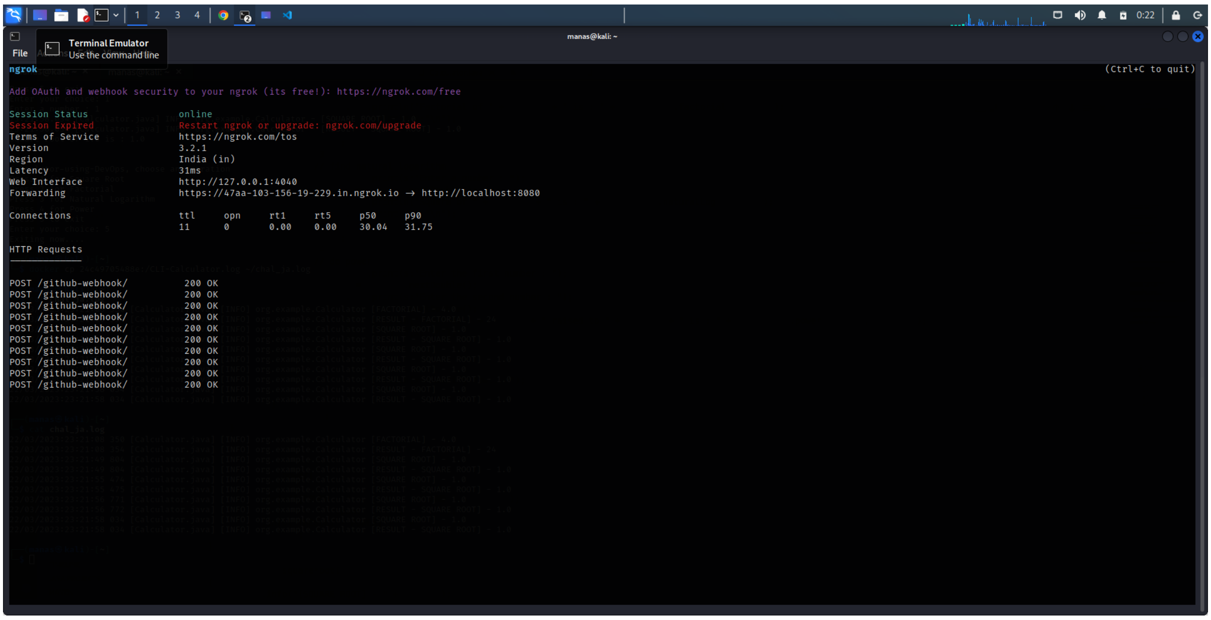
Delete CLI-Calculator.log Inazuma1002 committed 4 hours ago	Verified	5ee5514
Update Jenkinsfile Inazuma1002 committed 4 hours ago	Verified	e92628f
added imge push and cleanup stages Inazuma1002 committed 4 hours ago	Verified	6503e20
added buid stage Inazuma1002 committed 4 hours ago	Verified	4cf9403
Update Jenkinsfile Inazuma1002 committed 4 hours ago	Verified	519e9c6
Update README.md Inazuma1002 committed 4 hours ago	Verified	060c41c
Create Jenkinsfile Inazuma1002 committed 4 hours ago	Verified	3440c06
Update README.md Inazuma1002 committed 4 hours ago	Verified	bc9860b
Update README.md Inazuma1002 committed 5 hours ago	Verified	39771df
Update README.md Inazuma1002 committed 5 hours ago	Verified	f42d2e5
Create README.md Inazuma1002 committed 5 hours ago	Verified	10e30e4

A view of my commits

9. Ngrok

Using the utility ngrok, programmers can make a local web server accessible to the internet. It enables testing and sharing of applications without the need for a public-facing server by establishing a secure tunnel between the local computer and a public URL that can be viewed from anywhere. A temporary public URL is made available by ngrok when it is running so that users can reach the local server. The program is created can be viewed and used by others by sharing this URL with them. Since the ngrok tunnel uses HTTPS encryption, all communication between the local computer and the public URL is secure. During development and testing, developers can use ngrok to receive webhooks from external services.

Start Ngrok by entering the command “**ngrok http <web port-number>**”. The port number of the local web server we want to disclose should be substituted for <port-number>. A public URL created by Ngrok can be used to view our local web server from anywhere. The URL can be found in the "Forwarding" portion of the Ngrok terminal window. To try it, copy the URL and enter it in our web browser. The Ngrok public URL can now be used to reach the local web server from the internet. Remember that closing the terminal window or terminating the Ngrok process will terminate the current Ngrok session.



```
ngrok                                                                                    (Ctrl+C to quit)

Add OAuth and webhook security to your ngrok (its free!): https://ngrok.com/free

Session Status                online
Session Expired               Restart ngrok or upgrade: ngrok.com/upgrade
Terms of Service              https://ngrok.com/tos
Version                       3.2.1
Region                        India (in)
Latency                       31ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://47aa-103-156-19-229.in.ngrok.io -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              11      0       0.00    0.00    30.04   31.75

HTTP Requests
-------------

POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
```

10. Webhooks

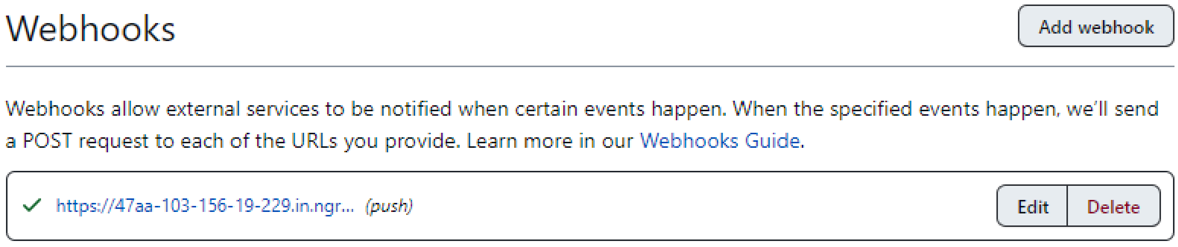
Jenkins can use GitHub webhooks to start builds or carry out other tasks in reaction to GitHub repository events like a new pull request being created or a code push to a particular branch.

The procedures below should be followed to configure Jenkins to use GitHub webhooks:

1) Jenkins can communicate with GitHub repositories and react to webhook events if the GitHub plugin is installed.
2) Create a new Jenkins job or configure an existing one to respond to a webhook event. Select the “GitHub hook trigger for GITScm polling” option in the Build Triggers section of the job configuration
3) In the GitHub repository settings, add a new webhook and specify the URL of the Jenkins server and the payload URL for the specific webhook event. You can choose to trigger the webhook event for all pushes, pull requests, or specific branches or tags.
4) Save the webhook configuration and test the integration by triggering a webhook event in the GitHub repository. Jenkins should receive the event and trigger a build or perform the specified action.

Jenkins can use GitHub webhooks to automate a number of steps in the development process, including creating and testing code, deploying apps, and informing team members of new events. We can automate development and minimize tedious work by using webhooks.

Choosing the webhooks options on the GitHub repository.



Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

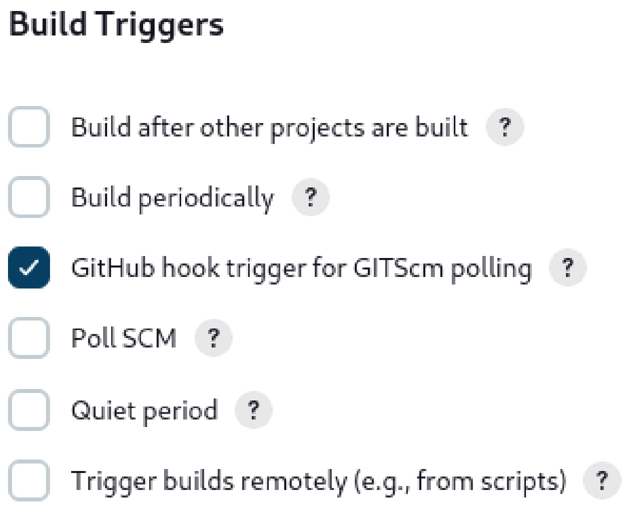
✓ https://47aa-103-156-19-229.in.ngr... *(push)*

Several different build trigger types are available in Jenkins to initiate a build procedure automatically:

1) **Polling SCM**: This trigger periodically scans the source code file for modifications and launches a build if any are found.
2) **Schedule**: This trigger initiates a build at a predetermined moment or periodically.

3) **Build after other projects are built**: This trigger initiates a build following the successful completion of one or more of the named projects.
4) **Remote build triggers**: With this trigger, remote scripts or programs can launch a build by making an API or URL request.
5) **Build periodically**: This trigger starts a build at regular intervals, such as every hour or every day
6) **GitHub hook trigger for GITScm polling**: This trigger allows Jenkins to listen for incoming webhooks from GitHub and start a build when changes are pushed to the repository.
7) **Pipeline trigger**: This trigger allows one pipeline to trigger another pipeline.

The Jenkins project settings allow for the configuration of these triggers, and a build can be started based on a variety of circumstances by combining several triggers.



Build Triggers

□ Build after other projects are built ?

□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

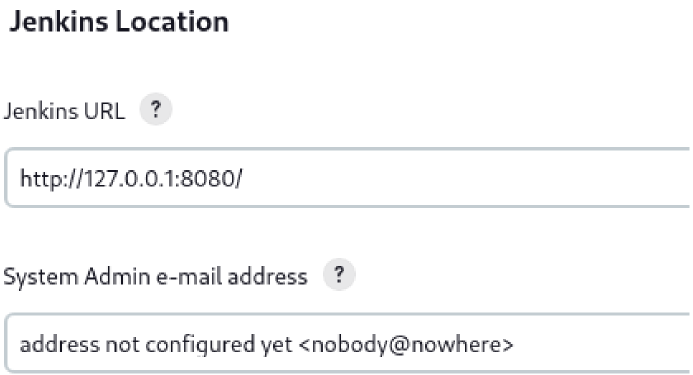
□ Poll SCM ?

□ Quiet period ?

□ Trigger builds remotely (e.g., from scripts) ?

Choosing the GitSCM polling mechanism in Jenkins

Adding the ngrok forwarding URL to Jenkins



Jenkins Location

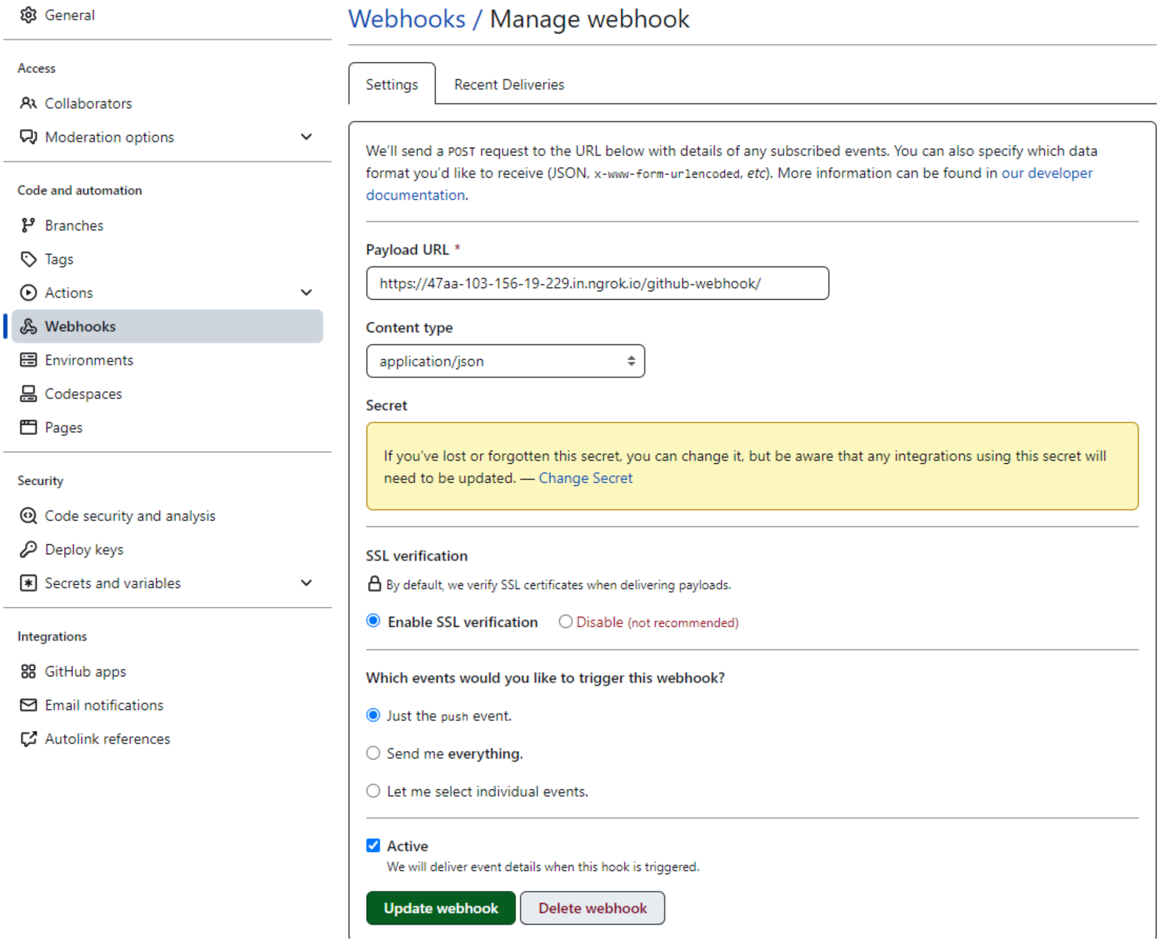
Jenkins URL ?

http://127.0.0.1:8080/

System Admin e-mail address ?

address not configured yet <nobody@nowhere>

Adding the ngrok forwarding URL to the GitHub repository



Webhooks / Manage webhook

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://47aa-103-156-19-229.in.ngrok.io/github-webhook/

Content type

application/json

Secret

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. — Change Secret

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.

○ Send me everything.

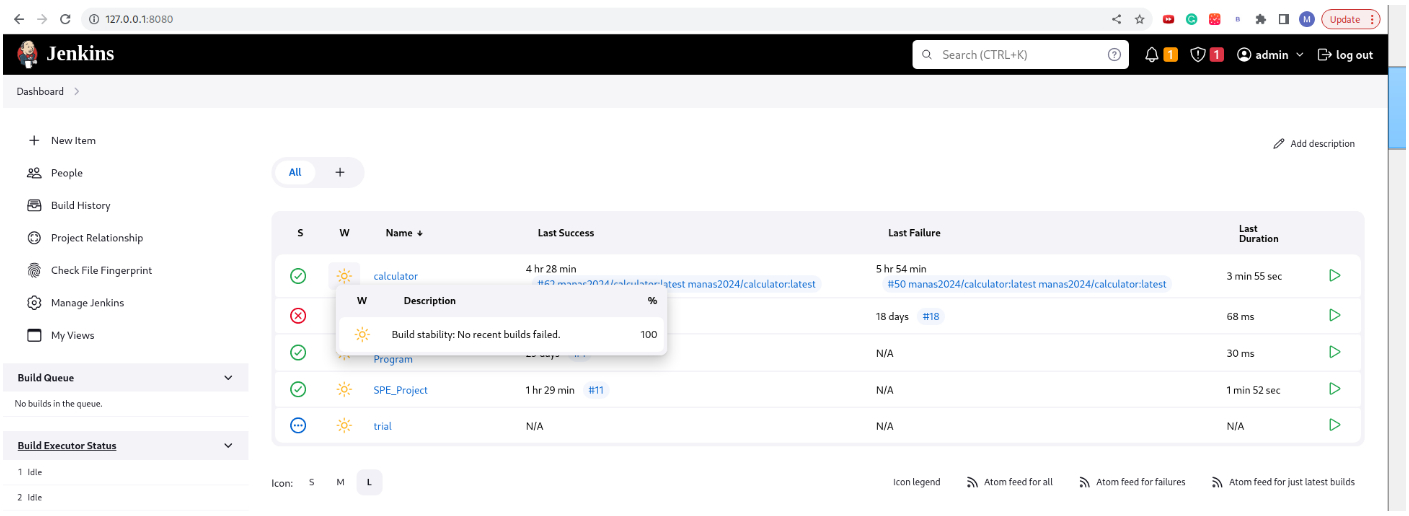
○ Let me select individual events.

☑ Active
We will deliver event details when this hook is triggered.

11. Continuous Integration Using Jenkins

Developers can create, test, and distribute software using Jenkins, an open-source automation server. It was initially created as a continuous integration (CI) tool for Java applications, but it has since expanded to handle a number of different platforms and programming languages. Jenkins automates every step of the software development process, including writing code, testing it, and delivering the finished product. For managing and configuring jobs, the separate activities that make up the development pipeline, it offers a user-friendly web interface.
These methods can be used to install Jenkins.

- ➢ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
- ➢ sudo sh -c ’echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list
- ➢ sudo apt install ca-certificates
- ➢ sudo apt-get update
- ➢ sudo apt-get install jenkins



No builds in the queue.

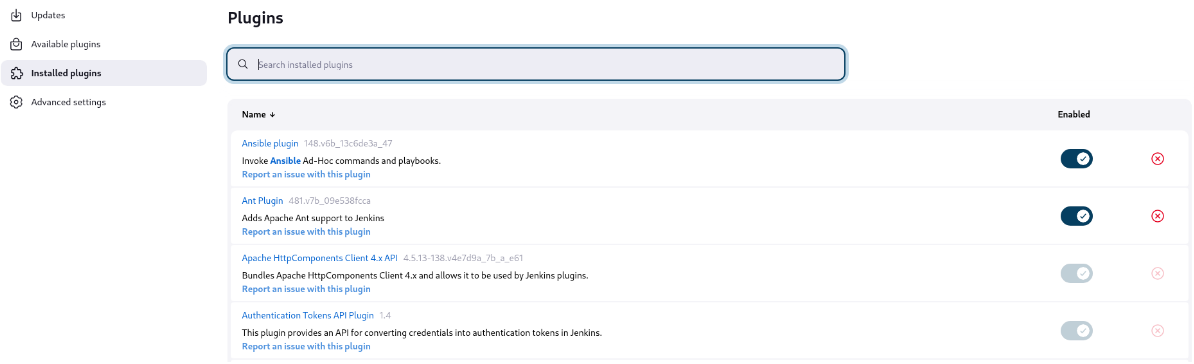
1 Idle
2 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	calculator	4 hr 28 min #62 manas2024/calculator:latest manas2024/calculator:latest	5 hr 54 min #50 manas2024/calculator:latest manas2024/calculator:latest	3 min 55 sec
✗				18 days #18	68 ms
✓	☀	Program	[illegible]	N/A	30 ms
✓	☀	SPE_Project	1 hr 29 min #11	N/A	1 min 52 sec
⋯	☀	trial	N/A	N/A	N/A

W	Description	%
☀	Build stability: No recent builds failed.	100

11.1 Plugins

Jenkins's key feature, plugins, enables programmers to increase the software's capability and adapt it to their particular requirements. Jenkins has a large plugin collection with over 1,500 plugins, and new plugins are constantly being created and added to the library. The plugins which we need for this project are the ones for Git, Docker, Ansible and Maven. Plugins can be installed from the “Plugin Manager” under “Manage Jenkins”.

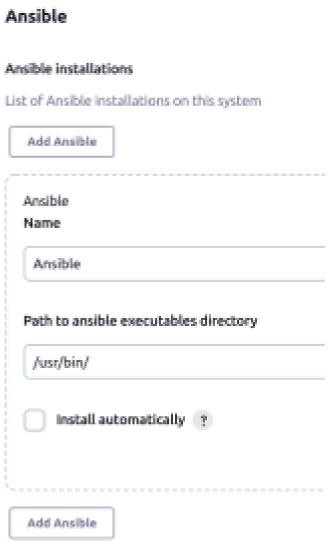


Plugins

Name ↓	Enabled
Ansible plugin 148.v6b_13c6de3a_47 Invoke Ansible Ad-Hoc commands and playbooks. Report an issue with this plugin	✓
Ant Plugin 481.v7b_09e538fcca Adds Apache Ant support to Jenkins Report an issue with this plugin	✓
Apache HttpComponents Client 4.x API 4.5.13-138.v4e7d9a_7b_a_e61 Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. Report an issue with this plugin	✓
Authentication Tokens API Plugin 1.4 This plugin provides an API for converting credentials into authentication tokens in Jenkins. Report an issue with this plugin	✓

11.2 Global Tool Configuration

We must define the configuration for Maven, Git, and Ansible under the Global Tool Configuration options in Manage Jenkins. Despite the fact that they are almost always automatically filled, it is a good notion to check.



Ansible

Ansible installations

List of Ansible installations on this system

Ansible

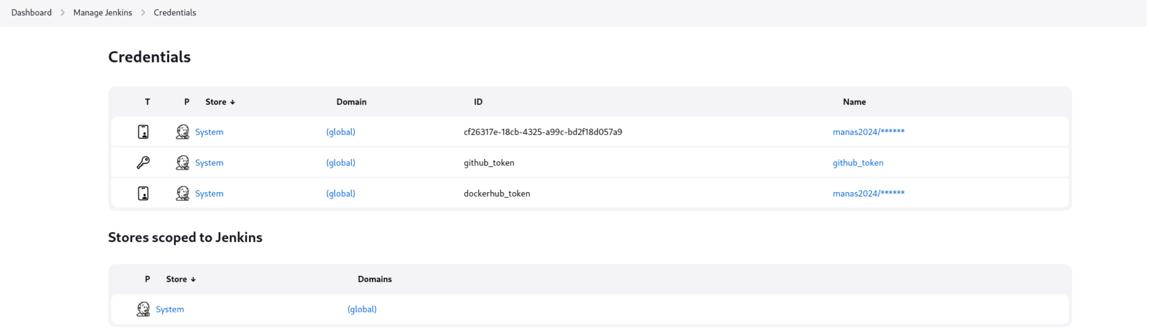
Name: Ansible

Path to ansible executables directory: /usr/bin/

□ Install automatically ?

11.3 Manage Credentials

Under the Manage Credentials options in Manage Jenkins, we have to specify the login credentials for Docker Hub and GitHub.



Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	cf26317e-18cb-4325-a99c-bd2f18d057a9	manas2024/******
		System	(global)	github_token	github_token
		System	(global)	dockerhub_token	manas2024/******

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

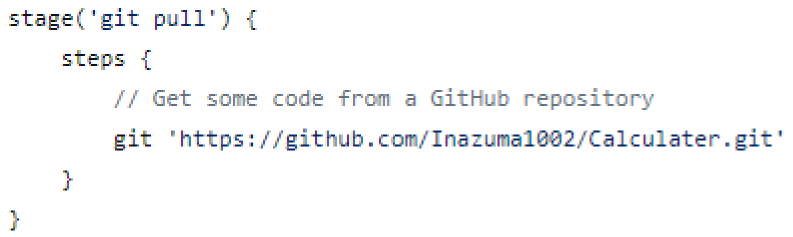
11.4 Jenkins Pipeline

A pipeline in Jenkins is a method to specify and automate the steps necessary to develop, test, and release software. A Jenkinsfile, a text file that lists the stages, actions, and configurations necessary to carry out a continuous delivery pipeline, is usually where a pipeline is defined.

The Jenkins web interface and the Jenkinsfile itself both offer configuration and administration options for Jenkins pipelines, enabling version-controlled and automated pipeline management.

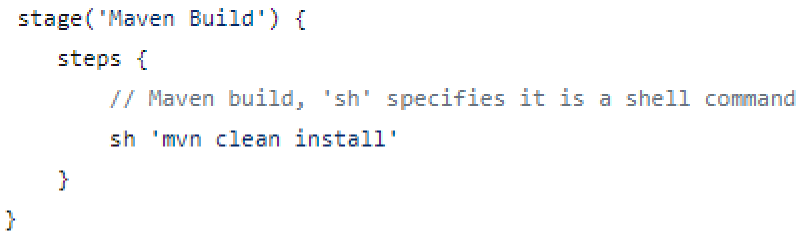
Our workflow script's various phases are:

1) **Git Pull:** Pull the codebase from a remote repository hosted on GitHub.



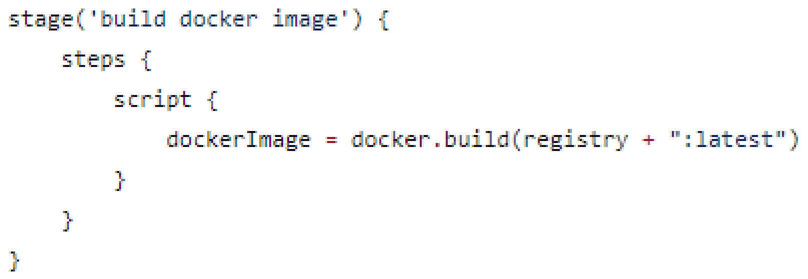
```
stage('git pull') {
    steps {
        // Get some code from a GitHub repository
        git 'https://github.com/Inazuma1002/Calculater.git'
    }
}
```

2) **Maven Build**: It generates a jar file containing all of the requirements for our source code. When the initial target folder containing the earlier dependencies is destroyed, a new target folder with the new jar file will be generated.



```
stage('Maven Build') {
   steps {
       // Maven build, 'sh' specifies it is a shell command
       sh 'mvn clean install'
   }
}
```

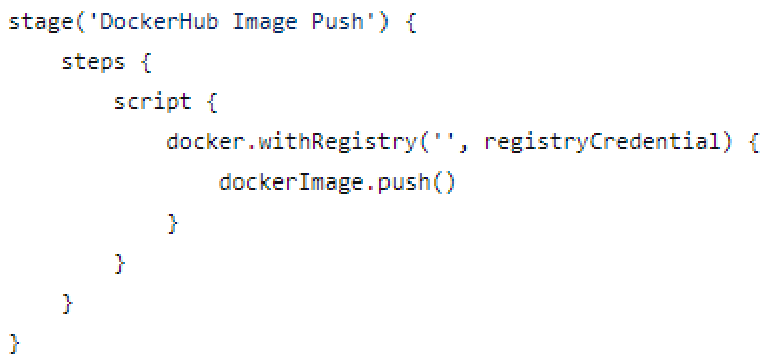
3) **Create Docker Images**: This command is used to produce images on our local system that are then uploaded to our Docker Hub, enabling us to retrieve the image and run the program on other systems. 'latest' is the tag assigned to this image.



```
stage('build docker image') {
    steps {
        script {
            dockerImage = docker.build(registry + ":latest")
        }
    }
}
```

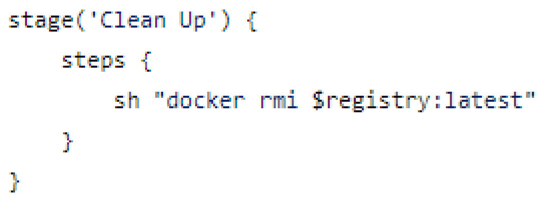
4) **Publish Docker Image**: At this point, the image has been uploaded to our Docker Hub so that anyone can obtain it. The command "sudo

chmod 666 /var/run/docker.sock" on the localhost must be used to give permissions.



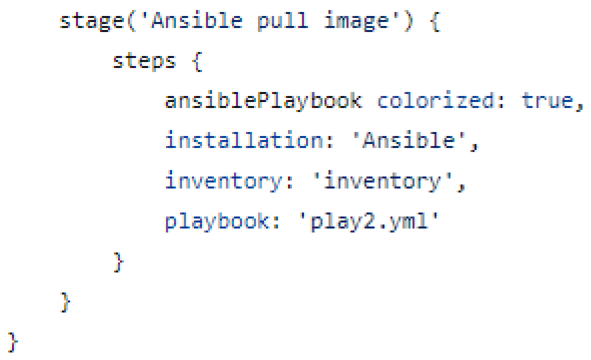
```
stage('DockerHub Image Push') {
    steps {
        script {
            docker.withRegistry('', registryCredential) {
                dockerImage.push()
            }
        }
    }
}
```

5) **Clean Docker Image**: In this step, we remove the Docker image from our local system.



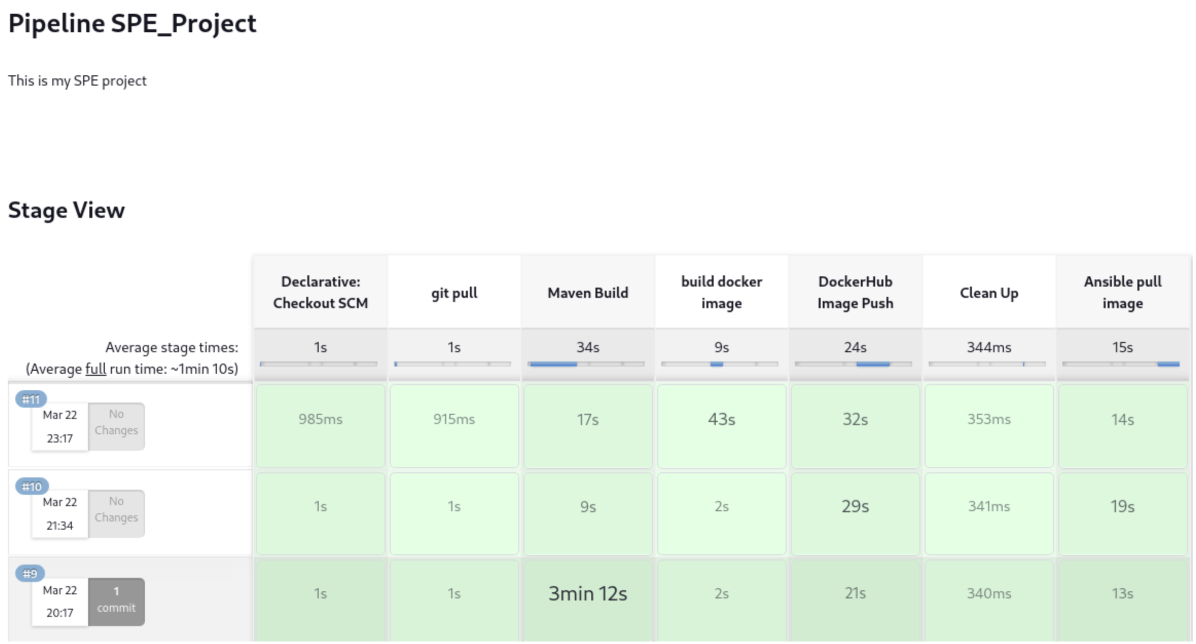
```
stage('Clean Up') {
    steps {
        sh "docker rmi $registry:latest"
    }
}
```

6) **Ansible Deploy**: Fetching the image from the Docker Hub and running it on the hosts specified in the “inventory” file.



```
    stage('Ansible pull image') {
        steps {
            ansiblePlaybook colorized: true,
            installation: 'Ansible',
            inventory: 'inventory',
            playbook: 'play2.yml'
        }
    }
}
```

11.5 Jenkins Run



Pipeline SPE_Project

This is my SPE project

Stage View

	Declarative: Checkout SCM	git pull	Maven Build	build docker image	DockerHub Image Push	Clean Up	Ansible pull image
Average stage times: (Average full run time: ~1min 10s)	1s	1s	34s	9s	24s	344ms	15s
#11 Mar 22 23:17 No Changes	985ms	915ms	17s	43s	32s	353ms	14s
#10 Mar 22 21:34 No Changes	1s	1s	9s	2s	29s	341ms	19s
#9 Mar 22 20:17 1 commit	1s	1s	3min 12s	2s	21s	340ms	13s

The stage view on the Jenkins dashboard

12. Ansible Deployment

Ansible is a free and open-source automation tool for managing and configuring computer networks. Users can automate repetitive chores like software installations, configuration management, and application deployment with its straightforward, agentless architecture. The Docker image is downloaded by Ansible from the Docker Hub and installed on numerous computers.

12.1 Dockerfile

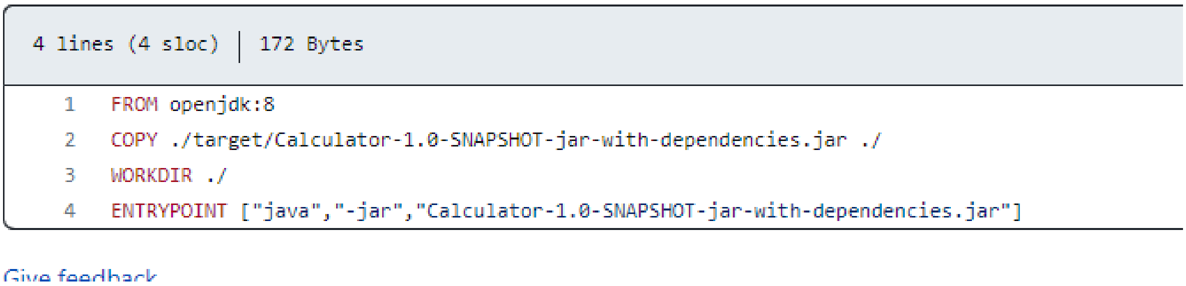
A text file called a Dockerfile holds a collection of instructions for creating a Docker image. The base image to use, any additional packages or software that must be installed, configuration settings, and any other actions that must be done to prepare the image are usually all described in a Dockerfile. Developers can automate the process of creating Docker images by using a Dockerfile, which makes it simpler to keep consistency across various environments and platforms. It is simpler to work together on developing and deploying apps in a Dockerized environment when Dockerfiles are version-controlled and shared among team members.

Using the docker build command, a Dockerfile can be used to construct a Docker image. The resulting image can then be used to launch containers with the stated configurations and software.

The following are the main fields of a Dockerfile:

1. FROM: Specifies the base image that the new image will be built upon.
2. RUN: Runs a command inside the container while building the image.
3. COPY or ADD: Copies files or directories from the host system into the container.
4. WORKDIR: Sets the working directory for any RUN, CMD, ENTRYPOINT, COPY, and ADD instructions that follow.
5. EXPOSE: Informs Docker that the container will listen on the specified network ports at runtime.
6. CMD or ENTRYPOINT: Specifies the command to be executed when the container starts.
7. ENV: Sets environment variables that will be available in the container.
8. ARG: Defines arguments that can be passed to the FROM, RUN, and CMD instructions.
9. LABEL: Adds metadata to the image.
10. USER: Sets the user that will run the subsequent CMD, ENTRYPOINT, COPY, and ADD instructions.
11. VOLUME: Creates a mount point for external storage volumes

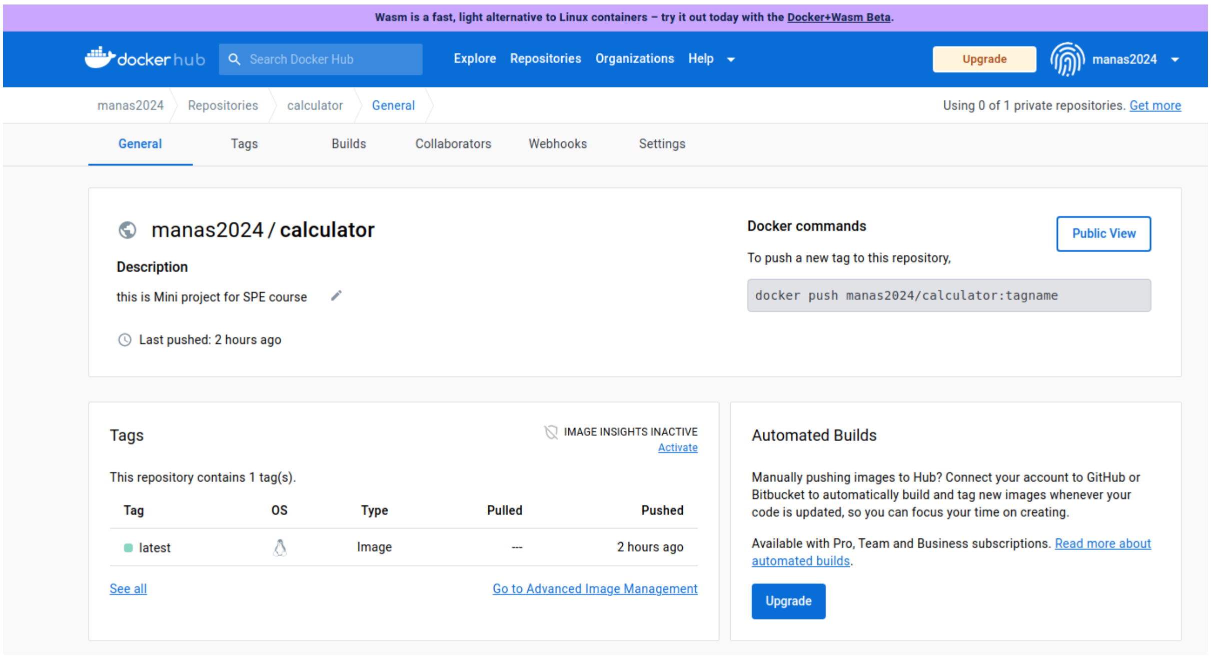
You can specify how the image should be created, what commands should be run inside the container, and how it should act when it is run using these fields.



```
FROM openjdk:8
COPY ./target/Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar ./
WORKDIR ./
ENTRYPOINT ["java","-jar","Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar"]
```

12.2 DockerHub

Pushing the generated image to Docker Hub. Developers can keep and share their Docker container images with other developers or users using Docker Hub, a cloud-based repository. In essence, it serves as a central repository for Docker images, allowing programmers to browse, download, and exchange pre-built Docker images that can be used to run applications in containers. The image can be found at this URL: "https://hub.docker.com/repository/docker/manas2024/calculator/general"



Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta.

manas2024 / calculator

Description

this is Mini project for SPE course

Last pushed: 2 hours ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push manas2024/calculator:tagname
```

Tags

IMAGE INSIGHTS INACTIVE Activate

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	🐧	Image	---	2 hours ago

See all

Go to Advanced Image Management

Automated Builds

Manually pushing images to 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.

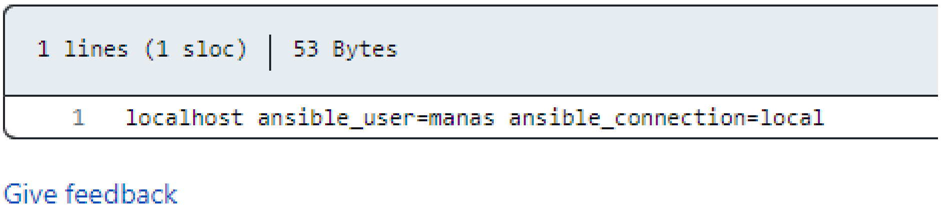
12.3 Ansible Installation

The following steps have to be followed for installing Ansible:

1) sudo apt install openssh-server
2) service start ssh
3) ssh-keygen -t rsa
4) ssh-copy-id username@<IP>
5) sudo apt update
6) sudo apt install ansible
7) ansible –version

12.4 Inventory

An inventory file in Ansible is a configuration file that lists the servers and host groups that Ansible will control. Ansible will connect to and perform actions on the hostnames or IP addresses listed in the inventory file.

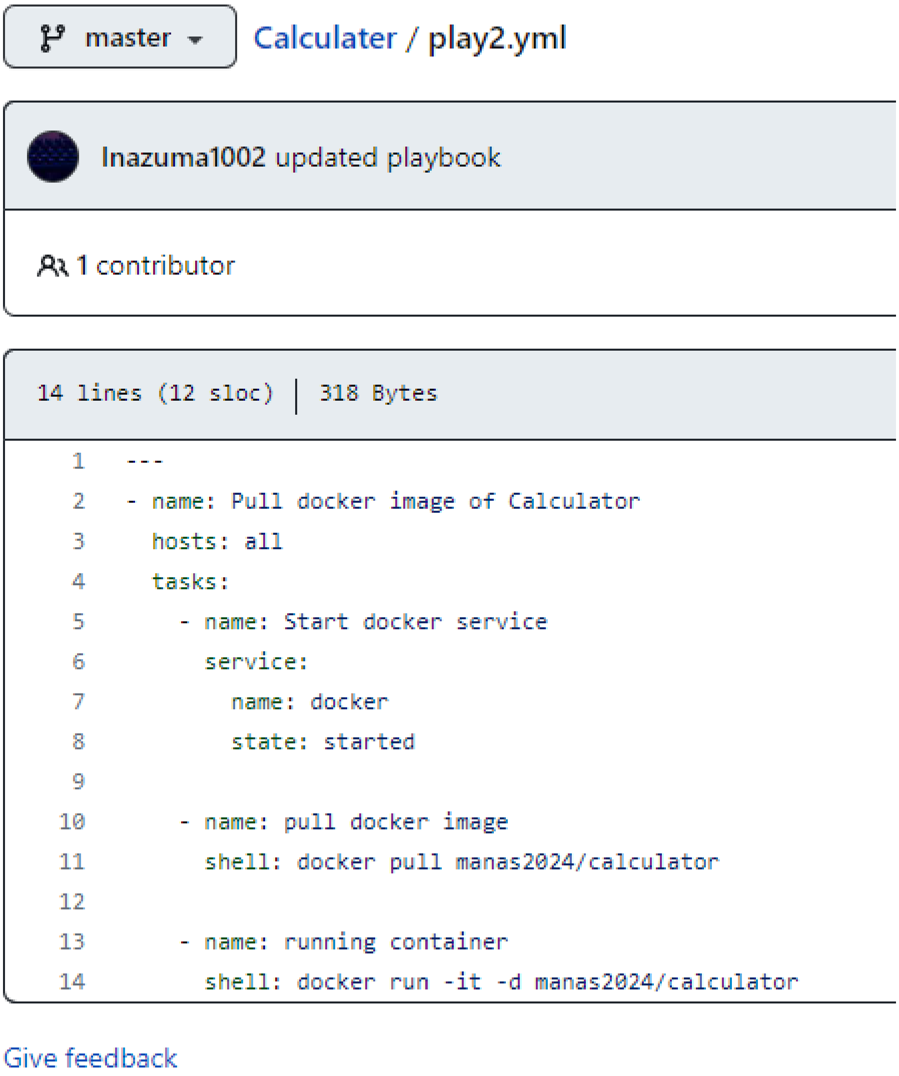


1 lines (1 sloc) | 53 Bytes

```
localhost ansible_user=manas ansible_connection=local
```

12.5 Playbook

A playbook in Ansible is a file or collection of files that specify a series of actions to be carried out on one or more hosts. To automate difficult chores like configuration management, application deployment, and orchestration, playbooks are written in YAML format. One or more "plays"—a collection of activities that are carried out on a number of hosts—make up a playbook. Each play lists the hosts to which it pertains as well as any variables, tasks, and handlers needed to carry out the intended operations.



master Calculater / play2.yml

Inazuma1002 updated playbook

1 contributor

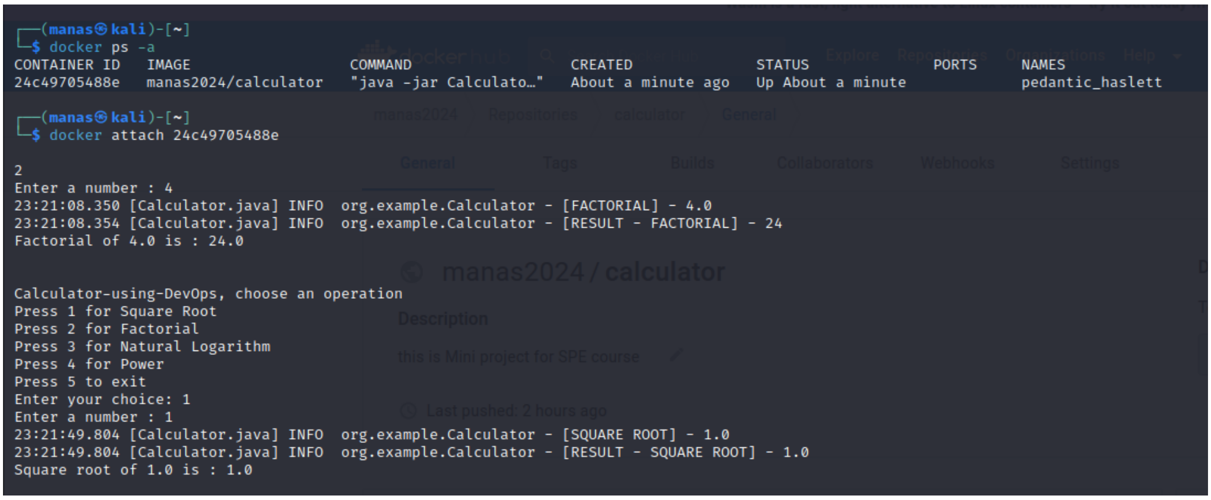
14 lines (12 sloc) | 318 Bytes

```
---
- name: Pull docker image of Calculator
  hosts: all
  tasks:
    - name: Start docker service
      service:
        name: docker
        state: started

    - name: pull docker image
      shell: docker pull manas2024/calculator

    - name: running container
      shell: docker run -it -d manas2024/calculator
```

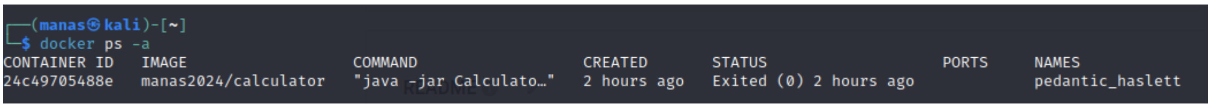
Give feedback

12.6 Ansible Run

Through the playbook, Ansible pulls the image from the Docker Hub and spwans a container on the target configuration(s), in this case, the localhost.



```
┌──(manas㉿kali)-[~]
└─$ docker ps -a
CONTAINER ID   IMAGE                 COMMAND                  CREATED              STATUS              PORTS     NAMES
24c49705488e   manas2024/calculator  "java -jar Calculato…"   About a minute ago   Up About a minute             pedantic_haslett

┌──(manas㉿kali)-[~]
└─$ docker attach 24c49705488e

2
Enter a number : 4
23:21:08.350 [Calculator.java] INFO  org.example.Calculator - [FACTORIAL] - 4.0
23:21:08.354 [Calculator.java] INFO  org.example.Calculator - [RESULT - FACTORIAL] - 24
Factorial of 4.0 is : 24.0


Calculator-using-DevOps, choose an operation
Press 1 for Square Root
Press 2 for Factorial
Press 3 for Natural Logarithm
Press 4 for Power
Press 5 to exit
Enter your choice: 1
Enter a number : 1
23:21:49.804 [Calculator.java] INFO  org.example.Calculator - [SQUARE ROOT] - 1.0
23:21:49.804 [Calculator.java] INFO  org.example.Calculator - [RESULT - SQUARE ROOT] - 1.0
Square root of 1.0 is : 1.0
```



```
┌──(manas㉿kali)-[~]
└─$ docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS                     PORTS     NAMES
24c49705488e   manas2024/calculator   "java -jar Calculato…"   2 hours ago     Exited (0) 2 hours ago               pedantic_haslett
```

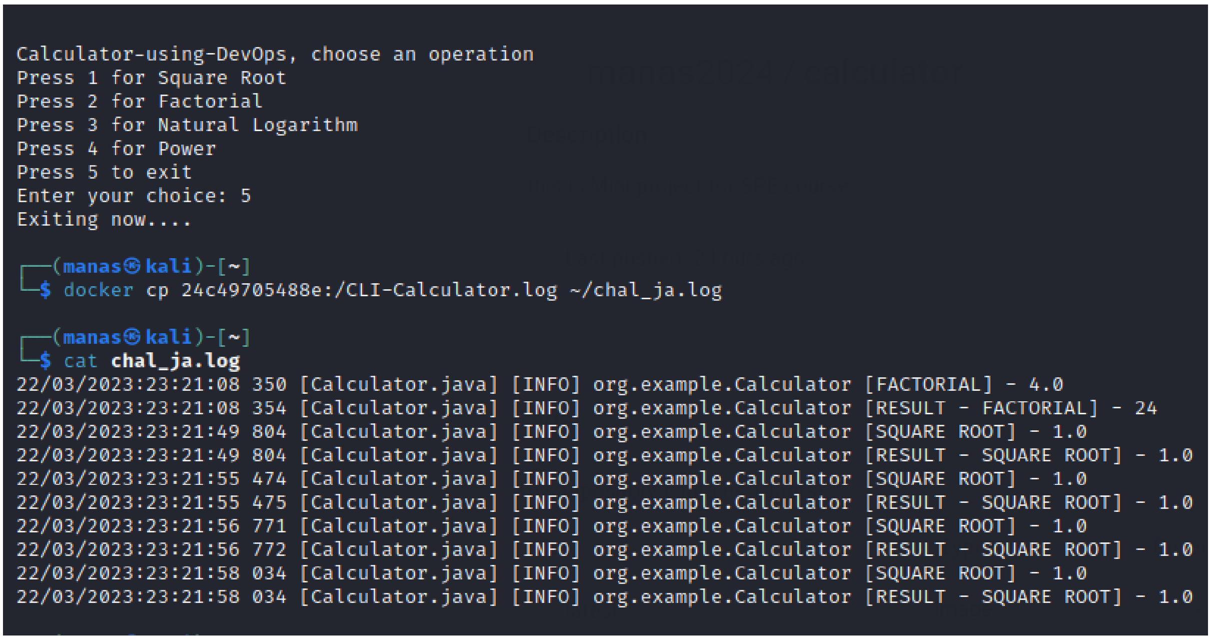
the jar file can be run by the command docker attach<CONTAINER ID>.The image shows the execution of the calculator program.

13 Continous Monitoring

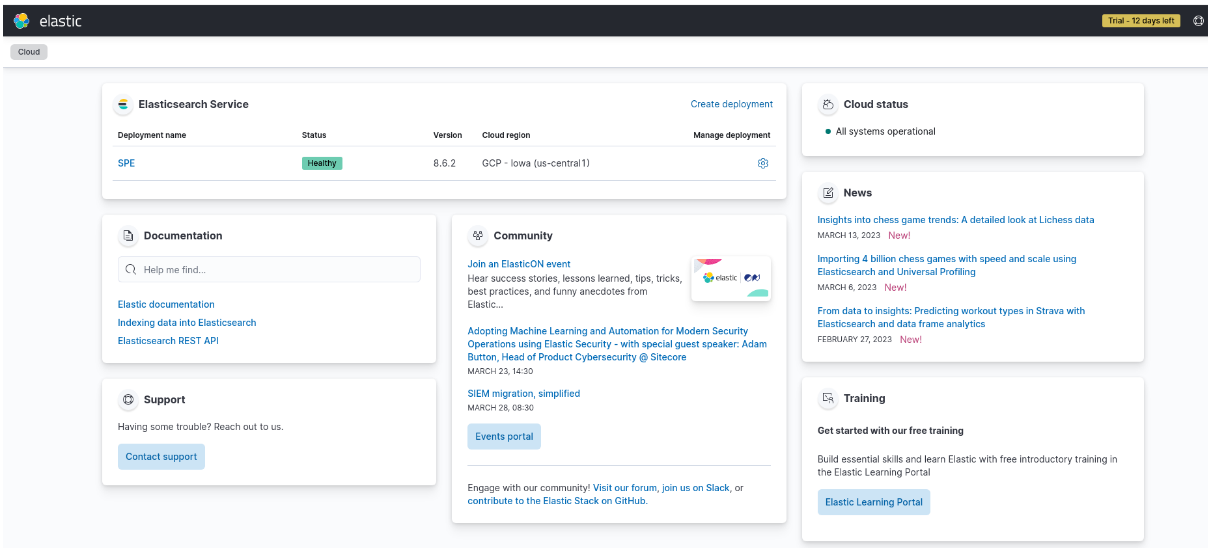
A group of open-source software programs known as the ELK stack are frequently used for log handling and analysis. Elasticsearch, Logstash, and Kibana are collectively referred to by the abbreviation ELK. Fast and effective full-text search skills are made possible by Elasticsearch, a distributed search and analytics engine. It offers sophisticated querying capabilities and is used to keep and index massive amounts of data in real-time.
Logstash: Data from different sources is gathered, processed, and transformed using a pipeline called Logstash. It is frequently employed to gather log data from various sources and parse it into a structured file for easy analysis.
Kibana is a data visualization tool that is used to create interactive dashboards and visualizations from data stored in Elasticsearch. It allows users to easily explore and analyze data using a web-based interface, and provides a range of visualization options, including charts, graphs, and maps.
Together, the ELK stack provides a powerful and flexible platform for log management and analysis, and is widely used in a variety of industries, including IT, security, and business intelligence.

The log file can be obtained by running:

1) docker start <CONTAINER ID>
2) docker exec -it <CONTAINER ID> /bin/bash
3) cat calculator.log



```
Calculator-using-DevOps, choose an operation
Press 1 for Square Root
Press 2 for Factorial
Press 3 for Natural Logarithm
Press 4 for Power
Press 5 to exit
Enter your choice: 5
Exiting now....

┌──(manas㉿kali)-[~]
└─$ docker cp 24c49705488e:/CLI-Calculator.log ~/chal_ja.log

┌──(manas㉿kali)-[~]
└─$ cat chal_ja.log
22/03/2023:23:21:08 350 [Calculator.java] [INFO] org.example.Calculator [FACTORIAL] - 4.0
22/03/2023:23:21:08 354 [Calculator.java] [INFO] org.example.Calculator [RESULT - FACTORIAL] - 24
22/03/2023:23:21:49 804 [Calculator.java] [INFO] org.example.Calculator [SQUARE ROOT] - 1.0
22/03/2023:23:21:49 804 [Calculator.java] [INFO] org.example.Calculator [RESULT - SQUARE ROOT] - 1.0
22/03/2023:23:21:55 474 [Calculator.java] [INFO] org.example.Calculator [SQUARE ROOT] - 1.0
22/03/2023:23:21:55 475 [Calculator.java] [INFO] org.example.Calculator [RESULT - SQUARE ROOT] - 1.0
22/03/2023:23:21:56 771 [Calculator.java] [INFO] org.example.Calculator [SQUARE ROOT] - 1.0
22/03/2023:23:21:56 772 [Calculator.java] [INFO] org.example.Calculator [RESULT - SQUARE ROOT] - 1.0
22/03/2023:23:21:58 034 [Calculator.java] [INFO] org.example.Calculator [SQUARE ROOT] - 1.0
22/03/2023:23:21:58 034 [Calculator.java] [INFO] org.example.Calculator [RESULT - SQUARE ROOT] - 1.0
```



Deployment name	Status	Version	Cloud region	Manage deployment
SPE	Healthy	8.6.2	GCP - Iowa (us-central1)	⚙

● All systems operational

Join an ElasticON event
Hear success stories, lessons learned, tips, tricks, best practices, and funny anecdotes from Elastic...

Adopting Machine Learning and Automation for Modern Security Operations using Elastic Security - with special guest speaker: Adam Button, Head of Product Cybersecurity @ Sitecore
MARCH 23, 14:30

SIEM migration, simplified
MARCH 28, 08:30

Engage with our community! Visit our forum, join us on Slack, or contribute to the Elastic Stack on GitHub.

Insights into chess game trends: A detailed look at Lichess data
MARCH 13, 2023 New!

Importing 4 billion chess games with speed and scale using Elasticsearch and Universal Profiling
MARCH 6, 2023 New!

From data to insights: Predicting workout types in Strava with Elasticsearch and data frame analytics
FEBRUARY 27, 2023 New!

Having some trouble? Reach out to us.

Get started with our free training

Build essential skills and learn Elastic with free introductory training in the Elastic Learning Portal