Software Production Engineering

CS 816

Mini Project: DevOps-Scientific Calculator

- Saad Patel (IMT2018514)
Mohammad.Saad@iiitb.ac.in

Problem Statement:

To create a scientific calculator program with user menu-driven operations

- Square root function - √x
- Factorial function - x!
- Natural logarithm (base e) - ln(x)
- Power function - x^b

I have also added some basic calculator functions like addition subtraction multiplication and division along with the above scientific functions.

DevOps tool chain

What and Why: DevOps is a set of philosophies, practices, and tools that help an organization deliver better products faster by facilitating the integration of the development and operations functions. It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a Solution. DevOps is part of the Agile Product Delivery competency of the Lean Enterprise. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches. And, it's gaining popularity at a rapid rate.

I used the following set of DevOps toolchains, The pipeline includes

1. Source Control Management Tool - **GitHub**
2. Testing - tested code using **JUnit**
3. Build - built code using tool **Maven**
4. Continuous Integration - Continuous integrated code using tool **Jenkins**
5. Containerize - Containerized code using **Docker**.
6. Pushed created Docker image to my **Docker hub**.
7. Deployment - Did configuration management and deployment using **Ansible**. Using Ansible I did configuration management and pulled the docker image and ran it on the managed hosts.
8. I deployed on the **local** machine.
9. Monitoring - for monitoring I used the **ELK stack**. Used - Elasticsearch, Logstash and Kibana were used to do Monitoring. Generated the log file for my mini project and passed it to my ELK stack.

Github: Source control management tool

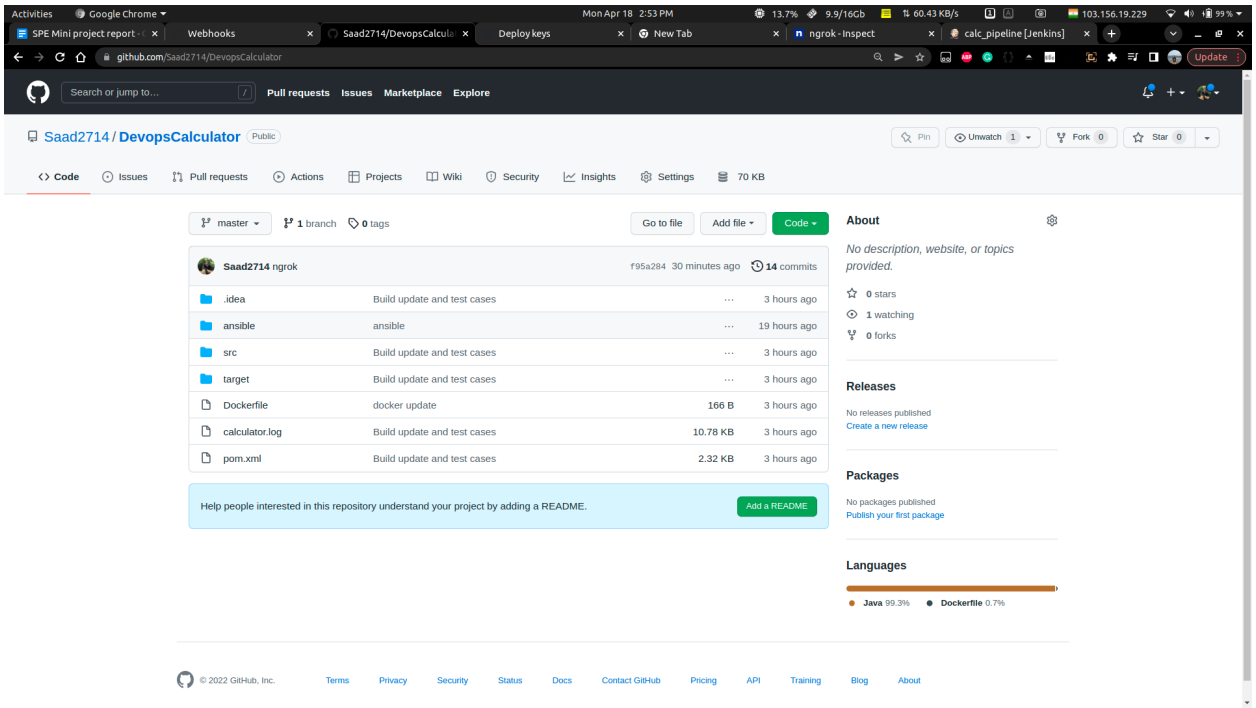
Source code management (SCM) is synonymous with Version control. It is a software tool that programmers use to manage source code. It tracks modifications to a source code repository and helps deal with merge conflicts. SCM has a lot of useful features that can make your work even more effective and more manageable.

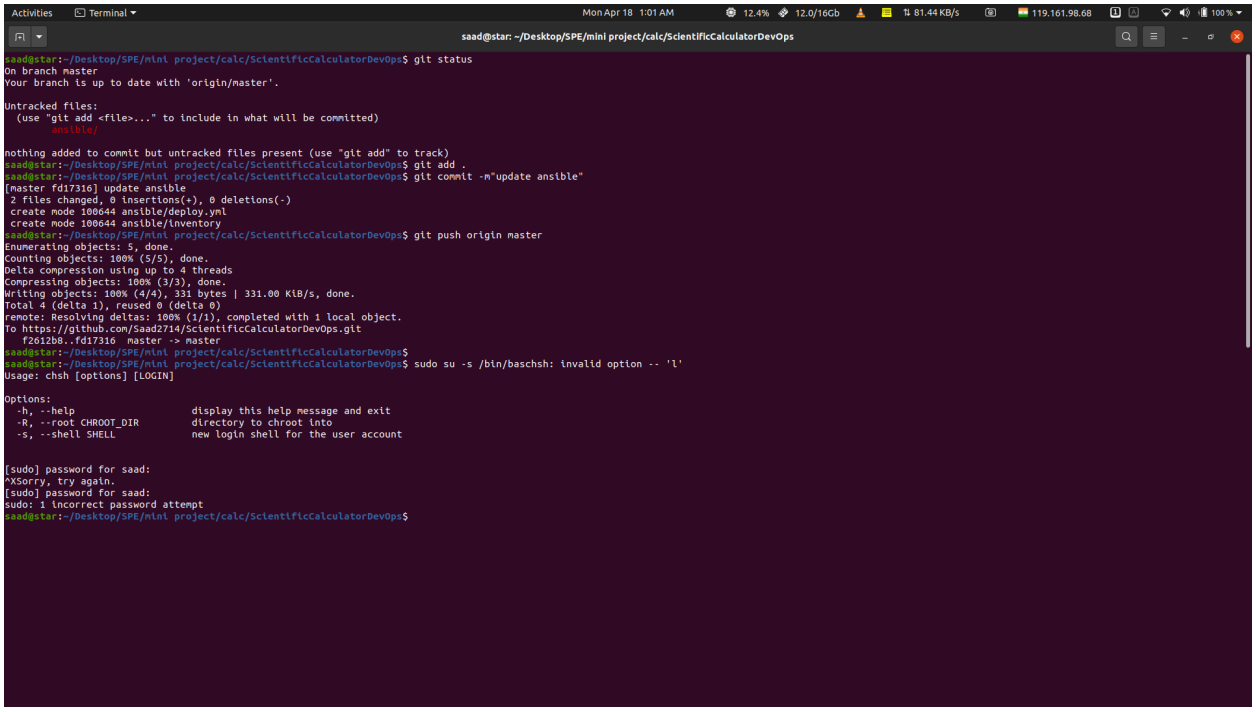
SCM is used for tracking the changes over time, thus creating a historical record, which can be used to find out where the bugs come from, compare the older versions and even undo some changes to the code base. Besides, it also archives these changes giving a cleaner look to the history log.

With the help of SCM, each developer works independently on a separate branch and once the work is done, all the branches are merged together.

GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. GitHub is an online software development platform used for storing, tracking, and collaborating on software projects. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects.

Github Repo Link: **https://github.com/Saad2714/DevopsCalculator**


Activities
Google Chrome
Mon Apr 18 2:53 PM
SPE Mini project report
Webhooks
Saad2714/DevopsCalcula
Deploy keys
New Tab
ngrok - Inspect
calc_pipeline [Jenkins]
github.com/Saad2714/DevopsCalculator
Update
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Saad2714 / DevopsCalculator
Public
Pin
Unwatch 1
Fork 0
Star 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
70 KB
master
1 branch
0 tags
Go to file
Add file
Code
Saad2714 ngrok
f95a284 30 minutes ago
14 commits
.idea
Build update and test cases
3 hours ago
ansible
ansible
19 hours ago
src
Build update and test cases
3 hours ago
target
Build update and test cases
3 hours ago
Dockerfile
docker update
166 B
3 hours ago
calculator.log
Build update and test cases
10.78 KB
3 hours ago
pom.xml
Build update and test cases
2.32 KB
3 hours ago
Help people interested in this repository understand your project by adding a README.
Add a README
About
No description, website, or topics provided.
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Java 99.3%
Dockerfile 0.7%
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About


Activities
Terminal
Mon Apr 18 1:01 AM
saad@star: ~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ git status
On branch master
Your branch is up to date with 'origin/master'.
Untracked files:
(use "git add <file>..." to include in what will be committed)
ansible/
nothing added to commit but untracked files present (use "git add" to track)
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ git add .
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ git commit -m"update ansible"
[master fd17316] update ansible
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ansible/deploy.yml
create mode 100644 ansible/inventory
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 331 bytes | 331.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Saad2714/ScientificCalculatorDevOps.git
f2612b8..fd17316 master -> master
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ sudo su -s /bin/baschsh: invalid option -- 'l'
Usage: chsh [options] [LOGIN]
Options:
-h, --help display this help message and exit
-R, --root CHROOT_DIR directory to chroot into
-s, --shell SHELL new login shell for the user account
[sudo] password for saad:
^XSorry, try again.
[sudo] password for saad:
sudo: 1 incorrect password attempt
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$

Maven Test and Build:

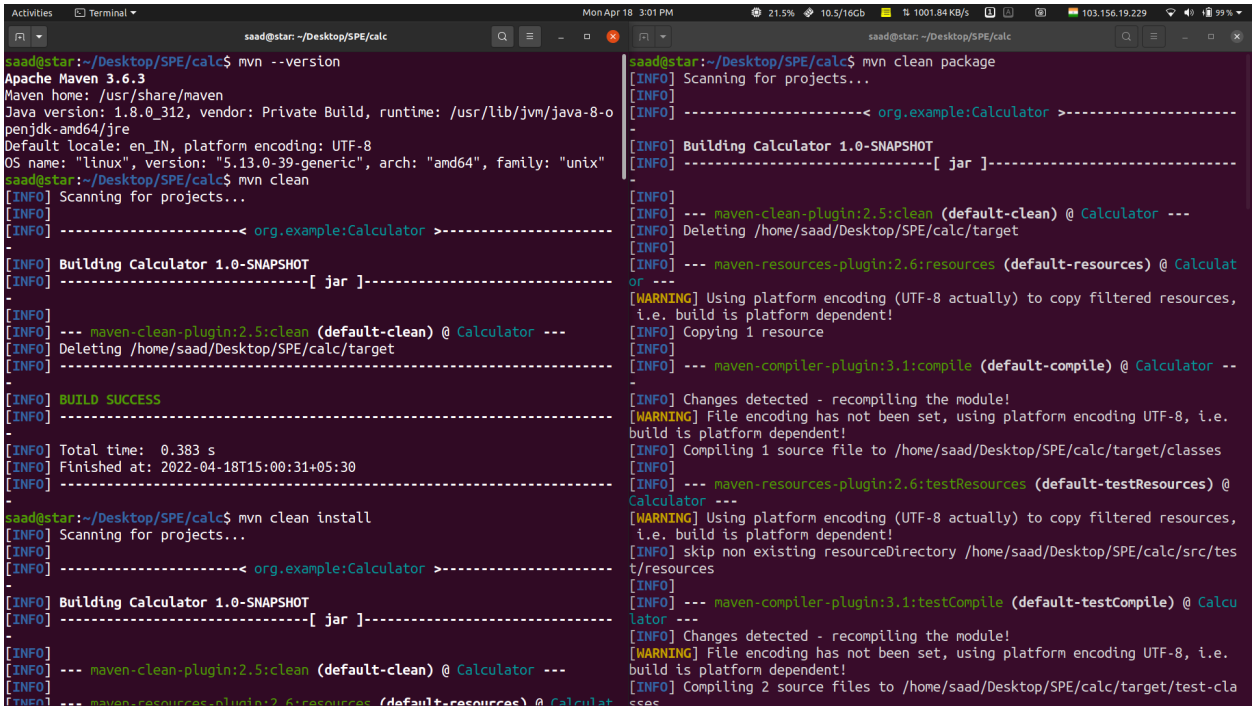
Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven contains a wide set of commands which you can execute. Maven commands are a mix of build life cycles, build phases and build goals, and can thus be a bit confusing. Therefore I will describe the common Maven commands in this tutorial, as well as explain which build life cycles, build phases and build goals they are executing.

Commands

mvn --version	Prints out the version of Maven you are running.
mvn clean	Clears the target directory into which Maven normally builds your project.
mvn package	Builds the project and packages the resulting JAR file into the target directory.
mvn clean install	Clears the target directory and builds the project described by your Maven POM file and installs the resulting artifact (JAR) into your local Maven repository

Following image shows working of all above commands in my system.



IntelliJIDEA:

IntelliJ IDEA is an integrated development environment written in Java for developing computer software. It is developed by JetBrains, and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development.

$ sudo snap install intellij-idea-community - -classic

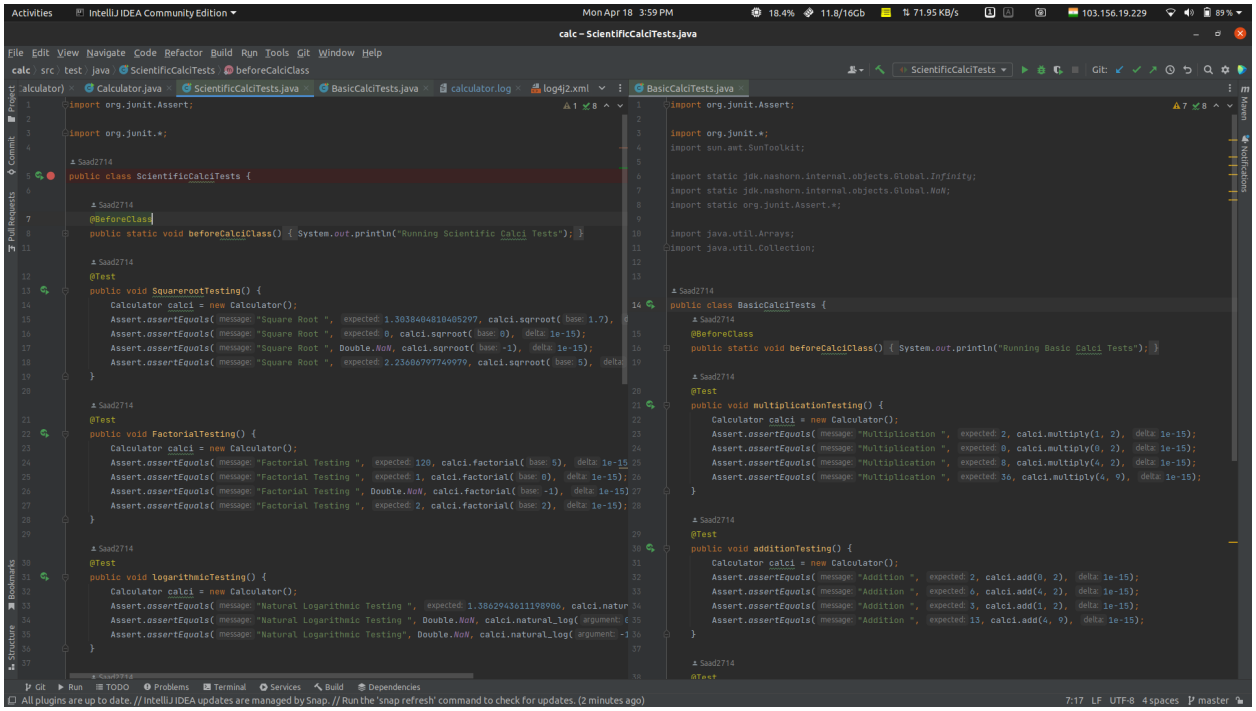
TESTS Basic and Scientific using JUNIT:

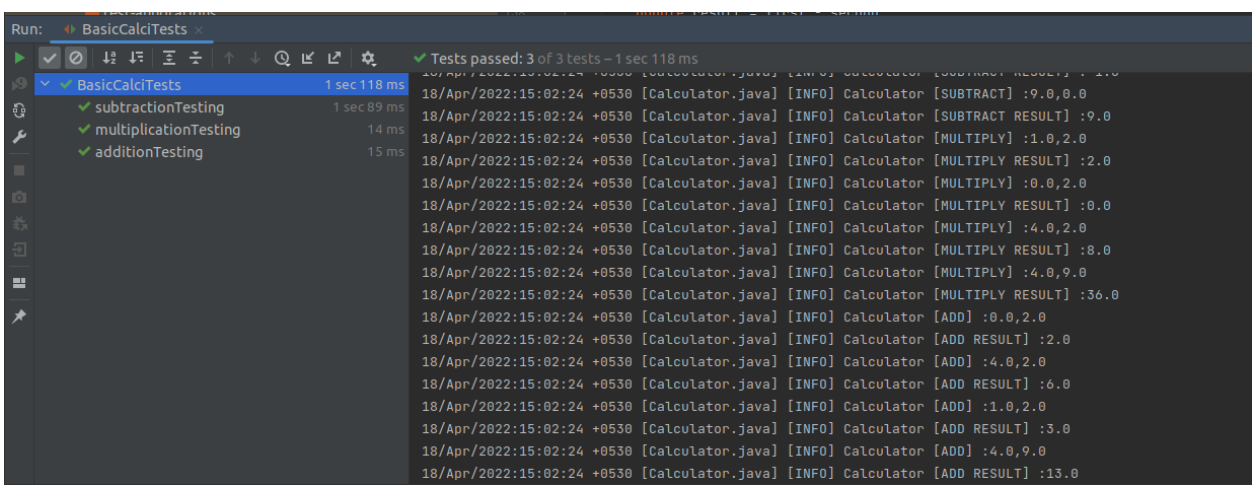
JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit. JUnit is linked as a JAR at compile-time

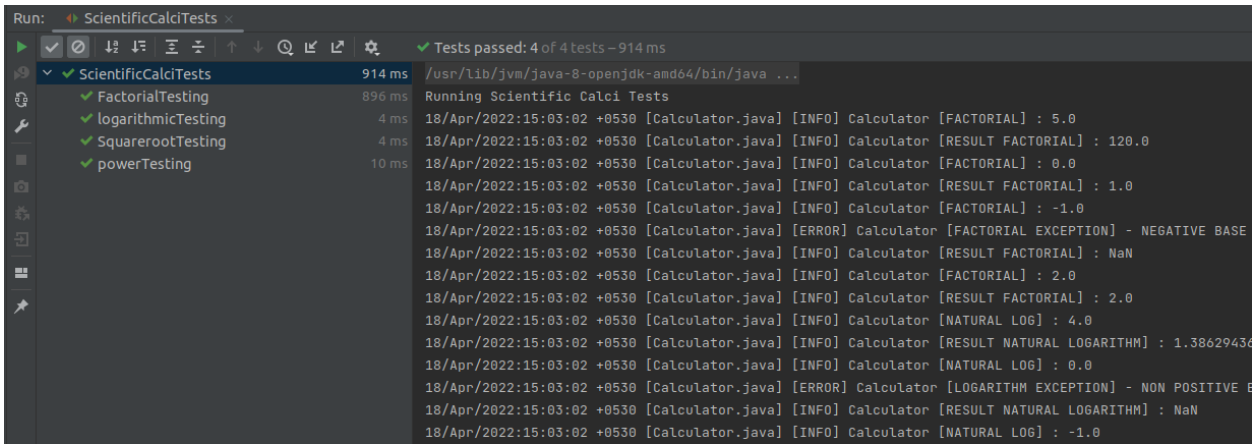
It's one of the best test methods for regression testing. An open-source framework, it is used to write and run repeatable automated tests.JUnit testing is used to test the behavior

of methods inside classes we have written. We test a method for the expected results and sometimes exception-throwing cases—whether the method is able to handle the exceptions in the way we want.

Following images show source code and tests running status for basic and scientific tests in IntelliJIDEA IDE .







Continuous Integration: Jenkins

Continuous Integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where builds and tests are then run.

Continuous Integration enables better transparency and farsightedness in the process of software development and delivery. It not only benefits the developers but all the segments of that company. These benefits make sure that the organization can make better plans and execute them following the market strategy.

Install Jenkins On Ubuntu

```
Step 1: Install Java
$ sudo apt update
$ sudo apt install openjdk-8-jdk
Step 2: Add Jenkins Repository
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key
| sudo apt-key add –
Step 3: Add Jenkins repo to the system
$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable
binary/ > /etc/apt/sources.list.d/jenkins.list'
Step 4: Install Jenkins
$ sudo apt update
$ sudo apt install Jenkins
Step 5: Verify installation
$ systemctl status Jenkins
Step 6: Once Jenkins is up and running, access it from the
link:
http://localhost:8080
```

Jenkins Pipeline

Jenkins pipeline is a single platform that runs the entire pipeline as code. Instead of building several jobs for each phase, you can now code the entire workflow and put it in a Jenkinsfile. Jenkinsfile is a text file that stores the pipeline as code. It is written using the Groovy DSL. It can be written based on two syntaxes:

- Scripted pipeline: Code is written on the Jenkins UI instance and is enclosed within the node block
- Declarative pipeline: Code is written locally in a file and is checked into a SCM and is enclosed within the pipeline block

Start, Stop & Restart Jenkins

$ sudo service jenkins restart

$ sudo service jenkins stop

$ sudo service jenkins start

```
saad@star:~/Desktop/SPE/calc$ jenkins --version
2.332.2
saad@star:~/Desktop/SPE/calc$ sudo service jenkins status
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-04-18 11:45:13 IST; 4h 32min ago
   Main PID: 1231 (java)
      Tasks: 43 (limit: 18982)
     Memory: 1.6G
     CGroup: /system.slice/jenkins.service
             └─1231 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroo

Apr 18 16:11:08 star jenkins[1231]: 2022-04-18 10:41:08.585+0000 [id=704]        INFO        c.n.j.
Apr 18 16:11:08 star jenkins[1231]: 2022-04-18 10:41:08.586+0000 [id=704]        INFO        c.n.j.
Apr 18 16:11:08 star jenkins[1231]: 2022-04-18 10:41:08.586+0000 [id=704]        INFO        c.n.j.
Apr 18 16:11:08 star jenkins[1231]: 2022-04-18 10:41:08.586+0000 [id=704]        INFO        hudson
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.584+0000 [id=705]        INFO        hudson
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.584+0000 [id=705]        INFO        c.n.j.
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.585+0000 [id=705]        INFO        c.n.j.
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.585+0000 [id=705]        INFO        c.n.j.
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.585+0000 [id=705]        INFO        c.n.j.
Apr 18 16:16:08 star jenkins[1231]: 2022-04-18 10:46:08.585+0000 [id=705]        INFO        hudson
lines 1-19/19 (END)
```

```
saad@star: ~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps
saad@star:~/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ sudo su -s /bin/bash jenkins
[sudo] password for saad:
jenkins@star:/home/saad/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:k02JMbclspkhP6RtgQ1pC+J6+JT/nGl88+2kkXdAdcg jenkins@star
The key's randomart image is:
+---[RSA 3072]----+
|      o=B o .....|
|  . . +*.% = .E. |
| . . o..X + .    |
|  .    .. = .    |
| o .     S . .   |
|o +      . . .   |
| + . .    o o .  |
|  . ..ooo  * .   |
|     o=. oo.o    |
+----[SHA256]-----+
jenkins@star:/home/saad/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$ ssh-copy-id saad@localh
ost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:7Ue/++hlDln31ReX2PS41iOwKnydebuBlfuYAv9EkXI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install t
he new keys
saad@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'saad@localhost'"
and check to make sure that only the key(s) you wanted were added.

jenkins@star:/home/saad/Desktop/SPE/mini project/calc/ScientificCalculatorDevOps$
```

```
saad@star: ~/Desktop/SPE/mini project/calc
saad@star:~/Desktop/SPE/mini project$ cd calc/
saad@star:~/Desktop/SPE/mini project/calc$ whoami
saad
saad@star:~/Desktop/SPE/mini project/calc$
```

Following image shows working of complete pipeline of the project (Build #33).

Stage View

	Git Pull stage	Build and Test maven	Docker Build Image	Push Docker Image	Deploy using Ansible
Average stage times: (Average full run time: ~50s)	1s	13s	4s	20s	703ms
#33 Apr 18 12:04 — 1 commit	943ms	7s	1s	25s	1s

Maven Build Stage:

```
[Pipeline] { (Build and Test maven)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ mvn clean install
[¤[1;34mINFO¤[m] Scanning for projects...
[¤[1;34mINFO¤[m]
[¤[1;34mINFO¤[m] ¤[1m-----------------------< ¤[0;36morg.example:Calculator¤[0;1m >-----------------------¤[m
[¤[1;34mINFO¤[m] ¤[1mBuilding Calculator 1.0-SNAPSHOT¤[m
[¤[1;34mINFO¤[m] ¤[1m--------------------------------[ jar ]---------------------------------¤[m
[¤[1;34mINFO¤[m]
[¤[1;34mINFO¤[m] ¤[1m--- ¤[0;32mmaven-clean-plugin:2.5:clean¤[m ¤[1m(default-clean)¤[m @ ¤[36mCalculator¤[0;1m ---¤[m
[¤[1;34mINFO¤[m] Deleting /var/lib/jenkins/workspace/calc_pipeline/target
[¤[1;34mINFO¤[m]
[¤[1;34mINFO¤[m] ¤[1m--- ¤[0;32mmaven-resources-plugin:2.6:resources¤[m ¤[1m(default-resources)¤[m @ ¤[36mCalculator¤[0;1m ---¤[m
[¤[1;33mWARNING¤[m] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[¤[1;34mINFO¤[m] Copying 1 resource
[¤[1;34mINFO¤[m]
[¤[1;34mINFO¤[m] ¤[1m--- ¤[0;32mmaven-compiler-plugin:3.1:compile¤[m ¤[1m(default-compile)¤[m @ ¤[36mCalculator¤[0;1m ---¤[m
[¤[1;34mINFO¤[m] Changes detected - recompiling the module!
[¤[1;33mWARNING¤[m] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[¤[1;34mINFO¤[m] Compiling 1 source file to /var/lib/jenkins/workspace/calc_pipeline/target/classes
```

Docker Build Stage:

```
[Pipeline] stage
[Pipeline] { (Docker Build Image)
[Pipeline] script
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t saad2714/scientific-calc:latest .
Sending build context to Docker daemon  12.61MB

Step 1/4 : FROM openjdk:8
 ---> 18fbe41f975e
Step 2/4 : COPY ./target/Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar ./
 ---> 99d0803ae6a0
Step 3/4 : WORKDIR ./
 ---> Running in d5844b534ff8
Removing intermediate container d5844b534ff8
 ---> b9d50c2b1a6a
Step 4/4 : CMD ["java", "-jar", "Calculator-1.0-SNAPSHOT-jar-with-dependencies.jar"]
 ---> Running in 648c494fcaaf
Removing intermediate container 648c494fcaaf
 ---> b45ef5626147
Successfully built b45ef5626147
Successfully tagged saad2714/scientific-calc:latest
```

Docker Pull Stage:

```
[Pipeline] stage
[Pipeline] { (Push Docker Image)
[Pipeline] script
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u saad2714 -p ******** https://index.docker.io/v1/
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/calc_pipeline@tmp/ddc253cd-776c-4f3b-9a97-
011eb70fef3e/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag saad2714/scientific-calc:latest saad2714/scientific-calc:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push saad2714/scientific-calc:latest
The push refers to repository [docker.io/saad2714/scientific-calc]
```

Ansible Deploy Stage:

```
[Pipeline] stage
[Pipeline] { (Deploy using Ansible)
[Pipeline] ansiblePlaybook
[calc_pipeline] $ /usr/bin/ansible-playbook ansible/deploy.yml -i ansible/inventory
[WARNING]: Could not match supplied host pattern, ignoring: local

PLAY [Pull Docker image for Scientific Calculator] ****************************
skipping: no hosts matched

PLAY RECAP *********************************************************************

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Jenkins Running status:

```
saad@star:~/Desktop/SPE/calc$ sudo systemctl status jenkins
[sudo] password for saad:
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-04-18 11:45:13 IST; 1h 56min ago
   Main PID: 1231 (java)
      Tasks: 42 (limit: 18982)
     Memory: 1.3G
     CGroup: /system.slice/jenkins.service
             └─1231 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Apr 18 13:36:08 star jenkins[1231]: 2022-04-18 08:06:08.585+0000 [id=452]        INFO        c.n.j.p.d.DockerContainerWatchdog$Statistics#writeStatisticsToLog: Watchdog Statistics: Number of overall executions:>
Apr 18 13:36:08 star jenkins[1231]: 2022-04-18 08:06:08.586+0000 [id=452]        INFO        c.n.j.p.d.DockerContainerWatchdog#loadNodeMap: We currently have 0 nodes assigned to this Jenkins instance, which we >
Apr 18 13:36:08 star jenkins[1231]: 2022-04-18 08:06:08.586+0000 [id=452]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Docker Container Watchdog check has been completed
Apr 18 13:36:08 star jenkins[1231]: 2022-04-18 08:06:08.586+0000 [id=452]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Finished DockerContainerWatchdog Asynchronous Periodic Work. 1 ms
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.585+0000 [id=453]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Started DockerContainerWatchdog Asynchronous Periodic Work
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.585+0000 [id=453]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Docker Container Watchdog has been triggered
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.585+0000 [id=453]        INFO        c.n.j.p.d.DockerContainerWatchdog$Statistics#writeStatisticsToLog: Watchdog Statistics: Number of overall executions:>
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.586+0000 [id=453]        INFO        c.n.j.p.d.DockerContainerWatchdog#loadNodeMap: We currently have 0 nodes assigned to this Jenkins instance, which we >
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.586+0000 [id=453]        INFO        c.n.j.p.d.DockerContainerWatchdog#execute: Docker Container Watchdog check has been completed
Apr 18 13:41:08 star jenkins[1231]: 2022-04-18 08:11:08.586+0000 [id=453]        INFO        hudson.model.AsyncPeriodicWork#lambda$doRun$1: Finished DockerContainerWatchdog Asynchronous Periodic Work. 1 ms
...skipping...
```

Dashboard › calc_pipeline ›

General | Build Triggers | Advanced Project Options | **Pipeline**

Pipeline script

Script ?

```
pipeline {
    environment{
        imageName = ""
    }
    agent any

    stages {
        stage('Git Pull stage') {
            steps {
                // Get some code from a GitHub repository
                // git 'https://github.com/Saad2714/ScientificCalculatorDevOps'
                git 'https://github.com/Saad2714/DevopsCalculator'
            }
        }
        stage('Build and Test maven') {
            steps {
                script {
                    sh 'mvn clean install'
                }
            }

        }
        stage('Docker Build Image')
        {
            steps{
                script{
                    imageName = docker.build "saad2714/scientific-calc:latest"
                }
            }
        }
        stage('Push Docker Image')
        {
            steps{
                script{
                    docker.withRegistry("", 'docker-login' ){
                        imageName.push()
                    }
                }
            }
        }
         stage('Deploy using Ansible'){
steps{
ansiblePlaybook disableHostKeyChecking: true, installation: 'calc_ansible', inventory: 'ansible/inventory', playbook: 'ansible/deploy.yml'
}
}


    }
}
```

Save | Apply

Jenkins Pipeline code:

```
pipeline {
    environment{
        imageName = ""
    }
    agent any

    stages {
        stage('Git Pull stage') {
            steps {
                // Get some code from a GitHub repository
                // git 'https://github.com/Saad2714/ScientificCalculatorDevOps'
                git 'https://github.com/Saad2714/DevopsCalculator'
            }
        }
        stage('Build and Test maven') {
            steps {
                script {
                    sh 'mvn clean install'
                }
            }
        }
        stage('Docker Build Image')
        {
            steps{
                script{
                    imageName = docker.build "saad2714/scientific-calc:latest"
                }
            }
        }
        stage('Push Docker Image')
        {
            steps{
                script{
                    docker.withRegistry("", 'docker-login' ){
                        imageName.push()
                    }
                }
            }
        }
        stage('Deploy using Ansible')
        {
            steps{
                    ansiblePlaybook disableHostKeyChecking: true, installation: 'calc_ansible', inventory: 'ansible/inventory',
                    playbook: 'ansible/deploy.yml'
                }
        }


    }
}
```

Containerization: Docker:

$ sudo apt-get remove docker docker-engine docker.io

$ sudo apt-get update

$ sudo apt install docker.io

$ sudo snap install docker

$ docker --version

```
saad@star:~/Desktop/SPE/calc$ docker --version
Docker version 20.10.14, build a224086
saad@star:~/Desktop/SPE/calc$ sudo docker images
[sudo] password for saad:
REPOSITORY                                          TAG       IMAGE ID       CREATED             SIZE
saad2714/scientific-calc                            latest    054b41f37b3e   About an hour ago   530MB
docker-elk_setup                                    latest    50bc7357da57   3 hours ago         1.19GB
docker.elastic.co/kibana/kibana                     8.1.2     a6d3a3c39d21   2 weeks ago         843MB
docker-elk_kibana                                   latest    a6d3a3c39d21   2 weeks ago         843MB
docker.elastic.co/elasticsearch/elasticsearch       8.1.2     0652ab468732   2 weeks ago         1.19GB
docker-elk_elasticsearch                            latest    0652ab468732   2 weeks ago         1.19GB
openjdk                                             8         18fbe41f975e   2 weeks ago         526MB
docker.elastic.co/logstash/logstash                 8.1.2     6b7ac898ceb0   2 weeks ago         753MB
docker-elk_logstash                                 latest    6b7ac898ceb0   2 weeks ago         753MB
saad@star:~/Desktop/SPE/calc$
```

DockerHub Repo Link:

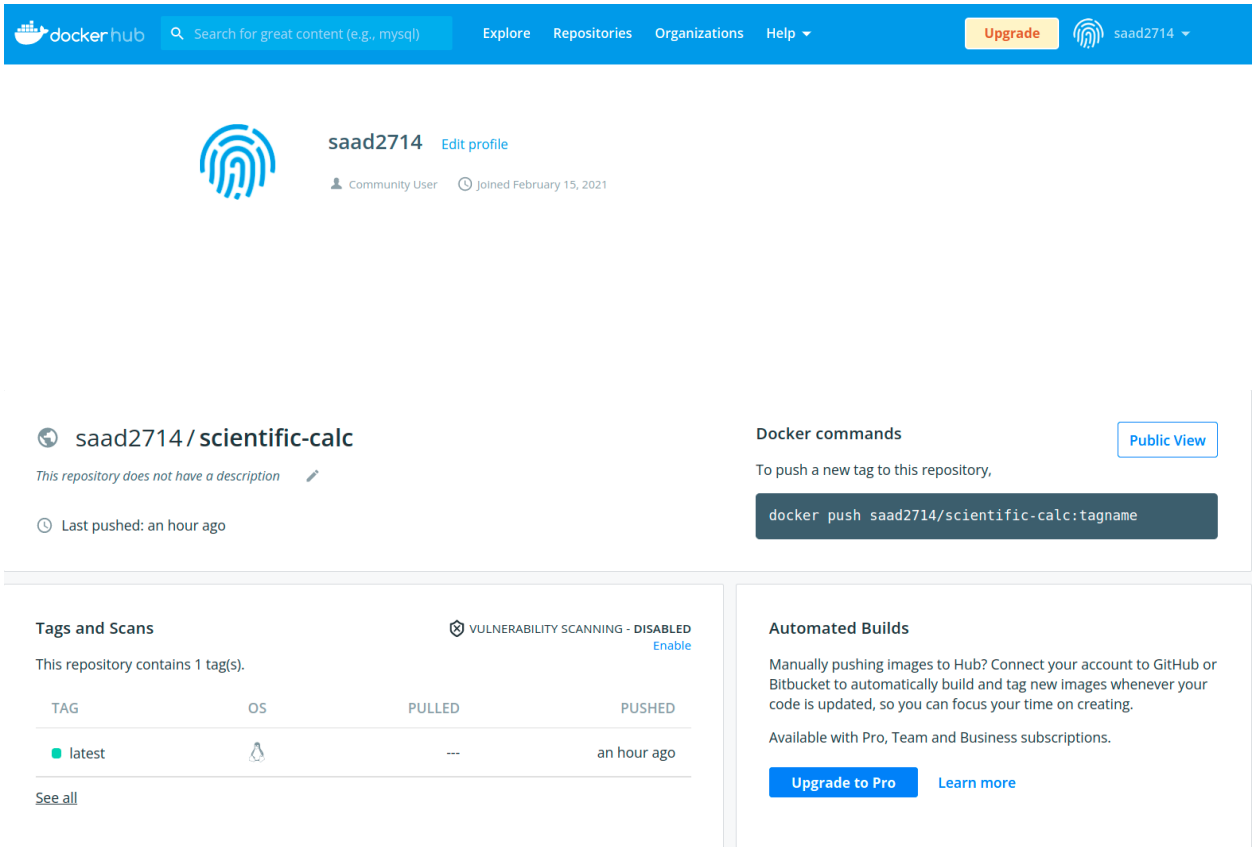
https://hub.docker.com/repository/docker/saad2714/scientific-calc

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allows you to run many containers simultaneously on a given host. Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host. You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

Dockerhub profile and image repo:



Docker pipeline code in Jenkins

```
stage('Docker Build Image')
{
    steps{
        script{
            imageName = docker.build "saad2714/scientific-calc:latest"
        }
    }
}
stage('Push Docker Image')
{
    steps{
        script{
            docker.withRegistry("", 'docker-login' ){
                imageName.push()
            }
        }
    }
}
```

Credentials for docker-login used in pipeline of Jenkins.

Credentials

T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	docker-login	saad2714/******
		Jenkins	(global)	ansibleid1	saad

Remote testing on local system of jenkins built application:

```
saad@star:~/Desktop/SPE/calc$ sudo docker images
REPOSITORY                                      TAG       IMAGE ID       CREATED             SIZE
saad2714/scientific-calc                        latest    054b41f37b3e   51 seconds ago      530MB
docker-elk_setup                                latest    50bc7357da57   About an hour ago   1.19GB
docker.elastic.co/kibana/kibana                 8.1.2     a6d3a3c39d21   2 weeks ago         843MB
docker-elk_kibana                               latest    a6d3a3c39d21   2 weeks ago         843MB
docker-elk_elasticsearch                        latest    0652ab468732   2 weeks ago         1.19GB
docker.elastic.co/elasticsearch/elasticsearch   8.1.2     0652ab468732   2 weeks ago         1.19GB
openjdk                                         8         18fbe41f975e   2 weeks ago         526MB
docker.elastic.co/logstash/logstash             8.1.2     6b7ac898ceb0   2 weeks ago         753MB
docker-elk_logstash                             latest    6b7ac898ceb0   2 weeks ago         753MB
saad@star:~/Desktop/SPE/calc$ sudo docker run -it 054
Hello, Welcome to my calculator, Choose from below to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press 5 to calculate Square Root
Press 6 to calculate Factorial
Press 7 to calculate Natural Logarithm
Press 8 to calculate Power
Press 9 to exit
Enter your choice: 5
Enter the number : 49
18/Apr/2022:09:13:50 Z [Calculator.java] [INFO] Calculator [SQUARE ROOT] : 49.0
18/Apr/2022:09:13:50 Z [Calculator.java] [INFO] Calculator [RESULT SQUARE ROOT] : 7.0
Division result is : 7.0


Hello, Welcome to my calculator, Choose from below to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press 5 to calculate Square Root
Press 6 to calculate Factorial
Press 7 to calculate Natural Logarithm
Press 8 to calculate Power
Press 9 to exit
Enter your choice: 9
Exiting....
saad@star:~/Desktop/SPE/calc$
```

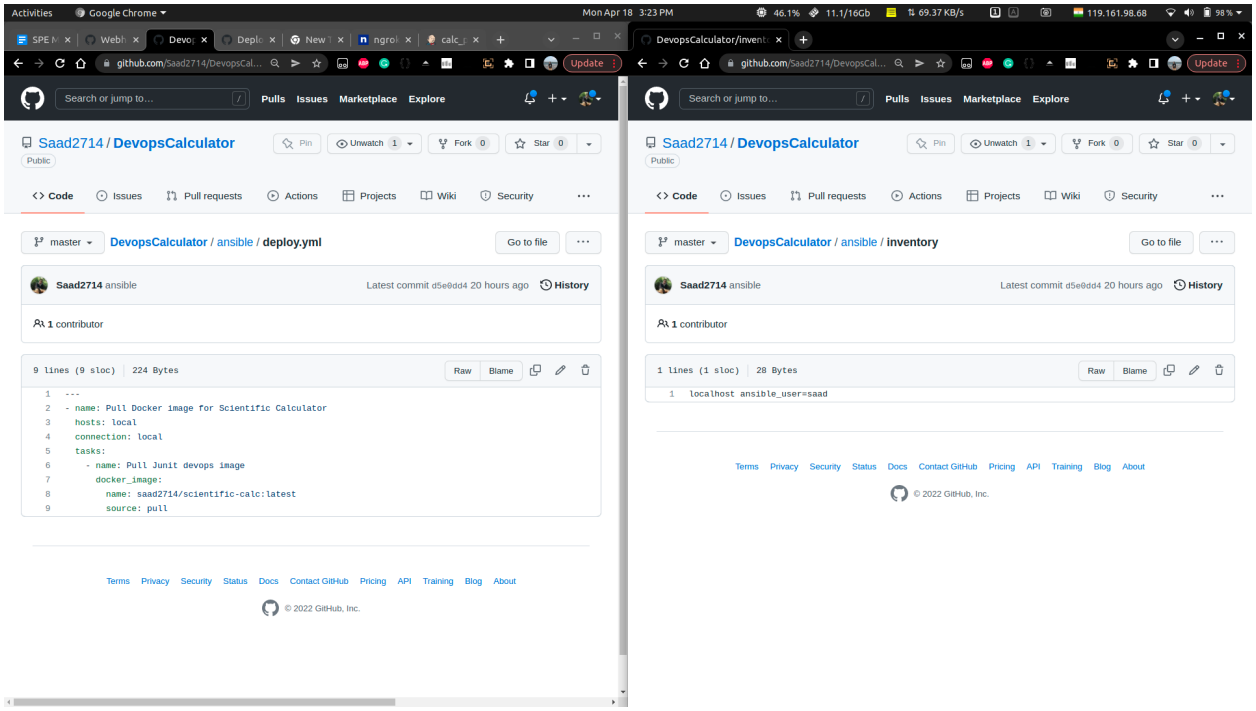
Ansible Deployment:

```
saad@star:~/Desktop/SPE/calc$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/saad/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Mar 15 2022, 12:22:08) [GCC 9.4.0]
saad@star:~/Desktop/SPE/calc$
```

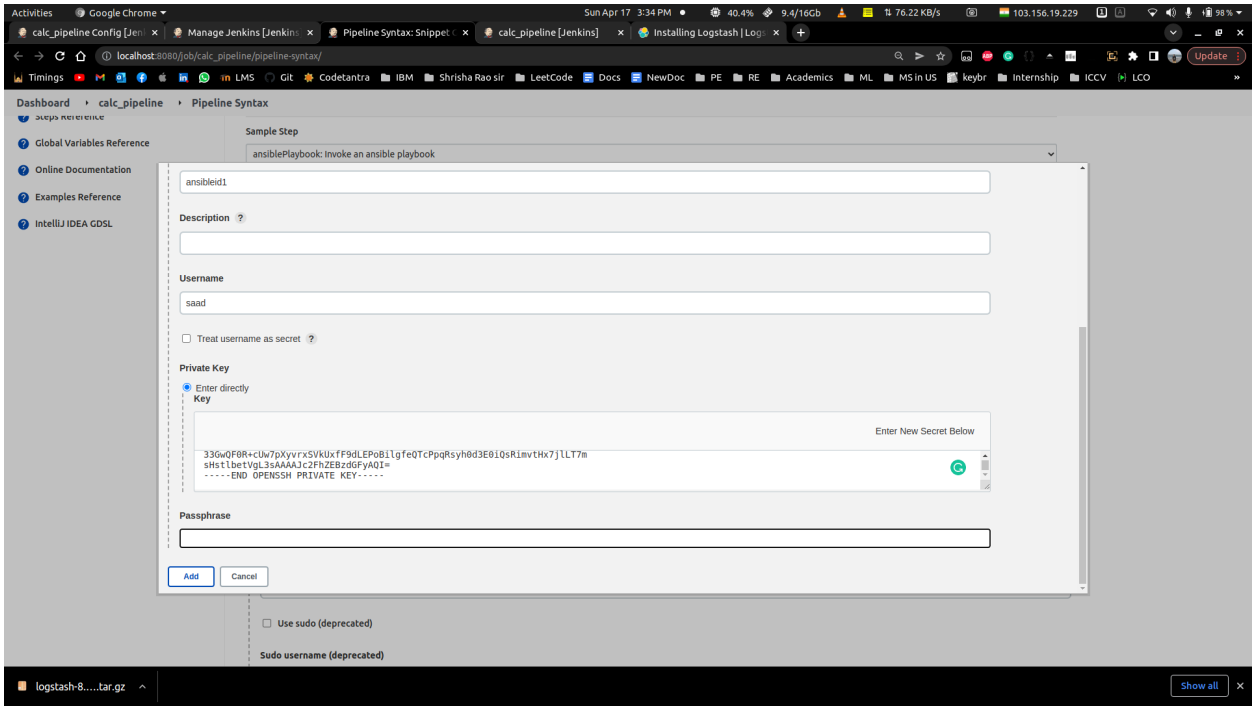
Ansible is the simplest way to deploy your applications. It gives you the power to deploy multi-tier applications reliably and consistently, all from one common framework. You can configure needed services as well as push application artifacts from one common system.

Ansible does not just automate but also simplifies the repetitive, complex, and strenuous tasks that bring substantial time savings and increases overall productivity. As we already know, Ansible helps us to automate server and cloud provisioning, configuration management, and application deployment.

Ansible deploy.yml and inventory source code in git repo:



Ansible private key in pipeline syntax :



Ansible pipeline code in Jenkins:

```
stage('Deploy using Ansible')
{
    steps{
            ansiblePlaybook disableHostKeyChecking: true, installation: 'calc_ansible', inventory: 'ansible/inventory',
            playbook: 'ansible/deploy.yml'
        }
}
```

Local Deployment:

```
saad@star:~/Desktop/SPE/calc$ sudo docker images
REPOSITORY                                          TAG       IMAGE ID       CREATED             SIZE
saad2714/scientific-calc                            latest    054b41f37b3e   51 seconds ago      530MB
docker-elk_setup                                    latest    50bc7357da57   About an hour ago   1.19GB
docker.elastic.co/kibana/kibana                     8.1.2     a6d3a3c39d21   2 weeks ago         843MB
docker-elk_kibana                                   latest    a6d3a3c39d21   2 weeks ago         843MB
docker-elk_elasticsearch                            latest    0652ab468732   2 weeks ago         1.19GB
docker.elastic.co/elasticsearch/elasticsearch       8.1.2     0652ab468732   2 weeks ago         1.19GB
openjdk                                             8         18fbe41f975e   2 weeks ago         526MB
docker.elastic.co/logstash/logstash                 8.1.2     6b7ac898ceb0   2 weeks ago         753MB
docker-elk_logstash                                 latest    6b7ac898ceb0   2 weeks ago         753MB
saad@star:~/Desktop/SPE/calc$ sudo docker run -it 054
Hello, Welcome to my calculator, Choose from below to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press 5 to calculate Square Root
Press 6 to calculate Factorial
Press 7 to calculate Natural Logarithm
Press 8 to calculate Power
Press 9 to exit
Enter your choice: 5
Enter the number : 49
18/Apr/2022:09:13:50 Z [Calculator.java] [INFO] Calculator [SQUARE ROOT] : 49.0
18/Apr/2022:09:13:50 Z [Calculator.java] [INFO] Calculator [RESULT SQUARE ROOT] : 7.0
Division result is : 7.0


Hello, Welcome to my calculator, Choose from below to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press 5 to calculate Square Root
Press 6 to calculate Factorial
Press 7 to calculate Natural Logarithm
Press 8 to calculate Power
Press 9 to exit
Enter your choice: 9
Exiting....
saad@star:~/Desktop/SPE/calc$
```

ELK Monitoring:

ELK is an acronym for several open source tools: Elasticsearch, Logstash, and Kibana. Elasticsearch is the engine of the Elastic Stack, which provides analytics and search functionalities.

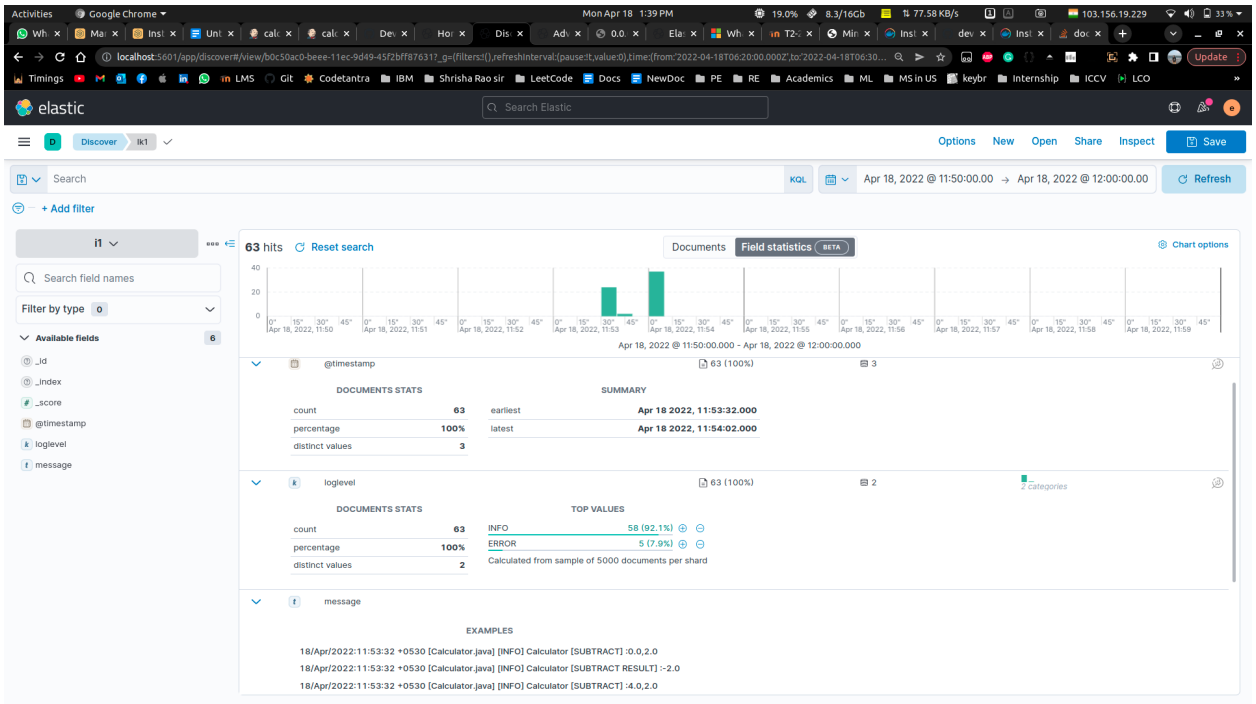
The ELK Stack is a comprehensive tool that sysadmins may find useful for real-time monitoring and analytics. It can also be integrated into other systems.

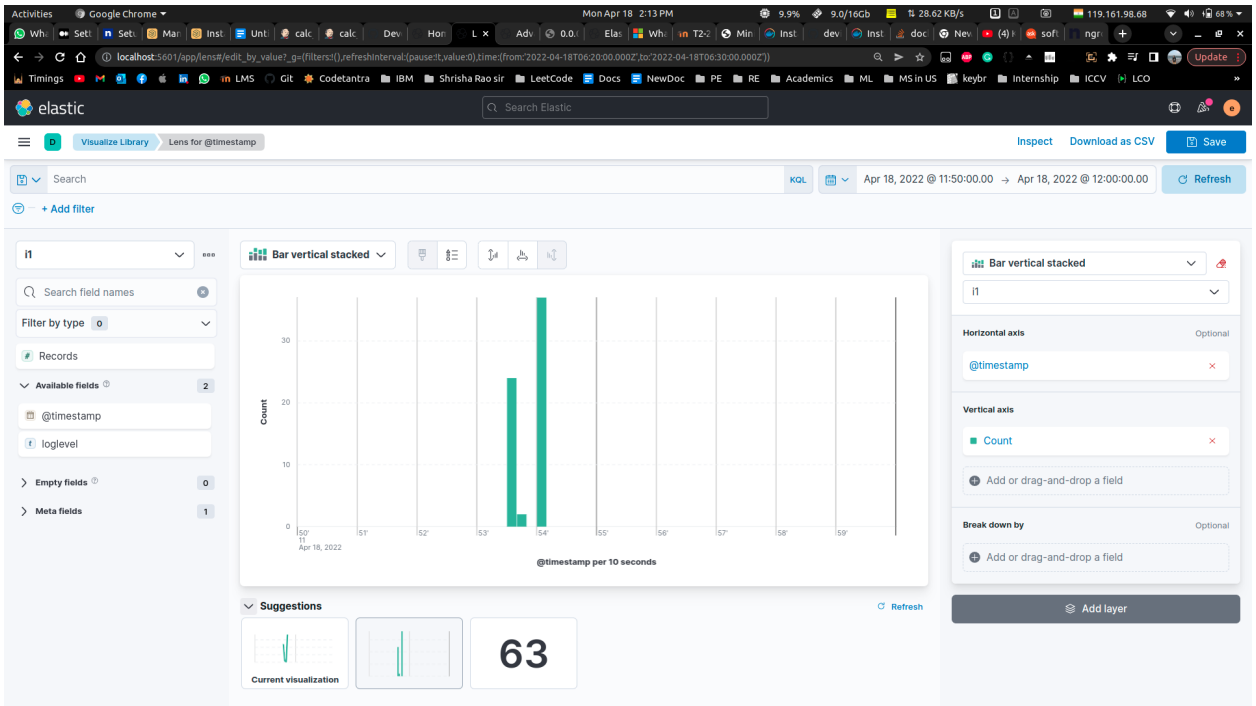
The ELK Stack is a comprehensive tool that sysadmins may find useful for real-time monitoring and analytics. It can also be integrated into other systems.

The ELK Stack helps by providing users with a powerful platform that collects and processes data from multiple data sources, stores that data in one centralized data store that can scale as data grows, and that provides a set of tools to analyze the data.

In order to use ELK to monitor your platform's performance, a couple of tools and integrations are needed. Probes are required to run on each host to collect various system performance metrics. Then, the data needs to be shipped to Logstash, stored and aggregated in Elasticsearch, and then turned into Kibana graphs. Ultimately, software service operations teams use these graphs to present their results.

Elastic monitoring running status screenshot:





GitHub Webhook trigger for GITScm polling using ngrok:

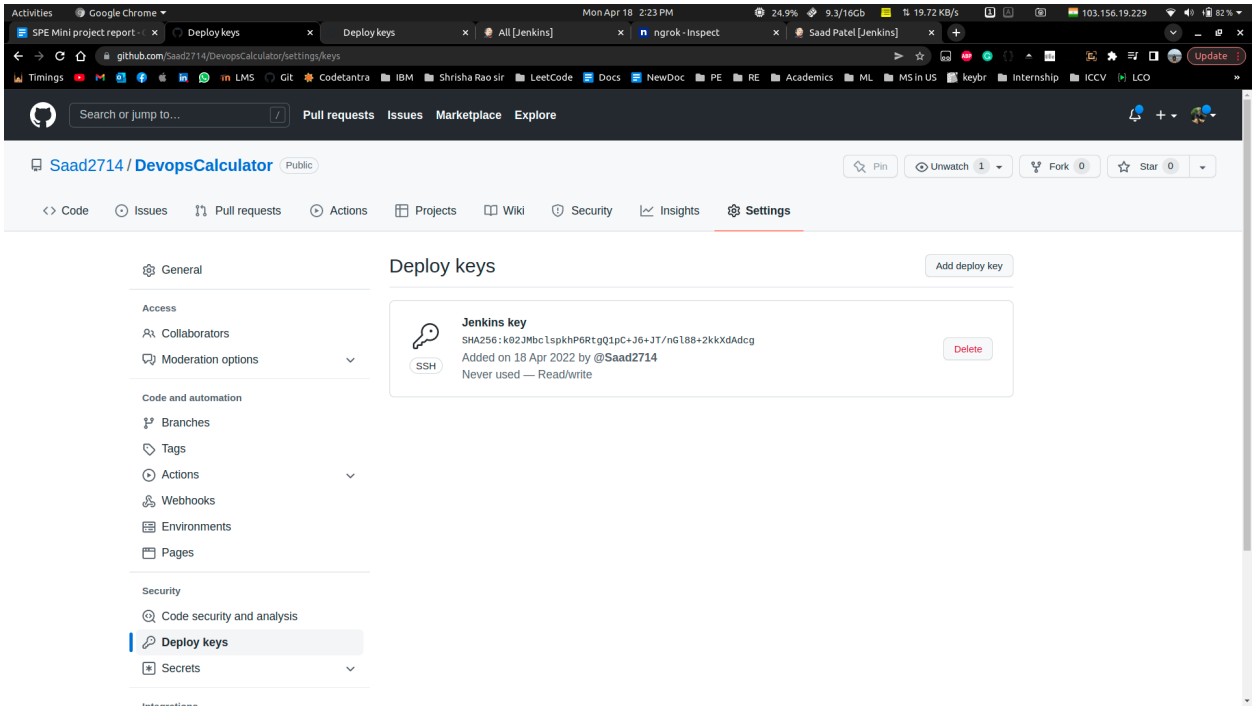
Jenkins is the leading open source automation server. It provides hundreds of plugins to support building, deploying and automating any project. In this article, we're going to look at how to configure GitHub triggers on our jobs and communicate with GitHub using a webhook indicating when to poll the job to build changes made on the project.

Ngrok is a reverse proxy that accepts traffic on a public address, relays that traffic through to the ngrok process running on your machine and then on to the local address you specified.

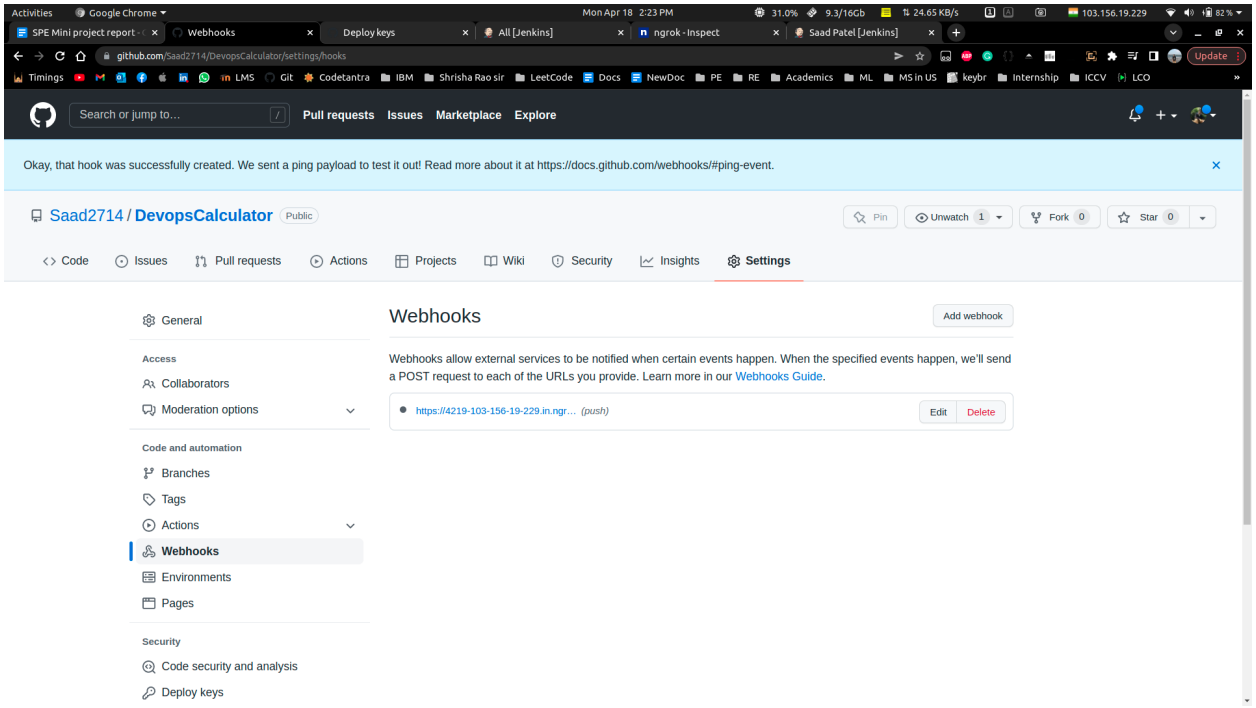
So signed up at https://ngrok.com/ and after installation launched at port 8080.

Received proxy hostname after running the command and it looks like this:
Forwarding http://xxxxx.ngrok.io -> http://localhost:8080, as shown in last image

Added Jenkins ssh key:



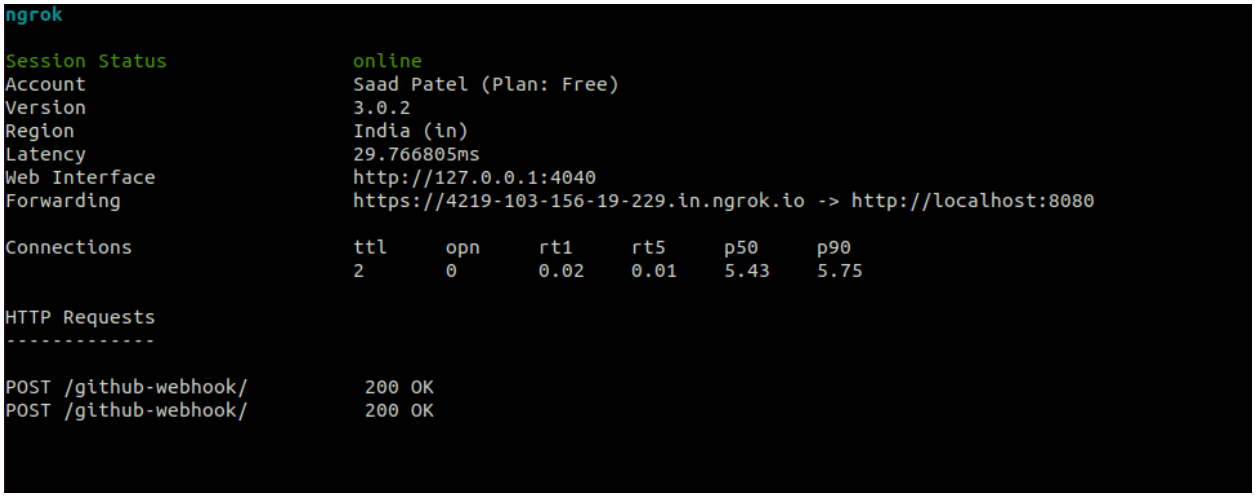
Added Webhooks in github repo:



Ngrok session status and working of complete project screenshot:

$./ngrok http 8080

POST /github-webhook/ **<u>Status 200 OK</u>**



```
ngrok

Session Status                online
Account                       Saad Patel (Plan: Free)
Version                       3.0.2
Region                        India (in)
Latency                       29.766805ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://4219-103-156-19-229.in.ngrok.io -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              2       0       0.02    0.01    5.43    5.75

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

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