Scientific Calculator Project Report

NEERAJ JETHA MT2020079

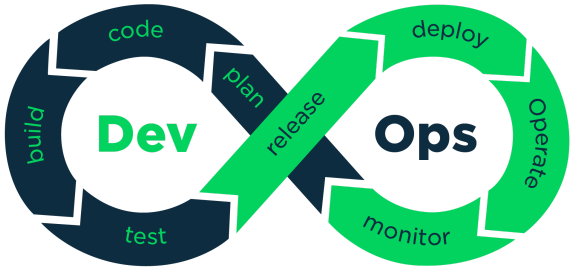
Project link :- https://github.com/njjetha/Calculator

Docker hub :- https://hub.docker.com/repository/registry-1.docker.io/njjetha/calculator-devops/tags

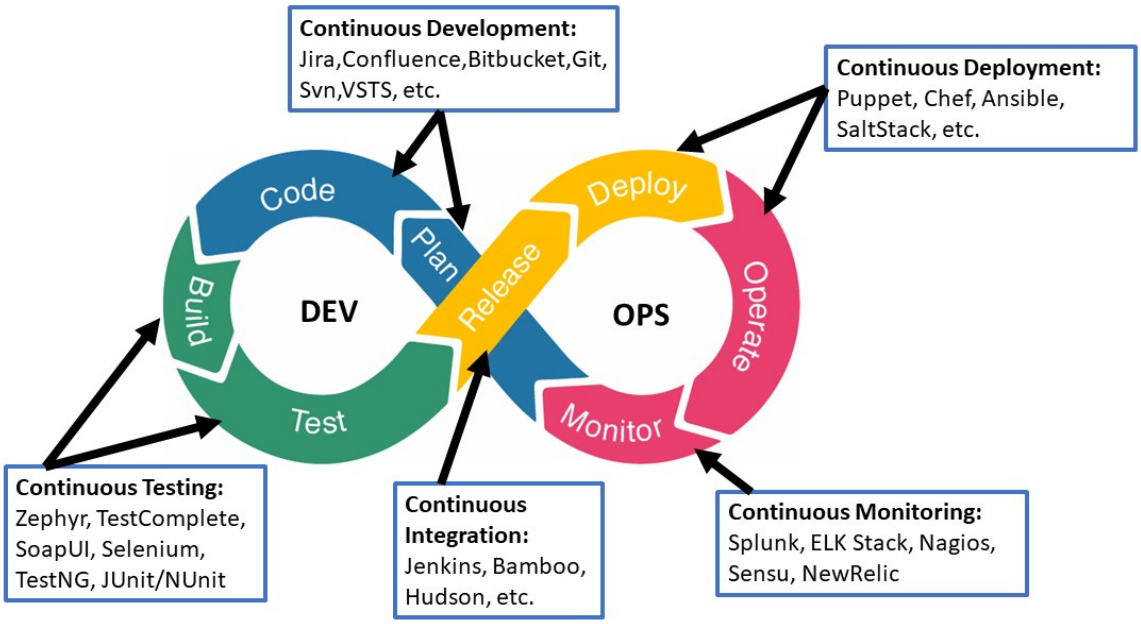
What is DevOps ?

DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

There are three primary practice areas that are usually discussed in context of DevOps.



- **Infrastructure Automation** – create your systems, OS configs, and app deployments as code.
- **Continuous Delivery** – build, test, deploy your apps in a fast and automated manner.
- **Site Reliability Engineering** – operate your systems; monitoring and orchestration, sure, but also designing for operability in the first place



Why there is need of DevOps?

The DevOps approach was designed to ensure that high-quality updated software gets into the hands of users more quickly. Though the DevOps is only a few years old, the DevOps movement is so widespread.

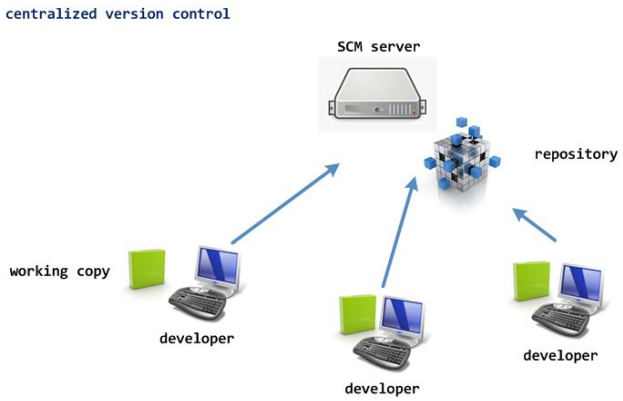
1. **Shorter Development Cycles, Faster Innovation :-** When development and operations teams are in separate silos, it's usually difficult to tell if an application is ready for operations.
   When development teams simply turn over an application, the operations' cycle times are extended needlessly.With a combined development and operations team, applications are ready for use much more quickly.
2. **Reduced Deployment Failures, Rollbacks, and Time to Recover** :- Part of the reason teams experience deployment failures is due to programming defects. The shorter development cycles with DevOps promote more frequent code releases. This, in turn, makes it easier to spot code defects. Therefore, teams can reduce the number of deployment failures using agile programming principles that call for collaboration and modular programming
3. **Improved Communication and Collaboration :-** DevOps improves the software development culture. Combined teams are happier and more productive. The culture becomes focused on performance rather than

individual goals. When the teams trust each other, they can experiment and innovate more effectively.

4. **Increased Efficiencies :-** Increased efficiency helps to speed the development process and make it less prone to error. There are ways to automate DevOps tasks. Continuous integration servers automate the process of testing code, reducing the amount of manual work required.
5. **Reduced Costs** :- All of the DevOps benefits translate to reduced overall costs and IT headcount requirements.

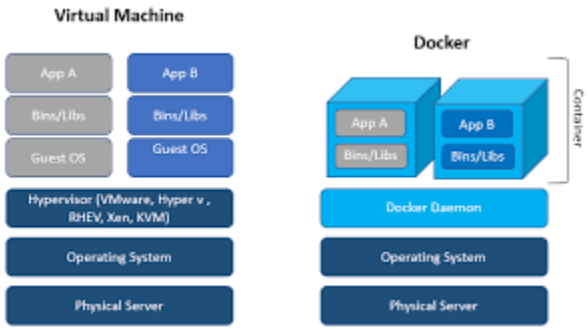
DevOps Toolchain

1. **Source Control Management tool :-** We all know that change is inevitable and we believe that change is a betterment. Thus, changes happen day in and out to everything and anything. So, it is extremely important for a program to maintain the history of these changes to the program artifacts, which is called control'. Hence, Version control or Source control is to manage the changes to project, be it code, documents, environment configuration or else.



1. **Testing :-** Testing in DevOps spans the whole software development and delivery lifecycle. Testers are no longer just focusing on functional testing and feature verification.As testers, we should also be involved in operations testing, performance testing, basic security testing, as well as being able to monitor and analyze production data and logs.

2. **Build** :- The build phase of a DevOps pipeline is crucial because it allows developers to detect errors in the code before they make their way down the pipeline and cause a major disaster. After the newly written code has been merged with the shared repository, developers run a series of automated test.
3. **Continuous Integration** :- It is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and
   tests are run. Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g. a CI or build service) and a cultural component (e.g. learning to integrate frequently). The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.
4. **Containerize** :- Containers, by contrast, isolate applications' execution environments from one another, but share the underlying OS kernel. They're typically measured in megabytes, use far fewer resources than VMs, and start up almost immediately.Containers provide a highly efficient and highly granular mechanism for combining software components into the kinds of application and service stacks needed in a modern enterprise, and for keeping those software components updated and maintained

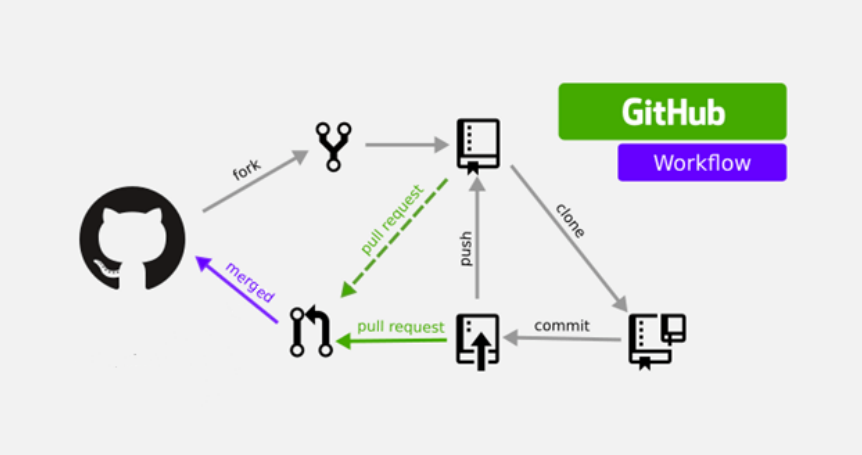


5. **Deployment** :- A Deployment pipeline is the process of taking code from version control and making it readily available to users of your application in an automated fashion. When a team of developers are working on projects or features they need a reliable and efficient way to build, test and deploy their work.

6. **Continuous Monitoring** :- Continuous Monitoring comes in at the end of the DevOps pipeline. Once the software is released into production, Continuous Monitoring will notify dev and QA teams in the event of specific issues arising in the prod environment. It provides feedback on what is going wrong, which allows the relevant people to work on necessary fixes as soon as possible.

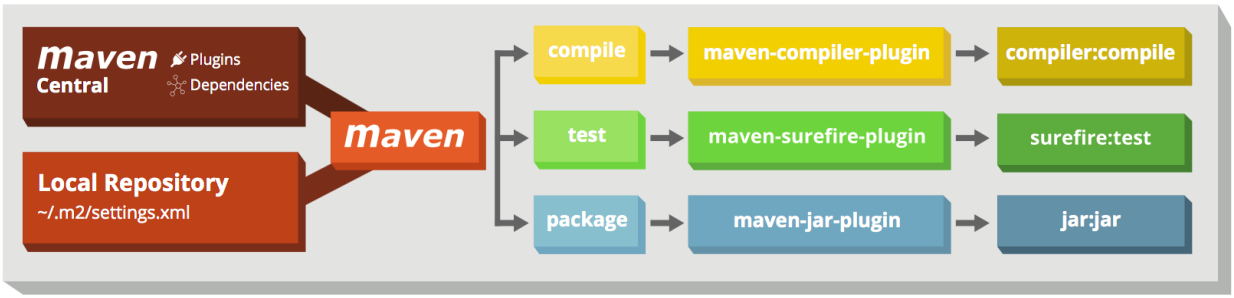
Tools used in Project

***Github* :-** GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

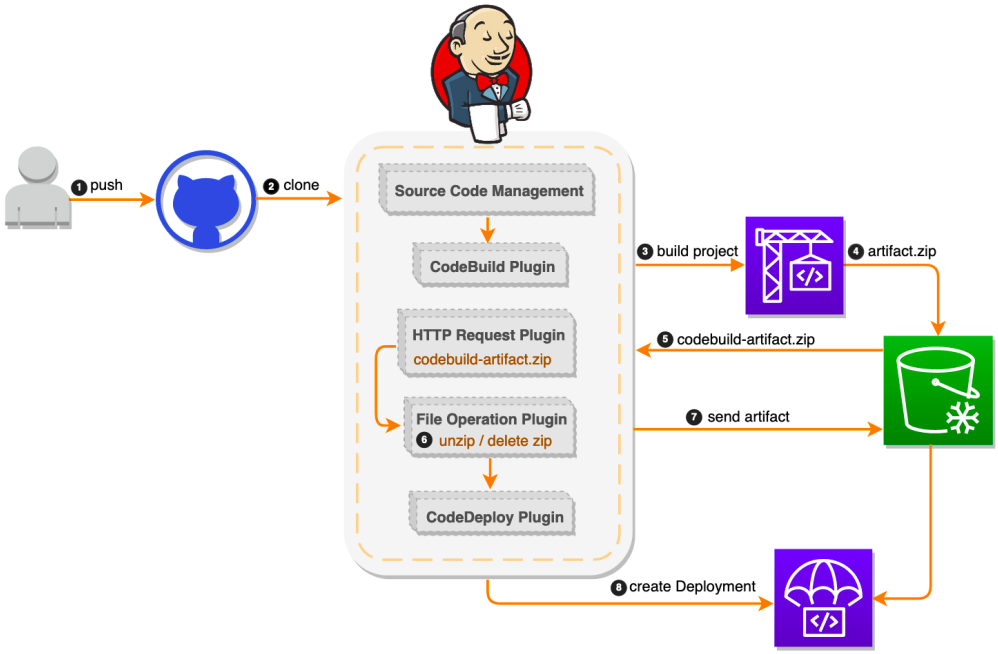


JUnit :- Unit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit, that originated with JUnit.Testing is the process of checking the functionality of an application to ensure it runs as per requirements. Unit testing comes into picture at the developers' level; it is the testing of single entity (class or method). Unit testing plays a critical role in helping a software company deliver quality products to its customers

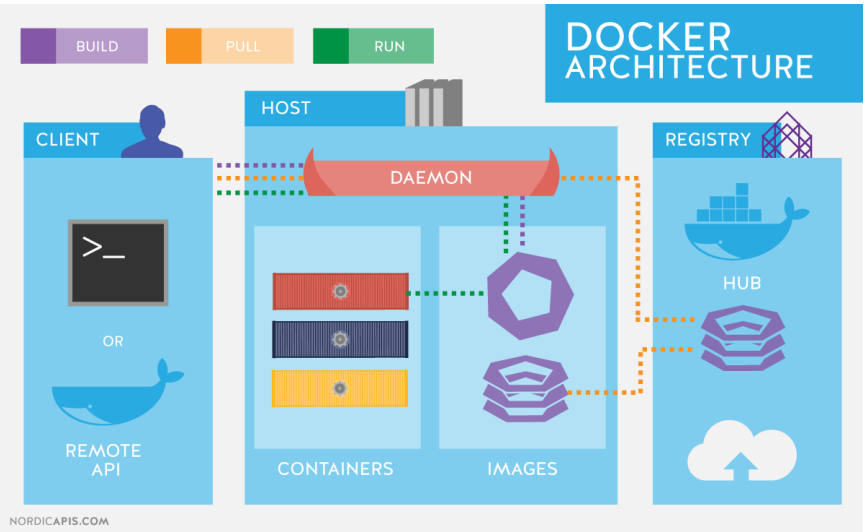
Maven:- Maven is a project management tool.Most popular use of Maven is for build management and dependencies.Maven automatically download the dependencies that are required in your project. Maven will make those jar files available during compile/run.



Jenkins :- Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment technologies.



Docker :- Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.



ELK :- "ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.

Steps by Steps Explanation of Project

Step :- 1 Maven Installation

Let’s look into the installation of Maven on our machine. To install Maven we want to first
install its dependency JAVA, here we will be installing java version 8, to do that

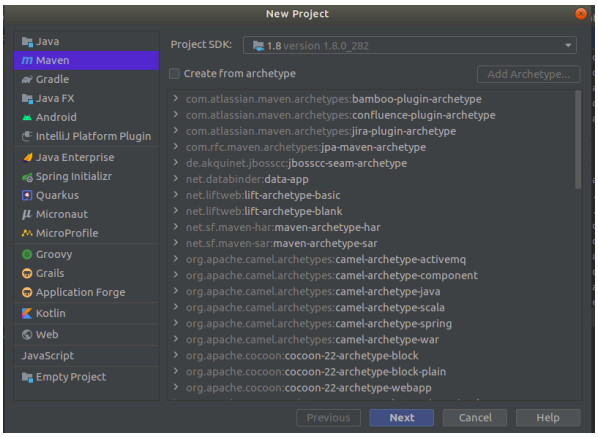
follow the
following commands.

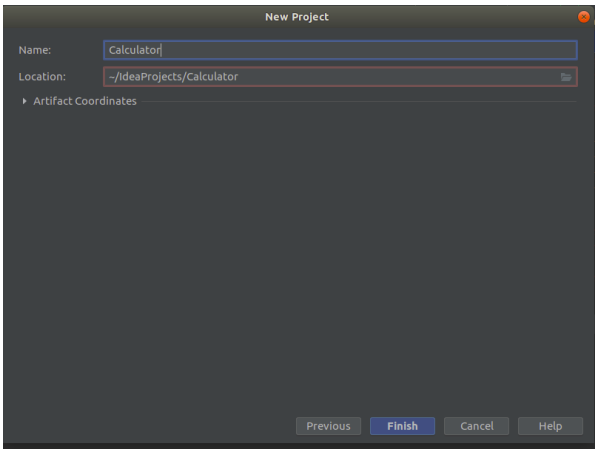
```
$ sudo apt update
$ sudo apt install maven
```

check Maven version

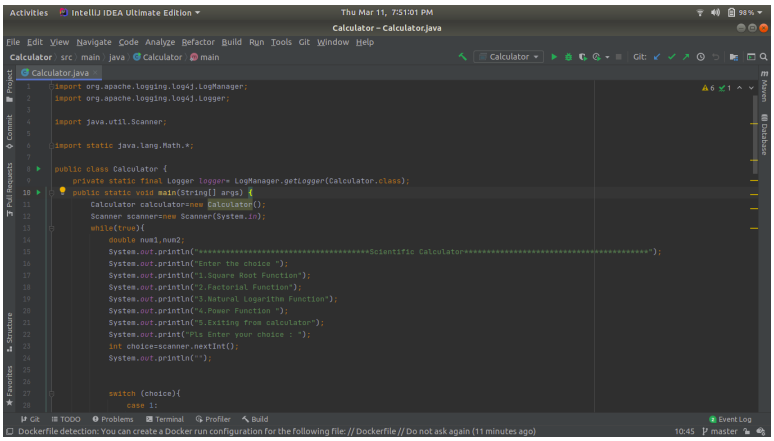
```
neeraj@njjetha:~$ mvn --version
Apache Maven 3.6.0
Maven home: /usr/share/maven
Java version: 1.8.0_282, vendor: Private Build, runtime: /usr/lib/jvm/java-8-ope
njdk-amd64/jre
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "5.4.0-65-generic", arch: "amd64", family: "unix"
```

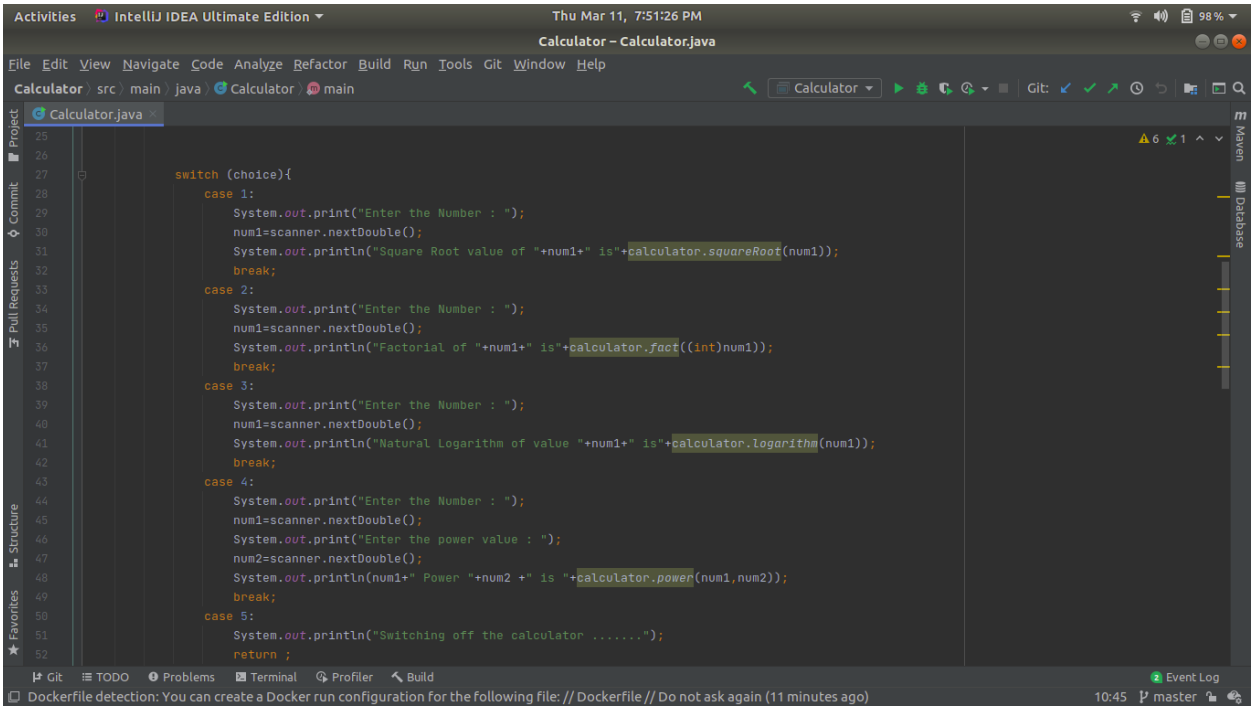
Start Maven Project and name of the project is Calculator





Typing the logic for the calculator





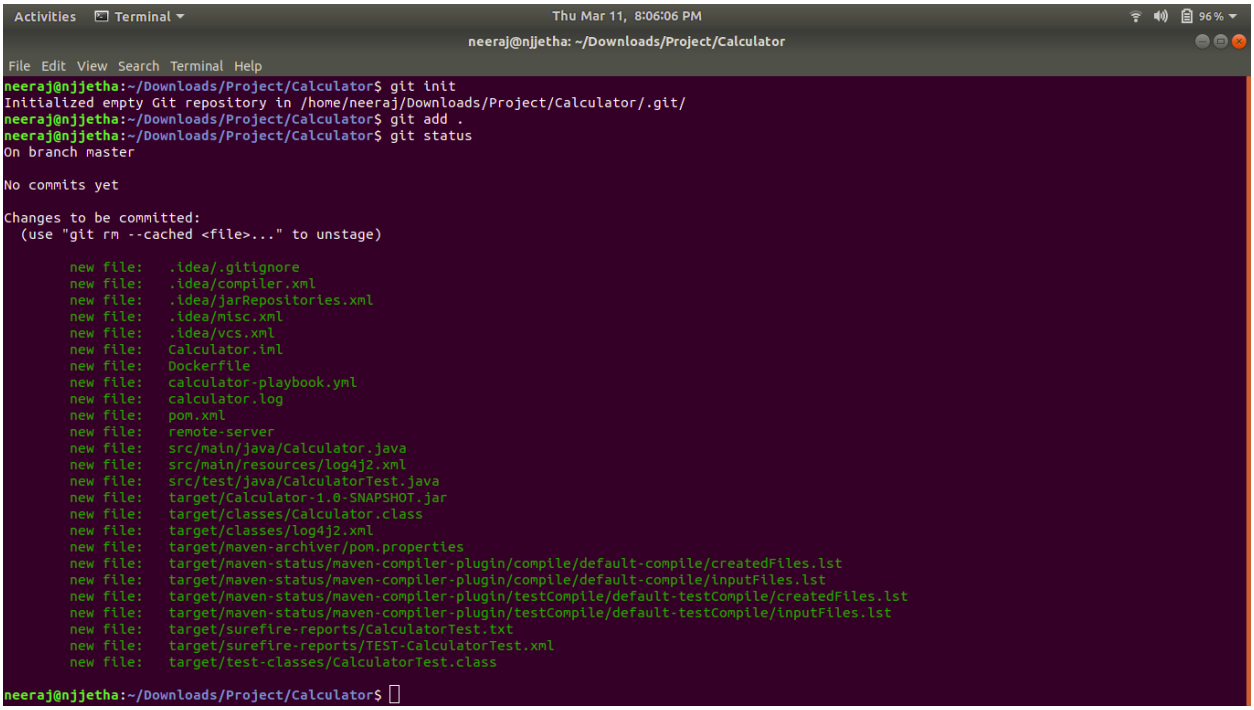
Step 2:- Initialize the git repository

```
neeraj@njjetha:~/Downloads/Project/Calculator$ git init
Reinitialized existing Git repository in /home/neeraj/Downloads/Project/Calculator/.git/
neeraj@njjetha:~/Downloads/Project/Calculator$
```

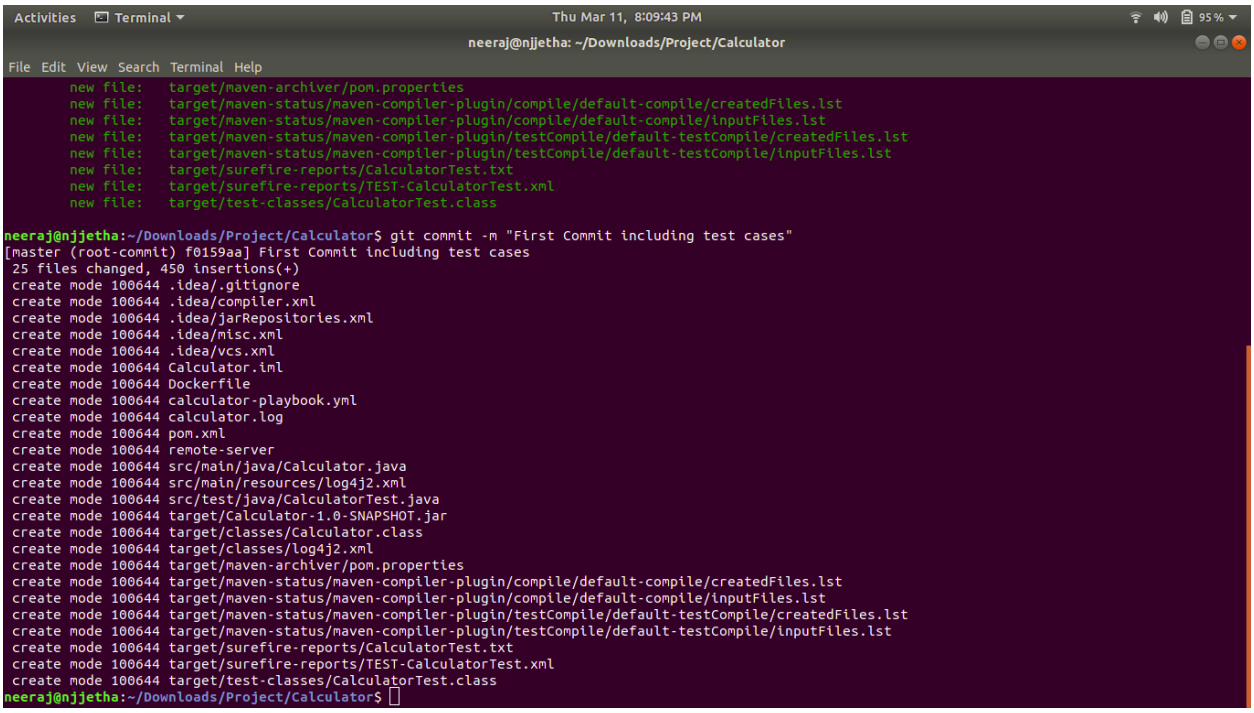
git add → The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the

next commit

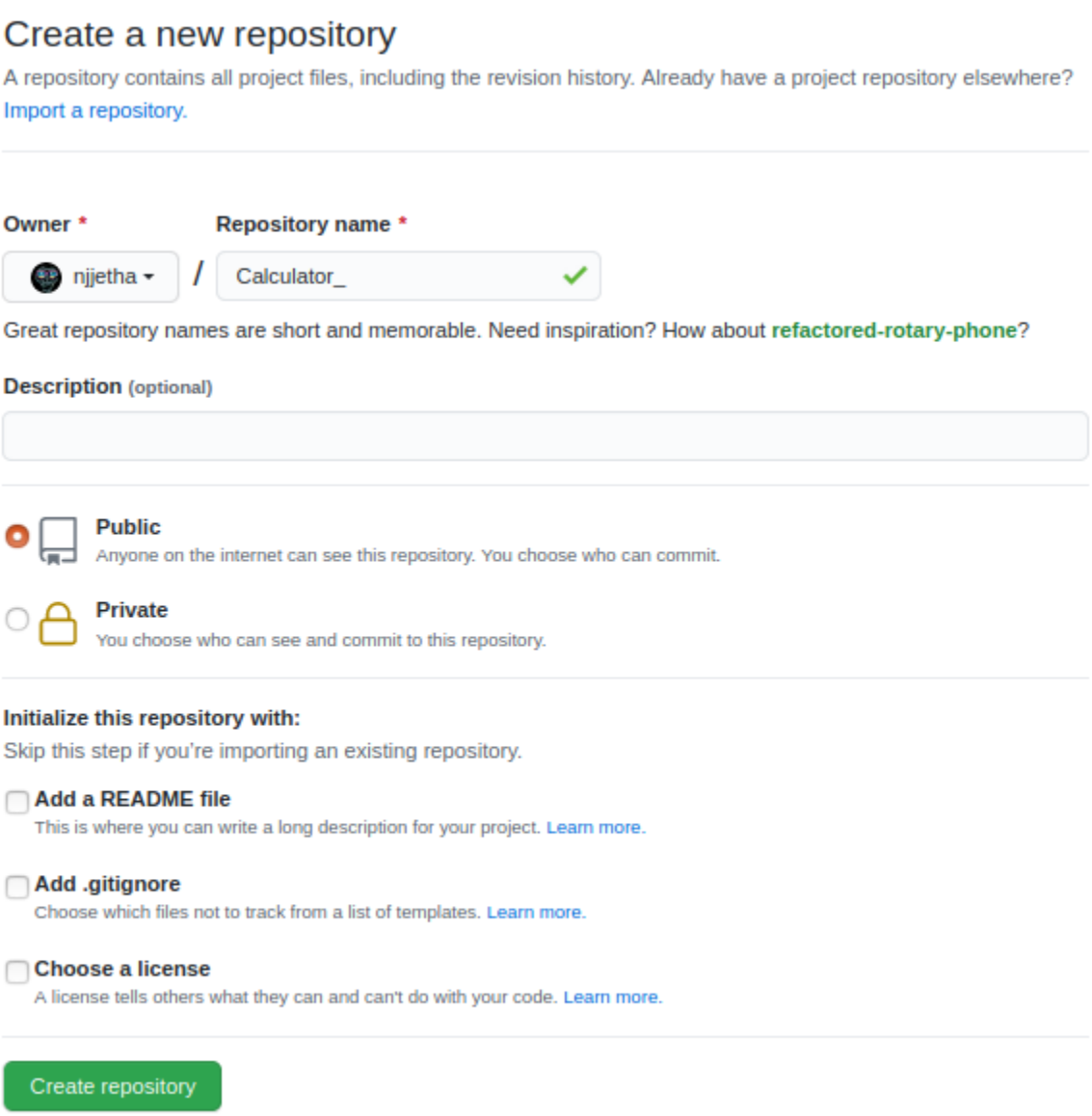
git status →The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history.



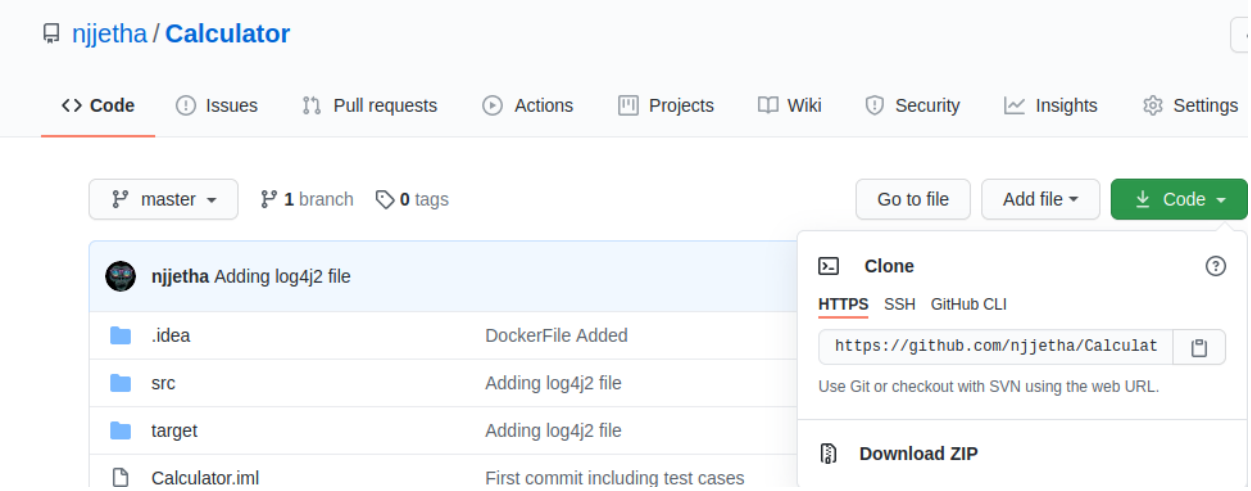
git commit → The git commit command captures a snapshot of the project's currently staged changes. ... Prior to the execution of git commit , The git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands git commit and git add are two of the most frequently used.



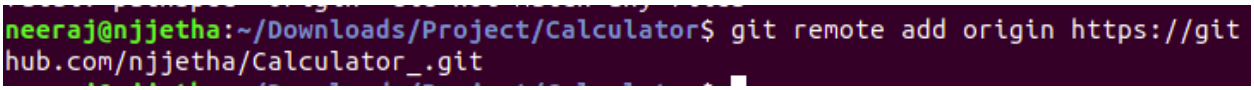
Create a new Repository name **Calculator**



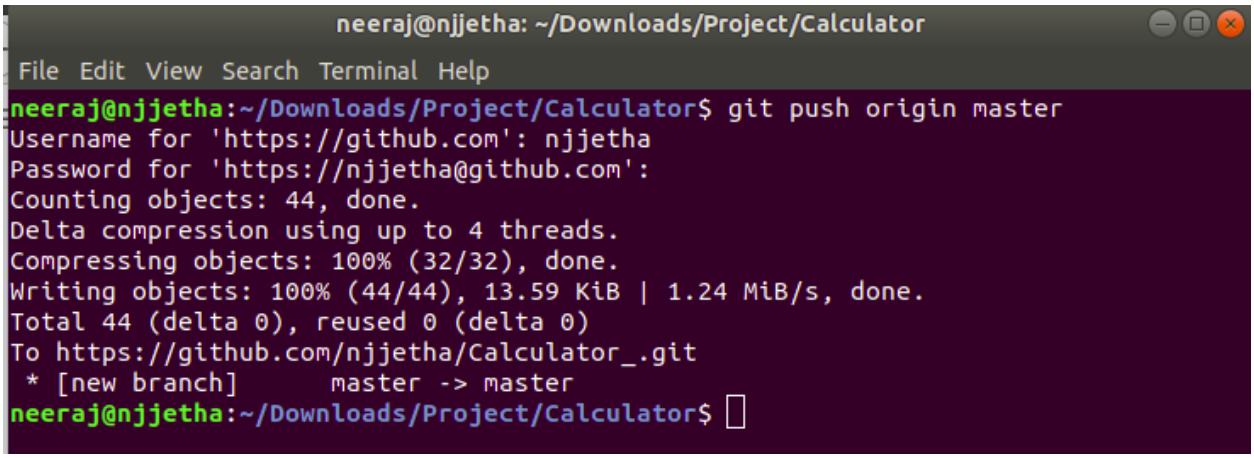
Copy the URL of the repository Calculator where we need to push our code from local repository to remote repositoty i.e Github



Setting the remote directory path as origin



Pushing the code onto the remote repository



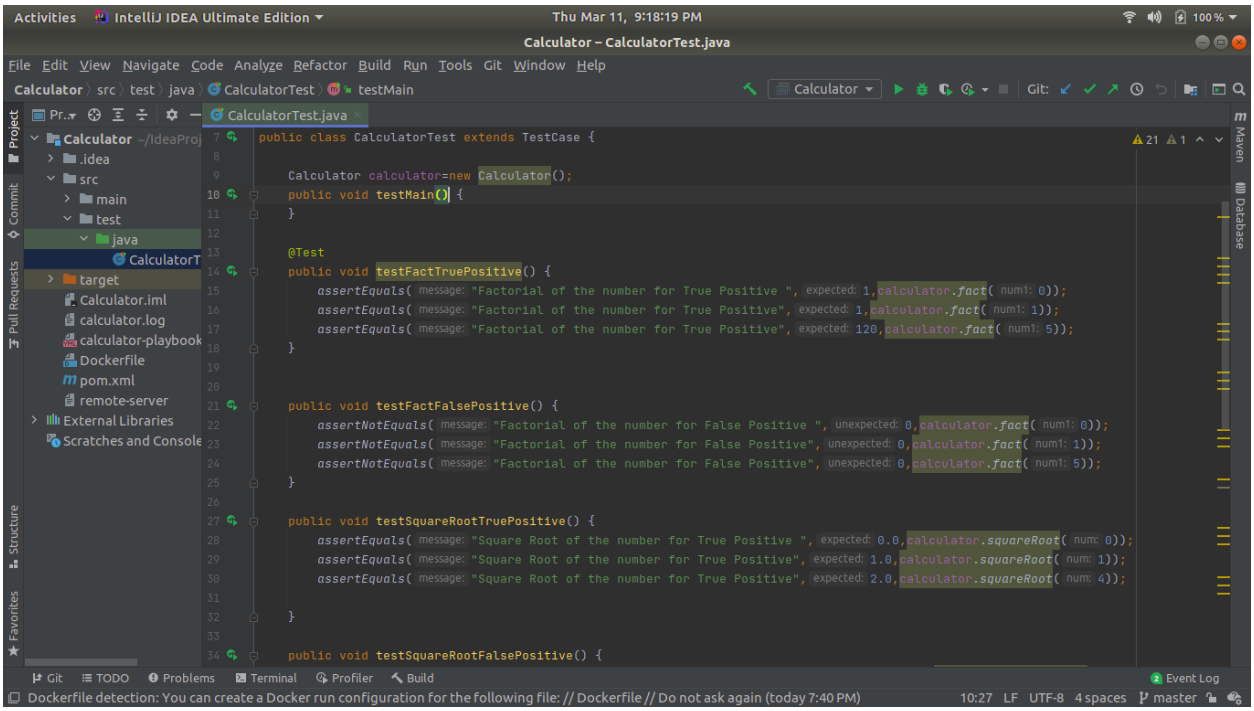
master ▾ 1 branch 0 tags Go to file Add file ▾ Code ▾

njjetha Adding log4j2 file 410db6f 8 hours ago 4 commits

.idea	DockerFile Added	19 days ago
src	Adding log4j2 file	8 hours ago
target	Adding log4j2 file	8 hours ago
Calculator.iml	First commit including test cases	21 days ago
Dockerfile	DockerFile Added	19 days ago
calculator-playbook.yml	ansible playbook added	18 days ago
calculator.log	Adding log4j2 file	8 hours ago
pom.xml	Adding log4j2 file	8 hours ago
remote-server	ansible playbook added	18 days ago

Help people interested in this repository understand your project by adding a README. Add a README

Step :- 3 JUnit Test case writing

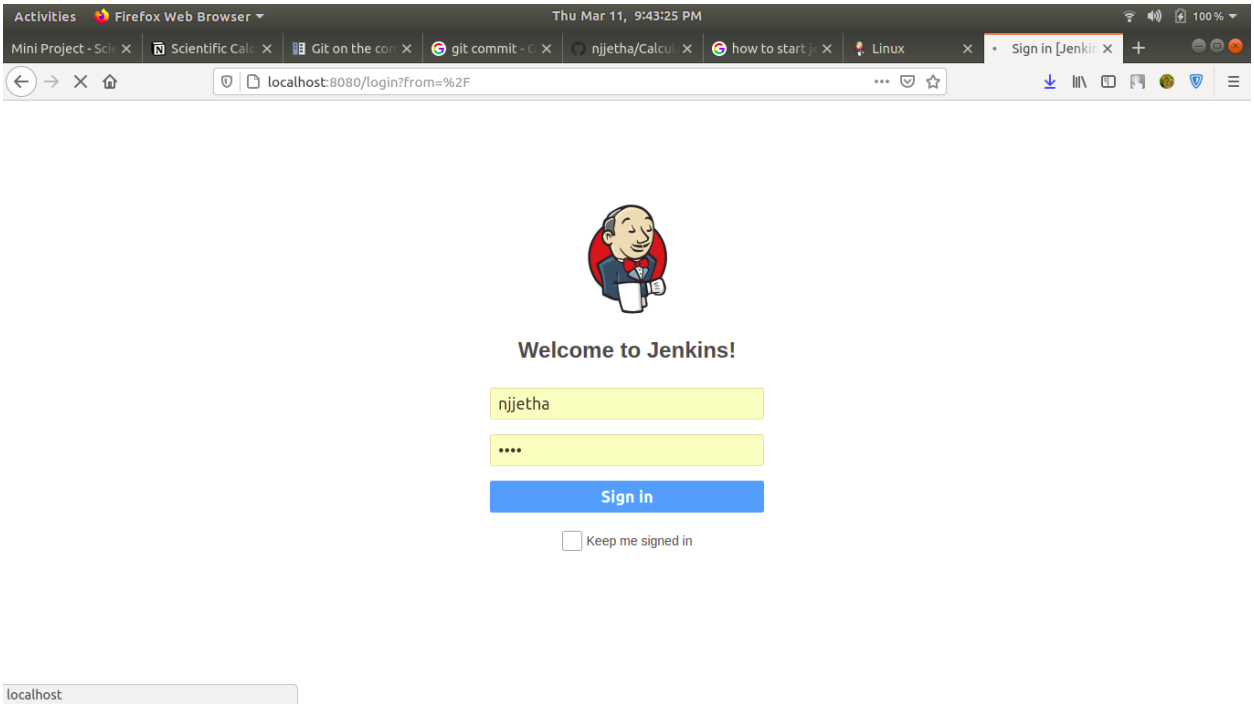


Step 4:- start Jenkins

command to start jenkins

```
neeraj@njjetha:~$ sudo systemctl start jenkins
neeraj@njjetha:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Thu 2021-03-11 02:46:34 IST; 18h ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2038 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCE

Mar 11 02:46:33 njjetha systemd[1]: Starting LSB: Start Jenkins at boot time...
Mar 11 02:46:33 njjetha jenkins[2038]: Correct java version found
Mar 11 02:46:33 njjetha jenkins[2038]:  * Starting Jenkins Automation Server jen
Mar 11 02:46:33 njjetha su[2149]: Successful su for jenkins by root
Mar 11 02:46:33 njjetha su[2149]: + ??? root:jenkins
Mar 11 02:46:33 njjetha su[2149]: pam_unix(su:session): session opened for user
Mar 11 02:46:33 njjetha su[2149]: pam_unix(su:session): session closed for user
Mar 11 02:46:34 njjetha jenkins[2038]:    ...done.
Mar 11 02:46:34 njjetha systemd[1]: Started LSB: Start Jenkins at boot time.
```

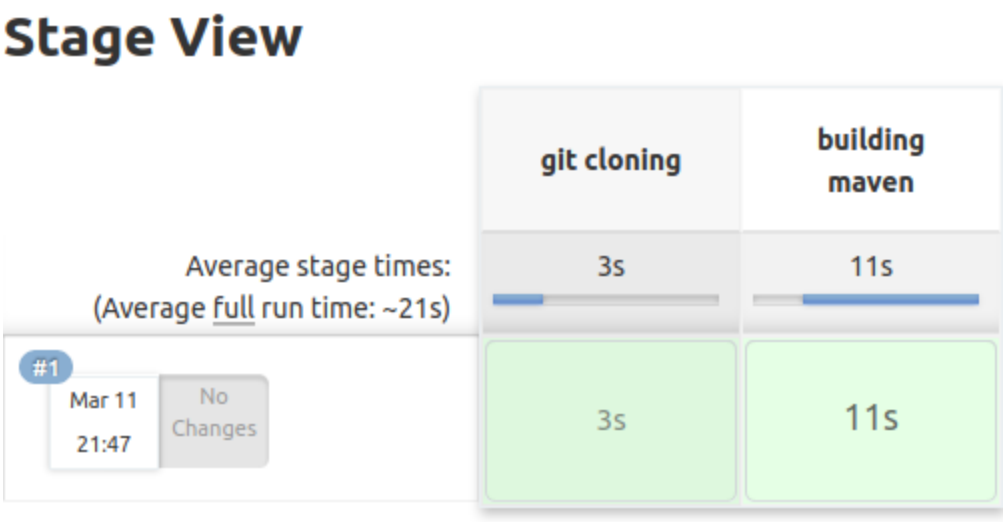


git cloning and building Maven automatically using jenkins

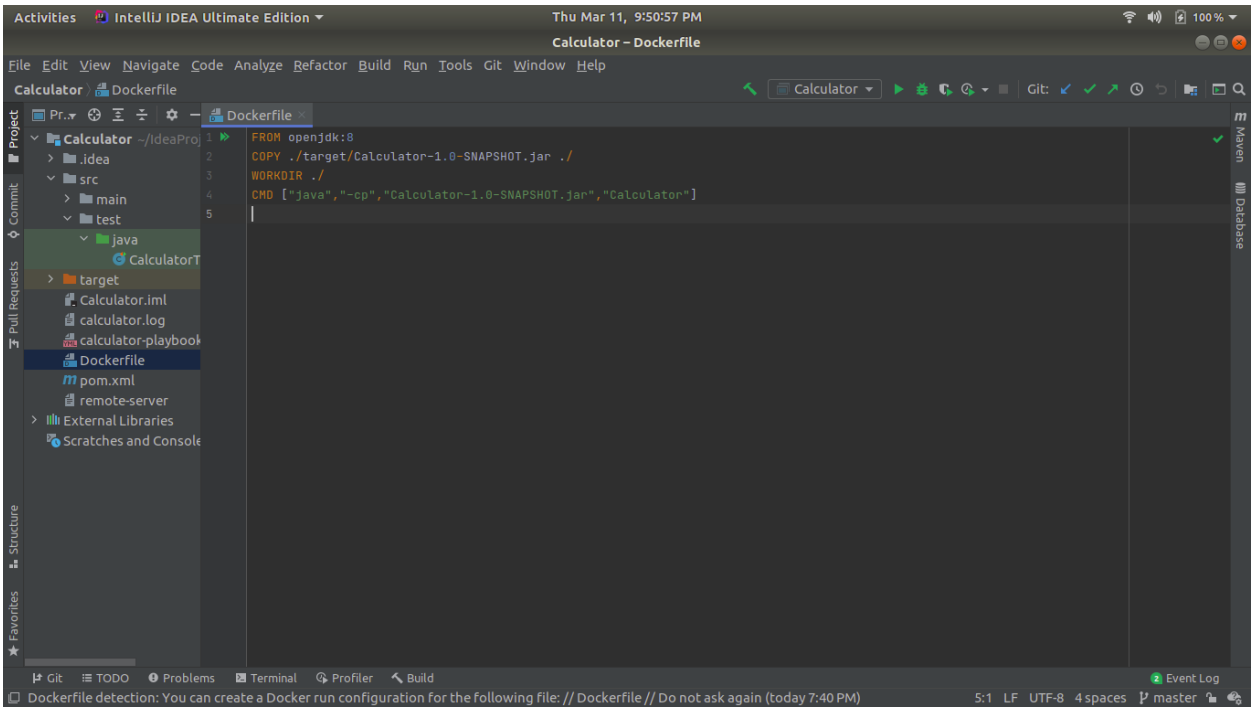
```
pipeline {
    agent any
    environment{
        imageName=""
    }
    stages {
        stage('git cloning') {
            steps {
```

```
                script{
                    git 'https://github.com/njjetha/Calculator.git'
                }
            }
        }


        stage('building maven') {
            steps {
                script{
                    sh 'mvn clean install'
                }
            }
    }
}
```

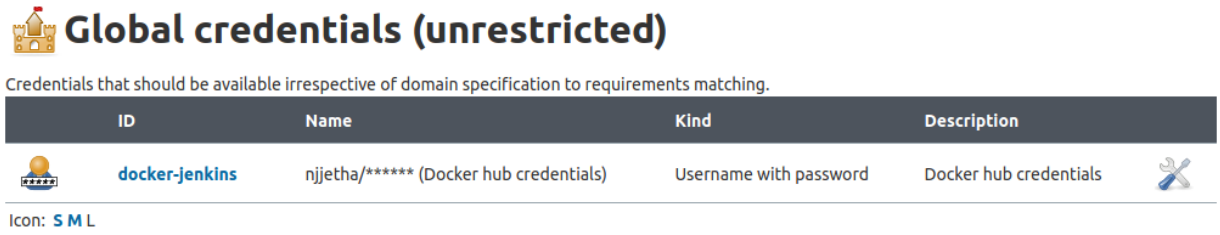


Step 5 :- Adding Docker file to the project directory



Step 6:- Adding Credentials of Docker Hub repository to the jenkins

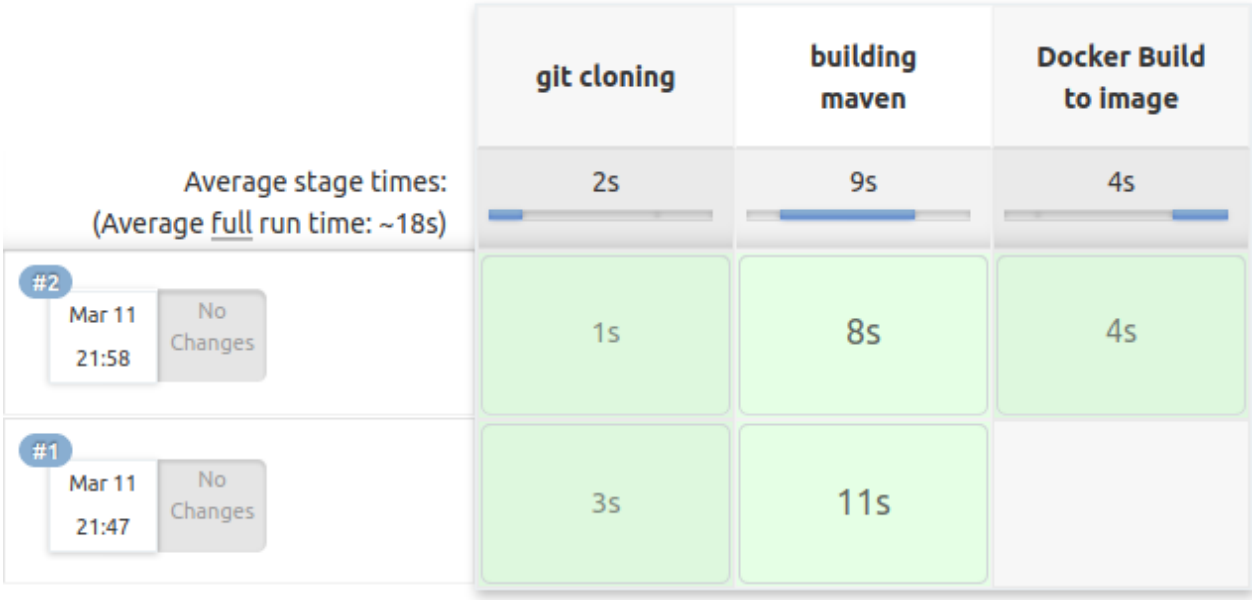
Jenkins will create the docker image and push the docker image into the docker repository



Credentials of docker hub repository added and id is docker-jenkins

Now we will write the script for creating the docker image

Stage View


git cloning
building maven
Docker Build to image
Average stage times:
(Average full run time: ~18s)
2s
9s
4s
#2
Mar 11
21:58
No Changes
1s
8s
4s
#1
Mar 11
21:47
No Changes
3s
11s

```
Sending build context to Docker daemon  227.8kB

Step 1/4 : FROM openjdk:8
 ---> 9324460525ca
Step 2/4 : COPY ./target/Calculator-1.0-SNAPSHOT.jar ./
 ---> 09d1f1fd0fc2
Step 3/4 : WORKDIR ./
 ---> Running in e9ba4f48b4d7
Removing intermediate container e9ba4f48b4d7
 ---> 2461f1e07090
Step 4/4 : CMD ["java","-cp","Calculator-1.0-SNAPSHOT.jar","Calculator"]
 ---> Running in 27916cc0c7c7
Removing intermediate container 27916cc0c7c7
 ---> 4e4269345624
Successfully built 4e4269345624
Successfully tagged njjetha/calculator-devops:latest
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Now pushing the docker image to docker hub repository

Stage View

	git cloning	building maven	Docker Build to image	Push Docker image
Average stage times: (Average full run time: ~25s)	2s	9s	2s	26s
#3 Mar 11 22:01 No Changes	1s	8s	1s	26s
#2 Mar 11 21:58 No Changes	1s	8s	4s	
#1 Mar 11 21:47 No Changes	3s	11s		

```
[Pipeline] withDockerRegistry
$ docker login -u njjetha -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/Calculator_DevOps@tmp/0e698342-db58-4aa1-b0fc
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] sh
+ docker tag njjetha/calculator-devops:latest njjetha/calculator-devops:latest
[Pipeline] isUnix
[Pipeline] sh
+ docker push njjetha/calculator-devops:latest
The push refers to repository [docker.io/njjetha/calculator-devops]
7eebbe56b93f: Preparing
02412b9dda81: Preparing
d7b2c55f7e50: Preparing
02f0a7f763a3: Preparing
da654bc8bc80: Preparing
4ef81dc52d99: Preparing
909e93c71745: Preparing
7f03bfe4d6dc: Preparing
4ef81dc52d99: Waiting
909e93c71745: Waiting
7f03bfe4d6dc: Waiting
da654bc8bc80: Layer already exists
d7b2c55f7e50: Layer already exists
02412b9dda81: Layer already exists
02f0a7f763a3: Layer already exists
909e93c71745: Layer already exists
4ef81dc52d99: Layer already exists
7f03bfe4d6dc: Layer already exists
7eebbe56b93f: Pushed
latest: digest: sha256:06922691afd203ac2a98ac43eae8b374ca9b94895191f698f9a54c7147ec976a size: 2002
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Docker hub private repository contain the image of the calculator project

Step 7:- Creating Playbook and Adding to the Project Directory

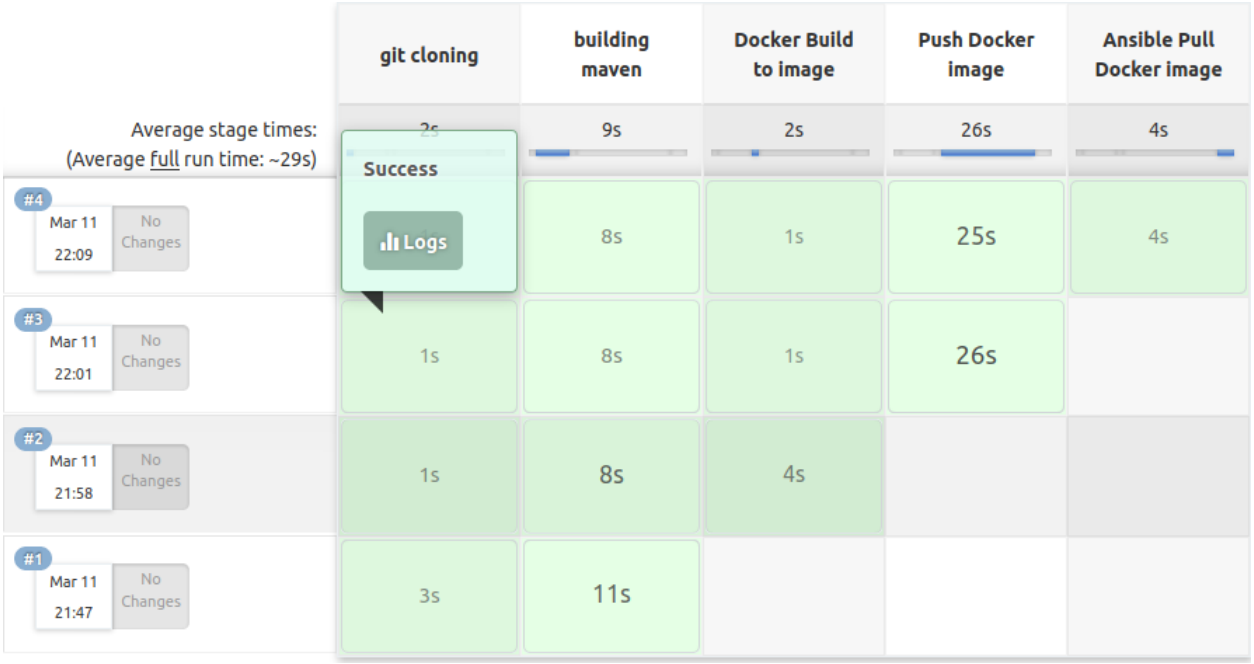
```
---
- name: Copy jar file to remoter server
```

```
  hosts: remoteserver
  tasks:
    - copy:
        src: ./target/Calculator-1.0-SNAPSHOT.jar
        dest: ~/
```

Inventory

```
[remoteserver]
127.0.1.1 ansible_user=neeraj
```

Playbook will pull the docker image into the specified machine defined in the inventory



```
[Calculator_DevOps] $ ansible-playbook calculator-playbook.yml -i remote-server

PLAY [Copy jar file to remoter server] *************************************

TASK [Gathering Facts] *****************************************************
[0;32mok: [127.0.1.1][0m

TASK [copy] ****************************************************************
[0;33mchanged: [127.0.1.1][0m

PLAY RECAP *****************************************************************
[0;33m127.0.1.1[0m                 : [0;32mok=2    [0m [0;33mchanged=1    [0m unreachable=0    failed=0

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

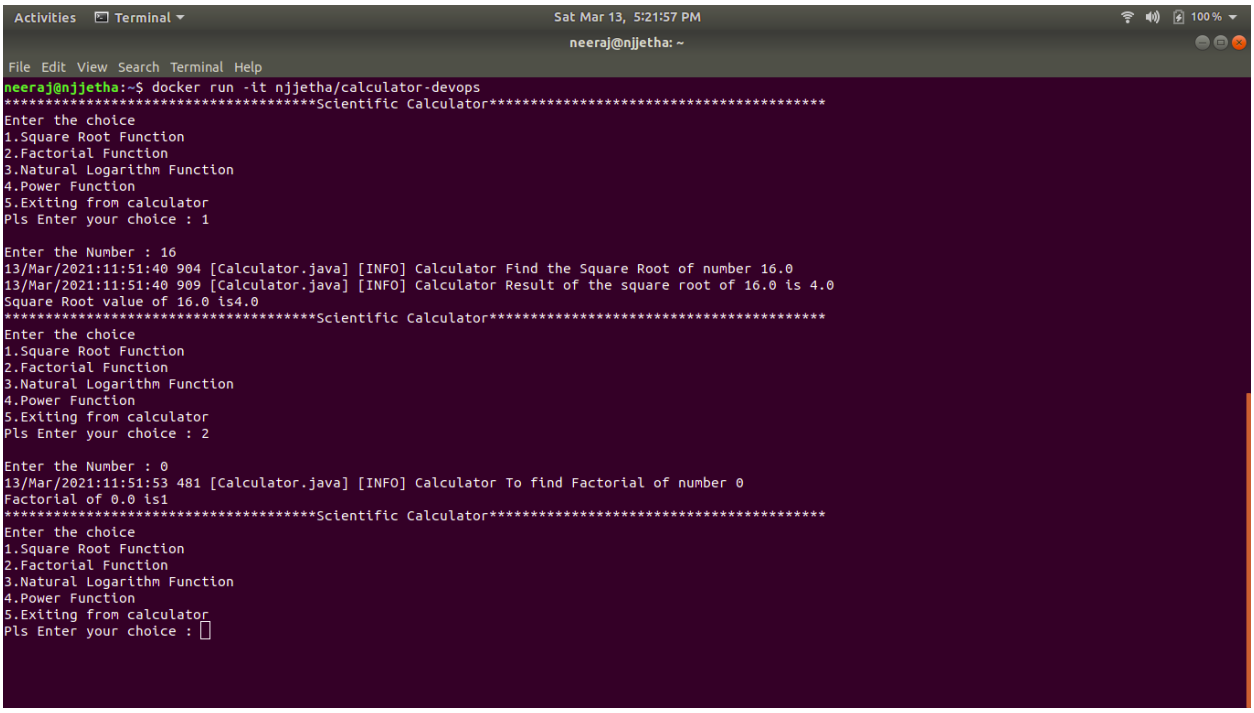
Command to see docker images is

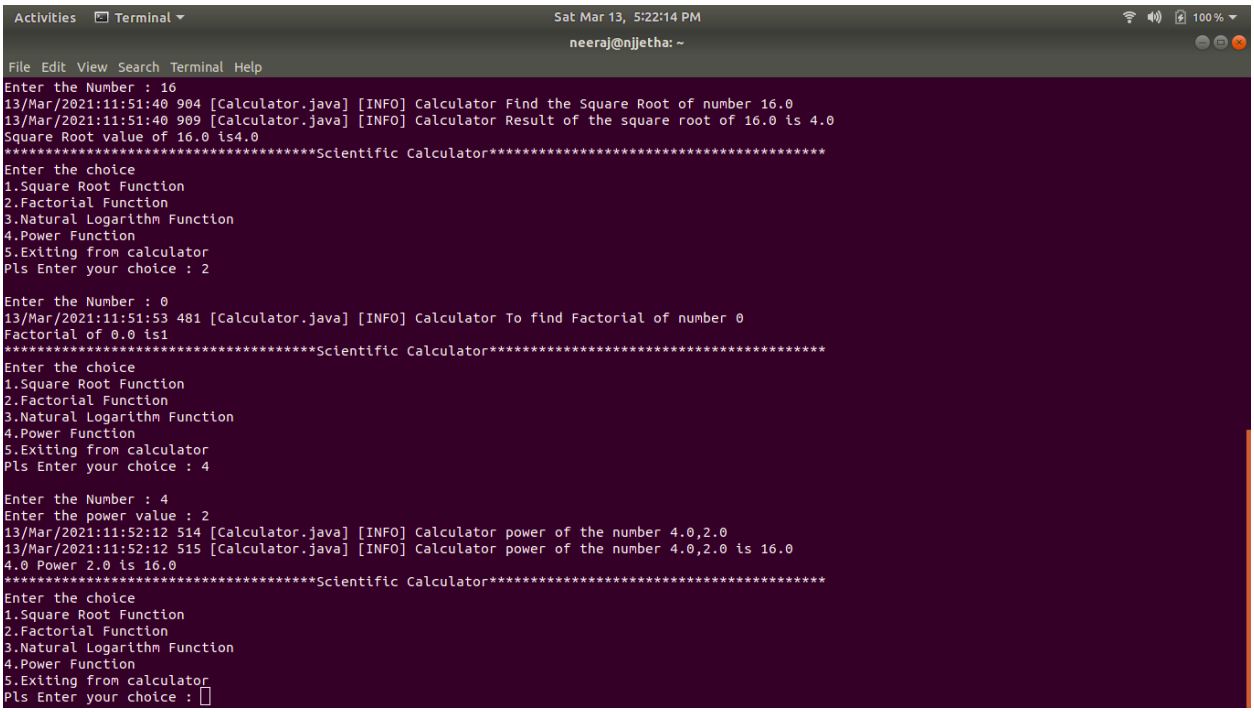
```
$ docker images
```

```
neeraj@njjetha:~$ docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
njjetha/calculator-devops   latest    f0bde47d220c   59 seconds ago   514MB
<none>                      <none>    4e4269345624   12 minutes ago   514MB
ubuntu-sleeper              latest    1ce51f10b74b   3 weeks ago      72.9MB
njjetha/ubuntu-sleeper      latest    1ce51f10b74b   3 weeks ago      72.9MB
tomcat                      latest    bf4709e77b18   4 weeks ago      667MB
openjdk                     8         9324460525ca   4 weeks ago      514MB
maven                       latest    3956fab279d0   5 weeks ago      753MB
ubuntu                      latest    f63181f19b2f   7 weeks ago      72.9MB
hello-world                 latest    bf756fb1ae65   14 months ago    13.3kB
jenkins                     2.60.3    cd14cecfdb3a   2 years ago      696MB
```

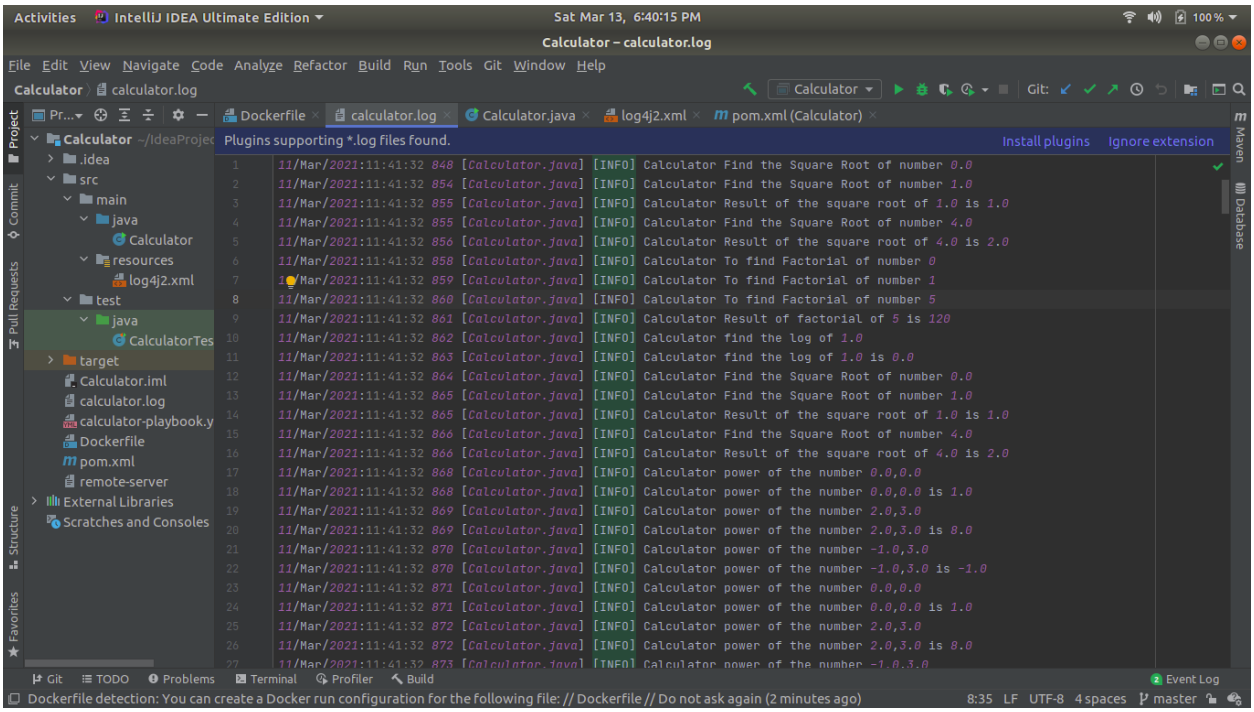
Command to run docker images is

```
docker run -it imageName
```



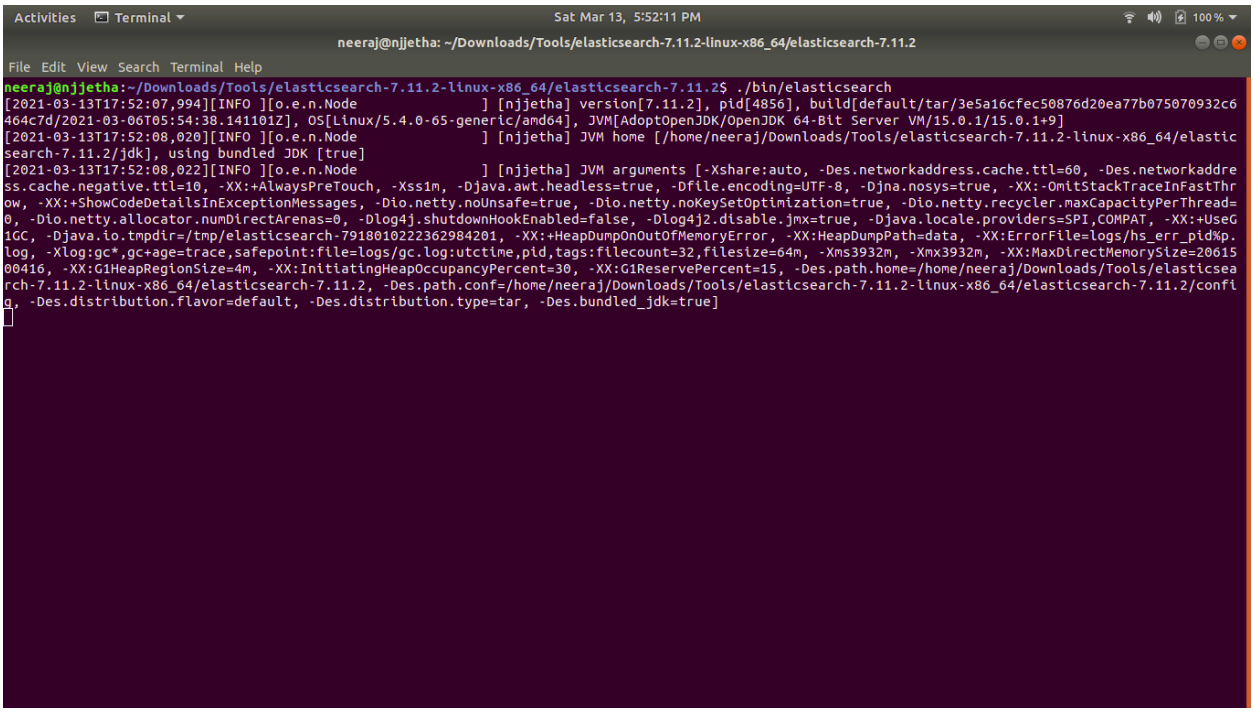


Step 8 :- The path of log file is added to the configuration file .



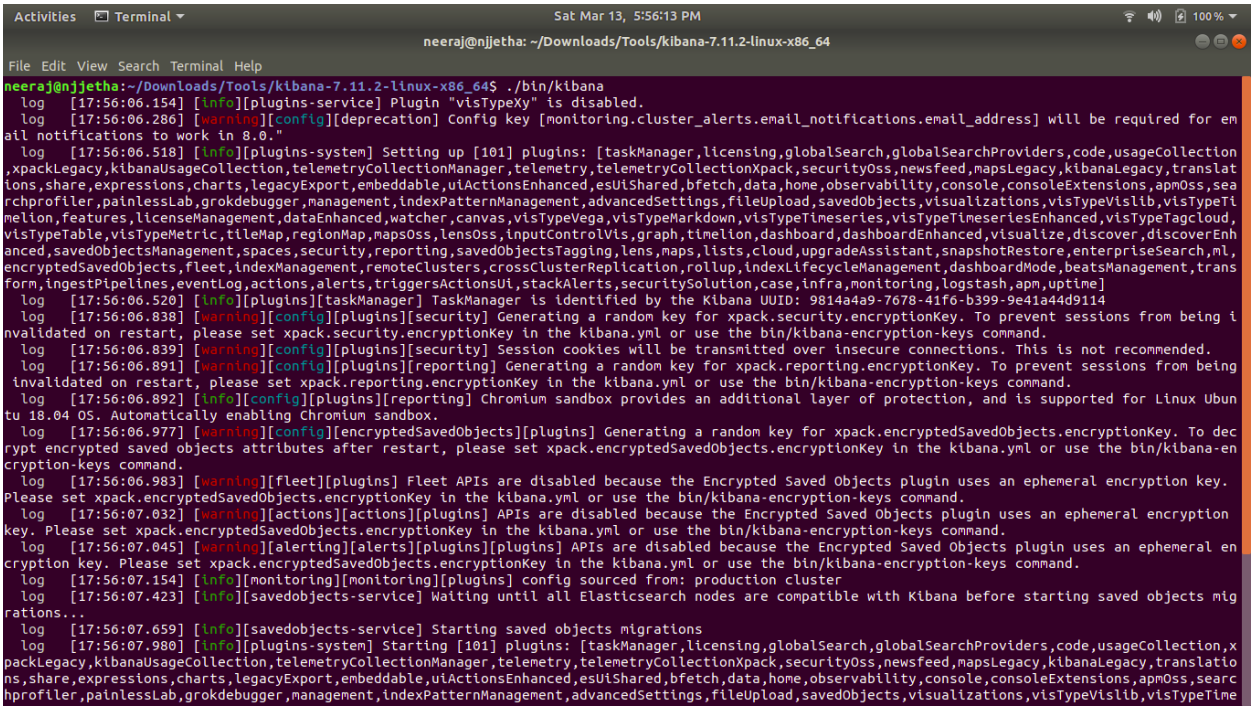
Run Elastic search

```
./bin/elasticsearch
It runs on port number 9200
```



Run Kibana

```
./bin/kibana
It runs on port number 5601
```

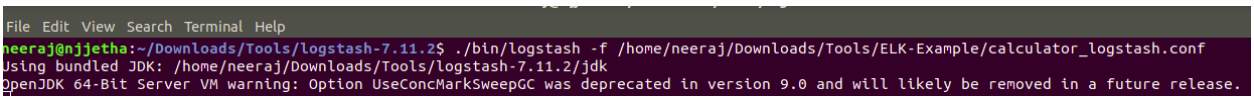


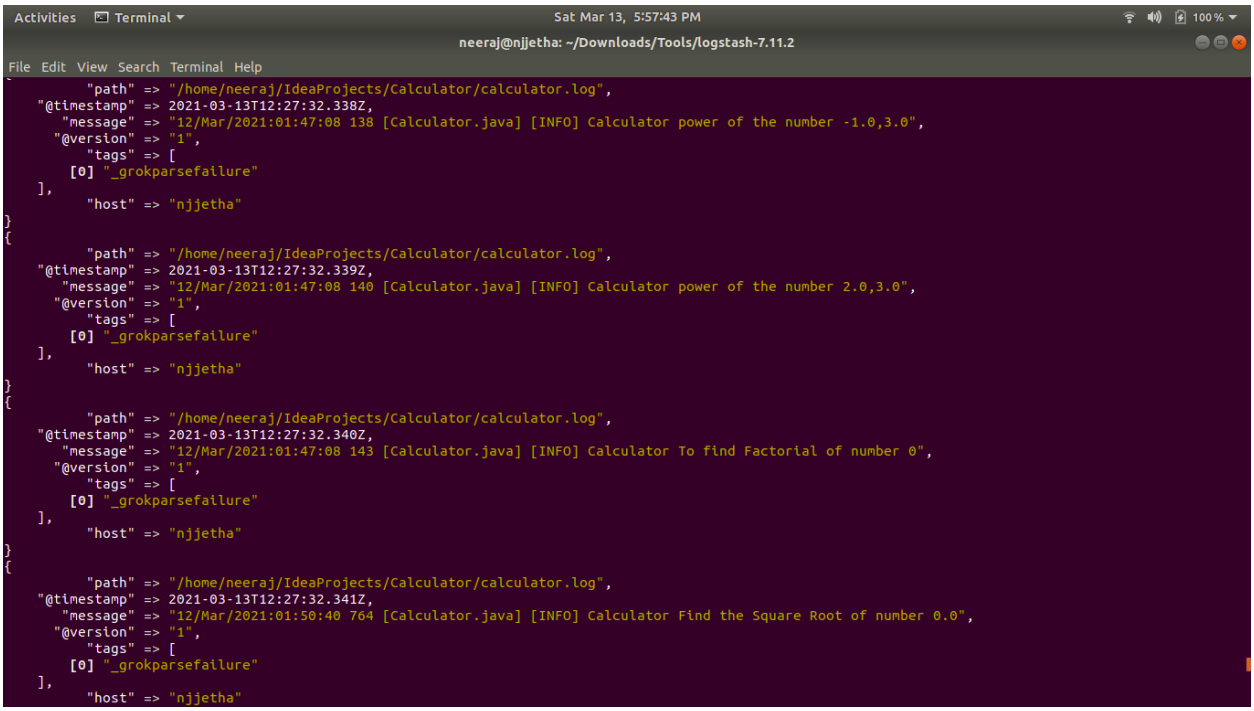
Run Logstash

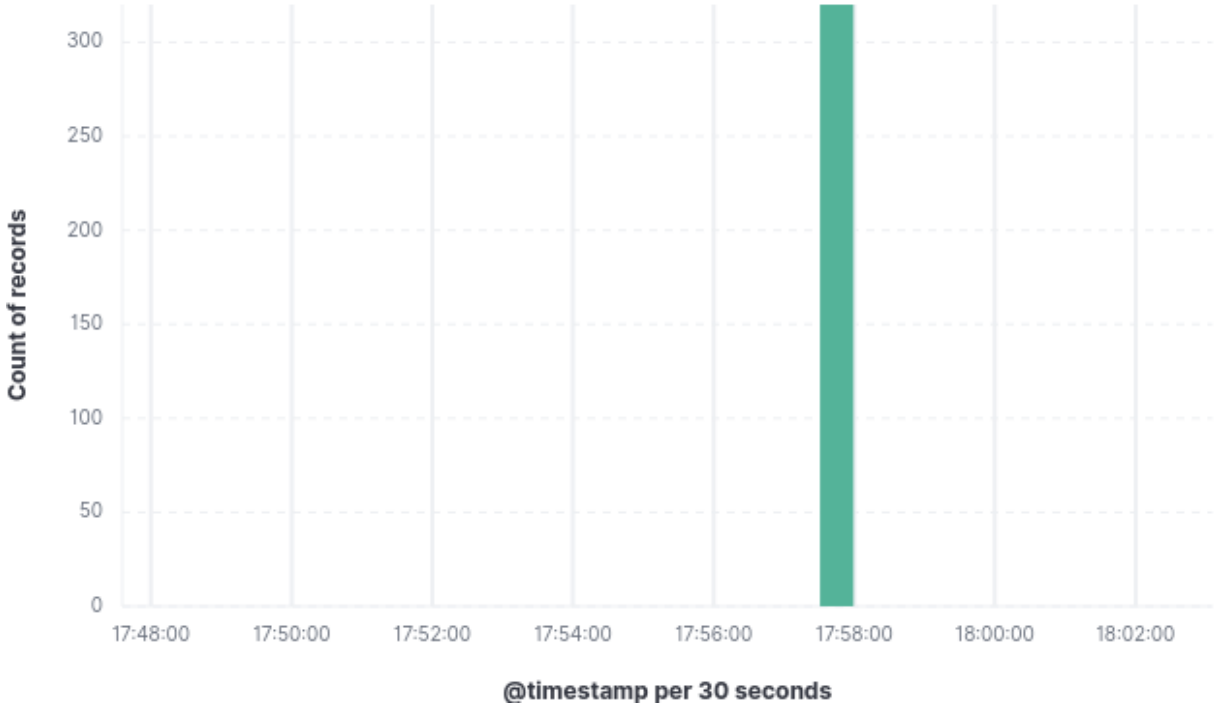
To run the logstash we have to input a configuration file on how toparse the data of a given log file, run this command after you change directory to logstash.

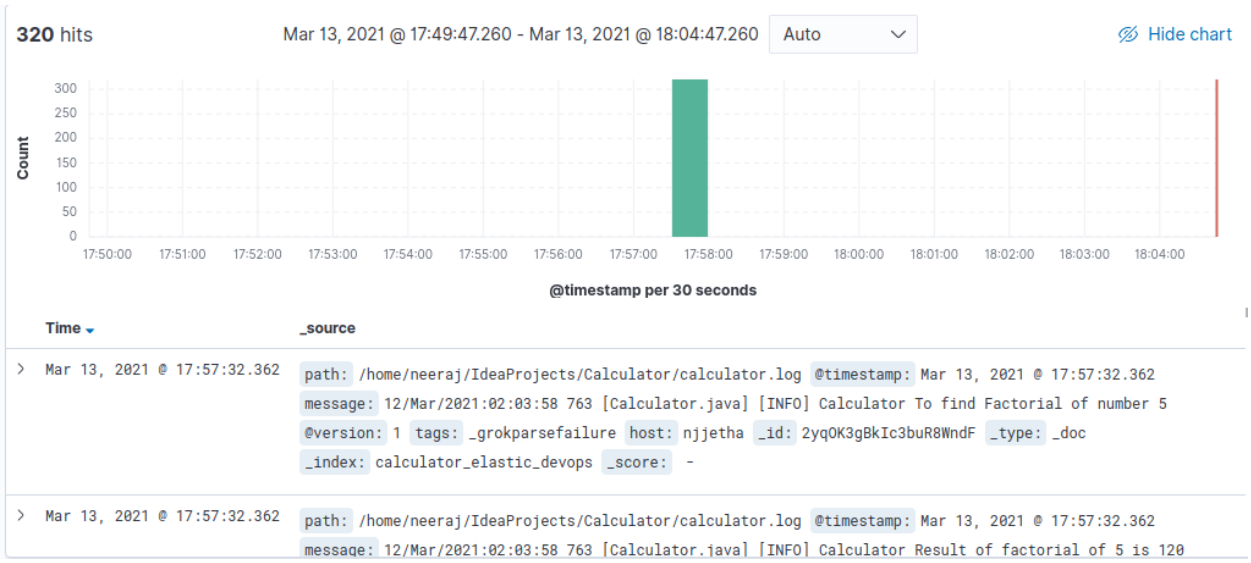
Logstash runs on port number **9600** and it forwards the filters of the log file to the elasticsearch server.

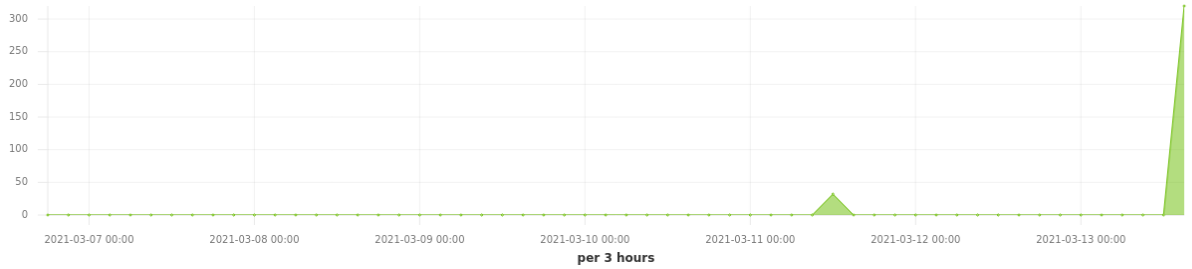
```
./bin/logstash -f /path/to/configuration/file
```




Activities
Terminal
Sat Mar 13, 5:57:43 PM
100 %
neeraj@njjetha: ~/Downloads/Tools/logstash-7.11.2
File Edit View Search Terminal Help
"path" => "/home/neeraj/IdeaProjects/Calculator/calculator.log",
"@timestamp" => 2021-03-13T12:27:32.338Z,
"message" => "12/Mar/2021:01:47:08 138 [Calculator.java] [INFO] Calculator power of the number -1.0,3.0",
"@version" => "1",
"tags" => [
[0] "_grokparsefailure"
],
"host" => "njjetha"
}
{
"path" => "/home/neeraj/IdeaProjects/Calculator/calculator.log",
"@timestamp" => 2021-03-13T12:27:32.339Z,
"message" => "12/Mar/2021:01:47:08 140 [Calculator.java] [INFO] Calculator power of the number 2.0,3.0",
"@version" => "1",
"tags" => [
[0] "_grokparsefailure"
],
"host" => "njjetha"
}
{
"path" => "/home/neeraj/IdeaProjects/Calculator/calculator.log",
"@timestamp" => 2021-03-13T12:27:32.340Z,
"message" => "12/Mar/2021:01:47:08 143 [Calculator.java] [INFO] Calculator To find Factorial of number 0",
"@version" => "1",
"tags" => [
[0] "_grokparsefailure"
],
"host" => "njjetha"
}
{
"path" => "/home/neeraj/IdeaProjects/Calculator/calculator.log",
"@timestamp" => 2021-03-13T12:27:32.341Z,
"message" => "12/Mar/2021:01:50:40 764 [Calculator.java] [INFO] Calculator Find the Square Root of number 0.0",
"@version" => "1",
"tags" => [
[0] "_grokparsefailure"
],
"host" => "njjetha"


Count of records
300
250
200
150
100
50
0
17:48:00
17:50:00
17:52:00
17:54:00
17:56:00
17:58:00
18:00:00
18:02:00
@timestamp per 30 seconds


320 hits
Mar 13, 2021 @ 17:49:47.260 - Mar 13, 2021 @ 18:04:47.260
Auto
Hide chart
Count
300
250
200
150
100
50
0
17:50:00
17:51:00
17:52:00
17:53:00
17:54:00
17:55:00
17:56:00
17:57:00
17:58:00
17:59:00
18:00:00
18:01:00
18:02:00
18:03:00
18:04:00
@timestamp per 30 seconds
Time
_source
Mar 13, 2021 @ 17:57:32.362
path: /home/neeraj/IdeaProjects/Calculator/calculator.log @timestamp: Mar 13, 2021 @ 17:57:32.362
message: 12/Mar/2021:02:03:58 763 [Calculator.java] [INFO] Calculator To find Factorial of number 5
@version: 1 tags: _grokparsefailure host: njjetha _id: 2yqOK3gBkIc3buR8WndF _type: _doc
_index: calculator_elastic_devops _score: -
Mar 13, 2021 @ 17:57:32.362
path: /home/neeraj/IdeaProjects/Calculator/calculator.log @timestamp: Mar 13, 2021 @ 17:57:32.362
message: 12/Mar/2021:02:03:58 763 [Calculator.java] [INFO] Calculator Result of factorial of 5 is 120


300
250
200
150
100
50
0
2021-03-07 00:00
2021-03-08 00:00
2021-03-09 00:00
2021-03-10 00:00
2021-03-11 00:00
2021-03-12 00:00
2021-03-13 00:00
per 3 hours