

**Mini Project on developing Scientific Calculator Program and using  
DevOps and it's tools**  
**Software Production Engineering**  
Abhinav Tiwari ([Abhinav.Tiwari@iiitb.org](mailto:Abhinav.Tiwari@iiitb.org))  
MT2020027

## CONTENTS

Introduction.....	1
Java (OpenJDK8).....	1
SCM and VCS.....	2
Testing.....	3
Maven.....	3
Containerization using Docker.....	4
Docker File.....	5
Create a DockerHub repository.....	6
Continuos Deployment using rundeck.....	7
Creating a Project in Rundeck.....	7
Creating a Job in Rundeck.....	8
Jenkins.....	9
Creating a pipeline to Integrate SCM, Build Image and Deploy using Jenkins.....	12
Onine Monitoring using ELK Stack.....	13
Setting up Elastic Cloud.....	13
Configuring Logstash with cloud services.....	14
Kibana.....	15
Visualization using Kibana.....	16
Screenshots of project components.....	18
Links.....	21
References.....	21

## INTRODUCTION :

This is a Mini Project to develop a scientific calculator program and to implement various DevOps tools and get to know about various phases of DevOps like Continuous Integration and Continuous Deployment through the project.

I have opted to develop a Java CLI program for the same.

## JAVA (OPENJDK8) :

Java is a high-level programming language developed by Sun Microsystems. Programs are interpreted by the Java Virtual Machine, or JVM, which runs on multiple platforms. This means all Java programs are multiplatform and can run on different platforms, including Macintosh, Windows, and Unix computers.

## Installation

```
$ apt-update  
$ apt install open-jdk-8-jdkConfigure $JAVA_HOME path
```

Open /etc/environment file

```
$ sudo nano /etc/environment
```

Add following path at the end

```
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/bin"
```

Restart the system to apply the changes or reload the file to apply the changes to current session.

```
$ source /etc/environment
```

Verify JAVA\_HOME path

```
$ echo $JAVA_HOME
```

## SCM and IDE :

I have used IntelliJ IDEA Ultimate IDE to write codes and test cases as well as other configuration files which generates POM.xml file after creating the project initially. This IDE provides inbuilt VCS and provide error support while writing code. GUI based tasks makes it easier for developer to write code.

## VCS(version control system)

I am using **GIT** as the VCS . Git is a distributed version control system, it is a tool to manage project source code history. Git is one of the most widely-used popular version control system in use today.

Installation:

**\$ apt update**

**\$ apt install git**

Configuration:

**\$ git config --global user.name "Your Name"**

**\$ git config --global user.email "YourEmailID@domain.com"**

**\$ git --version**

Add project into git repository using IntelliJ IDE:

Enabled the version control from VCS in IDE. Committed the changes in the IDE. Specified commit message. Pushed the changes to the git repository created by specifying the url. Pushed to the git repository previously created as the master branch.

The screenshot shows a GitHub repository page for 'CalculatorSpe'. The top navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name 'abhi93wari / CalculatorSpe' is displayed. The commit history shows 8 commits on the 'master' branch, dated from March 11, 2021, to March 15, 2021. Commits include adding Dockerfile, changes in log4j configuration, Jenkins pipeline file addition, and updates to pom.xml and calculator.log. The repository description is 'Mini Project of scientific calculator in DevOps for SPE subject'. It has 1 star and 0 forks. The 'About' section includes a link to 'Create a new release'. The 'Languages' section shows Java at 97.9% and Dockerfile at 2.1%. Below the commits, there are sections for 'Commits on Mar 15, 2021', 'Commits on Mar 12, 2021', and 'Commits on Mar 11, 2021', each listing specific commits with their authors and dates.

## TESTING :

I have used Junit 5 for testing which is the next generation of JUnit. JUnit is a unit testing framework for the Java programming language. As the usage of JUnit 5 requires multiple libraries to be present, you typically use it with a build system like Maven or Gradle. JUnit 5 needs at least Java 8 to run.

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13.1</version>
    <scope>test</scope>
</dependency>
```

A Junit *test* is a method contained in a class which is only used for testing. This is called a *Test class*. To define that a certain method is a test method, annotate it with the `@Test` annotation. This method executes the code under test. You use an *assert* method, provided by JUnit or another assert framework, to check an expected result versus the actual result. These calls are typically called *asserts* or *assert statements*.

We have to test for following situations ;

If the code is broken, but the test passes; that is a false positive, such cases should fail

If the code is correct, but the test fails; that is a false negative.

If the code is correct and the test passes; that is a true positive, such cases should pass

If the code fails and the test fails, that is a true negative.

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
calculatorspe src test java calculatortesting logarithmTruePositive
pom.xml(calculatorspe) < Calculator.java < calculatortesting.java < Dockerfile < log4j2.xml < calculator.log < Jenkinsfile < Calculator > ↻

1 import calculator.Calculator;
2 import org.junit.Test;
3
4 import static org.junit.Assert.assertEquals;
5
6 public class calculatortesting {
7     private static final double DELTA = 1e-15;
8     Calculator calc=new Calculator();
9
10 @Test
11 public void factorialTruePositive(){
12     assertEquals(message: "Checking factorial for True Positive", expected: 24.0, calc.factorial( number: 4),DELTA);
13     assertEquals(message: "Checking factorial for True Positive", Double.NaN, calc.factorial( number: -4),DELTA);
14 }
15 @Test
16 public void factorialFalsePositive(){
17     assertEquals(message: "Checking factorial for True Positive", expected: 1.0, calc.factorial( number: 4),DELTA);
18     assertEquals(message: "Checking factorial for True Positive", expected: 24.0, calc.factorial( number: -4),DELTA);
19 }
20 @Test
21 public void logarithmTruePositive(){
22     assertEquals(message: "Checking Natural log for True Positive", expected: 1.0, calc.logarithm(Math.exp(1)),DELTA);
23     assertEquals(message: "Checking Natural log for True Positive", Double.NaN, calc.logarithm( number: -1.0),DELTA);
24     assertEquals(message: "Checking Natural log for True Positive", Double.NaN, calc.logarithm( number: 0.0),DELTA);
25 }
26 @Test
27 public void logarithmFalsePositive(){
28     assertEquals(message: "Checking Natural log for True Positive", Double.NaN, calc.logarithm(Math.exp(1)),DELTA);
29     assertEquals(message: "Checking Natural log for True Positive", expected: 1.0, calc.logarithm((-1)*Math.exp(1)),DELTA);
30     assertEquals(message: "Checking Natural log for True Positive", expected: 0.0, calc.logarithm( number: 0.0),DELTA);
31 }
32 @Test
33 public void powerTruePositive(){
34     assertEquals(message: "Checking x raised to y for True Positive", expected: 16.0, calc.power(4,2),DELTA);
35     assertEquals(message: "Checking x raised to y for True Positive", Double.NaN, calc.power(-4,2.3),DELTA);
36 }
```

## MAVEN :

Maven is a project development management and comprehension tool. Based on the concept of a project object model: builds, dependency management, documentation creation, site publication, and distribution publication are all controlled from the pom.xml declarative file. Maven can be extended by plugins to utilise a number of other development tools for reporting or the build process.

Installation

```
$ apt-install maven
```

```
$ mvn -version
```

Run maven commands from project root directory.

```
$ mvn clean
```

```
$ mvn compile
```

```
$ mvn test
```

```
$ mvn install
```

I have builded my project using IDE.

```

[INFO] [org.apache.maven.DefaultBuildPluginManager] skipping
[INFO] [org.apache/logging/logging] already added, skipping
[INFO] [org.apache/logging/log4j] already added, skipping
[INFO] [META-INF/versions/] already added, skipping
[INFO] [META-INF/versions/9/] already added, skipping
[INFO] [META-INF/versions/9/org/apache/] already added, skipping
[INFO] [META-INF/versions/9/org/apache/logging/] already added, skipping
[INFO] [META-INF/services/] already added, skipping
[INFO] [META-INF/maven/] already added, skipping
[INFO] [META-INF/maven/org.apache.logging.log4j/] already added, skipping
[INFO] [META-INF/DEPENDENCIES] already added, skipping
[INFO] [META-INF/LICENSE] already added, skipping
[INFO] [META-INF/NOTICE] already added, skipping
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ calculatorspe ---
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.pom (19 kB at 20 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom (19 kB at 20 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom (19 kB at 20 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-components-1.1.7.pom (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (1.0 kB at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar (1.0 kB at 1.0 kB/s)
[INFO] Installing /home/abhinav/calculatorspe/target/calculatorspe-1.0-SNAPSHOT.jar to /root/.m2/repository/com/calculator/calculatorspe/1.0-SNAPSHOT/calculatorspe-1.0-SNAPSHOT.jar-with-dependencies.jar
[INFO] Installing /home/abhinav/calculatorspe/pom.xml to /root/.m2/repository/com/calculator/calculatorspe/1.0-SNAPSHOT/calculatorspe-1.0-SNAPSHOT.jar-with-dependencies.jar
[INFO] Installing /home/abhinav/calculatorspe/target/calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar to /root/.m2/repository/com/calculator/calculatorspe/1.0-SNAPSHOT/calculatorspe-1.0-SNAPSHOT.jar-with-dependencies.jar
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:09 min
[INFO] Finished at: 2021-03-15T02:47:04+05:30
[INFO] -----
root@abhinav-X541UJ:/home/abhinav/calculatorspe#

```

The screenshot shows an IDE interface with several tabs open: Calculator.java, calculatortesting.java, Dockerfile, log4j2.xml, calculator.log, and Jenkinsfile. On the right side, there is a Maven tool window displaying the project structure under 'calculator' and the 'Lifecycle' tab, which includes options like clean, validate, compile, test, package, and verify. Below the tool window, a terminal window displays the Maven build logs:

```

.7660.26/plugins/maven/lib/maven3/bin/m2.conf -Dmaven.ext.class.path=/home/abhinav/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/202.7660.26/plugins/maven/lib/maven-event-listener.jar -javaagent:/home/abhinav/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/202.7660.26/lib/idea_rt.jar=35099:/home/abhinav/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/202.7660.26/bin -Dfile.encoding=UTF-8 -classpath /home/abhinav/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/202.7660.26/plugins/maven/lib/maven3/boot/plexus-classworlds.lib.license=/home/abhinav/.local/share/JetBrains/Toolbox/apps/IDEA-U/ch-0/202.7660.26/plugins/maven/lib/maven3/boot/plexus-classworlds-2.6.0.jar org.codehaus.classworlds.Launcher -Didea.version=2020.2.3 compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----- com.calculator:calculatorspe <-----
[INFO] Building calculatorspe 1.0-SNAPSHOT
[INFO] ----- [ jar ] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ calculatorspe ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ calculatorspe ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] ----- com.calculator:calculatorspe <-----
[INFO] BUILD SUCCESS
[INFO]
[INFO] ----- Total time: 2.143 s
[INFO] Finished at: 2021-03-15T02:49:26+05:30
[INFO] -----

```

## CONTAINERIZATION(Docker Engine) :

I am using docker engine to implement containerization. Docker Engine is an open source containerization technology for building and containerizing your applications. Docker Engine acts as a client-server application with:

-A server with a long-running daemon process dockerd.

-APIs which specify interfaces that programs can use to talk to and instruct the Docker daemon.

-A command line interface (CLI) client docker.

The CLI uses Docker APIs to control or interact with the Docker daemon through scripting or direct CLI commands. Many other Docker applications use the underlying API and CLI. The daemon creates and manage Docker objects, such as images, containers, networks, and volumes.

### Installation :

Update the apt package index, and install the latest version of Docker Engine and containerd :

**\$ sudo apt update**

**\$sudo apt-get update**

**\$sudo apt-get install docker-ce docker-ce-cli containerd.io**

Check if the status of docker daemon is running to ensure docker is installed by running the hello-world image.(This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.)

**\$service docker status**

**\$sudo docker run hello-world**

Executing Docker commands without sudo

Add your username to the docker group

**\$ sudo usermod -aG docker \${USER}**

**\$ su - \${USER}**

Confirm that your user is now added to the docker group by typing:

**\$ id -nG**

```
root@abhinav-X541UJ:/home/abhinav# docker container run -it --name container2 abhinav6636/calculator_container
*****
Choose to perform operation
Press 1 to find factorial of a number
Press 2 to find natural log of a number
Press 3 to find value of number x raised to y
Press 4 to find square root of a number
Press any other key to exit
Enter your choice: 1
Enter the number : 7
14/Mar/2021:21:31:08 667 [calculator.java] [INFO] calculator.Calculator [FACTORIAL] - 7!
14/Mar/2021:21:31:08 725 [calculator.java] [INFO] calculator.Calculator [RESULT - FACTORIAL] - 5040.0
Factorial of the number is : 5040.0

*****
Choose to perform operation
Press 1 to find factorial of a number
Press 2 to find natural log of a number
Press 3 to find value of number x raised to y
Press 4 to find square root of a number
Press any other key to exit
Enter your choice: 1
14/Mar/2021:21:31:11 664 [calculator.java] [ERROR] calculator.Calculator Invalid input, Entered input is not a number
root@abhinav-X541UJ:/home/abhinav# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
495e097f5e73        abhinav6636/calculator_container   "java -jar calculato..."   About a minute ago   Up About a minute
root@abhinav-X541UJ:/home/abhinav# docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
7618d50cd1dc        abhinav6636/calculator_container   "java -jar calculato..."   44 seconds ago    Exited (0) 30 seconds ago   container2
495e097f5e73        abhinav6636/calculator_container   "java -jar calculato..."   2 minutes ago     Up About a minute      container1
034ca0c8c036        calculator           "java -jar calculato..."   3 days ago       Exited (0) 3 days ago    calculator_container2
4dee8c3e04c2        abhinav6636/calculator           "java -jar calculato..."   4 days ago       Exited (255) 3 days ago   calc_container
9d4561bd03f         test                "/bin/bash"          4 days ago       Exited (255) 3 days ago   app4
31080b35e30d        ubuntu              "/bin/bash"          5 days ago       Exited (0) 5 days ago    ubuntu_container
8483e6e2e104        test                "/bin/bash"          5 days ago       Exited (0) 5 days ago    app2
bf3eb0e9565         test                "/bin/bash"          5 days ago       Exited (0) 5 days ago    app1
56ff08244dac        test                "/bin/bash"          5 days ago       Exited (1) 5 days ago    testing
root@abhinav-X541UJ:/home/abhinav# 
```

Docker basic cmd's:

**\$docker pull ubuntu** - pull docker images from dockerhub

**\$docker images ls**- List all the docker images

**\$docker create --name <container-name> <image-name>** - create the container

**\$docker container run -i -t --name user1 ubuntu** -run the container using interactive terminal

**\$docker container run -i -t -d --name user1 ubuntu** -run the container in background using detach option

**\$docker container ls** -list all running containers

**\$docker start user1** -start container

**\$docker stop user1** -stop container

**\$docker attach user1** -to attach your terminal's standard input, output to a running container

Copy file from host to docker viceversa:

**\$docker cp <file> <container-name>:<file-name> (host to docker)**

**\$docker cp user1:/user.txt /home/abhinav/user.txt (docker to host)**

## Docker File

Create a Docker file to add calculator project's jar into new image from base image(here alpine) which has jdk8 installed in it. And add it to git root directory.

# Set the base image (alpine is the light weight base image for linux,72.5MB)

**FROM** openjdk:8-alpine

# File Author

**MAINTAINER** Abhinav

# Set Working Directory

**WORKDIR** .

# Copies the files from the source on the host into the container's set destination

**COPY ./target/calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar ./**

# Default container command to execute the program

**CMD ["java","-jar","calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar"]**

```

Step 1/5 : FROM openjdk:8-alpine
--> a3562aa0b991
Step 2/5 : MAINTAINER Abhinav
--> Using cache
--> cdf8633c9776
Step 3/5 : COPY ./target/calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar .
--> e614083fc7fe
Step 4/5 : WORKDIR .
--> Running in 05adb9c8d4d1
Removing intermediate container 05adb9c8d4d1
Step 5/5 : CMD ["java","-jar","calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar"]
--> 28f6c2f98b1c
Removing intermediate container 28f6c2f98b1c
--> c582254d906a
Successfully built c582254d906a
Successfully tagged calculator:latest
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker image ls
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
calculator          latest        c582254d906a   7 seconds ago      107MB
abhinav6636/calc  latest        8df6f336c13d0   37 hours ago     107MB
test               latest        3674df3e0dc8   2 days ago       107MB
ubuntu              latest        40d9f96f02    7 days ago      72.7MB
alpine              latest        28f6e2705743   3 weeks ago      5.61MB
openjdk              8-alpine     a3562aa0b991   22 weeks ago     105MB
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker image rm openjdk
Error: No such image: openjdk
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker image rm 8-alpine
Error: No such image: 8-alpine
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker image rm a3562aa0b991
Error response from daemon: conflict: unable to delete a3562aa0b991 (cannot be forced) - image has dependent child images
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker container run -itd --name calculator_container calculator
270f0a6545852d52d77a86620e603dee69ef3d235352f01893cf6ea1970cee
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
270f0a654585        calculator          "java -jar calculato..."   12 seconds ago     Up 8 seconds          calculator_container
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker push abhinav6636/calculator_container
The push refers to repository [docker.io/abhinav6636/calculator_container]
30d163450cdb: Pushing [=====>] 2.037MB
ceaf9e1ebef5: Mounted from abhinav6636/calc
9b9b7f3d56a0: Mounted from abhinav6636/calc
f1b5933fe4b5: Mounted from abhinav6636/calc
[...]

```

## Create a DockerHub repository

Docker Hub is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub.

Here, we can find official images created by companies as well as customized images from different users. We can create our own public repository and upload our docker images there.

Creating an account at DockerHub and creating a repository

Create account in <https://hub.docker.com/> and sign in then create a repository, or you can create a repository using docker.

**\$ docker login**

**give docker username and password when prompted**

**\$ docker tag <image-name> <docker userid>/<image-name>**

**\$ docker push <docker userid>/<image-name>**

```

The terminal view Search Terminal Help
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
270f0a654585        calculator          "java -jar calculato..."   12 seconds ago     Up 8 seconds          calculator_container
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker tag calculator abhinav6636/calculator_container
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker push abhinav6636/calculator_container
The push refers to repository [docker.io/abhinav6636/calculator_container]
30d163450cdb: Pushed
ceaf9e1ebef5: Mounted from abhinav6636/calc
9b9b7f3d56a0: Mounted from abhinav6636/calc
f1b5933fe4b5: Mounted from abhinav6636/calc

latest: digest: sha256:dca873277bb701954e08855d3f4ee3ffd1bafb04a7f0b2ee5bf64a9f7ea33bc0 size: 1158
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker tag calculator abhinav6636/calculator_container
root@abhinav-X541UJ:/home/abhinav/calculatorspe# docker push abhinav6636/calculator_container
The push refers to repository [docker.io/abhinav6636/calculator_container]
30d163450cdb: Pushed
ceaf9e1ebef5: Pushed
9b9b7f3d56a0: Pushed
f1b5933fe4b5: Pushed
latest: digest: sha256:dca873277bb701954e08855d3f4ee3ffd1bafb04a7f0b2ee5bf64a9f7ea33bc0 size: 1158
root@abhinav-X541UJ:/home/abhinav/calculatorspe# [...]

```

## CONTINUOUS DEPLOYMENT using RUNDECK :

Rundeck is an open source automation service with a web console, command line tools and a WebAPI. Rundeck allows you to run tasks on any number of nodes from a web-based or command-line interface. Rundeck also includes other features that make it easy to scale up your automation efforts including: access control, workflow building, scheduling, logging, and integration with external sources for node and option data.

Installation(Open Source rundeck ,in root mode)

```
$ sudo echo "deb https://rundeck.bintray.com/rundeck-deb /" | sudo tee -a
/etc/apt/sources.list.d/rundeck.list
$ sudo curl 'https://bintray.com/user/downloadSubjectPublicKey?username=bintray' | sudo
  apt-key add -
$ sudo apt-get update
$ sudo apt-get install rundeck
```

To start Rundeck:

**\$ sudo service rundeckd start**

To verify that the service started correctly, tail the logs:

**\$ tail -f /var/log/rundeck/service.log**

The service is ready once you see something similar to:Grails application running at <http://localhost:4440> in environment: production

### Logging in for the first time

Navigate to <http://localhost:4440> in a browser.

Log in with the username admin and password admin

Rundeck is now up and running!

### Creating a Project in Rundeck

Configuring the project

Create project ==> Give name of project, label ==> Press create.

I am running rundeck and deploying in my localhost but if we want to deploy in remote VM we can add node in following manner

Edit Nodes ==> Sources (for adding nodes) ==> From file ==> Specify Format : resourcexml

File path: /var/lib/rundeck/projects/CalculatorDevOps/etc/resources.xml

Select checkboxes :Generate, Include server node and Writeable ==>Save.

File gets generated ==> Go in Edit tab ==> Insert Node tag with following details and Save

```

<?xml version="1.0" encoding="UTF-8"?>
<project>
  <node name="client" description="Client node" tags="" hostname="172.17.0.5"
osArch="amd64" osFamily="unix" osName="Linux" osVersion="5.4.0-66-generic"
username="root" sudo-command-enabled="true" sudo-password-
option="option.sudoPassword"/>
</project>

```

Test whether nodes were added successfully

Commands tab on left ==> Choose node => Execute command: uname -a ==> Run.

```

<?xml version="1.0" encoding="UTF-8"?>
<project>
  <node name="client" description="Client node" tags="" hostname="172.17.0.5"
osArch="amd64" osFamily="unix" osName="Linux" osVersion="5.4.0-66-generic"
username="root" sudo-command-enabled="true" sudo-password-
option="option.sudoPassword"/>
</project>

```

## Create a job in Rundeck

This job will run inside node(container) which have been configured be it localhost or any virtual machine

Project ==> Jobs ==> Job actions ==> New Job ==> Job Name

Go to workflow tab ==> Add steps ==> As commands

> **docker rm -f calculator\_container (remove any existing calculator\_container)**

> **docker pull abhinav6636/calculator\_container (pull the image from dockerhub into host node)**

CalculatorDevOps					
10 Executions any time <small>Search...</small>					
<span>✓</span>	03/12/2021 1:45 AM	Last Friday at 1:45 AM	1 ok	9 seconds	by admin calculator
<span>✗</span>	03/12/2021 1:43 AM	Last Friday at 1:43 AM	1 failed	1 seconds	by admin calculator
<span>✓</span>	03/12/2021 1:25 AM	Last Friday at 1:25 AM	1 ok	15 seconds	by admin calculator
<span>✓</span>	03/11/2021 9:49 PM	Last Thursday at 9:49 PM	1 ok	12 seconds	by admin calculator
<span>✓</span>	03/11/2021 9:14 PM	Last Thursday at 9:14 PM	1 ok	9 seconds	by admin calculator
<span>✗</span>	03/11/2021 9:13 PM	Last Thursday at 9:13 PM	1 failed	17 seconds	by admin calculator
<span>✓</span>	03/11/2021 8:56 PM	Last Thursday at 8:56 PM	1 ok	14 seconds	by admin calculator
<span>✓</span>	03/11/2021 6:14 PM	Last Thursday at 6:14 PM	1 ok	25 seconds	by admin calculator
<span>✓</span>	03/11/2021 4:58 PM	Last Thursday at 4:58 PM	1 ok	7 seconds	by admin calculator
<span>✗</span>	03/11/2021 4:57 PM	Last Thursday at 4:57 PM	1 failed	27 seconds	by admin calculator

Edit Job: calculator e01fe93a-b715-4c6e-9d07-70671f2ce91e

Details Workflow Nodes Schedule Notifications Other

Options Undo Redo Revert All Changes

No Options + Add an option

Workflow If a step fails:

Stop at the failed step.  Run remaining steps before failing.

Strategy: Node First

Execute all steps on a node before proceeding to the next node.

Explain >

Global Log Filters + add

1. ✓ docker pull abhinav6636/calculator\_container pulling image from dockerhub

2. ✓ docker container run -itd --name container2 abhinav6636/calculator\_container running container in background in localhost

+ Add a step

Go to Nodes tab ==> select execute locally(default) if you are deploying in localhost

Go to Nodes tab ==> select Dispatch to Nodes ==> Search Node ==> Select ,if you are deploying in remote nodes

finally, create the Job.

Note the JobId of the job created. It will be required to trigger from Jenkins.

## JENKINS :

Jenkins is an open-supply automation server that automates the repetitive technical obligations worried withinside the non-stop integration and shipping of software program with plugins constructed for Continuous Integration purpose. Jenkins is used to construct and check your software program tasks constantly making it less complicated for builders to combine modifications to the project, and making it less complicated for customers to attain a sparkling construct. It additionally lets in you to constantly supply your software program through integrating with a large quantity of checking out and deployment technologies. With Jenkins, you may iterate and set up new code as fast as possible. It additionally lets in you to degree the fulfillment of every step of your pipeline.

## Installation :

Add the key

```
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | apt-key add -
```

When the key is added the system will return a response OK.

Add the repository, update local package index and install

```
$ sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

When both of these are in place, run update so that apt will use the new repository and finally install jenkins:

```
$ apt update
```

```
$ apt install jenkins
```

Starting Jenkins

```
$ sudo systemctl start jenkins
```

Check if it is active

```
$ sudo systemctl status jenkins (it should show status active)
```

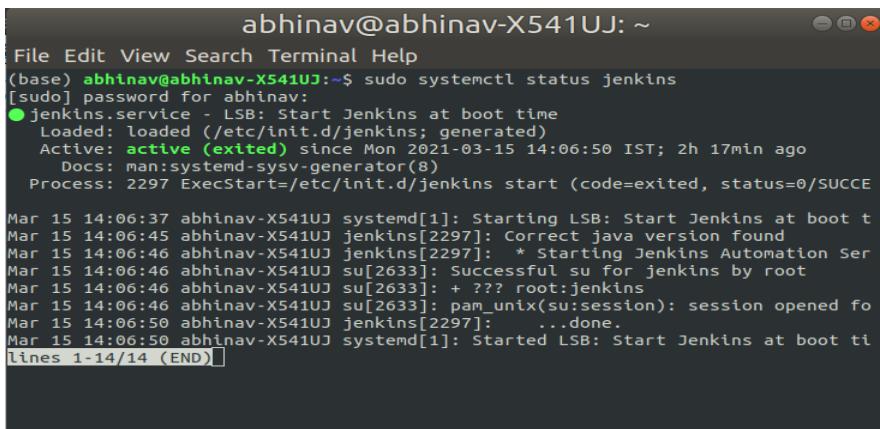
Jenkins runs on port 8080 by default, therefore to use Jenkins, open localhost:8080

Setting up Jenkins

Use the password provided by following command to unlock jenkins

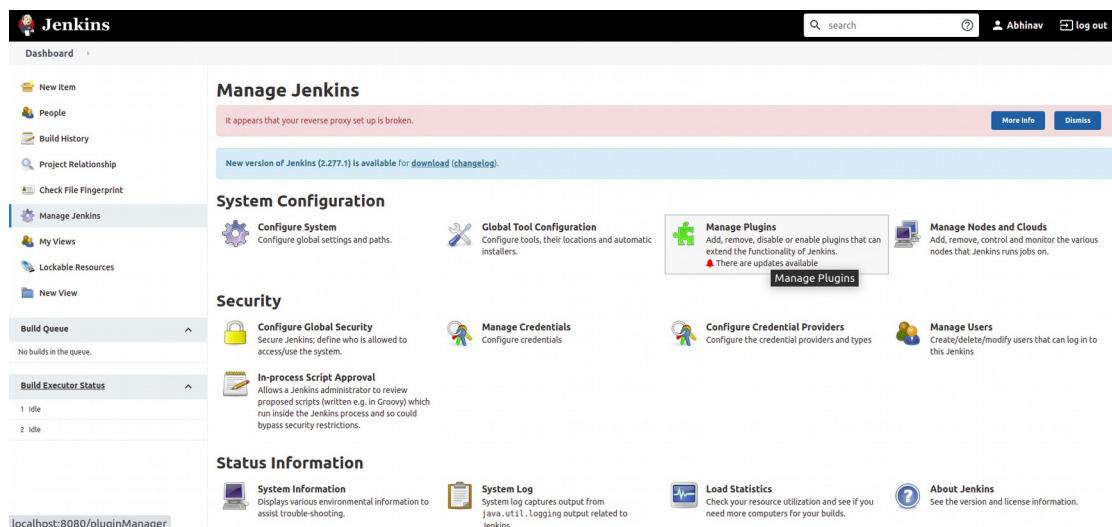
```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Choose install suggested plugins, configure username and password and it's done!



```
abhinav@abhinav-X541UJ: ~
File Edit View Search Terminal Help
(base) abhinav@abhinav-X541UJ:~$ sudo systemctl status jenkins
[sudo] password for abhinav:
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; generated)
  Active: active (exited) since Mon 2021-03-15 14:06:50 IST; 2h 17min ago
    Docs: man:systemd-sysv-generator(8)
   Process: 2297 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCE

Mar 15 14:06:37 abhinav-X541UJ systemd[1]: Starting LSB: Start Jenkins at boot t
Mar 15 14:06:45 abhinav-X541UJ jenkins[2297]: Correct java version found
Mar 15 14:06:46 abhinav-X541UJ jenkins[2297]: * Starting Jenkins Automation Ser
Mar 15 14:06:46 abhinav-X541UJ su[2633]: Successful su for jenkins by root
Mar 15 14:06:46 abhinav-X541UJ su[2633]: + ??? root:jenkins
Mar 15 14:06:46 abhinav-X541UJ su[2633]: pam_unix(su:session): session opened fo
Mar 15 14:06:50 abhinav-X541UJ jenkins[2297]: ...done.
Mar 15 14:06:50 abhinav-X541UJ systemd[1]: Started LSB: Start Jenkins at boot ti
lines 1-14/14 (END)
```



Jenkins

Dashboard

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

- My Views
- Lockable Resources
- New View

Build Queue

Build Executor Status

localhost:8080/pluginManager

Manage Jenkins

It appears that your reverse proxy set up is broken.

New version of Jenkins (2.277.1) is available for download (changelog).

More Info

Dismiss

System Configuration

Configure System

Global Tool Configuration

Manage Plugins

Manage Nodes and Clouds

Security

Configure Global Security

Manage Credentials

Configure Credential Providers

Manage Users

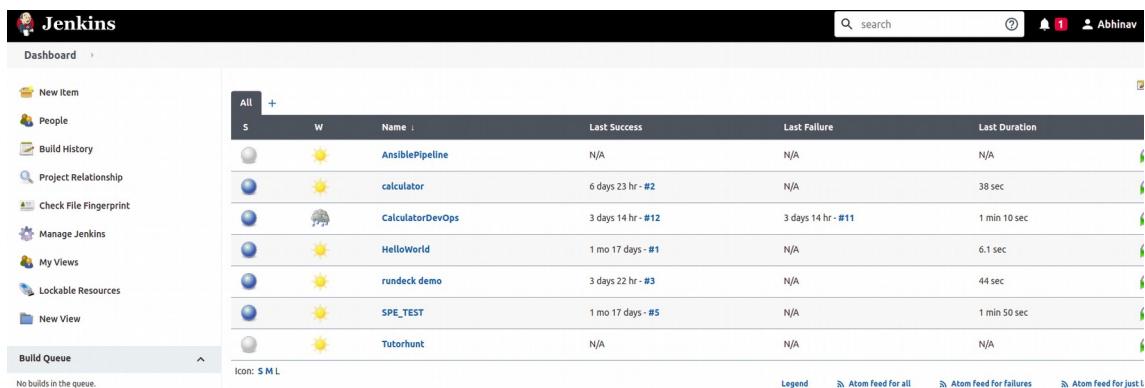
Status Information

System Information

System Log

Load Statistics

About Jenkins



Jenkins

Dashboard

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

Icon: S M L

S	W	Name :	Last Success	Last Failure	Last Duration
●	☀️	AnsiblePipeline	N/A	N/A	N/A
●	☀️	calculator	6 days 23 hr - #2	N/A	38 sec
●	⚡	CalculatorDevOps	3 days 14 hr - #12	3 days 14 hr - #11	1 min 10 sec
●	☀️	HelloWorld	1 mo 17 days - #1	N/A	6.1 sec
●	☀️	rundeck demo	3 days 22 hr - #3	N/A	44 sec
●	☀️	SPE_TEST	1 mo 17 days - #5	N/A	1 min 50 sec
●	☀️	Tutorhunt	N/A	N/A	N/A

Legend

Atom Feed for all

Atom Feed for failures

Atom Feed for just L

Creating a pipeline to Integrate SCM, Build Image and Deploy through Rundeck using Jenkins  
Configure Jenkins by installing plugins and make some configuration to run project in automated pipeline manner.

Install Plugins

If not-vulnerable

Manage Jenkins => Manage Plugins => Available => Filter => Install without restart

1. Git plugin
2. GitHub plugin
3. Maven Integration plugins
4. Docker plugins
5. Pipeline(A suite of plugins that lets you orchestrate automation, simple or complex)
6. Rundeck plugin

Manage Jenkins -> Configure System-> Add Rundeck

Instances:

Provide name

Provide URL of Rundeck server (we have, localhost:4440)

Put login id and password of rundeck

Test connection(successfull if rundeck running)

The screenshot shows the Jenkins configuration interface. On the left, there's a sidebar with various links like New Item, People, Build History, etc. The main area is titled 'Rundeck' and contains fields for 'Job cache' (checkbox for 'Enable Rundeck job cache') and 'Instances'. Under 'Instances', there are fields for 'Name' (set to 'Rundeck server'), 'URL' (set to 'http://localhost:4440'), 'Login' (set to 'admin'), 'Password' (set to 'Concealed'), 'Auth Token' (empty), and 'API Version' (empty). At the bottom, there are 'Save' and 'Apply' buttons.

## Global Tool Configuration for Jenkins

This involves providing path to various binaries to be used for Java, maven building, Git, etc.

Manage Jenkins => Global Tool Configuration.

JDK

**JAVA\_HOME: /usr/lib/jvm/java-11-openjdk-amd64**

Git

**Path to executable: /usr/bin/git**

Maven

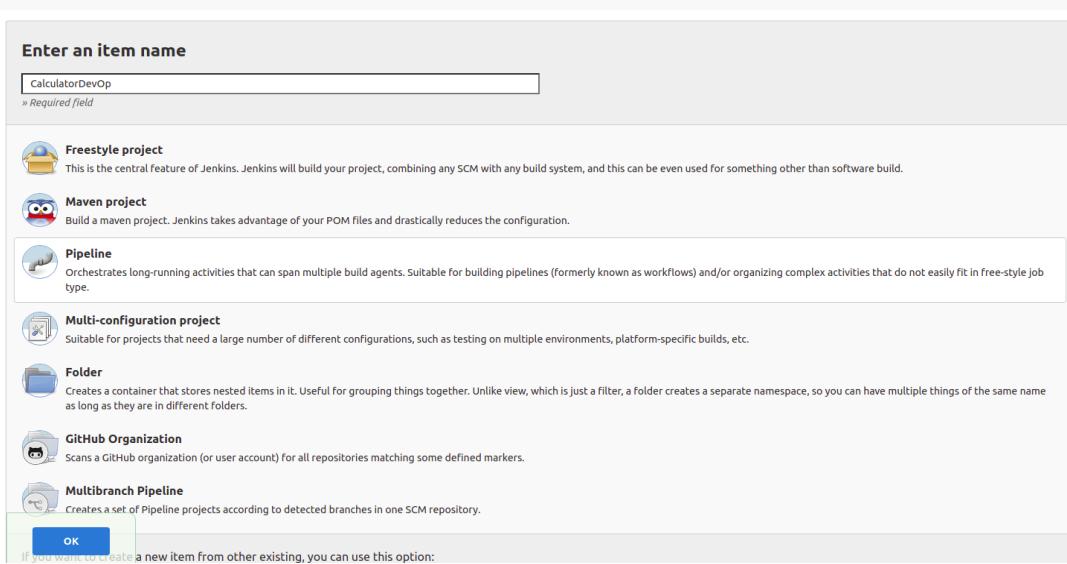
**I have opted to install automatically from apache ,maven 3.6.3**

Docker : **default**

The screenshot shows the 'Global Tool Configuration' page with sections for 'JDK', 'Git', 'Gradle', 'Ant', and 'Maven'.  
- \*\*JDK\*\*: Shows one entry named 'Java\_8' with 'JAVA\_HOME' set to '/usr/lib/jvm/java-11-openjdk-amd64'. There's a checkbox for 'Install automatically'.  
- \*\*Git\*\*: Shows one entry named 'Default' with 'Path to Git executable' set to '/usr/bin/git'. There's a checkbox for 'Install automatically'.  
- \*\*Gradle\*\*: Shows one entry named 'Gradle' with a note: 'List of Gradle installations on this system'.  
- \*\*Ant\*\*: Shows one entry named 'Ant' with a note: 'List of Ant installations on this system'.  
- \*\*Maven\*\*: Shows one entry named 'Maven' with a note: 'List of Maven installations on this system'.  
At the bottom, there are 'Save' and 'Apply' buttons.

## Creating Jenkins Pipeline :

The aim of the pipeline is to trigger whenever a commit happen on GitHub repository, the build should happen using dependencies defined in pom.xml and it should be tested automatically. If test is successful, then it builds an image from Dockerfile and pushes to DockerHub. If this is success, then Rundeck job is triggered from here, which will deploy the final containerized application to the host node(s).



Add the credentials of DockerHub to Jenkins so that it can push the image and later pull it from dockerhub repository in the jenkins pipeline.

Manage Credentials => System => global => Provide username and password => set ID and description as **dockerhub**

Create Jenkinsfile and add it to git.

Jenkins => New Item => Enter Item Name => Pipeline => OK => Pipeline Execution

Check Rundeck workflow Logs and Docker Image => Run the containerized program manually because it is CLI it can't be run on rundeck.

Add some new features and push it to GitHub. Pipeline will do the rest of the work for you.

## ONLINE MONITORING USING ELK Stack(Elasticsearch,Logstash,Kibana) :

"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.

Elasticsearch takes upto 8.2 GB(approx) of RAM while running ,so due to hardware constraint i took elastic.io cloud services which provides elasticsearch and kibana services under elastic stack over the cloud services provider you opted.

### Setting up elastic cloud :

To sign up, all you need is an email address:

- 1) Go to our Elasticsearch Service Trial page.
- 2) Enter your email address and password, or sign up with a Google account. Make sure you've read through our terms of service.

To get up and running with your deployment quickly:

- 1) From the Elasticsearch Service Console, click **Create deployment**.
- 2) Optimize your deployment for your use case by selecting a solution.

There are many solutions available ,I have opted Elastic stack one in which i have opted I/O optimized deployment which provides a balance of compute, memory, and storage

- Change your deployment settings by clicking Expand in the Deployment settings section.
  - a) Pick a cloud platform where they host your deployment. they support:

- Amazon Web Services (AWS)
  - Google Cloud Platform (GCP)
  - Microsoft Azure

I have opted AWS.

- Pick a region where your deployment gets hosted like:  
Asia Pacific (Tokyo),Asia Pacific (seoul),Asia Pacific (Mumbai) etc.,I have opted Asia Pacific (Mumbai)

4) Enable additional features, such as: Monitoring to ship your metrics to a dedicated deployment by enabling monitoring and selecting a pre-existing deployment.

5) Give your deployment a name if you prefer a name other than the one that has been pre-populated for you.

6) Click Create deployment to get started right away. Recommended if you are a trial user or new to the Elastic Stack.

The screenshot shows the Elasticsearch Service Console interface. On the left, a sidebar lists options: Cloud, Deployments, i-o-optimized-deployment (selected), Edit, Elasticsearch, Snapshots, API console, Kibana, APM, Enterprise Search, Logs and metrics, Activity, Security, Performance, Features, and Help. The main area displays the 'i-o-optimized-deployment' details. It includes fields for Deployment name (i-o-optimized-deployment), Deployment status (Healthy), Deployment ID (9781007), Deployment version (v7.11.2), Applications (Elasticsearch, Kibana, APM), and Cloud ID (aws, Mumbai (ap-south-1)). Below this, the 'Instances' section shows three instances across three zones: Zone ap-south-1a (Instance #0, healthy, v7.11.2 - 4 GB RAM, AWS DATA.H0H0i3 - data.hot, data.content - master, coordinating, ingest), Zone ap-south-1b (Tiebreaker #2, healthy, v7.11.2 - 1 GB RAM, AWS.MASTER.RSD - master eligible), and Zone ap-south-1c (Instance #0, healthy, v7.11.2 - 512 MB RAM - AWS.APM.RSD).

7) After your deployment is ready, write down the password for the elastic user and keep it somewhere safe. You need the password to connect to your Elasticsearch cluster, Kibana, and App Search.

We will use Logstash to transform and send the operational data to elasticsearch service using cloudID. Now we will install Logstash in our localhost by following below steps :

- Download and unzip Logstash.
  - Prepare a logstash.conf config file and place it in same folder as bin.
  - Run bin/logstash -f logstash.conf in working directory.

```

(base) abhinav@abhinav-X541U:~/logstash-7.11.2$ bin/logstash -f logstash-simple.conf
Using JAVA_HOME defined java: /usr/lib/jvm/java-11-openjdk-amd64/
WARNING, using JAVA_HOME while Logstash distribution comes with a bundled JDK
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepC was deprecated in version 9.0 and will likely be removed in a future release
Sending Logstash logs to /home/abhinav/logstash-7.11.2/logs which is now configured via log4j2.properties
[2021-03-15T18:55:55,859][INFO ][logstash.runner] Starting Logstash ["logstash.version"=>"7.11.2", "jruby.version"=>"jruby-9.2.13.0 (2021-03-15T18:55:55,859+0000) [jemalloc jit] [linux-x86_64]"]
[2021-03-15T18:55:56,553][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line parameters have been passed
[2021-03-15T18:55:59,121][INFO ][org.reflections.Reflections] Reflections took 79 ms to scan 1 urls, producing 23 keys and 47 values
[2021-03-15T18:56:00,999][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updated {:changes=>[], :added=>[], :removed=>[], :disabled=>[]}
[2021-03-15T18:56:04,545][WARN ][logstash.outputs.elasticsearch][main] Restored connection to ES instance {:url=>"https://elasticsearch-cloud-01.us-west-2.amazonaws.com:9243"}
[2021-03-15T18:56:04,897][INFO ][logstash.outputs.elasticsearch][main] ES Output version determined {:es_version=>7}
[2021-03-15T18:56:04,908][WARN ][logstash.outputs.elasticsearch][main] Detected a 6.x and above cluster: the 'type' event field won't be used to determine the document type and the 'geo_point' field won't be indexed by default
[2021-03-15T18:56:04,973][INFO ][logstash.outputs.elasticsearch][main] New Elasticsearch output {:class=>"Logstash::Outputs::ElasticSearch", :hosts=>"https://elasticsearch-cloud-01.us-west-2.amazonaws.com:9243"}
[2021-03-15T18:56:05,009][INFO ][logstash.outputs.elasticsearch][main] Using a default mapping template {:es_version=>7, :ecs_compatibility=>1.4, :index_pattern=>"logstash-%{+YYYY.MM.dd}"}
[2021-03-15T18:56:05,470][INFO ][logstash.javapipeline] [main] Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>500, "pipeline.sources"=>["/home/abhinav/logstash-7.11.2/logstash-simple.conf"], :thread=>"#Thread-0x7fffb38 run"}
[2021-03-15T18:56:05,480][INFO ][logstash.outputs.elasticsearch][main] Attempting to install template {:manage_template=>{"index_pattern"=>"logstash-*", "type"=>"_index", "template"=>{"@version"=>"1", "@source"=>"/path/to/messages.log", "properties"=>{"@version"=>"1", "@source"=>"path/to/messages.log", "geo_point"=>{"lat"=>40, "lon"=>-120}, "match_mapping_type"=>"string", "mapping"=>{"@type"=>"text", "norms"=>false, "fields"=>{"@version"=>{"@type"=>"geo_point"}, "location"=>{"@type"=>"geo_point"}, "lat"=>40, "lon"=>-120}}}
[2021-03-15T18:56:06,905][INFO ][logstash.javapipeline] [main] Pipeline Java execution initialization time {"seconds"=>1.43}
[2021-03-15T18:56:07,243][INFO ][logstash.inputs.file] [main] No $index_path set, generating one based on the 'path' setting (=es0f9173fce3e0ff522ec79ee1dd77938)
[2021-03-15T18:56:07,266][INFO ][logstash.javapipeline] [main] Pipeline started {"pipeline.id"=>"main"}
[2021-03-15T18:56:07,339][INFO ][logstash.agent] [main] Pipelines configured {:count=>1, :running_pipelines=>[{:main, :non_running_pipelines=>[]}]}
[2021-03-15T18:56:07,347][INFO ][filewatch.observingtail] [main]@a3fcf91378f05ef931ed2a55800056cab13aebc58bd342b62fccc39200cce245b]
[2021-03-15T18:56:07,682][INFO ][logstash.agent] [main] Successfully started Logstash API endpoint {:port=>9600}
{
    "path" => "/home/abhinav/calculatorspe/calculator.log",
    "logger" => "power -",
    "action" => "x raised to y",
    "level" => "INFO",
    "@timestamp" => 2021-03-14T19:30:38.472Z,
    "message" => "15/Mar/2021:01:00:38 472 [Calculator.java] [INFO] power - [x raised to y] - 3.0 ^ 7.0",
    "@version" => "1",
    "thread" => "Calculator.java",
    "host" => "abhinav-X541U",
    "line" => "3.0 ^ 7.0"
}

    "path" => "/home/abhinav/calculatorspe/calculator.log",
    "logger" => "power -",
    "action" => "RESULT - x raised to y",
    "level" => "INFO",
    "@timestamp" => 2021-03-14T19:30:38.473Z,
    "message" => "15/Mar/2021:01:00:38 473 [Calculator.java] [INFO] power - [RESULT - x raised to y] - 2187.0",
    "@version" => "1",
    "thread" => "Calculator.java",
    "host" => "abhinav-X541U",
    "line" => "2187.0"
}

```

Following is the structure of logstash.conf file to configure Logstash with cloud and send data to elasticsearch service running on our cloud.

```
input {
file {
path => "<input .log file>"
start_position => "beginning"
}
}

filter {
grok {
match => [
"message", "%{HTTPDATE:timestamp_string} \[%{GREEDYDATA:thread}\] \[%{LOGLEVEL:level}\] %\n{GREEDYDATA:logger} \[%{GREEDYDATA:action}\] \%- %{GREEDYDATA:line}""
]
}
mutate {
remove_field => [timestamp_string]
}
}

output {
elasticsearch {
cloud_id => "<cloud id displayed in Deployments home page>"
cloud_auth => "username:password"
}
```

```

index => "<index name>"  

}  

stdout {  

codec => <the name of Logstash codec used to represent the data>  

}  

}

```

**Input Plugin** - An input plugin enables a specific source of events to be read by Logstash .

**Filter Plugin**- A filter plugin performs intermediary processing on an event. Filters are often applied conditionally depending on the characteristics of the event

**Output Plugin**- An output plugin sends event data to a particular destination. Outputs are the final stage in the event pipeline.

Now run bin/logstash -f logstash.conf in working directory.This command will start Logstash and it will collect data from input log file structure it according to grok pattern provided and send the operational data to elasticsearch service running on our cloud and it will register the index by the name given by us to recognise the data.

## KIBANA :

Now after successful sending of operational data by Logstash we should launch KIBANA hosted on elastic cloud.

Sidenav => Management => Stack management => Index Management ,we should see our index given by us in logstash.conf file.

We can also create our index pattern to retreive data easily in kabana based on our index name ,we can see various fields associated with that index pattern.

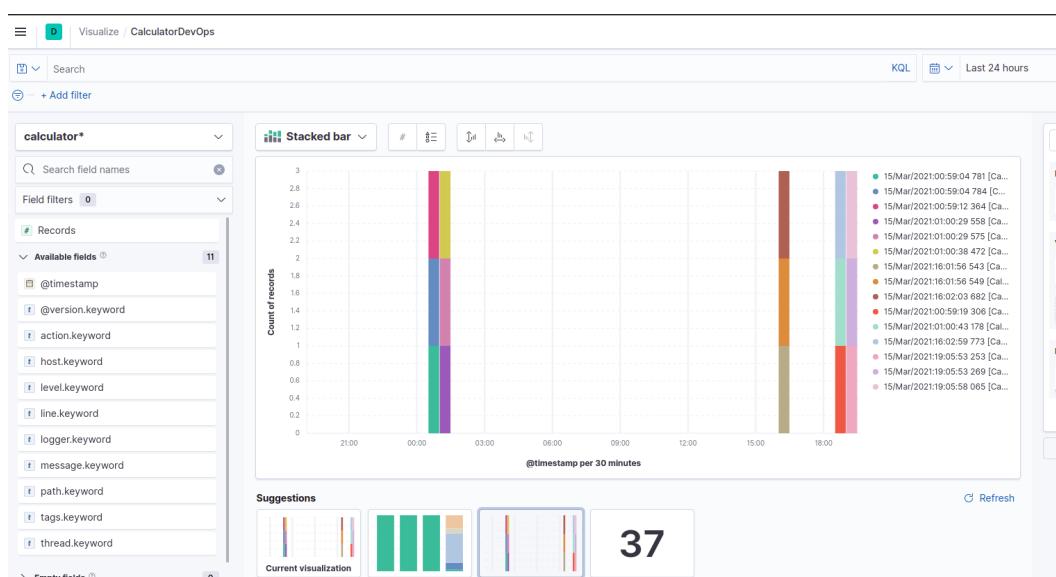
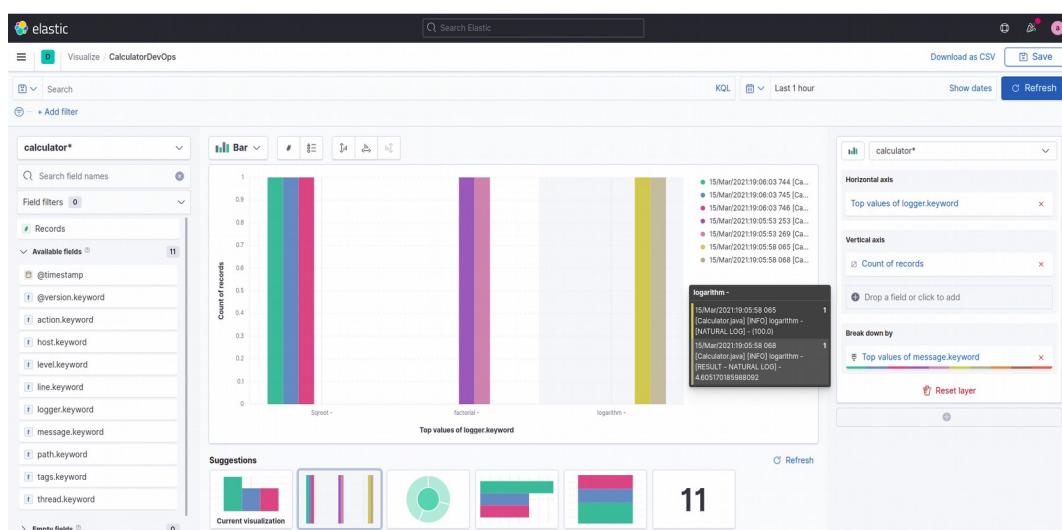
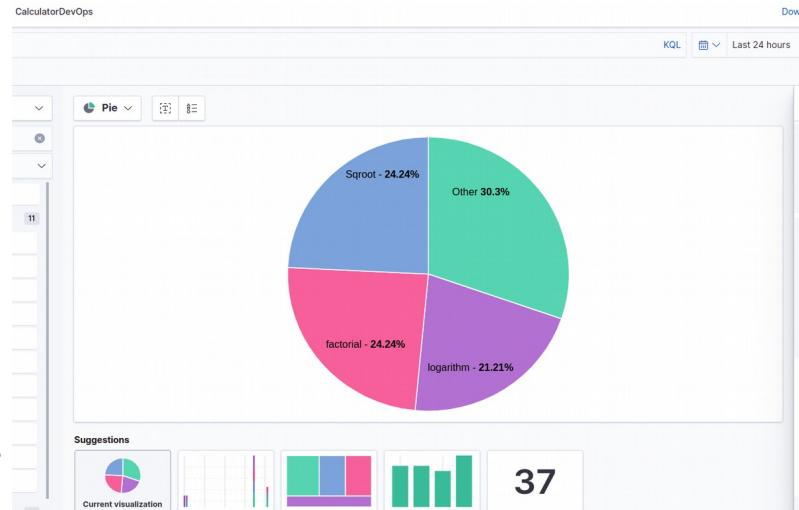
Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
apm-7.11.2-profile-000001	green	open	1	1	0	416b	
apm-7.11.2-span-000001	green	open	1	1	0	416b	
apm-7.11.2-metric-000001	green	open	1	1	0	416b	
apm-7.11.2-onboarding-2021.03.13	green	open	1	1	1	14.9kb	
calculator_elastic	green	open	1	1	57	179.8kb	
apm-7.11.2-transaction-000001	green	open	1	1	0	416b	
apm-7.11.2-error-000001	green	open	1	1	0	416b	

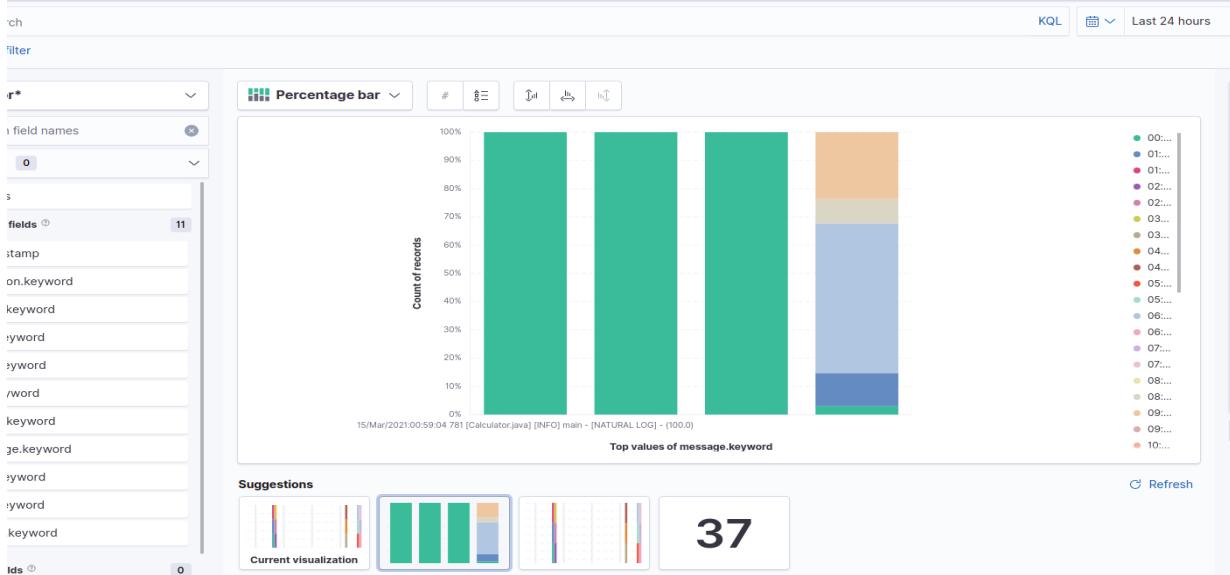
Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		✓	✓	✗
@version	string		✓	✓	✗
@version.keyword	string		✓	✓	✗
_id	string		✓	✓	✗
_index	string		✓	✓	✗
_score	number				
_source	_source				
_type	string		✓	✓	✗
action	string		✓	✓	✗
action.keyword	string		✓	✓	✗
host	string		✓	✓	✗
host.keyword	string		✓	✓	✗
level	string		✓	✓	✗
level.keyword	string		✓	✓	✗
line	string		✓	✓	✗
line.keyword	string		✓	✓	✗
logger	string		✓	✓	✗
logger.keyword	string		✓	✓	✗
message	string		✓	✓	✗
message.keyword	string		✓	✓	✗
path	string		✓	✓	✗
path.keyword	string		✓	✓	✗

## Visualization in Kibana :

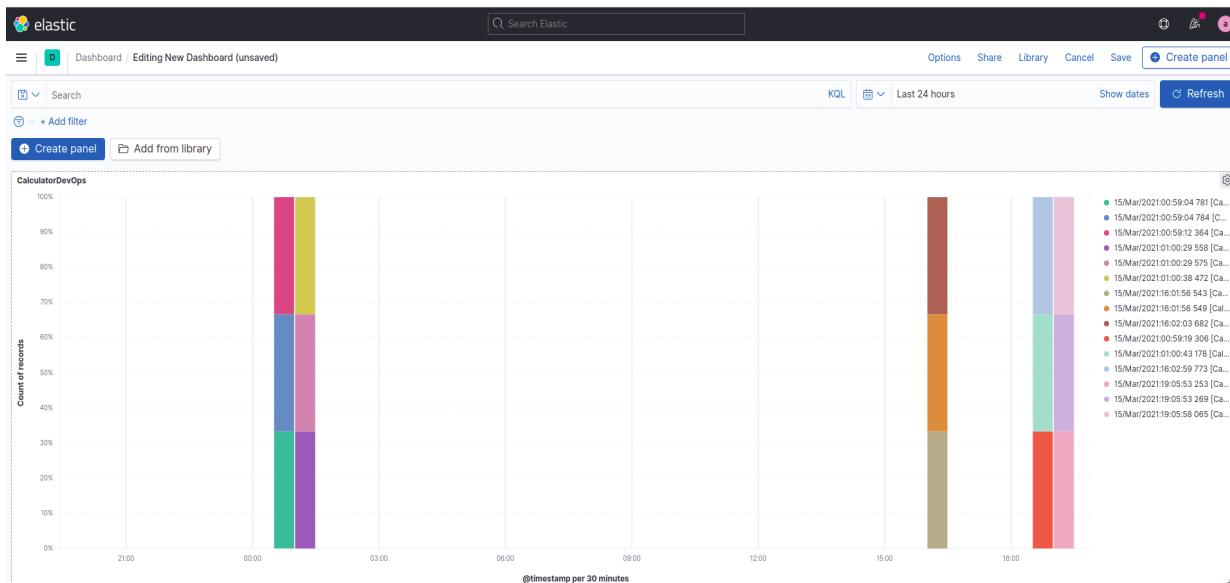
SideNav => Visualize => create visualization => Lens(we have different options to visualize) => select an index pattern to visualize based on it's fields.

Now run your program on host where Logstash is installed and the stack pipeline will perform rest of the work from generating log data to fetch it ,structure it, connect with elastic cloud,send it to elasticsearch and ultimately visualization through Kibana, all done after clicking run program button.





## Dashboard created from Panel :



## Logs based on which visualization is done :

```
m.xml(calculatorms) ✘ Calculator.java ✘ calculatortesting.java ✘ Dockerfile ✘ log4j2.xml ✘ calculator.log ✘ Jenkinsfile ✘
15/Mar/2021:00:59:04 784 [Calculator.java] [INFO] main - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:00:59:12 364 [Calculator.java] [INFO] main - [FACTORIAL] - 15!
15/Mar/2021:00:59:12 365 [Calculator.java] [INFO] main - [RESULT - FACTORIAL] - 2.004310016E9
15/Mar/2021:00:59:19 306 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:00:29 558 [Calculator.java] [INFO] factorial - [FACTORIAL] - 7!
15/Mar/2021:01:00:29 575 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 5040.0
15/Mar/2021:01:00:38 472 [Calculator.java] [INFO] power - [x raised to y] - 3.0 ^ 7.0
15/Mar/2021:01:00:38 473 [Calculator.java] [INFO] power - [RESULT - x raised to y] - 2187.0
15/Mar/2021:01:00:43 178 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:00:43 179 [Calculator.java] [INFO] factorial - [FACTORIAL] - 4!
15/Mar/2021:01:00:54 543 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 24.0
15/Mar/2021:01:00:54 549 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 24.0
15/Mar/2021:01:00:54 558 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (100.0)
15/Mar/2021:01:00:54 568 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:01:00:54 575 [Calculator.java] [INFO] power - [x raised to y] - 6.0 ^ 3.0
15/Mar/2021:01:00:58 525 [Calculator.java] [INFO] power - [x raised to y] - 216.0
15/Mar/2021:01:00:58 526 [Calculator.java] [INFO] power - [RESULT - x raised to y] - 216.0
15/Mar/2021:01:02:18 201 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt(100000.0)
15/Mar/2021:01:02:18 203 [Calculator.java] [INFO] Sqroot - [RESULT - SQUARE ROOT] - 316.22776601683796
15/Mar/2021:01:02:28 121 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (-89.0)
15/Mar/2021:01:02:28 121 [Calculator.java] [ERROR] logarithm - [EXCEPTION - NATURAL LOG] - Cannot find LOG of negative number or 0 Domain Error
15/Mar/2021:01:02:28 122 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - NaN
15/Mar/2021:01:02:34 077 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt(-9.0)
15/Mar/2021:01:02:34 078 [Calculator.java] [ERROR] Sqroot - [EXCEPTION - SQUARE ROOT] - Cannot find square root of negative number Case of NaN negative square root
15/Mar/2021:01:02:34 078 [Calculator.java] [INFO] Sqroot - [RESULT - SQUARE ROOT] - NaN
15/Mar/2021:01:02:48 621 [Calculator.java] [INFO] power - [x raised to y] - -2.0 ^ 2.9
15/Mar/2021:01:02:48 623 [Calculator.java] [INFO] power - [RESULT - x raised to y] - NaN
15/Mar/2021:01:02:54 302 [Calculator.java] [INFO] factorial - [FACTORIAL] - 7!
15/Mar/2021:01:02:54 303 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 5040.0
15/Mar/2021:01:02:59 773 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:05:53 253 [Calculator.java] [INFO] factorial - [FACTORIAL] - 18!
15/Mar/2021:01:05:53 269 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 3628800.0
15/Mar/2021:01:05:58 065 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (100.0)
15/Mar/2021:01:05:58 068 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:01:06:03 744 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt(-34.8)
15/Mar/2021:01:06:03 745 [Calculator.java] [ERROR] Sqroot - [EXCEPTION - SQUARE ROOT] - Cannot find square root of negative number Case of NaN negative square root
```

## SOME SCREENSHOTS FROM PROJECT COMPONENTS :

### Codes:

The screenshot shows two Java code files in a code editor. On the left is `Calculator.java`, which contains a `Calculator` class with methods for factorial, natural log, and square root. It uses `Scanner` for input and `Logger` for logging. On the right is `calculatortesting.java`, which contains a `CalculatorTesting` class with a `main` method that runs the calculator and logs results.

```
calculator - Calculator.java
dit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
@atospe src main java calculator Calculator Sqroot
xml(calculator) / Calculator.java calculatortesting.java Dockerfile log4j2.xml calculator.log
package calculator;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Calculator {
    private static final Logger logger = LogManager.getLogger(Calculator.class.getName());

    public Calculator() {
    }

    public static void main(String[] args) {
        Calculator calculator = new Calculator();
        Scanner scanner = new Scanner(System.in);
        int number;
        double number, exponent;
        do {
            System.out.println("\n*****Calculator*****\n");
            System.out.println("Choose to perform operation\n");
            System.out.print("Press 1 to find factorial of a number\nPress 2 to find natural log of a number\n" +
                "Press 3 to find value of number x raised to y\nPress 4 to find square root of a number\n" +
                "Press any other key to exit\nEnter your choice: ");
            int choice;
            try {
                choice = scanner.nextInt();
            } catch (InputMismatchException error) {
                logger.error("Invalid input, Entered input is not a number");
                return;
            }
            if (choice == 1) {
                System.out.println("Factorial of the number is : " + calculator.factorial(number));
            } else if (choice == 2) {
                System.out.println("Natural log of the number is : " + calculator.logarithm(number));
            } else if (choice == 3) {
                System.out.print("Enter the base : ");
                number = scanner.nextDouble();
                System.out.print("Enter the exponent : ");
                exponent = scanner.nextDouble();
                System.out.println("Value of " + number + " raised to " + exponent + " is : " + calculator.power(number, exponent));
            } else if (choice == 4) {
                System.out.println("Square root of the number is : " + calculator.sqrt(number));
            }
        } while (true);
    }

    public Double factorial(int number) {
        double result = 1;
        int fact = 1;
        try {
            logger.info("[FACTORIAL] - " + number + "!");
            if (number < 0) {
                result = Double.NaN;
                throw new ArithmeticException("Domain Error");
            } else {
                for (int i = 1; i <= number; i++) {
                    fact *= i;
                }
                result = fact;
            }
        } catch (ArithmeticException error) {
            logger.error("[EXCEPTION - FACTORIAL] - Cannot find FACTORIAL of negative number " +
                error.getLocalizedMessage());
        } finally {
            logger.info("[RESULT - FACTORIAL] - " + result);
        }
        return result;
    }

    public double logarithm(double number) {
        Double result = 0.0;
        try {
            logger.info("[NATURAL LOG] - " + "(" + number + ")");
            if (number <= 0) {
                result = Double.NaN;
                throw new ArithmeticException("Domain Error");
            }
        } catch (ArithmeticException error) {
            logger.error("[EXCEPTION - NATURAL LOG] - Cannot find NATURAL LOG of negative number " +
                error.getLocalizedMessage());
        } finally {
            logger.info("[RESULT - NATURAL LOG] - " + result);
        }
        return result;
    }

    public double sqrt(double number) {
        Double result = 0.0;
        try {
            logger.info("[SQUARE ROOT] - " + "(" + number + ")");
            if (number < 0) {
                result = Double.NaN;
                throw new ArithmeticException("Domain Error");
            }
        } catch (ArithmeticException error) {
            logger.error("[EXCEPTION - SQUARE ROOT] - Cannot find SQUARE ROOT of negative number " +
                error.getLocalizedMessage());
        } finally {
            logger.info("[RESULT - SQUARE ROOT] - " + result);
        }
        return result;
    }
}

calculatortesting - calculatortesting.java
import java.util.Scanner;
import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.LogManager;

public class CalculatorTesting {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number : ");
        int number = scanner.nextInt();
        System.out.println("Factorial of the number is : " + Calculator.factorial(number));
        System.out.println("Natural log of the number is : " + Calculator.logarithm(number));
        System.out.print("Enter the base : ");
        double base = scanner.nextDouble();
        System.out.print("Enter the exponent : ");
        double exponent = scanner.nextDouble();
        System.out.println("Value of " + number + " raised to " + exponent + " is : " + Calculator.power(number, exponent));
        System.out.println("Square root of the number is : " + Calculator.sqrt(number));
    }
}
```

### DockerFile:

The screenshot shows a `Dockerfile` in a terminal window. It defines a base image as `openjdk:8-alpine`, sets the working directory to `.`, copies the build artifacts to the container, and specifies the command to run the application as `java -jar calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar`.

```
pom.xml (calculatorspe) × Calculator.java × calculatortesting.java × Dockerfile ×
1 # Set the base image
2 >> FROM openjdk:8-alpine
3 # File Author
4 MAINTAINER Abhinav
5 # Copies the files from the source on the host into the container's set destination
6 COPY ./target/calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar .
7 # Set Working Directory
8 WORKDIR .
9 # Default container command to execute the program
10 CMD ["java", "-jar", "calculatorspe-1.0-SNAPSHOT-jar-with-dependencies.jar"]
```

## Log4j properties file :

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
    <Appenders>
        <Console name="ConsoleAppender" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{dd/MMM/yyyy:HH:mm:ss SSS} [%F] [%level] %M - %msg%n"/>
        </Console>
        <File name="FileAppender" fileName="/home/abhinav/calculatorspe/calculator.log" immediateFlush="false" append="true">
            <PatternLayout pattern="%d{dd/MMM/yyyy:HH:mm:ss SSS} [%F] [%level] %M - %msg%n"/>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="ConsoleAppender"/>
            <AppenderRef ref="FileAppender"/>
        </Root>
    </Loggers>
</Configuration>
```

## Log file:

```
15/Mar/2021:00:59:04 784 [Calculator.java] [INFO] main - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:00:59:12 344 [Calculator.java] [INFO] main - [FACTORIAL] - 15!
15/Mar/2021:00:59:12 345 [Calculator.java] [INFO] main - [RESULT - FACTORIAL] - 2.004310016E9
15/Mar/2021:00:59:19 304 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:00:29 558 [Calculator.java] [INFO] factorial - [FACTORIAL] - ?
15/Mar/2021:01:00:29 575 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 5040.0
15/Mar/2021:01:00:38 472 [Calculator.java] [INFO] power - [x raised to y] - 3.0 ^ 7.0
15/Mar/2021:01:00:38 473 [Calculator.java] [INFO] power - [RESULT - x raised to y] - 2187.0
15/Mar/2021:01:00:43 178 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:01:56 543 [Calculator.java] [INFO] factorial - [FACTORIAL] - 4!
15/Mar/2021:01:01:56 545 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 24.0
15/Mar/2021:01:02:08 682 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (100.0)
15/Mar/2021:01:02:08 683 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:01:02:08 525 [Calculator.java] [INFO] power - [x raised to y] - 6.0 ^ 3.0
15/Mar/2021:01:02:08 526 [Calculator.java] [INFO] power - [RESULT - x raised to y] - 216.0
15/Mar/2021:01:02:18 261 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt( 100000.0 )
15/Mar/2021:01:02:18 263 [Calculator.java] [INFO] Sqroot - [RESULT - SQUARE ROOT] - 316.22776601683796
15/Mar/2021:01:02:28 121 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (-89.0)
15/Mar/2021:01:02:28 221 [Calculator.java] [ERROR] logarithm - [EXCEPTION - NATURAL LOG] - Cannot find LOG of negative number or 0 Domain Error
15/Mar/2021:01:02:28 222 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - NaN
15/Mar/2021:01:02:34 077 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt( -9.0 )
15/Mar/2021:01:02:34 078 [Calculator.java] [ERROR] Sqroot - [EXCEPTION - SQUARE ROOT] - Cannot find square root of negative number Case of NaN negative square root
15/Mar/2021:01:02:34 078 [Calculator.java] [INFO] Sqroot - [RESULT - SQUARE ROOT] - NaN
15/Mar/2021:01:02:48 621 [Calculator.java] [INFO] power - [x raised to y] - -2.0 ^ 2.9
15/Mar/2021:01:02:48 623 [Calculator.java] [INFO] power - [RESULT - x raised to y] - NaN
15/Mar/2021:01:02:54 302 [Calculator.java] [INFO] factorial - [FACTORIAL] - ?
15/Mar/2021:01:02:54 303 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 5040.0
15/Mar/2021:01:02:54 773 [Calculator.java] [ERROR] main - Invalid input, Entered input is not a number
15/Mar/2021:01:05:53 253 [Calculator.java] [INFO] factorial - [FACTORIAL] - 10!
15/Mar/2021:01:05:53 269 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 3628800.0
15/Mar/2021:01:05:58 065 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (100.0)
15/Mar/2021:01:05:58 068 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - 4.605170185988092
15/Mar/2021:01:06:03 744 [Calculator.java] [INFO] Sqroot - [SQUARE ROOT] - sqrt( -34.0 )
15/Mar/2021:01:06:03 745 [Calculator.java] [ERROR] Sqroot - [EXCEPTION - SQUARE ROOT] - Cannot find square root of negative number Case of NaN negative square root
```

## Jenkins file :

```
pipeline {
    environment {
        registry = "abhinav6636/calculator_container"
        registryCredential = 'dockerhub'
        dockerImage = ''
    }
    agent any
    stages {
        stage('Cloning Git') {
            steps {
                git 'https://github.com/abhi93wari/CalculatorSpe.git'
            }
        }
        stage('Building JAR file'){
            steps{
                sh 'mvn install'
            }
        }
        stage('MVN TEST') {
            steps {
                echo "Running the test cases of the project"
                sh "mvn test"
            }
        }
        stage('Building image') {
            steps{
                script {
                    dockerImage = docker.build registry + ":$BUILD_NUMBER"
                }
            }
        }
        stage('Push Image to Dockerhub') {
            steps{
                script {
                    docker.withRegistry( '', registryCredential ) {
                        docker.push("calculator")
                    }
                }
            }
        }
    }
}
```

## Pom file :

```
m pom.xml (calculatorspe) ✘ Calculator.java ✘ calculatortesting.java ✘ Dockerfile ✘ log4j2.xml ✘ calculatortesting.log ✘
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.calculator</groupId>
8   <artifactId>calculatorspe</artifactId>
9   <version>1.0-SNAPSHOT</version>
10  <name>calculatorspe</name>
11
12  <properties>
13    <maven.compiler.target>1.8</maven.compiler.target>
14    <maven.compiler.source>1.8</maven.compiler.source>
15    <junit.version>5.6.2</junit.version>
16  </properties>
17
18  <dependencies>
19
20    <dependency>
21      <groupId>org.junit.jupiter</groupId>
22      <artifactId>junit-jupiter-api</artifactId>
23      <version>${junit.version}</version>
24      <scope>test</scope>
25    </dependency>
26    <dependency>
27      <groupId>org.junit.jupiter</groupId>
28      <artifactId>junit-jupiter-engine</artifactId>
29      <version>${junit.version}</version>
30      <scope>test</scope>
31    </dependency>
32    <dependency>
```

## Console Output :

```
*****Calculator*****
Choose to perform operation

Press 1 to find factorial of a number
Press 2 to find natural log of a number
Press 3 to find value of number x raised to y
Press 4 to find square root of a number
Press any other key to exit
Enter your choice: 1
Enter the number : 10
15/Mar/2021:19:05:53 253 [Calculator.java] [INFO] factorial - [FACTORIAL] - 10!
15/Mar/2021:19:05:53 269 [Calculator.java] [INFO] factorial - [RESULT - FACTORIAL] - 3628800.0
Factorial of the number is : 3628800.0

*****Calculator*****
Choose to perform operation

Press 1 to find factorial of a number
Press 2 to find natural log of a number
Press 3 to find value of number x raised to y
Press 4 to find square root of a number
Press any other key to exit
Enter your choice: 2
Enter the number : 100
15/Mar/2021:19:05:58 065 [Calculator.java] [INFO] logarithm - [NATURAL LOG] - (100.0)
15/Mar/2021:19:05:58 068 [Calculator.java] [INFO] logarithm - [RESULT - NATURAL LOG] - 4.605170185988092
Natural log of the number is : 4.605170185988092
```

## **LINKS :**

### **GitHub**

<https://github.com/abhi93wari/CalculatorSpe>

### **DockerHub**

[https://hub.docker.com/r/abhinav6636/calculator\\_container](https://hub.docker.com/r/abhinav6636/calculator_container)

## **REFERENCES :**

### Java and SCM

<https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-18-04>  
<https://github.com/anshgyl/devops-tutorial/>  
<https://www.multisoftvirtualacademy.com/blog/common-advantages-and-disadvantages-of-multithreading-in-java>

### Junit

<https://www.vogella.com/tutorials/JUnit/article.html>  
<https://findanyanswer.com/what-is-setup-and-teardown-in-junit/>

### Maven commands

<http://people.apache.org/~aramirez/maven-install-plugin/dependencies.html>  
<https://www.journaldev.com/33645/maven-commands-options-cheat-sheet>

### Installing and working with Docker

<http://web.archive.org/web/20210226180942/https://docs.docker.com/engine/>  
<https://stackoverflow.com/questions/60527336/what-is-the-difference-between-docker-daemon-and-docker-engine/>  
[http://web.archive.org/web/20210226185201/https://www.tutorialspoint.com/docker/docker\\_hub.htm](http://web.archive.org/web/20210226185201/https://www.tutorialspoint.com/docker/docker_hub.htm)  
<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>

### Rundeck

<http://web.archive.org/web/20210226183617/https://www.talentica.com/blogs/automate-deployment-using-rundeck>  
<http://web.archive.org/web/20210226183617/https://foxutech.com/what-is-rundeck/>  
<https://github.com/iamdhruvp>

### Create Jenkin pipeline

<https://www.edureka.co/blog/jenkins-pipeline-tutorial-continuous-delivery/>  
<https://www.edureka.co/community/55640/jenkins-docker-docker-image-jenkins-pipeline-docker-registry/>

### ELK

<https://harshityadav95.medium.com/getting-started-with-logstash-96f5f1000cb6>  
<https://qbox.io/blog/logstash-grok-filter-tutorial-patterns>  
[https://github.com/shubhamagarwal890/ELK-Example.git/](https://github.com/shubhamagarwal890/ELK-Example.git)  
<https://grokconstructor.appspot.com/do/match>  
<https://www.elastic.co/what-is/elk-stack>