Scientific Calculator

with

DevOps

Version 1.0

Prepared by IMT2018019

Chebrolu Suchith Kumar

Suchith.Kumar@iiitb.org

IIIT Bangalore

13-03-2021.

Table of Contents

Table of Contents....ii
Revision History....ii
1. Introduction....1
1.1 Problem Statement....1
1.2 DevOps tool chain....1
1.3 OpenJDK....1
1.4 Git, Github....2
2. Testing, Buidling....3
2.1 Junit....3
2.2 Maven....4
3. Continious Integration....5
3.1 What is continious integration....5
3.2 Jenkins....5
4. Containerizatoin....6
4.1 Docker....6
4.2 Create docker image....7
4.3 Docker file....8
4.4 Dockerhub....9
5. Deployment 9
5.1 Ansible....9
5.2 Implementing Playbooks....10
5.3 Running Playbooks....11
6. Monitoring Using ELK....11
6.1 Logstash, configutation file....12
6.2 Elasticsearch, Kibana visualization....13
7. Pipeline to Integrate SCM, Build Image and Deploy through Ansible using Jenkins….14
Links, References....17

Revision History

Name	Date	Reason For Changes	Version
SciCal	13-03-2021	First version	Version 1.0

1. Introduction

1.1 Problem Statement

Create a scientific calculator program with user menu driven operations

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

1.2 DevOps tool chain

You can use any set of DevOps tool chains you want, but the pipeline would be the same. The pipeline includes

→ Using a source control management tool - like GitHub, GitLab, BitBucket etc
→ Testing - test your code using either JUnit, Selenium, PyUnit and many more
→ Build - build your code using tool like Maven, Gradle, Ant and many more
→ Continuous Integration - Continuous integrate your code using tool like Jenkins, GitLab CLI, Travis CLI, and many more
→ Containerize - Containerize your code using Docker.
→ Push your created Docker image to Docker hub .
→ Deployment - Do configuration management and deployment using either Chef, Puppet, Ansible, Rundeck. Using these do configuration management and pull your docker image and run it on the managed hosts.
→ For Deployment you can either do it on your local machine or on Kubernetes cluster or OpenStack cloud. You can also use Amazon AWS or Google Cloud or some other 3rd party cloud.
→ Monitoring - for monitoring use the ELK stack. Use a log file to do the monitoring. Generate the log file for your mini project and pass in your ELK stack.

1.3 OpenJDK (Java)

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible.

Installation:
$apt-update
$apt install openjdk-8-jdk

Configure $JAVA_HOME path:

$sudo gedit /etc/environment

Add the following line at the end of the file
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"

Restart the system to apply the changes.
Verify the above using
$echo $JAVA_HOME

1.4 Git, Github

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:

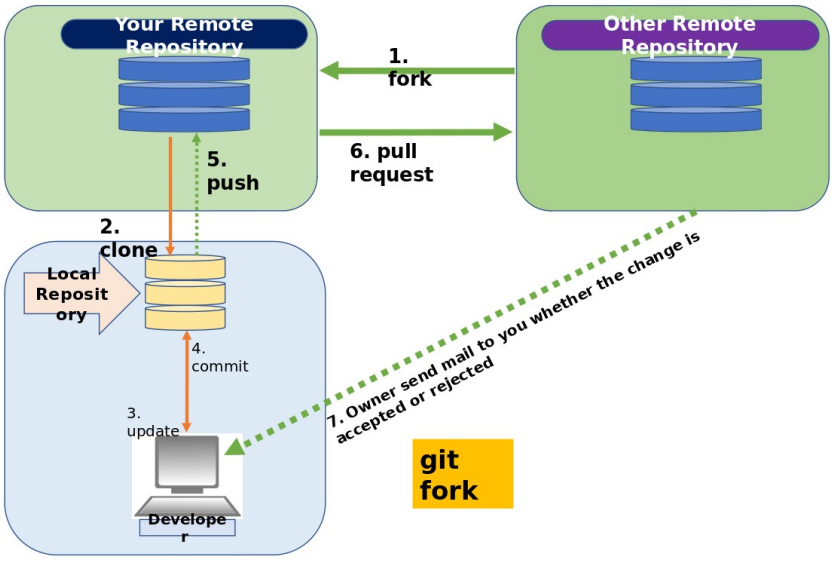
$apt update
$apt install git

Configuration:

$git config --global user.name "Your Name"
$git config --global user.email "YourEmailID@domain.com"
$ git –version

GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version controll and source code management (SCM) functionality of Git, plus its own features.
Git add, commit, push, clone, update, fork commands and explanation in a diagram-

```
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ git init
Reinitialized existing Git repository in /home/suchit/Downloads/SciCal-master/.git/
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ git add *
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ git commit -m "latest"
[master 2497763] latest
 6 files changed, 223 insertions(+), 27 deletions(-)
 create mode 100644 calculator.log
 create mode 100644 calculator_logstash.conf
 create mode 100644 src/main/resources/log4j2.xml
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ git remote add origin https://github.com/suchithkumarch/SciCal.git
fatal: remote origin already exists.
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ git push -u origin master
Username for 'https://github.com': suchithkumarch
Password for 'https://suchithkumarch@github.com':
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 8 threads
Compressing objects: 100% (34/34), done.
Writing objects: 100% (53/53), 13.55 KiB | 2.26 MiB/s, done.
Total 53 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/suchithkumarch/SciCal.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$
```



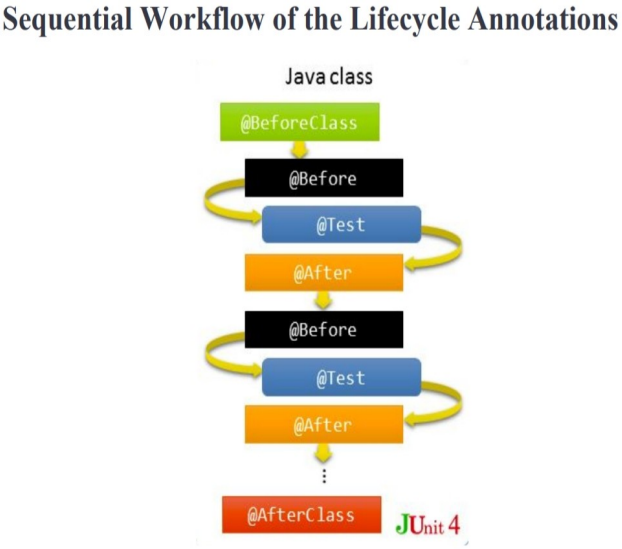
2. Testing, Building

2.1 Junit

Junit is a unit testing framework for the Java programming language. A JUnit test fixture is a Java object that allows us to write multiple test cases. Test methods must be annotated by the @Test. If the situation requires it, it is also possible to define a method to execute before (or after) each (or all) of the test methods with the @BeforeEach (or @AfterEach) and @BeforeAll (or @AfterAll) annotations.

Ex.

```
package SciCal;
//import SciCal.Main;
import static org.junit.Assert.*;
import org.junit.Test;

public class My_Test {
    private static final double DELTA = 1e-15;
    My_Main calculator = new My_Main();
//TruePositive
        @Test
        public void test_root() {
                double actual=calculator.root(16.0);
                double exp=4.0;
                assertEquals(actual,exp, DELTA);
        }
        @Test
        public void test_fact() {
                double actual=calculator.factorial(5);
                double exp=120;
                assertEquals(actual,exp, DELTA);
        }
        @Test
        public void test_log() {
                double actual=calculator.log(145.256);
                double exp=4.978497702968366;
                assertEquals(actual,exp, DELTA);
        }
        @Test
        public void test_power() {
                double actual=calculator.power(2.0,3.0);
                double exp=8.0;
                assertEquals(actual,exp, DELTA);
        }
//FalsePositive
        @Test
        public void test_root2() {
                double actual=calculator.root(17.0);
                double exp=4.0;
                assertNotEquals(actual,exp, DELTA);
        }
        @Test
        public void test_fact2() {
                double actual=calculator.factorial(6);
                double exp=120;
                assertNotEquals(actual,exp, DELTA);
        }
        @Test
        public void test_log2() {
                double actual=calculator.log(245.256);
                double exp=4.978497702968366;
                assertNotEquals(actual,exp, DELTA);
        }
        @Test
        public void test_power2() {
                double actual=calculator.power(3.0,2.0);
                double exp=8.0;
                assertNotEquals(actual,exp, DELTA);
        }
}
```



2.2 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 utilize a number of other development tools for reporting or the build process.

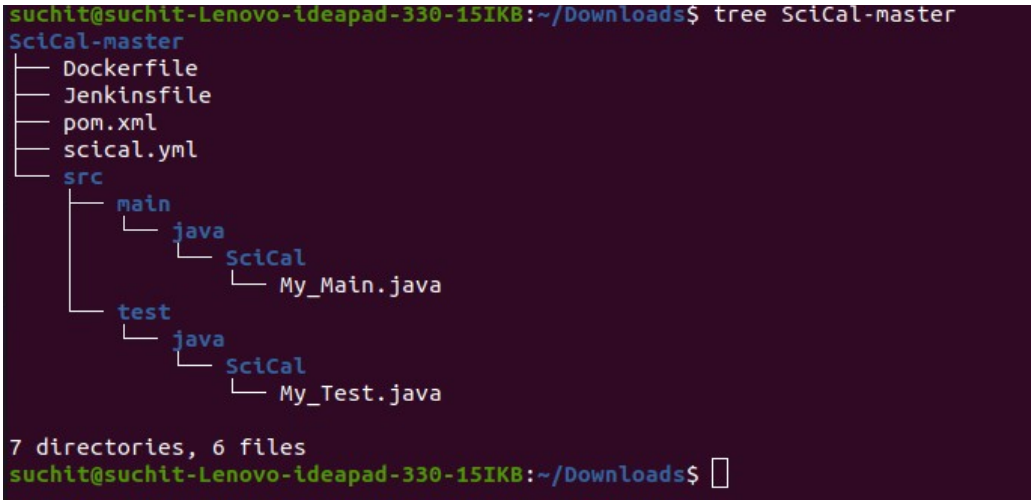
Installation:

$ apt-install maven

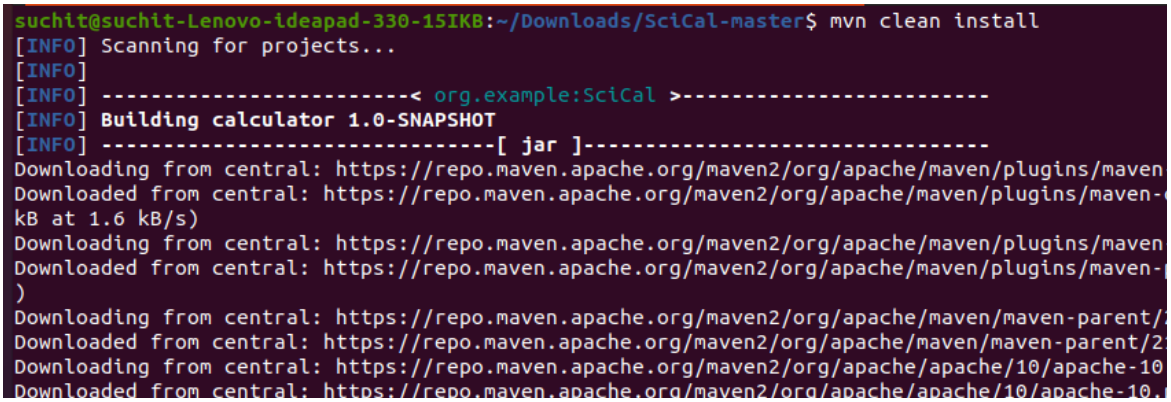
Verify using

$ mvn -version

Maven directory structure:



Run maven commands from project root directory.
$ mvn clean
$ mvn compile
$ mvn test
$ mvn install



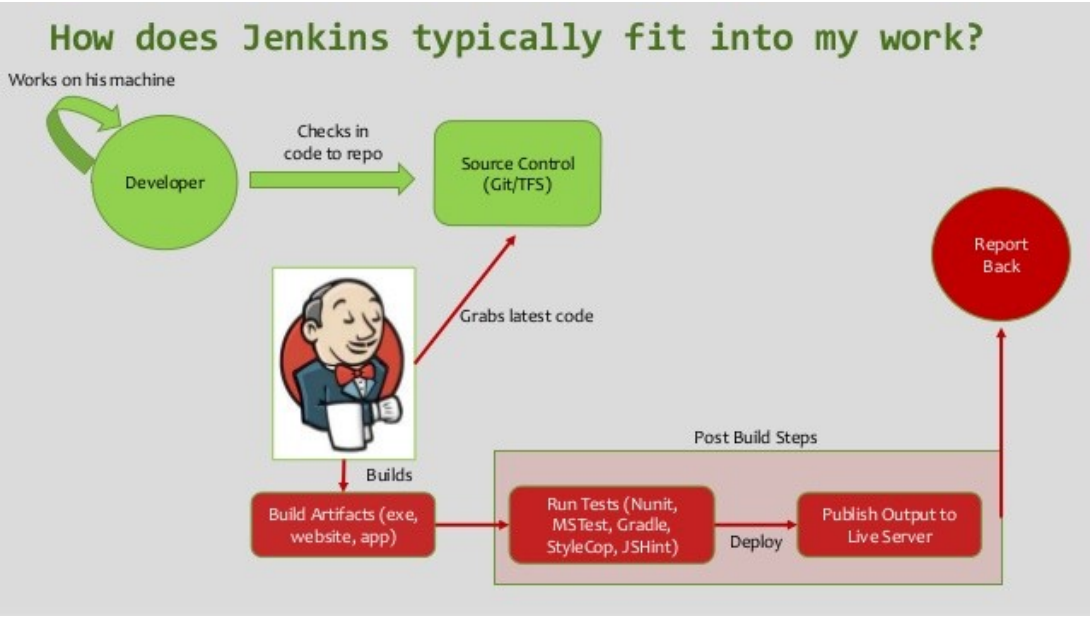
Build, compile, test and package the project into JAR. Running the jar file:

```
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ java -cp target/SciCal-1.0-SNAPSHOT.jar SciCal.My_Main
Hii, Welcome to Scientific Calculator!!
Choose which operation to do, by selecting the corrosponding number
Press 1 to get square root
Press 2 to get factorial
Press 3 to get natural logarithm
Press 4 to get power function
4
Enter the numbers
2.3 1.8
Power function result is : 4.478268254146011
Done, Bye!!
```

3. Continuous Integration

3.1 What is continuous integration?

It is a process of automating the build and testing of code every time when a change is commit to the version control. 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.



3.2 Jenkins

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

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'
$ apt update
$ apt install jenkins

Starting Jenkins
$ service jenkins start

Check if it is active
$ service jenkins status
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
$ cat /var/lib/jenkins/secrets/initialAdminPassword

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

4. Containerization

4.1 Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host.

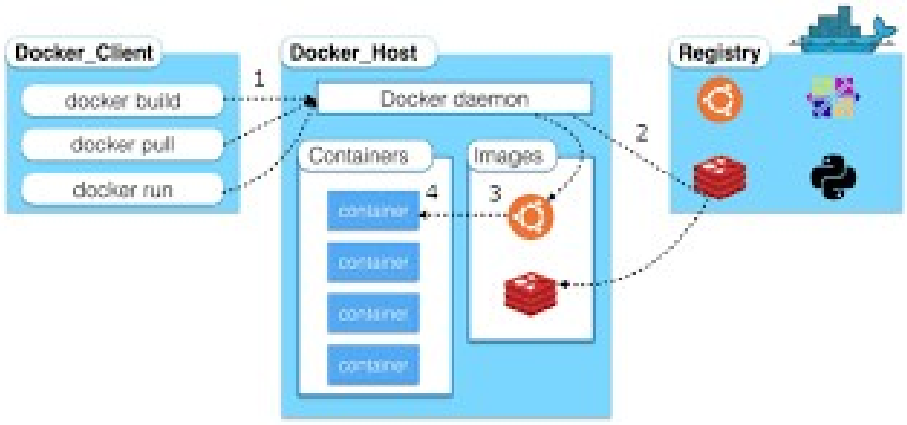
Installation:

Update your existing list of packages
$ sudo apt update
Add the Docker repository to APT sources
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable"Install Docker
$ sudo apt install docker-ce
Check if the status of docker daemon is running to ensure docker is installed

$ service docker status

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

Docker Commands:
To pull image and if not found on local machine pull from dockerhub
$ docker pull <image_name>To list images downloaded to your computer
$ docker images
To run image in interactive mode
$ docker run -i -t --name <Container_Name> <Image_Name>
To list active containers
$ docker ps -a
To commit the changes in a container to a new Docker image
$ docker commit -m "What you did to the image" -a "Author Name" container_id repository/new_image_name



4.2 Create Docker Image

To run JAR application in container, Java needs to be installed in the container.

Installing java in the base image[ubuntu] by running following commands:

$ apt-get install -y ant
$ apt-get clean
*$ rm -rf /var/lib/apt/lists/**
$ rm -rf /var/cache/oracle-jdk8-installer

Fix certificate issues, found as of:
$ apt-get update
$ apt-get install -y ca-certificates-java
$ apt-get clean
$ update-ca-certificates -f

*$ rm -rf /var/lib/apt/lists/**
$ rm -rf /var/cache/oracle-jdk8-installer
$ JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
$ export JAVA_HOME

Now commit the changes in a container to a new Docker image.

4.3 Docker File

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

```
# Set the base image
FROM openjdk:8
# File Author
MAINTAINER Suchith Kumar suchithkumar.ch@gmail.com
#Copy the jar file into the same directory of dockerfile
COPY ./target/SciCal-1.0-SNAPSHOT.jar ./
# Set Working Directory
WORKDIR ./
# Container command for executing jar file
CMD ["java", "-cp", "SciCal-1.0-SNAPSHOT.jar", "SciCal.My_Main"]
```

Image build from dockerfile, running the image, login and push into dockerhub commands:

```
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ docker image build -t suchithkumarch/scical:latest .
Sending build context to Docker daemon  162.3kB
Step 1/5 : FROM openjdk:8
 ---> f28d33bb11eb
Step 2/5 : MAINTAINER Suchith Kumar suchithkumar.ch@gmail.com
 ---> Using cache
 ---> 24b6fef05c3e
Step 3/5 : COPY ./target/SciCal-1.0-SNAPSHOT.jar ./
 ---> ba23ffa9a24a
Step 4/5 : WORKDIR ./
 ---> Running in c2e6b02f951e
Removing intermediate container c2e6b02f951e
 ---> 421b1b3a7d9f
Step 5/5 : CMD ["java", "-cp", "SciCal-1.0-SNAPSHOT.jar", "SciCal.My_Main"]
 ---> Running in e4030181aec7
Removing intermediate container e4030181aec7
 ---> 8443c1c1f956
Successfully built 8443c1c1f956
Successfully tagged suchithkumarch/scical:latest
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ docker container run --interactive suchithkumarch/scical:latest
Hii, Welcome to Scientific Calculator!!
Choose which operation to do, by selecting the corrosponding number
Press 1 to get square root
Press 2 to get factorial
Press 3 to get natural logarithm
Press 4 to get power function
1
Enter the number
3
Square root result is : 1.7320508075688772
Done, Bye!!
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: suchithkumarch
Password:
WARNING! Your password will be stored unencrypted in /home/suchit/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ docker push suchithkumarch/scical
```

4.4 Dockerhub

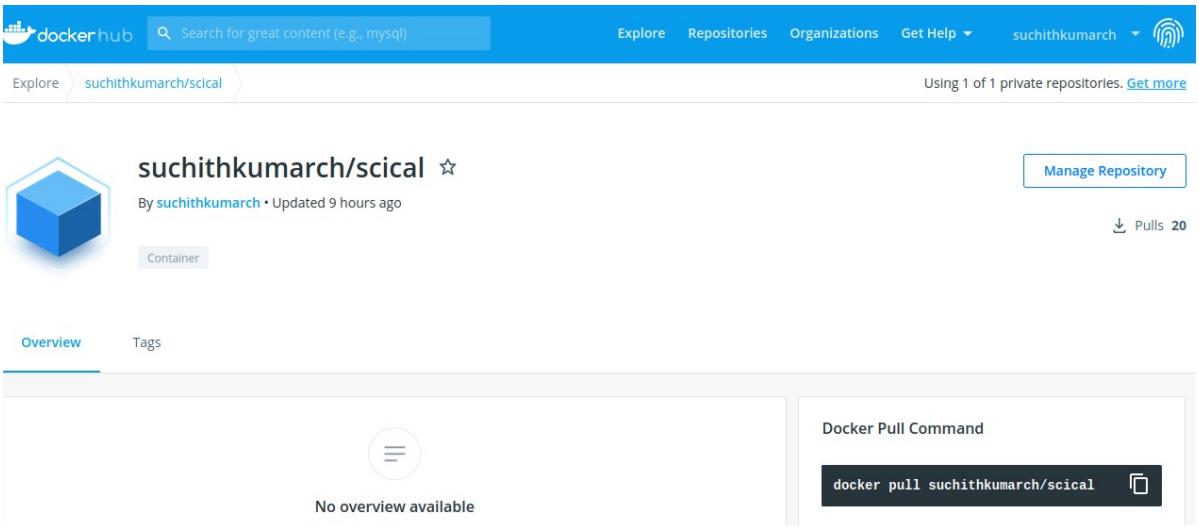
Docker Hub offers an easy way to create, manage, and deliver teams' container applications. Here, we can find official images created by companies as well as customized images from different users. We create our own repository.

Creating an account at DockerHub and creating a repository:

Signin into https://hub.docker.com/ account create a repository.

Adding the credentials of DockerHub account to Jenkins:

Credentials ->System - > Provide username and password of DockerHub -> Provide an ID for this credentials as DockerHub.



We will be using this later in the pipeline to build and deploy image on our DockerHub repository.

5. Deployment

5.1 Ansible

Ansible is an Open Source automation platform, an Automation Engine that runs ansible playbooks. Playbooks are defined tasks, where we define environments and workflows. Ansible connects to the hosts it manages using openssh or winrm and run tasks. These tasks are small programs pushed to the hosts.
Before installing ansible, python and ssh must be installed.

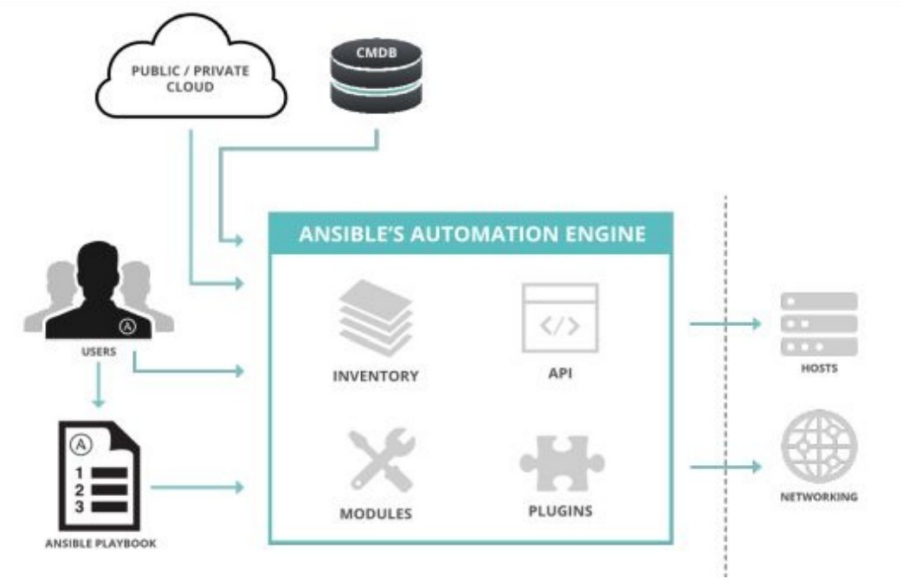
Python installation:
$sudo apt update
$sudo apt install software-properties-common
$sudo add-apt-repository ppa:deadsnakes/ppa

$sudo apt update
$sudo apt install python3.8

SSH installation:
$ sudo apt install openssh-server
$ ssh-keygen -t rsa

Ansible installation:
$ sudo apt update
$ sudo apt install ansible

Ansible Configuration:
$ sudo apt install sshpass
$ sudo gedit /etc/ssh/sshd_config
 # set PermitRootLogin as yes and uncomment
$ sudo systemctl restart sshd



5.2 Implementing Playbooks

Ad hoc commands can run a single, simple task against a set of targeted hosts as a one-time command. The real power of Ansible, however, is in learning how to use playbooks to run multiple, complex tasks against a set of targeted hosts in an easily repeatable manner. A play is an ordered set of tasks run against hosts selected from your inventory. A playbook is a text file containing a list of one or more plays to run in a specific order. A playbook is a text file written in YAML format, and is normally saved with the extension yml. The playbook uses indentation with space characters to indicate the structure of its data. To help you understand the format of a playbook, let's write an ad hoc command 'id' into a single task play and save it as a playbook. The resulting playbook appears as follows:

```
---
- name: Run id to playbook
hosts: all
tasks:
- name: Run id command
command:
cmd: id
```

5.3 Running Playbooks

The ansible-playbook command is used to run playbooks. The command is executed on the control node and the name of the playbook to be run is passed as an argument:

$ ansible-playbook playbook_file_name.yml

When you run the playbook, output is generated to show the play and tasks being executed. The output also reports the results of each task executed.
When you rerun the playbook, tasks in Ansible Playbooks are idempotent, and it is safe to run a playbook multiple times. If the targeted managed hosts are already in the correct state, no changes should be made.

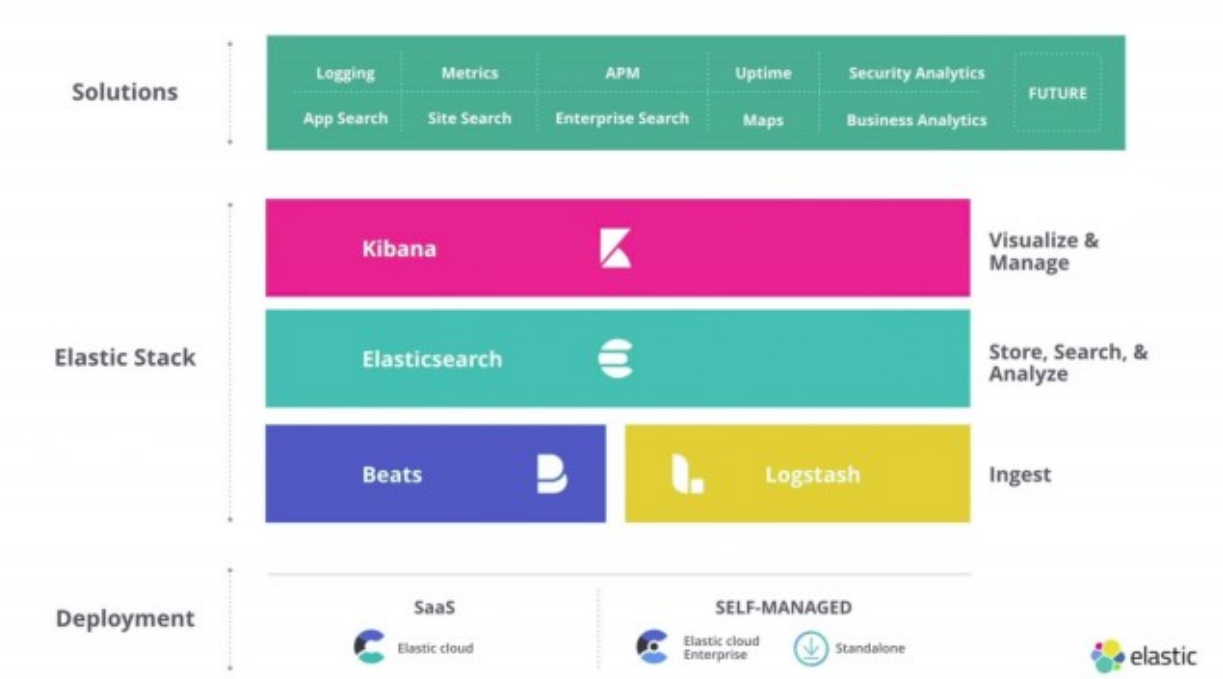
Syntax Verification:
$ ansible-playbook --syntax-check playbook_file_name.yml

Executing a dry run:
$ ansible-playbook -C playbook_file_name.yml

```
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$ ansible-playbook scical.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Deploy docker img] ********************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [localhost]

TASK [Pull scical image] ********************************************************************************************************
ok: [localhost]

PLAY RECAP **********************************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/SciCal-master$
```

6. Monitoring using ELK

Continuous monitoring provides near immediate feedback and insight into performance and interactions across the network.
"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.



6.1 Logstash, configuration file

Downloading Logstash:
https://www.elastic.co/downloads/logstash

Download appropriate version for your OS. (Ex. LinuxX86_64)
After the tar.gz file is downloaded, use this command to extract
$tar -xvf path/to/file/tar.gz

Configuration file:

The bare structureof the configurationfile of logstash is
input {
plugin {
settings
}
}
filter {
plugin {
settings
}
}
output {
plugin{
settings
}
}
To run logstash, we have to input a configuration file on how to parse the data of a given log file.
Go into the extracted logstach directory and execute this command to run.
$./bin/logstash -f /path/to/configuration/file

Command and the Output :

```
suchit@suchit-Lenovo-ideapad-330-15IKB:~/Downloads/logstash-7.11.2$ ./bin/logstash -f /home/suchit/Downloads/SciCal-master/calculator_logstash
.conf
Using JAVA_HOME defined java: /usr/lib/jvm/java-8-openjdk-amd64
WARNING, using JAVA_HOME while Logstash distribution comes with a bundled JDK
Sending Logstash logs to /home/suchit/Downloads/logstash-7.11.2/logs which is now configured via log4j2.properties
[2021-03-14T17:46:13,349][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"7.11.2", "jruby.version"=>"jruby 9.2.13.0
(2.5.7) 2020-08-03 9a89c94bcc OpenJDK 64-Bit Server VM 25.282-b08 on 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08 +indy +jit [linux-x86_64]"}
```

```
}
{
    "@timestamp" => 2021-03-14T07:52:39.839Z,
          "path" => "/home/suchit/Downloads/SciCal-master/calculator.log",
      "@version" => "1",
       "message" => "14/Mar/2021:13:22:39 839 [My_Main.java] [INFO] SciCal.My_Main [POWER] - 1.0, 3.0",
        "logger" => "SciCal.My_Main",
          "line" => "1.0, 3.0",
         "level" => "INFO",
          "host" => "suchit-Lenovo-ideapad-330-15IKB",
        "action" => "POWER",
        "thread" => "My_Main.java"
}
{
    "@timestamp" => 2021-03-14T07:55:21.205Z,
          "path" => "/home/suchit/Downloads/SciCal-master/calculator.log",
      "@version" => "1",
       "message" => "14/Mar/2021:13:25:21 205 [My_Main.java] [INFO] SciCal.My_Main [RESULT - ROOT] - 4.0",
        "logger" => "SciCal.My_Main",
          "line" => "4.0",
         "level" => "INFO",
          "host" => "suchit-Lenovo-ideapad-330-15IKB",
        "action" => "RESULT - ROOT",
        "thread" => "My_Main.java"
}
{
    "@timestamp" => 2021-03-14T07:55:21.213Z,
          "path" => "/home/suchit/Downloads/SciCal-master/calculator.log",
      "@version" => "1",
       "message" => "14/Mar/2021:13:25:21 213 [My_Main.java] [INFO] SciCal.My_Main [POWER] - 2.0, 3.0",
        "logger" => "SciCal.My_Main",
          "line" => "2.0, 3.0",
         "level" => "INFO",
          "host" => "suchit-Lenovo-ideapad-330-15IKB",
        "action" => "POWER",
        "thread" => "My_Main.java"
}
```

6.2 Elasticsearch, Kibana Visualizations

Download the Elasticsearch archive for your OS
https://www.elastic.co/downloads/elasticsearch

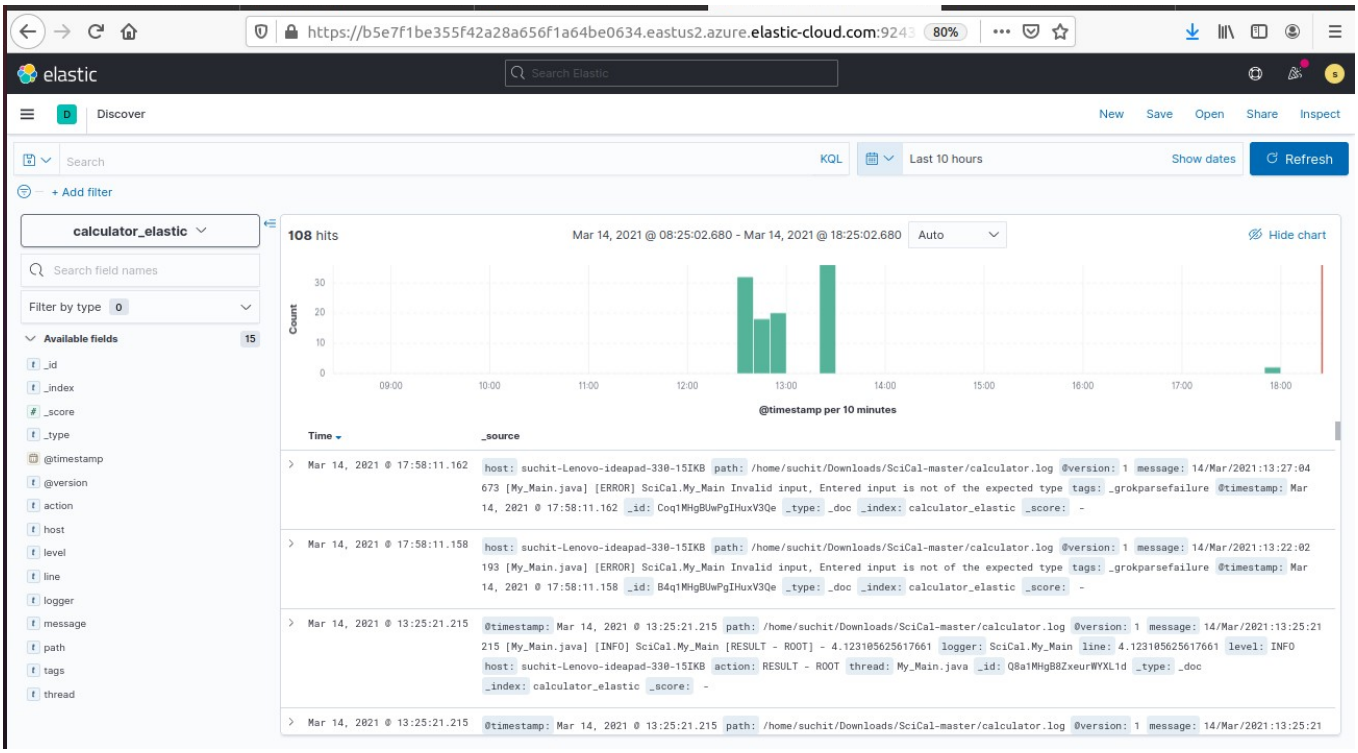
Download the Kibana archive for your OS,
https://www.elastic.co/downloads/kibana

Or use cloud service for the above. But the configuration file we have to be changed accordingly.

Kibana is an open source browser based visualization tool mainly used to analyze large volumes of logs in the form of line graph, bar graph, pie charts, heat maps, region maps, coordinate maps, gauge, goals, timelion etc. The visualization makes it easy to predict or to see the changes in trends of errors or other significant events of the input source.

With the index pattern we tell kibana what Elasticsearch index to analyze. Follow the following steps to set up the index pattern:

In Kibana, go to Management → Stack Management
*Look for Kibana → Index Patterns → Create Index Pattern , set your index pattern based on the index pattern provided in logstash configuration file followed by *. And in the next step select @timestamp as your Time field. Hit Create index pattern , and you are ready to analyze the data. Explore discover and Dashboard options.*



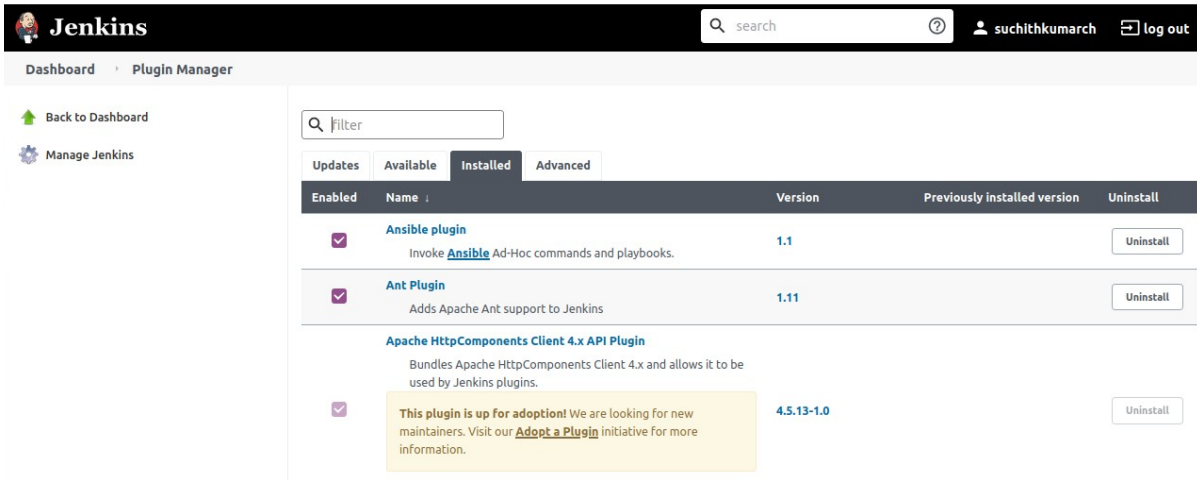
7. Pipeline to Integrate SCM, Build Image and Deploy through Ansible using Jenkins

Configure Jenkins by installing plugins and make some configuration to run project in automated pipeline manner.

Install Plugins:

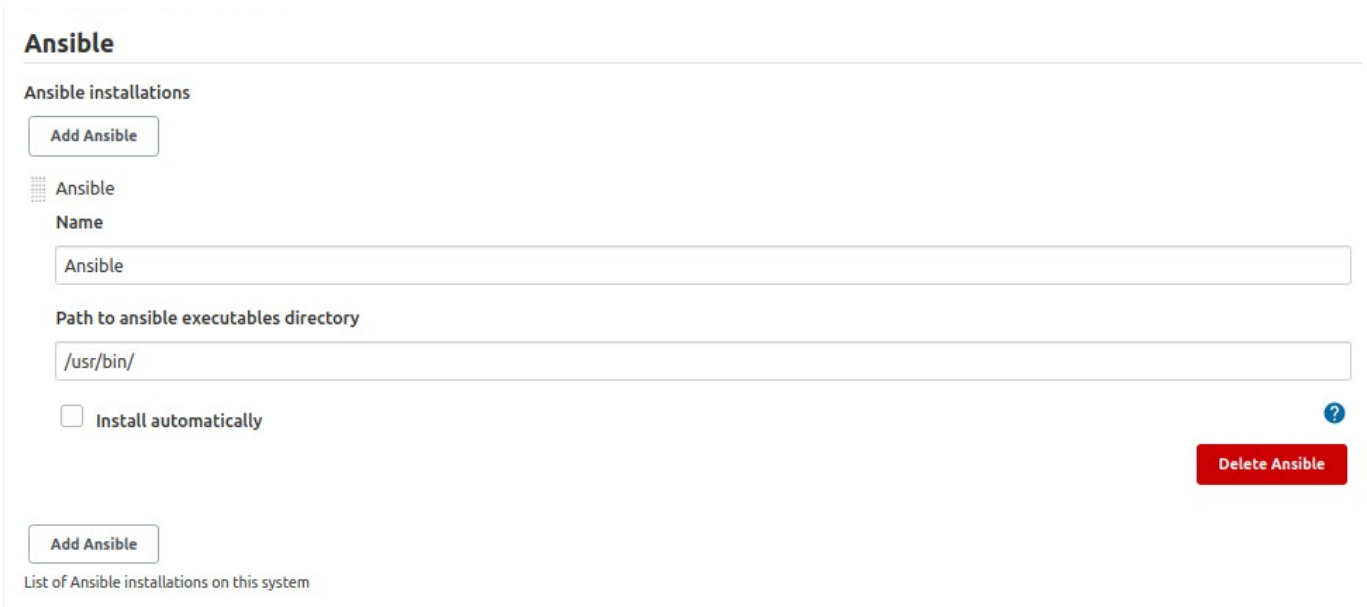
If not-vulnerable
Manage Jenkins -> Manage Plugins -> Available -> Filter -> Install without restart

If vulnerable
Download stable plugin file from https://updates.jenkins-ci.org/download/plugins/
Manage Jenkins -> Manage Plugins -> Advanced -> Upload HPI file -> Install without restart
1. Git
2. GitHub plugin
3. Maven Integration plugin
4. Docker plugin
5. Pipeline
6. Ansible plugin



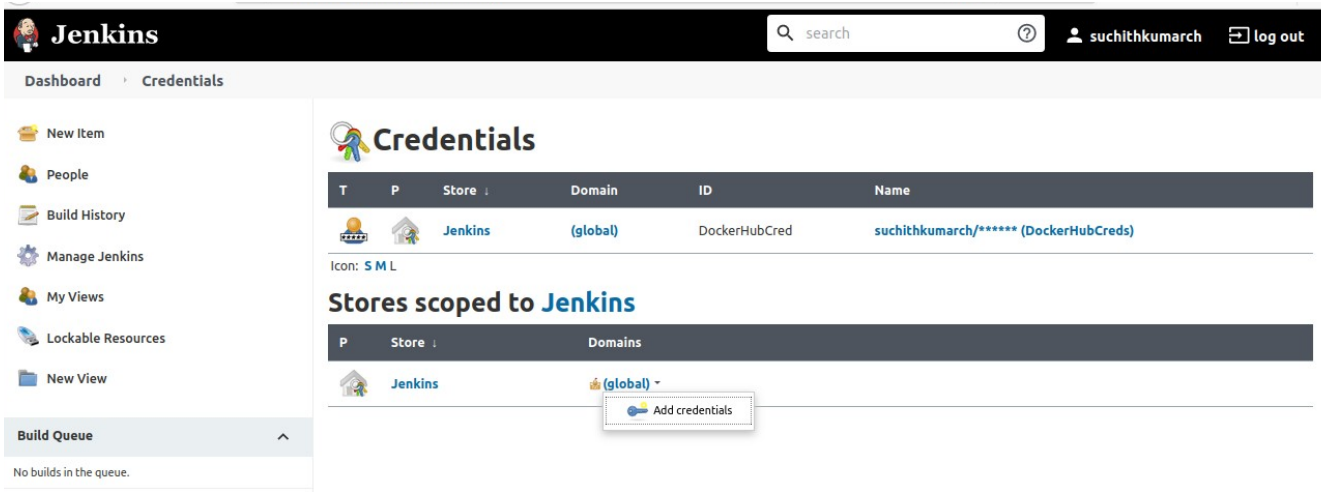
Configure Systems:

Manage Jenkins -> Configure System-> Add Ansible
Instances:
Provide name
Provide path - which can be found using
$whereis ansible



Adding the credentials of DockerHub account to Jenkins:

Credentials ->System - > Provide username and password of DockerHub -> Provide an ID for this credentials as DockerHub.

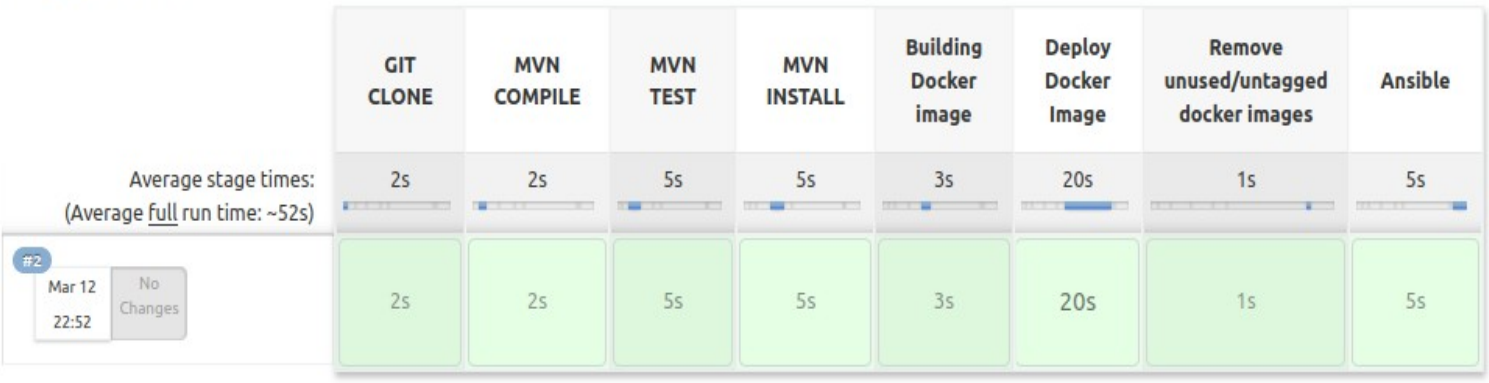


Create Jenkins Pipeline:

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 Ansible is triggered from here, which will deploy the final containerized application to the host node(s).

Create Jenkinsfile -
Jenkins -> New Item -> Enter Item Name-> Pipeline -> OK

Pipeline Execution:



Links

Github:
https://github.com/suchithkumarch/SciCal

Dockerhub:
https://hub.docker.com/r/suchithkumarch/scical

References

Maven commands
https://www.journaldev.com/33645/maven-commands-options-cheat-sheet

Installing Docker
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04

Create Jenkin pipeline
https://www.edureka.co/community/55640/jenkins-docker-docker-image-jenkins-pipeline-docker-registry