

CS 816 - Software Production Engineering (SPE)

Mini Project - Scientific Calculator with DevOps

Name : Akshay Girish Thite

Roll No. : MT2021010

Batch : MTech CSE

Problem Statement :

Create a scientific calculator program with user menu driven operations

- Square root function - \sqrt{x}
- Factorial function - $x!$
- Natural logarithm (base e) - $\ln(x)$
- Power function - x^y

Introduction :

In this project we implement a scientific calculator with the operations such as square root, factorial, natural logarithm and power function using DevOps tools. It is a command line menu driven application.

DevOps - DevOps can be defined as a set of practices that combine software development and IT operations. The main aim of DevOps is shortening the systems development life cycle and providing continuous delivery with high software quality.



Tools and Technology used :

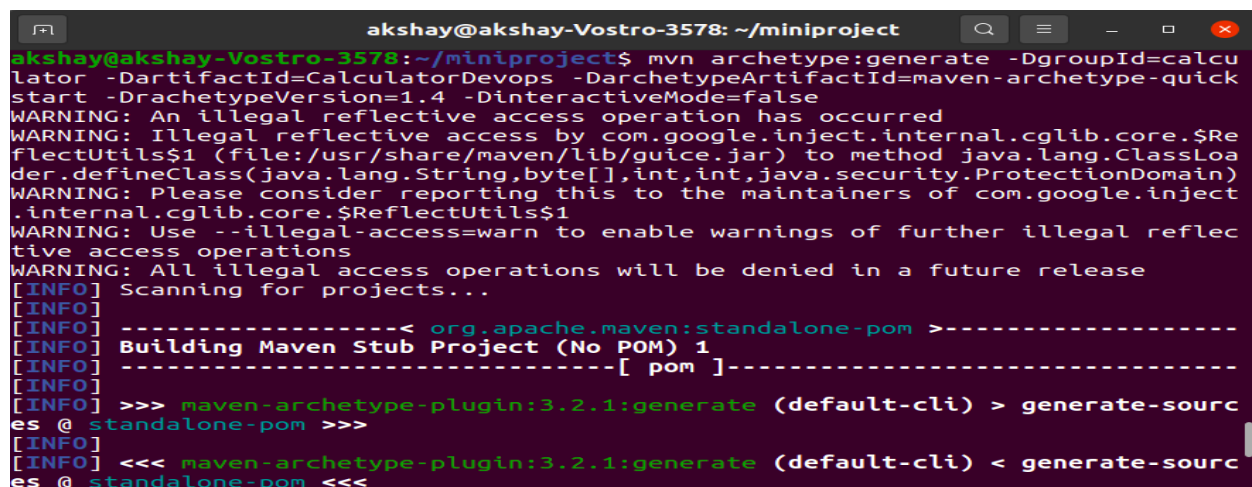
1. Programming language : Java
2. Testing : JUnit
3. Building tool : Apache Maven
4. Source Code Management : GitHub for Git
<https://github.com/AkshayThite10/CalculatorDevops.git>
5. Containerization : Docker
6. Continuous Integration : Jenkins
7. Continuous Deployment : Ansible
8. Generating Logs : Log4j
9. Monitoring : ELK Stack

Building Tool (Apache Maven) :

Maven is a powerful project management tool. It is based on the project object model (POM). Project build, dependency and documentation can be done using maven. It is basically a tool that is used for building and managing any Java based application. In this project, maven is used to build the java source code by configuring all the dependencies and creating a binary executable .jar file of it in the target directory. "pom.xml" is a configuration file in a maven project. It contains and manages the dependencies, plugins and metadata for the project.

Steps in building :

1. In the project directory, create maven project hierarchy

A terminal window titled 'akshay@akshay-Vostro-3578: ~/miniproject' shows the execution of the command 'mvn archetype:generate -DgroupId=calculator -DartifactId=CalculatorDevops -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false'. The output includes several warnings about illegal reflective access and a message to use '--illegal-access=warn'. It then shows the generation of a 'standalone-pom' file, displaying its contents: a Maven project with groupId 'org.apache.maven', artifactId 'standalone-pom', version '1', and packaging 'pom'. The terminal output ends with 'es @ standalone-pom <<<'.

Command : mvn archetype:generate -DgroupId=calculator
-DartifactId=CalculatorDevops -DarchetypeArtifactId=maven-archetype-quickstart
-DarchetypeVersion=1.4 -DinteractiveMode=false

2. Writing the calculator java program in App.java and JUnit testing program in AppTest.java.

Functions of App.java :

```
86     public static double sqroot(double num) {
87         double res;
88         res = Math.sqrt(num);
89         System.out.print("\n");
90         System.out.println("Square root = "+res);
91         System.out.print("\n");
92         logger.info("Square root computed");
93         return res;
94     }
95     public static int fact(int num){
96         int res=num,i;
97         if(num==0)
98         {
99             res=1;
100         }
101         for(i=num-1;i>=1;i--)
102         {
103             res=res*i;
104         }
105         System.out.print("\n");
106         System.out.println("Factorial = "+res);
107         System.out.print("\n");
108         logger.info("Factorial computed");
109         return res;
110     }
111     public static double natlog(double num){
112         double res = Math.log(num);
113         System.out.print("\n");
114         System.out.println("Natural log = "+ res);
115         System.out.print("\n");
116         logger.info("Natural log computed");
117         return res;
118     }
119     public static double powerfxn(double num, double exponent){
120         double res = Math.pow(num,exponent);
121         System.out.print("\n");
122         System.out.println("Power = "+ res);
123         System.out.print("\n");
124         logger.info("Power computed");
125         return res;
126     }
127 }
```

AppTest.java :

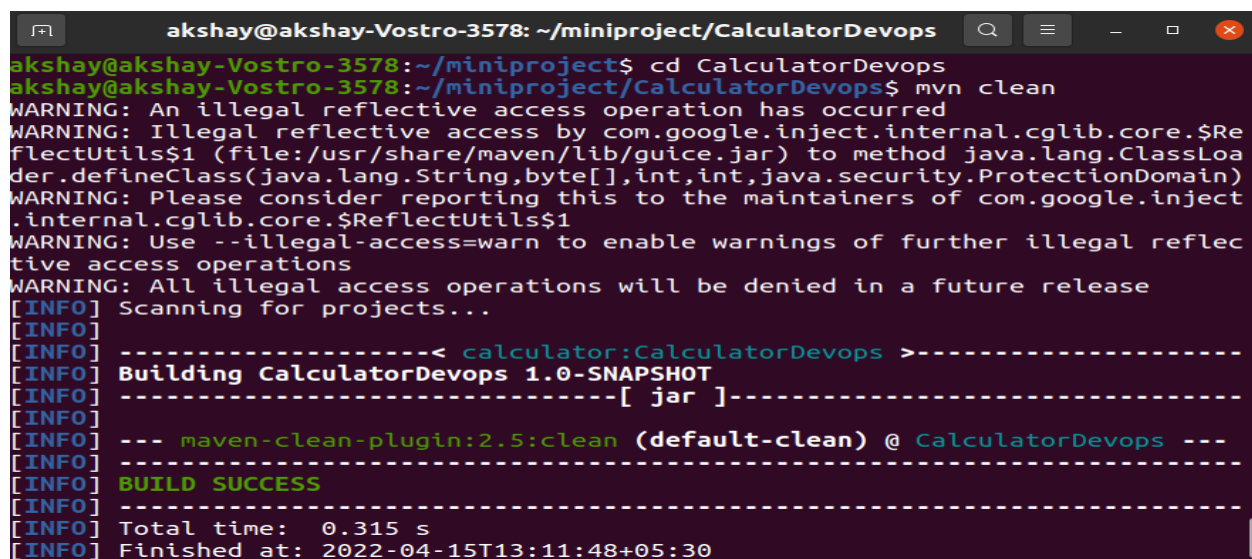
```
1  // package calculator;
2  import org.junit.Assert;
3  import org.junit.Before;
4  import org.junit.Test;
5
6  public class AppTest {
7      @Test
8      public void test_squareRoot() {
9          double a = 81.0;
10         double expectedResult = 9.0;
11         double result = App.sqroot(a);
12         Assert.assertEquals(expectedResult, result, 0.0f);
13     }
```

```

15     @Test
16     public void test_factorial() {
17         int a = 4;
18         int expectedResult = 24;
19         int result = App.fact(a);
20         Assert.assertEquals(expectedResult, result);
21     }
22
23     @Test
24     public void test_naturalLog() {
25         double a = 2.718;
26         double expectedResult = 1.0;
27         double result = App.natlog(a);
28         Assert.assertEquals(expectedResult, result, 0.2f);
29     }
30
31     @Test
32     public void test_power() {
33         double a = 2.0;
34         double b = 10.0;
35         double expectedResult = 1024.0;
36         double result = App.powerfxn(a,b);
37         Assert.assertEquals(expectedResult, result, 0.0f);
38     }
39 }
40

```

3. Maven clean, compile, install and site



```

akshay@akshay-Vostro-3578: ~/miniproject/CalculatorDevops
akshay@akshay-Vostro-3578:~/miniproject/CalculatorDevops$ mvn clean
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$Re
flectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoa
der.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject
.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflec
tive access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< calculator:CalculatorDevops >-----
[INFO] Building CalculatorDevops 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ CalculatorDevops ---
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.315 s
[INFO] Finished at: 2022-04-15T13:11:48+05:30

```

Command : mvn clean

It is used for cleaning the project hierarchy. It cleans cache in the maven hierarchy. If there are any previous builds in the 'target' directory then they would get deleted. Now, the new build can take place.

```
akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...
akshay@akshay-Vostro-3578: ~/Doc...
akshay@akshay-Vostro-3578: ~/Documents/miniproject/CalculatorDevops$ mvn compile
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for calculator:CalculatorDevops:jar:1.0-SNAPSHOT
[WARNING] 'dependencies.dependency.version' for junit:junit:jar is either LA TEST or RELEASE (both of them are being deprecated) @ line 25, column 15
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threat
```

Command : mvn compile

It compiles all the source code in the 'src/main/java/<packagename>' directory.

```
akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...
akshay@akshay-Vostro-3578: ~/Doc...
akshay@akshay-Vostro-3578: ~/Documents/miniproject/CalculatorDevops$ mvn install
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for calculator:CalculatorDevops:jar:1.0-SNAPSHOT
[WARNING] 'dependencies.dependency.version' for junit:junit:jar is either LA TEST or RELEASE (both of them are being deprecated) @ line 25, column 15
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threat
```

Command : mvn install

It is used for building the project. It creates a binary executable .jar file of the project in the 'target' folder.

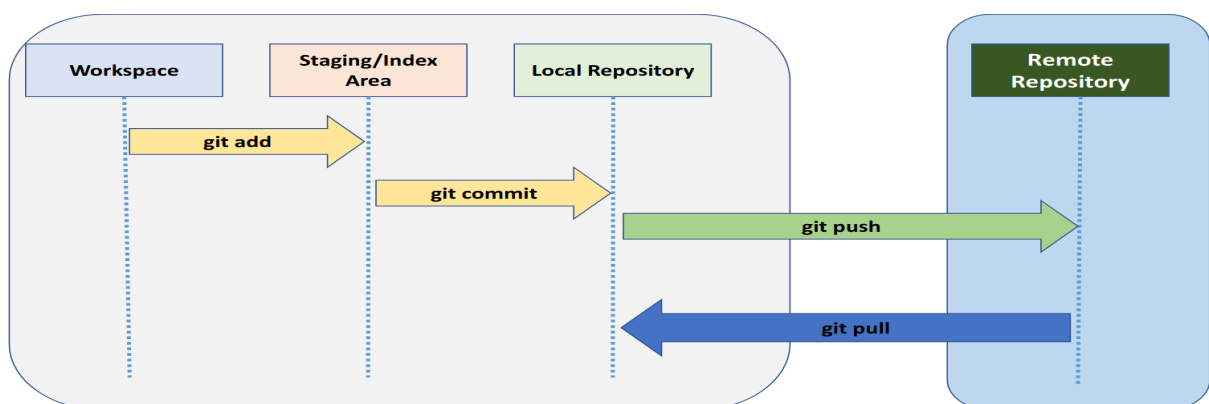
```
akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...
akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ mvn site
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for calculator:CalculatorDevops:jar:1.0-SNAPSHOT
[WARNING] 'dependencies.dependency.version' for junit:junit:jar is either LA TEST or RELEASE (both of them are being deprecated) @ line 25, column 15
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
```

Command : mvn site

It creates a documentation type report using HTML and CSS in the 'target/site' directory.

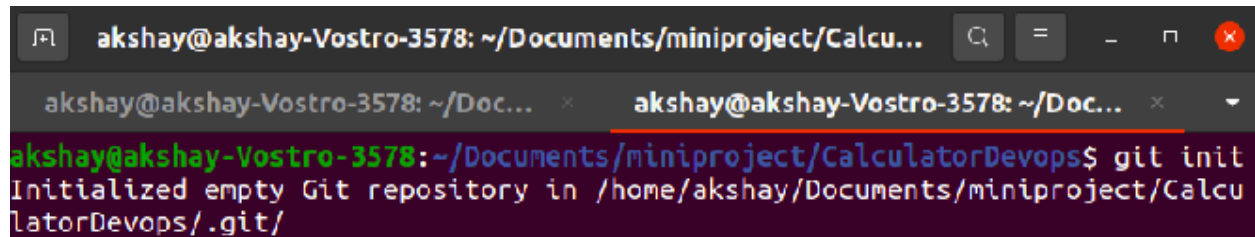
Source Code Management :

It is used to track modifications to the source code repository. It tracks the history of changes to the code base. It also helps to resolve conflicts when merging updates from multiple contributors. Source Code Management is also known as the Version Control System as it stores the various versions of the project which may be useful later.



Steps to push the Maven project from the workspace to the remote GitHub repository :

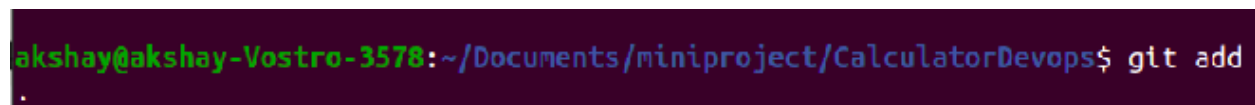
1. Initializing Git in the current workspace.

A terminal window with a dark background. The prompt is 'akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...'. The command 'git init' has been entered, and the output is 'Initialized empty Git repository in /home/akshay/Documents/miniproject/CalculatorDevops/.git/'.

```
akshay@akshay-Vostro-3578: ~/Documents/miniproject/Calcu...
akshay@akshay-Vostro-3578: ~/Doc...
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ git init
Initialized empty Git repository in /home/akshay/Documents/miniproject/CalculatorDevops/.git/
```

Command : git init

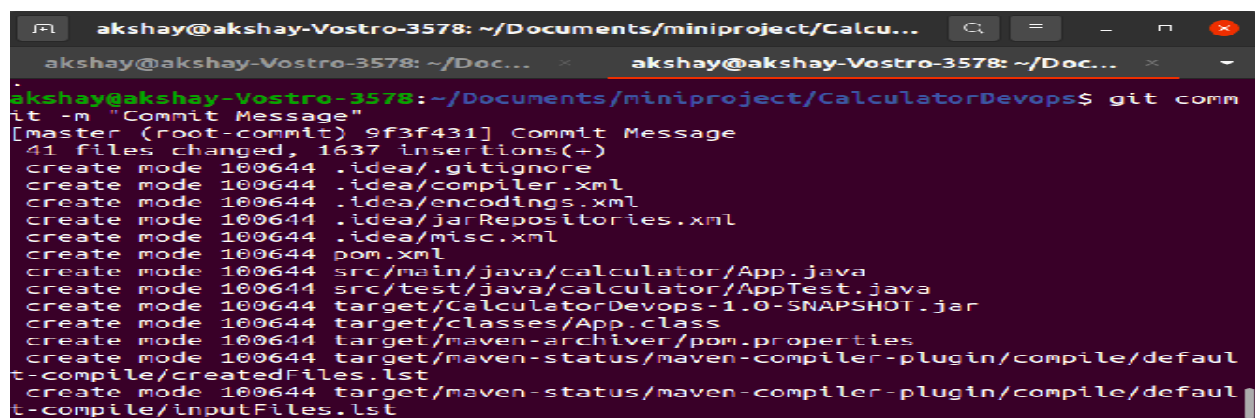
2. Adding the project to the staging area.

A terminal window showing the command 'git add .' being executed. The output is a single dot '.'.

```
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ git add
.
```

Command : git add .

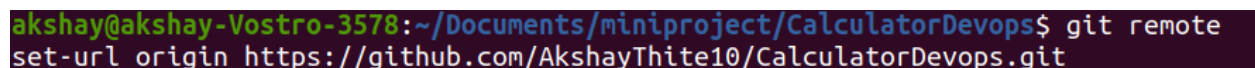
3. Committing the changes to the local git repository.

A terminal window showing the command 'git commit -m "Commit Message"' being executed. The output shows the commit details: '[master (root-commit) 9f3f431] Commit Message', '41 files changed, 1637 insertions(+)', and a list of files created with their modes and permissions.

```
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ git commit -m "Commit Message"
[master (root-commit) 9f3f431] Commit Message
41 files changed, 1637 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/compiler.xml
create mode 100644 .idea/encodings.xml
create mode 100644 .idea/jarRepositories.xml
create mode 100644 .idea/misc.xml
create mode 100644 pom.xml
create mode 100644 src/main/java/calculator/App.java
create mode 100644 src/test/java/calculator/AppTest.java
create mode 100644 target/CalculatorDevops-1.0-SNAPSHOT.jar
create mode 100644 target/classes/App.class
create mode 100644 target/maven-archiver/pom.properties
create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-t-compile/createdFiles.lst
create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-t-compile/inputFiles.lst
```

Command : git commit -m "Commit Message"

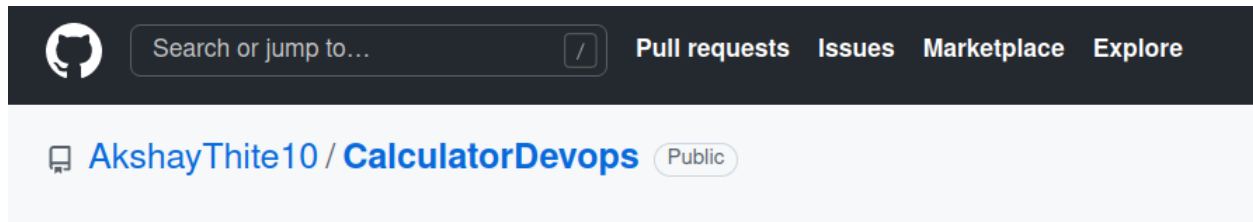
4. Setting an origin for a remote repository on GitHub.

A terminal window showing the command 'git remote set-url origin https://github.com/AkshayThite10/CalculatorDevops.git' being executed. The output is empty, indicating the command was successful.

```
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ git remote set-url origin https://github.com/AkshayThite10/CalculatorDevops.git
```


Command : `git remote add origin "url of the remote repository"` (if origin not set earlier)
`git remote set-url origin "url of the remote repository"` (if origin was set)

5. Pushing the project to the remote repository to the 'master' branch.



```
akshay@akshay-Vostro-3578:~/Documents/miniproject/CalculatorDevops$ git push -u origin master
Username for 'https://github.com': AkshayThite10
Password for 'https://AkshayThite10@github.com':
Branch 'master' set up to track remote branch 'master' from 'origin'.
Everything up-to-date
```

Command : `git push -u origin master`

Containerization (Docker) :

Docker is an open platform that is used for developing, shipping and running applications. It separates application from infrastructure. So, delivery of software is faster. It is possible to manage infrastructure in the same way as that of application.

Steps :

1. Install Docker.

A screenshot of a terminal window. The title bar shows the user 'akshay@akshay-Vostro-3578' and the current directory '~/miniproject/CalculatorDevops'. The terminal shows the command `sudo apt-get install docker.io` being executed. The output indicates that the package lists are read, the dependency tree is built, and state information is read. However, an error occurs: 'Some packages could not be installed. This may mean that you have requested an impossible situation or if you are using the unstable distribution that some required packages have not yet been created or been moved out of Incoming.' It then lists unmet dependencies for 'docker.io', specifically that it depends on 'containerd (>= 1.2.6-0ubuntu1~)' which is not available. The final message is 'E: Unable to correct problems, you have held broken packages.'

Command : `sudo apt-get install docker.io`

2. Creating a repository in dockerhub.

Link : https://hub.docker.com/r/akshaythite10/calculator_devops

TAG			
latest			
Last pushed a day ago by akshaythite10			
DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE ⓘ
e149ecf3355c	linux/amd64	---	227.8 MB

3. Creating a Dockerfile

Docker automatically builds the images according to the instructions present in the Dockerfile. Dockerfile is a file that contains all the commands to build an image. Here, we need to create a container that executes the built maven project.

```
📄 Dockerfile
1 FROM openjdk:8
2 COPY ./target/CalculatorDevops-1.0-SNAPSHOT-jar-with-dependencies.jar ./
3 WORKDIR ./
4 CMD ["java", "-cp", "CalculatorDevops-1.0-SNAPSHOT-jar-with-dependencies.jar", "App"]
5
```

After creating this Dockerfile, push it to the GitHub repository.

After running `docker build` using this Dockerfile as the source will create the required docker image that is ready to run on our application. Docker image needs to be built and then to be pushed to the DockerHub.

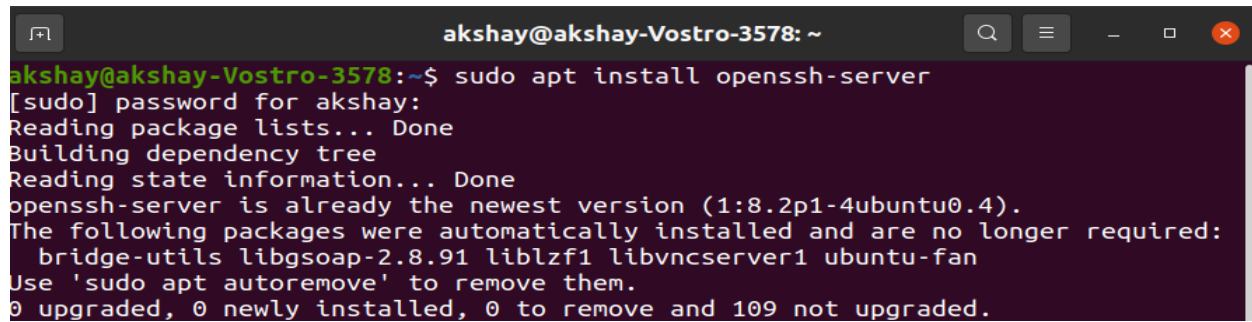
Continuous Integration (Jenkins) :

Steps :

1. Add Jenkins to the Docker group.

Command : `sudo apt install openssh-server`

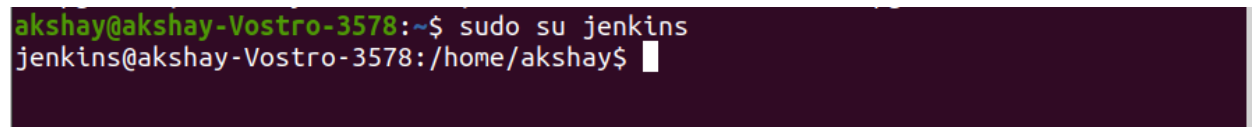
This will install the openssh server.



```
akshay@akshay-Vostro-3578: ~  
akshay@akshay-Vostro-3578:~$ sudo apt install openssh-server  
[sudo] password for akshay:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
openssh-server is already the newest version (1:8.2p1-4ubuntu0.4).  
The following packages were automatically installed and are no longer required:  
  bridge-utils libgsoap-2.8.91 liblzfl libvncserver1 ubuntu-fan  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 109 not upgraded.
```

Command : `sudo su jenkins`

It is used for logging in to jenkins.



```
akshay@akshay-Vostro-3578:~$ sudo su jenkins  
jenkins@akshay-Vostro-3578:/home/akshay$
```

Commands : `mkdir .ssh`

`cd .ssh`

`ssh-keygen -t rsa`

`ssh-copy-id akshay@localhost`

`ssh akshay@localhost`

These commands are used for configuring jenkins to use docker via ssh. After executing these commands we get directed out of Jenkins user.

Command : `sudo systemctl start jenkins`

It is used for starting jenkins at the respective port number.


2. Manage Plugins : All the required plugins need to be installed in jenkins such as Pipeline, Docker, Maven Integration, Ansible, GitHub etc. After installing, Jenkins needs to be restarted and Docker credentials need to be added. In Jenkins dashboard, credentials to the DockerHub repository needs to be added and a unique id is set that is equal to docker with Registry credentials id in the pipeline script.

3. Create a new pipeline in Jenkins.

Enter an item name


DevOps_Calculator1

» Required field




Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

4. Setup a Jenkins Pipeline : Write a pipeline script for cloning GitHub repository, Maven Build, Docker image build, push docker image to docker hub and Deployment through Ansible. After writing the pipeline script, build the Jenkins job. The job can be built manually or by using poll SCM.

Build Triggers

☐ Build after other projects are built ?
☐ Build periodically ?
☐ GitHub hook trigger for GITScm polling ?
☒ Poll SCM ?

Schedule ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H*****" to poll once per hour

Would last have run at Saturday, 16 April, 2022 at 12:44:07 AM India Standard Time; would next run at Saturday, 16 April, 2022 at 12:44:07 AM India Standard Time.

```

1 pipeline{
2   environment{
3     docker_image = ""
4     registryCredential='docker'
5   }
6   agent any
7   stages{
8     stage('Step 1: Git Clone'){
9       steps{
10         git branch: 'master', url: 'https://github.com/AkshayThite10/CalculatorDevops.git'
11       }
12     }
13     stage('Step 2: Maven Build'){
14       steps{
15         sh 'mvn clean compile assembly:single'
16       }
17     }
18     stage('Step 3: Build Docker Image'){
19       steps {
20         script {
21           docker_image = docker.build "akshaythite10/calculator_devops:latest"
22         }
23       }
24     }
25     stage('Step 4: Push docker image to hub') {
26       steps {
27         script {
28           docker.withRegistry('https://registry.hub.docker.com/', registryCredential) {
29             docker_image.push()
30           }
31         }
32       }
33     }
34     stage('Step 5: Ansible Deployment'){
35       steps{
36         ansiblePlaybook becomeUser: 'null',
37         colorized: true,
38         credentialsId: 'akshaygit',
39         installation: 'Ansible',
40         inventory: 'deployment/inventory',
41         playbook: 'deployment/deploy.yml',
42         sudoUser: 'null'
43       }
44     }
45   }
46 }

```

Continuous Deployment (Ansible) :

Ansible is an open source tool or platform. It is used for configuration management, application deployment, intraservice orchestration and provisioning

Steps :

1. Creating Ansible playbook : In the playbook we need to provide the correct python version and path of the docker image. This playbook is created in another directory inside our working directory.

```
! deploy.yml ×
deployment > ! deploy.yml
1 |---
2 | - name: Pull Docker image of Calculator
3 |   hosts: all
4 |   vars:
5 |     ansible_python_interpreter: /usr/bin/python3
6 |   tasks:
7 |     - name: Pull image
8 |       docker_image:
9 |         name: akshaythitel0/calculator_devops:latest
10 |        source: pull
```

```
≡ inventory ×
deployment > ≡ inventory
1 |localhost ansible_user=akshay
```

2. Configuring Ansible in Jenkins : Go to Jenkins -> Dashboard -> Manage Jenkins -> Global Tool Configuration. Add Ansible here by giving the correct path.

Ansible

Ansible installations

Add Ansible

Ansible

Name

Ansible

Path to ansible executables directory

/usr/bin/

3. Add Ansible stage in the Jenkins pipeline script.

```
34 stage('Step 5: Ansible Deployment'){
35     steps{
36         ansiblePlaybook becomeUser: 'null',
37         colorized: true,
38         credentialsId: 'akshaygit',
39         installation: 'Ansible',
40         inventory: 'deployment/inventory',
41         playbook: 'deployment/deploy.yml',
42         sudoUser: 'null'
43     }
```

Log Management (log4j) :

It is used for keeping track of operations performed in the application.

Steps :

1. Create a log4j file : Script is written in a file named log4j2.xml

```
log4j2.xml X
src > main > resources > log4j2.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Configuration status="INFO">
3      <Appenders>
4          <Console name="ConsoleAppender" target="SYSTEM_OUT">
5              <PatternLayout pattern="%d{dd/MMM/yyyy:HH:mm:ss SSS} [%F] [%level] %logger{36} %msg%n"/>
6          </Console>
7          <File name="FileAppender" fileName="calculator_devops.log" immediateFlush="false" append="true">
8              <PatternLayout pattern="%d{dd/MMM/yyyy:HH:mm:ss SSS} [%F] [%level] %logger{36} - %msg%n"/>
9          </File>
10     </Appenders>
11     <Loggers>
12         <Root level="debug">
13             <!-- <AppenderRef ref="ConsoleAppender"/>-->
14             <AppenderRef ref="FileAppender"/>
15         </Root>
16     </Loggers>
17 </Configuration>
18
```

2. Adding logger functions : Add these to the calculator program.

```
private static final Logger logger = LogManager.getLogger(App.class);
logger.info("Message"); // For generating the logs.
```

Building Jenkins Pipeline :

Stage View



Now, the docker image can be pulled and run inside the local machine to generate the logs which act as the input for the ELK Stack.

```
akshay@akshay-Vostro-3578:~$ docker images
REPOSITORY              TAG              IMAGE ID         CREATED          SIZE
akshaythite10/calculator_devops   latest          31428e23e78c    52 minutes ago  528MB
registry.hub.docker.com/akshaythite10/calculator_devops   latest          31428e23e78c    52 minutes ago  528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          b9d6e1f23b2a    32 hours ago    528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          1ded0602f868    32 hours ago    528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          e0f10756c4a1    32 hours ago    528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          ac9aaac7b88b    33 hours ago    528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          c2a138db85ec    37 hours ago    528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          fceedeac6c96    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          4cbf06df29b6    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          ab294932b825    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          8632a0d8c5b7    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          18c63ff486e2    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          f29c9fc4efc2    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          178eded678ba    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          52a6a0ee417e    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          c6e8233522b5    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          20d11e81ea2c    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          a5539cacf28a    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          b7c1beddbd08    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          11479b9c8742    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          633d07acf8ff    5 days ago      528MB
registry.hub.docker.com/akshaythite10/calculator_devops   <none>          42d0a65293c9    5 days ago      526MB
nodejs_app               latest          a87f31171f38    10 days ago     89.7MB
akshay/docker            latest          e3411ee836c7    10 days ago     124MB
openjdk                  8              57fdaf8c395f    10 days ago     124MB
ubuntu                   latest          18fbe41f975e    2 weeks ago     526MB
ubuntu                   latest          2b4cba85892a    6 weeks ago     72.8MB
ubuntu                   <none>          54c9d81cbb44    2 months ago    72.8MB
node                     10-alpine      aa67ba258e18    12 months ago   82.7MB
```

Command : docker images

It displays all the docker images.

Running the docker image inside a docker container.

```
akshay@akshay-Vostro-3578:~$ docker run -it akshaythite10/calculator_devops
.....Calculator Program.....
Operation choices :
1. Square root
2. Factorial
3. Natural log
4. Power
5. Exit

Enter your choice :
1
Square root
Enter a number :
64

Square root = 8.0

Operation choices :
1. Square root
2. Factorial
3. Natural log
4. Power
5. Exit

Enter your choice :
2
Factorial
Enter a number
6

Factorial = 720
```

```
Operation choices :
1. Square root
2. Factorial
3. Natural log
4. Power
5. Exit

Enter your choice :
3
Natural log
Enter a number :
1024

Natural log = 6.931471805599453

Operation choices :
1. Square root
2. Factorial
3. Natural log
4. Power
5. Exit

Enter your choice :
4
Power
Enter a number :
2
Enter exponent :
10

Power = 1024.0

Operation choices :
1. Square root
2. Factorial
3. Natural log
4. Power
5. Exit

Enter your choice :
5
akshay@akshay-Vostro-3578:~$
```

```

akshay@akshay-Vostro-3578:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
27bfb478d41a   akshaythite10/calculator_devops    "java -cp Calculator..." 6 minutes ago  Exited (0)   5 minutes ago           pensive_goodall
8af08e961571   1ded0602f868                       "java -cp Calculator..." 32 hours ago  Exited (255) 9 hours ago           flamboyant_maxwell
31befcb35e8b   e0f10756c4a1                       "java -cp Calculator..." 32 hours ago  Exited (255) 9 hours ago           dreamy_haibt
fca9b11aec5b   c2a138db85ec                       "java -cp Calculator..." 37 hours ago  Exited (255) 9 hours ago           romantic_khorana
577a4d401370   fceedeac6e96                       "java -cp Calculator..." 5 days ago    Exited (0)   5 days ago             elated_ardinghelli
fcc2ceca074f   fceedeac6e96                       "java -cp Calculator..." 5 days ago    Exited (0)   5 days ago             competent_yalow
9cf1ac7e87bd   4cbf06df29b6                       "java -cp Calculator..." 5 days ago    Exited (1)   5 days ago             practical_dewdney
2f82dbdfb1e5   8632a0d8c5b7                       "java -cp Calculator..." 5 days ago    Exited (1)   5 days ago             flamboyant_keldysh
9ed8362fc993   8632a0d8c5b7                       "java -cp Calculator..." 5 days ago    Exited (1)   5 days ago             optimistic_lovelace
6df7e3dbfc23   nodejs_app:latest                  "docker-entrypoint.s..." 10 days ago   Exited (0)   5 days ago             nodejs_app
a6830ace77f9   akshay/docker                      "./run.sh arg1"           10 days ago   Exited (0)   10 days ago            container1
a5e466dc35b   ubuntu                             "/bin/bash"               6 weeks ago   Exited (0)   6 weeks ago            Jenkins_Agent
d65e87d4d838   54c9d81cbb44                       "bash"                    6 weeks ago   Exited (0)   6 weeks ago            user1

```

Command : `sudo docker ps -a`

It displays all the details of the docker containers.

```

akshay@akshay-Vostro-3578:~$ docker start 27
27
akshay@akshay-Vostro-3578:~$ docker exec -it 27 "/bin/bash"
root@27bfb478d41a:/# ls
calculatorDevops-1.0-SNAPSHOT-jar-with-dependencies.jar  boot      dev  home  lib64  mnt  proc  run  srv  tmp  var
bin                                                       calculator_devops.log  etc  lib   media  opt  root  sbin  sys  usr
root@27bfb478d41a:/# cat calculator_devops.log
15/Apr/2022:20:04:38 239 [App.java] [INFO] App - Square root computed
15/Apr/2022:20:04:45 449 [App.java] [INFO] App - Factorial computed
15/Apr/2022:20:04:55 785 [App.java] [INFO] App - Natural log computed
15/Apr/2022:20:05:05 709 [App.java] [INFO] App - Power computed

```

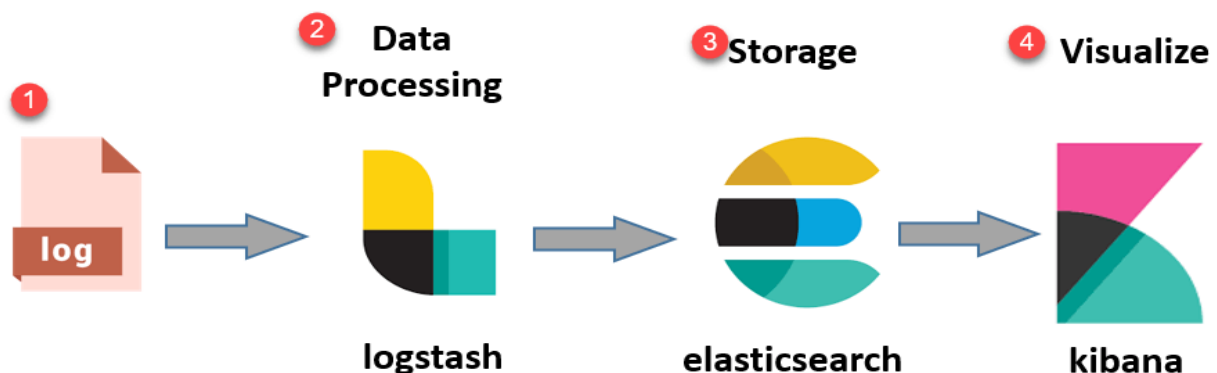
Commands : `docker start <container_id>`

`docker exec -it <container_id> "/bin/bash"`

These commands are used to start the docker container and then check for the log file.

Continuous Monitoring (ELK) :

ELK stands for Elasticsearch, Logstash and Kibana. It aggregates logs from systems and applications, analyzes these logs and creates visualization for application and infrastructure monitoring, faster troubleshooting and security analytics.



Steps :

1. Upload the log file using the following override settings

Override settings

Number of lines to sample

1000

Data format

semi_structured_text

Grok pattern

```
%{HTTPDATE:timestamp_string}  
\[%{GREEDYDATA:thread}\] \[%{LOGLEVEL:level}\]  
%{GREEDYDATA:logger} \- %{GREEDYDATA:msg}
```

Timestamp format


dd/MMM/yyyy:HH:mm:ss XX

[See more on accepted formats](#)

Time field

Close

Apply



Integrations

Upload file

More ways to add data

In addition to adding [Integrations](#), you can try our sample data or upload your own data.

Sample data

Upload file

calculator_devops.log

Import data

Simple

Advanced

Index name

☒ Create data view

Data view name

Combined fields

➕ Add combined field

Index settings

```
1 {
2   "number_of_shards": 1
3 }
```

Mappings

```
1 {
2   "properties": {
3     "@timestamp": {
4       "type": "date"
5     },
6     "level": {
7       "type": "keyword"
8     },
9     "logger": {
10      "type": "keyword"
11    },
12    "message": {
13      "type": "text"
14    },
15    "msg": {
16      "type": "keyword"
17    },
18    "thread": {
19      "type": "keyword"
20    },
21    "timestamp_string": {
22      "type": "keyword"
23    }
24  }
25 }
```

Ingest pipeline

```
1 {
2   "description": "Ingest pipeline created by text structure finder",
3   "processors": [
4     {
5       "grok": {
6         "field": "message",
7         "patterns": [
8           "%{HTTPDATE:timestamp} \\[%{GREYDATA:thread}\\] \\[%{LOGLEVEL:level}\\] \\[%{GREYDATA:logger}\\] - \\[%{GREYDATA:msg}\\]"
9         ]
10      },
11    ],
12    {
13      "date": {
14        "field": "timestamp",
15        "formats": [
16          "dd/MMM/yyyy:HH:mm:ss SSS"
17        ]
18      }
19    ]
20  }
```

Reset



File processed



Index created



Ingest pipeline created



Data uploaded



Data view created

✓ Import complete

Index	calc
Data view	calc
Ingest pipeline	calc-pipeline
Documents ingested	5



View index in Discover



Index Management

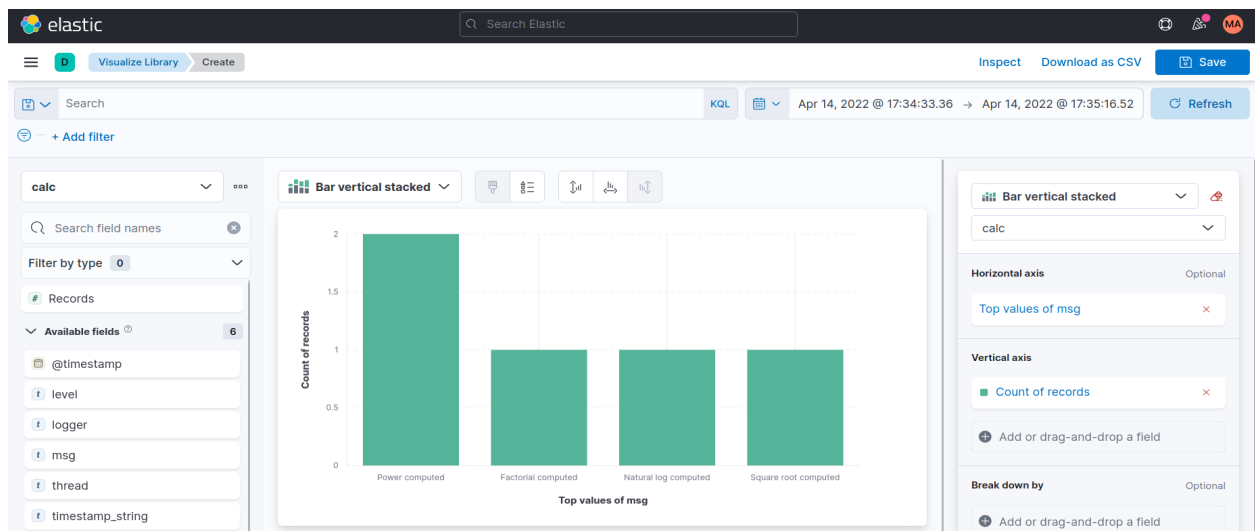


Data View Management



Create Filebeat configuration

2. Data Visualization



Errors Faced while Execution :

1. In the third stage of pipeline - Building Docker Image

Console Output

```
+ docker build -t akshaythite10/calculator_devops:latest .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post
"http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&
cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&
t=akshaythite10%2Fcalculator_devops%3Alatest&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission
denied
```

Solution : Modified permissions of /var/run/docker.sock

Command : sudo chmod 777 /var/run/docker.sock

2. In last stage of pipeline - Ansible Deployment

```
[DevOps_Calculator] $ sshpass ***** /usr/bin/ansible-playbook deployment/deploy.yml -i deployment/inventory -u AkshayThite10 -k

PLAY [Pull Docker image of Calculator] *****

TASK [Gathering Facts] *****
[0;32m[mok: [localhost][0m
[0;32m[0m
TASK [Pull image] *****
[0;31mfatal: [localhost]: FAILED! => {"changed": false, "msg": "Failed to import the required Python library (Docker SDK for Python:
docker (Python >= 2.7) or docker-py (Python 2.6)) on akshay-Vostro-3578's Python /usr/bin/python3. Please read module documentation
and install in the appropriate location. If the required library is installed, but Ansible is using the wrong Python interpreter,
please consult the documentation on ansible_python_interpreter, for example via `pip install docker` or `pip install docker-py`
(Python 2.6). The error was: No module named 'docker'"}[0m
[0;31m[0m
PLAY RECAP *****
[0;31mlocalhost[0m          : [0;32mok=1    [0m changed=0    unreachable=0    [0;31mfailed=1    [0m skipped=0    rescued=0
ignored=0

FATAL: command execution failed
hudson.AbortException: Ansible playbook execution failed
    at org.jenkinsci.plugins.ansible.AnsiblePlaybookBuilder.perform(AnsiblePlaybookBuilder.java:262)
    at org.jenkinsci.plugins.ansible.workflow.AnsiblePlaybookStep$AnsiblePlaybookExecution.run(AnsiblePlaybookStep.java:430)
    at org.jenkinsci.plugins.ansible.workflow.AnsiblePlaybookStep$AnsiblePlaybookExecution.run(AnsiblePlaybookStep.java:351)
    at
    org.jenkinsci.plugins.workflow.steps.AbstractSynchronousNonBlockingStepExecution$1$1.call(AbstractSynchronousNonBlockingStepExecution
.java:47)
        at hudson.security.ACL.impersonate2(ACL.java:449)
        at hudson.security.ACL.impersonate(ACL.java:461)
        at
    org.jenkinsci.plugins.workflow.steps.AbstractSynchronousNonBlockingStepExecution$1.run(AbstractSynchronousNonBlockingStepExecution.ja
va:44)
        at java.base/java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:515)
        at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)
        at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128)
        at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
        at java.base/java.lang.Thread.run(Thread.java:829)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: Ansible playbook execution failed
Finished: FAILURE
```

Solution : Installed a python library.

Command : pip install docker-py