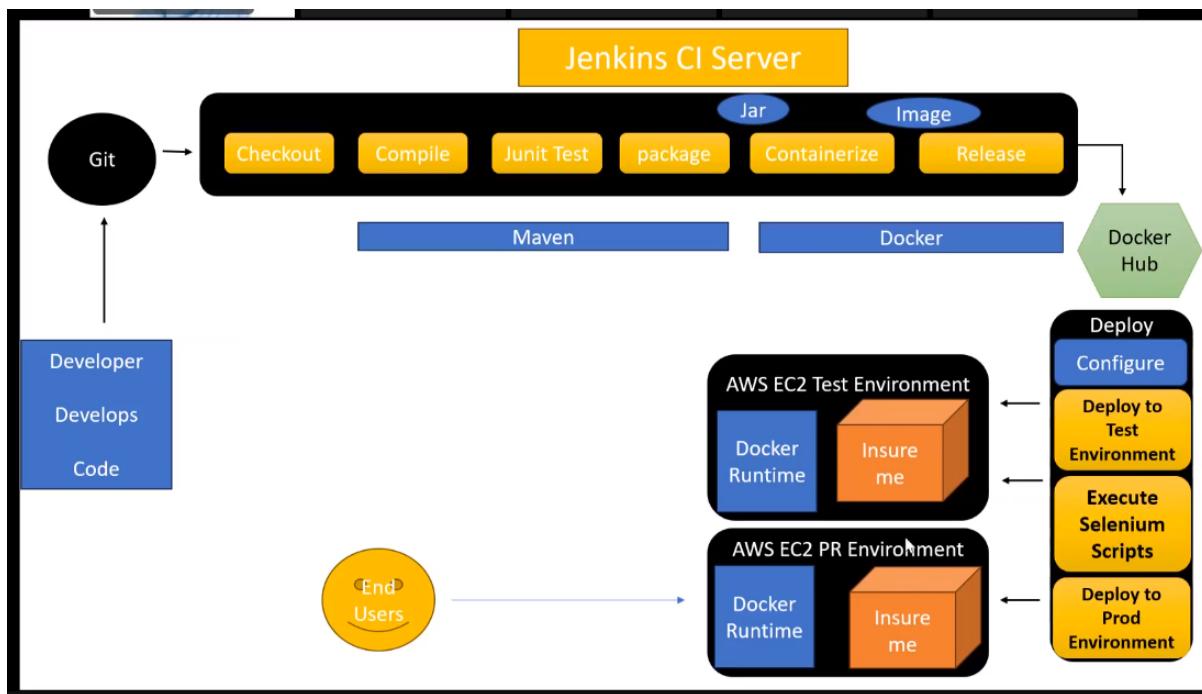


Capstone-Project:



- 1) Create a Jenkins-server on AWS and install git, maven, jenkins, java, ansible, docker to it.

Cloned repository to Kruti2202/Project-caps repository as below

<https://github.com/Kruti2202/Project-Caps>

```

Elastic Kubernetes Service IAM CloudFormation EC2 VPC
ansible 2.10.8
  config file = None
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0]
ubuntu@ip-172-31-13-242:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2023-08-04 00:05:41 UTC; 4min 8s ago
    Main PID: 7052 (java)
       Tasks: 36 (limit: 2349)
      Memory: 376.6M
        CPU: 46.846s
       CGroup: /system.slice/jenkins.service
              └─ 7052 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --htt

Aug 04 00:05:05 ip-172-31-13-242 jenkins[7052]: 4e71c388f6524e2e9de4ea62eb78b6bc
Aug 04 00:05:05 ip-172-31-13-242 jenkins[7052]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 04 00:05:05 ip-172-31-13-242 jenkins[7052]: ****
Aug 04 00:05:05 ip-172-31-13-242 jenkins[7052]: ****
Aug 04 00:05:05 ip-172-31-13-242 jenkins[7052]: ****
Aug 04 00:05:41 ip-172-31-13-242 jenkins[7052]: 2023-08-04 00:05:41.059+0000 [id=28] INFO jenkins.InitReactorRunner$...
Aug 04 00:05:41 ip-172-31-13-242 jenkins[7052]: 2023-08-04 00:05:41.082+0000 [id=22] INFO hudson.lifecycle.Lifecycle$...
Aug 04 00:05:41 ip-172-31-13-242 systemd[1]: Started Jenkins Continuous Integration Server.
Aug 04 00:05:41 ip-172-31-13-242 jenkins[7052]: 2023-08-04 00:05:41.500+0000 [id=44] INFO h.m.DownloadService$Downlo...
Aug 04 00:05:41 ip-172-31-13-242 jenkins[7052]: 2023-08-04 00:05:41.501+0000 [id=44] INFO hudson.util.Retrive#start:...
lines 1-20/20 (END)

i-0f23ae72dc23b4d0f (Jenkins-server)
PublicIPs: 54.67.124.102 PrivateIPs: 172.31.13.242
  
```

- Set up jenkins, tools and plugins (maven, ansible)

- 2) Code checkout and code build - jar file on jenkins and local machine.

On jenkins:
Create a CI pipeline:

```
node{  
    stage('code checkout'){  
        echo 'Checking out the application code'  
        git branch: 'main', url: 'https://github.com/Kruti2202/Capstone-Project.git'  
    }  
    stage('code build'){  
        echo 'Building application... compile.. test.. package..'  
        sh 'mvn clean package'  
    }  
}
```

The screenshot shows the Jenkins Pipeline interface for the 'insure-me-ci-pipeline'. The pipeline is defined with two stages: 'code checkout' and 'code build'. The 'code checkout' stage took 4 seconds and the 'code build' stage took 38 seconds. The pipeline has run once on Aug 03 at 17:17 with no changes. There are links for 'Build History', 'Atom feed for all', and 'Atom feed for failures'.

On local machine:
\$mvn clean package

```
aws [Alt+S] Services Search [Alt+S] N. California Kruti Patel  
AWS CloudFormation EC2 VPC  
Elastic Kubernetes Service IAM CloudFormation EC2 VPC  
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-analysis/8.0/asm-analysis-8.0.jar (33 KB at 28 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/guava/failureaccess/1.0.1/failureaccess-1.0.1.jar  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-android/guava-28.2-android.jar  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/guava/failureaccess/1.0.1/failureaccess-1.0.1.jar (4.6 kB at 3.0 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/guava/listenablefuture/9999.0-empty-to-avoid-conflict-with-guava/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-util/8.0/asm-util-8.0.jar (85 kB at 69 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/listenablefuture/9999.0-empty-to-avoid-conflict-with-guava/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar (2.2 kB at 1.8 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/com/google/errorprone/error_prone_annotations/2.3.4/error_prone_annotations-2.3.4.jar  
[INFO] Replacing main artifact with repackaged archive  
[INFO] BUILD SUCCESS  
[INFO] Total time: 42.812 s  
[INFO] Finished at: 2023-08-04T09:22:37Z  
[INFO] PublicIP: 54.67.124.102 PrivateIP: 172.31.15.242  
Ubuntu@p-172-31-13-242:/project-caps/Project-Caps$
```

3) Containerize jar file to image, hence we create docker image as below

Dockerfile

```
FROM openjdk:11
ARG JAR FILE-TARGET/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","app.jar"]
```

On local machine:

```
docker build -t kruti2202/insure-me:1.0 .
```

```
docker images
```

```
db38d58ec8ab: Pull complete
Digest: sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9cdb7c21ab
Status: Downloaded newer image for openjdk:11
--> 47a932d998b7
Step 2/4 : ARG JARFILE=target/*.jar
--> Running in 1daab36f8496
Removing intermediate container 1daab36f8496
--> f3d6a59ef2ed
Step 3/4 : COPY ${JARFILE} app.jar
--> 77f0aec1e18a
Step 4/4 : ENTRYPOINT ["java","-jar","app.jar"]
--> Running in 60b6a09eebe8
Removing intermediate container 60b6a09eebe8
--> c2c2e2155787
Successfully built c2c2e2155787
Successfully tagged kruti2202/insure-me:1.0
root@ip-172-31-13-242:/home/ubuntu/project-caps/Project-Caps# docker images
docker: 'mages' is not a docker command.
See 'docker --help'.
root@ip-172-31-13-242:/home/ubuntu/project-caps/Project-Caps# docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
kruti2202/insure-me  1.0      c2c2e2155787  8 seconds ago  695MB
openjdk              11       47a932d998b7  12 months ago  654MB
root@ip-172-31-13-242:/home/ubuntu/project-caps/Project-Caps#
```

i-Of23ae72dc23b4d0f (Jenkins-server)

PublicIPs: 54.67.124.102 PrivateIPs: 172.31.13.242

docker run -itd -p 8081:8081 kruti2202/insure-me:1.0

[Jenkin-server-public-P:8081/](#)



On jenkins:

*note: Add jenkins user to docker group in order it to run docker command:

- `$sudo usermod -a -G docker jenkins && newgrp docker`
For local machine:
- `$sudo usermod -a -G docker $USER && newgrp docker or sudo su`
- `$sudo chown <username>:<username> /var/run/docker.sock`

Pipeline Syntax for containerization:

```
stage('containerizer application') {
    echo 'Creating docker image for insure-me'
    sh 'docker build -t kruti2202/insure-me:4.0 .'
}
```

The screenshot shows the Jenkins Pipeline interface for the project "Pipeline insure-me-ci-pipeline". On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. The main area is titled "Stage View" and shows a grid of three stages: "code checkout", "code build", and "containerizer application". Each stage has a progress bar indicating its duration: 543ms for checkout, 22s for build, and 3s for application. Below the stages, a summary shows average stage times: 538ms for checkout, 22s for build, and 3s for application. The overall average full run time is approximately 25s. To the left of the stages, a "Build History" section lists two builds: #2 (Aug 4, 2023, 2:40 AM) and #1 (Aug 4, 2023, 2:39 AM). At the bottom, there are links for "Atom feed for all" and "Atom feed for failures".

4) After successfully building an image, release image to docker hub:

On local machine:

`$docker login`

`$docker push kruti2202/insure-me:4.0`

*note: for local machine credentials can be entered but for jenkins we need to add credentials prior into the syntax.

On Jenkins:

Set up the credentials on jenkins for docker hub and run the below pipeline

```
stage('push image to dockerhub'){
    echo 'Pushing insure-me image to docker hub'
    withCredentials([string(credentialsId: 'dockerpwd', variable: 'dockerHubPwd')]) {
        // some block
    }
}
```

```

sh "docker login -u kruti2202 -p ${dockerHubPwd}"
sh 'docker push kruti2202/insure-me:4.0'
}
}

```

The screenshot shows the Jenkins Stage View for the 'insure-me-ci-pipeline'. On the left, there's a sidebar with options like 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. Below that is a 'Build History' section listing three builds: #3 (Aug 4, 2023, 2:52 AM), #2 (Aug 4, 2023, 2:40 AM), and #1 (Aug 4, 2023, 2:39 AM). The main area is titled 'Stage View' and displays four stages: 'code checkout', 'code build', 'containerizer application', and 'push image to dockerhub'. Each stage has a bar indicating its duration: code checkout (543ms), code build (23s), containerizer application (3s), and push image to dockerhub (10s). An average stage time of 545ms is shown.

Image on Docker-hub:

The screenshot shows the Docker Hub repository page for 'kruti2202/insure-me'. At the top, there's a field to add a short description and a 'Public View' button. Below that is a 'Description' section which says 'This repository does not have a description'. It also shows the last push was 4 minutes ago. To the right is a 'Docker commands' section with a 'docker push kruti2202/insure-me:tagname' button. The main area is divided into 'Tags' and 'Automated Builds'. The 'Tags' section lists three tags: 4.0, 2.0, and 3.0, each with a pull and push timestamp. The 'Automated Builds' section explains how to automatically push images to Docker Hub using GitHub or Bitbucket. There's also an 'Upgrade' button.

5) configure ansible to deploy the application on test-server as below:

On local machine:(Jenkins server)

Inventory file: .etc/ansible/hosts:

```
ubuntu@ip-172-31-9-199:/etc/ansible$ cat hosts
[test-server]
54.176.147.34 ansible_ssh_private_key_file=/home/ubuntu/Capstone-Project.pem ansible_user=ubuntu
ubuntu@ip-172-31-9-199:/etc/ansible$
```

i-04a2ab4bc33dabb7a (Jenkin-server)

PublicIPs: 18.144.87.0 PrivateIPs: 172.31.9.199

\$ansible-playbook configure-test-server.yml

```
ubuntu@ip-172-31-9-199:/etc/ansible$ ansible-playbook /home/ubuntu/project/project-caps/configure-test-server.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [test-server] ****
TASK [Gathering Facts] ****
ok: [54.176.147.34]

TASK [update apt] ****
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather than running sudo
changed: [54.176.147.34]

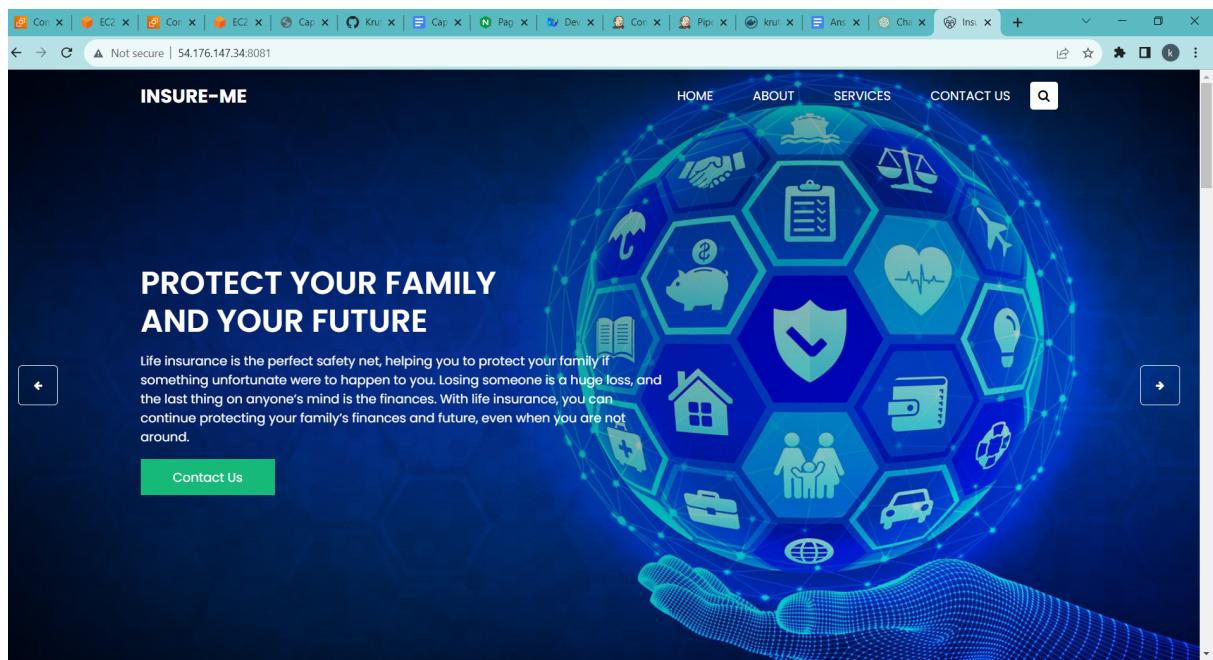
TASK [install docker] ****
changed: [54.176.147.34]

TASK [start docker service] ****
changed: [54.176.147.34]

TASK [deploy insureme application] ****
changed: [54.176.147.34]

PLAY RECAP ****
54.176.147.34 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

test-server-public-IP:8081/



On jenkins:

Build new pipeline : configuration and Deployment

Add ansible plugins :

Hosts (inventory file), SSH session private key, playbook

```
ubuntu@ip-172-31-9-199:/project/Project-Caps$ cd /etc/ansible
ubuntu@ip-172-31-9-199:/etc/ansible$ cat hosts
[test-server]
54.151.121.192

[prod-server]
18.144.17.68
ubuntu@ip-172-31-9-199:/etc/ansible$
```

i-04a2ab4bc33dabb7a (Jenkin-server)

PublicIPs: 54.193.203.46 PrivateIPs: 172.31.9.199

Pipeline:

```
node{
    stage('checkout playbooks'){
        git 'https://github.com/Kruti2202/Project-Caps.git'
    }
    stage('execute Ansible playbooks and deploy'){
        ansiblePlaybook become: true, credentialsId: 'ssh-key-ansible', disableHostKeyChecking: true, installation: 'Ansible', inventory: '/etc/ansible/hosts', playbook: 'test-server-ansible.yml'
    }
}
```

```
Last login: Fri Aug 4 22:23:07 2023 from 54.193.203.46
ubuntu@ip-172-31-23-191:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
758e595e7b7a krutি2202/insure-me:4.0 "java -jar app.jar" 2 hours ago Up 2 hours 0.0.0.0:8081->8081/tcp, :::8081->8081/tcp insure-me
ubuntu@ip-172-31-23-191:~$
```

i-0fdc135875a374c1 (Test-server)

PublicIPs: 54.151.121.192 PrivateIPs: 172.31.23.191

The screenshot shows the Jenkins dashboard with the following details:

- Top Bar:** Not secure | 18.144.87.0:8080
- User Information:** Kruti Patel (logged in)
- Dashboard Links:** New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views.
- Build Queue:** No builds in the queue.
- Build Executor Status:** No data shown.
- Pipeline List:**

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Configure and Deployment pipeline	9 min 41 sec #1	N/A	29 sec
✓	☀️	insure-me-ci-pipeline	2 hr 36 min #3	N/A	38 sec
✓	☁️	test	2 min 48 sec #2	6 min 56 sec #1	30 sec
- Icon Legend:** S (Stable), M (Medium), L (Large).
- Atom Feeds:** Atom feed for all, Atom feed for failures, Atom feed for just latest builds.

- [Executing Selenium script in test environment and deploying it over prod environment:](#)

- 1) Ubuntu AWS machine setup : chromedriver, chrome browser and permissions setup done.
- 2) Jenkins server is enabled to run Selenium script by adding below permission to the server.
- 3) Selenium script tested on local machine successfully.

Selenium script :

```
package com.insure_me.project.insure_me_project;

import java.io.File;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
import org.apache.commons.io.FileUtils;

public class App
{
    public static void main( String[] args ) throws
InterruptedException, IOException
    {
        System.setProperty("webdriver.chrome.driver",
"C:/Users/kruti/Desktop/DevOps/Software/chromedriver.exe");
    }
}
```

```
//System.setProperty("webdriver.chrome.driver",
"/usr/bin/chromedriver");

ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addArguments("--headless");
chromeOptions.addArguments("--no-sandbox");
chromeOptions.addArguments("--disable-dev-shm-usage");
chromeOptions.addArguments("--remote-allow-origins=*");
WebDriver driver = new ChromeDriver(chromeOptions);

driver.get("http://54.151.121.192:8081/contact.html");

//driver.get("http://172.31.90.46:8081/contact.html");

driver.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);

driver.findElement(By.id("inputName")).sendKeys("Kruti
Patel");

driver.findElement(By.id("inputNumber")).sendKeys("9999999999");

driver.findElement(By.id("inputMail")).sendKeys("kruti@gmail.com"
);

driver.findElement(By.id("inputMessage")).sendKeys("Welcome      to
the DevOps");

driver.findElement(By.id("my-button")).click();

String      message      =
driver.findElement(By.id("response")).getText();
if(message.equals("Message Sent")) {
    System.out.println("Script executed Successfully");
}
else {
    System.out.println("Script failed");
}

System.out.println("Taking Screenshot as proof");
//take the screenshot of the testcase
```

```

TakesScreenshot scrShot = ((TakesScreenshot)driver);

File screenShot = scrShot.getScreenshotAs(OutputType.FILE);

File destFile = new File("screenshot.png");

 FileUtils.copyFile(screenShot, destFile);

System.out.println("reports stored at : " + destFile.getAbsolutePath().toString());

Thread.sleep(3000);

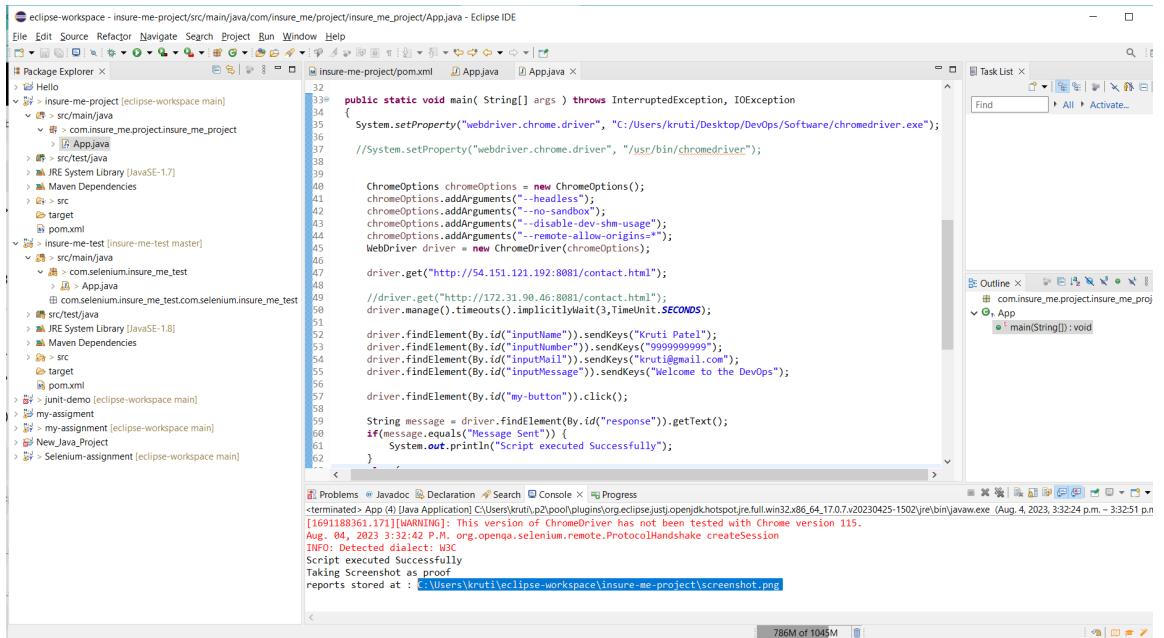
driver.quit();

}

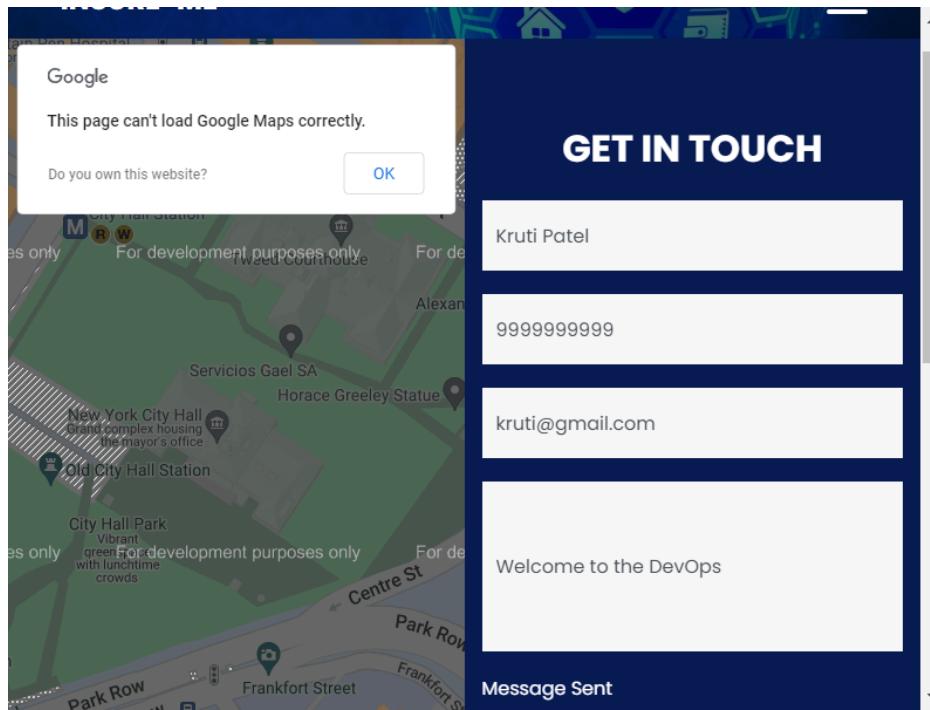
}

```

Successfully ran on local machine using eclipse IDE as below :



screenshot.png



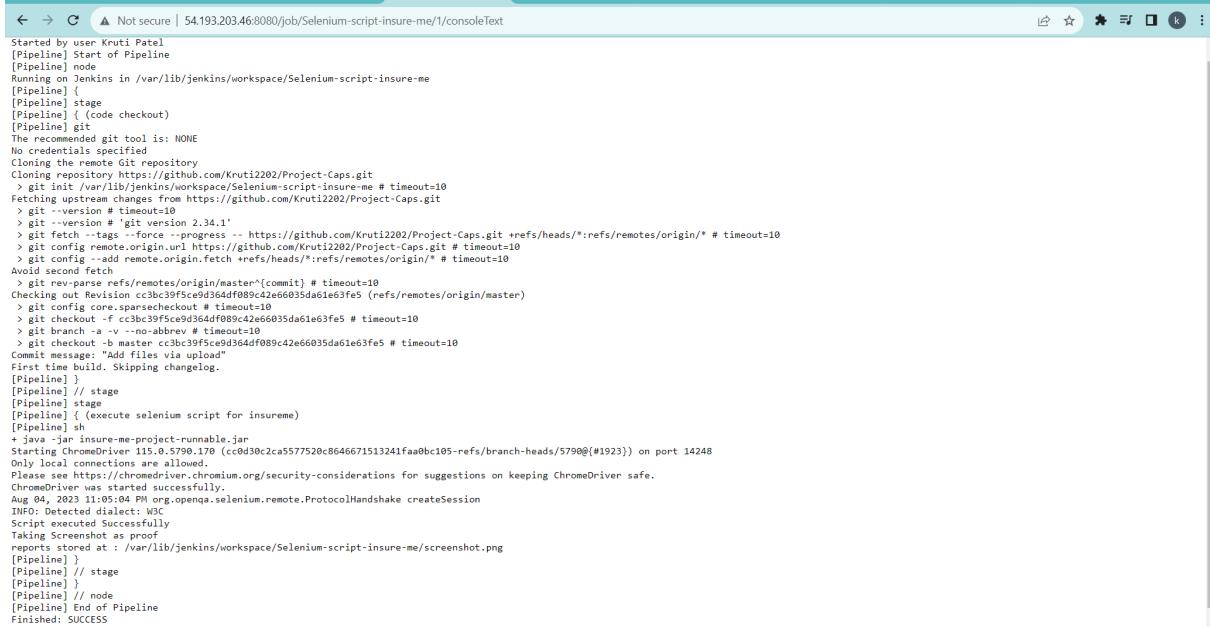
Selenium-script pipeline on jenkins:

```
node{  
    stage('code checkout') {  
        git 'https://github.com/Kruti2202/Project-Caps.git'  
    }  
    stage('execute selenium script for insureme') {  
        sh 'java -jar insure-me-project-runnable.jar'  
    }  
}
```

Build Now:

The screenshot shows the Jenkins Pipeline Stage View for the "Selenium-script-insure-me" pipeline. The pipeline consists of two stages: "code checkout" and "execute selenium script for insureme". The "code checkout" stage took 3s and was run on Aug 4, 16:04. The "execute selenium script for insureme" stage took 11s and was run on Aug 4, 16:04. There were no changes. The pipeline is currently in the "Status" view.

Console Output:



```
Started by user Kruti Patel
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Selenium-script-insure-me
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Code checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Kruti2202/Project-Caps.git
> git init /var/lib/jenkins/workspace/Selenium-script-insure-me # timeout=10
Fetching upstream changes from https://github.com/Kruti2202/Project-Caps.git
> git config remote.origin.url https://github.com/Kruti2202/Project-Caps.git # timeout=10
> git config remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoided selecting a credential
> git --no-parse refs/remotes/origin/master{commit} # timeout=10
Checking out Revision cc3b39f5cecd3d64df089c42e66035da61e63fe5 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f cc3b39f5ce9d364df089c42e66035da61e63fe5 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master cc3b39f5ce9d364df089c42e66035da61e63fe5 # timeout=10
Commit message: "Add files via upload"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (execute selenium script for insureme)
[Pipeline] sh
[jenkins-project] Starting ChromeDriver 115.0.5775.178 (cc630c2ca5577520c8646671513241faa@bc105-refs/branch-heads/5790@(#1923)) on port 14248
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Aug 04, 2023 11:05:04 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Script executed successfully
Taking screenshot after script
reports stored at : /var/lib/jenkins/workspace/Selenium-script-insure-me/screenshot.png
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- Deployment on production server:

Inventory file :/etc/ansible/hosts

```
command: sudo docker run -td -p 8001:8001 shubhamksrivani123/INSUREME
ubuntu@ip-172-31-9-199:~/project/Project-Caps$ cd /etc/ansible
ubuntu@ip-172-31-9-199:/etc/ansible$ cat hosts
[test-server]
54.151.121.192

[prod-server]
18.144.17.68
ubuntu@ip-172-31-9-199:/etc/ansible$
```

i-04a2ab4bc33dabb7a (Jenkin-server)

PublicIPs: 54.193.203.46 PrivateIPs: 172.31.9.199

Configure-prod-server.yml

```
---
- hosts: prod-server
  become: true
  tasks:
    - name: update apt
      command: sudo apt-get update

    - name: install docker
      command: sudo apt install -y docker.io
```

```

- name: start docker service
  command: sudo systemctl start docker

- name: deploy insureme application
  command: sudo docker run -itd -p 8081:8081
shubhamkushwah123/insure-me:1.0

```

Configure Pipeline Syntax:

```

node{
  stage('code checkout'){
    git 'https://github.com/Kruti2202/Project-Caps.git'
  }
  stage('execute selenium script for insureme'){
    sh 'java -jar insure-me-project-runnable.jar'
  }
  stage('checkout ansible script'){
    git 'https://github.com/Kruti2202/Project-Caps.git'
  }
  stage('Execute deploy to prod'){
    ansiblePlaybook become: true, credentialsId: 'ssh-key-test',
    disableHostKeyChecking: true, installation: 'Ansible', inventory:
    '/etc/ansible/hosts', playbook: 'configure-prod-server.yml'
  }
}

```

On Prod Server

```

ubuntu@ip-172-31-26-234:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
da988ac8f0d7        shubhamkushwah123/insure-me:1.0   "java -jar /app.jar"   3 minutes ago      Up 3 minutes       0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   reverent_chatterjee
ubuntu@ip-172-31-234:~$ i-0a70f9cfdd7bad0bf (Prod-server)
PublicIPs: 18.144.17.68  PrivateIPs: 172.31.26.234

```

Build Now on Jenkins:

Not secure | 54.193.203.46:8080/job/Selenium-script-insure-me/

Jenkins

Dashboard > Selenium-script-insure-me >

Pipeline Selenium-script-insure-me

- Status
- </> Changes
- ▷ Build Now
- ⚙ Configure
- Delete Pipeline
- Full Stage View
- Rename
- Pipeline Syntax

Add description

Disable Project

Stage View

	code checkout	execute selenium script for insureme	checkout ansible script	Execute deploy to prod
#3 Aug 04 16:52 No Changes	557ms	8s	486ms	56s
#2 Aug 04 16:48 No Changes	906ms	7s	517ms	758ms failed
#1 Aug 04 16:04 No Changes	3s	11s		

Average stage times: (Average full run time: ~40s)

Build History trend / Filter builds... #3 Aug 04 16:52 No Changes #2 Aug 04 16:48 No Changes #1 Aug 04 16:04 No Changes

Prod-server-public-IP:8081/ (<http://18.144.17.68:8081/>)

Not secure | 18.144.17.68:8081

INSURE-ME

HOME ABOUT SERVICES CONTACT US

PROTECT YOUR FAMILY AND YOUR FUTURE

Life insurance is the perfect safety net, helping you to protect your family if something unfortunate were to happen to you. Losing someone is a huge loss, and the last thing on anyone's mind is the finances. With life insurance, you can continue protecting your family's finances and future, even when you are not around.

Contact Us