

# DevOps Project – Insure Me Insurance Domain

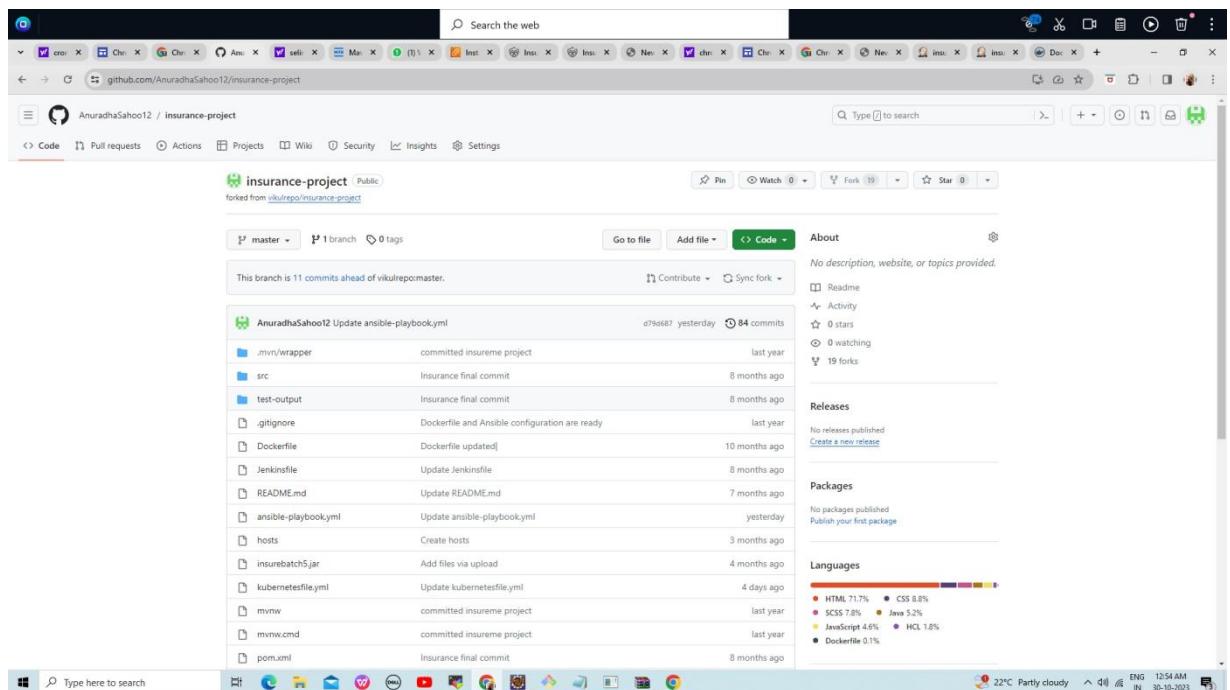
Name: Anuradha Sahoo

Date Of Submission: 30 oct 2023

## For this project I used following: -

- ✓ Git - For version control for tracking changes in the code files
- ✓ Jenkins - For continuous integration and continuous deployment
- ✓ Docker - For deploying containerized applications
- ✓ Ansible - Configuration management tools
- ✓ Selenium - For automating tests on the deployed web application
- ✓ ✓ AWS: For creating ec2 machines as servers and deploy the web application.

I have used git for version controlling:-



I have used Jenkins - For continuous integration and continuous deployment

Configured Jenkins Master and Jenkins Slave node. Created 2 EC2 instances. Jenkins Master and Jenkins Slave. Installed required software on Jenkins Master and Slave.

The screenshot shows the AWS EC2 Instances page. A success message at the top says "Successfully started i-08a8ce60152a4fb65". The main table lists three instances:

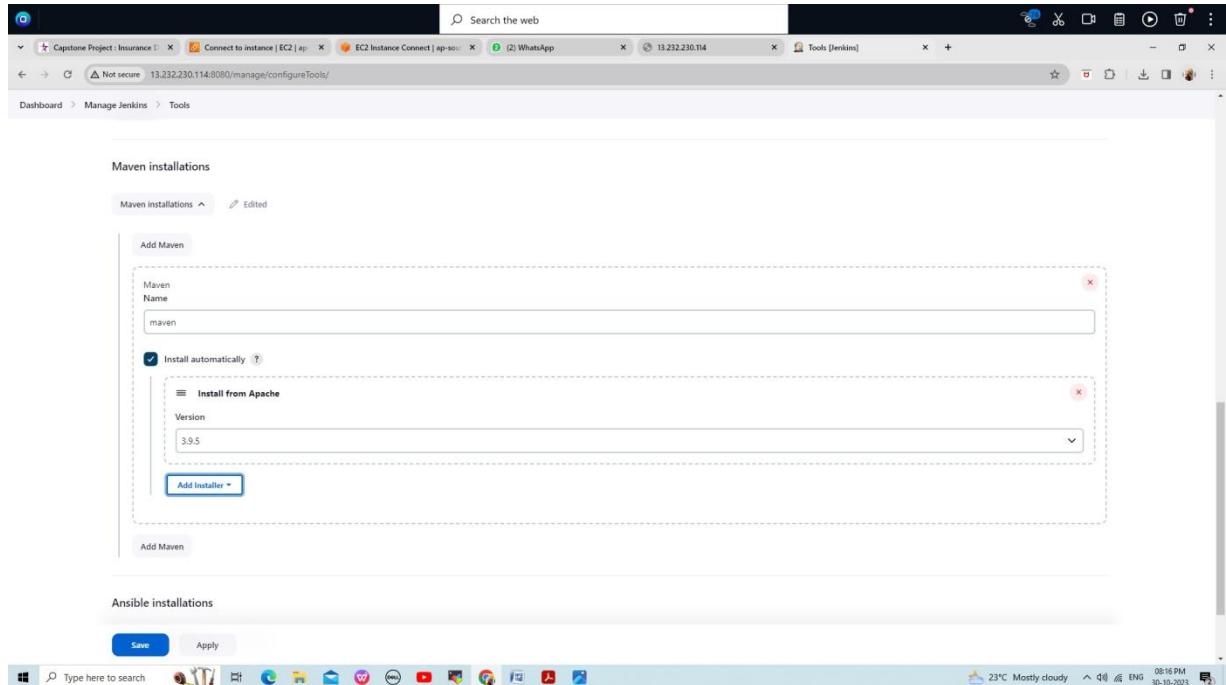
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
master_insurance	i-079d975bfe0637184	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-153-56.ap...	13.233.153.56	-
node_insurance	i-08a8ce60132a4fb65	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2-65-1-91-27.ap.sout...	65.1.91.27	-
test_server_in...	i-0fc4a2eeff78d0151	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-65-1-110-221.ap.s...	65.1.110.221	-

Installed Maven in master Jenkins

The screenshot shows the AWS CloudShell terminal window. The command "mvn --version" was run, and the output is displayed:

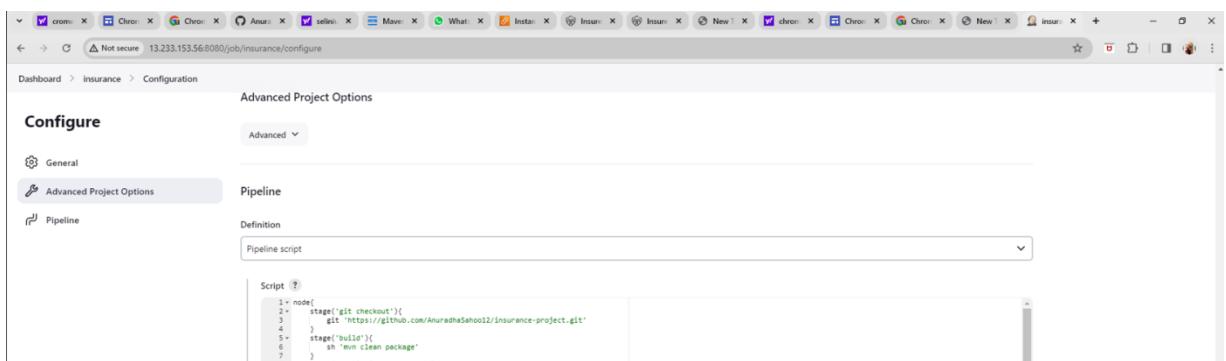
```
root@ip-172-31-1-109:~# mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.20.1, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-1040-aws", arch: "amd64", family: "unix"
```

Add the maven plugin in Jenkins Master and configure the maven tool under tools in Jenkins master to point the maven application path.

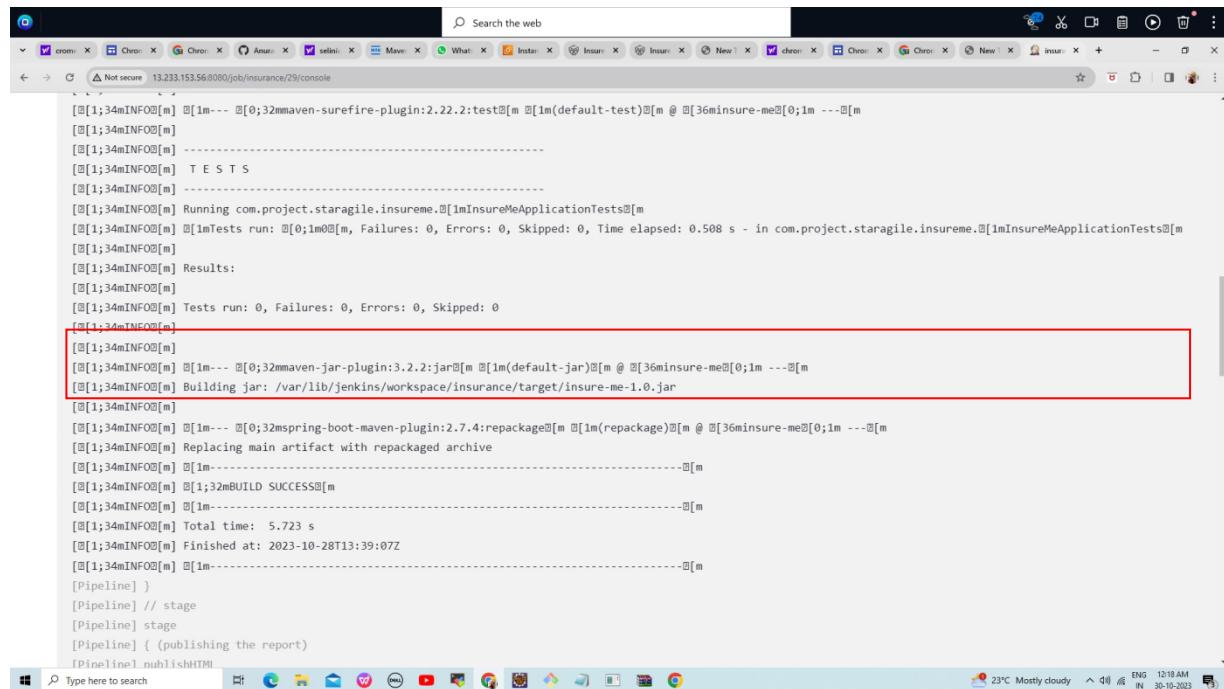


Wrote the Pipeline script to build the application.

1. Stage 1: SCM checkout where we get the code from GitHub repository.
2. Stage 2: Building package where we use maven clean and package to build the application artifacts.



Jar/Artifacts build successfully.



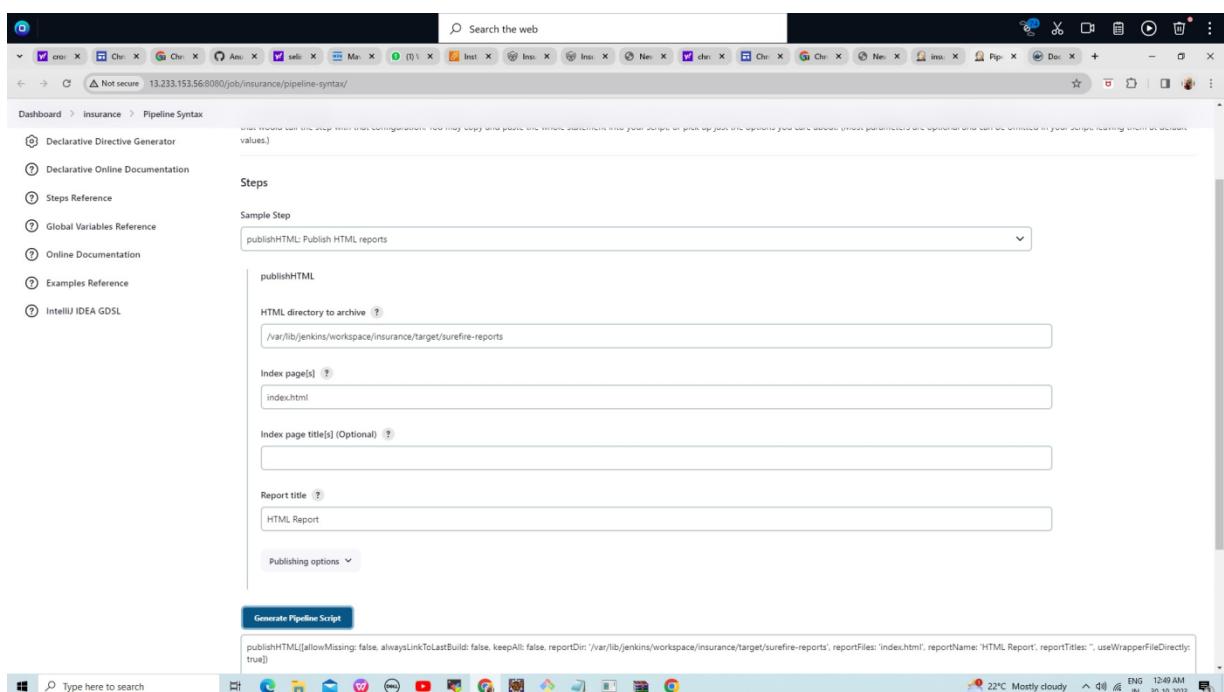
The screenshot shows a Jenkins job console window with multiple tabs open. The active tab displays the build log for job 'insurance/29/console'. The log output is as follows:

```
[0;34mINFO[m] [1m-- [0;32mmaven-surefire-plugin:2.22.2:test[m @ [1m(default-test)[m @ [36minsure-me[0;1m --[m
[0;1;34mINFO[m]
[0;1;34mINFO[m] -----
[0;1;34mINFO[m] T E S T S
[0;1;34mINFO[m] -----
[0;1;34mINFO[m] Running com.project.staragile.insureme.[1mInsureMeApplicationTests[m
[0;1;34mINFO[m] [1mtests run: [0;1m0[m, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.508 s - in com.project.staragile.insureme.[1mInsureMeApplicationTests[m
[0;1;34mINFO[m]
[0;1;34mINFO[m] Results:
[0;1;34mINFO[m]
[0;1;34mINFO[m] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[0;1;34mINFO[m]
[0;1;34mINFO[m] -----
[0;1;34mINFO[m] [1m-- [0;32mmaven-jar-plugin:3.2.2:jar[m @ [1m(default-jar)[m @ [36minsure-me[0;1m --[m
[0;1;34mINFO[m] Building jar: /var/lib/jenkins/workspace/insurance/target/insure-me-1.0.jar
[0;1;34mINFO[m]
[0;1;34mINFO[m] [1m-- [0;32mspring-boot-maven-plugin:2.7.4:repackage[m @ [1m(repackage)[m @ [36minsure-me[0;1m --[m
[0;1;34mINFO[m] Replacing main artifact with repackaged archive
[0;1;34mINFO[m] -----
[0;1;34mINFO[m] [1m-----[m
[0;1;34mINFO[m] [1;32mBUILD SUCCESS[m
[0;1;34mINFO[m] [1m-----[m
[0;1;34mINFO[m] Total time: 5.723 s
[0;1;34mINFO[m] Finished at: 2023-10-28T13:39:07Z
[0;1;34mINFO[m] [1m-----[m
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (publishing the report)
[Pipeline] publishHTML
```

A red box highlights the command `Building jar: /var/lib/jenkins/workspace/insurance/target/insure-me-1.0.jar` and the resulting jar file path.

### 3. Stage 3: Publish HTML reports.

Install HTML publisher. Add a stage publish HTML reports. Use syntax generator.



```

Script ?  

1< node{  

2>     stage('git checkout'){  

3>         git "https://github.com/AnuradhaSahoo12/insurance-project.git"  

4>     }  

5>     stage('build'){  

6>         sh 'mvn clean package'  

7>     }  

8>     stage('publishing the report'){  

9>         publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles  

10>        ])  

...

```

## Docker - For deploying containerized applications

Create Docker Hub account. Create docker hub account token. Add docker hub account details and token

### 4. Stage 4: Build docker image, login and publish image to Docker Hub.

T	P	Store	Domain	ID	Name
		System	(global)	dockerhub	dockerhub
		System	(global)	dockerhub1	dockerhub1
		System	(global)	ansible	jenkins
		System	(global)	ansible1	ubuntu
		System	(global)	ansible2	ubuntu1
		System	(global)	ansible3	ubuntu3

**Stores scoped to Jenkins**

P	Store	I	Domains
	System		(global)

Icon: S M L REST API Jenkins 2.414.3

Dashboard > insurance > Configuration

### Configure

General Advanced Project Options Pipeline

**Script**

```

1+ node{
2+   stages('git checkout'){
3+     git 'https://github.com/AnuradhaSahoo12/insurance-project.git'
4+   }
5+   stage('build'){
6+     sh 'mvn clean package'
7+   }
8+   stage('publishing the results'){
9+     publishHTML always: true, allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles:
10+    [
11+      'TEST-*.xml'
12+    ]
13+    stage('building the docker image'){
14+      sh ('docker build -t anuradhasahoo33412/insuranceproject:1.0 .')
15+    }
16+    step('Login/Logout push to dockerhub'){
17+      withCredentials([
18+        string(credentialsId: 'dockerhub1', variable: 'dockerpass')
19+      ]) {
20+        docker login -u anuradhasahoo33412 -p ${dockerpass}
21+        docker push anuradhasahoo33412/insuranceproject:1.0
22+      }
23+    }
24+  }
25+ }
26+

```

Use Groovy Sandbox ?

Pipeline Syntax

**Save** **Apply**

REST API Jenkins 2.414.3

Type here to search Search the web

master\_insurance Terminal Sessions View X server Tools Games Settings Macros Help Session Servers Tools Games Sessions View MultiTerm Tunneling Packages Settings Help

Quick connect... root@ip-172-31-1-109:~# docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
anuradhasahoo33412/insuranceproject	1.0	1ab9e48f120a	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	11fb3a3a11c1	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	93359a259e7e	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	7a046a79bd4d	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	50303a0a0a0a	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	fcc9df0fe5002	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	e8a7b5bea570	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	02202a0a0a0a	29 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	c51d11da9a02	30 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	a18e15ecbfef	30 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	1e90a0a0a0a0	30 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	e539c5411ea3	30 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	f3edf2ea9243	31 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	e0d724a05332	45 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	d008c026f980	45 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	42020a0a0a0a	45 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	f9129d4b0a07	45 hours ago	695MB
anuradhasahoo33412/insuranceproject	<none>	aa151025d1ca	46 hours ago	695MB
</none>	<none>	4110a0a0a0a0	47 hours ago	695MB
</none>	<none>	5b324d62f2ad	47 hours ago	695MB
</none>	<none>	ddc4d40808cb	47 hours ago	695MB
</none>	<none>	017d0a7e8005	2 days ago	695MB

root@ip-172-31-1-109:~#

Remote monitoring Follow terminal folder

UNREGISTERED VERSION - Please support MobaTerm by subscribing to the professional edition here: <https://mobaterm.mobatek.net>

Type here to search Search (CTRL+K)

Search the web

High winds today ENG 12:21 AM IN 30-10-2023

Not secure 13.233.153.56:8080

Jenkins anuradha sahoo log out

Dashboard >

+ New Item All +

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

The screenshot shows a Docker Hub repository page for 'anuradhasahoo333412/insuranceproject'. The repository has two tags: 1.0 and 2.0. The Docker commands section contains the command 'docker push anuradhasahoo333412/insuranceproject:tag'. The page also includes sections for Tags, Description, Docker commands, and Automated Builds.

## Ansible - Configuration management tools

Installed Ansible inside master Jenkins. And created another Ec2 instance as Test-server for deployment.

Added the public ip of test- server in /etc/Ansible/hosts in Ansible. check the connection is working by using the ping command from Ansible

```

# This is the default ansible 'hosts' file.
# It should live in /etc/ansible/hosts
#
# Comments begin with the '#' character
# Blank lines are ignored
# Groups of hosts are delimited by [header] elements
# You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrupped hosts, specify before any group headers:
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.1.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:
## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group:
## [dbservers]
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.50
## 10.25.1.51
## 10.25.1.52

Here's another example of host ranges, this time there are no
## www[001:006].example.com

## db [100.101]-node.example.com
## [node1]
## 10.25.1.50
## 10.25.1.51
## 10.25.1.52

```

The screenshot shows a terminal window titled 'master\_insurance' running on a Windows desktop. The terminal displays a command-line session where the user runs 'sudo su' and then 'ansible localhost -m ping'. The output shows a single host named 'localhost' with status 'SUCCESS' and a message 'pong'. A red box highlights this specific command and its output.

```

root@ip-172-31-1-109:~# sudo su
root@ip-172-31-1-109:~# cd
root@ip-172-31-1-109:~# su
root@ip-172-31-1-109:~# vi /etc/ansible/hosts
bashi: vi: /etc/ansible/hosts: No such file or directory
root@ip-172-31-1-109:~# vi /etc/ansible/hosts
root@ip-172-31-1-109:~# cd home/ubuntu
root@ip-172-31-1-109:/home/ubuntu# ansible localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
root@ip-172-31-1-109:/home/ubuntu#

```

Once the connection setup is done add the Ansible plugin in Jenkins Master. Go to tools and configure

### Ansible tool in Jenkins Master.

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. It displays two tabs: 'Maven installations' and 'Ansible installations'. The 'Ansible installations' tab is active, showing a configuration form. The 'Name' field contains 'ansible', and the 'Install automatically' checkbox is checked. At the bottom of the form are 'Save' and 'Apply' buttons. The Jenkins version 'Jenkins 2.414.3' is visible at the bottom right.

## Create the syntax pipeline for Ansible.

The screenshot shows a web-based configuration interface for an Ansible pipeline. The URL is 13.233.153.56:8080/job/insurance/pipeline-syntax/. The interface includes a search bar at the top and a sidebar with links to Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main form is titled "ansiblePlaybook: Invoke an ansible playbook". It contains fields for:

- Ansible tool:** ansible
- Playbook file path in workspace:** ansible-playbook.yml
- Inventory file path in workspace:** /etc/ansible/hosts
- SSH connection credentials:** ubuntu
- Vault credentials:** - none -
- Vault tmp path:** (empty field)
- Use become:**

The browser's address bar shows the URL 13.233.153.56:8080/job/insurance/pipeline-syntax/. The taskbar at the bottom shows various open tabs and icons.

This screenshot shows the same configuration interface as the first one, but with different settings. The URL is again 13.233.153.56:8080/job/insurance/pipeline-syntax/. The main form now includes fields for:

- Host subset:** root
- Tags:** (empty field)
- Tags to skip:** (empty field)
- Task to start at:** (empty field)
- Number of parallel processes to use:** (empty field)
- Disable the host SSH key check:**
- Colorized output:**
- Extra parameters:** (empty field)

A "Generate Pipeline Script" button is visible at the bottom. A preview of the generated script is shown below it:

```
ansiblePlaybook become: true, credentialsId: 'ansible1', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yml', vaultTmpPath: ''
```

The browser's address bar and taskbar are identical to the first screenshot.

Screenshot of a Jenkins pipeline configuration page. The pipeline script is defined as follows:

```

1+ node{
2+   stage('git checkout'){
3+     git 'https://github.com/AnuradhaSahoo12/insurance-project.git'
4+   }
5+   stage('build'){
6+     sh 'mvn clean package'
7+   }
8+   stage('publishing the report'){
9+     publishHTML(keepAll: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles:
10+       [
11+         'TEST-*.xml'
12+       ]
13+     )
14+   }
15+   stage('Login Login push to dockerhub'){
16+     withCredentials([
17+       string(credentialsId: 'dockerhub1', variable: 'dockerpass')
18+     ]) {
19+       docker login -u anuradhasahoo33412 -p ${dockerpass}
20+       docker push anuradhasahoo33412/insuranceproject1:0
21+     }
22+   }
23+   stage('deploy to the test server'){
24+     ansiblePlaybook become: true, credentialsId: 'ansible1', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yml'
25+   }
26+

```

The pipeline syntax is Groovy. There are two buttons at the bottom: 'Save' and 'Apply'.

## The ansible playbook file ansible-playbook.yml is: -

Screenshot of a GitHub repository showing the contents of the ansible-playbook.yml file. The file content is as follows:

```

- name : Configure Docker on EC2 Instances
  hosts : all
  become : true
  connection : ssh
  tasks :
    - name: updating apt
      command : sudo apt-get update
    - name : Install Docker
      command : sudo apt-get install -y docker.io
      become : yes
      become_user : root
    - name : Start Docker Service
      command : sudo systemctl start docker
      become : yes
      become_user : root
    - name: stop existing containers
      shell: docker stop $(docker ps -a -q)
    - name: delete all containers
      shell: docker rm $(docker ps -a -q)
    - name: Deploy Docker Container
      command: docker run -itd -p 8087:8081 anuradhasahoo33412/insuranceproject:1.0

```

I deployed my application to this test-server.

The screenshot shows the AWS EC2 Dashboard. A modal window at the top left says "Successfully stopped i-079d975bfe0637184". Below it, the "Instances (1/2) info" section shows one instance: "test\_server\_i... i-0fc4a2eef78d01511" is running, t2.micro, with 2/2 checks passed. The "Instance: i-0fc4a2eef78d01511 (test\_server\_insurance)" details pane is open, showing the instance summary, networking, and monitoring information. The instance has a Public IPv4 address of 65.1.110.221 and a Private IPv4 address of 172.31.12.131. It is connected to a VPC ID vpc-086a93d159cc9c6f4.

Project is deployed to test-server successfully

The screenshot shows the Jenkins Pipeline insurance stage view. The pipeline consists of six stages: git checkout, build, publishing the report, building the docker image, login, and deploy to the test server. The "git checkout" stage took 851ms. The "build" stage took 13s. The "publishing the report" stage took 52ms. The "building the docker image" stage took 3s. The "login" stage took 18s. The "deploy to the test server" stage took 2s. The total average stage time is 49s. The "Build History" table shows seven builds from Oct 28, 2023, with the last four failing. The Jenkins interface also includes a dashboard, changes, build now, configure, delete pipeline, full stage view, HTML report, rename, pipeline syntax, and a log out button.

Jenkins

Dashboard > insurance > #29

## Console Output

```
Started by user anuradha sahoo
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins In /var/lib/jenkins/workspace/insurance
[Pipeline] {
[Pipeline] stage
[Pipeline] { (git checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
+ git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/insurance/.git # timeout=10
Fetching changes from the remote Git repository
+ git config remote.origin.url https://github.com/AnuradhaSahoo12/insurance-project.git # timeout=10
Fetching upstream changes from https://github.com/AnuradhaSahoo12/insurance-project.git
+ git --version # timeout=10
+ git -version # git version 2.25.1'
+ git fetch --tags --force --progress -- https://github.com/AnuradhaSahoo12/insurance-project.git +refs/heads/*:refs/remotes/origin/* # timeout=10
+ git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision d79887c8ad#2990d15892663e59f9c07222cf75 (refs/remotes/origin/master)
+ git config core.sparsecheckout # timeout=10
+ git checkout -f d79887c8ad#2990d15892663e59f9c07222cf75 # timeout=10
+ git branch -a -v --no-abbrev # timeout=10
+ git branch -D master # timeout=10
+ git checkout -b master d79887c8ad#2990d15892663e59f9c07222cf75 # timeout=10
Commit message: "Update ansible-playbook.yml"
+ git rev-list --no-walk 853724dc4988124974cd6465bb0eb03c700cb0c3 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] sh
```

Dashboard > insurance > #29

```
[Pipeline] stage
[Pipeline] { (deploy to the test server)
[Pipeline] ansiblePlaybook
[Insurance] $ ansible-playbook ansible-playbook.yml -i /etc/ansible/hosts -b --become-user root --private-key /var/lib/jenkins/workspace/insurance/ssh134791078688982666291.key -u ubuntu
PLAY [Configure Docker on EC2 Instances] ****
TASK [Gathering Facts] ****
ok: [65.1.110.221]

TASK [Updating apt] ****
changed: [65.1.110.221]

TASK [Install Docker] ****
changed: [65.1.110.221]

TASK [Start Docker Service] ****
changed: [65.1.110.221]

TASK [Deploy Docker Container] ****
changed: [65.1.110.221]

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

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

Insurance project web page is:-

To run this page I used public ip of test-server:port no

INSURE-ME

HOME ABOUT SERVICES CONTACT US

PROTECT YOUR FAMILY AND YOUR FUTURE

Contact Us

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.

OUR SERVICES

We have brought you the world's best insurances. Choose one from the following...

**HOME INSURANCE**  
Home insurance also known as property insurance or homeowner insurance covers the structure as well as the content of the home  
[Read More](#)

**HEALTH INSURANCE**  
A medical insurance also called as health insurance, covers medical expenses for illnesses. It reimburses your medical bills.  
[Read More](#)

**CAR INSURANCE**  
Car insurance, is a type of vehicle insurance policy that protects you and your car from any risks of accidents and property damages.  
[Read More](#)

**LIFE INSURANCE**  
Life Insurance not only provides for financial support in the event of untimely death but also acts as a long term investment.  
[Read More](#)

[View All](#)

**ABOUT US**

Insure-Me is a leading multi-line insurer that serves its customers in global and local markets. With about 56,000 employees, it provides a wide range of property and casualty, life insurance products and services in more than 210 countries and territories. Zurich's customers include individuals, small businesses, and mid-sized and large companies, as well as multinational corporations.

[Read More](#)

EC2 | What's New | Install | GPG Key | Linux | insures | insures | Manager | insures | Pipeline | Pipeline | Docker | Settings | Google | 3.108.5 | 3.108.5 | Ins | +

Not secure 3.108.55.118:8088

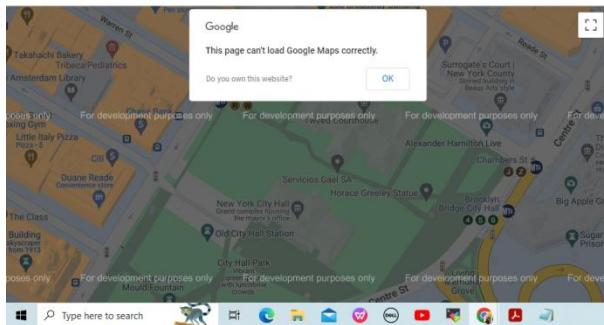
## BEST INSURANCE FOR YOUR FAMILY

There are many life insurance companies that provide life insurance plans globally. Life insurance is an agreement between an individual and the insurance company under which the insurance company promises to provide a death benefit to the family of life assured in case of an unfortunate demise of the life assured during the policy. We offer the best in class services.

[Read More](#)



Google  
This page can't load Google Maps correctly.  
Do you own this website? OK



23°C Partly cloudy 03/03 AM 28-10-2023

## GET IN TOUCH

Your Name

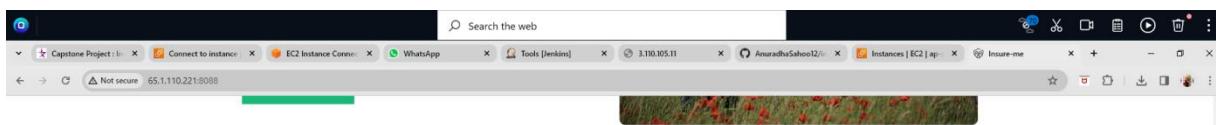
Mobile Number

Email

Message

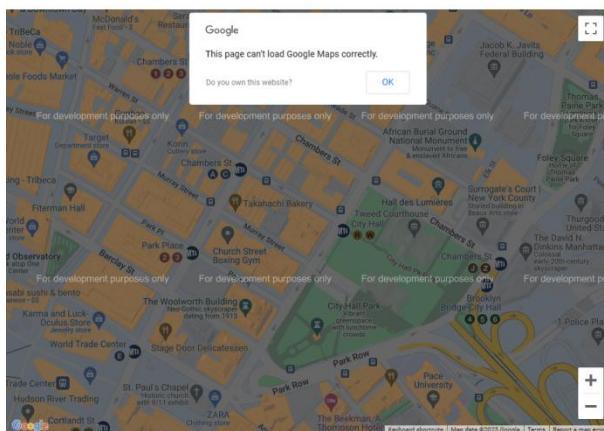
Search the web

Not secure 65.1.110.221:8088



23°C Partly cloudy 03/03 AM 28-10-2023

Google  
This page can't load Google Maps correctly.  
Do you own this website? OK



23°C Mostly cloudy 10:22 PM 30-10-2023

## GET IN TOUCH

Your Name

Mobile Number

Email

Message

[Send](#)



## Generate HTML Report using TestNG

The screenshot displays a TestNG HTML report titled "Surefire suite". The report structure is as follows:

- Info:** Contains the XML code for the test suite.
- Tests for Surefire suite:** Lists one test class: "Surefire test (1 class)".
- Groups for Surefire suite:** No groups listed.
- Times for Surefire suite:** Total running time: 0 ms.
- Reporter output for Surefire suite:** No output listed.
- 0 ignored methods:** No ignored methods listed.
- Methods in chronological order:** No methods listed.

## Selenium - For automating tests on the deployed web application

For this I imported this project on eclipse and added the dependencies on pom.xml file

```
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>seleniumhq.selenium</artifactId>
    <version>4.12.1</version>
</dependency>
```

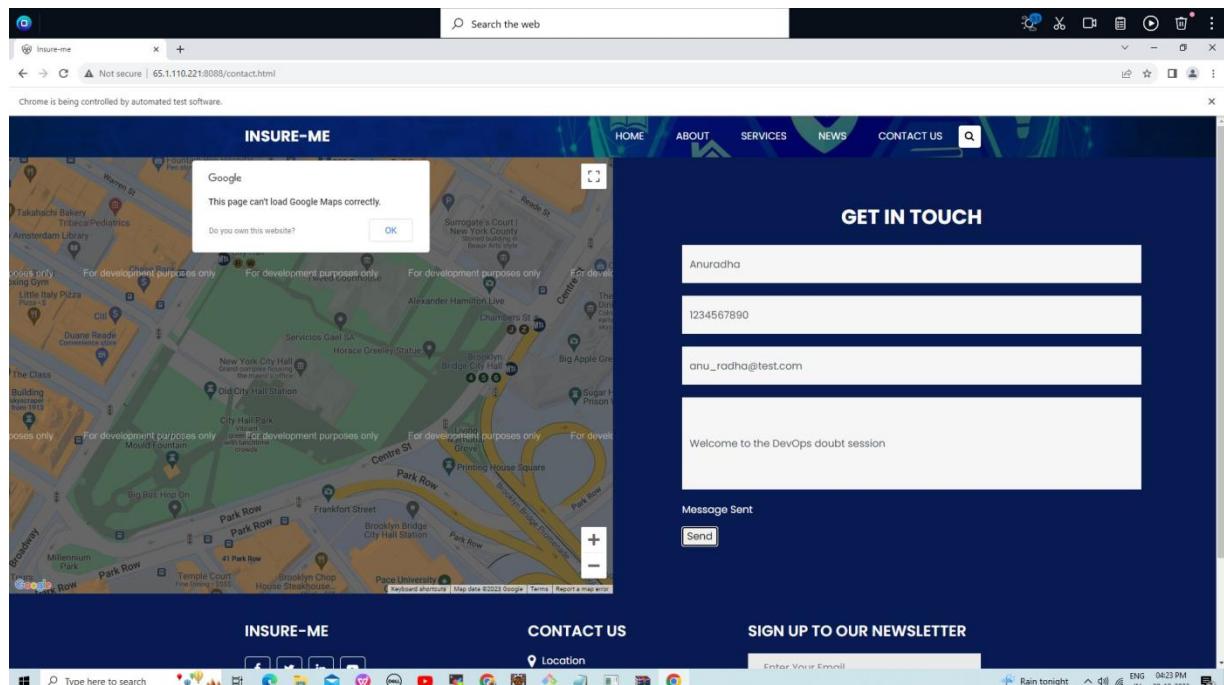
Created test.java application and added all the details to it

The screenshot shows the Eclipse IDE interface with the following details:

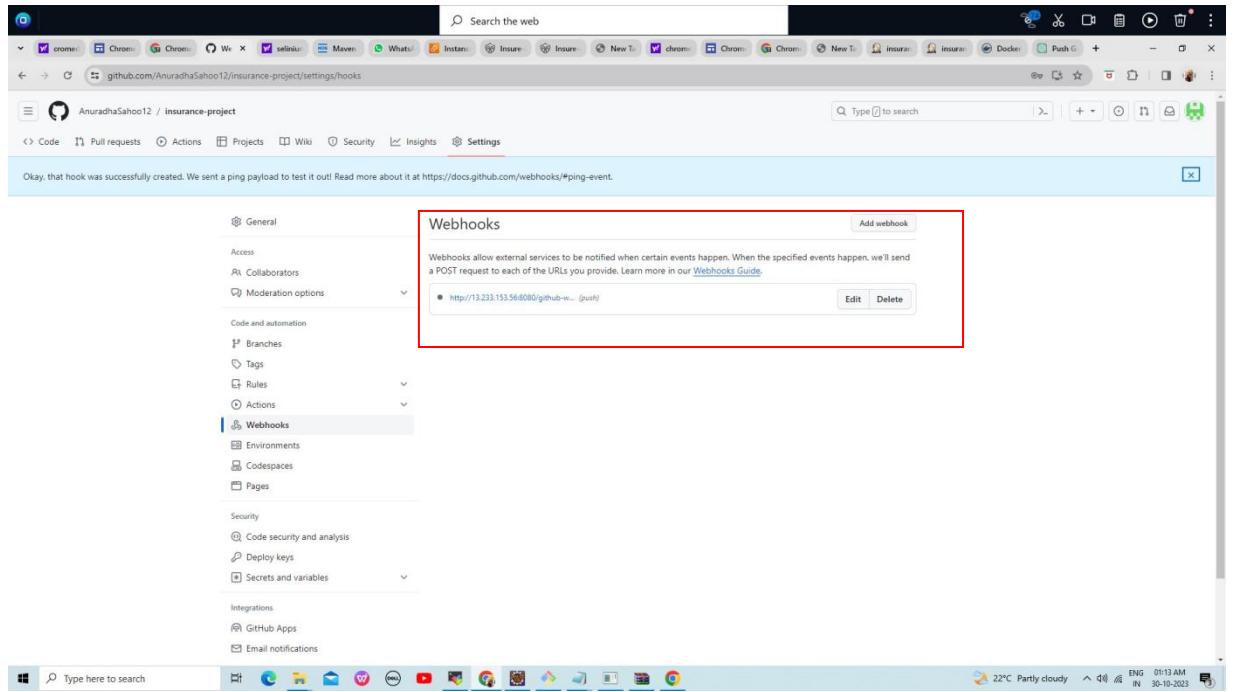
- Project Explorer:** Shows the project structure: star-agile-insurance-project, com.project.staragile.insureme, and test.java.
- Code Editor:** Displays the `test.java` file content:

```
1 package com.project.staragile.insureme;
2 import org.openqa.selenium.*;
3 import org.openqa.selenium.chrome.ChromeDriver;
4 import org.openqa.selenium.chrome.ChromeOptions;
5 import org.openqa.selenium.chrome.ChromeOptions;
6 import java.io.IOException;
7 import org.openqa.selenium.WebDriver;
8 public class test_se {
9
10    public static void main(String[] args) {
11        // TODO Auto-generated method stub
12        System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Downloads\\chromedriver-win64\\chromedriver.exe");
13        WebDriver driver = new ChromeDriver();
14        ChromeOptions chromeOptions = new ChromeOptions();
15        chromeOptions.addArguments("--remote-allow-origins=*", "ignore-certificate-errors");
16        chromeOptions.addArguments(new String[] { "--headless" });
17        chromeOptions.addArguments(new String[] { "--no-sandbox" });
18        chromeOptions.addArguments(new String[] { "--disable-dev-shm-usage" });
19        driver.get("http://65.1.10.221:8088/contact.html");
20        driver.manage().timeouts().implicitlyWait(5L, TimeUnit.SECONDS);
21        driver.findElement(By.id("inputName")).sendKeys(new CharSequence[]{ "Anurodha" });
22        driver.findElement(By.id("inputEmail")).sendKeys(new CharSequence[]{ "anu_radha@test.com" });
23        driver.findElement(By.id("inputMessage")).sendKeys(new CharSequence[]{ "Welcome to the DevOps doubt session" });
24        driver.findElement(By.id("buttonSubmit")).click();
25        final String message = driver.findElement(By.id("response")).getText();
26        if (message.equals("Message sent")) {
27            System.out.println("Script executed successfully");
28        } else {
29            System.out.println("Script failed");
30        }
31        Thread.sleep(3000L);
32        //driver.quit();
33    }
34
35
36
37 }
```

- Terminal:** Shows the command-line output of the Java application execution.
- System Tray:** Shows the date and time as 29-10-2023, 11:58 PM.



Created Git webhook and attached to the Jenkins master webhook for continuous integration and deployment.



The screenshot shows a Jenkins configuration page for a pipeline named 'insurance'. The 'Configure' tab is selected. Under 'General', there are several checkboxes: 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterized', and 'Throttle builds'. In the 'Build Triggers' section, there are several checkboxes: 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked and highlighted with a red box), 'Poll SCM', 'Quiet period', and 'Trigger builds remotely (e.g., from scripts)'. Below this, under 'Advanced Project Options', there is an 'Advanced' dropdown. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

Push your code into your GitHub Repository.

Push the code to GitHub

The screenshot shows the 'Push Branch master' dialog box. The 'Source' section shows 'master' and '7cbd710 first commit'. The 'Destination' section shows 'Remote: origin: https://github.com/AnuradhaSahoo12/star-agile-insurance-project.git' and 'Branch: master'. A checkbox 'Configure upstream for push and pull' is checked. Below it, a dropdown 'When pulling:' is set to 'Merge'. A checkbox 'Force overwrite branch in remote if it exists and has diverged' is unchecked. At the bottom right, there is a link 'Show advanced push dialog'. The dialog has standard buttons at the bottom: '?', '< Back' and 'Preview >', 'Push' (which is highlighted in blue), and 'Cancel'.

