Certification Project – Insure Me Insurance Domain

Submitted by Anupam Verma

Submitted to Vikul Kumar Sir

Date -30-10-2023

Insure Me is a Global leading Insurance provider based out of USA. The company offers products and services like Home Insurance, Health Insurance, Car Insurance and Life Insurances. Initially the company was using a Monolithic application architecture, As the company grown, It started facing difficulties in managing the application infrastructure and application deployments.

Insure-Me has decided to transform its monolithic application architecture to microservice application architecture and opted to go DevOps by implementing CICD pipeline and necessary automations. Insure me has decided to use AWS as primary cloud services provider to create servers, databases, and application deployments.

The company's goal is to deliver the product updates frequently to production with High quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers.

Following are the problems the company is facing at the moment

- ✓ Building Complex builds is difficult
- ✓ Manual efforts to test various components/modules of the project
- ✓ Incremental builds are difficult to manage, test and deploy
- ✓ Creation of infrastructure and configure it manually is very time consuming
- ✓ Continuous manual monitoring the application is quite challenging.

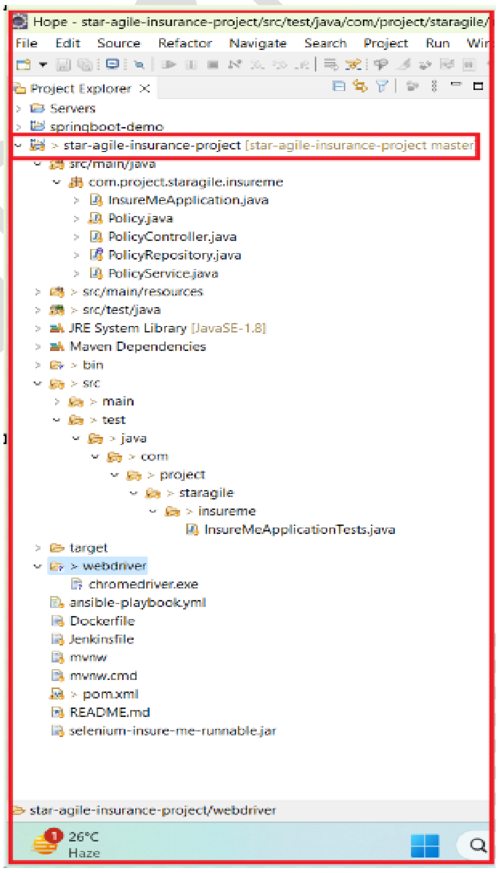
In order to implement a POC, you are requested to develop a mavenized microservice using spring boot and in memory h2 database.

1. a microservice which exposes below mentioned endpoints as APIs and uses in memory h2 database to store the data.

a. /createPolicy (HTTP Method : POST) (Request Body : JSON)
b. /updatePolicy/{policy id} (HTTP Method : PUT) (Request Body : JSON)
c. /viewPolicy/{policy id} (HTTP Method : GET) (No Request Body)
d. /deletePolicy/{policy id} (HTTP Method : DELETE) (No Request Body)

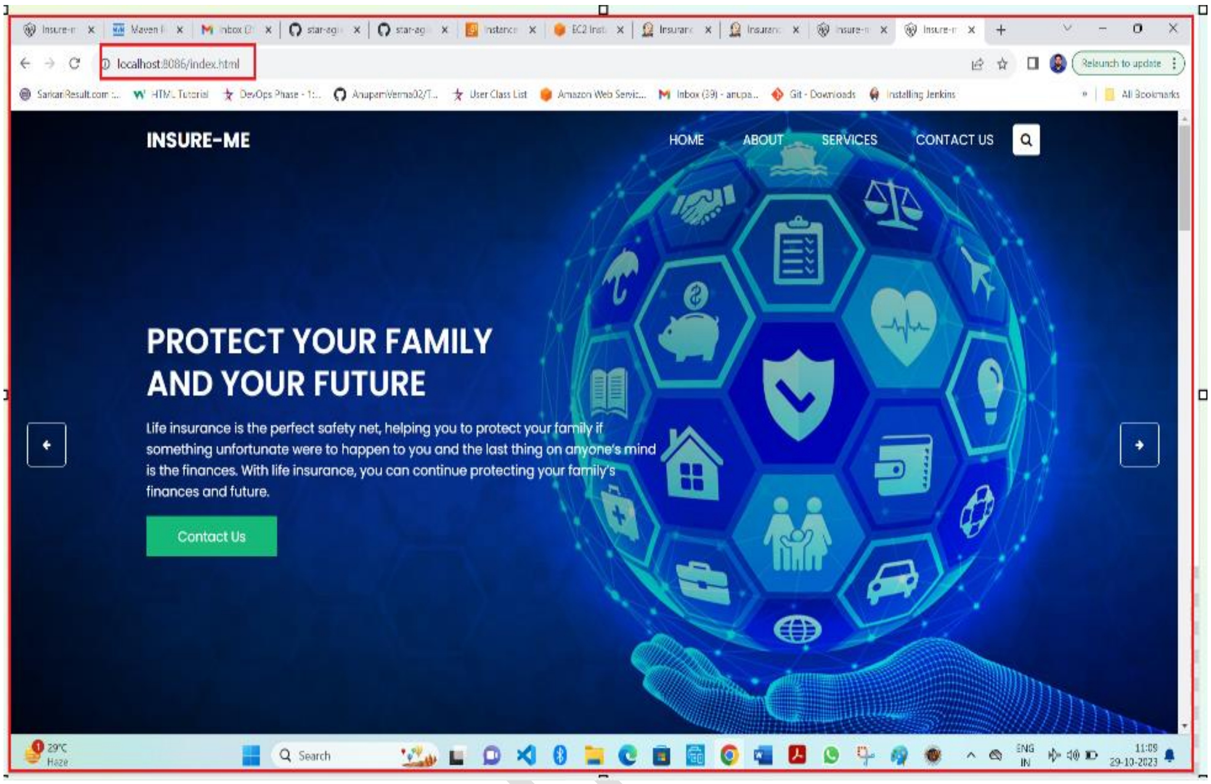
2. Write necessary Junit testcase.
3. Generate HTML report using TestNG.
4. Push your code into your GitHub Repository.

- Firstly I imported the git project into Eclipse—You can see my project is imported in Eclipse from Git.


Hope - star-agile-insurance-project/src/test/java/com/project/staragile/
File Edit Source Refactor Navigate Search Project Run Wi
Project Explorer

```
> Servers
> springboot-demo
v > star-agile-insurance-project [star-agile-insurance-project master
  v src/main/java
    v com.project.staragile.insureme
      > InsureMeApplication.java
      > Policy.java
      > PolicyController.java
      > PolicyRepository.java
      > PolicyService.java
  > > src/main/resources
  > > src/test/java
  > JRE System Library [JavaSE-1.8]
  > Maven Dependencies
  > > bin
  v > src
    > > main
    v > test
      v > java
        v > com
          v > project
            v > staragile
              v > insureme
                  InsureMeApplicationTests.java
  > target
  v > webdriver
      chromedriver.exe
    ansible-playbook.yml
    Dockerfile
    Jenkinsfile
    mvnw
    mvnw.cmd
    > pom.xml
    README.md
    selenium-insure-me-runnable.jar
```

After that I clean package to make the maven build.And then I ran my project as Java Application and you can see it below in screenshot.



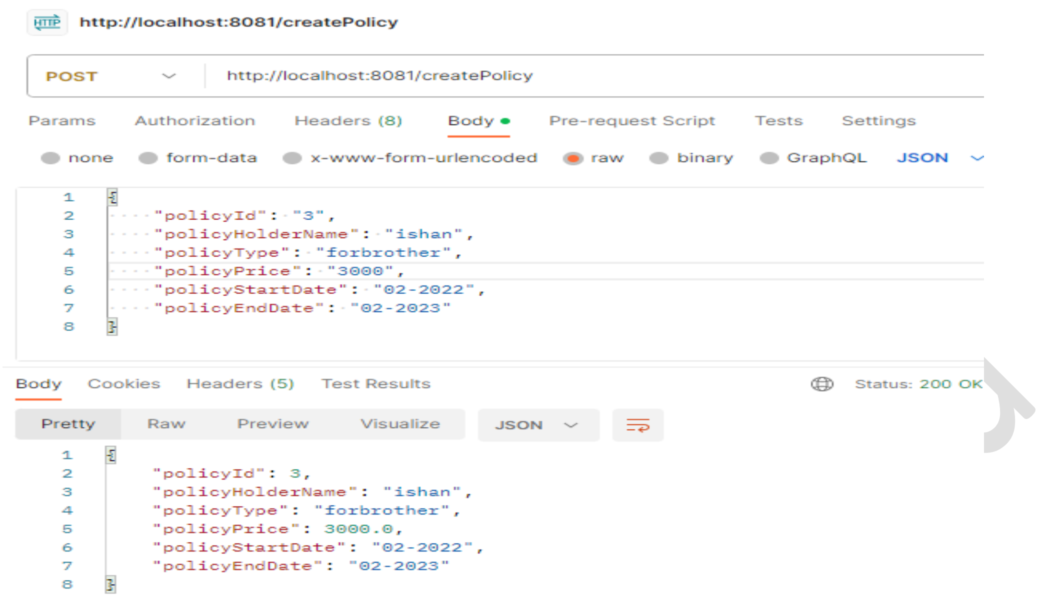
PROTECT YOUR FAMILY AND YOUR FUTURE

Life insurance is the perfect safety net, helping you to protect your family if something unfortunate were to happen to you and the last thing on anyone's mind is the finances. With life insurance, you can continue protecting your family's finances and future.

Then I checked the below services on postman.
a. /createPolicy (HTTP Method : POST) (Request Body : JSON)
b. /updatePolicy/{policy id} (HTTP Method : PUT) (Request Body : JSON)
c. /viewPolicy/{policy id} (HTTP Method : GET) (No Request Body)
d. /deletePolicy/{policy id} (HTTP Method : DELETE) (No Request Body)

There are few services which I checked on postman. First of all I downloaded Postman on my system . After configuring it I applied the above 4 services one by one. Here I am showing you that according to the developer’s Perspective how things done in the backend and how the API’s worked. Here I used the POST,PUT,GET and DELETE methods.

a. /createPolicy (HTTP Method : POST) (Request Body : JSON)



http://localhost:8081/createPolicy

POST http://localhost:8081/createPolicy

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

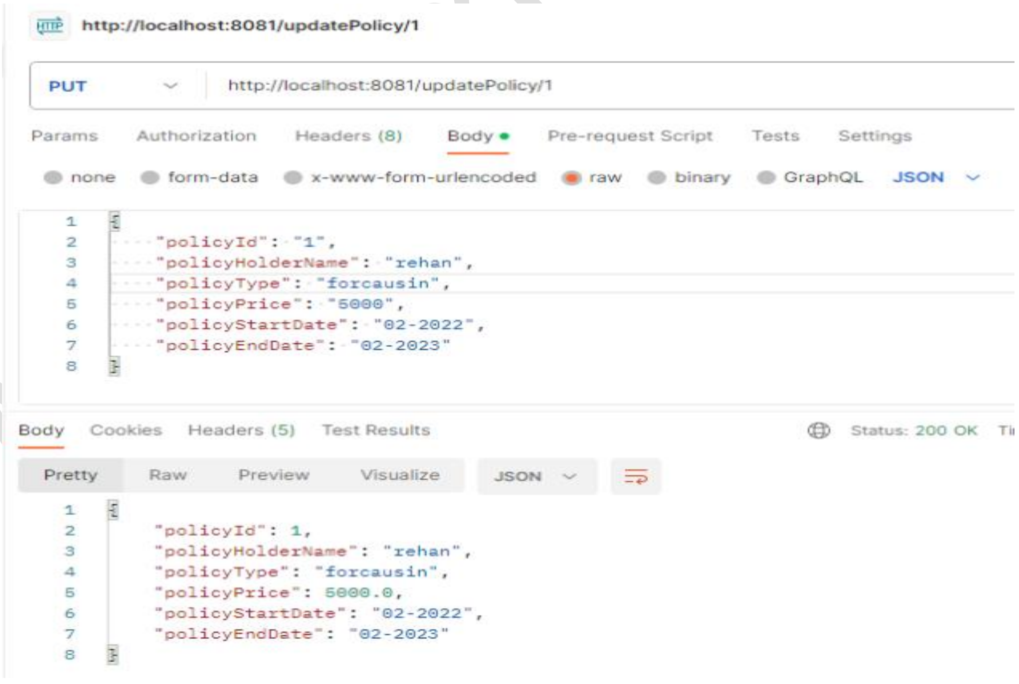
```
{
    "policyId": "3",
    "policyHolderName": "ishan",
    "policyType": "forbrother",
    "policyPrice": "3000",
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
{
    "policyId": 3,
    "policyHolderName": "ishan",
    "policyType": "forbrother",
    "policyPrice": 3000.0,
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

b. /updatePolicy/{policy id} (HTTP Method : PUT) (Request Body : JSON)



http://localhost:8081/updatePolicy/1

PUT http://localhost:8081/updatePolicy/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

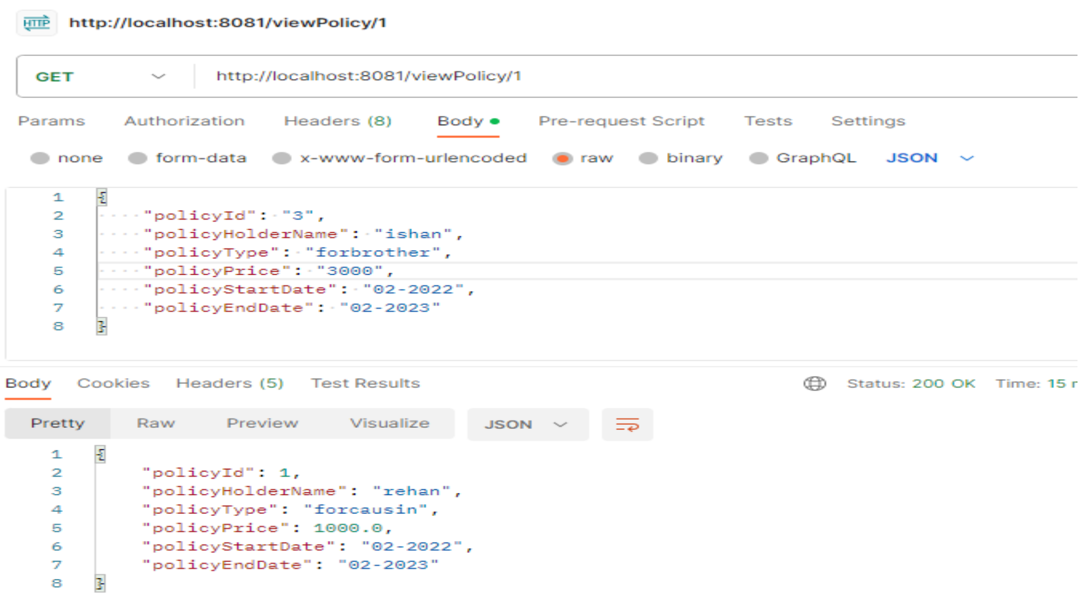
```
{
    "policyId": "1",
    "policyHolderName": "rehan",
    "policyType": "forcausin",
    "policyPrice": "5000",
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
{
    "policyId": 1,
    "policyHolderName": "rehan",
    "policyType": "forcausin",
    "policyPrice": 5000.0,
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

c. /viewPolicy/{policy id} (HTTP Method : GET) (No Request Body)



http://localhost:8081/viewPolicy/1

GET http://localhost:8081/viewPolicy/1

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded ● raw binary GraphQL JSON

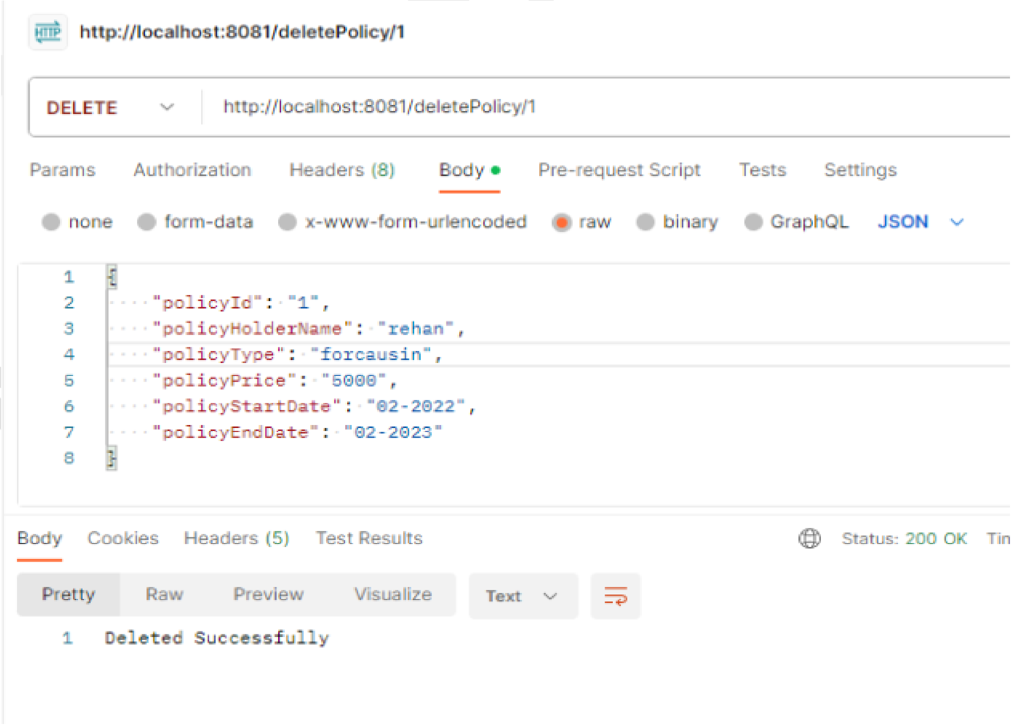
```
{
    "policyId": "3",
    "policyHolderName": "ishan",
    "policyType": "forbrother",
    "policyPrice": "3000",
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 15 r

Pretty Raw Preview Visualize JSON

```
{
    "policyId": 1,
    "policyHolderName": "rehan",
    "policyType": "forcausin",
    "policyPrice": 1000.0,
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

d. /deletePolicy/{policy id} (HTTP Method : DELETE) (No Request Body)



http://localhost:8081/deletePolicy/1

DELETE http://localhost:8081/deletePolicy/1

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded ● raw binary GraphQL JSON

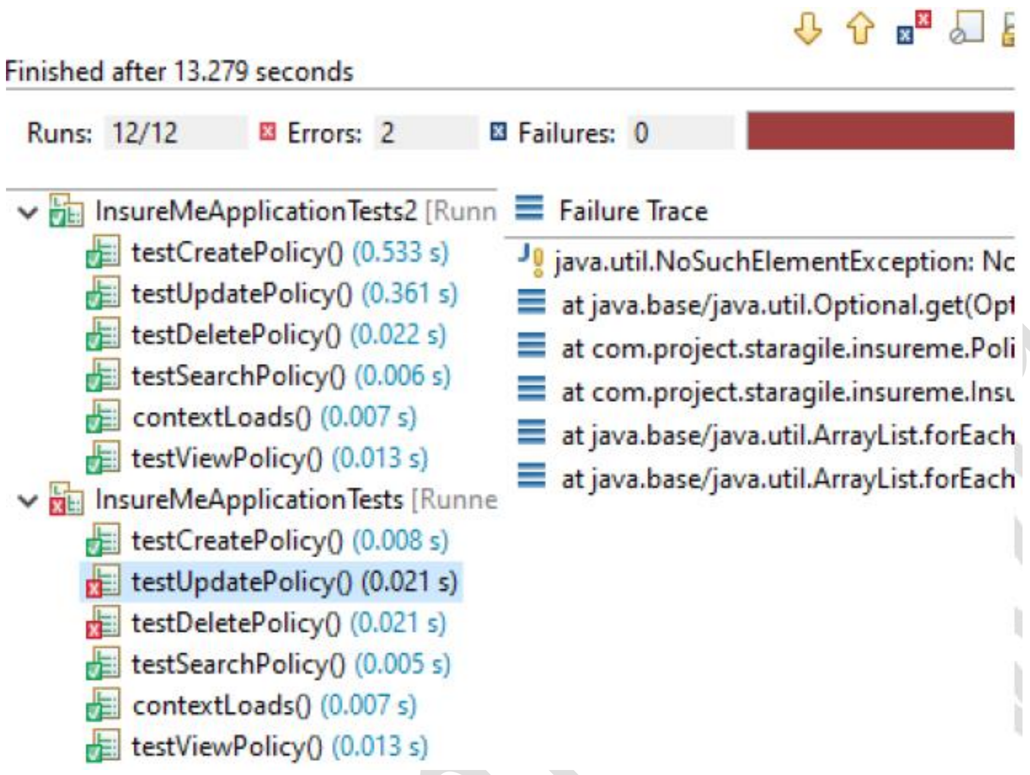
```
{
    "policyId": "1",
    "policyHolderName": "rehan",
    "policyType": "forcausin",
    "policyPrice": "5000",
    "policyStartDate": "02-2022",
    "policyEndDate": "02-2023"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK Tin

Pretty Raw Preview Visualize Text

```
Deleted Successfully
```

2. Write necessary Junit testcase. Run the Junit tests:- Here the necessary Junit testcase are present which I ran there and it gives me the desired result as expected.



Finished after 13.279 seconds

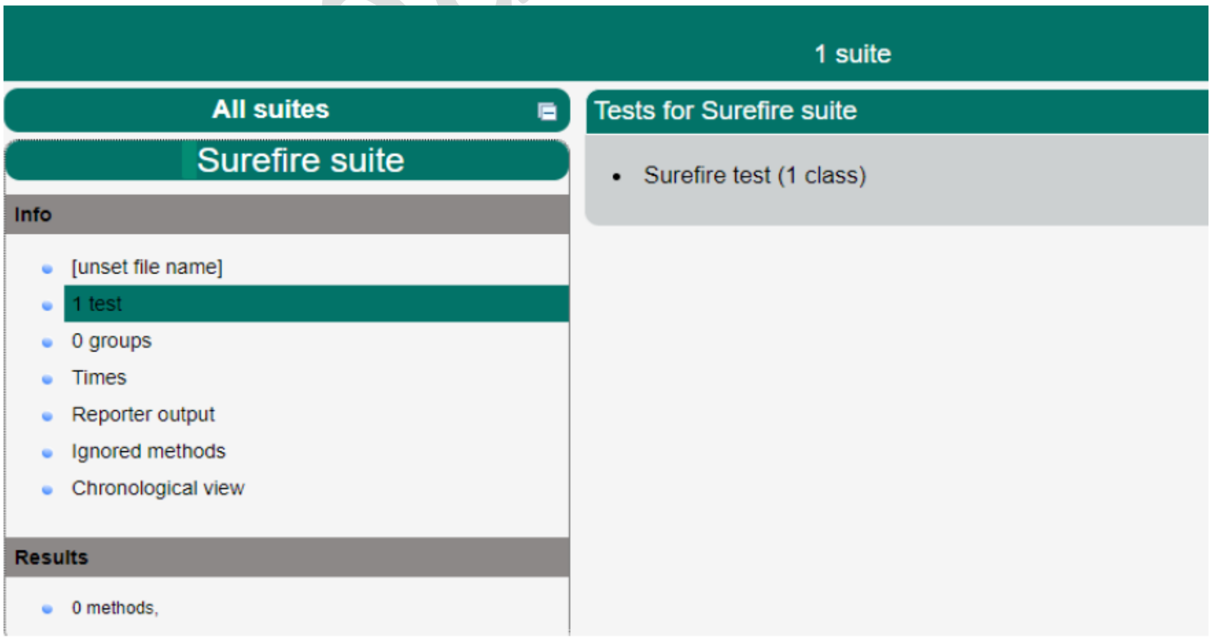
Runs: 12/12 Errors: 2 Failures: 0

```
InsureMeApplicationTests2 [Runn
    testCreatePolicy() (0.533 s)
    testUpdatePolicy() (0.361 s)
    testDeletePolicy() (0.022 s)
    testSearchPolicy() (0.006 s)
    contextLoads() (0.007 s)
    testViewPolicy() (0.013 s)
InsureMeApplicationTests [Runne
    testCreatePolicy() (0.008 s)
    testUpdatePolicy() (0.021 s)
    testDeletePolicy() (0.021 s)
    testSearchPolicy() (0.005 s)
    contextLoads() (0.007 s)
    testViewPolicy() (0.013 s)
```

Failure Trace

```
java.util.NoSuchElementException: Nc
at java.base/java.util.Optional.get(Opt
at com.project.staragile.insureme.Poli
at com.project.staragile.insureme.Insu
at java.base/java.util.ArrayList.forEach
at java.base/java.util.ArrayList.forEach
```

3.Generate HTML report using TestNG:- I generated the HTML Report using TestNG. You can see it below.



1 suite

All suites

Surefire suite

Info

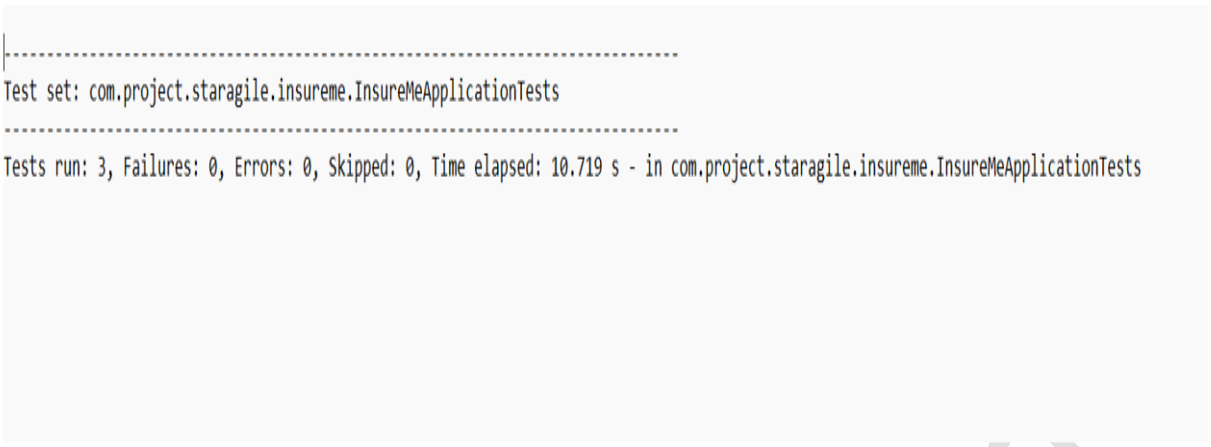
- [unset file name]
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 0 methods,

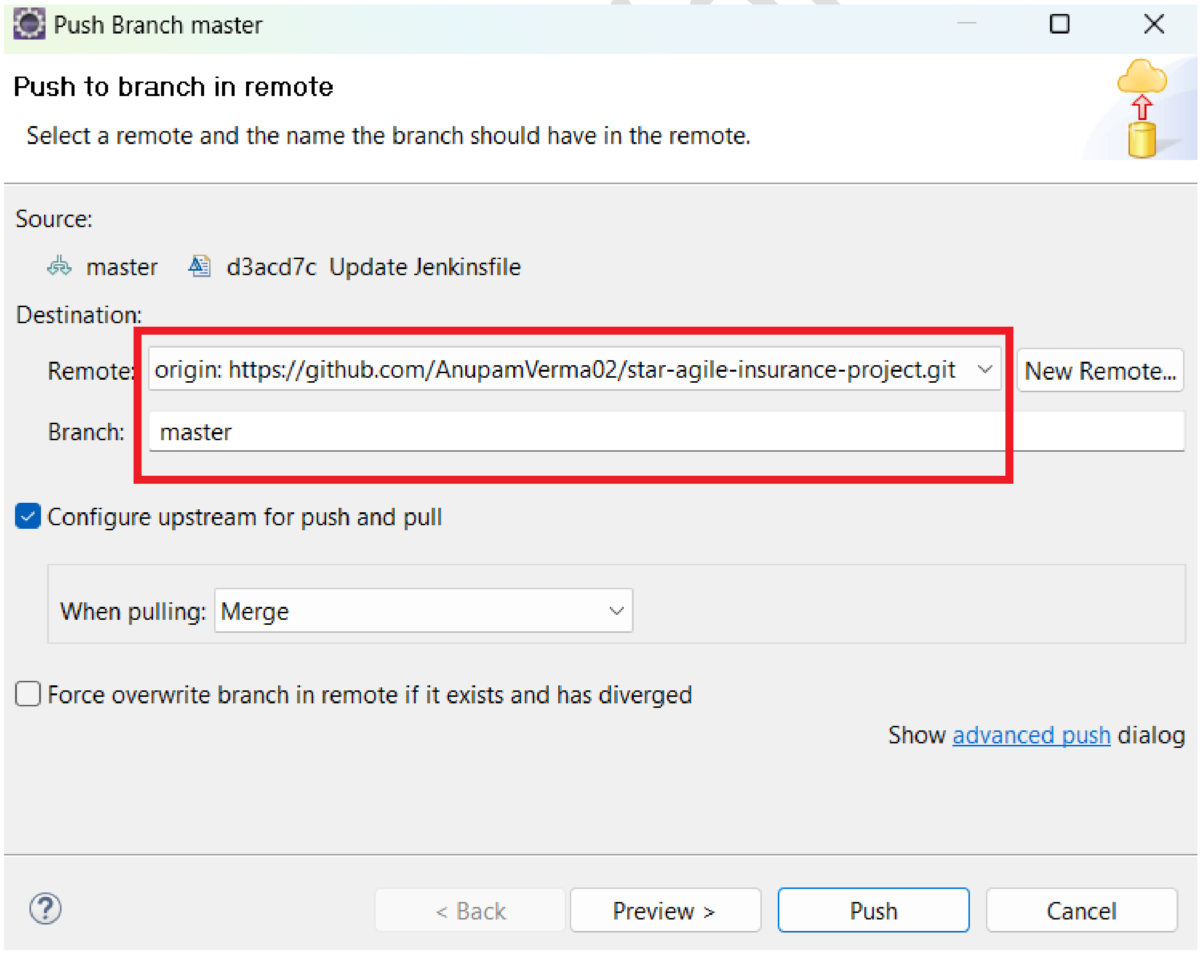
Tests for Surefire suite

- Surefire test (1 class)



```
-------------------------------------------------------------------------------
Test set: com.project.staragile.insureme.InsureMeApplicationTests
-------------------------------------------------------------------------------
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.719 s - in com.project.staragile.insureme.InsureMeApplicationTests
```

4. Push your code into your GitHub Repository:- Here, I pushed my code to my GitHub Repository.



Push Branch master

Push to branch in remote

Select a remote and the name the branch should have in the remote.

Source:

master d3acd7c Update Jenkinsfile

Destination:

Remote: origin: https://github.com/AnupamVerma02/star-agile-insurance-project.git

Branch: master

[✓] Configure upstream for push and pull

When pulling: Merge

[ ] Force overwrite branch in remote if it exists and has diverged

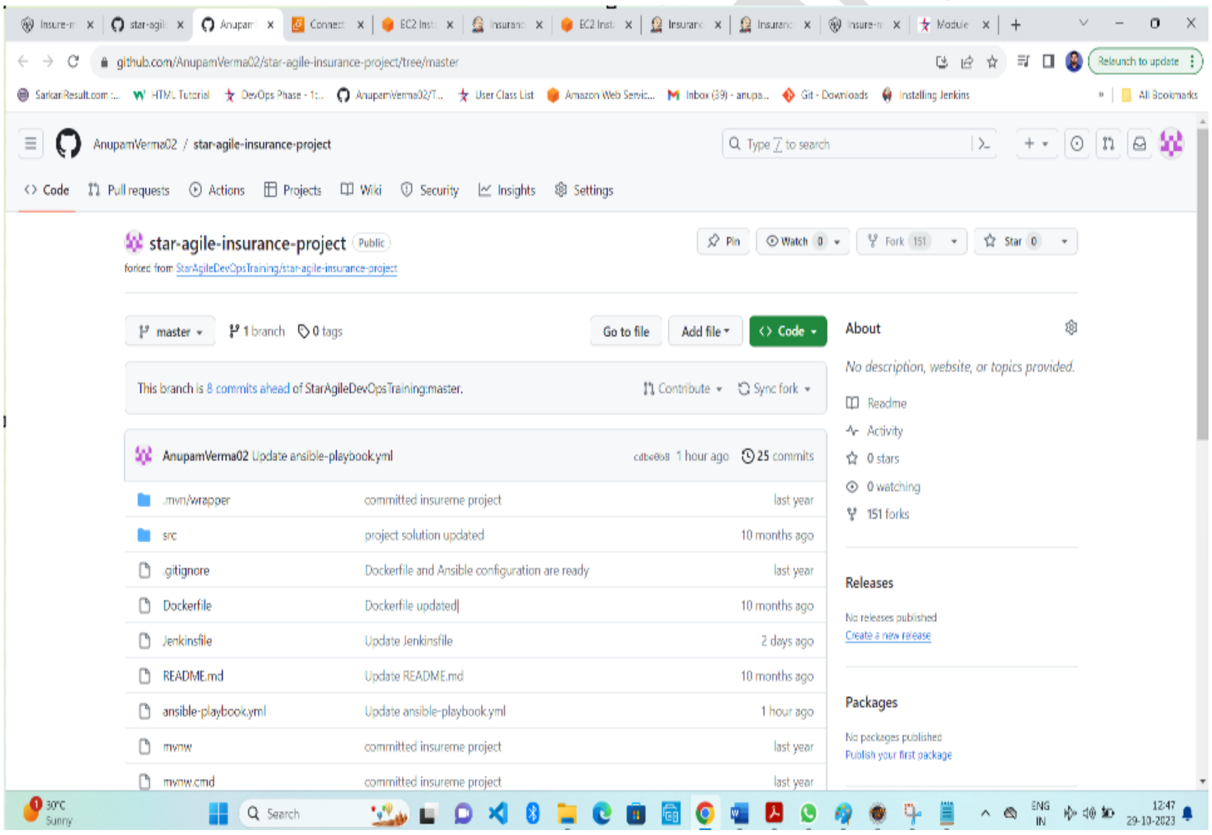
Show advanced push dialog

Later, you need to implement Continuous Integration & Continuous Deployment using following tools:

✓ 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.

Project pushed to GitHub

Git -I used here GIT for version control & for tracking changes in the code files



github.com/AnupamVerma02/star-agile-insurance-project/tree/master

AnupamVerma02 / star-agile-insurance-project

star-agile-insurance-project Public
forked from StarAgileDevOpsTraining/star-agile-insurance-project

master · 1 branch · 0 tags

This branch is 8 commits ahead of StarAgileDevOpsTraining:master.

AnupamVerma02 Update ansible-playbook.yml · cdbe0a8 1 hour ago · 25 commits

.mvn/wrapper	committed insureme project	last year
src	project solution updated	10 months ago
.gitignore	Dockerfile and Ansible configuration are ready	last year
Dockerfile	Dockerfile updated|	10 months ago
Jenkinsfile	Update Jenkinsfile	2 days ago
README.md	Update README.md	10 months ago
ansible-playbook.yml	Update ansible-playbook.yml	1 hour ago
mvnw	committed insureme project	last year
mvnw.cmd	committed insureme project	last year

About
No description, website, or topics provided.
Readme · Activity · 0 stars · 0 watching · 151 forks

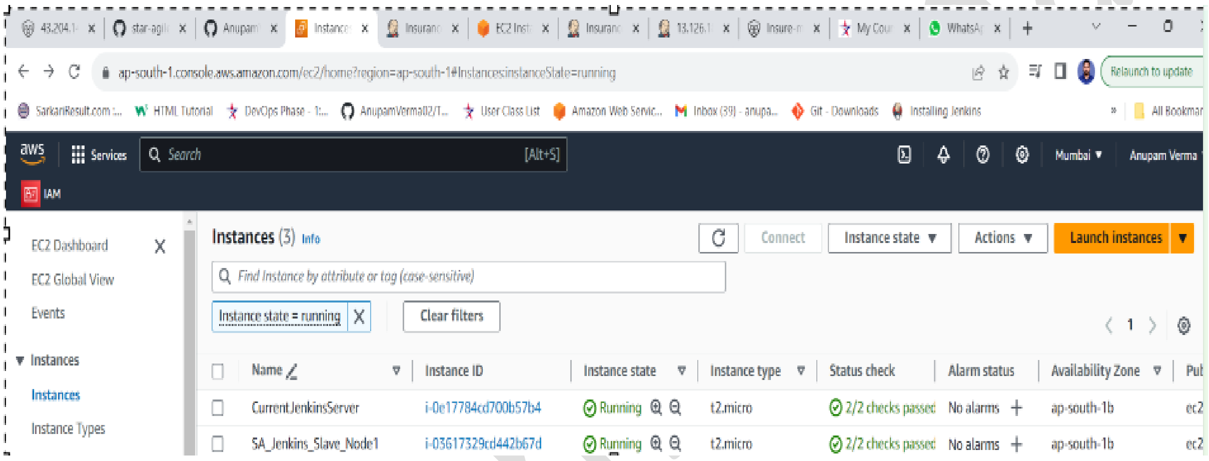
Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

DevOps Steps for Project

Jenkins -I used Jenkins here for continuous integration and continuous deployment.

1. Configuring Jenkins Master and Jenkins Slave node. Creating 2 EC2 instances. Jenkins Master and Jenkins Slave. Installing required software on Jenkins Master and Slave. Installing Maven in slave as we are going to build the project on Slave node and creating artifacts there.


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running

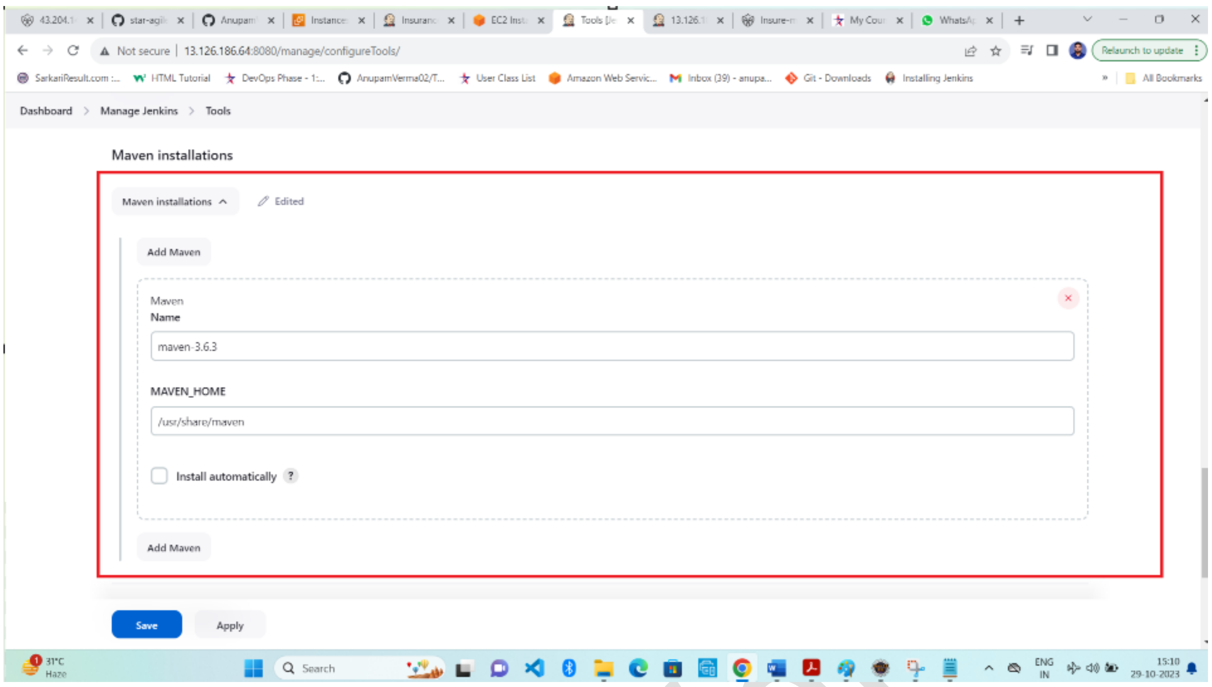
Instances (3)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
CurrentJenkinsServer	i-0e17784cd700b57b4	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2
SA_Jenkins_Slave_Node1	i-03617329cd442b67d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2

2. Creating and establishing the connection between Jenkins master and slave node using user devopsadmin and SSH keypair connection. Add the Slave node under nodes in Jenkins Master and enter the server details username and private key.

3. Then I Added the maven plugin in Jenkins Master and configure the maven tool under tools in Jenkins master to point the maven application path in slave node.You can see below in screenshot my configuration of Maven.


Dashboard > Manage Jenkins > Tools

Maven installations

Maven installations — Edited

Add Maven

Maven
Name: maven-3.6.3

MAVEN_HOME: /usr/share/maven

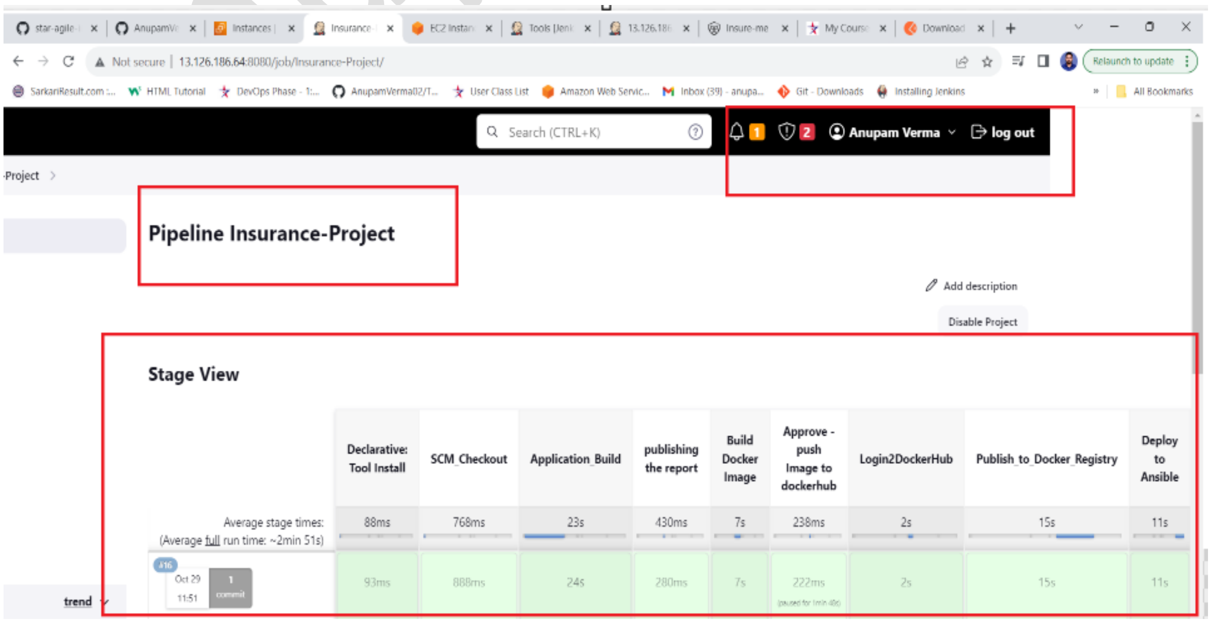
Install automatically ?

Add Maven

4. Write the Pipeline to build the application:-There are few stages here.

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.
Check if the application is running correctly and artifacts are generated correctly on slave node.I am showing below all the required Screenshots.

This is my screenshot after completing all the stages.I show you one by one.

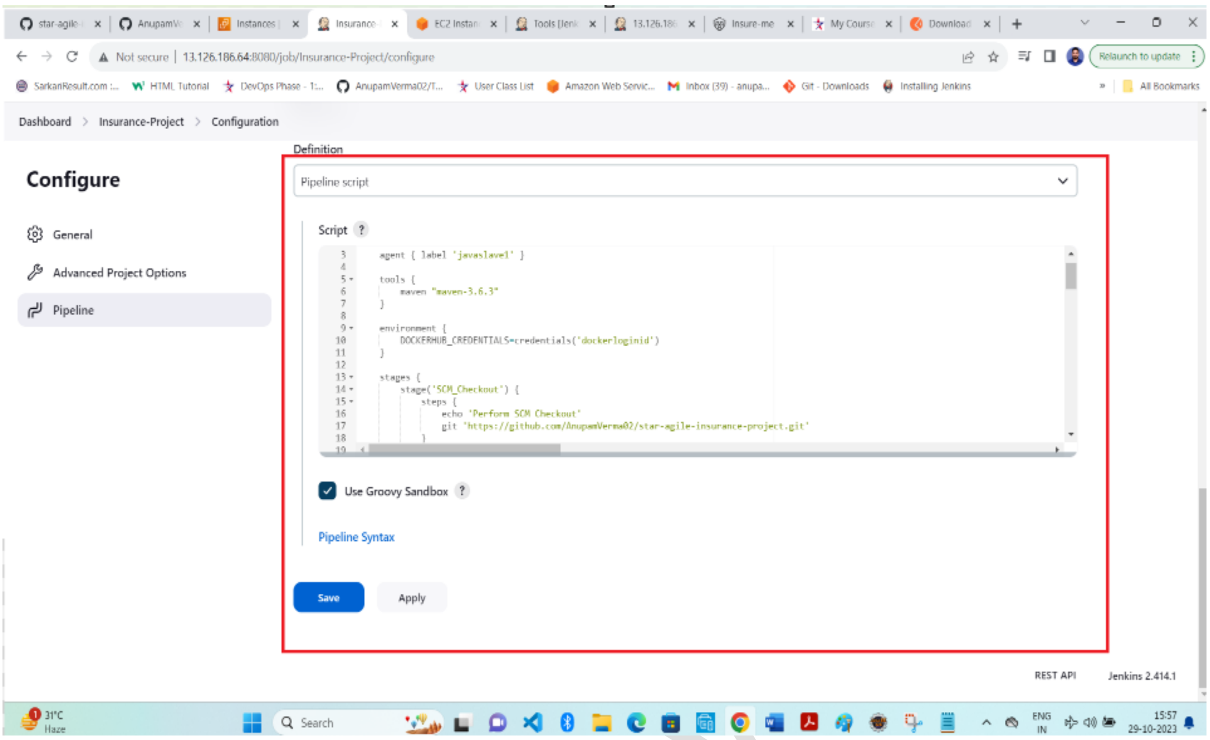

13.126.186.64:8080/job/Insurance-Project/

Pipeline Insurance-Project

Stage View

	Declarative: Tool Install	SCM_Checkout	Application_Build	publishing the report	Build Docker Image	Approve - push Image to dockerhub	Login2DockerHub	Publish_to_Docker_Registry	Deploy to Ansible
Average stage times: (Average full run time: ~2min 51s)	88ms	768ms	23s	430ms	7s	238ms	2s	15s	11s
#16 Oct 29 11:51 1 commit	93ms	888ms	24s	280ms	7s	222ms (paused for 1min 48s)	2s	15s	11s

Here you can see in the Pipeline code I did the SCM Code Checkout and build.



Dashboard > Insurance-Project > Configuration

Configure

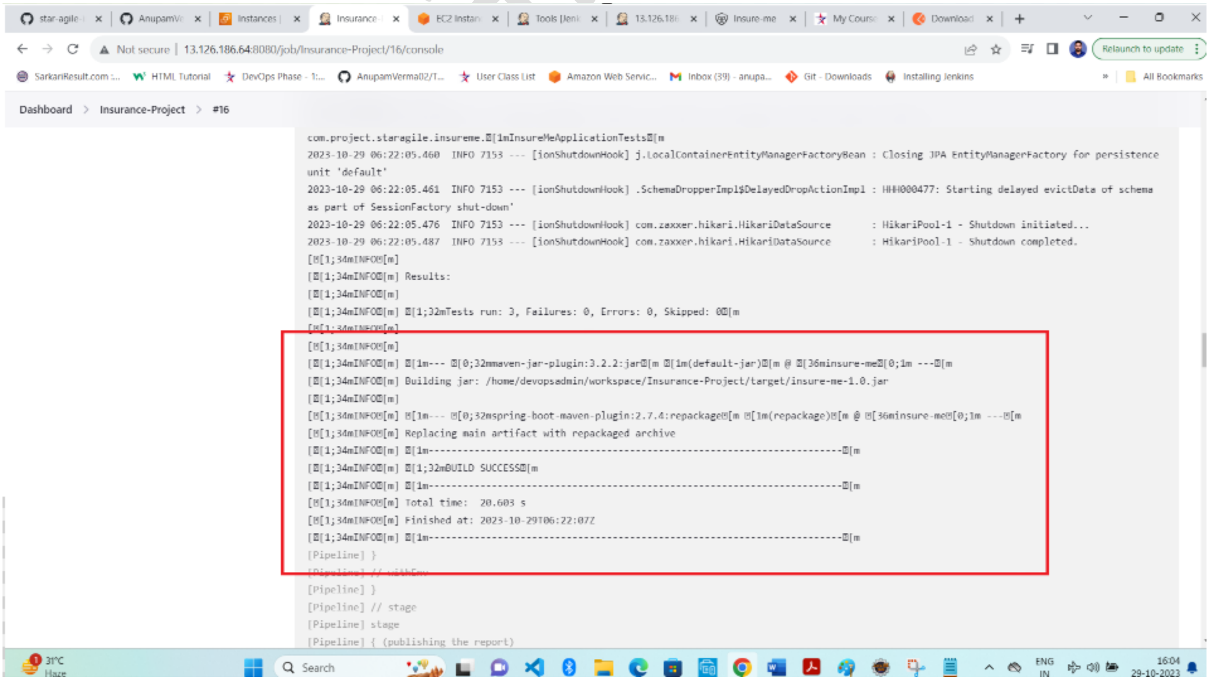
Definition: Pipeline script

Script

```
    agent { label 'javaslave1' }

    tools {
        maven "maven-3.6.3"
    }

    environment {
        DOCKERHUB_CREDENTIALS=credentials('dockerloginid')
    }

    stages {
        stage('SCM_Checkout') {
            steps {
                echo 'Perform SCM Checkout'
                git 'https://github.com/AnupamVerma02/star-agile-insurance-project.git'
            }
```

Use Groovy Sandbox: ✓

After that Jar/Artifacts build successfully in my slave machine.

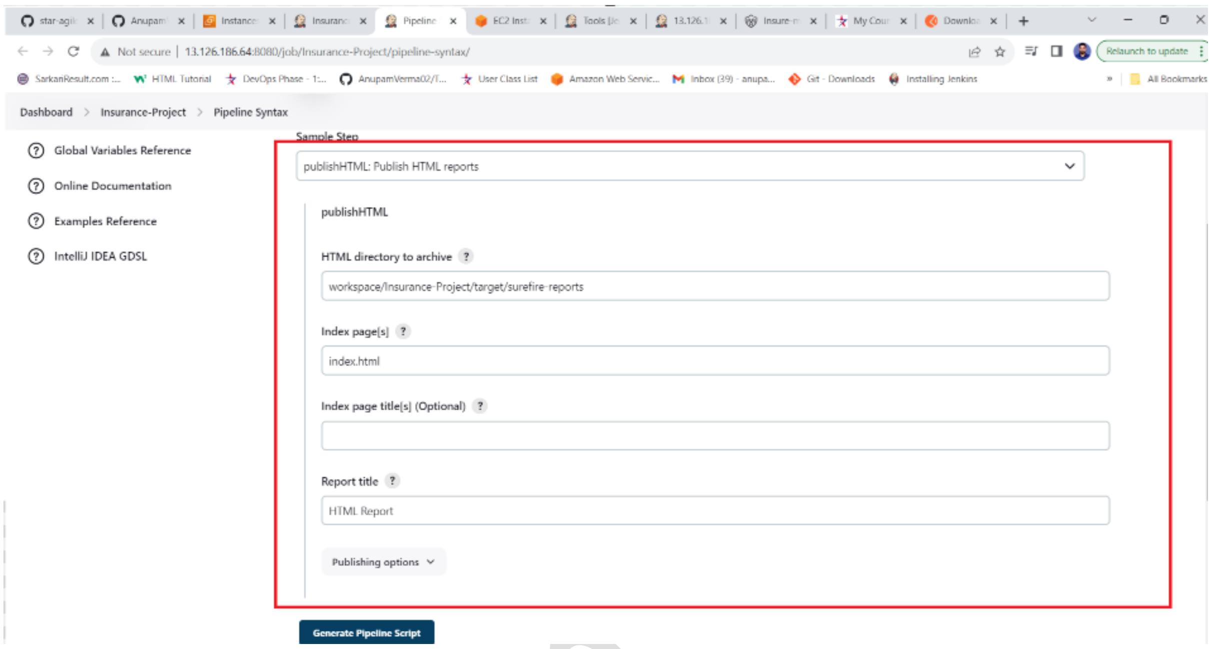


Dashboard > Insurance-Project > #16

```
com.project.staragile.insureme.[1mInsureMeApplicationTests[m
2023-10-29 06:22:05.460  INFO 7153 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence
unit 'default'
2023-10-29 06:22:05.461  INFO 7153 --- [ionShutdownHook] .SchemaDropperImpl$DelayedDropActionImpl : HHH000477: Starting delayed evictData of schema
as part of SessionFactory shut-down'
2023-10-29 06:22:05.476  INFO 7153 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2023-10-29 06:22:05.487  INFO 7153 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
[[1;34mINFO[m]
[[1;34mINFO[m] Results:
[[1;34mINFO[m]
[[1;34mINFO[m] [1;32mTests run: 3, Failures: 0, Errors: 0, Skipped: 0[m
[[1;34mINFO[m]
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mmaven-jar-plugin:3.2.2:jar[m [1m(default-jar)[m @ [36minsure-me[0;1m ---[m
[[1;34mINFO[m] Building jar: /home/devopsadmin/workspace/Insurance-Project/target/insure-me-1.0.jar
[[1;34mINFO[m]
[[1;34mINFO[m] [1m--- [0;32mspring-boot-maven-plugin:2.7.4:repackage[m [1m(repackage)[m @ [36minsure-me[0;1m ---[m
[[1;34mINFO[m] Replacing main artifact with repackaged archive
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] [1;32mBUILD SUCCESS[m
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] Total time:  20.603 s
[[1;34mINFO[m] Finished at: 2023-10-29T06:22:07Z
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (publishing the report)
```

3. Stage 3: Publish HTML reports.

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



Not secure | 13.126.186.64:8080/job/Insurance-Project/pipeline-syntax/
Dashboard > Insurance-Project > Pipeline Syntax
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Publishing options
Generate Pipeline Script

Sample Step: publishHTML: Publish HTML reports

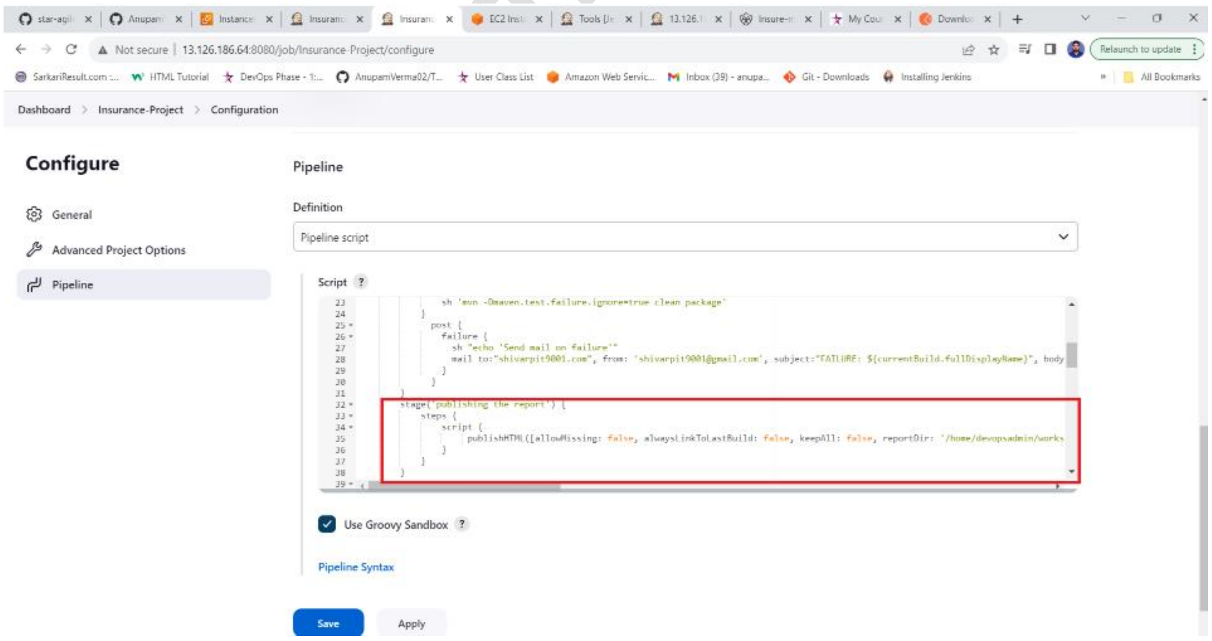
publishHTML

HTML directory to archive: workspace/Insurance-Project/target/surefire-reports

Index page[s]: index.html

Index page title[s] (Optional):

Report title: HTML Report



Not secure | 13.126.186.64:8080/job/Insurance-Project/configure
Dashboard > Insurance-Project > Configuration
Configure
General
Advanced Project Options
Pipeline
Pipeline Syntax
Save
Apply

Pipeline

Definition: Pipeline script

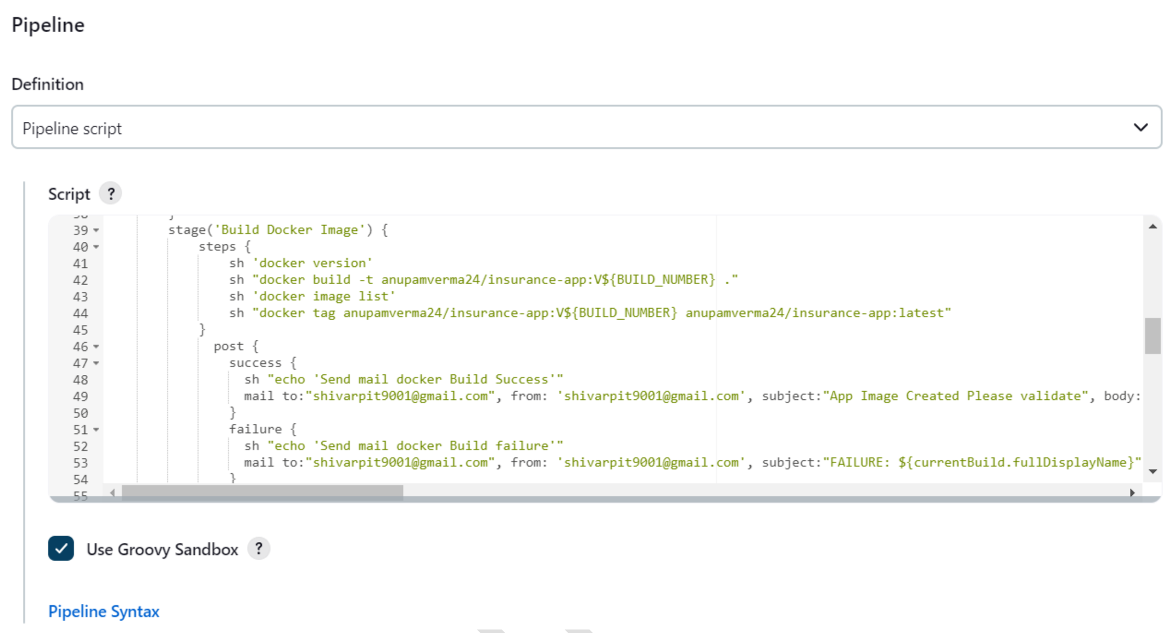
Script

```
            sh 'mvn -Dmaven.test.failure.ignore=true clean package'
        }
      post {
        failure {
          sh "echo 'Send mail on failure'"
          mail to:"shivarpit9001.com", from: 'shivarpit9001@gmail.com', subject:"FAILURE: ${currentBuild.fullDisplayName}", body
        }
      }
    }
    stage('publishing the report') {
        steps {
          script {
              publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/home/devopsadmin/works
          }
        }
    }
```

[✓] Use Groovy Sandbox

Docker - For deploying containerized applications I used Docker. Create DockerHub account. Create dockerHub account token. Add docker hub account details and token to credentials in Jenkins master. Install docker on Jenkins Slave. Add stage 4 building docker image to pipeline script.

Stage 4 : Build docker image and publish image to DockerHub Registry.



Pipeline

Definition

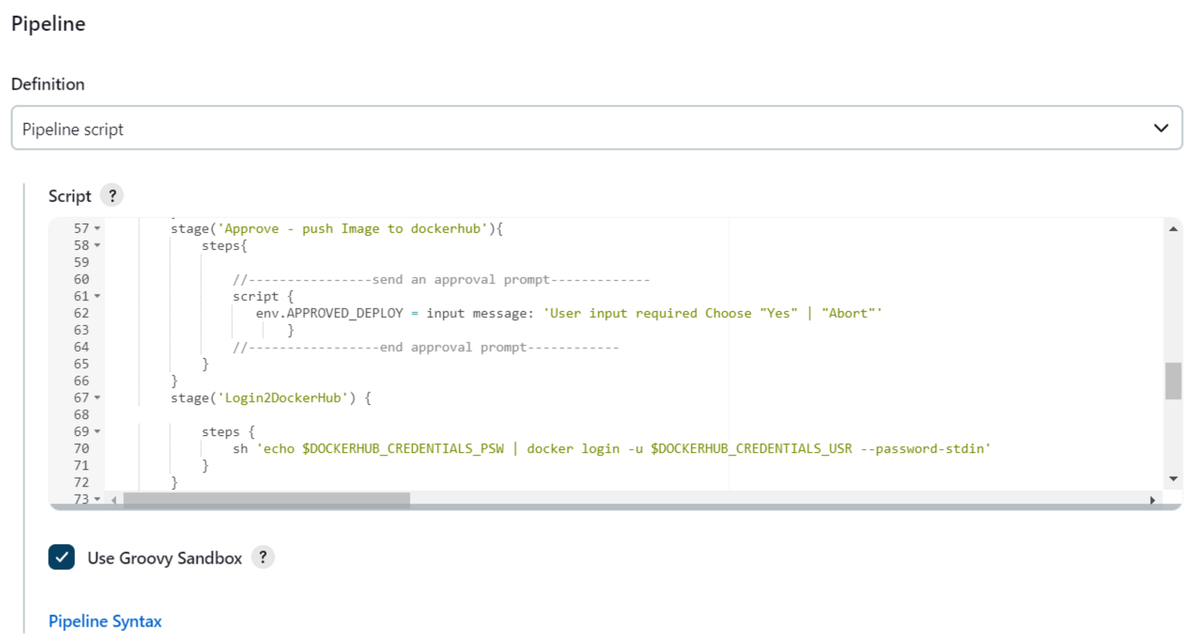
Pipeline script

Script

```
        stage('Build Docker Image') {
            steps {
                sh 'docker version'
                sh "docker build -t anupamverma24/insurance-app:V${BUILD_NUMBER} ."
                sh 'docker image list'
                sh "docker tag anupamverma24/insurance-app:V${BUILD_NUMBER} anupamverma24/insurance-app:latest"
            }
              post {
                success {
                  sh "echo 'Send mail docker Build Success'"
                  mail to:"shivarpit9001@gmail.com", from: 'shivarpit9001@gmail.com', subject:"App Image Created Please validate", body:
                }
                failure {
                  sh "echo 'Send mail docker Build failure'"
                  mail to:"shivarpit9001@gmail.com", from: 'shivarpit9001@gmail.com', subject:"FAILURE: ${currentBuild.fullDisplayName}"
                }
```

Use Groovy Sandbox ✓

Pipeline Syntax



Pipeline

Definition

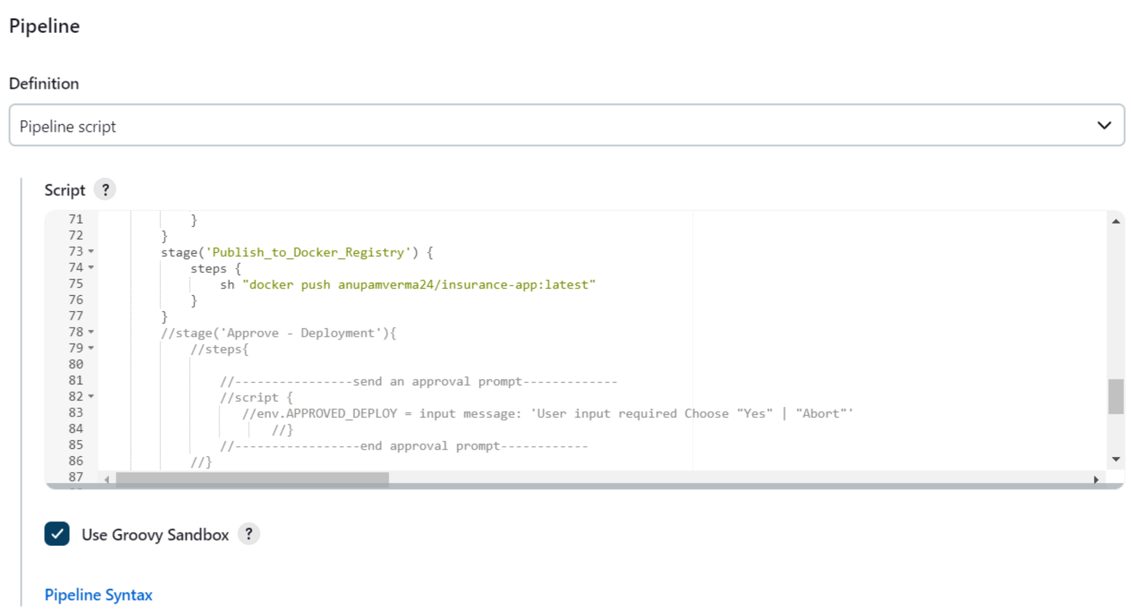
Pipeline script

Script

```
        stage('Approve - push Image to dockerhub'){
            steps{

                //---------------send an approval prompt------------
                script {
                   env.APPROVED_DEPLOY = input message: 'User input required Choose "Yes" | "Abort"'
                       }
                //----------------end approval prompt-----------
            }
        }
        stage('Login2DockerHub') {

            steps {
                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
            }
        }
```

Use Groovy Sandbox ✓

Pipeline Syntax



Pipeline

Definition

Pipeline script

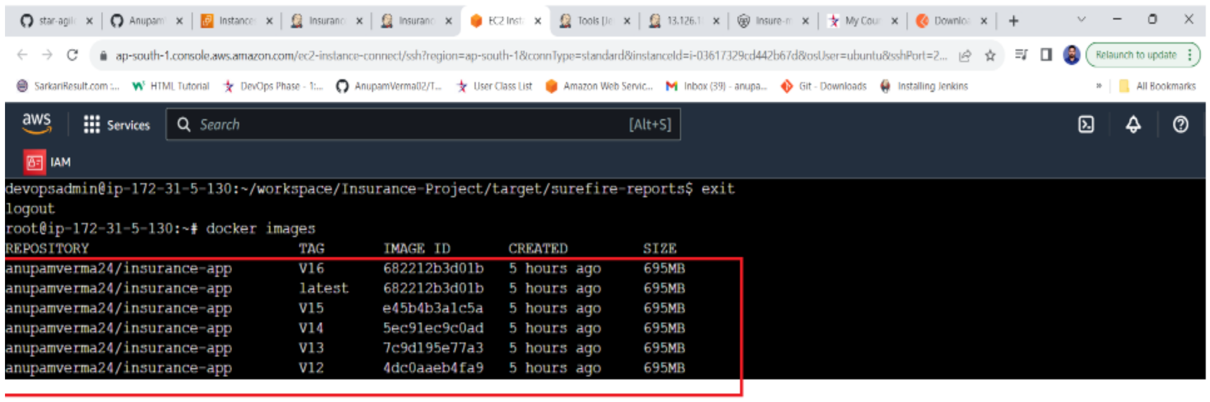
Script

```
        }
    }
    stage('Publish_to_Docker_Registry') {
        steps {
            sh "docker push anupamverma24/insurance-app:latest"
        }
    }
    //stage('Approve - Deployment'){
        //steps{

            //---------------send an approval prompt------------
            //script {
               //env.APPROVED_DEPLOY = input message: 'User input required Choose "Yes" | "Abort"'
                   //}
            //----------------end approval prompt-----------
        //}
```

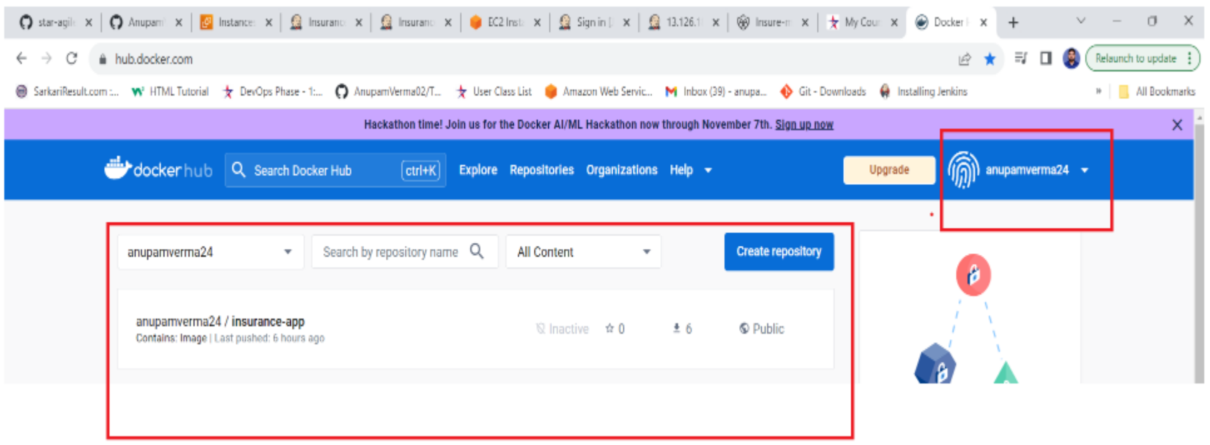
[✓] Use Groovy Sandbox

Pipeline Syntax

Here, you can see my image is present in my Docker Hub Account.



```
devopsadmin@ip-172-31-5-130:~/workspace/Insurance-Project/target/surefire-reports$ exit
logout
root@ip-172-31-5-130:~# docker images
REPOSITORY                      TAG      IMAGE ID       CREATED       SIZE
anupamverma24/insurance-app     V16      682212b3d01b   5 hours ago   695MB
anupamverma24/insurance-app     latest   682212b3d01b   5 hours ago   695MB
anupamverma24/insurance-app     V15      e45b4b3a1c5a   5 hours ago   695MB
anupamverma24/insurance-app     V14      5ec91ec9c0ad   5 hours ago   695MB
anupamverma24/insurance-app     V13      7c9d195e77a3   5 hours ago   695MB
anupamverma24/insurance-app     V12      4dc0aaeb4fa9   5 hours ago   695MB
```

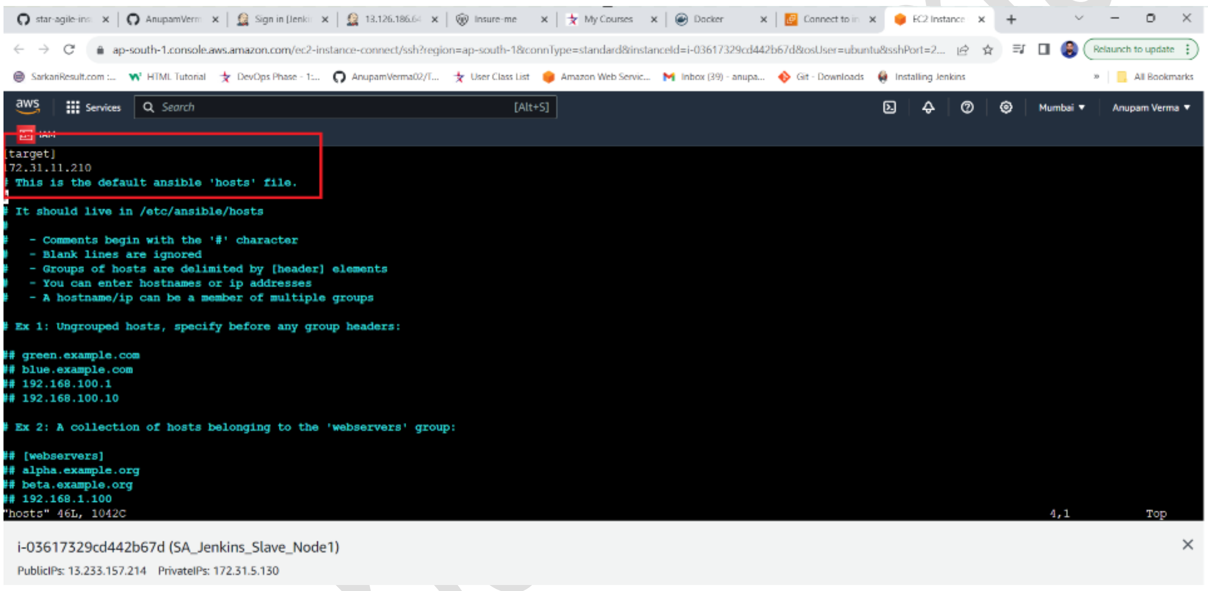


Hackathon time! Join us for the Docker AI/ML Hackathon now through November 7th. Sign up now

anupamverma24 / insurance-app
Contains: Image | Last pushed: 6 hours ago
Inactive ☆ 0 ⬇ 6 Public

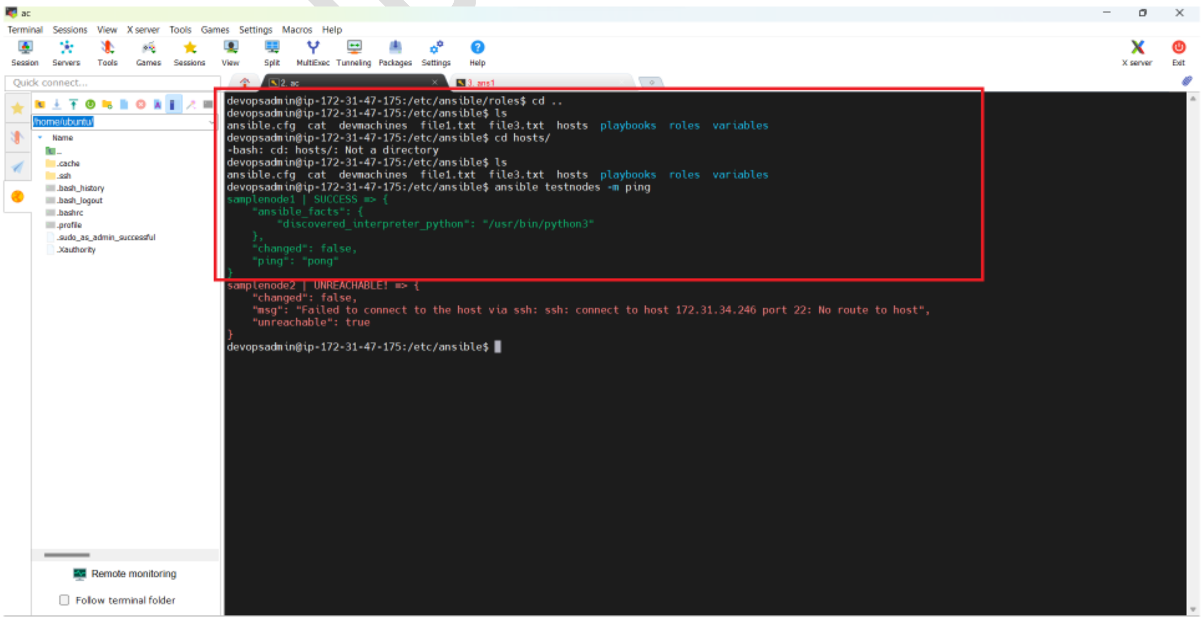
Ansible – Ansible is a Configuration management tool.

Add one more EC2 instance as Deployment server.
Install Ansible on slave node and make slave node as Ansible controller. Configure a user on deployment server and connect the user to ansible controller using SSH keys. Add deployment server IP in /etc/Ansible/hosts in Ansible controller as target to Ansible controller. (Deployment server would be target to Slave node) check the connection is working by using the ping command from Ansible controller.



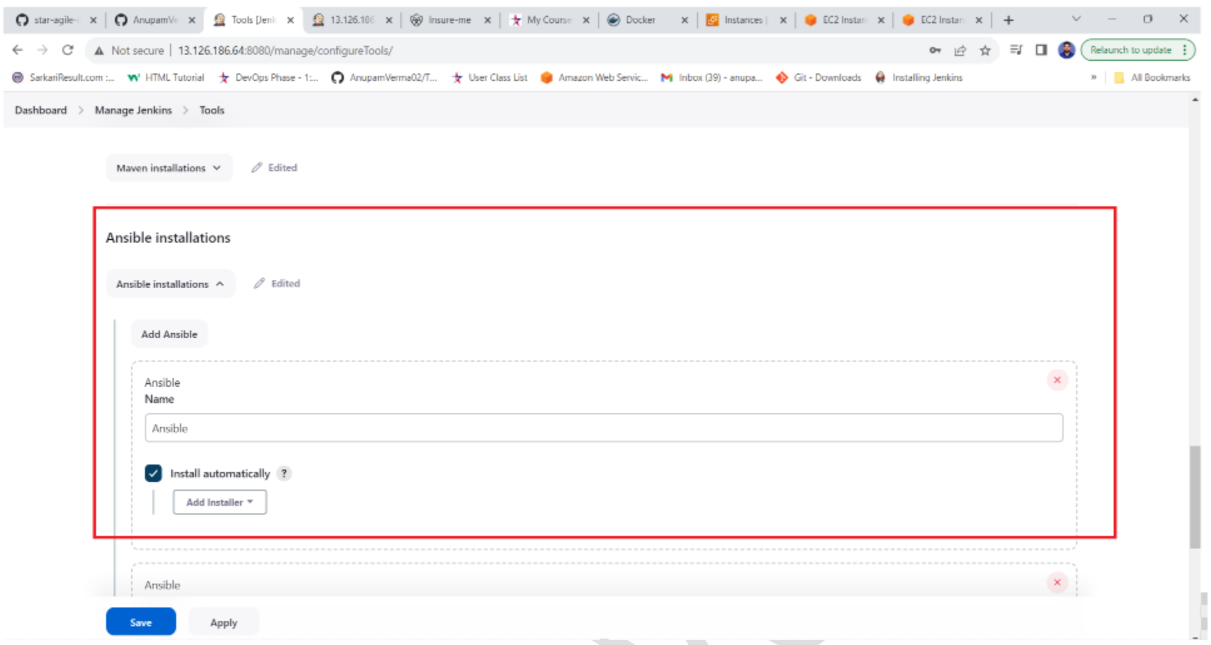
```
[target]
172.31.11.210
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
"hosts" 46L, 1042C                                                    4,1           Top
```

i-03617329cd442b67d (SA_Jenkins_Slave_Node1)
PublicIPs: 13.233.157.214 PrivateIPs: 172.31.5.130



```
devopsadmin@ip-172-31-47-175:/etc/ansible/roles$ cd ..
devopsadmin@ip-172-31-47-175:/etc/ansible$ ls
ansible.cfg  cat  devmachines  file1.txt  file3.txt  hosts  playbooks  roles  variables
devopsadmin@ip-172-31-47-175:/etc/ansible$ cd hosts/
-bash: cd: hosts/: Not a directory
devopsadmin@ip-172-31-47-175:/etc/ansible$ ls
ansible.cfg  cat  devmachines  file1.txt  file3.txt  hosts  playbooks  roles  variables
devopsadmin@ip-172-31-47-175:/etc/ansible$ ansible testnodes -m ping
samplenode1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
samplenode2 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: ssh: connect to host 172.31.34.246 port 22: No route to host",
    "unreachable": true
}
devopsadmin@ip-172-31-47-175:/etc/ansible$
```

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


Not secure | 13.126.186.64:8080/manage/configureTools/

Dashboard > Manage Jenkins > Tools

Maven installations — Edited

Ansible installations

Ansible installations — Edited

Add Ansible

Ansible Name: Ansible

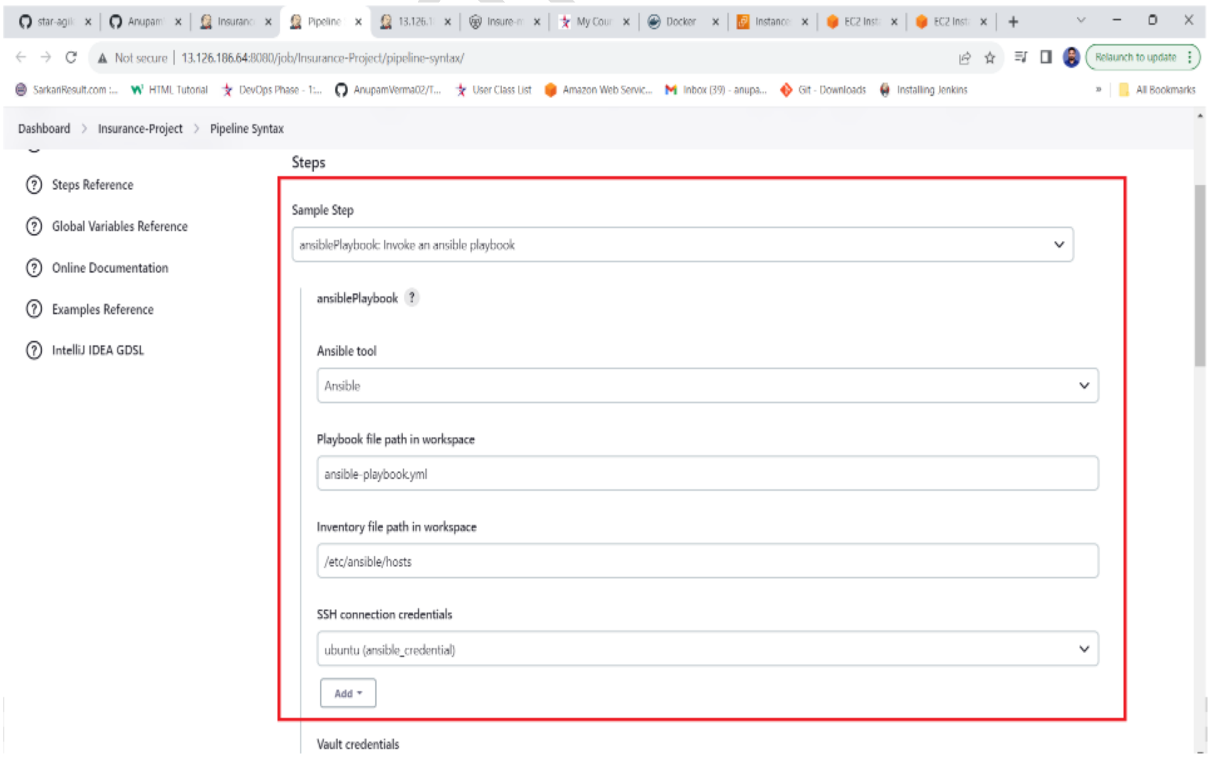
✓ Install automatically

Add Installer

Ansible

Save Apply

Create the syntax pipeline for Ansible:-Here I invoked the ansible playbook in my snipper generator below. Here I configured it.


Not secure | 13.126.186.64:8080/job/Insurance-Project/pipeline-syntax/

Dashboard > Insurance-Project > Pipeline Syntax

Steps Reference

Global Variables Reference

Online Documentation

Examples Reference

IntelliJ IDEA GDSL

Steps

Sample Step: ansiblePlaybook: Invoke an ansible playbook

ansiblePlaybook

Ansible tool: Ansible

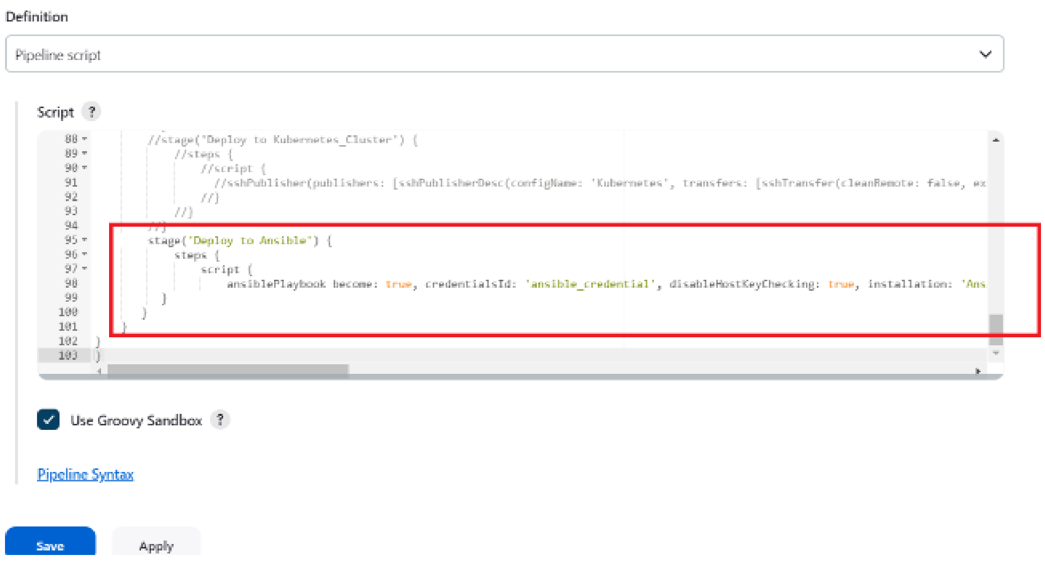
Playbook file path in workspace: ansible-playbook.yml

Inventory file path in workspace: /etc/ansible/hosts

SSH connection credentials: ubuntu (ansible_credential)

Add

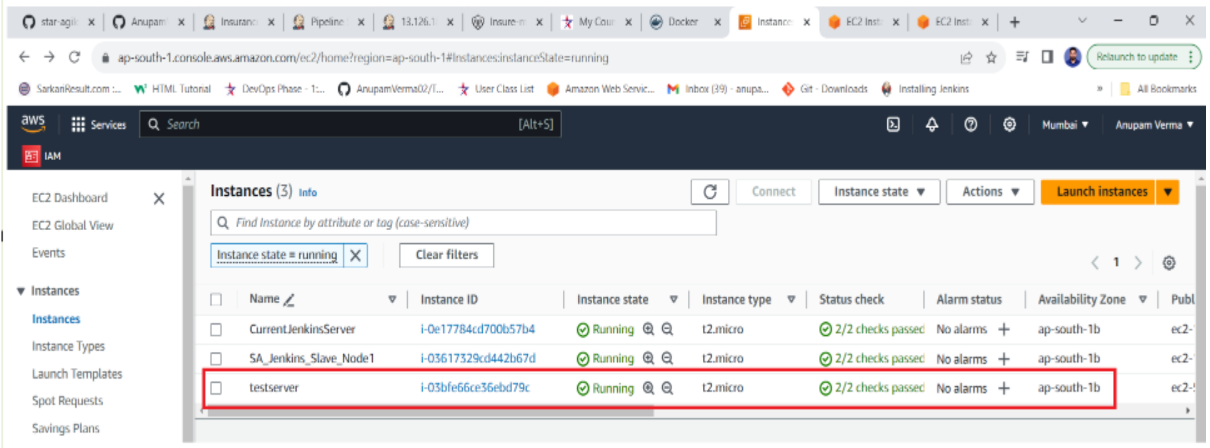
Vault credentials



```
        //stage('Deploy to Kubernetes_Cluster') {
            //steps {
                //script {
                  //sshPublisher(publishers: [sshPublisherDesc(configName: 'Kubernetes', transfers: [sshTransfer(cleanRemote: false, ex
                //}
            //}
        //}
        stage('Deploy to Ansible') {
            steps {
                script {
                    ansiblePlaybook become: true, credentialsId: 'ansible_credential', disableHostKeyChecking: true, installation: 'Ans
            }
        }
    }
}
}
```

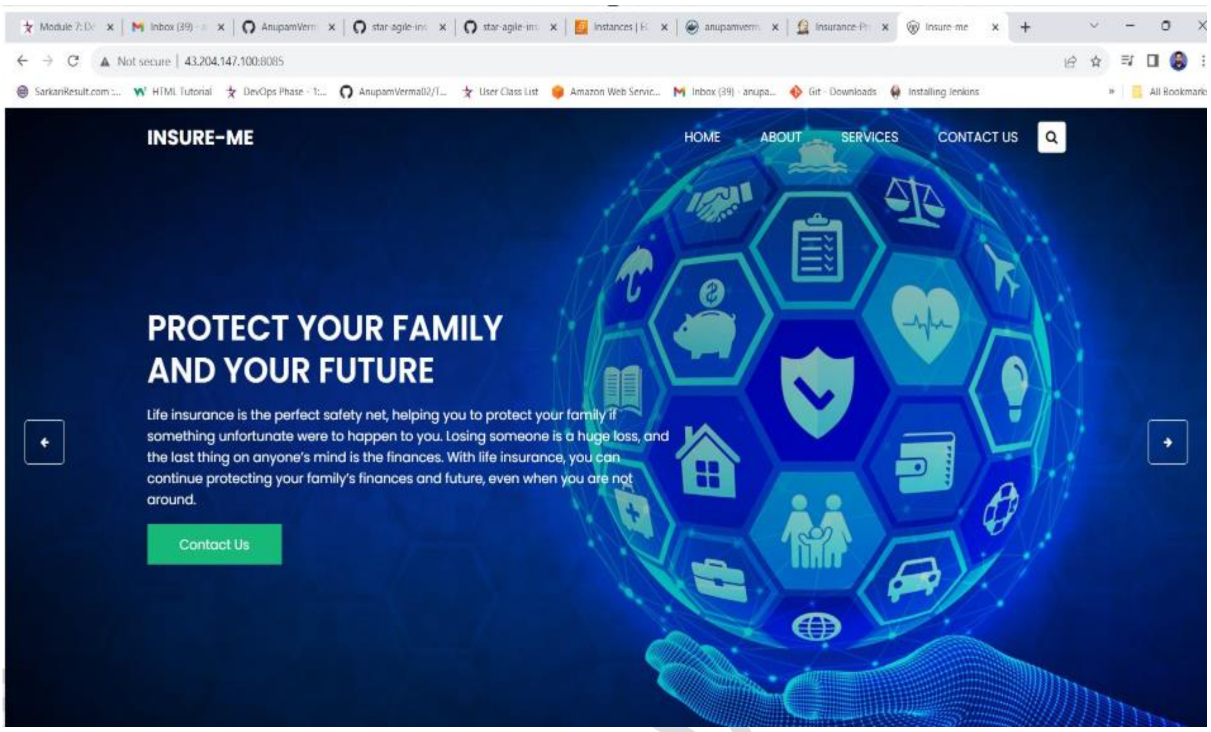
AWS : For creating ec2 machines as servers and deploy the web application.

Created Jenkins Master, Slave and testserver for application build and deployment.I deployed my web application on the testserver.



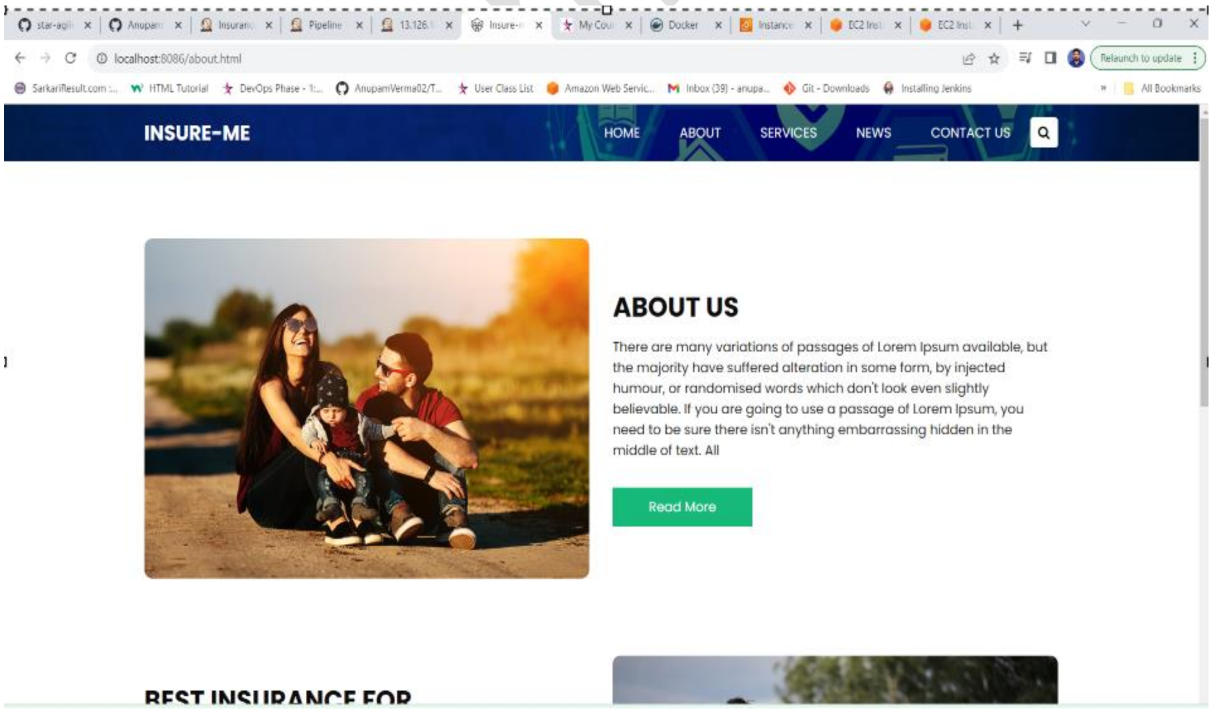
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
CurrentJenkinsServer	i-0e17784cd700b57b4	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-
SA_Jenkins_Slave_Node1	i-03617329cd442b67d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-
testserver	i-03bfe66ce36ebd79c	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-

Here, below you can see my web application of my Insurance_Project is up and running in the browser.Here, I showed you all the pages of my Insurance Application.



PROTECT YOUR FAMILY
AND YOUR FUTURE

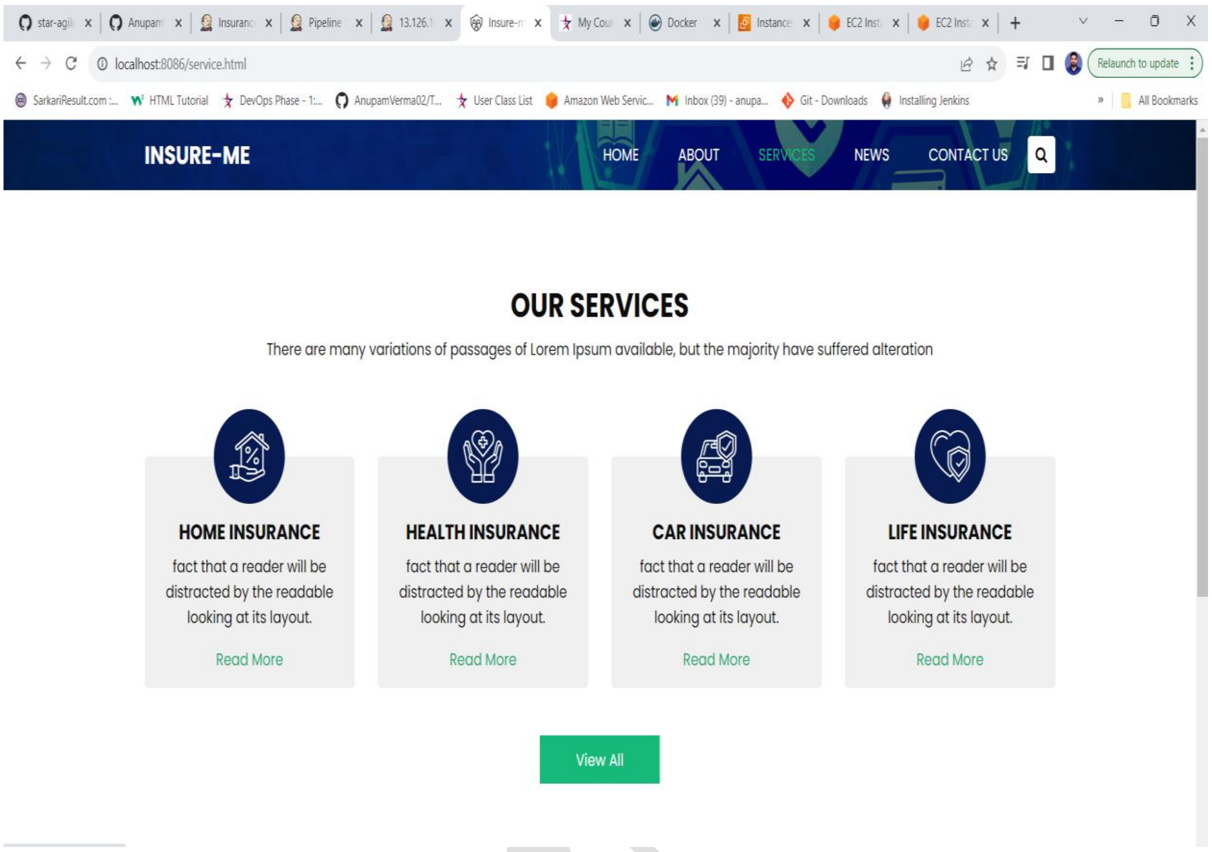
Life insurance is the perfect safety net, helping you to protect your family if something unfortunate were to happen to you. Losing someone is a huge loss, and the last thing on anyone's mind is the finances. With life insurance, you can continue protecting your family's finances and future, even when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



ABOUT US

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All

BEST INSURANCE FOR



OUR SERVICES

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration

HOME INSURANCE

fact that a reader will be distracted by the readable looking at its layout.

Read More

HEALTH INSURANCE

fact that a reader will be distracted by the readable looking at its layout.

Read More

CAR INSURANCE

fact that a reader will be distracted by the readable looking at its layout.

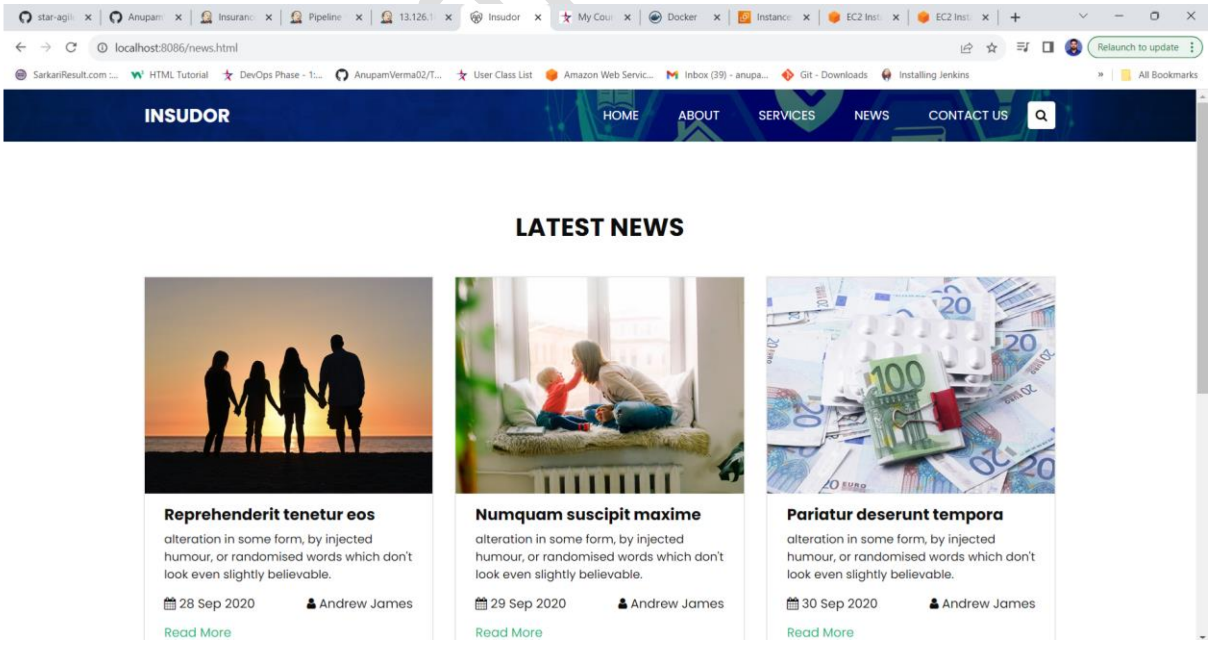
Read More

LIFE INSURANCE

fact that a reader will be distracted by the readable looking at its layout.

Read More

View All



LATEST NEWS

Reprehenderit tenetur eos

alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

28 Sep 2020 Andrew James

Read More

Numquam suscipit maxime

alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

29 Sep 2020 Andrew James

Read More

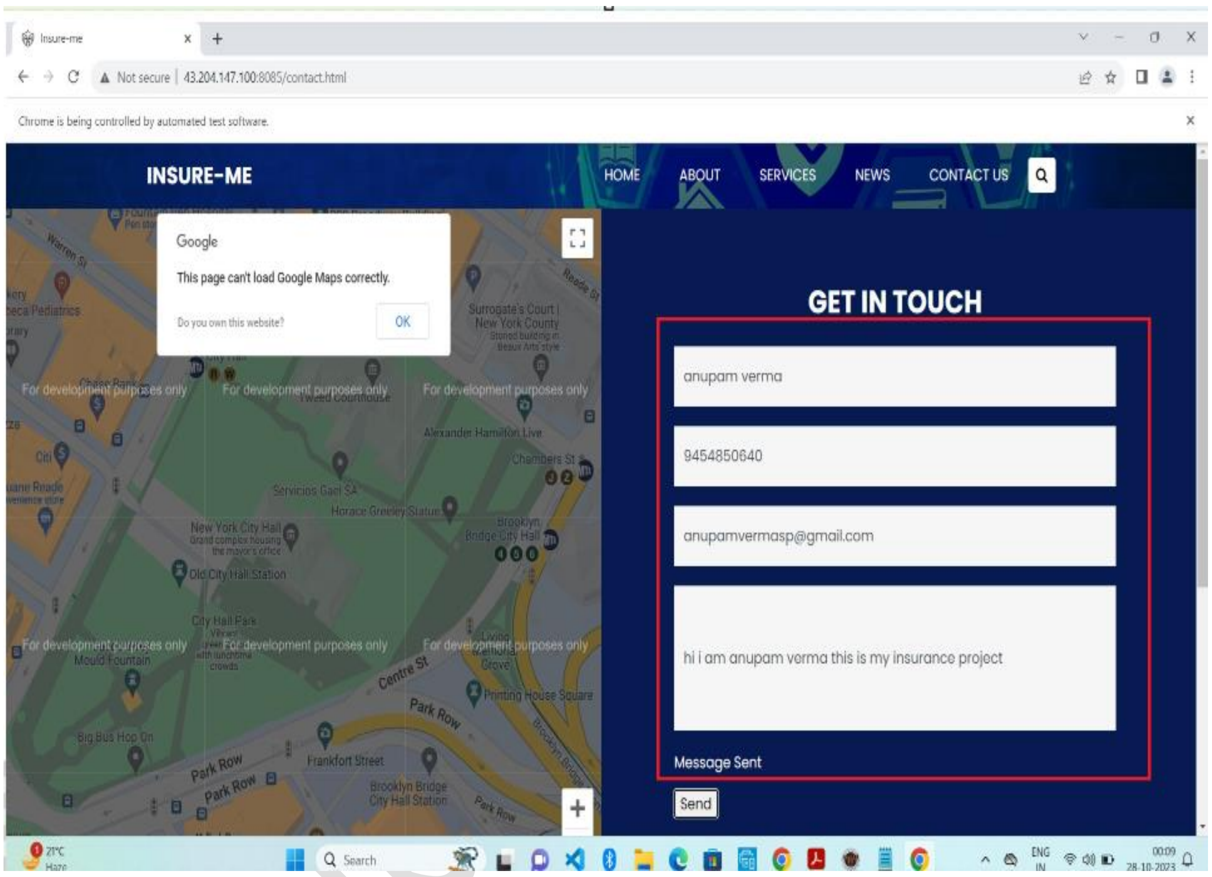
Pariatur deserunt tempora

alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

30 Sep 2020 Andrew James

Read More

Selenium Testing - For automating tests on the deployed web application, I used Selenium. Here you can see below that after performing the selenium testing where I gave my details as to be filled. You can see that it is successfully performing the selenium testing and my details are filled automatically.


Insure-me

Not secure | 43.204.147.100:8085/contact.html

Chrome is being controlled by automated test software.

INSURE-ME

HOME ABOUT SERVICES NEWS CONTACT US

Google

This page can't load Google Maps correctly.

Do you own this website? OK

GET IN TOUCH

anupam verma

9454850640

anupamvermasp@gmail.com

hi i am anupam verma this is my insurance project

Message Sent

Send