

Certification Project – Insure Me

Insurance Domain

Name: - Deepak Kumar

Date of Submission: -30.10.2023

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: -

The screenshot shows a browser window with multiple tabs open. The active tab is for the GitHub repository 'DeepakKumar094 / insurance-project'. The repository page displays a list of commits, including:

- DeepakKumar094 Update ansible-playbook.yml (yesterday)
- .mvn/wrapper committed insureme project (last year)
- src Insurance final commit (8 months ago)
- test-output Insurance final commit (8 months ago)
- .gitignore Dockerfile and Ansible configuration are ready (last year)
- Dockerfile Dockerfile updated (10 months ago)
- Jenkinsfile Update Jenkinsfile (8 months ago)
- README.md Update README.md (7 months ago)
- ansible-playbook.yml Update ansible-playbook.yml (yesterday)
- hosts Create hosts (3 months ago)
- insurebatch5.jar Add files via upload (4 months ago)
- kubernetesfile.yml Update kubernetesfile.yml (4 days ago)
- mvnw committed insureme project (last year)
- mvnw.cmd committed insureme project (last year)
- pom.xml Insurance final commit (8 months ago)

The repository has 19 forks and 0 stars. It also lists activity, releases, packages, and languages (HTML, CSS, SCSS, Java, JavaScript, HCL).

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.

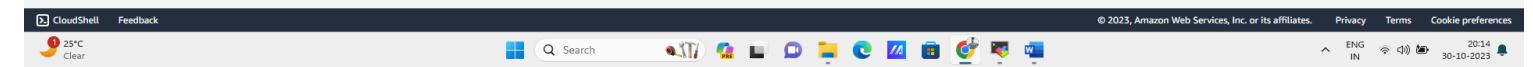
The screenshot shows the AWS CloudWatch Metrics interface. On the left, a sidebar lists various AWS services like EC2 Dashboard, Instances, and Network & Security. The main area displays four line charts under the 'Monitoring' tab. The first chart, 'CPU utilization (%)', shows usage over time with a sharp drop from 70.8% to 35.4% between 18:00 and 18:15. The second chart, 'Status check failed (any) (count)', shows a count of 1 failing status check at 18:15. The third chart, 'Status check failed (instance) (count)', shows a count of 1 failing status check at 18:15. The fourth chart, 'Status check failed (system) (count)', shows a count of 1 failing status check at 18:15. Below the charts, a note indicates the instances involved: 'Instances: i-0c7a457c7e64c0a5a (Project_insurance_Node), i-0355f61956ef13f21 (Project_insurance_Master), i-02b078247de4f93c5 (project_insurance_Test_server)'.

Installed Maven in master Jenkins

The screenshot shows a terminal session in CloudShell. The user is connected via SSH to an EC2 instance. A red box highlights the output of the 'mvn --version' command, which shows the following details:

```
root@ip-172-31-2-230:~# 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: "5.15.0-1048-aws", arch: "amd64", family: "unix"
root@ip-172-31-2-230:~#
```

i-0355f61956ef13f21 (Project_insurance_Master)
PublicIPs: 13.233.136.232 PrivateIPs: 172.31.2.230



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

The screenshot shows the Jenkins 'Tools' configuration page. Under the 'Maven installations' section, a new Maven installation is being added with the name 'maven'. The 'Install automatically' checkbox is checked, and the 'Version' dropdown is set to '3.9.5'. There is also an 'Add Installer' button. At the bottom of the page are 'Save' and 'Apply' buttons, along with a Windows taskbar at the bottom.

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.

Webhooks | Instances | EC2 Instances | Healthcare | Docker Hub | (615) If imma... | WhatsApp | Download | Project_Insurance | insurance | Certificate | +

Not secure | 13.233.136.232:8080/job/insurance/configure

Q2 Gmail YouTube Maps pmsonline.bih.nic.in... AllMoviesHub - 300... Candidate Home trellix DeepakKumar094/T... Resume Editor

Dashboard > insurance > Configuration

Configure

Quiet period ? Trigger builds remotely (e.g., from scripts) ?

General

Advanced Project Options

Pipeline

Advanced

Pipeline

Definition

Pipeline script

Script

```

1 node{
2   stage('git checkout'){
3     git 'https://github.com/DeepakKumar094/insurance-project.git'
4   }
5   stage('build'){
6     sh "mvn clean package"
7   }
8   stage("publishing the report"){
9   }

```

Use Groovy Sandbox ?

Pipeline Syntax

Save **Apply**

REST API Jenkins 2.414.3



Jar/Artifacts build successfully.

EC2 | ap-south-1 | Insure-me | General | Maven Repository | WhatsApp | Project_Insurance | Google Meet | Downloads | Insurance Domain | insurance #17 | +

Not secure | 13.235.87.142:8080/job/insurance/17/console

Q2 Gmail YouTube Maps pmsonline.bih.nic.in... AllMoviesHub - 300... Candidate Home trellix DeepakKumar094/T... Resume Editor

Dashboard > insurance > #17

```

[0;34m[INFO][m] Using 'UTF-8' encoding to copy filtered resources.
[0;34m[INFO][m] Using 'UTF-8' encoding to copy filtered properties files.
[0;34m[INFO][m] skip non existing resourceDirectory /var/lib/jenkins/workspace/insurance/src/test/resources
[0;34m[INFO][m]
[0;34m[INFO][m] @[0;32mmaven-compiler-plugin:3.10.1:testCompile@[m @ [36minsure-me@[0;1m ---@[m
[0;34m[INFO][m] Changes detected - recompiling the module!
[0;34m[INFO][m] Compiling 1 source file to /var/lib/jenkins/workspace/insurance/target/test-classes
[0;34m[INFO][m]
[0;34m[INFO][m] @[0;32mmaven-surefire-plugin:2.22.2:test@[m @ [36minsure-me@[0;1m ---@[m
[0;34m[INFO][m]
[0;34m[INFO][m] -----
[0;34m[INFO][m] T E S T S
[0;34m[INFO][m] -----
[0;34m[INFO][m] Running com.project.staragile.insureme.@[1mInsureMeApplicationTests@[m
[0;34m[INFO][m] @[0;1mTests run: 0, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.372 s - in com.project.staragile.insureme.@[1mInsureMeApplicationTests@[m
[0;34m[INFO][m]
[0;34m[INFO][m] Results:
[0;34m[INFO][m]
[0;34m[INFO][m] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[0;34m[INFO][m]
[0;34m[INFO][m] @[0;32mmaven-jar-plugin:3.2.2:jar@[m @ [1m(default-jar)@[m @ [36minsure-me@[0;1m ---@[m
[0;34m[INFO][m] Building jar: /var/lib/jenkins/workspace/insurance/target/insure-me-1.0.jar
[0;34m[INFO][m]
[0;34m[INFO][m] @[0;32mspring-boot-maven-plugin:2.7.4:repackage@[m @ [1m(repackage)@[m @ [36minsure-me@[0;1m ---@[m
[0;34m[INFO][m] Replacing main artifact with repackaged archive
[0;34m[INFO][m] @[1m-----@[m
[0;34m[INFO][m] @[1;32mBUILD SUCCESSFUL@m
[0;34m[INFO][m] @[1m-----@[m
[0;34m[INFO][m] Total time: 5.949 s
[0;34m[INFO][m] Finished at: 2023-10-28T13:09:42Z
[0;34m[INFO][m] @[1m-----@[m
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (publishing the report)
[Pipeline] publishHTML
[htmlpublisher] Archiving HTML reports...
[htmlpublisher] Archiving at PROJECT level /var/lib/jenkins/workspace/insurance/target/surefire-reports to /var/lib/jenkins/jobs/insurance/htmlreports/HTML_20Report
[Pipeline]

```

28°C Haze

Search

ENG IN 00:17 30-10-2023

3.Stage 3: Publish HTML reports.

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

The screenshot shows the Jenkins Pipeline Syntax configuration page. In the 'Steps' section, a 'Sample Step' is selected: 'publishHTML: Publish HTML reports'. The configuration fields for this step include:

- HTML directory to archive: '/var/lib/jenkins/workspace/insurance/target/surefire-reports'
- Index page(s): 'index.html'
- Index page title(s) (Optional): ''
- Report title: 'HTML Report'
- Publishing options: 'true'

At the bottom, a 'Generate Pipeline Script' button is visible, which generates the following Groovy script:

```
publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles: 'index.html', reportName: 'HTML Report', reportTitles: '', useWrapperFileDirectly: true])
```

The screenshot shows the Jenkins Pipeline configuration page. In the 'Pipeline' section, the 'Definition' is set to 'Pipeline script' with the following Groovy script:

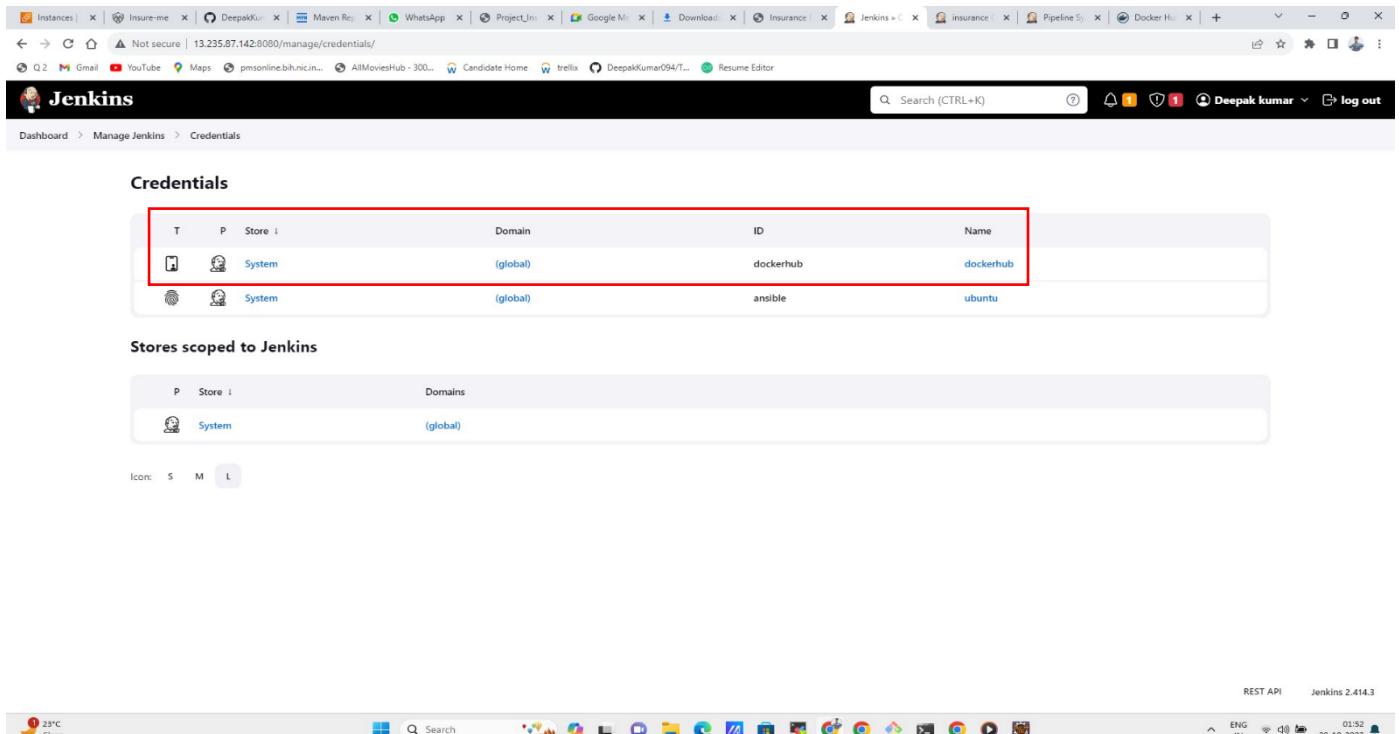
```
1 - node{
2 -   stage('git checkout'){
3 -     git 'https://github.com/DeepakKumar094/insurance-project.git'
4 -   }
5 -   stage('build'){
6 -     sh 'mvn clean package'
7 -   }
8 -   stage('publishing the report'){
9 -     publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles: 'index.html', reportName: 'HTML Report', reportTitles: '', useWrapperFileDirectly: true])
10  }
11 }
```

Below the script, there is a checkbox for 'Use Groovy Sandbox' which is checked. At the bottom, there are 'Save' and 'Apply' buttons.

Docker - For deploying containerized applications

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

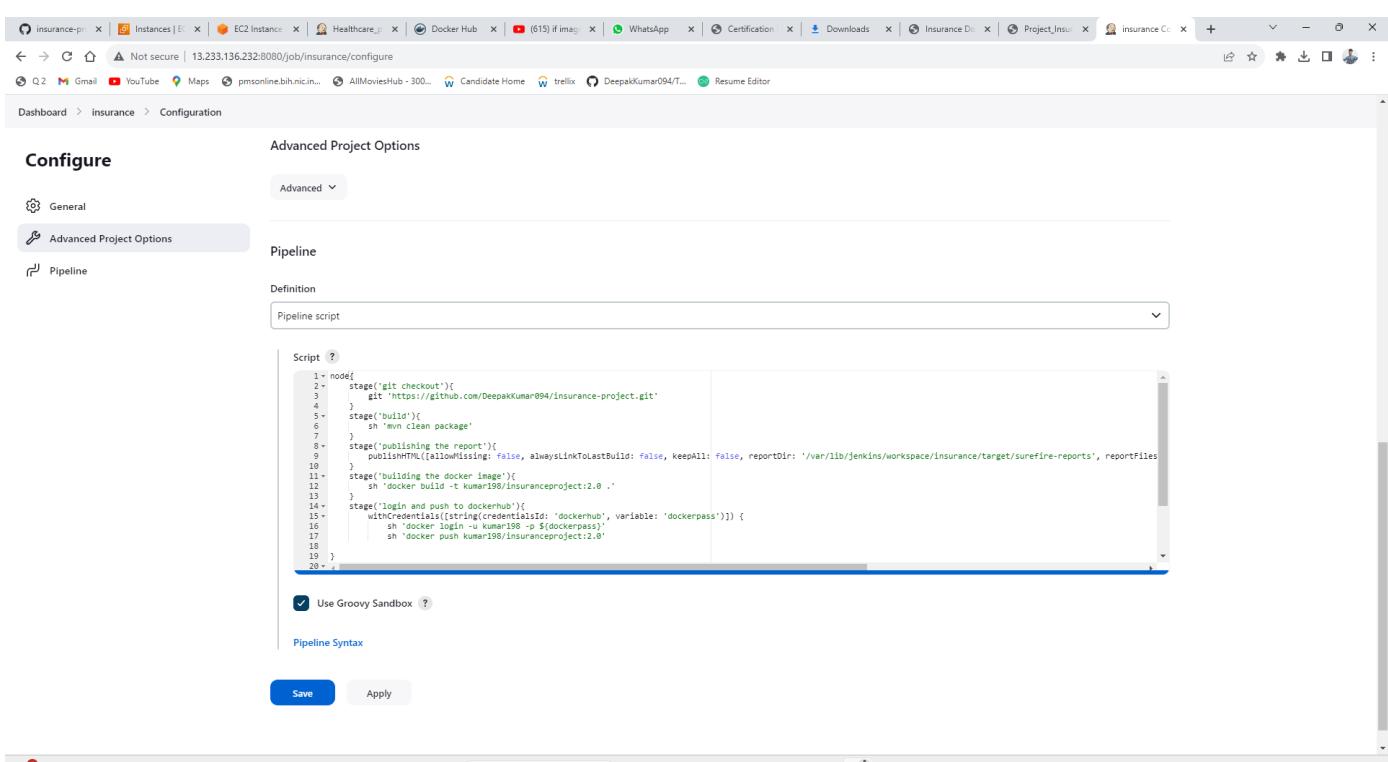
4.Stage 4: Build docker image, login and publish image to Docker Hub.



The screenshot shows the Jenkins 'Credentials' management interface. A red box highlights the first row of the table, which contains the credentials for 'dockerhub'. The table has columns for Type (T), Password (P), Store (S), Domain, ID, and Name. The 'dockerhub' entry has 'System' under Type, '(global)' under Domain, 'dockerhub' under ID, and 'dockerhub' under Name. The second row, which is not highlighted, shows 'ansible' with 'ubuntu' as its name.

Type	Password	Store	Domain	ID	Name
System			(global)	dockerhub	dockerhub
System			(global)	ansible	ubuntu

Icon: S M L REST API Jenkins 2.414.3



The screenshot shows the Jenkins Pipeline configuration page for the 'insurance' project. The 'Advanced Project Options' tab is selected. Under the 'Pipeline' section, the 'Definition' dropdown is set to 'Pipeline script'. The script content is a Groovy pipeline script:

```
node{
    stage('git checkout'){
        git 'https://github.com/DeepakKumar994/insurance-project.git'
    }
    stage('build'){
        sh 'mvn clean package'
    }
    stage('publishing the report'){
        publishHTML(isAlwaysMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles
    }
    stage('building the docker image'){
        sh 'docker build -t kumar198/insuranceproject:2.0 .'
    }
    stage('login and push to dockerhub'){
        withCredentials([string(credentialsId: 'dockermu', variable: 'dockerpass')]) {
            sh "echo $dockerpass | docker login -u kumar198 -p $dockerpass"
            sh 'docker push kumar198/insuranceproject:2.0'
        }
    }
}
```

At the bottom, there is a checked checkbox for 'Use Groovy Sandbox'.

Icon: 25°C Clear ENG IN 22:08 30-10-2023

Insurance

Terminal Sessions View Xserver Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect... 2 Master 3 Node Insurance

```
root@ip-172-31-2-230:~# docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
kumar198/insuranceproject  2.0        0ff4eb154dd1  30 hours ago  695MB
kumar198/insuranceproject <none>    6e0f780265642  30 hours ago  695MB
kumar198/insuranceproject  1.0        bd1f021fa490  30 hours ago  695MB
kumar198/insuranceproject <none>    a52adbac4d04  30 hours ago  695MB
kumar198/insuranceproject <none>    80453a1e755c  30 hours ago  695MB
kumar198/insuranceproject <none>    e76f79a9d25c  30 hours ago  695MB
kumar198/insuranceproject <none>    f8de8b44192e  45 hours ago  695MB
kumar198/insuranceproject <none>    4ced361054a  45 hours ago  695MB
kumar198/insuranceproject <none>    20c0688e2aa4  45 hours ago  695MB
kumar198/insuranceproject <none>    05fe6b1ec4c6  46 hours ago  695MB
<none>                <none>    01ed6b283712  47 hours ago  695MB
<none>                <none>    82e619f67609  2 days ago   695MB
<none>                <none>    fd56cedd078e  2 days ago   695MB
root@ip-172-31-2-230:~#
```

Remote monitoring Follow terminal folder

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

28°C Haze

Search

ENG IN 00:19 30-10-2023

Instances | EC2 | vikulrepo/insu | WhatsApp | Installation-gui | Training_Docu | Docker Hub | insurance [Jen] | Pipeline Syntax | Pipeline Syntax | Settings - Auto | Google Passw | +

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

[dockerhub](#) Search Docker Hub Ctrl+K Explore Repositories Organizations Help Upgrade kumar198

kumar198 / insuranceproject Contains: Image | Last pushed: a few seconds ago Inactive ⚡ 0 🔍 0 Public

kumar198 / insurance-image Contains: Image | Last pushed: 8 hours ago Inactive ⚡ 0 🔍 4 Public

kumar198 / loksai-eta-app Contains: Image | Last pushed: 9 hours ago Inactive ⚡ 0 🔍 4 Public

kumar198 / demo Contains: Image | Last pushed: 10 days ago Inactive ⚡ 0 🔍 6 Public

kumar198 / addressbook123 Contains: Image | Last pushed: 11 days ago Inactive ⚡ 0 🔍 3 Public

kumar198 / debian Contains: Image | Last pushed: 14 days ago Inactive ⚡ 0 🔍 1 Public

kumar198 / deepak-eta-app Contains: No content | Last pushed: 15 days ago Inactive ⚡ 0 🔍 0 Public

Create repository

Create an Organization Manage Docker Hub repositories with your team

Community All-Hands On-Demand

All sessions from our 6th Community All-Hands are now available on-demand! Over 35 talks cover best practices, demos, open source, product updates,

23°C Clear

Search

ENG IN 02:08 28-10-2023

The screenshot shows a web browser window with multiple tabs open. The active tab is for the Docker Hub repository `kumar198/insuranceproject`. The page content includes:

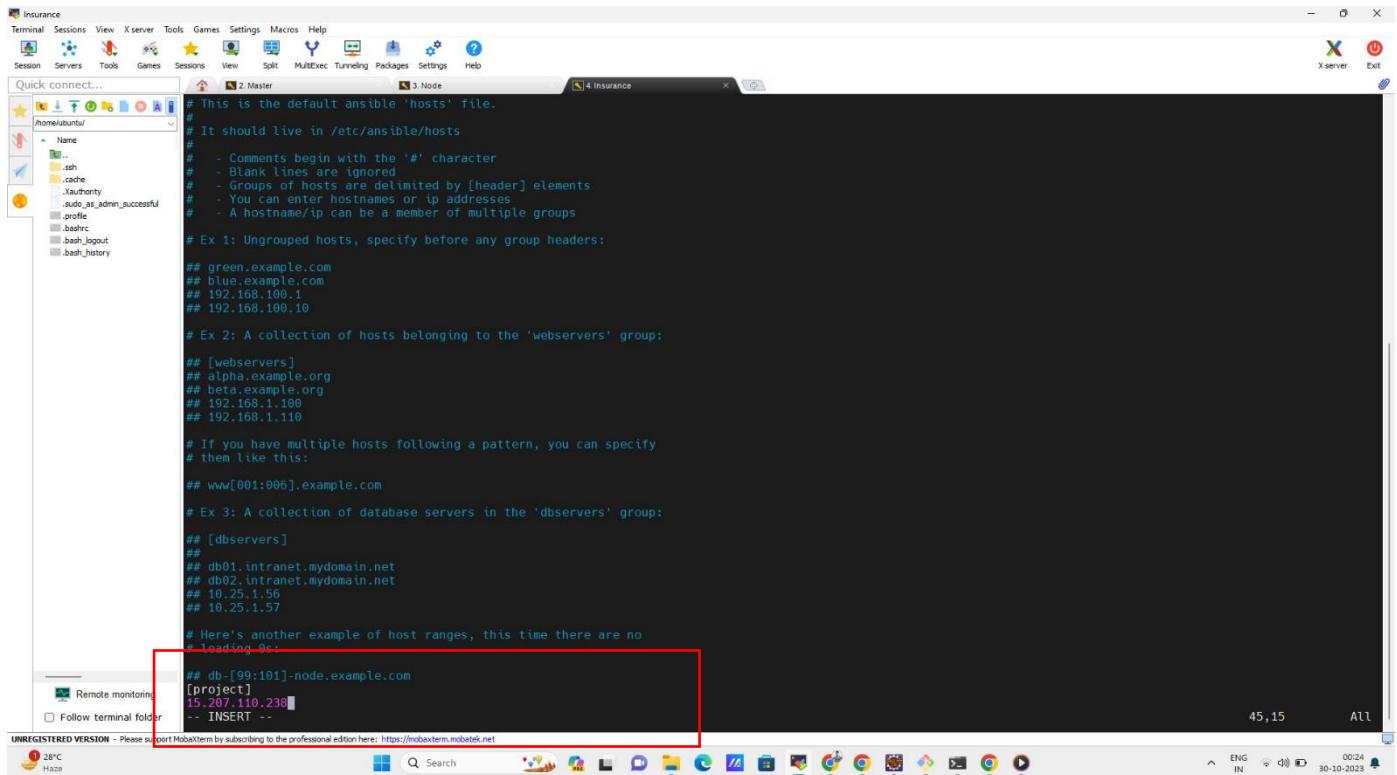
- General Information:** Shows the repository name `kumar198/insuranceproject`, a short description placeholder, and a note about the last push being a day ago.
- Docker commands:** A section for pushing new tags with the command `docker push kumar198/insuranceproject:tagname`.
- Tags:** A table showing two tags: `2.0` and `1.0`, both of which are Image types and were pushed and pulled a day ago.
- Automated Builds:** A section explaining how manually pushing images to Hub can trigger automatic builds and tagging.
- Repository overview:** A brief description of what the image does and how to run it, linking to the public view.

The browser taskbar at the bottom displays various system and application icons, along with the date and time (30-10-2023) and battery status.

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



The screenshot shows a terminal window titled "Insurance" with four tabs open: "Master", "Node", "Node", and "Insurance". The "Master" tab displays the contents of the Ansible hosts file, which includes comments about host syntax and examples of ungrouped hosts and grouped hosts under [webservers] and [dbservers] sections. The "Insurance" tab shows the output of a "ping" command from a root shell on the test server to the localhost of the master server. The command was run as "ansible -m ping localhost | SUCCESS => {". The output shows a "changed": false result and a "ping": "pong" message. The terminal interface includes a sidebar with session and server icons, and a status bar at the bottom.

```
# 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
## 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
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# gaps between the numbers:
## db-[00:101]-node.example.com
## [project]
## 15.207.110.238
-- INSERT --
```

root@ip-172-31-2-230:/home/ubuntu# cd /home/ubuntu
root@ip-172-31-2-230:/home/ubuntu# ansible -m ping localhost
localhost | SUCCESS => {
 "changed": false,
 "ping": "pong"
}
root@ip-172-31-2-230:/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 'Tools' configuration page. Under the 'Ansible installations' section, there is a form to add a new Ansible tool named 'ansible'. The 'Install automatically' checkbox is checked, and the 'Add Installer' button is visible. Below the form are 'Save' and 'Apply' buttons. The top navigation bar shows various Jenkins jobs and a 'Dashboard' link.

Create the syntax pipeline for Ansible.

The screenshot shows the Jenkins Pipeline Syntax configuration page. On the left, there is a sidebar with links to 'Global Variables Reference', 'Online Documentation', 'Examples Reference', and 'IntelliJ IDEA GDSDL'. The main area is titled 'Sample Step' and contains a dropdown menu set to 'ansiblePlaybook: Invoke an ansible playbook'. Below this are several input fields: 'Ansible tool' (set to 'ansible'), 'Playbook file path in workspace' (set to 'ansible-playbook.yml'), 'Inventory file path in workspace' (set to '/etc/ansible/hosts'), 'SSH connection credentials' (set to 'ubuntu'), 'Vault credentials' (set to '- none -'), and 'Vault tmp path' (empty). A checkbox for 'Use become' is checked. The bottom of the screen shows the Windows taskbar with various application icons.

Instances | Insure-me | General | Maven Repo | WhatsApp | Project_Insu | Google Meet | Downloads | Insurance Do | Pipeline Syntax | Docker Hub

← → ⌛ ⌂ ⌂ Not secure | 13.235.87.142:8080/job/insurance/pipeline-syntax/

Q 2 Gmail YouTube Maps pmsonline.bih.nic.in... AllMoviesHub - 300... Candidate Home trillix DeepakKumar094/T... Resume Editor

Dashboard > insurance > Pipeline Syntax

Tags to skip

Task to start at

Number of parallel processes to use

Disable the host SSH key check

Colorized output

Extra parameters

Generate Pipeline Script

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

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.



Instances | EC2 Instance | Healthcare | Docker Hub | (615) if imag | WhatsApp | Certification | Downloads | Insurance Do | Project_Insu | insurance Co

← → ⌛ ⌂ ⌂ Not secure | 13.233.136.232:8080/job/insurance/configure

Q 2 Gmail YouTube Maps pmsonline.bih.nic.in... AllMoviesHub - 300... Candidate Home trillix DeepakKumar094/T... Resume Editor

Dashboard > insurance > Configuration

Configure

Advanced Project Options

General

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script

Script

```
1 node{
2     stage('git checkout'){
3         git 'https://github.com/DeepakKumar094/insurance-project.git'
4     }
5     stage('build'){
6         sh 'mvn clean package'
7     }
8     stage('publishing the report'){
9         publishHTML(alwaysMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/insurance/target/surefire-reports', reportFiles:
10             stage('building the docker image'){
11                 sh 'docker build -t kumar198/insuranceproject:2.0 .'
12             }
13             stage('login and push to dockerhub'){
14                 withCredentials([string(credentialsId: 'dockernhub', variable: 'dockerpass')]) {
15                     sh 'docker login -u kumar198 -p ${dockerpass}'
16                     sh 'docker push kumar198/insuranceproject:2.0'
17                 }
18             }
19             stage('deploy to the test server'){
20                 ansiblePlayBook become: true, credentialsId: 'ansible', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/hosts', playbook: 'ansible-playbook.yml', vaultTmpPath: ''
21             }
22         }
23     }
24 }
```

Use Groovy Sandbox

Save **Apply**



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

The screenshot shows a GitHub repository page for 'insurance-project'. The left sidebar lists various files, with 'ansible-playbook.yml' selected. The main content area displays the contents of 'ansible-playbook.yml' as follows:

```
1  - name : Configure Docker on EC2 Instances
2  hosts : all
3  become: true
4  connection : ssh
5  tasks :
6    - name: updating apt
7      command : sudo apt-get update
8
9    - name : Install Docker
10   command : sudo apt-get install -y docker.io
11   become : yes
12   become_user : root
13
14   - name : Start Docker Service
15     command : sudo systemctl start docker
16     become : yes
17     become_user : root
18   - name : stop existing containers
19     shell: docker stop $(docker ps -a -q)
20   - name: delete all containers
21     shell: docker rm $(docker ps -a -q)
22   - name: Deploy Docker Container
23     command: docker run -itd -p 8080:8081 kumar198/insuranceproject:1.0
```

I deployed my application to this test-server.

The screenshot shows the AWS EC2 Instances page. It displays a table of instances with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
Project_insura...	i-0355f61956ef13f21	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-136-232.ap...	13.233.136.232	-
project_insura...	i-02b078247de4f93c5	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-15-207-110-238.ap...	15.207.110.238	-

Project is deployed to test-server successfully

The screenshot shows the Jenkins Pipeline insurance Stage View. It displays a table of build stages and their execution times. The stages are: git checkout, build, publishing the report, building the docker image, login and push to dockerhub, and deploy to the test server. The average stage times are: git checkout (1s), build (9s), publishing the report (66ms), building the docker image (3s), login and push to dockerhub (18s), and deploy to the test server (22s). Below the table is a list of build history entries, each with a timestamp and status (e.g., #17, 28 Oct 2023, 13:09, No Changes, succeeded).

The screenshot shows the Jenkins runonslave #2 Console Output. The log output shows the pipeline starting, cloning a Git repository from https://github.com/vikulrepo/Addressbook2.git, and checking out Revision c1c3b3bd079fb41789c13c57c326f7d7d8da489c. The commit message is "Update Babufile". The log concludes with "Finished: SUCCESS".

Console Output

```
Started by user deepak kumar
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave1 in /var/lib/jenkins/workspace/runonslave
[Pipeline] {
[Pipeline] stage
[Pipeline] { (git checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  > git init /var/lib/jenkins/workspace/runonslave # timeout=10
Fetching upstream changes from https://github.com/vikulrepo/Addressbook2.git
  > git -v --version # timeout=10
  > git --version # git version 2.34.1'
  > git fetch --tags --force --progress -- https://github.com/vikulrepo/Addressbook2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config remote.origin.url https://github.com/vikulrepo/Addressbook2.git # timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision c1c3b3bd079fb41789c13c57c326f7d7d8da489c (refs/remotes/origin/master)
Commit message: "Update Babufile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```
EC2 | ap-south-1 | Not secure | 13.235.87.142:8080/job/insurance/17/console
Gmail YouTube Maps pmsonline.bih.nic.in... AllMoviesHub - 300... Candidate Home trellix DeepakKumar094/T... Resume Editor
Dashboard > insurance > #17
PLAY [Configure Docker on EC2 Instances] ****
TASK [Gathering Facts] ****
ok: [15.207.110.238]

TASK [updating apt] ****
changed: [15.207.110.238]

TASK [Install Docker] ****
changed: [15.207.110.238]

TASK [Start Docker Service] ****
changed: [15.207.110.238]

TASK [stop existing containers] ****
changed: [15.207.110.238]

TASK [delete all containers] ****
changed: [15.207.110.238]

TASK [Deploy Docker Container] ****
changed: [15.207.110.238]

PLAY RECAP ****
15.207.110.238 : ok=7    changed=6    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
```

REST API Jenkins 2.414.3

ENG IN 00:07 30-10-2023

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

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.

Contact Us

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](#)

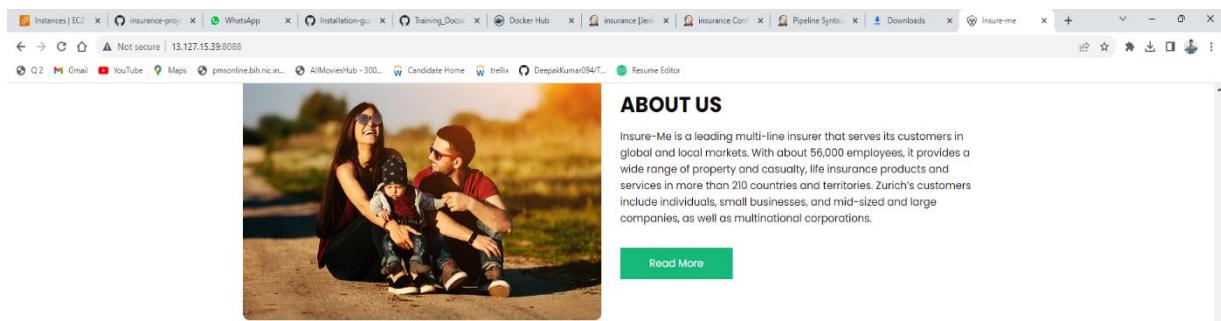
CAR INSURANCE
Car insurance, is a type of vehicle insurance policy that protects you and your car from any risks of accidents and property damages.
[Read More](#)

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

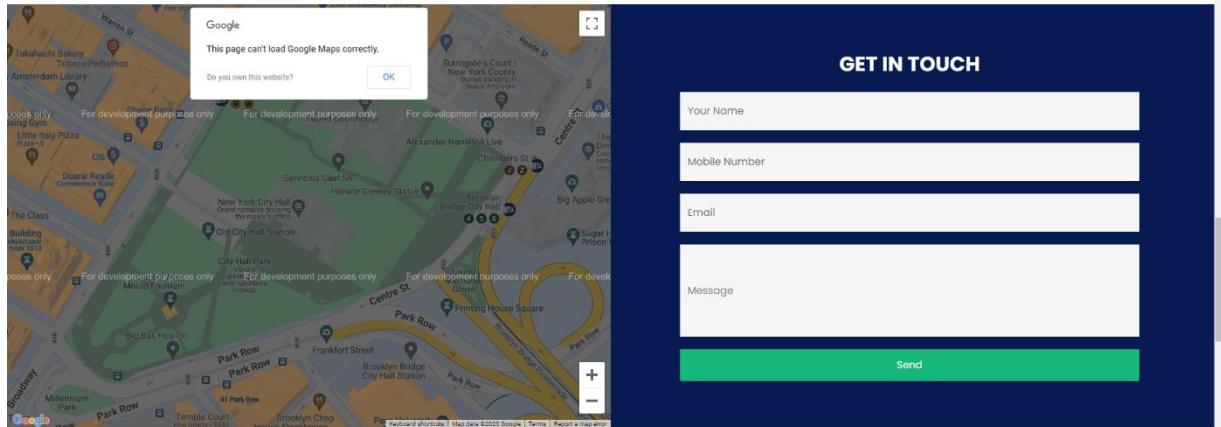
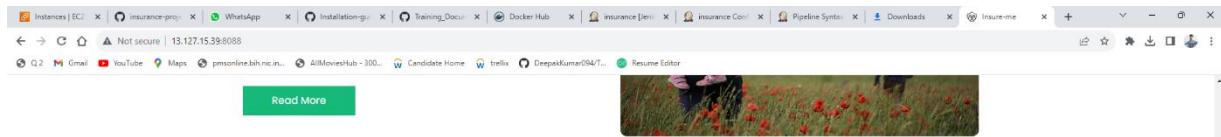
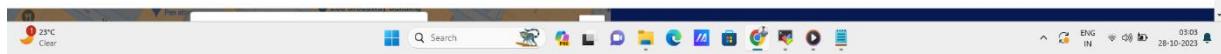
Insure-Me is a leading multi-line insurer that serves its customers in global and local markets. With about 56,000 employees, it provides a wide range of property and casualty, life insurance products and services in more than 210 countries and territories. Zurich's customers include individuals, small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



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](#)



Generate HTML Report using TestNG

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

- All suites**: Surefire suite
- Info**:
 - [unset file name]
 - 1 test
 - 0 groups
 - Times
 - Reporter output
 - Ignored methods
 - Chronological view
- Results**: 0 methods.
- Test results**: 1 suite

Details of the suite:

- Suite XML:**<xml version="1.0" encoding="UTF-8"><!DOCTYPE suite SYSTEM "https://testng.org/testng-1.8.dtd"><suite thread-count="1" name="Surefire suite" verbose="0"><test thread-count="1" name="Surefire test" verbose="0"><classes><class name="com.project.staragile.insureme.InsureMeApplicationTests"/></classes></test></-- Surefire test --></suite></-- Surefire suite -->
- Tests for Surefire suite**: Surefire test (1 class)
- Groups for Surefire suite**: None
- Times for Surefire suite**: Total running time: 0 ms
- Reporter output for Surefire suite**: None
- Ignored methods**: None
- Methods in chronological order**: None

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

The screenshot shows the Eclipse IDE interface with the pom.xml file open in the editor. The pom.xml file contains the following XML code, with a red box highlighting the Selenium dependency section:

```
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
</dependency>
```

The rest of the pom.xml file includes Maven settings, profiles, and other dependencies like Spring Boot and H2 Database.

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

```
1 package com.project.staragile.insureme;
2 import org.openqa.selenium.By;
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 import org.openqa.selenium.chrome.ChromeOptions;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.WebElement;
8 public class test {
9
10    public static void main(String[] args) {
11        System.setProperty("webdriver.chrome.driver", "C:\\webedriver\\chromedriver.exe");
12        final ChromeOptions chromeOptions = new ChromeOptions();
13        chromeOptions.addArguments("--remote-allow-origins=*", "ignore-certificate-errors");
14        //chromeOptions.addArguments(new String[] { "--headless" });
15        //chromeOptions.addArguments(new String[] { "--no-sandbox" });
16        //chromeOptions.addArguments(new String[] { "--disable-dev-shm-usage" });
17        final WebDriver driver = new ChromeDriver(chromeOptions);
18        driver.get("http://15.207.110.238:8088/contact.html");
19
20        driver.manage().timeouts().implicitlyWait(5L, TimeUnit.SECONDS);
21        driver.findElement(By.id("inputName")).sendKeys(new CharSequence[] { "Deepak" });
22        driver.findElement(By.id("inputEmail")).sendKeys(new CharSequence[] { "1234567890" });
23        driver.findElement(By.id("inputMail1")).sendKeys(new CharSequence[] { "Deepak@test.com" });
24        driver.findElement(By.id("inputPhone")).sendKeys(new CharSequence[] { "9876543210" });
25        driver.findElement(By.id("inputbutton")).click();
26
27        final String message = driver.findElement(By.id("response")).getText();
28        if (message.equals("Message Sent")) {
29            System.out.println("Script executed Successfully");
30        } else {
31            System.out.println("Script failed");
32        }
33        //Thread.sleep(3000L);
34        //driver.quit();
35    }
36}
37 }
```

```
<terminated> test [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (29-Oct-2023, 3:26:10 pm - 3:26:19 pm) [pid: 10112]
Cache-Control: no-cache
Content-Type: application/json; charset=utf-8
host: localhost:61515
accept: */*

Response DefaultHttpResponse(decodeResult: success, version: HTTP/1.1)
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
cache-control: no-cache
content-length: 24

15:26:19.167 [AsyncHttpClient-1-2] DEBUG org.asynchttpclient.netty.channel.ChannelManager - Adding key: http://localhost:61515 for channel [id: 0xb52b953b, L:/127.0.0.1:52410 - R:localhost/127.0.0.1:61515]
Script executed Successfully
```

The screenshot shows a web page titled "INSURE-ME". On the left, there is a map of New York City with several "For development purposes only" markers. A Google Maps error message box is overlaid on the map, stating "This page can't load Google Maps correctly." Below the map, there is a "GET IN TOUCH" section containing a form with fields for name ("Deepak"), phone number ("1234567890"), email ("Deepak@test.com"), and a message area ("Welcome to the DevOps session"). At the bottom, there are sections for "INSURE-ME" social media links, "CONTACT US" location sharing, and a "SIGN UP TO OUR NEWSLETTER" form.

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

The screenshot shows the GitHub "Webhooks" settings page for a repository named "DeepakKumar094 / insurance-project". The "Webhooks" section is highlighted with a red box. It contains a table with one row, showing a webhook URL: "http://13.235.87.142:8080/github-w... (push)". The "Edit" and "Delete" buttons are visible next to the URL. The sidebar on the left lists other settings like General, Access, Collaborators, and Webhooks.

The screenshot shows the Jenkins configuration page for a project named 'insurance'. The 'Configure' screen is displayed, with the 'General' tab selected. In the 'Build Triggers' section, there is a checkbox for 'GitHub hook trigger for GITScm polling', which is checked and highlighted with a red box. Other options like 'Build periodically' and 'Poll SCM' are also present but not selected.

Push your code into your GitHub Repository.

Push the code to GitHub

The screenshot shows the 'Push Branch master' dialog from GitHub. The title bar says 'Push Branch master'. The main heading is 'Push to branch in remote'. Below it, it says 'Select a remote and the name the branch should have in the remote.' On the right, there is an icon of a yellow cloud with a red arrow pointing up. The 'Source:' section shows a local repository with a 'master' branch and a commit hash '475f54e FIRST COMMIT'. The 'Destination:' section shows a remote repository 'origin: https://github.com/DeepakKumar094/star-agile-insurance-project.git' and a branch 'master'. A checkbox 'Configure upstream for push and pull' is checked. Below it, a dropdown 'When pulling:' is set to 'Merge'. There is also an unchecked checkbox 'Force overwrite branch in remote if it exists and has diverged'. At the bottom, there are buttons for '?', '< Back' (disabled), 'Preview >', 'Push', and 'Cancel'.