Certification Project – Insure Me

Domain name: Insurance project
Assignment name: **Certification Project – Insure Me:**
submitted by: **Subodh Kamble**
submitted on: 23/05/25

Overview of project : Insure Me is a Global leading Insurance provider based out of the 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 grew, 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. Ensure me has decided to use AWS as the 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.

Main Objectives : 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.

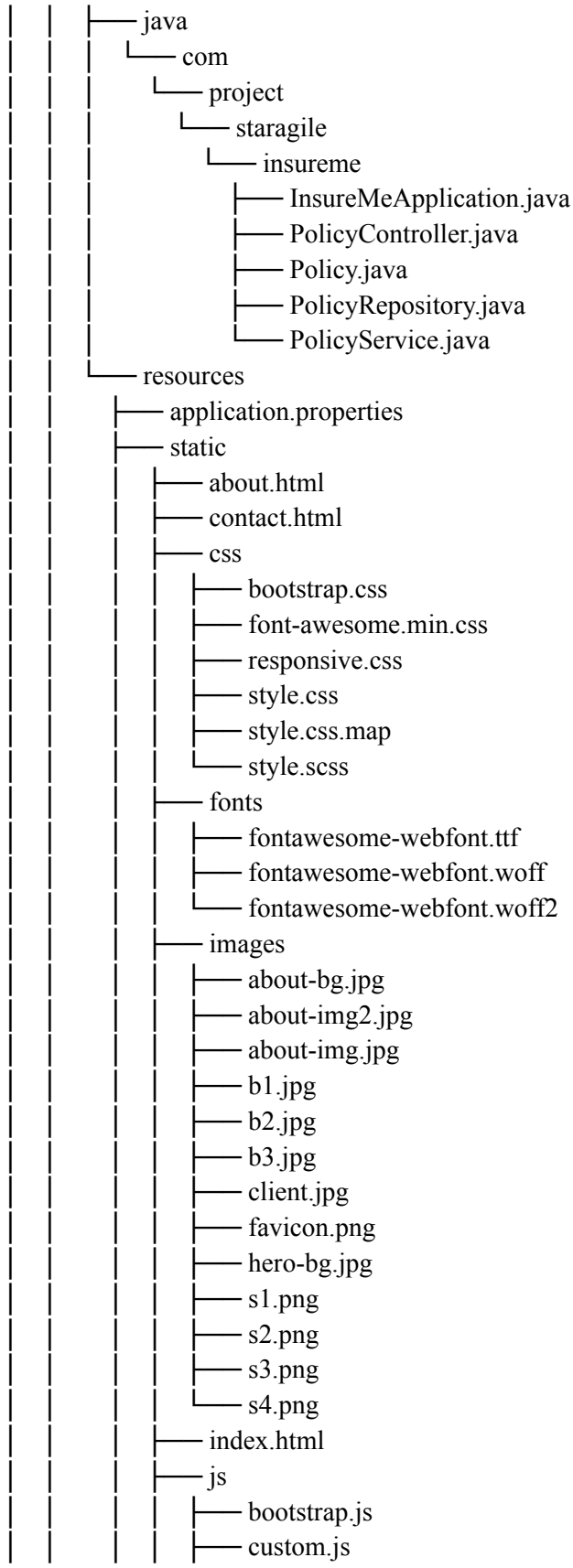
Technology used :
Git - For version control for tracking changes in the code files
Jenkins - For continuous integration and continuous deployment
Docker - For deploying containerized applications
Ansible - Configuration management tools
Terraform - for infrastructure deployment
AWS - For creating ec2 machines as servers and deploy the web application
docker-compose - For container orchestration.

Step 1. Create IAM role and access keys and secret access keys
Create a IAM user as terraform

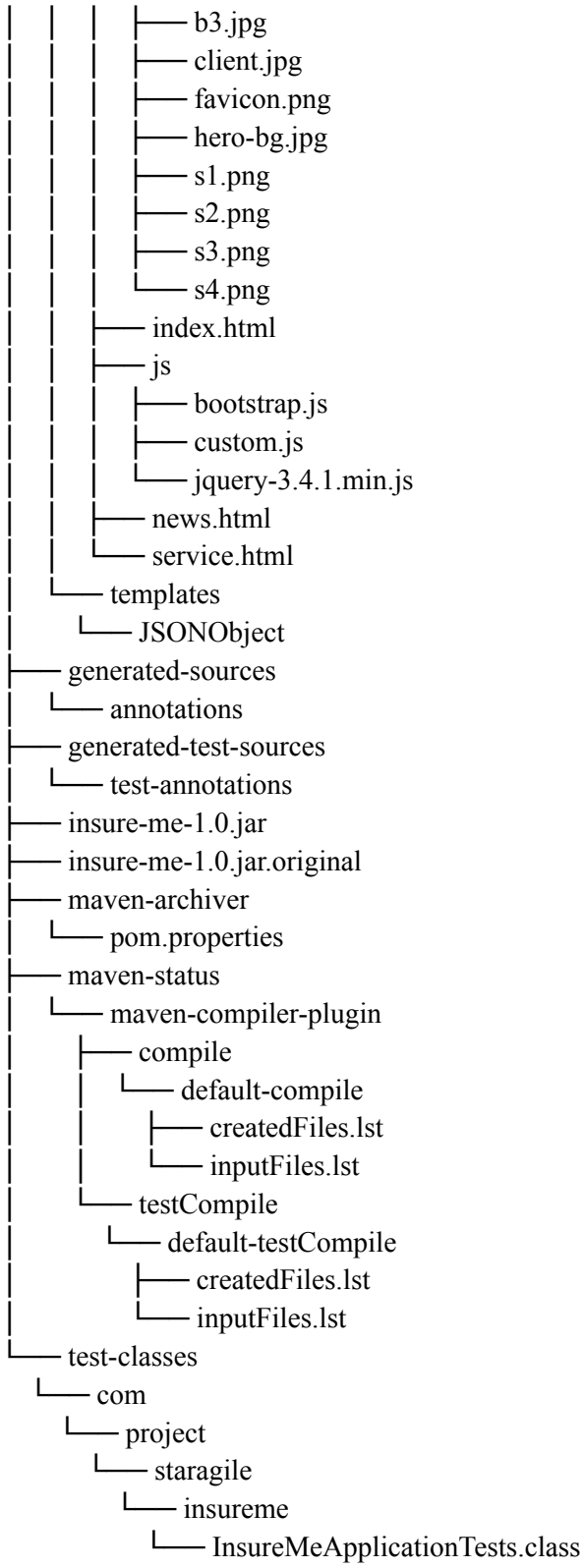
Structure of project:

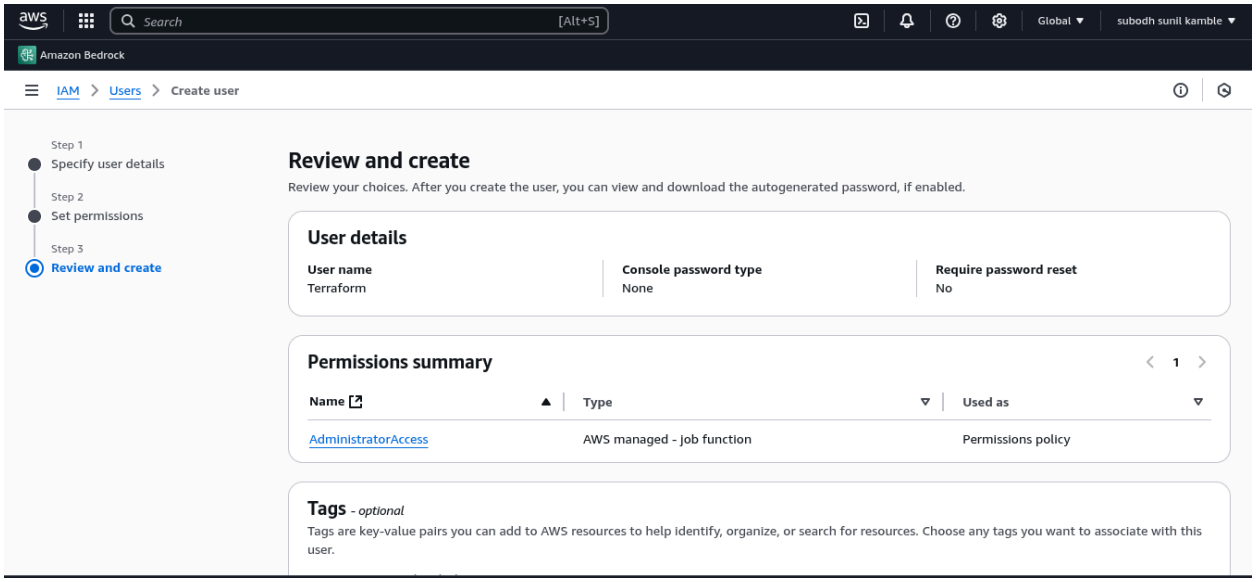
$ tree .

```
insurance_capstone/
├── infrastructure
│   ├── ansible
│   │   ├── ansible-playbook.yml
│   │   ├── docker-compose-playbook.yml
│   │   ├── docker.sh
│   │   ├── grafana.sh
│   │   ├── inventory
│   │   ├── k8s-master.sh
│   │   ├── k8s-nodes.sh
│   │   ├── test-deployment.yml
│   │   └── trivy.sh
│   ├── infra.sh
│   ├── plan.md
│   ├── README.md
│   └── terraform
│       ├── insurance-key.pem
│       ├── inventory.tmpl
│       ├── main.tf
│       ├── outputs.tf
│       ├── security.tf
│       ├── terraform.tfstate
│       ├── terraform.tfstate.backup
│       ├── terraform.tfvars
│       ├── tls.tf
│       └── variables.tf
├── Insurance Domain Project - Insure Me.pdf
├── LICENSE
├── README.md
└── star-agile-insurance-project
    ├── deployment.yml
    ├── docker-compose.yml
    ├── Dockerfile
    ├── Jenkinsfile
    ├── mvnw
    ├── mvnw.cmd
    ├── pom.xml
    ├── README.md
    ├── selenium-insure-me-runnable.jar
    ├── service.yml
    ├── src
    │   ├── main
```

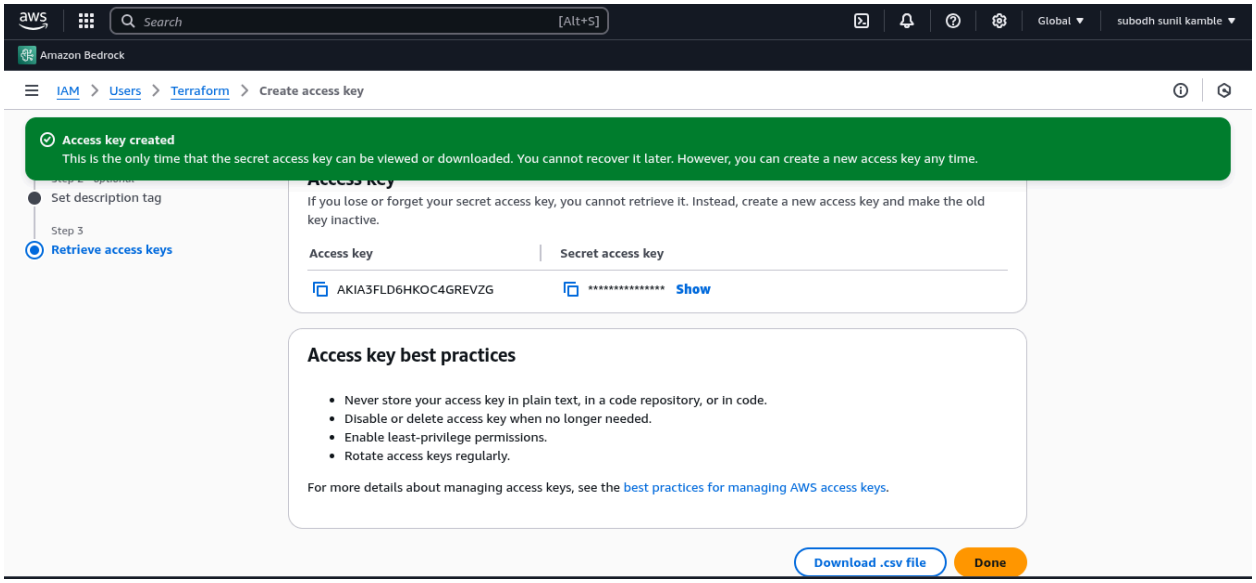

java
com
project
staragile
insureme
InsureMeApplication.java
PolicyController.java
Policy.java
PolicyRepository.java
PolicyService.java
resources
application.properties
static
about.html
contact.html
css
bootstrap.css
font-awesome.min.css
responsive.css
style.css
style.css.map
style.scss
fonts
fontawesome-webfont.ttf
fontawesome-webfont.woff
fontawesome-webfont.woff2
images
about-bg.jpg
about-img2.jpg
about-img.jpg
b1.jpg
b2.jpg
b3.jpg
client.jpg
favicon.png
hero-bg.jpg
s1.png
s2.png
s3.png
s4.png
index.html
js
bootstrap.js
custom.js

```
│   │   │   │   └── jquery-3.4.1.min.js
│   │   │   ├── news.html
│   │   │   └── service.html
│   │   └── templates
│   │       └── JSONObject
│   └── test
│       └── java
│           └── com
│               └── project
│                   └── staragile
│                       └── insureme
│                           └── InsureMeApplicationTests.java
└── target
    ├── classes
    │   ├── application.properties
    │   ├── com
    │   │   └── project
    │   │       └── staragile
    │   │           └── insureme
    │   │               ├── InsureMeApplication.class
    │   │               ├── Policy.class
    │   │               ├── PolicyController.class
    │   │               ├── PolicyRepository.class
    │   │               └── PolicyService.class
    │   ├── static
    │   │   ├── about.html
    │   │   ├── contact.html
    │   │   ├── css
    │   │   │   ├── bootstrap.css
    │   │   │   ├── font-awesome.min.css
    │   │   │   ├── responsive.css
    │   │   │   ├── style.css
    │   │   │   ├── style.css.map
    │   │   │   └── style.scss
    │   │   ├── fonts
    │   │   │   ├── fontawesome-webfont.ttf
    │   │   │   ├── fontawesome-webfont.woff
    │   │   │   └── fontawesome-webfont.woff2
    │   │   ├── images
    │   │   │   ├── about-bg.jpg
    │   │   │   ├── about-img2.jpg
    │   │   │   ├── about-img.jpg
    │   │   │   ├── b1.jpg
    │   │   │   ├── b2.jpg
```


b3.jpg
client.jpg
favicon.png
hero-bg.jpg
s1.png
s2.png
s3.png
s4.png
index.html
js
bootstrap.js
custom.js
jquery-3.4.1.min.js
news.html
service.html
templates
JSONObject
generated-sources
annotations
generated-test-sources
test-annotations
insure-me-1.0.jar
insure-me-1.0.jar.original
maven-archiver
pom.properties
maven-status
maven-compiler-plugin
compile
default-compile
createdFiles.lst
inputFiles.lst
testCompile
default-testCompile
createdFiles.lst
inputFiles.lst
test-classes
com
project
staragile
insureme
InsureMeApplicationTests.class



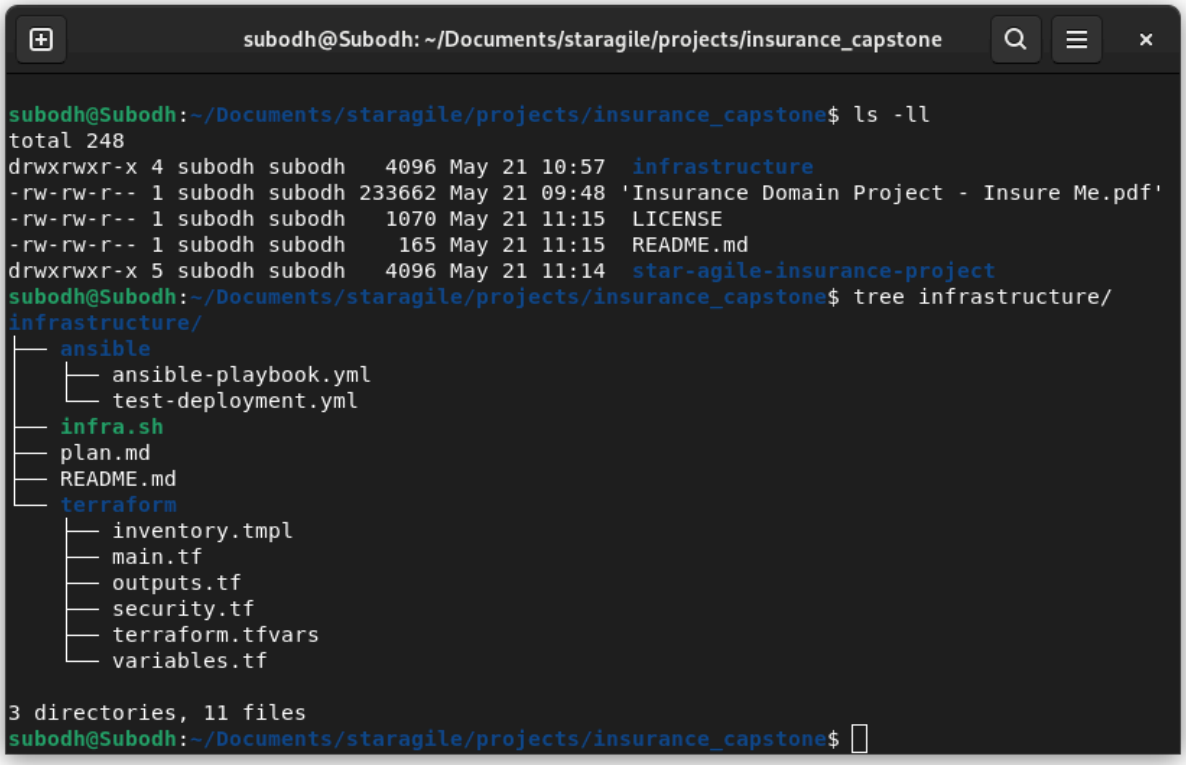
Create access keys and secret access keys



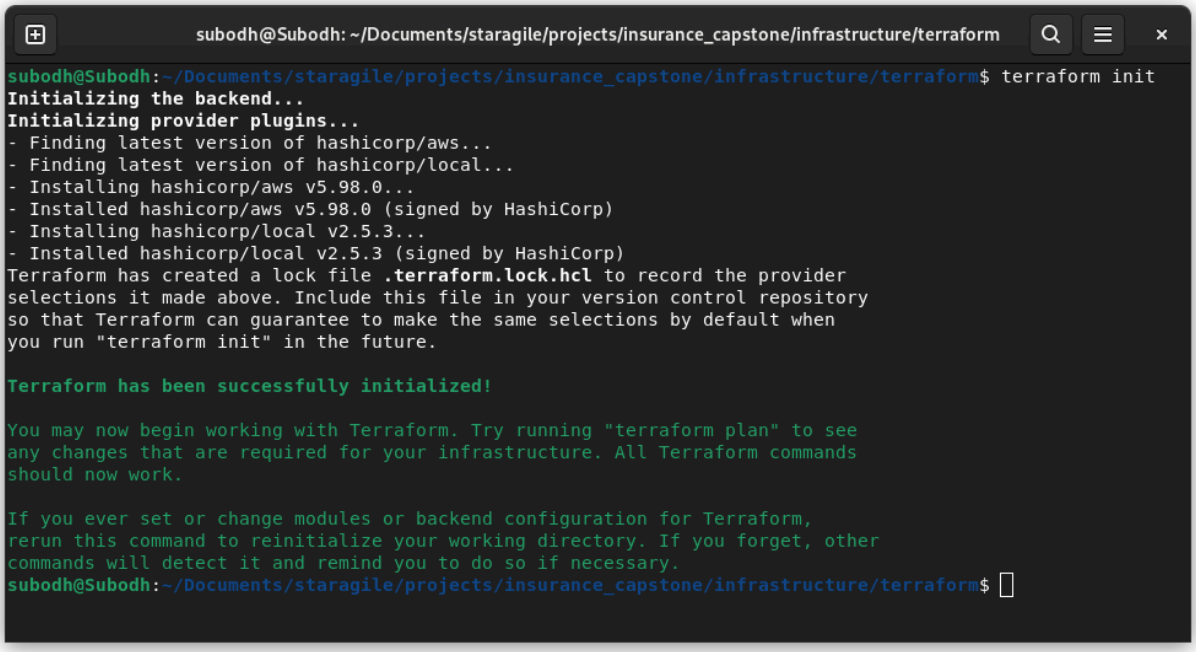
Step 2. Setup infrastructure

$ ls -ll

$ tree . infrastructure/



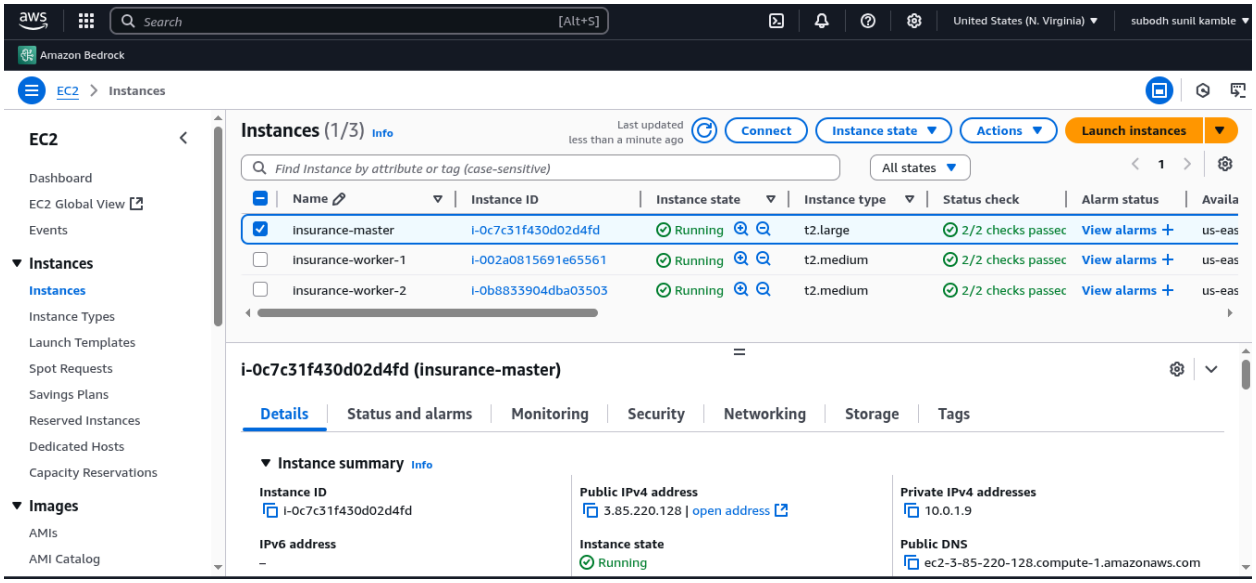
$ terraform init



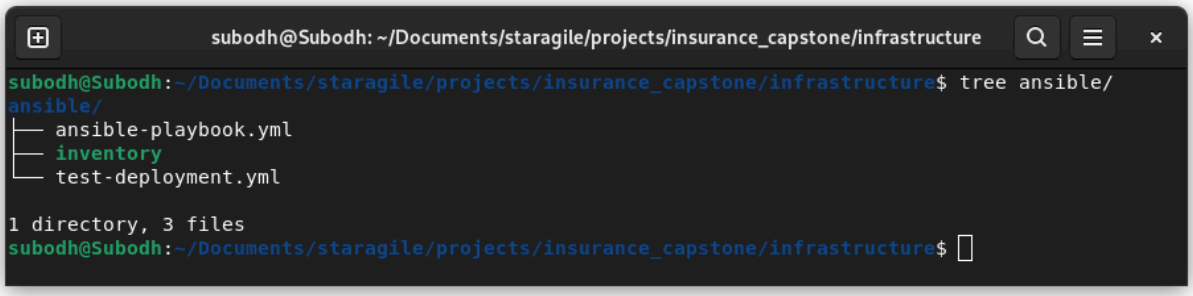
$ terraform apply –auto-approve

```
subodh@Subodh: ~/Documents/staragile/projects/insurance_capstone/infrastructure/terraform
aws_instance.master: Creating...
aws_instance.workers[0]: Still creating... [10s elapsed]
aws_instance.workers[1]: Still creating... [10s elapsed]
aws_instance.master: Still creating... [10s elapsed]
aws_instance.master: Creation complete after 15s [id=i-0a4e451919d8ddaa0]
aws_instance.workers[1]: Creation complete after 15s [id=i-04d6c24f56998975e]
aws_instance.workers[0]: Creation complete after 16s [id=i-0feda473dd91fd015]
local_file.ansible_inventory: Creating...
local_file.ansible_inventory: Creation complete after 0s [id=749b4f06d17e513c56405fd1c50ed4cfb09e4bea
]

Apply complete! Resources: 6 added, 0 changed, 5 destroyed.

Outputs:

master_private_ip = "10.0.1.247"
master_public_ip = "54.227.233.146"
worker_private_ips = [
  "10.0.1.214",
  "10.0.1.136",
]
worker_public_ips = [
  "54.198.8.233",
  "54.166.170.86",
]
subodh@Subodh:~/Documents/staragile/projects/insurance_capstone/infrastructure/terraform$
```

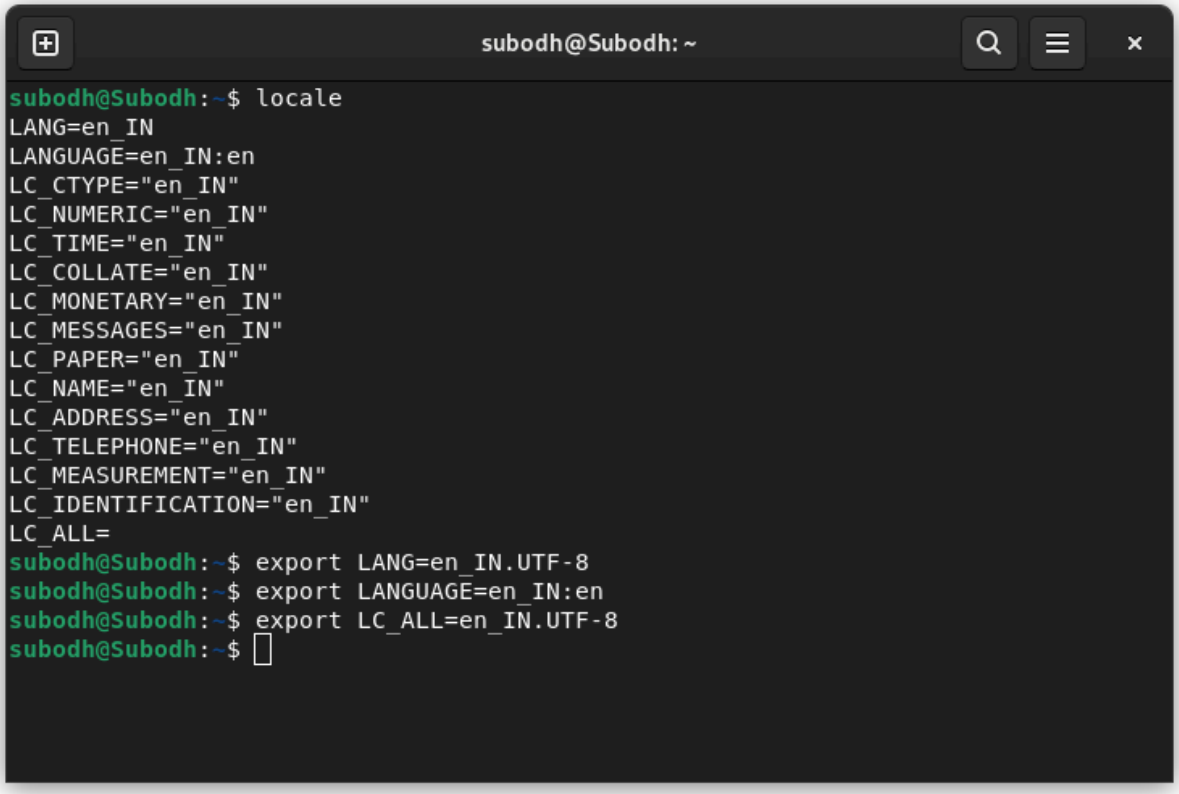
Successful execution of all the terraform script



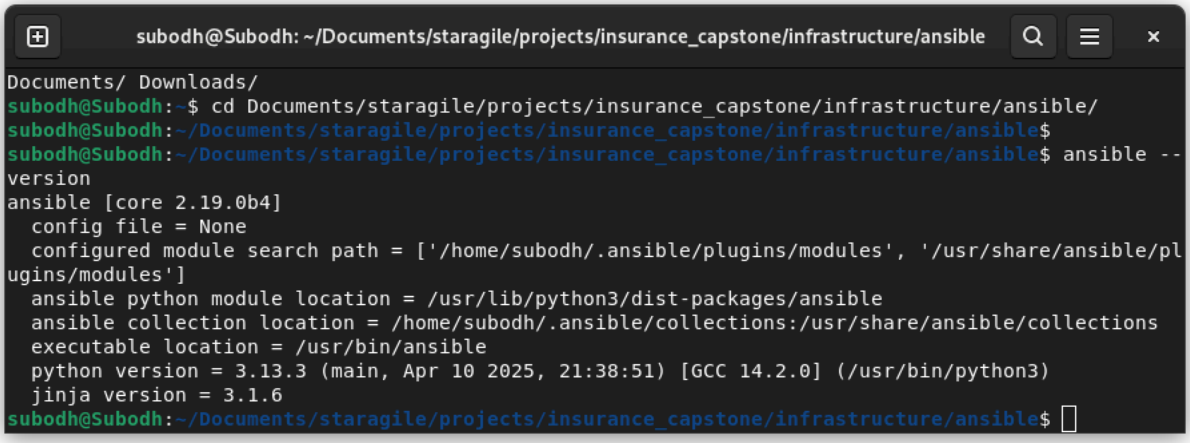
Ansible playbooks files



NOTE: if ansible error for UTF-8 will be there follow following commands
$ export LANG=en_IN.UTF-8
$ export LANGUAGE=en_IN:en
$ export LC_ALL=en_IN.UTF-8

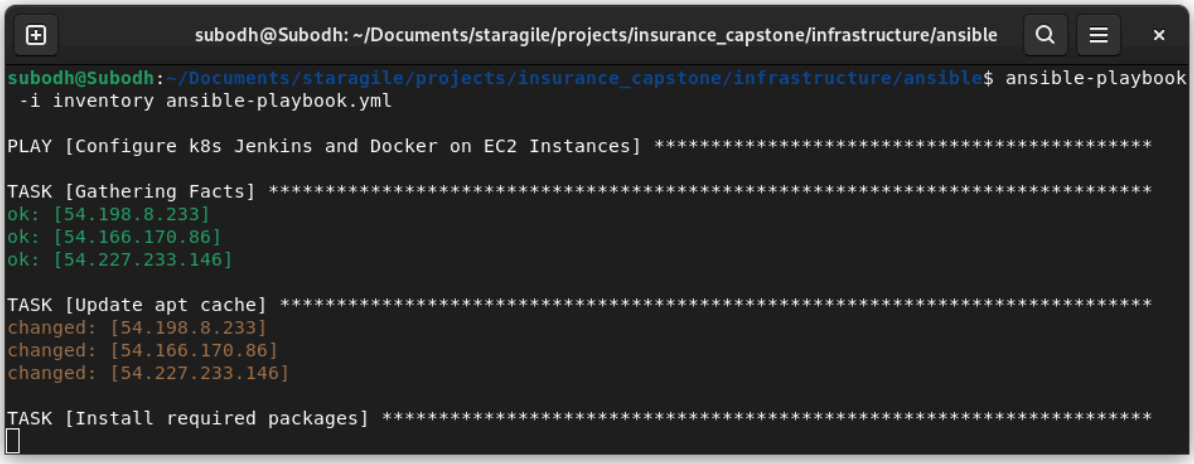


To check ansible version :
$ ansible –version



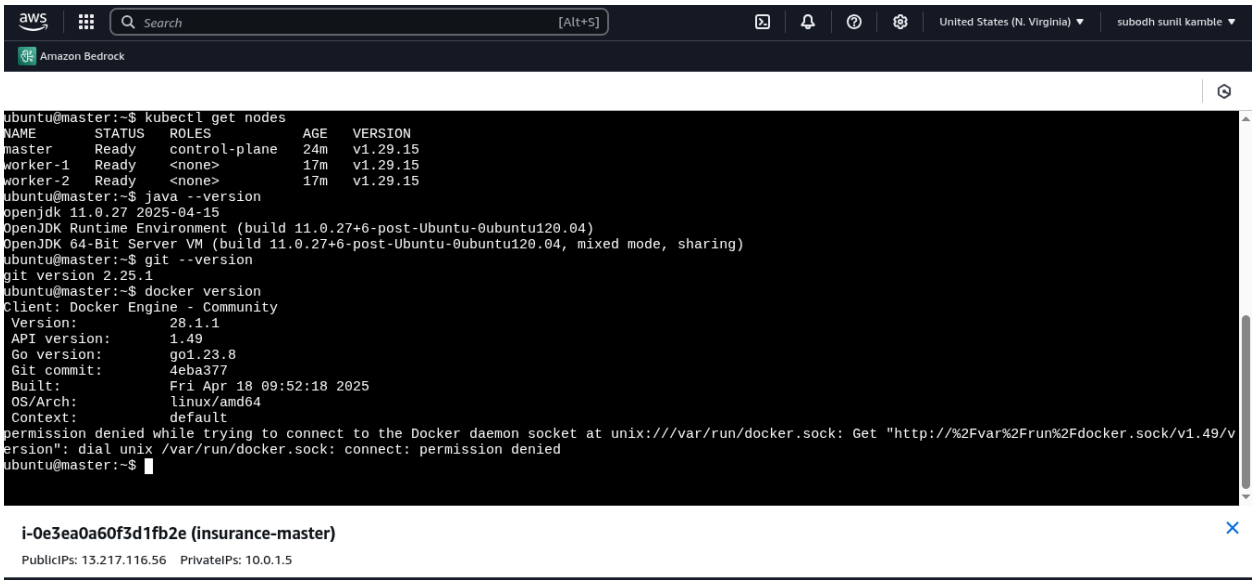
Ansible setup for keys
$ mkdir -p ~/.ssh && chmod 700 ~/.ssh && cp insurance-key.pem ~/.ssh/insurance-key.pem && chmod 400 ~/.ssh/insurance-key.pem

$ ansible-playbook -i inventory ansible-playbook.yaml



```
TASK [Gathering Facts] ********************************************************************************************
*************
ok: [23.20.184.205]
ok: [54.224.66.37]
ok: [13.217.116.56]

TASK [Install docker-compose] *************************************************************************************
*************
ok: [13.217.116.56]
changed: [23.20.184.205]
changed: [54.224.66.37]

PLAY RECAP ********************************************************************************************************
*************
13.217.116.56              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
23.20.184.205              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
54.224.66.37               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
```

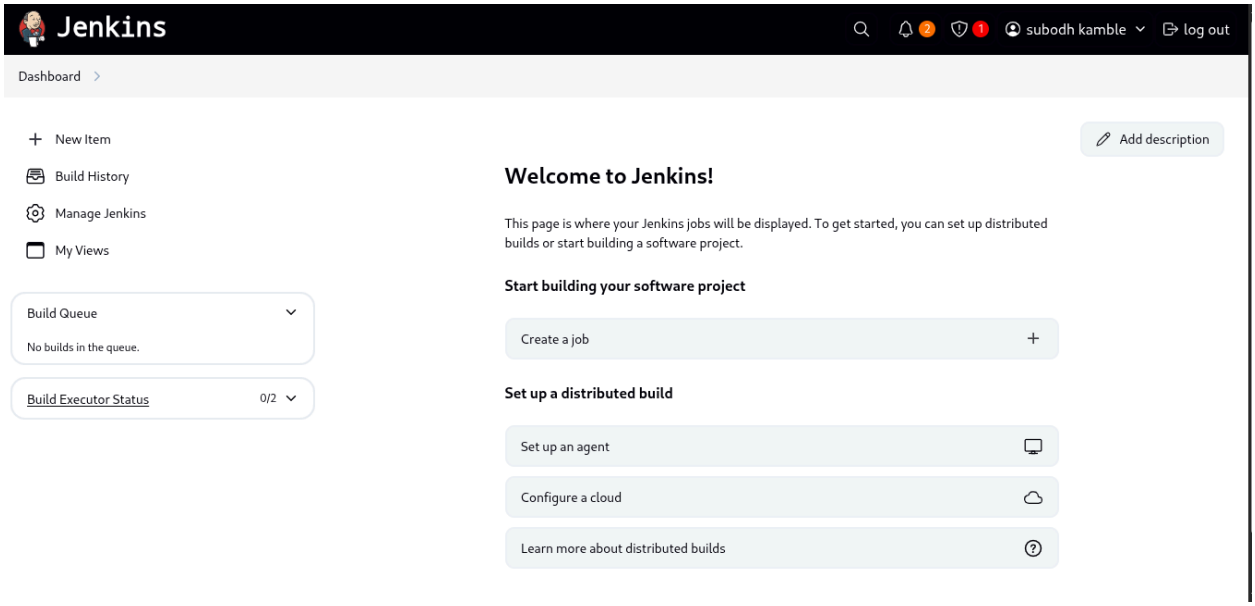
Errors need to solve manually in the server



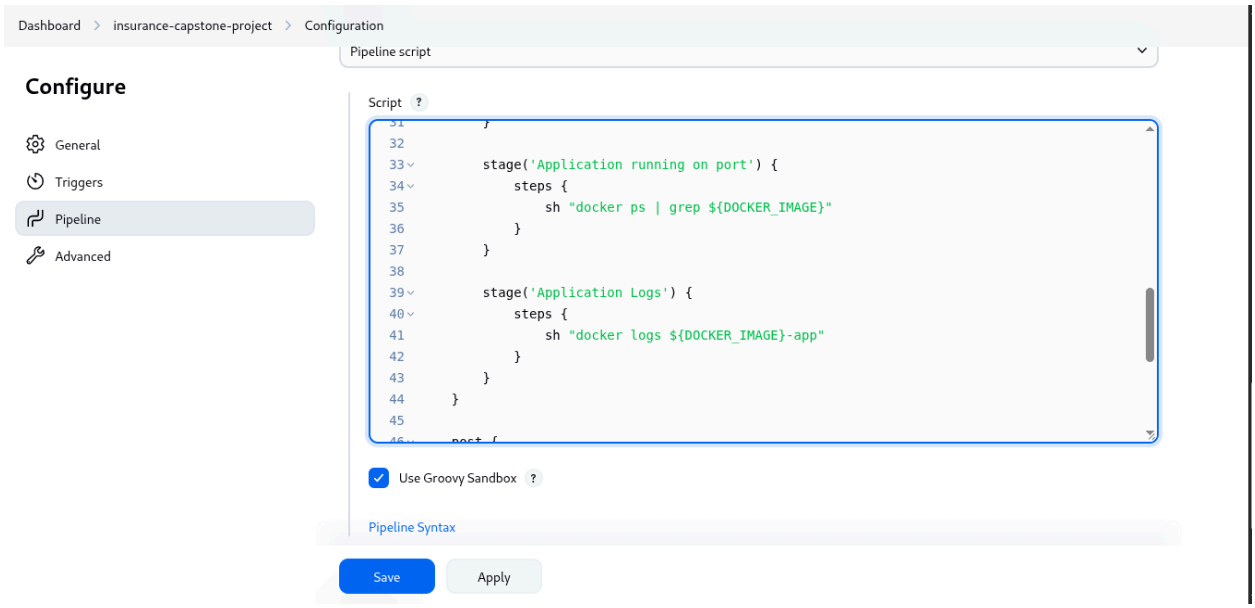
All the tools and software is installed using Ansible and is checked by printing versions and info

$ kubectl –version
$ java –version
$ git –version
$ docker version
$ docker-compose –version

Open jenkins on ip : <public ip>:8080
Setup username and other account settings and install suggested plugins



Create a new jenkins pipeline

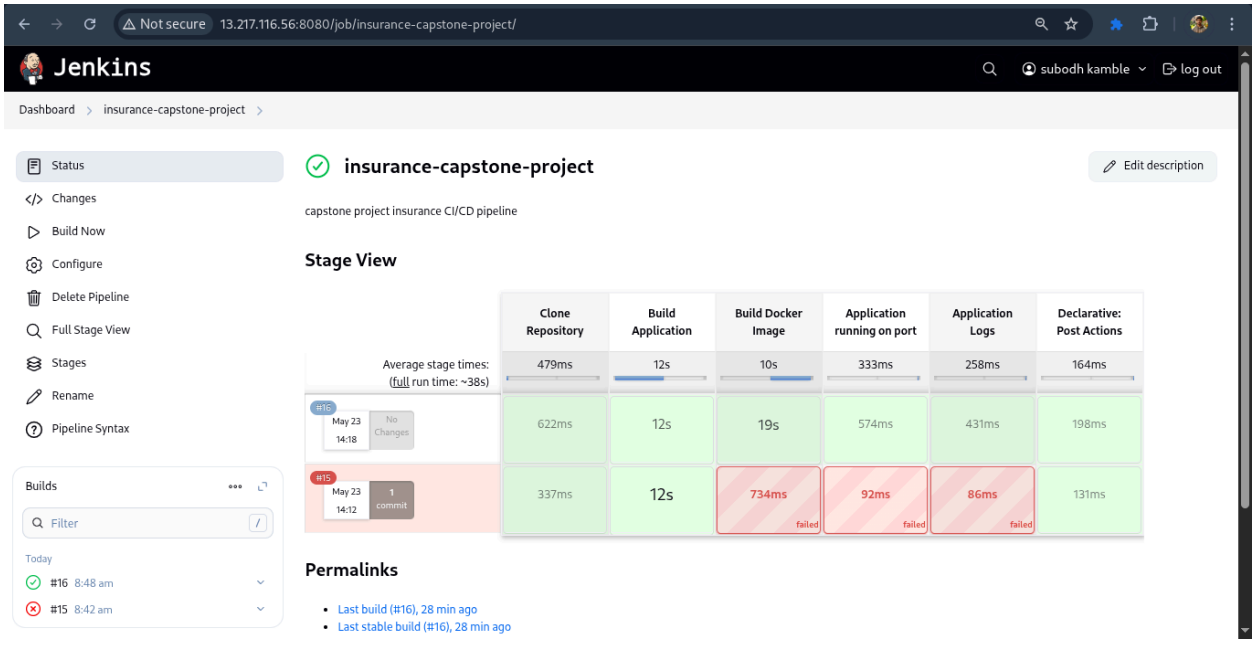



```
pipeline {
    agent any

    environment {
        DOCKER_IMAGE = 'insureme'
        DOCKER_TAG = 'latest'
        DEPLOY_DIR = 'star-agile-insurance-project'
    }
```

```
stages {
    stage('Clone Repository') {
        steps {
            git branch: 'main', url: 'https://github.com/tusuii/insurance_capstone.git'
        }
    }

    stage('Build Application') {
        steps {
            dir("${DEPLOY_DIR}") {
                sh 'mvn clean package'
            }
        }
    }

    stage('Build Docker Image') {
        steps {
            dir("${DEPLOY_DIR}") {
                sh 'docker-compose up --build -d'
            }
        }
    }

    stage('Application running on port') {
        steps {
            sh "docker ps | grep ${DOCKER_IMAGE}"
        }
    }

    stage('Application Logs') {
        steps {
            sh "docker logs ${DOCKER_IMAGE}-app"
        }
    }
}

post {
    success {
        echo "✅ Pipeline completed successfully!"
    }
    failure {
        echo "❌ Pipeline failed!"
    }
    always {
```

```
        echo "Pipeline execution completed."
    }
  }
}
```

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.4)

2025-05-23 08:49:14.543  INFO 1 --- [           main] c.p.s.insureme.InsureMeApplication       : Starting InsureMeApplication
v1.0 using Java 11.0.16 on a075e694df2a with PID 1 (/app.jar started by root in /)
2025-05-23 08:49:14.570  INFO 1 --- [           main] c.p.s.insureme.InsureMeApplication       : No active profile set, falling
back to 1 default profile: "default"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline execution completed.
[Pipeline] echo
✅ Pipeline completed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Application running on
http://13.217.116.56:8081/

Setup Observability:setup prometheus and grafana
Steps to setup prometheus and grafana

- **Machine 1 (Docker-server / prometheus-machine)**
  Responsibilities:
    - Install Docker
    - Run Ubuntu containers
    - Expose Docker metrics
    - Install and run Prometheus
- **Machine 2 (Grafana-monitoring-machine)**
  Responsibilities:
    - Install and run Grafana
    - Connect to Prometheus as a data source
    - Visualize Docker container metrics

Common Setup on Both Machines

Step 1: Switch to Superuser and Update

bash
CopyEdit
sudo su
apt update

Grafana Monitoring Machine Setup

Step 2: Download and Install Grafana

bash
CopyEdit
wget https://dl.grafana.com/enterprise/release/grafana-enterprise-8.4.4.linux-amd64.tar.gz
ls
tar -zxvf grafana-enterprise-8.4.4.linux-amd64.tar.gz
cd grafana-8.4.4
./bin/grafana-server

> **Note:** Grafana by default runs on port 3000.
> Ensure **port 3000 is open** in your instance's **Security Group** or firewall rules.

Step 3: Access Grafana in Browser

text
CopyEdit
http://<Grafana-machine-public-ip>:3000

- **Username:** admin
- **Password:** admin

Docker + Prometheus Machine Setup

Step 4: Install Docker and Run Containers

bash
CopyEdit
apt install docker.io -y
service docker start

```
docker run -dt --name c01 ubuntu
docker run -dt --name c02 ubuntu
```

Step 5: Enable Docker Metrics on Port 9323

Edit Docker Daemon Configuration

```
bash
CopyEdit
vi /etc/docker/daemon.json
```

Add the following JSON content:

```
json
CopyEdit
{
  "metrics-addr" : "0.0.0.0:9323",
  "experimental" : true
}
```

Restart Docker Service

```
bash
CopyEdit
service docker restart
```

Step 6: Install Prometheus

```
bash
CopyEdit
wget
https://github.com/prometheus/prometheus/releases/download/v2.34.0/prometheus-2.34.0.linux-amd64.tar
.gz
tar zxvf prometheus-2.34.0.linux-amd64.tar.gz
cd prometheus-2.34.0.linux-amd64
```

Open the metrics endpoint in browser or curl:

```
bash
CopyEdit
curl http://localhost:9323/metrics
```

Step 7: Configure Prometheus to Scrape Docker Metrics

Open and Edit Prometheus Configuration

bash
CopyEdit
vi prometheus.yml

Add the following scrape job:

yaml
CopyEdit

```
- job_name: "docker"
  static_configs:
    - targets: ["localhost:9323"]
```

Save and exit (Esc, then :wq)

Step 8: Start Prometheus

bash
CopyEdit
./prometheus

Prometheus runs on **port 9090**. Ensure **ports 9090 and 9323 are open** in your firewall/security group.

Access Prometheus:

text
CopyEdit
http://<Docker-server-public-ip>:9090

Configure Grafana to Use Prometheus

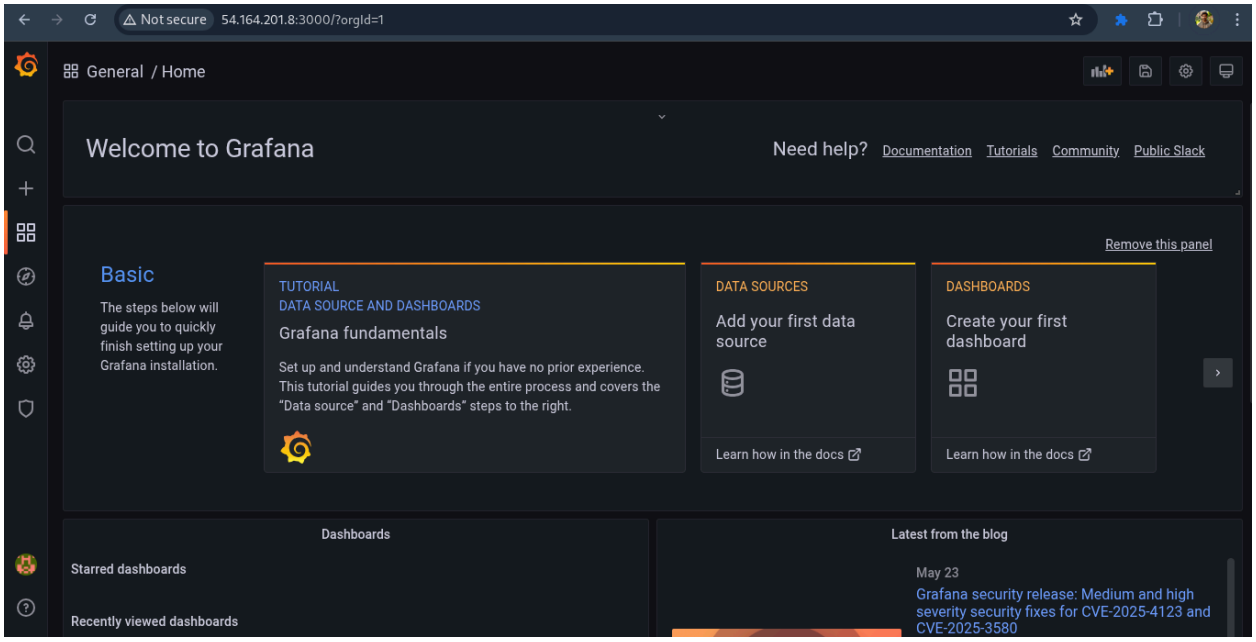
Step 9: Add Prometheus as a Data Source in Grafana

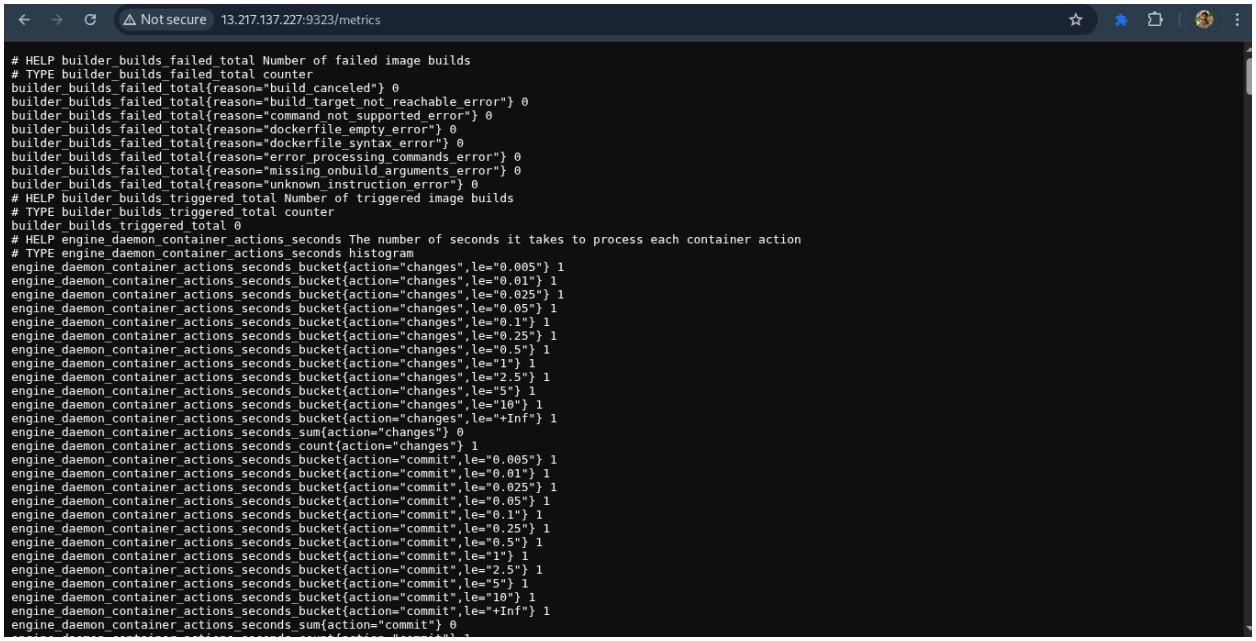
1. Open Grafana:
   http://<Grafana-machine-public-ip>:3000
2. Go to **"Configuration" → "Data Sources"**
3. Click **"Add data source"**
4. Select **Prometheus**

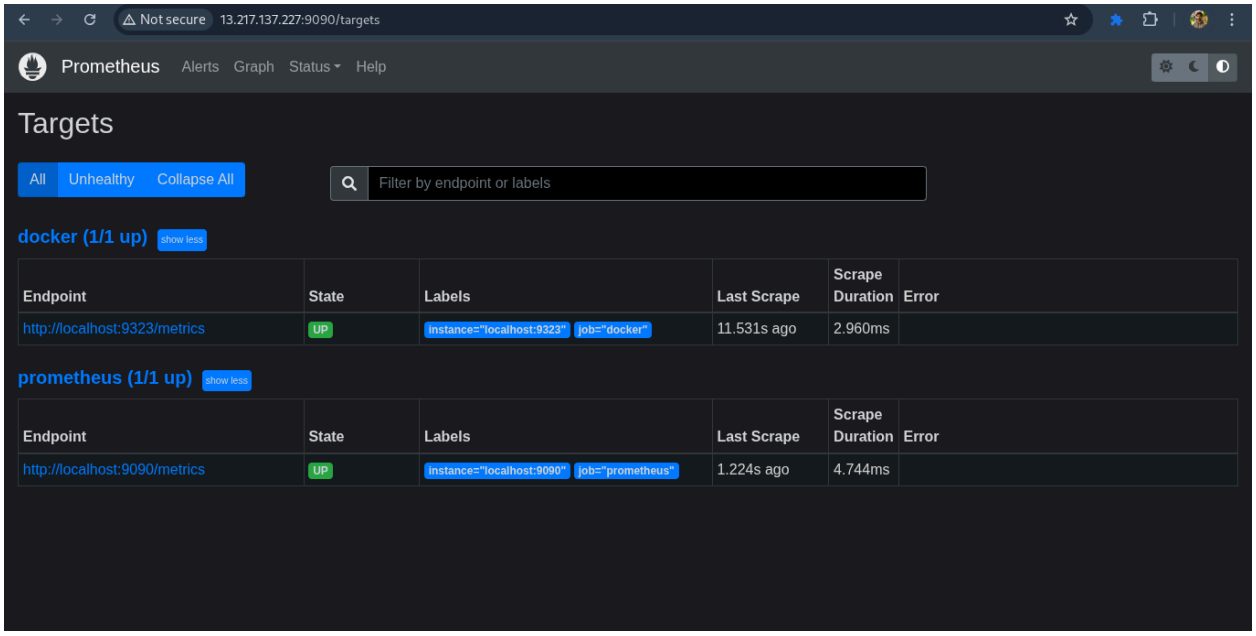
Set the URL:

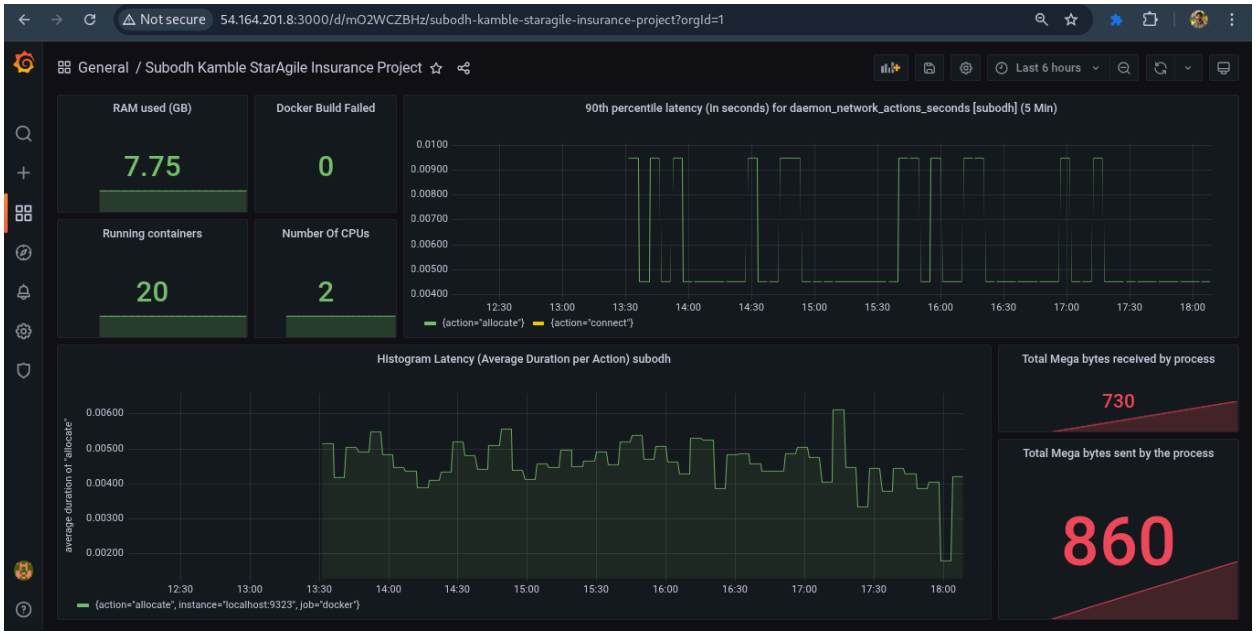
pgsql
CopyEdit
http://<Docker-server-public-ip>:9090/

5. Click **"Save & Test"**
   You should see a success message.









Final dashboard for the project

Giithub: https://github.com/tusuii/insurance_capstone.git