BIPOLAR FACTORY

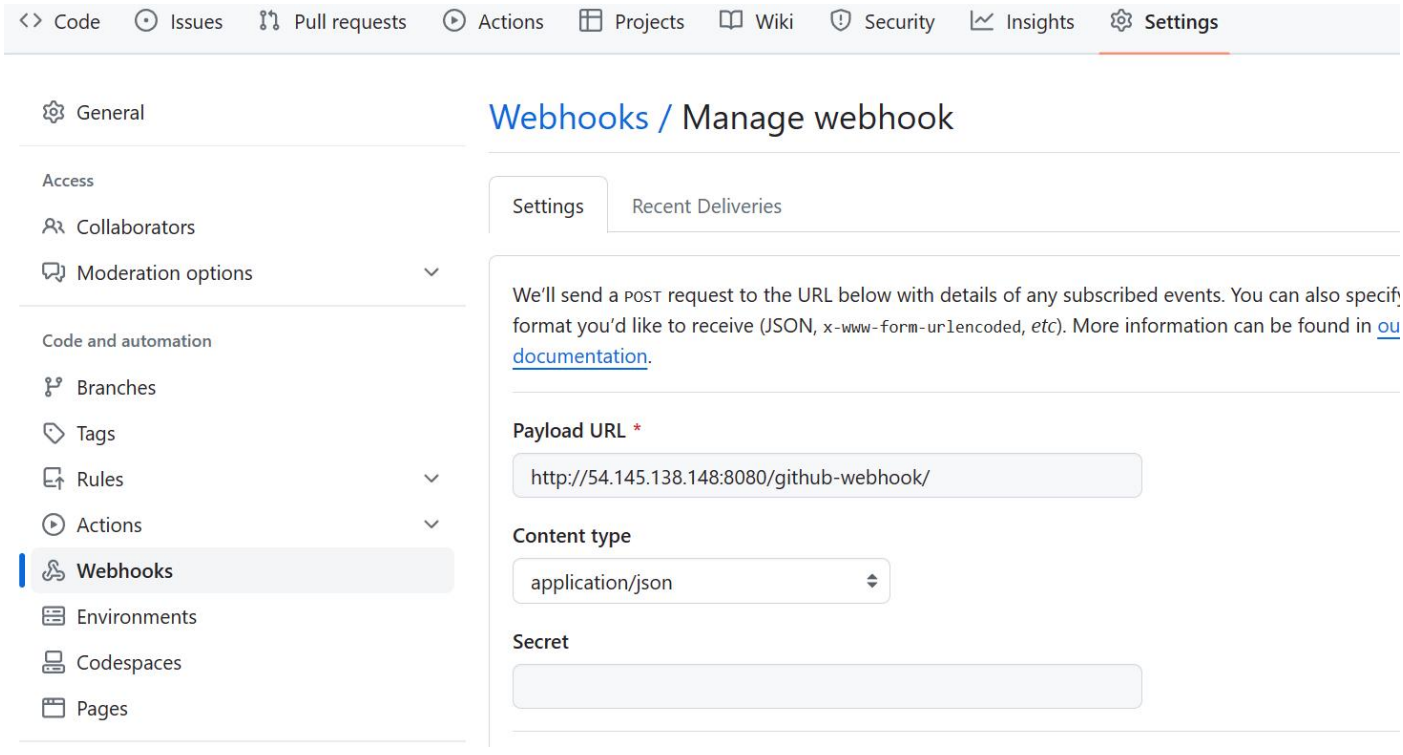
DevOps - Assignment

Submitted by - J. Sai Revanth kumar

➢ Created Spring boot Application of Helloworld :-

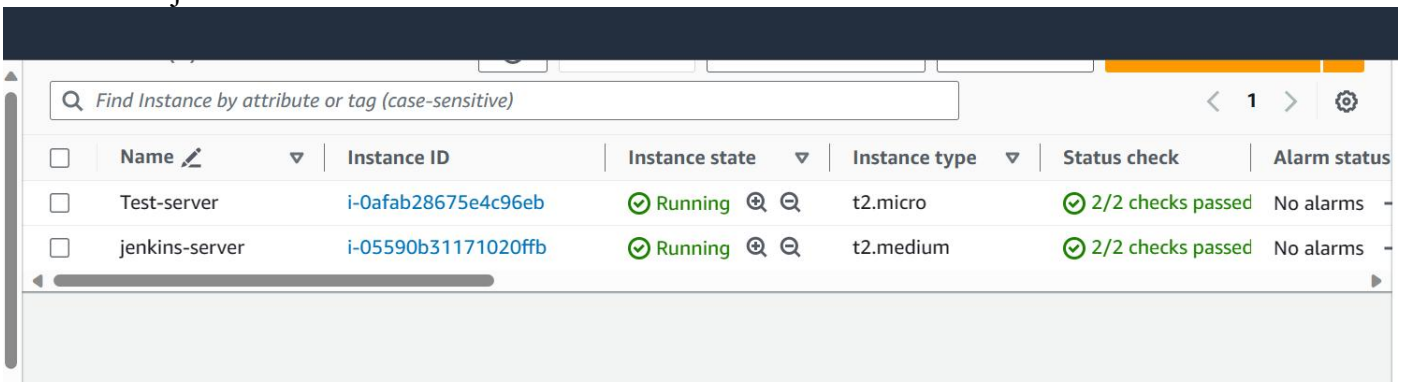
➢ Created GITHUB REPO :-

- Created Github repo :- **https://github.com/SaiRevanth-J/bipolar-test.git**
- Application files are uploaded in the repo.
- In repo setting added Webhook to automate the jenkins job whenever there is new push to repo.



➢ Launched Jenkins-Server for CI/CD pipeline,Monitoring and Test-server for application deployment :-

- In AWS jenkins-server and test-server are launched .



- Test-server is configured with following commands.

1. Sudo apt update -y
2. Sudo apt install docker.io -y

- jenkins-server is configured with following commands to start the setup .

1. Sudo apt update -y
2. Sudo apt install git maven docker.io -y
3. Sudo apt install openjdk-17-jdk -y
4. sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
5. echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
6. sudo apt-get update -y
7. sudo apt-get install jenkins .

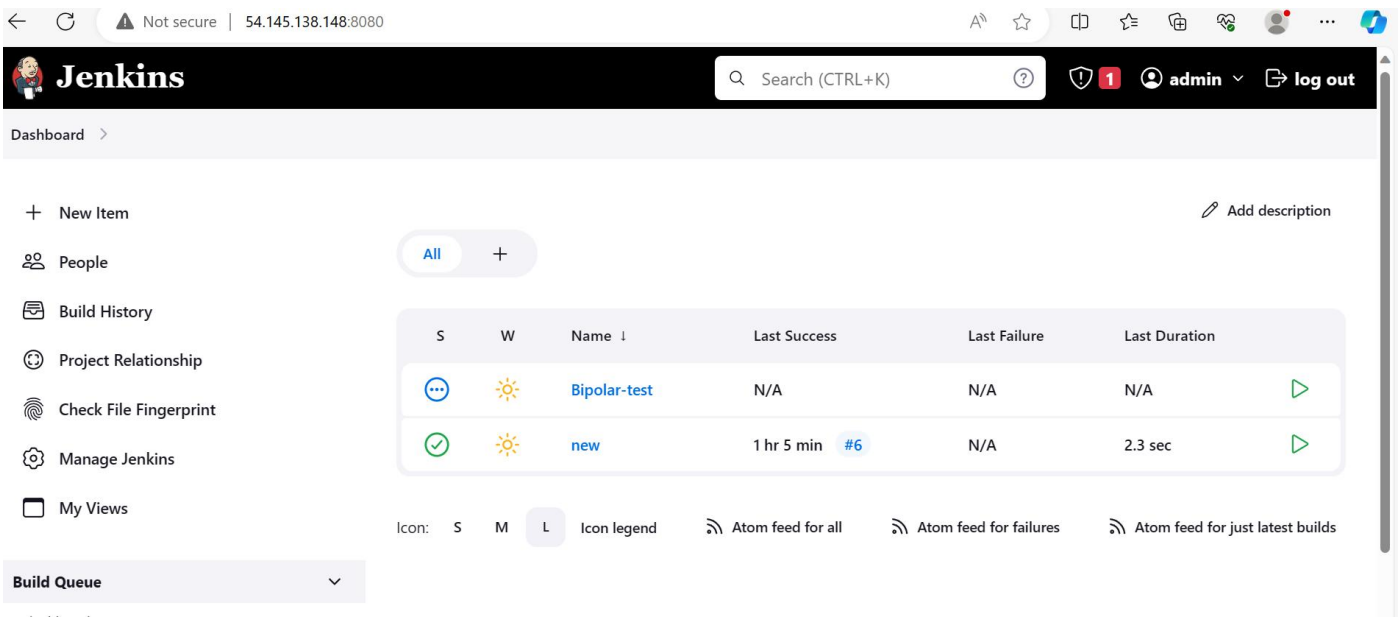
- Added Jenkins user to the sudeors file to give sudo permissions as shown below.

```
  GNU nano 6.2                                          /etc/sudoers.tmp

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
jenkins ALL=(ALL:ALL) NOPASSWD:ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

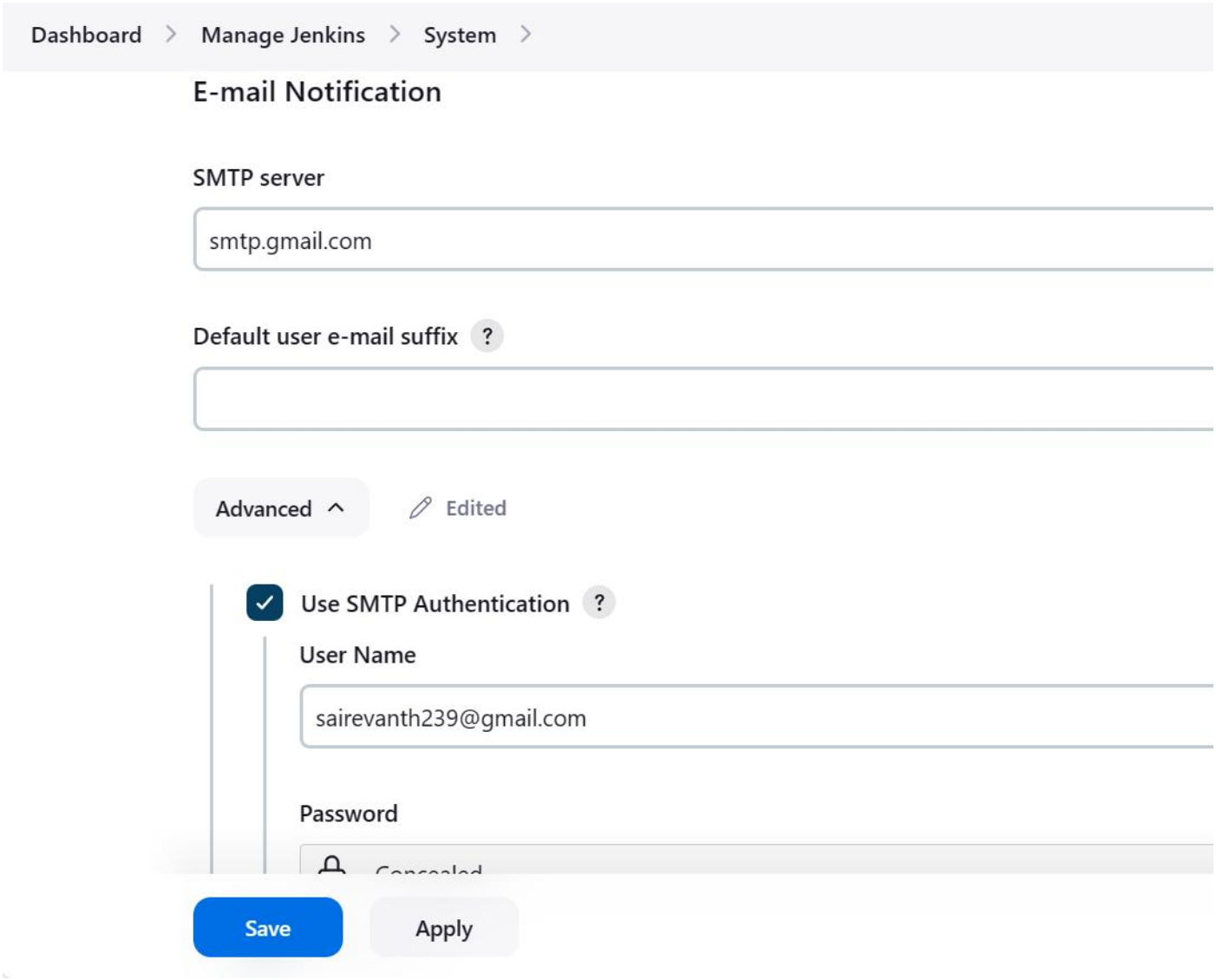
- Maven is used for build application.
- Docker is used to containerize the application .
- Selenium automated test case are written as shown below.

```
package testing.com;

import java.io.IOException;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;


public class testdemo {
    public static void main( String[] args )throws InterruptedException, IOException {
        //System.setProperty("webdriver.chrome.driver", "C:\\Users\\ASUS\\eclipse-workspace\\testing.com\\driver\\chromed
        System.setProperty("webdriver.chrome.driver", "/var/lib/jenkins/workspace/new/chromedriver-linux64/chromedriver")
        ChromeOptions chromeOptions = new ChromeOptions();
        chromeOptions.addArguments("--remote-allow-origins=*");
        chromeOptions.addArguments("start-maximized");
        chromeOptions.addArguments("--headless");
        chromeOptions.addArguments("--no-sandbox");
        chromeOptions.addArguments("--disable-dev-shm-usage");
        chromeOptions.addArguments("--ignore-ssl-errors=yes");
        chromeOptions.addArguments("--ignore-certificate-errors");
        //chromeOptions.setBinary("C:\\Users\\ASUS\\Downloads\\chrome-win64\\chrome-win64\\chrome.exe");
        chromeOptions.setBinary("/var/lib/jenkins/workspace/new/chrome-linux64/chrome");
        WebDriver driver = new ChromeDriver (chromeOptions);

        Thread.sleep(3000);

         Thread.sleep(3000);
         driver.get("http://54.173.163.134:8081");
         driver.manage().window().maximize();
         Thread.sleep(3000);

         String message = driver.findElement(By.xpath("//*[contains(text(),'Hello World!')]")).getText();
         if(message.equals("Hello World!")) {
             System.out.println(" Test Script Executed Successfully");
         } else
         {
             System.out.println("Script Failed");
         }

         driver.quit();

    }
}
```

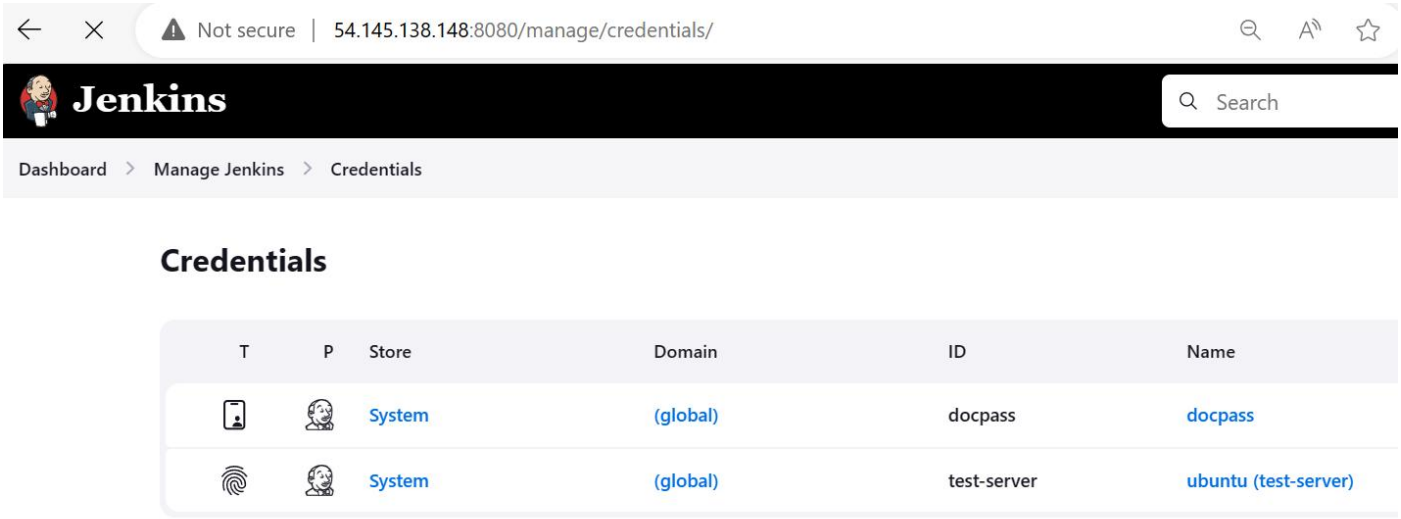
- Jenkins is Accessed at public ip of jenkins-server http://54.145.138.148:8080 as shown below.



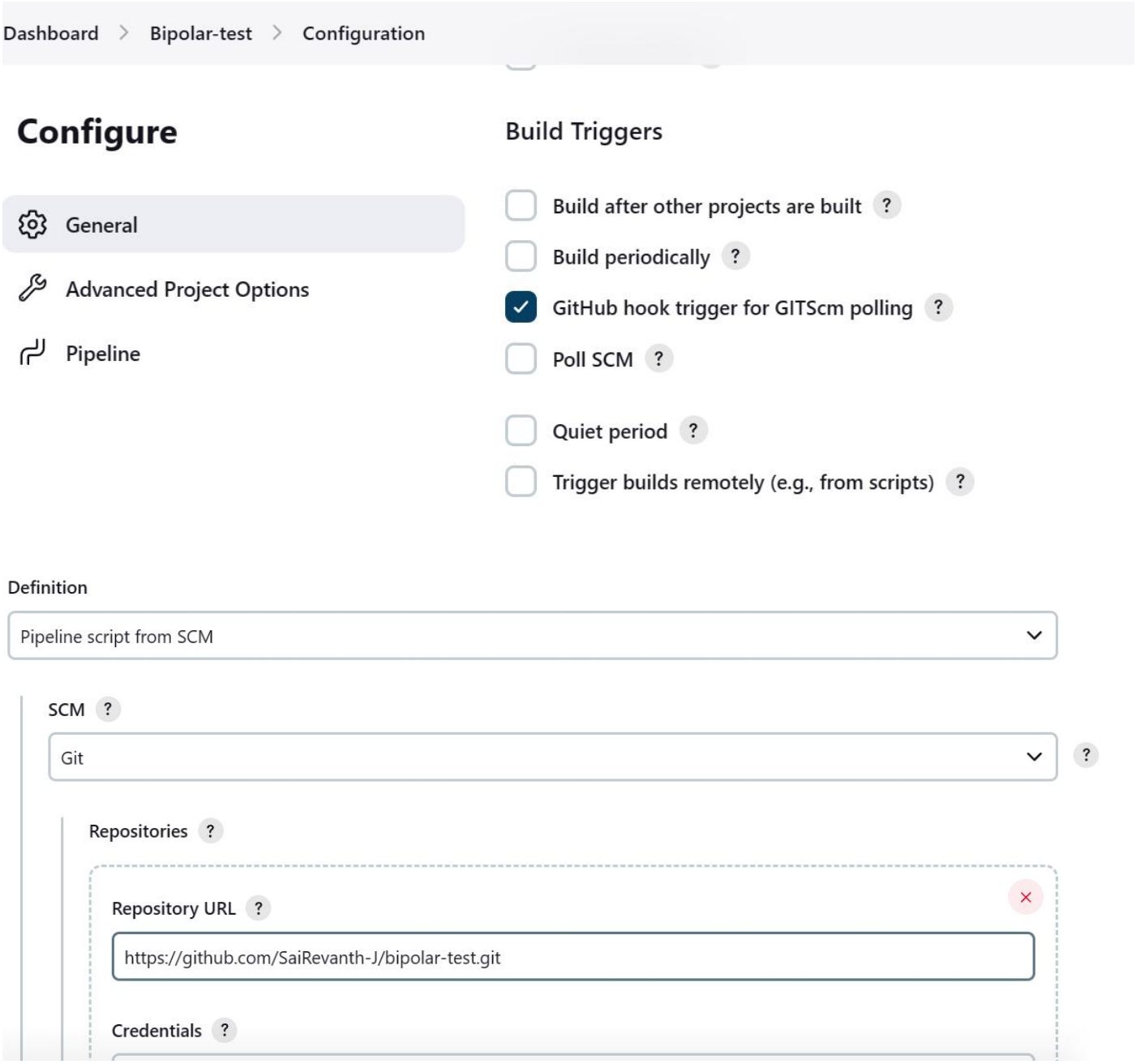
- Smtp Server is Configured in mange jenkins > system to notify the job failure through email



- In Manage Jenkins > Credentials , credentials are configured for dockerhub and test-server access as shown below.



- Created Jenkins pipeline Job with webhook trigger and declarative pipeline is taken from JenkinsFile.



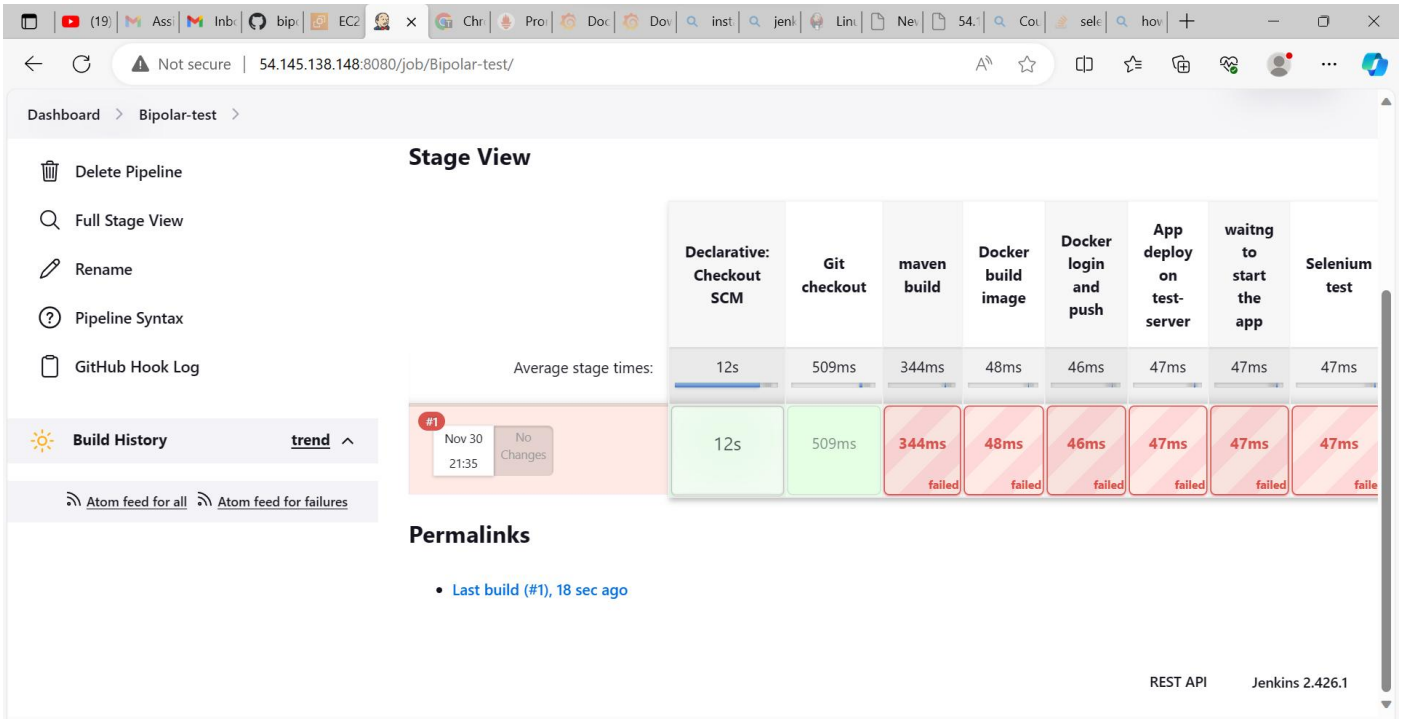
- Jenkinsfile pipeline is as shown below.

```
pipeline {
    agent any


    stages {
        stage('Git checkout') {
            steps {

                    git 'https://github.com/SaiRevanth-J/bipolar-test.git'

                }
            }
```

```
13          stage('maven build') {
14                steps {
15
16                      sh "mvn install package"
17                 }
18          }
19
20
21            stage('Docker build image') {
22                steps {
23
24                   sh' sudo docker system prune -af '
25                   sh ' sudo docker build -t revanthkumar9/bipolar:${BUILD_NUMBER}.0 .'
26
27                 }
28               }
29
30          stage('Docker login and push') {
31                steps {
32                     withCredentials([string(credentialsId: 'docpass', variable: 'docpasswd')]) {
33                   sh ' sudo docker login -u revanthkumar9 -p ${docpasswd} '
34                   sh ' sudo docker push revanthkumar9/bipolar:${BUILD_NUMBER}.0 '
35                   }
36                 }
37          }
38
39          stage('App deploy on test-server ') {
40                steps {
41                   withCredentials([sshUserPrivateKey(credentialsId: 'test-server', keyFileVariable: 'sshkey', passphraseVaria
42
43                       sh 'ssh -o StrictHostKeyChecking=no -i ${sshkey} ${ubuntu}@172.31.35.204 sudo docker system prune -af '
44                        sh 'ssh -o StrictHostKeyChecking=no -i ${sshkey} ${ubuntu}@172.31.35.204 sudo docker run -dt -p 8081:8
45    }
46                 }
47          }
48
49            stage('waitng to start the app') {
50                  steps {
51
52                     sh ' sleep 4'
53
54                   }
55               }
56
57            stage('Selenium test') {
58                  steps {
59
60                     sh 'sudo java -jar demotest.jar'
61                     sh"echo 'application testing  done' "
62
63                   }
64               }
65
66
67        }
```

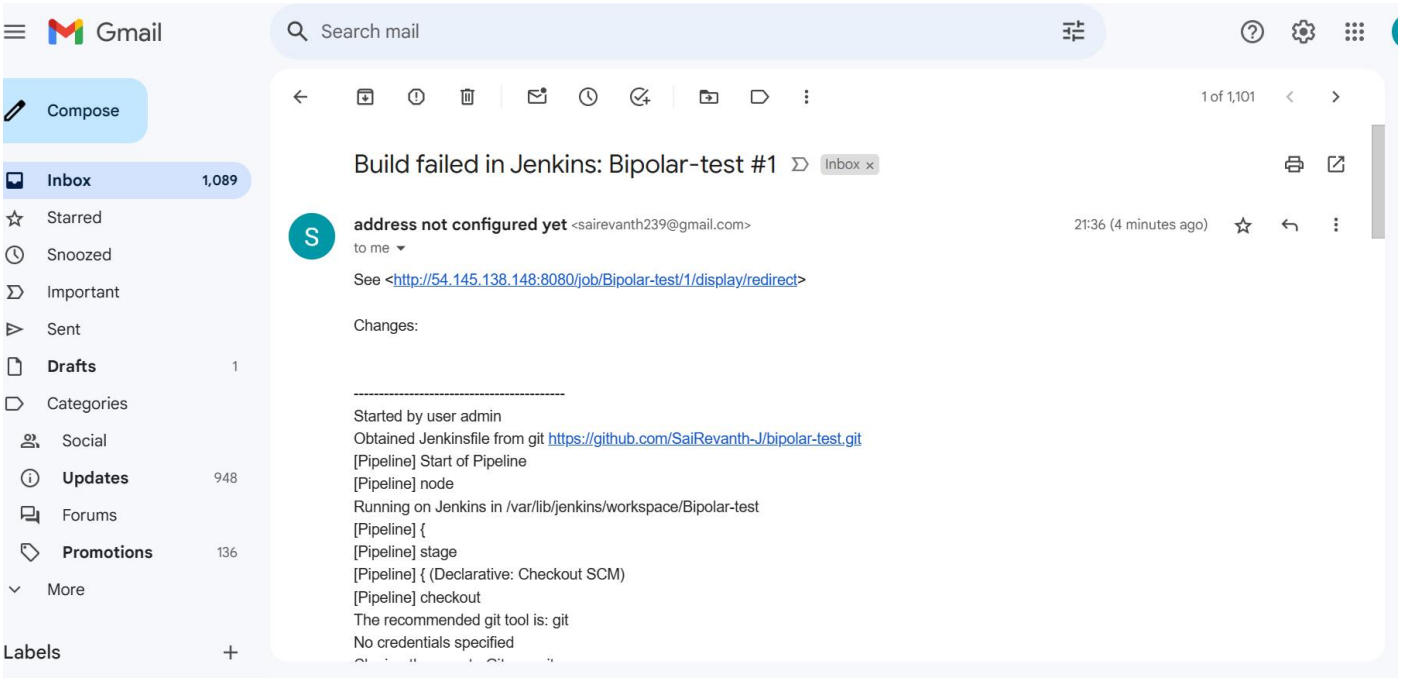
```
70            failure {
71                echo 'sending email notification from jenkins'
72
73                step([$class: 'Mailer',
74           notifyEveryUnstableBuild: true,
75           recipients: emailextrecipients([[$class: 'CulpritsRecipientProvider'],
76                                           [$class: 'RequesterRecipientProvider']])])
77
78
79            }
80        }
81    }
```

- Bipolar-test Pipeline Job-1 is failed intentionally to test the Email Notifications is working or not on job failure.




```
sending email notification from jenkins
[Pipeline] emailextrecipients
[Pipeline] step
Sending e-mails to: sairevanth239@gmail.com
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 127
Finished: FAILURE
```

- Email notification Received successfully on job failure.as shown below

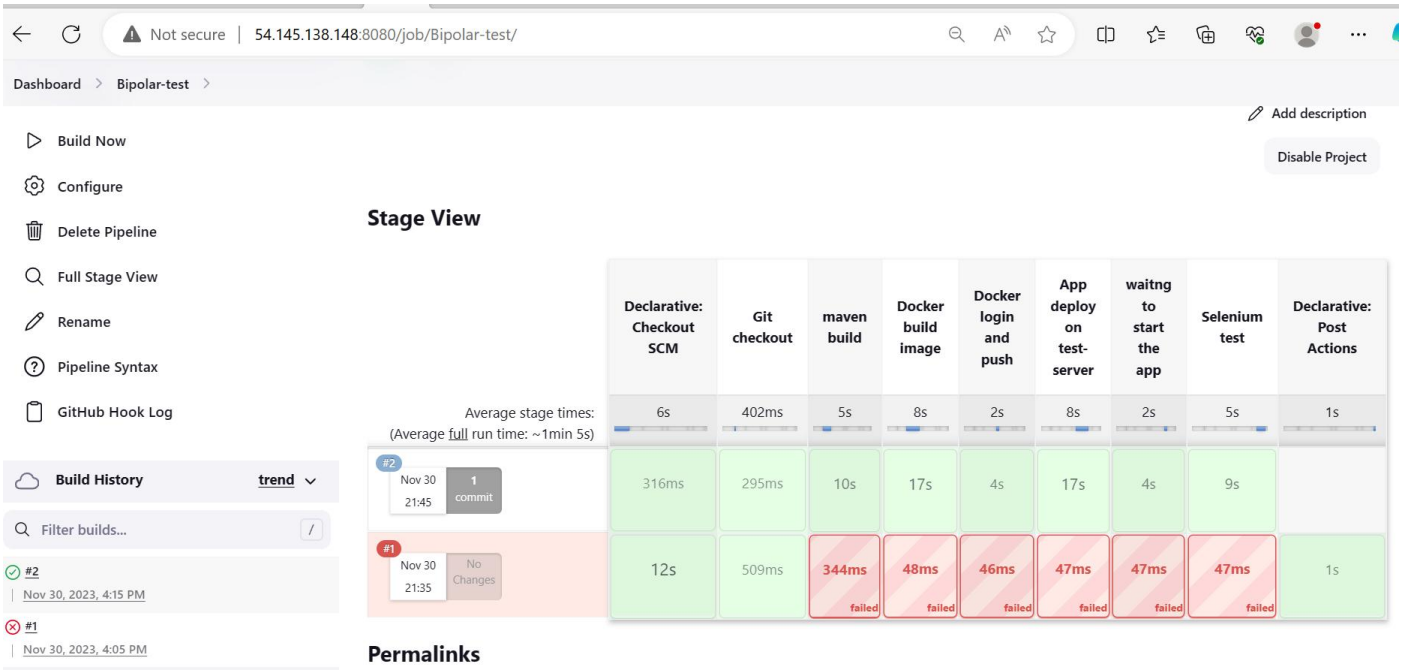


- After Rectification of Error in pipeline and after a new push, webhook triggered the job and run the pipeline successfully as below.

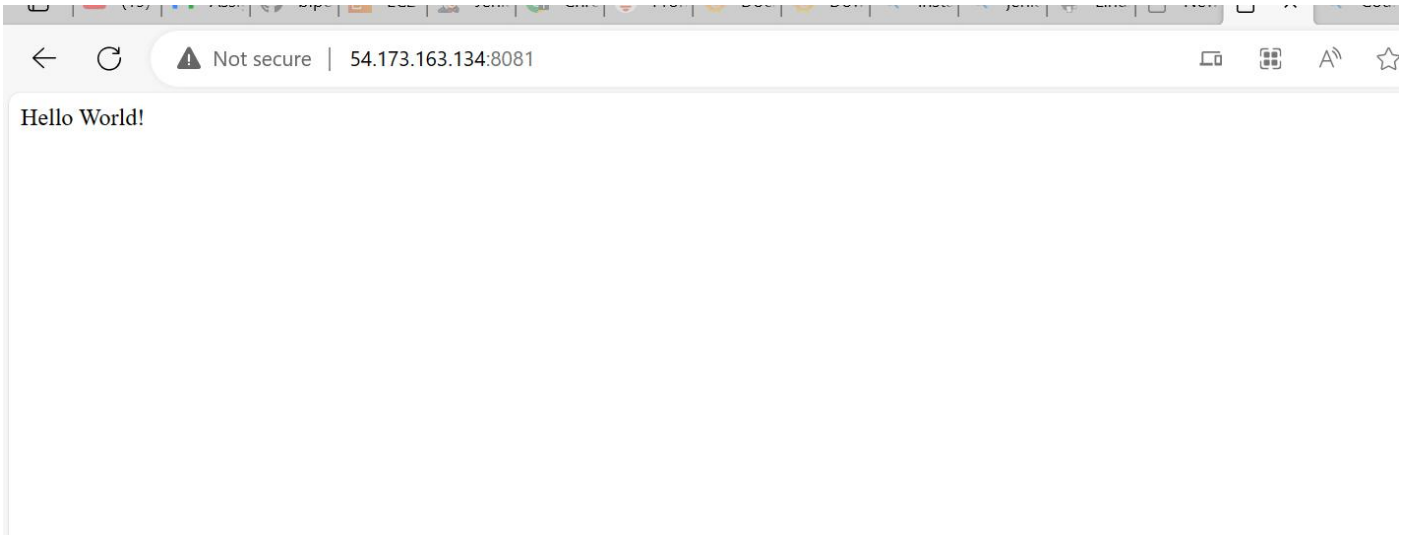
Last GitHub Push

```
Started on Nov 30, 2023, 4:15:22 PM
Started by event from 140.82.115.94 ⇒ http://54.145.138.148:8080/github-webhook/ on Thu Nov 30 16:15:21 UTC 2023
Using strategy: Default
[poll] Last Built Revision: Revision 4a86be5e8e14ae060526053ca66937edc7ed049a (refs/remotes/origin/master)
The recommended git tool is: git
No credentials specified
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git ls-remote -h -- https://github.com/SaiRevanth-J/bipolar-test.git # timeout=10
Found 1 remote heads on https://github.com/SaiRevanth-J/bipolar-test.git
[poll] Latest remote head revision on refs/heads/master is: 56fc1f44fd8963a0bafb86b4821ce052b41ee52d
Using strategy: Default
[poll] Last Built Revision: Revision 4a86be5e8e14ae060526053ca66937edc7ed049a (refs/remotes/origin/master)
The recommended git tool is: git
No credentials specified
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git ls-remote -h -- https://github.com/SaiRevanth-J/bipolar-test.git # timeout=10
Found 1 remote heads on https://github.com/SaiRevanth-J/bipolar-test.git
[poll] Latest remote head revision on refs/heads/master is: 56fc1f44fd8963a0bafb86b4821ce052b41ee52d
Done. Took 0.49 sec
Changes found
```

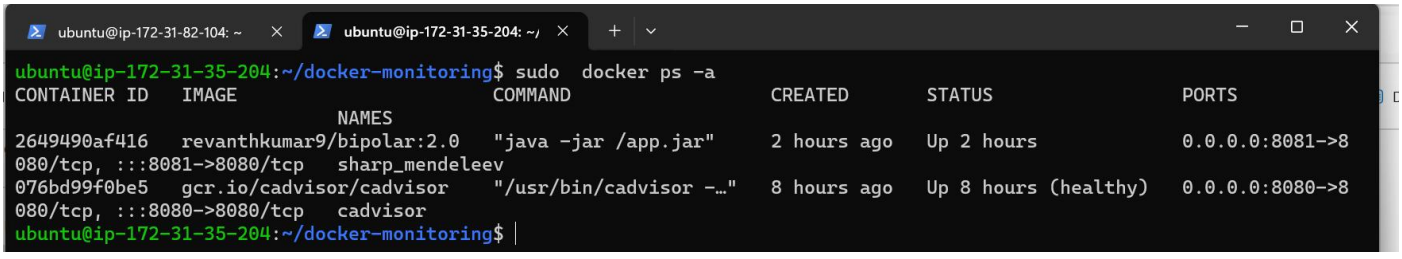
- Pipeline is executed successfully.



- Application is deployed on test-server and accessed on [54. 173. 163. 134:8081](http://54.173.163.134:8081)



- Application is deployed on test-server as below (container name : **sharp_mendeleev**).

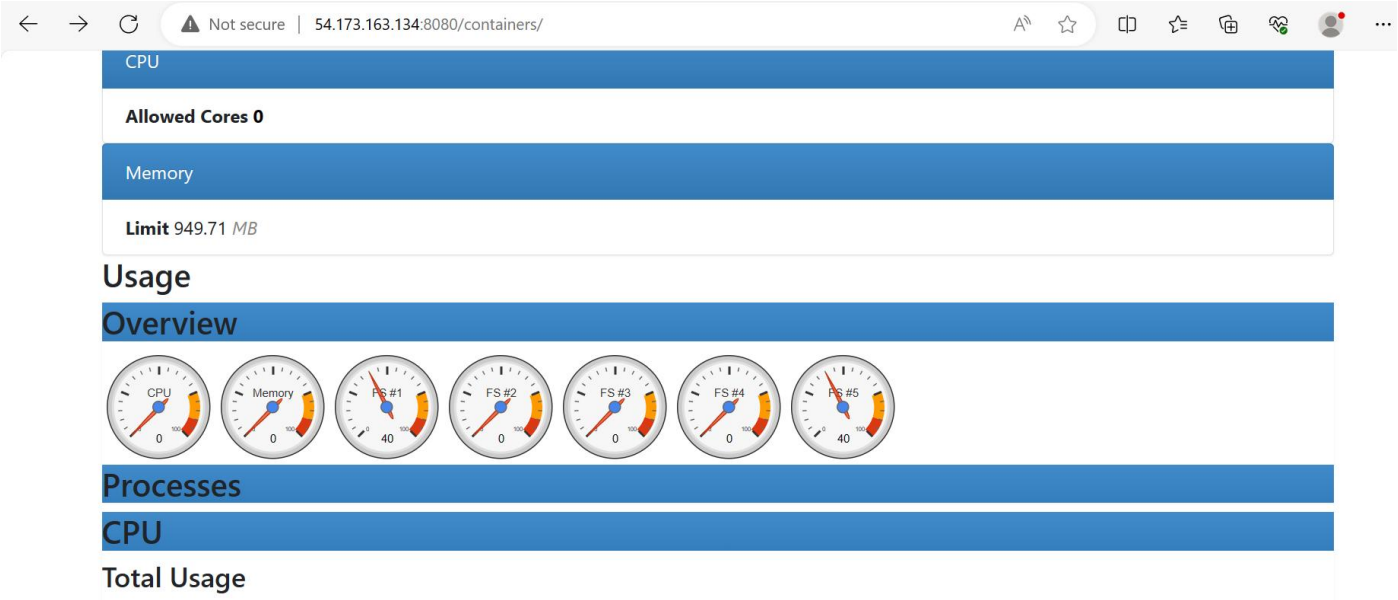


➢ Configured Monitoring and logging :-

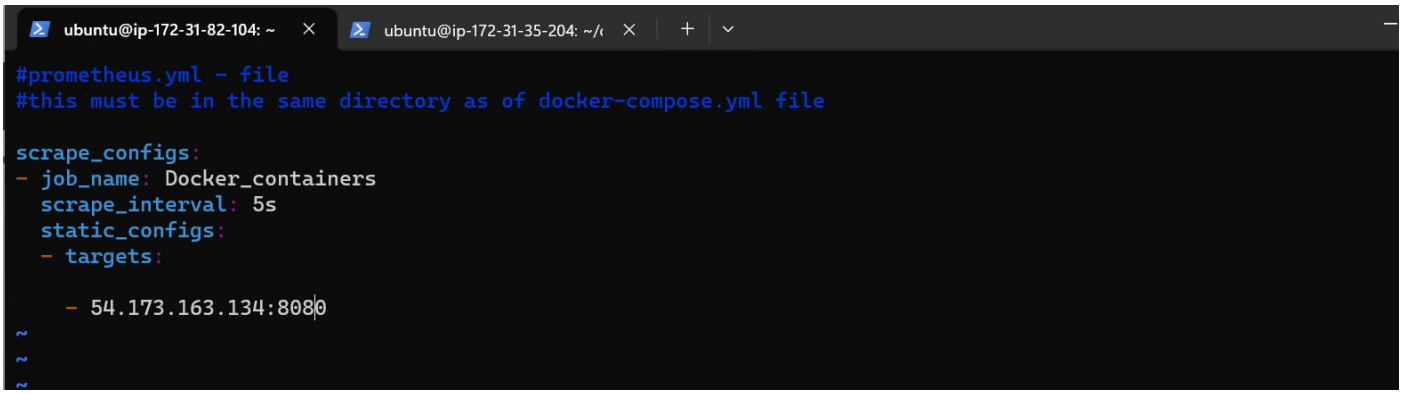
- Deployed application container is monitored by using prometheus , grafana and cAdviser.
- First cAadviser was started on test-server instance to monitor and scrap metrics of the containers with following command.

sudo docker run -d --name=cadvisor -p 8080:8080 -v /:/rootfs:ro -v /var/run:/var/run:ro -v /sys:/sys:ro -v /var/lib/docker/:/var/lib/docker:ro -v /dev/disk/:/dev/disk:ro --privileged --device=/dev/kmsg --restart=unless-stopped gcr.io/cadvisor/cadvisor

- cAdvisor capture the metrics of docker containers.
- Cadvisor container is up and running on test-server 54.173.163.134:8080 as shown below.



- Created prometheus.yml file in jenkins-server and configured to scrap the metrics of test-server docker containers as below from CAdvisor.

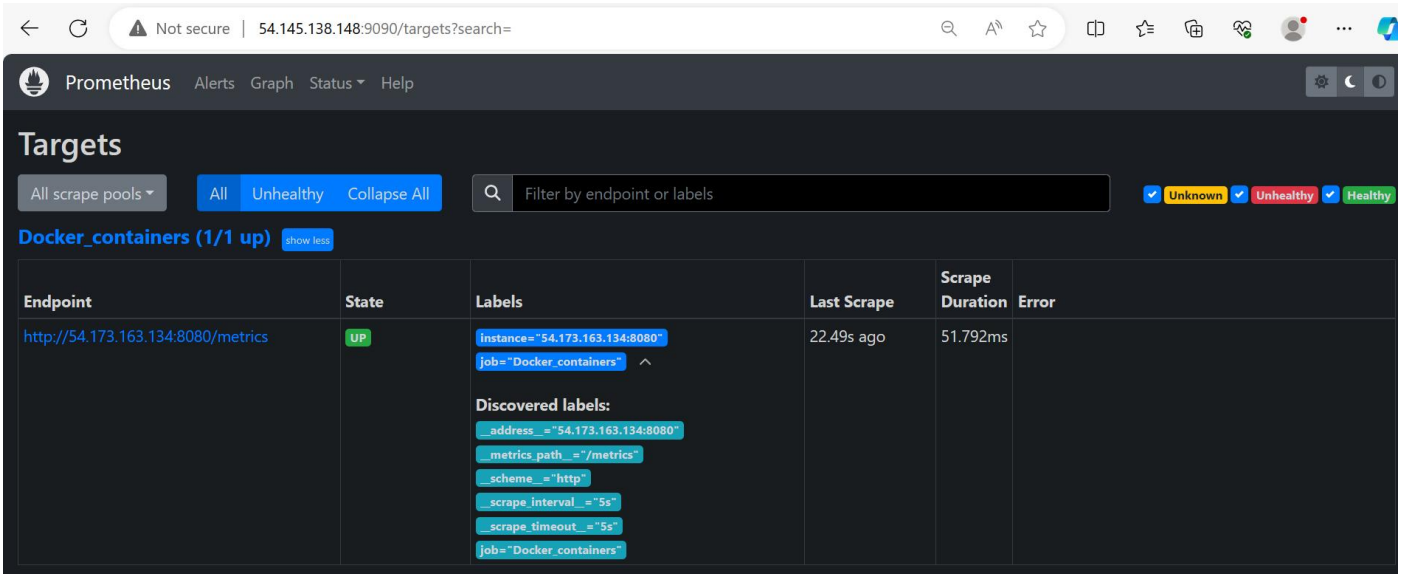


- In Jenkins-server Prometheus and grafana are launced as a docker container to stup monitor and logging with following commands

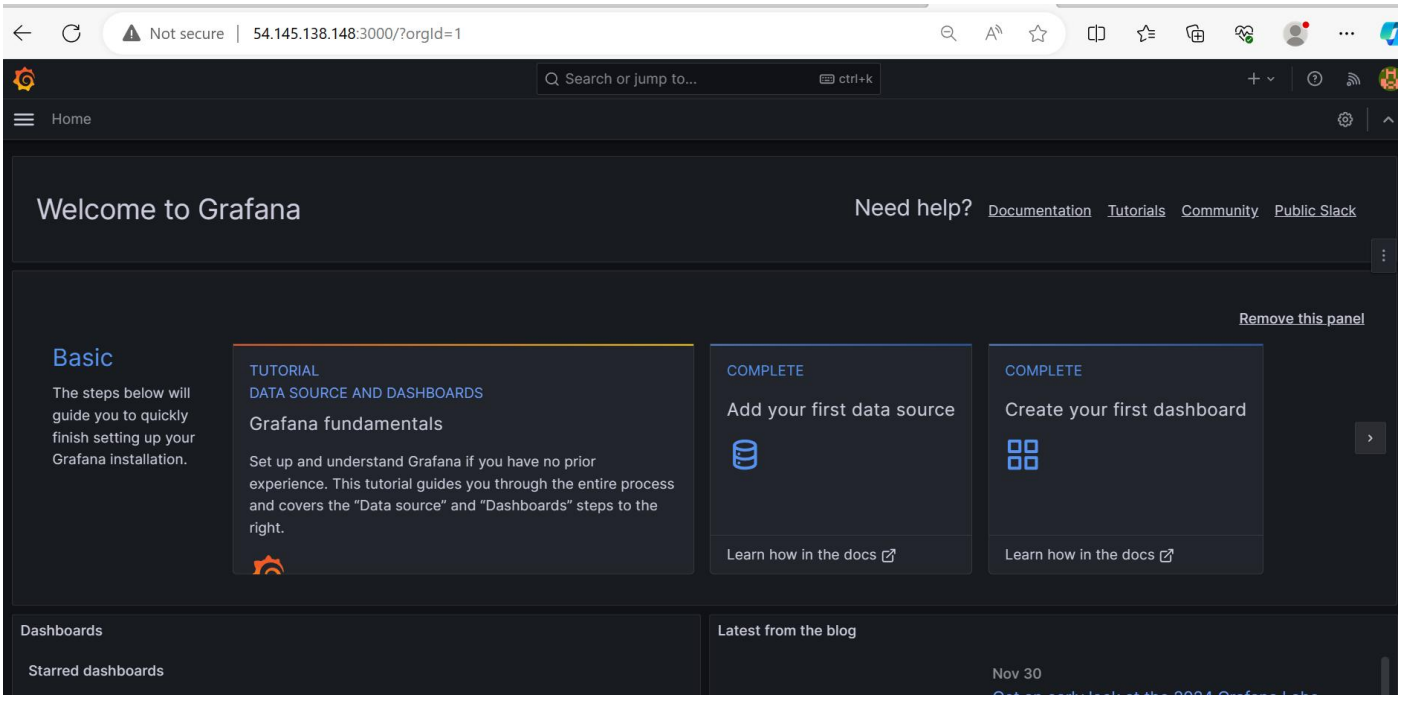
1. Sudo docker volume create prometheus-data
2. Sudo docker run -d \
   -p 9090:9090 \
   -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml \
   -v prometheus-data:/prometheus \
   prom/prometheus

3. Sudo docker run -d --name=grafana -p 3000:3000 grafana/grafana

- Prometheus is accessed on jenkins-server 54.145.138.148:9090 as shown below.



- Grafana is accessed on jenkins-server 54.145.138.148:3000 as shown below.



- Monitoring and logging Dashboard for deployed container is setup as below.