BIPOLAR FACTORY

DevOps - Assignment

Submitted by - J. Sai Revanth kumar

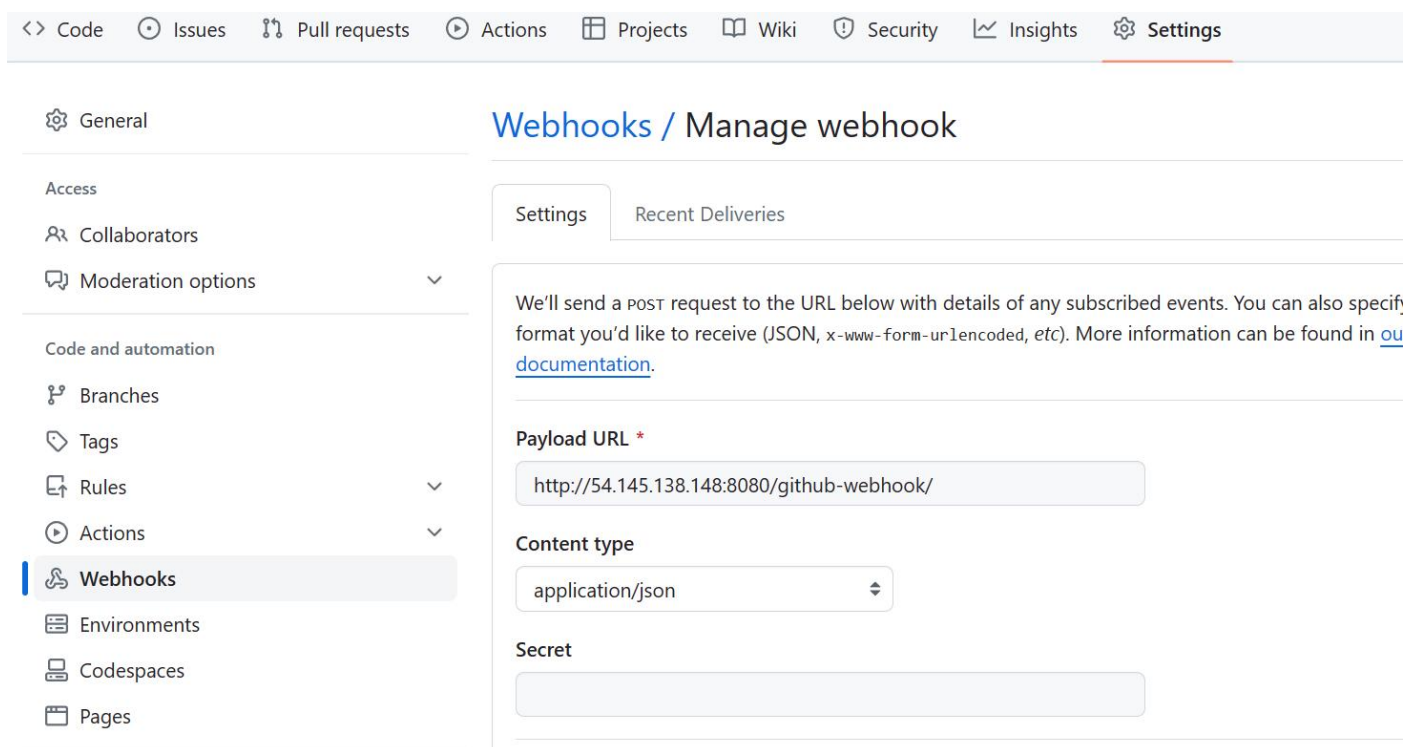## ➢ **Created Spring boot Application of Helloworld :-**

```java
package com.example.helloworld1;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@RestController
public class Helloworld1Application {

    public static void main(String[] args) {
        SpringApplication.run(Helloworld1Application.class, args);
    }

    @GetMapping("/")
    public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {
        return String.format("Hello %s!", name);
    }
}
```

Package Explorer:
- helloworld1
  - src/main/java
    - com.example.helloworld1
      - Helloworld1Application.java
      - ServletInitializer.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - demotest.jar
  - Dockerfile
  - HELP.md
  - Jenkinsfile
  - mvnw
  - mvnw.cmd
  - pom.xml

➢ **Created GITHUB REPO :-**

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

<> Code ⊙ Issues ⇄ Pull requests ⊙ Actions ⊞ Projects ⌑ Wiki ⊘ Security ⬚ Insights ⚙ Settings

Webhooks / Manage webhook

| General |
| Access |
| Collaborators |
| Moderation options |

| Settings | Recent Deliveries |

We'll send a POST request to the URL below with details of any subscribed events. You can also specify format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in ou documentation.
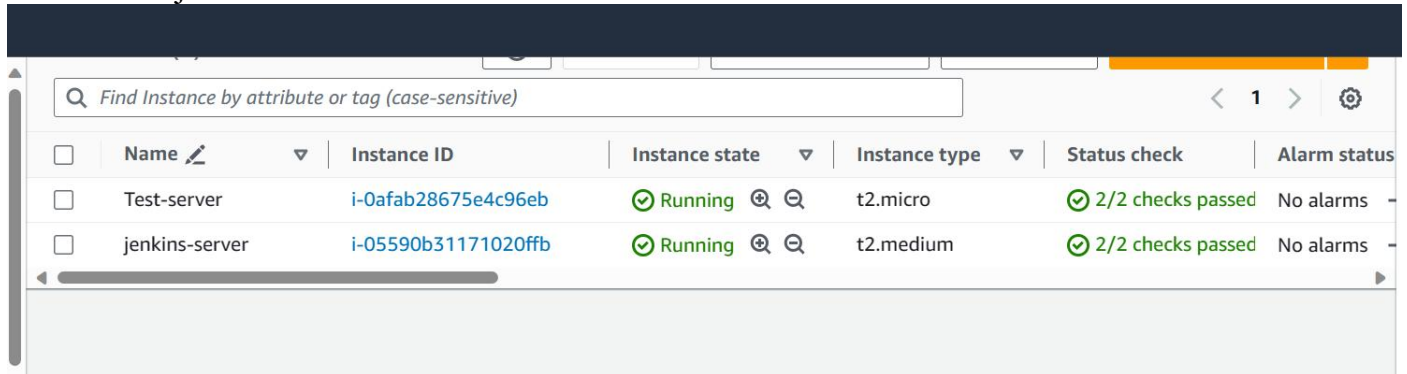
Code and automation
- Branches
- Tags
- Rules
- Actions
- **Webhooks**
- Environments
- Codespaces
- Pages

Payload URL *

http://54.145.138.148:8080/github-webhook/

Content type

application/json

Secret

- ➢ **Launced Jenkins-Server for CI/CD pipeline,Monitoring and Test-server for appliction deployment :-**

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

	Name ✎	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	Test-server	i-0afab28675e4c96eb	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passed	No alarms
☐	jenkins-server	i-05590b31171020ffb	⊘ Running ⊕ ⊖	t2.medium	⊘ 2/2 checks passed	No alarms

- 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
   7. sudo apt-get install jenkins .

- Added Jenkins user to the sudeors file to give sudo persmissions 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
```
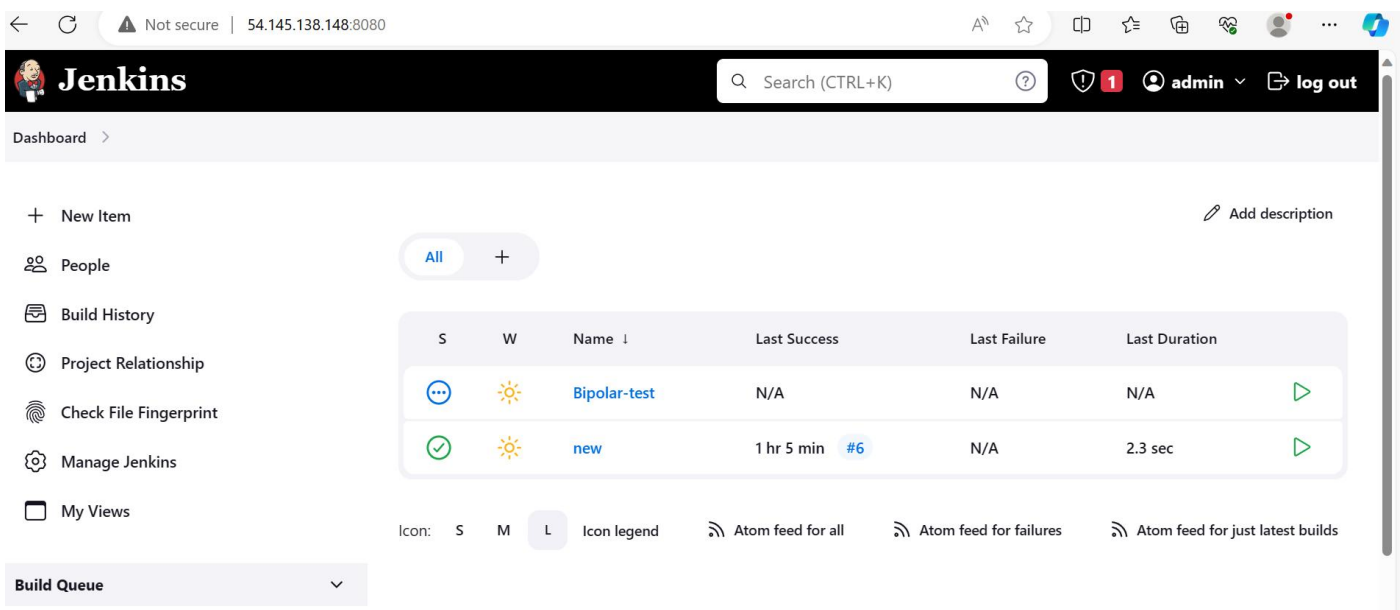
- Maven is used for build application.
- Docker is used to containerize the application .
- Selenium automated test case are written as shown below.

```
1 package testing.com;
2
3 import java.io.IOException;
4
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.chrome.ChromeOptions;
9
10
11 public class testdemo {
12     public static void main( String[] args )throws InterruptedException, IOException {
13         //System.setProperty("webdriver.chrome.driver", "C:\\Users\\ASUS\\eclipse-workspace\\testing.com\\driver\\chromed
14         System.setProperty("webdriver.chrome.driver", "/var/lib/jenkins/workspace/new/chromedriver-linux64/chromedriver")
15         ChromeOptions chromeOptions = new ChromeOptions();
16         chromeOptions.addArguments("--remote-allow-origins=*");
17         chromeOptions.addArguments("start-maximized");
18         chromeOptions.addArguments("--headless");
19         chromeOptions.addArguments("--no-sandbox");
20         chromeOptions.addArguments("--disable-dev-shm-usage");
21         chromeOptions.addArguments("--ignore-ssl-errors=yes");
22         chromeOptions.addArguments("--ignore-certificate-errors");
23         //chromeOptions.setBinary("C:\\Users\\ASUS\\Downloads\\chrome-win64\\chrome-win64\\chrome.exe");
24         chromeOptions.setBinary("/var/lib/jenkins/workspace/new/chrome-linux64/chrome");
25         WebDriver driver = new ChromeDriver (chromeOptions);
26
27         Thread.sleep(3000);

27         Thread.sleep(3000);
28         driver.get("http://54.173.163.134:8081");
29         driver.manage().window().maximize();
30         Thread.sleep(3000);
31
32         String message = driver.findElement(By.xpath("//*[contains(text(),'Hello World!')]")).getText();
33         if(message.equals("Hello World!")) {
34             System.out.println(" Test Script Executed Successfully");
35         } else
36         {
37             System.out.println("Script Failed");
38         }
39
40         driver.quit();
41
42     }
43 }
44
45
```

- 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

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced ∧   ✎ Edited

✓ Use SMTP Authentication ?

User Name

sairevanth239@gmail.com

Password

🔒 Concealed

Save   Apply

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

← ✕ ⚠ Not secure | 54.145.138.148:8080/manage/credentials/   ⊖ A⁰ ☆

🔴 **Jenkins**   🔍 Search

Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store	Domain	ID	Name
📇	👤	System	(global)	docpass	docpass
🔎	👤	System	(global)	test-server	ubuntu (test-server)

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

Configure

Build Triggers

- General
- Advanced Project Options
- Pipeline

	Build after other projects are built ?
	Build periodically ?
✓	GitHub hook trigger for GITScm polling ?
	Poll SCM ?
	Quiet period ?
	Trigger builds remotely (e.g., from scripts) ?

Definition

```
Pipeline script from SCM
```

SCM ?

```
Git
```
?

Repositories ?

Repository URL ?

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

Credentials ?

- Jenkinsfile pipeline is as shown below.

```
 1    pipeline {
 2        agent any
 3
 4
 5        stages {
 6            stage('Git checkout') {
 7                steps {
 8
 9                        git 'https://github.com/SaiRevanth-J/bipolar-test.git'
10
11                    }
12                }
```

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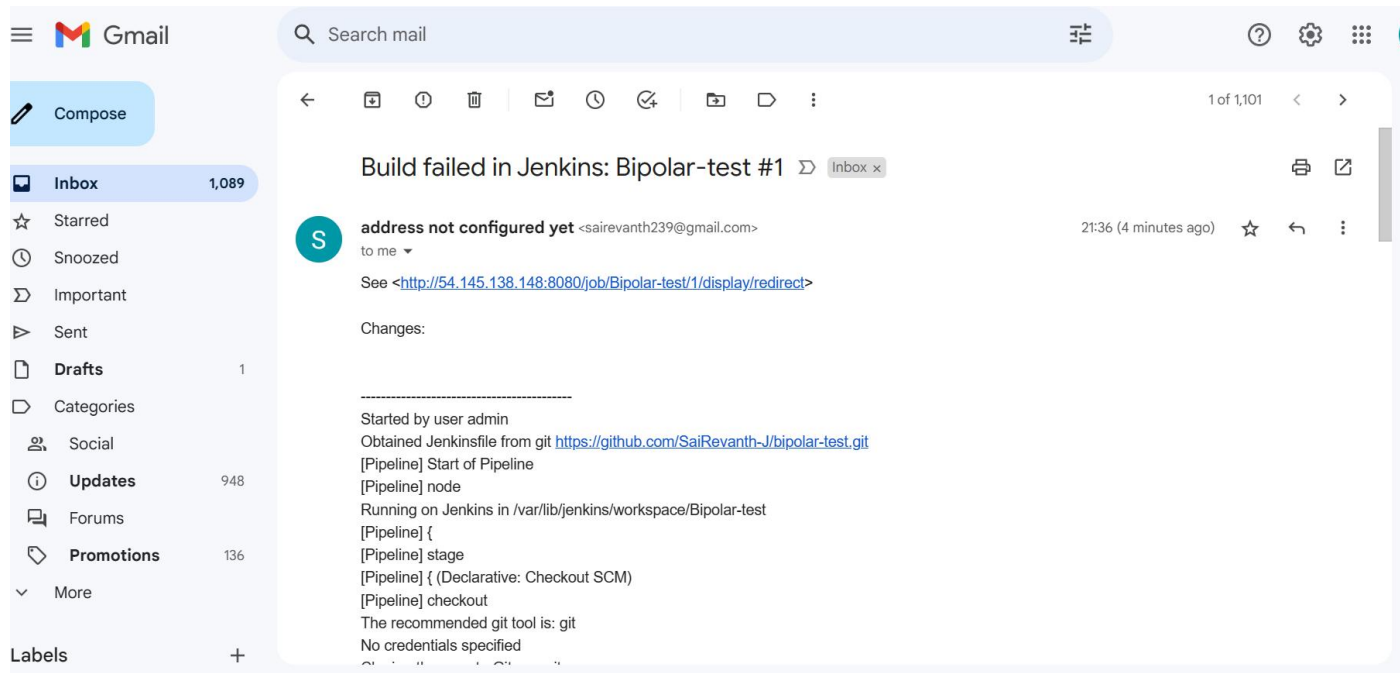
- After Rectification of Error in pipeline and after a new push, webhook triggered the job and run the pipeline successfully as below.

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.
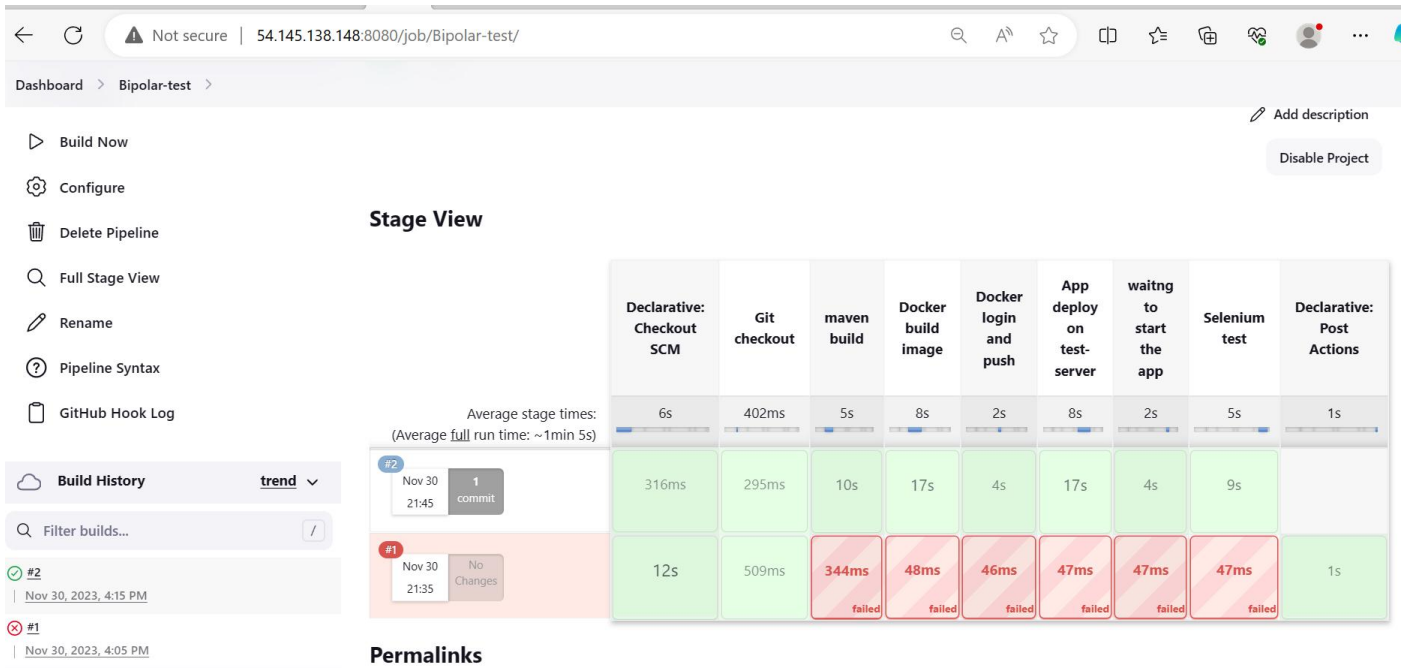
- Application is deployed on test-server and accessed on 54. 173. 163. 134:8081

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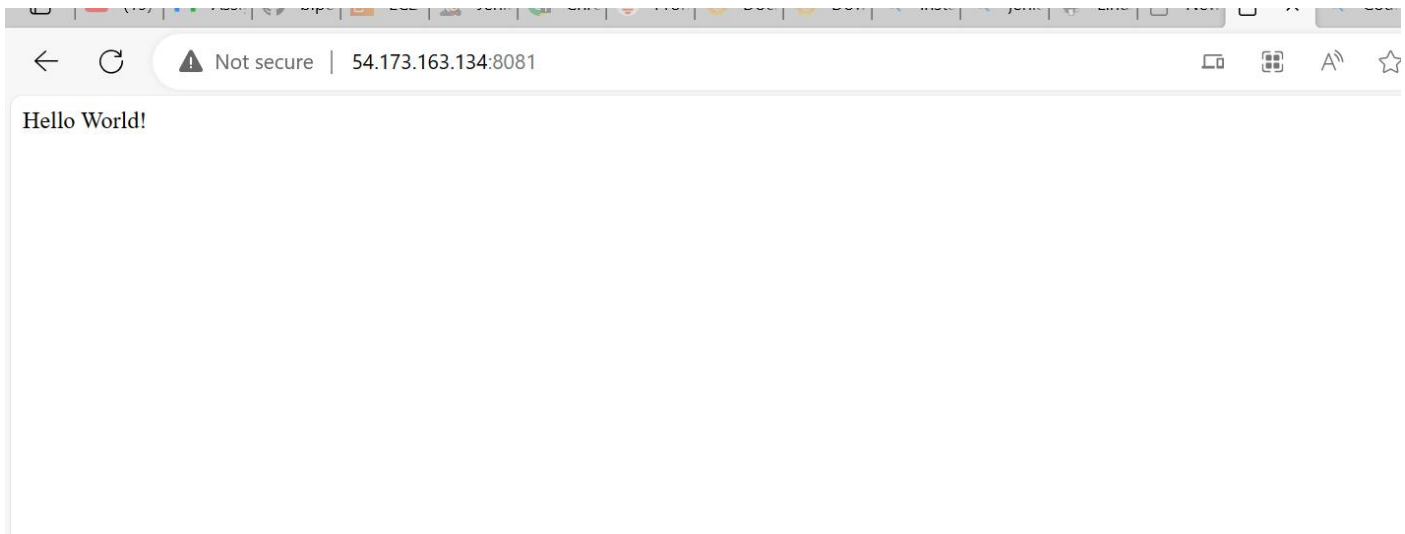
- Application is deployed on test-server as below (container name : **sharp_mendeleev**).

➢ **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 the running 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
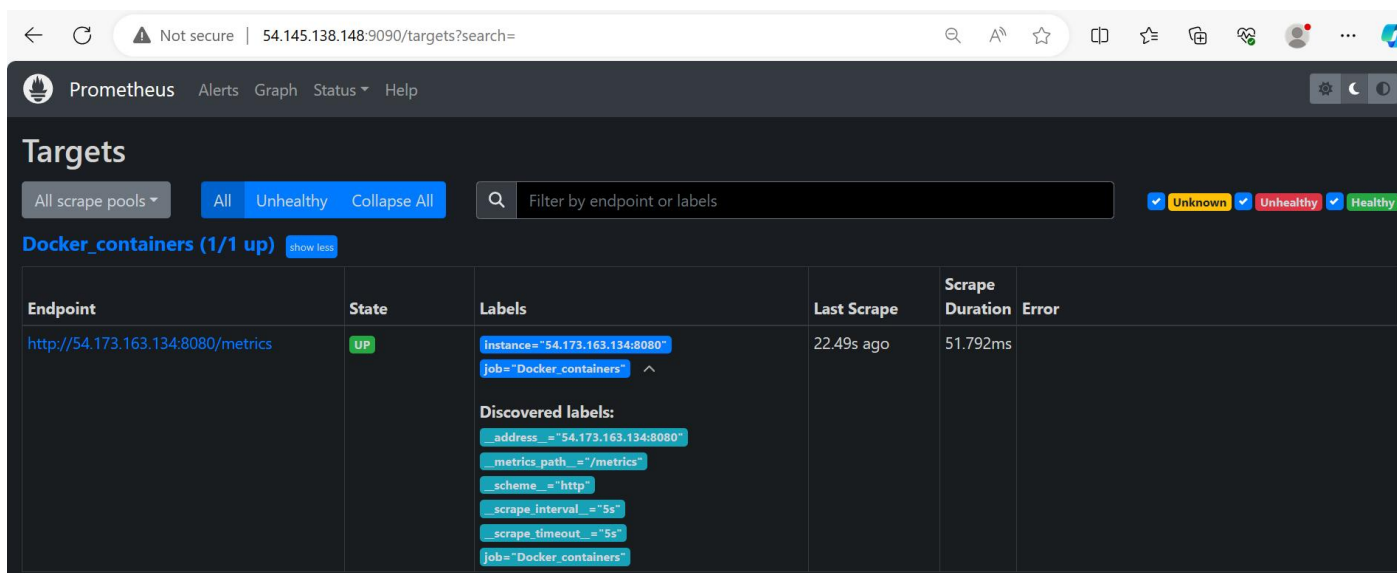
● cAdvisor capture the metrics of docker containers.

● Cadvisor container is up and running on test-server 54.173.163.134:8080 as shown below.
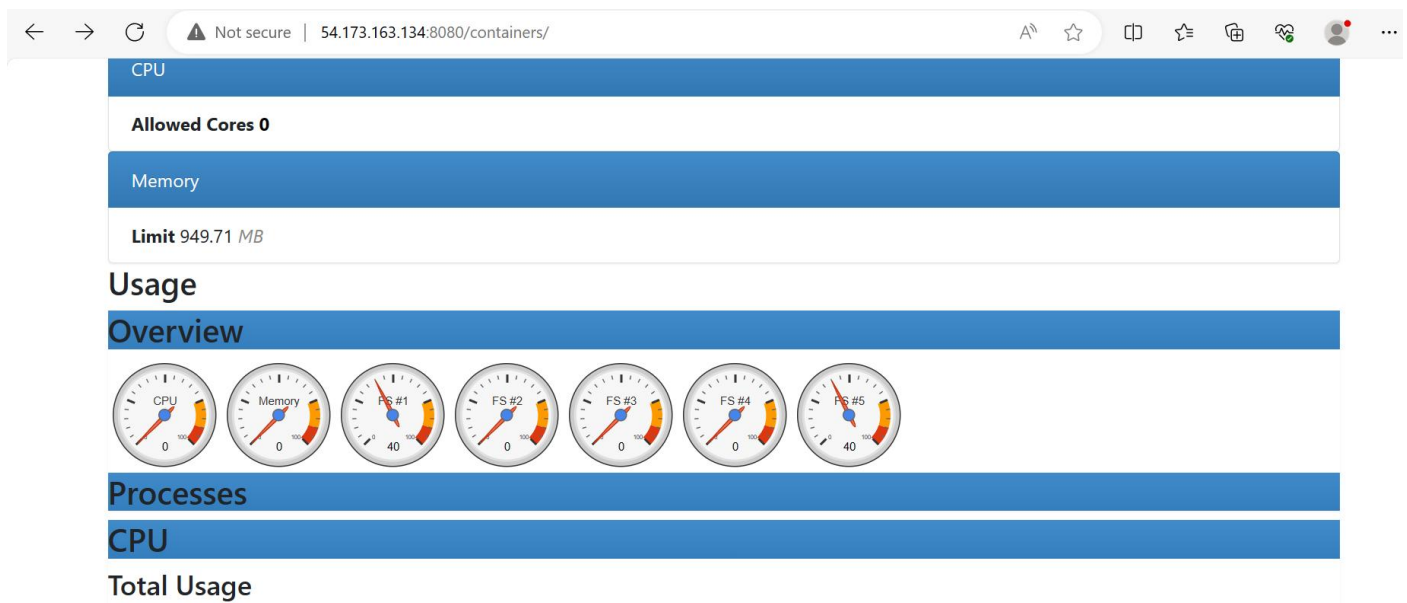
● Created promethus.yml file in jenkins-server and configured to scrap the metrics of test-server docker containers as below from CAdvisor.
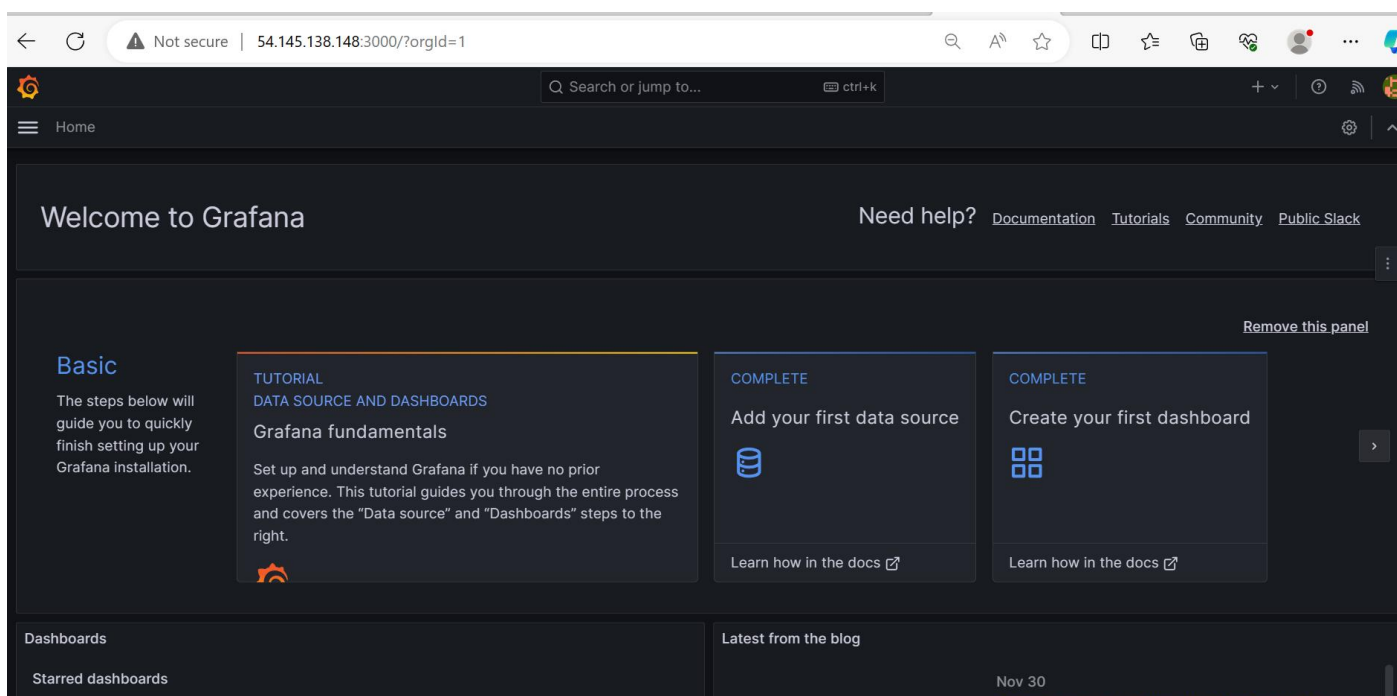
- In Jenkins-server Prometheus and grafana are launced as a docker container to stup monitor and logging with following commands

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

  3. 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.