

BIPOLAR FACTORY

DevOps - Assignment

Submitted by - J. Sai Revanth kumar

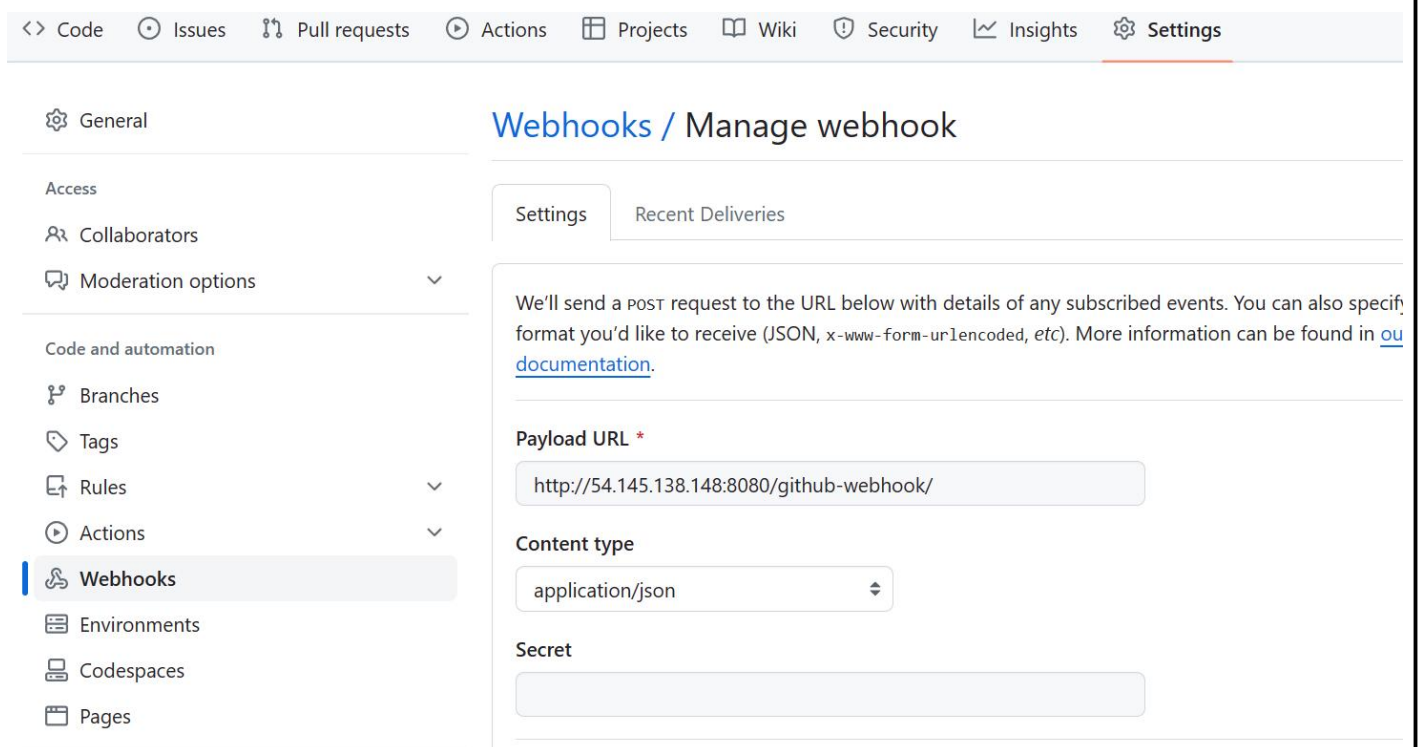
➤ Created Spring boot Application of Helloworld :-



```
1 package com.example.helloworld1;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @RestController
7 public class Helloworld1Application {
8
9     public static void main(String[] args) {
10         SpringApplication.run(Helloworld1Application.class, args);
11     }
12
13     @GetMapping("/")
14     public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {
15         return String.format("Hello %s!", name);
16     }
17 }
18
19
20
21
22
23
```

➤ 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.



<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Webhooks / Manage webhook

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

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 [our documentation](#).

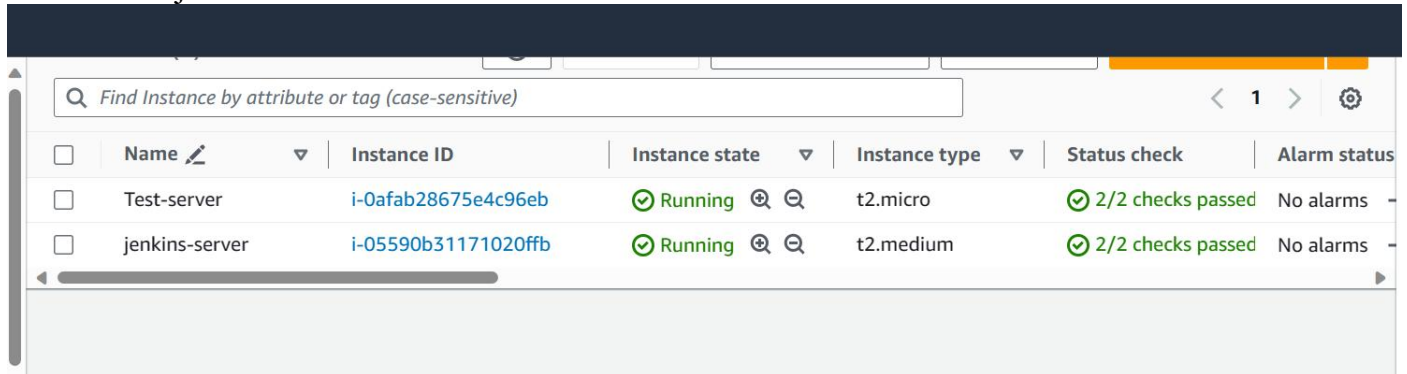
Payload URL *

Content type

Secret

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

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



The screenshot shows the AWS Management Console interface for EC2 instances. At the top, there is a search bar with the placeholder text "Find Instance by attribute or tag (case-sensitive)". Below the search bar, there is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Two instances are listed: "Test-server" and "jenkins-server". Both are in the "Running" state. The "Test-server" has an Instance ID of "i-0afab28675e4c96eb" and is of type "t2.micro". The "jenkins-server" has an Instance ID of "i-05590b31171020ffb" and is of type "t2.medium". Both instances show "2/2 checks passed" in the Status check column and "No alarms" in the Alarm status column.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	Test-server	i-0afab28675e4c96eb	Running	t2.micro	2/2 checks passed	No alarms
<input type="checkbox"/>	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 -y
7. sudo apt-get install jenkins .

- Added Jenkins user to the sudoers 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.

```

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

```

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

The screenshot shows the Jenkins web interface accessed at <http://54.145.138.148:8080>. The interface includes a search bar, user profile (admin), and a log out button. The main dashboard area displays a table of build history for the 'Bipolar-test' project.

S	W	Name	Last Success	Last Failure	Last Duration
...	☀	Bipolar-test	N/A	N/A	N/A
✓	☀	new	1 hr 5 min #6	N/A	2.3 sec

At the bottom, there is a 'Build Queue' section and a footer with icons for status (S, M, L), legend, and Atom feeds for all, failures, and latest builds.

- Smtplib Server is Configured in manage jenkins > system to notify the job failure through email

Dashboard > Manage Jenkins > System >

E-mail Notification

SMTP server

Default user e-mail suffix ?

Advanced ^ Edited

☒ Use SMTP Authentication ?

User Name

Password

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/ 🔍 A ☆

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.

Dashboard > Bipolar-test > Configuration

Configure

General

Advanced Project Options

Pipeline

Build Triggers

- ☐ 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     }
14 }
```



```

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.

Dashboard > Bipolar-test >

Stage View

Actions: Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, GitHub Hook Log

Average stage times:

	Declarative: Checkout SCM	Git checkout	maven build	Docker build image	Docker login and push	App deploy on test-server	waiting to start the app	Selenium test
#1 Nov 30 21:35 No Changes	12s	509ms	344ms	48ms	46ms	47ms	47ms	47ms

Permalinks

- Last build (#1), 18 sec ago

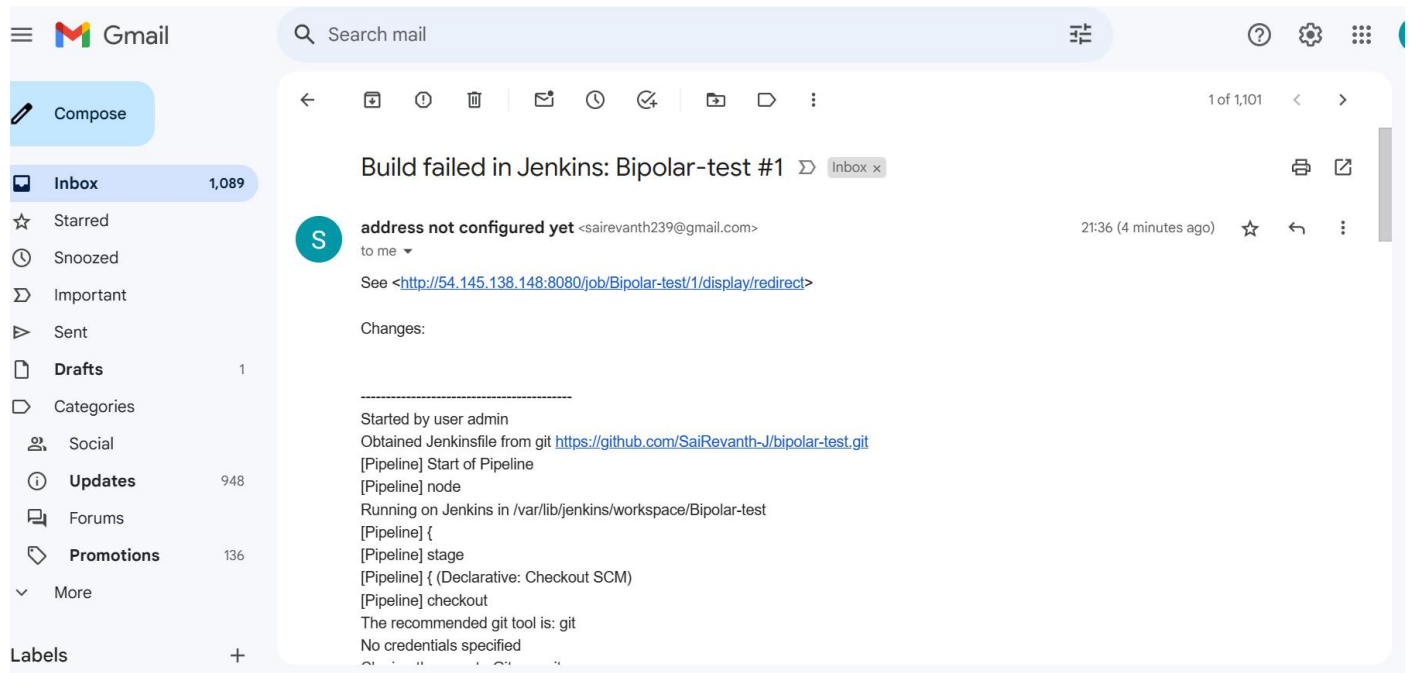
REST API Jenkins 2.426.1

```

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

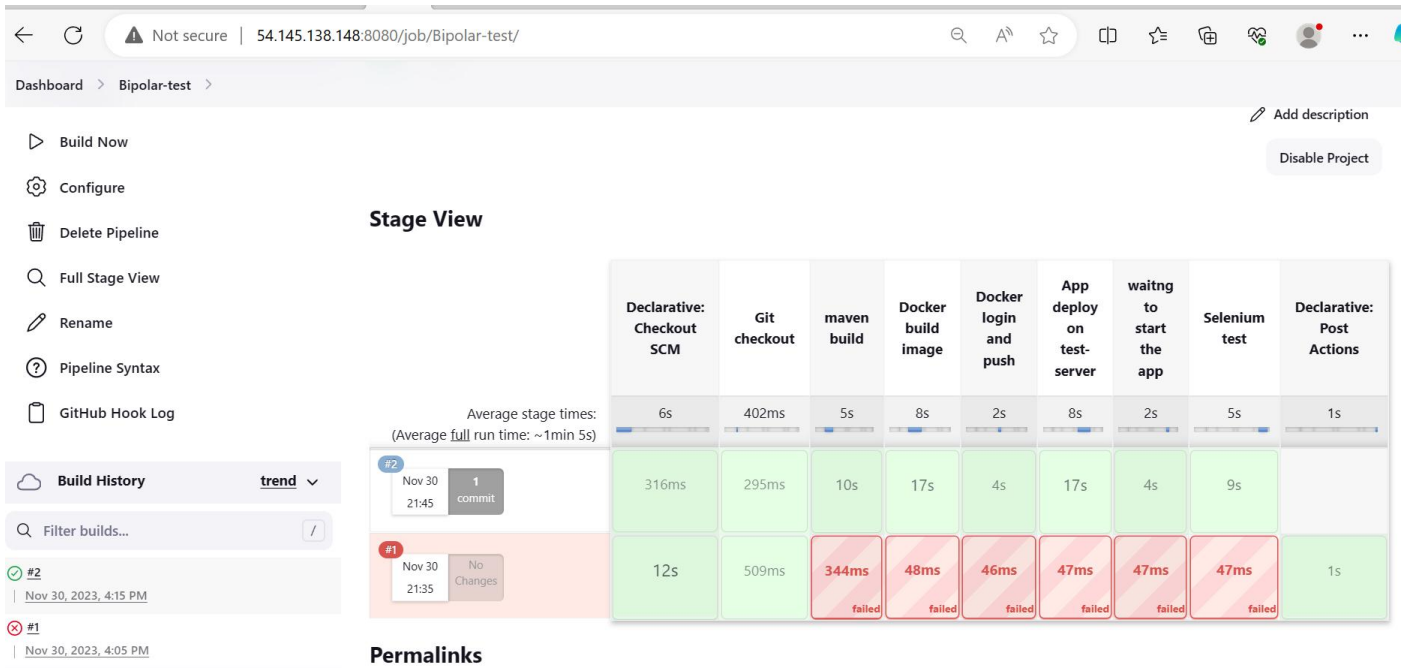


- After Rectification of Error in pipeline and after a new push, webhook triggered the job and run the pipeline successfully as below.

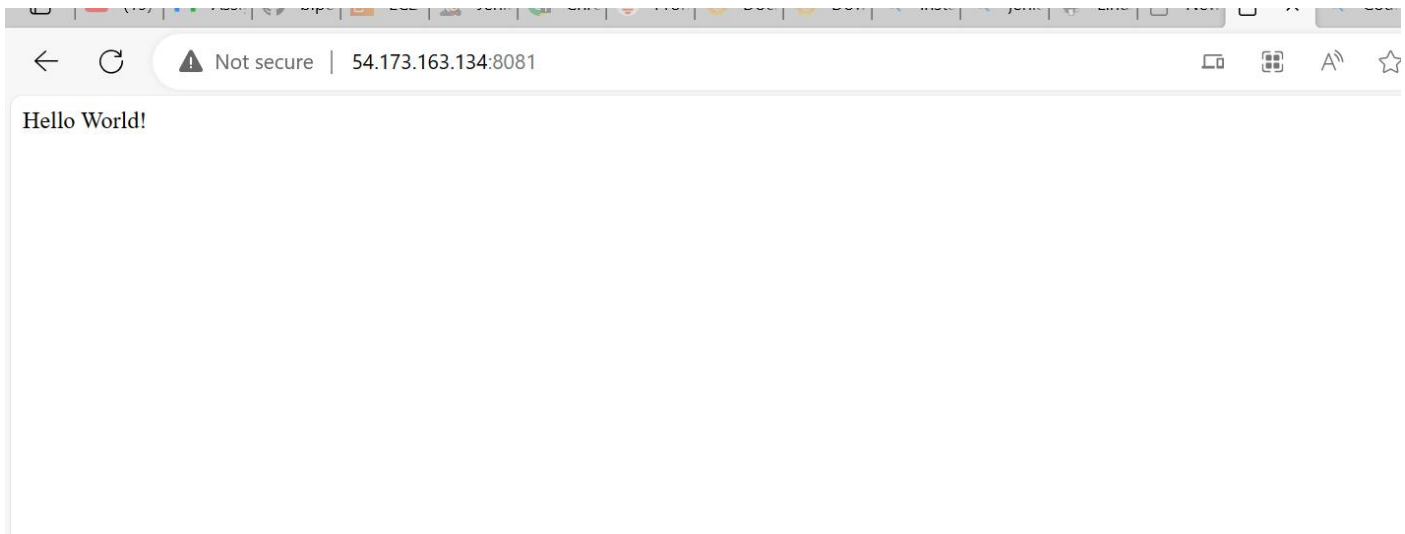
Last GitHub Push

```
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



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

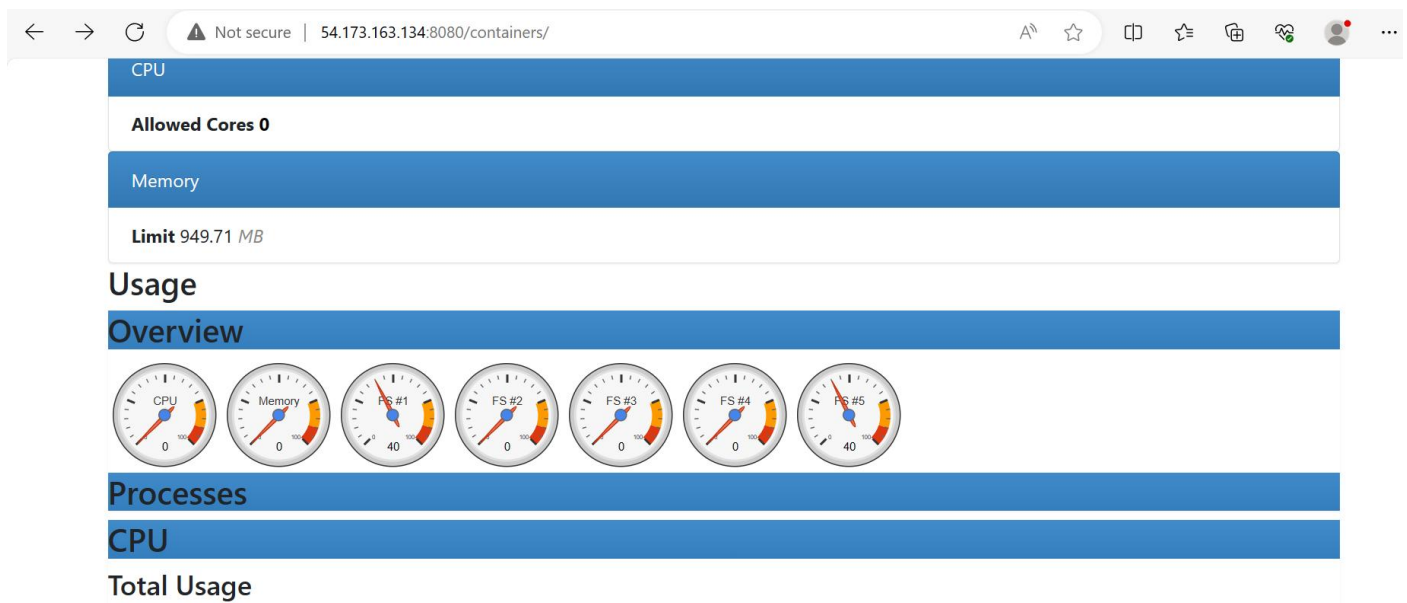
```
ubuntu@ip-172-31-82-104: ~  
ubuntu@ip-172-31-35-204: ~  
ubuntu@ip-172-31-35-204:~/docker-monitoring$ sudo docker ps -a  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS              PORTS  
2649490af416   revanthkumar9/bipolar:2.0          "java -jar /app.jar"    2 hours ago   Up 2 hours         0.0.0.0:8081->8080/tcp, :::8081->8080/tcp  
076bd99f0be5   gcr.io/cadvisor/cadvisor           "/usr/bin/cadvisor -..." 8 hours ago   Up 8 hours (healthy) 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  
ubuntu@ip-172-31-35-204:~/docker-monitoring$
```

➤ Configured Monitoring and logging :-

- Deployed application container is monitored by using prometheus , grafana and cAdvisor.
- First cAdvisor 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.

```
ubuntu@ip-172-31-82-104: ~ × ubuntu@ip-172-31-35-204: ~/ × + v
#prometheus.yml - file
#this must be in the same directory as of docker-compose.yml file

scrape_configs:
- job_name: Docker_containers
  scrape_interval: 5s
  static_configs:
  - targets:
    - 54.173.163.134:8080
```

- In Jenkins-server Prometheus and grafana are launched as a docker container to stop 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.

The screenshot shows the Prometheus web interface at the URL `54.145.138.148:9090/targets?search=`. The page title is "Targets". Below the title, there are tabs for "All", "Unhealthy", and "Collapse All". A search bar is present with the placeholder text "Filter by endpoint or labels". On the right, there are status indicators: "Unknown", "Unhealthy", and "Healthy".

The main content area shows a table with the following data:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://54.173.163.134:8080/metrics	UP	instance="54.173.163.134:8080" job="Docker_containers" Discovered labels: __address__="54.173.163.134:8080" __metrics_path__="/metrics" __scheme__="http" __scrape_interval__="5s" __scrape_timeout__="5s" job="Docker_containers"	22.49s ago	51.792ms	

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

The screenshot shows the Grafana web interface at the URL `54.145.138.148:3000/?orgId=1`. The page title is "Welcome to Grafana". Below the title, there is a search bar and a "Need help?" link. The main content area features a "Basic" section with a "TUTORIAL DATA SOURCE AND DASHBOARDS" card. This card contains the text "Grafana fundamentals" and "Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the 'Data source' and 'Dashboards' steps to the right." Below this text is a "Grafana" logo.

There are also two "COMPLETE" cards:

- "Add your first data source" with a database icon and a link to "Learn how in the docs".
- "Create your first dashboard" with a dashboard icon and a link to "Learn how in the docs".

At the bottom, there is a "Dashboards" section with "Starred dashboards" and a "Latest from the blog" section with a date "Nov 30".

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

