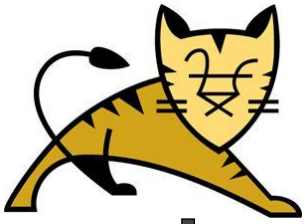
Project-1

DEPLOY AN NETFLIX APPLICATION IN TOMCAT SERVER USING CI/CD

TOOL USED

- **GIT: To maintain the source code.**
- **HCP: To manage and orchestrate cloud-based resources and infrastructure.**
- **JENKINS: To Integrate all the tools.**
- **MAVEN: To build the source code.**
- **NEXUS: To store the artifact.**
- **TOMCAT: Webserver to deploy an application.**
- **PROMETHEUS & GRAFANA: For monitoring and visualizing system metrics.**


EC2
AWS



STEP-1: CREATE THE INFRASTRUCTURE FROM HCP.

Link: https://github.com/Lalit-Mahajan/jenkins-java-project.git

Go to the HCP website

Sign up or log in to your HCP account

Create a github account Create repo -- > name -- > add new file -- > write terraform code -- > commit

Create an organization

Create your workspace

Integrate your vcs -- > github -- > selecet repo -- > next -- > continue

Add varaibles -- >

Aws_access_key_id : sensitive -- > save

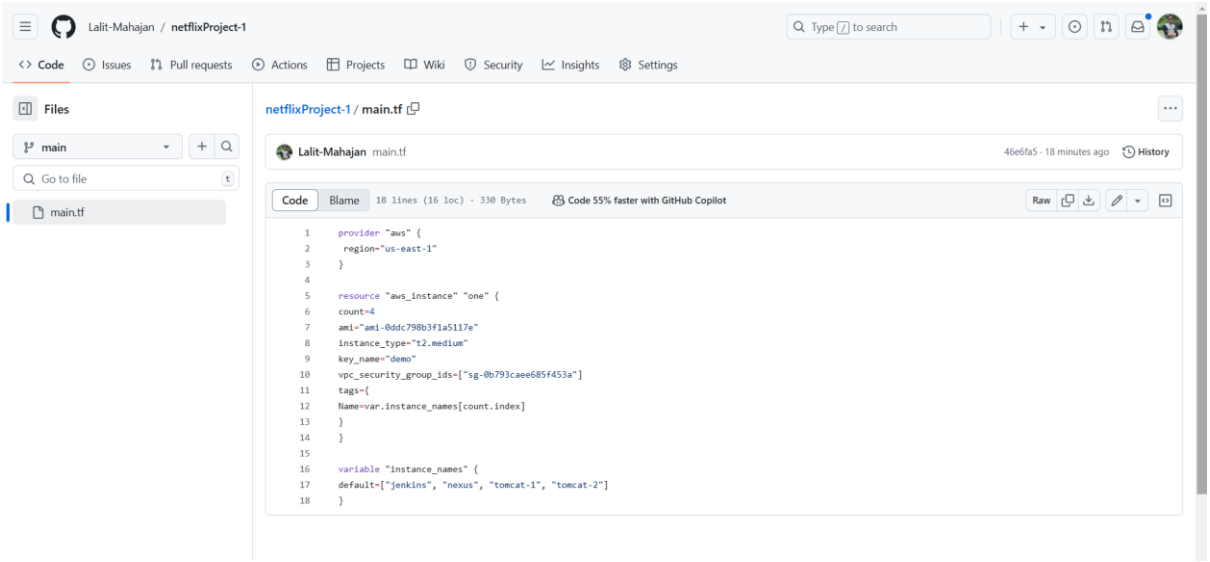
Aws_secret_access_key: sensitive -- > save

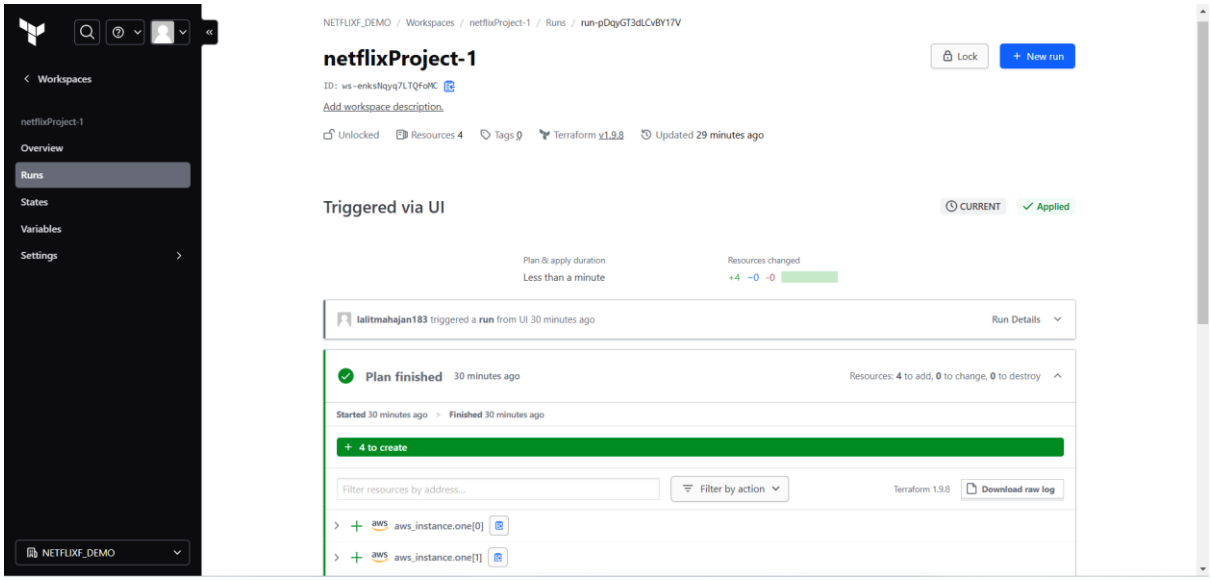
Note: mark them as env varibale and make sure no spaces are given

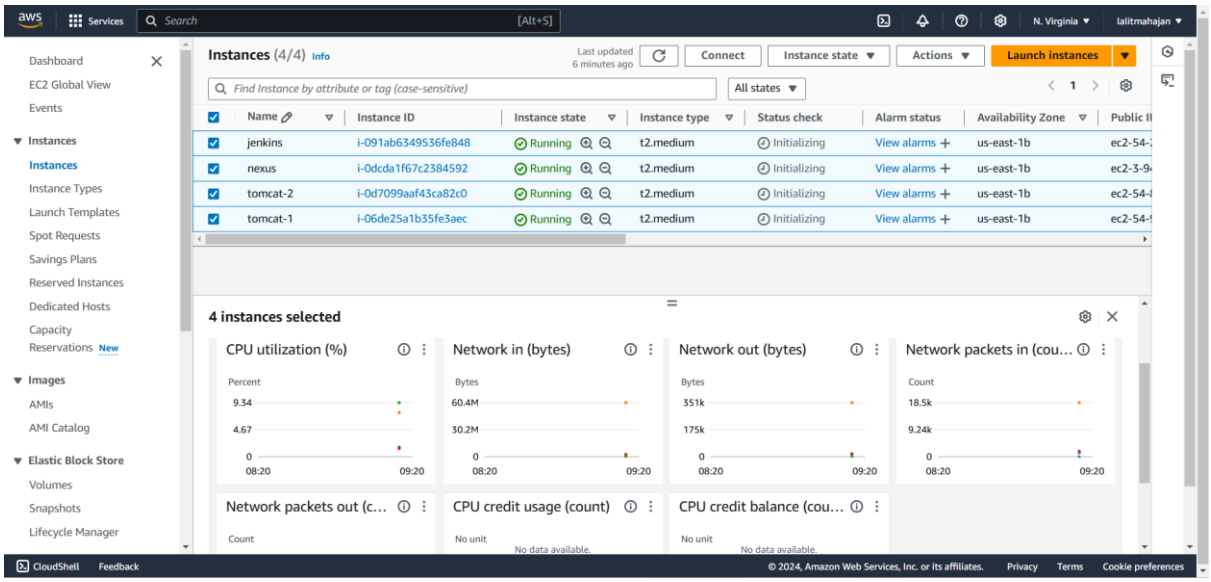
Runs -- > new run -- > start -- > confirm and apply

It will automatically plan & we need to apply by manual Second time when we change code it will automatically plan Plan & apply

Code to create infrastructure:







STEP-2: CONFIGURE THE SEVERS [SCRIPTS] [JENKINS, TOMCAT, NEXUS]

Link: https://github.com/RAHAMSHAIK007/all-setups.git

By using the scripts we can configure these Jenkins,tomcat ,nexus and grafana
vim Jenkins.sh → sh jenkins.sh (Jenkins server)

vim tomcat.sh → sh tomcat.sh (tomcat server)

vim nexus.sh → sh nexus.sh (nexus server)

vim pegion.sh → sh npegion.sh (monitoring server)

STEP-3: CREATE REPOSITORY IN NEXUS

Login to Nexus: Access the Nexus Repository Manager via your web browser.

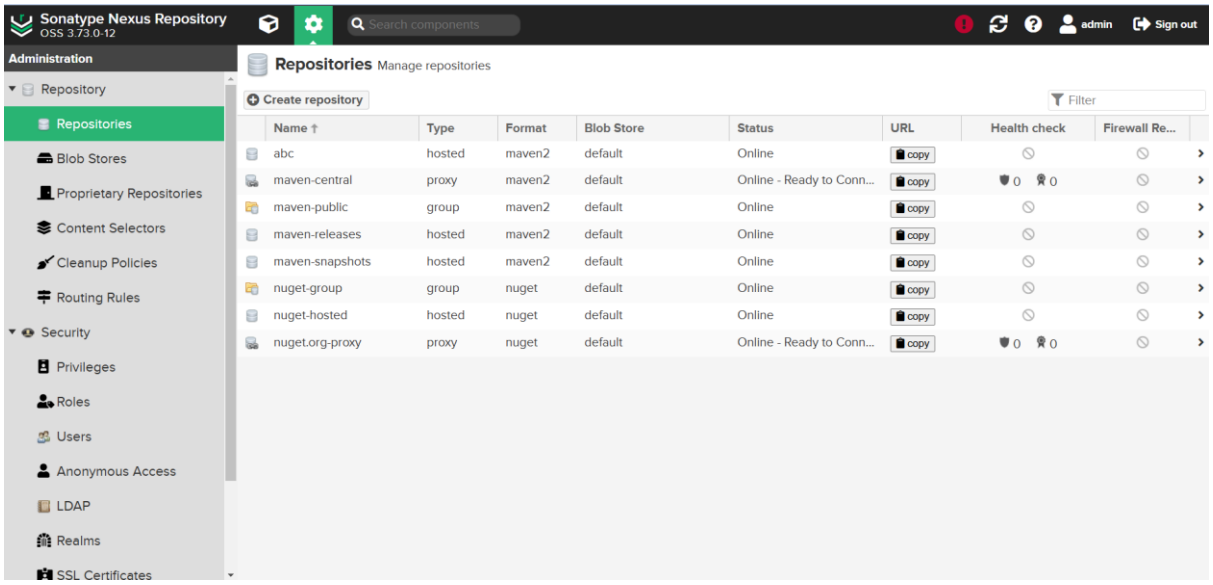
Navigate to Repositories: In the left-hand menu →settings symbol, go to "Repository" and select "Repositories".

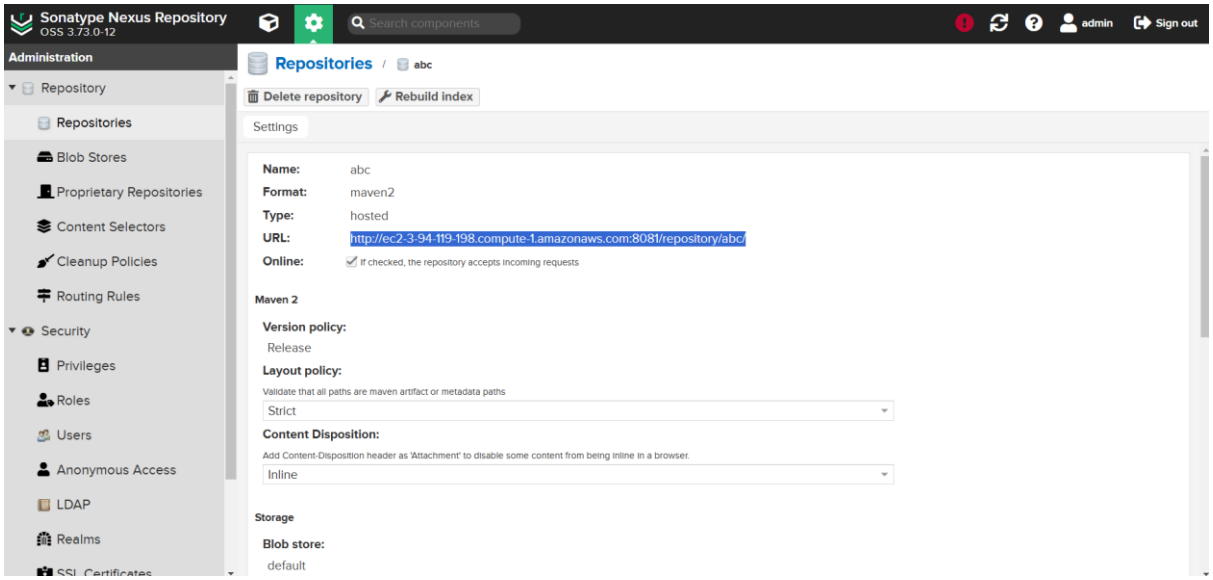
Create Repository: Click on the "Create repository" button.

Choose Repository Type: Select the type of repository maven2(hosted).

Configure Repository: Provide a name, configure access (public/private), and set any other necessary settings.

Save: Click "Save" to create the repository.





SETP-4: WRITE PIPELINE FOR CI/CD.

Download Plugins [pipeline stage view, nexus, Deploy to container, slack]

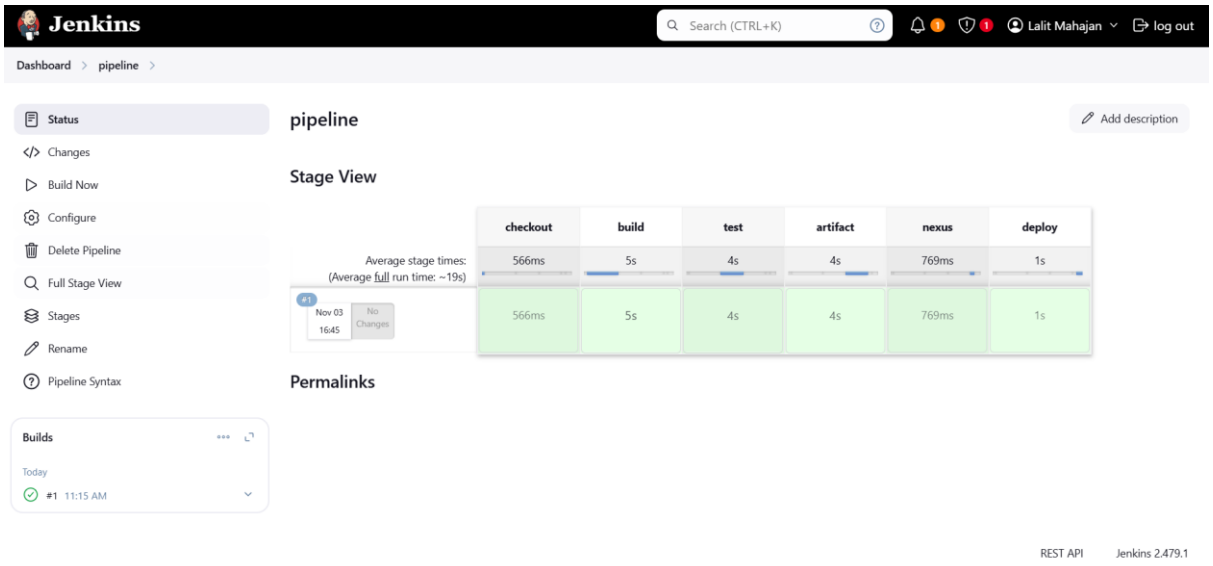
create pipeline—run the below pipeline script

```
pipeline {
  agent any


  stages {
    stage('checkout') {
      steps {
        git branch: '$branch', url: 'https://github.com/Lalit-Mahajan/jenkins-java-project.git'
      }
    }
    stage('build') {
      steps {
        sh 'mvn compile'
      }
    }
    stage('test') {
      steps {
```

```
            sh 'mvn test'
        }
    }
    stage('artifact') {
        steps {
            sh 'mvn package'
        }
    }
    stage('nexus') {
        steps {
            nexusArtifactUploader artifacts: [[artifactId: 'NETFLIX', classifier: '', file: 'target/NETFLIX-1.2.2.war', type: '.war']], credentialsId: '7e474c88-9310-42e5-8ac4-4e16feea9633', groupId: 'in.RAHAM', nexusUrl: 'ec2-3-94-119-198.compute-1.amazonaws.com:8081/', nexusVersion: 'nexus3', protocol: 'http', repository: 'abc', version: '1.2.2'
        }
    }
    stage('deploy') {
        steps {
            deploy adapters: [
                tomcat9(
                    credentialsId: '06fd25e8-80b4-4702-b79e-b0ebd46d0ae0',
                    path: '',
                    url: 'http://ec2-54-91-251-78.compute-1.amazonaws.com:8080/'
                    )
                ],
                contextPath: 'netflix',
                war: 'target/*.war'
        }
    }
}
post {
    always {
```

```
        echo 'Slack Notifications'
        slackSend (
          channel: '#netflixapp', message: "*${currentBuild.currentResult}:* Job ${env.JOB_NAME} \n build ${env.BUILD_NUMBER} \n More info at: ${env.BUILD_URL}"
          )
      }
  }
}
```



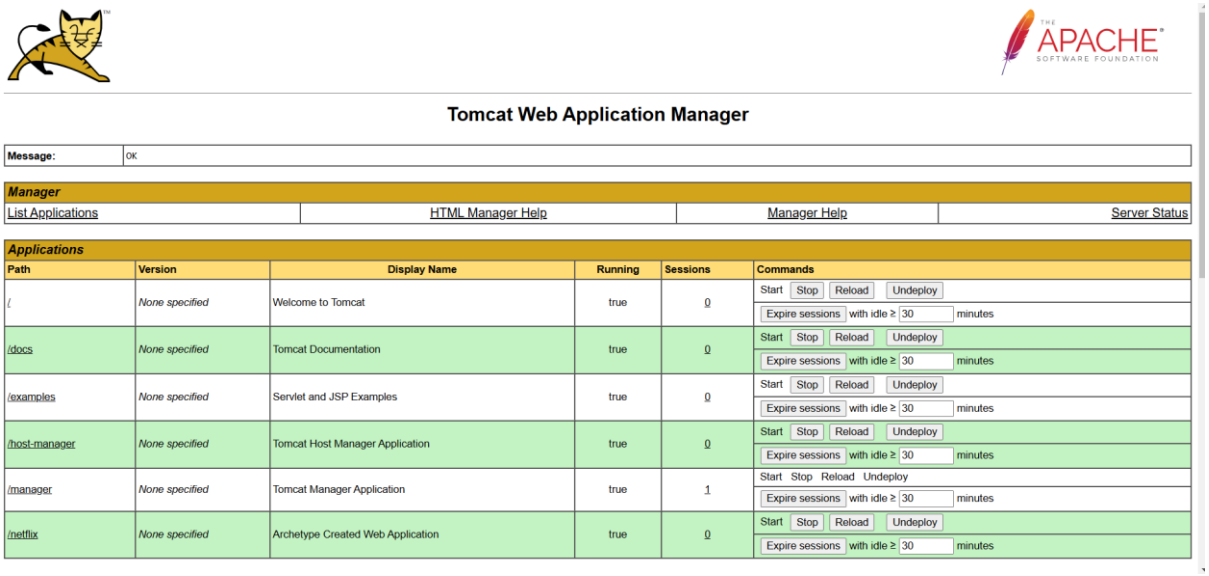
STEP-5:TOMCAT:

Copy Public IP: Use the public IP address of the server where Tomcat is hosted, followed by the port number (default is 8080), e.g., http://<public-ip>:8080.

Login to Manager:

Go to http://<public-ip>:8080/manager/html in your browser.

Enter the username and password for the Tomcat Manager application.



Tomcat Web Application Manager

Message: OK

Manager

List Applications | HTML Manager Help | Manager Help | Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/netflix	None specified	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

STEP-6: MONITOR THE APPLICATION FROM GRAFANA

PORT: 3000

username & passowrd: admin & admin

CONNECTING PROMETHEUS TO GARAFANA:

connect to grafana dashboard -- > Data source -- > add -- > promethus -- > url of prometheus -- > save & test -- > top of page -- >click on + symbol -- > import dashboard -- > 1860 -- > laod --- > prometheus -- > import

here we connect to tomat sever by using

vim /etc/hosts

public-ip node1 worker-1

Prometheus Alerts Graph Status Help

Targets

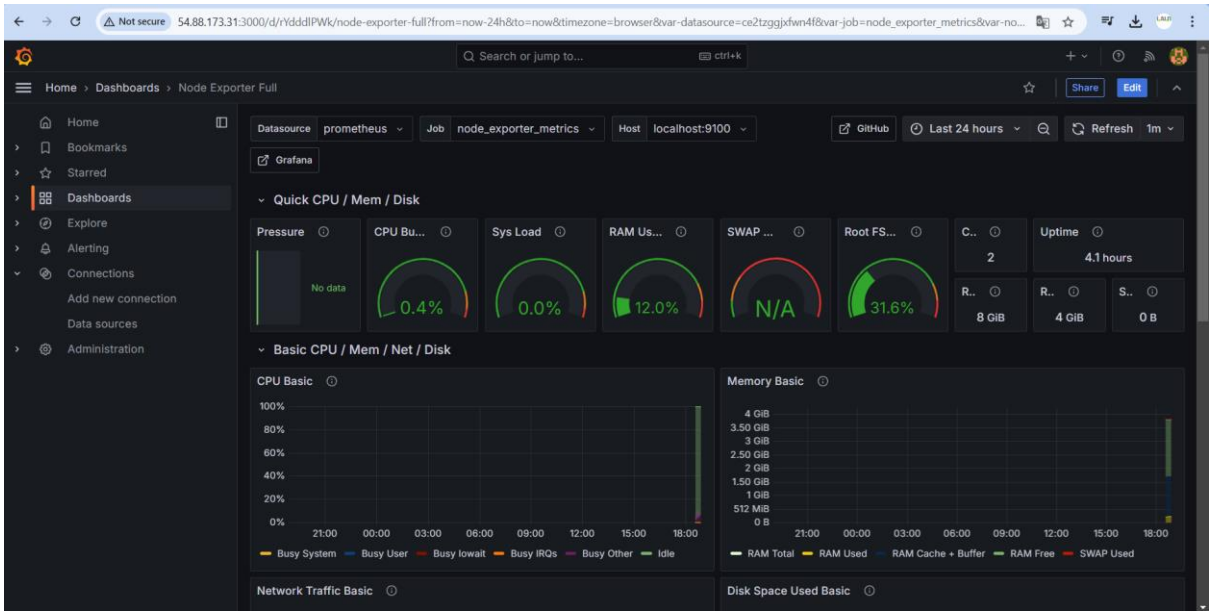
All scrape pools | All | Unhealthy | Collapse All | Filter by endpoint or labels | Unknown | Unhealthy | Healthy

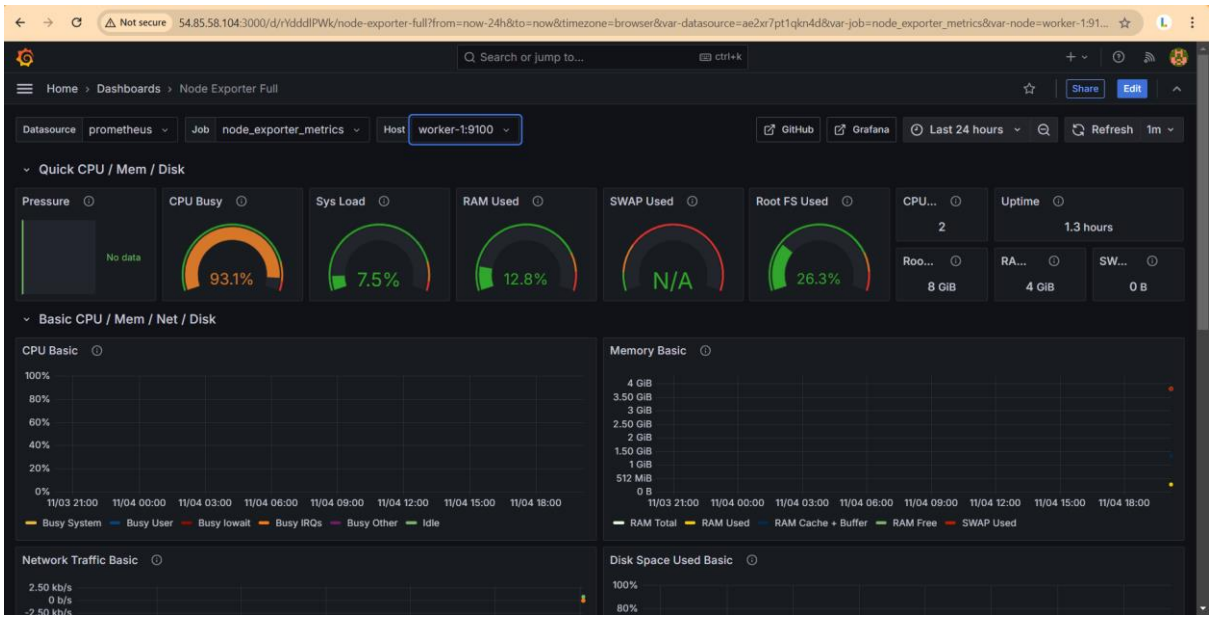
node_exporter_metrics (1/3 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter_metrics"	4.889s ago	7.822ms	
http://worker-1:9100/metrics	DOWN	instance="worker-1:9100" job="node_exporter_metrics"	2.354s ago	2.047ms	Get "http://worker-1:9100/metrics": dial tcp: lookup worker-1 on 172.31.0.2:53: no such host
http://worker-2:9100/metrics	DOWN	instance="worker-2:9100" job="node_exporter_metrics"	2.683s ago	4.946ms	Get "http://worker-2:9100/metrics": dial tcp: lookup worker-2 on 172.31.0.2:53: no such host

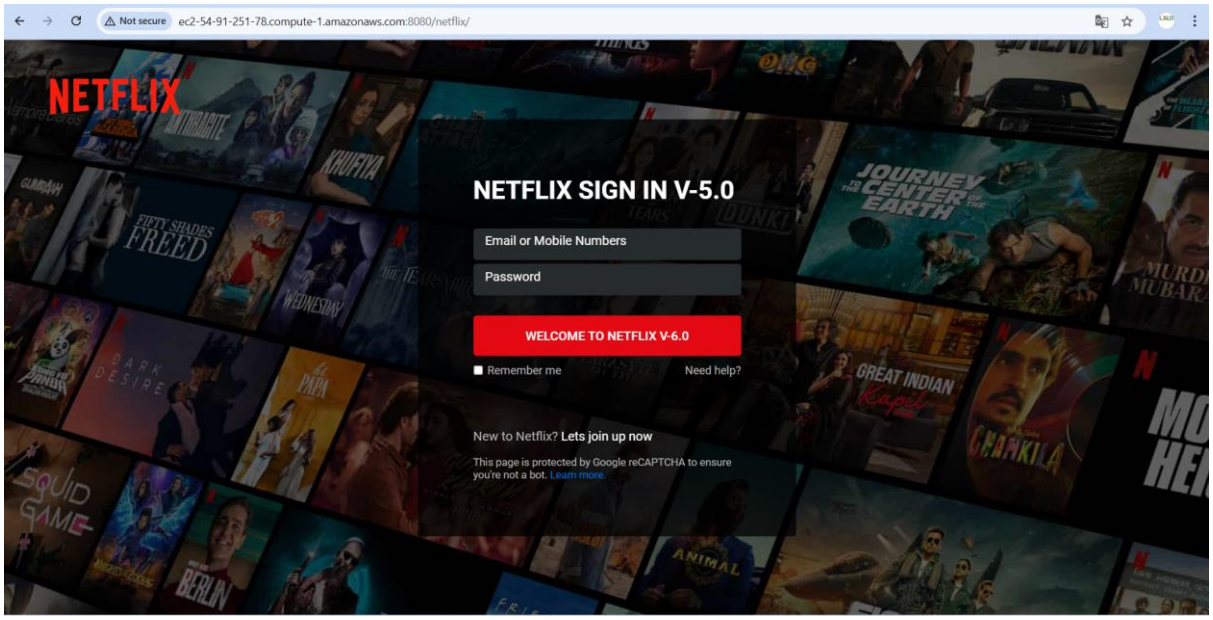
prometheus_metrics (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus_metrics"	4.442s ago	2.933ms	





OUTPUT:


Not secure
ec2-54-91-251-78.compute-1.amazonaws.com:8080/netflix/
NETFLIX
NETFLIX SIGN IN V-5.0
Email or Mobile Numbers
Password
WELCOME TO NETFLIX V-6.0
Remember me
Need help?
New to Netflix? Lets join up now
This page is protected by Google reCAPTCHA to ensure you're not a bot. Learn more.