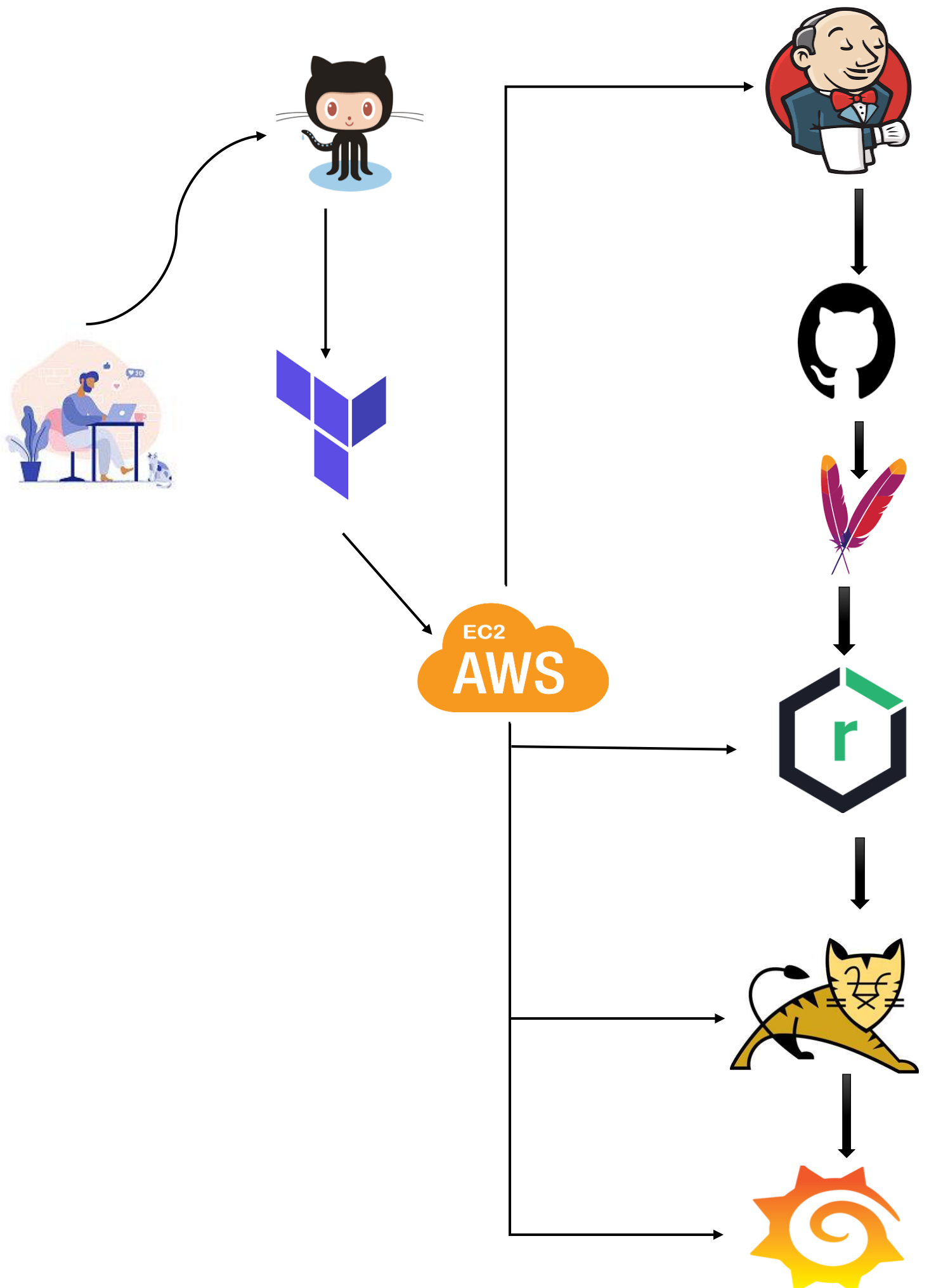


# Project-1

## 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.



## 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 -- > select repo -- > next -- > continue

Add variables -- >

Aws\_access\_key\_id : sensitive -- > save

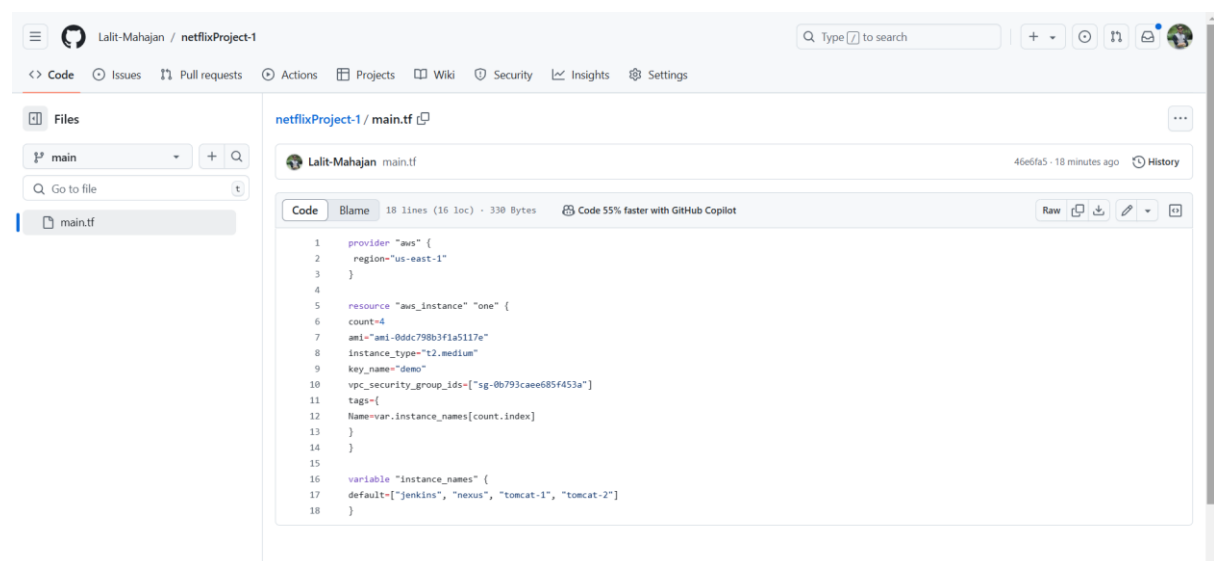
Aws\_secret\_access\_key: sensitive -- > save

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

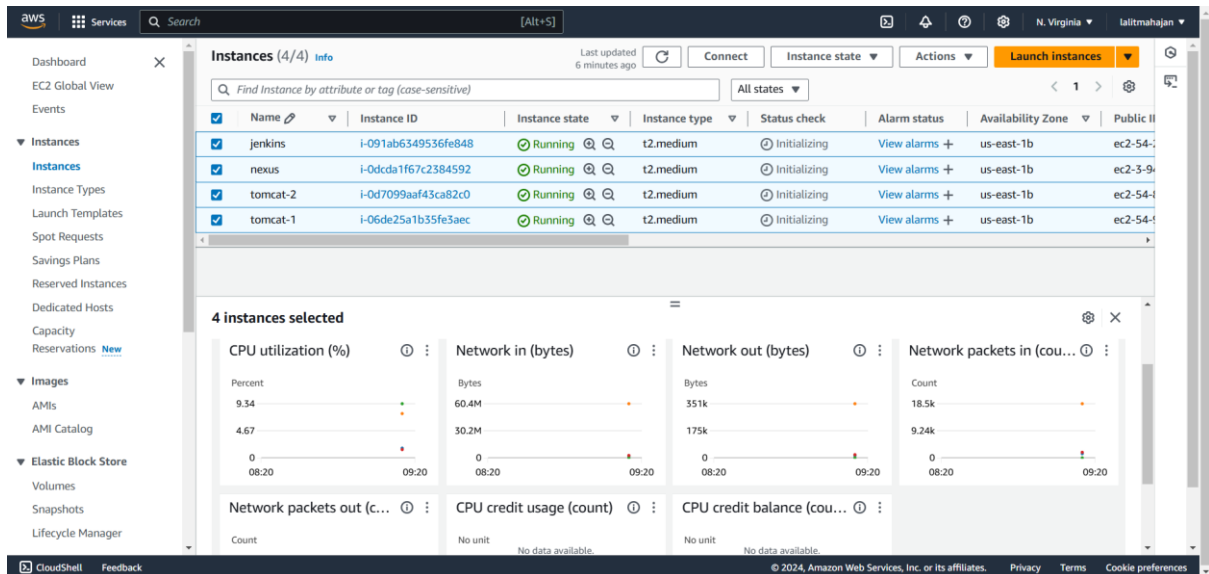
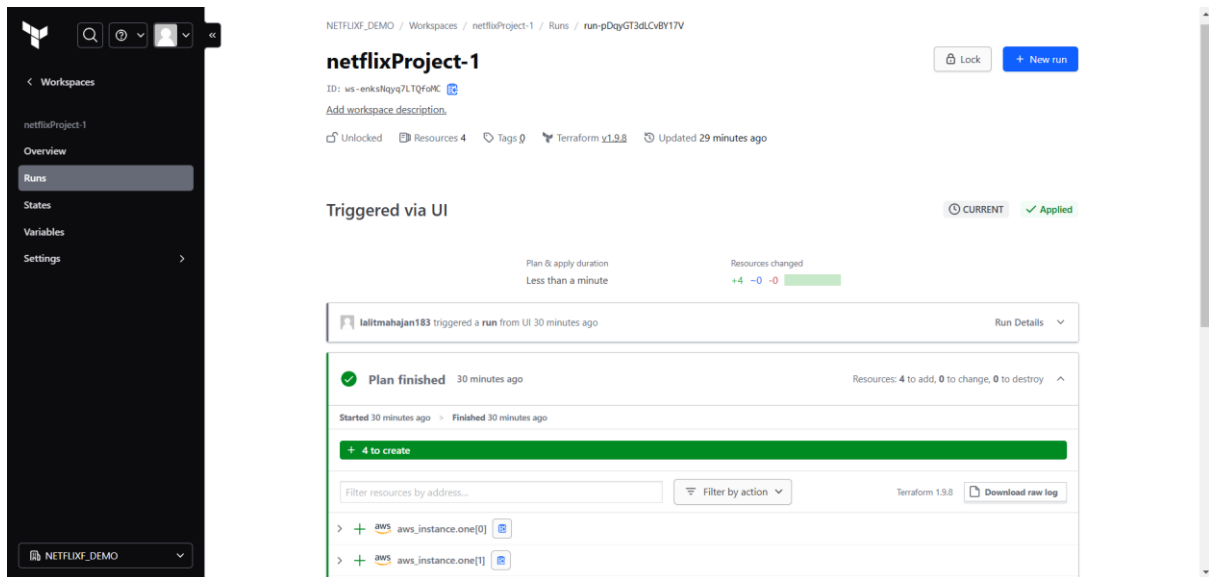
Runs -- > new run -- > start -- > confirm and apply

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:



```
1 provider "aws" {
2   region="us-east-1"
3 }
4
5 resource "aws_instance" "one" {
6   count=4
7   ami="ami-0ddc798b3f1a5117e"
8   instance_type="t2.medium"
9   key_name="demo"
10  vpc_security_group_ids=["sg-0b793caee685f453a"]
11  tags={
12    Name=var.instance_names[count.index]
13  }
14 }
15
16 variable "instance_names" {
17   default=["jenkins", "nexus", "tomcat-1", "tomcat-2"]
18 }
```



## 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 pegin.sh → sh npegin.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.

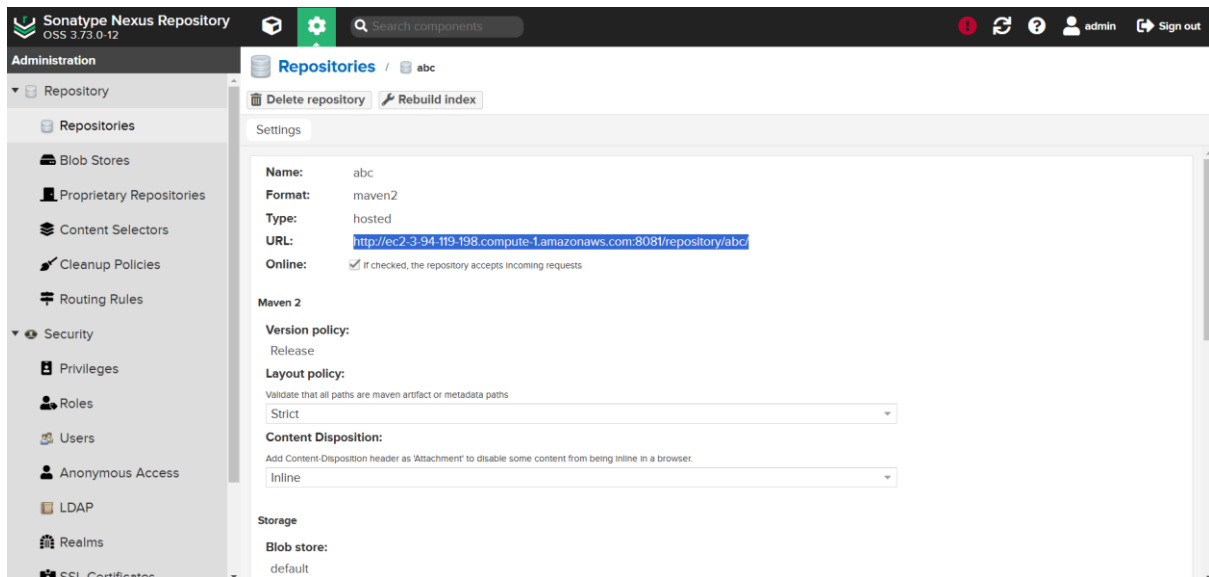
Choose Repository Type: Select the type of repository maven2(hosted).

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

Save: Click "Save" to create the repository.

The screenshot shows the Sonatype Nexus Repository OSS 3.73.0-12 interface. The left-hand menu is open, showing the 'Administration' section with 'Repository' selected. Under 'Repository', 'Repositories' is highlighted. The main area displays a table of existing repositories. The table has columns: Name, Type, Format, Blob Store, Status, URL, Health check, and Firewall Re... A 'Create repository' button is visible at the top left of the table area.

Name	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
abc	hosted	maven2	default	Online			
maven-central	proxy	maven2	default	Online - Ready to Conn...		0  0	
maven-public	group	maven2	default	Online			
maven-releases	hosted	maven2	default	Online			
maven-snapshots	hosted	maven2	default	Online			
nuget-group	group	nuget	default	Online			
nuget-hosted	hosted	nuget	default	Online			
nuget.org-proxy	proxy	nuget	default	Online - Ready to Conn...		0  0	



## 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 {

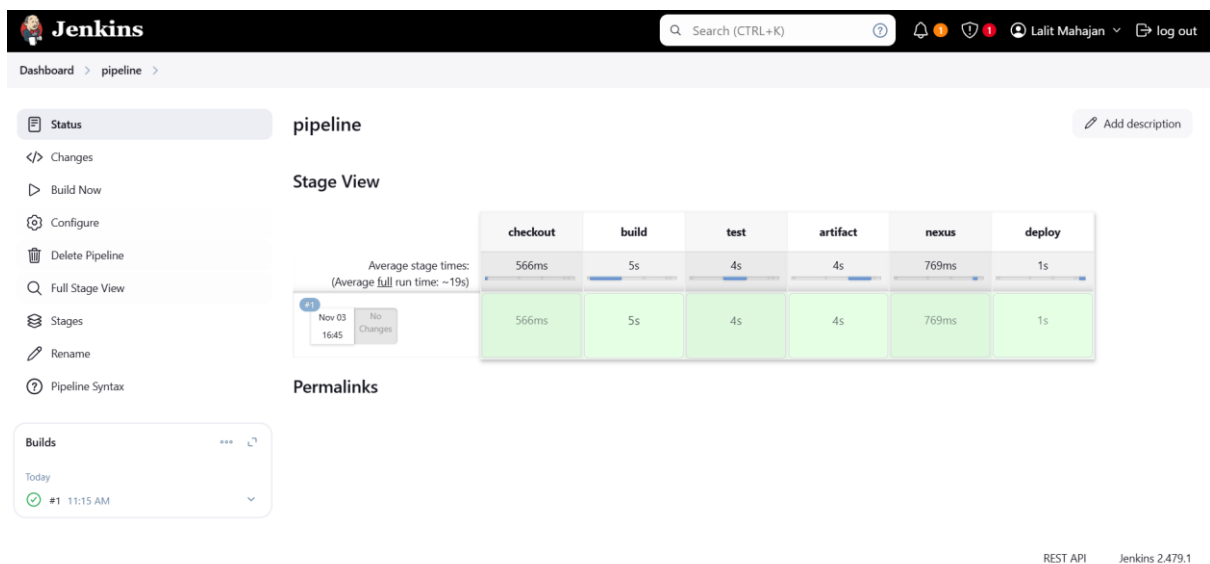
```

```

    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}"
    )
}
}
}
}
}

```



The screenshot shows the Jenkins web interface for a pipeline named 'pipeline'. The top navigation bar includes the Jenkins logo, a search bar, and user information (Lalit Mahajan). The left sidebar contains navigation links: Status (selected), Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main content area displays the 'Stage View' for the 'pipeline' job. It shows a table of stages with their respective durations and a 'Builds' section at the bottom.

Stage	checkout	build	test	artifact	nexus	deploy
Average stage times:	566ms	5s	4s	4s	769ms	1s
Average full run time: ~19s	566ms	5s	4s	4s	769ms	1s

**Builds**

Today

✓ #1 11:15 AM

REST API Jenkins 2.479.1



## 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	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>
/manager	None specified	Tomcat Manager Application	true	1	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>
/netflix	None specified	Archetype Created Web Application	true	0	<div><div>Start</div><div>Stop</div><div>Reload</div><div>Undeploy</div></div> <div>Expire sessions with idle ≥ 30 minutes</div>

## STEP-6: MONITOR THE APPLICATION FROM GRAFANA

PORT: 3000

username & passowrd: admin & admin

CONNECTING PROMETHEUS TO GARAFANA:

connect to grafana dashboard --> Data source --> add --> prometheus --> 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

## Targets

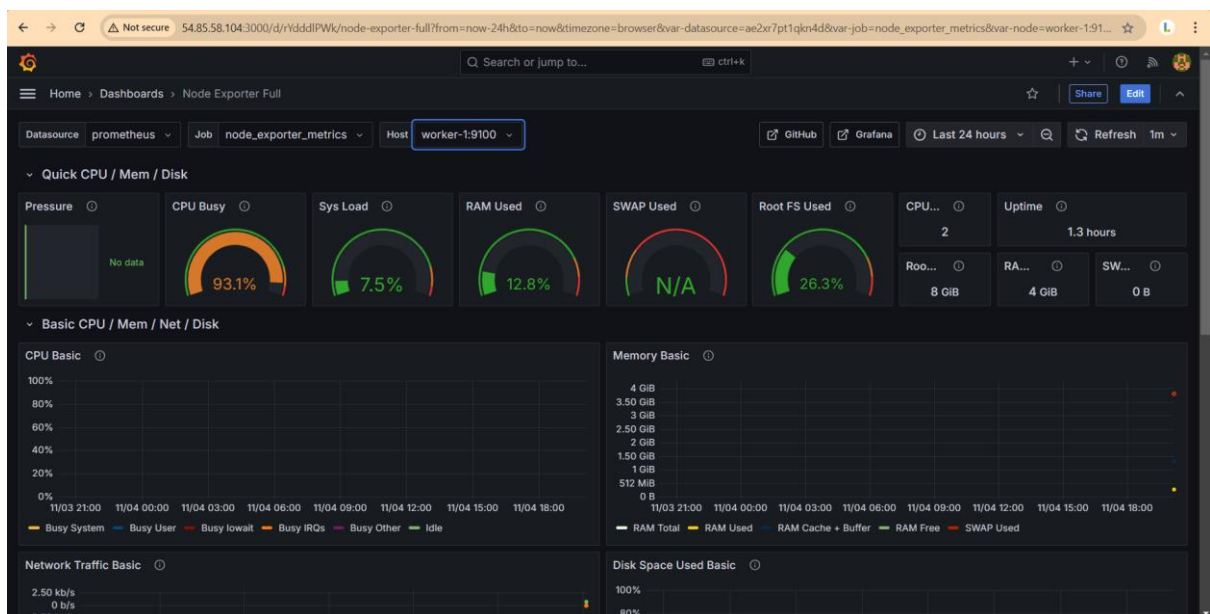
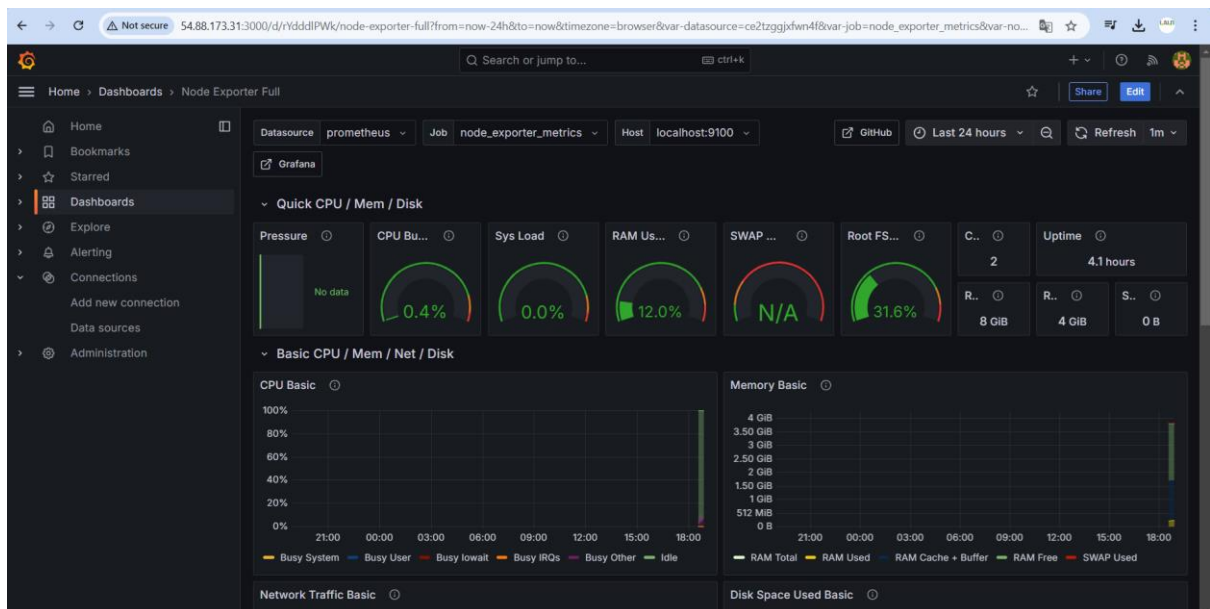
All scrape pools All Unhealthy Collapse All  Unknown Unhealthy Healthy

### node\_exporter\_metrics (1/3 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9100/metrics">http://localhost:9100/metrics</a>	UP	instance="localhost:9100" job="node_exporter_metrics"	4.889s ago	7.822ms	
<a href="http://worker-1-9100/metrics">http://worker-1-9100/metrics</a>	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
<a href="http://worker-2-9100/metrics">http://worker-2-9100/metrics</a>	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
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	instance="localhost:9090" job="prometheus_metrics"	4.442s ago	2.933ms	



**OUTPUT:**

