

Task : DevSecOps CI/CD : Deploying a Secure Netflix Clone

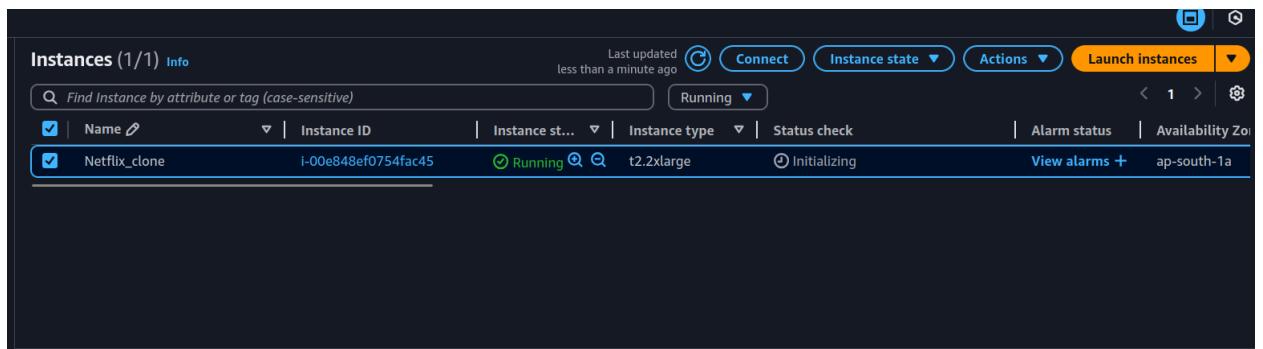
Prerequisites

- AWS account setup
- Basic knowledge of AWS services
- Understanding of DevSecOps principles
- Familiarity with Docker, Jenkins, Java, SonarQube, AWS CLI, Kubectl, and Terraform, Trivy
- Account on TMDB and a API key

Steps:

Step 1: Create an EC2 Instance.

- Creating an EC2 instance with Ubuntu AMI, t2.x2large, and 30 GB storage



- Assigning an IAM role with Admin access for learning purposes



Identity and Access Management (IAM)

IAM > Roles

Roles (2) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-

Roles Anywhere Info

Manage

For permission policy select Administrator Access (Just for learning purpose), click Next.

IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Add permissions Info

Permissions policies (1/897) Info

Choose one or more policies to attach to your new role.

Policy name	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...

Go to EC2 Dashboard and select the instance.

Click on Actions → Security → Modify IAM role.

EC2 Dashboard

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input checked="" type="checkbox"/> Test	i-0c92d797a5813a128	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	192.31.1

Actions ▾

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security** ▾
- Get Windows password
- Image and templates
- Monitor and troubleshoot

Launch instances ▾

EC2 > Instances > i-0b96bd49273a973ef > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID: i-0b96bd49273a973ef (Netflix_clone)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Devops_projects

Create new IAM role

Cancel Update

Step 2: Installation of Required Tools on the Instance

Give Necessary permission for the scripts for execution.

1. Script 1 for Java,Jenkins,Docker

```
#!/bin/bash
sudo apt update -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packag
sudo apt update -y
sudo apt install temurin-17-jdk -y
/usr/bin/java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
#install docker
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg -y
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
# Add the repository to Apt sources:
echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-b
sudo usermod -aG docker ubuntu
newgrp docker
```

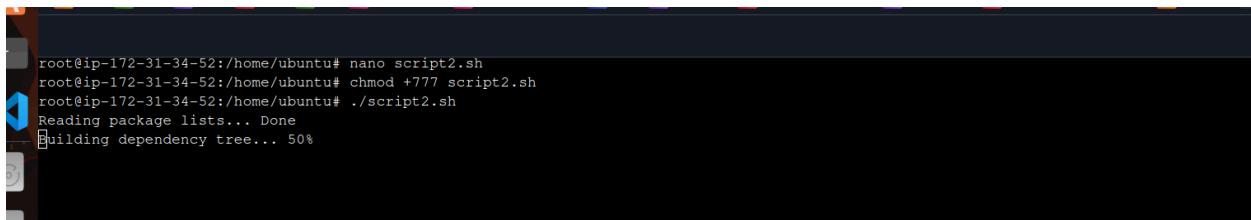
```
System information as of Sat Nov 23 11:18:09 UTC 2024
root@ip-172-31-34-52:/home/ubuntu# nano script1.sh
root@ip-172-31-34-52:/home/ubuntu# chmod +777 script1.sh
root@ip-172-31-34-52:/home/ubuntu# ./script1.sh
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [670 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [157 kB]
```

2. Script 2 for Terraform,kubectl,Aws cli

```
#!/bin/bash
#install terraform
sudo apt install wget -y
wget -O https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmo
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.
sudo apt update && sudo apt install terraform

#install Kubectl on Jenkins
sudo apt update
sudo apt install curl -y
curl -L0 https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/r
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client

#install Aws cli
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
sudo apt-get install unzip -y
unzip awscliv2.zip
sudo ./aws/install
```



The screenshot shows a terminal window with a black background and white text. It displays the command-line session of a root user on an Ubuntu system. The user has run the script2.sh file, which installs Terraform, Kubectl, and AWS CLI. The output shows the progress of package installations, including reading package lists and building dependency trees.

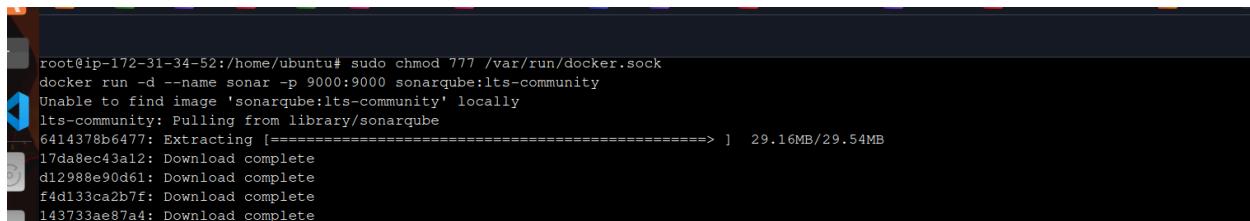
```
root@ip-172-31-34-52:/home/ubuntu# nano script2.sh
root@ip-172-31-34-52:/home/ubuntu# chmod +777 script2.sh
root@ip-172-31-34-52:/home/ubuntu# ./script2.sh
Reading package lists... Done
Building dependency tree... 50%
```

see whether it's installed docker, Terraform, Aws cli, Kubectl or not.

`docker --version ,aws --version,terraform --version,kubectl version`

Step 3: Now run the Sonarqube container.

```
sudo chmod 777 /var/run/docker.sock  
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

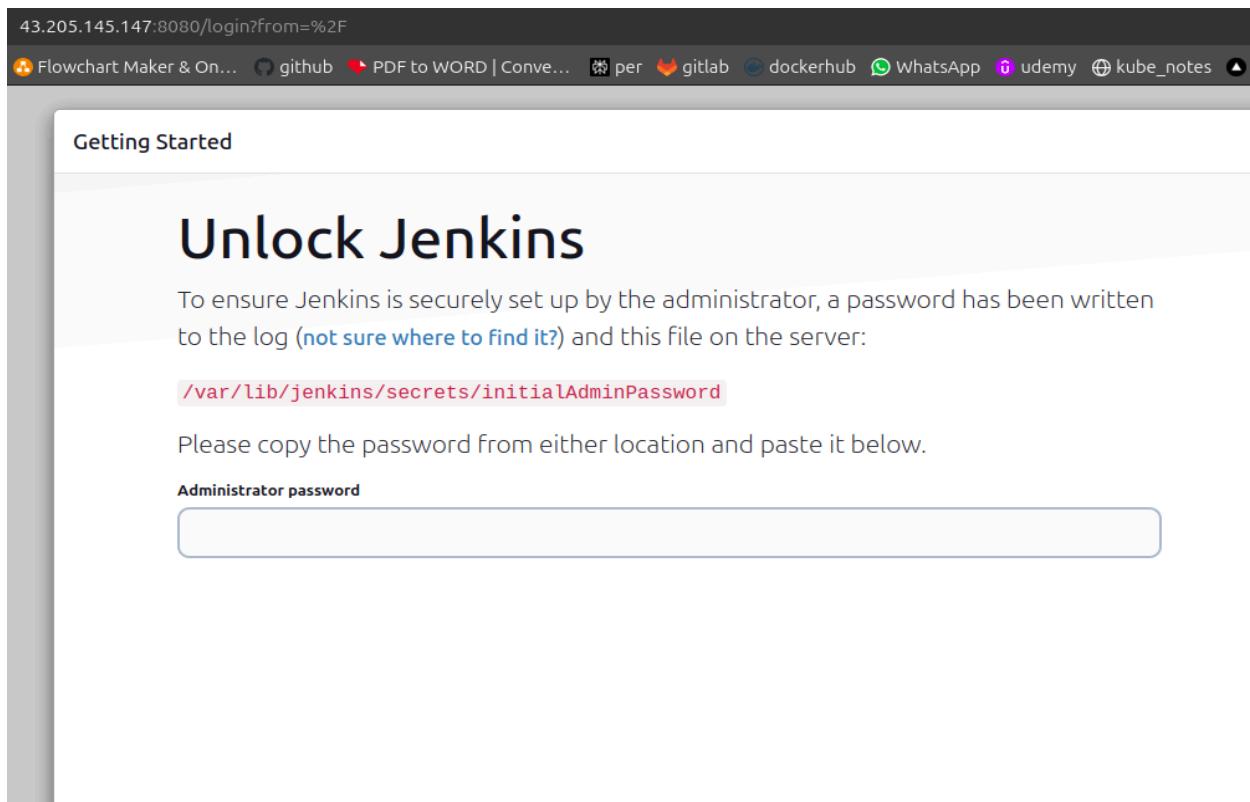


```
root@ip-172-31-34-52:/home/ubuntu# sudo chmod 777 /var/run/docker.sock  
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community  
Unable to find image 'sonarqube:lts-community' locally  
lts-community: Pulling from library/sonarqube  
6414378b6477: Extracting [=====>] 29.16MB/29.54MB  
17da8ec43a12: Download complete  
d12988e90d61: Download complete  
f4d133ca2b7f: Download complete  
143733ae87a4: Download complete
```

Step 4: Jenkins Setup

Now copy the public IP address of ec2 and paste it into the browser

Ec2-ip:8080 #you will Jenkins login page



```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

```
ubuntu@ip-172-31-33-57:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
0ed1cb07ea7447c5a47d723022e74968
ubuntu@ip-172-31-33-57:~$ █
```

Now, install the suggested plugins.

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Create an admin user

43.205.145.147:8080

Flowchart Maker & On... GitHub PDF to WORD | Conve... per gitlab dockerhub WhatsApp udemy kube_notes v

Getting Started

Create First Admin User

Username

seytan

Password

*

Confirm password

*

Full name

E-mail address

Click on save and continue.

Jenkins Dashboard

The screenshot shows the Jenkins dashboard at <http://43.205.145.147:8080>. The main header says "Dashboard". On the left sidebar, there are links for "New Item", "Build History", "Manage Jenkins", and "My Views". Below the sidebar are two dropdown menus: "Build Queue" (No builds in the queue) and "Build Executor Status" (0/2). The central content area has a heading "Welcome to Jenkins!" with a sub-instruction: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." It also features "Create a job" and "Set up a distributed build" buttons.

Step 5: Now Copy the public IP again and paste it into a new tab in the browser with 9000

ec2-ip:9000 #runs sonar container

The screenshot shows the SonarQube login page at http://43.205.145.147:9000/sessions/new?return_to=%2F. The title bar says "sonarQube". The main content is a "Log in to SonarQube" form with fields for "Login" and "Password", and buttons for "Log In" and "Cancel".

Enter username and password, click on login and change password

**username admin
password admin**

This is Sonar Dashboard.

The screenshot shows the SonarQube "Projects" creation page at <http://52.66.140.95:9000/projects/create>. The title bar says "Not secure | 52.66.140.95:9000/projects/create". The main content asks "How do you want to create your project?" and provides options to "Create your project from your favorite DevOps platform". It lists five integration methods: "From Azure DevOps", "From Bitbucket Server", "From Bitbucket Cloud", "From GitHub", and "From GitLab", each with a "Set up global configuration" link.

Step 6: Install Necessary Plugins on jenkins.

Go to Jenkins dashboard → Manage Jenkins → Plugins

The screenshot shows the Jenkins 'Manage Jenkins' section with the 'Plugins' tab selected. A sidebar on the left lists 'Updates', 'Available plugins' (which is currently selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area displays a table of available plugins:

Install	Name	Released
<input checked="" type="checkbox"/>	Terraform 1.0.10	4 yr 9 mo ago
<input checked="" type="checkbox"/>	NodeJS 1.6.2	3 mo 12 days ago
<input checked="" type="checkbox"/>	Pipeline: Stage View 2.34	1 yr 0 mo ago
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.3	4 days 22 hr ago
<input checked="" type="checkbox"/>	Docker 1.7.0	1 mo 9 days ago
<input checked="" type="checkbox"/>	Docker Commons 445.vb_646c962a_34	15 days ago
<input checked="" type="checkbox"/>	Docker Pipeline 500.vc0340686b_54	6 mo 5 days ago

Step 7: Create a Pipeline for setuping the EKS using Terraform in jenkins.

The screenshot shows the Jenkins 'New Item' dialog. It has two tabs: 'Freestyle project' and 'Pipeline'. The 'Pipeline' tab is selected and highlighted with a gray background. The 'Freestyle project' tab is described as 'Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.' The 'Multi-configuration project' tab is described as 'Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.' Below these tabs are buttons for 'Folder' and 'Cancel'.

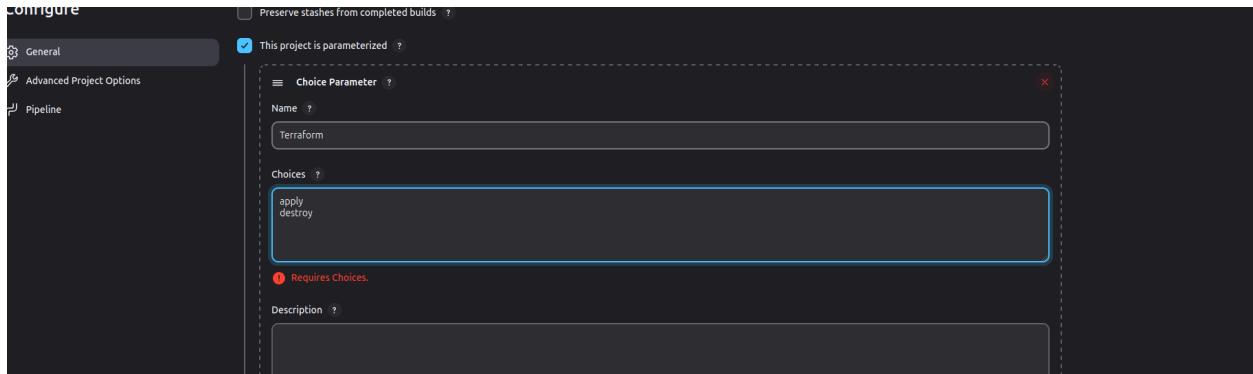
Now come back to Manage Jenkins → Tools

Add the terraform in Tools

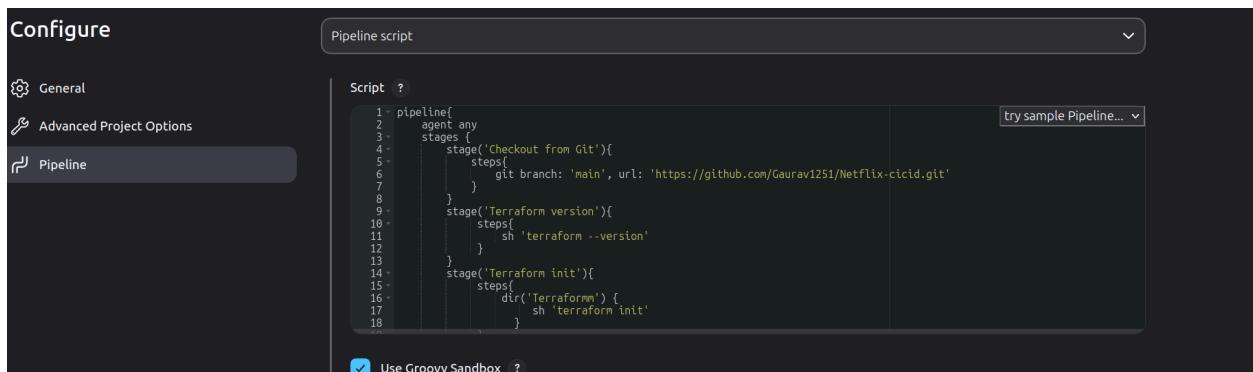
The screenshot shows the Jenkins 'Terraform installations' configuration dialog. It has a 'Terraform' section with a 'Name' field containing 'terraform' and an 'Install directory' field containing '/usr/bin/'. There is a checkbox for 'Install automatically' which is unchecked. At the bottom are 'Save' and 'Apply' buttons.

I want to do this with build parameters to apply and destroy while building only.

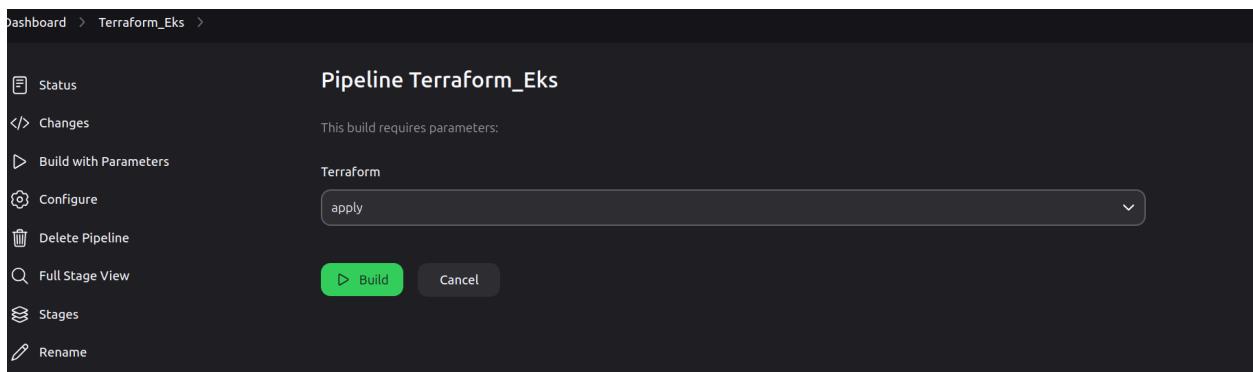
you have to add this inside job like the below image



Now Create a pipeline script for creating an EKS.



let's apply and save and Build with parameters and select action as apply



Stage view it will take max 10mins to provision

The screenshot shows the Jenkins Pipeline interface for a job named "Terraform_Eks". On the left, there's a sidebar with options like Status, Changes, Build with Parameters, Configure, Delete Pipeline, Full Stage View, Stages, and Rename. The main area is titled "Stage View" and displays a table of execution times for various steps:

	Checkout from Git	Terraform version	Terraform init	Terraform validate	Terraform plan	Terraform apply/destroy
Average stage times:	1s	468ms	4s	3s	3s	2min 41s
(Average full run time: ~8min 21s)	706ms	326ms	5s	4s	4s	8min 5s
Nov 23 17:12 No Changes						

Check in Your Aws console whether it created EKS or not.

The screenshot shows the AWS EKS service dashboard under the "Clusters" section. On the left, there's a sidebar with links for Amazon EKS Anywhere, Clusters, Related services (Amazon ECR, AWS Batch), and Console settings. The main area lists a single cluster named "EKS_CLOUD" with the status "Creating".

Step 8: Configure in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(16)→ Click on Apply and Save

The screenshot shows the Jenkins "Tools" configuration page. It's set up to install NodeJS automatically with the name "nodejs" and version "NodeJS 16.14.0". There are also sections for "Install from nodejs.org", "Force 32bit architecture", and "Global npm packages to install".

The top screenshot shows the 'SonarQube Scanner installations' configuration page. It has a 'Name' field set to 'sonar' and a checked 'Install automatically' checkbox. Under 'Install from Maven Central', the 'Version' dropdown is set to 'SonarQube Scanner 6.2.1.4610'. The bottom screenshot shows the 'JDK installations' configuration page, where 'Name' is 'jdk', 'Install automatically' is checked, and 'Version' is set to 'jdk-17.0.7+7' under 'Install from adoptium.net'.

Step 9: Configure Sonar Server in Manage Jenkins

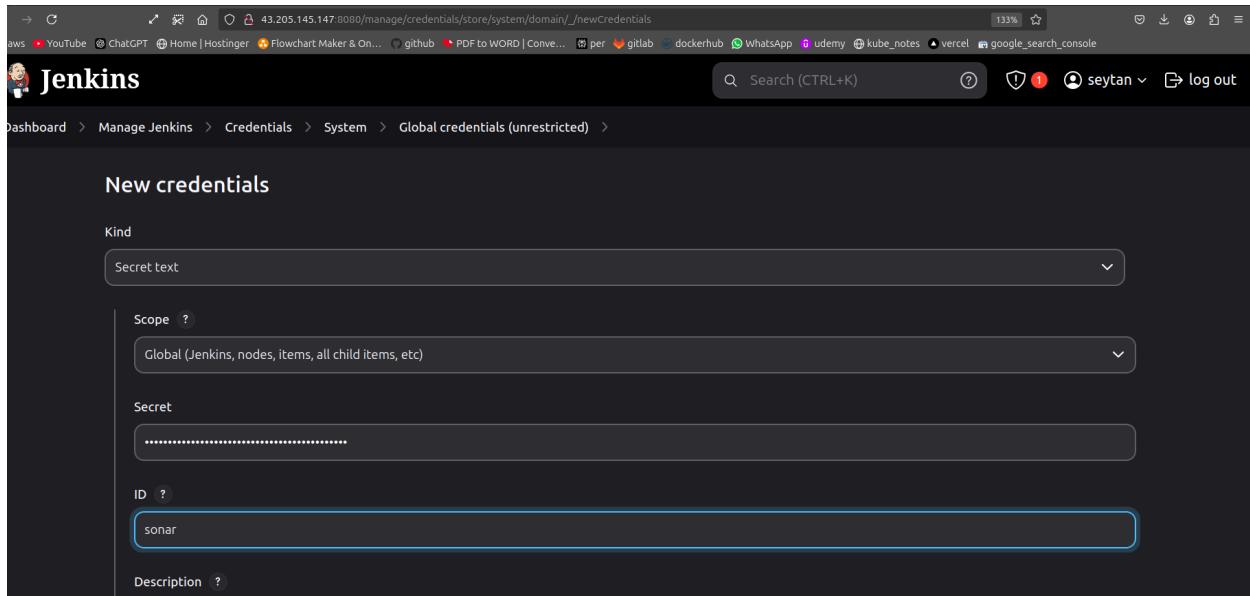
Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000, so <Public IP>:9000. Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token

The screenshot shows the Sonarqube Administration interface. The 'Administration' tab is selected. In the 'Configuration' dropdown menu, the 'Users' option is highlighted with a red box.

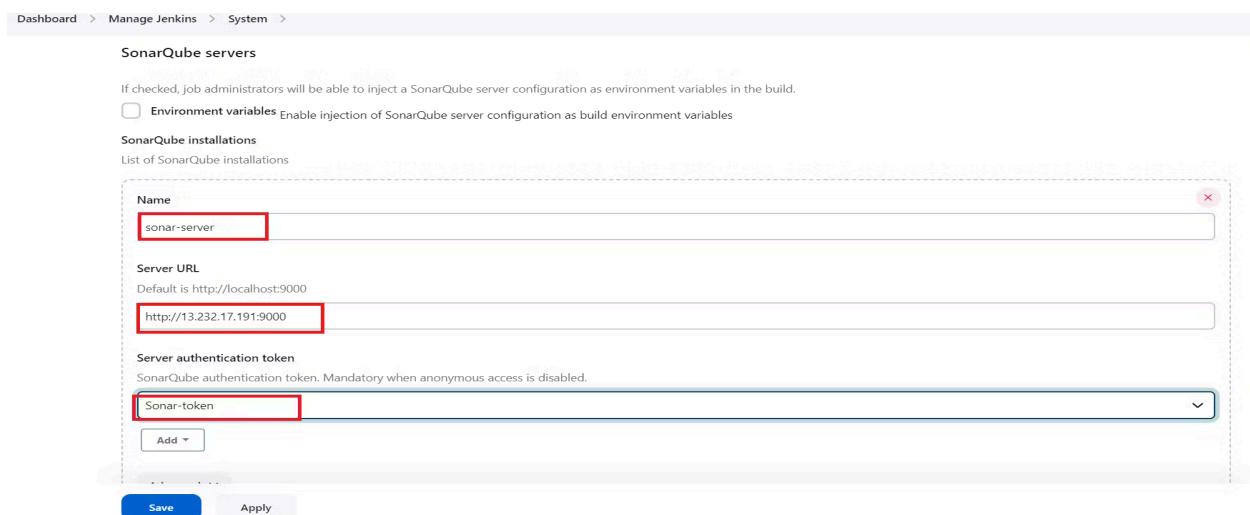
The screenshot shows the 'Tokens' configuration page. A message at the top states: 'If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.' Below this, there is a 'Generate Tokens' section with fields for 'Name', 'Type', and 'Expires in' (set to 30 days). A success message at the bottom says: 'New token "sonar" has been created. Make sure you copy it now, you won't be able to see it again!' with a 'Copy' button.

copy Token

Goto Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this



Now, go to Dashboard → Manage Jenkins → System and Add like the below image.



Step 10: Create a Pipeline upto Docker.

The screenshot shows the Jenkins interface for creating a new item. The title bar says 'Jenkins'. Below it, the URL is 43.205.145.147:8080/view/all/newJob. The main area is titled 'New Item'. A search bar says 'Search (CTRL+K)'. Below it, 'Dashboard > All > New Item' is shown. The 'New Item' form has 'Enter an item name' set to 'Netflix_clone'. Under 'Select an item type', 'Freestyle project' is listed with a description: 'Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.' The 'Pipeline' option is highlighted with a dark background, showing its icon (a blue square with a white circle) and description: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' Below it, 'Multi-configuration project' is listed with a description: 'Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.'

For Docker we will need docker credential so create them .

The screenshot shows the Jenkins 'Manage Jenkins' section under 'Credentials'. The URL is 43.205.145.147:8080/manage/credentials/store/system/domain/_/newCredentials. The path is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. The form is for creating a 'Kind' of 'Username with password'. It includes fields for 'Scope' (set to 'Global'), 'Username' (set to 'gaurav1251'), 'Treat username as secret' (checked), 'Password' (a redacted field), 'ID' (set to 'docker'), and 'Description' (empty). The 'ID' field is highlighted with a blue border.

Create a TMDB API Key

Next, we will create a TMDB API key

Open a new tab in the Browser and search for TMDB

The screenshot shows a Google search results page for "TMDB". The top result is "The Movie Database" (tmdb.org), which is described as a "popular, user editable database for movies and TV shows". Below it is a result for "Laravel TMDB", which is a clone of the official TMDB API. The search bar shows "All" selected, and there are links for Images, News, Videos, Shopping, and More. The results page includes a search bar and several links: API Reference, API key, Login to your account, Popular Movies, and Now Playing.

Click on the first result, you will see this page

The screenshot shows the TMDB homepage. At the top right, there is a user profile icon with a red box around it. A dropdown menu is open, showing options: Aj7Ay (View profile), Discussions, Lists, Ratings, Watchlist, Edit Profile, Settings (which is highlighted with a red box), and Logout. The main content area features a banner with the text "Welcome. Millions of movies, TV shows and people to discover. Explore now." and a search bar. Below the banner is a "Trending" section with cards for various movies like "Locke", "The Equalizer", "LUPIN", "Totally Killer", "Haunted Mansion", "The Nun", "Pet Sematary: Bloodlines", and "Ball".

Now click on API from the left side panel.

The screenshot shows the "Account Settings" page in the TMDB app. The user profile "Aj7Ay" is at the top left. On the left, there is a sidebar with a "Settings" tab and a list of options: Edit Profile, Account Settings (which is highlighted with a red box), Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, Sessions, API (which is highlighted with a red box), and Delete Account. The main content area shows "Account Settings" with fields for Default Language (English (en-US)), Fallback Language (None (Don't Fallback)), Country (India), Timezone (Auto detect? Asia - Kabul), and other settings like Include Adult Items in Search? (No) and Filter Profanity? (Yes).

Now click on create

The screenshot shows the TMDb API settings interface. On the left is a sidebar with a purple header labeled 'Settings'. Below it are several options: Edit Profile, Account Settings, Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, Sessions, and API. The 'API' option is currently selected. At the top right, there are three tabs: 'API', 'Overview', and 'Create', with 'Create' being the active tab and highlighted with a red box. The main content area contains sections for 'Documentation' (link to developer.themoviedb.org) and 'Support' (link to support forums). A 'Request an API Key' section includes a link to generate a new key.

Click on Developer

This screenshot shows the 'Request an API Key' page. The left sidebar has the same 'Settings' and 'API' tabs as the previous page. The main content asks 'What type of API key do you wish to register?' and lists two options: 'Developer' and 'Professional'. The 'Developer' option is highlighted with a red box. It includes a list of requirements: You are an individual, Your project is still in development, Your project is non profit, and Your project is ad supported. The 'Professional' option is also listed with its own requirements: You represent a company, Your project is for profit (not ad supported), and You are an OEM or hardware vendor.

Provide basic details Click on submit and you will get your API key.

Use this api key in the Pipeline script at appropriate position.

Once It is done write a Pipeline Script and build the Script.

The screenshot shows the Jenkins Pipeline configuration interface. The pipeline script is defined as follows:

```

1 pipeline {
2     agent any
3     tools {
4         jdk 'jdk'
5         nodejs 'nodejs'
6     }
7     environment {
8         SCANNER_HOME = tool 'sonar'
9     }
10    stages {
11        stage('clean workspace') {
12            steps {
13                cleanWs()
14            }
15        }
16        stage('Checkout from Git') {
17            steps {
18                git branch: 'main', changelog: false, poll: false, url: 'https://github.com/Gaurav1251/Netflix-ciCD'
}

```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Save](#) [Apply](#)

Go to the instance before building the Pipeline and install the Trivy.

```

ubuntu@ip-172-31-33-246: ~
ubuntu@ip-172-31-33-246: $ sudo apt-get install wget apt-transport-https gnupg lsb-release
ubuntu@ip-172-31-33-246: $ wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
ubuntu@ip-172-31-33-246: $ echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
ubuntu@ip-172-31-33-246: $ sudo apt-get update
ubuntu@ip-172-31-33-246: $ sudo apt-get install trivy
ubuntu@ip-172-31-33-246: $ Reading package lists... Done
ubuntu@ip-172-31-33-246: $ Building dependency tree... Done
ubuntu@ip-172-31-33-246: $ Reading state information... Done
ubuntu@ip-172-31-33-246: $ wget is already the newest version (1.21.4-1ubuntu4.1).
ubuntu@ip-172-31-33-246: $ wget set to manually installed.
ubuntu@ip-172-31-33-246: $ gnupg is already the newest version (2.4.4-2ubuntu17).

```

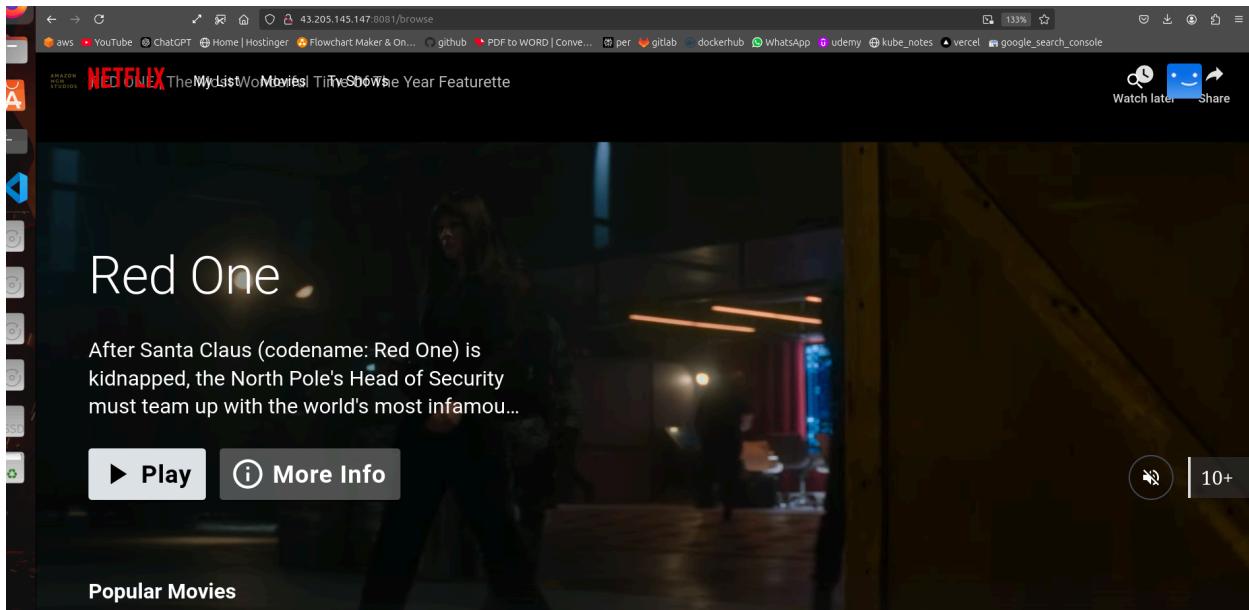
Once it is done Build the Script .

The Jenkins Pipeline stage view shows the following execution times:

Stage	Declarative: Tool Install	clean workspace	Checkout from Git	SonarQube Analysis	quality gate	Install Dependencies	TRIVY FS SCAN	Docker Build & Push
Average stage times: (Average full run time: ~2min 28s)	9s	254ms	1s	21s	335ms	18s	3s	1min 2s
Nov 23 17:36 No Changes	160ms	245ms	1s	20s	278ms	13s	6s	2min 4s

Copy the ec2 ip and paste it along with the port .

Ec2-ip:8081



Step 11: Add Kubernets Code in Pipeline Build the Project Again.

First run the Following command on your Jenkins instance .

- **aws eks update-kubeconfig --name CLUSTER NAME --region CLUSTER REGION**

```
root@ip-172-31-34-52:/home/ubuntu# aws eks update-kubeconfig --name EKS_CLOUD --region ap-south-1
Added new context arn:aws:eks:ap-south-1:678204272547:cluster/EKS_CLOUD to /root/.kube/config
root@ip-172-31-34-52:/home/ubuntu# kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-172-31-38-23.ap-south-1.compute.internal   Ready    <none>   23m   v1.31.2-eks-94953ac
root@ip-172-31-34-52:/home/ubuntu#
```

Now Give this command in CLI

```
cat /root/.kube/config
```

```
-->
root@ip-172-31-34-52:/home/ubuntu# cat /root/.kube/config
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0URS0tLS0tCk1JSURGVENDOWy20F3QUJBZ0lJUBc0dg81UVpzTDB3RFFZSk+vwKlc0dmNOVFVTEJRQX4GVETUJER0ExVUUKQXhN2ZvzmlaWDp1WlhShbN6QWVGdz
  server: https://172.31.34.52:6443
  name: k8s
  contexts:
- context:
  cluster: k8s
  user: k8s
  name: k8s
  current-context: k8s
contexts:
- context:
  cluster: k8s
  user: k8s
  name: k8s
  current-context: k8s
users:
- name: k8s
  password: 
  token: 
```

copy it and save it in documents or another folder save it as secret-file.txt

Install Kubernetes Plugin, Once it's installed successfully

Install	Name	Released
<input checked="" type="checkbox"/>	Kubernetes Client API 6.10.0-240.v57880ce8b_0b_2	10 mo ago
<input checked="" type="checkbox"/>	Kubernetes Credentials 190.v03c305394deb_	2 mo 6 days ago
<input checked="" type="checkbox"/>	Kubernetes 4296.v20a_7e4d77cf6	18 days ago
<input checked="" type="checkbox"/>	Kubernetes CLI 1.12.1	1 yr 2 mo ago

goto manage Jenkins → manage credentials → Click on Jenkins global → add credentials.

Kind
Secret file

Scope
Global (Jenkins, nodes, items, all child items, etc)

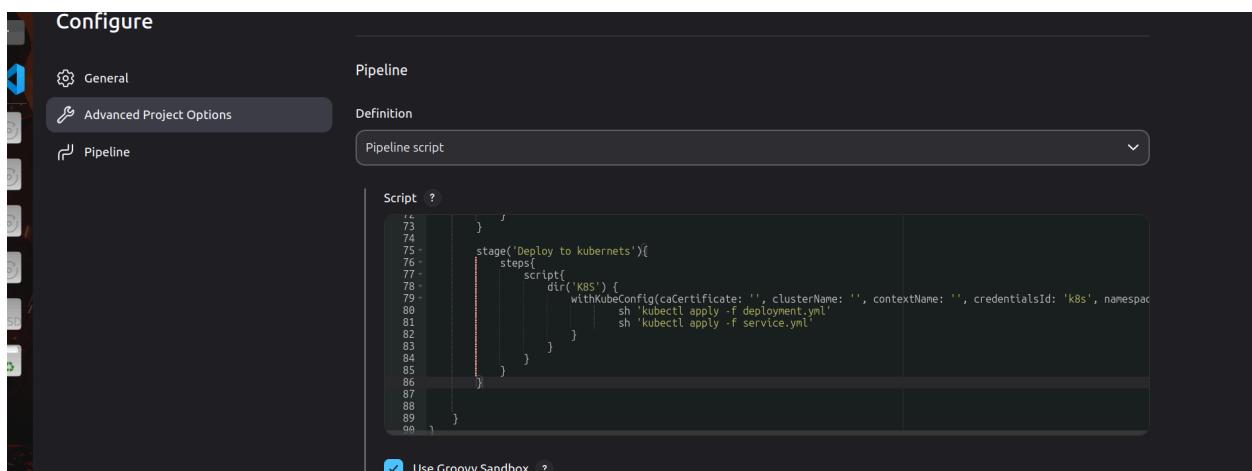
File
Browse... secret-file.txt

ID
k8s

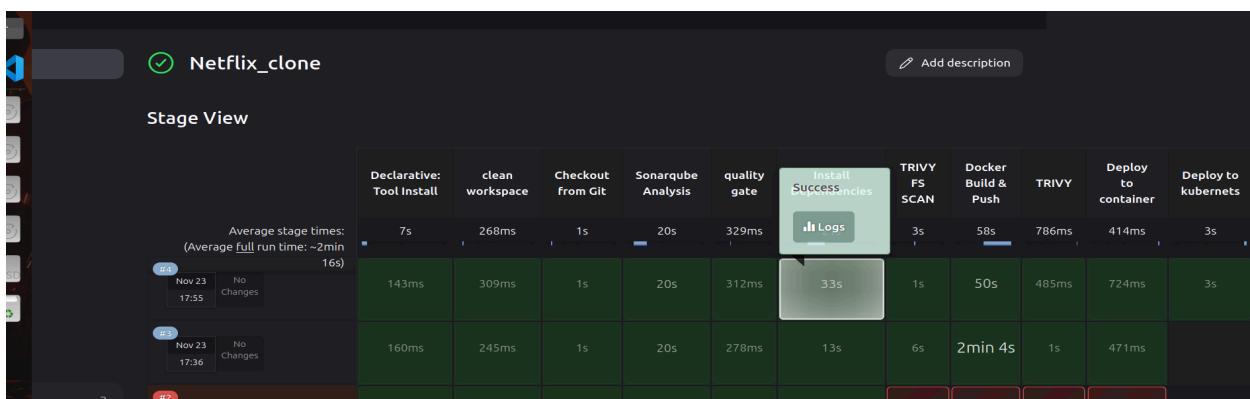
Description

Step 12: final step to deploy on the Kubernetes cluster

```
stage('Deploy to kubernets'){
    steps{
        script{
            dir('K8S') {
                withKubeConfig(caCertificate: "", clusterName: "", contextName: "",
                credentialsId: 'k8s', namespace: "", restrictKubeConfigAccess: false, serverUrl: "")
            {
                sh 'kubectl apply -f deployment.yml'
                sh 'kubectl apply -f service.yml'
            }
        }
    }
}
```

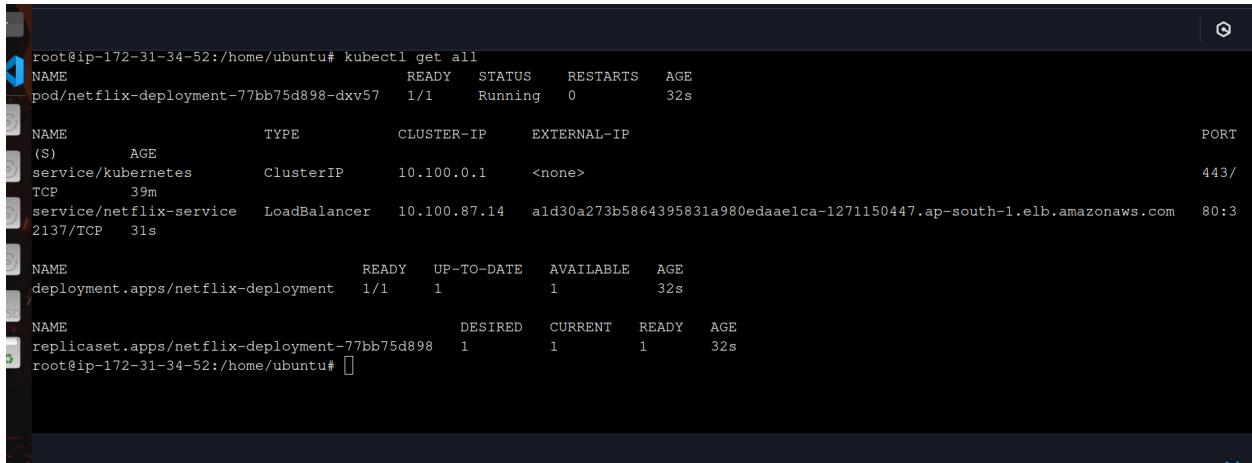


Build the Pipeline.



Give the command after pipeline success.

kubectl get all

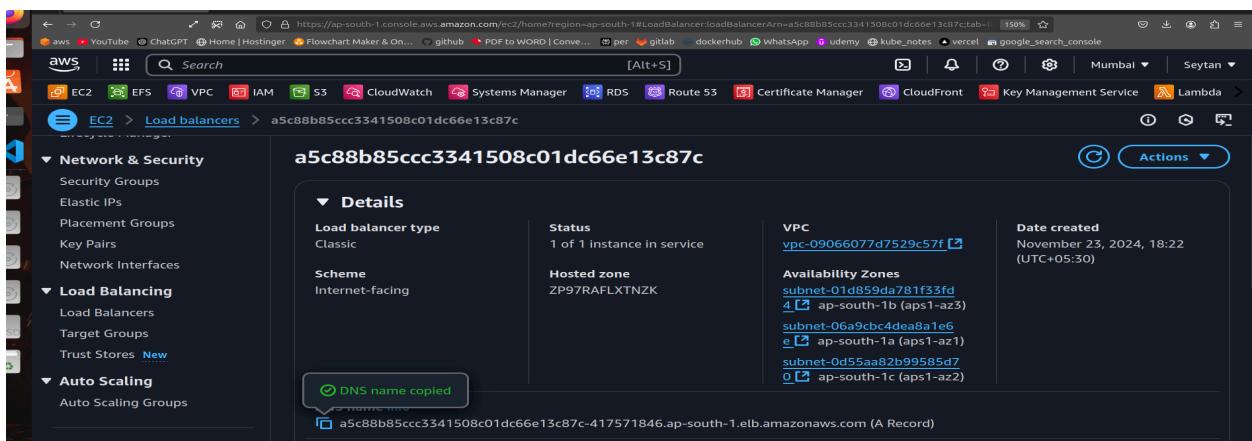


```
root@ip-172-31-34-52:/home/ubuntu# kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/netflix-deployment-77bb75d898-dxv57   1/1     Running   0          32s

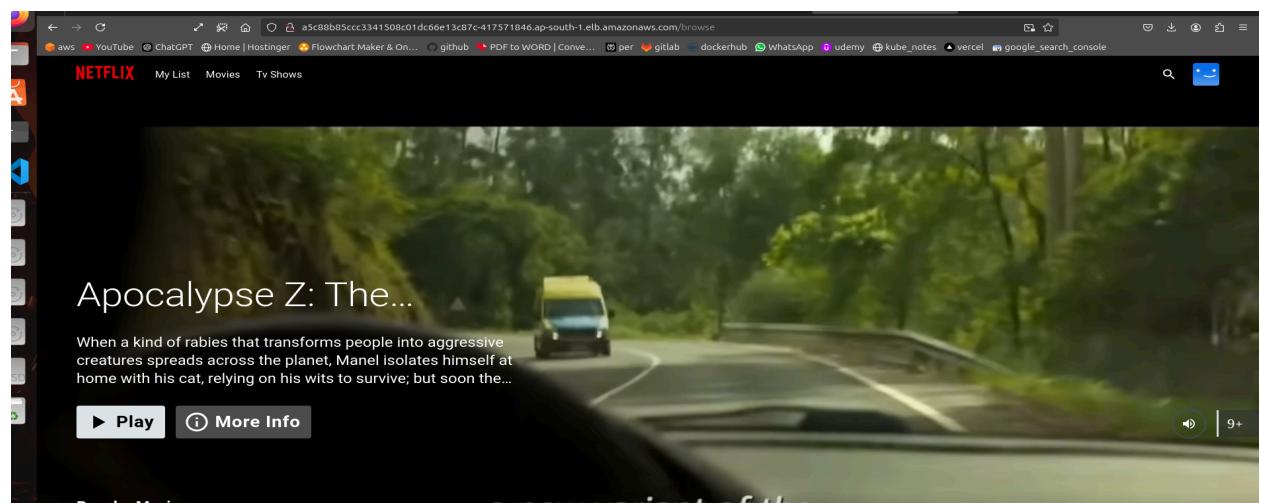
NAME                TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)
(S)    AGE
service/kubernetes   ClusterIP  10.100.0.1   <none>        443/
TCP    39m
service/netflix-service   LoadBalancer  10.100.87.14  a1d30a273b5864395831a980edaaelca-1271150447.ap-south-1.elb.amazonaws.com  80:3
2137/TCP  31s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/netflix-deployment   1/1     1           1           32s
NAME           DESIRED  CURRENT   READY   AGE
replicaset.apps/netflix-deployment-77bb75d898  1        1         1       32s
root@ip-172-31-34-52:/home/ubuntu#
```

Go to Ec2 Load-Balancer and check the dns and paste it in web browser.

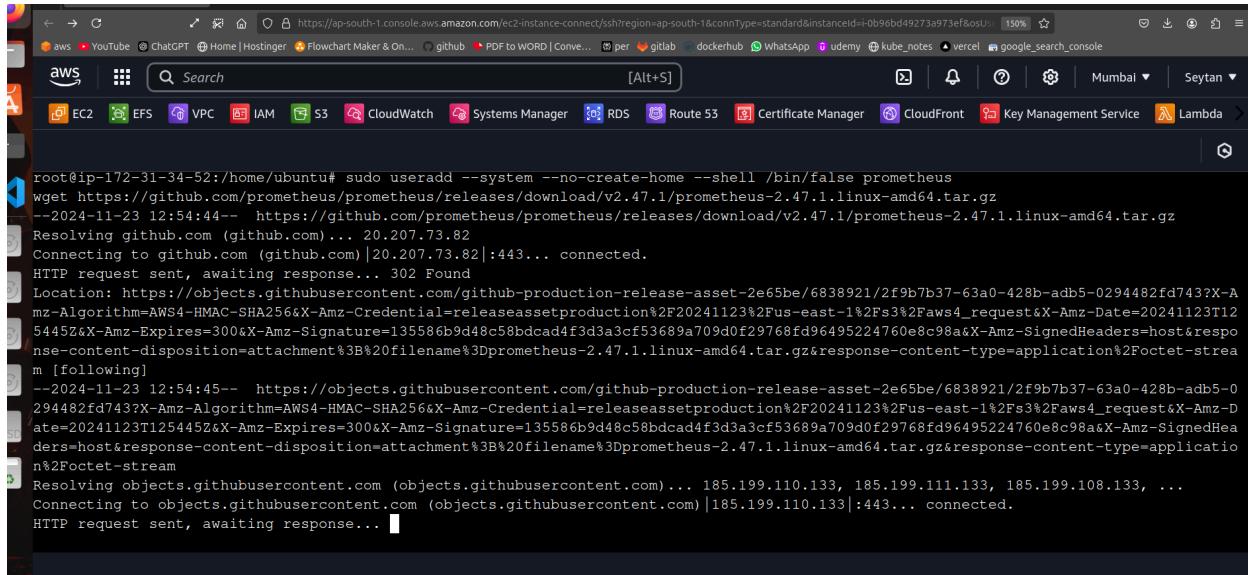


The screenshot shows the AWS CloudWatch interface with the URL <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancer:loadBalancerArn=a5c88b85ccc3341508c01dc66e13c87c>. The page displays details of a Load Balancer named 'a5c88b85ccc3341508c01dc66e13c87c'. The 'Details' section shows the Load balancer type as 'Classic', Status as '1 of 1 instance in service', Scheme as 'Internet-facing', Hosted zone as 'ZP97RAFLXTNZK', VPC as 'vpc-09066077d7529c57f', and Date created as 'November 23, 2024, 18:22 (UTC+05:30)'. The Availability Zones listed are: subnet-01d859da781f33fd (4), subnet-06a9cbc4dea8a1e6 (1), subnet-0d55aa82b99585d7 (0). A green message bubble indicates 'DNS name copied' with the value 'a5c88b85ccc3341508c01dc66e13c87c-417571846.ap-south-1.elb.amazonaws.com (A Record)'.

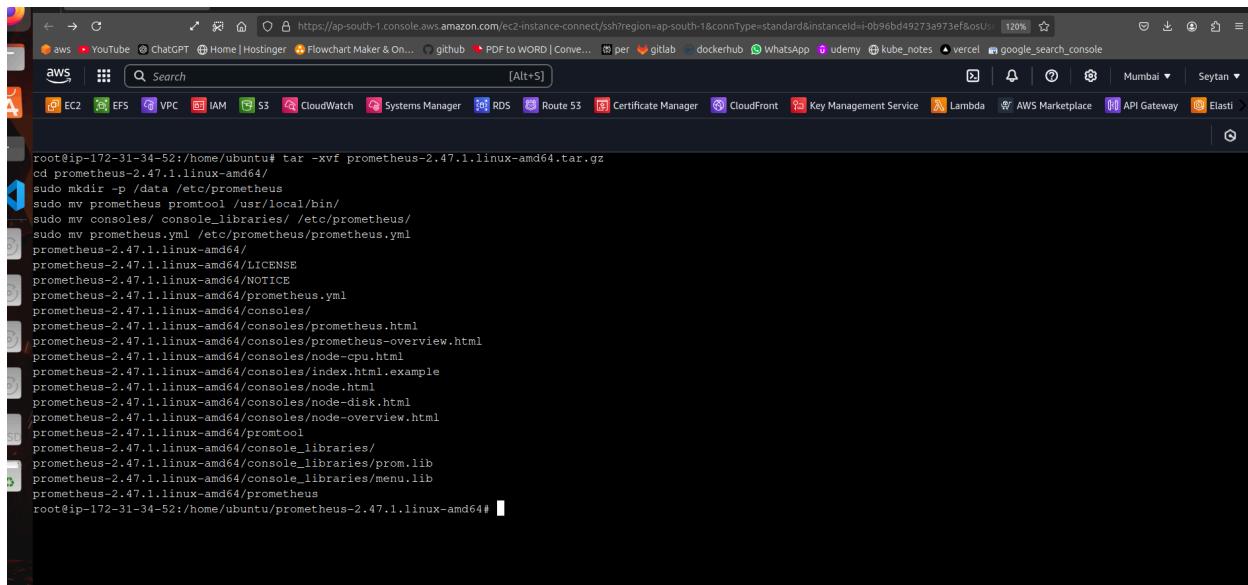


Step 13: Monitoring

Installing Prometheus:



```
root@ip-172-31-34-52:/home/ubuntu# sudo useradd --system --no-create-home --shell /bin/false prometheus
wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz
--2024-11-23 12:54:44-- https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com) |20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/2f9b7b37-63a0-428b-adb5-0294482fd743?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241123%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241123T125445Z&X-Amz-Expires=300&X-Amz-Signature=135586b9d48c58bdcad4f3d3a3cf53689a709d0f29768fd96495224760e8c98a&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.47.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-11-23 12:54:45-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/2f9b7b37-63a0-428b-adb5-0294482fd743?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241123%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241123T125445Z&X-Amz-Expires=300&X-Amz-Signature=135586b9d48c58bdcad4f3d3a3cf53689a709d0f29768fd96495224760e8c98a&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.47.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com) |185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 
```

```
root@ip-172-31-34-52:/home/ubuntu# tar -xvf prometheus-2.47.1.linux-amd64.tar.gz
cd prometheus-2.47.1.linux-amd64/
sudo mkdir -p /data/etc/prometheus
sudo mv prometheus promtool /usr/local/bin/
sudo mv consoles/ console_libraries/ /etc/prometheus/
sudo mv prometheus.yml /etc/prometheus/prometheus.yml
prometheus-2.47.1.linux-amd64/
prometheus-2.47.1.linux-amd64/LICENSE
prometheus-2.47.1.linux-amd64/NOTICE
prometheus-2.47.1.linux-amd64/prometheus.yml
prometheus-2.47.1.linux-amd64/consoles/
prometheus-2.47.1.linux-amd64/consoles/prometheus.html
prometheus-2.47.1.linux-amd64/consoles/prometheus-overview.html
prometheus-2.47.1.linux-amd64/consoles/node-cpu.html
prometheus-2.47.1.linux-amd64/consoles/index.html.example
prometheus-2.47.1.linux-amd64/consoles/node.html
prometheus-2.47.1.linux-amd64/consoles/node-disk.html
prometheus-2.47.1.linux-amd64/consoles/node-overview.html
prometheus-2.47.1.linux-amd64/promtool
prometheus-2.47.1.linux-amd64/console_libraries/
prometheus-2.47.1.linux-amd64/console_libraries/prom.lib
prometheus-2.47.1.linux-amd64/console_libraries/menu.lib
prometheus-2.47.1.linux-amd64/prometheus
root@ip-172-31-34-52:/home/ubuntu/prometheus-2.47.1.linux-amd64/ 
```

Check Prometheus is Running or not.

sudo systemctl status prometheus

```

root@ip-172-31-34-52:/home/ubuntu/prometheus-2.47.1.linux-amd64# sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
root@ip-172-31-34-52:/home/ubuntu/prometheus-2.47.1.linux-amd64# sudo nano /etc/systemd/system/prometheus.service
root@ip-172-31-34-52:/home/ubuntu/prometheus-2.47.1.linux-amd64# sudo systemctl enable prometheus
sudo systemctl start prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
root@ip-172-31-34-52:/home/ubuntu/prometheus-2.47.1.linux-amd64# sudo systemctl status prometheus
● prometheus.service - Prometheus
    Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
      Active: active (running) since Sat 2024-11-23 12:56:12 UTC; 8s ago
        PID: 15745 (prometheus)
       Tasks: 13 (limit: 38494)
      Memory: 18.4M (peak: 19.3M)
         CPU: 94ms
      CGroup: /system.slice/prometheus.service
              └─15745 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/data --web.console.templates=/etc/prometheus/consoles -s

Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.876Z caller=head.go:681 level=info component=tsdb msg="On-disk memory mappable chunks replayed"
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.876Z caller=head.go:689 level=info component=tsdb msg="Replaying WAL, this may take a while"
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.877Z caller=head.go:760 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=1
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.877Z caller=head.go:797 level=info component=tsdb msg="WAL replay completed" checkpoint_replay=true
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.879Z caller=main.go:1045 level=info fs_type="EXT4_SUPER_MAGIC"
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.879Z caller=main.go:1048 level=info msg="TSDB started"
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.879Z caller=main.go:1229 level=info msg="Loading configuration file" filename=/etc/prometheus/config.yaml
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.883Z caller=main.go:1266 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/config.yaml
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.883Z caller=main.go:1009 level=info msg="Server is ready to receive web requests."
Nov 23 12:56:12 ip-172-31-34-52 prometheus[15745]: ts=2024-11-23T12:56:12.883Z caller=manager.go:1009 level=info component="rule manager" msg="Starting rule manager..."

```

Lines 1-20/20 (END)

Installing Node Exporter:

```

root@ip-172-31-34-52:/home/ubuntu# sudo useradd --system --no-create-home --shell /bin/false node_exporter
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
--2024-11-23 12:57:09-- https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 304 Found
Location: https://objeccts.githubusercontent.com/github-production-release-asset-2e65be/9524057/5509b569-5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241123%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241123T125709Z&X-Amz-Expires=300X-Amz-Signature=8c609fb531b365fa6bc2c98bc6b79d315ffa62339f479f3c4935fc0b7508f4-X-Amz-SignedHeaders=host&response-content-disposition=attachment%3Bfilename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-11-23 12:57:09-- https://objeccts.githubusercontent.com/github-production-release-asset-2e65be/9524057/5509b569-5c34-471e-8598-c05c0733bb7f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241123%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241123T125709Z&X-Amz-Expires=300X-Amz-Signature=8c609fb531b365fa6bc2c98bc6b79d315ffa62339f479f3c4935fc0b7508f4-X-Amz-SignedHeaders=host&response-content-disposition=attachment%3Bfilename%3Dnode_exporter-1.6.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objeccts.githubusercontent.com (objeccts.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.109.133, ...
Connecting to objeccts.githubusercontent.com (objeccts.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response...

```

```

root@ip-172-31-34-52:/home/ubuntu# tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
sudo mv node_exporter-1.6.1.linux-amd64/node_exporter /usr/local/bin/
rm -rf node_exporter*
node_exporter-1.6.1.linux-amd64/
node_exporter-1.6.1.linux-amd64/NOTICE
node_exporter-1.6.1.linux-amd64/node_exporter
node_exporter-1.6.1.linux-amd64/LICENSE
root@ip-172-31-34-52:/home/ubuntu#

```

Check it running status.

sudo systemctl status node_exporter

```
root@ip-172-31-34-52:/home/ubuntu# sudo nano /etc/systemd/system/node_exporter.service
root@ip-172-31-34-52:/home/ubuntu# sudo systemctl enable node_exporter
sudo systemctl start node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
root@ip-172-31-34-52:/home/ubuntu# sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
    Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
      Active: active (running) since Sat 2024-11-23 12:58:11 UTC; 8s ago
        Main PID: 15865 (node_exporter)
          Tasks: 5 (limit: 38494)
        Memory: 2.9M (peak: 3.1M)
          CPU: 8ms
        CGrou... /system.slice/node_exporter.service
              └─15865 /usr/local/bin/node_exporter --collector.logind

Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=thermal_zone
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=time
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=timex
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=udp_queues
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=uname
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=vmsstat
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=xfs
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=node_exporter.go:117 level=info collector=zfs
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=tls_config.go:274 level=info msg="Listening on" address=[::]:9100
Nov 23 12:58:11 ip-172-31-34-52 node_exporter[15865]: ts=2024-11-23T12:58:11.915Z caller=tls_config.go:277 level=info msg="TLS is disabled." http2=false address=[::]:9100
Lines 1-20/20 (END)
```

Configure Prometheus Plugin Integration:

The screenshot shows the Jenkins interface for managing plugins. The user has searched for 'prome'. The results page lists three available plugins:

- Prometheus metrics (795.v99576210f28): Jenkins Prometheus Plugin expose an endpoint (default /prometheus) with metrics where a Prometheus Server can scrape. Released 22 days ago.
- Otel agent host metrics monitoring (1.3.0): This plugin allows monitoring of Jenkins agents by deploying Prometheus node exporters and Otel collectors to them and linking to a Grafana dashboard displaying those gathered metrics. Released 21 days ago.
- Cortex Metrics (1.0.1): Adds the ability to publish run results to Cortex directly using the Prometheus push endpoint. Released 3 yr 8 mo ago.

Prometheus Configuration:

To configure Prometheus to scrape metrics from Node Exporter and Jenkins, you need to modify the `prometheus.yml`

```
root@ip-172-31-34-52:/home/ubuntu# sudo vim /etc/prometheus/prometheus.yml
root@ip-172-31-34-52:/home/ubuntu# promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax

root@ip-172-31-34-52:/home/ubuntu# curl -X POST http://localhost:9090/-/reload
root@ip-172-31-34-52:/home/ubuntu#
```

```
- job_name: 'node_exporter'
  static_configs:
    - targets: ['localhost:9100']

- job_name: 'jenkins'
  metrics_path: '/prometheus'
  static_configs:
    - targets: ['<your-jenkins-ip>:<your-jenkins-port>']
```

Install Grafana on Ubuntu and Set it up to Work with Prometheus :

The screenshot shows a terminal window with a dark background and light-colored text. It contains four numbered steps for installing Grafana:

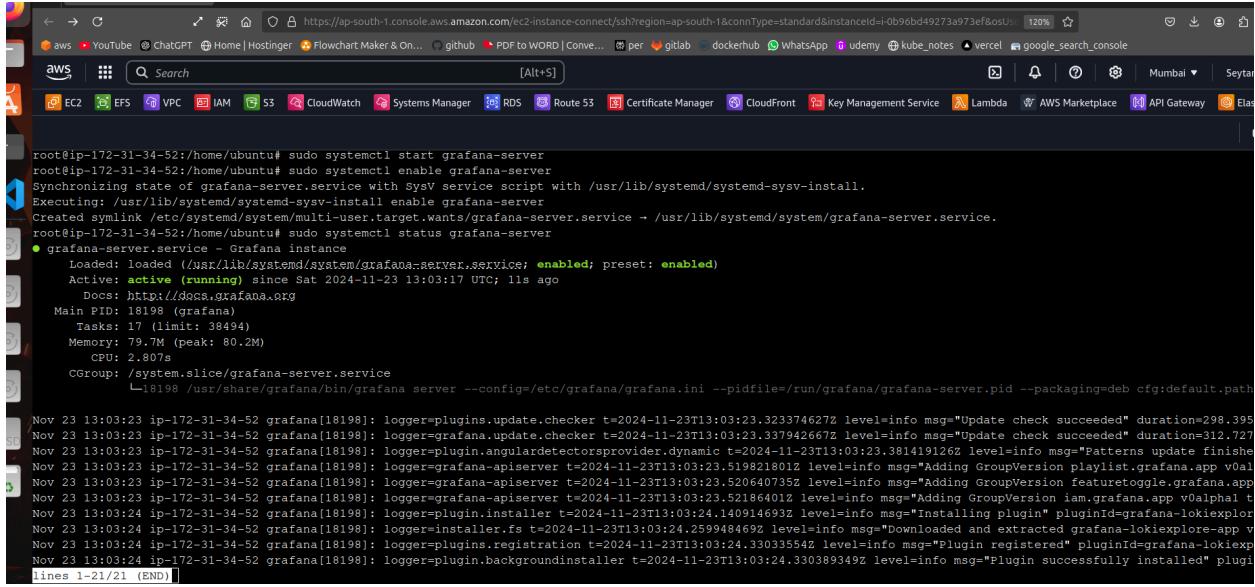
- Step 1:** First, ensure that all necessary dependencies are installed:

```
sudo apt-get update
sudo apt-get install -y apt-transport-https software-properties-common
```
- Step 2: Add the GPG Key:** Add the GPG key for Grafana:

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```
- Step 3: Add Grafana Repository:** Add the repository for Grafana stable releases:

```
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list
```
- Step 4: Update and Install Grafana:** Update the package list and install Grafana:

```
sudo apt-get update
sudo apt-get -y install grafana
```



```

root@ip-172-31-34-52:/home/ubuntu# sudo systemctl start grafana-server
root@ip-172-31-34-52:/home/ubuntu# sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /usr/lib/systemd/system/grafana-server.service.
root@ip-172-31-34-52:/home/ubuntu# sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
    Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
    Active: active (running) since Sat 2024-11-23 13:03:17 UTC; 11s ago
      Docs: http://docs.grafana.org
   Main PID: 18198 (grafana)
     Tasks: 17 (limit: 38494)
    Memory: 79.7M (peak: 80.2M)
       CPU: 2.807s
      CGroup: /system.slice/grafana-server.service
              └─18198 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg=default.path

Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-plugins.update.checker t=2024-11-23T13:03:23.323374627Z level=info msg="Update check succeeded" duration=298.395
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-grafana.update.checker t=2024-11-23T13:03:23.337942667Z level=info msg="Update check succeeded" duration=312.727
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-plugin.angulardetectorprovider.dynamic t=2024-11-23T13:03:23.381419126Z level=info msg="Patterns update finished"
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-grafana-apiserver t=2024-11-23T13:03:23.519821801Z level=info msg="Adding GroupVersion playlist.grafana.app v0alpha1"
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-grafana-apiserver t=2024-11-23T13:03:23.520640735Z level=info msg="Adding GroupVersion featuretoggle.grafana.app v0alpha1"
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-grafana-apiserver t=2024-11-23T13:03:23.52186401Z level=info msg="Adding GroupVersion iam.grafana.app v0alpha1"
Nov 23 13:03:23 ip-172-31-34-52 grafana[18198]: logger-grafana-apiserver t=2024-11-23T13:03:23.52186401Z level=info msg="Adding GroupVersion iam.grafana.app v0alpha1"
Nov 23 13:03:24 ip-172-31-34-52 grafana[18198]: logger-plugin.installer.t=2024-11-23T13:03:24.140914693Z level=info msg="Installing plugin" pluginId=grafana-lokiexplorer
Nov 23 13:03:24 ip-172-31-34-52 grafana[18198]: logger-installer.fs.t=2024-11-23T13:03:24.259948469Z level=info msg="Downloaded and extracted grafana-lokiexplorer-app v0.1.0"
Nov 23 13:03:24 ip-172-31-34-52 grafana[18198]: logger-plugins.registration.t=2024-11-23T13:03:24.33033554Z level=info msg="Plugin registered" pluginId=grafana-lokiexplorer
Nov 23 13:03:24 ip-172-31-34-52 grafana[18198]: logger-plugin.backgroundinstaller.t=2024-11-23T13:03:24.330389349Z level=info msg="Plugin successfully installed" pluginId=grafana-lokiexplorer
lines 1-21/21 (END)

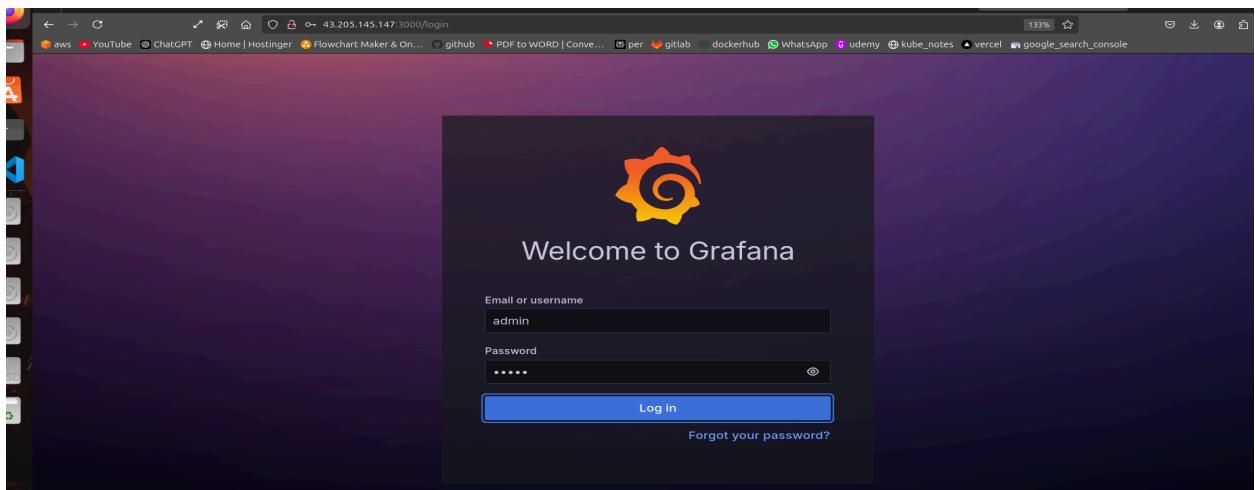
```

Step 14: Access Grafana Web Interface:

Open a web browser and navigate to Grafana using your server's IP address. The default port for Grafana is 3000. For example:

<http://<your-server-ip>:3000>

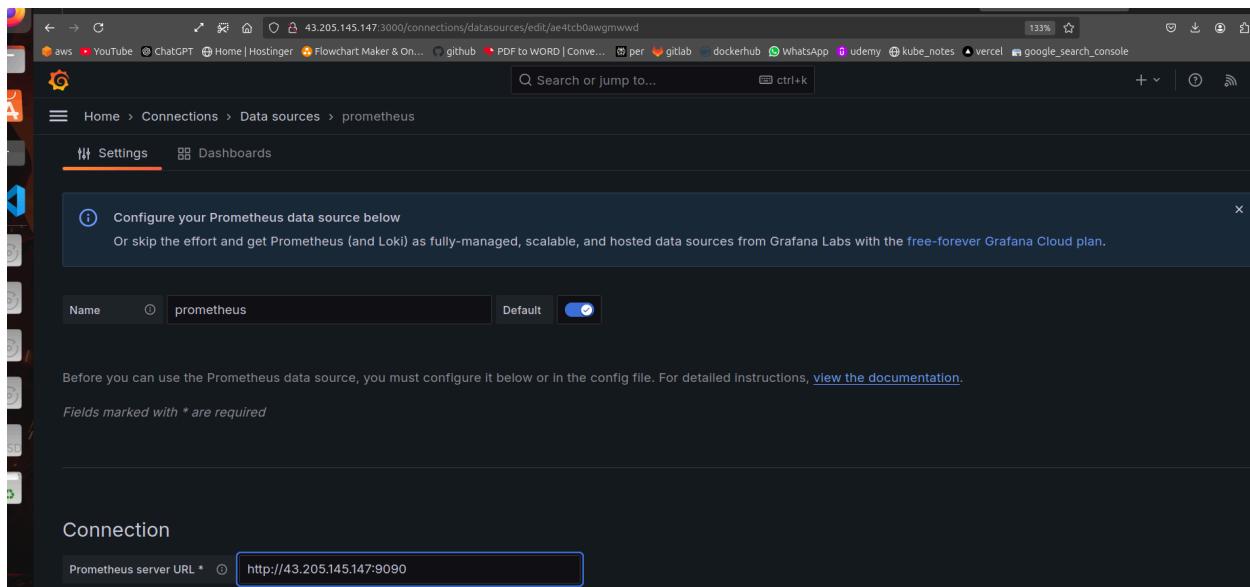
You'll be prompted to log in to Grafana. The default username is "admin," and the default password is also "admin."



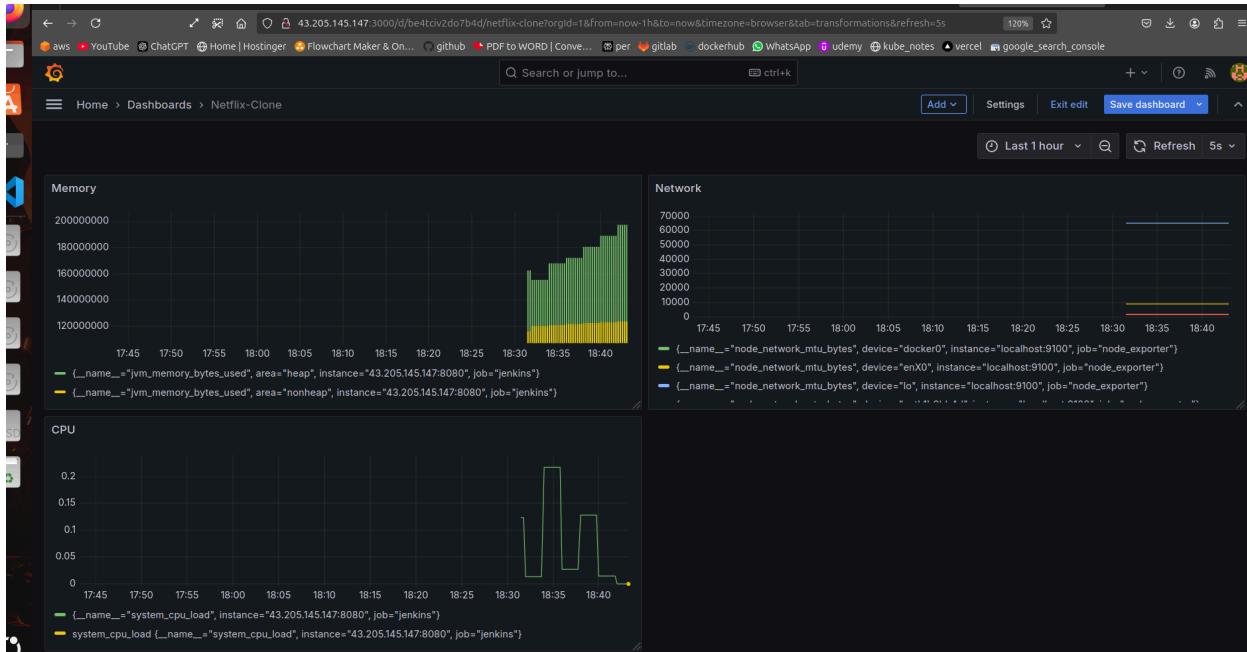
Add Prometheus Data Source:

To visualize metrics, you need to add a data source. Follow these steps:

- Click on the gear icon (in the left sidebar to open the "Configuration" menu.
- Select "Data Sources."
- Click on the "Add data source" button.
- Choose "Prometheus" as the data source type.
- In the "HTTP" section:
 - Set the "URL" to **http://localhost:9090** (assuming Prometheus is running on the same server).
 - Click the "Save & Test" button to ensure the data source is working.



Now Create a dashboard ,in that create multiple queries for visualizing the data and save the dashboard.



Step 15: Destruction

Now Go to Jenkins Dashboard and click on Terraform-Eks job

And build with parameters and destroy action

It will delete the EKS cluster that provisioned

The screenshot shows the Jenkins Pipeline configuration for the 'Terraform_Eks' job. The pipeline is named 'Pipeline Terraform_Eks'.

- Status:** Shows the pipeline status.
- Changes:** Shows recent changes.
- Build with Parameters:** A dropdown menu is open, showing the selected 'action' as 'destroy'.
- Configure:** A link to edit the pipeline configuration.
- Delete Pipeline:** A link to delete the pipeline.
- Full Stage View:** A link to view the full stage details.
- Stages:** A link to manage pipeline stages.
- Rename:** A link to rename the pipeline.
- Pipeline Syntax:** A link to view the pipeline syntax.

Builds:

Build #	Timestamp
#3	11:42 AM
#2	11:40 AM
#1	11:39 AM

