

PROJECT-6

DEPLOYMENT OF NETFLIX APPLICATION ON KUBERNETES (EKS)

-ANIL GOLLENA

Introduction

This document outlines the phased deployment of the Netflix application on Kubernetes using Amazon Elastic Kubernetes Service (EKS). The project follows a structured approach to ensure scalability, security, automation, and monitoring. Below is the detailed introduction for each phase of the deployment.

Phase 1: Deploy Netflix Application Locally on EC2 (t2.large)

- The first phase focuses on setting up the Netflix application in a controlled environment using AWS EC2 instances.
- **Objective:** Launch the Netflix application on an EC2 instance (t2.large) to validate its functionality in a local deployment environment.
- **Steps Involved:**
 - Provision an EC2 instance with sufficient resources (CPU, memory) to run the application.
 - Install necessary dependencies and runtime environments.
 - Deploy the application and verify its functionality locally.

Phase 2: Implementation of Security with SonarQube and Trivy

- Security is a critical aspect of this deployment.
- **Objective:** Ensure the code and deployment artifacts meet security standards.
- **Steps Involved:**
 - **SonarQube:** Analyze the application's source code for vulnerabilities and adherence to coding standards.
 - **Trivy:** Scan container images for known vulnerabilities before they are deployed.
 - Address identified issues to strengthen the overall security posture.

Phase 3: Automating the Deployment Using Jenkins Pipeline

- This phase introduces automation to improve deployment efficiency.
- **Objective:** Automate the build, test, and deployment processes of the Netflix application using Jenkins CI/CD pipelines.
- **Steps Involved:**
 - Set up Jenkins with a declarative pipeline.
 - Integrate SonarQube and Trivy scans into the pipeline for continuous security checks.
 - Automate deployment steps to ensure repeatability and minimize manual intervention.

Phase 4: Monitoring via Prometheus and Grafana

- Monitoring ensures the application's reliability and performance.
- **Objective:** Implement a robust monitoring solution using Prometheus and Grafana.
- **Steps Involved:**
 - Deploy Prometheus to collect metrics from the application and infrastructure.
 - Configure Grafana dashboards to visualize critical metrics and enable real-time monitoring.
 - Set up alerts for proactive issue detection and resolution.

Phase 5: Deploying on Kubernetes (EKS)

- The final phase involves deploying the application on a scalable Kubernetes cluster managed by EKS.
- **Objective:** Achieve high availability, scalability, and efficient resource management for the Netflix application.
- **Steps Involved:**
 - Create and configure an EKS cluster with worker nodes.
 - Deploy the Netflix application as Kubernetes deployments and services.
 - Use Kubernetes features like auto-scaling, load balancing, and rolling updates to enhance the deployment.
 - Integrate monitoring and alerting tools for end-to-end observability.

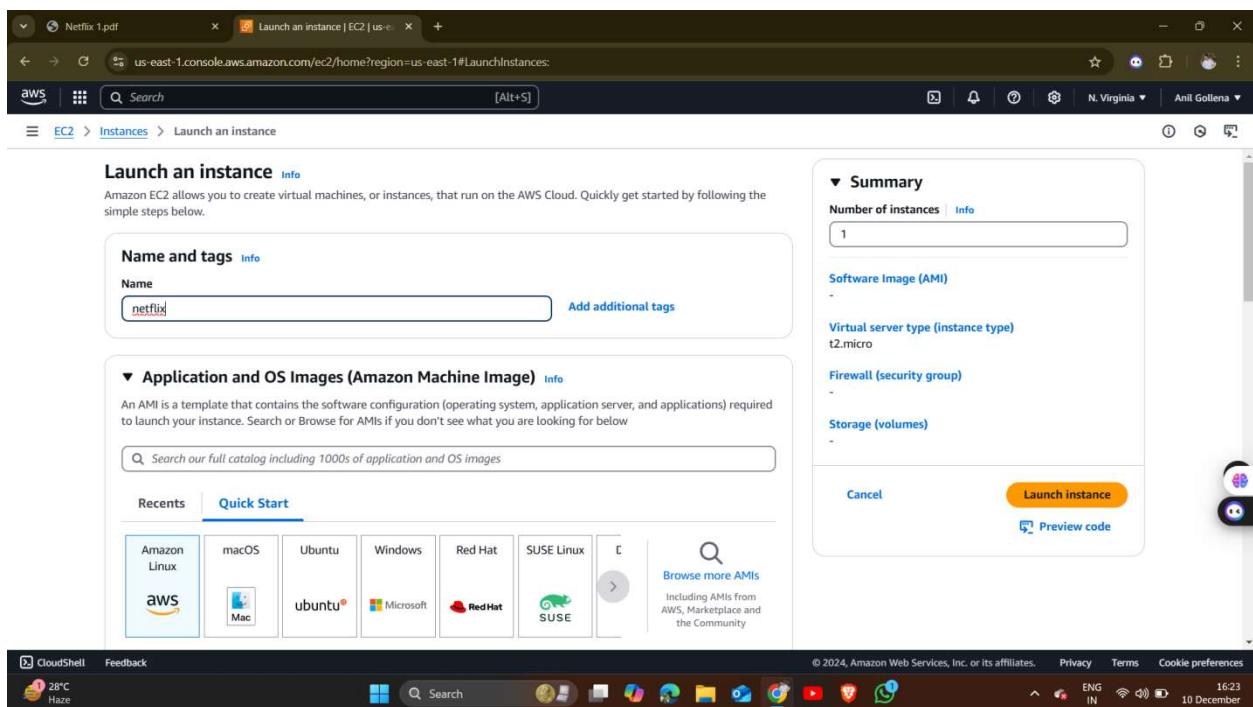
PHASE1: DEPLOYMENT OF NETFLIX APPLICATION ON EC2 (T2.LARGE).

Now go to aws management console and create a new instance with t2.large.

Choose ubuntu as operating system.

Increase the root volume 8GB→25GB.

Launch the ec2 instance.



Now connect to the server

Install git and clone the repository

Now create elastic ip and attach it with instance.

Associate Elastic IP address

Elastic IP address: 44.206.155.154

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance

Network interface

Instance

i-07351e323b0b939ba

Private IP address

172.31.97.24

Reassociation
Specify whether the Elastic IP address can be reassigned if it's already associated with a resource.

Allow this Elastic IP address to be reassigned

Cancel Associate

```
ubuntu@ip-172-31-97-24: ~ /Netflix
$ cd Downloads/
ANTILAnil-Gollena12 MINGW64 ~ (master)
$ ssh -i "anil.pem" ubuntu@ec2-44-206-155-154.compute-1.amazonaws.com
The authenticity of host 'ec2-44-206-155-154.compute-1.amazonaws.com (44.206.155.154)' can't be established.
ED25519 key fingerprint is SHA256:y12jb1s08.5mrfZptwV2zE0WrRRAg0o5h18tZMtB4.
This key is known by the user 'anil' from host 'ip-172-31-97-24'.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-206-155-154.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sun Dec 15 13:14:43 UTC 2024

System load: 0.0 Processes: 115
Usage of /: 2% of 24.05GB Users logged in: 1
Memory usage: 3% IPv4 address for eth0: 172.31.97.24
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

45 updates can be applied immediately.
35 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec 15 13:03:11 2024 from 49.204.100.179
ubuntu@ip-172-31-97-24:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://esm-2.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
37 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-97-24:~$ sudo git clone https://github.com/Gouserabbani44/Netflix.git
Cloning into 'Netflix'...
remote: Enumerating objects: 773, done.
remote: Counting objects: 100% (773/773), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 773 (delta 430), reused 768 (delta 436), pack-reused 0 (from 0)
Receiving objects: 100% (773/773), 13.47 MiB | 44.64 MiB/s, done.
Haze
```

Now install docker and setup docker and images.

Now run the following command to set up docker.

- sudo apt-get install docker.io
 - sudo chmod 666 /var/run/docker.sock
 - sudo usermod -aG docker ubuntu
 - sudo newgrp docker
 - exit
 - newgrp docker
 - sudo systemctl start docker sudo
 - systemctl enable docker
 - sudo systemctl status docker

```
ubuntu@ip-172-31-97-24: ~$Netfix
Processing triggers for dbus (1.12.20-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

ubuntu@ip-172-31-97-24:~$ sudo systemctl start docker
ubuntu@ip-172-31-97-24:~$ sudo systemctl enable docker
ubuntu@ip-172-31-97-24:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-12-15 13:15:49 UTC; 35s ago
       Docs: https://docs.docker.com
 Main PID: 2717 (dockerd)
   Tasks: 1
    Memory: 26.0M
      CPU: 252ms
     CGroup: /system.slice/docker.service
             └─2717 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 15 13:15:48 ip-172-31-97-24 systemd[1]: Starting Docker Application Container Engine...
Dec 15 13:15:48 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:48.750371450Z" level=info msg="Starting up"
Dec 15 13:15:48 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:48.751419810Z" level=info msg="detected 0.0.53 nameserver, assuming systemd-resolved, so using resolv.conf: /run/systemd/resolve/resolv.conf"
Dec 15 13:15:48 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:48.751420010Z" level=info msg="loading configuration from containerd config: done"
Dec 15 13:15:49 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:49.128837792Z" level=info msg="Loading containers: done"
Dec 15 13:15:49 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:49.159971912Z" level=info msg="Docker daemon" commit="24.0.7~Ubuntu22.04.1" graphdriver=overlay2 version=24.0.7
Dec 15 13:15:49 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:49.159942504Z" level=info msg="Daemon has completed initialization"
Dec 15 13:15:49 ip-172-31-97-24 dockerd[2717]: time="2024-12-15T13:15:49.209492984Z" level=info msg="API listen on /run/docker.sock"
Dec 15 13:15:49 ip-172-31-97-24 systemd[1]: Started Docker Application Container Engine.

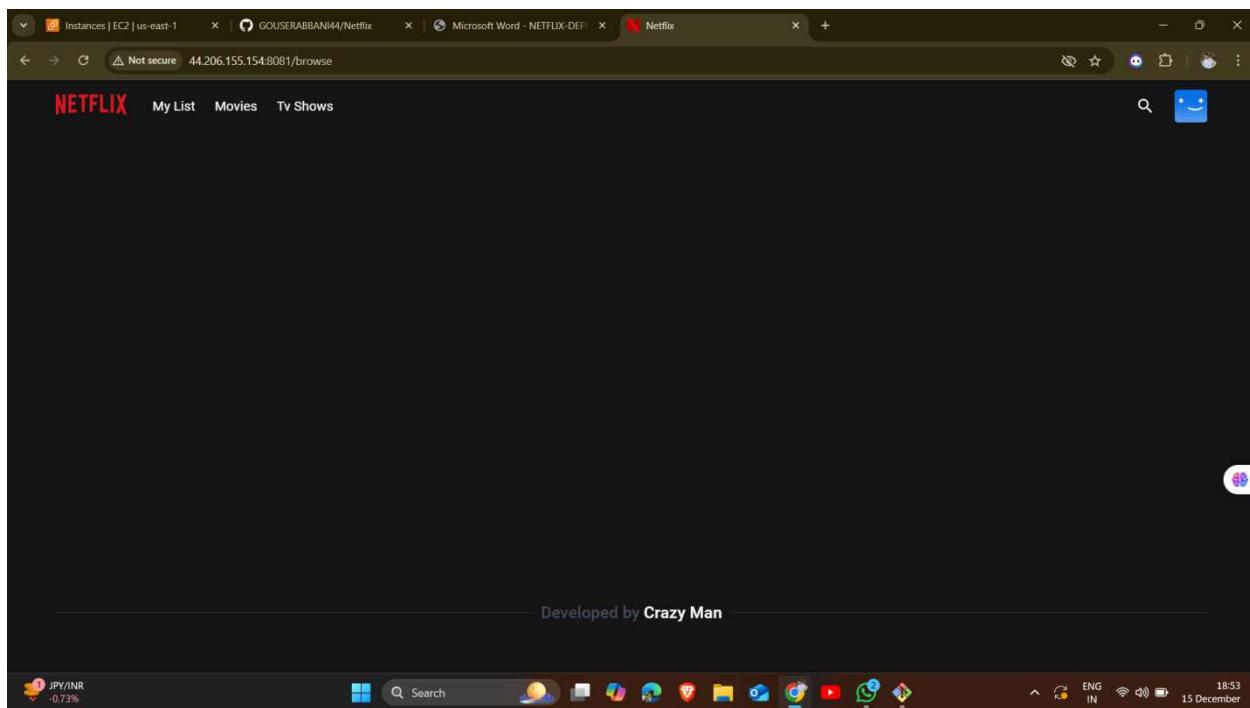
ubuntu@ip-172-31-97-24:~$ sudo chmod 666 /var/run/docker.sock
ubuntu@ip-172-31-97-24:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-97-24:~$ sudo newgrp docker
ubuntu@ip-172-31-97-24:~$ exit
exit
ubuntu@ip-172-31-97-24:~$ cd Netflix/
ubuntu@ip-172-31-97-24:~/Netflix$ ls
Dockerfile  package.json  pipeline.txt  public  tsconfig.json  tsconfig.node.json  vercel.json  vite.config.ts  yarn.lock
ubuntu@ip-172-31-97-24:~/Netflix$ docker build -dt --name netFlix -p 8081:80 netFlix:latest
Command 'docker' not found, did you mean:
  command 'docker' from snap docker (27.2.0)
  command 'docker' from deb docker.io (24.0.7~Ubuntu22.04.1)
  command 'docker' from deb docker (3.4.4-deb1-lubuntul.22.04.2)
  command 'socket' from deb socket (1.1-10build1)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-97-24:~/Netflix$ docker build -t netflix .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Instead, use the buildkit component to build images with BuildKit:
  https://docs.docker.com/guide/buildkit/
```

```

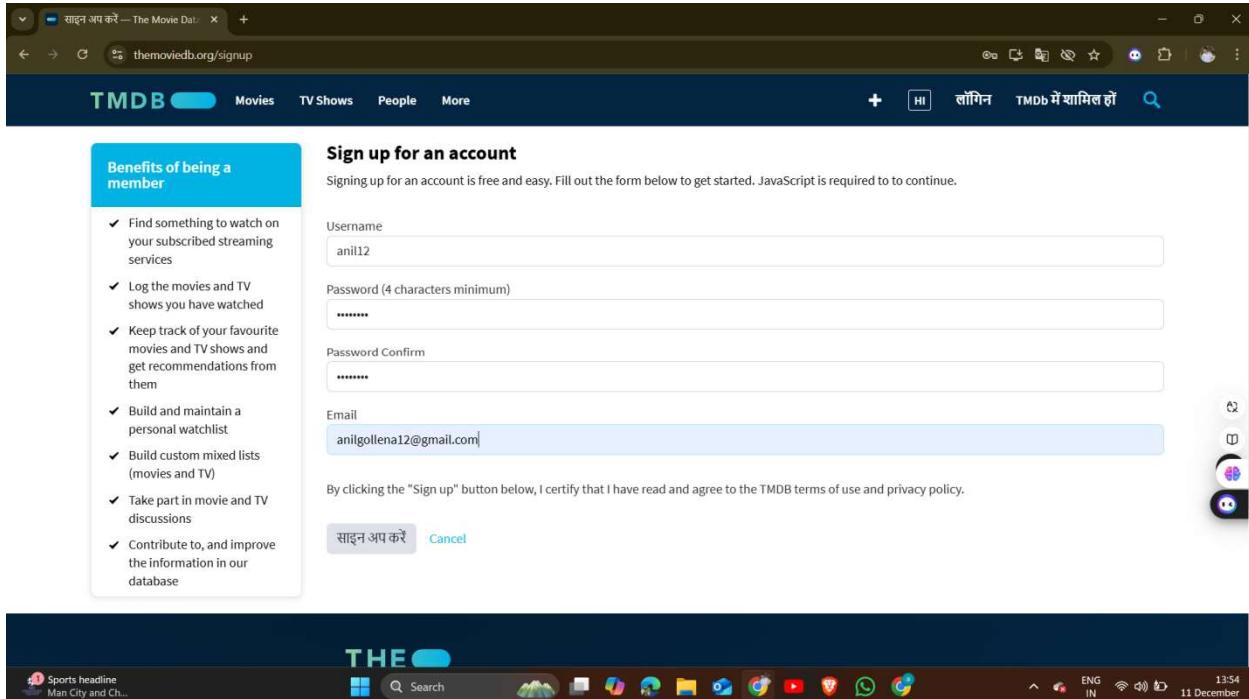
ubuntu@ip-172-31-97-24: ~/Netfli
--> 044c86c7b37f
Step 14/16 : FROM builder /app/dist .
--> 7ad50c9b6d1
Step 15/16 : EXPOSE 80
--> Running in 47887a47e864
Removing intermediate container 47887a47e864
--> 7ad50c9b6d1
Step 16/16 : ENTRYPOINT ["nginx", "-g", "daemon off;"]
--> Running in 1a96f10efb3d
Removing intermediate container 1a96f10efb3d
--> 7d3b00408547
Successfully built 7d3b00408547
Successfully tagged netflix:latest
ubuntu@ip-172-31-97-24:~/Netfli> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
netflix latest 7d3b00408547 8 seconds ago 584MB
<none> <none> ce81f3c8785a 5 seconds ago 844MB
nginx stable-alpine 60847f99ea69 4 months ago 48.8MB
node 16.17.0-alpine 5dcdf16157bd 2 years ago 115MB
ubuntu@ip-172-31-97-24:~/Netfli> docker run -dt --name netflix -p 8081:80 netflix:latest
Command 'docker' not found, did you mean:
command 'dockerc' from snap dockerc (27.2.0)
command 'dockerd' from deb socket (1.1-100+build1)
command 'dockers' from deb socket (1.1-100+build1)
command 'dockersocket' from deb socket (1.1-100+build1)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-97-24:~/Netfli> docker run -dt --name netflix -p 8081:80 netflix:latest
7fad50c9b6d1
ubuntu@ip-172-31-97-24:~/Netfli> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7fad50c9b6d1 netflix:latest "nginx -g 'daemon off;'" 5 seconds ago Up 4 seconds 0.0.0.0:8081->80/tcp, :::8081->80/tcp netflix
7fad50c9b6d1
ubuntu@ip-172-31-97-24:~/Netfli> docker rm 7d3b00408547
Error response from daemon: conflict: unable to delete 7d3b00408547 (must be forced) - image is being used by stopped container 7fad50c9b6d1
Error response from daemon: page not found
ubuntu@ip-172-31-97-24:~/Netfli> docker rmi 7d3b00408547
Error response from daemon: conflict: unable to delete 7d3b00408547 (must be forced) - image is being used by stopped container 7fad50c9b6d1
Error response from daemon: page not found
ubuntu@ip-172-31-97-24:~/Netfli> docker rm 7d3b00408547
Error response from daemon: conflict: unable to delete 7d3b00408547 (must be forced) - image is being used by stopped container 7fad50c9b6d1
ubuntu@ip-172-31-97-24:~/Netfli> sudo docker rm 7d3b00408547
Error response from daemon: No such container: 7d3b00408547
Error response from daemon: No such container: 7d3b00408547
ubuntu@ip-172-31-97-24:~/Netfli> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
netflix latest 7d3b00408547 5 minutes ago 584MB
<none> <none> ce81f3c8785a 5 minutes ago 844MB

```

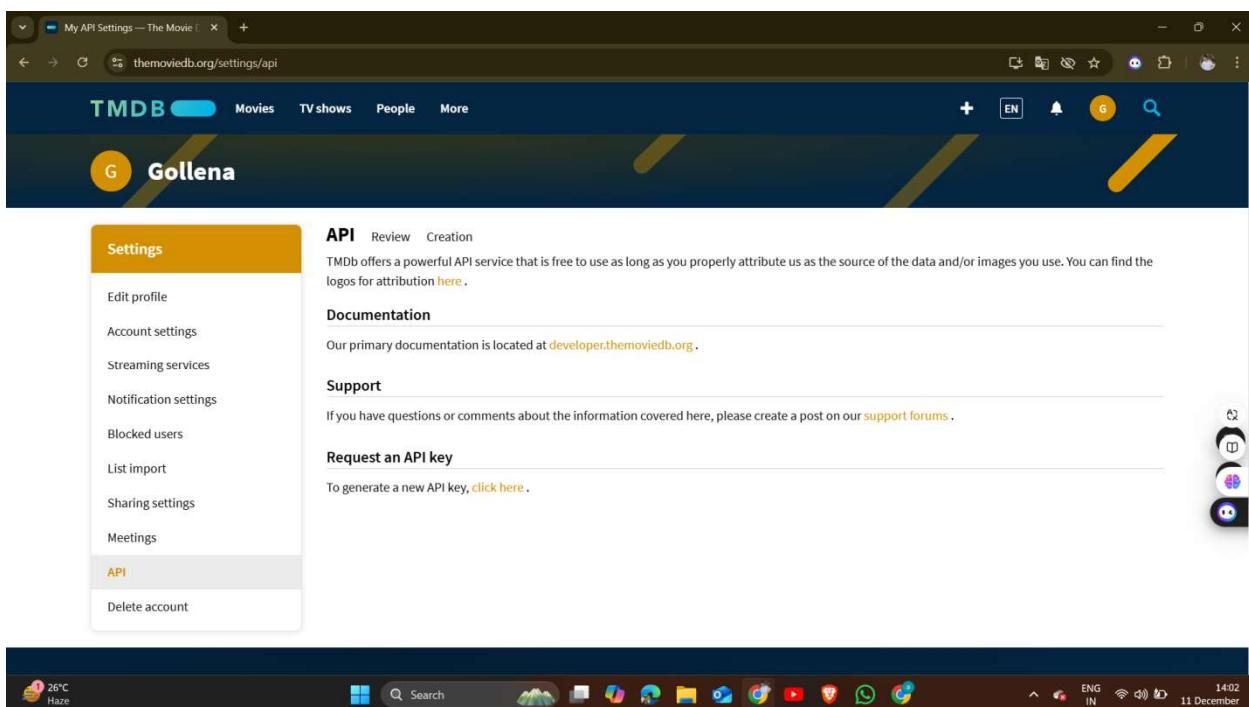
Now copy the public of ec2 instance, here Netflix page is deployed.



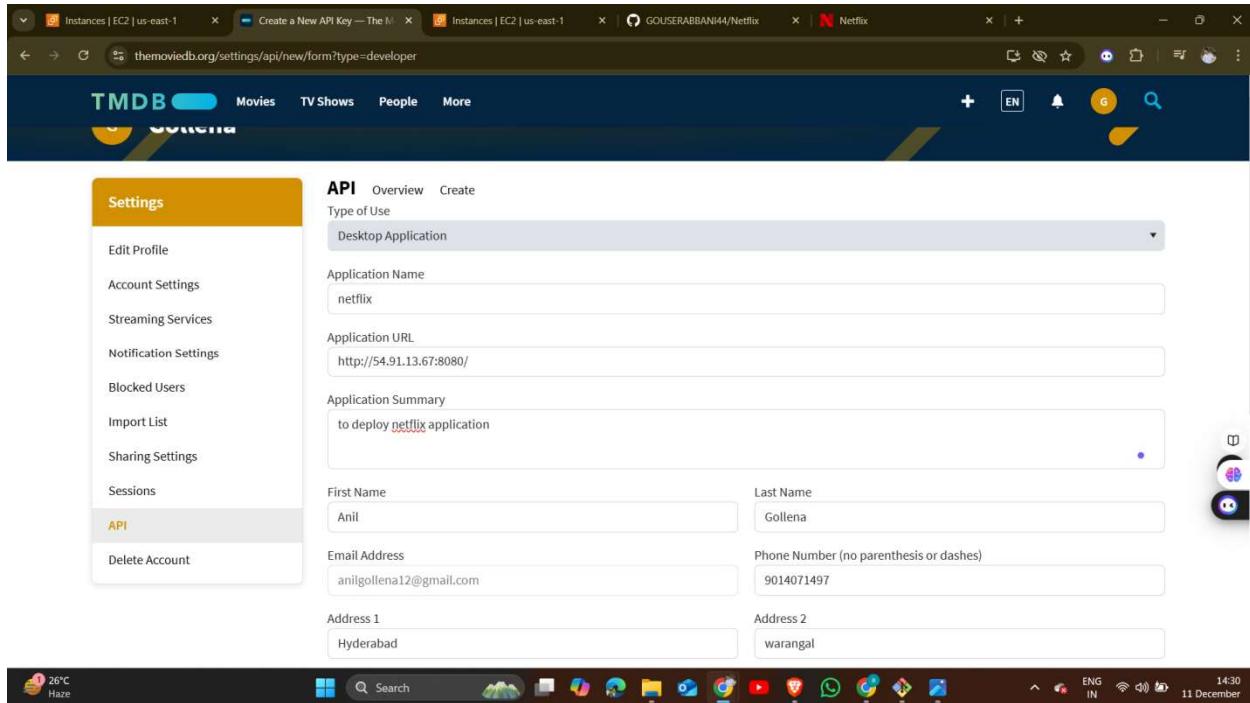
Now create TMDB movie database account in the browser here is the steps to create.



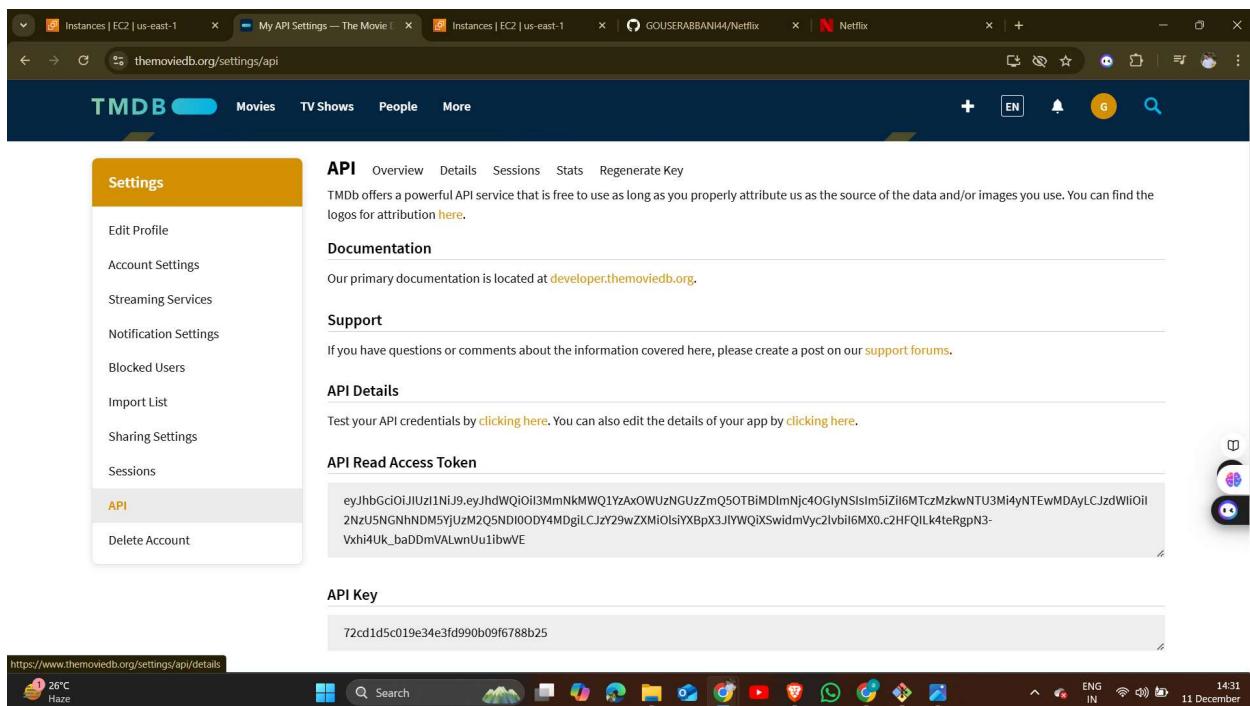
Now under settings → API → create a API



Choose desktop application.



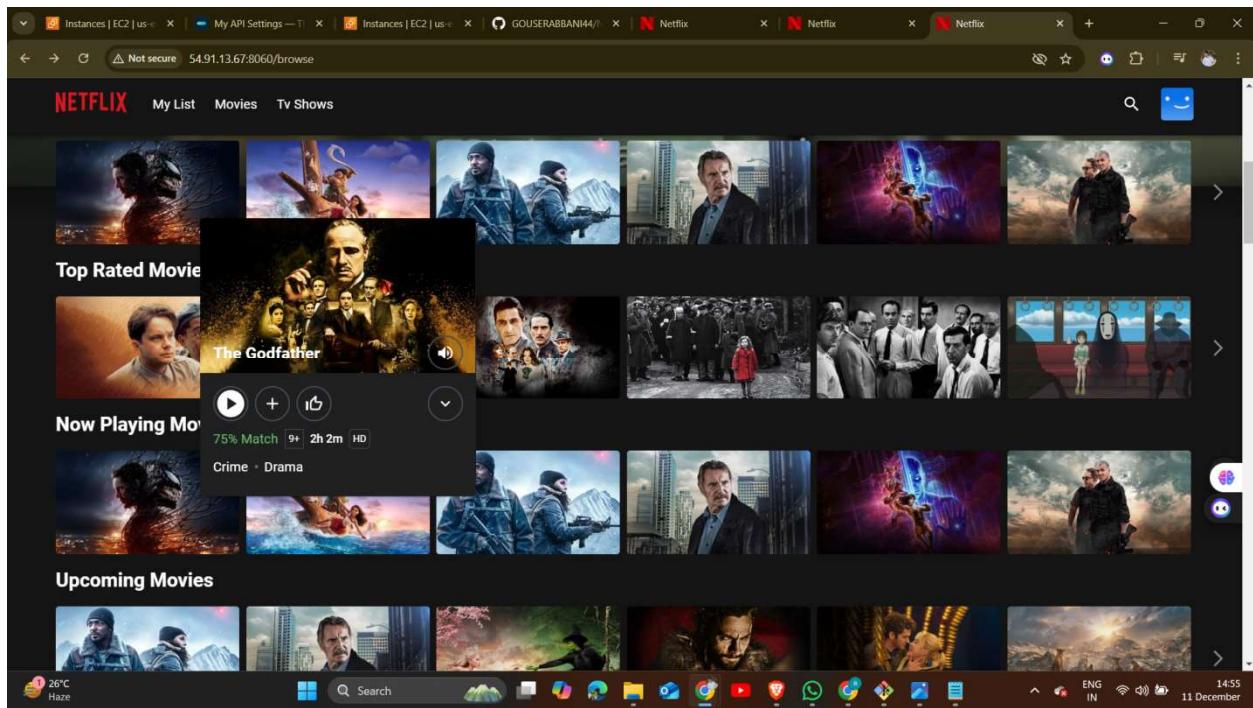
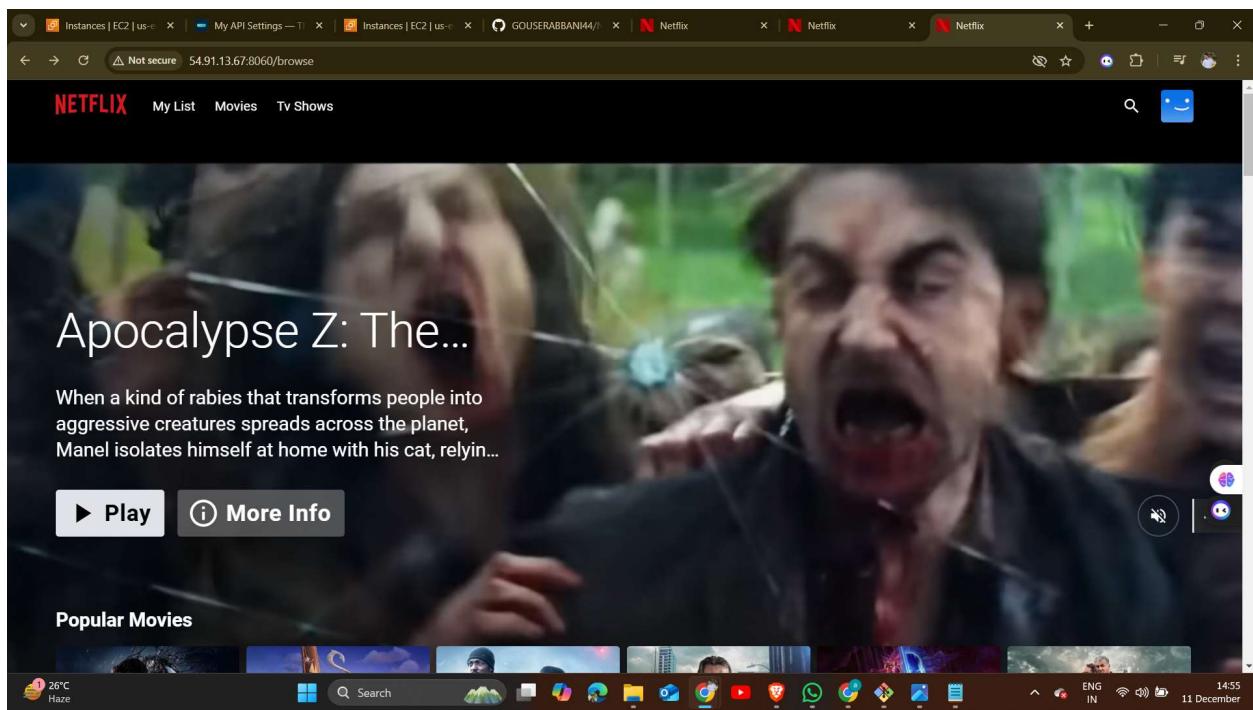
Here a new API is created copy that key



After creation of API now navigate to command shell and delete container and images and run the new container with the new API key

Now copy the public IP along with the port number in browser.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security. The main content area displays 'Instances (1/2) info' with a table showing two instances: 'netflix' (Running, t2.large, 2/2 checks passed) and 'ansible' (Terminated, t2.micro). A modal for the 'netflix' instance is open, showing its details: Instance ID (i-00df6c6abbbf48a01c), Public IPv4 address (54.91.13.67), Private IPv4 address (172.31.97.41), and Instance state (Running). The modal also contains a 'Copy' button for the Public IPv4 address.



PHSAE2: IMPLEMENTATION OF SECURITY WITH SONARQUBE AND TRIVY.

Overview of Tools:

- **SonarQube**: A code quality and security analysis tool that helps identify vulnerabilities, bugs, and code smells in source code during development.
 - **Trivy**: An open-source vulnerability scanner for containers, file systems, and dependencies.

Use Case:

- **SonarQube**: Ensures the codebase adheres to secure coding standards and identifies issues like SQL injection, cross-site scripting (XSS), and other vulnerabilities.
 - **Trivy**: Focuses on scanning container images, repositories, and dependencies for known security vulnerabilities.

here is the step by step process that I implementing security with sonarqube and trivy.

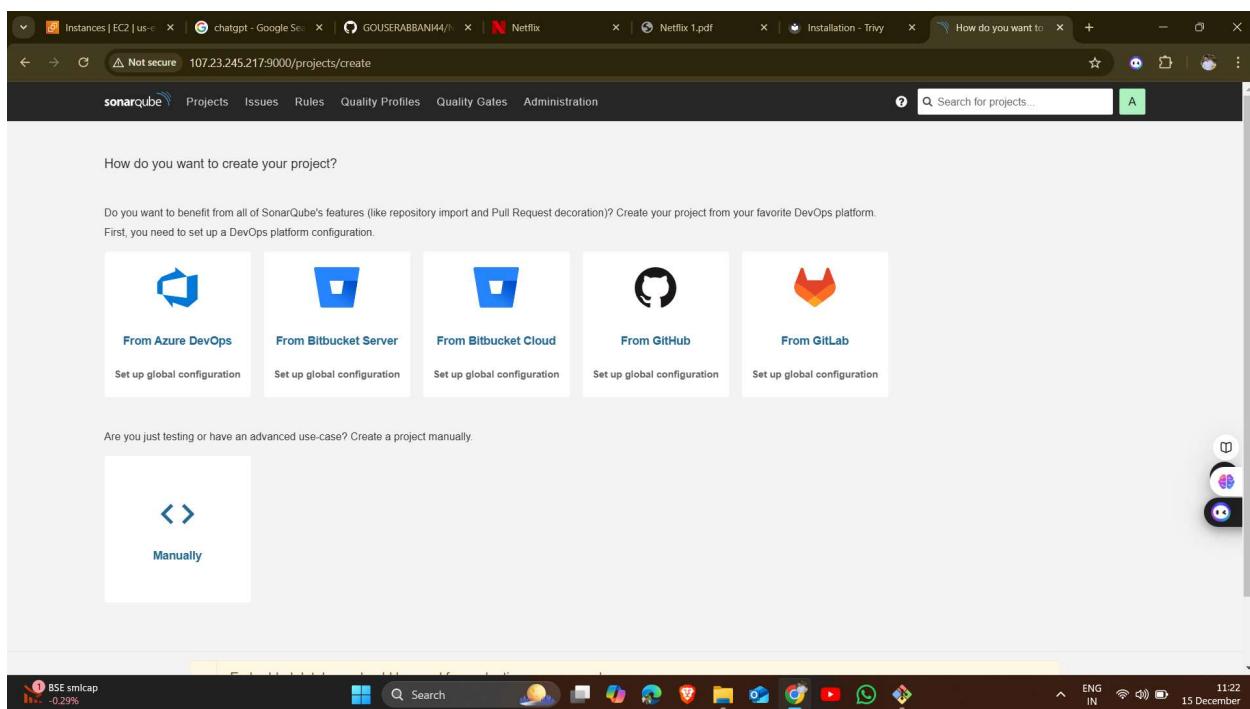
Now run the docker image with the SonarQube with the command

```
Docker run -dt --name sonar -p 9000:9000 sonarqube:lts-community
```

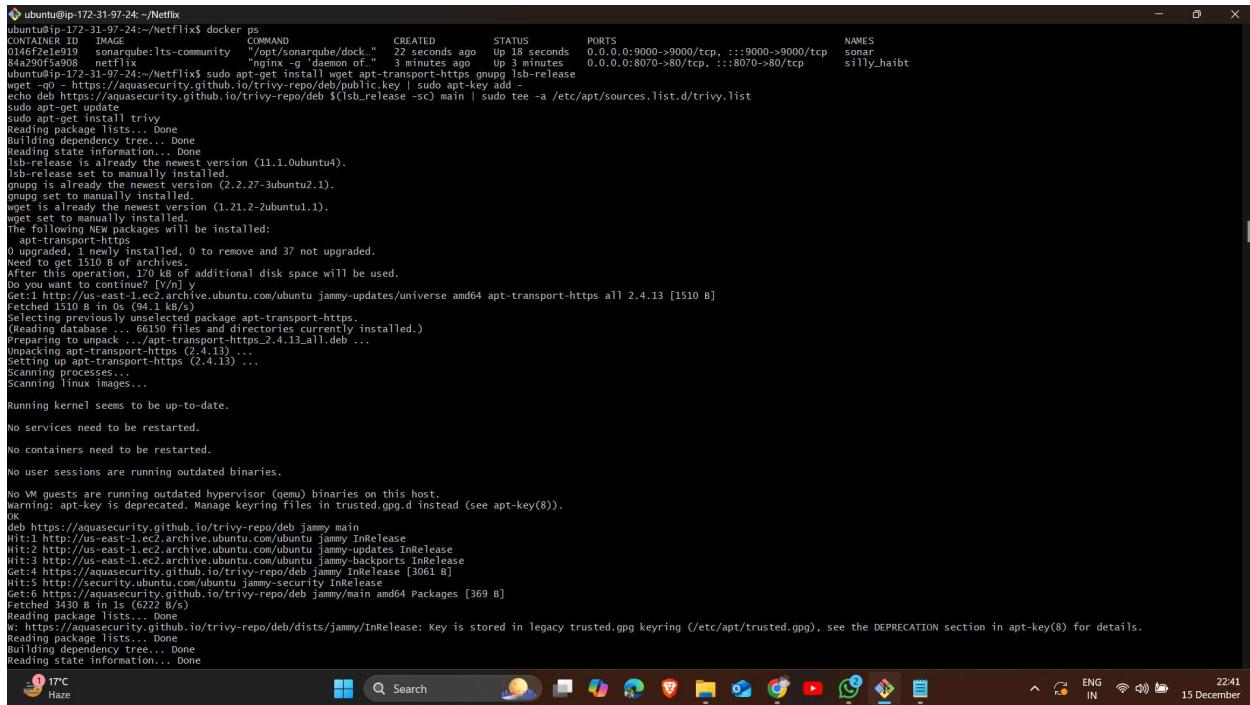
```

ubuntu@ip-172-31-97-24:~/Netflix
docker: Error response from daemon: Conflict. The container name "/sonar" is already in use by container "0146f2e1e9194cc4b4bea3662b87a50c74cadaa54ab5afa63c8ed363503f5ef2". You have to remove (or rename) that container or be able to reuse that name.
See 'docker run --help'.
ubuntu@ip-172-31-97-24:~/Netflix$ docker run -dt --name sonar -p 9000:9000 sonarqube:lts-community
docker: Error response from daemon: Conflict. The container name "/sonar" is already in use by container "0146f2e1e9194cc4b4bea3662b87a50c74cadaa54ab5afa63c8ed363503f5ef2". You have to remove (or rename) that container or be able to reuse that name.
See 'docker run --help'.
ubuntu@ip-172-31-97-24:~/Netflix$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
60a7252732a netflix "nginx -g 'daemon off;'" 2 minutes ago Up 2 minutes 0.0.0.0:8070->80/tcp, :::8070->80/tcp happy_pascal
ubuntu@ip-172-31-97-24:~/Netflix$ docker run -dt --name sonar -p 9000:9000 sonarqube:lts-community
docker: Error response from daemon: Conflict. The container name "/sonar" is already in use by container "0146f2e1e9194cc4b4bea3662b87a50c74cadaa54ab5afa63c8ed363503f5ef2". You have to remove (or rename) that container or be able to reuse that name.
See 'docker run --help'.
ubuntu@ip-172-31-97-24:~/Netflix$ ls
Dockerfile.kubernetes README.md index.html package.json pipeline.txt public src tsconfig.json vercel.json vite.config.ts yarn.lock
ubuntu@ip-172-31-97-24:~/Netflix$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
netflix latest eafbeb6d731d 3 hours ago 58MB
<none> <none> ce7a6f7e852c 3 hours ago 844MB
sonar <none> ce81f3c8785a 3 hours ago 844MB
sonarqube lts-community 2003d95074d 4 months ago 609MB
nginx stable-alpine 60847f99ea69 4 months ago 48.8MB
node 16.17.0-alpine 5dc1f6157bd 2 years ago 11MB
ubuntu@ip-172-31-97-24:~/Netflix$ sudo docker rmi -f 3eadd1da9b7ad
Deleted: sonarqubesha256:c137c407849de45a727f09db875779ad7b5784e0b02b096c1f8cd72e27a9fdc
Deleted: sha256:3eadd1da9b7ad19461a44ef41a28a28d0186f67efaf6754ba4e1e443c56e97
ubuntu@ip-172-31-97-24:~/Netflix$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
netflix latest eafbeb6d731d 3 hours ago 58MB
<none> <none> ce7a6f7e852c 3 hours ago 844MB
<none> <none> ce81f3c8785a 3 hours ago 844MB
nginx stable-alpine 60847f99ea69 4 months ago 48.8MB
node 16.17.0-alpine 5dc1f6157bd 2 years ago 11MB
ubuntu@ip-172-31-97-24:~/Netflix$ docker run -dt --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
Digest: sha256:c37c407849de45a727f09db875779ad7b5784e0b02b096c1f8cd72e27a9fdc
Status: Downloaded newer image for sonarqube:lts-community
docker: Error response from daemon: Conflict. The container name "/sonar" is already in use by container "0146f2e1e9194cc4b4bea3662b87a50c74cadaa54ab5afa63c8ed363503f5ef2". You have to remove (or rename) that container or be able to reuse that name.
See 'docker run --help'.
ubuntu@ip-172-31-97-24:~/Netflix$ docker run -dt --name Sonar -p 9000:9000 sonarqube:lts-community
213894bf9f954 sonarqube:lts-community "/opt/sonarqube/docker" 13 seconds ago Up 12 seconds 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp Sonar
60a7252732a netflix "nginx -g 'daemon off;'" 4 minutes ago Up 4 minutes 0.0.0.0:8070->80/tcp, :::8070->80/tcp happy_pascal
ubuntu@ip-172-31-97-24:~/Netflix$
```

Now access the sonarqube application by copying the IP along with the port number 9000



Now install trivy for that get address link from google and copy the address in the command shell.

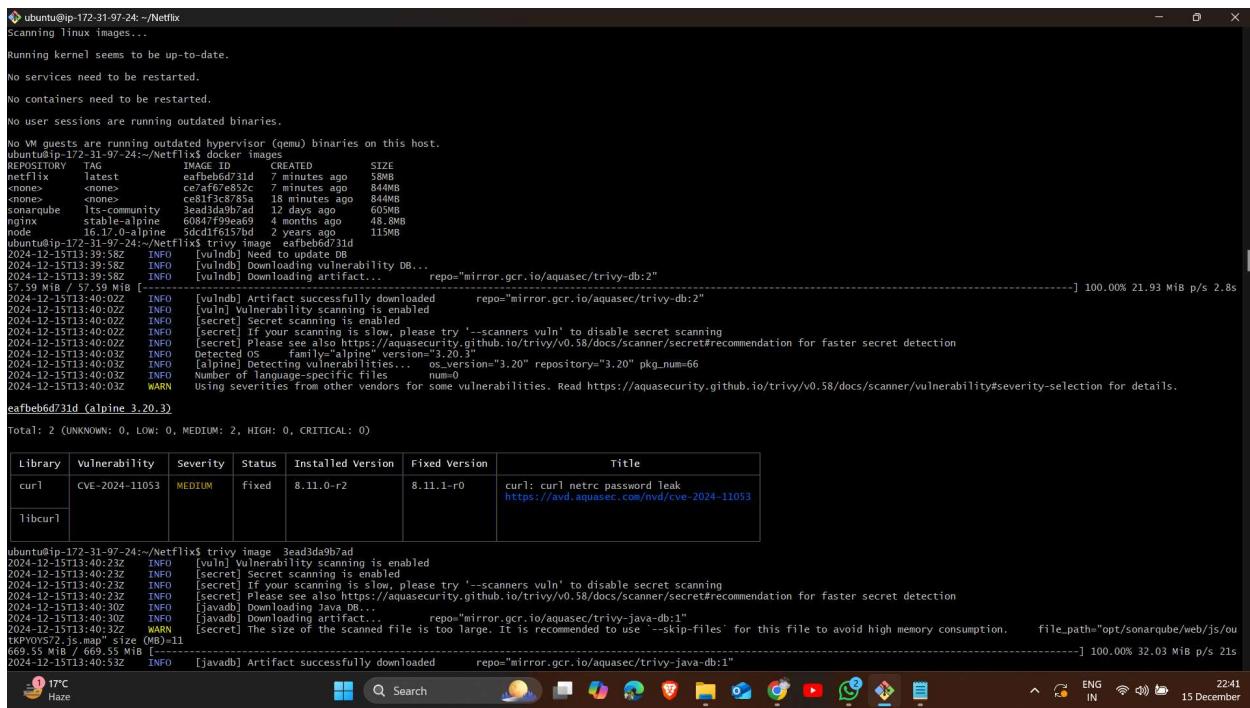


```
ubuntu@ip-172-31-97-24:~/Netflix
ubuntu@ip-172-31-97-24:~/Netflix$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
0146f2e1e919        sonarquebe:community   "/opt/sonarqube/dock"   22 seconds ago    Up 18 seconds      0.0.0.0:9000->9000/tcp, ::1:9000->9000/tcp
84a290fa5008        nginx               "nginx -g \"daemon off;\""
ubuntu@ip-172-31-97-24:~/Netflix$ sudo apt-get install wget apt-transport-https lsb-release
wget deb https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
apt-key fingerprint: 1E3457B19931BC0EB67D8DFA27339E6
echo deb https://aquasecurity.github.io/trivy-repo/deb $lsb_release -sc main | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed
wget is already the newest version (1.21.2-2ubuntu1.1).
wget set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 37 not upgraded.
Need to get 1510 B of archives.
After this operation, 170 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 https://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe apt-transport-https all 2.4.13 [1510 B]
Fetched 1510 B in 0s (94.1 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 66150 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.13_all.deb ...
Unpacking apt-transport-https (2.4.13) ...
Setting up apt-transport-https (2.4.13) ...
Scanning processes...
Scanning Linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
deb https://aquasecurity.github.io/trivy-repo/deb jammy main
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://aquasecurity.github.io/trivy-repo/deb jammy InRelease [3061 B]
Hit:5 https://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://aquasecurity.github.io/trivy-repo/deb jammy/main amd64 Packages [309 B]
Fetched 3430 B in 1s (2228 B/s)
Reading package lists... Done
w: https://aquasecurity.github.io/trivy-repo/deb/dists/jammy/InRelease: key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

ubuntu@ip-172-31-97-24:~/Netflix$
```

Now check the images through trivy.



```
ubuntu@ip-172-31-97-24:~/Netflix
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-97-24:~/Netflix$ docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
netflix            latest        eafbeb6d731d    7 minutes ago       98MB
<none>            <none>        cc7af67e852c    7 minutes ago       844MB
<none>            <none>        c4523a09794d    12 days ago        844MB
sonarqube          lts-community  3ead3d9b974d    4 months ago       48.8MB
nginx              stable-alpine  60847f99ea69    2 years ago        115MB
node               16.17.-alpine  5dcdf16157b0    2 years ago        115MB
ubuntu@ip-172-31-97-24:~/Netflix$ trivy image eafbeb6d731d
2024-12-15T13:39:58Z INFO [vulndb] Downloading vulnerability DB...
2024-12-15T13:39:58Z INFO [vulndb] Downloading artifact... repo="mirror.gcr.io/aquasec/trivy-db:2"
2024-12-15T13:39:58Z INFO [vulndb] Artifact successfully downloaded repo="mirror.gcr.io/aquasec/trivy-db:2"
2024-12-15T13:40:02Z INFO [vuln] Vulnerability scanning enabled
2024-12-15T13:40:02Z INFO [secret] Secret scanning is enabled
2024-12-15T13:40:02Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-12-15T13:40:02Z INFO [secret] Please see also https://aquasecurity.github.io/trivy/v0.58/docs/scanner/secret#recommendation for faster secret detection
2024-12-15T13:40:02Z INFO [detected] OS: family=Linux version='5.10.0-24-generic'
2024-12-15T13:40:03Z INFO [language] Detecting language vulnerabilities... osVersion='5.10.0-24-generic' repository='5.10.0-24-generic' pkgNum=0
2024-12-15T13:40:03Z INFO Number of language-specific files: num=0
2024-12-15T13:40:03Z WARN Using severities from other vendors for some vulnerabilities. Read https://aquasecurity.github.io/trivy/v0.58/docs/scanner/vulnerability#severity-selection for details.
eafbeb6d731d (alpine 3.20.3)

Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 2, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
curl	CVE-2024-11053	MEDIUM	fixed	8.11.0-r2	8.11.1-r0	curl: curl netrc password leak https://avd.aquasec.com/nvd/cve-2024-11053
libcurl						

```
ubuntu@ip-172-31-97-24:~/Netflix$ trivy image 3ead3d9b974d
2024-12-15T13:40:32Z INFO [vuln] Vulnerability scanning is enabled
2024-12-15T13:40:23Z INFO [secret] Secret scanning is enabled
2024-12-15T13:40:23Z INFO [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2024-12-15T13:40:23Z INFO [secret] Please see also https://aquasecurity.github.io/trivy/v0.58/docs/scanner/secret#recommendation for faster secret detection
2024-12-15T13:40:23Z INFO [detected] OS: family=Linux version='5.10.0-24-generic'
2024-12-15T13:40:30Z INFO [language] Detecting language vulnerabilities... osVersion='5.10.0-24-generic' repository='5.10.0-24-generic' pkgNum=0
2024-12-15T13:40:30Z INFO Number of language-specific files: num=0
2024-12-15T13:40:32Z WARN The size of the scanned file is too large. It is recommended to use '--skip-files' for this file to avoid high memory consumption. file_path="/opt/sonarqube/web/js/ouTKPQYS7.js.map" size (MB)=11
669.55 MB / 669.55 MB
2024-12-15T13:40:53Z INFO [javadv] Artifact successfully downloaded repo="mirror.gcr.io/aquasec/trivy-java-db:1"
```

```

ubuntu@ip-172-31-97-24: ~/Netflix
2024-12-15T13:40:30Z INFO [javadb] Downloading artifact... repo="mirror.gcr.io/aquasec/trivy-java-db:1"
2024-12-15T13:40:30Z WARN [secret] The size of the scanned file is too large. It is recommended to use '--skip-files' for this file to avoid high memory consumption. file_path="/opt/sonarqube/web/js/outputs/tkpvjyf7271.map" size (0x3e-11) 669.55 MiB [-]
2024-12-15T13:40:53Z INFO [javadb] Artifact successfully downloaded repo="mirror.gcr.io/aquasec/trivy-java-db:1"
2024-12-15T13:40:53Z INFO [javadb] Java DB is cached for 3 days. If you want to update the database more frequently, "trivy clean --java-db" command clears the DB cache.
2024-12-15T13:40:54Z INFO [secret] Detected OS: failoverrouter version="2.04"
2024-12-15T13:40:54Z INFO [ubuntu] Detecting vulnerabilities... os.version="2.04" pkg_num=143
2024-12-15T13:40:54Z INFO Number of language-specific files num=1
2024-12-15T13:40:54Z INFO [jar] Detecting vulnerabilities...
3ead3d9b7ad (ubuntu 22.04)

Total: 61 (UNKNOWN: 0, LOW: 45, MEDIUM: 16, HIGH: 0, CRITICAL: 0)

```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title	
coreutils	CVE-2016-2781	LOW	affected	8.32-4.1ubuntu1.2		coreutils: Non-privileged session can escape to the parent session in chroot https://avd.aquasec.com/nvd/cve-2016-2781	
dirmngr	CVE-2022-3219			2.2.27-3ubuntu2.1		gnupg: denial of service issue (resource consumption) using compressed packets https://avd.aquasec.com/nvd/cve-2022-3219	
gcc-12-base	CVE-2023-4039	MEDIUM			12.3.0-1ubuntu1~22.04		gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 https://avd.aquasec.com/nvd/cve-2023-4039
	CVE-2022-27943	LOW					binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle_const https://avd.aquasec.com/nvd/cve-2022-27943
gnupg-l10n	CVE-2022-3219				2.2.27-3ubuntu2.1		gnupg: denial of service issue (resource consumption) using compressed packets https://avd.aquasec.com/nvd/cve-2022-3219
gnupg-utils							
gpg							
gpg-agent							
gpg-wks-client							
gpg-wks-server							

```

ubuntu@ip-172-31-97-24: ~/Netflix

```

gpgv							
libc-bin	CVE-2016-20013			2.35-0ubuntu3.8		sha256crypt and sha512crypt through 0.6 allow attackers to cause a denial of service. https://avd.aquasec.com/nvd/cve-2016-20013	
libc6							
libgcc-s1	CVE-2023-4039	MEDIUM			12.3.0-1ubuntu1~22.04		gcc: -fstack-protector fails to guard dynamic stack allocations on ARM64 https://avd.aquasec.com/nvd/cve-2023-4039
	CVE-2022-27943	LOW					binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows stack exhaustion in demangle_const https://avd.aquasec.com/nvd/cve-2022-27943
libgcrypt20	CVE-2024-2236				1.9.4-3ubuntu3		libgcrypt: vulnerable to Marvin Attack https://avd.aquasec.com/nvd/cve-2024-2236
libgssapi-krb5-2	CVE-2024-26462	MEDIUM			1.19.2-2ubuntu0.4		krb5: Memory leak at /krb5/src/kdc/ndr.c https://avd.aquasec.com/nvd/cve-2024-26462
	CVE-2024-26458	LOW					krb5: Memory leak at /krb5/src/lib/rpc/pmap_rmt.c https://avd.aquasec.com/nvd/cve-2024-26458
	CVE-2024-26461					krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c https://avd.aquasec.com/nvd/cve-2024-26461	
libk5crypto3	CVE-2024-26462	MEDIUM				krb5: Memory leak at /krb5/src/kdc/ndr.c https://avd.aquasec.com/nvd/cve-2024-26462	
	CVE-2024-26458	LOW					krb5: Memory leak at /krb5/src/lib/rpc/pmap_rmt.c https://avd.aquasec.com/nvd/cve-2024-26458
	CVE-2024-26461						krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c https://avd.aquasec.com/nvd/cve-2024-26461
libkrb5-3	CVE-2024-26462	MEDIUM					krb5: Memory leak at /krb5/src/kdc/ndr.c https://avd.aquasec.com/nvd/cve-2024-26462
	CVE-2024-26458	LOW					krb5: Memory leak at /krb5/src/lib/rpc/pmap_rmt.c https://avd.aquasec.com/nvd/cve-2024-26458
	CVE-2024-26461						krb5: Memory leak at /krb5/src/lib/gssapi/krb5/k5sealv3.c https://avd.aquasec.com/nvd/cve-2024-26461
libkrb5support0	CVE-2024-26462	MEDIUM					krb5: Memory leak at /krb5/src/kdc/ndr.c https://avd.aquasec.com/nvd/cve-2024-26462
	CVE-2024-26458	LOW				krb5: Memory leak at /krb5/src/lib/rpc/pmap_rmt.c https://avd.aquasec.com/nvd/cve-2024-26458	

ubuntu@ip-172-31-97-24: ~/Netflix

login	CVE-2023-29383			1:4.8.1-2ubuntu2.2		shadow: Improper input validation in shadow-utils package utility chfn https://avd.aquasec.com/nvd/cve-2023-29383
ncurses-base	CVE-2023-45918			6.3-2ubuntu0.1		ncurses: NULL pointer dereference in tgetstr in tinfo/lib_ttermcap.c https://avd.aquasec.com/nvd/cve-2023-45918
	CVE-2023-50495					ncurses: segmentation fault via _nc_wrap_entry() https://avd.aquasec.com/nvd/cve-2023-50495
	CVE-2023-45918					ncurses: NULL pointer dereference in tgetstr in tinfo/lib_ttermcap.c https://avd.aquasec.com/nvd/cve-2023-45918
	CVE-2023-50495					ncurses: segmentation fault via _nc_wrap_entry() https://avd.aquasec.com/nvd/cve-2023-50495
openssl	CVE-2024-41996			3.0.2-0ubuntu1.18		openssl: remote attackers (from the client side) to trigger unnecessarily expensive server-side... https://avd.aquasec.com/nvd/cve-2024-41996
passwd	CVE-2023-29383			1:4.8.1-2ubuntu2.2		shadow: Improper input validation in shadow-utils package utility chfn https://avd.aquasec.com/nvd/cve-2023-29383
wget	CVE-2021-31879	MEDIUM		1.21.2-2ubuntu1.1		wget: authorization header disclosure on redirect https://avd.aquasec.com/nvd/cve-2021-31879

2024-12-15T13:40:54Z INFO Table result includes only package filenames. Use '--format json' option to get the full path to the package file.

Java (.jar)

Total: 37 (UNKNOWN: 0, LOW: 4, MEDIUM: 14, HIGH: 18, CRITICAL: 1)

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
ch.qos.logback:logback-classic y in logback receiver (sonar-scanner-engine-shaded-9.9.8.100196-all.jar) 3-6378	CVE-2023-6378	HIGH	fixed	1.2.10	1.3.12, 1.4.12, 1.2.13	logback: serialization vulnerability https://avd.aquasec.com/nvd/cve-2023-6378
ch.qos.logback:logback-classic (sonar-application-9.9.8.100196.jar)						
ch.qos.logback:logback-core (sonar-scanner-engine-shaded-9.9.8.100196-all.jar)						

17°C Haze

Search

22:41 15 December

ubuntu@ip-172-31-97-24: ~/Netflix

0-8908						
com.google.guava:guava (sonar-python-plugin-3.24.1.11916.jar)						
com.google.guava:guava creation (sonar-scanner-engine-shaded-9.9.8.100196-all.jar) 3-2976	CVE-2023-2976	MEDIUM		31.1-jre		guava: insecure temporary directory https://avd.aquasec.com/nvd/cve-2023-2976
com.google.guava:guava (sonar-application-9.9.8.100196.jar)						
com.google.guava:guava via temporary directory (sonar-scanner-engine-shaded-9.9.8.100196-all.jar)	CVE-2020-8908	LOW				guava: local information disclosure created with unsafe permissions https://avd.aquasec.com/nvd/cve-2020-8908
0-8908						
com.google.guava:guava (sonar-application-9.9.8.100196.jar)						
com.google.protobuf:protobuf-java ty in Protocol Buffers (sonar-csharp-plugin-8.51.0.59060.jar) 4-7294	CVE-2024-7254	HIGH		3.21.12	3.25.5, 4.27.5, 4.28.2	protobuf: StackOverflow vulnerability https://avd.aquasec.com/nvd/cve-2024-7254
com.google.protobuf:protobuf-java (sonar-vbnet-plugin-8.51.0.59060.jar)						
com.google.protobuf:protobuf-java (sonar-python-plugin-3.24.1.11916.jar)				3.21.7		

17°C Haze

Search

22:41 15 December

```

ubuntu@ip-172-31-97-24:~/Netfli
(sonar-javascript-plugin-9.13.0.20537.jar)
4-47554

commons-io:commons-io
(sonar-scanner-engine-shaded-9.9.8.100196-all.jar)

commons-io:commons-io (sonar-application-9.9.8.100196.jar)

commons-io:commons-io (sonar-iac-plugin-1.11.0.2847.jar) | 2.8.0

commons-io:commons-io (sonar-python-plugin-3.24.1.11916.jar)

commons-io:commons-io (sonar-ruby-plugin-1.11.0.3905.jar)

io.netty:netty-codec-http
ources without Limits or
(netty-codec-http-4.1.94.Final.jar)
4-29025

io.netty:netty-common (netty-common-4.1.94.Final.jar) | CVE-2024-47535 | MEDIUM | 4.1.94.Final | 4.1.115 | netty: Denial of Service attack on windows app using Netty
4-47535

17°C Haze

```

Search ENG IN WiFi 2241 15 December

```

ubuntu@ip-172-31-97-24:~/Netfli
n debug-log output
(sonar-application-9.9.8.100196.jar)
3-44483

org.eclipse.jgit:org.eclipse.jgit
(sonar-scanner-engine-shaded-9.9.8.100196-all.jar)
3-4759

org.elasticsearch.elasticsearch (elasticsearch-7.17.15.jar) | CVE-2023-49921 | MEDIUM | 7.17.15 | 7.17.16, 8.11.2 | elasticsearch: Insertion of Sensitive Information into Log
3-49921

disk unencrypted
4-23444

service when processing
4-23450

org.postgresql:postgresql (postgresql-42.5.1.jar) | CVE-2024-1597 | CRITICAL | 42.5.1 | 42.2.28, 42.3.9, 42.4.4, 42.5.5, 42.6.1, 42.7.2 | pgjdbc: PostgreSQL JDBC Driver allows attacker to inject SQL if using PreferQueryMode=SIMPLE...
4-1597

org.yaml:snakeyaml (snakeyaml-1.33.jar) | CVE-2022-1471 | HIGH | 1.33 | 2.0 | SnakeYAML: Constructor Deserialization Remote Code Execution
2-1471

org.yaml:snakeyaml (sonar-application-9.9.8.100196.jar)

ubuntu@ip-172-31-97-24:~/Netfli$ sudo apt install java-17 -
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package java-17

```

Search ENG IN WiFi 2241 15 December

Here the trivy check is passed for both the images.

This I implemented security using sonarqube and Trivy.

PHASE 3: AUTOMATE THE WHOLE DEPLOYMENT USING JENKINS CI/CD PIPELINE.

About Jenkins:

- Jenkins is an open-source automation server widely used for **continuous integration (CI)** and **continuous delivery/deployment (CD)**.
- It automates repetitive tasks like building, testing, and deploying software, enhancing productivity and reducing manual effort.
- Jenkins supports a vast ecosystem of **plugins** for integrating with tools, frameworks, and platforms.

Jenkins CI/CD Pipeline:

- A **Jenkins Pipeline** is a suite of plugins that enables defining and automating the steps involved in CI/CD as **code** (Pipeline-as-Code).
- It is written using a **Groovy-based Domain-Specific Language (DSL)**.
- Supports **Declarative** and **Scripted** syntax for creating pipelines.

Advantages of jenkins pipelines:

- Facilitates **DevOps** practices by automating CI/CD processes.
- Ensures faster delivery and higher software quality.
- Easily customizable for various workflows.

HERE, install Jenkins and set up before setting up install java dependency 17th version.

```
ubuntu@ip-172-31-97-24:~/Netfix
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

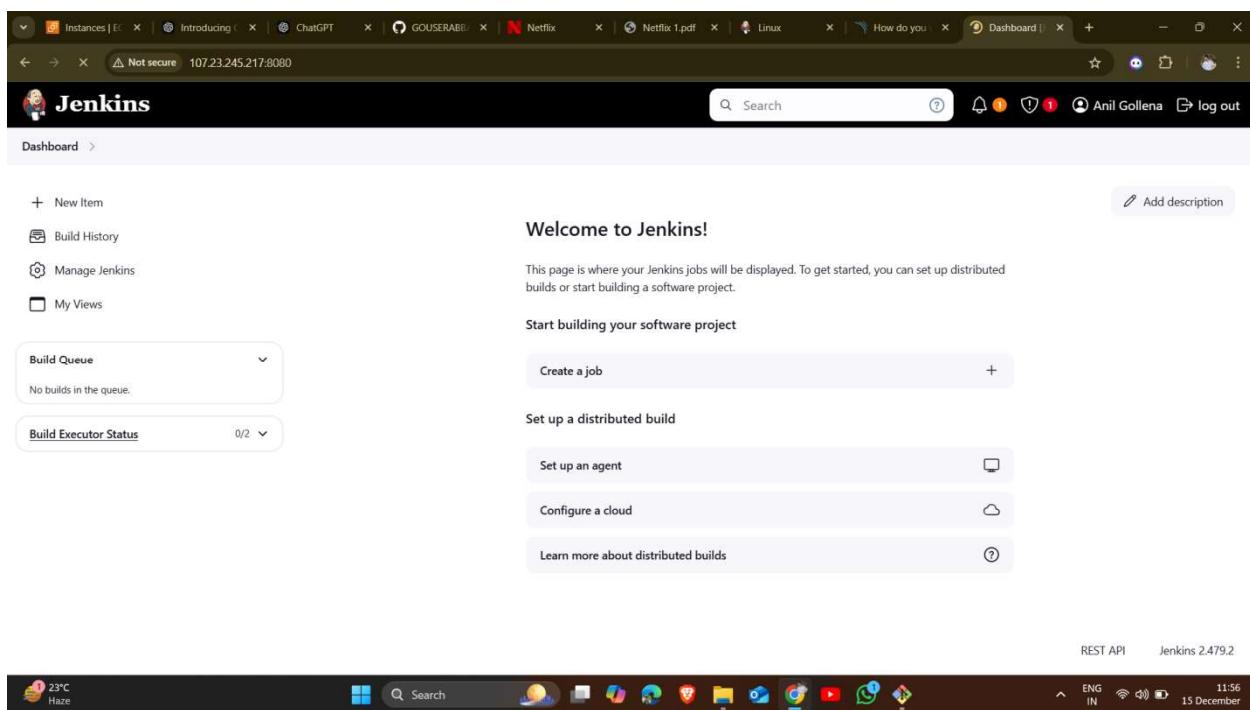
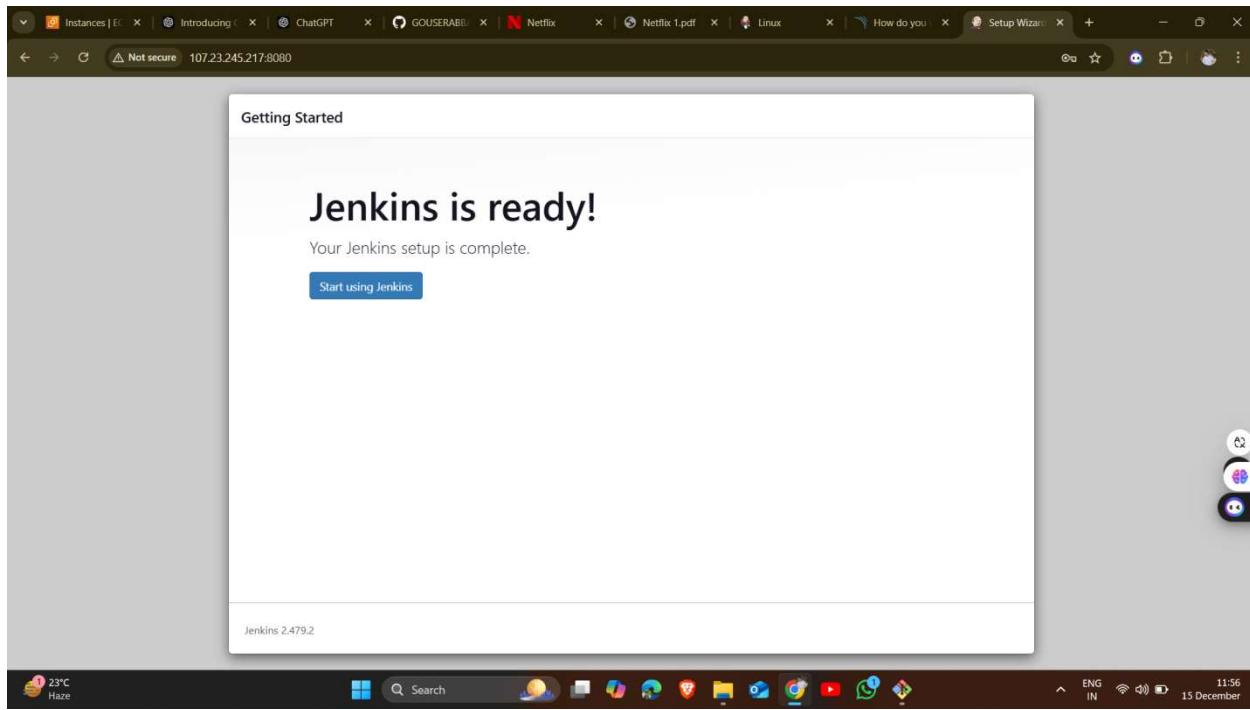
[100%[=====>] 3.10K --.-KB/s    in 0s

2024-12-15 13:43:33 (50.00 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 https://aptency-repo.deb.debian.org/debian InRelease
Hit:5 https://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:9 https://pkg.jenkins.io/debian-stable binary/ Packages [28.2 kB]
Fetched 31.1 kB in 1s (49.2 kB/s)
Reading package lists...
Reading package lists... Done
W: https://aquareasource.github.io/trivy-repo/deb/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 37 not upgraded.
Need to get 94.3 MB of archives.
After this operation, 96.1 MB of disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ Jenkins 2.479.2 [94.1 MB]
Fetched 94.1 MB in 9s (10.5 MB/s)
Reading package lists... Done
Reading database... Done
The following previously untracked package net-tools,
which were previously installed, are currently installed.
(Reading database ... /7250 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Preparing previously unpacked package jenkins.
Preparing previously unpacked package Jenkins 2.479.2_all.deb ...
Unpacking Jenkins (2.479.2)
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up Jenkins (2.479.2) ...
Processing triggers for libsystemd-systemunit (2.51-1) ...
Processing triggers for multi-user.target.wants/jenkins.service + /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning Linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-97-24:~/Netfix$ sudo systemctl start jenkins
```

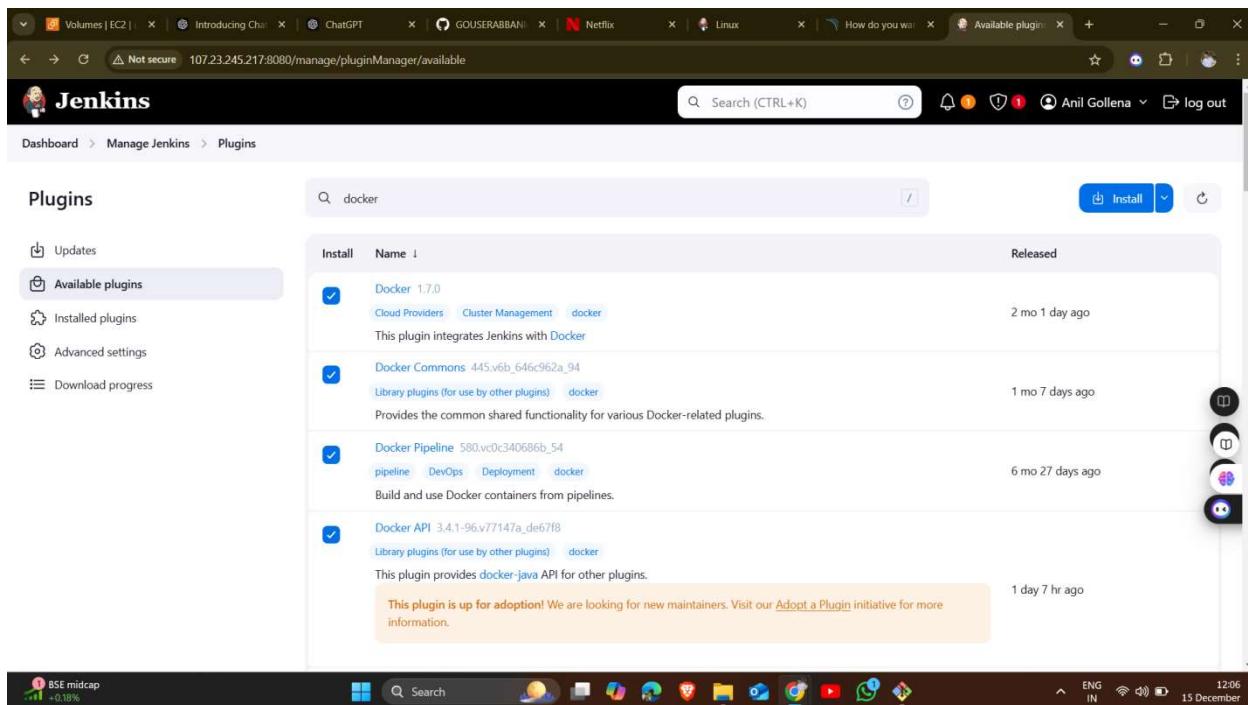
Now copy the public ip of Jenkins, access the Jenkins.



Initially, go to Dashboard → manage Jenkins → pluggins → Available pluggins

Here install few pluggins here are:

- i. Eclipse temurin installer
- ii. Sonarqube scanner
- iii. NodeJS
- iv. Email extension template
- v. OWSP dependency check
- vi. Prometheus metricks
- vii. And all docker related pluggins



The screenshot shows the Jenkins plugin manager interface. On the left, there's a sidebar with options: Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area lists several available plugins:

- Eclipse Temurin installer 1.5: Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net. Last updated 2 yr 2 mo ago.
- SonarQube Scanner 2.17.3: External Site/Tool Integrations, Build Reports. This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. Last updated 27 days ago.
- NodeJS 1.6.2: npm. NodeJS Plugin executes NodeJS script as a build step. Last updated 4 mo 4 days ago.
- Email Extension Template 219.v14ffff547f78d: Build Notifiers, emailext. This plugin allows administrators to create global templates for the Extended Email Publisher. Last updated 22 days ago. A note says: "This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information."
- OWASP Dependency-Check 5.6.0: Security, DevOps, Build Tools, Build Reports. This plugin can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities. Last updated 3 days 0 hr ago.
- Prometheus metrics 801.v98e119d8eeda...: Metrics, Monitoring, Prometheus. This plugin provides monitoring and alerting for Prometheus metrics. Last updated 1 day 19 hours ago.

At the bottom of the screen, the taskbar shows the URL <https://plugins.jenkins.io/emailext-template>, the weather (21°C Clear), and the system status (19:18, ENG IN, WiFi, battery, 15 December).

After installation of related plugins,

Now go to dashboard add sonarqube and docker credentials.

The screenshot shows the Jenkins dashboard under the System Configuration section. It includes the following management options:

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...
- Appearance**: Configure the look and feel of Jenkins.
- Security**: Secure Jenkins; define who is allowed to access/use the system.
- Credentials**: Configure credentials.
- Credential Providers**: Configure the credential providers and types.
- Users**: Create/delete/modify users that can log in to this Jenkins.

At the bottom of the screen, the taskbar shows the URL <https://107.23.245.217:8080/manage/credentials>, the weather (24°C Haze), and the system status (12:13, ENG IN, WiFi, battery, 15 December).

In the security field add credentials for that create a sonarqube token here is

The screenshot shows the SonarQube Administration - Users page. The user 'Administrator' (admin) is listed with the following details:

SCM Accounts	Last connection	Groups	Tokens
sonar-administrators	< 1 hour ago	sonar-users	0

A 'Create User' button is located at the top right of the users section.

Warning: Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

The screenshot shows the SonarQube Security - Tokens page. A new token named "mytoken" has been created. The token details are as follows:

Name	Type	Expires in
mytoken	User	Never

A "Copy" button is available to copy the token value: `squ_8ce18e5c4e10403c7082b72c23eeb10f3580a44d`.

An "Enter a new password" field is present for password management.

Copy that token and paste it in Jenkins

The screenshot shows the Jenkins 'New credentials' page. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains 'sonarqube'. The 'ID' field is 'sonarqube'. The 'Description' field is 'sonar authentication'. A blue 'Create' button is at the bottom.

Now add docker credentials using username and password

The screenshot shows the Jenkins 'New credentials' page. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field is 'anil132'. The 'Password' field contains '.....'. The 'ID' field is 'docker'. The 'Description' field is 'docker authentication'. A blue 'Create' button is at the bottom.

Here the credentials are added.

The screenshot shows the Jenkins Global credentials (unrestricted) page. It lists two credentials:

ID	Name	Kind	Description
sonarqube	sonar authentication	Secret text	sonar authentication
docker	anil132/***** (docker authentication)	Username with password	docker authentication

At the bottom, there are icons for S, M, and L, and a REST API link.

Now add the Netflix project into the sonarqube here is the steps to add.

The screenshot shows the SonarQube Projects creation page. It provides several ways to create a project:

- From Azure DevOps
- From Bitbucket Server
- From Bitbucket Cloud
- From GitHub
- From GitLab
- Manually (with a code editor icon)

At the bottom, there is a REST API link.

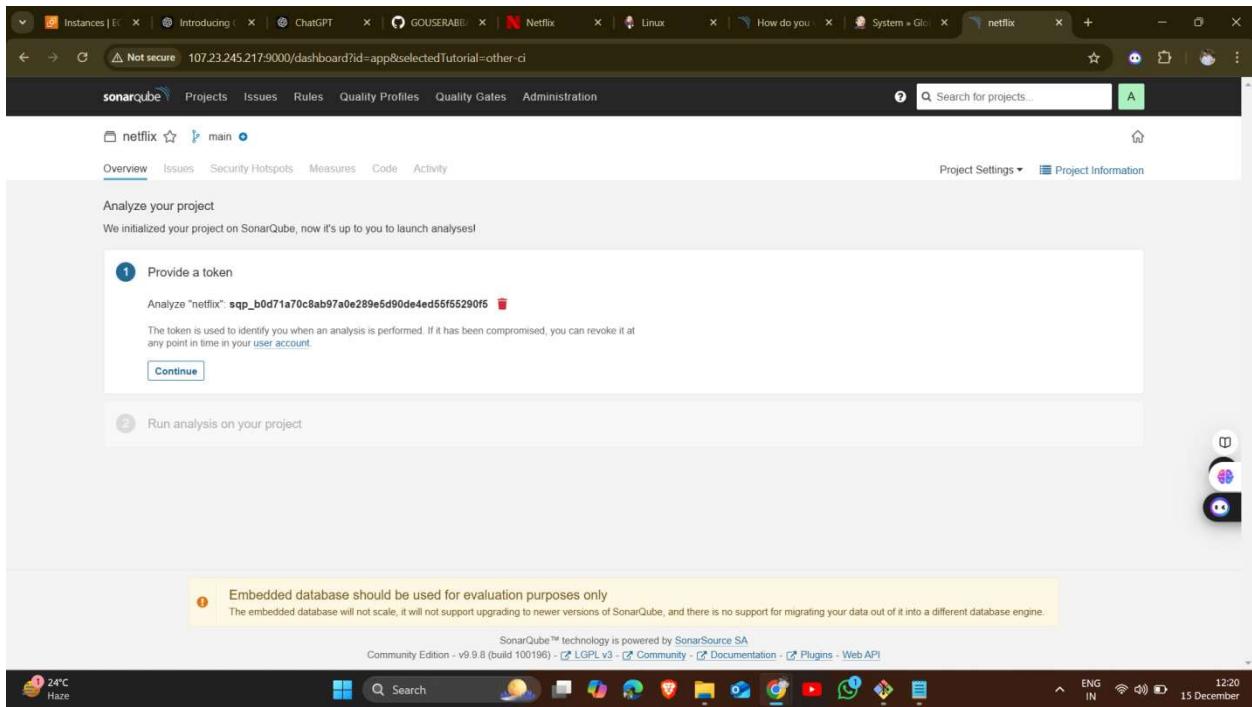
Now under projects → create a new project.

The screenshot shows a browser window with multiple tabs open. The active tab is 'Not secure 107.23.245.217:9000/admin/projects_management'. The page title is 'sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration'. A modal dialog box is centered, titled 'Create Project'. It contains fields for 'Project display name' (set to 'netflix'), 'Project key' (set to 'app'), 'Main branch name' (set to 'main'), and 'Visibility' (set to 'Public'). Below these fields are 'Create' and 'Cancel' buttons. In the background, there's a sidebar with 'Projects Management' and a message about using an embedded database. The bottom of the screen shows a taskbar with various icons and system status.

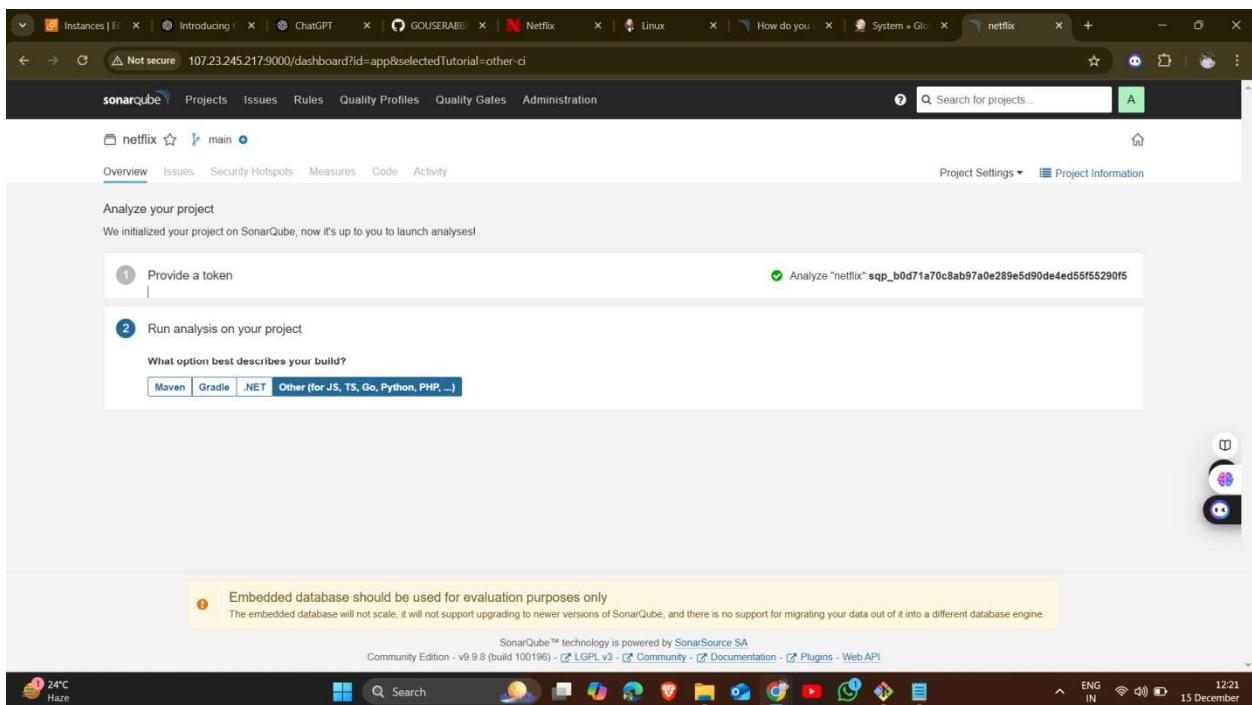
Now generate token

The screenshot shows a browser window with multiple tabs open. The active tab is 'Not secure 107.23.245.217:9000/dashboard?id=app&selectedTutorial=other-ci'. The page title is 'sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration'. The main content area is titled 'Analyze your project' with the sub-instruction 'We initialized your project on SonarQube, now it's up to you to launch analyses!'. A large callout box is centered, step 1: 'Provide a token'. It has two options: 'Generate a project token' (selected) and 'Use existing token'. Under 'Generate a project token', there are fields for 'Token name' (set to 'Analyze "netflix"') and 'Expires in' (set to '30 days'). A note below says: 'Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your user account. See the documentation for more information.' Step 2: 'Run analysis on your project' is partially visible below. The bottom of the screen shows a taskbar with various icons and system status.

Now generate.



Now press continue



Now click other

Analyze your project
We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token Analyze "netflix" sqp_b0d71a70c8ab97a0e289e5d90de4ed55f55290f5

2 Run analysis on your project

What option best describes your build?
Maven Gradle .NET Other (for JS, TS, Go, Python, PHP,...)

What is your OS?
Linux Windows macOS

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

After clicking the OS as linux it displays the commands and paste it on the local

Download and unzip the Scanner for Linux

```
export SONAR_SCANNER_VERSION=4.7.0.2747
export SONAR_SCANNER_HOME=$HOME/.sonar/sonar-scanner-$SONAR_SCANNER_VERSION-linux
curl --create-dirs -sLo $HOME/.sonar/sonar-scanner.zip https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-$SONAR_SCANNER_VERSION-linux.zip
unzip -o $HOME/.sonar/sonar-scanner.zip -d $HOME/.sonar/
export PATH=$SONAR_SCANNER_HOME/bin:$PATH
export SONAR_SCANNER_OPTS=-server
```

Configure a SONAR_TOKEN environment variable in your CI settings

- Add an environment variable called SONAR_TOKEN
- Give it the following value sqp_b0d71a70c8ab97a0e289e5d90de4ed55f55290f5

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder

```
sonar-scanner \
-Dsonar.projectKey=app \
-Dsonar.sources=. \
-Dsonar.host.url=http://107.23.245.217:9000
```

Please visit the [official documentation of the Scanner](#) for more details.

Is my analysis done? If your analysis is successful, this page will automatically refresh in a few moments.

Here the project is added into the sonarqube

The screenshot shows the SonarQube web interface. At the top, there's a navigation bar with tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is at the top right. Below the navigation, there's a 'Filters' section on the left containing various quality gate and reliability metrics. The main area displays a list of projects with one entry: 'netflix'. A message below it says 'Project's Main Branch is not analyzed yet.' with a 'Configure analysis' button. On the right side, there's a sidebar with icons for dashboard, community, documentation, and plugins. At the bottom, there's a footer with system status information like 'ENG IN', a date ('15 December'), and a time ('12:22').

Now go to Jenkins dashboard → manage Jenkins → under tools

Add configuration

The screenshot shows the Jenkins 'Manage Jenkins' page. Under the 'System Configuration' heading, there are several sections: 'System' (configure global settings), 'Tools' (configure tools and installers, highlighted in a light blue box), 'Nodes' (add, remove, control nodes), 'Managed files' (e.g., settings.xml for Maven), 'Clouds' (add, remove, configure cloud instances), 'Plugins' (add, remove, enable plugins), 'Appearance' (configure look and feel), 'Security' (secure Jenkins), 'Credentials' (configure credentials), 'Users' (create, delete, modify users), and 'Credential Providers' (configure credential providers). The URL in the address bar is '107.23.245.217:8080/manage/'. The bottom of the screen shows the Windows taskbar with various pinned icons.

Add JDK

The screenshot shows the Jenkins 'Tools' configuration interface. Under the 'JDK installations' section, a new entry named 'jdk17' is being added. The 'Install automatically' checkbox is checked. A sub-section titled 'Install from adoptium.net' shows the selected version as 'jdk-17.0.4.1+1'. Below the main form are 'Save' and 'Apply' buttons. The desktop taskbar at the bottom shows various icons and the date '15 December'.

No need to change git installations

The screenshot shows the Jenkins 'Tools' configuration interface. Under the 'Git installations' section, a new entry named 'Default' is being added. The 'Path to Git executable' field contains 'git'. The 'Install automatically' checkbox is unchecked. Below the main form are 'Add Git' and 'Save' buttons. The desktop taskbar at the bottom shows various icons and the date '15 December'.

Add sonarqube scanner

The screenshot shows the Jenkins management interface for adding tools. The URL is [Not secure 107.23.245.217:8080/manage/configureTools/](http://107.23.245.217:8080/manage/configureTools/). The page title is "SonarQube Scanner installations". A sub-section titled "Add SonarQube Scanner" is open, showing a configuration form. The "Name" field contains "sonar-server". The "Install automatically" checkbox is checked. Under "Install from Maven Central", the "Version" dropdown is set to "SonarQube Scanner 6.2.1.4610". There is also an "Add Installer" dropdown. At the bottom are "Save" and "Apply" buttons.

Add NodeJS

The screenshot shows the Jenkins management interface for adding tools. The URL is [Not secure 107.23.245.217:8080/manage/configureTools/](http://107.23.245.217:8080/manage/configureTools/). The page title is "SonarQube Scanner installations". A sub-section titled "Add SonarQube Scanner" is open, showing a configuration form. The "Name" field contains "sonar-server". The "Install automatically" checkbox is checked. Under "Install from nodejs.org", the "Version" dropdown is set to "NodeJS 16.2.0". There is a note about forcing the 32-bit architecture and a section for global npm packages to install. At the bottom are "Save" and "Apply" buttons.

Add docker

The screenshot shows the Jenkins Manage Jenkins > Tools page. A new tool named "Docker" is being added. The "Name" field contains "docker". The "Installation root" field contains "docker.com", which is highlighted in orange with a warning message: "⚠ docker.com is not a directory on the Jenkins controller (but perhaps it exists on some agents)". The "Install automatically" checkbox is checked. Under "Download from docker.com", the "Docker version" dropdown is set to "latest". There is a "Save" button at the bottom.

Add OWSP dependency check

The screenshot shows the Jenkins Manage Jenkins > Tools page. A new tool named "Dependency-Check" is being added. The "Name" field contains "DP-check". The "Install automatically" checkbox is checked. Under "Install from github.com", the "Version" dropdown is set to "dependency-check 9.2.0". There is a "Save" button at the bottom.

Now save and apply

Now go to Jenkins dashboard → manage Jenkins → system

Add sonarqube installations

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'System' section. A modal window titled 'SonarQube installations' is open, showing a list of installations. One entry, 'sonar-server', is selected. The configuration fields for this entry include:

- Name: sonar-server
- Server URL: http://107.23.245.217:9000 (Default is http://localhost:9000)
- Server authentication token: sonar authentication (selected from a dropdown menu)
- Advanced options (button)
- Save and Apply buttons

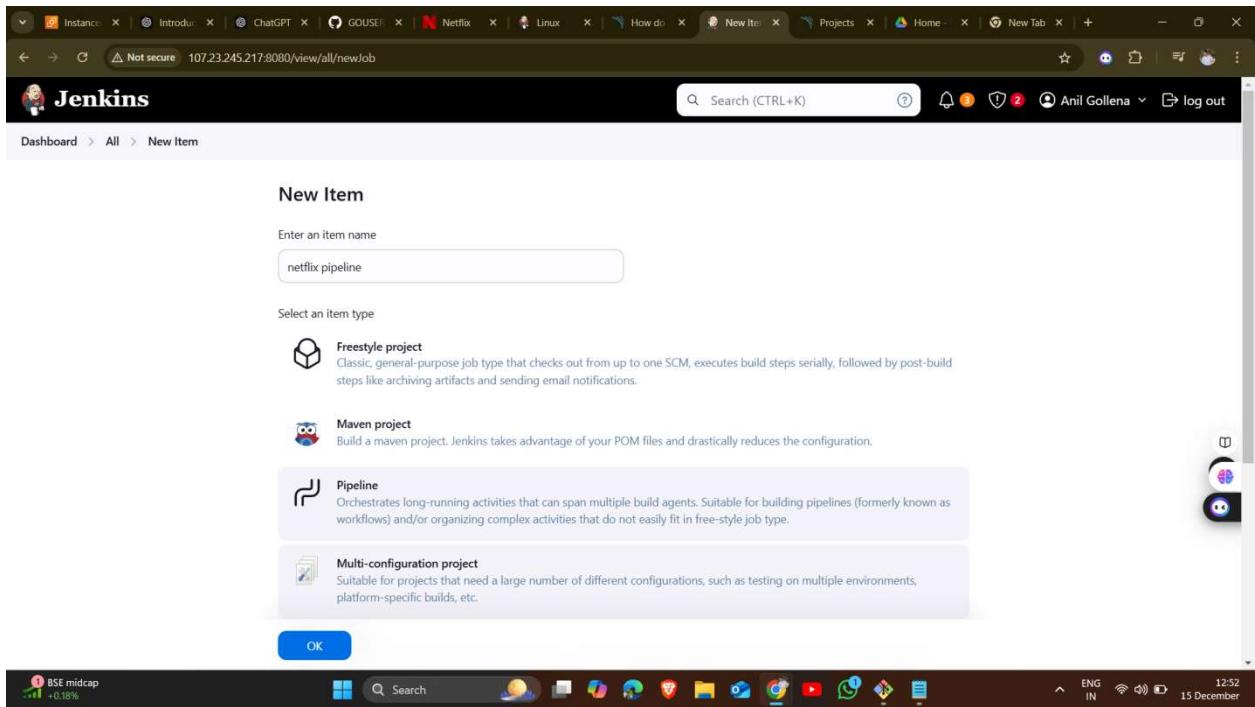
The Jenkins dashboard header at the top shows various tabs like 'Dashboard', 'Manage Jenkins', 'System', etc. The bottom taskbar includes icons for search, file explorer, and other applications.

Now save and apply

Now back to dashboard create a new job for pipeline

The screenshot shows the Jenkins dashboard. The main header says 'Jenkins'. The left sidebar has links for 'New Item', 'Build History', 'Manage Jenkins', 'My Views', and 'Docker Swarm Dashboard'. The right side features a 'Welcome to Jenkins!' message and a 'Start building your software project' section with links for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The bottom taskbar shows system status like 'BSE midcap +0.18%', network connectivity, and system date/time.

New item → give pipeline name → select pipeline → OK.



Under pipeline script give script below:

#Groovy script:

```
pipeline {  
    agent any  
    tools {  
        jdk 'jdk17'  
        nodejs 'node16'  
    }  
    environment {  
        SCANNER_HOME = tool 'sonar-server'  
    }  
}
```

```

stages {

    stage('Clean Workspace') {
        steps {
            cleanWs()
        }
    }

    stage('Checkout from Git') {
        steps {
            git branch: 'main', url:
            'https://github.com/Gouserabbani44/Netflix.git'
        }
    }

    stage('SonarQube Analysis') {
        steps {
            withSonarQubeEnv('sonar-server') {
                sh """
                    $SCANNER_HOME/bin/sonar-scanner \
                    -Dsonar.projectName=Netflix \
                    -Dsonar.projectKey=app
                    """
            }
        }
    }
}

```

```

}

stage('Quality Gate') {

    steps {
        script {
            // Wait for the quality gate to pass before proceeding
            waitForQualityGate abortPipeline: false, credentialsId: 'sonarkey'
        }
    }
}

stage('Install Dependencies') {

    steps {
        sh 'npm install'
    }
}

stage('OWASP Dependency Check') {

    steps {
        // Make sure that the Dependency-Check tool is configured correctly in
        Jenkins

        dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit --
        disableNodeAudit', odcInstallation: 'DP-check'
    }
}

```

```

stage('TRIVY File System Scan') {
    steps {
        // Ensure Trivy is installed and available on the agent
        sh 'trivy fs . > trivyfs.txt'
    }
}

stage('Docker Build & Push') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {
                sh """
                    docker build --build-arg
                    TMDB_V3_API_KEY=72cd1d5c019e34e3fd990b09f6788b25 -t netflix .
                    docker tag netflix anil132/netflix:latest
                    docker push anil132/netflix:latest
                """
            }
        }
    }
}

stage('TRIVY Image Scan') {
    steps {

```

```

sh 'trivy image anil132/netflix:latest > trivyimage.txt'

}

}

stage('Deploy to Container') {

    steps {

        sh 'docker run -d --name netflixcont1 -p 8082:80 anil132/netflix:latest'

    }

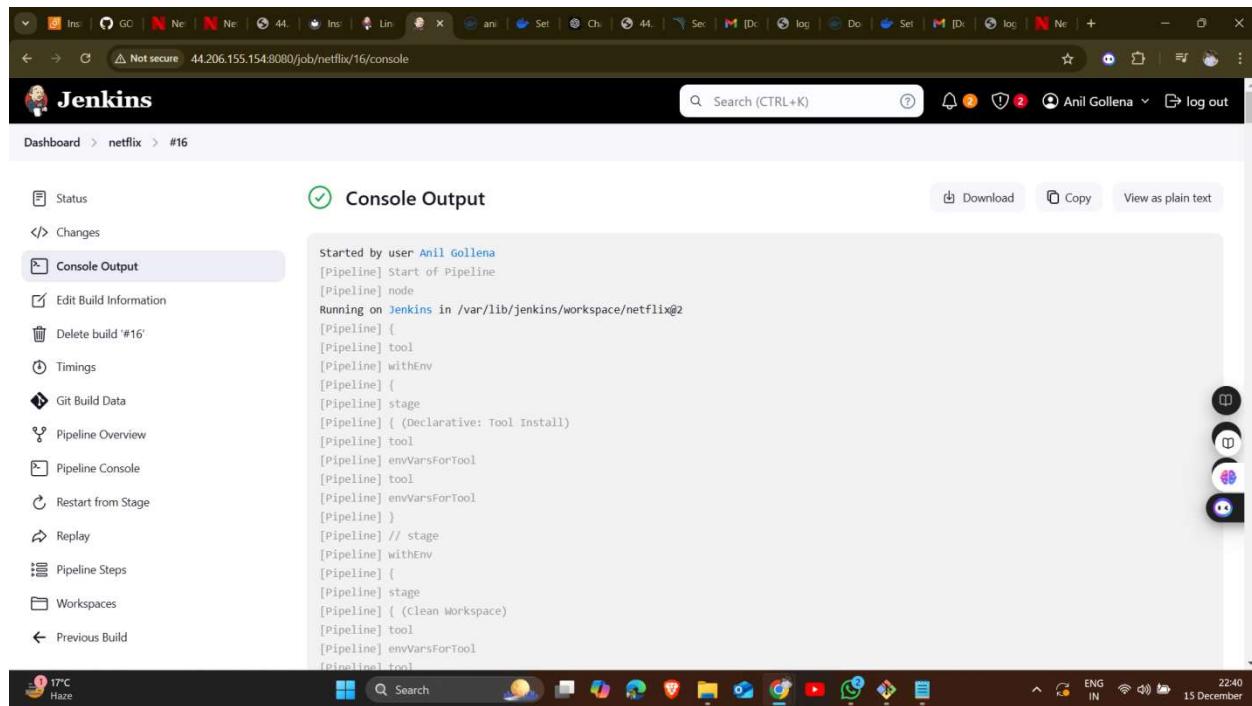
}

}

}

```

Now save and apply the job.



Jenkins

Dashboard > netflix >

Status: **Passed**

netflix deployment

SonarQube Quality Gate

netflix **Passed**
server-side processing: **Success**

Permalinks

- Last build (#16), 8 min 9 sec ago
- Last stable build (#16), 8 min 9 sec ago
- Last successful build (#16), 8 min 9 sec ago
- Last failed build (#15), 10 min ago
- Last unsuccessful build (#15), 10 min ago
- Last completed build (#16), 8 min 9 sec ago

Builds

Filter
Today
#16 5:00 pm
(X) #15 4:58 pm

17°C Haze

Now check the sonarqube, there you can see the output that all test case is passed.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... A

My Favorites All

Filters

Quality Gate: Passed

Reliability (Bugs)

- A rating: 1
- B rating: 0
- C rating: 0
- D rating: 0
- E rating: 0

Security (Vulnerabilities)

- A rating: 1
- B rating: 0
- C rating: 0
- D rating: 0
- E rating: 0

Security Review (Security Hotspots)

- ≥ 80%: 0
- 70% - 80%: 0

1 project(s)

Perspective: Overall Status Sort by: Name

netflix **Passed**

Last analysis: 35 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	0 A	0.0% E	18 A	0.0% (red)	0.0% (green)	3.2k S TypeScript...

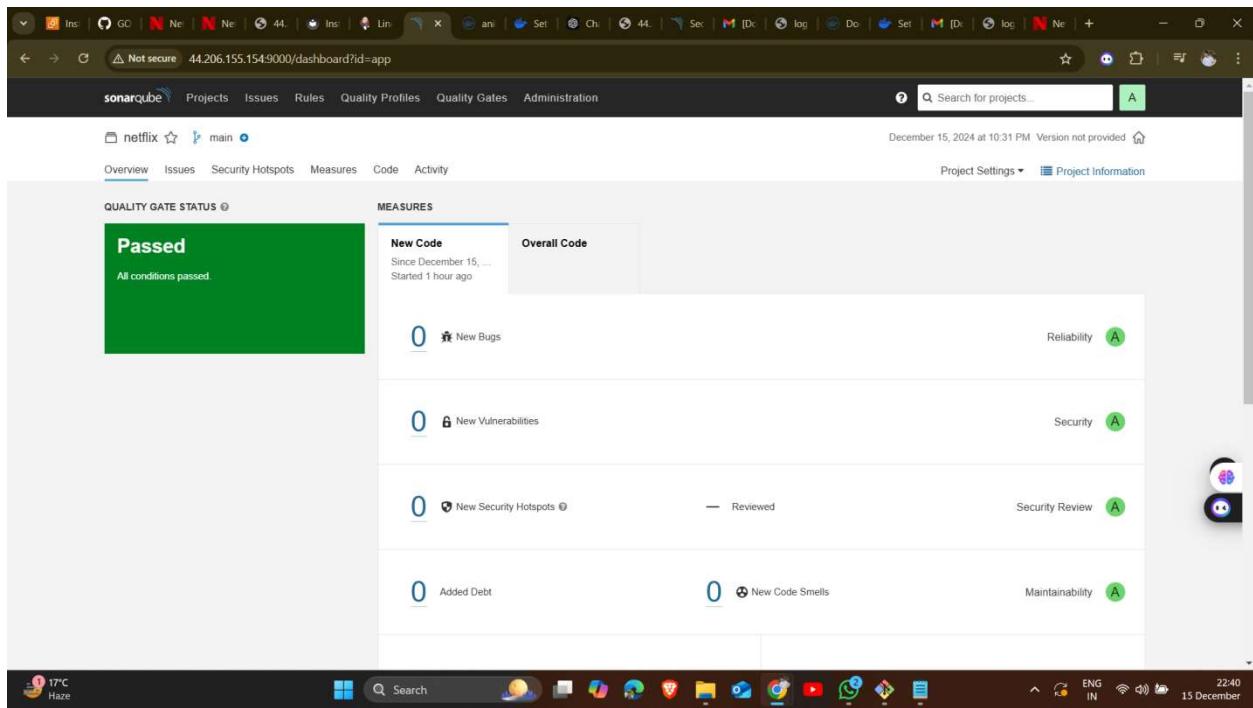
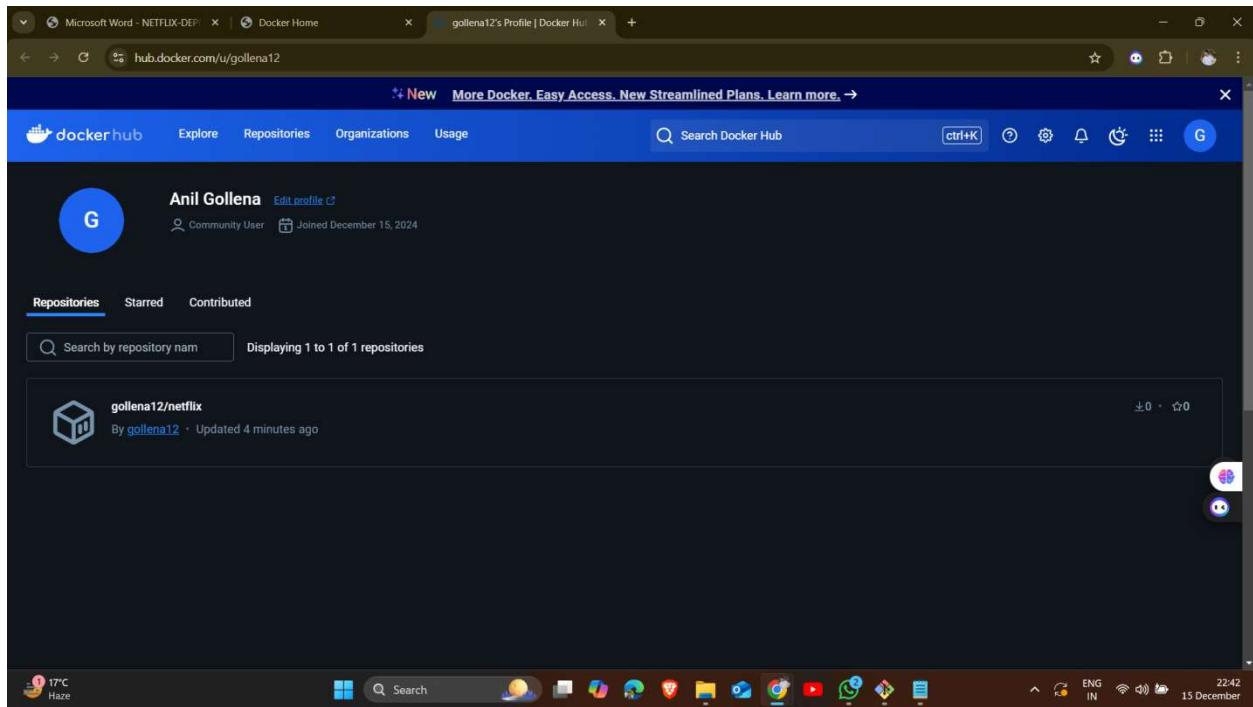
1 of 1 shown

Embedded database should be used for evaluation purposes only.
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

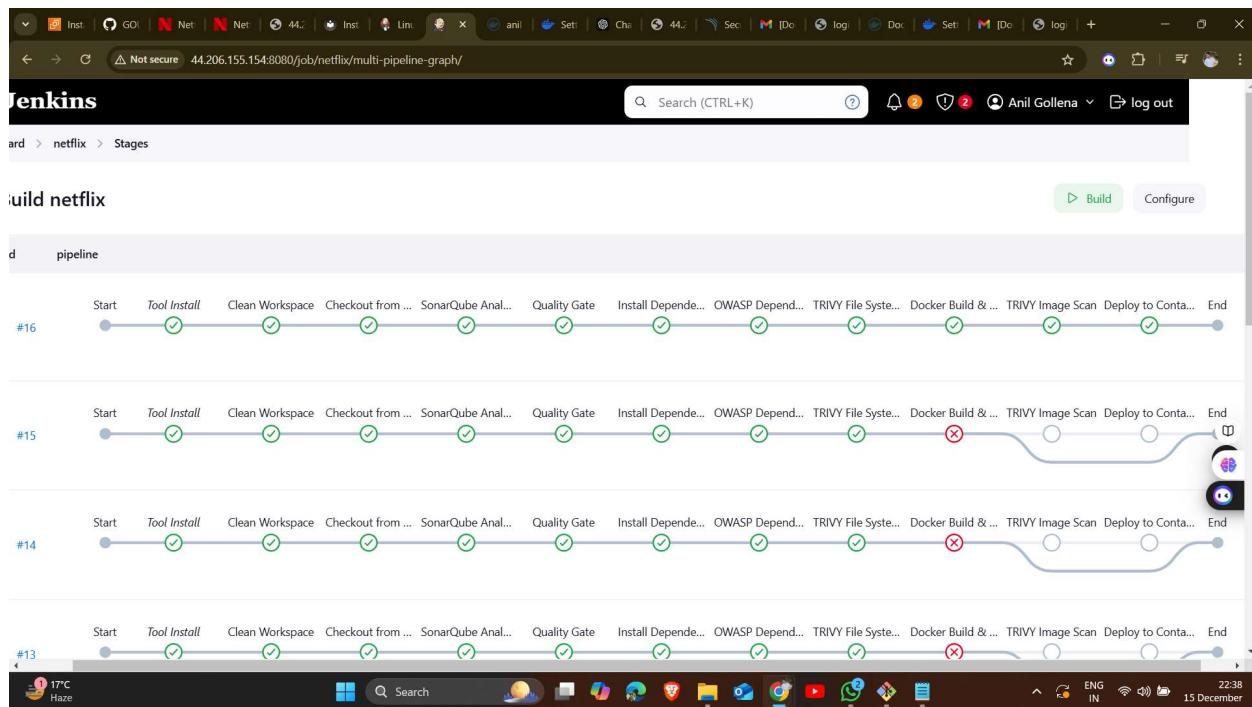
SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.8 (build 100196) - [LGPL v3](#) - [Community](#) - [Documentation](#) - [Plugins](#) - [Web API](#)

25°C Haze

As well as, the image which has created in uploaded in docker hub.

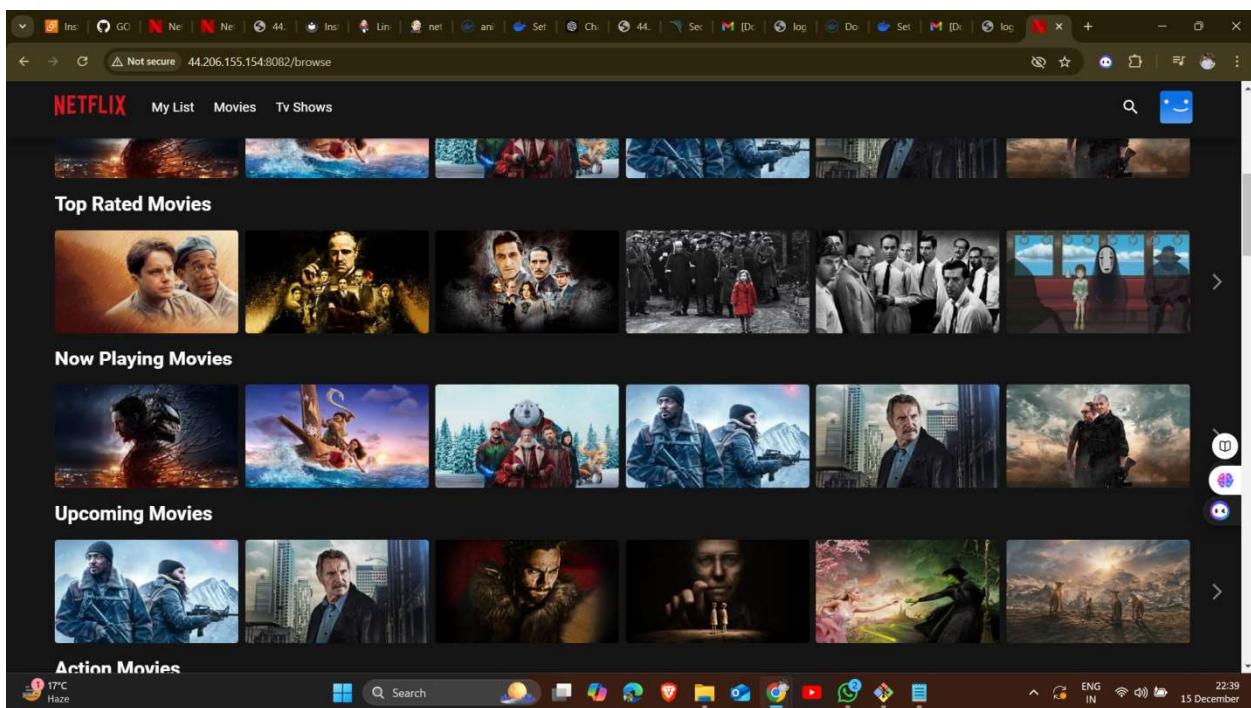
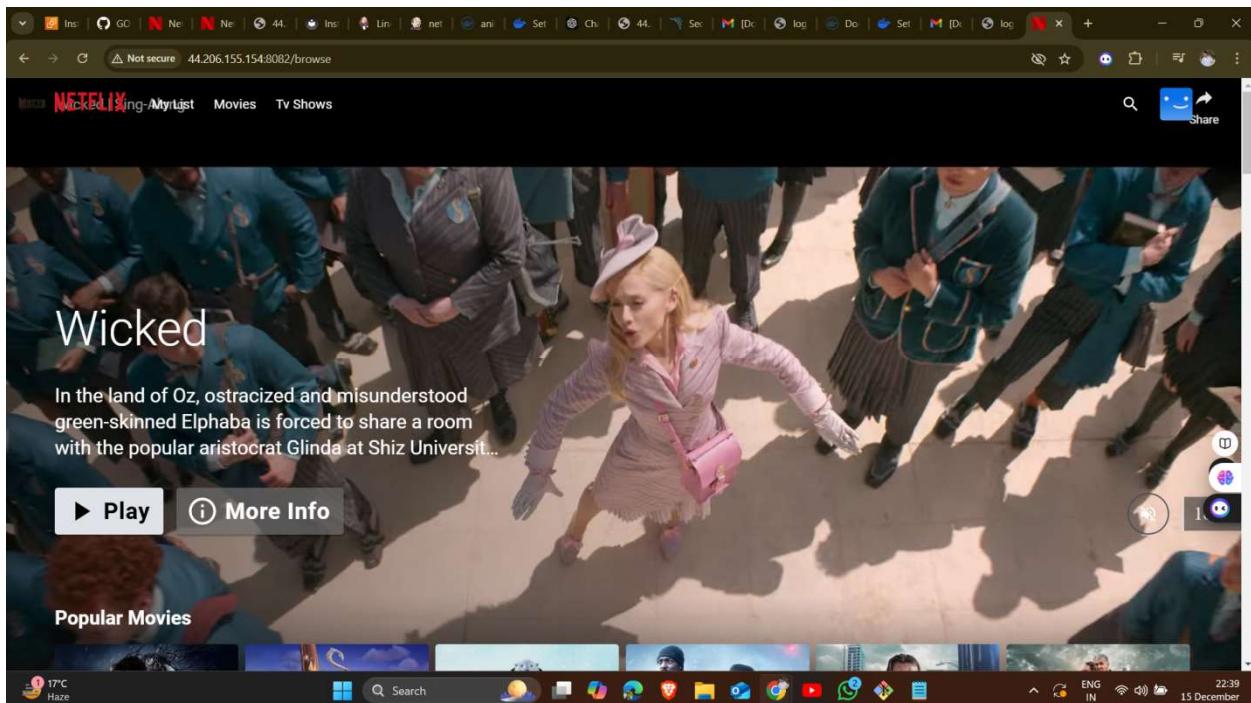


Here the stages that passed



The screenshot shows the Jenkins Pipeline Console for build #16. The left sidebar lists the stages: Tool Install, Clean Workspace, Checkout from Git, SonarQube Analysis, Quality Gate, Install Dependencies, OWASP Dependency Check, TRIVY File System Scan, Docker Build & Push, TRIVY Image Scan, and Deploy to Container. The main area shows the log output for each stage. The 'Tool Install' stage took 0.78 sec and was successful. The 'jdk17' stage used a tool from a predefined installation and took 33 ms. The 'node16' stage used a tool from a predefined installation and took 33 ms. The 'docker run -d --name netflixcont1 -p 8082:80 gollena12/netflix:latest' stage took 0.51 sec and contained the command: + docker run -d --name netflixcont1 -p 8082:80 gollena12/netflix:latest b2987e2061df4f8b7b3fed67299634c70ecf80aa23a3a3cce79ff61023315138. The Jenkins version 2.479.2 is shown at the bottom right.

Now copy the public of instance, now you can see that Netflix application is deployed.



PHASE 4: MONITORING THROUGH PROMETHEUS AND GRAFHANA

About Prometheus:

- **Prometheus** is an open-source monitoring and alerting toolkit designed for **metrics-based monitoring**.
- It collects, stores, and queries time-series data using a powerful **query language (PromQL)**.
- Key features include **multi-dimensional data collection, alerting**, and integrations with various systems.
- Commonly used for monitoring application performance, infrastructure, and services in cloud-native environments.

About Grafana:

- **Grafana** is an open-source visualization and analytics platform that allows users to create interactive **dashboards** from time-series data.
- It supports multiple data sources, including Prometheus, InfluxDB, and Elasticsearch.
- Key features include customizable visualizations like graphs, heatmaps, and tables, with real-time monitoring capabilities.
- Grafana makes complex metrics easier to interpret through intuitive dashboards.

Prometheus + Grafana Integration:

1. **Data Collection:** Prometheus scrapes metrics from monitored targets and stores them in its time-series database.
2. **Visualization:** Grafana uses Prometheus as a data source to build interactive dashboards and visualize metrics.
3. **Alerting:** Prometheus sends alerts based on defined conditions, and Grafana can display alert notifications on dashboards.

Now create a new instance using aws console and connect it with the server

```
ubuntu@ip-172-31-97-68:~  
[anil@anil-OptiPlex-2 MINNOW64 ~ (master)  
$ cd Downloads/  
[anil@anil-OptiPlex-2 MINNOW64 ~/Downloads (master)  
$ ssh -i "anil.pem" ubuntu@ec2-54-196-157-114.compute-1.amazonaws.com  
The authenticity of host 'ec2-54-196-157-114.compute-1.amazonaws.com (54.196.157.114)' can't be established.  
ED25519 key fingerprint is SHA256:x9OEUB0ehp7NqQPOThS0QUBBhhgJ4Dqidz5jt+3A.  
Are you sure you want to continue connecting (yes/no)?指纹  
warning: Permanently added 'ec2-54-196-157-114.compute-1.amazonaws.com' (ED25519  
) to the list of known hosts.  
welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
System information as of Sun Dec 15 17:16:11 UTC 2024  
System load: 0.48 Processes: 121  
Usage of /: 21.1% of 7.57GB Users logged in: 0  
Memory usage: 6% IPv4 address for eth0: 172.31.97.68  
Swap usage: 0K  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To run a command as administrator (user "root"), use "sudo <command>".  
see "man sudo_root" for details.  
ubuntu@ip-172-31-97-68:~$ sudo apt update  
Hit1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease [2  
8 kB]  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [12  
127 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [129 kB]  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packag  
es [14.1 MB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-  
Air Moderate  
Now ENG IN 13:14 16 December
```

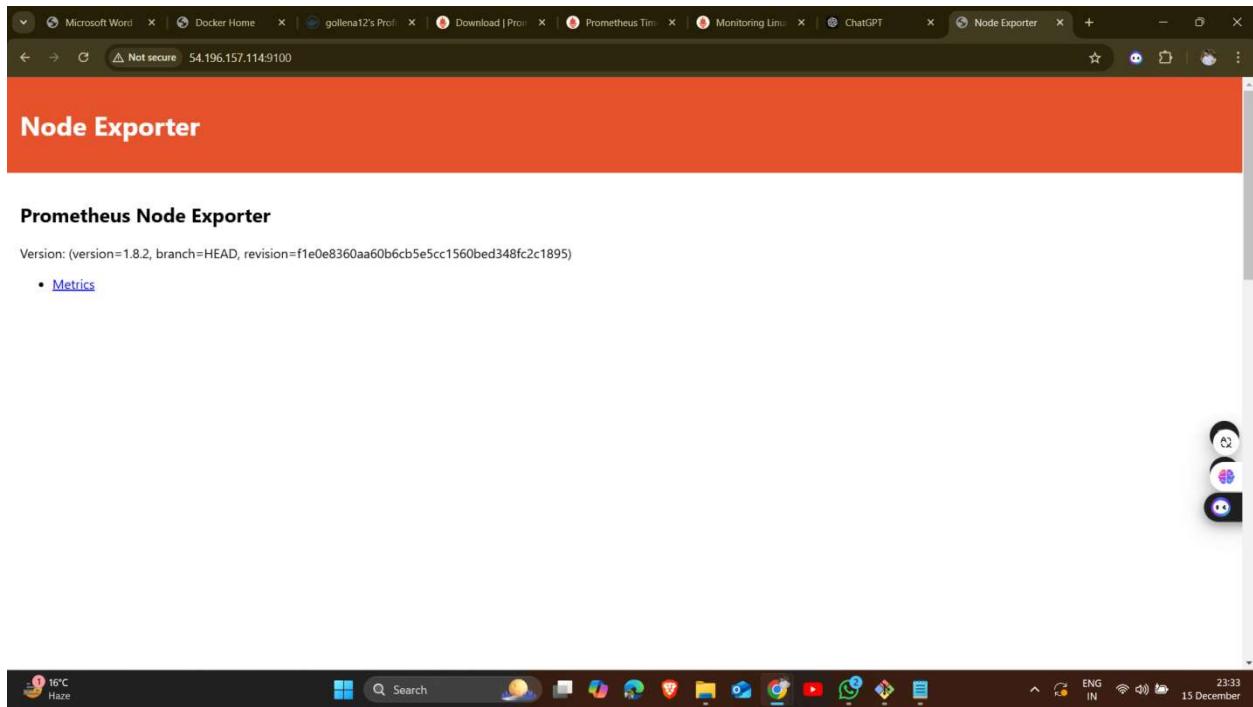
Now install Prometheus by using Prometheus official page copy the displayed commands

Now unzip the file

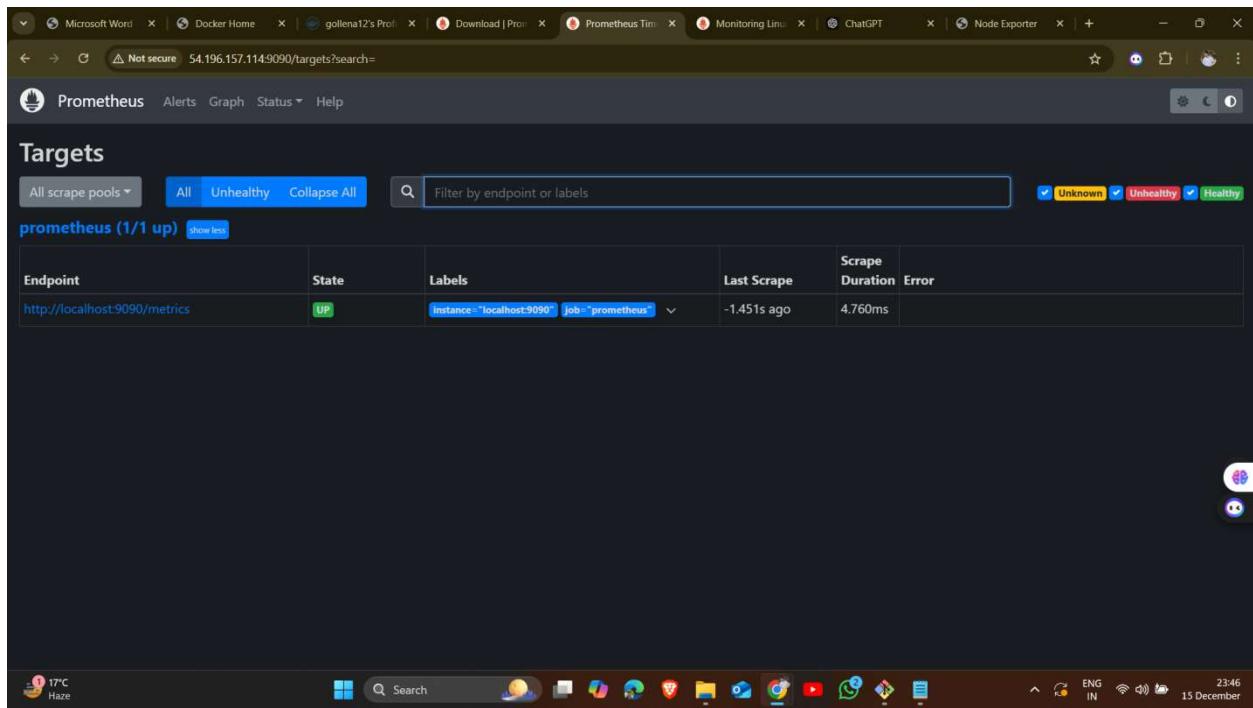
Now install node exporter from google copy the web link address

```
ubuntu@ip-172-31-97-68:~$ Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<head> op:771 level=info component=<tsdb> msg="Replaying WAL, this may take a while" Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<head> op:793 level=info component=<tsdb> msg="WAL segment loaded" segment=0 maxSegment=0 Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<head> op:830 level=info component=<tsdb> msg="WAL replay completed" checkpoint_replay_duration=41.826us wal_replay_duration=807us Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<main> op:1169 level=info msg="TSDB started" Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<main> op:1172 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<main> op:1193 level=info msg="Config loading old=100 new=75" Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<main> op:1402 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/prometheus.yml totalDuration=8ms Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<main> op:1133 level=info msg="Server is ready to receive web requests." Dec 15 15:47:02 ip-172-31-97-68 prometheus[16:G]: ts=2024-12-15T17:45:02+00:00 caller=<manager> op:164 level=info component="rule manager" msg="Starting rule manager..." ubuntu@ip-172-31-97-68:~$ sudo useradd --system --no-create-home --shell /bin/false node_exporter prometheus-2.53.3.linux-amd64.tar.gz ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/latest/download/node_exporter-<VERSION>.linux-amd64.tar.gz ubuntu@ip-172-31-97-68:~$ sudo wget http://github.com/prometheus/node_exporter/releases/latest/download/node_exporter-<VERSION>.linux-amd64.tar.gz -bash: VERSION: No such file or directory ubuntu@ip-172-31-97-68:~$ sudo apt update Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease Reading package lists... Done Reading package lists... Done Reading package lists... Done 37 packages can be upgraded. Run 'apt list --upgradable' to see them. ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/latest/download/node_exporter-<VERSION>.linux-amd64.tar.gz -bash: VERSION: No such file or directory ubuntu@ip-172-31-97-68:~$ sudo useradd --system --no-create-home --shell /bin/false node_exporter useradd: user 'node_exporter' already exists ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/latest/download/node_exporter-<VERSION>.linux-amd64.tar.gz -bash: VERSION: No such file or directory ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/download/v<VERSION>/node_exporter-<VERSION>.<OS>-<ARCH>.tar.gz tar xfzv node_exporter-<VERSION>.linux-amd64.tar.gz cd node_exporter-* & am64 ./node_exporter -bash: VERSION: No such file or directory tar: node_exporter-*: Cannot open: No such file or directory tar (child): Error is not recoverable; exiting now tar: Child returned status 2 tar: Error is not recoverable; exiting now -bash: cd: node_exporter-*: No such file or directory -bash: ./node_exporter-*: No such file or directory ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/download/v<VERSION>/node_exporter-<VERSION>.<OS>-<ARCH>.tar.gz -bash: VERSION: No such file or directory ubuntu@ip-172-31-97-68:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.8.2/node_exporter-1.8.2.linux-amd64.tar.gz -bash: VERSION: No such file or directory Resolving github.com [github.com]: 140.82.113.3 Connecting to github.com [github.com]: 140.82.113.3!443... connected. HTTP request sent, awaiting response... 302 Found Reusing connection: https://objects.githubusercontent.com/github-production/release-asset-2e65b9c5924057f7a70ef411-5543-402c-9000-26fafe32bb47x-Amz-Algorithm=AWS4-HMAC-SHA256&x-Amz-Credential=releaseassetproduction%2F20241215%2Fgithub%2Fraw%2B%2Fblob%2Fname%3Dnode-exporter-1.8.2.linux-amd64.tar.gz%2BContent-Type%3Dapplication%2Foctet-stream%2B%2Fname%3Dnode-exporter-1.8.2.linux-amd64.tar.gz%2BContent-Type%3Dapplication%2Foctet-stream [following] -2024-12-15 17:57:40X-Amz-Expires=3000X-Amz-Signature=3dd9ddff44709a74e603805abdf2afaf00794ba15b3c2ae39224f8X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20file%3Dnode-exporter-1.8.2.linux-amd64.tar.gz%2BContent-Type%3Dapplication%2Foctet-stream resolving objects.githubusercontent.com [objects.githubusercontent.com]: 185.199.100.133, 185.199.111.133, 185.199.108.133, ... ENG IN Wi-Fi 13:15
```

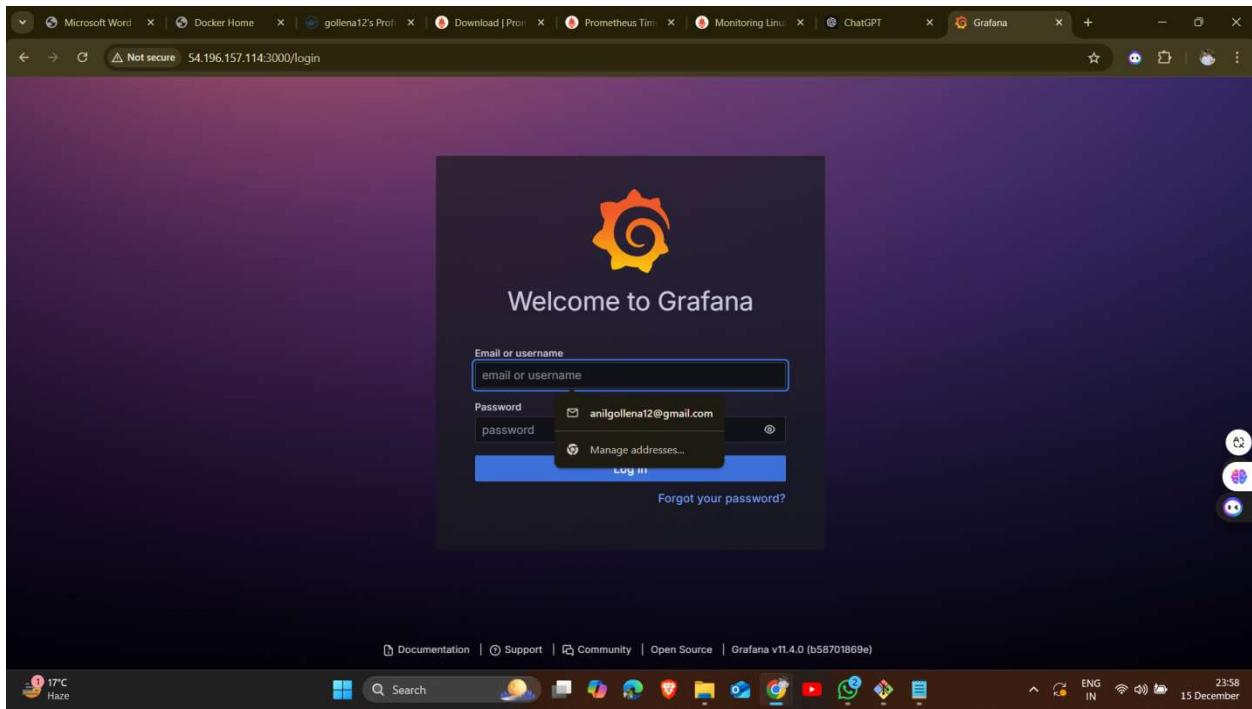
Now copy the public ip with port number 9100



Now copy the public ip with port number 9090



Copy the public ip with port number 3000



Here are the commands I used to setup these

```
ubuntu@ip-172-31-97-68:~  
*** System restart required ***  
Last login: Mon Dec 16 2024 from 49.204.100.179  
ubuntu@ip-172-31-97-68:~$ sudo vi prometheus.yaml  
ubuntu@ip-172-31-97-68:~$ history  
1 sudo apt update  
2 sudo useradd --system --no-create-home --shell /bin/false prometheus  
3 wget https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz  
4 ls  
5 tar -xvf prometheus-2.53.3.linux-amd64.tar.gz  
6 ls  
7 cd prometheus-2.53.3.linux-amd64/  
8 ls  
9 sudo mkdir -p /data/etc/prometheus  
10 ls  
11 sudo mv prometheus promtool /usr/local/bin/  
12 sudo mv prometheus promtool /usr/local/bin/  
13 sudo mkdir -p /data/etc/prometheus  
14 ls  
15 sudo mv prometheus promtool /usr/local/bin/  
16 ls  
17 cd  
18 cd prometheus-2.53.3.linux-amd64/  
19 ls  
20 cd  
21 sudo rm -rf prometheus-2.53.3.linux-amd64  
22 ls  
23 tar -xvf prometheus-2.53.3.linux-amd64.tar.gz  
24 ls  
25 cd prometheus-2.53.3.linux-amd64/  
26 ls  
27 sudo mkdir -p /data/etc/prometheus  
28 ls  
29 sudo mv prometheus promtool /usr/local/bin/  
30 ls  
31 cd /etc/  
32 ls  
33 sudo mkdir prometheus  
34 ls  
35 cd  
36 ls  
37 cd prometheus-2.53.3.linux-amd64/  
38 ls  
39 sudo mv console_libraries /etc/prometheus/  
40 ls  
41 sudo mv prometheus.yaml /etc/prometheus/prometheus.yaml  
42 ls  
43 sudo mv consoles/ console_libraries/ /etc/prometheus/  
44 sudo mv consoles /etc/prometheus/  
45 ls  
46 sudo chown -R prometheus:prometheus /etc/prometheus/ /data/  
47 ls  
48 sudo vi /etc/systemd/system/prometheus.service  
49 ls  
50 cd  
51 sudo systemctl start prometheus  
52 sudo systemctl enable prometheus  
53 sudo systemctl status prometheus  
54 sudo useradd --system --no-create-home --shell /bin/false node_exporter  
ubuntu@ip-172-31-97-68:~$ Air Moderate  
Now  
ubuntu@ip-172-31-97-68:~$ 13:13  
ubuntu@ip-172-31-97-68:~$ 16 December
```

```

ubuntu@ip-172-31-97-68:~$ 61 wget https://github.com/prometheus/node_exporter/releases/latest/download/node_exporter-<VERSION>.linux-amd64.tar.gz
ubuntu@ip-172-31-97-68:~$ 62 tar xfz node_exporter-<VERSION>.linux-amd64.tar.gz
ubuntu@ip-172-31-97-68:~$ 63 cd node_exporter
ubuntu@ip-172-31-97-68:~$ 64 ./node_exporter
ubuntu@ip-172-31-97-68:~$ 65 wget https://github.com/prometheus/node_exporter/releases/download/v<VERSION>/node_exporter-<VERSION>.<OS><ARCH>.tar.gz
ubuntu@ip-172-31-97-68:~$ 66 tar -xvf node_exporter-1.8.2.linux-amd64.tar.gz
ubuntu@ip-172-31-97-68:~$ 67 ls
ubuntu@ip-172-31-97-68:~$ 68 sudo mv node_exporter-1.8.2.linux-amd64/node_exporter /usr/local/bin/
ubuntu@ip-172-31-97-68:~$ 69 ls
ubuntu@ip-172-31-97-68:~$ 70 rm -rf node_exporter
ubuntu@ip-172-31-97-68:~$ 71 ls
ubuntu@ip-172-31-97-68:~$ 72 sudo vi /etc/systemd/system/node_exporter.service
ubuntu@ip-172-31-97-68:~$ 73 sudo systemctl start node_exporter
ubuntu@ip-172-31-97-68:~$ 74 sudo systemctl enable node_exporter
ubuntu@ip-172-31-97-68:~$ 75 sudo systemctl status node_exporter
ubuntu@ip-172-31-97-68:~$ 76 cd /etc/prometheus/
ubuntu@ip-172-31-97-68:~$ 77 ls
ubuntu@ip-172-31-97-68:~$ 78 sudo vi prometheus.yml
ubuntu@ip-172-31-97-68:~$ 79 sudo apt update
ubuntu@ip-172-31-97-68:~$ 80 sudo apt-get install -y apt-transport-https software-properties-common wget gpg
ubuntu@ip-172-31-97-68:~$ 81 cd /etc/apt
ubuntu@ip-172-31-97-68:~$ 82 sudo apt-get install -y apt-transport-https software-properties-common wget gpg
ubuntu@ip-172-31-97-68:~$ 83 ls
ubuntu@ip-172-31-97-68:~$ 84 cd /etc/prometheus/
ubuntu@ip-172-31-97-68:~$ 85 sudo apt-get install -y apt-transport-https software-properties-common wget gpg
ubuntu@ip-172-31-97-68:~$ 86 ls
ubuntu@ip-172-31-97-68:~$ 87 cd /etc/prometheus/
ubuntu@ip-172-31-97-68:~$ 88 ls
ubuntu@ip-172-31-97-68:~$ 89 ls
ubuntu@ip-172-31-97-68:~$ 90 sudo apt-get install -y apt-transport-https software-properties-common wget gpg
ubuntu@ip-172-31-97-68:~$ 91 sudo mkdir -p /etc/apt/keyrings/
ubuntu@ip-172-31-97-68:~$ 92 wget -q https://apt.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg
ubuntu@ip-172-31-97-68:~$ 93 sudo chmod 440 /etc/apt/keyrings/grafana.gpg
ubuntu@ip-172-31-97-68:~$ 94 wget -q https://apt.grafana.com/gpg_key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg
ubuntu@ip-172-31-97-68:~$ 95 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee /etc/apt/sources.list.d/grafana.list
ubuntu@ip-172-31-97-68:~$ 96 echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com beta main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
ubuntu@ip-172-31-97-68:~$ 97 sudo apt-get update
ubuntu@ip-172-31-97-68:~$ 98 sudo apt-get install -y grafana
ubuntu@ip-172-31-97-68:~$ 99 sudo systemctl enable grafana-server
ubuntu@ip-172-31-97-68:~$ 100 sudo systemctl start grafana-server
ubuntu@ip-172-31-97-68:~$ 101 sudo systemctl status grafana-server
ubuntu@ip-172-31-97-68:~$ 102 history
ubuntu@ip-172-31-97-68:~$ 103 sudo systemctl restart grafana-server
ubuntu@ip-172-31-97-68:~$ 104 sudo systemctl restart prometheus
ubuntu@ip-172-31-97-68:~$ 105 sudo systemctl status prometheus
ubuntu@ip-172-31-97-68:~$ 106 sudo systemctl enable prometheus
ubuntu@ip-172-31-97-68:~$ 107 sudo systemctl status prometheus
ubuntu@ip-172-31-97-68:~$ 108 ls
ubuntu@ip-172-31-97-68:~$ 109 cd prometheus-2.53.3.linux-amd64/
ubuntu@ip-172-31-97-68:~$ 110 sudo systemctl start prometheus
ubuntu@ip-172-31-97-68:~$ 111 sudo systemctl status prometheus
ubuntu@ip-172-31-97-68:~$ 112 sudo systemctl restart prometheus
ubuntu@ip-172-31-97-68:~$ 113 sudo systemctl status prometheus
ubuntu@ip-172-31-97-68:~$ 114 sudo vi prometheus.yml
ubuntu@ip-172-31-97-68:~$ 115 history
ubuntu@ip-172-31-97-68:~$ sudo vi /etc/systemd/system/prometheus.service
ubuntu@ip-172-31-97-68:~$ sudo vi /etc/systemd/system/node_exporter.service
ubuntu@ip-172-31-97-68:~$ |

```

And configure Prometheus and Grafana files

Node exporter configuration file.

```
ubuntu@ip-172-31-97-68: ~
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target
StartLimitIntervalSec=500
StartLimitBursts=5

[Service]
Exec=/usr/local/bin/node_exporter --collector.loginid
RestartSec=5s
Type=simple
Restart=on-failure
User=root
Group=root
Environment="NODE_EXPORTER_COLLECTOR_LOGINID=loginid"

[Install]
WantedBy=multi-user.target

```

```
ubuntu@ip-172-31-97-68:/etc/prometheus
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: [ 'localhost:9090' ]
    - job_name: 'node_exporter'

    static_configs:
      - targets: [ 'localhost:9100' ]
    - job_name: 'jenkins'

    metrics_path: '/prometheus'
    static_configs:
      - targets: [ '44.206.155.154:8080' ]

-- INSERT --
```

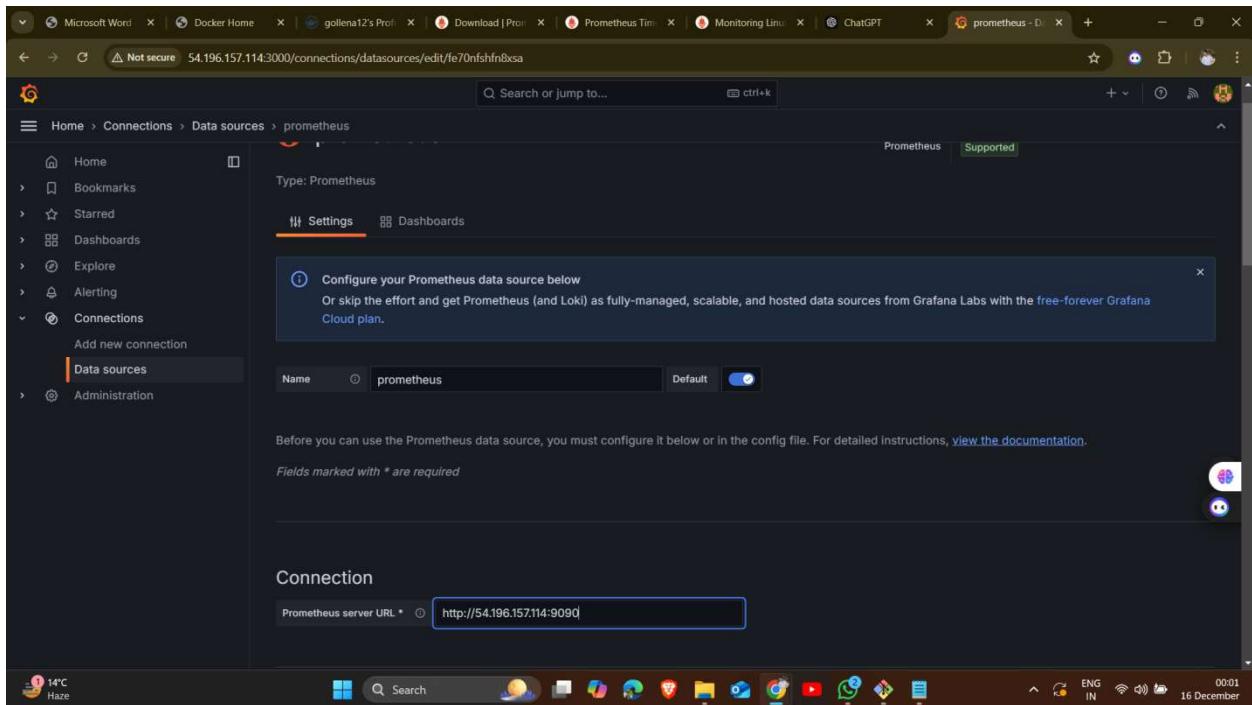
Now go to graphana page add some configuration

Now set up the Prometheus in Grafana for visualize and analyze data from Prometheus.

The screenshot shows the Grafana homepage with a dark theme. On the left, there's a sidebar with links like Home, Bookmarks, Starred, Dashboards, Explore, Alerting, Connections, and Administration. The main area has a "Welcome to Grafana" header and a "Basic" section with a "TUTORIAL" card about Grafana fundamentals and a "DATA SOURCES" card for adding a first data source. Below these are sections for Dashboards, Starred dashboards, and Recently viewed dashboards. A "Latest from the blog" section features a post about Grafana Labs in 2024. The bottom navigation bar includes icons for search, refresh, and various applications, along with system status indicators like battery level and date/time.

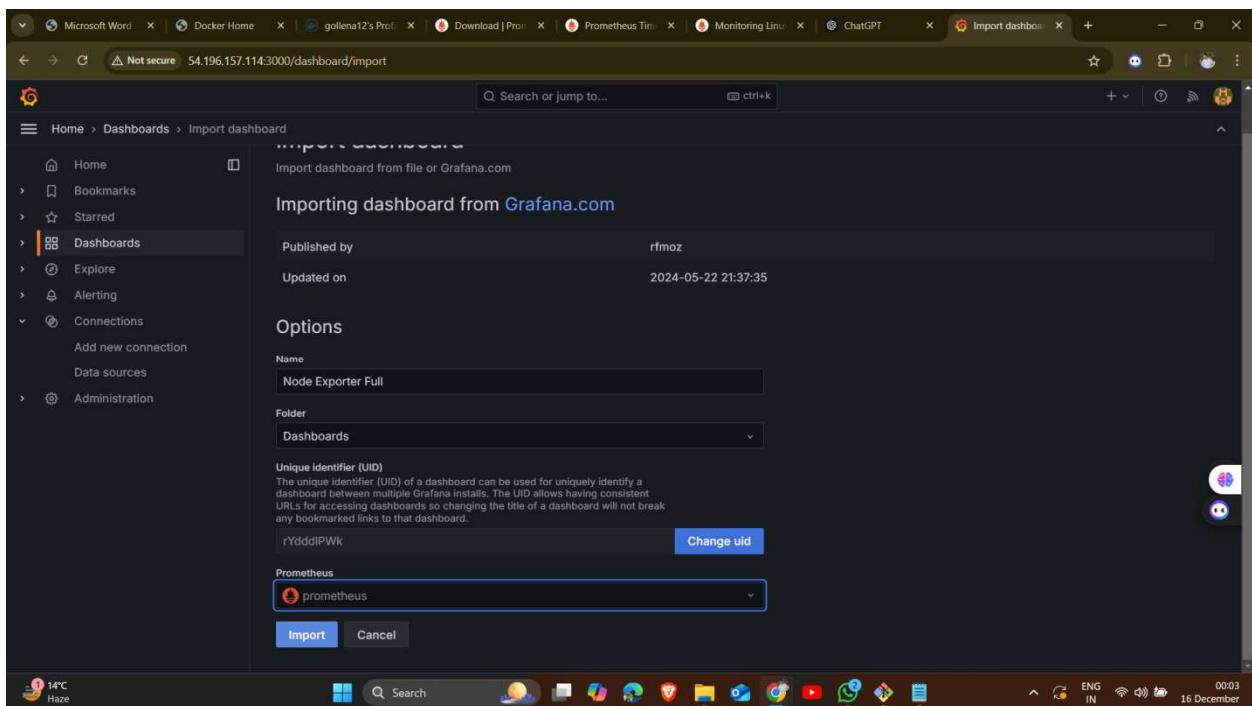
The screenshot shows the "Data sources" configuration page for Prometheus. The left sidebar shows the current path: Home > Connections > Data sources > prometheus. The main panel displays the "prometheus" data source settings. It shows the type as "Prometheus" and "Supported". The "Settings" tab is active, with a note about configuring the data source. Below it, a "Name" field is set to "prometheus" with a "Default" button. A note at the bottom says: "Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, view the documentation." The bottom navigation bar is similar to the previous screenshot, showing various application icons and system status.

Add localhost

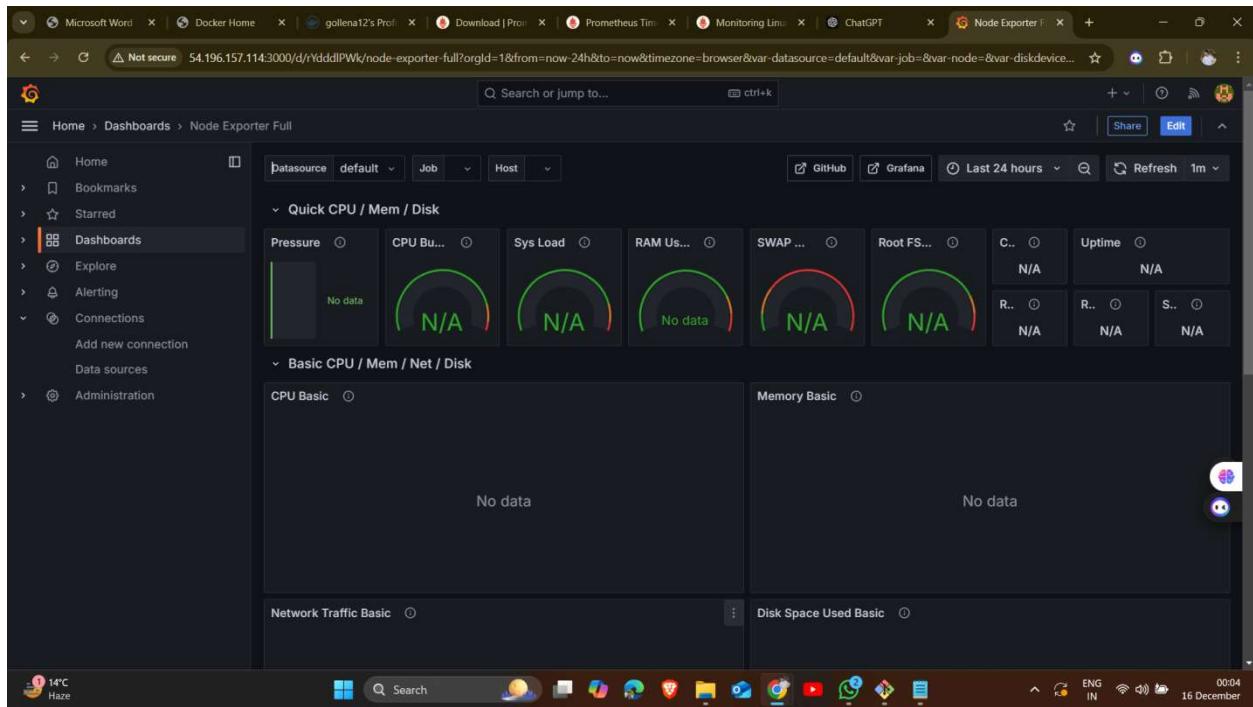


And save and apply the changes.

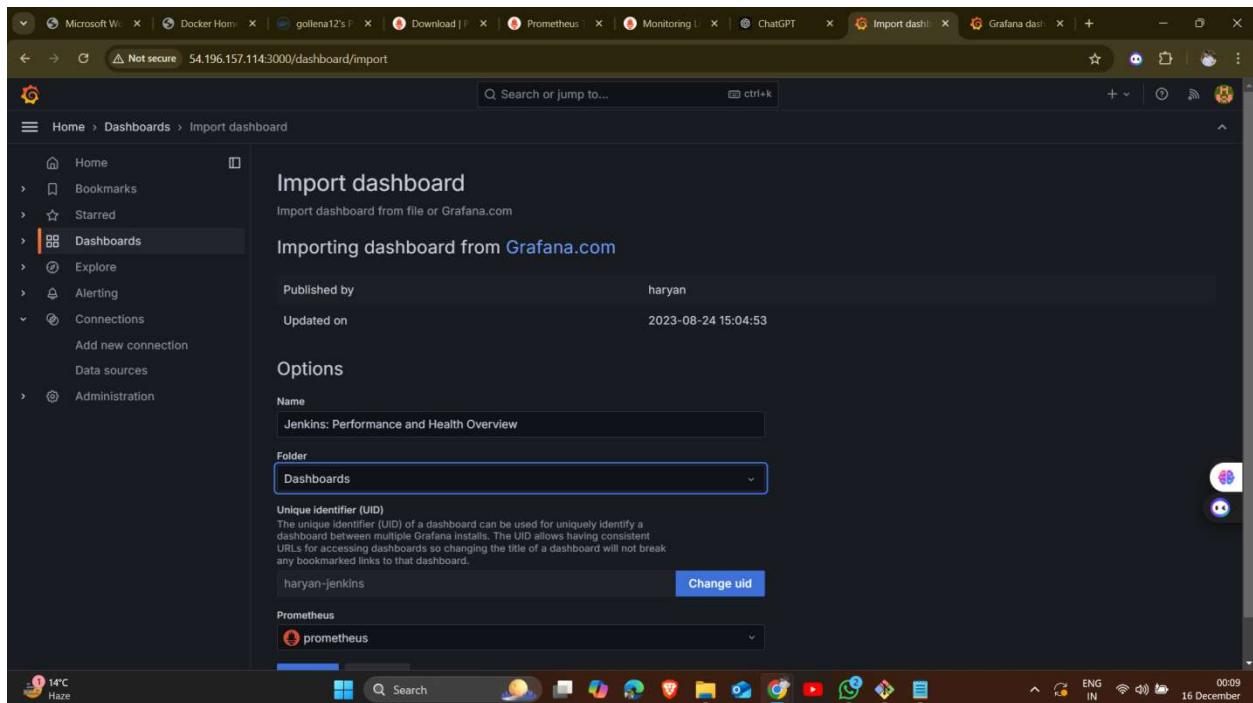
Now under dashboards → import dashboard



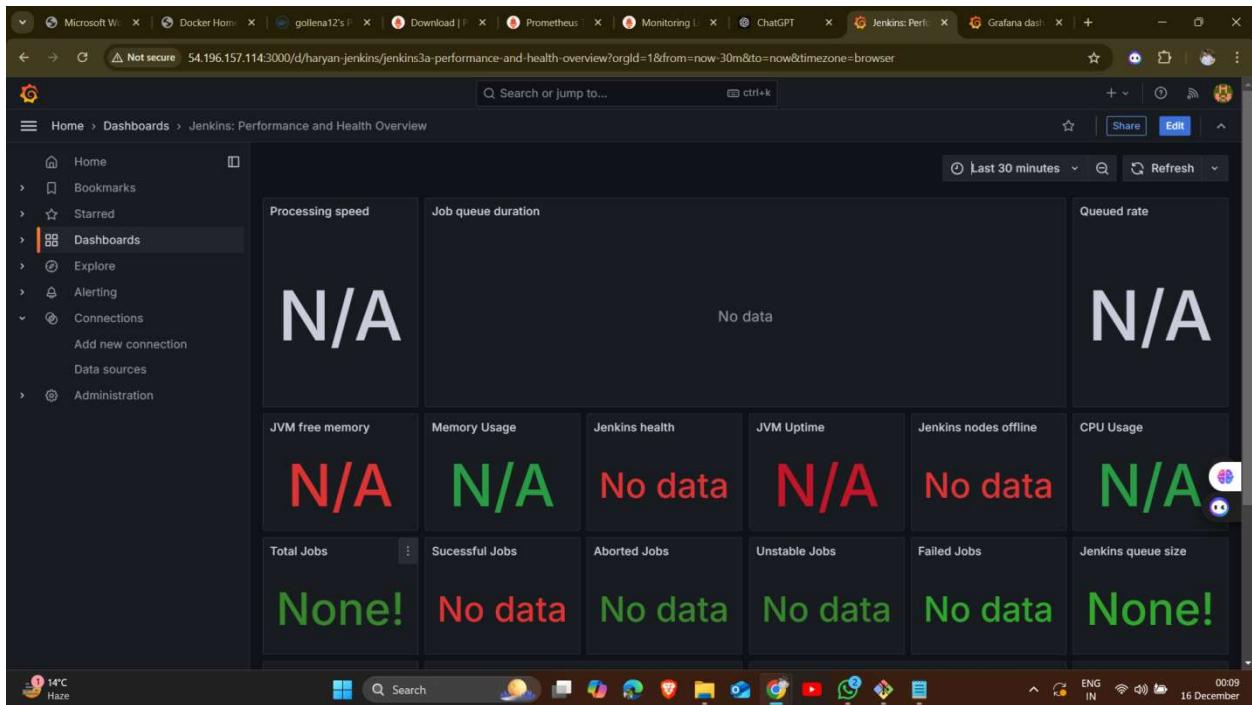
Now you can see the changes are made in node exporter



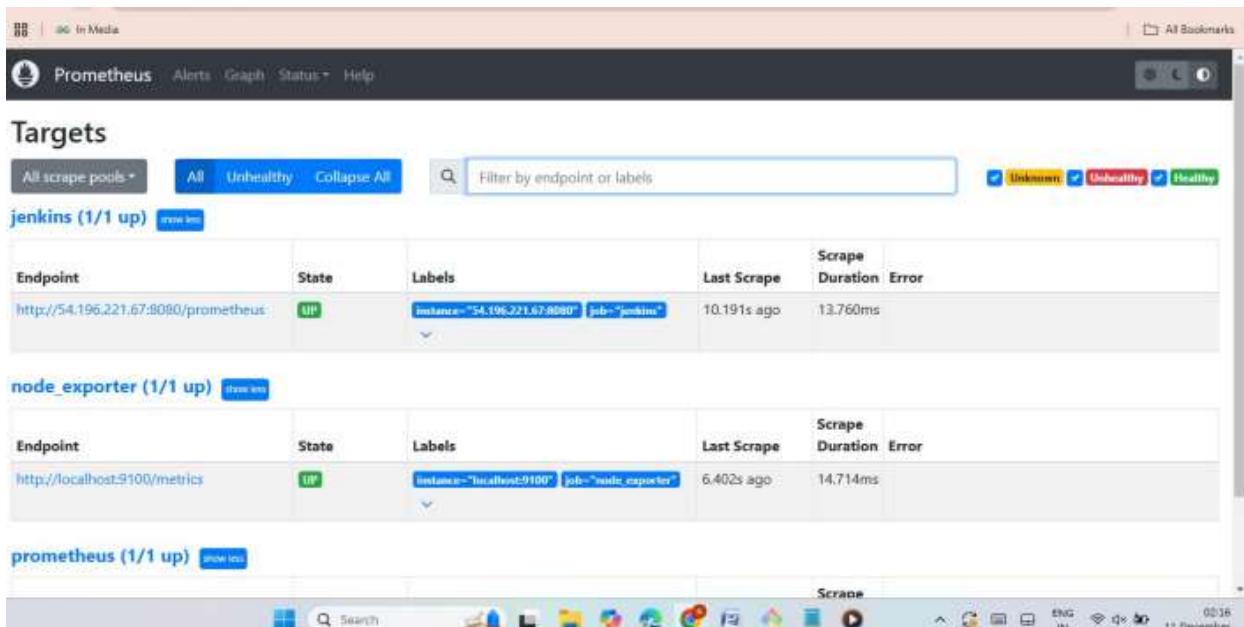
Now again import the Jenkins dashboard



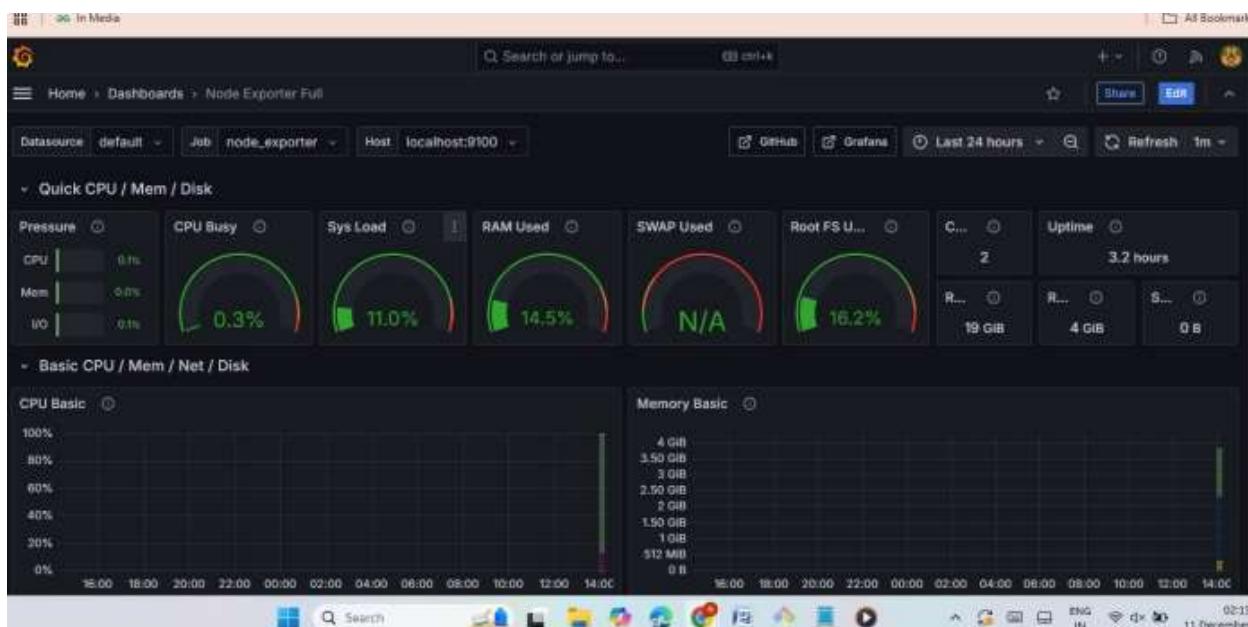
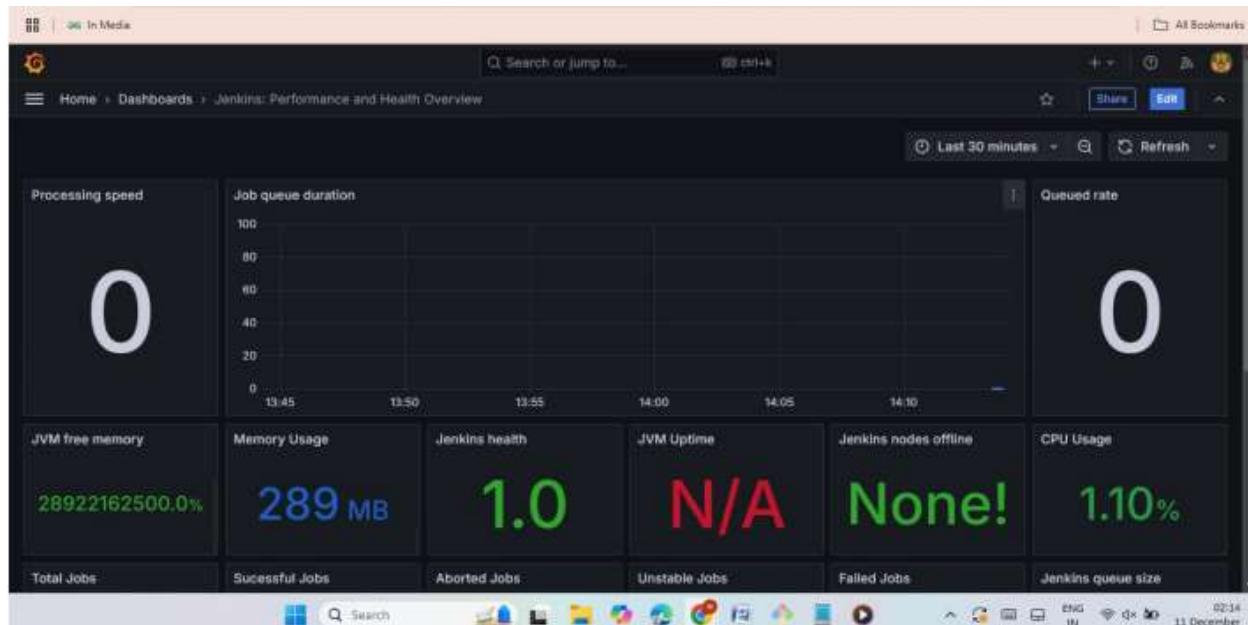
Now you can see the changes made by the Jenkins



- Now check the plugins in Jenkins.
- Install Prometheus metrics then restart the Jenkins.
- Restart the Prometheus.yaml file.



Here we can see the Jenkins performance and health overview in Grafana.



PHASE 5: DEPLOY NETFLIX APPLICATION WITH KUBERNETES EKS AND ARGOCD.

Kubernetes EKS:

- **Amazon Elastic Kubernetes Service (EKS)** is a **managed Kubernetes service** by AWS.
- Simplifies the deployment, scaling, and management of Kubernetes clusters in the cloud.
- Key features:
 - **Highly Available:** EKS manages the Kubernetes control plane across multiple availability zones.
 - **Secure:** Integrates with AWS Identity and Access Management (IAM) for access control.
 - **Scalable:** Easily scales worker nodes and applications as demand increases.
 - **Integrations:** Works seamlessly with AWS services like EC2, Fargate, and CloudWatch.

ArgoCD:

- **ArgoCD** is a declarative, GitOps-based **continuous delivery tool** for Kubernetes.
- Automates application deployment and lifecycle management in Kubernetes clusters.
- Key features:
 - **GitOps:** Uses Git repositories as the source of truth for application definitions and deployments.
 - **Sync and Drift Detection:** Continuously monitors the cluster state and identifies deviations from the desired state.
 - **Multi-Cluster Management:** Supports deploying applications to multiple Kubernetes clusters.
 - **UI and CLI:** Offers an intuitive web UI and CLI for managing deployments.

Kubernetes EKS + ArgoCD Integration:

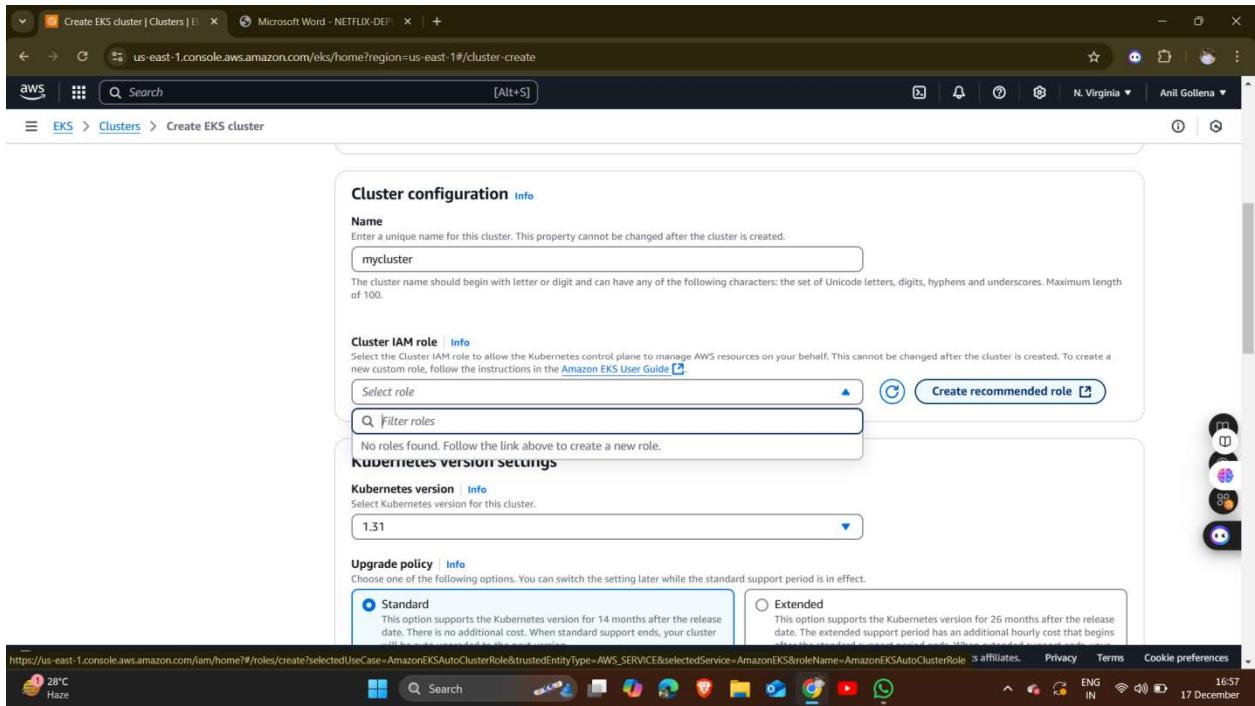
1. **Managed Clusters:** Use EKS for running Kubernetes clusters with minimal operational overhead.
2. **GitOps Workflow:** ArgoCD pulls deployment manifests from Git repositories and applies them to EKS clusters.
3. **Continuous Deployment:** Automate the deployment of new versions or configurations directly from Git.
4. **Real-Time Monitoring:** ArgoCD ensures EKS clusters match the desired state defined in Git.

Login to aws management console create a EKS cluster and node group for that here is the steps to create cluster.

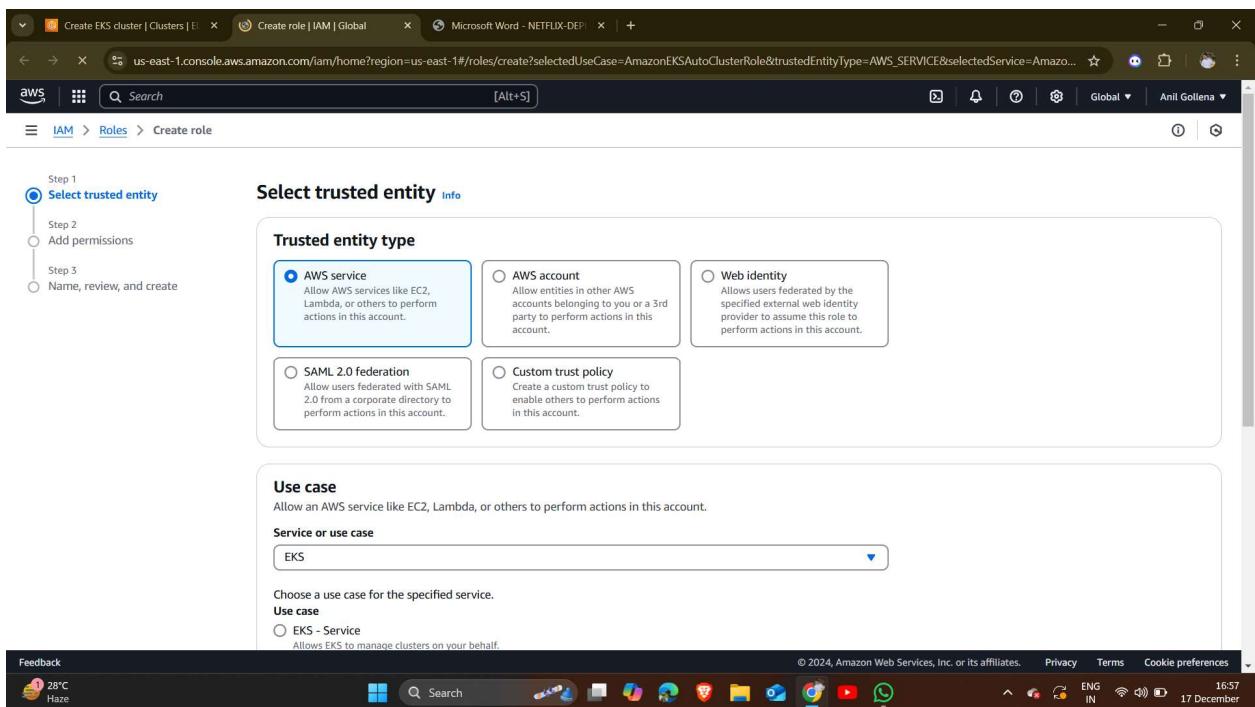
The screenshot shows the 'Configure cluster' step of the EKS cluster creation wizard. It offers two configuration options: 'Quick configuration (with EKS Auto Mode) - new' and 'Custom configuration'. The 'Custom configuration' option is selected. Below this, there's a 'Cluster configuration' section where the cluster name is set to 'attractive-funk-party' and the Kubernetes version is set to '1.31'. A 'Cluster IAM role' dropdown is also present. The browser interface includes standard navigation, search, and AWS-specific icons like CloudWatch and Lambda.

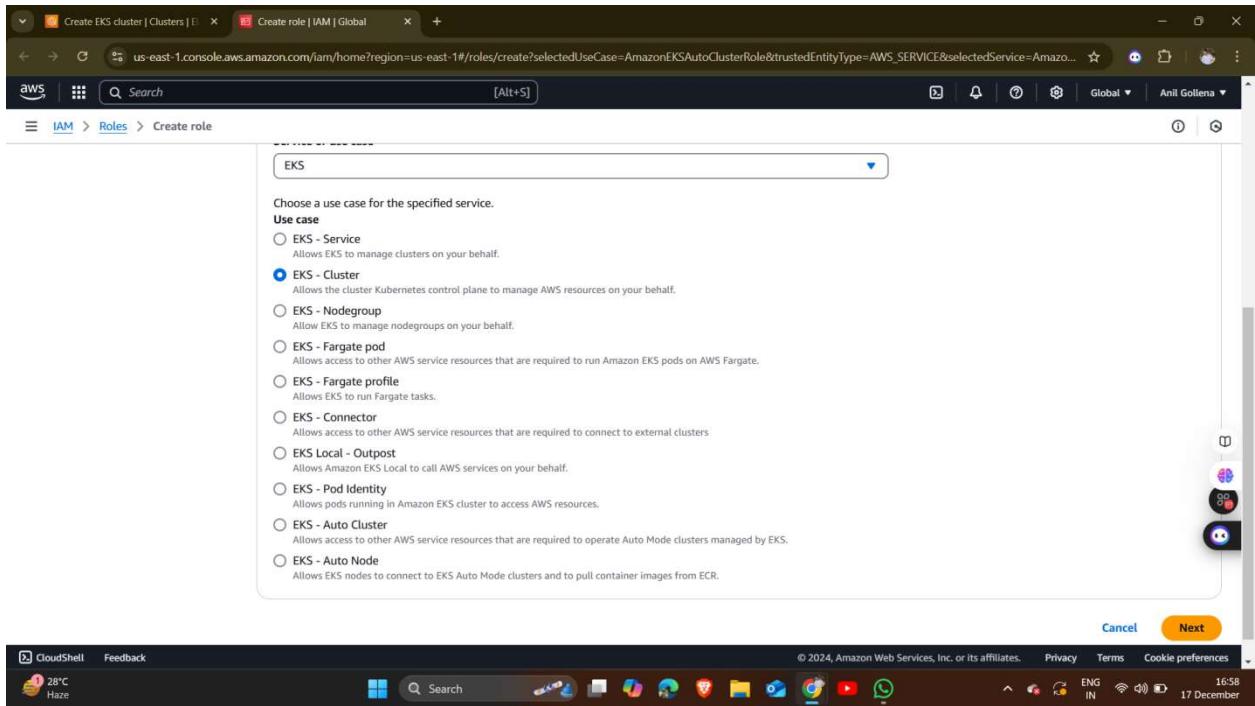
Select custom configuration.

This screenshot is identical to the one above, but it highlights the 'Custom configuration' option in the 'Configuration options' section. The rest of the interface, including the cluster configuration details and the sidebar navigation, remains the same.

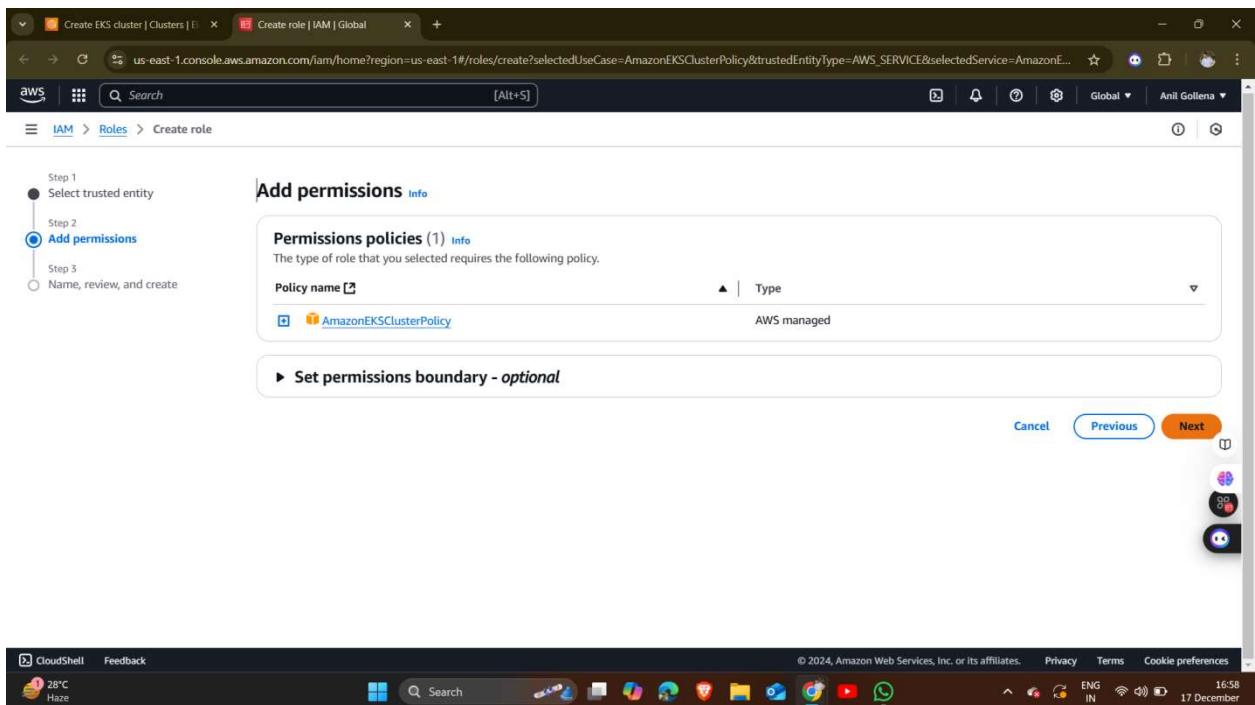


Now give cluster name and create IAM role for that click create recommended role.





Now click on next.



Now give role name.

Step 1

- Select trusted entity
- Add permissions
- Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and "-_.@_" characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=., @-[]{}#\$%^&`~`

Step 1: Select trusted entities

Trust policy

```
1+ [ {  
2+     "Version": "2012-10-17",  
3+     "Statement": [  
4+         {  
5+             "Effect": "Allow",  
6+             "Principal": {  
7+                 "Service": [  
8+                     "eks.amazonaws.com"  
9+                 ]  
10+            ]  
11+        }  
12+    ]  
13+ }]  
14 ]
```

Edit

Create role

Step 2: Add permissions

Policy name	Type	Attached as
AmazonEKSClusterPolicy	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Create role

Now create another role for EC2

The screenshot shows the 'Create role' wizard on the AWS IAM console. The current step is 'Use case'. The 'Service or use case' dropdown is set to 'EC2'. The 'Use case' section contains several options:

- EC2**: Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**: Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**: Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**: Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**: Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**: Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**: Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**: Allows EC2 Scheduled Instances to manage instances on your behalf.

Feedback bar at the bottom: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences. ZIM - AFG Live. 17:20 17 December.

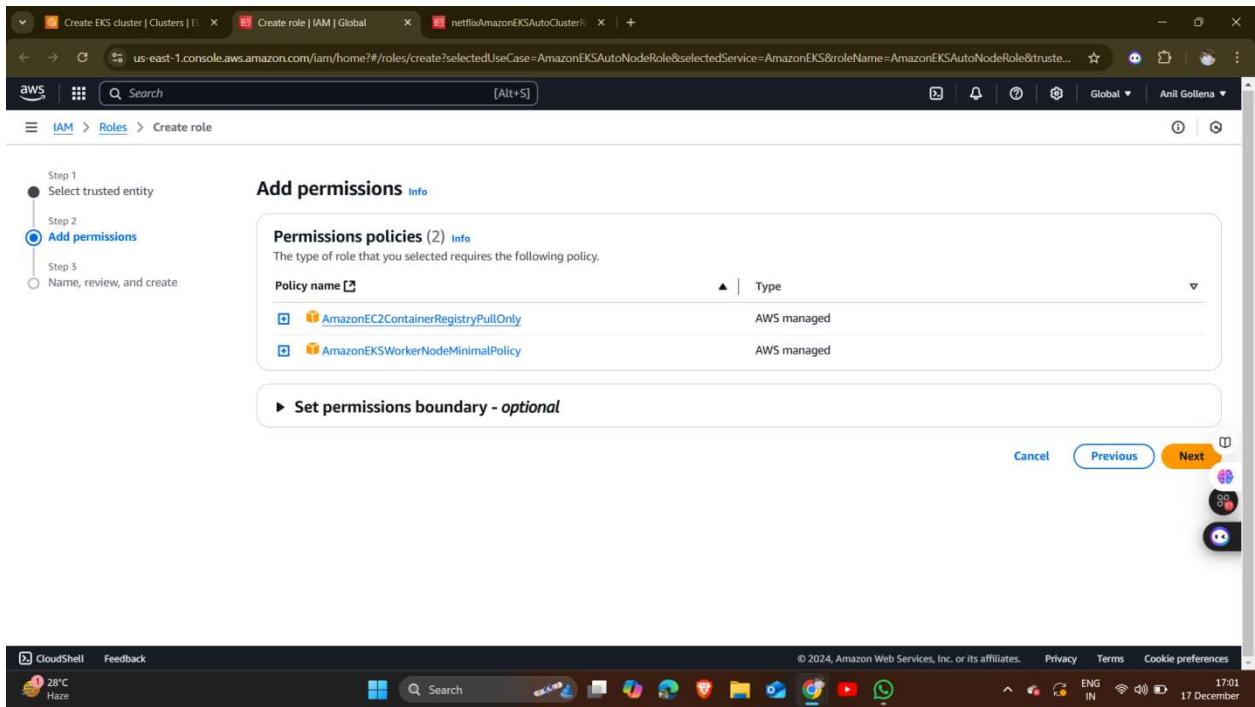
The screenshot shows the 'Create role' wizard on the AWS IAM console. The current step is 'Select trusted entity'. The 'Trusted entity type' section contains five options:

- AWS service**: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**: Create a custom trust policy to enable others to perform actions in this account.

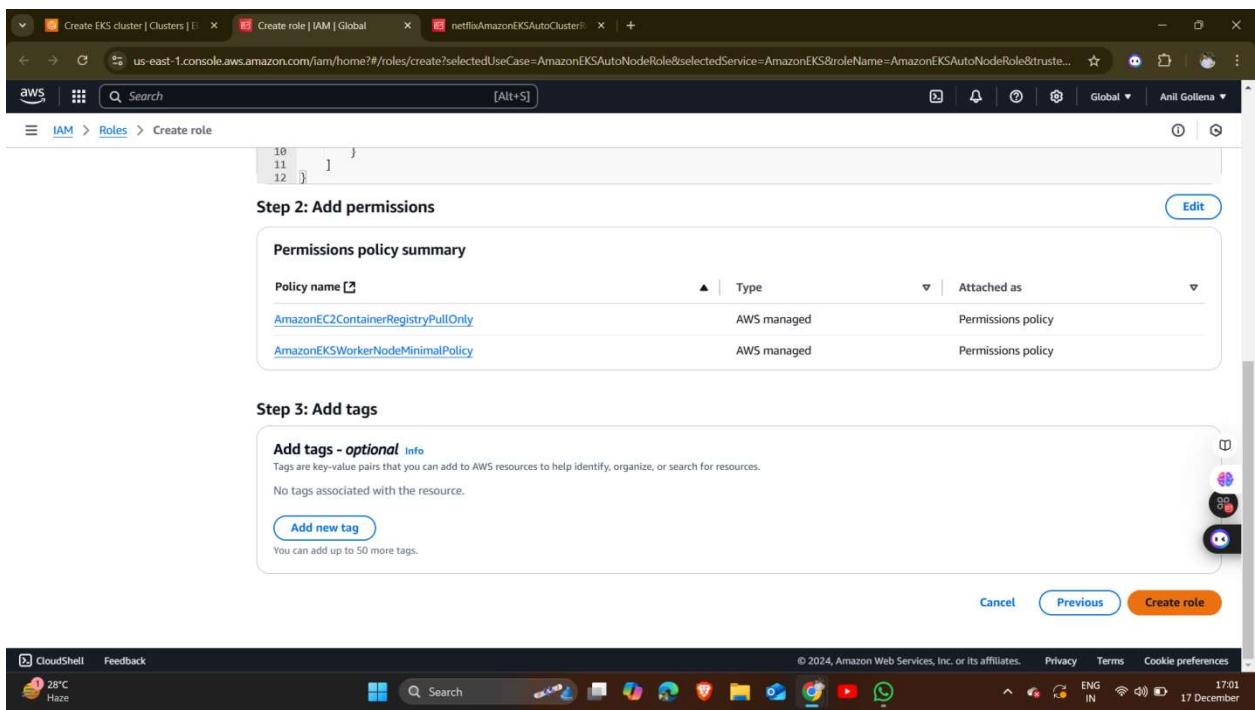
The 'Use case' section is identical to the previous screenshot, showing the 'EC2' service selected. The 'Service or use case' dropdown is set to 'EKS'.

Feedback bar at the bottom: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences. 28°C Haze. 17:00 17 December.

Now select ec2

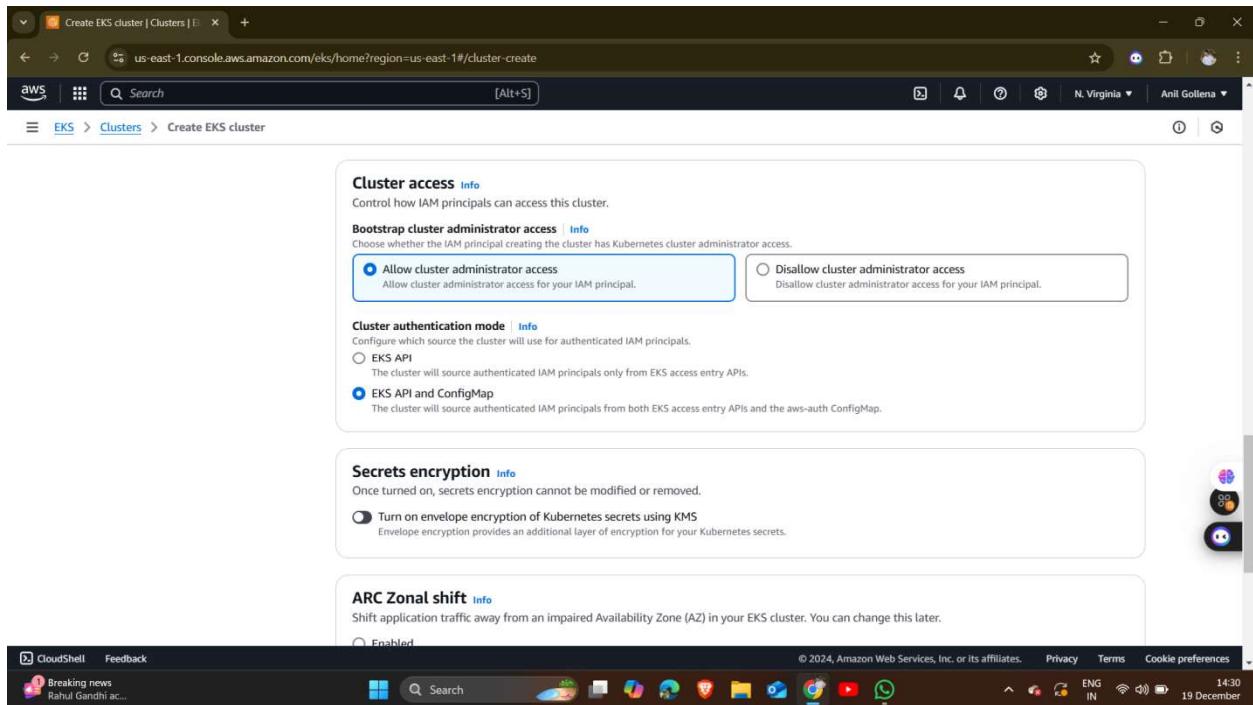


Create role

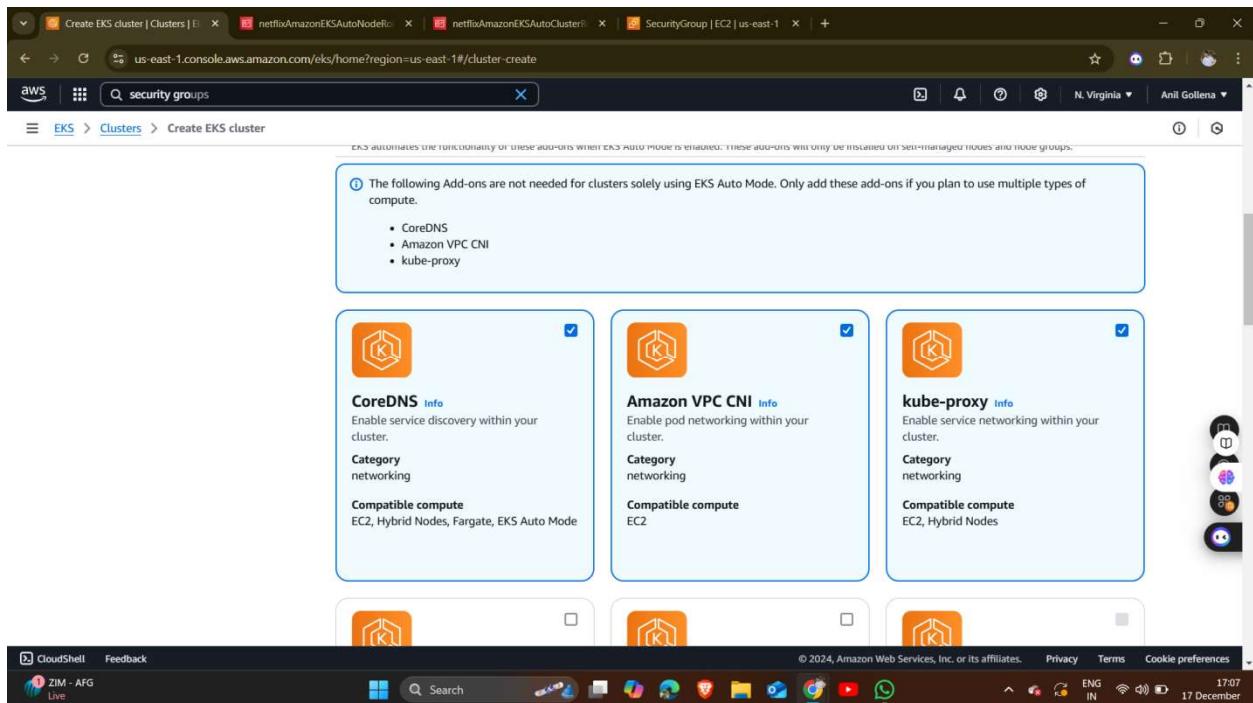


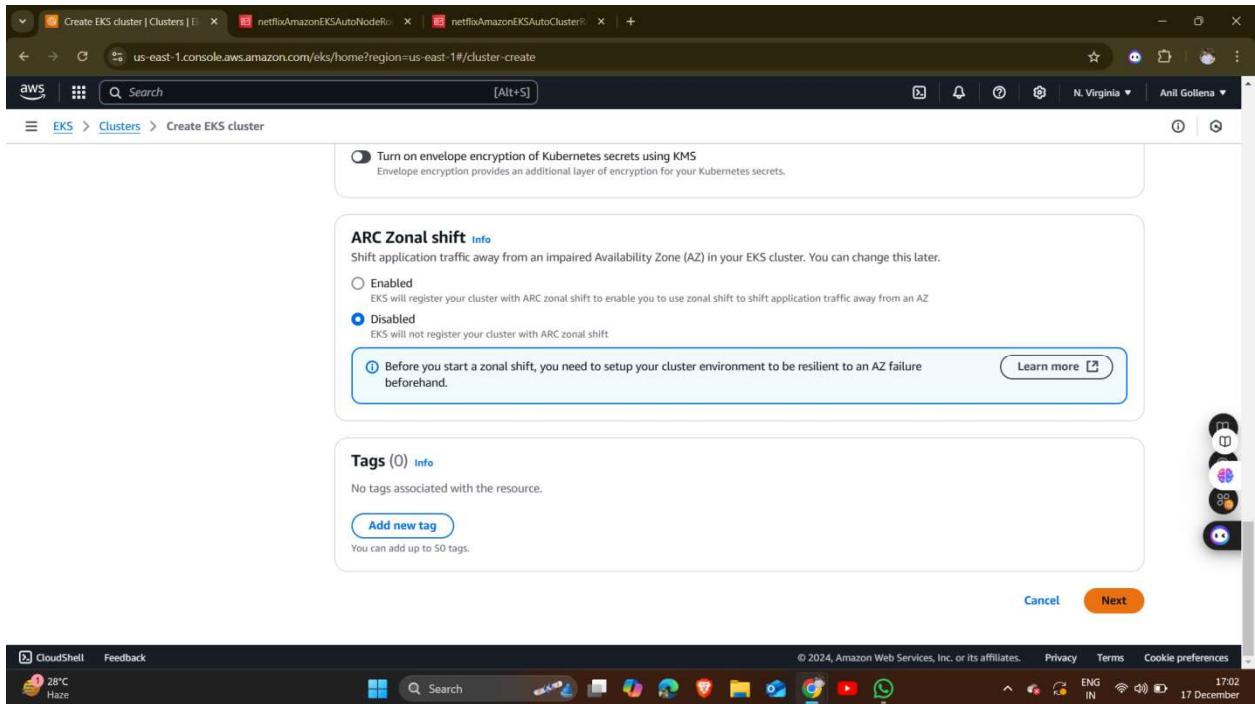
Now refresh the page so that, the roles are redirected to role place

Now give the cluster access

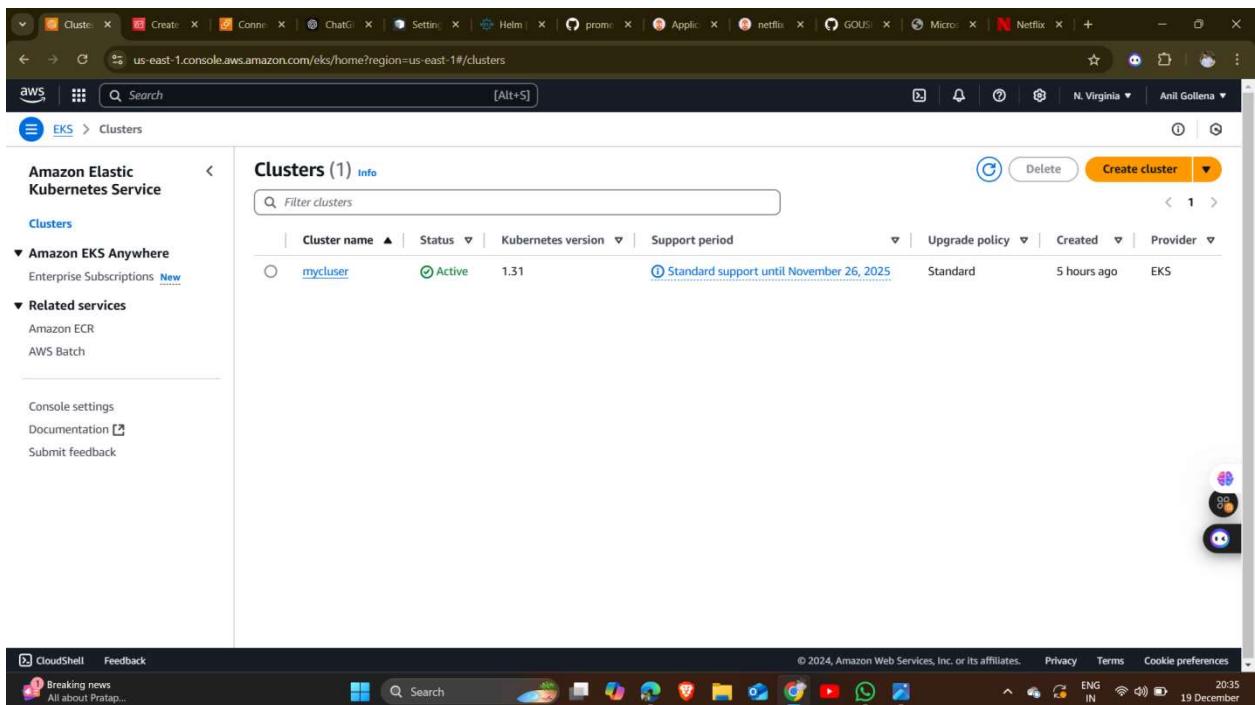


In the additional configuration select these





It may take few minutes to create cluster.



Now the cluster is in active state now create node group.

The screenshot shows the AWS EKS Node Groups creation wizard. The current step is "Node group configuration". The sidebar on the left lists steps: Step 1 (Configure node group), Step 2 (Set compute and scaling configuration, which is selected), Step 3 (Specify networking), and Step 4 (Review and create). The main area shows a "Name" field with "munode" entered. Below it is an "IAM role" dropdown set to "netflix" with a note: "The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion." A "Create recommended role" button is also present. A "Launch template" section follows, with a note: "These properties cannot be changed after the node group is created." It has a radio button for "Use launch template" which is selected, with the sub-note: "Configure this node group using an EC2 launch template."

Now create role and attach it with the node.

The screenshot shows the "Set compute and scaling configuration" step of the wizard. The sidebar still shows Step 2 is selected. The main area starts with a "Node group compute configuration" section with a note: "These properties cannot be changed after the node group is created." It includes an "AMI type" dropdown set to "Amazon Linux 2 (AL2_x86_64)", a "Capacity type" dropdown set to "On-Demand", and an "Instance types" dropdown with "t3.medium" selected. Below these are sections for "Disk size" (set to 20 GiB) and "Node group scaling configuration".

Now add subnets except us-east-1e

The screenshot shows the "Add node group" wizard in the AWS EKS console. The current step is "Specify networking".

Step 1: Configure node group

Step 2: Set compute and scaling configuration

Step 3: Specify networking (highlighted)

Step 4: Review and create

Specify networking

Node group network configuration
These properties cannot be changed after the node group is created.

Select subnets (dropdown menu)
Clear selected subnets

- subnet-0e49b73f02488d3e6 | RDS-Pvt-subnet-3
us-east-1a 172.31.97.0/25
- subnet-0cf4fb5b7422f0510c | RDS-Pvt-subnet-6
us-east-1d 172.31.98.128/25
- subnet-0412311cd4bb93e21 | RDS-Pvt-subnet-5
us-east-1b 172.31.98.0/25
- subnet-0cc00131861522b0d | RDS-Pvt-subnet-4
us-east-1f 172.31.97.128/25
- subnet-06e42b1af374394c4 | RDS-Pvt-subnet-1
us-east-1c 172.31.96.0/25

Configure remote access to nodes [Info](#)

Cancel Previous Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 17:22 ENG IN 17 December

The screenshot shows the "Add node group" wizard in the AWS EKS console. The current step is "Step 3: Networking".

Maximum unavailable
1 node

Node auto repair configuration
Node auto repair
Disabled

Step 3: Networking

Edit

Node group network configuration

Subnets
subnet-0e49b73f02488d3e6
subnet-0cf4fb5b7422f0510c
subnet-0412311cd4bb93e21
subnet-0cc00131861522b0d
subnet-06e42b1af374394c4

Configure remote access to nodes
off

Cancel Previous Create

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 17:23 ENG IN 17 December

Now node group is in active state.

The screenshot shows the AWS EKS Node Groups configuration page. The node group 'nodegroup' is listed with the following details:

- Node group configuration:**
 - Kubernetes version: 1.31
 - AMI type: AL2_x86_64
 - Status: Active
 - AMI release version: 1.31.3-20241213
 - Instance types: t3.medium
 - Disk size: 20 GB
- Details:**
 - Node group ARN: arn:aws:eks:us-east-1:905418085021:nodegroup/mycluster/nodegroup/dec9ef3a-1355-360b-47f5-5b5435368e8b
 - Autoscaling group name: eks-nodegroup-dec9ef3a-1355-360b-47f5-5b5435368e8b
 - Capacity type: On-Demand
 - Created: 2 minutes ago
 - Desired size: 2 nodes
 - Minimum size: 2 nodes
 - Maximum size: 2 nodes
 - Subnets: subnets (multiple listed)
 - Configure remote access to nodes: off

Here new two instance are created.

The screenshot shows the AWS EC2 Instances page. The table lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
EKS	i-04cda0219835bc562	Running	t2.medium	Initializing	View alarms	us-east-1a	ec2-18-212-
1	i-0dadacfb89ca61012	Running	t3.medium	3/3 checks passed	View alarms	us-east-1f	ec2-3-238-7
2	i-0384ea638c4e31763	Running	t3.medium	Initializing	View alarms	us-east-1b	ec2-44-200-

Details for instance i-04cda0219835bc562 (EKS):

- Public IPv4 address:** 18.212.122.154 | open address
- Private IP4 addresses:** 172.31.97.13
- Public IPv4 DNS:** ec2-18-212-122-154.compute-1.amazonaws.com | open address

Now back to ec2 dashboard create a new instance

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Name and tags' section has 'Name' set to 'EKS'. The 'Application and OS Images (Amazon Machine Image)' section shows a search bar and a 'Quick Start' tab selected, displaying options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and others. The 'Summary' panel on the right shows 1 instance selected, with 'Software Image (AMI)' set to t2.micro and 'Virtual server type (instance type)' also set to t2.micro. The 'Launch instance' button is highlighted in orange at the bottom right.

Now connect it with the server

```
ubuntu@ip-172-31-97-13:~$ 
[1]+ 11111111+ 0x01111111 MINGW64 ~ (master)
$ cd Downloads/
$ ssh -i "anil.pem" ubuntu@ec2-18-212-122-19.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.18.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Thu Dec 19 15:01:55 UTC 2024

system load: 0.0          Processes:      113
Usage of /: 37.6% of 7.57GB   Users logged in:    0
Memory usage: 8%           IPv4 address for eth0: 172.31.97.13
Swap usage: 0%             IPv6 address for eth0: fe80::501:1ff:fe17:231%eth0

Expanded Security Maintenance for Applications is not enabled.

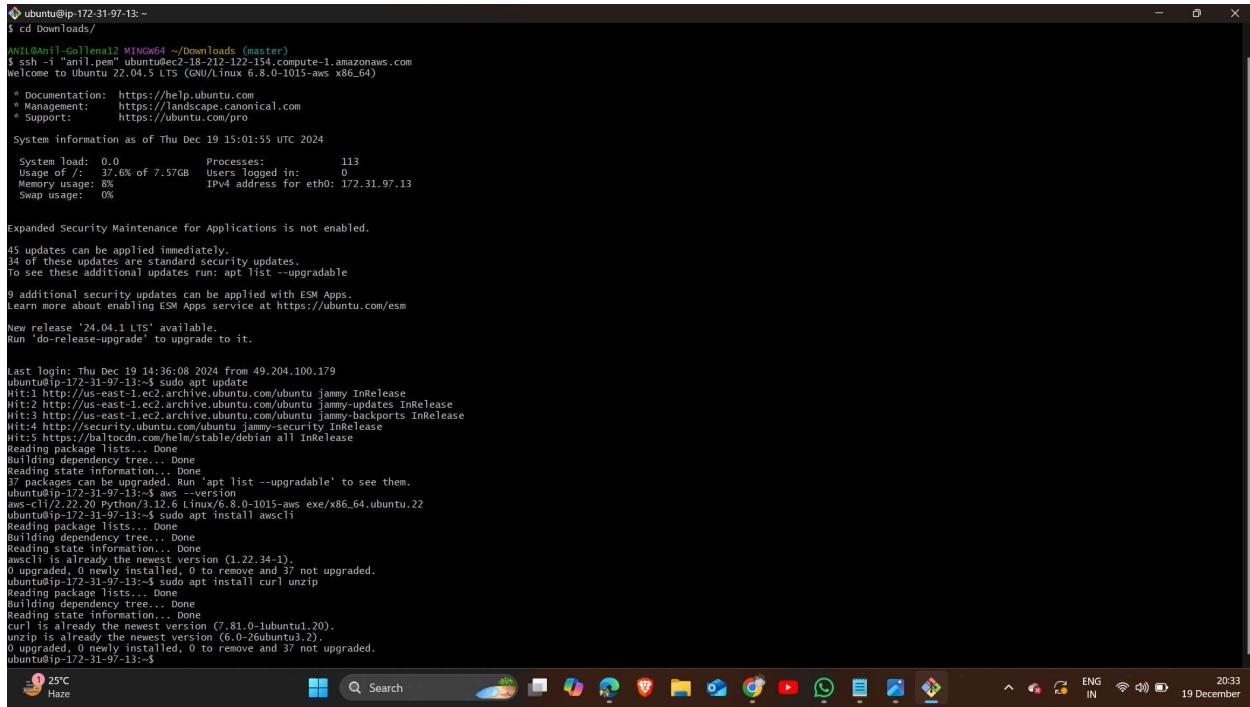
45 updates can be applied immediately.
34 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

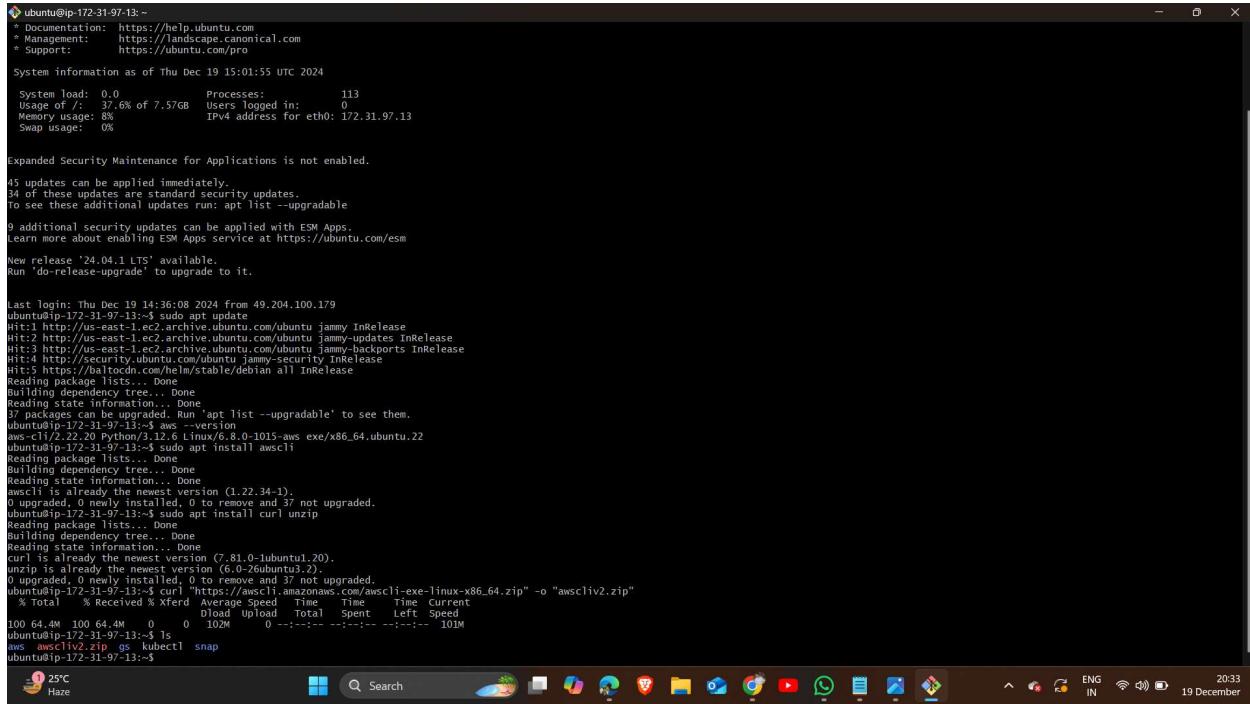
Last login: Thu Dec 19 14:36:08 2024 from 49.204.100.179
ubuntu@ip-172-31-97-13:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://apt.fedorainfracloud.org/ubuntu/stable/debian all InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded,
0 downloaded, 0 checksum errors.
ubuntu@ip-172-31-97-13:~$
```

And update the ubuntu install updated CLI



```
ubuntu@ip-172-31-97-13:~  
$ cd Downloads/  
ANILANIL-Gallenai2 MINGW64 ~/Downloads (master)  
$ ssh -i "anil.pem" ubuntu@ec2-18-212-122-154.compute-1.amazonaws.com  
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.18.0-1015-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Thu Dec 19 15:01:55 UTC 2024  
  
System load: 0.0 Processes: 113  
Usage of /: 37.6% of 7.57GB Users logged in: 0  
Memory usage: 8% IPv4 address for eth0: 172.31.97.13  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
45 updates can be applied immediately.  
34 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
9 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
New release '24.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Thu Dec 19 14:36:08 2024 from 49.204.100.179  
ubuntu@ip-172-31-97-13:~$ sudo apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:5 https://baltocdn.com/helm/stable/debian all InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
37 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-172-31-97-13:~$ curl --version  
awscli/2.22.20 Python/3.12.6 Linux/6.8.0-1015-aws exe/x86_64.ubuntu.22  
ubuntu@ip-172-31-97-13:~$ sudo apt install curl unzip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
curl is already the newest version (7.81.0-1ubuntu1.20).  
unzip is already the newest version (6.0-2ubuntu3.2).  
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.  
ubuntu@ip-172-31-97-13:~$  
$ 25C Haze 20:33 ENG IN 19 December
```

Now install kubectl unzip the files



```
ubuntu@ip-172-31-97-13:~  
$ cd Downloads/  
ANILANIL-Gallenai2 MINGW64 ~/Downloads (master)  
$ ssh -i "anil.pem" ubuntu@ec2-18-212-122-154.compute-1.amazonaws.com  
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.18.0-1015-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Thu Dec 19 15:01:55 UTC 2024  
  
System load: 0.0 Processes: 113  
Usage of /: 37.6% of 7.57GB Users logged in: 0  
Memory usage: 8% IPv4 address for eth0: 172.31.97.13  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
45 updates can be applied immediately.  
34 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
9 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
New release '24.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Thu Dec 19 14:36:08 2024 from 49.204.100.179  
ubuntu@ip-172-31-97-13:~$ sudo apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:5 https://baltocdn.com/helm/stable/debian all InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
37 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-172-31-97-13:~$ aws --version  
aws-cli/2.22.20 Python/3.12.6 Linux/6.8.0-1015-aws exe/x86_64.ubuntu.22  
ubuntu@ip-172-31-97-13:~$ sudo apt install awscli  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
awscli is already the newest version (1.22.34-1).  
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.  
ubuntu@ip-172-31-97-13:~$ sudo apt install curl unzip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
curl is already the newest version (7.81.0-1ubuntu1.20).  
unzip is already the newest version (6.0-2ubuntu3.2).  
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.  
ubuntu@ip-172-31-97-13:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
  % total    % received   % Xferd      Average Speed   Time     Current  
 100 64.4M 100 64.4M  0      0 102M 0 ---:---:---:---:---:--- 101M  
ubuntu@ip-172-31-97-13:~$ ls  
awscliv2.zip  kubectl snap  
ubuntu@ip-172-31-97-13:~$  
$ 25C Haze 20:33 ENG IN 19 December
```

Now connect it with the aws management console by using aws configure

Add secret key and access key

The screenshot shows a browser window for the AWS Management Console. The URL is `us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#security_credentials/access-key-wizard`. The page title is "Create access key". The main message is "Access key created" with a note: "This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time." Below this, there are two steps: "Step 1 Alternatives to root user access keys" and "Step 2 Retrieve access key". The "Retrieve access key" step is selected. It displays an "Access key" section with fields for "Access key" (containing "AKIASFTY7SKO7KDVWLVB") and "Secret access key" (containing "***** Show"). Below this is an "Access key best practices" section with a bulleted list: "Never store your access key in plain text, in a code repository, or in code.", "Disable or delete access key when no longer needed.", "Enable least-privilege permissions.", and "Rotate access keys regularly.". At the bottom right are "Download .csv file" and "Done" buttons.

The screenshot shows a terminal window on a Linux desktop. The terminal output includes:

```
ubuntu@ip-172-31-97-13:~$ Usage of /: 37.6% of 7.57GB Users logged in: 0
Memory usage: 8%          IPv4 address for eth0: 172.31.97.13
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
45 updates can be applied immediately.
34 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Dec 19 14:36:08 2024 from 49.204.100.179
ubuntu@ip-172-31-97-13:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://archive.ubuntu.com/helium/stable/debian all InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
37 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
awscli is already the newest version (1.22.34-1).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
ubuntu@ip-172-31-97-13:~$ sudo apt install curl unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.20).
unzip is already the newest version (6.0-26ubuntu3).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
ubuntu@ip-172-31-97-13:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 64.4M 100 64.4M    0     0  102M  0 ---:---:---:--- 101M
ubuntu@ip-172-31-97-13:~$ ls
awscliv2.zip  kubelet  snap
ubuntu@ip-172-31-97-13:~$ sudo snap install kubelet --classic
snap "kubelet" is already installed, see 'snap help refresh'
ubuntu@ip-172-31-97-13:~$ snap refresh
AWS Access Key ID [*****];
AWS Secret Access Key [*****];
Default region name [None];
Default output format [None];
ubuntu@ip-172-31-97-13:~$
```

```
ubuntu@ip-172-31-97-13:~$ 37 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-97-13:~$ aws --version
aws-cli/2.22.20 Python/3.12.6 Linux/6.8.0-1015-aws exe/x86_64/ubuntu.22
ubuntu@ip-172-31-97-13:~$ sudo apt install awscli
[sudo] password for ubuntu: 
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
awscli is already the newest version (1.22.34-1).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
ubuntu@ip-172-31-97-13:~$ sudo apt install curl unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-ubuntu1.20).
unzip is already the newest version (6.0-2ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
ubuntu@ip-172-31-97-13:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total   Spent    Left  Speed
 0       0      0     0    0:00:00 --:--:-- --:--:-- 0
ubuntu@ip-172-31-97-13:~$ ls
aws awsCliv2.zip gs kubectl snap
ubuntu@ip-172-31-97-13:~$ sudo snap install kubectl --classic
ubuntu@ip-172-31-97-13:~$ kubectl config set-context $(aws configure get aws_access_key_id) --current
ubuntu@ip-172-31-97-13:~$ aws configure
AWS Access Key ID [*****LWbV]: AWS Secret Access Key [*****npE]:
default region name [None]:
AWS Default Output Format [json]:
ubuntu@ip-172-31-97-13:~$ aws sts get-caller-identity
usage: aws [options] <command> [<subcommand> [<subcommand> ...] [parameters]
to see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help

aws: error: argument operation: Invalid choice, valid choices are:

assume-role           | assume-role-with-saml
assume-role-with-web-identity | assume-role-with-web-identity
decode-authorization-message | get-access-key-info
get-caller-identity    | get-federation-token
get-session-token      | help

Invalid choice: 'get-caller-identity', maybe you meant:
  * get-caller-identity

ubuntu@ip-172-31-97-13:~$ aws sts get-caller-identity
{
  "UserId": "905418085021",
  "AccountId": "905418085021",
  "Arn": "arn:aws:iam::905418085021:root"
}
ubuntu@ip-172-31-97-13:~$
```

```
ubuntu@ip-172-31-97-13: ~
inflate: aws/dist/awscli/examples/rds/modify-db-proxy-target-group.rst
replace aws/dist/awscli/examples/rds/purchase-reserved-instance.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/revoke-reserved-instance.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
replace aws/dist/awscli/examples/rds/create-db-instance-read-replica.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/create-db-instance-read-replica.rst
replace aws/dist/awscli/examples/rds/generate-auth-token.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/generate-auth-token.rst
replace aws/dist/awscli/examples/rds/describe-db-log-files.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/create-db-cluster-parameter-group.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
replace aws/dist/awscli/examples/rds/create-db-cluster-parameter-group.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
replace aws/dist/awscli/examples/rds/modify-db-snapshot-attribute.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/modify-db-snapshot-attribute.rst
replace aws/dist/awscli/examples/rds/modify-db-snapshots-to-descrIBE.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/modify-db-snapshots-to-descrIBE.rst
replace aws/dist/awscli/examples/rds/start-db-cluster.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/start-db-cluster.rst
replace aws/dist/awscli/examples/rds/modify-db-instance.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflate: aws/dist/awscli/examples/rds/modify-db-instance.rst
replace aws/dist/awscli/examples/rds/delete-db-security-group.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace aws/dist/awscli/examples/rds/create-event-subscription.rst? [y]es, [n]o, [A]ll, [N]one, [r]ename: nCubunto@ip-172-31-97-13:~$ ^C
Found preexisting AWS CLI installation /usr/local/aws-cli/v2/current. Please rerun install script with --update flag.
ubuntu@ip-172-31-97-13:~$ ls
aws awscliV2.zip gs kubectl snap
ubuntu@ip-172-31-97-13:~$ unzip awscliV2.zip
Archive: awscliV2.zip
Replace aws/install? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
inflate: aws/install
inflate: aws/README_md
inflate: aws/THIRD_PARTY_LICENSES
inflate: aws/dist/aws
inflate: aws/dist/aws_completer
inflate: aws/dist/libpython3.12.so.1.0
inflate: aws/dist/_ctypes.so.1.1.0
inflate: aws/dist/_cffi_backend.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/_ruamel_yaml.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/libbz2.so.1
inflate: aws/dist/liblzma.so.5
inflate: aws/dist/libbz2.so.1
inflate: aws/dist/_libffi.so.6
inflate: aws/dist/_libuid.so.1
inflate: aws/dist/_lgcgc_.so.1
inflate: aws/dist/_lumix_.so.0.0
inflate: aws/dist/base_library.zip
inflate: aws/dist/lib_dylonload_blaKE2.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_md5.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_sha1.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_sha256.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_random.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_bisect.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_opcode.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_csv.cpython-312-x86_64-linux-gnu.so
inflate: aws/dist/lib_dylonload_binacl1.cpython-312-x86_64-linux-gnu.so
```

Now install HELM from Google.

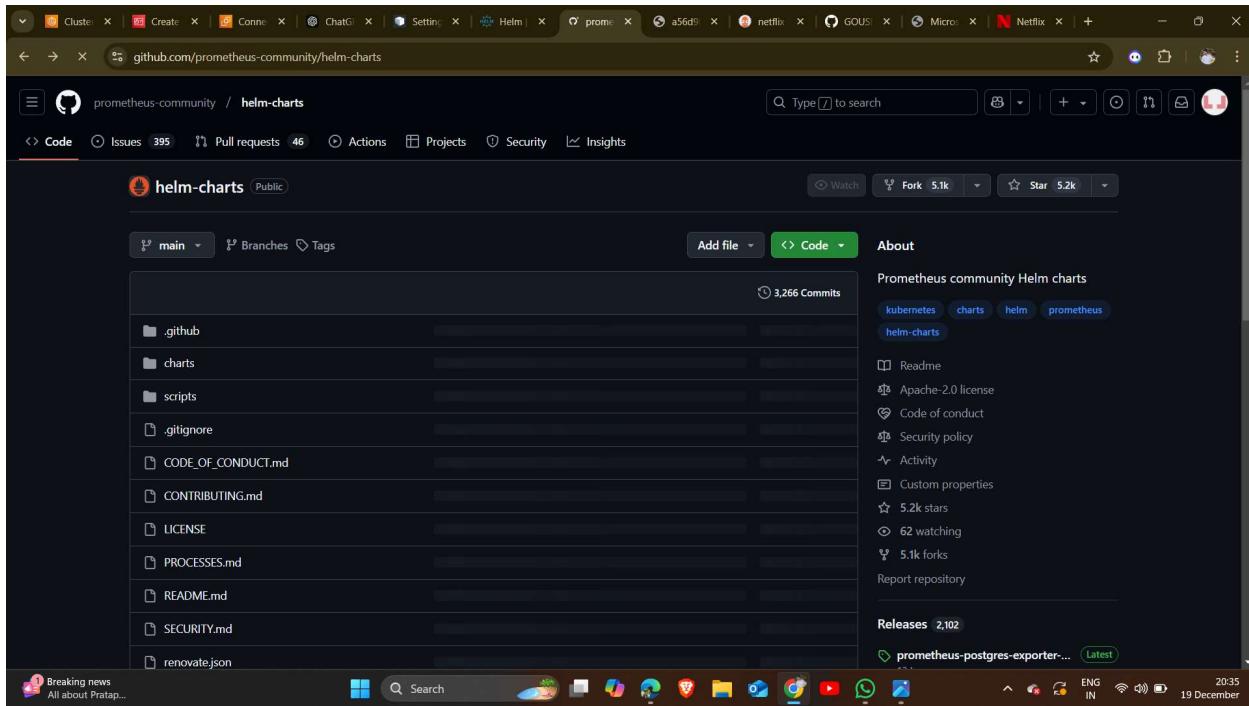
Helm is a package manager for Kubernetes that simplifies the process of deploying, managing, and configuring applications on Kubernetes clusters.

Think of it as a tool to automate Kubernetes deployments using pre-configured templates and reusable components called Helm Charts.

The screenshot shows a browser window displaying the Helm documentation at helm.sh/docs/intro/install/. The page has a header with tabs for Home, Docs, Charts, Blog, and Community, and a language selector for English v3.16.2. On the left, there's a sidebar with navigation links like Docs Home, Introduction, Quickstart Guide, Installing Helm, Using Helm, Cheat Sheet, How-To, Topics, Best Practices, Chart Template Guide, Helm Commands, and a Contribute to Docs button. The main content area has two sections: "From Apt (Debian/Ubuntu)" which includes a note about a Helm package for Apt and a command-line snippet for setting up GPG keys and installing the package; and "From dnf/yum (fedora)" which notes that helm is available in the official repository since Fedora 35 and provides a command to install it. The bottom of the screen shows a Windows taskbar with various icons and system status.

Now set up helm chart

The screenshot shows a GitHub repository page for "Prometheus Community Kubernetes Helm Charts" at github.com/prometheus-community/helm-charts/blob/main/README.md. The page displays the README.md file, which contains information about the repository, usage instructions, and contributing guidelines. It includes a sidebar with files like .github, charts, scripts, .gitignore, CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE, PROCESSES.md, README.md, SECURITY.md, and renovate.json. The GitHub interface shows a pull request by arukidou, a file preview, and release charts. The bottom of the screen shows a Windows taskbar with various icons and system status.



- Now Change API version to v1 then add interactiveMode: IfAvailable in config file.
 - Then AWS configure with the access key and secret key.
 - Now setup Prometheus node exporter namespace.

```
ubuntu@ip-172-31-97-13: ~
Usage:
  kubectl [flags] [options]
Use "kubectl <command> --help" for more information about a given command.
See "kubectl options" for a list of global command-line options (applies to all commands).
ubuntu@ip-172-31-97-13:~$ kubectl api-versions
error: interactiveMode must be specified for arn:aws:eks:us-east-1:905418085021:cluster/mycluser to use exec authentication plugin
ubuntu@ip-172-31-97-13:~$ kubectl plugin
Provide utilities for interacting with plugins.

Plugins provide extended functionality that is not part of the major command-line distribution. Please refer to the documentation and examples for more information about how write your own plugins.

The easiest way to discover and install plugins is via the kubernetes sub-project krew: [krew.sigs.k8s.io]. To install krew, visit https://krew.sigs.k8s.io/docs/user-guide/setup/install

Examples:
# List all available plugins
kubectl plugin list

# List only binary names of available plugins without paths
kubectl plugin list --name-only

Available Commands:
  list          list all visible plugin executables on a user's PATH

Usage:
  kubectl plugin [Flags] [options]

Use "kubectl plugin <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).
ubuntu@ip-172-31-97-13:~$ version
Client Version: v1.31.3
Kubelet Version: v1.31.3-eks-59bf375
Kubernetes Version: v1.31.3-eks-59bf375
ubuntu@ip-172-31-97-13:~$ kubectl version
error: interactiveMode must be specified for arn:aws:eks:us-east-1:905418085021:cluster/mycluser to use exec authentication plugin
ubuntu@ip-172-31-97-13:~$ sudo vi ./kube/config
ubuntu@ip-172-31-97-13:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-97-168.ec2.internal   Ready    <none>    77m   v1.31.3-eks-59bf375
ip-172-31-98.ec2.internal   Ready    <none>    73m   v1.31.3-eks-59bf375
ubuntu@ip-172-31-97-13:~$ aws eks --region us-east-1 update-kubeconfig --name mycluser
error: command not found: aws
ubuntu@ip-172-31-97-13:~$ curl -L https://amazon-eks.s3.us-west-2.amazonaws.com/1.31/2021-09-13/bin/linux/amd64/kubeconfig -o /home/ubuntu/.kube/config
ubuntu@ip-172-31-97-13:~$ kubectl create namespace prometheus-node-exporter
error: exec plugin: invalid apiverion "client.authentication.k8s.io/v1alpha1"
ubuntu@ip-172-31-97-13:~$ sudo vi ./kube/config
ubuntu@ip-172-31-97-13:~$ kubectl create namespace prometheus-node-exporter
error: interactiveMode must be specified for arn:aws:eks:us-east-1:905418085021:cluster/mycluser to use exec authentication plugin
ubuntu@ip-172-31-97-13:~$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-97-168.ec2.internal   Ready    <none>    80m   v1.31.3-eks-59bf375
ip-172-31-98.ec2.internal   Ready    <none>    76m   v1.31.3-eks-59bf375
ubuntu@ip-172-31-97-13:~$ kubectl create namespace prometheus-node-exporter
namespace/prometheus-node-exporter created
ubuntu@ip-172-31-97-13:~|
```

Now setup argoCD for that get commands from google

The screenshot shows a web browser window with several tabs open, including 'Clusters | Elastic Kibana', 'Create access key | ...', 'Microsoft Word - NE...', 'ChatGPT', 'Setting up Argo CD...', 'Helm | Installing Helm...', and 'prometheus-community | ...'. The main content area displays instructions for setting up Argo CD, mentioning dependencies and downloads. It includes terminal command examples:

```
$ helm repo add argo-cd https://argoproj.github.io/argo-helm
$ helm dep update charts/argo-cd/
```

This will create the `Chart.lock` and `charts/argo-cd-<version>.tgz` files. The `.tgz` file is only required for the initial installation from our local machine. To avoid accidentally committing it, we can add it to the `gitignore` file:

```
$ echo "charts/**/charts" >> .gitignore
```

Our custom chart is ready and can be pushed to our public Git repository:

```
$ git add charts/argo-cd
$ git commit -m 'add argo-cd chart'
$ git push
```

The next step is to install our chart.

Installing our Helm chart

We have to do the initial installation manually from our local machine, later we set up Argo CD to manage itself (meaning that Argo CD will automatically detect any changes to the helm chart and synchronize it):

```
ubuntu@ip-172-31-97-13:~$ sudo vi ~/.kube/config
134 kubectl get nodes
135 kubectl config set-context us-east-1 update-kubeconfig --name mycluster
136 kubectl create namespace prometheus-node-exporter
137 sudo vi ~/.kube/config
138 kubectl create namespace prometheus-node-exporter
139 kubectl get nodes
140 kubectl get config
141 kubectl get nodes
142 kubectl create namespace prometheus-node-exporter
143 helm install prometheus-node-exporter prometheus-community/prometheus-node-exporter --namespace prometheus-node-exporter
144 hispter@ip-172-31-97-13:~$ helm repo add argo-cd https://argoproj.github.io/argo-helm
$: command not found
ubuntu@ip-172-31-97-13:~$ helm repo add argo-cd https://argoproj.github.io/argo-helm
Command 'repo' not found, but can be installed with:
sudo apt install helm
ubuntu@ip-172-31-97-13:~$ helm repo add argo-cd https://argoproj.github.io/argo-helm
"argo-cd" has been added to your repositories
ubuntu@ip-172-31-97-13:~$ helm dep update charts/argo-cd/
Error: could not find charts/argo-cd: stat charts/argo-cd: no such file or directory
ubuntu@ip-172-31-97-13:~$ helm dep update charts/argo-cd/
Error: could not find charts/argo-cd: stat charts/argo-cd: no such file or directory
ubuntu@ip-172-31-97-13:~$ helm repo add argo-cd https://argoproj.github.io/argo-helm
"argo-cd" already exists with the same configuration, skipping
ubuntu@ip-172-31-97-13:~$ helm repo update
Hang tight while I grab the latest version of your chart repositories...
...Successfully got an update from the 'argo-cd' chart repository
...Successfully got an update from the 'prometheus-community' chart repository
Update Complete: *Happy Helm-ing!*
ubuntu@ip-172-31-97-13:~$ kubectl create namespace argocd
namespace/argocd created
ubuntu@ip-172-31-97-13:~$ helm install argocd argo-cd/argo-cd -n argocd
NAME: argocd
LAST DEPLOYED: Thu Dec 19 11:57:52 2024
NAMESPACE: argocd
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
In order to access the server UI you have the following options:
1. kubectl port-forward service/argocd-server -n argocd 8080:443
   and then open the browser on http://localhost:8080 and accept the certificate
2. enable ingress in the values file 'server.ingress.enabled' and either
   - Add the annotation for ssl passthrough: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-1-ssl-passthrough
   - Set the 'configs.params.'server.insecure' in the values file and terminate SSL at your ingress: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-2-multiple-ingress-objects-and-https
```

After reaching the UI the first time you can login with username: admin and the random password generated during the installation. You can find the password by running:

```
kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d
```

(You should delete the initial secret afterwards as suggested by the Getting Started Guide: https://argo-cd.readthedocs.io/en/stable/getting_started/#4-login-using-the-cli)

```

ubuntu@ip-172-31-97-13:~ 
ubuntu@ip-172-31-97-13:~ echo $ARGOCD_SERVER
kubectl get svc argo-cd-server -n argo-cd -o json | jq --raw-output .status.loadBalancer.ingress[0].hostname
ubuntu@ip-172-31-97-13:~ echo $ARGOCD_SERVER
Command 'jq' not found, but can be installed with:
sudo snap install jq # version 1.5+dfsg-1, or
sudo apt install jq # version 1.6-2.ubuntu3
sudo snap install jq # version 1.5+dfsg-1, or
sudo apt install jq # version 1.6-2.ubuntu3
See 'snap help install' for further options.
ubuntu@ip-172-31-97-13:~ sudo apt install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libjq1 libjq1v5
The following NEW packages will be installed:
  jq libjq1 libjq1v5
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
Need to get 357 kB of archives.
After this operation, 1087 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://us-east2.archive.ubuntu.com/ubuntu jammy/main amd64 libonig5 amd64 6.9.7.1-2build1 [172 kB]
Get:2 https://us-east2.archive.ubuntu.com/ubuntu jammy/main amd64 libjq1v5 amd64 1.6-2.ubuntu3 [133 kB]
Get:3 https://us-east2.archive.ubuntu.com/ubuntu jammy/main amd64 jq amd64 1.6-2.ubuntu3 [52.5 kB]
Fetched 357 kB in 0s (9990 kB/s)
Selecting previously unselected package libonig5:amd64.
(Reading database ... 7779 files and directories currently installed.)
Preparing to unpack .../libonig5_6.9.7.1-2build1_amd64.deb ...
Unpacking libonig5:amd64 (6.9.7.1-2build1) ...
Selecting previously unselected package libjq1:amd64.
Preparing to unpack .../libjq1_1.6-2.ubuntu3_amd64.deb ...
Unpacking libjq1:amd64 (1.6-2.ubuntu3) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.6-2.ubuntu3_amd64.deb ...
Unpacking jq (1.6-2.ubuntu3) ...
Setting up libonig5:amd64 (6.9.7.1-2build1) ...
Setting up libjq1:amd64 (1.6-2.ubuntu3) ...
Setting up jq (1.6-2.ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-97-13:~$ kubectl get svc argo-cd-server -n argo-cd -o json | jq --raw-output .status.loadBalancer.ingress[0].hostname
null
ubuntu@ip-172-31-97-13:~$
```

Here a new loadbalancer is created

```

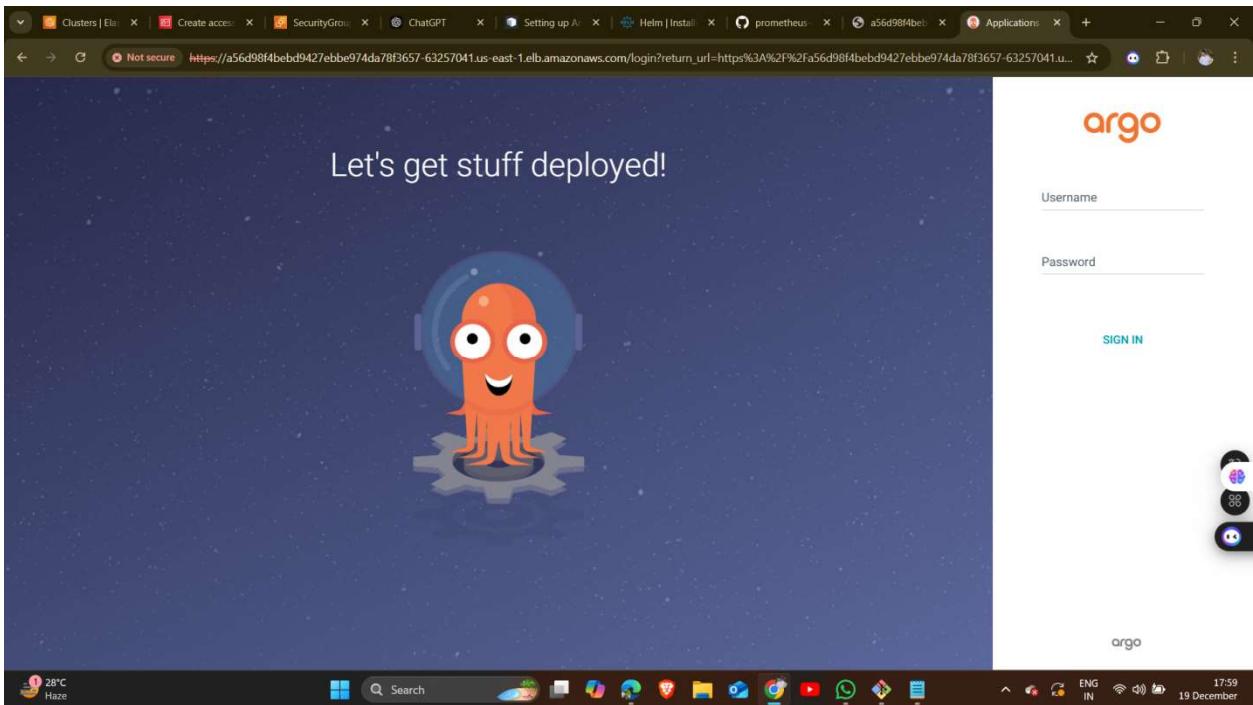
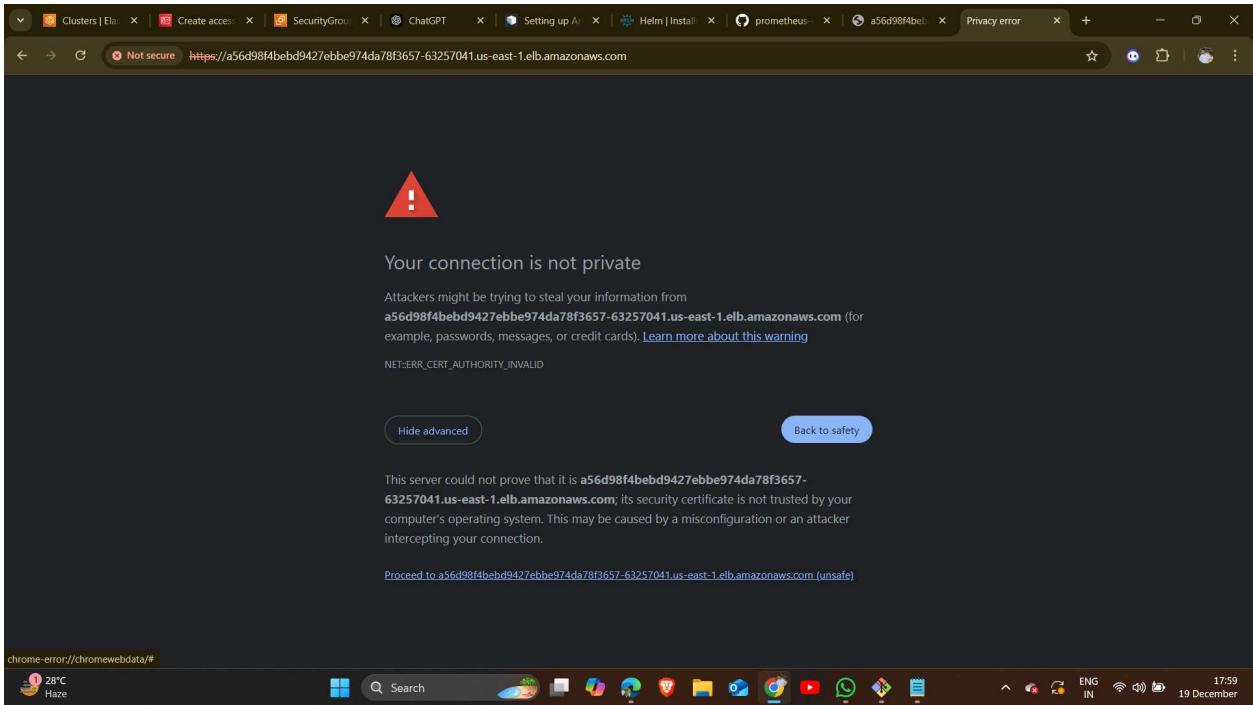
ubuntu@ip-172-31-97-13:~ 
29m Normal Created pod/argocd-server-748dd69b68-fj22
29m Warning FailedMount pod/argocd-application-controller-0
for the condition
29m Normal Pulling pod/argocd-redis-64987658bf-5484q
29m Warning FailedMount pod/argocd-applicationset-controller-65f874574b-lnrnr
or the condition
29m Normal FailedMount pod/argocd-applicationset-controller-65f874574b-lnrnr
the condition
29m Normal Created pod/argocd-repo-server-76dcfb4c-rdpnr
29m Normal Pulled pod/argocd-repo-server-76dcfb4c-rdpnr
29m Normal Pulling pod/argocd-dex-server-557cf46859-wh6rp
29m Normal Created pod/argocd-applicationset-controller-65f874574b-lnrnr
29m Normal Pulled pod/argocd-redis-64987658bf-5484q
29m Normal Created pod/argocd-redis-64987658bf-5484q
0) Image size: 19463286 bytes.
29m Normal Started pod/argocd-notifications-controller-66bd5bdb76-47x4d
29m Normal Started pod/argocd-application-controller-0
29m Normal Created pod/argocd-application-controller-0
29m Normal Created pod/argocd-application-controller-0
29m Normal Pulled pod/argocd-notifications-controller-66bd5bdb76-47x4d
29m Normal Started pod/argocd-redis-64987658bf-5484q
29m Normal Started pod/argocd-server-76dcfb4c-rj22
29m Normal Started pod/argocd-applicationset-controller-65f874574b-lnrnr
29m Normal Pulled pod/argocd-repo-server-76dcfb4c-rdpnr
29m Normal Created pod/argocd-repo-server-76dcfb4c-rdpnr
29m Normal Started pod/argocd-dex-server-557cf46859-wh6rp
29m Normal Created pod/argocd-dex-server-557cf46859-wh6rp
29m Normal Pulled pod/argocd-dex-server-557cf46859-wh6rp
80011034 bytes.
28m Normal Pulling pod/argocd-dex-server-557cf46859-wh6rp
28m Normal Created pod/argocd-dex-server-557cf46859-wh6rp
28m Normal Pulled pod/argocd-dex-server-557cf46859-wh6rp
01 bytes.
28m Normal Started pod/argocd-dex-server-557cf46859-wh6rp
ubuntu@ip-172-31-97-13:~$ kubectl patch svc argo-cd-server -n argo-cd -o yaml --patch='{"spec": {"type": "LoadBalancer"} }'
service/argo-cd-server patched
ubuntu@ip-172-31-97-13:~$ export ARGOCD_SERVER=$(kubectl get svc argo-cd-server -n argo-cd -o json | jq -r ".status.loadBalancer.ingress[0].hostname")
ubuntu@ip-172-31-97-13:~$ echo $ARGOCD_SERVER
a56d98f4bedbd9427ebbe974da7f83657-63257041.us-east-1.elb.amazonaws.com
ubuntu@ip-172-31-97-13:~$ export ARGOCD_SERVER=$(kubectl get svc argo-cd-server -n argo-cd -o json | jq -r ".status.loadBalancer.ingress[0].hostname")
ubuntu@ip-172-31-97-13:~$ export ARGOCD_PWD=$(kubectl get secret argo-cd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d)
ubuntu@ip-172-31-97-13:~$ echo $ARGOCD_PWD
RL1yIP90huASEB
ubuntu@ip-172-31-97-13:~$ client_loop: send disconnect: Connection reset by peer
ANIL@ANIL-Gallen12 MINGW64 ~ /Downloads (master)
$ ssh -i "anil.pem" ubuntu@ec2-18-212-122-154.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.18.0-1019-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

 25C Haze

```

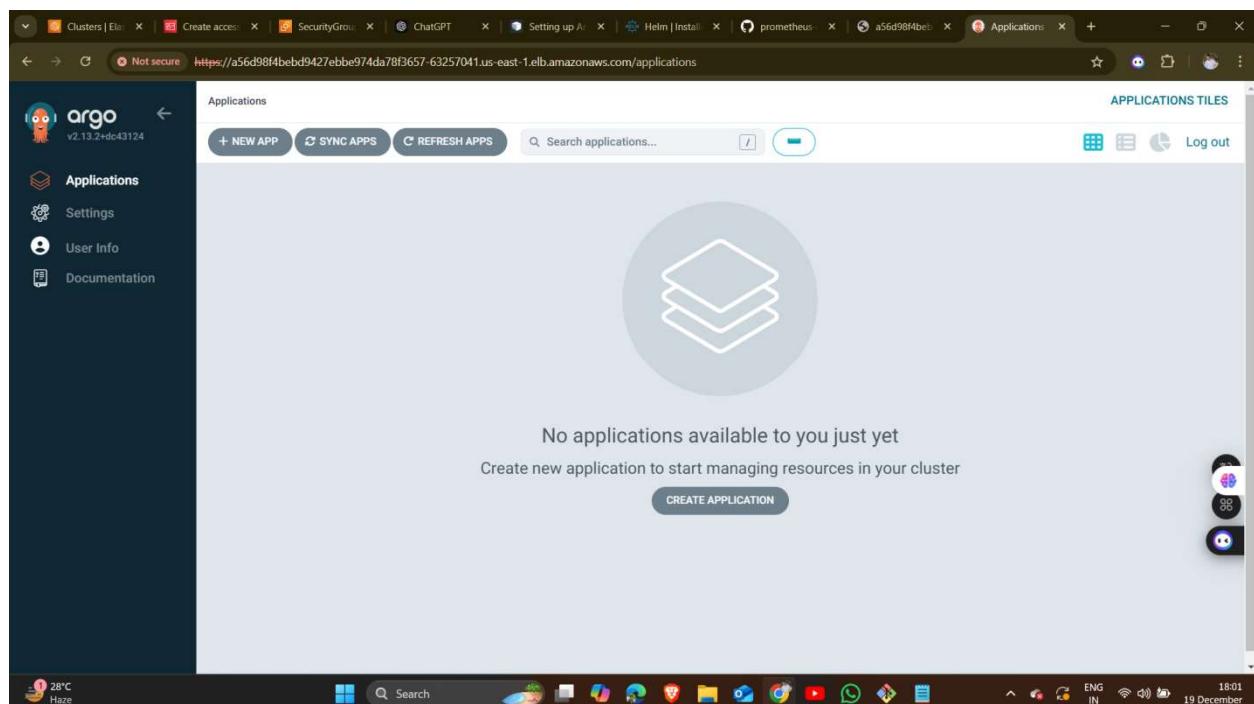
Copy the load balancer arn allow the port number



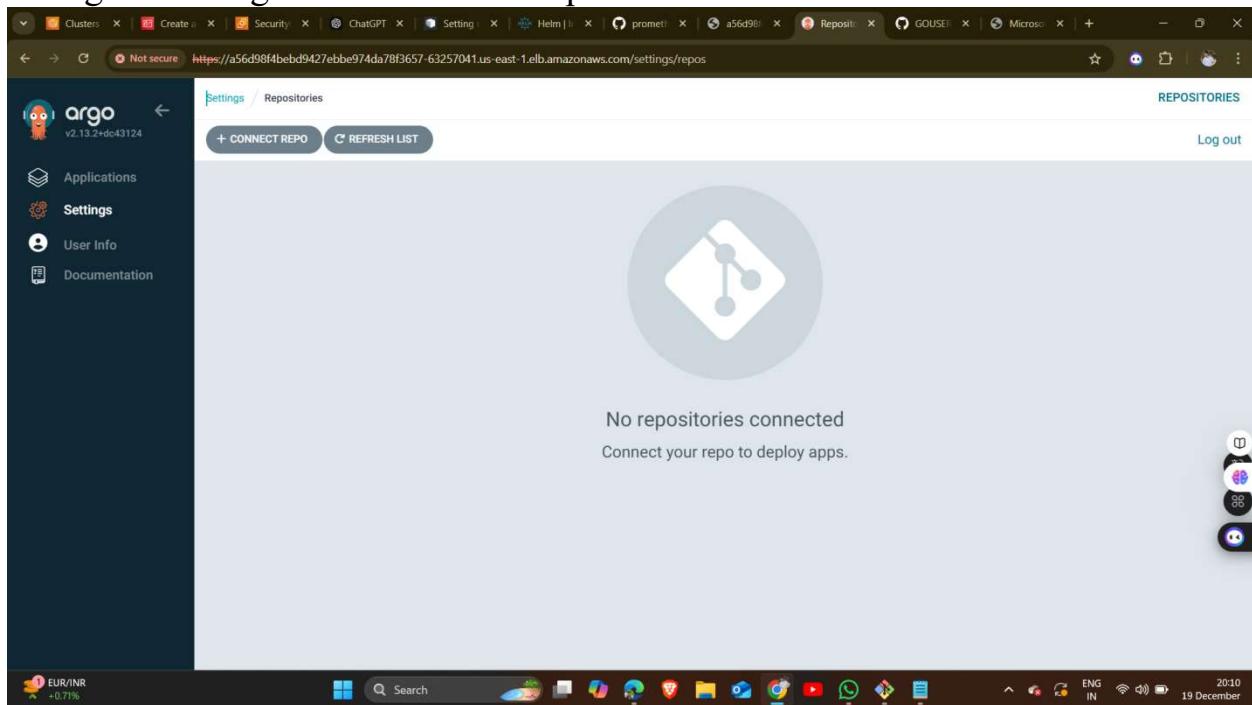
Here the user name is “admin” and for password run the command as

“echo \$ARGO _PWD” there you get password copy that password login to the page.

Here are the commands I used to implement the whole process

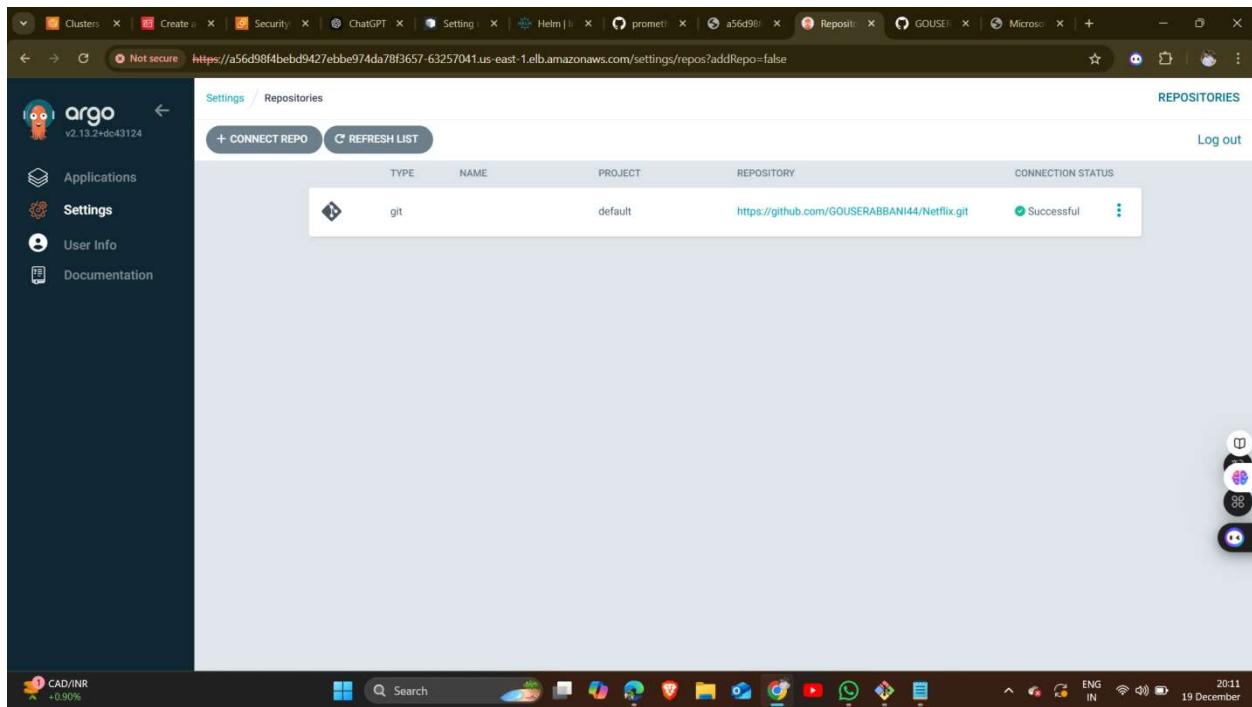


Now go to settings and connect the repo

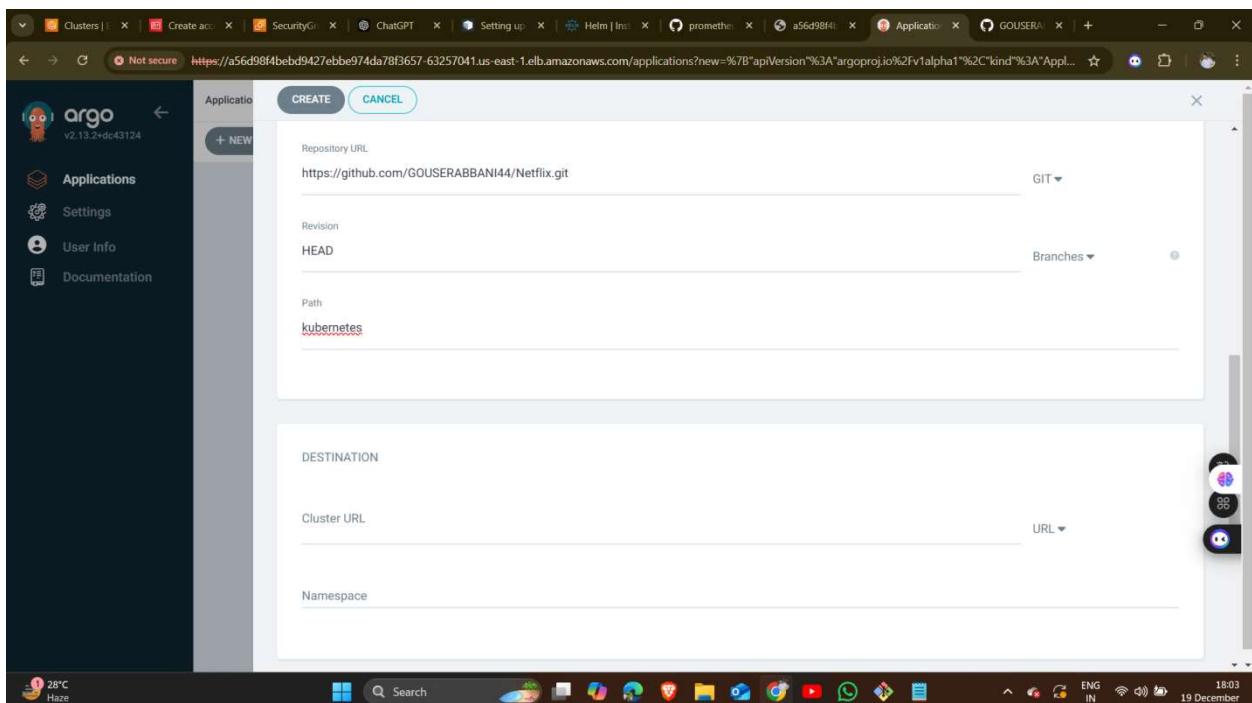
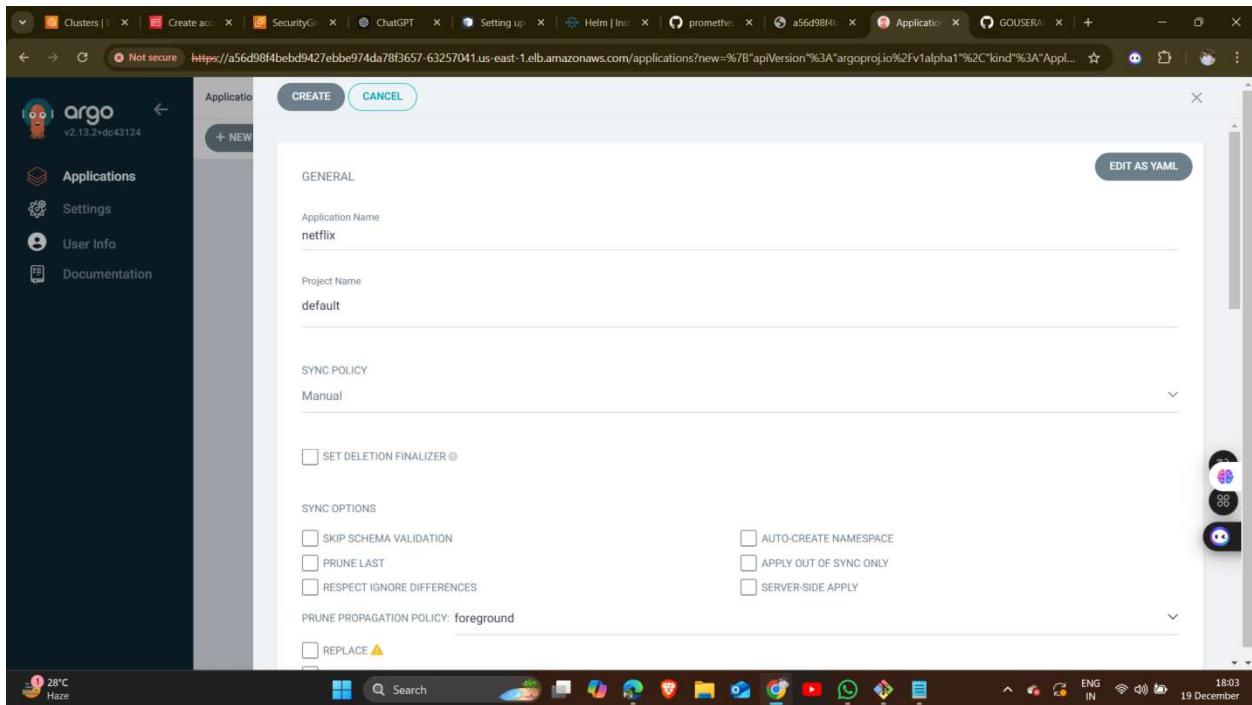


Click the connect repo

Now connect via HTTP, name as git and copy the Netflix application files under the git URL make project as default



Now create application



The screenshot shows the Argo UI interface. On the left, there's a sidebar with navigation links like 'Clusters', 'Create', 'Security', 'ChatGPT', 'Setting', 'Helm', 'prometheus', 'Application', 'GOUSER', 'Microservices', and 'Metrics'. The main area is titled 'Applications' and shows a single application named 'netflix'. The application details are as follows:

- Project: default
- Labels: (empty)
- Status: Missing, OutOfSync
- Repository: <https://github.com/Gouserabbani44/>
- Target Ref: HEAD
- Path: Kubernetes
- Destination: in-cluster
- Namespace: default
- Created: 12/19/2024 20:14:52 (in a few seconds)

Below the application details are three buttons: 'SYNC', 'REFRESH', and 'DELETE'. The bottom right corner of the window shows system icons for battery, signal, and date/time.

Now sync the application

The screenshot shows the Argo UI interface with the 'netflix' application selected. The 'SYNC STATUS' tab is active, showing the status as 'OutOfSync' from HEAD (503e507). The 'SYNC' button is highlighted. A large 'Synchronize' dialog box is open on the right side of the screen, containing the following information:

Synchronizing application manifests from <https://github.com/Gouserabbani44/Netflix.git>

Revision: HEAD

Sync Options:

- PRUNE
- DRY RUN
- APPLY ONLY
- FORCE

Sync Options (continued):

- SKIP SCHEMA VALIDATION
- AUTO-CREATE NAMESPACES
- PRUNE LAST
- APPLY OUT OF SYNC ONLY
- RESPECT IGNORE DIFFERENCES
- SERVER-SIDE APPLY

Prune Propagation Policy: foreground

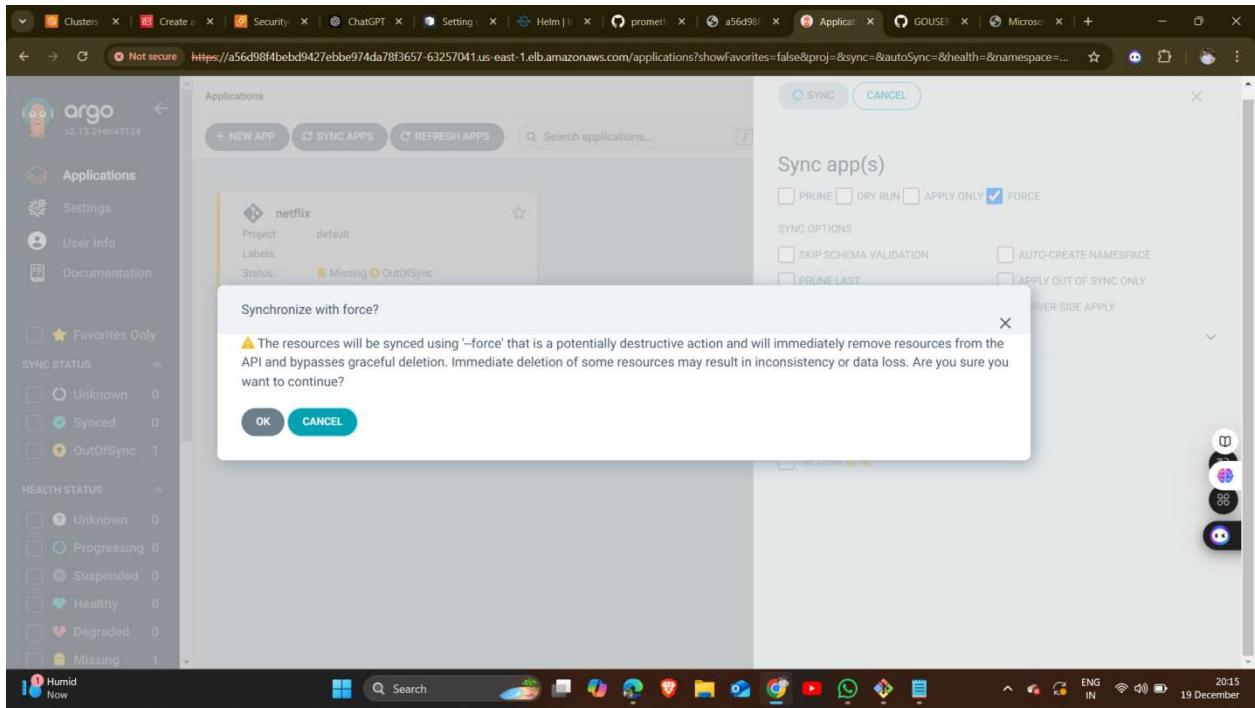
- REPLACE
- RETRY

Synchronize Resources: all / out of sync / none

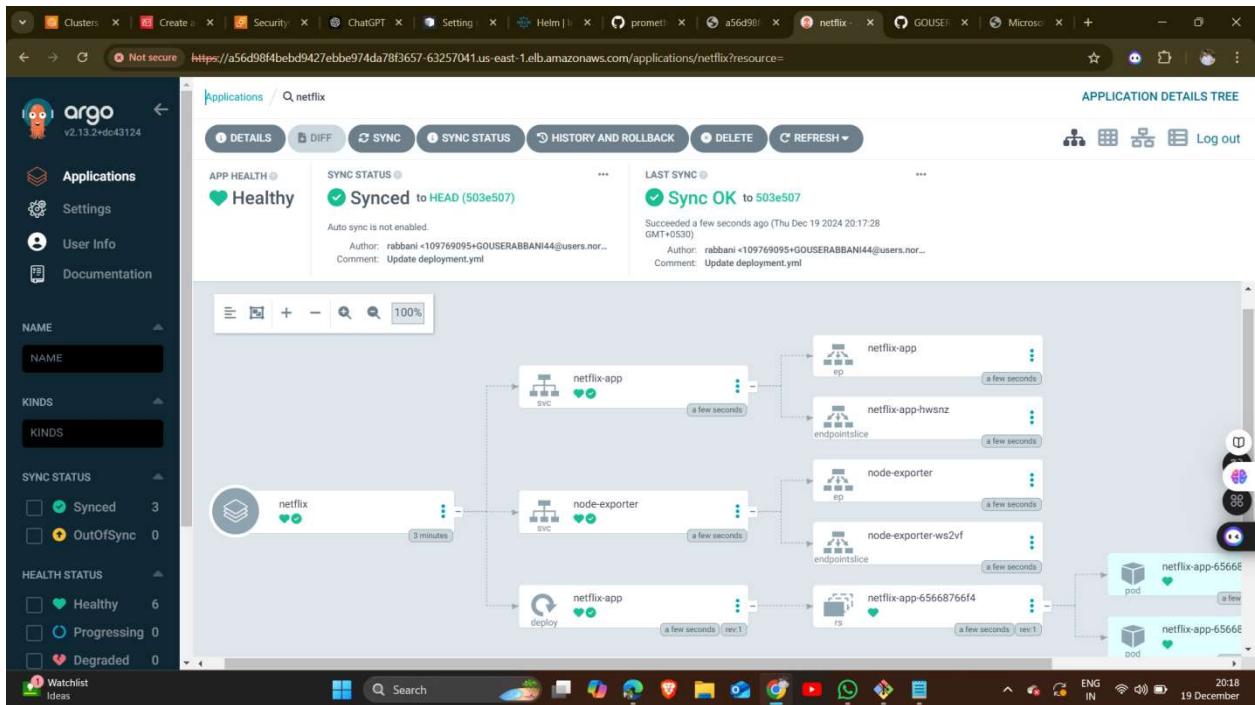
Selected resources:

- /SERVICE/DEFAULT/NETFLIX-APP
- /SERVICE/PROMETHEUS-NODE-EXPORTER/NODE-EXPORTER
- APPS/DEPLOYMENT/DEFAULT/NETFLIX-APP

The bottom right corner of the window shows system icons for battery, signal, and date/time.



Click on ok



Here is the summary of the application

The screenshot shows the Argo UI interface. On the left, there's a sidebar with navigation links like 'Applications', 'Settings', 'User Info', and 'Documentation'. The main area displays a summary for the 'netflix-app' application, which is listed under the 'svc' kind. The 'SUMMARY' tab is selected, showing details such as the name 'netflix-app', namespace 'default', creation date '12/19/2024 20:17:28 (7 minutes ago)', type 'NodePort', and status 'Synced' and 'Healthy'. There are also sections for hostnames, links, and events.

Here is the deployment file.

The screenshot shows the Argo UI interface. The deployment file for the 'netflix-app' service is displayed in the main area. The file content is as follows:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     kubectl.kubernetes.io/last-applied-configuration: >
6       {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"netflix-app","argocd.argoproj.io/instance":"netflix"}, "creationTimestamp":"2024-12-19T14:47:28Z"}, "name": "netflix-app", "namespace": "default", "resourceVersion": "97153", "uid": "4756aea1-fb0e-4337-bdb1-a8bc46f3fb6"}, "spec:
6   clusterIP: 10.100.54.147
7   clusterIPs:
8     - 10.100.54.147
9   externalTrafficPolicy: Cluster
10  internalTrafficPolicy: Cluster
11  ipFamilyPolicy: SingleStack
12  ports:
13    - nodePort: 30007
14      port: 80
15      protocol: TCP
16      targetPort: 80
17  selector:
18    app: netflix-app
19    sessionAffinity: None
20    type: NodePort
21  status:
22    loadBalancer: {}
```

Now copy the public IP along with the port number as 30007

