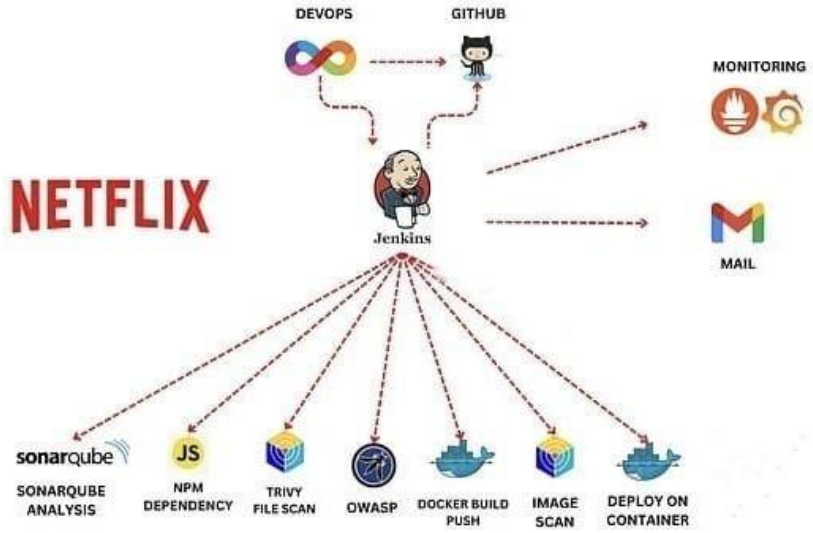
Netflix Clone CI-CD



Project created by : Janhavi Bagul
Link : https://github.com/JanhaviBagul315/Netflix_clone

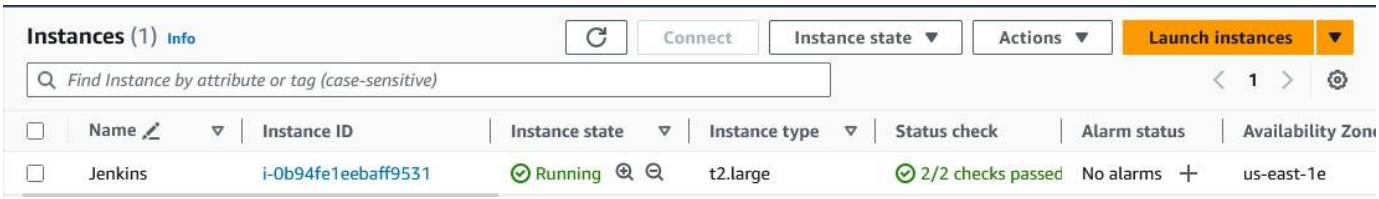
Project Overview

I will be deploying a Netflix clone. I will be using Jenkins as a CICD tool and deploying our application on a Docker container and I will monitor the Jenkins using Grafana, Prometheus and Node exporter.

Project Steps

- Step 1 — Launch an Ubuntu(22.04) T2 Large Instance
- Step 2 — Install Jenkins, Docker and Trivy. Create a Sonarqube Container using Docker.
- Step 3 — Create a TMDB API Key.
- Step 4 — Install Prometheus and Grafana On the new Server.
- Step 5 — Install the Prometheus Plugin and Integrate it with the Prometheus server.
- Step 6 — Email Integration With Jenkins and Plugin setup.
- Step 7 — Install Plugins like JDK, Sonarqube Scanner, Nodejs, and OWASP Dependency Check.
- Step 8 — Create a Pipeline Project in Jenkins
- Step 9 — Install OWASP Dependency Check Plugins
- Step 10 — Docker Image Build and Push
- Step 11 — Deploy the image using Docker
- Step 12 — Access the Netflix app on the Browser.
- Step 13 — Terminate the AWS EC2 Instances.

Step 1 : Launch ubuntu instance t2.large



Step 2 : Login to the Instance

```
ubuntu@ip-172-31-49-232: ~
login as: ubuntu
Authenticating with public key "Linuxkey"
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

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

  System information as of Sun Nov 19 08:02:26 UTC 2023

  System load:  0.0087890625      Processes:             113
  Usage of /:   5.4% of 28.89GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for eth0: 172.31.49.232
  Swap usage:   0%
```

Step 3 : Create one shell script file, bcz of we need to install Jenkins on this server

```
ubuntu@ip-172-31-49-232: ~
ubuntu@ip-172-31-49-232:~$ sudo vi jenkins.sh
```

Step 4 : Add the Jenkins download steps in the script file

```
#!/bin/bash
sudo apt update -y
#sudo apt upgrade -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee
/etc/apt/keyrings/adoptium.asc
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc]
https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION_CODENAME/{print$2}'
/etc/os-release) main" | tee /etc/apt/sources.list.d/adoptium.list
sudo apt update -y
sudo apt install temurin-17-jdk -y
/usr/bin/java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
                  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
                  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
                                /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
sudo systemctl status jenkins
```

Step 5 : Give Permission to the file : (**sudo chmod 777 jenkins.sh**)

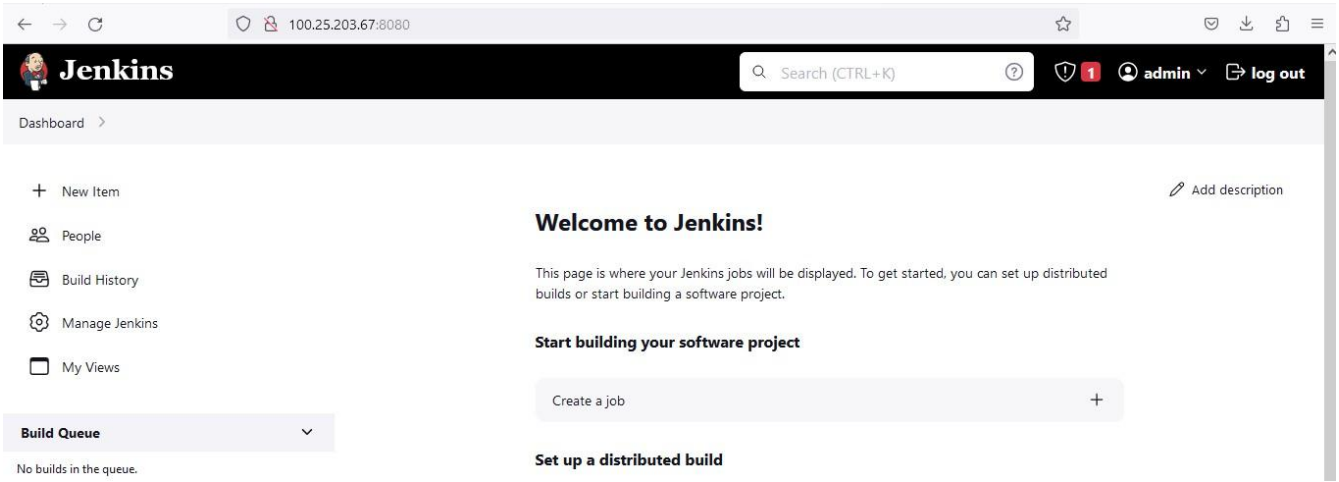
```
ubuntu@ip-172-31-49-232:~$ sudo chmod 777 jenkins.sh
```

Step 6 : Then run the script file (**./ Jenkins.sh**)

```
ubuntu@ip-172-31-49-232:~$ ./jenkins.sh
```

Step 7 : Copy the server ip with port no.8080 and paste in google u get the Jenkins page

Step 8 : Jenkins Installed successfully



Step 9 : Then Install docker on the same server

Commands
sudo apt-get update sudo apt-get install docker.io -y sudo usermod -aG docker $USER #my case is ubuntu newgrp docker sudo chmod 777 /var/run/docker.sock

Step 10 : Then run the sonar in container **(docker run -d --name sonar -p 9000:9000 sonarqube:lts- community)**

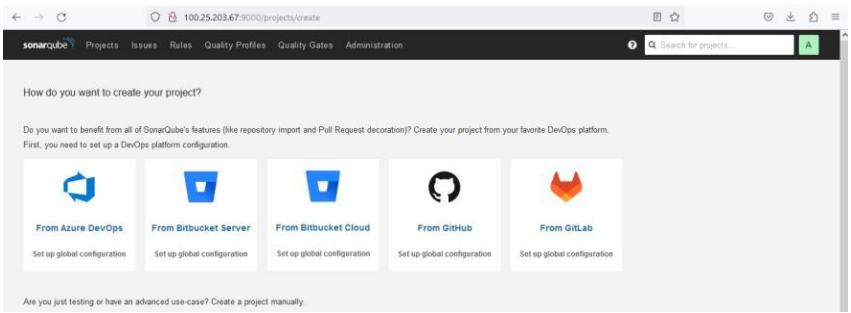
```
ubuntu@ip-172-31-49-232: ~
ubuntu@ip-172-31-49-232:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

Step 11 : Sonar Container created successfully

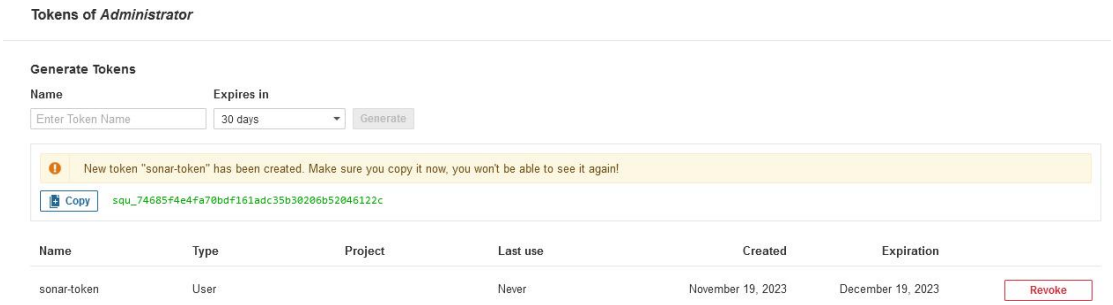
```
ubuntu@ip-172-31-49-232:~$ docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED              STATUS              PORTS                                       NAMES
1c1a6e94cbf2   sonarqube:lts-community   "/opt/sonarqube/dock…"   About a minute ago   Up About a minute   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp   sonar
ubuntu@ip-172-31-49-232:~$
```

Step 12 : Copy the server ip with port no 9000 u get the sonar page (username and pwd= admin)

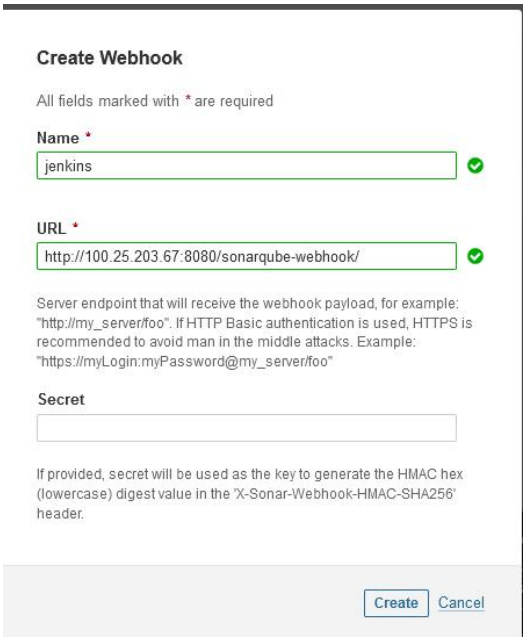
Step 13 : Sonar created successfully

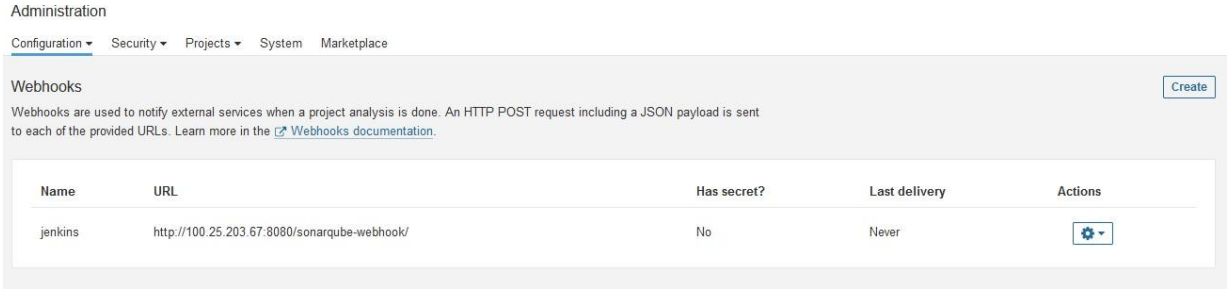


Step 14 : Create token in sonar (path – Administration/security/update token/create token)

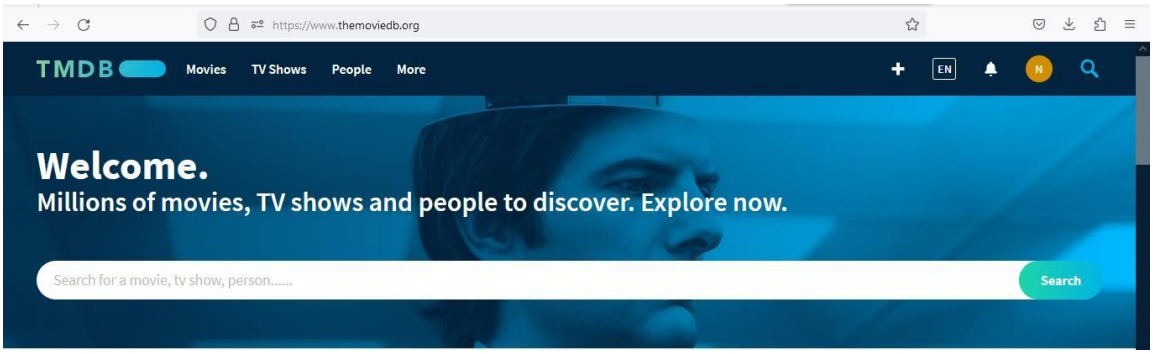


Step 15 : Create webhook in sonar (path – Configuration/webhooks/create webhooks, name = Jenkins and URL = Copy and paste the Jenkins URL and add /sonarqube-webhook/)

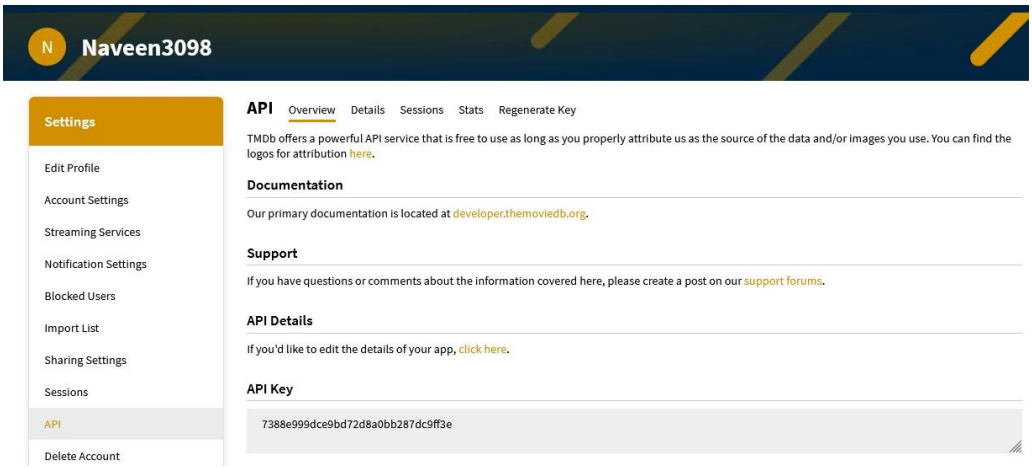




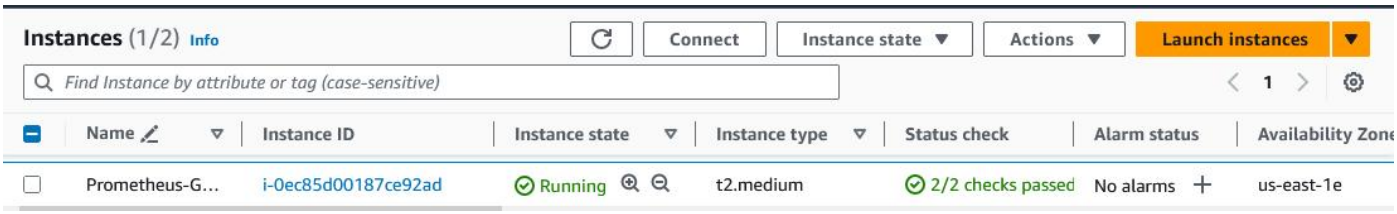
Step 16 : Go to google and search TMDB and click on first link



Step 17 : Create API (Path – click profile icon/setting/API/create API)



Step 18 : Create one more new server name of Prometheus/grafana



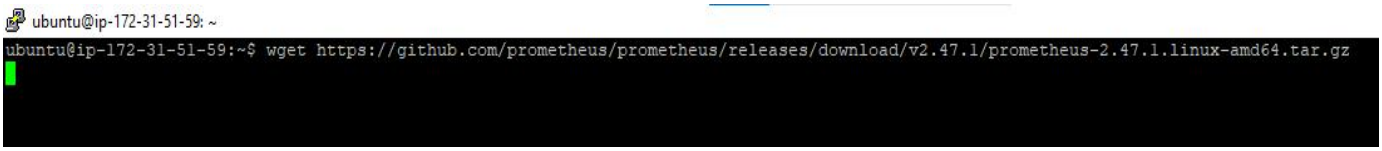
Step 19 : Instance created successfully

Step 20 : To create a system user or system account, run the following command:

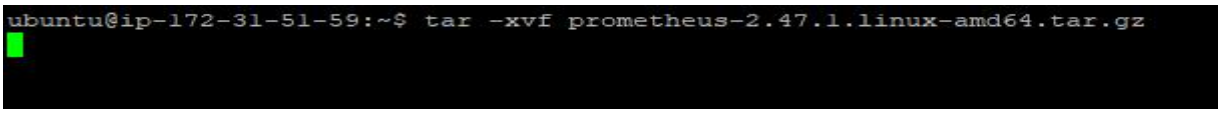
```
sudo useradd \
    --system \
    --no-create-home \
    --shell /bin/false prometheus
```

Step 21 : Download Prometheus :

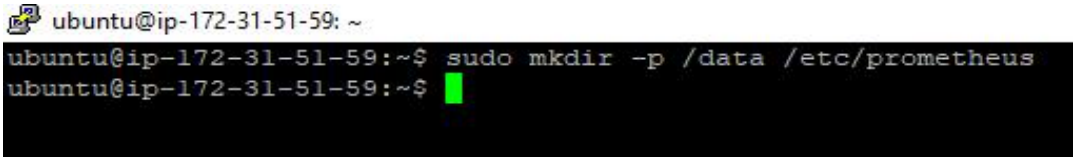
(wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz)



Step 22 : Untar the Prometheus file (**tar -xvf prometheus-2.47.1.linux-amd64.tar.gz**)



Step 23 : Create one Prometheus directory under etc directory (**sudo mkdir -p /data /etc/Prometheus**)



Step 24 : Enter into the Prometheus directory **(cd prometheus-2.47.1.linux-amd64/)**

```
ubuntu@ip-172-31-51-59:~$ cd prometheus-2.47.1.linux-amd64/
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ ls
LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$
```

Step 25 : Move the Prometheus & promtool file to /usr/local/bin

Move the console & console libraries and prometheus.yml file to etc/Prometheus

(sudo mv prometheus promtool /usr/local/bin/

sudo mv consoles/ console_libraries/ /etc/prometheus/

sudo mv prometheus.yml /etc/prometheus/prometheus.yml)

```
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ sudo mv prometheus promtool /usr/local/bin/
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ sudo mv consoles/ console_libraries/ /etc/prometheus/
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ sudo mv prometheus.yml /etc/prometheus/prometheus.yml
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$
```

Step 26 : To avoid permission issues, you need to set the correct ownership for the /etc/prometheus/ and data directory. **(sudo chown -R prometheus:prometheus /etc/prometheus/ /data/)**

```
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$
```

Step 27 : You can delete the archive and a Prometheus folder when you are done.

(cd ..

rm -rf prometheus-2.47.1.linux-amd64.tar.gz)

```
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ cd ..
ubuntu@ip-172-31-51-59:~$ ls
prometheus-2.47.1.linux-amd64  prometheus-2.47.1.linux-amd64.tar.gz
ubuntu@ip-172-31-51-59:~$ rm -rf prometheus-2.47.1.linux-amd64 prometheus-2.47.1.linux-amd64.tar.gz
```

Step 28 : We're going to use Systemd, which is a system and service manager for Linux operating systems. For that, we need to create a Systemd unit configuration file.

(sudo vim /etc/systemd/system/prometheus.service)

```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/systemd/system/prometheus.service
```

ubuntu@ip-172-31-51-59: ~

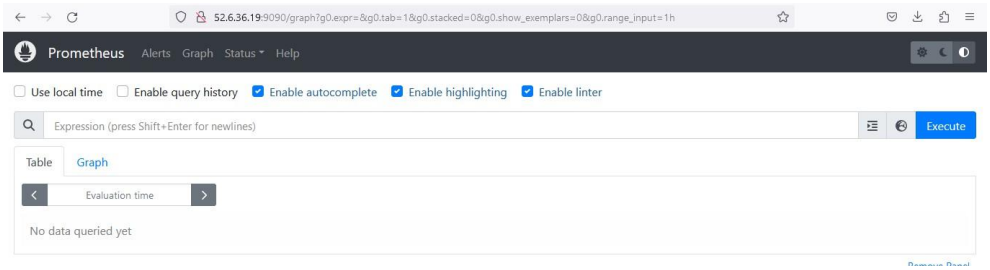
```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/data \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-lifecycle

[Install]
WantedBy=multi-user.target
```

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus  \
 --config.file=/etc/prometheus/prometheus.yml  \
 --storage.tsdb.path=/data  \
 --web.console.templates=/etc/prometheus/consoles  \
 --
web.console.libraries=/etc/prometheus/console_libraries
\
 --web.listen-address=0.0.0.0:9090  \
 --web.enable-lifecycle

[Install]
WantedBy=multi-user.target
```

Step 30 : Start the Prometheus service (**sudo systemctl enable Prometheus**)

```
ubuntu@ip-172-31-51-59:~$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
ubuntu@ip-172-31-51-59:~$ sudo systemctl start prometheus
ubuntu@ip-172-31-51-59:~$
```

Step 31 : Copy the Prometheus server ip with port no.9090 u get Prometheus page



Step 32 : To create a system user or system account, run the following command:

```
sudo useradd \
    --system \
    --no-create-home \
    --shell /bin/false node_exporter
```

Code

```
sudo useradd \
  --system \
  --no-create-home \
  --shell /bin/false node_exporter
```

Step 33 : Download node exporter (wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux- amd64.tar.gz)

```
ubuntu@ip-172-31-51-59:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
```

Step 34 : Untar the node exporter file (tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz)

```
ubuntu@ip-172-31-51-59: ~
ubuntu@ip-172-31-51-59:~$ ls
node_exporter-1.6.1.linux-amd64.tar.gz
ubuntu@ip-172-31-51-59:~$ tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
```

Step 35 : Move the node exporter file

```
sudo mv \
 node_exporter-1.6.1.linux-amd64/node_exporter  \
 /usr/local/bin/
```

```
ubuntu@ip-172-31-51-59:~$ ls
node_exporter-1.6.1.linux-amd64  node_exporter-1.6.1.linux-amd64.tar.gz
ubuntu@ip-172-31-51-59:~$ sudo mv \
  node_exporter-1.6.1.linux-amd64/node_exporter \
  /usr/local/bin/
```

Step 36 : After moving the node exporter file remove the tar file (rm -rf node_exporter*)

```
ubuntu@ip-172-31-51-59: ~
ubuntu@ip-172-31-51-59:~$ rm -rf node_exporter*
ubuntu@ip-172-31-51-59:~$
```

Step 37 : We're going to use Systemd, which is a system and service manager for Linux operating systems. For that, we need to create a Systemd unit configuration file.

(sudo vim /etc/systemd/system/node_exporter.service)

```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/systemd/system/node_exporter.service
```

Step 38 : Add script in the file

```
ubuntu@ip-172-31-51-59: ~
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter \
    --collector.logind

[Install]
WantedBy=multi-user.target
~
```

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target


StartLimitIntervalSec=500
StartLimitBurst=5


[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter \
  --collector.logind
[Install]
WantedBy=multi-user.target
```

Step 39 : Start the node exporter

service (sudo systemctl enable

node_exporter sudo systemctl start

node_exporter)

```
ubuntu@ip-172-31-51-59:~$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
ubuntu@ip-172-31-51-59:~$ sudo systemctl start node_exporter
ubuntu@ip-172-31-51-59:~$
```

Step 40 : Add the node exporter job in Prometheus.yml

file (sudo vim /etc/prometheus/prometheus.yml)

```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/prometheus/prometheus.yml
```

Step 41 : Node exporter job with port no.9100, copy this code and paste in Prometheus.yaml file

```
  - job_name: "Node-Exporter"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["52.6.36.19:9100"]
```

```
- job_name: node_export

  static_configs:

    - targets: ["localhost:9100"]
```

Step 42 : Start and reload the service

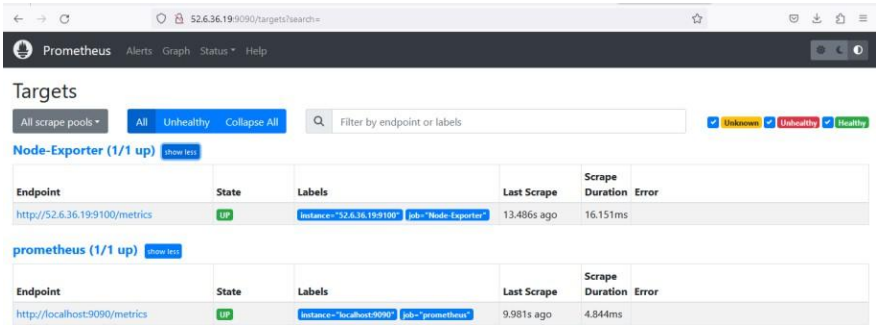
(promtool check config

/etc/prometheus/prometheus.yml) (curl -X POST

http://localhost:9090/-/reload)

```
ubuntu@ip-172-31-51-59:~$ promtool check config /etc/prometheus/prometheus.yml
Checking /etc/prometheus/prometheus.yml
 SUCCESS: /etc/prometheus/prometheus.yml is valid prometheus config file syntax

ubuntu@ip-172-31-51-59:~$ curl -X POST http://localhost:9090/-/reload
ubuntu@ip-172-31-51-59:~$
```

Step 43 : Go to Prometheus/target u saw the node exporter job



Step 44 : Download grafana

(sudo apt-get install -y apt-transport-https software-properties-common)

```
ubuntu@ip-172-31-51-59:~$ sudo apt-get install -y apt-transport-https software-properties-common
```

Step 45 : Download gpg key (wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -)

```
ubuntu@ip-172-31-51-59:~$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
ubuntu@ip-172-31-51-59:~$
```

Step 46 : Add this repository for stable releases.

(echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list)

```
ubuntu@ip-172-31-51-59:~$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/oss/deb stable main
ubuntu@ip-172-31-51-59:~$
```

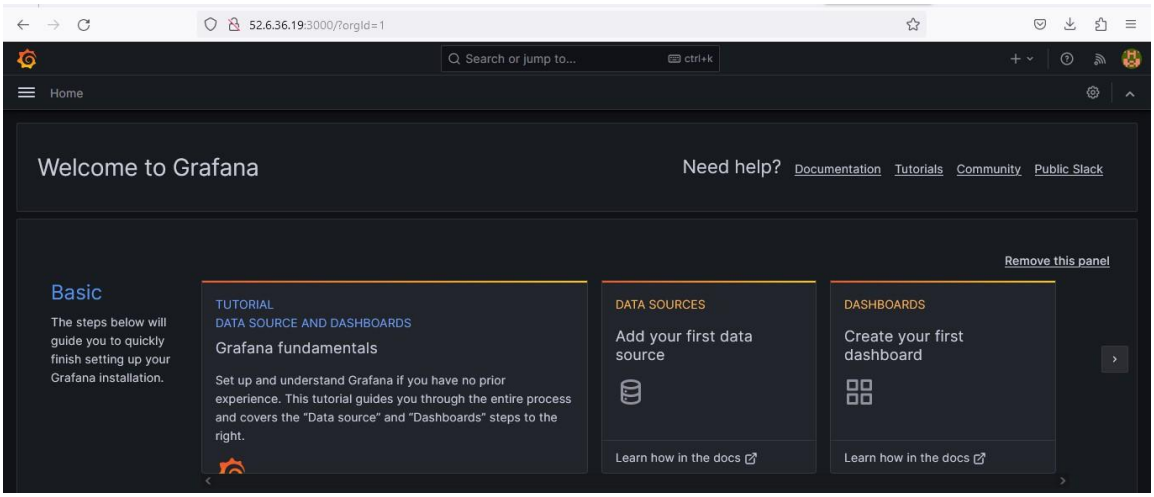
Step 47 : Download Grafana by the below command

```
sudo apt-get update
```

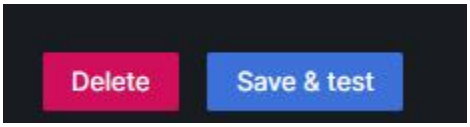
```
sudo apt-get -y install grafana
```

```
sudo systemctl start grafana-server
```

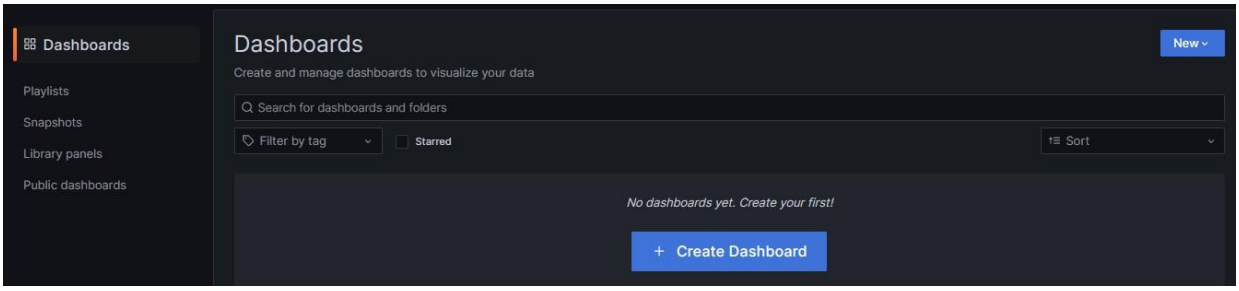
Step 48 : Copy Ip with port no.3000 and paste in chrome u get grafana page after u get the page click on data sources



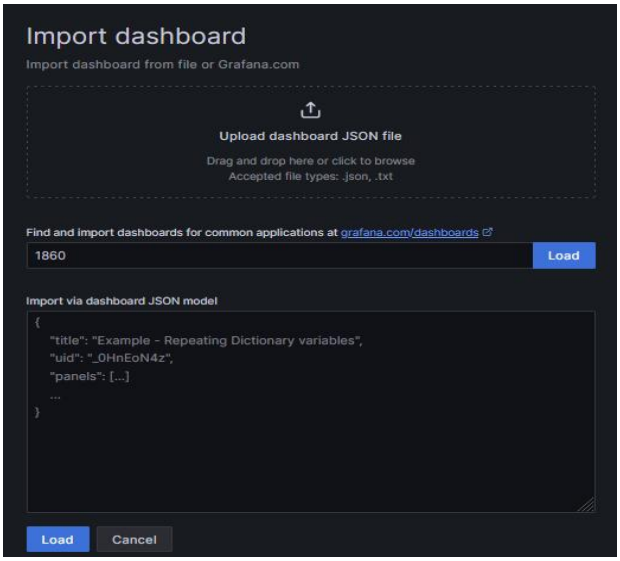
Step 49 : And Click on Prometheus add the Prometheus URL and click on save test



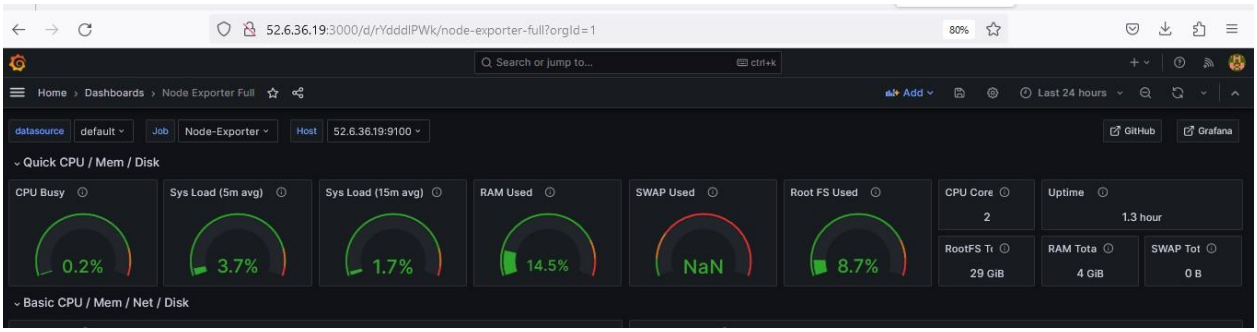
Step 50 : Click on new and click on Import dashboard



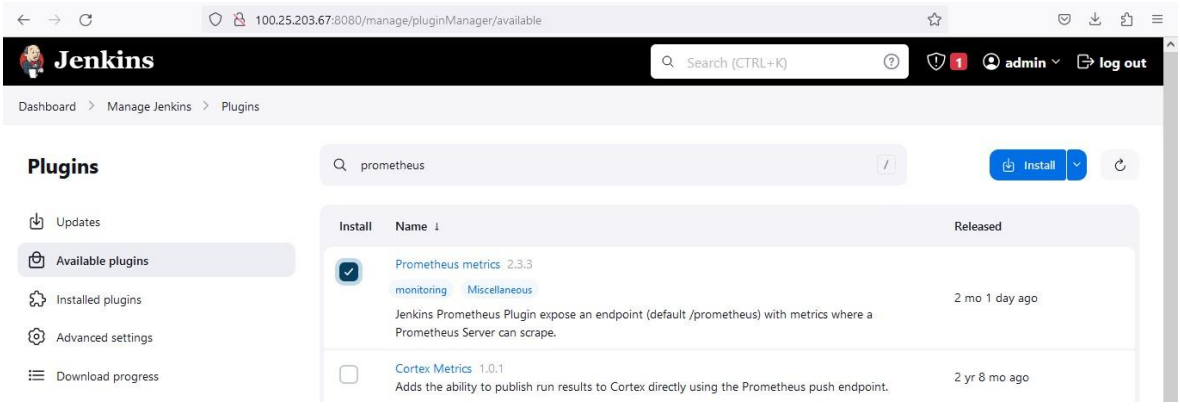
Step 51 : Enter no.1860 and click on load



Step 52 : U get monitoring page of node exporter



Step 53 : Go to Jenkins download Prometheus plugin



Step 54 : Add the Jenkins job in Prometheus.yaml (sudo vim /etc/prometheus/prometheus.yml)

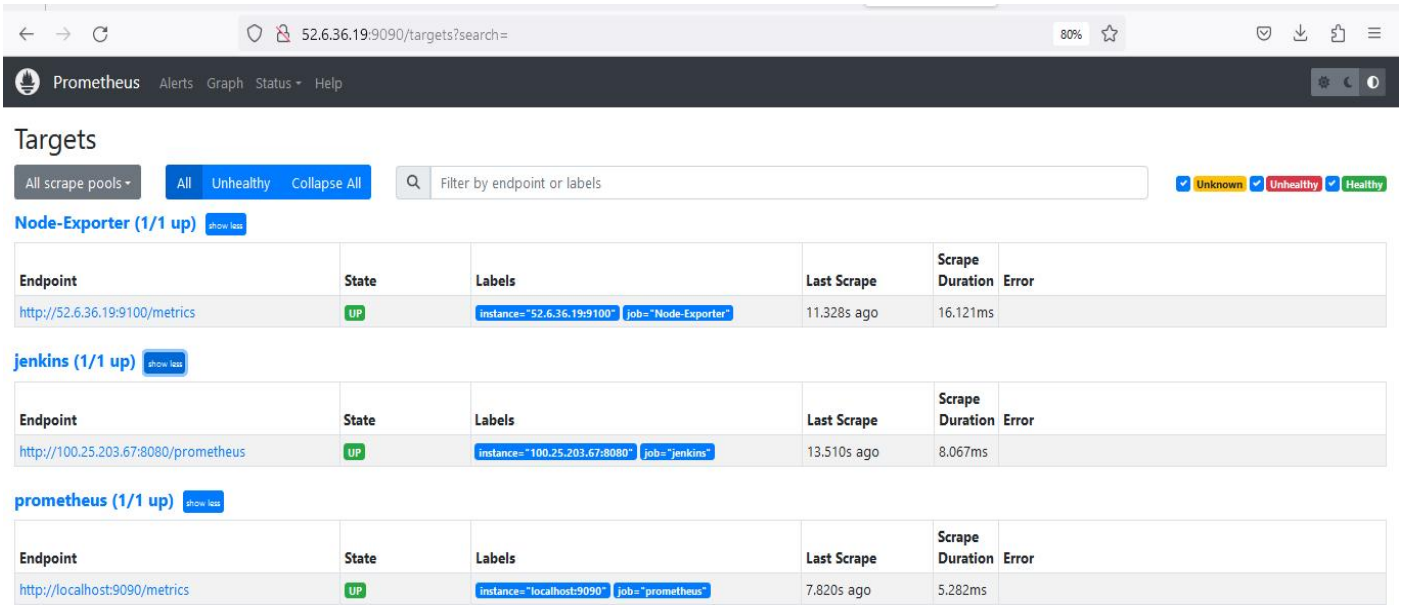
```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/prometheus/prometheus.yml
```

Step 55 : Add this Jenkins job in Prometheus.yml

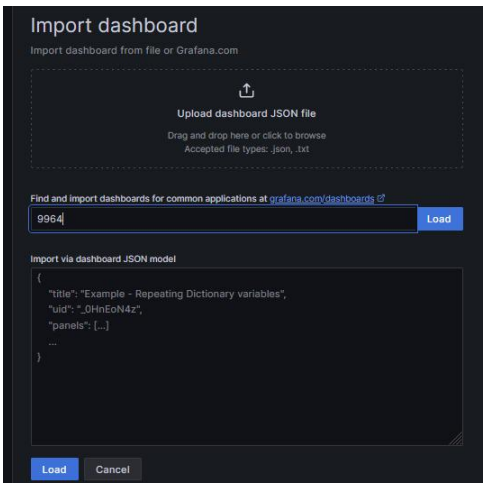
```
- job_name: 'jenkins'
  metrics_path: '/prometheus'
  static_configs:
    - targets: ['100.25.203.67:8080']
```

```
- job_name: 'jenkins'
  metrics_path: '/prometheus'
  static_configs:
      - targets: ['<jenkins-ip>:8080']
```

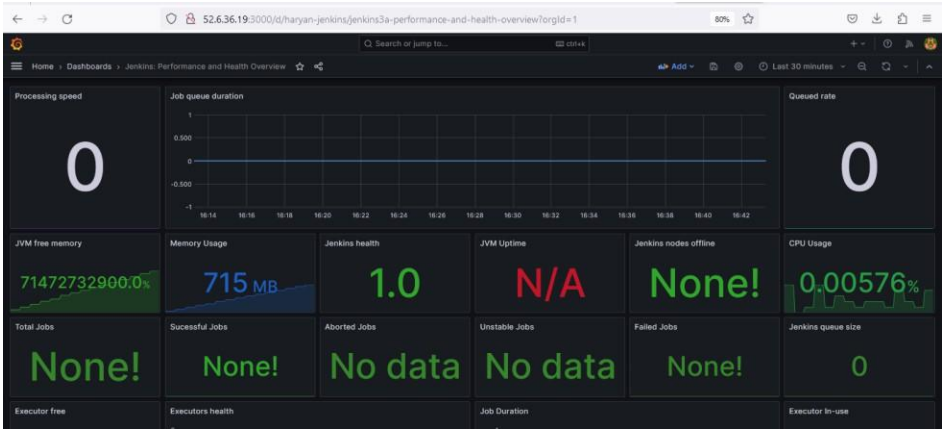
Step 56 : Go to Prometheus/Status/target u get Jenkins info



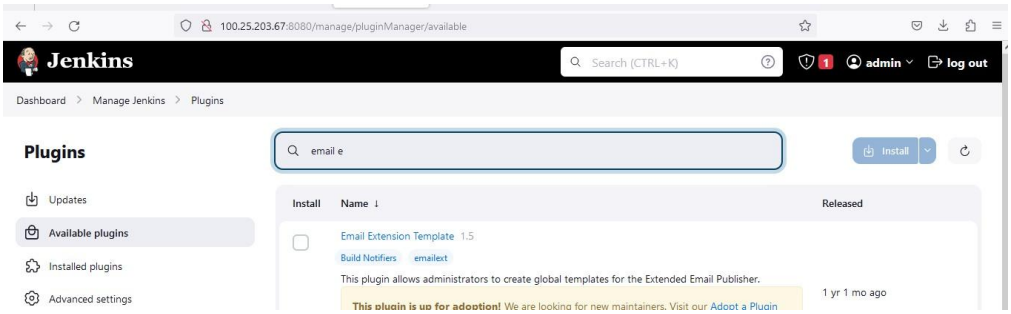
Step 57 : Click on Import dashboard and enter no 9964 and click load



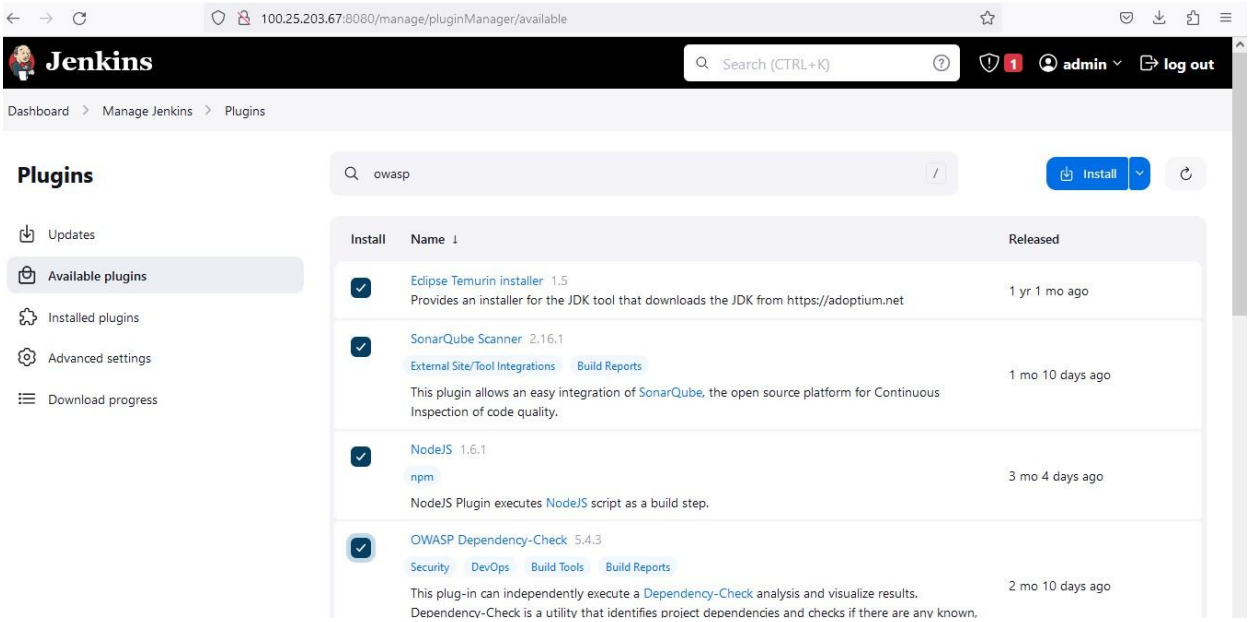
Step 58 : Now you will see the Detailed overview of Jenkins



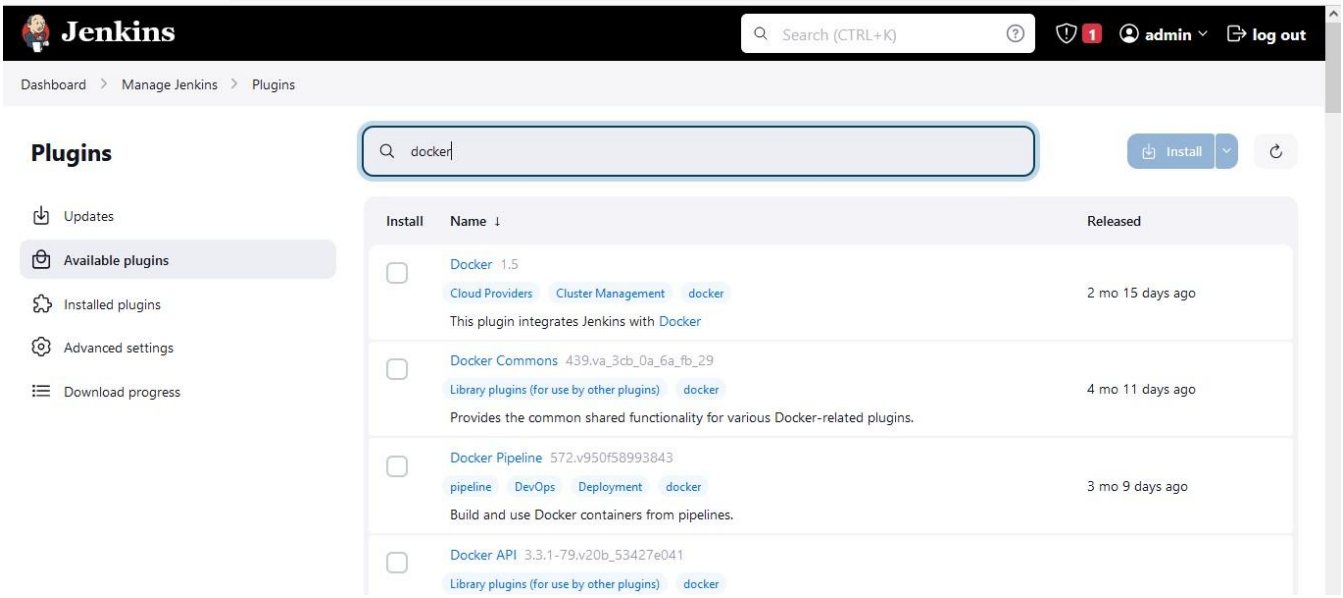
Step 59 : Download the email plugin in jenkins



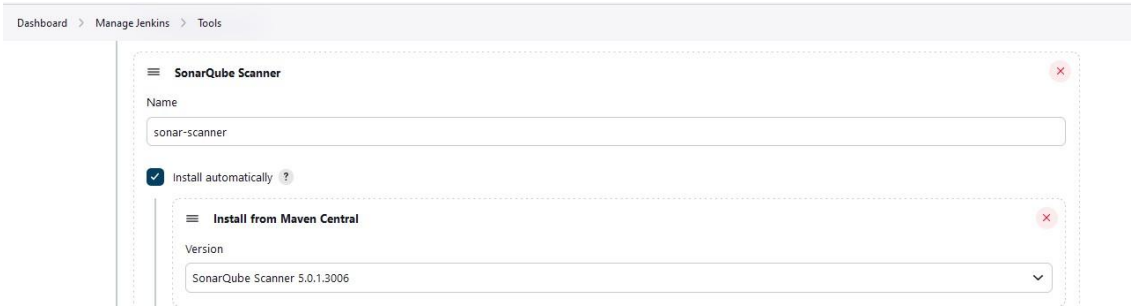
Step 61 : Download plugins like Eclipse Temurin Installer, sonarqube scanner, Node js, OWASP Depency check



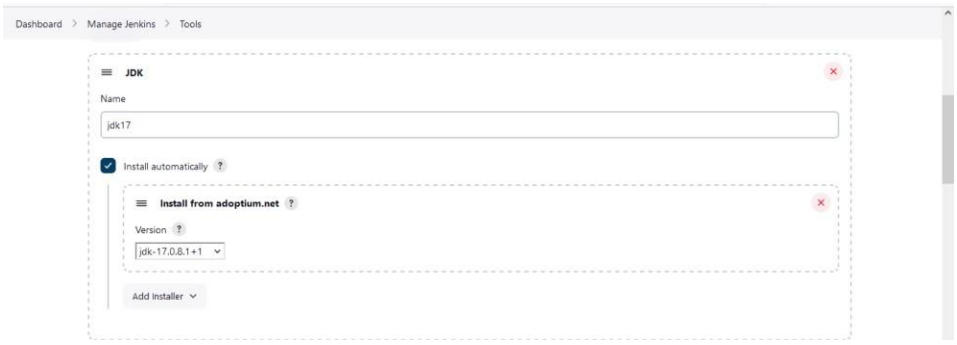
Step 62 : Download docker base all plugin (Total 5 Plugin)



Step 63 : Add the sonar server info in Jenkins tools page



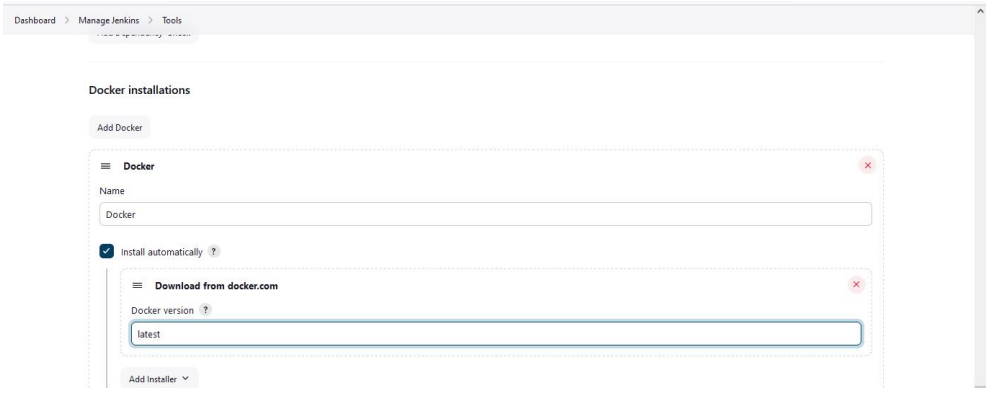
Step 64 : Add the jdk info in Jenkins tools page



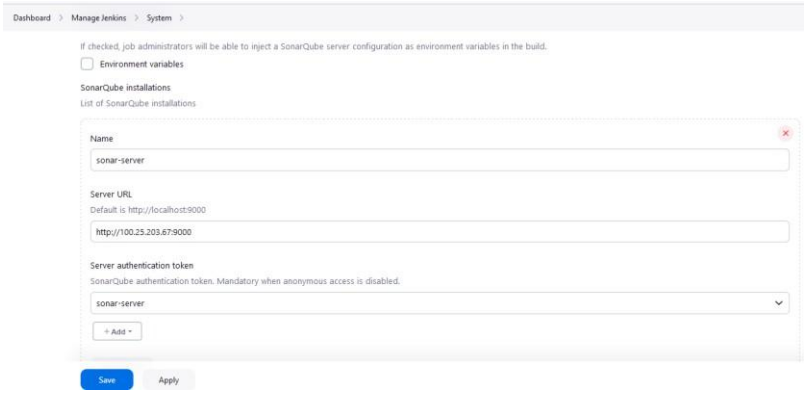
Step 65 : Add the nodejs info in Jenkins tools page

Step 66 : Add the dependency info in Jenkins tools page

Step 67 : Add the docker info in Jenkins tools page



Step 68 : Go to Jenkins/system and integrate sonarqube with jenkins



Step 69 : Then we need to install trivy, so we first of all create trivy shell script file

```
ubuntu@ip-172-31-49-232:~$ vi trivy.sh
```

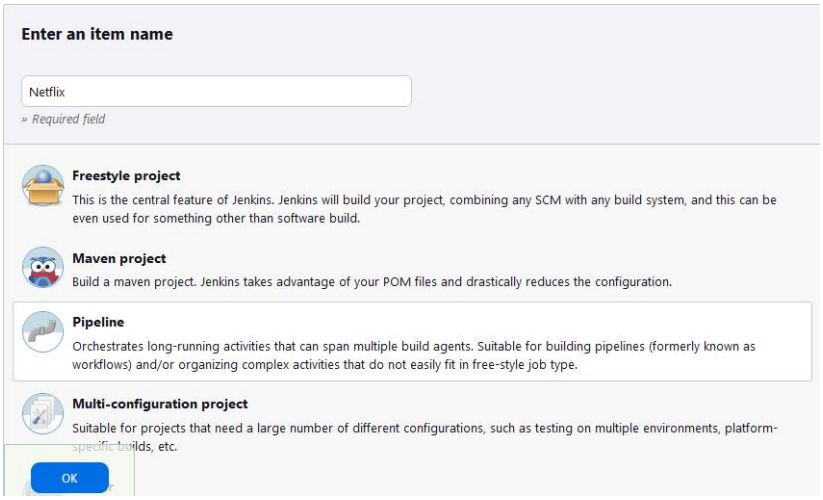
Step 70 : Add the trivy downloaded command in this file

```
sudo apt-get install wget apt-transport-https gnupg lsb-release -y

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] 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 -y
```

Step 71 : Give permission to the file and run the script file (sudo chmod 777 trivy.sh & ./trivy.sh)

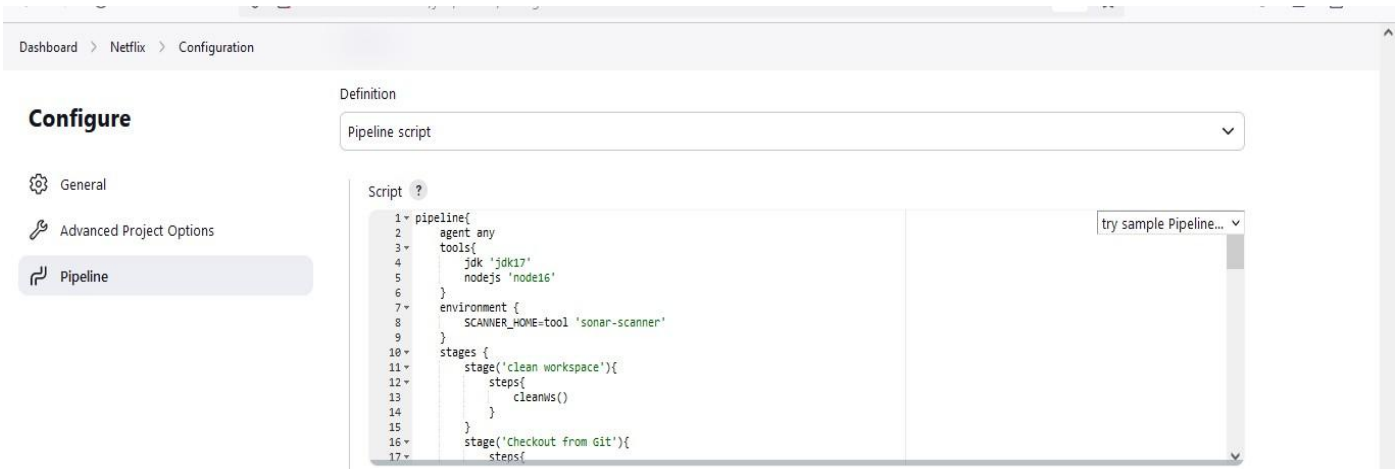
```
ubuntu@ip-172-31-49-232:~$ sudo chmod 777 trivy.sh
ubuntu@ip-172-31-49-232:~$ ./trivy.sh
```

Step 70 : After all this step create one pipeline name of Netflix



Step 71 : Add the script in the pipeline

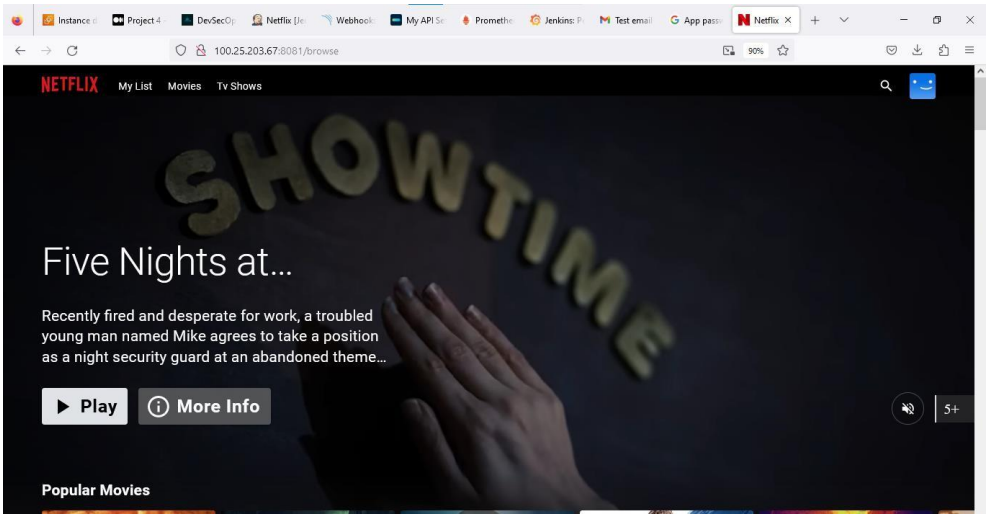
(Pipeline script stored in github : https://github.com/JanhaviBagul315/Netflix_clone/blob/main/Pipeline%20Script)

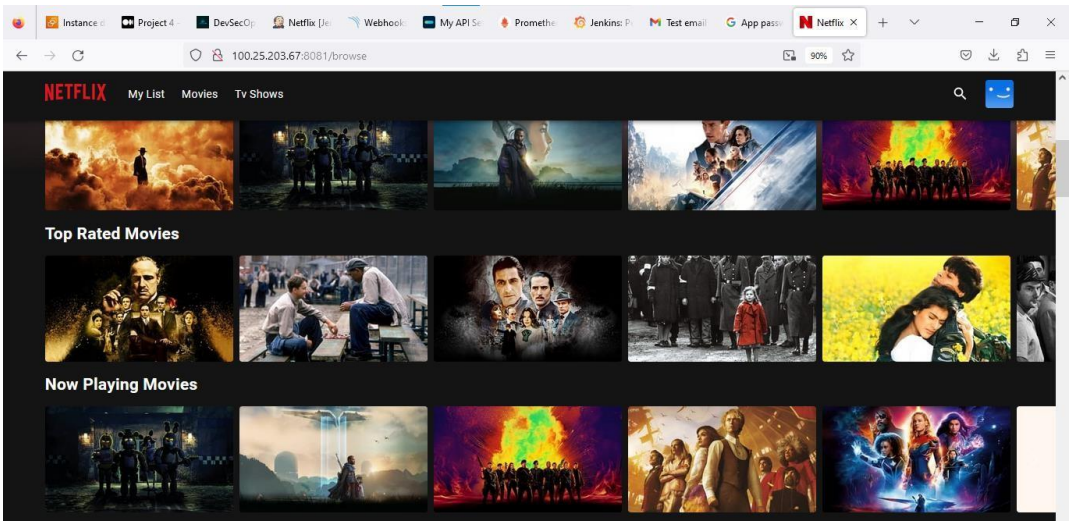


Step 72 : Run the pipeline

Step 73. Pipeline done and all stages are completed successfully

Step 73 : Copy the Ip with port no u get the Netflix page successfully




Netflix
100.25.203.67:8081/browse
NETFLIX
My List
Movies
Tv Shows
Top Rated Movies
Now Playing Movies