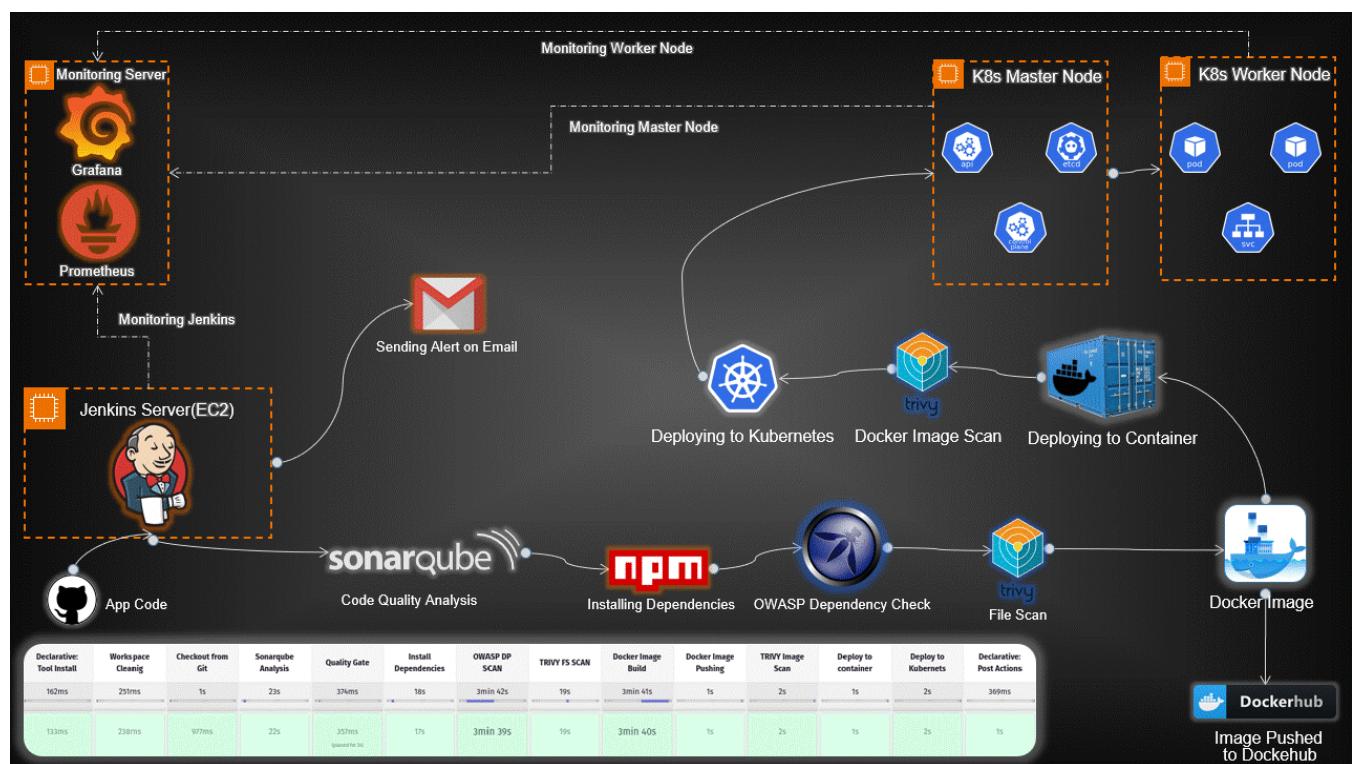


# End-to-End DevSecOps Kubernetes Project

Aman Pathak · [Follow](#)

Published in DevOps.dev

25 min read · Dec 30, 2023

[Listen](#)[Share](#)

## Introduction:

In today's rapidly evolving tech landscape, deploying applications using Kubernetes has become a crucial aspect of modern software development. This guide provides a detailed walkthrough for setting up an end-to-end Kubernetes project, covering everything from infrastructure provisioning to application deployment and monitoring.

## Prerequisites:

Before diving into the implementation, ensure you have the following in place:

1. Basic understanding of Kubernetes concepts.
2. Access to AWS or any other cloud provider for server instances.
3. A TMDB API key for accessing movie databases in your Netflix Clone application.
4. DockerHub account for pushing and pulling Docker images.
5. Gmail account for email notifications.
6. Jenkins, Kubernetes, Docker, and necessary plugins installed.

## **High-Level Overview:**

1. **Infrastructure Setup:** Provisioned servers for Jenkins, Monitoring, and Kubernetes nodes.
2. **Toolchain Integration:** Integrated essential tools like Jenkins, SonarQube, Trivy, Prometheus, Grafana, and OWASP Dependency-Check.
3. **Continuous Integration/Continuous Deployment (CI/CD):** Automated workflows with Jenkins pipelines for code analysis, building Docker images, and deploying applications on Kubernetes.
4. **Security Scanning:** Implemented Trivy and OWASP Dependency-Check to scan for vulnerabilities in code and Docker images.
5. **Monitoring and Visualization:** Set up Prometheus and Grafana for real-time monitoring and visualization of both hardware and application metrics.
6. **Email Notifications:** Configured Jenkins for email alerts based on pipeline results.

You will get the Jenkinsfile and Kubernetes Manifest files along with the Dockerfile. Feel free to modify it accordingly

**Project GitHub Repo-** <https://github.com/AmanPathak-DevOps/Netflix-Clone-K8S-End-to-End-Project>

We need four servers for our today's Project

**Jenkins Server-** On this Server, Jenkins will be installed with some other tools such as sonarqube(docker container), trivy, and kubectl.

**Monitoring Server-** This Server will be used for Monitoring where we will use Prometheus, Node Exporter, and Grafana.

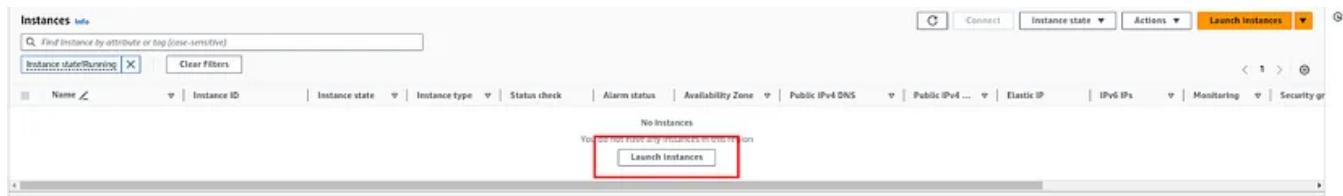
**Kubernetes Master Server-** This Server will be used as the Kubernetes Master Cluster Node which will deploy the applications on worker nodes.

**Kubernetes Worker Server-** This Server will be used as the Kubernetes Worker Node on which the application will be deployed by the master node.

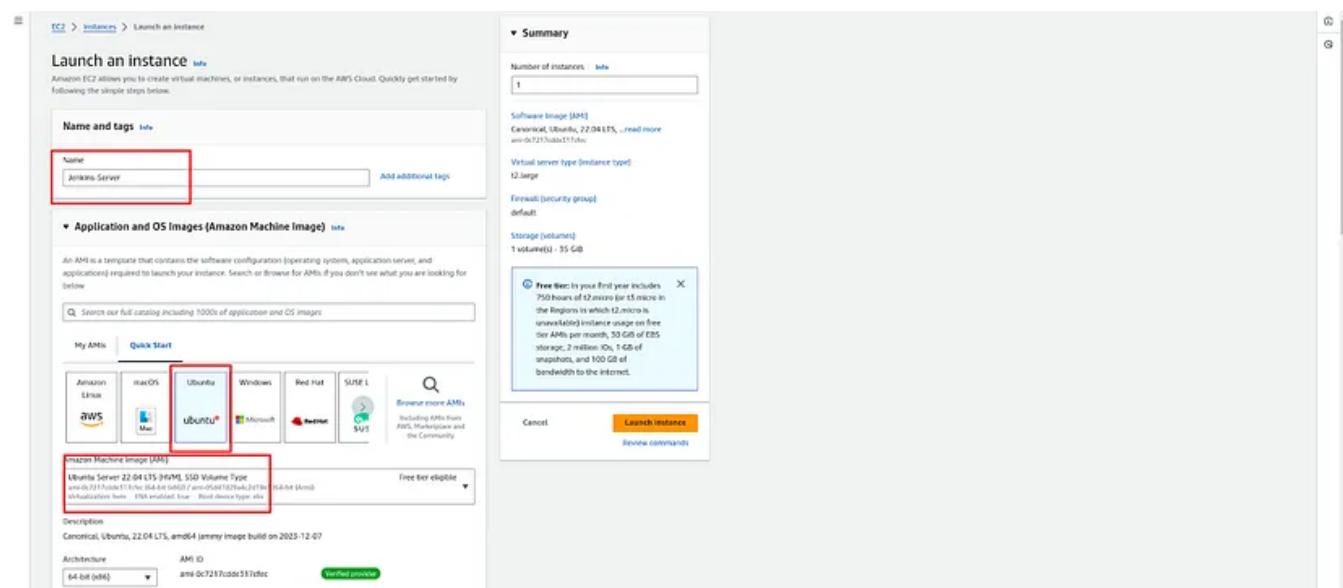
Let's create the following instances.

## Jenkins Server

Click on Launch Instances.

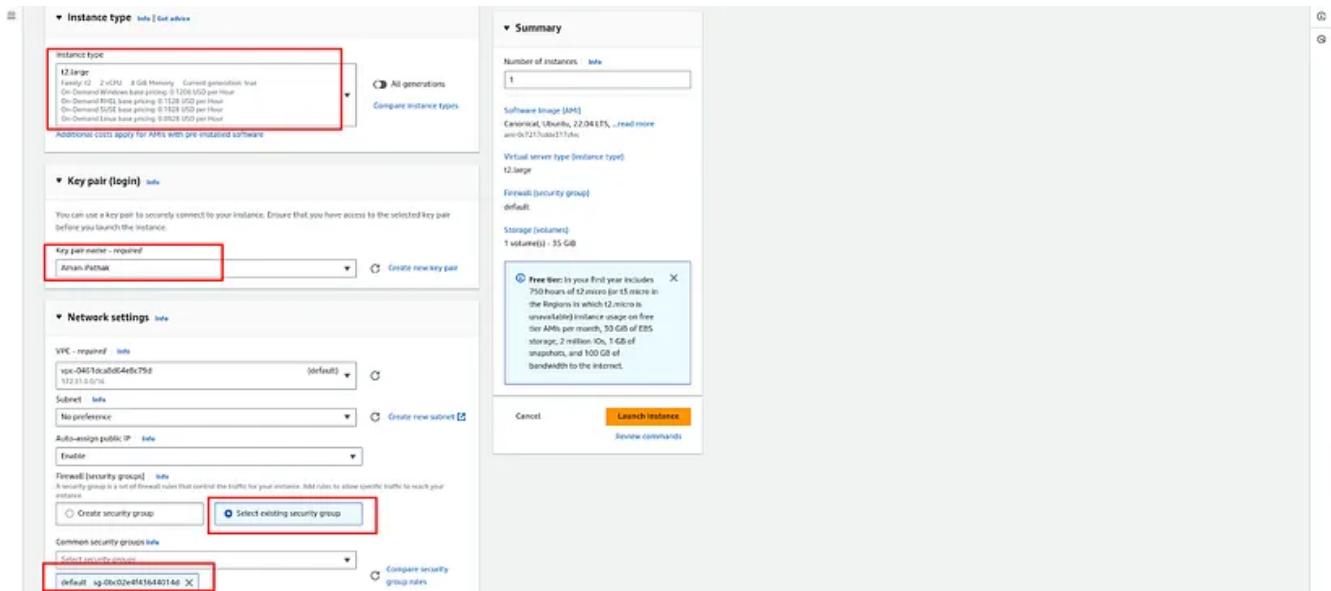


Provide the name of your Jenkins instance, and select the Ubuntu OS 22.04 version.

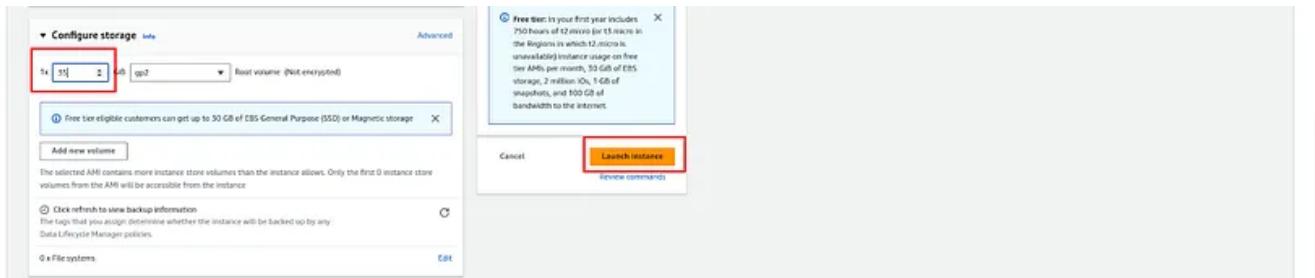


We need to configure multiple things on the Jenkins instance. So, select the t2.large instance type, provide the key or you can create if you want.

Keep the networking things as it is. But make sure to open all inbound and outbound traffic in the selected security groups.

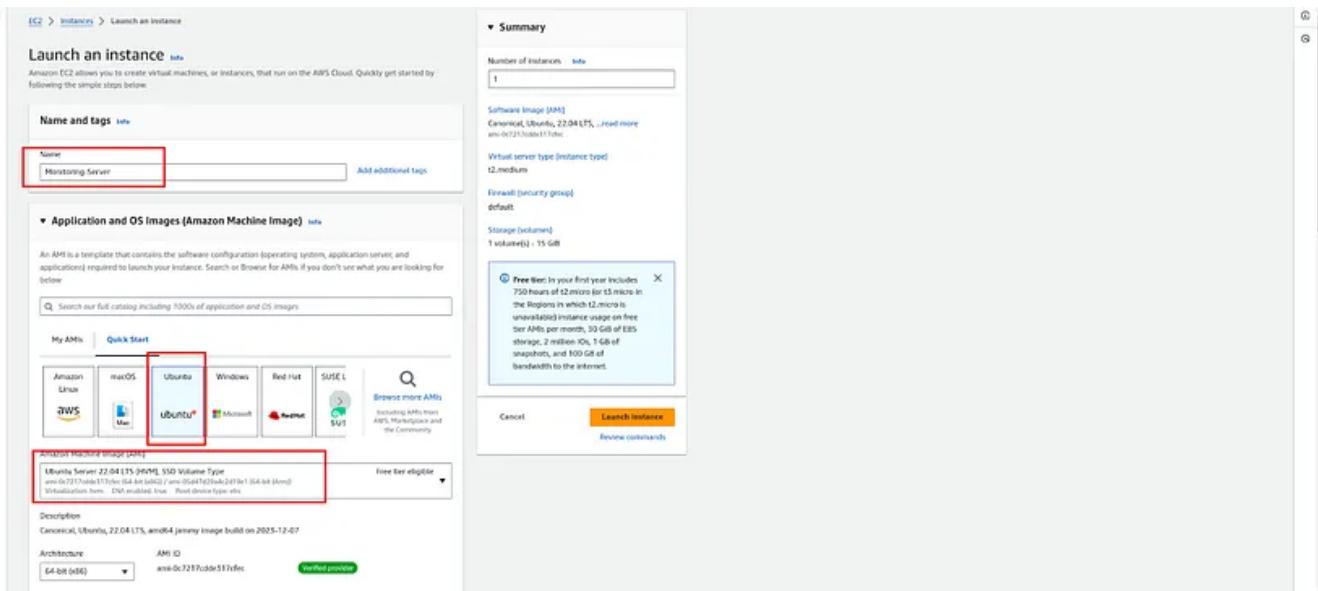


Increase the storage capacity for Jenkins Instance from 8GB to 35GB and click on **Launch Instance**.



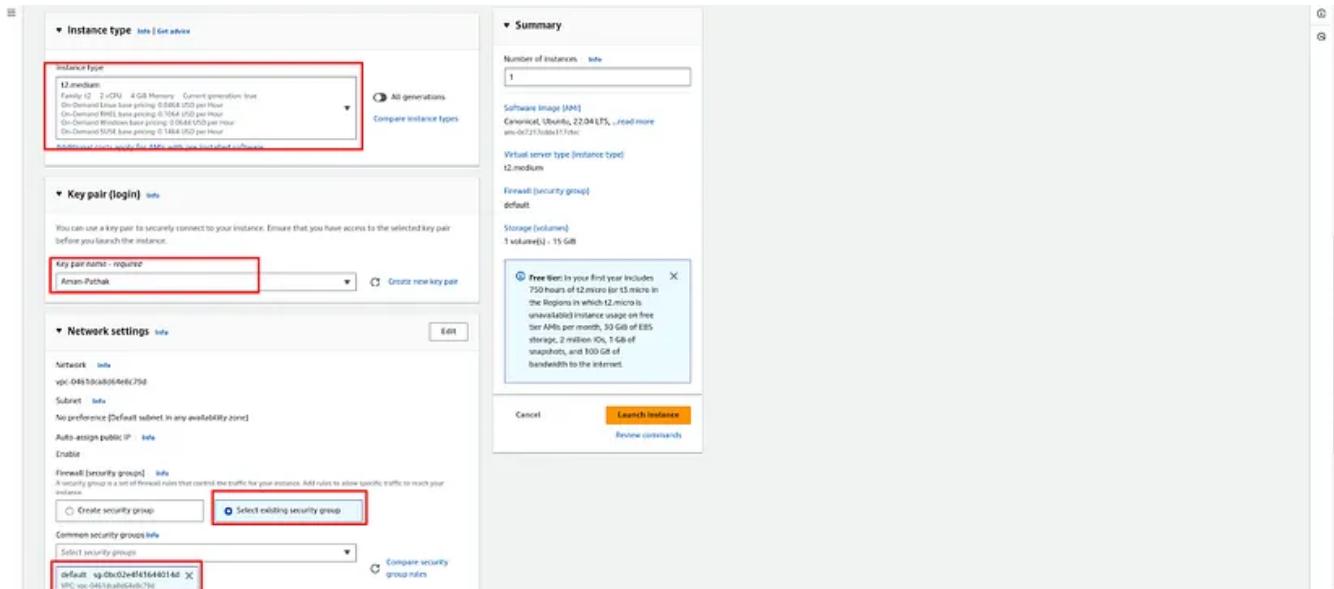
## Monitoring Server

Provide the name of your Monitoring Instance, and select the Ubuntu 22.04 OS.

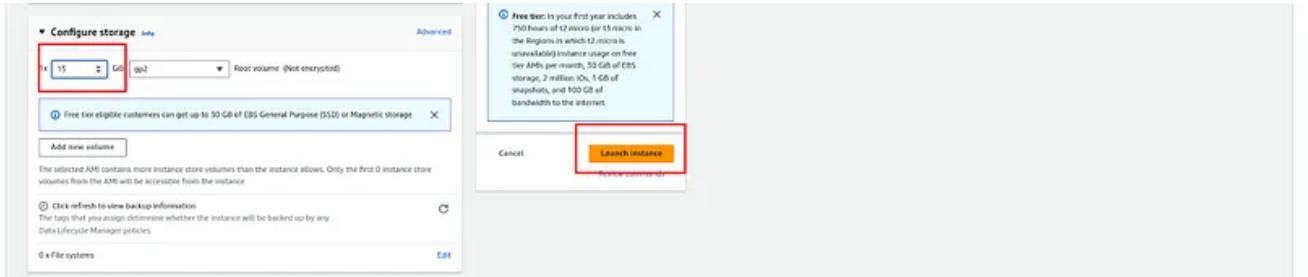


We need to configure the monitoring tools on this instance which needs a minimum of 4GB RAM. So, select the t2.medium instance type, provide the key or you can create if you want.

Keep the networking things as it is. But make sure to open all inbound and outbound traffic in the selected security groups.



Increase the storage capacity for Jenkins Instance from 8GB to 15GB and click on **Launch Instance**.

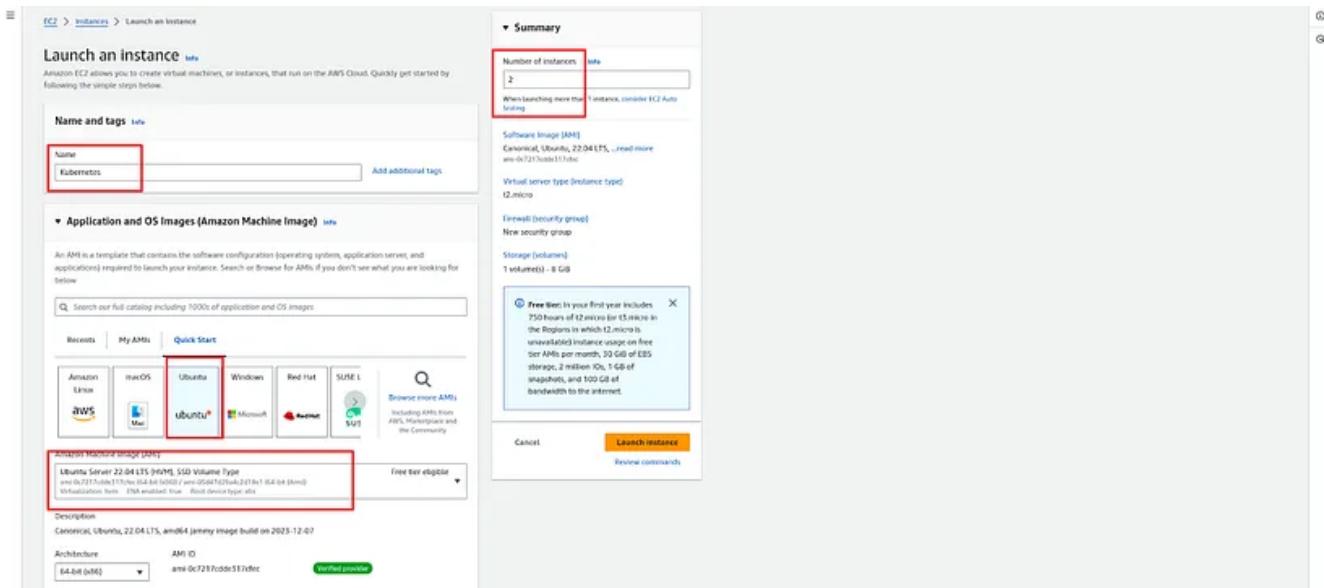


## Kubernetes Master & Worker Node

We have to create two Kubernetes Nodes which need at least 2 CPUs.

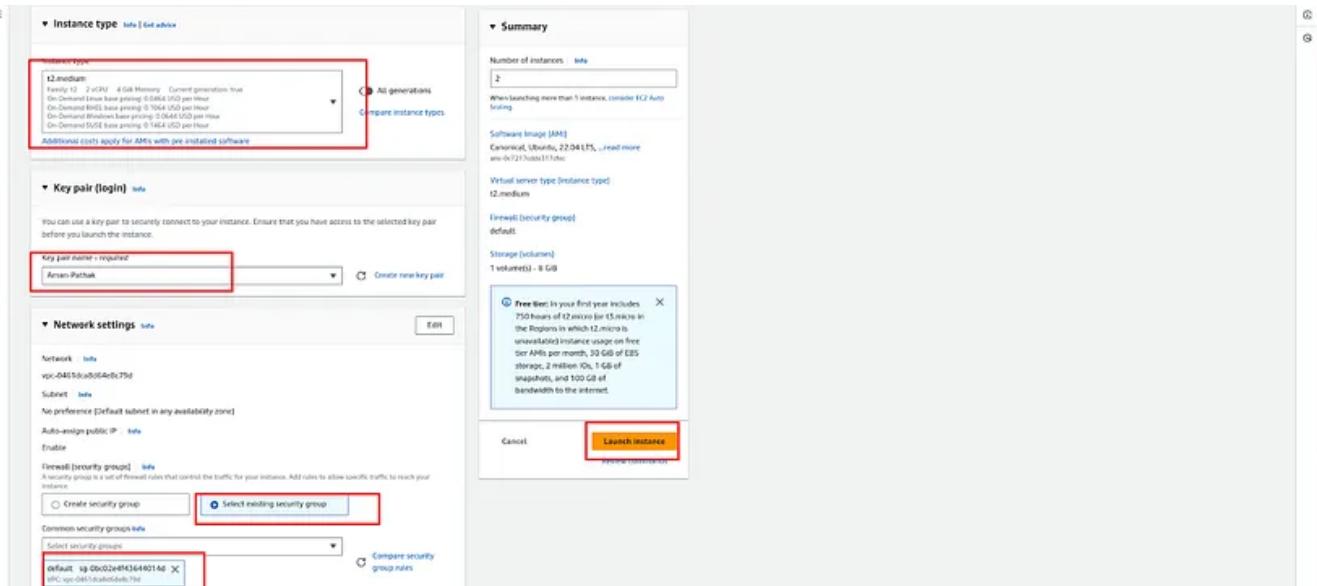
Provide the name of your Kubernetes Master Instance, and select the Ubuntu 22.04 OS.

In the Number of Instances, replace 1 with 2 because we need two Kubernetes Nodes.



Select the t2.medium instance type, provide the key or you can create if you want.

Keep the networking things as it is. But make sure to open all inbound and outbound traffic in the selected security groups then keep the rest of the things as it is and click on Launch Instance.



Rename the Kubernetes Servers and all four servers will look like the below snippet.

Instances (4) <a href="#">Info</a>													
<a href="#">Find instance by attribute or tag (case-sensitive)</a> <span style="float: right;">Instance state ▾ Actions ▾ Launch instances ▾</span>													
<span style="float: left;">Instance state is running</span> <span style="float: right;">Clear filters</span>													
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4	Elastic IP	IPv6 IPs	Monitoring	Security gr	
Kubernetes-Master	i-0373556f080d2125	<span>Running</span> ⓘ ⓘ	t2.medium	<span>Initializing</span>	No alarms +	us-east-1e	ec2-52-91-122-66.com...	52.91.122.66	-	-	disabled	default	
Kubernetes-Worker	i-0345d6640597e55b	<span>Running</span> ⓘ ⓘ	t2.medium	<span>Initializing</span>	No alarms +	us-east-1e	ec2-5-90-84-57.compute...	5.90.84.57	-	-	disabled	default	
Jenkins-Server	i-0aaef5f98bf1f9887	<span>Running</span> ⓘ ⓘ	t2.large	<span>2/2 checks passed</span>	No alarms +	us-east-1e	ec2-54-207-155-151.co...	54.207.155.151	-	-	disabled	default	
Monitoring-Server	i-0620098bd5ee812d	<span>Running</span> ⓘ ⓘ	t2.medium	<span>Initializing</span>	No alarms +	us-east-1e	ec2-54-152-127-65.co...	54.152.127.65	-	-	disabled	default	

## Log in to the Jenkins Server

```

amanspathak@pop-os:~$ Downloads$ ssh -i "Aman-Pathak.pem" ubuntu@ec2-34-207-155-151.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-155-151.compute-1.amazonaws.com (34.207.155.151)' can't be established.
ED25519 key fingerprint is SHA256:VDNhbgoH5oddmNH703RgPSc/vdHtIZF2j8B9LE4Jlc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-155-151.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

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

 System information as of Wed Dec 27 13:57:18 UTC 2023

System load: 0.16015625 Processes: 105
Usage of /: 4.6% of 33.74GB Users logged in: 0
Memory usage: 2% IPV4 address for eth0: 172.31.59.9
Swap usage: 0%

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-172-31-59-9:~$ 

```

## Download Open JDK and Jenkins

```

# Intalling Java
sudo apt update -y
sudo apt install openjdk-11-jre -y
java --version

```

```
# Installing Jenkins
curl -fsSL https://pkg.jenkins.io/debian/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 binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
```

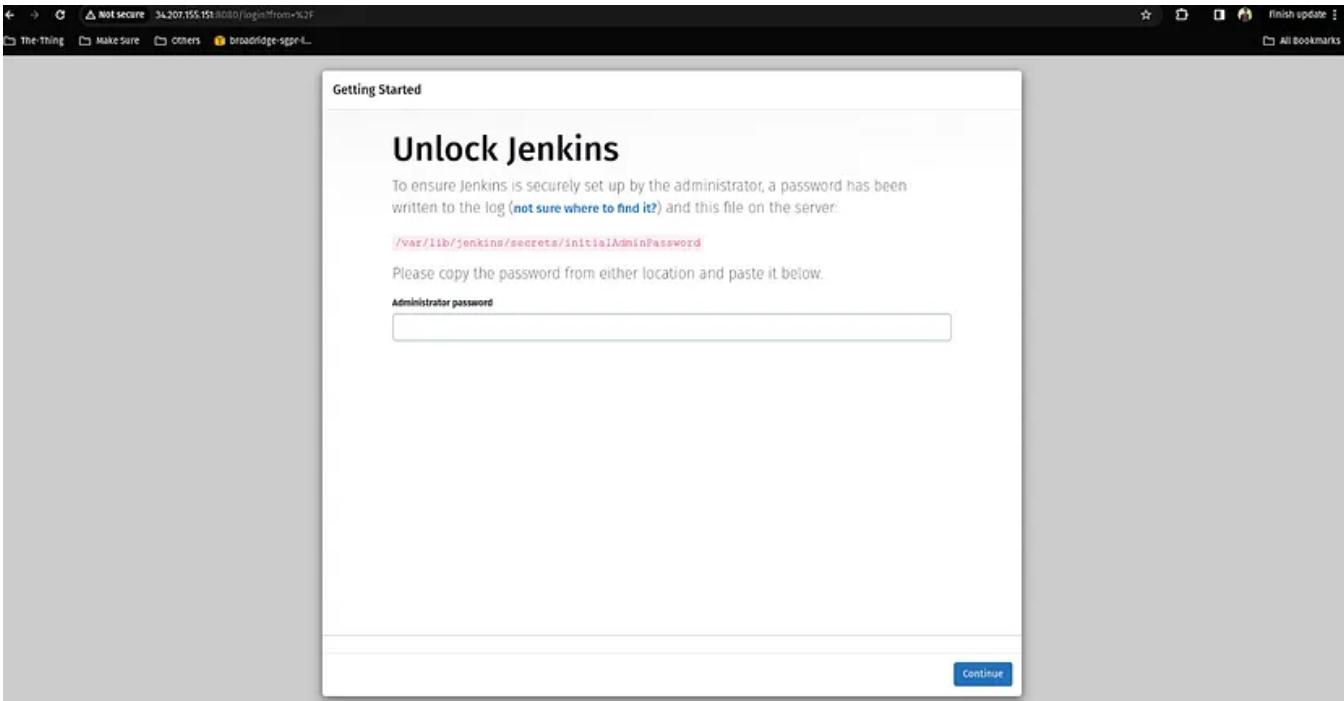
```
ubuntu@ip-172-31-59-9:~$ sudo apt update -y
sudo apt install openjdk-11-jre -y
java --version
# Installing Jenkins
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [116 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1851 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1263 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [260 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1250 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [203 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1020 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [226 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [200 kB]
```

## Check the status of the Jenkins server

```
ubuntu@ip-172-31-59-9:~$ systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-12-27 14:01:16 UTC; 1min 15 ago
       Main PID: 4896 (java)
          Tasks: 49 (limit: 9498)
            Memory: 2.2G
              CPU: 53.266s
             CGroup: /system.slice/jenkins.service
                       └─4896 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Dec 27 14:00:51 ip-172-31-59-9 jenkins[4896]: 5c267e0c42134562ad3b5b4c263784f2
Dec 27 14:00:51 ip-172-31-59-9 jenkins[4896]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Dec 27 14:00:51 ip-172-31-59-9 jenkins[4896]: ****
Dec 27 14:00:51 ip-172-31-59-9 jenkins[4896]: ****
Dec 27 14:00:51 ip-172-31-59-9 jenkins[4896]: ****
Dec 27 14:01:16 ip-172-31-59-9 jenkins[4896]: 2023-12-27 14:01:16.166+0000 [id=29] INFO jenkins.InitReactorRunner$1@onAttained: Completed initialization
Dec 27 14:01:16 ip-172-31-59-9 jenkins[4896]: 2023-12-27 14:01:16.192+0000 [id=22] INFO hudson.lifecycle.Lifecycle$OnReady: Jenkins is fully up and running
Dec 27 14:01:16 ip-172-31-59-9 systemd[1]: Started Jenkins Continuous Integration Server.
Dec 27 14:01:16 ip-172-31-59-9 jenkins[4896]: 2023-12-27 14:01:16.294+0000 [id=47] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.
Dec 27 14:01:16 ip-172-31-59-9 jenkins[4896]: 2023-12-27 14:01:16.295+0000 [id=47] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the
lines 1-20/20 (END)
```

Copy your Jenkins Server Public IP and paste it into your favorite browser with port number 8080.



Run the command on your Jenkins server

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

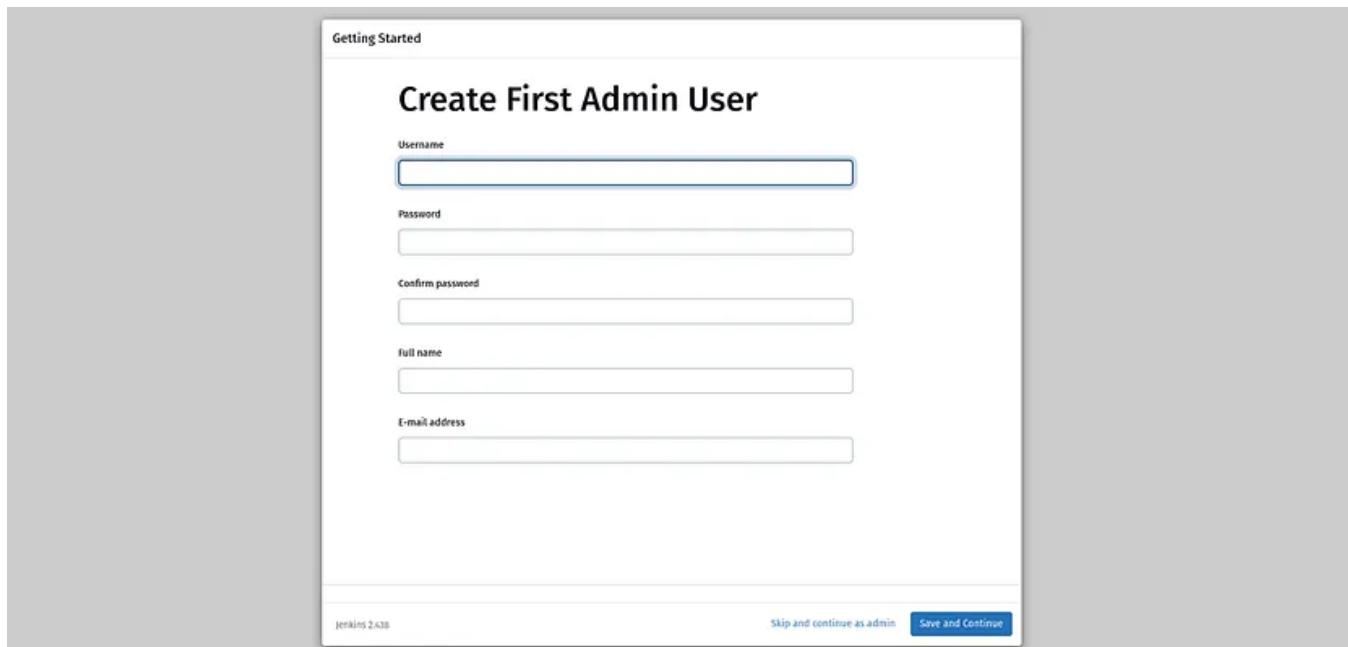
Copy the output and paste it into your above snippet text field and click on Continue.

```
ubuntu@ip-172-31-59-9:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
5c267e0cc42134562ad3b5b4c263784f2
ubuntu@ip-172-31-59-9:~$
```

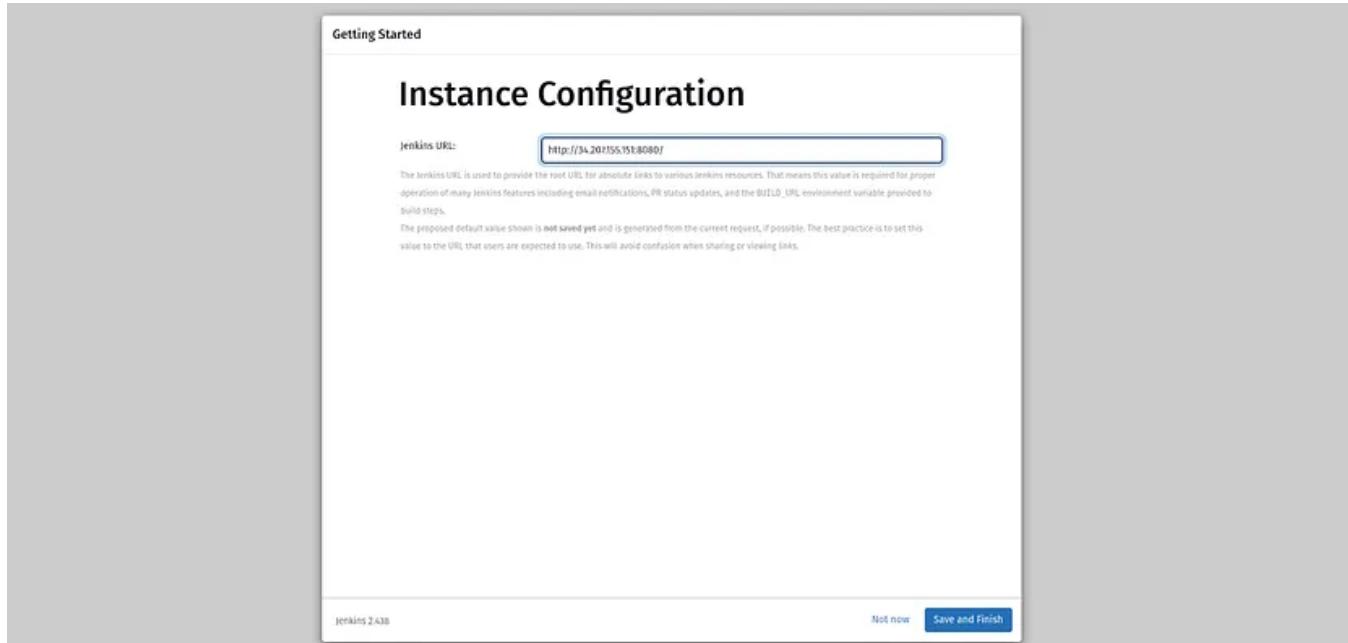
Click on the Install suggested plugins



Click on the Skip and continue as admin



Click on Save and Finish



## Install Docker and configure on the Jenkins Server

```
sudo apt update
sudo apt install docker.io -y
sudo usermod -aG docker jenkins
sudo usermod -aG docker ubuntu
sudo systemctl restart docker
sudo chmod 777 /var/run/docker.sock
```

```
ubuntu@ip-172-31-59-9:~$ sudo apt update
sudo apt install docker.io
sudo su -
sudo usermod -aG docker jenkins
sudo usermod -aG docker ubuntu
systemctl restart docker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
26 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 26 not upgraded.
Need to get 6997 kB of archives.
```

## Install Sonarqube on your Jenkins Server

We will use a docker container for Sonarqube

```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

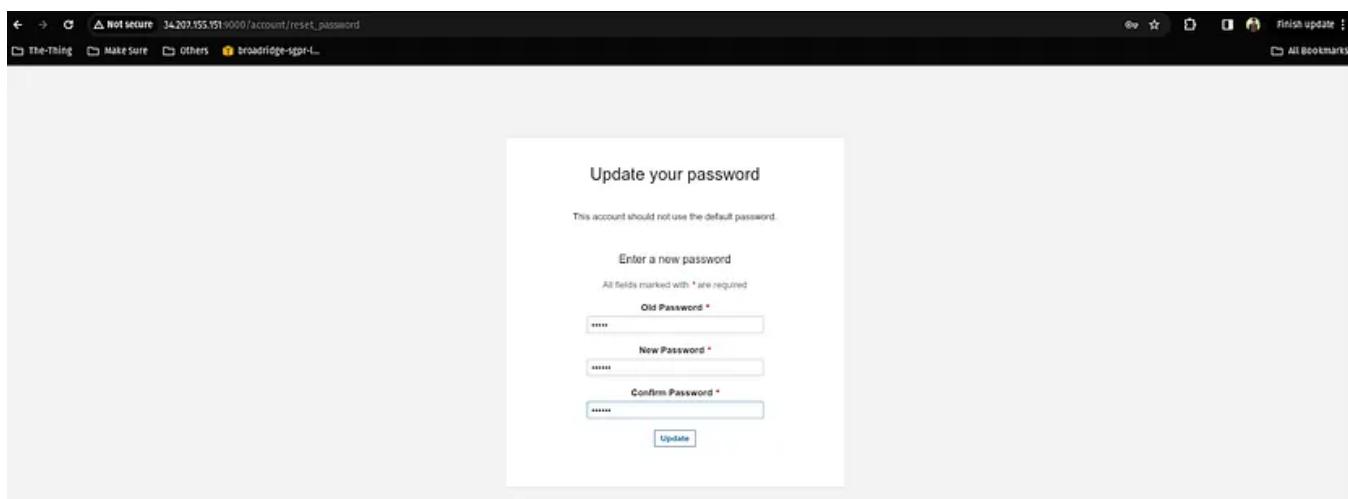
```
ubuntu@ip-172-31-59-9:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
3dd181f9be59: Pull complete
0fb308a55bddf: Pull complete
e7ee05bc88e6: Pull complete
51526e7965d8: Pull complete
ffcdcf7c6c160: Pull complete
9d141c530e5b: Pull complete
bb9a8113b2efe: Pull complete
Digest: sha256:49ac473fc9da87052cdd205e4581a5b369deaf65832830d62be86e419ea2elf
Status: Downloaded newer image for sonarqube:lts-community
08900b3945a5ddb0714716d59336e053a149779898d488a0ca9847e0def1fdcc
ubuntu@ip-172-31-59-9:~$ []
```

Now, copy your Public IP of Jenkins Server and add 9000 Port on your browser.

The username and password will be admin



Reset the password and click on Update



You will see your Sonarqube Server in the below snippet.

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

**From Azure DevOps** Set up global configuration

**From Bitbucket Server** Set up global configuration

**From Bitbucket Cloud** Set up global configuration

**From GitHub** Set up global configuration

**From GitLab** Set up global configuration

Are you just testing or have an advanced use-case? Create a project manually.

< >  
Manually

## Install the Trivy tool on the Jenkins Server

```
sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-
echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main
sudo apt-get update
sudo apt-get install trivy
```

```
ubuntu@ip-172-31-59-9:~$ sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
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
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release is set to manually installed.
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 26 not upgraded.
Need to get 1510 B of archives.
After this operation, 176 kB of additional 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-updates/universe amd64 apt-transport-https all 2.4.11 [1510 B]
Fetched 1510 B in 0s (90.5 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 66852 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.11_all.deb ...
Unpacking apt-transport-https (2.4.11) ...
Setting up apt-transport-https (2.4.11) ...
Scanning processes...
Scanning Linux images...
```

## Install and Configure the Prometheus, Node Exporter, and Grafana on the Monitoring Server

### Login to the Monitoring Server

```
amampathak@pop-01:~$ ssh -l "Aman-Pathak.pem" ubuntu@ec2-54-152-127-65.compute-1.amazonaws.com
The authenticity of host 'ec2-54-152-127-65.compute-1.amazonaws.com (54.152.127.65)' can't be established.
ED25519 key fingerprint is SHA256:srlARWn46d0Db/VlVnJduC1mWBv5NWhK2Ru10WiCBo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-152-127-65.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

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

System information as of Wed Dec 27 15:13:05 UTC 2023

System load:  0.0          Processes:           105
Usage of /:   10.8% of 14.36GB  Users logged in:      0
Memory usage: 5%
Swap usage:  0%

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-172-31-52-7:~$ 
```

# Create Prometheus user

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

```
ubuntu@ip-172-31-52-7:~$ sudo useradd \n    --system \n    --no-create-home \n    --shell /bin/false prometheus\nubuntu@ip-172-31-52-7:~$ [ ]
```

**Download the Prometheus file on the Monitoring Server**

```
wget https://github.com/prometheus/prometheus/releases/download/v2.49.0-rc.1/prometheus
```

## Untar the Prometheus downloaded package

```
tar -xvf prometheus-2.49.0-rc.1.linux-amd64.tar.gz
```

```
ubuntu@ip-172-31-52-7:~$ tar -xvf prometheus-2.49.0-rc.1.linux-amd64.tar.gz
prometheus-2.49.0-rc.1.linux-amd64/
prometheus-2.49.0-rc.1.linux-amd64/NOTICE
prometheus-2.49.0-rc.1.linux-amd64/LICENSE
prometheus-2.49.0-rc.1.linux-amd64/prometheus
prometheus-2.49.0-rc.1.linux-amd64/consoles/
prometheus-2.49.0-rc.1.linux-amd64/consoles/prometheus-overview.html
prometheus-2.49.0-rc.1.linux-amd64/consoles/node-cpu.html
prometheus-2.49.0-rc.1.linux-amd64/consoles/node.html
prometheus-2.49.0-rc.1.linux-amd64/consoles/index.html.example
prometheus-2.49.0-rc.1.linux-amd64/consoles/node-overview.html
prometheus-2.49.0-rc.1.linux-amd64/consoles/node-disk.html
prometheus-2.49.0-rc.1.linux-amd64/prometheus.yml
prometheus-2.49.0-rc.1.linux-amd64/promtool
prometheus-2.49.0-rc.1.linux-amd64/console_libraries/
prometheus-2.49.0-rc.1.linux-amd64/console_libraries/menu.lib
prometheus-2.49.0-rc.1.linux-amd64/console_libraries/prom.lib
ubuntu@ip-172-31-52-7:~$
```

Create two directories /data and /etc/prometheus to configure the Prometheus

```
sudo mkdir -p /data /etc/prometheus
```

Now, enter into the prometheus package file that you have untar **in** the earlier  
`cd prometheus-2.49.0-rc.1.linux-amd64/`

```
ubuntu@ip-172-31-52-7:~$ sudo mkdir -p /data /etc/prometheus
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ cd prometheus-2.49.0-rc.1.linux-amd64/
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64/
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ ls
LICENSE NOTICE console_libraries consoles prometheus prometheus.yml promtool
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$
```

Move the prometheus and promtool files package in /usr/local/bin

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

```
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ ls
LICENSE NOTICE console_libraries consoles prometheus prometheus.yml promtool
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo mv prometheus promtool /usr/local/bin/
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$
```

Move the console and console\_libraries and prometheus.yml in the /etc/prometheus

```
sudo mv consoles console_libraries/ prometheus.yml /etc/prometheus/
```

```
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo mv consoles console_libraries/ prometheus.yml /etc/prometheus/
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ ls /etc/prometheus/
console_libraries consoles prometheus.yml
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ 
```

Provide the permissions to prometheus user

```
sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
```

```
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ 
```

Check and validate the Prometheus

```
prometheus --version
```

```
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ prometheus --version
prometheus, version 2.49.0-rc.1 (branch: HEAD, revision: le306856a6716366c74dbde6fce160fe5fd1fcda)
  build user:        root@23ee6cc014eb
  build date:      20231220-09:30:04
  go version:     go1.21.5
  platform:       linux/amd64
  tags:          netgo,builtinassets,stringlabels
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ 
```

Create a systemd configuration file for prometheus

Edit the file /etc/systemd/system/prometheus.service

```
sudo vim /etc/systemd/system/prometheus.service
```

and paste the below configurations in your prometheus.service configuration file and save it

```
[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
```

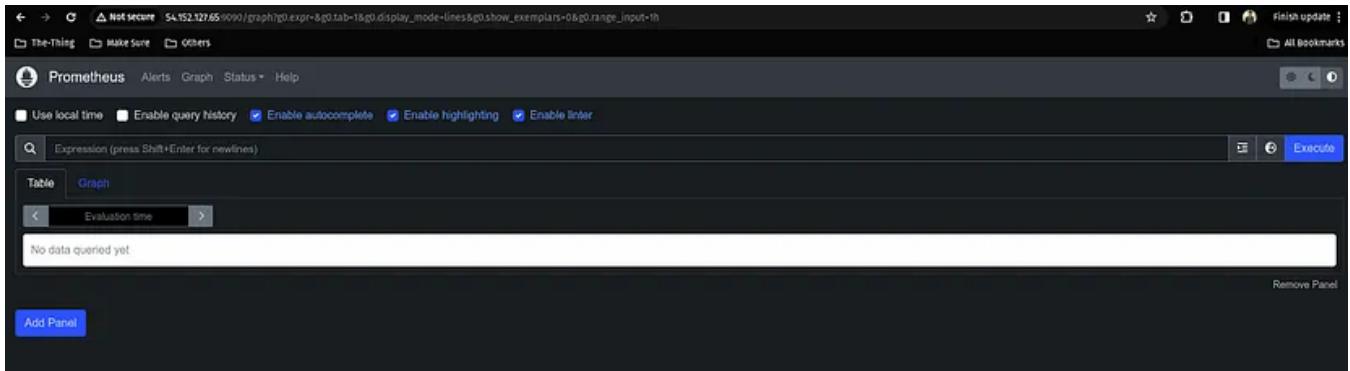
Once you write the systemd configuration file for Prometheus, then enable it and start the Prometheus service.

```
sudo systemctl enable prometheus.service
sudo systemctl start prometheus.service
systemctl status prometheus.service
```

```
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo systemctl enable prometheus.service
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo systemctl start prometheus.service
ubuntu@ip-172-31-52-7:~/prometheus-2.49.0-rc.1.linux-amd64$ sudo systemctl status prometheus.service
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-12-27 15:39:56 UTC; 5s ago
       PID: 1793 (prometheus)
      Tasks: 7 (limit: 4667)
        CPU: 67ms
       CGroup: /system.slice/prometheus.service
           └─1793 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/data --web.console.templates=/etc/prometheus/consoles --web.console.libraries=

Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.662Z caller=head.go:697 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=3.355µs
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.662Z caller=head.go:695 level=info component=tsdb msg="Replaying WAL, this may take a while"
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.662Z caller=head.go:766 level=info component=tsdb msg="WAL segment loaded: segment=0 maxSegment=0"
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.662Z caller=head.go:777 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=34.161µs wal_size=0
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.666Z caller=main.go:1060 level=info msg="TSDB started"
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.666Z caller=main.go:1245 level=info msg="Loading configuration file" filename=/etc/prometheus/prometheus.yml
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.679Z caller=main.go:1282 level=info msg="Completed loading of configuration file" filename=/etc/prometheus/prometheus.yml
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.679Z caller=main.go:1024 level=info msg="Server is ready to receive web requests."
Dec 27 15:39:56 ip-172-31-52-7 prometheus[1793]: ts=2023-12-27T15:39:56.679Z caller=manager.go:146 level=info component="rule manager" msg="Starting rule manager..."
lines 1-20/20 (END)
```

Once the Prometheus service is up and running then, copy the public IP of your **Monitoring Server** and paste it into your favorite browser with a 9090 port.



Now, we have to install a node exporter to visualize the machine or hardware level data such as CPU, RAM, etc on our Grafana dashboard.

To do that, we have to create a user for it.

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

```
ubuntu@ip-172-31-52-7:~$ sudo useradd \
--system \
--no-create-home \
--shell /bin/false node_exporter
ubuntu@ip-172-31-52-7:~$
```

Download the node exporter package

```
wget https://github.com/prometheus/node\_exporter/releases/download/v1.7.0/node\_
```

```
ubuntu@ip-172-31-52-7:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.49.0-rc.1/prometheus-2.49.0-rc.1.linux-amd64.tar.gz
--2023-12-27 15:15:08-- https://github.com/prometheus/prometheus/releases/download/v2.49.0-rc.1/prometheus-2.49.0-rc.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/7b134043-d3ed-4b37-bbe2-789435ef9583?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVHE53AV2F20231227%2Fus-east-1%2Fs3%2Faws4-request%2FAmz-Date=20231227T151508Z&X-Amz-Expires=3006X-Amz-Signature=7957799ca66d731c180d174fe00f64630b03d19bb0f85d98e7973c28731736X-Amz-SignedHeaders=host&actor_id=6&repo_id=6838921&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.49.0-rc.1.linux-amd64.tar.gz&response-content-type=application/x-tar
[following]
--2023-12-27 15:15:08-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/7b134043-d3ed-4b37-bbe2-789435ef9583?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVHE53AV2F20231227%2Fus-east-1%2Fs3%2Faws4-request%2FAmz-Date=20231227T151508Z&X-Amz-Expires=3006X-Amz-Signature=7957799ca66d731c180d174fe00f64630b03d19bb0f85d98e7973c28731736X-Amz-SignedHeaders=host&actor_id=6&repo_id=6838921&response-content-disposition=attachment%3B%20filename%3Dprometheus-2.49.0-rc.1.linux-amd64.tar.gz&response-content-type=application/x-tar
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 97161690 (93M) [application/octet-stream]
Saving to: 'prometheus-2.49.0-rc.1.linux-amd64.tar.gz'

prometheus-2.49.0-rc.1.linux-amd64.tar.gz      100%[=====] 92.66M  131MB/s   in 0.7s

2023-12-27 15:15:09 [131 MB/s] - 'prometheus-2.49.0-rc.1.linux-amd64.tar.gz' saved [97161690/97161690]

ubuntu@ip-172-31-52-7:~$ ls
prometheus-2.49.0-rc.1.linux-amd64.tar.gz
ubuntu@ip-172-31-52-7:~$ 
```

Untar the node exporter package file and move the node\_exporter directory to the /usr/local/bin directory

```
tar -xvf node_exporter-1.7.0.linux-amd64.tar.gz
sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
```

```
ubuntu@ip-172-31-52-7:~$ tar -xvf node_exporter-1.7.0.linux-amd64.tar.gz
node_exporter-1.7.0.linux-amd64/
node_exporter-1.7.0.linux-amd64/LICENSE
node_exporter-1.7.0.linux-amd64/node_exporter
node_exporter-1.7.0.linux-amd64/NOTICE
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
ubuntu@ip-172-31-52-7:~$ 
```

Validate the version of the node exporter

```
node_exporter --version
```

```
ubuntu@ip-172-31-52-7:~$ node_exporter --version
node_exporter, version 1.7.0 (branch: HEAD, revision: 7333465abf9efba81876383bb57e6fad946041b)
  Build user:        root@035918982f6d8
  Build date:       2023112-23:53:35
  Go version:       go1.18.4
  Platform:         linux/amd64
  Target:           netgo osusergo static_build
ubuntu@ip-172-31-52-7:~$ 
```

Create the systemd configuration file for node exporter.

Edit the file

```
sudo vim /etc/systemd/system/node_exporter.service
```

Copy the below configurations and paste them into the /etc/systemd/system/node\_exporter.service file.

```
[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
]
```

Enable the node exporter systemd configuration file and start it.

```
sudo systemctl enable node_exporter
sudo systemctl enable node_exporter
systemctl status node_exporter.service
```

```
ubuntu@ip-172-31-52-7:~$ 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-172-31-52-7:~$ sudo systemctl start node_exporter.service
ubuntu@ip-172-31-52-7:~$ sudo systemctl status node_exporter.service
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-12-27 15:53:27 UTC; 10s ago
       Main PID: 1987 (node exporter)
         Tasks: 5 (limit: 4667)
        Memory: 2.4M
          CPU: 9ms
        CGroup: /system.slice/node_exporter.service
                └─1987 /usr/local/bin/node_exporter --collector.logind

Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=thermal_zone
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=time
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=timex
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=udp_queues
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=uname
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=vmstat
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.699Z caller=node_exporter.go:117 level=info collector=xfs
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.700Z caller=node_exporter.go:117 level=info collector=zfs
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.700Z caller=tls_config.go:274 level=info msg="Listening on" address=[::]:9100
Dec 27 15:53:27 ip-172-31-52-7 node_exporter[1987]: ts=2023-12-27T15:53:27.700Z caller=tls_config.go:277 level=info msg="TLS is disabled." http2=false address=[::]:9100
ubuntu@ip-172-31-52-7:~$
```

Now, we have to add a node exporter to our Prometheus target section. So, we will be able to monitor our server.

edit the file

```
sudo vim /etc/prometheus/prometheus.yml
```

Copy the content in the file

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

```
# 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:
# Here it's Prometheus itself.
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"]
```

After saving the file, validate the changes that you have made using promtool.

```
promtool check config /etc/prometheus/prometheus.yml
```

```
ubuntu@ip-172-31-52-7:~$ 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-172-31-52-7:~$
```

If your changes have been validated then, push the changes to the Prometheus server.

```
curl -X POST http://localhost:9090/-/reload
```

```
ubuntu@ip-172-31-52-7:~$ curl -X POST http://localhost:9090/-/reload
ubuntu@ip-172-31-52-7:~$
```

Now, go to your Prometheus server and this time, you will see one more target section as node\_exporter which should be up and running.

The screenshot shows the Prometheus Targets page. At the top, there are tabs for 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below the tabs, there's a search bar and a filter for 'All scrape pools' with options for 'All', 'Unhealthy', and 'Collapse All'. A checkbox for 'Unknown' is checked. There are also checkboxes for 'Unhealthy' and 'Healthy'.

**Targets**

**node\_exporter (1/1 up)**

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter"	8.326s ago	13.824ms	

**prometheus (1/1 up)**

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	7.569 ago	4.867ms	

Now, install the Grafana tool to visualize all the data that is coming with the help of Prometheus.

```
sudo apt-get install -y apt-transport-https software-properties-common wget
sudo mkdir -p /etc/apt/keyrings/
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/ap
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com sta
```

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com beta
sudo apt-get update
```

```
ubuntu@ip-172-31-52-7:~$ sudo apt-get install -y apt-transport-https software-properties-common wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'apt' instead of 'apt-transport-https'
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
apt is already the newest version (2.4.11).
apt set to manually installed.
software-properties-common is already the newest version (0.99.22.8).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ sudo mkdir -p /etc/apt/keyrings/
ubuntu@ip-172-31-52-7:~$ wget -q -O https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com beta main
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com jammy universe
ubuntu@ip-172-31-52-7:~$ 
ubuntu@ip-172-31-52-7:~$ # Updates the list of available packages
sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 https://apt.grafana.com stable InRelease [5984 B]
Get:6 https://apt.grafana.com stable/main amd64 Packages [5976 B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:8 https://apt.grafana.com stable/main amd64 Packages [384 kB]
Err:6 https://apt.grafana.com beta InRelease
```

## Install the Grafana

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

```
ubuntu@ip-172-31-52-7:~$ # Installs the latest OSS release:
sudo apt-get install grafana
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 musl
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core grafana libfontconfig1 musl
0 upgraded, 5 newly installed, 0 to remove and 26 not upgraded.
Need to get 105 MB of archives.
After this operation, 388 MB of additional 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 fonts-dejavu-core all 2.37-2build1 [1041 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [29.1 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfontconfig1 amd64 2.13.1-4.2ubuntu5 [131 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 grafana amd64 1.2.2.4 [487 kB]
Get:5 https://apt.grafana.com stable/main amd64 grafana amd64 10.2.3 [104 kB]
Fetched 105 MB in 3s (36.6 MB/s)
Selecting previously unselected package fonts-dejavu-core.
(Reading database ... 64799 files and directories currently installed.)
```

## Enable and start the Grafana Service

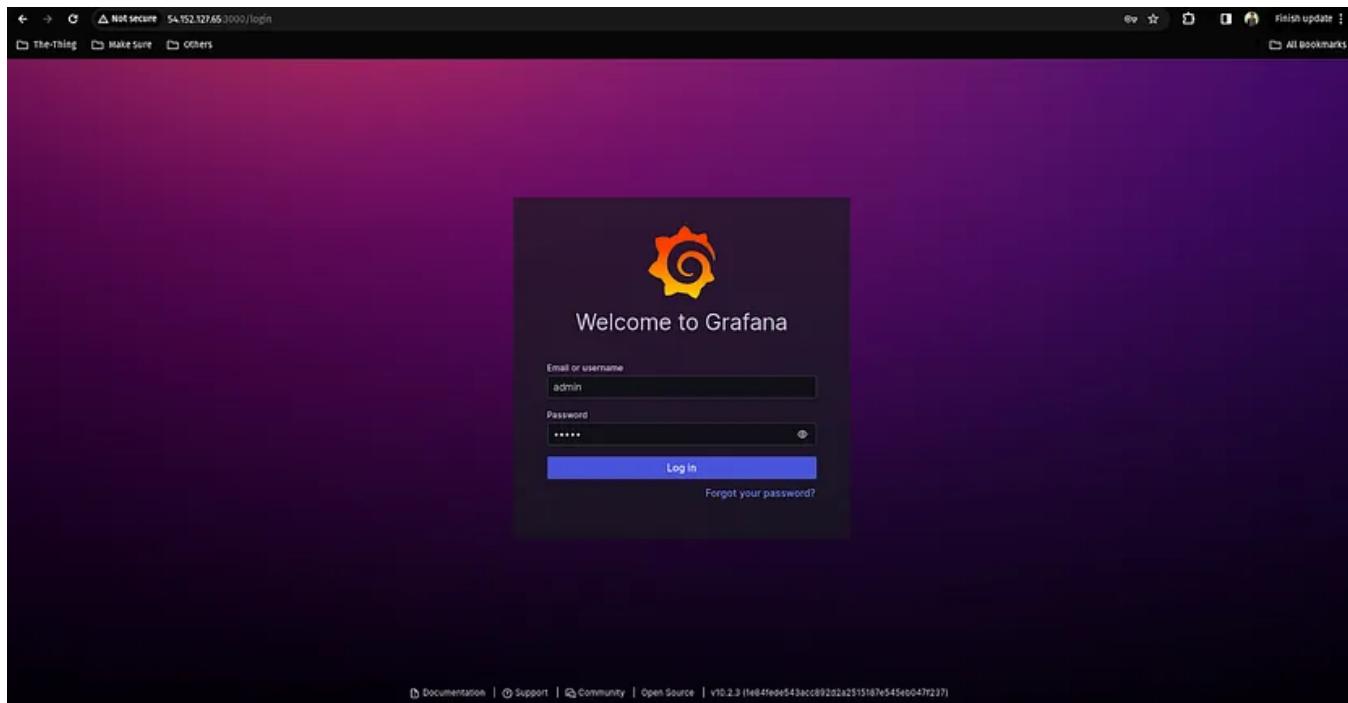
```
sudo systemctl enable grafana-server.service
sudo systemctl start grafana-server.service
sudo systemctl status grafana-server.service
```

```
ubuntu@ip-172-31-52-7:~$ sudo systemctl enable grafana-server.service
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemctl-sysv-install.
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.
ubuntu@ip-172-31-52-7:~$ sudo systemctl start grafana-server.service
ubuntu@ip-172-31-52-7:~$ sudo systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-12-27 16:07:44 UTC; 5s ago
       Docs: http://docs.grafana.org
   Main PID: 2834 (grafana)
      Tasks: 6 (limit: 4667)
        Memory: 37.8M
         CPU: 1.772s
      CGroup: /system.slice/grafana-server.service
              └─2834 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg=default.paths.logs=/var/log/grafana

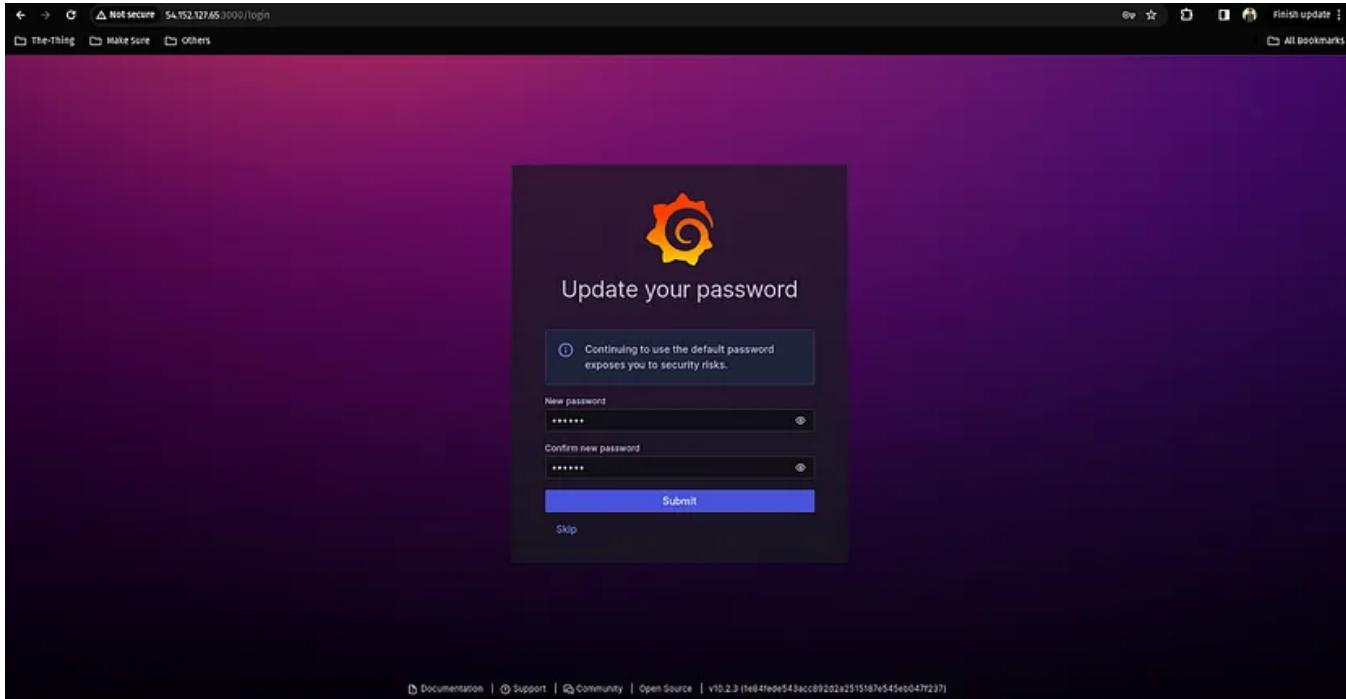
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.050051795Z level=info msg="Migration successfully executed" id="Update uid column values in playlist" duration=1ms
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.057250519Z level=info msg="Executing migration" id="Add index for uid in playlist"
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.058422538Z level=info msg="Migration successfully executed" id="Add index for uid in playlist" duration=1ms
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.06478448Z level=info msg="Executing migration" id="update group index for alert rules"
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.065304835Z level=info msg="Migration successfully executed" id="update group index for alert rules" duration=1ms
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.071421989Z level=info msg="Executing migration" id="managed folder permissions alert actions repeated migrations"
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.071752122Z level=info msg="Migration successfully executed" id="managed folder permissions alert actions" duration=1ms
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.077458169Z level=info msg="Executing migration" id="admin only folder/dashboard permission"
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.083563943Z level=info msg="Migration successfully executed" id="admin only folder/dashboard permission" duration=1ms
Dec 27 16:07:50 ip-172-31-52-7 grafana[2834]: logger=migrator t=2023-12-27T16:07:50.083563943Z level=info msg="Executing migration" id="add action column to seed_assignment"
lines 1-21 (ENDO)
```

To access the Grafana dashboard, copy the public IP address of the **Monitoring Server** and paste it into your favorite browser with port 3000

username and password will be admin



Reset the password



## Click on Data sources

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

**Basic**  
The steps below will guide you to quickly finish setting up your Grafana installation.

**TUTORIAL**  
**DATA SOURCE AND DASHBOARDS**  
Grafana fundamentals  
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

**DATA SOURCES**  
Add your first data source  
Learn how in the docs

**DASHBOARDS**  
Create your first dashboard  
Learn how in the docs

**Dashboards**  
Starred dashboards  
Recently viewed dashboards

Latest from the blog

- Dec 27 **Grafana dashboards in 2023: Memorable use cases of the year**  
As the number of Grafana users grows each year, so does the variety of reasons people are using Grafana dashboards. During 2023, members of our community — both inside and outside of the company — shared some of their incredible professional and personal projects, including how Grafana has allowed them to successfully launch a rocket, cut back on carbon emissions, and even help balance a national power grid. Let's take a look back at some of the most unforgettable dashboards we've seen this year!
- Dec 26 **Grafana Labs in 2023: Year in review**  
Open source is the foundation of everything we do here at Grafana Labs, and that was on full display this year as we celebrated the 10th anniversary of Grafana and continued to improve and expand our lineup of OSS projects. But 2023 was also a banner year for Grafana Cloud,

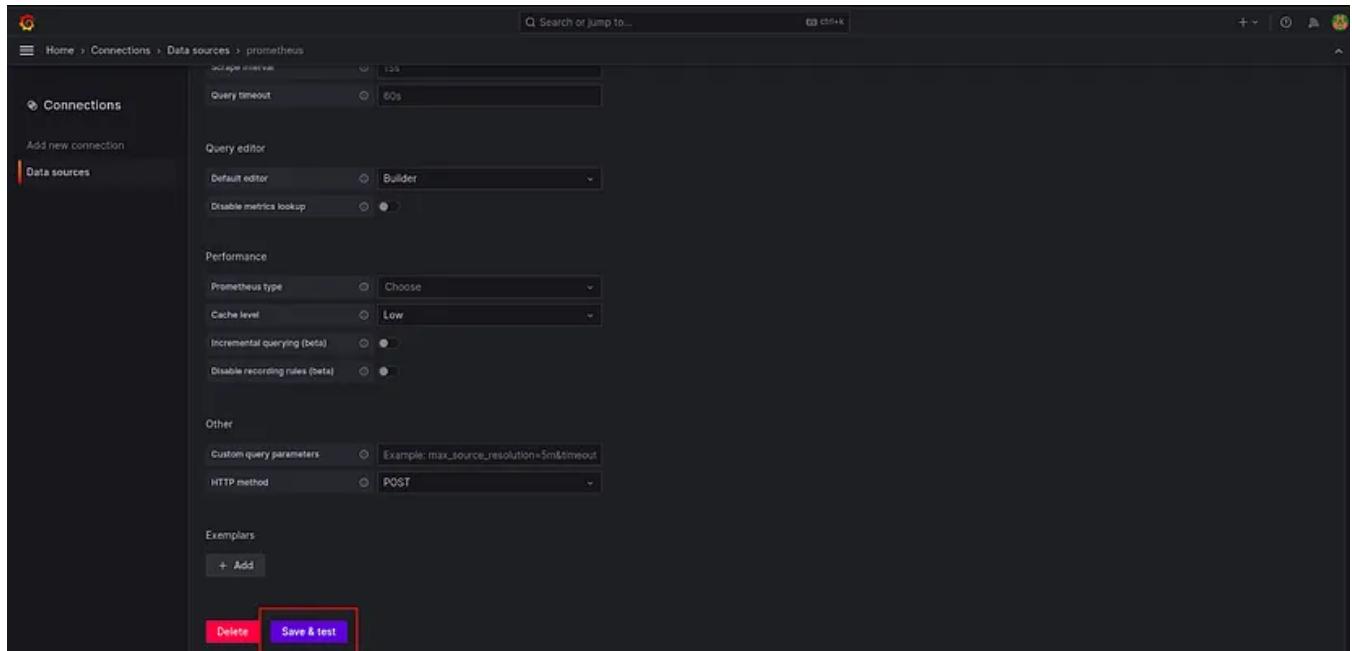
## Select the Prometheus

The screenshot shows the 'Add data source' dialog in Grafana. Under the 'Time series databases' heading, the 'Prometheus' option is selected and highlighted with a red box. Other options listed include Graphite, InfluxDB, and OpenTSDB. Below this section, there are sections for 'Logging & document databases' containing Loki and Elasticsearch.

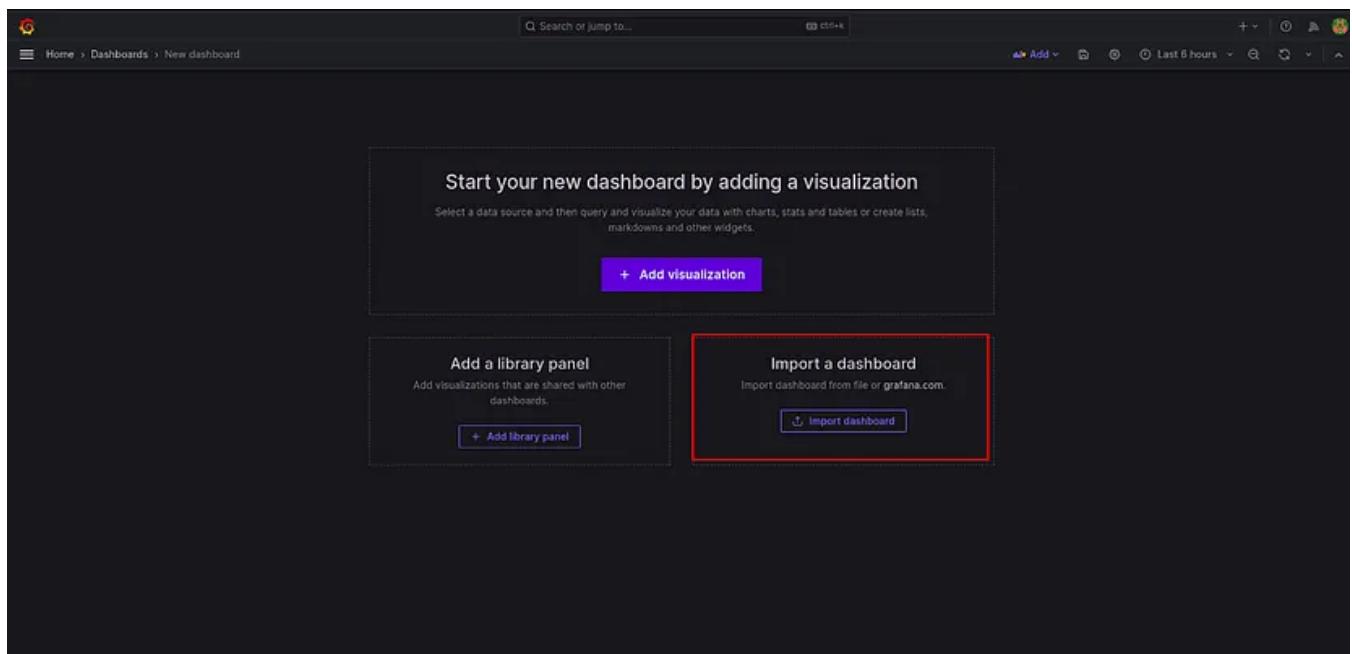
**Provide the Monitoring Server Public IP with port 9090 to monitor the Monitoring Server.**

The screenshot shows the 'Settings' page for the 'prometheus' data source. The 'Type' is set to 'Prometheus'. The 'Name' field contains 'prometheus'. The 'Prometheus server URL' field is highlighted with a red box and contains 'http://54.152.127.65:9090'. The 'Authentication methods' section is visible at the bottom.

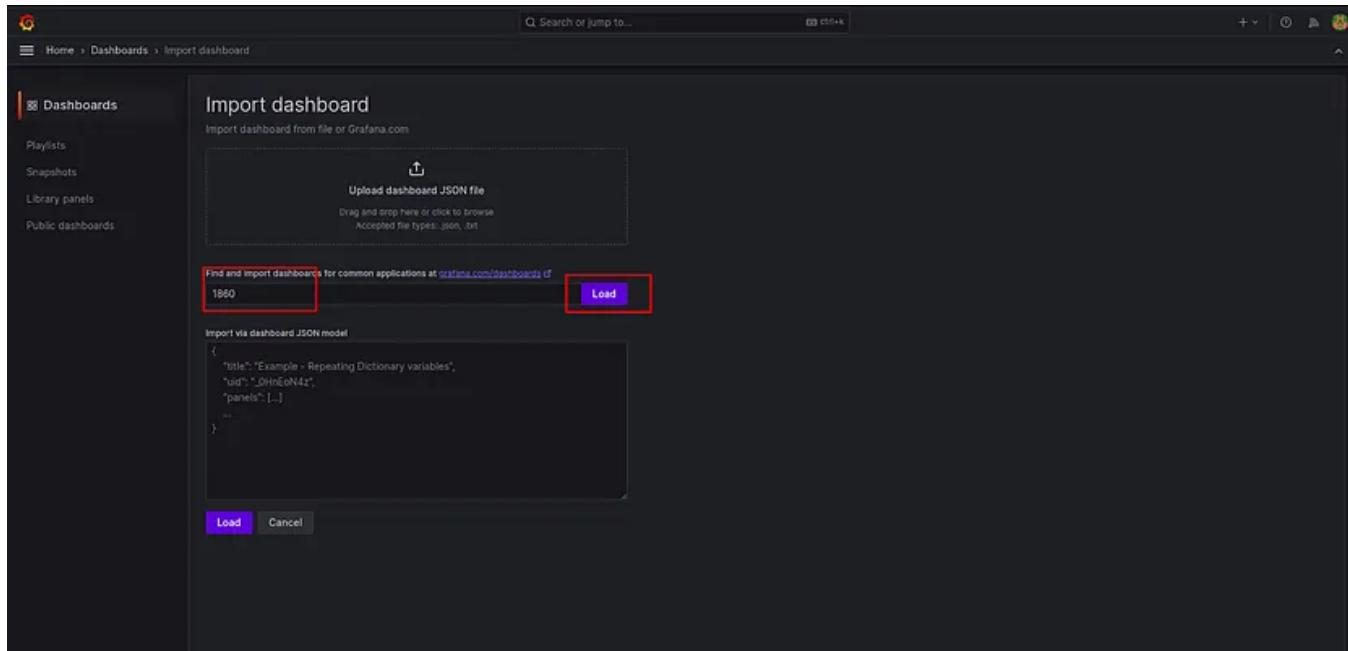
**Click on Save and test.**



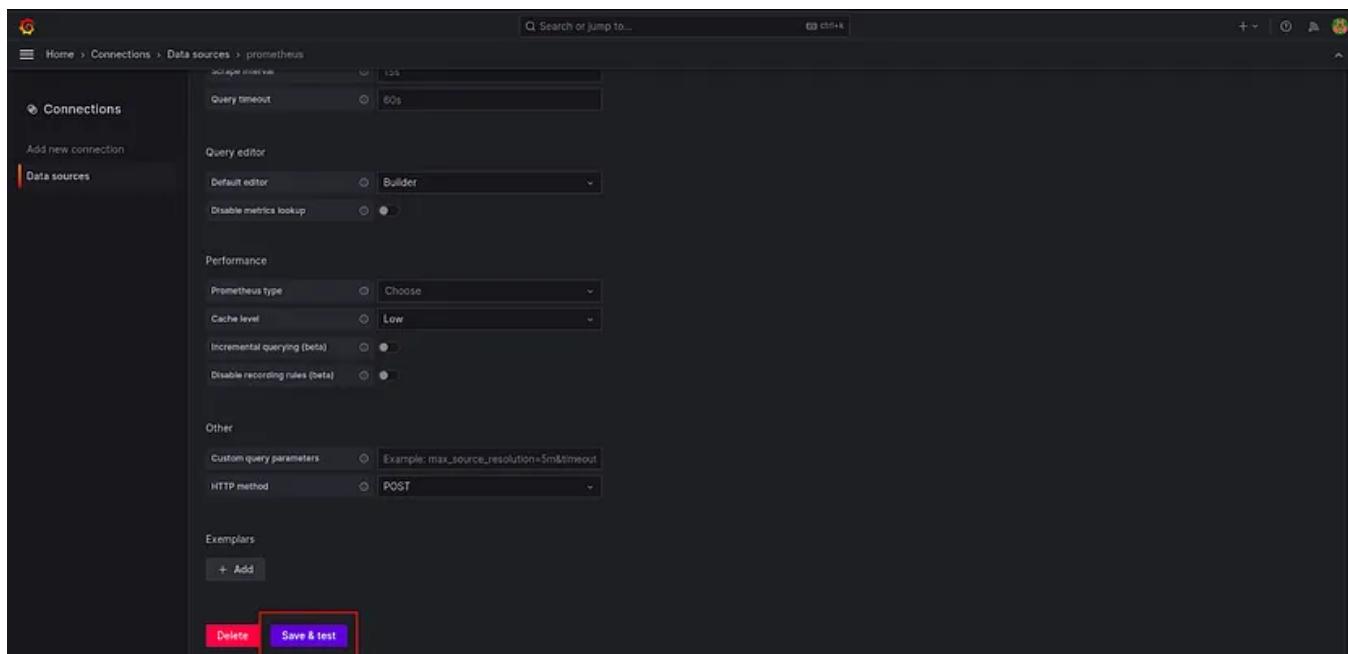
Go to the dashboard section of Grafana and click on the Import dashboard.



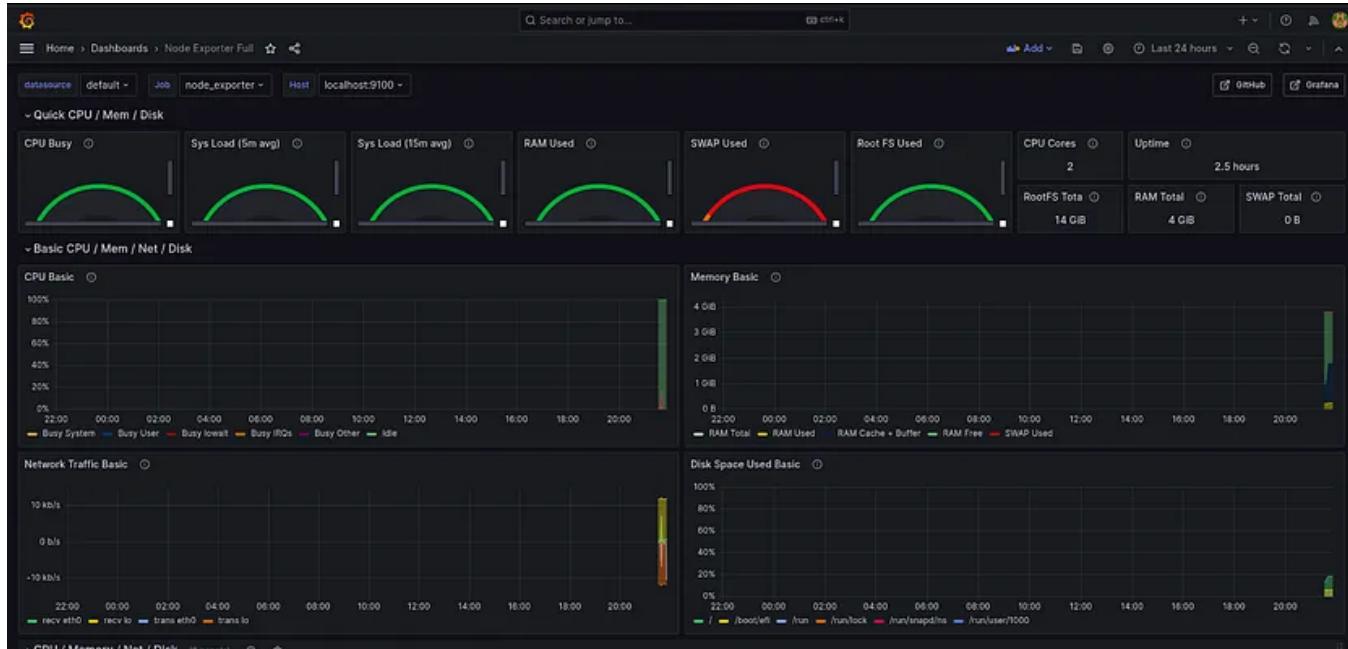
Add 1860 for the node exporter dashboard and click on Load.



Then, select the Prometheus from the drop down menu and click on Import



The dashboard will look like this



Now, we have to monitor our Jenkins Server as well.

For that, we need to install the Prometheus metric plugin on our Jenkins.

Go to Manage Jenkins -> Plugin search for Prometheus metrics install it and restart your Jenkins.

Install	Name	Released
<input checked="" type="checkbox"/>	Prometheus metrics 2.4.2 monitoring   Miscellaneous Jenkins Prometheus Plugin expose an endpoint (default: /prometheus) with metrics where a Prometheus Server can scrape.	1 day 0 hr ago
<input type="checkbox"/>	Cortex Metrics 1.0.1 Adds the ability to publish run results to Cortex directly using the Prometheus push endpoint.	2 yr 9 mo ago
<input type="checkbox"/>	Otel agent host metrics monitoring 1.3.3 monitoring   observability This plugin allows monitoring of Jenkins agents by deploying Prometheus node exporters and Otel collectors to them and linking to a Grafana dashboard displaying those gathered metrics.	1 mo 20 days ago

Edit the /etc/prometheus/prometheus.yml file

```
sudo vim /etc/prometheus/prometheus.yml
```

```
- job_name: "jenkins"
  static_configs:
    - targets: ["<jenkins-server-public-ip>:8080"]
```

```
# 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:
# Here it's Prometheus itself.
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"
    static_configs:
      - targets: ["34.207.155.151:8080"]
```

37,41 All

Once you add the Jenkins job, validate the Prometheus config file whether it is correct or not by running the below command.

```
promtool check config /etc/prometheus/prometheus.yml
```

Now, push the new changes on the Prometheus server

```
curl -X POST http://localhost:9090/-/reload
```

```
ubuntu@ip-172-31-51-7:~$ sudo vim /etc/prometheus/prometheus.yml
ubuntu@ip-172-31-51-7:~$ ubuntu@ip-172-31-51-7:~$ 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-172-31-51-7:~$ curl -X POST http://localhost:9090/-/reload
ubuntu@ip-172-31-51-7:~$ █
```

Copy the public IP of your **Monitoring Server** and paste on your favorite browser with a 9090 port with /target. You will see the targets that you have added in the /etc/prometheus/prometheus.yml file.

The screenshot shows the Prometheus Targets page with three healthy targets listed:

- Jenkins (1/1 up)**: Endpoint `http://34.207.155.151:8080/prometheus`, State UP, Labels `instance="34.207.155.151:8080"`, `joker="jenkins"`, Last Scrape 13.958s ago, Scrape Duration 23.806ms.
- node\_exporter (1/1 up)**: Endpoint `http://localhost:9100/metrics`, State UP, Labels `instance="localhost:9100"`, `joker="node_exporter"`, Last Scrape 9.720s ago, Scrape Duration 15.258ms.
- prometheus (1/1 up)**: Endpoint `http://localhost:9090/metrics`, State UP, Labels `instance="localhost:9090"`, `joker="prometheus"`, Last Scrape 8.450s ago, Scrape Duration 5.688ms.

To add the Jenkins Dashboard on your Grafana server.

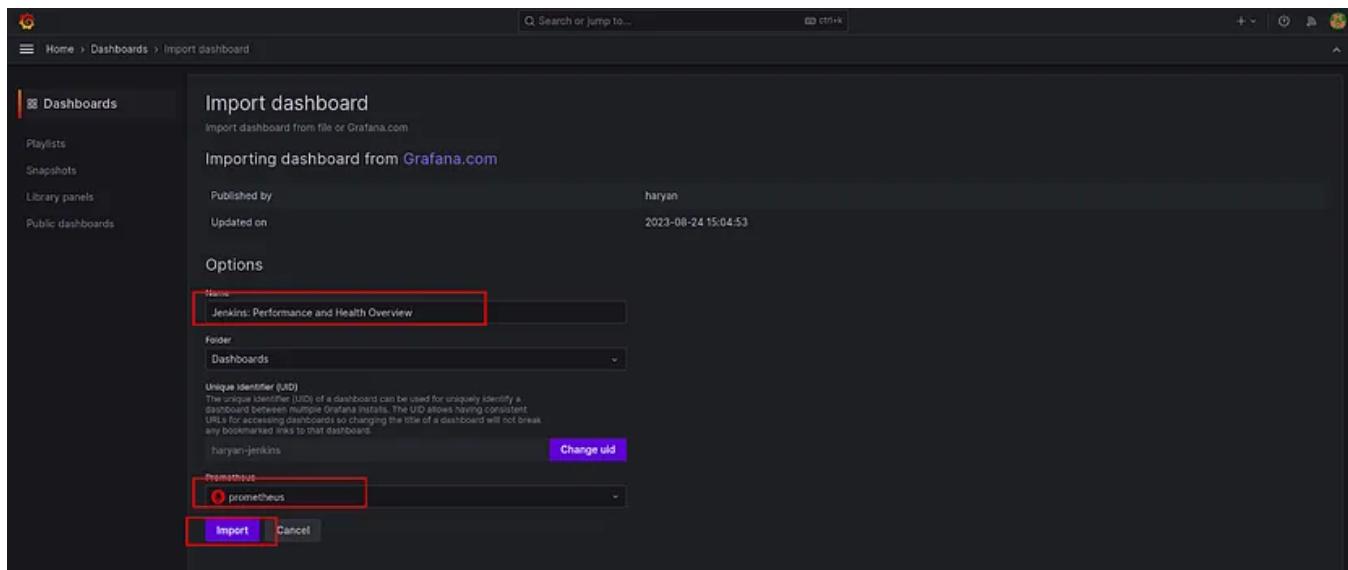
Click on New -> Import.

The screenshot shows the Grafana Dashboards page. On the right side, there is a purple "New" button with a dropdown menu. The "Import" option in this menu is highlighted with a red box.

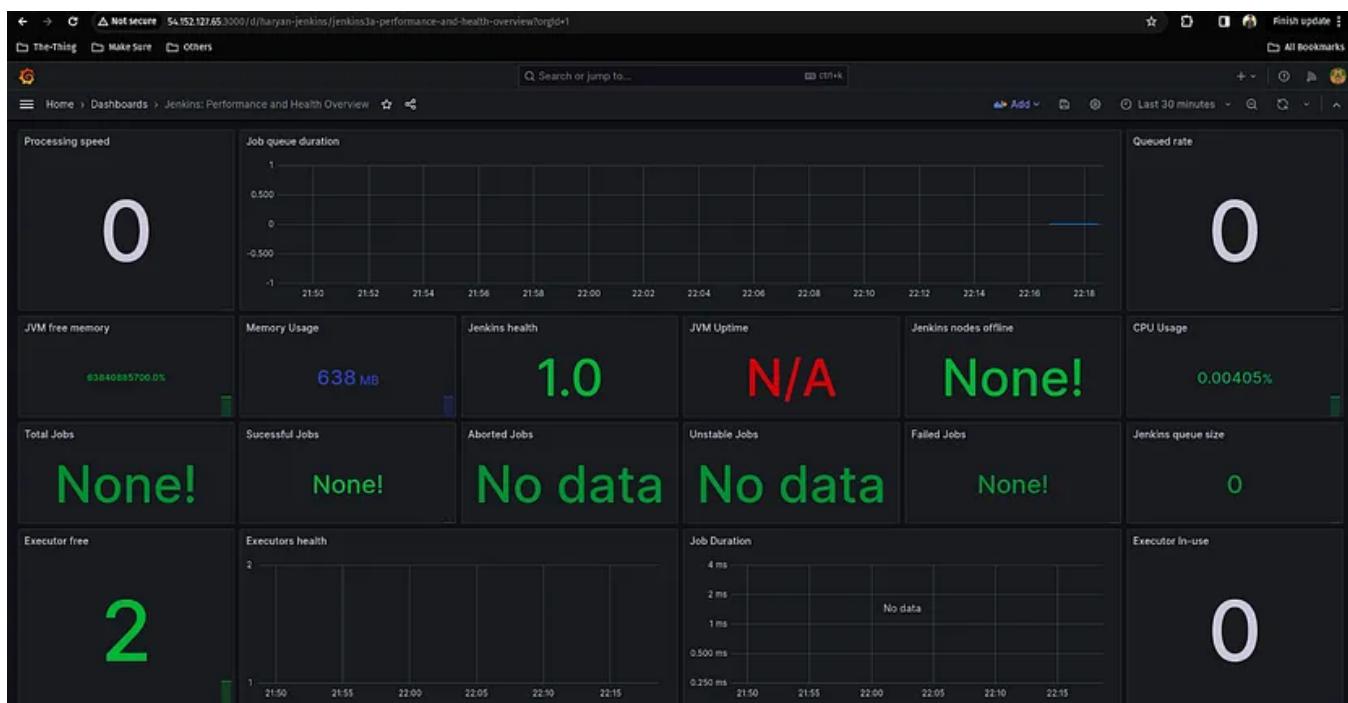
Provide the 9964 to Load the dashboard.

The screenshot shows the "Import dashboard" dialog. In the "Import dashboard from file or Grafana.com" section, the file path `9964` is highlighted with a red box. Below it, the "Load" button is also highlighted with a red box.

Select the default Prometheus from the drop-down menu and click on Import.



You will see your Jenkins Monitoring dashboard in the below snippet.



Now, we have to integrate Email Alert. So, if our Jenkins pipeline will succeed or fail we will get a notification alert on our email.

To do that, we need to install the Jenkins Plugin, whose name is **Email Extension Template**.

Manage Jenkins -> Plugins and install the **Email Extension Template** plugin.

The screenshot shows the Jenkins Plugins page. On the left sidebar, under the 'Available plugins' section, the 'Email Extension Template' plugin is listed. It has a green checkmark next to it, indicating it is installed. The plugin's name is 'Email Extension Template 1.5', and it is categorized under 'Build Notifiers' and 'email-ext'. A note below says, 'This plugin allows administrators to create global templates for the Extended Email Publisher.' A yellow banner at the bottom states, 'This plugin is up for adoption! We are looking for new maintainers. visit our [Adopt a Plugin](#) initiative for more information.' The status bar at the top right shows the user is 'admin'.

After installing the plugin, go to your email ID and click on Manage account and you will see what looks like the below snippet.

The screenshot shows the Google Account Security page. On the left sidebar, under the 'Security' section, there is a blue highlighted box. The main content area is titled 'Security' and includes a sub-section 'You have security recommendations' with a shield icon containing an exclamation mark. Below it is a 'Protect your account' button.

In the Security section, search for App passwords and click on it.

The screenshot shows the Google Account Security page with a search bar at the top containing 'app'. In the search results, the 'App passwords' link under 'Your connections to third-party apps & services' is highlighted with a red box. The main content area shows a 'Recent security activity' section with several entries:

- NoiseFit: Health & Fitness was given account access - Dec 21 - Maharashtra, India
- CRED was given account access - Dec 20 - Maharashtra, India
- New sign-in on OnePlus Nord2 5G - Dec 20 - Maharashtra, India

Gmail will prompt you for the password. Provide the password then you have to provide the name of your app where you are integrating email service.

[← App passwords](#)

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

[Learn more](#)

You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name

Jenkins

[Create](#)

You will get your password below. Copy the password and keep it secure somewhere.

The screenshot shows a modal dialog box titled "Generated app password". Inside the dialog, the text "Your app password for your device" is displayed, followed by a large, blurred 16-character password. Below this, under "How to use it", there is explanatory text and a "Done" button. On the left side of the dialog, there is a sidebar with the heading "Your app passwords" and a list containing "Jenkins". At the bottom of the sidebar, there are links for "To create a new app" and "App name". The background of the entire interface is a light gray color.

Add your email ID and the password that you have generated in the previous step.

Go to Manage Jenkins -> Credentials.

Click on (global).

Dashboard &gt; Manage Jenkins &gt; Credentials

**Credentials**

T	P	Store +	Domain	ID	Name
<b>Stores scoped to Jenkins</b>					
P	Store +	Domains			

System (global)

**Click on Add credentials**

Dashboard &gt; Manage Jenkins &gt; Credentials &gt; System &gt; Global credentials (unrestricted) &gt;

**Global credentials (unrestricted)**[+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty. How about <a href="#">adding some credentials?</a>			

Icon: S M L

**Select the Username with password in Kind.**

Provide your mail ID and generated password then provide the ID as mail to call both credentials.

Dashboard &gt; Manage Jenkins &gt; Credentials &gt; System &gt; Global credentials (unrestricted) &gt;

**New credentials**

Kind	<input type="text" value="Username with password"/>
Scope	<input type="text" value="Global (Jenkins, nodes, items, all child items, etc)"/>
Username	<input type="text" value="XXXXXXXXXX"/>
<input type="checkbox"/> Treat username as secret	
Password	<input type="password" value="*****"/>
ID	<input type="text" value="mail"/>
Description	<input type="text" value=""/>
<a href="#">Create</a>	

You can see we have added the credentials for the mail.

**Global credentials (unrestricted)**[+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

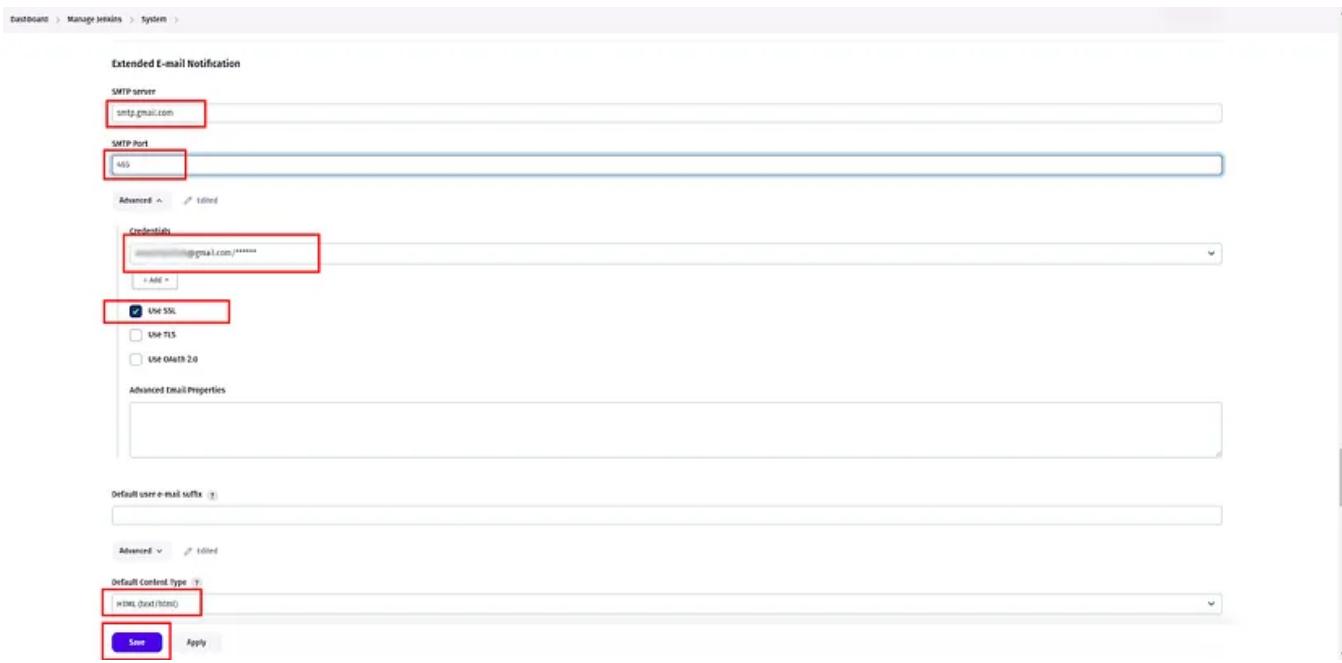
ID	Name	Kind	Description
mail	XXXXXXXXXX@gmail.com*****	Username with password	

Now, we have to configure our mail for the alerts.

Go to Jenkins -> Manage Jenkins -> System

Search for Extend E-mail Notification.

Provide the smtp.gmail.com in the **SMTP server** and 465 in the **SMTP port**.



Then, On the same page Search for Extend E-mail Notification.

Provide the smtp.gmail.com in the **SMTP server** and 465 in the **SMTP port**.

Select **Use SMTP Authentication** and provide the Gmail ID and its password in the **Username** and **password**.

To validate whether Jenkins can send the emails to you or not, check the **Test configuration** by sending a test e-mail.

E-mail Notification

SMTP server: smtp.gmail.com

Default user e-mail suffix:

Advanced ▾ Edited

Use SMTP Authentication

User Name: aman07pathak@gmail.com

Password: concealed

Use SSL

Use TLS

Subject: lets

Reply-To Address:

Charset: UTF-8

Test configuration by sending test e-mail

**Save** **Apply**

You can see below for the reference.

Test email #1 Inbox

 address not configured yet <aman07pathak@gmail.com>  
to me ▾

This is test email #1 sent from Jenkins

✉ Reply ✉ Forward 😊

Now, we will set up our Jenkins Pipeline. But there are some plugins required to work with them.

Download the following plugins

Eclipse Temurin installer

SonarQube Scanner

NodeJS

Plugins

Search: NodeJS

Install	Name	Released
<input checked="" type="checkbox"/>	Eclipse Temurin installer 1.5	1 yr 2 mo ago
<input checked="" type="checkbox"/>	SonarQube Scanner 2.16.1	2 mo 18 days ago
<input checked="" type="checkbox"/>	NodeJS 1.6.1	4 mo 12 days ago

Open in app [↗](#)

[Sign up](#) [Sign in](#)

Medium

Search

GO TO MANAGE JENKINS [↗](#) TOOLS

Click on Add JDK and provide the following things below

JDK installations

Add JDK

Name: jdk

Install automatically

Install from adoptium.net

Version: JDK 21+35

Add installer [▼](#)

Click on Add NodeJS and provide the following things below

NodeJS

Name: nodejs

Install automatically

Install from nodejs.org

Version: NodeJS 21.5.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail  Force 32bit architecture

Global npm packages to install  
Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Global npm packages refresh hours  
Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

Add installer [▼](#)

Now, we will configure Sonarqube

To access the sonarqube, copy the Jenkins Server public IP with port number 9000

Then, click Security and click on Users.

The screenshot shows the SonarQube Administration interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. The 'Administration' section is active. Under 'Configuration', the 'Security' tab is selected, and a red box highlights the 'Users' link. Other options like 'Groups', 'Edit global', 'Global Permissions', and 'Permission Templates' are also visible.

Click on the highlighted blue box on the right to generate the token.

The screenshot shows the SonarQube 'Users' page. The 'Tokens' tab is selected. A table lists tokens, including one named 'Administrator admin'. To the right of this token, there is a 'Tokens' section with a 'Generate' button, which is highlighted with a red box.

Now provide the name of your token and click on Generate.

The screenshot shows the 'Tokens of Administrator' page. In the 'Generate Tokens' section, the 'Name' field is set to 'end-to-end-kubernetes-project' and the 'Expires in' field is set to '30 days'. The 'Generate' button is highlighted with a red box.

Copy the generated token and keep it somewhere.

The screenshot shows the 'Tokens of Administrator' page after generating a token. A success message at the top states: 'New token "end-to-end-kubernetes-project" has been created. Make sure you copy it now, you won't be able to see it again!'. Below this, a 'Copy' button is highlighted with a red box, and the generated token value is displayed next to it.

Now, add the token to your Jenkins credentials

Go to Manage Jenkins -> Credentials.

Select the Secret text in Kind.

Provide your token then provide the ID as sonar-token to call the credentials.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

### New credentials

Kind	Secret Text
Scope	Global (Jenkins, nodes, items, all child items, etc)
Secret	sonar-token
ID	sonar-token
Description	sonar-token

**Create**

Go to Manage Jenkins -> System

Click on Add Sonarqube

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Add SonarQube

Provide the name sonar-server with the Server URL and select the credentials that we have added.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name	sonar-server
Server URL	Default is <a href="http://localhost:9000">http://localhost:9000</a> <a href="http://34.207.355.151:9000">http://34.207.355.151:9000</a>
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. sonar-token
<b>+ Add</b>	
Advanced	

Go to Manage Jenkins -> Tools

Find Sonarqube Scanner and click on Add

SonarQube Scanner installations

Add SonarQube Scanner

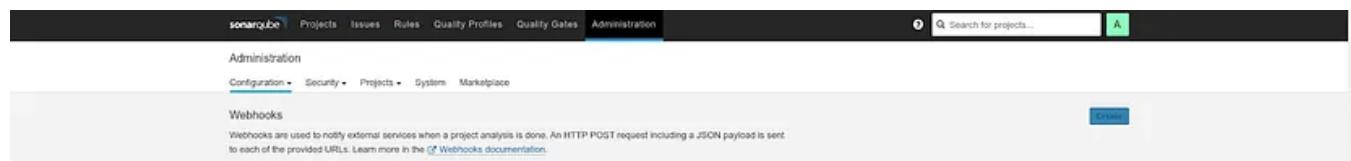
Provide the name sonar-server and select the latest version of Sonarqube.



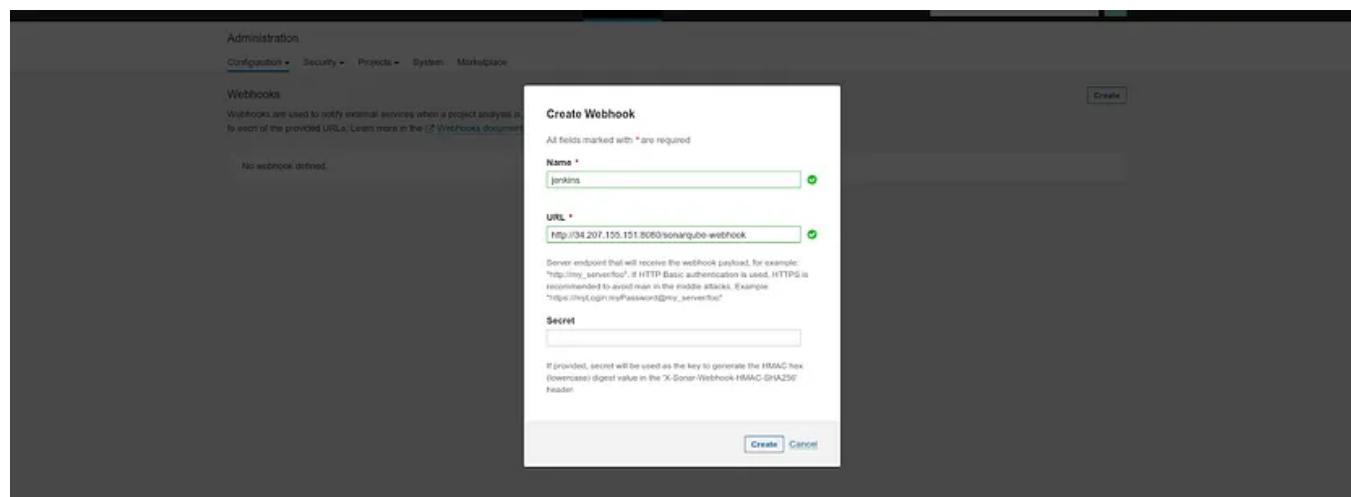
To create a webhook, click on Configuration and select Webhooks.



Click on Create.



Provide the name and Jenkins URL like below and click on Create.



The Webhook will be showing the below snippet.

Administration

Configuration • Security • Projects • System • Marketplace

Webhooks

Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the [Webhooks documentation](#).

Name	URL	Has secret?	Last delivery	Actions
jenkins	<a href="http://34.207.155.131:8080/sonarqube-webhook">http://34.207.155.131:8080/sonarqube-webhook</a>	No	Never	<a href="#"></a>

To create a project, click on **Manually**.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... [A](#)

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

From Azure DevOps	From Bitbucket Server	From Bitbucket Cloud	From GitHub	From GitLab
<a href="#">Set up global configuration</a>				

Are you just testing or have an advanced use-case? Create a project manually.

Manually
----------

**Important** 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

Provide the name of your project and click on **Set up**.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... [A](#)

Create a project

All fields marked with \* are required

Project display name \*

Up to 255 characters. Some scanners might override the value you provide.

Project key \*

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, "-", "\_", "(underscore)", ":" (colon) and ":" (colon), with at least one non-digit.

Main branch name \*

The name of your project's default branch [Learn More](#)

[Set Up](#)

**Important** 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

**Get the most out of SonarQube!**

- + Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

[Learn More](#) [Dismiss](#)

Select the existing token and click on **continue**.

Analyze your project

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

**1 Provide a token**

Generate a project token  
 Use existing token

**Existing token value**

`sqc_6f230e5c315841a486e7b667dd8436d3fc7c13eb`

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your user account.

**2 Run analysis on your project**

**Get the most out of SonarQube!**

- + Take advantage of the whole ecosystem by using SonarQube's free IDE plugins that help you find and fix issues earlier in your workflow.
- + Connect SonarQube to SonarCloud to sync rule sets and issue states.

[Learn More](#) [Dismiss](#)

Select the Other as your build and Linux as OS.

Analyze your project

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

**1 Provide a token**

**2 Run analysis on your project**

What option best describes your build?

Maven  
 Gradle  
 .NET  
 Other (for Jenkins, Travis CI, Go, Python, PHP, ...)

What is your OS?

Linux  
 Windows  
 macOS

**Get the most out of SonarQube!**

- + Take advantage of the whole ecosystem by using SonarQube's free IDE plugins that help you find and fix issues earlier in your workflow.
- + Connect SonarQube to SonarCloud to sync rule sets and issue states.

[Learn More](#) [Dismiss](#)

Now, we will create the Jenkins Pipeline

Click on Create item.

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

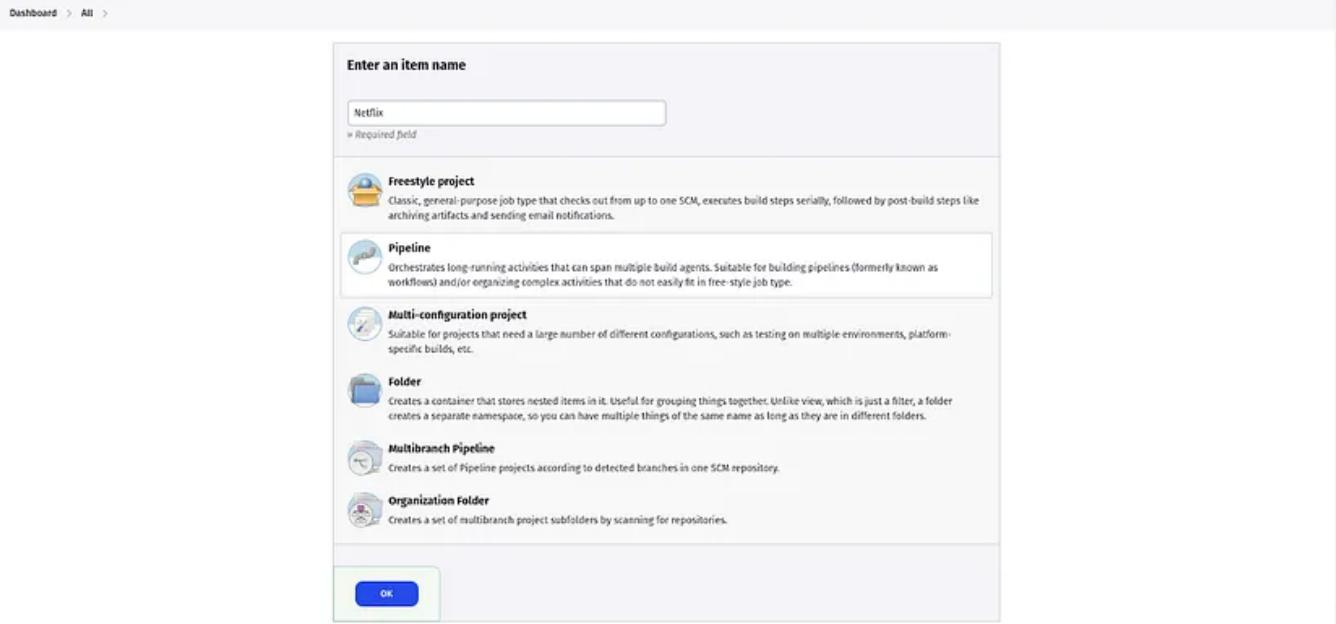
Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Provide the name of your Jenkins Pipeline and select Pipeline.



Currently, we are just creating a pipeline for Sonarqube analysis of the code, quality gate for Sonarqube, and installing the dependencies.

In the post-build, we have added email alerts for the success or failure of the pipeline.

```

pipeline{
    agent any
    tools{
        jdk 'jdk'
        nodejs 'nodejs'
    }
    environment {
        SCANNER_HOME=tool 'sonar-server'
    }
    stages {
        stage('Workspace Cleaning'){
            steps{
                cleanWs()
            }
        }
        stage('Checkout from Git'){
            steps{
                git branch: 'master', url: 'https://github.com/AmanPathak-DevOp'
            }
        }
        stage("Sonarqube Analysis"){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=
-Dsonar.projectKey=Netflix \

```

```

        }
    }
}

stage("Quality Gate"){
    steps {
        script {
            waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
        }
    }
}

stage('Install Dependencies') {
    steps {
        sh "npm install"
    }
}

post {
    always {
        emailext attachLog: true,
        subject: "${currentBuild.result}",
        body: "Project: ${env.JOB_NAME}<br/>" +
            "Build Number: ${env.BUILD_NUMBER}<br/>" +
            "URL: ${env.BUILD_URL}<br/>",
        to: 'aman07pathak@gmail.com',
        attachmentsPattern: 'trivyfs.txt,trivyimage.txt'
    }
}
}
}

```

The screenshot shows the Jenkins Pipeline configuration interface. On the left, there's a sidebar with 'Configure' and 'General' tabs, and a dropdown for 'Advanced Project Options'. The main area has tabs for 'Pipeline' (selected) and 'Definition'. Under 'Definition', a dropdown menu shows 'Pipeline script'. Below it is a large text area containing the Jenkinsfile code shown above. At the bottom of this area is a 'try sample Pipeline...' button. At the very bottom of the page are 'Save' and 'Apply' buttons.

Click on build pipeline and after getting the success of the pipeline.

You will see the Sonarqube code quality analysis which will look like the below snippet.

Now, we have to add one more tool for our application named OWASP Dependency-check.

Go to Manage Jenkins -> Plugins

Search for OWASP Dependency-Check and install it.

After installing, make sure to configure the OWASP.

Provide the name select the latest version of OWASP and click on Save.

Now, add the OWASP dependency check stage in the Jenkins pipeline and click on Save.

```
stage('OWASP DP SCAN') {
    steps {
        dependencyCheck additionalArguments: '--scan ./ --disableYarnAu
        dependencyCheckPublisher pattern: '**/dependency-check-report.x
    }
}

stage('TRIVY FS SCAN') {
    steps {
        sh "trivy fs . > trivyfs.txt"
    }
}
```

Now, click on Build Now.

The screenshot shows the Jenkins interface for the Netflix project. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, and Pipeline Syntax. The main area displays the Netflix logo and a green checkmark icon. A 'Dependency-Check Trend' chart is present, showing a single data point at index 1. Below it is a 'Stage View' table with columns: Declarative: Tool Install, clean workspace, Checkout from Git, Sonarqube Analysis, quality gate, Install Dependencies, OWASP DP SCAN, TRIVY FS SCAN, and Declarative: Post Actions. The table shows three stages with their respective times and status. The 'SonarQube Quality Gate' section indicates a failure with a red exclamation mark. The bottom right corner shows a 'Pipeline Syntax' link.

Once your pipeline is successful. Then, scroll down and you will see a dependency check.

Click on it and you will see the output like the below snippet.

The screenshot shows the Jenkins interface for a build named "Netflix #14". The left sidebar has a "Dependency-Check" link under "Pipeline Steps". The main content area is titled "Dependency-Check Results" and displays a "SEVERITY DISTRIBUTION" chart. Below the chart is a table with the following data:

File Name	Vulnerability	Severity	Weakness
postcss8.4.15	CVE-2023-44270	Medium	CWE-74
vite:3.2.2	CVE-2023-34092	High	CWE-706

In the bottom right corner of the page, it says "jenkins 2A3B".

Now, we have to build our Docker Image and push it to DockerHub

To do that, we need to configure the following things.

Go to Manage Jenkins -> Credentials

Add Docker Credentials to your Jenkins

The screenshot shows the Jenkins "Manage Jenkins" section with "Credentials" selected. The page title is "Global credentials (unrestricted)". It shows a table of credentials:

ID	Name	Kind	Description
mail	aman07pathak@gmail.com/*****	Username with password	
sonar-token	sonar-token	Secret text	sonar-token

At the bottom, there are links for "REST API" and "jenkins 2A3B".

Add your credentials and click on **Create**.

The screenshot shows the Jenkins 'New credentials' page under 'Manage Jenkins > System > Global credentials (unrestricted)'. A 'Username with password' credential is being created with the following fields:

- Kind:** Username with password
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Username:** avantiy
- Treat username as secret:** (unchecked)
- Password:** (redacted)
- ID:** docker
- Description:** (empty)

A blue 'Create' button is at the bottom left.

Install the following Docker plugins on your Jenkins

Docker  
Docker Commons  
Docker Pipeline  
Docker API  
docker-build-step

The screenshot shows the Jenkins 'Available plugins' page under 'Manage Jenkins > Plugins'. The search bar is set to 'Docker'. The 'Available plugins' tab is selected. The 'Docker' plugin is listed with the following details:

Install	Name	Released
<input checked="" type="checkbox"/>	Docker 1.5	3 mo 24 days ago
<input checked="" type="checkbox"/>	Docker Commons 479.via_3ch_0a_6a_1b_29	5 mo 20 days ago
<input checked="" type="checkbox"/>	Docker Pipeline 572.v950f50993843	4 mo 18 days ago
<input checked="" type="checkbox"/>	Docker API 3.3.a-86.v9b_a_Sede342c	29 days ago
<input checked="" type="checkbox"/>	docker-build-step 2.10	2 mo 24 days ago
<input type="checkbox"/>	CloudBees Docker Build and Publish 1.4.0	1 yr 6 mo ago

The 'Install' button is located at the top right of the table.

## Restart your Jenkins

**Plugins**

**Download progress**

Plugin	Status
Cloud Statistics	Success
Authentication Tokens API	Success
Docker Commons	Success
Apache HttpComponents Client 5.x API	Success
Docker API	Success
Docker	Success
Docker Commons	Success
Docker Pipeline	Success
Docker API	Success
Javadoc	Success
ISch dependency	Success
Maven Integration	Success
docker-build-step	Success
Loading plugin extensions	Success
Restarting Jenkins	Running

→ Go back to the top page  
(you can start using the installed plugins right away)

→  Restart Jenkins when installation is complete and no jobs are running

## Configure the tool in Jenkins

Go to Manage Jenkins -> Tools and provide the below details.

**Docker installations**

Add Docker

Name: docker

Install automatically

Download from docker.com

Docker version: latest

Add Installer

Save Apply

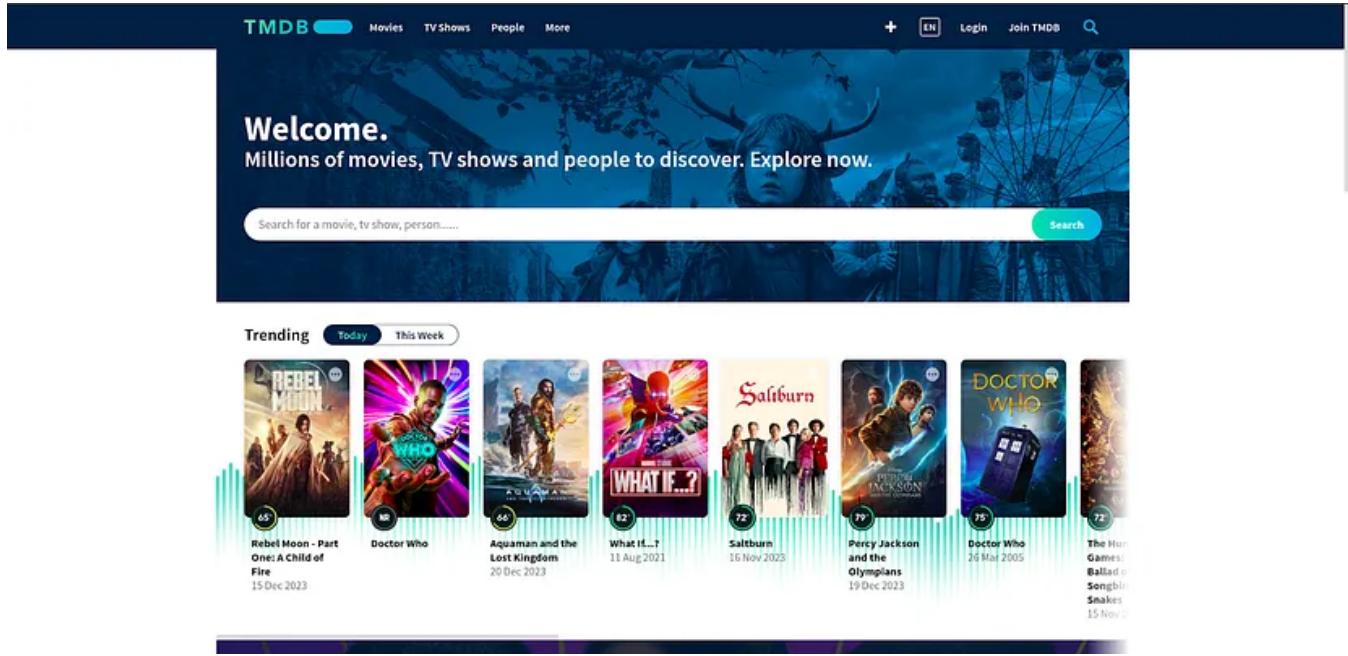
Our application is Netflix Clone. So we need some movie databases on our application.

For that, we have one application that will provide the API. So, we can use the API to get the movies on our application.

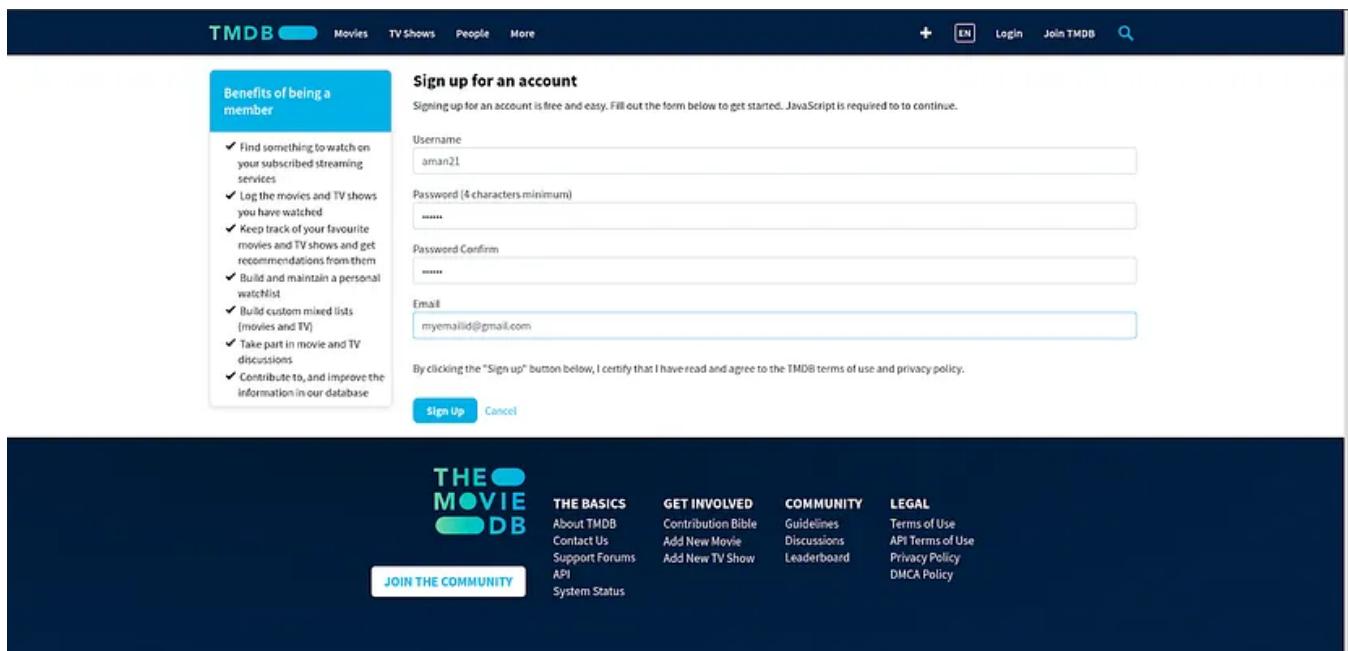
TMDB is one of them

Go to this link <https://www.themoviedb.org/>

Click on Join TMDB



Enter the details and click on SignUp



Once you sign up, you will get a confirmation email on your account. Confirm it.

Log in to your TMDB account and go to the settings.

The screenshot shows the TMDB homepage. At the top, there's a dark header with the TMDB logo and navigation links for Movies, TV Shows, People, and More. On the right side of the header, there's a user profile dropdown for 'sadmin902' with options like View profile, Discussions, Lists, Ratings, Watchlist, Edit Profile, Settings (which is highlighted with a red box), and Logout. Below the header, a large 'Welcome.' banner says 'Millions of movies, TV shows and people to discover. Explore now.' followed by a search bar with placeholder text 'Search for a movie, tv show, person.....'. Underneath the banner, there's a 'Trending' section with cards for various movies and TV shows, each with a small thumbnail, a title, and a rating. The cards include 'Rebel Moon - Part One: A Child of Fire' (85), 'Doctor Who' (86), 'Aquaman and the Lost Kingdom' (82), 'What If...?' (82), 'Saltburn' (72), 'Percy Jackson and the Olympians' (79), 'Doctor Who' (75), and 'The Hunger Games: Ballad of Songbirds and Snakes' (72). Below the trending section, there's a URL in the address bar: <https://www.themoviedb.org/settings/account>.

Go to the API section.

The screenshot shows the 'Account Settings' page for the user 'sadmin902'. The left sidebar has a 'Settings' tab selected, with sub-options: Edit Profile, Account Settings (which is highlighted with a red box), Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, Sessions, API, and Delete Account. The main content area is titled 'Account Settings' and contains several configuration fields with dropdown menus and checkboxes. These include: Default Language (English (en-GB)), Fallback Language (None (Don't Fallback)), Country (India), Timezone - Auto detect? (Asia - Kabul), Include Adult Items in Search? (No), Filter Profanity? (Yes), and Enable Keyboard Shortcuts? (Yes). At the bottom of the form is a 'Save' button.

Click on Create to generate API

**Settings**

- [Edit Profile](#)
- [Account Settings](#)
- [Streaming Services](#)
- [Notification Settings](#)
- [Blocked Users](#)
- [Import List](#)
- [Sharing Settings](#)
- [Sessions](#)
- [API](#)

**API** Overview Create

TMDb offers a powerful API service that is free to use as long as you properly attribute us as the source of the data and/or images you use. You can find the logos for attribution [here](#).

**Documentation**

Our primary documentation is located at [developer.themoviedb.org](#).

**Support**

If you have questions or comments about the information covered here, please create a post on our [support forums](#).

**Request an API Key**

To generate a new API key, click [here](#).

## Select Developer.

**Settings**

- [Edit Profile](#)
- [Account Settings](#)
- [Streaming Services](#)
- [Notification Settings](#)
- [Blocked Users](#)
- [Import List](#)

**API** Overview Create

What type of API key do you wish to register?

**Developer**

- You are an individual
- Your project is still in development
- Your project is non profit
- Your project is ad supported

**Professional**

- You represent a company
- Your project is for profit (not ad supported)
- You are an OEM or hardware vendor

## Accept the Terms & Conditions.

**12. General Terms**

- 1. Relationship of the Parties.** Notwithstanding any provision hereof, for all purposes of the Terms of Use, you and TMDb shall be and act independently and not as partner, joint venturer, agent, employee or employer of the other. You shall not have any authority to assume or create any obligation for or on behalf of TMDB, express or implied, and you shall not attempt to bind TMDB to any contract.
- 2. Invalidity of Specific Terms.** If any provision of the Terms of Use is found by a court of competent jurisdiction to be invalid, the parties nevertheless agree that the other provisions of this agreement will remain in full force and effect and the court should endeavor to give effect to the parties' intentions as reflected in the invalid provision.
- 3. Location of Lawsuit and Choice of Law.** THE TERMS OF USE AND THE RELATIONSHIP BETWEEN YOU AND TMDB SHALL BE GOVERNED BY THE LAWS OF THE STATE OF CALIFORNIA WITHOUT REGARD TO ITS CONFLICT OF LAW PROVISIONS. YOU AND TMDB AGREE TO SUBMIT TO THE PERSONAL JURISDICTION OF THE COURTS LOCATED WITHIN THE COUNTY OF SAN MATEO, CALIFORNIA.
- 4. No Waiver of Rights by TMDB.** TMDB's failure to exercise or enforce any right or provision of the Terms of Use shall not constitute a waiver of such right or provision.
- 5. No Transfer.** The rights and obligations of these Terms of Use is personal to you and may not be transferred by you, either voluntarily or by operation of law.
- 6. Notice.** Any notice to be sent to you under these Terms of Use may be sent via email, post, or any other reasonable means, at the contact information provided by you to TMDB from time to time. It is your obligation to insure that this information is current.

**Miscellaneous.** The section headings and subheadings contained in this agreement are included for convenience only, and shall not limit or otherwise affect the terms of the Terms of Use. Any construction or interpretation to be made of the Terms of Use shall not be construed against the drafter. The Terms of Use constitute the entire agreement between TMDB and you with respect to the subject matter hereof.

This Agreement was last updated on: July 28, 2014.

Cancel
Accept

Provide the basic details and click on **Submit**.

Sharing Settings

Sessions

**API**

Delete Account

First Name: Code

Last Name: Word

Email Address: [REDACTED]

Address 1: dddddd

City: ffffffssss

Zip Code: 225588

Phone Number (no parenthesis or dashes):

- Mali
- Malta
- Marshall Islands
- Martinique
- Mauritania
- Mauritius**
- Mavotte
- India

**Submit**

After clicking on **Submit**. You will get your **API**. Copy the API and keep it somewhere.

TMDB

Movies TV Shows People More

sadmin902

**API** Overview Details Sessions Stats Regenerate Key

TMDB offers a powerful API service that is free to use as long as you properly attribute us as the source of the data and/or images you use. You can find the logos for attribution [here](#).

**Documentation**

Our primary documentation is located at [developer.themoviedb.org](#).

**Support**

If you have questions or comments about the information covered here, please create a post on our [support forums](#).

**API Details**

If you'd like to edit the details of your app, [click here](#).

**API Key**

8b174e589e2039fd8123907bd7800c

**API Read Access Token**

eyJhbGciOiJIUzI1NiJ9eyJndWQjOihYJ3NGU1ODIiMmYwM2Y5ZmQ4MTIxOTA3ImQ3ODAwYyIsInNlYiI6YjY1DGE5N2RlZGQyNTg5NzFhZTJMDy5NCisInNjb3BicyI6WyjheGfcmv%2CjdLCjZXJzaW9uIjoxQVvdSeuE3xTGnPkYiIB6GgtYTb3vFL7\_QWb3EBUk4

THE MOVIE DB THE BASICS GET INVOLVED COMMUNITY LEGAL

Now, we have to configure our Docker images where we will build our image with the help of new code and then, push it to DockerHub.

After pushing the image, we will scan our DockerHub Image to find the vulnerabilities in the image.

Make sure to replace the API with your API and if you are pushing Dockerfile on your Dockerhub account then, replace my username of the Dockerhub with yours.

Dashboard > Netflix > Configuration

## Pipeline

### Configure

General Advanced Project Options Pipeline

#### Definition

Pipeline script

```
script {
    waitForQualityGate abortIf(pipeline) false, credentialsId: 'sonar-token'
}
stage("Install Dependencies") {
    steps {
        sh "npm install"
    }
}
stage("NPM DP SCAN") {
    steps {
        dependencyCheck additionalArguments: "-scans ./ --disableStandard --disableNodeAudit", osInstallation: 'osx-10.14-x64', dependenciesCheckPublisher: 'dependency-check-report.html'
    }
}
stage("TRIVY FS SCAN") {
    steps {
        sh "trivy fs .. + trivyfs.txt"
    }
}
stage("Docker Build & Push") {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {
                sh "docker build -t netflix/trivy:latest -f Dockerfile.trivy ."
                sh "docker tag netflix/trivy:latest netflix/trivy:latest"
                sh "docker push netflix/trivy:latest"
            }
        }
    }
}
stage("Deploy") {
    steps {
        sh "trivy image avionis/netflixlatest > trivyImage.txt"
    }
}
post {
    always {
        emailex.attachLogs(true)
        emailex.subject("Deployment result")
        emailex.body("Project: ${env.PWD}/")
        emailex.to("trivyservice@netflix.com")
        emailex.cc("trivyservice@netflix.com")
        emailex.attachmentsPath("trivyfs.txt,trivyImage.txt")
    }
}
}
```

Save Apply

## Click on Build

The screenshot shows the Jenkins interface for the Netflix project. The left sidebar contains links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, and Pipeline Syntax. The main area displays the Netflix pipeline status with a green checkmark icon. A 'Dependency-Check Trend' chart is present, showing a red line with markers for Unassigned, Low, Medium, High, and Critical levels. The 'Stage View' section shows stages: Declarative: Tool Install (223ms), clean workspace (512ms), Checkout from Git (1s), Sonarqube Analysis (21s), quality gate (530ms), Install Dependencies (16s), OWASP DP SCAN (16min 45s), TRIVY FS SCAN (26s), and Declarative: Post Actions (134ms). The SonarQube Quality Gate stage shows a failure with a red circle and 'no changes' message. The bottom navigation bar includes links for Jenkins, Help, and Log Out.

As you can see Our Pipeline is successful.

The screenshot shows the Jenkins Pipeline Stage View for the Netflix project. On the left, there's a sidebar with options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, and Pipeline Syntax. The main area is titled "Stage View" and displays a grid of build stages. Each stage has a title, duration, and status. A legend at the bottom indicates the status colors: Unassigned (grey), Low (green), Medium (yellow), High (orange), and Critical (red). A dependency check trend chart is located at the top right, showing a timeline from #13 to #19 with a color scale from grey to red.

Now, validate whether the docker image has been pushed to DockerHub or not.

Log in to your Dockerhub account.

As you can see in the below screenshot, Our Docker image is present on Docker Hub.

The screenshot shows the DockerHub interface. The top navigation bar includes links for Explore, Repositories (which is selected), Organizations, and a search bar. Below the navigation, there's a list of repositories under the user 'avian19'. The repository 'avian19 / netflix' is highlighted with a red box. Other repositories listed are 'avian19 / mysql', 'avian19 / backend-image', 'avian19 / frontend-image', 'avian19 / python-flask-application', 'avian19 / choco-shop-order', and 'avian19 / choco-shop-reviews'. Each repository entry shows its status (inactive), number of stars, and whether it's public. To the right of the repository list, there are promotional banners for creating organizations and a community all-hands event.

Now, we have to deploy our application using Kubernetes.

To do that, we need to install kubectl on the Jenkins server.

```

sudo apt update
sudo apt install curl
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable)
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client

```

```

ubuntu@ip-172-31-59-9:~$ sudo apt update
sudo apt install curl
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable)/bin/linux/amd64/kubectl
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [110 kB]
Hit:4 https://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:5 https://squasecurity.github.io/trivy-repo/deb jammy InRelease
Ign:6 https://pkg.jenkins.io/debian binary/ InRelease
Hit:7 https://pkg.jenkins.io/debian binary/ Release
Fetched 229 kB in 1s (304 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://squasecurity.github.io/trivy-repo/deb/dists/jammy: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
% Total    % Received % Xferd  Average Speed   Time   Current
          Dload  Upload   Total Spent   Left  Speed
100  138  100  138    0     0  1132    0 ---:---:---:---:---:--- 1140
100 47.4M  100 47.4M   0     0  76.0M    0 ---:---:---:---:---:--- 76.0M
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
ubuntu@ip-172-31-59-9:~$ 
ubuntu@ip-172-31-59-9:~$ kubectl --version
error: unknown flag: --version
See 'kubectl --help' for usage.
ubuntu@ip-172-31-59-9:~$ kubectl version
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fversion%3Ftimeout%3D32s'><script id='redirect' data-redirect-url='/login?from=%2Fversion%3Ftimeout%3D32s' src='/static/f25a94ce/scripts/redirect.js'></script></head><body style='background-color:white; color:white;'>
Authentication required
<!--
-->
</body></html>
ubuntu@ip-172-31-59-9:~$ 

```

As you know, we have two Kubernetes Nodes of which one is the Master and the other one is the Worker Node.

Login to your both Kubernetes Master and Worker Nodes

## Master Node

```
amanpathak@pop-os:~/Downloads$ ssh -i "Aman-Pathak.pem" ubuntu@ec2-52-91-122-66.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-122-66.compute-1.amazonaws.com (64:f9b:345b:7642)' can't be established.
ED25519 key fingerprint is SHA256:0xteG2IqHk3JMcCY1SD8VLOUHLyFMKE9DjttabGE=0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-122-66.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

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

System information as of Thu Dec 28 06:11:01 UTC 2023

System load: 0.0 Processes: 106
Usage of /: 23.5% of 7.57GB Users logged in: 0
Memory usage: 5% IPv4 address for eth0: 172.31.59.154
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
19 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

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-172-31-59-154:~$ 
```

## Worker Node

```
amanpathak@pop-os:~/Downloads$ ssh -i "Aman-Pathak.pem" ubuntu@ec2-3-98-84-57.compute-1.amazonaws.com
The authenticity of host 'ec2-3-98-84-57.compute-1.amazonaws.com (64:f9b:35a:5439)' can't be established.
ED25519 key fingerprint is SHA256:wNyngpVNhDP5hrasUz5grS/Wf0gTszX6xCa+paub0s.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-98-84-57.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

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

System information as of Thu Dec 28 06:11:50 UTC 2023

System load: 0.088078125 Processes: 105
Usage of /: 23.5% of 7.57GB Users logged in: 0
Memory usage: 5% IPv4 address for eth0: 172.31.52.140
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
19 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

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-172-31-52-140:~$ 
```

Add the hostname to your Kubernetes master node

```
sudo hostnamectl set-hostname K8s-Master
```

```
ubuntu@ip-172-31-59-154:~$ sudo hostnamectl set-hostname K8s-Master
ubuntu@ip-172-31-59-154:~$ 
```

Add the hostname to your Kubernetes worker node

```
sudo hostnamectl set-hostname K8s-Worker
```

```
ubuntu@ip-172-31-52-140:~$ sudo hostnamectl set-hostname K8s-Worker
ubuntu@ip-172-31-52-140:~$
```

Run the below commands on the both Master and worker Nodes.

```
sudo su
swapoff -a; sed -i '/swap/d' /etc/fstab
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
# Apply sysctl params without reboot
sudo sysctl - system
apt update
sudo apt-get install -y apt-transport-https ca-certificates curl
curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo gpg - c
echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://
apt update
apt-get install -y kubelet kubeadm kubectl kubernetes-cni
apt install docker.io -y
sudo mkdir /etc/containerd
sudo sh -c "containerd config default > /etc/containerd/config.toml"
sudo sed -i 's/ SystemdCgroup = false/ SystemdCgroup = true/' /etc/containerd/c
systemctl restart containerd.service
systemctl restart kubelet.service
systemctl enable kubelet.service
```

Now, run the following commands **Only on the Master Node**, and then you will get the command that is highlighted in the below snippet.

```
kubeadm config images pull
kubeadm init
```

```
[kubecfg] Writing "controller-manager.conf" kubeconfig file
[kubecfg] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the Kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 11.002665 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node k8s-master as control-plane by adding the labels: {node-role.kubernetes.io/control-plane:control-plane}
[bootstrapstrap-token] Using token: deq9nl.y34go2zii0fu8c1
[bootstrapstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrapstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrapstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrapstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrapstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrapstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:
kubeadm join 172.31.59.154:6443 --token deq9nl.y34go2zii0fu8c1 \
--discovery-token-ca-cert-hash sha256:e93c56bd59b175b81845a671a82ffd1839e42272d922f9c43ca8d8f6d145ce02
root@K8s-Master:/home/ubuntu# ]
```

Exit from the root user and run the below commands

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
ubuntu@ip-172-31-59-154:~$ mkdir -p $HOME/.kube
ubuntu@ip-172-31-59-154:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
ubuntu@ip-172-31-59-154:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-59-154:~$ ls .kube/
config
ubuntu@ip-172-31-59-154:~$ ]
```

## Run on the Worker Node

Run the below command as a root user

```
kubeadm join 172.31.59.154:6443 --token deq9nl.y34go2zii0fu8c1 \
--discovery-token-ca-cert-hash sha256:e93c56bd59b175b81845a671a82ffd1839e42272
```

```
root@K8s-Worker:/home/ubuntu# kubeadm join 172.31.59.154:6443 --token deq9n1.y34pq2ii1i0fu8c1 \
--discovery-token-ca-cert-hash sha256:e93c56bd50b175b81845a671a827fd1830e42272d022f0c43ca8d8f6d145ce02
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@K8s-Worker:/home/ubuntu#
```

Both nodes are not ready because the network plugin is not installed on the master node

```
ubuntu@K8s-Master:~$ kubectl get nodes
NAME     STATUS   ROLES    AGE     VERSION
k8s-master   NotReady   control-plane   9m30s   v1.28.2
k8s-worker   NotReady   <none>      56s     v1.28.2
ubuntu@K8s-Master:~$
```

## Only on the Master Node

Run the below command to install the network plugin on the Master node

```
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0
```

```
ubuntu@K8s-Master:~$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
configmap/calico-config created
customresourcedefinition.apirextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/caliconodesstatus.es.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/globornetworksets.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/iplockss.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ippoolconfigs.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ippoolranges.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/irspools.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/irspoolranges.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@K8s-Master:~$
```

Both nodes are ready.

```
ubuntu@K8s-Master:~$ kubectl get nodes
NAME     STATUS   ROLES    AGE     VERSION
k8s-master   Ready   control-plane   10m     v1.28.2
k8s-worker   Ready   <none>      2m22s   v1.28.2
ubuntu@K8s-Master:~$
```

Install the following Kubernetes Plugins on your Jenkins

Kubernetes  
Kubernetes Credentials

## Kubernetes Client API

## Kubernetes CLI

## Kubernetes Credential Provider

The screenshot shows the Jenkins Plugins page with the search bar set to "Kubernetes". The left sidebar has "Available plugins" selected. The main table lists several Kubernetes-related plugins:

Plugin Name	Description	Released
Kubernetes Client API	Library plugins (for use by other plugins)	4 mo 0 days ago
Kubernetes Credentials	Common classes for Kubernetes credentials	4 mo 0 days ago
Kubernetes	This plugin integrates Jenkins with Kubernetes	9 days ago
Kubernetes CLI	Configure Kubectl for Kubernetes	3 mo 29 days ago
Kubernetes Credentials Provider	Provides a read only credentials store backed by Kubernetes.	2 mo 10 days ago
Kubernetes :: Pipeline :: DevOps Steps	Pipeline - kubernetes	4 yr 11 mo ago
Google Kubernetes Engine	This plugin allows Jenkins to deploy build artifacts to a Kubernetes cluster running on GKE.	8 days 1 hr ago

The screenshot shows the Jenkins Plugins page with the search bar set to "Kubernetes". The left sidebar has "Available plugins" selected. The main table lists several Kubernetes-related plugins, including some from the previous list and new ones:

Plugin Name	Description	Released
Kubernetes Credentials Provider	Provides a read only credentials store backed by Kubernetes.	2 mo 10 days ago
Kubernetes :: Pipeline :: DevOps Steps	Pipeline - kubernetes	4 yr 11 mo ago
Google Kubernetes Engine	This plugin allows Jenkins to deploy build artifacts to a Kubernetes cluster running on GKE.	8 days 1 hr ago
GitLab Credentials - Kubernetes Integration	Integrates gitlabToken credential type from the gitlab-branch-source-plugin with the k8s credential provider.	2 mo 10 days ago
OpenShift Pipeline	This plugin facilitates the construction of jobs, pipelines, and workflows that operate on a Kubernetes based OpenShift server.	3 yr 9 mo ago
Azure Container Service	Deploy Kubernetes, DC/OS, Docker Swarm application configurations to Azure Container Service cluster.	2 yr 10 mo ago

Now, we will set Kubernetes Monitoring for both Master and worker Nodes

Run the below command on both Kubernetes Nodes

```
sudo useradd \
--system \
```

```
--no-create-home \
--shell /bin/false prometheus
```

```
ubuntu@K8s-Master:~$ sudo useradd \
...system \
--no-create-home \
--shell /bin/false node_exporter
ubuntu@K8s-Master:~$
```

Download the node exporter package on both Kubernetes Nodes and Untar the node exporter package file and move the node\_exporter directory to the /usr/local/bin directory

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_
tar -xvf node_exporter-1.7.0.linux-amd64.tar.gz
sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
```

```
ubuntu@K8s-Master:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz
--2023-12-28 10:24:31-- https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 304 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/01323270-6ecb-47aa-813f-52a2a89cdc64?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWHJ
YAX4CSVEH53A4F2F0231228%2Fus-east-1%2F53%2Faws4_request&X-Amz-Date=20231228T102235Z&X-Amz-Expires=3006X-Amz-Signature=6fbef0d31f4bf2f847c4b26939efef7929439c3bda4759bbde92c385a467c6146X-Amz-S
ignedHeaders=host&actor_id=0&repo_id=9524057&response-content-disposition=attachment%3Bfilename%3Dnode_exporter-1.7.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-
-stream [following]
--2023-12-28 10:24:31-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/01323270-6ecb-47aa-813f-52a2a89cdc64?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Cred
ential=AKIAIWHJYAX4CSVEH53A4F2F0231228%2Fus-east-1%2F53%2Faws4_request&X-Amz-Date=20231228T102235Z&X-Amz-Expires=3006X-Amz-Signature=6fbef0d31f4bf2f847c4b26939efef7929439c3bda4759bbde92c385a467c6146X-Amz-S
ignedHeaders=host&actor_id=0&repo_id=9524057&response-content-disposition=attachment%3Bfilename%3Dnode_exporter-1.7.0.linux-amd64.tar.gz&response-content-type=appl
ication%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10419253 (9.8M) [application/octet-stream]
Saving to: 'node_exporter-1.7.0.linux-amd64.tar.gz'

node_exporter-1.7.0.linux-amd64.tar.gz    100%[=====]  9.94M  ---KB/s   in 0.07s

2023-12-28 10:24:31 (143 MB/s) - 'node_exporter-1.7.0.linux-amd64.tar.gz' saved [10419253/10419253]
ubuntu@K8s-Master:~$ ls
node_exporter-1.7.0.linux-amd64.tar.gz
ubuntu@K8s-Master:~$ tar -xvf node_exporter-1.7.0.linux-amd64.tar.gz
node_exporter-1.7.0.linux-amd64/
node_exporter-1.7.0.linux-amd64/LICENSE
node_exporter-1.7.0.linux-amd64/node_exporter
node_exporter-1.7.0.linux-amd64/NOTICE
ubuntu@K8s-Master:~$
```

Create the systemd configuration file for node exporter.

Edit the file

```
sudo vim /etc/systemd/system/node_exporter.service
```

Copy the below configurations and paste them into the /etc/systemd/system/node\_exporter.service file.

```
[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
]
```

Enable the node exporter systemd configuration file and start it.

```
sudo systemctl enable node_exporter
sudo systemctl enable node_exporter
systemctl status node_exporter.service
```

```
ubuntu@K8s-Master:~$ sudo vim /etc/systemd/system/node_exporter.service
ubuntu@K8s-Master:~$ 
ubuntu@K8s-Master:~$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
ubuntu@K8s-Master:~$ 
ubuntu@K8s-Master:~$ sudo systemctl start node_exporter
ubuntu@K8s-Master:~$ 
ubuntu@K8s-Master:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-12-28 10:26:31 UTC; 8s ago
       Main PID: 133987 (node_exporter)
          Tasks: 5 (limit: 4667)
         Memory: 2.0M
            CPU: 9ms
           CGroup: /system.slice/node_exporter.service
                   ↳ 133987 /usr/local/bin/node_exporter --collector.logind

Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.814Z caller=node_exporter.go:117 level=info collector=thermal_zone
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.814Z caller=node_exporter.go:117 level=info collector=cpu_time
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.814Z caller=node_exporter.go:117 level=info collector=timex
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.814Z caller=node_exporter.go:117 level=info collector=udp_queues
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.815Z caller=node_exporter.go:117 level=info collector=uname
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.815Z caller=node_exporter.go:117 level=info collector=vmstat
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.815Z caller=node_exporter.go:117 level=info collector=xfs
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.815Z caller=node_exporter.go:117 level=info collector=zfs
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.815Z caller=tls_config.go:274 level=info msg="Listening on" address=[::]:9100
Dec 28 10:26:31 K8s-Master node_exporter[133987]: ts=2023-12-28T10:26:31.816Z caller=tls_config.go:277 level=info msg="TLS is disabled." http2=false address=[::]:9100
ubuntu@K8s-Master:~$ 
```

Now, we have to add a node exporter to our Prometheus target section. So, we will be able to monitor both Kubernetes Servers.

edit the file

```
sudo vim /etc/prometheus/prometheus.yml
```

Add both job names(Master & Worker nodes) with their respective public.

```
# 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:
# Here it's Prometheus itself.
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: ["34.207.155.151:8080"]
  - job_name: node_export_masterk8s
    static_configs:
      - targets: ["52.93.122.66:9100"]
  - job_name: node_export_workerk8s
    static_configs:
      - targets: ["3.90.84.57:9100"]
```

After saving the file, validate the changes that you have made using promtool.

```
promtool check config /etc/prometheus/prometheus.yml
```

If your changes have been validated then, push the changes to the Prometheus server.

```
curl -X POST http://localhost:9090/-/reload
```

```
ubuntu@ip-172-31-52-7:~$ vim /etc/prometheus/prometheus.yml
ubuntu@ip-172-31-52-7:~$ 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-172-31-52-7:~$ curl -X POST http://localhost:9090/-/reload
ubuntu@ip-172-31-52-7:~$ [ ]
```

As you know, Jenkins will deploy our application on the Kubernetes Cluster. To do that, Jenkins must have the access keys or something to connect with the master node.

To do that copy the content inside .kube/config on Kubernetes Master node.

```
cat .kube/config
```

Save the file with the .txt extension.

```
*Untitled Document 1                               secret-file.txt

1 apiVersion: v1
2 clusters:
3 - cluster:
4   certificate-authority-data:
5     LS0tLS1CRUdJTiBvR0VJUzU2Q0FRS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ01J5FlsQ991T18wUD83RFFZ5ktWkIodmN0QVFTEjRQxdGVETUJF0ExVUUKQxHNS2EzVmIaMpEp1WlhsbGn6QWVGdz8STXpFeU1qZ3d0akV6TkRwYUZ3MhPNekv5TR0RFZhTUJVea
6   server: https://37.31.59.154:6443
7   name: kubernetes
8 contexts:
9 - context:
10   cluster: kubernetes
11   user: kubernetes-admin
12   name: kubernetes-admin@kubernetes
13 current-context: kubernetes-admin@kubernetes
14 kind: Config
15 preferences: {}
16 users:
17 - name: kubernetes-admin
18   user:
19     client-certificate-data:
20       LS0tLS1CRUdJTiBvR0VJUzU2Q0FRS0tLS0tCk1JSURCVENDQWd0T0F3SUJBZ01J3R0RGc2lkIcdN0Nh3RFFZ5ktWkIodmN0QVFTEjRQxdGVETUJF0ExVUUKQxHNS2EzVmIaMpEp1WlhsbGn6QWVGdz8STXpFeU1qZ3d0akV6TkRwYUZ3MhL0REv5Twj0d5qRT0RghhTURReA
21     client-key-data:
22       LS0tLS1CRUdJTiBvR0VJUzU2Q0FRS0tLS0tCk1JSURCVENDQWd0T0F3SUJBZ01J3QFB50NUUVBdm9rRklFRVtLek53NmtlNMDIUGhGlzJiejhxWtJ3XVobFB1t19MUURY0WxzU3c4CkTveVlWehDjYK9wsQ4NTSuJvHnmJBL215Q1pnZFGQnfPRTNBL1I3VVE3WUfJKCJM20EbkkxT0VpnTm
```

Now, add the Secret file in Jenkins Credentials.

Click on Add credentials.

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

ID	Name	Kind	Description
mail	aman0pathak@gmail.com/*****	Username with password	
sonar-token	sonar-token	Secret text	sonar-token
docker	avian19/*****	Username with password	

A purple "Add Credentials" button is visible in the top right corner.

Select the **Secret file** and provide the Secret file that you have saved earlier enter the ID k8s then click on **Create**.

The screenshot shows the "New credentials" creation form. The fields are:

- Kind:** Secret file (highlighted with a red box)
- Scope:** Global (Jenkins, nodes, items, all child items, etc.)
- file:** secret-file.txt (highlighted with a red box)
- ID:** k8s (highlighted with a red box)
- Description:** (empty)

A purple "Create" button is at the bottom.

Now, Add the deploy to the Kubernetes stage in your Jenkins pipeline.

```
stage('Deploy to Kubernetes'){
    steps{
        script{
            dir('Kubernetes') {
                withKubeConfig(caCertificate: '', clusterName: '', contextName: '') {
                    sh 'kubectl apply -f deployment.yml'
                    sh 'kubectl apply -f service.yml'
                    sh 'kubectl get svc'
                    sh 'kubectl get all'
                }
            }
        }
    }
}
```

```

    }
}

```

Dashboard > Netflix > Configuration

### Pipeline

**Configure**

- General
- Advanced Project Options
- Pipeline**

**Definition**

**Script**

```

44
45
46
47
48+
49+
50+
51+
52+
53+
54+
55+
56+
57+
58+
59+
60+
61+
62+
63+
64+
65+
66+
67+
68+
69+
70+
71+
72+
73+
74+
75+
76+
77+
78+
79+
80+
81+
82+
83+
84+
85+
86+
87+
88+
89+
90+
91+
92+
93+
94+

```

**Deploy to Kubernetes**

**Script**

```

        withKubeConfig(credentialsId: "k8s", namespace: "netflix", credentialsId: "k8s", namespace: "netflix", restrictKubeConfigAccess: false, serverId: "1") {
            sh "kubectl apply -f ${file}"
            sh "kubectl apply -f ${file}"
            sh "kubectl get svc"
        }
    }
}

stage("Deploy to Kubernetes") {
    steps {
        script {
            withKubeConfig(credentialsId: "k8s", clusterName: "", contextName: "", credentialsId: "k8s", namespace: "", restrictKubeConfigAccess: false, serverId: "1") {
                sh "kubectl apply -f ${file}"
                sh "kubectl apply -f ${file}"
                sh "kubectl get svc"
            }
        }
    }
}

post {
    always {
        mail(text: attaching: from,
            subject: "[$currentBuild.result]",
            body: "Project: [${env.BRANCH_NAME}]",
            to: ["${env.MAIL_TO}"],
            "bcc": ["${env.MAIL_BCC}"],
            attachmentsPattern: "trivy*.txt,trivyimage.txt")
    }
}

```

**SAVE** **Apply**

## Click on Build Now

You will see that our Application has been deployed successfully on Kubernetes.

dashboard > Netflix >

**Status** **Netflix**

- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- SonarQube
- Browse
- Pipeline Syntax
- Github Work Log
- Git Pulling Log

**Stage View**

	Declarative: Tool Install	Workspace Cleaning	Checkout from Git	Sonarcube Analysis	Quality Gate	Install Dependencies	OWASP DP Scan	TRIVY FS Scan	Docker Image Build	Docker Image Pushing	TRIVY Image Scan	Deploy to Container	Deploy to Kubernetes	Declarative: Post Actions
Average stage times: Manager (all build) -> 500 ms	502 ms	21ms	7s	21s	20ms	10s	3min 42s	79s	2min 41s	7s	2s	7s	2s	7s
Build History	100ms	20ms	87ms	27s	25ms (passive)	5s	3min 29s	79s	3min 40s	7s	2s	7s	2s	7s
Build History	100ms	20ms	99ms	23s	300ms (passive)	5s	3min 39s	79s	3min 39s	7s	2s	7s	2s	300ms
Build History	140ms	20ms	7s	22s	450ms (passive)	5s	3min 25s	79s	3min 44s	7s	2s	7s	2s	300ms
Build History	27ms	20ms	7s	23s	170ms (passive)	10s	3min 55s	79s	3min 43s	7s	2s	7s	2s	300ms

**SonarQube Quality Gate**

NetFlix **Passed** Sonar code processing **Success**

Latest Dependency-Check

Permalinks

You can validate whether your pods are running or not from your Kubernetes master node.

```
ubuntu@K8s-Master: ~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
netflix-app-57d865b6b5-8mk48  1/1     Running   0          3m8s
netflix-app-57d865b6b5-jrxld  1/1     Running   0          3m18s
ubuntu@K8s-Master: ~$ kubectl describe pod netflix-app-57d865b6b5-8mk48
Name:           netflix-app-57d865b6b5-8mk48
Namespace:      default
Priority:      0
Service Account: default
Node:          k8s-worker/172.31.52.140
Start Time:    Thu, 28 Dec 2023 11:20:42 +0000
Labels:         app=netflix-app
Annotations:   cni.projectcalico.org/containerID: a9672b032d00f378b10de0d6f21639896de39517074775c2b14aab5708e870c9
               cni.projectcalico.org/podIP: 192.168.254.140/32
               cni.projectcalico.org/podIPs: 192.168.254.140/32
Status:        Running
IP:            192.168.254.140
IPs:
IP-:          192.168.254.140
Controlled By: ReplicaSet/netflix-app-57d865b6b5
Containers:
  netflix-app:
    Container ID:  containerd://9dd2a1d5e50e3f95f9979ee78b0ff3fab2371201a82db6c17f8ac6af3af91a7f
    Image:         avian19/netflix:latest
    Image ID:     docker.io/avian19/netflix@sha256:5b099d44ba5c498bded22efbf365884b8f22ae0f0501f63e61bf35d929829e5
    Port:          80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:   Thu, 28 Dec 2023 11:20:43 +0000
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fzh29 (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-fzh29:
    Type:      projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:   true
    HostPath:     <host>
    SecretName:   kube-api-access-fzh29
```

Also, you can check the Console logs for the earlier results.

```
Dashboard > Netflix > #30
deployment.apps/netflix-app configured
[Pipeline] sh
+ kubectl apply -f service.yaml
service/netflix-app unchanged
[Pipeline] sh
+ kubectl get svc
NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP  10.96.0.1    <none>        443/TCP   5h1m
netflix-app  NodePort   10.109.178.85 <none>        80:32000/TCP 24m
[Pipeline] sh
+ kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/netflix-app-57d865b6b5-jrxld  0/1     ContainerCreating   0          1s
pod/netflix-app-96848cc9d-qeqwx  1/1     Running   0          11m
pod/netflix-app-96848cc9d-qeqyj  1/1     Running   0          11m

NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP  10.96.0.1    <none>        443/TCP   5h1m
service/netflix-app  NodePort   10.109.178.85 <none>        80:32000/TCP 24m

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/netflix-app  2/2     2           2           24m

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/netflix-app-57d865b6b5  1        1        0        1s
replicaset.apps/netflix-app-74f4458c59  0        0        0        24m
replicaset.apps/netflix-app-96848cc9d  2        2        2        11m
[Pipeline]
[kubernetes-client] kubectl configuration cleaned up
[Pipeline] // withKubeConfig
[Pipeline]
[Pipeline] // dir
[Pipeline]
[Pipeline] // script
[Pipeline]
[Pipeline] // WithEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
```

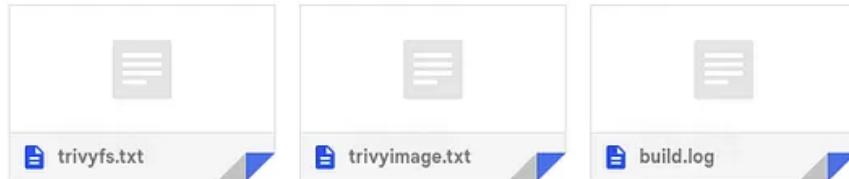
We got the email that our pipeline was successful.

'SUCCESS' □ Inbox

**address not configured yet** <aman07pathak@gmail.com>  
to me ▾

Project: Netflix  
Build Number: 31  
URL: <http://34.207.155.151:8080/job/Netflix/31/>

### 3 Attachments • Scanned by Gmail ⓘ



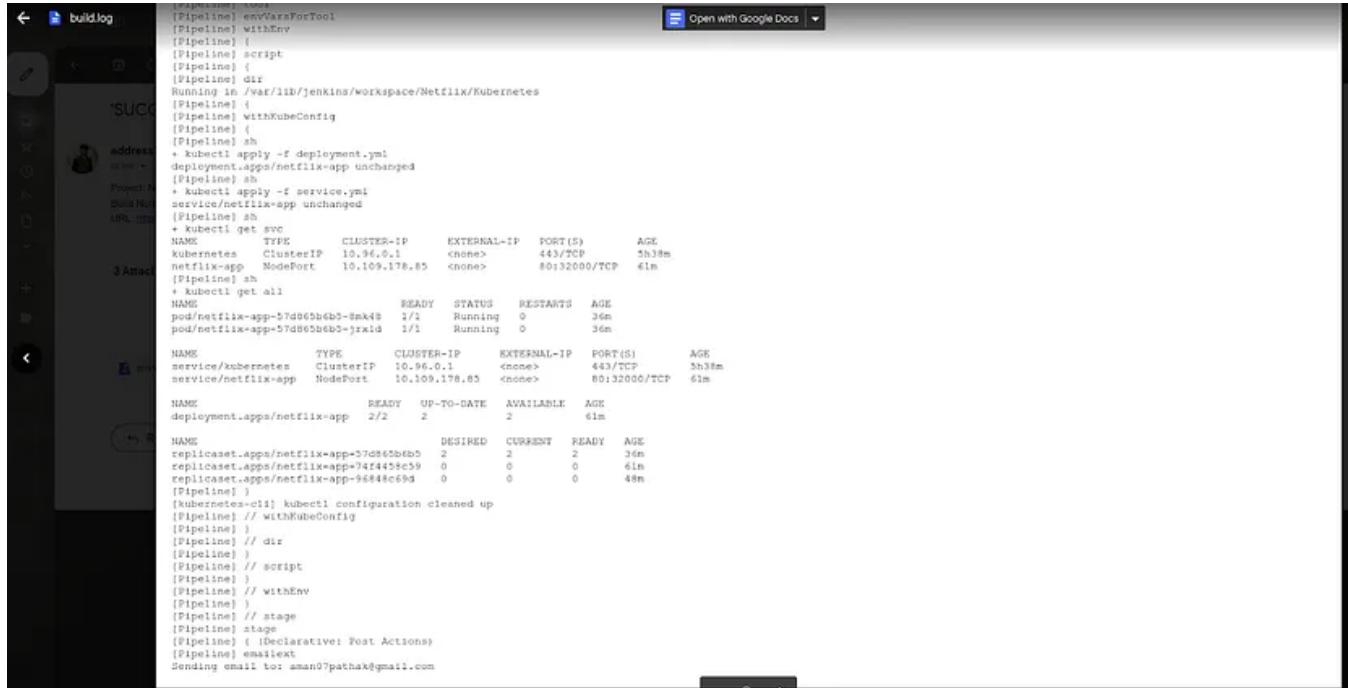
We get the `trivyfs.txt` file which contains the vulnerabilities.

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
postcss	CVE-2023-44270	MEDIUM	fixed	8.4.18	8.4.31	An issue was discovered in PostCSS before 8.4.31. The vulnerability affects ... <a href="https://avd.aquasec.com/nvd/cve-2023-44270">https://avd.aquasec.com/nvd/cve-2023-44270</a>
vite	CVE-2023-34092	HIGH		3.2.2	2.9.16, 3.2.7, 4.0.3, 4.1.5, 4.2.3, 4.3.9	Vite Server Options (server.ts.deny) can be bypassed using double forward-slash (/ /) <a href="https://avd.aquasec.com/nvd/cve-2023-34092">https://avd.aquasec.com/nvd/cve-2023-34092</a>

Also, we got the vulnerabilities for our Docker Image.

avlan19/netflix:latest (alpine 3.17.6)
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)

Jenkins sent the console logs by email.



```

[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/Netflix/Kubernetes
[Pipeline] {
[Pipeline] withKubeConfig
[Pipeline] {
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f deployment.yaml
deployment.apps/netflix-app unchanged
[Pipeline] sh
+ kubectl apply -f service.yaml
service/netflix-app unchanged
[Pipeline] sh
+ kubectl get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)      AGE
kubernetes    ClusterIP  10.96.0.1   <none>       443/TCP     5h38m
netflix-app   NodePort   10.109.178.83  <none>       80:32000/TCP  61m
[Pipeline] sh
+ kubectl get all
NAME           READY   STATUS    RESTARTS   AGE
pod/netflix-app-57d865b6b5-8mk4b  1/1    Running   0          36m
pod/netflix-app-57d865b6b5-jxwid  1/1    Running   0          36m
[Pipeline] sh
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)      AGE
service/kubernetes  ClusterIP  10.96.0.1   <none>       443/TCP     5h38m
service/netflix-app  NodePort   10.109.178.83  <none>       80:32000/TCP  61m
[Pipeline] sh
NAME          READY   UP-TO-DATE  AVAILABLE   AGE
deployment.apps/netflix-app  2/2      2           2           61m
[Pipeline] sh
NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/netflix-app-57d865b6b5  2       2       2       36m
replicaset.apps/netflix-app-7474455c59  0       0       0       61m
replicaset.apps/netflix-app-98848c693  0       0       0       48m
[Pipeline] sh
[kubernetes-csi] kubecfg configuration cleaned up
[Pipeline] // withKubeConfig
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { [Declarative: Post Actions]
[Pipeline] emailExt
Sending email to: aman07pathak@gmail.com

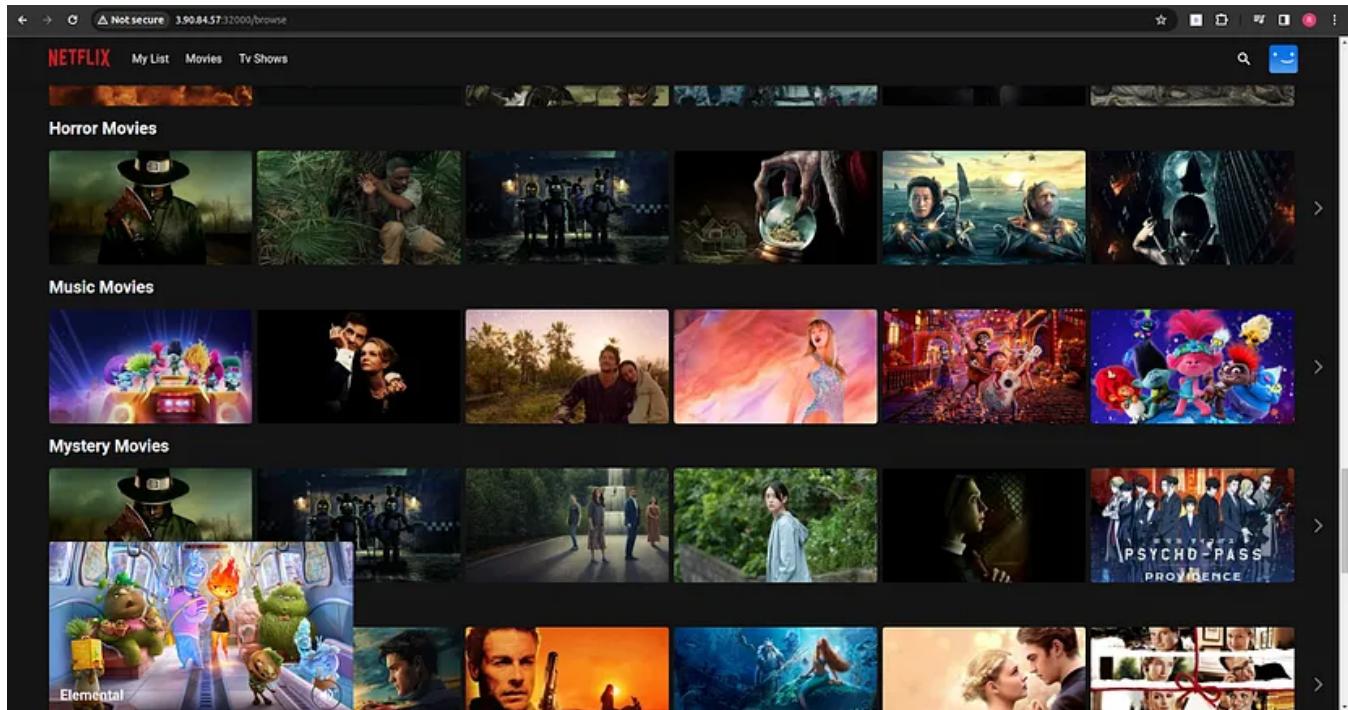
```

If you want to access your Netflix Clone Application.

Copy the Public IP of Worker Node and paste it on your favorite browser with port 32000 and see the magic.

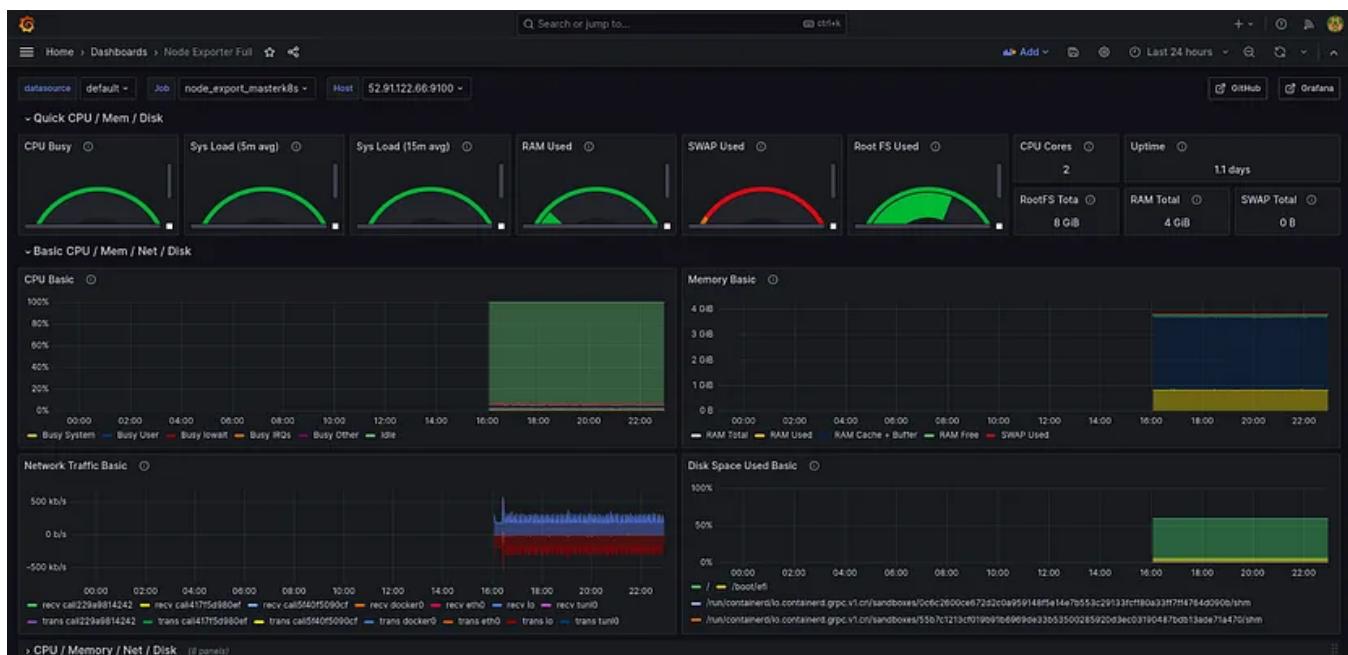


Another Snippet of our Netflix Clone application.



Go to the Grafana Dashboard and select Node Exporter.

You will see the real-time hardware specs of your Kubernetes master node.



You will see the real-time hardware specs of your Kubernetes worker node.