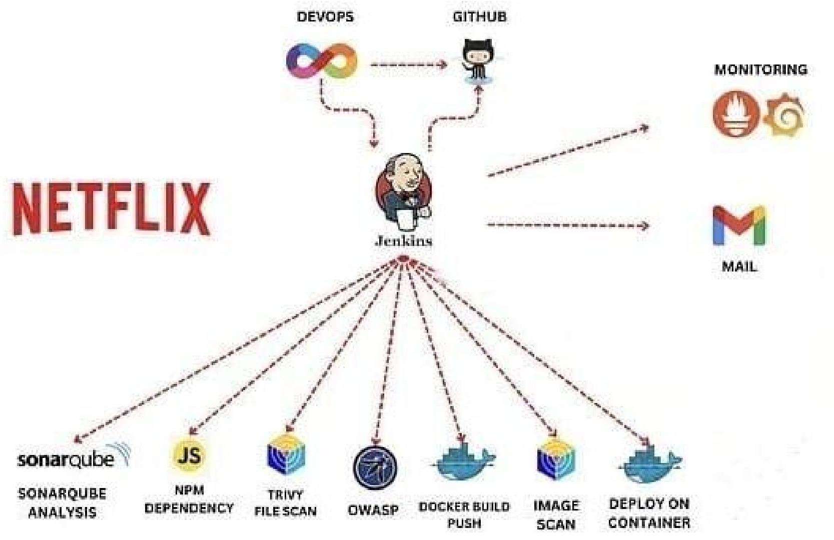
Netflix Clone CI-CD



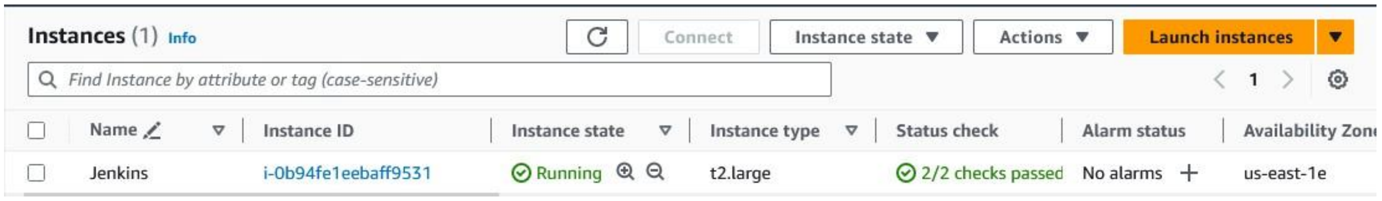
Project Overview

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

Project Steps

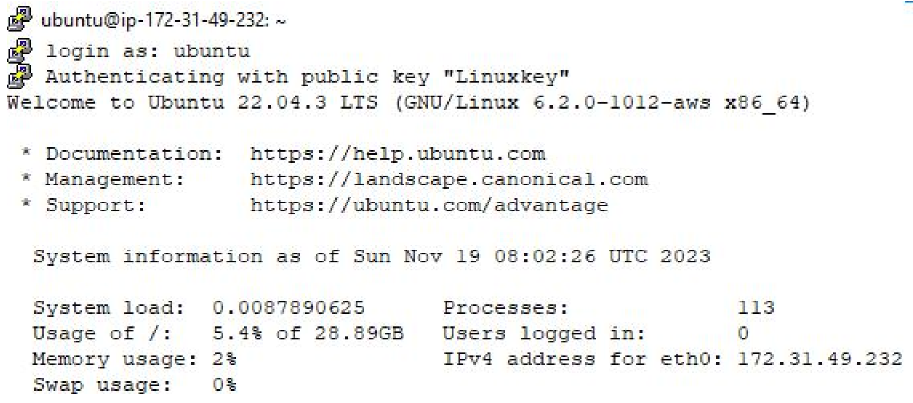
- Step 1 — Launch an Ubuntu(22.04) T2 Large Instance
- Step 2 — Install Jenkins, Docker and Trivy. Create a Sonarqube Container using Docker.
- Step 3 — Create a TMDB API Key.
- Step 4 — Install Prometheus and Grafana On the new Server.
- Step 5 — Install the Prometheus Plugin and Integrate it with the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

Step 1 : Launch ubuntu instance t2.large



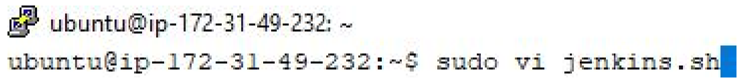
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Jenkins	i-0b94fe1eebaff9531	Running	t2.large	2/2 checks passed	No alarms	us-east-1e

Step 2 : Login to the Instance



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

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

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

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

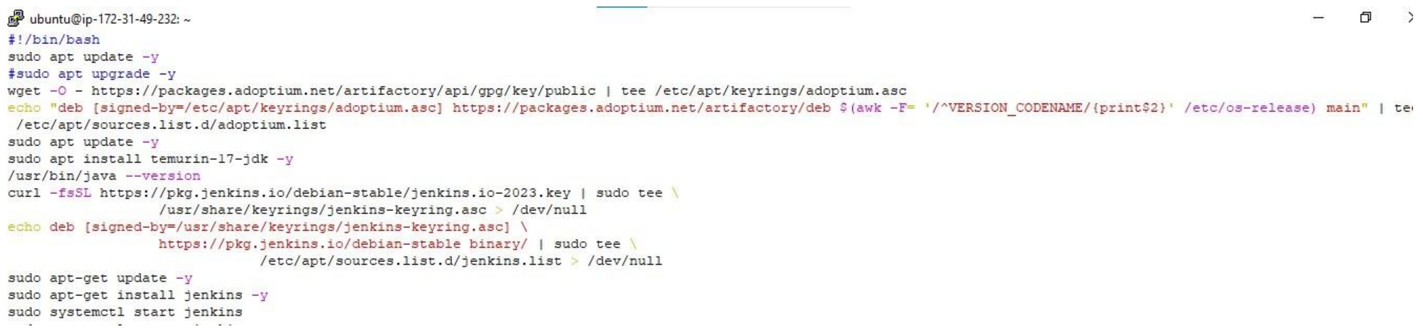
Step 3 : Create one shell script file, bcz of we need to install Jenkins on this server



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

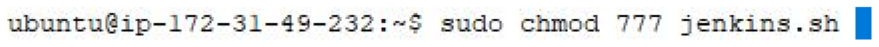
Step 4 : Add the Jenkins download steps in the script file

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



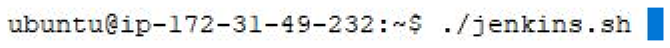
```
ubuntu@ip-172-31-49-232: ~
#!/bin/bash
sudo apt update -y
#sudo apt upgrade -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee /etc/apt/keyrings/adoptium.asc
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | te
 /etc/apt/sources.list.d/adoptium.list
sudo apt update -y
sudo apt install temurin-17-jdk -y
/usr/bin/java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
                  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
                  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
                              /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
```

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



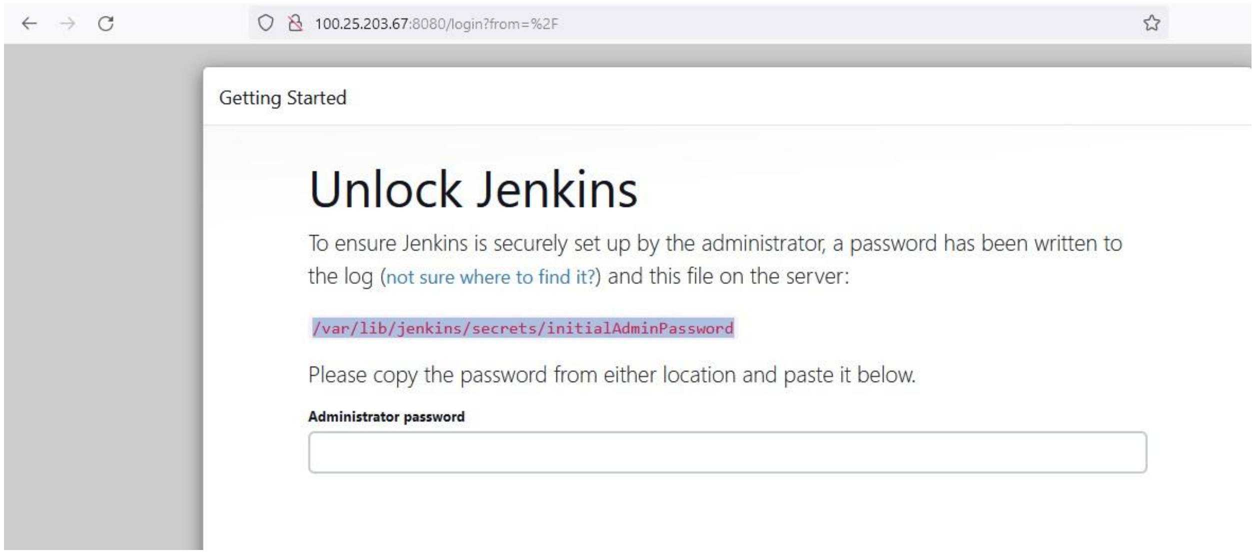
```
ubuntu@ip-172-31-49-232:~$ sudo chmod 777 jenkins.sh
```

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



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

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



100.25.203.67:8080/login?from=%2F

Getting Started

Unlock Jenkins

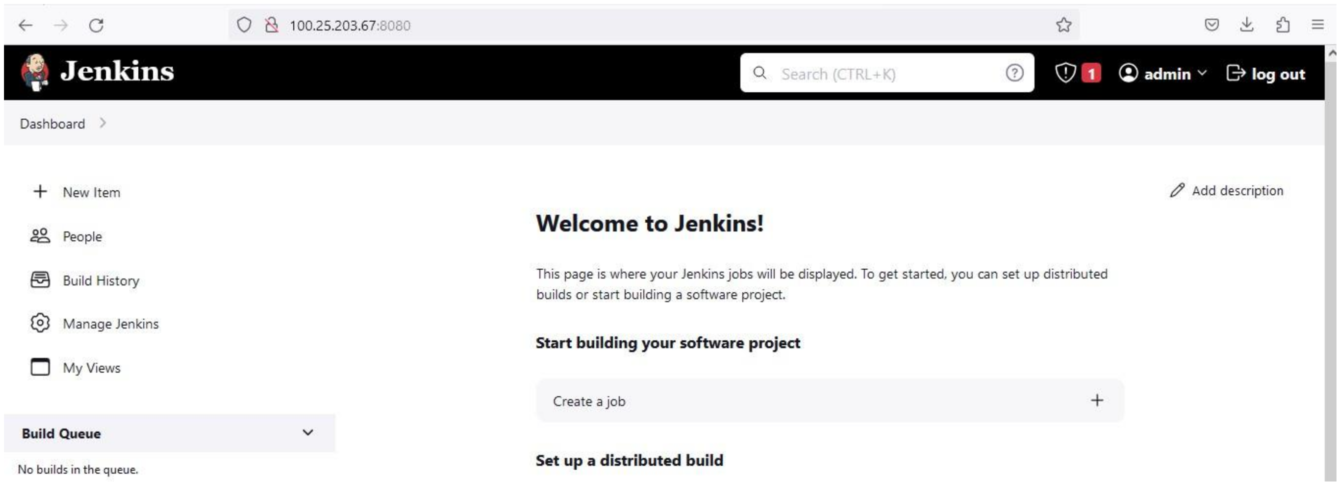
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Step 8 : Jenkins Installed successfully



100.25.203.67:8080

No builds in the queue.

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

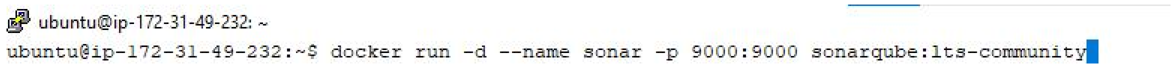
Create a job

Set up a distributed build

Step 9 : Then Install docker on the same server

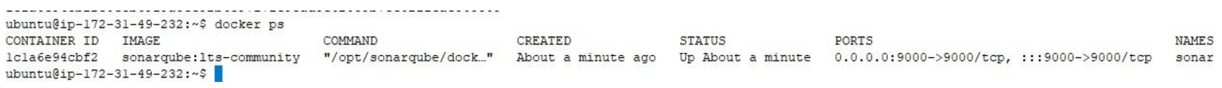
```
Commands

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

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



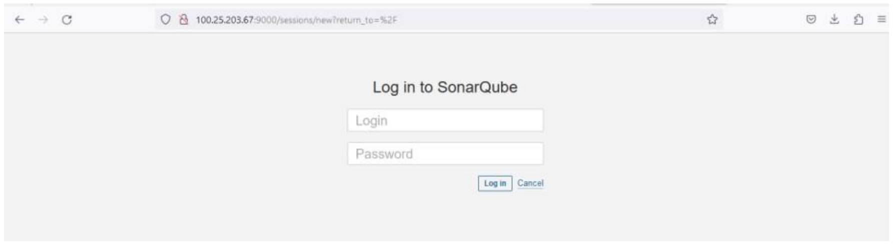
```
ubuntu@ip-172-31-49-232: ~
ubuntu@ip-172-31-49-232:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

Step 11 : Sonar Container created successfully



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

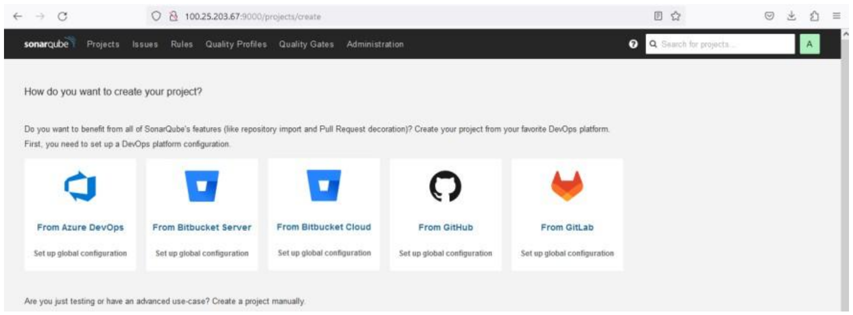
Step 12 : Copy the server ip with port no 9000 u get the sonar page (username and pwd= admin)



100.25.203.67:9000/sessions/new?return_to=%2F

Log in to SonarQube

Step 13 : Sonar created successfully



100.25.203.67:9000/projects/create

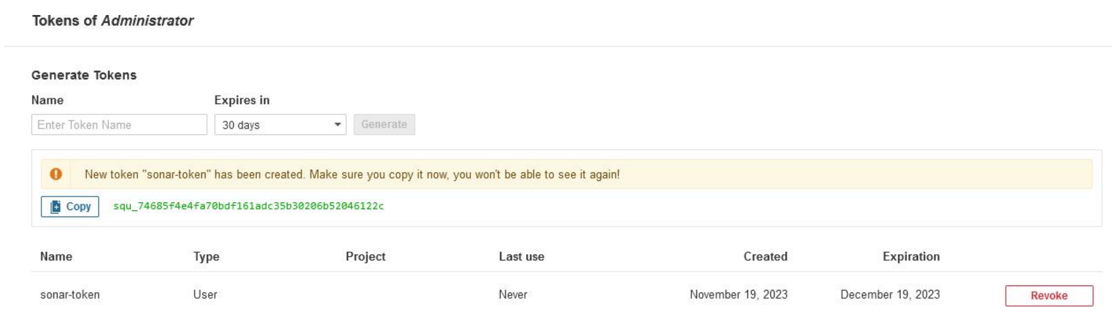
How do you want to create your project?

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.

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



Tokens of *Administrator*

Generate Tokens

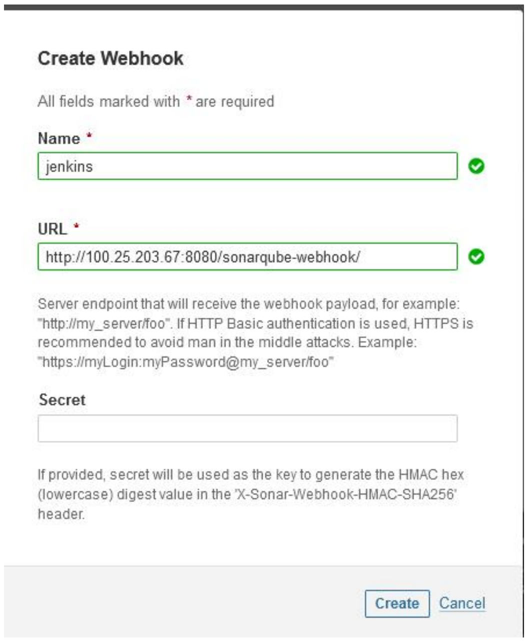
Name	Expires in	
Enter Token Name	30 days	Generate

New token "sonar-token" has been created. Make sure you copy it now, you won't be able to see it again!

Copy `squ_74685f4e4fa70bdf161adc35b30206b52046122c`

Name	Type	Project	Last use	Created	Expiration	
sonar-token	User		Never	November 19, 2023	December 19, 2023	Revoke

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



Create Webhook

All fields marked with * are required

Name *: jenkins

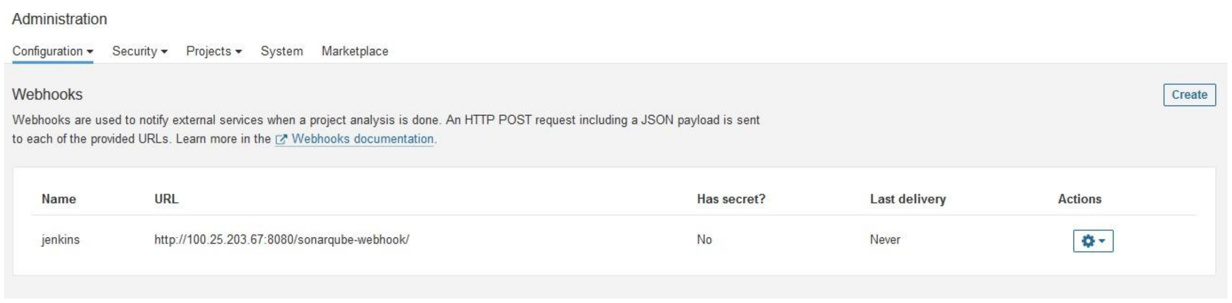
URL *: http://100.25.203.67:8080/sonarqube-webhook/

Server endpoint that will receive the webhook payload, for example: "http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example: "https://myLogin:myPassword@my_server/foo"

Secret:

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

Create Cancel



Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

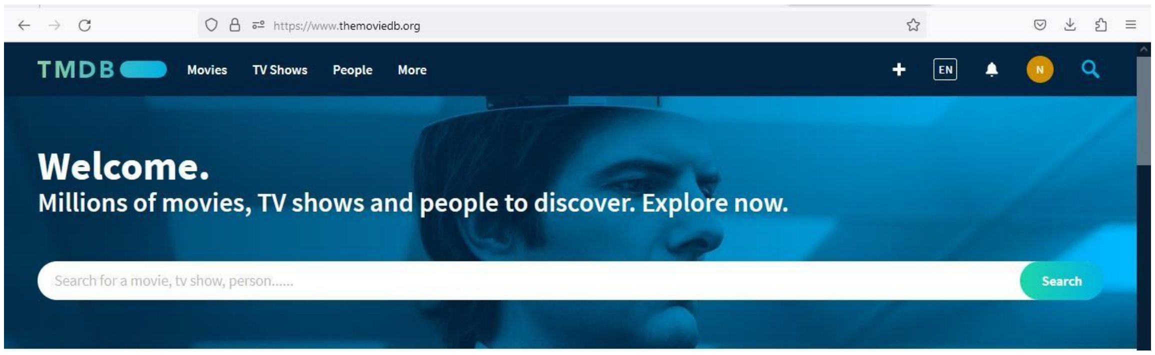
Webhooks

Create

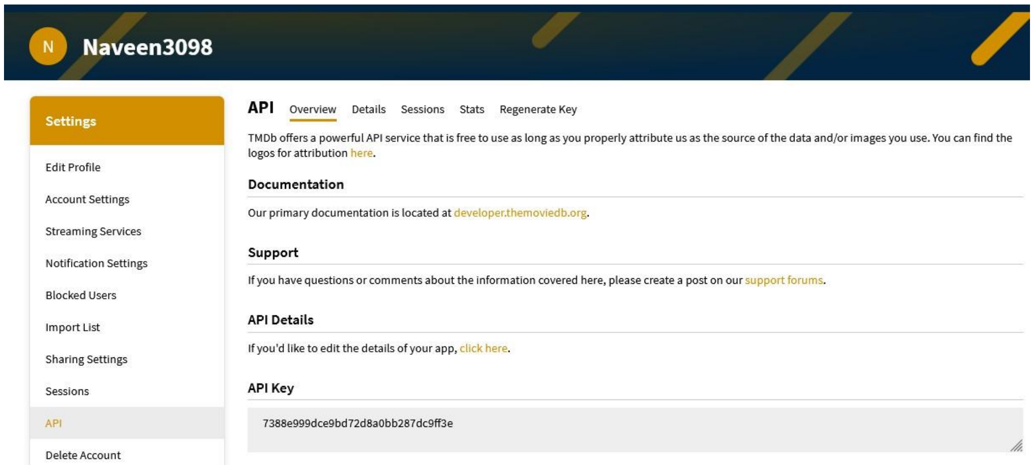
Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation.

Name	URL	Has secret?	Last delivery	Actions
jenkins	http://100.25.203.67:8080/sonarqube-webhook/	No	Never	⚙ ▾

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


https://www.themoviedb.org
TMDB
Movies
TV Shows
People
More
EN
N
Welcome.
Millions of movies, TV shows and people to discover. Explore now.
Search for a movie, tv show, person......
Search

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



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.

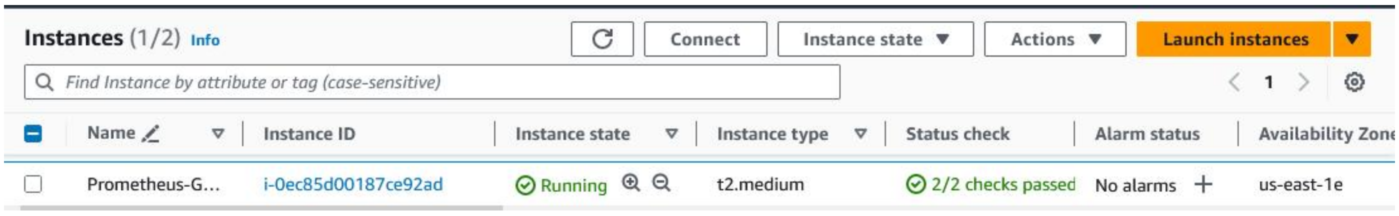
API Details

If you'd like to edit the details of your app, click here.

API Key

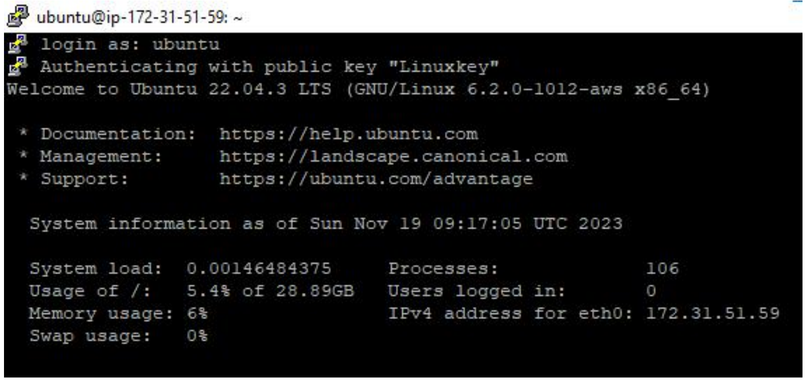
```
7388e999dce9bd72d8a0bb287dc9ff3e
```

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



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Prometheus-G...	i-0ec85d00187ce92ad	Running	t2.medium	2/2 checks passed	No alarms	us-east-1e

Step 19 : Instance created successfully



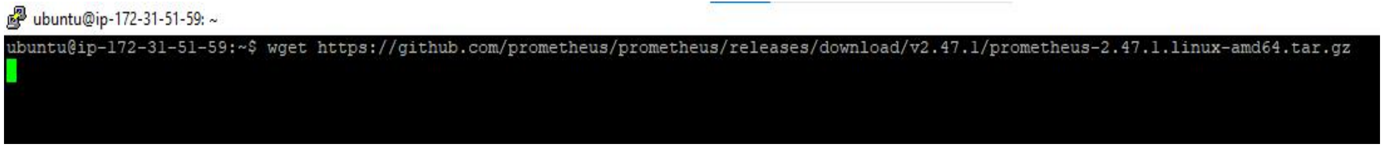
```
ubuntu@ip-172-31-51-59: ~
login as: ubuntu
Authenticating with public key "Linuxkey"
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

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

  System information as of Sun Nov 19 09:17:05 UTC 2023

  System load:  0.00146484375     Processes:             106
  Usage of /:   5.4% of 28.89GB   Users logged in:       0
  Memory usage: 6%                IPv4 address for eth0: 172.31.51.59
  Swap usage:   0%
```

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

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

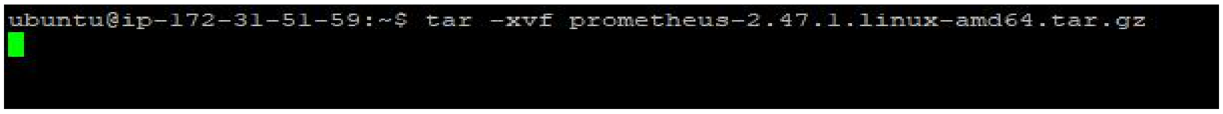
Step 21 : Download Prometheus :

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



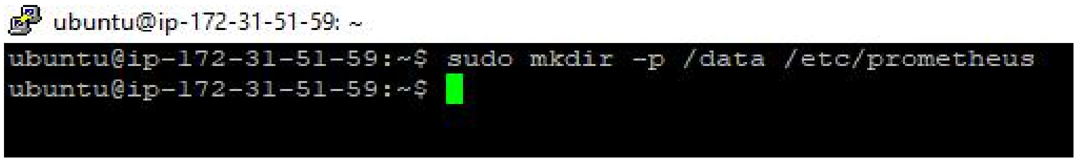
```
ubuntu@ip-172-31-51-59: ~
ubuntu@ip-172-31-51-59:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.47.1/prometheus-2.47.1.linux-amd64.tar.gz
```

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



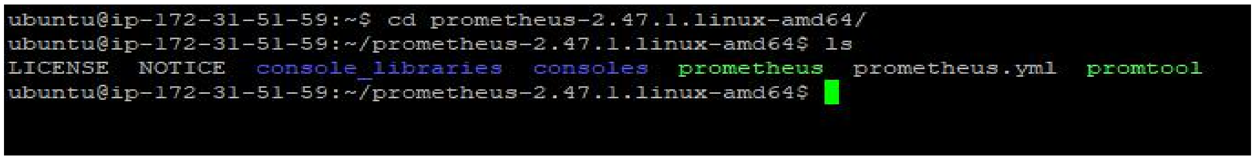
```
ubuntu@ip-172-31-51-59:~$ tar -xvf prometheus-2.47.1.linux-amd64.tar.gz
```

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



```
ubuntu@ip-172-31-51-59: ~
ubuntu@ip-172-31-51-59:~$ sudo mkdir -p /data /etc/prometheus
ubuntu@ip-172-31-51-59:~$
```

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



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

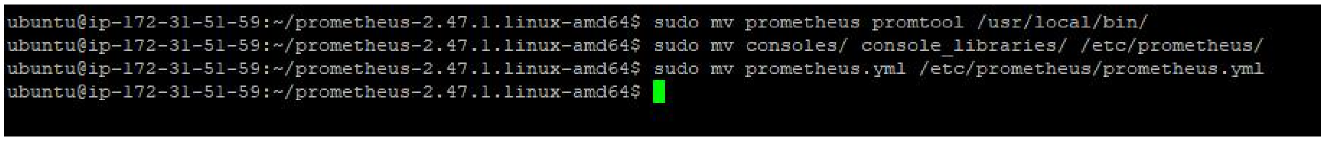
Step 25 : Move the Prometheus & promtool file to /usr/local/bin

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

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

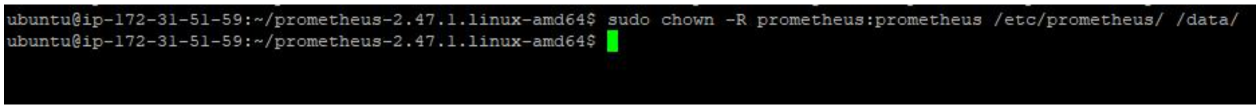
sudo mv consoles/ console_libraries/ /etc/prometheus/

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



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

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

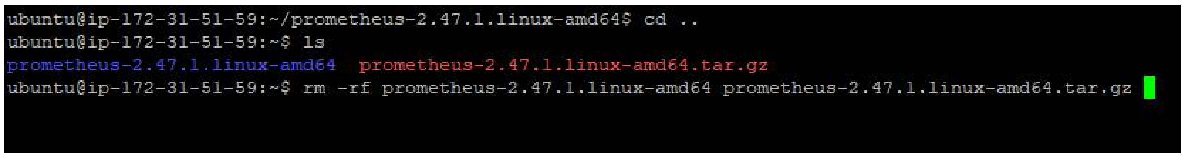


```
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ sudo chown -R prometheus:prometheus /etc/prometheus/ /data/
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$
```

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

(cd ..

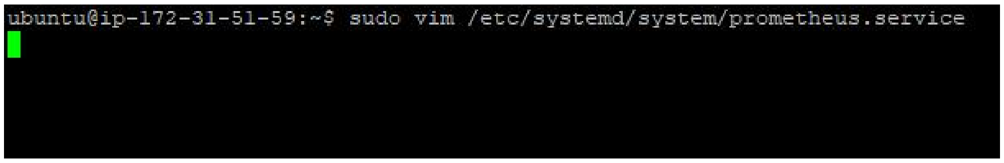
rm -rf prometheus-2.47.1.linux-amd64.tar.gz)



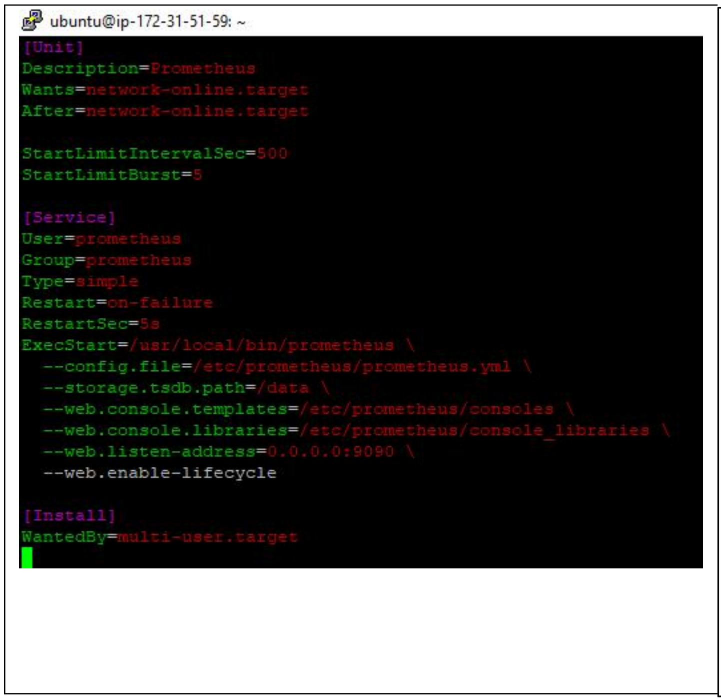
```
ubuntu@ip-172-31-51-59:~/prometheus-2.47.1.linux-amd64$ cd ..
ubuntu@ip-172-31-51-59:~$ ls
prometheus-2.47.1.linux-amd64  prometheus-2.47.1.linux-amd64.tar.gz
ubuntu@ip-172-31-51-59:~$ rm -rf prometheus-2.47.1.linux-amd64 prometheus-2.47.1.linux-amd64.tar.gz
```

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

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



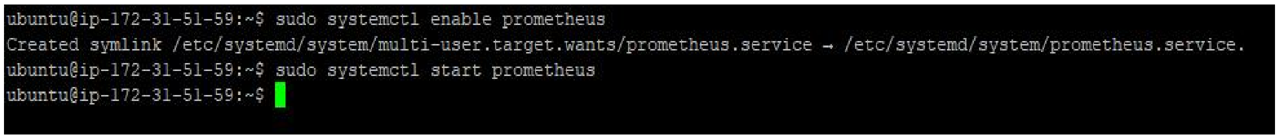
```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/systemd/system/prometheus.service
```



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

StartLimitIntervalSec=500
StartLimitBurst=5

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

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

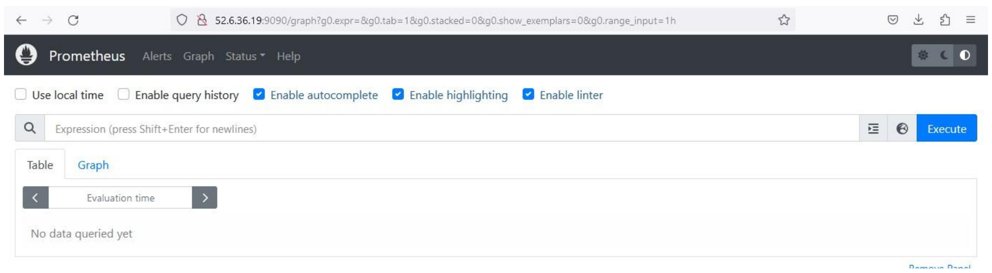
```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

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


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

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



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

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

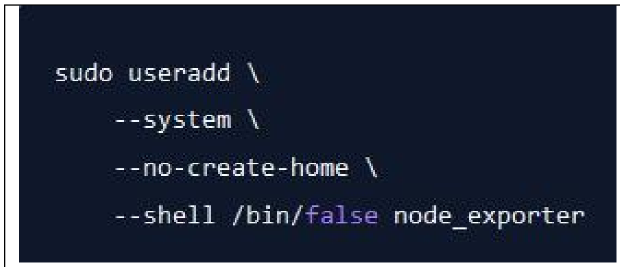


52.6.36.19:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h

☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

No data queried yet

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

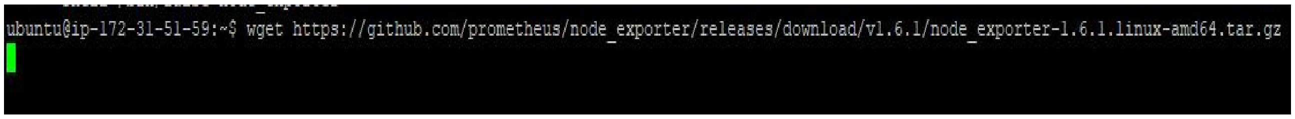


```
sudo useradd \
    --system \
    --no-create-home \
    --shell /bin/false node_exporter
```

Code

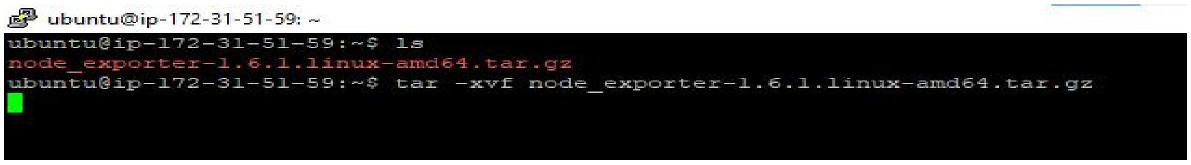
```
sudo useradd \
  --system \
  --no-create-home \
  --shell /bin/false node_exporter
```

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



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

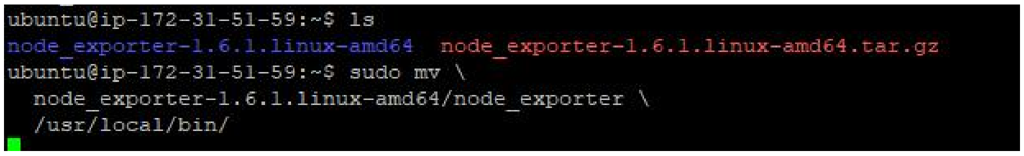
Step 34 : Untar the node exporter file (tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz)



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

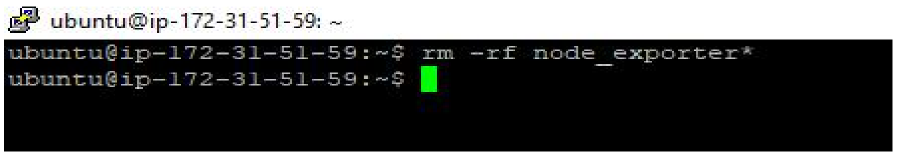
Step 35 : Move the node exporter file

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



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

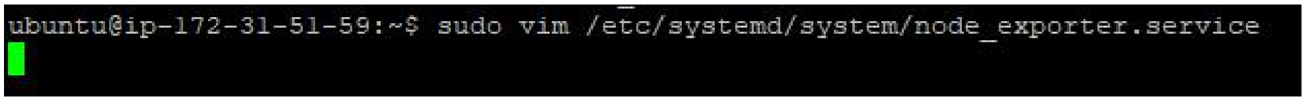
Step 36 : After moving the node exporter file remove the tar file (rm -rf node_exporter*)



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

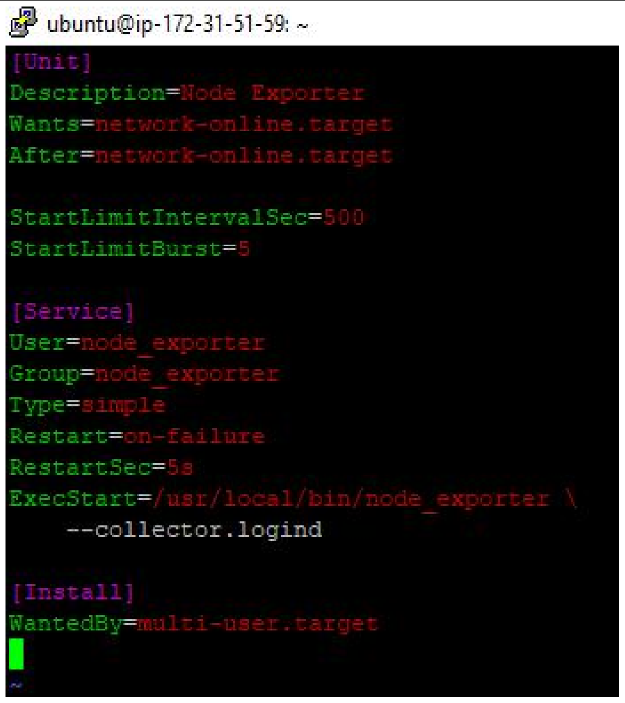
Step 37 : We're going to use Systemd, which is a system and service manager for Linux operating systems. For that, we need to create a Systemd unit configuration file.

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



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

Step 38 : Add script in the file

 ubuntu@ip-172-31-51-59: ~	[Unit]
	Description=Node Exporter
	Wants=network-online.target
	After=network-online.target
	
	StartLimitIntervalSec=500
	StartLimitBurst=5
	
	[Service]
	User=node_exporter
	Group=node_exporter
	Type=simple
	Restart=on-failure
	RestartSec=5s
	ExecStart=/usr/local/bin/node_exporter \\
	--collector.logind
	[Install]
	WantedBy=multi-user.target

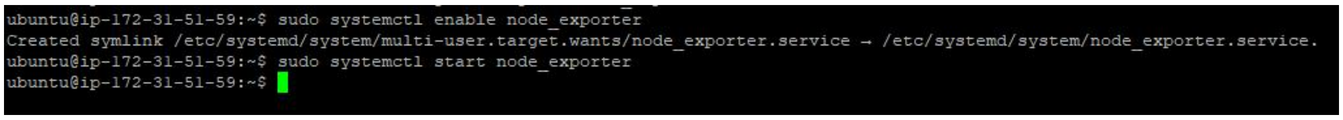
```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

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

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

Step 39 : Start the node exporter

service (sudo systemctl enable

node_exporter sudo systemctl start

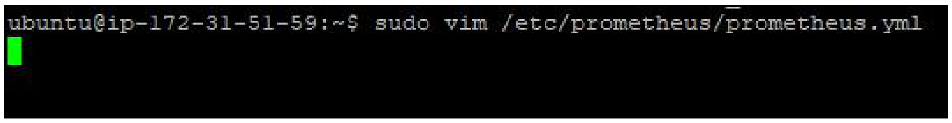
node_exporter)



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

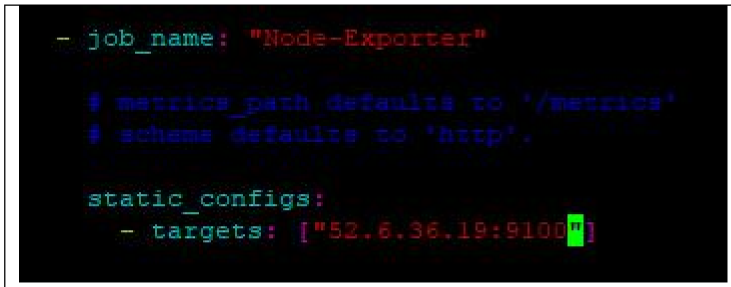
Step 40 : Add the node exporter job in Prometheus.yml

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



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

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



```
  - job_name: "Node-Exporter"

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

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

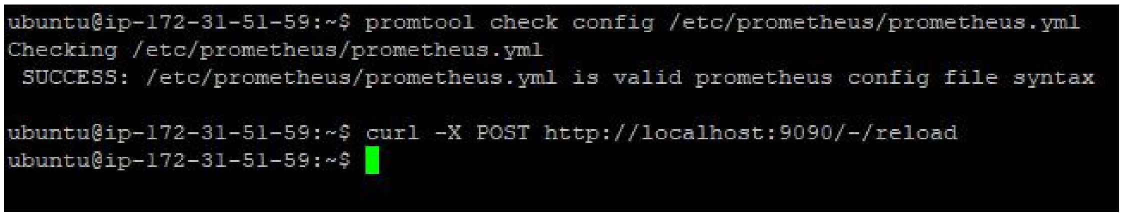
```
- job_name: node_export

  static_configs:

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

Step 42 : Start and reload the service

(promtool check config

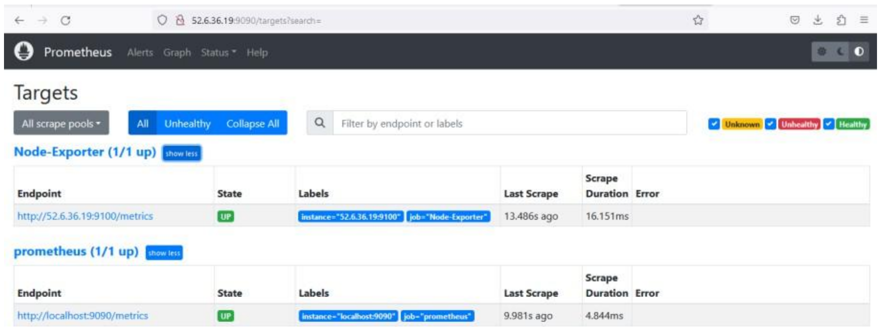
/etc/prometheus/prometheus.yml) (curl -X POST

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



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

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

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



Targets

Node-Exporter (1/1 up)

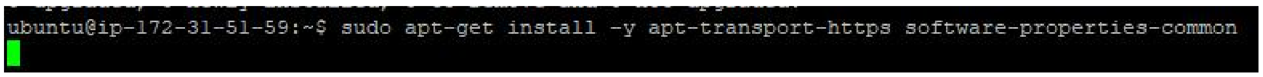
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://52.6.36.19:9100/metrics	UP	instance="52.6.36.19:9100" job="Node-Exporter"	13.486s ago	16.151ms	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	9.981s ago	4.844ms	

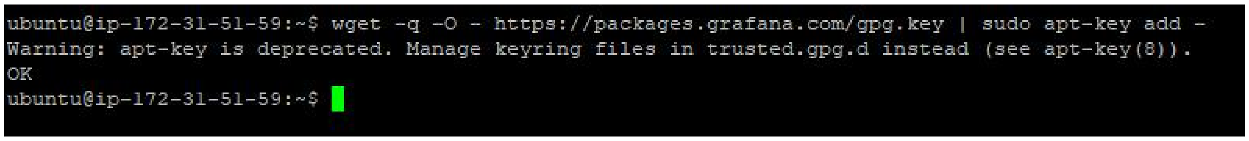
Step 44 : Download grafana

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



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

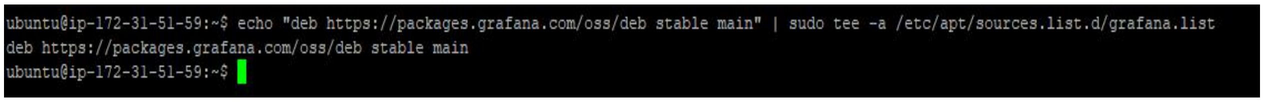
Step 45 : Download gpg key (wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -)



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

Step 46 : Add this repository for stable releases.

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



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

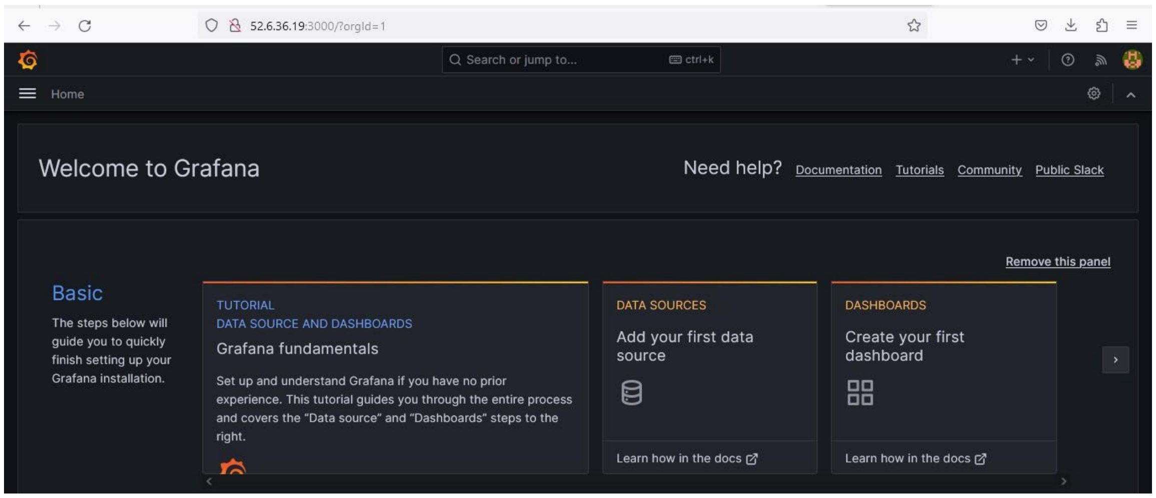
Step 47 : Download Grafana by the below command

```
sudo apt-get update
```

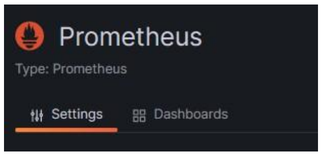
```
sudo apt-get -y install grafana
```

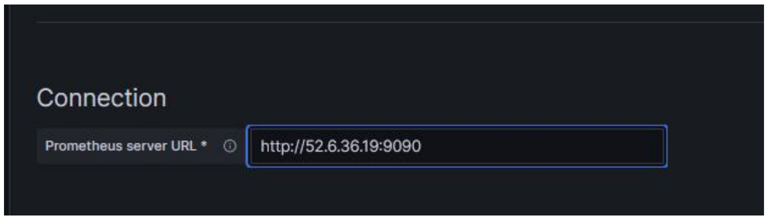
```
sudo systemctl start grafana-server
```

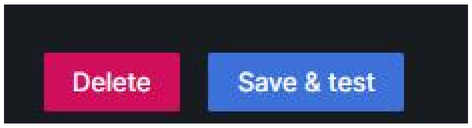
Step 48 : Copy Ip with port no.3000 and paste in chrome u get grafana page after u get the page click on data sources


52.6.36.19:3000/?orgId=1
Search or jump to...
Home
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
Remove this panel
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

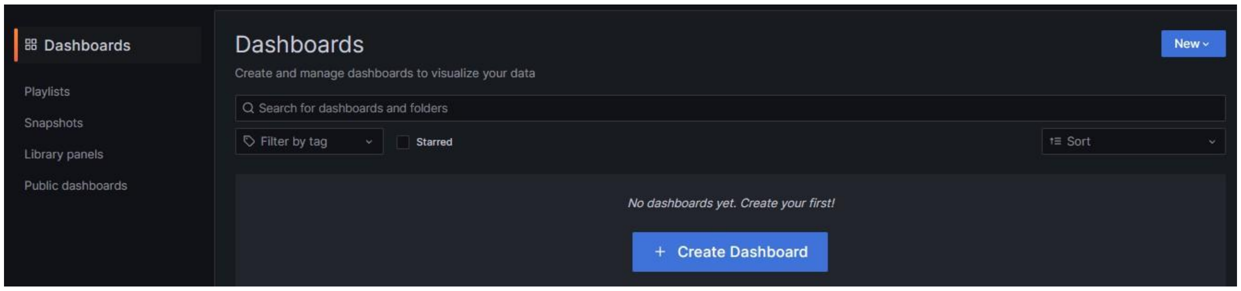
Step 49 : And Click on Prometheus add the Prometheus URL and click on save test



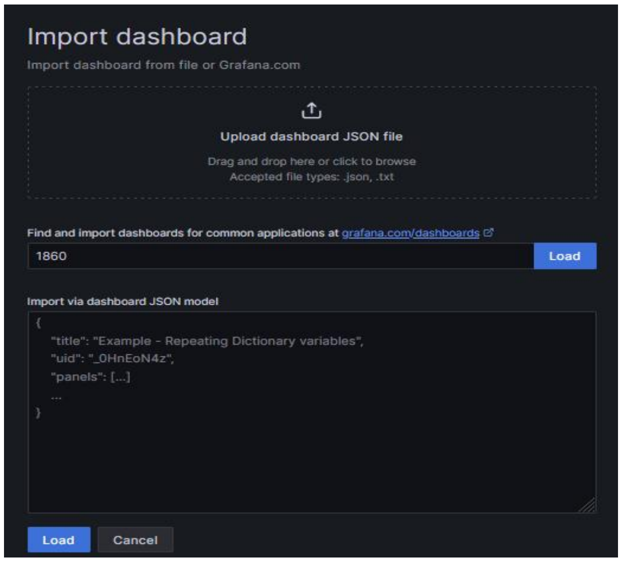




Step 50 : Click on new and click on Import dashboard


Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Dashboards
Create and manage dashboards to visualize your data
New
Search for dashboards and folders
Filter by tag
Starred
Sort
No dashboards yet. Create your first!
Create Dashboard

Step 51 : Enter no.1860 and click on load



Import dashboard

Import dashboard from file or Grafana.com

Upload dashboard JSON file

Drag and drop here or click to browse
Accepted file types: .json, .txt

Find and import dashboards for common applications at grafana.com/dashboards

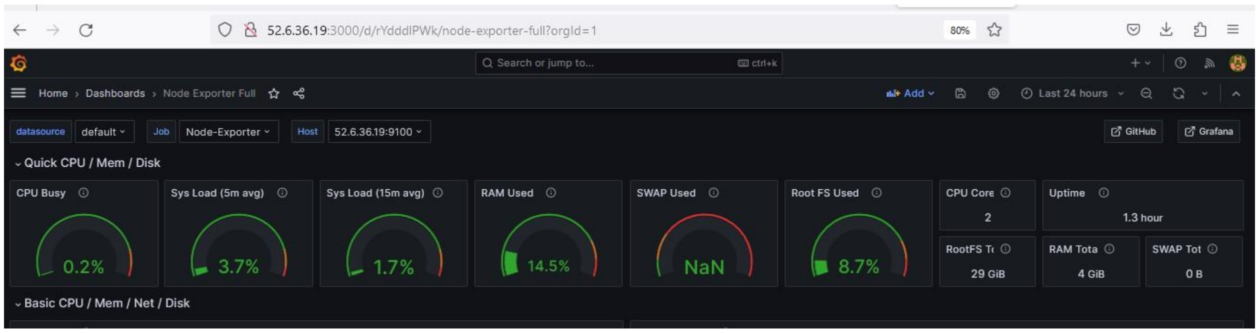
1860 | Load

Import via dashboard JSON model

```
{
    "title": "Example - Repeating Dictionary variables",
    "uid": "_0HnEoN4z",
    "panels": [...]
    ...
}
```

Load | Cancel

Step 52 : U get monitoring page of node exporter



52.6.36.19:3000/d/rYdddlPWk/node-exporter-full?orgId=1

Home › Dashboards › Node Exporter Full

datasource: default | Job: Node-Exporter | Host: 52.6.36.19:9100

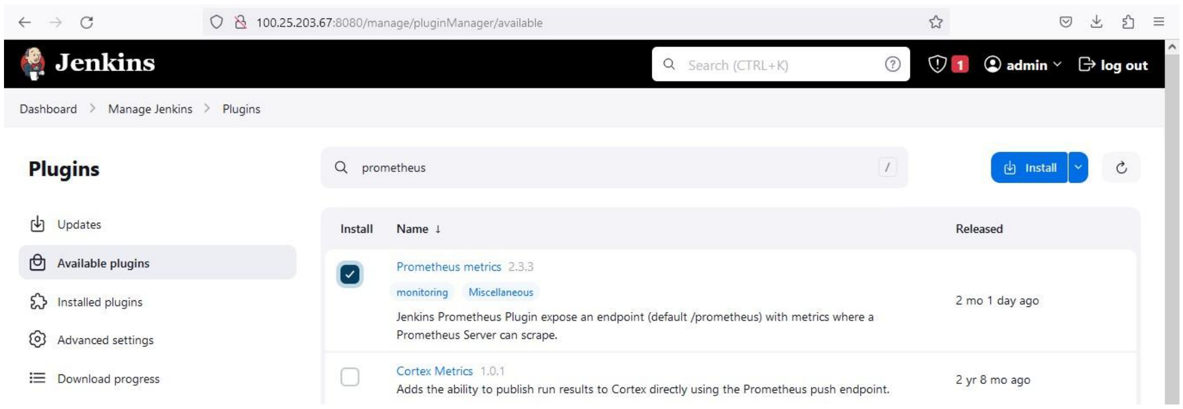
Quick CPU / Mem / Disk

CPU Busy	Sys Load (5m avg)	Sys Load (15m avg)	RAM Used	SWAP Used	Root FS Used	CPU Core	Uptime
0.2%	3.7%	1.7%	14.5%	NaN	8.7%	2	1.3 hour

RootFS Tt	RAM Tota	SWAP Tot
29 GiB	4 GiB	0 B

Basic CPU / Mem / Net / Disk

Step 53 : Go to Jenkins download Prometheus plugin



100.25.203.67:8080/manage/pluginManager/available

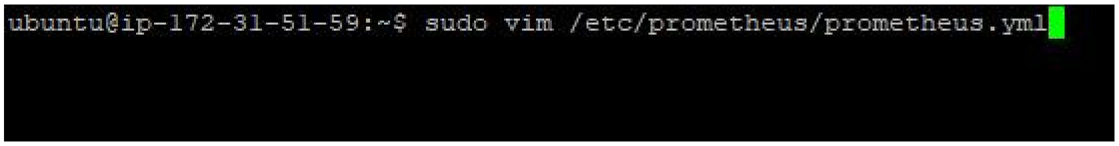
Dashboard › Manage Jenkins › Plugins

Plugins

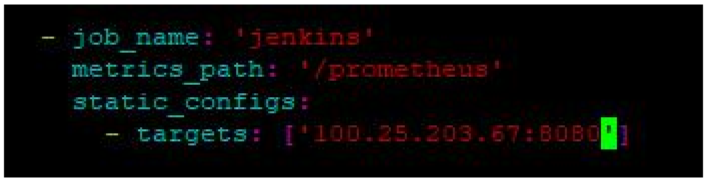
Search: prometheus

Install	Name ↓	Released
✓	Prometheus metrics 2.3.3 — monitoring, Miscellaneous — Jenkins Prometheus Plugin expose an endpoint (default /prometheus) with metrics where a Prometheus Server can scrape.	2 mo 1 day ago
	Cortex Metrics 1.0.1 — Adds the ability to publish run results to Cortex directly using the Prometheus push endpoint.	2 yr 8 mo ago

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

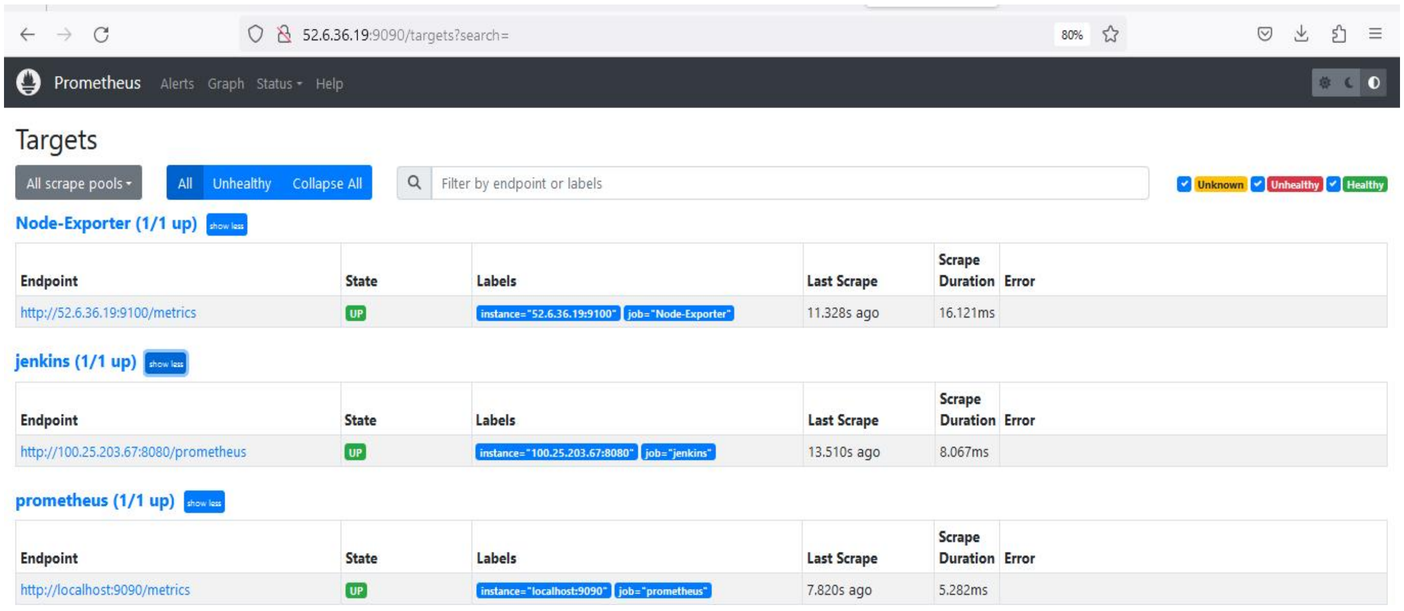

```
ubuntu@ip-172-31-51-59:~$ sudo vim /etc/prometheus/prometheus.yml
```

Step 55 : Add this Jenkins job in Prometheus.yml


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

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

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



Targets

Node-Exporter (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://52.6.36.19:9100/metrics	UP	instance="52.6.36.19:9100" job="Node-Exporter"	11.328s ago	16.121ms	

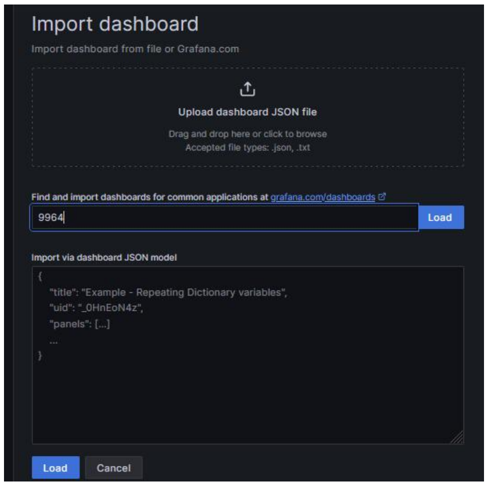
jenkins (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://100.25.203.67:8080/prometheus	UP	instance="100.25.203.67:8080" job="jenkins"	13.510s ago	8.067ms	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	7.820s ago	5.282ms	

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



Import dashboard

Import dashboard from file or Grafana.com

Upload dashboard JSON file

Drag and drop here or click to browse
Accepted file types: .json, .txt

Find and import dashboards for common applications at grafana.com/dashboards

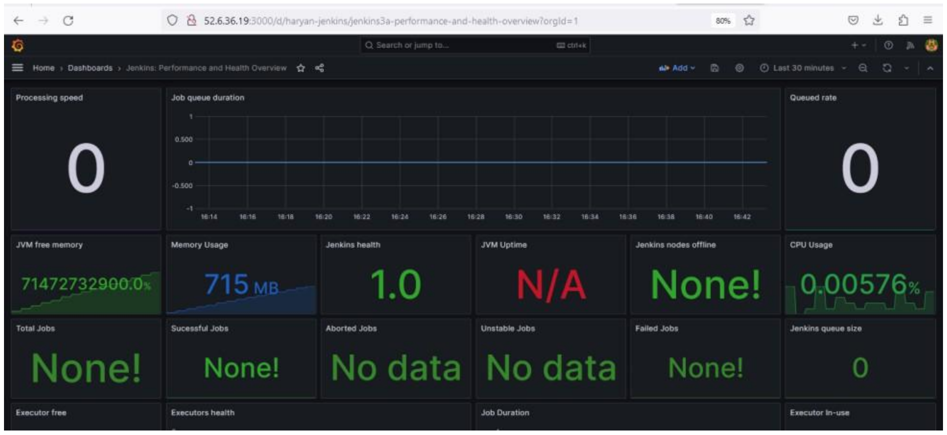
9964 Load

Import via dashboard JSON model

```
{
   "title": "Example - Repeating Dictionary variables",
   "uid": "_0HnEoN4z",
   "panels": [...]
   ...
}
```

Load Cancel

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



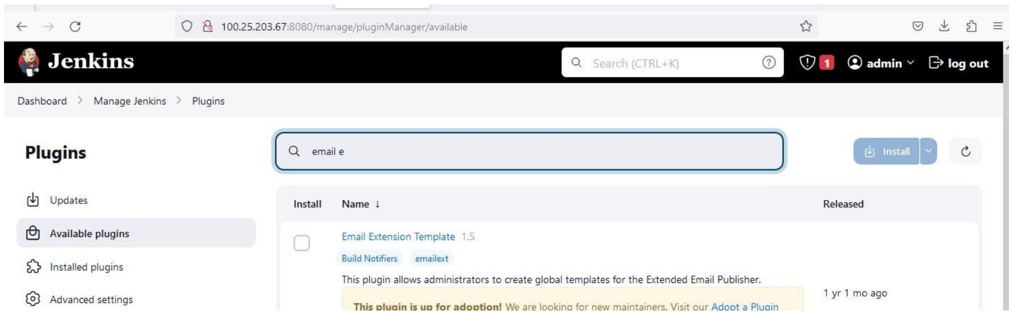
52.6.36.19:3000/d/haryan-jenkins/jenkins3a-performance-and-health-overview?orgId=1

Home › Dashboards › Jenkins: Performance and Health Overview

Last 30 minutes

Panel	Value
Processing speed	0
Job queue duration	
Queued rate	0
JVM free memory	71472732900.0%
Memory Usage	715 MB
Jenkins health	1.0
JVM Uptime	N/A
Jenkins nodes offline	None!
CPU Usage	0.00576%
Total Jobs	None!
Sucessful Jobs	None!
Aborted Jobs	No data
Unstable Jobs	No data
Failed Jobs	None!
Jenkins queue size	0
Executor free	
Executors health	
Job Duration	
Executor In-use	

Step 59 : Download the email plugin in jenkins



100.25.203.67:8080/manage/pluginManager/available

Jenkins   Search (CTRL+K)   admin   log out

Dashboard > Manage Jenkins > Plugins

Plugins

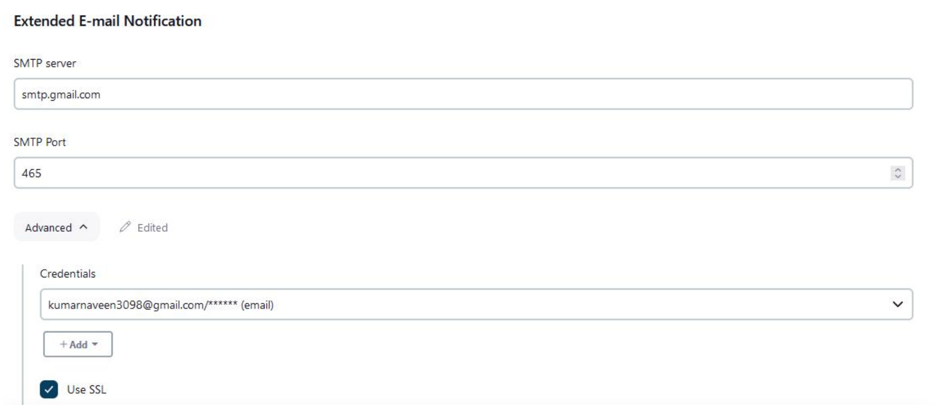
- Updates
- Available plugins
- Installed plugins
- Advanced settings

email e

Install

Install	Name ↓	Released
☐	Email Extension Template 1.5 Build Notifiers emailext This plugin allows administrators to create global templates for the Extended Email Publisher. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin	1 yr 1 mo ago

Step 60



Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

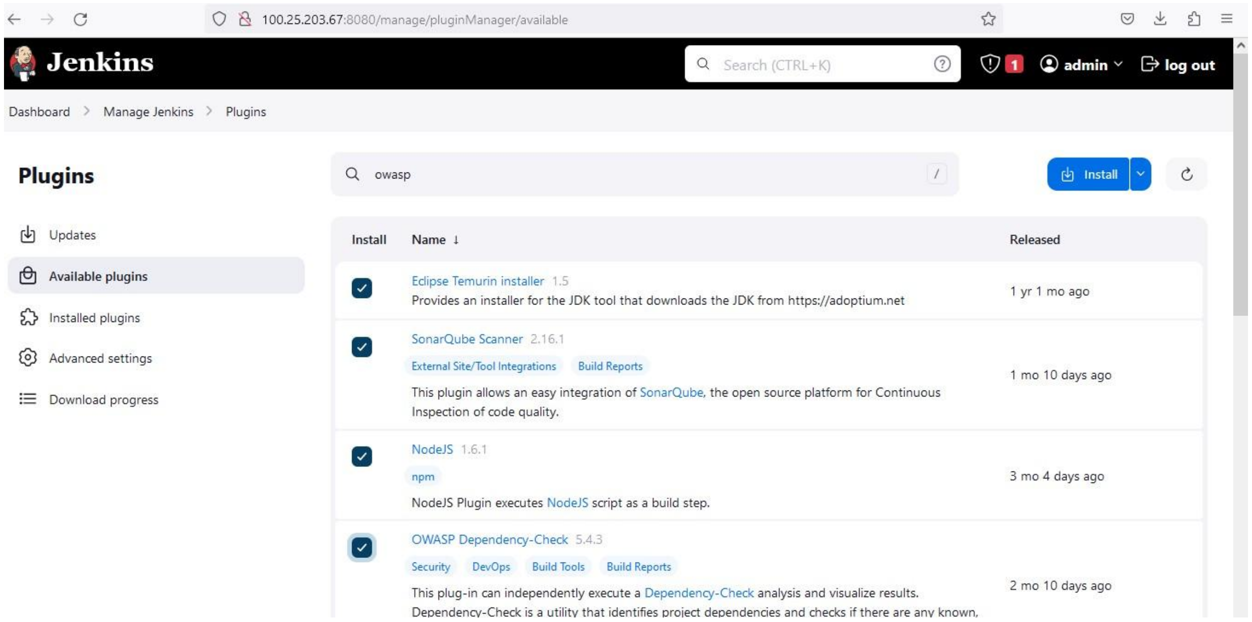
465

Credentials

kumarnaveen3098@gmail.com/****** (email)

✓ Use SSL

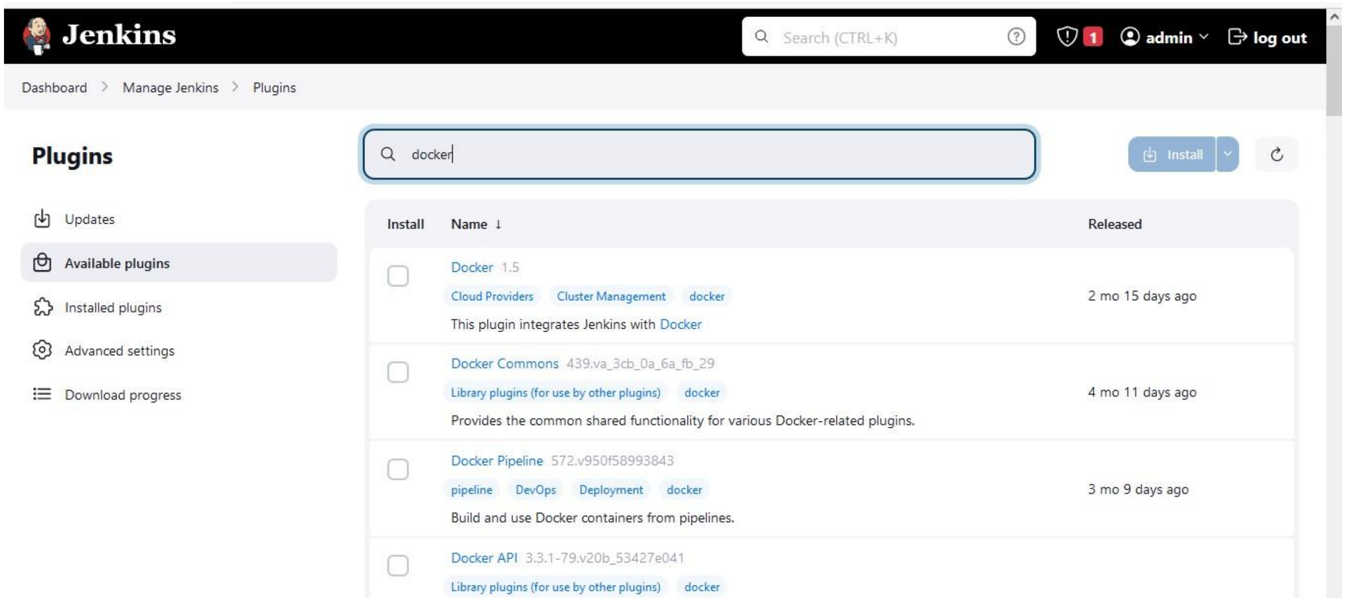
Step 61 : Download plugins like Eclipse Temurin Installer, sonarqube scanner, Node js, OWASP Depency check



100.25.203.67:8080/manage/pluginManager/available

Install	Name ↓	Released
✓	Eclipse Temurin installer 1.5 Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net	1 yr 1 mo ago
✓	SonarQube Scanner 2.16.1 External Site/Tool Integrations · Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	1 mo 10 days ago
✓	NodeJS 1.6.1 npm NodeJS Plugin executes NodeJS script as a build step.	3 mo 4 days ago
✓	OWASP Dependency-Check 5.4.3 Security · DevOps · Build Tools · Build Reports This plug-in can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known,	2 mo 10 days ago

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



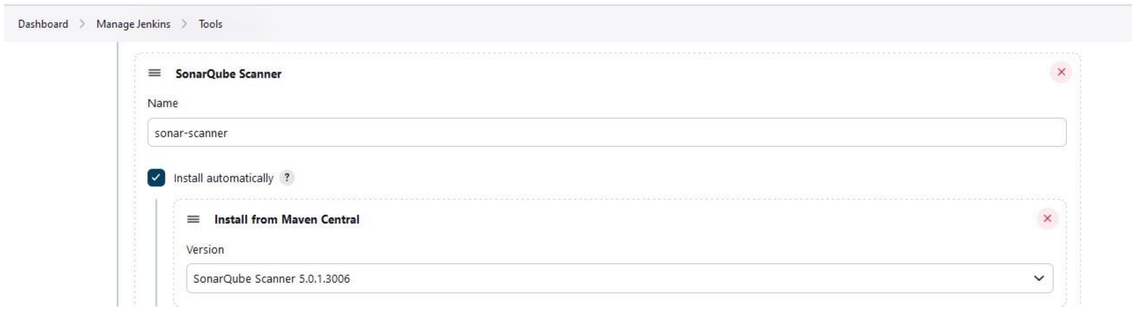
Dashboard > Manage Jenkins > Plugins

Plugins

Search: docker

Install	Name ↓	Released
☐	Docker 1.5 — Cloud Providers, Cluster Management, docker — This plugin integrates Jenkins with Docker	2 mo 15 days ago
☐	Docker Commons 439.va_3cb_0a_6a_fb_29 — Library plugins (for use by other plugins), docker — Provides the common shared functionality for various Docker-related plugins.	4 mo 11 days ago
☐	Docker Pipeline 572.v950f58993843 — pipeline, DevOps, Deployment, docker — Build and use Docker containers from pipelines.	3 mo 9 days ago
☐	Docker API 3.3.1-79.v20b_53427e041 — Library plugins (for use by other plugins), docker	

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



Dashboard > Manage Jenkins > Tools

SonarQube Scanner

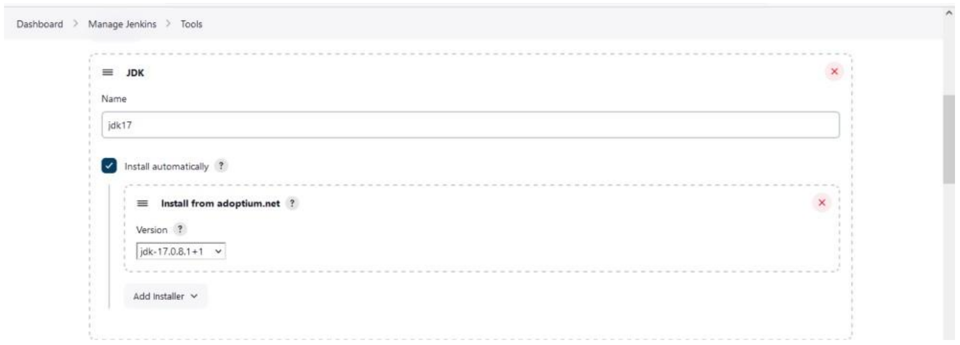
Name: sonar-scanner

☑ Install automatically

Install from Maven Central

Version: SonarQube Scanner 5.0.1.3006

Step 64 : Add the jdk info in Jenkins tools page



Dashboard > Manage Jenkins > Tools

JDK

Name: jdk17

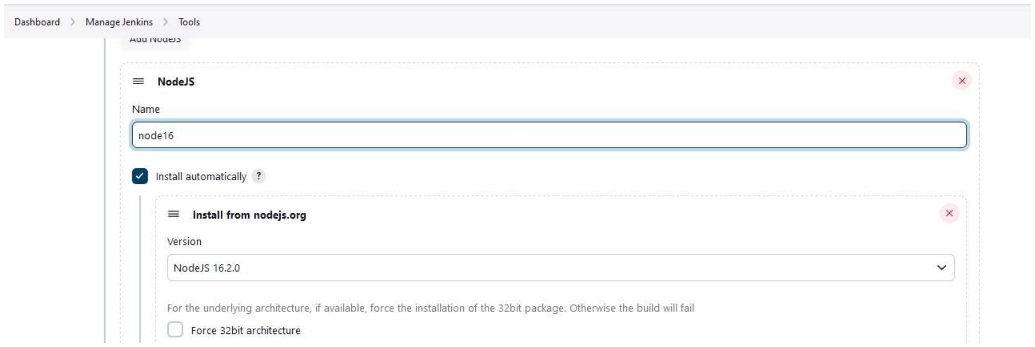
☑ Install automatically

Install from adoptium.net

Version: jdk-17.0.8.1+1

Add Installer

Step 65 : Add the nodejs info in Jenkins tools page


Dashboard > Manage Jenkins > Tools

Add NodeJS

NodeJS

Name

node16

Install automatically

Install from nodejs.org

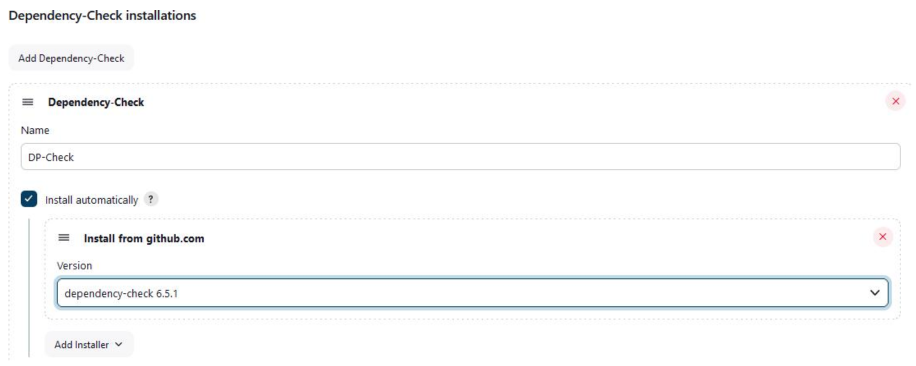
Version

NodeJS 16.2.0

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

Force 32bit architecture

Step 66 : Add the dependency info in Jenkins tools page


Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name

DP-Check

Install automatically

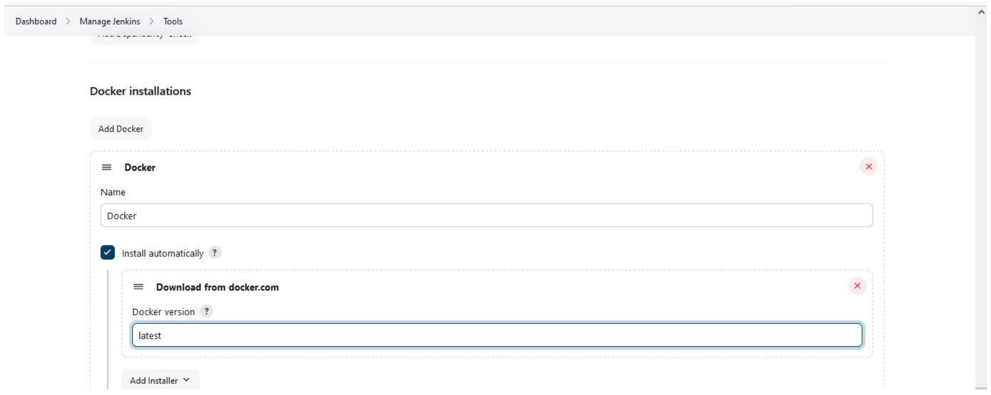
Install from github.com

Version

dependency-check 6.5.1

Add Installer

Step 67 : Add the docker info in Jenkins tools page


Dashboard > Manage Jenkins > Tools

Docker installations

Add Docker

Docker

Name

Docker

Install automatically

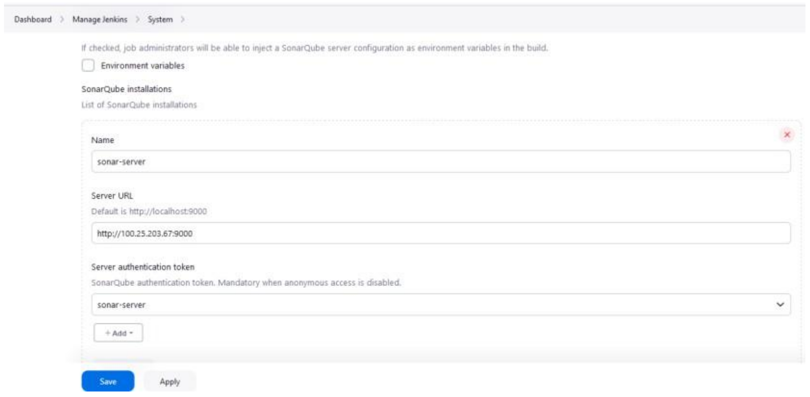
Download from docker.com

Docker version

latest

Add Installer

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



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 http://localhost:9000

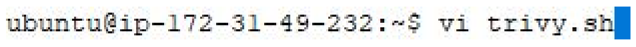
http://100.25.203.67:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-server

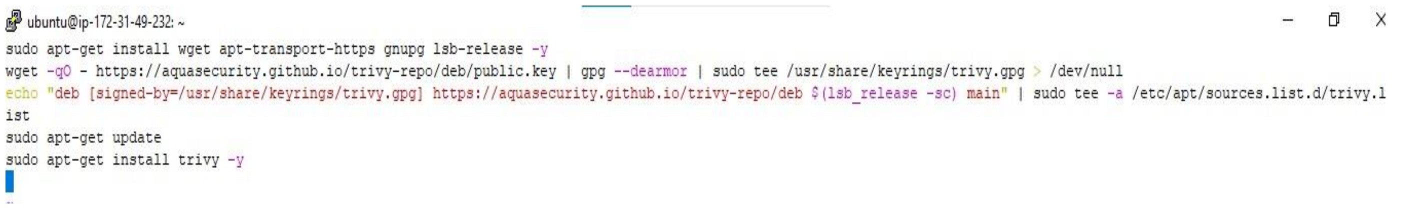
Step 69 : Then we need to install trivy, so we first of all create trivy shell script file



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

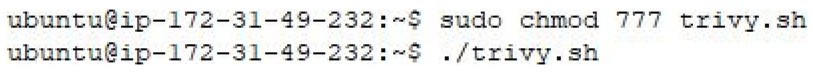
Step 70 : Add the trivy downloaded command in this file

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

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

echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy -y
```



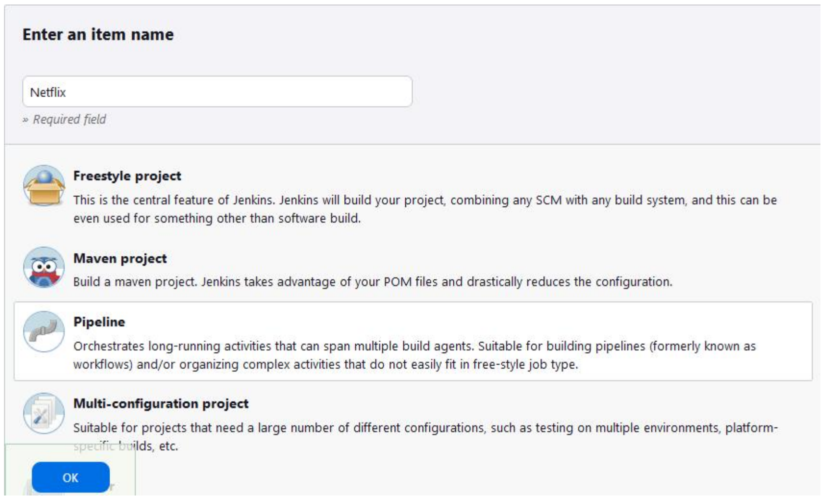
```
sudo apt-get install wget apt-transport-https gnupg lsb-release -y
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.l
ist
sudo apt-get update
sudo apt-get install trivy -y
```

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



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

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



Enter an item name

Netflix

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

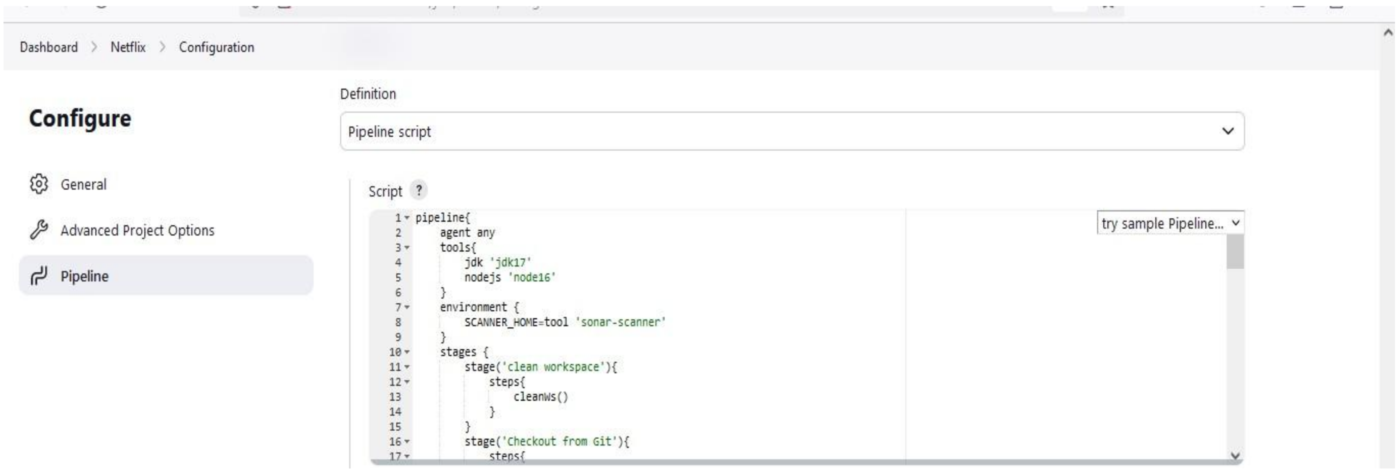
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Step 71 : Add the script in the pipeline

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



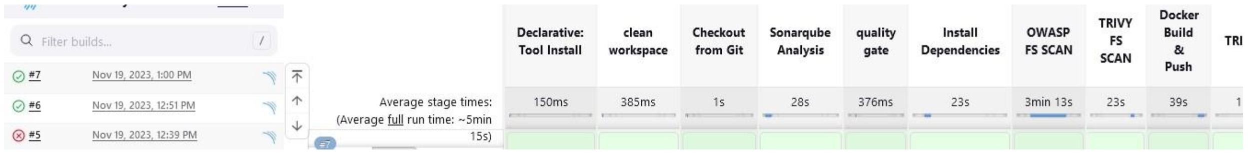
Definition

Pipeline script

Script

```
pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node16'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('clean workspace'){
            steps{
                cleanWs()
            }
        }
        stage('Checkout from Git'){
            steps{
```

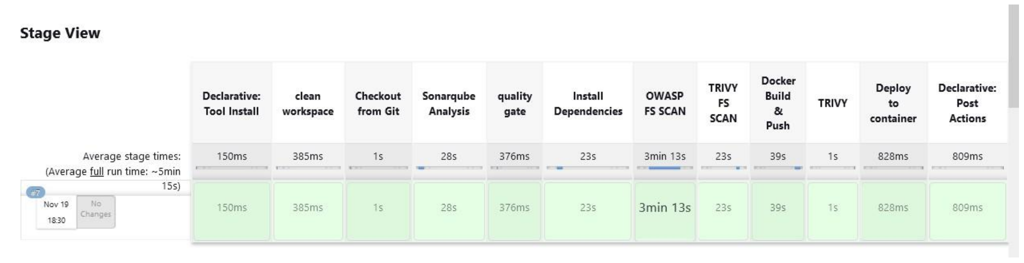
Step 72 : Run the pipeline


Filter builds...

#7	Nov 19, 2023, 1:00 PM
#6	Nov 19, 2023, 12:51 PM
#5	Nov 19, 2023, 12:39 PM

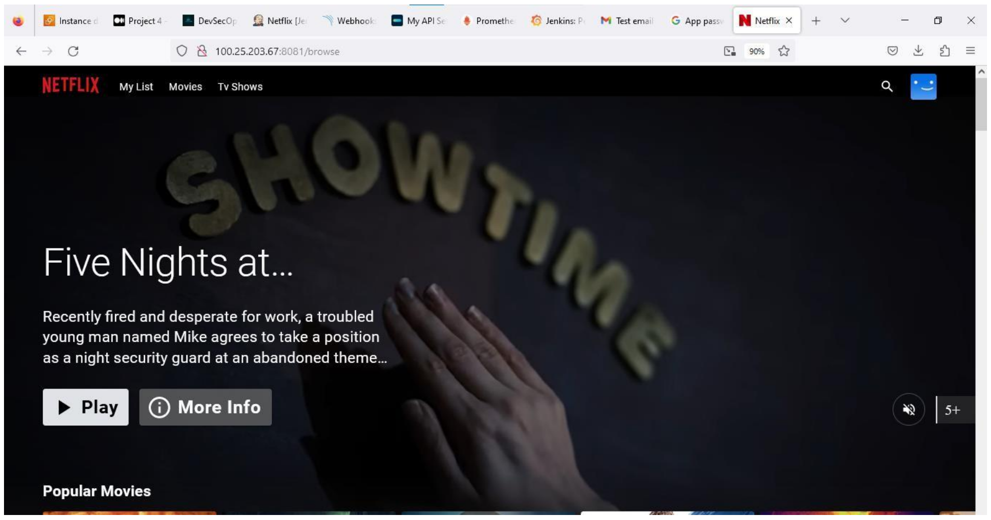
	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRI
Average stage times: (Average full run time: ~5min 15s)	150ms	385ms	1s	28s	376ms	23s	3min 13s	23s	39s	1

Step 73. Pipeline done and all stages are completed successfully


Stage View

	Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY	Deploy to container	Declarative: Post Actions
Average stage times: (Average full run time: ~5min 15s)	150ms	385ms	1s	28s	376ms	23s	3min 13s	23s	39s	1s	828ms	809ms
#7 Nov 19 18:30 No Changes	150ms	385ms	1s	28s	376ms	23s	3min 13s	23s	39s	1s	828ms	809ms

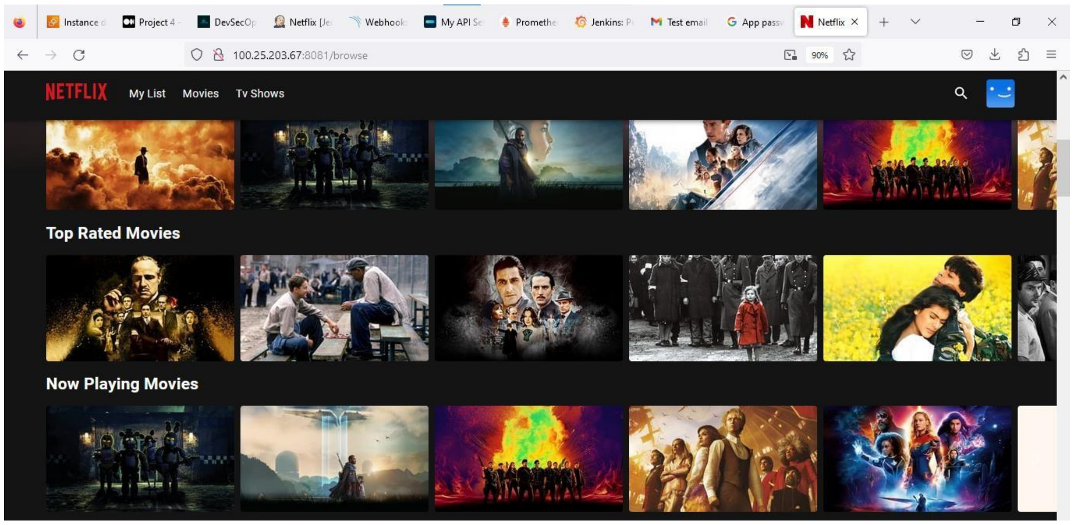
Step 73 : Copy the Ip with port no u get the Netflix page successfully



Five Nights at...

Recently fired and desperate for work, a troubled young man named Mike agrees to take a position as a night security guard at an abandoned theme...

Popular Movies


Instance
Project 4
DevSecOps
Netflix [Je
Webhook
My API Se
Promethe
Jenkins: P
Test email
App pass
Netflix
100.25.203.67:8081/browse
90%
NETFLIX
My List
Movies
Tv Shows
Top Rated Movies
Now Playing Movies