

# APPLICATION DEPLOYMENT

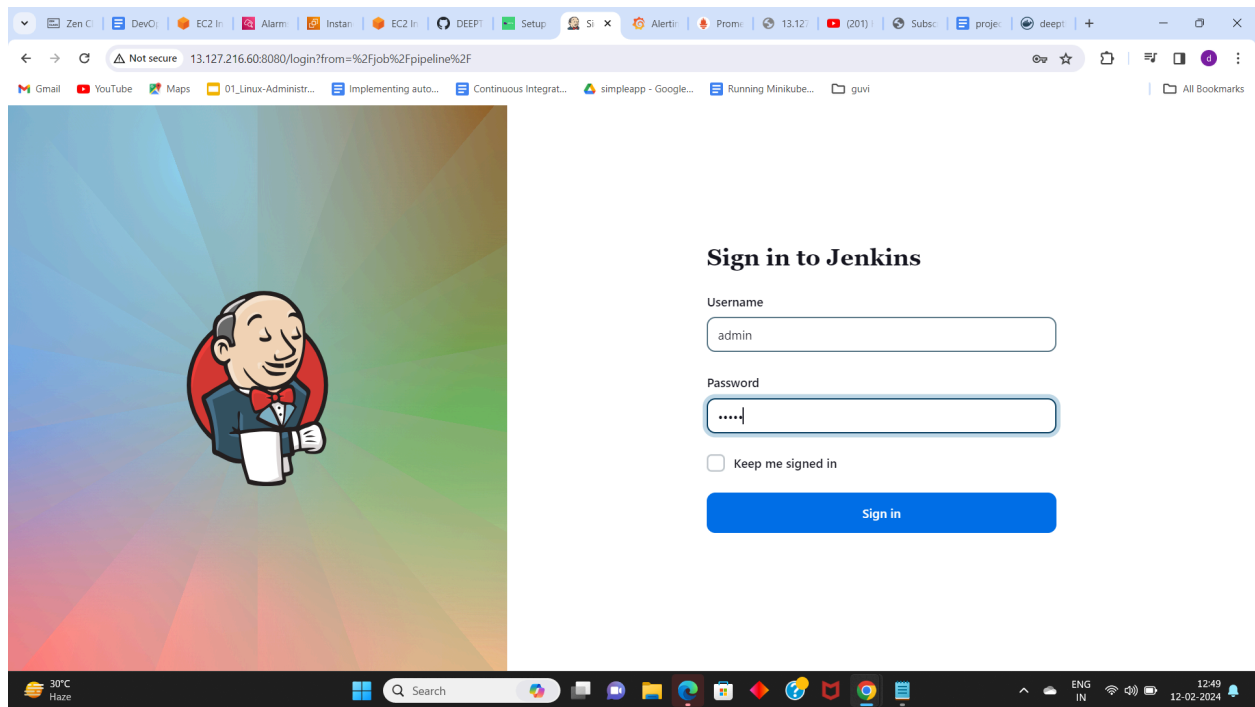
GITHUB REPO URL: <https://github.com/DEEPTHI-92/project-react.git>

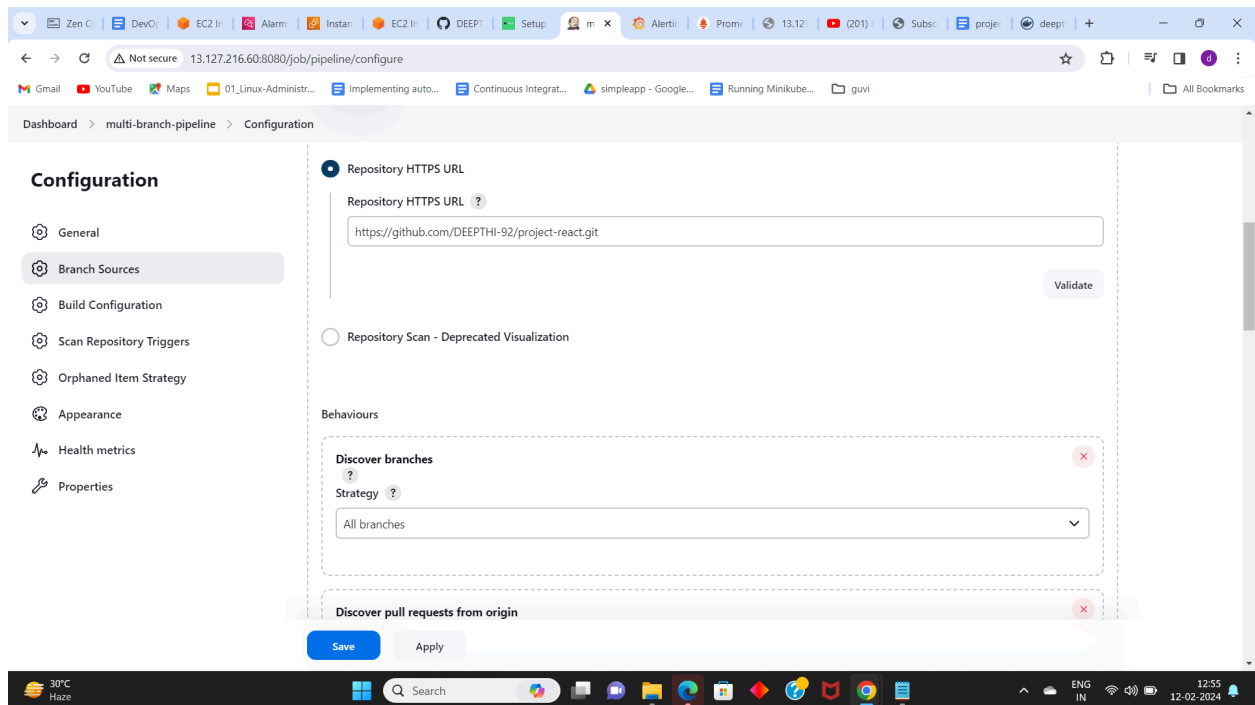
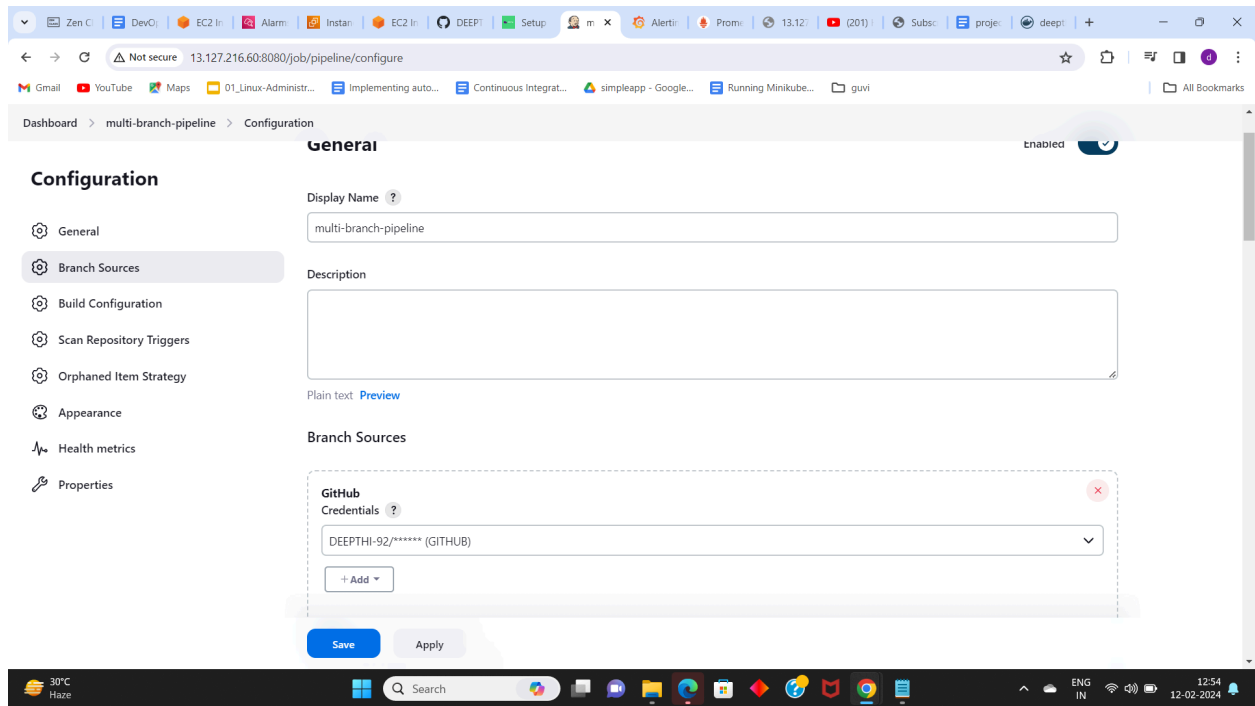
DEPLOYEDSITE URL: <http://13.127.216.60:80>

DOCKER TAGNAME : `deepthiarmour/dev` , `deepthiarmour/prod`

## SCREENSHOTS:

### JENKINS:





Dashboard > multi-branch-pipeline > Configuration

## Configuration

- General
- Branch Sources
- Build Configuration
- Scan Repository Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Add property ▾

Add source ▾

### Build Configuration

Mode  
by Jenkinsfile ▾

Script Path ?  
Jenkinsfile

### Scan Repository Triggers

☐ Periodically if not otherwise run ?

Save Apply

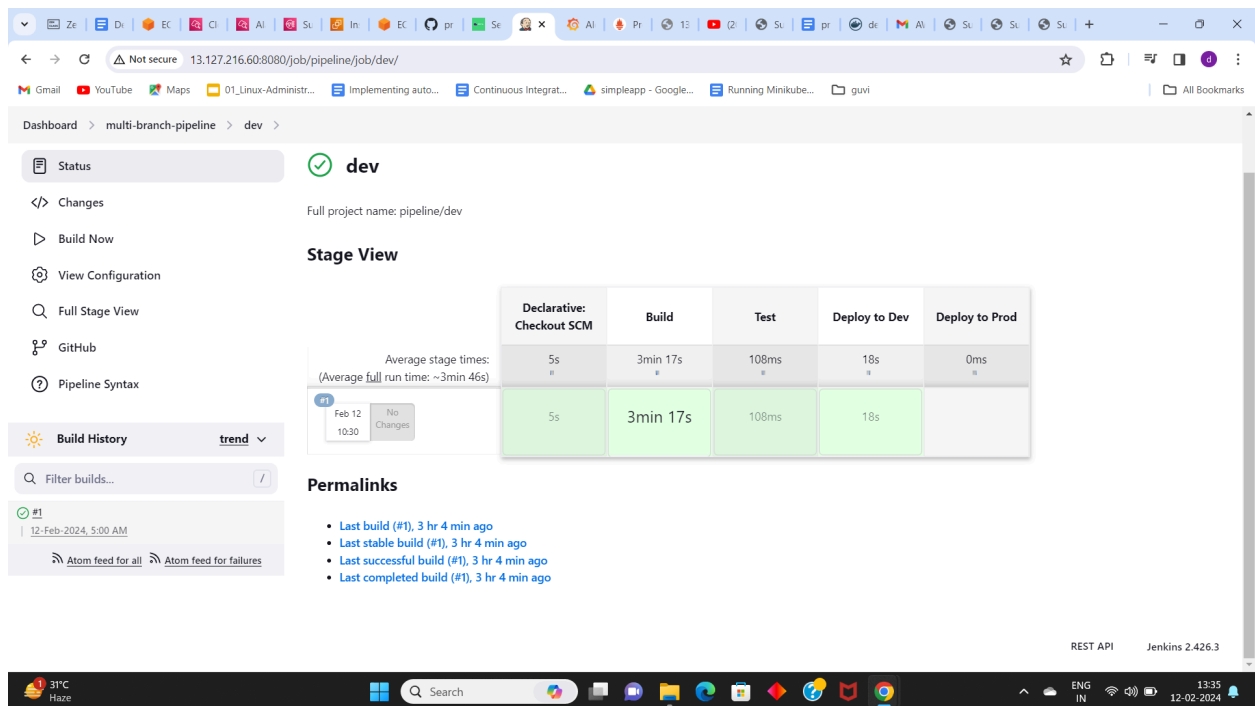
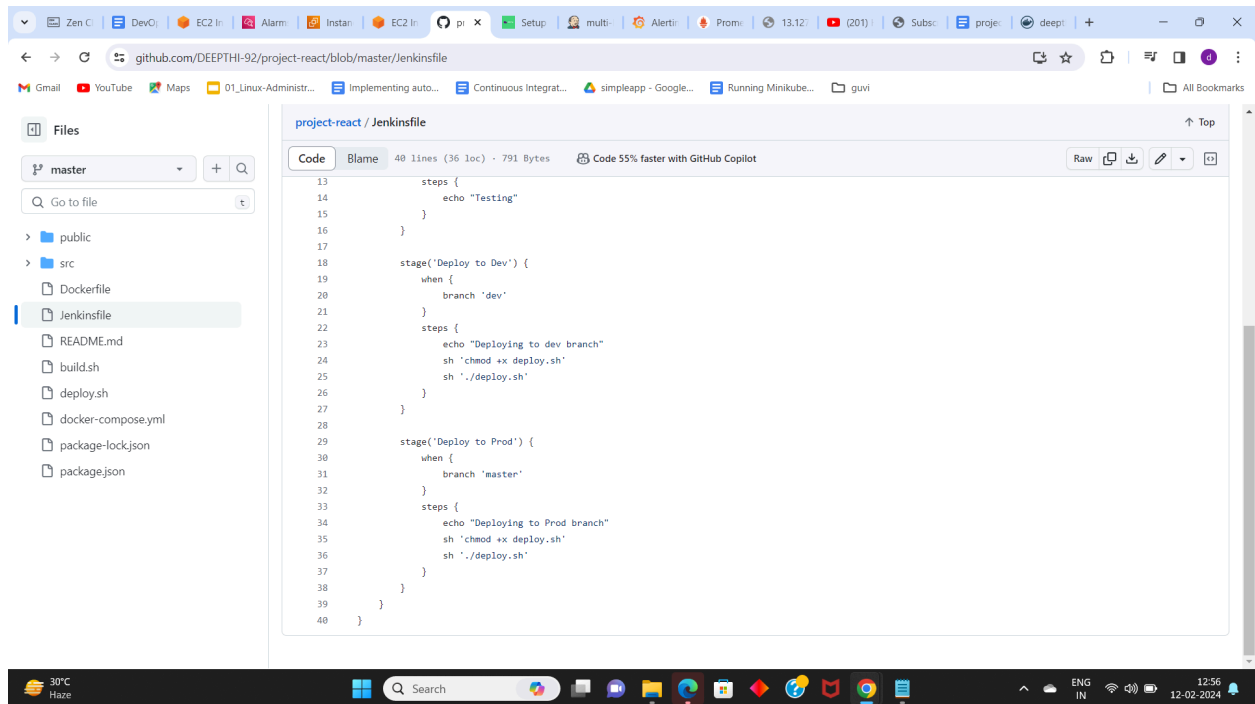
github.com/DEEPTHI-92/project-react/blob/master/Jenkinsfile

Files

- master
- Go to file
- public
- src
  - Dockerfile
  - Jenkinsfile
  - README.md
  - build.sh
  - deploy.sh
  - docker-compose.yml
  - package-lock.json
  - package.json

Code Blame 40 lines (36 loc) · 791 Bytes Code 55% faster with GitHub Copilot

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Build') {
6       steps {
7         sh 'chmod +x build.sh'
8         sh './build.sh'
9       }
10    }
11
12    stage('Test') {
13      steps {
14        echo "Testing"
15      }
16    }
17
18    stage('Deploy to Dev') {
19      when {
20        branch 'dev'
21      }
22      steps {
23        echo "Deploying to dev branch"
24        sh 'chmod +x deploy.sh'
25        sh './deploy.sh'
26      }
27    }
28
29    stage('Deploy to Prod') {
30      when {
31        branch 'master'
```



13.127.216.60:8080/job/pipeline/job/master/

**Jenkins** Search (CTRL+K) admin log out

Dashboard > multi-branch-pipeline > master

Status master

Full project name: pipeline/master

Stage View

	Declarative: Checkout SCM	Build	Test	Deploy to Dev	Deploy to Prod
Average stage times: (Average full run time: ~3min 45s)	2s	3min 20s	112ms	0ms	16s
#1 Feb 12 10:30 No Changes	2s	3min 20s	112ms		16s

Permalinks

- Last build (#1), 3 hr 5 min ago
- Last stable build (#1), 3 hr 5 min ago
- Last successful build (#1), 3 hr 5 min ago
- Last completed build (#1), 3 hr 5 min ago

Build History trend

Filter builds...

#1  
12-Feb-2024, 5:00 AM

Atom feed for all Atom feed for failures

31°C Haze

AWS :

ec2console:

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-055223fa6767c324f&osUser=ubuntu&region=ap-south-1&sshPort=22#/?

Last login: Mon Feb 12 04:48:45 2024 from 13.233.177.5

```
ubuntu@ip-172-31-40-208:~$ history
1 sudo apt update
2 sudo apt update
3 sudo apt install git
4 sudo apt install docker.io
5 sudo apt install docker-compose -y
6 sudo apt install openjdk-11-jdk -y
7 sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
8 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
9 sudo apt-get update
10 sudo apt-get install jenkins
11 git clone https://github.com/DEEPTHI-92/project-react.git
12 git clone https://github.com/DEEPTHI-92/prometheus.git
13 sudo su
14 ls
15 cd prometheus
16 ls
17 ./install-node-exporter.sh
18 cd
19 history
ubuntu@ip-172-31-40-208:~$
```

i-055223fa6767c324f (target-project)

PublicIPs: 13.127.216.60 PrivateIPs: 172.31.40.208

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

30°C Haze

## SECURITY GROUP:

The screenshot shows the AWS Management Console for a Security Group. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays the 'Inbound rules' and 'Outbound rules' sections.

**Inbound rules:**

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0c6d97223149a1a0a	3000	TCP	0.0.0.0/0
-	sgr-0e78396c7fbc336d3	80	TCP	0.0.0.0/0
-	sgr-053a071bf33b43c23	All	All	0.0.0.0/0
-	sgr-023c1e020ab2159ce	8080	TCP	0.0.0.0/0
-	sgr-0707fffbec04f839	22	TCP	0.0.0.0/0
-	sgr-06cf60be8474927	9090	TCP	0.0.0.0/0
-	sgr-03b5feec0a69b362e	9100	TCP	0.0.0.0/0

**Outbound rules:**

Name	Security group rule ID	Port range	Protocol	Destination
-	sgr-0165a56bc9a7d04ee	All	All	0.0.0.0/0

## DOCKERHUB REPO:

The screenshot shows the Docker Hub repository page for 'deepthiarmour/prod'. The page includes a search bar, navigation tabs (General, Tags, Builds, Collaborators, Webhooks, Settings), and a description of the repository. The 'Tags' section shows a single tag 'latest' pushed 2 hours ago. The 'Automated Builds' section provides information on how to connect the repository to GitHub or Bitbucket for automated builds.

**Repository:** deepthiarmour / prod

**Description:** This repository does not have a description. Last pushed: about 2 hours ago.

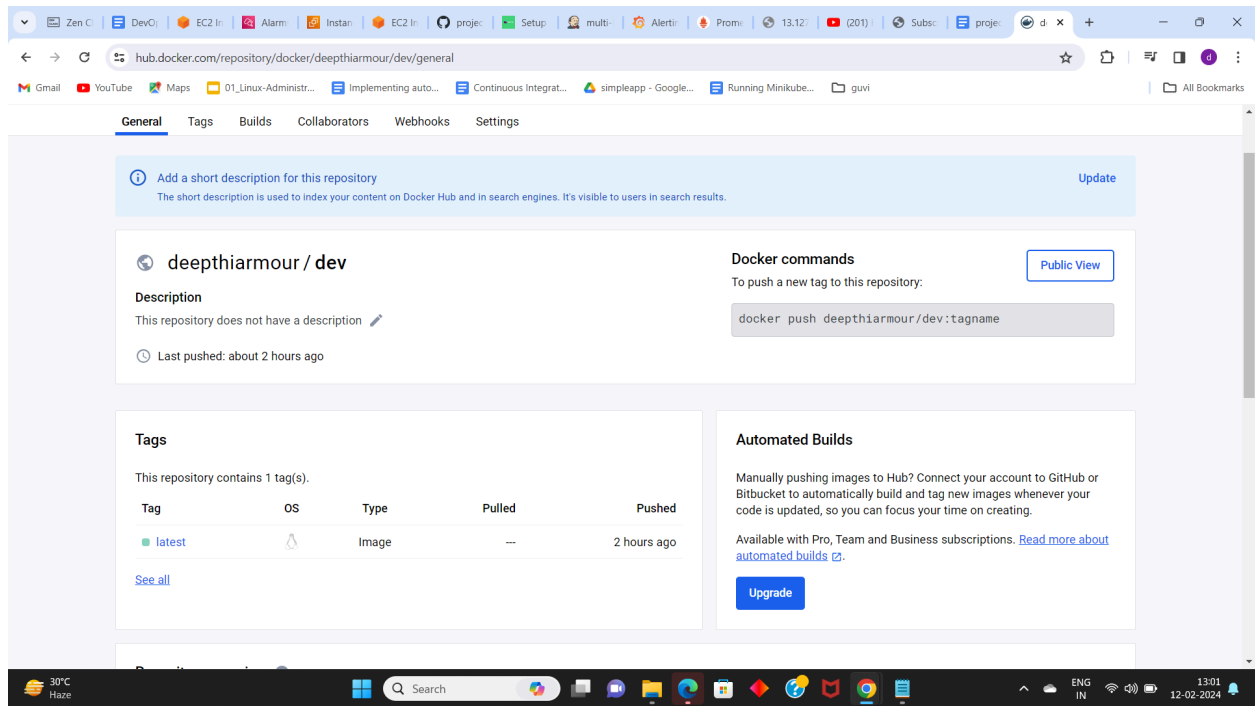
**Docker commands:** To push a new tag to this repository: `docker push deepthiarmour/prod:tagname`

**Tags:** This repository contains 1 tag(s).

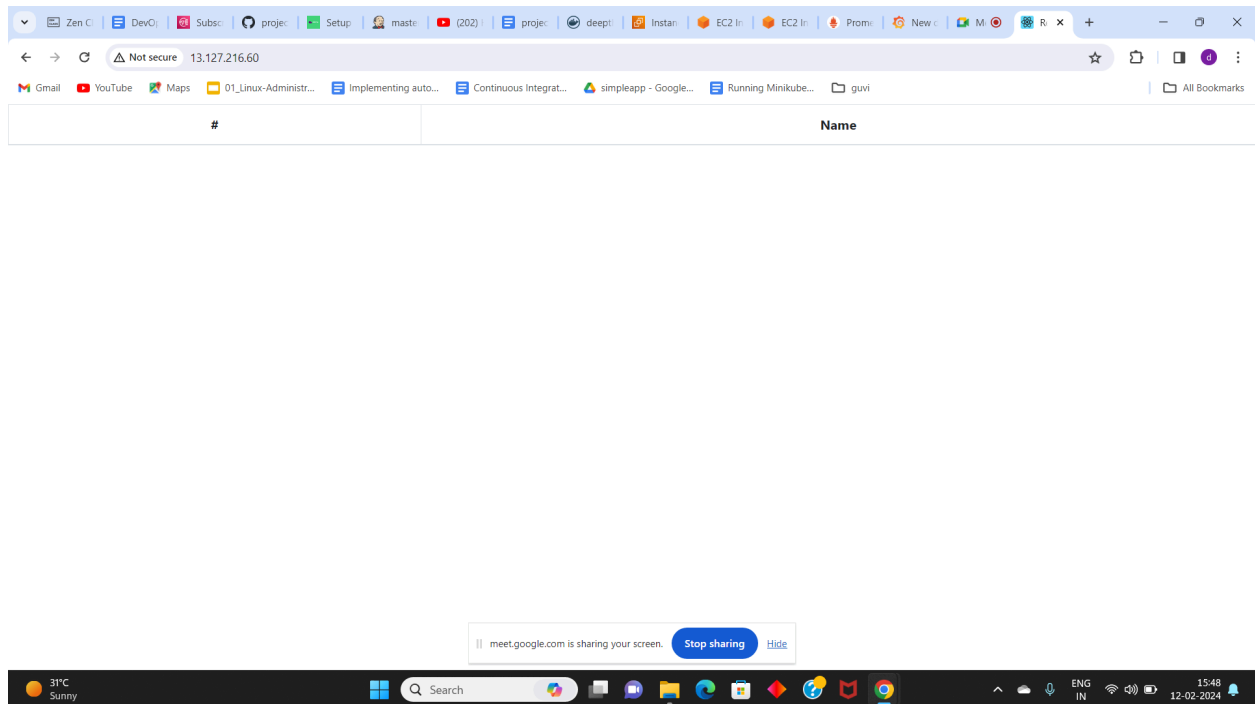
Tag	OS	Type	Pulled	Pushed
latest	linux	Image	---	2 hours ago

[See all](#)

**Automated Builds:** Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating. Available with Pro, Team and Business subscriptions. [Read more about automated builds](#). [Upgrade](#)

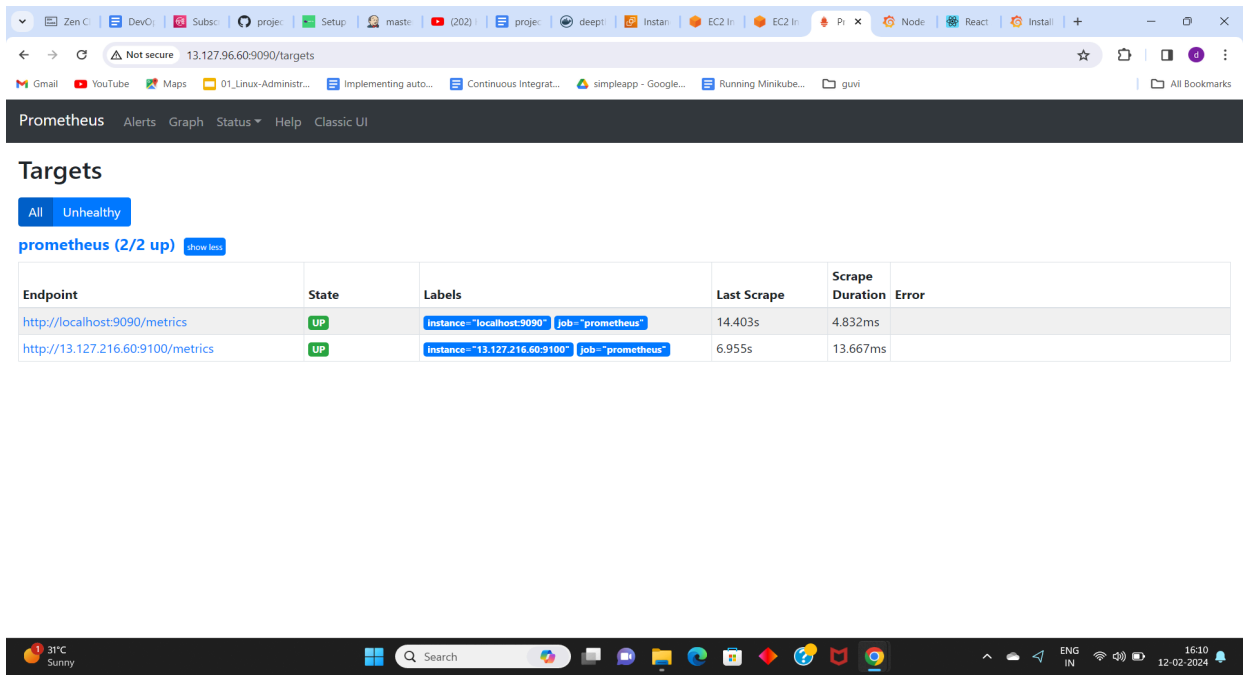


## DEPLOYED SITE :



# MONITORING TOOL:

# PROMETHEUS:



# GRAFANA:

