

Project

DevOps Application Deployment Capstone

GitHub Link - <https://github.com/Vijay254088/capstone-project.git>

Jenkins Console Output - https://drive.google.com/file/d/1xea4gLuNGRAPqp_7150kH4P6VcZR4Tcq/view?usp=sharing

Docker images name - capstoneimage

Create Ec2 Instance in Name of Capstone Project

Generate a Keypair in Name of Capstoneproject

After Creating EC2 instance Set a Security Group for nginx,jenkins ,docker-compose

```
PS C:\Users\balaji ravichandran> cd downloads
PS C:\Users\balaji ravichandran\downloads> ssh -i "capstoneproject.pem" ec2-user@ec2-18-206-231-249.compute-1.amazonaws.com
,
#_
~\_ ####_      Amazon Linux 2023
~~ \####\|
~~ \###|_
~~ \|/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V-`-->
~~ ./
~~ ./m/
[ec2-user@ip-172-31-89-19 ~]$
```

Cd into Downloads and launch the instance using Vs code

```
[ec2-user@ip-172-31-89-19 ~]$ sudo yum install git
Last metadata expiration check: 0:04:39 ago on Mon Mar 31 18:26:24 2025.
Dependencies resolved.
=====
Package          Architecture Version       Repository   Size
=====
Installing:
git              x86_64      2.47.1-1.amzn2023.0.2   amazonlinux 54 k
Installing dependencies:
git-core         x86_64      2.47.1-1.amzn2023.0.2   amazonlinux 4.7 M
git-core-doc    noarch      2.47.1-1.amzn2023.0.2   amazonlinux 2.8 M
perl-Error       noarch      1:0.17029-5.amzn2023.0.2  amazonlinux 41 k
perl-File-Find  noarch      1.37-477.amzn2023.0.6   amazonlinux 26 k
perl-Git         noarch      2.47.1-1.amzn2023.0.2   amazonlinux 42 k
perl-TermReadKey x86_64      2.38-9.amzn2023.0.2    amazonlinux 36 k
perl-lib         x86_64      0.65-477.amzn2023.0.6   amazonlinux 15 k
=====
Transaction Summary
=====
Install 8 Packages

Total download size: 7.7 M
Installed size: 37 M
Is this ok [y/N]: y
Downloading Packages:
(1/8): git-2.47.1-1.amzn2023.0.2.x86_64.rpm           1.2 MB/s | 54 kB  00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm   1.6 MB/s | 41 kB  00:00
(3/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm 869 kB/s | 26 kB  00:00
(4/8): git-core-doc-2.47.1-1.amzn2023.0.2.noarch.rpm   21 MB/s | 2.8 MB  00:00
(5/8): perl-Git-2.47.1-1.amzn2023.0.2.noarch.rpm     1.1 MB/s | 42 kB  00:00
(6/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 1.6 MB/s | 36 kB  00:00
```

Install git
Sudo yum install git -y

```
Complete!
[ec2-user@ip-172-31-89-19 ~]$ git clone https://github.com/sriram-R-krishnan/devops-build.git
Cloning into 'devops-build'...
remote: Enumerating objects: 21, done.
remote: Total 21 (delta 0), reused 0 (delta 0), pack-reused 21 (from 1)
Receiving objects: 100% (21/21), 720.09 KiB | 20.00 MiB/s, done.
[ec2-user@ip-172-31-89-19 ~]$
```

To Clone Git Repo With Your Local Machine

```
[ec2-user@ip-172-31-89-19 ~]$ vi Installingprerequisite
[New] 43L, 745B written
[ec2-user@ip-172-31-89-19 ~]$ sudo chmod a+x Installingprerequisite
[ec2-user@ip-172-31-89-19 ~]$ ./Installingprerequisite
Last metadata expiration check: 0:20:28 ago on Mon Mar 31 18:26:24 2025.
No match for argument: -y
Error: No packages marked for upgrade.
--2025-03-31 18:46:52-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'
```

Installing prerequisites with vi in name of **Installingprerequisites**

```
#importing key file from jenkins cli
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

#upgrade
sudo yum upgrade

#installing java
sudo yum install java-17-amazon-corretto -y

#installing jenkins
sudo yum install jenkins -y

#enable jenkins
sudo yum install jenkins -y

#start jenkins
sudo systemctl start jenkins

#status jenkins
sudo systemctl status jenkins

#installing docker
sudo yum install docker -y

#starting docker
sudo service docker start

#to run docker command without sudo
sudo usermod -a -G docker ec2-user

#checkdocker evrsion
```

In that installing java , jenkins , docker , docker-compose

```
[ec2-user@ip-172-31-89-19 ~]$ sudo chmod a+x  Installingprerequisite
[ec2-user@ip-172-31-89-19 ~]$ ./Installingprerequisite
Last metadata expiration check: 0:20:28 ago on Mon Mar 31 18:26:24 2025.
```

Then give all permissions and execute

```
[ec2-user@ip-172-31-89-19 ~]$ ls  
Installingprerequisite devops-build  
[ec2-user@ip-172-31-89-19 ~]$ cd devops-build  
[ec2-user@ip-172-31-89-19 devops-build]$ ls  
build
```

Then check with ls and cd into devops-build folder

```
[ec2-user@ip-172-31-89-19 devops-build]$ sudo docker --version  
Docker version 25.0.8, build 0bab007  
[ec2-user@ip-172-31-89-19 devops-build]$ vi dockerfile  
[ec2-user@ip-172-31-89-19 devops-build]$
```

After that check docker version and create docker file

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY build/ .
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

In devops-build path create docker file

```
Start Walkthroughs
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
version: "3.8"
services:
  web:
    image: capstoneimage
    container_name: capstone_nginx
    ports:
      - "8080:80"
    restart: always
```

In same path create docker-compose.yml file

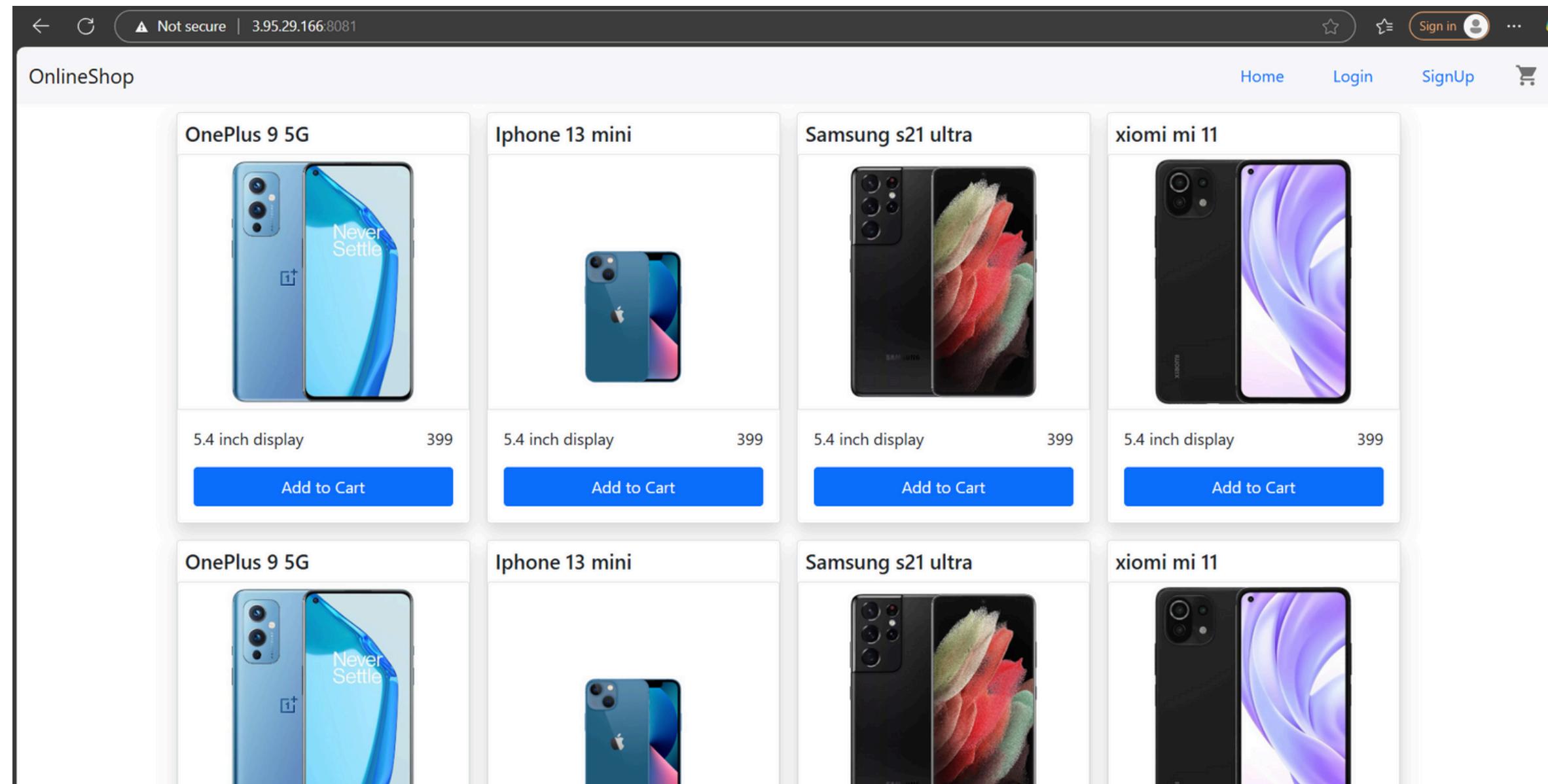
```
#building dockerimage  
└docker build -t capstoneimage .
```

Then create build.sh script file to build docker image
In That check docker images
Then give all permissions and execute

```
[ec2-user@ip-172-31-89-19 devops-build]$ sudo docker-compose up -d
[+] Running 1/1
✓ Container capstone_nginx Started
[ec2-user@ip-172-31-89-19 devops-build]$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS          NAMES
569b46602cf1   capstoneimage  "/docker-entrypoint..."  13 seconds ago  Up 13 seconds  0.0.0.0:8081->80/tcp, :::8081->80/tcp  capstone_nginx
[ec2-user@ip-172-31-89-19 devops-build]$
```

Then do `sudo docker-compose up -d`

Open a web browser and enter the URL



Docker File Output From Instance 1

Pushing a Docker image to docker hub with help of deploy.sh script



```
PROBLEMS OUTPUT TERMINAL PORTS
ssh + v ... ^

#!/bin/bash

#docker login
docker login -u vijay192001 -p Raghava@2024

#dcoker tag
docker tag capstoneimage vijay192001/dev

#docker push
docker push vijay192001/dev
~
```

Deploy.sh file

First Login to dockerhub

Then tag that image

Then Push the Image

Then give all permissions and execute

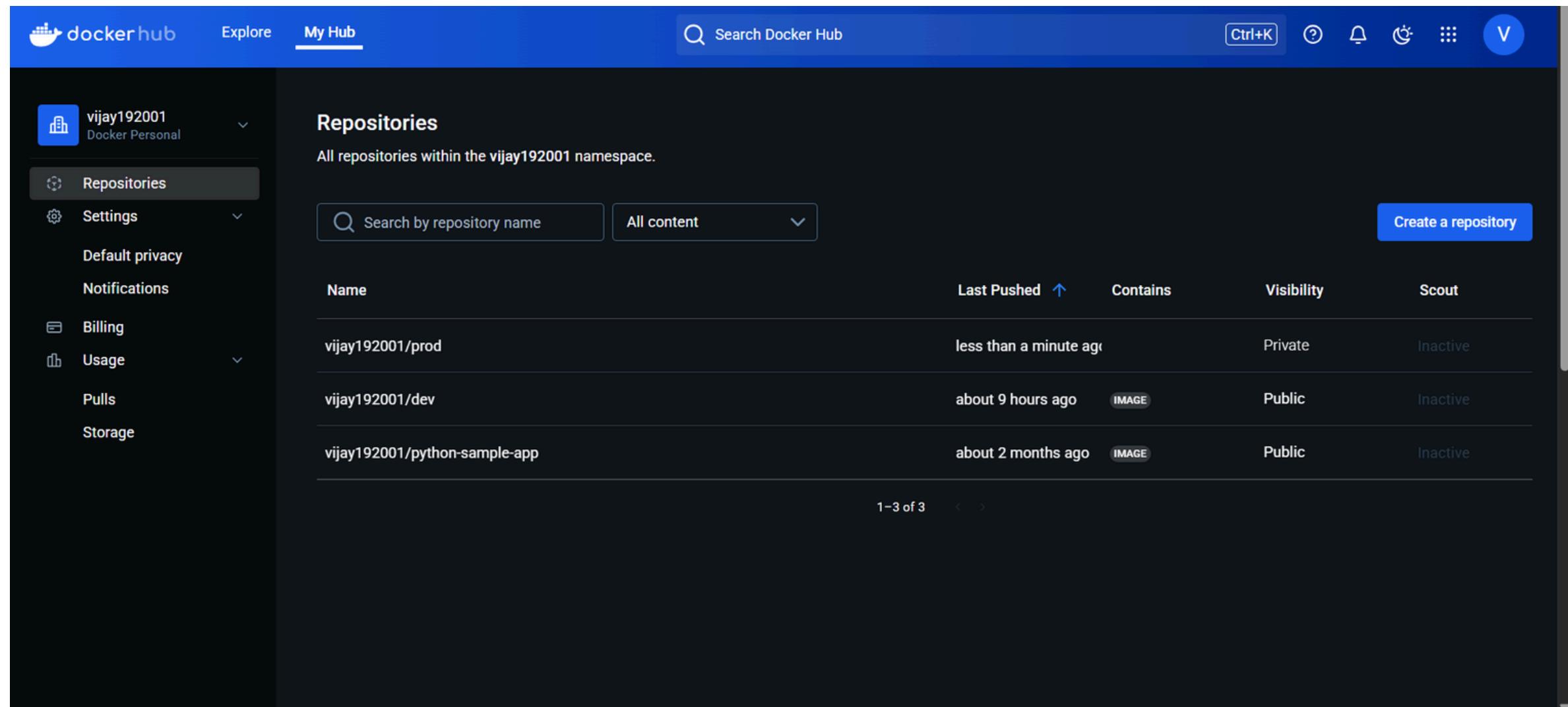
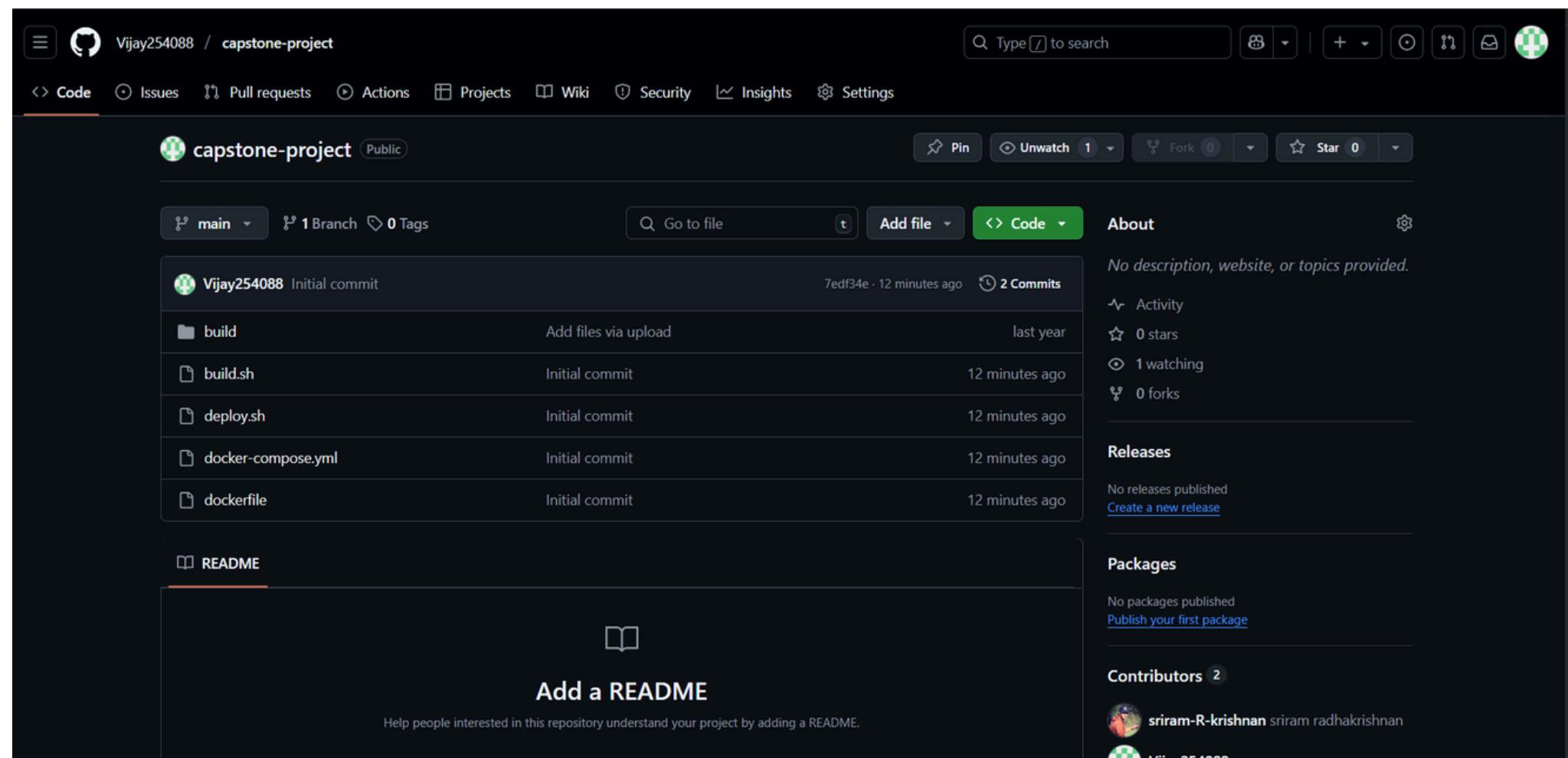


Image moved to docker hub

Install Git login into git

```
Last metadata expiration check: 1:05:32 ago on Mon Mar 31 18:46:53 2025.  
Package git-2.47.1-1.amzn2023.0.2.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-172-31-89-19 devops-build]$ git config --global user.name "Vijay254088"  
git config --global user.email "vijayakumar192.16.10.1@gmail.com"
```

Then Create a Repo in name of capstone image in github



Before this process generate clasic tocken

```
[ec2-user@ip-172-31-89-19 devops-build]$ git remote set-url origin https://github.com/vijay254088/capstone-project.git
[ec2-user@ip-172-31-89-19 devops-build]$ git push -u origin main
Username for 'https://github.com': vijay254088
Password for 'https://vijay254088@github.com':
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (27/27), 720.95 KiB | 80.11 MiB/s, done.
Total 27 (delta 0), reused 21 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote:   https://github.com/Vijay254088/capstone-project.git
To https://github.com/vijay254088/capstone-project.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-89-19 devops-build]$ ]
```

- git add .
- git commit -m "Initial commit"
- git push -u origin main
- git remote set-url origin https://github.com/vijay254088/capstone-project.git
- git push -u origin main

```
Vijay254088@ip-172-31-89-19:~/devops-build$ git branch dev
fatal: a branch named 'dev' already exists
[Vijay254088@ip-172-31-89-19:~/devops-build]$ git checkout dev
M    build.sh
M    deploy.sh
M    docker-compose.yml
Already on 'dev'
[Vijay254088@ip-172-31-89-19:~/devops-build]$ git push origin dev
Username for 'https://github.com': Vijay254088
Password for 'https://Vijay254088@github.com':
Everything up-to-date
```

After that from main branch move dev branch

- git push origin dev
- git branch dev
- git checkout dev
- git push origin dev

Create Two Repos Dev (Public)And Prod (Capstoneimage)(Private) For Pushing Images

The screenshot shows the Docker Hub interface under the "My Hub" tab. On the left, there's a sidebar for the user "vijay192001" with options like Repositories, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main area is titled "Repositories" and shows a list of repositories within the "vijay192001" namespace. The list includes:

Name	Last Pushed	Contains	Visibility	Scout
vijay192001/prod	less than a minute ago	IMAGE	Private	Inactive
vijay192001/dev	about 9 hours ago	IMAGE	Public	Inactive
vijay192001/python-sample-app	about 2 months ago	IMAGE	Public	Inactive

At the top right, there are search and filter options ("Search Docker Hub", "Ctrl+K", notifications, and a profile icon). A blue button at the top right says "Create a repository".

Login into Jenkins

The screenshot shows the Jenkins dashboard at the URL <http://54.157.38.236:8080>. The top navigation bar includes a 'Sign in' button and a 'vijayakumar' user profile. The main content area features a 'Welcome to Jenkins!' message and instructions for starting a project. On the left, there's a sidebar with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below this are sections for 'Build Queue' (empty) and 'Build Executor Status' (one node listed as 'offline'). The right side has sections for creating a job, setting up distributed builds, and configuring agents or clouds.

Not secure | 54.157.38.236:8080

Jenkins

Dashboard >

+ New Item

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 +

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds ?

Build Queue

No builds in the queue.

Build Executor Status 0/2

Built-In Node (offline)

REST API Jenkins 2.492.2

Install Necessary Plugins

The screenshot shows the Jenkins Manage Jenkins > Plugins page. A search bar at the top contains the text "GitHub Plugin". Below it, the "Installed plugins" section is selected. A table lists several GitHub-related plugins:

Name	Enabled
GitHub API Plugin	Enabled (blue switch)
GitHub Branch Source Plugin	Disabled (gray switch)
GitHub plugin	Enabled (blue switch)
Pipeline: GitHub Groovy Libraries	Enabled (blue switch)

At the bottom right of the page, it says "Jenkins 2.492.2".

The screenshot shows the Jenkins Manage Jenkins > Plugins page. A search bar at the top contains the text "Docker Pipeline Plugin". Below it, the "Installed plugins" section is selected. A table lists the Docker Pipeline plugin:

Name	Enabled
Docker Pipeline	Enabled (blue switch)

At the bottom right of the page, it says "Jenkins 2.492.2".

- GitHub Plugin: For GitHub integration.
- Docker Pipeline Plugin: For Docker build and push operations.
- Pipeline Plugin: For pipeline automation.

The screenshot shows the Jenkins Manage Jenkins > Plugins page. A search bar at the top contains the text "Pipeline Plugin". Below it, the "Installed plugins" section is selected. A table lists several Pipeline-related plugins:

Name	Enabled
Docker Pipeline	Enabled (blue switch)
Pipeline	Enabled (blue switch)
Pipeline Graph Analysis Plugin	Enabled (blue switch)
Pipeline Graph View Plugin	Enabled (blue switch)
Pipeline: API	Enabled (blue switch)
Pipeline: Basic Steps	Enabled (blue switch)

Create New Pipeline For Capstone-project

The screenshot shows the Jenkins web interface for creating a new item. The top navigation bar includes the Jenkins logo, a search icon, a notifications icon with two red dots, a security icon with two red dots, the user name 'vijayakumar', and a 'log out' button. The main title 'New Item' is displayed above a form where the item name 'capstone-project' has been entered. Below the name input, there is a section titled 'Select an item type' containing four options: 'Freestyle project', 'Pipeline', 'Multi-configuration project', and 'Folder'. The 'Pipeline' option is highlighted with a light gray background. At the bottom of the form is a blue 'OK' button.

Dashboard > All > New Item

New Item

Enter an item name

capstone-project

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different

OK

Jenkins

Dashboard > capstone-project1 > Configuration

Configure General

Enabled

Description
Creating Capstone Project

Plain text [Preview](#)

Discard old builds [?](#)

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

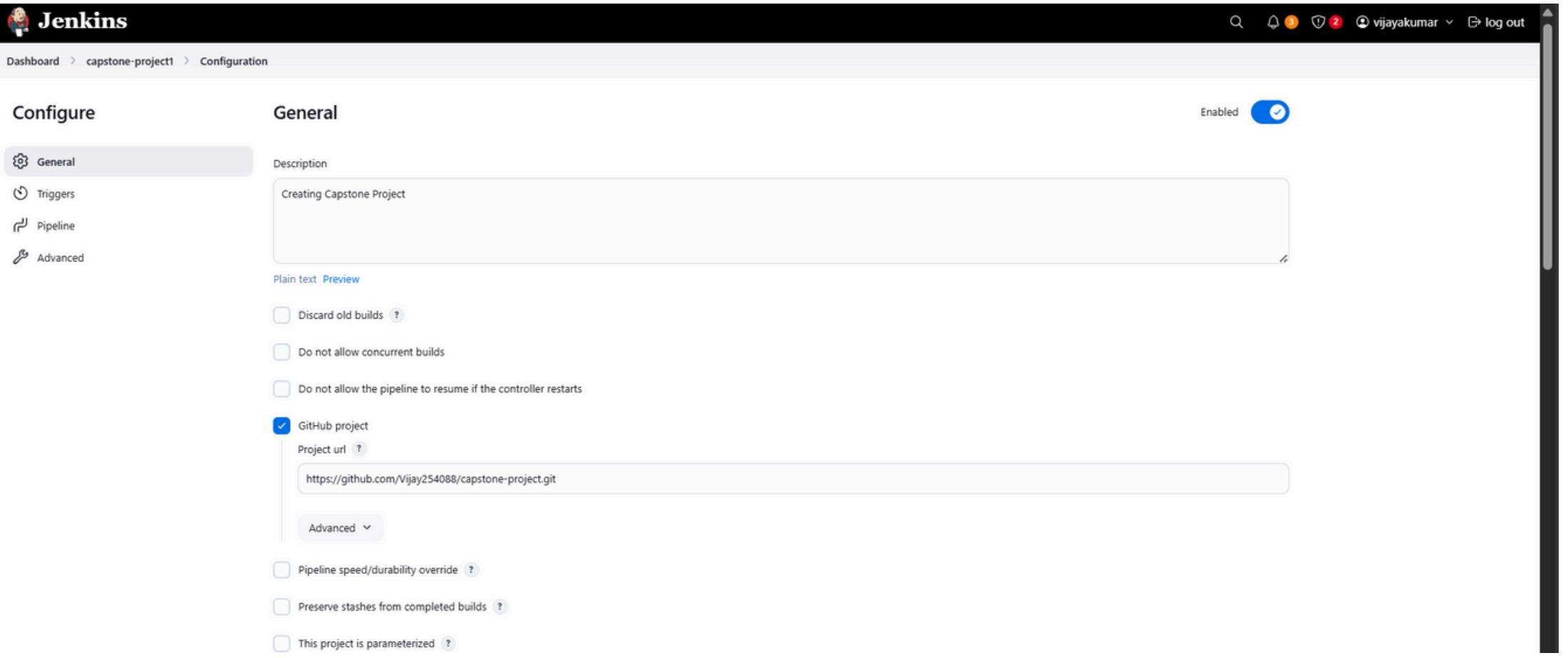
Project url [?](#)
`https://github.com/Vijay254088/capstone-project.git`

Advanced [▼](#)

Pipeline speed/durability override [?](#)

Preserve stashes from completed builds [?](#)

This project is parameterized [?](#)



Not secure | 3.87.204.245:8080/jenkins/capstone-project1/configure

Dashboard > capstone-project1 > Configuration

Configure

General

Definition
Pipeline script from SCM

SCM
Git

Repositories
Repository URL
`https://github.com/Vijay254088/capstone-project.git`

Credentials
Vijay254088

+ Add

Advanced [▼](#)

Add Repository

Branches to build
Branch Specifier (blank for 'any')
*main

Add Branch

Repository browser
(Auto)

Additional Behaviours
Add

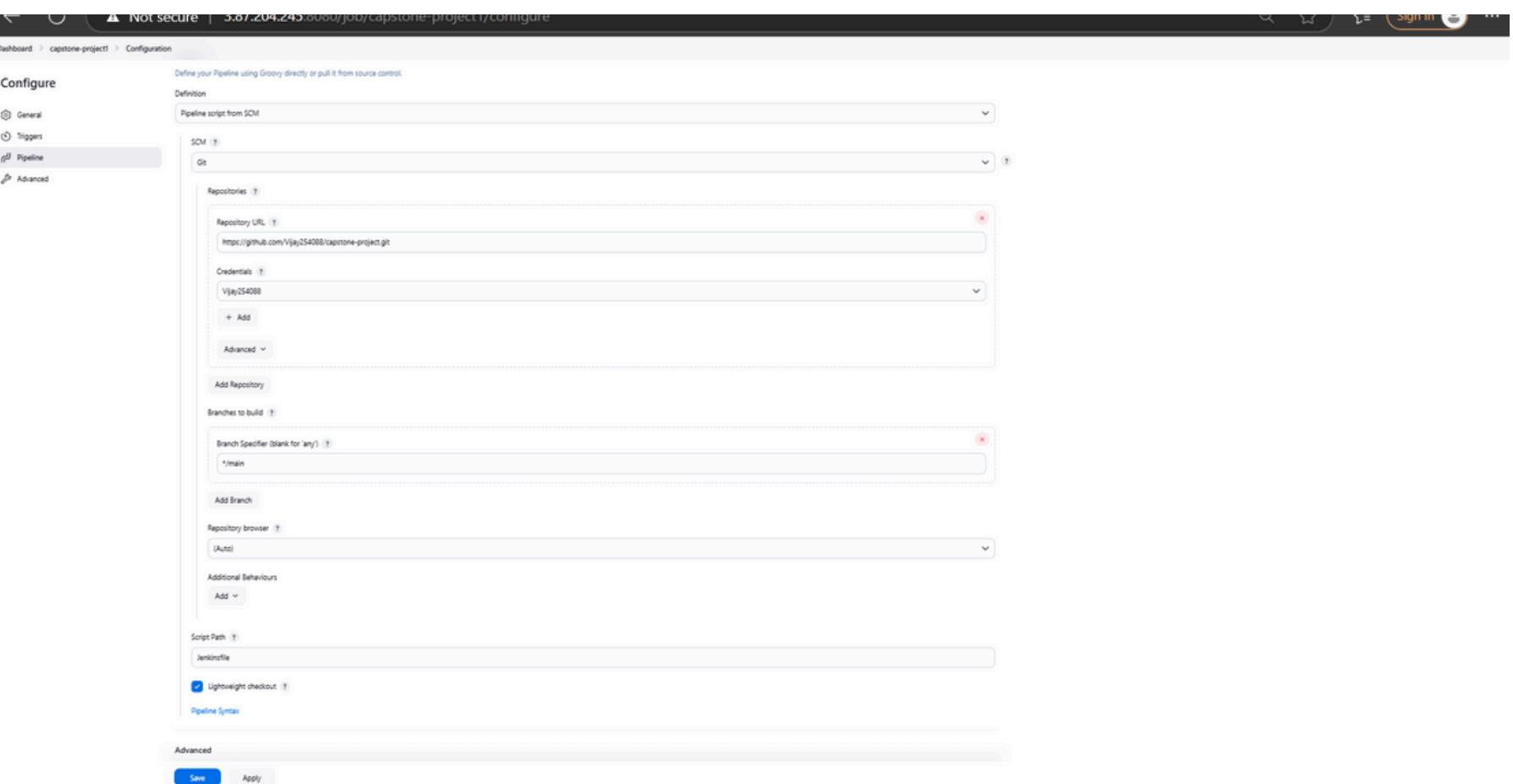
Script Path
Jenkinsfile

Lightweight checkout [?](#)

Pipeline Syntax

Advanced

Save Apply



Project Description:

- In the description section, write: "Creating Capstone Project".

GitHub Repository:

- In the GitHub project settings, add the following repository URL:
- <https://github.com/Vijay254088/capstone-project.git>

Jenkins Pipeline Setup:

- In Jenkins, select Pipeline Script from SCM as the definition.

Select SCM:

- Choose Git as the Source Code Management (SCM) option.

Git Configuration:

- Add the Git repository URL: <https://github.com/Vijay254088/capstone-project.git>.
- Add the appropriate credentials for accessing the Git repository.

Branch Selection:

- Select the main branch for the pipeline.

Jenkinsfile Location:

- In the pipeline configuration, specify the path to the Jenkinsfile as:
- Jenkinsfile (or the path relative to the repository root where your Jenkinsfile is located).

Dashboard > Manage Jenkins > System

In This Path add Docker Docker credentials

The screenshot shows the Jenkins Global properties screen under the System configuration. The 'Environment variables' checkbox is selected, and two variables are listed:

- DOCKER-USR**: Value: vijay192001
- DOCKER_PSWD**: Value: Raghava@2024

An 'Add' button is visible at the bottom left.

Dashboard >

+ New Item

Build History All +

Manage Jenkins

My Views

Build Queue (1) part of capstone-project #5

Build Executor Status (0 of 2 executors busy)

S W Name Last Success Last Failure Last Duration

capstone-project 1 hr 2 min #4 1 hr 25 min #3 34 min

Icon: S M L

Dashboard > capstone-project > #4

Status #4 (Apr 3, 2025, 6:07:54 PM) Add description Keep this build forever

</> Changes

Console Output Started by user vijayakumar Started 1 hr 5 min ago Took 34 min

Edit Build Information

Delete build '#4'

Timings

Git Build Data Revision: 1022e46e7982a1bfba9ddefd60c5ae02ba3d320 Repository: <https://github.com/Vijay254088/capstone-project.git>

Pipeline Overview

Pipeline Console

Restart from Stage

</> No changes.

Replay

Pipeline Steps

Workspaces

← Previous Build

→ Next Build

Console Output of Jenkins

ashboard > capstone-project > #4

Status Changes Console Output Edit Build Information Delete build '#4' Timings Git Build Data Pipeline Overview Pipeline Console Restart from Stage Replay Pipeline Steps Workspaces Previous Build Next Build

Console Output

Started by user vijayakumar
Obtained Jenkinsfile from git <https://github.com/Vijay254088/capstone-project.git>
[Pipeline] Start of Pipeline
[Pipeline] node
Still waiting to schedule task
Waiting for next available executor
Resuming build at Thu Apr 03 18:33:55 UTC 2025 after Jenkins restart
Ready to run at Thu Apr 03 18:33:55 UTC 2025
Resuming build at Thu Apr 03 18:38:31 UTC 2025 after Jenkins restart
Ready to run at Thu Apr 03 18:38:31 UTC 2025
Running on Jenkins in /var/lib/jenkins/workspace/capstone-project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/capstone-project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/Vijay254088/capstone-project.git> # timeout=10
Fetching upstream changes from <https://github.com/Vijay254088/capstone-project.git>
> git --version # timeout=10
> git --version # 'git version 2.47.1'
> git fetch --tags --force --progress -- <https://github.com/Vijay254088/capstone-project.git> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checkout out Revision 1022e46e7982a1bfba9ddeaf6d0c5ae02ba3d320 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 1022e46e7982a1bfba9ddeaf6d0c5ae02ba3d320 # timeout=10
Commit message: "Merge pull request #3 from Vijay254088/dev"
> git rev-list --no-walk 1022e46e7982a1bfba9ddeaf6d0c5ae02ba3d320 # timeout=10
[Pipeline] }

Download Copy View as plain text

The docker image pushed to prod repo in docker hub

The screenshot shows the Docker Hub interface for the user 'vijay192001'. The left sidebar contains navigation links: 'Repositories' (selected), 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage' (with 'Pulls' and 'Storage' sub-links), and 'Create a repository'. The main content area is titled 'Repositories' and displays three entries:

Name	Last Pushed	Contains	Visibility	Scout
vijay192001/prod	37 minutes ago	IMAGE	Private	Inactive
vijay192001/dev	1 day ago	IMAGE	Public	Inactive
vijay192001/python-sample-app	about 2 months ago	IMAGE	Public	Inactive

At the bottom, it says '1-3 of 3'.

Create Instance 2 For Capstone-Project

The screenshot shows the AWS EC2 Instances details page for an instance named "Capstone-project instance 2".

Instance summary:

- Instance ID:** i-0c8e492d44815b2d2
- Public IPv4 address:** 54.210.54.142 | [open address](#)
- Private IPv4 addresses:** 172.31.95.82
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-95-82.ec2.internal
- Public IPv4 DNS:** ec2-54-210-54-142.compute-1.amazonaws.com | [open address](#)
- Instance type:** t2.micro
- VPC ID:** vpc-0b2b782ea4280df32
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)
- Subnet ID:** subnet-0e10cedf84b3b9e64
- Auto Scaling Group name:** -
- Instance ARN:** arn:aws:ec2:us-east-1:038462760044:instance/i-0c8e492d44815b2d2
- Managed:** false

Details Tab:

- AMI ID:** ami-00a929b66ed6e0de6
- Monitoring:** disabled
- Platform details:** Linux/UNIX
- AMI name:** al2023-ami-2023.7.20250331.0-kernel-6.1-x86_64
- Allowed image:** -
- Termination protection:** Disabled

Security Group For Instance 2

The screenshot shows the AWS EC2 Security Groups console. The left sidebar is collapsed. The main area displays the details of a security group named "launch-wizard-13". A green banner at the top indicates that inbound security group rules were successfully modified. Below the banner, the security group's name, ID, owner, and VPC ID are listed. The "Inbound rules" tab is selected, showing 8 permission entries. The table lists the following inbound rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-01c68ab07a557fe9f	IPv4	Custom TCP	TCP	9090	0.0.0.0/0	-
-	sgr-02b31c379cc8086b0	IPv4	Custom TCP	TCP	3000	0.0.0.0/0	-
-	sgr-0a4ec1ab4e55e5259	IPv4	Custom TCP	TCP	9091	0.0.0.0/0	-
-	sgr-05f7c338f595078d8	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-00d4ed229247953c3	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-09e818dfb648bc036	IPv4	Custom TCP	TCP	8081	0.0.0.0/0	-
-	sgr-0fb8bb111eeecd0b	IPv4	Custom TCP	TCP	9100	0.0.0.0/0	-
-	sgr-07bcbba187040b3a0	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	-

In that Ec2 instance Install docker , Start Docker , Enable docker

```
Complete!
Redirecting to /bin/systemctl start docker.service
Docker version 25.0.8, build 0bab007
[ec2-user@ip-172-31-95-82 ~]$ vi Installingprerequisite
[ec2-user@ip-172-31-95-82 ~]$ 11L, 201B written
[ec2-user@ip-172-31-95-82 ~]$ sudo yum update -y
Last metadata expiration check: 0:05:28 ago on Thu Apr  3 19:30:45 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-95-82 ~]$ 
```

The screenshot shows a terminal window in Visual Studio Code with the following content:

```
Installing docker
do yum install docker -y

Starting docker
do service docker start

o run docker command without sudo
do usermod -a -G docker ec2-user

Check docker version
docker --version
```

The terminal tab is selected in the top navigation bar. The status bar at the bottom right shows "11,16" and "All". On the far right, there are icons for "ssh" and "sftp".

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
installing docker
sudo yum install docker -y

starting docker
sudo service docker start

to run docker command without sudo
sudo usermod -a -G docker ec2-user

check docker evrision
docker --version

enable docker
sudo systemctl enable docker
```

Docker Login

```
[ec2-user@ip-172-31-95-82 ~]$ sudo docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: vijay192001
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Pull Docker image to instance 2

```
Login Succeeded
[ec2-user@ip-172-31-95-82 ~]$ sudo docker pull vijay192001/dev:latest
latest: Pulling from vijay192001/dev
6e909acdb790: Pull complete
5sea34f5b9c2: Pull complete
417c4bccf534: Pull complete
e7e0ca015e55: Pull complete
373fe654e984: Pull complete
97f5c0f51d43: Pull complete
c22eb46e871a: Pull complete
4f4fb700ef54: Pull complete
eb86de2b24e8: Pull complete
Digest: sha256:19edf3ad7249cd14af394e5efabf041edb8b186765e6fc8dcf60f4de3be4de67
Status: Downloaded newer image for vijay192001/dev:latest
docker.io/vijay192001/dev:latest
```

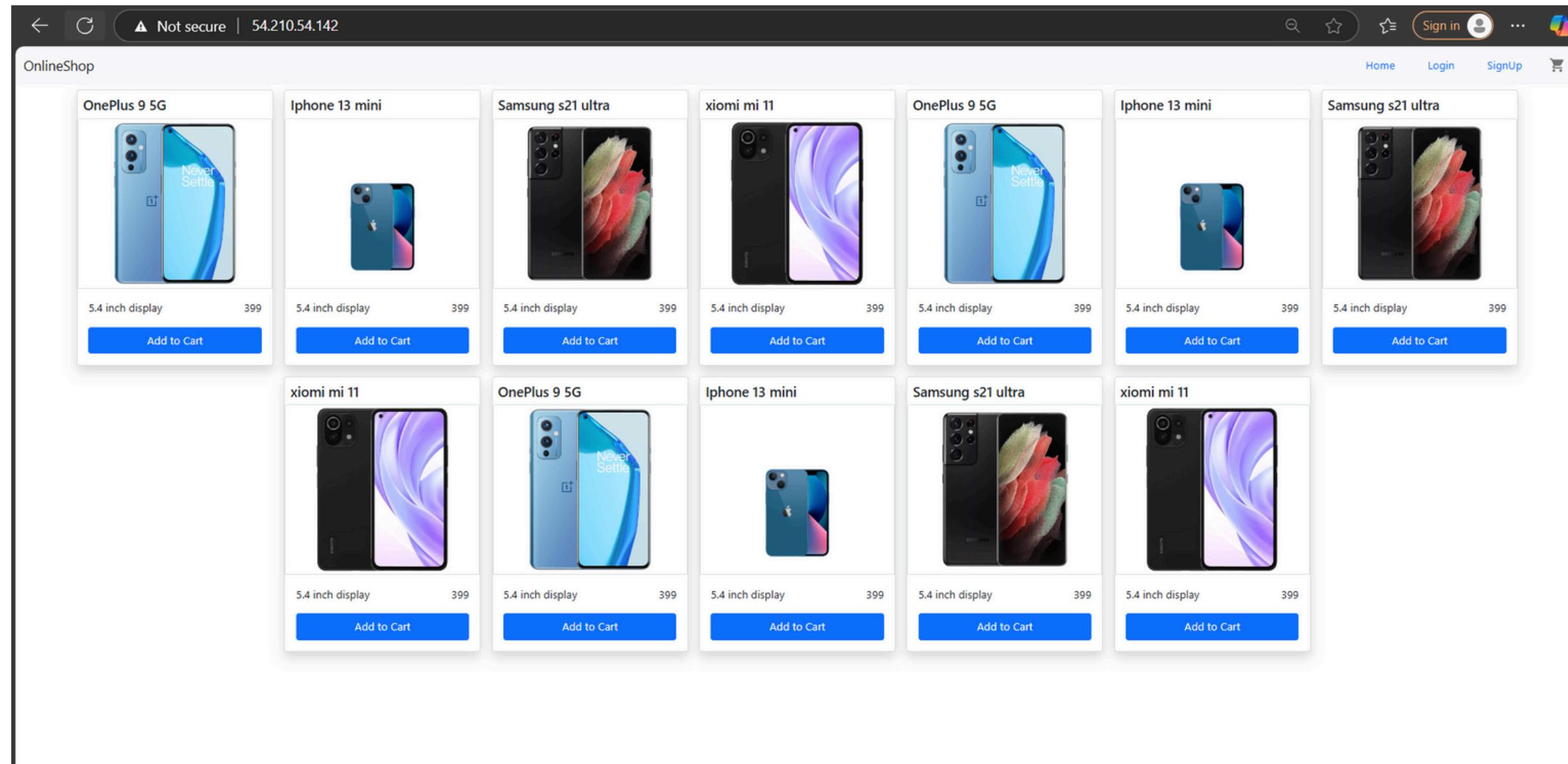
sudo docker pull vijay192001/dev:latest

Run The Docker Image in Instances 2

```
[ec2-user@ip-172-31-95-82 ~]$ sudo docker run -d -p 80:80 vijay192001/dev:latest
665d7698c72879e39561b6bea6170f2097907bf6f9d0aa0fdb062ace29182f63
[ec2-user@ip-172-31-95-82 ~]$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                         NAMES
665d7698c728        vijay192001/dev:latest   "/docker-entrypoint..."   14 seconds ago    Up 13 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp   stoic_ride
```

sudo docker run -d -p 80:80 vijay192001/dev:latest

Docker Image Output In Instances 2



Monitoring

Install prometheus node exporter alert manager
Then set up prometheus.yml file , alertmanager.yml
file and alertrules.yml file in etc folder

Install Prometheus and grafana For Monitoring

```
[ec2-user@ip-172-31-80-54 prometheus-2.40.0.linux-amd64]$ cd /home/ec2-user
```

First **cd /home/ec2-user**

```
[ec2-user@ip-172-31-80-54 prometheus-2.40.0.linux-amd64]$ wget https://github.com/prometheus/prometheus/releases/download/v2.40.0/prometheus-2.40.0.linux-amd64.tar.gz
```

Then **wget https://github.com/prometheus/prometheus/releases/download/v2.40.0/prometheus-2.40.0.linux-amd64.tar.gz**

```
[ec2-user@ip-172-31-80-54 prometheus-2.40.0.linux-amd64]$ tar -xvf prometheus-2.40.0.linux-amd64.tar.gz
```

Then **tar -xvf prometheus-2.40.0.linux-amd64.tar.gz**

```
[ec2-user@ip-172-31-80-54 prometheus-2.40.0.linux-amd64]$ cd prometheus-2.40.0.linux-amd64
```

Then **cd prometheus-2.40.0.linux-amd64**

```
[ec2-user@ip-172-31-80-54 prometheus-2.40.0.linux-amd64]$ ./prometheus --config.file=prometheus.yml
```

Then **./prometheus --config.file=prometheus.yml**

```
alerting:  
  alertmanagers:  
    - static_configs:  
      - targets:  
        - '54.208.150.44:9093' # Address of your Alertmanager  
  
  # Load rules once and periodically evaluate them according to the global 'evaluation_interval'.  
rule_files:  
  - "alertrules.yml" # Ensure this file exists and has the alert rules  
  
# Global configuration  
global:  
  scrape_interval: 15s # Set the scrape interval globally (15 seconds)  
  evaluation_interval: 15s # Set evaluation interval for alert rules  
  
# Scrape configuration  
scrape_configs:  
  
  # Scrape Prometheus itself  
  - job_name: "prometheus"  
    static_configs:  
      - targets: ['54.208.150.44:9090'] # Scrape Prometheus metrics  
  
  # Scrape Node Exporter  
  - job_name: 'node_exporter'  
    static_configs:  
      - targets: ['54.208.150.44:9100'] # Target for Node Exporter (make sure port 9100 is correct)  
  
  # Scrape your custom target (vijay192001/dev)  
  - job_name: 'vijay192001/dev'  
    static_configs:  
      - targets: ['54.208.150.44:80'] # Target for your custom service (port 80)  
      metrics_path: /metrics # Ensure your service exposes Prometheus-compatible metrics at this path
```

0 △ 0

Prometheus.yml file

- **cd ..**
- **cd prometheus**
- **ls**
- **cd /etc**
- **cd prometheus**
- **ls**
- **sudo vi prometheus.yml**
- **sudo vi alertrules.yml**
- **sudo vi alertmanager.yml**

A screenshot of a terminal window titled "alertmanager configuration". The window shows two files: "alertrules.yml" and "alertmanager.yml".

The "alertrules.yml" file contains the following configuration:

```
groups:
- name: example-alerts
  rules:
    - alert: InstanceDown
      expr: up == 0
      for: 5m
      labels:
        severity: critical
      annotations:
        summary: "Instance {{ $labels.instance }} is down"
```

The "alertmanager.yml" file contains the following configuration:

```
# Global configuration
global:
  smtp_smarthost: 'smtp.example.com:587'
  smtp_from: 'vijayakumar2652001@gmail.com'
  smtp_auth_username: 'vijayakumar2652001@gmail.com'
  smtp_auth_password: 'fexz abiu ofwz ccrx'

# Define the route for alert delivery
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'web.hook' # Default receiver is web.hook

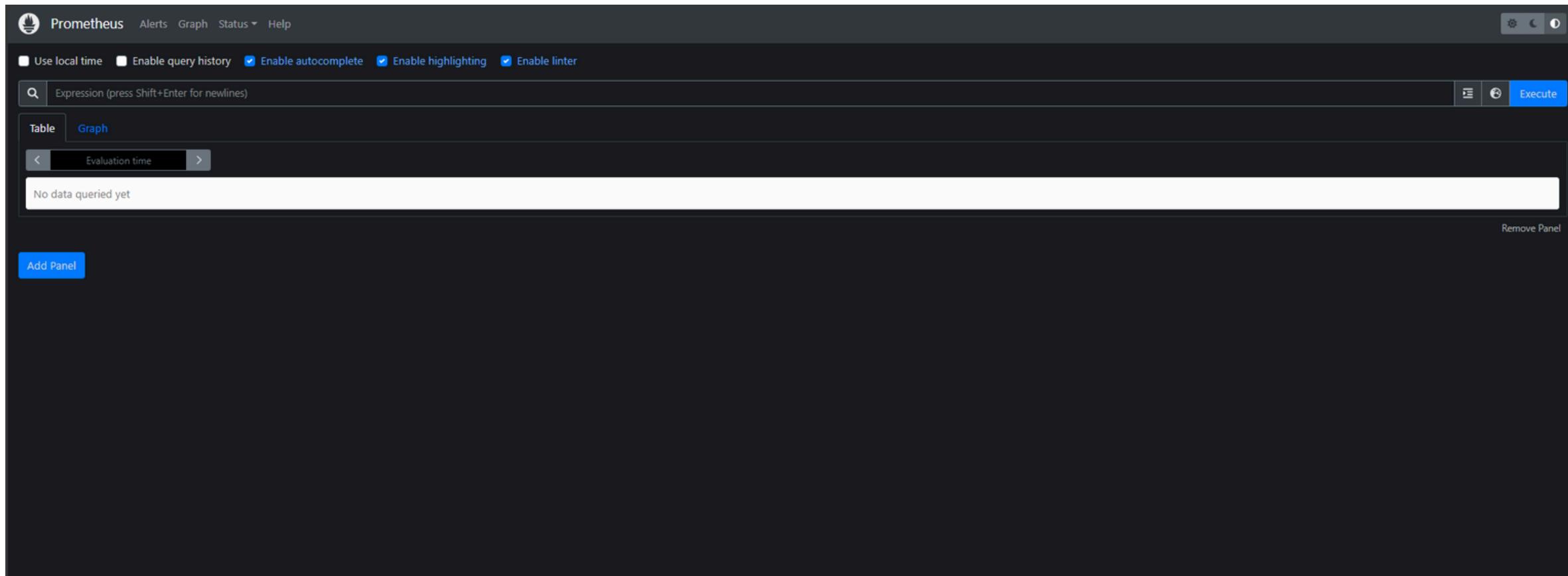
# Define the receivers
receivers:
  - name: 'web.hook'
    webhook_configs:
      - url: 'http://127.0.0.1:5001/'
  - name: 'email'
    email_configs:
      - to: 'vijayakumar2652001@gmail.com'
        send_resolved: true

# Inhibit rule for severity levels
inhibit_rules:
  - source_match:
      severity: 'critical'
    target_match:
      severity: 'warning'
      equal: ['alertname', 'dev', 'instance']
```

sudo vi alertrules.yml

sudo vi alertmanager.yml

Instaled Prometheus



Checking Targets

The screenshot shows the Prometheus Targets page with three sections:

- node_exporter (1/1 up)**: Shows one endpoint at `http://18.207.222.74:9100/metrics` in UP state, last scraped 5.511s ago, with a scrape duration of 11.660ms.
- prometheus (1/1 up)**: Shows one endpoint at `http://18.207.222.74:9090/metrics` in UP state, last scraped 7.323s ago, with a scrape duration of 3.686ms.
- vijay192001/dev (0/1 up)**: Shows one endpoint at `http://18.207.222.74:metrics` in DOWN state, last scraped 6.554s ago, with a scrape duration of 1.403ms. The error message indicates a connection refused: "Get \"http://18.207.222.74:80/metrics\": dial tcp 18.207.222.74:80: connect: connection refused".

Installing Node Exporter

Install Node exporter and cd into that node expoter

```
[ec2-user@ip-172-31-92-56 ~]$ wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
tar -xvzf node_exporter-1.3.1.linux-amd64.tar.gz
--2025-04-06 11:22:37-- https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24a9d0383?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduct
ion%2F20250406%2Fsus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250406T112237Z&X-Amz-Expires=300&X-Amz-Signature=08ae5db95d4c3666661f788d7bfe56796b1455a8dbffabc49000704a2ade7abc&X-Amz-SignedHeaders=host&re
sponse-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-04-06 11:22:37-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24a9d0383?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=rel
easeassetproduction%2F20250406%2Fsus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250406T112237Z&X-Amz-Expires=300&X-Amz-Signature=08ae5db95d4c3666661f788d7bfe56796b1455a8dbffabc49000704a2ade7abc&X-Amz-Signed
Headers=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%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: 9033415 (8.6M) [application/octet-stream]
Saving to: 'node_exporter-1.3.1.linux-amd64.tar.gz.1'

node_exporter-1.3.1.linux-amd64.tar.gz.1      100%[=====] 8.61M --KB/s in 0.1s

2025-04-06 11:22:37 (86.4 MB/s) - 'node_exporter-1.3.1.linux-amd64.tar.gz.1' saved [9033415/9033415]

node_exporter-1.3.1.linux-amd64/
node_exporter-1.3.1.linux-amd64/LICENSE
node_exporter-1.3.1.linux-amd64/NOTICE
node_exporter-1.3.1.linux-amd64/node_exporter
[ec2-user@ip-172-31-92-56 ~]$ ls
NetrwTreeListing          alertmanager.yml           installingprerequisites    prometheus-2.40.0.linux-amd64
alert_rules.yml            grafana-9.2.6-1.x86_64.rpm   node_exporter-1.3.1.linux-amd64  prometheus-2.40.0.linux-amd64.tar.gz
alertmanager-0.24.0.linux-amd64  grafana-enterprise-11.6.0.linux-amd64.tar.gz  node_exporter-1.3.1.linux-amd64.tar.gz  prometheus-2.40.0.linux-amd64.tar.gz.1
alertmanager-0.24.0.linux-amd64.tar.gz  grafana-v11.6.0          node_exporter-1.3.1.linux-amd64.tar.gz.1  prometheus.yml
[ec2-user@ip-172-31-92-56 ~]$ cd node_exporter-1.3.1.linux-amd64
[ec2-user@ip-172-31-92-56 node_exporter-1.3.1.linux-amd64]$ ls
LICENSE NOTICE node_exporter prometheus
[ec2-user@ip-172-31-92-56 node_exporter-1.3.1.linux-amd64]$ 
```

- **wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz**
- **tar -xvzf node_exporter-1.3.1.linux-amd64.tar.gz**
- **cd node_exporter-1.3.1.linux-amd64**
- **./node_exporter &**

Node exporter port number 9100

Configure Prometheus to Scrape Node Exporter Metrics

```
# Scrape Node Exporter
- job_name: 'node'
  static_configs:
    - targets: ['54.205.119.245:9100'] # Target for Node Exporter
```

After that in prometheus.yml Add the following scrape_configs block to scrape Node Exporter metrics:

```
# Scrape Node Exporter
- job_name: 'node'
  static_configs:
    - targets: ['54.205.119.245:9100'] # Target for Node Exporter
```

54.205.119.245:9100



Installing Alertmanager

Install alertmanager and cd into alertmanager

```
[ec2-user@ip-172-31-92-56 ~]$ wget https://github.com/prometheus/alertmanager/releases/download/v0.24.0/alertmanager-0.24.0.linux-amd64.tar.gz
tar -xvzf alertmanager-0.24.0.linux-amd64.tar.gz
cd alertmanager-0.24.0.linux-amd64
--2025-04-06 11:48:24-- https://github.com/prometheus/alertmanager/releases/download/v0.24.0/alertmanager-0.24.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/11452538/d3143d23-5eab-4248-ae7b-2d04eb2e6084?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250406%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250406T114824Z&X-Amz-Expires=300&X-Amz-Signature=19908cc5acf4ad1942a64487052dc33a81941fd21889011474d78b62c5a1d9ee&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dalertmanager-0.24.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-04-06 11:48:24-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/11452538/d3143d23-5eab-4248-ae7b-2d04eb2e6084?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250406%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250406T114824Z&X-Amz-Expires=300&X-Amz-Signature=19908cc5acf4ad1942a64487052dc33a81941fd21889011474d78b62c5a1d9ee&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dalertmanager-0.24.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25880024 (25M) [application/octet-stream]
Saving to: 'alertmanager-0.24.0.linux-amd64.tar.gz.1'

alertmanager-0.24.0.linux-amd64.tar.gz.1      100%[=====] 24.68M  89.9MB/s   in 0.3s

2025-04-06 11:48:24 (89.9 MB/s) - 'alertmanager-0.24.0.linux-amd64.tar.gz.1' saved [25880024/25880024]

alertmanager-0.24.0.linux-amd64/
alertmanager-0.24.0.linux-amd64/alertmanager.yml
alertmanager-0.24.0.linux-amd64/LICENSE
alertmanager-0.24.0.linux-amd64/NOTICE
```

wget https://github.com/prometheus/alertmanager/releases/download/v0.24.0/alertmanager-0.24.0.linux-amd64.tar.gz
tar -xvzf alertmanager-0.24.0.linux-amd64.tar.gz
cd alertmanager-0.24.0.linux-amd64

```
[ec2-user@ip-172-31-92-56 alertmanager-0.24.0.linux-amd64]$ ./alertmanager --config.file=alertmanager.yml
ts=2025-04-06T11:50:06.098Z caller=main.go:231 level=info msg="Starting Alertmanager" version="(version=0.24.0, branch=HEAD, revision=f484b17fa3c583ed1b2c8bbcec20ba1db2aa5f11)"
ts=2025-04-06T11:50:06.098Z caller=main.go:232 level=info build_context="(go=go1.17.8, user=root@265f14f5c6fc, date=20220325-09:31:33)"
ts=2025-04-06T11:50:06.102Z caller=cluster.go:185 level=info component=cluster msg="setting advertise address explicitly" addr=172.31.92.56 port=9094
ts=2025-04-06T11:50:06.119Z caller=cluster.go:680 level=info component=cluster msg="Waiting for gossip to settle..." interval=2s
ts=2025-04-06T11:50:06.139Z caller=coordinator.go:113 level=info component=configuration msg="Loading configuration file" file=alertmanager.yml
ts=2025-04-06T11:50:06.139Z caller=coordinator.go:126 level=info component=configuration msg="Completed loading of configuration file" file=alertmanager.yml
ts=2025-04-06T11:50:06.144Z caller=main.go:535 level=info msg="Listening address=:9093"
ts=2025-04-06T11:50:06.144Z caller=tls_config.go:195 level=info msg="TLS is disabled." http2=false
ts=2025-04-06T11:50:08.120Z caller=cluster.go:705 level=info component=cluster msg="gossip not settled" polls=0 before=0 now=1 elapsed=2.001010184s
[]
```

./alertmanager --config.file=alertmanager.yml

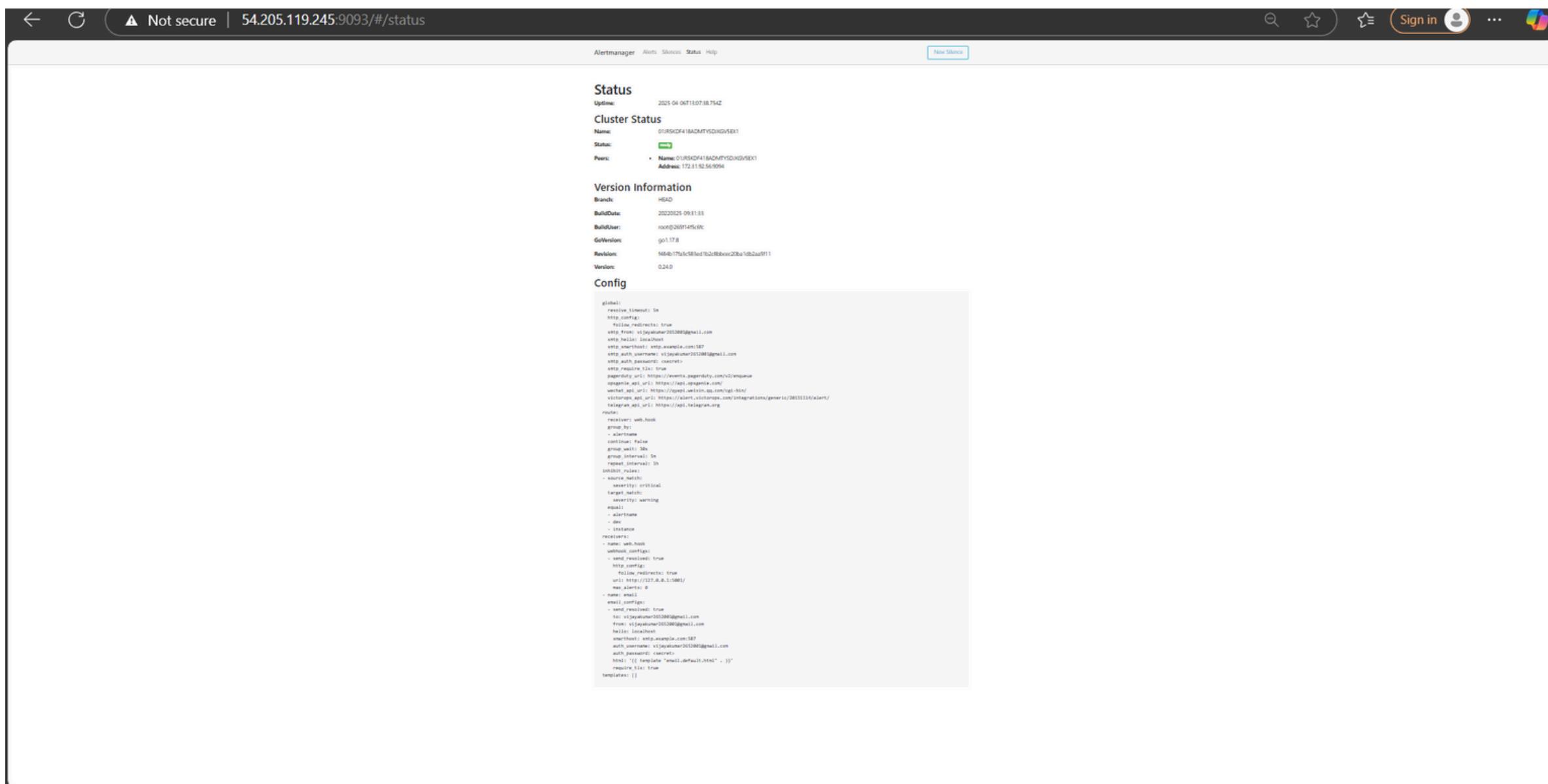
Alertmanager port number is 9093

http://localhost:9093

The screenshot shows the Alertmanager interface running locally at port 9093. The top navigation bar includes links for 'Alertmanager', 'Alerts', 'Silences', 'Status', and 'Help', along with a 'New Silence' button. The main area features a search bar with 'Filter' and 'Group' tabs, and a 'Custom matcher, e.g. env="production"' input field. Below this is a summary section with a '+ Expand all groups' link and a 'Not grouped' section containing '1 alert'. The alert details are as follows:

- Timestamp: 2025-04-07T07:38:24.210Z
- Severity: Info
- Source: Source
- Silence: Silence
- Labels:
 - alername="InstanceDown"
 - instance="18.207.222.74:80"
 - job="vijay192001/dev"
 - severity="critical"

Checking Status Alertmanager



Configure Alertmanager to Send Notifications

Open the alertmanager.yml configuration file.

The screenshot shows the VS Code interface with the following details:

- Title Bar:** Welcome
- Top Bar Buttons:** Clone Git Repository..., Connect to..., Get Started with WSL (Updated)
- Menu Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS
- Terminal Content:** A snippet of an alertmanager configuration file (`alertmanager.yml`). The code includes sections for global settings, route definitions, and receiver configurations.
- Right Sidebar:** Shows a list of available terminals, including PowerShell and SSH sessions.
- Bottom Status Bar:** "alertmanager.yml" 19L, 448B, 14,42, All

```
global:
  resolve_timeout: 5m

route:
  group_by: ['Docker']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 12h
  receiver: 'email_notifications'

receivers:
  - name: 'email_notifications'
    email_configs:
      - to: 'vijayakumar2652001@gmail.com'
        from: 'vijayakumar2652001@gmail.com'
        smarthost: 'smtp.gmail.com:587'
        auth_username: 'vijayakumar2652001@gmail.com'
        auth_password: 'fexz abiu ofwz ccrx'

~
```

- In this file set up email app password
 - Save the updated alertmanager.yml file
 - In route set Docker details

```
[ec2-user@ip-172-31-92-56 alertmanager-0.24.0.linux-amd64]$ vi alertmanager.yml
[ec2-user@ip-172-31-92-56 alertmanager-0.24.0.linux-amd64]$ vi alertmanager.yml
[ec2-user@ip-172-31-92-56 alertmanager-0.24.0.linux-amd64]$ ./alertmanager --config.file=alertmanager.yml
ts=2025-04-06T12:04:34.381Z caller=main.go:231 level=info msg="Starting Alertmanager" version="(version=0.24.0, branch=HEAD, revision=f484b17fa3c583ed1b2c8bbcec20ba1db2aa5f11)"
ts=2025-04-06T12:04:34.381Z caller=main.go:232 level=info build_context="(go=go1.17.8, user=root@265f14f5c6fc, date=20220325-09:31:33)"
ts=2025-04-06T12:04:34.385Z caller=cluster.go:185 level=info component=cluster msg="setting advertise address explicitly" addr=172.31.92.56 port=9094
ts=2025-04-06T12:04:34.398Z caller=cluster.go:680 level=info component=cluster msg="Waiting for gossip to settle..." interval=2s
ts=2025-04-06T12:04:34.421Z caller=coordinator.go:113 level=info component=configuration msg="Loading configuration file" file=alertmanager.yml
ts=2025-04-06T12:04:34.421Z caller=coordinator.go:126 level=info component=configuration msg="Completed loading of configuration file" file=alertmanager.yml
ts=2025-04-06T12:04:34.425Z caller=main.go:431 level=info component=configuration msg="skipping creation of receiver not referenced by any route" receiver=email
ts=2025-04-06T12:04:34.426Z caller=main.go:535 level=info msg=Listening address=:9093
ts=2025-04-06T12:04:34.426Z caller=tls_config.go:195 level=info msg="TLS is disabled." http2=false
ts=2025-04-06T12:04:36.398Z caller=cluster.go:705 level=info component=cluster msg="gossip not settled" polls=0 before=0 now=1 elapsed=2.000330263s
ts=2025-04-06T12:04:44.402Z caller=cluster.go:697 level=info component=cluster msg="gossip settled; proceeding" elapsed=10.003625489s
[]
```

run this command `./alertmanager --config.file=alertmanager.yml`

Configure Prometheus to Send Alerts to Alertmanager

```
groups:
- name: example-alerts
  rules:
    - alert: InstanceDown
      expr: up == 0
      for: 5m
      labels:
        severity: critical
      annotations:
        summary: "Instance {{ $labels.instance }} is down"
```

Create a new file called `alert.rules` in the same directory as `prometheus.yml` with the following content to trigger an alert if a node becomes unreachable

prometheus.yml

```
# Global configuration
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:
        - '54.205.119.245:9093'

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

# Scrape configuration
scrape_configs:
  # Scrape Prometheus itself
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"] # Scrape Prometheus metrics

  # Scrape Node Exporter
  - job_name: 'node'
    static_configs:
      - targets: ['54.205.119.245:9100'] # Target for Node Exporter
  □

  ~
  ~
  ~
"prometheus.yml" 32L, 907B
```

- After that Open the prometheus.yml file and add an alerting section
- Then add on alert rule for detecting when the application goes down

The screenshot shows the Prometheus Alerts interface. The top navigation bar includes links for Prometheus, Alerts, Graph, Status, and Help. The main content area displays a list of alerts under the heading '/etc/prometheus/alertrules.yml > example-alerts'. A yellow header bar highlights 'pending (5)'. Below this, a section titled 'InstanceDown (5 active)' contains the configuration for the alert:

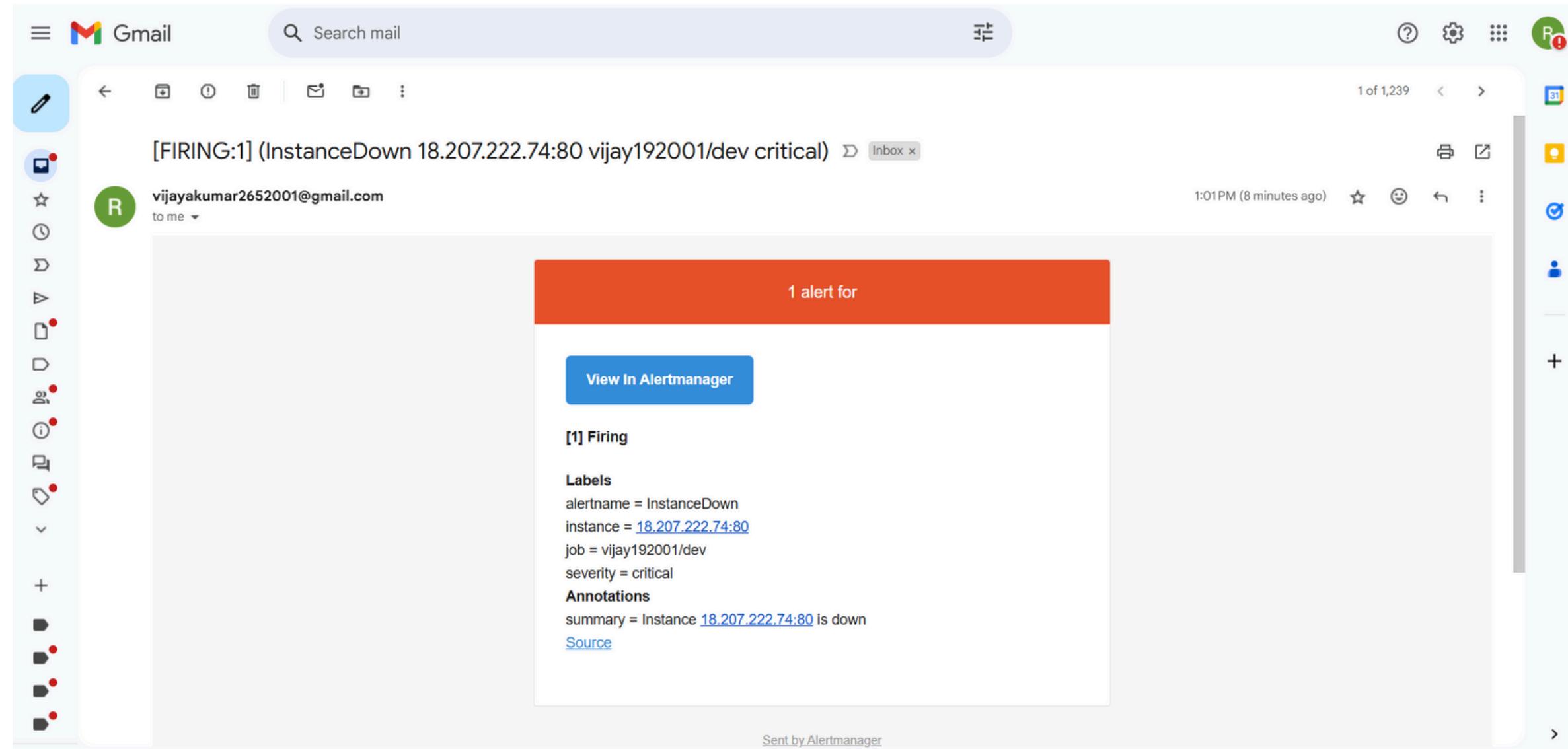
```
name: InstanceDown
expr: up == 0
for: 5m
labels:
  severity: critical
annotations:
  summary: Instance {{ $labels.instance }} is down
```

Below the configuration, a table lists five pending alerts with the following details:

Labels	State	Active Since	Value
alertname=InstanceDown instance=54.208.150.44:9090 job=vijay192001/dev severity=critical	PENDING	2025-04-07T10:38:24.210916183Z	0
alertname=InstanceDown instance=18.207.222.74:9090 job=vijay192001/dev severity=critical	PENDING	2025-04-07T10:38:09.210916183Z	0
alertname=InstanceDown instance=18.207.222.74:9090 job=prometheus severity=critical	PENDING	2025-04-07T10:38:09.210916183Z	0
alertname=InstanceDown instance=18.207.222.74:9100 job=node_exporter severity=critical	PENDING	2025-04-07T10:38:09.210916183Z	0
alertname=InstanceDown instance=54.208.150.44:9100 job=node_exporter severity=critical	PENDING	2025-04-07T10:38:24.210916183Z	0

- After Restarting Prometheus
- Then instance will fire and email will generated

Finally, you will receive an email notification if the Docker container stops



Finally, you will receive an email notification if the Docker container stops

Alertmanager port number is 9093

Node Exporter Port number 9100

Prometheus Port number 9090

1. Jenkins (login page, configuration settings, execute step commands). - done
2. AWS (EC2 console, SG configurations) - done
3. Docker Hub repo with image tags - done
4. Deployed site page -done
5. Monitoring health check status - done (along with the mail trigger)