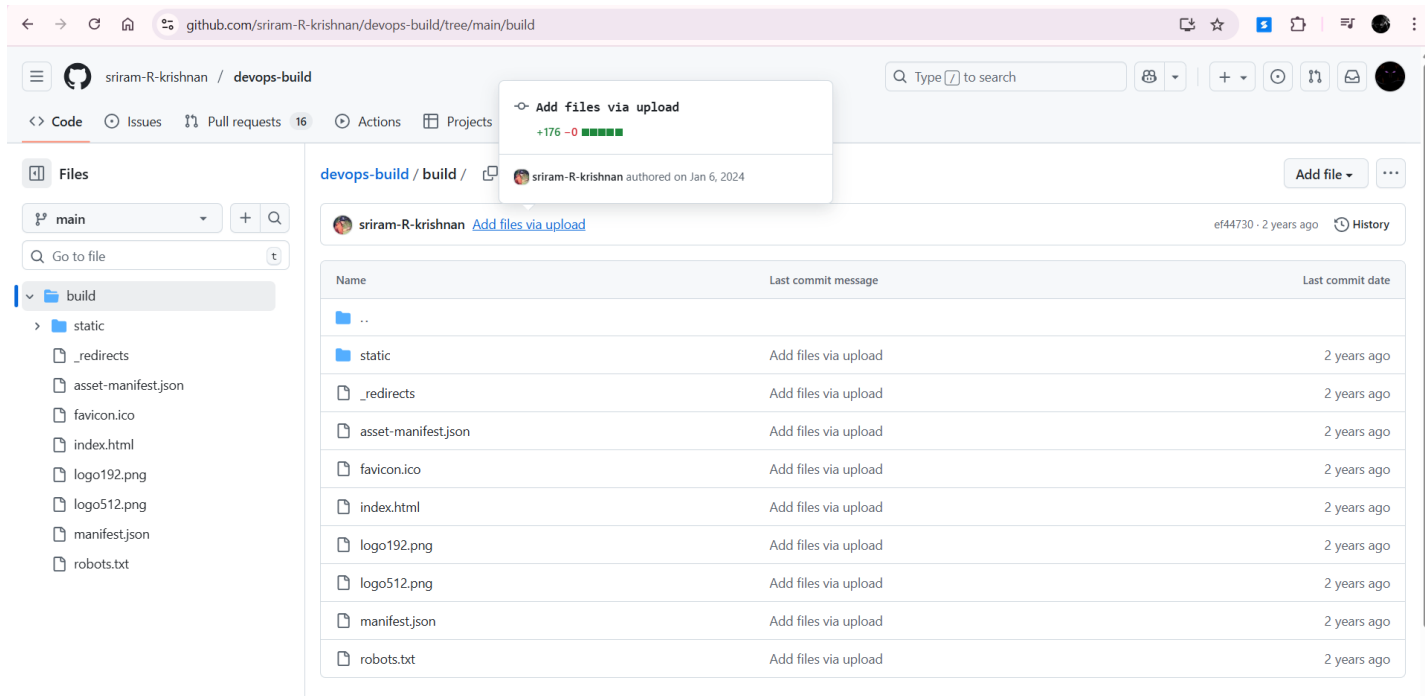
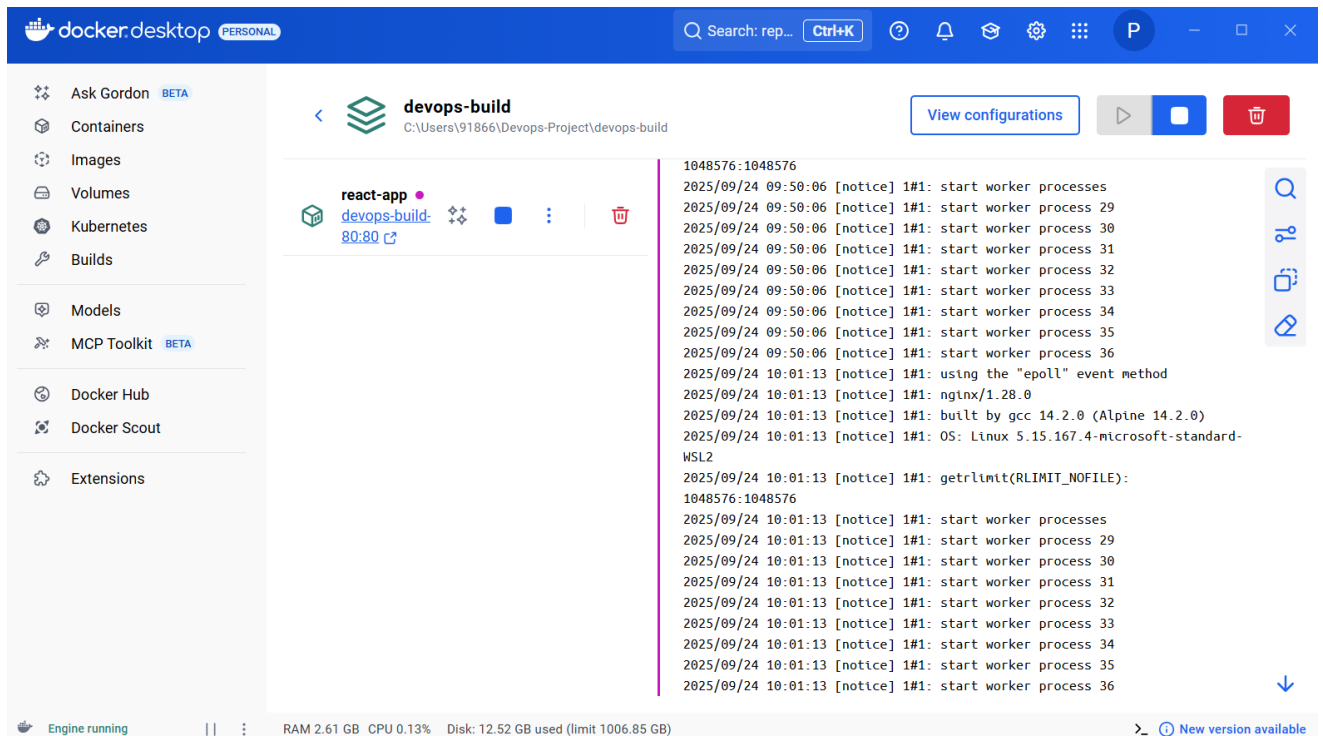


Application Deployment

Step 1: Clone the repo and inspect the project.



Step 2: Dockerize the react-app and run the app container locally to confirm.



Step 3: Create branches and push images to Docker Hub.

pranit162
Docker Personal

Repositories

Collaborations

Settings

Default privacy

Notifications

Billing

Usage

Repositories

All repositories within the pranit162 namespace.

All content

Create a repository

Name	Last Pushed	Contains	Visibility	Scout
pranit162/devops-project-prod	about 4 hours ago	IMAGE	Private	Inactive
pranit162/devops-project-dev	about 4 hours ago	IMAGE	Public	Inactive

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0e32113c1210eea15&osUser=ubuntu&sshPort=22&addressFamily=ipv4

Search [Alt+S]

Asia Pacific (Mumbai)

Account ID: 9079-6992-9387

Pranit

```
ubuntu@ip-172-31-9-180:~$ docker login

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: WLNJ-BQXC
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
ubuntu@ip-172-31-9-180:~$ docker pull pranit162/devops-project-dev:3
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/images/create?fromImage=pranit162%2Fdevops-project-dev&tag=3": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-9-180:~$ sudo docker pull pranit162/devops-project-dev:3
3: Pulling from pranit162/devops-project-dev
Digest: sha256:5a09bd713534052421d1b20681c78df16b3d544ed83fe08401b226bbeb186659
Status: Image is up to date for pranit162/devops-project-dev:3
docker.io/pranit162/devops-project-dev:3
ubuntu@ip-172-31-9-180:~$ sudo docker run -d -p 80:80 --name react-app pranit162/devops-project-dev:3
2850d4a686ffe9ee78f2214f0f31302d14c3e0aa6f4dd31d16b851dce58eb8f6
ubuntu@ip-172-31-9-180:~$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
2850d4a686ff  pranit162/devops-project-dev:3      "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp react-app
ubuntu@ip-172-31-9-180:~$
```

i-0e32113c1210eea15 (jenkins-ec2)

Public IPs: 13.233.106.169 Private IPs: 172.31.9.180

Step 4: Jenkins configuration and creation of pipeline.



+ New Item

📄 Build History



✎ Add description

Build Queue ▾

No builds in the queue.

Build Executor Status 0/2 ▾

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 ?

REST API Jenkins 2.516.3



/ Manage Jenkins / Plugins



Plugins

🔍 github inte

📦 Install ▾ ↻

📄 Updates


📦 Available plugins

⚙ Installed plugins

⚙ Advanced settings

⌵ Download progress

Install	Name ↕	Released	Health
✓	Git Push 34.vd474e0fa7b_ec git Source Code Management This plugin allows to perform a git push as a post build step via Git Plugin	3 yr 7 mo ago	85
✓	Docker 1274.vc0203fd2e74 Cloud Providers Cluster Management docker This plugin integrates Jenkins with Docker	6 mo 20 days ago	100
✓	Docker Pipeline 621.va_73f881d9232 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	3 mo 27 days ago	96
✓	GitHub Pipeline for Blue Ocean 1.27.21 External Site/Tool Integrations User Interface BlueOcean GitHub organization pipeline creator	3 mo 9 days ago	97
✓	GitHub Integration 0.7.2 emailx Build Triggers GitHub Integration Plugin for Jenkins	8 mo 12 days ago	87

**Jenkins**

react-app-pipeline / Configuration

13.233.106.169:8080

Search

Settings

Menu

User

Configure

General

Triggers

Pipeline

Advanced

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Wanted162/devops-project.git

Credentials ?

Wanted162/*****

+ Add

Advanced

Add Repository


Branches to build ?

Branch Specifier (blank for 'any') ?

*/dev

Save

Apply

**Jenkins**

react-app-pipeline / Configuration

REST API

Jenkins 2.516.3

Search

Settings

Menu

User

Configure

General

Triggers

Pipeline

Advanced

(Auto)

Additional Behaviours

Add

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?


Pipeline Syntax

Advanced

Advanced

Save

Apply

**Jenkins**

react-app-pipeline

Status

Changes

Build Now

Configure

Delete Pipeline

Favorite

Open Blue Ocean

...

react-app-pipeline

Permalinks

- Last build (#3), 5 min 35 sec ago
- Last stable build (#3), 5 min 35 sec ago
- Last successful build (#3), 5 min 35 sec ago
- Last failed build (#2), 14 min ago
- Last unsuccessful build (#2), 14 min ago
- Last completed build (#3), 5 min 35 sec ago

Add description

Step 5:

A) Launch EC2 and deploy Docker images

```
sudo docker pull pranit162/devops-project-dev:3
sudo docker run -d -p 80:80 --name react-app pranit162/devops-project-dev:3
```

B) Create monitoring Docker Network

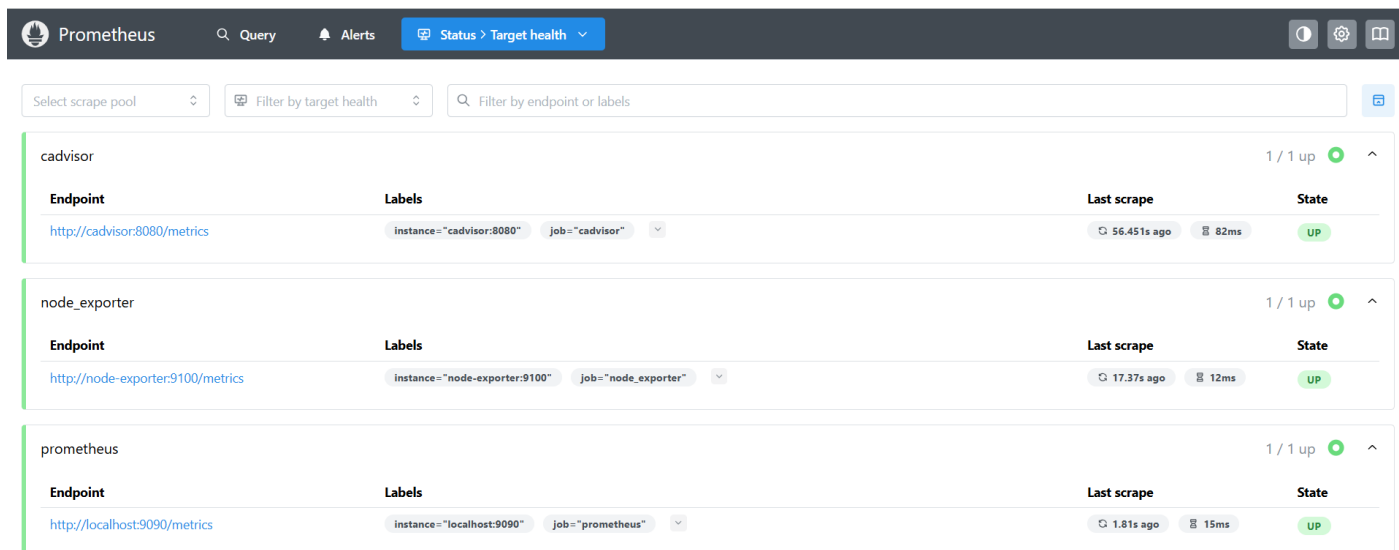
```
sudo docker network create monitoring
```

C) Run Prometheus, Node Exporter as Docker Container

```
sudo docker run -d --name prometheus --network monitoring -p 9090:9090 -v /opt/prometheus/pro
```

```
sudo docker run -d --name node-exporter --network monitoring -p 9100:9100 prom/node-exporter
```

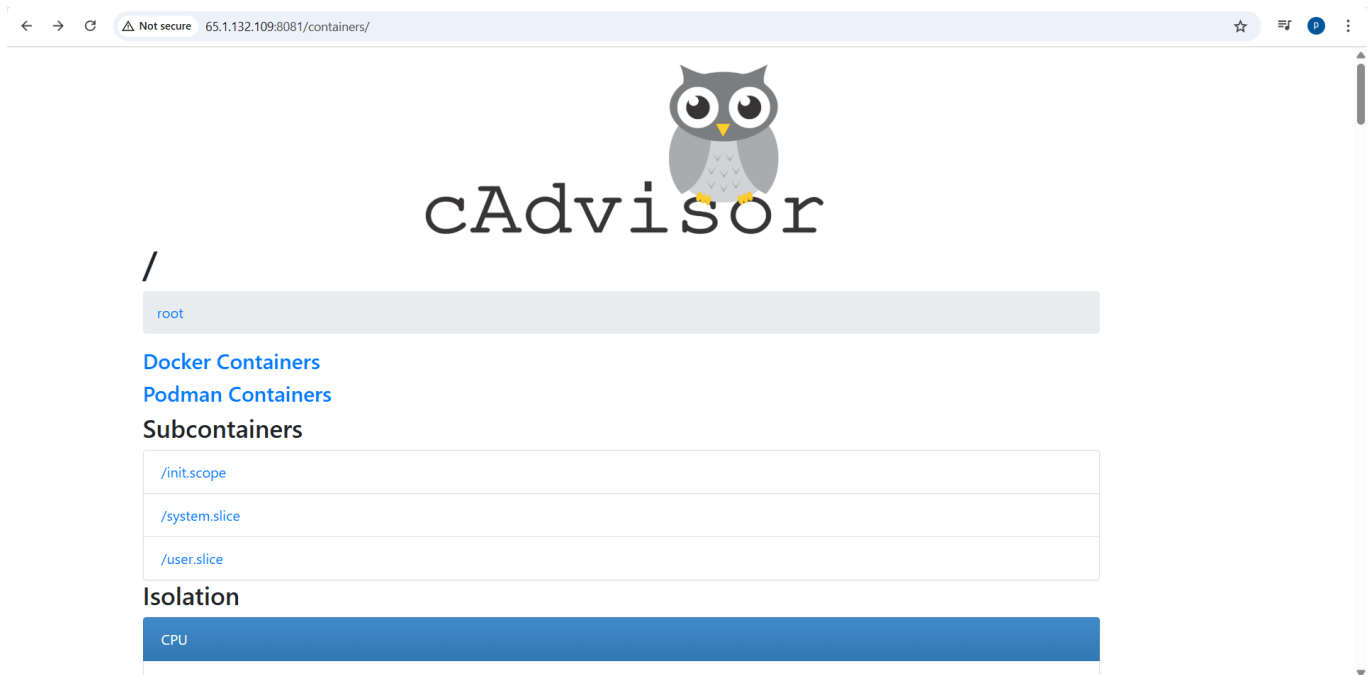
D) Update prometheus.yml to scrape Node Exporter and restart Prometheus



The screenshot shows the Prometheus web interface with the following details:

- Header:** Prometheus logo, Query, Alerts, and Status > Target health dropdown.
- Filters:** Select scrape pool, Filter by target health, and Filter by endpoint or labels.
- Targets List:**
 - cadvisor:** 1 / 1 up. Endpoint: <http://cadvisor:8080/metrics>. Labels: instance="cadvisor:8080", job="cadvisor". Last scrape: 56.451s ago, 82ms. State: UP.
 - node_exporter:** 1 / 1 up. Endpoint: <http://node-exporter:9100/metrics>. Labels: instance="node-exporter:9100", job="node_exporter". Last scrape: 17.37s ago, 12ms. State: UP.
 - prometheus:** 1 / 1 up. Endpoint: <http://localhost:9090/metrics>. Labels: instance="localhost:9090", job="prometheus". Last scrape: 1.81s ago, 15ms. State: UP.

Step 6: Add cAdvisor as Docker Container

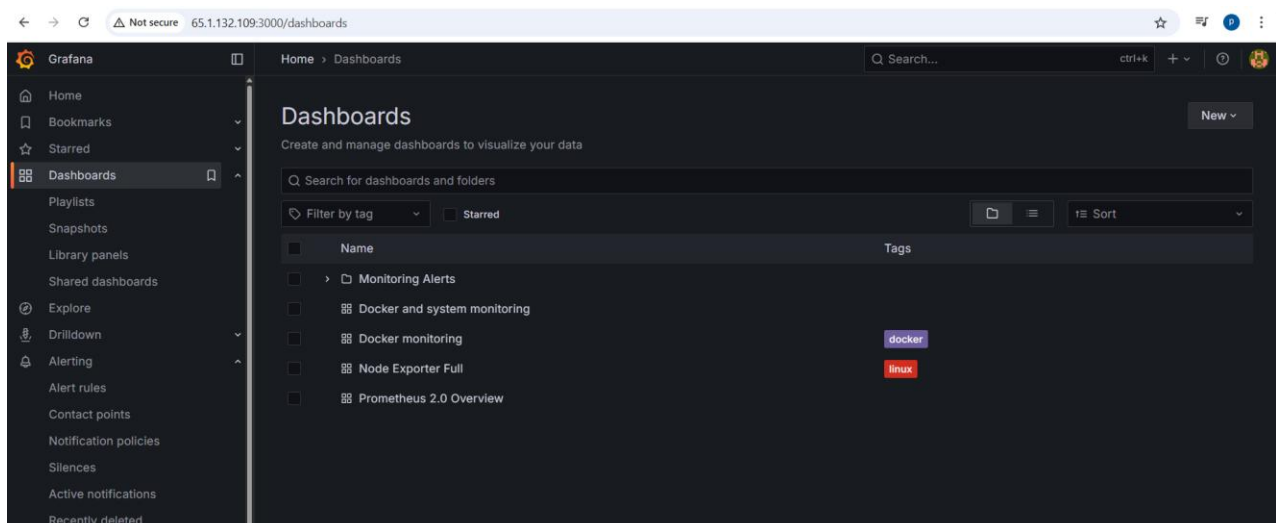


Step 7:

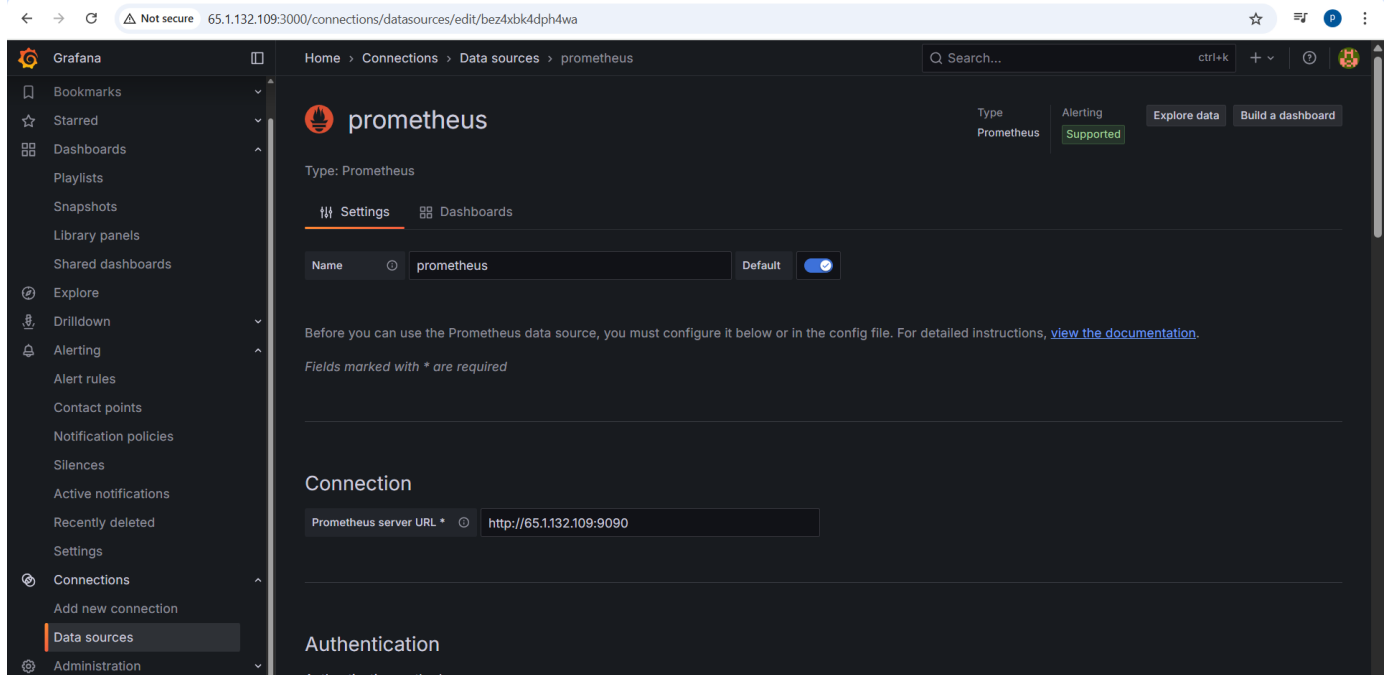
A) Install Grafana

```
sudo docker run -d --name=grafana --network monitoring -p 3000:3000 grafana/grafana
```

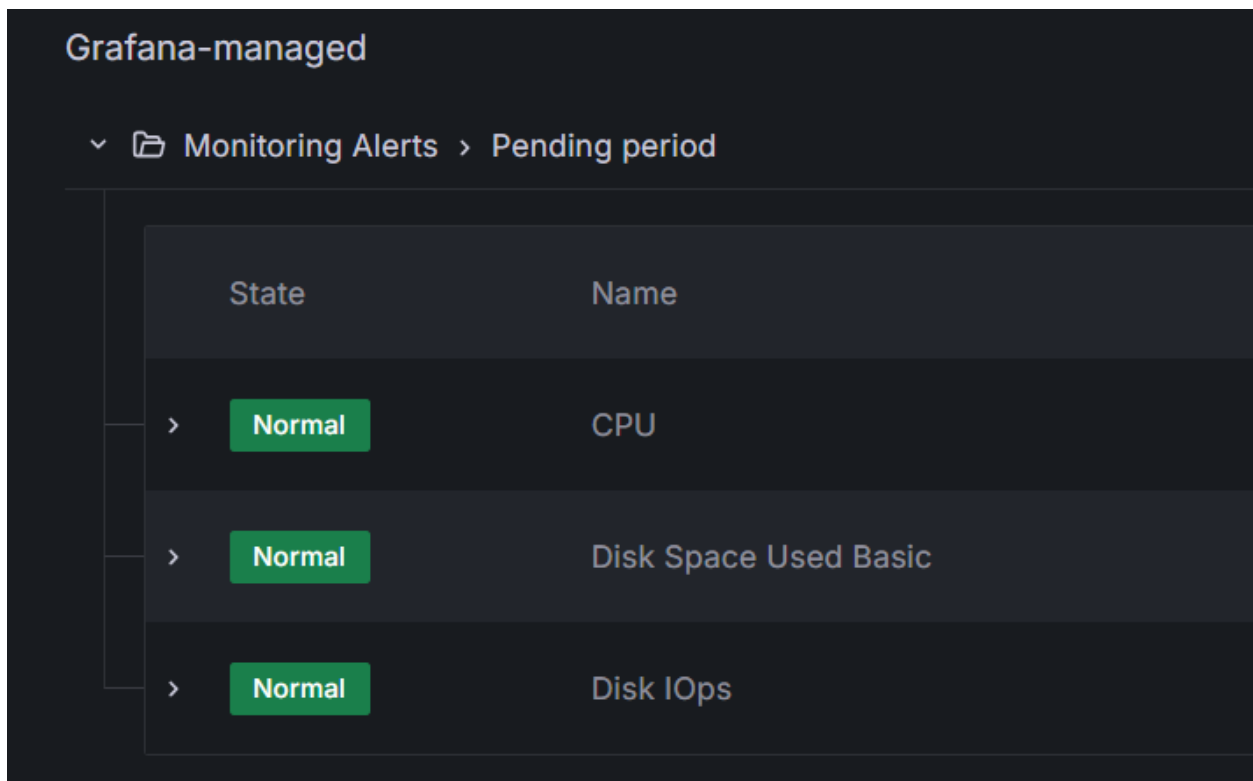
B) Import Grafana Dashboards



C) Configure Grafana data source to Prometheus



D) Set-up Alerts in Grafana



Step 8: Configure Grafana Notifications

```
sudo mkdir -p /opt/grafana-storage
sudo chown -R 472:472 /opt/grafana-storage
sudo docker rm -f grafana
sudo docker run -d --name=grafana -p 3000:3000 --network=monitoring \
  -v /opt/grafana-storage:/var/lib/grafana \
  -e GF_SMTP_ENABLED=true -e GF_SMTP_HOST=smtp.gmail.com:587 \
  -e GF_SMTP_USER=pisalpranit1@gmail.com \
  -e GF_SMTP_PASSWORD="(your-app-password)" \
  -e GF_SMTP_FROM_ADDRESS=pisalpranit1@gmail.com \
  -e GF_SMTP_FROM_NAME="Grafana Alerts" \
  grafana/grafana
```

Step 9: Test Alerting – Stress CPU which will send notification to your mail

```
sudo apt-get install stress -y
stress --cpu 4 --timeout 120
```

The screenshot shows a Gmail inbox with an email from 'Grafana Alerts <pisalpranit1@gmail.com>' titled '[FIRING:1] DatasourceNoData Monitoring Alerts (bez4xbk4dph4wa B,C,D,E,F,G,H,I CPU)'. The email was received at 4:47 PM (2 hours ago). Below the email, the Grafana web interface is displayed, showing the alert details for 'DatasourceNoData'.

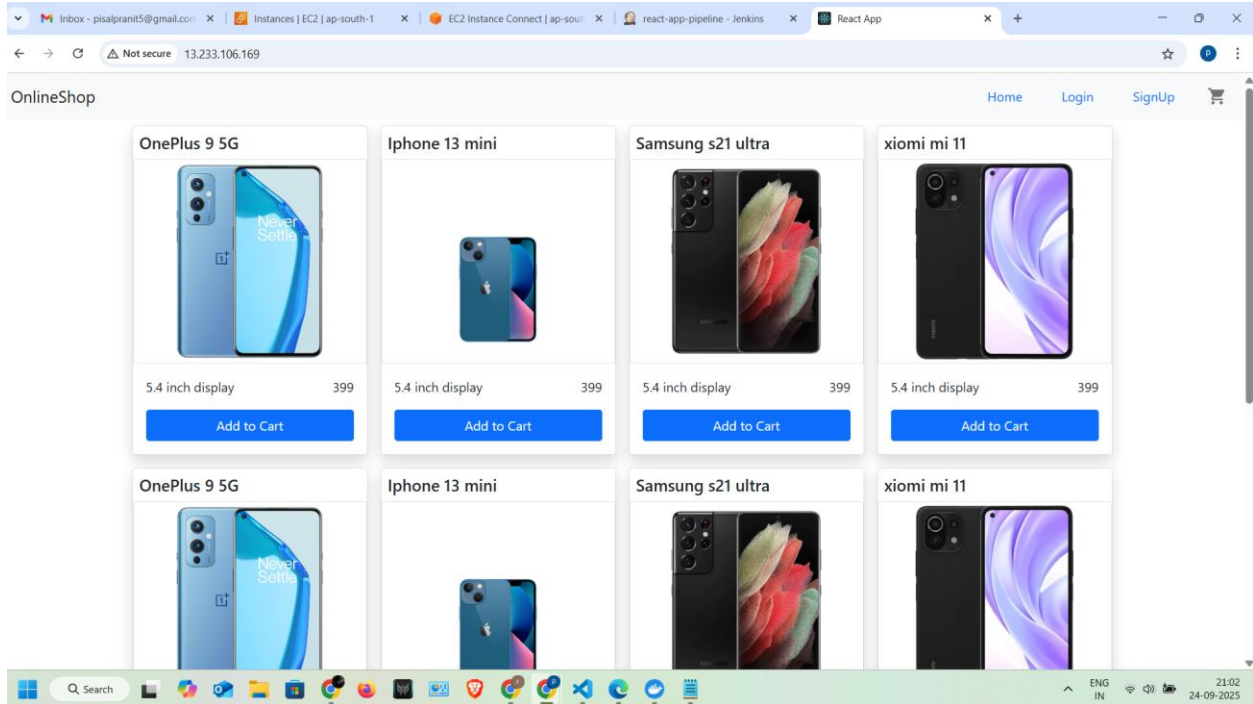
Grafana Alert Details:

- Alert Name:** DatasourceNoData
- Alert UID:** bez4xbk4dph4wa
- Grafana Folder:** Monitoring Alerts
- Ref ID:** B,C,D,E,F,G,H,I
- Rule Name:** CPU

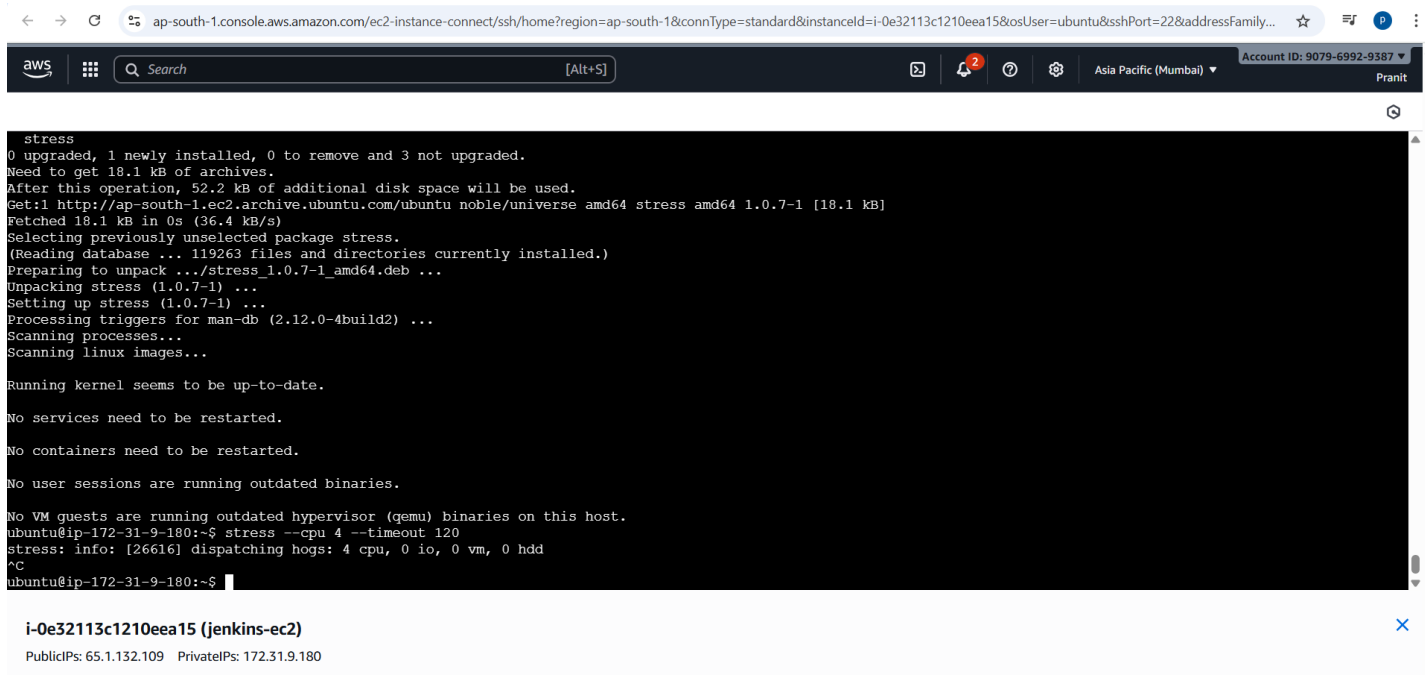
The alert is currently in a 'Firing' state, as indicated by the red 'Firing' label. The interface also shows buttons for 'View alert', 'Silence', 'View dashboard', and 'View panel'.

Step 10: Final status

- Deployed site page



- EC2 Console



- SG configs

AWS							
[Alt+S]							
Asia Pacific (Mumbai)							
Account ID: 9079-6992-9387							
Pranit							
EC2 > Security Groups > sg-036d75322e0dfd0cb - launch-wizard-4 > Edit inbound rules							
Security group rule ID	Type	Protocol	Port range	Source	Description - optional		
sgr-03182099e85253065	SSH	TCP	22	Custom	0.0.0.0		Delete
sgr-0bed791eeb66bd1f6	Custom TCP	TCP	8080	Custom	0.0.0.0	jenkins	Delete
sgr-0ffa39fcb91b84cdd	Custom TCP	TCP	8081	Custom	0.0.0.0	Advisor	Delete
sgr-0942b2ed305d2bfe0	HTTP	TCP	80	Custom	0.0.0.0		Delete
sgr-0646f8acd89b08828	Custom TCP	TCP	9100	Custom	0.0.0.0		Delete
sgr-0ac94c9c12c1dc455	Custom TCP	TCP	3000	Custom	0.0.0.0	Grafana	Delete
sgr-030a71bcc5eb57a80	Custom TCP	TCP	3001	Custom	0.0.0.0	uptime kuma	Delete
sgr-0b4ed15a22a8d6c8d	Custom TCP	TCP	9090	Custom	0.0.0.0	Prometheus	Delete

- Site Health Check Status

