Trendstore

GitHub Link : https://github.com/Vennilavan12/Trend.git

Overview

React application deployment to a production-ready state.

Brief description of the application, its purpose, and key features.

Example:

This project showcases deploying a React application from code to a production-ready, containerized application on Kubernetes (Amazon EKS) using a CI/CD pipeline with AWS CodeBuild, CodePipeline.

Architecture Overview

- Application: React
- Containerization: Docker
- Orchestration: Kubernetes (Amazon EKS)
- CI/CD Pipeline: Jenkins
- Cloud Provider: AWS
- Monitoring & Observability: Prometheus and Grafana dashboards

Setup Instructions

Prerequisites

- Docker
- kubectl
- AWS CLI
- eksctl
- Kubectl
- Helm

🚀 CI/CD Pipeline Explanation

Pipeline Stages

1. Pipeline Stages
2. Code Checkout
3. Push Docker Image to Docker Hub
4. Deploy to Kubernetes

Pipeline Flow

- On code push to the Git repository, Jenkins automatically triggers the build via webhooks.
- The Docker image is built and pushed to Docker Hub.
- Kubernetes manifests are automatically applied to deploy the application.

Project Screenshots and Proof of Implementation

This section includes screenshots that provide visual proof of the successful implementation and execution of the project. The screenshots demonstrate key stages and components of the deployment, including:

- ❖ Source code repositories and project structure
- ❖ Successful Docker image build and push to Docker Hub
- ❖ Jenkins CI/CD pipeline execution (checkout, build, push, and deploy stages)
- ❖ Amazon EKS cluster creation and running status
- ❖ Kubernetes deployments and services running successfully
- ❖ Application accessibility via the exposed application URL

- ❖ Prometheus metrics collection and Grafana dashboards displaying real-time cluster and application health

These screenshots collectively validate the end-to-end workflow, from source code to a production-ready, containerized application deployed on Kubernetes.

GIT URL : Project files

The full project setup and configuration artifacts are version-controlled and available in the Git repository.

URL :
https://github.com/udhayakumarethiraj1/enterprise/tree/a03c4c416cbb50ea36980bd640562255cff3e0fc/Project_2_Trend_store

Trendstore.git (Private): Contains all supporting configuration and infrastructure files, including the Dockerfile, Jenkinsfile, Kubernetes manifests (deployment.yaml, service.yaml, node-exporter-lb.yaml, kube-state-metrics-lb.yaml), .gitignore, and Terraform files.

URL : https://github.com/udhayakumarethiraj-git/Trendstore.git

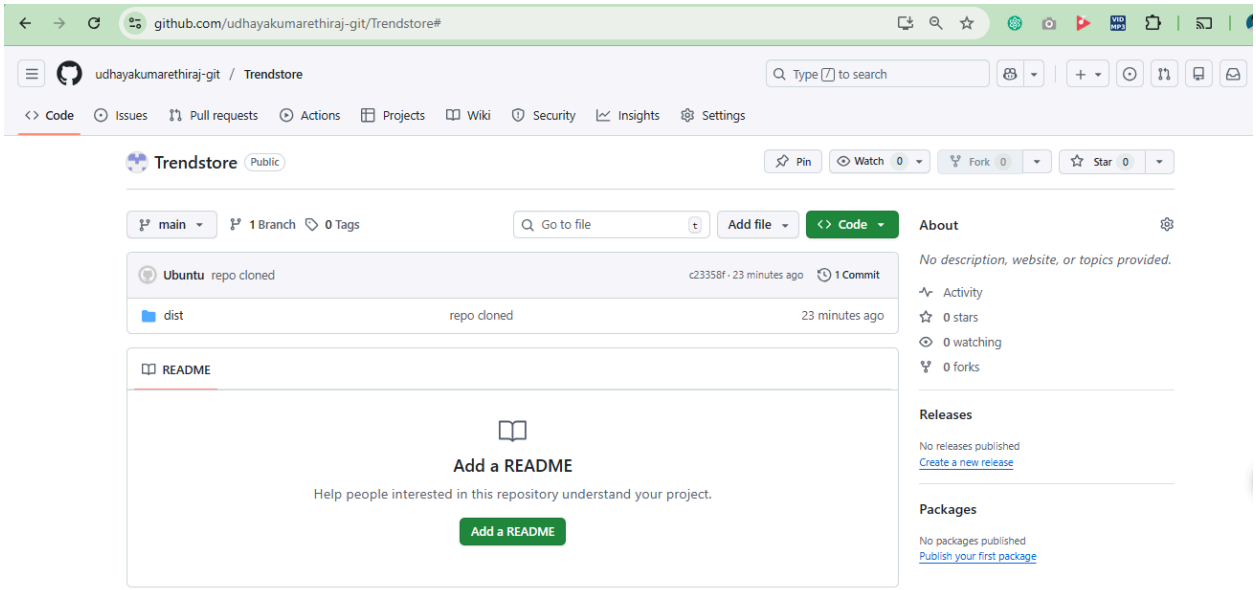
Stage : Application

Git Repo Cloned to my repo (Trend)

https://github.com/udhayakumarethiraj-git/project_2_trend_store.git

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ git clone https://github.com/Vennilavan12/Trend.git
Cloning into 'Trend'...
remote: Enumerating objects: 77, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 77 (delta 0), reused 0 (delta 0), pack-reused 75 (from 1)
Receiving objects: 100% (77/77), 8.58 MiB | 14.95 MiB/s, done.
Resolving deltas: 100% (1/1), done.
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ ll
total 12
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 20 05:54 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 20 05:54 ../
drwxrwxr-x 4 ubuntu ubuntu 4096 Jan 20 05:54 Trend/
```

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ git remote set-url origin git@github.com:udhayakumarethiraj-git/Trendstore.git
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ git push origin main
Enumerating objects: 77, done.
Counting objects: 100% (77/77), done.
Delta compression using up to 2 threads
Compressing objects: 100% (76/76), done.
Writing objects: 100% (77/77), 8.58 MiB | 2.33 MiB/s, done.
Total 77 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:udhayakumarethiraj-git/Trendstore.git
 * [new branch]      main -> main
```

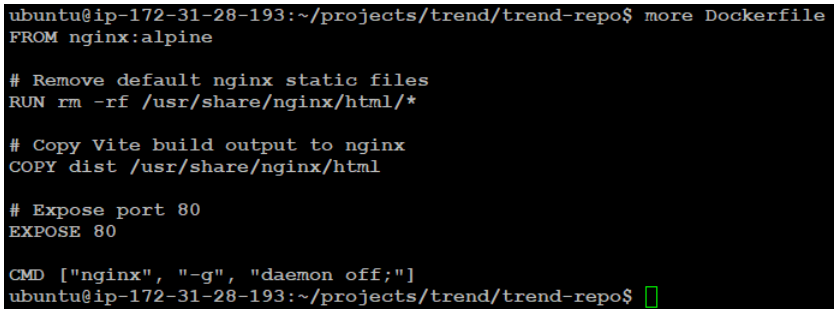


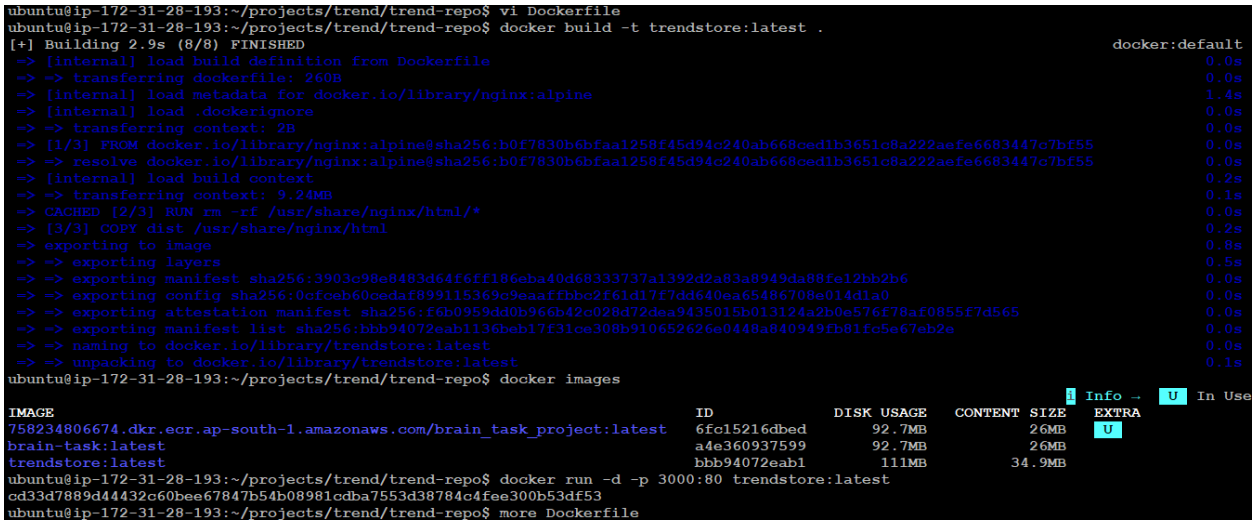
Stage : Docker

Containerize the application by creating a Dockerfile, building the image, and testing the output.

Dockerfile

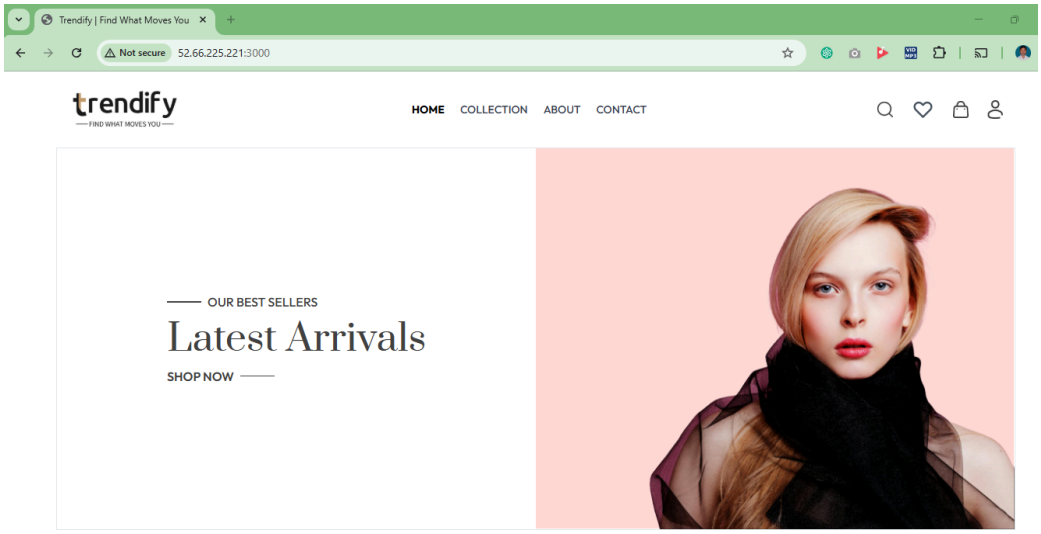
```
FROM nginx:alpine

# Remove default nginx static files
RUN rm -rf /usr/share/nginx/html/*

# Copy Vite build output to nginx
COPY dist /usr/share/nginx/html

# Expose port 80
EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```





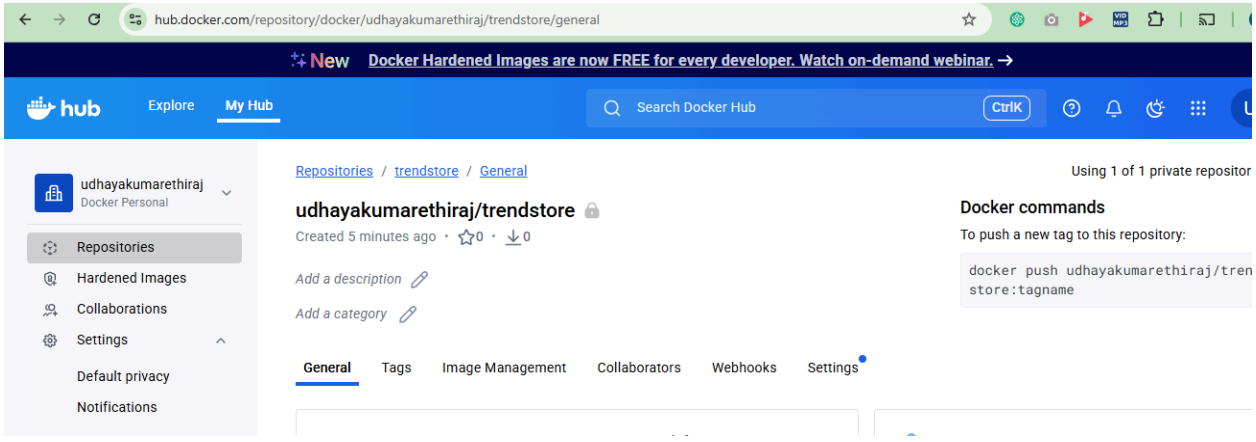
```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED         STATUS         PORTS                                       N
AMES
cd33d7889d44   trendstore:latest   "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes   0.0.0.0:3000->80/tcp, [::]:3000->80/tcp     n
ostalgic_fermi
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

Output Test result



Stage : DockerHUB

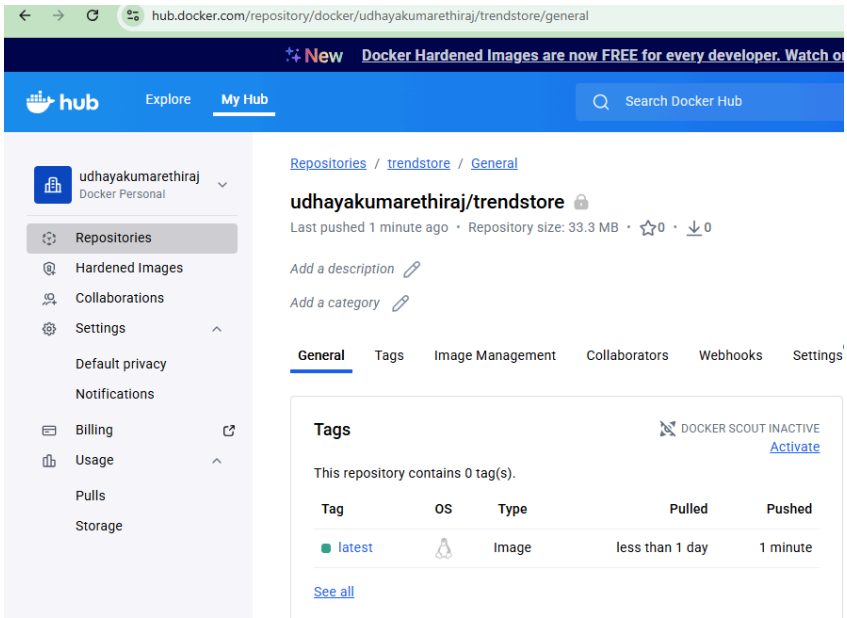
Docker Hub repo created



```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ docker login -u udhayakumarethiraj

i Info → A Personal Access Token (PAT) can be used instead.
         To create a PAT, visit https://app.docker.com/settings


Password:

WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ docker tag trendstore:latest udhayakumarethiraj/trendstore:latest
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ docker push udhayakumarethiraj/trendstore:latest
The push refers to repository [docker.io/udhayakumarethiraj/trendstore]
da7c973d8b92: Pushed
33f95a0f3229: Pushed
085c5e5aaa8e: Pushed
f5aa8e0adca0: Pushed
567f84da6fbd: Pushed
0abf9e567266: Pushed
25f453064fd3: Pushed
d1081cf8f0ea: Pushed
1074353eec0d: Pushed
e096540205d5: Pushed
f56ea78a1a5f: Pushed
latest: digest: sha256:bbb94072eab1136beb17f31ce308b910652626e0448a840949fb81fc5e67eb2e size: 856
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ docker images
                                                                                  i Info →  U  In Use
IMAGE                                                                ID             DISK USAGE   CONTENT SIZE   EXTRA
758234806674.dkr.ecr.ap-south-1.amazonaws.com/brain_task_project:latest   6fc15216dbed       92.7MB         26MB      U
brain-task:latest                                                    a4e360937599       92.7MB         26MB
trendstore:latest                                                    bbb94072eab1        111MB         34.9MB    U
udhayakumarethiraj/trendstore:latest                                 bbb94072eab1        111MB         34.9MB    U
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

Docker image pushed to Docker hub



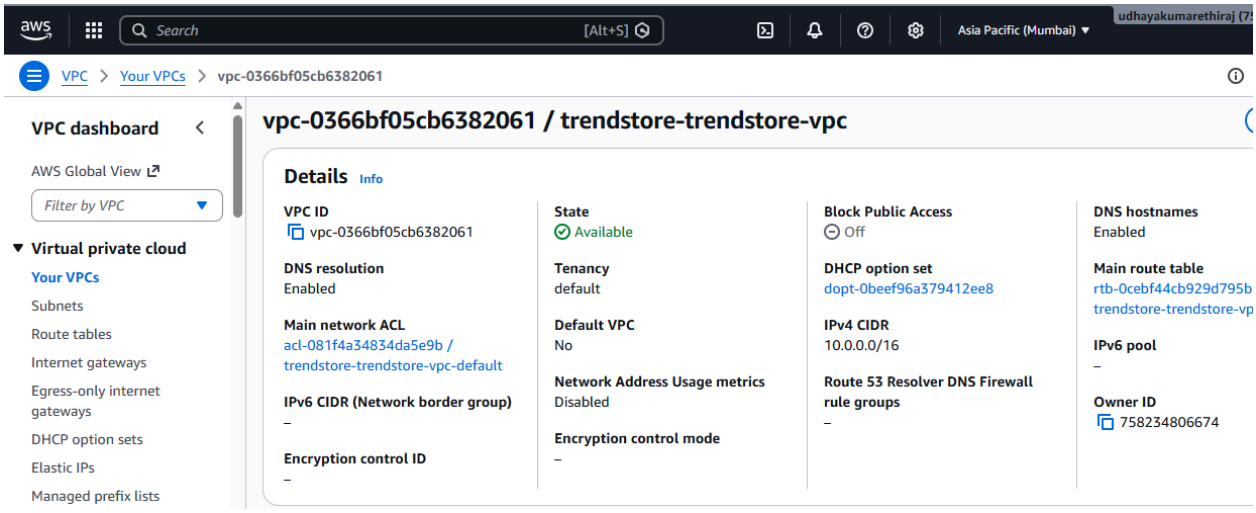
Stage : Terraform

Using Terraform VPC, IAM, EC2 created with Jenkins and EKS deployed

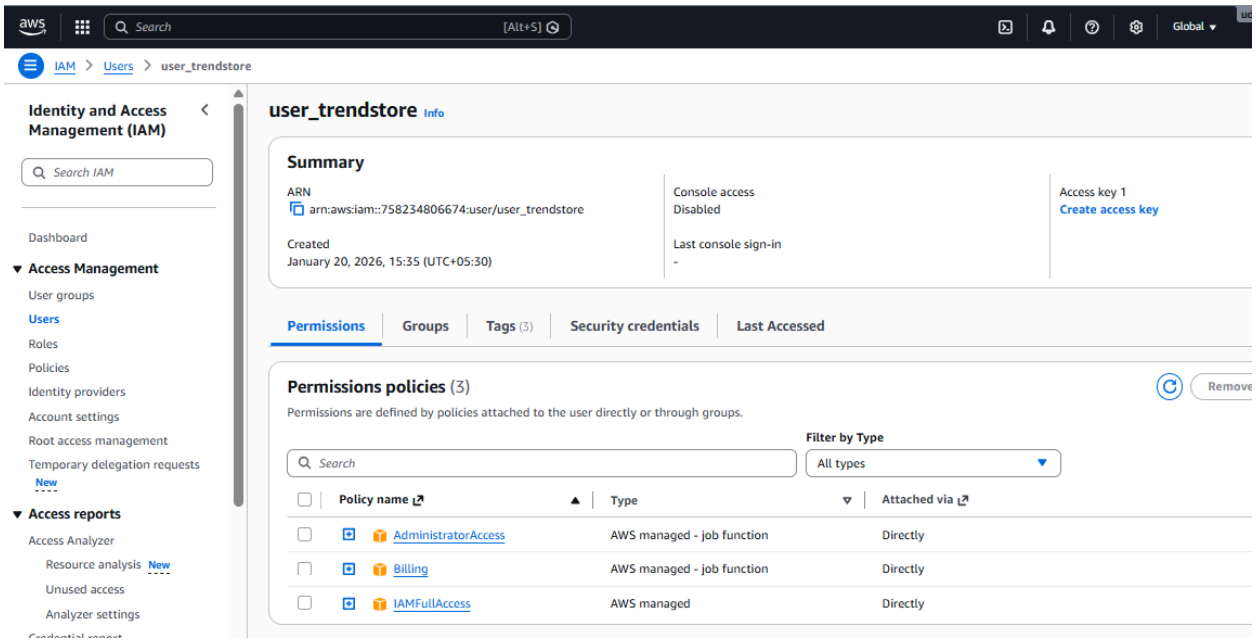
```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo/Terraform$ ll
total 28
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan 27 08:23 ./
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 27 07:04 ../
-rw-rw-r-- 1 ubuntu ubuntu 1429 Jan 27 08:22 ec2.tf
-rw-rw-r-- 1 ubuntu ubuntu  996 Jan 27 08:22 eks.tf
-rw-rw-r-- 1 ubuntu ubuntu 1036 Jan 27 08:22 iam_user.tf
-rw-rw-r-- 1 ubuntu ubuntu 1843 Jan 27 08:22 user_data.sh
-rw-rw-r-- 1 ubuntu ubuntu 1564 Jan 27 08:22 vpc.tf
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo/Terraform$
```

VPC

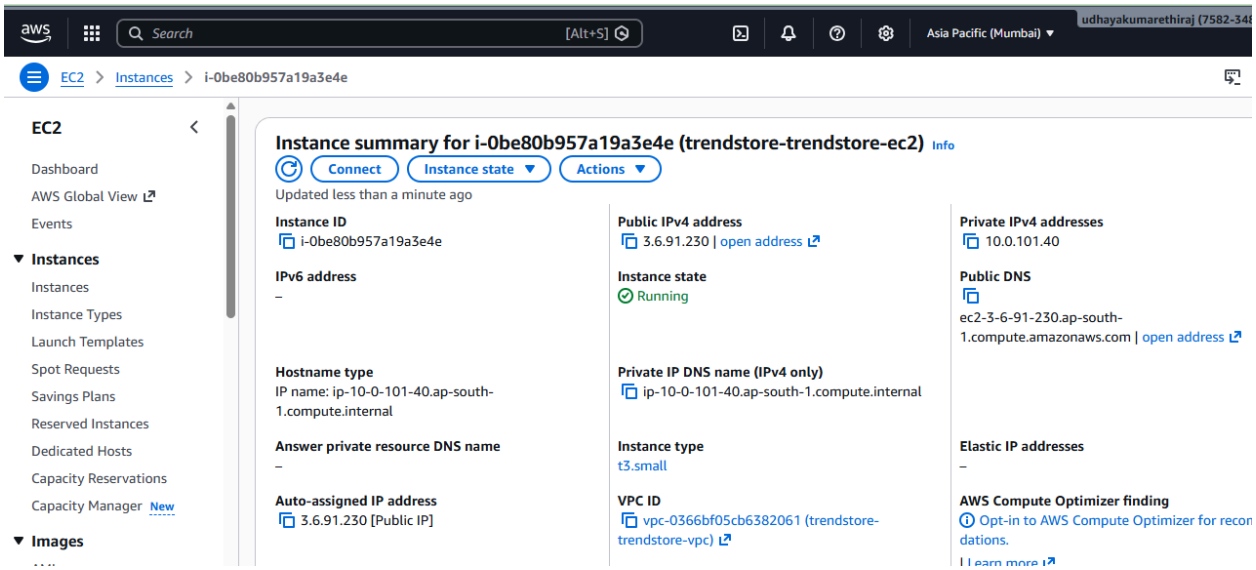
```
ubuntu@ip-172-31-28-193: ~/projects/trend/trend-repo/Terraform/envs/trendstore
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
              # (1 unchanged attribute hidden)
          },
      ]
    + name                   = "trendstore-trendstore-ec2-sg"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + region                 = "ap-south-1"
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Environment" = "trendstore"
        + "Name"        = "trendstore-trendstore-ec2-sg"
        + "Project"     = "trendstore"
      }
    + tags_all               = {
        + "Environment" = "trendstore"
        + "Name"        = "trendstore-trendstore-ec2-sg"
        + "Project"     = "trendstore"
      }
    + vpc_id                 = "vpc-0366bf05cb6382061"
  }

Plan: 2 to add, 0 to change, 0 to destroy.
module.ec2.aws_security_group.ec2_sg: Creating...
module.ec2.aws_security_group.ec2_sg: Creation complete after 2s [id=sg-0e5905d82ab83aef3]
module.ec2.aws_instance.this: Creating...
module.ec2.aws_instance.this: Still creating... [00m10s elapsed]
module.ec2.aws_instance.this: Creation complete after 12s [id=i-0be80b957a19a3e4e]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```



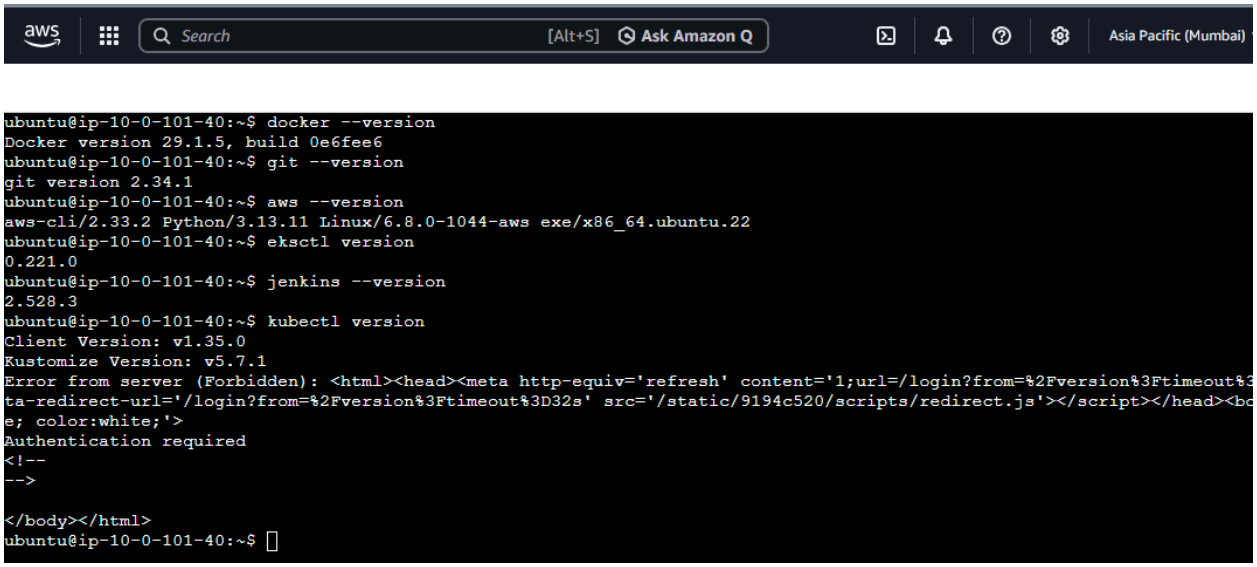
IAM



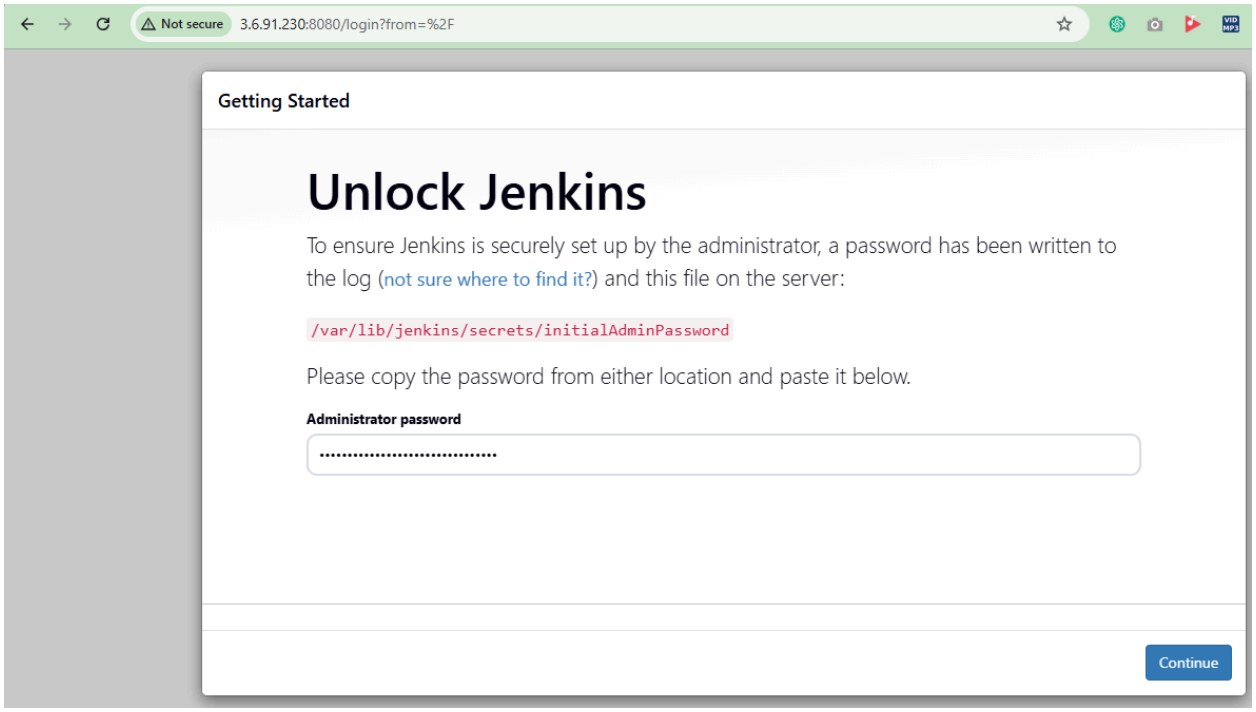
EC2 Instance

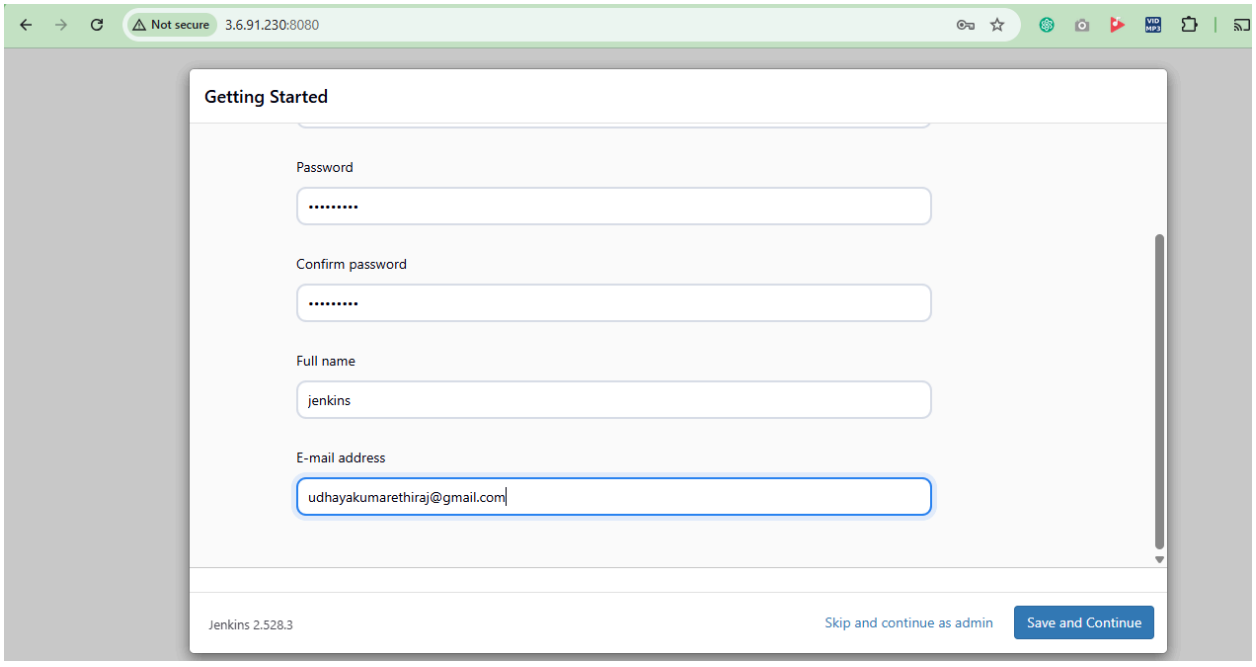


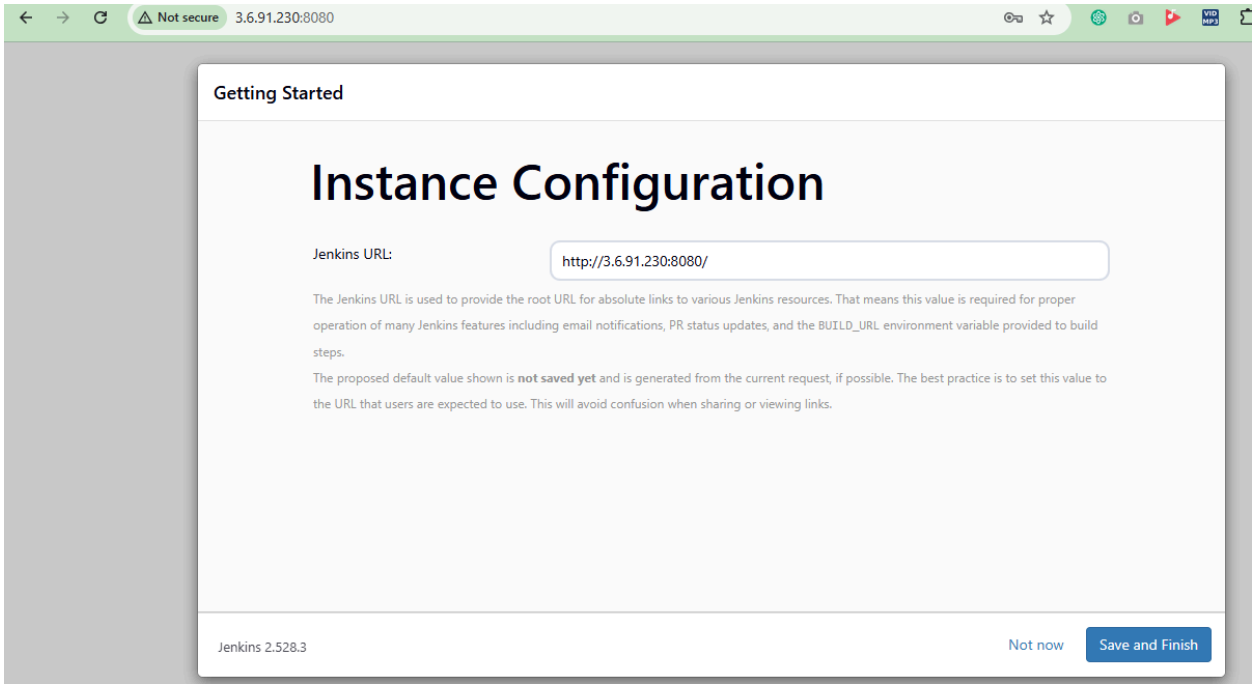
Installed packages

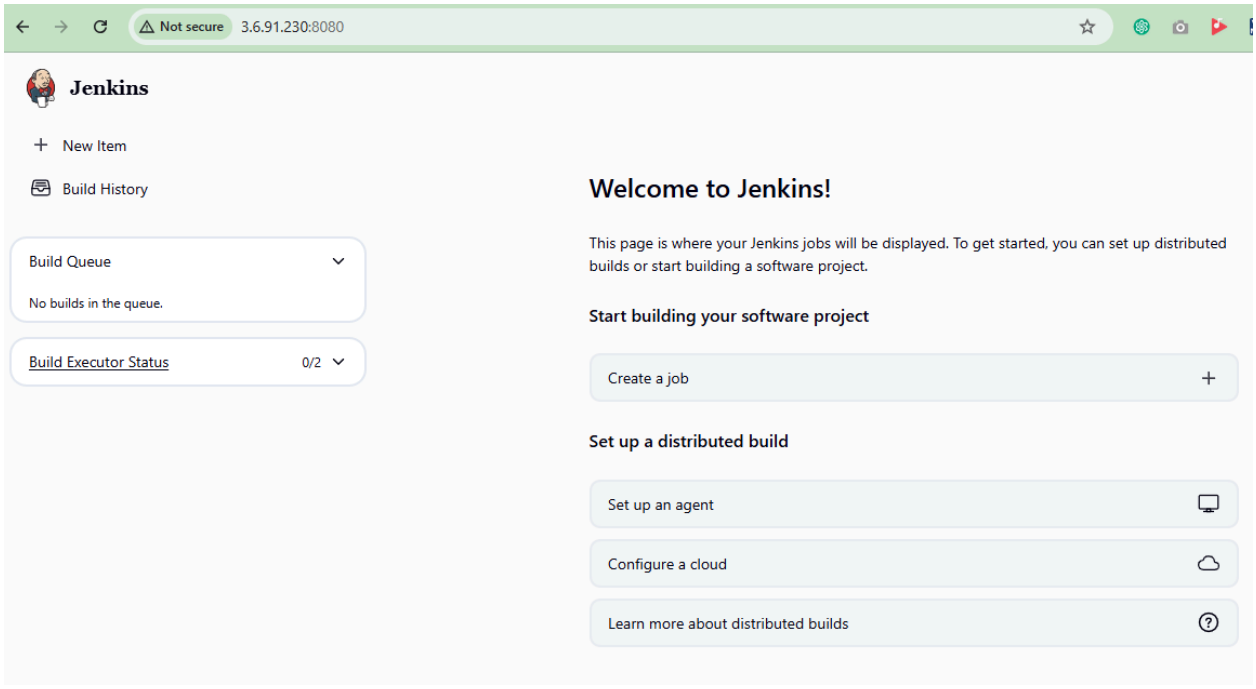


Jenkins installed


Not secure
3.6.91.230:8080/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
Continue


Not secure
3.6.91.230:8080
Getting Started
Password
Confirm password
Full name
jenkins
E-mail address
udhayakumarethiraj@gmail.com
Jenkins 2.528.3
Skip and continue as admin
Save and Continue





Stage : Kubernets

EKS Kubernets cluster created using Terraform

```
ubuntu@ip-172-31-28-193: ~/projects/eks_test
module.eks.aws_eks_addon.before_compute["vpc-cni"]: Creation complete after 14s [id=my-cluster:vpc-cni]
module.eks.time_sleep.this[0]: Still creating... [20s elapsed]
module.eks.aws_eks_addon.before_compute["eks-pod-identity-agent"]: Still creating... [20s elapsed]
module.eks.aws_eks_addon.before_compute["eks-pod-identity-agent"]: Creation complete after 24s [id=my-cluster:eks-pod-identity-ager
module.eks.time_sleep.this[0]: Still creating... [30s elapsed]
module.eks.time_sleep.this[0]: Creation complete after 30s [id=2026-01-23T06:41:07Z]
module.eks.module.eks_managed_node_group["default"].module.user_data.null_resource.validate_cluster_service_cidr: Creating...
module.eks.module.eks_managed_node_group["default"].module.user_data.null_resource.validate_cluster_service_cidr: Creation complete
ter 0s [id=8183692902810453811]
module.eks.module.eks_managed_node_group["default"].aws_launch_template.this[0]: Creating...
module.eks.module.eks_managed_node_group["default"].aws_launch_template.this[0]: Creation complete after 6s [id=lt-097e77f2b21f536e
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Creating...
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [10s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [20s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [30s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [40s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [50s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m0s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m10s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m20s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m30s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Still creating... [1m40s elapsed]
module.eks.module.eks_managed_node_group["default"].aws_eks_node_group.this[0]: Creation complete after 1m48s [id=my-cluster:defaul
026012306411275010000000d]
module.eks.aws_eks_addon.this["coredns"]: Creating...
module.eks.aws_eks_addon.this["kube-proxy"]: Creating...
module.eks.aws_eks_addon.this["coredns"]: Still creating... [10s elapsed]
module.eks.aws_eks_addon.this["kube-proxy"]: Still creating... [10s elapsed]
module.eks.aws_eks_addon.this["coredns"]: Creation complete after 14s [id=my-cluster:coredns]
module.eks.aws_eks_addon.this["kube-proxy"]: Still creating... [20s elapsed]
module.eks.aws_eks_addon.this["kube-proxy"]: Creation complete after 24s [id=my-cluster:kube-proxy]

Apply complete! Resources: 59 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-28-193:~/projects/eks_test$ eksctl get cluster
NAME            REGION          EKSCTL CREATED
my-cluster      ap-south-1      False
```

Kubernets Cluster details

Kubernets Node details

Kubernets Pod details

```
ubuntu@ip-172-31-28-193: ~/projects/trend/trend-repo
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ eksctl get cluster
NAME            REGION          EKSCTL CREATED
my-cluster      ap-south-1      False
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ kubectl get nodes
NAME                                        STATUS   ROLES    AGE   VERSION
ip-10-0-0-71.ap-south-1.compute.internal    Ready    <none>   34m   v1.32.9-eks-ecaa3a6
ip-10-0-2-35.ap-south-1.compute.internal    Ready    <none>   34m   v1.32.9-eks-ecaa3a6
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
trendstore-6dcf4bb887-fzcpt   1/1     Running   0          5m31s
trendstore-6dcf4bb887-jdjgm   1/1     Running   0          5m31s
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ kubectl get pods -A
NAMESPACE     NAME                                READY   STATUS    RESTARTS   AGE
default       trendstore-6dcf4bb887-fzcpt         1/1     Running   0          5m36s
default       trendstore-6dcf4bb887-jdjgm         1/1     Running   0          5m36s
kube-system   aws-node-86m91                      2/2     Running   0          34m
kube-system   aws-node-sbjv9                      2/2     Running   0          34m
kube-system   coredns-6f9885fdb6-6sc4q            1/1     Running   0          33m
kube-system   coredns-6f9885fdb6-bztf5            1/1     Running   0          33m
kube-system   eks-pod-identity-agent-5z6xh        1/1     Running   0          34m
kube-system   eks-pod-identity-agent-9ddpl        1/1     Running   0          34m
kube-system   kube-proxy-stx2p                    1/1     Running   0          33m
kube-system   kube-proxy-z9w9c                    1/1     Running   0          33m
```

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ ll
total 36
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 23 07:10 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 21 08:51 ../
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan 20 16:13 .git/
-rw-rw-r-- 1 ubuntu ubuntu  216 Jan 20 10:43 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu  221 Jan 20 06:23 Dockerfile
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 23 07:14 Terraform/
-rw-rw-r-- 1 ubuntu ubuntu  481 Jan 23 07:07 deployment.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 20 05:54 dist/
-rw-rw-r-- 1 ubuntu ubuntu  417 Jan 23 07:10 service.yaml
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

```
ubuntu@ip-172-31-28-193: ~/projects/trend/trend-repo
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ ll
total 36
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 23 07:10 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 21 08:51 ../
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan 20 16:13 .git/
-rw-rw-r-- 1 ubuntu ubuntu  216 Jan 20 10:43 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu  221 Jan 20 06:23 Dockerfile
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 23 07:14 Terraform/
-rw-rw-r-- 1 ubuntu ubuntu  481 Jan 23 07:07 deployment.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 20 05:54 dist/
-rw-rw-r-- 1 ubuntu ubuntu  417 Jan 23 07:10 service.yaml
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ kubectl get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)
       AGE
kubernetes   ClusterIP      172.20.0.1      <none>                                                                   443/TCP
       38m
trendstore   LoadBalancer   172.20.128.99   a8392e2ba0a6844878839c8fa2307040-b2ec893399960f82.elb.ap-south-1.amazonaws.com   3000:3115
2/TCP   4m19s
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

Kubernets deployment

Kubernets deployment and service created

Secrets added for Docker hub private image pull request

kubectl create secret generic dockerhub-secret
--from-file=.dockerconfigjson=$HOME/.docker/config.json
--type=kubernetes.io/dockerconfigjson

```
ubuntu@ip-172-31-28-193:~/projects/eks_test$ kubectl create secret generic dockerhub-secret   --from-file=.dockerconfigjson=$HOME/.doc
ker/config.json   --type=kubernetes.io/dockerconfigjson
secret/dockerhub-secret created
ubuntu@ip-172-31-28-193:~/projects/eks_test$ kubectl get secrets
NAME              TYPE                             DATA   AGE
dockerhub-secret   kubernetes.io/dockerconfigjson   1      3m1s
ubuntu@ip-172-31-28-193:~/projects/eks_test$
```

Application deployed using Deployment.yaml & Service.yaml

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ more deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: trendstore
  labels:
    app: trendstore
spec:
  replicas: 2
  selector:
    matchLabels:
      app: trendstore
  template:
    metadata:
      labels:
        app: trendstore
    spec:
      imagePullSecrets:
        - name: dockerhub-secret
      containers:
        - name: trendstore
          image: udhayakumarethiraj/trendstore:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 3000

ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ more service.yaml
apiVersion: v1
kind: Service
metadata:
  name: trendstore
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: "classic"
    service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
spec:
  type: LoadBalancer
  selector:
    app: trendstore
  ports:
    - port: 3000          # external port (what clients use)
      targetPort: 3000    # container port inside the pod (updated)
      protocol: TCP
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

Stage : Jenkins

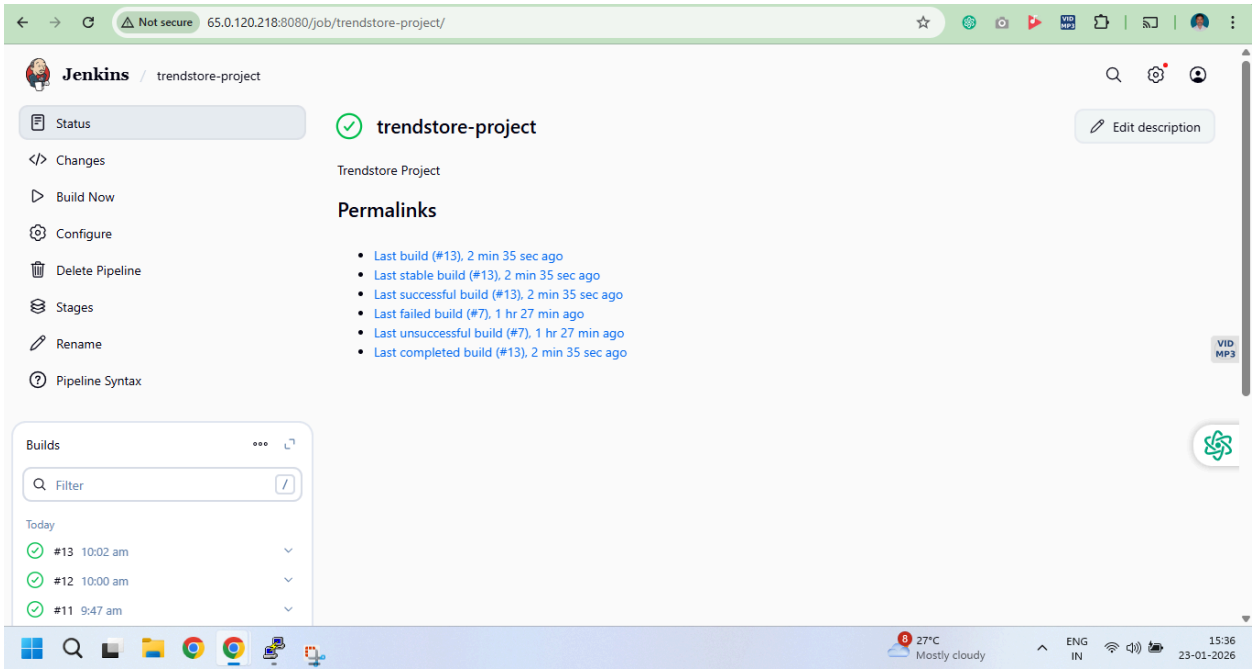
Jenkins setup

A Jenkinsfile created to define the CI/CD pipeline stages, including checkout, build, push, and deployment.

```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ ll
total 40
drwxrwxr-x 5 ubuntu ubuntu 4096 Jan 23 10:00 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 23 07:28 ../
drwxrwxr-x 8 ubuntu ubuntu 4096 Jan 23 10:01 .git/
-rw-rw-r-- 1 ubuntu ubuntu  216 Jan 20 10:43 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu  221 Jan 20 06:23 Dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 2184 Jan 23 10:00 Jenkinsfile
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 23 07:14 Terraform/
-rw-rw-r-- 1 ubuntu ubuntu  481 Jan 23 07:07 deployment.yaml
drwxrwxr-x 3 ubuntu ubuntu 4096 Jan 20 05:54 dist/
-rw-rw-r-- 1 ubuntu ubuntu  417 Jan 23 07:10 service.yaml
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$
```

Jenkins CICD Pipeline

Trendstore pipeline project created



Jenkins Console Output : [URL]

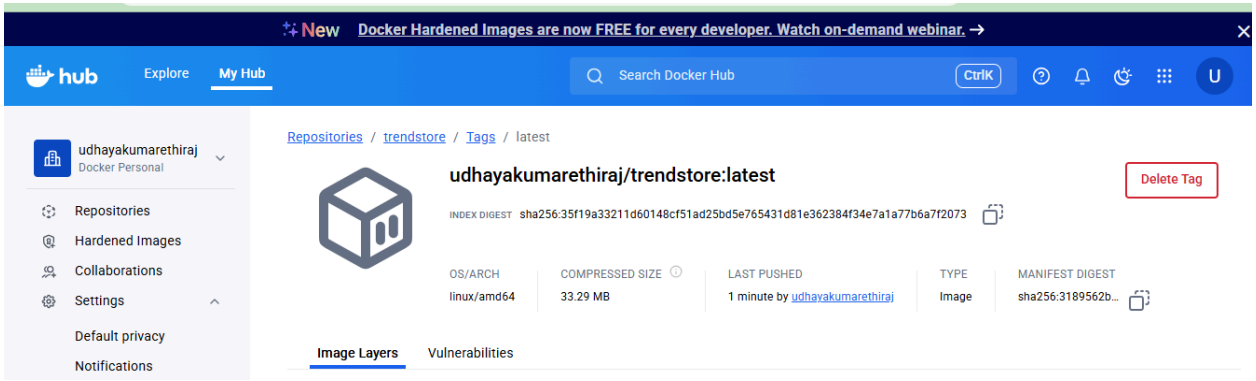
Jenkins Build verification

Implemented a CI/CD pipeline using Jenkins to build Docker images, push them to Docker Hub, and deploy them to Kubernetes. The Docker Image ID was validated to ensure the same image was consistently used across the Jenkins build, Docker Hub registry, and Kubernetes deployment stages.

Jenkins build Docker Image pushed details from Jenkins log

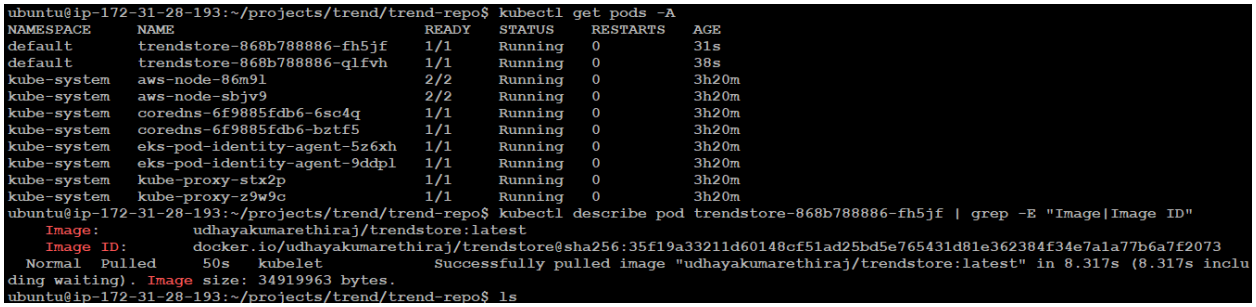
Docker image Console output log

Pushed Image details in Docker Hub



The same Image deployed in Kubernets

pod image details



Application output :

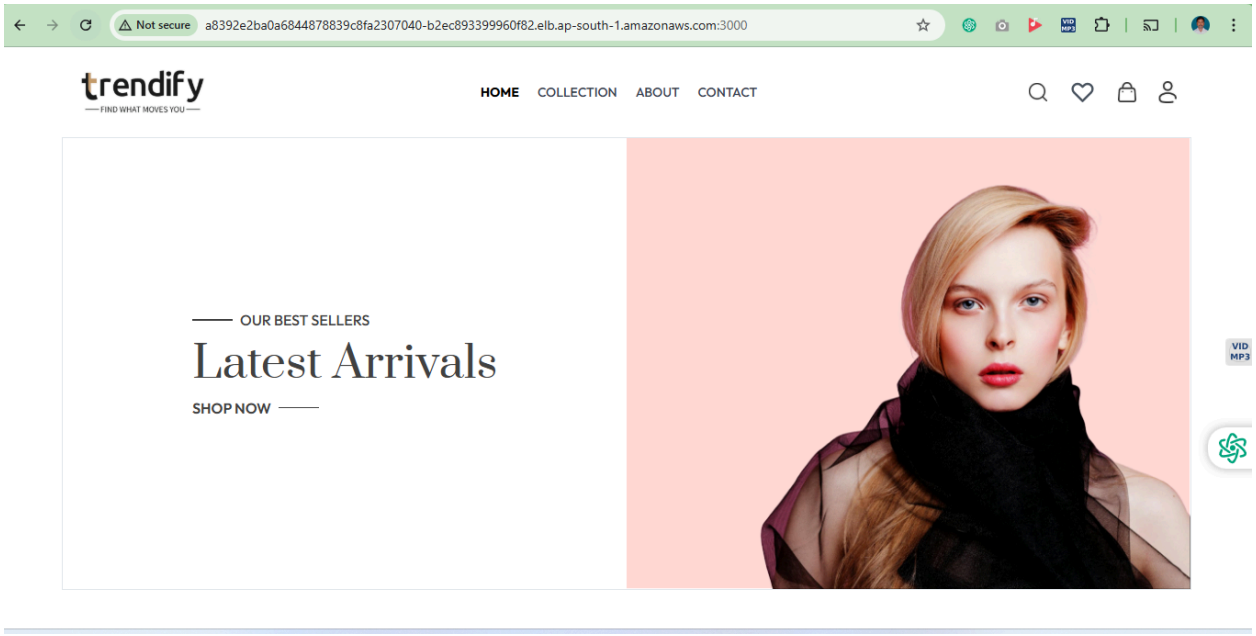
http://a8392e2ba0a6844878839c8fa2307040-b2ec893399960f82.elb.ap-south-1.amazonaws.com:3000/

Load balancer ARN

arn:aws:elasticloadbalancing:ap-south-1:758234806674:loadbalancer/net/a8392e2ba0a6844878839c8fa2307040/b2ec893399960f82

Load balancer DNS

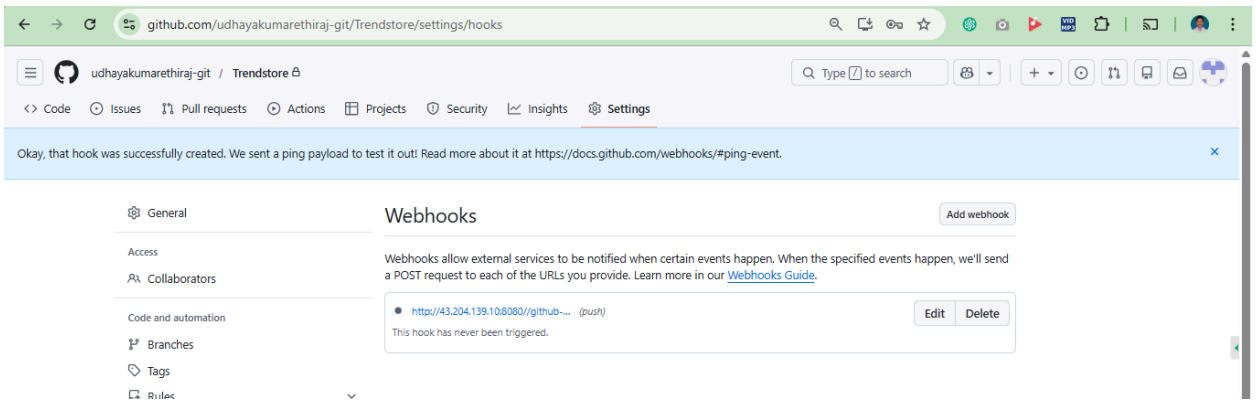
a8392e2ba0a6844878839c8fa2307040-b2ec893399960f82.elb.ap-south-1.amazonaws.com



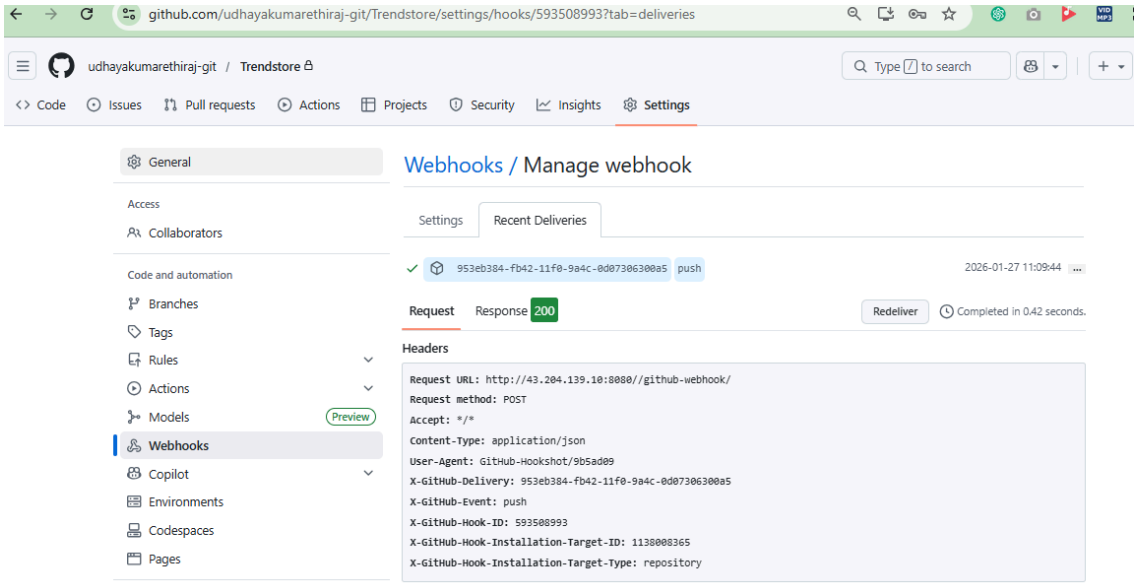
GitHub and Jenkins Integration

Integrated GitHub with Jenkins using webhooks to automatically trigger builds on every commit, enabling continuous integration and eliminating manual build execution.

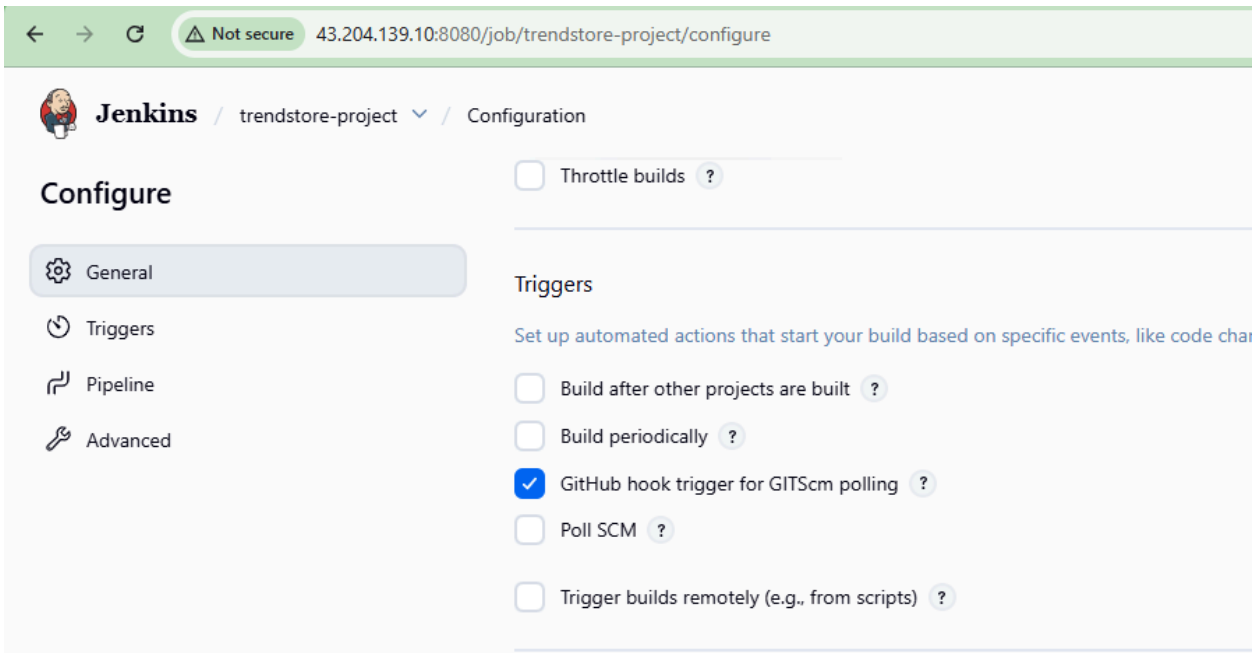
Webhook added for build auto trigger



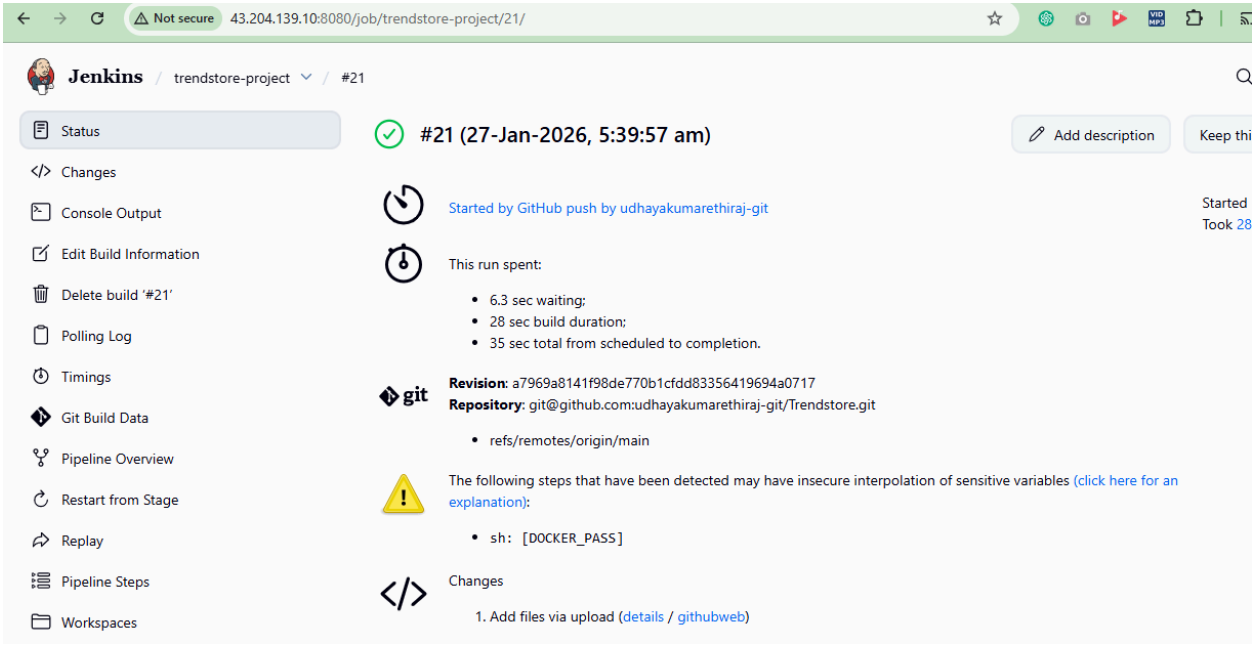
Github webhook Auto Triggered



Jenkins Build config



Jenkins Build status



Stage : Monitoring

An open-source monitoring system was implemented using Prometheus to collect cluster and application metrics, and Grafana was configured with dashboards to monitor application health, application URL status, and real-time metrics data.

Install Node Exporter

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update

helm install node-exporter prometheus-community/prometheus-node-exporter \
  --namespace monitoring
```

Install Kube Metrics

```
helm install kube-state-metrics prometheus-community/kube-state-metrics \
  --namespace monitoring
```

Verify

```
kubectl get pods -n monitoring
```

Expose it via LoadBalancer

Node Exporter LB

```
kubectl expose daemonset node-exporter-prometheus-node-exporter \
  -n monitoring \
  --name node-exporter-lb \
  --type LoadBalancer \
  --port 9100 \
  --target-port 9100
```

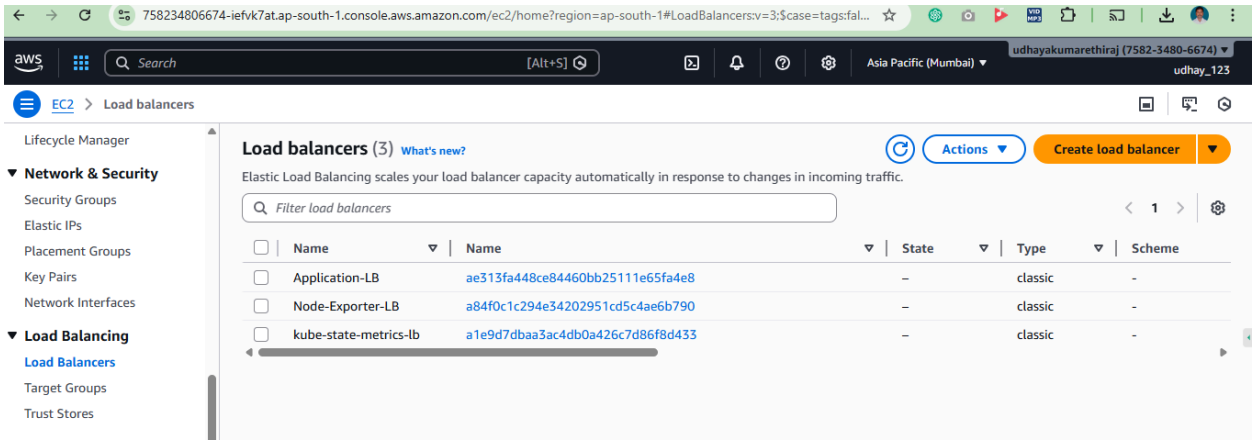
Kube State Metrics LB

```
kubectl expose deployment kube-state-metrics \
  -n monitoring \
  --name kube-state-metrics-lb \
  --type LoadBalancer \
  --port 8080 \
  --target-port 8080
```

Verify

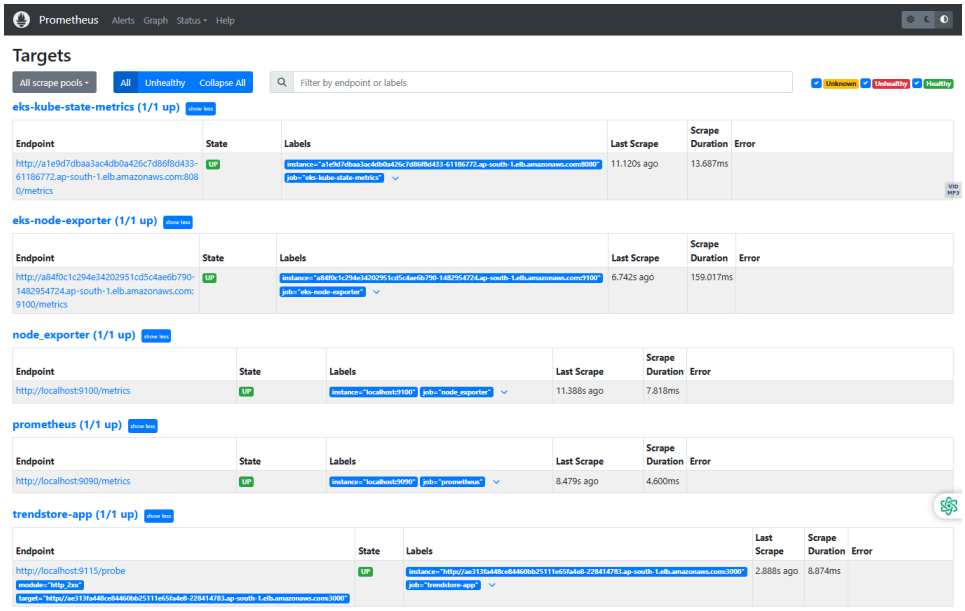
```
kubectl get svc -n monitoring
```

Kubernets metrics forwarded to Prometheus Server via loadbalancers

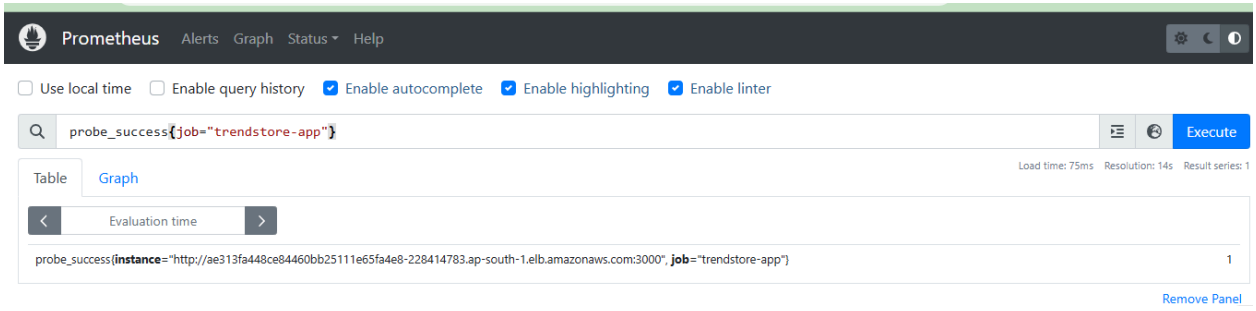


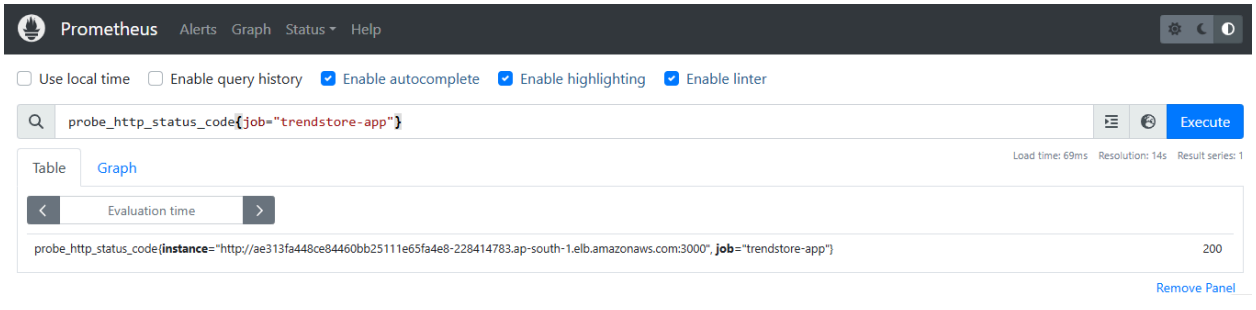
Prometheus Data from Kubernets

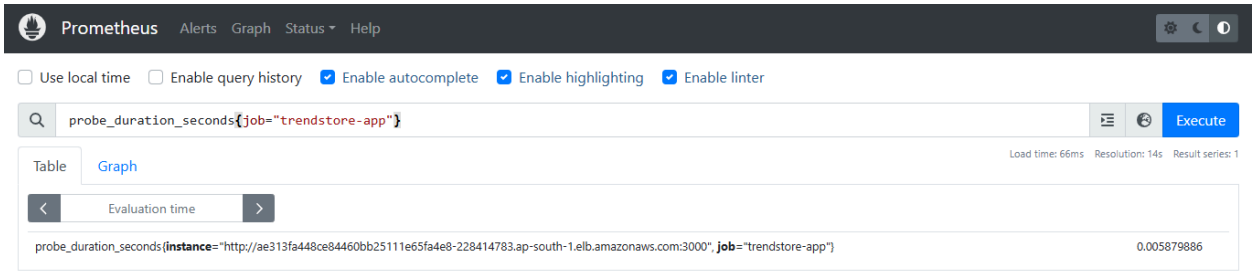
Targets added to Prometheus



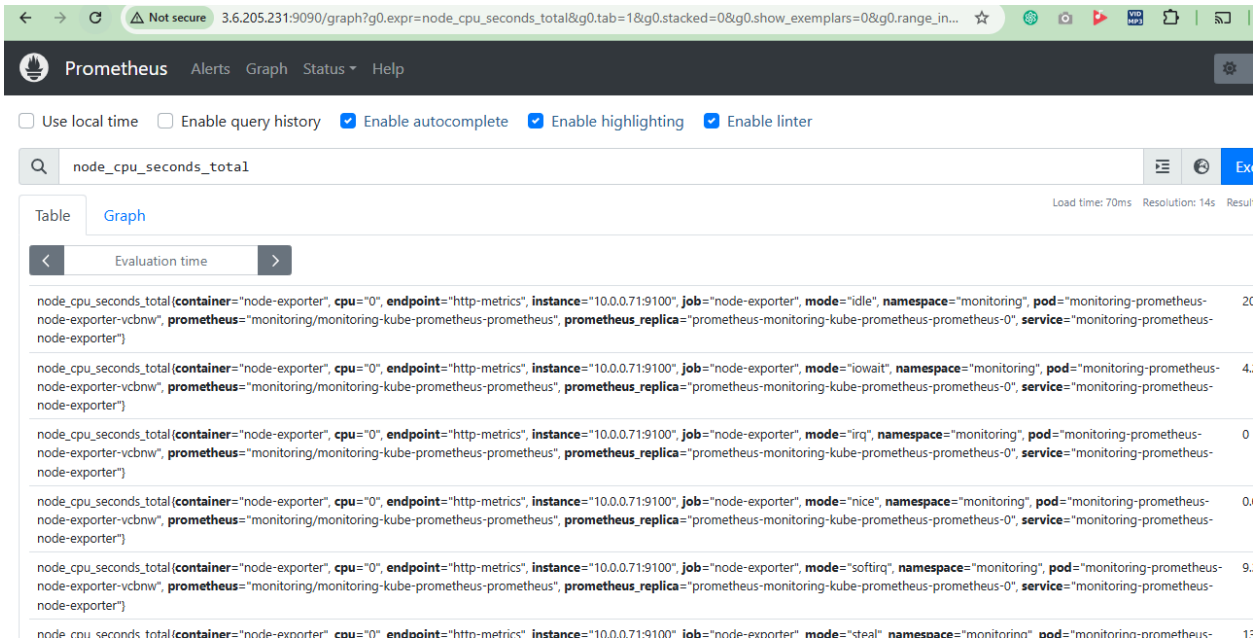
Query response



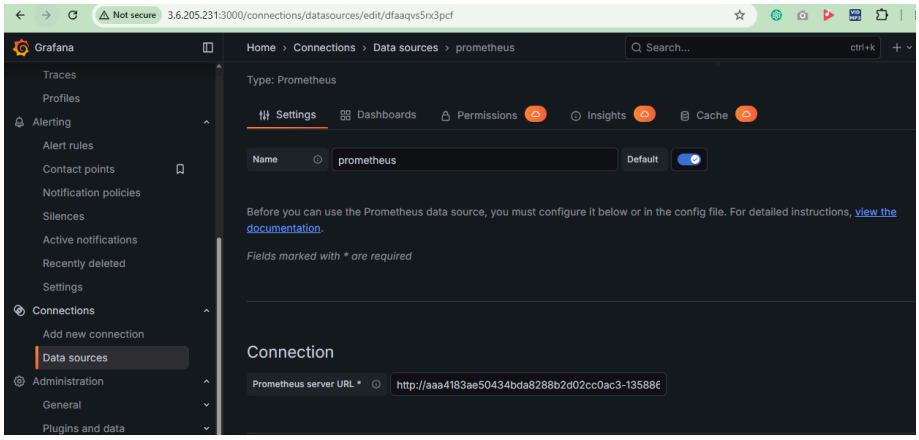





```
ubuntu@ip-172-31-28-193:~/projects/trend/trend-repo$ kubectl get svc monitoring-kube-prometheus-prometheus -n monitoring -w
NAME                                      TYPE           CLUSTER-IP     EXTERNAL-IP
                PORT(S)                         AGE
monitoring-kube-prometheus-prometheus     LoadBalancer   172.20.243.2   aaa4183ae50434bda8288b2d02cc0ac3-1358863498.ap-south-1.elb.amazo
naws.com      9090:32226/TCP,8080:31249/TCP   112m
```

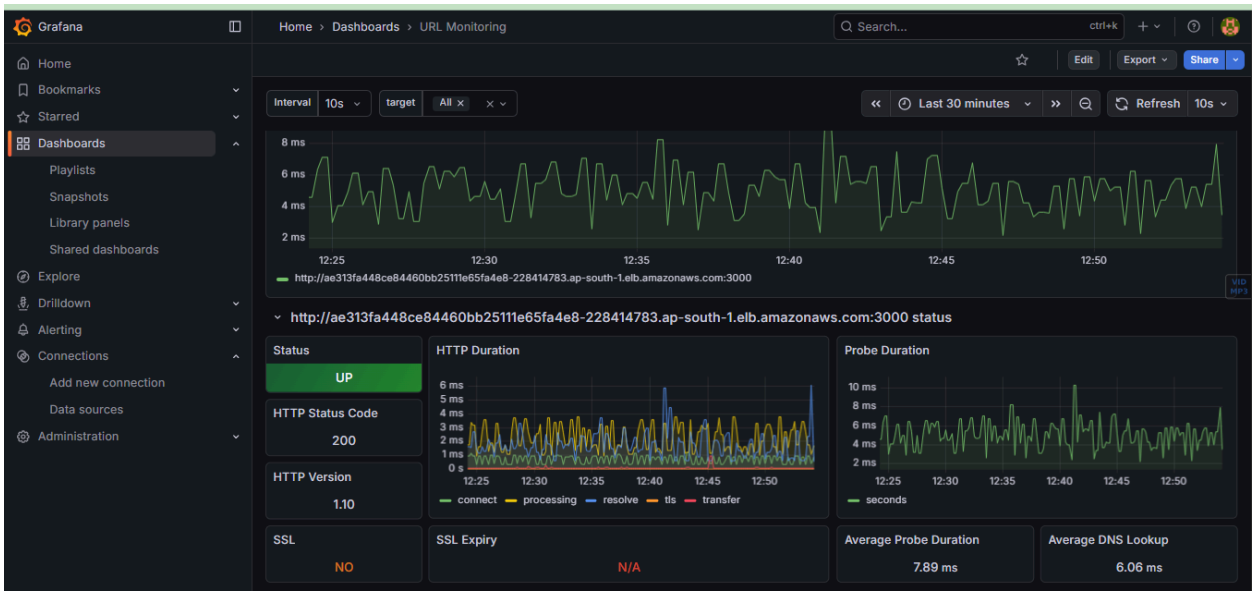


Added Kubernets Data source to Grafana

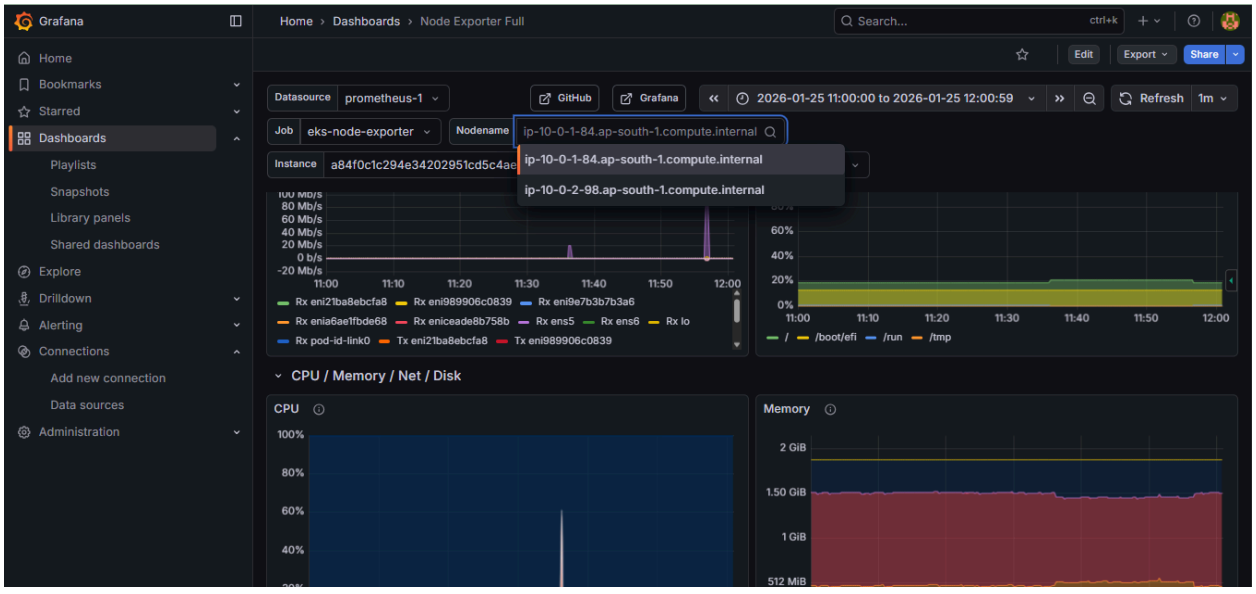


Grafana Dashboards :

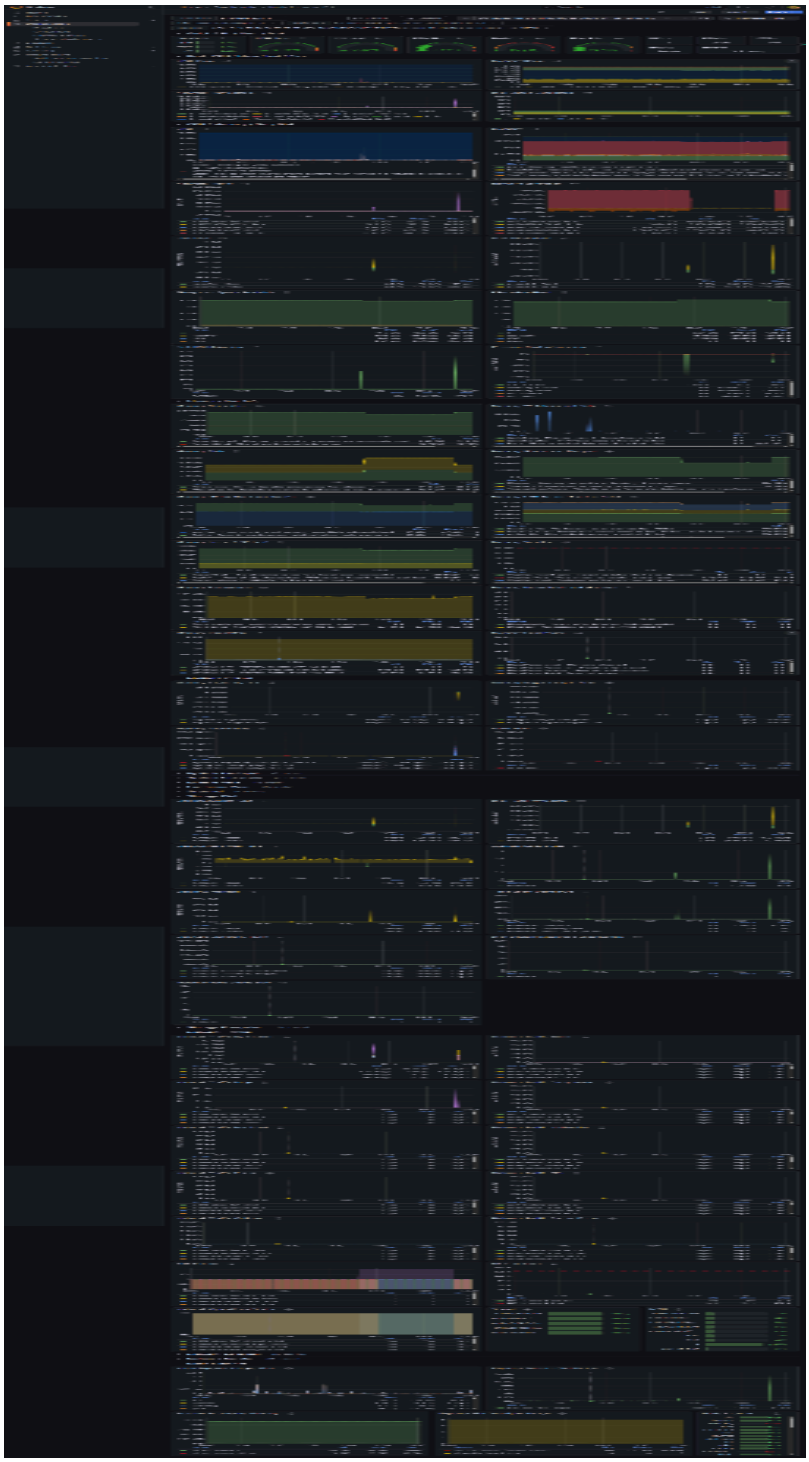
Application URL



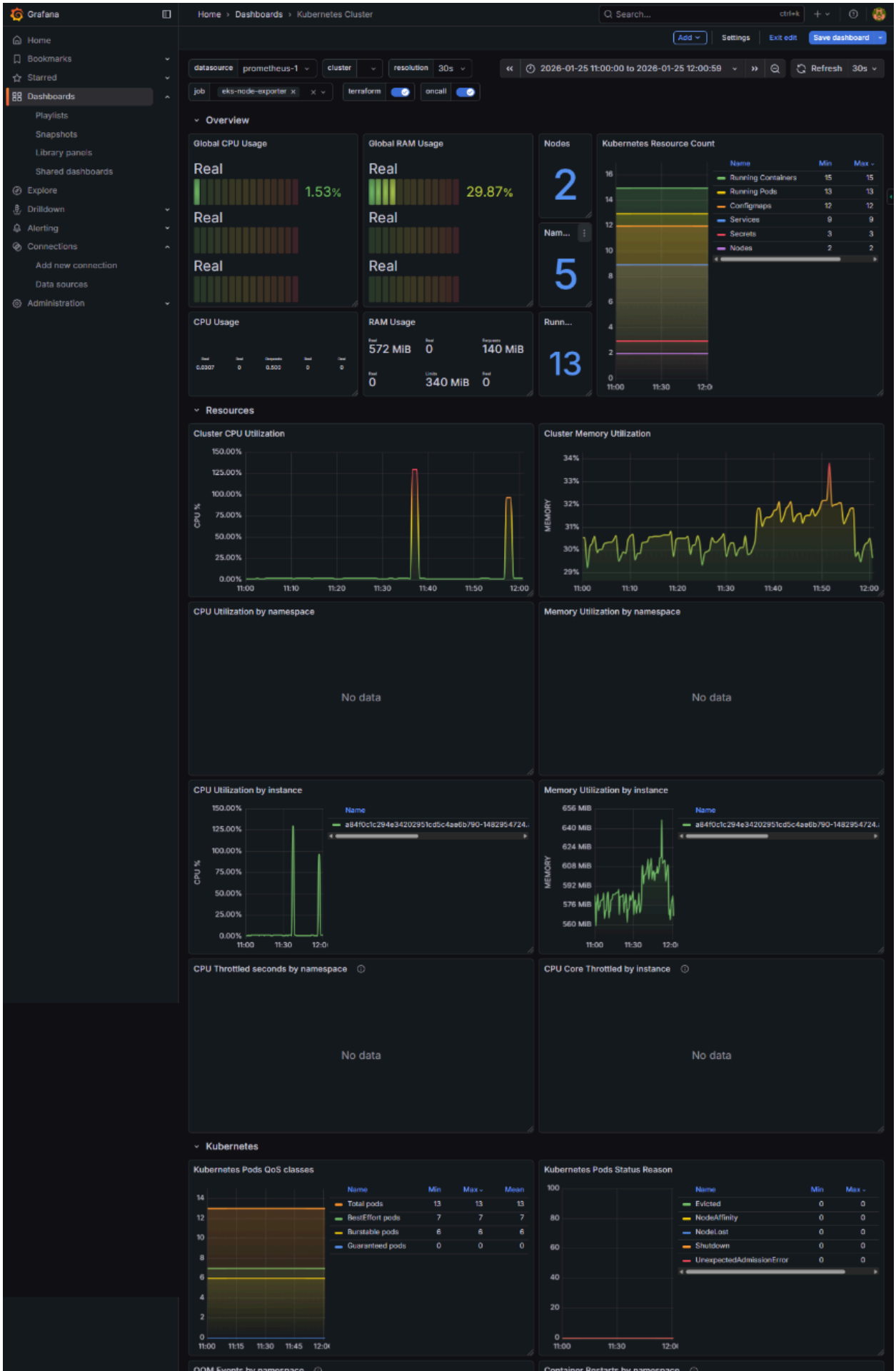
Node Monitoring Data



Node Monitoring full page data



Cluster Monitoring Data


Grafana
Home › Dashboards › Kubernetes Cluster
Search...
Add
Settings
Exit edit
Save dashboard
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Connections
Add new connection
Data sources
Administration
datasource prometheus-1
cluster
resolution 30s
2026-01-25 11:00:00 to 2026-01-25 12:00:59
Refresh 30s
job eks-node-exporter
terraform
oncall
Overview
Global CPU Usage
Real
1.53%
Real
Real
Global RAM Usage
Real
29.87%
Real
Real
Nodes
2
Nam...
5
Runn...
13
Kubernetes Resource Count
Name Min Max
Running Containers 15 15
Running Pods 13 13
Configmaps 12 12
Services 9 9
Secrets 3 3
Nodes 2 2
CPU Usage
RAM Usage
572 MiB
0
140 MiB
0
340 MiB
0
Resources
Cluster CPU Utilization
CPU %
150.00%
125.00%
100.00%
75.00%
50.00%
25.00%
0.00%
11:00 11:10 11:20 11:30 11:40 11:50 12:00
Cluster Memory Utilization
MEMORY
34%
33%
32%
31%
30%
29%
11:00 11:10 11:20 11:30 11:40 11:50 12:00
CPU Utilization by namespace
No data
Memory Utilization by namespace
No data
CPU Utilization by instance
Name
a84f0c1c294e34202951cd5c4ae6b790-1482954724..
Memory Utilization by instance
656 MiB
640 MiB
624 MiB
608 MiB
592 MiB
576 MiB
560 MiB
CPU Throttled seconds by namespace
No data
CPU Core Throttled by instance
No data
Kubernetes
Kubernetes Pods QoS classes
Name Min Max Mean
Total pods 13 13 13
BestEffort pods 7 7 7
Burstable pods 6 6 6
Guaranteed pods 0 0 0
Kubernetes Pods Status Reason
Name Min Max
Evicted 0 0
NodeAffinity 0 0
NodeLost 0 0
Shutdown 0 0
UnexpectedAdmissionError 0 0