

Brain-Tasks-App Deployment Project

Project Overview

This project demonstrates the deployment of a React application to a production-ready environment using Docker, AWS Codepipeline, AWS EKS, and CloudWatch for monitoring. The application is containerized and deployed to an EKS cluster with CI/CD automation via CodePipeline.

Features

- React frontend running on port 80
- Dockerized for container management
- AWS CodePipeline for automated builds and deployments
- AWS EKS for scalable Kubernetes deployment
- CloudWatch logs integration for monitoring application and cluster activity

Prerequisites

- AWS CLI configured
- Kubernetes CLI (kubectl)
- Docker installed
- Jenkins server with access to GitHub repository
- AWS EKS cluster setup

Deployment Steps

1. Clone the Repository

```
git clone https://github.com/Deepak-r-2001/Brain-Tasks-App.git
```

```
cd Brain-Tasks-App
```

2. Docker Build and Push

```
docker build -t deepwhoo/brain-tasks-app:latest .
```

```
docker push deepwhoo/brain-tasks-app:latest
```

3. Kubernetes Deployment

```
kubectl apply -f deployment.yaml
```

```
kubectl apply -f service.yaml
```

```
kubectl rollout status deployment brain-tasks-deployment
```

kubectrl get pods

4. Enable CloudWatch Logs for Monitoring

kubectrl create namespace amazon-cloudwatch

kubectrl apply -f cwagent-fluent-bit.yaml

kubectrl get pods -n amazon-cloudwatch

5. Jenkins CI/CD Pipeline

- Set up AWS CodePipeline with GitHub repo integration
- Configure build steps:
 - Build Docker image
 - Push to DockerHub
 - Deploy to EKS using kubectrl or CodeDeploy
- Enable CloudWatch logs tracking for monitoring

Repository

GitHub Repository: <https://github.com/Deepak-r-2001/Brain-Tasks-App>

Monitoring

- CloudWatch logs show build, deploy, and application logs
- Visualization can be added to CloudWatch dashboard

Project Screenshots

Screenshot 1

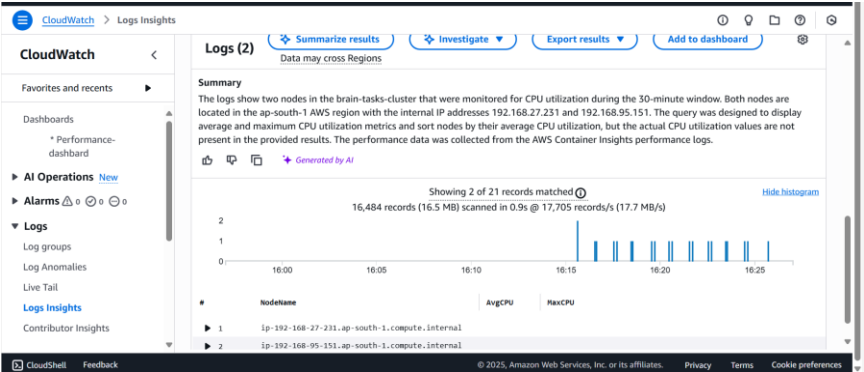
Amazon CloudWatch Add on

```
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ aws eks describe-addon \
  --cluster-name brain-tasks-cluster \
  --addon-name amazon-cloudwatch-observability \
  --region ap-south-1 \
  --query "addon.status"
"ACTIVE"

EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$
```

Screenshot 2

CPU usage



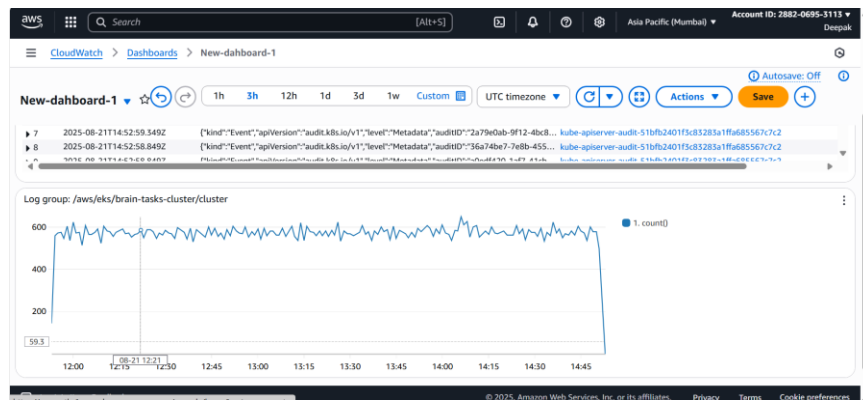
Screenshot 3

Code deployment

```
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ aws deploy get-application \
  --application-name brain-tasks-app-deploy \
  --region ap-south-1
{
  "application": {
    "applicationId": "62d416ce-c1b1-476f-ae8d-b5267bed0e1a",
    "applicationName": "brain-tasks-app-deploy",
    "createTime": "2025-08-20T18:40:45.439000+05:30",
    "linkedToGitHub": false,
    "computePlatform": "ECS"
  }
}
```

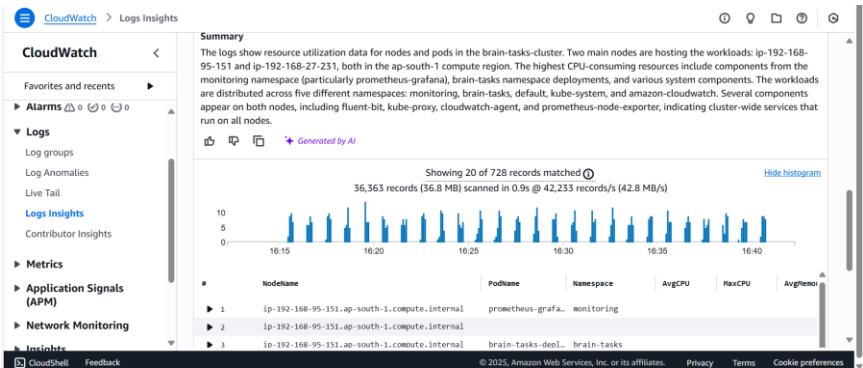
Screenshot 4

Dashboard 1



Screenshot 5

Data utilisation of nodes and pods



Screenshot 6

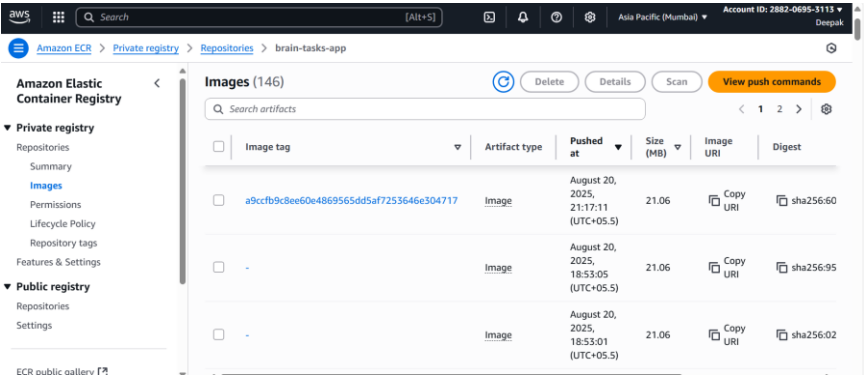
Deployment group

```
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ aws deploy list-deployment-groups \
  --application-name brain-tasks-app-deploy \
  --region ap-south-1
{
  "deploymentGroups": []
}

EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$
```

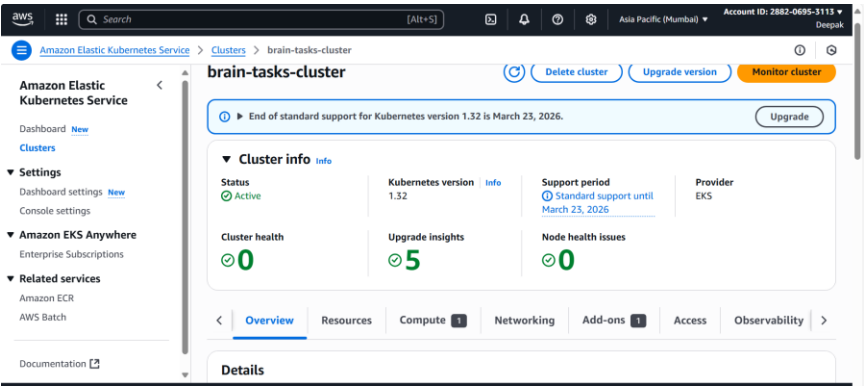
Screenshot 7

ECR Repository



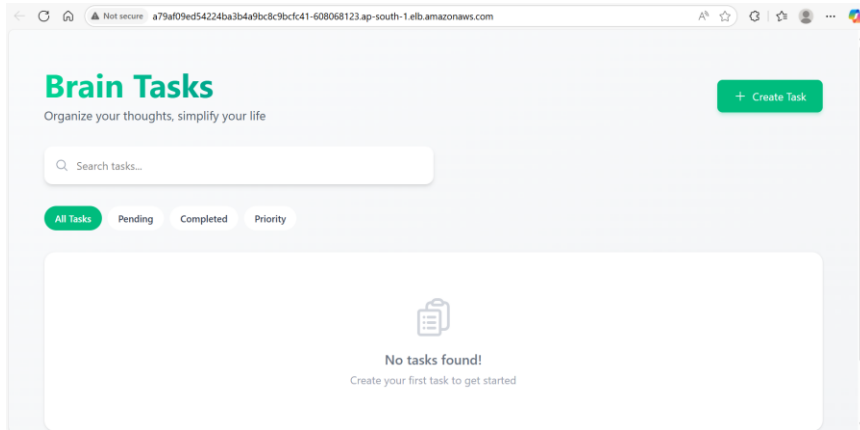
Screenshot 8

EKS cluster status



Screenshot 9

Exposed app via LoadBalancer & tested public URL



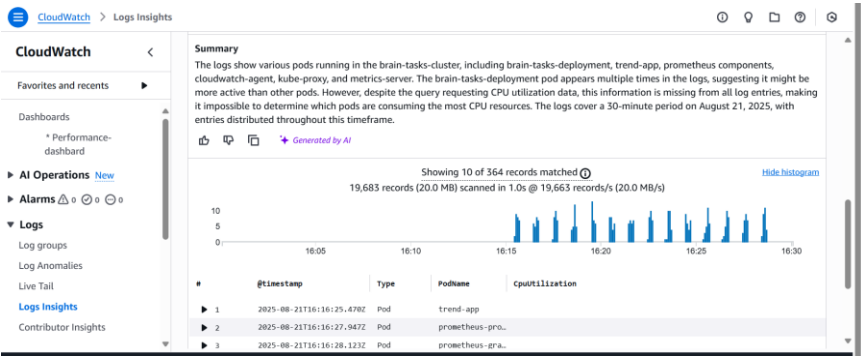
Screenshot 10

Git Clone

```
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ git clone https://github.com/Vennilavan12/Brain-Tasks-App.git
Cloning into 'Brain-Tasks-App'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (8/8), 100.04 KiB | 1.01 MiB/s, done.
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
> $
```

Screenshot 11

High CPU usage pods



Screenshot 12

Kubernetes DaemonSets

```
EAD+dr222@GB-5CD228DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ kubectl get daemonset -n amazon-cloudwatch
```

NAME	AGE	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR
cloudwatch-agent	74m	2	2	2	2	2	kubernetes.io/os=linux
cloudwatch-agent-windows	74m	0	0	0	0	0	kubernetes.io/os=window
cloudwatch-agent-windows-container-insights	74m	0	0	0	0	0	kubernetes.io/os=window
dcm-exporter	74m	0	0	0	0	0	kubernetes.io/os=linux
fluent-bit	75m	2	2	2	2	2	kubernetes.io/os=linux
fluent-bit-windows	75m	0	0	0	0	0	kubernetes.io/os=window

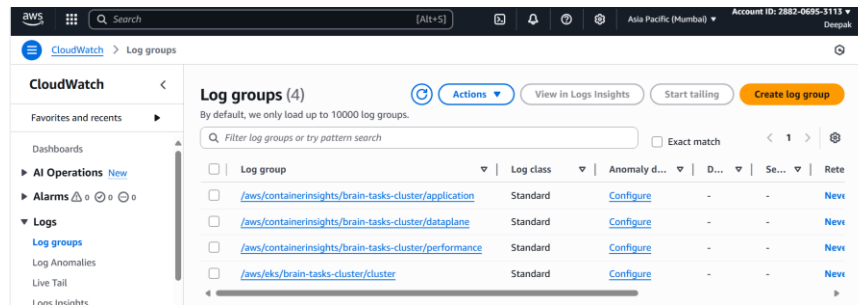
Screenshot 13

Kubernetes Deployment

```
• EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ kubectl get pods -n default
NAME                                READY   STATUS    RESTARTS   AGE
brain-tasks-deployment-9557664f-hqrb1 1/1     Running   0           27h
brain-tasks-deployment-9557664f-q6rdn 1/1     Running   0           27h
trend-app-55d46c64c4-d9s9g            1/1     Running   0           13d
trend-app-55d46c64c4-gp86c            1/1     Running   0           13d
EAD+dr222@GB-5CD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
```

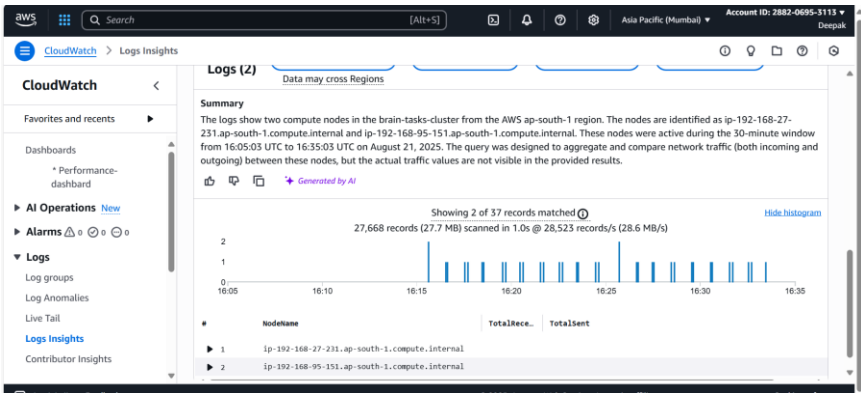
Screenshot 14

Log groups



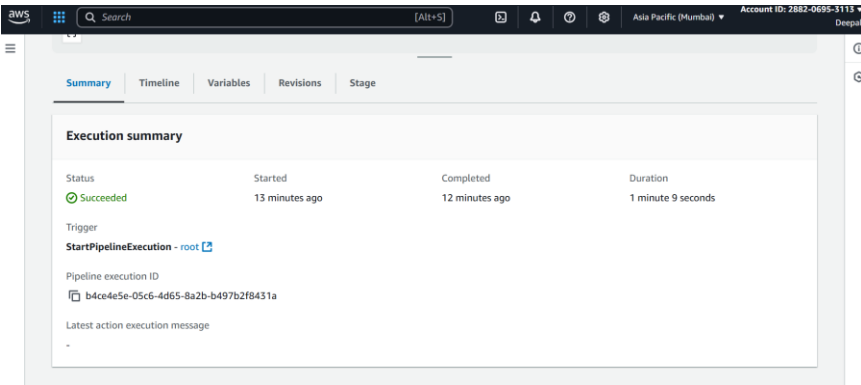
Screenshot 15

Network performance



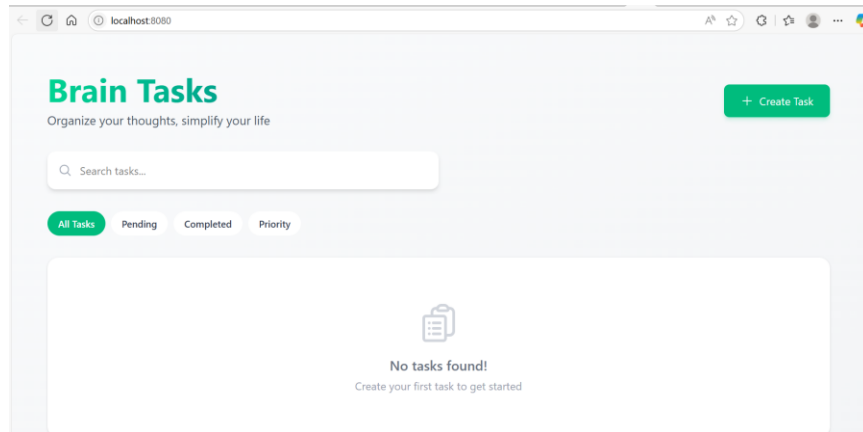
Screenshot 16

Pipeline execution summary



Screenshot 17

React app loaclly



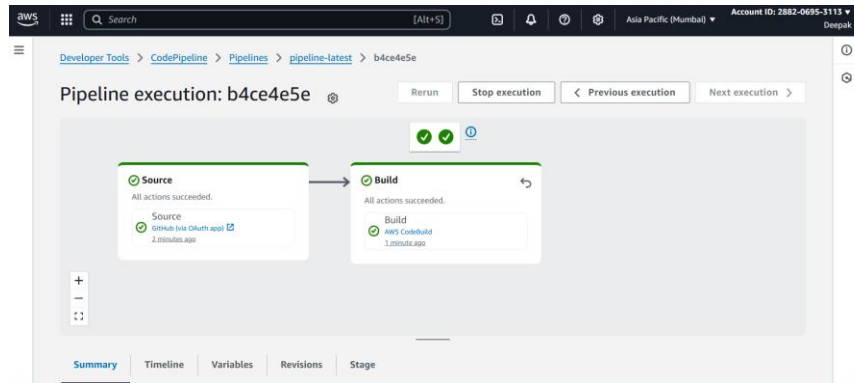
Screenshot 18

Service Exposure

```
EADwdr222868~SCD2288DXL MINGW64 ~/Documents/Guvi/Brain-Tasks-App (main)
$ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP
brain-tasks-service                 LoadBalancer        10.100.183.157   a79af09ed54224ba3b4a9bc8c9bcfc41-608068123.ap-south-1.elb.amazonaws.com
kubernetes                          ClusterIP            10.100.0.1       <none>
trend-service                       LoadBalancer        10.100.18.164    aa382ab88486b40ea8db56fd257aa0f0-1581967591.ap-south-1.elb.amazonaws.com
```

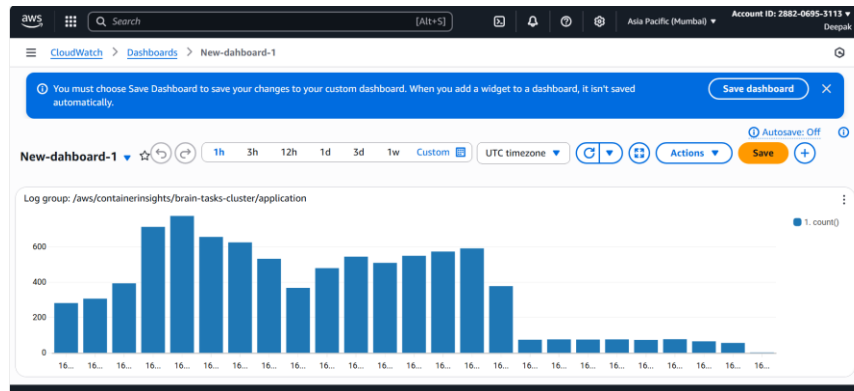
Screenshot 19

code pipeline



Screenshot 20

error occurrence of app per min



Screenshot 21

memory usage

