Project 1: Application Deployment

(Deploy the given React application to a production ready state)

Application:

Clone the below mentioned repository and deploy the application (Run application in port 3000).

Repo URL : https://github.com/Vennilavan12/Brain-Tasks-App.git

Docker:

- Dockerize the application by creating Dockerfile
- Build an application and check output using docker image.

ECR:

- Create an AWS ECR repository for store docker images.

Kubernetes:

- Setup Kubernetes in AWS EKS and Confirm EKS cluster is running.
- Write deployment and service YAML files.
- Deploy using kubectl via Codedeploy.

CodeBuild:

- Create a CodeBuild project:
- Source: Connect to your repository
- Environment: Use managed image (Amazon Linux, Ubuntu)
- Write and define commands in buildspec.yml.

CodeDeploy:

- Create codedeploy application.
- create appspec.yml file to deploy applications in EKS.

Version Control:

- Push the codebase to a Git provider (GitHub).
- Use CLI commands to push code.

CodePipeline:

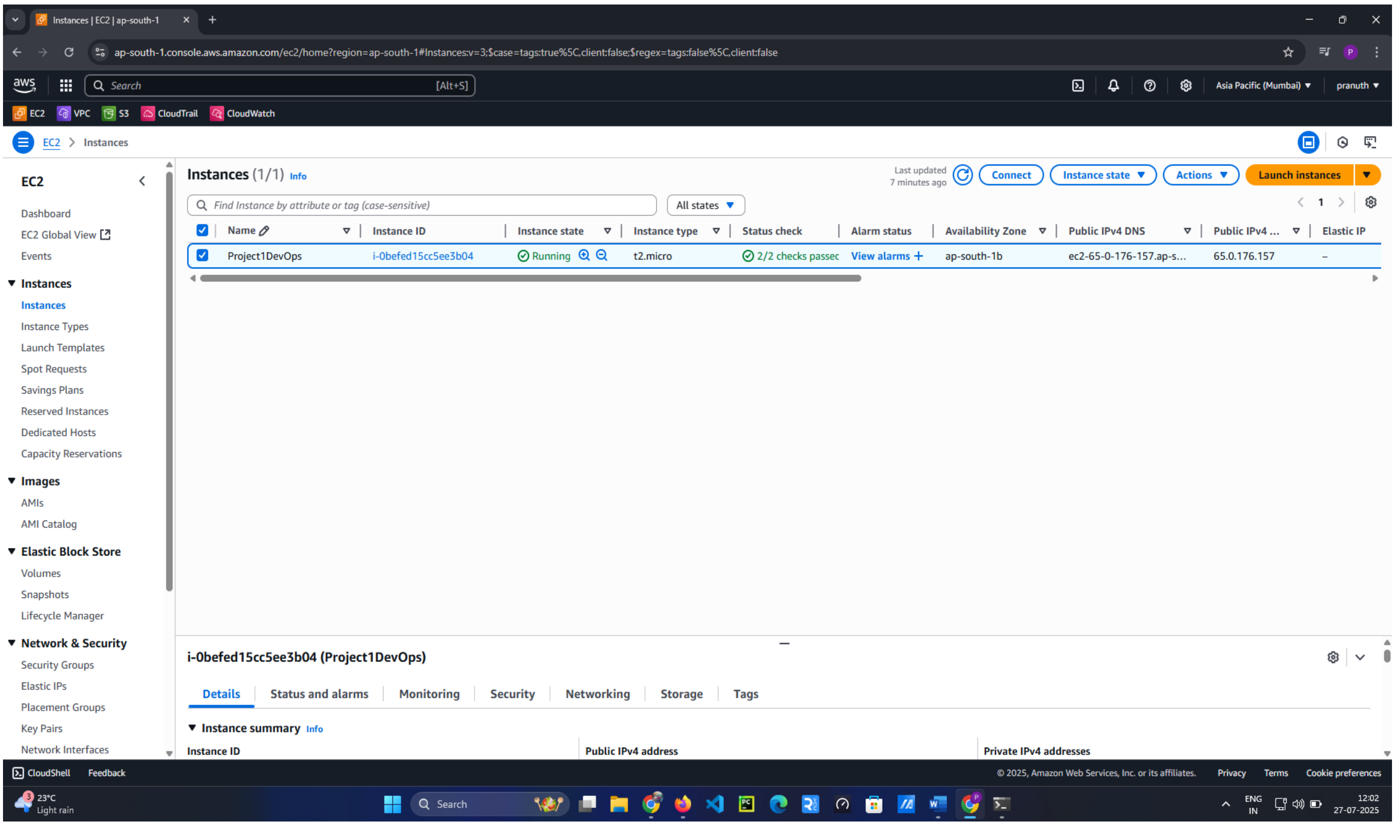
- Source: GitHub
- Build: AWS CodeBuild project
- Deploy: AWS CodeDeploy or deploy to EKS via Lambda or custom script.

Monitoring:

- Use CloudWatch Logs to track build, deploy, and application logs.

Project Solution:

Step1: Launch Ubuntu EC2 Instance



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Project1DevOps	i-0befed15cc5ee3b04	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-65-0-176-157.ap-s...	65.0.176.157	–

i-0befed15cc5ee3b04 (Project1DevOps)

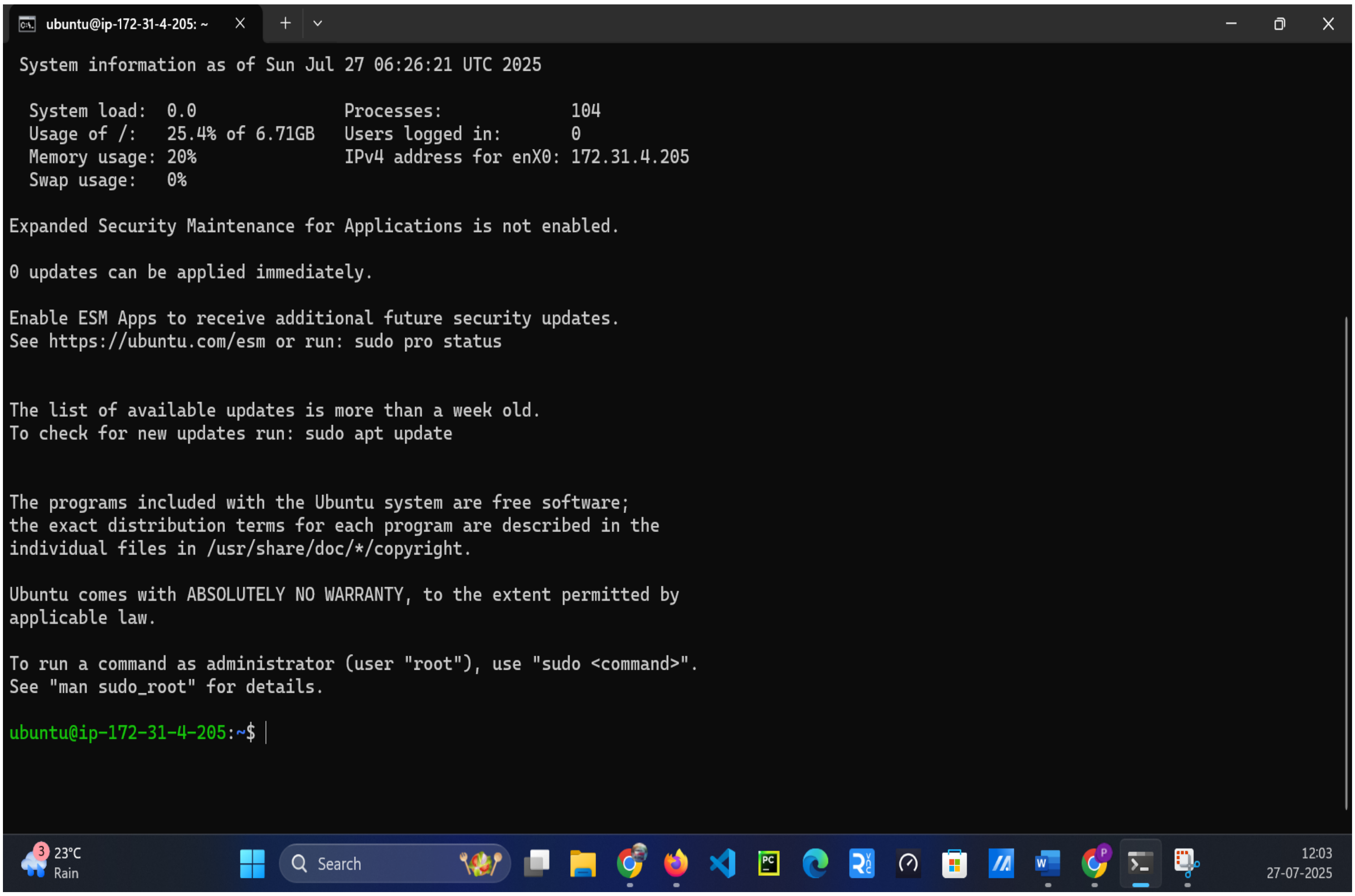
Instance summary Info

Instance ID

Public IPv4 address

Private IPv4 addresses

Step2: Connect the Ubuntu VM via ssh to local terminal



```
 System information as of Sun Jul 27 06:26:21 UTC 2025

  System load:  0.0               Processes:             104
  Usage of /:   25.4% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.4.205
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-4-205:~$
```

Step 3: Install Prerequisites

i. *sudo apt-get update && sudo apt-get upgrade -y*
ii. *sudo apt-get install git -y*
iii. *sudo apt install -y docker.io*
iv. *sudo usermod -aG docker ubuntu*
v. *sudo apt install -y nginx*
vi. *sudo apt install ruby-full*
vii. *sudo apt install wget*
viii. *cd /home/ubuntu*
ix. *wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install*
x. *chmod +x ./install*
xi. *sudo ./install auto*
xii. *systemctl status codedeploy-agent*

Step 4: Clone the Application Repository

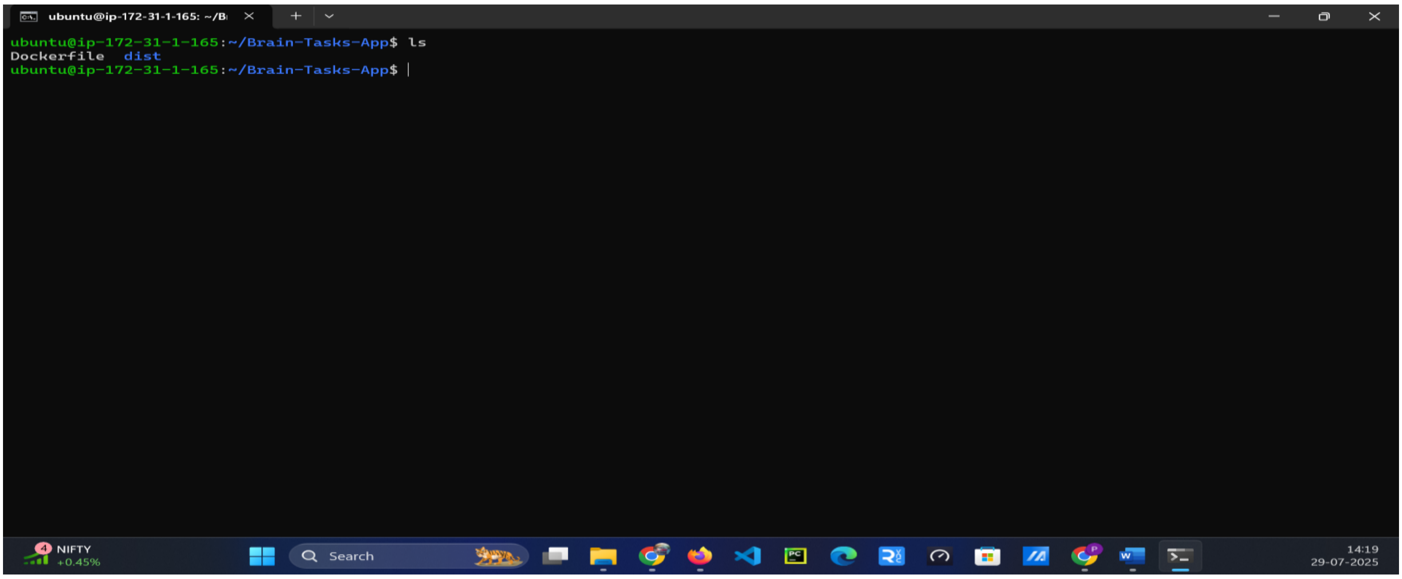
i. *git clone https://github.com/Vennilavan12/Brain-Tasks-App.git*
ii. *cd Brain-Tasks-App*

Step5: Dockerize the Application

Docker File code:

```
# Use the official NGINX image

FROM nginx:alpine

# Copy static site to NGINX public directory

COPY dist/ /usr/share/nginx/html

# Expose port 80

EXPOSE 80

# Start NGINX

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

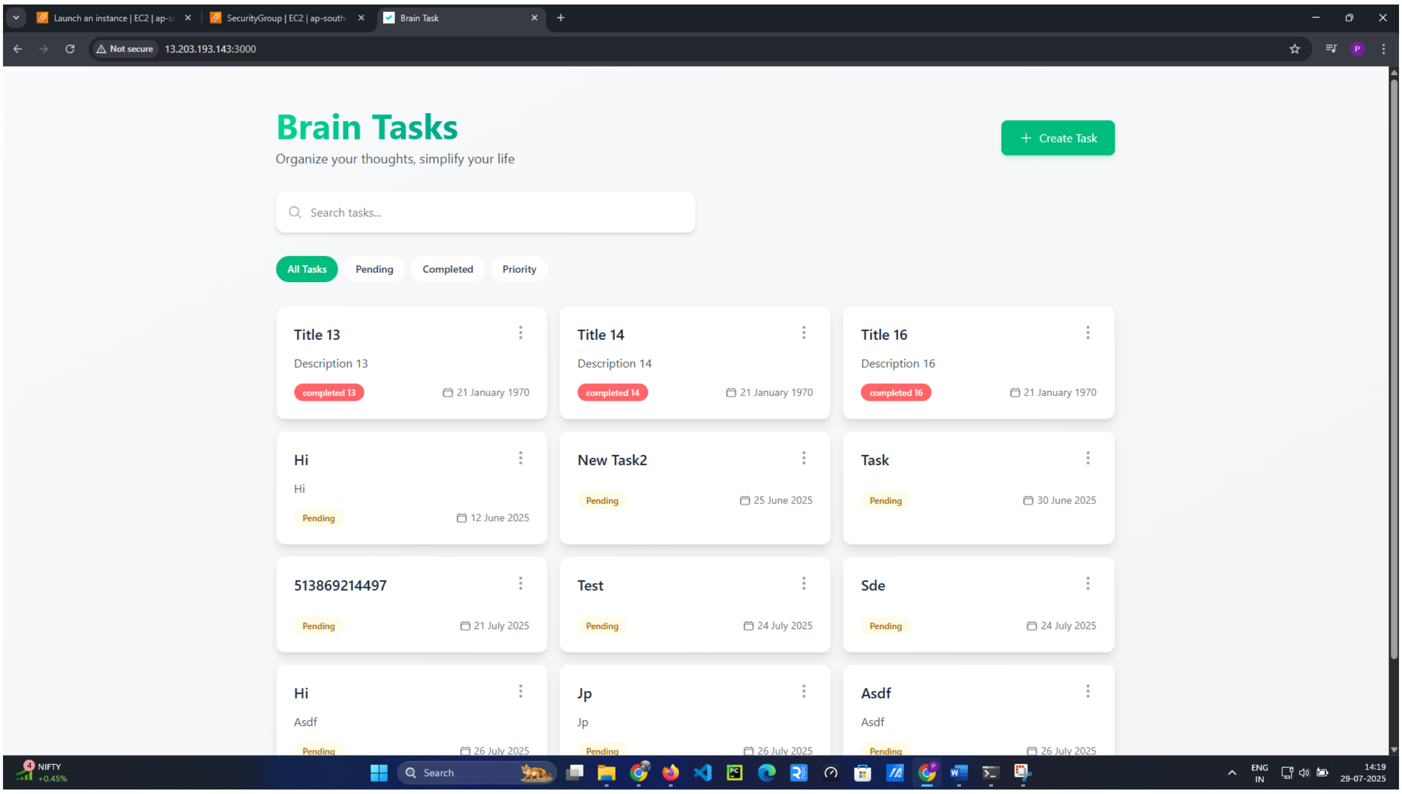
In project root (Brain-Tasks-App/), add a Dockerfile:



```
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ ls
Dockerfile  dist
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ |
```

Step6: Build and Test Docker Image

i. *docker build -t brain-tasks-app .*
ii. *docker run --rm -d -p 3000:80 --name brain-tasks brain-tasks-app*
iii. *Now visit http://<EC2-PUBLIC-IP>:3000 static React app should load, as it is just serving the dist/ folder.*



Step 7 : Push the docker image to ECR

Before that configure AWS CLI in your instance

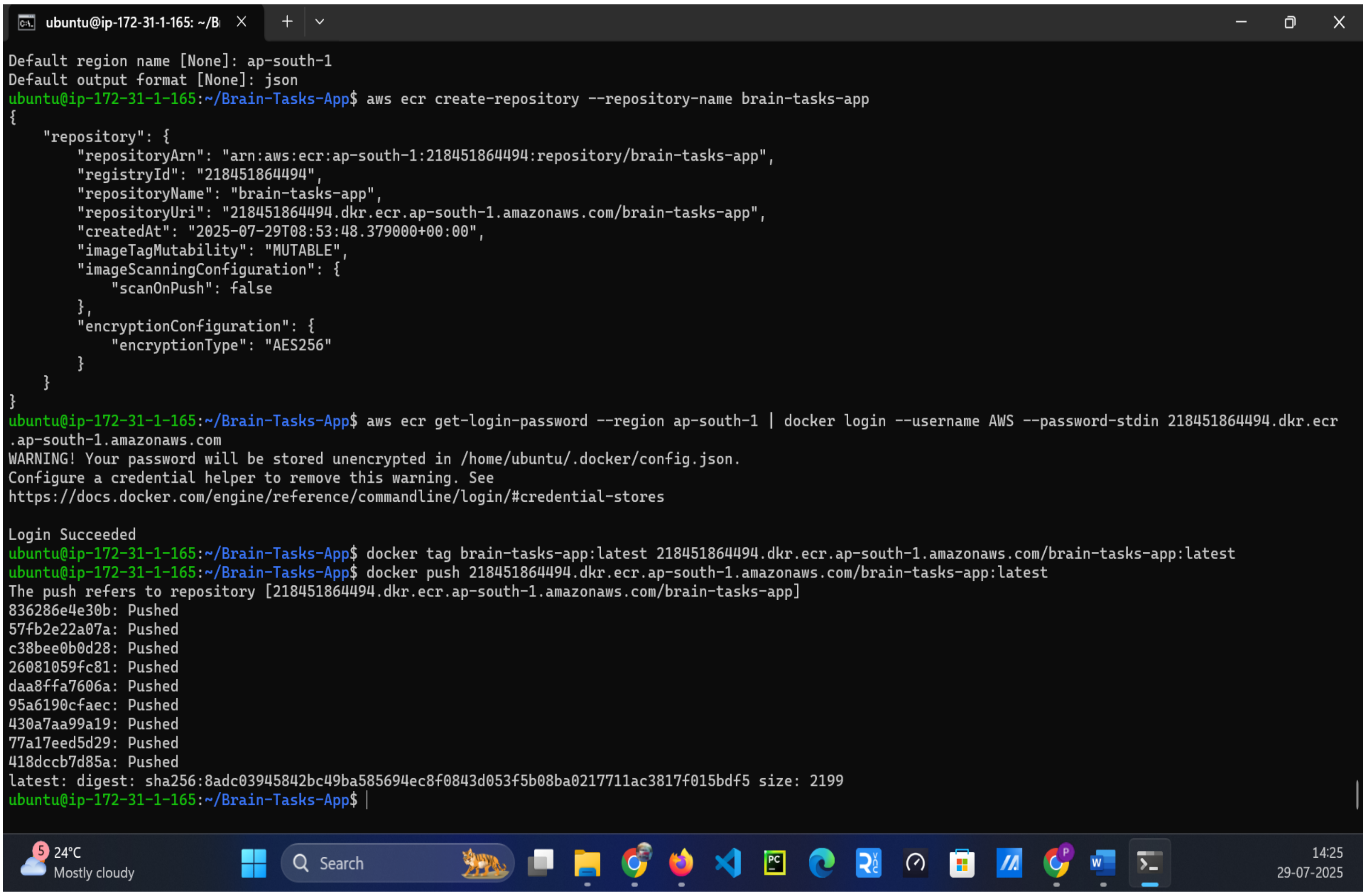
i. *sudo apt update && sudo apt upgrade -y*
ii. *sudo apt install unzip curl -y*
iii. *curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"*
iv. *unzip awscliv2.zip*
v. *sudo ./aws/install*
vi. *aws –version*

Authenticate Docker with ECR

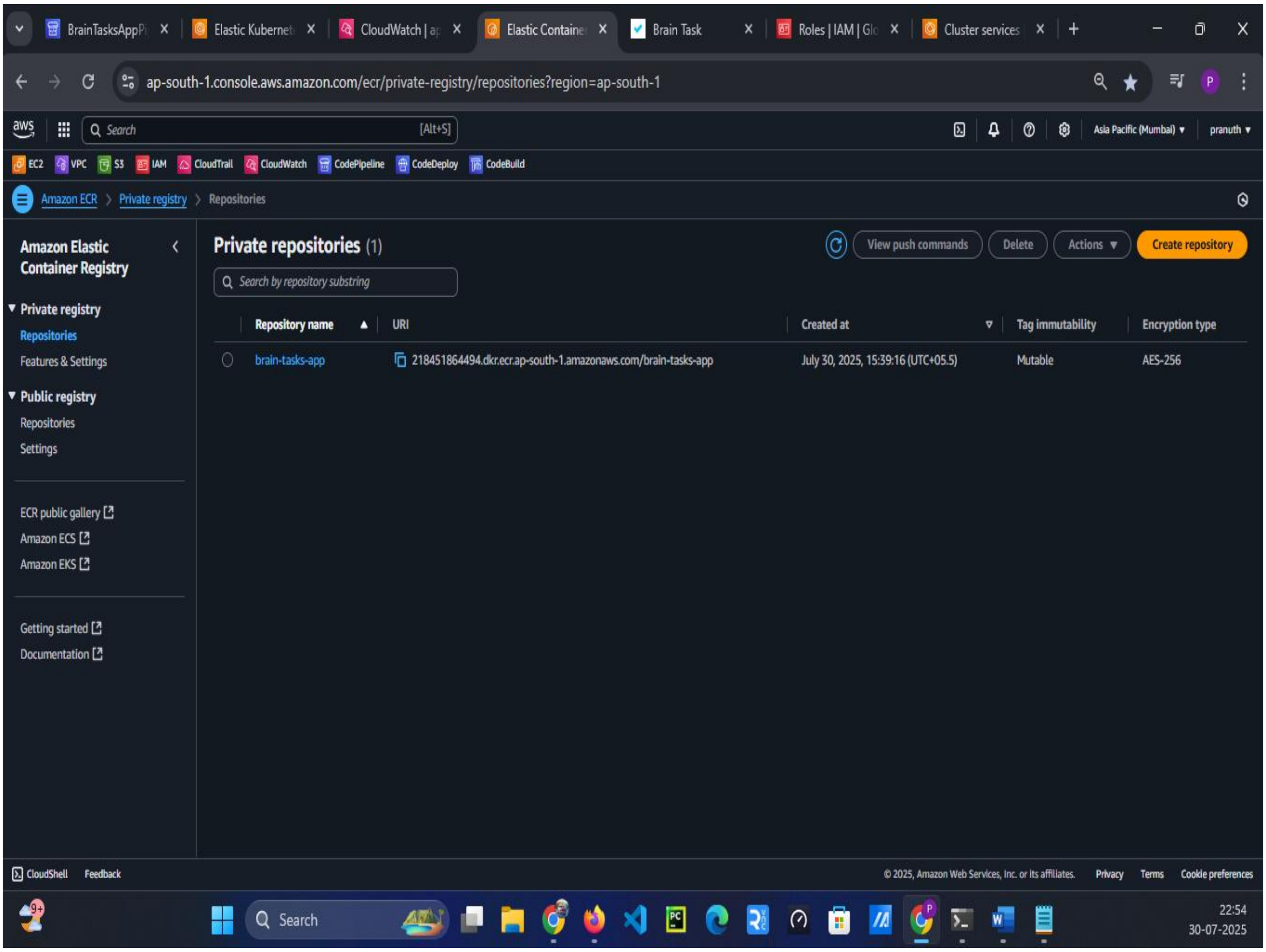
i. *aws configure*
ii. *aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin $Account_ID.dkr.ecr.ap-south-1.amazonaws.com*

Then run below commands to Tag and Push :

i. *aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 218451864494.dkr.ecr.ap-south-1.amazonaws.com.*
ii. *docker tag brain-tasks-app:latest 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest*
iii. *docker push 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest*

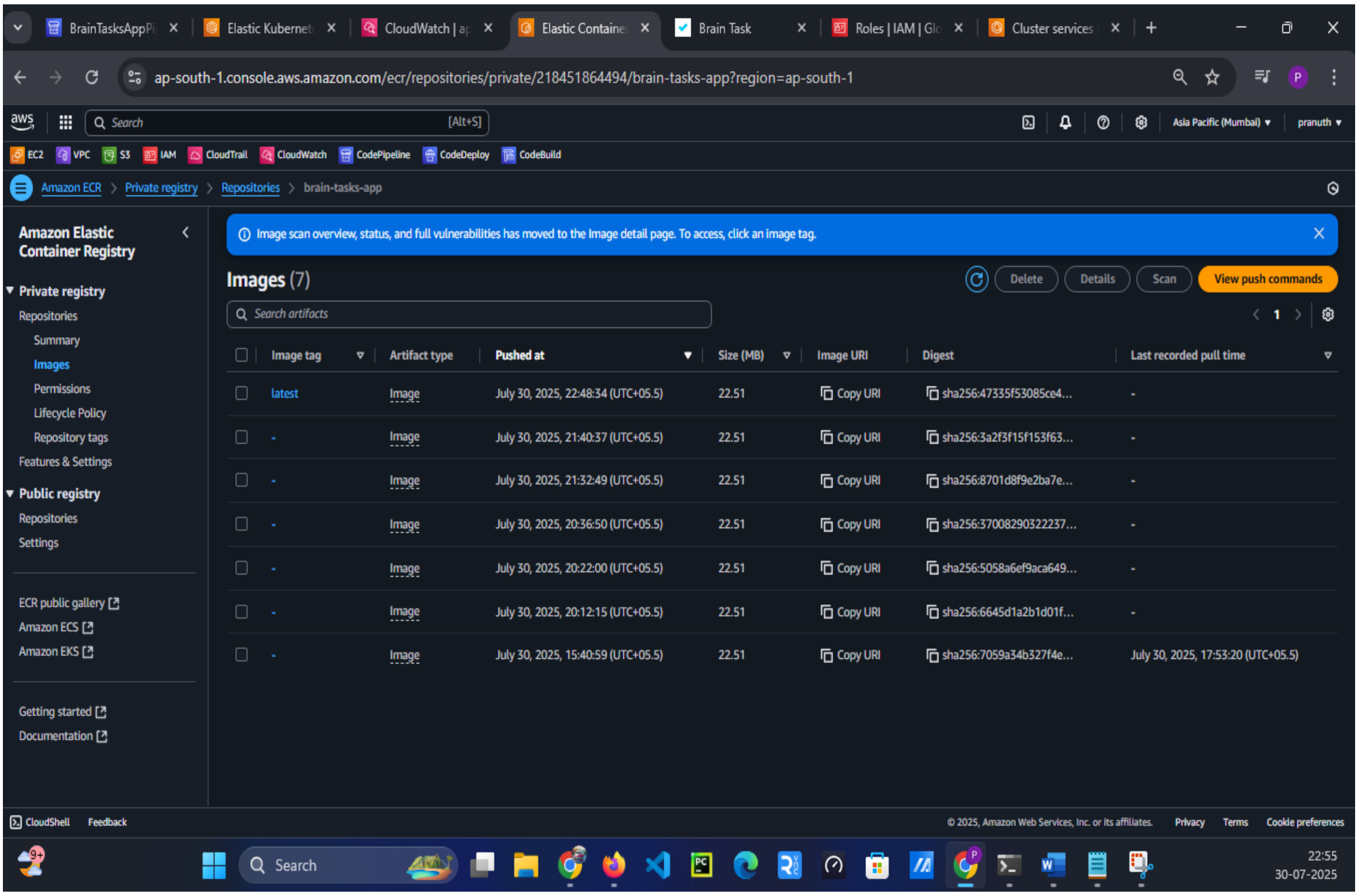


```
Default region name [None]: ap-south-1
Default output format [None]: json
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ aws ecr create-repository --repository-name brain-tasks-app
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:ap-south-1:218451864494:repository/brain-tasks-app",
        "registryId": "218451864494",
        "repositoryName": "brain-tasks-app",
        "repositoryUri": "218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app",
        "createdAt": "2025-07-29T08:53:48.379000+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 218451864494.dkr.ecr
.ap-south-1.amazonaws.com
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-172-31-1-165:~/Brain-Tasks-App$ docker tag brain-tasks-app:latest 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ docker push 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
The push refers to repository [218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app]
836286e4e30b: Pushed
57fb2e22a07a: Pushed
c38bee0b0d28: Pushed
26081059fc81: Pushed
daa8ffa7606a: Pushed
95a6190cfaec: Pushed
430a7aa99a19: Pushed
77a17eed5d29: Pushed
418dccb7d85a: Pushed
latest: digest: sha256:8adc03945842bc49ba585694ec8f0843d053f5b08ba0217711ac3817f015bdf5 size: 2199
ubuntu@ip-172-31-1-165:~/Brain-Tasks-App$ |
```



Private repositories (1)

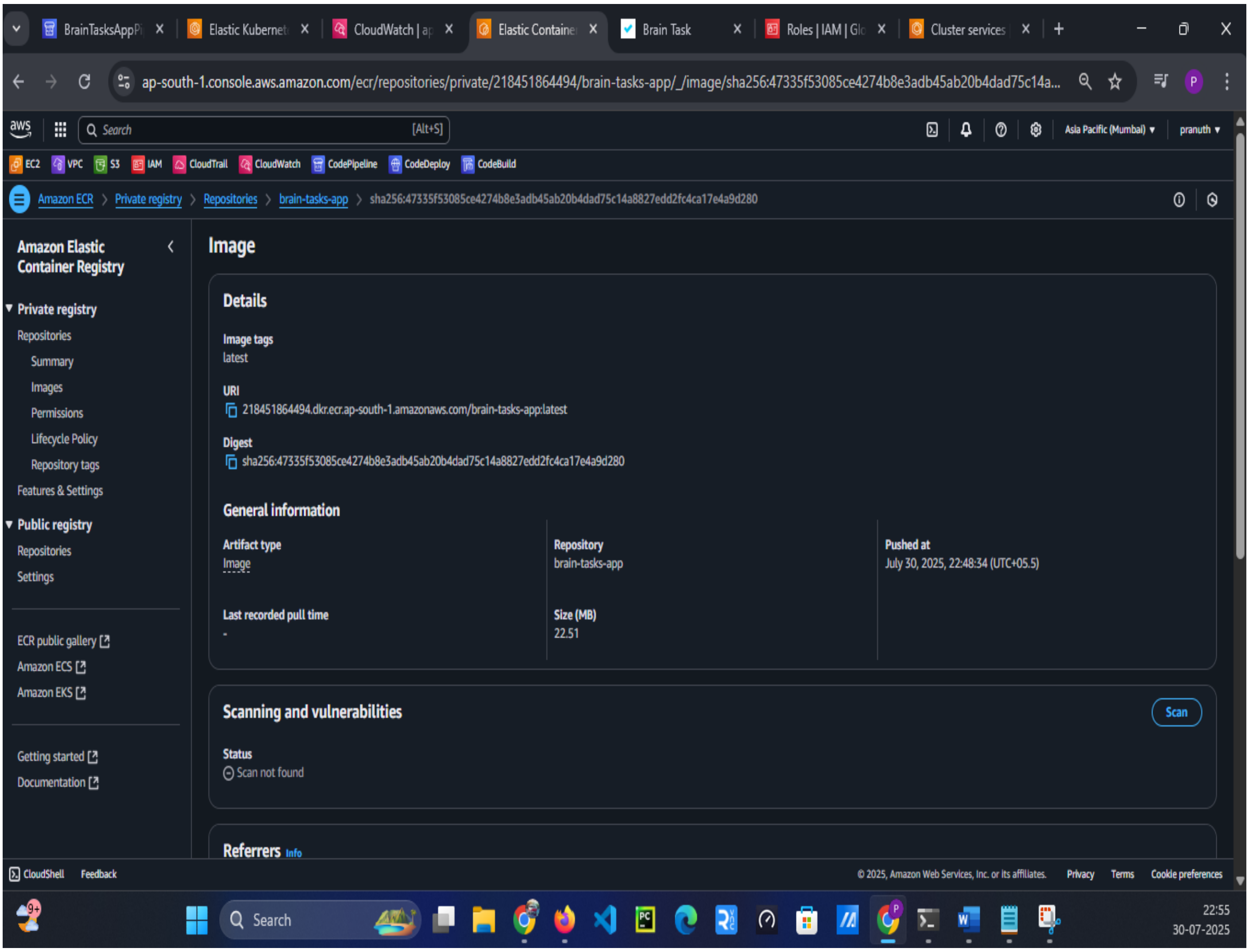
	Repository name ▲	URI	Created at ▿	Tag immutability	Encryption type
○	brain-tasks-app	218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app	July 30, 2025, 15:39:16 (UTC+05.5)	Mutable	AES-256



ⓘ Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (7)

□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
□	latest	Image	July 30, 2025, 22:48:34 (UTC+05.5)	22.51	Copy URI	sha256:47335f53085ce4...	-
□	-	Image	July 30, 2025, 21:40:37 (UTC+05.5)	22.51	Copy URI	sha256:3a2f3f15f153f63...	-
□	-	Image	July 30, 2025, 21:32:49 (UTC+05.5)	22.51	Copy URI	sha256:8701d8f9e2ba7e...	-
□	-	Image	July 30, 2025, 20:36:50 (UTC+05.5)	22.51	Copy URI	sha256:37008290322237...	-
□	-	Image	July 30, 2025, 20:22:00 (UTC+05.5)	22.51	Copy URI	sha256:5058a6ef9aca649...	-
□	-	Image	July 30, 2025, 20:12:15 (UTC+05.5)	22.51	Copy URI	sha256:6645d1a2b1d01f...	-
□	-	Image	July 30, 2025, 15:40:59 (UTC+05.5)	22.51	Copy URI	sha256:7059a34b327f4e...	July 30, 2025, 17:53:20 (UTC+05.5)



Image

Details

Image tags
latest

URI
218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest

Digest
sha256:47335f53085ce4274b8e3adb45ab20b4dad75c14a8827edd2fc4ca17e4a9d280

General information

Artifact type
Image

Repository
brain-tasks-app

Pushed at
July 30, 2025, 22:48:34 (UTC+05.5)

Last recorded pull time
-

Size (MB)
22.51

Scanning and vulnerabilities

Status
⊝ Scan not found

Referrers Info

Step 8: Create Kubernetes (EKS) Cluster:

A. Install eksctl

i.
```
curl -LO https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz
```

ii.
```
tar -xzf eksctl_Linux_amd64.tar.gz
```

iii.
```
sudo mv eksctl /usr/local/bin
```

iv.
```
eksctl version
```

B. Install kubectl (Kubernetes CLI)

v.
```
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
```

vi.
```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

vii.
```
kubectl version –client
```

C. Install AWS IAM Authenticator

viii.
```
curl -o aws-iam-authenticator https://amazon-eks.s3.us-west-2.amazonaws.com/1.15.10/2020-02-22/bin/linux/amd64/aws-iam-authenticator
```

ix.
```
chmod +x ./aws-iam-authenticator
```

x.
```
sudo mv ./aws-iam-authenticator /usr/local/bin
```

D.Create a Cluster:

xi.
```
eksctl create cluster --name brain-app-cluster --region ap-south-1 --nodegroup-name standard-workers --node-type t3.medium --nodes 1
```

xii.
```
eksctl get cluster --region ap-south-1
```

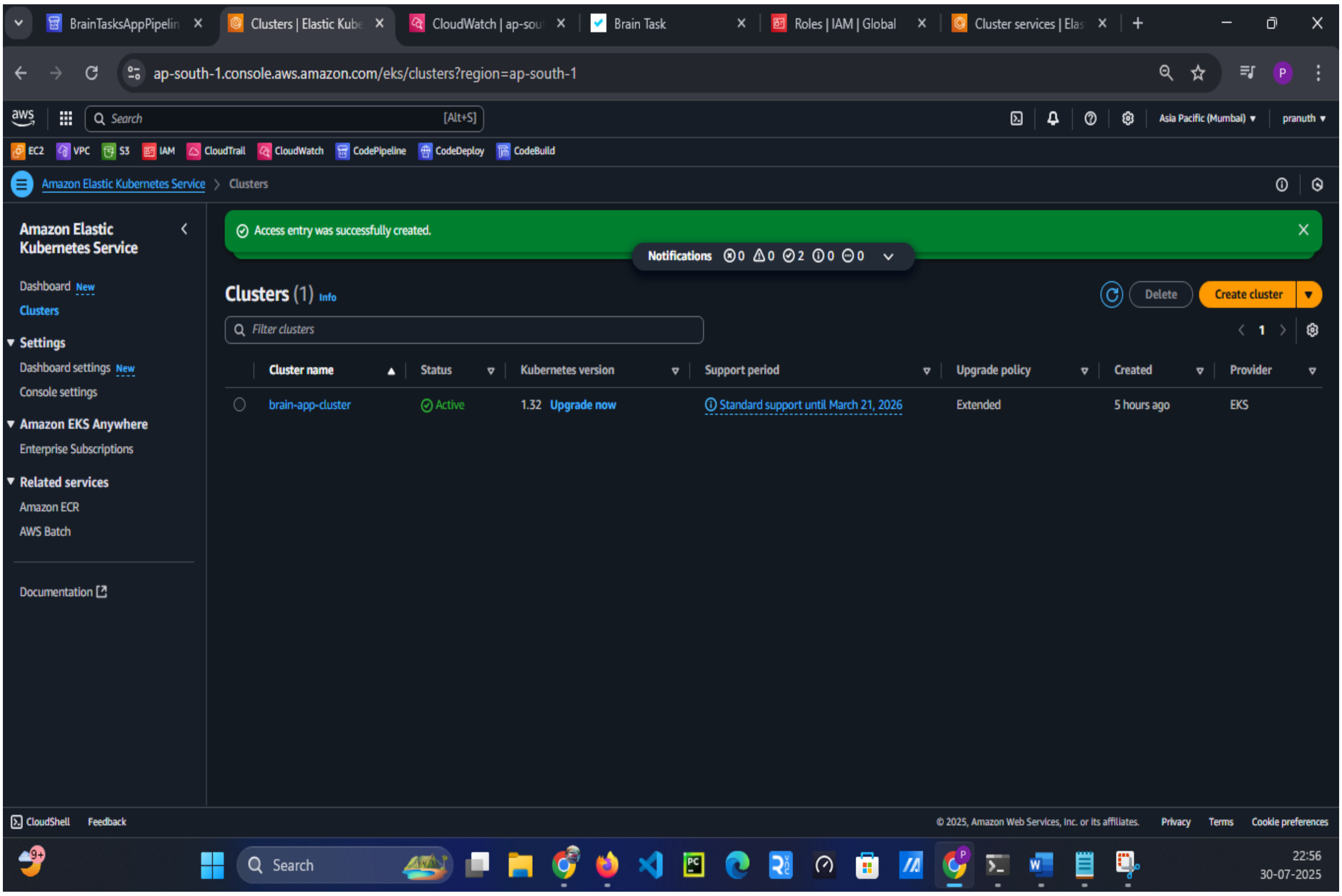
- This command creates:
  - EKS cluster named brain-tasks-eks
  - 1 EC2 worker nodes (t3.medium)
  - Configures everything like (VPC, IAM, etc.)

E. Update Kubeconfig

xiii.
```
aws eks update-kubeconfig --region ap-south-1 --name brain-app-cluster
```

F. Validate Cluster Connection

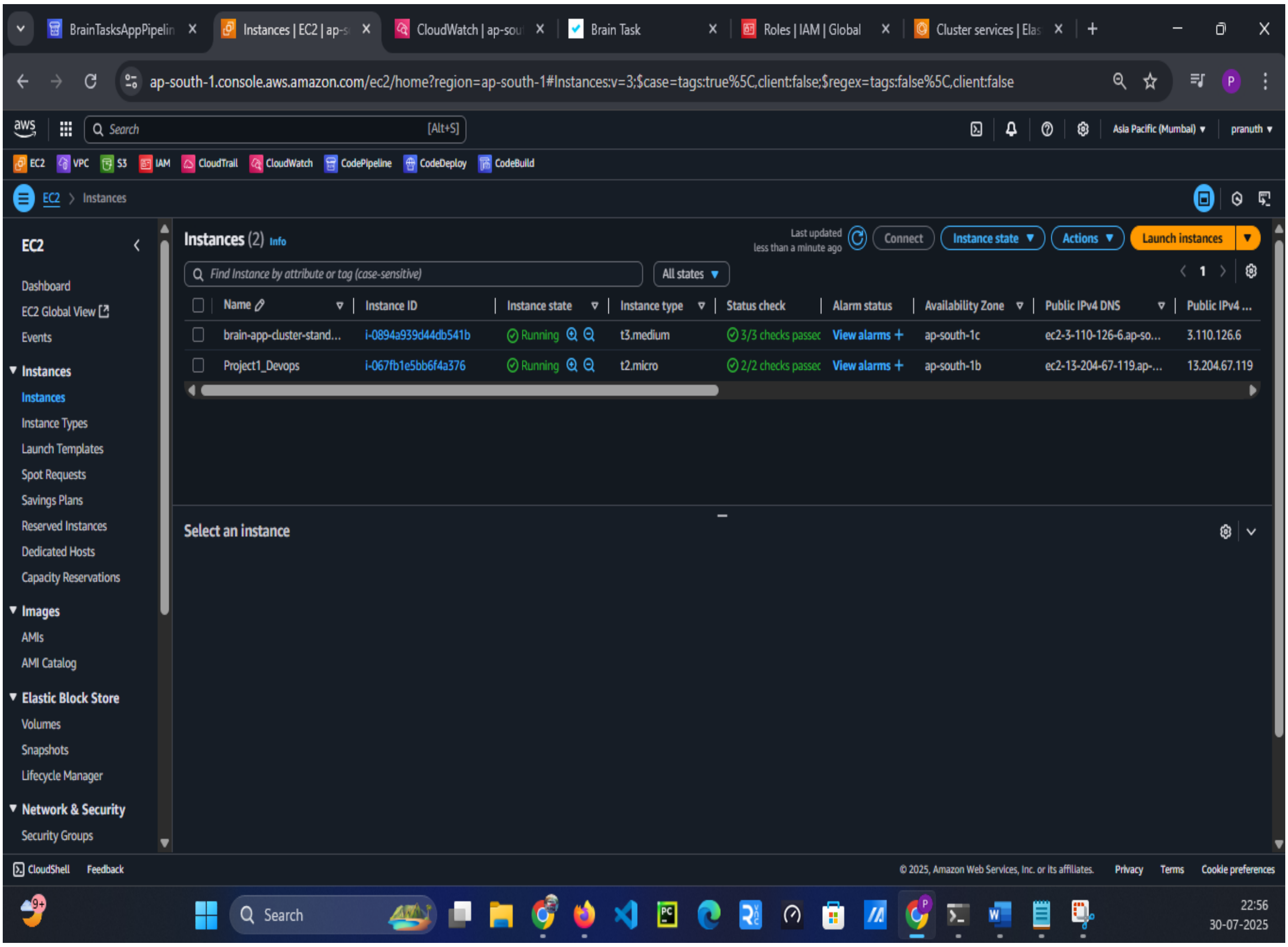
xiv.
```
kubectl get nodes
```



✓ Access entry was successfully created.

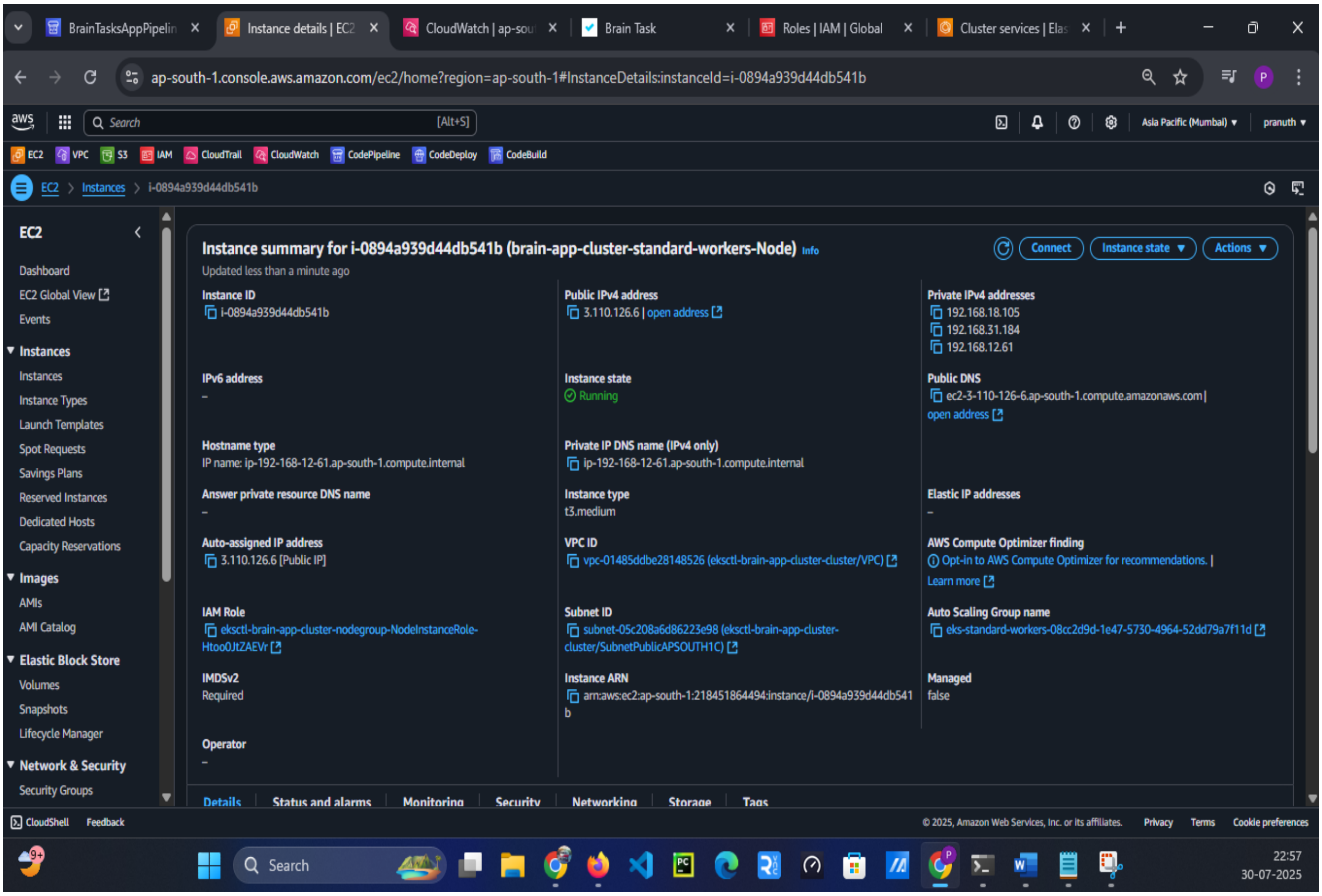
Clusters (1) Info

	Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
○	brain-app-cluster	✓ Active	1.32 Upgrade now	ⓘ Standard support until March 21, 2026	Extended	5 hours ago	EKS



Instances (2) Info

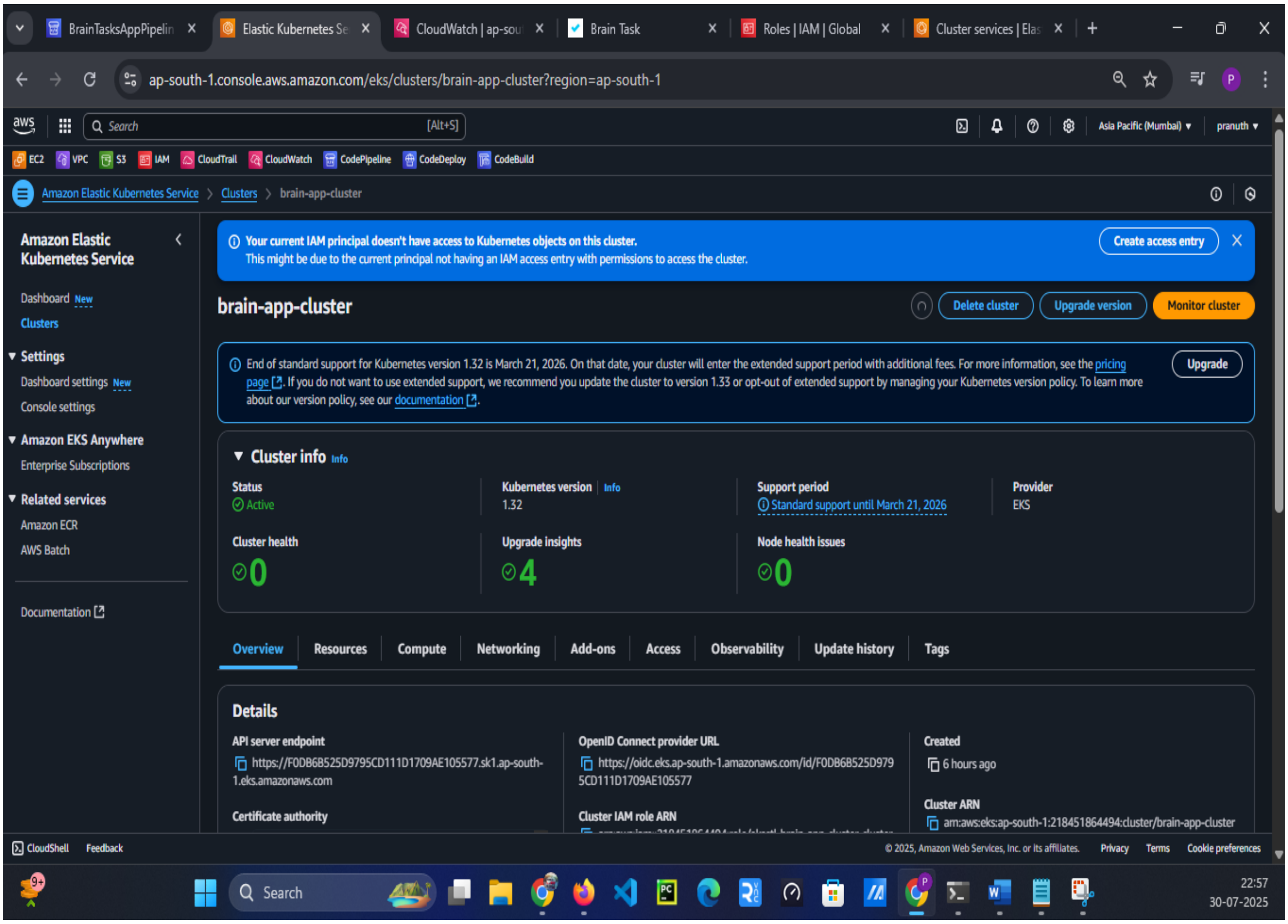
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
□	brain-app-cluster-stand...	i-0894a939d44db541b	✓ Running	t3.medium	✓ 3/3 checks passec	View alarms +	ap-south-1c	ec2-3-110-126-6.ap-so...	3.110.126.6
□	Project1_Devops	i-067fb1e5bb6f4a376	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-13-204-67-119.ap-...	13.204.67.119



Instance summary for i-0894a939d44db541b (brain-app-cluster-standard-workers-Node) Info

Updated less than a minute ago

Instance ID i-0894a939d44db541b	Public IPv4 address 3.110.126.6 | open address	Private IPv4 addresses 192.168.18.105 192.168.31.184 192.168.12.61
IPv6 address –	Instance state Running	Public DNS ec2-3-110-126-6.ap-south-1.compute.amazonaws.com | open address
Hostname type IP name: ip-192-168-12-61.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-192-168-12-61.ap-south-1.compute.internal	
Answer private resource DNS name –	Instance type t3.medium	Elastic IP addresses –
Auto-assigned IP address 3.110.126.6 [Public IP]	VPC ID vpc-01485ddbe28148526 (eksctl-brain-app-cluster-cluster/VPC)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role eksctl-brain-app-cluster-nodegroup-NodeInstanceRole-Htoo0JtZAEVr	Subnet ID subnet-05c208a6d86223e98 (eksctl-brain-app-cluster-cluster/SubnetPublicAPSOUTH1C)	Auto Scaling Group name eks-standard-workers-08cc2d9d-1e47-5730-4964-52dd79a7f11d
IMDSv2 Required	Instance ARN arn:aws:ec2:ap-south-1:218451864494:instance/i-0894a939d44db541b	Managed false
Operator –		



Your current IAM principal doesn't have access to Kubernetes objects on this cluster.
This might be due to the current principal not having an IAM access entry with permissions to access the cluster.

brain-app-cluster

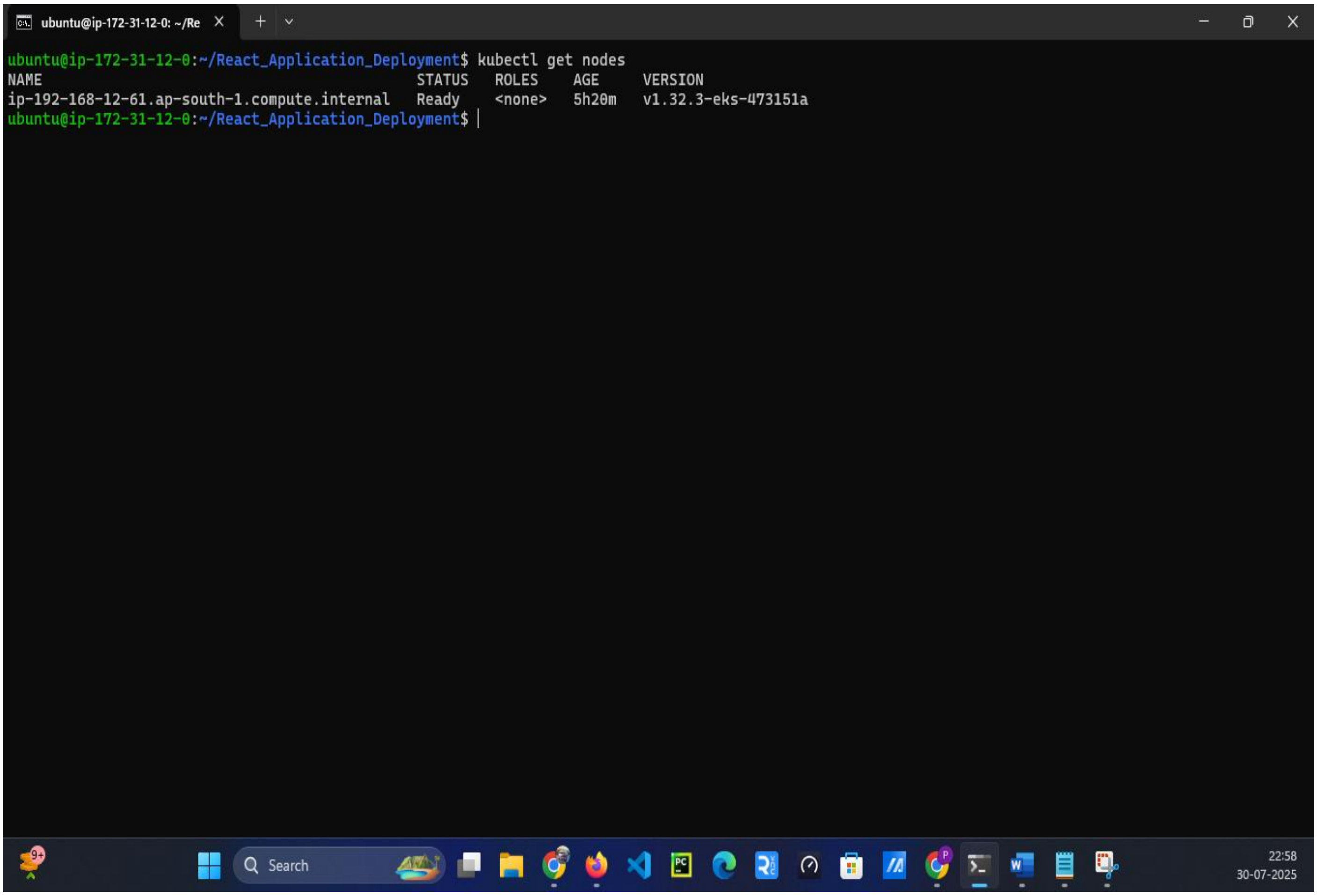
End of standard support for Kubernetes version 1.32 is March 21, 2026. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page. If you do not want to use extended support, we recommend you update the cluster to version 1.33 or opt-out of extended support by managing your Kubernetes version policy. To learn more about our version policy, see our documentation.

▾ Cluster info Info

Status Active	Kubernetes version Info 1.32	Support period Standard support until March 21, 2026	Provider EKS
Cluster health 0	Upgrade insights 4	Node health issues 0	

Details

API server endpoint https://F0DB6B525D9795CD111D1709AE105577.sk1.ap-south-1.eks.amazonaws.com	OpenID Connect provider URL https://oidc.eks.ap-south-1.amazonaws.com/id/F0DB6B525D9795CD111D1709AE105577	Created 6 hours ago
Certificate authority	Cluster IAM role ARN	Cluster ARN arn:aws:eks:ap-south-1:218451864494:cluster/brain-app-cluster



```
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ kubectl get nodes
NAME                                         STATUS   ROLES    AGE     VERSION
ip-192-168-12-61.ap-south-1.compute.internal   Ready    <none>   5h20m   v1.32.3-eks-473151a
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ |
```

9.Kubernetes Deployment and Service YAML

a. Create an ImagePullSecret:

If your EKS worker nodes do not have an IAM role that allows them to pull images from your ECR repository, you'll need an ImagePullSecret. It's generally better to grant the EKS node IAM role permissions to ECR, but this secret method works if the IAM role is not sufficient.

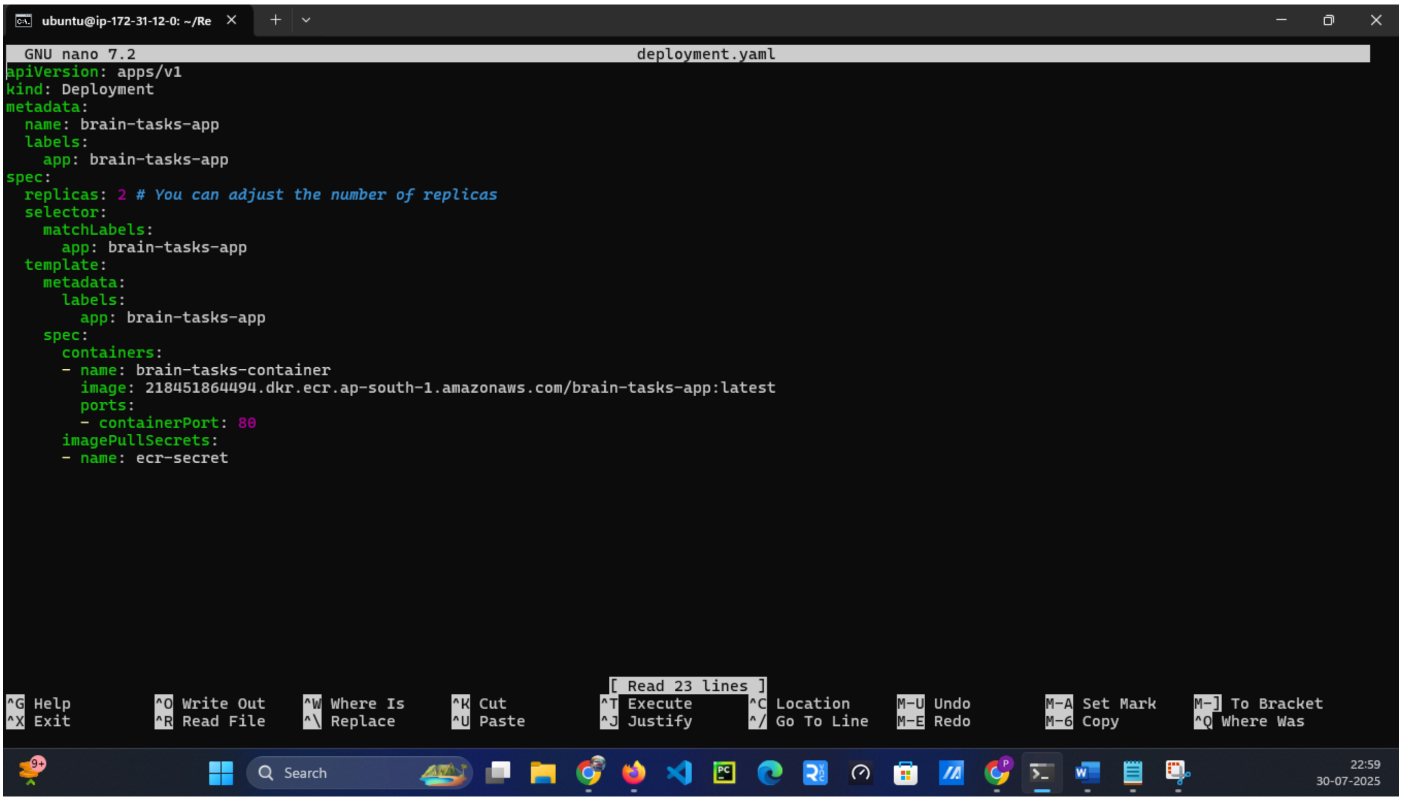
```
kubectl create secret docker-registry ecr-secret \
  --docker-server=<your-account-id>.dkr.ecr.ap-south-1.amazonaws.com \
  --docker-username=AWS \
  --docker-password="$(aws ecr get-login-password --region ap-south-1)" \
  --namespace=default
kubectl get secrets
```

b. Write Deployment YAML (deployment.yaml)

- Handles the creation of pods running Docker image.
- Replace **<aws-account-id>** and **<region>** with your actual AWS account info.

```
  apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-app
  labels:
    app: brain-tasks-app
spec:
  replicas: 2 # You can adjust the number of replicas
  selector:
    matchLabels:
      app: brain-tasks-app
  template:
    metadata:
      labels:
        app: brain-tasks-app
    spec:
      containers:
      - name: brain-tasks-container
        image: 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
        ports:
```

```
    - containerPort: 80
  imagePullSecrets:
  - name: ecr-secret
```

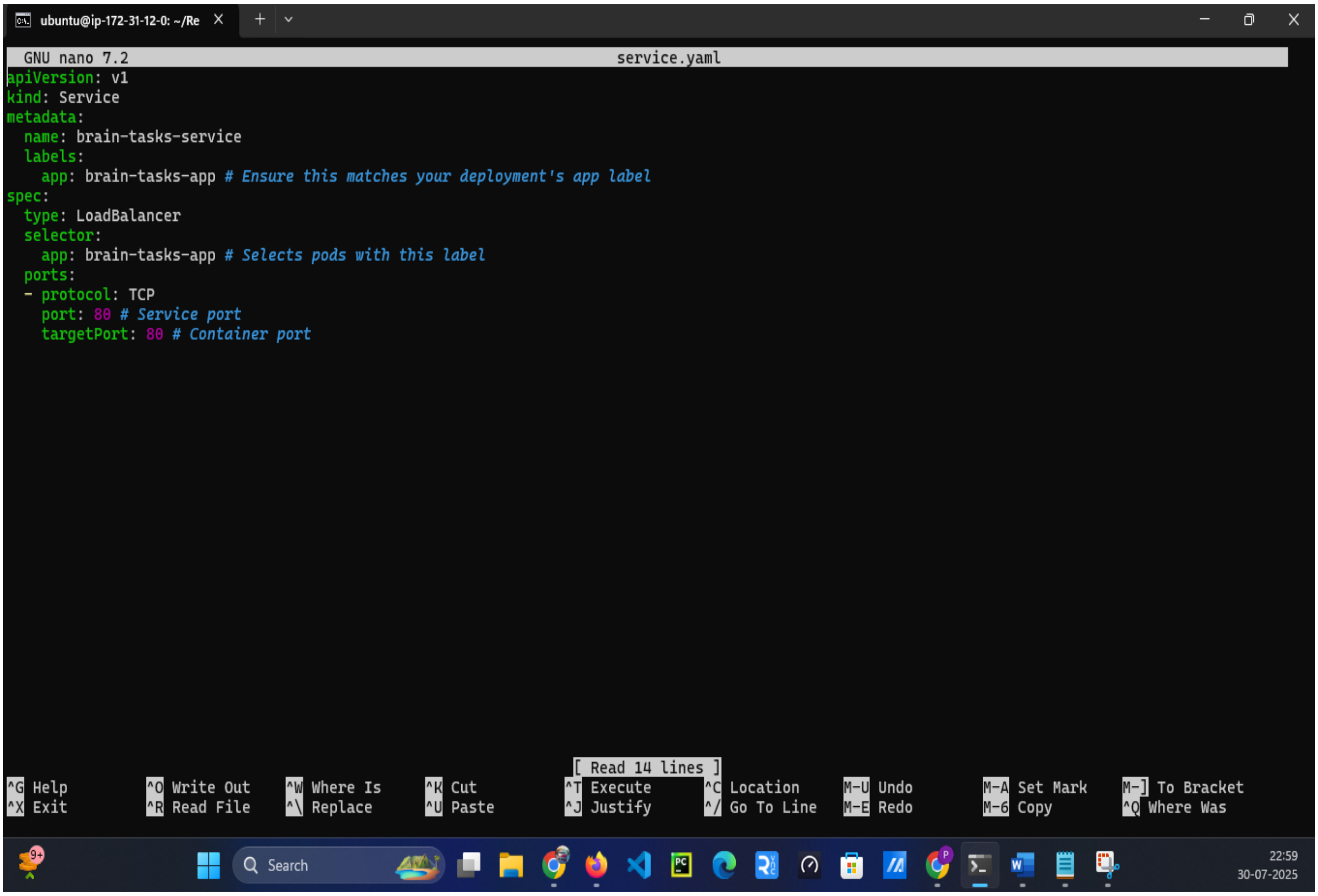


```
  GNU nano 7.2                                deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-app
  labels:
    app: brain-tasks-app
spec:
  replicas: 2 # You can adjust the number of replicas
  selector:
    matchLabels:
      app: brain-tasks-app
  template:
    metadata:
      labels:
        app: brain-tasks-app
    spec:
      containers:
      - name: brain-tasks-container
        image: 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
        ports:
        - containerPort: 80
      imagePullSecrets:
      - name: ecr-secret

                                      [ Read 23 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

c.Write Service YAML (service.yaml)

- Exposes deployment via a LoadBalancer service for public access.

```
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
  labels:
    app: brain-tasks-app # Ensure this matches your deployment's app label
spec:
  type: LoadBalancer
  selector:
    app: brain-tasks-app # Selects pods with this label
  ports:
  - protocol: TCP
    port: 80 # Service port
    targetPort: 80 # Container port
```



```
  GNU nano 7.2                                   service.yaml
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
  labels:
    app: brain-tasks-app # Ensure this matches your deployment's app label
spec:
  type: LoadBalancer
  selector:
    app: brain-tasks-app # Selects pods with this label
  ports:
  - protocol: TCP
    port: 80 # Service port
    targetPort: 80 # Container port

                                        [ Read 14 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

d.Apply the YAML Files

Deploy to your EKS cluster:

i. *kubectl apply -f deployment.yaml*

Command to expose application.

ii. *kubectl apply -f service.yaml*

To Check pods are running:

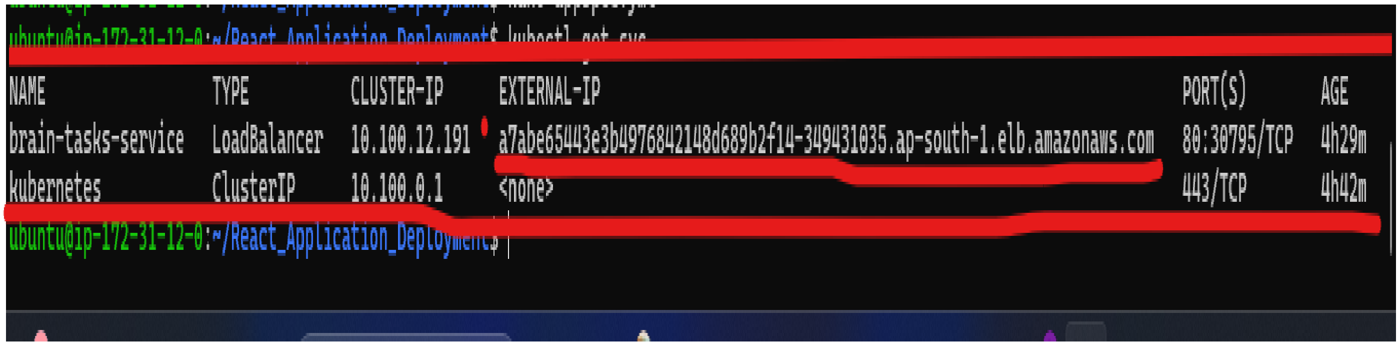
i. *kubectl get pods*

To Get the service details (and external IP):

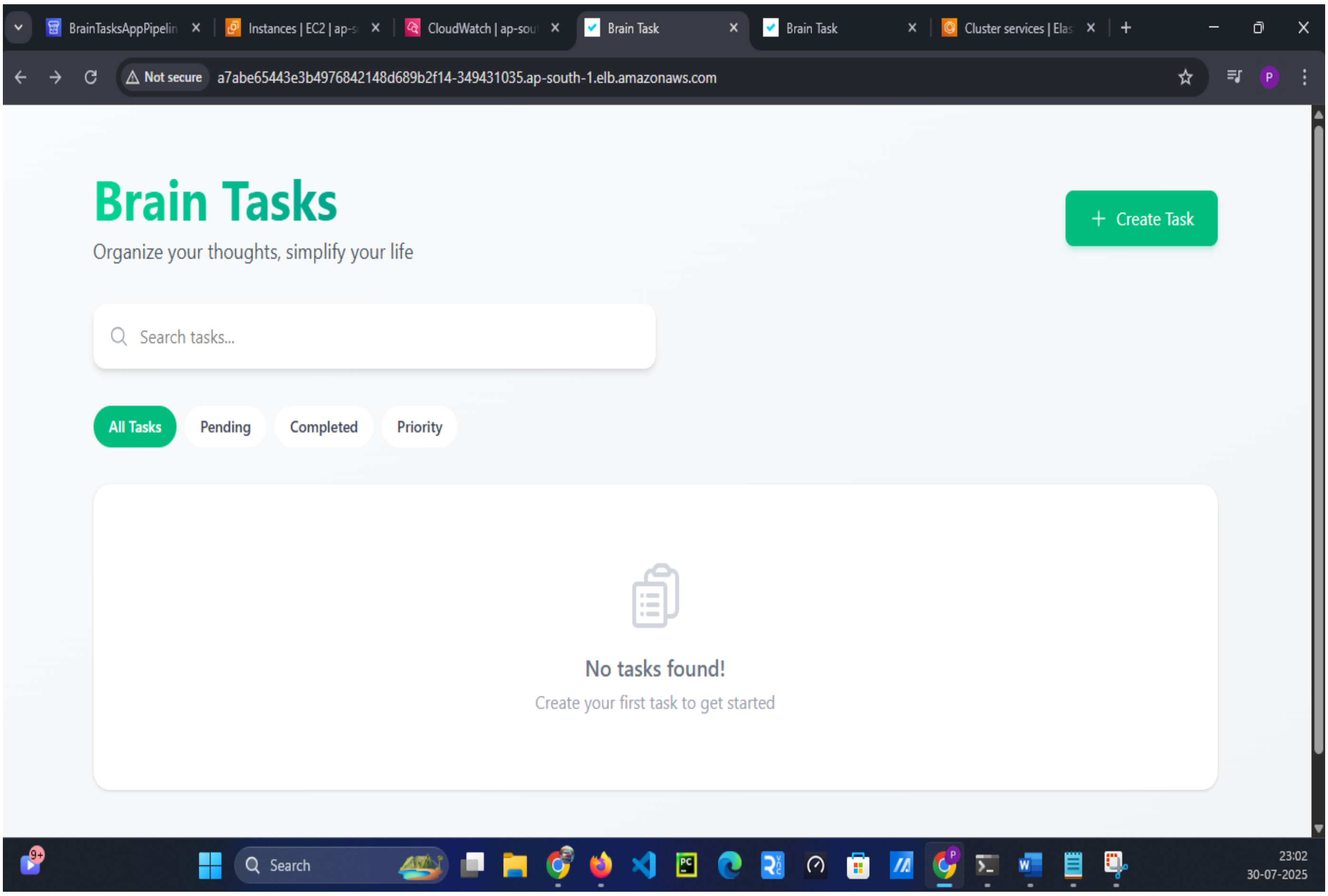
i. *kubectl get svc brain-tasks-service*

Get the External Load Balancer URL:

http://a7abe65443e3b4976842148d689b2f14-349431035.ap-south-1.elb.amazonaws.com/



```
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ kubectl get svc
NAME                  TYPE           CLUSTER-IP     EXTERNAL-IP                                                                 PORT(S)        AGE
brain-tasks-service   LoadBalancer   10.100.12.191  a7abe65443e3b4976842148d689b2f14-349431035.ap-south-1.elb.amazonaws.com    80:30795/TCP   4h29m
kubernetes            ClusterIP      10.100.0.1     <none>                                                                      443/TCP        4h42m
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$
```



Brain Tasks

Organize your thoughts, simplify your life

No tasks found!

Create your first task to get started

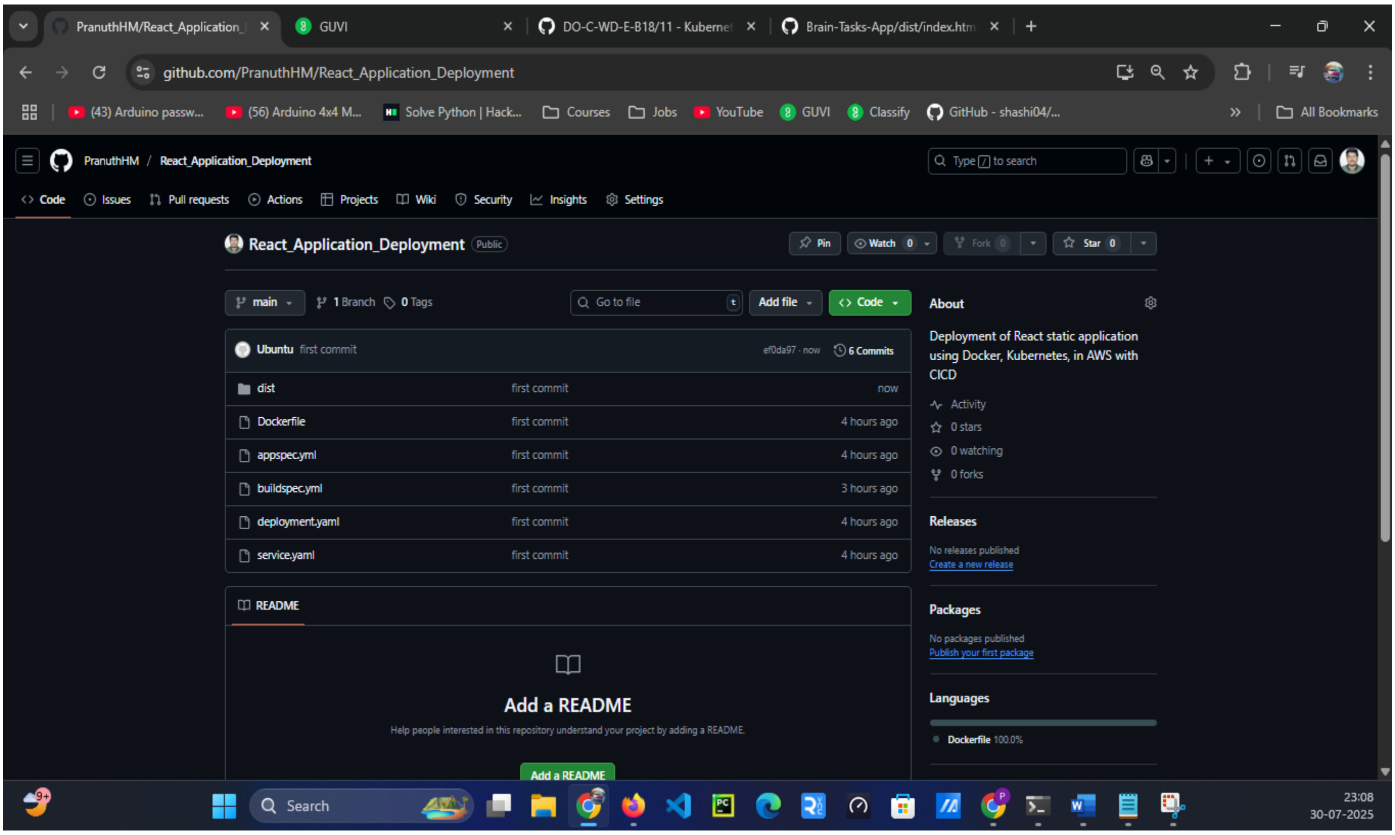
10. CI/CD Pipeline using AWS

AWS CodePipeline for React on EKS: CI/CD Setup Guide

This guide outlines the essential configurations for building a CI/CD pipeline using AWS CodePipeline to deploy this React application to an Amazon EKS cluster.

Prerequisites :

Upload Code: React app in a Git repository with Dockerfile, buildspec.yml, deployment.yaml, service.yaml at the root. i.e., https://github.com/PranuthHM/React_Application_Deployment.git



React_Application_Deployment Public

Ubuntu first commit ef0da97 · now 6 Commits

dist	first commit	now
Dockerfile	first commit	4 hours ago
appspec.yml	first commit	4 hours ago
buildspec.yml	first commit	3 hours ago
deployment.yaml	first commit	4 hours ago
service.yaml	first commit	4 hours ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

Deployment of React static application using Docker, Kubernetes, in AWS with CICD

Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Dockerfile 100.0%

Project Files (Overview)

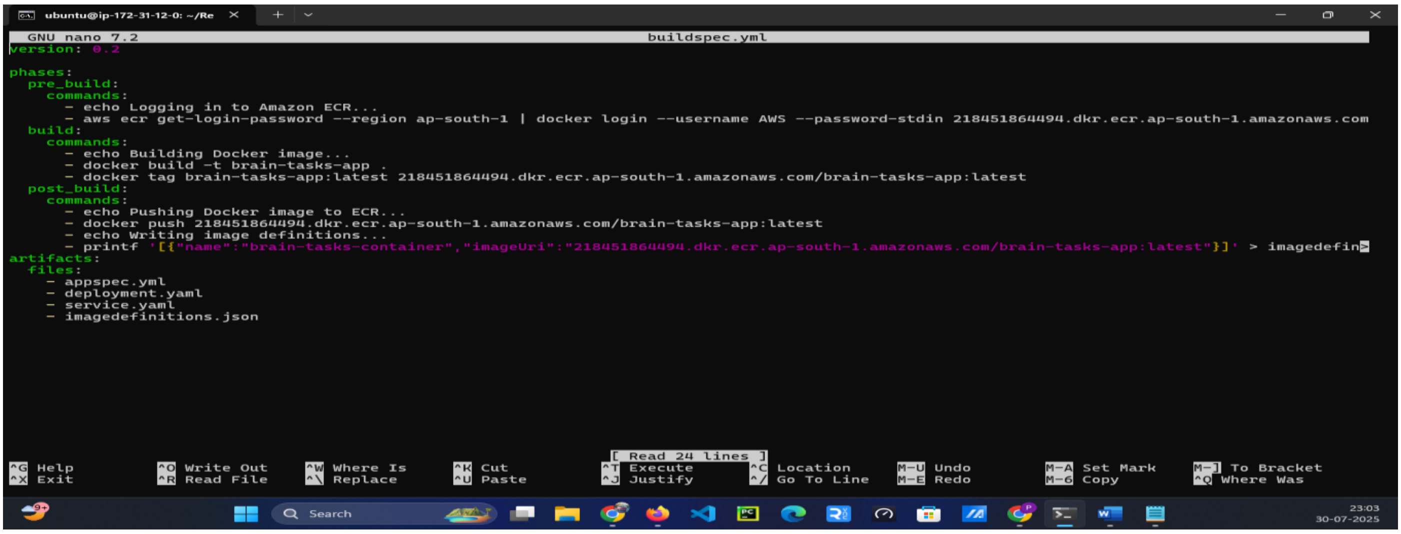
Pipeline relies on specific files in source repository's root:

- Dockerfile: Defines how React app is containerized.
- buildspec.yml: Instructions for AWS CodeBuild (building, testing, pushing Docker image, preparing artifacts).
- deployment.yaml: Kubernetes manifest for application's deployment.
- AWS Infrastructure: An existing EKS cluster and an ECR repository.
- service.yaml: Kubernetes manifest for application's service (e.g., LoadBalancer).

Step 1: Prepare buildspec.yml and appspe.YML for CodeBuild

This file tells CodeBuild how to process code. Place it at the root of your repository.

```
version: 0.2

phases:

  pre_build:

    commands:

      - echo Logging in to Amazon ECR...

      - aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin
218451864494.dkr.ecr.ap-south-1.amazonaws.com  build:

    commands:

      - echo Building Docker image...

      - docker build -t brain-tasks-app .

      - docker tag brain-tasks-app:latest 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-
app:latest

  post_build:

    commands:

      - echo Pushing Docker image to ECR...

      - docker push 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest

      - echo Writing image definitions...

  - printf '[{"name":"brain-tasks-container","imageUri":"218451864494.dkr.ecr.ap-south-
1.amazonaws.com/brain-tasks-app:latest"}]' > imagedefin>artifacts:

 files:

    - appspec.yml

    - deployment.yaml

    - service.yaml

    - imagedefinitions.json
```



```
  GNU nano 7.2                                    buildspec.yml
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 218451864494.dkr.ecr.ap-south-1.amazonaws.com
  build:
    commands:
      - echo Building Docker image...
      - docker build -t brain-tasks-app .
      - docker tag brain-tasks-app:latest 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
  post_build:
    commands:
      - echo Pushing Docker image to ECR...
      - docker push 218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
      - echo Writing image definitions...
      - printf '[{"name":"brain-tasks-container","imageUri":"218451864494.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest"}]' > imagedefin>
artifacts:
  files:
    - appspec.yml
    - deployment.yaml
    - service.yaml
    - imagedefinitions.json

                                        [ Read 24 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

Appspec.yml

```
version: 0.0
Resources:
  - myApp:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:<region>:218451864494:task-definition/dummy:1"
        LoadBalancerInfo:
          ContainerName: "brain-tasks-container" # Must match container name in imagedefinitions.json
          ContainerPort: 80 # Must match container port in imagedefinitions.json
Hooks:
  AfterInstall:
    - location: scripts/deploy_to_eks.sh
      timeout: 300
      runas: root
```

Step 2: Create AWS CodePipeline

1. **Go to CodePipeline Console:** console.aws.amazon.com/codepipeline/home
2. **Create Pipeline:** Click "Create pipeline".
    - **Pipeline Settings:**
        - **Pipeline name:** BrainTasksAppPipeline (or your chosen name)
        - **Service role:** Select New service role. AWS will create an IAM role for CodePipeline. **Note its ARN later for EKS permissions.**
        - **Artifact store:** Leave as Default location.
        - Click Next.
    - **Source Stage:**
        - **Source provider:** Choose your Git provider (e.g., GitHub)
        - **Connection:** Set up a new connection or select an existing one to your Git account.
        - **Repository name:** Select your repository (e.g., https://github.com/PranuthHM/React_Application_Deployment.git).
        - **Branch name:** Select the branch (e.g., main).
        - **Output artifact format:** CodePipeline default.
        - Click Next.
    - **Build Stage (AWS CodeBuild):**
        - **Build provider:** AWS CodeBuild.

        - **Region:** Your AWS region (e.g., ap-south-1).
        - **Project name:** Click Create project. A new tab will open.
            - **Project name:** brain-tasks-app-build (or your chosen name).
            - **Environment image:** Managed image
            - **Operating system:** Amazon Linux 2
            - **Runtime:** Standard
            - **Image:** Select an image that supports Docker and Node.js (e.g., aws/codebuild/amazonlinux2-x86_64-standard:4.0 or later).
            - **Image version:** Always use the latest image version.
            - **Service role:** Select New service role.
            - **Buildspec:** Use a buildspec file (CodeBuild will automatically find buildspec.yml).
            - Click Continue to CodePipeline (this closes the CodeBuild tab).
        - Ensure your new CodeBuild project is selected.
        - Click Next.
    - **Deploy Stage (Amazon EKS):**
        - **Deploy provider:** Amazon EKS.
        - **Region:** Your AWS region (e.g., ap-south-1).
        - **Cluster name:** Select your EKS cluster (e.g., brain-app-cluster).
        - **Namespace:** default (or your target namespace).
        - **Deployment file:** deployment.yaml
        - **Service file:** service.yaml
        - **Image definitions file:** imagedefinitions.json
        - Click Next.
    - **Review:**
        - Review all settings.
        - Click Create pipeline.

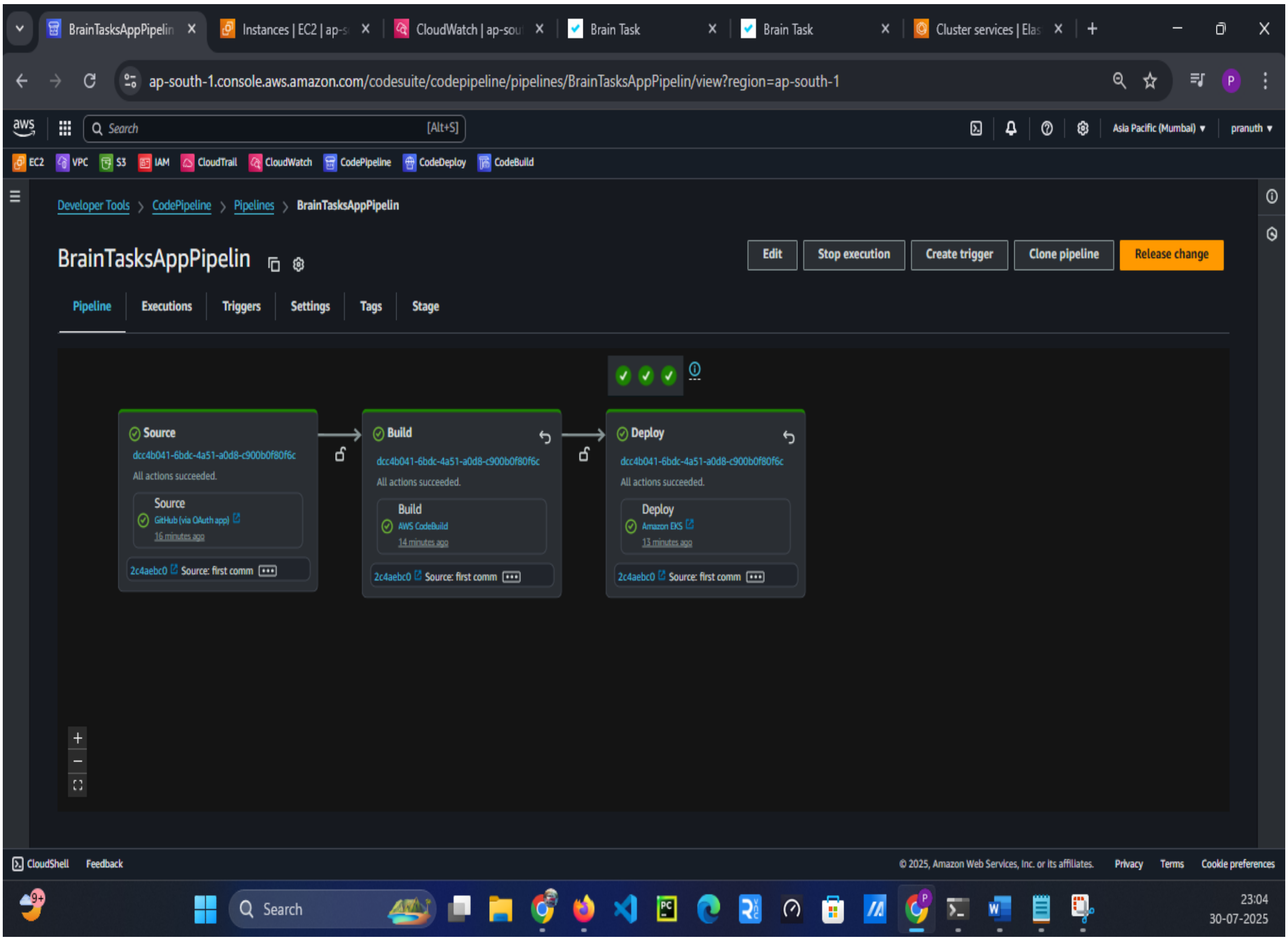
Step 3: Grant EKS Permissions to CodePipeline's Service Role

The CodePipeline role needs permission to interact with your EKS cluster.

1. **Get CodePipeline Role ARN:**
    - Go to **IAM console -> Roles**.
    - Find the role named AWSCodePipelineServiceRole-YourPipelineName-Region (e.g., AWSCodePipelineServiceRole-ap-south-1-BrainTasksAppPipeline).
    - **Copy its ARN.**

2. **Add EKS Access Entry:**
   - Go to **EKS console** -> **Clusters**.
   - Select your cluster (e.g., brain-app-cluster).
   - Go to the **"Access"** tab -> **"Access entries"** -> **"Create access entry"**.
   - **Configure IAM access entry:**
     - **IAM principal ARN:** Paste the CodePipeline service role ARN.
     - **Type:** Standard.
     - **Groups :** Optional: Leave this field BLANK.
     - Click Next.
   - **Add access policy:**
     - **Policy to associate:** Select AmazonEKSClusterAdminPolicy.
     - **Access scope:** Cluster.
     - Click Add policy.
     - Click Next.
   - **Review and create:** Verify details.

Step 4: Verify Deployment



ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/BrainTasksAppPipelin/view?region=ap-south-1

Developer Tools > CodePipeline > Pipelines > BrainTasksAppPipelin

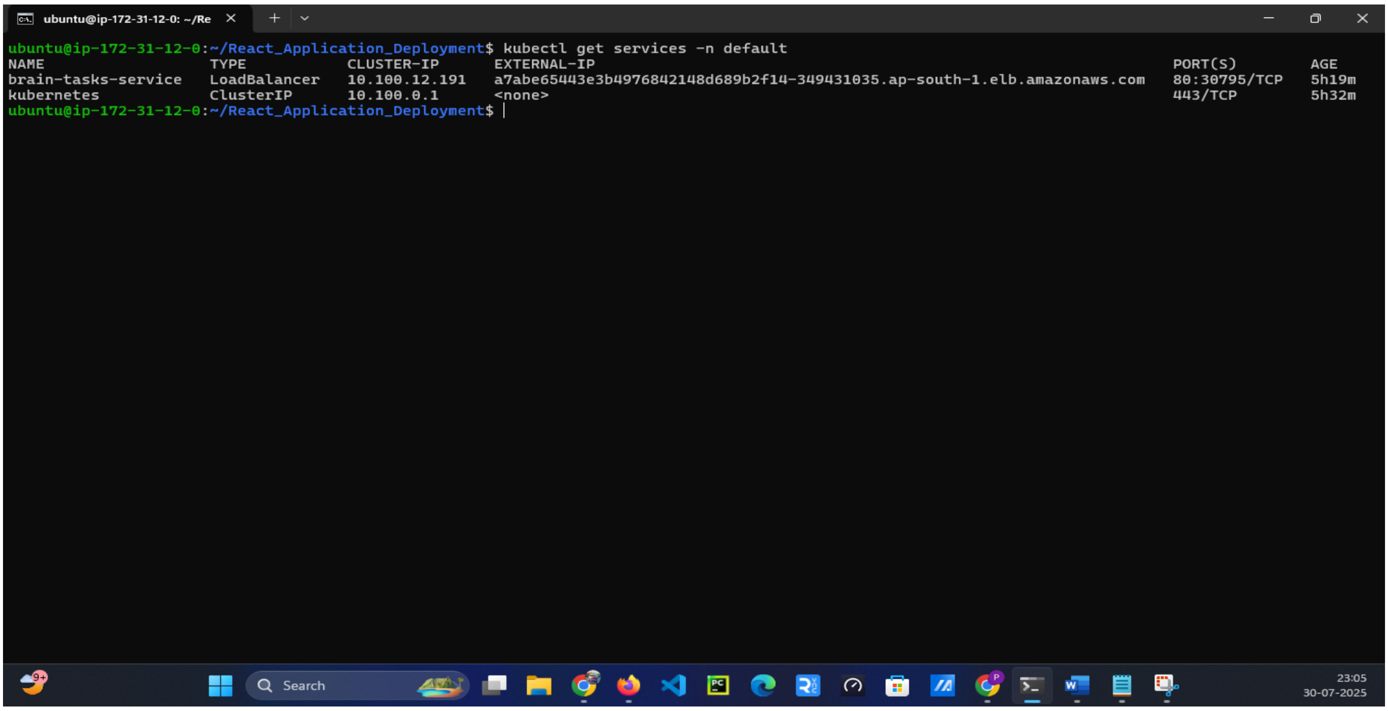
BrainTasksAppPipelin

Source
dcc4b041-6bdc-4a51-a0d8-c900b0f80f6c
All actions succeeded.
Source
GitHub (via OAuth app)
16 minutes ago
2c4aebc0 Source: first comm

Build
dcc4b041-6bdc-4a51-a0d8-c900b0f80f6c
All actions succeeded.
Build
AWS CodeBuild
14 minutes ago
2c4aebc0 Source: first comm

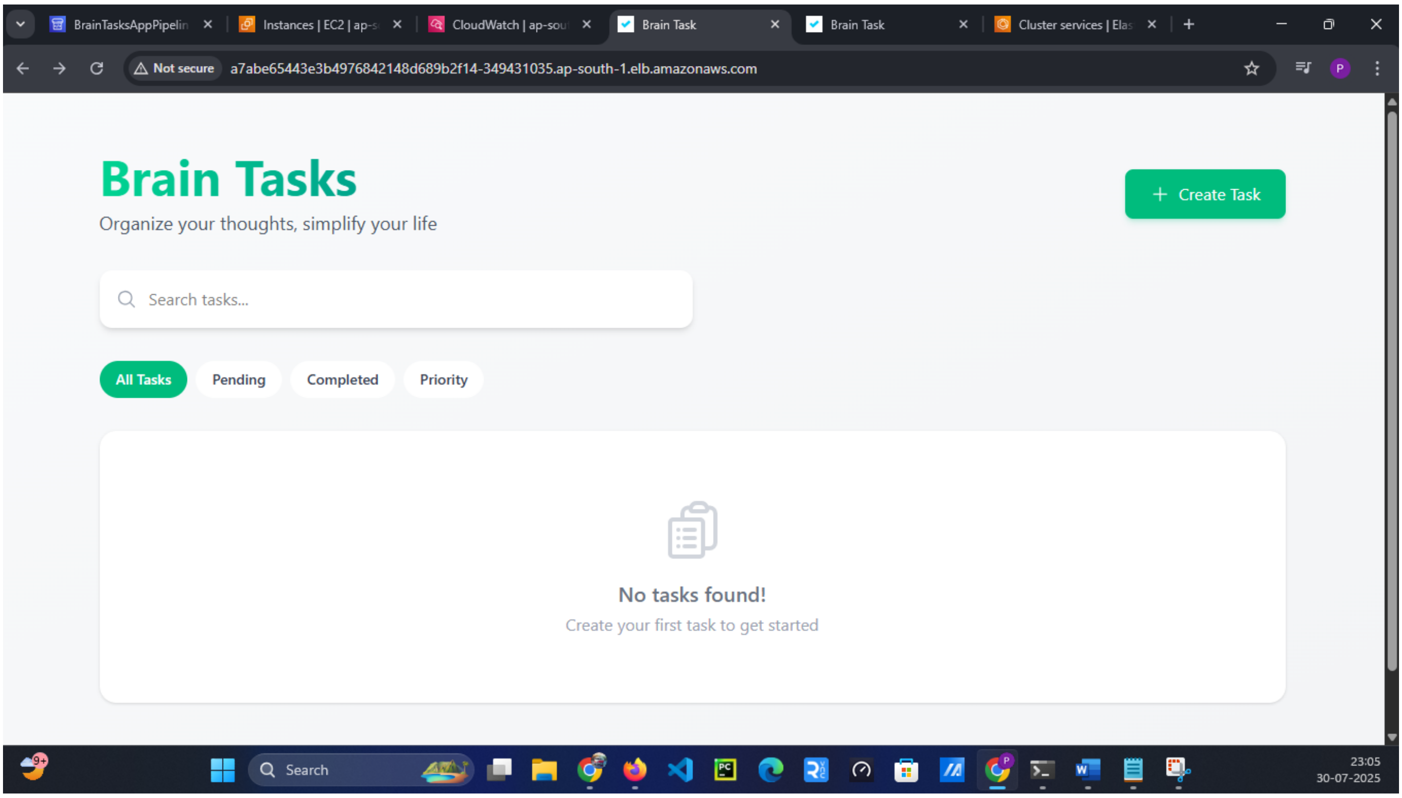
Deploy
dcc4b041-6bdc-4a51-a0d8-c900b0f80f6c
All actions succeeded.
Deploy
Amazon EKS
13 minutes ago
2c4aebc0 Source: first comm

1. **Monitor Pipeline:** The pipeline will start automatically. Ensure all stages (Source, Build, Deploy) complete successfully.
2. **Check EKS Resources:** Use kubectl (configured for your cluster) to verify:
   - kubectl get deployments -n default (Check for brain-tasks-app deployment status).
   - kubectl get services -n default or kubectl get svc (Check for brain-tasks-service and its EXTERNAL-IP if LoadBalancer).
   - kubectl get pods -n default -l app=brain-tasks-app (Check pod status).

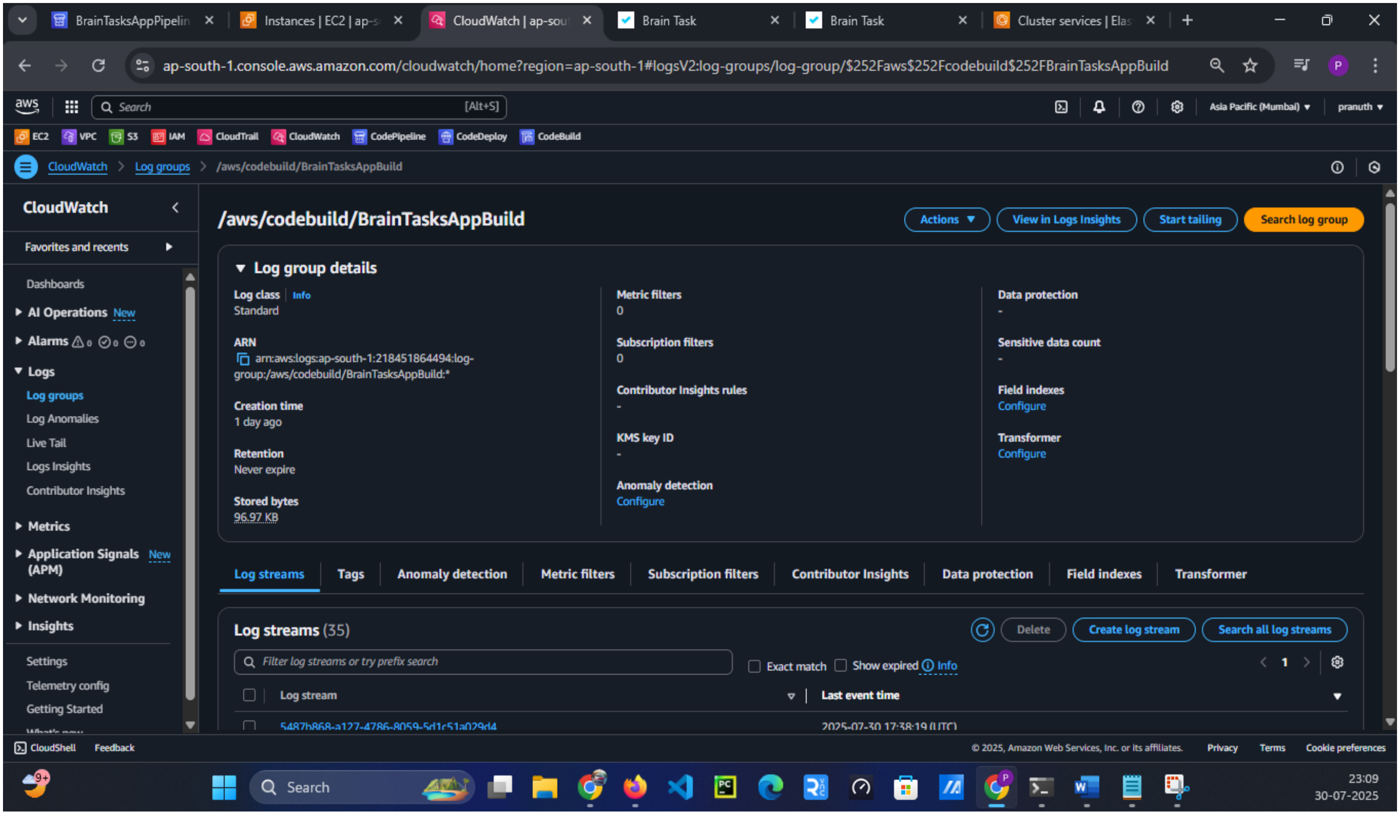


```
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ kubectl get services -n default
NAME                  TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)        AGE
brain-tasks-service   LoadBalancer   10.100.12.191   a7abe65443e3b4976842148d689b2f14-349431035.ap-south-1.elb.amazonaws.com   80:30795/TCP   5h19m
kubernetes            ClusterIP      10.100.0.1      <none>                                                                   443/TCP        5h32m
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ |
```

Access Application: Use the EXTERNAL-IP from the service to access your deployed React application in a web browser.



Cloud watch



ap-south-1.console.aws.amazon.com/cloudwatch/home?region=ap-south-1#logsV2:log-groups/log-group/$252Faws$252Fcodebuild$252FBrainTasksAppBuild

CloudWatch > Log groups > /aws/codebuild/BrainTasksAppBuild

/aws/codebuild/BrainTasksAppBuild

▼ Log group details

Log class | Info
Standard

ARN
arn:aws:logs:ap-south-1:218451864494:log-group:/aws/codebuild/BrainTasksAppBuild:*

Creation time
1 day ago

Retention
Never expire

Stored bytes
96.97 KB

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

KMS key ID
-

Anomaly detection
Configure

Data protection
-

Sensitive data count
-

Field indexes
Configure

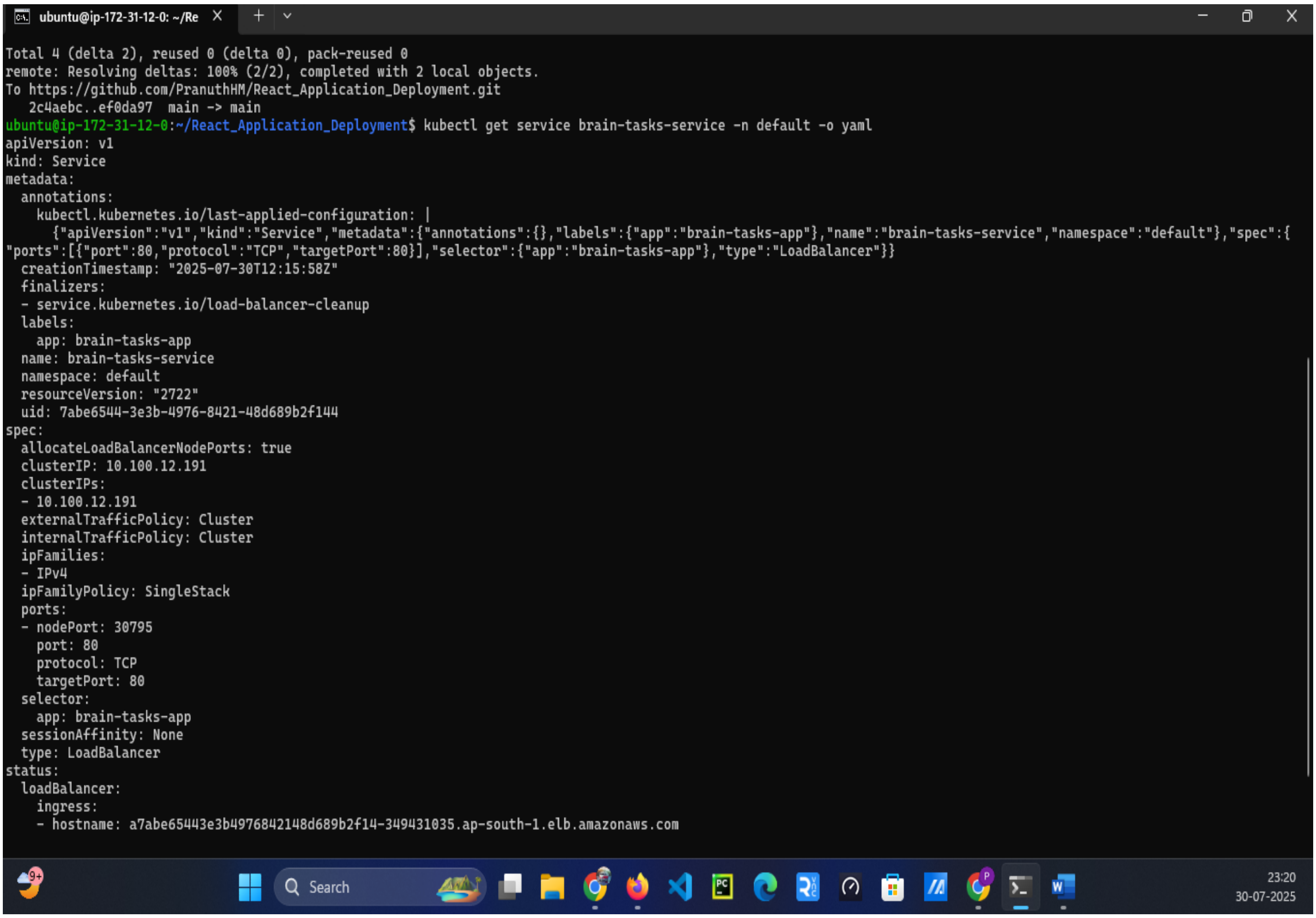
Transformer
Configure

Log streams (35)

□ Exact match □ Show expired Info

□	Log stream	Last event time
□	5487b868-a127-4786-8059-5d1c51a029d4	2025-07-30 17:38:19 (UTC)

- **Application deployed kubernetes Loadbalancer ARN.**



```
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/PranuthHM/React_Application_Deployment.git
   2c4aebc..ef0da97  main -> main
ubuntu@ip-172-31-12-0:~/React_Application_Deployment$ kubectl get service brain-tasks-service -n default -o yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"brain-tasks-app"},"name":"brain-tasks-service","namespace":"default"},"spec":{
"ports":[{"port":80,"protocol":"TCP","targetPort":80}],"selector":{"app":"brain-tasks-app"},"type":"LoadBalancer"}}
  creationTimestamp: "2025-07-30T12:15:58Z"
  finalizers:
  - service.kubernetes.io/load-balancer-cleanup
  labels:
    app: brain-tasks-app
  name: brain-tasks-service
  namespace: default
  resourceVersion: "2722"
  uid: 7abe6544-3e3b-4976-8421-48d689b2f144
spec:
  allocateLoadBalancerNodePorts: true
  clusterIP: 10.100.12.191
  clusterIPs:
  - 10.100.12.191
  externalTrafficPolicy: Cluster
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - nodePort: 30795
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: brain-tasks-app
  sessionAffinity: None
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
    - hostname: a7abe65443e3b4976842148d689b2f14-349431035.ap-south-1.elb.amazonaws.com
```

a7abe65443e3b4976842148d689b2f14-349431035.ap-south-1.elb.amazonaws.com/