Application Deployment (Brain-Task-App)

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

A production-ready static web application deployed on AWS EKS (Elastic Kubernetes Service) with an automated CI/CD pipeline using AWS Code Pipeline, Code Build, Code Deploy, and Lambda functions for Kubernetes orchestration.

Project Overview

Brain Tasks App is a lightweight static web application served using Nginx, demonstrating enterprise-level DevOps practices:

- **Containerization**: Dockized Nginx-based static site
- **Container Registry**: AWS ECR for secure image storage
- **Orchestration**: Kubernetes deployment on AWS EKS
- **CI/CD Automation**: CodePipeline → CodeBuild → CodeDeploy
- **Serverless Deployment**: Lambda function for kubectl operations
- **High Availability**: Multi-replica deployment with LoadBalancer

Tech stack that needs to install

Docker: https://docs.docker.com/engine/install/ubuntu/

Kubectl : https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html

Awscli: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

Nodejs: https://nodejs.org/en/download

First need to test the application locally to install the all the tools its working fine or not using these commands

npm install : command install all the dependencies related to the NodeJS

npm start: start the application on the locally port 3000 to test it working fine

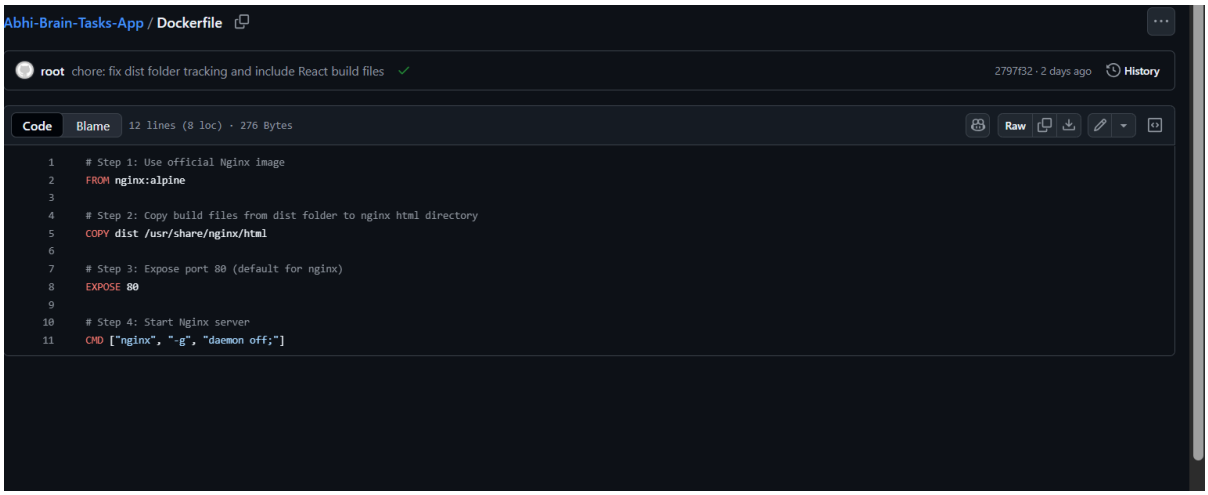
npm -v: checking version with the help of these command

AWS Services Used

- **Amazon ECR**: Container image registry
- **Amazon EKS**: Kubernetes cluster (1.28+)
- **AWS Lambda**: Serverless kubectl execution
- **AWS CodePipeline**: CI/CD orchestration

- **AWS CodeBuild**: Docker image building
- **AWS CodeDeploy**: Deployment automation
- **Amazon CloudWatch**: Logging and monitoring

First step Docker Configuration



Why this approach?

- **Alpine Linux**: Minimal image size (~23MB vs ~140MB for standard Nginx)
- **Static Content**: No Node.js runtime needed in production
- **Nginx**: Battle-tested, high-performance web server
- **Simple**: No complex builds steps, just copy and serve

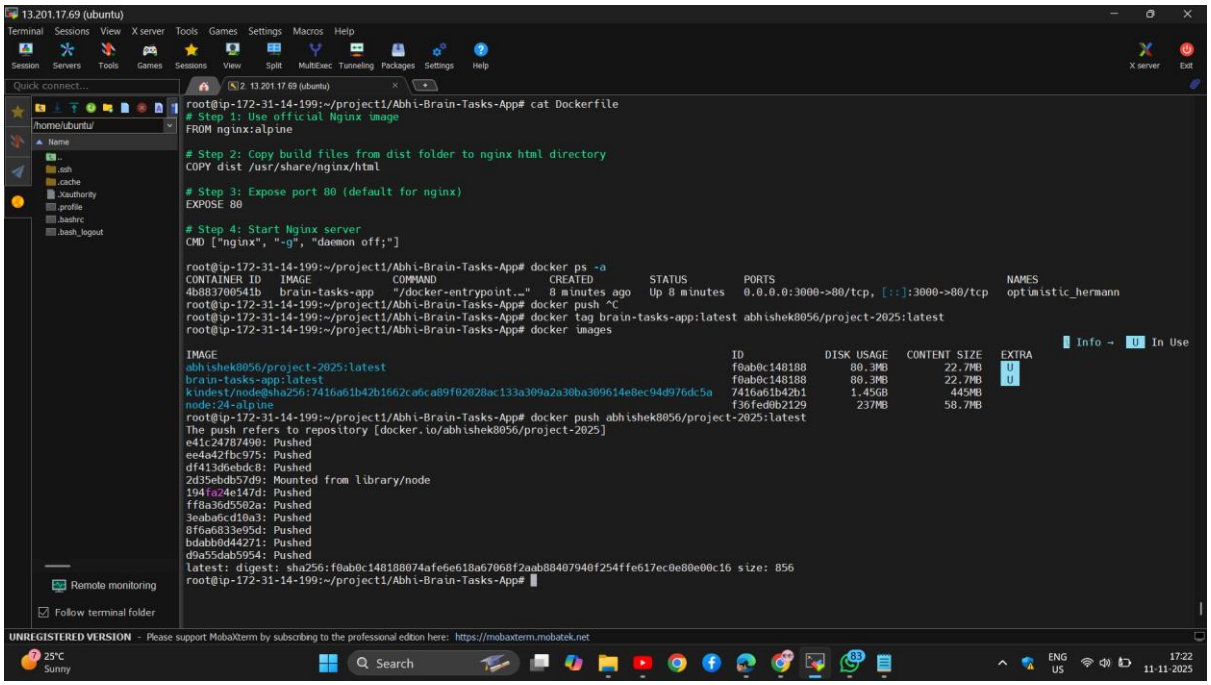
Build Docker Image Locally

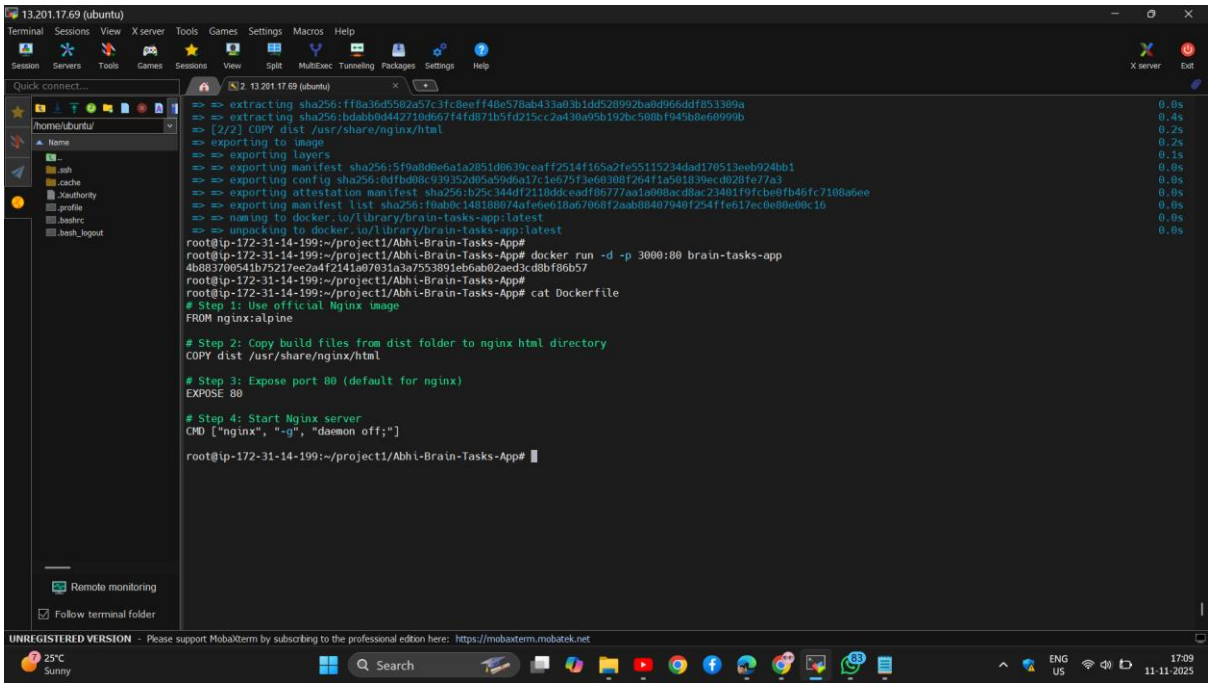
```
# Build the image
docker build -t brain-task-app.

# Verify image was created
docker images | grep brain-task-app
```

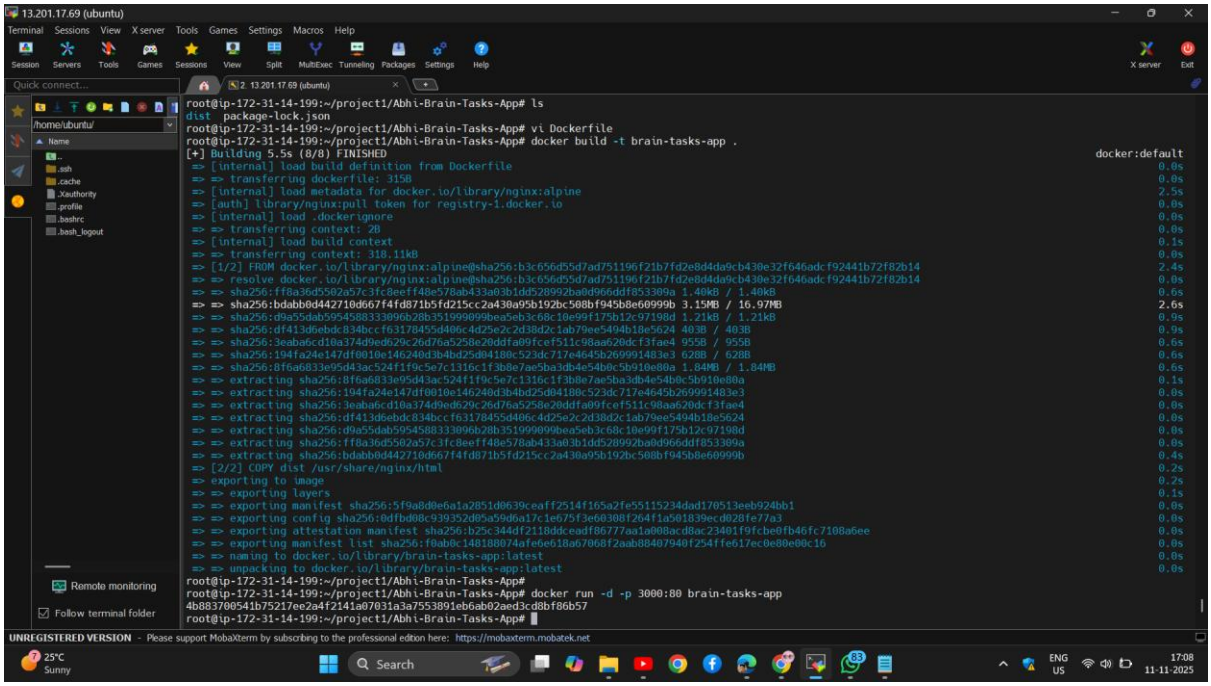
Test Docker Container Locally

```
# Run container
docker run -d -p 8080:80 --name brain-task-test brain-task-app

# Test in browser
curl http://localhost:8080

# View logs
docker logs brain-task-test

# Stop and remove container
docker stop brain-task-test
docker rm brain-task-test
```

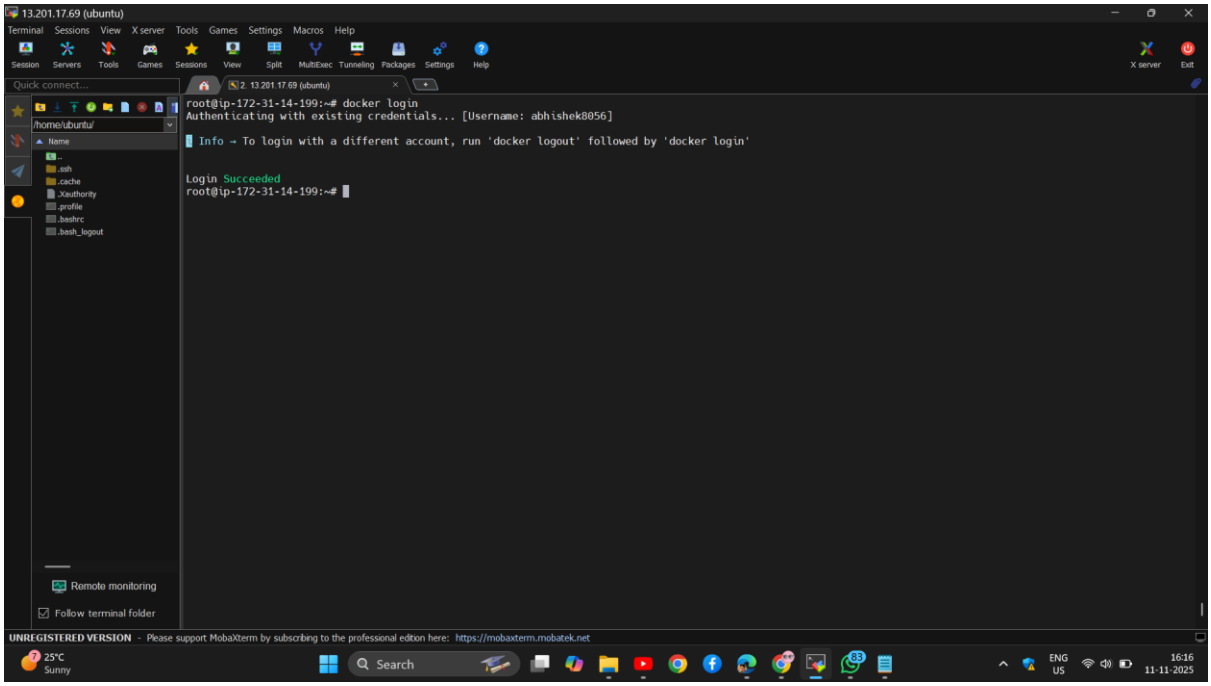
docker steps:



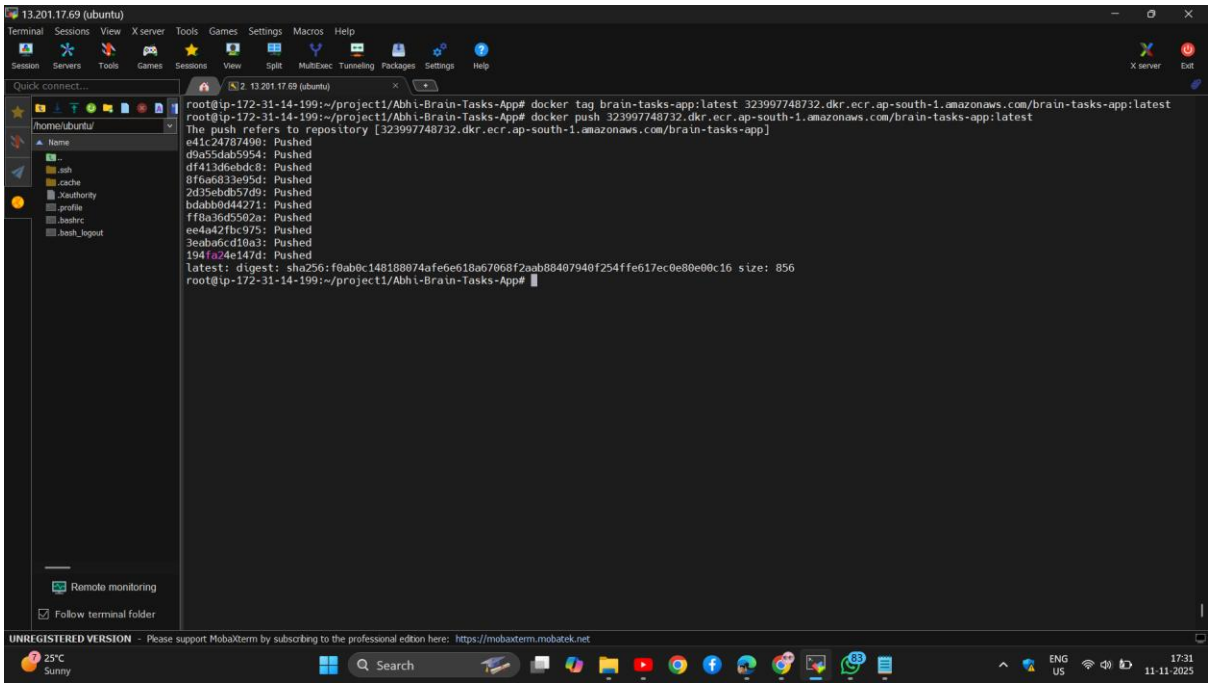


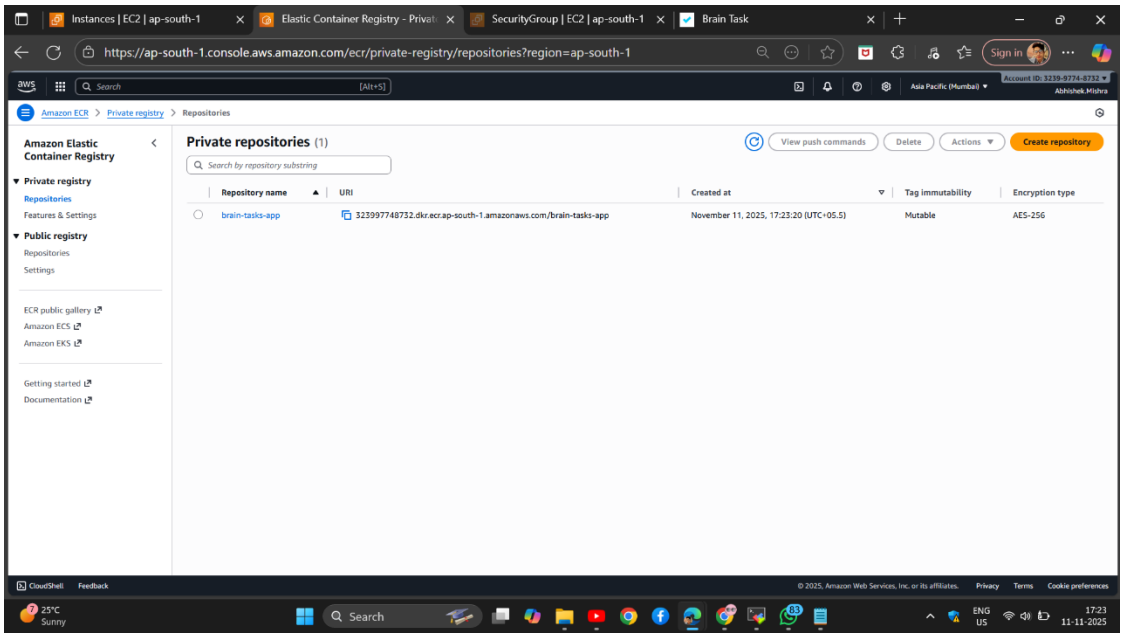


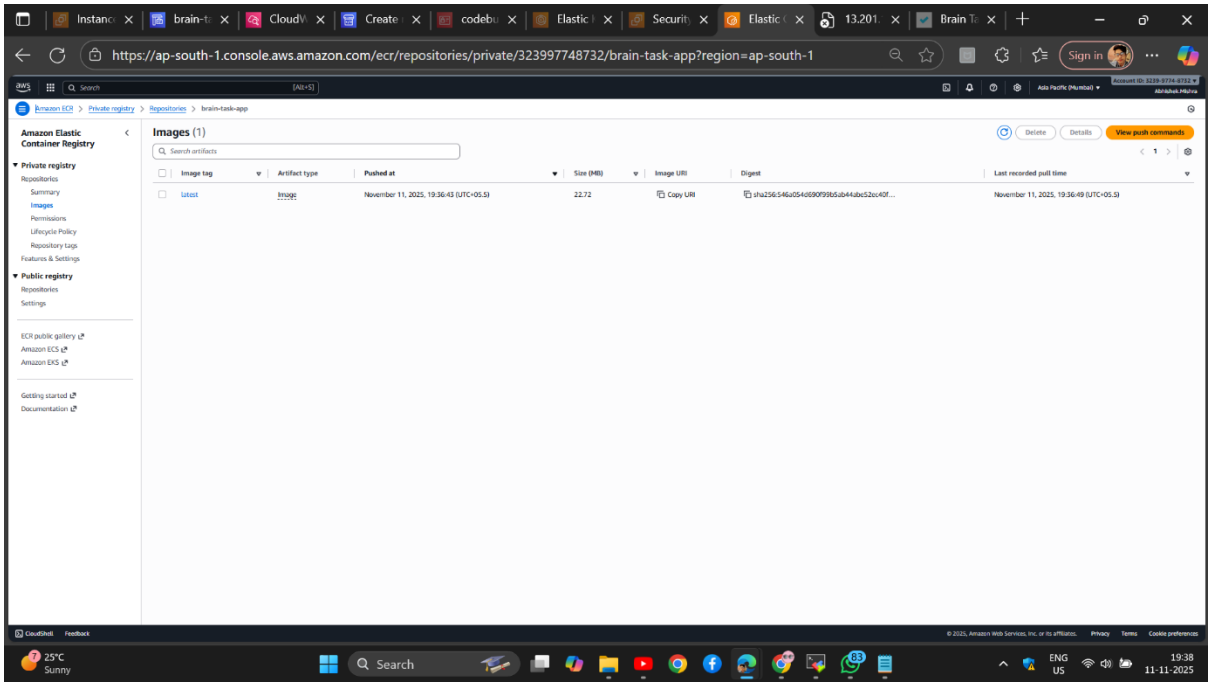
push image also for docker hub for future use:



Pushing image to Aws ECR







AWS ECR Setup

1. Create ECR Repository

```
aws ecr create-repository \
    --repository-name brain-task-app \
    --region ap-south-1 \
    --image-scanning-configuration scanOnPush=true
```

Authenticate Docker to ECR

```
aws ecr get-login-password --region ap-south-1 | \
docker login --username AWS --password-stdin 323997748732.dkr.ecr.ap-south-1.amazonaws.com
```

Expected Output:

```
Login Succeeded
```

Tag Docker Image

```
docker tag brain-task-app:latest \
323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
```

4. Push Image to ECR

```
docker push 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
```

5. Verify Image in ECR

```
aws ecr describe-images \
    --repository-name brain-task-app \
    --region ap-south-1
```

EKS Cluster Setup

1. Create EKS Cluster

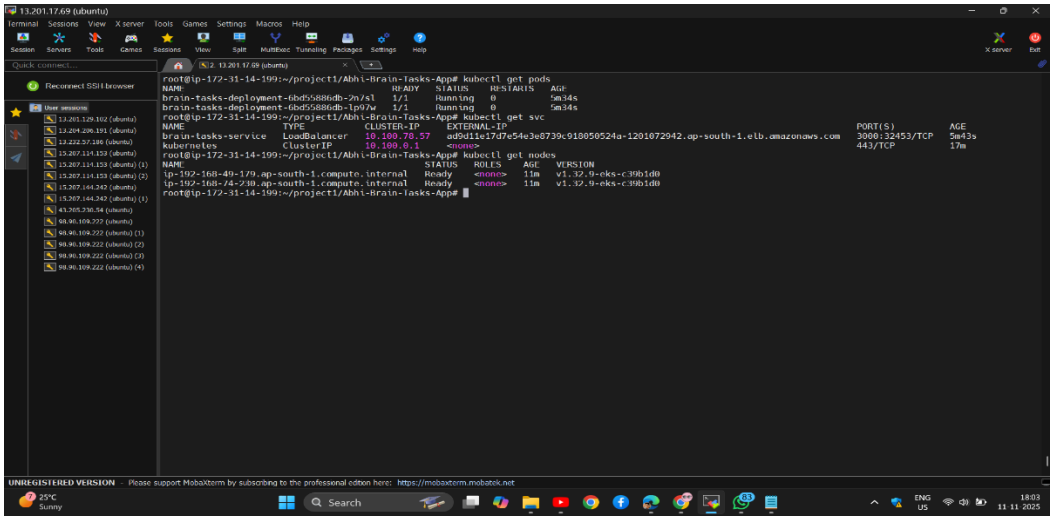
```
eksctl create cluster \
    --name brain-task-cluster \
    --region ap-south-1 \
    --nodegroup-name standard-workers \
    --node-type t3.medium \
    --nodes 2 \
    --nodes-min 2 \
    --nodes-max 4 \
    --managed
```

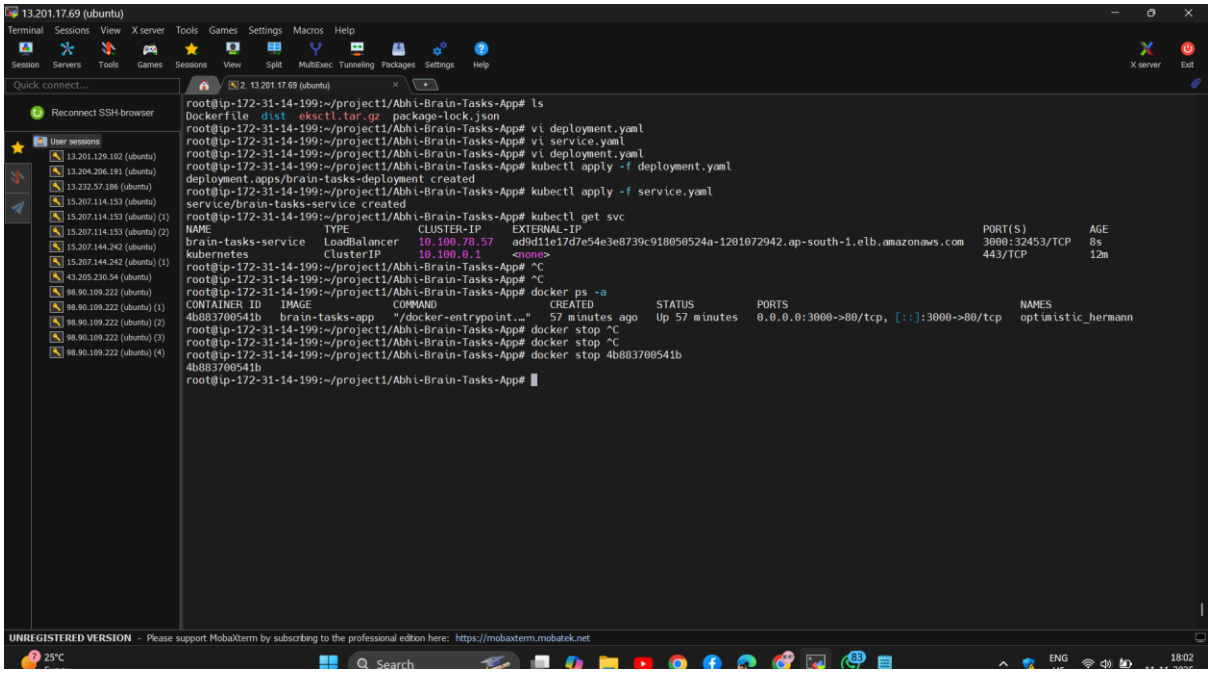
Note: This process takes 15-20 minutes.

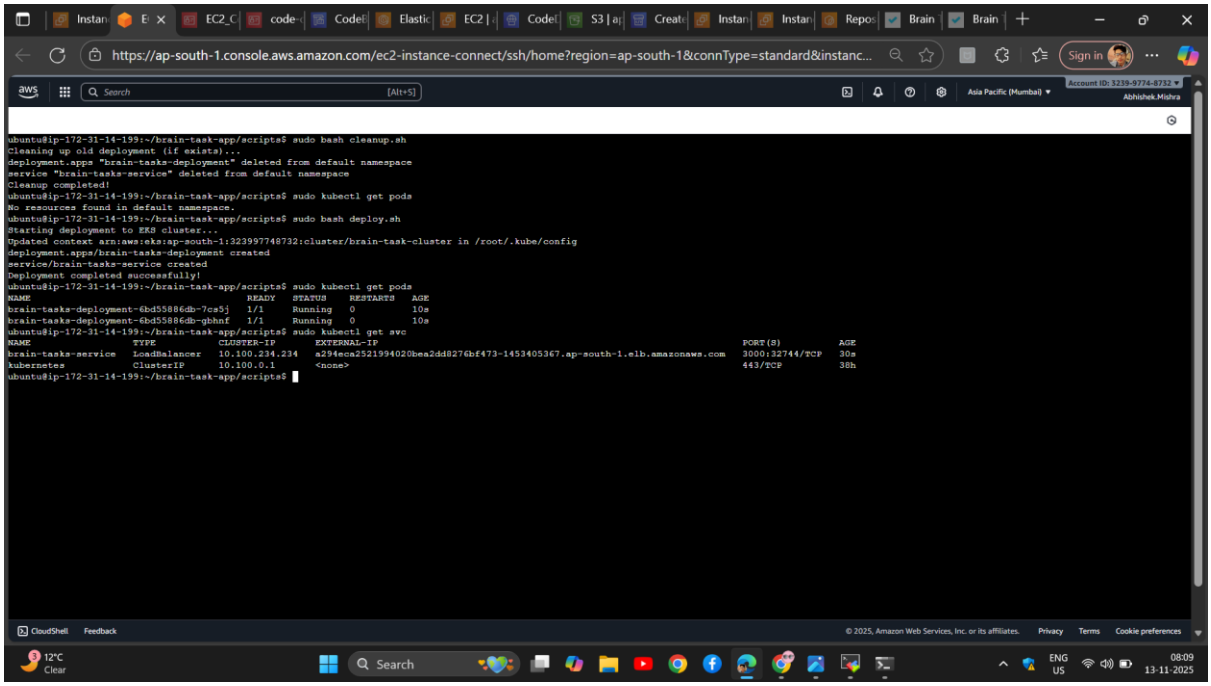
Verify Cluster Creation

```
# Check cluster status
aws eks describe-cluster --name brain-task-cluster --region ap-south-1 --query
cluster.status

# Update local kubeconfig
aws eks update-kubeconfig --name brain-task-cluster --region ap-south-1

# Test connection
kubectl get nodes
```

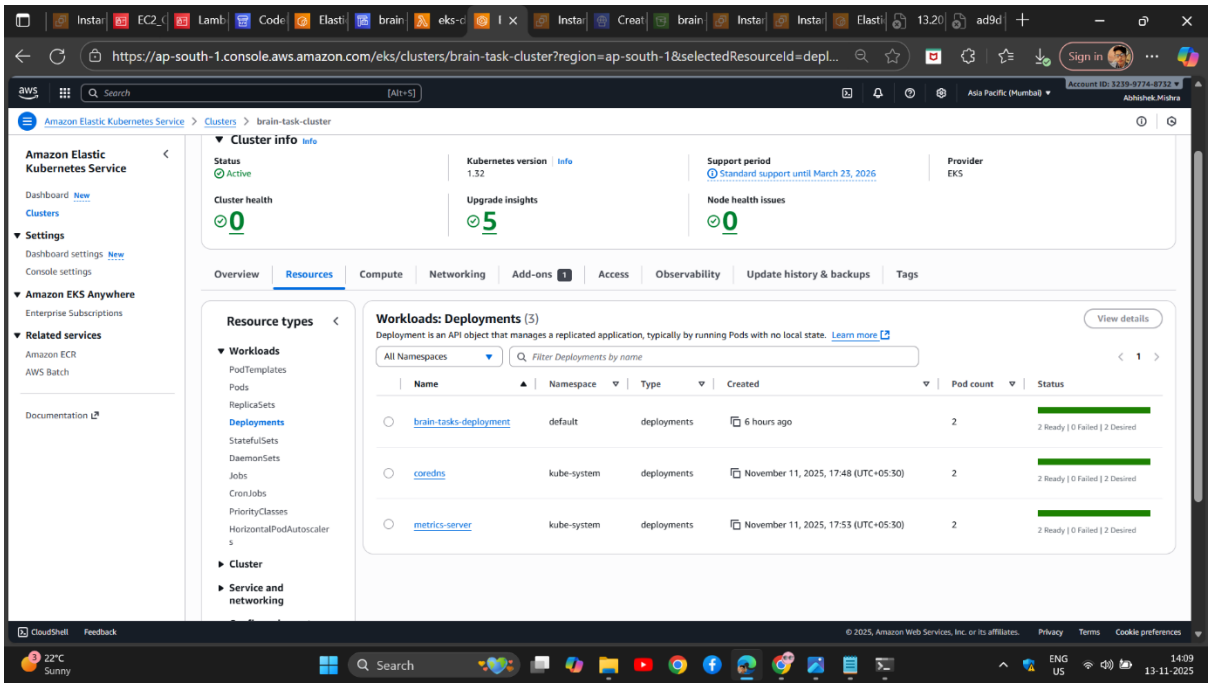
outputs:



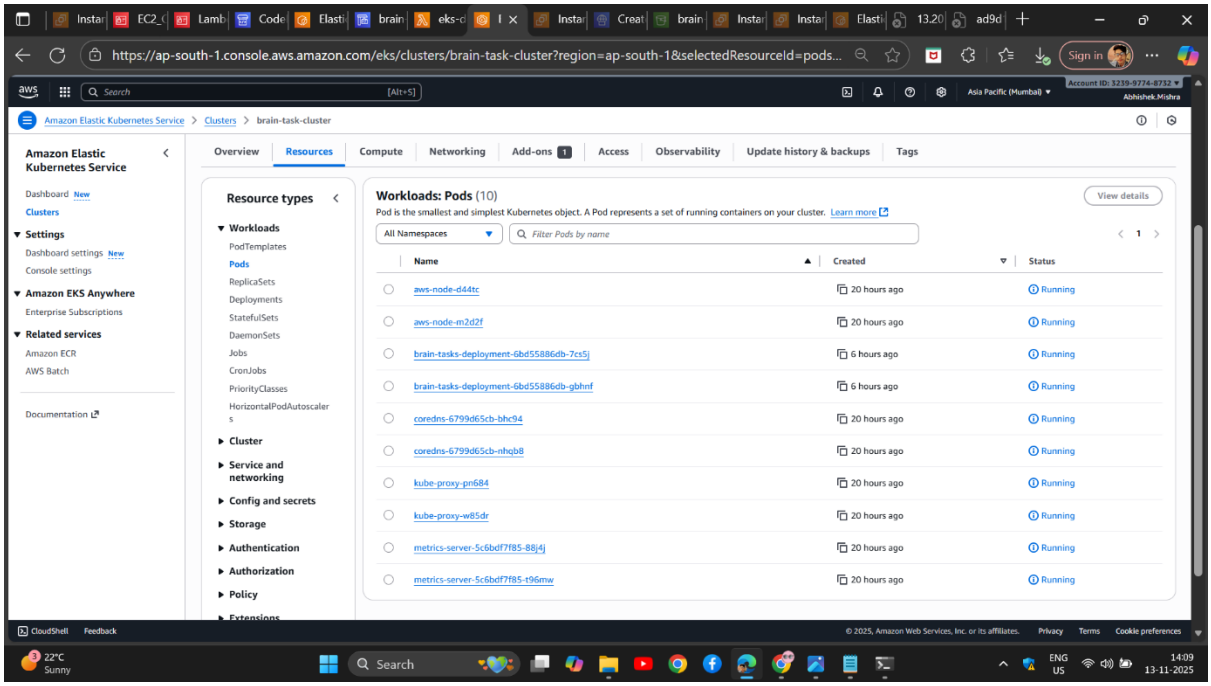

13.201.17.69 (ubuntu)
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# ls
Dockerfile dist eksctl.tar.gz package-lock.json
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# vi deployment.yaml
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# vi service.yaml
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# vi deployment.yaml
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# kubectl apply -f deployment.yaml
deployment.apps/brain-tasks-deployment created
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# kubectl apply -f service.yaml
service/brain-tasks-service created
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# kubectl get svc
NAME   TYPE   CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
brain-tasks-service LoadBalancer   10.100.78.57   ad9d11e17d7e54e3e8739c918050524a-1201072942.ap-south-1.elb.amazonaws.com   3000:32453/TCP   8s
kubernetes   ClusterIP   10.100.0.1   <none>   443/TCP   12m
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# ^C
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# ^C
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker ps -a
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
4b883700541b   brain-tasks-app   "/docker-entrypoint.…"   57 minutes ago   Up 57 minutes   0.0.0.0:3000->80/tcp, [::]:3000->80/tcp   optimistic_hermann
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker stop ^C
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker stop ^C
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker stop 4b883700541b
4b883700541b
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net


https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanc...
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$ sudo bash cleanup.sh
Cleaning up old deployment (if exists)...
deployment.apps "brain-tasks-deployment" deleted from default namespace
service "brain-tasks-service" deleted from default namespace
Cleanup completed!
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$ sudo kubectl get pods
No resources found in default namespace.
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$ sudo bash deploy.sh
Starting deployment to EKS cluster...
Updated context arn:aws:eks:ap-south-1:323997748732:cluster/brain-task-cluster in /root/.kube/config
deployment.apps/brain-tasks-deployment created
service/brain-tasks-service created
Deployment completed successfully!
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$ sudo kubectl get pods
NAME   READY   STATUS   RESTARTS   AGE
brain-tasks-deployment-6bd55886db-7cs5j   1/1   Running   0   10s
brain-tasks-deployment-6bd55886db-gbhnf   1/1   Running   0   10s
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$ sudo kubectl get svc
NAME   TYPE   CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
brain-tasks-service LoadBalancer   10.100.234.234   a294eca2521994020bea2dd8276bf473-1453405367.ap-south-1.elb.amazonaws.com   3000:32744/TCP   30s
kubernetes   ClusterIP   10.100.0.1   <none>   443/TCP   38h
ubuntu@ip-172-31-14-199:~/brain-task-app/scripts$
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

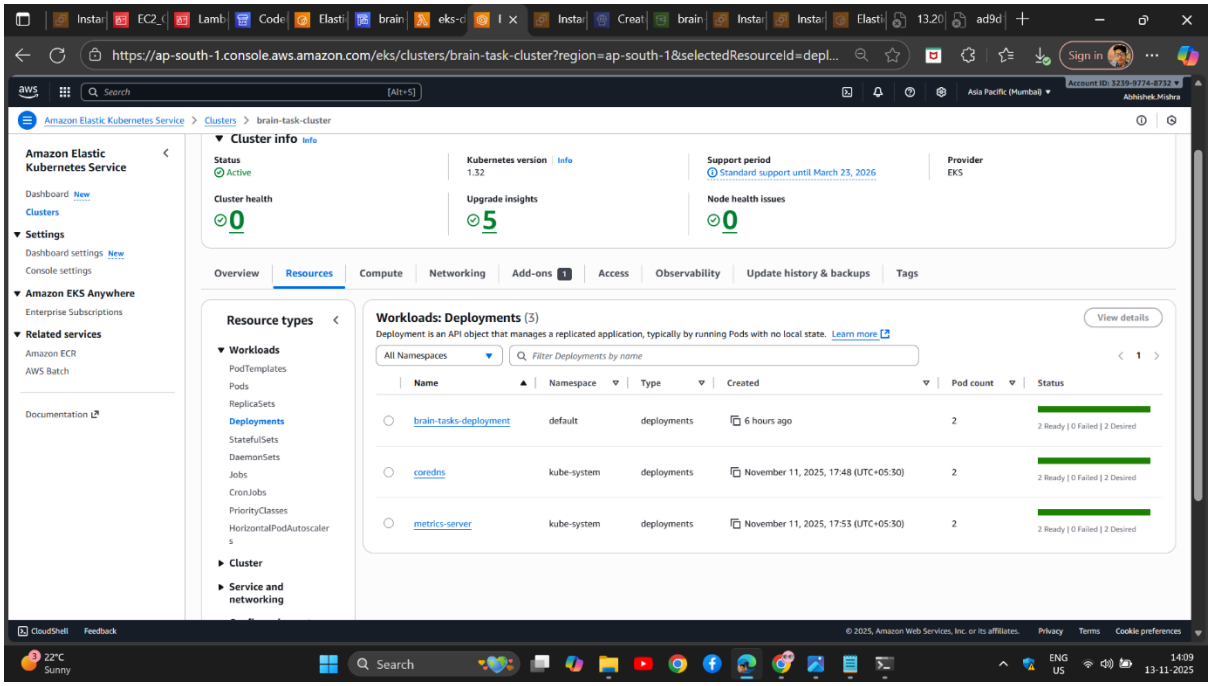
Cluster output:



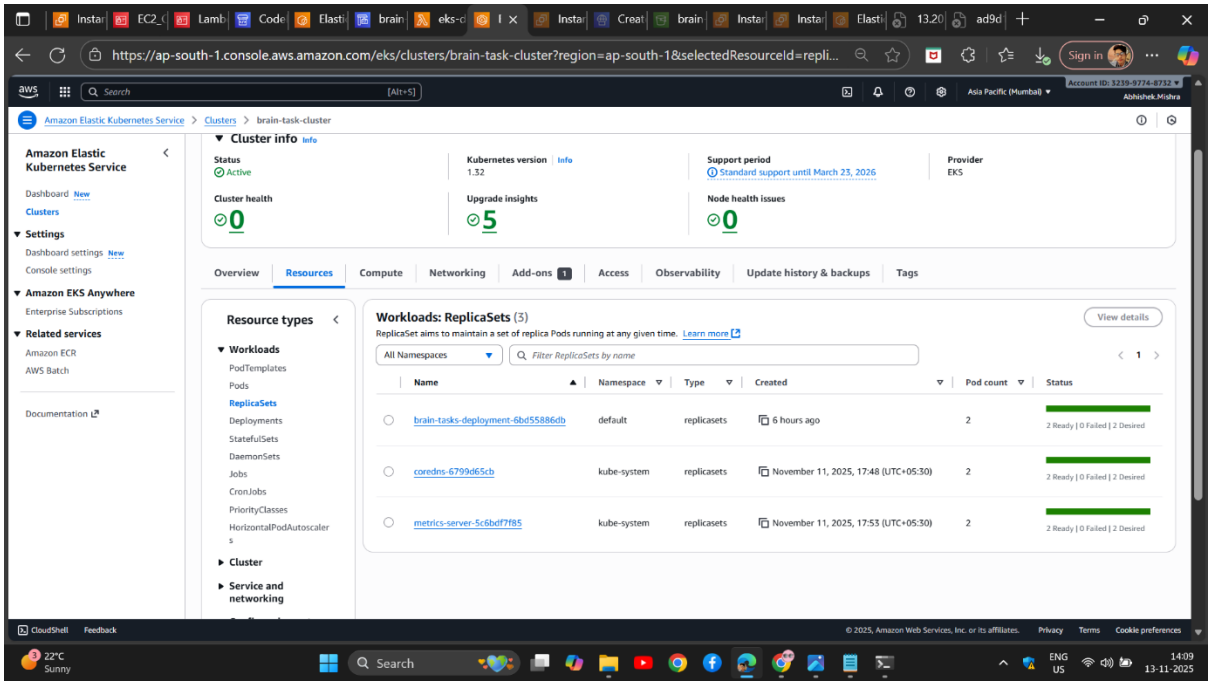
Workloads pods:



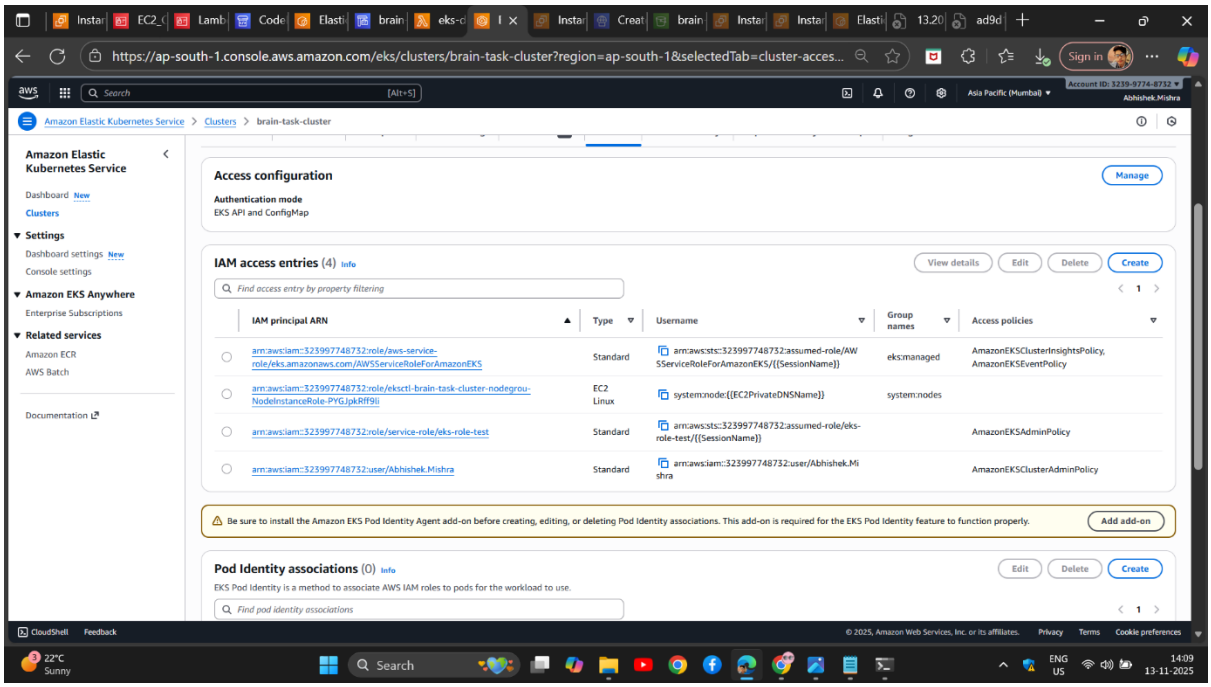
Deployment:



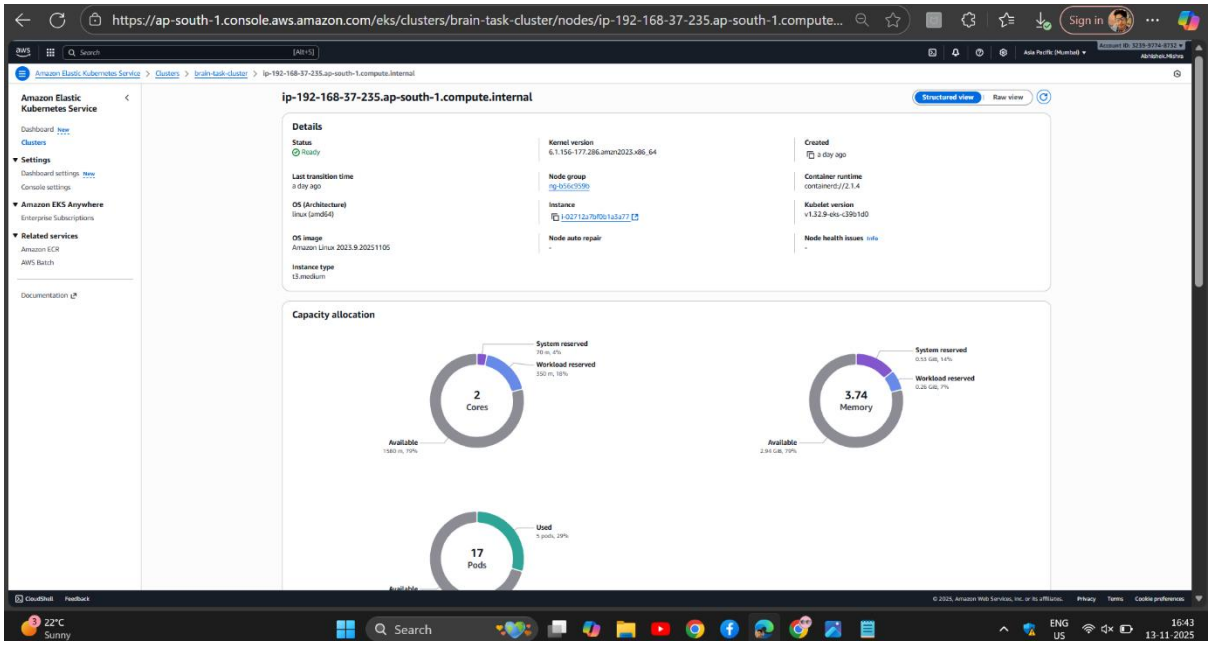
Replicasets

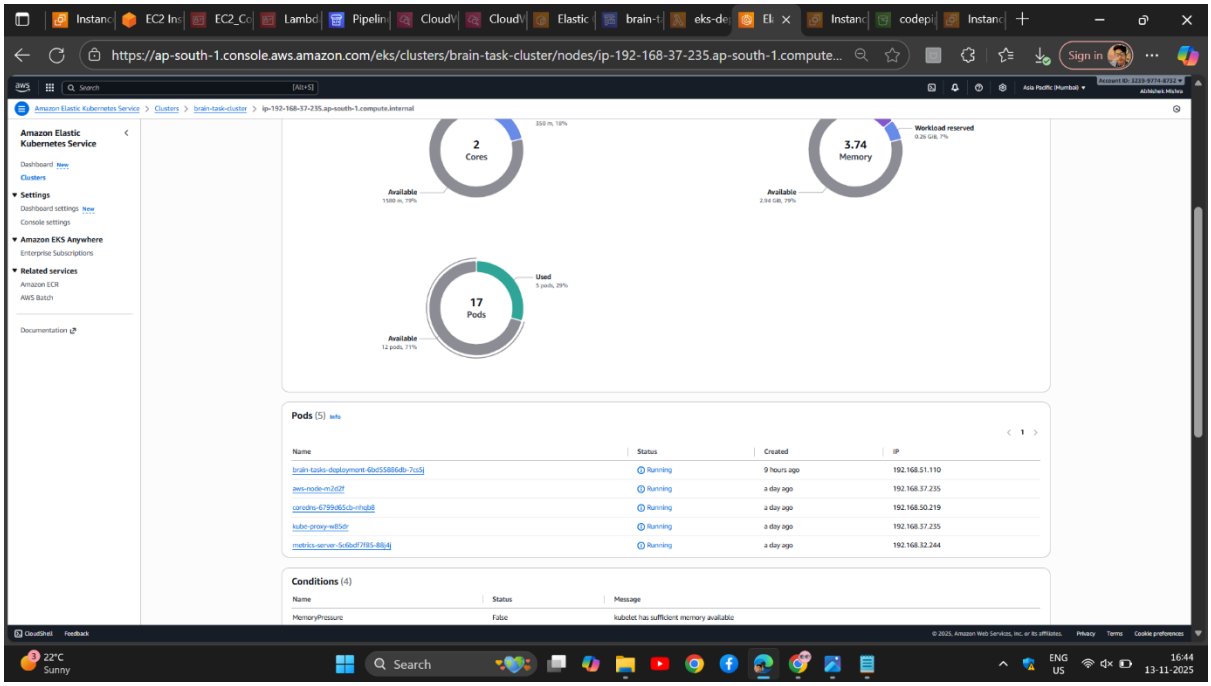


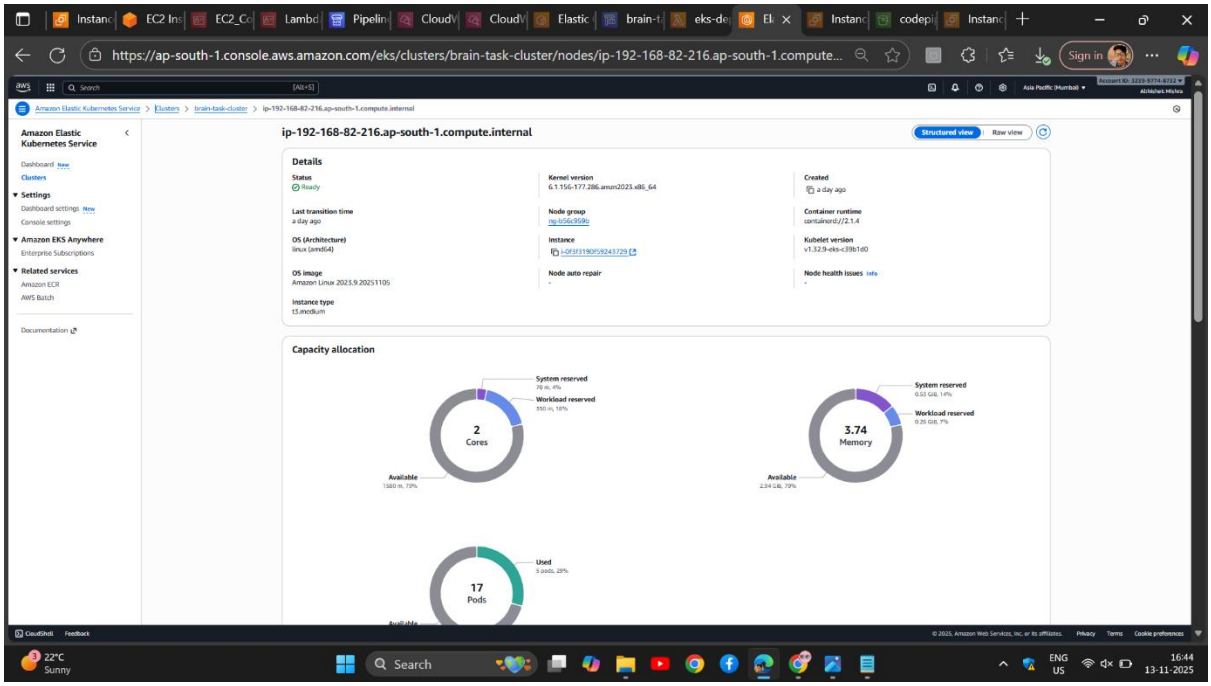
IAM access identity that access the codepipline in default namespace

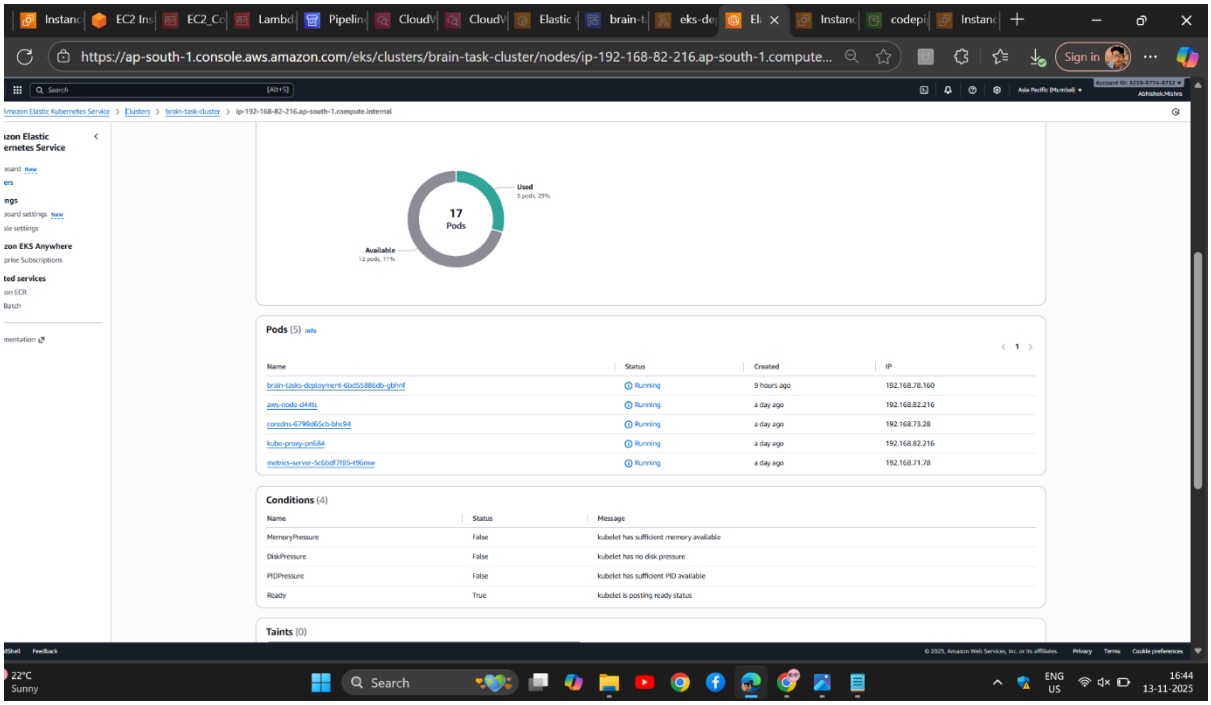


Cluster nodeport (using 2 instance with t3 medium for better performance)

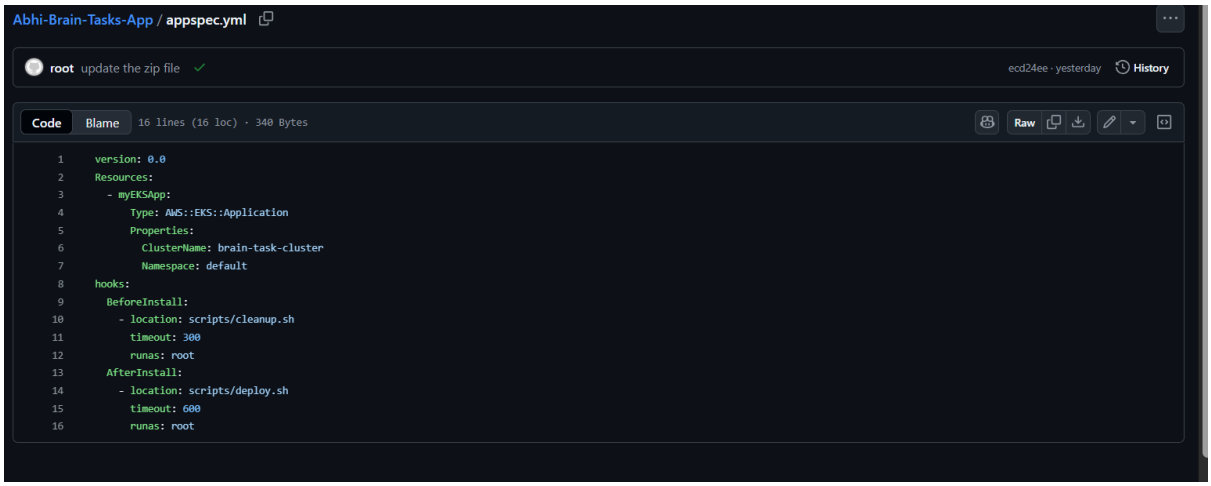



https://ap-south-1.console.aws.amazon.com/eks/clusters/brain-task-cluster/nodes/ip-192-168-37-235.ap-south-1.compute...
Amazon Elastic Kubernetes Service
Clusters > brain-task-cluster > ip-192-168-37-235.ap-south-1.compute.internal
2 Cores
Available
3.74 Memory
Workload reserved
Available
17 Pods
Used
Available
Pods (5)
Name
Status
Created
IP
brain-tasks-deployment-6bd55886db-7cs5j
Running
9 hours ago
192.168.51.110
aws-node-m2d2f
Running
a day ago
192.168.37.235
coredns-6799d65cb-rlhzb8
Running
a day ago
192.168.50.219
kube-proxy-w85dr
Running
a day ago
192.168.37.235
metrics-server-5c6bdf7f85-88j4j
Running
a day ago
192.168.32.244
Conditions (4)
Name
Status
Message
MemoryPressure
False
kubelet has sufficient memory available
22°C
Sunny
Search
ENG
US
16:44
13-11-2025


https://ap-south-1.console.aws.amazon.com/eks/clusters/brain-task-cluster/nodes/ip-192-168-82-216.ap-south-1.compute...
Amazon Elastic Kubernetes Service
Clusters > brain-task-cluster > ip-192-168-82-216.ap-south-1.compute.internal
ip-192-168-82-216.ap-south-1.compute.internal
Structured view
Raw view
Details
Status
Ready
Kernel version
6.1.156-177.286.amzn2023.x86_64
Created
a day ago
Last transition time
a day ago
Node group
Container runtime
containerd://2.1.4
OS (Architecture)
linux (amd64)
Instance
Kubelet version
v1.32.9-eks-c39b1d0
OS image
Amazon Linux 2023.9.20251105
Node auto repair
Node health issues
Instance type
t3.medium
Capacity allocation
System reserved
Workload reserved
2 Cores
Available
3.74 Memory
System reserved
Workload reserved
Available
17 Pods
Used
22°C
Sunny
Search
ENG
US
16:44
13-11-2025



Deployment using the k8s(files)



Abhi-Brain-Tasks-App / **buildspec.yml**

root chore: fix dist folder tracking and include React build files ✓ 2797f32 · 2 days ago History

Code Blame 25 lines (20 loc) · 863 Bytes Raw

```yaml
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 323997748732.dkr.ecr.ap-south-1.amazonaws.com

  build:
    commands:
      - echo Build started on `date`
      - docker build -t brain-task-app .
      - docker tag brain-task-app:latest 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest

  post_build:
    commands:
      - echo Pushing Docker image to ECR...
      - docker push 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
      - echo Writing image definitions file...
      - printf '[{"name":"brain-task-container","imageUri":"323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest"}]' > imagedefinitions.json

artifacts:
  files:
    - imagedefinitions.json
```

Abhi-Brain-Tasks-App / **service.yaml**

root chore: fix dist folder tracking and include React build files ✓ 2797f32 · 2 days ago History

Code Blame 13 lines (12 loc) · 189 Bytes Raw

```yaml
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
spec:
  selector:
    app: brain-tasks
  ports:
  - protocol: TCP
    port: 3000
    targetPort: 80
  type: LoadBalancer
```

Abhi-Brain-Tasks-App / **deployment.yaml**

root chore: fix dist folder tracking and include React build files ✓ 2797f32 · 2 days ago History

Code Blame 20 lines (19 loc) · 400 Bytes Raw

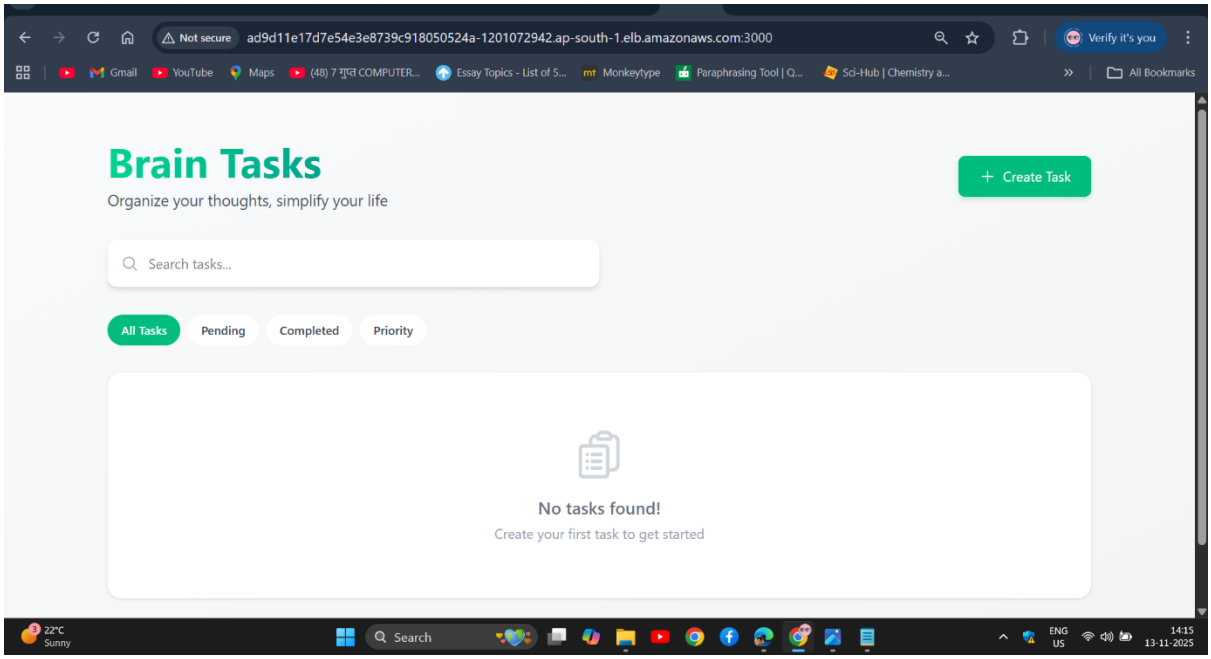
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: brain-tasks
  template:
    metadata:
      labels:
        app: brain-tasks
    spec:
      containers:
      - name: brain-tasks-container
        image: 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app
        ports:
        - containerPort: 80
```

Deployment Process

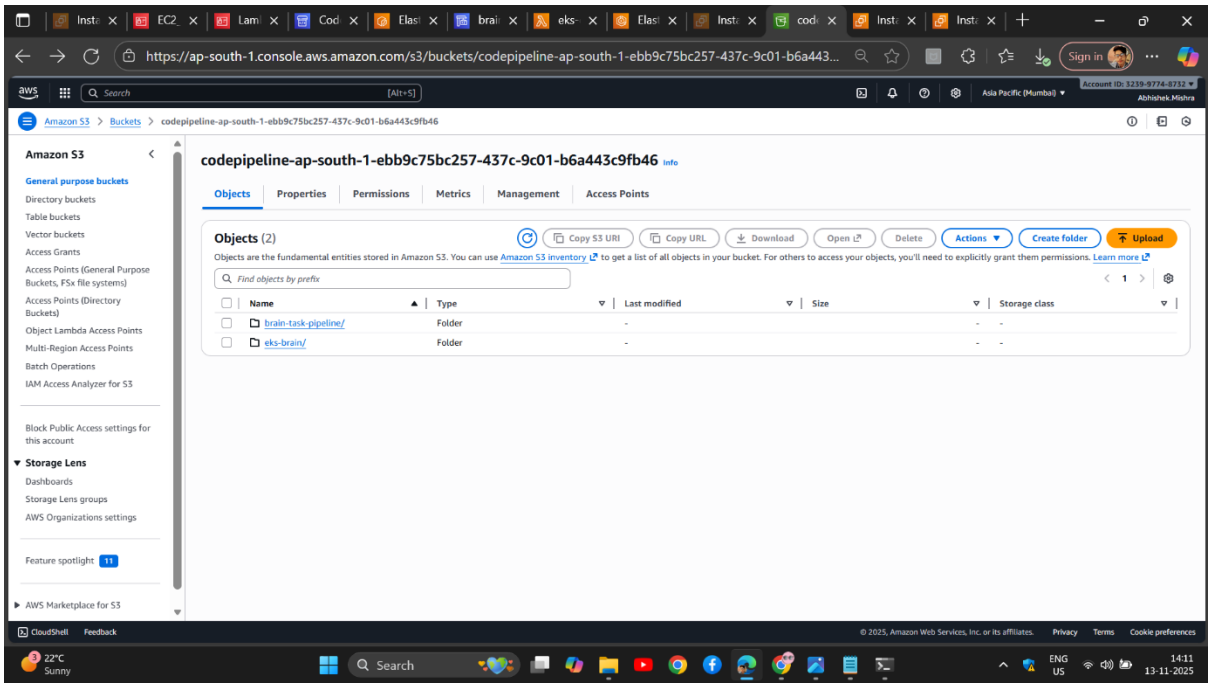
Automated Deployment Flow

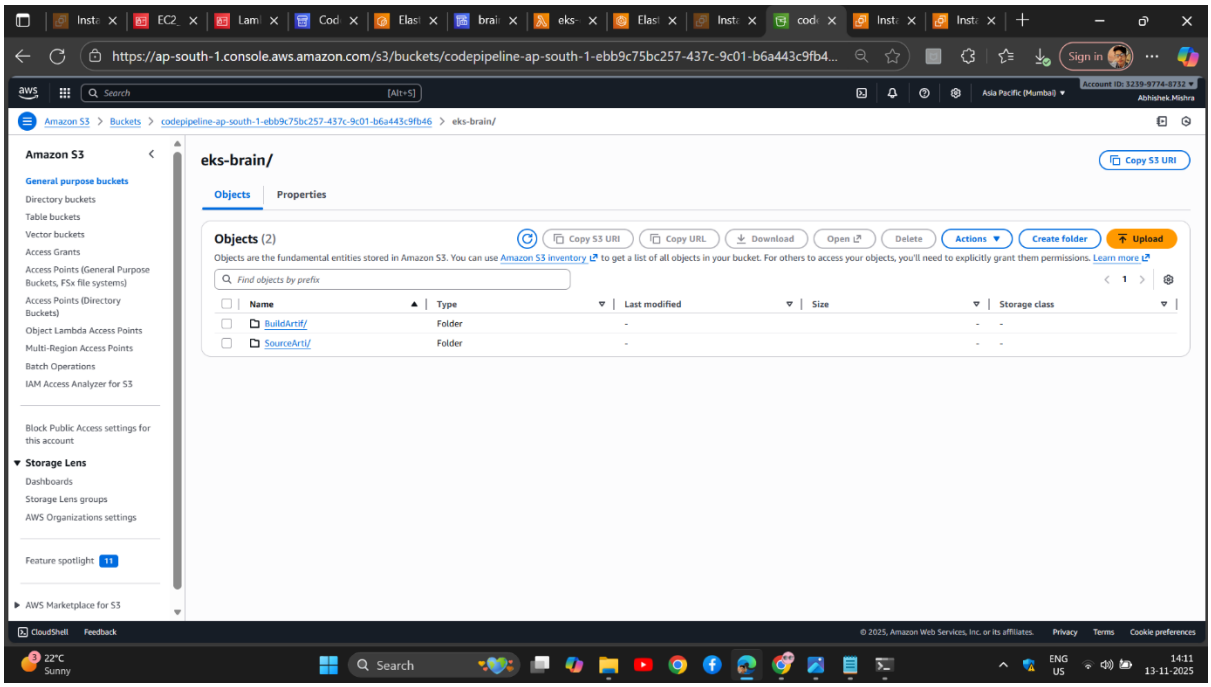
1. Developer pushes code to GitHub repository
2. CodePipeline detects the commit (webhook trigger)
3. CodeBuild executes buildspec.yml:
    - Authenticates with ECR
    - Builds Docker image
    - Pushes image to ECR
    - Creates artifacts
4. CodeDeploy triggers deployment:
    - Executes BeforeInstall hook (cleanup.sh)
    - Invokes Lambda function
    - Lambda updates EKS cluster
    - Executes AfterInstall hook (deploy.sh)
5. EKS updates running pods with new image
6. LoadBalancer routes traffic to new pods

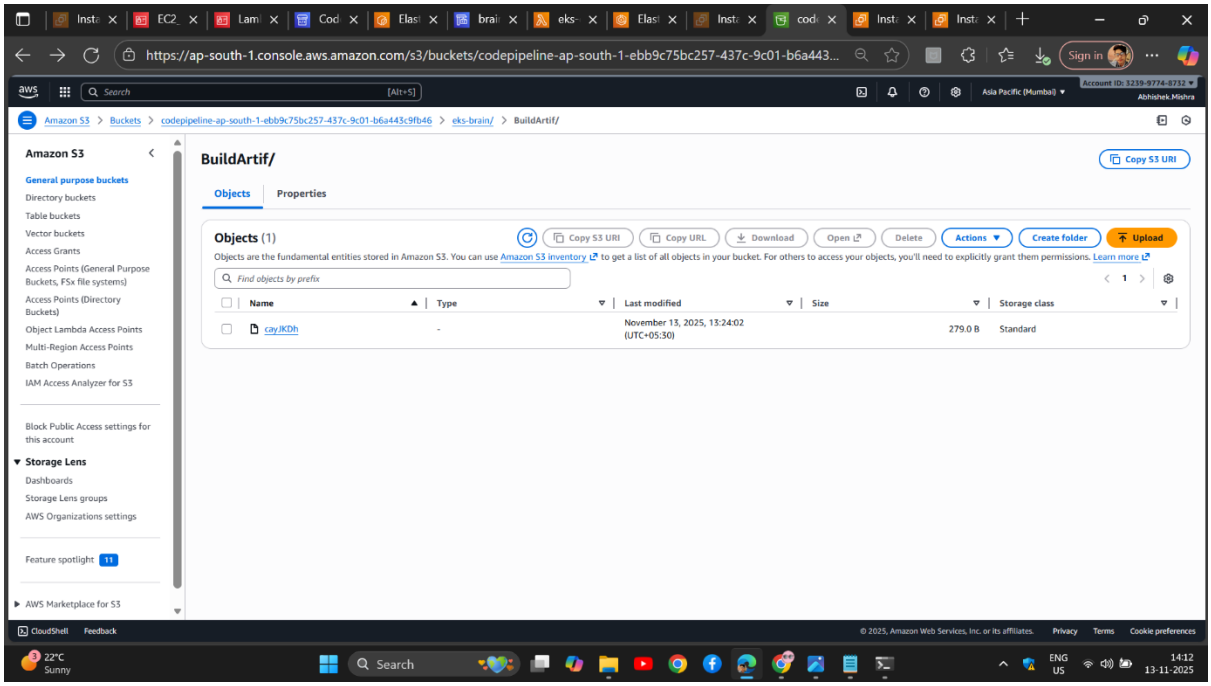
Output final :

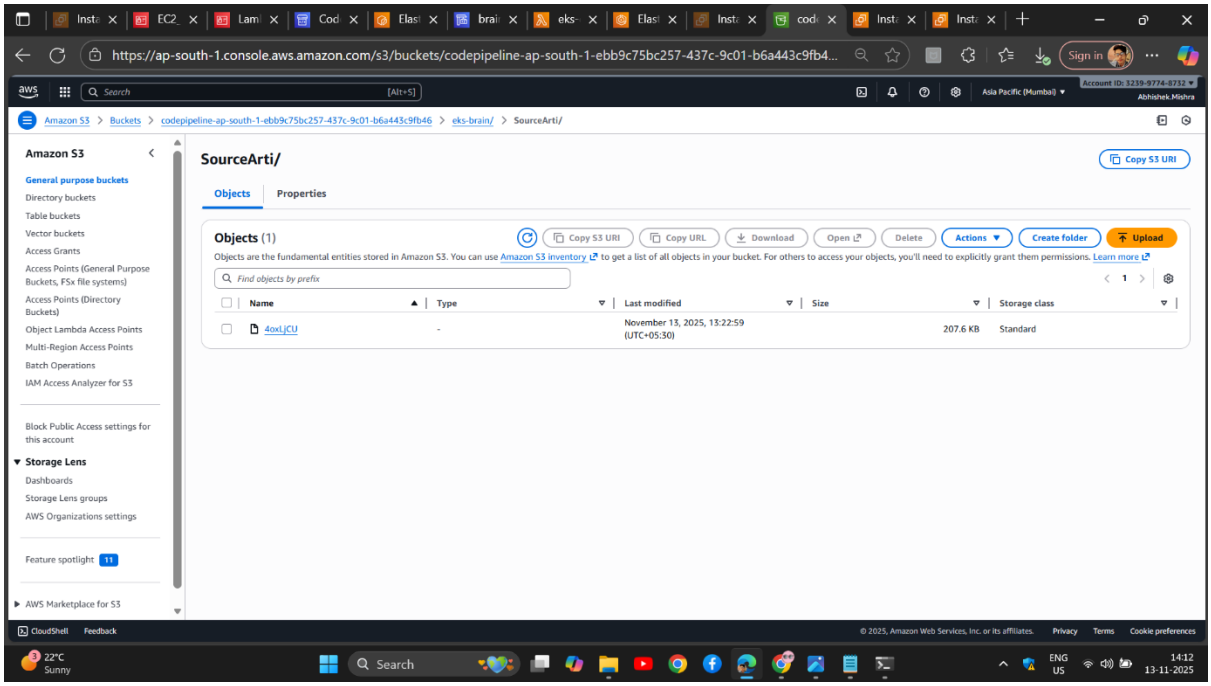


Using s3 for storing the artifact.

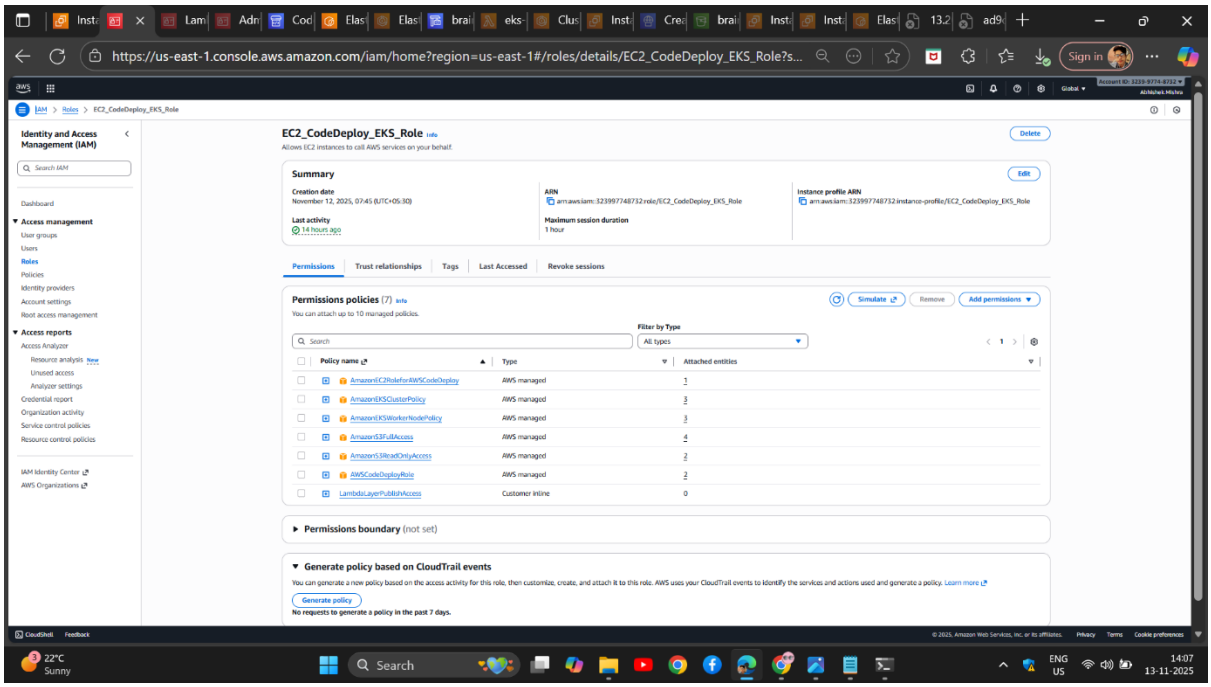


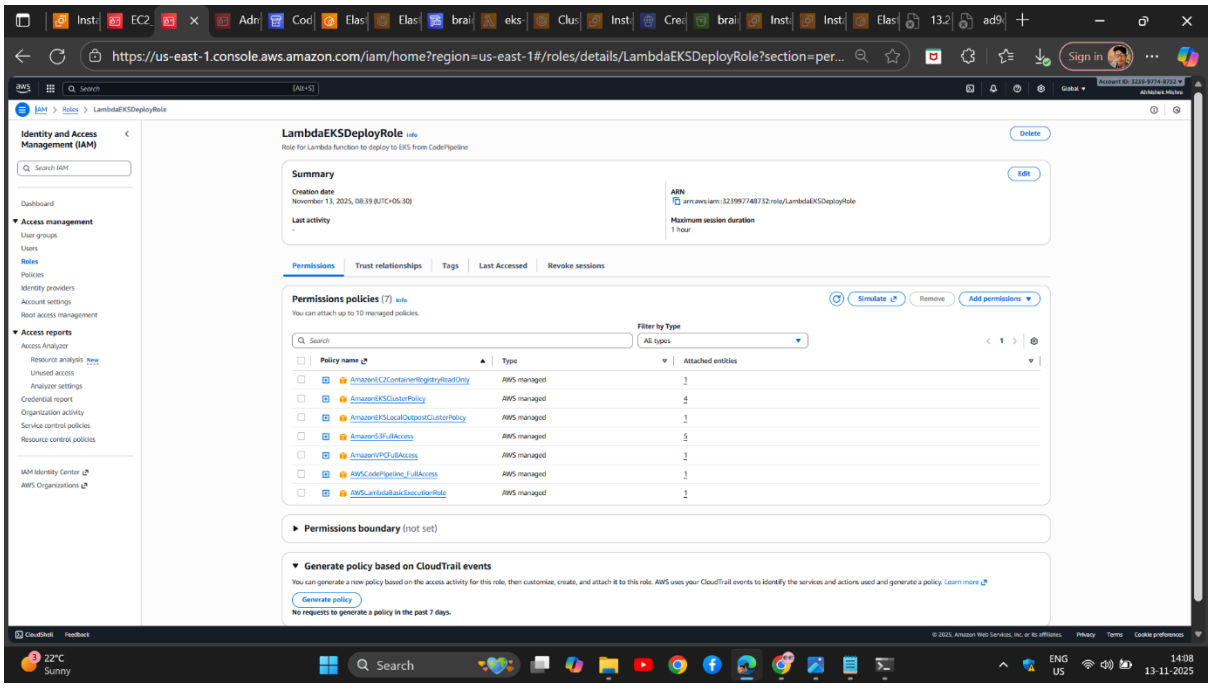



https://ap-south-1.console.aws.amazon.com/s3/buckets/codepipeline-ap-south-1-ebb9c75bc257-437c-9c01-b6a443...
Amazon S3 > Buckets > codepipeline-ap-south-1-ebb9c75bc257-437c-9c01-b6a443c9fb46 > eks-brain/ > BuildArtif/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
BuildArtif/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
cayJKDh
November 13, 2025, 13:24:02 (UTC+05:30)
279.0 B
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
22°C Sunny
Search
ENG US
14:12 13-11-2025

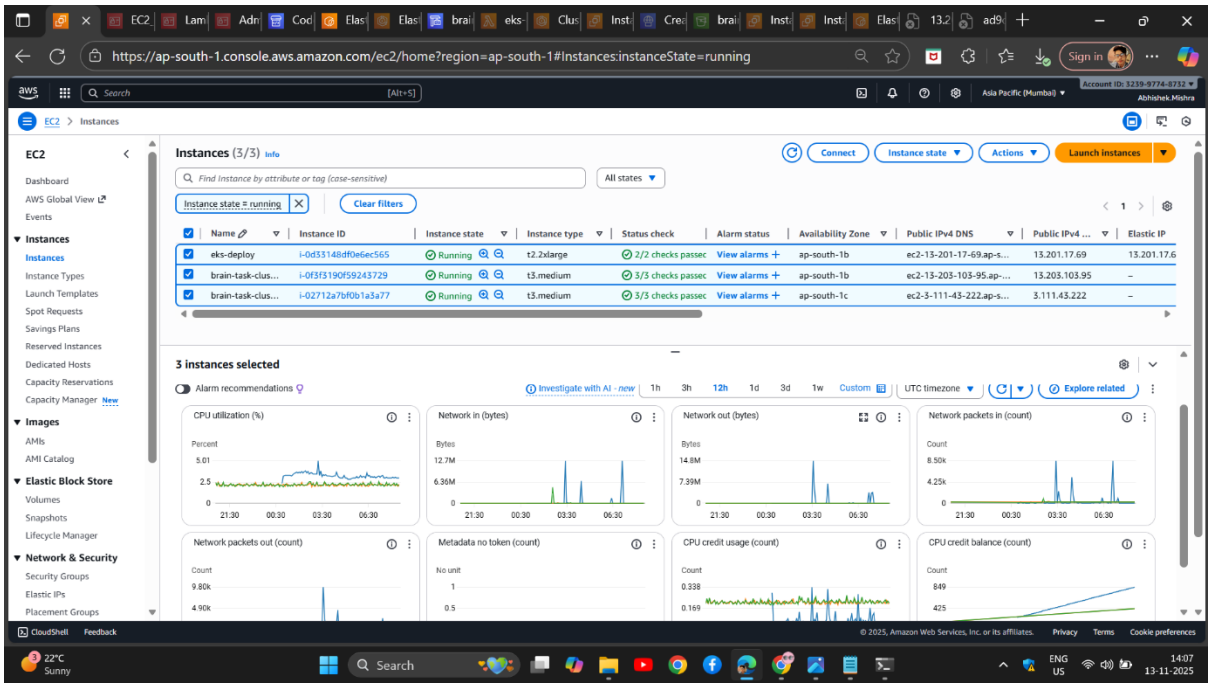

https://ap-south-1.console.aws.amazon.com/s3/buckets/codepipeline-ap-south-1-ebb9c75bc257-437c-9c01-b6a443c9fb4...
Amazon S3 > Buckets > codepipeline-ap-south-1-ebb9c75bc257-437c-9c01-b6a443c9fb46 > eks-brain/ > SourceArti/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
SourceArti/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
4oxLJCU
November 13, 2025, 13:22:59 (UTC+05:30)
207.6 KB
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
22°C Sunny
Search
ENG US
14:12 13-11-2025

Roles for different services

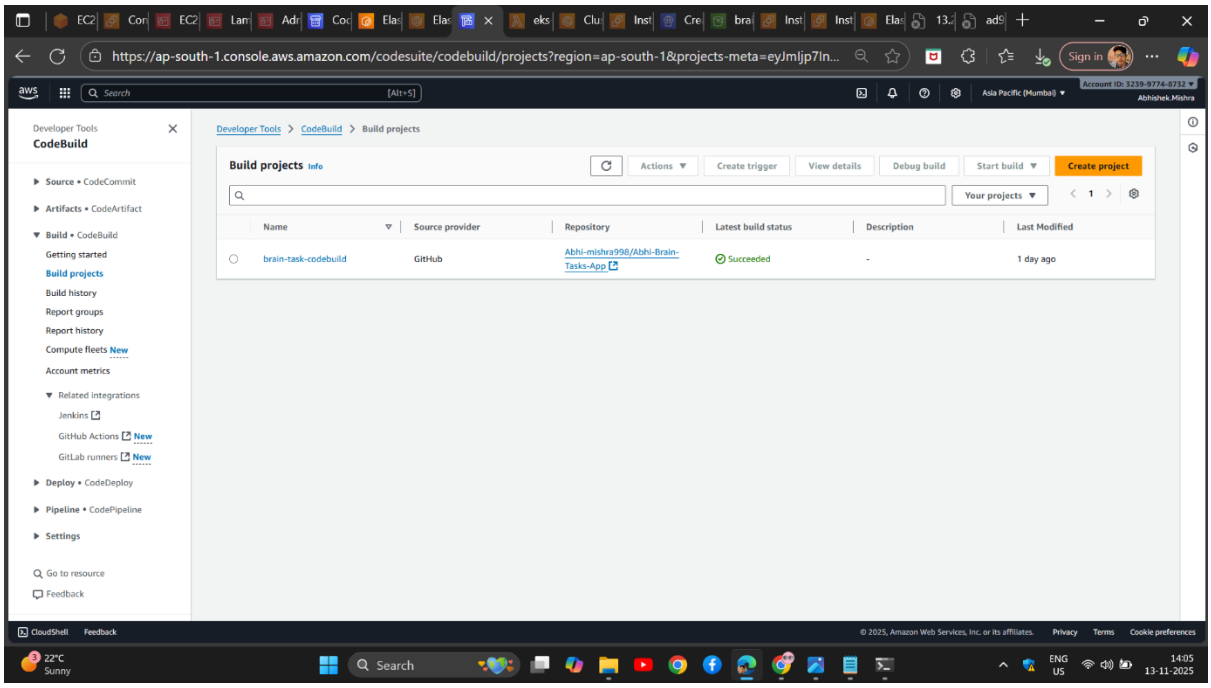


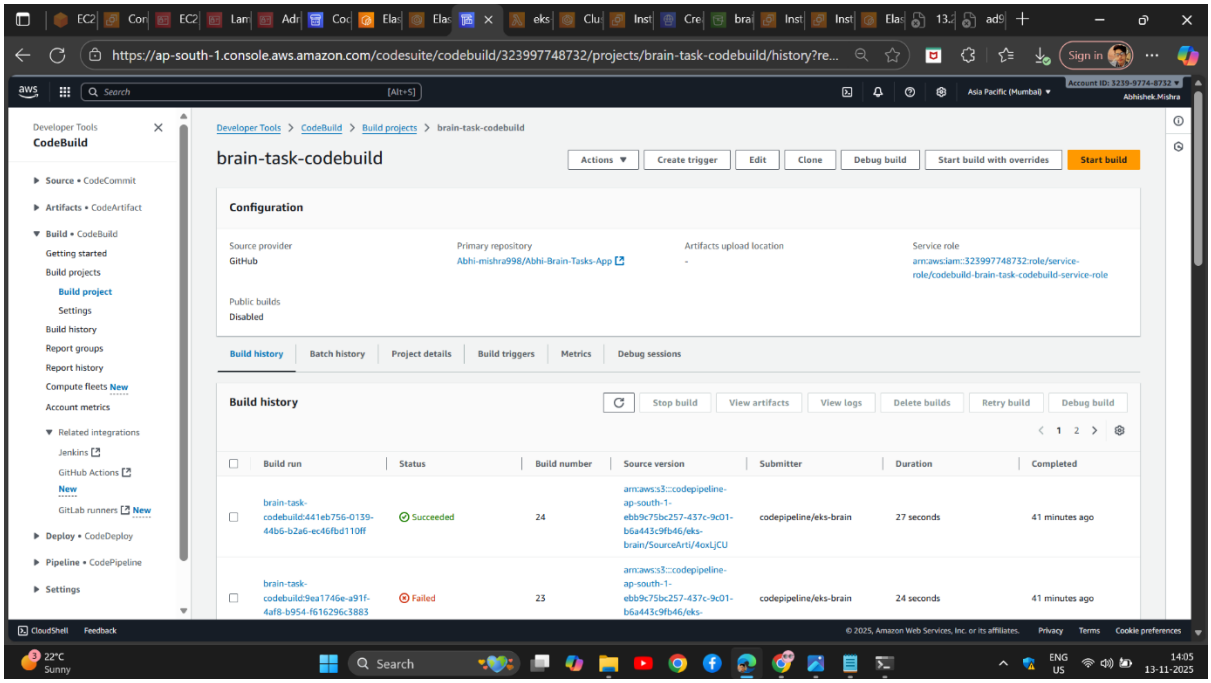


Two instances for the cluster one for my particle:

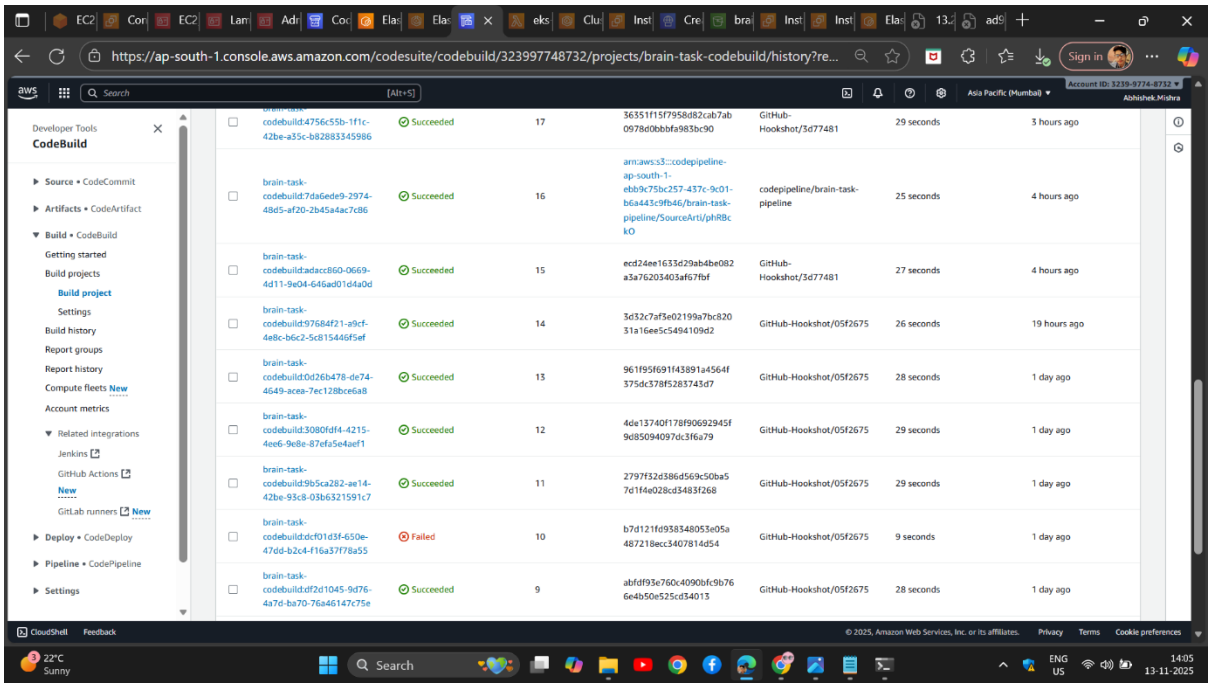


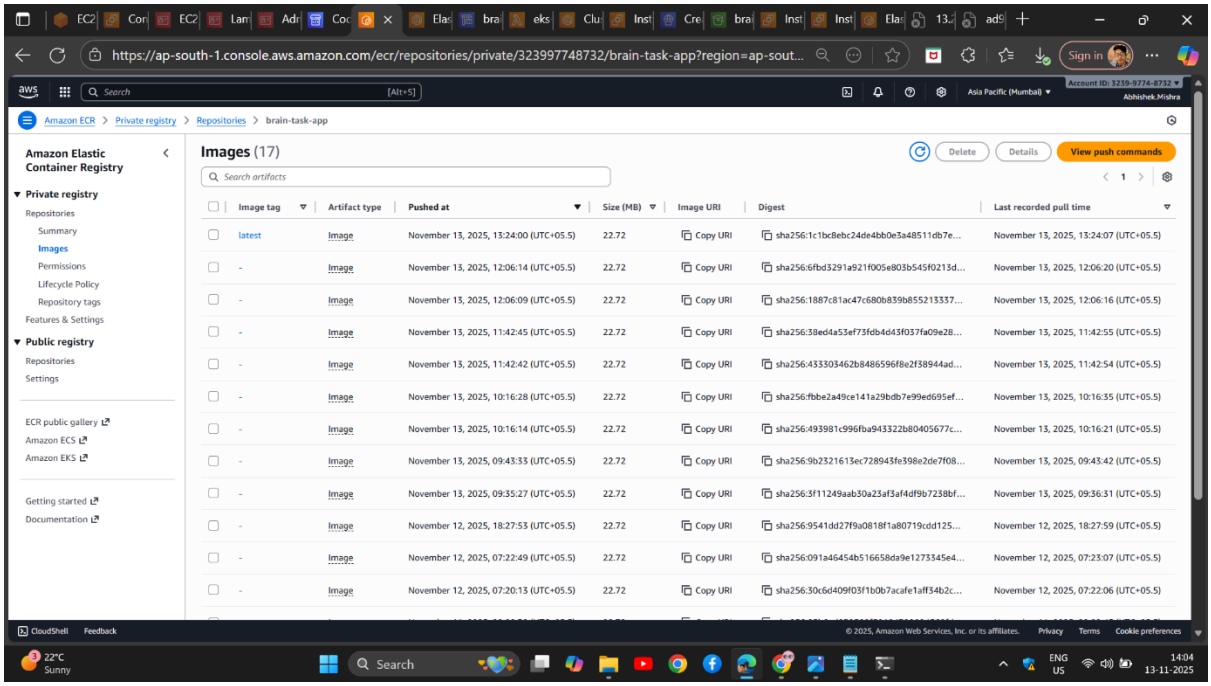
Codebuild setup:



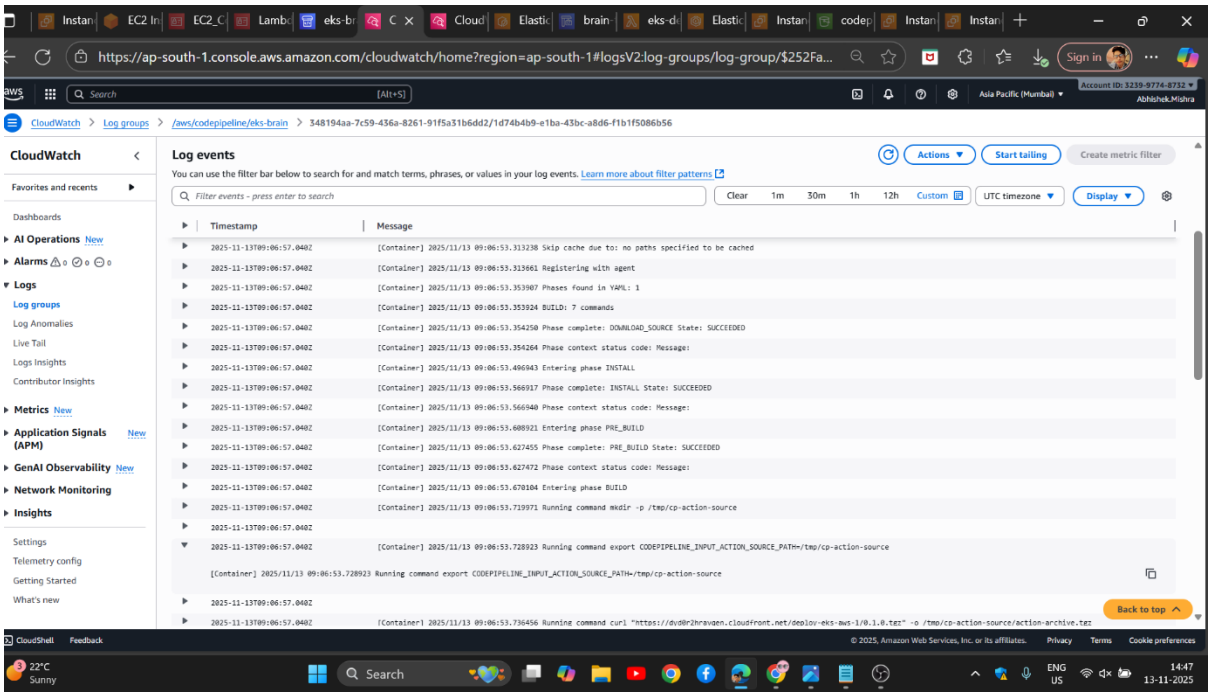


Testing multiple time to see it working fine or not

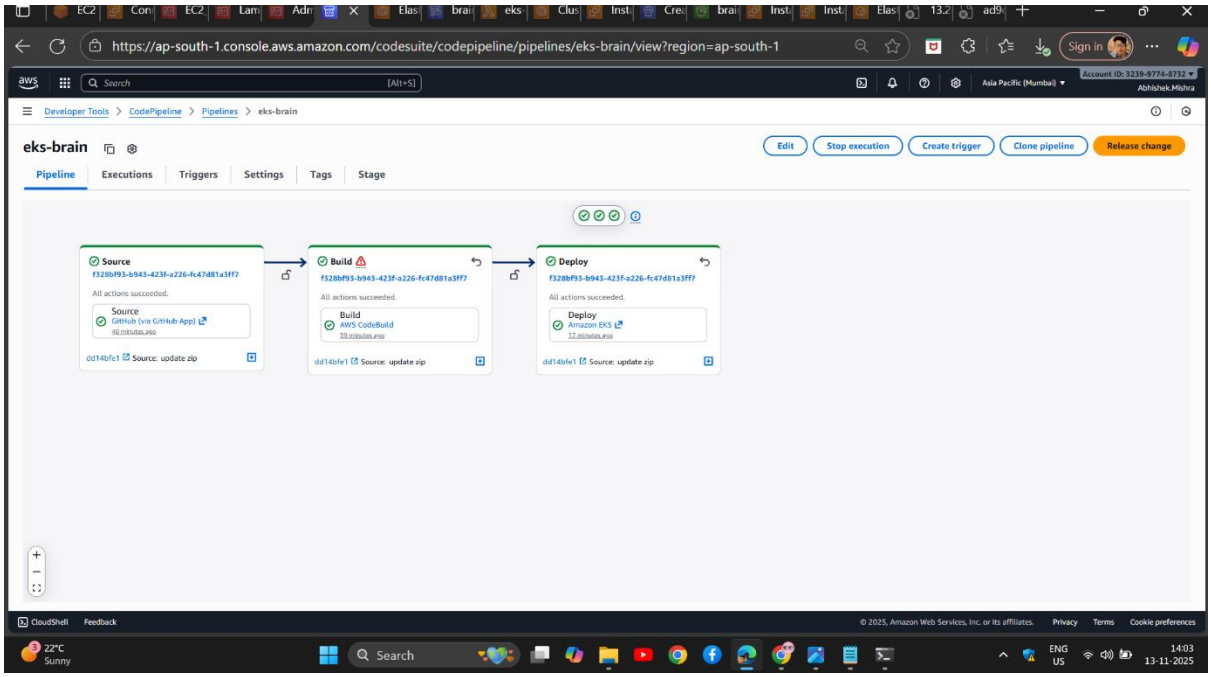


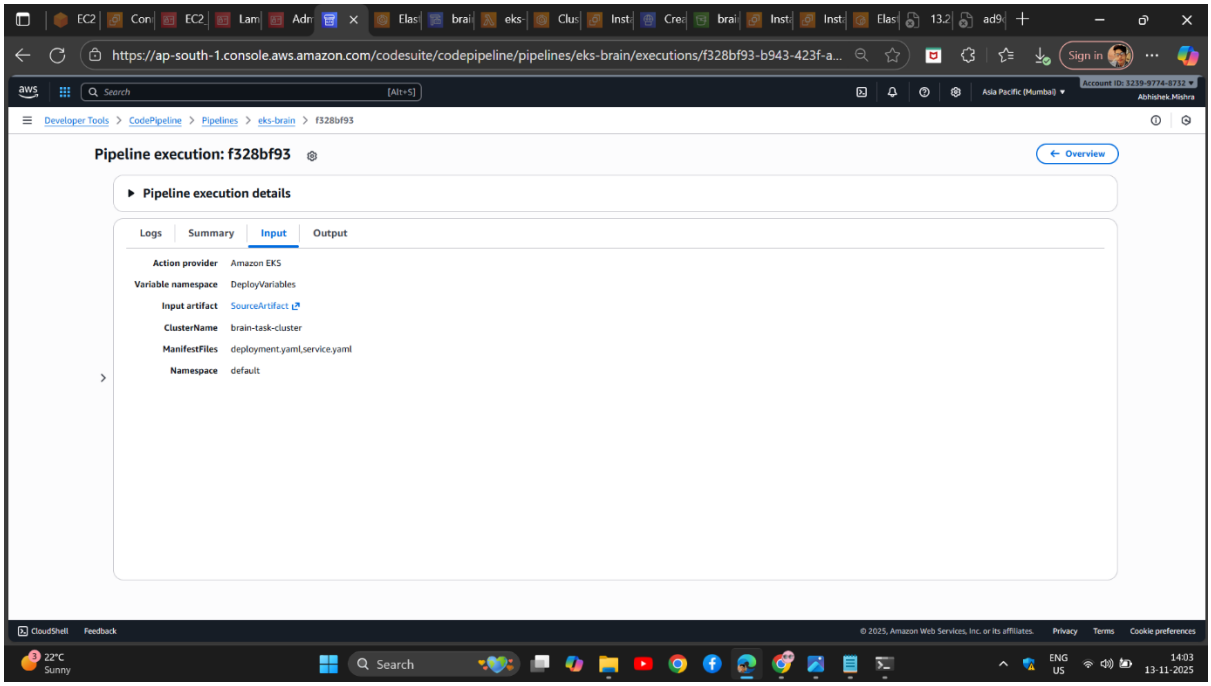


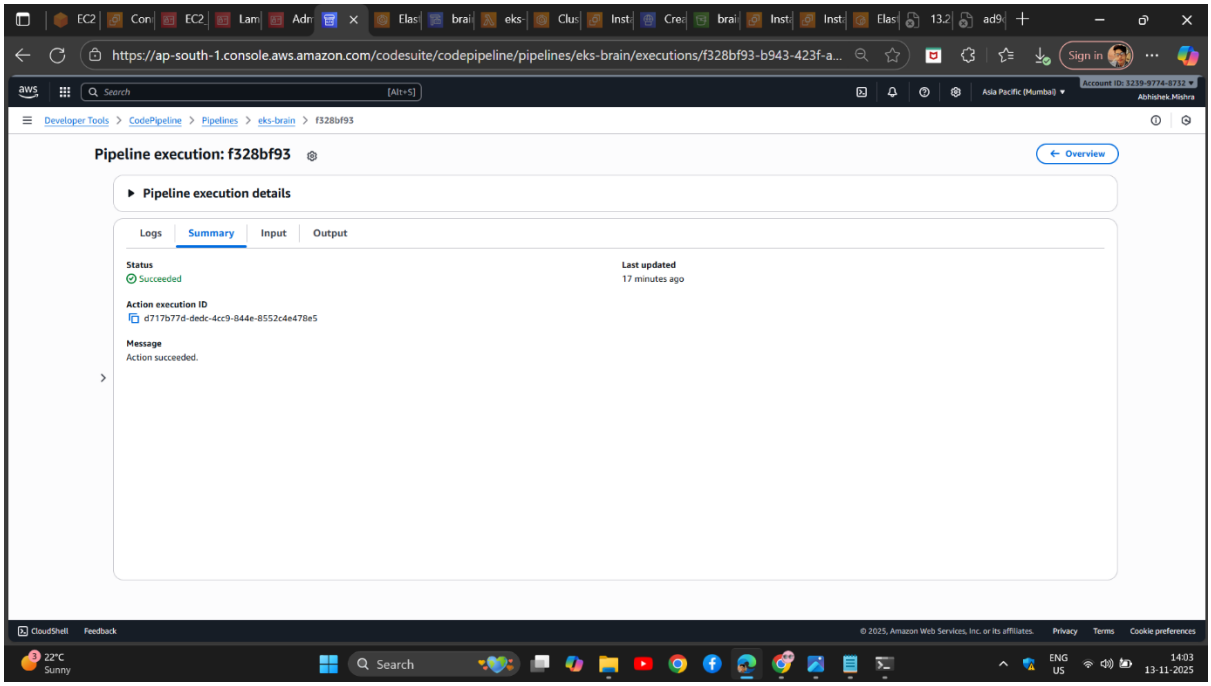
Checking CloudWatch logs

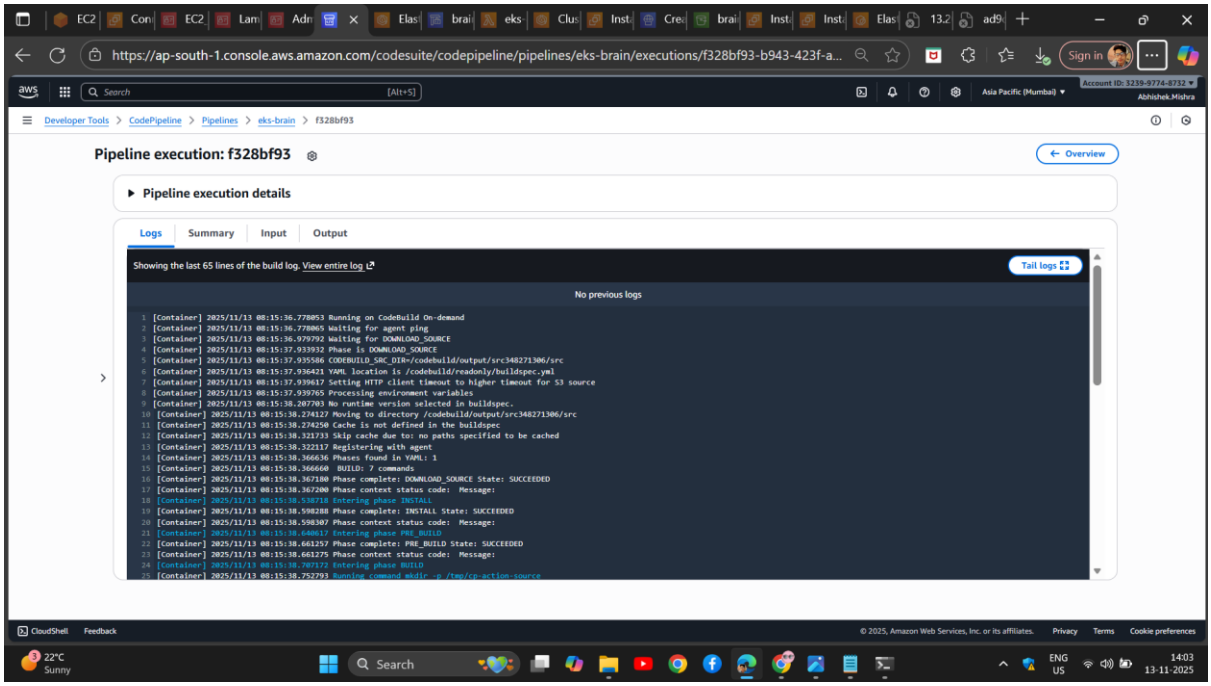


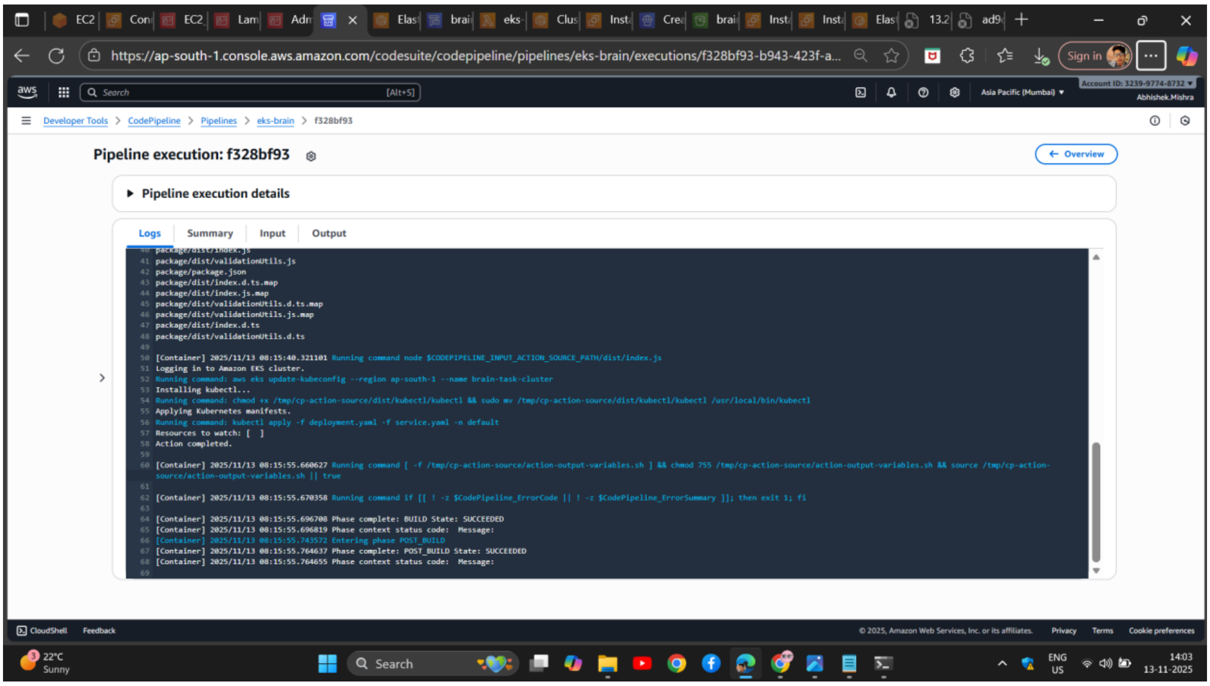
CodePipeline working I test it multiple time also checking it rollback:

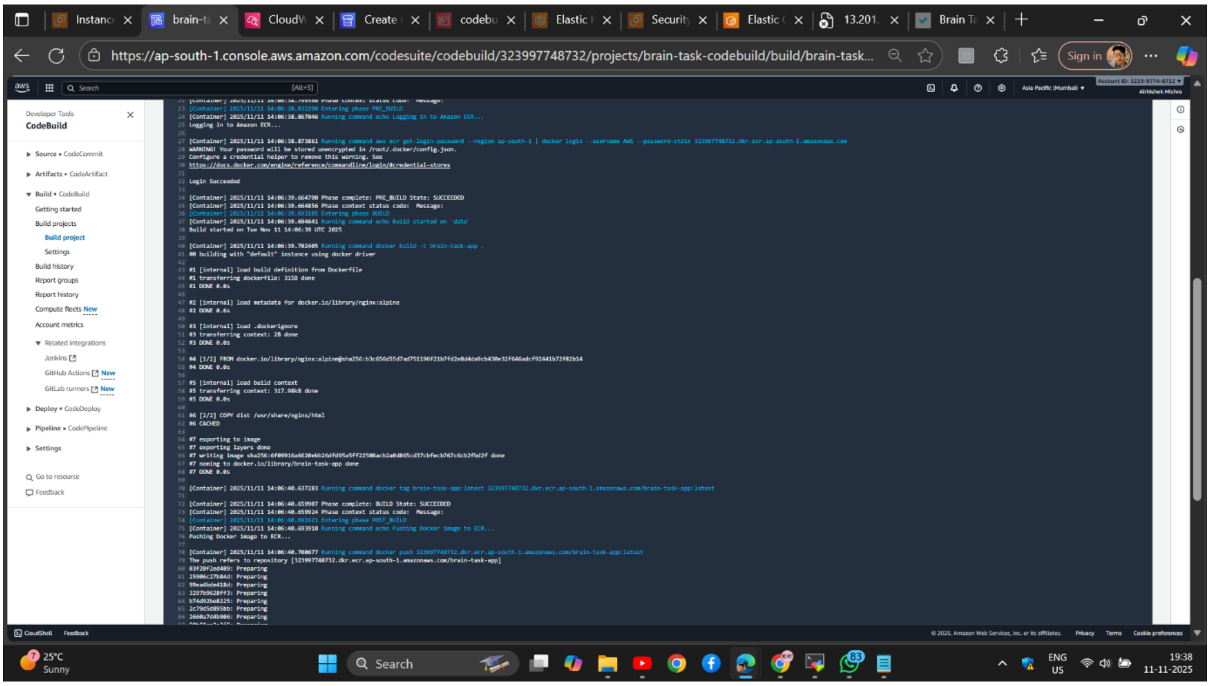


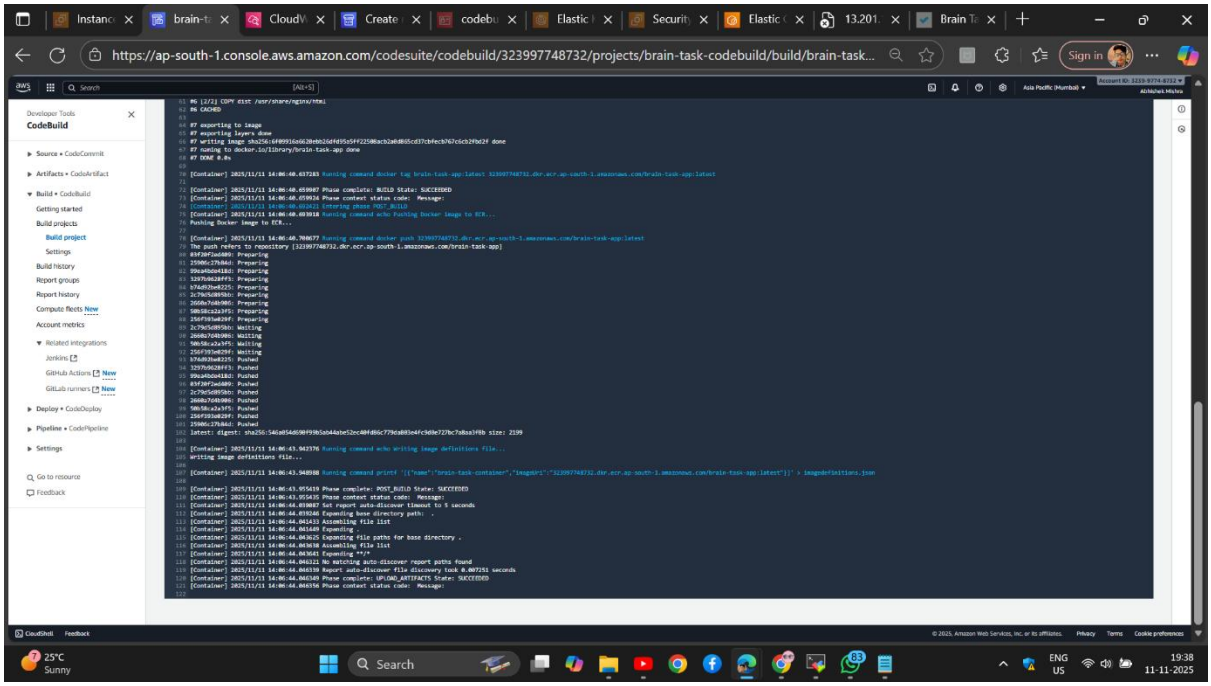


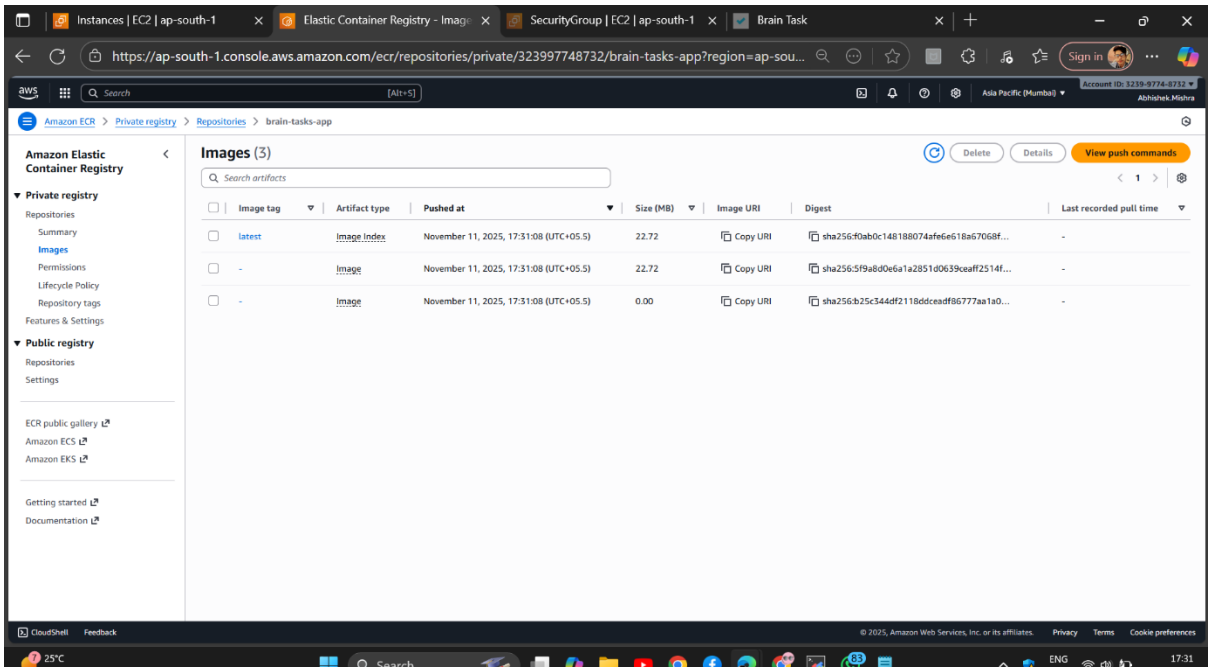

https://ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/eks-brain/executions/f328bf93-b943-423f-a...
Developer Tools > CodePipeline > Pipelines > eks-brain > f328bf93
Pipeline execution: f328bf93
Overview
Pipeline execution details
Logs
Summary
Input
Output
Status
Succeeded
Last updated
17 minutes ago
Action execution ID
d717b77d-dedc-4cc9-844e-8552c4e478e5
Message
Action succeeded.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
22°C
Sunny
Search
ENG
US
14:03
13-11-2025

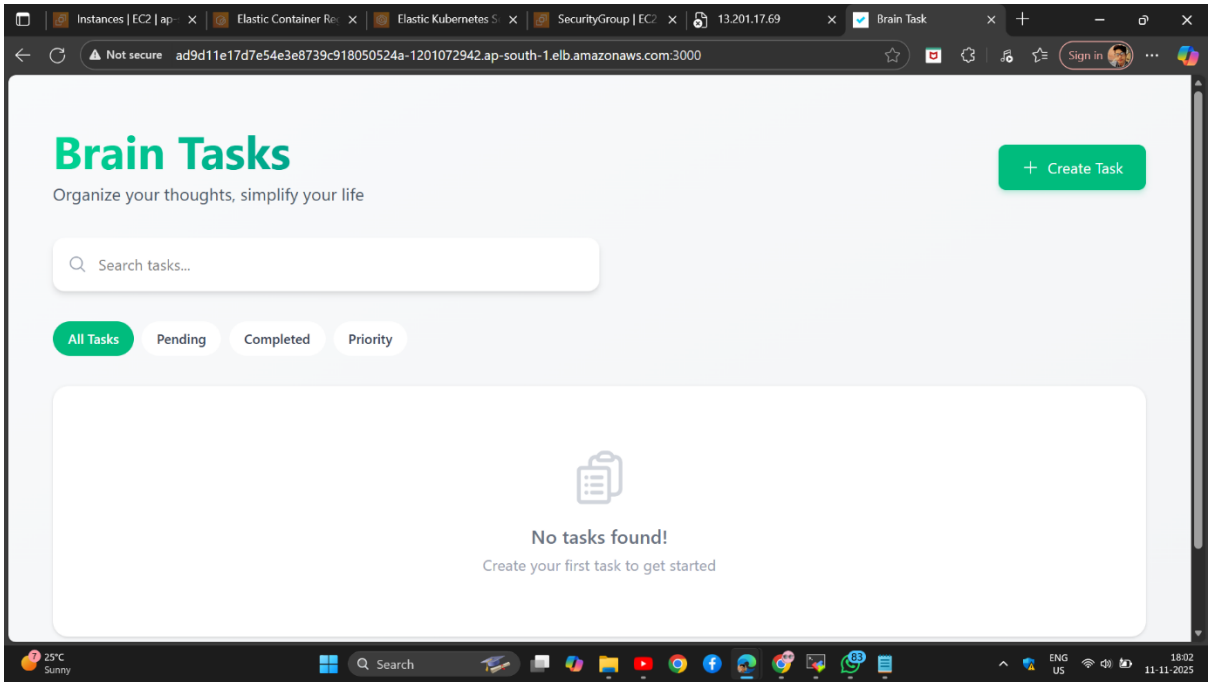

https://ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/eks-brain/executions/f328bf93-b943-423f-a...
Developer Tools > CodePipeline > Pipelines > eks-brain > f328bf93
Pipeline execution: f328bf93
Overview
Pipeline execution details
Logs
Summary
Input
Output
Showing the last 65 lines of the build log. View entire log
Tail logs
No previous logs
[Container] 2025/11/13 08:15:36.778053 Running on CodeBuild On-demand
[Container] 2025/11/13 08:15:36.778065 Waiting for agent ping
[Container] 2025/11/13 08:15:36.979792 Waiting for DOWNLOAD_SOURCE
[Container] 2025/11/13 08:15:37.933932 Phase is DOWNLOAD_SOURCE
[Container] 2025/11/13 08:15:37.935586 CODEBUILD_SRC_DIR=/codebuild/output/src348271306/src
[Container] 2025/11/13 08:15:37.936421 YAML location is /codebuild/readonly/buildspec.yml
[Container] 2025/11/13 08:15:37.939617 Setting HTTP client timeout to higher timeout for S3 source
[Container] 2025/11/13 08:15:37.939765 Processing environment variables
[Container] 2025/11/13 08:15:38.207703 No runtime version selected in buildspec.
[Container] 2025/11/13 08:15:38.274127 Moving to directory /codebuild/output/src348271306/src
[Container] 2025/11/13 08:15:38.274250 Cache is not defined in the buildspec
[Container] 2025/11/13 08:15:38.321733 Skip cache due to: no paths specified to be cached
[Container] 2025/11/13 08:15:38.322117 Registering with agent
[Container] 2025/11/13 08:15:38.366636 Phases found in YAML: 1
[Container] 2025/11/13 08:15:38.366660 BUILD: 7 commands
[Container] 2025/11/13 08:15:38.367180 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] 2025/11/13 08:15:38.367200 Phase context status code: Message:
[Container] 2025/11/13 08:15:38.530718 Entering phase INSTALL
[Container] 2025/11/13 08:15:38.598288 Phase complete: INSTALL State: SUCCEEDED
[Container] 2025/11/13 08:15:38.598307 Phase context status code: Message:
[Container] 2025/11/13 08:15:38.640617 Entering phase PRE_BUILD
[Container] 2025/11/13 08:15:38.661257 Phase complete: PRE_BUILD State: SUCCEEDED
[Container] 2025/11/13 08:15:38.661275 Phase context status code: Message:
[Container] 2025/11/13 08:15:38.707172 Entering phase BUILD
[Container] 2025/11/13 08:15:38.752793 Running command mkdir -p /tmp/cp-action-source
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
22°C
Sunny
Search
ENG
US
14:03
13-11-2025


https://ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/eks-brain/executions/f328bf93-b943-423f-a...
Developer Tools > CodePipeline > Pipelines > eks-brain > f328bf93
Pipeline execution: f328bf93
Overview
Pipeline execution details
Logs
Summary
Input
Output
package/dist/validationUtils.js
package/package.json
package/dist/index.d.ts.map
package/dist/index.js.map
package/dist/validationUtils.d.ts.map
package/dist/validationUtils.js.map
package/dist/index.d.ts
package/dist/validationUtils.d.ts
[Container] 2025/11/13 08:15:40.321101 Running command node $CODEPIPELINE_INPUT_ACTION_SOURCE_PATH/dist/index.js
Logging in to Amazon EKS cluster.
Running command: aws eks update-kubeconfig --region ap-south-1 --name brain-task-cluster
Installing kubectl...
Running command: chmod +x /tmp/cp-action-source/dist/kubectl/kubectl && sudo mv /tmp/cp-action-source/dist/kubectl/kubectl /usr/local/bin/kubectl
Applying Kubernetes manifests.
Running command: kubectl apply -f deployment.yaml -f service.yaml -n default
Resources to watch: [ ]
Action completed.
[Container] 2025/11/13 08:15:55.660627 Running command [-f /tmp/cp-action-source/action-output-variables.sh] && chmod 755 /tmp/cp-action-source/action-output-variables.sh && source /tmp/cp-action-source/action-output-variables.sh || true
[Container] 2025/11/13 08:15:55.670358 Running command if [[! -z $CodePipeline_ErrorCode || ! -z $CodePipeline_ErrorSummary]]; then exit 1; fi
[Container] 2025/11/13 08:15:55.696708 Phase complete: BUILD State: SUCCEEDED
[Container] 2025/11/13 08:15:55.696819 Phase context status code: Message:
[Container] 2025/11/13 08:15:55.743572 Entering phase POST_BUILD
[Container] 2025/11/13 08:15:55.764637 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2025/11/13 08:15:55.764655 Phase context status code: Message:
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
22°C
Sunny
14:03
13-11-2025

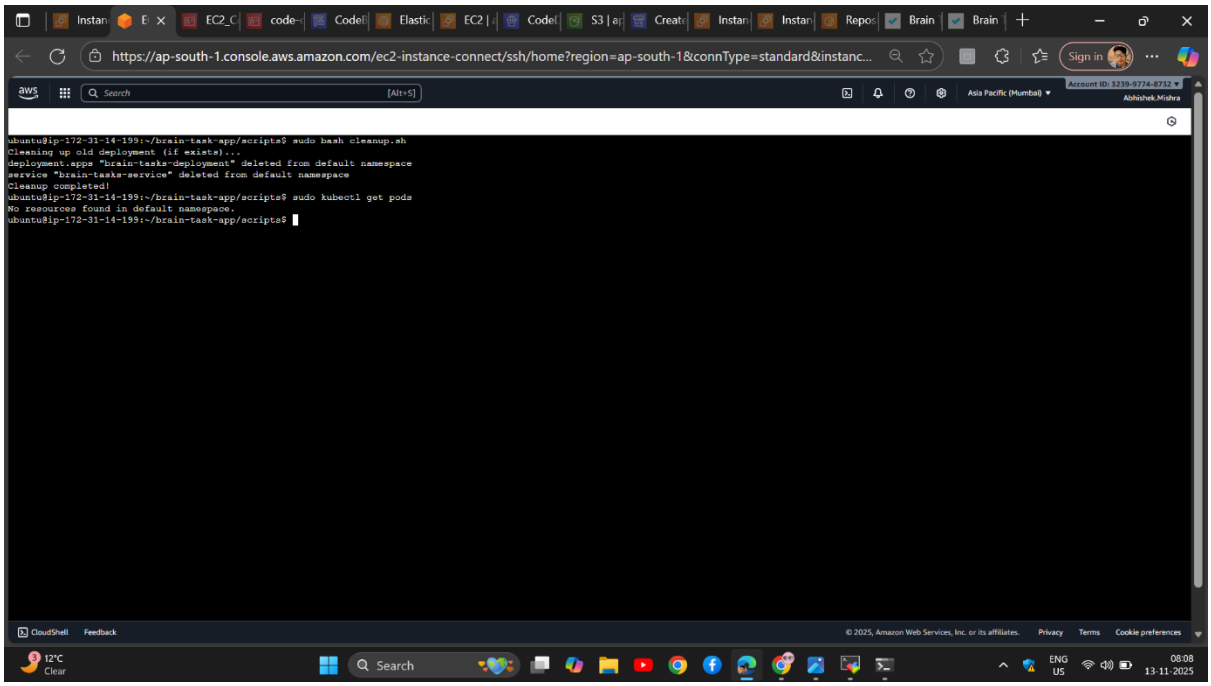

https://ap-south-1.console.aws.amazon.com/codesuite/codebuild/323997748732/projects/brain-task-codebuild/build/brain-task...
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets
Account metrics
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
25°C
Sunny
19:38
11-11-2025

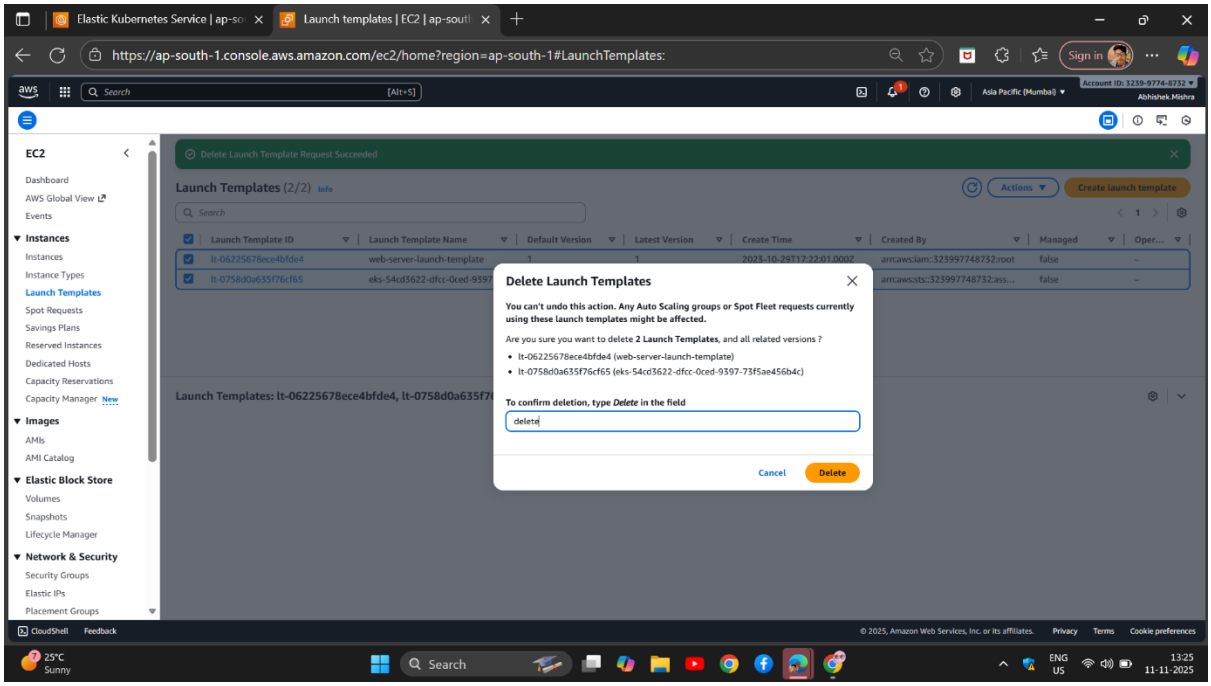

https://ap-south-1.console.aws.amazon.com/codesuite/codebuild/323997748732/projects/brain-task-codebuild/build/brain-task...
CodeBuild
Build project
Search
25°C
Sunny
ENG
US
19:38
11-11-2025


Instances | EC2 | ap-south-1
Elastic Container Registry - Image
SecurityGroup | EC2 | ap-south-1
Brain Task
https://ap-south-1.console.aws.amazon.com/ecr/repositories/private/323997748732/brain-tasks-app?region=ap-sou...
Amazon ECR > Private registry > Repositories > brain-tasks-app
Amazon Elastic Container Registry
Private registry
Repositories
Summary
Images
Permissions
Lifecycle Policy
Repository tags
Features & Settings
Public registry
Repositories
Settings
ECR public gallery
Amazon ECS
Amazon EKS
Getting started
Documentation
Images (3)
Search artifacts
Delete
Details
View push commands
Image tag
Artifact type
Pushed at
Size (MB)
Image URI
Digest
Last recorded pull time
latest
Image Index
November 11, 2025, 17:31:08 (UTC+05.5)
22.72
Copy URI
sha256:f0ab0c148188074afe6e618a67068f...
Image
sha256:5f9a8d0e6a1a2851d0639ceaff2514f...
sha256:b25c344df2118ddceadf86777aa1a0...
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
ENG
17:31



Cleanup everything





GitHub-repo: https://github.com/Abhi-mishra998/Abhi-Brain-Tasks-App.git