# Brain Tasks App - Production Deployment (AWS EKS)

This document contains setup instructions and a pipeline explanation for deploying the React-based Brain Tasks App using AWS services including ECR, EKS, CodeBuild, and CloudWatch.

## ✈️ Setup Instructions

### 1. Clone the Repository

git clone https://github.com/parthiban4626/Brain-Tasks-App.git  
cd Brain-Tasks-App

### 2. Create an ECR Repository

aws ecr create-repository --repository-name brain-task-app

Note the ECR URI: 977099019525.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app

### 3. Set up EKS Cluster

eksctl create cluster --name brain-task-cluster --region ap-south-1 --nodes 2 --node-type t3.medium

### 4. Enable OIDC Provider

eksctl utils associate-iam-oidc-provider --cluster brain-task-cluster --approve

### 5. IAM Role for Fluent Bit

* Create a trust-policy.json file
* Create the role and attach policy:

aws iam create-role \  
 --role-name FluentBitIAMRole \  
 --assume-role-policy-document file://trust-policy.json  
  
aws iam attach-role-policy \  
 --role-name FluentBitIAMRole \  
 --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy

### 6. Apply Kubernetes Manifests

kubectl apply -f deployment.yaml  
kubectl apply -f service.yaml  
kubectl apply -f fluent-bit.yaml

### 7. Create CodeBuild Project

* Source: GitHub
* Environment: Amazon Linux 2 with Docker
* Attach ECR and EKS permissions
* Use buildspec.yml in root directory

### 8. Create CodePipeline

* Source: GitHub (connect repo)
* Build: Link to the above CodeBuild project
* Deploy: Done within CodeBuild using kubectl

## 🌐 Pipeline Explanation

### Source Stage:

* Trigger: Code pushed to GitHub
* Action: CodePipeline detects change via GitHub Webhook

### Build Stage (CodeBuild):

* Authenticates with ECR using IAM role
* Builds Docker image from Dockerfile
* Tags and pushes image to ECR
* Uses kubectl apply to deploy app to EKS using deployment.yaml and service.yaml

### Deploy Stage:

* Managed within CodeBuild (no separate CodeDeploy used)
* Executes kubectl commands directly to update EKS cluster

### Monitoring:

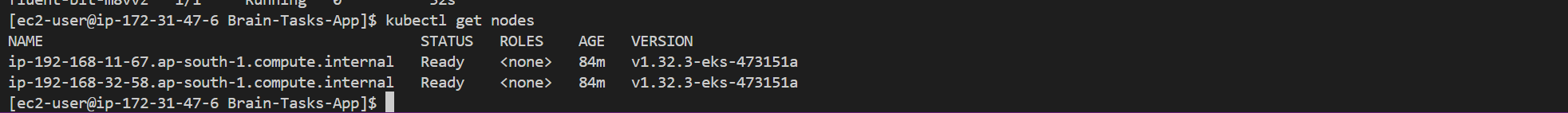
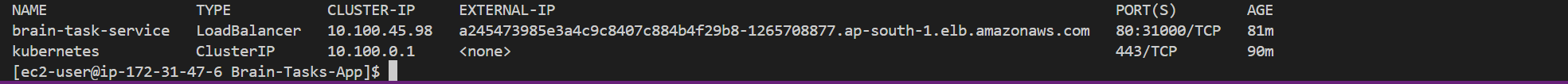
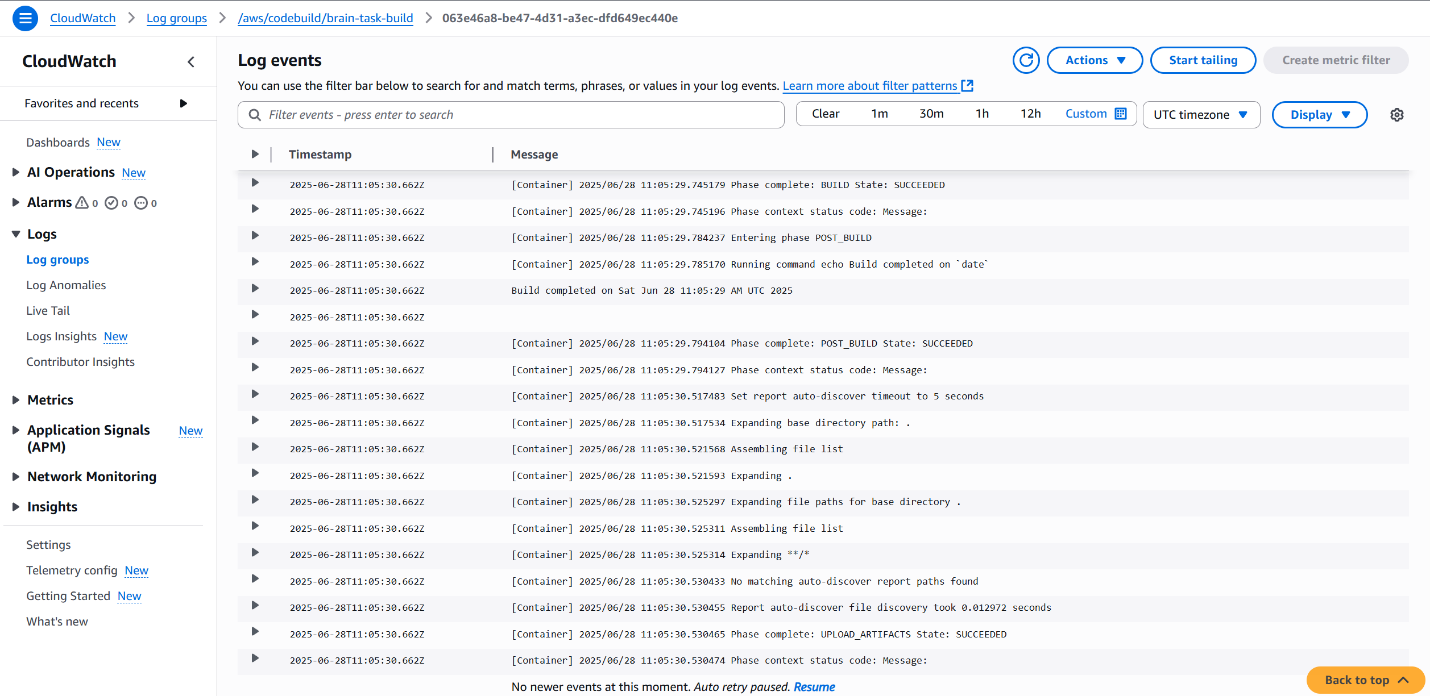
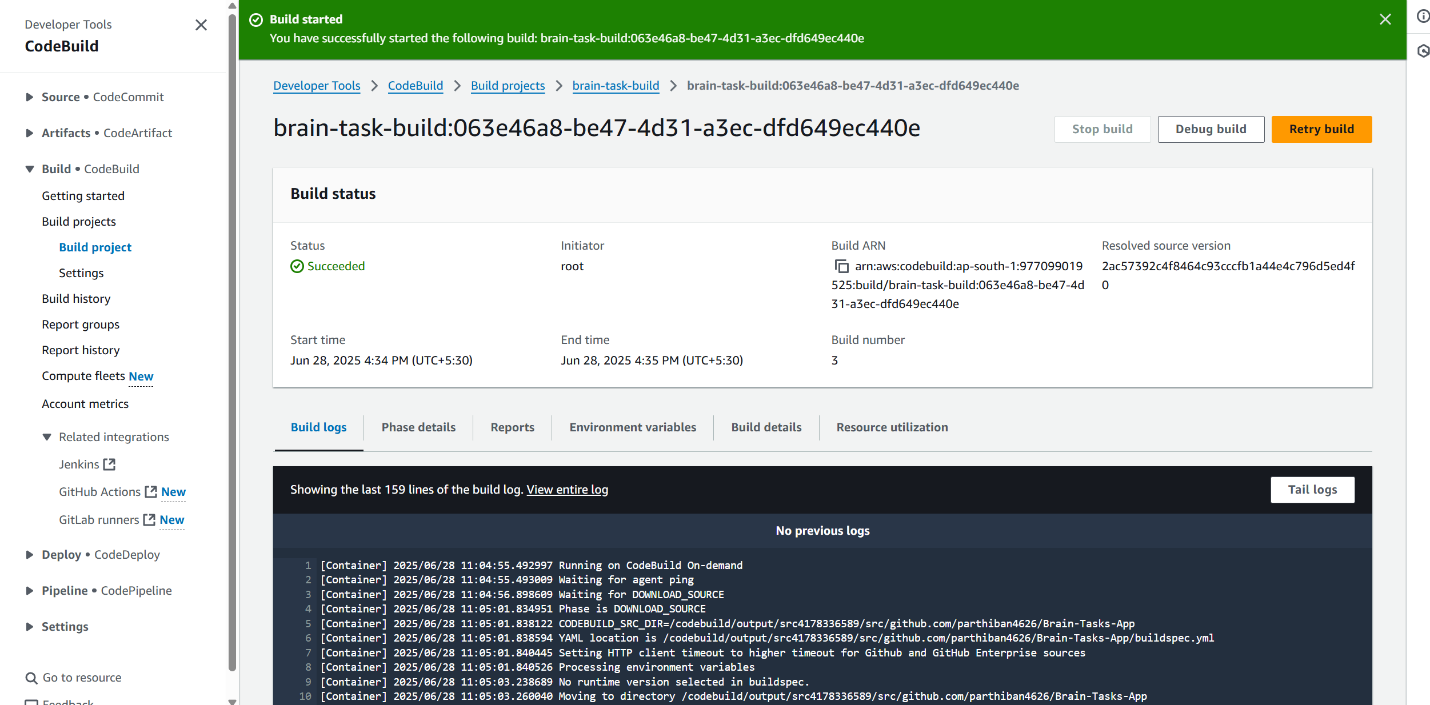
* Fluent Bit sidecar streams logs to AWS CloudWatch
* Logs are visible in CloudWatch Log Groups under EKS cluster

## 🔗 Accessing the Application

* Run kubectl get svc
* Note the EXTERNAL-IP of the LoadBalancer
* Open in browser:

http://a245473985e3a4c9c8407c884b4f29b8-1265708877.ap-south-1.elb.amazonaws.com

## 📷 Screenshots for Submission

* EKS nodes in Ready state
* LoadBalancer EXTERNAL-IP
* CloudWatch Logs streaming output
* Successful CodeBuild build

## 📄 Files Included

* Dockerfile: React app Dockerization
* deployment.yaml: Kubernetes Deployment spec
* service.yaml: LoadBalancer Service
* buildspec.yml: CodeBuild script
* fluent-bit.yaml: Log shipping config
* trust-policy.json: IAM Trust policy for Fluent Bit

## ✅ Deployment Complete

* Application deployed via EKS LoadBalancer
* Image pushed to ECR
* Monitoring enabled with Fluent Bit and CloudWatch