

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

The screenshot shows a GitHub commit for the file 'Dockerfile' in the repository 'Abhi-Brain-Tasks-App'. The commit message is 'chore: fix dist folder tracking and include React build files'. The Dockerfile contains the following code:

```

1 # Step 1: Use official Nginx image
2 FROM nginx:alpine
3
4 # Step 2: Copy build files from dist folder to nginx html directory
5 COPY dist /usr/share/nginx/html
6
7 # Step 3: Expose port 80 (default for nginx)
8 EXPOSE 80
9
10 # Step 4: Start Nginx server
11 CMD ["nginx", "-g", "daemon off;"]

```

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

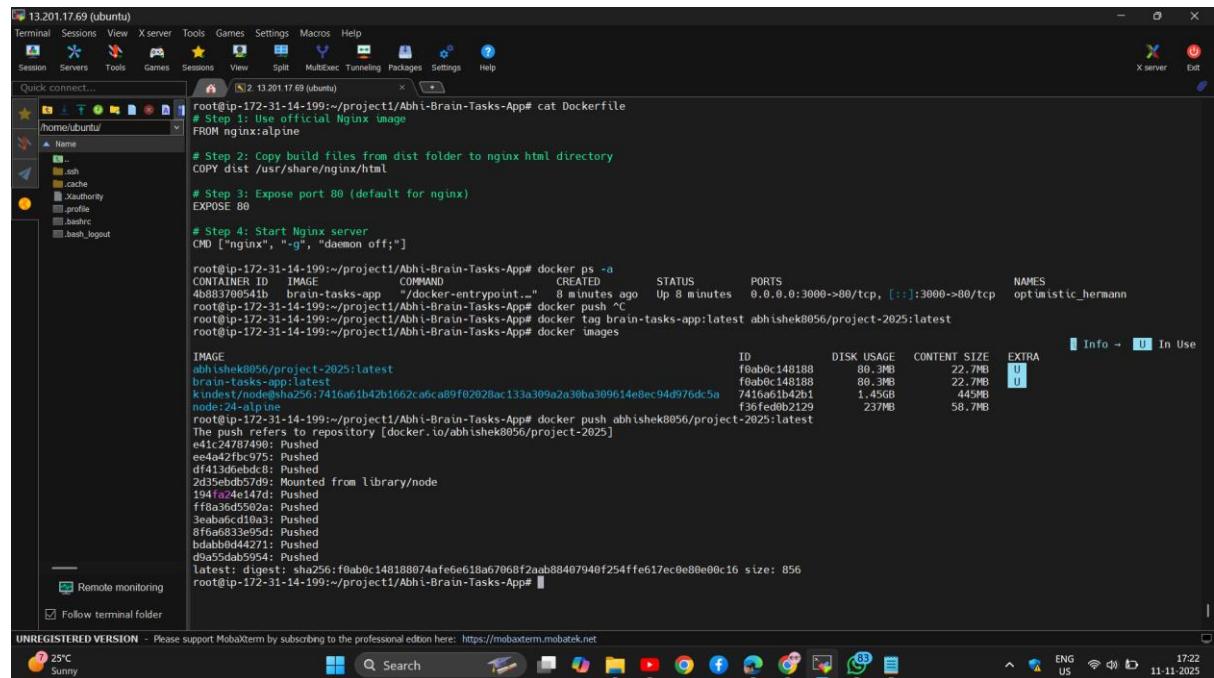
• Build Docker Image Locally

- # 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:



```
# Step 1: Use official Nginx image
FROM nginx:alpine

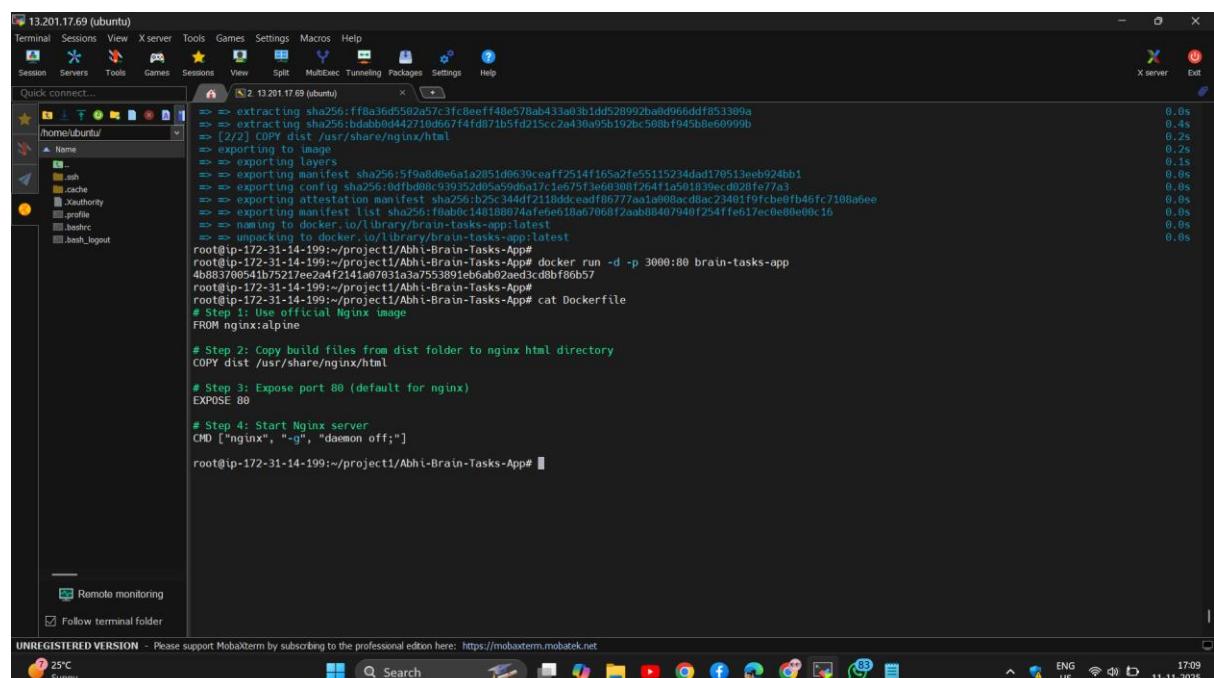
# Step 2: Copy build files from dist folder to nginx html directory
COPY dist /usr/share/nginx/html

# Step 3: Expose port 80 (default for nginx)
EXPOSE 80

# Step 4: Start Nginx server
CMD ["nginx", "-g", "daemon off;"]

root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4bb83700541b brain-tasks-app "/docker-entrypoint..." 8 minutes ago Up 8 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 push .c
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker tag brain-tasks-app:latest abhishek8056/project-2025:latest
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker images
The push refers to repository [docker.io/abhishek8056/project-2025]
e41c2478490: Pushed
ee4ad2fbc975: Pushed
df413ebdc0b: Pushed
23034b00a25d: Pushed (image pulled from library/node)
194f24e147d: Pushed
ff8a36d5902a: Pushed
3eaba6c610a3: Pushed
8f6a683e95d: Pushed
bdabb0d44271: Pushed
d9a55da5954: Pushed
latest: digest: sha256:f0ab0c148188074afe6e618a67068f2aab88407940f254ffe617ec0e80e00c16 size: 856
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>



```
--> extracting sha256:ff8a36d5502a57c3fc8eeff48e578ab433a03b1dd528992ba0d966ddf853109a 0.0s
--> extracting sha256:bdabb0d442710d667f4fd871b5fd215cc2a430a95b192b500bf945b8e66999b 0.0s
--> 0.25s
--> 0.2s
--> exporting to image
--> exporting layers
--> exporting manifest sha256:5f9ab0d061a2851d0639ceaff2514f165a2fe55115234dad170513ebe924bb1 0.1s
--> 0.0s
--> exporting config sha256:0ddfb08c939352d05a590d017c1e67513e66300f264ff1a501039ec0d28fe77a3 0.0s
--> 0.0s
--> exporting attestation manifest sha256:b25c344df2110ddceaf0677faa1a008acdbac2340f19fcbe0fb46fc7108a6ee 0.0s
--> 0.0s
--> exporting manifest list sha256:f0ab0c148188074afe6e618a67068f2aab88407940f254ffe617ec0e80e00c16 0.0s
--> 0.0s
--> 0.0s
--> naming to docker.io/abhishek8056/project-2025:latest
--> 0.0s
--> unpadding and writing Dockerfile to /home/ubuntu/project1/Abhi-Brain-Tasks-App
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# docker run -d -p 3000:80 brain-tasks-app
4bb83700541b75217e2a4f2141a0781a3a753891eb6a6b2ae3d3c8bf86b57
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App#
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App# cat Dockerfile
# Step 1: Use official Nginx image
FROM nginx:alpine

# Step 2: Copy build files from dist folder to nginx html directory
COPY dist /usr/share/nginx/html

# Step 3: Expose port 80 (default for nginx)
EXPOSE 80

# Step 4: Start Nginx server
CMD ["nginx", "-g", "daemon off;"]

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>

```

root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# ls
dist package-lock.json
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# vi Dockerfile
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker build -t brain-tasks-app .
[+] Building 5.5s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transfer from Dockerfile: 315B
=> [internal] load history from Docker.io/library/nginx:alpine
=> [auth] library/nginx pull token for registry-1.docker.io
=> [internal] load dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> = transferring context: 318.1KB
=> [1/2] FROM nginx:alpine@sha256:b3c656d5d7ad75119ef21bf7d2e04da9cb430e32f646adcf92441b72f82b14
=> => resolve to image io/library/nginx@alpine@sha256:b3c656d5d7ad75119ef21bf7d2e04da9cb430e32f646adcf92441b72f82b14
=> => sha256:ff8a364592a57c3f18e7fb48e879ab43a30d1dd28992ba0a96d105330301 1.09kB / 1.40kB
=> => sha256:bd8bb0d427108033300331662081519e054596063999 1.59MB / 1.59MB
=> => sha256:3d413d9e9dc834bccf617784554064d259c2c2d38d2c1ab70ea5e404b1ba5624 40.3B / 40.3B
=> => sha256:3eabaaed10a374d9ed69c26d76a258e20ddfa99ce1f51c198aa629dcf3fae4 955B / 955B
=> => sha256:194f1a24e147df0010e146240384bd25d84180a523dc717e4645b26999148e3 628B / 628B
=> => sha256:0f6a6833e95d43c524f119c5e7c1316c1f3be7ae5b3d4be54b0c5b910e80a 1.84MB / 1.84MB
=> => extracting sha256:0f6a6833e95d43c524f119c5e7c1316c1f3be7ae5b3d4be54b0c5b910e80a
=> => extracting sha256:194f1a24e147df0010e146240384bd25d84180c523dc717e4645b26999148e3
=> => extracting sha256:3eabaaed10a374d9ed69c26d76a258e20ddfa99ce1f51c198aa629dcf3fae4
=> => sha256:3d413d9e9dc834bccf631784554064d259c2c2d38d2c1ab70ea5e404b1ba5624
=> => sha256:ff8a364592a57c3f18e7fb48e879ab43a30d1dd28992ba0a96d105330301 1.09kB / 1.09kB
=> => sha256:1f8a364592a57c3f18e7fb48e879ab43a30d1dd28992ba0a96d105330301 1.09kB / 1.09kB
=> => sha256:bdabb0d42710d667141fd07b51d215cc2a430a95b192b5988f945b06660999b
=> [2/2] COPY dist /usr/share/nginx/html
=> exporting to image
=> exporting layers
=> => exporting manifest sha256:5f9a8d0e6a1a2851d0639ceaff2514f105a2fe55115234dad170513e0b924bb1
=> => exporting config sha256:0d1bd08c939352d05e599d6a17c6e7513e60030f2641fa501839ec0d20fe77a3
=> => exporting attestation manifest sha256:b25c344df2118ddcedf8677fa1a008acdbac234019fcbe0fb46fc7108a6ee
=> => exporting manifest sha256:0d1bd08c939352d05e599d6a17c6e7513e60030f2641fa501839ec0d20fe77a3
=> => naming to docker.io/library/brain-tasks-app:latest
=> => marking to docker.io/library/brain-tasks-app:latest
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker run -d -p 3000:80 brain-tasks-app
4b883700541b75217ee2a4f2141a0a7031a3a753891eb6a6b2a0ed3c2d8bf86b57
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>

push image also for docker hub for future use:

```

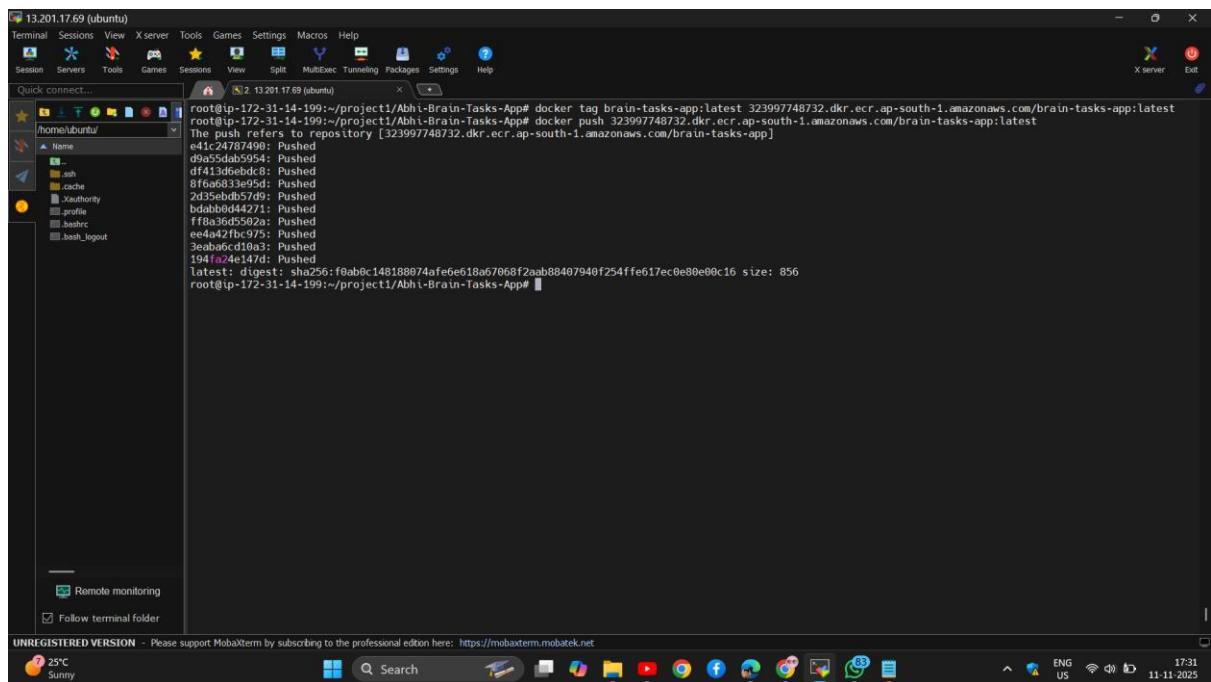
root@ip-172-31-14-199:# docker login
Authenticating with existing credentials... [Username: abbishek8056]
Info - To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
root@ip-172-31-14-199:#

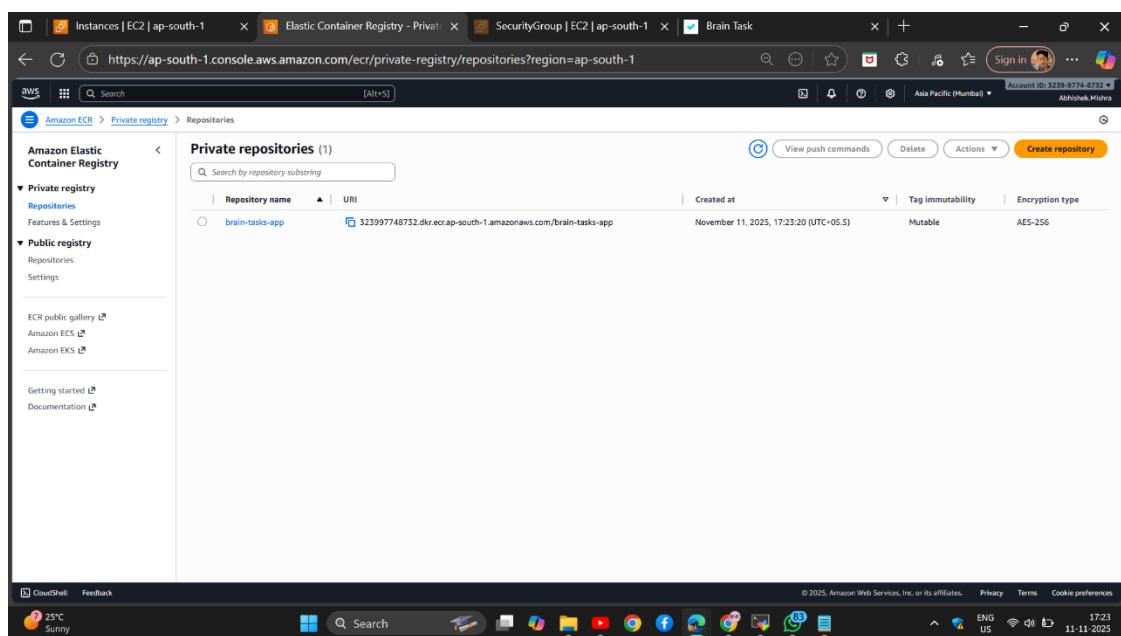
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

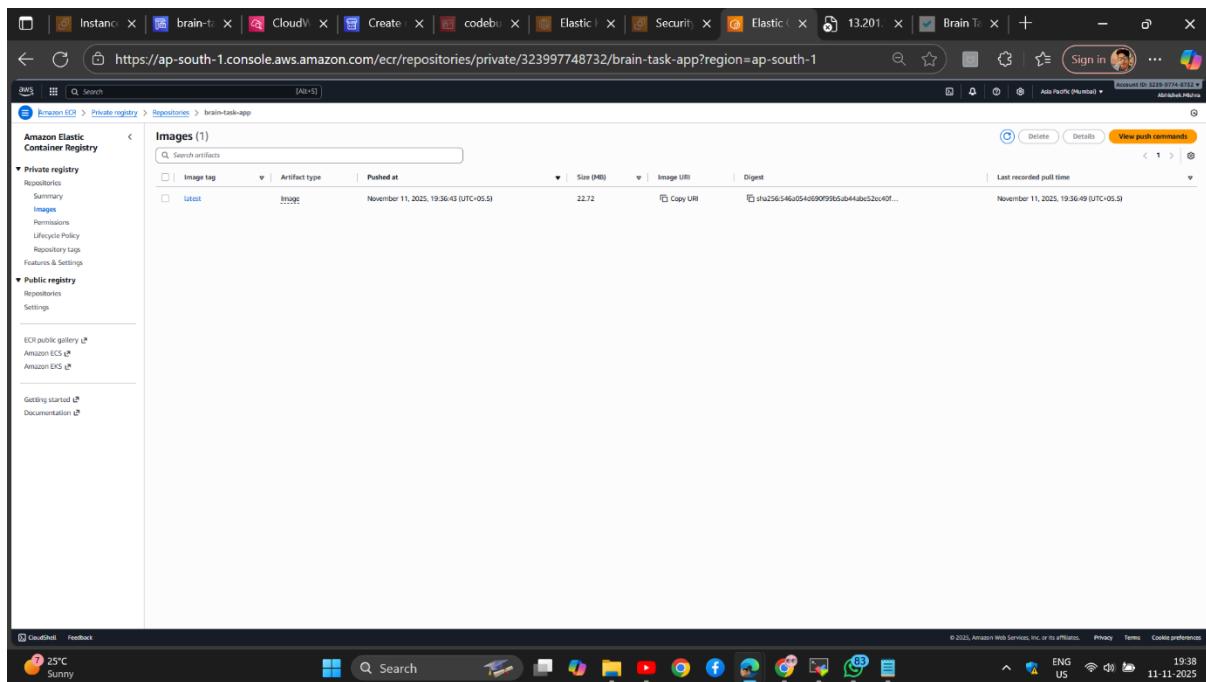
Pushing image to Aws ECR



```
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker tag brain-tasks-app:latest 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker push 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
e41c24787490: Pushed
d9a55dab5954: Pushed
df413d6ebdc3: Pushed
8f6a683e95d1: Pushed
2d35ebdb57d9: Pushed
bdabb0d44271: Pushed
ff8a36d5502a: Pushed
ee4a421fc975: Pushed
3eab06cd10a3: Pushed
194f24e147e0: Pushed
latest: digest: sha256:f0ab8c148188874afe6e618a67968f2aab88407940f254ffe617ec0e80e00c16 size: 856
root@ip-172-31-14-199:~/project1/Abhi-Brain-Tasks-App#
```



The screenshot shows the AWS ECR private registry interface. On the left, there's a sidebar with navigation options like 'Private registry' (selected), 'Public registry', 'ECR public gallery', 'Amazon ECS', and 'Amazon EKS'. The main content area displays a table titled 'Private repositories (1)'. The table has columns for 'Repository name', 'URI', 'Created at', 'Tag immutability', and 'Encryption type'. One row is listed: 'brain-tasks-app' with URI '323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app', created on 'November 11, 2025, 17:23:20 (UTC+05:5)', 'Mutable' tag immutability, and 'AES-256' encryption type. At the top right of the table, there are buttons for 'View push commands', 'Delete', 'Actions', and 'Create repository'. The browser address bar shows the URL 'https://ap-south-1.console.aws.amazon.com/ecr/private-registry/repositories?region=ap-south-1'. The bottom of the screen shows a Windows taskbar with various icons and system status.



☁ 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.

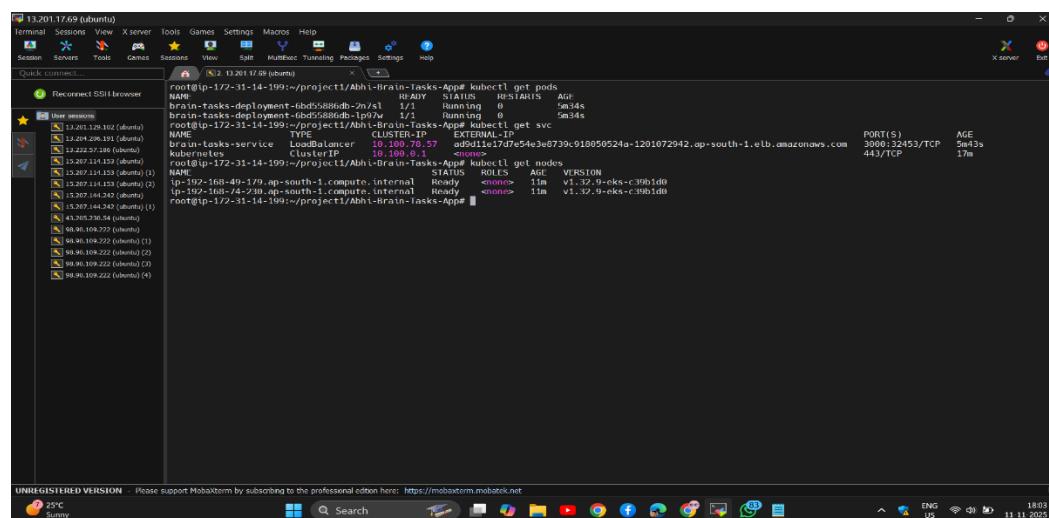
Verify Cluster Creation

```
# 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:



The screenshot shows a terminal window titled "13.201.17.69 (ubuntu)" running on Mobatik. The terminal displays the following command outputs:

```
root@ip-172-31-14-199:~/project/Abhi-Brain-Tasks-App# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
brain-tasks-deployment-6bd55886db-2y7s   1/1     Running   0          5m34s
root@ip-172-31-14-199:~/project/Abhi-Brain-Tasks-App# kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
brain-tasks-service   ClusterIP   10.109.0.1       <none>        3000:32453/TCP   5m43s
kubernetes     ClusterIP   10.109.0.1       <none>        443/TCP        17m
root@ip-172-31-14-199:~/project/Abhi-Brain-Tasks-App# kubectl get nodes
NAME                  STATUS   ROLES   AGE   VERSION
ip-192-168-40-179   ap-south-1   compute, internal   Ready   11m   v1.32.0+eks-c39b1d0
ip-192-168-74-210   ap-south-1   compute, internal   Ready   11m   v1.32.0+eks-c39b1d0
root@ip-172-31-14-199:~/project/Abhi-Brain-Tasks-App#
```

At the bottom of the terminal, there is a footer note: "UNREGISTERED VERSION - Please support Mobatik by subscribing to the professional edition here: <https://mobatik.mobatik.net>".

13.201.17.69 (ubuntu)

Terminal Sessions View Xserver Tools Games Settings Macros Help

Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

Reconnect SSH-browser

User sessions

- 13.201.129.102 (ubuntu)
- 13.204.206.191 (ubuntu)
- 13.232.57.186 (ubuntu)
- 15.207.114.151 (ubuntu)
- 15.207.114.151 (ubuntu) (1)
- 15.207.114.151 (ubuntu) (2)
- 15.207.144.242 (ubuntu)
- 43.205.230.54 (ubuntu)
- 98.90.109.222 (ubuntu) (1)
- 98.90.109.222 (ubuntu) (2)
- 98.90.109.222 (ubuntu) (3)
- 98.90.109.222 (ubuntu) (4)

Dockerfile dist eksctl.tar.gz package-lock.json

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

```
deployment "brain-tasks" 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	ad9d11e17de54e3e8739c918050524a-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# docker ps -a
```

CONTAINER ID	IMAGE	CREATED	STATUS	PORTS	NAMES
4b883700541b	brain-tasks-app "/docker-entrypoint..."	57 minutes ago	Up 57 minutes	0.0.0.0:3000->80/tcp	optimistic_hermann

```
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker stop 4b883700541b
```

```
root@ip-172-31-14-199:/project1/Abhi-Brain-Tasks-App# docker stop 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>

25°C

Search

ENG 18:02

AWS

Instances EC2 Code Deploy Elastic EC2 Code S3 Instances Instances Repos Brain Brain

<https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/sshd/home?region=ap-south-1&connType=standard&instanc...>

Account ID: 3239-0774-8732 Abhishek.Mishra

aws

Search [Alt+S]

ubuntu@ip-172-31-14-199:~/brain-task-app\$ sudo bash cleanup.sh

Cleaning up old deployment (if exists).

Deployment "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\$ sudo kubectl get pods

No results found in current namespace.

ubuntu@ip-172-31-14-199:~/brain-task-app\$ sudo bash deploy.sh

Starting deployment to EKS cluster...

Updated context arn:aws:eks:ap-south-1:323997749732: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\$ sudo kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
brain-tasks-deployment-6dd5886db-7ca5j	1/1	Running	0	10s
brain-tasks-deployment-6dd5886db-ghbf7	1/1	Running	0	10s

ubuntu@ip-172-31-14-199:~/brain-task-app\$ sudo kubectl get svc

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
brain-tasks-service	LoadBalancer	10.100.234.234	a294eca25194020bea2dd8276bf473-1453405367.ap-south-1.elb.amazonaws.com	3000:32744/TCP	30m
kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	38h

ubuntu@ip-172-31-14-199:~/brain-task-app\$

CloudShell Feedback

12°C Clear

Search

ENG US 08:09 13-11-2025

Cluster output:

The screenshot shows the AWS EKS Cluster Overview page for the 'brain-task-cluster'. The left sidebar includes sections for Amazon Elastic Kubernetes Service, Settings, Amazon EKS Anywhere, and Related services. The main content area displays cluster info (Status: Active, Kubernetes version: 1.32, Support period: Standard support until March 23, 2026, Provider: EKS), cluster health (0 issues), and deployment insights (0 issues). The 'Resources' tab is selected, showing a table of workloads. The 'Deployments' section lists three entries: 'brain-tasks-deployment' (default namespace, 2 pods, 2 ready, 2 desired), 'coredns' (kube-system namespace, 2 pods, 2 ready, 2 desired), and 'metrics-server' (kube-system namespace, 2 pods, 2 ready, 2 desired).

Workloads pods:

The screenshot shows the AWS EKS Cluster Overview page for the 'brain-task-cluster'. The left sidebar includes sections for Amazon Elastic Kubernetes Service, Settings, Amazon EKS Anywhere, and Related services. The main content area displays cluster info (Status: Active, Kubernetes version: 1.32, Support period: Standard support until March 23, 2026, Provider: EKS), cluster health (0 issues), and deployment insights (0 issues). The 'Resources' tab is selected, showing a table of workloads. The 'Pods' section lists ten entries: 'aws-node-d44tc', 'aws-node-m2o2f', 'brain-tasks-deployment-6bd55886db-7ct5j', 'brain-tasks-deployment-6bd55886db-gbhnt', 'coredns-6799d65cb-bhc94', 'coredns-6799d65cb-nhqj8', 'kube-proxy-prn684', 'kube-proxy-w85dr', 'metrics-server-5c6bdff7f85-88j4j', and 'metrics-server-5c6bdff7f85-196mw'. All pods are in a 'Running' status.

Deployment:

The screenshot shows the AWS EKS Cluster Details page for the 'brain-task-cluster'. The 'Resources' tab is selected. Under 'Resource types', 'Deployments' is chosen. The table lists three deployments:

Name	Namespace	Type	Created	Status
brain-tasks-deployment	default	deployments	6 hours ago	2 Ready 0 Failed 2 Desired
coredns	kube-system	deployments	November 11, 2025, 17:48 (UTC+05:30)	2 Ready 0 Failed 2 Desired
metrics-server	kube-system	deployments	November 11, 2025, 17:53 (UTC+05:30)	2 Ready 0 Failed 2 Desired

Replicsets

The screenshot shows the AWS EKS Cluster Details page for the 'brain-task-cluster'. The 'Resources' tab is selected. Under 'Resource types', 'ReplicaSets' is chosen. The table lists three replica sets:

Name	Namespace	Type	Created	Status
brain-tasks-deployment-6bd55886db	default	replicasets	6 hours ago	2 Ready 0 Failed 2 Desired
coredns-6799d65cb	kube-system	replicasets	November 11, 2025, 17:48 (UTC+05:30)	2 Ready 0 Failed 2 Desired
metrics-server-5c6bdf7f85	kube-system	replicasets	November 11, 2025, 17:53 (UTC+05:30)	2 Ready 0 Failed 2 Desired

IAM access identity that access the codepipeline in default namespace

The screenshot shows the 'Access configuration' section of the AWS EKS Cluster Management interface. It lists four IAM access entries:

IAM principal ARN	Type	Username	Group names	Access policies
arn:aws:iam::323997748732:role/aws-service-role/eks.amazonaws.com/AWSServiceRoleForAmazonEKS	Standard	arn:awssts::323997748732:assumed-role/AWSServiceRoleForAmazonEKS/{{SessionName}}	eks:managed	AmazonEKSClusterInsightsPolicy, AmazonEKSEventPolicy
arn:aws:iam::323997748732:role/eksctl-brain-task-cluster-nodegroup-NodeInstanceRole-PYGJpkRfF9i	EC2 Linux	system:node[{EC2PrivateDNSName}]	system:nodes	
arn:aws:iam::323997748732:role/service-role/eks-role-test	Standard	arn:awssts::323997748732:assumed-role/eks-role-test/{{SessionName}}		AmazonEKSAdminPolicy
arn:aws:iam::323997748732:user/Abhishek.Mishra	Standard	arn:aws:iam::323997748732:user/Abhishek.Mishra		AmazonEKSClusterAdminPolicy

A note at the bottom states: "Be sure to install the Amazon EKS Pod Identity Agent add-on before creating, editing, or deleting Pod identity associations. This add-on is required for the EKS Pod Identity feature to function properly." A "Create add-on" button is also present.

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

The screenshot shows the 'ip-192-168-37-235.ap-south-1.compute.internal' node details page. It displays the following information:

- Details:** Status: Ready, Last transition time: a day ago, OS Architecture: linux/amd64, OS image: Amazon Linux 2023.9.20251105, Instance type: t3.medium.
- Capacity allocation:** Two circular charts show resource usage:
 - Cores:** Available: 1.80 (79%), Used: 2 Cores (4%).
 - Memory:** Available: 2.34 Gb (79%), Used: 3.74 Gb (14%).

Screenshot of the Amazon Elastic Kubernetes Service (EKS) Cluster Node Details page for node ip-192-168-37-235.ap-south-1.compute.internal.

Resource Utilization:

- Cores:** 2 Cores (Available: 1.68 m, 79%)
- Memory:** 3.74 GiB (Available: 2.04 GiB, 99%)
- Pods:** 17 Pods (Used: 9 pods, 53%)

Name	Status	Created	IP
brain-tasks-deployment-49cf55806db-7nsj	Running	9 hours ago	192.168.51.110
zero-node-1-202	Running	a day ago	192.168.37.235
coredns-6799d5c5-rhgd8	Running	a day ago	192.168.50.219
kube-proxy-wt5dr	Running	a day ago	192.168.37.235
metrics-server-5c60df7f05-88q4	Running	a day ago	192.168.32.244

Conditions:

Name	Status	Message
MemoryPressure	False	kublet has sufficient memory available

CloudShell Feedback: 22°C Sunny

Screenshot of the Amazon Elastic Kubernetes Service (EKS) Cluster Node Details page for node ip-192-168-82-216.ap-south-1.compute.internal.

Resource Utilization:

- Cores:** 2 Cores (Available: 1.68 m, 79%)
- Memory:** 3.74 GiB (Available: 2.04 GiB, 99%)
- Pods:** 17 Pods (Used: 9 pods, 53%)

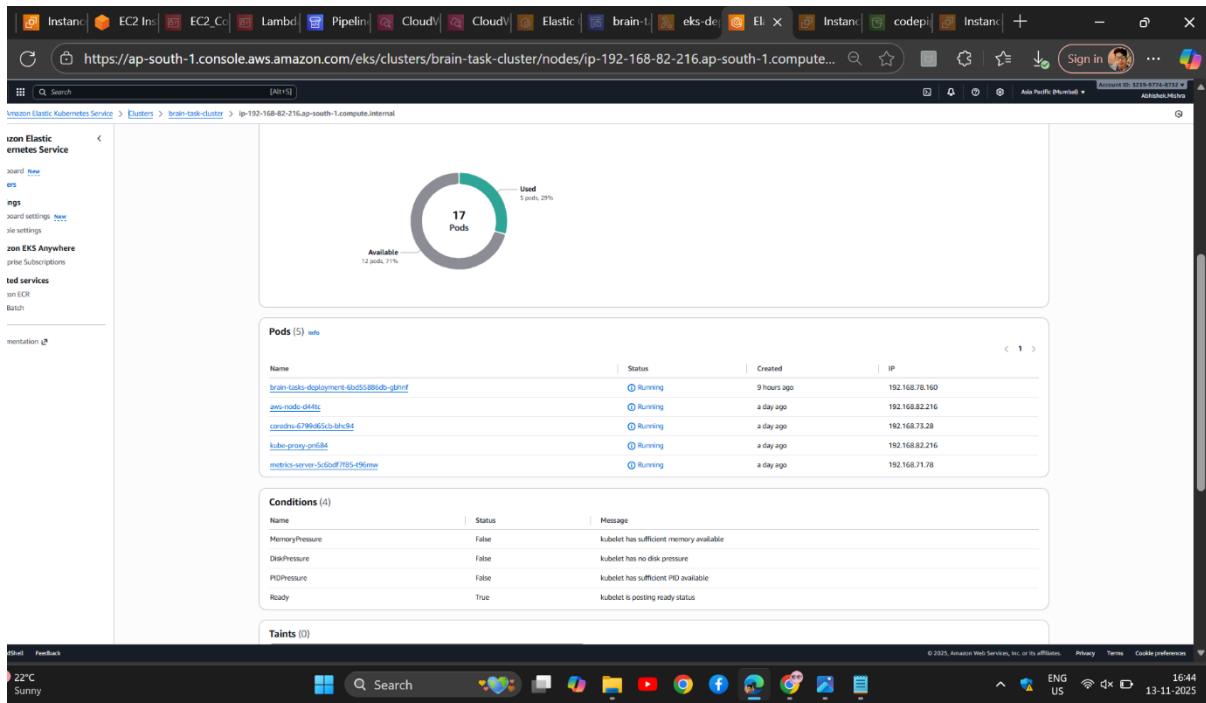
Node Details:

Details	Value
Status	Ready
Last transition time	a day ago
OS Architecture	linux (arm64)
OS Image	Amazon Linux 2023.9.20251105
Instance type	t3.medium

Capacity allocation:

- Cores:** 2 Cores (System reserved: 0.16 m, 8%), Workload reserved: 0.56 m, 14%, Available: 1.68 m, 79%)
- Memory:** 3.74 GiB (System reserved: 0.32 GiB, 1%), Workload reserved: 0.32 GiB, 7%, Available: 2.04 GiB, 99%)
- Pods:** 17 Pods (Used: 9 pods, 53%)

CloudShell Feedback: 22°C Sunny



Deployment using the k8s(files)

Abhi-Brain-Tasks-App / appspec.yml

```

root update the zip file ✓ ecd24ee - yesterday History

Code Blame 16 lines (16 loc) · 348 Bytes

1 version: 0.0
2 Resources:
3   - myEKSApp:
4     Type: AWS::EKS::Application
5     Properties:
6       ClusterName: brain-task-cluster
7       Namespace: default
8     hooks:
9       BeforeInstall:
10      - location: scripts/cleanup.sh
11        timeout: 300
12        runas: root
13     AfterInstall:
14      - location: scripts/deploy.sh
15        timeout: 600
16        runas: root

```

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

```
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

```
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

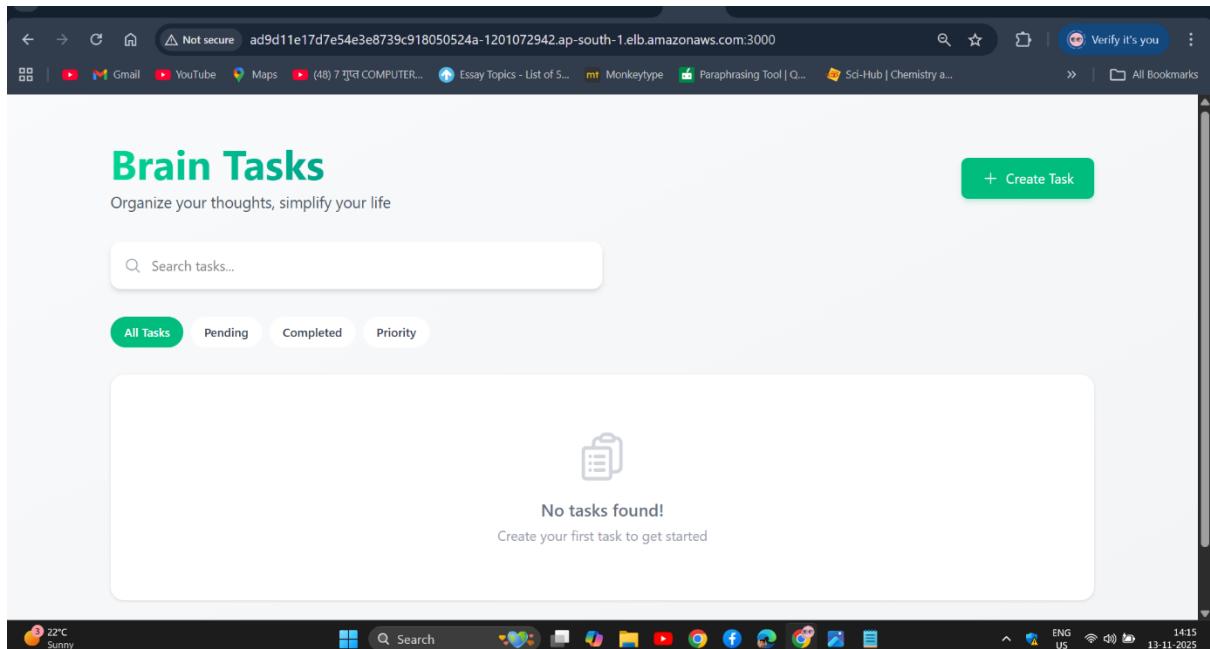
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: brain-tasks
  template:
    metadata:
      labels:
        app: brain-tasks
    spec:
      containers:
        - name: brain-tasks-container
          image: 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:
 - o Authenticates with ECR
 - o Builds Docker image
 - o Pushes image to ECR
 - o Creates artifacts
4. CodeDeploy triggers deployment:
 - o Executes BeforeInstall hook (cleanup.sh)
 - o Invokes Lambda function
 - o Lambda updates EKS cluster
 - o 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.

The screenshot shows the AWS S3 console interface. The left sidebar lists various AWS services like EC2, Lambda, CloudWatch Metrics, and others. Under the 'Amazon S3' section, it shows 'General purpose buckets' and 'Storage Lens'. The main content area displays the contents of the bucket 'codepipeline-ap-south-1-ebb9c75bc257-437c-9c01-b6a443c9fb46'. The 'Objects' tab is selected, showing two items:

Name	Type	Last modified	Size	Storage class
brain-task-pipeline/	Folder	-	-	-
eks-brain/	Folder	-	-	-

This screenshot shows the contents of the 'eks-brain/' folder within the root bucket. The 'Objects' tab is selected, displaying two items:

Name	Type	Last modified	Size	Storage class
BuildArtif/	Folder	-	-	-
SourceArtif/	Folder	-	-	-

BuildArtif/

Objects (1)

Name	Type	Last modified	Size	Storage class
copyIKDh	-	November 13, 2025, 13:24:02 (UTC+05:30)	279.0 B	Standard

SourceArtif/

Objects (1)

Name	Type	Last modified	Size	Storage class
4oxIJCU	-	November 13, 2025, 13:22:59 (UTC+05:30)	207.6 KB	Standard

Roles for different services

The screenshot shows the AWS IAM Role details page for the role `EC2_CodeDeploy_EKS_Role`. The role allows EC2 instances to call AWS services on behalf of the user. It has a creation date of November 12, 2025, and a maximum session duration of 1 hour. The ARN is `arn:aws:iam::323997748732:role/EC2_CodeDeploy_EKS_Role`. The instance profile ARN is `arn:aws:iam::323997748732:instance-profile/EC2_CodeDeploy_EKS_Role`. The Permissions tab shows seven attached policies:

Policy name	Type	Attached entities
<code>AmazonEC2RoleforAWSCodeDeploy</code>	AWS managed	3
<code>AmazonEKSWorkerPolicy</code>	AWS managed	3
<code>AmazonEKSSignerPolicy</code>	AWS managed	3
<code>AmazonESFleetAccess</code>	AWS managed	4
<code>AmazonVSSReadOnlyAccess</code>	AWS managed	2
<code>AWSCodeDeployRole</code>	AWS managed	2
<code>LambdaLayerPublishAccess</code>	Customer inline	0

The Permissions boundary is not set. There are no CloudTrail events to generate a policy based on.

The screenshot shows the AWS IAM Role details page for the role `LambdaEKSDeployRole`. This role is for Lambda functions to deploy to EKS from CodePipeline. It was created on November 13, 2025, with a maximum session duration of 1 hour. The ARN is `arn:aws:iam::323997748732:role/LambdaEKSDeployRole`. The Permissions tab shows eight attached policies:

Policy name	Type	Attached entities
<code>AmazonECSContainerRegistryReadOnly</code>	AWS managed	3
<code>AmazonEKSWorkerPolicy</code>	AWS managed	4
<code>AmazonEKSSigningPolicy</code>	AWS managed	3
<code>AmazonESFleetAccess</code>	AWS managed	5
<code>AmazonPCCFullAccess</code>	AWS managed	1
<code>AWSCodePipeline_FullAccess</code>	AWS managed	1
<code>AWSCognitoIdentityFullAccess</code>	AWS managed	1
<code>AWSLambdaTaskExecutionRole</code>	AWS managed	1

The Permissions boundary is not set. There are no CloudTrail events to generate a policy based on.

Two instances for the cluster one for my particle:

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manage, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, CloudShell, and Feedback. The main area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, and Elastic IP. Three instances are selected: 'eks-deploy' (t2.2xlarge), 'brain-task-clus...' (t3.medium), and 'brain-task-clus...' (t3.medium). Below the table are several monitoring charts for CPU utilization, network traffic, and other metrics over a 1-hour period.

Codebuild setup:

The screenshot shows the AWS CodeBuild Build projects page. The left sidebar includes sections for Developer Tools (CodeCommit, Artifacts, CodeArtifact, Build + CodeBuild, Getting started, Build projects, Build history, Report groups, Report history, Compute fleets, Account metrics, Related integrations, Jenkins, GitHub Actions, GitLab runners, Deploy + CodeDeploy, Pipeline + CodePipeline, Settings, Go to resource, and Feedback). The main content area shows a table of build projects with columns: Name, Source provider, Repository, Latest build status, Description, and Last Modified. One project, 'brain-task-codebuild', is listed with a GitHub source provider, repository 'Abhi-mishra998/Abhi-Brain-Tasks-App', latest build status 'Succeeded', and last modified '1 day ago'. There are also buttons for Actions, Create trigger, View details, Debug build, Start build, and Create project.

The screenshot shows the AWS CodeBuild console interface. On the left, the navigation sidebar includes sections for Source, Artifacts, Build, Deploy, Pipeline, and Settings. The main area displays the 'Configuration' tab for the 'brain-task-codebuild' project. It shows the source provider as GitHub, the primary repository as 'Abhi-mishra98/Abhi-Brain-Tasks-App', and the artifacts upload location as '-' (disabled). The service role is 'arn:aws:iam::323997748732:role/service-role/codebuild-brain-task-codebuild-service-role'. Below this, the 'Build history' tab is selected, showing two build runs. The first build run (ID: 24) was successful ('Succeeded') and completed 41 minutes ago. The second build run (ID: 23) failed ('Failed') and completed 41 minutes ago. The build logs for both runs are visible.

Testing multiple time to see it working fine or not

This screenshot shows a very long list of build history entries for the 'brain-task-codebuild' project. The list spans multiple pages, indicating frequent testing. Most builds are marked as 'Succeeded', while one build (ID: 10) is listed as 'Failed'. The builds were performed over a period of approximately 3 hours ago. The details for each build include the build ID, status, duration, and the GitHub commit hash.

Build ID	Status	Duration	GitHub Commit Hash
36351f15f7958d82cab7ab0978ebbfef983bc90	Succeeded	29 seconds	gitHub-Hookshot/3d77481
3d52c7af5e02199a7bc820	Succeeded	25 seconds	codepipeline/brain-task-pipeline
ec024ee1633d29ab4be082	Succeeded	27 seconds	gitHub-Hookshot/3d77481
5d32c7af5e02199a7bc820	Succeeded	26 seconds	GitHub-Hookshot/05f2675
961195f691143891a4564f375dc578f5285743d7	Succeeded	28 seconds	GitHub-Hookshot/05f2675
4de13740f178f0692945f9085094097c3f6a79	Succeeded	29 seconds	GitHub-Hookshot/05f2675
2797f52d386d569c50ba57d1f4e028cd3483f268	Succeeded	29 seconds	GitHub-Hookshot/05f2675
b7d121f0938348053e05a487218ecc3407814d54	Failed	9 seconds	GitHub-Hookshot/05f2675
abfdf93e760c4090bfc9b766e450e525cd34013	Succeeded	28 seconds	GitHub-Hookshot/05f2675

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

CloudShell Feedback

22°C Sunny

Search

View push commands

Images (17)

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	November 13, 2025, 13:24:00 (UTC+05:5)	22.72	[Copy URI]	sha256:1c1bc8ebc24de4bb0e5a48511db7e...	November 13, 2025, 13:24:07 (UTC+05:5)
-	Image	November 13, 2025, 12:06:14 (UTC+05:5)	22.72	[Copy URI]	sha256:6bd5291a921f0050803b545f0213d...	November 13, 2025, 12:06:20 (UTC+05:5)
-	Image	November 13, 2025, 12:06:09 (UTC+05:5)	22.72	[Copy URI]	sha256:1887c81ac47c680b859b855213337...	November 13, 2025, 12:06:16 (UTC+05:5)
-	Image	November 13, 2025, 11:42:45 (UTC+05:5)	22.72	[Copy URI]	sha256:38ed4a53ef73fdb4d43f037fa09e28...	November 13, 2025, 11:42:55 (UTC+05:5)
-	Image	November 13, 2025, 11:42:42 (UTC+05:5)	22.72	[Copy URI]	sha256:433303462b8486596fe8e238944ad...	November 13, 2025, 11:42:54 (UTC+05:5)
-	Image	November 13, 2025, 10:16:28 (UTC+05:5)	22.72	[Copy URI]	sha256:fbbbe2a49ce141a29bdb7e9ed695ef...	November 13, 2025, 10:16:55 (UTC+05:5)
-	Image	November 13, 2025, 10:16:14 (UTC+05:5)	22.72	[Copy URI]	sha256:493981c996fb9a493322b80405677c...	November 13, 2025, 10:16:21 (UTC+05:5)
-	Image	November 13, 2025, 09:43:33 (UTC+05:5)	22.72	[Copy URI]	sha256:9b2321613ec728943fe398e2de7f08...	November 13, 2025, 09:43:42 (UTC+05:5)
-	Image	November 13, 2025, 09:35:27 (UTC+05:5)	22.72	[Copy URI]	sha256:3f11249ab30a23af3af4df9b7238bf...	November 13, 2025, 09:36:31 (UTC+05:5)
-	Image	November 12, 2025, 18:27:53 (UTC+05:5)	22.72	[Copy URI]	sha256:9541dd27f9a081f1a80719cd125...	November 12, 2025, 18:27:59 (UTC+05:5)
-	Image	November 12, 2025, 07:22:49 (UTC+05:5)	22.72	[Copy URI]	sha256:091a46454b516658d9e1273345e4...	November 12, 2025, 07:23:07 (UTC+05:5)
-	Image	November 12, 2025, 07:20:13 (UTC+05:5)	22.72	[Copy URI]	sha256:30c6d409f03fb0b7aca1ff34b2c...	November 12, 2025, 07:22:06 (UTC+05:5)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG US 14:47 13-11-2025

Checking CloudWatch logs

CloudWatch

Favorites and recent

CloudWatch

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

Actions Start tailing Create metric filter

Clear 1m 30m 1h 12h Custom UTC timezone Display

Timestamp Message

2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.313288 Skip cache due to: no paths specified to be cached
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.313661 Registering with agent
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.353967 Phases found in YAML: 1
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.353924 BUILD: 7 commands
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.354250 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.354264 Phase context status code: Message:
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.406943 Entering phase INSTALL
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.566937 Phase complete: INSTALL State: SUCCEEDED
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.566940 Phase context status code: Message:
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.608932 Entering phase PRE_BUILD
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.627455 Phase complete: PRE_BUILD State: SUCCEEDED
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.627472 Phase context status code: Message:
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.670104 Entering phase BUILD
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.712971 Running command mkdir -p /tmp/cp-action-source
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.712971 Running command export CODEPIPELINE_INPUT_ACTION_SOURCE_PATH=/tmp/cp-action-source
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.728923 Running command export CODEPIPELINE_INPUT_SOURCE_PATH=/tmp/cp-action-source
2025-11-13T09:06:57.040Z	[Container] 2025/11/13 09:06:53.728923 Running command curl "https://d1dr2hrwien.cloudfront.net/deploy-eks-aws-1.0.1.e.tar.gz" -o /tmp/cp-action-source/action-archive.tar

Back to top

CloudShell Feedback

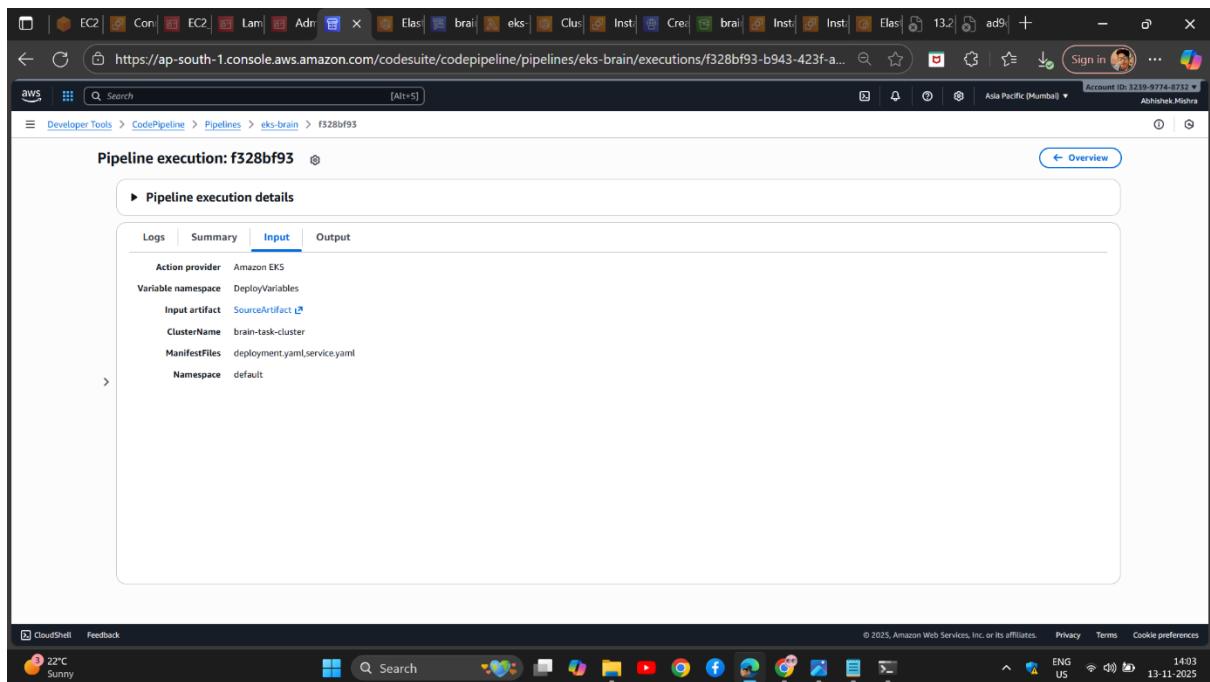
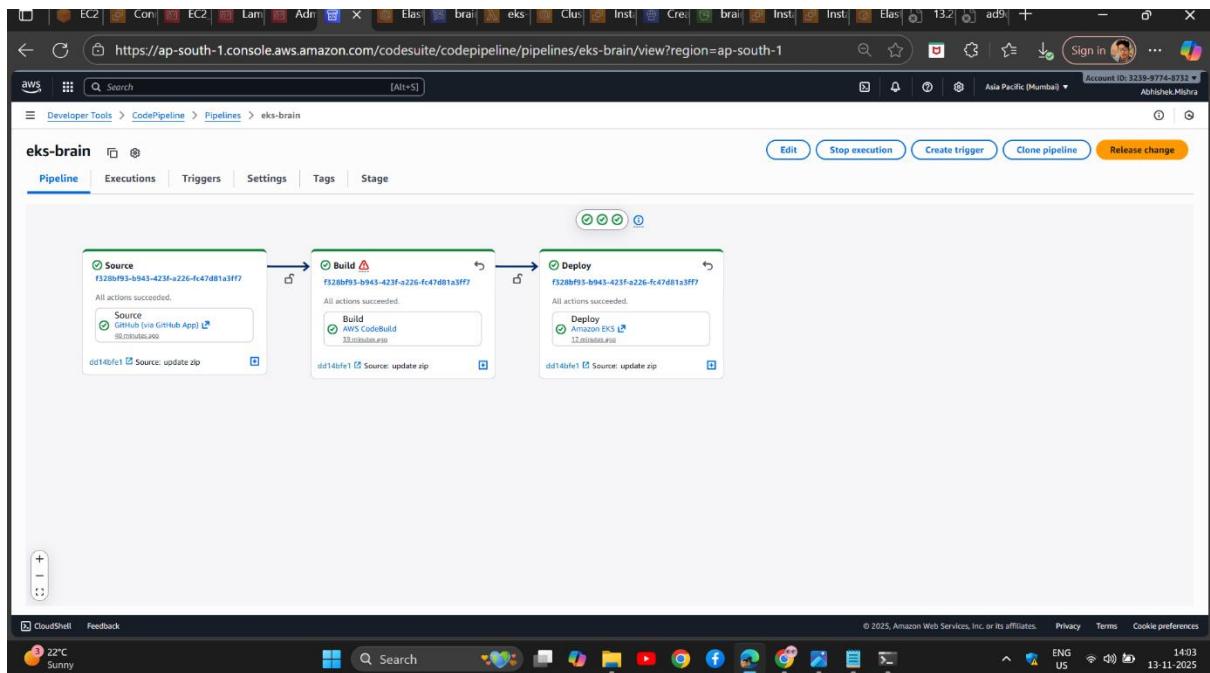
22°C Sunny

Search

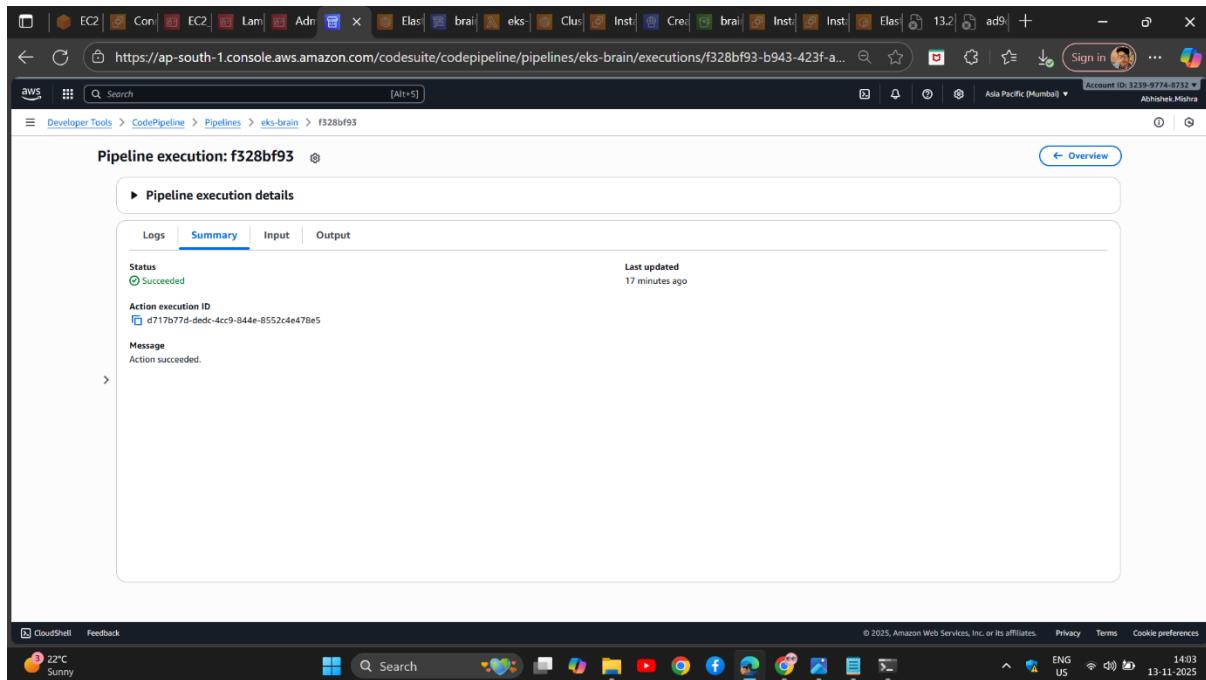
Privacy Terms Cookie preferences

ENG US 14:47 13-11-2025

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



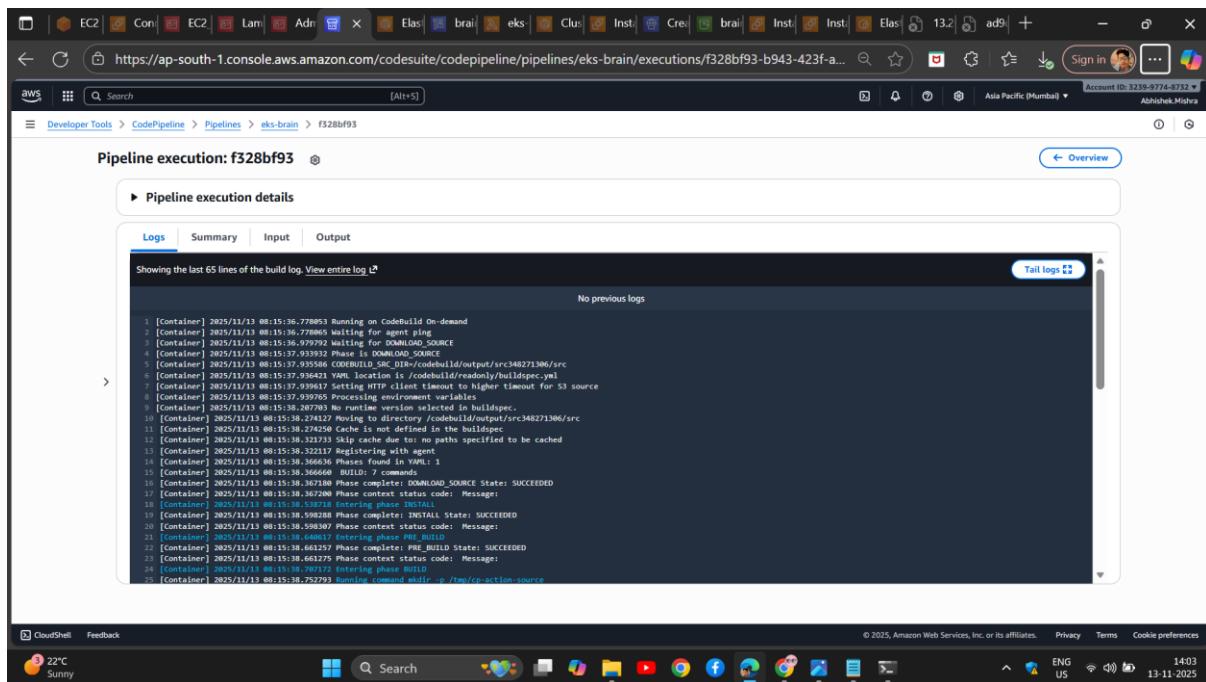
Screenshot of the AWS CloudWatch Pipeline execution details page for pipeline f328bf93. The status is Succeeded. The action execution ID is 4717b77d-dedc-4cc9-844e-8552c4e470e5. The message indicates the action succeeded.



Screenshot of the AWS CloudWatch Pipeline execution details page for pipeline f328bf93. The status is Succeeded. The logs tab shows the build log, which includes the following output:

```
Showing the last 65 lines of the build log. View entire log ↗
[Tail log ↗]
No previous logs

1 [Container] 2025/11/13 08:15:36.778053 Running on CodeBuild On-demand
2 [Container] 2025/11/13 08:15:36.778055 Waiting for agent ping
3 [Container] 2025/11/13 08:15:36.801056 Setting up environment SOURCE
4 [Container] 2025/11/13 08:15:37.039073 Phase: DOWNLOAD_SOURCE
5 [Container] 2025/11/13 08:15:37.935586 CODEBUILD_SRC_DIR=/codebuild/output/src348271306/src
6 [Container] 2025/11/13 08:15:37.936421 YAML location is /codebuild/readonly/buildspec.yml
7 [Container] 2025/11/13 08:15:37.939795 Setting HTTP client timeout to higher timeout for S3 source
8 [Container] 2025/11/13 08:15:38.007093 No runtime version selected in buildspec.
9 [Container] 2025/11/13 08:15:38.207793 No runtime version selected in buildspec.
10 [Container] 2025/11/13 08:15:38.274127 Moving to directory /codebuild/output/src348271306/src
11 [Container] 2025/11/13 08:15:38.274258 Cache is not defined in the buildspec
12 [Container] 2025/11/13 08:15:38.302117 No cache paths specified to be cached
13 [Container] 2025/11/13 08:15:38.332117 Registering with vcs
14 [Container] 2025/11/13 08:15:38.366636 Phases found in YAML: 1
15 [Container] 2025/11/13 08:15:38.366668 BUILD 7 commands
16 [Container] 2025/11/13 08:15:38.367108 Phase complete: DOWNLOAD_SOURCE state: SUCCEEDED
17 [Container] 2025/11/13 08:15:38.367110 Phase context status code: Message:
18 [Container] 2025/11/13 08:15:38.530718 Entering phase INSTALL
19 [Container] 2025/11/13 08:15:38.590288 Phase complete: INSTALL State: SUCCEEDED
20 [Container] 2025/11/13 08:15:38.598307 Phase context status code: Message:
21 [Container] 2025/11/13 08:15:38.622257 Phase complete: PRE_BUILD State: SUCCEEDED
22 [Container] 2025/11/13 08:15:38.662275 Phase context status code: Message:
23 [Container] 2025/11/13 08:15:38.707172 Entering phase BUILD
24 [Container] 2025/11/13 08:15:38.752793 Running command while in /tmp/cp action_source
```



Screenshot of AWS CloudShell showing the logs for Pipeline execution f328bf93.

The logs show the following steps:

- Logging in to Amazon EKS cluster.
- Installing kubectl.
- Applying Kubernetes manifests.
- Resource to watch: []
- Action completed.
- Running command: [-f /tmp/cp-action-source/action-output-variables.sh] && chmod 755 /tmp/cp-action-source/action-output-variables.sh || true
- Phase complete: BUILD State: SUCCEEDED
- Phase context status code: Message: [Container] 2025/11/13 08:15:55.764657 Phase complete: POST_BUILD State: SUCCEEDED
- Phase context status code: Message: [Container] 2025/11/13 08:15:55.764655 Phase context status code: Message:

CloudShell status: 22°C Sunny

Screenshot of AWS CloudShell showing the logs for CodeBuild build brain-task-323997748732.

The logs show the following steps:

- Entering phase PRE_BUILD.
- Logging in to Amazon EKS.
- Starting build on Tue Nov 14 08:46:39 UTC 2025.
- Build started using docker driver.
- Transferring dockerfile: 319 bytes.
- Transferring context: 28 bytes.
- Writing image sha256:4ff916a630ab2af6d95a5ff23198ac52ad805c3751fe6b767dc52f2c2f.
- Pushing docker image to ECR.
- Pushing command docker push 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
- Phase complete: BUILD State: SUCCEEDED
- Phase context status code: Message: [Container] 2025/11/14 08:46:48.05924 Phase context status code: Message:
- Phase context status code: Message: [Container] 2025/11/14 08:46:48.05918 Pushing command echo Pushing Docker image to ECR...
- Pushing Docker image to ECR...
- Pushing command docker push 323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
- The image was successfully pushed to the repository [323997748732.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app]

CloudShell status: 25°C Sunny

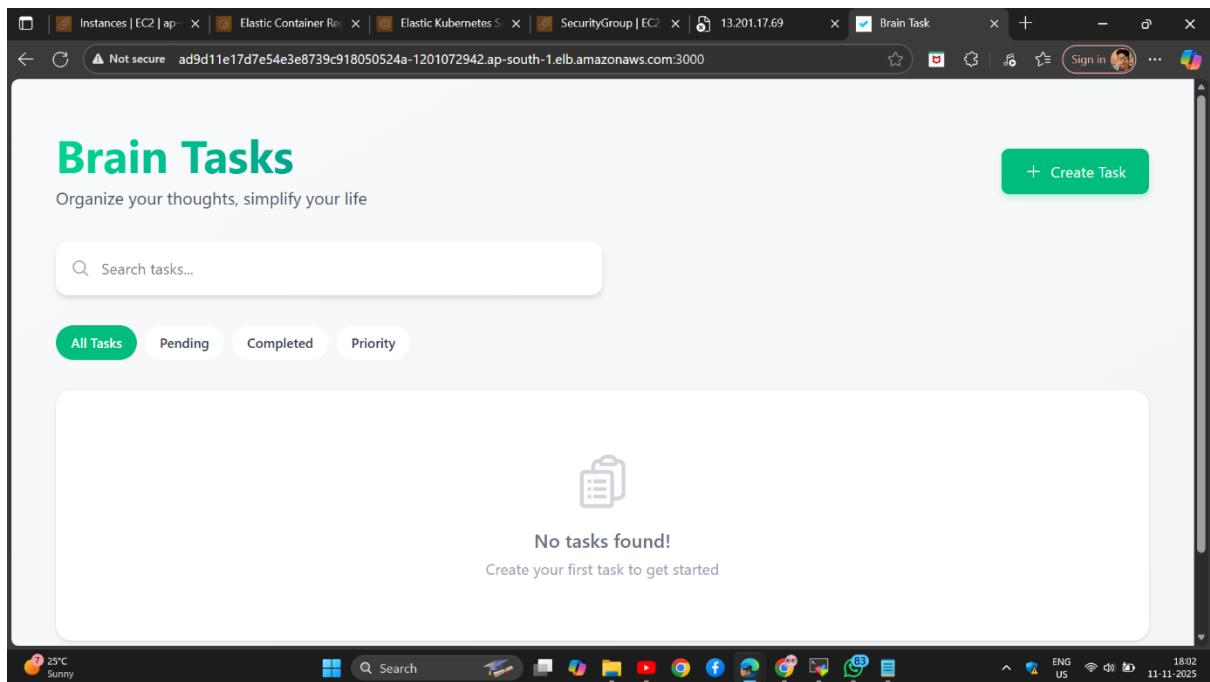
The screenshot shows the AWS CodeBuild console in a Microsoft Edge browser. The URL is <https://ap-south-1.console.aws.amazon.com/codesuite/codebuild/323997748732/projects/brain-task-codebuild/build/brain-task-1/11>. The page displays the build logs for the latest build, showing the deployment of a Docker image to AWS Lambda. The logs include commands like 'Running command docker tag brain-task-app:latest', 'Pushing image to AWS Lambda', and 'Lambda function deployed'. The build status is 'SUCCEEDED'.

The screenshot shows the AWS ECR console with the following details:

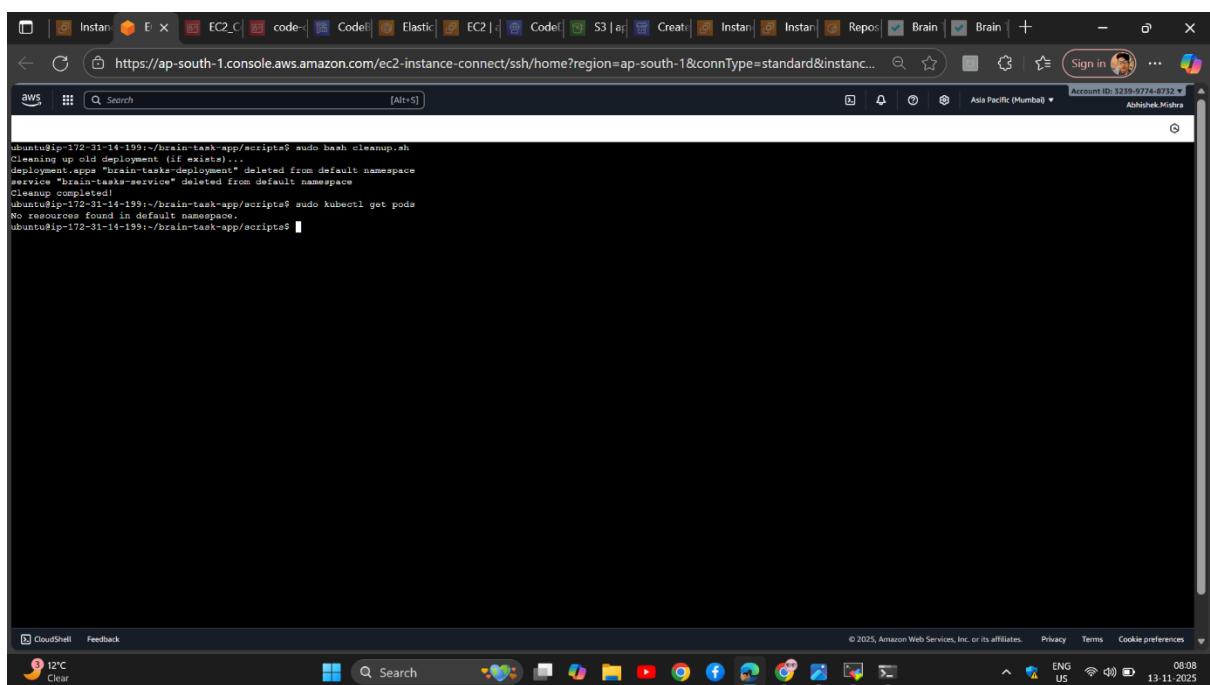
- Instances | EC2 | ap-south-1**, **Elastic Container Registry - Image**, **SecurityGroup | EC2 | ap-south-1**, and **Brain Task** tabs are open.
- The URL is <https://ap-south-1.console.aws.amazon.com/ecr/repositories/private/323997748732/brain-tasks-app?region=ap-south-1>.
- The account ID is 5239-9774-8732, and the region is Asia Pacific (Mumbai).
- The user is signed in as Abhishek Mishra.
- The main navigation bar includes **aws**, **Search**, and **[Alt+S]**.
- The breadcrumb path is **Amazon ECR > Private registry > Repositories > brain-tasks-app**.
- The left sidebar shows the **Private registry** section with **Repositories**, **Summary**, **Images** (selected), **Permissions**, **Lifecycle Policy**, **Repository tags**, and **Features & Settings**. The **Public registry** section includes **ECR public gallery**, **Amazon ECS**, and **Amazon EKS**. The **Getting started** and **Documentation** sections are also listed.
- The main content area displays the **Images (3)** table with the following data:

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:f0ab0c148188074afe6e61ba67068f...	-
-	image	November 11, 2025, 17:31:08 (UTC+05:5)	22.72	Copy URI	sha256:5f9a8d0e6a1a2851d0639ceaf2514f...	-
-	image	November 11, 2025, 17:31:08 (UTC+05:5)	0.00	Copy URI	sha256:b25c344df2118ddceaf86777aa1a0...	-

- Actions for each image include **Details** and **View push commands**.



Cleanup everything



Delete Launch Template Request Succeeded

Launch Templates (2/2) Info

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Managed	Oper...
lt-06225678ece4bfde4	web-server-launch-template	1	1	2025-10-29T17:22:01.000Z	arnawsiam-323997748732:root	false	-
lt-0758d0a635f76xf65	eks-54cd5622-dfcc-0ced-9397-73f5aa456b4c				arnawssts-323997748732:ass...	false	-

Delete Launch Templates

You can't undo this action. Any Auto Scaling groups or Spot Fleet requests currently using these launch templates might be affected.

Are you sure you want to delete 2 Launch Templates, and all related versions?

- lt-06225678ece4bfde4 (web-server-launch-template)
- lt-0758d0a635f76xf65 (eks-54cd5622-dfcc-0ced-9397-73f5aa456b4c)

To confirm deletion, type Delete in the field

Cancel **Delete**

Launch Templates: lt-06225678ece4bfde4, lt-0758d0a635f76xf65

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback 25°C Sunny ENG US 13:25 11-11-2025

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