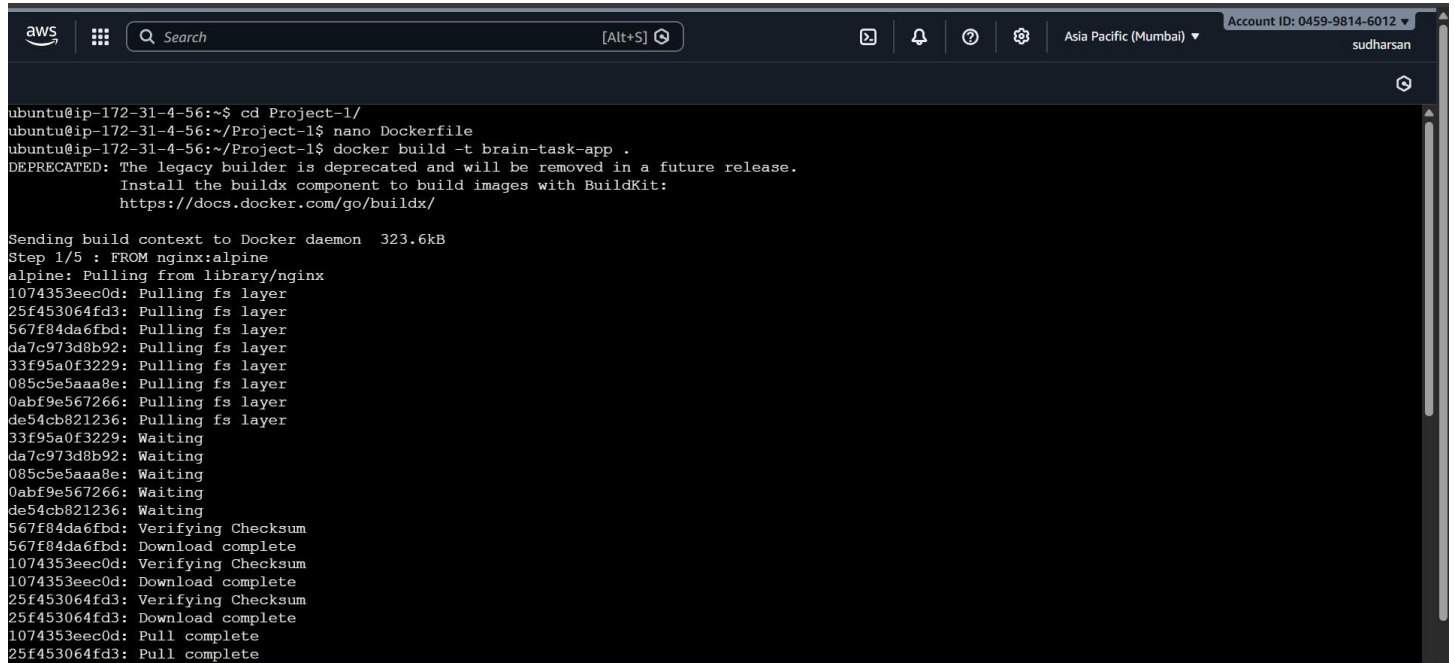


# FINAL PROJECT – 1

## MindTrack

### SCREENSHOT IMAGES

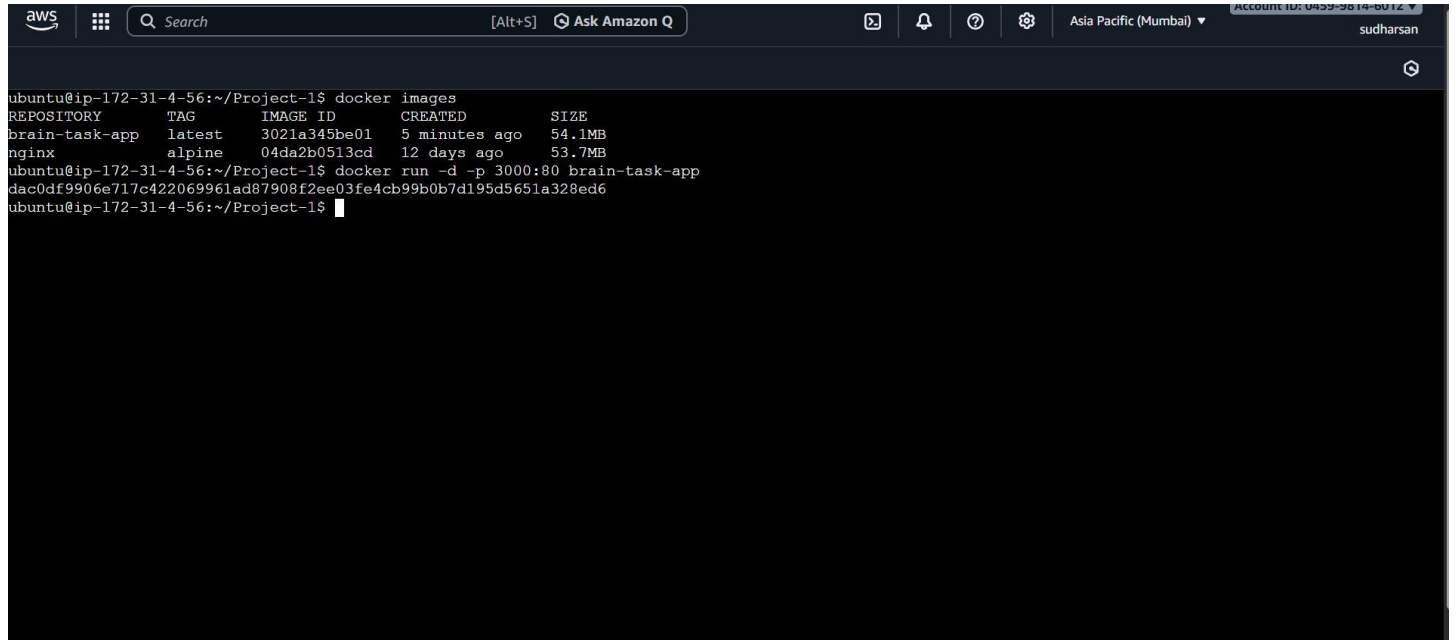
#### 1] docker build



```
ubuntu@ip-172-31-4-56:~$ cd Project-1/
ubuntu@ip-172-31-4-56:~/Project-1$ nano Dockerfile
ubuntu@ip-172-31-4-56:~/Project-1$ docker build -t brain-task-app .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  323.6kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
1074353eec0d: Pulling fs layer
25f453064fd3: Pulling fs layer
567f84da6fbd: Pulling fs layer
da7c973d8b92: Pulling fs layer
33f95a0f3229: Pulling fs layer
085c5e5aaa8e: Pulling fs layer
0abf9e567266: Pulling fs layer
de54cb821236: Pulling fs layer
33f95a0f3229: Waiting
da7c973d8b92: Waiting
085c5e5aaa8e: Waiting
0abf9e567266: Waiting
de54cb821236: Waiting
567f84da6fbd: Verifying Checksum
567f84da6fbd: Download complete
1074353eec0d: Verifying Checksum
1074353eec0d: Download complete
25f453064fd3: Verifying Checksum
25f453064fd3: Download complete
1074353eec0d: Pull complete
25f453064fd3: Pull complete
```

#### 2] docker image, run



```
ubuntu@ip-172-31-4-56:~/Project-1$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
brain-task-app       latest          3021a345be01   5 minutes ago   54.1MB
nginx                alpine         04da2b0513cd   12 days ago     53.7MB
ubuntu@ip-172-31-4-56:~/Project-1$ docker run -d -p 3000:80 brain-task-app
dac0df9906e717c422069961ad87908f2ee03fe4cb99b0b7d195d5651a328ed6
ubuntu@ip-172-31-4-56:~/Project-1$
```

### 3] ECR repo creation

```
aws
[Alt+S]
Account ID: 0459-9814-6012
sudharsan

ubuntu@ip-172-31-4-56:~/Project-1$ aws --version
aws-cli/2.32.25 Python/3.13.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-4-56:~/Project-1$ aws ecr create-repository \
--repository-name brain-task-app \
--region ap-south-1
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:ap-south-1:045998146012:repository/brain-task-app",
    "registryId": "045998146012",
    "repositoryName": "brain-task-app",
    "repositoryUri": "045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app",
    "createdAt": "2025-12-30T09:29:16.838000+00:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}
ubuntu@ip-172-31-4-56:~/Project-1$ docker tag brain-tasks-app:latest 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
Error response from daemon: No such image: brain-tasks-app:latest
ubuntu@ip-172-31-4-56:~/Project-1$ docker tag brain-task-app:latest 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
ubuntu@ip-172-31-4-56:~/Project-1$ docker push 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
The push refers to repository [045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app]
93f4306c3a98: Preparing
1d30230573f3: Preparing
e6fe11fa5b7f: Preparing
67ea0b046e7d: Preparing
ed5fa8595c7a: Preparing
```

### 4] image push to ECR from docker

```
aws
[Alt+S]
Account ID: 0459-9814-6012
sudharsan

67ea0b046e7d: Preparing
ed5fa8595c7a: Preparing
8ae63eb1f31f: Waiting
b3e3dlbb64d: Waiting
48078b7e3000: Waiting
fdle40d7f74b: Waiting
7bb20cf5ef67: Waiting
no basic auth credentials
ubuntu@ip-172-31-4-56:~/Project-1$ aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app

WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-4-56:~/Project-1$ docker push 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
The push refers to repository [045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app]
93f4306c3a98: Pushed
1d30230573f3: Pushed
e6fe11fa5b7f: Pushed
67ea0b046e7d: Pushed
ed5fa8595c7a: Pushed
8ae63eb1f31f: Pushed
b3e3dlbb64d: Pushed
48078b7e3000: Pushed
fdle40d7f74b: Pushed
7bb20cf5ef67: Pushed
latest: digest: sha256:f6770b4c6758db269f2aecd2f9d5bc26d0b5a4301da849667a2092124d1a5ae9 size: 2406
ubuntu@ip-172-31-4-56:~/Project-1$
```

## 5] ECR repo in aws

The screenshot displays the AWS Management Console for the Amazon Elastic Container Registry (ECR). The left sidebar shows the navigation menu with 'Private registry' selected. The main content area shows the 'brain-task-app' repository with the 'Images' tab active. A table lists the images, showing one image named 'latest' created on 30 December 2025.

| Image tags | Type  | Created at                            | Image size | Image digest    | Last pulled at |
|------------|-------|---------------------------------------|------------|-----------------|----------------|
| latest     | Image | 30 December 2025, 15:05:52 (UTC+05.5) | 23.10      | sha256:f6770... | -              |

## 6] EKS cluster creation

```
ubuntu@ip-172-31-4-56:~/Project-1$ eksctl version
0.221.0
ubuntu@ip-172-31-4-56:~/Project-1$ eksctl create cluster \
> --name brain-cluster \
> --region ap-south-1 \
> --nodegroup-name brain-nodes \
> --node-type t3.small \
> --nodes 2 \
> --managed
2025-12-30 09:51:21 [i] eksctl version 0.221.0
2025-12-30 09:51:21 [i] using region ap-south-1
2025-12-30 09:51:21 [i] setting availability zones to [ap-south-1b ap-south-1a ap-south-1c]
2025-12-30 09:51:21 [i] subnets for ap-south-1b - public:192.168.0.0/19 private:192.168.96.0/19
2025-12-30 09:51:21 [i] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
2025-12-30 09:51:21 [i] subnets for ap-south-1c - public:192.168.64.0/19 private:192.168.160.0/19
2025-12-30 09:51:21 [i] nodegroup "brain-nodes" will use "" [AmazonLinux2023/1.32]
2025-12-30 09:51:21 [i] Auto Mode will be enabled by default in an upcoming release of eksctl. This means managed node groups and managed networking add-ons will no longer be created by default. To maintain current behavior, explicitly set 'autoModeConfig.enabled: false' in your cluster configuration. Learn more: https://eksctl.io/usage/auto-mode/
2025-12-30 09:51:21 [i] using Kubernetes version 1.32
2025-12-30 09:51:21 [i] creating EKS cluster "brain-cluster" in "ap-south-1" region with managed nodes
2025-12-30 09:51:21 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-12-30 09:51:21 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=brain-cluster'
2025-12-30 09:51:21 [i] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "brain-cluster" in "ap-south-1"
2025-12-30 09:51:21 [i] CloudWatch logging will not be enabled for cluster "brain-cluster" in "ap-south-1"
2025-12-30 09:51:21 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=brain-cluster'
2025-12-30 09:51:21 [i] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-12-30 09:51:21 [i]
```

## 7] EKS cluster and node

```
aws [Alt+S] Ask Amazon Q Asia Pacific (Mumbai) Account ID: 0459-9814-6012 sudharsan

2025-12-30 10:02:24 [i] waiting for CloudFormation stack "eksctl-brain-cluster-nodgroup-brain-nodes"
2025-12-30 10:02:55 [i] waiting for CloudFormation stack "eksctl-brain-cluster-nodgroup-brain-nodes"
2025-12-30 10:03:45 [i] waiting for CloudFormation stack "eksctl-brain-cluster-nodgroup-brain-nodes"
2025-12-30 10:05:38 [i] waiting for CloudFormation stack "eksctl-brain-cluster-nodgroup-brain-nodes"
2025-12-30 10:05:38 [i] waiting for the control plane to become ready
2025-12-30 10:05:39 [✓] saved kubeconfig as "/home/ubuntu/.kube/config"
2025-12-30 10:05:39 [i] no tasks
2025-12-30 10:05:39 [✓] all EKS cluster resources for "brain-cluster" have been created
2025-12-30 10:05:39 [i] nodegroup "brain-nodes" has 2 node(s)
2025-12-30 10:05:39 [i] node "ip-192-168-12-163.ap-south-1.compute.internal" is ready
2025-12-30 10:05:39 [i] node "ip-192-168-87-115.ap-south-1.compute.internal" is ready
2025-12-30 10:05:39 [i] waiting for at least 2 node(s) to become ready in "brain-nodes"
2025-12-30 10:05:39 [i] nodegroup "brain-nodes" has 2 node(s)
2025-12-30 10:05:39 [i] node "ip-192-168-12-163.ap-south-1.compute.internal" is ready
2025-12-30 10:05:39 [i] node "ip-192-168-87-115.ap-south-1.compute.internal" is ready
2025-12-30 10:05:39 [✓] created 1 managed nodegroup(s) in cluster "brain-cluster"
2025-12-30 10:05:39 [i] creating addon: metrics-server
2025-12-30 10:05:40 [i] successfully created addon: metrics-server
2025-12-30 10:05:41 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2025-12-30 10:05:41 [✓] EKS cluster "brain-cluster" in "ap-south-1" region is ready
ubuntu@ip-172-31-4-56:~/Project-1$ eksctl get cluster
Error: AWS Region must be set, please set the AWS Region in AWS config file or as environment variable
ubuntu@ip-172-31-4-56:~/Project-1$ eksctl get cluster --region ap-south-1
NAME          REGION    EKSCTL CREATED
brain-cluster ap-south-1 True
ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-192-168-12-163.ap-south-1.compute.internal Ready    <none>    15m    v1.32.9-eks-ecaa3a6
ip-192-168-87-115.ap-south-1.compute.internal Ready    <none>    15m    v1.32.9-eks-ecaa3a6
ubuntu@ip-172-31-4-56:~/Project-1$
```

## 8] EKS deployment and service(LB)

```
aws [Alt+S] Ask Amazon Q Asia Pacific (Mumbai) Account ID: 0459-9814-6012 sudharsan

ubuntu@ip-172-31-4-56:~/Project-1$ nano Deployment.yaml
ubuntu@ip-172-31-4-56:~/Project-1$ kubectl apply -f Deployment.yaml
deployment.apps/brain-task-deployment created
ubuntu@ip-172-31-4-56:~/Project-1$ nano Service.yaml
ubuntu@ip-172-31-4-56:~/Project-1$ kubectl apply -f Service.yaml
service/brain-task-service created
ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get Deployment
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
brain-task-deployment 2/2        2              2            3m5s
ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
brain-task-service LoadBalancer 10.100.56.167  a466e7725455c4f239d05b8f319bb328-1281879047.ap-south-1.elb.amazonaws.com 80:30546/TCP 63s
kubernetes    ClusterIP     10.100.0.1     <none>          443/TCP          28m
ubuntu@ip-172-31-4-56:~/Project-1$
```



## 9] EKS dash in aws

The screenshot displays the Amazon Elastic Kubernetes Service (EKS) console for the 'brain-cluster'. The left sidebar shows the navigation menu with options like Dashboard, Clusters, Settings, Amazon EKS Anywhere, and Related services. The main content area shows the cluster's status as 'Active' and provides a warning that the current Kubernetes version (1.32) will reach the end of standard support on March 23, 2026. A table of cluster health metrics shows 0 issues across various categories. The 'Details' section lists the API server endpoint, OpenID Connect provider URL, and creation time (44 minutes ago).

**Amazon Elastic Kubernetes Service**

Dashboard  
Clusters

▼ Settings  
Dashboard settings  
Console settings

▼ Amazon EKS Anywhere  
Enterprise Subscriptions

▼ Related services  
Amazon ECR  
AWS Batch

Documentation

### brain-cluster

[Delete cluster](#) [Upgrade version](#) [Monitor cluster](#)

⚠️ Your cluster's Kubernetes version (1.32) will reach the end of standard support on March 23, 2026. [Upgrade](#)

▼ **Cluster info** [Info](#)

|                               |  |   |                                  |
|-------------------------------|--|---|----------------------------------|
| <b>Status</b><br>✔️ Active    | <b>Kubernetes version</b> <a href="#">Info</a><br>1.32 | <b>Support period</b><br>⚠️ Standard support until March 23, 2026 | <b>Provider</b><br>EKS           |
| <b>Cluster health</b><br>✔️ 0 | <b>Upgrade insights</b><br>✔️ 5                        | <b>Node health issues</b><br>✔️ 0                                 | <b>Capability issues</b><br>✔️ 0 |

◀ **Overview** Resources Compute Networking Add-ons 1 Capabilities Access Observability ▶

#### Details

|  |  |                                  |
|--|--|----------------------------------|
| <b>API server endpoint</b><br>https://231EB050284B70220496FA0B8035 | <b>OpenID Connect provider URL</b><br>https://oidc.eks.ap-south-1.amazonaws.co | <b>Created</b><br>44 minutes ago |
|--|--|----------------------------------|

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

## 10] code builds in aws

The screenshot shows the AWS CodeBuild console for a specific build. A green banner at the top indicates that the build started successfully. The build status is 'Succeeded'. The console provides details such as the build ARN, resolved source version, start time, end time, and build number. The left sidebar shows the navigation menu for Developer Tools, including Source, Artifacts, Build, and Related integrations.

Developer Tools  
**CodeBuild**

► Source • CodeCommit  
► Artifacts • CodeArtifact  
▼ **Build • CodeBuild**  
Getting started  
Build projects  
**Build project**  
Settings  
Build history  
Report groups  
Report history  
Compute fleets [New](#)  
Account metrics  
▼ Related integrations  
Jenkins

**Build started**  
You have successfully started the following build: brain-task-app-build:e9f337ed-f03f-4caf-946a-8b2789087f14

### brain-task-app-build:e9f337ed-f03f-4caf-946a-8b2789087f14

[Stop build](#) [Debug build](#) [Retry build](#)

#### Build status

|  |  |   |
|--|--|---|
| <b>Status</b><br>✔️ Succeeded  | <b>Initiator</b><br>root                             | <b>Build ARN</b><br>arn:aws:codebuild:ap-south-1:045998146012:build/brain-task-app-build:e9f337ed-f03f-4caf-946a-8b2789087f14 |
| <b>Resolved source version</b><br>949ecf100539985a8dd65bd73deb1627502b6d35 | <b>Start time</b><br>Dec 30, 2025 6:03 PM (UTC+5:30) | <b>End time</b><br>Dec 30, 2025 6:04 PM (UTC+5:30)  |
| <b>Build number</b><br>5   |  |   |

**Build logs** **Phase details** Reports Environment variables Build details Resource utilization

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

11] code build logs in aws

Developer Tools

CodeBuild

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Getting started

Build projects

Build project

Settings

Build history

Report groups

Report history

Compute fleets New

Account metrics

Related integrations

Jenkins

Build logs

Phase details

Reports

Environment variables

Build details

Resource utilization

| Name             | Status    | Context | Duration | Start time                      | End time                        |
|------------------|-----------|---------|----------|---------------------------------|---------------------------------|
| SUBMITTED        | Succeeded | -       | <1 sec   | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:03 PM (UTC+5:30) |
| QUEUED           | Succeeded | -       | <1 sec   | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:03 PM (UTC+5:30) |
| PROVISIONING     | Succeeded | -       | 10 secs  | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:03 PM (UTC+5:30) |
| DOWNLOAD_SOURCE  | Succeeded | -       | 3 secs   | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:03 PM (UTC+5:30) |
| INSTALL          | Succeeded | -       | <1 sec   | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:03 PM (UTC+5:30) |
| PRE_BUILD        | Succeeded | -       | 10 secs  | Dec 30, 2025 6:03 PM (UTC+5:30) | Dec 30, 2025 6:04 PM (UTC+5:30) |
| BUILD            | Succeeded | -       | 6 secs   | Dec 30, 2025 6:04 PM (UTC+5:30) | Dec 30, 2025 6:04 PM (UTC+5:30) |
| POST_BUILD       | Succeeded | -       | 1 sec    | Dec 30, 2025 6:04 PM (UTC+5:30) | Dec 30, 2025 6:04 PM (UTC+5:30) |
| UPLOAD_ARTIFACTS | Succeeded | -       | <1 sec   | Dec 30, 2025 6:04 PM (UTC+5:30) | Dec 30, 2025 6:04 PM (UTC+5:30) |
| FINALIZING       | Succeeded | -       | <1 sec   | Dec 30, 2025 6:04 PM (UTC+5:30) | Dec 30, 2025 6:04 PM (UTC+5:30) |
| COMPLETED        | Succeeded | -       | -        | Dec 30, 2025 6:04 PM (UTC+5:30) | -                               |

12] code pipeline

Developer Tools

CodePipeline

Pipelines

brain-task-app

brain-task-app

Edit

Stop execution

Create trigger

Clone pipeline

Release change

Pipeline

Executions

Triggers

Settings

Tags

Stage

Source

5cad58ee-24d3-4880-b8b0-38b8697f699c

All actions succeeded.

Source

GitHub (via OAuth app)

16 minutes ago

6f1a327e Source: changes to

Build

5cad58ee-24d3-4880-b8b0-38b8697f699c

All actions succeeded.

Build

AWS CodeBuild

15 minutes ago

6f1a327e Source: changes to

deploy

5cad58ee-24d3-4880-b8b0-38b8697f699c

All actions succeeded.

deploy

Amazon EKS

13 minutes ago

6f1a327e Source: changes to

## 13] codepipeline logs in aws cloudwatch

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > Log management > /aws/codebuild/brain-task-app-build > e768a783-345a-4fcb-a63c-dd3b874e3526. The left sidebar contains navigation links for CloudWatch, Favorites and recents, All alarms, AI Operations, GenAI Observability, Application Signals (APM), Infrastructure Monitoring, and Logs. The main content area is titled 'Log events' and includes a search bar, filters (1m, 1h, UTC timezone), and a 'Display' button. A table of log events is displayed with columns for Timestamp and Message. The messages show the execution of a CodeBuild task, including expanding directories, assembling file lists, and completing the upload of artifacts.

| Timestamp                | Message  |
|--------------------------|--|
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.516723 Expanding base directory path: .                          |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.519754 Assembling file list                                      |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.519766 Expanding .   |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.522808 Expanding file paths for base directory .                 |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.522821 Assembling file list                                      |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.522824 Expanding **/*  |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.526211 No matching auto-discover report paths found              |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.526238 Report auto-discover file discovery took 0.009553 seconds |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.526262 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED         |
| 2025-12-30T12:05:07.699Z | [Container] 2025/12/30 12:05:07.526272 Phase context status code: Message:                       |

No newer events at this moment. Auto retry paused. [Resume](#)

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > Log management > /aws/codepipeline/brain-task-app > 4cf8330f-4ae0-4c85-89f8-7a1fc9971f4d/9a02923a-3f81-44e8-86ba-6... The left sidebar contains navigation links for CloudWatch, Favorites and recents, All alarms, AI Operations, GenAI Observability, Application Signals (APM), Infrastructure Monitoring, and Logs. The main content area is titled 'Log events' and includes a search bar, filters (1m, 1h, UTC timezone), and a 'Display' button. A table of log events is displayed with columns for Timestamp and Message. The messages show the execution of a CodePipeline task, including running commands, checking for errors, and completing the build phase.

| Timestamp                | Message   |
|--------------------------|---|
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.788875 Running command [ -f /tmp/cp-action-source/action-output-variables.sh ]_   |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.797580 Running command if [[ ! -z \$CodePipeline_ErrorCode    ! -z \$CodePipel... |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.804107 Command did not exit successfully if [[ ! -z \$CodePipeline_ErrorCode ]]   |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.826066 Phase complete: BUILD State: FAILED  |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.826083 Phase context status code: COMMAND_EXECUTION_ERROR Message: Error while... |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.877265 Entering phase POST_BUILD  |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.896799 Phase complete: POST_BUILD State: SUCCEEDED                                |
| 2025-12-30T15:06:02.135Z | [Container] 2025/12/30 15:06:01.896818 Phase context status code: Message:  |

No newer events at this moment. Auto retry paused. [Resume](#)



## 14] EKS cluster log in aws cloudwatch

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation indicates the path: CloudWatch > Log management > /aws/eks/brain-cluster/cluster > kube-apiserver-audit-9db13d13f6b0144980d172c8bb627179. The left sidebar contains a 'Log Management' section with options like Log Anomalies, Live Tail, and Logs Insights. The main area is titled 'Log events' and displays a list of log entries with columns for Timestamp and Message. The messages are JSON objects representing audit events. At the bottom, there is a 'Back to top' button and a 'Resume' link.

| Timestamp                | Message  |
|--------------------------|--|
| 2025-12-27T15:20:09.829Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Request", "auditID": "b1e3817c-fffb-4a41-9dd7-9f6f4eb1f1..."} |
| 2025-12-27T15:20:10.830Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "fb396fd6-3580-4b7d-bd16-c550eb547..."} |
| 2025-12-27T15:20:11.081Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "7a45614d-0518-474d-8227-f3ceff5e6..."} |
| 2025-12-27T15:20:11.332Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "bb3ee4a8-cfd6-4117-ba3d-5e2d0e37b..."} |
| 2025-12-27T15:20:11.332Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "409eec42-2e4b-466a-90ba-4608465af..."} |
| 2025-12-27T15:20:11.585Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "a2e1317a-68a8-483a-a636-4369bd257..."} |
| 2025-12-27T15:20:12.336Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "d803d16c-1ee5-4d63-826a-8a2ec0cd4..."} |
| 2025-12-27T15:20:13.088Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "14b57d36-e13b-4bc9-a1b8-a5205d2f2..."} |
| 2025-12-27T15:20:13.590Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "b9190812-5926-40ce-a172-fb961190d..."} |
| 2025-12-27T15:20:13.841Z | {"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "685f26bb-585e-4a5d-b504-4fe893db7..."} |

## 15] loadbalancer service

The screenshot shows the AWS CloudShell terminal interface. The terminal displays a series of commands and their outputs. The commands include 'kubectl get deployments', 'kubectl get svc', 'kubectl delete -f Deployment.yaml', 'kubectl delete -f Service.yaml', and 'kubectl apply -f Deployment.yaml'. The outputs show the status of the 'brain-task-deployment' and 'brain-task-service' resources. The 'brain-task-deployment' is shown as 'Scaled up replica set brain-task-deployment-65c5c4f688 from 0 to 1' and 'Scaled down replica set brain-task-deployment-660f7cf47 from 2 to 1'. The 'brain-task-service' is shown as 'Scaled down replica set brain-task-deployment-660f7cf47 from 1 to 0' and 'Scaled up replica set brain-task-deployment-65c5c4f688 from 1 to 2'.

```
Events:
Type      Reason      Age      From      Message
-----
Normal    ScalingReplicaSet  49s     deployment-controller    Scaled up replica set brain-task-deployment-65c5c4f688 from 0 to 1
Normal    ScalingReplicaSet  47s     deployment-controller    Scaled down replica set brain-task-deployment-660f7cf47 from 2 to 1
Normal    ScalingReplicaSet  47s     deployment-controller    Scaled up replica set brain-task-deployment-65c5c4f688 from 1 to 2
Normal    ScalingReplicaSet  45s     deployment-controller    Scaled down replica set brain-task-deployment-660f7cf47 from 1 to 0

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
brain-task-deployment  2/2     2            2           19h

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
brain-task-service  LoadBalancer  10.100.56.167    a466e7725455c4f239d05b8f319bb328-1281879047.ap-south-1.elb.amazonaws.com  80:30546/TCP  19h
kubernetes           ClusterIP      10.100.0.1      <none>           443/TCP      20h

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl delete -f Deployment.yaml
deployment.apps "brain-task-deployment" deleted from default namespace

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl delete -f Service.yaml
service "brain-task-service" deleted from default namespace

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl apply -f Deployment.yaml
deployment.apps/brain-task-deployment created

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl apply -f Service.yaml
service/brain-task-service created

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
brain-task-deployment  2/2     2            2           19s

ubuntu@ip-172-31-4-56:~/Project-1$ kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
brain-task-service  LoadBalancer  10.100.238.64    aecbfa337889b4768a16a8bf980a8f88-1995190548.ap-south-1.elb.amazonaws.com  80:32120/TCP  20s
kubernetes           ClusterIP      10.100.0.1      <none>           443/TCP      20h

ubuntu@ip-172-31-4-56:~/Project-1$
```



