

Application Deployment

Project :Mini Project-1

Title :Mind-Track

Application:

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

Repo URL : <https://github.com/M-Ashok07/Mind-Track.git>

Prerequisites:

I launched a new EC2 instance using AWS, which would serve as the base server for my deployment. Once the instance was up and running, I connected to it and began installing the required tools.

I installed and verified the necessary components for containerization and Kubernetes setup, including:

- Docker
- AWS CLI
- eksctl (for managing EKS clusters)
- kubectl (for interacting with Kubernetes)
- AWS IAM Authenticator

Each of these tools was installed and configured properly. I also configured my AWS credentials to enable secure access and interaction with AWS services from within the instance.

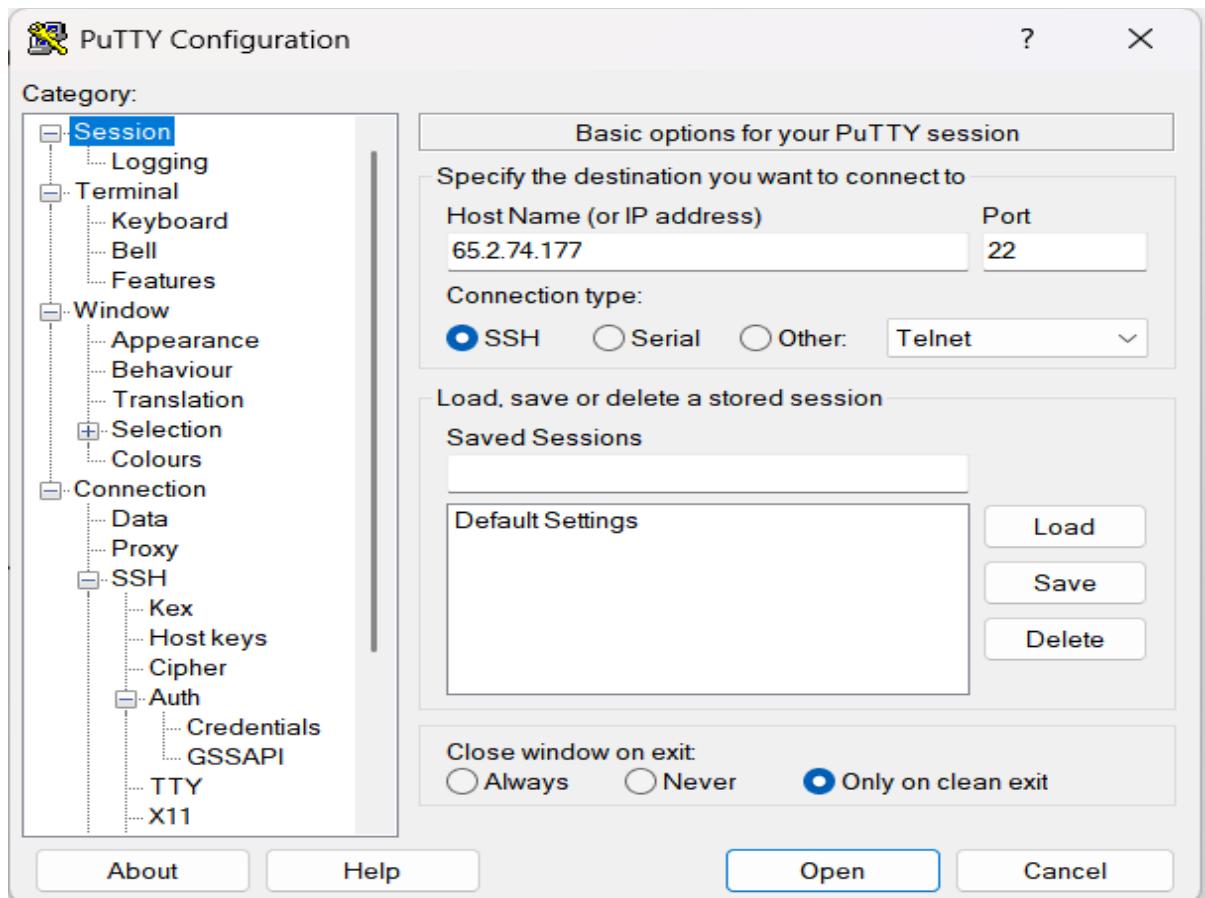
With everything successfully installed and the instance environment ready, I was all set to proceed with deploying and managing the application using Kubernetes on AWS.

-EC2 Instance

The screenshot shows the AWS EC2 Instances page with the following details for the instance i-08c25b0003aaca6c4:

- Instance ID:** i-08c25b0003aaca6c4 (Mind-Track)
- Public IPv4 address:** 65.2.74.177
- Private IP DNS name (IPv4 only):** ip-172-31-5-101.ap-south-1.compute.internal
- Instance state:** Running
- Instance type:** t3.micro
- VPC ID:** vpc-05b7a1450b0c195ef (default -vpc)
- Subnet ID:** subnet-0621502d030288dbc (default-sub-c)
- Instance ARN:** arn:aws:ec2:ap-south-1:304534110140:instance/i-08c25b0003aaca6c4
- Managed:** false

-Putty



```
ubuntu@ip-172-31-5-101:~$ login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Aug 22 04:42:19 UTC 2025

System load: 0.22 Temperature: -273.1 C
Usage of /: 11.9% of 14.46GB Processes: 117
Memory usage: 24% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.5.101

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

ubuntu@ip-172-31-5-101:~$
```

Docker :

```

root@ip-172-31-5-101:/home/Brain-Tasks-App# docker --version
Docker version 27.5.1, build 27.5.1-0ubuntu3-24.04.2
root@ip-172-31-5-101:/home/Brain-Tasks-App# 

● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-08-22 04:46:42 UTC; 4min 25s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
 Main PID: 2097 (dockerd)
   Tasks: 9
    Memory: 34.0M (peak: 35.3M)
      CPU: 38.9ms
     Group: 5
      CPU: /system.slice/docker.service
           └─ 2097 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 22 04:46:41 ip-172-31-5-101 systemd[1]: Starting docker.service - Docker Application Container Engine...
Aug 22 04:46:41 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:41.505403187Z" level=info msg="Starting up"
Aug 22 04:46:41 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:41.505416542Z" level=info msg="OTEL tracing is not configured, using no-op tracer provider"
Aug 22 04:46:41 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:41.505400037Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using resolv.conf: /run/resolvconf/resolv.conf"
Aug 22 04:46:42 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:42.057159308Z" level=info msg="Loading containers: start."
Aug 22 04:46:42 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:42.057159308Z" level=info msg="Loading containers: done."
Aug 22 04:46:42 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:42.057159308Z" level=info msg="Docker daemon" commit="27.5.1-0ubuntu3-24.04.2" containerd-snapshotter=false storage-driver="overlay2"
Aug 22 04:46:42 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:42.087461300Z" level=info msg="Daemon has completed initialization"
Aug 22 04:46:42 ip-172-31-5-101 dockerd[2097]: time="2025-08-22T04:46:42.128025839Z" level=info msg="API listen on /run/docker.sock"
Aug 22 04:46:42 ip-172-31-5-101 systemd[1]: Started docker.service - Docker Application Container Engine.
lines=1-2/22 (END)

```

AWS CLI:

```

root@ip-172-31-5-101:/home/Brain-Tasks-App# ./aws/install
You can now run: /usr/local/bin/aws --version
root@ip-172-31-5-101:/home/Brain-Tasks-App#

```

-kubernetes(EKS)

```

root@ip-172-31-5-101:/home/Brain-Tasks-App# curl -LO "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz"
tar -xzf eksctl_Linux_amd64.tar.gz
sudo mv eksctl /usr/local/bin
  % Total    % Received % Xferd  Average Speed   Time   Time Current
                                         Dload  Upload Total Spent   Left Speed
0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0
0     0     0     0     0     0     0     0     0     0     0     0     0
100 33.5M 100 33.5M 0     0     33.3M 0  0:00:01 0:00:01 0:00:01 33.3M
root@ip-172-31-5-101:/home/Brain-Tasks-App# eksctl version
0.212.0
root@ip-172-31-5-101:/home/Brain-Tasks-App# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
  % Total    % Received % Xferd  Average Speed   Time   Time Current
                                         Dload  Upload Total Spent   Left Speed
100 138 100 138 0     0     439 0     0     0     439
100 57.3M 100 57.3M 0     0     113M 0     0     0     113M
root@ip-172-31-5-101:/home/Brain-Tasks-App# kubectl version --client
Client Version: v1.33.4
Kustomize Version: v5.6.0
root@ip-172-31-5-101:/home/Brain-Tasks-App# curl -o aws-iam-authenticator https://amazon-eks.s3.us-west-2.amazonaws.com/1.15.10/2020-02-22/bin/linux/amd64/aws-iam-authenticator
  % Total    % Received % Xferd  Average Speed   Time   Time Current
                                         Dload  Upload Total Spent   Left Speed
100 33.6M 100 33.6M 0     0     0:00:04 0:00:04 0:00:04 8143k
root@ip-172-31-5-101:/home/Brain-Tasks-App# chmod +x ./aws-iam-authenticator
root@ip-172-31-5-101:/home/Brain-Tasks-App# mv ./aws-iam-authenticator /usr/local/bin
root@ip-172-31-5-101:/home/Brain-Tasks-App# aws-iam-authenticator version
{"Version":"v0.5.0","Commit":"1fce2a90f68381eacd/bddcfab2bf689e7feeb8b4b"}
root@ip-172-31-5-101:/home/Brain-Tasks-App# curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
  % Total    % Received % Xferd  Average Speed   Time   Time Current
                                         Dload  Upload Total Spent   Left Speed
100 11913 100 11913 0     0     297k 0     0     0     0     298k
Downloading https://get.helm.sh/heilm-v3.18.6-linux-amd64.tar.gz
Verifying checksum... done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/heilm
root@ip-172-31-5-101:/home/Brain-Tasks-App# helm version
version.BuildInfo{Version:"v3.18.6", GitCommit:"b76a950f6835474e0906b96c9ec68a2eff3a6430", GitTreeState:"clean", GoVersion:"go1.24.6"}
root@ip-172-31-5-101:/home/Brain-Tasks-App# curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.13.3/docs/install/iam_policy.json
  % Total    % Received % Xferd  Average Speed   Time   Time Current
                                         Dload  Upload Total Spent   Left Speed
100 8955 100 8955 0     0     34058 0     0     0     0     34179
root@ip-172-31-5-101:/home/Brain-Tasks-App# aws iam create-policy \

```

Application:

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

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

Step 1: Project Setup and Cloning

I began the project by creating a local workspace and cloning the **Brain Tasks App** from the provided GitHub repository. After downloading the source code, I checked the project folder to ensure all necessary files were present.

Inside the project, I found a dist folder containing the main HTML file and other assets needed for deployment. With the files successfully cloned, the project was ready to be set up and run locally.

Screenshot 1:

```
ubuntu@ip-172-31-5-101:~$ sudo su -
root@ip-172-31-5-101:~# cd ..
root@ip-172-31-5-101:~/home#
root@ip-172-31-5-101:~/home# ls
ubuntu
root@ip-172-31-5-101:~/home# git clone https://github.com/Vennilavan12/Brain-Tasks-App.git
Cloning into 'Brain-Tasks-App'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 5 (from 1)
Receiving objects: 100% (8/8), 100.04 KiB | 11.12 MiB/s, done.
root@ip-172-31-5-101:~/home#
```

Step1:

Docker:

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

Step 2: Setting Up the EC2 Environment

After cloning the project, I moved on to setting up the cloud environment needed to deploy and test the application.

```
root@ip-172-31-5-101:~/home/Brain-Tasks-App/dist# ls
Dockerfile  assets  index.html  vite.svg
root@ip-172-31-5-101:~/home/Brain-Tasks-App/dist# mv Dockerfile assets index.html
vite.svg  /home/Brain-Tasks-App/
root@ip-172-31-5-101:~/home/Brain-Tasks-App/dist# cd ..
root@ip-172-31-5-101:~/home/Brain-Tasks-App# ls
Dockerfile  awscliv2.zip  eksctl_linux_amd64.tar.gz  k8s
assets      buildspec.yaml  iam_policy.json        kubeconfig
aws         dist          index.html            vite.svg
root@ip-172-31-5-101:~/home/Brain-Tasks-App# rm -r dist/
root@ip-172-31-5-101:~/home/Brain-Tasks-App# ls
Dockerfile  awscliv2.zip  iam_policy.json        kubeconfig
assets      buildspec.yaml  index.html            vite.svg
aws         eksctl_linux_amd64.tar.gz  k8s
root@ip-172-31-5-101:~/home/Brain-Tasks-App# nano .dockerignore
root@ip-172-31-5-101:~/home/Brain-Tasks-App# nano Dockerfile
root@ip-172-31-5-101:~/home/Brain-Tasks-App# docker build -t brain-task-app-1 .
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 399.7MB
Step 1/5 : FROM nginx:alpine
--> 4a86014ec699
Step 2/5 : RUN rm -rf /usr/share/nginx/html/*
--> Using cache
--> 87c52ba53a3a
Step 3/5 : COPY . /usr/share/nginx/html/
--> 9b45799e737e
Step 4/5 : EXPOSE 80
--> Running in fc8230bc5bf0
--> Removed intermediate container fc8230bc5bf0
--> 3eea95958257
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
--> Running in d88f4e82c4f6
--> Removed intermediate container d88f4e82c4f6
--> 33f12f2d7d12
Successfully built 33f12f2d7d12
Successfully tagged brain-task-app-1:latest
root@ip-172-31-5-101:~/home/Brain-Tasks-App#
```

Build Docker Image:

The Docker image was built using the command:

```
docker build -t brain-task-app1
```

Confirmation:

The image was verified using:

```
docker images
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App# 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 288.5MB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
9824c27679d3: Pull complete
6bc572a340ec: Pull complete
403e3f251637: Pull complete
9adfbbae99cb7: Pull complete
7a8a46741e18: Pull complete
c9ebbe2ff2fd2c: Pull complete
a592fbc61ecc: Pull complete
cb1ff408ef82: Pull complete
Digest: sha256:42a516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98605ba5e3b26ab8
Status: Downloaded newer image for nginx:alpine
--> 4a86014ec699
Step 2/5 : RUN rm -rf /usr/share/nginx/html/*
--> Running in 688a63de494f
--> Removed intermediate container 688a63de494f
--> 87c52ba53a3a
Step 3/5 : COPY dist/ /usr/share/nginx/html/
--> cd9df1ef6bc9
Step 4/5 : EXPOSE 80
--> Running in c4874f779bd4
--> Removed intermediate container c4874f779bd4
--> 2f120f374083
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
--> Running in fb712eb8f621
--> Removed intermediate container fb712eb8f621
--> f7b29c42d832
Successfully built f7b29c42d832
Successfully tagged brain-task-app:latest
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker images
REPOSITORY          TAG           IMAGE ID    CREATED       SIZE
brain-task-app     latest        f7b29c42d832   6 seconds ago  52.9MB
nginx              alpine        4a86014ec699   8 days ago   52.5MB
root@ip-172-31-5-101:/home/Brain-Tasks-App#
```

Run Docker Container

The application container was started using:

```
docker run -itd -p 3000:80 brain-task:latest
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App#
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker run -itd -p 3000:80 brain-task-app
6745b41f163c5376c45fa0715c6bdef97a1964d563637839066361e394b3eb6b
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                 NAMES
6745b41f163c        brain-task-app      "/docker-entrypoint..."   3 seconds ago      Up 3 seconds        0.0.0.0:3000->80/tcp, [::]:3000->80/tcp   unruffled_easley
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                 NAMES
6745b41f163c        brain-task-app      "/docker-entrypoint..."   8 seconds ago      Up 8 seconds        0.0.0.0:3000->80/tcp, [::]:3000->80/tcp   unruffled_easley
root@ip-172-31-5-101:/home/Brain-Tasks-App#
```

Explanation:

- -itd: Runs the container in interactive, TTY, and detached mode.
- brain-task:latest: The name and tag of the built image.

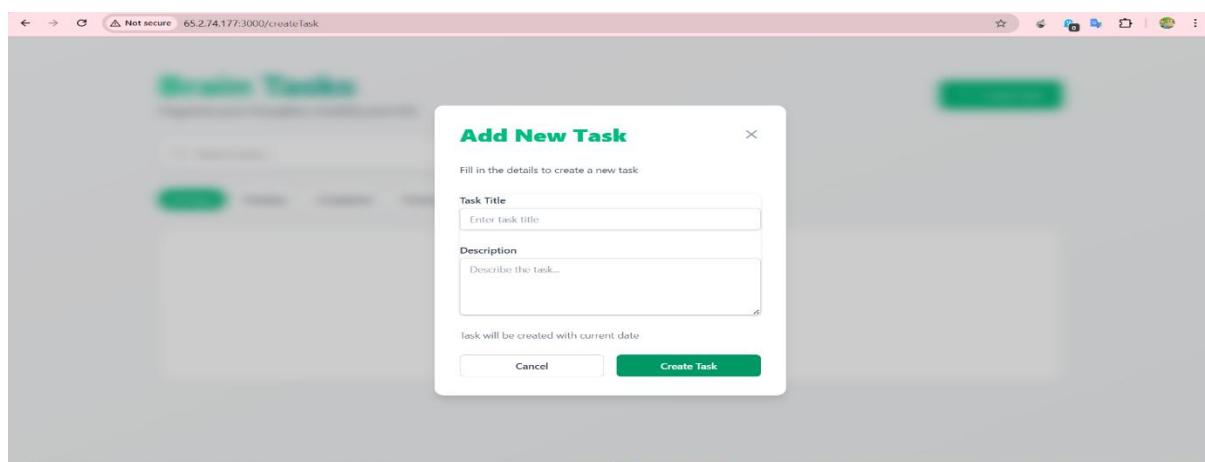
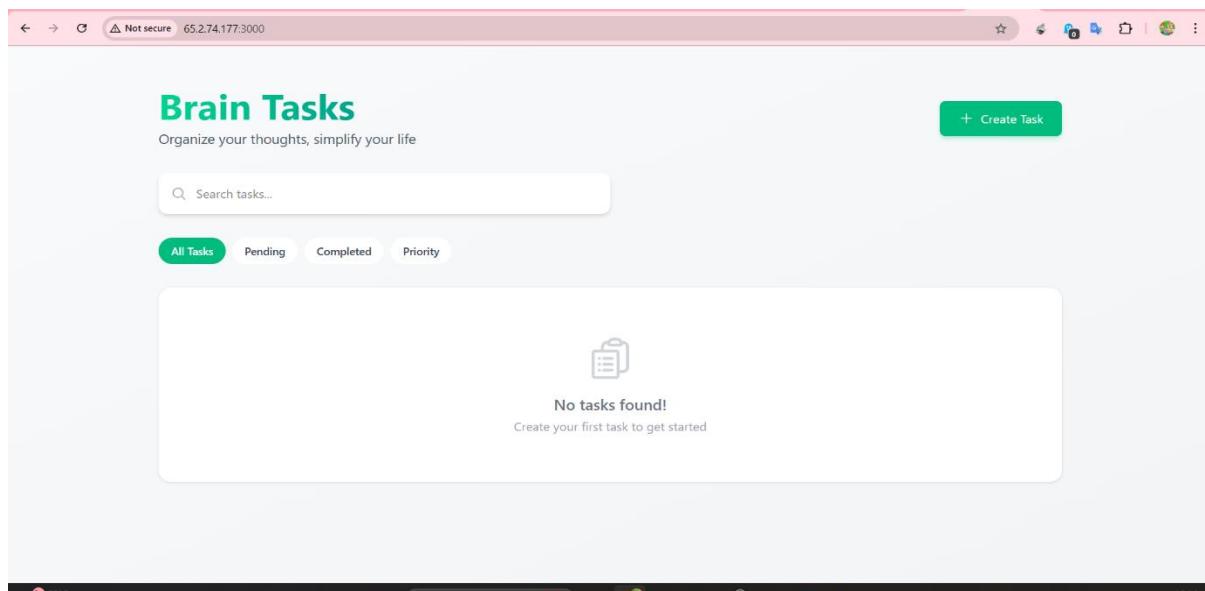
Result:

- A running container ID was shown.
- The application became accessible via the browser.

Example URL:<http://65.2.74.177:3000>

Output in Browser

Navigating to <http://13.204.47.24:3000> in a browser displayed the Brain Tasks UI



Step2:

ECR: Create an AWS ECR Repository to Store Docker Images

To store the Docker image for deployment, an AWS Elastic Container Registry (ECR) repository was created and used.

```
root@ip-172-31-5-101:/home/Brain-Tasks-App
root@ip-172-31-5-101:/home/Brain-Tasks-App# aws ecr create-repository --repository-name brain-task-app --region ap-south-1
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:ap-south-1:304534110140:repository/brain-task-app",
    "registryId": "304534110140",
    "repositoryName": "brain-task-app",
    "repositoryUri": "304534110140.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app",
    "createdAt": "2025-08-22T10:32:53.720000+00:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}
root@ip-172-31-5-101:/home/Brain-Tasks-App#
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App
root@ip-172-31-5-101:/home/Brain-Tasks-App# aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 304534110140.dkr.ecr.ap-south-1.amazonaws.com
WARNING! Your password will be stored unencrypted in '/root/.docker/config.json'.
This password will be valid for only 12 hours. To improve security, consider using a short expiration time or a more secure release.
Install the buildx component to build images with BuildKit:
  https://docs.docker.com/go/buildx/
Login Succeeded
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker build -t brain-task-app .
Step 1/5 : FROM alpine:latest
--> 4a86014ec99
Step 2/5 : RUN rm -rf /usr/share/nginx/html/*
--> 87c520ba33a3
Step 3/5 : COPY ./nginx /usr/share/nginx/html/
--> Using cache
--> 9b45799e737e
Step 4/5 : EXPOSE 80
--> Using cache
--> 3ec0a95958257
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
--> Using cache
Successfully built 33f12f2d7d12
Successfully tagged brain-task-app:latest
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker tag brain-task-app:latest 304534110140.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
root@ip-172-31-5-101:/home/Brain-Tasks-App# docker push 304534110140.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
The push refers to a repository [304534110140.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app]
dcb7007f007f: Pushed
377fb89511200: Pushed
d200138be39d: Pushed
a3b57470e9f1: Pushed
d3131333533d: Pushed
36ca725632e5: Pushed
79923333439d: Pushed
bfff0212304e: Pushed
418dc77d5a: Pushed
last: digest: 304534110140:b8993648c6f400b96bf7214ab0e5bb4da76d5110900dca50fe982d64e72686 size: 2409
root@ip-172-31-5-101:/home/Brain-Tasks-App#
```

The screenshots show the AWS ECR console interface. The top screenshot displays the 'Private repositories' list, showing a single repository named 'brain-task-app'. The bottom screenshot shows the detailed view of the 'brain-task-app' repository, specifically the 'Images' list, which contains one image tag named 'latest'.

Repository name	URI	Created at	Tag immutability	Encryption type
brain-task-app	304534110140.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app	August 22, 2025, 16:02:53 (UTC+05:5)	Mutable	AES-256

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	August 22, 2025, 10:50:14 (UTC+05:5)	22.57	<input type="button" value="Copy URI"/>	<input type="button" value="sha256:b4e2e0bde6c..."/>	-

Step3:

Kubernetes:

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

AWS ECR and Kubernetes Deployment

ECR (Elastic Container Registry):

- Created an AWS ECR repository named **brain-task** to store Docker images.
- Pushed a Docker image to the ECR repository using the following steps:

Created a shell script ECR_repo for repository creation.

1. Gave execute permission and ran the script.
2. Logged in to ECR using the AWS CLI and Docker.
3. Successfully pushed the Docker image to the ECR repository.

Kubernetes on AWS EKS:

- Created an EKS cluster named **Brain-task** in the **ap-south-1** region using eksctl.
- Provisioned with one managed node (type t3.medium) and default networking setup.
- Confirmed that:
 - EKS cluster stack was created via CloudFormation.
 - All EKS components and add-ons (like metrics-server, vpc-cni, kube-proxy, and coredns) were installed.
 - Node registered and was in a ready state.

Deployment on Kubernetes:

1. Created a namespace:
2. kubectl create namespace mind
3. Applied deployment and service YAML files:
4. kubectl apply -f deployment.yaml
5. kubectl apply -f ingress.yaml
6. kubectl apply -f service.yaml
7. Verified deployment and pod status:
 - Checked if pods were created and running.
 - Encountered a note: No resources found in the default namespace, ensure to use the correct namespace (-n mind) while querying.

```
kubectl get pods -n mind
```

```

root@ip-172-31-5-101:/home/Brain-Tasks-App
2025-08-22 05:31:30 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2025-08-22 05:32:30 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2025-08-22 05:33:30 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2025-08-22 05:34:30 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2025-08-22 05:34:30 [!] creating addon: metrics-server
2025-08-22 05:34:31 [!] successfully created addon: metrics-server
2025-08-22 05:34:31 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-08-22 05:34:31 [!] creating addon: vpc-cni
2025-08-22 05:34:31 [!] successfully created addon: vpc-cni
2025-08-22 05:34:31 [!] creating addon: kube-proxy
2025-08-22 05:34:32 [!] successfully created addon: kube-proxy
2025-08-22 05:34:32 [!] creating addon: coredns
2025-08-22 05:34:32 [!] successfully created addon: coredns
2025-08-22 05:36:33 [!] building managed nodegroup stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:36:33 [!] deploying stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:36:33 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:37:03 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:38:00 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:38:48 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:39:22 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:41:21 [!] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-standard-nodes"
2025-08-22 05:41:21 [!] waiting for EKS cluster to become ready
2025-08-22 05:41:21 [!] waiting for EKS cluster to become ready
2025-08-22 05:41:21 [!] waiting for EKS cluster to become ready
2025-08-22 05:41:21 [!] no tasks
2025-08-22 05:41:21 [!] all EKS cluster resources for "brain-tasks-cluster" have been created
2025-08-22 05:41:21 [!] nodegroup "standard-nodes" has 2 node(s)
2025-08-22 05:41:21 [!] node "ip-192-168-34-27.ap-south-1.compute.internal" is ready
2025-08-22 05:41:21 [!] node "ip-192-168-88-85.ap-south-1.compute.internal" is ready
2025-08-22 05:41:21 [!] waiting for at least 1 node(s) to become ready in "standard-nodes"
2025-08-22 05:41:21 [!] nodegroup "standard-nodes" has 2 node(s)
2025-08-22 05:41:21 [!] node "ip-192-168-34-27.ap-south-1.compute.internal" is ready
2025-08-22 05:41:21 [!] node "ip-192-168-88-85.ap-south-1.compute.internal" is ready
2025-08-22 05:41:21 [!] created 1 managed nodegroup(s) in cluster "brain-tasks-cluster"
2025-08-22 05:41:21 [!] creating Managed nodegroup(s) in cluster "brain-tasks-cluster"
2025-08-22 05:41:22 [!] cluster should be functional despite missing (or misconfigured) client binaries
2025-08-22 05:41:22 [!] EKS cluster "brain-tasks-cluster" in "ap-south-1" region is ready
root@ip-172-31-5-101:/home/Brain-Tasks-App# aws eks update-kubeconfig --region a
p-south-1 --name brain-tasks-cluster
Added new context:arn:aws:eks:ap-south-1:304534110140:cluster/brain-tasks-cluste
F to /root/.kube/config
root@ip-172-31-5-101:/home/Brain-Tasks-App# kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-192-168-34-27.ap-south-1.compute.internal   Ready   <none>   4m49s   v1.32.7
-eks-3abhc1
ip-192-168-88-85.ap-south-1.compute.internal   Ready   <none>   4m52s   v1.32.7
-eks-3abhc1
root@ip-172-31-5-101:/home/Brain-Tasks-App# 
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App# mkdir k8s
root@ip-172-31-5-101:/home/Brain-Tasks-App# cd k8s/
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano deployment.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano service.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano ingress.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# ls
deployment.yaml  ingress.yaml  service.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# cd ..
root@ip-172-31-5-101:~/# nano builspec.yaml
root@ip-172-31-5-101:~/# ls
builspec.yaml  snap
root@ip-172-31-5-101:~/# rm -r builspec.yaml
root@ip-172-31-5-101:~/# cd /home
root@ip-172-31-5-101:/home# ls
Brain-Tasks-App  ubuntu
root@ip-172-31-5-101:/home# cd Brain-Tasks-App/
root@ip-172-31-5-101:/home/Brain-Tasks-App# nano builspec.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App# ls
Dockerfile  aws  awscliv2.zip  builspec.yaml  dist  eksctl_Linux_amd64.tar.gz  iam_policy.json  k8s  kubectl
root@ip-172-31-5-101:/home/Brain-Tasks-App# cd k8s/
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f deployment.yaml
deployment.apps/brain-tasks-deployment created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f service.yaml
service/brain-tasks-service created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f ingress.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/brain-tasks-ingress created
```

```

root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s aws eks describe-cluster --name brain-tasks-cluster --region <your-region> \
--query "cluster.identity.oidc.issuer" --output text
-bash your-region: No such file or directory
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s aws eks describe-cluster --name brain-tasks-cluster --region ap-south-1 --query "cluster.identity.oidc.issuer" --output text
https://oidc.eks.ap-south-1.amazonaws.com/.well-known/openid-configuration
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s "C
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s eksctl utils associate-iam-oidc-provider \
--region ap-south-1 \
--cluster brain-tasks-cluster \
--approve
2025-08-22 06:00+04 [0] will create IAM Open ID Connect provider for cluster "brain-tasks-cluster" in "ap-south-1"
2025-08-22 06:00+04 [1] created IAM Open ID Connect provider for cluster "brain-tasks-cluster" in "ap-south-1"
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s curl -o iam.policy.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/main/docs/install/iam_policy.json
  Total  % Received % Xferd  Average Speed   Time   Time Current
          Dload Upload Total Spent   Left Speed
100 8955  100 8955  0  0 35304  0:--:--:--:--:--:--:-- 35395
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file:///iam_policy.json

An error occurred (EntityAlreadyExists) when calling the CreatePolicy operation: A policy called AWSLoadBalancerControllerIAMPolicy already exists. Duplicate names are not allowed.
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s eksctl create iamserviceaccount \
--cluster=brain-tasks-cluster \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--attach-policy-arm=arn:aws:iam::004534110140:policy/AWSLoadBalancerControllerIAMPolicy \
--override-existing-serviceaccounts \
--approve
2025-08-22 06:01+04 [0] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2025-08-22 06:01+04 [1] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2025-08-22 06:01+04 [0] 1 task(s):
  2 sequential sub-tasks:
    1 create IAM role for iamserviceaccount "kube-system/aws-load-balancer-controller",
      CreateServiceAccountRole "eksctl-brain-tasks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
    ) 2025-08-22 06:01+04 [0] Building iamserviceaccount stack "eksctl-brain-tasks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2025-08-22 06:01+04 [0] deploying stack "eksctl-brain-tasks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2025-08-22 06:01+04 [0] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2025-08-22 06:01+04 [0] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2025-08-22 06:01+04 [0] created serviceaccount "kube-system/aws-load-balancer-controller"
root@ip-172-31-5-101:/home/Brain-Tasks-App#k8s

```

Helm libraries :

```
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# helm repo add eks https://aws.github.io/eks-charts
helm repo update

helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
--in kube-system \
--set clusterName=brain-tasks-cluster \
--set serviceAccount.create=false \
--set serviceAccount.name=aws-load-balancer-controller \
--set region=ap-south-1 \
--set vpcId=vpc-07e07f379cb5742bf
"eks" has been added to your repositories
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "eks" chart repository
Update Complete. Happy Helming!
NAME: aws-load-balancer-controller
LAST DEPLOYED: Fri Aug 22 06:04:06 2025
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s#
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s#
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
brain-tasks-service   ClusterIP  10.100.193.124 <none>       80/TCP    23m
kubernetes       ClusterIP  10.100.0.1    <none>       443/TCP   72m
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl delete service brain-tasks-service
service "brain-tasks-service" deleted
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
brain-tasks-deployment-7b586f49fd-2kjlw   1/1    Running   0          50m
brain-tasks-deployment-7b586f49fd-fnrjg   1/1    Running   0          50m
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP  10.100.0.1    <none>       443/TCP   73m
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano service.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano service.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# nano ingress.yaml
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f service.yaml
service/brain-tasks-service created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f ingress.yaml
ingress.networking.k8s.io/brain-tasks-ingress configured
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s#
```

Step 1: Trying to set up permissions for the load balancer

You tried to create an IAM service account for the AWS Load Balancer Controller, but it didn't work at first because your EKS cluster didn't have the right OIDC provider linked. Basically, Kubernetes needs this to manage permissions properly with AWS.

No worries, you moved past this by installing the Load Balancer Controller directly using Helm, skipping the IAM service account step. And it installed fine!

Step 2: Checking if the load balancer controller is running

You then checked if the controller was actually running in your cluster, and yep — the pods are up and healthy. That means it's ready to do its job managing load balancers for your services.

Step 3: Creating a Kubernetes service with a LoadBalancer

You applied your service YAML file, which tells Kubernetes to create a LoadBalancer type service for your app. This gets you a public IP (well, a public DNS name that points to the AWS load balancer) so people can access your app from the internet.

You got this long URL that AWS assigned to your app, something like:

k8s-mind-brintas-1784b43c6e-b5724ada734ac91d.elb.ap-south-1.amazonaws.com

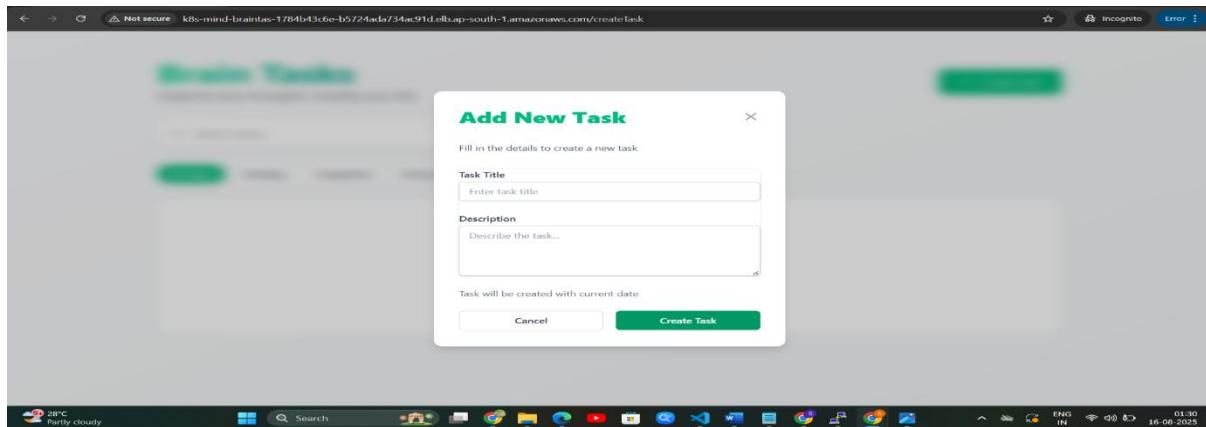
Step 4: Opening the app in a browser

You opened that URL in a browser, and boom — your Brain Tasks app loaded up! It looks good, ready for you to start adding and managing tasks.

```
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f deployment.yaml
deployment.apps/brain-tasks-deployment created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f service.yaml
service/brain-task-service created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl apply -f ingress.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/brain-task-ingress created
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-192-168-14-92.ap-south-1.compute.internal Ready <none> 9m11s v1.32.7-eks-3abbecl
ip-192-168-50-153.ap-south-1.compute.internal Ready <none> 9m9s v1.32.7-eks-3abbecl
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
brain-task-service ClusterIP 10.100.205.24 <none> 80/TCP 49s
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 15m
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s# kubectl get ingress
NAME CLASS HOSTS ADDRESS PORTS AGE
brain-task-ingress <none> * [k8s-default-braintas-589e456145-287728214.ap-south-1.elb.amazonaws.com] 80 53s
root@ip-172-31-5-101:/home/Brain-Tasks-App/k8s#
```

The screenshot shows the homepage of the Brain Tasks application. At the top, there is a navigation bar with icons for back, forward, refresh, and search. The URL is "Not secure k8s-default-braintas-cd8c721420-1528003788.ap-south-1.elb.amazonaws.com". Below the header, the title "Brain Tasks" is displayed in a large, bold, teal font. A subtitle "Organize your thoughts, simplify your life" follows. On the right side, there is a green button labeled "+ Create Task". A search bar with the placeholder "Search tasks..." is positioned below the title. Underneath the search bar, there are four filter tabs: "All Tasks" (which is selected and highlighted in green), "Pending", "Completed", and "Priority". In the center of the page, there is a large, light-gray rectangular area containing a small icon of a notepad and a pen. Below the icon, the text "No tasks found!" is displayed in a small, dark font. Underneath that, a smaller line of text says "Create your first task to get started".

The screenshot shows a modal dialog titled "Add New Task" centered on the screen. The background of the main window is blurred, showing the "Brain Tasks" interface. The modal has a white background and a close button "X" in the top right corner. Inside the modal, there is a instruction message "Fill in the details to create a new task". Below this, there are two input fields: "Task Title" and "Description". The "Task Title" field contains the placeholder "Enter task title". The "Description" field contains the placeholder "Describe the task...". At the bottom of the modal, there is a note "Task will be created with current date". Finally, at the very bottom, there are two buttons: "Cancel" on the left and "Create Task" on the right, which is highlighted in green.



I deployed an application in Kubernetes, and to check the external hostname of the service's load balancer, I used this command:

```
kubectl get svc brain-task-svc -n mind -o jsonpath='{.status.loadBalancer.ingress[0].hostname}'
```

This gave me the DNS name of the load balancer automatically created by Kubernetes in the AWS environment.

Next, I wanted to get more details about the actual AWS Elastic Load Balancer associated with this DNS name. So, I ran this AWS CLI command:

```
aws elbv2 describe-load-balancers --query "LoadBalancers[?DNSName=='SLB_DNS'].LoadBalancerArn" --output text --region ap-south-1
```

Replacing 'SLB_DNS' with the DNS name I got from Kubernetes. This returned the ARN (Amazon Resource Name) of the load balancer, which confirms the link between the Kubernetes service and the AWS ELB resource.

In short, these commands help me trace the load balancer details from Kubernetes service to the actual AWS resource managing the traffic.

```
root@ip-172-31-10-220:~/app/k8s# kubectl get svc brain-task-svc -n mind -o jsonpath='{.status.loadBalancer.ingress[0].hostname}'  
k8s-mind-briantas-1784b43c6e-b5724ada734ac91d.elb.ap-south-1.amazonaws.comroot@ip-172-31-10-220:~/app/k8s#  
9 28°C  
root@ip-172-31-10-220:~/app/k8s# aws elbv2 describe-load-balancers --query "LoadBalancers[?DNSName=='SLB_DNS'].LoadBalancerArn" --output text --region ap-south-1  
arn:aws:elasticloadbalancing:ap-south-1:304534110140:loadbalancer/net/k8s-mind-briantas-1784b43c6e/b5724ada734ac91d  
root@ip-172-31-10-220:~/app/k8s#
```

CodeBuild:

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

Connect to your repository

CodeBuild needs to know where your code lives. This could be GitHub, Bitbucket, CodeCommit, or even an S3 bucket. By connecting your repository, CodeBuild can automatically pull your latest code whenever a build is triggered.

Choose the environment

Every build runs inside a container that AWS manages for you. You don't have to set up a server or worry about dependencies. You can choose from pre-built environments such as **Amazon Linux** or **Ubuntu** that already include common tools (like Git, Docker, Node.js, Java, Python, etc.). This is where your build commands will actually run.

Define your build instructions (**buildspec.yml**)

CodeBuild needs to know what steps to follow to build your application. Instead of clicking buttons every time, you create a file called **buildspec.yml** inside your repo.

- This file is like a recipe book: it lists the commands for installing dependencies, running tests, compiling code, and packaging it.
- For example, you might use it to install Node.js packages, build a React app, or build and push a Docker image to ECR.

```
root@ip-172-31-10-220:~/app# aws codebuild create-project --name brain-task-build --source type=GITHUB,location=https://github.com/M-Ashok07/MindTrack.git --artifacts type=NO_ARTIFACTS --environment '{ "type": "LINUX_CONTAINER", "image": "aws/codebuild/standard:6.0", "computeType": "BUILD_GENERAL1_MEDIUM", "privilegedMode": true, "environmentVariables": [ { "name": "REPOSITORY_NAME", "value": "MindTrack"}, { "name": "LB_ARN", "value": "arn:aws:elasticloadbalancing:ap-south-1:304534110140:loadbalancer/net/k8s-mind-braintas-1784b43c6e/b5724ada734ac91d"} ] --service-role arn:aws:iam::304534110140:role/codebuild-service-role'
```



```
root@ip-172-31-10-220:~/app# --service-role arn:aws:iam::304534110140:role/codebuild-service-role
{
  "project": {
    "name": "brain-task-build",
    "arn": "arn:aws:codebuild:ap-south-1:304534110140:project/brain-task-build",
    "source": {
      "type": "GITHUB",
      "location": "https://github.com/M-Ashok07/MindTrack.git",
      "reportBuildStatus": false,
      "insecureSsl": false
    },
    "artifacts": {
      "type": "NO_ARTIFACTS"
    },
    "cache": {
      "type": "NO_CACHE"
    },
    "environment": {
      "type": "LINUX_CONTAINER",
      "image": "aws/codebuild/standard:6.0",
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "environmentVariables": [
        {
          "name": "REPOSITORY_NAME",
          "value": "MindTrack",
          "type": "PLAINTEXT"
        },
        {
          "name": "LB_ARN",
          "value": "arn:aws:elasticloadbalancing:ap-south-1:304534110140:loadbalancer/net/k8s-mind-braintas-1784b43c6e/b5724ada734ac91d",
          "type": "PLAINTEXT"
        }
      ],
      "privilegedMode": true,
      "imagePullCredentialsType": "CODEBUILD"
    },
    "serviceRole": "arn:aws:iam::304534110140:role/codebuild-service-role",
    "imageMinimumImageVersion": "2025-08-15T20:56:40.526000+00:00",
    "queueDrainTimeoutInMinutes": 400,
    "encryptionKey": "arn:aws:kms:ap-south-1:304534110140:alias/aws/s3",
    "lastModified": "2025-08-15T20:56:40.526000+00:00",
    "lastModifiedC": "2025-08-15T20:56:40.526000+00:00",
    "badge": {
      "badgeEnabled": false
    },
    "projectVisibility": "PRIVATE"
  }
}
root@ip-172-31-10-220:~/app#
```

Screenshot of the AWS CodeBuild console showing the 'Build projects' page. The left sidebar is titled 'CodeBuild' and includes sections for Source, Artifacts, Build, Report groups, Report history, Compute fleets, Account metrics, Related integrations (Jenkins, GitHub Actions, GitLab runners), Deploy, and Pipeline. The main content area shows a table with columns: Name, Source provider, Repository, Latest build status, Description, and Last Modified. A message at the bottom states 'No results' and 'There are no results to display.'

Screenshot of the 'Create build project' wizard. Step 1: Project configuration. It asks for a 'Project name' (set to 'brain-task-build') and 'Project type'. The 'Default project' option is selected, with a note: 'Create a custom CodeBuild project.' The 'Runner project' option is also available, described as 'Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.' Step 2: Additional configuration, which is collapsed. Step 3: Source, which is expanded. It shows 'Source 1 - Primary' and a 'Source provider' dropdown set to 'No source'. There is a 'Add source' button.

Source

Add source

Source 1 - Primary

Source provider

GitHub

Credential

Your account is successfully connected by using an AWS managed GitHub App. [Manage account credentials](#).

Use override credentials for this project only

Repository

Repository in my GitHub account Public repository GitHub scoped webhook

Repository

https://github.com/M-Ashok07/Mind-Track

Source version - optional [Info](#)

Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

Additional configuration

Git clone depth, Git submodules, Build status config

Environment

Provisioning model [Info](#)

On-demand
Automatically provision build infrastructure in response to new builds.

Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image

Managed image
Use an image managed by AWS CodeBuild

Custom image
Specify a Docker image

Compute

EC2
Optimized for flexibility during action runs

Lambda
Optimized for speed and minimizes the start up time of workflow actions

Running mode

Container
Running on Docker container

Instance
Running on EC2 instance directly

Operating system

Ubuntu

Runtimes

▼ Primary source webhook events [Info](#)

Webhook - optional [Info](#)

Rebuild every time a code change is pushed to this repository

Build type

Single build
Triggers single build

Batch build
Triggers multiple builds as single execution

Comment approval

ALL_PULL_REQUESTS

Approver roles

GITHUB_WRITE GITHUB_MAINTAIN GITHUB_ADMIN

▶ Webhook event filter groups [Add filter group](#)

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

▶ Additional configuration

aws [Alt+S] Account ID: 3045-3411-0140 Ashok

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
 - GitHub Actions
 - GitLab runners

Phase details						
Name	Status	Context	Duration	Start time	End time	
SUBMITTED	Succeeded	-	<1 sec	Aug 23, 2025 11:12 AM (UTC+5:30)	Aug 23, 2025 11:12 AM (UTC+5:30)	
QUEUED	Succeeded	-	<1 sec	Aug 23, 2025 11:12 AM (UTC+5:30)	Aug 23, 2025 11:12 AM (UTC+5:30)	
PROVISIONING	Succeeded	-	12 secs	Aug 23, 2025 11:12 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
DOWNLOAD_SOURCE	Succeeded	-	9 secs	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
INSTALL	Succeeded	-	<1 sec	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
PRE_BUILD	Succeeded	-	11 secs	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
BUILD	Succeeded	-	9 secs	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
POST_BUILD	Succeeded	-	<1 sec	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
UPLOAD_ARTIFACTS	Succeeded	-	9 secs	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
FINALIZING	Succeeded	-	<1 sec	Aug 23, 2025 11:13 AM (UTC+5:30)	Aug 23, 2025 11:13 AM (UTC+5:30)	
COMPLETED	Succeeded	-	-	Aug 23, 2025 11:13 AM (UTC+5:30)	-	

Introducing the new pipeline experience
We've redesigned the pipeline view to streamline the monitoring and debugging experience. Let us know what you think. Allow cookies to return to the old experience.

Developer Tools > CodePipeline > Pipelines > pipeline-code

pipeline-code  

Edit Stop execution Create trigger Clone pipeline Release change

Pipeline Executions Triggers Settings Tags Stage

Execution ID	Status	Source revisions	Trigger	Started	Duration	Completed
d0979ab7	 Succeeded	Source - 9149c908 : lambda commit	CreatePipeline - root	Aug 23, 2025 11:06 AM (UTC+5:30)	13 minutes 10 seconds	Aug 23, 2025 11:19 AM (UTC+5:30)

Developer Tools  **CodeBuild**

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
 - Getting started
 - Build projects**
 - Build history
 - Report groups
 - Report history
 - Compute fleets [New](#)
 - Account metrics
- Related integrations
 - Jenkins 
 - GitHub Actions 
 - GitLab runners 
- Deploy • CodeDeploy
- Pipeline • CodePipeline

Developer Tools > CodeBuild > Build projects

Build projects  Actions Create trigger View details Debug build Start build 

Search Your projects   

Name	Source provider	Repository	Latest build status	Description	Last Modified
brain-tasks-build	GitHub	M-Ashok07/Mind-Track	 Succeeded	-	18 hours ago

Version Control:

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

```
root@ip-172-31-5-101:/home/Brain-Tasks-App
Dockerfile assets aws awscli v2.zip buildspec.yaml eksctl_Linux_amd64.tar.gz iam_policy.json index.html k8s kubectl vi
root@ip-172-31-5-101:/home/Brain-Tasks-App# git init
Reinitialized existing Git repository in /home/Brain-Tasks-App/.git/
root@ip-172-31-5-101:/home/Brain-Tasks-App# git add .
root@ip-172-31-5-101:/home/Brain-Tasks-App# git commit -m "second commit"
[main 4201236] second commit
Committer: root <root@ip-172-31-5-101.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

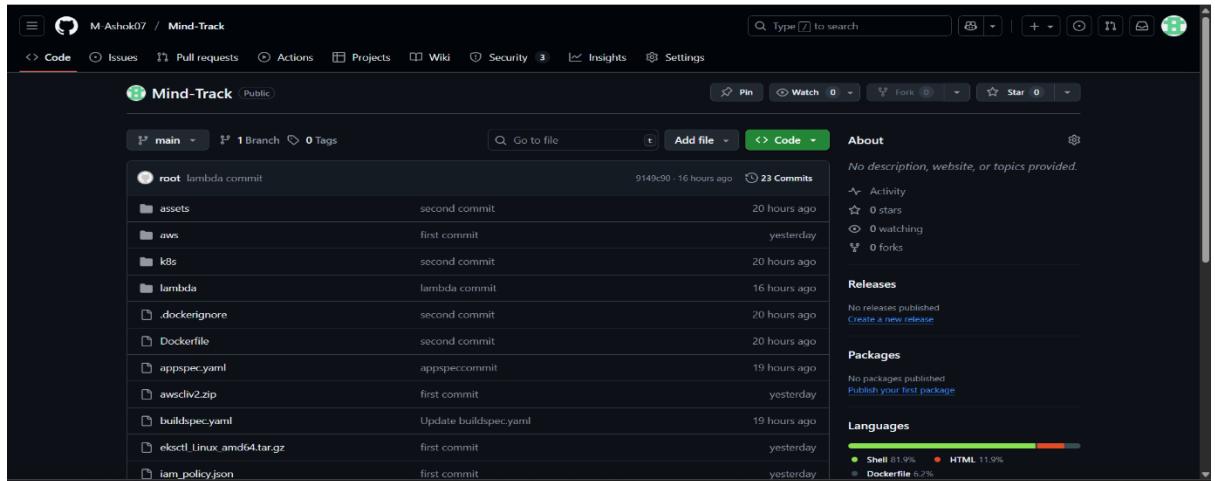
git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

9 files changed, 10 insertions(+), 3 deletions(-)
create mode 100644 .dockerignore
rename {dist/assets => assets}/index-BHGiHu50.js (100%)
rename {dist/assets => assets}/index-DPTLVrPB.css (100%)
rename dist/index.html => index.html (100%)
rename dist/vite.svg => vite.svg (100%)
root@ip-172-31-5-101:/home/Brain-Tasks-App# git branch -M main
root@ip-172-31-5-101:/home/Brain-Tasks-App# git push -u origin main
```

```
root@ip-172-31-5-101:/home/Brain-Tasks-App
root@ip-172-31-5-101:/home/Brain-Tasks-App# git pull origin main --rebase
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 27 (delta 16), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (27/27), 9.07 KiB | 1.51 MiB/s, done.
From https://github.com/M-Ashok07/Mind-Track
 * branch    main      -> FETCH_HEAD
 3eb5628..16147e1 main      -> origin/main
Successfully rebased and updated refs/heads/main.
root@ip-172-31-5-101:/home/Brain-Tasks-App# git push -u origin main
Username for 'https://github.com': M-Ashok07
Password for 'https://M-Ashok07@github.com':
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 2 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 112.80 KiB | 56.40 MiB/s, done.
Total 13 (delta 4), reused 5 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File kubect is 57.34 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/M-Ashok07/Mind-Track.git
 16147e1..294ffel main -> main
branch 'main' set up to track 'origin/main'.
root@ip-172-31-5-101:/home/Brain-Tasks-App#
```



CodeDeploy:

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

The image consists of three vertically stacked screenshots from the AWS CodeDeploy console.

Screenshot 1: Application Configuration

This screenshot shows the "Create application" page under "Application configuration". The "Application name" field contains "code-deploy". The "Compute platform" dropdown is set to "EC2/On-premises". A "Tags" section with an "Add tag" button is also visible. At the bottom are "Cancel" and "Create application" buttons.

Screenshot 2: Deployment Created

This screenshot shows a success message "Deployment created" for deployment ID "d-AS9V1W2HD". It displays deployment details: Application "code-deploy", Deployment ID "d-AS9V1W2HD", Deployment configuration "CodeDeployDefault.AllAtOnce", Deployment group "code-deploy-group", and Status "Created" (Initiated by User action). Below this, revision details show a revision location "s3://mind-track-123-s/my_.project.zip", revision created "Just now", and revision description "Application revision registered by Deployment ID: d-AS9V1W2HD".

Screenshot 3: Deployment Group Configuration

This screenshot shows the configuration for a new deployment group. It includes fields for "Deployment group name" ("code-deploy-group"), "Service role" ("arn:aws:iam::304534110140:role/Codedeployeksandcr"), and "Deployment type". Under "Deployment type", the "In-place" option is selected, with a note explaining it updates instances in the deployment group with the latest application revisions. The "Blue/green" option is also listed.

Deployment group name

Enter a deployment group name
code-deploy-group
100 character limit

Service role

Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.
arn:aws:iam::304534110140:role/Codedeployeksandecr

Deployment type

Choose how to deploy your application

- In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update
- Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the

Success Deployment group created

Developer Tools > CodeDeploy > Applications > code-deploy > code-deploy-group

code-deploy-group

Deployment group details

Deployment group name	Application name	Compute platform
code-deploy-group	code-deploy	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::304534110140:role/Codedeployeksandecr	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

Environment configuration: Amazon EC2 Auto Scaling groups

Name

Success Deployment created

Developer Tools > CodeDeploy > Deployments > d-AS9V1W2HD

d-AS9V1W2HD

Deployment details

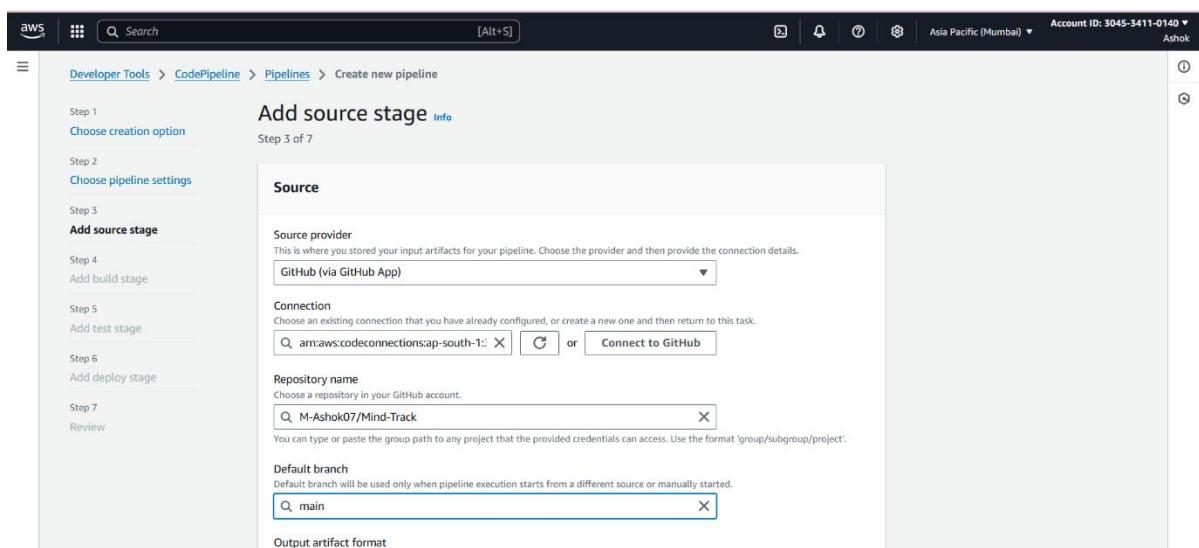
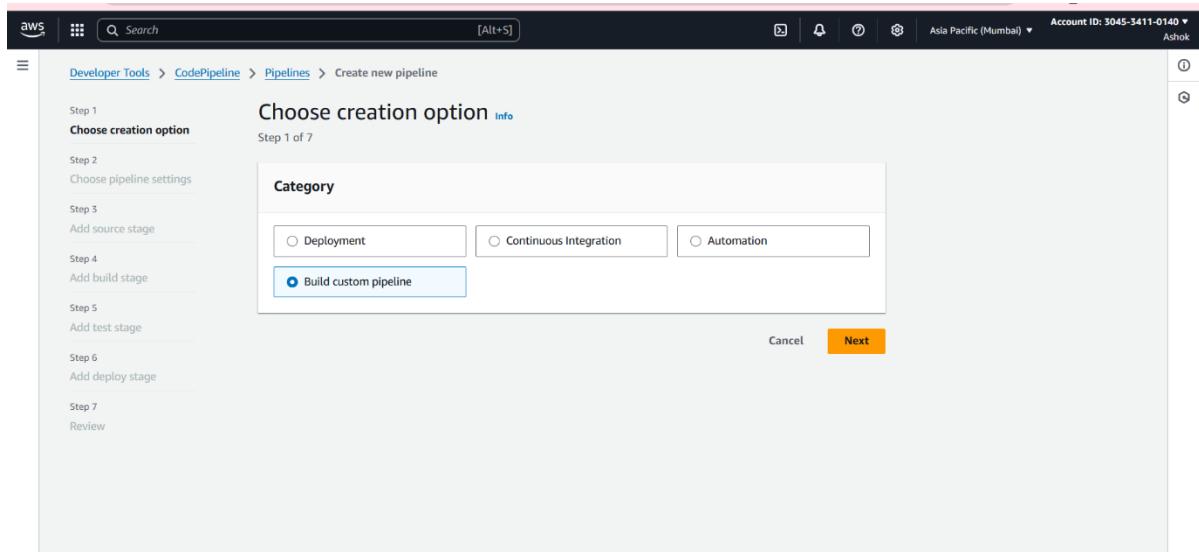
Application	Deployment ID	Status
code-deploy	d-AS9V1W2HD	Created
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.AllAtOnce	code-deploy-group	User action
Deployment description	-	

Revision details

Revision location	Revision created	Revision description
s3://mind-track-123-s/my_project.zip	Just now	Application revision registered by Deployment ID: d-AS9V1W2HD

CodePipeline:

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



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose creation option

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

Add build stage Info

Step 4 of 7

Build - optional

Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.

Commands Other build providers

AWS CodeBuild

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

brain-tasks-build X or Create project

Define buildspec override - optional
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

Environment variables - optional
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Add environment variable

The screenshot shows the AWS CodePipeline 'Create new pipeline' interface. The left sidebar lists steps from Step 1 to Step 7, with 'Add deploy stage' currently selected. The main area is titled 'Add deploy stage' with an 'Info' link. It shows 'Step 6 of 7'. The configuration section is titled 'Deploy - optional'. It includes fields for 'Deploy provider' (set to 'Amazon S3'), 'Region' (set to 'Asia Pacific (Mumbai)'), and 'Input artifacts'. A dropdown menu for 'Input artifacts' contains 'BuildArtifact' (Defined by: Build). Below this is a note: 'No more than 100 characters'. The 'Bucket' field contains 'mind-track-123-s'. At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'. The top right corner shows the account ID '3045-3411-0140', region 'Asia Pacific (Mumbai)', and a user icon.

The screenshot shows the AWS CodePipeline execution details page. The top navigation bar includes the AWS logo, a search bar, and account information (Account ID: 3045-3411-0140, Region: Asia Pacific (Mumbai)). A prominent blue banner at the top left introduces the new pipeline experience, stating: "Introducing the new pipeline experience. We've redesigned the pipeline view to streamline the monitoring and debugging experience. Let us know what you think. Allow cookies to return to the old experience." Below the banner, the breadcrumb trail shows: Developer Tools > CodePipeline > Pipelines > pipeline-code > d0979ab7. The main title is "Pipeline execution: d0979ab7". To the right are buttons for "Rerun", "Stop execution", "Previous execution", and "Next execution". The execution details are presented in three stages: "Source", "Build", and "Deploy". Each stage has a green checkmark icon and a status message: "All actions succeeded." The "Source" stage shows a GitHub action named "GitHub (via GitHub App)" that ran 7 minutes ago. The "Build" stage shows an AWS CodeBuild action that ran 8 minutes ago. The "Deploy" stage shows an Amazon S3 action that ran 1 minute ago. At the bottom, there are tabs for "Summary" (selected), "Timeline", "Variables", "Revisions", and "Stage".

The screenshot shows the AWS CodePipeline console with a banner at the top stating "Introducing the new pipeline experience". Below the banner, the pipeline "pipeline-code" is displayed. The "Executions" tab is selected, showing a single execution row:

Execution ID	Status	Source revisions	Trigger	Started	Duration	Completed
d0979ab7	Succeeded	Source - 9149c908 [lambda commit]	CreatePipeline - root	Aug 23, 2025 11:06 AM (UTC+5:30)	13 minutes 10 seconds	Aug 23, 2025 11:19 AM (UTC+5:30)

The screenshot shows the "Execution summary" page for the execution d0979ab7. It includes tabs for Summary, Timeline, Variables, Revisions, and Stage. The Summary tab displays the following details:

Status	Started	Completed	Duration
Succeeded	27 minutes ago	14 minutes ago	13 minutes 10 seconds

Trigger: CreatePipeline - root

Pipeline execution ID: d0979ab7-34a2-4729-b60c-e44a15b1c5bd

Latest action execution message: Deployment Succeeded

The screenshot shows the "Debug pipeline-code" view for the execution d0979ab7. On the left, a tree view shows the stages: Source (GitHub), Build (AWS CodeBuild), and Deploy (Amazon S3). The Deploy stage is currently selected. On the right, the "Pipeline execution details" section is expanded, showing the "Input" tab with the following configuration:

Action provider	Amazon S3
Variable namespace	DeployVariables
Input artifact	BuildArtifact
BucketName	mind-track-123-s
Extract	true

So I set up a pipeline using AWS that basically handles everything from pushing code to getting it live — no manual steps anymore.

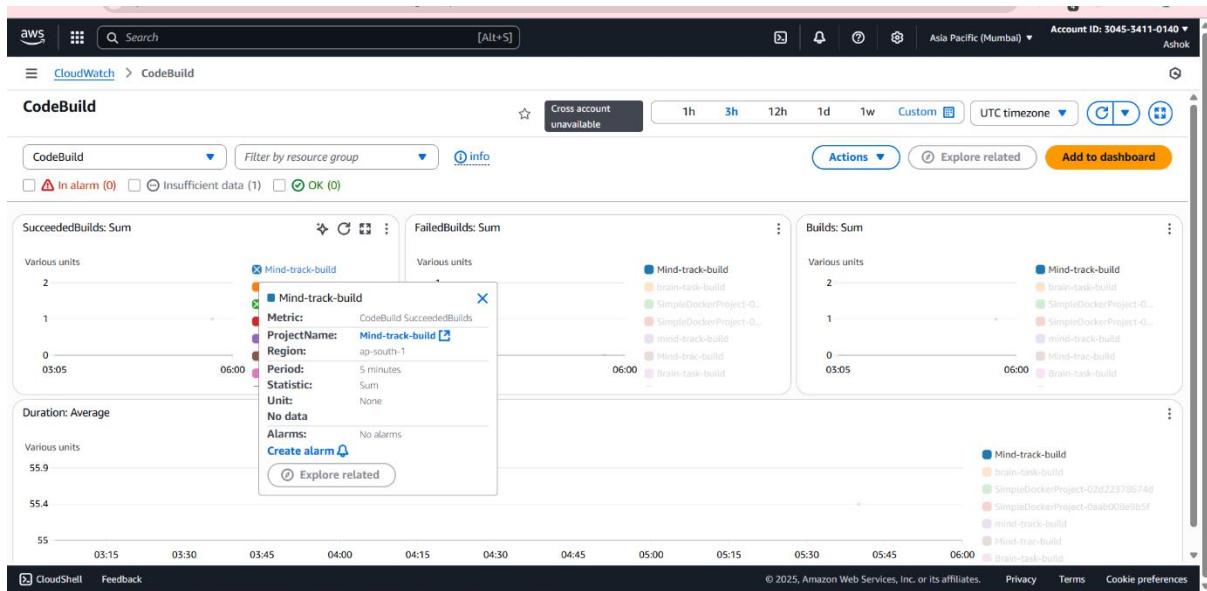
Whenever I push something to GitHub, AWS picks it up, builds it automatically, and uploads the final version straight to an S3 bucket. The whole process takes a few minutes and I don't have to

do anything. It grabs the latest code, builds it in the background, and once it's done, it pushes it live — super seamless.

In the screenshot, it just shows that everything worked: the code was pulled, built, and deployed successfully. Green checks across the board. It's nice because now every update I make just flows straight to production without me touching a thing.

Monitoring:

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



```
root@ip-172-31-5-101:~# kubectl logs -l app=brain-tasks -n default
192.168.18.216 -- [23/Aug/2025:06:26:39 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:26:43 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:44 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:54 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:58 +0000] "GET / HTTP/1.1" 200 603 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36" "+35.162.25.189"
192.168.64.142 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:27:09 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:27:13 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:27:14 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:29 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:26:43 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:44 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:45 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:27:09 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:27:13 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:27:14 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:29 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:26:43 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:44 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:26:45 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 -- [23/Aug/2025:06:27:09 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 -- [23/Aug/2025:06:27:13 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 -- [23/Aug/2025:06:27:14 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
root@ip-172-31-5-101:~#
```

```
192.168.64.142 - - [23/Aug/2025:06:26:43 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:26:44 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:26:54 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:26:56 +0000] "GET / HTTP/1.1" 200 603 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.91 Safari/537.36" "-"
192.168.64.142 - - [23/Aug/2025:06:26:58 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 - - [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:27:09 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:27:13 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 - - [23/Aug/2025:06:27:14 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 - - [23/Aug/2025:06:26:29 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:26:39 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:26:43 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 - - [23/Aug/2025:06:26:44 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:26:54 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:26:55 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:26:59 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.18.216 - - [23/Aug/2025:06:27:09 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.64.142 - - [23/Aug/2025:06:27:13 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "-"
192.168.61.105 - - [23/Aug/2025:06:27:14 +0000] "GET / HTTP/1.1" 200 603 "-" "ELB-HealthChecker/2.0" "
root@ip-172-31-5-101:~# helm repo update

# Create namespace for logging
kubectl create namespace amazon-cloudwatch || true

# Install aws-for-fluent-bit
helm upgrade --install aws-for-fluent-bit eks/aws-for-fluent-bit \
--namespace amazon-cloudwatch \
--set cloudWatchLogs.enabled=true \
--set cloudWatchLogs.logGroupName=eks/brain-tasks/app \
--set cloudWatchLogs.region=ap-south-1
"eks" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "eks" chart repository
Update Complete! Happy Helm-ing!
namespace/amazon-cloudwatch created
Release "aws-for-fluent-bit" does not exist. Installing it now.
NAME: aws-for-fluent-bit
LAST DEPLOYED: Sat Aug 23 06:28:21 2025
NAMESPACE: amazon-cloudwatch
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
aws-for-fluent-bit has been installed or updated. To check the status of pods, run:

kubectl get pods -n amazon-cloudwatch
root@ip-172-31-5-101:~#
```

```
kubectl get pods -n amazon-cloudwatch
root@ip-172-31-5-101:~# ^C
root@ip-172-31-5-101:~# kubectl get pods -n amazon-cloudwatch
NAME           READY   STATUS    RESTARTS   AGE
aws-for-fluent-bit-fvrpn  1/1     Running   0          14s
aws-for-fluent-bit-g8ng2  1/1     Running   0          14s
aws-for-fluent-bit-lnhrl  1/1     Running   0          14s
aws-for-fluent-bit-nkxqc  1/1     Running   0          14s
aws-for-fluent-bit-rlvlw  1/1     Running   0          14s
aws-for-fluent-bit-w887r  1/1     Running   0          14s
root@ip-172-31-5-101:~#
```

```

root@ip-172-31-5-101:~# aws logs describe-log-groups --log-group-name-prefix "eks/brain-tasks/app" --region ap-south-1
{
  "logGroups": [
    {
      "logGroupName": "/eks/brain-tasks/app",
      "creationTime": 1755930801377,
      "metricFilterCount": 0,
      "arn": "arn:aws:logs:ap-south-1:304534110140:log-group:/eks/brain-tasks/app::",
      "storedBytes": 0,
      "logGroupClass": "STANDARD",
      "logGroupArn": "arn:aws:logs:ap-south-1:304534110140:log-group:/eks/brain-tasks/app"
    }
  ]
}
root@ip-172-31-5-101:~# ^C
root@ip-172-31-5-101:~# aws logs describe-log-streams \
--log-group-name "/eks/brain-tasks/app" \
--region ap-south-1 \
--order-by LastEventTime \
--descending \
--limit 5
{
  "logStreams": [
    {
      "logStreamName": "fluentbit-kube.var.log.containers.aws-for-fluent-bit-wjtt_amazon-cloudwatch_aws-for-fluent-bit-6c94a5d71f2f667fa29cca35da0127ba2d47bf9fb6c80b6eceea0731ea5f69
5.log",
      "creationTime": 175593083518,
      "firstEventTimestamp": 1755930835040,
      "lastEventTimestamp": 1755930835111,
      "lastIngestionTime": 1755930835627,
      "uploadSequenceToken": "#03985964174086911072251565763979011559726090766652177",
      "arn": "arn:aws:logs:ap-south-1:304534110140:log-group:/eks/brain-tasks/app:log-stream:fluentbit-kube.var.log.containers.aws-for-fluent-bit-wjtt_amazon-cloudwatch_aws-for-fluen
t-bit-6c94a5d71f2f667fa29cca35da0127ba2d47bf9fb6c80b6eceea0731ea5f69c.log",
      "storedBytes": 0
    },
    {
      "logStreamName": "fluentbit-kube.var.log.containers.aws-for-fluent-bit-js768_amazon-cloudwatch_aws-for-fluent-bit-fe32dd8c6405cc07e82f5731af1f4c2d1b4a64bcc96009618805be155ae99da
5.log",
      "creationTime": 1755930833366,
      "firstEventTimestamp": 1755930832353,
      "lastEventTimestamp": 1755930832440,
      "lastIngestionTime": 1755930833373,
      "arn": "arn:aws:logs:ap-south-1:304534110140:log-group:/eks/brain-tasks/app:log-stream:fluentbit-kube.var.log.containers.aws-for-fluent-bit-js768_amazon-cloudwatch_aws-for-fluen
t-bit-fe32dd8c6405cc07e82f5731af1f4c2d1b4a64bcc96009618805be155ae99da5.log"
    }
  ]
}

```

Screenshot of the AWS CloudWatch Log events interface.

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

Actions | Start tailing | Create metric filter

Timestamp | Message

- 2025-08-23T06:33:55.074Z {"time": "2025-08-23T06:33:55.0745362382", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [http_server] listen ifa...}
- 2025-08-23T06:33:55.074Z {"time": "2025-08-23T06:33:55.0746224912", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [sp] stream processor st...}
- 2025-08-23T06:33:55.098Z {"time": "2025-08-23T06:33:55.0984677812", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [input:tail:tail1.0] inot...}
- 2025-08-23T06:33:55.098Z {"time": "2025-08-23T06:33:55.0988565972", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [input:tail:tail1.0] inot...}
- 2025-08-23T06:33:55.098Z {"time": "2025-08-23T06:33:55.0986209462", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [input:tail:tail1.0] inot...}
- 2025-08-23T06:33:55.111Z {"time": "2025-08-23T06:33:55.1115378142", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [input:tail:tail1.0] inot...}
- 2025-08-23T06:33:55.111Z {"time": "2025-08-23T06:33:55.1115605122", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [input:tail:tail1.0] inot...}
- 2025-08-23T06:33:55.792Z {"time": "2025-08-23T06:33:55.7920825262", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [output:cloudwatch_logs:...]
- 2025-08-23T06:33:55.794Z {"time": "2025-08-23T06:33:55.7946448872", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [output:cloudwatch_logs:...]
- 2025-08-23T06:33:55.817Z {"time": "2025-08-23T06:33:55.8172491412", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [output:cloudwatch_logs:...]
- 2025-08-23T06:33:55.823Z {"time": "2025-08-23T06:33:55.8238214022", "stream": "stderr", "_p": "F", "log": "[2025/08/23 06:33:55] [info] [output:cloudwatch_logs:...]

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

CloudShell Feedback

31°C

Screenshot of the AWS CloudWatch Log groups interface.

Log groups (12)

By default, we only load up to 10000 log groups.

Filter log groups or try pattern search | Exact match

Log group	Log class	Anomaly d...	Data pr...	Sensitiv...	Retention	Metric f...
/aws/codebuild/MySampleAppBuild	Standard	Configure	-	-	Never expire	-
/aws/codebuild/SimpleDockerProject-02d22378674d	Standard	Configure	-	-	Never expire	-
/aws/codebuild/SimpleDockerProject-0aab008e9b5f	Standard	Configure	-	-	Never expire	-
/aws/codebuild/brain-task-build	Standard	Configure	-	-	Never expire	-
/aws/codebuild/build-test	Standard	Configure	-	-	Never expire	-
/aws/codebuild/buildproject	Standard	Configure	-	-	Never expire	-
/aws/codebuild/hotel-wesite	Standard	Configure	-	-	Never expire	-
/aws/codebuild/mind-track-build	Standard	Configure	-	-	Never expire	-
/aws/codepipeline/brain-task-build	Standard	Configure	-	-	Never expire	-
/aws/codepipeline/pipeline-EKS	Standard	Configure	-	-	Never expire	-
/aws/lambda/eks-deolover	Standard	Configure	-	-	Never expire	-

CloudShell Feedback

Screenshot of the AWS Amazon S3 Buckets interface.

codepipeline-ap-south-1-268231a57f40-4248-9602-5a85ee2ee03e

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

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
pipeline-code/	Folder	-	-	-

Actions | Create folder | Upload

CloudShell Feedback

Amazon S3 > Buckets > codepipeline-ap-south-1-268231a57f40-4248-9602-5a85ee2ee03e > pipeline-code/

pipeline-code/

Objects Properties

Objects (2)

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
BuildArtif/	Folder	-	-	-
SourceArti/	Folder	-	-	-

Actions Create folder Upload

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 12:22

M-Ashok07 / Mind-Track

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Mind-Track Public

main 1 Branch 0 Tags Go to file Add file Code

root: lambda commit 9149:90 - 16 hours ago 23 Commits

File	Commit Message	Time
assets	second commit	20 hours ago
aws	first commit	yesterday
k8s	second commit	20 hours ago
lambda	lambda commit	16 hours ago
.dockerignore	second commit	20 hours ago
Dockerfile	second commit	20 hours ago
appspec.yaml	appspeccommit	19 hours ago
awscliv2.zip	first commit	yesterday
buildspec.yaml	Update buildspec.yaml	19 hours ago
ekscctl_Linux_amd64.tar.gz	first commit	yesterday
iam_policy.json	first commit	yesterday

About No description, website, or topics provided.

Activity 0 stars 0 watching 0 forks

Releases No releases published [Create a new release](#)

Packages No packages published [Publish your first package](#)

Languages Shell 81.9% HTML 11.9% Dockerfile 5.2%

