

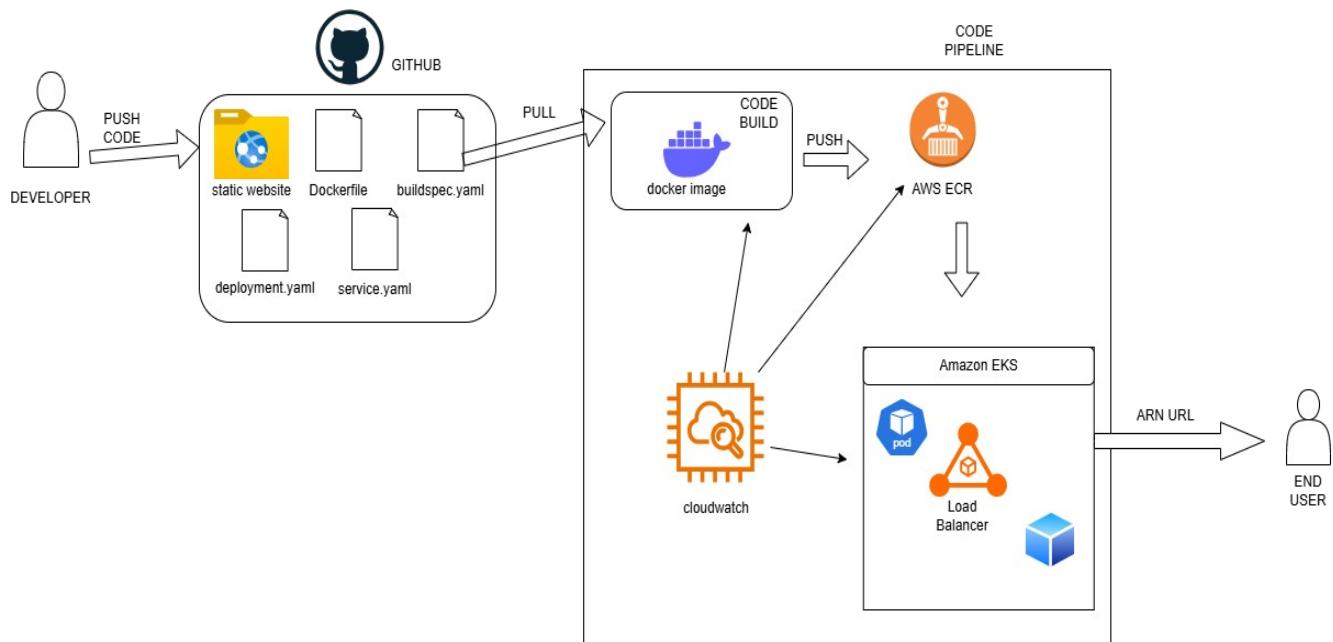
Application Deployment

OVERVIEW OF THE PROJECT

This project demonstrates an **end-to-end CI/CD pipeline** for deploying a **static web application** on **AWS EKS (Elastic Kubernetes Service)** using **GitHub, AWS CodePipeline, CodeBuild, Docker, ECR, and CloudWatch**.

The objective of the project is to achieve **fully automated application deployment** where **any code change pushed to GitHub automatically triggers build, containerization, image storage, and deployment to Kubernetes**, without any manual intervention.

PROJECT ARCHITECTURE



REQUIRED RESOURCES

Hardware	Software	Essential softwares
i3 system with 8gb ram laptop (Window 11 OS)	Install virtualbox (Linuxmint 22.2 xfce os installed for ubuntu environment)	git docker Aws cli kubectl eksctl
		Required Services
AWS Free Tier Account		ECR EKS CodeBuild CodeDeploy IAM Code pipeline Cloudwatch

Technologies Used

- **Version Control:** GitHub
- **CI/CD Orchestration:** AWS CodePipeline
- **Build Automation:** AWS CodeBuild
- **Containerization:** Docker
- **Container Registry:** Amazon ECR
- **Orchestration Platform:** Amazon EKS (Kubernetes)
- **Monitoring & Logs:** Amazon CloudWatch
- **Infrastructure Management:** eksctl, kubectl

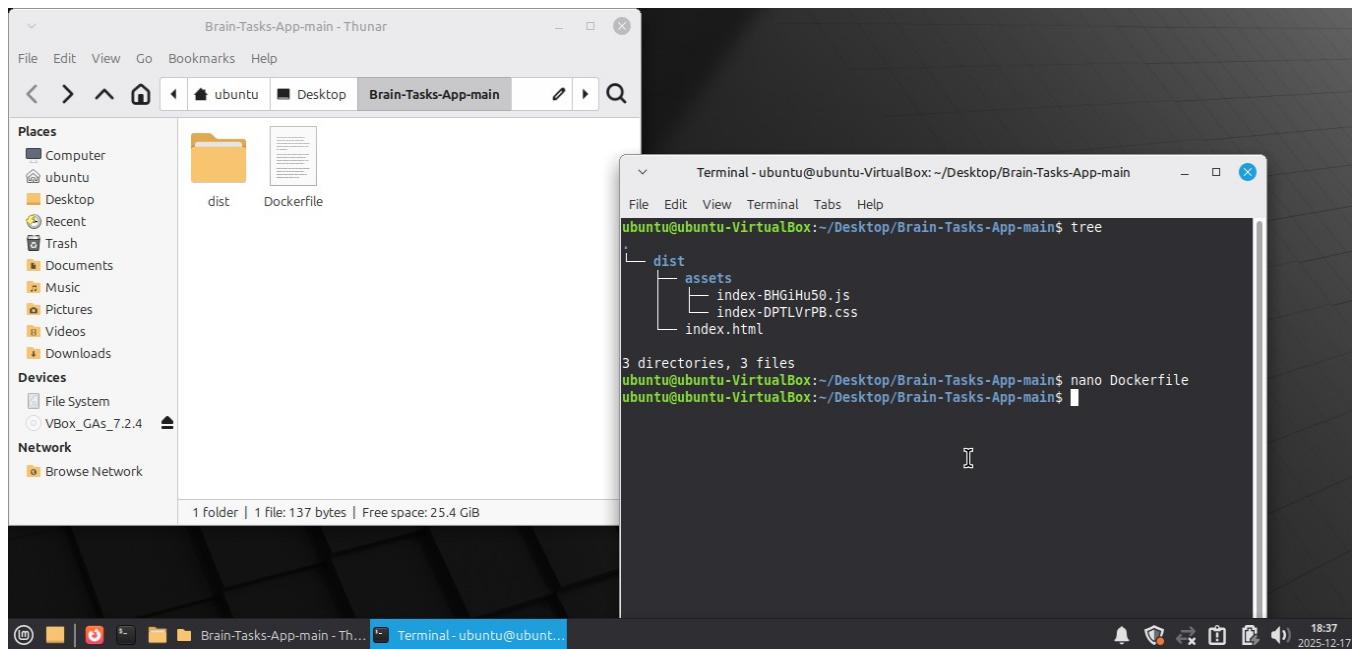
CI/CD Workflow

- Developer pushes code to GitHub.
- GitHub triggers by AWS CodePipeline.
- CodePipeline starts the CodeBuild project.
- CodeBuild:
 - Builds Docker image
 - Pushes image to Amazon ECR
 - Executes `kubectl` to deploy to EKS
- Kubernetes performs a rolling update.
- Application is exposed via an EKS LoadBalancer service.
- Logs and deployment status are monitored in CloudWatch.

Repository Structure

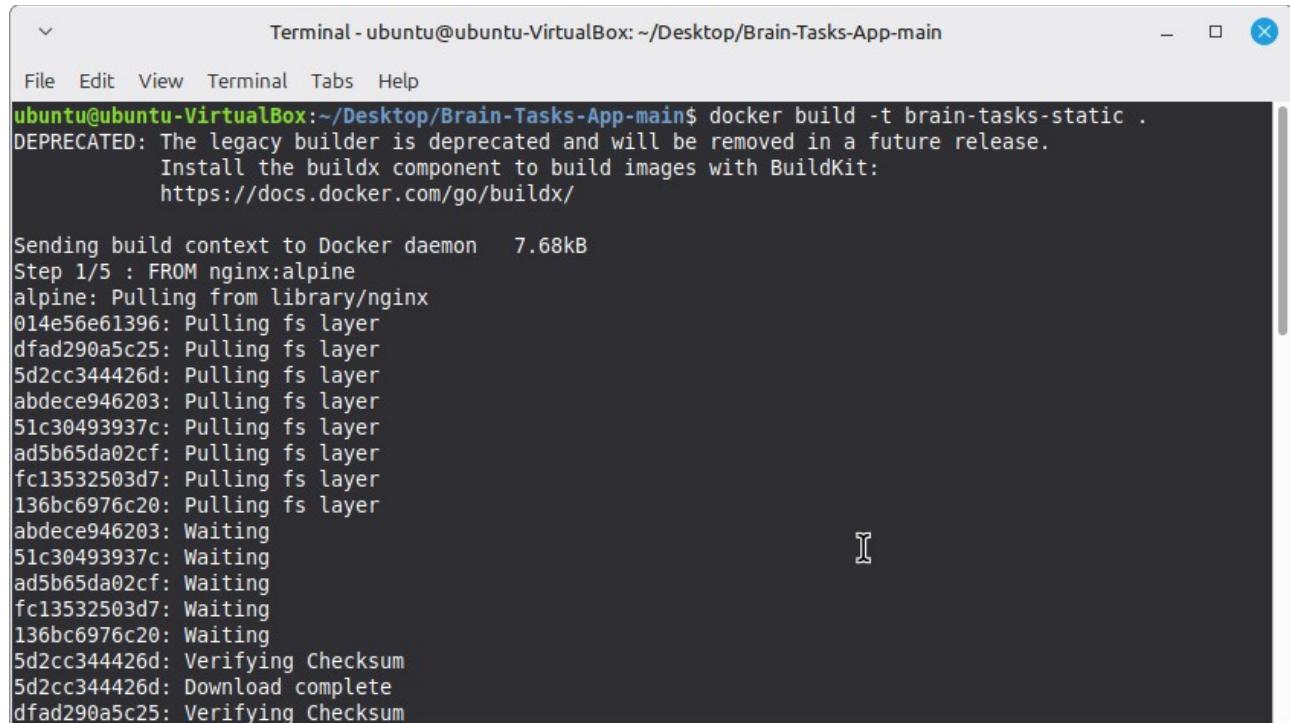
```
Brain-Tasks-App/
|
+-- dist/
|   +-- assets/
|   |   |-- index-BHGiHu50.js
|   |   |-- index-DPTLVrPB.css
|   +-- index.html
|
+-- k8s/
|   +-- deployment.yaml
|   +-- service.yaml
|
+-- Dockerfile
+-- buildspec.yml
+-- README.md
```

MANUAL TEST OF APPLICATION ON LOCAL SYSTEM



Created Dockerfile

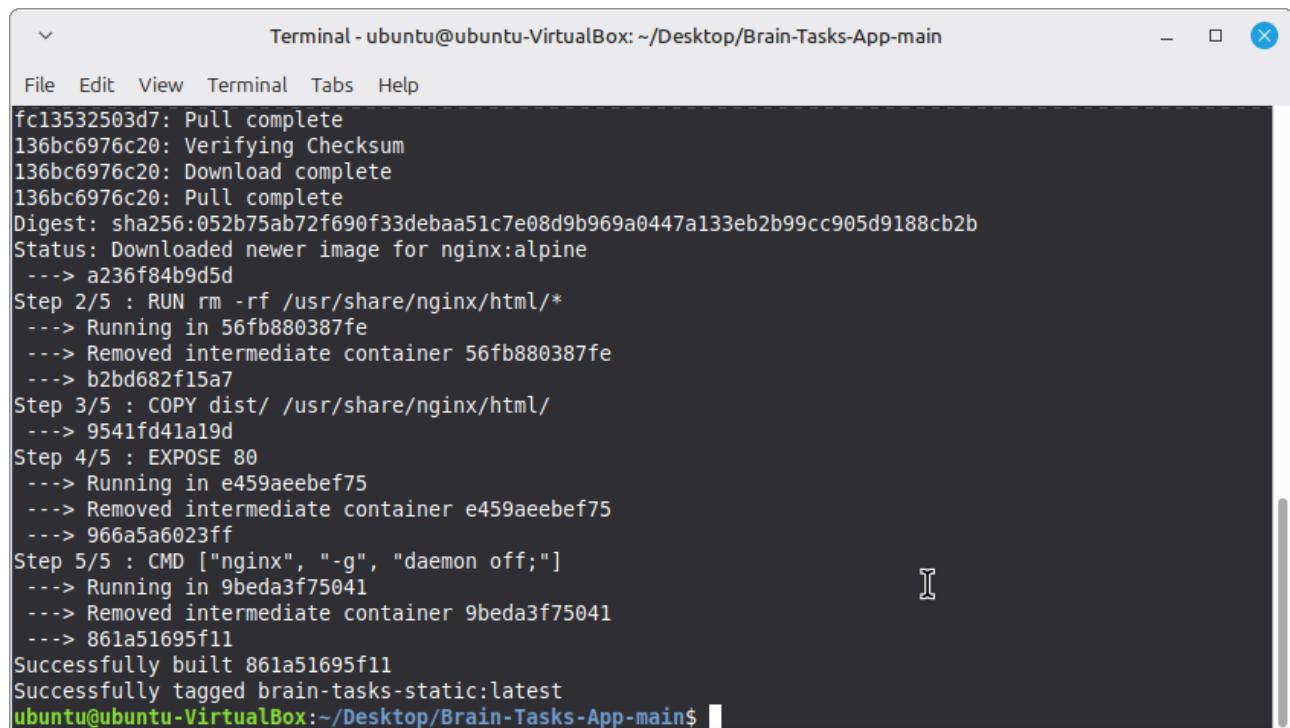
```
FROM nginx:alpine
RUN rm -rf /usr/share/nginx/html/*
COPY dist/ /usr/share/nginx/html/
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```



A terminal window titled "Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main". The window shows the command "docker build -t brain-tasks-static ." being run. It outputs a warning about the deprecated legacy builder and instructions to use buildx. Then it shows the step-by-step build process, including pulling layers for the nginx:alpine image and creating the final image.

```
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ docker build -t brain-tasks-static .
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/

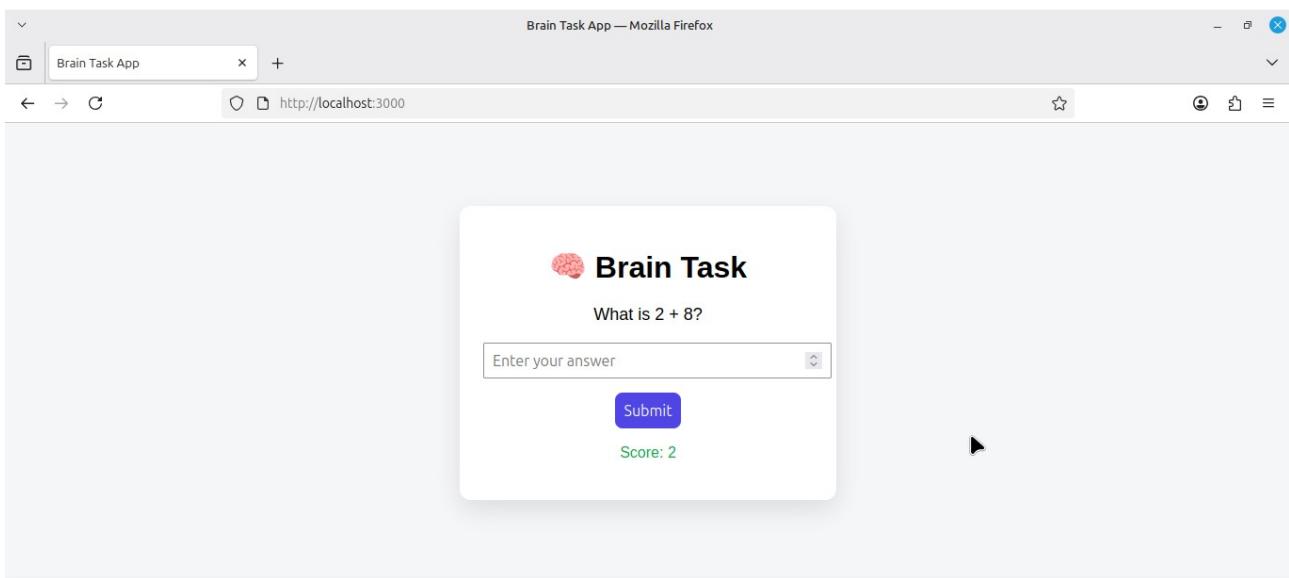
Sending build context to Docker daemon    7.68kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
014e56e61396: Pulling fs layer
dfad290a5c25: Pulling fs layer
5d2cc344426d: Pulling fs layer
abdece946203: Pulling fs layer
51c30493937c: Pulling fs layer
ad5b65da02cf: Pulling fs layer
fc13532503d7: Pulling fs layer
136bc6976c20: Pulling fs layer
abdece946203: Waiting
51c30493937c: Waiting
ad5b65da02cf: Waiting
fc13532503d7: Waiting
136bc6976c20: Waiting
5d2cc344426d: Verifying Checksum
5d2cc344426d: Download complete
dfad290a5c25: Verifying Checksum
```



A terminal window titled "Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main". The window shows the command "docker build -t brain-tasks-static ." being run again. This time, it shows the entire process from pulling the base image to copying the application files and exposing port 80, finally resulting in a successfully tagged image.

```
fc13532503d7: Pull complete
136bc6976c20: Verifying Checksum
136bc6976c20: Download complete
136bc6976c20: Pull complete
Digest: sha256:052b75ab72f690f33debaa51c7e08d9b969a0447a133eb2b99cc905d9188cb2b
Status: Downloaded newer image for nginx:alpine
---> a236f84b9d5d
Step 2/5 : RUN rm -rf /usr/share/nginx/html/*
---> Running in 56fb880387fe
---> Removed intermediate container 56fb880387fe
---> b2bd682f15a7
Step 3/5 : COPY dist/ /usr/share/nginx/html/
---> 9541fd41a19d
Step 4/5 : EXPOSE 80
---> Running in e459aeebef75
---> Removed intermediate container e459aeebef75
---> 966a5a6023ff
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
---> Running in 9beda3f75041
---> Removed intermediate container 9beda3f75041
---> 861a51695f11
Successfully built 861a51695f11
Successfully tagged brain-tasks-static:latest
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$
```

```
Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main
File Edit View Terminal Tabs Help
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ docker run -p 3000:80 brain-tasks-static
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/12/17 13:12:55 [notice] 1#1: using the "epoll" event method
2025/12/17 13:12:55 [notice] 1#1: nginx/1.29.4
2025/12/17 13:12:55 [notice] 1#1: built by gcc 15.2.0 (Alpine 15.2.0)
2025/12/17 13:12:55 [notice] 1#1: OS: Linux 6.14.0-29-generic
2025/12/17 13:12:55 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/12/17 13:12:55 [notice] 1#1: start worker processes
2025/12/17 13:12:55 [notice] 1#1: start worker process 30
2025/12/17 13:12:55 [notice] 1#1: start worker process 31
172.17.0.1 - - [17/Dec/2025:13:14:15 +0000] "GET / HTTP/1.1" 200 653 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:142.0) Gecko/20100101 Firefox/142.0" "-"
172.17.0.1 - - [17/Dec/2025:13:14:16 +0000] "GET /assets/index-DPTLVrPB.css HTTP/1.1" 200 775 "http://localhost:3000/" "Mozilla/5.0 (X11; Linux x86_64; rv:142.0) Gecko/20100101 Firefox/142.0" "-"
172.17.0.1 - - [17/Dec/2025:13:14:16 +0000] "GET /assets/index-BHGiHu50.js HTTP/1.1" 200 851 "http://localhost:3000/" "Mozilla/5.0 (X11; Linux x86_64; rv:142.0) Gecko/20100101 Firefox/142.0" "-"
```



It is working on <http://localhost:3000> port

PUSHED THE CODE TO GITHUB REPOSITORY

```
Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main$ git add .
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git commit -m "Static site with Dockerfile"
Author identity unknown

*** Please tell me who you are.

Run
  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit -global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'ubuntu@ubuntu-VirtualBox.(none)')
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git config --global user.name "PrabuSR"
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git config --global user.email "prabusr1988@gmail.com"
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git commit -m "Static site with Dockerfile"
[master (root-commit) b1c94d8] Static site with Dockerfile
 4 files changed, 133 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 dist/assets/index-BHGiHu50.js
 create mode 100644 dist/assets/index-DPTLVrPB.css
 create mode 100644 dist/index.html
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ 

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Dockerfile
    dist/

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ 
```

```
Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main$ git pull origin main --allow-unrelated-histories --no-rebase
From https://github.com/prabuSR/Brain-Tasks-App
 * branch            main      -> FETCH_HEAD
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git push -u origin main
Username for 'https://github.com': prabuSR
Password for 'https://prabuSR@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.96 KiB | 668.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prabuSR/Brain-Tasks-App.git
 15a0a93..6e38ebf main -> main
branch 'main' set up to track 'origin/main'.
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ 
```

The screenshot shows a GitHub repository page for 'Brain-Tasks-App'. The repository is public and has 1 branch and 0 tags. The main file listed is 'README.md'. The repository has 3 commits, with the latest being a merge from 'main' by 'prabuSR' 1 minute ago. Other files listed include 'dist', 'Dockerfile', and 'README.md'. The 'About' section indicates no description, website, or topics are provided. The 'Activity' section shows 0 stars, 0 watching, and 0 forks.

CODE PIPELINE PROCESS STARTS

CREATE CODEBUILD YML FILE

The screenshot shows a Linux desktop environment with a dark theme. A Thunar file manager window is open, showing a folder named 'Brain-Tasks-App-main' containing four files: 'dist', 'buildspec.yml', 'Dockerfile', and 'README.md'. A terminal window is also open, showing the command 'nano buildspec.yml' being run in the directory '~/Desktop/Brain-Tasks-App-main'. The terminal window title is 'Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main'. The desktop bar at the bottom shows various icons and the date '2025-12-17'.

buildspec.yaml

version: 0.2

env:

variables:

```
AWS_DEFAULT_REGION: ap-south-1  
ECR_URI: 736296213120.dkr.ecr.ap-south-1.amazonaws.com  
IMAGE_REPO_NAME: brain-tasks-static  
IMAGE_TAG: latest  
CLUSTER_NAME: brain-eks
```

phases:

install:

commands:

```
- echo Installing kubectl  
- curl -LO "https://dl.k8s.io/release/v1.29.0/bin/linux/amd64/kubectl"  
- chmod +x kubectl  
- mv kubectl /usr/local/bin/
```

pre_build:

commands:

```
- echo Logging in to Amazon ECR  
- aws ecr get-login-password --region $AWS_DEFAULT_REGION |  
  docker login --username AWS --password-stdin $ECR_URI  
- echo Updating kubeconfig  
- aws eks update-kubeconfig --region $AWS_DEFAULT_REGION --name  
$CLUSTER_NAME
```

build:

commands:

```
- echo Building Docker image  
- docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .  
- docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $ECR_URI/$IMAGE_REPO_NAME:  
$IMAGE_TAG  
- docker push $ECR_URI/$IMAGE_REPO_NAME:$IMAGE_TAG
```

post_build:

commands:

```
- echo Restarting Kubernetes deployment  
- kubectl rollout restart deployment brain-tasks-deployment -n default  
- kubectl rollout status deployment brain-tasks-deployment -n default
```

```

Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main
File Edit View Terminal Tabs Help
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ nano buildspec.yml
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git add buildspec.yml
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git commit -m "Update buildspec with ECR URI
[main 2136ab7] Update buildspec with ECR URI
1 file changed, 30 insertions(+)
create mode 100644 buildspec.yml
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ git push
Username for 'https://github.com': prabuSR
Password for 'https://prabuSR@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 705 bytes | 705.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prabuSR/Brain-Tasks-App.git
  6e38ebf..2136ab7  main -> main
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ 

```

CREATE AWS ECR TO PULL GITHUB buildspec.yml and build dockerimage and push to ECR

The screenshot shows the AWS ECR homepage. At the top right, it displays the account ID: 7362-9621-3120 and the region: Asia Pacific (Mumbai). A user named PRABU S R is logged in. The main heading is "Amazon Elastic Container Registry". Below it, the sub-headline reads "Share and deploy container software, publicly or privately". A call-to-action button labeled "Create a repository" with an orange "Create" button is prominently displayed. A small description at the bottom left explains what ECR is: "Amazon Elastic Container Registry (ECR) is a fully managed container registry that makes it easy to store, manage, share, and deploy your container images and artifacts anywhere." A "Pricing (US)" link is located at the bottom right.

The screenshot shows the "Create private repository" wizard in the AWS ECR console. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 7362-9621-3120, Region: Asia Pacific (Mumbai), User: PRABU S R). The current path is "Amazon ECR > Private registry > Repositories > Create private repository". The main section is titled "Create private repository". It has two main sections: "General settings" and "Image tag settings". In "General settings", there is a "Repository name" field containing "736296213120.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-static". In "Image tag settings", there is a "Image tag mutability" section with two options: "Mutable" (selected) and "Immutable". The "Mutable" option is described as allowing image tags to be overwritten. The "Immutable" option is described as preventing image tags from being overwritten. A note at the bottom states: "Tags that match these filters will be immutable (can't be overwritten). Using wildcards (*) will match zero or more image tag characters." A "Pricing (US)" link is also visible on the right side of the page.

The screenshot shows the AWS ECR console. In the top navigation bar, the account ID is 7362-9621-3120, the region is Asia Pacific (Mumbai), and the user is PRABU S R. The left sidebar has sections for Amazon Elastic Container Service, Private registry (Repositories, Features & Settings), and Public registry (Repositories, Settings). Below these are links to ECR public gallery, Amazon ECS, and Amazon EKS. The main content area shows a success message: "Successfully created private repository, brain-tasks-static". The "Private repositories (1)" section lists the repository "brain-tasks-static" with details: URI 736296213120.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-static, Created at December 17, 2025, 19:06:53 (UTC+05.5), Tag immutability Mutable, and Encryption type AES-256. There are buttons for View push commands, Delete, Actions, and Create repository.

AWSCLI ,kubectl,eksctl installed and configured locally.

```
Terminal - ubuntu@ubuntu-VirtualBox:~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu-VirtualBox:~$ aws configure  
AWS Access Key ID [None]: AKIA2W3VMTKAAS27DTEL  
AWS Secret Access Key [None]: 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  
Default region name [None]: ap-south-1  
Default output format [None]: json  
ubuntu@ubuntu-VirtualBox:~$
```

```
Terminal - ubuntu@ubuntu-VirtualBox:~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu-VirtualBox:~$ curl -LO "https://dl.k8s.io/release/$(curl -Ls https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
% Total    % Received % Xferd  Average Speed   Time     Time      Current  
          Dload Upload Total   Spent   Left  Speed  
100  138  100  138    0     0   200      0  --:--:--  0:01:36  --:--:--  200  
100 57.7M  100 57.7M    0     0   615k      0  0:01:36  0:01:36  --:--:--  612k  
ubuntu@ubuntu-VirtualBox:~$ chmod +x kubectl  
ubuntu@ubuntu-VirtualBox:~$ sudo mv kubectl /usr/local/bin/  
[sudo] password for ubuntu:  
ubuntu@ubuntu-VirtualBox:~$ kubectl version --client  
Client Version: v1.34.3  
Kustomize Version: v5.7.1  
ubuntu@ubuntu-VirtualBox:~$
```

```
Terminal - ubuntu@ubuntu-VirtualBox:~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu-VirtualBox:~$ curl -L0 "https://dl.k8s.io/release/$(curl -Ls https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current  
                                         Dload Upload Total  Spent   Left  Speed  
100  138  100  138    0     0   200      0 --:--:-- --:--:-- --:--:--  200  
100 57.7M  100 57.7M    0     0  615k      0  0:01:36  0:01:36 --:--:-- 612k  
ubuntu@ubuntu-VirtualBox:~$ chmod +x kubectl  
ubuntu@ubuntu-VirtualBox:~$ sudo mv kubectl /usr/local/bin/  
[sudo] password for ubuntu:  
ubuntu@ubuntu-VirtualBox:~$ kubectl version --client  
Client Version: v1.34.3  
Kustomize Version: v5.7.1  
ubuntu@ubuntu-VirtualBox:~$ curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_Linux_amd64.tar.gz" | tar xz  
ubuntu@ubuntu-VirtualBox:~$ sudo mv eksctl /usr/local/bin/  
ubuntu@ubuntu-VirtualBox:~$ eksctl version  
0.220.0  
ubuntu@ubuntu-VirtualBox:~$
```

AWS EKS CLUSTER CREATION

```
Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main/k8s  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ eksctl create cluster \  
--name brain-eks \  
--region ap-south-1 \  
--nodegroup-name linux-nodes \  
--node-type t3.medium \  
--nodes 2  
2025-12-17 19:42:58 [i] eksctl version 0.220.0  
2025-12-17 19:42:58 [i] using region ap-south-1  
2025-12-17 19:43:00 [i] setting availability zones to [ap-south-1a ap-south-1c ap-south-1b]  
2025-12-17 19:43:00 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.96.0/19  
2025-12-17 19:43:00 [i] subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19  
2025-12-17 19:43:00 [i] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19  
2025-12-17 19:43:00 [i] nodegroup "linux-nodes" will use "" [AmazonLinux2023/1.32]  
2025-12-17 19:43:00 [!] 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/  
2025-12-17 19:43:00 [i] using Kubernetes version 1.32  
2025-12-17 19:43:00 [i] creating EKS cluster "brain-eks" in "ap-south-1" region with managed nodes  
2025-12-17 19:43:00 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup  
2025-12-17 19:43:00 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=brain-eks'  
2025-12-17 19:43:00 [i] Kubernetes API endpoint access will use default of {publicAccess=true, priva
```

```

Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main/k8s

File Edit View Terminal Tabs Help
2025-12-17 19:54:19 [i] building managed nodegroup stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:54:20 [i] deploying stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:54:20 [i] waiting for CloudFormation stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:54:50 [i] waiting for CloudFormation stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:55:35 [i] waiting for CloudFormation stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:56:47 [i] waiting for CloudFormation stack "eksctl-brain-eks-nodegroup-linux-nodes"
2025-12-17 19:56:47 [i] waiting for the control plane to become ready
2025-12-17 19:56:50 [✓] saved kubeconfig as "/home/ubuntu/.kube/config"
2025-12-17 19:56:50 [i] no tasks
2025-12-17 19:56:50 [✓] all EKS cluster resources for "brain-eks" have been created
2025-12-17 19:56:50 [i] nodegroup "linux-nodes" has 2 node(s)
2025-12-17 19:56:50 [i] node "ip-192-168-19-47.ap-south-1.compute.internal" is ready
2025-12-17 19:56:50 [i] node "ip-192-168-58-83.ap-south-1.compute.internal" is ready
2025-12-17 19:56:50 [i] waiting for at least 2 node(s) to become ready in "linux-nodes"
2025-12-17 19:56:50 [i] nodegroup "linux-nodes" has 2 node(s)
2025-12-17 19:56:50 [i] node "ip-192-168-19-47.ap-south-1.compute.internal" is ready
2025-12-17 19:56:50 [i] node "ip-192-168-58-83.ap-south-1.compute.internal" is ready
2025-12-17 19:56:50 [✓] created 1 managed nodegroup(s) in cluster "brain-eks"
2025-12-17 19:56:51 [i] creating addon: metrics-server
2025-12-17 19:56:51 [i] successfully created addon: metrics-server
2025-12-17 19:56:55 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2025-12-17 19:56:55 [✓] EKS cluster "brain-eks" in "ap-south-1" region is ready
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ █

```

```

Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main/k8s

File Edit View Terminal Tabs Help
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
ip-192-168-19-47.ap-south-1.compute.internal   Ready    <none>   3m18s   v1.32.9-eks-ecaa3a6
ip-192-168-58-83.ap-south-1.compute.internal   Ready    <none>   3m18s   v1.32.9-eks-ecaa3a6
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ █

```

The screenshot shows the AWS EKS console interface. On the left, there's a navigation sidebar with sections like 'Amazon Elastic Kubernetes Service', 'Clusters', 'Settings', 'Amazon EKS Anywhere', and 'Related services'. The main area is titled 'Clusters (1)' and shows a table with one row for the cluster 'brain-eks'. The table columns include 'Cluster name' (set to 'brain-eks'), 'Status' (set to 'Active'), 'Kubernetes version' (set to '1.32'), 'Support period' (set to 'Standard support until March 23, 2026'), 'Upgrade policy' (set to 'Extended support'), and 'Created' (set to '15 minut'). At the top right, there are buttons for 'Delete' and 'Create cluster'.

Amazon Elastic Kubernetes Service > Clusters > brain-eks

Node groups (1)

Node groups implement basic compute scaling through EC2 Auto Scaling groups.

Group name	Desired size	AMI release version	Launch template	Status
linux-nodes	2	1.32.9-20251209	eksctl-brain-eks-nodegroup-linux-nodes (1)	Active

Fargate profiles (0)

No Fargate profiles

This cluster does not have any Fargate profiles.

Add Fargate profile

CREATE deployment and service.yaml file.

k8s - Thunar

File Edit View Go Bookmarks Help

Places

- Computer
- ubuntu
- Desktop
- Recent
- Trash
- Documents
- Music
- Pictures
- Videos
- Downloads

Devices

- File System
- VBox_GAS_7.2.4

Network

- Browse Network

2 files: 622 bytes | Free space: 24.6 GiB

Terminal - ubuntu@ubuntu:~/Desktop/Brain-Tasks-App-main/k8s

File Edit View Terminal Tabs Help

```
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ nano deployment.yaml
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$ nano service.yaml
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main/k8s$
```

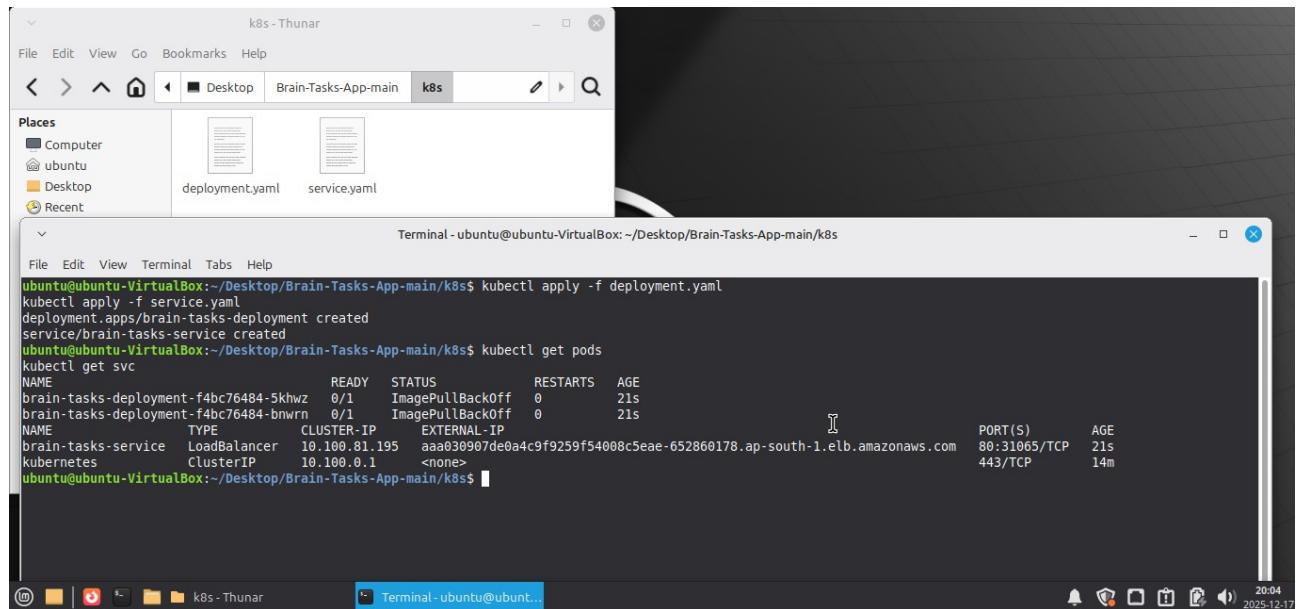
deployment.yaml to pull the code from ECR and Starts creating clusters

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-deployment
  labels:
    app: brain-tasks
spec:
  replicas: 2
  selector:
    matchLabels:
      app: brain-tasks
  template:
    metadata:
      labels:
        app: brain-tasks
    spec:
      containers:
        - name: brain-tasks
          image: 736296213120.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-static:latest
          ports:
            - containerPort: 80
```

service.yaml to create LOAD BALANCER and generate ARN URL

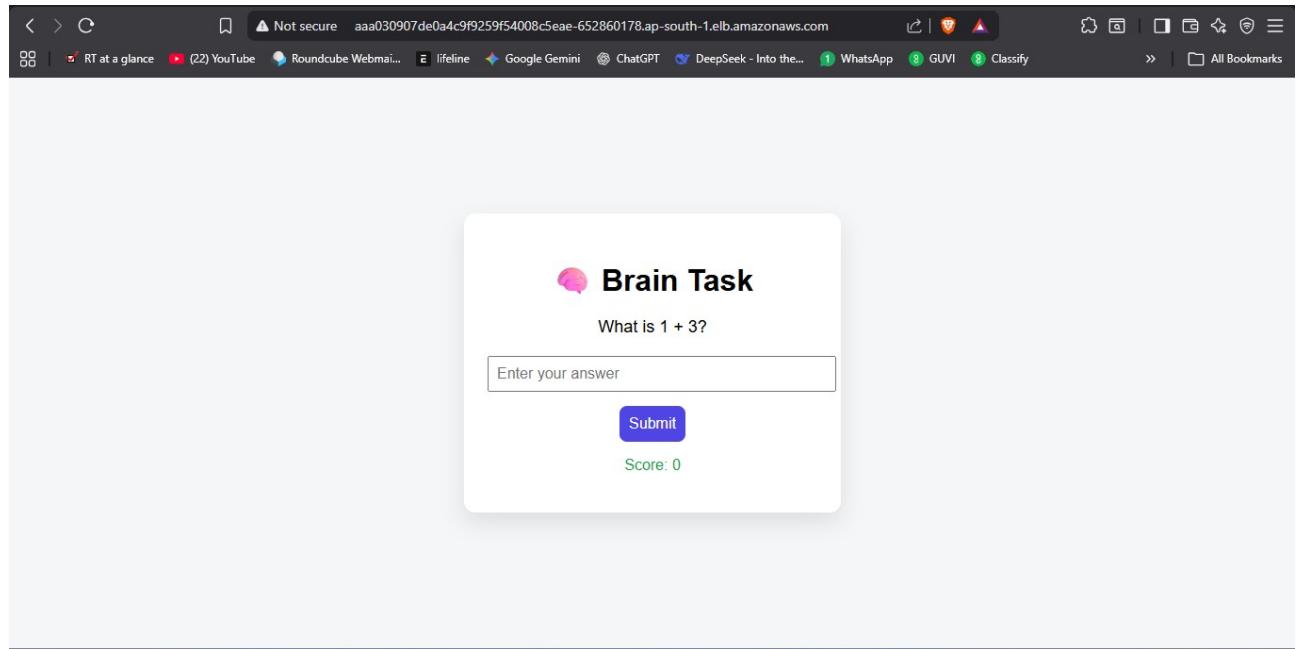
```
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
spec:
  type: LoadBalancer
  selector:
    app: brain-tasks
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

EKS MANUAL TESTING

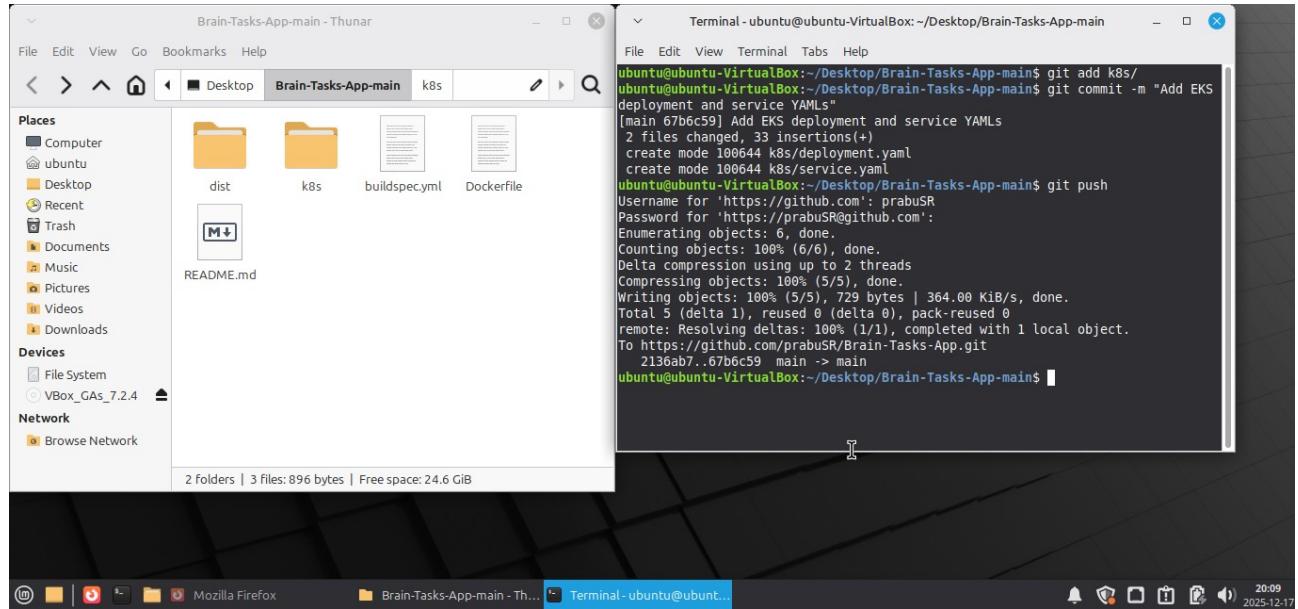


Error: ImagePullBackOff status show that nothing images in the ECR to pull

LOCALLY IT IS WORKING FINE



PUSHED TO GITHUB



The screenshot shows a GitHub repository page for 'Brain-Tasks-App' owned by 'prabuSR'. The repository is public and has 1 branch and 0 tags. The code tab is selected, showing the commit history:

- prabuSR Add EKS deployment and service YAMLS (commit 67b6c59, 2 minutes ago)
- dist Static site with Dockerfile (1 hour ago)
- k8s Add EKS deployment and service YAMLS (2 minutes ago)
- Dockerfile Static site with Dockerfile (1 hour ago)
- README.md readme (1 hour ago)
- buildspec.yml Update buildspec with ECR URI (1 hour ago)

The repository details sidebar includes sections for About (No description, website, or topics provided), Activity (0 stars, 0 forks, 0 watching), Releases (No releases published, Create a new release), and Packages.

CREATE AWS CODE BUILD in aws console to trigger buildspec.yml

The screenshot shows the AWS CodeBuild 'Build projects' page. On the left, there's a sidebar with 'CodeBuild' selected under 'Developer Tools'. The main area displays a table of build projects. One project, 'brain-tasks-build', is listed with details: Source provider is GitHub, Repository is 'prabuSR/Brain-Tasks-App', Latest build status is 'Succeeded', and it was modified 14 hours ago.

This screenshot shows the 'Create build project' wizard, step 1: Source. It asks for a source provider, which is set to GitHub. Below that, there's a section for credentials, indicating a successful connection to a GitHub App. There are options to use override credentials or a GitHub scoped webhook. The repository URL is set to 'https://github.com/prabuSR/Brain-Tasks-App'. A 'Source version - optional' field is present, though empty. An 'Additional configuration' section is at the bottom.

This screenshot shows the 'Create build project' wizard, step 2: Buildspec. It includes sections for 'Build specifications' (with options for 'Insert build commands' or 'Use a buildspec file'), 'Buildspec name - optional' (with a text input for 'buildspec.yml'), and 'Batch configuration' (with a note about running builds as a single execution). The 'Use a buildspec file' option is selected.

Screenshot of the AWS CodeBuild 'Create build project' configuration page.

Provisioning model

- 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

Amazon Linux

Screenshot of the AWS CodeBuild 'Create build project' configuration page, showing additional configuration options.

Service role

- New service role: Create a service role in your account.
- Existing service role: Choose an existing service role from your account.

Role name

codebuild-brain-tasks-build-service-role

Type your service role name

Additional configuration

Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

Buildspec

Build specifications

- Insert build commands: Store build commands as build project configuration.
- Use a buildspec file: Store build commands in a YAML-formatted buildspec file.

Screenshot of the AWS CodeBuild 'brain-tasks-build' project details page.

Project created

You have successfully created the following project: brain-tasks-build

[Create a notification rule for this project](#)

brain-tasks-build

Actions ▾ Create trigger Edit Clone Debug build Start build with overrides Start build

Configuration

Source provider	Primary repository	Artifacts upload location	Service role
GitHub	prabuSR/Brain-Tasks-App	-	arn:aws:iam::736296213120:role/service-role/codebuild-brain-tasks-build-service-role

Public builds

Disabled

Build history Batch history Project details Build triggers Metrics Debug sessions

CREATE IAM ROLE FOR CODEBUILD TO ACCESS ANOTHER ECR SERVICE.

The screenshot shows the AWS IAM Roles page. A green success message at the top states: "Policy was successfully attached to role." Below it, a table lists three policies attached to the role:

Policy name	Type	Attached entities
AmazonEC2ContainerRegistryPowerUser	AWS managed	1
CodeBuildBasePolicy-brain-tasks-build-ap-south-1	Customer managed	1
CodeBuildCodeConnectionsSourceCredentialsPolicy-br...	Customer managed	1

Below the table, sections for "Permissions boundary" (not set) and "Generate policy based on CloudTrail events" are visible. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, Console Mobile App, and copyright information.

EKS TEST AFTER ECR IMAGE PUSHED BY buildspec.yml

```
Terminal - ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main
File Edit View Terminal Tabs Help
Restart Count: 0
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-q66bg (ro)
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-q66bg:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    Optional:     false
    DownwardAPI:   true
  QoS Class:  BestEffort
  Node-Selectors: <none>
  Tolerations:   node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age           From            Message
  ----  ----   --           --            --
  Normal  BackOff  5m18s (x375 over 90m)  kubelet  Back-off pulling image "736296213120.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-static:latest"
  Warning Failed  5m18s (x375 over 90m)  kubelet  Error: ImagePullBackOff
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
brain-tasks-deployment-f4bc76484-5khwz  1/1     Running   0          99m
brain-tasks-deployment-f4bc76484-bnwrn  1/1     Running   0          99m
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$
```

Dockerimage is pushed to ECR automatically.

The screenshot shows a web browser window with a pop-up titled "Brain Task". The challenge asks: "What is 1 + 3?". A text input field is provided for the answer, and a blue "Submit" button is below it. The score is currently 0.

CODE DEPLOY (My Scenario)

The screenshot shows the AWS CodeDeploy Applications page. The left sidebar is titled "CodeDeploy" and includes links for Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy - Getting started, Deployments, Applications, Deployment configurations, On-premises instances), and Pipeline (CodePipeline). The main content area is titled "Applications" and shows a table with columns for Application name, Compute platform, and Created. A search bar is at the top, and buttons for Notify, View details, Deploy application, and Create application are available. The message "No results" and "There are no results to display." is centered.

The screenshot shows the "Create application" configuration page. It has a section for "Application configuration" with fields for "Application name" (containing "brain-tasks-codedeploy") and "Compute platform" (set to "EC2/On-premises"). Below this is a "Tags" section with a "Add tag" button. At the bottom are "Cancel" and "Create application" buttons.

As I am using AWS free tier account it is not possible to create Code Deploy in AWS, So I Planned to Trigger Code Deploy in buildspec.yaml file itself and triggered using code pipeline.

post_build:

commands:

- echo Restarting Kubernetes deployment
- kubectl rollout restart deployment brain-tasks-deployment -n default
- kubectl rollout status deployment brain-tasks-deployment -n default

This will triggers the eks of rolling update when codebuild yaml file is triggered.

After this update IAM role of codebuild need to be update as it trigger the eks service

The screenshot shows the AWS IAM Roles page. In the top navigation bar, there are several tabs like 'RT at a glance', 'YouTube', 'Roundcube Webmail', 'lifeline', 'Google Gemini', 'ChatGPT', 'DeepSeek - Into the...', 'WhatsApp', 'GUVI', and 'Classify'. The main content area shows a success message: 'Policies have been successfully attached to role.' Below this, a table lists policies attached to the role:

Policy name	Type	Attached entities
AmazonEC2ContainerRegistry...	AWS managed	1
AmazonEKS_CNI_Policy	AWS managed	4
AmazonEKSClusterPolicy	AWS managed	5
AmazonEKSWorkerNodePolicy	AWS managed	4
CodeBuildBasePolicy-brain-tasks...	Customer managed	1
CodeBuildCodeConnectionsSource...	Customer managed	1

At the bottom of the table, it says 'Permissions boundary (not set)'. The footer of the page includes links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

The screenshot shows a terminal window titled 'Terminal - ubuntu@ubuntu-VirtualBox: ~/Desktop/Brain-Tasks-App-main'. The terminal output is as follows:

```
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ eksctl create iamidentitymapping \
--cluster brain-eks \
--region ap-south-1 \
--arn arn:aws:iam::736296213120:role/codebuild-brain-tasks-build-service-role \
--group system:masters \
--username codebuild
2025-12-17 22:56:55 [i] checking arn arn:aws:iam::736296213120:role/codebuild-brain-tasks-build-service-role against entries in the auth ConfigMap
2025-12-17 22:56:55 [i] adding identity "arn:aws:iam::736296213120:role/codebuild-brain-tasks-build-service-role" to auth ConfigMap
ubuntu@ubuntu-VirtualBox:~/Desktop/Brain-Tasks-App-main$ █
```

CODE PIPELINE

The screenshot shows the AWS CodePipeline Pipelines page. At the top, there is a search bar and navigation links for developer tools. Below the header, the left sidebar lists stages: Source (CodeCommit), Artifacts (CodeArtifact), and Build (CodeBuild). The main area is titled "Pipelines" with a "Info" link. It features a search bar and a table with columns: Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. A message "No results" is displayed below the table.

The screenshot shows the "Choose creation option" step of the pipeline creation wizard. On the left, a sidebar lists steps from 1 to 7. Step 1 is "Choose creation option". The main area shows a "Category" section with three options: Deployment, Continuous Integration, and Automation. The "Build custom pipeline" option is selected. At the bottom right are "Cancel" and "Next" buttons.

The screenshot shows the "Build - optional" configuration step. The sidebar lists steps 1 through 7. Step 1 is "Choose pipeline settings". The main area shows a "Build provider" section where "Other build providers" is selected. A dropdown menu shows "AWS CodeBuild". The "Project name" field contains "brain-tasks-build". There is also a "Create project" button. Below this is a checkbox for "Define buildspec override - optional". The "Environment variables - optional" section is also visible at the bottom.

Installed GitHub App - AWS Connector for GitHub - Brave

github.com/settings/installations/87748500

Notifications

Permissions

- Read access to actions, deployments, environments, issues, and metadata
- Read and write access to administration, code, commit statuses, pull requests, and repository hooks

Repository access

All repositories
This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

Only select repositories
Select at least one repository. Also includes public repositories (read-only).

Select repositories ▾

Selected 1 repository:

prabuSR/Brain-Tasks-App

Save **Cancel**

aws | Search [Alt+S] Account ID: 7362-9621-3120 Asia Pacific (Mumbai) PRABU S R

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose creation option

Step 2 Choose pipeline settings

Step 3 Add source stage

Add source stage Info Step 3 of 7

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codeconnections:ap- X C or Connect to GitHub

Repository name
Choose a repository in your GitHub account.

prabuSR/Brain-Tasks-App

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

ap-south-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/brain-tasks-pipeline/view/regio... All Bookmarks

aws | Search [Alt+S] Account ID: 7362-9621-3120 Asia Pacific (Mumbai) PRABU S R

Developer Tools > CodePipeline > Pipelines > brain-tasks-pipeline

Success
Congratulations! The pipeline brain-tasks-pipeline has been created.

brain-tasks-pipeline Edit Stop execution Create trigger Clone pipeline Release change

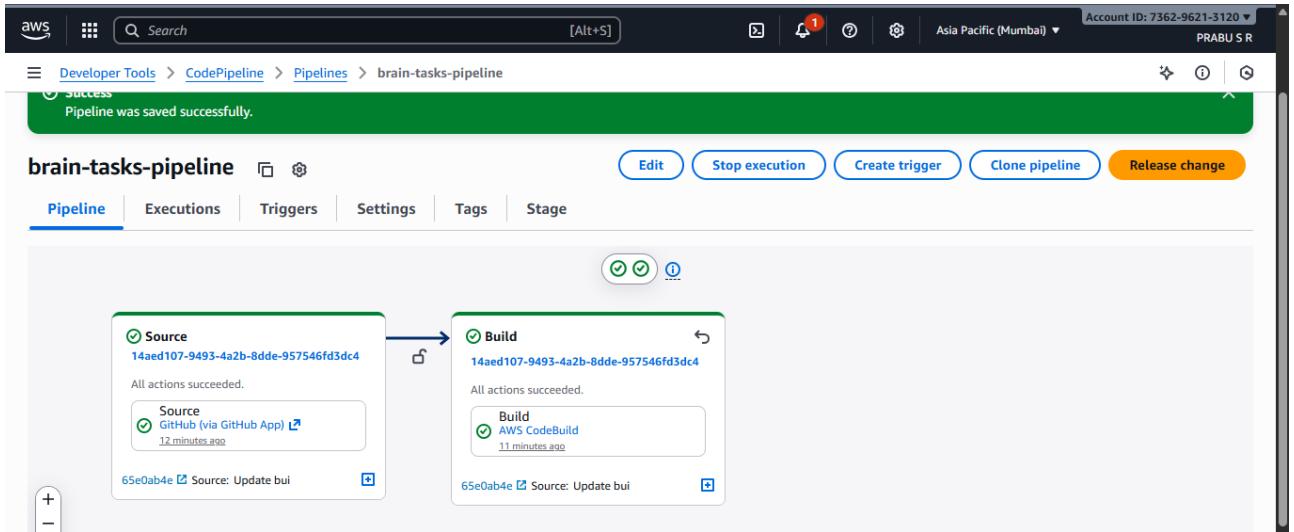
Pipeline Executions Triggers Settings Tags Stage

Source
44547b1c-5ed5-42c1-b22a-39677c4cd5e9
All actions succeeded.
Source GitHub (via GitHub App) Just now
65e0ab4e Source: Update bui

Build
44547b1c-5ed5-42c1-b22a-39677c4cd5e9
In progress: 1
Build AWS CodeBuild Just now
65e0ab4e Source: Update bui

Test
Didnt Run
Test AWS CodeBuild

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



CLOUDWATCH LOGS

CloudWatch

Favorites and recents

Container Insights

Database Insights

Lambda Insights

EC2 Resource Health

Logs

Log Management New

Log Anomalies

Live Tail

Logs Insights New

Contributor Insights

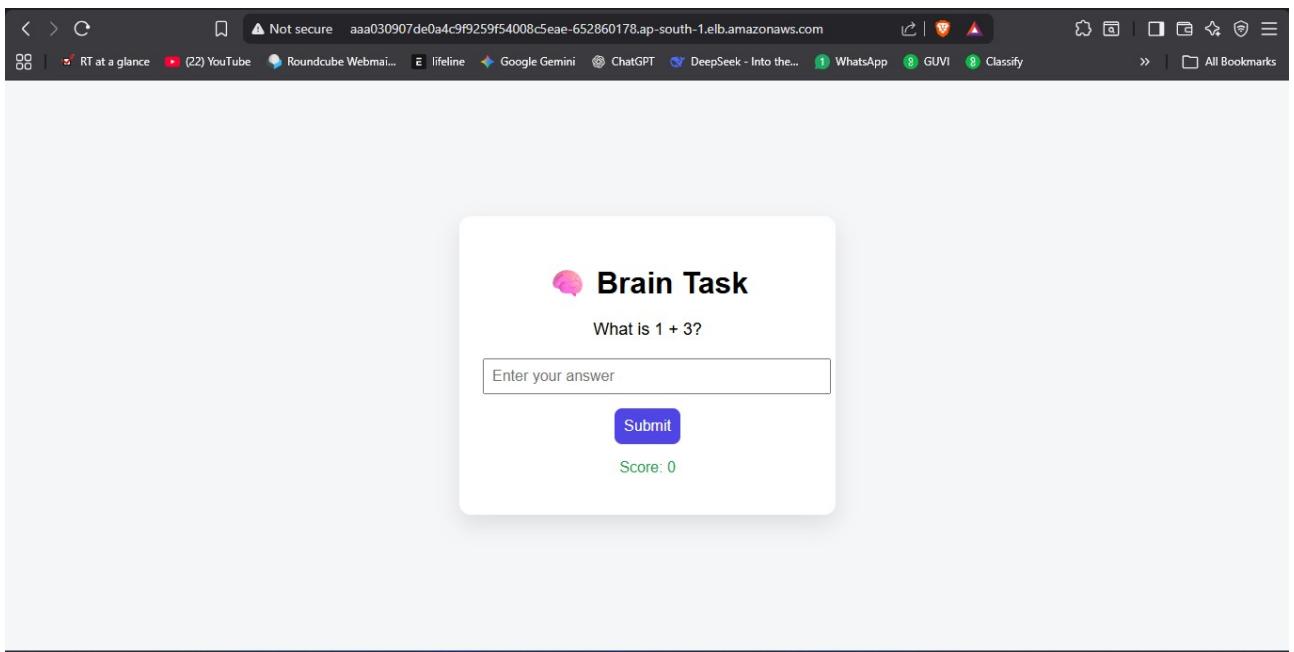
Metrics

All metrics

Log groups (2)

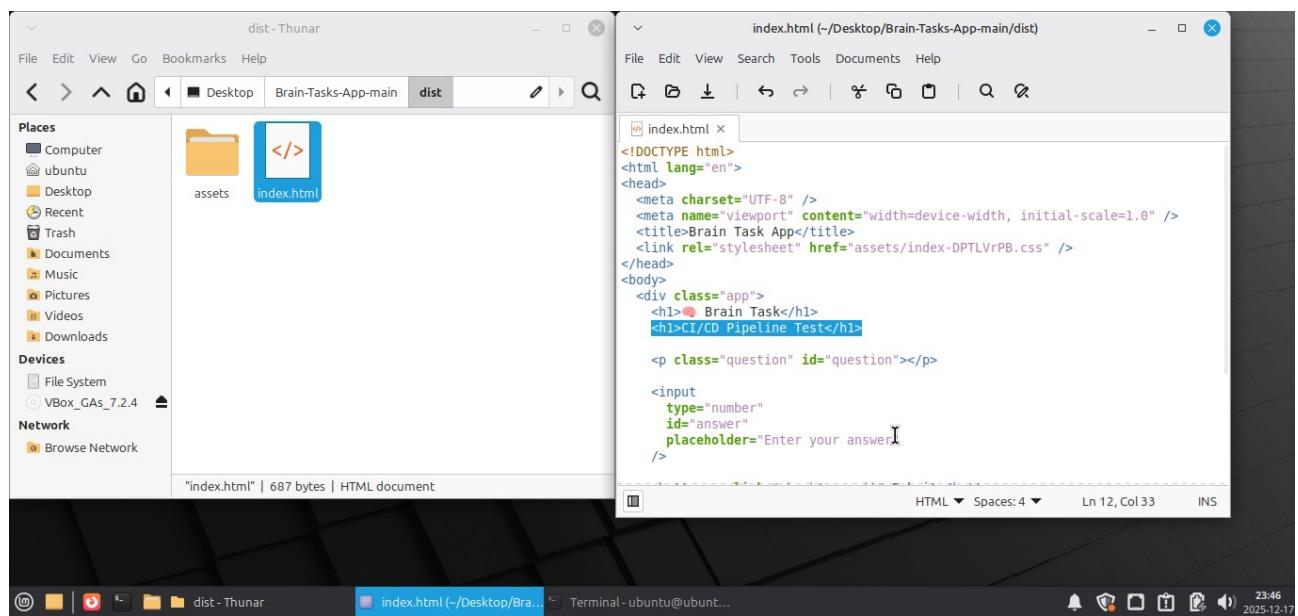
By default, we only load up to 10,000 log groups.

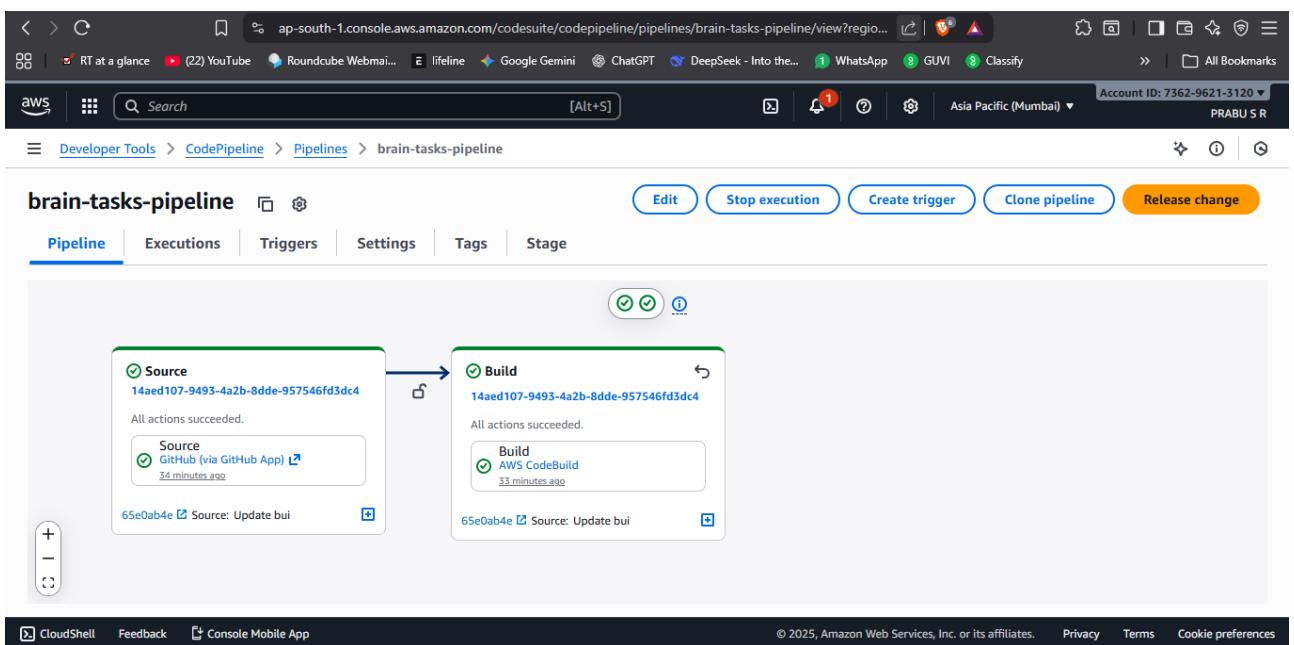
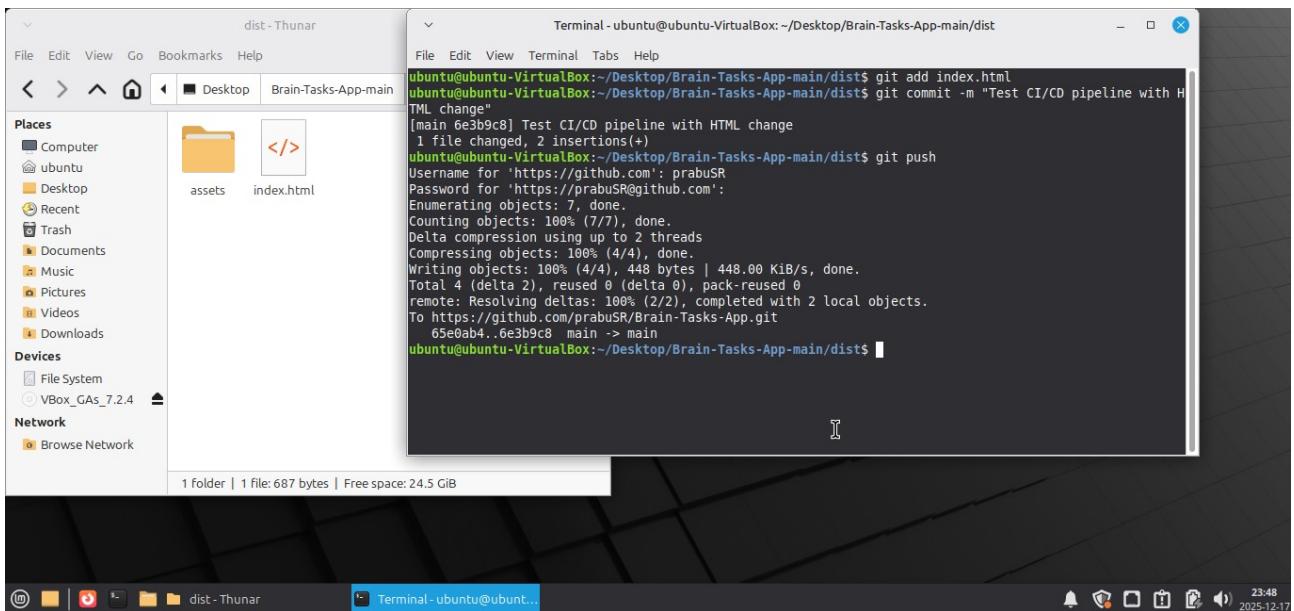
Action	Log group	Log class	Anomaly d...	Delete...	Data...	Se...	Reten...
Actions	/aws/codebuild/brain-tasks-build	Standard	Configure	Off	-	-	Never
Actions	/aws/eks/trendstore-eks/cluster	Standard	Configure	Off	-	-	3 mon



TESTING THE PROJECT

Any code change pushed to GitHub automatically triggers build, containerization, image storage, and deployment to Kubernetes, without any manual intervention.

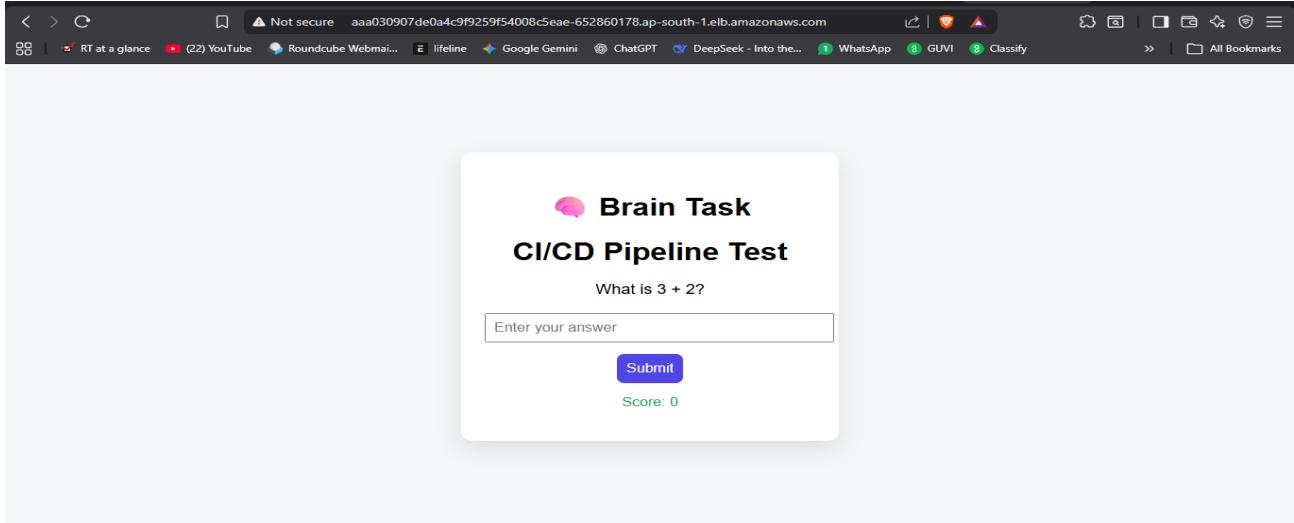




RESULT:

LB ARN URL:

<http://aaa030907de0a4c9f9259f54008c5eae-652860178.ap-south-1.elb.amazonaws.com>



Conclusion:

Overall, this project reflects a **real-world DevOps deployment model** used in production environments and demonstrates practical expertise in cloud-native CI/CD automation, container orchestration, and AWS infrastructure services. It serves as a strong foundation for scaling applications, improving release velocity, and adopting infrastructure-as-code and monitoring enhancements in future iterations.