

Brain Tasks App – End-to-End DevOps Deployment

This project demonstrates deploying a React application into a **production-ready DevOps pipeline** using **Docker, AWS ECR, AWS EKS, CodeBuild, and CodePipeline**.

The goal of this project is to showcase **CI/CD automation**, containerization, Kubernetes deployment, and AWS cloud services integration.

Application Overview

- **Application:** Brain Tasks App (React)
 - **Port:** 3000
 - **Container Runtime:** Docker
 - **Container Registry:** Amazon ECR
 - **Orchestration:** Amazon EKS (Kubernetes)
 - **CI/CD:** AWS CodeBuild + AWS CodePipeline
 - **Monitoring:** Amazon CloudWatch Logs
-

Repository Structure

```
.  
├── Dockerfile  
├── nginx.conf  
├── buildspec.yml  
├── deployment.yaml  
├── service.yaml  
├── dist/  
└── README.md
```

Dockerization

The React application is built and served using **Nginx** inside a Docker container.

Dockerfile

- Uses nginx:alpine
- Serves production build from /usr/share/nginx/html
- Exposes port 3000

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ docker build -t brain-tasks-app .
failed to fetch metadata: fork/exec /usr/local/lib/docker/cli-plugins/docker-buildx: no such file or directory

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 487.4kB
Step 1/5 : FROM nginx:alpine
--> 04da2b0513cd
Step 2/5 : COPY dist /usr/share/nginx/html
--> 7196768a29b0
Step 3/5 : COPY nginx.conf /etc/nginx/conf.d/default.conf
--> 54abf53bcce6
Step 4/5 : EXPOSE 3000
--> Running in aa05186afca8
--> Removed intermediate container aa05186afca8
--> 84b0e7299219
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
--> Running in a18587e81f14
--> Removed intermediate container a18587e81f14
--> 9736170df068
Successfully built 9736170df068
Successfully tagged brain-tasks-app:latest
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ docker run -d -p 3000:3000 brain-tasks-app
5bbdc258b2b62afdd8714a777d90dad592e061c30a3b4fd0cfef7382a22f990fd
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS        PORTS
NAMES
5bbdc258b2b6   brain-tasks-app   "/docker-entrypoint..."   12 seconds ago   Up 11 seconds   80/tcp, 0.0.0
.0:3000->3000/tcp, [::]:3000->3000/tcp   strange_hawking
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

Amazon ECR (Elastic Container Registry)

- An ECR repository was created to store Docker images
- Docker image was tagged and pushed to ECR

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ aws ecr create-repository \
--repository-name brain-tasks-app \
--region ap-south-1

An error occurred (RepositoryAlreadyExistsException) when calling the CreateRepository operation: The repository with name 'brain-tasks-app' already exists in the registry with id '727598134512'
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ aws ecr describe-repositories --region ap-south-1
{
  "repositories": [
    {
      "repositoryArn": "arn:aws:ecr:ap-south-1:727598134512:repository/brain-tasks-app",
      "registryId": "727598134512",
      "repositoryName": "brain-tasks-app",
      "repositoryUri": "727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app",
      "createdAt": "2025-12-25T04:06:01.434000+00:00",
      "imageTagMutability": "MUTABLE",
      "imageScanningConfiguration": {
        "scanOnPush": false
      },
      "encryptionConfiguration": {
        "encryptionType": "AES256"
      }
    }
  ]
}
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ docker push 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
The push refers to repository [727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app]
61a05dab69dd: Pushed
80110fdb89e5: Pushed
e6fe11fa5b7f: Layer already exists
67ea0b046e7d: Layer already exists
ed5fa8595c7a: Layer already exists
8ae63eb1f31f: Layer already exists
b3e3d1bb64d: Layer already exists
48078b7e3000: Layer already exists
fd1e40d7f74b: Layer already exists
7bb20cf5ef67: Layer already exists
latest: digest: sha256:64b38f4c36ffb55ebe77c5d80b139ed03072c8da20ae8b090f1bc4bb3fe80505 size: 2406
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ -

```

Kubernetes Deployment (Amazon EKS)

EKS Cluster

- EKS cluster created using eksctl
- Worker nodes provisioned successfully

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App
7bb20cf5ef67: Layer already exists
latest: digest: sha256:64b38f4c36ffb55ebe77c5d80b139ed03072c8da20ae8b090f1bc4bb3fe80505 size: 2406
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl version --client
eksctl version
Client Version: v1.35.0
Kustomize Version: v5.7.1
0.221.0
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ eksctl create cluster \
  --name brain-tasks-cluster \
  --region ap-south-1 \
  --nodegroup-name brain-nodes \
  --node-type t3.medium \
  --nodes 2
2026-01-15 09:35:31 [ℹ] eksctl version 0.221.0
2026-01-15 09:35:31 [ℹ] using region ap-south-1
2026-01-15 09:35:32 [ℹ] setting availability zones to [ap-south-1c ap-south-1a ap-south-1b]
2026-01-15 09:35:32 [ℹ] subnets for ap-south-1c - public:192.168.0.0/19 private:192.168.96.0/19
2026-01-15 09:35:32 [ℹ] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
2026-01-15 09:35:32 [ℹ] subnets for ap-south-1b - public:192.168.64.0/19 private:192.168.160.0/19
2026-01-15 09:35:32 [ℹ] nodegroup "brain-nodes" will use "" [AmazonLinux2023/1.32]
2026-01-15 09:35:32 [!] 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/
2026-01-15 09:35:32 [ℹ] using Kubernetes version 1.32
2026-01-15 09:35:32 [ℹ] creating EKS cluster "brain-tasks-cluster" in "ap-south-1" region with managed nodes
2026-01-15 09:35:32 [ℹ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2026-01-15 09:35:32 [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=brain-tasks-cluster'
2026-01-15 09:35:32 [ℹ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "brain-tasks-cluster" in "ap-south-1"
2026-01-15 09:35:32 [ℹ] CloudWatch logging will not be enabled for cluster "brain-tasks-cluster" in "ap-south-1"
2026-01-15 09:35:32 [ℹ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=brain-tasks-cluster'
2026-01-15 09:35:32 [ℹ] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2026-01-15 09:35:32 [ℹ]
2 sequential tasks: { create cluster control plane "brain-tasks-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "brain-nodes",
  }
}
2026-01-15 09:35:32 [ℹ] building cluster stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:35:33 [ℹ] deploying stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:36:06 [ℹ] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App
2026-01-15 09:35:32 [i] 2 sequential tasks: { create cluster control plane "brain-tasks-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "brain-nodes",
  }
}
2026-01-15 09:35:32 [i] building cluster stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:35:33 [i] deploying stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:36:06 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:36:39 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:37:43 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:38:47 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:39:51 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:40:57 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:42:03 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:43:10 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:44:16 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-cluster"
2026-01-15 09:44:19 [!] 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 p
ermissions 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 upda
te addon`
2026-01-15 09:44:19 [i] creating addon: vpc-cni
2026-01-15 09:44:19 [i] successfully created addon: vpc-cni
2026-01-15 09:44:20 [i] creating addon: kube-proxy
2026-01-15 09:44:20 [i] successfully created addon: kube-proxy
2026-01-15 09:44:20 [i] creating addon: coredns
2026-01-15 09:44:21 [i] successfully created addon: coredns
2026-01-15 09:46:36 [i] building managed nodegroup stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:46:37 [i] deploying stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 09:46:37 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:47:10 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:48:15 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:49:27 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:50:20 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:51:47 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:53:24 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:55:02 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
des"
2026-01-15 09:56:59 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-no
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App
2026-01-15 10:05:59 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:07:32 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:08:46 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:10:29 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:11:11 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:11:45 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:13:16 [i] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:13:16 [!] 1 error(s) occurred and cluster hasn't been created properly, you may wish to check CloudFormation console
2026-01-15 10:13:16 [i] to cleanup resources, run 'eksctl delete cluster --region=ap-south-1 --name=brain-tasks-cluster'
2026-01-15 10:13:16 [X] exceeded max wait time for StackCreateComplete waiter
Error: failed to create cluster "brain-tasks-cluster"
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ eksctl delete nodegroup \
--cluster brain-tasks-cluster \
--region ap-south-1 \
--name brain-nodes
2026-01-15 10:40:36 [i] 1 nodegroup (brain-nodes) was included (based on the include/exclude rules)
2026-01-15 10:40:36 [i] will drain 1 nodegroup(s) in cluster "brain-tasks-cluster"
2026-01-15 10:40:36 [i] starting parallel draining, max in-flight of 1
2026-01-15 10:40:37 [!] no nodes found in nodegroup "brain-nodes" (label selector: "alpha.eksctl.io/nodegroup-name=brain-nodes")
2026-01-15 10:40:37 [i] will delete 1 nodegroups from cluster "brain-tasks-cluster"
2026-01-15 10:40:37 [i] 1 task: { 1 task: { delete nodegroup "brain-nodes" [async] } }
2026-01-15 10:40:37 [i] will delete stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:40:37 [✓] deleted 1 nodegroup(s) from cluster "brain-tasks-cluster"
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ eksctl create nodegroup \
--cluster brain-tasks-cluster \
--region ap-south-1 \
--name brain-nodes \
--node-type t3.small \
--nodes 1 \
--nodes-min 1 \
--nodes-max 1
2026-01-15 10:40:59 [i] will use version 1.32 for new nodegroup(s) based on control plane version
2026-01-15 10:41:02 [i] nodegroup "brain-nodes" will use "" [AmazonLinux2023/1.32]
2026-01-15 10:41:02 [i] 1 nodegroup (brain-nodes) was included (based on the include/exclude rules)
2026-01-15 10:41:02 [i] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "brain-tasks-cluster"
2026-01-15 10:41:02 [i] 2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "brain-nodes" } } }
2026-01-15 10:41:02 [i] checking cluster stack for missing resources
2026-01-15 10:41:03 [i] cluster stack has all required resources
```

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App
2026-01-15 10:41:37 [ ] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:42:34 [ ] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:43:42 [ ] waiting for CloudFormation stack "eksctl-brain-tasks-cluster-nodegroup-brain-nodes"
2026-01-15 10:43:42 [ ] no tasks
2026-01-15 10:43:42 [✓] created 0 nodegroup(s) in cluster "brain-tasks-cluster"
2026-01-15 10:43:42 [ ] nodegroup "brain-nodes" has 1 node(s)
2026-01-15 10:43:42 [ ] node "ip-192-168-3-94.ap-south-1.compute.internal" is ready
2026-01-15 10:43:42 [ ] waiting for at least 1 node(s) to become ready in "brain-nodes"
2026-01-15 10:43:42 [ ] nodegroup "brain-nodes" has 1 node(s)
2026-01-15 10:43:42 [ ] node "ip-192-168-3-94.ap-south-1.compute.internal" is ready
2026-01-15 10:43:42 [✓] created 1 managed nodegroup(s) in cluster "brain-tasks-cluster"
2026-01-15 10:43:43 [ ] checking security group configuration for all nodegroups
2026-01-15 10:43:43 [ ] all nodegroups have up-to-date cloudformation templates
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl get nodes
E0115 10:44:22.504217 3414 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: the server has asked for the client to provide credentials"
E0115 10:44:24.195825 3414 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: the server has asked for the client to provide credentials"
E0115 10:44:26.134077 3414 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: the server has asked for the client to provide credentials"
E0115 10:44:30.815659 3414 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: the server has asked for the client to provide credentials"
^C
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ Traceback (most recent call last):
File "PyInstaller/hooks/rthooks/pyi_rth_multiprocessing.py", line 54, in <module>
File "PyInstaller/hooks/rthooks/pyi_rth_multiprocessing.py", line 17, in _pyi_rthook
File "<frozen importlib._bootstrap>", line 1360, in _find_and_load
File "<frozen importlib._bootstrap>", line 1331, in _find_and_load_unlocked
File "<frozen importlib._bootstrap>", line 935, in _load_unlocked
File "PyInstaller/loader/pyimod02_importers.py", line 384, in exec_module
File "multiprocessing/spawn.py", line 19, in <module>
File "<frozen importlib._bootstrap>", line 1360, in _find_and_load
File "<frozen importlib._bootstrap>", line 1331, in _find_and_load_unlocked
File "<frozen importlib._bootstrap>", line 935, in _load_unlocked
File "PyInstaller/loader/pyimod02_importers.py", line 373, in exec_module
File "PyInstaller/loader/pyimod02_importers.py", line 452, in get_code
File "PyInstaller/loader/pyimod01_archive.py", line 129, in extract
KeyboardInterrupt
[PYI-3517:ERROR] Failed to execute script 'pyi_rth_multiprocessing' due to unhandled exception!
^C
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ aws eks update-kubeconfig \
--region ap-south-1 \
--name brain-tasks-cluster
Added new context arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster to /home/junaid/.kube/config
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ 

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ aws eks update-kubeconfig \
--region ap-south-1 \
--name brain-tasks-cluster
Added new context arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster to /home/junaid/.kube/config
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl config get-contexts
CURRENT      NAME                                     CLUSTER
                                         AUTHINFO
NAMESPACE
    arn:aws:eks:ap-south-1:727598134512:cluster/brain-cluster          arn:aws:eks:ap-south-1:727598134512:cluster/brain-cluster
8134512:cluster/brain-cluster          arn:aws:eks:ap-south-1:727598134512:cluster/brain-cluster
*      arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster   arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster
8134512:cluster/brain-tasks-cluster   arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster
                                         arn:aws:eks:us-east-1:727598134512:cluster/trend-eks-cluster   arn:aws:eks:us-east-1:727598134512:cluster/trend-eks-cluster
134512:cluster/trend-eks-cluster     arn:aws:eks:us-east-1:727598134512:cluster/trend-eks-cluster
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl get nodes
NAME                  STATUS  ROLES   AGE     VERSION
ip-192-168-3-94.ap-south-1.compute.internal  Ready   <none>  4m28s  v1.32.9-eks-ecaa3a6
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ 

```

Kubernetes Resources

- **Deployment:** Defines application pods
- **Service:** Exposes application using LoadBalancer

kubectl get pods

kubectl get svc

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ aws eks update-kubeconfig \
--region ap-south-1 \
--name brain-tasks-cluster
Added new context arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster to /home/junaid/.kube/
config
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl config get-contexts
CURRENT      NAME                                     CLUSTER
NAMESPACE
    arn:aws:eks:ap-south-1:727598134512:cluster/brain-cluster      arn:aws:eks:ap-south-1:72759
8134512:cluster/brain-cluster      arn:aws:eks:ap-south-1:727598134512:cluster/brain-cluster
*      arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster  arn:aws:eks:ap-south-1:72759
8134512:cluster/brain-tasks-cluster  arn:aws:eks:ap-south-1:727598134512:cluster/brain-tasks-cluster
    arn:aws:eks:us-east-1:727598134512:cluster/trend-eks-cluster     arn:aws:eks:us-east-1:727598
134512:cluster/trend-eks-cluster     arn:aws:eks:us-east-1:727598134512:cluster/trend-eks-cluster
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl get nodes
NAME                  STATUS   ROLES   AGE     VERSION
ip-192-168-3-94.ap-south-1.compute.internal   Ready   <none>  4m28s   v1.32.9-eks-ecaa3a6
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ cat << 'EOF' > deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: brain-tasks
  template:
    metadata:
      labels:
        app: brain-tasks
    spec:
      containers:
        - name: brain-tasks
          image: 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
          ports:
            - containerPort: 3000
EOF
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: brain-tasks-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: brain-tasks
  template:
    metadata:
      labels:
        app: brain-tasks
    spec:
      containers:
        - name: brain-tasks
          image: 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
          ports:
            - containerPort: 3000
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ cat << 'EOF' > service.yaml
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
spec:
  type: LoadBalancer
  selector:
    app: brain-tasks
  ports:
  - port: 80
    targetPort: 3000
EOF
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-service
spec:
  type: LoadBalancer
  selector:
    app: brain-tasks
  ports:
  - port: 80
    targetPort: 3000
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
deployment.apps/brain-tasks-deployment created
service/brain-tasks-service created
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
brain-tasks-deployment-55bb4d84fd-nkr67  1/1     Running   0          32s
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ kubectl get svc brain-tasks-service
NAME            TYPE      CLUSTER-IP   EXTERNAL-IP
                PORT(S)    AGE
brain-tasks-service   LoadBalancer  10.100.124.89   aae583ba7d7fc4bea8053c2417b2b71c-7188186.ap-south-1
.elb.amazonaws.com   80:31701/TCP  45s
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

[SCREENSHOT: Application LoadBalancer URL]

Filename: 08_App_LoadBalancer_URL.png

CI/CD with AWS CodeBuild

CodeBuild Project

- Source: GitHub repository
- Build environment: Managed image
- Build logic defined in buildspec.yml

Build Actions

- Login to Amazon ECR
- Build Docker image
- Push Docker image to ECR

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ cat << 'EOF' > buildspec.yml
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR
      - aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 72
7598134512.dkr.ecr.ap-south-1.amazonaws.com

  build:
    commands:
      - echo Building Docker image
      - docker build -t brain-tasks-app .
      - docker tag brain-tasks-app:latest 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:
latest

  post_build:
    commands:
      - echo Pushing Docker image
      - docker push 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
EOF

```

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote -v
origin https://github.com/Vennilavan12/Brain-Tasks-App.git (fetch)
origin https://github.com/Vennilavan12/Brain-Tasks-App.git (push)
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote set-url origin https://github.com
/Junaid-dot-max/Brain-Tasks-App.git
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote -v
origin https://github.com/Junaid-dot-max/Brain-Tasks-App.git (fetch)
origin https://github.com/Junaid-dot-max/Brain-Tasks-App.git (push)
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 542 bytes | 38.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Junaid-dot-max/Brain-Tasks-App.git
  b7d121f..fe7b1f1 main -> main
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

The screenshot shows the AWS CloudWatch Metrics interface. A chart titled 'AWS Lambda' displays the metric 'Function Invocations' over a one-hour period. The data shows a significant spike starting around 10:30 AM UTC on March 21, 2024, peaking at approximately 10,000 invocations. The Y-axis represents the number of invocations from 0 to 10,000, and the X-axis shows the time from 10:00 AM to 11:00 AM UTC.

Queued timeout
Default time in build queue is 8 hours

Hours Minutes

Timeout must be between 5 minutes and 8 hours

Privileged
 Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Report auto-discover info Disable report auto-discover

Auto-discover directory - optional
CodeBuild will search for supported report file types in this directory. **/* by default

Certificate
If you have a self-signed certificate or a certificate signed by a certification authority, choose the option to install it from your S3 bucket.

Do not install any certificate Install certificate from your S3 bucket

VPC
Select a VPC that your AWS CodeBuild project will access.

Compute

- 2 vCPUs, 4 GiB memory
- 4 vCPUs, 8 GiB memory
- 8 vCPUs, 16 GiB memory

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 Amazon Linux

Runtime(s) Standard

Image [aws/codebuild/amazonlinux-x86_64-standard:5.0](#)

Image version Always use the latest image for this runtime version

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

Create build project | CodeBuild **CodeBuild** **Build projects** **Create build project** **CodeBuild** **CodeBuild** **Build projects** **brain-tasks-build** **Create notification rule for this project** **Actions** **Create trigger** **Edit** **Clone** **Debug build** **Start build with overrides** **Start build**

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

Configuration

| | | | |
|------------------------|---|---------------------------|---|
| Source provider GitHub | Primary repository Junaid-dot-max/Brain-Tasks-App | Artifacts upload location | Service role arn:aws:iam::727598134512:role/service-role/codebuild-brain-tasks-build-service-role |
| Public builds Enabled | | | |

Build history

| Build run | Status | Build number | Source version | Submitter | Duration | Completed |
|------------|----------------------------------|--------------|----------------|-----------|----------|-----------|
| No results | There are no results to display. | | | | | |

Screenshot of the AWS CloudWatch Logs interface showing the log stream for the build step. The logs are displayed in a monospaced font.

```

1 [Container] 2026/01/15 11:26:25.391971 Running on CodeBuild-on-demand
2 [Container] 2026/01/15 11:26:25.391971 Waiting for agent ping
3 [Container] 2026/01/15 11:26:25.391971 Received agent ping
4 [Container] 2026/01/15 11:26:31.765272 Phase is DOWNLOAD_SOURCE
5 [Container] 2026/01/15 11:26:31.765272 Container location: /codebuild/output/src2171181391/github.com/junaid-dot-max/Brain-Tasks-app
6 [Container] 2026/01/15 11:26:31.765272 Code source type: GitHub
7 [Container] 2026/01/15 11:26:31.765272 GitHub location: /codebuild/output/src2171181391/github.com/junaid-dot-max/Brain-Tasks-app/buildspec.yml
8 [Container] 2026/01/15 11:26:31.765272 Setting HTTP client timeout to higher timeout for GitHub and GitHub Enterprise sources
9 [Container] 2026/01/15 11:26:31.765272 GitHub timeout: 1000
10 [Container] 2026/01/15 11:26:32.987209 No runtime version selected in buildspec.
11 [Container] 2026/01/15 11:26:33.956431 Cache was not found in the local cache directory /codebuild/output/src2171181391/github.com/junaid-dot-max/Brain-Tasks-app
12 [Container] 2026/01/15 11:26:34.073813 Skip cache due to no paths specified to be cached
13 [Container] 2026/01/15 11:26:34.073813 Cache was not found in the local cache directory /codebuild/output/src2171181391/github.com/junaid-dot-max/Brain-Tasks-app
14 [Container] 2026/01/15 11:26:34.073813 Phases found in YAML: 3
15 [Container] 2026/01/15 11:26:34.073813 Phase: build, Type: BuildSpec
16 [Container] 2026/01/15 11:26:34.073813 BUILD: 3 commands
17 [Container] 2026/01/15 11:26:34.073848 POST_BUILD: 2 commands
18 [Container] 2026/01/15 11:26:34.073848 Phase build completed. SOURCE_SOURCE status: SUCCESSFUL
19 [Container] 2026/01/15 11:26:34.074557 Phase context status code: Message: 
20 [Container] 2026/01/15 11:26:34.074557 Phase context status code: SUCCESS
21 [Container] 2026/01/15 11:26:35.714262 Phase complete: INSTALL status: SUCCESS
22 [Container] 2026/01/15 11:26:35.714262 Phase complete: PRE_BUILD status: FAILURE
23 [Container] 2026/01/15 11:26:36.275362 Running command echo Logging in to Amazon ECR
24 [Container] 2026/01/15 11:26:36.275362 Logging in to Amazon ECR
25 [Container] 2026/01/15 11:26:36.282688 Running command aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 727598134512.dkr.ecr.ap-south-1.amazonaws.com
26 [Container] 2026/01/15 11:26:36.282688 An error occurred (AccessDeniedException) when calling the GetAuthorizationToken operation: user: arn:aws:sts::727598134512:assumed-role/codebuild-brain-tasks-build-service-role/e0c0d61d-0344-5583-f641-4084-b40a-d9
27 [Container] 2026/01/15 11:26:36.282688 Caused by: The user is not authorized to perform the requested action on resource: * because no identity-based policy allows the ecr:DescribeRepositories action
28 [Container] 2026/01/15 11:26:36.282688 Error: Cannot perform an interactive login from a non TTY device
29 [Container] 2026/01/15 11:26:46.494615 Command did not exit successfully: aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 727598134512.dkr.ecr.ap-south-1.amazonaws.com exit
30 [Container] 2026/01/15 11:26:46.500384 Phase complete: PRE_BUILD status: FAILURE
31 [Container] 2026/01/15 11:26:46.500384 Phase context status code: COMMAND_EXECUTION_ERROR message: Error while executing command: aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-
32 [Container] 2026/01/15 11:26:46.500384 stdin 727598134512.dkr.ecr.ap-south-1.amazonaws.com. Reason: exit status 1

```

Screenshot of the AWS IAM Roles interface showing the details of the "codebuild-brain-tasks-build-service-role".

Summary

- Creation date: January 15, 2026, 16:53 (UTC+05:30)
- Last activity: 1 hour
- ARN: arn:aws:iam::727598134512:role/service-role/codebuild-brain-tasks-build-service-role
- Maximum session duration: 1 hour

Permissions

Permissions policies (2)

You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|---|------------------|-------------------|
| CodeBuildBasePolicy-brain-tasks-build-ap-south-1 | Customer managed | 1 |
| CodeBuildCodeConnectionsSourceCredentialsPolicy-brain-ta... | Customer managed | 1 |

Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

[Generate policy](#)

No requests to generate a policy in the past 7 days.

The screenshot shows the AWS IAM Roles page. The main content area displays a success message: "Policy was successfully attached to role." Below this, it shows the creation date as January 15, 2026, at 16:53 (UTC+05:30), and the ARN as arn:aws:iam::727598134512:role/service-role/codebuild-brain-tasks-build-service-role. It also indicates a maximum session duration of 1 hour.

The "Permissions" tab is selected, showing three managed 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-brain-ta... | Customer managed | 1 |

Below the permissions section, there is a "Permissions boundary (not set)" section and a "Generate policy based on CloudTrail events" section. The "Generate policy" button is visible in the latter section, along with a note that no requests have been generated in the past 7 days.

```

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App
-bash: junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$: No such file or directory
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ ^C
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote -v
origin https://github.com/Vennilavan12/Brain-Tasks-App.git (fetch)
origin https://github.com/Vennilavan12/Brain-Tasks-App.git (push)
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote set-url origin https://github.com/Junaid-dot-max/Brain-Tasks-App.git
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git remote -v
origin https://github.com/Junaid-dot-max/Brain-Tasks-App.git (fetch)
origin https://github.com/Junaid-dot-max/Brain-Tasks-App.git (push)
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 542 bytes | 38.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Junaid-dot-max/Brain-Tasks-App.git
  b7d121f..fe7b1f1  main -> main
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ ls
Dockerfile buildspec.yml deployment.yaml dist nginx.conf service.yaml
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git add Dockerfile nginx.conf
git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  Dockerfile
    new file:  nginx.conf

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    deployment.yaml
    service.yaml

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git commit -m "Add Dockerfile and nginx config for CodeBuild"
[main 4cc2da9] Add Dockerfile and nginx config for CodeBuild
2 files changed, 16 insertions(+)
create mode 100644 Dockerfile
create mode 100644 nginx.conf
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 610 bytes | 30.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Junaid-dot-max/Brain-Tasks-App.git
  fe7b1f1..4cc2da9  main -> main
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid/Brain-Tasks-App$
```

Build started
You have successfully started the following build: brain-tasks-build:b2a8a0a5-45c7-4c05-bc8a-5200cafd6a70

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

| Build status | | Build ARN | | Resolved source version |
|--------------|---------------------------------|-----------|---------------------------------|--|
| Status | Succeeded | Initiator | root | arn:aws:codebuild:ap-south-1:727598134512:build/brain-tasks-build:b2a8a0a5-45c7-4c05-bc8a-5200cafd6a70 |
| Start time | Jan 15, 2026 5:02 PM (UTC+5:30) | End time | Jan 15, 2026 5:04 PM (UTC+5:30) | Build number |
| | | | | 3 |

[Build logs](#) [Phase details](#) [Reports](#) [Environment variables](#) [Build details](#) [Resource utilization](#)

Showing the last 100 lines of the build log. [View entire log](#)

No previous logs

1 [Container] 2026/01/15 11:33:45.262237 Running on CodeBuild On-demand
2 [Container] 2026/01/15 11:33:45.262258 Waiting for agent ping

[Tail logs](#)

The screenshot shows a browser window with multiple tabs open, including YouTube, Hotstar, Prime Video, Amazon.in, Flipkart, Netflix, and several AWS services like Application Dep., brain-tasks-build:b2a8a05-45c7-4c05-bc8a-52, and codebuild-bran. The main content area displays AWS Developer Tools > CodeBuild > Build projects > brain-tasks-build. A detailed log of a Docker build process is shown, starting with the command:

```
git clone https://github.com/junaid-al/brain-tasks-app.git /tmp/app
cd /tmp/app
cp -r . /var/www/html
rm -rf .git
```

The log continues through various stages of the build, including:

- Phase complete: BUILD state: SUCCEEDED
- Phase context status code: Message: Phase complete: POST_BUILD
- Running Command docker push 727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app:latest
- Pushing Docker Image
- Pushed to repository [727598134512.dkr.ecr.ap-south-1.amazonaws.com/brain-tasks-app]

At the bottom of the log, it shows the final state:

```
Phase complete: UPDATING_ARTIFACTS state: SUCCEEDED
```

A right-click context menu is visible over the log output, with options like Snipping Tool, Screenshot copied to clipboard, Automatically saved to screenshots folder, and Mark-up and share.

 CI/CD with AWS CodePipeline

Pipeline Stages

1. Source

- GitHub (via GitHub App)

2. Build

- AWS CodeBuild project

3. Deploy

- Skipped (Application already deployed to EKS)

The pipeline automatically triggers on every push to the GitHub repository.

Choose pipeline settings Info

Step 2 of 7

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

brain-tasks-pipeline

No more than 100 characters

Execution mode Info

Choose the execution mode for your pipeline. This determines how the pipeline is run.

- Superseded
- Queued
- Parallel

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

AWSCodePipelineServiceRole-ap-south-1-brain-tasks-pipeline

Type your service role name

- Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

► Advanced settings

Configure artifact store location, encryption settings, and pipeline variables for your pipeline.

[Cancel](#)

[Previous](#)

[Next](#)

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



or [Connect to GitHub](#)

Repository name

Choose a repository in your GitHub account.

https://github.com/Junaid-dot-max/Brain-Tasks-App X

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.

main

⚠ Not Found (Click here to retry)

Output artifact format

Choose the output artifact format.

CodePipeline default

AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

Full clone

AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions. [Learn more](#)

Enable automatic retry on stage failure

Webhook events

Webhook - optional

Start your pipeline on push and pull request events.

► Webhook event filters - optional

Starts your pipeline on a specific event

[Remove filters](#)

Add build stage - *optional* 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](#)

Build type

Single build

Triggers a single build.

Batch build

Triggers multiple builds as a single execution.

Region

Asia Pacific (Mumbai) ▾

Input artifacts

Choose an input artifact for this action. [Learn more](#) ↗

SourceArtifact X

Defined by: Source

Enable automatic retry on stage failure

[Cancel](#)

[Previous](#)

[Skip build stage](#)

[Next](#)

Review Info

Step 7 of 7

Step 2: Choose pipeline settings

Pipeline settings

Pipeline name

brain-tasks-pipeline

Pipeline type

V2

Execution mode

QUEUED

Artifact location

A new Amazon S3 bucket will be created as the default artifact store for your pipeline.

Service role name

AWSCodePipelineServiceRole-ap-south-1-brain-tasks-pipeline

Step 3: Add source stage

Source action provider

Source action provider

Github (via GitHub App)

OutputArtifactFormat

CODE_ZIP

DetectChanges

true

FullRepositoryId

https://github.com/Junaid-Din-Max/Brain-Tasks-App

Default branch

main

ConnectionArn

arn:aws:codeconnection:ap-south-1:727598154512:connection/SsecaQs-6e51-4568-8d10-e00de620a298

Enable automatic retry on stage failure

Enabled

Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type

No filter

Step 4: Add build stage

Build action provider

Build action provider

AWS CodeBuild

ProjectName

brain-tasks-build

Commands

-

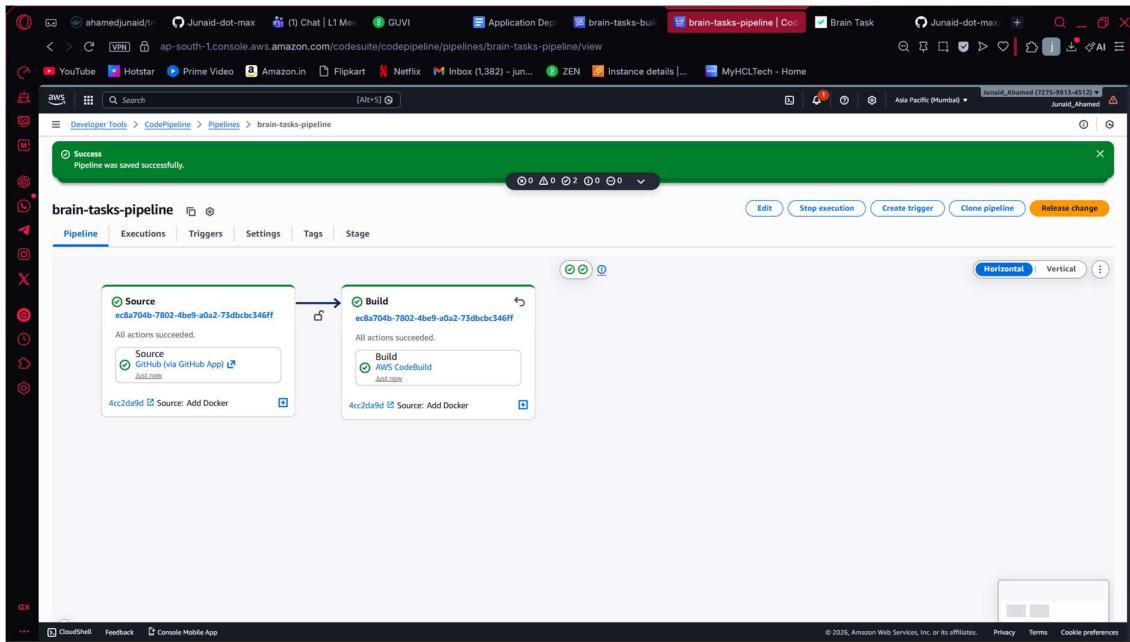
Enable automatic retry on stage failure

Enabled

Cancel

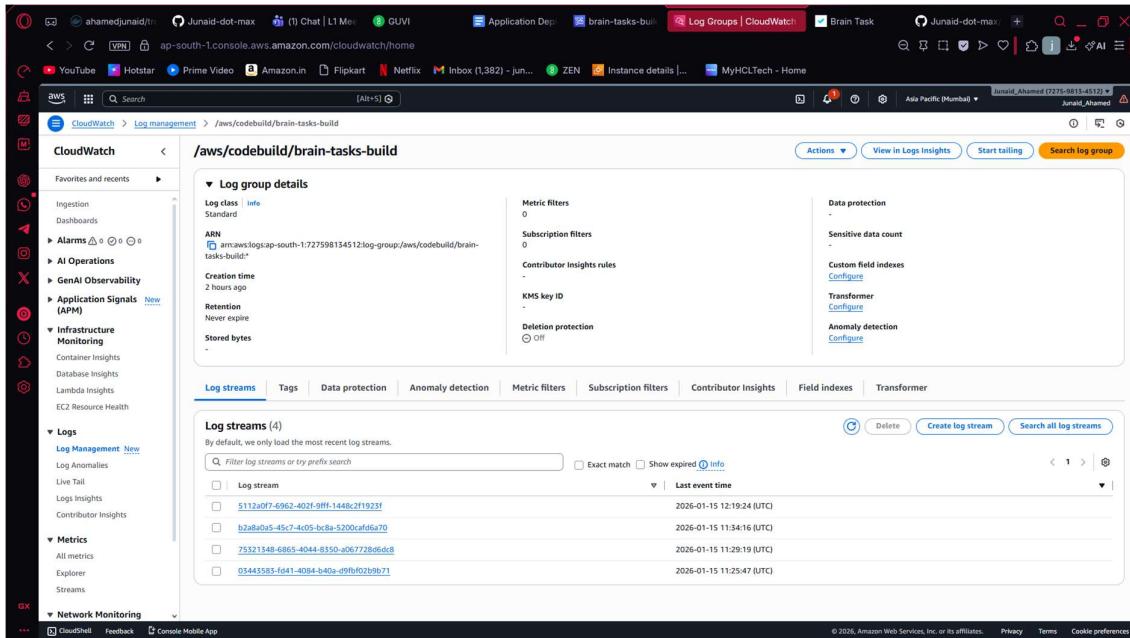
Previous

Create pipeline



Monitoring & Logs (CloudWatch)

- Build logs monitored using **Amazon CloudWatch**
- CodeBuild logs available under /aws/codebuild/brain-tasks-build



Final Outcome

- Application successfully deployed on **AWS EKS**

- Docker image stored in **Amazon ECR**
 - CI pipeline implemented using **CodeBuild**
 - End-to-end automation using **CodePipeline**
 - Logs monitored via **CloudWatch**
-

Key Learnings

- Dockerizing frontend applications
 - Working with AWS ECR and EKS
 - Writing Kubernetes manifests
 - Implementing CI pipelines using CodeBuild
 - Orchestrating CI with CodePipeline
 - Debugging real-world CI/CD issues
-

Cleanup (Optional – After Submission)

To avoid AWS charges:

```
eksctl delete cluster --name brain-tasks-cluster --region ap-south-1
```

Author

Junaid Ahamed

DevOps | Cloud | Kubernetes | AWS