

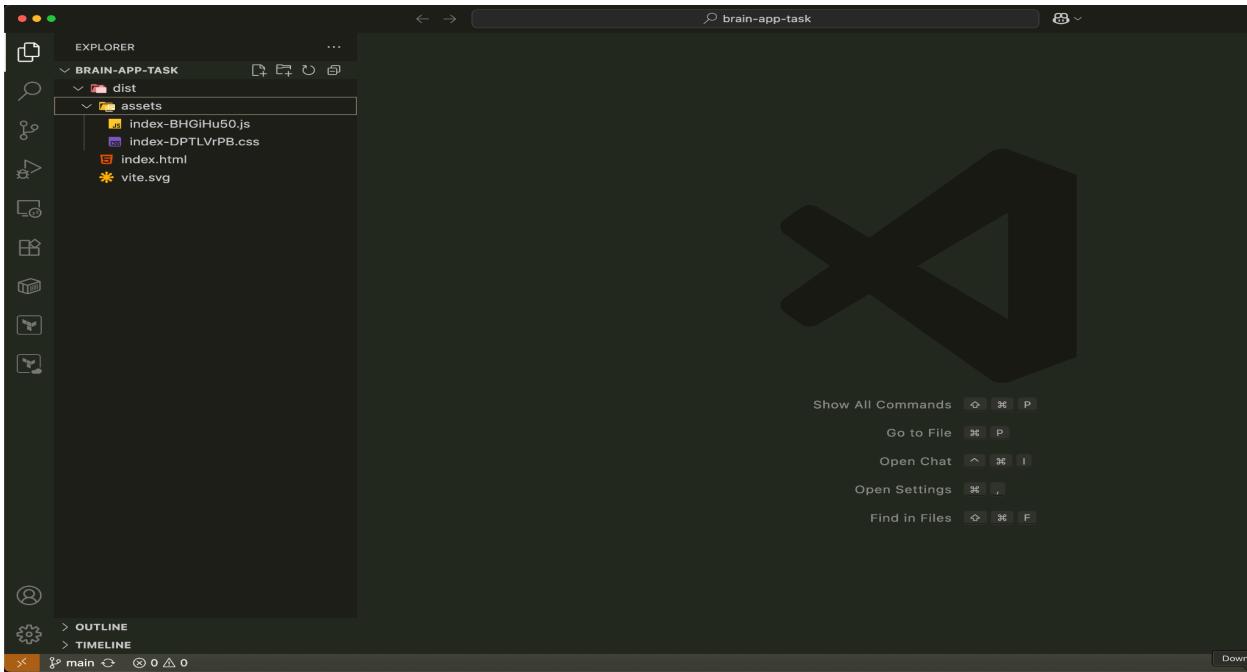
Brain Task App Deployment - Screenshots

1. Git Clone the React build folder

1a) Git Cloning the Repository

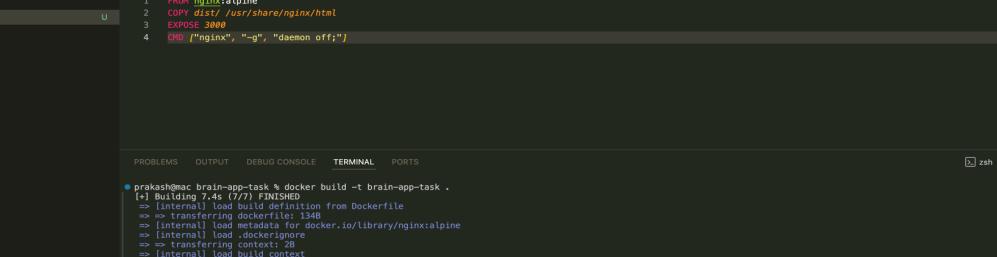
```
prakash@mac ~ % git clone https://github.com/VennilaSavan12/Brain-Tasks-App.git brain-app-task
Cloning into 'brain-app-task'...
remote: Enumerating objects: 8, done.
remote: Total 8 (deltas 0), reused 0 (deltas 0), pack-reused 8 (from 1)
Receiving objects: 100% (8/8) 100.04 KiB | 562.00 KiB/s, done.
prakash@mac ~ % cd brain-app-task
prakash@mac brain-app-task % ls
dist
prakash@mac brain-app-task % code .
```

1b) Opening and Verification of Project Folder



2. Dockerize the App

2a) Adding Dockerfile and creating image



The screenshot shows a terminal window titled "Dockerfile U" with the following content:

```
FROM nginx:alpine
COPY dist /usr/share/nginx/html
EXPOSE 3000
CMD ["nginx", "-g", "daemon off;"]
```

Below the terminal, a status bar indicates "zsh" and "docker:desktop:linux". The bottom of the screen shows a navigation bar with icons for "OUTLINE", "TIMELINE", "main*", "Timeline", "0 0 0 0", and "Go Live".

2b) Running created image on localhost 3000 port

2c) Verification of App Running localhost 3000 port

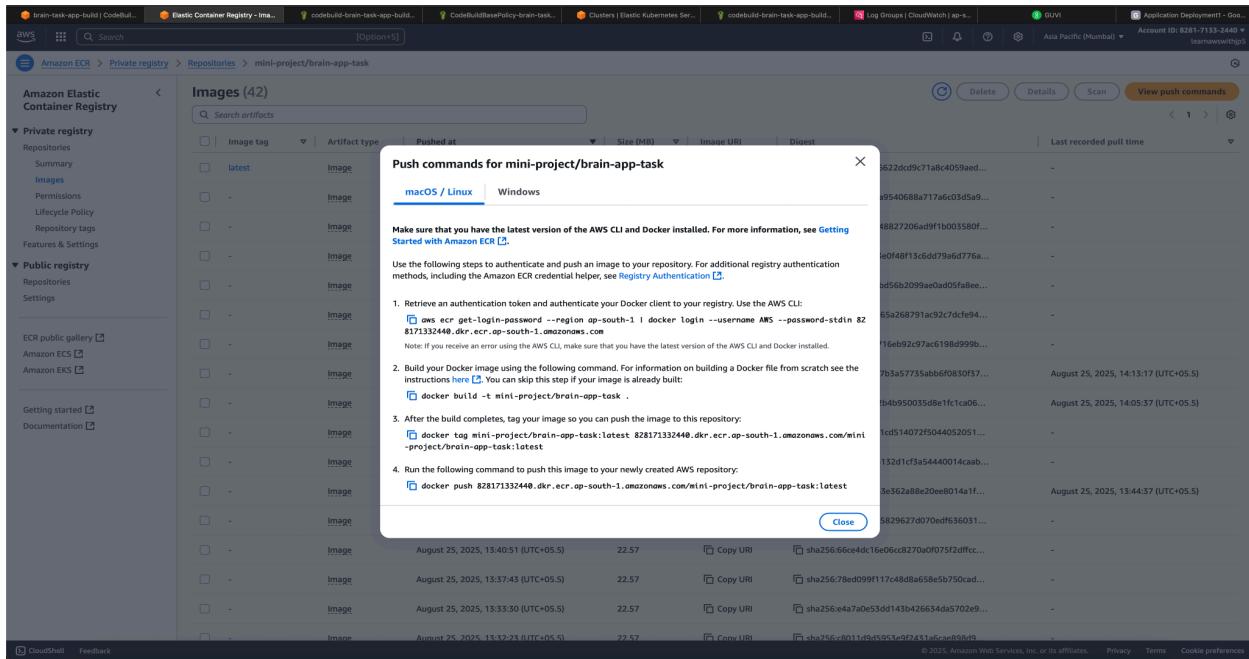
The screenshot shows a web application titled "Brain Tasks" running on localhost:3000. The interface is clean with a white background and light gray grid lines. At the top, there's a navigation bar with a search bar containing "Search tasks...". Below it are four filter buttons: "All Tasks" (which is selected), "Pending", "Completed", and "Priority". The main area contains a grid of 12 task cards arranged in three rows of four. Each card has a title, a short description, a status indicator (e.g., "Completed" or "Pending"), and a due date. A green "Create Task" button is located in the top right corner of the main content area.

3. AWS ECR config

3a) Creating New Repo

The screenshot shows the "Create a private repository" wizard in the AWS ECR console. The current step is "General settings". It includes a "Repository name" field with the value "417836837538.dkr.ecr.ap-south-1.amazonaws.com/brain-app-task". Below it is an "Image tag mutability" section with two options: "Mutable" (selected) and "Immutable". The "Image tag exclusion" section contains a text input field and a "Add filter" button. The next step, "Encryption settings", is visible below with a note about encryption settings being immutable once created. It offers "AES-256" and "AWS KMS" as encryption options. The final step, "Image scanning settings - deprecated", is shown at the bottom. At the very bottom of the screen, there are standard AWS navigation links for CloudShell, Feedback, and a footer with copyright information and cookie preferences.

3b) Push Commands



3c) Pushing image to new repo

4. EKS setup and config

4a) EKS cluster creation

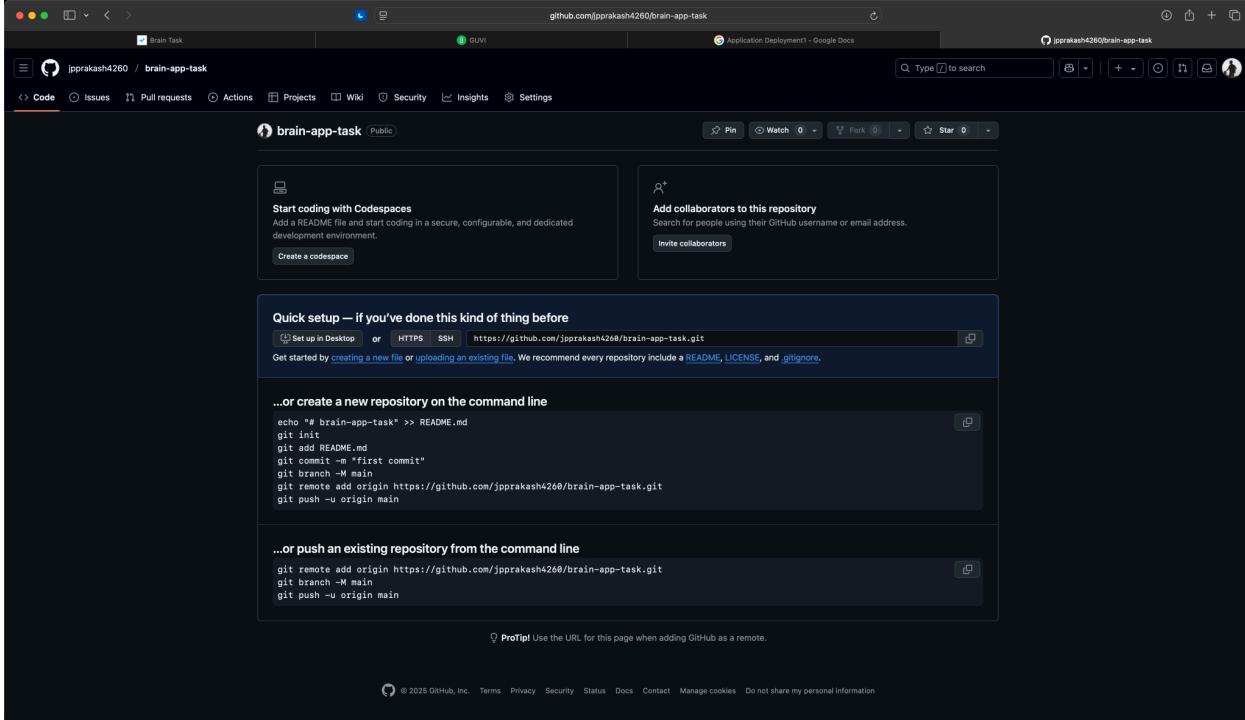
```
prakash@Mac: ~ % eksctl create cluster \
--name brain-tasks-app \
--region ap-south-1 \
--region ap-south-1 \
--nodegroup-name brain-tasks-nodes \
--node-type t3.medium \
--nodes 2 \
--node-labels app=brain-tasks \
--nodes-max 4 \
--profile EKS_user
2025-08-25 11:47:59 [x] eksctl version 0.211.0
2025-08-25 using region ap-south-1
2025-08-25 setting availability zones to '(ap-south-1b ap-south-1c ap-south-1a)'
2025-08-25 for ap-south-1b - public:[192.168.0.4/19 private:[192.168.1.4/19]
2025-08-25 for ap-south-1c - public:[192.168.0.5/19 private:[192.168.1.5/19]
2025-08-25 for ap-south-1a - public:[192.168.0.6/19 private:[192.168.1.6/19]
2025-08-25 for ap-south-1b - public:[192.168.0.7/19 private:[192.168.1.7/19]
2025-08-25 for ap-south-1c - public:[192.168.0.8/19 private:[192.168.1.8/19]
2025-08-25 for ap-south-1a - public:[192.168.0.9/19 private:[192.168.1.9/19]
2025-08-25 11:47:59 [x] using Kubernetes version 1.33
2025-08-25 11:47:59 [x] creating EKS cluster 'brain-tasks-app' in 'ap-south-1' region with managed nodes
2025-08-25 11:47:59 [x] you can enable any addons, check CloudFormation console or try 'eksctl util describe-stacks --region=ap-south-1 --cluster=brain-tasks-app'
2025-08-25 11:47:59 [x] CloudWatch logging will not be enabled for cluster 'brain-tasks-app' in 'ap-south-1'
2025-08-25 11:47:59 [x] you can enable it with 'eksctl util update-cluster-logging --enable-types=SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all) --region=ap-south-1 --cluster=brain-tasks-app'
2025-08-25 11:47:59 [x] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-08-25 11:47:59 [x] 2 sequential sub-tasks:
  2 sequential sub-tasks:
    2 sequential sub-tasks:
      2 sequential sub-tasks:
        2 sequential sub-tasks:
          2 sequential sub-tasks:
            2 sequential sub-tasks:
              2 sequential sub-tasks:
                2 sequential sub-tasks:
                  2 sequential sub-tasks:
                    2 sequential sub-tasks:
                      2 sequential sub-tasks:
                        2 sequential sub-tasks:
                          2 sequential sub-tasks:
                            2 sequential sub-tasks:
                              2 sequential sub-tasks:
                                2 sequential sub-tasks:
                                  2 sequential sub-tasks:
                                    2 sequential sub-tasks:
                                      2 sequential sub-tasks:
                                        2 sequential sub-tasks:
                                          2 sequential sub-tasks:
                                            2 sequential sub-tasks:
                                              2 sequential sub-tasks:
                                                2 sequential sub-tasks:
                                                  2 sequential sub-tasks:
                                                    2 sequential sub-tasks:
                                                      2 sequential sub-tasks:
                                                        2 sequential sub-tasks:
                                                          2 sequential sub-tasks:
                                                            2 sequential sub-tasks:
                                                              2 sequential sub-tasks:
                                                                2 sequential sub-tasks:
                                                                  2 sequential sub-tasks:
                                                                    2 sequential sub-tasks:
                                                                      2 sequential sub-tasks:
                                                                        2 sequential sub-tasks:
                                                                          2 sequential sub-tasks:
                                                                            2 sequential sub-tasks:
                                                                              2 sequential sub-tasks:
                                                                                2 sequential sub-tasks:
                                                                                  2 sequential sub-tasks:
                                                                                    2 sequential sub-tasks:
                                                                                      2 sequential sub-tasks:
                                                                                      recommended policies were found for 'vpc-cni' addon, but since CIDR 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
2025-08-25 11:48:00 [x] creating add-on: vpc-cni
2025-08-25 11:48:00 [x] deploying stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:49:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster' to become ready
2025-08-25 11:50:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:51:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:52:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:53:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:54:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:55:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:56:00 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-node-cluster'
2025-08-25 11:57:00 [x] creating add-on: kube-proxy
2025-08-25 11:58:00 [x] successfully created add-on: kube-proxy
2025-08-25 11:59:00 [x] successfully created add-on: coredns
2025-08-25 11:59:50 [x] creating add-on: metrics-server
2025-08-25 11:59:50 [x] successfully created add-on: metrics-server
2025-08-25 11:59:51 [x] building managed nodegroup stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:51 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes' to become ready
2025-08-25 11:59:51 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:52 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:52 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:53 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:53 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:54 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:54 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:55 [x] waiting for CloudFormation stack 'eksctl-brain-tasks-ap-nodegroup-brain-tasks-nodes'
2025-08-25 11:59:55 [x] waiting for the control plane to become ready
2025-08-25 11:59:57 [x] saved kubeconfig as '/Users/prakash/.kube/config'
2025-08-25 12:00:00 [x] all EKS cluster resources for 'brain-tasks-app' have been created
2025-08-25 12:00:00 [x] node 'ip-192-168-36-13.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] node 'ip-192-168-36-14.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] node 'ip-192-168-36-15.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] node 'ip-192-168-36-16.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] node 'ip-192-168-36-17.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] node 'ip-192-168-36-18.ap-south-1.compute.internal' is ready
2025-08-25 12:00:00 [x] created 1 managed nodegroup(s) in cluster "brain-tasks-app"
2025-08-25 12:00:00 [x] to get the status of your cluster, run 'eksctl get clusters'
2025-08-25 12:00:00 [x] EKS cluster "brain-tasks-app" in "ap-south-1" region is ready
prakash@Mac: ~ %
```

4b) Verifying Nodes and user Profile

```
prakash@MacBook-Pro ~ % aws sts get-caller-identity --profile EKS_User
{
    "UserId": "AIDAB4BLu3VNMU2BXKC3Q",
    "Account": "928171332449",
    "Arn": "arn:aws:iam::928171332449:user/EKS_User"
}
prakash@MacBook-Pro ~ % kubectl get nodes
NAME                               STATUS   ROLES      AGE     VERSION
ip-192-168-36-143.ap-south-1.compute.internal   Ready   <none>    4h8m   v1.33.3-eks-3abbec1
ip-192-168-81-148.ap-south-1.compute.internal   Ready   <none>    4h8m   v1.33.3-eks-3abbec1
prakash@MacBook-Pro ~ %
```

5. Git Setup and config

5a) New Repo Created on Github



5b) Git Commands and pushing

```
prakash@mac brain_app_task % git init
Initialized empty Git repository in /Users/prakash/brain_app_task/.git/
prakash@mac brain_app_task % git remote add origin https://github.com/jpprakash4260/brain-app-task.git
prakash@mac brain_app_task % git add .
prakash@mac brain_app_task % git commit -m "1st commit"
[main (root-commit) 3e777ee] 1st commit
10 files changed, 179 insertions(+)
create mode 100644 .gitignore
create mode 100644 Dockerfile
create mode 100644 buildspec.yml
create mode 100644 dist/assets/index-BHGIHu50.js
create mode 100644 dist/assets/index-DFTLVRPB.css
create mode 100644 dist/index.html
create mode 100644 dist/vite.svg
create mode 100644 k8s/deployment.yml
create mode 100644 k8s/service.yml
create mode 100644 nginx.conf
prakash@mac brain_app_task % git pull origin main
remote: Enumerating objects: 180, done.
remote: Counting objects: 100 (delta 0), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 180 (delta 0), reused 2 (delta 0), pack-reused 177 (from 1)
Receiving objects: 100% (180/180), 119.00 KiB | 427.00 KiB/s, done.
Resolving deltas: 100% (180/180), done.
From https://github.com/jpprakash4260/brain-app-task
 * [new branch]    main      -> FETCH_HEAD
 * [new branch]    main      -> origin/main
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false # merge
hint:   git config pull.rebase true  # rebase
hint:   git config pull.ff only     # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: repository.
fatal: Need to specify how to reconcile divergent branches.
prakash@mac brain_app_task % git config pull.rebase true
prakash@mac brain_app_task % git push --set-upstream origin main
To https://github.com/jpprakash4260/brain-app-task.git
 ! [rejected]    main      -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/jpprakash4260/brain-app-task.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the local changes (e.g.
hint: 'git pull...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
prakash@mac brain_app_task % git push origin main
From https://github.com/jpprakash4260/brain-app-task
 * branch    main      -> FETCH_HEAD
dropping 3e777ee6fe380d53c259bc16517e7685989844d 1st commit -- patch contents already upstream
Successfully rebased and updated refs/heads/main.
prakash@mac brain_app_task % git push --set-upstream origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
prakash@mac brain_app_task %
```

6. Code Pipeline

6a) GitHub Connection Stage

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

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.

arnaws:codeconnections:ap-south-1:828171532440:connection/c41 X or Connect to GitHub

Repository name
Choose a repository in your GitHub account.

pprakash4260/brain-app-task 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

Output artifact format
Choose the output artifact format.

CodePipeline default AWS CodePipeline uses the default zip format for your artifact. This provides the most efficient storage and delivery. It also includes Git metadata about the repository.

Full clone AWS CodePipeline copies all metadata and history from the repository to each subsequent action to do a full Git clone. Only select actions support 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.

Cancel Previous Next

6b) Code Build Project

Continue to CodePipeline
Create a new CodeBuild build project and return to CodePipeline to finish configuring your pipeline.

Create build project

Project configuration

Project name
brain-task-app-build

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Project type
Select what type of project you would like to create. Info

Default project Create a custom CodeBuild project.

Runner project Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

Additional configuration Description, public build access, build badge, concurrent build limit, tags

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)

Amazon Linux

CloudShell Feedback

6c) Code Build Environment

The screenshot shows the AWS CodeBuild 'Additional configuration' section. It includes fields for provisioning model (On-demand or Reserved capacity), environment image (Managed image or Custom image), compute type (EC2 or Lambda), running mode (Container or Instance), operating system (Amazon Linux), runtime (Standard), image (aws/codebuild/amazonlinux-x86_64-standard:5.0), image version (Always use the latest image for this runtime version), and service role (New service role or Existing service role). A 'Role name' field contains 'codebuild-brain-task-app-build-service-role'. Below this is an 'Additional configuration' section with options for timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, and registry credential.

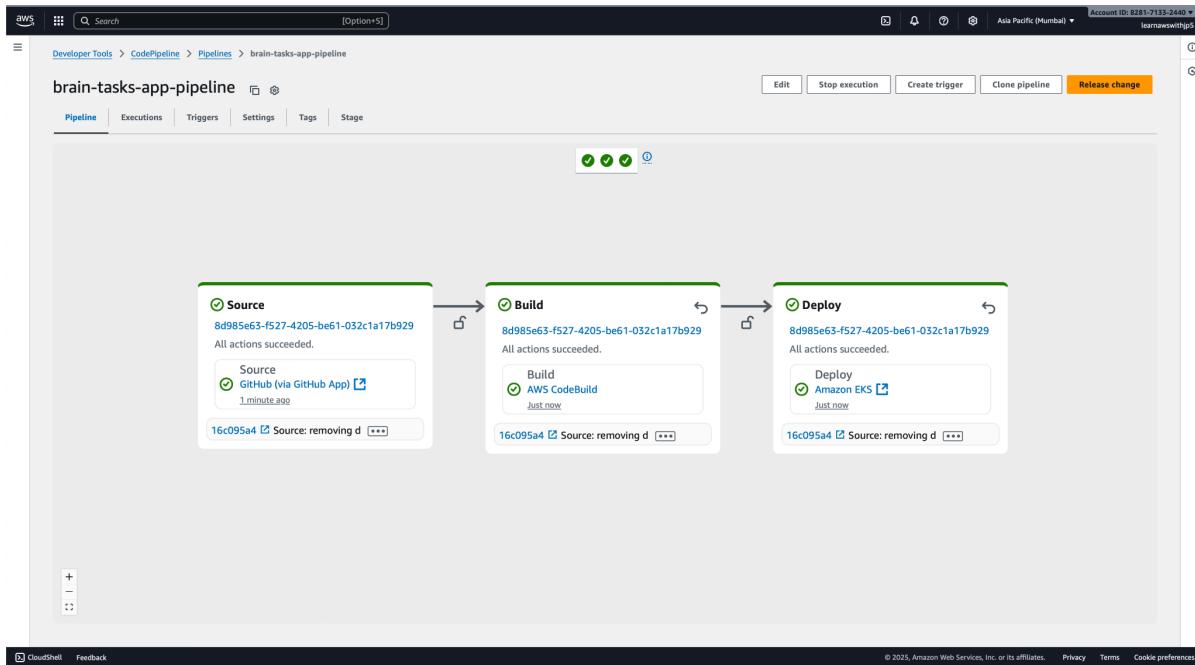
6d) Using buildspec.yml file

The screenshot shows the 'Buildspec' configuration section of the AWS CodeBuild project creation page. It includes options for 'Build specifications': 'Insert build commands' (Store build commands as build project configuration) and 'Use a buildspec file' (Store build commands in a YAML-formatted buildspec file). The 'Use a buildspec file' option is selected, with 'buildspec.yml' specified as the file path. Below this is a 'Batch configuration' section with an unchecked checkbox for 'Define batch configuration - optional'. The 'Logs' section includes 'CloudWatch' settings: 'CloudWatch logs - optional' (checked) and 'Group name - optional' (set to 'aws/codebuild/brain-task-app-build'). It also includes 'Stream name prefix - optional' (empty) and 'SS logs - optional' (unchecked). At the bottom are 'Cancel' and 'Continue to CodePipeline' buttons.

6e) Now Using Aws EKS for direct Deployment

The screenshot shows the AWS CodePipeline configuration interface for a 'Deploy' stage. The stage is set to use 'Amazon EKS' as the provider, with the region configured to 'Asia Pacific (Mumbai)'. An input artifact named 'BuildArtifact' is selected. The 'EKS Cluster Name' is set to 'brain-tasks-app'. The 'Deploy configuration type' is chosen as 'Kubectl'. Under 'Manifest file paths', the path 'k8s/deployment.yaml,k8s/service.yaml' is specified. The 'Kubernetes namespace - optional' field contains 'default'. In the 'Subnet IDs' section, several subnet IDs are listed: 'subnet-023c46e7c733753d2', 'subnet-0e8b87abf9d3aebb7', 'subnet-097d00490198b227f', 'subnet-0ace1fe71079d9780', 'subnet-02d36200fabd9f2aa', and 'subnet-0caac10b7a43e7050'. The 'Security group IDs' section lists 'sg-007672fe22debd652' and 'sg-0621798da33e7c4df'. The 'Environment variables - optional' section is empty. A checkbox for 'Configure automatic rollback on stage failure' is checked. A note at the bottom indicates 'Enable automatic retry on stage failure'. At the bottom right, there are 'Cancel', 'Previous', 'Skip deploy stage', and 'Next' buttons.

6f) Code Pipeline was Successfully Deployed

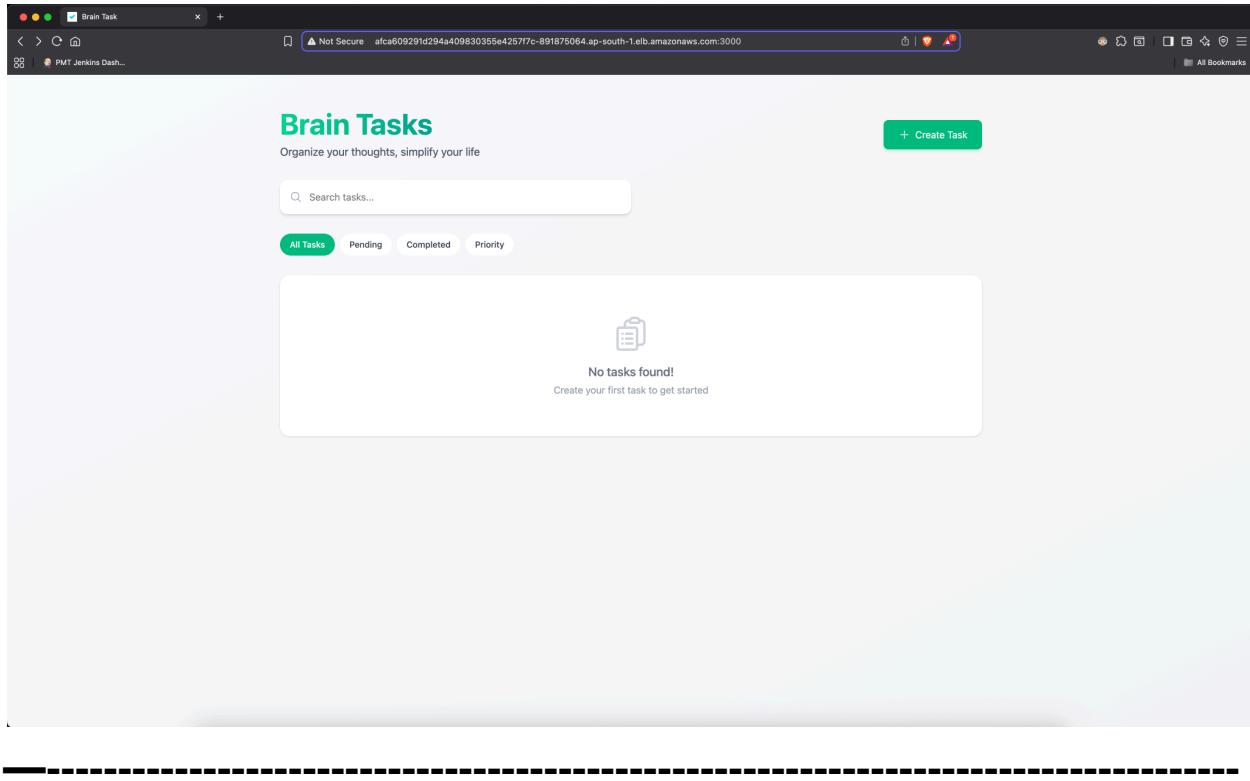


7. App Testing

7a) list pods and svc

```
prakash@mac ~ % kubectl get pods -n default -l app=brain-tasks-app
NAME           READY   STATUS    RESTARTS   AGE
brain-tasks-app-77d76b4585-dq9fw  1/1     Running   0          6h46m
brain-tasks-app-77d76b4585-pwcm   1/1     Running   0          6h45m
brain-tasks-app-77d76b4585-xptqg  1/1     Running   0          6h46m
prakash@mac ~ % kubectl get svc brain-tasks-app -n default -o yaml
apiVersion: v1
kind: Service
metadata:
  name: brain-tasks-app
spec:
  ports:
  - port: 31579
    protocol: TCP
    targetPort: 3000
  selector:
    app: brain-tasks-app
  type: LoadBalancer
status:
  loadBalancer:
    ip: 10.100.194.77
    ingress:
      - ip: afca609291d294a409830355e4257f7c-891875064.ap-south-1.elb.amazonaws.com
prakash@mac ~ %
```

7b) Testing App with External IP on 3000 port



8. Monitoring

8a) CloudWatch Log

The screenshot shows the AWS CloudWatch Logs console. On the left, there's a navigation sidebar with links like CloudWatch, Dashboards, Alarms, Logs, Metrics, Application Signals, Network Monitoring, and Insights. The main area is titled '/aws/codebuild/brain-task-app-build' under 'Log group details'. It shows metrics like ARN (arn:aws:logs:ap-south-1:828171332440:log-group:/aws/codebuild/brain-task-app-build*), Creation time (5 days ago), Retention (Never expire), and Stored bytes (300.9 kB). On the right, sections include Metric filters, Subscription filters, Contributor Insights rules, KMS key ID, Anomaly detection, Data protection (with Sensitive data count), and Custom field indexes. Below this is a table titled 'Log streams (97)' with columns for Log stream and Last event time. The table lists several log stream names and their corresponding last event times.

8b) While Code Build Project Creating Enabling CloudWatch Logs

The screenshot shows the 'Create build project' wizard on the AWS CodeBuild console. In the 'Artifacts' section, there's a note about defining batch configuration. In the 'Logs' section, under 'CloudWatch', the 'CloudWatch logs - optional' checkbox is checked. This checkbox is described as uploading build output logs to CloudWatch. Below it, there's a 'Group name - optional' input field containing 'aws/codebuild/brain-tasks-build', a note about the default log group name, and a 'Stream name prefix - optional' input field containing 'S3'. A note below it says 'S3 logs - optional' and describes it as uploading build output logs to S3. At the bottom of the page, there are 'Cancel' and 'Create build project' buttons.