

FINAL PROJECT – 1

MindTrack

SCREENSHOT IMAGES

1] docker build

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

Sending build context to Docker daemon 323.6kB
Step 1/5 : FROM nginx:alpine
alpine: Pulling from library/nginx
1074353eec0d: Pulling fs layer
25f453064fd3: Pulling fs layer
567f84da6fb0: Pulling fs layer
da7c973d8b92: Pulling fs layer
33f95a0f3229: Pulling fs layer
085c5e5aaa8e: Pulling fs layer
0abf9e567266: Pulling fs layer
de54cb821236: Pulling fs layer
33f95a0f3229: Waiting
da7c973d8b92: Waiting
085c5e5aaa8e: Waiting
0abf9e567266: Waiting
de54cb821236: Waiting
567f84da6fb0: Verifying Checksum
567f84da6fb0: Download complete
1074353eec0d: Verifying Checksum
1074353eec0d: Download complete
25f453064fd3: Verifying Checksum
25f453064fd3: Download complete
1074353eec0d: Pull complete
25f453064fd3: Pull complete
```

2] docker image, run

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

3] ECR repo creation

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

4] image push to ECR from docker

```
ubuntu@ip-172-31-4-56:~/Project-1$ aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
ubuntu@ip-172-31-4-56:~/Project-1$ docker push 045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app:latest
The push refers to repository [045998146012.dkr.ecr.ap-south-1.amazonaws.com/brain-task-app]
93f4306c3a98: Pushed
1d30230573f3: Pushed
e6fe11fa5b7f: Pushed
67ea0b046e7d: Pushed
ed5fa8595c7a: Pushed
8ae63eb1f31f: Pushed
b3e3dlbbb64d: Pushed
48078b7e3000: Pushed
fd1e40d7f74b: Pushed
7bb20cf5ef67: Pushed
latest: digest: sha256:f6770b4c6758db269f2aedc2f9d5bc26d0b5a4301da849667a2092124d1a5ae9 size: 2406
ubuntu@ip-172-31-4-56:~/Project-1$
```

5] ECR repo in aws

The screenshot shows the AWS ECR console interface. On the left, there's a sidebar with navigation options like 'Amazon Elastic Container Service', 'Private registry', 'Public registry', and links to 'ECR public gallery', 'Amazon ECS', and 'Amazon EKS'. The main area is titled 'brain-task-app' and has tabs for 'Summary', 'Images', and 'Repository tags'. Under 'Images', it says '(1) Info' and shows a table with one row. The table columns are 'Image tags', 'Type', 'Created at', 'Image size', 'Image digest', and 'Last pulled at'. The single entry is 'latest' (Image type), created on '30 December 2025, 15:05:52 (UTC+05.5)', with a size of '23.10' and a digest of 'sha256:f6770...'. There are buttons for 'Delete', 'Copy URI', 'Details', 'Scan', and 'View push commands'.

6] EKS cluster creation

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

7] EKS cluster and node

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

8] EKS deployment and service(LB)

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

9] EKS dash in aws

The screenshot shows the EKS dashboard for the 'brain-cluster'. On the left sidebar, there are sections for Dashboard, Clusters (selected), Settings, Amazon EKS Anywhere, and Related services (Amazon ECR, AWS Batch). Below the sidebar is a 'Documentation' link. The main content area displays the cluster's name 'brain-cluster' and its status as 'Active'. It includes a warning message about the end of standard support for version 1.32 on March 23, 2026. Below this, there are four cards: Cluster health (0), Upgrade insights (5), Node health issues (0), and Capability issues (0). A navigation bar at the bottom includes Overview, Resources, Compute, Networking, Add-ons (1), Capabilities, Access, and Observability. At the very bottom, there are links for CloudShell, Feedback, and Console Mobile App.

10] code builts in aws

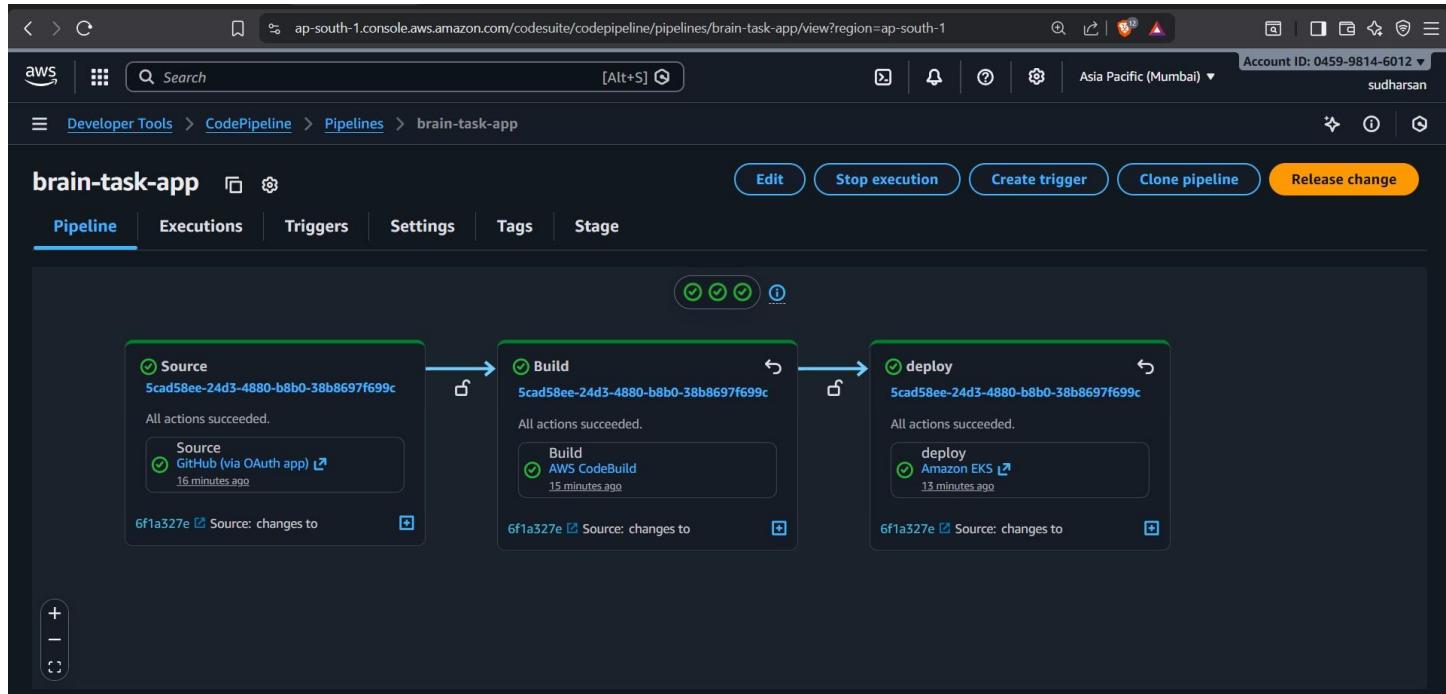
The screenshot shows the CodeBuild dashboard for the 'brain-task-app-build' project. The left sidebar lists Developer Tools, CodeBuild (selected), Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild, Getting started, Build projects, Build project, Settings, Build history, Report groups, Report history, Compute fleets New, Account metrics), and Related integrations (Jenkins). The main content area shows a green notification bar stating 'Build started' with the message 'You have successfully started the following build: brain-task-app-build:e9f337ed-f03f-4caf-946a-8b2789087f14'. Below this, there is a summary card for the build: 'brain-task-app-build:e9f337ed-f03f-4caf-946a-8b2789087f14'. It shows the build status as 'Succeeded', initiator as 'root', resolved source version as '949ecf100539985a8dd65bd73deb1627502b6d3 5', start time as 'Dec 30, 2025 6:03 PM (UTC+5:30)', and end time as 'Dec 30, 2025 6:04 PM (UTC+5:30)'. There are buttons for Stop build, Debug build (highlighted in blue), and Retry build. Below the summary are tabs for Build logs, Phase details (selected), Reports, Environment variables, Build details, and Resource utilization. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

11] code build logs in aws

The screenshot shows the AWS CodeBuild console for a project named "brain-task-app-build". The "Phase details" tab is selected, displaying a table of build phases and their outcomes:

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

12] code pipeline



13] codepipeline logs in aws cloudwatch

The screenshot shows the AWS CloudWatch Log Management interface. The left sidebar is titled "CloudWatch" and includes sections for "AI Operations", "GenAI Observability", "Application Signals (APM)", "Infrastructure Monitoring", and "Logs". Under "Logs", "Log Management" is selected. The main area is titled "Log events" and displays a table of log entries. The columns are "Timestamp" and "Message". The messages show the container performing tasks such as expanding base directory paths, assembling file lists, and reporting auto-discovery results. One message indicates a phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED.

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

The screenshot shows the AWS CloudWatch Log Management interface, similar to the previous one but for a different task. The left sidebar shows the same navigation options. The main area is titled "Log events" and displays a table of log entries. The messages show the container running commands, checking for errors, and reporting phase completion. One message indicates a phase complete: BUILD State: FAILED.

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

14] EKS cluster log in aws cloudwatch

The screenshot shows the AWS CloudWatch Log Management interface for an EKS cluster named 'brain-cluster'. The left sidebar lists various monitoring categories like AI Operations, GenAI Observability, Application Signals (APM), Infrastructure Monitoring, and Logs, with 'Log Management' selected. The main area displays a table of log events with columns for Timestamp and Message. The first few messages are audit logs from the kube-apiserver, indicating requests for 'audit.k8s.io/v1' with various levels (Request, Metadata) and audit IDs. A search bar at the top allows filtering by search terms. Buttons for Actions, Start tailing, Create metric filter, and Display are available at the top right. A 'Back to top' button is located at the bottom right of the log table.

15] loadbalancer service

The screenshot shows the AWS CloudWatch Events interface. The left sidebar shows the 'Events' section. The main area displays a table of events with columns for Type, Reason, Age, From, and Message. The events listed are related to the 'brain-task-deployment' and 'brain-task-service'. The deployment was scaled up from 0 to 1 replica, and the service was created with an external IP of 10.100.56.167. Subsequent events show the deployment being deleted and then recreated with a new IP address of 10.100.238.64. The service was also deleted and then recreated. The logs are timestamped and show the progression of these operations over approximately 20 hours.

16] brain-task -app

The screenshot shows a web browser window with the URL `35.154.113.106:3000`. The page title is "Brain Tasks" with the subtitle "Organize your thoughts, simplify your life". A search bar contains the placeholder "Search tasks...". Below the search bar are four filter buttons: "All Tasks" (highlighted in green), "Pending", "Completed", and "Priority". In the center, there is a large white box with a clipboard icon and the text "No tasks found! Create your first task to get started".

