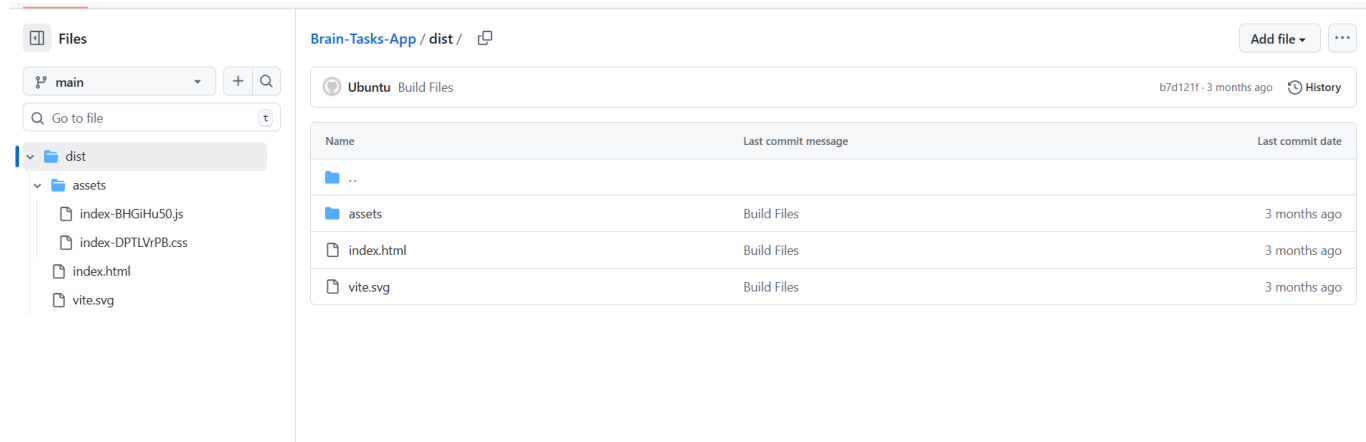
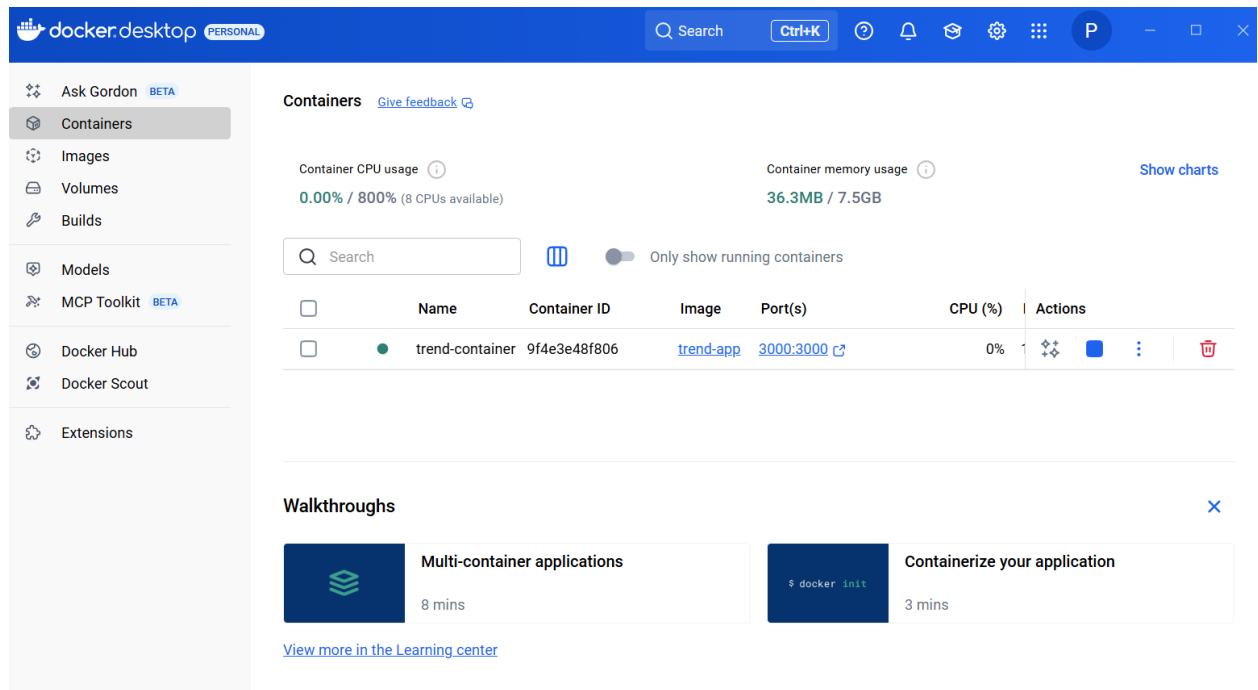


Project Trend-App

➔ Step 1: Preparation — repo + local workspace



➔ Step 2: Dockerize the React app and build the image



➔ Step 3: Terraform infra

Purpose: provision VPC, subnets (2 AZs), IGW, route tables, security groups, EC2 for Jenkins, IAM roles, EKS cluster and node group.

Files: main.tf, eks.tf

➔ Step 4: Jenkins EC2 provisioning and configuration

In Jenkins UI:

- Install required plugins: Docker, Git, Kubernetes, Pipeline.
- Add credentials: DockerHub username/password, GitHub token

➔ Step 5: Create EKS cluster & worker nodegroup and Deploy Kubernetes manifests and Install AWS Load Balancer Controller

The screenshot displays the AWS Management Console interface for an Elastic Load Balancing (ELB) Application Load Balancer. The breadcrumb navigation at the top shows the path: EC2 > Load balancers > k8s-trendapp-trending-c80c7ce2ef. The left-hand navigation pane is expanded to 'Load Balancing' > 'Load Balancers'. The main content area shows the details for the load balancer 'k8s-trendapp-trending-c80c7ce2ef'. A 'DNS name copied' tooltip is visible over the DNS name field.

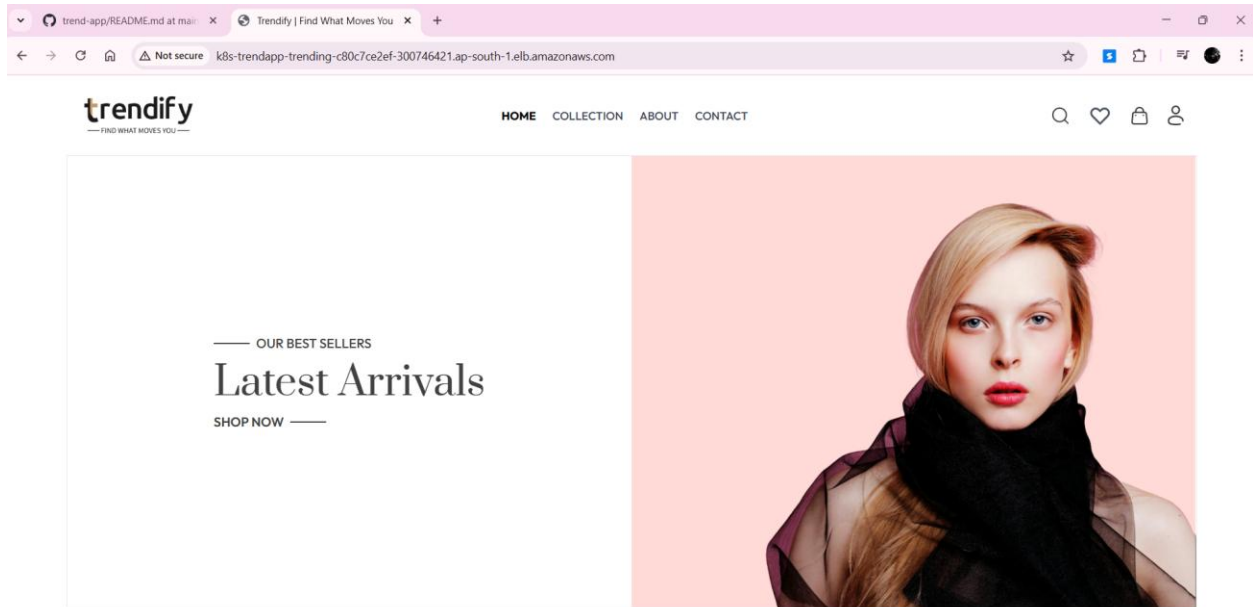
k8s-trendapp-trending-c80c7ce2ef			
Details			
Load balancer type Application	Status Active	VPC vpc-0164bc29bcca5f5ca	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0822c1ba4c725ab6a ap-south-1a (aps1-az1) subnet-0ba6c49d0da6d4aa8 ap-south-1a (aps1-az3)	Date created September 17, 2025, 21:21 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:907969929387:loadbalancer/app/k8s-trendapp-trending-c80c7ce2ef/4daf9ddd2c928059		DNS name k8s-trendapp-trending-c80c7ce2ef-300746421.ap-south-1.elb.amazonaws.com (A Record)	

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

➔ Step 6: Test the application in the cluster



➔ Step 7: Auto-scaling and load-testing

```
PS C:\Users\91866\projects\Trend\k8s> kubectl get hpa -n trend-app
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
trend-app-hpa	Deployment/trend-app-deployment	cpu: 15%/30%	2	5	2	25h

➔ Step 8: Push the Project to GitHub & Terminate the AWS resources to avoid costs.