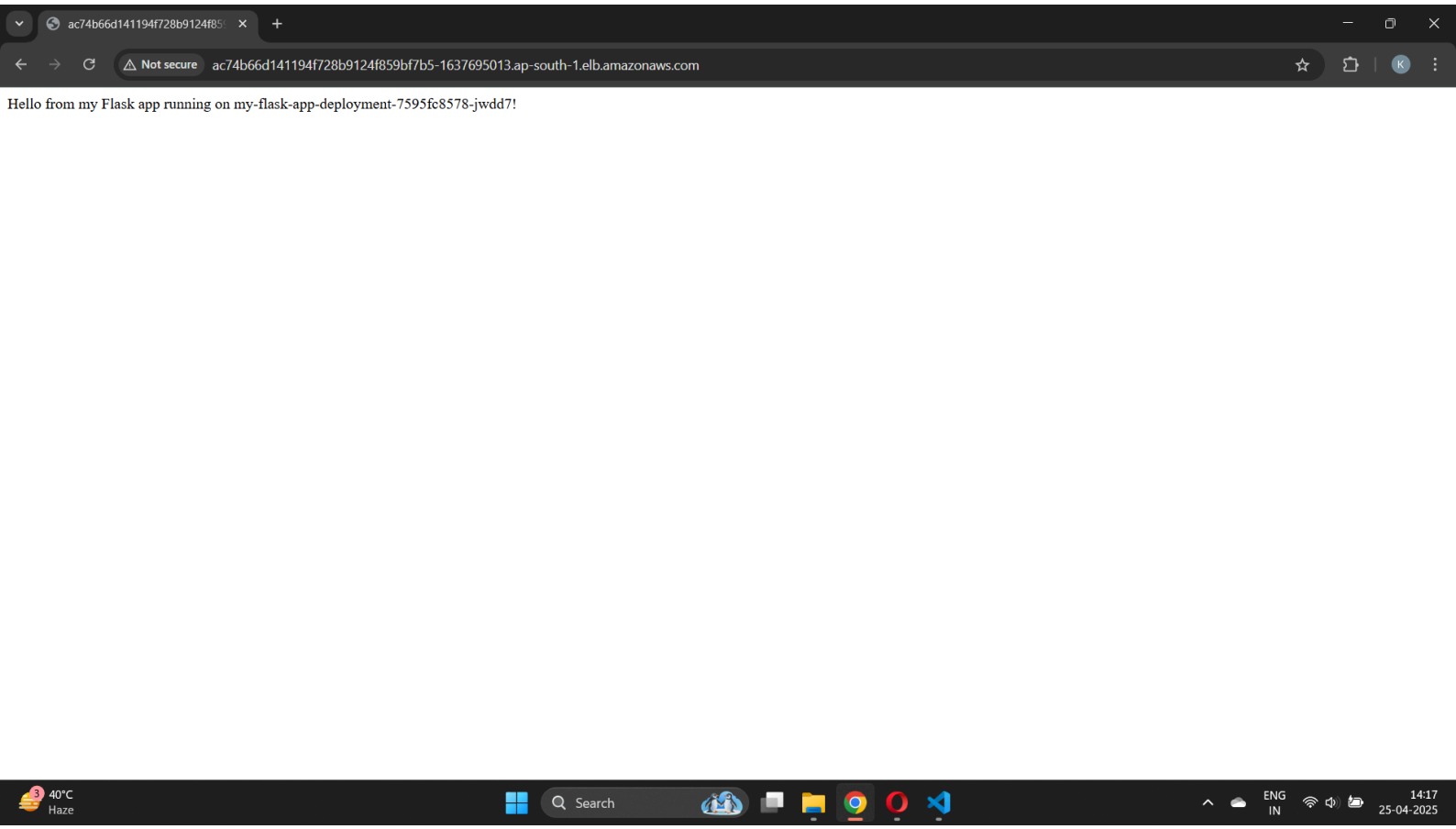
# Step-by-Step Summary

Here's a concise summary of the steps you've taken:

1. **Developed a Simple Flask Application**: You created the basic Python code (app.py) and its dependencies (requirements.txt).
2. **Containerized the Application with Docker**: You wrote a Dockerfile defining how to build a Docker image for your Flask app.
3. **Built the Docker Image**: You used the docker build command to create a Docker image from your Dockerfile.
4. **Pushed the Docker Image to Docker Hub**: You tagged and pushed your Docker image to a public container registry, making it accessible to your Kubernetes cluster.
5. **Created an Amazon EKS Cluster**: You used eksctl to provision a managed Kubernetes cluster on AWS.
6. **Configured kubectl**: You verified that your local kubectl command-line tool was correctly set up to communicate with your EKS cluster.
7. **Defined a Kubernetes Deployment**: You created a deployment.yaml file to tell Kubernetes how to run and manage multiple instances (Pods) of your application.
8. **Applied the Deployment**: You used kubectl apply to create the Deployment in your EKS cluster, which started the desired number of application Pods.
9. **Defined a Kubernetes Service**: You created a service.yaml file to expose your application running in the Pods to the outside world using a LoadBalancer.
10. **Applied the Service**: You used kubectl apply to create the Service in your EKS cluster, which provisioned an AWS Load Balancer.
11. **Accessed the Application**: You retrieved the external IP address or DNS name of the Load Balancer and successfully accessed your running Flask application in a web browser.

.


ac74b66d141194f728b9124f85
Not secure
ac74b66d141194f728b9124f859bf7b5-1637695013.ap-south-1.elb.amazonaws.com
Hello from my Flask app running on my-flask-app-deployment-7595fc8578-jwdd7!


40°C
Haze
Search
ENG
IN
14:17
25-04-2025