#### **Project Overview**

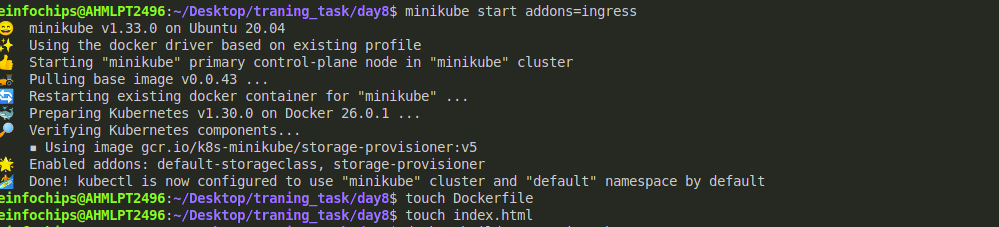
Participants are required to deploy a simple static web application on a Kubernetes cluster using Minikube, set up advanced ingress networking with URL rewriting and sticky sessions, and configure horizontal pod autoscaling to manage traffic efficiently. The project will be divided into stages, with each stage focusing on specific aspects of Kubernetes ingress, URL rewriting, sticky sessions, and autoscaling.

#### **Requirements and Deliverables**

### **Stage 1: Setting Up the Kubernetes Cluster and Static Web App**

1. **Set Up Minikube:**
   * Ensure Minikube is installed and running on the local Ubuntu machine.
   * Verify the Kubernetes cluster is functioning correctly.

Minikube start –addons=ingress

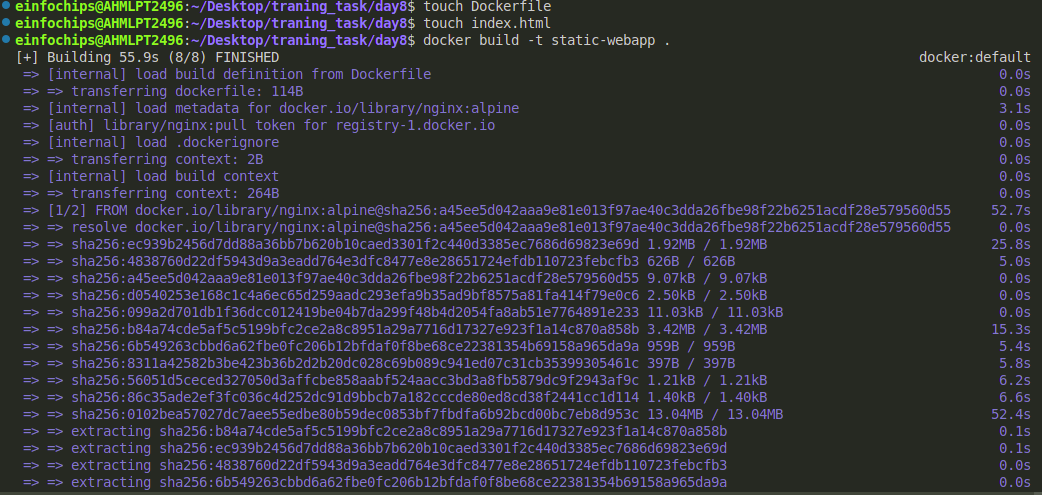


1. **Deploy Static Web App:**
   * Create a Dockerfile for a simple static web application (e.g., an HTML page served by Nginx).
   * Build a Docker image for the static web application.
   * Push the Docker image to Docker Hub or a local registry.

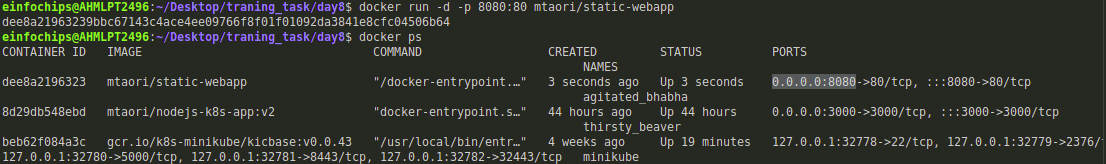
Touch Dockerfile

Touch index.html

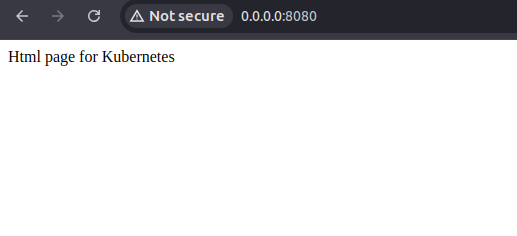
Docker build -t static-webapp



Check weather the docker image is running locally or not for this

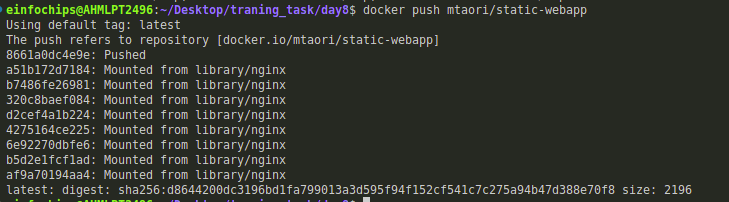


Docker run -p -d 8080:80 mtaori/static-webapp:latest

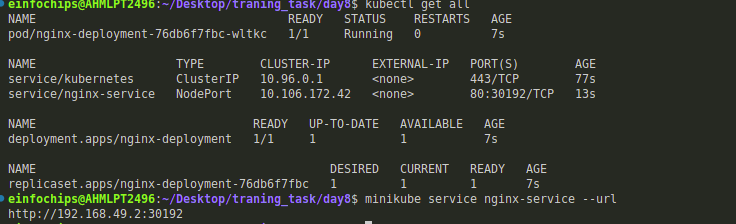
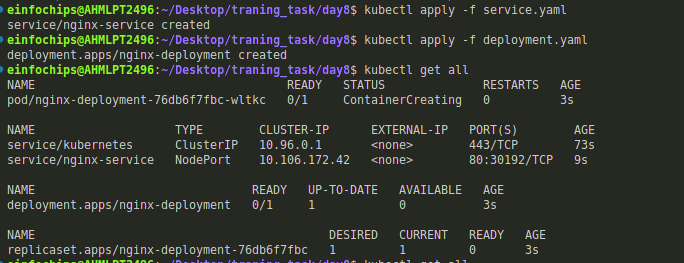


Push you docker image in docker hub because kubernetes pull the image from docker image from hub.

Docker pull mtaori/static-webapp



1. **Kubernetes Deployment:**
   * Write a Kubernetes deployment manifest to deploy the static web application.
   * Write a Kubernetes service manifest to expose the static web application within the cluster.
   * Apply the deployment and service manifests to the Kubernetes cluster.

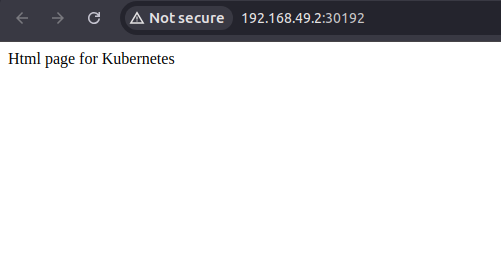


After apply all the file by:

Kubectl apply -f .

This will deploy all the files.

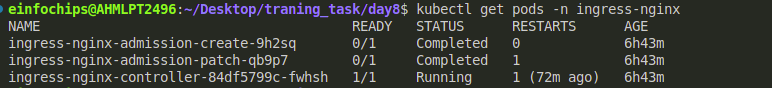
Screenshot of output:



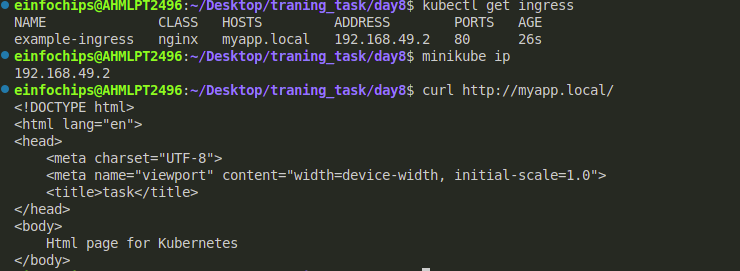
### 

### **Stage 2: Configuring Ingress Networking**

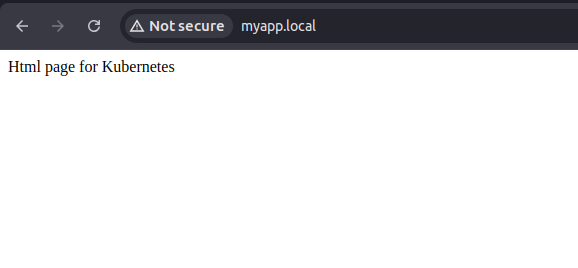
1. **Install and Configure Ingress Controller:**
   * Install an ingress controller (e.g., Nginx Ingress Controller) in the Minikube cluster.
   * Verify the ingress controller is running and accessible.



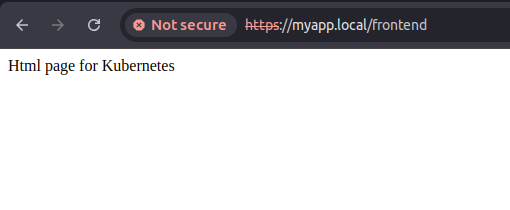
1. **Create Ingress Resource:**
   * Write an ingress resource manifest to route external traffic to the static web application.
   * Configure advanced ingress rules for path-based routing and host-based routing (use at least two different hostnames and paths).
   * Implement TLS termination for secure connections.

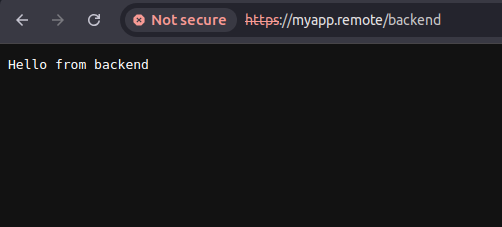


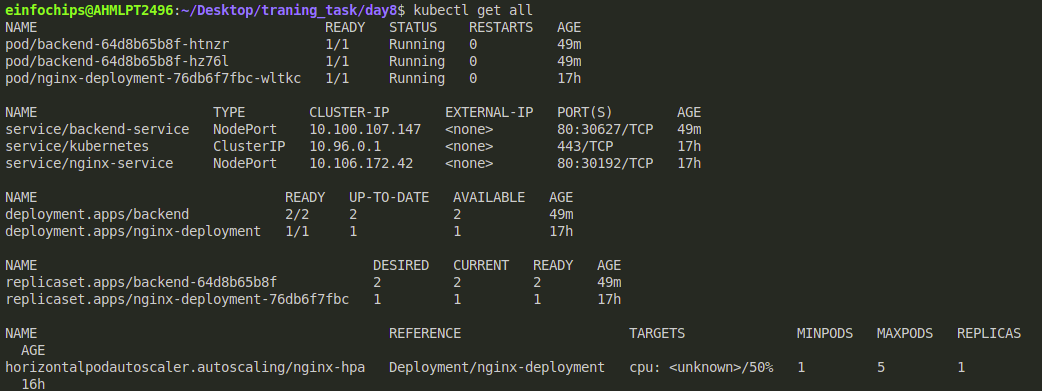
Now type myapp.local on any browser:



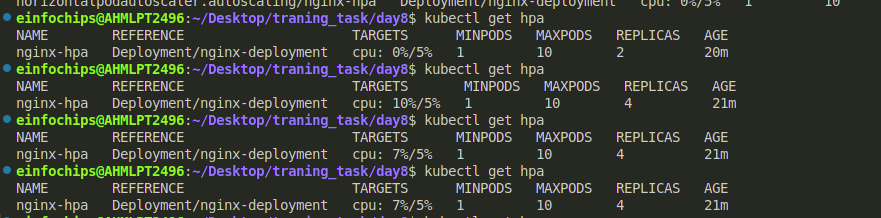
* + Configure URL rewriting in the ingress resource to modify incoming URLs before they reach the backend services.
  + Enable sticky sessions to ensure that requests from the same client are directed to the same backend pod.







1. **Stress Testing:**
   * Perform stress testing to simulate traffic and validate the HPA configuration.
   * Monitor the scaling behavior and ensure the application scales up and down based on the load.



When we hit the kubectl get pods then shows:  
