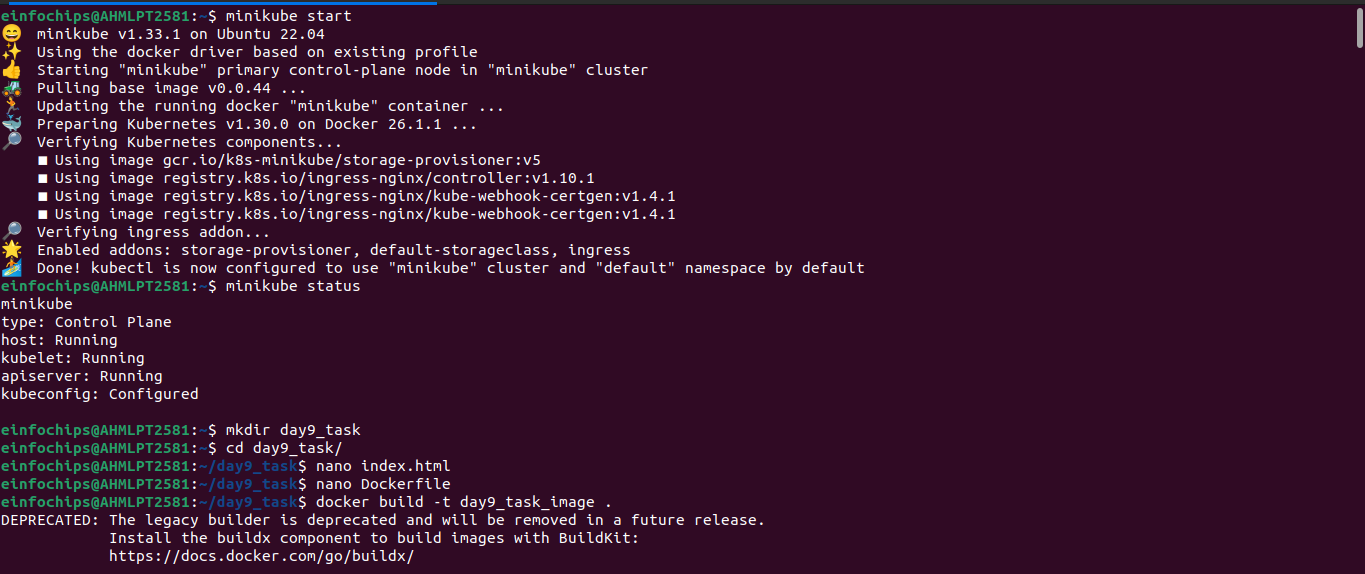
### **Stage 1: Setting Up the Kubernetes Cluster and Static Web App**

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

* Start Minikube (minikube start)
* Create a directory named static-web-api in the current working directory (mkdir static-web-api)

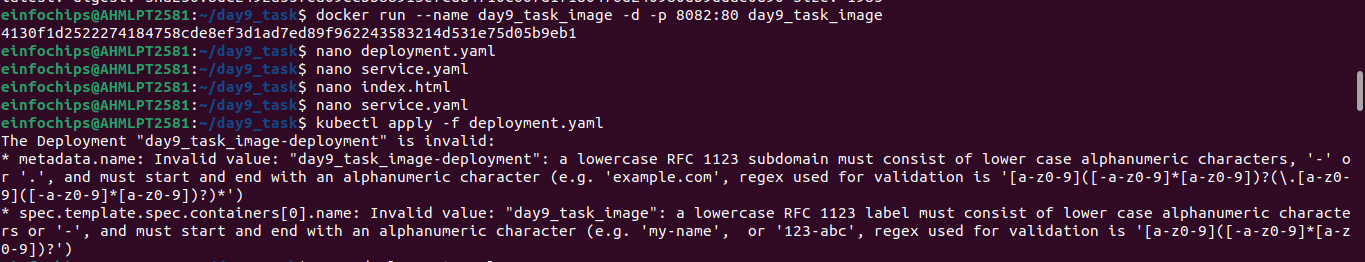
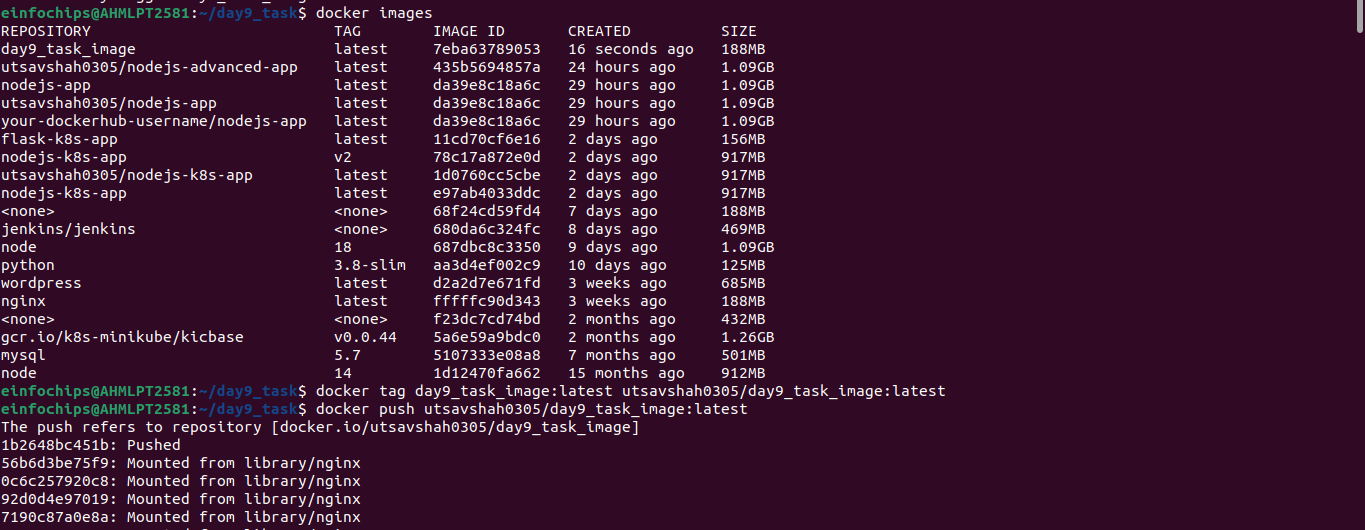
.

Create a file deployment.yaml

Create a file service.yaml

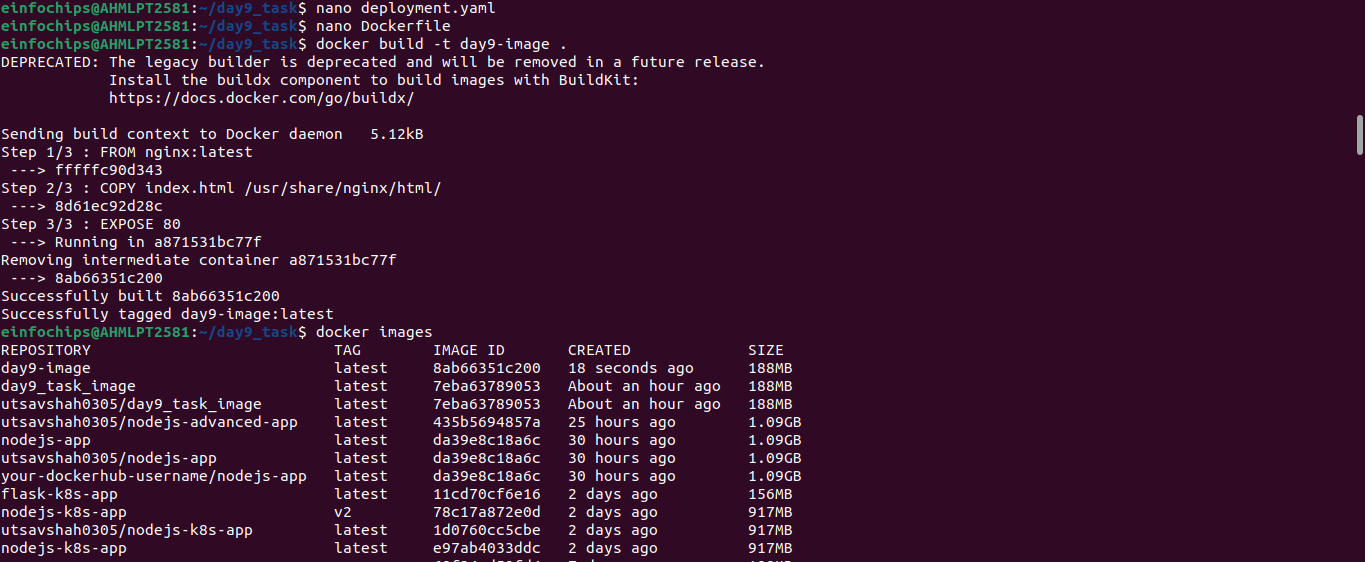
Apply the Deployment and Service Manifests

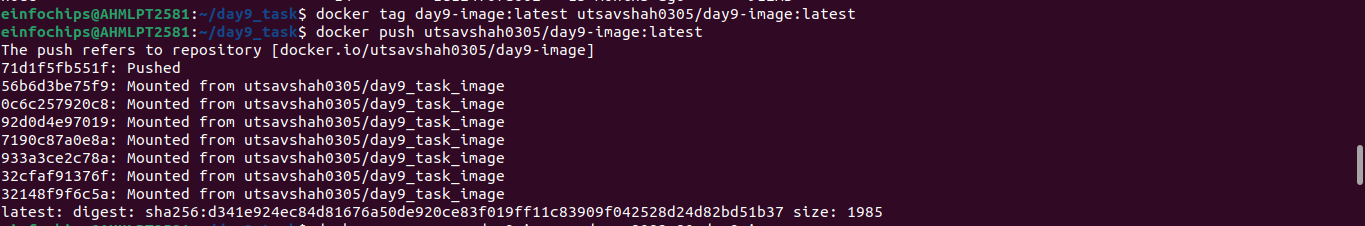
Minikube IP address

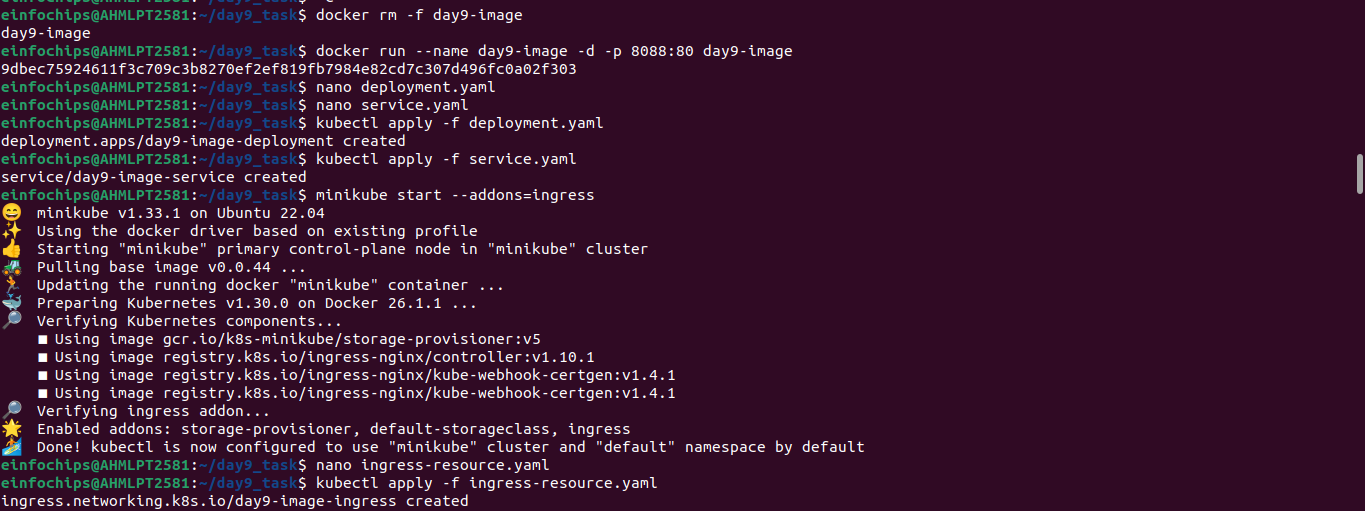
.

**2. Deploy Static Web App:**

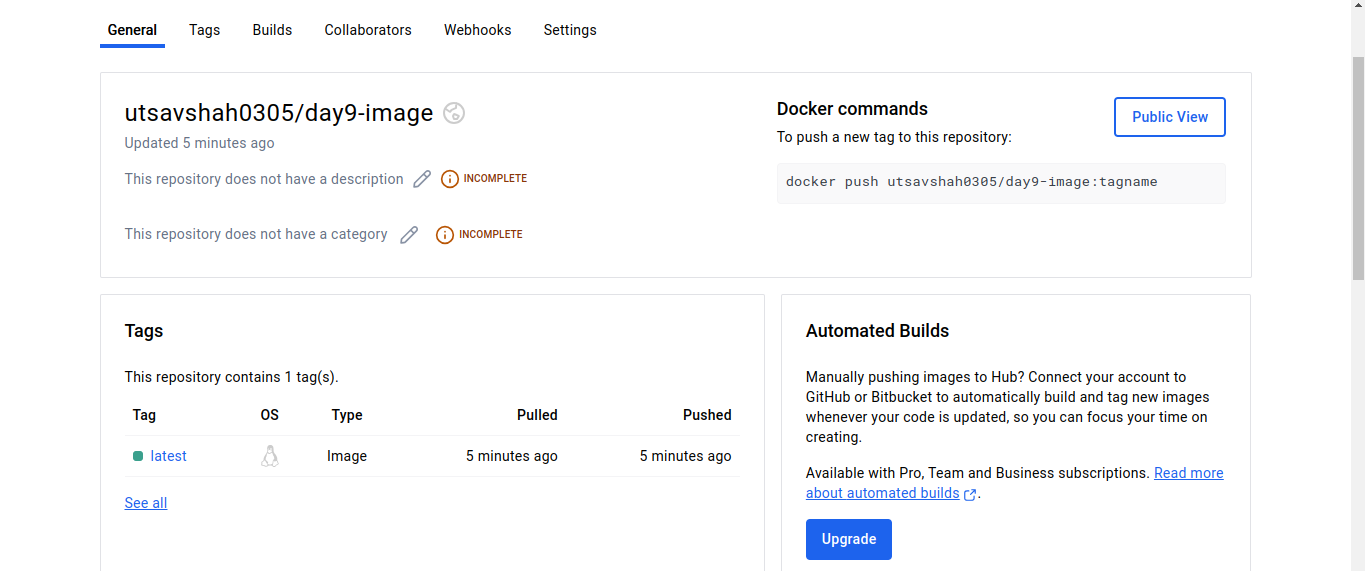
* + Create a Dockerfile for a simple static web application (e.g., an HTML page served by Nginx).
  + Build a Docker image for the static web application.
  + Push the Docker image to Docker Hub or a local registry.

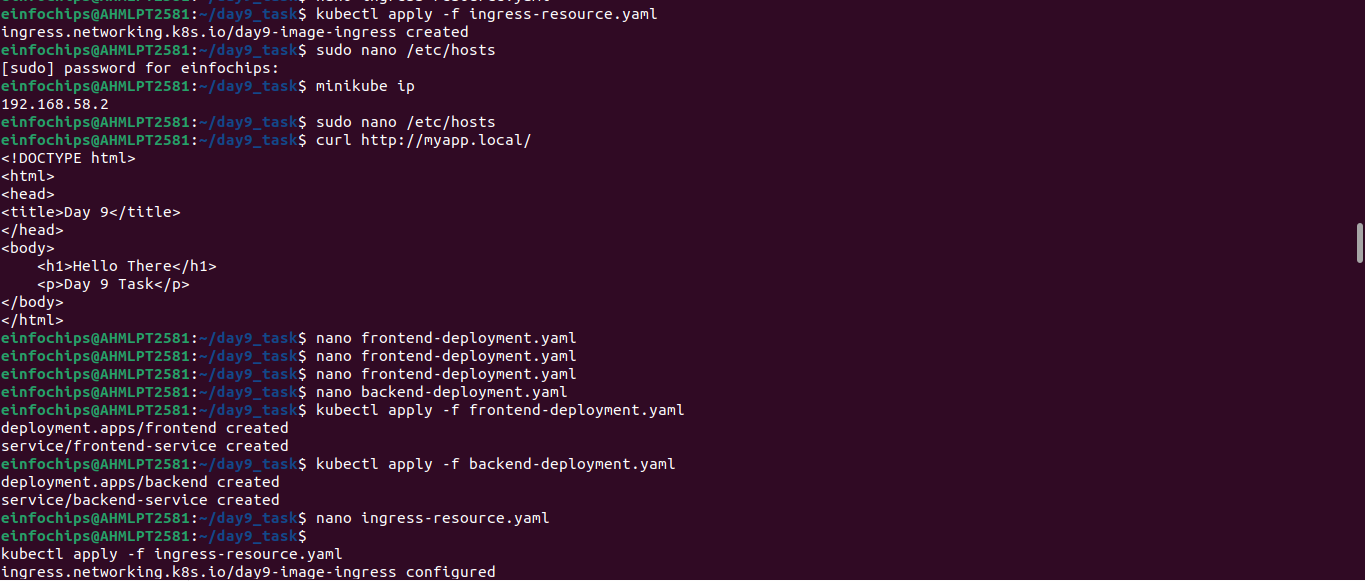
.

.

.

**3. Kubernetes Deployment:**

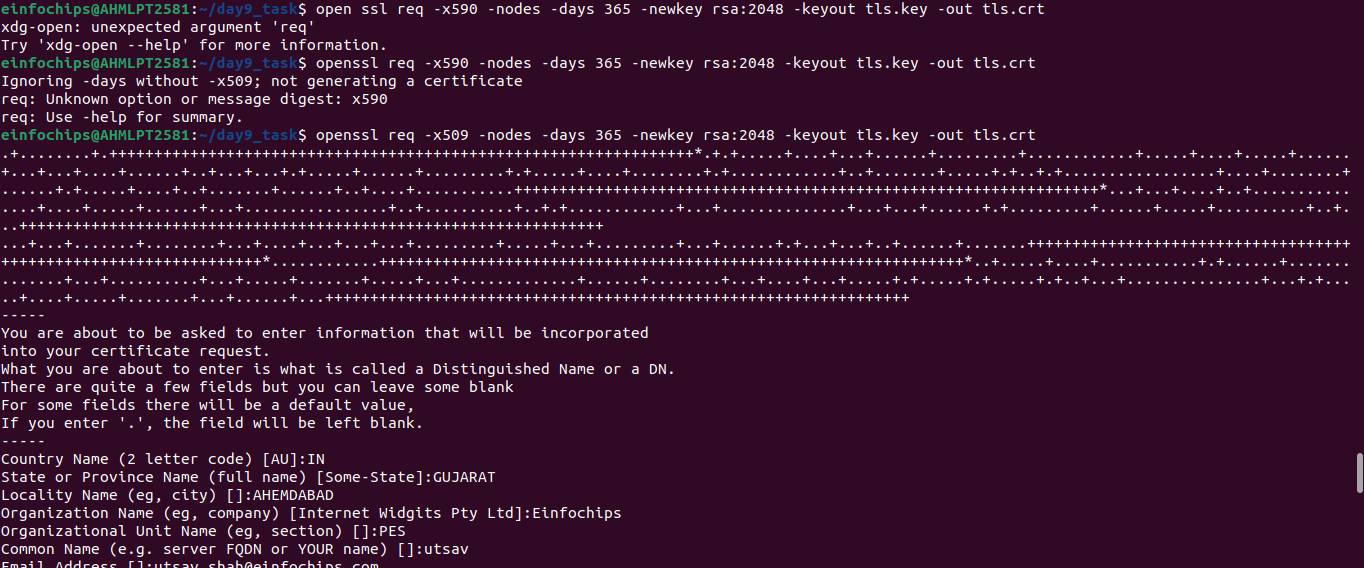
* + Write a Kubernetes deployment manifest to deploy the static web application.
  + Write a Kubernetes service manifest to expose the static web application within the cluster.
  + Apply the deployment and service manifests to the Kubernetes cluster.
  + 

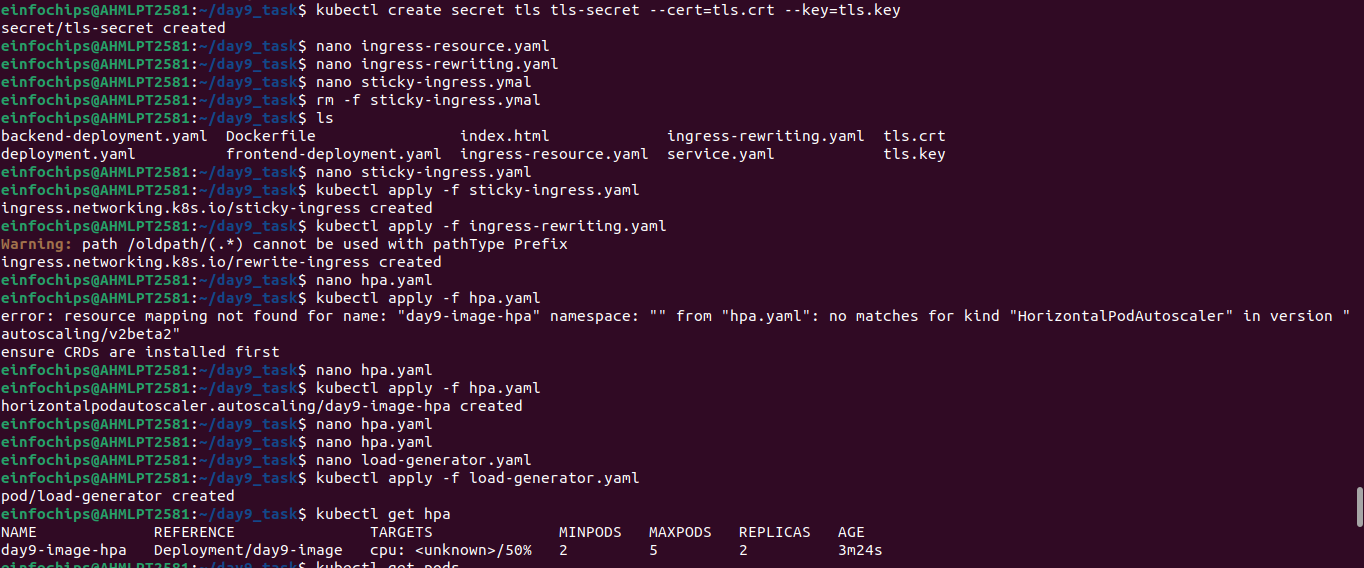
.

.

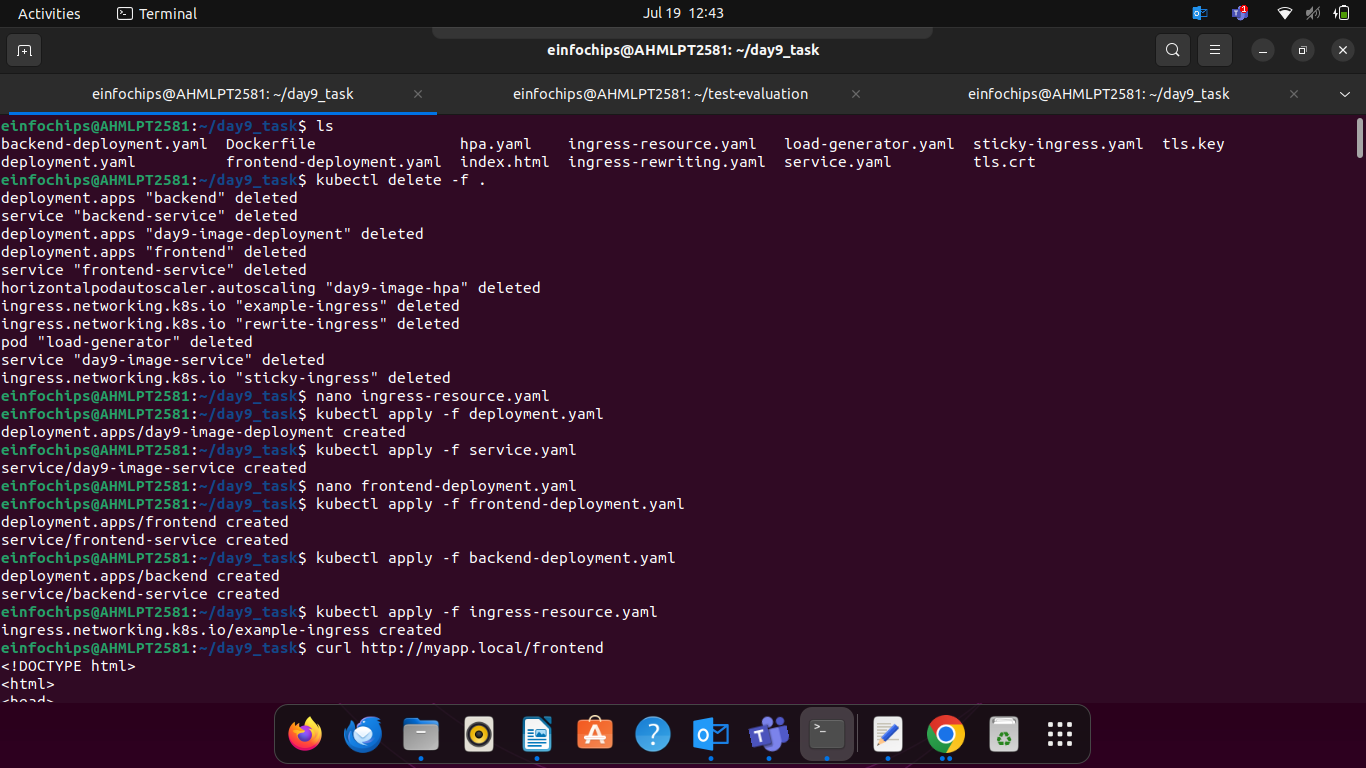
1. **Create Ingress Resource:**
   * Write an ingress resource manifest to route external traffic to the static web application.
   * Configure advanced ingress rules for path-based routing and host-based routing (use at least two different hostnames and paths).
   * Implement TLS termination for secure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

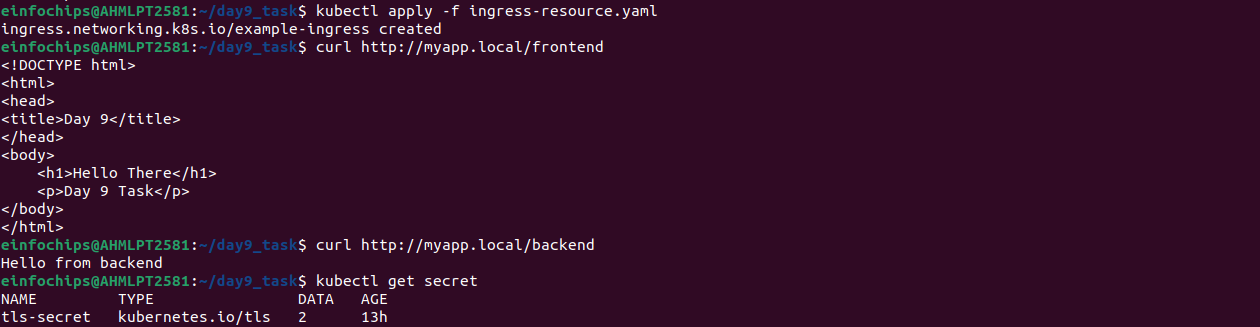
* minikube addons enable ingress
* Create a Kubernetes Secret to store the TLS certificate
* kubectl create secret tls tls-secret --cert=tls.crt –key=tls.key
* nano ingress-rewriting.yaml
* kubectl apply f ingressrewring.yaml
* Create a ingress-resourse.yaml
* Apply the Ingress-resourse.yaml
* Create a deployment.yaml
* Create a service.yaml

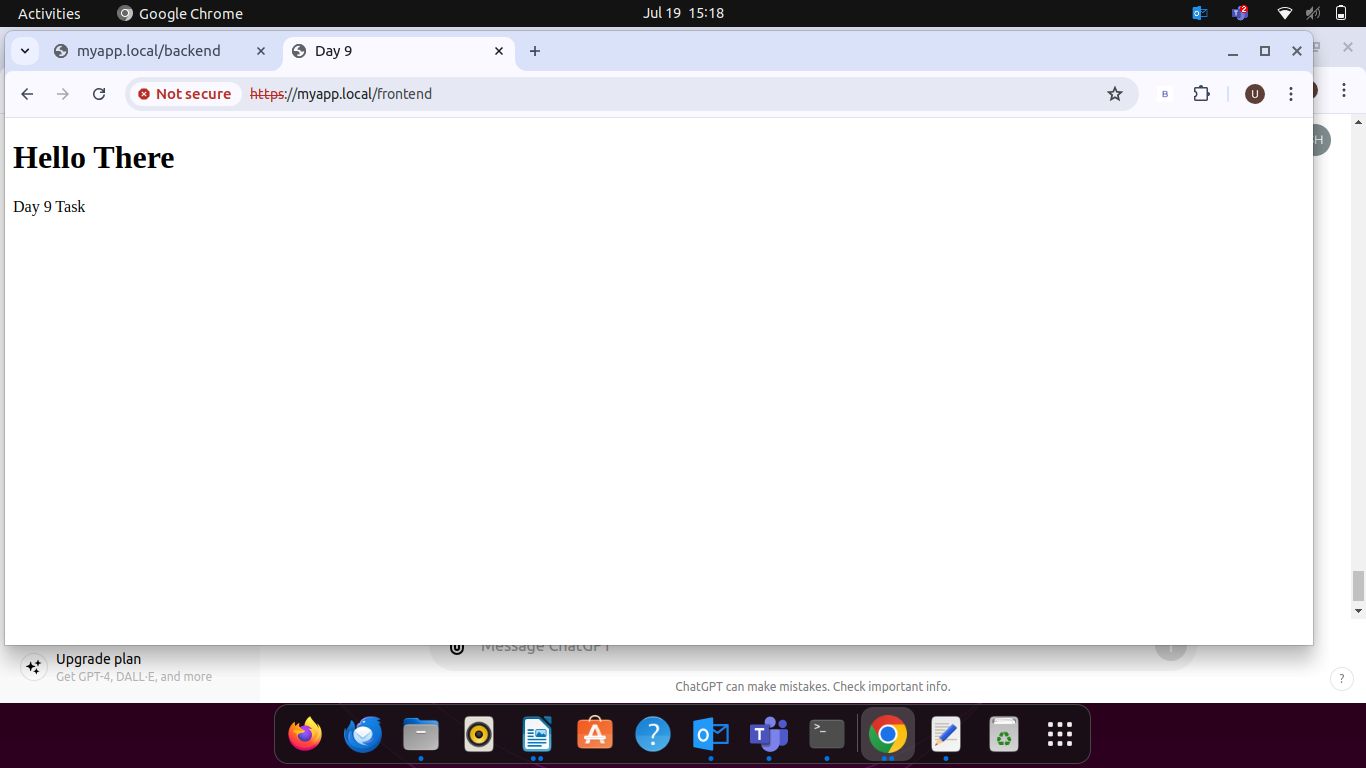
.

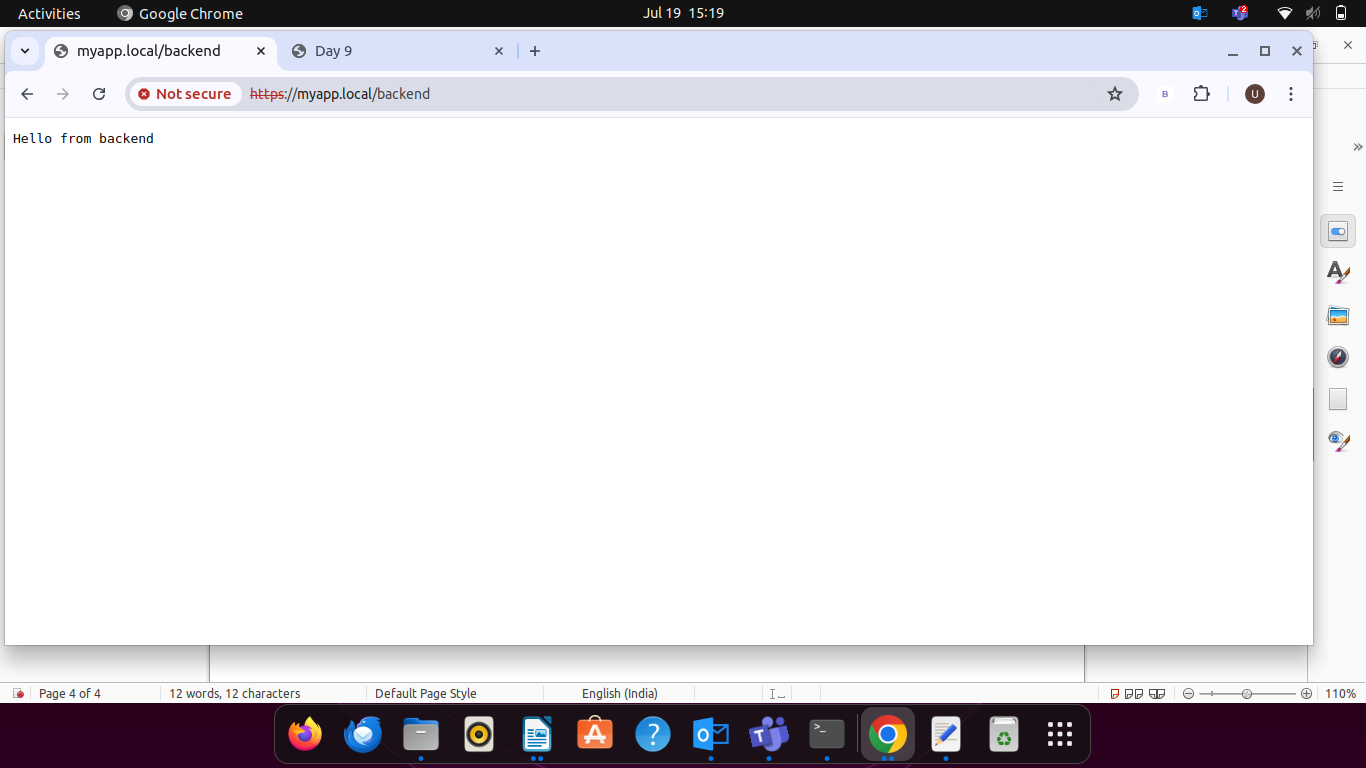
.

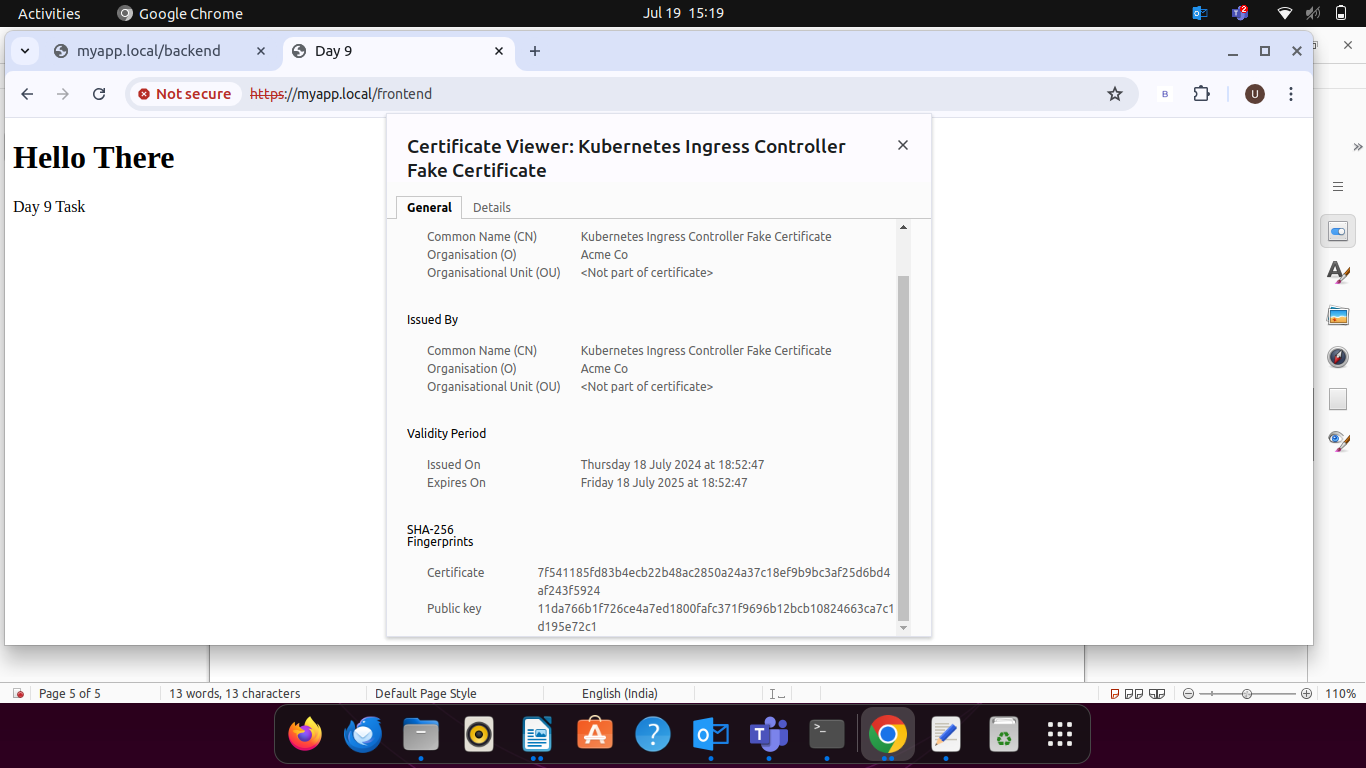
1. **Configure Horizontal Pod Autoscaler:**
   * Write a horizontal pod autoscaler (HPA) manifest to automatically scale the static web application pods based on CPU utilization.
   * Set thresholds for minimum and maximum pod replicas.

.

.

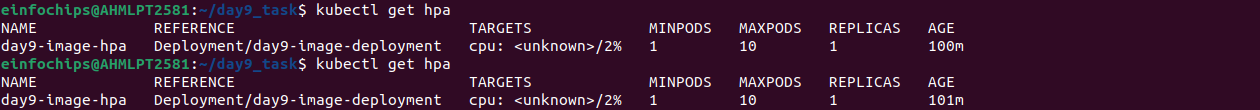
.

.

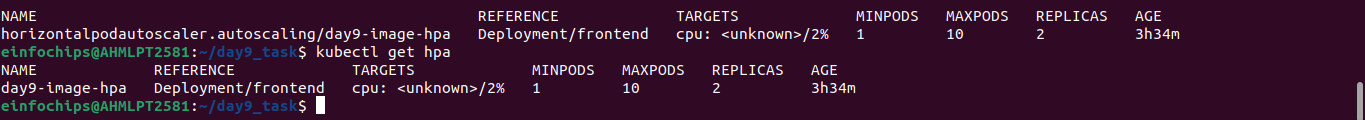
.

**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.

.

.

.

