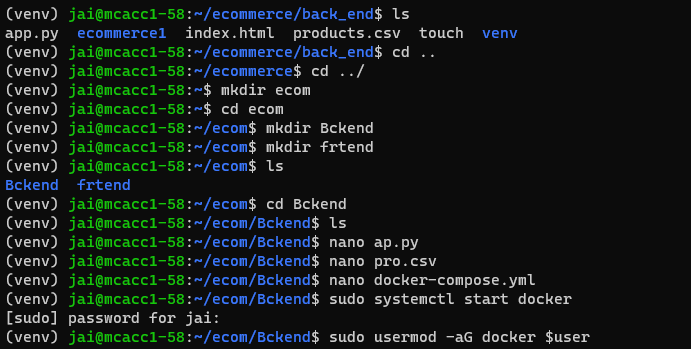
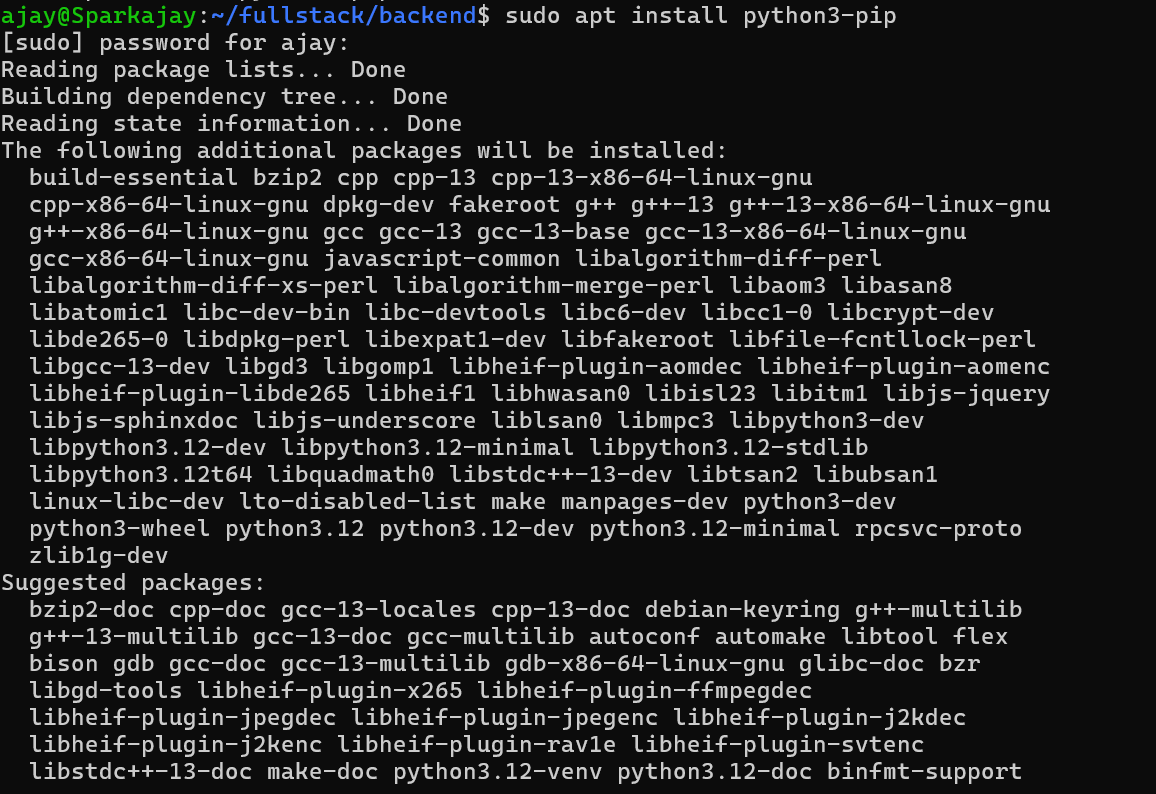
**DEVOPS**

**DAY-4 Kubernetes**

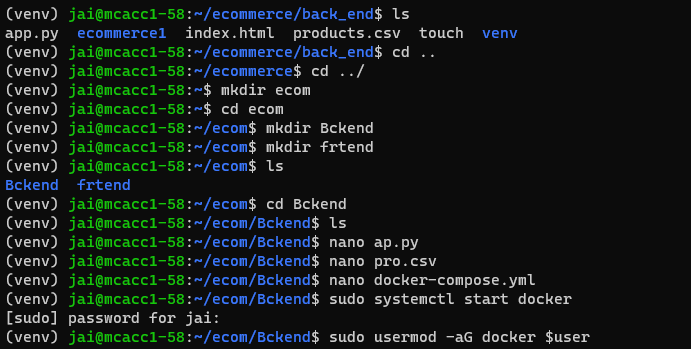
To create a one Directory



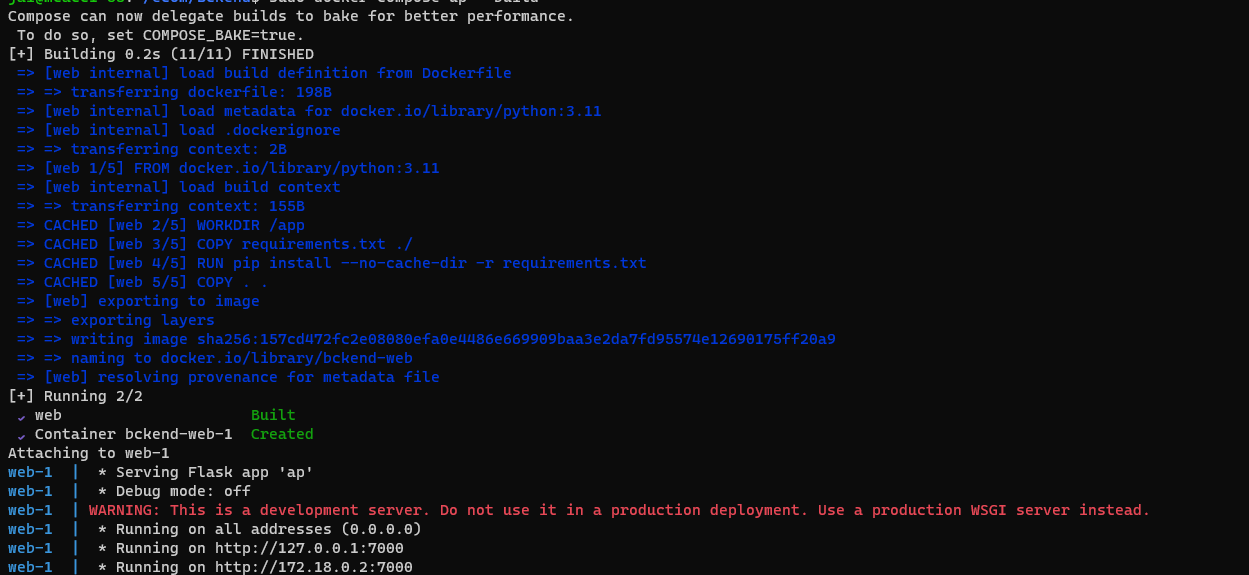
To install python:



After Creating Python, Create an **Backend** to store the files,

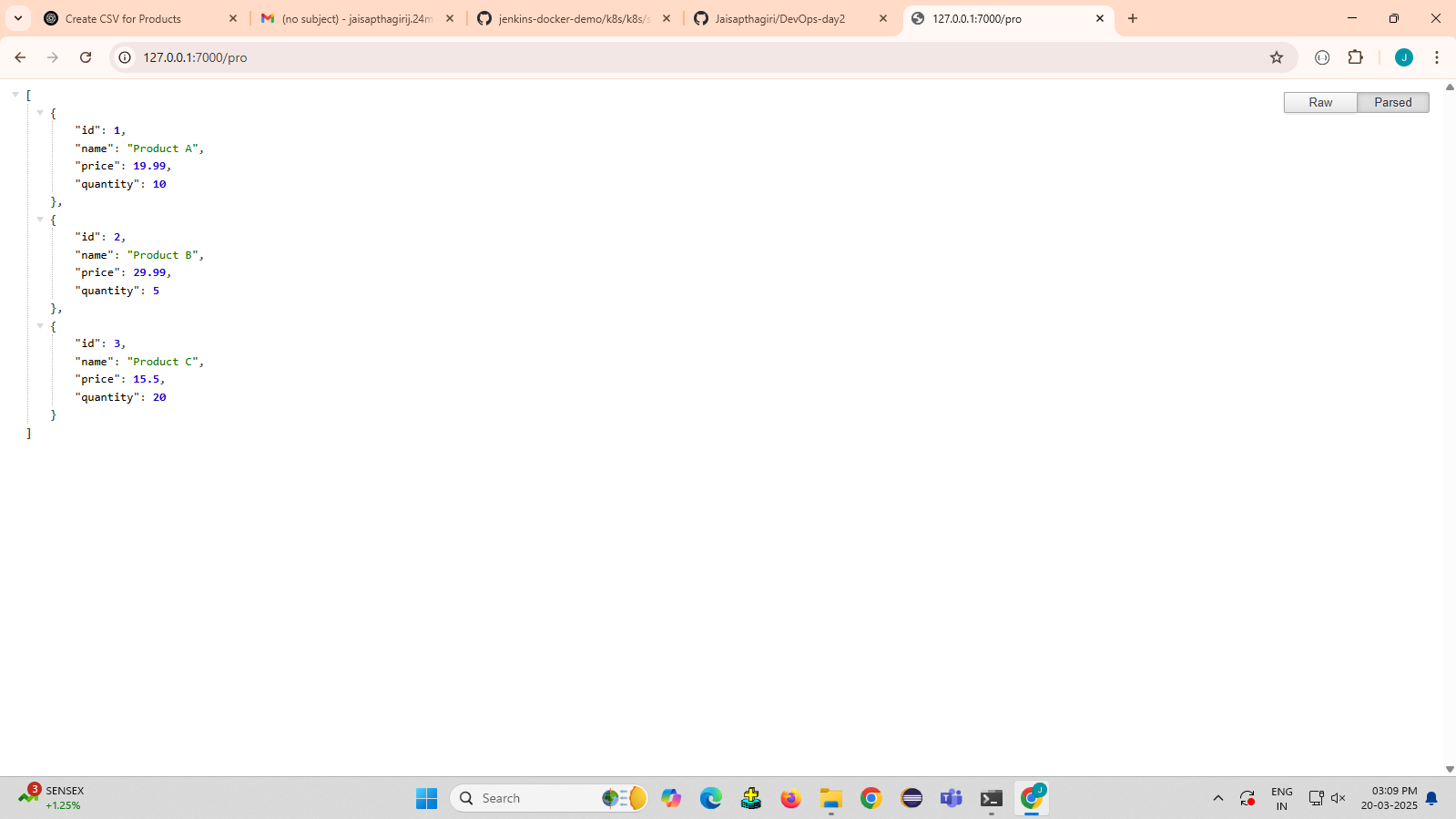


then build Docker and Run the Docker



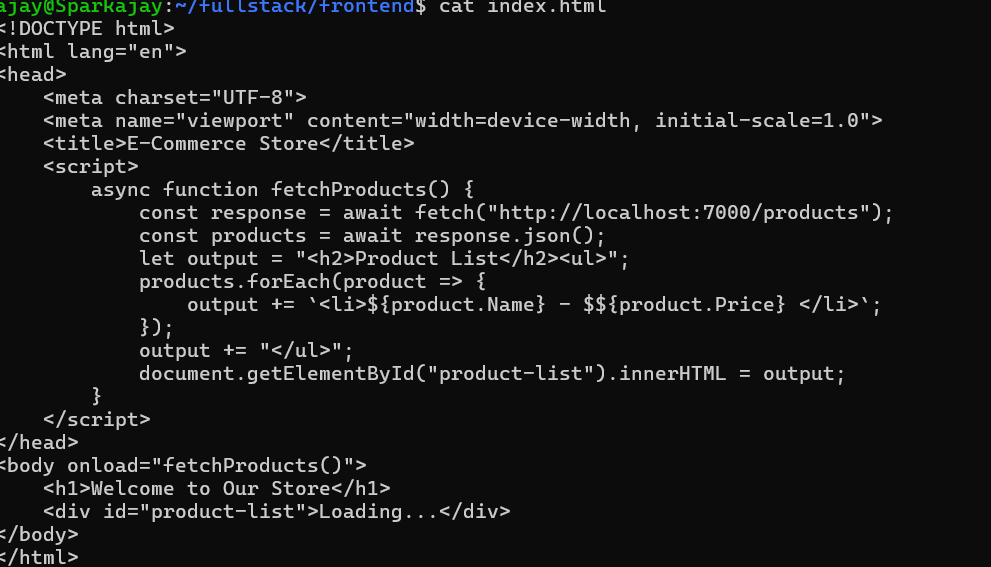
**To see the Output, goto the website and put the URL:**

then it will display the Json format in website

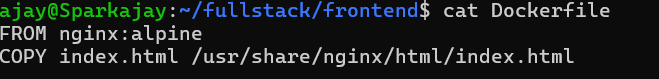
****

In **frontend**, Create the files like

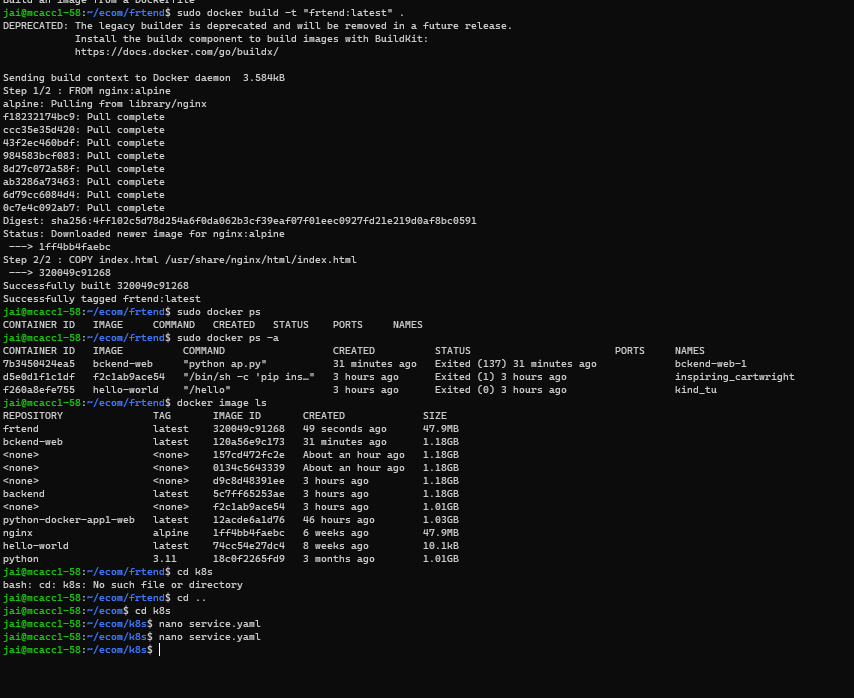
**index.html**



**Dockerfile**

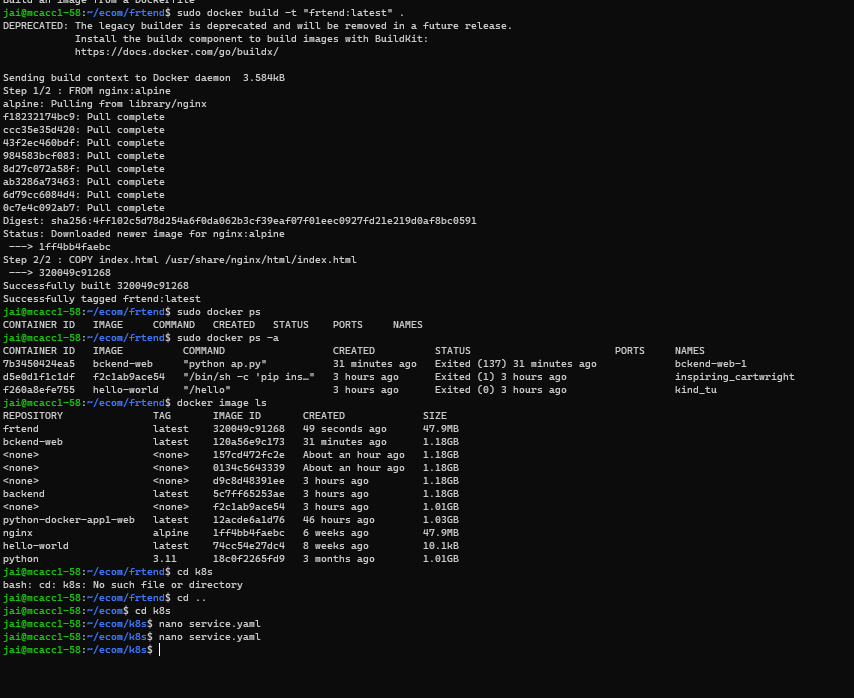
****

**Build the Docker image in Frontend**

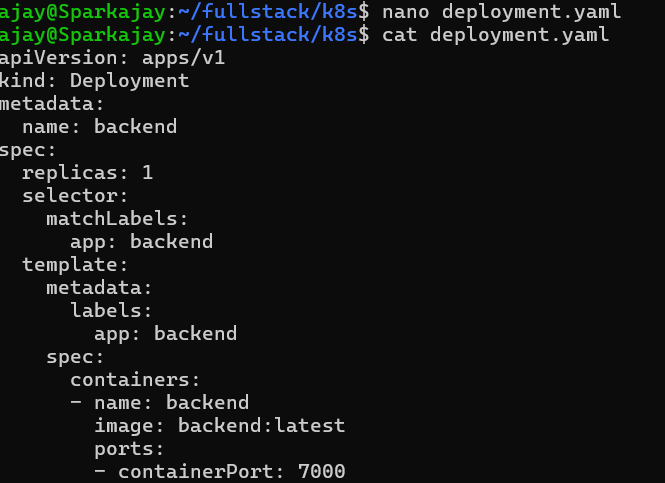
****

**To create an k8s folder in fullstack using mkdir command:**

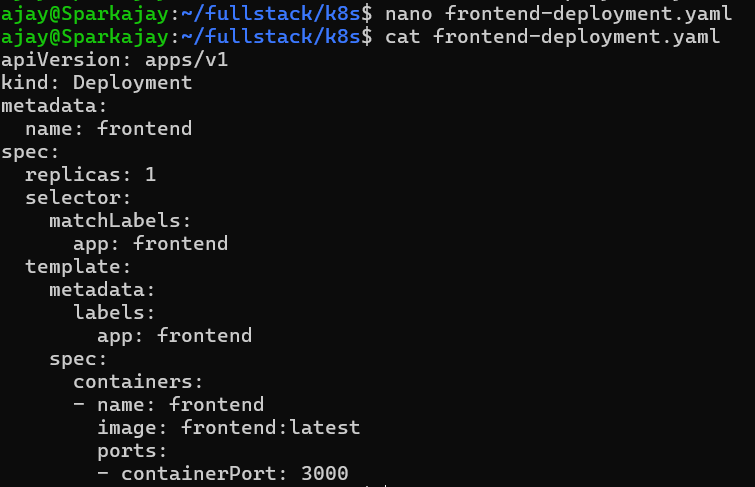
**For deployment use Kubernetes**

****

**Deployment.yaml (for Backend)**

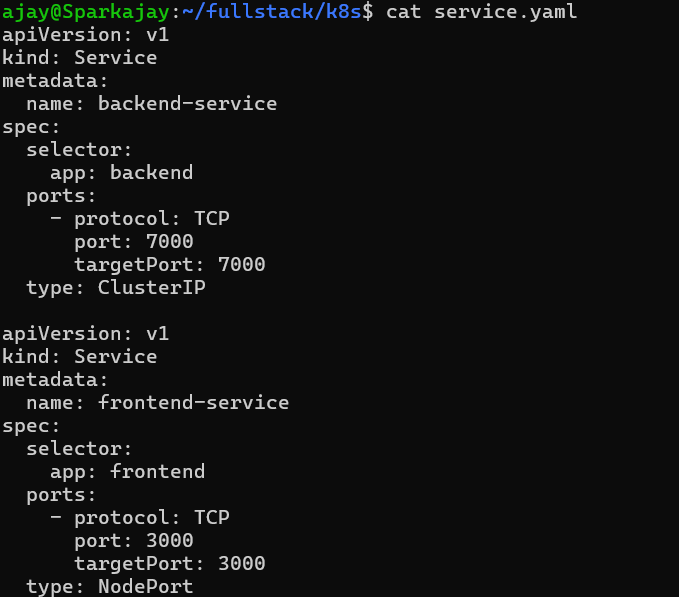
****

**Deployment.yaml (for Frontend)**

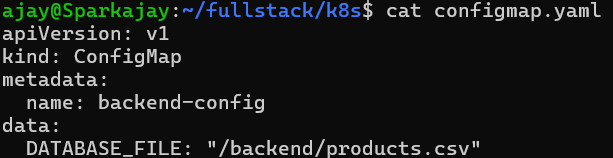
****

**Service.yaml:**

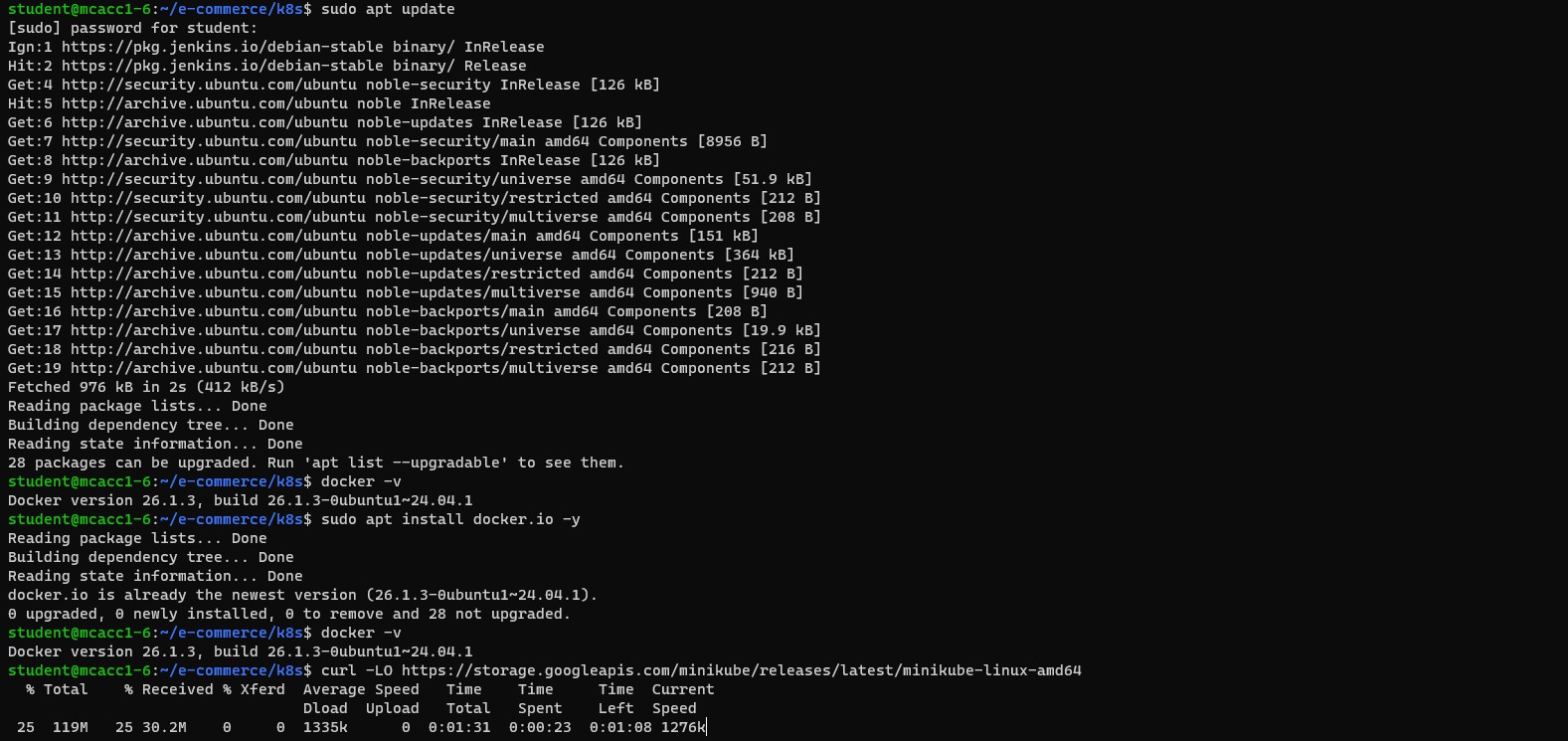
For **frontend-** nodeport **backend-**cluster IP

****

**Configmap.yaml:**

****

Install minikube

Minikube is a tool that allows you to run a Kubernetes cluster locally on our machine. It is designed for developers who want to test and experiment with Kubernetes without needing a full-scale cloud-based cluster.

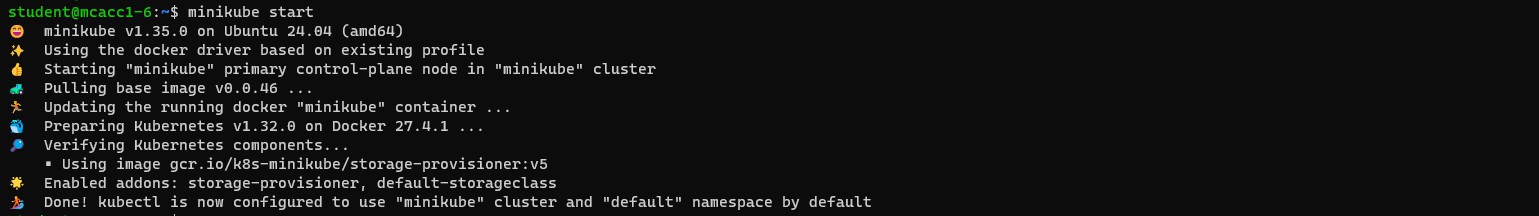
Install kubectl

kubectl is the command-line tool used to interact with a Kubernetes cluster. It allows you to deploy applications, inspect and manage cluster resources, and troubleshoot issues.

Grant permission for **kubectl chmod +x kubectl**

Move to kubectl to root

Check the **minikube** and **kubectl** installed properly

****Start minicube: **minikube start**

**Verify minikube is running**

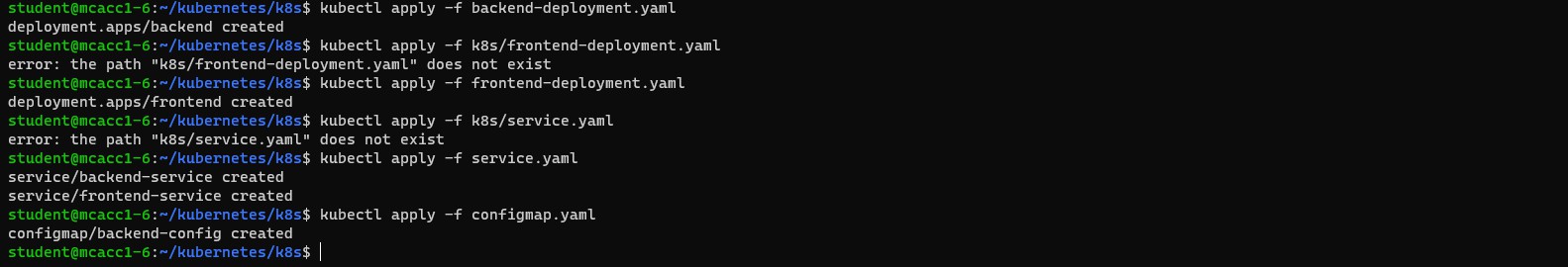
****

**Load the image to the minikube Befor loading images**

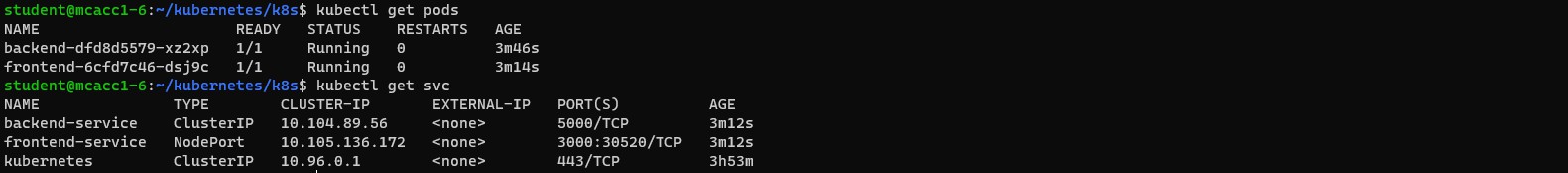
**Perform this commend:** eval $(minikube docker-env) minikube image load frontend:latest

minikube image load backend:latest

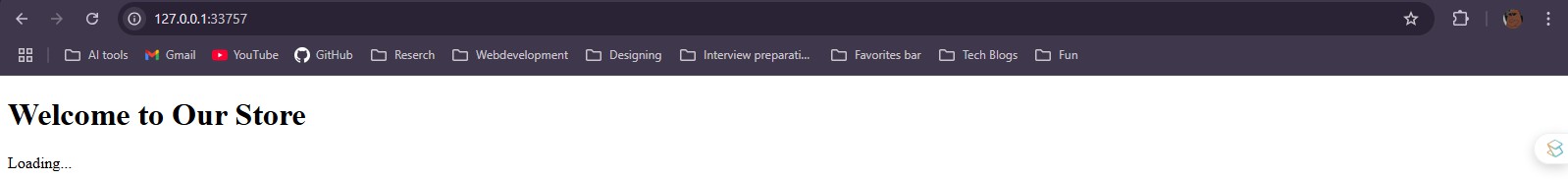
**Check the images are loaded**

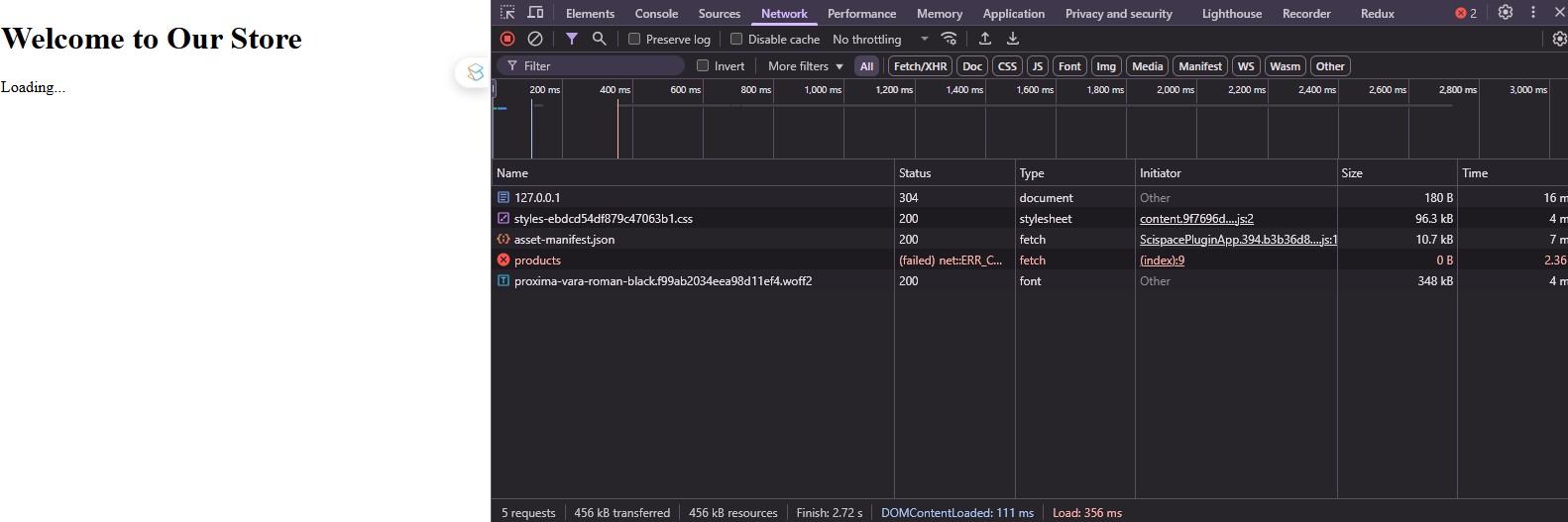
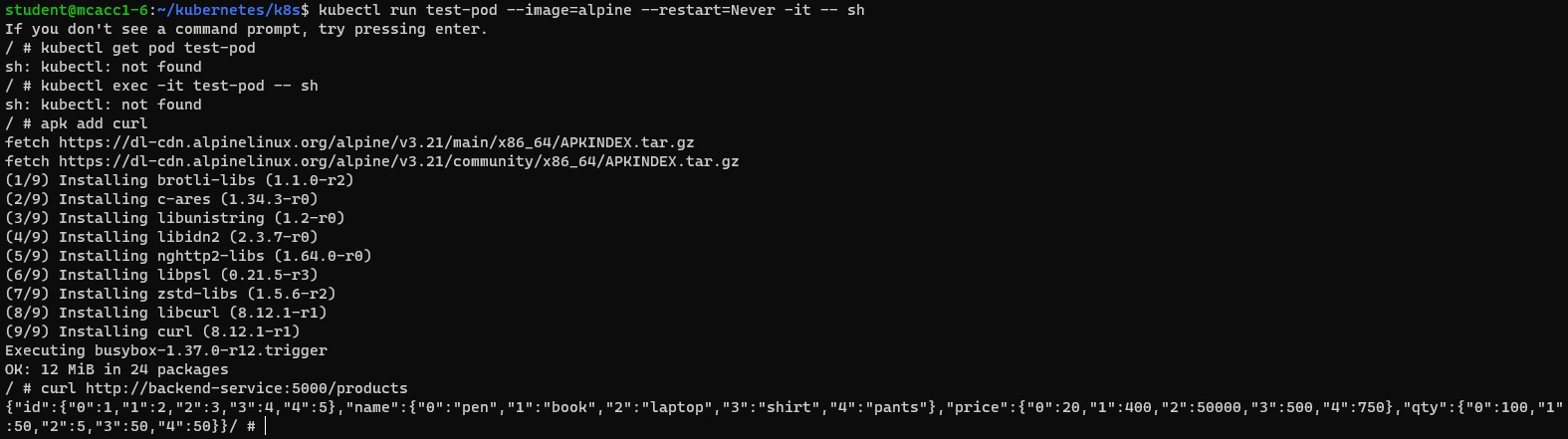
Commands are used to deploy your application components (backend and frontend), expose them through a service, and provide them with the necessary configuration via a ConfigMap.

These commands are used to list and inspect the running resources in your Kubernetes cluster:

**kubectl get pods kubectl get svc**

**To test Frontend**

****

**To Test backend**

this kind of output because we are running this frontend on localhost.

this kind of output because we are running this frontend on localhost.