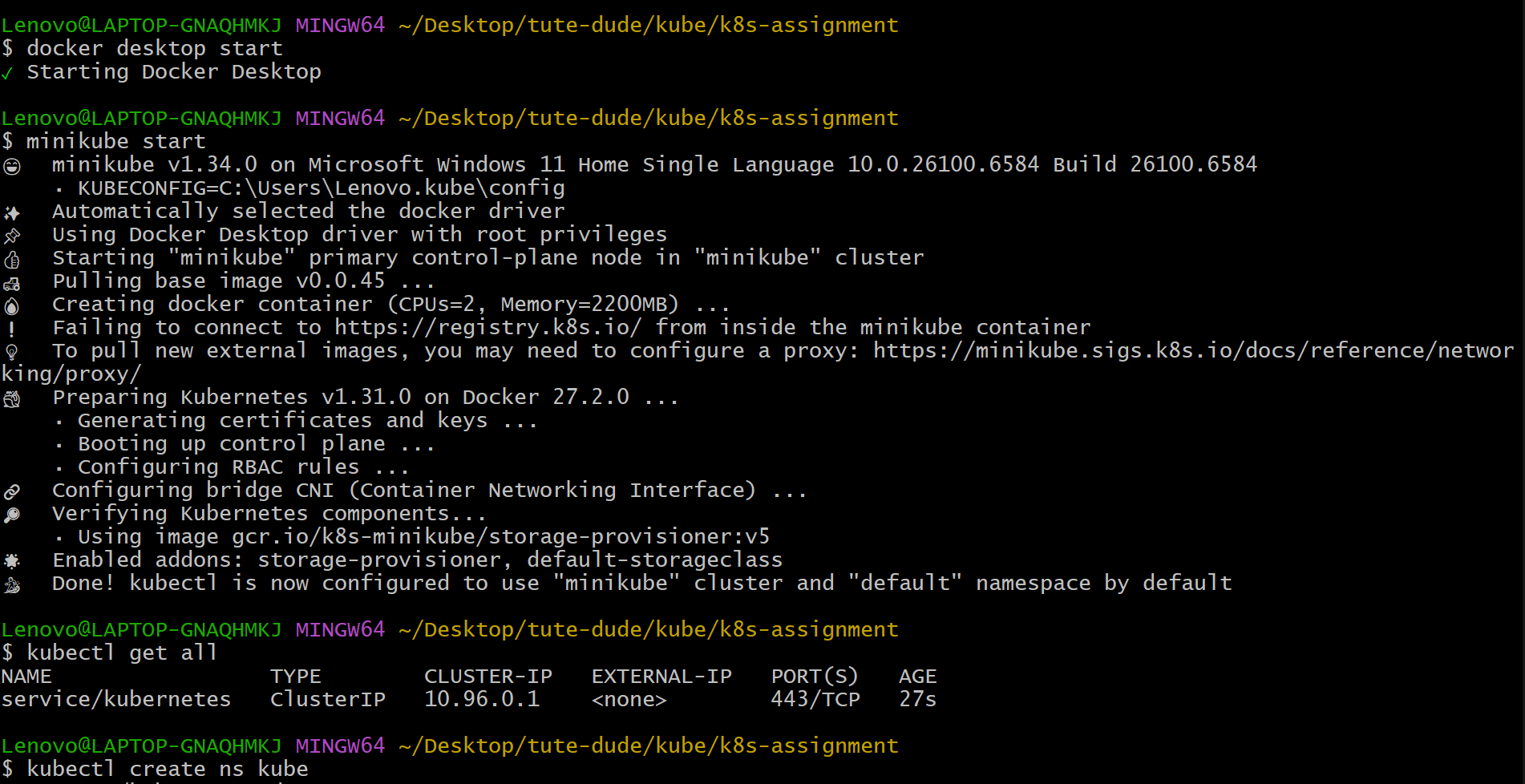
1. Creating Minikube Cluster and create a namespace called “kube” and switch to “kube” namespace from default :

Commands:

1. Start Docker engine
2. minikube start
3. kubectl get all
4. kubectl create ns kube (creating namespace)
5. kubens kube (switching namespace)



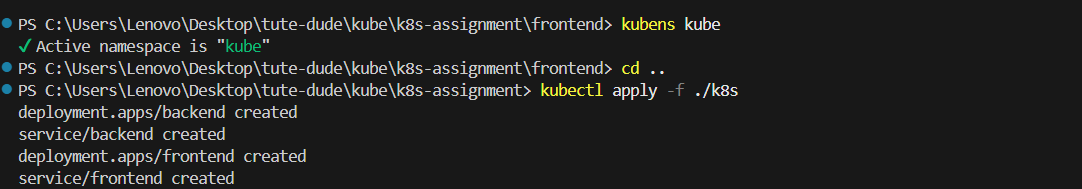
1. Accessing minikube dashboard:

Commands:

1. minikube dashboard
2. Applying backend and frontend yaml manifest to create deployments with pods, services and replica sets.

Commands:

1. Kubectl apply -f ./k8s (folder with yaml file)



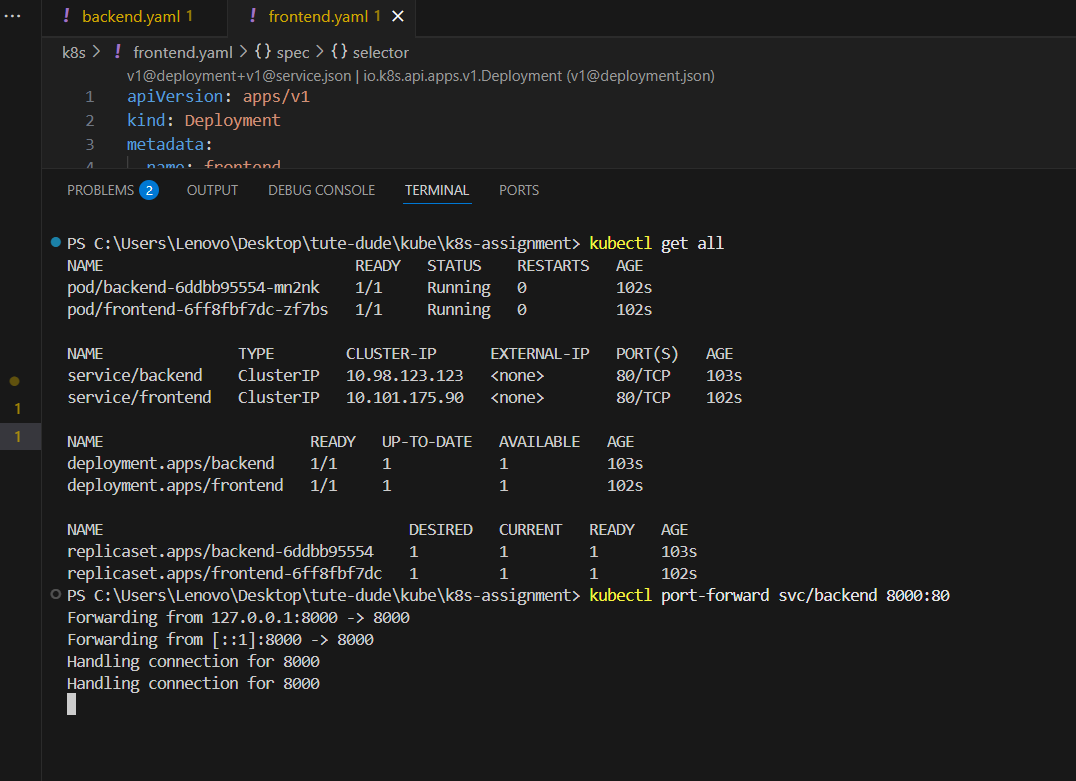
1. Checking if the pods are running and deployments , services are ready and up-to-date.

Commands:

1. Kubectl get all
2. Port forwarding by mapping external port ‘8000’ with backend service port ‘80’ to access backend application running inside the cluster on browser.

Commands:

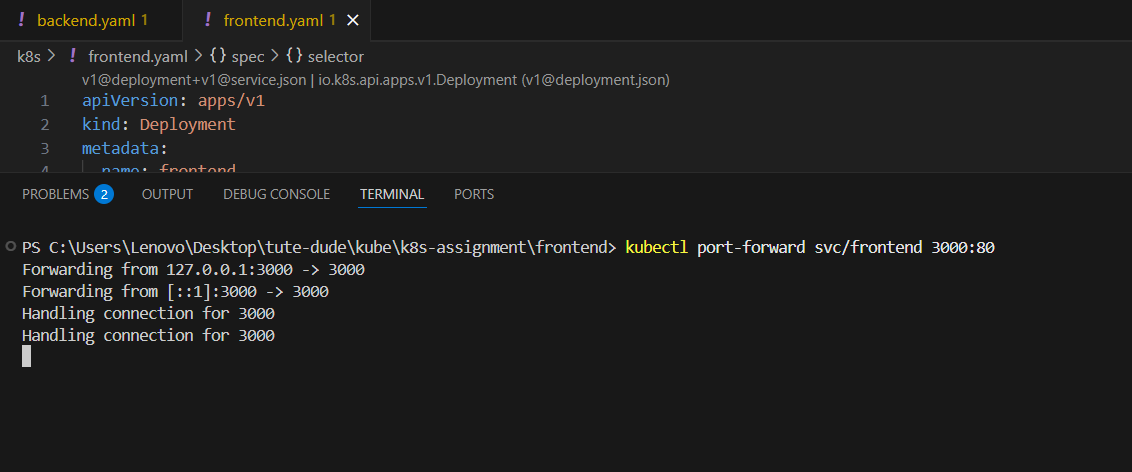
1. Kubectl port-forward svc/backend 8000:80



1. Port forwarding by mapping external port ‘3000’ with frontend service port ‘80’ to access frontend application running inside the cluster on browser.

Commands:

1. Kubectl port-forward svc/frontend 3000:80



1. Accessing frontend and backend application on browser and checking API is working fine by adding data in frontend form and clicking place order and check whether we are getting desired output and access backend to check the same data is stored.
2. Access frontend:

<http://localhost:3000>

enter the form data and click on place order button.

Output: It will take you to <http://localhost:3000/submit> page with payment successful and order placed message.

1. Access backend:

<http://localhost:8000/data>

Output: It should display same data entered in place order form.

