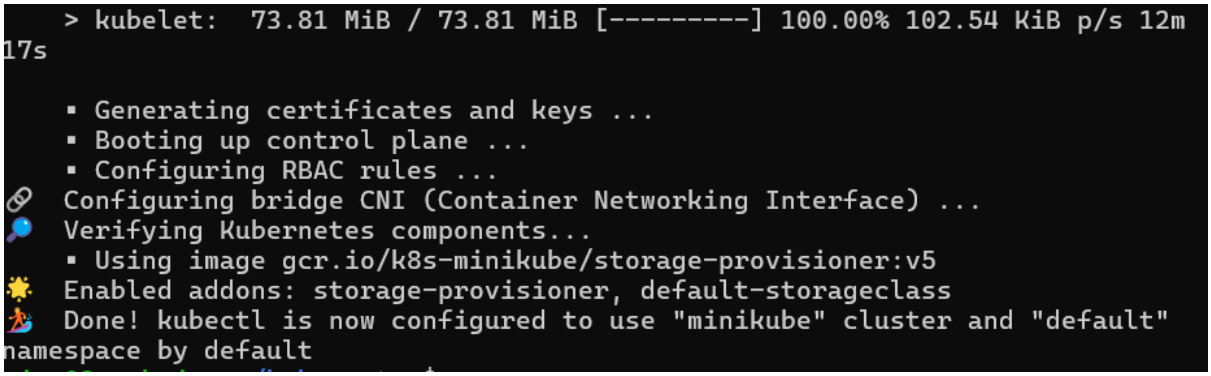
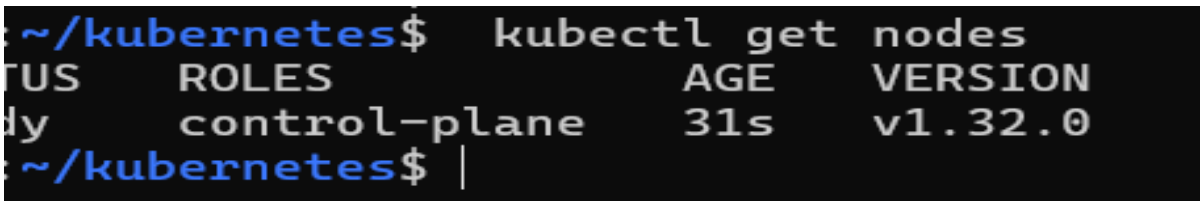
Day-4

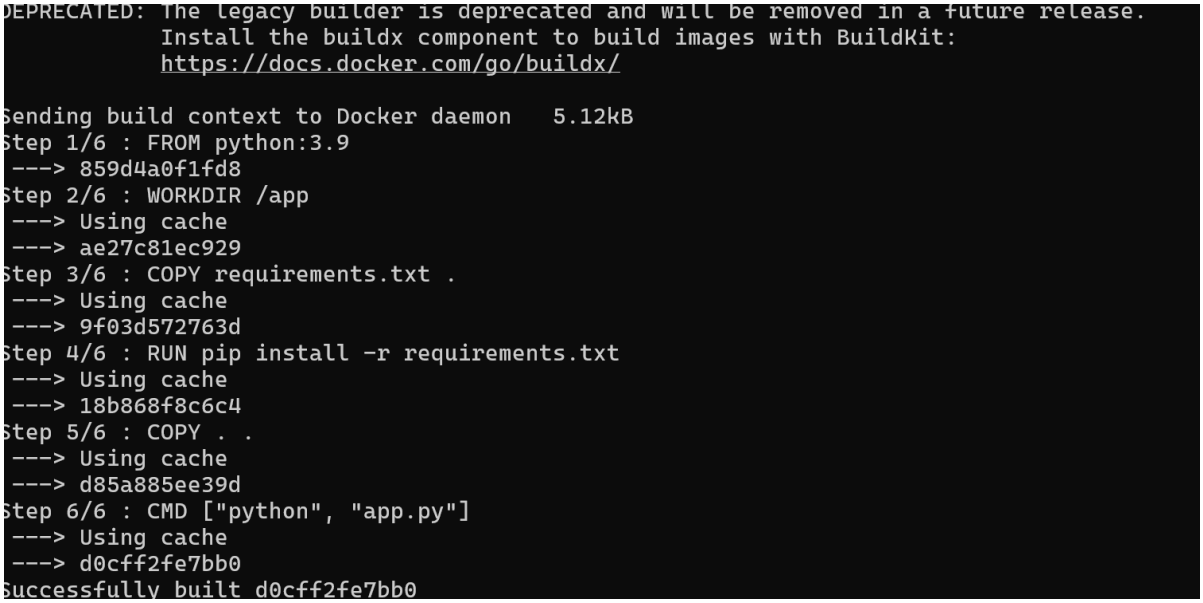
Minikube start:



Kubectl nodes:



To Build docker in Backend



Minikube for backend:

```
~/kubernetes/backend$ minikube image load backend:latest
```

To Build Docker in Frontend:

```
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   5.12kB
Step 1/6 : FROM python:3.9
 ---> 859d4a0f1fd8
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> ae27c81ec929
Step 3/6 : COPY requirements.txt .
 ---> Using cache
 ---> 9f03d572763d
Step 4/6 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> 18b868f8c6c4
Step 5/6 : COPY . .
 ---> Using cache
 ---> d85a885ee39d
Step 6/6 : CMD ["python", "app.py"]
 ---> Using cache
```

Minikube for frontend:

```
/kubernetes/frontend$ minikube image load frontend:latest
```

To create a Deployment file for Kubernetes for frontend, Backend, service.yaml,

```
~/kubernetes/backend$ cd ..
~/kubernetes$ cd k8s/
~/kubernetes/k8s$ kubectl apply -f backend-deployment.yaml --validate=false
/backend created
~/kubernetes/k8s$ kubectl apply -f frontend-deployment.yaml --validate=false
/frontend created
~/kubernetes/k8s$ kubectl apply -f service.yaml --validate=false
-service created
d-service created
~/kubernetes/k8s$ kubectl apply -f configmap.yaml --validate=false
nd-config created
~/kubernetes/k8s$ kubectl get pods
          READY   STATUS    RESTARTS   AGE
79-8rdzg   1/1     Running   0          63s
46-5txnb   1/1     Running   0          54s
~/kubernetes/k8s$ kubectl get svc
    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
    ClusterIP   10.110.154.68   <none>        5000/TCP         87s
e   NodePort    10.98.250.114   <none>        3000:32434/TCP   87s
    ClusterIP   10.96.0.1       <none>        443/TCP          12m
~/kubernetes/k8s$ minikube service frontend-service --url
1:42597
 are using a Docker driver on linux, the terminal needs to be open to run it.
```

Create an pods and svc

```
:~/kubernetes/k8s$ kubectl get pods
          READY   STATUS    RESTARTS   AGE
579-8rdzg   1/1     Running   0          63s
c46-5txnb   1/1     Running   0          54s
:~/kubernetes/k8s$ kubectl get svc
    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
e   ClusterIP   10.110.154.68   <none>        5000/TCP         87s
ce  NodePort    10.98.250.114   <none>        3000:32434/TCP   87s
    ClusterIP   10.96.0.1       <none>        443/TCP          12m
```

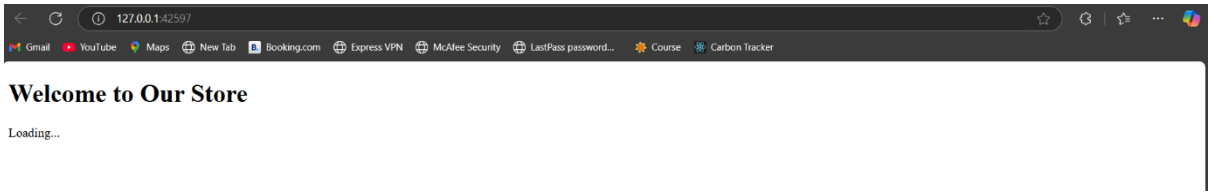
We can see the Output by using Curl command

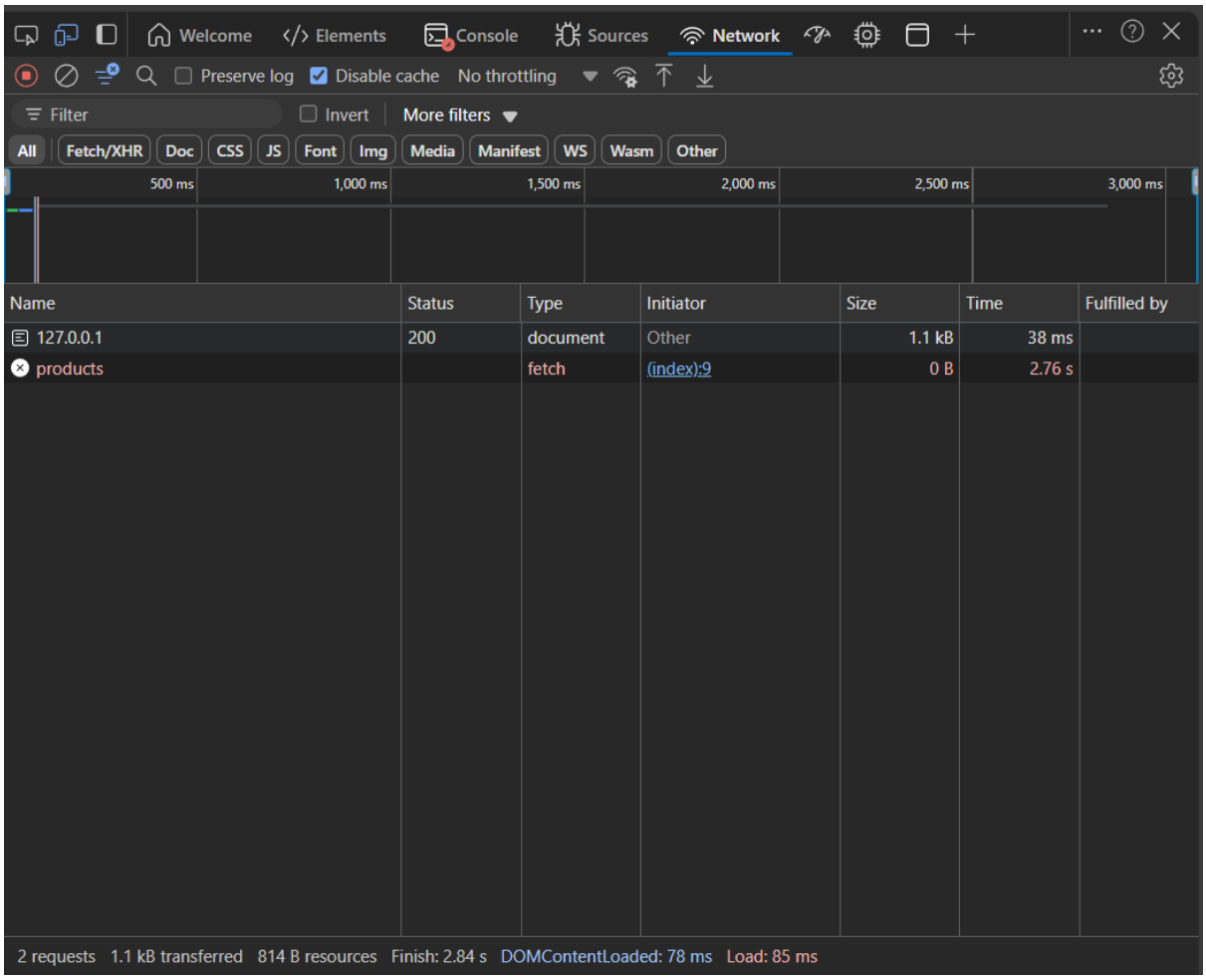
```
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{"id":2,"name":"Laptop","price":799.99},{"id":3,"name":"Headphone
s","price":49.99},{"id":4,"name":"Tablet","price":199.99}]
/ # exit
```

To run the frontend

```
~/kubernetes/k8s$ minikube service frontend-service --url
1:42597
! are using a Docker driver on linux, the terminal needs to be open to run it.
```

Output





Note : Since, we are expected this kind of output, because we are running this frontend in localhost.