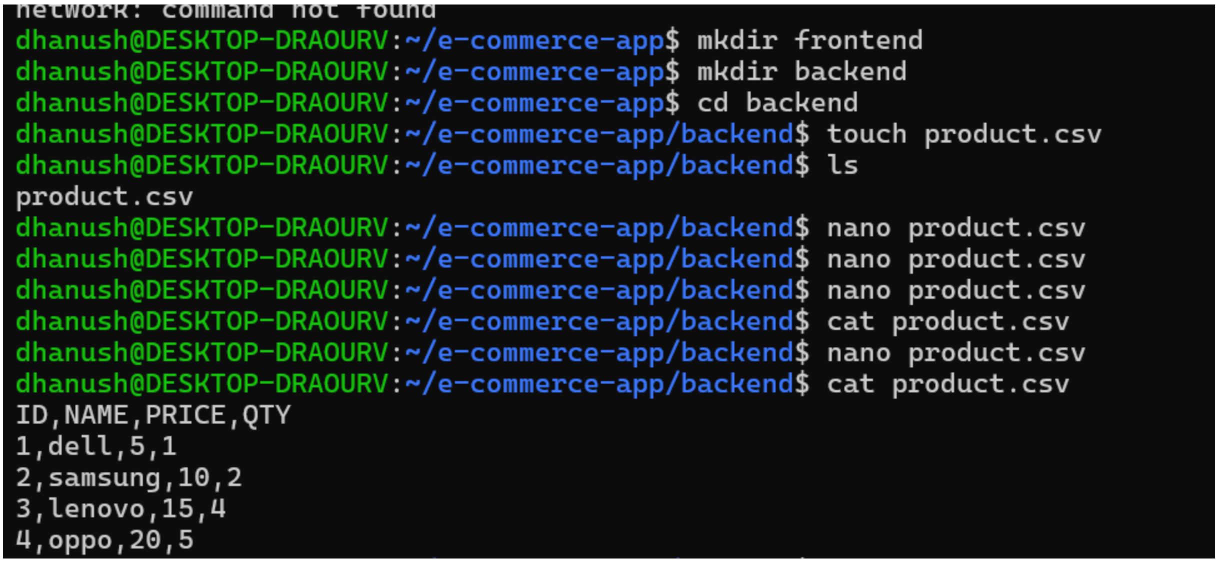
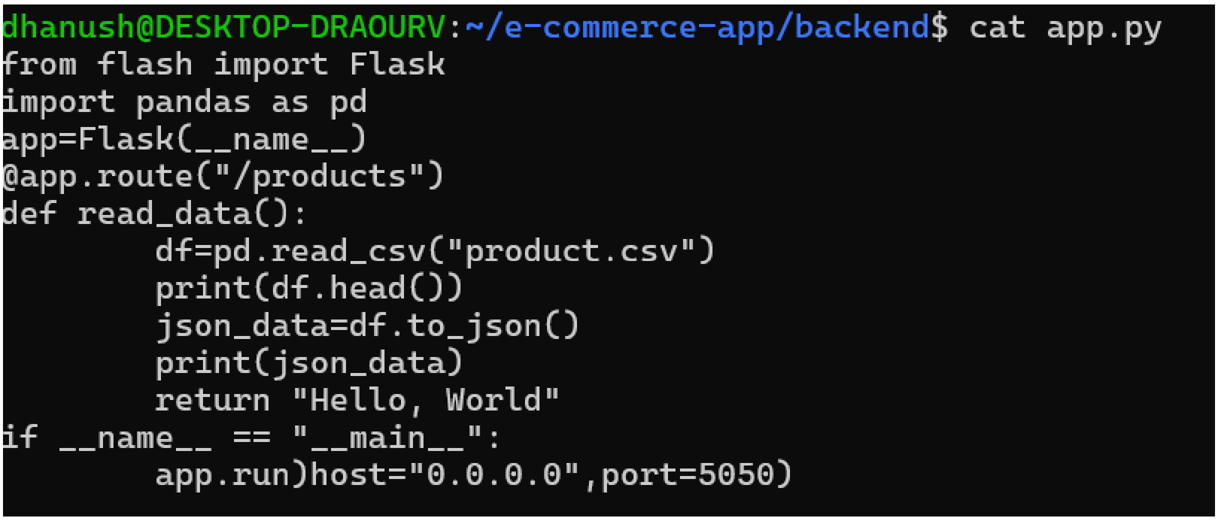
Day 4: Kubernetes

Create a directory ‘e-commerce’ and its required folders and files

Create a **products.csv** file and **app.py**

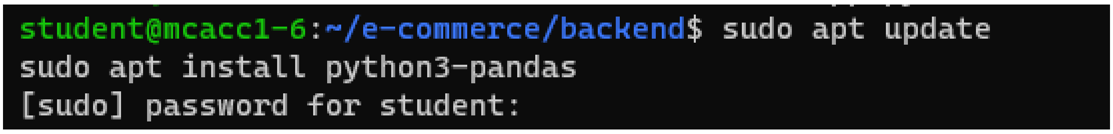


```
network: command not found
dhanush@DESKTOP-DRAOURV:~/e-commerce-app$ mkdir frontend
dhanush@DESKTOP-DRAOURV:~/e-commerce-app$ mkdir backend
dhanush@DESKTOP-DRAOURV:~/e-commerce-app$ cd backend
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ touch product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ ls
product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ nano product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ nano product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ nano product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ cat product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ nano product.csv
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ cat product.csv
ID,NAME,PRICE,QTY
1,dell,5,1
2,samsung,10,2
3,lenovo,15,4
4,oppo,20,5
```



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ cat app.py
from flash import Flask
import pandas as pd
app=Flask(__name__)
@app.route("/products")
def read_data():
        df=pd.read_csv("product.csv")
        print(df.head())
        json_data=df.to_json()
        print(json_data)
        return "Hello, World"
if __name__ == "__main__":
        app.run)host="0.0.0.0",port=5050)
```

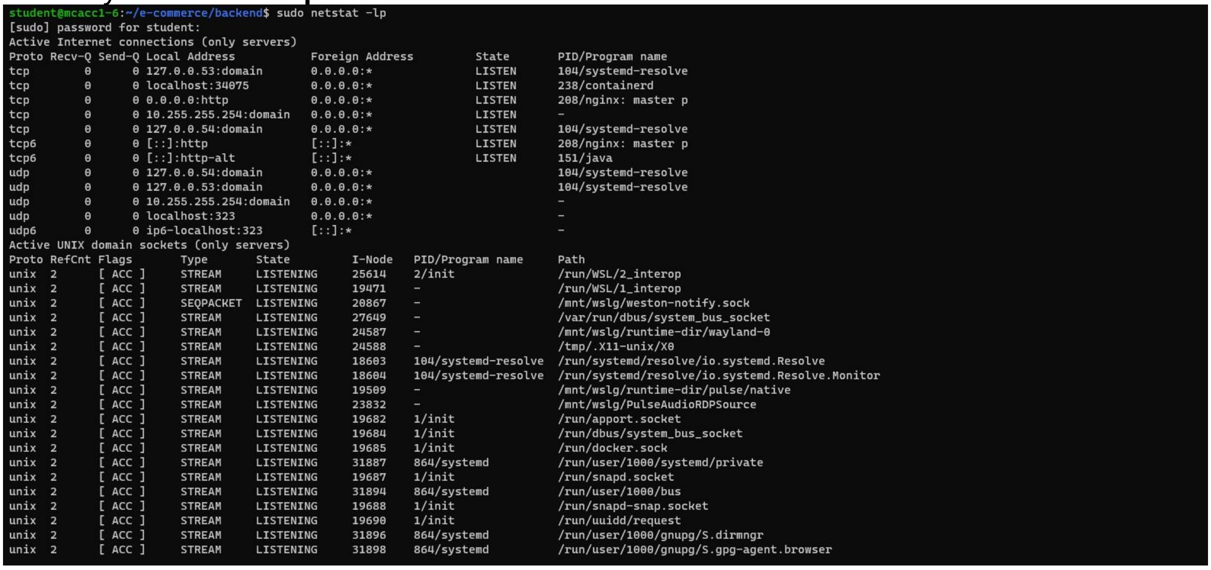
Install the **pandas** library:



```
student@mcacc1-6:~/e-commerce/backend$ sudo apt update
sudo apt install python3-pandas
[sudo] password for student:
```

Ensure that the CSV file is read and correctly parsed into **JSON format**.

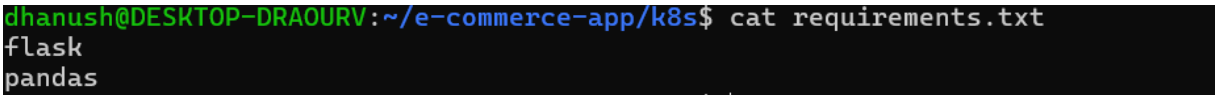
To verify the **available port numbers**



```
student@mcacc1-6:~/e-commerce/backend$ sudo netstat -lp
[sudo] password for student:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:domain       0.0.0.0:*               LISTEN      104/systemd-resolve
tcp        0      0 localhost:34075         0.0.0.0:*               LISTEN      238/containerd
tcp        0      0 0.0.0.0:http            0.0.0.0:*               LISTEN      208/nginx: master p
tcp        0      0 10.255.255.254:domain   0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.54:domain       0.0.0.0:*               LISTEN      104/systemd-resolve
tcp6       0      0 [::]:http               [::]:*                  LISTEN      208/nginx: master p
tcp6       0      0 [::]:http-alt           [::]:*                  LISTEN      151/java
udp        0      0 127.0.0.54:domain       0.0.0.0:*                           104/systemd-resolve
udp        0      0 127.0.0.53:domain       0.0.0.0:*                           104/systemd-resolve
udp        0      0 10.255.255.254:domain   0.0.0.0:*                           -
udp        0      0 localhost:323           0.0.0.0:*                           -
udp6       0      0 ip6-localhost:323       [::]:*                              -
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name     Path
unix  2      [ ACC ]     STREAM     LISTENING     25614    2/init               /run/WSL/2_interop
unix  2      [ ACC ]     STREAM     LISTENING     19471    -                    /run/WSL/1_interop
unix  2      [ ACC ]     SEQPACKET  LISTENING     20867    -                    /mnt/wslg/weston-notify.sock
unix  2      [ ACC ]     STREAM     LISTENING     27649    -                    /var/run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     24587    -                    /mnt/wslg/runtime-dir/wayland-0
unix  2      [ ACC ]     STREAM     LISTENING     24588    -                    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     18603    104/systemd-resolve  /run/systemd/resolve/io.systemd.Resolve
unix  2      [ ACC ]     STREAM     LISTENING     18604    104/systemd-resolve  /run/systemd/resolve/io.systemd.Resolve.Monitor
unix  2      [ ACC ]     STREAM     LISTENING     19509    -                    /mnt/wslg/runtime-dir/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     23832    -                    /mnt/wslg/PulseAudioRDPSource
unix  2      [ ACC ]     STREAM     LISTENING     19682    1/init               /run/apport.socket
unix  2      [ ACC ]     STREAM     LISTENING     19684    1/init               /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     19685    1/init               /run/docker.sock
unix  2      [ ACC ]     STREAM     LISTENING     31887    864/systemd          /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     19687    1/init               /run/snapd.socket
unix  2      [ ACC ]     STREAM     LISTENING     31894    864/systemd          /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     19688    1/init               /run/snapd-snap.socket
unix  2      [ ACC ]     STREAM     LISTENING     19690    1/init               /run/uuidd/request
unix  2      [ ACC ]     STREAM     LISTENING     31896    864/systemd          /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     31898    864/systemd          /run/user/1000/gnupg/S.gpg-agent.browser
```

Create **requirements.txt** file

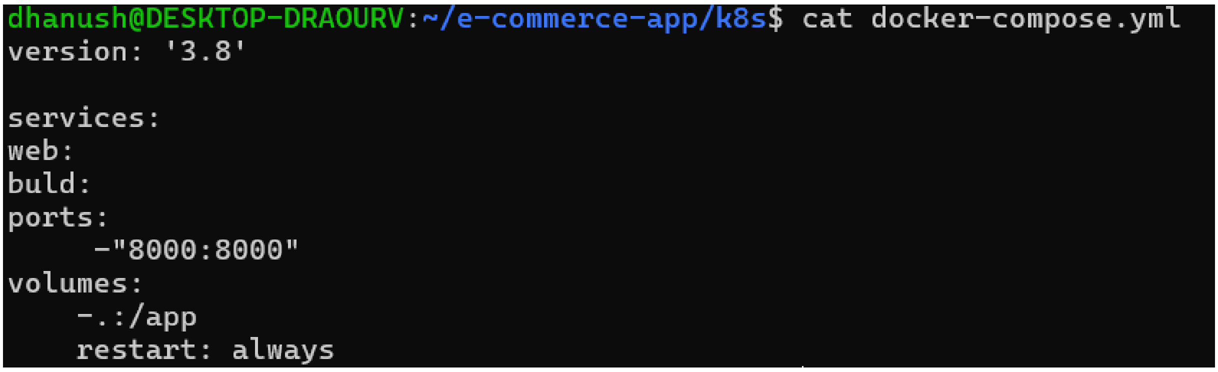
The `requirements.txt` file is used in Python projects to list all the dependencies (packages) that the application needs to run.



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat requirements.txt
flask
pandas
```

Create **docker-compose.yml** file

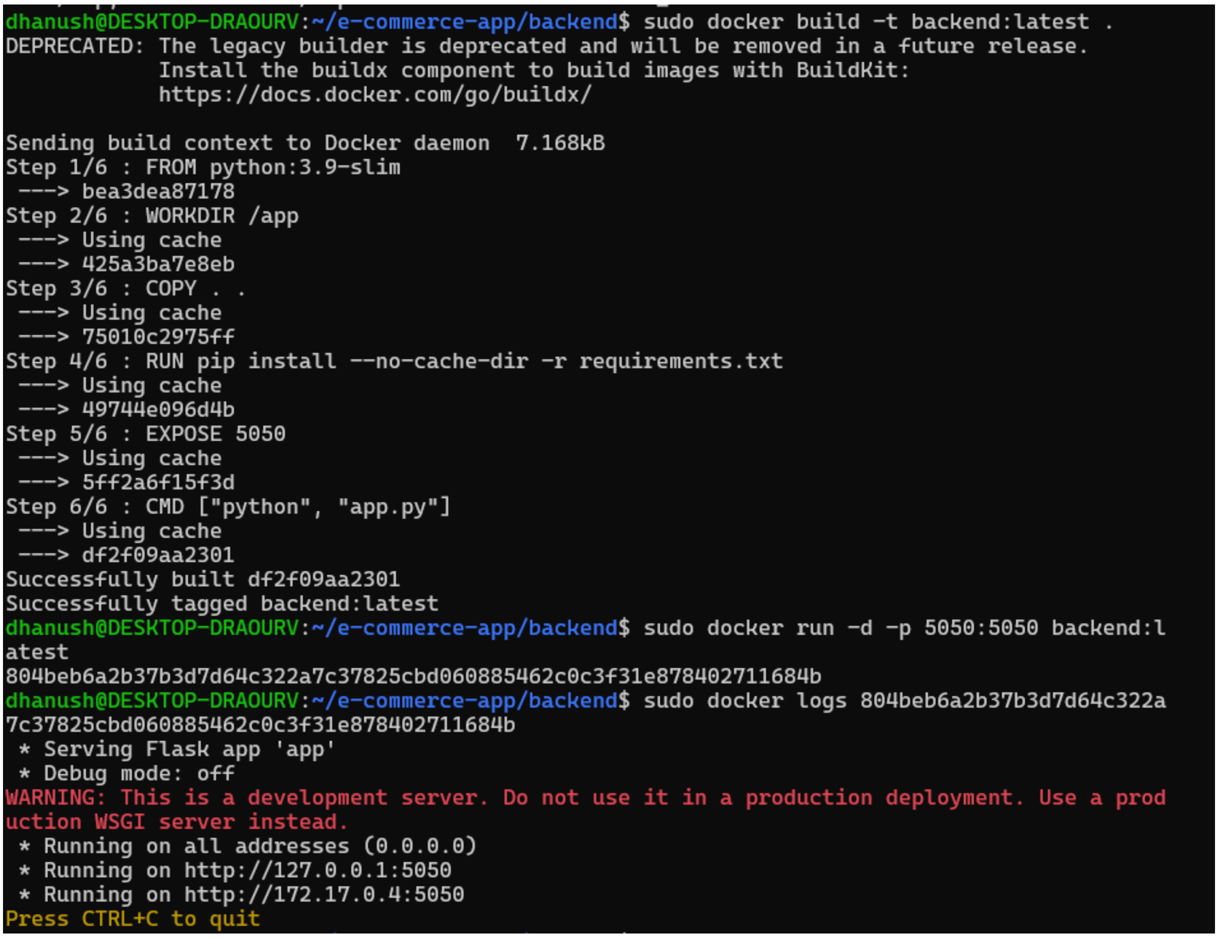
`docker-compose.yml` is a YAML configuration file used to define and run multi-container Docker applications.



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat docker-compose.yml
version: '3.8'

services:
web:
buld:
ports:
     -"8000:8000"
volumes:
    -.:/app
    restart: always
```

Build Docker image

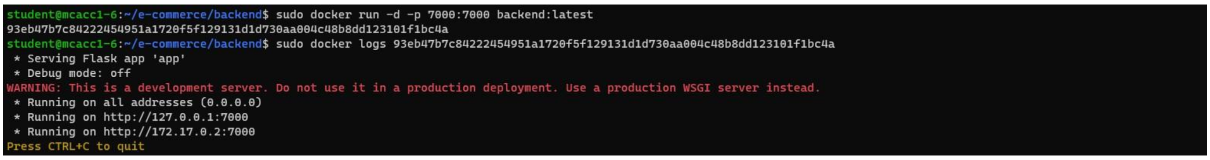
Sudo docker build -t backend:latest



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ sudo docker build -t backend:latest .
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  7.168kB
Step 1/6 : FROM python:3.9-slim
 ---> bea3dea87178
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> 425a3ba7e8eb
Step 3/6 : COPY . .
 ---> Using cache
 ---> 75010c2975ff
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> 49744e096d4b
Step 5/6 : EXPOSE 5050
 ---> Using cache
 ---> 5ff2a6f15f3d
Step 6/6 : CMD ["python", "app.py"]
 ---> Using cache
 ---> df2f09aa2301
Successfully built df2f09aa2301
Successfully tagged backend:latest
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ sudo docker run -d -p 5050:5050 backend:l
atest
804beb6a2b37b3d7d64c322a7c37825cbd060885462c0c3f31e878402711684b
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/backend$ sudo docker logs 804beb6a2b37b3d7d64c322a
7c37825cbd060885462c0c3f31e878402711684b
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a prod
uction WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5050
 * Running on http://172.17.0.4:5050
Press CTRL+C to quit
```

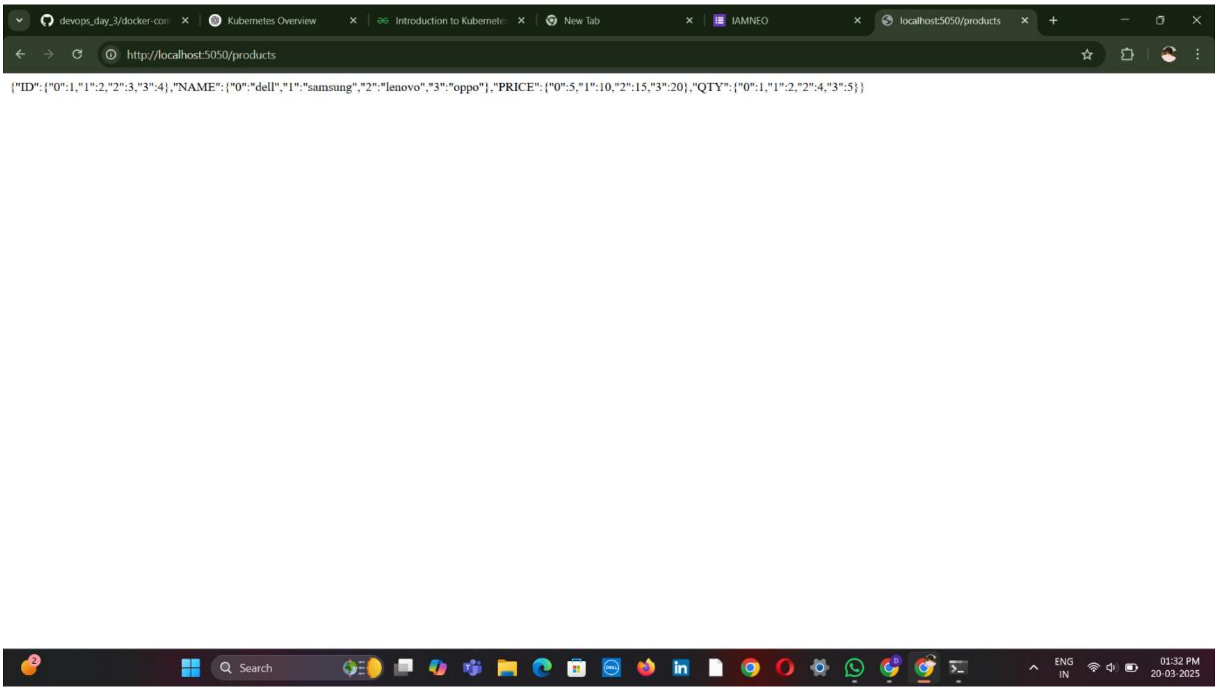
Run the docker:

sudo docker run -d -p 7000:7000 backend:latest
sudo docker logs <Generated number>



```
student@mcacc1-6:~/e-commerce/backend$ sudo docker run -d -p 7000:7000 backend:latest
93eb47b7c84222454951a1720f5f129131d1d730aa004c48b8dd123101f1bc4a
student@mcacc1-6:~/e-commerce/backend$ sudo docker logs 93eb47b7c84222454951a1720f5f129131d1d730aa004c48b8dd123101f1bc4a
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:7000
 * Running on http://172.17.0.2:7000
Press CTRL+C to quit
```

Run the application in the 5050/products

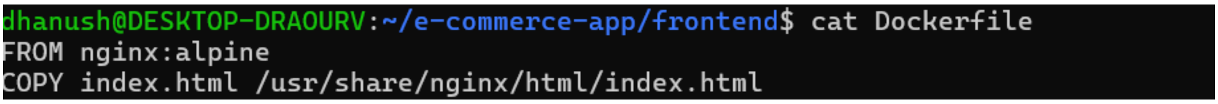


```
{"ID":{"0":1,"1":2,"2":3,"3":4},"NAME":{"0":"dell","1":"samsung","2":"lenovo","3":"oppo"},"PRICE":{"0":5,"1":10,"2":15,"3":20},"QTY":{"0":1,"1":2,"2":4,"3":5}}
```

The JSON data is displayed at our port: 5050/products.

Create a container in frontend

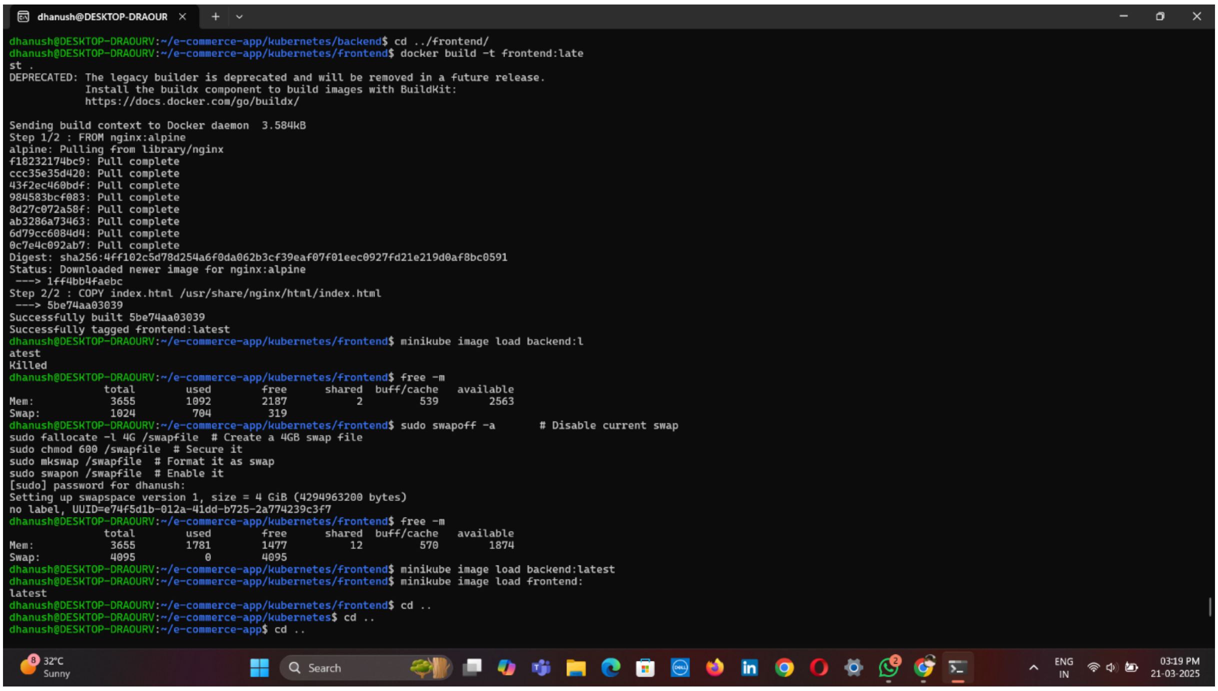
Create **index.html file** and **Dockerfile**



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/frontend$ cat Dockerfile
FROM nginx:alpine
COPY index.html /usr/share/nginx/html/index.html
```

Build the image using the command:

sudo docker build -t frontend:latest.

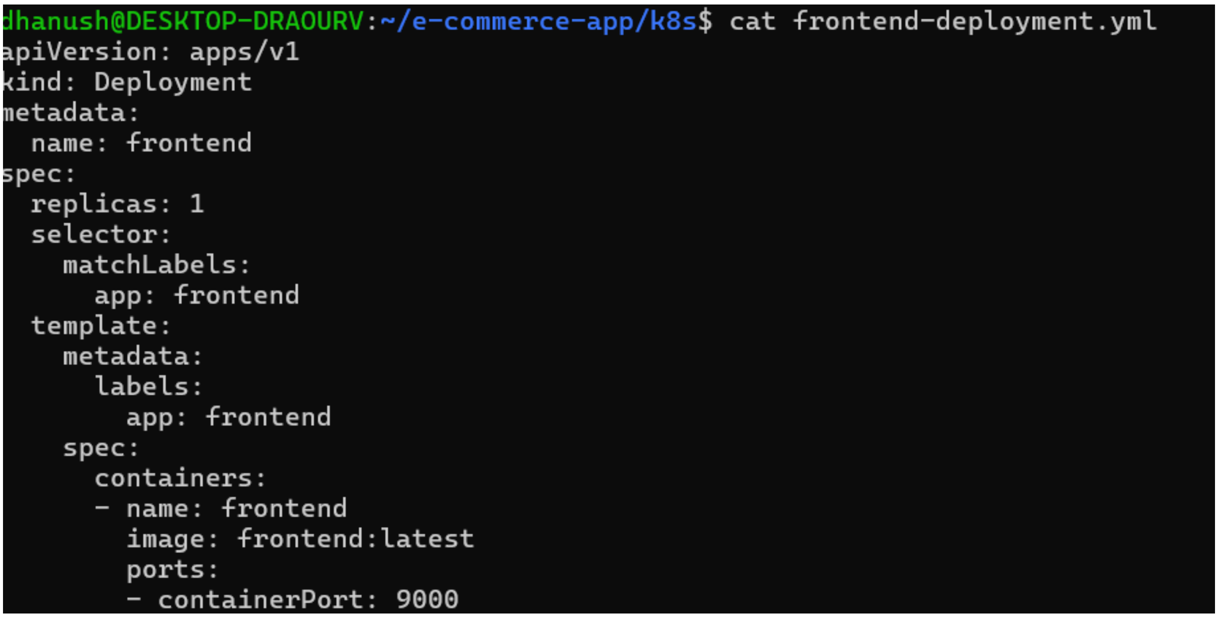


```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/backend$ cd ../frontend/
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ docker build -t frontend:late
st .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.584kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
f18232174bc9: Pull complete
ccc35e35d420: Pull complete
43f2ec460bdf: Pull complete
984583bcf083: Pull complete
8d27c072a58f: Pull complete
ab3286a73463: Pull complete
6d79cc6084d4: Pull complete
0c7e4c092ab7: Pull complete
Digest: sha256:4ff102c5d78d254a6f0da062b3cf39eaf07f01eec0927fd21e219d0af8bc0591
Status: Downloaded newer image for nginx:alpine
 ---> 1ff4bb4faebc
Step 2/2 : COPY index.html /usr/share/nginx/html/index.html
 ---> 5be74aa03039
Successfully built 5be74aa03039
Successfully tagged frontend:latest
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ minikube image load backend:l
atest
Killed
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3655        1092        2187           2         539        2563
Swap:           1024         704         319
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ sudo swapoff -a       # Disable current swap
sudo fallocate -l 4G /swapfile  # Create a 4GB swap file
sudo chmod 600 /swapfile  # Secure it
sudo mkswap /swapfile  # Format it as swap
sudo swapon /swapfile  # Enable it
[sudo] password for dhanush:
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
no label, UUID=e74f5d1b-012a-41dd-b725-2a774239c3f7
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3655        1781        1477          12         570        1874
Swap:           4095           0        4095
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ minikube image load backend:latest
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ minikube image load frontend:
latest
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/frontend$ cd ..
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes$ cd ..
dhanush@DESKTOP-DRAOURV:~/e-commerce-app$ cd ..
```

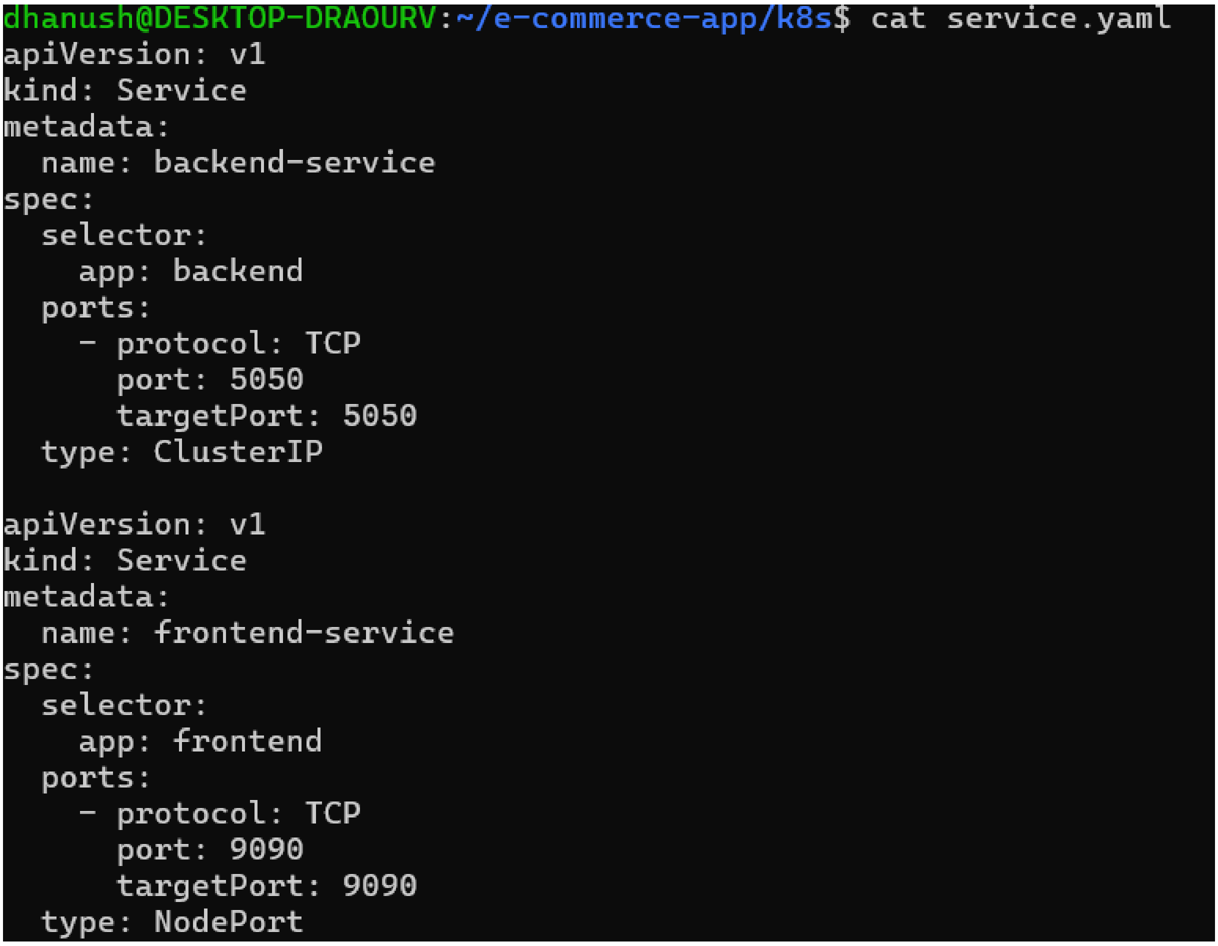
Kubernetes Deployment YAML Files

Create **backend-deployment.yaml file** and **frontend-deployment.yaml** in a folder k8s

These files define how our application should be deployed in the cluster.



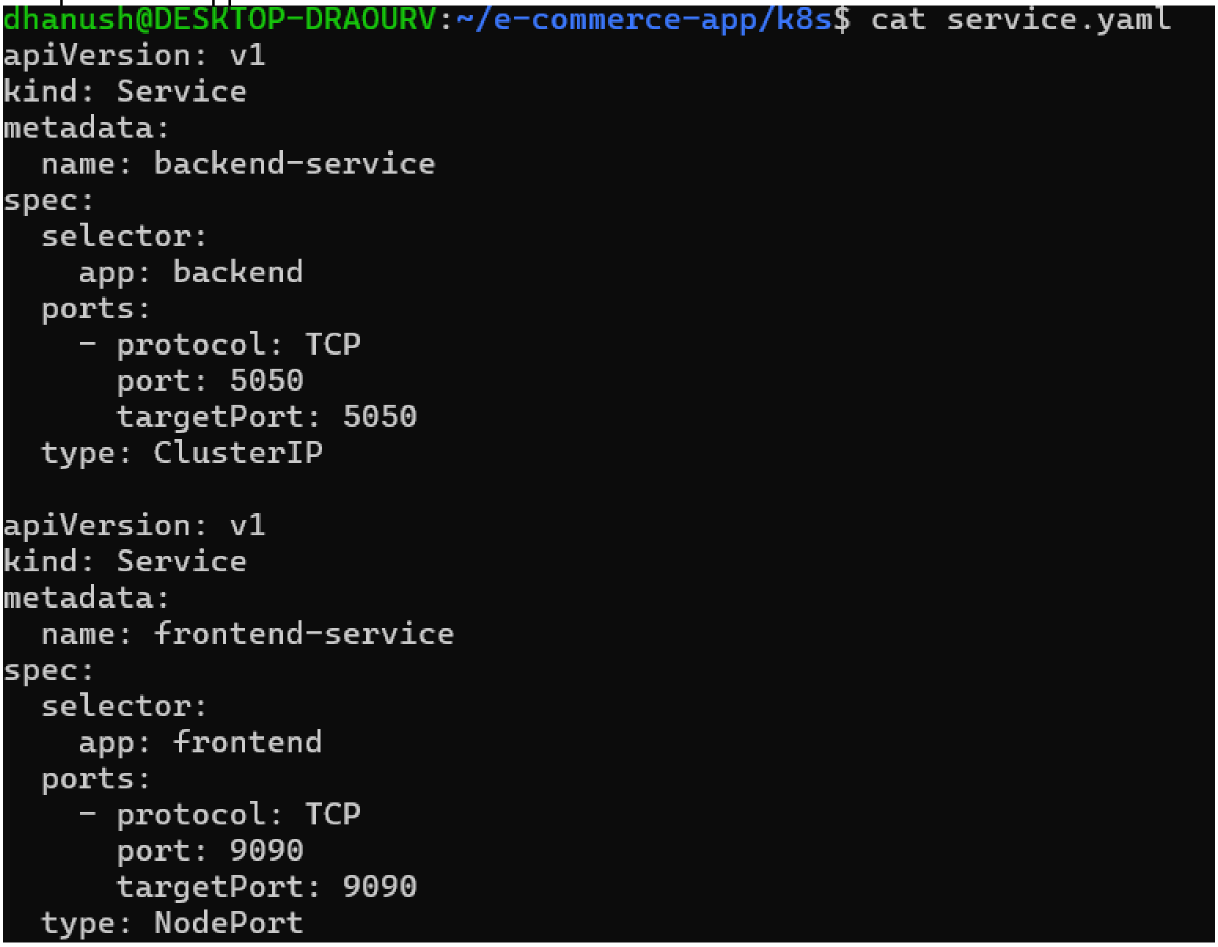
```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat frontend-deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
      - name: frontend
        image: frontend:latest
        ports:
        - containerPort: 9000
```



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  selector:
    app: backend
  ports:
    - protocol: TCP
      port: 5050
      targetPort: 5050
  type: ClusterIP

apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  ports:
    - protocol: TCP
      port: 9090
      targetPort: 9090
  type: NodePort
```

Create **service.yaml** file

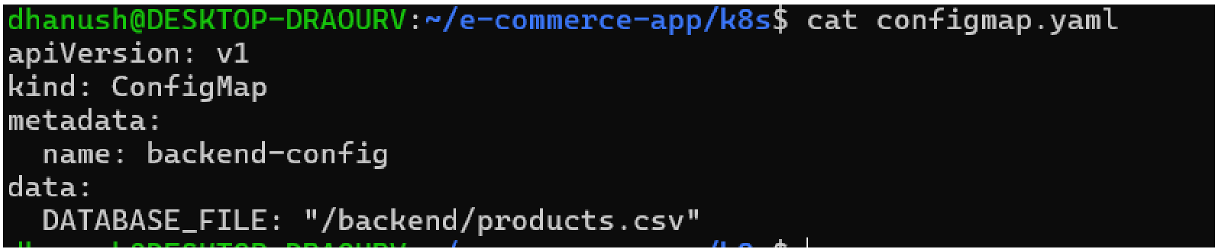
It exposes our application within or outside the cluster.



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  selector:
    app: backend
  ports:
    - protocol: TCP
      port: 5050
      targetPort: 5050
  type: ClusterIP

apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  ports:
    - protocol: TCP
      port: 9090
      targetPort: 9090
  type: NodePort
```

Create **configmap.yaml** file

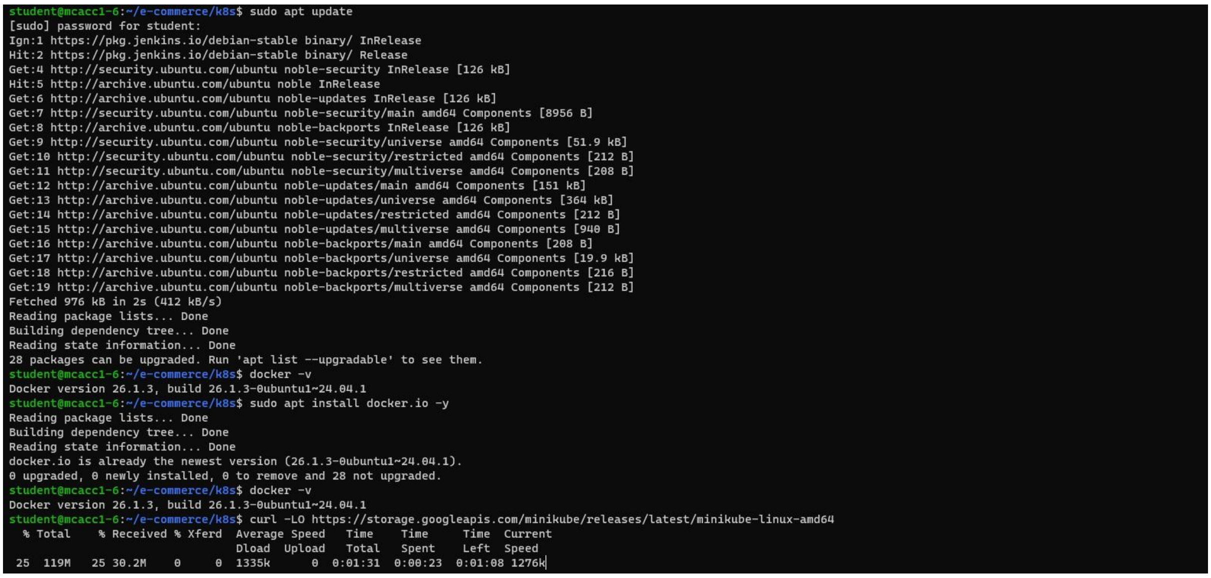
Stores configuration data as key-value pairs.



```
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/k8s$ cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: backend-config
data:
  DATABASE_FILE: "/backend/products.csv"
```

Install minikube

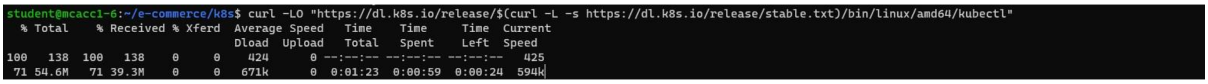
`Minikube` is a tool that allows you to run a Kubernetes cluster locally on our machine. It is designed for developers who want to test and experiment with Kubernetes without needing a full-scale cloud-based cluster.



```
student@mcacc1-6:~/e-commerce/k8s$ sudo apt update
[sudo] password for student:
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8956 B]
Get:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:16 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [19.9 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:19 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 976 kB in 2s (412 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
student@mcacc1-6:~/e-commerce/k8s$ docker -v
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
student@mcacc1-6:~/e-commerce/k8s$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1~24.04.1).
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
student@mcacc1-6:~/e-commerce/k8s$ docker -v
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
student@mcacc1-6:~/e-commerce/k8s$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 25  119M   25 30.2M    0     0  1335k      0  0:01:31  0:00:23  0:01:08 1276k
```

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.

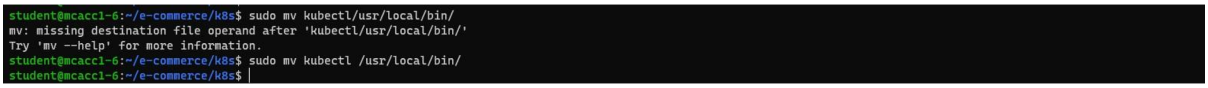


```
student@mcacc1-6:~/e-commerce/k8s$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    424      0 --:--:-- --:--:-- --:--:--   425
 71 54.6M   71 39.3M    0     0   671k      0  0:01:23  0:00:59  0:00:24  594k
```

Grant permission for **kubectl**

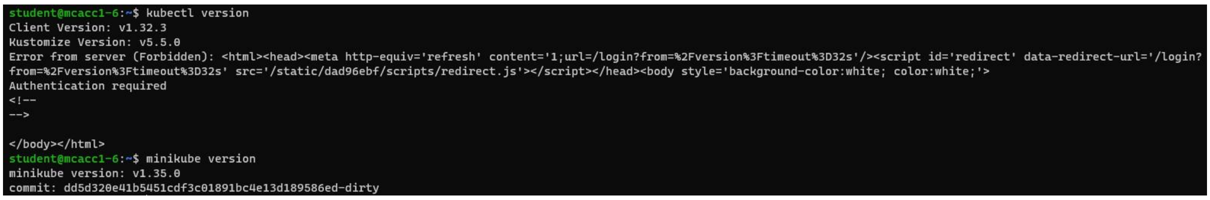
chmod +x kubectl

Move to kubectl to root



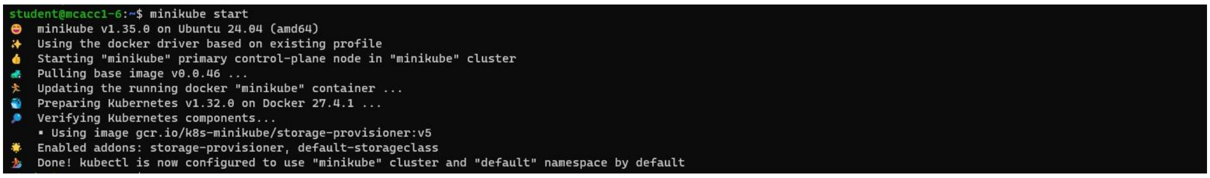
```
student@mcacc1-6:~/e-commerce/k8s$ sudo mv kubectl/usr/local/bin/
mv: missing destination file operand after 'kubectl/usr/local/bin/'
Try 'mv --help' for more information.
student@mcacc1-6:~/e-commerce/k8s$ sudo mv kubectl /usr/local/bin/
student@mcacc1-6:~/e-commerce/k8s$ 
```

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



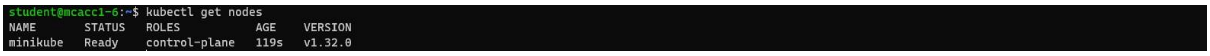
```
student@mcacc1-6:~$ kubectl version
Client Version: v1.32.3
Kustomize Version: v5.5.0
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fversion%3Ftimeout%3D32s'/><script id='redirect' data-redirect-url='/login?
from=%2Fversion%3Ftimeout%3D32s' src='/static/dad96ebf/scripts/redirect.js'></script></head><body style='background-color:white; color:white;'>
Authentication required
<!--
-->

</body></html>
student@mcacc1-6:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
```

Start minicube: **minikube start**



```
student@mcacc1-6:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🏃  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Verify minikube is running



```
student@mcacc1-6:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   119s   v1.32.0
```

Load the image to the minikube

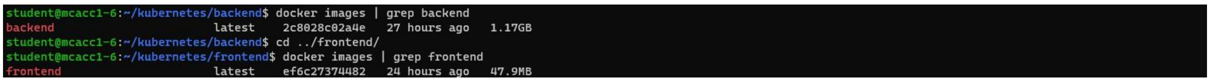
Befor loading images

Perform this commend: eval $(minikube docker-env)

minikube image load frontend:latest

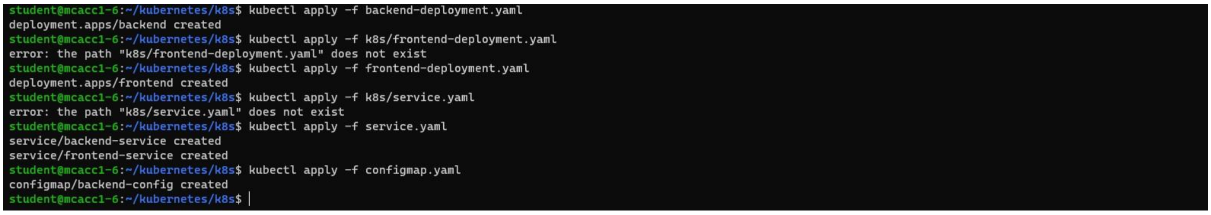
minikube image load backend:latest

Check the images are loaded



```
student@mcacc1-6:~/kubernetes/backend$ docker images | grep backend
backend                        latest    2c8028c02a4e   27 hours ago   1.17GB
student@mcacc1-6:~/kubernetes/backend$ cd ../frontend/
student@mcacc1-6:~/kubernetes/frontend$ docker images | grep frontend
frontend                       latest    ef6c27374482   24 hours ago   47.9MB
```

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.

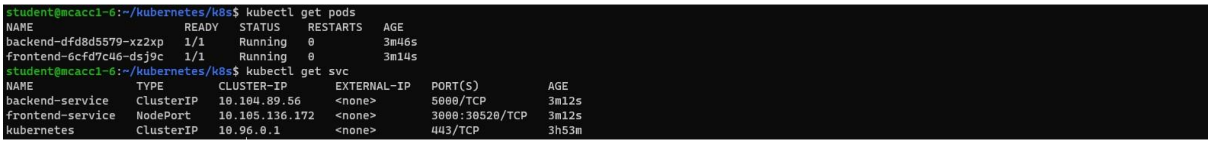


```
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f k8s/frontend-deployment.yaml
error: the path "k8s/frontend-deployment.yaml" does not exist
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f k8s/service.yaml
error: the path "k8s/service.yaml" does not exist
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f service.yaml
service/backend-service created
service/frontend-service created
student@mcacc1-6:~/kubernetes/k8s$ kubectl apply -f configmap.yaml
configmap/backend-config created
student@mcacc1-6:~/kubernetes/k8s$ |
```

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

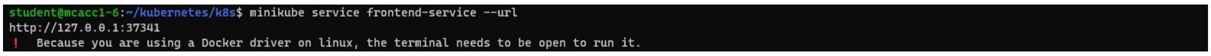
kubectl get pods

kubectl get svc

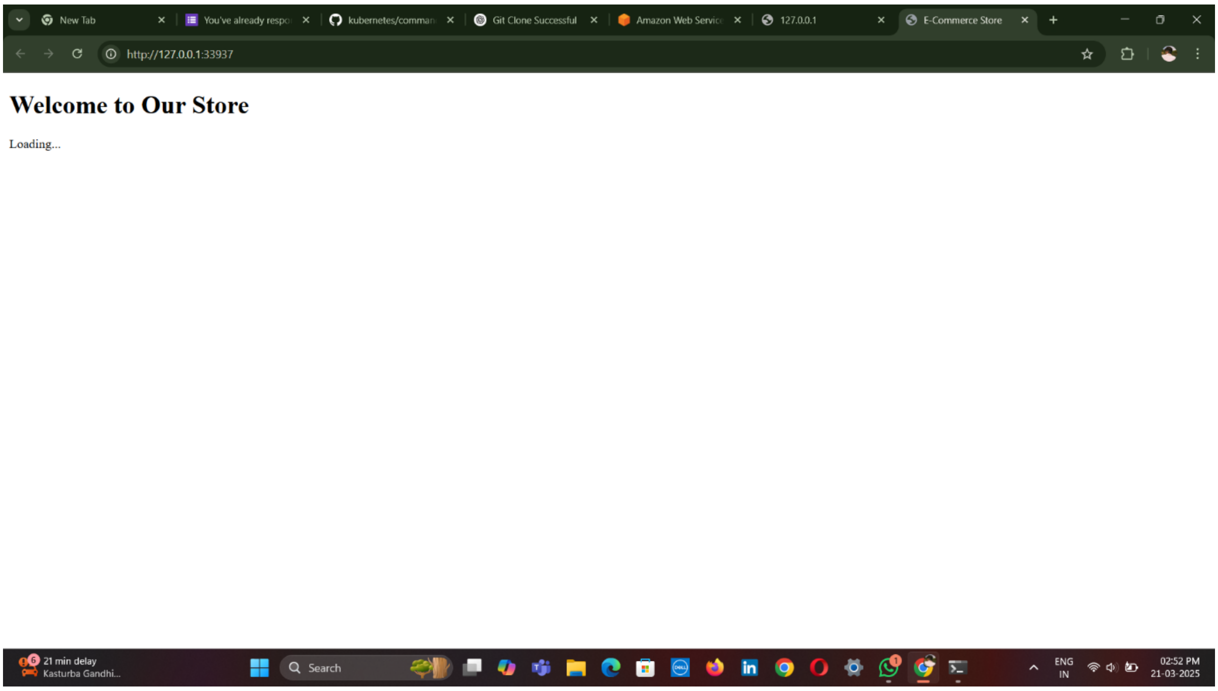


```
student@mcacc1-6:~/kubernetes/k8s$ kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-xz2xp     1/1     Running   0          3m46s
frontend-6cfd7c46-dsj9c     1/1     Running   0          3m14s
student@mcacc1-6:~/kubernetes/k8s$ kubectl get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
backend-service    ClusterIP   10.104.89.56     <none>        5000/TCP         3m12s
frontend-service   NodePort    10.105.136.172   <none>        3000:30520/TCP   3m12s
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP          3h53m
```

To test Frontend



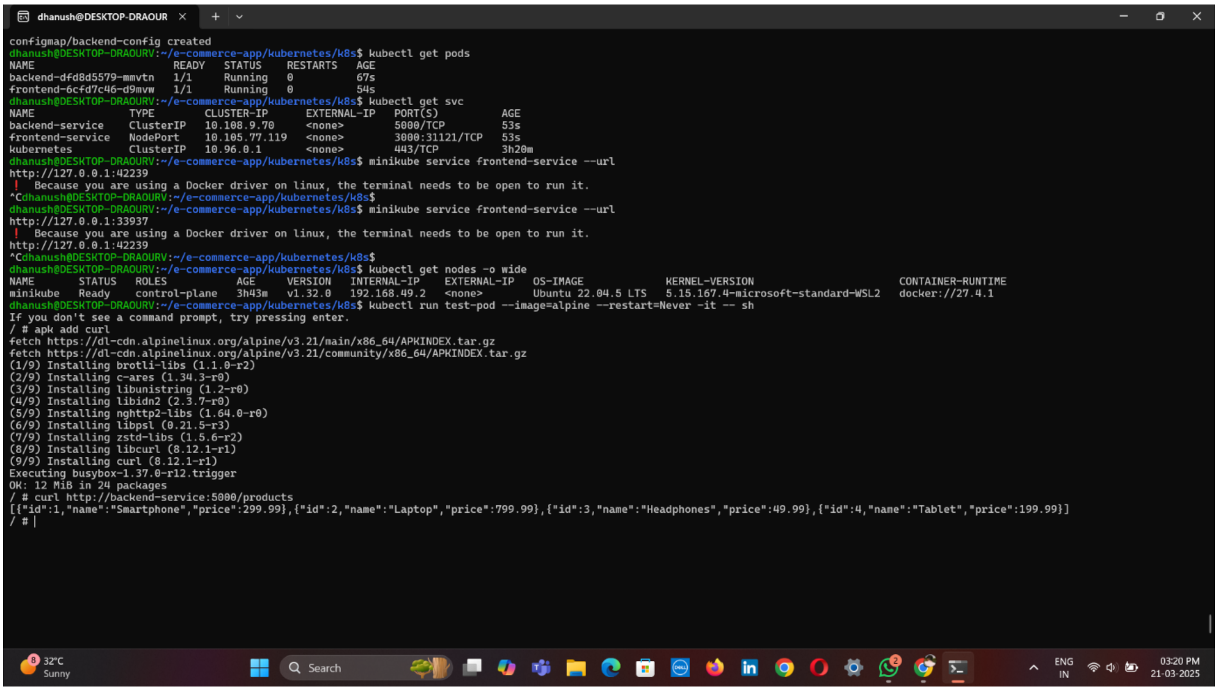
```
student@mcacc1-6:~/kubernetes/k8s$ minikube service frontend-service --url
http://127.0.0.1:37341
!   Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```



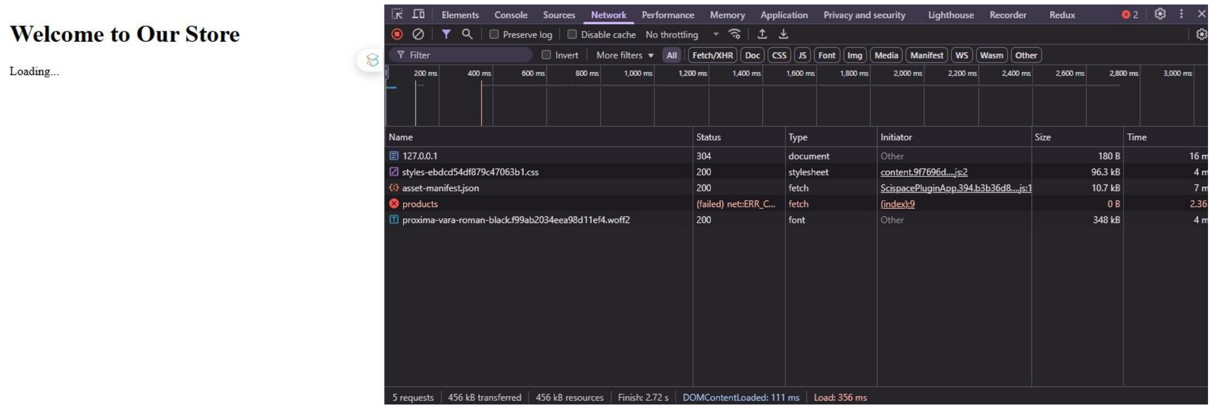
Welcome to Our Store

Loading...

To Test backend



```
configmap/backend-config created
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-mmvtn     1/1     Running   0          67s
frontend-6cfd7c46-d9mvw     1/1     Running   0          54s
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ kubectl get svc
NAME               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
backend-service    ClusterIP   10.108.9.70     <none>        5000/TCP         53s
frontend-service   NodePort    10.105.77.119   <none>        3000:31121/TCP   53s
kubernetes         ClusterIP   10.96.0.1       <none>        443/TCP          3h20m
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ minikube service frontend-service --url
http://127.0.0.1:42239
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^Cdhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ minikube service frontend-service --url
http://127.0.0.1:33937
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
http://127.0.0.1:42239
^Cdhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ kubectl get nodes -o wide
NAME       STATUS   ROLES           AGE     VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION                       CONTAINER-RUNTIME
minikube   Ready    control-plane   3h43m   v1.32.0   192.168.49.2   <none>        Ubuntu 22.04.5 LTS   5.15.167.4-microsoft-standard-WSL2   docker://27.4.1
dhanush@DESKTOP-DRAOURV:~/e-commerce-app/kubernetes/k8s$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # 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":"Headphones","price":49.99},{"id":4,"name":"Tablet","price":199.99}]
/ # |
```



Welcome to Our Store

Loading...

Name	Status	Type	Initiator	Size	Time
127.0.0.1	304	document	Other	180 B	16 m
styles-ebdcd54df879c47063b1.css	200	stylesheet	content.9f7696d....js:2	96.3 kB	4 m
asset-manifest.json	200	fetch	ScispacePluginApp.394.b3b36d8....js:1	10.7 kB	7 m
products	(failed) net::ERR_C...	fetch	(index):9	0 B	2.36
proxima-vara-roman-black.f99ab2034eea98d11ef4.woff2	200	font	Other	348 kB	4 m

5 requests | 456 kB transferred | 456 kB resources | Finish: 2.72 s | DOMContentLoaded: 111 ms | Load: 356 ms

Note: We expect this kind of output because we are running this frontend on localhost.

— COMPLETED —