Day 3: Kubernetes

Create a directory 'e-commerce' and its required folders and files

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

```
student@mcacc1-6:~/docker-python-app/jenkins-docker-demo$ cd ..
student@mcacc1-6:~/docker-python-app$ cd ..
student@mcacc1-6:~$ mkdir e-commerce
student@mcacc1-6:~$ ls
docker-python-app  e-commerce
student@mcacc1-6:~$ cd e-commerce/
student@mcacc1-6:~/e-commerce$ mkdir frontend backend
student@mcacc1-6:~/e-commerce$ ls
backend  frontend
student@mcacc1-6:~/e-commerce$ cd backend/
student@mcacc1-6:~/e-commerce/backend$ nano products.csv
student@mcacc1-6:~/e-commerce/backend$ cat products.csv
id, name, price, qty
1, pen, 20, 100
2, book, 400, 50
3, laptop, 50000, 5,
4, shirt, 500, 50,
5, pants, 750, 50,

student@mcacc1-6:~/e-commerce/backend$ ls
products.csv
student@mcacc1-6:~/e-commerce/backend$ nano app.py
student@mcacc1-6:~/e-commerce/backend$ nano products.csv
student@mcacc1-6:~/e-commerce/backend$ ls
products.csv
student@mcacc1-6:~/e-commerce/backend$ nano app.py
student@mcacc1-6:~/e-commerce/backend$ nano products.csv
student@mcacc1-6:~/e-commerce/backend$ nano products.csv
student@mcacc1-6:~/e-commerce/backend$ cat products.csv
id,name,price,qty
1,pen,20,100
2,book,400,50
3,laptop,50000,5
4,shirt,500,50
5,pants,750,50
```

```python
student@mcacc1-6:~/e-commerce/backend$ cat app.py
import pandas as pd
from flask import Flask

app = Flask(__name__)

@app.route("/products", method=['GET'])
def read_data():
        df = pd.read_csv("./products.csv")
        print(df.head())
        json_data = df.to_json()
        print(json_data)
        return json_data

if __name__ == "__main__":
        app.run(host="0.0.0.0", port=7000)
```

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.

```
student@mcacc1-6:~/e-commerce/backend$ nano requirements.txt
student@mcacc1-6:~/e-commerce/backend$ 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.

```
student@mcacc1-6:~/e-commerce/backend$ nano docker-compose.yml
student@mcacc1-6:~/e-commerce/backend$ cat docker-compose.yml
version: '3.8'

services:
  web:
    build: .
    ports:
      - "7000:7000"
    volumes:
      - .:/app
    restart: always
```

Build Docker image

Sudo docker build -t backend:latest

```
student@mcacc1-6:~/e-commerce/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  6.144kB
Step 1/7 : FROM python:3.11
 ---> 18c0f2265fd9
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> fef9babcc368
Step 3/7 : COPY requirements.txt .
 ---> e25d384df2aa
Step 4/7 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Running in e75e42a7a000
Collecting flask (from -r requirements.txt (line 1))
  Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting pandas (from -r requirements.txt (line 2))
  Downloading pandas-2.2.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (89 kB)
                                  89.9/89.9 kB 4.1 MB/s eta 0:00:00
Collecting Werkzeug>=3.1 (from flask->-r requirements.txt (line 1))
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask->-r requirements.txt (line 1))
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting itsdangerous>=2.2 (from flask->-r requirements.txt (line 1))
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask->-r requirements.txt (line 1))
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting blinker>=1.9 (from flask->-r requirements.txt (line 1))
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting numpy>=1.23.2 (from pandas->-r requirements.txt (line 2))
  Downloading numpy-2.2.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (62 kB)
                                  62.0/62.0 kB 29.0 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas->-r requirements.txt (line 2))
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas->-r requirements.txt (line 2))
  Downloading pytz-2025.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas->-r requirements.txt (line 2))
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask->-r requirements.txt (line 1))
  Downloading MarkupSafe-3.0.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas->-r requirements.txt (line 2))
```
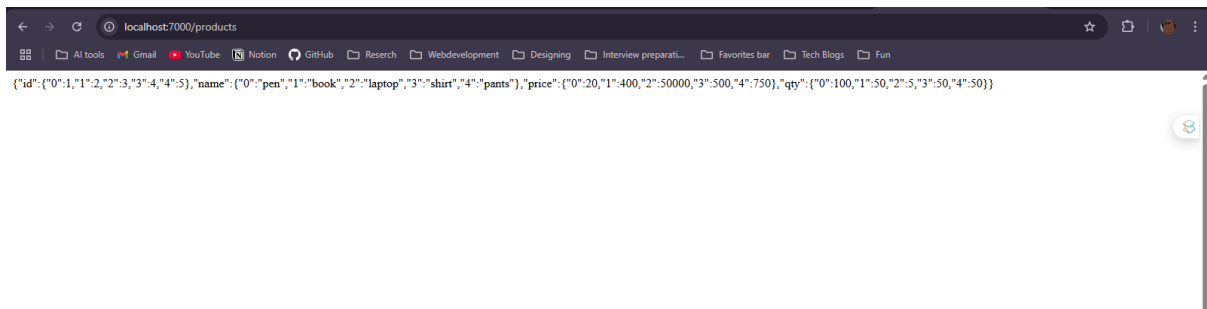
Run the docker:

```
 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 7000/products

{"id":{"0":1,"1":2,"2":3,"3":4,"4":5},"name":{"0":"pen","1":"book","2":"laptop","3":"shirt","4":"pants"},"price":{"0":20,"1":400,"2":50000,"3":500,"4":750},"qty":{"0":100,"1":50,"2":5,"3":50,"4":50}}

The JSON data is displayed at our port: 7000/products.

Create a container in frontend

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

Build the image using the command:

sudo docker build -t frontend:latest.

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.

```
student@mcacc1-6:~/e-commerce/k8s$ cat backend-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - name: backend
        image: backend:latest
        ports:
        - containerPort: 7000
student@mcacc1-6:~/e-commerce/k8s$ cat frontend-deployment.yaml
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: 7500
student@mcacc1-6:~/e-commerce/k8s$
```

Create **service.yaml** file

It exposes our application within or outside the cluster.

```
student@mcacc1-6:~/e-commerce/k8s$ nano service.yaml
student@mcacc1-6:~/e-commerce/k8s$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
spec:
  selector:
    app: backend
  ports:
    - protocol: TCP
      port: 7000
      targetPort: 7000
  type: ClusterIP

apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  ports:
    - protocol: TCP
      port: 7500
      targetPort: 7500
  type: NodePort
```

Create **configmap.yaml** file

Stores configuration data as key-value pairs.

```
student@mcacc1-6:~/e-commerce/k8s$ nano configmap.yaml
student@mcacc1-6:~/e-commerce/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$
```