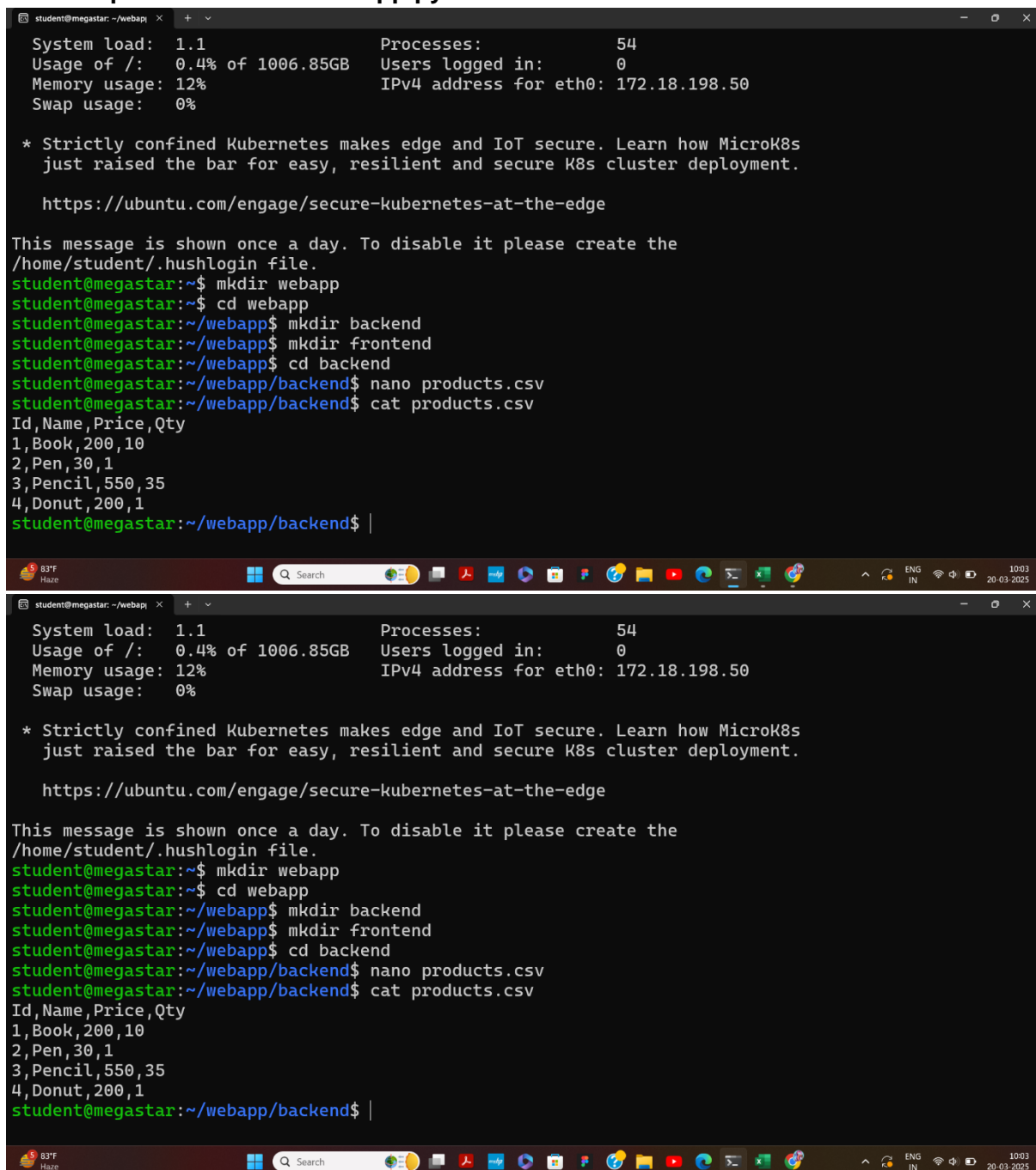


# Day 3 and Day 4: Kubernetes

Create a directory 'Web app' and its required folders and files

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



The image displays two screenshots of a terminal window on a Linux system, showing the steps to create a web application directory structure and a CSV file.

**Terminal Screenshot 1:**

```
student@megastar: ~/webapp
System load: 1.1          Processes: 54
Usage of /: 0.4% of 1006.85GB  Users logged in: 0
Memory usage: 12%        IPv4 address for eth0: 172.18.198.50
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/student/.hushlogin file.
student@megastar:~$ mkdir webapp
student@megastar:~$ cd webapp
student@megastar:~/webapp$ mkdir backend
student@megastar:~/webapp$ mkdir frontend
student@megastar:~/webapp$ cd backend
student@megastar:~/webapp/backend$ nano products.csv
student@megastar:~/webapp/backend$ cat products.csv
Id,Name,Price,Qty
1,Book,200,10
2,Pen,30,1
3,Pencil,550,35
4,Donut,200,1
student@megastar:~/webapp/backend$ |
```

**Terminal Screenshot 2:**

```
student@megastar: ~/webapp
System load: 1.1          Processes: 54
Usage of /: 0.4% of 1006.85GB  Users logged in: 0
Memory usage: 12%        IPv4 address for eth0: 172.18.198.50
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/student/.hushlogin file.
student@megastar:~$ mkdir webapp
student@megastar:~$ cd webapp
student@megastar:~/webapp$ mkdir backend
student@megastar:~/webapp$ mkdir frontend
student@megastar:~/webapp$ cd backend
student@megastar:~/webapp/backend$ nano products.csv
student@megastar:~/webapp/backend$ cat products.csv
Id,Name,Price,Qty
1,Book,200,10
2,Pen,30,1
3,Pencil,550,35
4,Donut,200,1
student@megastar:~/webapp/backend$ |
```

```

student@mac31-0:~/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)

```

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

To verify the **available port numbers**

```

student@megastar: ~/webapp
Successfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 itsdangerou
s-2.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system pa
ckage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
---> Removed intermediate container e991af9c5d82
---> 0f89a5b62654
Step 5/7 : COPY . .
---> 1a38ff877e73
Step 6/7 : EXPOSE 7000
---> Running in 174f5af258d2
---> Removed intermediate container 174f5af258d2
---> da35386fca27
Step 7/7 : CMD ["python", "app.py"]
---> Running in 8d40b29e7ca5
---> Removed intermediate container 8d40b29e7ca5
---> 0978980e4ec3
Successfully built 0978980e4ec3
Successfully tagged backend:latest
student@megastar:~/webapp/backend$ sudo docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
backend              latest      0978980e4ec3  18 seconds ago 1.03GB
sugumar3232/docker-app latest      943c48aa9e31  20 hours ago  1.04GB
test                 latest      77d678a7a322  40 hours ago  1.03GB
<none>               <none>      c54a35aa3814  40 hours ago  1.03GB
<none>               <none>      1e948401f938  40 hours ago  1.03GB
<none>               <none>      a1c9c7632711  43 hours ago  1.03GB
python               3.11       18c0f2265fd9  3 months ago  1.01GB
student@megastar:~/webapp/backend$

```

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.

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@megastar: ~/webapp$ sudo docker build -t backend:latest
Successfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 itsdangerou
s-2.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system pa
ckage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
---> Removed intermediate container e991af9c5d82
---> 0f89a5b62654
Step 5/7 : COPY . .
---> 1a38ff877e73
Step 6/7 : EXPOSE 7000
---> Running in 174f5af258d2
---> Removed intermediate container 174f5af258d2
---> da35386fca27
Step 7/7 : CMD ["python", "app.py"]
---> Running in 8d40b29e7ca5
---> Removed intermediate container 8d40b29e7ca5
---> 0978980e4ec3
Successfully built 0978980e4ec3
Successfully tagged backend:latest
student@megastar:~/webapp/backend$ sudo docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
backend              latest      0978980e4ec3  18 seconds ago 1.03GB
sugumar3232/docker-app latest      943c48aa9e31  20 hours ago  1.04GB
test                 latest      77d678a7a322  40 hours ago  1.03GB
<none>               <none>      c54a35aa3814  40 hours ago  1.03GB
<none>               <none>      1e948401f938  40 hours ago  1.03GB
<none>               <none>      a1c9c7632711  43 hours ago  1.03GB
python               3.11       18c0f2265fd9  3 months ago  1.01GB
student@megastar:~/webapp/backend$
```

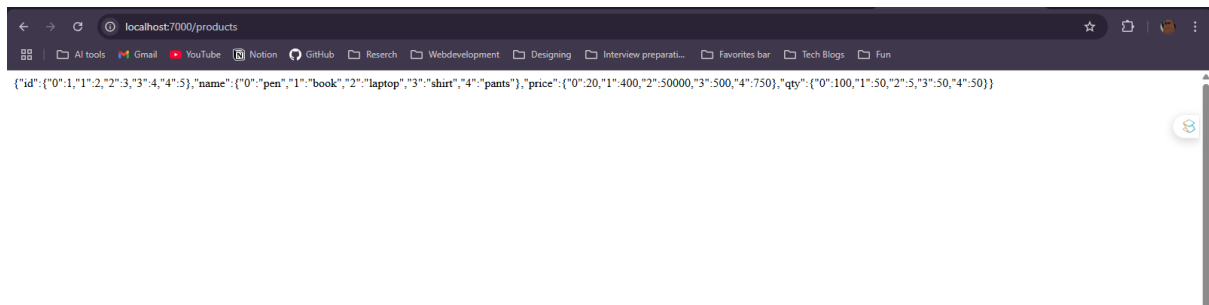
Run the docker:

```
sudo docker run -d -p 7000:7000 backend:latest
```

```
sudo docker logs <Generated number>
```

```
student@megastar: ~/webapp$ sudo docker run -d -p 7000:7000 backend
172.17.0.1 - - [20/Mar/2025 06:19:55] "GET /products HTTP/1.1" 500 -
student@megastar:~/webapp/backend$ nano app.py
student@megastar:~/webapp/backend$ docker build -t backend .
DEPRECATED
student@megastar:~/webapp/backend$ sudo docker logs 0978980e4ec3
Sending build context to Docker daemon 18.00kB
Step 1/7 : FROM python:3.11
---> Using cache
---> 5e3ff
Step 2/7 : WORKDIR /app
---> Using cache
---> 8cfc
Step 3/7 : COPY . .
---> Using cache
---> 8cfc
Step 4/7 : EXPOSE 7000
---> Using cache
---> 8cfc
Step 5/7 : CMD ["python", "app.py"]
---> Using cache
---> 8cfc
Step 6/7 :
---> Running
---> Removed intermediate container 8cfc
---> 8cfc
Step 7/7 :
---> Running
---> Removed intermediate container 8cfc
---> 8cfc
Successfully built 0978980e4ec3
Successfully tagged backend:latest
student@megastar:~/webapp/backend$ docker run -p 7000:7000 backend
* 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://172.17.0.1:7000
* Running on http://172.17.0.2:7000
Press CTRL-C to quit
172.17.0.1 - - [20/Mar/2025 06:23:19] "GET /products HTTP/1.1" 200 -
```

Run the application in the 7000/products



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

## Create a container in frontend

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

```
student@megastar: ~/webapp$ stop
student@megastar: ~/webapp$ exit

^Cstudent@megastar:~/webapp/backend$ cd ..
student@megastar:~/webapp$ ls
backend frontend
student@megastar:~/webapp$ cd frontend
student@megastar:~/webapp/frontend$ nano index.html
student@megastar:~/webapp/frontend$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>E-Commerce Store</title>
  <script>
    async function fetchProducts() {
      const response = await fetch("http://localhost:7000/products");
      const products = await response.json();
      let output = "<h2>Product List</h2><ul>";
      products.forEach(product => {
        output += '<li>${product.name} - ${product.price}</li>';
      });
      output += "</ul>";
      document.getElementById("product-list").innerHTML = output;
    }
  </script>
</head>
<body onload="fetchProducts()">
  <h1>Welcome to Our Store</h1>
  <div id="product-list">Loading...</div>
</body>
</html>
student@megastar:~/webapp/frontend$
```

Build the image using the command:

sudo docker build -t frontend:latest.

```
student@mcaccl-6:~/e-commerce/frontend$ sudo docker build -t frontend:latest .
[sudo] password for student:
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
ccc35e35d428: Pull complete
43f2ec468bdf: Pull complete
984583bcbf83: Pull complete
8d27c972a58f: Pull complete
ab3286a73463: Pull complete
6d79cc6084d4: Pull complete
8c7e4c092ab7: 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
--> ef6c27374482
Successfully built ef6c27374482
Successfully tagged 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@megastar: ~/webapp  x  +  v
8d27c972a58f: Pull complete
ab3286a73463: Pull complete
6d79cc6084d4: Pull complete
8c7e4c092ab7: 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
--> aca3b9b58a97
Successfully built aca3b9b58a97
Successfully tagged frontend:latest
student@megastar:~/webapp/frontend$ cd ..
student@megastar:~/webapp$ mkdir k8s
student@megastar:~/webapp$ cd k8s
student@megastar:~/webapp/k8s$ nano backend-deployment.yaml
student@megastar:~/webapp/k8s$ nano backend-deployment.yaml
student@megastar:~/webapp/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@megastar:~/webapp/k8s$ |
```

Create **service.yaml** file

It exposes our application within or outside the cluster.

```
student@megastar: ~/webapp$ nano frontend-deployment.yaml
GNU nano 7.2 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: 3000

[ line 1/20 ( 5%), col 1/20 ( 5%), char 0/322 ( 0%) ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^I Execute    ^C Location   ^U Undo       ^M-A Set Mark ^M-J To Bracket
^X Exit      ^R Read File  ^L Replace    ^U Paste      ^J Justify    ^/_ Go To Line ^E Redo       ^M-G Copy     ^_Q Where Was

94°F Sunny
```

Create **configmap.yaml** file

Stores configuration data as key-value pairs.

```
student@megastar: ~/e-commerce/k8s$ nano configmap.yaml
student@megastar: ~/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@megastar: ~/webapp$ nano configmap.yaml
student@megastar: ~/webapp$ cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: backend-config
data:
  DATABASE_FILE: "/backend/products.csv"
student@megastar: ~/webapp$ sudo apt update -y
[sudo] password for student:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:4 https://pkg.jenkins.io/debian-stable binary/ Release
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/restricted 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 8s (121 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
29 packages can be upgraded. Run 'apt list --upgradable' to see them.
student@megastar: ~/webapp$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
student@megastar: ~/webapp$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
student@megastar: ~/webapp$ minikube version
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
student@megastar: ~/webapp$
```

**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.

```
chmod +x kubectl
```

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

Start minicube: **minikube start**

## Verify minikube is running

## Load the image to the minikube

### Before 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@maccl-6:~/kubernetes/backend$ docker images | grep backend
backend          latest          2c8028c02a4e   27 hours ago   1.17GB
student@maccl-6:~/kubernetes/backend$ cd ../frontend/
student@maccl-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@megastar: ~/kubern
System information as of Fri Mar 21 03:44:01 UTC 2025
System load:  0.99          Processes:      52
Usage of /:   0.5% of 1006.85GB    Users logged in: 0
Memory usage: 12%          IPv4 address for eth0: 172.18.198.50
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/student/.hushlogin file.
student@megastar:~$ git clone "https://github.com/PadmavathyNarayanan/jenkins-docker-demo.git"
Cloning into 'jenkins-docker-demo'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (89/89), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 89 (delta 45), reused 9 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (89/89), 28.05 KiB | 129.00 KiB/s, done.
Resolving deltas: 100% (45/45), done.
student@megastar:~$ ls
config.xml  docker-python-app  jenkins-docker-demo  webapp
student@megastar:~$ git clone "https://github.com/PadmavathyNarayanan/kubernetes.git"
Cloning into 'kubernetes'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 22 (delta 2), reused 15 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (22/22), 4.41 KiB | 4.41 MiB/s, done.
Resolving deltas: 100% (2/2), done.
student@megastar:~$ ls
config.xml  docker-python-app  jenkins-docker-demo  kubernetes  webapp
student@megastar:~$ cd kubernetes
student@megastar:~/kubernetes$ ls
README.md  backend  commands-to-stop-instances  frontend  k8s
student@megastar:~/kubernetes$ kubectl delete all --all --force --grace-period=0
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
```

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

**kubectl get pods**

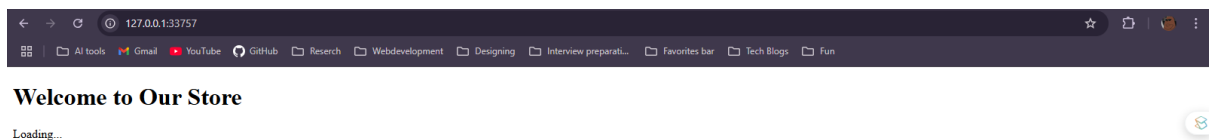
**kubectl get svc**

```
student@maccl-6:~/kubernetes/k8s$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
backend-qfd8d5579-xz2xp   1/1     Running   0           3m46s
frontend-6cfd7c46-dsj9c   1/1     Running   0           3m14s
student@maccl-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    ClusterIP   10.195.136.172 <none>        3000:30520/TCP  3m12s
kubernetes          ClusterIP   10.96.0.1    <none>        443/TCP          3h53m
```

## To test Frontend

```
student@maccl-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.
```



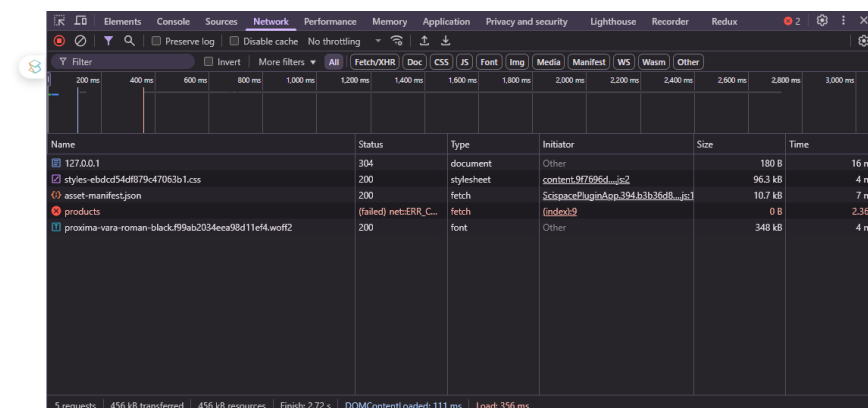


## To Test backend

```
student@maccl1-8:~/kubernetes/k8s$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # kubectl get pod test-pod
sh: kubectl: not found
/ # kubectl exec -it test-pod -- sh
sh: kubectl: not found
/ # 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":{"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}}/ #
```

## Welcome to Our Store

Loading...



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

— COMPLETED —