Deploy a Fullstack Application Using Kubernetes and Ansible Playbook

Why Ansible Playbook?

An Ansible Playbook is a YAML file that defines a series of tasks and configurations to be executed automatically on remote systems. It is used to manage, configure, deploy, and automate systems in a consistent and repeatable way using Ansible.

- Automates repetitive tasks: installs Docker, starts Minikube, deploys Kubernetes manifests, and forwards ports automatically.
- Saves time: no need to run multiple shell commands manually.
- Handles dependencies: checks Docker status, copies manifests, waits for pods.
- Background operations: forwards ports so apps stay accessible even if the terminal closes.

STEP1

Image Creation Source Link

https://drive.google.com/drive/folders/1VpP7jcqpCvWNXROi2krgGynkJqcorgtq?usp=sharing

Open the Docker Desktop & Docker Hub. Build and push the images using following commands.

1 Build Docker images

Make sure you are in the directory containing your Dockerfiles.

Backend

```
docker build -f Dockerfile.backend -t srithar1234/ecommerce-backend1:v1 .
```

Frontend

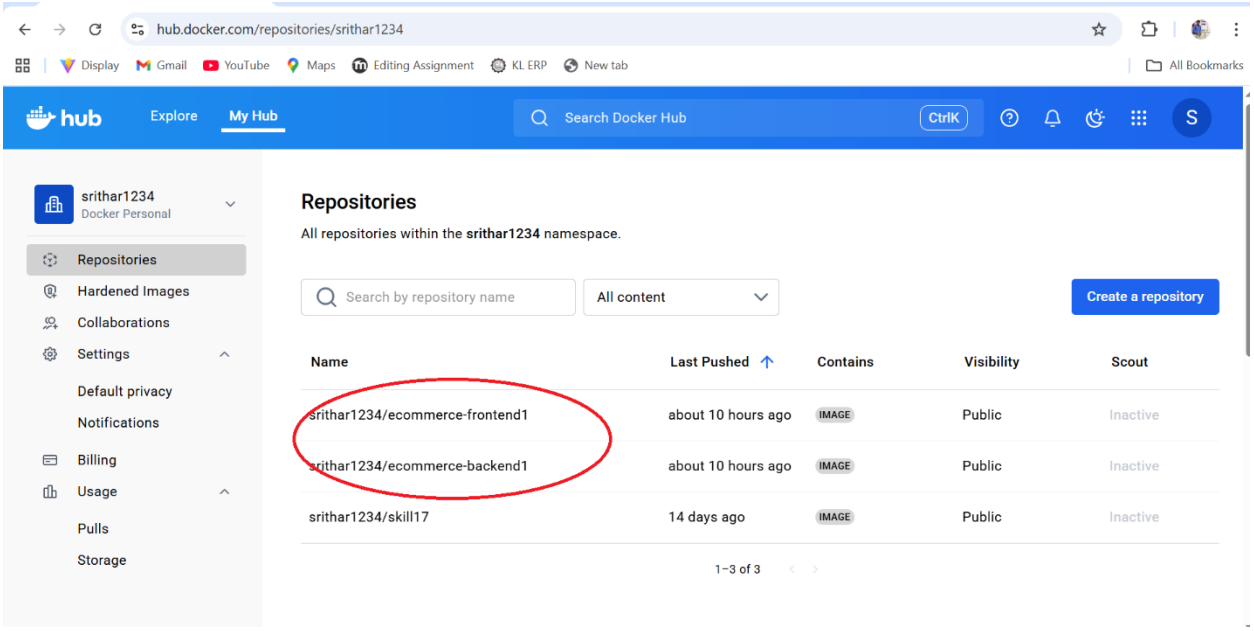
```
docker build -f Dockerfile.frontend -t srithar1234/ecommerce-frontend1:v1 .
```

---

2 Push images to Docker Hub

```
docker push srithar1234/ecommerce-backend1:v1
```

```
docker push srithar1234/ecommerce-frontend1:v1
```



Now you can close the Docker Desktop.

STEP1

Download the play book

https://drive.google.com/drive/folders/17qICNsL42I55997mLUau5Ke4rLivfSDo?usp=sharing

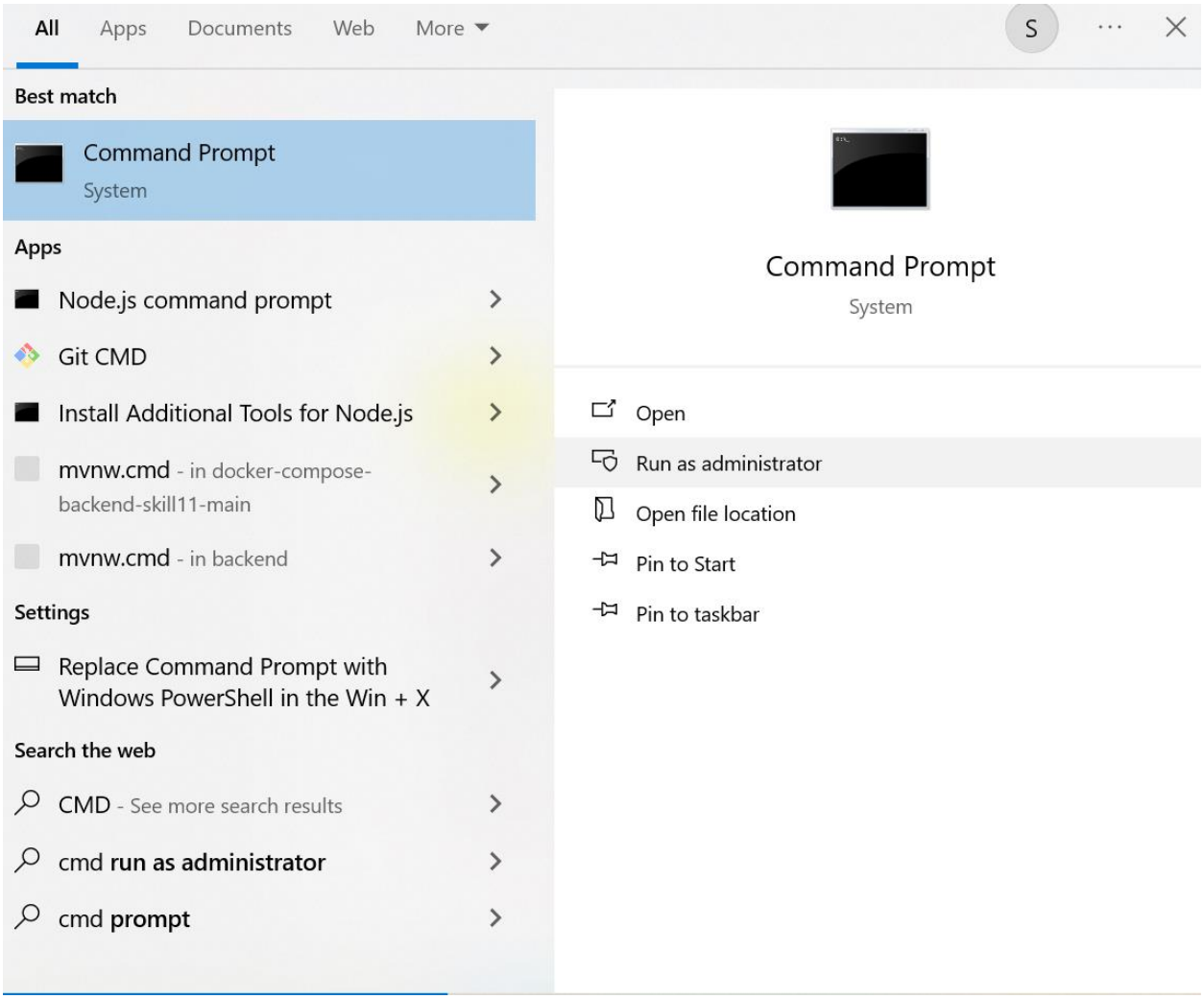
In, playbook/k8s/fullstakdeployment.yaml change the Docker dektop name.

```
    ''
spec:
  initContainers:
    - name: wait-for-mysql
      image: busybox
      command: ['sh', '-c', 'until nc -z mysql 3306; do echo waiting for mysql; sleep 2; done;']
  containers:
    - name: backend
      image: srithar1234/ecommerce-backend1:v1
      ports:
        - containerPort: 8080
      env:
```

```
      app: frontend
  spec:
    containers:
      - name: frontend
        image: srithar1234/ecommerce-frontend1:v1
        ports:
          - containerPort: 8080
        env:
```

STEP3

Open cmd-->Run as administrator



>wsl --install

Username: ubuntu

Password: ubuntu

>cd "/mnt/c/Users/HP/Desktop/practical10/playbook/ansible"

>ls

Note: Whatever changes doing in local machine files will reflect in ubuntu machine.

Install the following dependencies

----

1 Install Ansible

>sudo apt update

>sudo apt install -y ansible

----

2 Install Docker

----

sudo apt update && sudo apt install -y ca-certificates curl gnupg lsb-release && \

sudo mkdir -p /etc/apt/keyrings && \

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg && \

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null && \

sudo apt update && \

sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin && \

sudo service docker start && \

sudo systemctl enable docker && \

```
sudo usermod -aG docker $USER && \
newgrp docker && \
docker --version

# -------------------------
# 3 Install Minikube
# -------------------------
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && \
sudo install minikube-linux-amd64 /usr/local/bin/minikube && \
rm minikube-linux-amd64

# -------------------------
# 4 Start Minikube using Docker driver
# -------------------------
minikube start --driver=docker --memory=2000 --cpus=2 && \
minikube status

# -------------------------
# 5 Install kubectl
# -------------------------
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl" && \
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl && \
rm kubectl && \
kubectl version --client
```

Final run,

>ansible-playbook -i inventory playbook.yaml

Check the status of containers

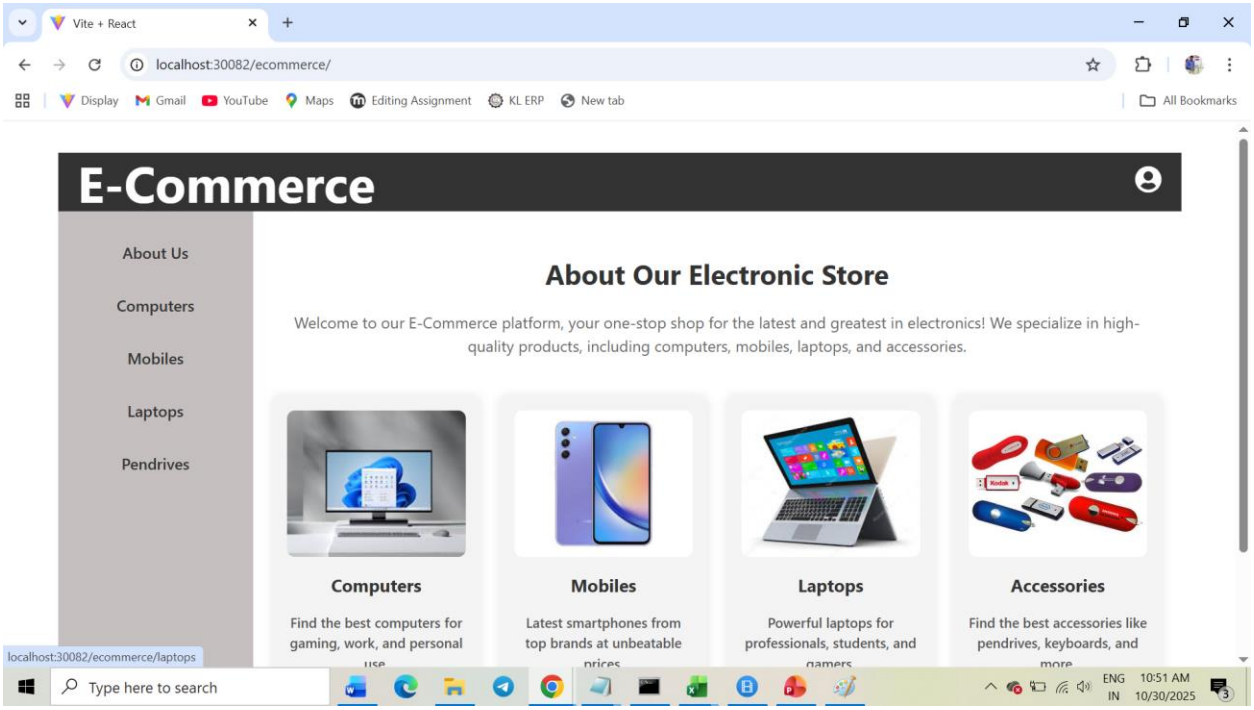
>kubectl get deployments

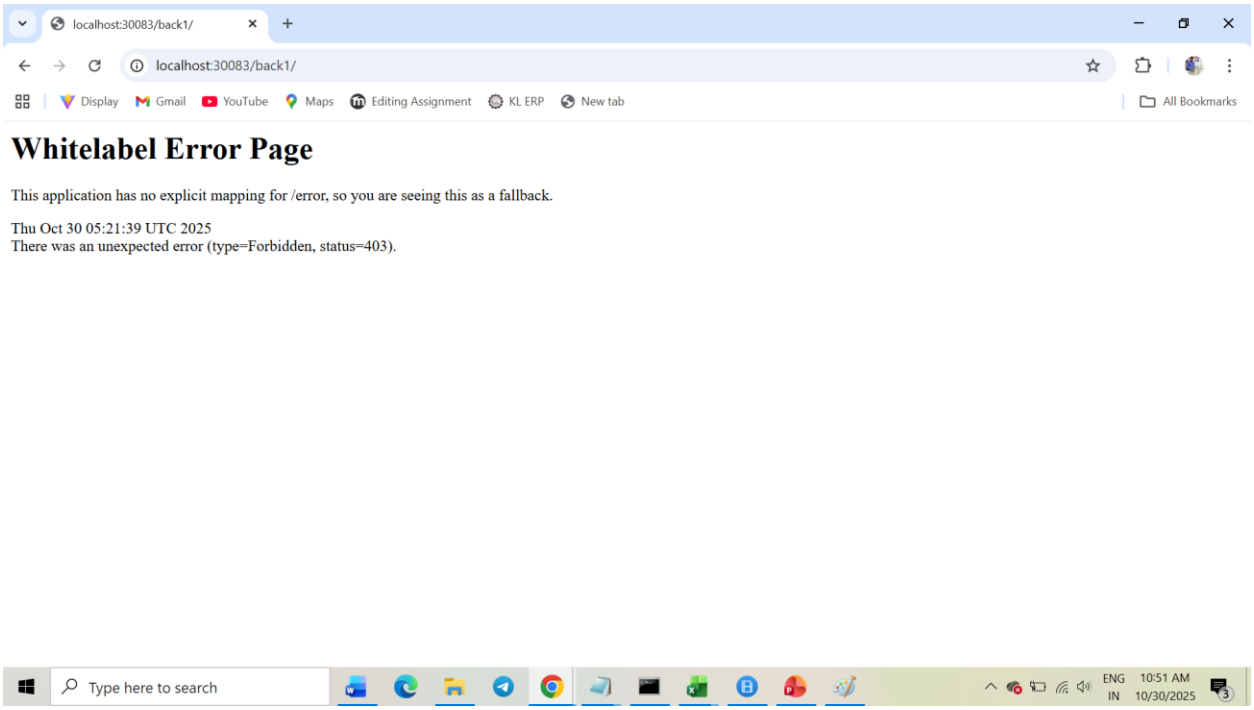
>kubectl get pods

Open the browser and check the deployment in browser,

Frontend: http://localhost:30082/ecommerce/

Backend: http://localhost:30083/back1/





To insert the produts

>kubectl get pods

>kubectl exec -it **mysql-546788b65f-tmdv9** -- mysql -uroot -proot

Now mysql container terminal is opened.

Now execute the following commands in the mysql container terminal.

mysql>SHOW DATABASES;

mysql>USE ecommerce;

mysql>SHOW TABLES;

Query:-

-- Insert Computers

mysql>INSERT INTO ecommerce.products (name, category, price, image_path) VALUES

('Gaming PC', 'computers', 1200.00, 'gaming_pc.jpeg'),

('Office Desktop', 'computers', 800.00, 'office_desktop.jpeg'),

('Mini PC', 'computers', 500.00, 'mini_pc.jpeg'),

('Workstation', 'computers', 2500.00, 'workstation.jpeg');

-- Insert Mobiles

mysql>INSERT INTO ecommerce.products (name, category, price, image_path) VALUES

('iPhone 14', 'mobiles', 999.00, 'iphone_14.jpeg'),

('Samsung Galaxy S23', 'mobiles', 899.00, 'samsung_galaxy_s23.jpeg'),

('Google Pixel 7', 'mobiles', 799.00, 'google_pixel_7.jpeg'),

('OnePlus 11', 'mobiles', 749.00, 'oneplus_11.jpeg');

-- Insert Laptops

mysql>INSERT INTO products (name, category, price, image_path) VALUES

('MacBook Air', 'laptops', 1099.00, 'macbook_air.jpeg'),

('Dell XPS 15', 'laptops', 1299.00, 'dell_xps_15.jpeg'),

('Lenovo ThinkPad', 'laptops', 999.00, 'lenovo_thinkpad.jpeg'),

('HP Spectre x360', 'laptops', 1199.00, 'hp_spectre_x360.jpeg');

-- Insert Pendrives

mysql>INSERT INTO products (name, category, price, image_path) VALUES

('SanDisk 64GB', 'pendrives', 15.00, 'sandisk_64gb.jpeg'),

('Kingston 128GB', 'pendrives', 25.00, 'kingston_128gb.jpeg'),

('Sony 256GB', 'pendrives', 50.00, 'sony_256gb.jpeg'),

('Samsung 512GB', 'pendrives', 80.00, 'samsung_512gb.jpeg');

mysql>select * from products;