IUDX_DEVOPS_Task

Name: **Tejashwini Kottha**
Mail: tejashwinikottha@gmail.com
Mobile: +918179038873

Task Instructions
1) Pick a application/use your existing/easy to do projects from internet with following criteria :
a) A basic HTTP web application in any language which takes some input through an API endpoint and process it and store the data to db
b) The database can be mysql, psql, mongodb etc.
NOTE: We strongly suggest to pick application language and db which you are familiar with for containerisation and deployment and we are not concerned with the code until unless it meets the criteria above.
2) Containerise and deploy the above application components in local machine using docker-compose a) Refer docker docs and docker-compose docs
b) Expose the web application in docker to outside world, so that it can be accessed through something like http://localhost:35622/xyz/
3) Containerise and deploy the above application in local K8s minikube cluster (using kubectl only):-.
a) Please see the instructions on how to create a minikube on a local machine at https://github.com/datakaveri/iudx-deployment/tree/master/K8s-deployment/K8s-cluster/minikube .
b) Add auto scaling to any one of the components - web server or database. c) Expose the web application in K8s to outside world, so that it can be accessed through something like http://localhost:35622/xyz/ i.e. through localhost of host machine

Steps To solve this:

1. Set up Docker — Install docker desktop , activate docker,login.
2. Make sure our flask app runs successfully in host app
3. After make a docker file and build an image for the flask app.
4. Pull a database container and and run
5. Run the flask app container
6. Run the database container
7. Now connect these two containers using same network
8. After connecting , make Generate docker compose file for the setup
9. Setup minikube and kubectl
10. Minikube start
11. Write the deployment files like service file, persistent volume file, cluster ip and nodeport files along with autoscaling and exposing process
12. Deploy those files using cmds
13. Check for the pods, deployments , services whether they are launched or not !!

IUDX_DEVOPS_Task

Used Docker and kubernetes commands are :

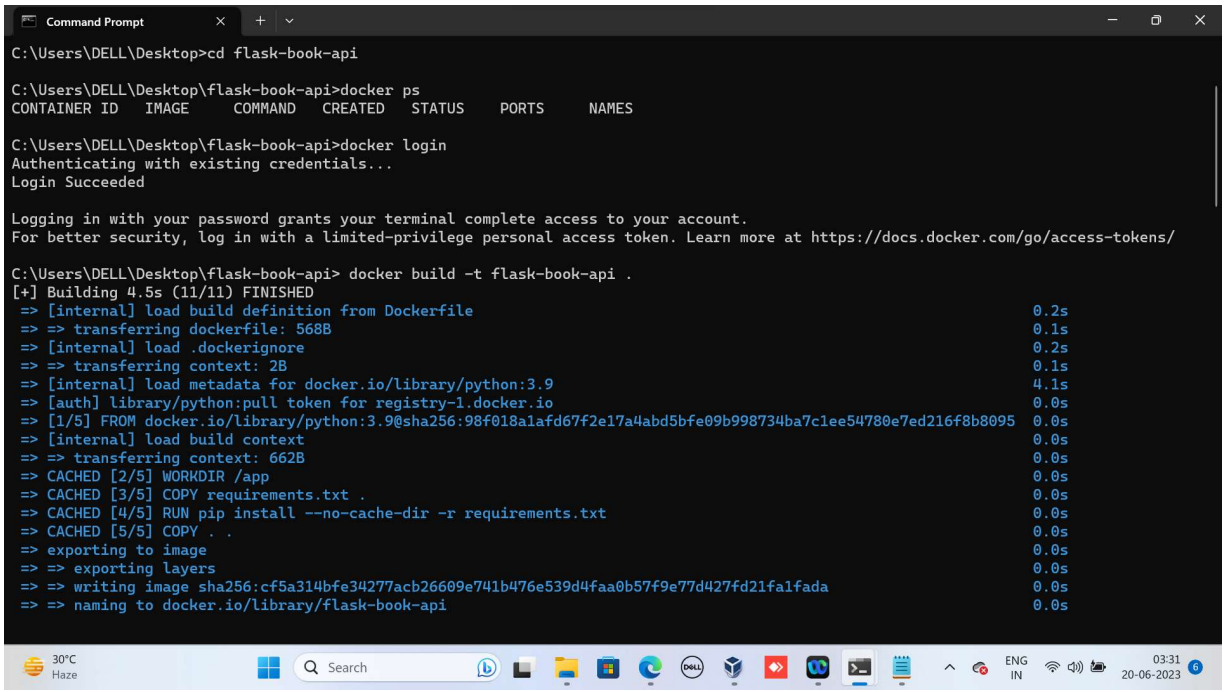
- docker build -t flask-book-api .
- docker run --name flask-book-api -d -p 8000:5000 --rm flask-book-api:latest
- docker network create mynetwork2
- docker run --name sqlite-container1 --network mynetwork2 -v /path/to/your/sqlite/db:/data -d sqlite
- docker run --name flask-app-container1 --network mynetwork2 -p 5000:5000 your-flask-image
- docker volume create myvolume
- docker run --rm -v myvolume:/data -v /path/to/your/sqlite/db:/backup sqlite \cp /backup/mydatabase.db /data/mydatabase.db
- docker run --name new-sqlite-container2 -v myvolume:/data -d keinos/sqlite
- minikube start
- kubectl apply -f flask-app.yaml
- kubectl get deployments
- kubectl get services
- minikube ip
- minikube service flask-app-service

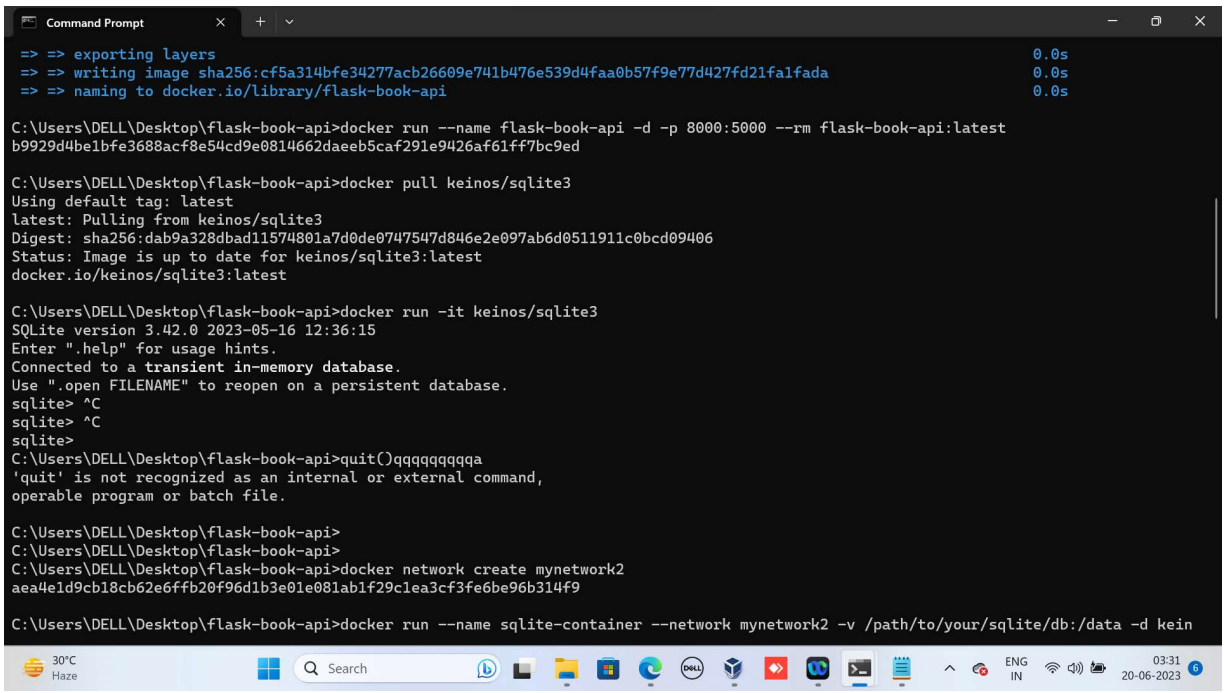
For more detailed code and commands , you can refer github url :

https://github.com/Tejashwini690/IUDX_Task.git

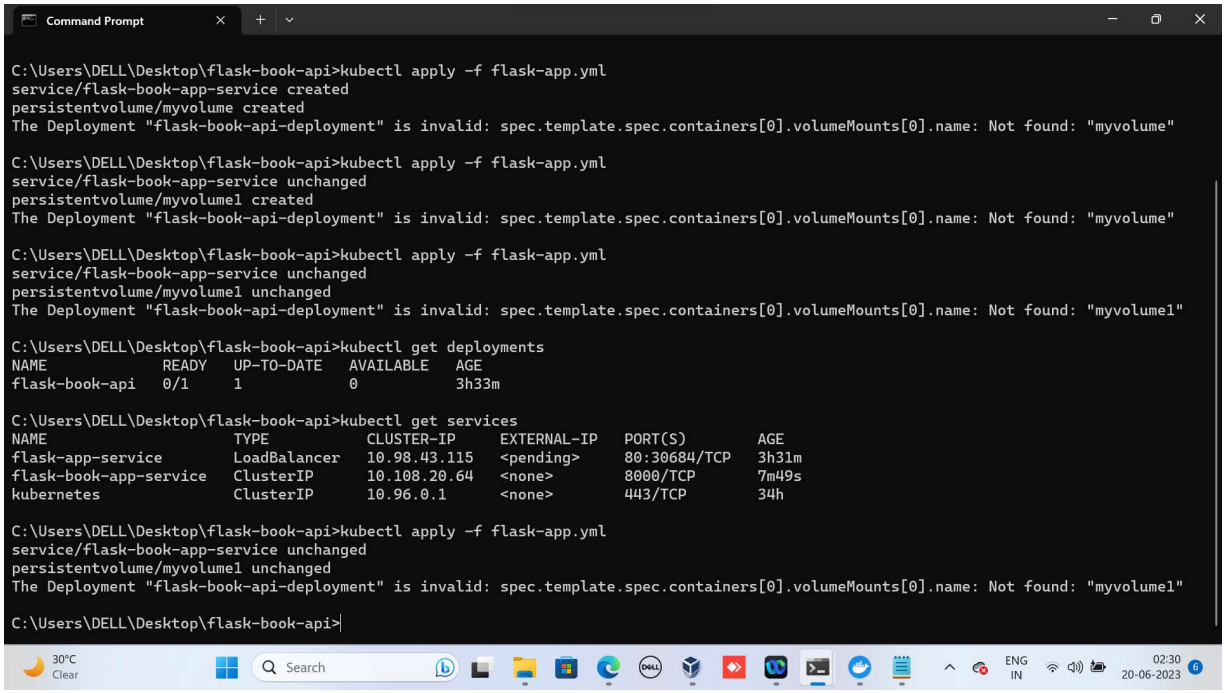
IUDX_DEVOPS_Task

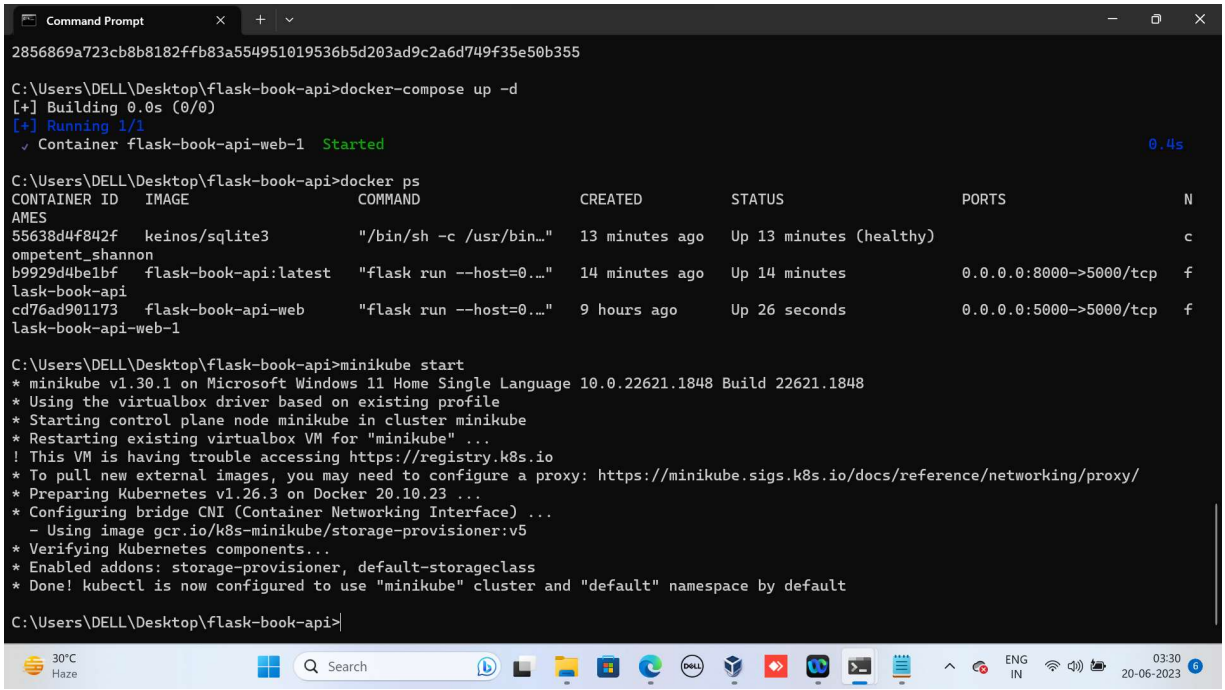
Screens after executing commands and yaml files:



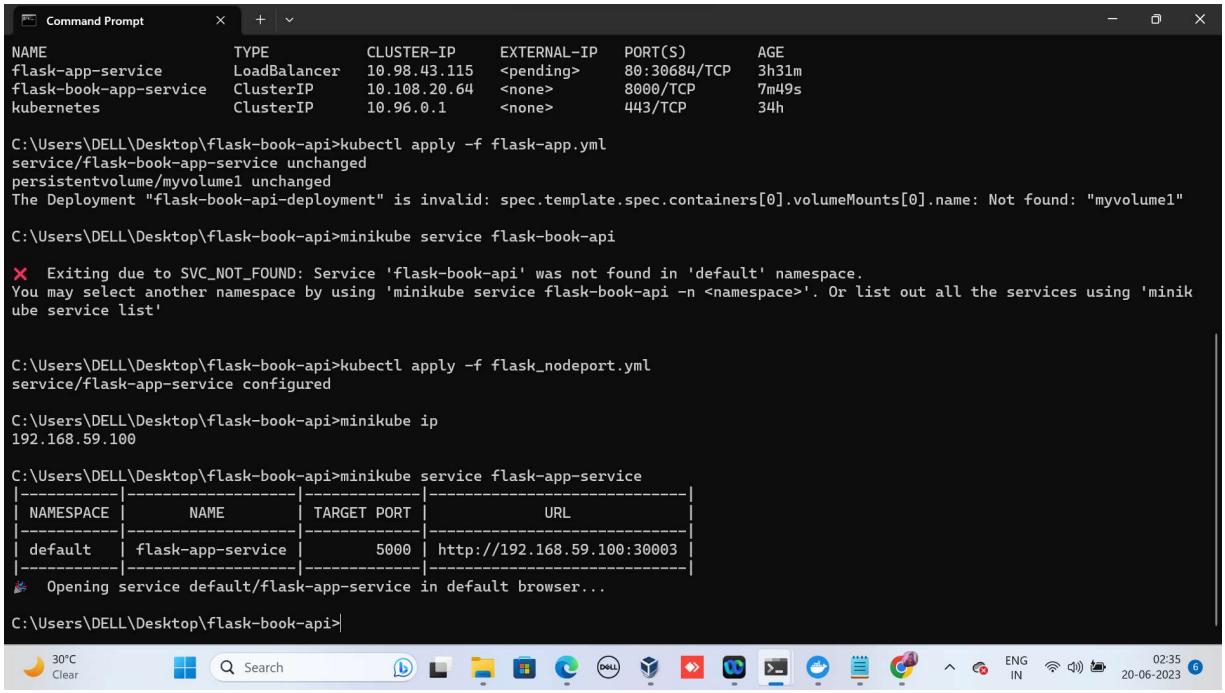


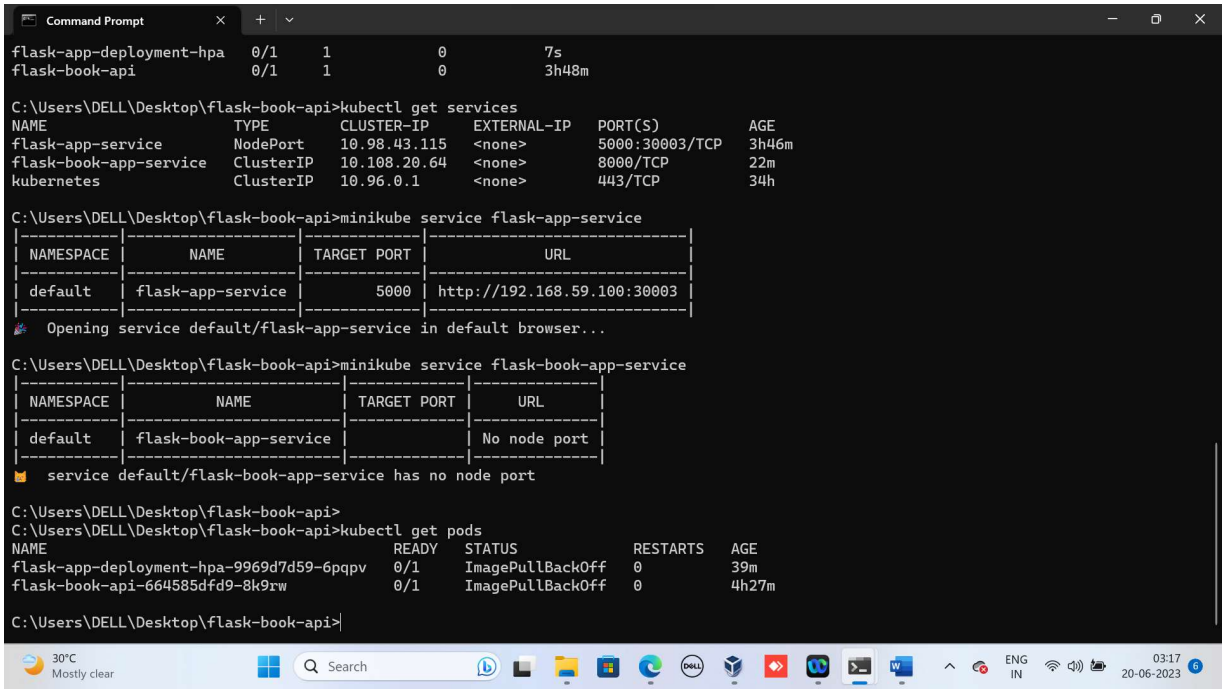
IUDX_DEVOPS_Task



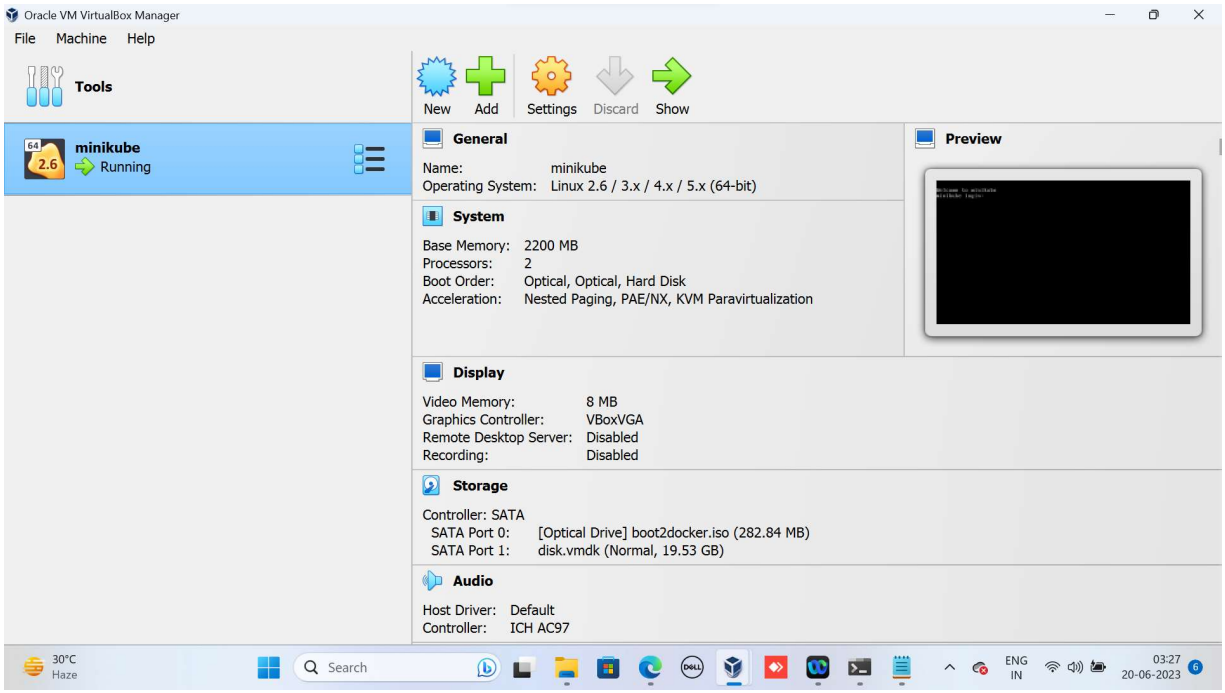


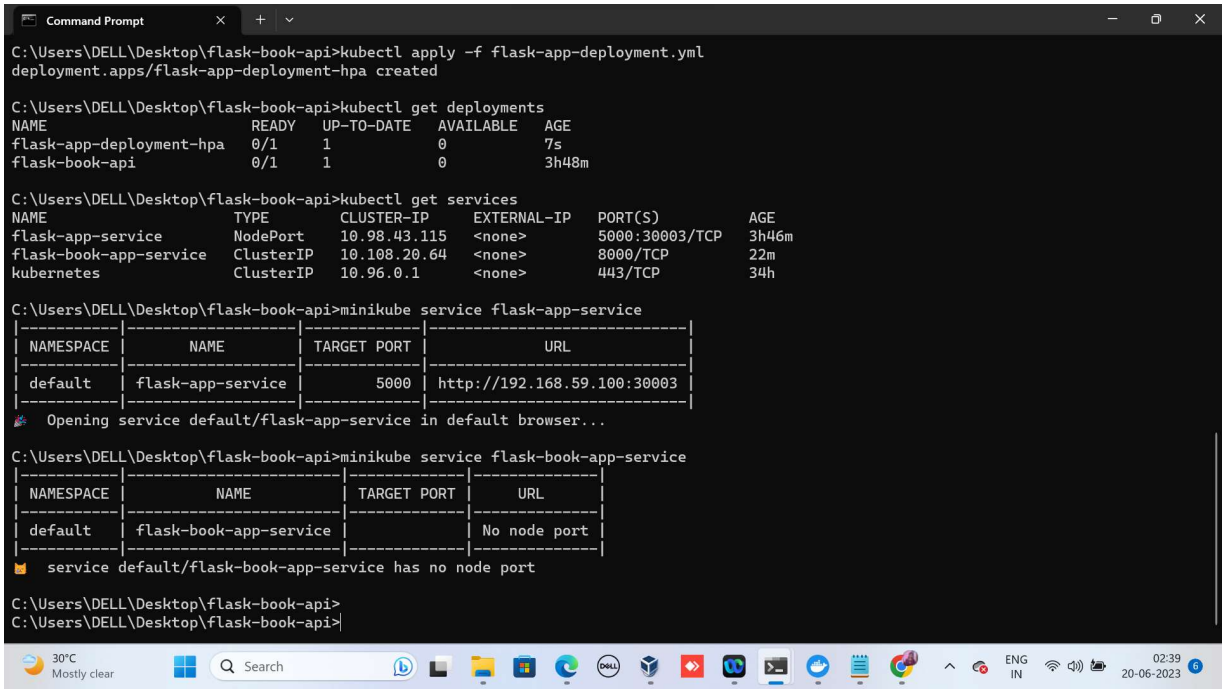
IUDX_DEVOPS_Task





IUDX_DEVOPS_Task





----------END----------