Restful API And Microservices with Python

Day 13

Day 13 - Overview

- Product Management Service - Completion and evaluation
- Dockerizing User Management Service
- Running User Management Service from docker compose

Prerequisite

- VM with windows OS
- Python 3.8 or >
- Visual Studio Code - Code Editor
- Postman
- GIT

https://github.com/saurav-samantray/flask-microservices-training/blob/main/slides/Setup%20GIT%20in%20your%20Local%20system.pdf

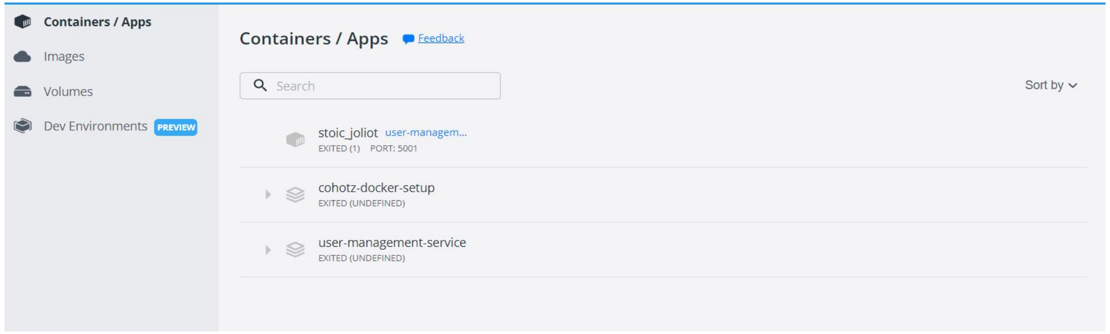
- Docker

Sync your fork for Day 13 activities

- Follow the below document to sync your fork and update local repository.

  https://github.com/saurav-samantray/flask-microservices-training/blob/main/slides/Setup%20GIT%20in%20your%20Local%20system.pdf

- Stop any already running container from Docker Desktop



stoic_joliot user-managem...
EXITED (1) PORT: 5001

cohotz-docker-setup
EXITED (UNDEFINED)

user-management-service
EXITED (UNDEFINED)

Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

A simplistic explanation of Docker and Containerization

https://medium.com/nerd-for-tech/application-containerization-using-docker-oversimplified-4a0b0891ad9

Creating a Dockerfile

```
FROM python:3.10-slim-buster

LABEL maintainer="saurav.samantray@gmail.com"
WORKDIR /user-management-service

COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "server.py"]
```

Note: Make sure host is mapped to 0.0.0.0 in app.run(host=”0.0.0.0”)

Update Config.py to add a Docker specific config

```
class DockerConfig(Config):
    SQLALCHEMY_DATABASE_URI =
'postgresql://postgres:postgres@pgdb:5432/user-management-db'
```

Building the Docker Image

docker build -t user-management-service .

Command Break Down

- **docker build** - to specify the docker action, which is building image
- **-t user-management-service** - specifying the name of the image/tag of image
- **.** - The path of Dockerfile. Dot indicated it’s present in current folder

Updating the docker compose

```
  user-management-service:
    image: user-management-service
    container_name: user-management-service
    ports:
      - "5000:5000"
    environment:
      FLASK_ENV: docker
    depends_on:
      - db
```

Running Docker container

Running Docker container using compose

- docker-compose run

Stopping all container and removing volumes

- docker-compose down -v

Docker cheat sheet

- ***docker ps*** - List all running docker containers
- ***docker images*** - List all images available in local machine
- ***docker stats*** - check CPI, Memory and network stats of running container
- ***docker image rm <image-name>*** - delete image
- ***docker system prune -v*** - delete all unused images, container and volumes from local machine
- ***docker-compose up -d*** - running docker compose in detached mode.
- ***docker-compose down*** - stopping docker compose container running in detached mode

Q and A