Containerization

installing docker:https://docs.docker.com/compose/install/#install-compose

a) Get Docker Toolbox (for older systems)

1.Docker Machine for running docker-machine commands

2.Docker Engine for running the docker commands

3.Docker Compose for running the docker-compose commands

4.Kitematic, the Docker GUI

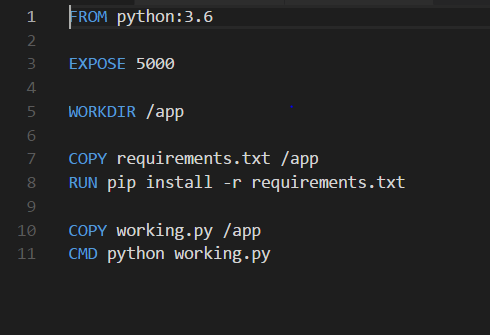
5.a shell preconfigured for a Docker command-line environment

6.Oracle VirtualBox

b) working.py - it has the flask application which connects the application with the database and has the rest api.

c) init.sql -has all the queries to initialize the database.

d) Creating a Docker image:



it contains instructions describing our image and allows it to build automatically, which can run in container.

creating a docker file in our application directory.

1.python (python run time image)

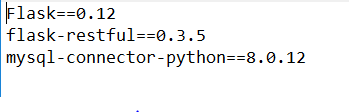
2.exposing the port in container

3.setting the working directory

4.installing the specified requirements

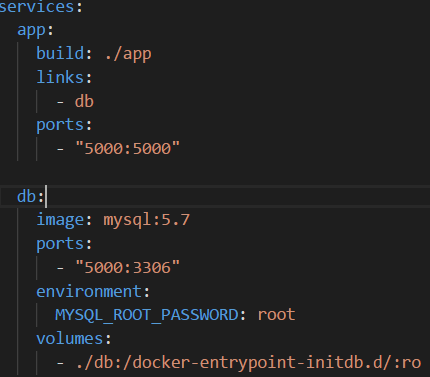
5.running the application

e) creating requirement file:



we will need sql for our docker image, so we will need MySQL connector.

f) Creating the Docker- Compose file:(yml)



The separate parts of your application (the different images) should run in separate

pods.

using 2 services, one for the api and one for the database, using docker-compose for 2 containers into one application.

one container has the application and the other has the database.

build: specifies the directory where the Dockerfile has instructions for building the service

links: links services to other containers.

port: used for mapping the port (host and Container)

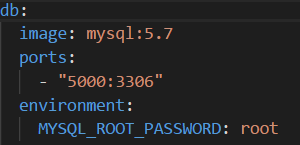


image: using existing image from the repository

volumes: initializing the database with our schema

Now connecting the application to the database:



running the application

: using the command docker-compose up

Have some issues in the configuration of port in the windows