Created by : Sonia Khatri

Docker commands

1. To run a docker-compose ,

Docker-compose up

1. To tag an image ,

docker tag 597307553dd7 sonia679/merakipoc:objdetectionimage

1. To push images to the repository in docker hub :

docker push sonia679/merakipoc

Here , merakipoc : repository name

Objectdetectionimage : tagname

597307553dd7 : local image id

1. For stopping all the running containers which came up by doing docker-compose up, you can issue the command :

docker-compose down

1. For removing the images which is not required :

Docker rmi <imageid>

For forceful removal, docker rmi -f <imageid>

1. In order to stop a docker container manually,

Docker stop <containerid>

1. To remove a container manually,

Docker rm <containerid>

1. To check all existing images

Docker images

EXPLANATION OF A DOCKERFILE

EXAMPLE DockerFile.

FROM python:3

RUN mkdir /predict

COPY . /predict/

ADD requirements.txt ./

RUN pip install -r requirements.txt

WORKDIR /predict/

RUN chmod 644 PredictionAnalysis.ipynb

ENTRYPOINT [ "runipy" ]

CMD [ "PredictionAnalysis.ipynb" ]

FROM python:3

>> Take the python image with version 3.x with latest tag ( default) from the docker hub and build the current image on top of that image.

RUN mkdir /predict

Make the directory named predict in the root folder of the image.

COPY . /predict/

Copy all the contents of the current cmd prompt directory from where the dockerfile is present , to the root folder /predict of the image which is being created.

ADD requirements.txt ./

>> add the file requirements.txt to the current folder in docker image so it is available for the next command

RUN pip install -r requirements.txt

>> run pip install for all the modules present in the requirements.txt to build the image file

WORKDIR /predict/

>> change directory to predict/

Execute next commands inside the predict directory

RUN chmod 644 PredictionAnalysis.ipynb

Give permissions to the file which is to be run

ENTRYPOINT [ "runipy" ]

CMD [ "PredictionAnalysis.ipynb" ]

Execute the command : runipy PredictionAnalysis.ipynb inside the docker container once it starts running.

EXPLANATION OF A DOCKER-COMPOSE FILE.

version: '3.1'

volumes:

database\_data:

driver: local

services:

db:

#build: .

image: sonia679/merakipoc:databaseimage

hostname: db

restart: always

environment:

POSTGRES\_DB: merakidb

POSTGRES\_USER: postgres

POSTGRES\_PASSWORD: postgres

ports:

- 5000:5432

volumes:

- database\_data:/var/lib/postgresql/data

networks:

- elk

web:

#build: ../

image: sonia679/merakipoc:middlewareimage

hostname: web

expose:

- 4004

ports:

- 4004:4004

volumes:

- ./api:/usr/src/app/

links:

- db

networks:

- elk

pyapp:

#build: ../../Weapon\_Detection/Tensorflow/models

image: sonia679/merakipoc:objdetectionimage

hostname: pyapp

ports:

- 5555:5555

links:

- web

#image: pyapp

networks:

- elk

predictionapp:

#build: ../prediction/

image: sonia679/merakipoc:predictionimage

hostname: predictionapp

ports:

- 4100:4100

links:

- db

#image: predictionapp

networks:

- elk

networks:

elk:

driver: bridge

version: '3.1'

>> Docker version

services:

db:

>> db is the name of the service

#build: .

When you want to build an image locally with the code present in local directory, use this option.

image: sonia679/merakipoc:databaseimage

Name of the image in docker hub.

Sonia679 – username

Merakipoc – repository name

Databaseimage – tagname

hostname: pyapp

This is used by the other containers to access the url of the linked container.

ports:

- 4100:4100

Left side : externally opened port ( on the machine)

Right side : docker container network internal port number

networks:

- elk

>> any network name can be provided and it is used among different services to link the services with each other in a common network

links:

- db

This is used to link one container to the other.