How to Dockerize a Project/API

Docker : Docker is a software platform. It enables software developers to develop, ship and run applications within its containers. Containers are lightweight software applications.

Docker File : A docker file is a text file that contains the set of instructions for the Docker platform.

Docker Image : An image includes everything needed to run an application — the code or binary, runtime, dependencies, and any other file system objects required.

Docker Container : Docker containers run the application code.

Example of a Minimal python Docker Container:

Step 1: Specifying the base image
FROM alpine:latest

RUN apk add cmd:pip3 \
&& apk add --no-cache python3-dev \
&& pip3 install --upgrade pip

Step 2: Setting a directory for the app
WORKDIR /app

Step 3 : Copy all the files to the container
COPY . /app

Step 4 : Install all the dependencies from requirements.txt file
RUN pip3 --no-cache-dir install -r requirements.txt

Step 5 : Defining the port number the container should expose
EXPOSE 5000

#ENTRYPOINT ["python3"]

Step 6 : Command to run the application
CMD [**"python3"**,**"app.py"**]
[** to get the requirements.txt use pip install pipreqs => pipreqs ./ . requirements.txt file will be created.]

Above codes will be in Dockerfile. So our Dockerfile is ready. We can now build the Docker image from Dockerfile.

Step 7 Building an Image :

docker build -t username/filename .
docker build -t auishik/mp3_specs . [use sudo to run in your local machine]

So our docker image is ready now.

Step 8: Run the docker Image in a container :

docker run -p 8888:5000 auishik/mp3_specs [use sudo to run in your local machine][8888 – external port and 5000 – internal port for the server inside the container]

Now Python Flask web server is now running within a docker container.