**This document consists of steps and codes involved in creating Docker image and connecting different images through network.**

# Have to import the python version used for your project from a Docker hub

**FROM python: 3.6-slim-stretch**

# Directory has to be created for the project in the Docker Container (named "project" here)

**RUN mkdir -p /project**

# Copying the all the project datas to the Created Directory inside docker Container

**COPY projectFiles /project/projectFiles**

# Listing the files in copied directory for verification purpose

**RUN ls -la /project/\***

# Required libraries has to be installed for the project by using requirements text(**pip freeze > rquirements.txt**) file collected from the virtual environment used for the project.

**RUN pip install -r /project/projectFiles/requirements.txt**

# To Avoid requirements running again while creating the same image for modification:

**WORKDIR /project/projectFiles**

**ADD ./requirements.txt /project/projectFiles/requirements.txt**

**RUN pip install -r /project/projectFiles/requirements.txt**

**ADD . /project/projectFiles ( ‘.’ --refers to current directory)**

# Setting the Python path to make code executable

**ENV PYTHONPATH "${PYTHONPATH}:/project/projectFiles"**

# Install Necessary requirements files into container using "RUN" Command, as show in examples below

* **RUN [ "python", "-c", "import nltk; nltk.download('stopwords')" ]**
* **RUN python -m nltk.downloader punkt**
* **RUN python -m nltk.downloader wordnet**

# Exposing the Port given in api files while running docker image

**EXPOSE 5000**

# Using the "CMD" to run the python script inside the container

**CMD python /project/projectFiles/test.py**

# Once docker file is created, use Docker commands to create Docker images and execute the created image(-t -- refers to tag)

**docker build -t "image name" .**

#To check the images created, see our newly created docker image here)

**docker images**

#To check the docker images currently running

**docker ps**

**# To check all the running images and history**

**docker ps –a**

# To run Docker using Port

* Giving only image name in iteration Mode

**docker run -it -p 5000:5000 “image name”**

* Giving only image name in detach Mode

**docker run -d -p 5000:5000 “image name”**

* Giving container and image name

**docker run -it -p 5000:5000 --name=“container name” “image name”**

# To save docker file

**docker save 'Image name' > 'Image name'.tar**

# Creating a network

**docker network create -d bridge "network name"**

#Adding a container to it you can add any number of containers

**docker network connect "network name" "container name"**

# To inspect the network for running containers inside network

**docker network inspect “network name"**

**Additional Commands used which will be useful while using linux while creating docker image and connecting to network:**

# To start Docker service in linux

**systemctl start docker.service**

# to Remove the docker images with none tags (While creating repeated images on same name none tag image will be added)

**docker images | grep none | awk '{ print $3; }' | xargs docker rmi–f**

# xgboost will cause(You are running 32-bit Python on a 64-bit OS) error, so we have to add the below installation to avoid the same

**RUN apt-get update && \**

**apt-get -y --no-install-recommends install \**

**libgomp1**

# To remove Contanier (First stop and then remove)

**docker stop “Conatiner name”**

**docker container rm "Container Name"**