Name: Anand Kumar Gupta Email id: ananddkg99.9@gmail.com

Assignment 14

Ques 1 : Explain the below DockerFile

```
FROM nvidia/cuda:11.4.2-cudnn8-runtime-ubuntu20.04


#set up environment
RUN apt-get update && apt-get install --no-install-recommends --no-install-suggests -y curl
RUN apt-get install unzip
RUN apt-get -y install python3
RUN apt-get -y install python3-pip

# Copy our application code
WORKDIR /var/app

# . Here means current directory.
COPY . .


RUN pip3 install --no-cache-dir -r requirements.txt
RUN python3 download_HF_Question_Generation_summarization.py

ENV LC_ALL=C.UTF-8
ENV LANG=C.UTF-8


EXPOSE 80

# Start the app
CMD ["gunicorn", "-b", "0.0.0.0:80","app:app","--workers","1","-k","uvicorn.workers.UvicornWorker"]
```

Soln:

1) FROM nvidia/cuda:11.4.2-cudnn8-runtime-ubuntu20.04
Specifies the base image to use, in this case it's a CUDA image for Ubuntu 20.04.

2) RUN apt-get update && apt-get install –-no-install-recommends –-no-install-suggests –y curl
Runs the apt-get update command to update the package index, then installs curl using apt-get install. The options --no-install-recommends and --no-install-suggests prevent the installation of recommended and suggested packages. The -y option automatically answers yes to prompts.

3) RUN apt-get install unzip
Installs the unzip package.

4) RUN apt-get –y install python3
Installs the python3 package.

5) RUN apt-get –y install python3-pip
Installs the python3-pip package, which is the package installer for Python 3.

6) WORKDIR /var/app
Sets the working directory to /var/app.

7) COPY . .
Copies the current directory (where the Dockerfile is located) to the /var/app directory in the image.

8) RUN pip3 install –-no-cache-dir –r requirements.txt
Installs the dependencies listed in the requirements.txt file using pip3. The --no-cache-dir option prevents the caching of packages.

9) RUN python3 download_HF_Question_Generation_summarization.py
Runs the download_HF_Question_Generation_summarization.py script using python3.

10) ENV LC_ALL=C.UTF-8 ENV LANG=C.UTF-8
Sets the environment variables LC_ALL and LANG to “C.UTF-8”. “C.UTF-8” is a locale setting that specifies the character encoding and locale used by the system. The C part specifies the character encoding (which is typically ASCII-based), while the .UTF-8 part specifies the locale that uses UTF-8 as the character encoding.

11) EXPOSE 80
Exposes port 80 to allow external connections.

12) CMD["gunicorn","-b","0.0.0.0:80","app:app","--workers","1","-k","uvicorn.workers.UvicornWorker"]

*****gunicorn - This is the command to start the Gunicorn server.
*-b 0.0.0.0:80 - This option specifies the address and port to bind the server to. The 0.0.0.0 address is a special address that means all available network interfaces, so this option is binding the server to all available network interfaces on port 80.
*app:app - This option specifies the module and application callable for Gunicorn to use.
*--workers 1 - This option specifies the number of worker processes to run. The value 1 is setting the number of workers to one.
*-k uvicorn.workers.UvicornWorker - This option specifies the type of worker to run. The uvicorn.workers.UvicornWorker value is setting the worker to be a Uvicorn worker.