Questions:

1. **Create a python script that will poll a directory and all .txt files from the directory**
2. **This python script will count the number of words in each text file and create an output like file name - number of words**
3. **Create a docker image that will read a directory from the host machine. The host machine directory has to be shared**
4. **Run the docker image**
5. **Show that your script works by putting some text files in the host machine directory and your python script from docker image automatically loads this and writes the output**

Description of this Project:

Initially, I have created a python file where I write my python script which helps to load this autometically and give the output over the terminal on each update of any files present that directory named “Innovaccer_Document”. I have shared my script with it in below.

After that we have created a Dockerfile which is helping to create a docker image based on my python script, which is also shared in below.

After that we have to build an docker image by using - **sudo docker build -t <image_name> <folder_name>** [For current folder we can use ‘.’ over there]. Here we used,

sudo docker build -t new_image .

After that we have to run that newly created docker image named as “new_image” by using – sudo docker run -it -v <host_path>:<volume_path> <image_name> [Though in docker file, Entrypoint and CMD are provided over here, so we doesn’t need to give the execuable name (python3) and the file name with this command]. Here we used -

sudo docker run -it -v /home/souvik/Desktop/HeroVired/Assignment/Innovaccer_Document/:/usr/src/app/myVolume new_image

Where,

-v <host_path>:<volume_path> helps to create a shared volume into that container which will point a particular directory of the host.

After runing this above command, the python script, “Assignment.py” will start working and it will work on the shared folder named as “myVolume”. And initially, It will give a word count of each file and whenever any files of that directory is being updated, it will print that update count on that terminal. It will load autometically each and every time.

Screenshot of the whole process:

1. Create a **“Assignment.py”** file on VS-Code.

Assignment.py > ...

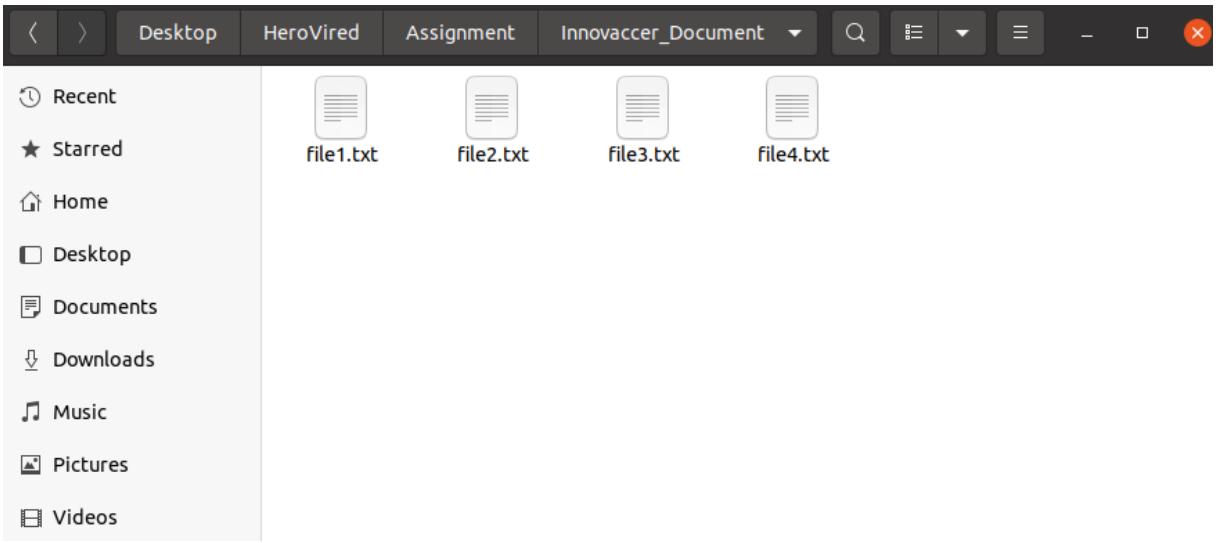
```python
import re, os, glob, time

cwd = os.getcwd()
file_modification_time = dict()
flag = 0

def word_count(files):
    with open(files,"r") as f:
        text_file = f.read()
        counter = len(re.findall(r'\w+', text_file))
        if counter > 1:
            print(files.split("/")[-1],"-",counter,"words")
        else:
            print(files.split("/")[-1],"-",counter,"word")


try:
    while (True):
        file_to_open = glob.glob(os.path.join(cwd, "myVolume/*.txt"), recursive=False)
        for files in file_to_open:
            if files not in file_modification_time:
                file_modification_time[files] = os.path.getmtime(files)
                if (flag):
                    print("-------------------------------------------------")
                    print("New File,", files.split("/")[-1][:-4] ,"is added.",end="\n")
                word_count(files)
            elif file_modification_time[files] != os.path.getmtime(files):
                file_modification_time[files] = os.path.getmtime(files)
                print("-------------------------------------------------")
                print(files.split("/")[-1][:-4],"is Updated.",end="\n")
                word_count(files)
        flag = 1
        time.sleep(5)
except KeyboardInterrupt as KI:
    print("-------------------------------------------\n",end="\n")
    print(">> Program is being closed due to Keyboard Interrupt\n")

```

2. Create a **“Dockerfile”** on the same directory.

Dockerfile > ...

```
#It is the Base-Image location of our docker-hub
FROM python:3

#From Whare our application will start working
WORKDIR /usr/src/app

#Copy src dest => Copy all files from current working to the docker working directory
COPY "Assignment.py" .

#CMD is used to provide the default file name
#If we run docker image without filename then it will work/
CMD ["Assignment.py"]

#It is used as a executable when docker image will run
ENTRYPOINT [ "python3" ]

```

3. Then create a directory into my host system named as **“Innovaccer_Document”** and create few text file over there as – [file1.txt, file2.txt, file3.txt, file4.txt]



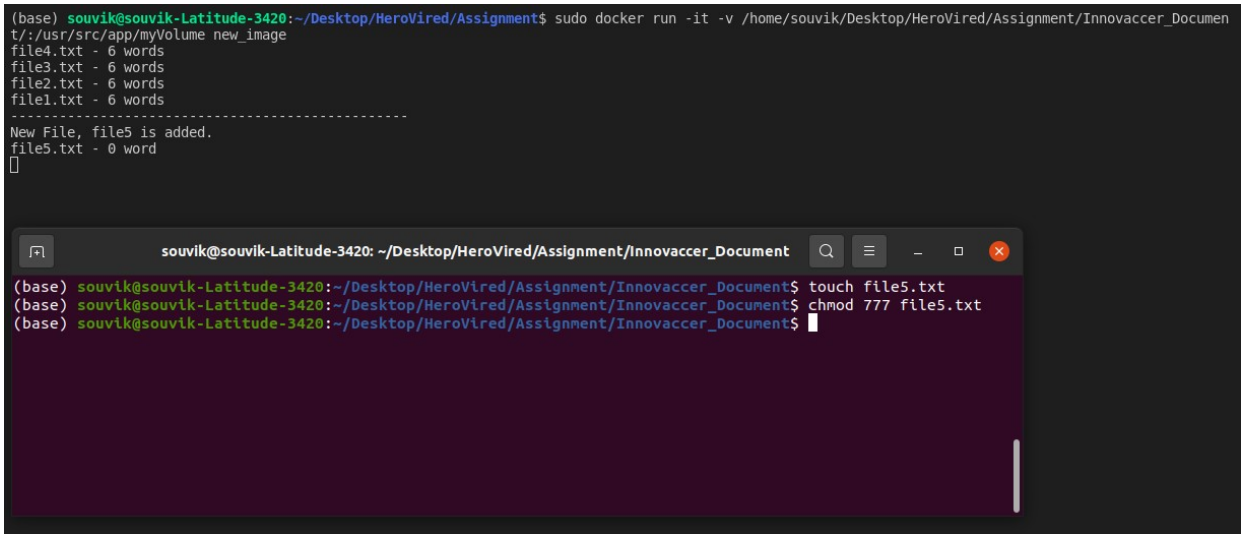
4. Then we use the command - docker build into my terminal to create the image.

```
(base) souvik@souvik-Latitude-3420:~/Desktop/HeroVired/Assignment$ sudo docker build -t new_image .
[sudo] password for souvik:
Sending build context to Docker daemon  9.216kB
Step 1/5 : FROM python:3
 ---> c05c608cfa20
Step 2/5 : WORKDIR /usr/src/app
 ---> Using cache
 ---> 9a72568cd02c
Step 3/5 : COPY "Assignment.py" .
 ---> 36eb29ef293e
Step 4/5 : CMD ["Assignment.py"]
 ---> Running in 3d828d8e6a35
Removing intermediate container 3d828d8e6a35
 ---> eac76259b780
Step 5/5 : ENTRYPOINT [ "python3" ]
 ---> Running in 0fd6c013a7b1
Removing intermediate container 0fd6c013a7b1
 ---> f9547cedb8f9
Successfully built f9547cedb8f9
Successfully tagged new_image:latest
```

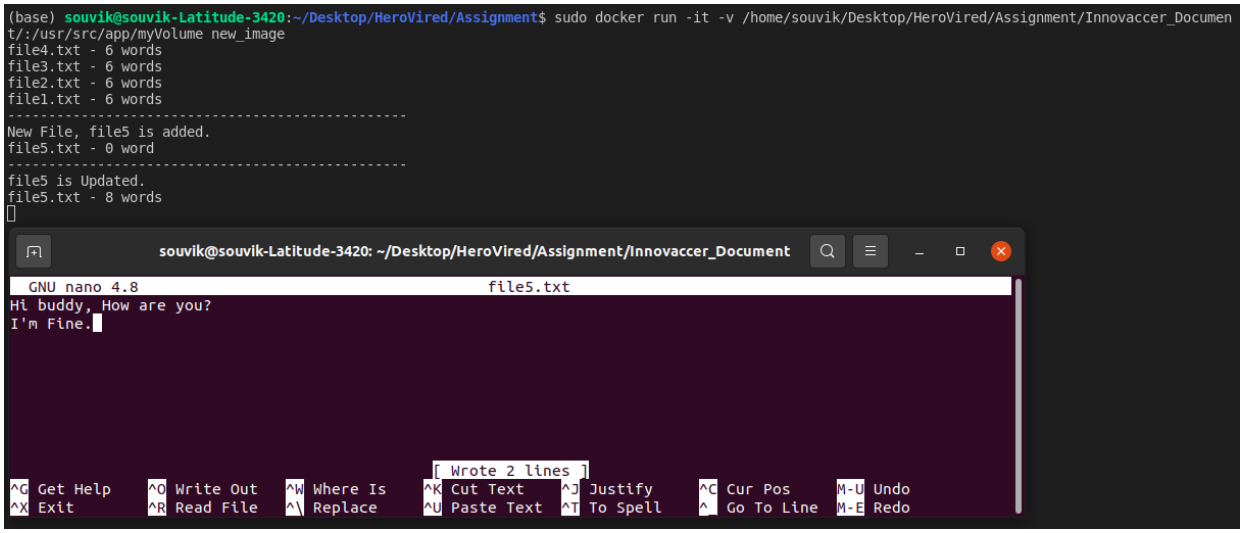
5. We use the command to run the docker image as runing that python file “Assignment.py” which is pointing to the shared volume. And it will provide the output of the count of word of each file.

```
(base) souvik@souvik-Latitude-3420:~/Desktop/HeroVired/Assignment$ sudo docker run -it -v /home/souvik/Desktop/HeroVired/Assignment/Innovaccer_Documen
t/:/usr/src/app/myVolume new_image
file4.txt - 6 words
file3.txt - 6 words
file2.txt - 6 words
file1.txt - 6 words
```

6. Now we will create a file into the shared directory from host using touch command in an another terminal as “file5.txt” and make its permission to 777. And whenever new file is created, it will show the count of that new file after creation of this file into that shared directory.



7. Now using “nano” text-editor, we will edit that file to provide few words into it to check the update. And we can see that file is update over there.



8. Finally to stop that program we will press “Ctrl+C” to give keyboardInterrupt and docker container will also stop after stoping that python script.