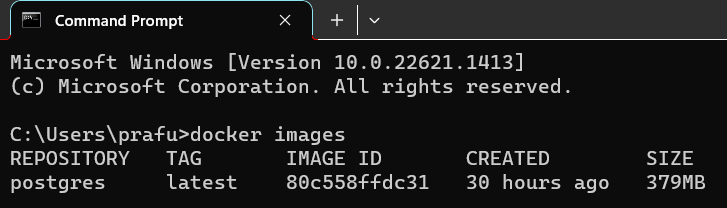
**Docker Desktop App & Docker Site profile page must be logged in before start**

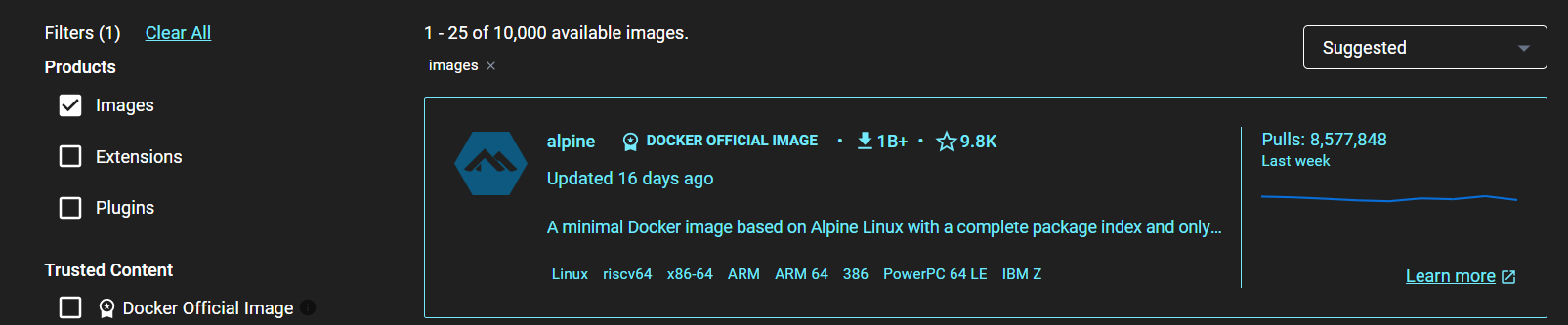


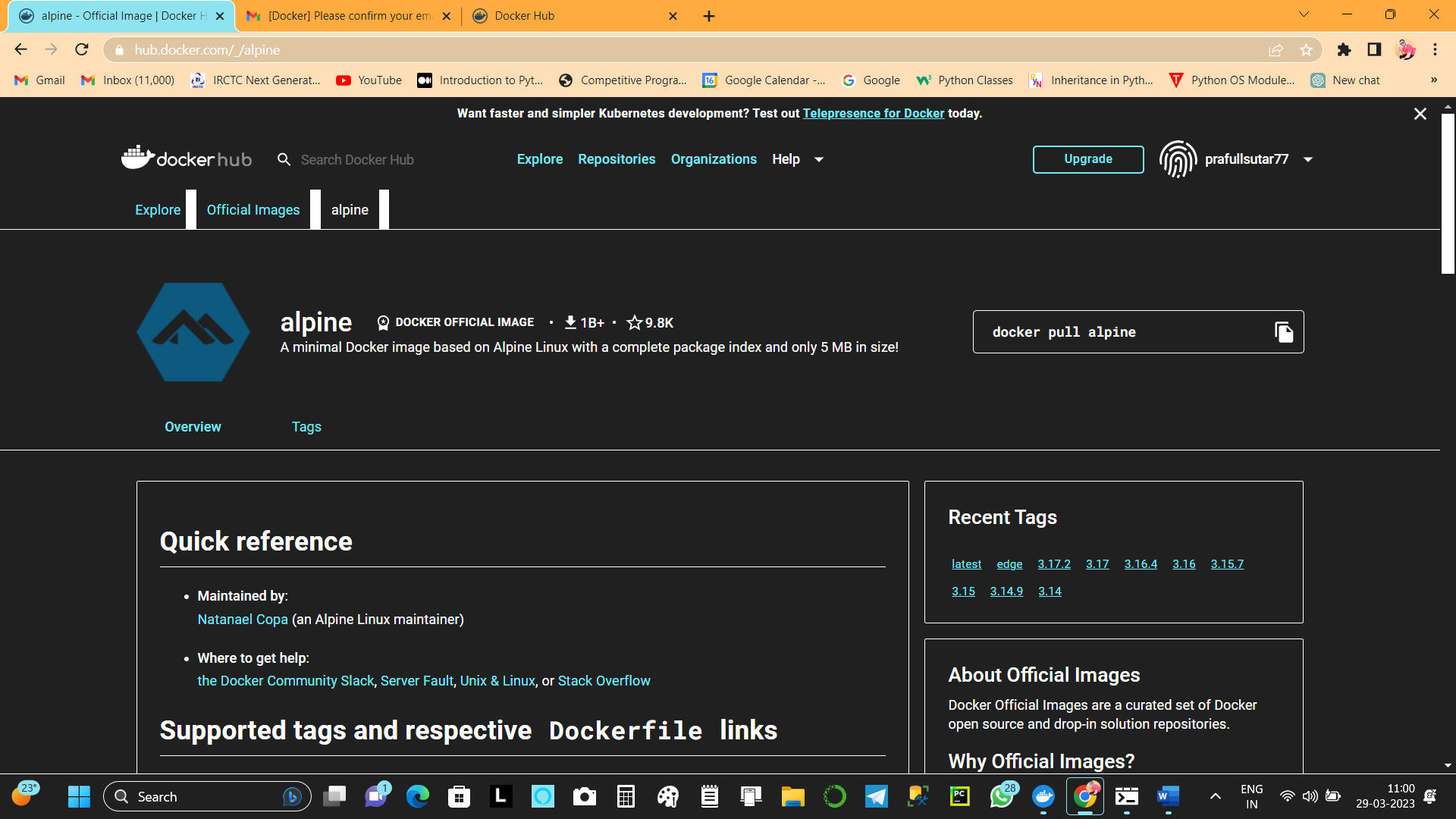
1.docker images: - To see available docker images

2.docker image rm postgres: - to delete postgres image

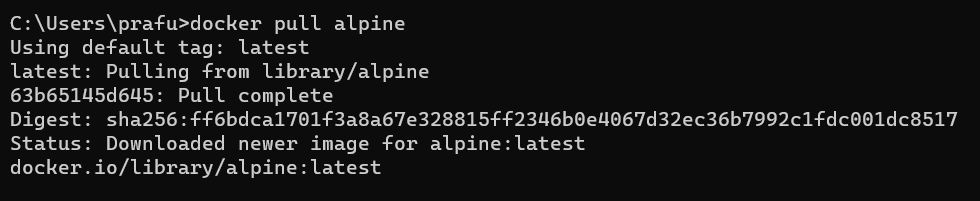
3.docker image rm postgres -f: - to delete postgres image forcefully

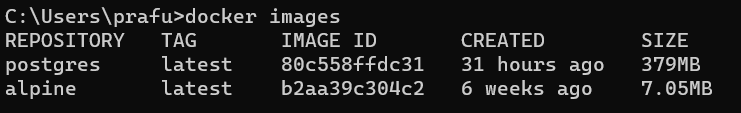
\*\* To pull any image to local repository: -



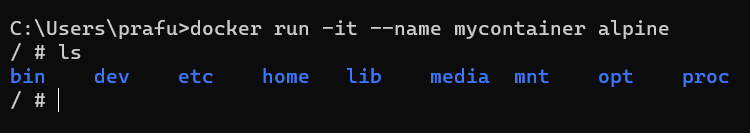


4. docker pull alpine: - To pull image alpine to local repo





5.docker run: - is used to create new container of latest image by using



docker run -it --name mycontainer alpine

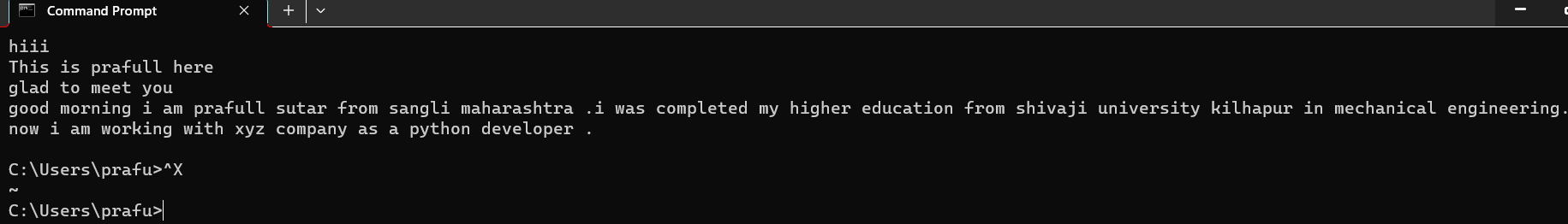
command rename new name from which image

***NOW WE ARE INSIDE ALPINE***

a.ls to see available file

b.touch test.txt : - to create text file

c.vi test.txt : - to write inside of that file

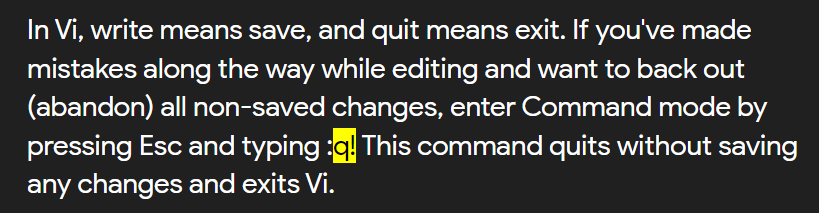


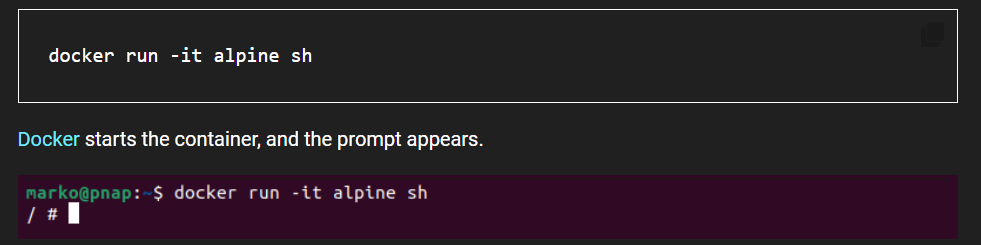
Important commands:-

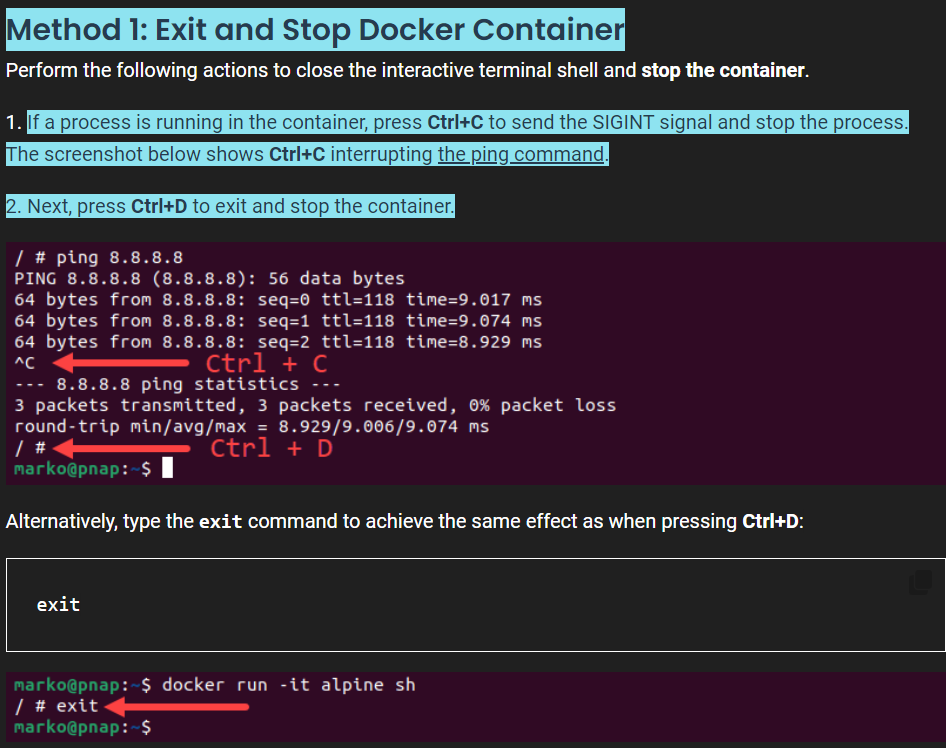
1.Ctrl x – direct exit from container without make it down (running in background)

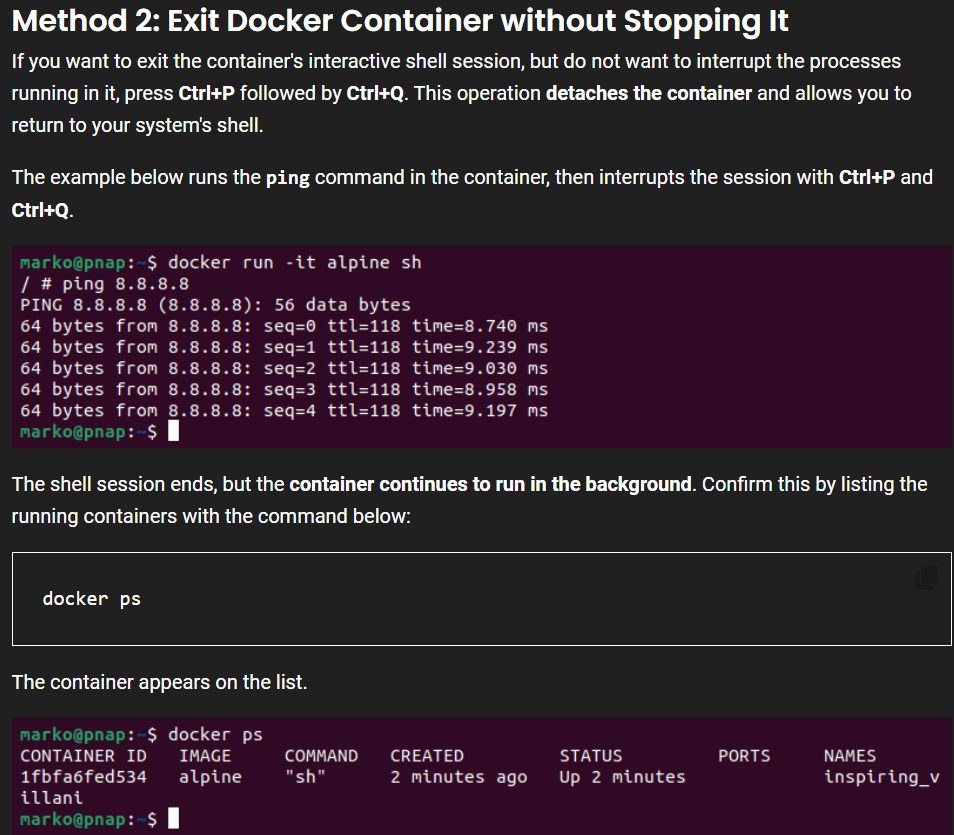
2.Ctrl c + ctrl d – to exit from container without running it in background

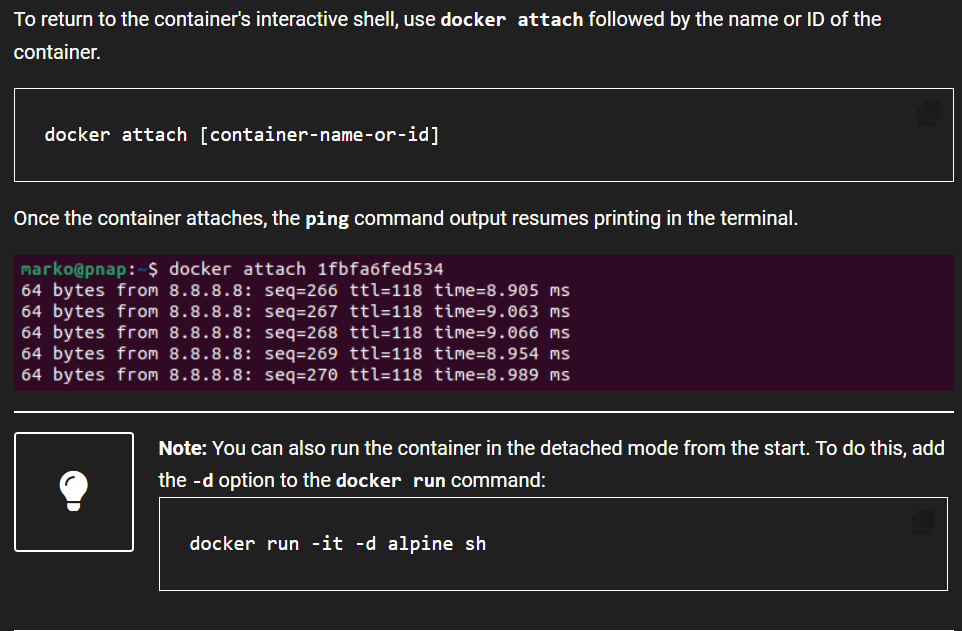
3.vi test.txt – go inside test.txt file + escape(Esc) +:q! to exit from from container without running it in background

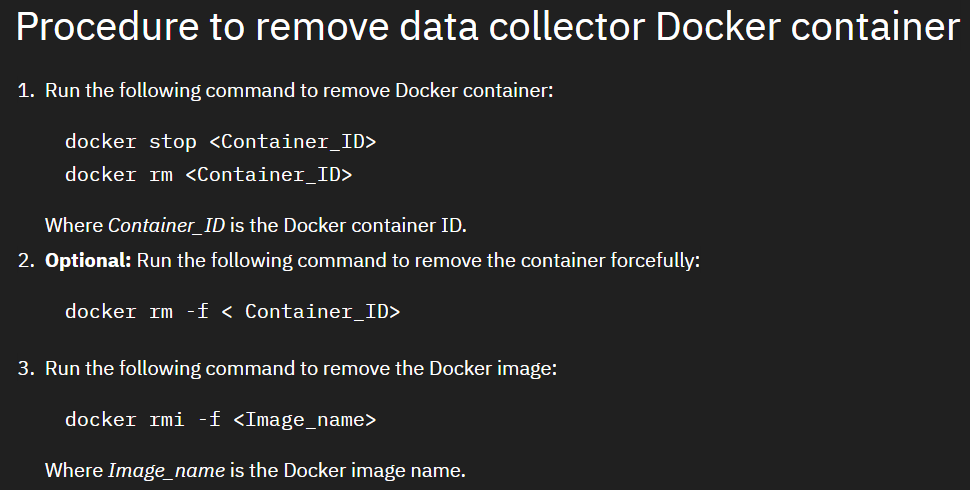


4. 



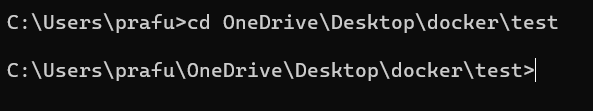




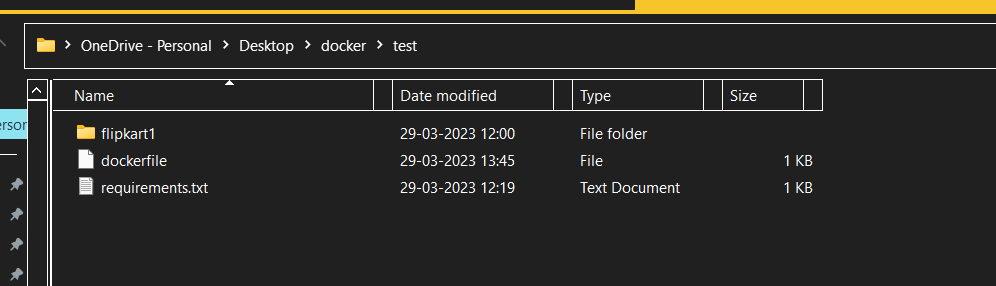


**CODE TO IMAGE:-**

**1.1ST GO INSIDE THE DIRECTORY (C:\Users\prafu\OneDrive\Desktop\docker\test) by using cd command**



2.Create folders

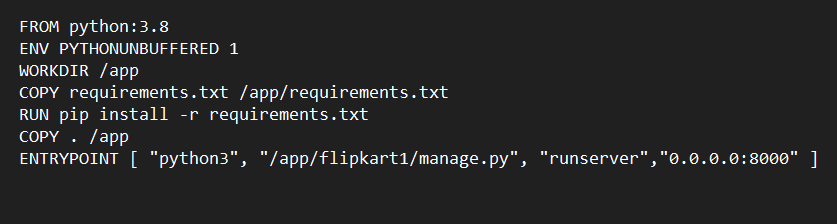


4.**dockerfile**-Must be without extension

Add all the run command inside this

From below folder and also add local local repo address

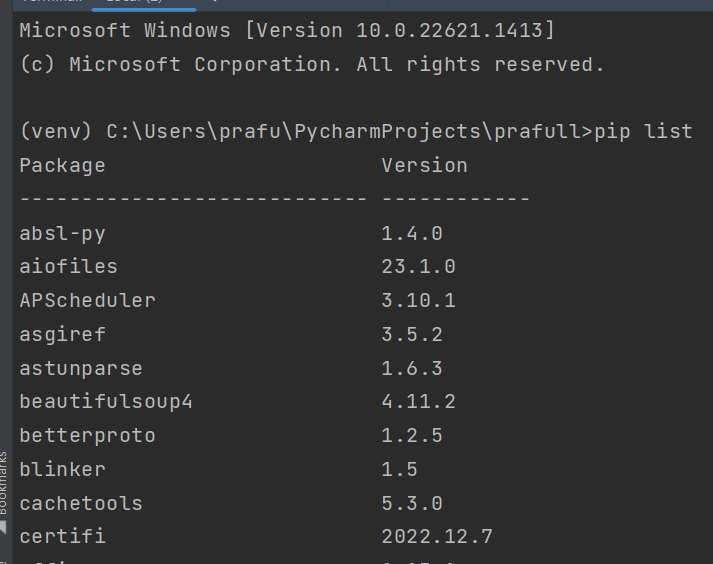
Site-https://dev.to/foadlind/dockerizing-a-django-mysql-project-g4m



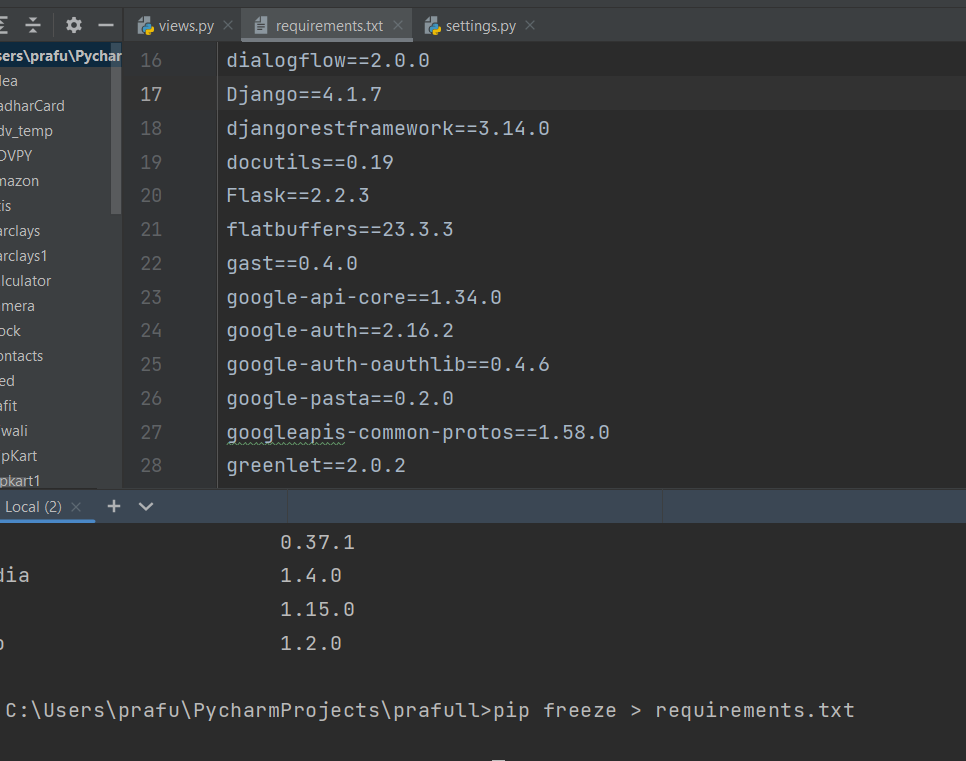
5.Inside the ***Requirements.txt***

At Pycharm CMD prompt

Run below command

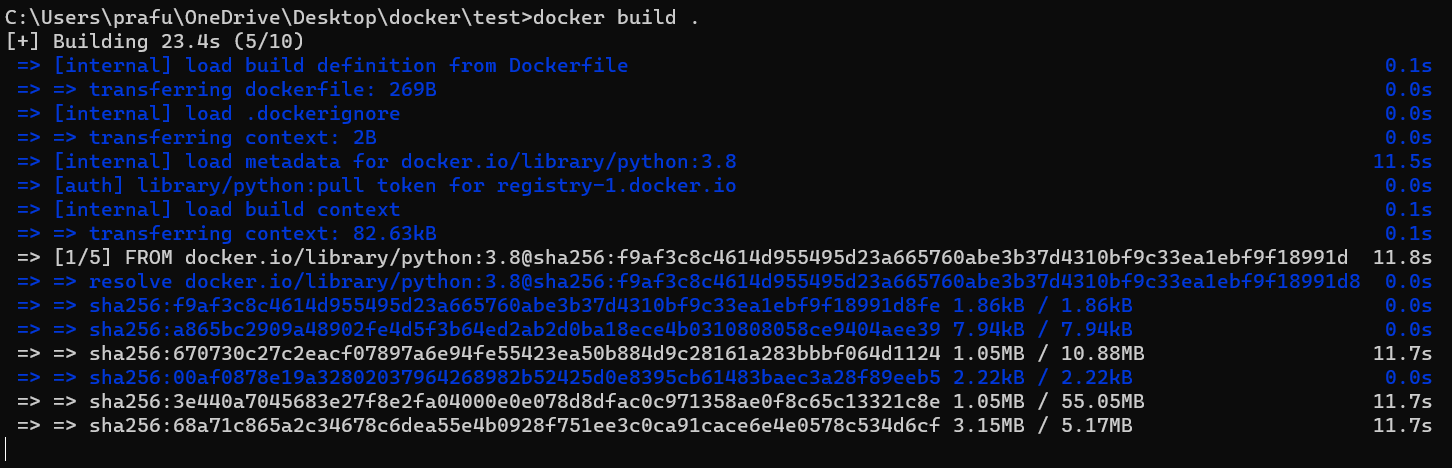


Then to create requirements.txt file & it will automatically create a file of text

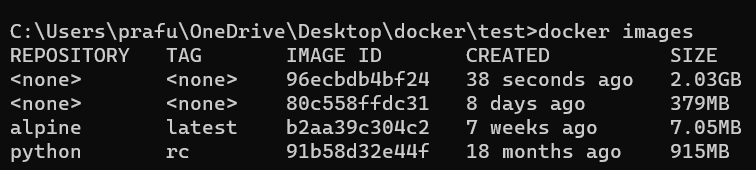


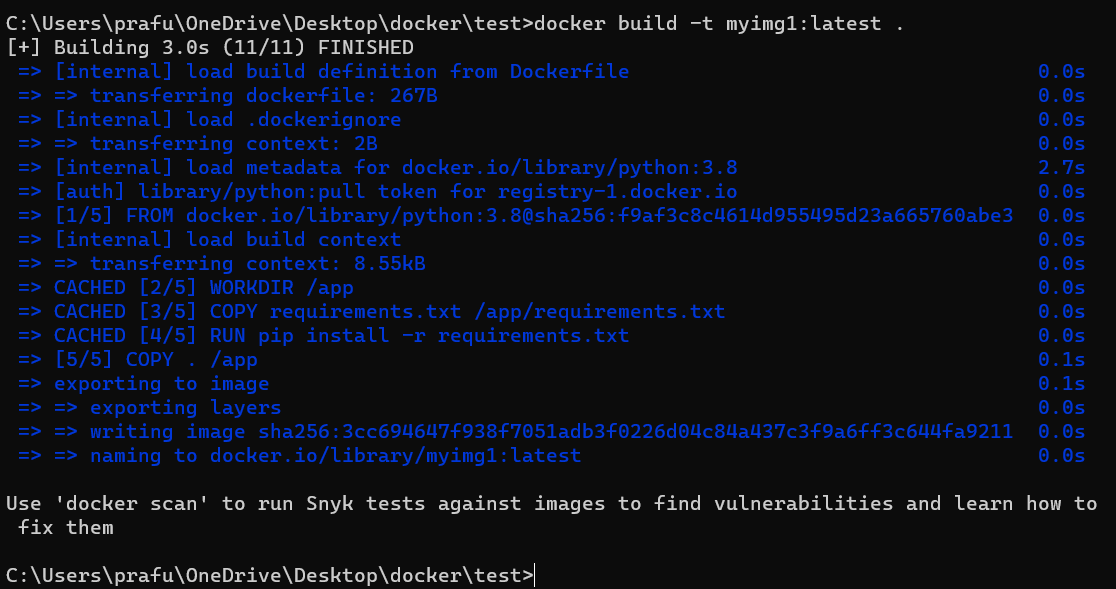
Copy this requirements.txt file at our located file in desktop

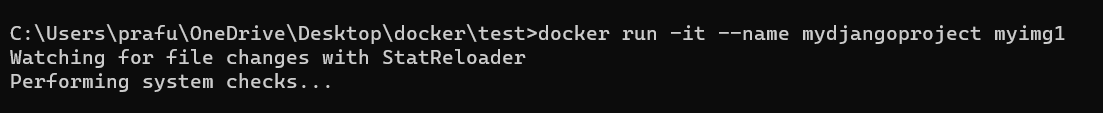
5. docker build .(space must after build ->space + full stop)

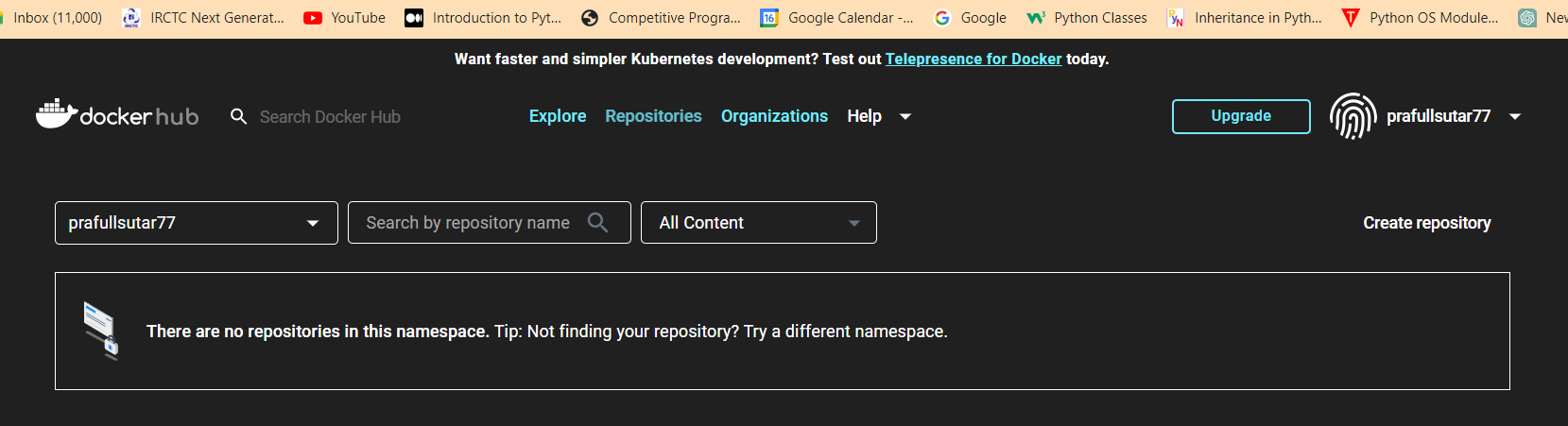


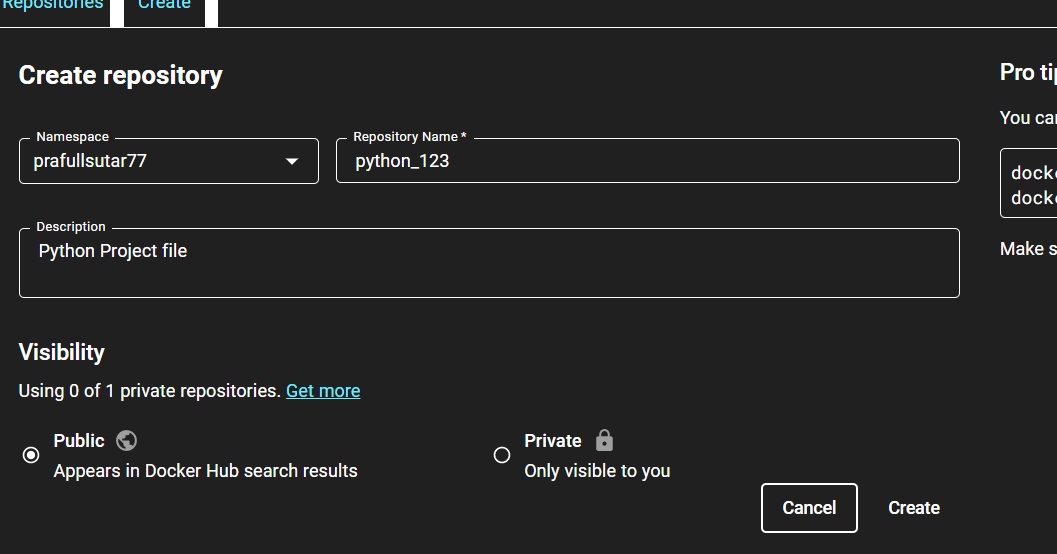
6. docker images



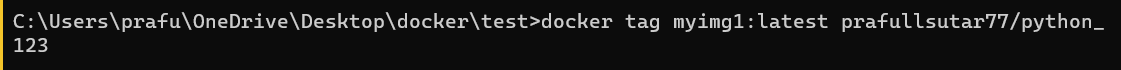
7. **docker build -t myimg1:latest .(space + full stop)**

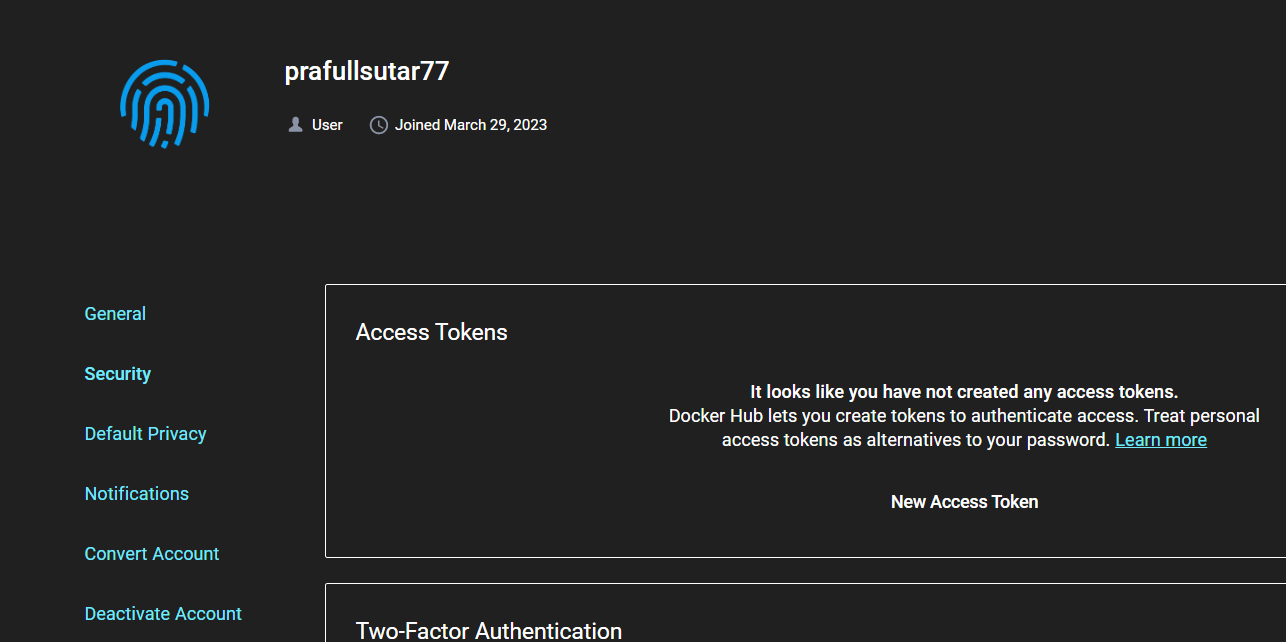
8. docker run -it - -name mydjangoproject myimg1

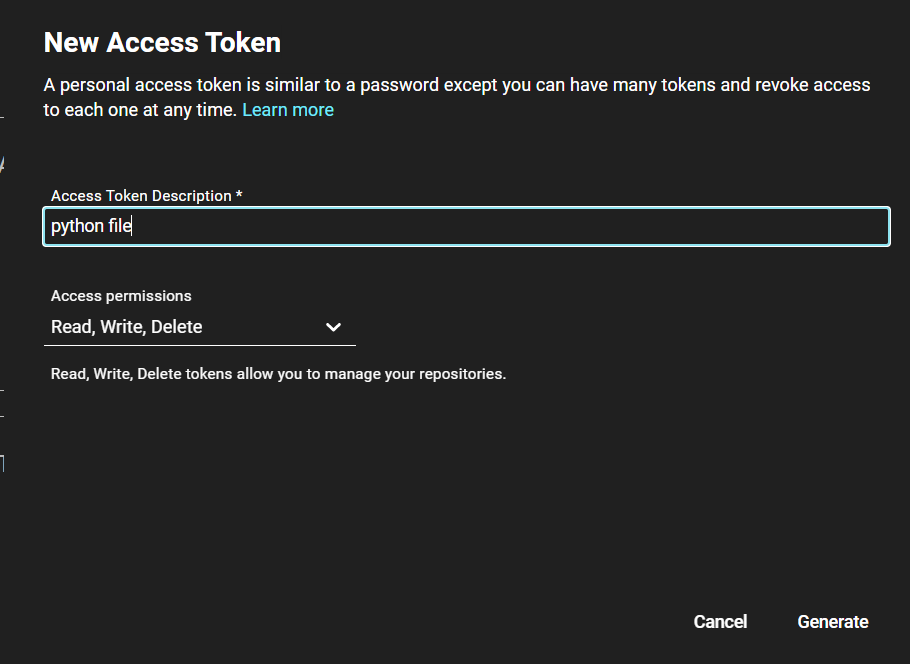
9. 

10. 

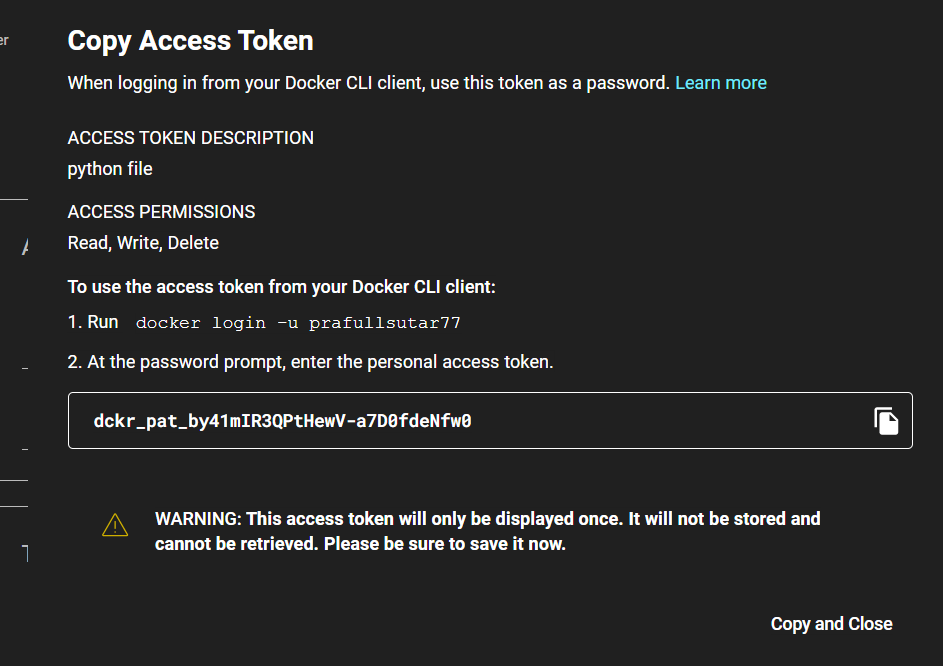
11.to create tag



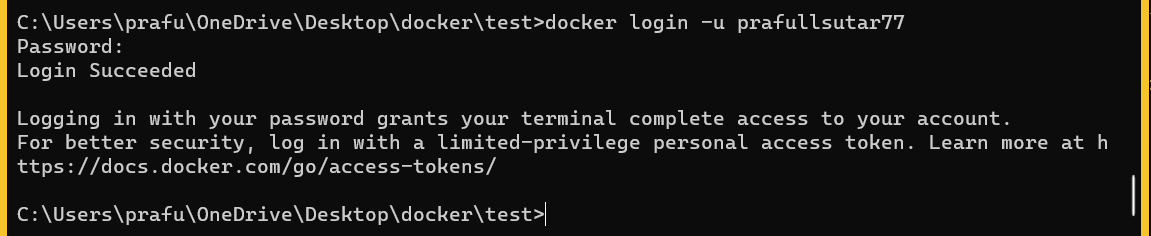
12. 

13. 

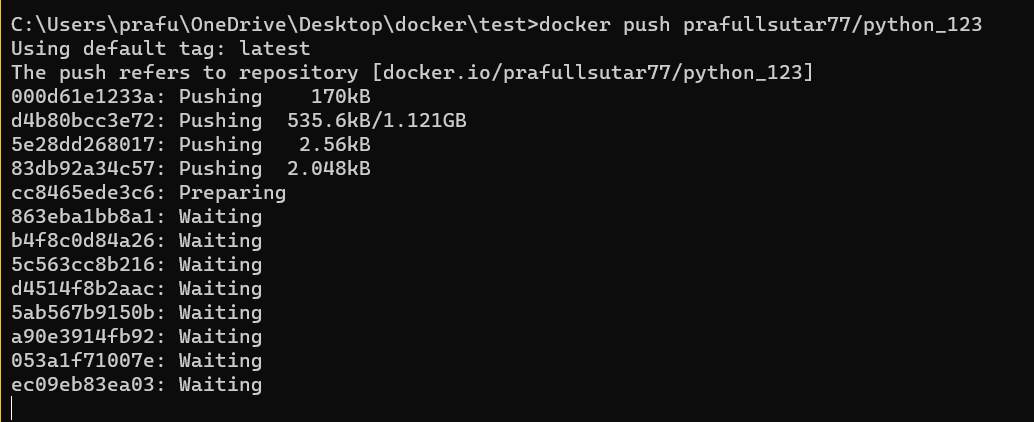
14.

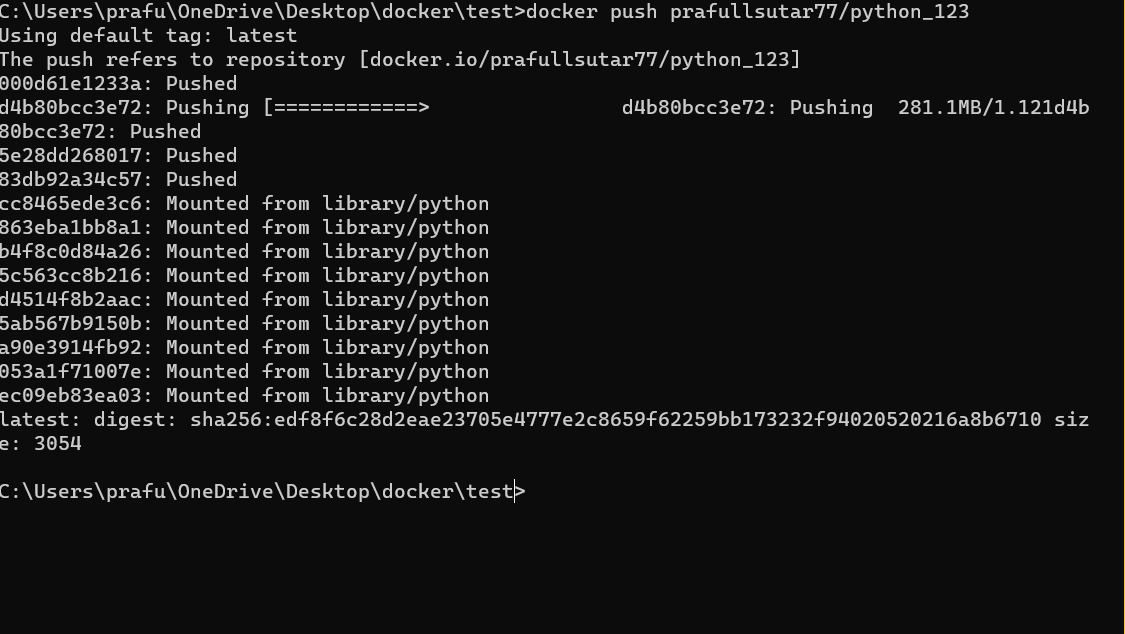


15.



16





Completed.

