ENSF 607 - Fall 2023
Assignment 5

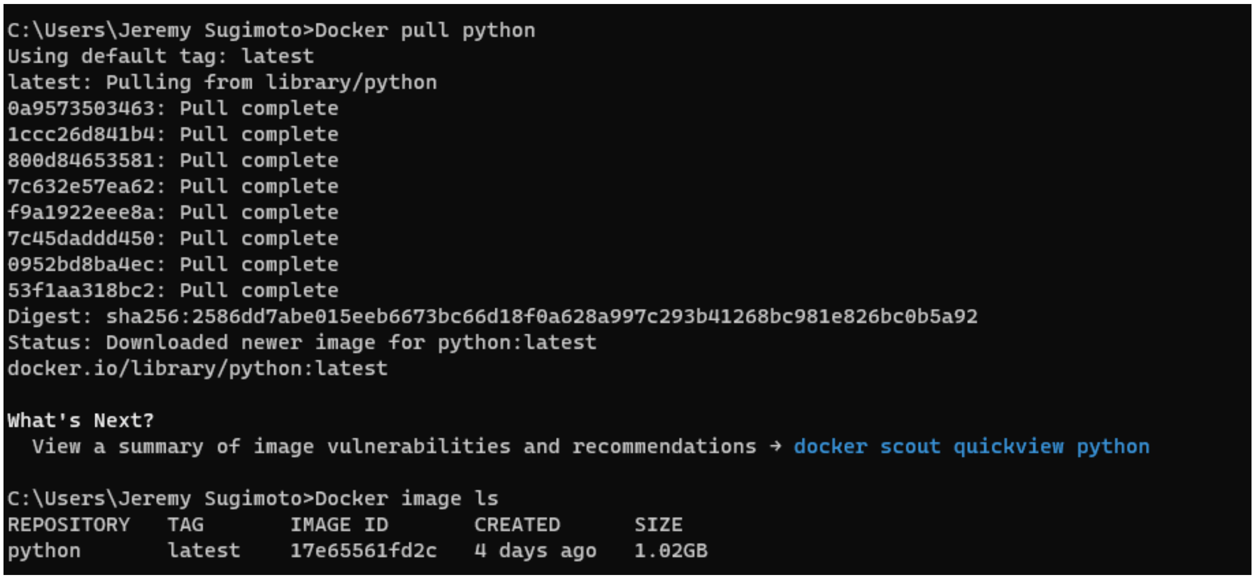
Jeremy Sugimoto (30232526)

November 3rd, 2023

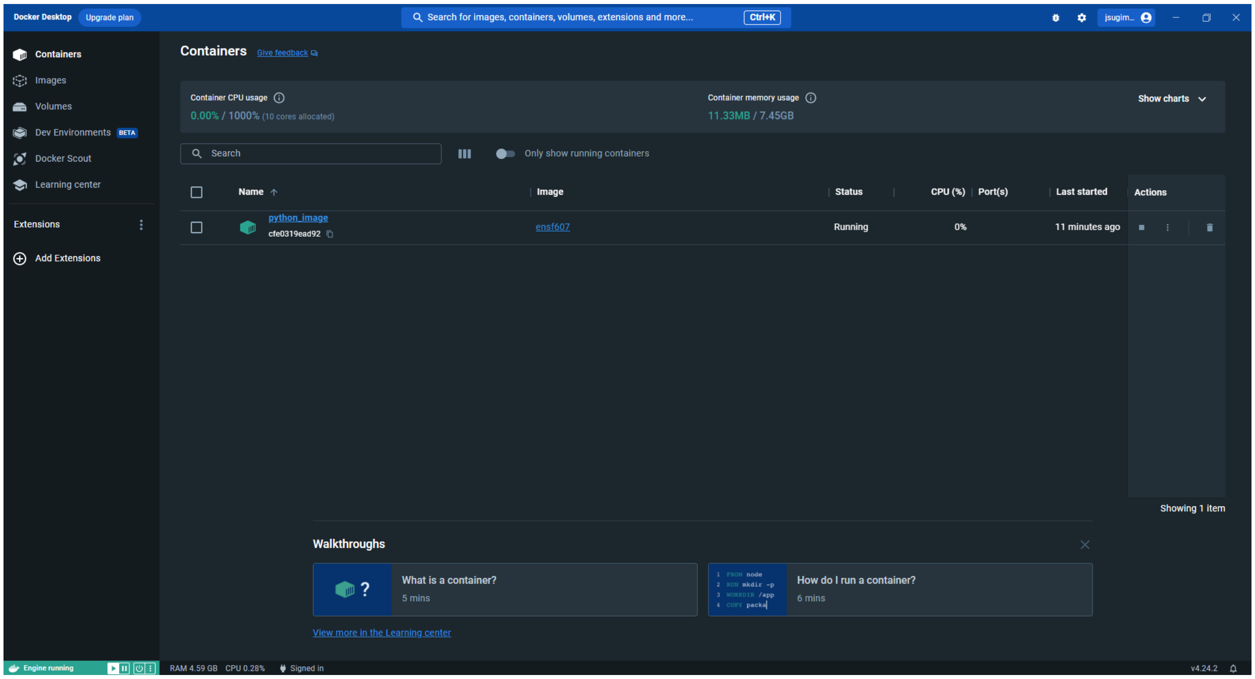
Steps 1-3 completed.

4. Now go to the cmd line on your computer and run the following command.

5. Once done you can run the following command:



```
C:\Users\Jeremy Sugimoto>Docker pull python
Using default tag: latest
latest: Pulling from library/python
0a9573503463: Pull complete
1ccc26d841b4: Pull complete
800d84653581: Pull complete
7c632e57ea62: Pull complete
f9a1922eee8a: Pull complete
7c45daddd450: Pull complete
0952bd8ba4ec: Pull complete
53f1aa318bc2: Pull complete
Digest: sha256:2586dd7abe015eeb6673bc66d18f0a628a997c293b41268bc981e826bc0b5a92
Status: Downloaded newer image for python:latest
docker.io/library/python:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview python

C:\Users\Jeremy Sugimoto>Docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
python       latest    17e65561fd2c   4 days ago   1.02GB
```



Containers

Container CPU usage
0.00% / 1000% (10 cores allocated)

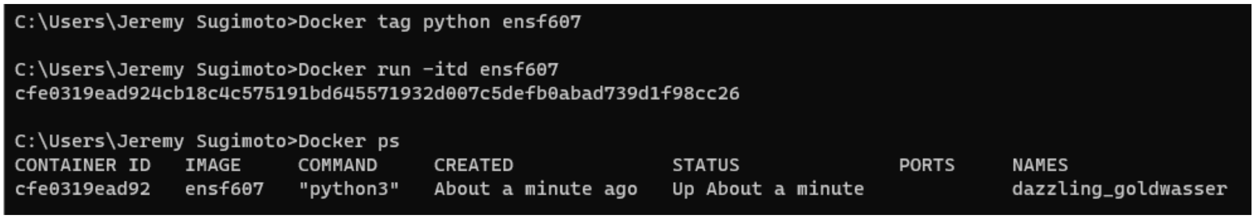
Container memory usage
11.33MB / 7.45GB

□	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
□	python_image cfe0319ead92	ensf607	Running	0%		11 minutes ago	■ ⋮ 🗑

Showing 1 item

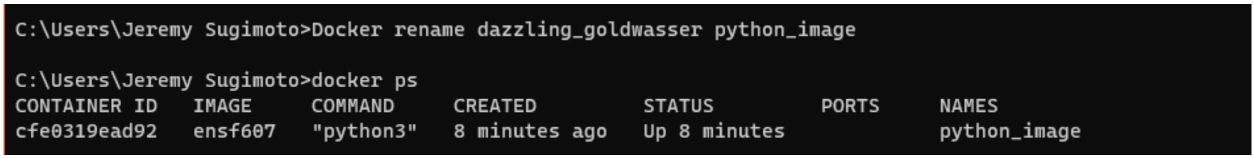
RAM 4.59 GB CPU 0.28%

6. Now rename this image to something more useful
7. Now run a container using this image within the docker
8. Now you should see all the active containers that are running with
9. You will see that docker assigns a random name (id far right) to your loaded image.



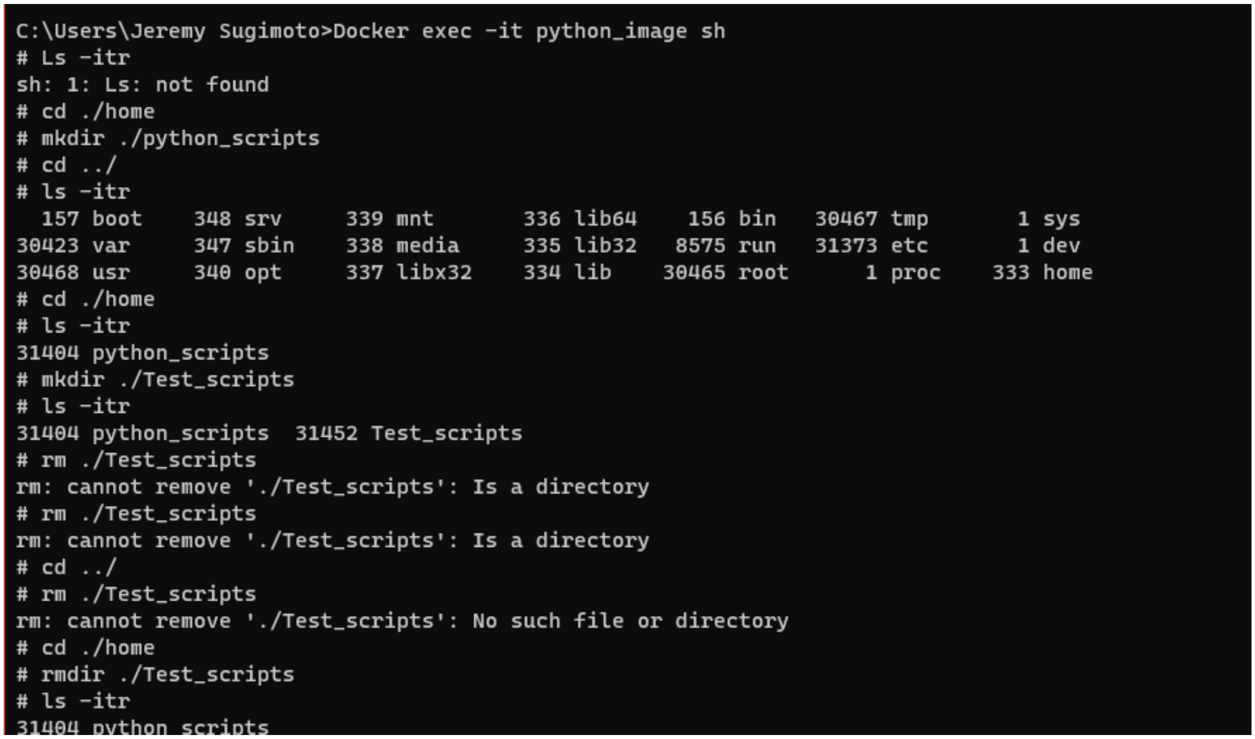
```
C:\Users\Jeremy Sugimoto>Docker tag python ensf607

C:\Users\Jeremy Sugimoto>Docker run -itd ensf607
cfe0319ead924cb18c4c575191bd645571932d007c5defb0abad739d1f98cc26

C:\Users\Jeremy Sugimoto>Docker ps
CONTAINER ID   IMAGE     COMMAND     CREATED              STATUS              PORTS     NAMES
cfe0319ead92   ensf607   "python3"   About a minute ago   Up About a minute             dazzling_goldwasser
```

10. Rename the Python container.



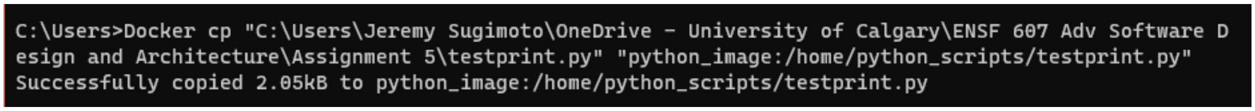
```
C:\Users\Jeremy Sugimoto>Docker rename dazzling_goldwasser python_image

C:\Users\Jeremy Sugimoto>docker ps
CONTAINER ID   IMAGE     COMMAND     CREATED         STATUS         PORTS     NAMES
cfe0319ead92   ensf607   "python3"   8 minutes ago   Up 8 minutes             python_image
```

11. Now we want to go into the container. This is a linux container so you need to use linux commands
12. To enter the container run the command.

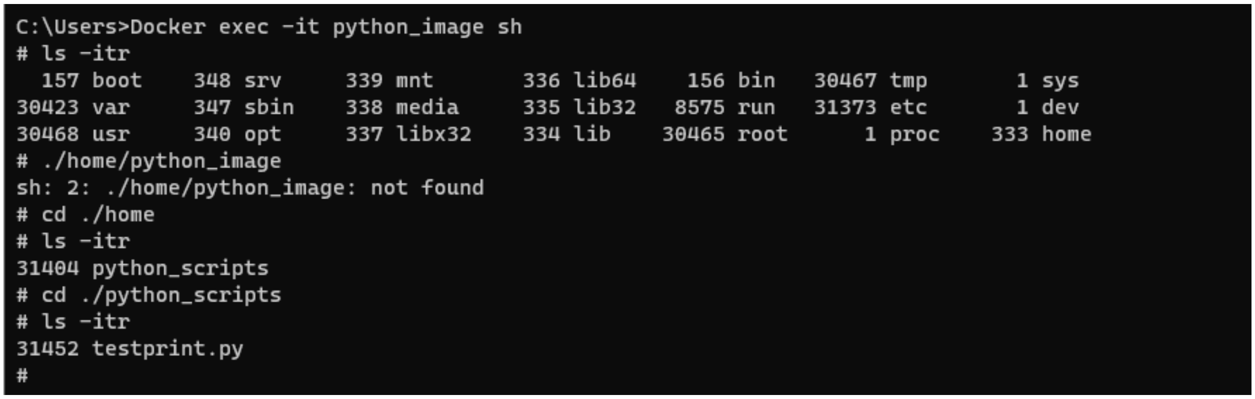


```
C:\Users\Jeremy Sugimoto>Docker exec -it python_image sh
# Ls -itr
sh: 1: Ls: not found
# cd ./home
# mkdir ./python_scripts
# cd ../
# ls -itr
  157 boot    348 srv     339 mnt       336 lib64    156 bin   30467 tmp       1 sys
30423 var     347 sbin    338 media     335 lib32   8575 run   31373 etc       1 dev
30468 usr     340 opt     337 libx32    334 lib    30465 root      1 proc    333 home
# cd ./home
# ls -itr
31404 python_scripts
# mkdir ./Test_scripts
# ls -itr
31404 python_scripts  31452 Test_scripts
# rm ./Test_scripts
rm: cannot remove './Test_scripts': Is a directory
# rm ./Test_scripts
rm: cannot remove './Test_scripts': Is a directory
# cd ../
# rm ./Test_scripts
rm: cannot remove './Test_scripts': No such file or directory
# cd ./home
# rmdir ./Test_scripts
# ls -itr
31404 python_scripts
```

13. Now we want to upload our little Python script named testprint.py.
14. Run the following command to upload the python script.

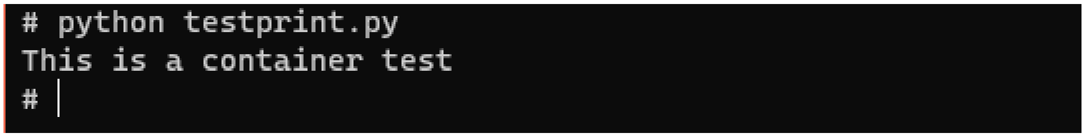


```
C:\Users>Docker cp "C:\Users\Jeremy Sugimoto\OneDrive - University of Calgary\ENSF 607 Adv Software D
esign and Architecture\Assignment 5\testprint.py" "python_image:/home/python_scripts/testprint.py"
Successfully copied 2.05kB to python_image:/home/python_scripts/testprint.py
```



```
C:\Users>Docker exec -it python_image sh
# ls -itr
  157 boot    348 srv     339 mnt       336 lib64    156 bin   30467 tmp       1 sys
30423 var     347 sbin    338 media     335 lib32   8575 run   31373 etc       1 dev
30468 usr     340 opt     337 libx32    334 lib    30465 root      1 proc    333 home
# ./home/python_image
sh: 2: ./home/python_image: not found
# cd ./home
# ls -itr
31404 python_scripts
# cd ./python_scripts
# ls -itr
31452 testprint.py
#
```

15. Run the command:



```
# python testprint.py
This is a container test
# |
```

Explore what the options -itd mean and are used for. Please include the description of these options in your document:

1. -i (--interactive)

- This option is used to keep STDIN (standard input) open even if it's not attached. It allows you to interact with the container's command-line interface.
- Often used with options like `-t` to create an interactive session for running commands inside the container.

2. -t (--tty):

- This option allocates a pseudo-TTY (teletypewriter) and associates it with the container. It is typically used in conjunction with the `-i` option to provide a terminal interface for the container.
- When you run a container with `-it`, you can enter and interact with the container's command-line just like you would on a local terminal.

3. -d (--detach):

- This option tells Docker to run the container in the background (detached mode). When a container is run in detached mode, it won't hold your terminal, and you can continue to use it for other commands without the container's output appearing in your terminal.
- The container will continue running in the background until you stop it explicitly using the `docker stop` command.

So, in the command `docker run -itd ensf607`, the `-it` options are used to create an interactive session, and the `-d` option is used to run the container in the background.