

Balkarn Gill - 30202219 - ENSF607 Assignment 5

4)

```
Last login: Mon Oct 30 05:34:21 on console
(base) gill@gills-mbp ~ % docker pull python
Using default tag: latest
latest: Pulling from library/python
8024d4fb53b2: Pull complete
3d826ee8aa65: Pull complete
198068495d09: Pull complete
509db9a897ae: Pull complete
8fd6b529c225: Pull complete
113a282385bf: Pull complete
a32c3e5bd8ca: Pull complete
c34a97c23859: Pull complete
Digest: sha256:7b8d65a924f596eb65306214f559253c468336bcae09fd575429774563460caf
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
(base) gill@gills-mbp ~ %
```

5)

```
View a summary of image vulnerabilities and recommendations → docker scout quickview python
(base) gill@gills-mbp ~ % Docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
python latest 69ee9abe2bda 2 weeks ago 1.02GB
(base) gill@gills-mbp ~ %
```

6)

```
(base) gill@gills-mbp ~ % Docker tag python ensf607
(base) gill@gills-mbp ~ %
```

7)

```
(base) gill@gills-mbp ~ % Docker run -itd ensf607
32267e66a14973c9afca104788f25dc2a81451a4037475e43b858bf0ab31f25f
```

-i (or --interactive): This option allows you to keep the standard input (stdin) of the container open. It is used to make the container interactive, meaning you can interact with the running container's command prompt. It's commonly used with options like -t to allocate a pseudo-TTY, which helps in formatting the output and interacting with the container's command line.

-t (or --tty): This option is used to allocate a pseudo-TTY (terminal). It is often used in conjunction with the -i option to create an interactive session with the container. A TTY is a

terminal emulation that allows you to interact with the command-line interface of the container as if you were working in a traditional terminal.

-d (or --detach): This option is used to run the container in the background, detaching it from the terminal. When you run a container with this option, you will receive a container ID as output, and the container will run in the background. This is useful for running containers as services or daemons.

8)

```
(base) gill@gills-mbp ~ % Docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
32267e66a149   ensf607   "python3"               About a minute ago    Up About a minute                vigorous_clarke
(base) gill@gills-mbp ~ %
```

10)

```
(base) gill@gills-mbp ~ % docker rename vigorous_clarke bkdocker
(base) gill@gills-mbp ~ %
```

12)

```
(base) gill@gills-mbp ~ % docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
32267e66a149   ensf607   "python3"               3 minutes ago    Up 3 minutes                bkdocker
(base) gill@gills-mbp ~ %
```

13)

```
# ls -ltr
total 52
drwxr-xr-x  2 root root 4096 Sep 29 20:04 boot
drwxr-xr-x  1 root root 4096 Oct 30 00:00 var
drwxr-xr-x  1 root root 4096 Oct 30 00:00 usr
drwxr-xr-x  2 root root 4096 Oct 30 00:00 srv
lrwxrwxrwx  1 root root    8 Oct 30 00:00 sbin -> usr/sbin
drwxr-xr-x  2 root root 4096 Oct 30 00:00 opt
drwxr-xr-x  2 root root 4096 Oct 30 00:00 mnt
drwxr-xr-x  2 root root 4096 Oct 30 00:00 media
lrwxrwxrwx  1 root root    7 Oct 30 00:00 lib -> usr/lib
lrwxrwxrwx  1 root root    7 Oct 30 00:00 bin -> usr/bin
drwxr-xr-x  1 root root 4096 Nov  1 02:04 run
drwx----- 1 root root 4096 Nov  1 09:23 root
drwxrwxrwt  1 root root 4096 Nov  1 09:33 tmp
drwxr-xr-x  1 root root 4096 Nov  4 04:13 etc
dr-xr-xr-x 12 root root    0 Nov  4 04:13 sys
dr-xr-xr-x 245 root root    0 Nov  4 04:13 proc
drwxr-xr-x  5 root root 360 Nov  4 04:13 dev
drwxr-xr-x  1 root root 4096 Nov  4 04:18 home
# cd ./home
# mkdir ./python_scripts
mkdir: cannot create directory './python_scripts': File exists
# cd ../
# ./
```

14)

```
(base) gill@gills-mbp ~ % docker cp /Users/gill/Downloads/testprint.py bkdocker:/home/python_scripts/testprint.py
Successfully copied 2.05kB to bkdocker:/home/python_scripts/testprint.py
(base) gill@gills-mbp ~ %
```

15)


```
(base) gill@gills-mbp ~ % docker exec -it bkcontainer sh
# cd /home/python_scripts
#
# python testprint.py
This is a container test
```





Containers [Give feedback](#)

Container CPU usage ⓘ
0.00% / 1000% (10 cores available)

Container memory usage ⓘ
11.4MB / 7.48GB

Show charts ▼

 ☒ Only show running containers

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 bkcontainer c23dbc7e1a7	ensf607	Running	0%		16 minutes ago	  

Showing 1 item