


Assignment 1

Explain the use of the following docker Commands using an example :-

A 1-) Pull:- The docker pull is a cmd used to download a docker image or repository locally on the host from a public/private registry. Because when we run any container and the docker image is not locally on the machine it first pulls it from the registry. Most images are downloaded from hub.docker.com and that is when we create a custom docker image as we use the main docker image as a base image. An example of a docker pull command is:

- a) docker pull python
- b) docker pull ubuntu

c) docker pull openjdk
d) docker pull reactnativecommunity

A2) Build- The docker build is a cmd used to build an image from the docker file in the current directory and then able to tag the image. The docker build can refer to any files located in that specified Uniform Resource Locator (URL) or in the path. An example of a docker build command is:-

- a) docker build -t myimage1:1.0
- b) docker build -< Dockerfile
- c) docker build .
- d) docker build http://myrepo.git

A3.) Run:- The docker run is a cmd first creates a writeable container layer over the specified image and it then starts using the specified command. Docker run can also be used with a combination of docker commit to change the command that a container runs. E.g.

- (a) docker run -w/usr/francis/dir
-l -t ubuntu pwd [to set working directory]
- (b) docker run -p 127.0.1:80:8080/tcp
ubuntu bash [publish port]
- (c) docker run --expose 8080 ubuntu bash
[expose port]
- (d) docker container run --name web
-p 8080:80 alpine 4.0

A4) `attach`: It is used to attach the terminal standard input, output and also error to a running container using the container name. It also allows one terminal to attach also a running container to it. It allows multiple connections to view the same container and to see what each is typing there. E.g.

- a) `docker attach <container>`
- b) `docker attach test`
- c) `docker attach nodeapi`

PART B

Write a Dockerfile that will create a docker image using the most recent OpenJDK

```
FROM openjdk:latest
WORKDIR /usr/src/app
COPY ..
EXPOSE 80
CMD ["bin/bash"]
```

PART C

Write a docker-compose.yml file
that will create 2 containers from
an OpenJDK image.

version: "3.9"

services:

java1:

image: openjdk:latest

container_name: java1

ports:

- "80:8080"

volumes:

- /Users/francis/cmpt/436/java/code

links:

- java2

expose:

- "8080"

java2:

image: openjdk:latest

container_name: java2

Ports:

- "81:8080"

Volumes:

- /Users/francis/cmpt/436/java/code