Richard Karras

Rek236 10527391

CMPT 353

A) Explain the usage of the following docker commands.

A1) *build:* Build creates a new Docker image relying on a Dockerfile and a "context". The context is a set of files at a specified location, be that a file path or url.

Usage is "docker build [options/switches] PATH | URL | -"

A2) *run:* Run creates a writeable container layer using a specified image and starts the container using the specified command.

Usage is "docker run [options/switches] IMAGE [COMMAND] [Arguments]"

A stopped container can be restarted with "docker start" rather than "docker run"

B) Write a Dockerfile that will create a docker image tagged my/python using the most recent python image on Docker Hub. Make sure that port 8080 is exposed and that a command line is available upon starting the image in a container.

Dockerfile:

```
FROM python:latest

EXPOSE 8080

CMD ["/bin/bash"]
```

Command to build (Windows):

```
docker build -t my/python:1.0 .
```

C) Write the docker command that will start a container named python1 using the image tagged my/python. Ensure that the container has access to port 8080 and a folder on the host's file system.

Command:

```
docker run -v "/mnt/c/users/Richard/OneDrive - University of
Saskatchewan/cmpt 353/Assignments/Assignment 1/python-demo":/python-demo/
--name python1 -p 80:8080 -it my/python
```

D) Explain how a developer can use the container named python1 to develop and run a simple "hello world" python program.

With the Dockerfile from part B, and the command from part C, a developer could use a text editor to create a simple "Hello world" script for python and run it inside the container. Currently with the access as it is, they would **run** the container for the first time, or **start** if they had previously built and run it. This will take them to a bash terminal. If they have created a .py file and stored it in the python-demo folder noted in part C, they could access from the local file system and run it, rather than editing inside the container. It would be available inside the container at /python-demo/hello-world.py (or however they named the file)