Docker Tutorial Answers

Nian Ran

1. What user are you logged in as by default?
- root

2. If you start and then exit an interactive container, and then use the docker run -it ubuntu:xenial /bin/bash command again, is it the same container? How can you tell?
No, running the same command: docker run -it ubuntu:xenial /bin/bash again creates a new container every time. Because each run gets a different container ID, I can see through “docker ps -a”, it gives the following output:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
5a107db0b9e1	ubuntu:xenial	"/bin/bash"	23 seconds ago	Exited (0) 16 seconds ago
b66d397c3ac5	ubuntu:xenial	"/bin/bash"	About a minute ago	Exited (0) 27 seconds ago

3. Run the image you just built. Since we specified the default CMD, you can just do docker run -it mypython:latest. What do you observe?
Docker automatically launches an interactive Python 3.6 REPL when the container starts.
So the moment the container runs, it drops you directly into the Python interpreter instead of a shell.

4. Write and build a Dockerfile that installs the packages fortune and fortunes-min and runs the fortune executable (located in /usr/games/fortune after you install it). Note that you won’t need to use the -it flags when you run the container as fortune doesn’t need STDIN. Submit your Dockerfile with this lab. Hint: if you’re having trouble writing your Dockerfile, try booting an interactive container and installing both packages. Translate what you did into a Dockerfile. How can you translate what you did interactively to a Dockerfile?
- By taking each command I ran interactively (like apt-get update, apt-get install, and the program I executed) and writing them into the Dockerfile as RUN and CMD instructions.

5. Paste the output of your docker images command after questions 1 and 2.
- Q1 output:

```
(ai) root@pm-621f:~/nian/dockerlab# docker run -it mypython:latest
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Q2 output:

```
(ai) root@pm-621f:~/nian/dockerlab# docker run myfortune:latest
```

Don't hate yourself in the morning -- sleep till noon.
(ai) root@pm-621f:~/nian/dockerlab# docker run myfortune:latest
You have a deep appreciation of the arts and music.

6. With httpd running in a detached container, run /bin/bash on the same container and paste the output of ps aux. Observe that there's very few processes running as compared to running ps aux on your VM. Why is this the case?
- Inside a container, only the processes started by the container itself run, usually just the main process defined by the image (in this case, httpd) plus anything you exec manually. Containers don't run a full operating system like a VM does, so they have no background services, no systemd, and no daemons. Therefore ps aux shows only a few processes instead of the many system processes you would see on a full VM.

7. Paste your Dockerfile for the Node.js web application
- see attached Dockerfile

8. Paste your docker-compose.yml file
- see attached docker-compose.yml