Docker Tutorial

Charlie Drury

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?**
   It's a new container each time (but based on the same image). In both xenial and fedora examples you can see the user and container by user@<container_id> in the shell prompt, and I can see the container id is different each time.

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?**

   We're now interactively running Python 3.6.9 via REPL on a linux base, independently of my personal mac machine.

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?**

   (Attached dockerfile.)

5. **Paste the output of your docker images command after questions 1 and 2.**

   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.
   >>>

   You will have long and healthy life.

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?**

The container processes are very minimal, without many of the background daemons or services that we expect. Only the processes that are explicitly started are running. This is following the ethos of keeping containers to be as lightweight as possible.

7. **Paste your Dockerfile for the Node.js web application**
   (attached)

8. **Paste your docker-compose.yml file**
   (attached)