

Docker Tutorial

Abhishek Das

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

Ans- By default, we are logged in 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?

Ans- Every time you exit an interactive container and run the same command, you get a new container. You can tell if you see the container id, it is different every 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?

Ans- When I run the image I just built, I see an interactive Python 3.6.9 running on a Linux base. It is not running on my Mac, but rather on a separate Linux base.

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?

Ans- I have uploaded it on the repo

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

Ans-

```
(base) abjodas@Abhisheks-MacBook-Pro effective_python % 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.
>>> █
```

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?

Ans- There is not too many processes running compared to my Mac machine. This is because the docker container is an isolated process on the same kernel as the host. Therefore, there isn't too many background daemons or init system or login services.

7. Paste your Dockerfile for the Node.js web application

Ans- Pushed to the repo

8. Paste your docker-compose.yml file

Ans- Pushed to the repo