Play with Docker Classroom

Time spent: 60 min

Concepts in this exercise:

- Docker engine
- Containers & images
- Image registries and Docker Store (AKA Docker Hub)
- Container isolation

1.0 Running your first container

```
[node1] (local) root@192.168.0.33 ~
$ docker container run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash
```

Differences between containers and VM

- The VM is a hardware abstraction: it takes physical CPUs and RAM from a host, and divides and shares it across several smaller virtual machines
- A container is an application abstraction: the focus is really on the OS and the application, and not so much the hardware abstraction

1.1 Docker Images

Docker Container Run

```
[node1] (local) root@192.168.0.33 ~
$ docker image pull alpine
Using default tag: latest
latest: Pulling from library/alpine
4fe2ade4980c: Pull complete
Digest: sha256:621c2f39f8133acb8e64023a94dbdf0d5ca81896102b9e57c0dc184cadaf5528
Status: Downloaded newer image for alpine:latest
[node1] (local) root@192.168.0.33 ~
$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              latest              196d12cf6ab1        7 weeks ago         4.41MB
hello-world         latest              4ab4c602aa5e        7 weeks ago         1.84kB
```

```
[node1] (local) root@192.168.0.33 ~
$ docker container run alpine ls -l
total 8
drwxr-xr-x    2 root     root          4096 Sep 11 20:23 bin
drwxr-xr-x    5 root     root           340 Oct 31 23:55 dev
drwxr-xr-x    1 root     root            66 Oct 31 23:55 etc
drwxr-xr-x    2 root     root             6 Sep 11 20:23 home
drwxr-xr-x    5 root     root           278 Sep 11 20:23 lib
drwxr-xr-x    5 root     root            44 Sep 11 20:23 media
drwxr-xr-x    2 root     root             6 Sep 11 20:23 mnt
dr-xr-xr-x  297 root     root             0 Oct 31 23:55 proc
drwx------    2 root     root             6 Sep 11 20:23 root
drwxr-xr-x    2 root     root             6 Sep 11 20:23 run
drwxr-xr-x    2 root     root          4096 Sep 11 20:23 sbin
drwxr-xr-x    2 root     root             6 Sep 11 20:23 srv
dr-xr-xr-x   13 root     root             0 Sep 25 16:51 sys
drwxrwxrwt    2 root     root             6 Sep 11 20:23 tmp
drwxr-xr-x    7 root     root            66 Sep 11 20:23 usr
drwxr-xr-x   11 root     root           125 Sep 11 20:23 var
[node1] (local) root@192.168.0.33 ~
$ docker container run alpine echo "hello from alpine"
hello from alpine
```

```
[node1] (local) root@192.168.0.43 ~
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STAT
US                      PORTS               NAMES
e313f967e9f9        alpine              "/bin/sh"                32 seconds ago      Exit
ed (0) 11 seconds ago                       lucid_dubinsky
83793049220f        alpine              "/bin/sh"                37 seconds ago      Exit
ed (0) 36 seconds ago                       vibrant_nightingale
b898de6414df        alpine              "echo 'hello from al…"   44 seconds ago      Exit
ed (0) 43 seconds ago                       awesome_fermat
[node1] (local) root@192.168.0.43 ~
$
```

Summary

In this lab, I have learned how to run the container and pull the image, and I have studied the mechanism of isolation containers.