Lab 1-1 - Container fundamentals

In this lab we will be getting acquainted with the fundamentals of using docker to manage images and containers

1. Ensure you have an SSH client (eg, Putty on windows)
2. Use an SSH client to log into the server using the information provided by the instructor
3. Ensure docker is installed

```
$ docker --version

Docker version 19.03.8, build afacb8b7f0
```

4. Look at docker system information

```
$ docker info

Client:
Debug Mode: false

Server:
Containers: 1
Running: 0
Paused: 0
Stopped: 1
Images: 1
Server Version: 19.03.8
...
```

5. See if there are any docker images loaded

```
$ docker image ls

REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
```

6. See if there are any docker containers

```
$ docker container ls -a
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

7. Get the docker hello-world image

```
$ docker pull hello-world

Using default tag: latest
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest:
sha256:6a65f928fb91fcfbc963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f1
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
```

8. repeat stages 5. and 6. to see any changes (you should see one image, and no containers)

9. Run the hello world image

```
$ docker run hello-world

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

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

10. repeat stages 5. and 6. to see any changes (you should see one image, and one container).

> FYI: Docker uses a semi-random name generator to assign a readable name to a container if you did not provide one, the names often contain the surnames of people who are notable in the

> IT/Computer science/Science world, such as Torvalds, Dijkstra, Gates, Stallman and Wozniak. The name exists to make it easier to manage containers without having to use a long number

11. Run the hello-world container again, this time we'll specify our own name, and repeat stage 6.

```
$ docker run --name hello-world hello-world
...
$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES
00a610fa9164        hello-world         "/hello"            2 seconds ago
Exited (0) 1 second ago                         hello-world
3a8997b1e409        hello-world         "/hello"            5 minutes ago
Exited (0) 5 minutes ago                        quizzical_dijkstra
```

    There are now two containers from the one image

12. Lets try using the option to remove a container once it's finished executing

```
$ docker run --rm hello-world
...

$ docker container ls -a
```

    You will notice that there are still only two containers.

13. What happens if you try to remove the local copy of the hello-world image

```
$ docker rmi hello-world

Error response from daemon: conflict: unable to remove repository reference
"hello-world" (must force) - container 3a8997b1e409 is using its referenced
image bf756fb1ae65
```

    Even though the containers aren't running, the image is still referenced by the two containers we've previously run, so we can't remove that image.

14. Remove the containers, your first container will have different names to what is shown above, use the name from running $ docker container ls -a on your system.

```
$ docker rm quizzical_dijkstra
quizzical_dijkstra

$ docker rm hello-world
hello-world
```

```
$ docker container ls -a
```

You should now have no containers left, try to remove the image again

```
$ docker rmi hello-world

Untagged: hello-world:latest
Untagged: hello-
world@sha256:6a65f928fb91fcfbc963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f
1
Deleted:
sha256:bf756fb1ae65adf866bd8c456593cd24beb6a0a061dedf42b26a993176745f6b
Deleted:
sha256:9c27e219663c25e0f28493790cc0b88bc973ba3b1686355f221c38a36978ac63
```

15. Check to see if there are any images

```
$ docker image ls
```

16. We should be back to a clean slate, lets see how much disk is being used by docker

```
$ docker system df

TYPE                TOTAL               ACTIVE              SIZE
RECLAIMABLE
Images              0                   0                   0B
0B
Containers          0                   0                   0B
0B
Local Volumes       0                   0                   0B
0B
Build Cache         0                   0                   0B
0B
```

17. From this state, try to run a hello-world container

```
$ docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest:
sha256:6a65f928fb91fcfbc963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f1
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.
...
```

docker detected that the image did not exist locally, it checked the current registry for the hello-world repository and by default, downloaded and used the :latest tag, you are now back to having one image and one container (try 5. and 6. again)

That is the end of Lab 1-1, you have learned how to get images from a registry, run containers from those images and do some management of containers and images.