

Exercise 1: Running containers

In this exercise, we'll learn the basics of pulling images, starting, stopping, and removing containers.

Pulling an image

To run containers, we'll first need to pull some images.

1. Let's see what images we have currently on our machine, by running `docker images`:

```
alap@sf-cpu-036:~/Exercises-docker$ docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
alap@sf-cpu-036:~/Exercises-docker$
```

2. On a fresh Docker installation, we should have no images. Let's pull one from Dockerhub.

We can search for images using `docker search <keyword>`

```
alap@sf-cpu-036:~/Exercises-docker$ docker search redis
NAME                DESCRIPTION                STARS     OFFICIAL   AUTOMATED
redis               Redis is an open source key-value store that... 12030     [OK]
redislabs/redisearch Redis With the RedisSearch module pre-loaded... 56
redislabs/redisinsight RedisInsight - The GUI for Redis 87
```

Run `docker pull ubuntu:22.04` to pull an image of Ubuntu 22.04 from DockerHub.

```
alap@sf-cpu-036:~/Exercises-docker$ docker pull ubuntu:22.04
22.04: Pulling from library/ubuntu
2ab09b027e7f: Pull complete
Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babc295a0428a6d21
Status: Downloaded newer image for ubuntu:22.04
docker.io/library/ubuntu:22.04
```

3. We can also pull different versions on the same image.

Run `docker pull ubuntu:16.10` to pull an image of Ubuntu 16.10.

```
alap@sf-cpu-036:~/Exercises-docker$ docker pull ubuntu:16.10
16.10: Pulling from library/ubuntu
dca7be20e546: Pull complete
40bca54f5968: Pull complete
61464f23390e: Pull complete
d99f0bcd5dc8: Pull complete
120db6f90955: Pull complete
Digest: sha256:8dc9652808dc091400d7d5983949043a9f9c7132b15c14814275d25f94bca18a
Status: Downloaded newer image for ubuntu:16.10
docker.io/library/ubuntu:16.10
```

Then when we run `docker images` again, we should get:

```
alap@sf-cpu-036:~/Exercises-docker$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	22.04	08d22c0ceb15	6 weeks ago	77.8MB
ubuntu	16.10	7d3f705d307c	5 years ago	107MB

4. Over time, your machine can collect a lot of images, so it's nice to remove unwanted images.

Run `docker rmi <IMAGE ID>` to remove the Ubuntu 16.10 image we won't be using.

```
alap@sf-cpu-036:~/Exercises-docker$ docker rmi 7d3f705d307c
Untagged: ubuntu:16.10
Untagged: ubuntu@sha256:8dc9652808dc091400d7d5983949043a9f9c7132b15c14814275d25f94bca18a
Deleted: sha256:7d3f705d307c7c225398e04d4c4f8512f64eb8a65959a1fb4514dfde18a047e7
Deleted: sha256:d9db289b9342d9617596cd6ee3bba988629e24d9afa5db4e4b0e4e491c65007d
Deleted: sha256:a87725e8597b97f2399bc3aa50b0e2eec903b8ce19055668d3befb012918205c
Deleted: sha256:38cf10a2801529348366953e9b933d3524360dedc91d3e4d5d7f941da0c973c9
Deleted: sha256:61172966249d43026dbd017eec3a9575e37bddf8a269a9f09ecb559d7bfe7fef
Deleted: sha256:57145c01eb80040fdd0a24cde20af4788605b49593188d4f7efab099af89a08e
```

Alternatively, you can delete images by tag or by a partial image ID. In the previous example, the following would have been equivalent:

1. `docker rmi 7d`
2. `docker rmi ubuntu:16.10`

Running `docker images` should reflect the deleted image.

```
alap@sf-cpu-036:~/Exercises-docker$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	22.04	08d22c0ceb15	6 weeks ago	77.8MB

A nice shortcut for removing all images from your system is `docker rmi $(docker images -a -q)`

Running our container

Using the Ubuntu 22.04 image we downloaded, we can run a our first container. Unlike a traditional virtualization framework like VirtualBox or VMWare, we can't just start a virtual machine running this image without anything else: we have to give it a command to run.

The command can be anything you want, as long as it exists on the image. In the case of the Ubuntu image, it's a Linux kernel with many of the typical applications you'd find in a basic Linux environment.

2. Let's do a very simple example. Run `docker run ubuntu:22.04 /bin/echo 'Hello world!'`

```
alap@sf-cpu-036:~/Exercises-docker$ docker run ubuntu:22.04 /bin/echo 'Hello World'
Hello World
```

The `/bin/echo` command is a really simple application that just prints whatever you give it to the terminal. We passed it 'Hello world!', so it prints Hello world! to the terminal.

When you run the whole `docker run` command, it creates a new container from the image specified, then runs the command inside the container. From the previous example, the Docker container started, then ran the `/bin/echo` command in the container.

3. Let's check what containers we have after running this. Run `docker ps`

```
alap@sf-cpu-036:~/Exercises-docker$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
```

That's strange: no containers right? The `ps` command doesn't show stopped containers by default, add the `-a` flag.

```
alap@sf-cpu-036:~/Exercises-docker$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
581d38f8e2fe  ubuntu:22.04  "echo 'Hello World'"    6 minutes ago  Exited (0) 6 minutes ago  objective_buck
ab685adebf93  ubuntu:22.04  "/bin/echo 'Hello Wo..." 6 minutes ago  Exited (0) 6 minutes ago  bold_spence
```

Okay, there's our container. But why is the status "Exited"?

Docker containers only run as long as the command it starts with is running. In our example, it ran `/bin/echo` successfully, printed some output, then exited with status code 0 (which means no errors.) When Docker saw this command exit, the container stopped.

- Let's do something a bit more interactive. Run `docker run ubuntu:22.04 /bin/bash`

```
alap@sf-cpu-036:~/Exercises-docker$ docker run -it ubuntu:22.04 /bin/bash
root@44f6ab287cc1:/# pwd
/
root@44f6ab287cc1:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@44f6ab287cc1:/# exit
exit
```

Removing containers

We can remove the containers using `docker rm <container_id>/<container_name>`

```
alap@sf-cpu-036:~/Exercises-docker$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
44f6ab287cc1   ubuntu:22.04 "/bin/bash"             About a minute ago    Exited (0) About a minute ago
581d38f8e2fe   ubuntu:22.04 "echo 'Hello World'"    17 minutes ago       Exited (0) 17 minutes ago
ab685adebf93   ubuntu:22.04 "/bin/echo 'Hello Wo..." 18 minutes ago       Exited (0) 18 minutes ago
alap@sf-cpu-036:~/Exercises-docker$ docker rm 44
44
alap@sf-cpu-036:~/Exercises-docker$ docker rm 58
58
alap@sf-cpu-036:~/Exercises-docker$ docker rm ab
ab
alap@sf-cpu-036:~/Exercises-docker$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
```