Running Docker containers

INTRODUCTION TO DOCKER

Tim Sangster
Software Engineer @ DataCamp

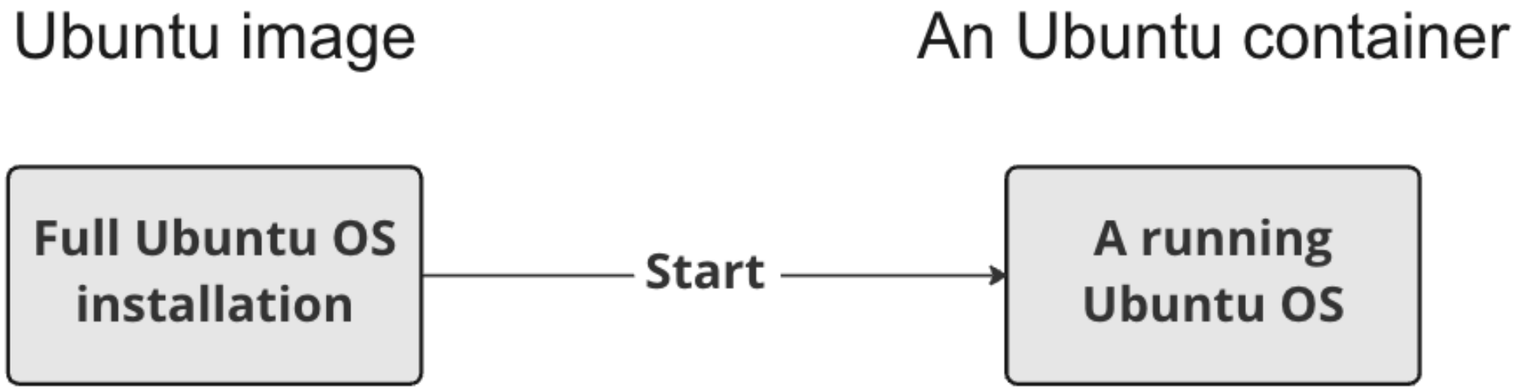
datacamp

Prerequisite

Command	Usage
nano <file-name>	Opens <file-name> in the nano text editor
touch <file-name>	Creates an empty file with the specified name
echo "<text>"	Prints <text> to the console
<command> >> <file>	Pushes the output of <command> to the end of <file>
<command> -y	Automatically respond yes to all prompts from <command>

The Docker CLI

- Docker command line interface will send instructions to the Docker daemon.
- Every command starts with `docker` .



Docker container output

```
docker run <image-name>
```

```
docker run hello-world
```

```
Hello from Docker!

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

Choosing Docker container output

```
docker run <image-name>
```

```
docker run ubuntu
```

```
repl@host:/# docker run ubuntu


repl@host:/#
```

An interactive Docker container

Adding `-it` to `docker run` will give us an interactive shell in the started container.

```
docker run -it <image-name>
```

```
docker run -it ubuntu
```

```
docker run -it ubuntu
repl@container:/#
```

```
repl@container:/# exit
exit
repl@host:/#
```

Running a container detached

Adding `-d` to `docker run` will run the container in the background, giving us back control of the shell.

```
docker run -d <image-name>
docker run -d postgres
```

```
repl@host:/# docker run -d postgres
4957362b5fb7019b56470a99f52218e698b85775af31da01958bab198a32b072
repl@host:/#
```

Listing and stopping running containers

```
docker ps
```

```
repl@host:/# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED
4957362b5fb7   postgres   "docker-entrypoint.s…"   About a minute ago
STATUS              PORTS      NAMES
Up About a minute   5432/tcp   awesome_curie
```

```
docker stop <container-id>
```

```
repl@host:/# docker stop cf91547fd657
cf91547fd657
```

Summary of new commands

Usage	Command
Start a container	docker run <image-name>
Start an interactive container	docker run -it <image-name>
Start a detached container	docker run -d <image-name>
List running containers	docker ps
Stop a container	docker stop <container-id>

Let's practice!

INTRODUCTION TO DOCKER

Working with Docker containers

INTRODUCTION TO DOCKER

Tim Sangster
Software Engineer @ DataCamp

Listing containers

```
repl@host:/# docker ps
CONTAINER ID   IMAGE     .. CREATED          STATUS          ...  NAMES
3b87ec116cb6   postgres    2 seconds ago   Up 1 second     ...  adoring_germain
8a7830bbc787   postgres    3 seconds ago   Up 2 seconds    ...  exciting_heisenberg
fefdf1687b39   postgres    3 seconds ago   Up 2 seconds    ...  vigilant_swanson
b70d549d4611   postgres    4 seconds ago   Up 3 seconds    ...  nostalgic_matsumoto
a66c71c54b92   postgres    4 seconds ago   Up 4 seconds    ...  lucid_matsumoto
8d4f412adc3f   postgres    6 seconds ago   Up 5 seconds    ...  fervent_ramanujan
fd0b3b2a843e   postgres    7 seconds ago   Up 6 seconds    ...  cool_dijkstra
0d1951db81c4   postgres    8 seconds ago   Up 7 seconds    ...  happy_sammet
...
```

Named containers

```
docker run --name <container-name> <image-name>
```

```
repl@host:/# docker run --name db_pipeline_v1 postgres
repl@host:/# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED
43aa37614330   postgres   "docker-entrypoint.s…"   About a minute ago
STATUS              PORTS      NAMES
Up About a minute   5432/tcp   db_pipeline_v1
```

```
docker stop <container-name>
```

```
repl@host:/# docker stop db_pipeline_v1
```

Filtering running containers

```
docker ps -f "name=<container-name>"
```

```
repl@host:/# docker ps -f "name=db_pipeline_v1"
CONTAINER ID   IMAGE      COMMAND                  CREATED
43aa37614330   postgres   "docker-entrypoint.s…"   About a minute ago
STATUS              PORTS      NAMES
Up About a minute   5432/tcp   db_pipeline_v1
```

Container logs

```
docker logs <container-id>
```

```
repl@host:/# docker logs 43aa37614330
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".

PostgreSQL init process complete; ready for start up.

2022-10-24 12:10:40.318 UTC [1] LOG:  database system is ready to accept connect..
```

Live logs

```
docker logs -f <container-id>
```

```
repl@host:/# docker logs -f 43aa37614330
PostgreSQL init process complete; ready for start up.

2022-10-24 12:10:40.309 UTC [1] LOG:  starting PostgreSQL 14.5 (Debian 14.5-1.pg..
2022-10-24 12:10:40.309 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port ..
2022-10-24 12:10:40.309 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2022-10-24 12:10:40.311 UTC [1] LOG:  listening on Unix socket "/var/run/postgre..
2022-10-24 12:10:40.315 UTC [62] LOG:  database system was shut down at 2022-10-..
2022-10-24 12:10:40.318 UTC [1] LOG:  database system is ready to accept connect..
```

Cleaning up

```
docker container rm <container-id>
```

```
repl@host:/# docker stop 43aa37614330
43aa37614330
repl@host:/# docker container rm 43aa37614330
43aa37614330
```

Summary of new commands

Usage	Command
Start container with a name	docker run --name <container-name> <image-name>
Filter running container on name	docker ps -f "name=<container-name>"
See existing logs for container	docker logs <container-id>
See live logs for container	docker logs -f <container-id>
Exit live log view of container	CTRL+C
Remove stopped container	docker container rm <container-id>

Let's practice!

INTRODUCTION TO DOCKER

Managing local docker images

INTRODUCTION TO DOCKER

Tim Sangster
Software Engineer @ DataCamp

datacamp

Docker Hub is the world's largest library and community for container images

Browse over 100,000 container images from software vendors, open-source projects, and the community.

busybox
Official
1B+

ubuntu
Official
1B+

httpd
Official
1B+

mariadb
Official
1B+

memcached
Official
1B+

registry
Official
1B+

influxdb
Official
500M+

postgres
Official
1B+

haproxy
Official
500M+

openjdk
Official
1B+

Pulling an image

```
docker pull <image-name>
```

```
docker pull postgres
docker pull ubuntu
```

```
repl@host:/# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
7050e35b49f5: Pull complete
Digest: sha256:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
```

Image versions

Supported tags and respective `Dockerfile` links

- `18.04`, `bionic-20221019`, `bionic`
- `20.04`, `focal-20221019`, `focal`
- `22.04`, `jammy-20221020`, `jammy`, `latest`
- `22.10`, `kinetic-20221024`, `kinetic`, `rolling`
- `14.04`, `trusty-20191217`, `trusty`
- `16.04`, `xenial-20210804`, `xenial`

```
docker pull <image-name>:<image-version>
```

```
docker pull ubuntu:22.04
docker pull ubuntu:jammy
```

Listing images

```
docker images
```

```
repl@host:/# docker images
REPOSITORY    TAG                 IMAGE ID       CREATED          SIZE
hello-world   latest              46331d942d63   7 months ago     9.14kB
ubuntu        bionic-20210723     7c0c6ae0b575   15 months ago    56.6MB
postgres      12.7                f076c2fa35f5   15 months ago    300MB
postgres      10.3                cbb7481ff9d5   4 years ago      232MB
...
```

Removing images

```
docker image rm <image-name>
```

```
repl@host:/# docker image rm hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f35..
Deleted: sha256:46331d942d6350436f64e614d75725f6de3bb5c63e266e236e04389820a234c4
Deleted: sha256:efb53921da3394806160641b72a2cbd34ca1a9a8345ac670a85a04ad3d0e3507
```

```
repl@host:/# docker image rm hello-world
Error response from daemon: conflict: unable to remove repository reference
"hello-world" (must force) - container 96a7b7b0c535 is using its
referenced image 46331d942d63
```

Cleaning up containers

```
docker container prune
```

```
repl@host:/# docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
4a7f7eebae0f63178aff7eb0aa39cd3f0627a203ab2df258c1a00b456cf20063
f98f9c2aa1eaf727e4ec9c0283bc7d4aa4762fbdba7f26191f26c97f64090360

Total reclaimed space: 212 B
```

Cleaning up images

```
docker image prune -a
```

```
repl@host:/# docker image prune -a
WARNING! This will remove all images without at least one container associated t..
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: alpine:latest
untagged: alpine@sha256:3dcdb92d7432d56604d4545cbd324b14e647b313626d99b889d0626d..
deleted: sha256:4e38e38c8ce0b8d9041a9c4fefe786631d1416225e13b0bfe8cfa2321aec4bba
deleted: sha256:4fe15f8d0ae69e169824f25f1d4da3015a48feeeeebb265cd2e328e15c6a869f

Total reclaimed space: 16.43 MB
```

Dangling images

```
docker images
```

```
repl@host:/# docker images
REPOSITORY    TAG                 IMAGE ID       CREATED          SIZE
testsql       latest              6c49f0cce145   7 months ago     3.73GB
<none>        <none>              a22b8450b88f   7 months ago     3.73GB
<none>        <none>              10dd2d03f59c   7 months ago     3.73GB
<none>        <none>              878bae40320b   7 months ago     3.73GB
<none>        <none>              4ea70583ba54   7 months ago     3.75GB
<none>        <none>              3c64576a3a7d   7 months ago     3.75GB
```

Summary of new commands

Usage	Command
Pull an image	docker pull <image-name>
Pull a specific version of an image	docker pull <image-name>:<image-version>
List all local images	docker images
Remove an image	docker image rm <image-name>
Remove all stopped containers	docker container prune
Remove all images	docker image prune -a

Let's practice!

INTRODUCTION TO DOCKER

Distributing Docker Images

INTRODUCTION TO DOCKER

Tim Sangster
Software Engineer @ DataCamp

Private Docker registries

- Unlike Docker official images there is no quality guarantee
- Name starts with the url of the private registry

```
dockerhub.myprivateregistry.com/classify_spam
```

```
docker pull dockerhub.myprivateregistry.com/classify_spam:v1
```

```
Using tag: v1
latest: Pulling from dockerhub.myprivateregistry.com
ed02c6ade914: Pull complete
Digest: sha256:b6b83d3c331794420340093eb706b6f152d9c1fa51b262d9bf34594887c2c7ac
Status: Downloaded newer image for dockerhub.myprivateregistry.com/classify_spam:v1
dockerhub.myprivateregistry.com/classify_spam:v1
```

Pushing to a registry

```
docker image push <image name>
```

Pushing to a specific registry --> name of the image needs to start with the registry url

```
docker tag classify_spam:v1 dockerhub.myprivateregistry.com/classify_spam:v1
```

```
docker image push dockerhub.myprivateregistry.com/classify_spam:v1
```

Authenticating against a registry

- Docker official images --> No authentication needed
- Private Docker repository --> Owner can choose

```
docker login dockerhub.myprivateregistry.com
```

```
user@pc ~ % docker login dockerhub.myprivateregistry.com
Username: student
Password:
Login succeeded
```

Docker images as files

Sending a Docker image to one or a few people? Send it as a file!

Save an image

```
docker save -o image.tar classify_spam:v1
```

Load an image

```
docker load -i image.tar
```

Summary of new commands

Usage	Command
Pull image from private registry	docker pull <private-registry-url>/<image-name>
Name an image	docker tag <old-name> <new-name>
Push an image	docker image push <image-name>
Login to private registry	docker login <private-registry-url>
Save image to file	docker save -o <file-name> <image-name>
Load image from file	docker load -i <file-name>

Let's practice!

INTRODUCTION TO DOCKER