2.1 Run a static website in a container

The image that you are going to use is a single-page website that was already created for this demo and is available on the Docker Store as dockersamples/static-site. You can download and run the image directly in one go using docker run as follows.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker run -d dockersamples/static-site
Unable to find image 'dockersamples/static-site:latest' locally
latest: Pulling from dockersamples/static-site
fdd5d7827f33: Pull complete
a3ed95caeb02: Pull complete
716f7a5f3082: Pull complete
7b10f03a0309: Pull complete
aff3ab7e9c39: Pull complete
Digest: sha256:daa686c61d7d239b7977e72157997489db49f316b9b9af3909d9f10fd28b2dec
Status: Downloaded newer image for dockersamples/static-site:latest
be1800e69ab0e3a73e9d338c55950fdf76ebfdd344697ca8a37ce4c9e3825cab
```

Since we ran the container in detached mode, we don't have to launch another terminal to do this. Run docker ps to view the running containers.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker ps
CONTAINER ID   IMAGE                       COMMAND                  CREATED          STATUS                  PORTS
      NAMES
be1800e69ab0   dockersamples/static-site   "/bin/sh -c 'cd /usr…"   26 seconds ago   Up 25 seconds           80/tcp,
443/tcp      funny_mcclintock
927257842b95   alpine                      "/bin/sh"                3 minutes ago    Up 3 minutes
      nifty_hamilton
2496a73f07b7   portainer/portainer-ce      "/portainer"             9 months ago     Up Less than a second   0.0.0.0:
8000->8000/tcp, :::8000->8000/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp
      portainer
aad4d783513f   bitnami/mongodb:4.4         "/opt/bitnami/script…"   9 months ago     Up 21 minutes           0.0.0.0:
27017->27017/tcp, :::27017->27017/tcp
      desktop_mongodb_1
ad13952b9c23   nats:2.1.8-alpine3.11       "docker-entrypoint.s…"   9 months ago     Up 21 minutes           0.0.0.0:
4222->4222/tcp, :::4222->4222/tcp, 0.0.0.0:8222->8222/tcp, :::8222->8222/tcp, 6222/tcp   desktop_nats_1
```

Check out the CONTAINER ID column. You will need to use this CONTAINER ID value, a long sequence of characters, to identify the container you want to stop, and then to remove it. The example below provides the CONTAINER ID on our system; you should use the value that you see in your terminal.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker stop be1800e69ab0
be1800e69ab0
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker rm be1800e69ab0
be1800e69ab0
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS                  PORTS
      NAMES
927257842b95   alpine                   "/bin/sh"                9 minutes ago   Up 9 minutes
      nifty_hamilton
2496a73f07b7   portainer/portainer-ce   "/portainer"             9 months ago    Up Less than a second   0.0.0.0:8000
->8000/tcp, :::8000->8000/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp
      portainer
aad4d783513f   bitnami/mongodb:4.4      "/opt/bitnami/script…"   9 months ago    Up 27 minutes           0.0.0.0:2701
7->27017/tcp, :::27017->27017/tcp
      desktop_mongodb_1
ad13952b9c23   nats:2.1.8-alpine3.11    "docker-entrypoint.s…"   9 months ago    Up 27 minutes           0.0.0.0:4222
->4222/tcp, :::4222->4222/tcp, 0.0.0.0:8222->8222/tcp, :::8222->8222/tcp, 6222/tcp   desktop_nats_1
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

Now, let's launch a container in **detached** mode as shown below:

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker run --name static-site -e AUTHOR="Your Name" -d -P dockersamples
/static-site
12bf72ca8705fd0d32f241859e6fa9b8daa924ded6f18d47f5180c4b6ae5da94
```

Now you can see the ports by running the docker port command.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker port static-site
80/tcp -> 0.0.0.0:49154
80/tcp -> :::49154
443/tcp -> 0.0.0.0:49153
443/tcp -> :::49153
```

Hello Your Name!

This is being served from a **docker** container running Nginx.

You can also run a second webserver at the same time, specifying a custom host port mapping to the container's webserver.

But first, let's stop and remove the containers since you won't be using them anymore.

Let's use a shortcut to remove the second site:

docker ps

Run docker ps to make sure the containers are gone.

ANSWERS FOR ALL ABOVE COMMANDS

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker stop static-site
static-site
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker rm static-site
static-site
hater@hater:~/Desktop/se/assignment 2/Step 2$  docker rm -f static-site-2
static-site-2
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS                  PORTS
                                                                                                          NAMES
927257842b95   alpine                   "/bin/sh"                21 minutes ago   Up 21 minutes
                                                                                                          nifty_hamilton
2496a73f07b7   portainer/portainer-ce   "/portainer"             9 months ago     Up Less than a second   0.0.0.0:800
0->8000/tcp, :::8000->8000/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp                                 portainer
aad4d783513f   bitnami/mongodb:4.4      "/opt/bitnami/script…"   9 months ago     Up 39 minutes           0.0.0.0:270
17->27017/tcp, :::27017->27017/tcp                                                                        desktop_mongodb_1
ad13952b9c23   nats:2.1.8-alpine3.11    "docker-entrypoint.s…"   9 months ago     Up 39 minutes           0.0.0.0:422
2->4222/tcp, :::4222->4222/tcp, 0.0.0.0:8222->8222/tcp, :::8222->8222/tcp, 6222/tcp                       desktop_nats_1
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

2.2 Docker Images

Docker images are the basis of containers. In the previous example, you pulled the *dockersamples/static-site* image from the registry and asked the Docker client to run a container based on that image. To see the list of images that are available locally on your system, run the docker images command.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker images
REPOSITORY                      TAG                 IMAGE ID       CREATED         SIZE
alpine                          latest              0ac33e5f5afa   3 weeks ago     5.57MB
hello-world                     latest              feb5d9fea6a5   7 months ago    13.3kB
bitnami/mongodb                 4.4                 459fbddc53f3   9 months ago    427MB
portainer/portainer-ce          latest              865cf8021627   9 months ago    210MB
nats                            2.1.8-alpine3.11    96815e17bf11   20 months ago   16.3MB
devopsdockeruh/ports_exercise   latest              40f5014f7503   23 months ago   119MB
dockersamples/static-site       latest              f589ccde7957   6 years ago     191MB
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

For example you could pull a specific version of ubuntu image as follows:

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker pull ubuntu:12.04
12.04: Pulling from library/ubuntu
d8868e50ac4c: Pull complete
83251ac64627: Pull complete
589bba2f1b36: Pull complete
d62ecaceda39: Pull complete
6d93b41cfc6b: Pull complete
Digest: sha256:18305429afa14ea462f810146ba44d4363ae76e4c8dfc38288cf73aa07485005
Status: Downloaded newer image for ubuntu:12.04
docker.io/library/ubuntu:12.04
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

So for example, the docker pull command given below will pull an image named ubuntu:latest:

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
125a6e411906: Pull complete
Digest: sha256:26c68657ccce2cb0a31b330cb0be2b5e108d467f641c62e13ab40cbec258c68d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

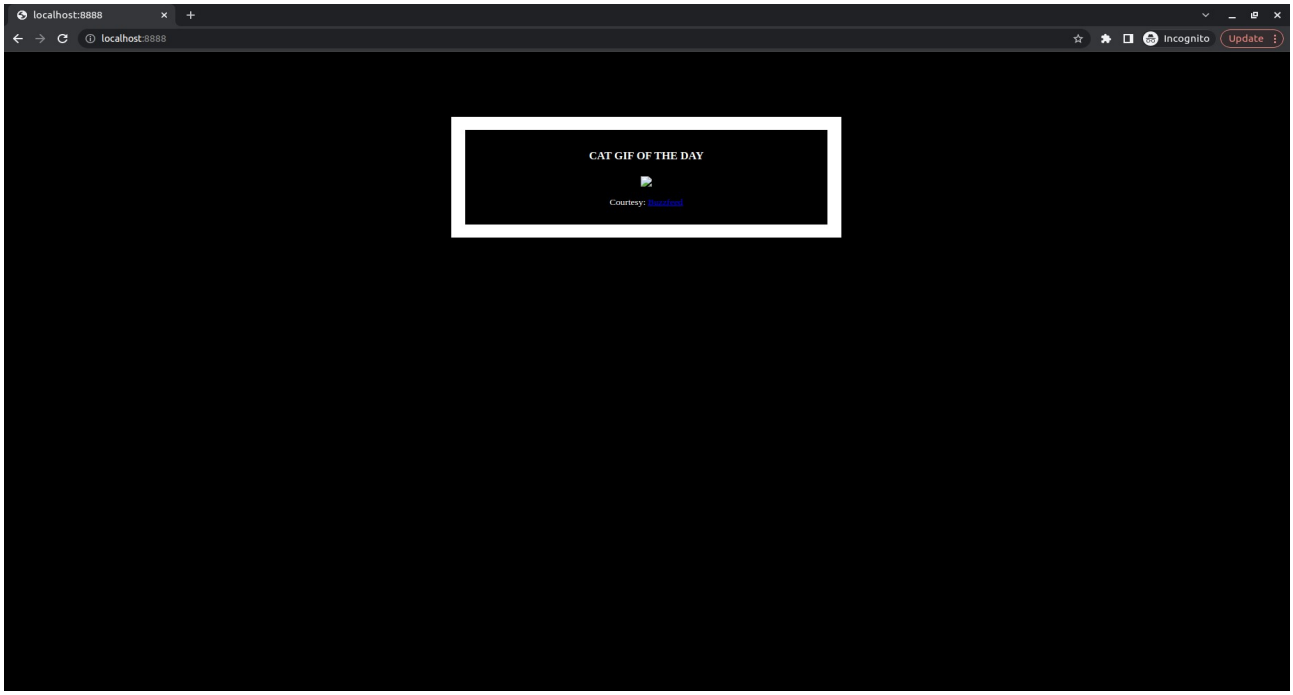
2.3 Create your first image

The docker build command is quite simple - it takes an optional tag name with the -t flag, and the location of the directory containing the Dockerfile - the . indicates the current directory:

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker build -t alisher2605/myfirstapp .
Sending build context to Docker daemon  8.192kB
Step 1/9 : FROM alpine:3.5
 ---> f80194ae2e0c
Step 2/9 : RUN apk add --update py2-pip
 ---> Using cache
 ---> 6bfbe3548d06
Step 3/9 : RUN pip install --upgrade pip
 ---> Using cache
 ---> 8ca2ae4940c4
Step 4/9 : COPY requirements.txt /usr/src/app/
 ---> Using cache
 ---> b145a8bb55d4
Step 5/9 : RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
 ---> Using cache
 ---> 735bbb6a439f
Step 6/9 : COPY app.py /usr/src/app/
 ---> Using cache
 ---> 1139c46d2289
Step 7/9 : COPY templates/index.html /usr/src/app/templates/
 ---> 811531c10fbb
Step 8/9 : EXPOSE 5000
 ---> Running in 958a741b6eb1
Removing intermediate container 958a741b6eb1
 ---> ff4f3abb72ae
Step 9/9 : CMD ["python", "/usr/src/app/app.py"]
 ---> Running in bddaaee02071
Removing intermediate container bddaaee02071
 ---> fac1bfa39be0
Successfully built fac1bfa39be0
Successfully tagged alisher2605/myfirstapp:latest
hater@hater:~/Desktop/se/assignment 2/Step 2$
```

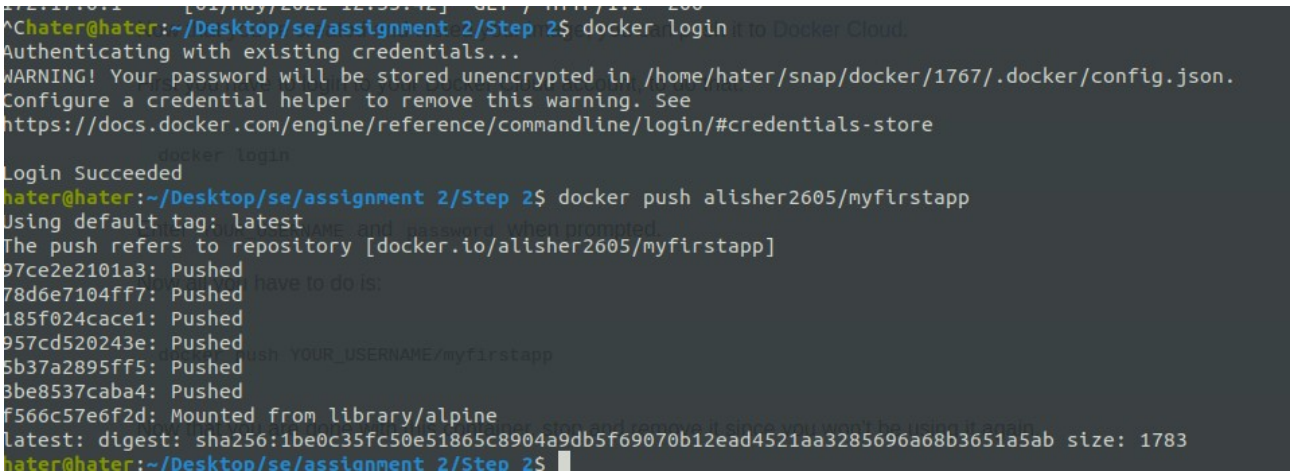
The next step in this section is to run the image and see if it actually works.

```
hater@hater:~/Desktop/se/assignment 2/Step 2$ docker run -p 8888:5000 --name myfirstapp alisher2605/myfirstapp
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [01/May/2022 12:53:32] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [01/May/2022 12:53:42] "GET / HTTP/1.1" 200 -
```



Now all you have to do is:

docker push YOUR_USERNAME/myfirstapp



Open another terminal window and execute the following commands: