# Docker Hub

In this lesson, you will learn how to publish a private image to Docker Hub.

Docker Hub is a Docker Registry offered by Docker Inc. It allows unlimited storage of public images, and paid plans to host your private images. A public image may be accessed by others, which is precisely what you want when you make your software widely available - less for internal enterprise software.

To publish images on Docker Hub, you need to create an account. I'm going to create one. For this, I head to https://hub.docker.com/ and click the *Sign Up* link. The *Docker ID* I select will be the prefix of images I publish to the Docker Hub.

When creating your account, make sure you select the right ID since it will be part of your images' names. Suppose your ID name is *short-name*, your images should be tagged:

```
<short_name>/<name>:<tag>
```

For this course, I created the *learnbook* account, and I want to publish the *webserver* image I created earlier. It's as simple as naming it correctly and pushing it to the Registry. I need to name it *learnbook/webserver*, and there are two ways to do this.

The first is to run the *docker build* command again using the correct name. This is a good option when you're using it from the start, but if I do this now it will result in two separate images on my disk, the same contents but two different names and IDs which is not a good idea.

My second option is much better; use the *docker tag* command. A Docker image can have several names as needed, and they can be added to an already existing image thanks to the *docker tag* command. The *docker tag* doesn't duplicate the image contrary to running *docker build* again.

The *docker tag* command accepts two arguments; first, the name of an existing image, and second, the name you want to add to that image.

```
docker tag webserver learnbook/webserver
```

I now have a single Docker image known by my machine under two names: *webserver* and *short_name/webserver*. When I run the *docker image ls* command, it will appear as two separate lines:

| REPOSITORY | TAG | IMAGE ID |
|---|---|---|
| short_name / webserver | latest | c067edac5ec1 |
| webserver | latest | c067edac5ec1 |

Note that the *image ID* is the same for both lines, which means there really is just one image on my machine, known under two different names.

Now that my image is named correctly, I can publish it to the Docker hub. It's just a matter of running two commands:

```
docker login
docker push learnbook/webserver
```

> The *docker login* command asks for my Docker Hub credentials interactively, but I could have just as well provided them as arguments to the command.

The *docker push* command is smart enough to push only the bits that differ from the base *nginx* image I used since it is already stored in the Docker Hub, as the output shows:

```
The push refers to repository [docker.io/learnbook/webserver]
The push refers to repository [docker.io/learnbook/webserver]
0f83d7865e1e: Pushed
6b5e2ed60418: Mounted from library/nginx
92c15149e23b: Mounted from library/nginx
0a07e81f5da3: Mounted from library/nginx
latest: digest: sha256:e8597...1ae0b1efb1 size: 1155
```

That’s it! The Docker Hub now hosts my image for everyone to see and run it.

> Everyone can see my image because it is public. In case I don’t want to share it with the whole world, I can set it as private.

---

In the next lesson, let’s run the image we just published.