

Docker Hands-on Guide - Docker Hub



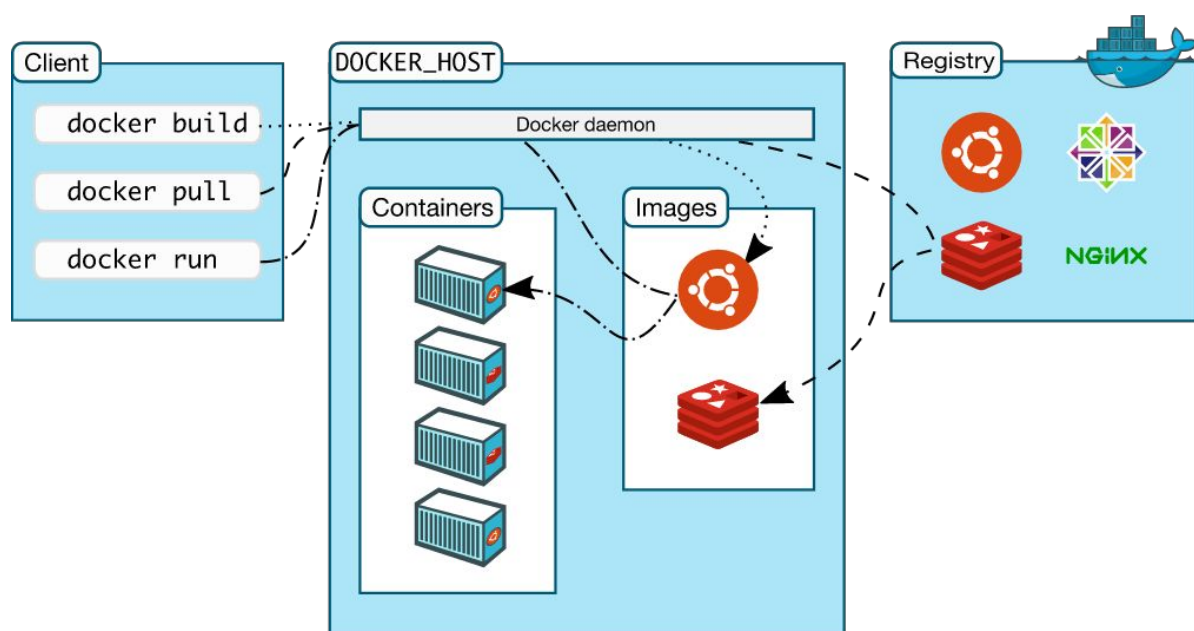
In this session, we shall look at working with the Docker registry. Some of this has already been covered, but it is nice to go through these commands again plus this session has a little bit more on the Docker Registry (Hub).

Recap

Recollect that the Docker toolset consists of:

- 1) Docker Daemon
- 2) Docker Client
- 3) Docker Hub

The Docker Architecture diagram is reproduced below again:



Let us focus on the **Registry** part in the diagram. The Registry is also called the Hub. So for the purpose of this document, the 2 terms are interchangeable.

Docker hosts public repositories called the Docker Registry (Hub) where you can find a list of public Docker images for your use. This is handy, since a lot of developers have worked hard to get the images ready for us and all we need to do is pull those images and start launching containers based on them.

The implications of this for the developer is huge. It cuts down your time drastically in downloading/setting up applications that you want to have running locally as fast as possible. For e.g. consider that you have to setup MySQL. One option available to you is to go down the traditional path and download the installer binary from the official MySQL site. Then follow the instructions to set things up and more. You know that this process is time consuming and error prone.

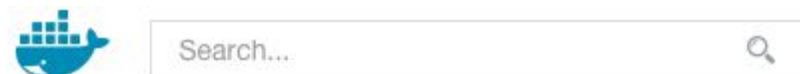
By now, you would have realised that a standard format for containers makes things not just easy and quick with a toolchain like Docker but you are also sure that it's going to run not just on your machine but if you need to give that image to someone else, you are confident of it running well there too.

So to summarize, do the following steps:

- 1) Visit the Docker Hub : <https://hub.docker.com>
- 2) Do sign up for the hub, this can help you push your own Docker images into the Hub too.
- 3) Check out the above Hub site. See the huge list of Docker images available for almost any software that you have used so far.

Searching for Docker Images





Assuming you are logged in with your Account (Search actually works without an account too!) -- try searching for any images via the Web Search box as shown below:



For e.g. type in Java, press <return> and see the list of repositories that are returned.

The screenshot below shows some of the repositories in the list:

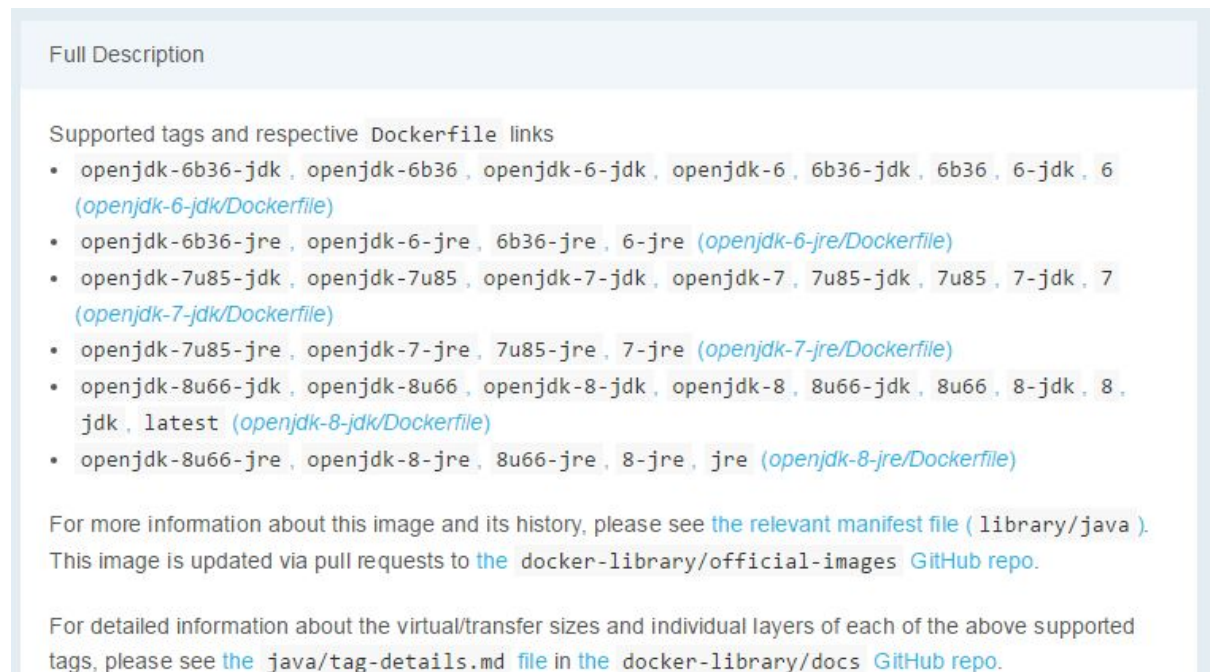
Repositories (2854)

All			
 java official	472 STARS	1.2 M PULLS	DETAILS
 nimmis/java public automated build	5 STARS	2.2 K PULLS	DETAILS
 1science/java public automated build	5 STARS	1.5 K PULLS	DETAILS
 andreluiznsilva/java public automated build	5 STARS	1.9 K PULLS	DETAILS

Notice the stars and the downloads. It gives you a good idea of its popularity.

Click on the **java official** repository and you will be lead to a page that will show various images (by their tags) and instructions for running a container based on that image and so on.

Shown below is a screenshot of the images and their tags:



Try It Out : In the image above, search for the **latest** tag. It will be there. Click that and it will lead you to a Dockerfile (more on that in another hands-on). Even if you do not understand all aspects of the Dockerfile, do take a look slowly and you will begin to see a pattern image. Hint : OS -> Software -> Any other dependencies , etc are slowly built one by one into your image.

Pulling an Image

Now that you are familiar with IMAGES and their TAGS, it should be clear to you that those are the only two pieces of information that you need to pull down an image locally to your step.

You can do that via the docker pull command as shown below:

```
$ docker pull java
```

or

```
$ docker pull java:latest
```

or

```
$ docker pull java:8
```

and so on.

Viewing a list of images

You can check out the list of images that you have locally via the docker images command:

```
$ docker images
```

Understand the output. It will have REPOSITORY and the TAG in the output result.

Launching a Container

Now that you have a local image, you can launch a container via :

```
$ docker run <imagename>:<tag>
```

Searching for Images

Earlier we saw how we could search for Docker images via the Docker Hub Web interface. We can do that via the docker command line too.

```
$ docker search <TERM>
```

For e.g.

```
$ docker search httpd
```

```
$ docker search java
```

```
$ docker search mysql
```