Tag, push, and pull your image

Estimated reading time: 6 minutes

In this section, you tag and push your `docker-whale` image to your new repository, then test the repository by pulling your new image.

Step 1: Tag and push the image

1. If you don’t already have a terminal open, open one now.

2. Run `docker images` to list the images stored locally:

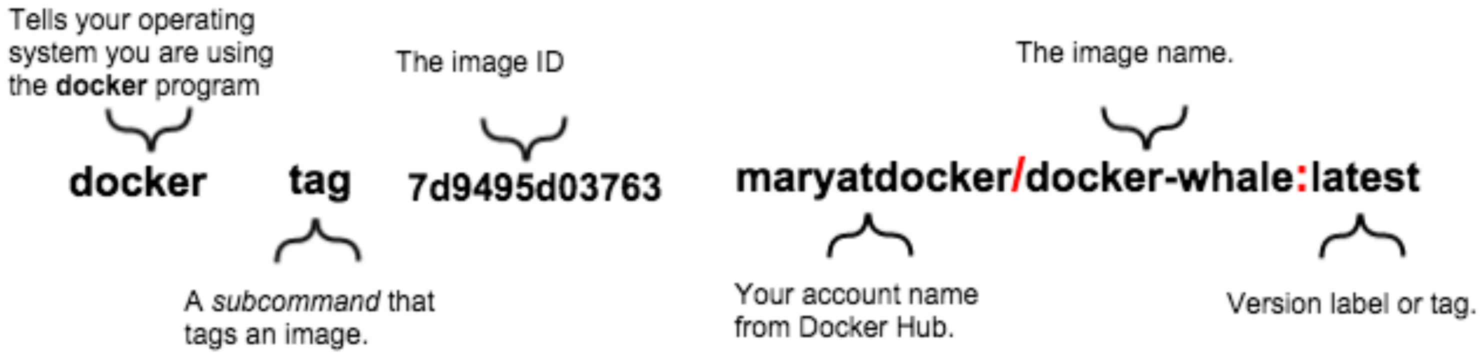
   ```
   $ docker images

   REPOSITORY           TAG          IMAGE ID            CREATED
   docker-whale         latest       7d9495d03763        38 minutes ago
   <none>               <none>       5dac217f722c        45 minutes ago
   docker/whalesay      latest       fb434121fc77        4 hours ago
   hello-world          latest       91c95931e552        5 weeks ago
   ```

3. Find the image ID for the `docker-whale` image, in the third column. In this example, the id is `7d9495d03763`, but yours will be different.

   > **Note**: Currently, the repository shows the repo name `docker-whale` with no namespace. You need to include the `namespace` for Docker Hub to associate it with your account. The `namespace` is the same as your Docker Hub account name. The next step adds the namespace to the image name, like `YOUR_DOCKERHUB_NAME/docker-whale`.

4. Tag the `docker-whale` image using the `docker tag` command and the image ID.

   The command you type looks like this:


Tells your operating system you are using the **docker** program
The image ID
The image name.
docker
tag
7d9495d03763
maryatdocker/docker-whale:latest
A *subcommand* that tags an image.
Your account name from Docker Hub.
Version label or tag.

Make sure to use your own Docker Hub account name.

```
$ docker tag 7d9495d03763 maryatdocker/docker-whale:latest
```

5. Run `docker images` again to verify that the `docker-whale` image has been tagged.

```
$ docker images

REPOSITORY                   TAG       IMAGE ID        CREATED
maryatdocker/docker-whale    latest    7d9495d03763    5 minutes ago
docker-whale                 latest    7d9495d03763    2 hours ago
<none>                       <none>    5dac217f722c    5 hours ago
docker/whalesay              latest    fb434121fc77    5 hours ago
hello-world                  latest    91c95931e552    5 weeks ago
```

The same image ID actually now exists in two different repositories.

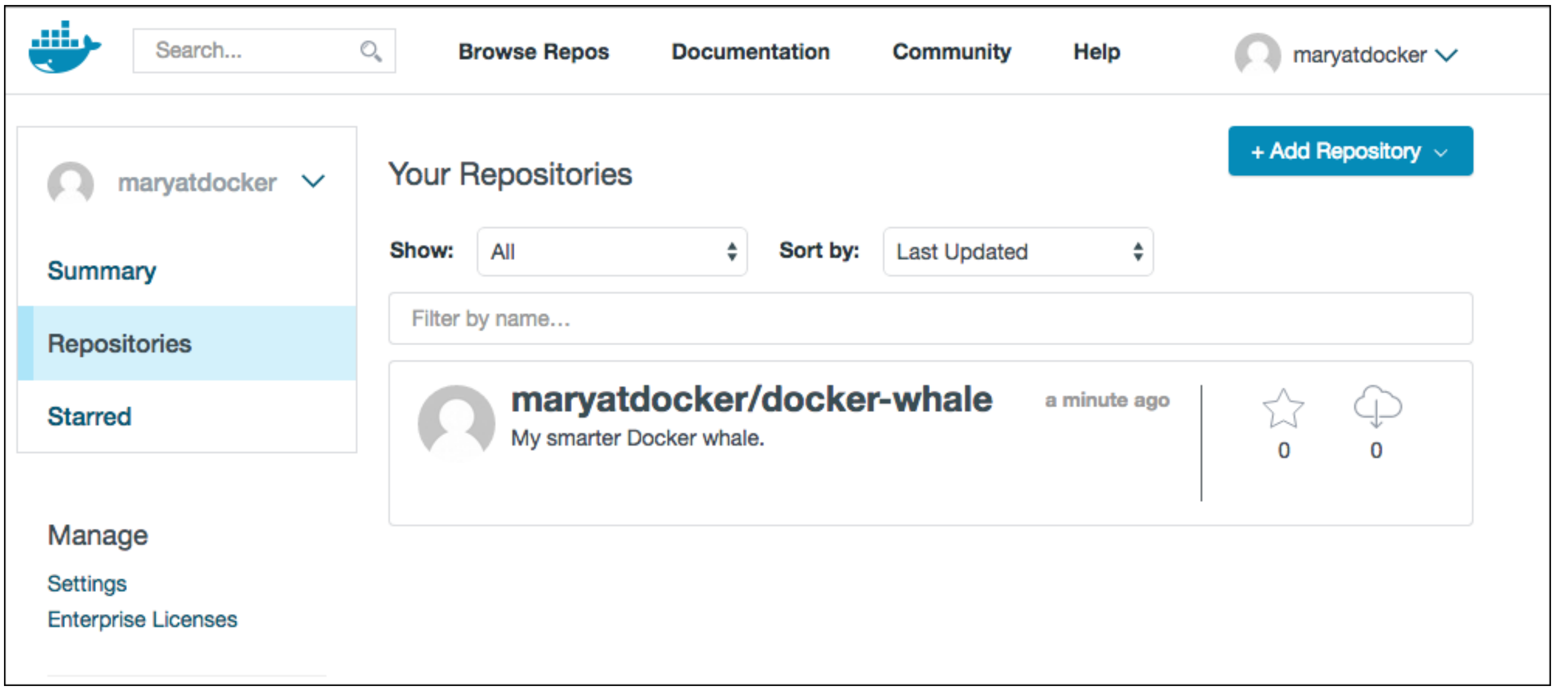
6. Before you can push the image to Docker Hub, you need to log in, using the `docker login` command. The command doesn't take any parameters, but prompts you for the username and password, as below:

```
$ docker login

    Username: *****
    Password: *****
    Login Succeeded
```

7. Push your tagged image to Docker Hub, using the `docker push` command. A lot of output is generated, as each layer is pushed separately. That output is truncated in the example below.

```
$ docker push maryatdocker/docker-whale

The push refers to a repository [maryatdocker/docker-whale] (len: 1)
7d9495d03763: Image already exists
...
e9e06b06e14c: Image successfully pushed
Digest: sha256:ad89e88beb7dc73bf55d456e2c600e0a39dd6c9500d7cd8d10256
```

8. Go back to the Docker Hub website to see the newly-pushed image.



Your Repositories

Show: All

Sort by: Last Updated

maryatdocker/docker-whale a minute ago

My smarter Docker whale.

Step 2: Pull your new image

The goal of pushing the image to Docker Hub is so that you can access it from any Docker host using `docker pull`. First, though, you need to remove the local copy. Otherwise, `docker pull` will not have any work to do, because it will see that you already have the latest version of the image locally.

1. If you don't already have a terminal open, open one now.

2. Use `docker images` to list the images you have locally.

```
$ docker images

REPOSITORY                    TAG       IMAGE ID        CREATED
maryatdocker/docker-whale     latest    7d9495d03763    5 minutes
docker-whale                  latest    7d9495d03763    2 hours ag
<none>                        <none>    5dac217f722c    5 hours ag
docker/whalesay               latest    fb434121fc77    5 hours ag
hello-world                   latest    91c95931e552    5 weeks ag
```

In the next step, you will remove both versions of the `docker-whale` image from your local system. They share the same ID. Make a note of it.

3. Use the `docker rmi` command to remove the images. You can refer to an image by its ID or its name. Since they share an ID, if you wanted to keep one of them, you'd need to refer to the other one by name. For this example, use the ID to remove both of them. Your ID will be different from the one below.

```
$ docker rmi -f 7d9495d03763
```

> **Tip**: You can also remove an image with `docker image rm -f` followed by image ID or name in a similar fashion.

4. When you use `docker run` it automatically downloads (pulls) images that don't yet exist locally, creates a container, and starts it. Use the following command to pull and run the `docker-whale` image, substituting your Docker Hub username.

```
$ docker run yourusername/docker-whale
```

Since the image is no longer available on your local system, Docker downloads it. The output below is truncated.

```
$ docker run maryatdocker/docker-whale

Unable to find image 'maryatdocker/docker-whale:latest' locally
latest: Pulling from maryatdocker/docker-whale
eb06e47a01d2: Pull complete
c81071adeeb5: Pull complete
        ...
fb434121fc77: Already exists
Digest: sha256:ad89e88beb7dc73bf55d456e2c600e0a39dd6c9500d7cd8d1
Status: Downloaded newer image for maryatdocker/docker-whale:lat
 ________________________________________
/ Having wandered helplessly into a      \
| blinding snowstorm Sam was greatly     |
| relieved to see a sturdy Saint Bernard |
| dog bounding toward him with the       |
| traditional keg of brandy strapped to  |
| his collar.                            |
|                                        |
| "At last," cried Sam, "man's best      |
\ friend -- and a great big dog, too!"   /
 ----------------------------------------
                \
                 \
                  \
                          ##        .
                    ## ## ##       ==
                 ## ## ## ##      ===
             /""""""""""""""""___/ ===
        ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
             \______ o          __/
              \    \        __/
                \____\______/
```

Next steps

After finishing this tutorial, you've done all of the following fundamental Docker tasks.

- You installed Docker.
- You ran a software image in a container.
- You located an interesting image on Docker Hub and ran it on your own machine.
- You modified an image to create your own, and created and ran a container based on that image.
- You created a Docker Hub account and repository.
- You pushed your custom image to Docker Hub, and made it available both for yourself and other Docker users.

Tweet #dockerdocs

You’ve only scratched the surface of what Docker can do. Learn more (https://docs.docker.com/engine/getstarted/last_page/) about where to go next.

Feedback? Suggestions? Can't find something in the docs?
Edit this page (https://github.com/docker/docker.github.io/edit/master/engine/getstarted/step_six.md) • Request docs changes (https://github.com/docker/docker.github.io/issues/new?assignee=mstanleyjones&body=File: [engine/getstarted/step_six.md](https://docs.docker.com/engine/getstarted/step_six/), CC @mstanleyjones) • Get support (https://www.docker.com/docker-support-services)
Rate this page:
153 3

What is Docker (https://www.docker.com/what-docker)

What is a Container (https://www.docker.com/what-container)

Use Cases (https://www.docker.com/use-cases)

Customers (https://www.docker.com/customers)

Partners (https://www.docker.com/partners/partner-program)

For Government (https://www.docker.com/industry-government)

About Docker (https://www.docker.com/company)

Management (https://www.docker.com/company/management)

Press & News (https://www.docker.com/company/news-and-press)

Careers (https://www.docker.com/careers)

Product (https://www.docker.com/products/overview)

Pricing (https://www.docker.com/pricing)

Community Edition (https://www.docker.com/docker-community)

Enterprise Edition (https://www.docker.com/enterprise)

Docker Datacenter (https://www.docker.com/products/docker-datacenter)

Docker Cloud (https://cloud.docker.com/)

Docker Store (https://store.docker.com/)

Docker for Mac (https://www.docker.com/docker-mac)

Docker for Windows (PC) (https://www.docker.com/docker-windows)

Docker for AWS (https://www.docker.com/docker-aws)

Docker for Azure (https://www.docker.com/docker-microsoft-azure)

Docker for Windows Server (https://www.docker.com/docker-windows-server)

Docker for CentOS distribution (https://www.docker.com/docker-centos)

Docker for Debian (https://www.docker.com/docker-debian)

Docker for Fedora® (https://www.docker.com/docker-fedora)

Docker for Oracle Enterprise Linux (https://www.docker.com/docker-oracle-linux)

Docker for RHEL (https://www.docker.com/docker-rhel)

Docker for SLES (https://www.docker.com/docker-sles)

Docker for Ubuntu (https://www.docker.com//docker-ubuntu)

Documentation (/)

Learn (https://www.docker.com/docker)

Blog (https://blog.docker.com)

Training (https://training.docker.com/)

Support (https://www.docker.com/docker-support-services)

Knowledge Base (https://success.docker.com/kbase)

Resources (https://www.docker.com/products/resources)

Community (https://www.docker.com/docker-community)

Open Source (https://www.docker.com/technologies/overview)

Events (https://www.docker.com/community/events)

Forums (https://forums.docker.com/)

Docker Captains (https://www.docker.com/community/docker-captains)

Scholarships (https://www.docker.com/docker-community/scholarships)

Community News (https://blog.docker.com/curated/)

Status (http://status.docker.com/) Security (https://www.docker.com/docker-security)

Legal (https://www.docker.com/legal) Contact (https://www.docker.com/company/contact)

(http (http (http (http (http (http (http (http