

Docker Registries

The primary function of a registry are

- Store image files
- Pulling images down
- Push images back up

A repository within a registry is a collection of **one or more images**. A single repo can hold many docker images, each stored as a **TAG**. The tag refers to a version of the image, but they also can refer different variations of an image.

```
docker pull docker.io/ubuntu:focal
docker pull registry.hub.docker.com/library/ubuntu:focal
# Same tag but different DNS name.

# If we check the images
docker images
>> ubuntu
>> registry.hub.docker.com/library/ubuntu
# Equal Image ID. They are the same image

# RegistryAddress/Repository/Image:Tag
```

Credentials

How would log in, if a repository is set to private. There are a number of popular container image registries such Amazon Elastic Container Registry, Quay from Red Hat and Google's Container Registry.

For business apps it's common to have your own registry. Docker offers the option to make repositories private, requiring you in to gain access to the registry. Examining the trade-off between running your own registry for complete control or using a vendor, will usually help business decide whether or not host their images.

To access a private registry use the `docker login` command. In order to create and access your own repos, create an account on Docker Hub.

Docker Hub

DockerHub is the default registry, and for this reason you don't have to specify a registry name with the docker login command. You can use this command to access images hosted on Docker Hub.

```
docker login OPTIONS SERVER -u username -p password
```