



Build an image from the Dockerfile in the current directory and tag the image

docker build -t myimage:0.1 .

List all images that are locally stored with Docker

docker images

Delete an image from the local image store

docker rmi myimage:0.1



Pull an image from a registry

docker pull myimage:0.1

Retag a local image

docker tag myimage:0.1 myrepo/myimage:1.0

Push an image to registry

docker push myrepo/myimage:1.0



Run a command in the container

docker exec -it myimage:0.1 whoami

Get's inside a container

docker exec -it myimage:0.1 bash

Save a running containers as an image

docker commit container name myimage:0.2



Run a container with name "web", expose port 8080 externally mapped to port 80 inside the container, from the Nginx version alpine image.

docker container run --name web -p 8080:80 nginx:alpine

Stop a running container through SIGTERM

docker container stop web

Stop a running container through SIGKILL

docker container kill web

List the running containers (add --all or -a to include stopped containers)

docker container ls

or

docker ps

Delete all running and stopped containers

docker container rm -f \$(docker ps -aq)

List the networks

docker network 1s

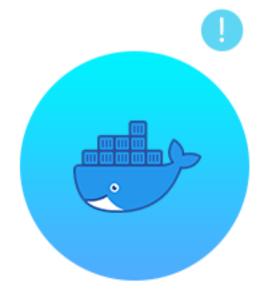




Image - a read-only layer that is the base of your container. Might have a parent image

Container - a runnable instance of the image

Registry/Hub - central place where images live

Docker Machine - a VM to run Docker (Linux does this natively)

Dockerfile - Document that executed in order to build a Docker image

Layer - a set of modification to the image, represented by an instruction in the Dockerfile

Docker Compose - a utility to run multiple containers as a system