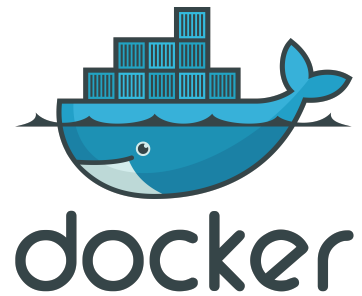


1 Resources

- **Github Repo:**
<https://bit.ly/3qJtZOX>
- **DockerHub Repo:**
<https://dockr.ly/344M828>
- **Command Cheatsheet:**
<https://bit.ly/3pK6lme>
- **DockerHub:**
<https://hub.docker.com>
- **Docker Install Instructions:**
<https://dockr.ly/3EW5of5>



2 Terms

- **Registry:** The storage location for images (think like Github), most popular is DockerHub, but private registries can be made
- **Image:** An immutable version of a “stack”, defined as the software stack required for the particular file.
- **Container:** The currently executing image, that has state added.
- **Dockerfile:** A file that describes how an image should be created.
- **Docker Compose File:** A .yaml file that describes how a collection of images should be invoked.

3 Common Commands

3.1 Executing Containers

- `docker run [container_name]` : Runs a container, non-daemon.
- `docker run -d [container_name]` : Runs a container, daemon.
- `docker-compose up <-d>` : Runs a compose collection, optional daemon.

3.2 Investigating Containers

- `docker ps` : Lists running containers.
- `docker logs [container_name_or_hash]` : Runs a container, non-daemon.

3.3 Killing/Cleanup of Containers

- `docker kill [container_name_or_hash]` : Kills a container.
- `docker stop [container_name_or_hash]` : Stops a container (graceful kill).
- `docker-compose down` : Stops a compose collection.

- `docker rm` : Removes a stopped container (kept around due to daemon).
- `docker container prune` : Removes stopped containers.

3.4 Working with Images

- `docker images` : Lists available images.
- `docker pull [image_name]` : Pulls an image from a repository.
- `docker push [image_name]` : Pushes an image.
- `docker rmi [image_name_or_hash]` : Removes an image locally.
- `docker image prune` : Removes dangling images (images without a tag).