

# Docker Cheat Sheet

## ORCHESTRATE

Initialize swarm mode and listen on a specific interface  
`docker swarm init --advertise-addr 10.1.0.2`

Join an existing swarm as a manager node  
`docker swarm join --token <manager-token> 10.1.0.2:2377`

Join an existing swarm as a worker node  
`docker swarm join --token <worker-token> 10.1.0.2:2377`

List the nodes participating in a swarm  
`docker node ls`

Create a service from an image exposed on a specific port and deploy 3 instances  
`docker service create --replicas 3 -p 80:80 --name web nginx`

List the services running in a swarm  
`docker service ls`

Scale a service  
`docker service scale web=5`

List the tasks of a service  
`docker service ps web`

## BUILD

Build an image from the Dockerfile in the current directory and tag the image  
`docker build -t myapp:1.0 .`

List all images that are locally stored with the Docker engine  
`docker images`

Delete an image from the local image store  
`docker rmi alpine:3.4`

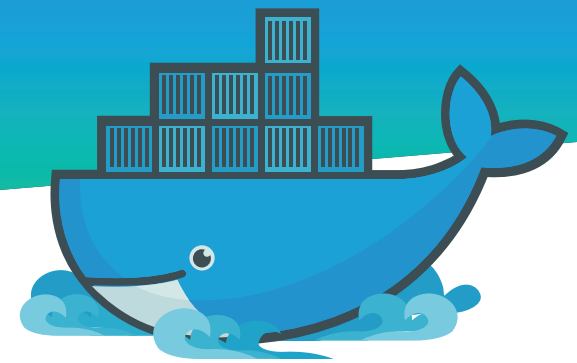
## SHIP

Pull an image from a registry  
`docker pull alpine:3.4`

Retag a local image with a new image name and tag  
`docker tag alpine:3.4 myrepo/myalpine:3.4`

Log in to a registry (the Docker Hub by default)  
`docker login my.registry.com:8000`

Push an image to a registry  
`docker push myrepo/myalpine:3.4`



## RUN

`docker run`  
--rm remove container automatically after it exits  
-it connect the container to terminal  
--name web name the container  
-p 5000:80 expose port 5000 externally and map to port 80  
-v ~/dev:/code create a host mapped volume inside the container  
alpine:3.4 the image from which the container is instantiated  
/bin/sh the command to run inside the container

Stop a running container through SIGTERM  
`docker stop web`

Stop a running container through SIGKILL  
`docker kill web`

Create an overlay network and specify a subnet  
`docker network create --subnet 10.1.0.0/24 --gateway 10.1.0.1 -d overlay mynet`

List the networks  
`docker network ls`

List the running containers  
`docker ps`

Delete all running and stopped containers  
`docker rm -f $(docker ps -aq)`

Create a new bash process inside the container and connect it to the terminal  
`docker exec -it web bash`

Print the last 100 lines of a container's logs  
`docker logs --tail 100 web`