Docker Cheatsheet v2.0 14-01-2020

```
docker image ls -a
docker image rm HASH_ID
docker image rm $(docker image ls -a -q)
docker image --help

docker tag IMAGE_ID username/repo:tag

# might be needed before next: docker login
docker push username:repository:tag
```

```
docker container ls [-a -q -s]
docker container start [HASH_ID | NAME]
docker container stop [HASH_ID | NAME]
docker container stop [HASH_ID | NAME]
docker container exec -it 8ead bash
docker container rm HASH_ID
docker container rm $(docker container ls -aq)

docker container --help
```

```
docker service ls
docker service ps srv1  #
list tasks in a service

docker inspect
task_or_container
```

container file system contents do persist when starting and stopping a container (with `**container start/stop ...**`, not with `**docker [container] run ...**` which creates a *new container* to run), but **this is not the recommended or safe way to persist data** with Docker!

```
docker build -t name:tag .  # uses ./Dockerfile
```

Dockerfile

Image 1

an **image** runs as a **container**

```
docker run -it -p 8080:80 \
    userx/foo:latest [--name funcky_container]
docker run -d -p 4000:80 \
    -v "pgdata:/var/lib/postgresql/data" \
    postgres
```

Container 1

Container 2

replicas of a **container** are configured to run as a **service**

docker-compose.yml

Service 1

Service 2

containers running inside a service are called tasks

Stack

Service 1

Task 10

Task 11

Service 2

Task 20

=

```
# this before stack commands
#    (since stack needs a swarm to run):
docker swarm init  # run on swarm manager node
```

```
docker stack deploy -c docker-compose.yml mystack
# (above also updates a running stack after config changes)
docker stack ls
docker stack ps mystack  # list tasks in stack
docker stack services mystack  # list services in stack
docker stack rm mystack
```

```
# this after stack commands:
docker swarm leave --force  # run on swarm manager node
```

a **stack** can be *deployed* on a **swarm**

Swarm

Cluster

Node 1

Machine 1

Node 2

Machine 2

```
# Create a VM (Mac, Win7, Linux):
docker-machine create --driver virtualbox myvm1
# Win10 (use "myswitch" for myvm2 too):
docker-machine create -d hyperv \
    --hyperv-virtual-switch "myswitch" myvm1
# init swarm (outputs command to be run on other
#    nodes to have them join the swarm):
docker-machine ssh myvm1 \
    "docker swarm init --advertise-addr <myvm1 ip>"
```

```
docker-machine ssh myvm1 "docker node ls"
docker-machine ssh myvm1 \
    "docker node inspect <node ID>"
# View join token:
docker-machine ssh myvm1 \
   "docker swarm join-token -q worker"
```

```
# set environment to run stack commands on manager node
#    (without the need for `docker-machine ssh myvm1 "..."`)
eval $(docker-machine env myvm1)
# reset docker-machine environment
eval $(docker-machine env -u)
```

```
docker-machine ls  #lists VMs (and their IPs)
docker-machine start myvm1
docker-machine stop myvm1

docker-machine scp ./local-file myvm1:~/data
```

DOCS:

https://docs.docker.com/
https://docs.docker.com/get-started/
https://docs.docker.com/engine/reference/builder/
https://docs.docker.com/storage/volumes/
https://docs.docker.com/compose/compose-file/
https://docs.docker.com/compose/
https://docs.docker.com/machine/