

DOCKER

Docker is an open source platform for building, deploying, and managing containerized applications.

DOCKER IMAGES

Download an image

```
docker pull nginx
```

Upload an image to a repository

```
docker push myimage:1.0
```

Delete an image

```
docker rmi nginx
```

Show a list of all Images

```
docker images
```

Delete dangling images

```
docker image prune
```

Build an image from a Dockerfile

```
docker build .
```

Tag an image

```
docker tag ubuntu ubuntu:18.04
```

Save an image

```
docker save nginx > nginx.tar
```

Load an image from a .tar file

```
docker load -i nginx.tar
```

RUN CONTAINERS

Start a new Container from an Image

```
docker run nginx
```

Assign it a name

```
docker run --name web nginx
```

Map a port

```
docker run -p 8080:80 nginx
```

Map all ports

```
docker run -P nginx
```

RUN CONTAINERS (cont)

Start container in background

```
docker run -d nginx
```

Assign it a hostname

```
docker run --hostname srv nginx
```

MANAGE CONTAINERS

Show a list of running containers

```
docker ps
```

Show a list of all containers

```
docker ps -a
```

Delete a container

```
docker rm web
```

Delete a running container

```
docker rm -f web
```

Delete stopped containers

```
docker container prune
```

Stop a running container

```
docker stop web
```

Start a stopped container

```
docker start web
```

Create an image out of container

```
docker commit web
```

NETWORK

List networks

```
docker network ls
```

Create a local network

```
docker network create mynet
```

Show Information on one or more networks

```
docker network inspect (network)
```

Connect a container to a network

```
docker network connect (network) (container)
```

NETWORK (cont)

```
docker network connect (network) (container)
```

```
docker network disconnect (network) (container)
```

Disconnect a container from a network

```
docker network disconnect (network) (container)
```

DOCKER COMPOSE

Start your docker-compose defined resources in detached mode

```
docker-compose up -d -f <docker-compose.yml>
```

Stop all docker-compose resources

```
docker-compose stop
```

Destroy all docker-compose resources

```
docker-compose down
```

ORCHESTRATE

Initialize swarm

```
Docker swarm init --advertise-addr IP
```

Join an existing swarm as manager/worker node

```
Docker swarm join --token<manager-token> IP
```

```
Docker swarm join --token<worker-token> IP
```

Create a service

```
Docker service create --replicas 3 -p 80:80 --name web nginx
```

DOCKER MACHINE

Create a machine

```
docker-machine create --driver VirtualBox NAME
```

List all the machines

```
docker-machine ls
```

Connect to a machine

```
docker-machine ssh NAME
```

VOLUMES

List volumes

```
docker volume ls
```

Create a volume

```
docker volume create <volume>
```

Delete a volume

```
docker volume rm <volume>
```

Mount a local directory to your container

```
docker run -v <local_dir>:/<container_dir> <image>
```

TROUBLESHOOTING

Show the logs of a container

```
docker logs <container>
```

Show a 'top' view of processes running on a container

```
docker top <container>
```

Show any files that have changed since startup

```
docker diff <container>
```

Connect to an already running container

```
docker attach <container>
```

Execute a command on a container

```
docker exec -it <container_id> /bin/bash
```

Show docker disk space used

```
docker system df
```