



# Cheatsheet for Docker CLI

## Run a new Container

Start a new Container from an Image

```
docker run IMAGE  
docker run nginx
```

...and assign it a name

```
docker run --name CONTAINER IMAGE  
docker run --name web nginx
```

...and map a port

```
docker run -p HOSTPORT:CONTAINERPORT IMAGE  
docker run -p 8080:80 nginx
```

...and map all ports

```
docker run -P IMAGE  
docker run -P nginx
```

...and start container in background

```
docker run -d IMAGE  
docker run -d nginx
```

...and assign it a hostname

```
docker run --hostname HOSTNAME IMAGE  
docker run --hostname srv nginx
```

...and add a dns entry

```
docker run --add-host HOSTNAME:IP IMAGE
```

...and map a local directory into the container

```
docker run -v HOSTDIR:TARGETDIR IMAGE  
docker run -v ~/.:/usr/share/nginx/html nginx
```

...but change the entrypoint

```
docker run -it --entrypoint EXECUTABLE IMAGE  
docker run -it --entrypoint bash 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 CONTAINER  
docker rm web
```

Delete a running container

```
docker rm -f CONTAINER  
docker rm -f web
```

Delete stopped containers

```
docker container prune
```

Stop a running container

```
docker stop CONTAINER  
docker stop web
```

Start a stopped container

```
docker start CONTAINER  
docker start web
```

Copy a file from a container to the host

```
docker cp CONTAINER:SOURCE TARGET  
docker cp web:/index.html index.html
```

Copy a file from the host to a container

```
docker cp TARGET CONTAINER:SOURCE  
docker cp index.html web:/index.html
```

Start a shell inside a running container

```
docker exec -it CONTAINER EXECUTABLE  
docker exec -it web bash
```

Rename a container

```
docker rename OLD_NAME NEW_NAME  
docker rename 096 web
```

Create an image out of container

```
docker commit CONTAINER  
docker commit web
```

## Manage Images

Download an image

```
docker pull IMAGE[:TAG]  
docker pull nginx
```

Upload an image to a repository

```
docker push IMAGE  
docker push myimage:1.0
```

Delete an image

```
docker rmi IMAGE
```

Show a list of all Images

```
docker images
```

Delete dangling images

```
docker image prune
```

Delete all unused images

```
docker image prune -a
```

Build an image from a Dockerfile

```
docker build DIRECTORY  
docker build .
```

Tag an image

```
docker tag IMAGE NEWIMAGE  
docker tag ubuntu ubuntu:18.04
```

Build and tag an image from a Dockerfile

```
docker build -t IMAGE DIRECTORY  
docker build -t myimage .
```

Save an image to .tar file

```
docker save IMAGE > FILE  
docker save nginx > nginx.tar
```

Load an image from a .tar file

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

## Info & Stats

Show the logs of a container

```
docker logs CONTAINER  
docker logs web
```

Show stats of running containers

```
docker stats
```

Show processes of container

```
docker top CONTAINER  
docker top web
```

Show installed docker version

```
docker version
```

Get detailed info about an object

```
docker inspect NAME  
docker inspect nginx
```

Show all modified files in container

```
docker diff CONTAINER  
docker diff web
```

Show mapped ports of a container

```
docker port CONTAINER  
docker port web
```



## definitions



image

a static snapshot of container's configuration.



container

an application sandbox.  
each container is based on an image.



layer

image is composed of read-only file system layers.  
container creates single writable layer.



docker registry

remote server for storing Docker images



Dockerfile

a configuration file with build instructions for Docker images



docker engine

Docker platform installation running on a given host



docker client

client application that talks to local or remote Docker daemon



docker daemon

service process that listens to Docker client commands over local or remote network



docker host

server that runs Docker engine



volume

directory shared between host and container

## docker run

**docker run** [OPTIONS] IMAGE[:TAG] [COMMAND]

Run a command in a new container.

metadata

**--name=CNTNR\_NAME** Assign a name to the container.

**-l, --label NAME[=VALUE]**  
Set metadata on the container.

process

**-d, --detach** Run in the background.

**-i, --interactive** Keep STDIN open.

**-t, --tty** Allocate a pseudo-TTY.

**--rm** Automatically remove the container when process exits.

**-u USER** Run as username or UID.

**--privileged** Give extended privileges.

**-w DIR** Set working directory.

**-e NAME=VALUE** Set environment variable.

**--restart=POLICY** Restart policy.  
**no on-failure[:RETRIES]**  
**always unless-stopped**

network

**-P, --publish-all** Publish all exposed ports to random ports.

**-p HOST\_PORT:CNTNR\_PORT**  
Expose a port or a range of ports.

**--network=NETWORK\_NAME**  
Connect container to a network.

**--dns=DNS\_SERVER1[,DNS\_SERVER2]**  
Set custom dns servers.

**--add-host=HOSTNAME:IP**  
Add a line to /etc/hosts.

file system

**--read-only** Mount the container's root file system as read only.

**-v, --volume [HOST\_SRC:]CNTNR\_DEST**  
Mount a volume between host and the container file system.

**--volumes-from=CNTNR\_ID**  
Mount all volumes from another container.

## Dockerfile

**FROM** <image\_id>

base image to build this image from

**RUN** <command> **shell form**

**RUN** ["<executable>",  
"<param1>",  
...,  
"<paramN>"]

**exec form**  
executes command to modify container's file system state

**MAINTAINER** <name>

provides information about image creator

**LABEL** <key>=<value>

adds searchable **metadata** to image

**ARG** <name>[=<default value>]

defines overridable **build-time parameter**:

docker build --build-arg <name>=<value> .

**ENV** <key>=<value>

defines **environment variable** that will be visible during image build-time and container run-time

**ADD** <src> <dest>

copies files from <src> (**file, directory or URL**) and adds them to container file system under <dst> path

**COPY** <src> <dest> similar to **ADD**, does not support URLs

**VOLUME** <dest>

defines **mount point** to be shared with host or other containers

**EXPOSE** <port>

informs Docker engine that container listens to **port** at run-time

**WORKDIR** <dest>

sets build-time and run-time working directory

**USER** <user>

defines run-time user to start container process

**STOPSIGNAL** <signal>

defines signal to use to notify container process to stop

**ENTRYPOINT** **shell form** or **exec form**

defines run-time **command prefix** that will be added to all run commands executed by docker run

**CMD** **shell form** or **exec form**

defines run-time **command** to be executed by default when docker run command is executed

