# docker CHEATSHEET

## 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 wih 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 containters*

**`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*

extreme automation