

Docker CLI

container

- `docker ps` shows all running container
 - `docker ps -a` shows all existing container
- `docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]`
 - `--name <container-name>` gives the created container a custom name
 - `--rm` removes the container after stopping
 - `-d` detached mode—stops the container after main process exits usually not desired
 - `-it` for interactive processes - keep STDIN open (`i`), allocate a pseudo-tty (`t`)
- typical example for testing: `docker run -it --rm --name test php:7.4 bash`
- use bash in running container: `docker exec -it containerName bash`

network

- list networks `docker network ls`
- add a network connection `docker network connect [OPTIONS] NETWORK CONTAINER`

logs

```
docker logs --follow <container ID>
```

- `--follow` streams logs until command is killed (`^c`)
- `--until=3s` streams logs for 3 seconds
- `--since 2019-03-02` show logs since specific date

build dockerfile

```
docker build .
```

- arguments
 - `.` build context (configs, mounts, ...)
 - `-f dockerfile` specify a custom dockerfile (independent from the build context)
 - `-t vendor/image-name:version` specify a name (and a version) for the build image
 - `--no-cache` build image without using caches
 - `--build-arg var_name=test` set ARG

- most used: `docker build -t draab/image-name:latest .`

best practice

- use **user** flag: `--user $(id -u):$(id -g)` to avoid problems on creating and accessing files
-

Created 2022-10-27 09:23:50 UTC by Daniel Raab

Updated 2023-01-30 09:49:49 UTC by Daniel Raab