

# docker-run

Start a new container from an image	<code>docker run [OPTIONS] image</code>
Delete container upon completion	<code>--rm</code>
Add a name for the container	<code>--name demo</code>
Explicit port mapping	<code>-p 8080:8081</code>
Map all ports	<code>-P</code>
Start container in background (detached)	<code>-d</code>
Adds DNS entry	<code>--add-hosts hostname:ip</code>
Maps a local directory into the container	<code>-v ./demo:/usr/share/demo</code>

```
docker run --rm --name demo nginx
```

shell

## Docker Interactions

Objective	Command	Sample
List (all) containers	<code>ps</code>	<code>docker ps -a</code>
Start the shell in a running container	<code>exec</code>	<code>docker exec -it nginx /bin/sh</code>
Output of information about the container	<code>inspect</code>	<code>docker inspect nginx</code>
Shows logs of the container	<code>logs</code>	<code>docker logs --follow --tail 2 demo</code>
Shows container processes	<code>top</code>	<code>docker top demo</code>
Shows events related to the containers	<code>events</code>	<code>docker events</code>
Download an image	<code>pull</code>	<code>docker pull ubuntu:latest</code>
Upload an image to the repository	<code>push</code>	<code>docker push gcr.io/trion/demo</code>
Save an image to an archive	<code>save</code>	<code>docker save demo &gt; demo.tgz</code>
Load an image from an archive	<code>load</code>	<code>docker load &lt; demo.tgz</code>
Delete a container	<code>rm</code>	<code>docker rm demo</code>
Delete an image	<code>rmi</code>	<code>docker rmi ubuntu</code>
Rename an image	<code>tag</code>	<code>docker tag &lt;source&gt; localhost/demo</code>
Shows how many resources are still free	<code>system df</code>	<code>docker system df</code>
Deletes all untagged containers, networks and images (or all unused ones with <code>-a</code> and also volumes with <code>--volumes</code> )	<code>system prune</code>	<code>docker system prune -a --volumes</code>

# Dockerfile

Description	Instruction (Sample)
Base image, first statement inside Dockerfile	FROM ubuntu:latest
Executes a command and saves the result	RUN ls -l
Adds metadata labels to the image	LABEL maintainer="Thomas Kruse <tk@trion.de>"
Defines default command	CMD
Defines the entry point	ENTRYPOINT
Announces that a port is needed	EXPOSE 3000
Creates an environment variable <code>Key</code> with value <code>Value</code>	ENV Key=Value
Copies files/folders from the host system ( <code>demo</code> and <code>foo</code> ) to the container (to the <code>bar</code> directory)	COPY ./demo ./foo /bar
Sets the user ID for all following commands	USER 1001
Defines path for volume mount in container	VOLUME /path/to/demo
Defines current working directory for all subsequent commands	WORKDIR

Dockerfile

```
FROM debian:11-slim

WORKDIR /app
COPY spring-demo /app/

RUN mvn clean package
```

Description	Image
Ubuntu Linux	ubuntu
Ubuntu 21.10	ubuntu:21.10
Debian 11	debian:11
Slimmed down Debian 11	debian:11-slim
OpenJDK Java 11	openjdk:11
OpenJDK Java 11 (slim)	openjdk:11-alpine
OpenJDK Java 11 (Alpine)	openjdk:11-slim
Traefik Proxy Version 2.5	traefik:v2.5
busybox	busybox
Angular CLI	trion/ng-cli

# docker-compose

Description	Instruction (Sample)	Type
Contains all services (key: service name)	<code>services</code>	Map
Container image used for the service	<code>image: trion/ng-cli:13.0.1</code>	String
Overwrites CMD from Dockerfile	<code>command</code>	String
Specifies ports for port mappings. Format: <code>Host-Port:Container-Port</code>	<code>ports:</code> <code>- 9090:8080</code>	Array
Specifies volumes for volume mounts. Format: <code>Host-Pfad:Container-Pfad</code>	<code>volumes:</code> <code>- ./foo:/bar</code>	Array
Network mode ( <code>host</code> , <code>bridge</code> , <code>service-name</code> , <code>none</code> )	<code>network_mode</code>	String
Specifies environment variables as key-value	<code>environment</code>	Map

Escape the \$-sign with \$\$

```
version: '3'

services:
  redis:
    image: redis:5-alpine
    command: redis-server --notify-keyspace-events KEA
    ports:
      - 6379:6379

  client-dev:
    image: trion/ng-cli:13.0.1
    command: /bin/sh -c "npm ci; ng serve --host 0.0.0.0"
    network_mode: host
    volumes:
      - ../../client:/app

  mariadb:
    image: mariadb:10.3
    environment:
      MYSQL_USER: "foo"
      MYSQL_PASSWORD: "bar"
      MYSQL_DATABASE: "demo"
    ports:
      - 3306:3306
    volumes:
      - ./initdb.d:/docker-entrypoint-initdb.d
```

docker-compose.yml