

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
```

Docker Cheat Sheet



dockerlux.github.io

@gcuisinier

General Usage

Start a container in background

```
$> docker run -d jenkins
```

Start an interactive container

```
$> docker run -it ubuntu bash
```

Start a container automatically removed on stop

```
$> docker run --rm ubuntu bash
```

Export port from a container

```
$> docker run -p 80:80 -d nginx
```

Start a named container

```
$> docker run --name mydb redis
```

Restart a stopped container

```
$> docker start mydb
```

Stop a container

```
$> docker stop mydb
```

Add metadata to container

```
$> docker run -d \
label=traefik.backend=jenkins jenkins
```

Build Images

Build an image from Dockerfile in current directory

```
$> docker build --tag myimage .
```

Force rebuild of Docker image

```
$> docker build --no-cache .
```

Convert a container to image

```
$> docker commit c7337 myimage
```

Remove all unused images

```
$> docker rmi $(docker images \
-q -f "dangling=true")
```

Debug

Run another process in running container

```
$> docker exec -it c7337 bash
```

Show live logs of running daemon container

```
$> docker logs -f c7337
```

Show exposed ports of a container

```
$> docker port c7337
```

Volumes

Create a local volume

```
$> docker volume create --name myvol
```

Mounting a volume on container start

```
$> docker run -v myvol:/data redis
```

Destroy a volume

```
$> docker volume rm myvol
```

List volumes

```
$> docker volume ls
```

Create a local network

```
$> docker network create mynet
```

Attach a container to a network on start

```
$> docker run -d --net mynet redis
```

Connect a running container from a network

```
$> docker network connect mynet c7337
```

Disconnect container to a network

```
$> docker network disconnect mynet c7337
```

Manage Containers

List running containers

```
$> docker ps
```

List all containers (running & stopped)

```
$> docker ps -a
```

Inspect containers metadatas

```
$> docker inspect c7337
```

List local available images

```
$> docker images
```

Delete all stopped containers

```
$> docker rm $(docker ps --filter status=exited -q)
```

List all containers with a specific label

```
$> docker ps --filter label=traefik.backend
```

Query a specific metadata of a running container

```
$> docker inspect -f '{{.NetworkSettings.IPAddress}}' c7337
```

Legend

Image name

```
redis, jenkins, nginx
```

Container name or commit ID

```
mydb #name
```

```
c7337 #commit id
```