



Glossary

Build	build is the process of building Docker images using a Dockerfile.
Container	a container is a runtime instance of a Docker image.
Dockerfile	a Dockerfile is a text document that contains all the commands you would normally execute manually in order to build a Docker image. Docker can build images automatically by reading the instructions from a Dockerfile.
Docker Daemon	the background service running on the host that manages building, running and distributing Docker containers. The daemon is the process that runs in the operating system which clients talk to.
Image	a Docker image is a read-only template that contains a set of instructions for creating a container.
Stack	a group of containers that runs together in order to create a service. Example: [wordpress] + [mysql]
Volume	Docker volumes are file systems mounted into a Docker containers to preserve data generated by the running container.

Author this cheat sheet has been created by TheFax

Run containers

docker run	parameter	description
	-d	Run in background / detached mode
	-it	Interactive, tty
	--rm	Remove container after it stops.
	-p 83:80	Port mapping host:container
	-v /dirHost:/dir	Mount file or directories /host/dir:/container/dir
	--name pluto	Set container name
	--restart no	Restart policy: no : stops the container when it exit. on-failure[:max-retries] : Restart only if the container exits with a non-zero exit status. Optionally, limit the number of restart retries. always : restart the container regardless of the exit status. The container will also always start on daemon startup. unless-stopped : Always restart the container regardless of the exit status, except if the container was put into a stopped state.
	IMAGE	Image name is the LAST parameter.

Manage containers

docker start/stop CONTAINER	Start or stop a container
docker pause/unpause CONTAINER	Pause or unpause a container



Manage containers (cont)

<code>docker restart CONTAINER</code>	Stop and restart a container
<code>docker attach CONTAINER</code>	Connect terminal to a running container
<code>docker rename OLD_NAME NEW_NAME</code>	Rename a container
<code>docker container rm CONTAINER</code> <code>docker rm CONTAINER</code>	Remove the specified container
<code>docker container rm -f CONTAINER</code> <code>docker rm -f CONTAINER</code>	Force the removing of specified container.
<code>docker container prune</code>	Delete all stopped containers

Manage images

<code>docker rmi IMAGE</code> , or <code>docker image rm IMAGE</code>	Remove image
<code>docker build URL</code>	Create an image from a dockerfile
<code>docker commit CONTAINER NEW_IMAGE_NAME</code>	Create an image from a container
<code>docker search TERM</code>	Search the Docker Hub for images

Information

<code>docker stats</code>	Live data about running containers
<code>docker ps</code>	List running containers
<code>docker ps -a</code>	List ALL containers, also stopped ones
<code>docker port CONTAINER</code>	List port mapping of specified container
<code>docker image ls</code> , or <code>docker images</code>	List all locally stored images
<code>docker container ls</code>	List running containers
<code>docker container ls -a</code>	List all containers
<code>docker volume ls</code>	List volumes

Tips :-)

Where are docker images and files?	<code>/var/lib/docker</code>
How can I refer to a container?	Via its name (set by <code>--name Pluto</code>) or... Via its ID (example: <code>4a7f7eebae0f63178aff7eb0aa39f0627a203ab2df258c1a00b456cf20063</code>) or... Via the first letters of its ID (example: <code>4a7f</code>)
How can I test my docker?	Try one of these commands: <code>docker run hello-world</code> <code>docker run -it --rm -d --name test -p 83:80 nginx</code>



