



Docker cheatsheet

This is a quick reference cheat sheet for Docker. And you can find the most common Docker commands here.

Getting Started

Getting started

Create and run a container in background

```
$ docker run -d -p 80:80 docker/getting-started
```

-d - Run the container in detached mode

-p 80:80 - Map port 80 to port 80 in the container

docker/getting-started - The image to use

Create and run a container in foreground

```
$ docker run -it -p 8001:8080 --name my-nginx nginx
```

-it - Interactive bash mode

-p 8001:8080 - Map port 8001 to port 8080 in the container

--name my-nginx - Specify a name

nginx - The image to use

General commands

```
docker ps
```

List running containers

```
docker ps -a
```

List all containers

<code>docker ps -s</code>	List running containers (with CPU / memory)
<code>docker images</code>	List all images
<code>docker exec -it <container> bash</code>	Connecting to container
<code>docker logs <container></code>	Shows container's console log
<code>docker stop <container></code>	Stop a container
<code>docker restart <container></code>	Restart a container
<code>docker rm <container></code>	Remove a container
<code>docker port <container></code>	Shows container's port mapping
<code>docker top <container></code>	List processes
<code>docker kill <container></code>	Kill a container

Docker Containers

Starting & Stopping

<code>docker start my-nginx</code>	Starting
<code>docker stop my-nginx</code>	Stopping
<code>docker restart my-nginx</code>	Restarting
<code>docker pause my-nginx</code>	Pausing
<code>docker unpause my-nginx</code>	Unpausing
<code>docker wait my-nginx</code>	Blocking a Container
<code>docker kill my-nginx</code>	Sending a SIGKILL
<code>docker attach my-nginx</code>	Connecting to an Existing Container

Information

<code>docker ps</code>	List running containers
<code>docker ps -a</code>	List all containers
<code>docker logs my-nginx</code>	Container Logs
<code>docker inspect my-nginx</code>	Inspecting Containers

<code>docker events my-nginx</code>	Containers Events
<code>docker port my-nginx</code>	Public Ports
<code>docker top my-nginx</code>	Running Processes
<code>docker stats my-nginx</code>	Container Resource Usage
<code>docker diff my-nginx</code>	Lists the changes made to a container.

Creating

```
docker create [options] IMAGE
-a, --attach           # attach stdout/err
-i, --interactive     # attach stdin (interactive)
-t, --tty             # pseudo-tty
    --name NAME       # name your image
-p, --publish 5000:5000 # port map (host:container)
    --expose 5432      # expose a port to containers
-P, --publish-all    # publish all ports
    --link container:alias # linking
-v, --volume `pwd`: /app # mount (absolute paths needed)
-e, --env NAME=hello    # env vars
```

Example

```
$ docker create --name my_redis --expose 6379 redis:3.0.2
```

Manipulating

Renaming a Container

```
docker rename my-nginx my-nginx
```

Removing a Container

```
docker rm my-nginx
```

Updating a Container

```
docker update --cpu-shares 512 -m 300M my-nginx
```

Docker Images

Manipulating

<code>docker images</code>	Listing images
<code>docker rmi nginx</code>	Removing an image
<code>docker load < ubuntu.tar.gz</code>	Loading a tarred repository
<code>docker load --input ubuntu.tar</code>	Loading a tarred repository
<code>docker save busybox > ubuntu.tar</code>	Save an image to a tar archive
<code>docker history</code>	Showing the history of an image
<code>docker commit nginx</code>	Save a container as an image.
<code>docker tag nginx eon01/nginx</code>	Tagging an image
<code>docker push eon01/nginx</code>	Pushing an image

Building Images

```
$ docker build .
$ docker build github.com/creack/docker-firefox
$ docker build - < Dockerfile
$ docker build - < context.tar.gz
$ docker build -t eon/my-nginx .
$ docker build -f myOtherDockerfile .
$ curl example.com/remote/Dockerfile | docker build -f - .
```

Docker Networking

Manipulating

Removing a network

```
docker network rm MyOverlayNetwork
```

Listing networks

```
docker network ls
```

Getting information about a network

```
docker network inspect MyOverlayNetwork
```

Connecting a running container to a network

```
docker network connect MyOverlayNetwork nginx
```

Connecting a container to a network when it starts

```
docker run -it -d --network=MyOverlayNetwork nginx
```

Disconnecting a container from a network

Creating Networks

```
docker network create -d overlay MyOverlayNetwork
```

```
docker network create -d bridge MyBridgeNetwork
```

```
docker network create -d overlay \
  --subnet=192.168.0.0/16 \
  --subnet=192.170.0.0/16 \
  --gateway=192.168.0.100 \
  --gateway=192.170.0.100 \
  --ip-range=192.168.1.0/24 \
  --aux-address="my-router=192.168.1.5" \
  --aux-address="my-switch=192.168.1.6" \
  --aux-address="my-printer=192.170.1.5" \
  --aux-address="my-nas=192.170.1.6" \
  MyOverlayNetwork
```

Clean Up

Clean All

Cleans up dangling images, containers, volumes, and networks (ie, not associated with a container)

```
docker system prune
```

Additionally, remove any stopped containers and all unused images (not just dangling images)

```
docker system prune -a
```

Containers

Stop all running containers

```
docker stop $(docker ps -a -q)
```

Delete stopped containers

```
docker container prune
```

Images

Remove all dangling (not tagged and is not associated with a container) images:

```
docker image prune
```

Remove all images which are not used by existing containers

```
docker image prune -a
```

Volumes

```
docker volume prune
```

Remove all volumes not used by at least one container

Miscellaneous

Docker Hub

```
docker search search_word
```

Search docker hub for images.

```
docker pull user/image
```

Downloads an image from docker hub.

```
docker login
```

Authenticate to docker hub

```
docker push user/image
```

Uploads an image to docker hub.

Registry commands

Login to a Registry

```
$ docker login
```

```
$ docker login localhost:8080
```

Logout from a Registry

```
$ docker logout
```

```
$ docker logout localhost:8080
```

Searching an Image

```
$ docker search nginx
$ docker search nginx --stars=3 --no-trunc busybox
```

Pulling an Image

```
$ docker pull nginx
$ docker pull eon01/nginx localhost:5000/myadmin/nginx
```

Pushing an Image

```
$ docker push eon01/nginx
$ docker push eon01/nginx localhost:5000/myadmin/nginx
```

Batch clean

```
docker stop -f $(docker ps -a -q)
```

 Stopping all containers

```
docker rm -f $(docker ps -a -q)
```

 Removing all containers

```
docker rmi -f $(docker images -q)
```

 Removing all images

Volumes

Check volumes

```
$ docker volume ls
```

Cleanup unused volumes

```
$ docker volume prune
```

Top Cheatsheet

Python Cheatsheet
Quick Reference

Vim Cheatsheet
Quick Reference

Remote Ws
Quick Reference

Homebrew
Quick Reference

JavaScript

Bash Cheatsheet

PyTorch Cheatsheet

Taskset Cheatsheet



QuickRef.ME

Share quick reference and cheat sheet for developers.

中文版 #Notes

