

# Docker Documentation

Kyle Chapman

February 2020

## Contents

<b>1</b>	<b>Structure</b>	<b>2</b>
<b>2</b>	<b>Commands</b>	<b>2</b>
2.1	ps . . . . .	2
2.2	run . . . . .	2
2.3	stop . . . . .	3
2.4	rm . . . . .	3
2.5	rmi . . . . .	3
2.6	images . . . . .	3
2.7	pull . . . . .	3
2.8	exec . . . . .	3
2.9	attach . . . . .	4
2.10	inspect . . . . .	4
2.11	logs . . . . .	4

# 1 Structure

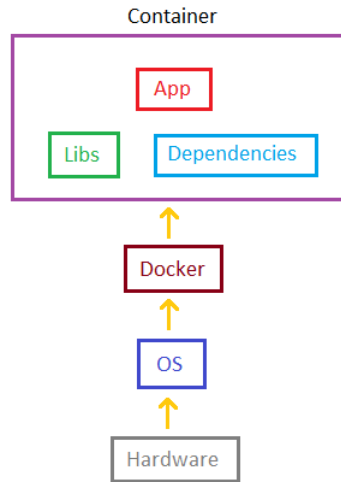


Figure 1: Structure of Docker

A **Dockerfile** is used to create images, and an image is used to deploy an application. A container only lives as long as the process inside it is alive.

## 2 Commands

Prefix all the following commands with **docker**:

### 2.1 ps

Lists all the running containers.

Examples:

- **ps**  $\Rightarrow$  List all the running containers.
- **ps -a**  $\Rightarrow$  List all containers that are currently running or have exited.

### 2.2 run

Starts a docker container with the given image, downloading the image if not already present.

Examples:

- **run <imageName>**  $\Rightarrow$  Runs a specific image in a docker container.
- **run <imageName>:<versionNumber>**  $\Rightarrow$  Runs a specific image with a specific version number. If no version number is provided, defaults to the latest version.
- **run -d <imageName>**  $\Rightarrow$  Runs a specific image in a detached state, i.e. no console output. See 2.9 to see console output.
- **run -p <externalPort>:<internalPort> <imageName>**  $\Rightarrow$  Runs a specific image and binds an external port to an internal port, so that users outside of the docker host can access the app using the **externalPort**, which would redirect to the **internalPort** inside the docker host.

- `run -v <localDir>:<dockerDir> <imageName>`  $\Rightarrow$  Runs a specific image and persists all changes to the `dockerDir` directory to the `localDir` directory.
- `run -it <imageName>`  $\Rightarrow$  Runs a specific image in interactive mode (`-i`) with a pseudo-terminal (`-t`), allowing for user input.

## 2.3 stop

Stops the running container that matches the container ID.

Example:

- `stop <containerID>`  $\Rightarrow$  Stops the container that matches the container ID.

## 2.4 rm

Removes a container that matches the container ID.

Example:

- `rm <containerID>`  $\Rightarrow$  Removes the container that matches the container ID from the local file system.

## 2.5 rmi

Removes an image that matches the image name.

Example:

- `rmi <imageName>`  $\Rightarrow$  Remove a specific image that has been downloaded.

## 2.6 images

Shows all of the downloaded docker images.

## 2.7 pull

Download a specific docker image from docker hub.

Example:

- `pull <imageName>`  $\Rightarrow$  Download the image with the matching image name.

## 2.8 exec

Executes a command on a docker container.

Example:

- `exec <containerID> <command>`  $\Rightarrow$  Executes a specific command on the matching container ID, where `<command>` can be `ls -la` as an example.

## 2.9 attach

Attaches the console to the container ID specified.

Example:

- `attach <containerID>`  $\Rightarrow$  Attach a console to the container ID specified.

## 2.10 inspect

Shows detailed information about the container specified.

Example:

- `inspect <containerID>`  $\Rightarrow$  Show detailed information about the container with the matching ID.

## 2.11 logs

Shows the logs of the container that has been specified.

Example:

- `logs <containerID>`  $\Rightarrow$  Show logs for container with matching ID.