<https://docs.docker.com/get-started/#recap-and-cheat-sheet>

Recap and cheat sheet

## List Docker CLI commands

docker

docker container --help

## Display Docker version and info

docker --version

docker version

docker info

## Execute Docker image

docker run hello-world

## List Docker images

docker image ls

## List Docker containers (running, all, all in quiet mode)

docker container ls

docker container ls --all

docker container ls -aq

## Conclusion of part one

Containerization makes [CI/CD](https://www.docker.com/use-cases/cicd) seamless. For example:

* applications have no system dependencies
* updates can be pushed to any part of a distributed application
* resource density can be optimized.

With Docker, scaling your application is a matter of spinning up new executables, not running heavy VM hosts.

<https://docs.docker.com/get-started/part2/#dockerfile>

<https://docs.docker.com/get-started/part2/#recap-and-cheat-sheet-optional>

https://docs.docker.com/toolbox/toolbox\_install\_windows/

Here is a list of the basic Docker commands from this page, and some related ones if you’d like to explore a bit before moving on.

docker build -t friendlyhello . # Create image using this directory's Dockerfile

docker run -p 4000:80 friendlyhello # Run "friendlyname" mapping port 4000 to 80

docker run -d -p 4000:80 friendlyhello # Same thing, but in detached mode

docker container ls # List all running containers

docker container ls -a # List all containers, even those not running

docker container stop <hash> # Gracefully stop the specified container

docker container kill <hash> # Force shutdown of the specified container

docker container rm <hash> # Remove specified container from this machine

docker container rm $(docker container ls -a -q) # Remove all containers

docker image ls -a # List all images on this machine

docker image rm <image id> # Remove specified image from this machine

docker image rm $(docker image ls -a -q) # Remove all images from this machine

docker login # Log in this CLI session using your Docker credentials

docker tag <image> username/repository:tag # Tag <image> for upload to registry

docker push username/repository:tag # Upload tagged image to registry

docker run username/repository:tag # Run image from a registry

<https://docs.docker.com/get-started/part3/>