**Instalación y post-instalación.**

Agregar a docker como non-root user:

<https://docs.docker.com/engine/install/linux-postinstall/>

sudo groupadd docker

sudo usermod -aG docker $USER

(restartear terminal y quizás re-loguearse)

newgrp docker

docker run hello-world

Si sigue fallando, probar:

sudo chmod 666 /var/run/docker.sock

**Create and run a container using an image**

docker run <image-name>

Ej

docker run hello-world

**Overriding default commands**

docker run <image-name> <command>

eg:

docker run busybox echo hi there

docker run busybox ls

docker run busybox ping [www.google.com](http://www.google.com/) (queda corriendo)

**Listing running containers**

docker ps

docker ps –all (all the containers ever created)

**Lifecycle**

docker run = docker create + docker start

Eg.

docker create hello-world

docker create busybox echo hi there

(devuelve el id del container creado)

docker start -a <id-del-container>

-a hace que se imprima el output del container.

**Restarting stopped containers**

docker ps –all

(obtener todos los containers)

docker start <id\_del\_container>

(puede ser con -a para ver las salidas)

**Removing stopped containers**

docker system prune

(elimina todo lo pendiente de docker)

**Retrieving output logs**

docker logs <container\_id>

Eg:

docker create busybox echo hi there

docker start <id>

docker logs <id>

**Stopping containers**

docker create busybox ping [www.google.com](http://www.google.com/)

(devuelve el id)

docker start <id>

docker logs <id>

docker ps

(container is running)

Para frenarlo (manda SIGTERM):

docker stop <id>

(luego de 10 segundos sin que se detenga manda el kill).

Para matarlo (más fuerte, SIGKILL):

docker kill <id>

**Multi-command containers (docker exec)**

Eg.:

docker run redis

(corre el Redis server)

docker exec -it <container\_id> <command>

Ejemplo:

docker exec -it <id> redis-cli

i: ancla el standard input.

t: ancla el standard output de forma linda.

**Getting a command prompt in a containers**

docker exec -it <id> sh

(abre un shell en el container).

Otras opciones bash, powershell, zsh, sh (si lo tienen)

comando “exit” para salir

**Starting with a shell**

docker run -it <image-name> sh

Eg:

docker run -it busybox sh

Problema: no se corre otro comando que tendría la imagen.

**Build a Docker image from a Dockerfile**

docker build .

(devuelve id de image)

Luego,

docker run <id>

**Tagging an image-name**

docker build -t <tag\_name> .

A partir de ahí, se puede usar el tag\_name para el run.

Eg:

(convención).

tag\_name: docker\_id/project\_name:version

version puede ser “latest”.

Eg:

docker build -t example:latest .

docker run example:latest

Se puede omitir el :latest.

**Manual creation of an image (uso raro)**

eg:

docker run -it alpine sh

Dentro, instalo cualquier cosa o hago lo que sea.

Abro otra terminal, y ejecuto:

docker ps

(para obtener id)

docker commit -c ‘CMD [“redis-server”]’ <id\_container>

(output id de la imagen).

**Specifying port forwarding**

(when running or starting)

docker run -p <host\_port>:<container\_port> <image\_id>

ej

docker run -p 8080:8080 simpleweb

**docker-compose**

Para listar imágenes:

docker-compose ps

Borrar alguna

docker-compose rm <image\_name>