dockerFile

docker build -f jar,Dockerfile -t my-api .

**using -f option to specify the name or our Dockerfile**

**using -t specify the name of the image**

list of images docker

docker images

Start container run -p

docker run -p 9000:8080 -it my-api

the container show for port 8080 to the host port 9000

Docker

* Sudo apt-get install docker.io

Comando para instalar docker

sudo docker –version

Saber la version de docker

Para crear una imagen de docker debes crear un archivo Dockerfie

Ejemplo:

FROM ubuntu:18.04

MAINTAINER Edwin Zamora "edzamo13@gmail.com"

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/\*

COPY ./pagina /var/www/html

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

EXPOSE 80

COPY ./pagina /var/www/html

ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

Para ejecutar y compilar docker sudo docker buil -t el nombre del imagen debe tener el usario de DockerHub/nombre que le asignes : verson y el punto para que tome la infirmacion en la carpeta que estas.

* sudo docker build -t edzamo13/aplicacionweb1:v1 .

Listar las Imagenes descargadas de docker

* Sudo docker image ls

Para correr el cotenedor de docker sudo docker run –name o nombre que le des mas -d -p indicando el puerto mas en nombre de la imagen que creaste.

* sudo docker run --name aplweb -d -p 80:80 edzamo13/aplicacionweb1:v1

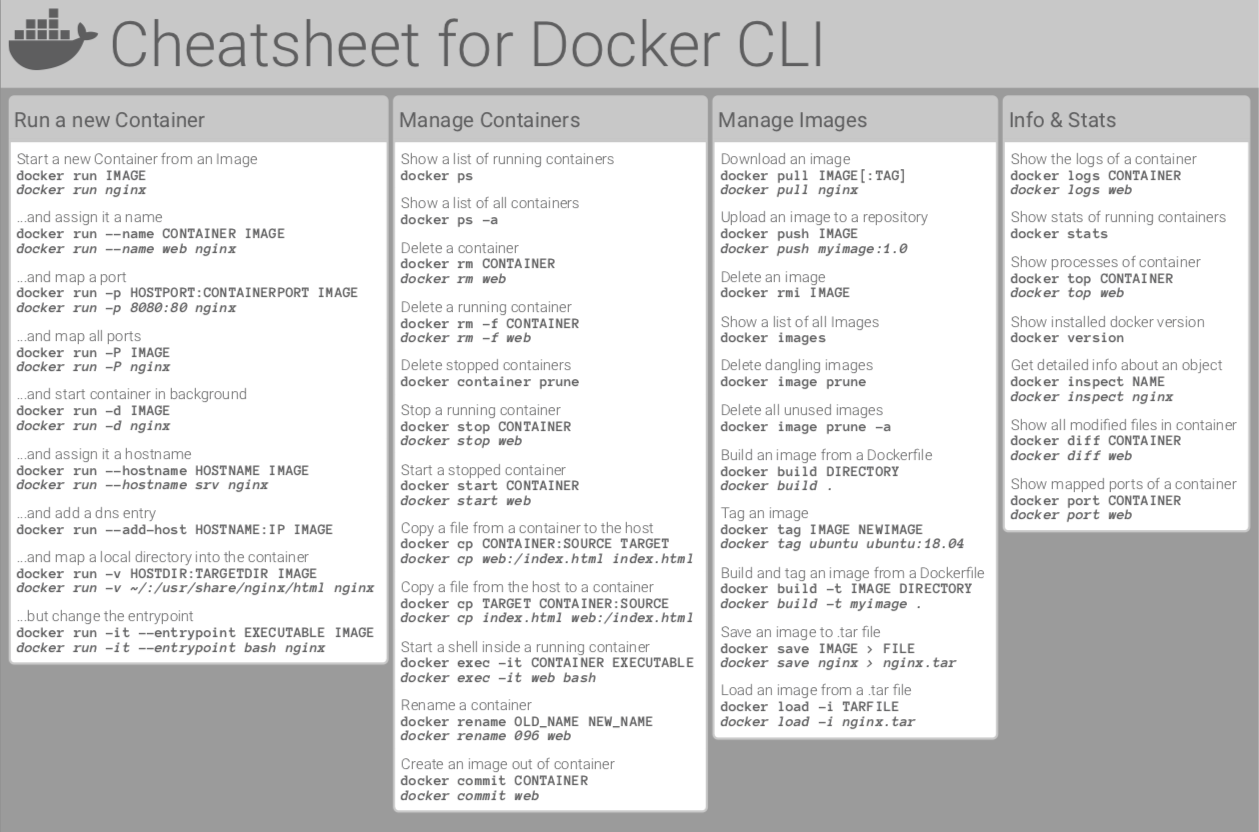
Listar los cotenedores

* Sudo docker ps

ingresar a la informacion del contenedor.

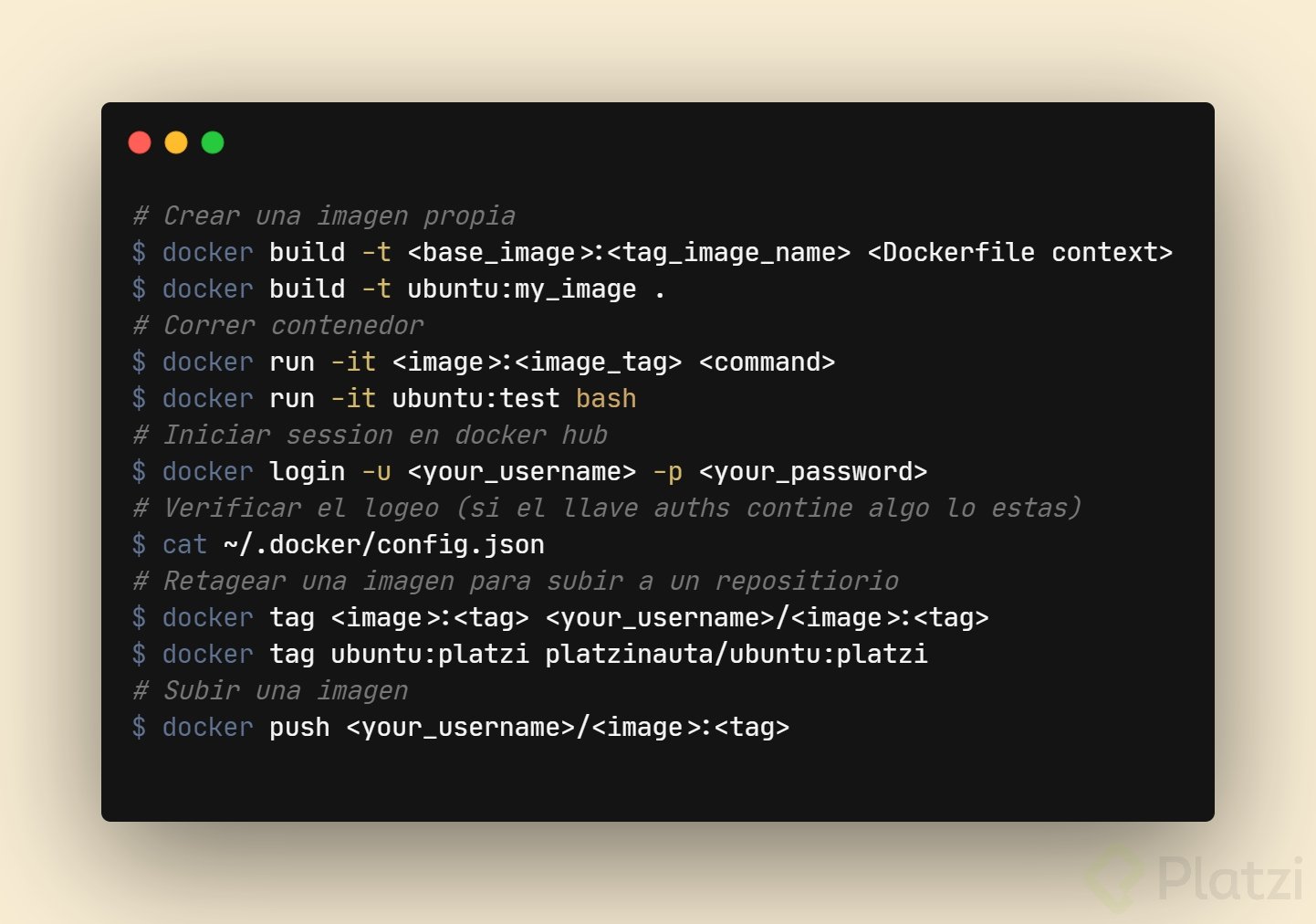
* sudo docker exec -it aplweb /bin/bash

Docker

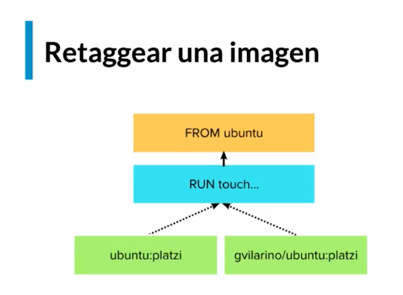


Docker history Ubuntu:platzi

docker run --rm -p 3000:3000 platziapp



If you need Retaggear an image or change the name , checkout that example





Docker tag ubuntu:platzi edzamo13/ubuntu:platzi

That command say how is owner/ SO :version SO

Dive:

Is a tools, that help you see the history about image



Docker network

<https://docs.docker.com/network/>

Tipos de redes son:

* Host.
* Bridge.
* None.
* Overlay.
* Macvlan.

With that command created a network and say that other container can join

docker network create --attachable platzinet

info about that network

docker network inspect platzinet

summary

Comandos:

$ docker network ls (listo las redes)

$ docker network create --atachable plazinet (creo la red)

$ docker inspect plazinet (veo toda la definición de la red creada)

$ docker run -d --name db mongo (creo el contenedor de la BBDD)

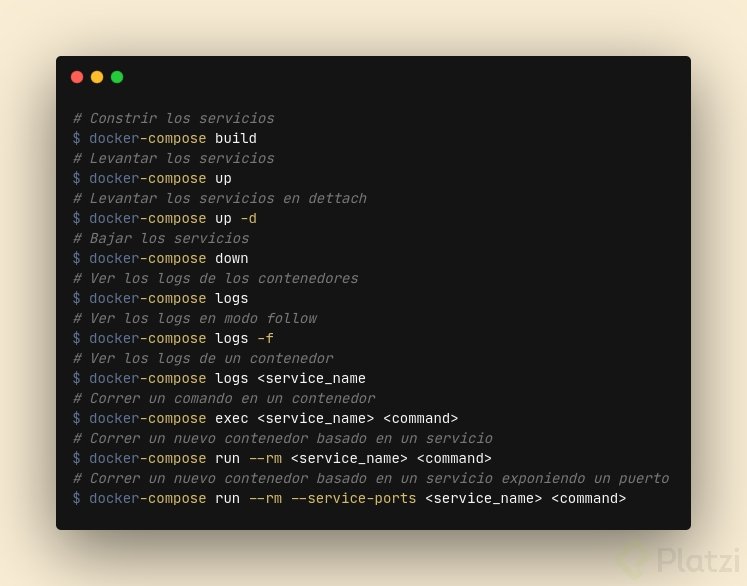
$ docker network connect plazinet db (conecto el contenedor “db” a la red “platzinet”)

$ docker run -d -name app -p 3000:3000 --env MONGO\_URL=mondodb://db:27017/test platzi (corro el contenedor “app” y le paso una variable)

$ docker network connect plazinet app (conecto el contenedor “app” a la red “plazinet”)

Docker Compose

|\_\_ docker-compose.yml  
|\_\_ Dockerfile



Example docker-compose up .. show you all the container logs

docker-compose up -d, no show you any container log

Comandos:

$ docker network ls (listo las redes)

$ docker network inspect docker\_default (veo la definición de la red)

$ docker-compose logs (veo todos los logs)

$ docker-compose logs app (solo veo el log de “app”)

$ docker-compose logs -f app (hago un follow del log de app)

$ docker-compose exec app bash (entro al shell del contenedor app)

$ docker-compose ps (veo los contenedores generados por docker compose)

$ docker-compose down (borro todo lo generado por docker compose)

2 instances up

docker-compose up -d --scale app=2

$ docker ps -a (veo todos los contenedores de mi máquina)

$ docker container prune (borra todos los contenedores inactivos)

$ docker rm -f $(docker ps -aq) (borra todos los contenedores que estén corriendo o apagados)

$ docker network ls (lista todas las redes)

$ docker volume ls (lista todos los volumes)

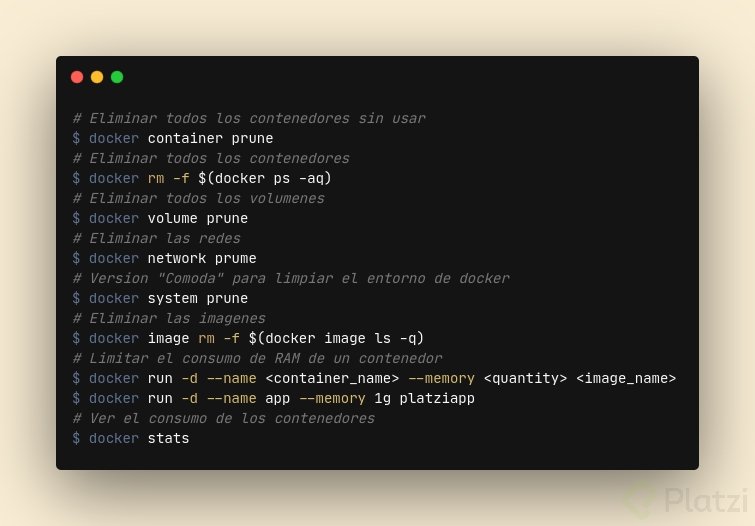
$ docker image ls (lista todas las imágenes)

$ docker system prune (borra todo lo que no se esté usando)

$ docker run -d --name app --memory 1g platziapp (limito el uso de memoria)

$ docker stats (veo cuantos recursos consume docker en mi sistema)

$ docker inspect app (puedo ver si el proceso muere por falta de recursos)



$ docker build -t loop . (construyo la imagen)

$ docker run -d --name looper loop (corro el contenedor)

$ docker stop looper (le envía la señal SIGTERM al contenedor)

$ docker ps -l (muestra el ps del último proceso)

$ docker kill looper (le envía la señal SIGKILL al contenedor)

$ docker exec looper ps -ef (veo los procesos del contenedor)

