List of steps

Here a list of step whether you start from scratch or you want launch your created image dockers correctly.

Creating container

This series of commands must be written in root project, that is, to open a terminal inside of **cointers_sample1** folder

`user:~/Documents/worksapce/containers$`

> docker build -t db-misitio ./misitio_db/
>
> docker build -t db-misitio ./misitio_nodejs/

Once it's has been created, these containers, in their respective folders there is a **launcher. sh** which automatically creates an image and launch it in.

`user:~/Documents/worksapce/containers$`

> sudo ./misitio_db/launch.sh
>
> sudo ./misitio_nodejs/launch.sh

There is an order of how these images have to be launched due to dependences between projects. In this case, project_node depend of project_db. When there are more dependencies and more and more projects this current method is an impossible task. This problem would be resolved using Compose.

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration

Stop image

```
image=$(docker ps -a --filter="name=$NAME_IMAGE" -q | xargs)

docker stop $image
```

Start image

```
image=$(docker ps -a --filter="name=$NAME_IMAGE" -q | xargs)

docker start $image
```

Useful command series

```
manu:~/containers_sample1$ docker ps -a --filter="name=mysql-running" -q |
xargs
e0439377d704
```

```
manu:~/containers_sample1$ docker inspect -f '{{.Name}} -
{{.NetworkSettings.IPAddress }}' $(docker ps -aq)
/mysql-running -
manu:~/containers_sample1$ docker ps -a --filter="name=mysql-running" -q |
xargs
e0439377d704
manu:~/containers_sample1$ image=$(docker ps -a --filter="name=mysql-running"
-q | xargs)
manu:~/containers_sample1$ docker start $image
e0439377d704
manu:~/containers_sample1$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
      STATUS              PORTS                                NAMES
e0439377d704        db-misitio          "docker-entrypoint.s…"   12 hours ago
       Up 7 seconds        33060/tcp, 0.0.0.0:6603->3306/tcp   mysql-running

manu@manu-GS60-6QE:~/containers_sample1$ docker inspect -f '{{.Name}
{{.NetworkSettings.IPAddress }}' $(docker ps -aq)

/mysql-running - 172.17.0.2
manu@manu-GS60-6QE:~/containers_sample1$
```