Prácticas Docker

1. Inspects

- Nos bajamos la imagen de MongoDB

```
docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
b0568b191983: Pull complete
1e8b5b4e67a0: Pull complete
a87b29dbb553: Pull complete
908c259a6a99: Pull complete
2fdec16e62a3: Pull complete
e233c6c476cb: Pull complete
671c302f3b40: Pull complete
207ff3c88601: Pull complete
3a7998bea9fd: Pull complete
18d605d67f26: Pull complete
Digest:
sha256:d969194a7c4dcd817cae5dc726b1493547a3ad3f1b30f7f5885
7adc9ae6a4483
Status: Downloaded newer image for mongo:latest
```

- Comprobamos que la tenemos

```
docker images
REPOSITORY          TAG                 IMAGE ID
CREATED             SIZE
nginx               latest              73acd1f0cfad
7 days ago          109MB
mongo               latest              5b1317f8158f
8 days ago          366MB
fedora              26                  ce241ce855c8
2 weeks ago         232MB
busybox             latest              f6e427c148a7
3 weeks ago         1.15MB
hello-world         latest              f2a91732366c
4 months ago        1.85kB
```

-

- Comprobamos las propiedades de la imagen. Debe salir bastante información

```
docker image inspect mongo
…..
….
    "Name": "devicemapper"
        },
        "RootFS": {
            "Type": "layers",
            "Layers": [

"sha256:43efe85a991cac5894f91ee8f45b328bbacd14966d89a8a00b
0d06060c64b5ad",

"sha256:54e8db6ab32d7f1440c785b926f844c8034ad8bd669084f009
bcb4e814240672",

"sha256:217a81d3bde9d92d8d59529034afb10f878d68acb1152ce687
d78c5ba951c45f",

"sha256:d1a481118c6ef327aea3f03746ae8d56b04afa80151bee3d80
75f26c68d8a746",

"sha256:b597eb624250fab9c1aa22e29d67ce7d989c33424668a8e785
27e508bf323bc9",

"sha256:4786aaf122f108a3b4f5da7ac80871e4e2014e97ff22883729
1e55ba71a75cb0",

"sha256:2ecbdcef31f175b7005d62bcb981d79f8543ed0564a3fd43c9
de6ccd3783930c",

"sha256:d6ac487f77161c42f76480b4c8ca533c2578aaaa55b114b61b
ca56b9cd1a856a",

"sha256:5388bfbc2c0172298c169f266f196d0b40b27a80447044b52d
9496f7a576e1d5",

"sha256:99099bc0f52d2ba7654b36eff2ef63af7d50d7c103af342512
a03e18e94373fc"
```

```
            ]
        },
        "Metadata": {
            "LastTagTime": "0001-01-01T00:00:00Z"
        }
```

- Podemos usar GREP para encontrar información más concreta

```
docker image inspect mongo | grep MONGO_VERSION
Error: No such object: image
                "MONGO_VERSION=3.6.3"
                "MONGO_VERSION=3.6.3"
```

- Lo mejor es mandar la salida a un fichero para inspeccionarlo después

```
docker image inspect mongo > mongo.json
```

- Creamos ahora un contenedor con esa imagen

```
docker run -d --name mongo1 mongo
2a08487b2fdfa9774586b20df7a53792cc23ff66bd95eac0c6f957dff0
ab92d5


docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
2a08487b2fdf        mongo               "docker-
entrypoint.s…"   6 minutes ago       Up 6 minutes
27017/tcp           mongo1
```

- Lanzamos un inspect contra el contenedor. También debe aparecer mucha información

```
docker inspect mongo1
…
…
…
..
..
```

```
"Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID":
"898b3b68d1579099e41e46778825e580f3c7b5859acfb712d3237d247
94dc7ed",
                    "EndpointID":
"47e0db1eefac39fb381a9248eeae6a0c6ceaa2ddec65a1f67dab11f51
6532d40",
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:02",
                    "DriverOpts": null
                }
            }
```

- Podemos usar de nuevo el GREP. Por ejemplo, hay un campo que nos permite ver la dirección IP que ha asignado al contenedor
-

```
docker inspect mongo1 | grep IPAddress
            "SecondaryIPAddresses": null,
            "IPAddress": "172.17.0.2",
                    "IPAddress": "172.17.0.2",
```

- También podemos ver la memoria compartida asignada
-

```
docker inspect mongo1 | grep ShmSize
            "ShmSize": 67108864,
```

- Tenemos la posibilidad de utilizar un formato para encontrar determinada información, aunque eso nos obliga a conocer el nombre completo de la

propiedad y su jerarquía. Se usa la opción –format. Se usan plantilla creada con el lenguaje GO

- Por ejemplo, para saber las direcciones IP que puede tener el contenedor:

```
docker inspect --format='{{range
.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mongo1


172.17.0.2
```

- O donde deja el fichero log

```
docker inspect --format='{{.LogPath}}' mongo1
/var/lib/docker/containers/2a08487b2fdfa9774586b20df7a5379
2cc23ff66bd95eac0c6f957dff0ab92d5/2a08487b2fdfa9774586b20d
f7a53792cc23ff66bd95eac0c6f957dff0ab92d5-json.log
```

- Y por último podemos ver la imagen en la que está basado

```
docker inspect --format='{{.Config.Image}}'  mongo1
mongo
```