Horas de Libre Configuración

2019-2020

Fco. Javier Martínez

1. *Instalar Docker en Ubuntu.*
2. *Ejecuta la imagen “hello-world”.*
3. *Muestra los contenedores Docker*
4. *Edita el fichero Dockerfile*
5. *Construye el contenedor*
6. *Ejecutalo*
7. *Create una cuenta en hub.docker.com*
8. *Publícalo*
9. *Descarga la imagen de Ubuntu*
10. *Descarga la imagen de hello-world*
11. *Descarga la imagen nginx*
12. *Muestra un listado de todas la imágenes*
13. *Ejecuta un contenedor hello-world y dale nombre “myhello1”*
14. *Ejecuta un contenedor hello-world y dale nombre “myhello2”*
15. *Ejecuta un contenedor hello-world y dale nombre “myhello3”*
16. *Muestra los contenedores que se están ejecutando*
17. *Para el contenedor “myhello1”*
18. *Para el contenedor “myhello2”*
19. *Borra el contendor “myhello1”*
20. *Muestra los contenedores que se están ejecutando*
21. *Borra todos los contenedores*
22. *Crea un archivo Dockerfile con el siguiente contenido*
    i. *FROM httpd:2.4*
    ii. *COPY ./public-html/ /usr/local/apache2/htdocs/*
23. *Crea una página web y guardala en “public-html”*
24. *Crea una imagen y llámala myhttp*
25. *Crea un contenedor usando una imagen “Apache”*
26. *Incluye un volumen conectado con la máquina anfitriona*
27. *Crea una página web en dicho directorio*
28. *Comprueba el resultado*

1. .

- Lo primero que vamos ha hacer es un update con “sudo apt update”. Después, vamos a instalar algunos paquetes de requisitos previos que le permiten a apt usar paquetes mediante HTTPS con “ sudo apt install apt-transport-https ca-certificates curl software-properties-common”. Más tarde, agregaremos la clave GPG para añadir el repositorio oficial de Docker con “curl -fsSl https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –“. También agregaremos el repositorio de Docker a las fuentes de APT con “sudo add-apt-repository “deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable”, tras esto, volveremos a hacer un Update. Cuando finalice, instalaremos Docker con “ sudo apt install Docker-ce”.

```
root@examencliente:~# apt install apt-transport-https ca-certificate
ware-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20180409).
fijado ca-certificates como instalado manualmente.
software-properties-common ya está en su versión más reciente (0.96.
```

```
root@examencliente:~# curl -fsSl https://download.docker.com/linux/u
 sudo apt-key add -
OK
```

```
root@examencliente:~# add-apt-repository "deb [arch=amd64] https://d
ker.com/linux/ubuntu bionic stable"
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [8
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [
Des:4 https://download.docker.com/linux/ubuntu bionic InRelease [64,
Des:5 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Des:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 D
```

```
root@examencliente:~# apt install docker-ce
```

2. .

- *Para ejecutar la imagen hello-world debemos introducir “sudo Docker run hello-world” .*

```
root@examencliente:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:fc6a51919cfeb2e6763f62b6d9e8815acbf7cd2e476ea35374357
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working corr

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker
    (amd64)
 3. The Docker daemon created a new container from that image which
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, whi
    to your terminal.

To try something more ambitious, you can run an Ubuntu container wit
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

3. .

- *Ejecutando “Docker ps -a” mostramos todos los contenedores.*

```
root@examencliente:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
 STATUS                      PORTS               NAMES
6fec6b8c9c99        hello-world         "/hello"            3 minut
 Exited (0) 3 minutes ago                        optimistic_montalci
```

4. .

- Empezaremos descargando el proyecto introduciendo:

```
root@examencliente:~# curl -LO https://github.com/dockersamples/node-b
oard/archive/master.zip unzip master.zip cd node-bulletin-board-master
-board-app
```

-
- El proyecto es una aplicación simple de tablón de anuncios, escrita en Node.js.

```
                                 Dload  Upload   Total   Spent    Left  Spe
100   138  100   138    0     0    370      0 --:--:-- --:--:-- --:--:--
100 36004    0 36004    0     0  31145      0 --:--:--  0:00:01 --:--:-- 11
root@examencliente:/# unzip master.zip
Archive:  master.zip
49ee8e8183de21e642f62c3569518fafc8b7a9ed
   creating: node-bulletin-board-master/
  inflating: node-bulletin-board-master/.gitignore
  inflating: node-bulletin-board-master/LICENSE
   creating: node-bulletin-board-master/bulletin-board-app/
 extracting: node-bulletin-board-master/bulletin-board-app/.gitignore
  inflating: node-bulletin-board-master/bulletin-board-app/Dockerfile
  inflating: node-bulletin-board-master/bulletin-board-app/LICENSE
  inflating: node-bulletin-board-master/bulletin-board-app/app.js
   creating: node-bulletin-board-master/bulletin-board-app/backend/
  inflating: node-bulletin-board-master/bulletin-board-app/backend/api.js
  inflating: node-bulletin-board-master/bulletin-board-app/backend/events.j
  inflating: node-bulletin-board-master/bulletin-board-app/backend/index.js
   creating: node-bulletin-board-master/bulletin-board-app/fonts/
   creating: node-bulletin-board-master/bulletin-board-app/fonts/geomanist/
 extracting: node-bulletin-board-master/bulletin-board-app/fonts/geomanist/
ed-Geomanist-Book.woff2
  inflating: node-bulletin-board-master/bulletin-board-app/index.html
  inflating: node-bulletin-board-master/bulletin-board-app/package.json
  inflating: node-bulletin-board-master/bulletin-board-app/readme.md
  inflating: node-bulletin-board-master/bulletin-board-app/server.js
```

-
- Nos dirigimos al directorio node-bulletin-board/bulletin-board-app y entramos con nano a editar el fichero Dockerfile.

```
root@examencliente:/# cd node-bulletin-board-master/bulletin-board-app
root@examencliente:/node-bulletin-board-master/bulletin-board-app# ls
app.js   Dockerfile  index.html  package.json  server.js
backend  fonts       LICENSE     readme.md     site.css
root@examencliente:/node-bulletin-board-master/bulletin-board-app# nano Dockerf
ile
root@examencliente:/node-bulletin-board-master/bulletin-board-app#
```

-

```
  GNU nano 2.9.3                    Dockerfile

FROM node:current-slim

WORKDIR /usr/src/app
COPY package.json .
RUN npm install

EXPOSE 8080
CMD [ "npm", "start" ]

COPY . .
```

- Ahora vamos a construir la imagen:

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker image
 build -t bulletinboard:1.0 .
Sending build context to Docker daemon  45.57kB
Step 1/7 : FROM node:current-slim
current-slim: Pulling from library/node
6d28e14ab8c8: Downloading  10.33MB/22.51MB
467f76cd7b76: Download complete
5267769fc2ec: Downloading  11.08MB/24.15MB
0001c8b9bd2f: Download complete
2469aef99d27: Waiting
```

-

5. .

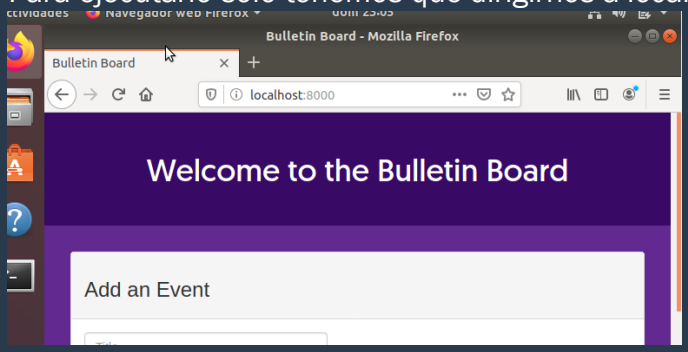
- *Ahora vamos a comenzar un contendor basado en la imagen:*

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker
iner run --publish 8000:8080 --detach --name bb bulletinboard:1.0
de8cfc505f18d176878792971d969a8dbdf96f044c48ca312b67c4135b4571fd
```

-
-

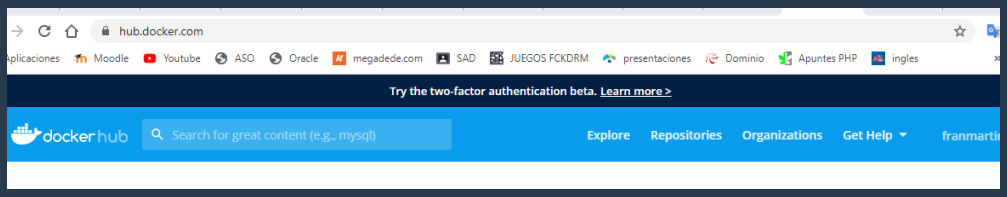
6. .

- Para ejecutarlo solo tenemos que dirigirnos a localhost:8000



7. .

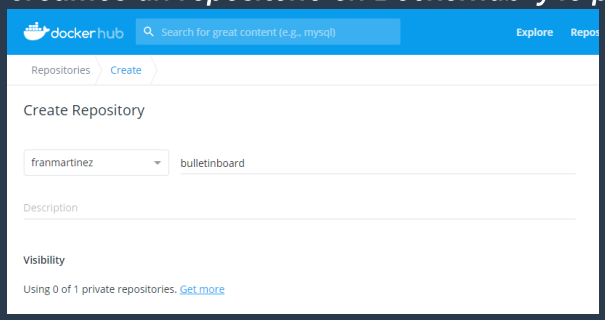
-



8. .

- *Creamos un repositorio en Dockerhub y le pondremos bulletinboard*

-



- *En el terminal iniciamos sesion escribiendo "docker login"*

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
 have a Docker ID, head over to https://hub.docker.com to create one.
Username: franmartinez
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker image
 tag bulletinboard:1.0 franmartinez/bulletinboard:1.0
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker image
 push franmartinez/bulletinboard:1.0
The push refers to repository [docker.io/franmartinez/bulletinboard]
9263fb05f7d4: Pushed
f3c8ea9bfadb: Pushed
06dafc991019: Pushed
cc40e612ed9d: Pushed
7eac09f55f5f: Mounted from library/node
d03234c17ca7: Mounted from library/node
d7fc38da5a48: Mounted from library/node
8ef78c7801b9: Mounted from library/node
f66ed577df6e: Mounted from library/node
1.0: digest: sha256:a0d935441d0c8735f045a6f9ea63d5a579630a79bf22dad67e02f585889
a843c size: 2201
```

-

9. .

- *Ejecutamos en la terminal "Docker search Ubuntu" y "Docker pull Ubuntu"*

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker pull
ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
423ae2b273f4: Downloading  5.827MB/26.69MB
de83a2304fa1: Download complete
f9a83bce3af0: Download complete
b6b53be908de: Waiting
```

-

10. .

- *Ejecutamos en la terminal "Docker search hello-world" y "Docker pull hello-world"*

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker pull
hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:fc6a51919cfeb2e6763f62b6d9e8815acbf7cd2e476ea353743570610737b752
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
```

-

11. *Ejecutamos en la terminal "Docker search nginx" y "Docker pull nginx"*

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker pull
nginx
Using default tag: latest
latest: Pulling from library/nginx
68ced04f60ab: Downloading  8.121MB/27.09MB
28252775b295: Downloading  7.134MB/23.92MB
a616aa3b0bf2: Download complete
```

-

12. .

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker image
s
REPOSITORY                TAG                 IMAGE ID            CREATED
          SIZE
frannmartinez/bulletinboard   1.0                 feeded403515        32 minutes
 ago       154MB
bulletinboard             1.0                 feeded403515        32 minutes
 ago       154MB
nginx                     latest              6678c7c2e56c        4 days ago
          127MB
node                      current-slim        26932a190e66        11 days ag
o         140MB
ubuntu                    latest              72300a873c2c        2 weeks ag
o         64.2MB
hello-world               latest              fce289e99eb9        14 months
ago       1.84kB
```

13. .

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker run h
ello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docke
e hello-world myhello1
```

14. .

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docke
e hello-world myhello2
```

15. .

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docke
e hello-world myhello3
```

16. .

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
      STATUS              PORTS               NAMES
de8cfc505f18        bulletinboard:1.0   "docker-entrypoint.s…"   38 minutes ago
```

17. .

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker
myhello1
```

18. –

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docke
myhello2
```

19. –

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker
hello1
```

20. –

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
      STATUS              PORTS               NAMES
de8cfc505f18        bulletinboard:1.0   "docker-entrypoint.s…"   38 minutes ago
```

21. –

-

```
root@examencliente:/node-bulletin-board-master/bulletin-board-app# docker
docker ps -a -q)
```

22. –

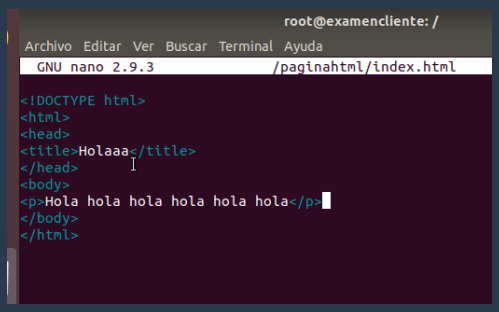
-

```
root@examencliente: /node-bulletin-board-master/b
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  GNU nano 2.9.3                  Dockerfile

FROM httpd:2.4
COPY ./public-html/ /usr/local/apache2/htdocs/
```

23. –

-


root@examencliente: /
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /paginahtml/index.html
<!DOCTYPE html>
<html>
<head>
<title>Holaaa</title>
</head>
<body>
Hola hola hola hola hola hola
</body>
</html>