Hadrián
Villar Cuadrado

Tema 4: Docker

Condiciones de entrega:

La forma de entregar los ejercicios será en un fichero ZIP con el formato **ApellidosNombreTarea4.pdf**.

Puntuación:

El boletín está puntuado sobre 10 puntos. En caso de no entregarlo en fecha, hay una prórroga de 3 días en las que se puede entregar con penalización de 1 punto por día hasta un mínimo de 3.5. A partir del tercer día la tarea será puntuada con un cero.
El ejercicio debe funcionar cuando se ejecute.

La nota mínima para considerar aprobado el boletín es de 3.5 puntos.

Ejercicios:

Para hacer estos ejercicios debemos utilizar la máquina virtual de Docker.

La forma de entregar los ejercicios será en un PDF donde vayáis explicando, mediante pantallazos, cómo vais haciendo el ejercicio, y cada comando que utilizáis.

1. Ejercicio 1 (3 puntos)

    a. Crearemos un contenedor **Ubuntu** y ejecutaremos el comando "**ls -l**" dentro de él.
    b. Comprobamos qué imagen se ha descargado.
    c. Comprobamos qué contenedores hay activos.
    d. Eliminamos el contenedor.

2. Ejercicio 2 (3 puntos)

    a. Crearemos un contenedor **Ubuntu**, con nombre Ejercicio2 y ejecutaremos **de manera interactiva** la instrucción "bin/bash".
    b. Comprobamos qué contenedores hay activos
    c. Salimos de ese contenedor, **PARÁNDOLO**
    d. Comprobamos que el contenedor Ejercicio2 **NO está activo**
    e. Arrancamos ese contenedor
    f. Comprobamos que el contenedor **ESTÁ ACTIVO**
    g. Borramos el contenedor **SIN PARARLO**

3. Ejercicio 3 (3 puntos)

Queremos montar un servidor Tomcat para que nos sirva de servidor de aplicaciones. Para ello, lanzaremos un contenedor **Tomcat:8.0** con las siguientes características:

a. Nombre: Ejercicio3
b. Autor: Nombre del alumno
c. Puerto público: 8082
d. Puerto del contenedor: 8080

Nos conectaremos desde nuestro navegador al localhost:8082 y mostramos el resultado obtenido.

4. Ejercicio 4 (1 puntos)

a. Crearemos una imagen con el contenedor del Ejercicio3
b. Ejecutaremos esa nueva imagen
c. Subiremos esa imagen a nuestro repositorio de Docker Hub

Antes de trabajar con los contenedores e imágenes de Docker, hemos de crearnos una cuenta en su página oficial y saber un poco mas de los comandos que utilizaremos leyendo api o documentación oficial de Docker.

Ejercicio 1.

Para crear un contenedor, primero debemos de buscar la imagen deseada desde la cual queremos crear el contenedor, por ello, hemos de realizar una búsqueda en los repositorios de Docker con el comando "**docker search NombreImagen**". En este caso, utilizaremos una imagen de Ubuntu:

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker search ubuntu
NAME                                                      DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
ubuntu                                                    Ubuntu is a Debian-based Linux operating sys…   11933               [OK]
dorowu/ubuntu-desktop-lxde-vnc                            Docker image to provide HTML5 VNC interface …   503                                     [OK]
websphere-liberty                                         WebSphere Liberty multi-architecture images …   267                 [OK]
rastasheep/ubuntu-sshd                                    Dockerized SSH service, built on top of offi…   250                                     [OK]
consol/ubuntu-xfce-vnc                                    Ubuntu container with "headless" VNC session…   234                                     [OK]
ubuntu-upstart                                            Upstart is an event-based replacement for th…   110                 [OK]
neurodebian                                               NeuroDebian provides neuroscience research s…   80                  [OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5      ubuntu-16-nginx-php-phpmyadmin-mysql-5          50                                      [OK]
ubuntu-debootstrap                                        debootstrap --variant=minbase --components=m…   44                  [OK]
open-liberty                                              Open Liberty multi-architecture images based…   42                  [OK]
nuagebec/ubuntu                                           Simple always updated Ubuntu docker images w…   24                                      [OK]
i386/ubuntu                                               Ubuntu is a Debian-based Linux operating sys…   24
1and1internet/ubuntu-16-apache-php-5.6                    ubuntu-16-apache-php-5.6                        14                                      [OK]
1and1internet/ubuntu-16-apache-php-7.0                    ubuntu-16-apache-php-7.0                        13                                      [OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10   ubuntu-16-nginx-php-phpmyadmin-mariadb-10       11                                      [OK]
1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4         ubuntu-16-nginx-php-5.6-wordpress-4             8                                       [OK]
1and1internet/ubuntu-16-apache-php-7.1                    ubuntu-16-apache-php-7.1                        6                                       [OK]
darksheer/ubuntu                                          Base Ubuntu Image -- Updated hourly             5                                       [OK]
1and1internet/ubuntu-16-nginx-php-7.0                     ubuntu-16-nginx-php-7.0                         4                                       [OK]
pivotaldata/ubuntu                                        A quick freshening-up of the base Ubuntu doc…   4
pivotaldata/ubuntu16.04-build                             Ubuntu 16.04 image for GPDB compilation         2
smartentry/ubuntu                                         ubuntu with smartentry                          1                                       [OK]
pivotaldata/ubuntu-gpdb-dev                               Ubuntu images for GPDB development              1
1and1internet/ubuntu-16-php-7.1                           ubuntu-16-php-7.1                               1                                       [OK]
pivotaldata/ubuntu16.04-test                              Ubuntu 16.04 image for GPDB testing             0
```

Una vez elijamos la imagen deseada ejecutaremos el comando "**docker pull NombreImagen**" para poder tener la imagen en local y poder trabajar con ella.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5d3b2c2d21bb: Pull complete
3fc2062ea667: Pull complete
75adf526d75b: Pull complete
Digest: sha256:b4f9e18267eb98998f6130342baacaeb9553f136142d40959a1b46d6401f0f2b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Una vez ya descargada la imagen podremos ejecutarla finalmente con el comando "**docker run NombreImagen**", en este caso hemos de hacer el comando "**ls -l**" dentro del contenedor por lo que debemos de ejecutar el comando de Docker como se muestra a continuación:

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker run ubuntu ls -l
total 48
lrwxrwxrwx   1 root root    7 Feb 17 01:04 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 15  2020 boot
drwxr-xr-x   5 root root  340 Mar 13 18:05 dev
drwxr-xr-x   1 root root 4096 Mar 13 18:05 etc
drwxr-xr-x   2 root root 4096 Apr 15  2020 home
lrwxrwxrwx   1 root root    7 Feb 17 01:04 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Feb 17 01:04 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Feb 17 01:04 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Feb 17 01:04 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Feb 17 01:04 media
drwxr-xr-x   2 root root 4096 Feb 17 01:04 mnt
drwxr-xr-x   2 root root 4096 Feb 17 01:04 opt
dr-xr-xr-x 252 root root    0 Mar 13 18:05 proc
drwx------   2 root root 4096 Feb 17 01:19 root
drwxr-xr-x   1 root root 4096 Mar  4 02:24 run
lrwxrwxrwx   1 root root    8 Feb 17 01:04 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Feb 17 01:04 srv
dr-xr-xr-x  13 root root    0 Mar 13 18:05 sys
drwxrwxrwt   2 root root 4096 Feb 17 01:19 tmp
drwxr-xr-x   1 root root 4096 Feb 17 01:04 usr
drwxr-xr-x   1 root root 4096 Feb 17 01:19 var
```

***NOTA**: El comando "**docker run NombreImagen**", si no tiene descargada una imagen con ese nombre la descarga y ejecuta automáticamente.*

Para comprobar que imágenes están descargadas en nuestro equipo, utilizaremos el comando "**docker images**", este, nos mostrará el listado de imágenes con el nombre de la imagen/repositorio, tag, id de la imagen, fecha de creación y tamaño de cada una de estas.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              4dd97cefde62        9 days ago          72.9MB
```

Si queremos comprobar que contenedores están activos, emplearemos el comando "**docker ps -a**", esto nos mostrará todos los contenedores activos.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
26728c4065ce        ubuntu              "ls -l"             24 minutes ago      Exited (0) 24 minutes ago                       musing_rubin
```

***NOTA**: En el comando "**docker ps -a**", el modificador "**-a/--all**" nos permite mostrar tanto los contenedores que están parados como corriendo. Sin este solo mostraría los que están corriendo.*

En el caso de querer eliminar un contenedor, hemos de identificar cual es el que queremos que desaparezca, para ello listaremos todos los contenedores con "**docker ps -a**" y nos quedaremos con el **id del container** (no hace falta recordar todo el id, solo la parte que no coincide con los demás, ya que debemos especificar el contenedor que borraremos).

Cuando tengamos el id, lo eliminaremos con el comando "**docker rm IdContendor**". Para comprobar que lo ha eliminado listaremos de nuevo los contenedores.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
26728c4065ce        ubuntu              "ls -l"             24 minutes ago      Exited (0) 24 minutes ago                       musing_rubin
root@alumnodaw-VirtualBox:/home/alumnodaw# docker rm 26
26
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Ejercicio 2.

Si queremos crear un contenedor y a este, nombrarlo, utilizaremos el siguiente comando "**docker run --name NombreCont NombreImagen**". Como en este caso se trata de una imagen de Ubuntu, con nombre Ejercicio2 y ejecutaremos de manera interactiva la instrucción "bin/bash".

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker run -it --name Ejercicio2 ubuntu
root@5445792e1d2b:/# dir
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@5445792e1d2b:/#
```

***NOTA**: En el comando "**docker run**", el modificador "**-it**" nos permite ejecutar el comando de forma interactiva, conectando la consola con el comando del contenedor.*

Nuevamente, si queremos ver que contenedores están activos, emplearemos el comando "**docker ps -a**", comprobando así, que el nombre del contenedor es el deseado.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
5445792e1d2b        ubuntu              "/bin/bash"         About a minute ago   Up About a minute                       Ejercicio2
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Para acceder a un contenedor que se está ejecutando en segundo plano utilizaremos el comando "**docker attach IdContenedor**", esto nos permitirá pararlo con el comando "**exit**" dentro de él.

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker attach 54
root@5445792e1d2b:/# exit
exit
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

***NOTA**: Para salir de un contenedor podemos utilizar "**exit**" y parar el contenedor o la combinación de teclas "**Ctrl-Q + Ctrl-P**" para que se siga ejecutando en un segundo plano.*

Ejecutaremos de nuevo el comando "**docker ps**" para comprobar que **NO** esté activo como se muestra a continuación:

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Para arrancar un contenedor utilizaremos "**docker start IdContenedor**". Este comando, hará que dicho contenedor se ejecute. Para asegurarnos mostramos el listado de los contenedores activos con "**docker ps**".

Para eliminar un contenedor sin pararlo, usaremos un modificador del comando rm que fuerza la eliminación de este, estando o no en ejecución "**docker rm -f IdContenedor**"

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker start 54
54
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
5445792e1d2b        ubuntu              "/bin/bash"         7 minutes ago       Up 57 seconds                           Ejercicio2
root@alumnodaw-VirtualBox:/home/alumnodaw# docker rm -f 54
54
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Ejercicio 3.

Para montar un servidor Tomcat que nos sirva de servidor de aplicaciones, lanzaremos un contenedor Tomcat:8.0 que cumpla las siguientes características:

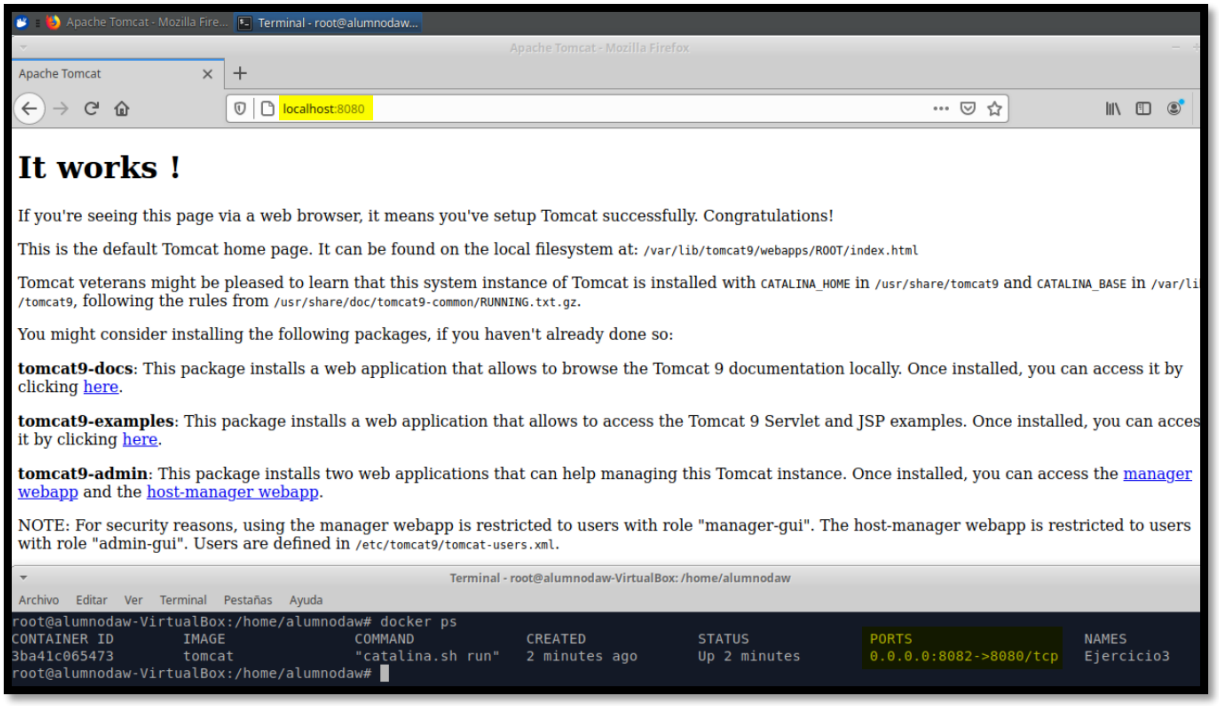
1. Nombre: Ejercicio3
2. Autor: Nombre
3. Puerto público: 8082
4. Puerto del contenedor: 8080

Una vez encontrada la imagen deseada ("**docker search tomcat**"), haremos un "**pull**" de esta y ejecutaremos el contenedor con los siguientes parámetros o modificadores, es importante que la sintaxis de estos, sea como se muestra o dará error:

1. --name NombreContenedor → Da nombre al contenedor.
2. -e AUTHOR="Nombre" → Asigna autor al contenedor.
3. -d → Ejecuta en segundo plano.
4. -p XXXX:YYYY → Redirige el puerto X de la máquina al puerto Y del contenedor.
5. NombreImagen

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker run --name Ejercicio3 -e AUTHOR="Hadrian" -d -p 8082:8080 tomcat
78b7cbe7c303bfc76ce06c3e1cf1a6076dc02b3ebebc25f96afcfa8b99e75988
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
78b7cbe7c303        tomcat              "catalina.sh run"   About a minute ago   Up About a minute   0.0.0.0:8082->8080/tcp   Ejercicio3
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Como resultado, obtendremos algo parecido a la siguiente captura:



Ejercicio 4.

Para crear una imagen a partir del contendor que ya tenemos de tomcat con nombre "Ejercicio3", debemos de usar el comando "**docker commit IdContenedor NombreImagen**" como se muestra en la imagen posterior:

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
78b7cbe7c303        tomcat              "catalina.sh run"   11 minutes ago      Up 11 minutes       0.0.0.0:8082->8080/tcp   Ejercicio3
root@alumnodaw-VirtualBox:/home/alumnodaw# docker commit 78 ejercicio3tomcat
sha256:2173f286bc727d07208f480d9fa963e1273ab207ebcb9c6f20d110c692711872
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ejercicio3tomcat    latest              2173f286bc72        About a minute ago  667MB
tomcat              latest              08efef7ca980        4 hours ago         667MB
ubuntu              latest              4dd97cefde62        9 days ago          72.9MB
root@alumnodaw-VirtualBox:/home/alumnodaw#
```

Una vez ejecutada la imagen para ver si funciona, la subiremos a nuestro repositorio de DockerHub que hemos creado antes de empezar con la tarea. En mi caso, he seguido los pasos de un usuario de *Stackoverflow* que soluciona un problema que daba cuando trataba de publicar la imagen en mi repositorio. Los pasos que hay que seguir son:

1. Cierra y abre sesión desde la línea de comandos de docker:
   - "**docker logout**"
   - "**docker login**"
2. Incluye el namespace para que Docker Hub lo asocie con tu cuenta usando el tag:
   - "**docker tag NuevaImagen namespace/NuevaImagen**"
3. Sube la imagen al repositorio:
   - "**docker push namespace/NuevaImagen**"

```
root@alumnodaw-VirtualBox:/home/alumnodaw# docker tag ejercicio3tomcat hadrianvillarcuadrado/ejercicio3tomcat
root@alumnodaw-VirtualBox:/home/alumnodaw# docker push hadrianvillarcuadrado/ejercicio3tomcat
The push refers to repository [docker.io/hadrianvillarcuadrado/ejercicio3tomcat]
73c3c896d958: Pushed
34c7884ee125: Pushed
7e6c506447e9: Pushing [==========>                                        ]  4.226MB/20.25MB
6921406a2378: Pushed
7e6c506447e9: Pushing [========================>                          ]  10.26MB/20.25MB
7e6c506447e9: Pushing [==========================================>        ]  17.72MB/20.25MB
b219714e1f91: Pushing [======>                                            ]  43.21MB/342.1MB
b219714e1f91: Pushing [=========>                                         ]  65.78MB/342.1MB
a4ed737b0c8f: Pushed
59f1e8e1ce66: Pushing [==================================================>]  11.54MB
bde301416dd2: Pushing [========>                                          ]  23.59MB/145.5MB
81496d8c72c2: Pushing [============>                                      ]   4.44MB/17.54MB
644448d6e877: Waiting
0e41e5bdb921: Waiting
```

IMPORTANTE: *Los nombres de las imágenes deben de ser todo minúsculas o nos lanzará un error, "NuevaImagen", etc son solo ejemplos.*

Si todo ha salido correctamente se publicará el contenedor en la página web oficial de DockerHub bajo tu nombre como se muestra a continuación o clicando en el siguiente enlace.