

Práctica Docker 1

1. Comproba que funciona o “docker run hello-world”, que o que fai é saudar “Hello from Docker” e mostrar mensaxe. Comproba despois:
 - a) Con **‘docker ps’** comproba que o contedor non se está executando
 - b) Con **‘docker ps -a’** podes ver que existe. Comproba o seu nome e o ID.
 - c) Elimínoo con **‘docker rm idContedor’** ou **‘docker rm nomeContedor’**
2. Crea un contedor de debian, que se borre en canto remate (empregando o modificador **‘- -rm’** de docker run **–rm**), e mostra a versión de debian descargada (debe executar **‘cat /etc/issue’**).
3. Está correndo o contedor? Emprega **‘docker ps’** e compróboo. Existe no sistema? Compróboo con **‘docker ps -a’**. Comproba as imaxes que existen no sistema con **‘docker images’**
4. Como sería o comando para executar nese debian o comando **echo ‘Ola mundo’**?
5. Emprega o mesmo comando anterior para ver como se chama o contedor internamente (visualizar o ficheiro **/etc/hostname**).
6. O contedor perde todo o que se fai internamente nel se non se garda a información nun volume. Para comprobalo inicia unha **sesión interactiva** en debian, e instala nano:
 - a) apt update
 - b) apt install nano
 - c) nano /etc/hostname
 - d) Remata a sesión, volve facer unha sesión interactiva. Comproba que “nano /etc/hostname” non funciona, porque nano xa non está instalado.
7. Contedor en segundo plano (como un demo). O modificador **-d**. Executa agora o debian en segundo plano executando:
 - a) `$ docker run -d --name contedor debian bash -c "while true; do echo hello world; sleep 1; done"`
 - b) Comproba que o contedor se está executando
 - c) Comproba que está facendo o contedor con: **docker logs contedor**
 - d) Para o contedor con **docker stop contedor**, e bórrao con **docker rm contedor**.
8. Comproba que podes ter un contedor executándose coa orde dos apuntes, e outro con outro porto 9091:
 - a) `docker run -d -p 9090:80 --name apache-probas ubuntu/apache2`
Fai a comprobación conectándote a **localhost:9090**, e **localhost:9091**
9. A imaxe de apache para docker chámase **httpd**, que corre un debian moi lixeiro só co necesario para executar apache, e vale para construír imaxes personalizadas. Crea un contedor empregando httpd:2.4 no porto 9100. Comproba que funciona en localhost:9100.

TEMPO DE LIMPEZA. É tempo de parar e eliminar os contedores creados, para limpar o disco. Para isto:

- a) Para todos os contedores: executa **docker ps**, e vai facendo un **docker stop** de cada un deles
- b) Comproba cantos contedores existen parados: **docker ps -a**. Fai un **docker rm** de cada un deles.
- c) Teríamos agora que ver as imaxes que nos fomos baixando nesta práctica. Executa “**docker images**”, e verás as imaxes descargadas e o seu tamaño. Elimina as imaxes creadas nesta práctica con “**docker image rm ID_imaxe**”

10. Crea e inicia o contedor httpd, con **docker run --rm ...**, no porto 9000. Se vas á documentación da imaxe httpd de docker Hub, (https://hub.docker.com/_/httpd), podemos ver o DocumentRoot está en /usr/local/apache2/htdocs:

- a) Comproba que isto é así.
- b) Cambia o index.html. Podes Empregar o comando dentro do contedor **echo “<h1>En Docker! </h1>” > index.html** para sobreescribir o index.html
- c) Comproba que creamos ese contido de novo en localhost: 9000
- d) Comproba que este contido se perde se borras o contedor e o volves a iniciar.

11. A imaxe de mysql no docker Hub, chámase mysql (https://hub.docker.com/_/mysql). Podemos:

- a) iniciar un contedor: **docker run --name mysql -e MYSQL_ROOT_PASSWORD=abc123. -d mysql**
- b) Comproba que existe a variable de entorno con **docker exec -it mysql env**
- c) Podemos acceder ao contedor con **docker exec -it mysql bash**. Comproba que podes acceder como root co contrasinal indicado con **mysql -u root -p**
- d) Finalmente, elimina o contedor creado

12. Accederemos a mariadb desde desde o equipo real. Para iso podemos mapear o porto do contedor ao porto do equipo real:

- a) **docker run -d -p 3306:3306 --name mariadb -e MYSQL_ROOT_PASSWORD=Noia123. mariadb**
- b) Comproba que os portos están mapeados e que o contedor está correndo
- c) Deberías poder conectarte agora a localhost, con **mysql -u root -p -h 127.0.0.1**

13. Busca en <https://hub.docker.com/> a imaxe de nginx. Execútao no porto 8080 e comproba que te podes conectar desde o navegador.

14. Mira na páxina de https://hub.docker.com/_/nginx como terías que empregar un volume para mostrar un contido estático. Monta unha carpeta cun contido index.html, para que se mostre con nginx.