

## Práctica Docker 2. Imaxes. Persistencia

1. Comproba que imaxes de docker tes dispoñibles no sistema.
2. Queremos iniciar varios contedores, pero primeiro quero dispoñer das imaxes no sistema. Emprega “docker pull NOME\_imaxe”, e descarga as imaxes debian:latest, httpd, php:apache e alpine:latest. Compara o tamaño de ubuntu coa de alpine, unha imaxe mínima baseada en Linux Alpine
3. Crea un contedor demo coa imaxe php:apache, coa orde “docker run -d ... “. Lembra que tes que empregar o modificador -p (porto). Comproba, desde outra consola, o tamaño do contedor no disco (docker ps -a -s). Non pares o contedor php.
4. Conéctate co navegador a localhost. Debería dar un erro porque o contedor non ten un index.html en /var/www/html
5. Conéctate ao contedor con “docker exec... “ e crea un ficheiro *phpinfo.php* na túa carpeta:  

```
root@9c94a6221086:/var/www/html# echo "<? echo phpinfo() ?>" > index.php
```

  
Comproba que podes conectarte agora ao contedor desde o navegador.
6. Canto ocupa o contedor agora? Debes ver que o contedor ocupa agora os bytes engadidos no index.php e os MB (virtuais) da súa imaxe.
7. Borra as imaxes creadas antes de nada. Lembra que podes usar “**docker image rm nomeImaxe**” ou “**docker rmi nomeImaxe**”

### Volumes. Persistencia con volume -v:

8. Comproba que funciona o exemplo dos apuntes, dando persistencia ao debian, co exemplo de “docker run ... -v volumedebian ...”. Crea un ficheiro en /var/datos, con: *echo “texto novo” > proba.txt*. Borra o contedor, e comproba que podes volver a inicialo e o ficheiro está accesible no contedor en /var/datos
9. NextCloud é un software Open Source para alojar arquivos na nube. Segundo a documentación de NextCloud a forma máis sinxela de non perder información é crear un volume para gardar o directorio /var/www/html do contedor. Segundo isto:
  - a) Crea e inicia un contedor empregando a imaxe nextcloud, que expoña o porto 9002, e monte un volume “clouding”
  - b) Conéctate a localhost:9002, e crea un usuario en NextCloud. Sube un ficheiro ao servidor de ficheiros ou crea un novo desde a web
  - c) Comproba que está gardado
  - d) Para e elimina (se non empregaches o modificador - -rm) o contedor.

e) Crea e inicia de novo o contedor, e comproba que segue existindo o usuario e o ficheiro creado.

f) Borra a imaxe de nextcloud (que ocupa moito disco): `docker rmi nextcloud`

10. Inicia agora un contedor a partir da imaxe `httpd`, dando persistencia cun volume, sabendo que a carpeta que se exporta é a `/usr/local/apache2/htdocs`:

```
$docker run - - rm -d --name alpineapache -p 80:80 -v ...
```

Modifica o ficheiro `index.html`, e comproba que se paras, borras e reinicias o contedor o contido mantense. Para de novo o contedor, e bórrao e comproba que o volume segue existindo con:

```
$docker volume ls
```

**BORRADO:** É tempo de borrar os volumes que se crearon. Podes facelo con

```
$docker volume rm
```

### Volumes. Persistencia con bind mount

11. Crea unha carpeta `web`, comproba que vai o exemplo dos apuntes:

```
$docker run -d --name my-apache-app --mount  
type=bind,src=./web,dst=/usr/local/apache2/htdocs -p 8080:80 httpd:2.4
```

Crea un `index.html` na túa carpeta `web`. Comproba que se paras o contedor, e o creas de novo segue servindo o mesmo ficheiro.

12. Podemos montar a mesma carpeta en diferentes contedores. Por exemplo, se queremos comprobar se algún código pode funcionar en 2 proxectos de PHP que teñen diferente versión: un terá PHP8 e outro PHP7 (`str_contains` foi creado na versión 8). Empezaremos por ter o noso código nunha carpeta **webphp**. Crea un ficheiro **testversion.php** nesa carpeta `webphp` co seguinte contido:

```
<?php  
$texto = "calor";  
$frase = "Agora non vai calor";  
if (str_contains($frase, $texto) ) { echo "o texto está na frase"; }  
else {  
    echo "o texto non está na frase";  
}  
?>
```

Agora iniciamos un contedor con `php7` e outr con `PHP8`:

```
docker run -d -p 8081:80 --name php7 -v ~/webphp:/var/www/html:ro php:7.4-apache  
docker ... php:8.2-apache
```

Debes comprobar desde o navegador que o código é correcto na versión `php8`, pero dá un erro na versión 7.