

## Tarea Docker Compose

1.

- a. Docker Compose es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores.
- b. Permite diseñar y ejecutar aplicaciones multi-contenedor.
- c. `sudo curl -L https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose`

```
estudiante@DAW1:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
[sudo] password for estudiante:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
  0     0     0     0     0     0      0      0  --:--:-- --:--:-- --:--:--     0
100 11.6M  100 11.6M    0     0 5549k      0  0:00:02  0:00:02 --:--:-- 7854k
```

```
estudiante@DAW1:~$ sudo chmod +x /usr/local/bin/docker-compose
estudiante@DAW1:~$ docker-compose --version
docker-compose version 1.27.4, build 40524192
estudiante@DAW1:~$
```

2.

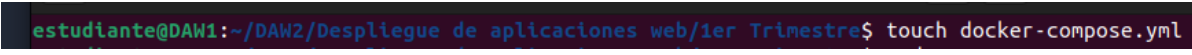
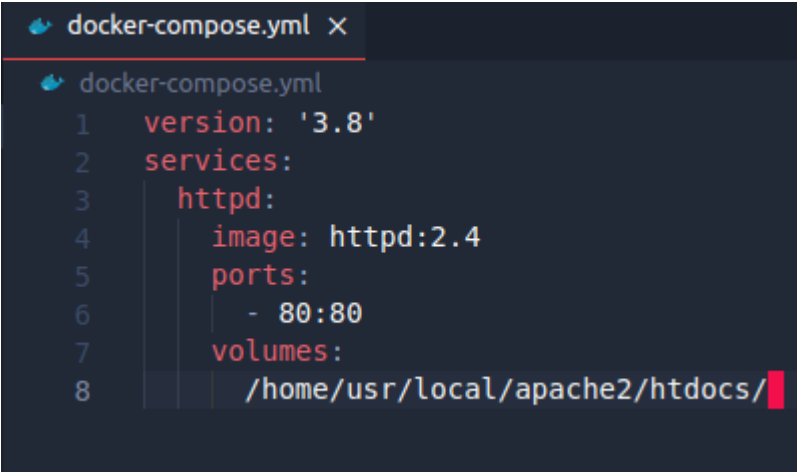
- a.
  - i. **docker-compose up**: Crear los contenedores (servicios) que están descritos en el docker-compose.yml.
  - ii. **docker-compose up -d**: Crear en modo detach los contenedores (servicios) que están descritos en el docker-compose.yml. Eso significa que no muestran mensajes de log en el terminal y que se nos vuelve a mostrar un prompt.
  - iii. **docker-compose stop**: Detiene los contenedores que previamente se han lanzado con docker-compose up.
  - iv. **docker-compose run**: Inicia los contenedores descritos en el docker-compose.yml que estén parados.
  - v. **docker-compose rm**: Borra los contenedores parados del escenario. Con las opción -f elimina también los contenedores en ejecución.
  - vi. **docker-compose pause**: Pausa los contenedores que previamente se han lanzado con docker-compose up.
  - vii. **docker-compose unpause**: Reanuda los contenedores que previamente se han pausado.
  - viii. **docker-compose restart**: Reinicia los contenedores. Orden ideal para reiniciar servicios con nuevas configuraciones.
  - ix. **docker-compose down**: Para los contenedores, los borra y también borra las redes que se han creado con docker-compose up (en caso de haberse creado).

- x. **docker-compose down -v:** Para los contenedores y borra contenedores, redes y volúmenes.
- xi. **docker-compose logs:** Muestra los logs de todos los servicios del escenario. Con el parámetro -fpodremos ir viendo los logs en “vivo”.
- xii. **docker-compose logs servicio1:** Muestra los logs del servicio llamado servicio1 que estaba descrito en el docker-compose.yml.
- xiii. **docker-compose exec servicio1 /bin/bash:** Ejecuta una orden, en este caso /bin/bash en un contenedor llamado servicio1 que estaba descrito en el docker-compose.yml
- xiv. **docker-compose build:** Ejecuta, si está indicado, el proceso de construcción de una imagen que va a ser usado en el docker-compose.yml a partir de los ficheros Dockerfile que se indican.
- xv. **docker-compose top:** Muestra los procesos que están ejecutándose en cada uno de los contenedores de los servicios.

3.

- a. Es un lenguaje de declaración de datos que facilita la legibilidad y la capacidad de escritura del usuario.
- b.
  - i. La sintaxis básica de este fichero indica que cada archivo YAML inicia con --- y su extensión es .yaml
  - ii. Permite datos como caracteres, cadenas, valores flotantes, números enteros, así como colecciones (por ejemplo matrices) y listas construidas partiendo de otros datos básicos.
  - iii. YAML solo admite espacios y tiene la capacidad de distinguir entre mayúsculas y minúsculas.
  - iv. Ejemplo  
---  
apiVersion: v1  
kind: Pod  
metadata:  
  name: rss-site  
  labels:  
    app: web

4.

- a. 
- b. 

```
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/1er Trimestre$ docker-compose up
Creating network "1ertrimestre_default" with the default driver
Pulling httpd (httpd:2.4)...
2.4: Pulling from library/httpd
Digest: sha256:4400fb49c9d7d218d3c8109ef721e0ec1f3897028a3004b098af587d565f4ae5
Status: Downloaded newer image for httpd:2.4
Creating 1ertrimestre_httpd_1 ... done
Attaching to 1ertrimestre_httpd_1
httpd_1 | AH00558: httpd: Could not reliably determine the server's fully qualified domain na
me, using 172.18.0.2. Set the 'ServerName' directive globally to suppress this message
httpd_1 | AH00558: httpd: Could not reliably determine the server's fully qualified domain na
me, using 172.18.0.2. Set the 'ServerName' directive globally to suppress this message
httpd_1 | [Fri Oct 14 07:38:46.020110 2022] [mpm_event:notice] [pid 1:tid 139789676715328] AH
00489: Apache/2.4.54 (Unix) configured -- resuming normal operations
httpd_1 | [Fri Oct 14 07:38:46.020262 2022] [core:notice] [pid 1:tid 139789676715328] AH00094
: Command line: 'httpd -D FOREGROUND'
^CGracefully stopping... (press Ctrl+C again to force)
Stopping 1ertrimestre_httpd_1 ... done
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/1er Trimestre$ docker compose ps
NAME                COMMAND                  SERVICE    STATUS    PORTS
1ertrimestre_httpd_1 "httpd-foreground"      httpd      exited (0)
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/1er Trimestre$
```

c.

5.

```
estudiante@DAW1:~/DAW2$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
117ea3ac8430   mysql:5.7.22 "docker-entrypoint.s..." 2 weeks ago Up 3 seconds 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp mysql
estudiante@DAW1:~/DAW2$
```

a.

b.

- i. Container id: El identificador del ID
- ii. Image: El nombre de la imagen y su versión
- iii. Command:
- iv. Created: Cuando fue creado el contenedor
- v. Status: El estado que se encuentra el contenedor
- vi. Ports: El puerto que usa el contenedor
- vii. Name: Nombre dado al contenedor (puede ser diferente al nombre original)

c.

- i. Docker ps: Muestra una lista de todos los contenedores activos.
- ii. Docker compose ps: Enumera contenedores para un proyecto de Compose, con el estado actual y los puertos expuestos. De forma predeterminada, se muestran tanto los contenedores en ejecución como los detenidos

6.

```
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/1er Trimestre$ cat docker-compose.yml
version: '3.8'
services:
  httpd:
    image: httpd:2.4
    ports:
      - 80:80
    volumes:
      - /home/usr/local/apache2/htdocs:/usr/local/apache2/htdocs
estudiante@DAW1:~/DAW2/Despliegue de aplicaciones web/1er Trimestre$
```

a.

b.

- i. version: indica la versión de fichero utilizado. Para poder utilizar stacks es necesario mínimo la versión 3.
- ii. services: indica los servicios contenerizados que se van a ejecutar
  1. httpd: el contenedor
  2. image: Indica el nombre de la imagen asociada al servicio.

- 3. ports: Permite mapear <puerto\_host> a <puerto\_contenedor>
- iii. volumes: Permite mapear elementos del sistema de archivos del interior del contenedor con elementos del sistema de archivos en el exterior