

Jose Antonio Castillejo Lobato

1

Es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores.

```
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 4081k    0  0:00:02  0:00:02 --:--:-- 5835k
estudiante@DAW1:~$ sudo chmod +x /usr/local/bin/docker-compose
chmod: missing operand after '+x/usr/local/bin/docker-compose'
Try 'chmod --help' for more information.
estudiante@DAW1:~$ sudo chmod +x /usr/local/bin/docker-compose
estudiante@DAW1:~$ docker-compose --version
docker-compose version 1.27.4, build 40524192
```

2

docker-compose up	Crear los contenedores
docker-compose up -d	Crear en modo detach los contenedores
docker-compose.yml	Eso significa que no muestran mensajes de log en el terminal y que se nos vuelve a mostrar un prompt.
docker-compose stop	Detiene los contenedores
docker-compose run	Inicia los contenedores
docker-compose rm	Borra los contenedores parados
docker-compose pause	Pausa los contenedores que previamente se han lanzado
docker-compose unpause	Reanuda los contenedores que previamente se han pausado.
docker-compose restart	Reinicia los contenedores
docker-compose down	Para los contenedores, los borra y también borra las redes que se han creado
docker-compose down -v	Para los contenedores y borra contenedores, redes y volúmenes
docker-compose logs	Muestra los logs de todos los servicios del escenario
docker-compose logs servicio1	Muestra los logs del servicio llamado servicio1

<code>docker-compose exec servicio1 /bin/bash</code>	Ejecuta una orden, en este caso <code>/bin/bash</code> en un contenedor llamado <code>servicio1</code> que estaba descrito en el <code>docker-compose.yml</code>
<code>docker-compose build</code>	Ejecuta, si está indicado,
<code>docker-compose top</code>	Muestra los procesos que están ejecutándose en cada uno de los contenedores de los servicios.

3

Es un lenguaje de declaración de datos que facilita la legibilidad y la capacidad de escritura del usuario

La estructura de los archivos puede ser un mapa o una lista.

Las secuencias de las listas empiezan con un guion (-) y un espacio, y se utiliza la sangría para separarlas del elemento principal, de manera similar a las listas de Python o los conjuntos en Bash o en Perl.

```
version: '3'

# Empieza la sección de servicios.

services:

# Declaro un servicio con nombre miapache

miapache:

# Ese contenedor usará como imagen de base la imagen httpd(Servidor Apache) de DockerHub.

image: httpd

# Le doy nombre al contenedor cuando arranque (equivalente al flag --name de docker un)

container_name: web
```

```
services:

# Declaro un servicio con nombre datos

datos:

# Ese contenedor usará como imagen de base la imagen mariadb (Servidor de base de datos) de DockerHub.

image: mariadb

# Le doy nombre al contenedor cuando arranque (equivalente al flag --name de docker un)

container_name: bd

# Establezco las variables de entorno para configurar el servicio

environment:

  MYSQL_ROOT_PASSWORD: 123456

  MYSQL_DATABASE: test

  MYSQL_USER: pepe

  MYSQL_PASSWORD: pepe
```

```
version: '3'

# Empieza la sección de servicios.

services:

  # Declaro un servicio con nombre miapache

  miapache:

    # Ese contenedor usará como imagen de base la imagen httpd(Servidor Apache) de DockerHub.

    image: httpd

    # Le doy nombre al contenedor cuando arranque (equivalente al flag --name de docker un)

    container_name: web

    # Establezco la redirección de puertos

    ports:

      - 8080:80
```

```
services:

  # Declaro un servicio con nombre miapache

  miapache:

    # Ese contenedor usará como imagen de base la imagen httpd(Servidor Apache) de DockerHub.

    image: httpd

    # Le doy nombre al contenedor cuando arranque (equivalente al flag --name de docker un)

    container_name: web

    # Establezco un bind bound de la carpeta src de mi equipo en la carpeta /app del contenedor.

    volumes:

      - "/src:/app"

    # Notación alternativa a lo anterior

    - type: bind

      source: "/src"

      target: /app
```

```
# Le doy nombre al contenedor cuando arranque (equivalente al flag --name de docker un)

container_name: bd

.....

# Establezco que los datos de la base de datos van a persistir en el volumen datosapp que se montará en /var/lib/mysql

volumes:

  - "datosapp:/var/lib/mysql"

# Notación alternativa a lo anterior

volumes:

  - type: volume

    src: datosapp

    target: "/var/lib/mysql"

# Sección para la definición de los volúmenes. Está al mismo nivel de la sección services

volumes:

  datosapp: local
```

```

networks:
  - ejemplo
...
# Sección de definición de redes. Está al mismo nivel que services y volumes
networks:
  # Definición de la red ejemplo
  ejemplo:
    # Tipo de red
    driver: bridge

    # Opciones de la red

    ipam:
      driver: default
      config:
        subnet: 172.20.0.0/16

```

```

services:

  # Declaro un servicio con nombre miapache

  miapache:

    # Ese contenedor usará como imagen de base la imagen httpd(Servidor Apache) de DockerHub.
    image: httpd

    ....

    # Este contenedor arrancará después el contenedor del servicio datos

    depends_on:
      - datos

  # Comienzo de descripción del servicio de datos

  datos:

    image: mariadb

    ....

```

```

estudiante@DAW1:~$ touch docker-compose.yml
estudiante@DAW1:~$ code .
estudiante@DAW1:~$ docker-compose up
ERROR: The Compose file './docker-compose.yml' is invalid because:
Invalid top-level property "container_name". Valid top-level sections for this Compose file are: version, services, networks, volumes, secrets, configs, and extensions starting with "x-".

You might be seeing this error because you're using the wrong Compose file version. Either specify a supported version (e.g. "2.2" or "3.3") and place your service definitions under the 'services' key, or omit the 'version' key and place your service definitions at the root of the file to use version 1.
For more on the Compose file format versions, see https://docs.docker.com/compose/compose-file/
estudiante@DAW1:~$ docker-compose up
ERROR: The Compose file './docker-compose.yml' is invalid because:
Unsupported config option for services.httpd: 'volumens'
estudiante@DAW1:~$ docker-compose up
ERROR: The Compose file './docker-compose.yml' is invalid because:
Unsupported config option for services.httpd: 'volumens'
estudiante@DAW1:~$ docker-compose up
Creating network "estudiante_default" with the default driver
Pulling httpd (httpd:2.4)...
2.4: Pulling from library/httpd
bd159e379b3b: Pull complete
36d838c2f6d6: Pull complete
b55eda22bb18: Pull complete
f6e6bfa28393: Pull complete
a1b49b7ecb8a: Pull complete
Digest: sha256:4400fb49c9d7d218d3c8109ef721e0ec1f3897028a3004b098af587d565f4ae5
Status: Downloaded newer image for httpd:2.4
Creating estudiante_httpd_1 ... done
Attaching to estudiante_httpd_1
httpd_1 | AH00558: httpd: Could not reliably determine the server's fully qualified domain name, 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 name, using 172.18.0.2. Set the 'ServerName' directive globally to suppress this message
httpd_1 | [Fri Oct 14 07:04:30.869101 2022] [mpm_event:notice] [pid 1:tid 139899503820096] AH00489: Apache/2.4.54 (Unix) configured -- resuming normal operations
httpd_1 | [Fri Oct 14 07:04:30.869251 2022] [core:notice] [pid 1:tid 139899503820096] AH00094: Command line: 'httpd -D FOREGROUND'

```

```

docker-compose.yml
1  version: '3.8'
2  services:
3    httpd:
4      image: httpd:2.4
5      ports:
6        - 80:80
7      volumens:
8        hombe|usr/local/apache2/htdocs/

```

5

```

estudiante@DAW1:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
c8f21232b96d   mysql:5.7.22   "docker-entrypoint.s..." 2 weeks ago   Up 2 seconds
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp   mysql

estudiante@DAW1:~$ docker compose ps
NAME           COMMAND                  SERVICE    STATUS
estudiante_httpd_1  "httpd-foreground"      httpd      exited (0)

estudiante@DAW1:~$ docker-compose.yml
docker-compose.yml: command not found
estudiante@DAW1:~$

```

Nos muestra los contenedores con su id, el nombre y versión de la imagen, la fecha de creación y su estado actual.

La diferencia entre docker ps y docker compose ps es que el primero te muestra los contenedores que tienes en marcha y el otro te muestra los ficheros de docker compose

6

```
estudiante@DAW1:~$ cat docker-compose.yml
version: '3.8'
services:
  httpd:
    image: httpd:2.4
    ports:
      - 80:80
    volumes:
      - hombre/usr/local/apache2/htdocs/estudiante@DAW1:~$
```

Nos muestra la versión que es 3.8, httpd image nos muestra la imagen y su versión, el ports muestra los puertos y los volúmenes nos muestra la ruta