

Actividad 11-10-2022

1. ¿Que es Docker Compose? ¿Qué utilidades tiene? Instálalo y comprueba que está instalado
Docker Compose es una herramienta para definir y ejecutar aplicaciones de Docker de varios contenedores. El archivo Compose proporciona una forma de documentar y configurar todas las dependencias de servicio de la aplicación.

```
estudiante@DAW1:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
[sudo] password for estudiante:
Sorry, try again.
[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 2588k      0  0:00:04 0:00:04 --:--:-- 2970k
estudiante@DAW1:~$ sudo chmod +x /usr/local/bin/docker-compose
estudiante@DAW1:~$ docker-compose --version
docker-compose version 1.26.0, build d4451659
estudiante@DAW1:~$
```

2. Haz un glosario de comandos de Docker Compose

docker-compose build

Se usa para generar la imagen, basada en las especificaciones del servicio en el docker-compose.

docker-compose up

En caso de no existir la imagen se arma la misma, y siempre se crea el container y luego se inicia.

docker-compose start

Se inicia un container existente (frenado mediante el comando de stop).

docker-compose stop

Frena un container, sin eliminarlo, para luego poder iniciarse nuevamente con start.

docker-compose down

Se utiliza para frenar el/los container/s, eliminarlos.

docker-compose ps

Lista los containers corriendo en el contexto de este docker-compose.

3. ¿Qué es un fichero YAML? ¿Qué estructura tiene? Investiga y detállalo. Haz también un glosario de YAML.

El fichero YAML es un lenguaje de declaración de datos que facilita la legibilidad y la capacidad de escritura del usuario.

Es fundamental que se deje un espacio entre los indicadores.

Este formato de serialización también permite datos como caracteres, cadenas, valores flotantes, números enteros, así como colecciones (por ejemplo matrices) y listas construidas partiendo de otros datos básicos.

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx
5    labels:
6      app: nginx
7  spec:
8    volumes:
9      - name: negu-volume
10        hostPath:
11          path: /mnt/web
12          type: Directory
13    containers:
14      - name: front
15        image: nginx:latest
16        volumeMounts:
17          - mountPath: "/usr/share/nginx/html"
18            name: negu-volume
19
20  apiVersion: v1
21  kind: Service
22  metadata:
23    name: nginx
24  spec:
25    type: LoadBalancer
26    selector:
27      app: nginx
28    ports:
29      - protocol: TCP
30        port: 80
31        targetPort: 80
32
33  apiVersion: batch/v1beta1
34  kind: CronJob
35  metadata:
36    name: nginx
37  spec:
38    schedule: "*/1 * * * *"
39    jobTemplate:
40      spec:
41        template:
42          spec:
43            volumes:
44              - name: negu-volume
45                hostPath:
46                  path: /mnt/web
47                  type: Directory
48            containers:
49              - name: nginx
50                image: ubuntu:latest
51                command: ["/bin/sh"]
52                args: ["-c", "echo '.' >> /usr/share/nginx/html/index.html"]
53            volumeMounts:
54              - mountPath: "/usr/share/nginx/html"
55                name: negu-volume
56    restartPolicy: OnFailure
```

4. Crea un archivo de configuración para una aplicación que contiene un único servicio:
- Imagen: httpd:2.4 (del servidor web Apache)
 - Puerto: el host de Docker publicará el puerto 80 y hará una redirección con el puerto 80 del contenedor.
 - Bind mount: lo creamos entre el directorio actual del host de Docker y el directorio `usr/local/apache2/htdocs/` del contenedor (que es el directorio que utiliza el servidor web para servir el contenido que encuentre en su interior).

```
GNU nano 6.2 ejemplo.yml *
version: "3.9"
services:
  ports:
    - "80:80"
  volumes:
    - .:usr/local/apache2/htdocs/
  apache:
    image: "httpd:2.4"
```

5. Consulta la lista de contenedores que están en ejecución y explica la salida. ¿Qué diferencia hay entre “`docker ps` y `docker compose ps`”?

```
estudiante@DAW1:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
estudiante@DAW1:~$ docker compose ps
no configuration file provided: not found
```

Docker ps enumera todos los contenedores en ejecución y docker compose ps enumera los contenedores relacionados con las imágenes declaradas en docker-compose file.

6. Muestra la salida de docker-compose.yml

```
estudiante@DAW1:~$ docker-compose up
ERROR: yaml.parser.ParserError: while parsing a block mapping
in "./docker-compose.yml", line 1, column 1
expected <block end>, but found '<block mapping start>'
in "./docker-compose.yml", line 7, column 3
```

7. Visiona el video relativo a docker compose del módulo 7.4.
Visto.