

Docker Compose

Arquitectura de software

NRC: 10-66167

Uniminuto Zipaquirá

Prof. William Alexander Matallana Porras

Luis Alejandro Alarcón Chaves

Id: 792825

20 de marzo del 2025

- **ARCHIVOS .YML VS JSON.**

Características.	.YML(YAML)	JSON(.json)
Sintaxis	Utiliza indentación basada en espacio para definir estructuras, lo que lo hace mas legible.	Emplea una estructura basada en llaves [], corchetes {} y comillas “”, similar a la sintaxis de JavaScript.
Legibilidad	YML es considerado mas legible y fácil de escribir para personas debido a su simplicidad y claridad.	Aunque legible, puede ser mas verboso y menos intuitivo en estructuras complejas.
Soporte de comentarios	Permite la inclusión de comentarios utilizando el símbolo #.	No admite comentarios de forma nativa, lo que puede limitar la documentación dentro del archivo.
Tipos de datos	Admite una amplia gama de tipos de datos, incluyendo cadenas, números, listas, diccionarios, booleanos y null.	En este caso es similar a YAML, soporta cadenas, números, matrices(listas), objetos (diccionarios), booleanos y null.
Uso común	Frecuentemente utilizado en archivos de configuración y herramientas de automatización debido a su legibilidad.	Predominante en el intercambio de datos en aplicaciones web y API's por su simplicidad y amplia adopción
Compatibilidad	Aunque es mas flexible, puede ser mas propenso a errores si la indentación no es correcta.	Su estructura estricta reduce la posibilidad de errores de sintaxis.
Soporte de bibliotecas	Dispones de bibliotecas en múltiples lenguajes, aunque su soporte puede ser menos extenso que JSON.	Amplio soporte en prácticamente todos los lenguajes de programación modernos.

- **DOCKER-COMPOSE.YML**

Docker Compose es una herramienta versátil que te permite definir y gestionar aplicaciones multi-contenedor de forma sencilla. Con Docker Compose, puedes describir la configuración de tu entorno de desarrollo en un archivo YAML, especificando los servicios, volúmenes y redes necesarios para tu aplicación. Luego, con un solo comando, puedes crear y ejecutar todos los contenedores definidos en tu archivo de configuración.

Parámetros y opciones comunes en Docker Compose

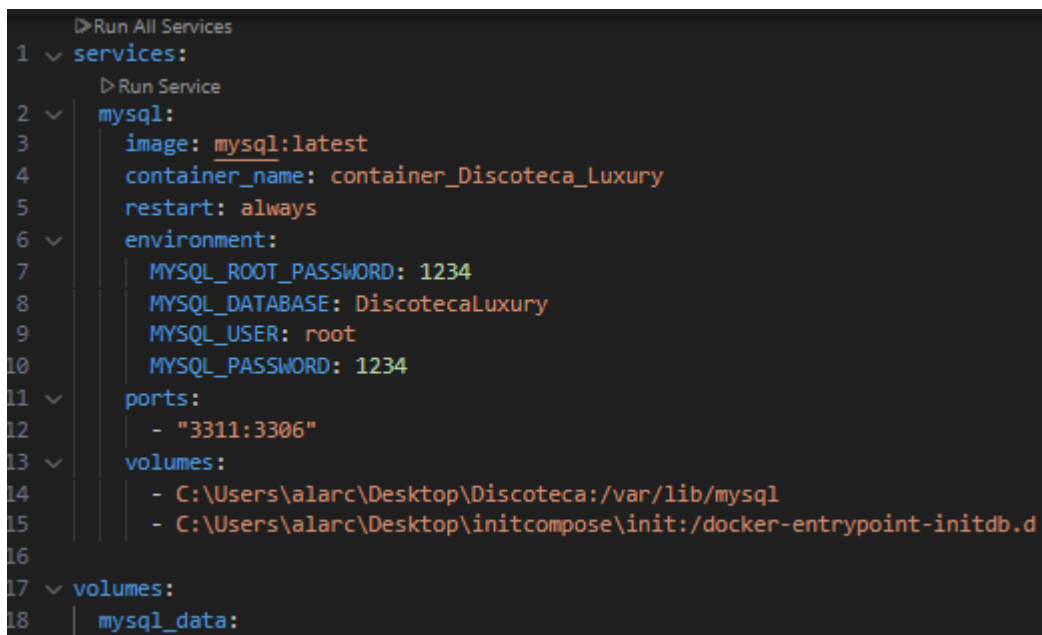
En el archivo docker-compose.yml, podemos utilizar varios parámetros y opciones para configurar nuestros servicios y contenedores.

- `image`: especifica la imagen del contenedor a utilizar.
- `ports`: mapea los puertos del contenedor a puertos de la máquina anfitriona.
- `volumes`: monta volúmenes o directorios del sistema de archivos en el contenedor.
- `environment`: define variables de entorno para el contenedor.
- `depends_on`: especifica las dependencias entre servicios.

Docker Compose utiliza un **archivo YAML** para definir los servicios, redes y volúmenes que necesita una aplicación. Su sintaxis clara y estructurada facilita la gestión de entornos.

```
version: '3.8'
services:
  web:
    image: nginx
    ports:
      - "80:80"
  db:
    image: postgres
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: secret
```

• COMO CREAR UN CONTENEDOR CON UN ARCHIVO.YML



```

1  services:
2    mysql:
3      image: mysql:latest
4      container_name: container_Discoteca_Luxury
5      restart: always
6      environment:
7        MYSQL_ROOT_PASSWORD: 1234
8        MYSQL_DATABASE: DiscotecaLuxury
9        MYSQL_USER: root
10       MYSQL_PASSWORD: 1234
11     ports:
12       - "3311:3306"
13     volumes:
14       - C:\Users\alarc\Desktop\Discoteca:/var/lib/mysql
15       - C:\Users\alarc\Desktop\initcompose\init:/docker-entrypoint-initdb.d
16
17  volumes:
18    mysql_data:

```

Para crear un contenedor con el docker-compose debemos usar la siguiente código en el cual vamos a indicar todos los parámetros que tendrá nuestro contenedor y donde creara su respectivo volumen y la estructura de la base de datos.

- Crear el diseño de la base de datos.

```
C:\Users\> alarc\Desktop> initcompose > init.sql
1 CREATE TABLE Clientes(
2     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
3     nombre VARCHAR(50),
4     apellido VARCHAR (50),
5     correo_electronico VARCHAR(100),
6     telefono VARCHAR(20)
7 );
8
9 CREATE TABLE Reservas(
10     id_reserva INT AUTO_INCREMENT PRIMARY KEY,
11     id_cliente INT,
12     fecha_reserva DATETIME,
13     numero_personas INT,
14     estado ENUM ('pendiente', 'confirmada', 'cancelada'),
15     FOREIGN KEY(id_cliente) REFERENCES Clientes(id_cliente)
16 );
17
18 CREATE TABLE Empleados(
19     id_empleado INT
20     AUTO_INCREMENT PRIMARY KEY,
21     nombre VARCHAR(50),
22     apellido VARCHAR(50),
23     puesto VARCHAR(50),
24     salario DECIMAL(10, 2)
25 );
26
27 CREATE TABLE Inventario(
28     id_producto INT
29     AUTO_INCREMENT PRIMARY KEY,
30     nombre_producto VARCHAR(100),
31     cantidad INT,
32     precio DECIMAL(10, 2)
33 );
34
35 CREATE TABLE Ventas(
36     id_venta INT
37     AUTO_INCREMENT PRIMARY KEY,
38     id_producto INT,
39     fecha_venta DATETIME,
```

Ahora vamos a crear nuestra base de datos que va a tener nuestro contendor.

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage ⓘ

No containers are running.

Container memory usage ⓘ

No containers are running.

Show charts

☰

☒ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> BD_Arquitectura	dbad829eb46f	mysql:lates	3310:3306	N/A	3 days ago	▶ ⋮ 🗑
<input type="checkbox"/>	<input type="radio"/> Contenedor	8a9d4919cc02	mysql:lates	3308:3306	N/A	9 days ago	▶ ⋮ 🗑
<input type="checkbox"/>	<input checked="" type="radio"/> > <input checked="" type="radio"/> dockercompose	-	-	-	N/A	2 seconds ago	▶ ⋮ 🗑

Y después revisamos se creo el contenedor.

Referencias

Isaac. (2025b, marzo 15). *Guía completa para usar Docker Compose*. Mundobytes. <https://mundobytes.com/usar-docker-compose/>

De Imagina, E. (2025, 20 marzo). *Qué es Docker Compose y Cómo Usarlo | Tutorial Completo*. Imagina Formación. <https://imaginaformacion.com/tutoriales/que-es-docker-compose>