

Docker compose.yml

Javier Mauricio Montenegro

Ingeniería de sistemas, séptimo semestre, Corporación Universitaria

Minuto de Dios

NRC 66167: Arquitectura de software

Ing. William Alexander Matallana Porras

20 de mar. de 25

Documentación

Archivos. yml VS json

YAML y JSON son dos formatos de serialización de datos que proporcionan un mecanismo de intercambio de datos legible por humanos. Dan formato a los datos de manera estandarizada para su intercambio entre aplicaciones de software, y su texto también es legible por humanos. Ambos representan datos como pares de clave y valor. JSON admite objetos de datos como valores, mientras que YAML no. Sin embargo, YAML admite más tipos de datos y utiliza de forma más próxima el lenguaje natural para que los desarrolladores puedan utilizarlo.

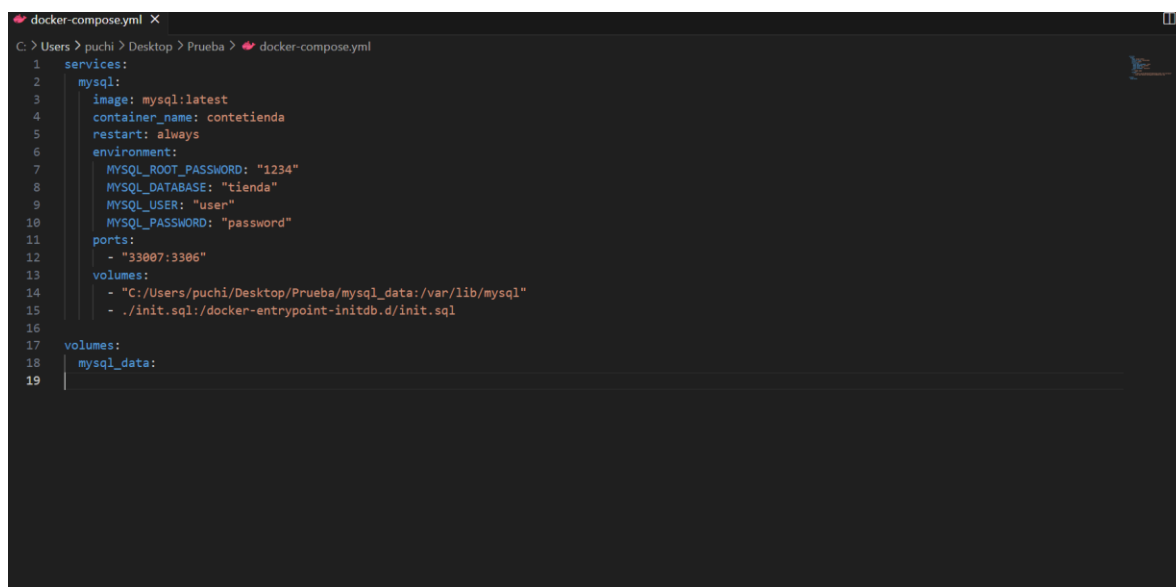
Resumen de las diferencias entre YAML y JSON

	JSON	YAML
¿Qué es?	Un formato de serialización de datos para intercambiar datos estructurados entre aplicaciones y servicios de software. Da prioridad al uso de las aplicaciones antes que al uso humano.	Un formato de serialización de datos para intercambiar datos estructurados entre aplicaciones y servicios de software. Da prioridad al uso humano antes que al de las aplicaciones.
Principales casos de uso	Ampliamente extendido en plataformas, lenguajes, comunicaciones de software distribuido, aplicaciones web, archivos de configuración y API.	Archivos de configuración en muchas herramientas y servicios de automatización, DevOps e infraestructura como código (IaC).
Legibilidad	Fácil.	Lo más fácil.
Tipos de datos	Número, booleano, nulo, cadena, matriz y objeto.	Admite todos los tipos de datos mediante colecciones de datos anidados que comprenden secuencias, escalares y asignaciones.
Admite comentarios	No.	Sí.
Admite objetos de datos como valores	Sí.	No.
Control de versiones	Sí, pero no es tan sencillo analizar y comprender de un solo vistazo las diferencias entre las versiones.	Sí, y es fácil analizar y comprender de un solo vistazo las diferencias entre las versiones.

Docker- compose.yml-uso

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

Como se crea un contenedor



```
1 services:
2   mysql:
3     image: mysql:latest
4     container_name: contetienda
5     restart: always
6     environment:
7       MYSQL_ROOT_PASSWORD: "1234"
8       MYSQL_DATABASE: "tienda"
9       MYSQL_USER: "user"
10      MYSQL_PASSWORD: "password"
11     ports:
12       - "33007:3306"
13     volumes:
14       - "C:/Users/puchi/Desktop/Prueba/mysql_data:/var/lib/mysql"
15       - ./init.sql:/docker-entrypoint-initdb.d/init.sql
16
17 volumes:
18   mysql_data:
```

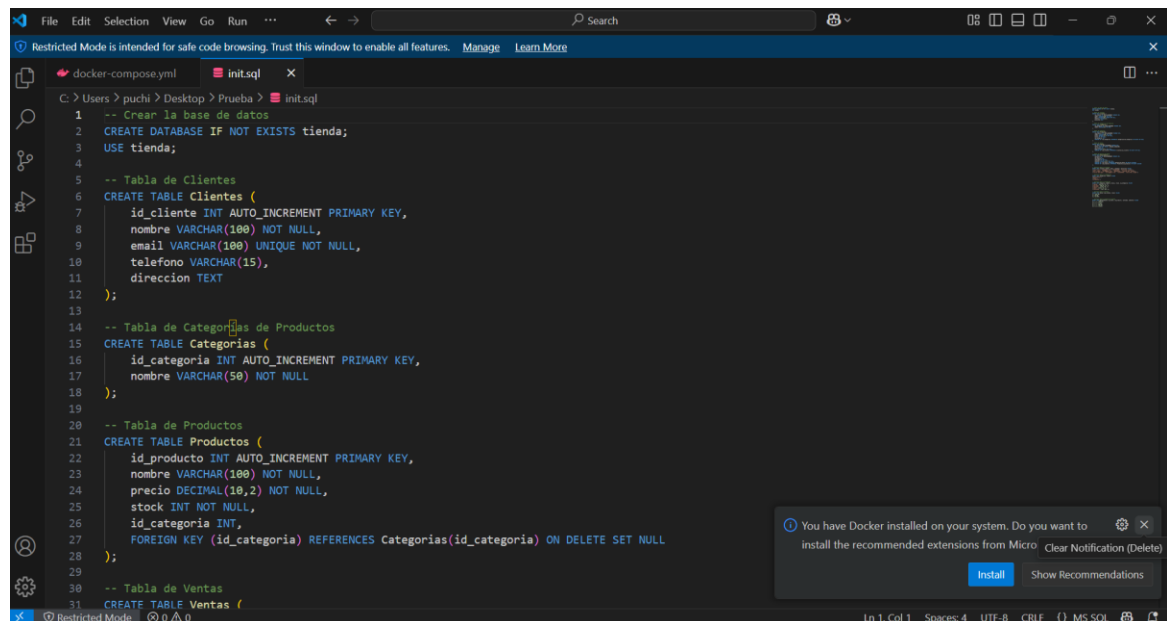
Este archivo de configuración de Docker Compose define un servicio MySQL con varias especificaciones. El servicio utiliza la imagen más reciente de MySQL y crea un contenedor llamado *contetienda* que siempre se reiniciará automáticamente si se detiene. Se configuran variables de entorno clave: la contraseña del usuario root (1234), una base de datos llamada *tienda*, un usuario personalizado (*user*) con su contraseña (*password*). El puerto 3306 del contenedor se expone en el puerto 33007 del host, permitiendo conexiones externas a MySQL.

Para la persistencia de datos, se monta el directorio C:/Users/puchi/Desktop/Prueba/mysql_data del host en /var/lib/mysql del contenedor, asegurando que los datos sobrevivan al reinicio del contenedor. Adicionalmente, el archivo init.sql del host se monta en /docker-entrypoint-initdb.d/init.sql, lo que ejecutará

automáticamente este script SQL al iniciar la base de datos por primera vez (creando tablas, insertando datos iniciales, etc.).

Aunque se declara un volumen llamado `mysql_data` al final, no está siendo utilizado por el servicio, ya que este usa directamente la ruta del host. Este archivo garantiza una configuración reproducible de MySQL con personalización de acceso, red, almacenamiento e inicialización automática de esquemas.

Creación de las bases de datos



```
1 -- Crear la base de datos
2 CREATE DATABASE IF NOT EXISTS tienda;
3 USE tienda;
4
5 -- Tabla de Clientes
6 CREATE TABLE Clientes (
7     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
8     nombre VARCHAR(100) NOT NULL,
9     email VARCHAR(100) UNIQUE NOT NULL,
10    telefono VARCHAR(15),
11    direccion TEXT
12 );
13
14 -- Tabla de Categorías de Productos
15 CREATE TABLE Categorías (
16     id_categoria INT AUTO_INCREMENT PRIMARY KEY,
17     nombre VARCHAR(50) NOT NULL
18 );
19
20 -- Tabla de Productos
21 CREATE TABLE Productos (
22     id_producto INT AUTO_INCREMENT PRIMARY KEY,
23     nombre VARCHAR(100) NOT NULL,
24     precio DECIMAL(10,2) NOT NULL,
25     stock INT NOT NULL,
26     id_categoria INT,
27     FOREIGN KEY (id_categoria) REFERENCES Categorías(id_categoria) ON DELETE SET NULL
28 );
29
30 -- Tabla de Ventas
31 CREATE TABLE Ventas (
```

The image shows a code editor window with a dark theme. The file explorer on the left shows a project structure with folders like 'docker-compose.yml' and 'init.sql'. The main editor area displays a MySQL script. A notification bubble in the bottom right corner states: 'You have Docker installed on your system. Do you want to install the recommended extensions from Micro?' with buttons for 'Install', 'Show Recommendations', and 'Clear Notification (Delete)'. The status bar at the bottom indicates 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'MS SQL'.

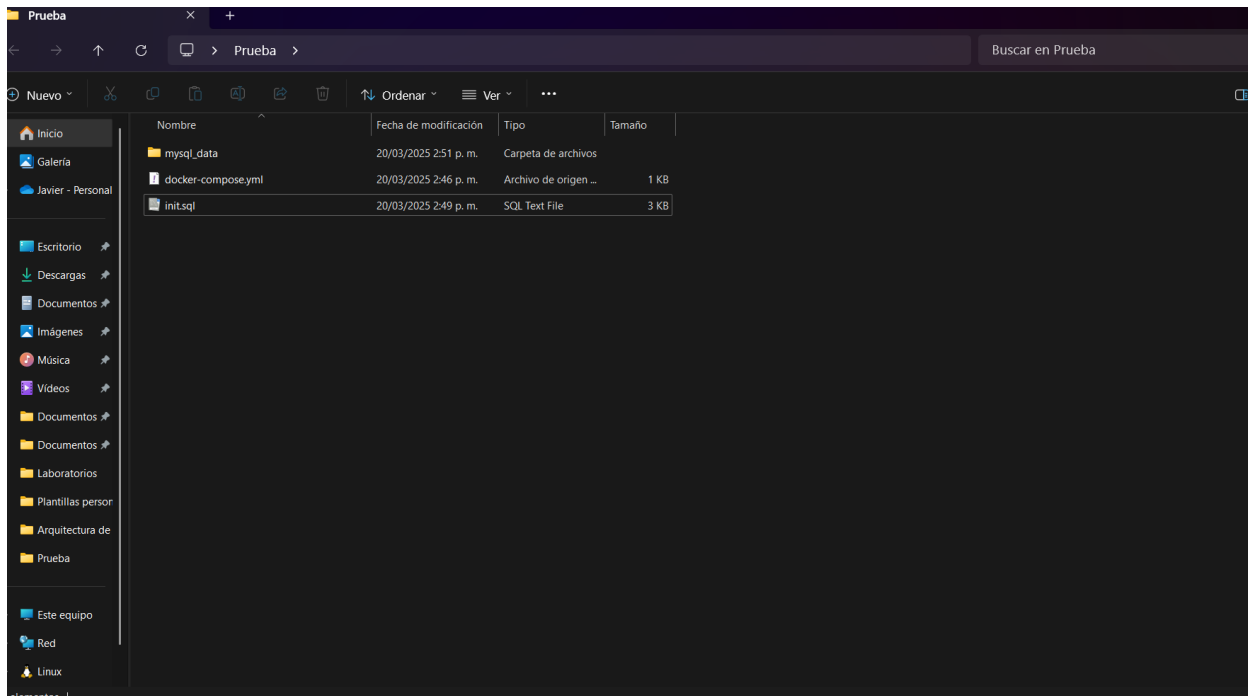
Comandos necesarios para el proceso

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\puchi\Desktop\Prueba> docker-compose up -d
[+] Running 2/2
  ✓ Network prueba_default    Created          0.2s
  ✓ Container contetienda     Started         0.3s
PS C:\Users\puchi\Desktop\Prueba> docker ps
```

Si el proceso funciono y todo quedo bien creado nos debería crear una carpeta nueva:



Por último para revisar que las tablas y las inserciones de datos fue correcta ejecutamos lo siguiente

```
PS C:\Users\puchi\Desktop\Prueba> docker start contetienda
contetienda
PS C:\Users\puchi\Desktop\Prueba>
PS C:\Users\puchi\Desktop\Prueba> docker exec -it contetienda mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

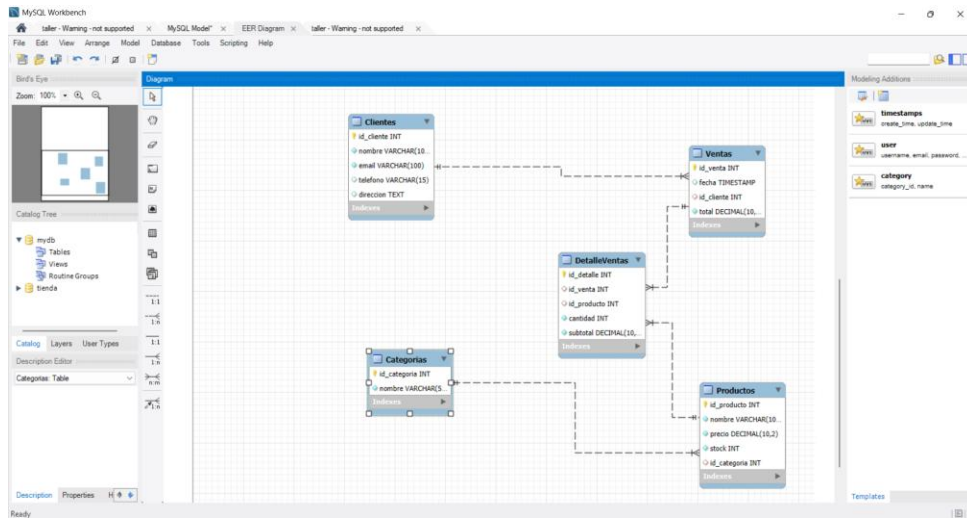
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE taller;
ERROR 1049 (42000): Unknown database 'taller'
mysql> USE TALLER;
ERROR 1049 (42000): Unknown database 'taller'
mysql> USE tienda;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_tienda |
```

```
mysql> SELECT * FROM Categorías;
+-----+-----+
| id_categoria | nombre |
+-----+-----+
| 1 | Electrónica |
| 2 | Ropa |
| 3 | Alimentos |
+-----+-----+
3 rows in set (0.00 sec)
```

Ingeniería inversa



}

Referencias

- Amazon Web Services. (n.d.). ¿Qué diferencias hay entre YAML y JSON?. AWS. <https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>
- Docker, Inc.. (n.d.). Compose documentation. Docker Docs. <https://docs.docker.com/compose/>