Taller Compose

Cristian Esteban Cano Vega ID:824426

Corporación Universitaria Minuto de Dios

Ingeniería de Sistemas, 7° Semestre

Arquitectura de Software NRC:66167

Pr. William Alexander Matallana Porras

20 de marzo de 2025

YML vs JSON

Dentro del mundo de los lenguajes de programación, es común considerar a JSON (JavaScript Object Notation) como el formato ideal para la serialización de datos. Su amplia adopción y estructura lo hacen una opción confiable para la transmisión segura de información.

Sin embargo, la comparación entre JSON y YAML (YAML Ain't Markup Language) es más compleja. Aunque JSON cuenta con una gran popularidad, YAML puede resultar más adecuado en ciertos casos, dependiendo de factores como el tipo de datos y si el objetivo principal es el almacenamiento o la transferencia de información.

La elección entre ambos formatos dependerá del propósito específico y de quién los utilizará.

¿Qué es JSON?

JSON es un formato ligero para intercambiar datos entre sistemas. Su estructura basada en texto y su sintaxis similar a JavaScript lo hacen fácil de leer y escribir tanto para humanos como para máquinas.

Este formato de serialización de datos surgió alrededor del mismo tiempo que YAML, en 2001, y se popularizó como una alternativa más simple y liviana en comparación con XML, un formato más antiguo utilizado para el almacenamiento y transferencia de datos.

Actualmente, JSON es considerado un estándar en la industria y es el formato más utilizado para la transmisión de datos. Sin embargo, no es adecuado para todas las situaciones, ya que está diseñado principalmente para serializar y transferir datos estructurados a través de redes, más que para almacenamiento.

Es un formato seguro que maneja números, cadenas, objetos y matrices. No obstante, si se requiere trabajar con tipos de datos más complejos, como valores anidados o marcas de tiempo, puede ser necesario considerar otros formatos.

¿Qué es YAML?

YAML es un formato de serialización de datos conocido por su alta legibilidad. Muchos desarrolladores lo encuentran más intuitivo que JSON, ya que su sintaxis es más cercana al lenguaje natural. De hecho, YAML es un superconjunto de JSON, lo que significa que puede representar cualquier estructura de datos que JSON maneje, pero con mayor flexibilidad.

Su sintaxis es similar a Python, lo que facilita su adopción para equipos familiarizados con este lenguaje. YAML se emplea en archivos de configuración y en aplicaciones donde se requiere la transferencia de datos, pero a diferencia de JSON, también es útil para almacenar información.

Una de sus principales ventajas es la capacidad de manejar una mayor variedad de tipos de datos, incluyendo fechas, marcas de tiempo, secuencias, valores anidados, valores nulos y booleanos. Aunque ofrece más funcionalidades que JSON, este último sigue siendo más seguro en ciertos casos, por lo que la elección entre ambos dependerá de las necesidades específicas de cada proyecto.

Casos de uso de JSON

JSON es ampliamente reconocido como un estándar en la serialización de datos. Puede ser la mejor opción si buscas un formato que:

Sea fácil de aprender y adoptar: Su sintaxis es similar a JavaScript, lo que facilita su aprendizaje para desarrolladores familiarizados con este lenguaje. En contraste, YAML tiene una sintaxis diferente que puede requerir más tiempo de adaptación.

Permita una validación rápida y sencilla: JSON es más tolerante con errores menores en comparación con YAML, que depende del espaciado y la indentación, lo que puede hacer que un pequeño error estructural genere problemas.

Sea compacto y eficiente: JSON es más ligero y rápido de analizar que YAML, lo que mejora el rendimiento en entornos donde la velocidad es crucial.

Ofrezca mayor seguridad: JSON es menos propenso a ciertas vulnerabilidades de seguridad en comparación con YAML, lo que lo convierte en una opción más segura en entornos donde la protección de datos es prioritaria.

Facilite la integración con otros sistemas: JSON es ampliamente compatible con diversas tecnologías, lo que lo hace ideal para el intercambio de datos entre aplicaciones.

Sea ideal para el intercambio de datos en API: JSON es el formato más utilizado en APIs, permitiendo la comunicación entre clientes (como aplicaciones web o móviles) y servidores de manera eficiente.

Casos de uso de YAML

Aunque JSON tiene muchas ventajas, YAML puede ser una mejor opción en ciertos escenarios, especialmente si necesitas un formato de serialización de datos que:

Maneje múltiples tareas: YAML es ideal para trabajar con distintos tipos de datos, soporta comentarios y es útil tanto para almacenar como para transferir información.

Sea compatible con una mayor variedad de datos: YAML permite trabajar con enteros, flotantes y estructuras de datos más complejas, como listas y matrices asociativas, que JSON no maneja tan fácilmente.

Permita agregar comentarios: A diferencia de JSON, YAML admite comentarios dentro del código, lo que facilita la documentación y mantenimiento.

Utilice una sintaxis más natural: YAML está diseñado para ser legible y fácil de entender, lo que lo hace accesible para principiantes y mejora la claridad del código.

Sea ideal para configuraciones de API: YAML es ampliamente utilizado en la configuración de servicios y APIs, definiendo endpoints, métodos de autenticación y esquemas de datos.

Si buscas un formato más flexible y fácil de leer que pueda manejar diversos tipos de datos y configuraciones, YAML es una excelente opción.

Docker-compose.yml

Docker Compose es una herramienta que permite definir y administrar contenedores de Docker mediante un archivo YAML, facilitando la configuración y ejecución de múltiples servicios en un solo comando. En este caso, el archivo define un servicio para MySQL, especificando la imagen a utilizar, las credenciales de acceso y las configuraciones necesarias para su ejecución.

El servicio MySQL se ejecuta dentro de un contenedor llamado "Tienda" y se reinicia automáticamente en caso de fallos. Se establecen variables de entorno que configuran la base de

datos inicial, el usuario y su contraseña. Además, el puerto 3306 del contenedor se expone en el host en el puerto 3310, permitiendo la conexión desde fuera del contenedor.

El sistema de volúmenes se usa para la persistencia de datos. Se monta un volumen en la ruta C:/Users/canox/OneDrive/Escritorio/Prueba/mysql_data dentro del contenedor en /var/lib/mysql, asegurando que los datos de la base de datos no se pierdan tras detener el contenedor. También se monta un archivo SQL (base.sql) en la carpeta docker-entrypoint-initdb.d/, lo que permite ejecutar automáticamente sentencias de inicialización cuando el contenedor se crea por primera vez.

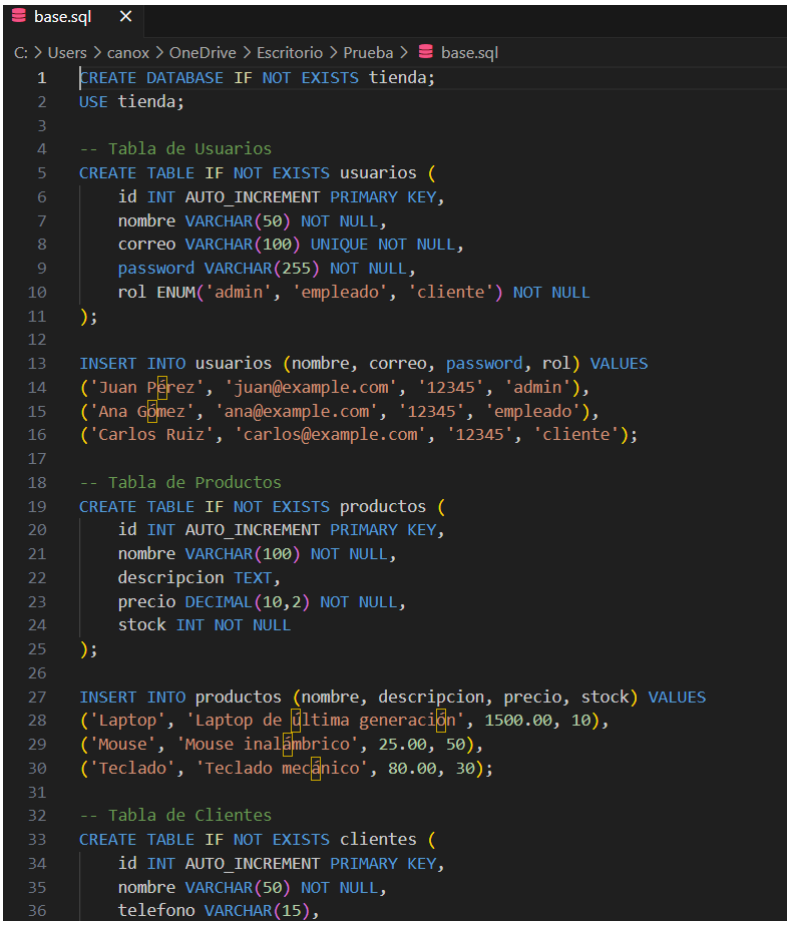
```
services:
  mysql:
    image: mysql:latest
    container_name: Tienda
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: 12345
      MYSQL_DATABASE: "tienda"
      MYSQL_USER: "user"
      MYSQL_PASSWORD: "password"
    ports:
      - "3310:3306"
```

```
    volumes:
      - C:/Users/canox/OneDrive/Escritorio/Prueba/mysql_data:/var/lib/mysql
      - ./base.sql:/docker-entrypoint-initdb.d/base.sql
volumes:
  mysql_data:
```

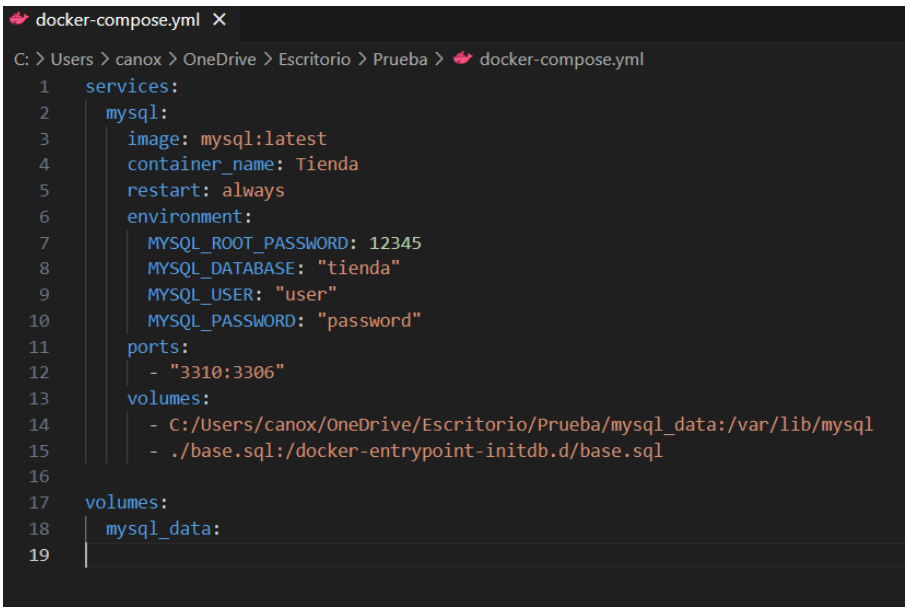
Paso a Paso

Creamos una carpeta donde vamos a guardar todos los archivos

Creamos un archivo .sql donde vamos a guardar una base de datos con diferentes registros

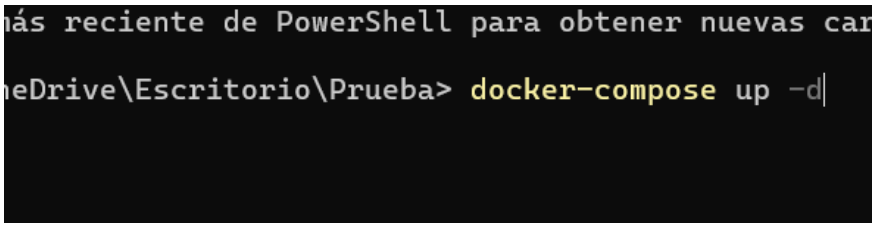


Creamos un archivo. yml e ingresamos los siguientes ajustes teniendo en cuenta el nombre de la base de datos que crea el archivo y el nombre del archivo .sql

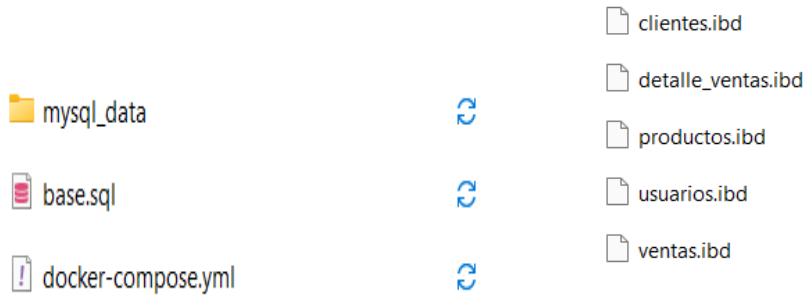


Abrimos un terminal en la carpeta y ejecutamos el siguiente comando para ejecutar el compose

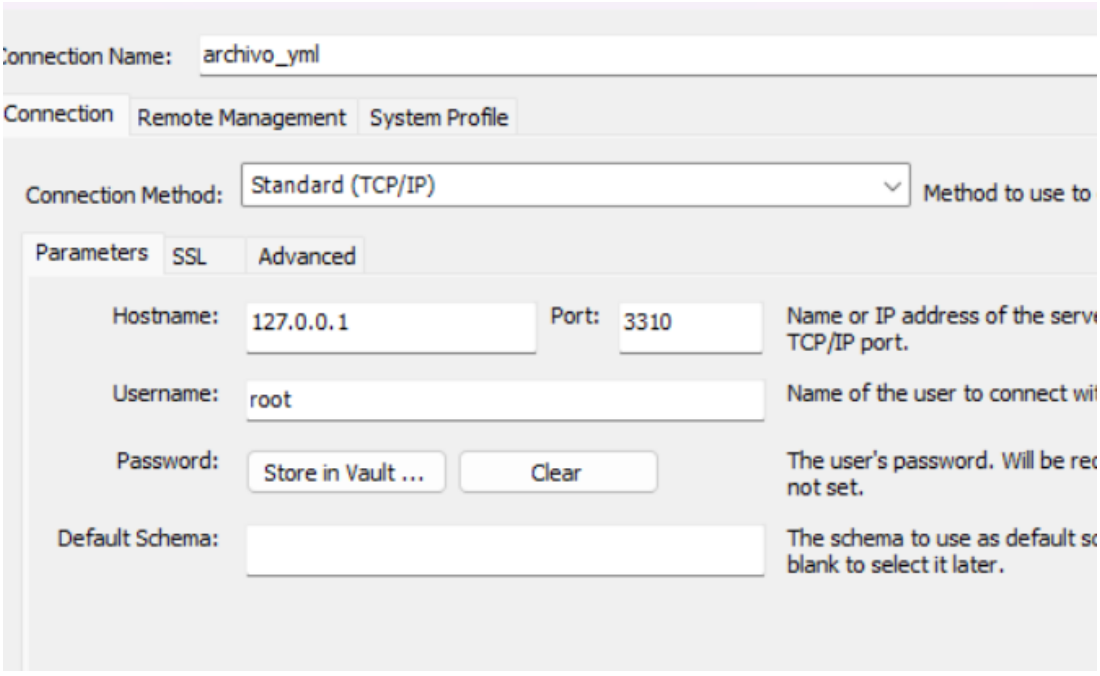
Docker-compose up -d

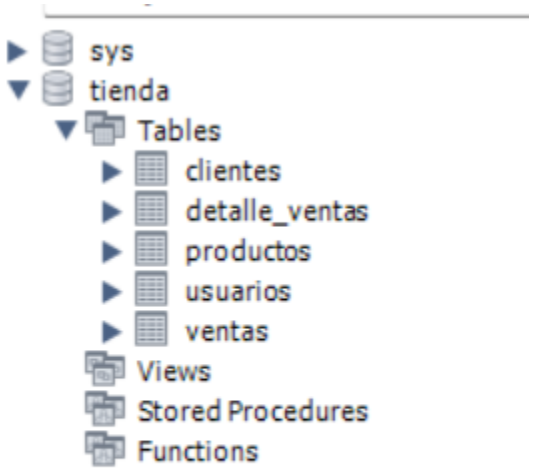


Se creará el volumen y dentro estará la base de datos con sus tablas



Verificamos la conexión en el workbench





Referencias

Courtois, S. (2024, julio 5). *JSON vs YAML: What's the Difference, and Which One Is Right for Your Enterprise?* SnapLogic. https://www.snaplogic.com/blog/json-vs-yaml-whats-the-difference-and-which-one-is-right-for-your-enterprise

Qué es Docker Compose y Cómo Usarlo. (s/f). Imaginaformacion.com. Recuperado el 20 de marzo de 2025, de https://imaginaformacion.com/tutoriales/que-es-docker-compose