

Consultas Arquitectura de Software

Karen Natalia Fernández Cortés

ID:832458

Ingeniería de sistemas, Facultad de ingeniería, Corporación Universitaria Minuto de Dios

NRC:10-66168: Desarrollo Basado en Plataformas

Prof. William Alexander Matallana Porras

Marzo 20, 2025

Consultas Arquitectura de Software

1. Archivos .yaml vs json

YAML y JSON son formatos de serialización de datos que facilitan el intercambio de información entre aplicaciones, siendo ambos legibles por humanos. Ambos representan datos como pares de clave y valor, pero tienen diferencias importantes. JSON se asemeja más al código de programación, utilizando llaves para objetos, dos puntos para separar pares clave-valor, corchetes para matrices y comillas para cadenas. Su sintaxis se deriva de JavaScript.

Por otro lado, YAML está diseñado para ser más legible y natural. Usa sangrías para representar objetos, dos puntos para separar pares clave-valor, guiones para listas y almohadillas para comentarios. Se creó con el objetivo de simplificar la representación de datos en comparación con XML.

Para ilustrar la diferencia, un documento que describe un producto (una camiseta) en

YAML se vería así:

```
type: Camiseta
price: 20,00
sizes:
  - S
  - M
  - L
reviews:
  - username: usuario1
    rating: 4
```

created_at: 2023-04-19T12:30:00Z

- username: usuario2

rating: 5

created_at: 2023-05-02T15:00:00Z

Mientras que en JSON se vería así:

```
{
  "product": {
    "type": "Camiseta",
    "price": 20.00,
    "sizes": ["S", "M", "L"],
    "reviews": [
      { "username": "usuario1", "rating": 4, "created_at": "2023-04-19T12:30:00Z" },
      { "username": "usuario2", "rating": 5, "created_at": "2023-05-02T15:00:00Z" }
    ]
  }
}
```

2. Docker-compose.yml-uso

Una de las principales características de **Docker-compose.yml** es que contiene las instrucciones necesarias para que Docker Compose pueda manejar múltiples contenedores al mismo tiempo. Por defecto, este archivo se encuentra en la ruta ./docker-compose.yml. Su

configuración es bastante sencilla, permitiendo al usuario definir una variedad de comandos directamente en el archivo, sin necesidad de usar la línea de comandos.

Funciones de Docker-compose.yml

Docker-compose.yml tiene varias funciones importantes. Por ejemplo, puede especificar cómo debe actuar el sistema, desde dónde obtener el archivo Docker o Dockerfile, hasta cómo construir imágenes con requisitos específicos. También permite establecer qué puertos deben ser expuestos y cómo vincular contenedores entre sí.

Esta herramienta es esencial para definir los servicios que componen una o más aplicaciones, permitiendo que se ejecuten juntas en un entorno aislado. En resumen, **Docker-compose.yml** simplifica el proceso de gestión de aplicaciones en contenedores al ofrecer un formato fácil de entender y escribir.

3. Como se crea un contenedor usando yaml

1. Crear el archivo docker-compose.yml

El primer paso es crear un archivo llamado docker-compose.yml. Este archivo es esencial porque contiene toda la configuración necesaria para que Docker Compose sepa qué hacer. Utiliza la estructura YAML, que es un formato legible por humanos, para definir varios aspectos de los contenedores.

Estructura básica del archivo:

```
yaml

version: '3.8' # Especifica la versión de Docker Compose que estás utilizando.

services:      # Aquí comienzas a definir los servicios (contenedores) que vas a usar.

  mi_servicio: # Nombre que le das a tu servicio. Puedes elegir cualquier nombre
    descriptivo.

    image: nginx:latest # Indica la imagen de Docker que se usará. En este caso, la última
    versión de Nginx.

    ports:      # Aquí defines el mapeo de puertos.

      - "8080:80" # Mapea el puerto 80 del contenedor al puerto 8080 de tu máquina local.
```

En este ejemplo se está configurando un contenedor que ejecuta Nginx, un servidor web popular. El puerto 80 del contenedor (donde Nginx escucha) se asigna al puerto 8080 de tu máquina local, lo que significa que puedes acceder al servidor web desde tu navegador en `http://localhost:8080`.

2. Ejecutar el contenedor

Una vez que hayas creado y guardado tu archivo `docker-compose.yml`, el siguiente paso es ejecutar los contenedores definidos en él.

Instrucciones para ejecutar:

1. Abre una terminal en la carpeta donde guardaste tu archivo `docker-compose.yml`.
2. Ejecuta el siguiente comando:

```
bash
```

`docker-compose up -d`

- up: Este comando crea y ejecuta los contenedores definidos en tu archivo YAML. Si los contenedores ya existen, los reiniciará.

-d: Esta opción indica que se desea ejecutar los contenedores en segundo plano (detached mode). Esto significa que después de ejecutar el comando, podrás seguir usando la terminal sin que se bloquee con los logs del contenedor.

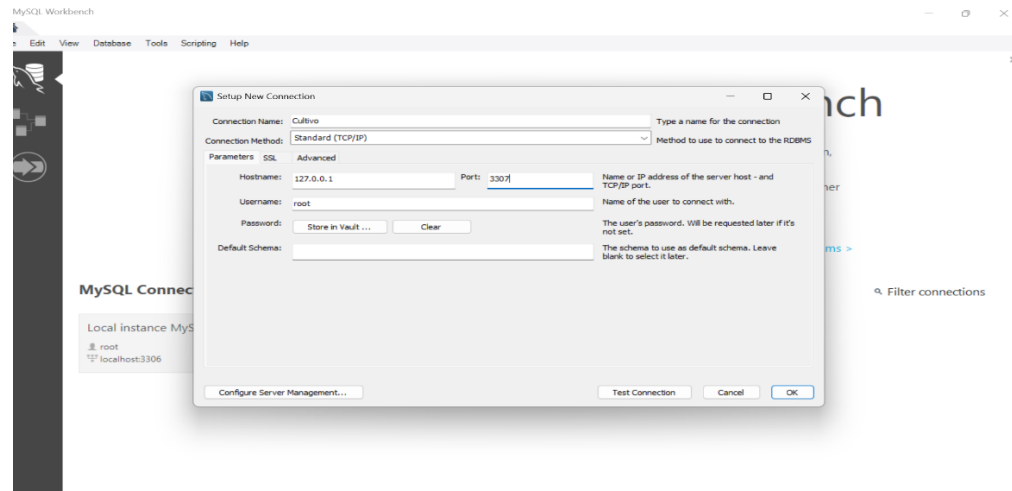
Para ver los contenedores activos:

Si se desea verificar qué contenedores están actualmente en ejecución, se puede usar el siguiente comando:

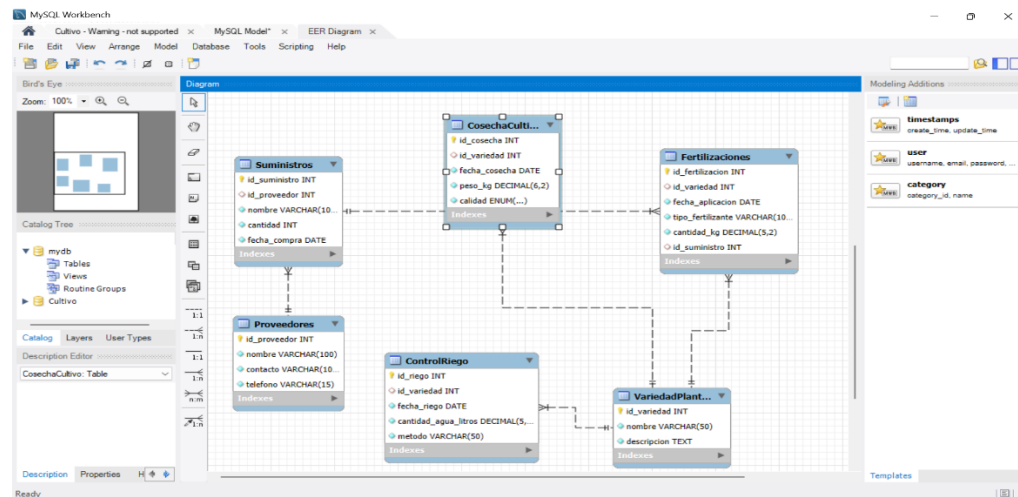
`docker ps`

Este comando te mostrará una lista de todos los contenedores activos junto con información útil como el ID del contenedor, la imagen utilizada, el estado y los puertos expuestos.

4. Crea una base de datos que tenga tablas y registros insertados.
- Configuración de la nueva conexión en MySQL Workbench.



- Modelo relacional obtenido de la base de datos.



Referencias

- AWS. (s.f). ¿Cuál es la diferencia entre YAML y JSON? AWS.

<https://aws.amazon.com/es/compare/the-difference-between-yaml-and-json/>

- Mallón Xoán. (2024, octubre 25). ¿Qué es Docker-compose.yml? keep coding.

<https://keepcoding.io/blog/que-es-docker-compose-yml/>