Comparación de despliegue de un gestor de base de datos NoSQL mediante Docker

Marko Antonio Rivas Rios

Jorge Luis Mamani Maquera

Orlando Antonio Acosta Ortiz

Yofer Nain Catari Cabrera

Orestes Ramirez Ticona

Roberto Zegarra Reyes

Universidad Privada de Tacna

Junio 22, 2019

Abstract

Docker is an open source project created in 2013 and which has been a revolution for the development and deployment of operations. Docker abstracts the host's hardware and operating system by running the applications in containers, isolated compartments that contain all the resources for an application or service. In this work we will see how to use Docker for the development of simple applications, learning how to deploy a NOSQL database with Docker.

Abstract

Docker es un proyecto open source creado en 2013 y que ha supuesto una revolución para el desarrollo y despliegue de operaciones. Docker abstrae el hardware y el sistema operativo del host ejecutando las aplicaciones en contenedores, compartimentos aislados que contienen todos los recursos para una aplicación o servicio.

En este trabajo veremos cómo usar Docker para el desarrollo de aplicaciones sencillas, aprendiendo a desplegar una base de datos NOSQL con Docker.

I. Introducción

El potente concepto de Microservicios está cambiando poco a poco la industria. Grandes servicios monolíticos están dando paso lentamente al enjambre microservicios pequeños y autónomos que trabajan en conjunto. El proceso va acompañado de otra tendencia del mercado: la contenerización. Juntos, ayudan a construir sistemas sin precedentes. La contenerización cambia no sólo la arquitectura de los servicios, sino también la estructura de ambientes utilizados para crearlos.

Ahora, cuando el software se distribuye en contenedores, los desarrolladores tienen plena libertad para decidir qué aplicaciones necesitan. Como resultado, incluso los entornos complejos, como los servidores de grades bases de datos e infraestructura de análisis complejos pueden crear instancias en cuestión de segundos. El desarrollo de software se hace más fácil y más eficaz.

II. Objetivos

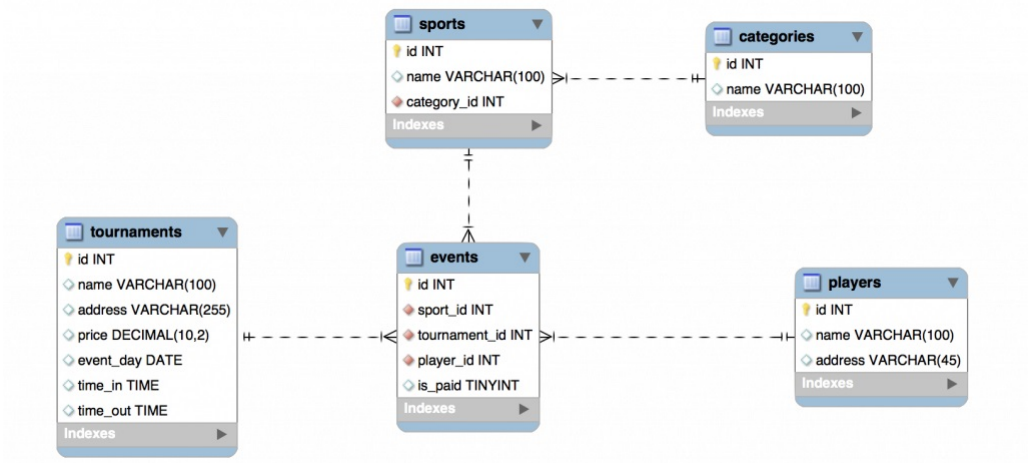
Se busca saber un poco mas sobre:

- Las bases de datos relacionales y no relacionales.
- Realizar comparaciones en cuanto funcionamiento, metodologias etc.

III. SQL y NoSQL

i. Base de datos SQL

Structured Query language (SQL) o lenguaje de consulta estructurada, utiliza un estándar para manipular el contenido de las bases de datos relacionales. Cuenta con su propio lenguaje para consultar, insertar, actualizar y eliminar registros de la base de datos.
Se compone de una o varias tablas y cada tabla puede contener múltiples relaciones. En el siguiente elemento pongo como ejemplo en como está estructurada una sencilla base de datos por medio de tablas.



Las tablas representan la entidad de cada modelo de datos. Las relaciones permiten garantizar la persistencia de datos, evita la duplicidad de datos y controla la eliminación en cascada. Para ingresar un nuevo nuevo evento, bastaría con ingresar un nuevo registro en la tabla events.
SQL cuenta con un lenguaje de alto nivel que permite manipular la base de datos casi de cualquier forma. Por su manera de operar ayuda a mantener la integridad de los datos.

Por ejemplo:

Para obtener la lista de todos los deportes SELECT id, name FROM sports
Ingresar varios registros a la tabla categoría

INSERT INTO categories VALUES (Varonil, Femenil, mixto)
Su uso es más frecuente es en sistemas donde se necesite cuidar la integridad de datos como los bancos, sistemas de ventas, hospitales, CMS como WordPress.

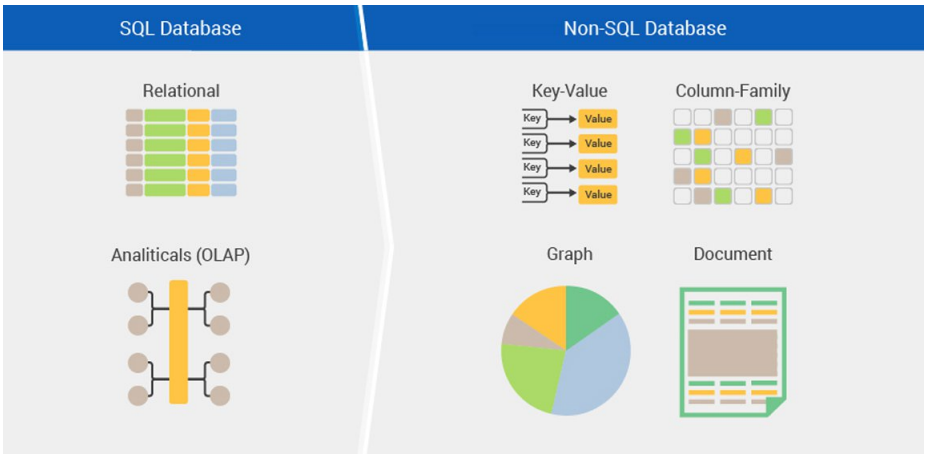
Ejemplos de sistemas de gestión de base de datos son: MySQL, MariaDB, PostgreSQL.

ii. Base de datos NoSQL

También existen otros mecanismos de almacenamientos alternativos a las base de datos relacionales que se definen por tablas. Este tipo de base de datos, se apoyan en otros formatos. Los NOSQL son más adecuados para aquellos que manejan grandes volúmenes de datos. Es por eso que empresas como Facebook, Google, Amazon, Twitter, entro otros han popularizado estos tipos de base de datos.

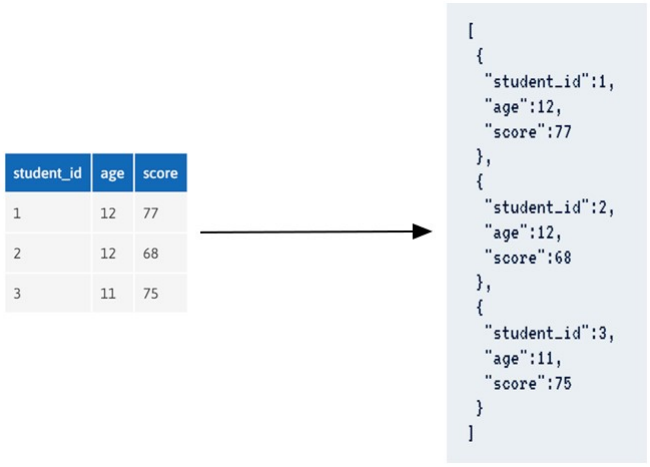
Esas empresas tenían que enfrentarse a grandes desafíos motivados a la alta demanda de internet. Donde surge la necesidad de proporcionar información de grandes volúmenes de datos e indeterminado número de usuarios en el menor tiempo posible. Esas compañías se dieron cuenta que podían sacrificar la consistencia de los datos que garantiza una base de datos relacional para poder obtener un mejor rendimiento y ofrecer soluciones en tiempo real.

Los puedes usar en soluciones donde requieres interactuar con datos en tiempo real como en un chat, para análisis estadísticos donde procesas mucha información o incluso en blogs. Algunos tipos de bases de datos NoSQL incluyen diferentes tipos de almacenamiento, por ejemplo: con columnas, documentos, key value store, gráficos, objetos (JSON), XML y otros modos de almacenamiento de datos.



iii. Formatos

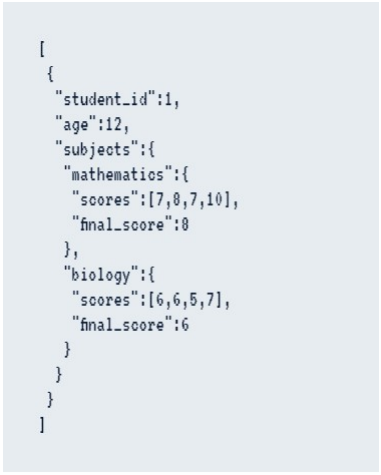
La información puede organizarse en tablas o en documentos. Cuando organizamos información en un Excel, lo hacemos en formato tabla y, cuando los médicos hacen fichas a sus pacientes, están guardando la información en documentos. Lo habitual es que las bases de datos basadas en tablas sean bases de datos relacionales y las basadas en documentos sean no relacionales, pero esto no tiene que ser siempre así. En realidad, una tabla puede transformarse en documentos, cada uno formado por cada fila de la tabla. Solo es una cuestión de visualización.



Los dos esquemas de la imagen contienen exactamente la misma información. Lo único que cambia aquí es el formato: cada documento de la figura de la derecha es una fila de la figura de la izquierda.

Se ve más claro en la tabla, ¿verdad? Lo que pasa es que a menudo en una base de datos no relacional una unidad de datos puede llegar a ser demasiado compleja como para plasmarlo en una tabla.

Por ejemplo, en el documento JSON de la imagen que se muestra a continuación, al tener elementos jerárquicos, es más difícil plasmarlo en una tabla plana. Una solución sería plasmarlo en varias tablas y, por tanto, necesitar de relaciones.



```
[
 {
  "student_id":1,
  "age":12,
  "subjects":{
   "mathematics":{
    "scores":[7,8,7,10],
    "final_score":8
   },
   "biology":{
    "scores":[6,6,5,7],
    "final_score":6
   }
  }
 }
]
```

Esto explica por qué las bases de datos relacionales suelen servirse de tablas y las no relacionales de documentos JSON. En cualquier caso, a día de hoy, **las bases de datos más competitivas suelen permitir, de una forma u otra, operaciones de los dos tipos.**

Por ejemplo, el servicio de base de datos en la nube BigQuery que ofrece Google es, en principio, una base de datos de lenguaje de consulta SQL, por lo que permite fácilmente relacionar varias tablas, pero, a su vez, permite insertar elementos jerárquicos JSON, más propios de bases de datos no relacionales.

La diferencia entre el éxito y el fracaso recae, sobre todo, en el diseño del modelo.
Es decir, si se decide que el mejor enfoque es usar una base de datos relacional, no es suficiente con meter la información a lo bruto en una base de datos relacional y esperar a que se relacione sola, porque eso no va a ocurrir. De nada sirve elegir la base de datos más apropiada para nuestro sistema, si luego no se hace un buen diseño.

iv. Ventajas y Desventajas de SQL y NoSQL

Ventajas de las Bases de Datos relacionales
- Está más adaptado su uso y los perfiles que las conocen son mayoritarios y más baratos.

- Debido al largo tiempo que llevan en el mercado, estas herramientas tienen un mayor soporte y mejores suites de productos y add-ons para gestionar estas bases de datos.

- La atomicidad de las operaciones en la base de datos. Esto es, que en estas bases de datos o se hace la operación entera o no se hace utilizando la famosa técnica del rollback.

- Los datos deben cumplir requisitos de integridad tanto en tipo de dato como en compatibilidad.

Desventajas de las Bases de Datos relacionales
- La atomicidad de las operaciones juega un papel crucial en el rendimiento de las bases de datos.

- Escalabilidad, que aunque probada en muchos entornos productivos suele, por norma, ser inferior a las bases de datos NoSQL.

Ventajas de una base de datos NoSQL

- La escalabilidad y su carácter descentralizado. Soportan estructuras distribuidas.

- Suelen ser bases de datos mucho más abiertas y flexibles. Permiten adaptarse a necesidades de proyectos mucho más fácilmente que los modelos de Entidad Relación.

- Se pueden hacer cambios de los esquemas sin tener que parar bases de datos.

- Escalabilidad horizontal: son capaces de crecer en número de máquinas, en lugar de tener que residir en grandes máquinas.

- Se pueden ejecutar en máquinas con pocos recursos.

- Optimización de consultas en base de datos para grandes cantidades de datos.

Desventajas de una base de datos NoSQL
- No todas las bases de datos NoSQL contemplan la atomicidad de las instrucciones y la integridad de los datos. Soportan lo que se llama consistencia eventual.

- Problemas de compatibilidad entre instrucciones SQL. Las nuevas bases de datos utilizan sus propias características en el lenguaje de consulta y no son 100 por ciento compatibles con el SQL de las bases de datos relacionales. El soporte a problemas con las queries de trabajo en una base de datos NoSQL es más complicado.
- Falta de estandarización. Hay muchas bases de datos NoSQL y aún no hay un estándar como sí lo hay en las bases de datos relacionales. Se presume un futuro incierto en estas bases de datos.

- Soporte multiplataforma. Aún quedan muchas mejoras en algunos sistemas para que soporten sistemas operativos que no sean Linux.

- Suelen tener herramientas de administración no muy usables o se accede por consola.

v. SQL Vs NoSQL

Cuándo utilizar qué tipo de base de datos

- Cuando los datos deben ser consistentes sin dar posibilidad al error utilizar una base de datos relacional, SQL.

- Cuando nuestro presupuesto no se puede permitir grandes máquinas y debe destinarse a máquinas de menor rendimiento, NoSQL.

- Cuando las estructuras de datos que manejamos son variables, NoSQL.

- Análisis de grandes cantidades de datos en modo lectura, NoSQL.

- Captura y procesado de eventos, NoSQL.

- Tiendas online con motores de inteligencia complejos, NoSQL.

Es muy importante en este punto insistir en que, aunque parece que en estos momentos lo suyo es migrar a bases de datos NoSQL, debemos tener muy en cuenta antes de tomar esta decisión si las características de nuestra base de datos necesitan una base de datos NoSQL o relacional. Una base de datos NoSQL incluye simplicidad de diseño, escala horizontal más simple para grupos de máquinas y un mejor control de la disponibilidad.

IV. ¿Cuándo debería usarse NoSQL?

Cuando gran cantidad de datos necesitan ser almacenados y recuperados. La relación entre los datos que almacenas no es tan importante Los datos cambian con el tiempo y no están estructurados. No se requiere soporte de restricciones y uniones a nivel de base de datos Los datos crecen continuamente y usted necesita escalar la base de datos regularmente para manejar los datos.

V. Conclusiones

Un NoSQL originalmente referido a no SQL o no relacional es una base de datos que proporciona un mecanismo para el almacenamiento y recuperación de datos. Estos datos se modelan en medios distintos a las relaciones tabulares utilizadas en las bases de datos relacionales. Siempre es bueno evaluar las diferentes alternativas que ofrece el mundo del software, para poder dar solución a muchas de los requerimientos que se presentan en el medio, pero siempre hay que considerar que cualquier solución de software que se seleccione va a tener sus ventajas y desventajas al momento de su desarrollo como también en la operación.

La aparición del nuevo NoSQL es relativamente nueva, todavía le queda mucho tiempo para poder afirmar si es la solución a muchos de los problemas que hoy en día se presentan en el

mundo de la base de datos, una de las ventajas muy importantes que ofrece NoSQL es que puede manejar grandes cantidades de información, pudiendo devolver el resultado de una consulta en cuestión de segundo, cuando un manejador tradicional RDBMS puede tardar 1000 veces mas el tiempo que el anterior haya resuelto.

Antes de tomar la decisión de migrar una base de datos relacional a una no relacional hay que considerar todos estos puntos ventajas y desventajas para no caer en un punto del cual ya no se puede regresar y no tener perdidas de información importantes.

References

1. http://revistatelematica.cujae.edu.cu/ index.php/tele/article/view/23/21
2. https://programarfacil.com/blog/ que-es-un-orm/
3. https://www.beeva.com/beeva-view/tecnologia/mas-alla-de-la-virtualizacion-contenedores/
4. https://searchdatacenter.techtarget.com/ es/definicion/virtualizacion-basada-en-contenedores-virtualizacion-a-nivel-de-sistema-operativo
5. https://www.incibe-cert.es/blog/asegurando-virtualizacion-tus-sistemas-control
6. http://www.datakeeper.es/?p=716
7.https://sigmodrecord.org/publications/sigmodRecord/1012/pdfs/04.surveys.cattell.pdf

8.https://ieeexplore.ieee.org/abstract/document/6625441
9.-http://nosql-database.org/