INFORME SEMINARIO

BASE DE DATOS NO SQL 2024

DOCENTE:

Ing. Viviana Ferragine

INTEGRANTES:

Alvarez, Paulo Manuel

Belaunzaran, María Josefina

González, Miriam Alicia

Toledo López, María Agustina

Sistema de Alquiler de autos con Base de Datos No SQL Cassandra

Para un sistema de alquiler de autos que requiere una base de datos NoSQL, vamos a utilizar Cassandra. La misma se identifica por ser distribuida y altamente escalable, diseñada para manejar grandes volúmenes de datos a través de múltiples servidores o nodos, sin un único punto de fallo. E

Está diseñada para ser horizontalmente escalable, lo que significa que se puede agregar más servidores (nodos) a un clúster de manera sencilla. A medida que los datos y la carga de trabajo aumentan, simplemente se añaden más nodos, y Cassandra redistribuye los datos entre ellos automáticamente. Además, si un nodo falla, el resto del clúster puede continuar operando sin interrupciones, gracias a su sistema de replicación de datos.

Cassandra usa un modelo de columnas (equivalentes a las tablas) con una estructura flexible que con una estructura flexible que puede ser consultada a través de claves de partición y claves de clustering, lo que permite acceder rápidamente a los datos sin necesidad de hacer joins como en bases de datos relacionales.

Usa un lenguaje de consultas similar a SQL llamado CQL (Cassandra Query Language, por sus siglas en inglés), que facilita su adopción para quienes están acostumbrados a SQL, pero con las características y limitaciones de una base de datos NoSQL (inconsistencia eventual, falta de soporte para transacciones complejas, entre otros).

VENTAJAS DE USAR CASSANDRA:

Cassandra es una opción adecuada para este tipo de aplicación, debido a su alta escalabilidad y capacidad para manejar grandes volúmenes de datos distribuidos. La misma ofrece las siguientes ventajas:

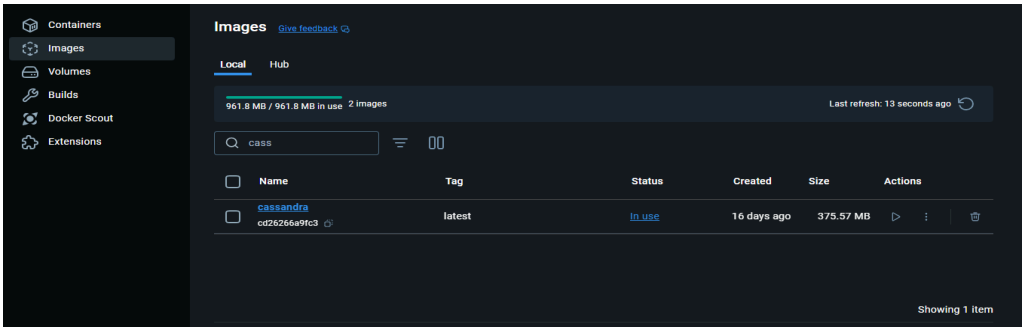
- Escalabilidad Horizontal: Se pueden añadir más nodos fácilmente para aumentar la capacidad de la base de datos sin interrupciones significativas.
- Desempeño/ rendimiento en Tiempo Real: Permite la actualización y consulta rápida de datos, lo que es esencial para la funcionalidad de posicionamiento en tiempo real.
- Modelo de Datos Flexible: puede adaptarse a los cambios en los requisitos del sistema.
- Consultas Complejas: Debe ser capaz de realizar consultas complejas relacionadas con la disponibilidad de autos, precios, e historial de alquileres.

De este modo se puede garantizar que el sistema de alquileres de autos maneje eficientemente tanto los datos en tiempo real como los históricos, manteniendo un buen desempeño a medida que el sistema crece.

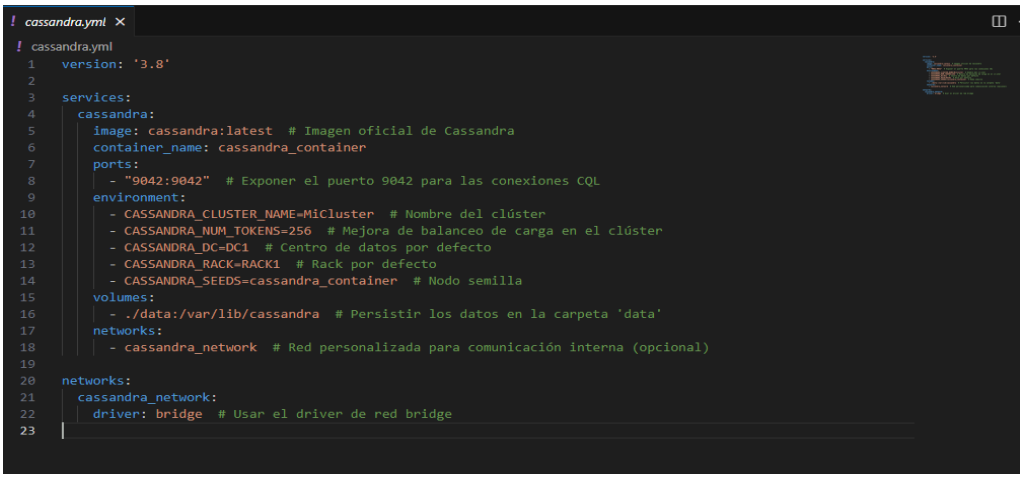
Antes de comenzar con el modelo conceptual de la base de datos, se debe establecer una conexión, para eso en este caso se utiliza la plataforma Docker.

CONEXIÓN CON LA BASE DE DATOS CASSANDRA:

Como primer paso se debe descargar una imagen Cassandra en la plataforma Docker.



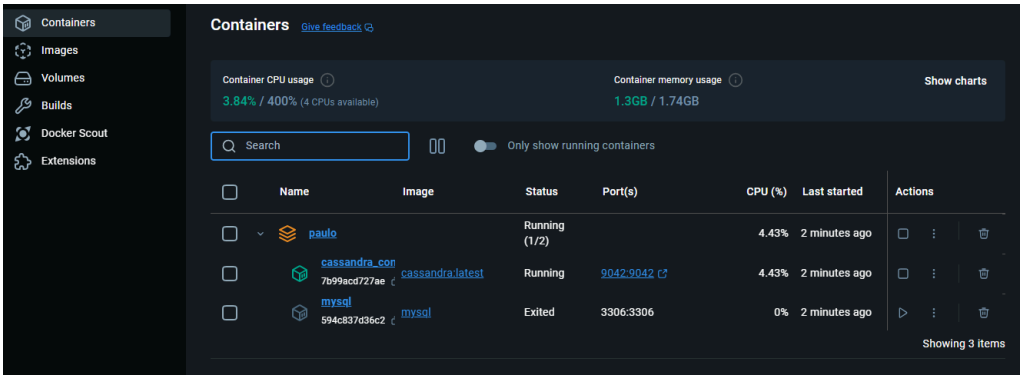
Luego se necesita crear un contenedor, para ello se requiere un archivo de configuración yaml.



Una vez creado el archivo, se ejecuta por consola el siguiente comando:

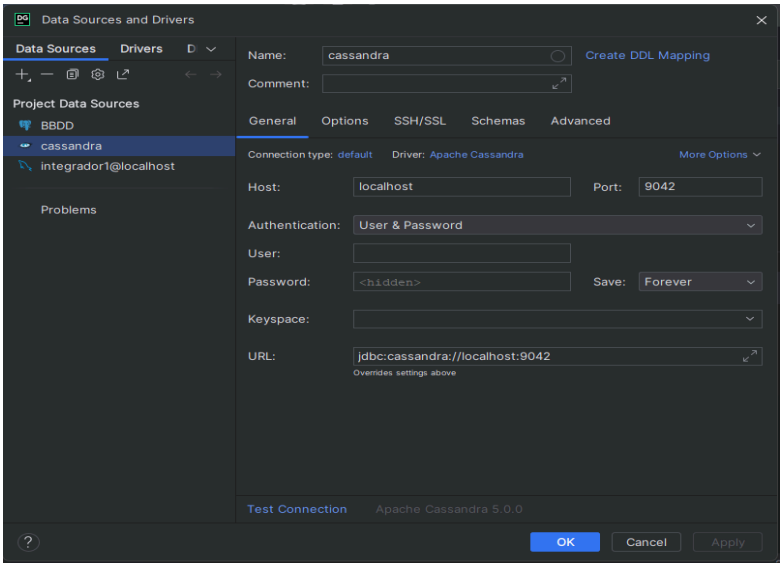
- docker-compose -f cassandra.yml up

Esto crea el contenedor de la imágen Cassandra que se puede comprobar desde la aplicación.

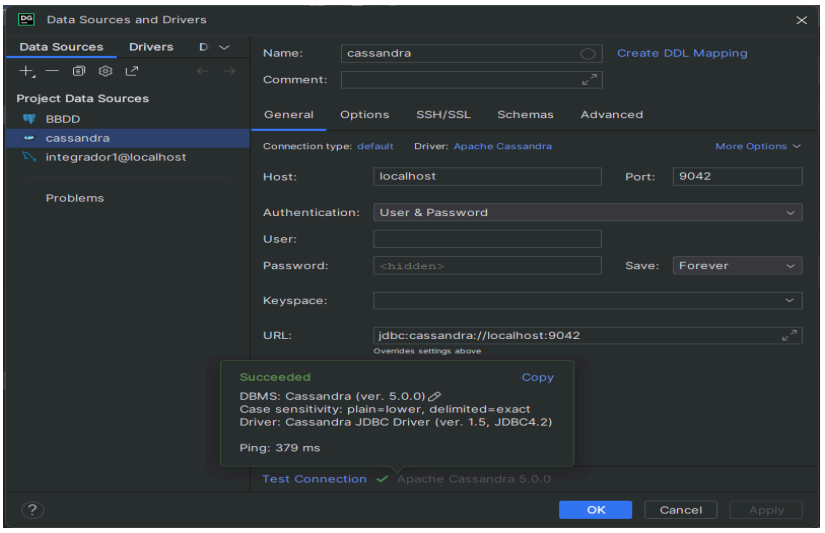


Una vez que está corriendo el contenedor de la base de datos, se procede a interactuar mediante un cliente, en este caso se elige DataGrip.

Para ello se crea previamente un nuevo data source de Apache Cassandra, y se establece la conexión mediante el puerto 9042 de la siguiente manera:



Se realiza el Test de conexión:



Y a partir de ahora se puede comenzar a trabajar con la base de datos.

METODOLOGÍA QUERY-DRIVEN:

Se diseña la base de datos respondiendo a las siguientes consultas:

Q1 ⟶ Dado un id de auto mostrar todos sus datos

Q2 ⟶ Dada la clave de un cliente dar todos sus datos

Q3 ⟶ Dada una fecha de inicio mostrar el auto y precio por día

Q4 ⟶ Dada la clave de un auto mostrar su ubicación (latitud y longitud)

La estructura de datos seleccionada es de ***familia de columnas***, con ***cuatro*** nodos (y uno más para replicación y tolerancia a fallos) que la representan:

1) **Autos:**
   - **ID del auto** (clave primaria)
   - **Marca:** Nombre de la marca del auto.
   - **Modelo:** Modelo del auto.
   - **Año:** Año de fabricación.
   - **Estado:** Estado actual del auto (por ejemplo, disponible, alquilado, en mantenimiento).
   - **Ubicación Actual:** Datos de posicionamiento en tiempo real (latitud, longitud).
   - **Precio por Día:** Precio de alquiler por día.
   - **Descripción:** Información adicional sobre el auto (color, características especiales).
   - **PRIMARY KEY ((marca, modelo), id_auto):** Marca y modelo son la clave de partición, e id_auto es parte de la clave de clustering, lo que asegura la unicidad.

2) **Personas:**
   - **ID de Persona (clave primaria)**
   - **Nombre:** Nombre completo de la persona.
   - **Correo Electrónico:** Correo electrónico de contacto.
   - **Teléfono:** Número de teléfono de contacto.
   - **Historial de Alquileres:** Registro de alquileres previos (referencias a IDs de autos y fechas de alquiler).

3) **Precios:**
   - **ID de Auto (clave primaria, referencia a la tabla de Autos)**
   - **Fecha de Inicio:** Fecha en la que se aplica el precio.
   - **Precio por Día:** Precio de alquiler por día.
   - **Tipo de Temporada:** Temporada o eventos especiales que pueden afectar el precio.

4) **Posicionamiento:**
   - **id_auto UUID:** ID único del auto
   - **fecha_hora TIMESTAMP:** Fecha y hora del posicionamiento
   - **latitud DOUBLE:** Coordenada de latitud
   - **longitud DOUBLE:** Coordenada de longitud
   - **PRIMARY KEY (id_auto, fecha_hora):** Clave primaria compuesta por id_auto y fecha_hora.
   - **WITH CLUSTERING ORDER BY (fecha_hora DESC)**: Orden por la fecha más reciente.

1. **Nodo de almacenamiento autos**:
   - Incluye **ID de autos**, **marca**, **modelo**, **ubicación**, etc.
   - **PRIMARY KEY ((marca, modelo), id_auto):** Marca y modelo son la clave de partición, e id_auto es parte de la clave de clustering, lo que asegura la unicidad.

2. **Nodo de almacenamiento personas**:

- Incluye clientes e historial de alquileres.
- **Clave de partición**: id_persona según su UUID.

3. **Nodo de almacenamiento precios**:
   - Incluye los precios de alquiler de cada auto en diferentes fechas.
   - **Clave de partición**: id_auto, donde los precios de alquiler están en el mismo nodo. Dentro de cada partición, las filas se ordenan por fecha_inicio.

4. **Nodo de almacenamiento posicionamiento**:
   - Almacenan las ubicaciones geográficas de los autos con la fecha correspondiente.
   - **Clave de partición**: id_auto, con posiciones ordenadas por fecha_hora en el mismo nodo.
   - Útil para realizar consultas geoespaciales o rastreo de vehículos en tiempo real.

5. **Nodo para replicación de datos (tolerancia a fallos)**:

   - El **factor de replicación** determina cuántas copias de tus datos se almacenarán en diferentes nodos. El mismo guarda copias de los datos, y comúnmente se replican en 3 nodos diferentes del clúster. Así, si uno falla,, los otros dos nodos que contienen las réplicas aún pueden servir los datos.

Usuarios globales:

Estrategia de replicación avanzada *"NetworkTopologyStrategy"* para replicar los datos en diferentes centros de datos (regiones geográficas).

También utilizamos comandos tales como **uuid()**, con el objeto de generar identificadores únicos universales (por sus siglas en inglés). Estos son valores de 128 bits únicos en el espacio y el tiempo, lo que los hace ideales para usarse como claves primarias en tablas donde no se quiere o no se puede usar un identificador secuencial o basado en el usuario, incluso si se crean en diferentes nodos de un clúster distribuido. Este comando permite que cada nodo origine de manera independiente un identificador único.

En nuestro trabajo usamos uuid() en la configuración de los nodos, y también el comando **toTimestamp(now())** por ejemplo, para añadir al historial un alquiler reciente de un auto, donde el primero representa el ID del auto alquilado y el segundo captura la fecha y hora actuales.

ENLACE AL SCRIPT DE LA BASE DE DATOS CASSANDRA:

https://drive.google.com/file/d/1RjlpA_-vBwcTtuehWf-SYsJrSrytTlVQ/view?usp=sharing