UNIR - Universidad Internacional de la Rioja

Ciudad de México

**ACTIVIDAD 2:**

**Uso avanzado de bases de datos NoSQL**

**Alumnos:**

Paola Michelle Figueroa Benitez

Lowenski Paredes Rosario

Carlos Damian Rodriguez Uitzil

Cuenca Roa Leonard Jose

**Grupo: 1001**

**Equipo 01C**

**ÍNDICE**

Prepara el entorno de trabajo 3

Inicia la monitorización en MongoDB 3

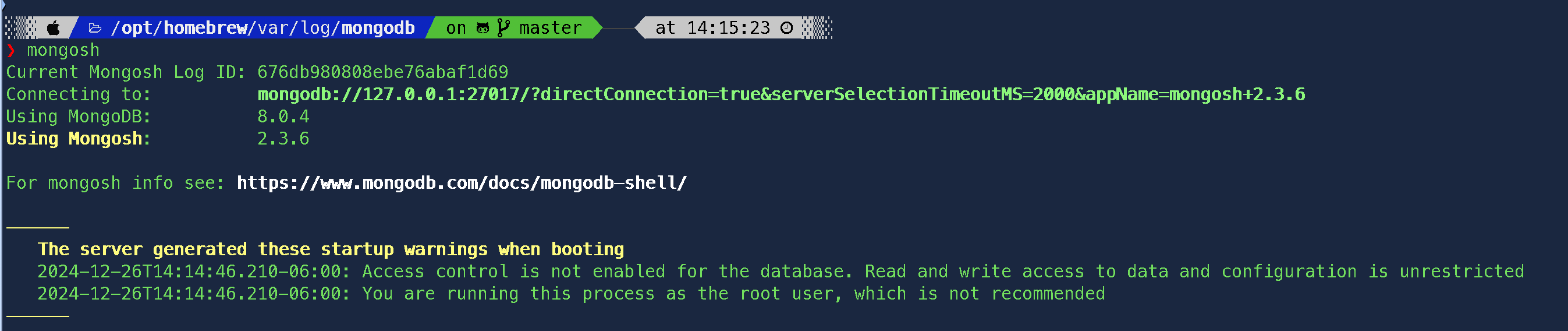
Importar la base de datos 4

Exportar la base de datos 12

Restaurar una base de datos 13

Caso de uso 15

Prepara el entorno de trabajo

Se implementó un entorno de desarrollo local consistente en MongoDB versión 8.0.4 y Mongosh 2.3.6 en cada equipo de los participantes. Esta configuración permitió a cada miembro del equipo trabajar de manera independiente y sincronizada, facilitando la colaboración y el testing. 

**Inicia la monitorización en MongoDB**

Para llevar a cabo esta actividad, se implementó una estrategia de ejecución simultánea en tres terminales. En la primera, se utilizó el demonio de MongoDB (mongod) para crear la base de datos 'inspections y countries' y las colecciones 'act-grupal-city\_inspections', 'act-grupal-countries-big' y 'act-grupal-countries-small'. Esta configuración inicial se realizó directamente desde la línea de comandos de MongoDB.

Con el objetivo de monitorear el rendimiento y la salud de la base de datos durante el proceso, se abrieron dos terminales adicionales. En estas, se ejecutaron las herramientas de monitoreo mongotop y mongostat. El comando 'mongotop 10' proporcionó una vista en tiempo real de las operaciones más costosas en la base de datos, mientras que 'mongostat --discover' ofreció un resumen general del estado del servidor y las estadísticas de las bases de datos.

| Creación de Base Datos y sus colecciones | |
| --- | --- |
| // Crear la base de datos  use inspections;  // crear colecciones  db.createCollection(´act-grupal-city\_inspections´)  // visualizar lo que se generó  show collections  //Otra forma de ver  db.getCollectionNames() |  |
|
|
|

| Creación de Base Datos y sus colecciones | |
| --- | --- |
| // Crear la base de datos  use countries;  // crear colecciones  db.createCollection(´act-grupal-countries-big´)  db.createCollection(´act-grupal-countries-small´)  // visualizar lo que se generó  show collections  //Otra forma de ver  db.getCollectionNames() |  |
|
|
|

Importar la base de datos

Ahora para monitoreo de la carga de los datos la actividad número dos, nos proporciono los ficheros que contiene los datos json para las colecciones creadas se procedió a cargar, validar y analizar los resultados que arroja las terminales de monitoreo, Para esta tarea de importación de datos, se aprovechó la rapidez y facilidad que ofrece Compass, una excelente herramienta de administración de bases de datos MongoDB. Previamente, se configuró Compass para utilizar la base de datos instalada localmente

| Importar datos | |
| --- | --- |
| Se importó los datos en la BD llamada **inpections**, validando que en efecto esté el total de datos 81047. | Captura de pantalla 2024-12-26 a la(s) 16.21.30.png  Captura de pantalla 2024-12-26 a la(s) 16.22.01.png |
|
|
|

| Importar datos | |
| --- | --- |
| Se importó los datos en la BD llamada **countries**, validando que en efecto esté el total de datos 21640 para la colección act-grupal-countries-big. | Captura de pantalla 2024-12-26 a la(s) 17.28.16.png  Captura de pantalla 2024-12-26 a la(s) 17.29.08.png |
|
|
|

| Importar datos | |
| --- | --- |
| Se importó los datos en la BD llamada **countries**, validando que en efecto esté el total de 1 para la colección act-grupal-countries-small. | Captura de pantalla 2024-12-26 a la(s) 17.28.24.png  Captura de pantalla 2024-12-26 a la(s) 17.29.15.png |
|
|
|

| Análisis del comando mongotop al importar “act-grupal-city\_inspections.json” | |
| --- | --- |
| Al momento de importar los datos, se nos presenta un proceso dividido en tres fases que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.  **Primera fase:** En esta etapa, se utilizó la operación de lectura durante 37 milisegundos.  **Segunda fase**: Continuando con el proceso, se mantuvo el uso de la operación de lectura, esta vez durante 68 milisegundos.  **Tercera fase:** En esta fase, se observa un cambio en la operación, pasando a utilizar la operación de escritura, la cual consumió 560 milisegundos.  **Análisis:**  A partir de los datos obtenidos, podemos concluir que las operaciones de escritura son más complejas que las de lectura, lo que conlleva un mayor tiempo de procesamiento por parte del procesador. Asimismo, se puede deducir que, durante el proceso de importación, la herramienta realiza una lectura completa del documento JSON antes de proceder a escribir los datos en la base de datos denominada 'inspections' | |
| Captura de pantalla 2024-12-26 a la(s) 16.18.41.png | |
|  |  |

| Análisis del comando mongostat —discover al importar “act-grupal-city\_inspections.json” | |
| --- | --- |
| Este tipo de herramienta nos proporciona en tiempo real la cantidad de operaciones que se realizan por segundo, para el caso de importar la colección nos arroja que se realizo un conjuntos de inserciones sobre la colección mencionada, durante la instancia de Mongo DB en local nos arroja que se realizaron mas de 21.000 mil operaciones durante un segundo terminando el proceso de inserción en 4 bloques de procesos registrados usando un promedio de memoria de 0.725%. | |
| Captura de pantalla 2024-12-26 a la(s) 16.17.27.png | |
|  |  |

| Análisis del comando mongotop al importar “act-grupal-countries-big.json” | |
| --- | --- |
| Al momento de importar los datos, se nos presenta un proceso en una sola fase que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.  **Primera fase**: En esta etapa, se utilizó la operación de lectura durante 246 milisegundos y en segundo plano las operaciones de escritura generando un tiempo de 117 milisegundos con un total de la operación de 363 milisegundos.  **Análisis**:  A partir de los datos obtenidos, podemos concluir que las operaciones de lectura son más complejas que las de escritura, lo que conlleva un mayor tiempo de procesamiento por parte del procesador. Asimismo, se puede deducir que, durante el proceso de importación, la herramienta realiza una lectura completa del documento JSON antes de proceder a escribir los datos en la base de datos denominada ‘countries’. | |
| Captura de pantalla 2024-12-26 a la(s) 17.19.59.png | |
|  |  |

| Análisis del comando mongostat —discover al importar “act-grupal-countries-big.json” | |
| --- | --- |
| Esta herramienta nos proporciona un seguimiento instantáneo del número de operaciones que se ejecutan por segundo en la base de datos. Durante la importación de la colección, se observó un alto volumen de inserciones, superando las 18,000 operaciones por segundo en la instancia local de MongoDB. El proceso se finalizó en dos fases distintas, consumiendo un promedio de solo 0.1% de la memoria disponible. | |
| Captura de pantalla 2024-12-26 a la(s) 17.18.32.png | |
|  |  |

| Análisis del comando mongotop al importar “act-grupal-countries-small.json” | |
| --- | --- |
| Al momento de importar los datos, se nos presenta un proceso en dos fases que reflejan el tiempo total empleado, el cual es la suma del tiempo invertido en las operaciones de lectura y escritura realizadas por el procesador en ese instante preciso.  **Primera fase:** En esta etapa, se utilizó la operación de lectura durante 60 milisegundos y en escritura de 0 milisegundos.  **Segunda fase:** En esta etapa, se utilizó la operación de lectura durante 13 milisegundos y una escritura de 2 milisegundos.  **Análisis:**  A partir de los datos obtenidos, podemos concluir que las operaciones cuando las colecciones son de menor complejidad o tamaño se pueden ejecutar en un mismo bloque, para este caso se uso mas tiempo para la lectura y poco segundos para la escritura. | |
| Captura de pantalla 2024-12-26 a la(s) 17.21.04.png | |
|  |  |

| Análisis del comando mongostat —discover al importar “act-grupal-countries-small.json” | |
| --- | --- |
| La herramienta reportó un promedio de [número] operaciones por segundo durante la importación de la colección. Se registró una inserción inicial y 5 consultas adicionales para validar los datos. La instancia local de MongoDB manejó la carga sin problemas, utilizando solo 1.3% de memoria y un promedio de 0.0% de CPU. El análisis del tráfico de red mostró 3.84 KB de datos entrantes y 91.3 KB de datos salientes. | |
| Captura de pantalla 2024-12-26 a la(s) 17.26.53.png | |
|  |  |

Exportar la base de datos

| Usando comando mongoexport | |
| --- | --- |
| Para validar el funcionamiento de este comando, se siguió la sugerencia de la actividad y se generó el siguiente comando: “mongoexport --db inspections —collection act-grupal-city\_inspections --out inspections.json;” Al obtener el resultado esperado, se adjunta una captura de pantalla para mostrar la operación. | |
| Captura de pantalla 2024-12-26 a la(s) 18.33.12.png | |
|  |  |

Restaurar una base de datos

| Usando comando mongoexport | |
| --- | --- |
| Siguiendo las indicaciones de la actividad, se descomprimió el archivo comprimido 'act-grupal-people.zip', obteniendo el archivo en formato BSON 'people.bson'. A continuación, se ejecutó el comando 'mongorestore people.bson' desde la ruta donde se encontraba el archivo descomprimido para verificar su correcto funcionamiento. | |
| Captura de pantalla 2024-12-26 a la(s) 18.49.55.pngCaptura de pantalla 2024-12-26 a la(s) 18.50.04.png  Captura de pantalla 2024-12-26 a la(s) 18.52.58.png | |
|  |  |

Caso de uso

Restricción de terrazas en Madrid por covid-19. Contratan a vuestro equipo para actualizar las restricciones de ciertos locales y terrazas en Madrid por cuestiones del covid-19. Los datos para actualizar están en el fichero llamado act-grupal-openDataLocalesMadrid.cvs y se os pide que consolidéis dichos cambios en una base de datos MongoDB llamada Madrid con la colección Terrazas. La información del dataset a utilizar lo tenéis en la siguiente URL: Open Data Censo de locales, sus actividades y terrazas de hostelería y restauración (Terrazas).

|  | |
| --- | --- |
| Se genera la Base de datos Madrid y su colección Terrazas. | |
| Captura de pantalla 2024-12-26 a la(s) 19.10.00.png | |
|  |  |