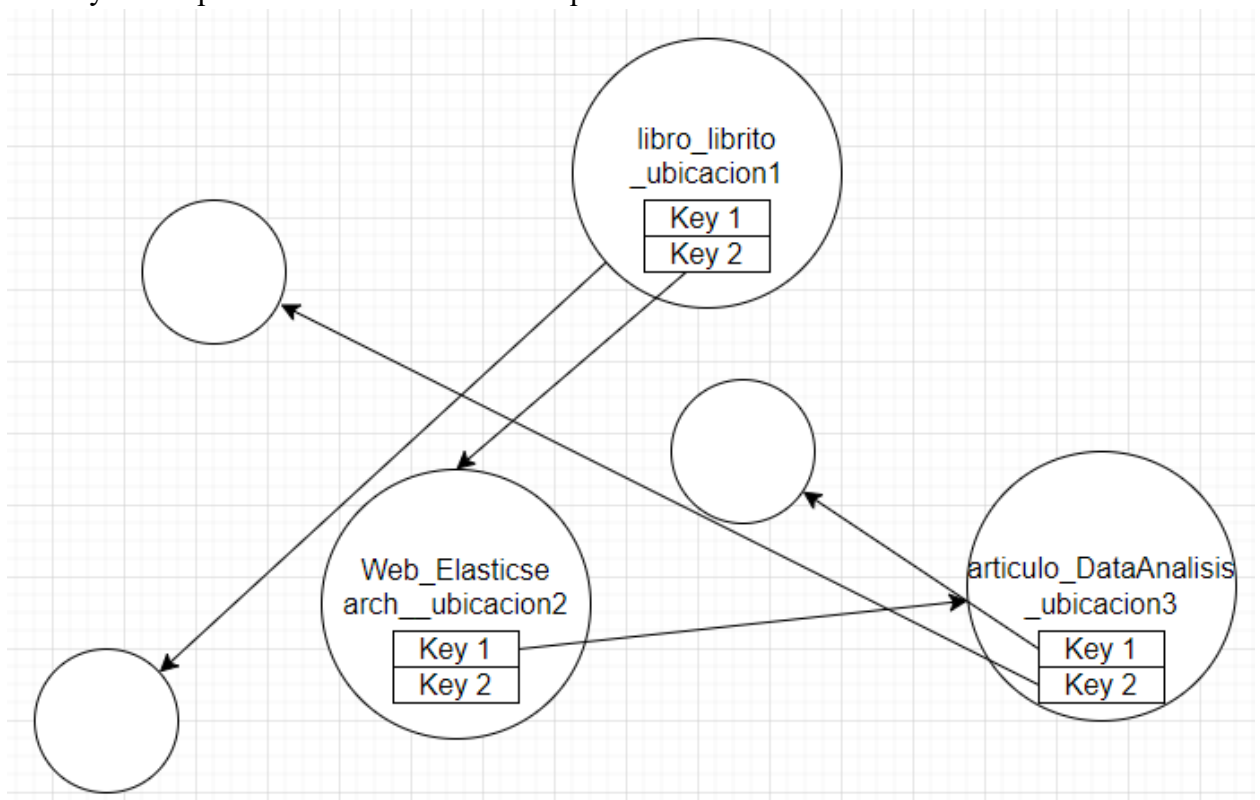


Pregunta 1.

1. Para implementar la navegación entre los distintos elementos utilizaría una base de datos de grafos ya que estas son muy útiles para almacenar y representar relaciones entre datos gracias a las cualidades que tienen por almacenar la información en grafos. Específicamente utilizaría la base de datos Neo4J. Respecto a la información, debido a que se tiene una cantidad masiva de información que ya se almacenará en otra base de datos, en Neo4j solo almacenaría información importante para poder reconocer a que documento pertenecen los datos por ejemplo nombre único del documento creado por tipo_nombre_ubicacion en la base de datos y además los datos o palabras clave que se necesiten para poder realizar las relaciones entre estos documentos que se usarían como llaves y estas apuntarían al documento a la que se relacionan.



2. Para almacenar los elementos de información se utilizaría una base de datos como Elasticsearch ya que gracias a su diseño que implementa un índice invertido permite realizar búsquedas de texto completo muy rápidamente. Elasticsearch distribuye los documentos en contenedores o shards los cuales también están duplicados para poder protegerse contra posibles fallos de hardware. En Elasticsearch se almacenan los datos como documentos JSON los cuales relacionan datos con valores. (Elasticsearch, n.d.)
3. Por medio de Elasticsearch realizaría una búsqueda que el usuario deseara, una vez que se tiene el o los documentos posibles el usuario seleccionaría el documento deseado y al cargarlo por se realizaría una llamada Node4j para obtener el documento, sus keys y las relaciones para poder utilizar las mismas para poder abrir los documentos relacionados en

caso de ser necesario.

Por ejemplo, si el usuario abre el libro librito 1 se cargarían las keys y apuntadores de dicho archivo y en caso de seleccionar el key2 se abriría la información de la web Elasticsearch

4. Gracias a que Elasticsearch permite la distribución de los datos en shards hace que se tengan diferentes puntos de acceso a la información almacenada. Además, de que se reparten los datos se pueden hacer copias de los shards para tener duplicidad de datos y de esta manera estar protegidos en caso de la pérdida de alguno de los shards.
5. Gracias al uso del índice invertido implementado en Elasticsearch se logran realizar búsquedas en los documentos en tiempos que no son perceptibles para el usuario y ya que se tienen los datos de los archivos conseguir el nombre del grafo de Neo4j tampoco consumiría demasiado tiempo por lo que se podría mantener un tiempo de respuesta muy bajo. (Elasticsearch, n.d.)
6. Al implementar la localidad se podría saber que documentos es factible que el usuario vaya a acceder según el documento que este realizando actualmente. Gracias a que se conocen estos documentos se pueden cargar los mismos a la cache para que así el momento en que el usuario los va a acceder no haya que gastar tiempo en ir hasta memoria a buscar los datos lo que reduce el tiempo necesario que se necesita para tener el documento desplegado.

Pregunta 2

¿Qué índices definiría para aumentar la velocidad de todo el sistema? Tome en cuenta todos los tipos de índices estudiados en el curso.

1. ¿Qué índices definiría para aumentar la velocidad de todo el sistema? Tome en cuenta todos los tipos de índices estudiados en el curso.
Debido a que los queries que más carga producen en la base de datos son los 3 señalados podría implementar un índice solo la columna name de la tabla artista, otro sobre la columna name de la tabla álbum. Ya que estas dos columnas son necesarias en el ultimo query no se implementarían si no que se utilizaría solo el índice que se hablara mas adelante ya que es mejor tener un índice con columnas extra que varios índices definidos sobre una sola columna ya que esto disminuye la velocidad de la consulta. Ahora bien para el ultimo query se aplicaría el índice sobre las dos columnas anteriores además de sobre la columna nombre de la tabla álbum a este índice se le aplicaría un include para tener las columnas extra de la consulta para evitar hacer los JOINS al momento de hacer las consultas.
2. ¿Qué base de datos SQL o NoSQL recomendaría para reemplazar la base de datos actual? Justifique su respuesta.
Para esta base de datos se podría implementar una base de datos NoSQL como lo es mongodb. Se escoge una base NoSQL ya que debido al contexto de uso de la base de datos no es tan importante mantener una alta consistencia o la atomicidad. Además, se escogió mongodb ya que mongo está preparada para realizar escalamiento horizontal y permite realizar rápidas lecturas lo que ayudaría de manera significativa en este contexto en el que no se necesitan hacer tantas escrituras, pero si una gran cantidad de lecturas
3. ¿Existirá alguna otra forma de mejorar el rendimiento de la base de datos relacional en especial para la tercera consulta? Comente.
Una opción para mejorar el rendimiento de la base de datos es realizar (aún más) desnormalización de la base de datos para crear tablas que contengan todos los datos necesarios y reducir la cantidad de JOINS utilizados en las consultas.
Además, se deberían de revisar los queries que se están utilizando para poder asegurar de que estos sean lo más optimo posibles.

Pregunta 3

Como primer punto para verificar que solo se realicen conexiones desde los puntos validos y por los puertos correspondientes se deberían de establecer firewalls en todos los puntos desde la read privada, la instancia, el motor de la BD la red de almacenamiento y el disco.

Por otra parte, se deben de encriptar los datos desde que salen de la aplicación y de preferencia mantenerlos encriptados en todo el proceso incluso al memento de que llegan al motor de la base de datos y al disco para así evitar que los datos puedan ser leídos en caso de que logren ser interceptados en algún momento.

Se debe de asegurar mantener el software del motor de base de datos actualizado para así evitar cualquier vulnerabilidad de seguridad que no haya sido corregida en versiones antiguas de la misma.

Se deben establecer contraseñas seguras que no sean fáciles de obtener por algún método. Además, es importante eliminar los usuarios con permisos de administrador que están configurados por defecto y configurar los usuarios administradores propios para evitar que un atacante pueda ingresar al sistema por medio de estas credenciales “default”.

Se deben agregar sistemas de monitorización en la mayor cantidad de componentes para así poder detectar cualquier acceso indebido o no usual en el tráfico.

Pregunta 4

1. ¿Por qué las bases de datos de series de tiempo son tan utilizadas en soluciones de Observabilidad? Realice un análisis desde el punto de vista de la naturaleza de los datos que se recolectan.

En la observabilidad se busca tener toda la información en tiempo real sobre la información de la base de datos como lo son los datos de rendimiento, métricas, registro y eventos. Estos datos se van generando constante mente durante la ejecución haciendo que se conviertan en datos de series de tiempo. (Amy, 2021)

Las bases de datos de series de tiempo están diseñadas específicamente para manejar y analizar datos temporales eficientemente ya que estas tienen estructuras de almacenamiento y algoritmos que sirven muy bien con datos generados en intervalos regulares. Esto permite que se puedan hacer consultas y análisis de los datos basado en las marcas de tiempo para poder detectar patrones, anomalías y tendencias de los datos. (Amy, 2021)

2. ¿Es posible utilizar BigTable como una base de datos de series de tiempo que se pueda utilizar como parte de una solución de Observabilidad? Justifique su respuesta desde el punto de vista de la naturaleza de la base de datos.

Sí, es posible utilizar Bigtable como una base de datos de series de tiempo en una solución de Observabilidad. Bigtable es un almacén de claves/columnas anchas que funciona bien para el almacenamiento y consulta eficiente de datos de series temporales. (Google Cloud, 2022)

La naturaleza de Bigtable, junto con su compatibilidad con la API de HBase, lo hace adecuado para el almacenamiento de datos de series de tiempo. La API de HBase es utilizada por OpenTSDB, que es una herramienta popular para el almacenamiento y análisis de datos de series de tiempo. Esto significa que se puede utilizar OpenTSDB junto con Bigtable para almacenar y consultar datos de series de tiempo de manera eficiente. (Google Cloud, 2022)

3. Suponiendo que tenemos una solución de Observabilidad que utiliza Elasticsearch, ¿Cómo podemos ahorrar dinero con información histórica?

En Elasticsearch se puede implementar una arquitectura con nodos hot, warm y cold los que tiene precios diferentes siendo el hot el mas caro y el cold el mas barato ya que los hot permite tener la información mas “a la mano” y por su parte los cold tienen información que esta almacenada en un lugar donde es poco probable que se utilice por lo que es más barato tenerlo. (Jake, 2022)

En relación con la información histórica los datos se podrían mover entre los diferentes nodos disponibles, haciendo que los datos mas recientes se mantengan en nodos hot para poder acceder rápido a la información y según los datos se van haciendo mas viejos se van moviendo a los nodos warm o cold ya que estos van a ser cada vez menos solicitados. Este movimiento de los datos a nodos mas baratos que los nodos hot hace que se genere un

ahorro económico ya que de mantener todos los datos en nodos host seria muy costoso de mantener

4. Comente las ventajas y las desventajas de utilizar un servicio de Observabilidad on-premise (por ejemplo, Prometheus y Grafana) vs un Managed Service (como Datadog), justifique su respuesta con la experiencia obtenida en la tarea corta 1 de este curso.

On-premise

ventajas

- Los datos se mantienen privados ya que no tienen que pasar por ningún tercero porque las herramientas están instaladas en equipos propios del dueño de los datos.
- Ya que es el usuario el que hace la instalación tiene completa libertad de como quiere configurar la herramienta de Observabilidad
- Ya que la herramienta esta instalada en maquinas del usuario no se deben pagar cuotas extra por el uso de la herramienta o del equipo

Desventajas

- Ya que el hardware es del usuario es más difícil realizar el escalamiento de los servicios ya que sale mas costoso realizar la compra del equipo necesario para realizarlo.
- Aunque no se pagan costos extra mensuales, pero si se debe hacer una inversión inicial para poder adquirir el equipo donde se instalaran las herramientas en caso de no tener el equipo.
- Si no se cuenta con una persona con conocimiento previo en la configuración e instalación de estas herramientas es mas dificultoso realizar estos procesos

Managed Service

Ventajas:

- Usualmente los proveedores poseen una fácil configuración de las herramientas reduciéndose en ocasiones a unos cuantos clicks lo que permite una implementación rápida y fácil.
- Gracias a que es el proveedor el encargado del hardware la escalabilidad es muy sencilla para el usuario ya que solo debe de solicitar el equipo extra y el proveedor es el que se encarga de proporcionar el equipo y de realizar la instalación y configuración de las herramientas
- Otra ventaja de usar managed services es que el usuario no debe preocuparse por las actualizaciones de software o el mantenimiento del equipo.

Desventajas

- No siempre se pueden aplicar las configuraciones que se deseen ya que los proveedores tienen una cantidad limitada de opciones que no necesariamente se adecuan a las necesidades del usuario
- A pesar de que el usuario no debe de preocuparse por la compra inicial del equipo, este si debe de realizar un pago mensual para poder seguir recibiendo el servicio.

REFERENCIAS

Amy, R. (2021, Noviembre 30). *¿Qué es la observabilidad?* Retrieved from New Relic:
<https://newrelic.com/es/blog/best-practices/what-is-observability>

Elasticsearch. (n.d.). *Elasticsearch*. Retrieved from ¿Qué es Elasticsearch?:
<https://www.elastic.co/es/what-is/elasticsearch>

Google Cloud. (2022, Enero 12). *Monitoring time-series data with OpenTSDB on Cloud Bigtable and GKE : Cloud Architecture Center : Google Cloud*. Retrieved from Google Cloud: https://cloud-google-com.translate.goog/architecture/monitoring-time-series-data-opentsdb?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#:~:text=Bigtable%20is%20a%20key%2Fwide,Apache%20HBase%2C%20such%20as%20OpenTSDB.

Jake, L. (2022, Febrero 23). *Implementación de arquitectura caliente-tibia-fría en Elasticsearch con gestión de ciclo de vida del índice*. Retrieved from Elastic Blog:
<https://www.elastic.co/es/blog/implementing-hot-warm-cold-in-elasticsearch-with-index-lifecycle-management>