Fundamentals of Databricks lakehouse

viernes, 21 de julio de 2023 14:17

Data Lake: Datos estructurados, semi-estructurados y no estructurados pueden vivir en el mismo espacio siendo recolectados a gran velocidad.

Cons: No soportan transaccionalidad y no pueden garantizar calidad de los datos. Desafios a la hora de gobernar sobre los datos en temas como seguridad y privacidad.

LakeHouse: Poder almacenar todo tipo de datos en una sola fuenta, una unica fuente de verdad para poder dar datos tanto al BI como al ML.

Servicios:
Delta Lake: Data Lake Foundation
Unity Catalog: Gobierno para AI y datos.

Ventajas:
Unifica el data warehousing con los modelos de IA con el propósito de dejar de lado posibles problemas, como: silos de datos, estruturas complejas y complejidad a la hora de gobernar sobre los datos.

Es multinube.

Fiabilidad y performance de los datos:
Garbage data in = garbage data out.

Data lakes contras:

1. Data lakes tienen un contra a la hora de ofrecer performance y fiabilidad de los datos en comparacion con un data warehouse.
2. Sin soporte para operaciones ACID.
3. No se establece un esquema para los datos, haciendo que la calidad de los mismos sea mala.
4. Mala integracion con un catalogo de datos.
5. Tener archivos muy pequeños (Daña la performance de los queries)

Delta Lake: Formato de guardado de archivos open source, garantiza operaciones ACID (sin archivos corruptos o a la mitad), escalabilidad de los datos y manejo de metadatos, historial y viajes en el tiempo (revisar estados de tablas anteriores, rollbacks), log de cambios efectuados en la data, enforzamiento de esquema y evolucion, soporte para borrado, actualizacion y fusionado de datos.

Delta Lake transaction log: Contiene un historico (log) de todas las transacciones y operaciones ejecutadas en los datos. Actúa como fuente de la verdad para siempre proveer la información más actualizada. Cuando se hace un query con Spark, Spark mira el log de transacciones y actualiza el log y la tabla de ser necesario garantizando el uso de la última versión de los datos.

Photon: Es un motor de queries que permite mejorar el performance a la hora de ingestar la data, hacer streamings de datos, queries a los datos del data lake. Es un motor diseñado especialmente para LakeHouse.

GOBIERNO DE DATOS:

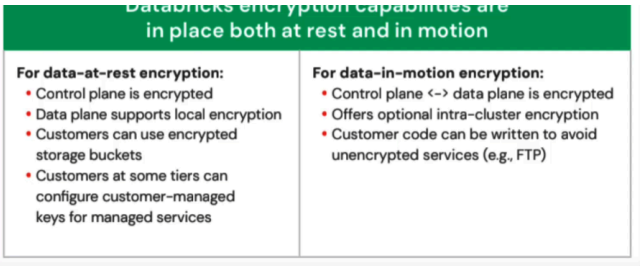
Unity Catalog: Solución para el gobierno de los datos.
Se puede hacer uso de ANSI SQL para poder definir accesos a los datos.
Se pueden definir permisos sobre las filas y columnas que ciertos usuarios pueden o no consultar.
También da un catálogo para auditar las acciones sobre los datos ¿Quién hizo qué sobre qué datos?

Los conjuntos de datos poseen un linaje definido de datos: De donde vinieron los datos, que transformaciones se hicieron, que datos se juntaron, y demás están en este linaje.

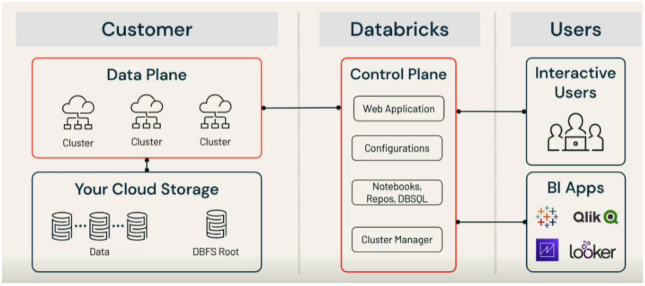
Delta Sharing: Como solución para poder compartir seguramente datos en vivo.
Los dueños de los datos siguen teniendo el control con la habilidad de auditar y ver el uso que se hace de sus datos compartidos.
Posee una integración con PowerBI, Tableu, Spark y demas.
Es un protocolo REST que da el acceso a una parte de los datos de una base de datos en la nube.

SEGURIDAD:
Plano de control:
Servicios backend conectados a la nube de donde estes usando la plataforma que Databricks provee, todo este plano lo corre Databricks en su plataforma. La metadata y las listas de acceso de control a los datos se almacenan aquí.
Plano de datos: Donde son procesados tus datos, los recursos de compute corren en la nube donde estes usando la plataforma de Databricks.

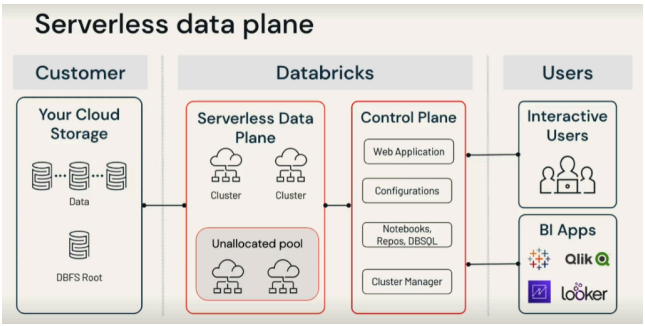
Los clústeres de Databricks no guardan información una vez destruidos.



Arquitectura de DataBricks:
Normal Data Plane:



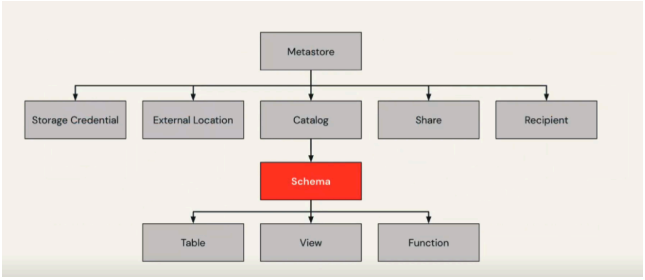
Serverless Data Plane:



Por el momento solo esta disponible para el uso de SQL y se llama Databricks serverless SQL
Son clusteres que Databricks se encarga de manejar en su misma cuenta de nube, escalando y de escalando automáticamente.
Una vez que la tarea termina los recursos son liberados.

MANEJO DE DATOS:

Unity Catalog: Un espacio para gobernar todos los espacios de trabajo en diferentes nubes. Un espacio para poder controlar y organizar la metadata y los datos. Es una referencia a una colección de metadatos y a un link a un contendor donde se almacenan los mismos.

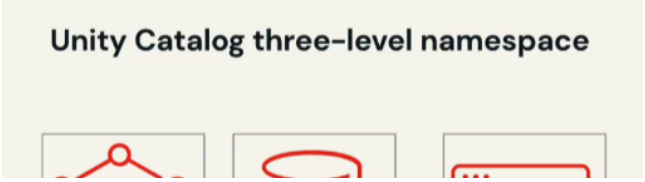


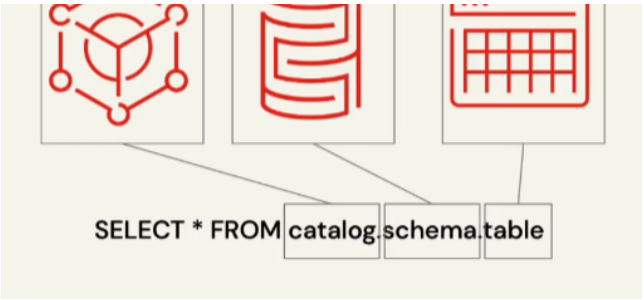
MetaStore: Constructo que representa la metadata.
Metadata: Información de los objetos que están siendo almacenados y manejador por la metastore.

Catalog: Contendor superior para los datos en Unity Catalog. Una MetaStore puede tener n Catálogos.

Ya que el catálogo es el primero en la jerarquía para hacer referencia a los datos guardados dentro del catálogo hacemos una referencia de 3 niveles.

Da un tercer nivel con el propósito de mejorar la segregación de los datos.





Schema: Contendor para activos de datos como tablas y vistas. Los catálogos pueden contener n esquemas.

Tablas: En Databricks las tablas se diferencias por dos elementos: metadata, los datos (filas y columnas de la tabla)
Managed Tables: Los datos son almacenados en el metastore de Databricks.
External Tables: Los datos son almacenados por fuera de Databricks. Ç

Views: Son almacenar los datos salientes de la ejecucion de un querie. Read only. No tienen la habilidad para modificar los datos de base.

User defined functions: Encapsular una funcionalidad en una funcion que puede ser llamada por medio de queries.

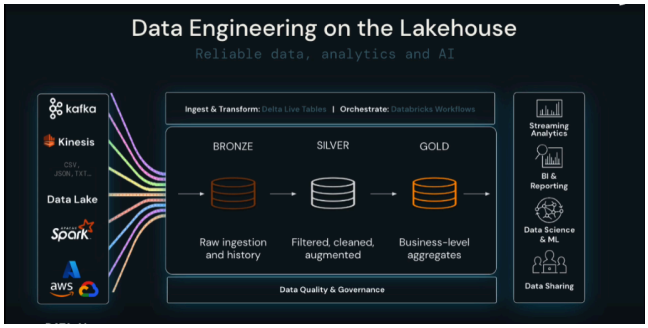
Storage Credentials: Se usan para poder indentificarse con almacenamientos externos o internos.

External location: Proveer acceso de control a nivel de archivo.

Share: Usado por Delta Sharing. Declarar de forma explícita datos de solo lectura de las tablas. Se pueden compartir con n recipients dentro o fuera de la organización.
Recipient: Son usados por Delta Sharing.

Data engineer:

- Revisiones de la calidad de los datos que fluyen por las ETLS pueden ser programados dentro de DataBricks
- Se ofrece observalidad de los pipelines, poder ver el estado y salud del pipeline de datos.



El uso de esta arquitectura es un patron con el proposito de mejorar la calidad de los datos.

A medida de que los datos llegan al lakehouse, Databricks infiere el esquema y lo evoluciona.

Auto Loader: Herramienta de ingesta de datos, que detecta el esquema de los datos y los hace cumplir con el mismo.

Se puede llevar toda una carpeta a una tabla Delta Lake por medio del comando COPY INTO SQL

Delta Live Tables: Framework ETL que usa una escritura declarativa. Trabaja tanto con streaming como con bach y se programa tanto en SQL como en Python.

Databricks WorkFlows: herramienta para crear y gestionar flujos de datos con cualquier herramienta de cualquier nube desde Databricks.

DATA STREAMING:
Casos de uso:

- Real time ML
- Real time analytics
- Real time applications

DATA SCIENCE:
MlFlow: Servicio para monitorear el entrenamiento de los modelos, asi como empaquetar y reutilizar los modelos facilmente

AutoML: Entrena y hace el feature tunning automaticamente de modelos de ML, reporta las métricas de resultados de los modelos y el código (parametros) usado para entrenar el modelo.