**¿Qué experiencias has tenido como ingeniero de datos en AWS? ¿Cuál ha sido el proyecto más retador, y por qué?**

Se presento la necesidad de cambiar toda la estructura de datos que ya se tenia montada para responder a un proyecto, dadas las necesidades del procesamiento y del negocio mismo.

Cuando esto se presenta toda la arquitectura de datos se encontraba montada por parte del equipo de desarrollo con tablas cargadas como vistas y vistas materializadas formadas por las vistas, además de tener una dependencia absoluta del dwh para poder contar con información.

Conforme con lo anterior lo primero que teníamos que hacer era encontrar la forma remover la dependencia de la aplicación del DWH, ya que desde allí se procesa la data y se entrega a la aplicación por medio de una trasferencia de datos manual, pero además de esta fuente, la aplicación requiere de muchas otras fuentes para su correcto funcionamiento y un procesamiento bastante denso en cada una de ellas.

Dado lo anterior y tomando en consideración las necesidades completas de la aplicación para su funcionamiento y las necesidades del negocio, nos sentamos a diseñar una infraestructura que pudiera soportar todo el procesamiento de datos y el mejo de errores de los diferentes source para garantizar tanto la disponibilidad como la continuidad del servicio, además de asumir todo el procesamiento de los datos y las reglas de negocio requeridas para poder garantizar la normalización de funcionamiento de la aplicación generando un procesamiento y persistencia de los datos para cada source dentro de la misma aplicación ya que esta los requiere.

Esto implico generar en principio un entendimiento del proyecto, además de un entendimiento de la arquitectura ya implementada y la lógica de la implementación ya realizada para poder modelar teniendo en cuenta todo el funcionamiento actual para no generar mucho impacto

**2. ¿Que estrategias has aplicado para crear los recursos necesarios en AWS para mantener una**

**arquitectura y pipelines de datos?**

Siempre que comienzo un diseño lo que considero más importante es contar con el entendimiento en general del proyecto, además de sus reglas de negocio y necesidades de funcionalidad, persistencia y funcionamiento del proyecto.

Conforme con lo anterior me parece fundamental implementar considerando la escalabilidad, los costos, rendimiento y seguridad del proyecto y para ello tengo en cuenta crear

* Diseños modulares y escalables como (CloudFormation o Terraform permiten)
* Gestionar los datos y el almacenamiento por data lake con niveles jerárquicos y políticas para manejo de costos, además de configuraciones de compresión y particionamiento.
* Automatizar y Orquestar los flujos
* Procesamiento de los datos según la necesidad bien sea en Bach o en tiempo real según se requiera.
* Seguridad y Gobernanza de datos gestionando accesos con políticas a usuarios y servicios e implementando procesos de Cifrados y monitoreos
* Poder asegurar una alta disponibilidad y escalabilidad de los datos.

**3. ¿Qué consideraciones tomarías al decidir entre almacenar datos en Amazon S3, RDS o Redshift?**

Creo que todo depende de como se usen los datos y para que se usan, por ejemplo, si la data de logs de auditoría, control de cambios histórico con datos crudos y en diferentes formatos, preferiría almacenar en s3 por flexibilidad, escalabilidad, bajo costo, disponibilidad.

Sin embargo, si hablamos de algo para aplicaciones transaccionales, en este caso sería mejor utilizar un RDS con data estructurada para menor latencia y gestión automática, pero si por el contrario.

Por otro lado, si hablamos de procesos para Realizar análisis de datos con consultas muy complejas de inteligencia de negocios o reportaría o ciencia de datos con grandes volúmenes de datos, es mejor usar Redshift.

**4. ¿Qué beneficios y desventajas ves al utilizar AWS Glue en comparación con Lambda o Step Functions para orquestación ETL?**

Como he dicho antes, todo depende de la finalidad o uso del procesamiento a implementar, ya que usar glue implica muchos beneficios si vamos a procesar grandes volúmenes de datos y requerimos muchas transformaciones sobre los datos, ya que, si bien es más costoso que una lambda, la lambda no podría realizar dicho procesamiento por tiempos y capacidad.

ahora bien, si los procesamientos cortos y sin mayor complejidad, en este caso usar glue no sería una buena opción

por otro también podríamos usar **Step Functions** para un procesamiento complejo y de gran cantidad de datos, pero los costos serian mucho mas altos

**5. ¿Cómo garantizarías la integridad y seguridad de los datos de un datalake construido en Amazon S3?**

para poder garantizar la integridad y seguridad de los datos de un datalake en S3, necesitamos una combinación de configuraciones de seguridad, monitoreo y buenas prácticas como:

* **Seguridad de los Datos**
* Roles y políticas de IAM (principio de menor privilegio).
* Accesos específicos solo a los buckets o prefijos requeridos.
* Restringir el acceso al Bucket basado en IP, origen VPC o etiquetas de recursos.
* Bloqueo de Acceso Público a nivel de bucket
* Configuracion AWS Lake Formation: para gestionar el acceso granular a tablas y datos dentro del Data Lake.
* **Integridad de los Datos**
* Versioning (Control de Versiones): versionado en los buckets en caso de cambios no deseados o eliminaciones accidentales.
* S3 Object Lock: configuración modo WORM (Write Once Read Many), lo que protege los objetos contra eliminaciones o modificaciones.
* Replication Rules: replicación entre regiones para redundancia y recuperación ante fallos.
* Backups: aunque S3 tiene redundancia integrada, considera backups adicionales para datos críticos usando replicación o servicios externos.
* Lifecycle Policies: Implementa reglas de ciclo de vida para mover datos infrecuentes a clases de almacenamiento como S3 Glacier y automatizar su archivo seguro.
* Rendimiento y Costos
* Usar diferentes clases de almacenamiento para balancear costos y accesibilidad según la frecuencia de acceso.
* Diseñar particiones lógicas para optimizar consultas (e.g., por fecha, región o categoría).
* Capacitación y Cultura
* Capacita a los equipos en seguridad y manejo adecuado de los datos.
* Implementa procedimientos claros para gestionar accesos y cambios.
* Revisión Regular:
  + Realiza auditorías frecuentes de políticas, configuraciones y registros de actividad.