**Documentación del Pipeline de Datos para la Compañía Comercializadora de Energía**

**1. Descripción General**

El presente documento describe la implementación de un pipeline de datos para una compañía comercializadora de energía utilizando servicios de AWS. El pipeline procesa información exportada en formato CSV desde el sistema transaccional de la compañía. La infraestructura fue implementada mediante Terraform, asegurando una solución automatizada, escalable, reproducible y mantenible.

**2. Arquitectura del Pipeline**

A diagram of a data flow

Description automatically generatedEl pipeline se basa en una arquitectura de DataLake en S3, organizada en tres capas: Raw, Processed y Analytics. Cada capa contiene datos estructurados y transformados según su etapa en el flujo de datos.

**Componentes Clave:**

1. **S3 Buckets:**
   * **Raw:** Almacena los archivos CSV originales, organizados en particiones por dominio y fecha.
   * **Processed:** Almacena los datos transformados en formato Parquet**.**
   * **Analytics:** Almacena resultados de consultas realizadas en Athena.
2. **AWS Glue:**
   * **Crawlers:** Catalogan los datos en Raw y Processed.
   * **Jobs:** Transforman los datos de Raw a Processed.
3. **AWS EventBridge y Lambda:**
   * **EventBridge:** Programa la ejecución automática de Glue Jobs.
   * **Lambda:** Gestiona la ejecución de múltiples jobs y consultas en Athena desde Python.
4. **AWS Athena:**
   * Permite consultas SQL sobre los datos procesados.

**3. Flujo de Datos**

1. **Carga de Datos:**
   * Los archivos CSV se cargan automáticamente en el bucket Raw mediante Terraform.
   * Los archivos se organizan en particiones por dominio (proveedores, clientes y transacciones) y fecha (YYYY/MM/DD).
2. **Catalogación:**
   * Los crawlers de Glue catalogan los datos en Raw y Processed.
   * Se generan tablas en el AWS Glue Data Catalog.
3. **Transformación:**
   * Glue Jobs limpian y transforman los datos, convirtiéndolos a formato Parquet.
   * Los datos procesados se almacenan en el bucket Processed.
4. **Consulta:**
   * Athena consulta los datos procesados para análisis y generación de insights.

**4. Configuración del Pipeline**

**4.1. Automatización**

* **Carga Inicial:** Los datos se cargan automáticamente desde Terraform, la partición de fecha se genera dinámicamente para el día actual.
* **Ejecución Periódica:** EventBridge programa la ejecución diaria de Glue Jobs mediante una Lambda.

**4.3. Glue Jobs**

* **Transformaciones Realizadas:**
  + Limpieza de nombres de columnas y data.
  + Conversión de datos a formato Parquet.
  + Almacenamiento en la capa Processed.

**5. Permisos y Políticas**

**5.1. Roles Clave**

1. **Glue Role:**
   * Asociado a Glue Crawlers y Jobs.
   * Permite acceso a los buckets de S3.
2. **Lambda Role:**
   * Asociado a las Lambdas de ejecución de Glue Jobs y consultas en Athena.
   * Incluye permisos para invocar Glue Jobs y consultar Athena.

**5.2. Políticas IAM**

* Las políticas incluyen permisos específicos para s3, glue, y athena.

**6. Consultas en Athena desde Lambda**

Una Lambda personalizada ejecuta consultas SQL en Athena. Los resultados se almacenan en el bucket Analytics.

**Ejemplo de Payload para la Lambda:**

La query que ejecuta la lambda es modificable dentro del script, se puede automatizar para que reciba un payload en una próxima mejora (como lo hace la lambda que ejecuta los Jobs).

**7. Puntos Adicionales**

1. **Infraestructura como Código:**
   * Terraform se utilizó para implementar y gestionar todos los recursos.
   * El código está organizado, modularizado y automatizado para facilitar su mantenimiento.
2. **No Implementado:**
   * **AWS LakeFormation:** Se decidió no incluirlo en esta prueba por dos motivos: LakeFormation no tiene soporte nativo para Terraform y se prefirió no implementar manualidades por medio de consola o CLI (Se puede implementar con CDK en una próxima mejora).
   * **Redshift:** Por razones de optimización de costos, no se integró un DataWarehouse.
3. **Extensibilidad:**
   * La solución puede ampliarse para incluir LakeFormation y Redshift en el futuro.