DISEÑO Y ARQUITECTURA DEL PIPELINE DE DATOS PARA ANÁLISIS DE TENDENCIAS ESPACIALES

CRISTHIAN CAMILO LÓPEZ PÉREZ

12 de Marzo 2025

Monokera

Bogotá D.C.

Contenido

Diseño y Arquitectura del Pipeline de Datos para Análisis de Tendencias Espaciales

Este documento presenta el diseño y arquitectura de un pipeline de datos para procesar información de la industria espacial usando la API de Spaceflight News. El sistema permite la ingesta diaria de artículos y eventos mediante Lambda, su procesamiento con Apache Spark y Python, y finalmente almacenamiento en Amazon Redshift y/o Postgres

La arquitectura incluye mecanismos de monitoreo, respaldo y recuperación, garantizando la confiabilidad y eficiencia del sistema. A continuación, se detallan los componentes clave, la estrategia de almacenamiento y el plan de contingencia para un flujo de datos robusto y escalable.

Requisitos del Sistema

Objetivos del Pipeline

- Ingesta diaria de nuevos artículos y eventos.
- Clasificación de contenido por temas.
- Almacenamiento histórico para análisis de tendencias.
- Implementación de estrategias de monitoreo y backup.

Requisitos Específicos

- Integración con la API de Spaceflight News para la extracción de datos.
- Capacidad de procesamiento distribuido para análisis de contenido.
- Almacenamiento optimizado para consultas rápidas y análisis.
- Mecanismos de recuperación y alta disponibilidad.

Estimación de Volumen de Datos

- Ingesta estimada de 30+ nuevos documentos diarios.
- Tamaño promedio por documento: 5 KB.
- Volumen diario estimado: ~2.5 MB.
- Volumen anual estimado: ~900 MB - 1 GB.

Diseño de la Arquitectura

Componentes AWS y Justificación

- AWS Lambda: Para la ingesta automatizada desde la API, en desarrollo y ambientes locales se sugiere uso mediante Docker.
- Amazon S3: Almacenamiento de datos crudos y respaldos.
- Amazon Redshift: Almacenamiento de datos procesados para análisis, en desarrollo y ambientes locales se sugiere uso de Postgres SQL.
- Amazon EMR (con Apache Spark): Procesamiento distribuido para análisis de contenido.
- AWS CloudWatch: Monitoreo de la infraestructura.
- AWS Step Functions / Apache Airflow: Orquestación de tareas.
- AWS Backup: Sistema de recuperación ante fallos.

Flujo de Datos

- Ingesta: AWS Lambda extrae los datos desde la API y los almacena en S3.
- Transformación y Procesamiento: Apache Spark analiza tendencias, extrae entidades y clasifica artículos.
- Almacenamiento: Redshift almacena datos procesados.
- Orquestación: Airflow gestiona la ejecución de las tareas del pipeline.
- Monitoreo y Backups: CloudWatch supervisa el sistema y AWS Backup protege la información.

Estrategia de Almacenamiento y Procesamiento

Formatos y Ubicación de Datos

- S3: Datos crudos almacenados en formato CSV/Parquet.
- Redshift/Postgres SQL: Datos procesados en un modelo dimensional optimizado.
- Particionamiento: División de datos por fecha para eficiencia en consultas.

Plan de Contingencia

- Fallback en API: Uso de últimos datos almacenados en caso de fallo en la API.
- Sistema de Retries: Reintentos automáticos en Airflow ante fallos de red.
- Respaldo diario en S3: Copia de datos almacenados en Redshift.
- Alta disponibilidad: Replicación en múltiples zonas de AWS.
- Recuperación ante fallos: Restauración de datos desde backups automáticos.

Monitoreo y Optimización

- AWS CloudWatch: Monitoreo de latencia y errores.
- AWS SNS (Simple Notification Service): Alertas ante eventos críticos.
- Validaciones de datos: Integridad asegurada en Airflow.
- Alarmas para Redshift: Identificación de problemas de rendimiento.

Conclusiones y Siguientes Pasos

Este diseño garantiza una ingesta confiable, almacenamiento escalable y procesamiento eficiente de datos para el análisis de tendencias en la industria espacial. Las futuras mejoras podrían incluir:

- Implementación de modelos de machine learning para análisis predictivo.
- Mayor optimización en el uso de recursos en AWS.
- Automatización de reportes visuales para análisis rápido de tendencias.