📄 Documento Técnico del Proyecto SpaceFlight Data Pipeline

1️⃣ Estimación de Volumen de Datos 📊

Tipo de Datos	Formato	Tamaño Promedio	Estimación Diaria	Volumen Total Estimado (Mensual)
Datos Crudos (API SpaceFlight News)	JSON	1 KB – 5 KB	Articles: ~1,000/díaBlogs: ~500/díaReports: ~300/día	60 MB – 150 MB
Datos Procesados (Parquet)	Parquet	Reducción de ~70%	Procesados y particionados por `published_date`	30 MB – 90 MB

Detalle de Volumen Anual

- **Datos Crudos:** 720 MB – 1.8 GB
- **Datos Procesados:** 360 MB – 1.1 GB

Datos Crudos (API SpaceFlight News)

- **Tipo de datos:** JSON
- **Tamaño promedio por artículo:** 1 KB – 5 KB
- **Estimación diaria:**
  - *Articles:* ~1,000 registros/día
  - *Blogs:* ~500 registros/día
  - *Reports:* ~300 registros/día
- **Volumen total estimado:**
  - **Diario:** ~2 MB – 5 MB
  - **Mensual:** ~60 MB – 150 MB
  - **Anual:** ~720 MB – 1.8 GB

Datos Procesados (Parquet)

- **Formato:** Parquet, particionado por `published_date`
- **Reducción de tamaño:** ~70% en comparación con JSON
- **Volumen estimado procesado:**
  - **Diario:** ~1 MB – 3 MB
  - **Mensual:** ~30 MB – 90 MB
  - **Anual:** ~360 MB – 1.1 GB

2️⃣ Estrategia de Almacenamiento y Búsqueda 🎁

Almacenamiento

- **Amazon S3**
  - **Bucket `spaceflight-data-pipeline`:** Almacena los datos crudos (`raw/`) y procesados (`processed/`).
  - **Bucket `spaceflight-data-results`:** Almacena los resultados de consultas de Athena para visualización en Looker Studio.
- **AWS Glue Catalog**
  - **Estructura del esquema:** Define tablas para facilitar las consultas SQL con Athena.
    - `dim_news_source`: Información de fuentes de noticias.
    - `dim_topic`: Clasificación por temas.
    - `fact_article`: Datos detallados de los artículos, particionados por fecha (`published_date`).

Búsqueda y Consulta

- **Amazon Athena**
  - Consultas SQL sobre los datos procesados en formato Parquet.
  - **Particionamiento:** Mejora el rendimiento al limitar las consultas por `published_date`.

3️⃣ Plan de Contingencia ⚠️

Backup y Recuperación

1. **Backup Automático:**
   - Copias de seguridad periódicas del bucket `processed/` a otro bucket en una **región secundaria** de AWS.
2. **Recuperación:**
   - Restauración rápida desde el bucket de respaldo en caso de pérdida de datos o corrupción.
   - **Comando para actualizar las particiones:**
   - `MSCK REPAIR TABLE fact_article;`

Gestión de Errores en el Pipeline

- **Lambda y Glue Job:** Configuración de reintentos automáticos.
- **EventBridge:** Notificación de errores críticos a **Amazon CloudWatch** para activar alertas.

---

4 Sistema de Monitoreo 📈

Amazon CloudWatch

- **Monitoreo de Lambda:** Logs de ejecución, tiempos de respuesta y errores. Monitorea métricas clave como el **número de errores por ejecución**, la **latencia promedio** y el **número de reintentos**.
- **Glue Job Logs:** Logs del Spark UI para analizar el proceso de transformación de datos y diagnosticar problemas de rendimiento.
- **Alerta de Errores:** Configuración de alertas en CloudWatch que notifican por correo electrónico o Slack en caso de fallos.
- **Monitoreo de Lambda:** Logs de ejecución, tiempos de respuesta y errores.
- **Glue Job Logs:** Logs del Spark UI para analizar el proceso de transformación de datos.
- **Alerta de Errores:** Configuración de alertas en CloudWatch que notifican por correo electrónico o Slack en caso de fallos.

Amazon S3 Metrics

- Monitoreo del uso de almacenamiento, accesos y tasas de error.

Amazon Athena Query History

- Historial de consultas para identificar consultas lentas o mal optimizadas.

Conclusión y Siguientes Pasos

El pipeline está diseñado para manejar volúmenes de datos moderados con escalabilidad en AWS. El sistema de monitoreo y el plan de contingencia garantizan una operación confiable y recuperación rápida en caso de fallos.

Optimización de Consultas en Athena:

1. **Uso de Particiones:** Particionar por `published_date` para reducir el volumen de datos escaneados y mejorar la velocidad de las consultas.
2. **Compresión de Datos:** Utilizar el formato Parquet con compresión para minimizar el almacenamiento y acelerar el procesamiento.
3. **Índices y Filtrado:** Aprovechar columnas clave como `source_id` y `topic_id` para filtrar y ordenar datos más eficientemente.
4. **Historial de Consultas:** Monitorear el historial de consultas en Athena para identificar patrones de uso y optimizar consultas frecuentes.

Siguientes Pasos:

1. Automatización completa del pipeline con **Airflow DAG**.
2. Optimización de consultas en Athena para mejorar el rendimiento.
3. Mejora de la visualización en Looker Studio para agregar gráficos adicionales y alertas de tendencias. El pipeline está diseñado para manejar volúmenes de datos moderados con escalabilidad en AWS. El sistema de monitoreo y el plan de contingencia garantizan una operación confiable y recuperación rápida en caso de fallos.

Siguientes Pasos:

1. Automatización completa del pipeline con **Airflow DAG**.
2. Optimización de consultas en Athena para mejorar el rendimiento.
3. Mejora de la visualización en Looker Studio para agregar gráficos adicionales y alertas de tendencias.
