🚀 Pasos para Ejecutar el Proyecto

1️⃣ Ingesta de Datos desde la API a S3 (Lambda Function)

1. **Función Lambda (`spaceflight_ingestion_to_s3`)**:
   - Extrae datos desde la API `https://api.spaceflightnewsapi.net/v4`.
   - Guarda los datos crudos en **Amazon S3** (`s3://spaceflight-data-pipeline/raw`).
   - Al finalizar, envía un evento a **Amazon EventBridge**.
2. **Configuración de Amazon EventBridge**:
   - Escucha eventos de Lambda cuando la ingesta termina (`SUCCEEDED`).
   - Desencadena el Glue Job automáticamente.

2️⃣ Procesamiento de Datos (AWS Glue Job)

1. **Glue Job (`news-data-processing-job`)**:
   - Procesa los datos desde `s3://spaceflight-data-pipeline/raw`.
   - Realiza el análisis de contenido y tendencias (extracción de palabras clave y conteo por fecha).
   - Guarda los resultados en **Parquet** en `s3://spaceflight-data-pipeline/processed`.
2. **Actualizar Glue Crawler**:
   - Registra las particiones y nuevas tablas en el catálogo de Glue.

3️⃣ Consultas SQL y Visualización (Athena + Looker Studio)

1. **Amazon Athena**:
   - Consulta las tablas procesadas (`fact_article`, `dim_news_source`, `dim_topic`).
   - Optimiza las consultas utilizando particiones (`published_date`).
   - **Resultados de consultas:** Almacenados en `s3://spaceflight-data-results`.
2. **Looker Studio**:
   - Conecta **Athena** a **Looker Studio** para crear dashboards interactivos.
   - Visualiza palabras clave y tendencias de noticias.

4️⃣ Sistema de Monitoreo y Plan de Contingencia

1. **Amazon CloudWatch**:
   - Monitorea la función Lambda y el Glue Job.
   - Configura alertas por errores y latencia alta.
2. **Backup Automático**:
   - Copias de seguridad periódicas del bucket `processed/` a una región secundaria.

📌 Resumen Técnico

- **Pipeline Escalable y Modular**: Maneja datos desde la extracción hasta la visualización.
- **Costo Controlado**: S3 y Athena reducen costos comparados con Redshift o BigQuery.
- **Fácil de Monitorizar y Recuperar**: CloudWatch y Glue Crawler aseguran la continuidad operativa.

Alternativa 1: Apache Airflow (AWS MWAA) 🚀

¿Por qué usar Airflow?

- Ofrece mayor flexibilidad para diseñar flujos de trabajo complejos.
- Fácil manejo de dependencias entre tareas (DAGs).
- Control detallado de ejecución, retries y notificaciones de fallos.

Propuesta de DAG (Directed Acyclic Graph):

1. **Tarea 1:** Ejecutar la función Lambda para extraer datos desde la API y almacenarlos en S3 (`raw/`).
2. **Tarea 2:** Esperar la finalización del evento en S3 (`raw/`) y luego lanzar el Glue Job (`news-data-processing-job`).
3. **Tarea 3:** Registrar las particiones en el Glue Catalog mediante el Glue Crawler.
4. **Tarea 4:** Ejecutar consultas en Athena para verificar los datos procesados y almacenarlos en `spaceflight-data-results`.
5. **Tarea 5:** Notificación por correo o Slack al finalizar el proceso.

Ventajas:

- Gran control sobre el proceso y personalización.
- Historial de ejecuciones con posibilidad de reintento manual.
- Escalabilidad y compatibilidad con AWS (usando MWAA).

Alternativa 2: AWS Step Functions + EventBridge 🔄

¿Por qué usar Step Functions?

- Más simple y directo que Airflow para flujos secuenciales.
- Integración nativa con AWS Lambda, Glue y S3.
- Control de errores y reintentos automáticos.

Propuesta de Flujo:

1. **Estado 1:** Ejecutar Lambda (`spaceflight_ingestion_to_s3`).
2. **Estado 2:** Verificar el estado del proceso (espera y validación).
3. **Estado 3:** Iniciar el Glue Job (`news-data-processing-job`).
4. **Estado 4:** Registrar las particiones con un Glue Crawler.
5. **Estado 5:** Notificación final a CloudWatch o SNS.

Ventajas:

- Bajo costo en comparación con Airflow.
- Fácil integración y despliegue sin necesidad de configurar servidores.
- Perfecto para procesos secuenciales y no tan complejos.

¿La Alternativa recomendada por costos?

1. **Para un proyecto de prueba (costos reducidos):**
   AWS Step Functions + Lambda y EventBridge.
   - **Costo:** Muy bajo (sin infraestructura permanente).
   - **Simplicidad:** Fácil de mantener y escalar para pruebas y proyectos pequeños.
2. **Para producción y procesos más complejos:**
   Apache Airflow (MWAA).
   - **Costo:** Mayor, pero con gran control y visibilidad de todo el flujo.
   - **Ideal:** Para múltiples dependencias, tareas concurrentes y notificaciones avanzadas.