DIAGRAMA DE ARQUITECTURA PROYECTO CONSUMO API SPACEFLIGHT

1. Introducción

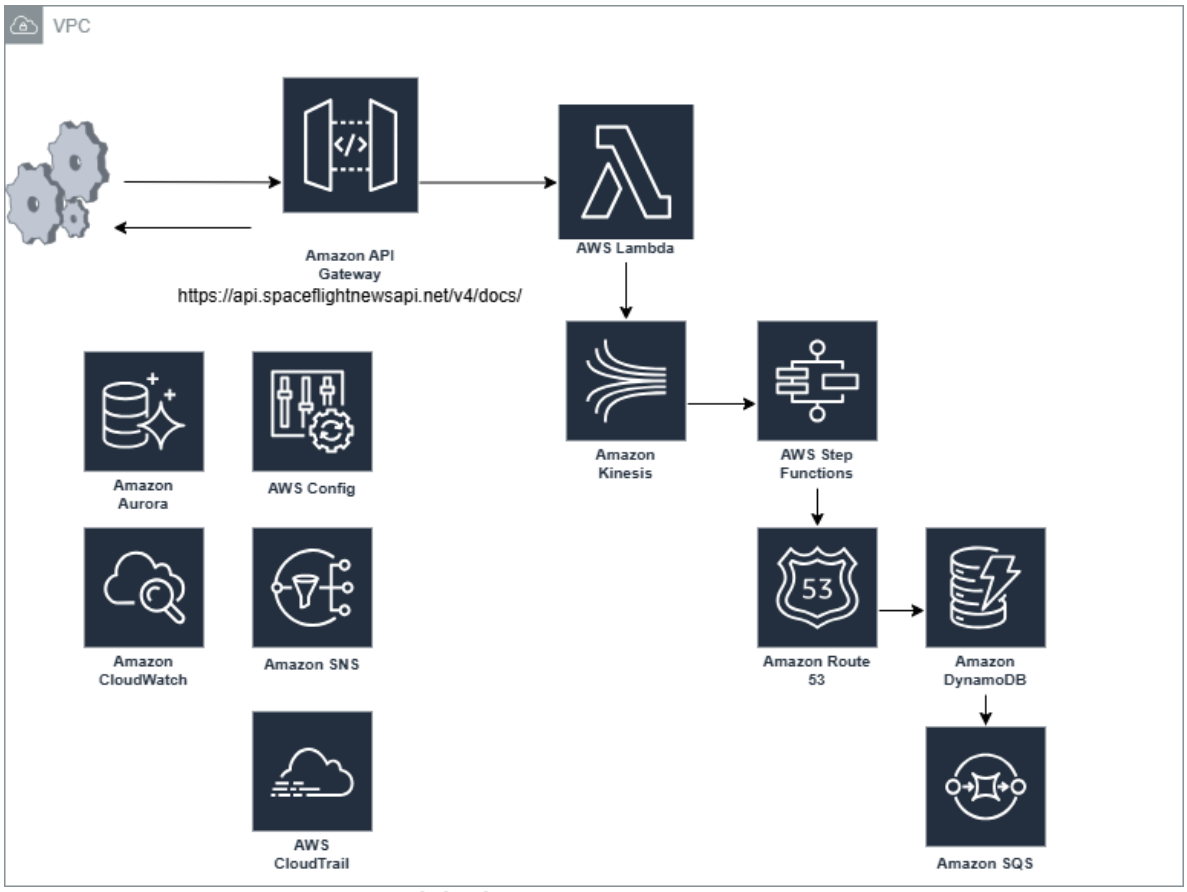
Este documento detalla los componentes de tecnología AWS requeridos en la arquitectura, también el flujo de datos detallado del sistema, diseñado para extraer, procesar y analizar datos desde la API de Spaceflight News. Incluye la estructura de datos, etapas del flujo, almacenamiento, procesamiento y análisis avanzado.

De igual manera se contempla un sistema de backup y recuperación ante incidentes que permita brindar respaldo y disponibilidad de la información.

2. Arquitectura y flujo de componentes AWS

Los componentes que conforman la infraestructura diseñada para el sistema encargado de realizar la respectiva ingesta diaria de nuevos artículos y eventos generados en la API de Spaceflight News son los siguientes:

- **Source**: Es la fuente de información del sistema, la API https://api.spaceflightnewsapi.net/v4/docs/#/
- **API Gateway**: para exponer y gestionar el endpoint del API.
- **Lambda**: Para procesamiento sin servidores.
- **DynamoDB** o **RDS**: Para almacenamiento.
- **S3**: Para almacenar archivos o datos procesados.
- **Kinesis**: Para el procesamiento de flujos de datos en tiempo real.
- **Step Functions**: Para orquestar procesos complejos.
- **IAM**: Para gestionar roles y políticas de seguridad.
- **CloudWatch**: Para monitoreo y registros.
- **VPC**: Para redes seguras y privadas.
- **SNS** o **SQS**: Para mensajería entre componentes.



Flujo de Componentes

3. Flujo de Datos Detallado para el Sistema de Extracción y Análisis de Spaceflight News API

A continuación, se detalla el flujo de datos del sistema diseñado para extraer, procesar y analizar datos desde la API de Spaceflight News. Incluye la estructura de datos, etapas del flujo, almacenamiento, procesamiento y análisis.

3.1. Componentes Principales

3.1.1 Fuente de Datos

- **Origen**: Spaceflight News API.
- **Endpoints**:
  - /articles: Artículos con paginación.
  - /blogs: Blogs con paginación.
  - /reports: Reportes con paginación.
  - /info: Metadatos sobre la API.
- **Frecuencia de Consulta**: Cada 24 horas para datos nuevos.

3.1.2. Sistema de Ingesta

- **Aplicación**: Python-based extractor.
- **Procesos clave**:
  - Paginación eficiente.
  - Manejo de rate limits.
  - Deduplicación de artículos.
- **Formato de Almacenamiento Inicial**: JSON almacenado en Amazon DynamoDB.

3.1.3. Almacenamiento Inicial

- **Destino**: Amazon DynamoDB.
- **Estructura**:
  - Tabla por tipo de datos (articles, blogs, reports).
  - Índices secundarios para consultas rápidas.

3.1.4. Procesamiento

- **Plataforma**: Apache Spark.
- **Operaciones principales**:
  - Extracción de palabras clave y temas principales.
  - Identificación de entidades (compañías, personas, lugares).
  - Clasificación de artículos por tema.
  - Análisis de tendencias por tiempo.
  - Identificación de fuentes más activas.
- **Salida**: Resultados almacenados en DynamoDB.

3.1.5. Almacenamiento Analítico

- **Destino**: Amazon DynamoDB.
- **Estructura**: Tablas diseñadas para consultas analíticas con los siguientes esquemas:
  - dim_article (artículos transformados).
  - dim_keywords (palabras clave).
  - dim_entities (entidades).
  - fact_trends (tendencias por tiempo).

3.1.6. Consulta y Análisis

- **Herramientas**: DynamoDB para consultas rápidas, Redis para caching de resultados frecuentes.
- **Interfaz de usuario**: Reportes y dashboards creados con herramientas como Tableau o Power BI.

3.2. Flujo de Datos Detallado

3.2.1. Paso 1: Extracción de Datos

- **Input**:
  - URL del endpoint.
  - Parámetros para paginación (limit, offset).
- **Transformación**:
  - Recolección de datos brutos desde la API.
  - Almacenamiento en DynamoDB en formato estructurado.
- **Output**:
  - Registros en tablas de DynamoDB.

3.2.2. Paso 2: Almacenamiento Inicial

- **Input**:
  - Registros brutos en DynamoDB.
- **Transformación**:
  - Validación de datos.
  - Limpieza y normalización.
  - Deduplicación basada en id.
- **Output**:
  - Datos limpios en tablas específicas de DynamoDB.

3.2.3. Paso 3: Procesamiento

- **Input**:
  - Datos limpios desde DynamoDB.
- **Transformación**:
  - **Palabras clave**:
    - Extracción utilizando spaCy.
    - Almacenamiento en la tabla dim_keywords.
  - **Clasificación**:
    - Identificación de temas (lanzamientos, investigación, general).
    - Almacenamiento en dim_article.
  - **Entidades**:
    - Identificación de nombres de compañías, personas y lugares.
    - Almacenamiento en dim_entities.
  - **Tendencias**:
    - Análisis de la frecuencia de temas por tiempo.
    - Almacenamiento en fact_trends.
- **Output**:
  - Resultados procesados en tablas dedicadas de DynamoDB.

3.2.4. Paso 4: Consulta y Análisis

- **Input**:
  - Tablas en DynamoDB y caché en Redis.
- **Transformación**:
  - Consultas analíticas para tendencias.
  - Dashboards para visualización.
- **Output**:
  - Respuestas optimizadas para los usuarios finales.

3.3. Detalles Técnicos Adicionales

3.3.1. Particionamiento de Datos Históricos

- **Criterio**: Fecha de publicación (published_at).
- **Ventaja**: Consultas más rápidas y costos reducidos en almacenamiento.

3.3.2. Caching de Resultados Frecuentes

- **Herramienta**: Redis.
- **Estrategia**:
  - Almacenar resultados de consultas frecuentes como tendencias semanales.
  - TTL (Time to Live): 24 horas.

3.3.3. Monitoreo

- **Logs**:
  - Detalles de extracción, errores, y tiempo de ejecución en AWS CloudWatch.
- **Alertas**:
  - Configuración de métricas en CloudWatch para identificar fallos en Spark o la API.

3.4. Conclusión

Este flujo de datos detalla cómo se gestionan los datos desde su extracción hasta su análisis final. La arquitectura asegura eficiencia, escalabilidad y capacidad de análisis profundo.

4. SISTEMA DE BACKUP Y RECUPERACION

4.1 Respaldo Automático:

- Datos críticos en DynamoDB y S3 se respaldan automáticamente con **AWS Backup**.

4.2 Recuperación ante Desastres:

- Base de datos replicada con **Global Tables** o **Aurora Global**.
- Configuración de recuperación en otra región con balanceo de carga mediante **Route 53**.

4.3 Supervisión:

Logs y métricas en CloudWatch y alarmas para notificaciones.

4.4 Añadir Respaldo Automático

Configura respaldos regulares de los datos y servicios críticos:

Opciones Clave:

- **Amazon DynamoDB**:
  - Configura backups continuos con **DynamoDB Point-In-Time Recovery (PITR)**.
- **AWS Backup**:
  - Usa este servicio centralizado para automatizar y administrar los respaldos de:
    - Bases de datos RDS y DynamoDB.
    - Instancias EC2.
    - Sistemas de archivos (EFS).

4.5. Añadir Monitoreo y Alertas

Asegúrate de que los problemas se detecten rápidamente:

- Configura **Amazon CloudWatch Alarms** para eventos anómalos, como latencia alta o errores en el API.
- Usa **AWS CloudTrail** para auditar accesos y cambios en la infraestructura.
- Implementa **AWS Config** para verificar el cumplimiento de configuraciones predefinidas.