

Arquitectura planteada

**Componentes arquitectura**

AWS Lambda:

Este servicio permitirá ser un orquestador el cual será el encargado de procesar los datos extraídos de la fuente de datos

RDS :

Este servicio será el encargado de almacenar los registros necesarios para el proceso de insert y update de los documentos, este servicio se utiliza para la estrategia planteada de insert y update

Amazon Bedrock:

Este servicio permitirá la clasificación de los documentos y extracción de data necesaria mediante modelos de LLM.

Amazon DynamoDB:

Este servicio servira para el almacenamiento estructurado y rápido acceso a los artículos, de igual manera permite consultas rápidas junto a una escalabilidad automática.

Amazon ElasticSearch Service:

Este servicio permitirá realizar una búsqueda avanzada a los datos extraído y almacenados en la base de DynamoDB

Amazon S3:

Este servicio almacenará los artículos históricos para análisis y almacenamiento de larga duración. S3 será el lugar para los backups y recuperación de datos en caso de desastre.

AWS Backup:

Se utilizará para crear copias de seguridad automáticas de los datos almacenados en la RDS y DynamoDB, permitiendo una recuperación rápida en caso de pérdida de datos.

Amazon Quicksight:

Este servicio se utilizará como servicio de BI para la elaboración de gráficos los cuales permitan realizar el análisis pertinente a los datos de forma gráfica, generando conocimiento agregado.

Amazon CloudWatch:

Este servicio será utilizado para monitorear la arquitectura y las métricas en cada una de las capas de la arquitectura siendo un servicio transversal en todo el proceso, asegurando que todos los servicios están operando de manera óptima y proporcionando alertas si es necesario.

**Estrategia de Flujo de datos**

La estrategia planteada para el proceso del flujo de datos esta divida en 3 secciones las cuales permitirán hacen una ingesta en batch, una ingesta de artículos nuevos y una actualización de los artículos que han sido modificados.

**Ingesta en batch:**

En este proceso la lambda function se encargará de hacer la petición para extraer todos los datos del API de SpaceflightNews, una vez realizada la extracción de estos datos se implementara una estrategia para extraer el id máximo del documento insertado y de la última fecha de actualización del campo **update\_at esto** con el fin de obtener nuestras marcas de agua que permitirán ser usados en la capa de inserción de artículos nuevos y de artículos actualizados. Una vez obtenido estas marcas de agua se almacenarán en una RDS MySQL para su posterior uso, seguido a esto la lamda implementara el servicio de Amazon Bedrock con el fin de extraer palabras claves y la clasificación de los documentos según su tema.

Una vez extraídos y procesados los datos por el lambda function se procederá a almacenarlos en la base de DynamoDB para así poder tener una acceso rápido y realizar posteriores consultas, al igual se almacenaran los datos en un bucket de S3con el fin de conservar y generar data histórica de los datos que han sido extraídos y procesados, estos servicios de almacenamiento se configuraran con redundancia geográfica con el fin de salvaguardar la información a la vez que se implementara el servicio de Amazon backups para generar almacenar y generar backups de seguridad.

Se implementa una capa de búsqueda implementando el servicio de Amazon ElasticSearch el cual permitirá realizar consultas avanzadas de la data almacenada en DynamoDB, y se contará con una capa de visualización mediante el servicio de Amazon Quicksight para la construcción de gráficos para la visualización de información.

Contaremos con un servicio transversal de Amazon CloudWatch el cual estará encargado de monitorear y almacenar métricas necesarias para su posterior análisis. Permitiendo así un monitoreo de todos los servicios.

**Ingesta de artículos nuevos:**

En esta capa estrategia nuestra lambda function se encargará primero de leer el id\_maximo almacenado en la RDS seguido a esto extraerá los datos del api de SpaceflightNews donde estos registros sean mayores al id\_maximo almacenado. Una vez extraído estos nuevos registros se analizará cual es el nuevo id\_maximo y se actualizara en la RDS, seguido a esto la lamda implementara el servicio de Amazon Bedrock con el fin de extraer palabras claves y la clasificación de los documentos según su tema.

Una vez extraídos y procesados los datos por el lambda function se procederá a almacenarlos en la base de DynamoDB para así poder tener una acceso rápido y realizar posteriores consultas, al igual se almacenaran los datos en un bucket de S3con el fin de conservar y generar data histórica de los datos que han sido extraídos y procesados, estos servicios de almacenamiento se configuraran con redundancia geográfica con el fin de salvaguardar la información a la vez que se implementara el servicio de Amazon backups para generar almacenar y generar backups de seguridad.

Se implementa una capa de búsqueda implementando el servicio de Amazon ElasticSearch el cual permitirá realizar consultas avanzadas de la data almacenada en AmazonDB y se contará con una capa de visualización mediante el servicio de Amazon Quicksight para la construcción de gráficos para la visualización de información.

Contaremos con un servicio transversal de Amazon CloudWatch el cual estará encargado de monitorear y almacenar métricas necesarias para su posterior análisis. Permitiendo así un monitoreo de todos los servicios.

**Actualización de artículos:**

En esta capa estrategia nuestra lambda function se encargará primero de leer la fecha de actualización máxima de todos los artículos el cual esta almacenado en la RDS seguido a esto extraerá los datos del api de SpaceflightNews donde estos registros tengan una fecha de actualización mayor a la fecha de actualización máxima que fue almacenado. Una vez extraído estos nuevos registros se analizará cual es la fecha de actualización máxima y se actualizara en la RDS, seguido a esto la lamda implementara el servicio de Amazon Bedrock con el fin de extraer palabras claves y la clasificación de los documentos según su tema.

Una vez extraídos y procesados los datos por el lambda function se procederá a almacenarlos en la base de DynamoDB para así poder tener una acceso rápido y realizar posteriores consultas, al igual se almacenaran los datos en un bucket de S3con el fin de conservar y generar data histórica de los datos que han sido extraídos y procesados, estos servicios de almacenamiento se configuraran con redundancia geográfica con el fin de salvaguardar la información a la vez que se implementara el servicio de Amazon backups para generar almacenar y generar backups de seguridad.

Se implementa una capa de búsqueda implementando el servicio de Amazon ElasticSearch el cual permitirá realizar consultas avanzadas de la data almacenada en AmazonDB y se contará con una capa de visualización mediante el servicio de Amazon Quicksight para la construcción de gráficos para la visualización de información.

Contaremos con un servicio transversal de Amazon CloudWatch el cual estará encargado de monitorear y almacenar métricas necesarias para su posterior análisis. Permitiendo así un monitoreo de todos los servicios.

**Estrategia de almacenamiento y búsqueda**

En esta estrategia se contara con servicios de almacenamiento de DynamoDB el cual almacenara los artículos mas recientes o aquellos que tengan mayor acceso permitiendo así obtener datos rápidamente sin la necesidad de realizar búsquedas complejas de igual manera se contara con el servicio de S3 el cual permitirá almacenar aquella data histórica de los artículos que no se utiliza con frecuencia o que por políticas internas ya no debe utilizarse pero si se debe almacenar para procesos de auditoría.

En la estrategia de almacenamiento se implementará ElasticSearch con el fin de generar una búsqueda mas avanzada sobre los documentos almacenados.

**Estrategia de backups**

En esta estrategia se implementa el servicio de Amazon Backup con el fin de crear copias automáticas y programadas de los datos almacenados en RDS, DynamoDB y en su defecto el bucket de S3, al igual cada servicio se configura con redundancia geográfica para replicar la información entre regiones y tener la data segura.

**Estrategia de Monitoreo**

Se implementa el servicio de amazon cloudwatch con el fin de monitorear las métricas y flujos de trabajo de los diferentes servicios de la arquitectura planteada.

**Estimación de volumen de datos.**

**Frecuencia de ingesta:**

En esta frecuencia de ingesta validamos atravez del api SpaceflightNews filtrando atravez del campo de **published\_at** una serie de días observando cuantos documentos se publicaron cada día, seguido a esto se realiza un promedio de los documentos obteniendo asi el valor de 33 artículos cargados diariamente, pero a este valor aumentamos el 20% para así contar con un margen de error con lo cual obtenemos un promedio de 39.6 documentos lo cual redondeamos a 40 documentos cargados diariamente.

**Tamaño de cada articulo:**

Para determinar el tamaño de cada articulo se escoge una muestra aleatoria de documentos y se calcula el tamaño a través de namlabs online el cual permite saber el tamaño aproximado de cada documento. En el cual el promedio de tamaño de cada documento esta alrededor de 1.73 KB seguido a esto se aumenta el 20% para mantener un margen de error con el cual obtenemos un promedio de tamaño de cada documento de 2.3 KB.

**Volumen diario:**

40 documentos \* 2.3kb = 92 KB al día

92KB \* 30 Días = 2760 KB

En un mes seria 2760 KB aproximadamente, el cual es flujo moderado para ser almacenado en DynamoDB y en bucket de S3.