



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

ICADE

CIHS

Sprint 2

Grupo DASIMIOS

Ignacio Felices, Teresa Franco, Ulises Diez

Tecnologías de Procesamiento Big Data
3º Grado en Ingeniería Matemática e Inteligencia Artificial

Índice

INTRODUCCIÓN	3
METODOLOGÍA	4
RESULTADOS	6
CONCLUSIÓN	7

Introducción

La introducción debe proporcionar un contexto general del problema que se está resolviendo, su relevancia, y los objetivos específicos del sprint. Incluir en un breve párrafo:

1. **Contexto:** Explica brevemente el ámbito en el que se desarrolla el sprint.
2. **Descripción del problema y objetivos del sprint:** Introduce de manera clara y concisa el problema que estás resolviendo en el sprint, así como los objetivos principales.
3. **Justificación:** Explica por qué es relevante abordar este tipo de problema.

En el contexto del gobierno de datos y la gestión eficiente de grandes volúmenes de información, es fundamental contar con herramientas que permitan la organización, catalogación y procesamiento de datos almacenados en la nube. En este sentido, AWS Glue proporciona una solución automatizada y escalable para la gestión de metadatos y la preparación de datos en entornos distribuidos.

El problema central abordado en este sprint es la automatización del proceso de catalogación de datos históricos almacenados en Amazon S3 mediante AWS Glue Data Catalog y AWS Glue Crawler. El objetivo principal es desarrollar un script en Python que facilite la creación del catálogo de datos y la configuración del crawler, asegurando la correcta nomenclatura de bases de datos y tablas.

Este sprint es continuo con el desarrollo ya que permite establecer un marco de gobernanza de datos sobre la información almacenada en S3, asegurando su accesibilidad, integridad y estructuración para futuros procesos de análisis. Al automatizar la gestión de metadatos, se optimizan los flujos de trabajo, se reduce la intervención manual y se mejora la eficiencia operativa de las consultoras encargadas del tratamiento de estos datos.

Metodología

Esta sección describe en detalle cómo se ha implementado la solución, abordando las decisiones técnicas, las tecnologías utilizadas y el proceso de desarrollo. Incluye:

1. **Descripción del entorno de desarrollo:** Herramientas utilizadas: p.ej Python, librerías, etc., así como el entorno de ejecución.
2. **Diseño de la solución:** Describe el diseño y funcionamiento de la arquitectura del sistema. Puede incluirse texto, diagramas, u otros recursos visuales que ayuden a comunicar la solución de manera efectiva.
3. **Pruebas realizadas:** Explica cómo se realizaron las pruebas para verificar el correcto funcionamiento del sistema.

Extracción de datos de la plataforma de inversión

Primero ha de mencionarse que la metodología seguida en el Sprint 1 de la estructura de nuestro Bucket se ha tenido que cambiar. En el Sprint 1, pensamos que lo óptimo fuera hacer para cada criptomoneda una carpeta, y dentro de cada criptomoneda, hacer una carpeta distinta para cada año y así en cada carpeta tener una tabla distinta. En cambio, al comenzar a crear los crawlers, nos dimos cuenta de que no era la forma mas eficiente de estructurar el bucket. Es por ello, que hemos hecho de nuevo la estructura del bucket, donde hemos creado una sola carpeta para cada criptomoneda, y dentro de ella las cuatro tablas de cada año.

Objetos (10)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [Inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
/	Carpeta	-	-	-
/ADAE/	Carpeta	-	-	-
/ADA/	Carpeta	-	-	-
/BTC/	Carpeta	-	-	-
/DOGE/	Carpeta	-	-	-
/DOT/	Carpeta	-	-	-
/ETH/	Carpeta	-	-	-
/SHIB/	Carpeta	-	-	-
/SOL/	Carpeta	-	-	-
/XLM/	Carpeta	-	-	-
/XRP/	Carpeta	-	-	-

BTC/

Objetos (4)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [Inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
BTC_2021.csv	csv	11 Feb 2025 8:42:25 PM CET	23.1 KB	Estándar
BTC_2022.csv	csv	11 Feb 2025 8:42:25 PM CET	21.6 KB	Estándar
BTC_2023.csv	csv	11 Feb 2025 8:42:25 PM CET	21.7 KB	Estándar
BTC_2024.csv	csv	11 Feb 2025 8:42:25 PM CET	21.7 KB	Estándar

Para crear el nuevo bucket tuvimos que hacer los ajustes necesarios en el S3DataUpload.py para volver a crear y subir los datos de forma automatizada.

Implementación Crawlers

Para implementar en el crawler, vamos a trabajar en el ambiente del AWS Glue.

Para ello hemos creado un script llamado CrawlerScript.py, donde se crea un crawler para cada carpeta de cada criptomoneda. Este código interactúa con AWS Glue y automatiza la creación de una base de datos, crawlers y su ejecución para procesar datos de criptomonedas almacenados en un bucket de S3.

Primero configuramos los parámetros clave como la región de AWS, el nombre de la base de datos en Glue, el ARN del rol de IAM y el bucket de S3 donde están almacenados los datos históricos. Después con la función `create_database()` creamos una base de datos en AWS Glue con el nombre:

`trade_data_imat3a08`

Como nos ha indicado la nomenclatura de la práctica (`trade_data_<grupo>`). Si la base de datos ya existe, el código simplemente lo notifica sin intentar recrearla, evitando errores innecesarios.

La función `create_crawlers()` obtiene la lista de crawlers existentes en AWS Glue para evitar duplicaciones. Luego, para cada criptomoneda en la lista, define el nombre del crawler correspondiente (`crawler_AAVE`, `crawler_BTC`) y apunta a la ubicación de los datos en S3:

`s3://crypto-data-4e822955/AAVE/, s3://crypto-data-4e822955/BTC/`

Si el crawler ya existe, el código lo omite. Si no existe, se crea un nuevo crawler en AWS Glue con los siguientes parámetros:

- Rol IAM necesario para que AWS Glue acceda a los datos
- Base de datos destino (`trade_data_imat3a08`)
- Un prefijo de tabla (`trade_data_`)
- Una programación diaria para correr a las 12 PM
- Política de cambios de esquema que permite actualizar estructuras de datos automáticamente sin intervención manual.

En conclusión, este código permite automatizar la creación de una base de datos en AWS Glue, generar crawlers específicos para cada criptomoneda y ejecutar los crawlers para extraer y catalogar los datos almacenados en S3. De esta manera, facilita la gobernanza y estructuración de los datos para su posterior análisis en entornos de Big Data.

```
[ec2-user@ip-172-31-2-56 ~]$ python3 CrawlerScript.py
La base de datos 'trade_data_imat3a08' ya existe.
Crawler 'crawler_AAVE' creado exitosamente para AAVE.
Crawler 'crawler_ADA' creado exitosamente para ADA.
Crawler 'crawler_BTC' creado exitosamente para BTC.
Crawler 'crawler_DOGE' creado exitosamente para DOGE.
Crawler 'crawler_DOT' creado exitosamente para DOT.
Crawler 'crawler_ETH' creado exitosamente para ETH.
Crawler 'crawler_SHIB' creado exitosamente para SHIB.
Crawler 'crawler_SOL' creado exitosamente para SOL.
Crawler 'crawler_XLM' creado exitosamente para XLM.
Crawler 'crawler_XRP' creado exitosamente para XRP.
Crawler 'crawler_AAVE' iniciado para AAVE.
Crawler 'crawler_ADA' iniciado para ADA.
Crawler 'crawler_BTC' iniciado para BTC.
Crawler 'crawler_DOGE' iniciado para DOGE.
Crawler 'crawler_DOT' iniciado para DOT.
Crawler 'crawler_ETH' iniciado para ETH.
Crawler 'crawler_SHIB' iniciado para SHIB.
Crawler 'crawler_SOL' iniciado para SOL.
Crawler 'crawler_XLM' iniciado para XLM.
Crawler 'crawler_XRP' iniciado para XRP.
```

Resultados

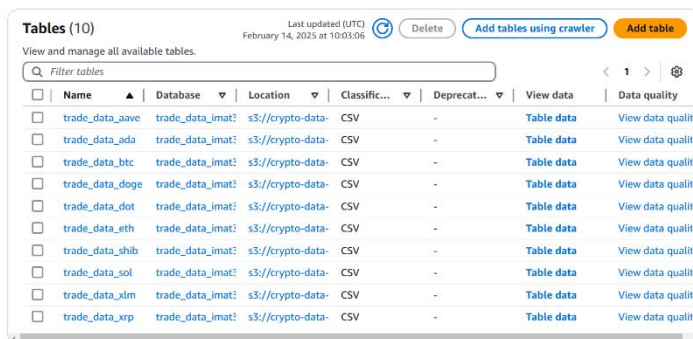
En esta sección se presentan los resultados obtenidos durante la ejecución del sprint, demostrando cómo la solución implementada resolvió el problema planteado. Debe incluir:

1. **Descripción de los resultados:** Describe los resultados obtenidos a partir de la implementación.
2. **Pantallazos de la ejecución (IMPORTANTE):** Incluye capturas de pantalla que muestren la ejecución del sistema en una terminal o entorno de pruebas.
3. **Discusión de los resultados:** Comparar los resultados obtenidos con los esperados. Este apartado pretende responder a preguntas como: ¿Se comporta el sistema de la manera prevista? ¿Qué factores han afectado a cada resultado? ¿Ha habido algún comportamiento inesperado del sistema? ¿Por qué?

Tablas Creadas en AWS Glue

Una vez ejecutamos el CrawlerScript, se crean diez tablas en AWS Glue, una para cada criptomoneda. En cada tabla se encuentran los datos de los últimos cuatro años, almacenados en un único conjunto. Como se mencionó anteriormente, esta decisión se tomó para facilitar las consultas sobre una criptomoneda en particular, ya que al tener toda la información consolidada en una sola tabla, se evita la necesidad de realizar uniones entre varias tablas.

En la siguiente captura se muestran las diez tablas creadas en nuestro AWS Glue Data Catalog:



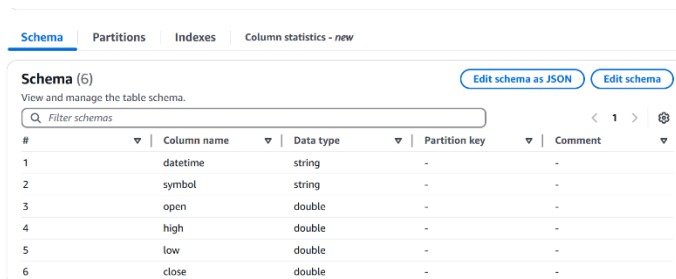
Tables (10)

List updated (UTC) February 14, 2025 at 10:05:06

View and manage all available tables.

Name	Database	Location	Classific...	Deprecat...	View data	Data quality
trade_data_aave	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_ada	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_btc	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_doge	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_dot	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_eth	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_shib	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_sol	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_xlm	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual
trade_data_xrp	trade_data_imat	s3://crypto-data-	CSV	-	Table data	View data qual

Dado que AWS Glue almacena metadatos, proporciona información relevante sobre las características de los datos almacenados. Para cada tabla, podemos observar la siguiente estructura:



Schema (6)

View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	datetime	string	-	-
2	symbol	string	-	-
3	open	double	-	-
4	high	double	-	-
5	low	double	-	-
6	close	double	-	-

Vemos que nos informa de:

Datetime – string
 Symbol – string
 Open – double
 High – double
 Low – double
 Close – double

Estos son los campos extraídos de los datos almacenados en nuestro bucket de S3.

Ejecución de los Glue Crawlers

Al ejecutar los Glue Crawlers, que descubren y organizan automáticamente los datos en S3, verificamos que los diez crawlers se han ejecutado correctamente.

Crawlers (10) Info Last updated (UTC) February 14, 2025 at 10:14:46 Action Run Create crawler

View and manage all available crawlers.

Filter crawlers

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run ...	Log	Table cha...
<input type="checkbox"/>	crawler_AAVE	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_ADA	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_BTC	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_DOGE	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_DOT	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_ETH	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_SHIB	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_SOL	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_XLM	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-
<input type="checkbox"/>	crawler_XRP	Ready	At 12:00 PM	Succeeded	February 13,...	View log	-

Para cada uno de los crawlers, podemos obtener su información relevante. Cada uno de los crawlers son llamados: crawler_XXX donde XXX es el símbolo de la criptomoneda. Veamos para el crawler que se encargo de escanear los datos de la table AAVE.

crawler_AAVE Last updated (UTC) February 14, 2025 at 10:18:02 Run crawler Edit Delete

Crawler properties Name crawler_AAVE Description Crawler para datos preprocesados de AAVE Maximum table threshold - ▶ Advanced settings	IAM role AWSGlueServiceRole-workshop Security configuration -	Database trade_data_imat3a08 Lake Formation configuration -	State READY Table prefix trade_data_
---	--	--	---

[Crawler runs](#) [Schedule](#) [Data sources](#) [Classifiers](#) [Tags](#)

Crawler runs (3) Stop run View CloudWatch logs View run details

The list of crawler runs for this crawler.

Filter data

<input type="radio"/>	Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
<input type="radio"/>	February 13, 2025 at 12:00:2	February 13, 2025 at 12:01:3	01 min 15 s	Completed	0.046	-
<input type="radio"/>	February 12, 2025 at 12:00:4	February 12, 2025 at 12:02:2	01 min 38 s	Completed	0.038	-
<input type="radio"/>	February 11, 2025 at 20:54:5	February 11, 2025 at 20:55:4	53 s	Completed	0.054	1 table change, 0 partition changes

Podemos obtener información relevante del crawler, como:

- La fecha y hora de ejecución (Crawler runs).
- La base de datos en la que se almacenó la tabla (trade_data_imat3a08).
- La configuración de ejecución automática, programada correctamente para ejecutarse diariamente a las 12 PM.

[Crawler runs](#) [Schedule](#) [Data sources](#) [Classifiers](#) [Tags](#)

Schedule <small>Info</small> Status Scheduled	Frequency Daily	Time At 12:00 PM
--	--------------------	---------------------

Consulta de Datos en AWS Athena

Por último, cuando queremos visualizar los datos de una tabla, AWS Glue nos redirige a AWS Athena, ya que Glue solo se encarga de gestionar los metadatos y ejecutar los crawlers.

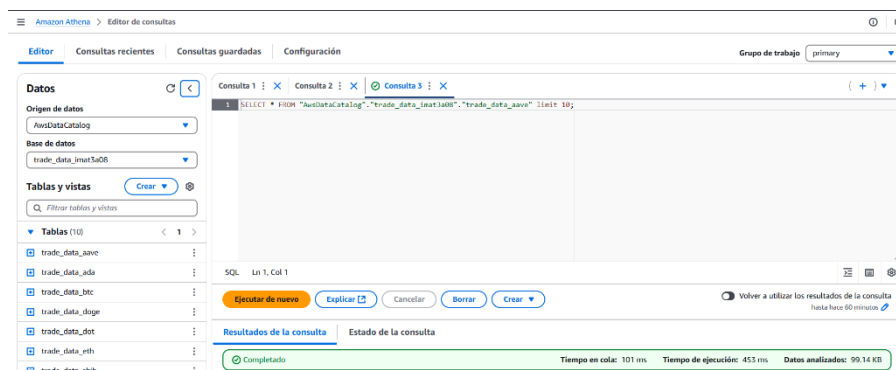
En AWS Athena, podemos ejecutar consultas sobre las tablas creadas. Para verificar que los datos han sido correctamente importados desde el crawler, ejecutamos la siguiente consulta SQL:

```
SELECT * FROM "AwsDataCatalog"."trade_data_imat3a08"."trade_data_aave" limit 10
```

Esta consulta selecciona los primeros 10 registros de la tabla "trade_data_aave", dentro de la base de datos "trade_data_imat3a08", registrada en el AWS Glue Data Catalog.

Este proceso se ha repetido para todas las tablas, confirmando que las diez tablas están correctamente creadas y disponibles para consulta en AWS Athena, ya que los diez crawlers han ejecutado su tarea de manera exitosa.

A continuación, se muestra la consulta ejecutada:



Amazon Athena > Editor de consultas

Editor Consultas recientes Consultas guardadas Configuración Grupo de trabajo: primary

Datos

Origen de datos: AwsDataCatalog

Base de datos: trade_data_imat3a08

Tablas y vistas: Crear

Tablas (10):

- trade_data_aave
- trade_data_ada
- trade_data_btc
- trade_data_doge
- trade_data_dot
- trade_data_eht
- trade_data_ehtb

Consulta 1: X Consulta 2: X Consulta 3: X

1 SELECT * FROM "AwsDataCatalog"."trade_data_imat3a08"."trade_data_aave" limit 10;

SQL Lin 1, Col 1

Ejecutar de nuevo Explicar Cancelar Borrar Crear

Volver a utilizar los resultados de la consulta hasta hace 90 minutos

Resultados de la consulta Estado de la consulta

Completado Tiempo en cola: 101 ms Tiempo de ejecución: 453 ms Datos analizados: 99.14 KB

Resultado:

Resultados (10) Copiar Descargar resultados en formato CSV

Filas de búsqueda

#	datetime	symbol	open	high	low	close
1	2024-02-13	CRYPTO-AAVEUSD	90.85166869	91.38534107	88.09464916	89.6092013
2	2024-02-14	CRYPTO-AAVEUSD	89.60727266	92.74680688	89.1004728	91.05497444
3	2024-02-15	CRYPTO-AAVEUSD	91.05211587	93.52916774	90.10933429	92.33657693
4	2024-02-16	CRYPTO-AAVEUSD	92.33657693	95.04357596	90.52477884	94.61578851
5	2024-02-17	CRYPTO-AAVEUSD	94.61578851	94.86397651	91.14362762	93.97648545
6	2024-02-18	CRYPTO-AAVEUSD	93.9564772	95.4768835	92.64398758	94.66672048
7	2024-02-19	CRYPTO-AAVEUSD	94.66797086	98.02440247	94.28744356	96.20865304
8	2024-02-20	CRYPTO-AAVEUSD	96.21008195	97.12997028	90.09670731	93.86314535
9	2024-02-21	CRYPTO-AAVEUSD	93.8606448	94.23164256	88.1166892	91.73223502
10	2024-02-22	CRYPTO-AAVEUSD	91.73223502	93.88286046	90.08527923	91.96496757

Podemos concluir que la tarea de creación de tablas mediante AWS Glue Crawlers ha sido exitosa, ya que en AWS Athena las consultas se ejecutan correctamente, validando que los datos fueron correctamente subidos al Data Catalog.

Conclusión

La conclusión debe resumir los principales hallazgos y aprendizajes obtenidos durante el sprint, así como destacar la relevancia de la solución implementada. Incluye:

1. **Resumen del proceso:** Recapitula brevemente el proceso seguido desde la identificación del problema hasta la obtención de los resultados.
2. **Principales logros:** Destaca los logros más importantes, como el correcto funcionamiento del sistema, en base a los resultados obtenidos.

Durante este sprint, hemos desarrollado un sistema eficiente para la subida de datos históricos de criptomonedas mediante AWS Glue Crawlers.

Resumen del proceso

Podemos concluir que la tarea de creación y gestión de tablas en AWS Glue mediante Glue Crawlers ha sido exitosa en el contexto de nuestro sprint. Se han configurado y ejecutado diez crawlers, cada uno encargado de procesar los datos históricos de una criptomoneda almacenados en Amazon S3. Estos crawlers han generado diez tablas dentro del AWS Glue Data Catalog, siguiendo la nomenclatura establecida (trade_data_<tabla>). Cada tabla contiene los datos de los últimos cuatro años en un formato consolidado, lo que facilita las consultas y el análisis de la información.

Principales logros

Se ha validado que los crawlers han sido programados correctamente para su ejecución automática diaria a las 12 PM, asegurando la actualización continua del catálogo de datos. Posteriormente, se verificó la correcta creación y carga de las tablas en AWS Athena, donde se realizaron consultas SQL para comprobar la integridad de los datos. Por lo tanto, se confirma que el proceso de descubrimiento, catalogación y consulta de datos ha sido implementado con éxito, permitiendo que los datos almacenados en S3 sean accesibles y consultables de manera eficiente a través de AWS Glue y Athena.