AWS Cloud Data Pipeline

Serverless Analytics Pipeline Using Amazon Web Services

Esther Quirós
Developer

Nieves Pérez
Data Analyst

Elena Pavón
Data Analyst

Mayka Durán
Data Analyst

Claudia Cervantes
Data Analyst · Scrum Master

Team Project • AWS re/Start Program

¿Por qué este proyecto?

Como equipo principalmente de Data Analysts, queríamos llevar nuestros conocimientos de Data Analytics a un entorno cloud real utilizando servicios de AWS.

El objetivo fue construir un pipeline automatizado capaz de:

- ingestión de datos
- transformación ETL
- automatización
- almacenamiento en la nube
- consultas SQL analytics

El dataset fue el caso práctico — el verdadero reto fue construir el pipeline.

Arquitectura General

Pipeline Analytics End-to-End en AWS

AWS User

S3 RAW
(CSV uploads)

S3 Event

AWS Lambda

AWS Glue
ETL

S3 CLEAN
(Parquet datasets)

Parquet data

Amazon Athena

S3 CLEAN
(Athena query outputs)

Metadata/
Schema

AWS Glue
Crawler

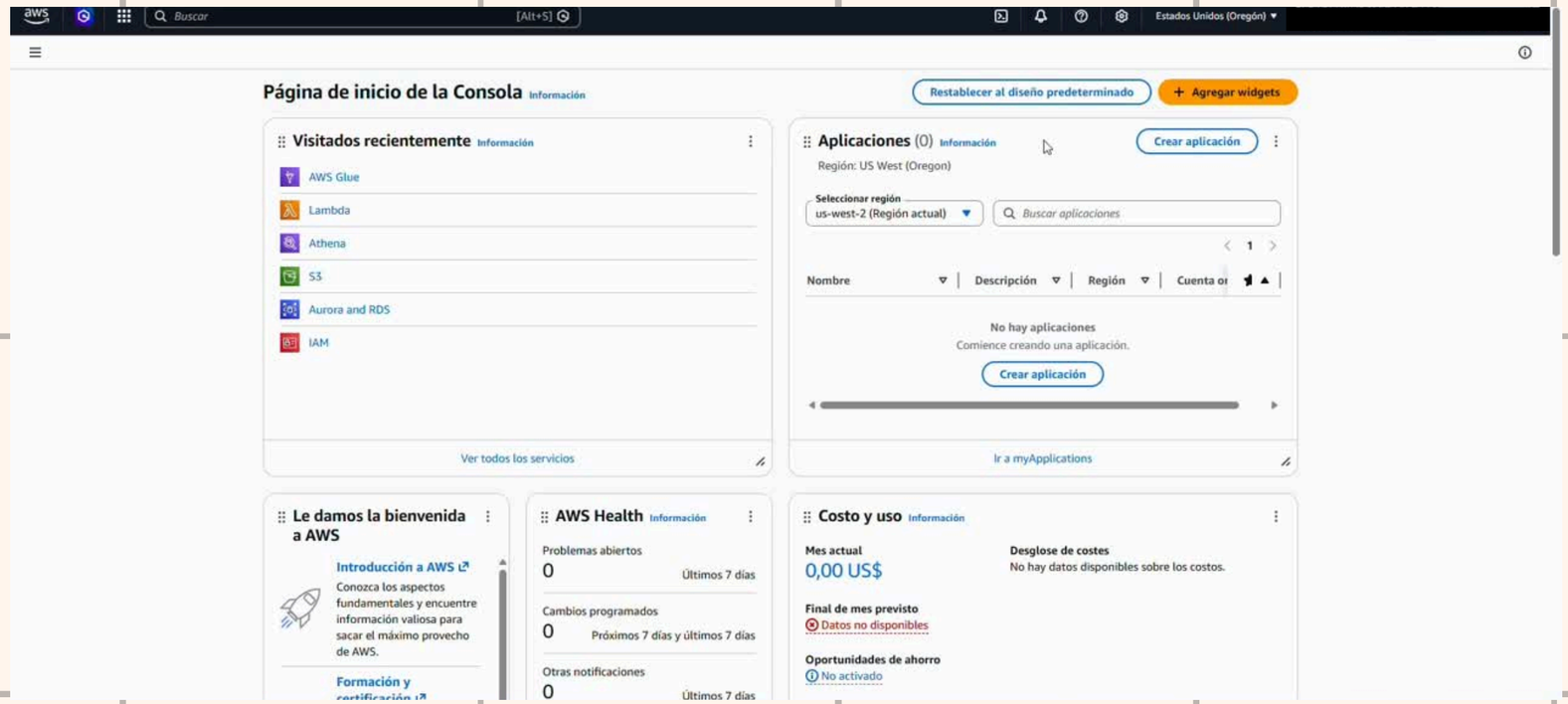
AWS Glue
Data Catalog

Amazon S3 como Data Lake

- Amazon S3 = almacenamiento escalable y seguro
- Actúa como Data Lake
- Guarda datos RAW y CLEAN
- Integración con:

AWS Glue
Amazon Athena
AWS Lambda

Organización del almacenamiento



Bucket 1 — RAW

- Datos originales (CSV)
- Sin modificar
- Fuente del pipeline

Bucket 2 — CLEAN

- Datos transformados
- Formato Parquet
- Listos para análisis

¿Por qué lo separamos?

-No perder datos originales
-Evitar errores
-Pipeline más claro

Seguridad y configuración

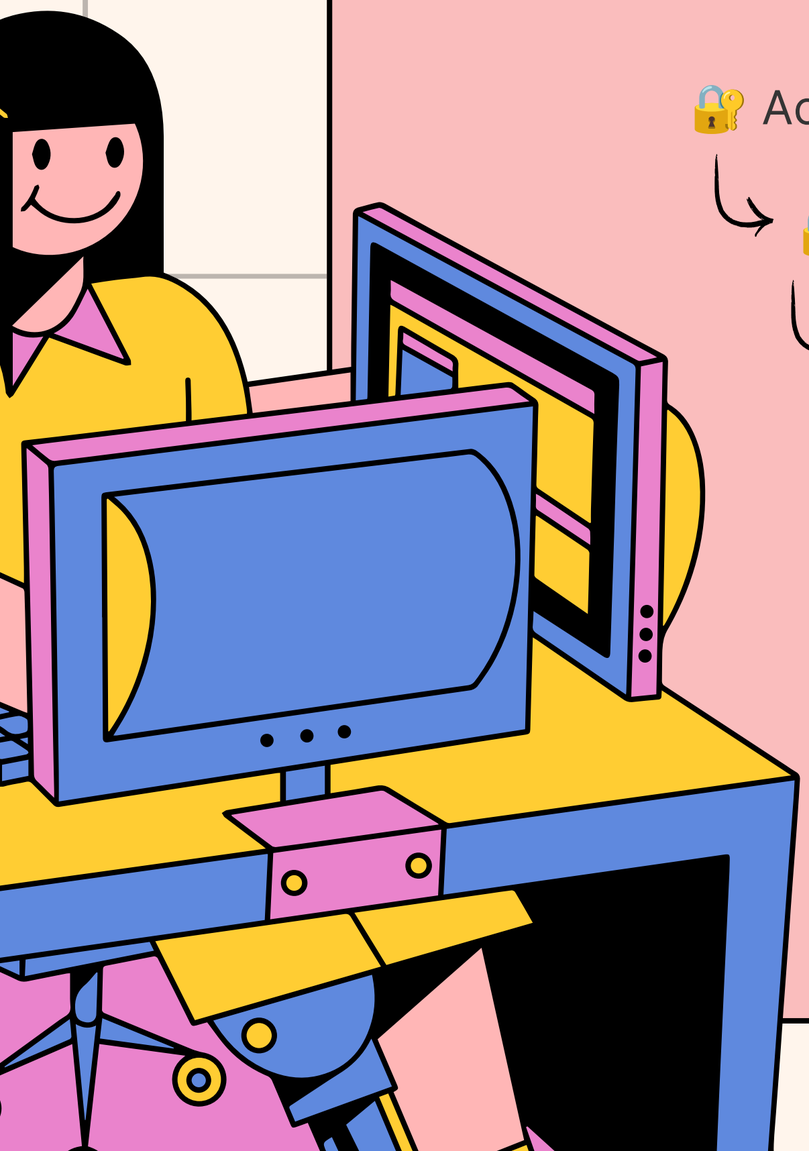
🔐 Acceso mediante roles (IAM)

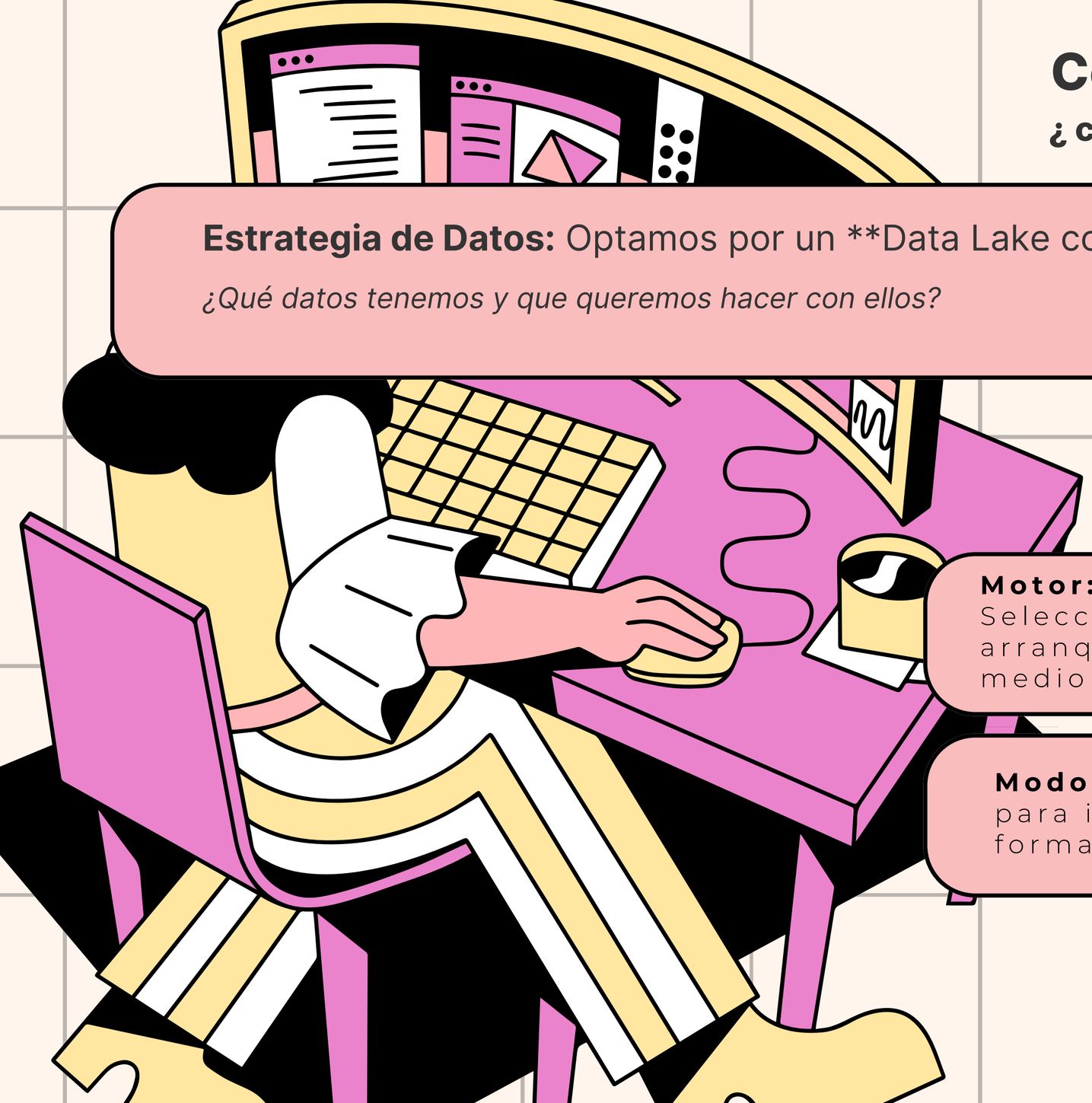
↳ 🔒 Acceso público bloqueado

↳ ♻️ Versionado: desactivado (laboratorio)

Objetivo

Seguridad + simplicidad + control ✓





Configuraciónde AWS GLUE

¿ cómo preparamos el "Robot" de limpieza?

Estrategia de Datos: Optamos por un **Data Lake con AWS Glue** en lugar de Aurora SQL.

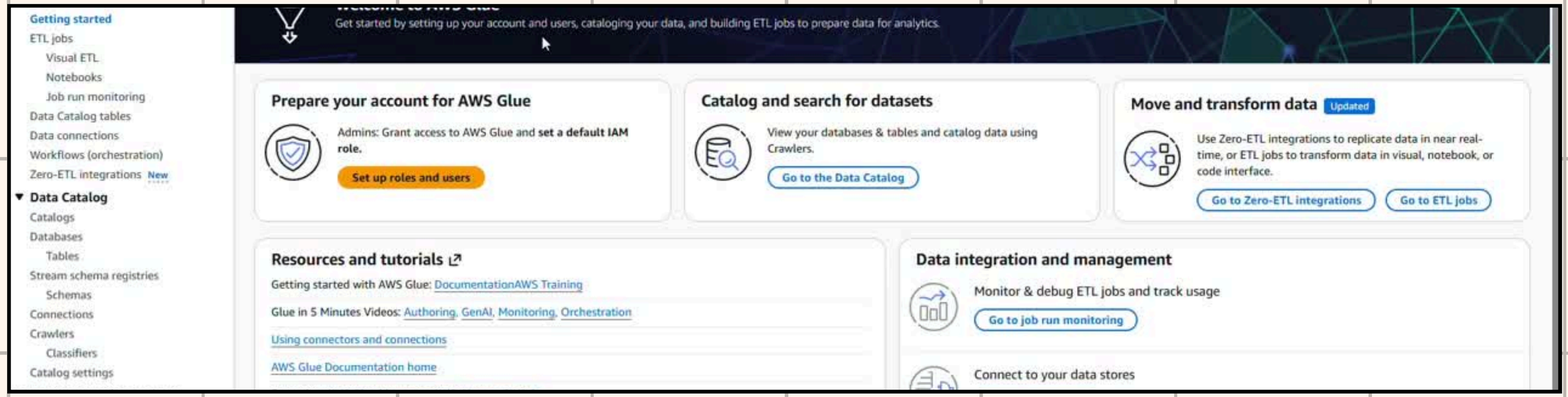
¿Qué datos tenemos y que queremos hacer con ellos?

CONFIGURACIÓN TÉCNICA

SCRIPT EDITOR

Motor:
Seleccionamos *Python Shell* por su rapidez de arranque y eficiencia en datasets de tamaño medio (Kaggle).

Modo "Start Fresh": Usamos un editor limpio para inyectar nuestro código personalizado de forma directa



1. **Extracción (s3--> Boto3/Pandas)**

1. **Proceso de Transformación:**

- **Estandarización de nombres**
- **Eliminación de duplicados**
- **Manejo de valores nulos**
- **Homogeneización de Ratings**
- **Enriquecimiento de fechas**
- **Parsing de Duración**

3. **Carga (S3 en formato Parquet):**

- **queries más rápidas**
- **menor costo Athena**
- **mejor compresión**
- **preserva schema**
- **soporta arrays/listas**
- **mejor performance**

El Script ETL y la Magia de Parquet

De CSV "sucio" a un formato de alto rendimiento.

DEL ARCHIVO A LA CONSULTA SQL (CRAWLER & CATALOG)

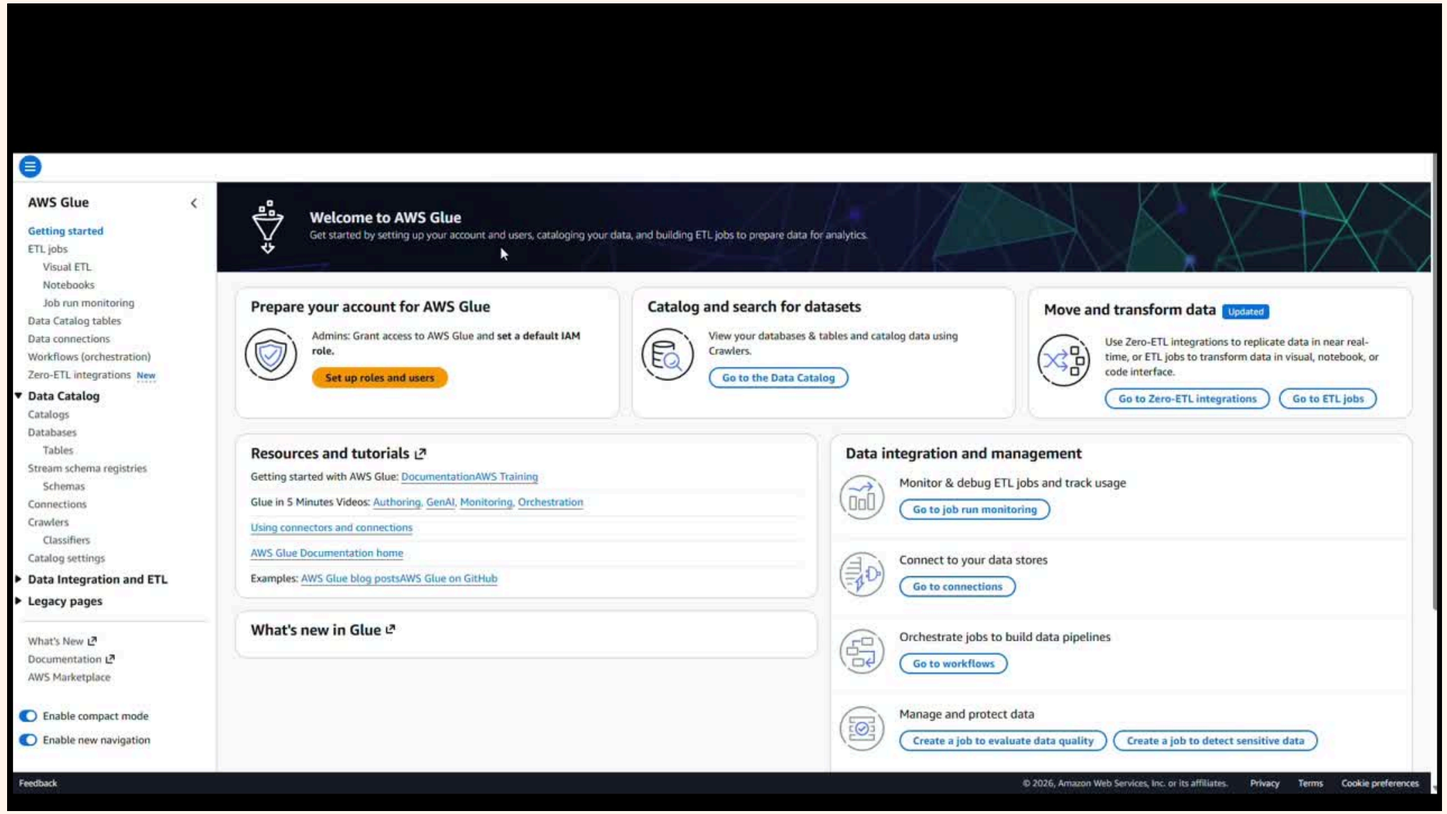
SEGURIDAD Y PERMISOS:

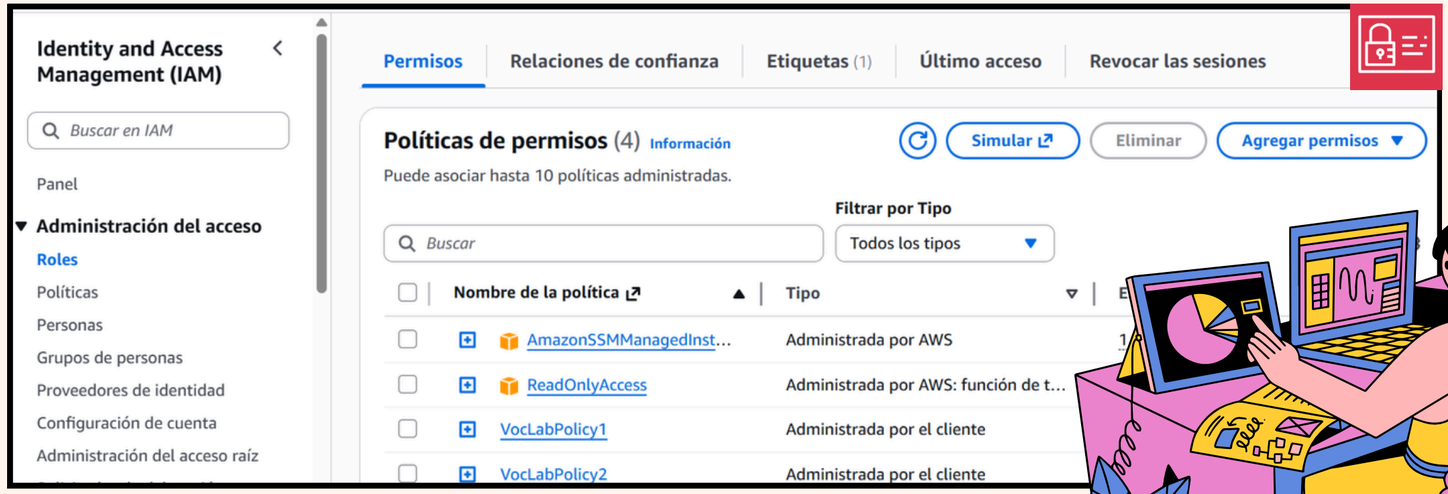
- USO DEL ***LABROLE*** (ENTORNO AWS ACADEMY).
- Asociación de políticas clave: `*S3FullAccess*` y `*AWSGlueServiceRole*` para permitir la comunicación entre servicios.

* Asociación de políticas clave: `S3FullAccess` y `AWSGlueServiceRole` para permitir la comunicación entre servicios.

PROCESO DE CATALOGACIÓN:

1. **Crawler (El Explorador):** Escanea el *bucket clean*, identifica el esquema de los archivos *Parquet* y detecta cambios.
2. **Data Catalog (La Biblioteca):** Repositorio central de metadatos que guarda el "mapa" de nuestra información.
3. **Database:** Agrupación lógica donde residen las tablas virtuales.





¿Qué es Athena?

Servicio de consultas interactivo que permite analizar grandes volúmenes de datos almacenados en Amazon S3 con SQL estándar

- ↳ serverless
- ↳ escala automáticamente
- ↳ funciona directamente sobre Parquet
- ↳ evita cargar datos en MySQL

¿Por qué usamos Athena?

Porque nuestro proyecto es un **analytical workload**

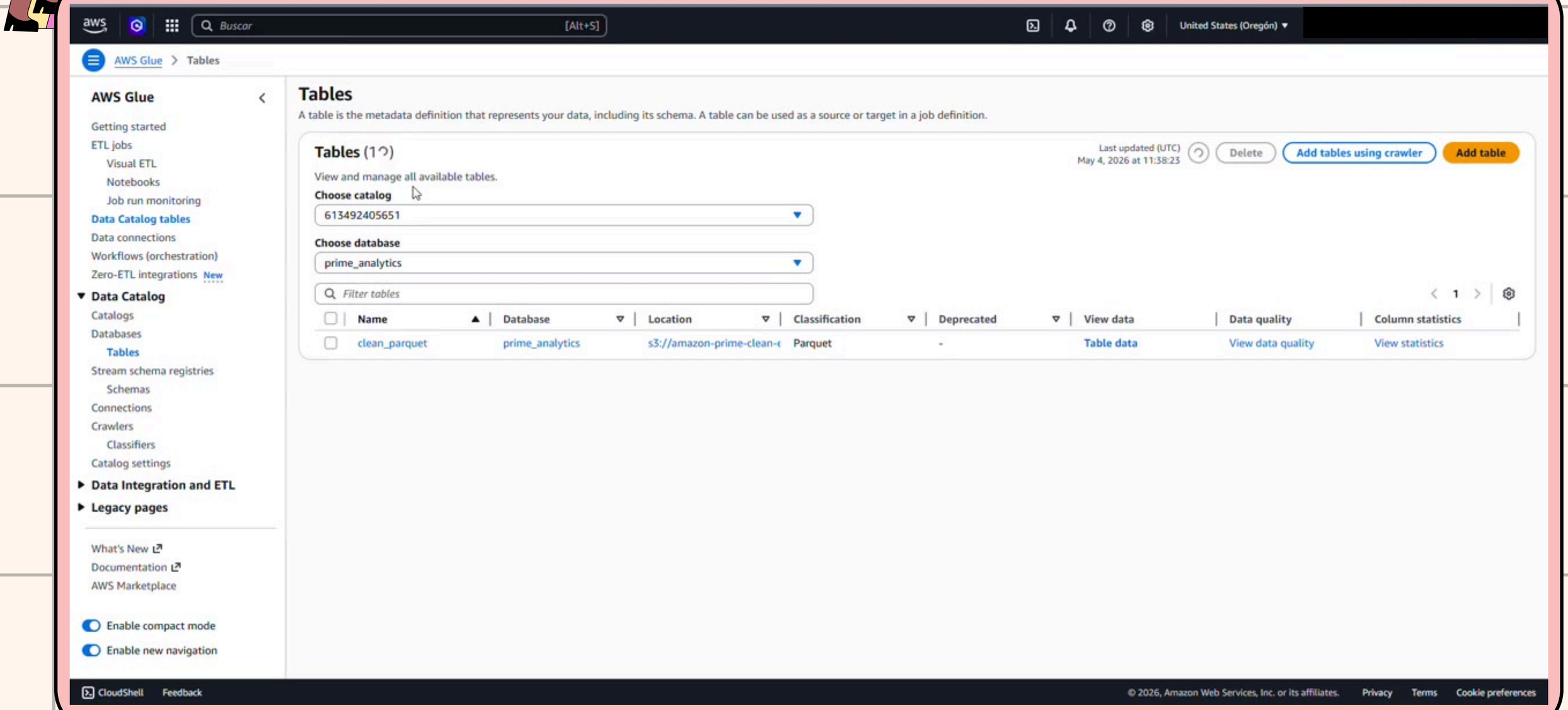
✅ Queríamos

↳ Analizar datos
↳ Agrupar información
↳ Generar insights

❌ NO:

↳ Manejar transacciones
↳ Actualizar usuarios
↳ Crear aplicaciones

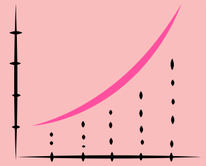
Demo de Athena



¿Porqué S3 Triggers y AWS Lambda?

Eficiencia - Latencia Cero

Reacción instantánea ante la llegada de datos. Sin esperas manuales ni procesos programados innecesarios.



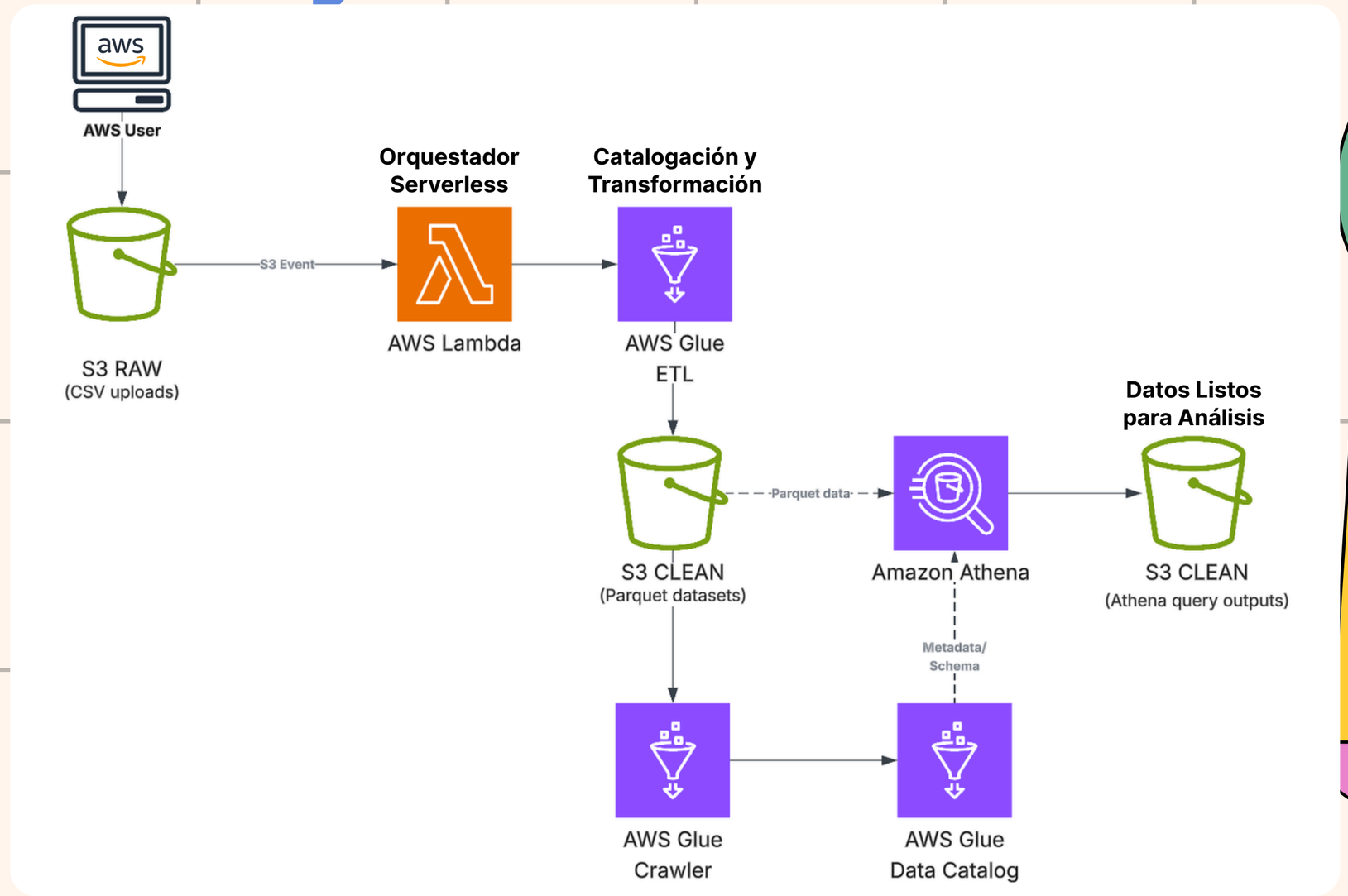
Escalabilidad

AWS Lambda escala automáticamente para procesar múltiples subidas simultáneas sin cuellos de botella.

Coste $0

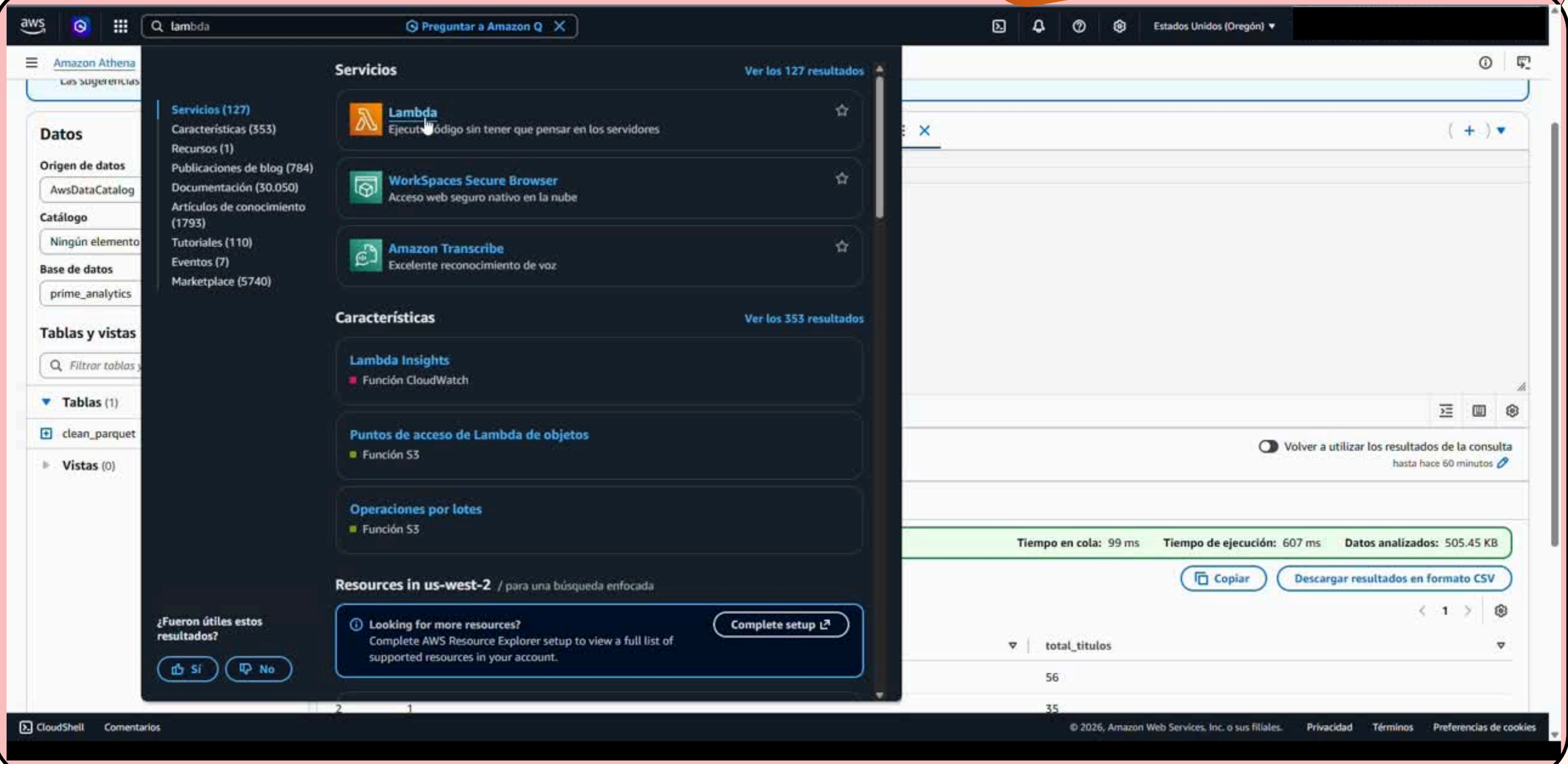
Solo pagamos por los milisegundos de ejecución. Eficiencia total en el uso del presupuesto cloud.

Lambda Automation
El Flujo del Dato



Demo Lambda


aws


aws
lambda
Preguntar a Amazon Q
Estados Unidos (Oregón)
Amazon Athena
Datos
Origen de datos
AwsDataCatalog
Catálogo
Ningún elemento
Base de datos
prime_analytics
Tablas y vistas
Filtrar tablas
Tablas (1)
clean_parquet
Vistas (0)
Servicios
Ver los 127 resultados
Servicios (127)
Características (353)
Recursos (1)
Publicaciones de blog (784)
Documentación (30.050)
Artículos de conocimiento (1793)
Tutoriales (110)
Eventos (7)
Marketplace (5740)
Lambda
Ejecutar código sin tener que pensar en los servidores
WorkSpaces Secure Browser
Acceso web seguro nativo en la nube
Amazon Transcribe
Excelente reconocimiento de voz
Características
Ver los 353 resultados
Lambda Insights
Función CloudWatch
Puntos de acceso de Lambda de objetos
Función S3
Operaciones por lotes
Función S3
Resources in us-west-2 / para una búsqueda enfocada
Looking for more resources?
Complete AWS Resource Explorer setup to view a full list of supported resources in your account.
Complete setup
¿Fueron útiles estos resultados?
Sí
No
Volver a utilizar los resultados de la consulta
hasta hace 60 minutos
Tiempo en cola: 99 ms
Tiempo de ejecución: 607 ms
Datos analizados: 505.45 KB
Copiar
Descargar resultados en formato CSV
total_titulos
56
35
CloudShell
Comentarios
© 2026, Amazon Web Services, Inc. o sus filiales.
Privacidad
Términos
Preferencias de cookies

Comprobación

aws | Buscar [Alt+S] | Estados Unidos (Oregón)

Lambda > Funciones > trigger-glue-prime-etl

trigger-glue-prime-etl

Limitación | Copiar ARN | Acciones

El desencadenador amazon-prime-raw-equipo-mdm se agregó correctamente a la función trigger-glue-prime-etl.

Información general de la función Información

Exportar a Infrastructure Composer | Descargar

Diagrama | Plantilla

trigger-glue-prime-etl

Capas (0)

S3

+ Agregar desencadenador

+ Agregar destino

ARN de la función
arn:aws:lambda:us-west-2:613492405651:function:trigger-glue-prime-etl

Descripción
-

Última modificación
hace 35 segundos

Código | Probar | Monitorear | Configuración | Alias | Versiones

Configuración

- Configuración general
- Desencadenadores
- Permisos
- Destinos
- URL de la función

Desencadenadores (1) Información

Corregir errores | Editar | Eliminar | Agregar desencadenador

Búsqueda de desencadenantes

1

Desencadenador
S3: amazon-prime-raw-equipo-mdm arn:aws:s3:::amazon-prime-raw-equipo-mdm ▶ Detalles

CloudShell | Comentarios | © 2026, Amazon Web Services, Inc. o sus filiales. | Privacidad | Términos | Preferencias de cookies

Arquitectura y decisiones técnicas

Durante el diseño inicial consideramos utilizar servicios como Amazon RDS y EC2.

Sin embargo, al analizar el tipo de *workload* y los objetivos analytics del proyecto, evolucionamos hacia una arquitectura serverless basada en S3, Glue y Athena.

Decisiones clave:

- Data Lake en S3 con consultas SQL serverless mediante Athena
- transformación ETL serverless con AWS Glue
- automatización orientada a eventos mediante Lambda
- reducción de infraestructura para simplificar costes y operación

La arquitectura final priorizó escalabilidad, simplicidad operativa y coste eficiente.

Conclusión

Construimos un pipeline analytics moderno, automatizado y serverless utilizando servicios reales de AWS.

El pipeline permitió:

- ingestión y transformación ETL automatizada
- almacenamiento optimizado en formato Parquet
- consultas SQL serverless sobre S3 con Athena
- arquitectura cloud-native escalable
- costes operativos mínimos

Next Steps:

- alertas automáticas con SNS + CloudWatch
- dashboards BI conectados a Athena

Coste estimado: ~$0.77/mes, datasets pequeños y uso académico

Muchas Gracias!

Pueden consultar el video del despliegue: