

# AWS Student Community Day

Perú

13/12/2025



AWS Cloud Clubs



*National University of Engineering*

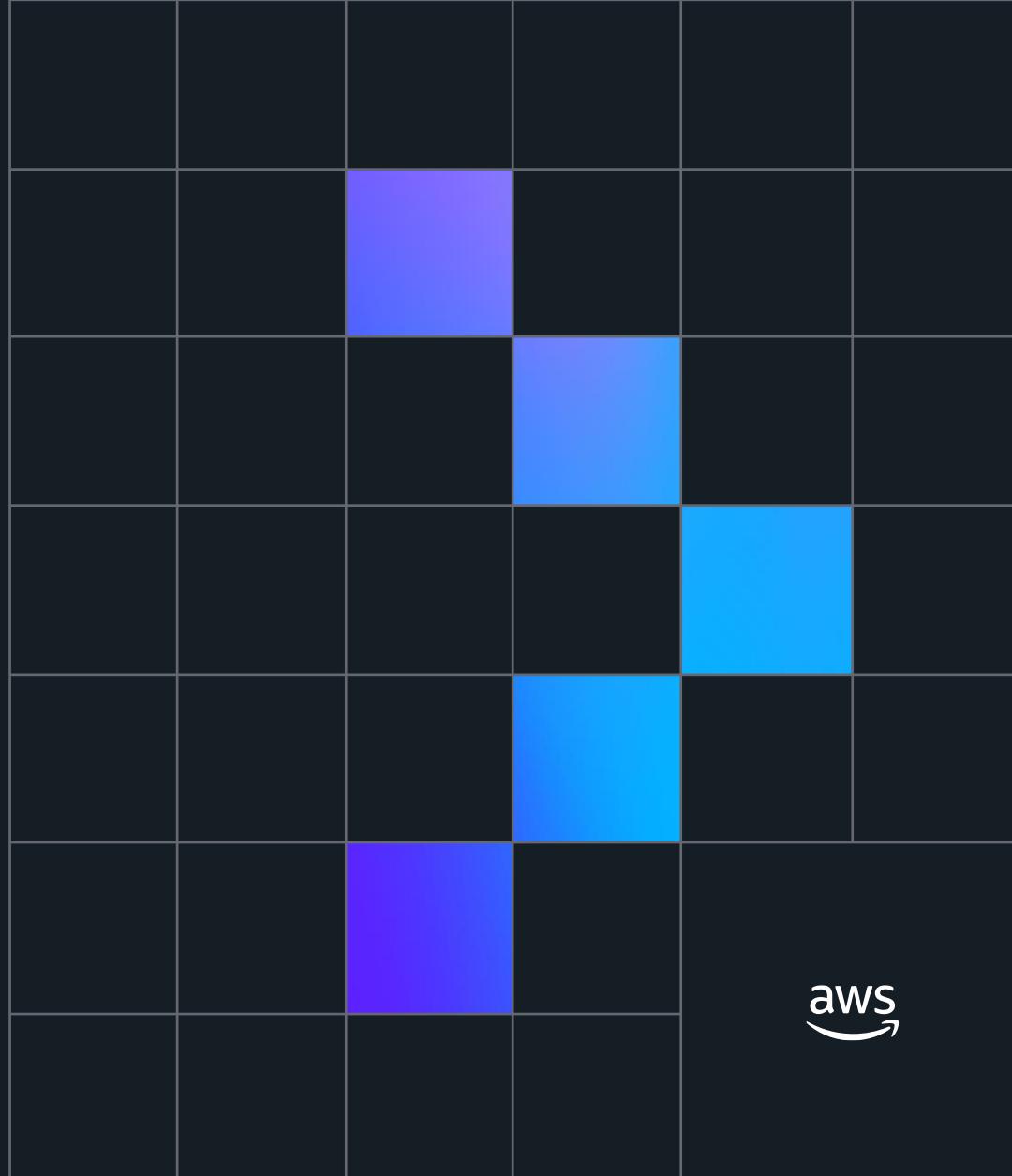
# Entornos de datos más serverless: una guia general



Carlos “Fer” Chicata  
Data & Solution Engineer - TRD

# Agenda

- La importancia de la arquitectura de datos
- Entendamos los artefactos de una arquitectura
- Definiendo “Serverless”
- Artefactos serverless
- Consejos y buenas practicas



Geoffrey Moore dijo:

“Sin análisis de big data, las empresas están ciegas y sordas, vagando por la web como ciervos en una autopista.”



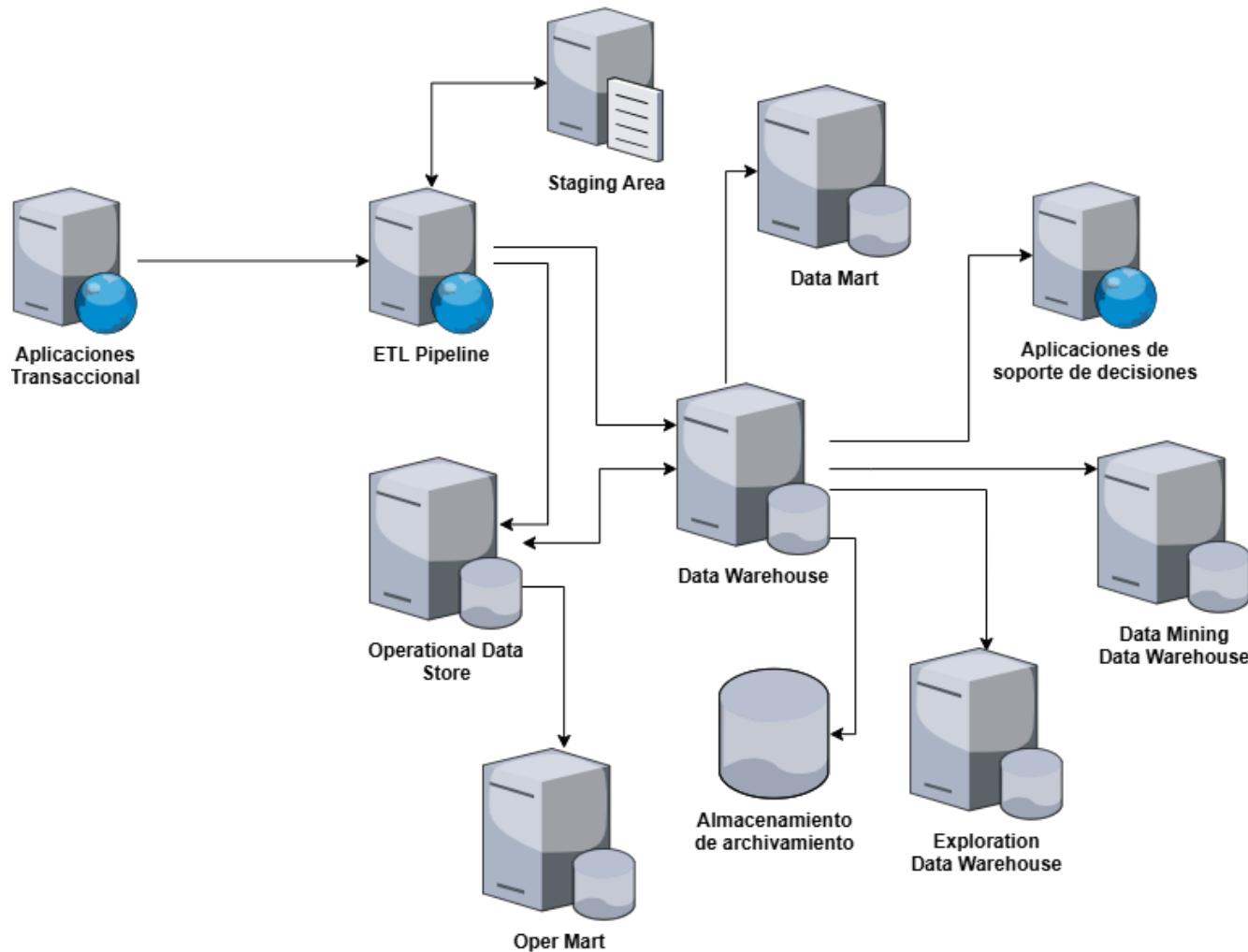
# La realidad empresarial

- Análisis de datos específicos.
- Requerimientos únicos de entrega de los datos.
- Colaboración dentro y entre áreas.
- Formas de trabajar actuales.
- Métricas y umbrales específicos.
- Historia del área.
- Herramientas utilizadas.
- Procesos de cada área.



“Tener datos confiables e  
integrables para actuar y  
comprender”

# La clave: la arquitectura de datos



La importancia de la  
arquitectura de datos

# Clasificación (personal) de artefactos de la arquitectura de datos

## Procesamiento

Procesamiento de datos

## Almacenamiento

Almacenamiento

Interacción

Gestión de mensajes

# Lista de aplicaciones de la arquitectura de datos

## CAPAS DE ARCHIVAMIENTO

- Guardado de datos de todo tipo que son poco usados.
- Eficiente en costo y disponibilidad.

## DATA LAKE

- Receptor de todos los tipos de datos.
- La clave es la organización de los datos.
- Enfoque operativo.

## DATA MART

- Análisis de datos específico y limitado.
- El enfoque depende de los requerimientos.

## DATA WAREHOUSE

- Datos corporativos integrales históricos para analítica.
- Enfoque estratégico y táctico.
- "Fuente de verdad".

## OPER MART

- Similar al Data Mart pero con un enfoque operativo y datos actualizados.
- Depende del enfoque de los requerimientos.

## OPERATIONAL DATA STORE

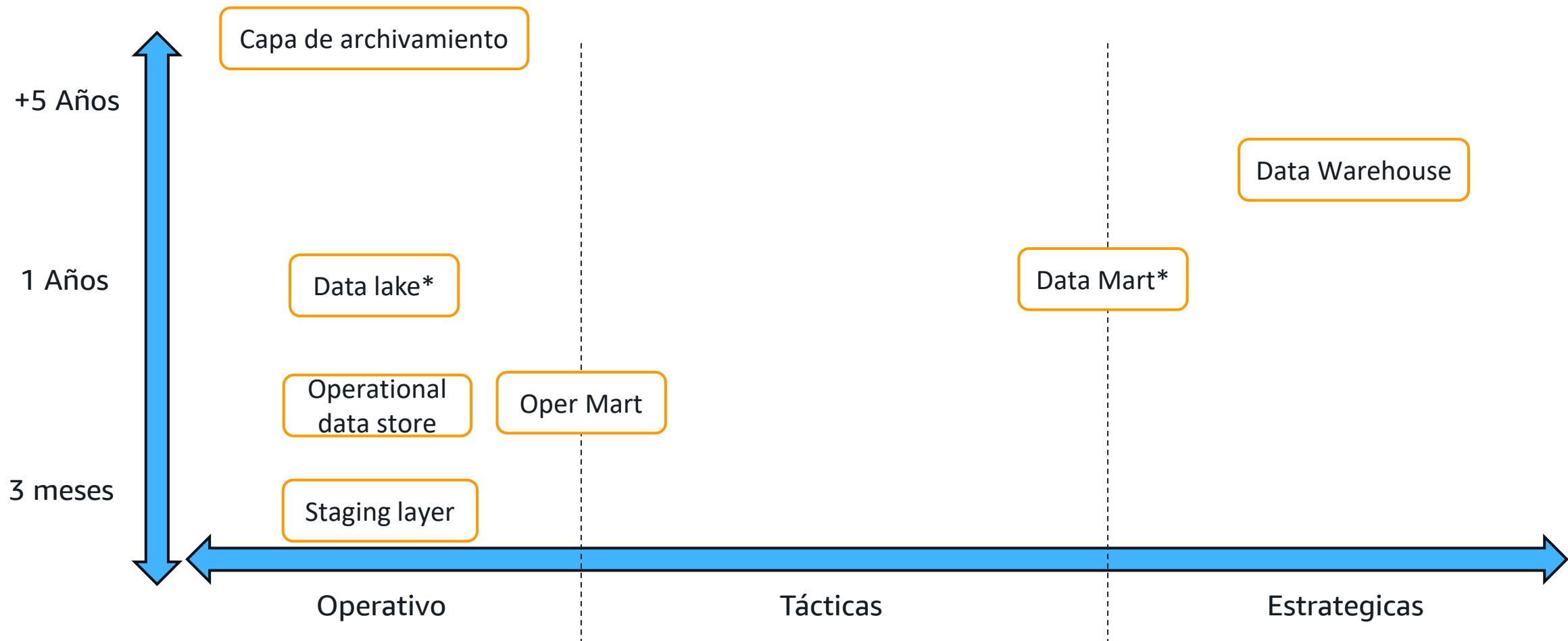
- Datos corporativos integrales recientes para analítica.
- Enfoque operativo y táctico.
- No hay mucho o nada de data histórica.

## STAGING LAYER

- Almacenamiento de apoyo para ETL pipelines.
- La temporalidad de los datos es debatible.
- Entorno exclusivo.

# APPS DE DATOS

# Funcionalidad de los artefactos



Facto #1:  
"Artefacto > herramienta"

## Facto #2:

“Metricas y umbrales + artefactos gestionados + servicio adecuados = buena arquitectura”

# Servicios en AWS: tipo

## Serverfull



## Serverfull



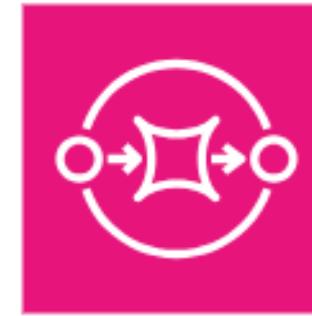
AWS Aurora

## Serverless



AWS Aurora  
Serverless V2

## Serverless



AWS SQS

# ¿Qué significa realmente “serverless”?

(parece fácil pero no)

# Una definición general

## SERVERLESS

Esto significa:

- #1: Pagar por lo que se consume solamente.
- #2: No preocuparse por el escalamiento.
- #3: No necesitar administrar o aprovisionar servidores



Yan Cui – AWS Serverless Hero

# Características serverless generales

- (+) Nula o mínima gestión de infraestructura.
- (+) Mecanismo automáticos de disponibilidad y/o escalabilidad.
- (+) Costo eficiencias en gastos.
  
- (-) Limitaciones por hardware y software.
- (-) Rendimiento afectado por “cold start”.
  
- (?) Agilidad en el ciclo de desarrollo de software
- (?) Tener una definición de carga de trabajo “impredecible” y “variable”.

# Serverless en los tipos de artefactos

## Procesamiento de datos

- (+) Escalamiento automático.
- (+) Minima o nula gestión de infraestructura.
- (+) Pago solo por ejecución.
- (+) Escalamiento automático.

- (-) Template o limites en cambio de hardware y software.
- (-) Rendimiento afectado por "Cold Start".

## Almacenamiento de datos

- (+) Minima o nula gestión de base de datos.
- (+) Escalamiento y Disponibilidad automática.
- (+) Pago solo por ejecución y almacenamiento.

(?) Costo mínimo por almacenamiento.

- (-) Definir carga "variable".
- (-) Rendimiento afectado por "Cold Start".
- (-) Template o limites en cambio de hardware y software.

## Gestión de mensajes

- (+) Minima o nula gestión de plataformas de mensajes.
- (+) Escalamiento y Disponibilidad automática.
- (+) Pago solo por ejecución, red y/o almacenamiento.

- (-) Template o limites de hardware para almacenamiento.
- (-) Limites para consumo de datos.
- (-) Rendimiento por latencia y throughput.

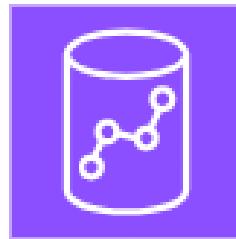
# ¿Cuán “serverless” puede ser mis artefactos?

# Servicios serverless y tipos de artefactos



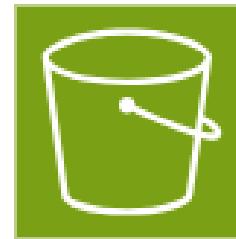
# Servicios serverless y artefactos populares

Data Warehouse



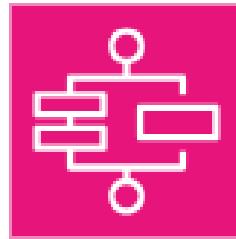
AWS Redshift

Data Lake



AWS S3

Orquestamiento



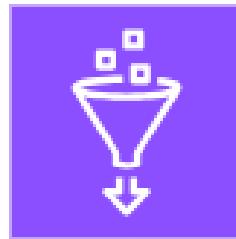
AWS Step Function

Interacción de datos



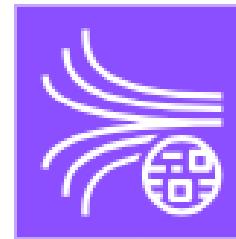
AWS Athena

Integración de datos



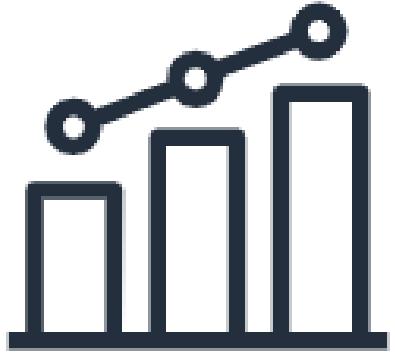
AWS Glue

Gestión de flujo de datos

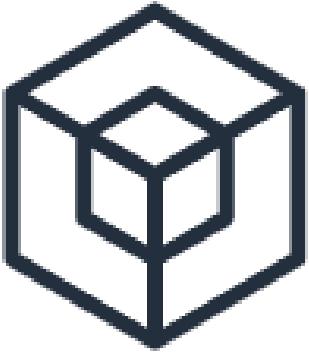


AWS Kinesis  
Data Stream

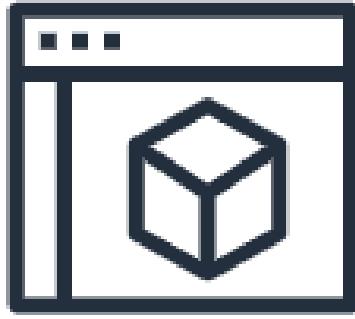
# Los criterios clave en la decisión



Métricas del  
negocio & equipo



Artefacto a  
implementar



Servicios serverless  
A utilizar



Novedades de  
la comunidad

Los criterios clave en la decisión



Métricas del  
negocio & equipo

ES RELATIVO

Artefactos  
implementados

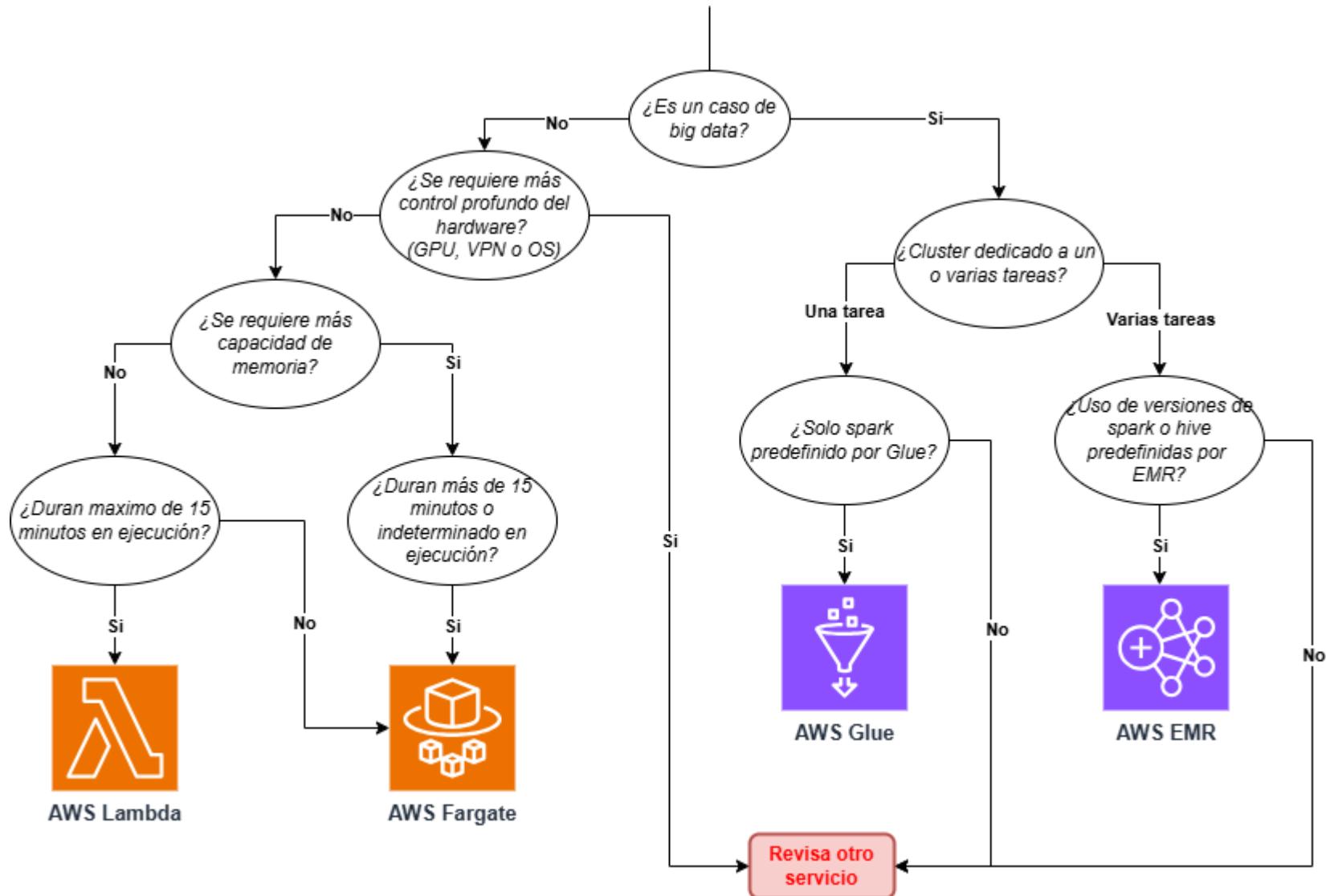
Servicios serverless

A utilizar

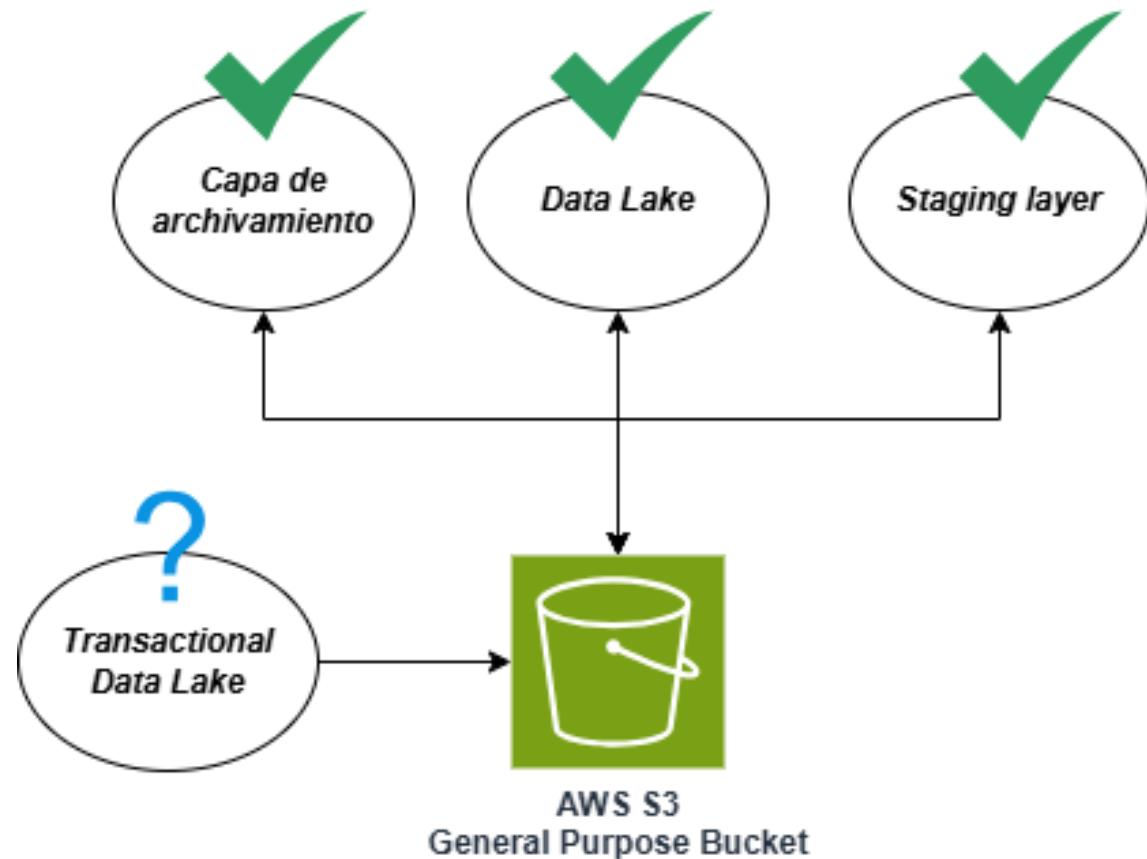


Novedades de  
la comunidad

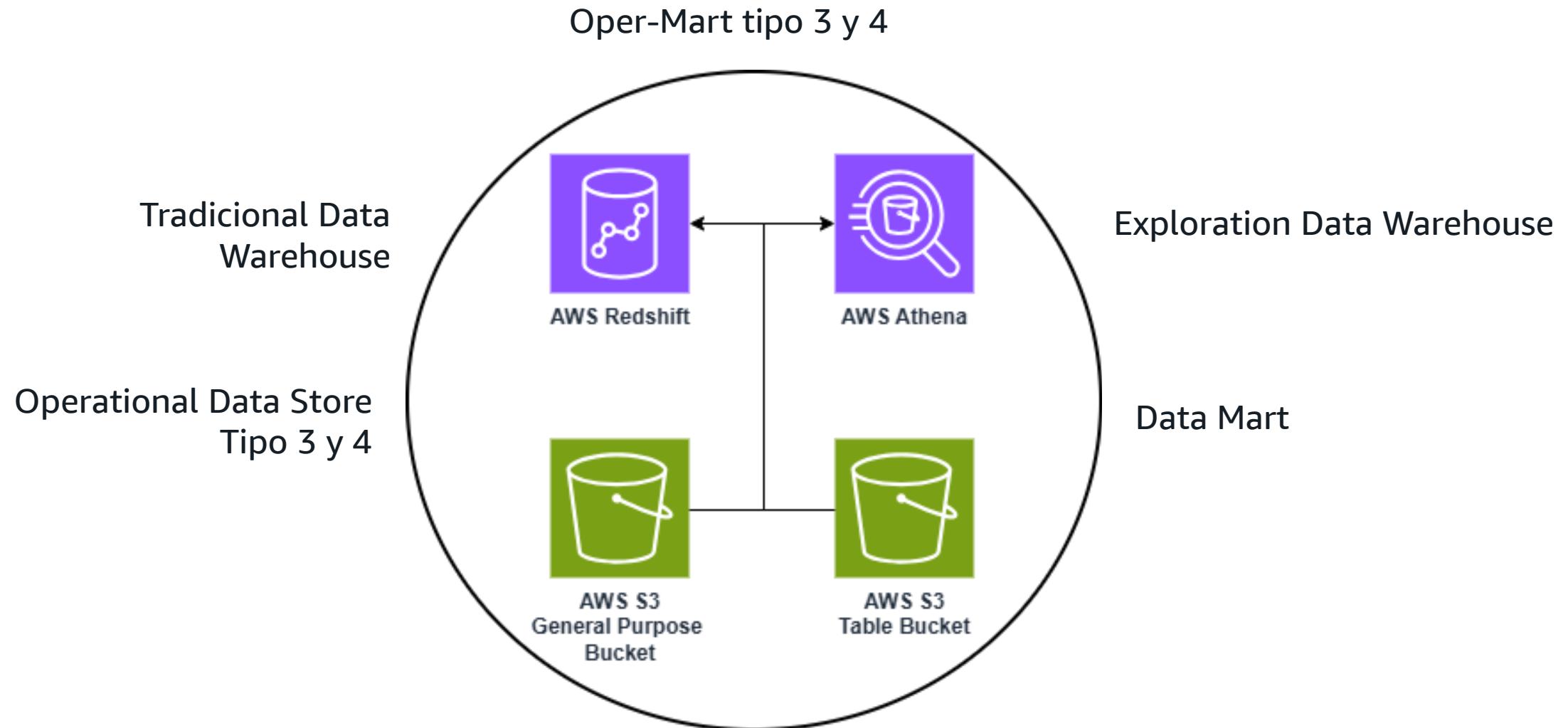
# Caso #1: Procesamiento de datos - Transformación



## Caso #2: Almacenamiento



## Caso #3: Almacenamiento



# Criterios para evaluar los artefactos serverless

## Procesamiento de datos

- #1 Tiempo de ejecución.
- #2 Capacidad de hardware y software a usar.
- #3 Funcionalidades a realizar.
- #4 Modelo de costos.
- #4 Gestiones para actividades de desarrollo (deploy, control de versiones, dependencias, etc)

## Almacenamiento de datos

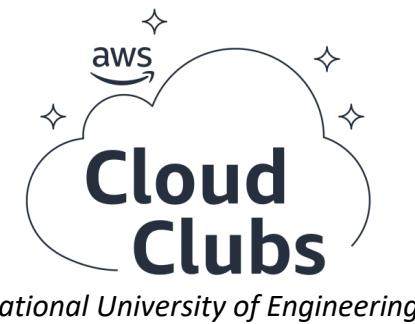
- #1 Tiempo de ejecución.
- #2 Evolución de la comunidad de usuarios.
- #3 Capacidad de software a usar.

## Gestión de mensajes

- #1 Como gestiona los datos.
- #2 Latencia & Throughput para consumo
- #3 Capacidad de almacenamiento.

## Interacción

- #1 Integración con otros servicios para flujos de trabajo.
- #2 Gestión operativa ante el equipo en situaciones de ejecución, recuperación ante incidentes y observabilidad.



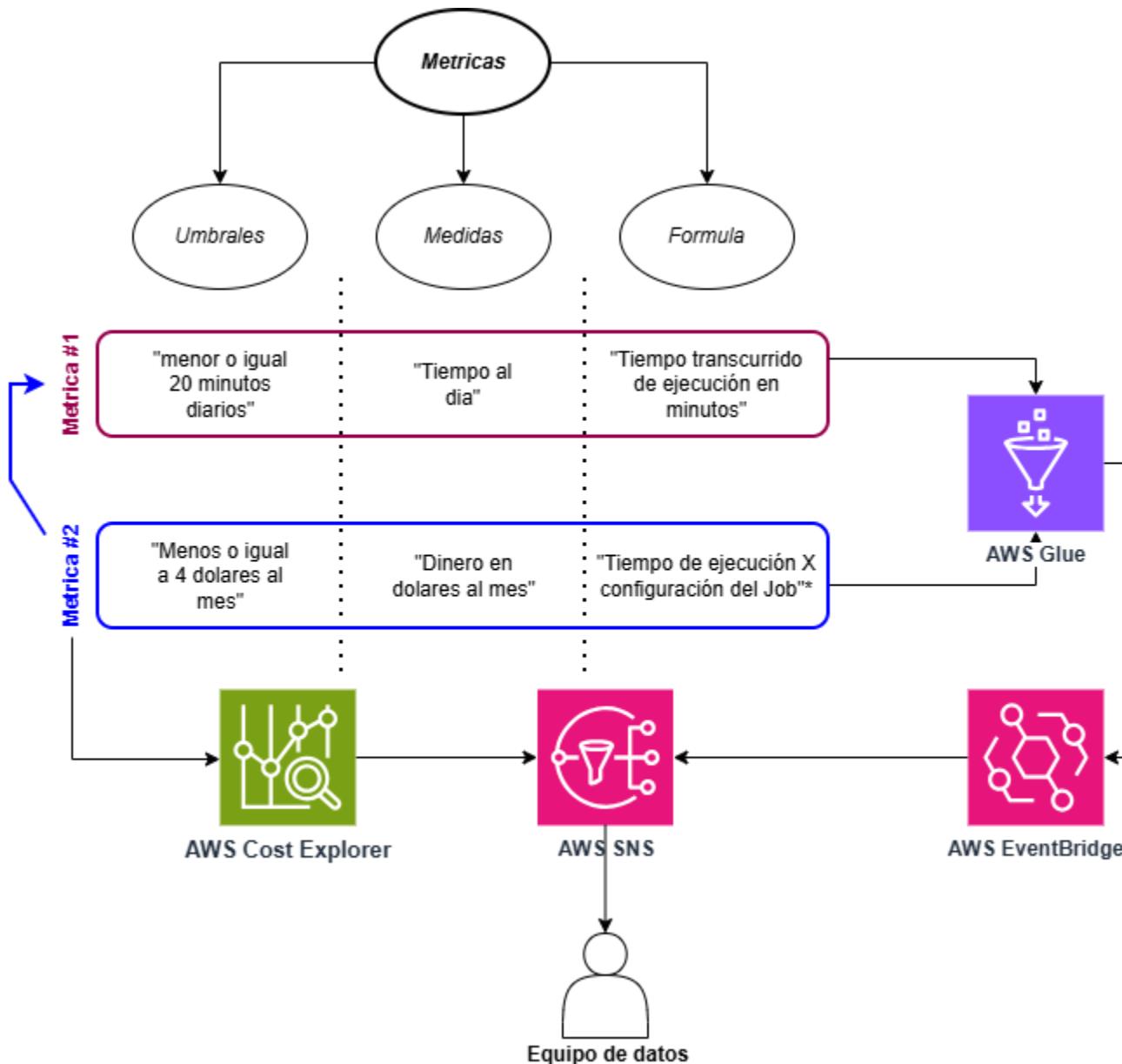
# Consejos y sus buenas prácticas



Consejos y buenas  
prácticas

Consejo #1:  
"Entiende las metricas de  
negocio desde el punto  
técnico"

# Consejo #1: Ejemplo



# Consejo #1: Buenas prácticas



# Consejo #2: “Monitoreo y observabilidad everywhere”

## Consejo #2: Buenas prácticas

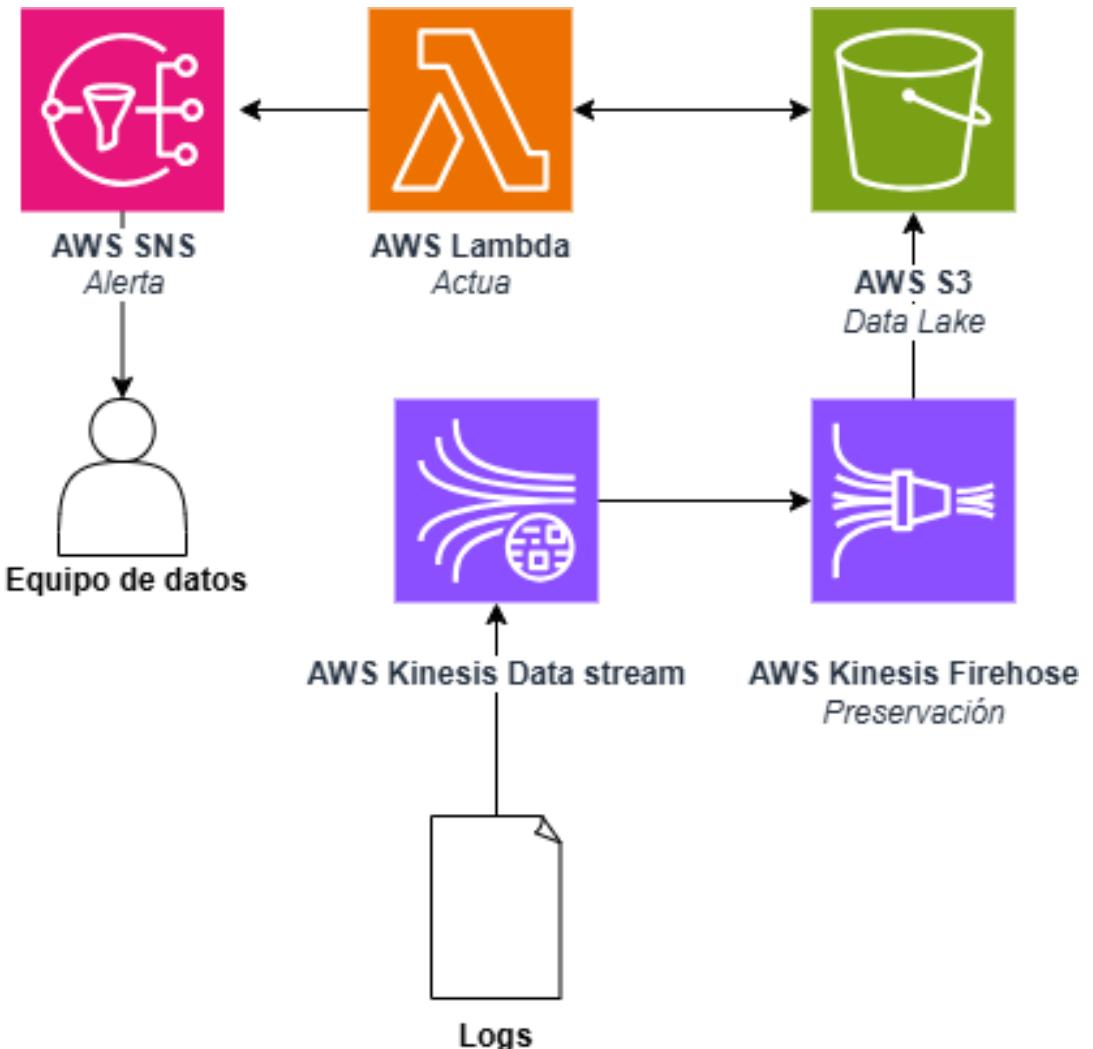
#1: Ten un estándar en tus logs.

- Fechas, proceso, tarea, error, artefactos involucrados.

#2: Centraliza la información en un data lake.

#3 Monitorea situaciones, no eventos aislados.

#4 Ten un plan de contingencia e incidentes automatizados lo máximo posible.



¡Todo beneficios trae complejidad como complicaciones y eso esta bien: aprende a gestionarlo!



# ¡Gracias totales!

 Sigueme en mis redes!



Carlos “Fer” Chicata  
Data & Solution Engineer



# Q&A



# ¿Qué les parecio la charla?

Dime en la encuesta que te parecio



Carlos “Fer” Chicata  
Data & Solution Engineer

