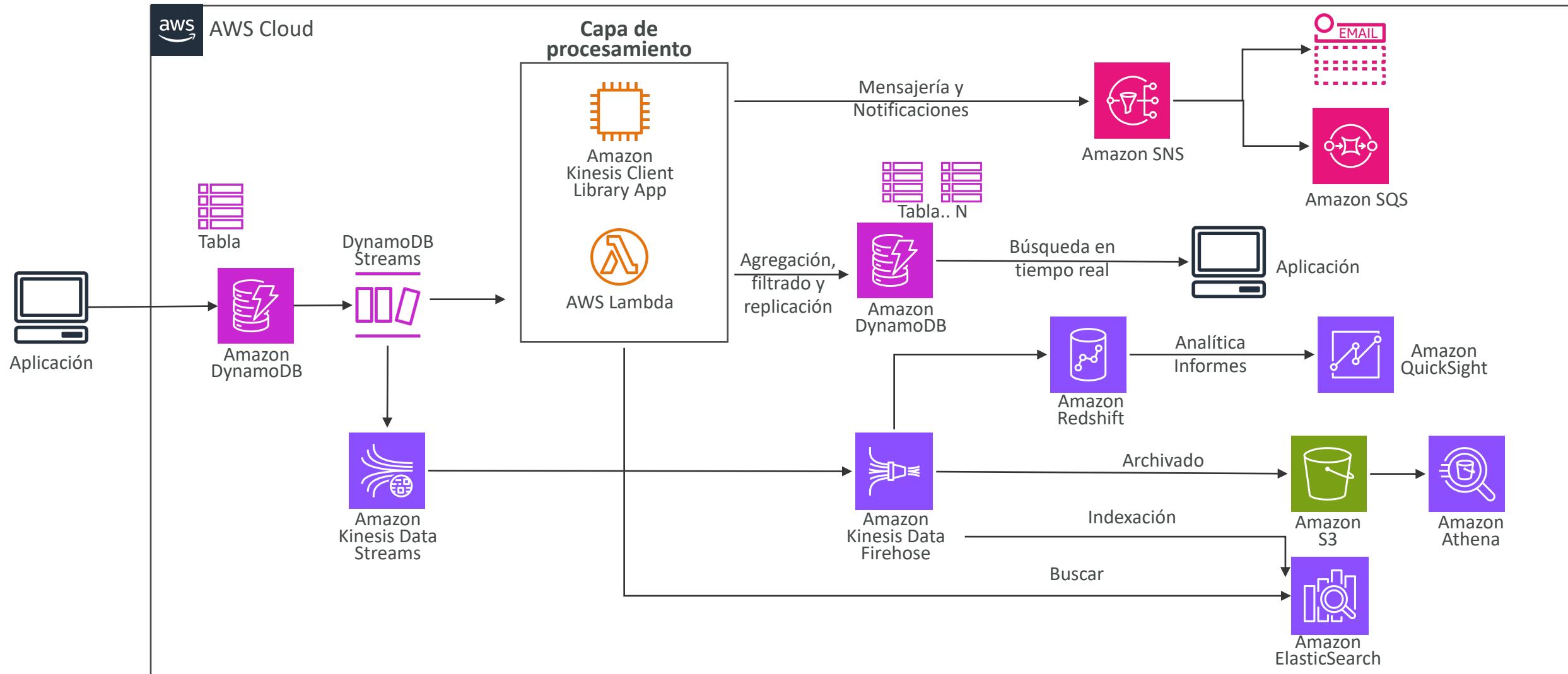
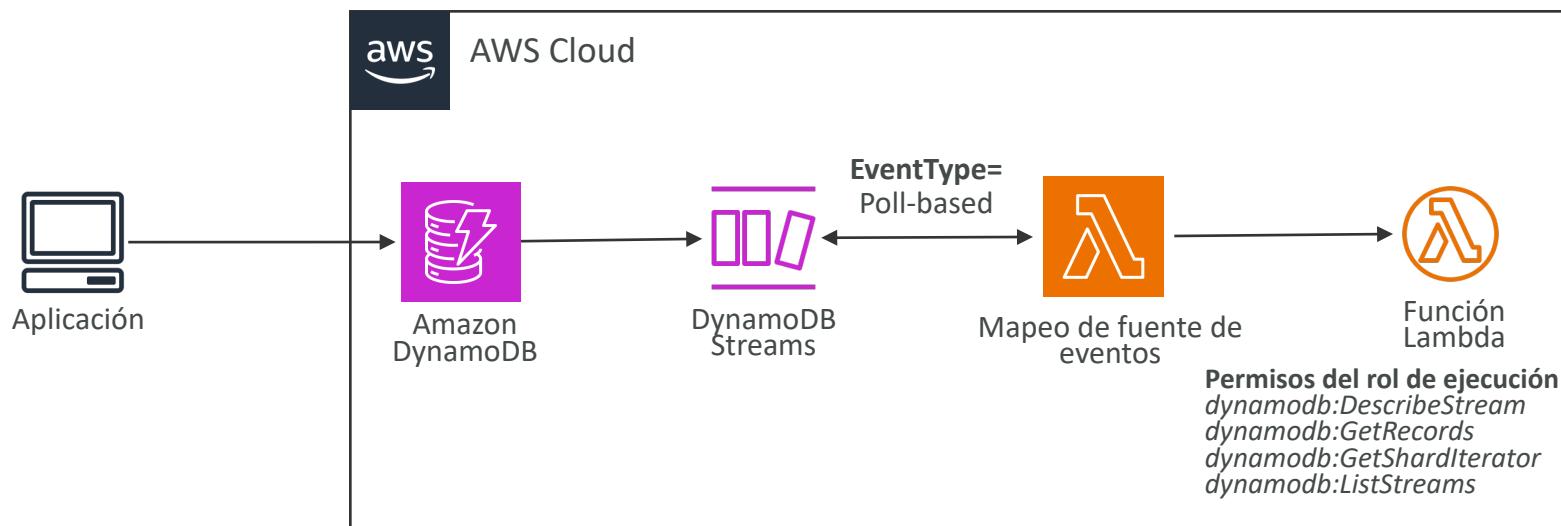


# DynamoDB Streams



# DynamoDB Streams y AWS Lambda

- Necesitas definir un **mapeo de fuente de eventos** para leer de un DynamoDB Streams
  - \*El mapeo de fuente de eventos automáticamente invoca la función Lambda en respuesta a eventos específicos detectados en la fuente de eventos
- Necesitas asegurarte de que la función Lambda tiene los **permisos adecuados**
- **Tu función Lambda se invoca de forma sincrónica**

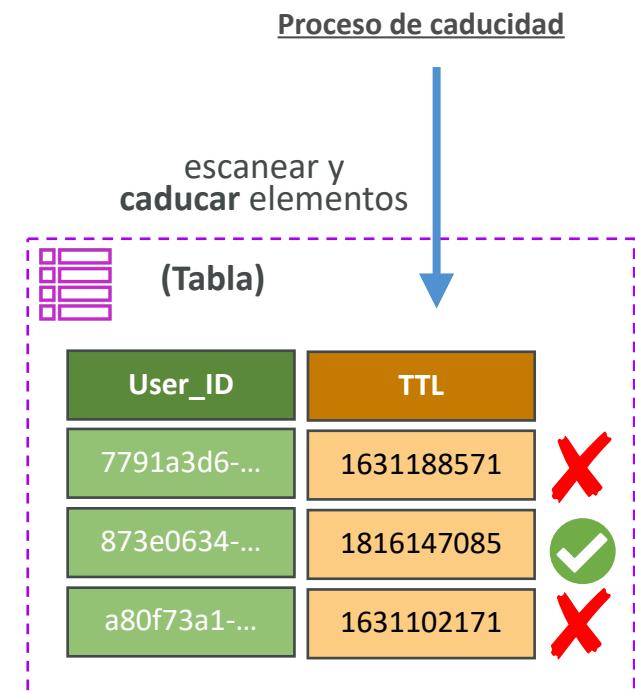


```
{  
  "Records": [  
    {  
      "eventID": "1",  
      "eventVersion": "1.0",  
      "dynamodb": {  
        "Keys": {  
          "Id": {  
            "N": "101"  
          }  
        },  
        "NewImage": {  
          "Message": {  
            "S": "New item!"  
          },  
          "Id": {  
            "N": "101"  
          }  
        },  
        "StreamViewType": "NEW_AND_OLD_IMAGES",  
        "SequenceNumber": "111",  
        "SizeBytes": 26  
      }  
    }  
  ]  
}
```

# DynamoDB – Time To Live (TTL)

- Elimina automáticamente los elementos después de una fecha de caducidad
- No consume ninguna WCU (es decir, no tiene coste adicional)
- El atributo TTL debe ser un tipo de dato “**Number**” con valor “**Unix Epoch timestamp**”
- Los artículos caducados se eliminan en las 48 horas siguientes a su caducidad
- Los elementos caducados, que no se han borrado, aparecen en las lecturas/consultas/escaneos (si no los quieres, filtralos)
- Casos de uso: reducir los datos almacenados conservando sólo los elementos actuales, cumplir las obligaciones normativas, ...

Domingo, 19 de Mayo de 2024, 19:31:25 PM  
(Epoch timestamp: 1716147085)



# CLI de DynamoDB - Detalles relevantes

- Con la CLI de DynamoDB puedes ejecutar consultas y escaneos directamente desde la línea de comandos
- Por ejemplo, si deseamos escanear una tabla:

```
aws dynamodb scan --table-name TuNombreDeTabla
```

# CLI de DynamoDB - Detalles relevantes

- Algunos detalles importantes son:
  - **--projection-expression**: uno o más atributos a recuperar
  - **--filter-expression**: filtra los elementos antes de devolvértelos
- Por ejemplo:

```
aws dynamodb scan \
  --table-name TuNombreDeTabla \
  --projection-expression "Atributo1, Atributo2" \
  --filter-expression “Atributo2 = 'ValorEspecifico'”
```

# CLI de DynamoDB - Detalles relevantes

- Opciones generales de paginación de la CLI de AWS:
  - **--page-size**: especifica que la CLI de AWS recupere la lista completa de elementos, pero con un mayor número de llamadas a la API en lugar de una sola (por defecto: 1000 elementos)
  - **--max-items**: número máximo de elementos a mostrar en la CLI (devuelve **NextToken**)
  - **--starting-token**: especifica el último **NextToken** para recuperar el siguiente conjunto de elementos

# Transacciones DynamoDB

- Operaciones coordinadas de **todo o nada**
- Por ejemplo, **añadir/actualizar/eliminar** en varios elementos de una o varias tablas
- Proporciona atomicidad, consistencia, aislamiento y durabilidad (ACID)
- **Consumo el doble de WCUs y RCUs**
  - DynamoDB realiza 2 operaciones por cada elemento (preparar y confirmar)
- **Casos de uso:**
  - Transacciones financieras
  - Gestión de pedidos
  - Juegos Multijugador



# Transacciones de DynamoDB - Cálculo de capacidad

-  **MUY IMPORTANTE** para el examen! 
- **Ejemplo I:** 5 escrituras transaccionales por segundo, con un tamaño de elemento de 4 KB

## SOLUCIÓN

Necesitamos  $5 * \left( \frac{4 \text{ KB}}{1 \text{ KB}} \right) * 2$  (*transactional cost*) = 40 WCUs

# Transacciones de DynamoDB - Cálculo de capacidad

-  ¡MUY IMPORTANTE para el examen! 
- **Ejemplo 2:** 10 lecturas de transacciones por segundo, con un tamaño de elemento de 9 KB

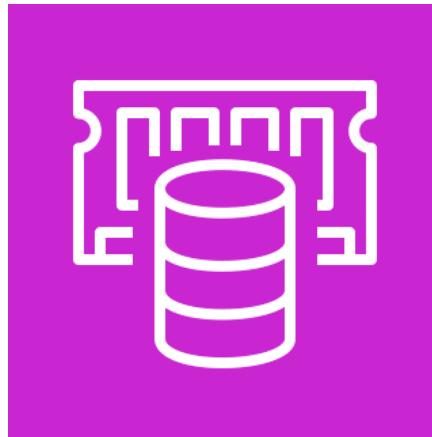
## SOLUCIÓN

Necesitamos  $10 * \left( \frac{12 \text{ } KB}{4 \text{ } KB} \right) * 2$  (*transactional cost*) = 60 RCU's

\*(9 se redondea al 4 KB superior)

# DynamoDB vs. ElastiCache

## ElastiCache



- ElastiCache es en memoria, pero DynamoDB es sin servidor
- Ambos son almacenes de claves/valores

# DynamoDB vs. EFS

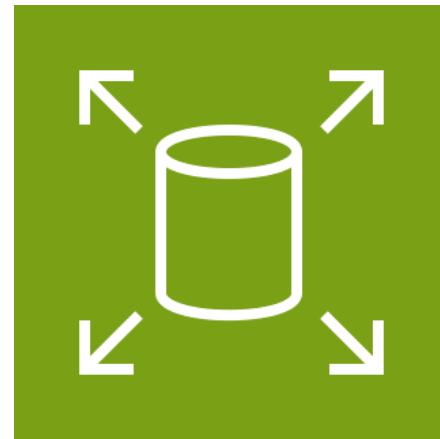
EFS



- EFS debe conectarse a las instancias EC2 como una unidad de red

# DynamoDB vs. EBS / Instance Store

EBS



- EBS e Instance Store sólo pueden utilizarse para el almacenamiento en caché local, no para el almacenamiento en caché compartido

# DynamoDB vs. S3

S3



- S3 tiene mayor latencia y no está pensado para objetos pequeños

# Fragmentación de escritura en DynamoDB

- Imagina que tenemos una aplicación de votación con dos candidatos, el **candidato A** y el **candidato B**
- ⚡⚠ Si la **clave de partición** es “**Candidate\_ID**”, esto da lugar a dos particiones, lo que generará problemas (por ejemplo, partición en caliente)
- Una estrategia que permite distribuir mejor los elementos uniformemente entre las particiones
- **Añade un sufijo al valor de la clave de partición**

Clave de partición

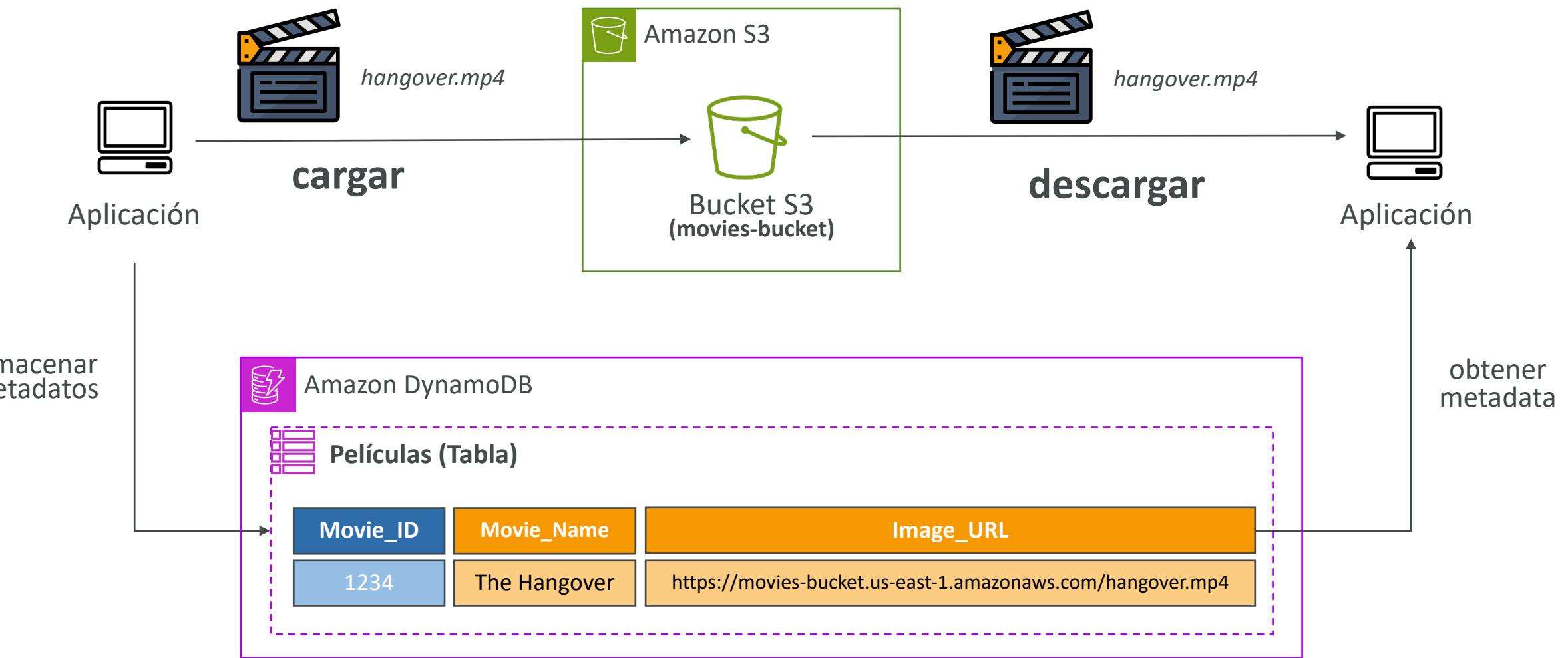
| Candidate_ID   |
|----------------|
| Candidate_A-98 |
| Candidate_B-77 |

Atributos

| Voter_ID   |
|------------|
| 439812123A |
| 219834522b |

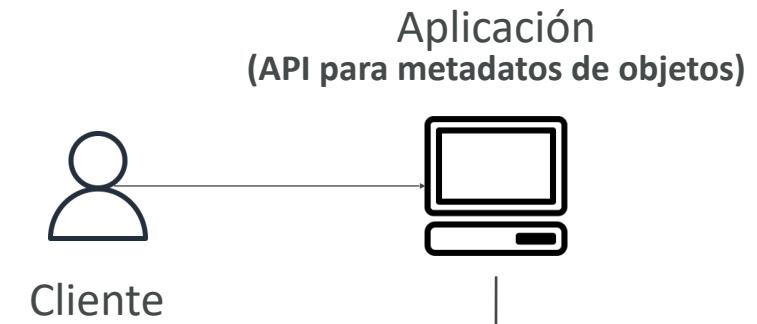
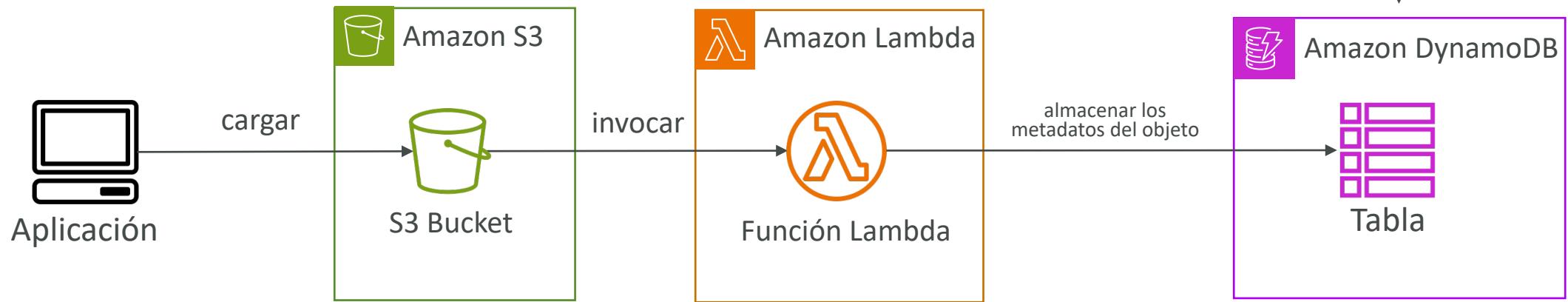


# DynamoDB - Patrón de objetos grandes



# DynamoDB - Indexación de metadatos de objetos S3

- Asegurar la consistencia eventual entre los datos almacenados en S3 y sus metadatos indexados en DynamoDB
- Configurar triggers de AWS Lambda para actualizar índices en DynamoDB automáticamente al modificar objetos en S3
- Utilizar políticas de IAM para controlar el acceso a los metadatos en DynamoDB basado en roles de usuario



# DynamoDB - Seguridad y otras características

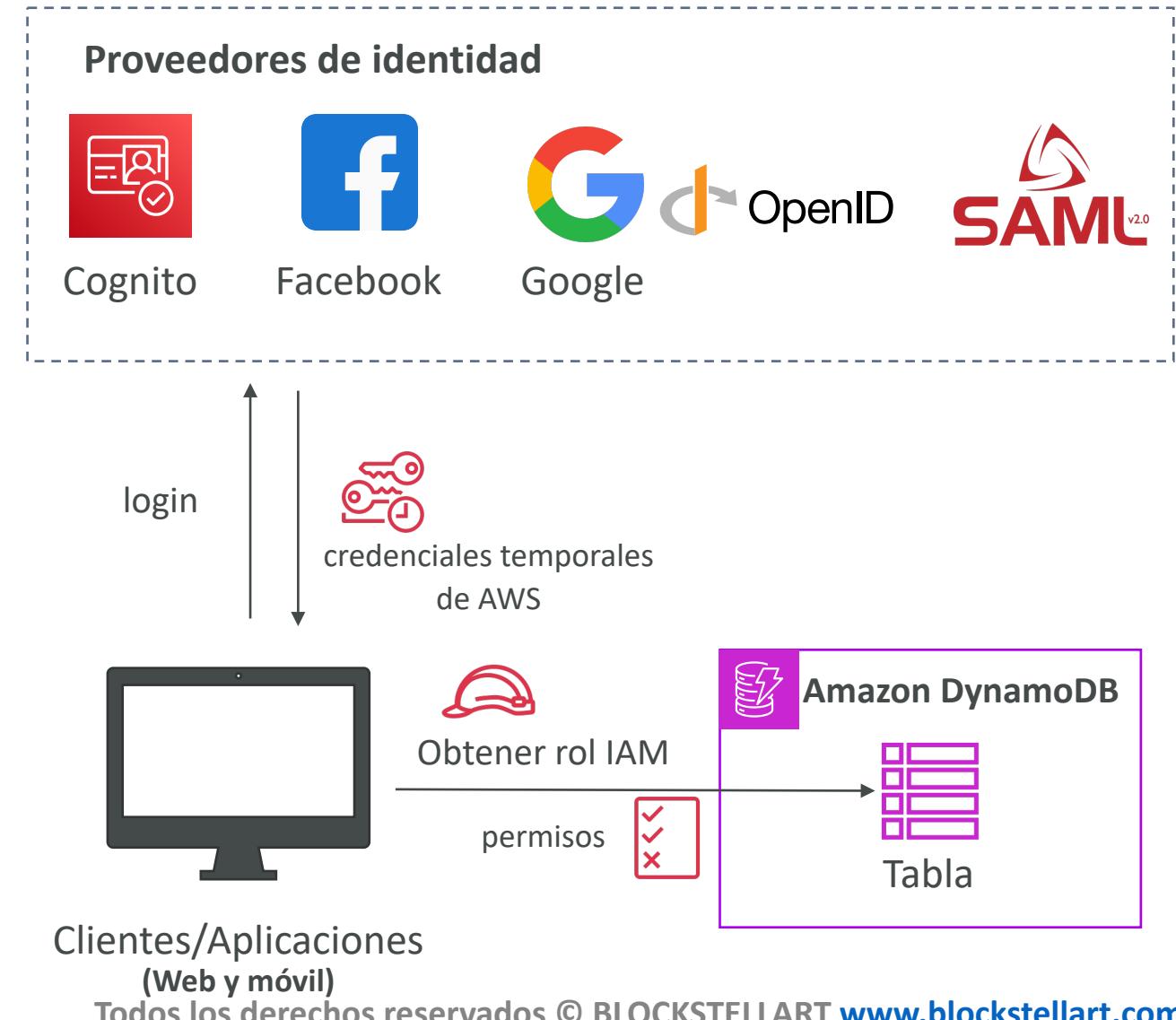
- **Seguridad**
  - **Endpoints VPC:** Acceso seguro a DynamoDB sin necesidad de conexión a Internet
  - **Control de acceso:** Gestión completa mediante AWS IAM para un control granular
- **Cifrado**
  - **En reposo:** Utiliza AWS Key Management Service (KMS) para cifrar datos almacenados
  - **En tránsito:** Protección mediante SSL/TLS para garantizar la seguridad de los datos durante su transmisión
- **Resiliencia de datos**
  - **Copia de seguridad y restauración:** Opciones integradas para respaldar y restaurar datos fácilmente
  - **Recuperación Puntual (PITR):** Similar a RDS, permite la restauración de datos a cualquier punto en el tiempo, sin impactar el rendimiento

# DynamoDB - Seguridad y otras características

- **Tablas globales**
  - Configuración multiregión, multiactiva, completamente replicada para un alto rendimiento y disponibilidad
- **DynamoDB local**
  - Permite el desarrollo y pruebas de aplicaciones localmente sin necesidad de conectarse a la red de AWS
- **Migración a DynamoDB**
  - AWS Database Migration Service (DMS): Facilita la migración a DynamoDB desde diversas fuentes como MongoDB, Oracle, MySQL y Amazon S3

# DynamoDB - Interacción con DynamoDB

- Antes de interactuar con DynamoDB, los **usuarios se autentican** a través de proveedores de identidad externos, incluidos Amazon Cognito User Pools, Google, Facebook, OpenID Connect y SAML
- Tras la autenticación, los usuarios obtienen roles de AWS IAM que definen sus permisos específicos para acceder y operar sobre DynamoDB



# DynamoDB - Control de acceso detallado

- Existe la opción de especificar condiciones al conceder permisos mediante una política de IAM:
  - Conceder permisos para que los usuarios puedan obtener acceso de solo lectura a determinados elementos y atributos de una tabla o un índice secundario
  - Conceder permisos para que los usuarios puedan obtener acceso de solo escritura a determinados atributos de una tabla, según la identidad del usuario en cuestión

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html>

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "LimitAccessToCertainAttributesAndKeyValues",  
            "Effect": "Allow",  
            "Action": [  
                "dynamodb:UpdateItem",  
                "dynamodb:GetItem",  
                "dynamodb:Query",  
                "dynamodb:BatchGetItem"  
            ],  
            "Resource": [  
                "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",  
                "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores/index/TopScoreDateTimeIndex"  
            ],  
            "Condition": {  
                "ForAllValues:StringEquals": {  
                    "dynamodb:LeadingKeys": [  
                        "${graph.facebook.com:id}"  
                    ],  
                    "dynamodb:Attributes": [  
                        "attribute-A",  
                        "attribute-B"  
                    ]  
                },  
                "StringEqualsIfExists": {  
                    "dynamodb:Select": "SPECIFIC_ATTRIBUTES",  
                    "dynamodb:ReturnValues": [  
                        "NONE",  
                        "UPDATED_OLD",  
                        "UPDATED_NEW"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

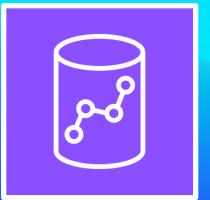


# Amazon Redshift

[www.blockstellart.com](http://www.blockstellart.com)

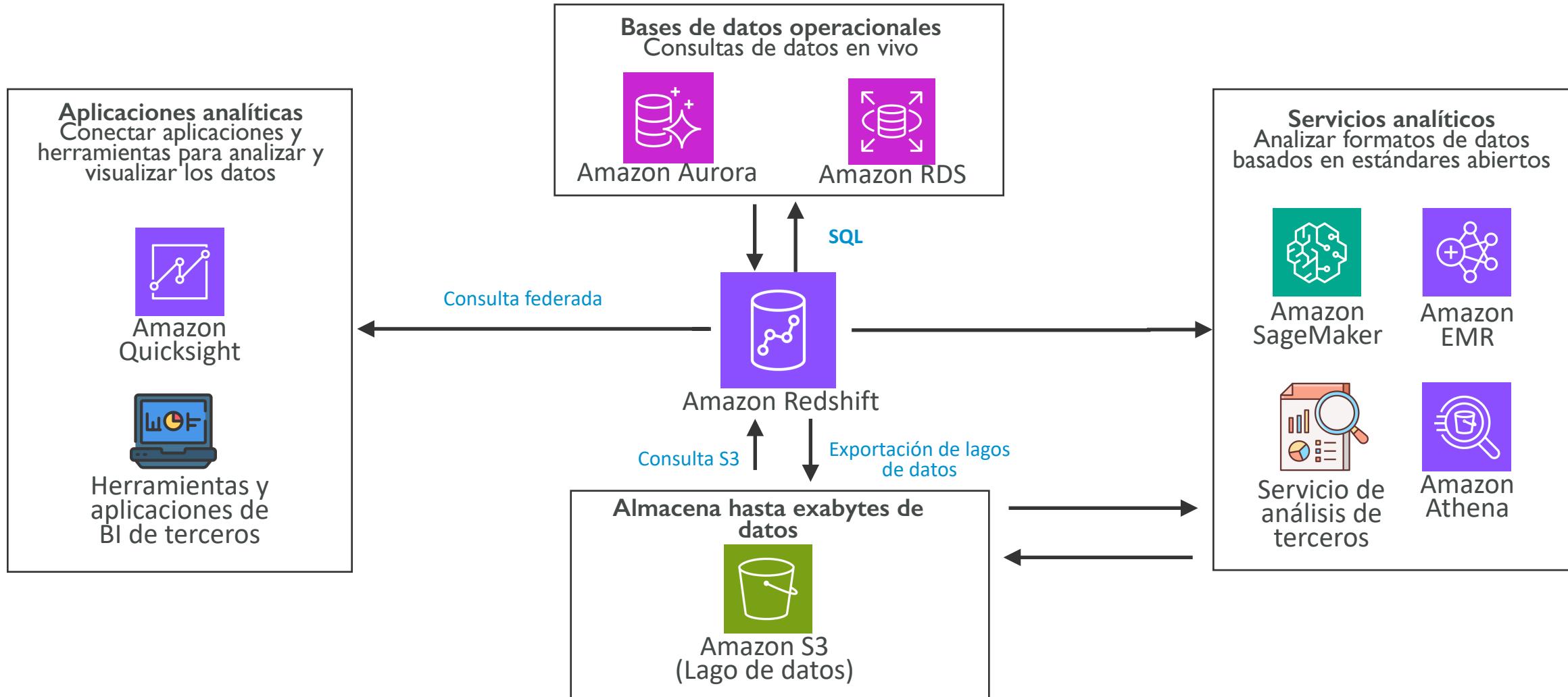
Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Visión general de Redshift

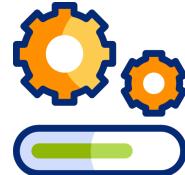


- Redshift se basa en PostgreSQL, **pero no se utiliza para OLTP**
- **Es OLAP - procesamiento analítico en línea (análisis y almacenamiento de datos)**
- Carga los datos una vez cada hora, no cada segundo
- Rendimiento 10 veces superior al de otros almacenes de datos, escala a PBs de datos
- Almacenamiento de datos **en columnas** (en lugar de en filas)
- Ejecución de consultas en paralelo masivo (MPP), con alta disponibilidad
- Paga a medida que avanza en función de las instancias aprovisionadas
- Tiene una interfaz SQL para realizar las consultas
- Las herramientas de BI, como AWS Quicksight o Tableau, se integran con ella

# Visión general de Redshift



# Casos de uso de Redshift



Acelerar las cargas de trabajo analíticas



Modernización del almacén de datos



Almacenar datos históricos de transacciones bursátiles



Analizar impresiones de anuncios y clics



Almacén de datos unificado y lago de datos



Analizar datos de ventas globales



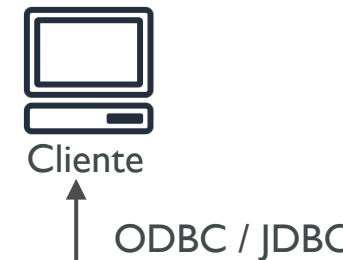
Agregar datos de juegos



Analizar tendencias sociales

# Arquitectura de Redshift

- **Nodo líder:** Coordina las consultas y la distribución de tareas entre los nodos de cómputo, sin almacenar datos ni procesar consultas directamente.



- **Nodos de cómputo:**

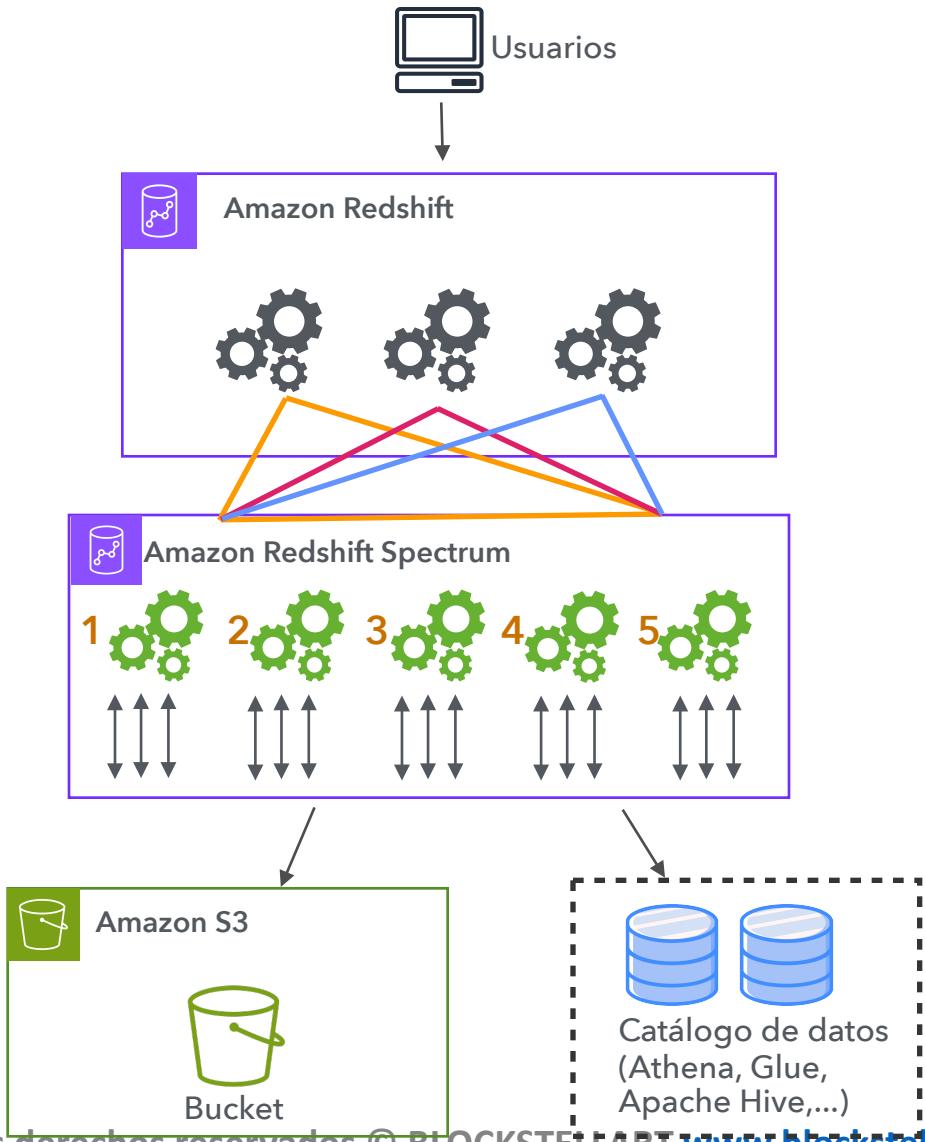
- **Densidad de almacenamiento (DS)**
- **Densidad de cómputo (DC)**

- **Densidad de almacenamiento (DS):** Ideal para grandes cantidades de datos, con más almacenamiento pero menos capacidad de procesamiento
- **Densidad de cómputo (DC):** Optimizados para un procesamiento rápido de consultas, ofreciendo más CPU y RAM pero menos espacio de almacenamiento

\*Node Slices=subdivisiones de un nodo de cómputo

# Redshift Spectrum

- Permite la consulta de exabytes de datos no estructurados en S3 sin necesidad de cargarlos previamente
- Soporta múltiples usuarios y consultas simultáneas sin limitaciones
- Capacidad de aumentar recursos horizontalmente para manejar más carga de trabajo sin degradar el rendimiento
- Los recursos de almacenamiento y computación están desacoplados, lo que permite optimizar costos y escalabilidad
- Soporta una amplia variedad de formatos de datos
- Compatible con compresión Gzip y Snappy para una gestión eficiente del almacenamiento



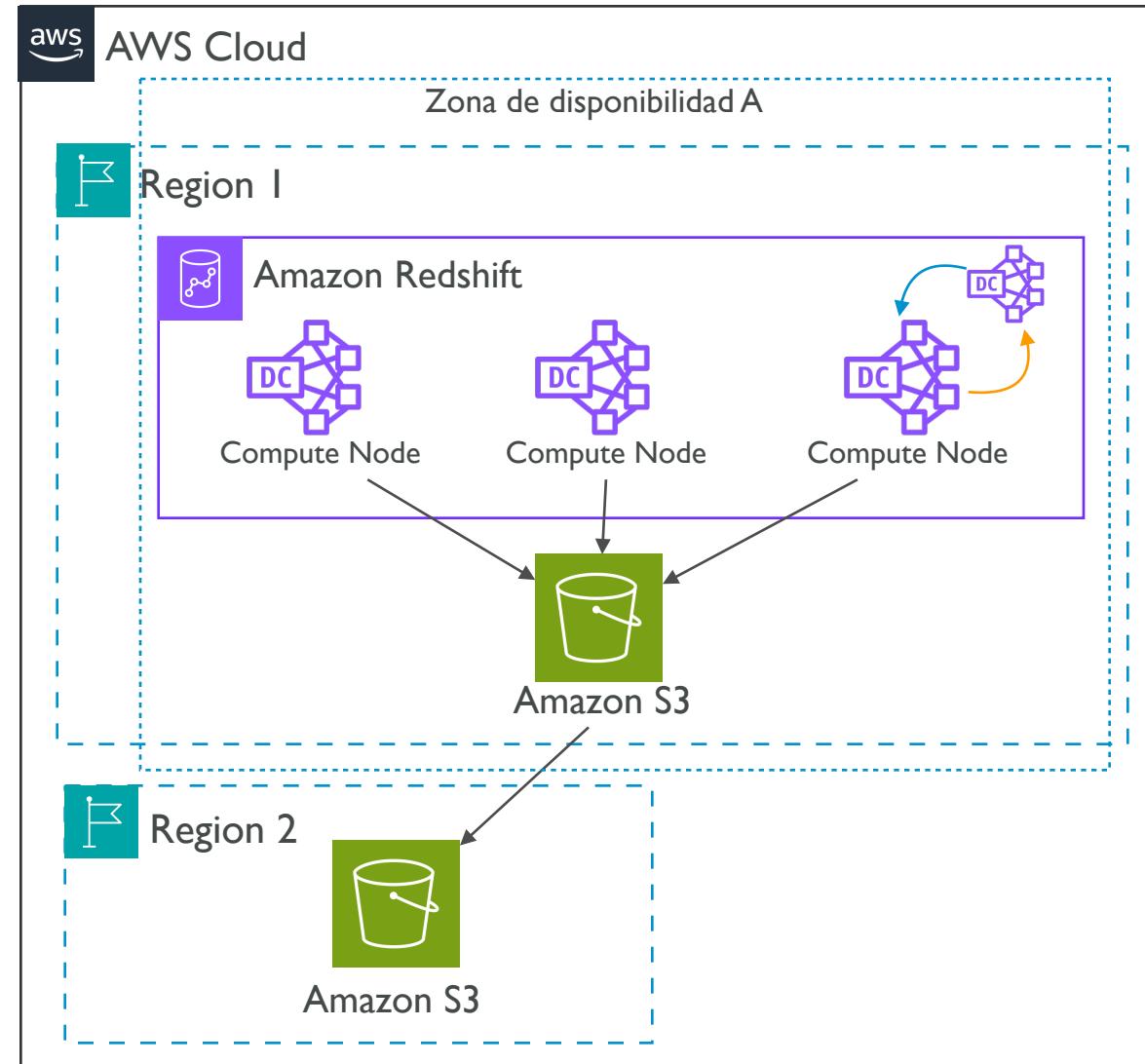
# Rendimiento de Redshift

- **Procesamiento Paralelo Masivo (MPP):** Aprovecha la arquitectura MPP para ejecutar consultas complejas de manera eficiente y rápida sobre grandes volúmenes de datos
- **Almacenamiento de datos en columnas:** Optimiza el rendimiento de las consultas al almacenar datos en formato columnar, lo que permite un acceso más rápido a las columnas específicas necesarias para las consultas
- **Compresión de columnas:** Mejora la eficiencia del almacenamiento y la velocidad de las consultas al comprimir los datos en las columnas, reduciendo así el espacio de almacenamiento necesario y acelerando el procesamiento de datos



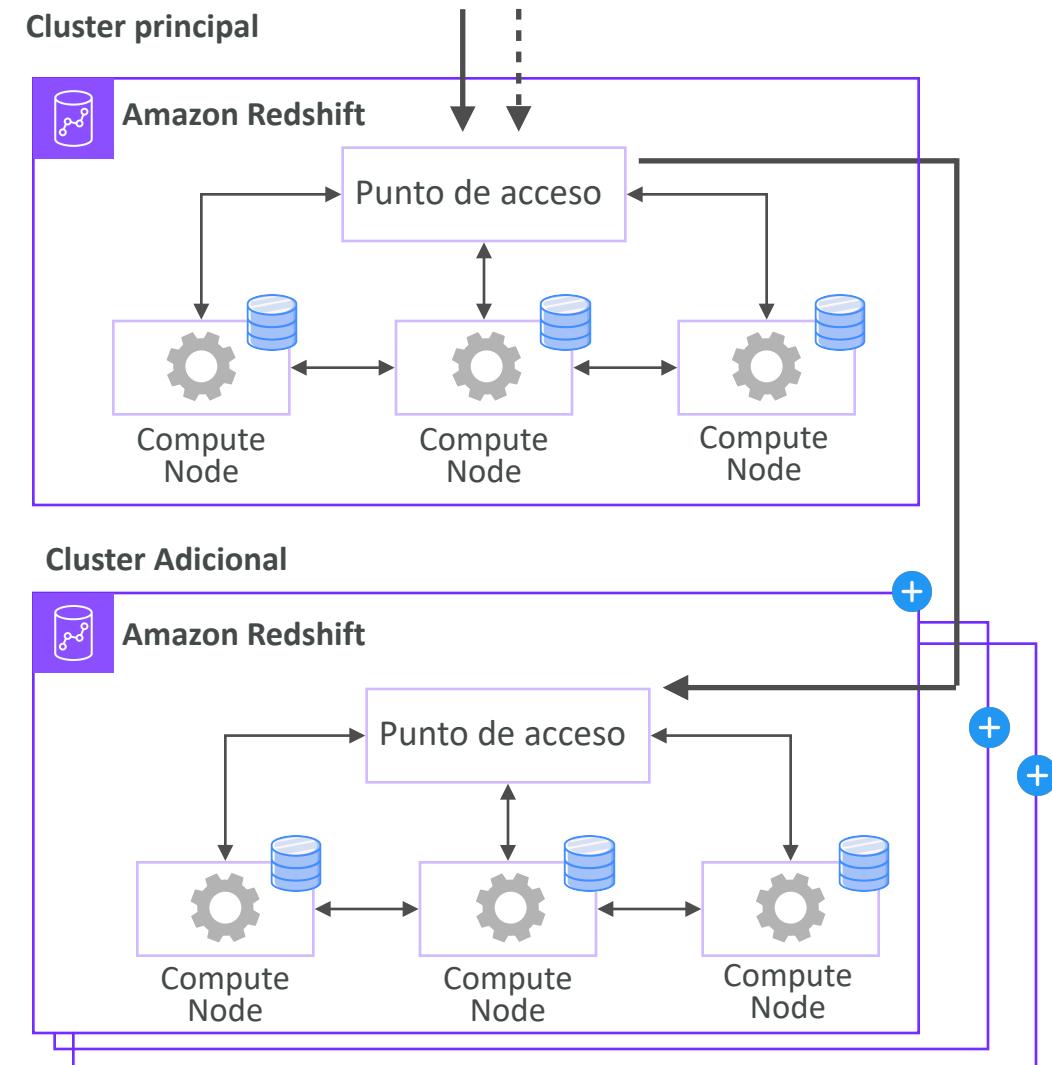
# Durabilidad de Redshift

- **Replicación dentro del clúster:** Garantiza la integridad de los datos mediante la duplicación dentro del mismo clúster
- **Respaldo en S3:**
  - Replicación asincrónica hacia otra región para mayor seguridad
- **Snapshots automatizados:** Los snapshots se crean automáticamente para proteger los datos
- **Reemplazo automático de discos/nodos fallidos:** Los componentes dañados son sustituidos automáticamente sin interrumpir el servicio
- **Limitaciones:**
  - Originalmente limitado a una sola AZ
  - Clústeres RA3 ahora admiten Multi-AZ para mayor resiliencia



# Escalado de Redshift

- **Escalado vertical y horizontal:** Se puede escalar tanto en capacidad como en número de nodos según la demanda
- **Durante el escalado:**
  - Se crea un nuevo clúster mientras el antiguo sigue disponible para lecturas
  - El CNAME se cambia al nuevo clúster (unos minutos de inactividad)
  - Los datos se trasladan en paralelo a los nuevos nodos de cómputo



# Estilos de distribución de Redshift

- Cuando creas una tabla, puedes designar uno de los siguientes estilos de distribución:
    - AUTO, EVEN, KEY o ALL
  - Si no se especifica un estilo de distribución, Amazon Redshift usa la distribución **AUTO**
- 

## • **AUTO**

- Redshift determina automáticamente el estilo de distribución basándose en el tamaño de los datos

## • **EVEN**

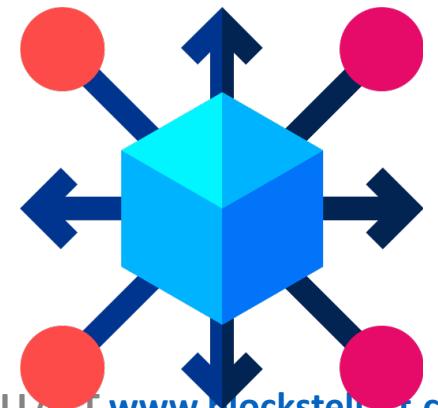
- Las filas se distribuyen equitativamente a través de las particiones en un método round-robin

## • **KEY**

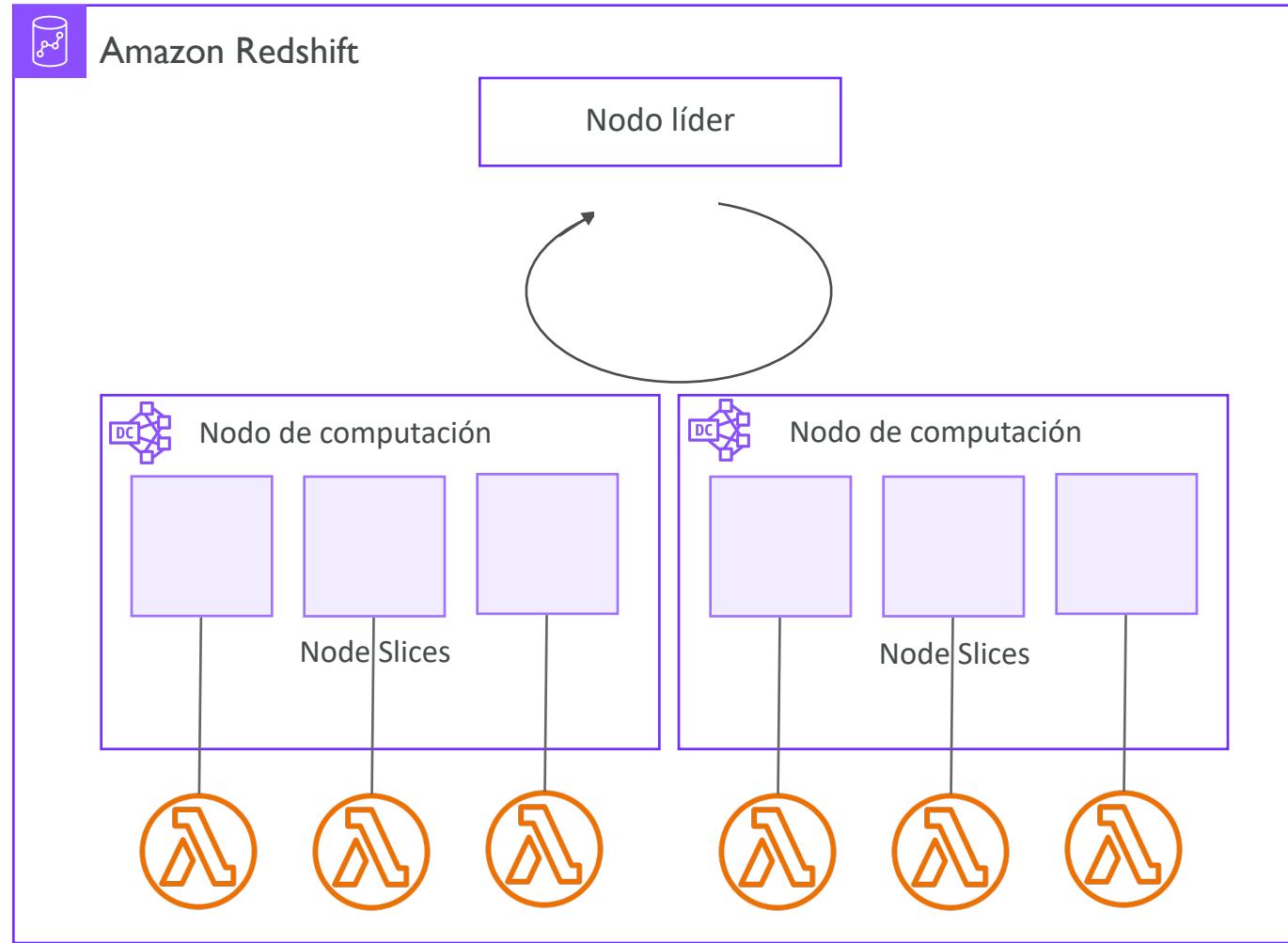
- Las filas se distribuyen basándose en una columna específica

## • **ALL**

- La tabla completa se replica en cada nodo

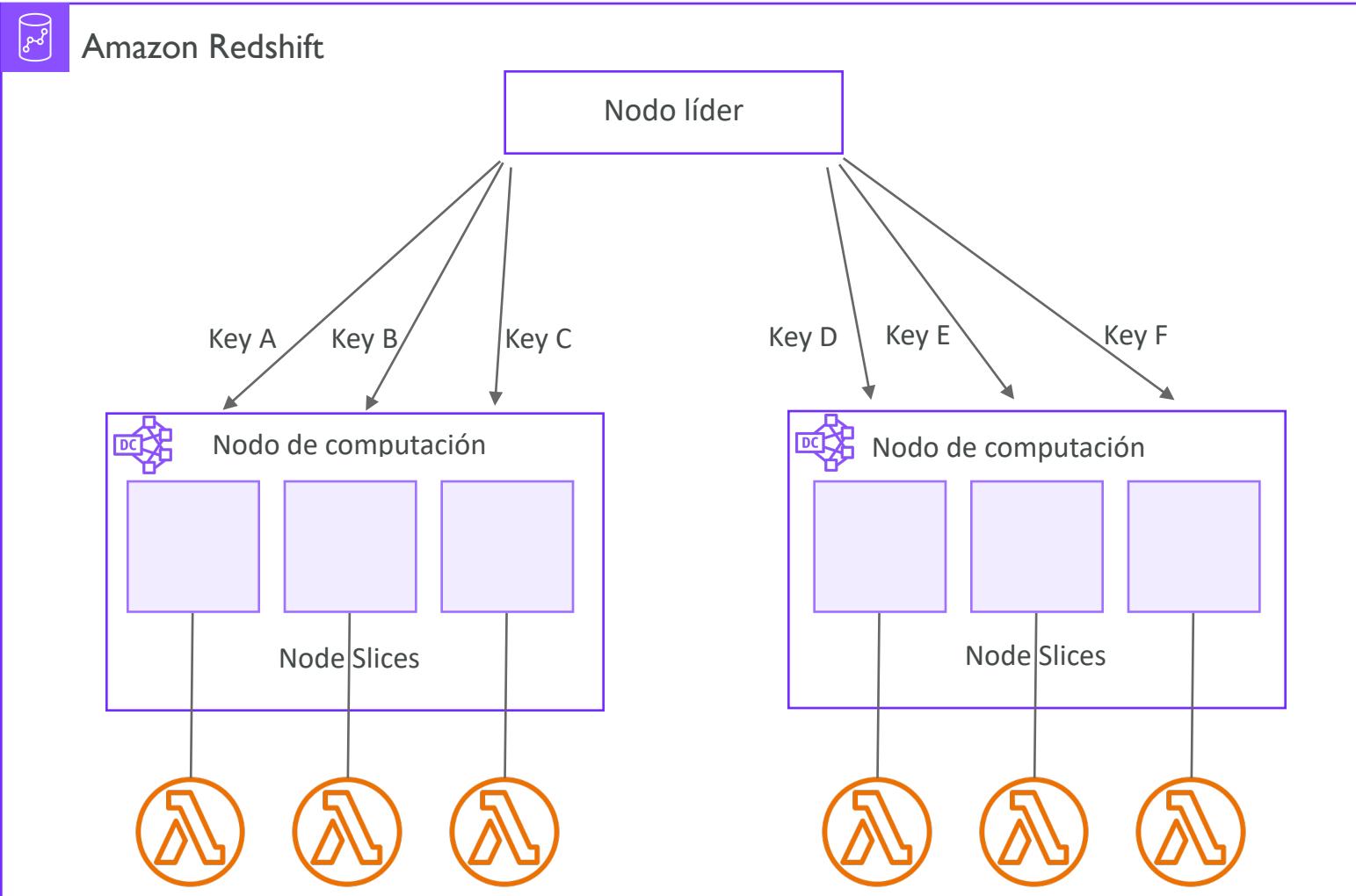


# Distribución EVEN



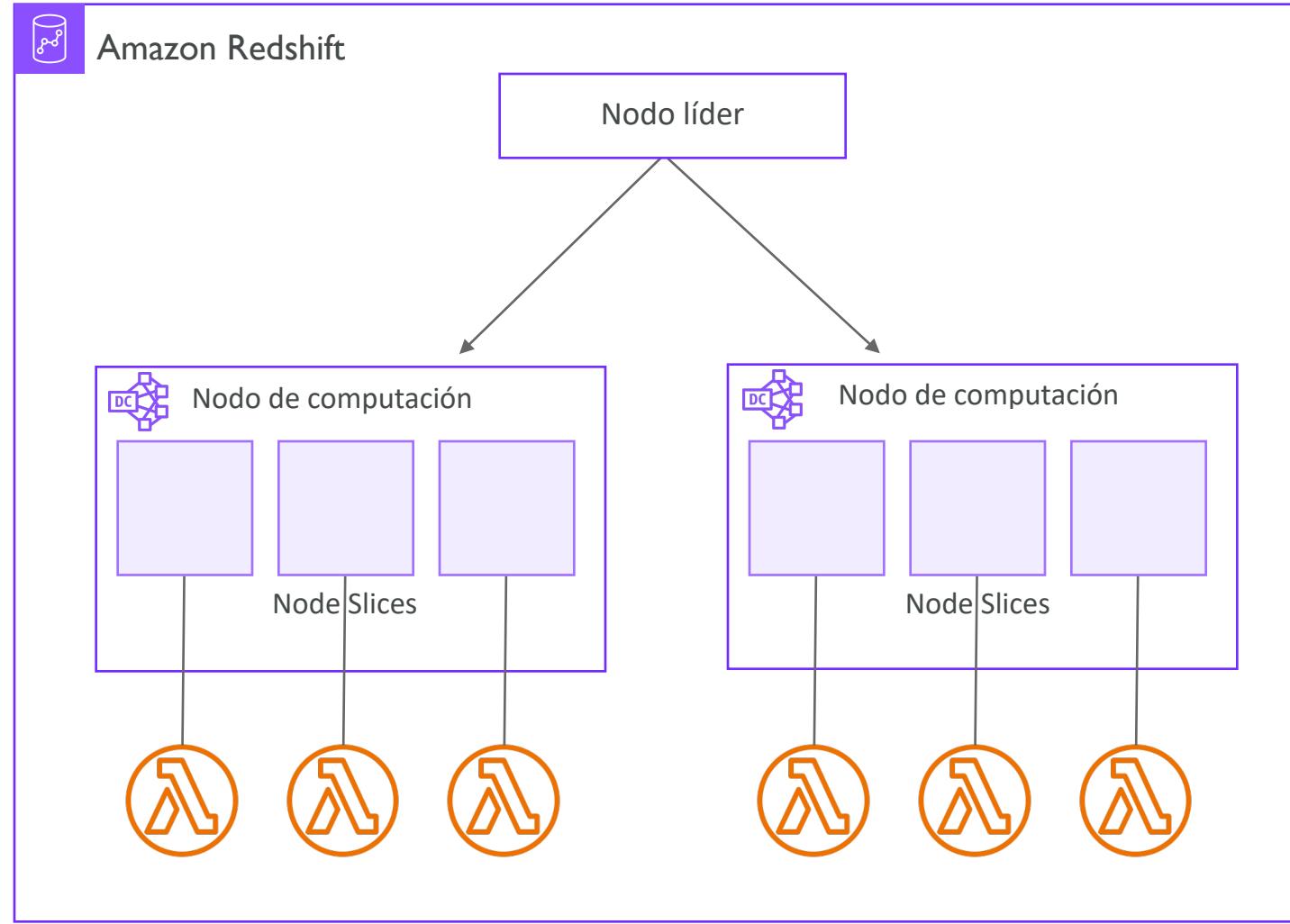
- La distribución **EVEN** permite que los datos y las cargas de trabajo se distribuyan uniformemente entre todos los nodos disponibles
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas
- Esta arquitectura permite una distribución equitativa (EVEN) de la carga de trabajo, mejorando la escalabilidad y la resiliencia del sistema

# Distribución KEY



- La distribución **KEY** permite que los datos se distribuyan de manera eficiente entre todos los nodos disponibles basándose en las claves específicas
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas asignadas por clave
- Esta arquitectura permite una distribución basada en claves (KEY), mejorando la eficiencia y el balance de carga del sistema

# Distribución ALL



- La distribución **ALL** permite que los datos se distribuyan uniformemente entre todos los nodos disponibles
- Cada nodo maneja una porción de la carga total, y dentro de cada nodo, los fragmentos (Node Slices) procesan tareas específicas
- Las funciones Lambda se utilizan para manejar eventos y realizar tareas asignadas de manera eficiente
- Esta arquitectura es ideal para aplicaciones que requieren alta disponibilidad y escalabilidad, ya que permite el procesamiento paralelo de tareas y la rápida respuesta a las solicitudes de los usuarios

# Importación / Exportación de datos

- Comando **COPY**
  - Desde S3, EMR, DynamoDB, hosts remotos
  - S3 requiere un archivo manifiesto y rol de IAM
- Comando **UNLOAD**
  - Descarga datos de una tabla a archivos en S3
- **Otras funcionalidades:**
  - Integración sin ETL de Amazon Aurora
    - Replicación automática desde Aurora a Redshift
  - Ingesta de Streaming en Redshift
    - Desde Kinesis Data Streams o MSK



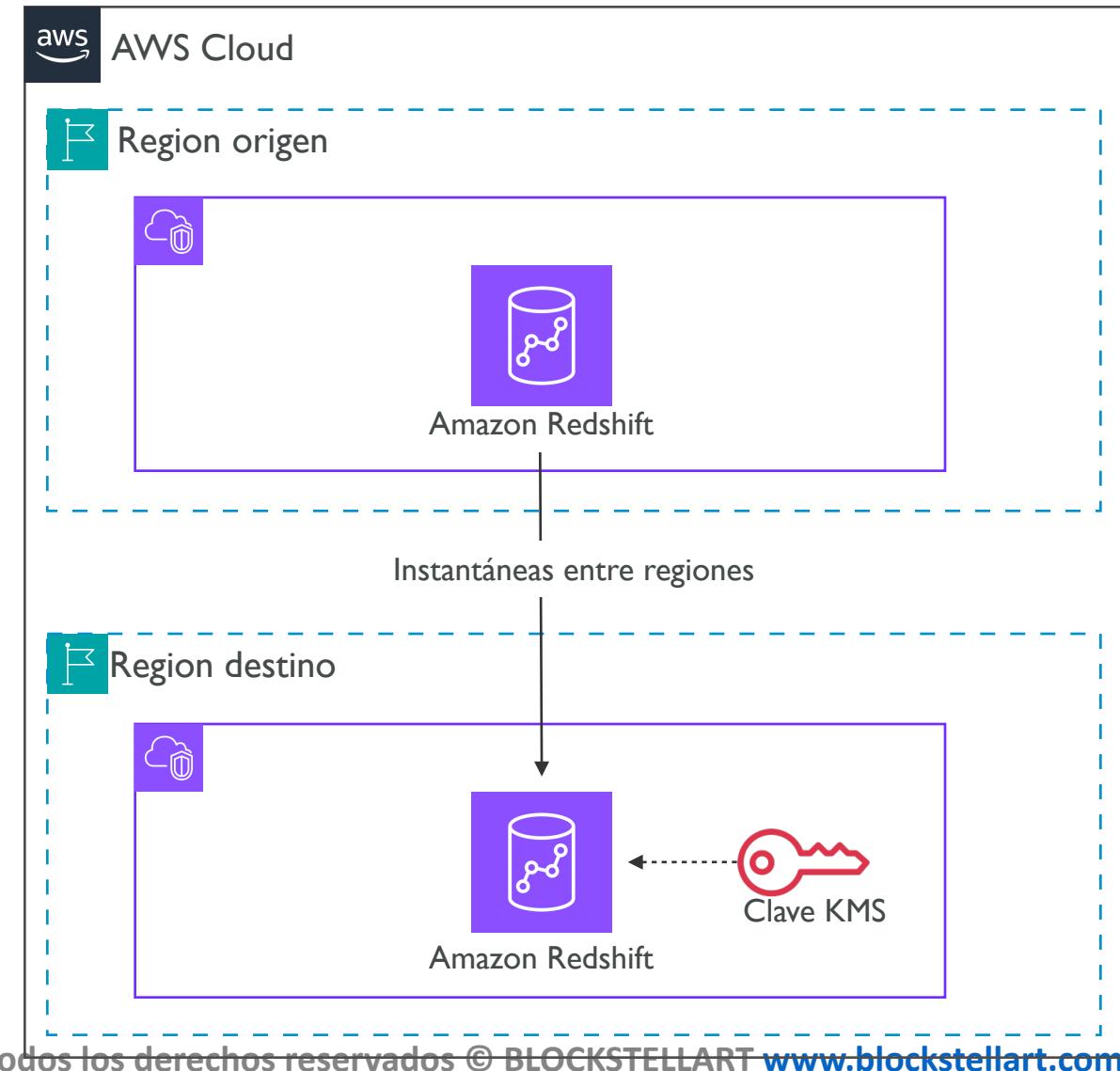
# Comando COPY en profundidad

- **Usa el comando COPY para cargar grandes cantidades de datos desde fuera de Redshift**
- Si los datos ya están en Redshift en otra tabla:
  - Usa: *INSERT INTO <...> SELECT*
  - O también puedes usar: *CREATE TABLE AS*
- COPY puede descifrar datos mientras se cargan desde S3
- Se admite la compresión Gzip, Izop y bzip2 para acelerarlo aún más
- Opción de compresión automática
  - Analiza los datos que se están cargando y determina el esquema de compresión óptimo para almacenarlos
- Caso especial: tablas con muchas filas, pocas columnas
  - Carga con una sola transacción COPY si es posible
  - De lo contrario, las columnas de metadatos ocultas consumen demasiado espacio



# Concesiones de copia de Redshift para copias de instantáneas entre regiones (COPY GRANT)

- Supongamos que tienes un clúster de Redshift cifrado con KMS y una instantánea del mismo
- Quieres copiar esa instantánea a otra región para realizar una copia de seguridad
- En la región de destino de AWS:
  - Crea una clave KMS si aún no tienes una
  - Especifica un nombre único para tu concesión de copia de instantáneas
  - Especifica el ID de la clave KMS para la cual estás creando la concesión de copia
- En la región de origen de AWS:
  - Habilita la copia de instantáneas a la concesión de copia que acabas de crear



# DBLINK

- Conectar Redshift a PostgreSQL (posiblemente en RDS)
- Buena forma de copiar y sincronizar datos entre PostgreSQL y Redshift

```
CREATE EXTENSION postgres_fdw;
```

```
CREATE EXTENSION dblink;
```

```
CREATE SERVER foreign_server
```

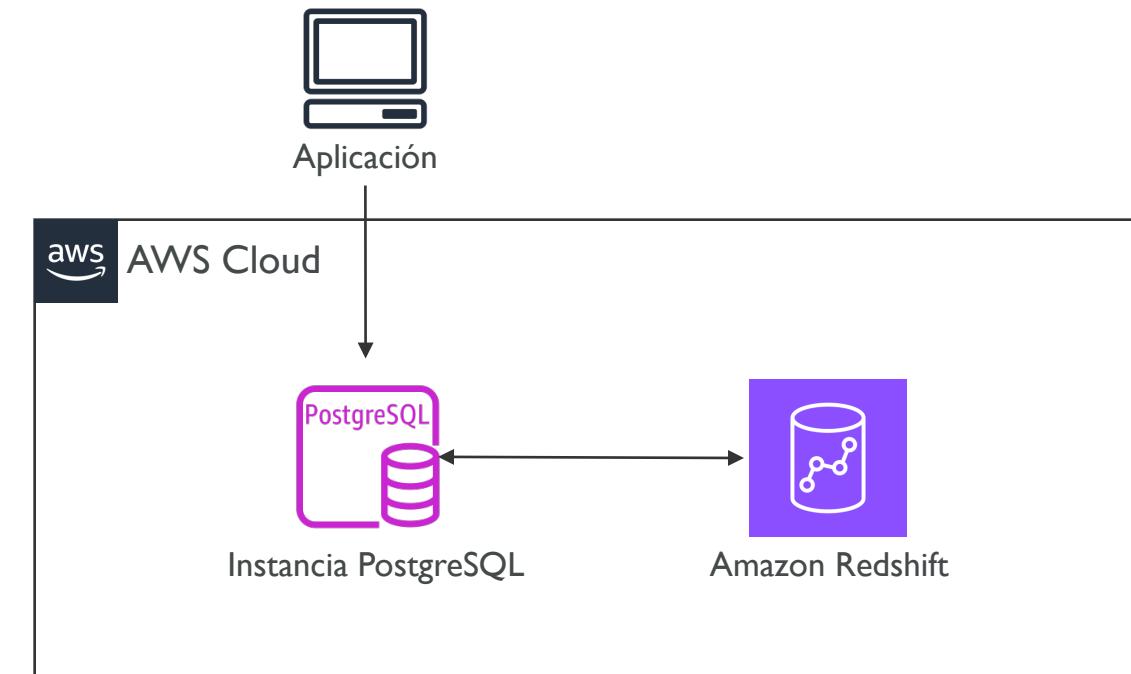
```
    FOREIGN DATA WRAPPER postgres_fdw
```

```
    OPTIONS (host '<amazon_redshift_ip>', port '<puerto>', dbname '<nombre_base_de_datos>', sslmode 'require');
```

```
CREATE USER MAPPING FOR <nombre_usuario_postgresql_rds>
```

```
    SERVER foreign_server
```

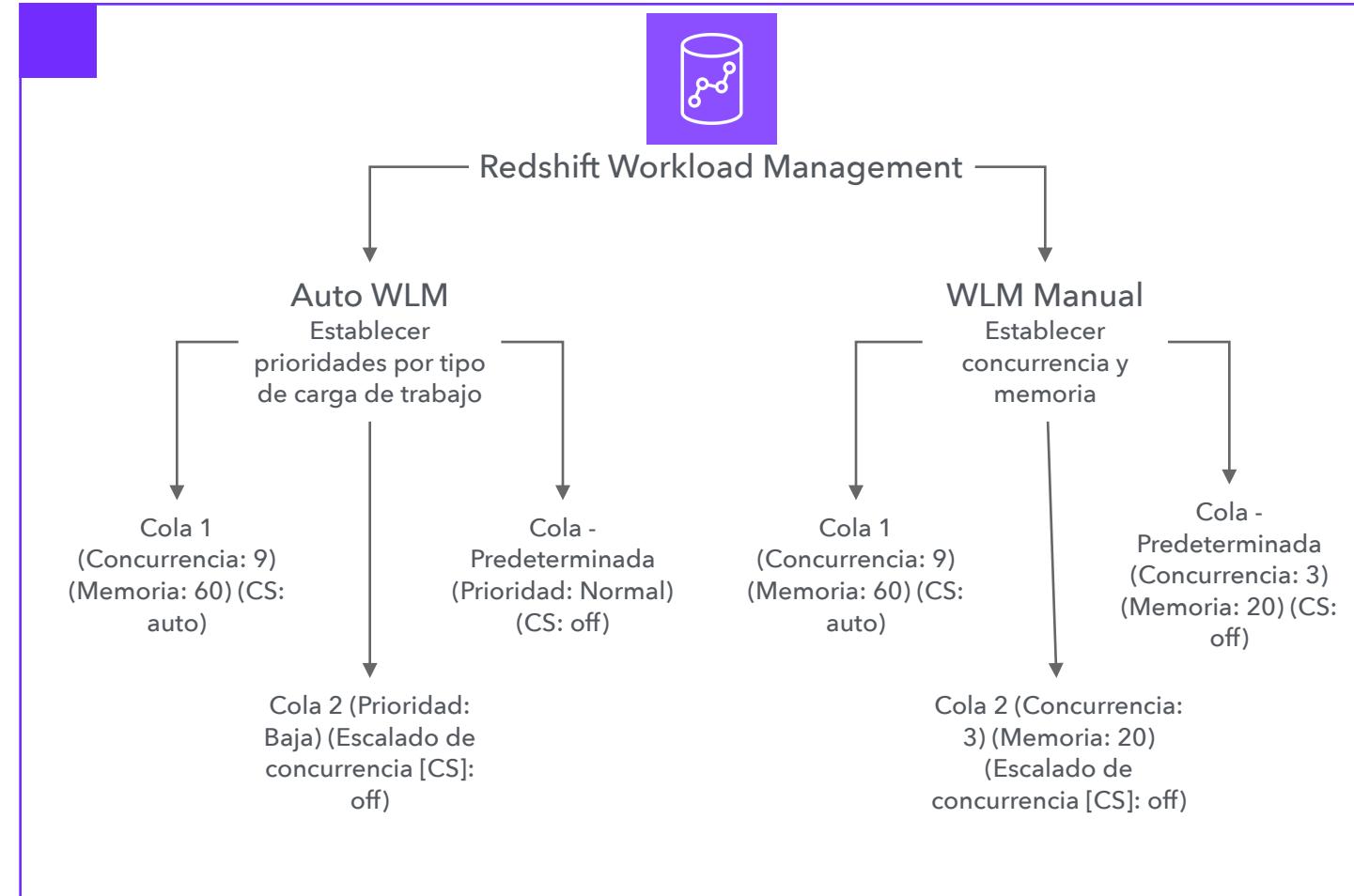
```
    OPTIONS (user '<nombre_usuario_redshift_amazon>', password '<contraseña>');
```



# Redshift Workload Management (WLM) / Administración de cargas de trabajo (WLM)

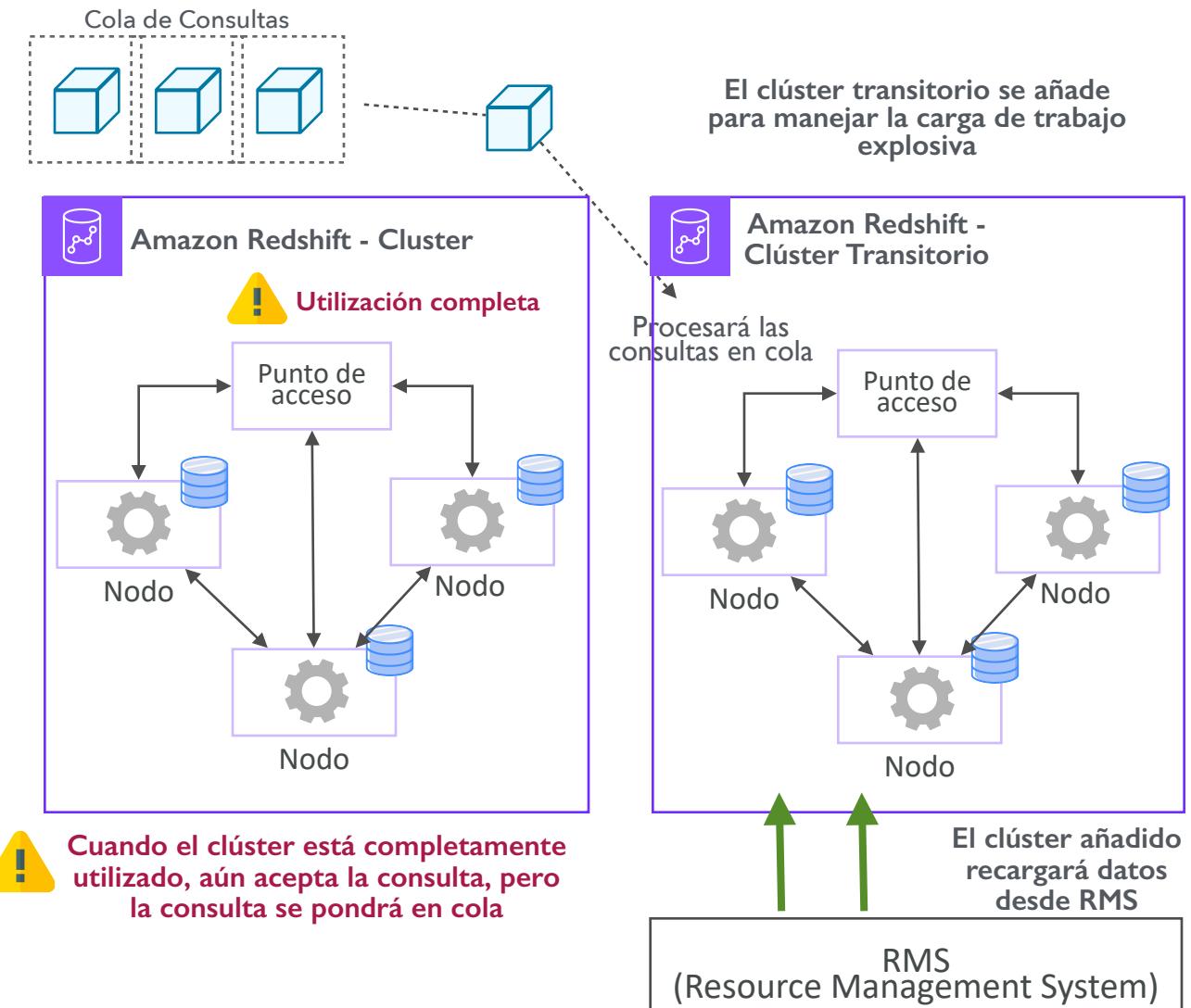
- Priorizar consultas cortas y rápidas frente a consultas largas y lentas
- Colas de consultas
- A través de la consola, CLI o API

**En Auto WLM, la concurrencia y la memoria son gestionadas por Amazon Redshift**



# Escalado de concurrencia

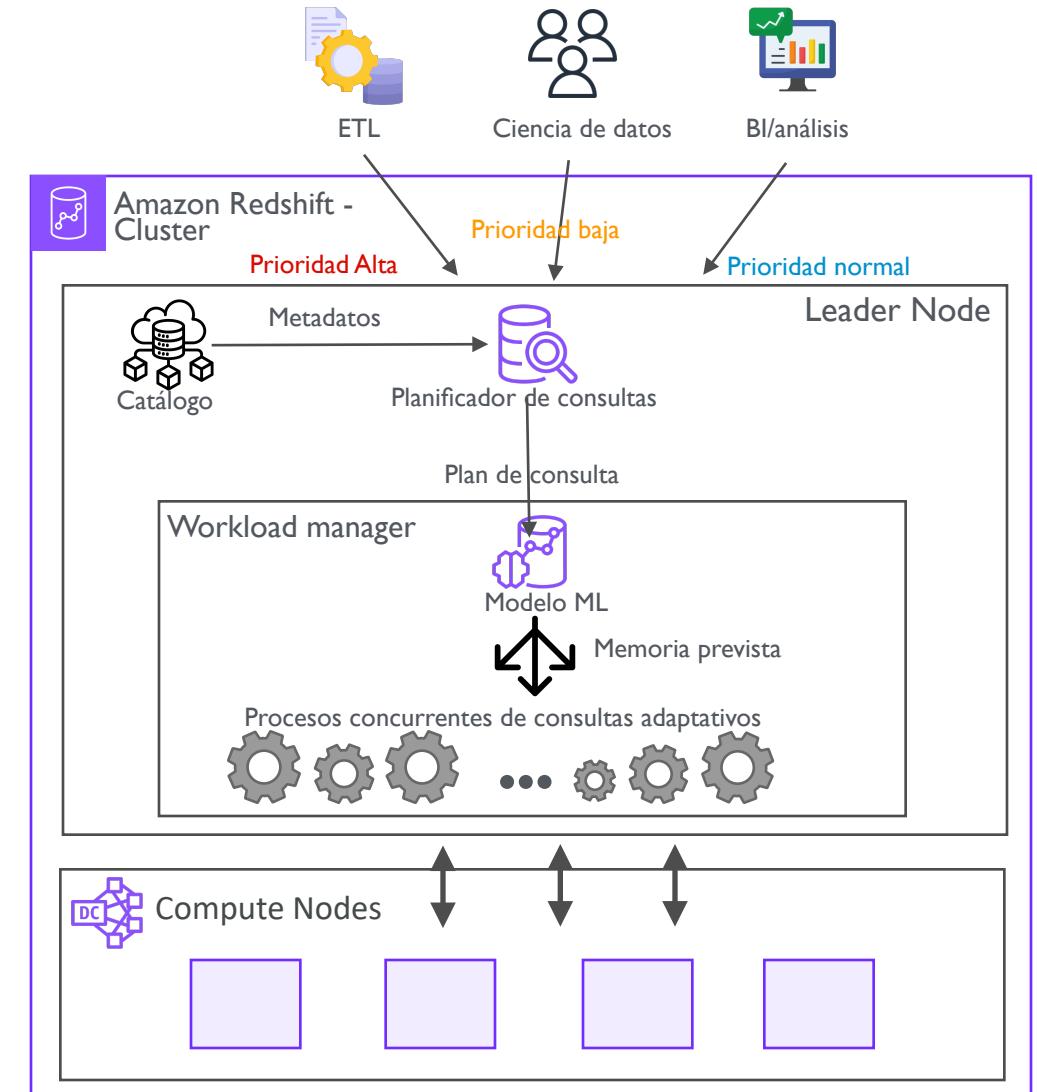
- Añade automáticamente capacidad al clúster para manejar el aumento de consultas de lectura concurrentes
- Soporta prácticamente usuarios y consultas concurrentes ilimitadas



# AutoWLM

## Administración automática de cargas de trabajo (WLM)

- El **AutoWLM** de Redshift es responsable del control de admisión, la programación y la asignación de recursos
- Después de recibir la consulta, AutoWLM convierte el plan de ejecución y las estadísticas optimizadas en un formato vectorial
- Redshift utiliza el resultado del modelo para colocar la consulta en la cola: basándose en el tiempo de ejecución predicho
- AutoWLM emplea un mecanismo de turno rotativo ponderado para programar consultas de mayor prioridad más frecuentemente que las de baja prioridad



# Manual WLM

## Administración manual de cargas de trabajo (WLM)

- Con WLM manual, puedes administrar el rendimiento del sistema y la experiencia de los usuarios mediante la modificación de la configuración de la WLM para crear colas independientes para las consultas de ejecución prolongada y las de ejecución corta
- Amazon Redshift configura las siguientes colas de consultas:
  - **Una cola de superusuario:** La cola de superusuario está reservada únicamente para superusuarios y no se puede configurar
  - **Una cola de usuario predeterminada :** La cola predeterminada está configurada inicialmente para ejecutar cinco consultas de forma simultánea

# Manual WLM

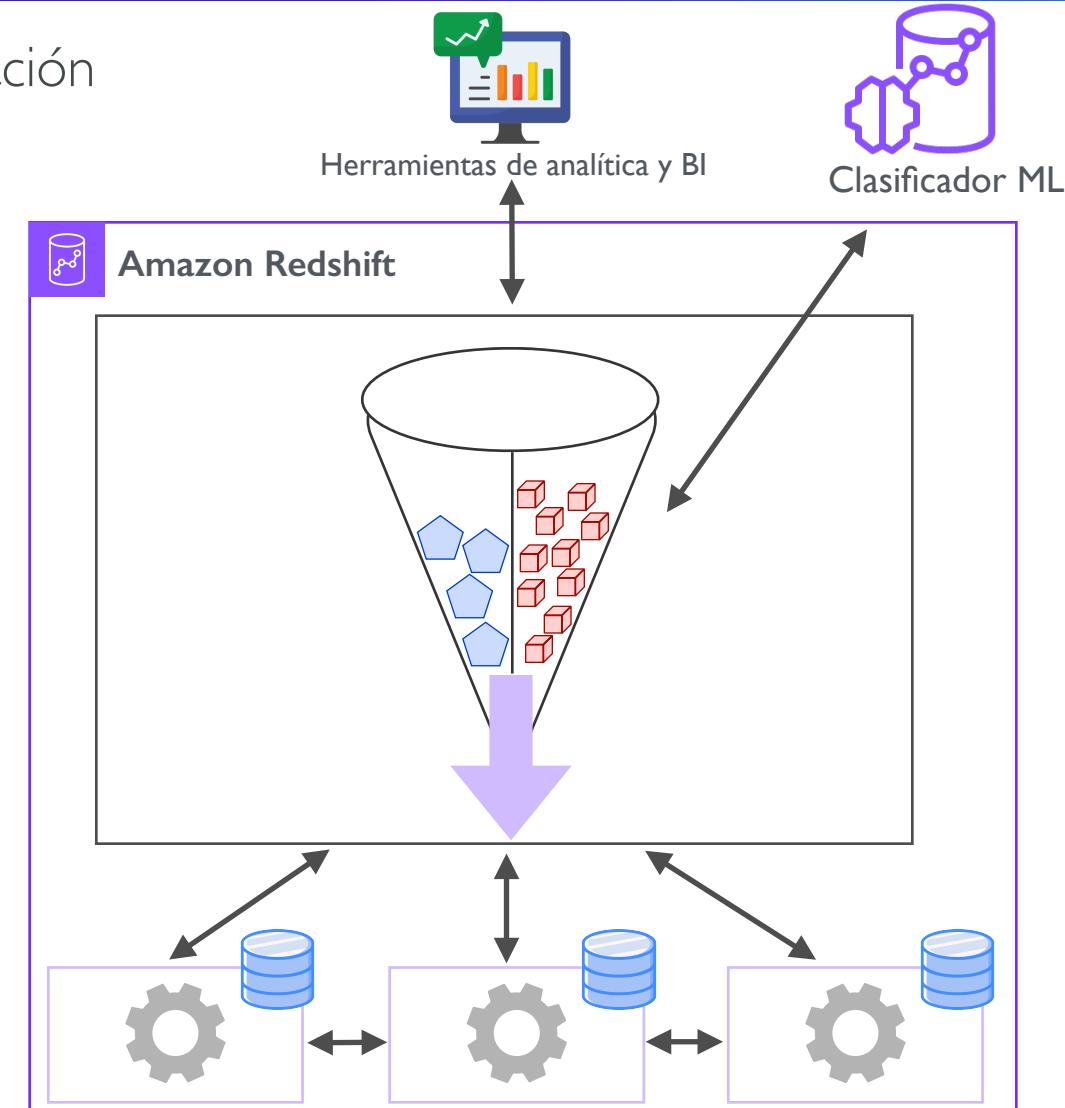
## Administración manual de cargas de trabajo (WLM)

- Puedes añadir colas de consultas adicionales a la configuración de WLM predeterminada, hasta 8 colas y hasta un nivel de concurrencia de 50
- Para cada cola de consultas, puedes configurar lo siguiente:
  - Modo de escalado de simultaneidad
  - Nivel de simultaneidad
  - Grupos de usuarios
  - Grupos de consultas
  - Porcentaje de memoria de WLM por utilizar
  - Tiempo de espera de WLM
  - Salto de cola de consultas de WLM
  - Reglas de monitorización de consultas

# Short Query Acceleration (SQA)

## Aceleración de Consultas Cortas (SQA)

- Prioriza consultas de corta duración sobre las de larga duración
- Las consultas cortas se ejecutan en un espacio dedicado, no esperan en la cola detrás de consultas largas
- Se puede usar en lugar de las colas de WLM para consultas cortas
- Funciona con:
  - **CREATE TABLE AS (CTAS)**
  - **Consultas de solo lectura (instrucciones SELECT)**
- Utiliza un algoritmo de machine learning para analizar cada una de las consultas que reúnan los requisitos necesarios y predecir su tiempo de ejecución
- WLM asigna dinámicamente un valor para el tiempo de ejecución máximo de SQA en función del análisis de la carga de trabajo del clúster
- Como alternativa, puede especificar un valor fijo comprendido entre 1 y 20 segundos



# Comando VACUUM

- Reordena las filas y recupera espacio en una tabla especificada o en todas las tablas de la base de datos actual
- Parámetros destacables de VACUUM:
  - **VACUUM FULL**
    - Combina la reclamación de espacio y la reordenación de los datos
  - **VACUUM DELETE ONLY**
    - Se enfoca únicamente en recuperar espacio eliminando filas que ya no son necesarias
  - **VACUUM SORT ONLY**
    - Realiza solo la reordenación de los datos según las claves de ordenación y no intenta recuperar espacio
  - **VACUUM REINDEX**
    - Realiza primero una re-análisis de la distribución de las columnas de clave de ordenación para identificar si hay desajustes o ineficiencias
    - Luego realiza un VACUUM completo



# Redimensionamiento de clústeres de Redshift

## • Redimensionamiento elástico

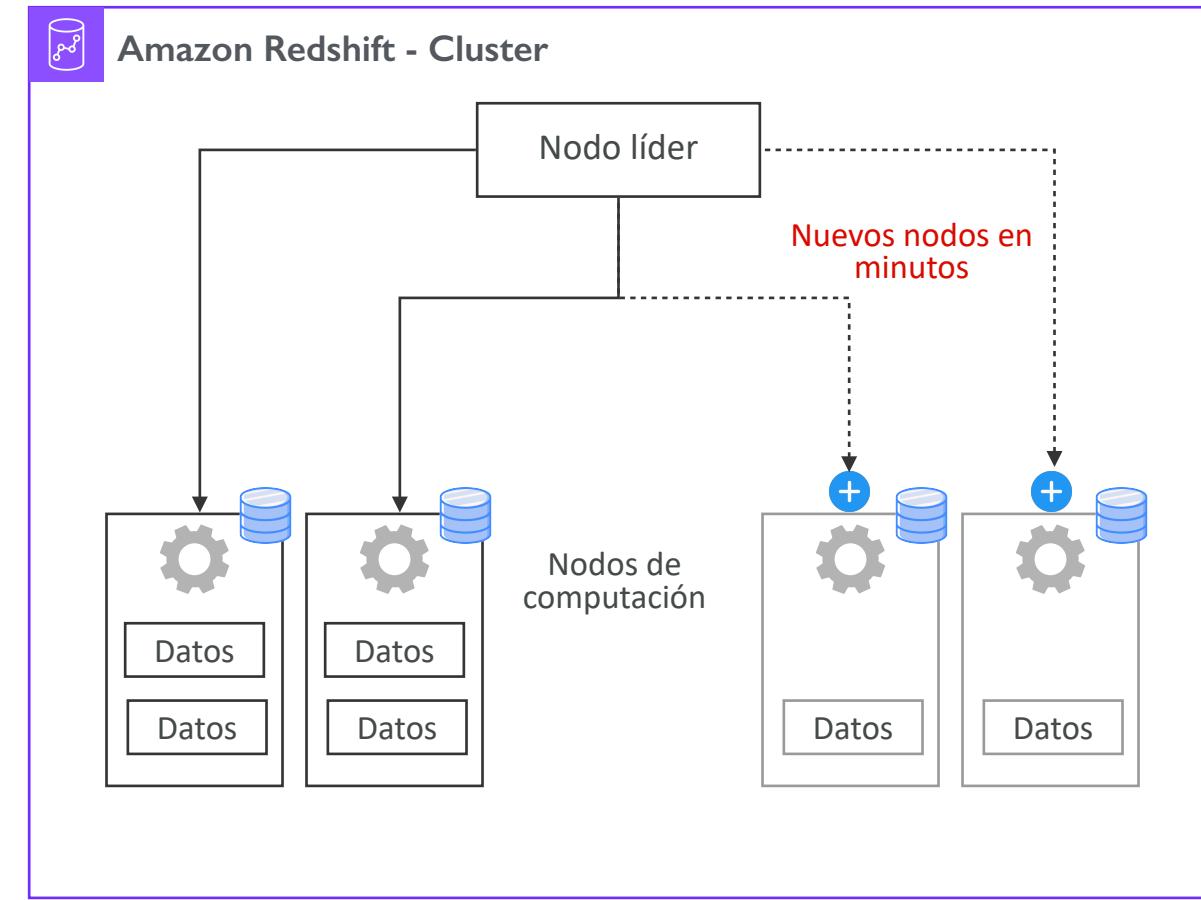
- Agregar o eliminar nodos del mismo tipo rápidamente
- El clúster está inactivo por unos minutos
- Intenta mantener las conexiones abiertas durante el tiempo de inactividad
- Limitado a duplicar o reducir a la mitad para algunos tipos de nodos dc2 y ra3.

## • Redimensionamiento clásico

- Cambiar el tipo de nodo y/o el número de nodos
- El clúster es de solo lectura durante horas o días

## • Instantánea, restauración, redimensionamiento

- Usado para mantener el clúster disponible durante un redimensionamiento clásico (Copiar clúster, redimensionar nuevo clúster)



# Novedades de Redshift

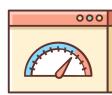
- **Nodos RA3 con almacenamiento gestionado**

- Permiten escalado independiente de computación y almacenamiento
- Basado en SSD



- **Exportación de data lake de Redshift**

- Descarga consultas de Redshift a S3 en formato Apache Parquet
- Parquet es 2 veces más rápido para descargar y consume hasta 6 veces menos almacenamiento
- Compatible con Redshift Spectrum, Athena, EMR, SageMaker
- Particionado automáticamente



- **Tipos de datos espaciales**

- GEOMETRY, GEOGRAPHY

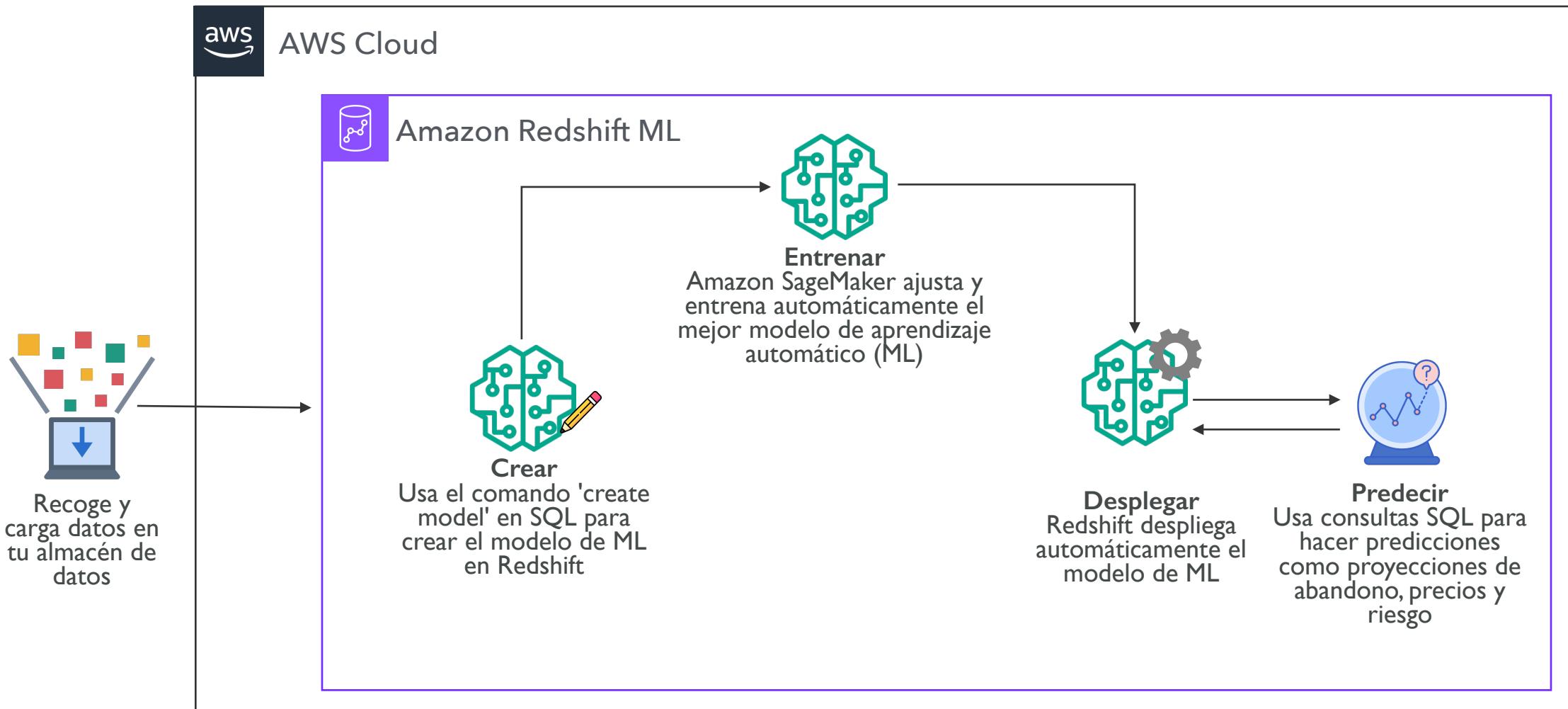


- **Compartición de datos entre regiones**

- Compartir datos en vivo entre clústeres de Redshift sin copiar
- Requiere el nuevo tipo de nodo RA3
- Seguro, entre regiones y entre cuentas



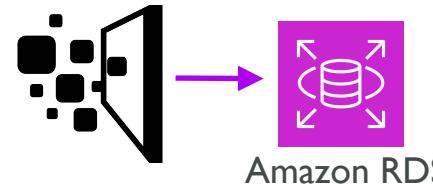
# Amazon Redshift ML



# Anti-patrones de Redshift

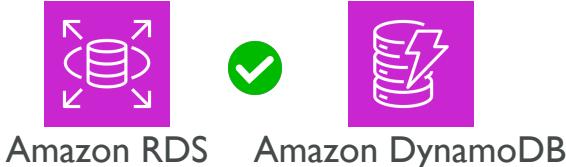
- **Conjuntos de datos pequeños**

Usar RDS en su lugar



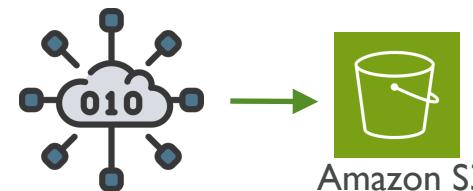
- **OLTP**

Usar RDS o DynamoDB en su lugar



- **Datos BLOB**

Almacenar referencias a archivos binarios grandes en S3, no los archivos en sí mismos.



- **Datos no estructurados**

Primero realizar ETL con EMR, etc.



# Redshift Serverless

- **Escalado automático y aprovisionamiento para la carga de trabajo**
- Optimiza costes y el rendimiento
  - Pagas solo cuando está en uso
- Utiliza ML para mantener el rendimiento a través de cargas de trabajo variables y esporádicas
- Fácil configuración de entornos de desarrollo y prueba
- Obtienes un punto final sin servidor, conexión JDBC/ODBC, o simplemente consultas a través del editor de consultas de la consola

Amazon Redshift sin servidor > Comience a utilizar Amazon Redshift sin servidor

## Comience a utilizar Amazon Redshift sin servidor Información

Para comenzar a utilizar Amazon Redshift sin servidor, configure un almacenamiento de datos sin servidor y cree una base de datos.

Recibirá un crédito de \$300 USD para su uso de Redshift sin servidor en esta cuenta.

### Configuración

**Usar la configuración predeterminada**  
La configuración predeterminada se ha definido para ayudarte a comenzar. Puedes modificarla en cualquier momento más adelante.

**Personalizar la configuración**  
Personalice la configuración según sus necesidades específicas.

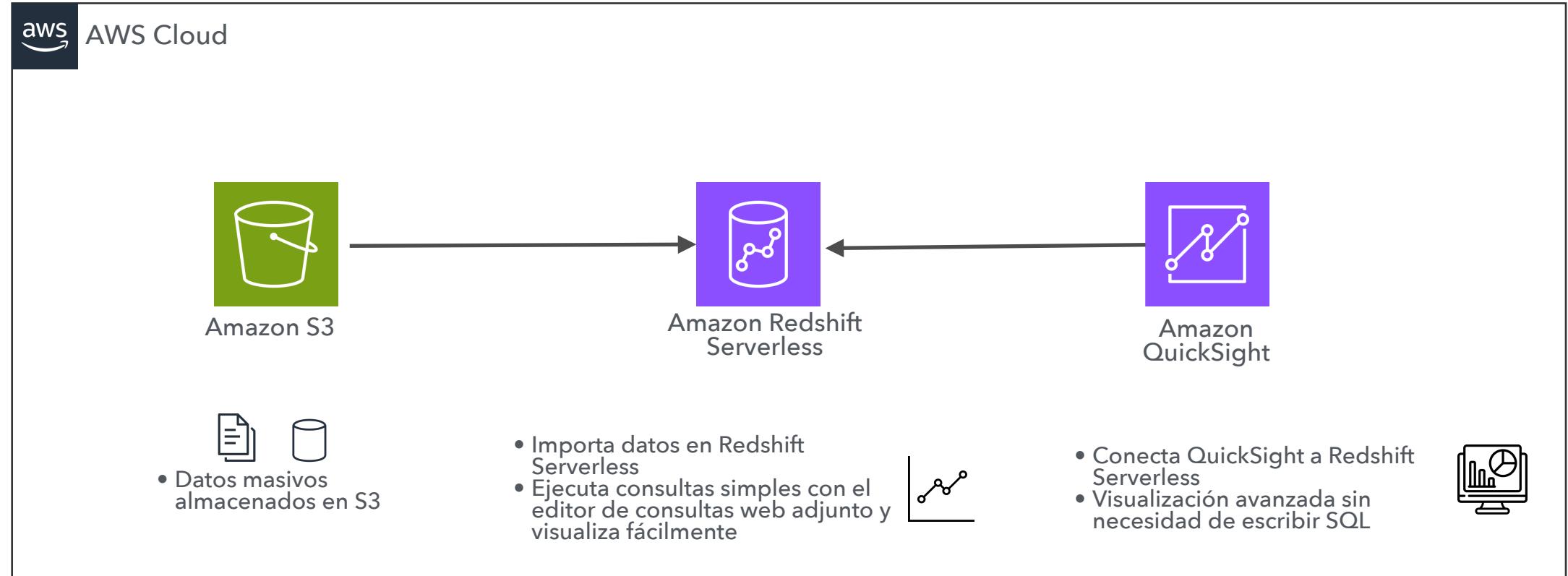
### Espacio de nombres Información

Un espacio de nombres es una colección de objetos de base de datos y usuarios. Las propiedades de datos incluyen el nombre y la contraseña de la base de datos, los permisos, el cifrado y la seguridad.

**⚠️** Los datos se cifran de forma predeterminada con una clave propiedad de AWS. Para elegir una clave diferente, elige **Personalizar configuración**.

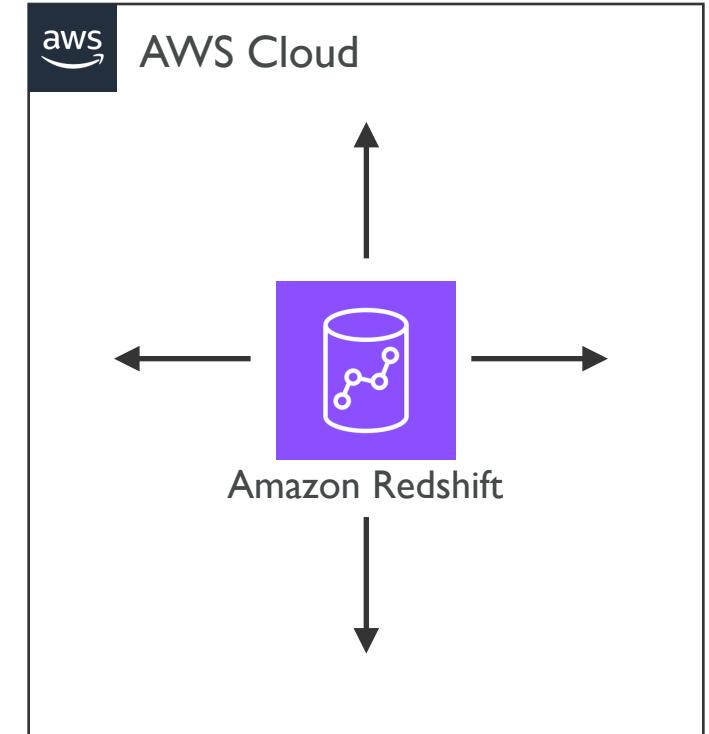
Espacio de nombres de destino  
**default-namespace**

# Redshift Serverless



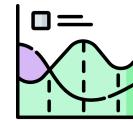
# Escalado de recursos en Redshift Serverless

- La capacidad se mide en **Unidades de Procesamiento de Redshift (RPU's)**
- Pagas por horas de RPU + almacenamiento
- **RPU's base**
  - Puedes ajustar la capacidad base
  - Por defecto está en AUTO
  - Pero puedes ajustar de 32 a 512 RPU's para mejorar el rendimiento de las consultas
- **RPU's máximas**
  - Puedes establecer un límite de uso para controlar costos
  - O, aumentarlo para mejorar el rendimiento



# Redshift Serverless

Hace todo lo que Redshift puede, **excepto:**



Redshift  
Spectrum



Grupos de  
parámetros



Gestión de cargas  
de trabajo



Integración con  
socios de AWS



Ventanas de  
mantenimiento / rutas  
de versiones

# Monitorización de Redshift Serverless

- **Vistas** de monitoreo

- SYS\_QUERY\_HISTORY
- SYS\_LOAD\_HISTORY
- SYS\_SERVERLESS\_USAGE
- ...y muchas más



- **Métricas** de CloudWatch

- QueriesCompletedPerSecond, QueryDuration, QueriesRunning, etc.
- Dimensiones: DatabaseName, latency (short/medium/long), QueryType, stage



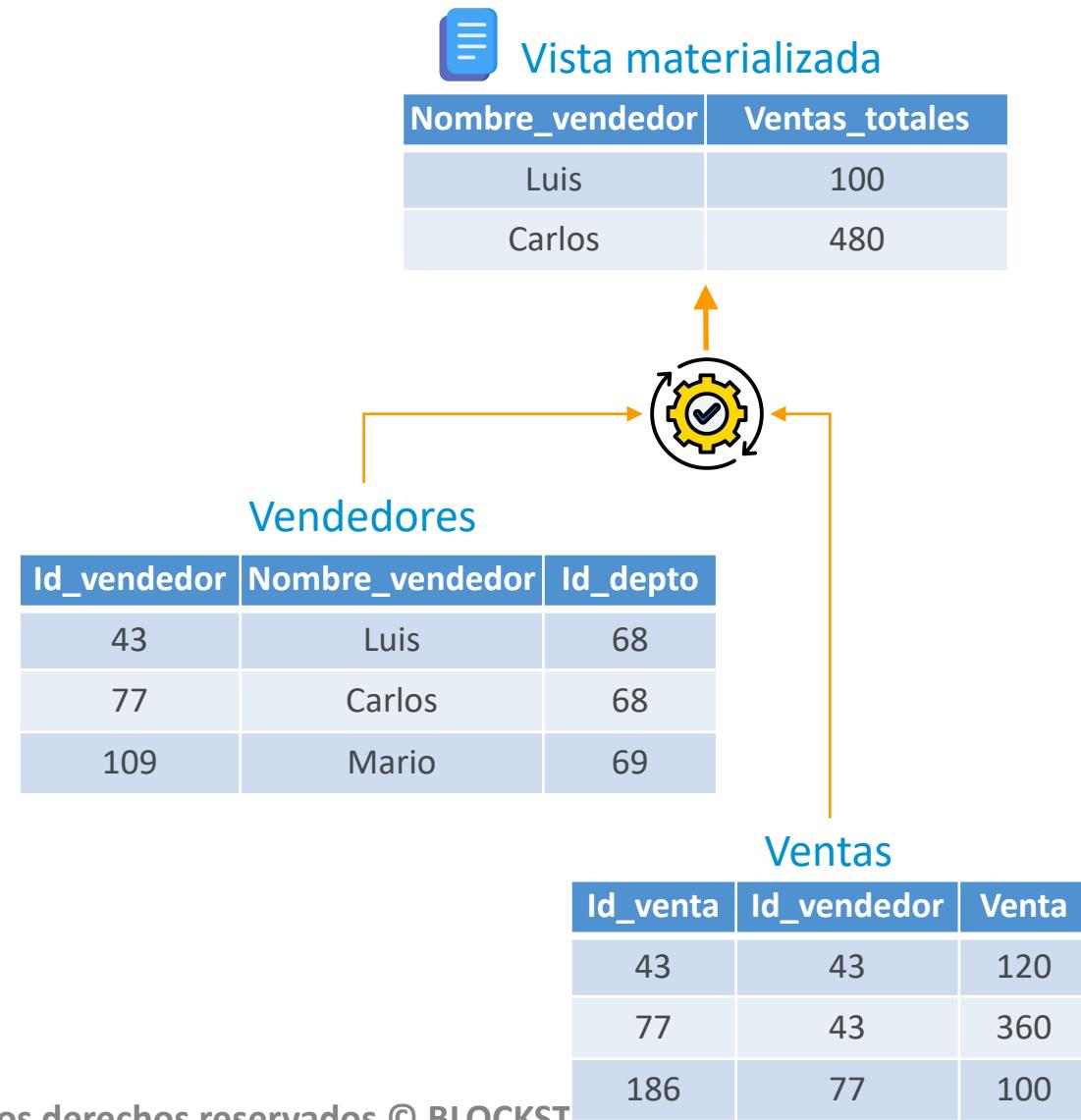
- **Logs** de CloudWatch

- Logs de conexión y de usuario habilitados por defecto
- Datos de logs de actividad de usuario opcionales
- Bajo /aws/redshift/serverless/



# Vistas materializadas de Redshift

- Contienen resultados precomputados basados en consultas SQL sobre una o más tablas base
- Proporcionan una forma de acelerar consultas complejas en un entorno de almacén de datos, especialmente en tablas grandes
- Puedes consultar vistas materializadas como cualquier otra tabla o vista
- Las consultas devuelven resultados más rápido ya que utilizan resultados precomputados sin acceder a las tablas base



# Compartir datos de Redshift

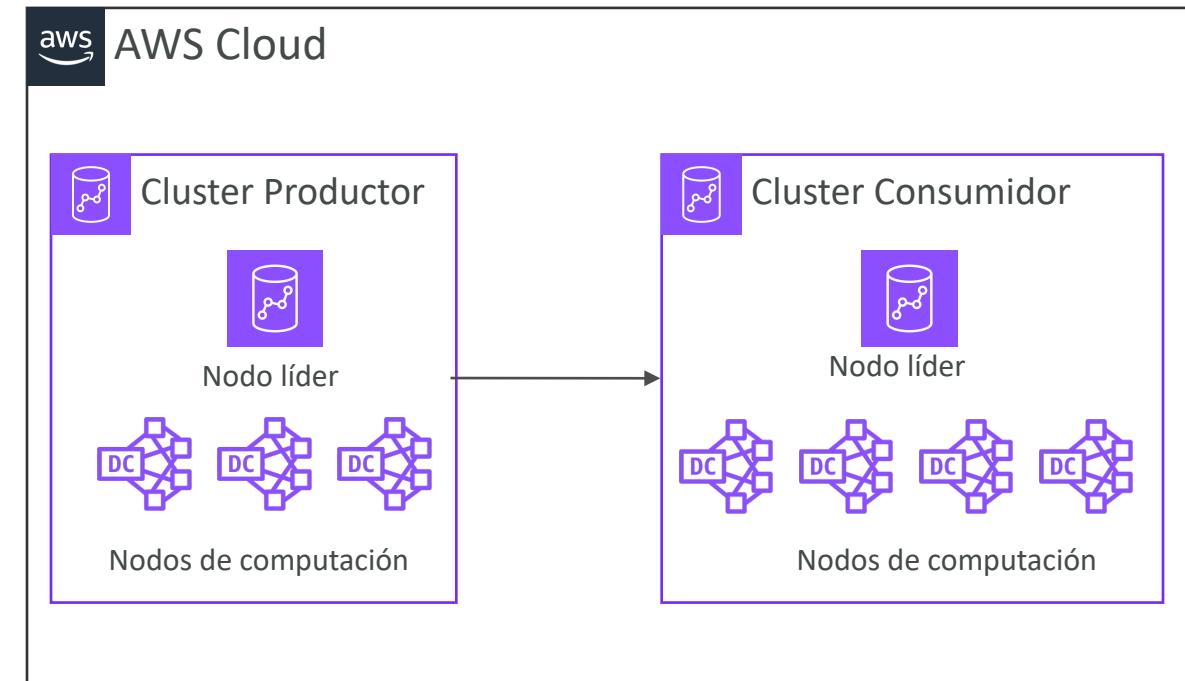
- Comparte datos de manera segura entre clústeres de Redshift con fines de lectura
- ¿Por qué?
  - Aislamiento de cargas de trabajo
  - Colaboración entre grupos
  - Compartir datos entre desarrollo/pruebas/producción
- Se pueden compartir bases de datos, esquemas, tablas, vistas, etc.



# Compartir datos de Redshift

- **Arquitectura productor / consumidor**

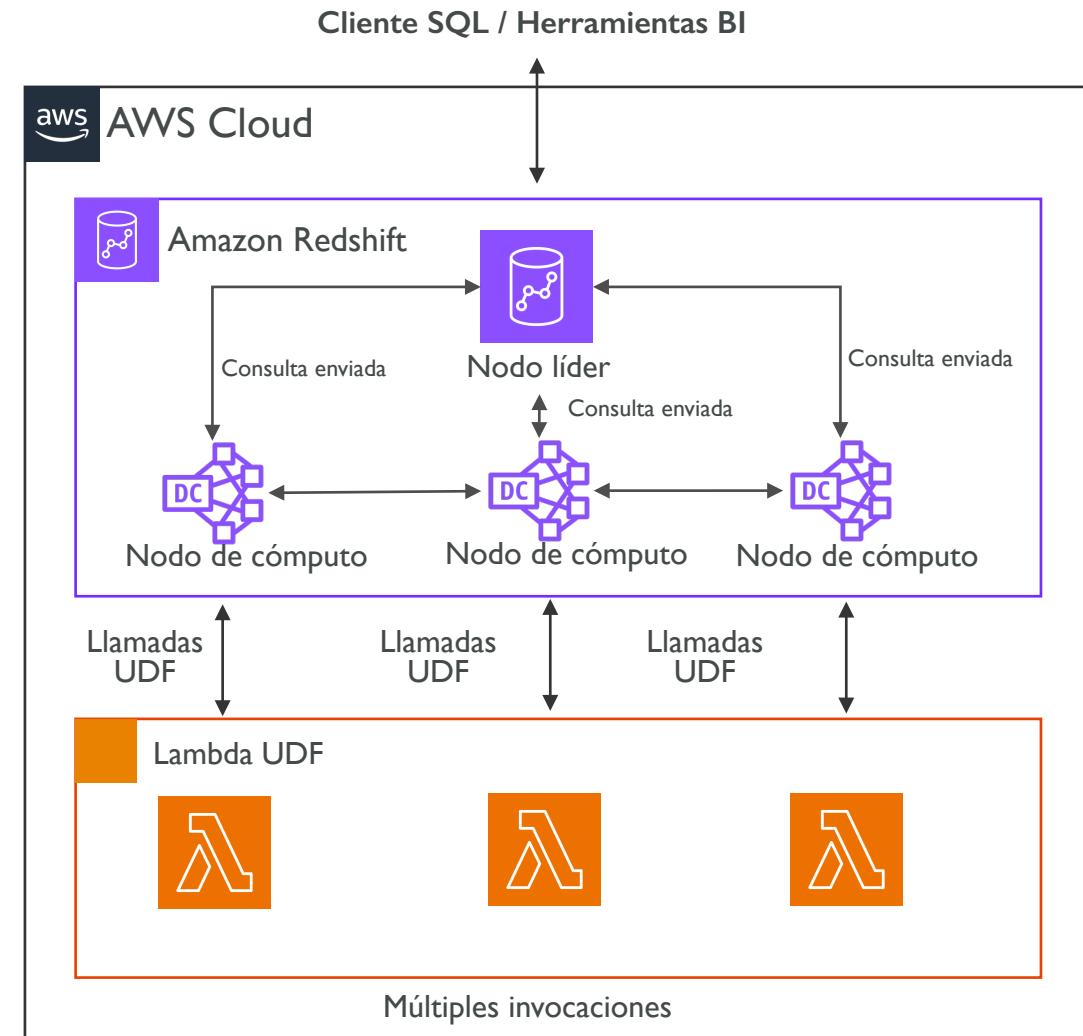
- El productor controla la seguridad
- Aislamiento para asegurar que el rendimiento del productor no sea afectado por los consumidores
- Los datos están en vivo y son transaccionalmente consistentes
- Ambos deben estar cifrados, deben usar nodos RA3
- Compartir datos entre regiones implica cargos de transferencia



# Redshift Lambda UDF

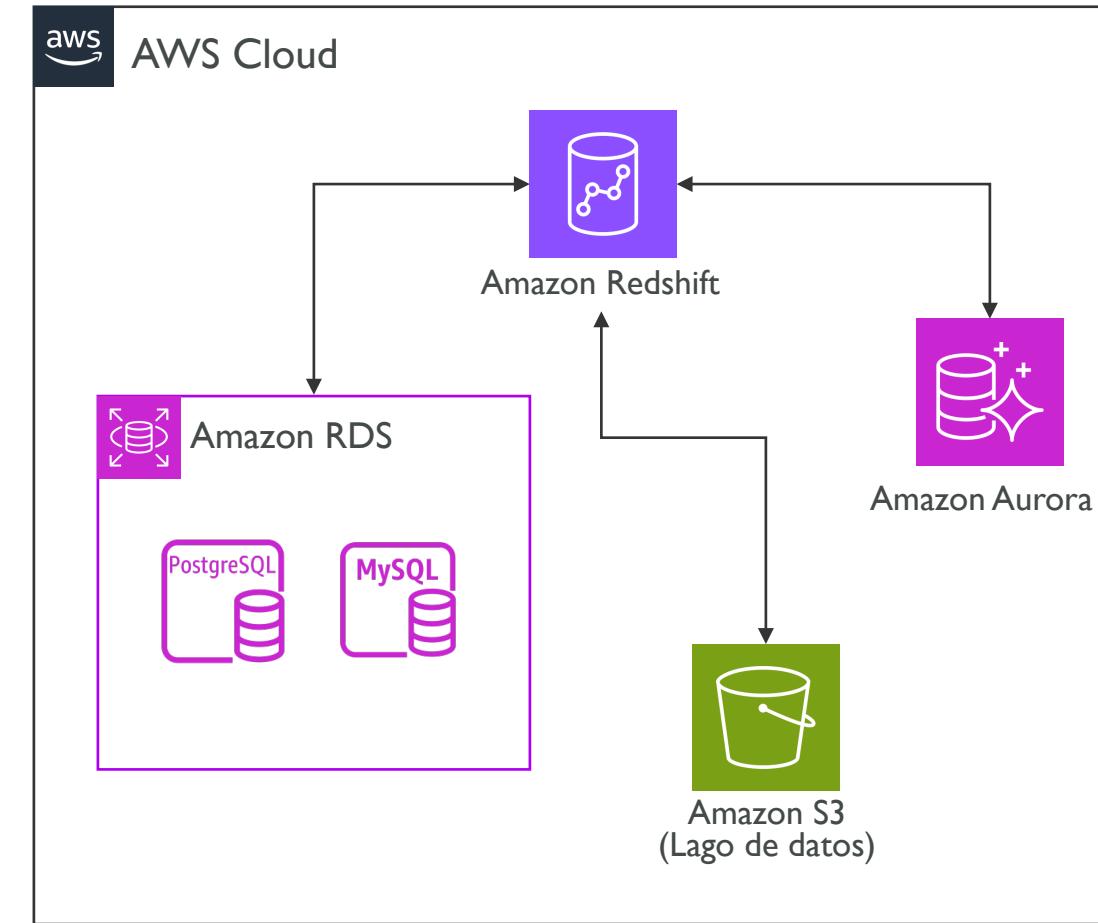
- Usa funciones personalizadas en AWS Lambda dentro de consultas SQL
  - ¡Usando cualquier lenguaje que desees!
  - Puedes hacer una variedad de acciones:
    - Llama a otros servicios (¿IA?)
    - Accede a sistemas externos
    - Integra con el servicio de ubicación

```
SELECT a, b FROM t1 WHERE lambda_multiply(a, b) = 32;  
  
CREATE EXTERNAL FUNCTION  
    custom_func(INT, INT)  
RETURNS INT VOLATILE LAMBDA  
    'lambda_sum'  
    IAM_ROLE  
    'arn:aws:iam::1234554321:role/Redshift-Sum-Test';
```



# Consultas federadas de Redshift

- Consulta y analiza a través de bases de datos, almacenes y Data Lakes
- Vincula Redshift a Amazon RDS o Aurora para PostgreSQL y MySQL
  - Incorpora datos en vivo en RDS a tus consultas de Redshift
  - Evita la necesidad de pipelines ETL
  - Puedes consultar RDS/Aurora desde Redshift, pero no al revés
- Desplaza la computación a bases de datos remotas para reducir el movimiento de datos
- Acceso de solo lectura a fuentes de datos externas
- Los costos serán incurridos en las bases de datos externas



# Tablas y vistas del sistema Redshift



- Contiene información sobre cómo está funcionando Redshift
- Tipos de tablas/vistas del sistema
  - **Vistas SYS:** Monitorean el uso de consultas y cargas de trabajo
  - **Tablas STV:** Datos transitorios que contienen instantáneas de los datos actuales
  - **Vistas SVV:** Metadatos sobre objetos de la BD que hacen referencia a las tablas STV
  - **Vistas STL:** Generadas a partir de registros persistidos en disco
  - **Vistas SVCS:** Detalles sobre consultas en clústeres principales y de escalado de concurrencia
  - **Vistas SVL:** Detalles sobre consultas en clústeres principales
- Muchas vistas y tablas de monitoreo del sistema son solo para clústeres aprovisionados, no para serverless

# Otras bases de datos

# Amazon DocumentDB



- Aurora es una "implementación de AWS" de PostgreSQL / MySQL ...
- **DocumentDB es lo mismo que MongoDB (que es una base de datos NoSQL)**

- MongoDB se utiliza para almacenar, consultar e indexar datos JSON
- “Conceptos de despliegue” similares a los de Aurora
- Totalmente gestionado, de alta disponibilidad con replicación a través de 3 AZ
- El almacenamiento de DocumentDB crece automáticamente en incrementos de 10 GB, hasta 128 TB
- Escala automáticamente a cargas de trabajo con millones de peticiones por segundo



# Amazon Neptune



- Base de datos **gráfica** totalmente gestionada
- Facilita la construcción y ejecución de aplicaciones que trabajan con conjuntos de **datos altamente conectados**
- Un **conjunto de datos de grafos** popular sería una **red social**
  - Los usuarios tienen amigos
  - Las publicaciones tienen comentarios
  - Los comentarios tienen likes de los usuarios
  - Los usuarios comparten y les gustan las publicaciones
- Puede almacenar hasta miles de millones de relaciones y consultar el gráfico con una latencia de milisegundos
- Alta disponibilidad con réplicas a través de múltiples AZs
- Casos de uso:
  - Aplicaciones de redes sociales
  - Recomendaciones de productos
  - Redes de conocimiento



# Amazon Keyspaces (para Apache Cassandra)



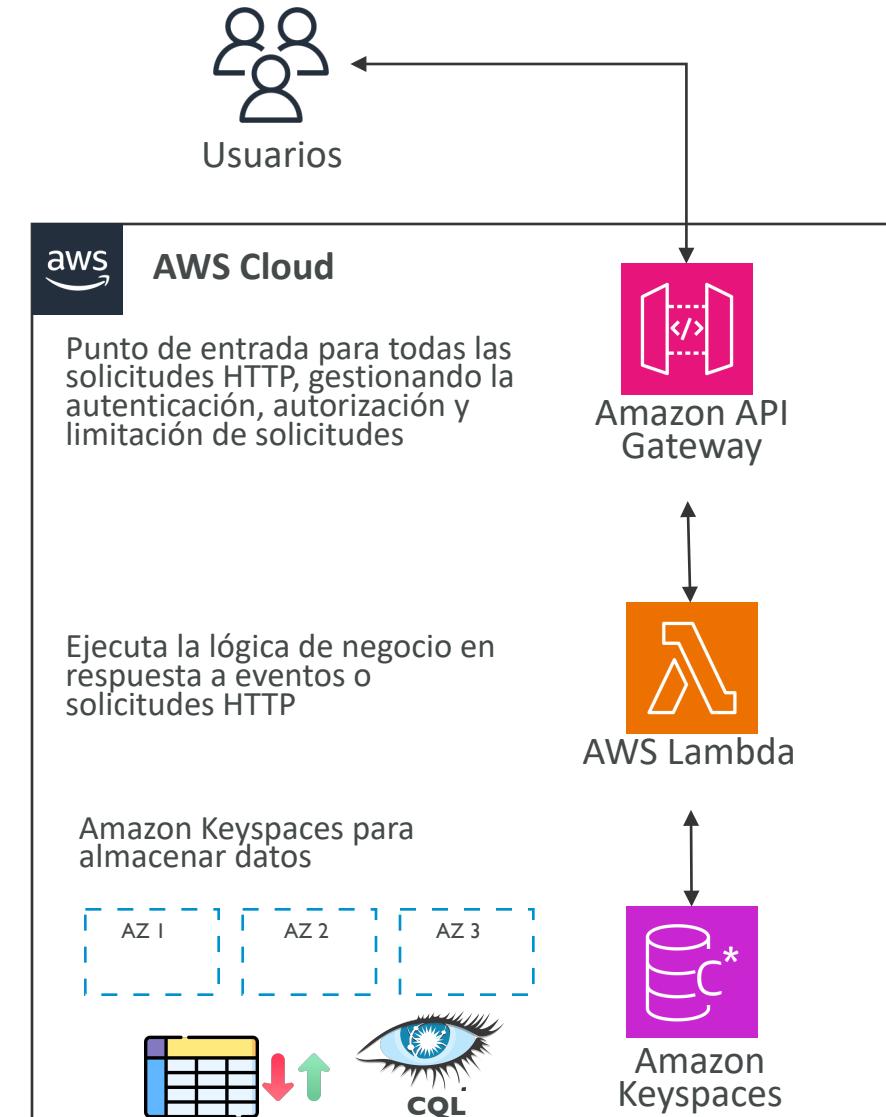
- Apache Cassandra = base de datos distribuida NoSQL de código abierto
- Amazon Keyspaces = servicio compatible con Apache Cassandra, que permite a los usuarios ejecutar sus aplicaciones Cassandra en AWS sin tener que gestionar la infraestructura subyacente
- Ofrece compatibilidad con CQL (Cassandra Query Language)
- Casos de uso:
  - Aplicaciones web a gran escala
  - Datos de series temporales
  - Sistemas de recomendación en tiempo real
  - Análisis de fraude
  - Juegos en línea



# Amazon Keyspaces (para Apache Cassandra)



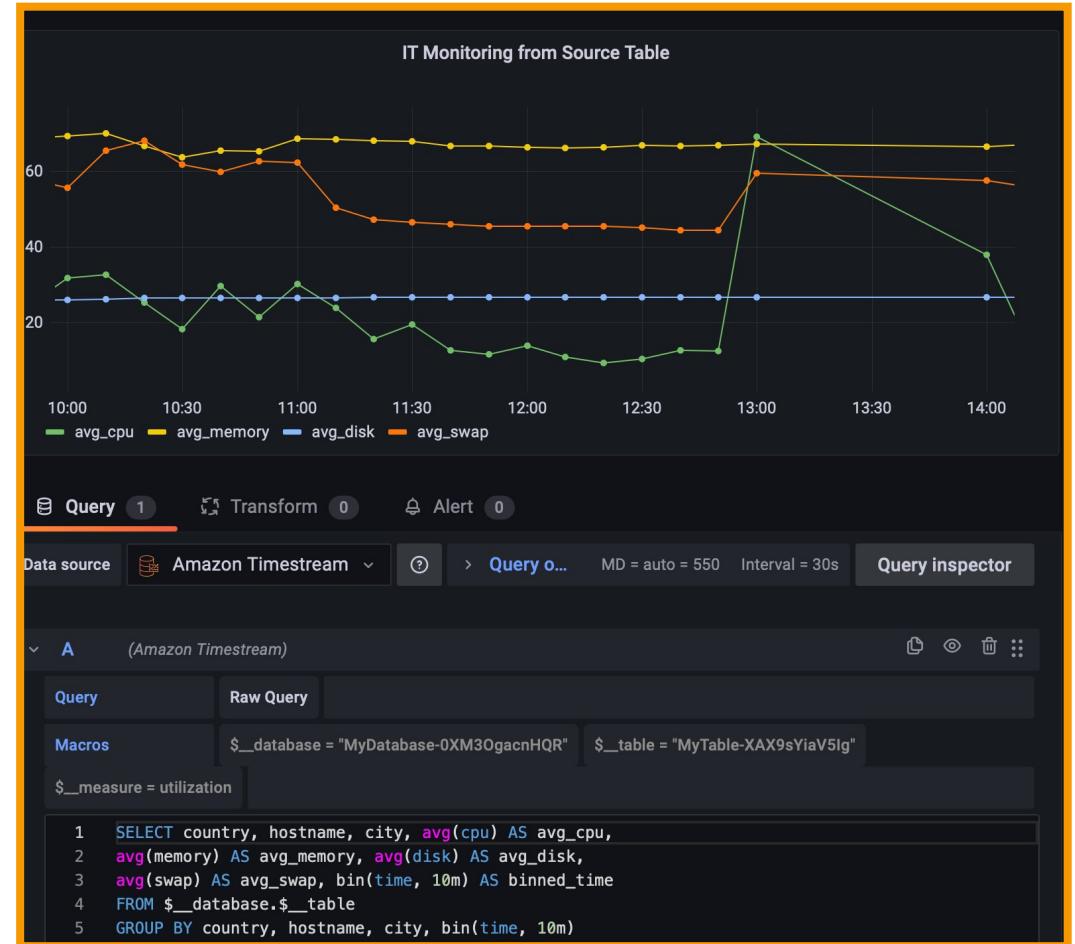
- Sin servidor, escalable, de alta disponibilidad, totalmente administrado
- Escala automáticamente las tablas hacia arriba/abajo en función del tráfico de la aplicación
- Las tablas se replican en múltiples AZ
- Latencia de un milisegundo a cualquier escala, miles de solicitudes por segundo
- Modo bajo demanda o modo provisionado con autoescalado
- Cifrado, copia de seguridad, recuperación puntual (PITR) de hasta 35 días



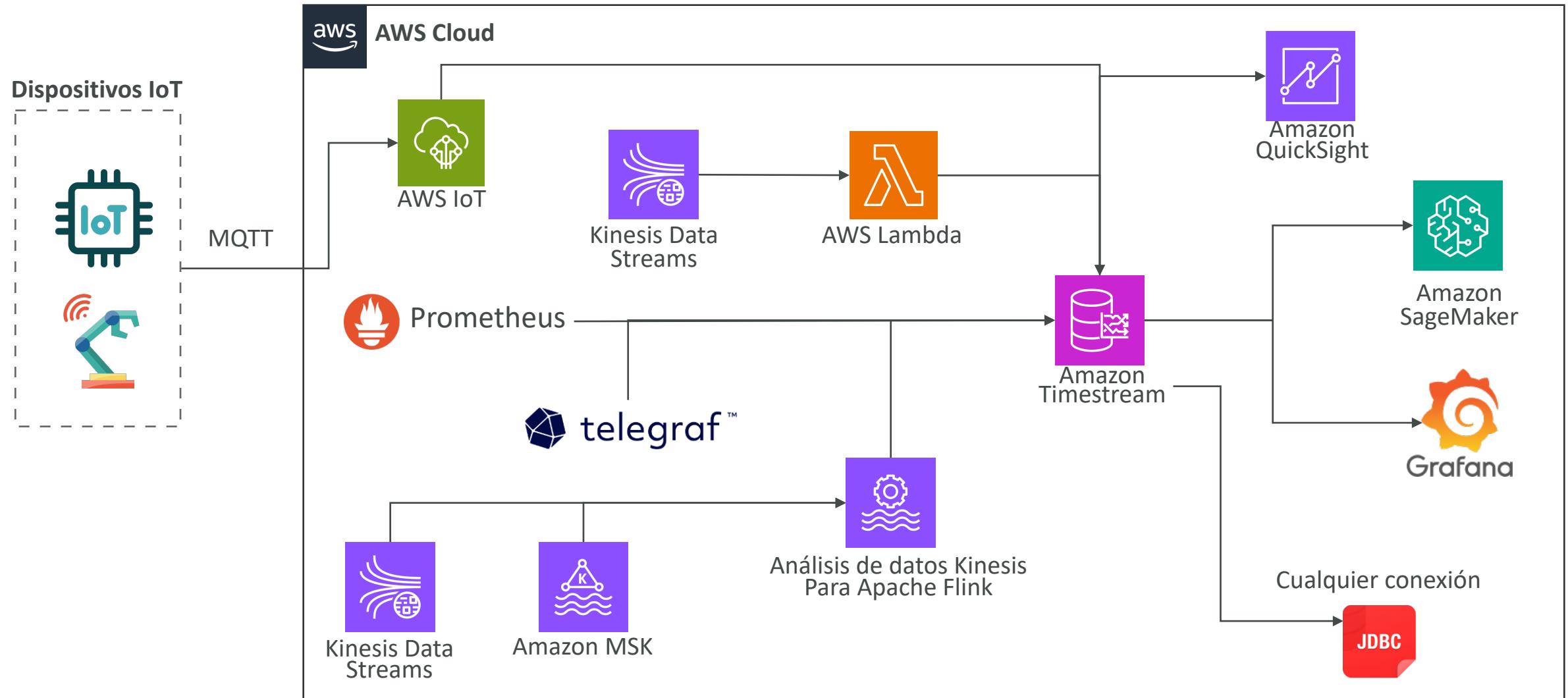
# Amazon Timestream



- Base de datos de **series temporales** totalmente gestionada, rápida, escalable y sin servidor (compatibilidad con SQL)
- Se amplía y reduce automáticamente para ajustar la capacidad
- Almacena y analiza billones de eventos al día
- 1000 veces más rápida y 1/10 del coste de las bases de datos relacionales
- Almacenamiento de datos por niveles: los datos recientes se guardan en la memoria y los históricos en un almacenamiento de coste optimizado
- Cifrado en tránsito y en reposo
- Casos de uso:
  - Aplicaciones IoT
  - Análisis en tiempo real



# Amazon Timestream – Arquitectura



# Amazon QLDB



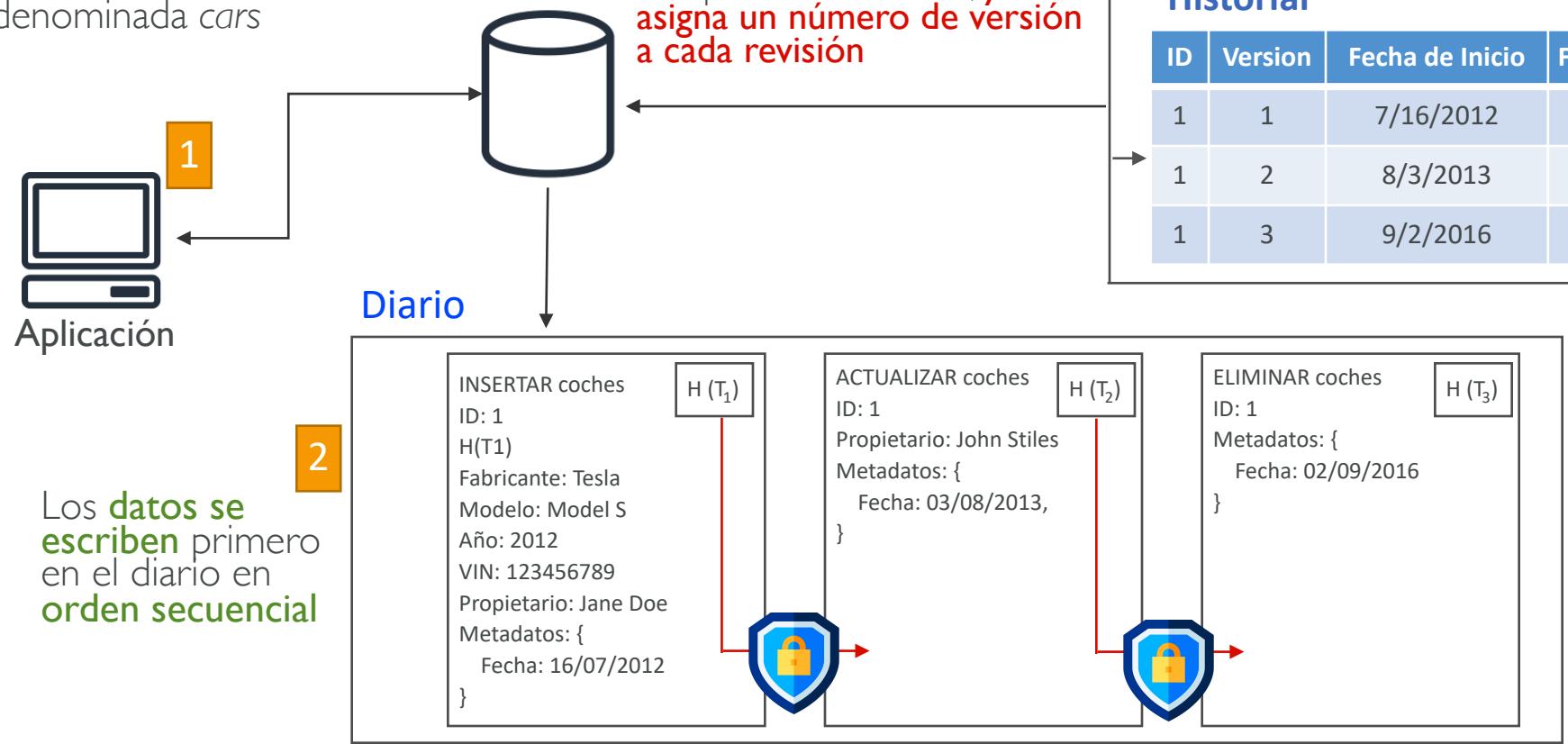
- QLDB significa "Quantum Ledger Database" (base de datos de libros contables)
- Historial secuenciado de todos los cambios de datos de aplicaciones
- Sistema **inmutable**: ninguna entrada puede ser eliminada o modificada, verificable criptográficamente
- Totalmente gestionada, sin servidor, de alta disponibilidad, con replicación en múltiples AZ
- Se utiliza para **revisar el historial de todos los cambios realizados en los datos de tu aplicación** a lo largo del tiempo
- Diferencia con Amazon Managed Blockchain: **No hay componente de descentralización en QLDB**



# Amazon QLDB



Una aplicación se conecta a un libro mayor y ejecuta transacciones que insertan, actualizan y eliminan un documento en una tabla denominada *cars*



Estado actual (antes de la eliminación)

| ID | Fabricante | Modelo   | Año  | VIN       | Propietario |
|----|------------|----------|------|-----------|-------------|
| 1  | Tesla      | Modelo S | 2012 | 123456789 | John Stiles |

## Historial

| ID | Version | Fecha de Inicio | Fabricante | Modelo   | Año  | VIN       | Propietario |
|----|---------|-----------------|------------|----------|------|-----------|-------------|
| 1  | 1       | 7/16/2012       | Tesla      | Modelo S | 2012 | 123456789 | Jane Doe    |
| 1  | 2       | 8/3/2013        | Tesla      | Modelo S | 2012 | 123456789 | John Stiles |
| 1  | 3       | 9/2/2016        |            |          |      |           | Eliminado   |

Procesador de Eventos (streaming)



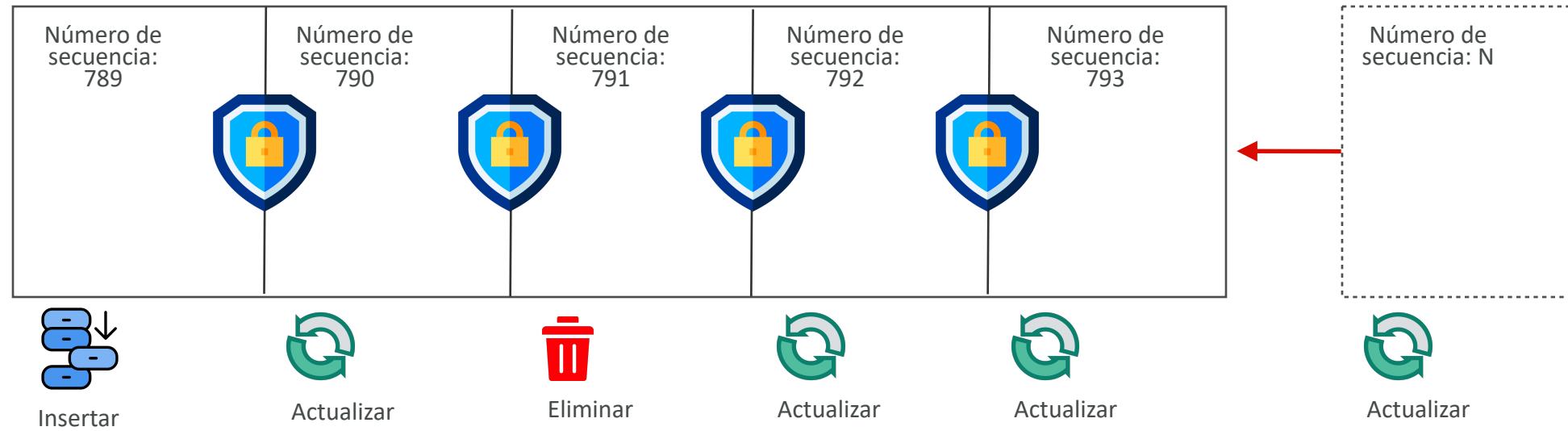
Se puede exportar o transmitir datos directamente desde QLDB

# Amazon QLDB



- QLDB es únicamente un anexo, mantiene un registro completo de todos los cambios en los datos que no se puede modificar ni sobrescribir
- No es posible alterar los datos confirmados mediante API ni ningún otro método

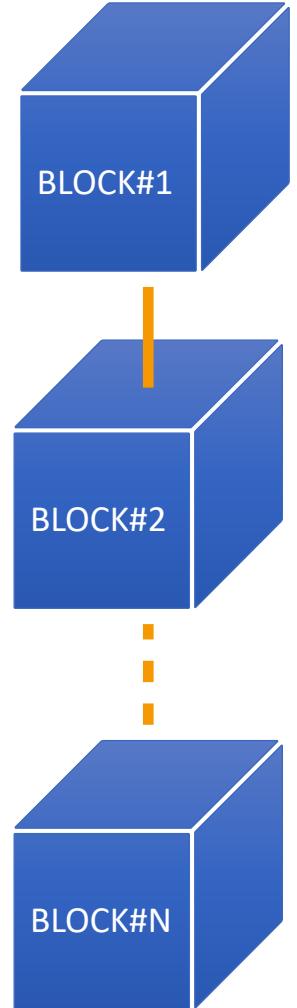
## Los registros no pueden alterarse



# Amazon Managed Blockchain



- Blockchain permite crear aplicaciones en las que varias partes pueden ejecutar transacciones **sin necesidad de una autoridad central de confianza**
- Facilita la creación y gestión de redes blockchain escalables utilizando tecnologías populares como **Hyperledger Fabric** y **Ethereum**
- Amazon Managed Blockchain es un servicio gestionado para:
  - Unirte a redes públicas de blockchain
  - O crear tu propia red privada escalable
- Ajusta automáticamente los recursos necesarios para satisfacer las demandas de la red, permitiendo un rendimiento óptimo sin intervención manual
- Permite la creación de redes blockchain en múltiples regiones de AWS





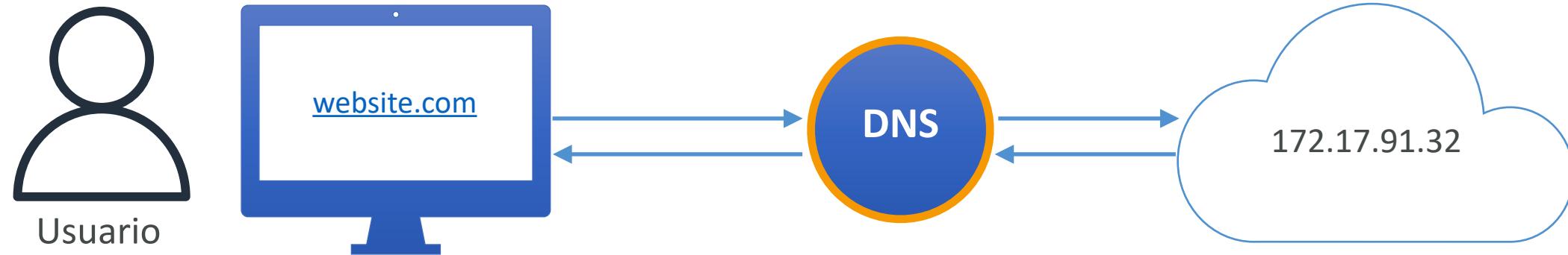
# Route 53

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Visión general del DNS

- El DNS, o sistema de nombres de dominio, traduce los nombres de dominios aptos para lectura humana.

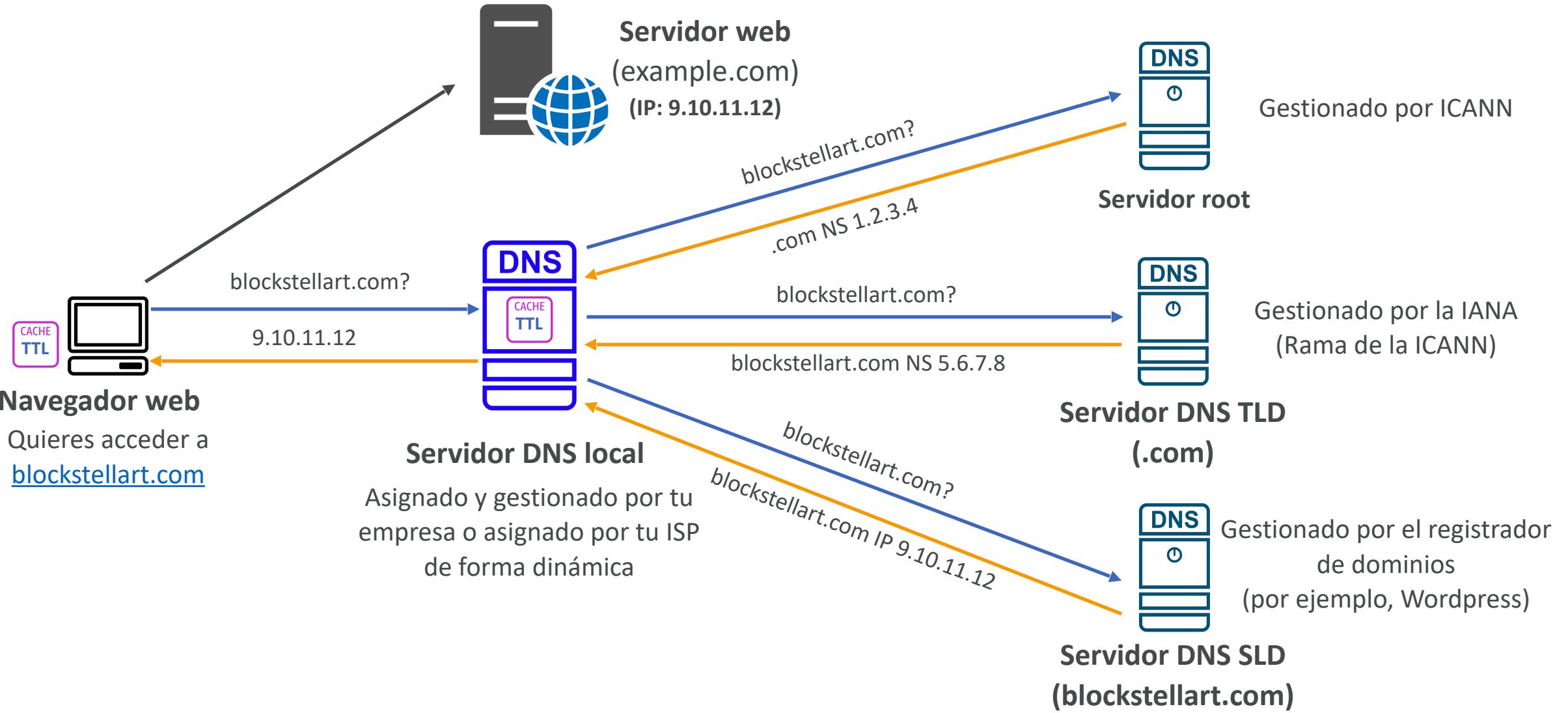


- Es una de las **tecnologías fundamentales en el funcionamiento de Internet**
- Es muy difícil memorizar una dirección como 172.217.3.78 (dirección IPv4 de Google)
- Se complica más, cuando los usuarios accedemos a varias webs y servicios. Y se vuelve todo un reto cuando las direcciones de esos servicios son dinámicas (cambian con el pasar de los días).

## SOLUCIÓN:

Tener una base de datos de nombres de dominios a los que le relaciona con una dirección IP

# ¿Cómo funciona el DNS?



# Visión general de Route 53

- **Una forma fiable y rentable de dirigir a los usuarios finales a las aplicaciones**
- Dirige a los usuarios finales a un sitio de forma confiable mediante servidores de sistema de nombres de dominio (DNS) distribuidos a nivel mundial y **con escalado automático**
- Posibilidad de crear DNS **público** y **privado**
- Route 53 también tiene la funcionalidad de **registrador de dominios** (ejemplo: [blockstellart.com](http://blockstellart.com))
- 53 es una referencia al puerto DNS tradicional



McDonald's administra el enrutamiento del tráfico global con Amazon Route 53

The Netflix logo, featuring the word "NETFLIX" in its signature red sans-serif font.

Netflix mejoró la resiliencia de las aplicaciones con Amazon Route 53

The Slack logo, which includes a colorful icon of overlapping rounded rectangles followed by the word "slack" in a black sans-serif font.

Slack mejoró la seguridad y el rendimiento de la API con Amazon Route 53

# Diagrama del uso de Amazon Route 53



# Zonas de alojamiento de Amazon Route 53

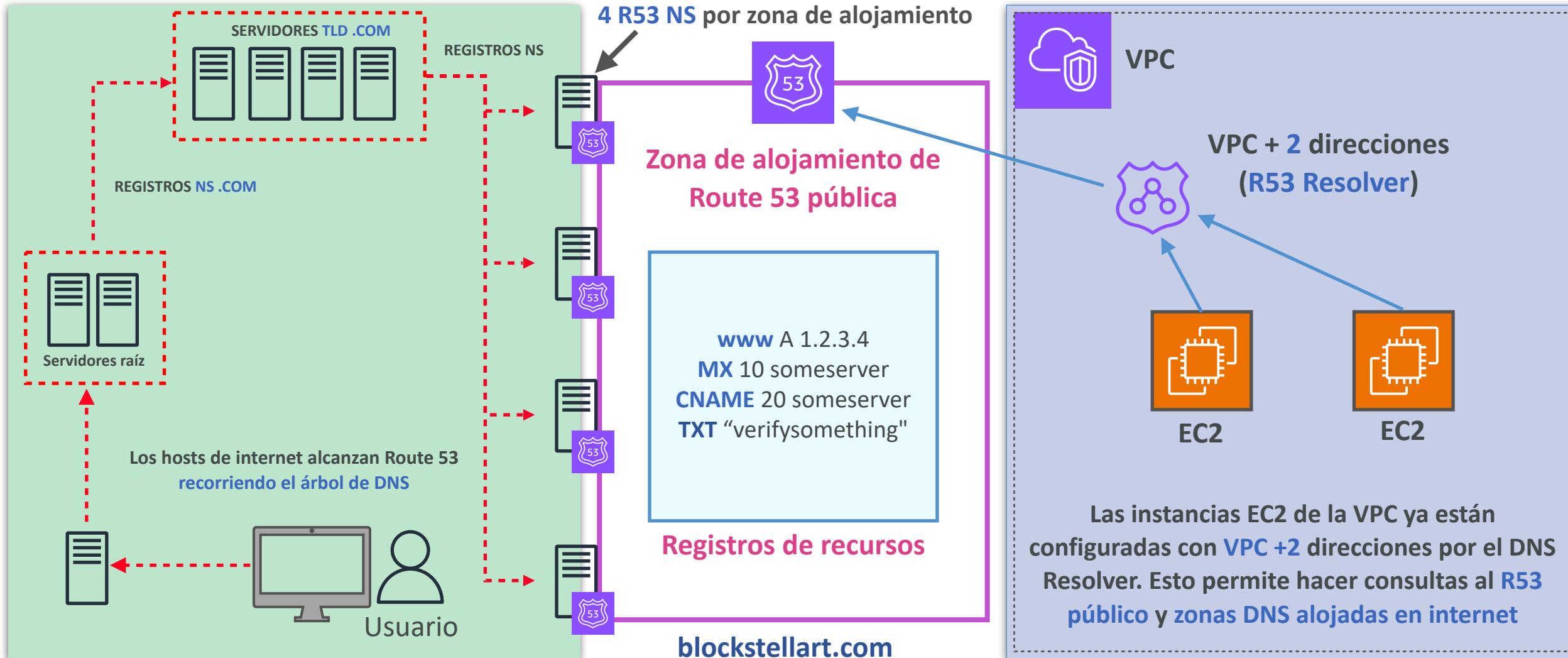
## Zona de alojamiento pública

- Accesibilidad garantizada desde cualquier punto de la red global de **internet** y **desde las VPCs de AWS**
- **Cuatro** servidores de nombres específicos de Route 53 (NS) proporcionan **alojamiento dedicado y exclusivo** para la zona de alojamiento
- Configuración mediante registros **NS** que direccionan hacia estos servidores de nombres
- Capacidad para que **dominios registrados fuera** de AWS dirijan el tráfico hacia los servicios alojados en la zona pública de Route 53

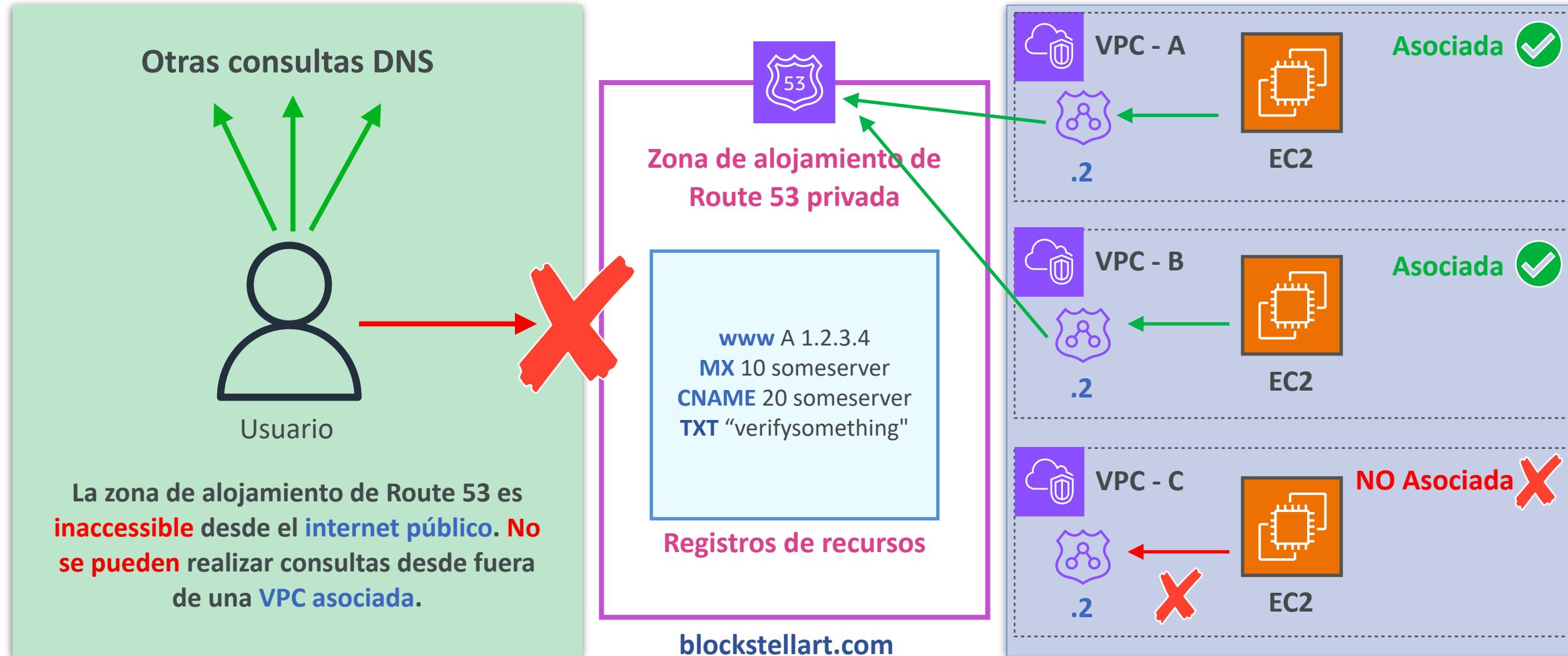
## Zona de alojamiento privada

- Una zona de alojamiento que **no es pública**
- Asociada con VPCs
- **Sólo accesible por las VPCs asociadas**
- Garantiza que el tráfico de DNS dentro de la VPC **permanezca privado** y no sea interceptado por entidades externas
- **Caso de uso:** Crear entornos de desarrollo y prueba con nombres de dominio que simulan el entorno de producción

# Zonas de alojamiento públicas de Route 53



# Zonas de alojamiento privadas de Route 53



# Tipos de registros de Amazon Route 53

- Los registros DNS en Amazon Route 53 son esencialmente instrucciones que le dicen a otros servidores cómo interactuar con tu dominio.
- Cada registro DNS consta de:
  - **Nombre del dominio/subdominio:** Esto identifica el sitio web, como blockstellart.com o api.blockstellart.com
  - **Tipo de registro:** Indica el propósito del registro, como A (para direcciones IPv4) o AAAA (para direcciones IPv6)
  - **Valor:** La dirección IP o el valor específico al que apunta el registro, por ejemplo, 45.24.76.88.
  - **Política de enrutamiento:** Define cómo Route 53 responderá a las consultas, como balanceo de carga o geolocalización
  - **TTL:** Especifica cuánto tiempo los resolvers DNS deben almacenar en caché la respuesta del registro

# Tipos de registros de Amazon Route 53

- Route 53 soporta los siguientes tipos de registros DNS:
  - **Básicos:** A / AAAA / CNAME / NS / MX
  - **Avanzados:** CAA / DS / NAPTR / PTR / SOA / TXT / SPF / SRV
- **Registro A y AAAA:** Asocian un nombre de dominio con una dirección IP. El registro A se usa para direcciones IPv4, y el AAAA para IPv6
- **Registro CNAME (Canonical Name):** Permite asociar un alias de dominio (como www) con otro nombre de dominio. Es útil para apuntar múltiples nombres de dominio a un solo dominio de destino
  - Ejemplos: api.blockstellart.com, demo.blockstellart.com
- **Registro NS (Name Server):** Servidores de nombres para la zona alojada. Permite controlar cómo se enruta el tráfico de un dominio
- **Registro MX (Mail Exchange):** Indica los servidores de correo que manejarán el email de tu dominio, permitiendo priorizar si tienes más de uno.

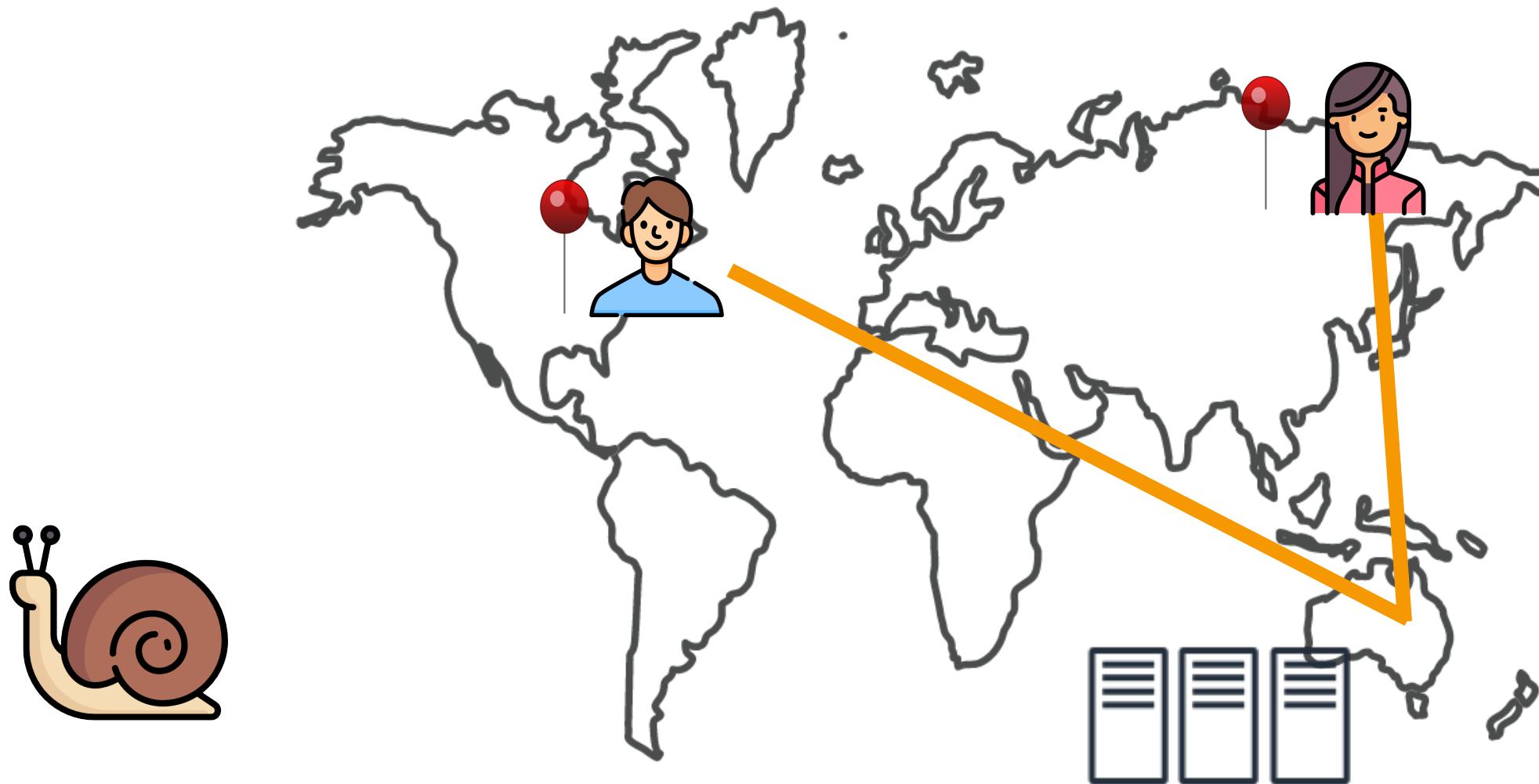


# CloudFront

[www.blockstellart.com](http://www.blockstellart.com)

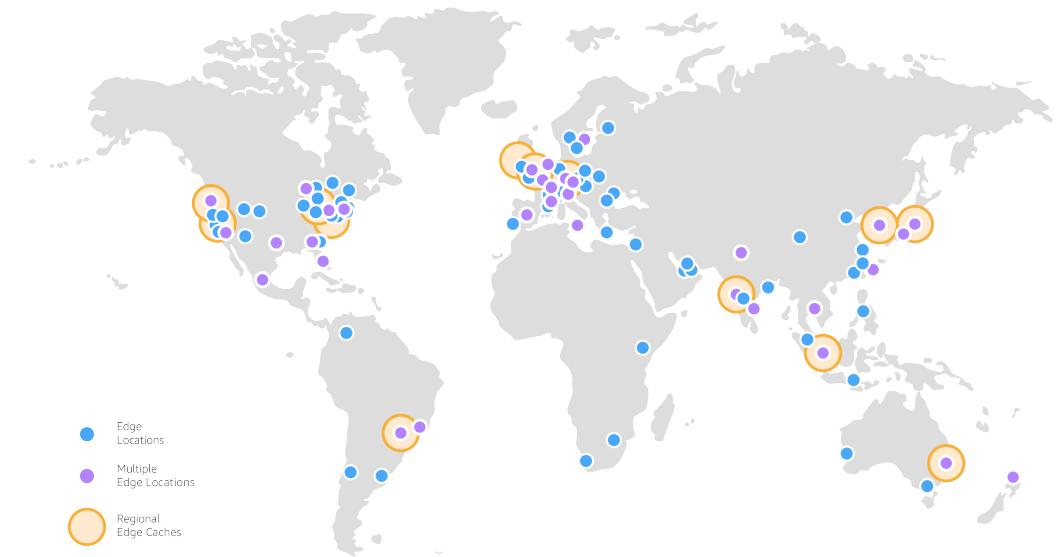
Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Contextualización del uso de CloudFront



# Visión general de AWS CloudFront

- Red de entrega de contenidos (CDN)
- Mejora el rendimiento de lectura, el contenido se almacena en caché en edge location
- Mejora la experiencia de los usuarios
- +700 puntos de presencia a nivel mundial (ubicaciones edge)
- Protección DDoS, integración con Shield, AWS Web Application Firewall
- Términos destacables de CloudFront:
  - **Origen:** La ubicación fuente de tu contenido
    - Origen de S3 o personalizado
  - **Distribución:** La unidad de 'configuración' de CloudFront
  - **Edge Location:** Caché local de tus datos
  - **Regional Edge Cache:** Versión más grande de una edge location. Proporciona otra capa de almacenamiento en caché



Fuente: <https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2>

# Orígenes de CloudFront



## Bucket S3

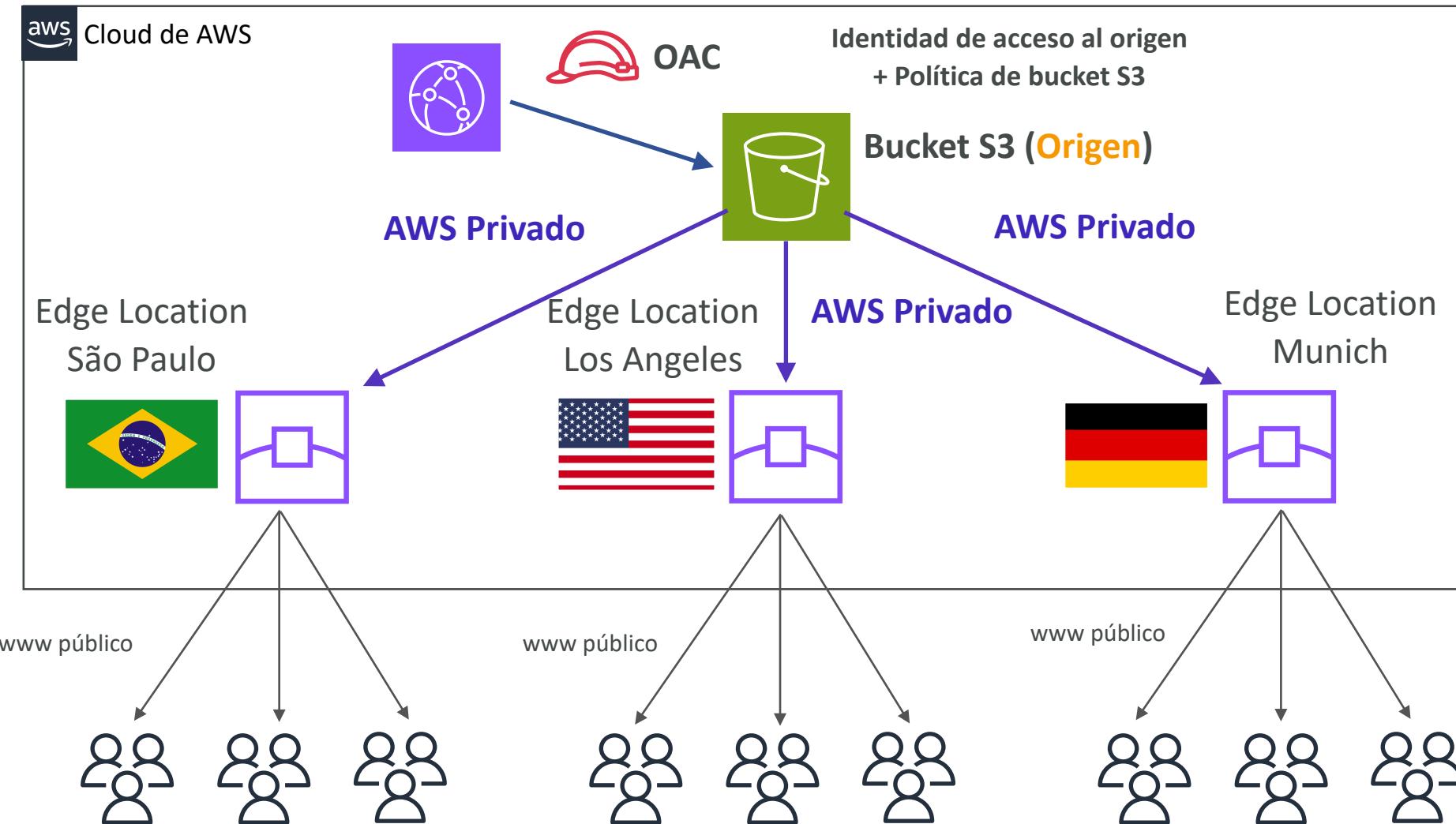
- Para distribuir archivos y almacenarlos en caché en el borde
- Seguridad mejorada con CloudFront
- OAC sustituye a Origin Access Identity (OAI)
- CloudFront puede utilizarse como entrada (para subir archivos a S3)



## Origen personalizado (HTTP)

- Application Load Balancer
- Instancia EC2
- Sitio web de S3 (primero debes habilitar el bucket como sitio web estático de S3)
- Cualquier backend HTTP que quieras

# CloudFront - S3 como origen



# CloudFront - ALB o EC2 como origen



## ¿Cómo sabemos la IP pública de las Edge Locations?

- <http://d7uri8nf7uskq.cloudfront.net/tools/list-cloudfront-ips>

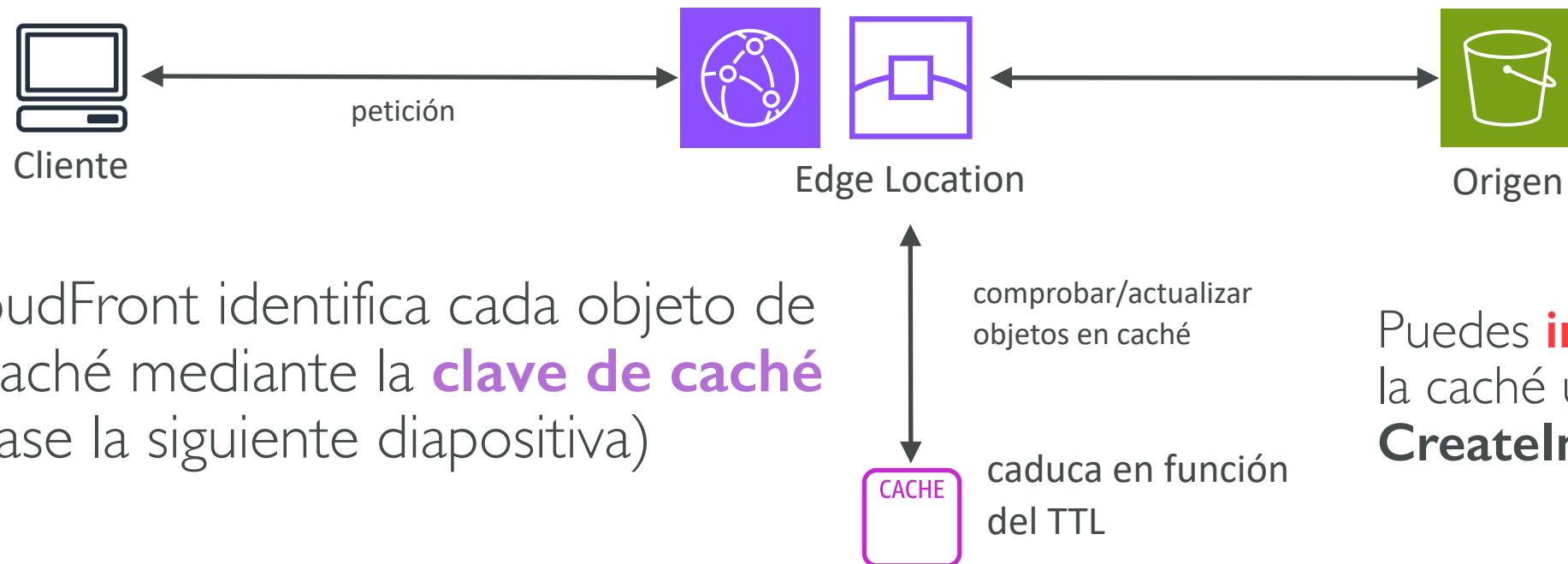
# CloudFront - ALB o EC2 como origen



# Almacenamiento en caché CloudFront

**La caché vive en cada Edge Location de CloudFront**

Quieres maximizar el ratio del golpe de caché (Cache Hit) para **minimizar las peticiones al origen**



CloudFront identifica cada objeto de la caché mediante la **clave de caché** (véase la siguiente diapositiva)

Puedes **invalidar** parte de la caché utilizando la API **CreateInvalidation**

# Clave de caché de CloudFront

## Un identificador único para cada objeto de la caché

GET /src/html/history.html?ref=123abc&split-pages=false HTTP/1.1

Host: **blockstellart.com**



Puedes añadir otros elementos (cabeceras HTTP, cookies, cadenas de consulta) a la clave de caché mediante las **políticas de caché de CloudFront**

Golpe de caché  
obtener objeto si existe

Por defecto, consiste en el **hostname + la parte del recurso de la URL**

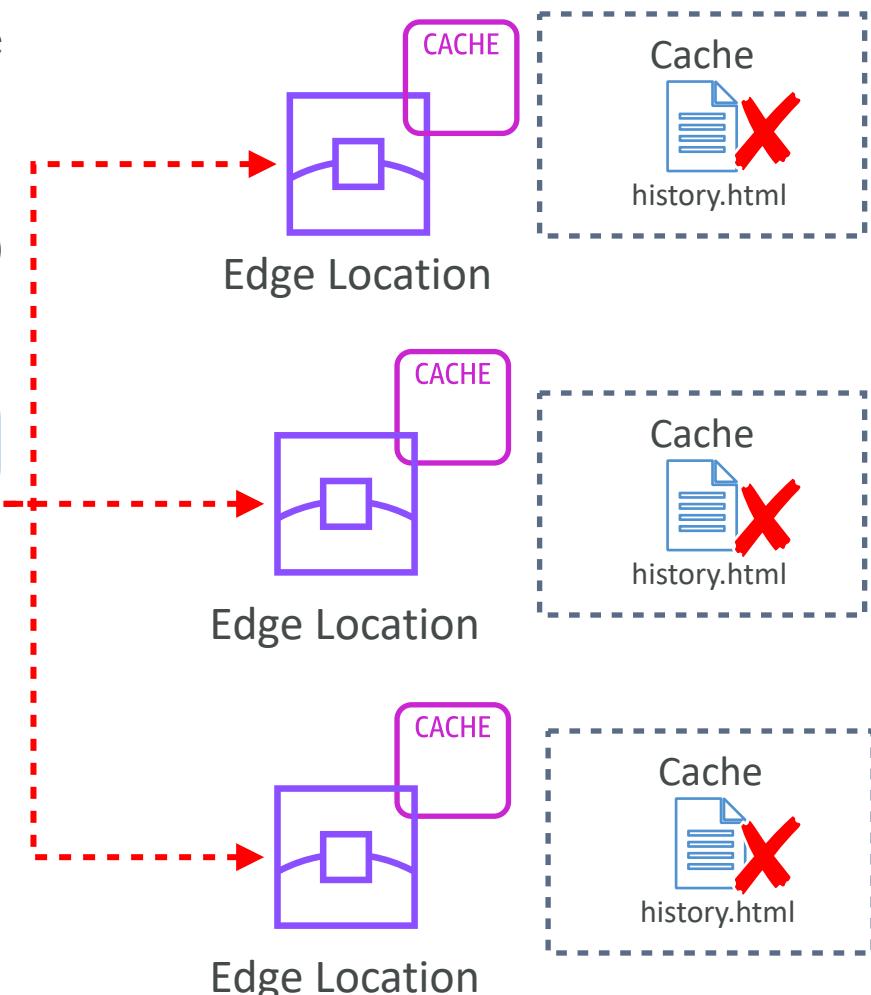
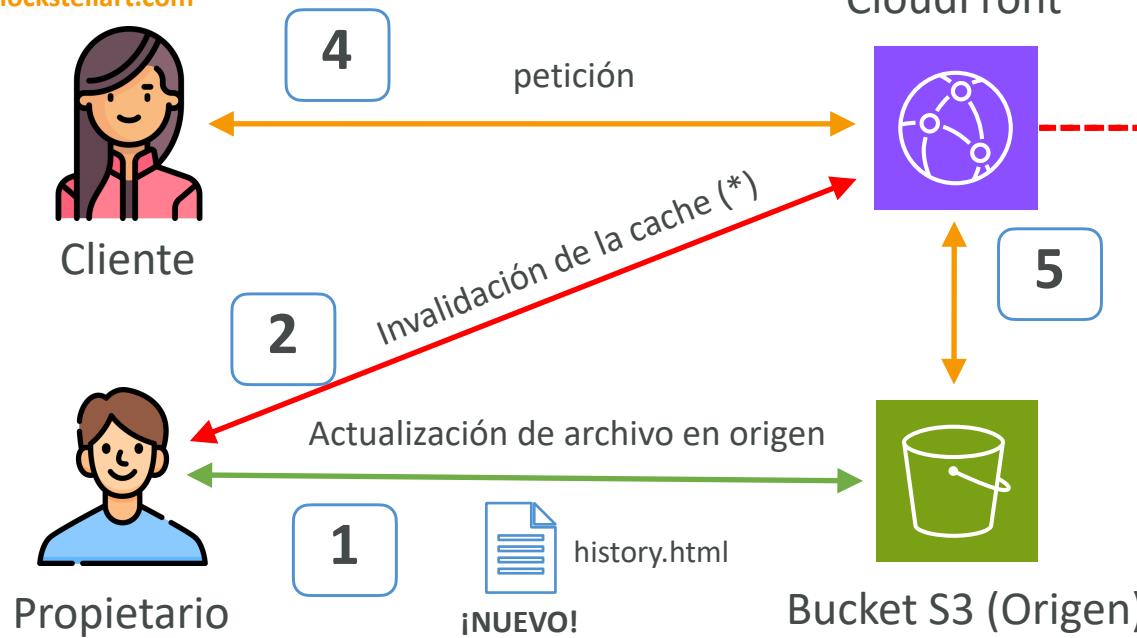
| Clave de caché (caché basada en) | Objeto |
|----------------------------------|--------|
| - blockstellart.com              |        |
| - /src/html/history.html         |        |

caduca en función  
del TTL

# Invalidación de la caché en CloudFront

- En caso de que actualices el origen del back-end, CloudFront no lo sabe y sólo obtendrá el contenido refrescado cuando el TTL haya expirado
- **Puedes forzar una actualización total o parcial de la caché** (obviando así el TTL) realizando una **Invalidación de CloudFront**
- Puedes invalidar todos los archivos (\*) o una ruta especial (/índices/\*)

GET /src/html/history.html?ref=123abc&split-pages=false HTTP/1.1  
Host: **blockstellart.com**



# Docker en AWS: ECR, ECS, Fargate, EKS

# ¿Qué es Docker?

- Docker es una plataforma de desarrollo de software para desplegar aplicaciones
- Las aplicaciones se empaquetan en **contenedores** que pueden ejecutarse en cualquier sistema operativo
- **Las aplicaciones se ejecutan igual, independientemente de dónde se ejecuten**
  - Cualquier máquina
  - No hay problemas de compatibilidad
  - Comportamiento predecible
  - Menos trabajo
  - Más fácil de mantener y desplegar
  - Funciona con cualquier lenguaje, cualquier sistema operativo y cualquier tecnología
- Amplía y reduce los contenedores muy rápidamente (en segundos)



# Docker en un sistema operativo

- Los contenedores de Docker se ejecutan igual en cualquier infraestructura:
  - Máquina local
  - Centro de datos corporativo
  - Cloud
- Son ligeros
  - En comparación con las máquinas virtuales
- Docker proporciona aislamiento para los contenedores



# Conceptos clave sobre Docker



**Dockerfile** - Es un archivo de texto con las instrucciones necesarias para crear una imagen (algo similar a un plano de construcción)



**Imagen** - Es un archivo construido por capas, que contiene todas las dependencias para ejecutarse



**Contenedor** - Es la instancia de una image en un ambiente aislado



**Portable** - Autónomo, siempre funciona como se espera



**Ligero** - Se utiliza el sistema operativo principal y se comparten las capas del sistema de archivos.

# Almacenamiento de imágenes de Docker

- Las imágenes Docker se almacenan en **repositorios Docker**



- **Docker Hub (<https://hub.docker.com>)**

- Repositorio **público**
- Repositorio **privado (para planes Pro, Team y Business)**
- Imágenes base para muchas tecnologías o sistemas operativos (por ejemplo, Ubuntu, MySQL, ...)



- **Azure Container Registry (ACR)**

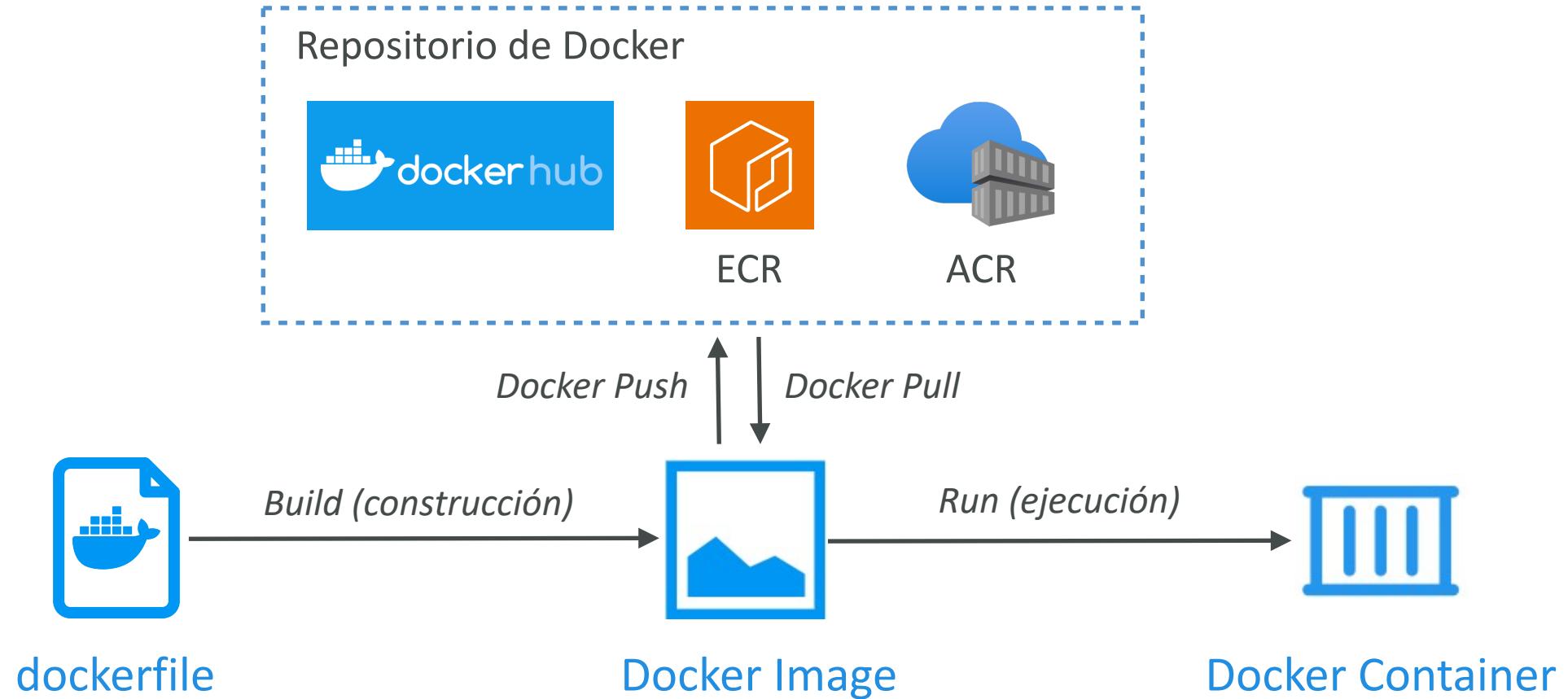


Amazon ECR

- **Amazon ECR (Registro elástico de contenedores de Amazon)**

- Repositorio **público (Galería pública de Amazon ECR <https://gallery.ecr.aws>)**
- Repositorio **privado**

# Conceptos clave sobre Docker



# Gestión de contenedores Docker en AWS



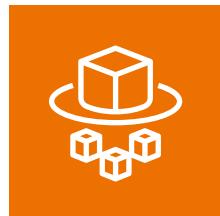
## **Amazon Elastic Container Service (Amazon ECS):**

Plataforma de contenedores propia de Amazon



## **Servicio Amazon Elastic Kubernetes (Amazon EKS):**

Kubernetes administrado por Amazon (código abierto)



## **AWS Fargate:** Plataforma de contenedores sin servidor

propia de Amazon y funciona con ECS y con EKS



## **Amazon ECR:** Almacena imágenes de contenedores



# Visión general de Amazon ECS

- Amazon ECS es un **servicio de orquestación de contenedores** completamente administrado que facilita la implementación, la administración y el escalado de aplicaciones en contenedores.

\*Lanzar contenedores Docker en AWS = Lanzar **tareas ECS** en clústeres ECS

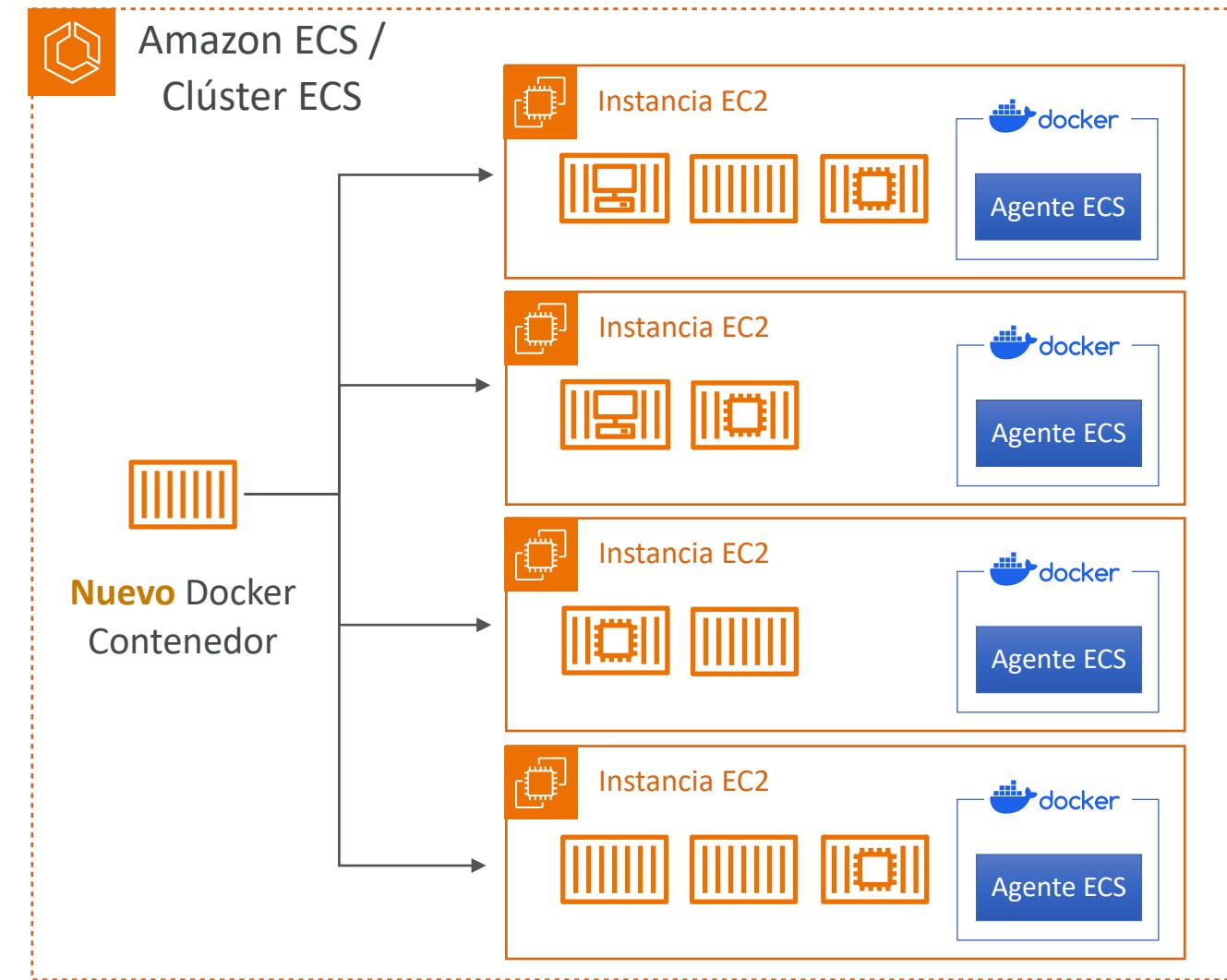
- Tipos de lanzamiento de contenedores con Amazon ECS:

- **Instancias EC2**: debe aprovisionar y mantener la infraestructura (las instancias EC2)
- **Fargate**: No aprovisionas la infraestructura (no hay instancias EC2 que administrar)



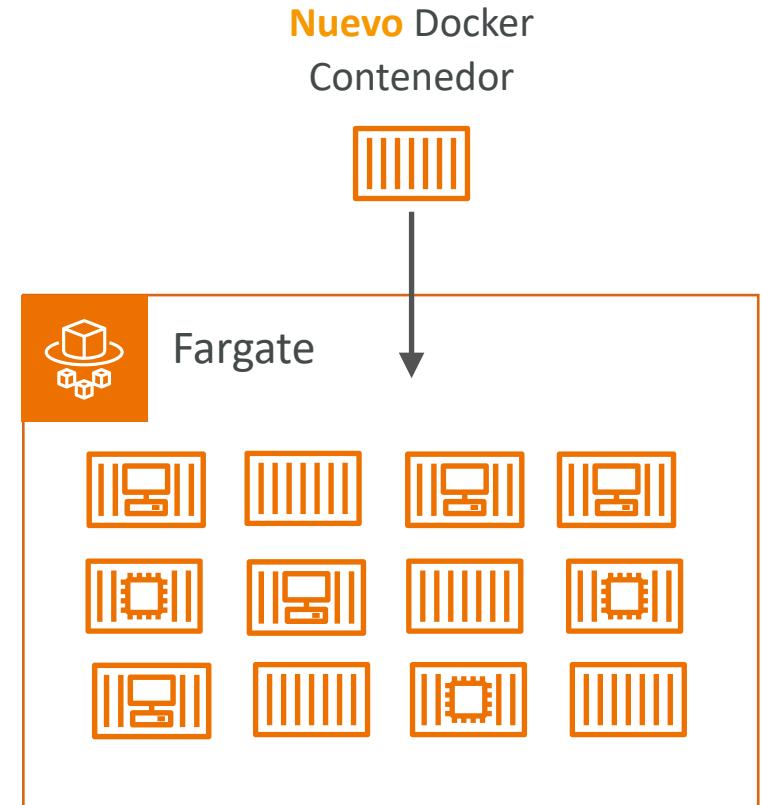
# Lanzamiento de ECS usando instancias EC2

- **Tipo de lanzamiento EC2: debe aprovisionar y mantener la infraestructura (las instancias EC2)**
- Puedes seleccionar el tipo de instancia EC2 y la cantidad que mejor se adapte a tus necesidades de carga de trabajo y rendimiento
- Tienes control sobre la configuración del sistema operativo y la capacidad de instalar software personalizado en las instancias
- Cada Instancia EC2 debe ejecutar el Agente ECS para registrarse en el Cluster ECS
- AWS se encarga de iniciar / detener los contenedores



# Lanzamiento de ECS usando Fargate

- **No aprovisionas la infraestructura**
- **No hay instancias EC2 que administrar**
- **Todo es Serverless (sin servidor)**
- Sólo tienes que crear definiciones de tareas
- Fargate permite escalar automáticamente y pagar por uso
- AWS ejecuta las tareas ECS por ti en función de la CPU / RAM que necesites
- Fargate aísla cada contenedor, mejorando la seguridad en la ejecución de tus aplicaciones



# Amazon ECS – Tareas

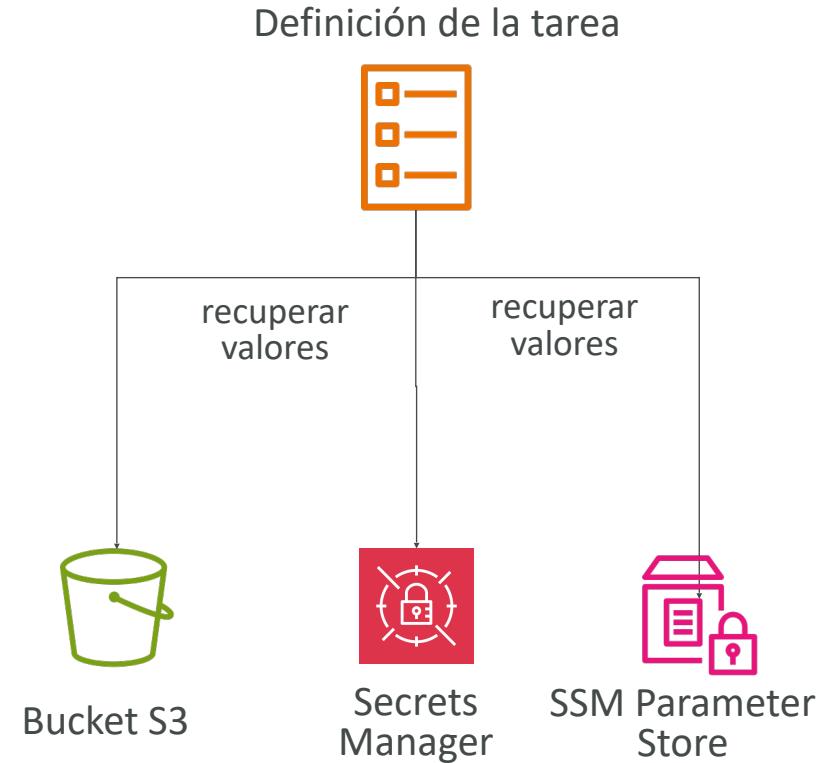
- Se trata de un archivo de texto en formato JSON que describe los parámetros y uno o varios contenedores que forman la aplicación.
- Entre los parámetros que se pueden especificar en una definición de tareas se incluyen los siguientes:
  - El **tipo de lanzamiento** a utilizar, que determina la infraestructura en la que se alojan las tareas
  - La **imagen de Docker** que se va a utilizar con cada contenedor en su tarea
  - La cantidad de **CPU y de memoria** que se va a utilizar con cada tarea o cada contenedor dentro de una tarea
  - El modo de **red de Docker** que utilizar para los contenedores
  - Los **volúmenes de datos** que se utilizan con los contenedores en la tarea
  - El **rol de IAM** que las tareas utilizan
  - Comportamiento de la tarea si el contenedor finaliza o falla
  - El comando que el contenedor ejecuta al iniciarse



```
{  
  "family": "my-task",  
  "containerDefinitions": [  
    {  
      "name": "my-container",  
      "image": "my-image",  
      "memory": 512,  
      "portMappings": [  
        {  
          "containerPort": 80,  
          "hostPort": 80,  
          "protocol": "tcp"  
        }  
      ],  
      "essential": true  
    }  
  ]  
}
```

# Amazon ECS – Tareas con variables de entorno

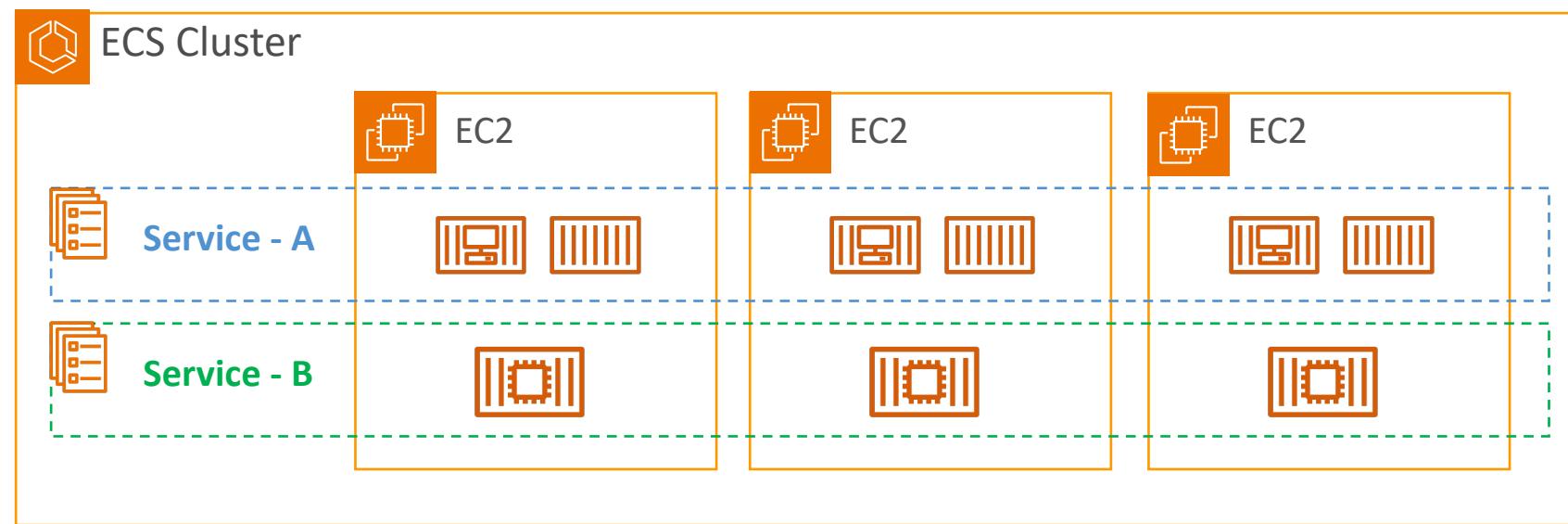
- Variable de entorno
  - **Hardcoded** - por ejemplo, URLs
  - **Almacén de Parámetros SSM** - variables sensibles (por ejemplo, claves API, configuraciones compartidas)
  - **Secrets Manager** - variables sensibles (por ejemplo, contraseñas de BD)
- Archivos de entorno (masivos) - Amazon S3



```
{  
  "family": "my-task",  
  "containerDefinitions": [  
    {  
      "name": "my-container",  
      "image": "my-image",  
      "memory": 512,  
      "environment": [  
        {  
          "name": "ENV_VAR_1",  
          "value": "value1"  
        },  
        {  
          "name": "ENV_VAR_2",  
          "value": "value2"  
        }  
      ],  
      "portMappings": [  
        {  
          "containerPort": 80,  
          "hostPort": 80,  
          "protocol": "tcp"  
        }  
      ],  
      "essential": true  
    }  
  ]  
}
```

# Amazon ECS – Servicios

- Definen cuántas tareas deben ejecutarse y cómo deben ejecutarse
- Garantizan que el número de tareas deseado se ejecuta en toda nuestra flota de instancias EC2
- Pueden vincularse a Elastic Load Balancer (ALB o NLB) si es necesario
- ¡Vamos a crear nuestro primer servicio!





# AWS Glue

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# ¿Qué es AWS Glue?



- AWS Glue es un servicio serverless de integración de datos completamente gestionado que facilita la preparación y la carga de datos para su análisis
- Compatible con:



Amazon  
Redshift



Amazon  
DynamoDB



Amazon RDS



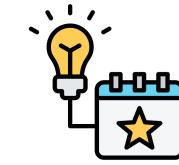
Amazon S3



Compatible con la  
mayoría de otras  
bases de datos SQL

- Trabajos ETL personalizados

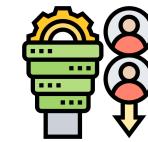
Activados por:



Eventos



Programados



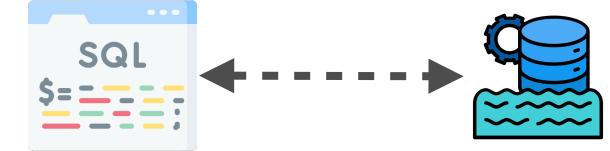
Bajo demanda

- Completamente gestionado

# Casos de uso de AWS Glue

## 1. Consultas en un lago de datos S3:

- Permite realizar análisis de datos sin la necesidad de mover los datos físicamente, aprovechando AWS Glue para la preparación de los datos



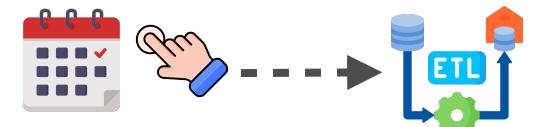
## 2. Análisis de datos de registro:

- Proporciona la capacidad de transformar y enriquecer los datos de registro almacenados en un almacén de datos utilizando scripts ETL



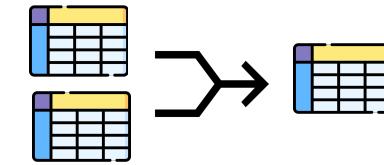
## 3. Pipelines ETL impulsados por eventos:

- Facilita la automatización de procesos ETL en respuesta a la llegada de nuevos datos, usando funciones Lambda para ejecutar trabajos de AWS Glue



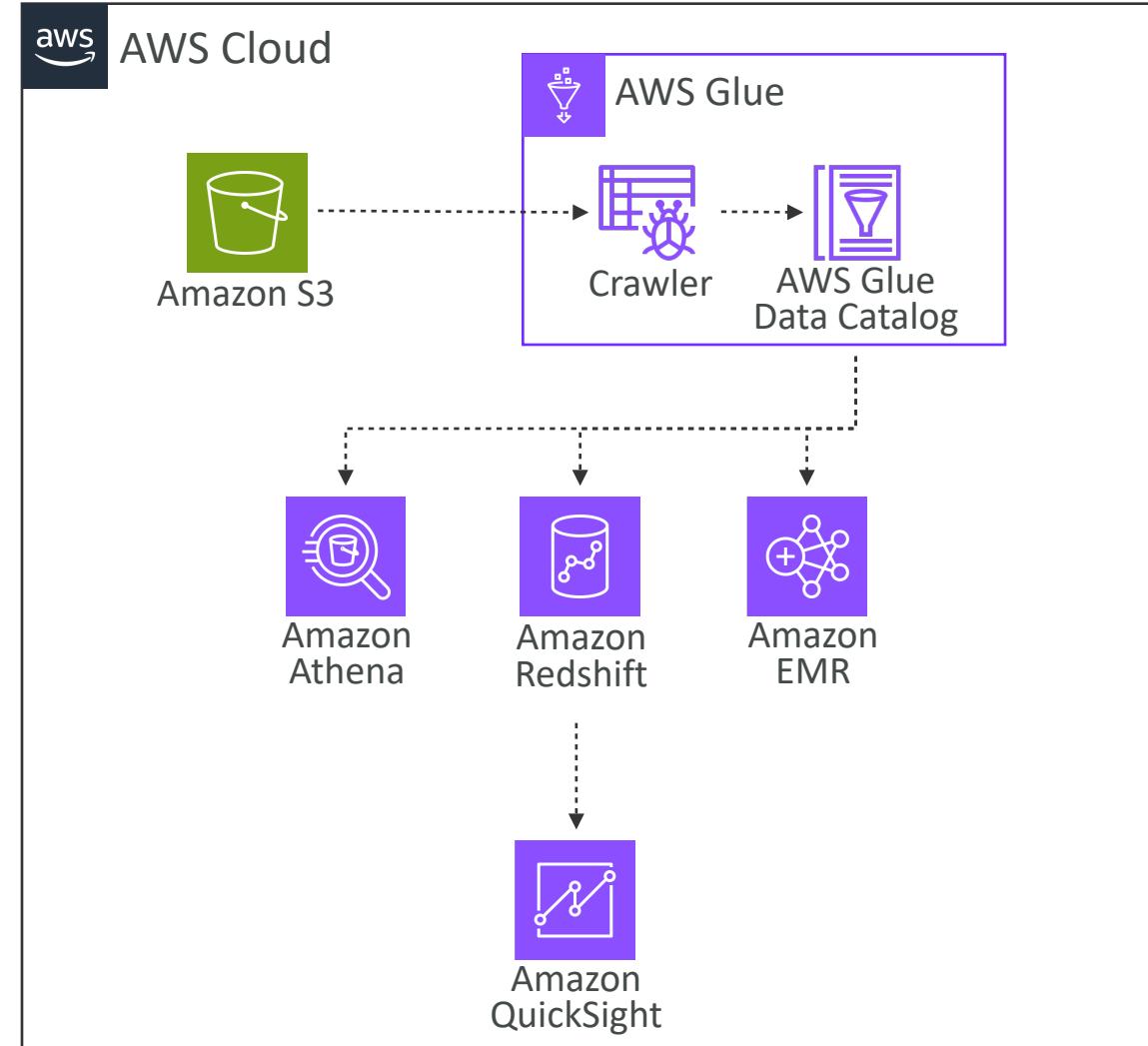
## 4. Vista unificada de datos:

- Utiliza AWS Glue Data Catalog para centralizar y gestionar metadatos, lo que permite una fácil búsqueda y descubrimiento de conjuntos de datos distribuidos en diferentes almacenes de datos



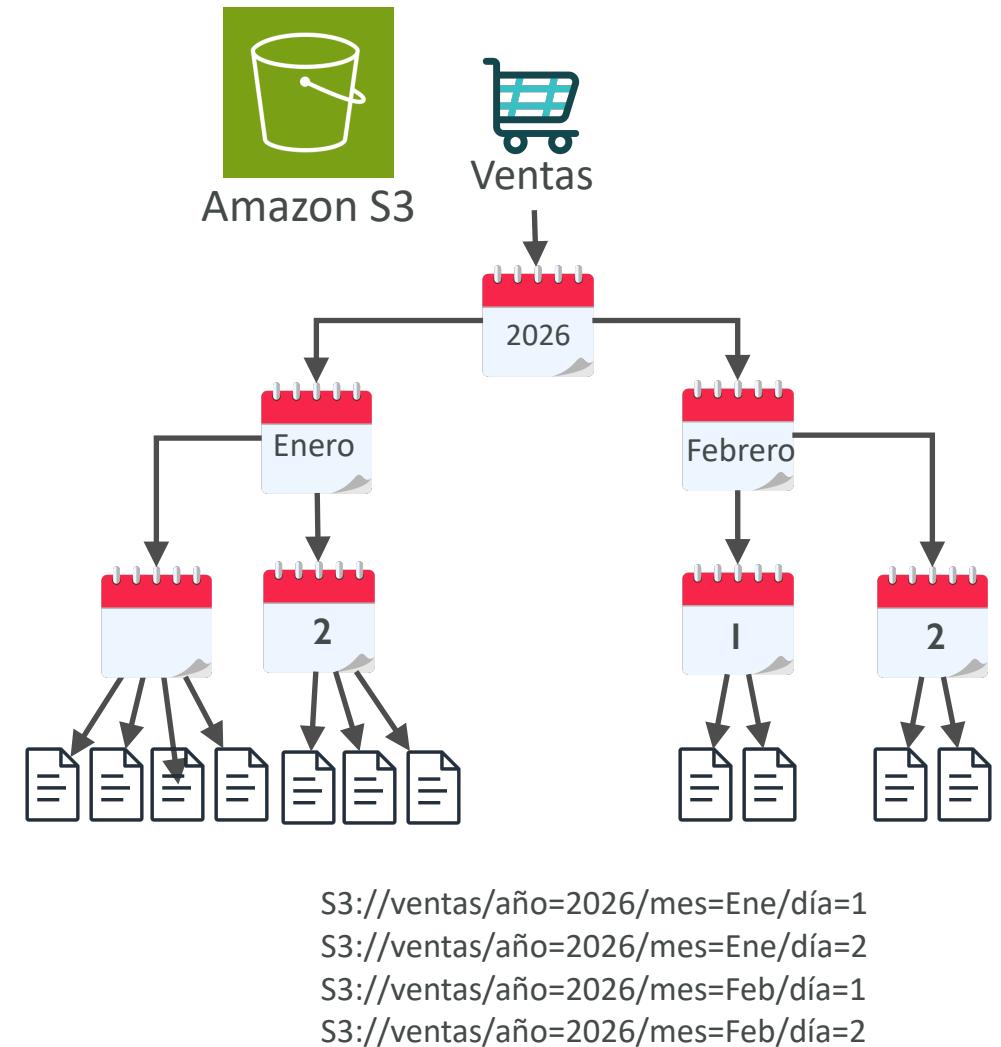
# AWS Glue Crawler / Data Catalog

- Glue crawler escanea datos en S3, crea esquemas
- Puede ejecutarse periódicamente
- El catálogo de datos de Glue (Data Catalog):
  - Solo almacena la definición de la tabla
  - Los datos originales permanecen en S3
- Una vez catalogados, puedes tratar tus datos no estructurados como si fueran estructurados:
  - Amazon Redshift Spectrum
  - Amazon Athena
  - Amazon EMR
  - Amazon QuickSight



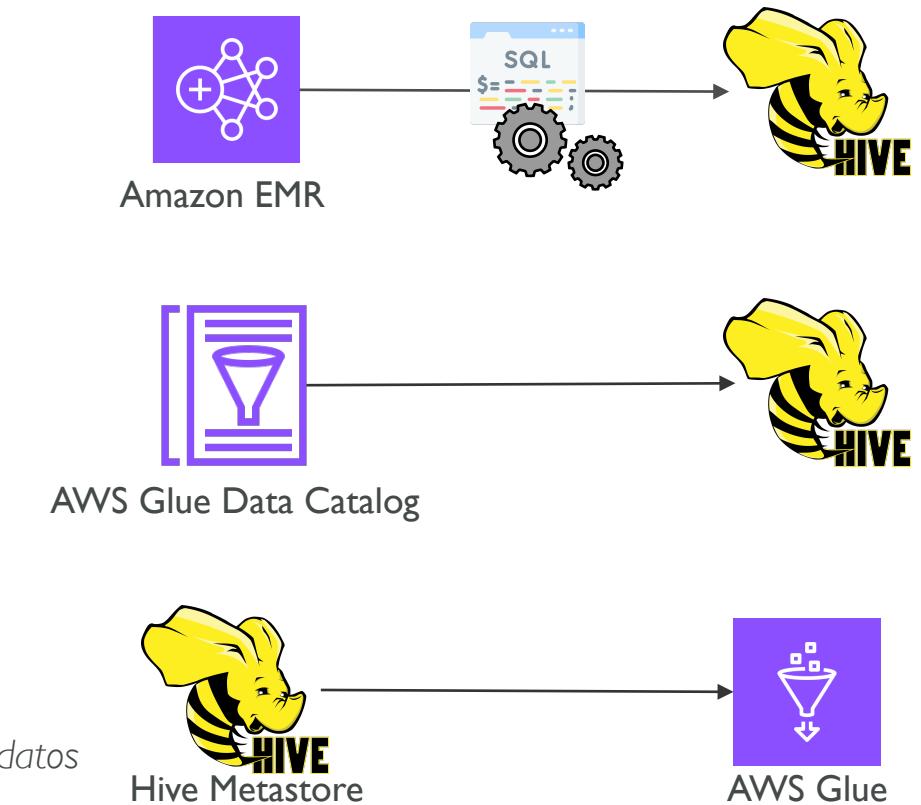
# AWS Glue y Particiones S3

- El crawler de Glue extraerá particiones basadas en cómo están organizados tus datos en S3
- Piensa de antemano en cómo consultarás tu lago de datos en S3
- Por ejemplo, si el crawler detecta múltiples carpetas en un bucket de S3 que siguen una estructura jerárquica por fecha (año, mes, día), determinará automáticamente estas carpetas como particiones



# AWS Glue + Hive

- Hive facilita la **consulta y la gestión de grandes conjuntos de datos** que residen en almacenamiento distribuido utilizando SQL
  - Hive te permite ejecutar **consultas tipo SQL desde EMR**
  - El catálogo de datos de Glue puede servir como un **metastore** de Hive
  - También puedes **importar un metastore de Hive** en Glue
- *Un metastore de Hive es un sistema de almacenamiento centralizado que guarda los metadatos de las estructuras y ubicaciones de datos en un sistema de almacenamiento distribuido*



# AWS Glue ETL

- Facilita la preparación, transformación y carga de datos para análisis y procesamiento (**Scala o Python**)
- Encriptación:
  - Del lado del servidor (en reposo)
  - SSL (en tránsito)
- Puede ser impulsado por eventos
- Puede aprovisionar **Unidades adicionales de Procesamiento de Datos (DPU's)** para aumentar el rendimiento de los trabajos de Spark subyacentes
  - Habilitar las métricas de trabajos puede ayudarte a entender la capacidad máxima en DPU's que necesitas
- Errores reportados a CloudWatch
  - Podrían integrarse con SNS para notificaciones



# AWS Glue ETL

- Transformar datos, limpiar datos, enriquecer datos (antes de realizar el análisis)



Generar código ETL en Python o Scala, puedes modificar el código



Puedes proporcionar tus propios scripts de Spark o PySpark

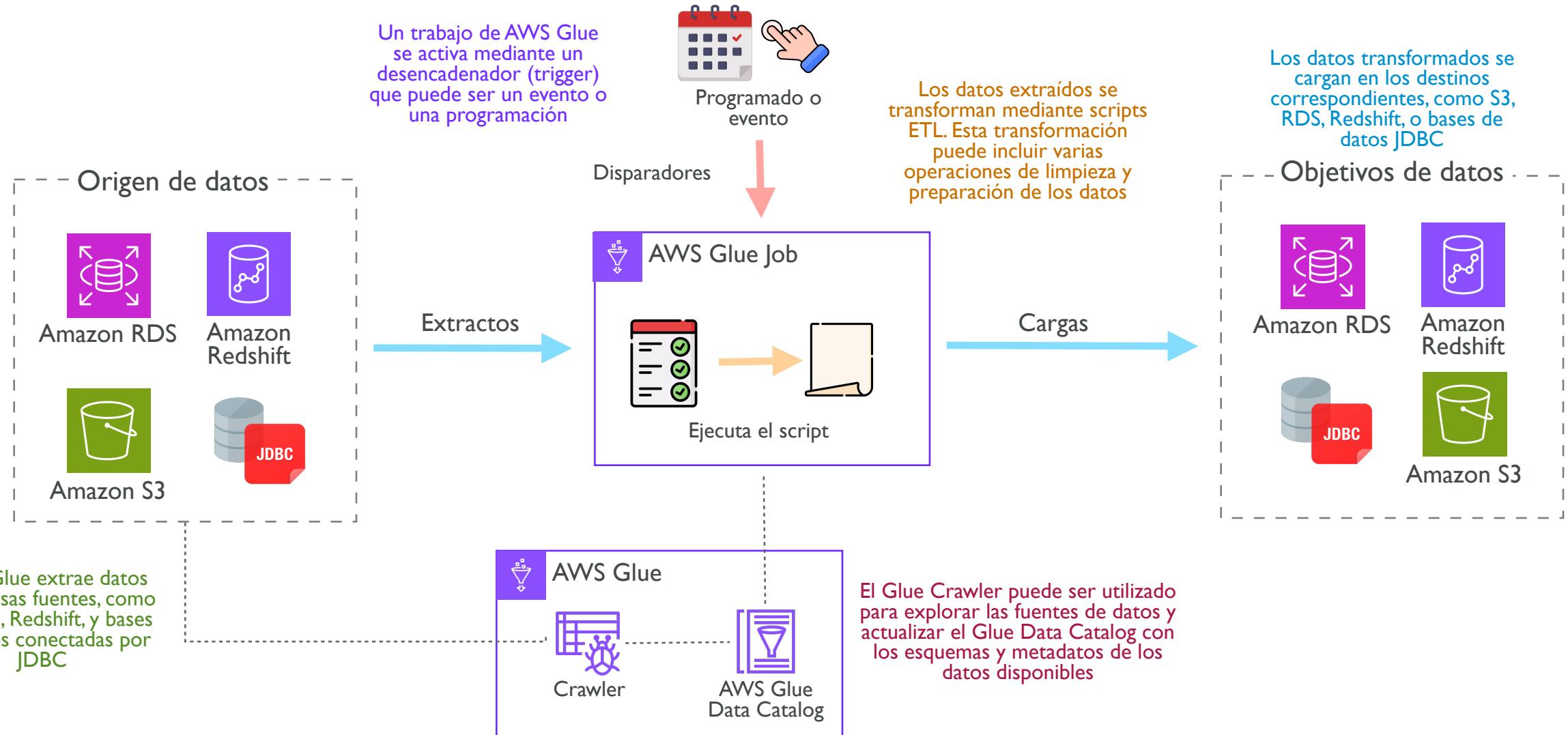


El destino puede ser S3, JDBC (RDS, Redshift), o en el Catálogo de Datos de Glue

- Totalmente gestionado, rentable, paga solo por los recursos consumidos
- Los trabajos se ejecutan en una plataforma Spark sin servidor
- Glue Scheduler para programar los trabajos
- Glue Triggers para automatizar la ejecución de trabajos basados en "eventos"



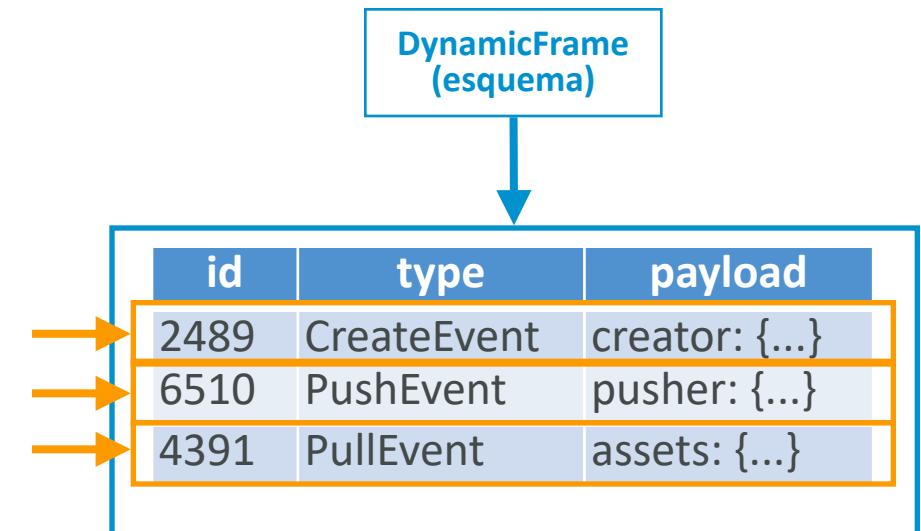
# Diagrama del uso de AWS Glue ETL



# AWS Glue ETL - DynamicFrame

- Un **DynamicFrame** es un marco que maneja datos heterogéneos. Básicamente una colección de **DynamicRecords**
  - Los DynamicRecords se describen a sí mismos, tienen un esquema
  - Muy parecido a un DataFrame de Spark, pero con más funcionalidades ETL
  - APIs en Scala y Python

El esquema del DynamicFrame incluye las columnas comunes: id, type y payload



Cada registro dinámico tiene un id, un type y un payload.  
El payload puede contener diferentes tipos de información dependiendo del type.

# AWS Glue ETL - Transformaciones

- Transformaciones Incluidas:



**DropFields, DropNullFields:**  
Elimina campos (nulos)



**Filter:** Especifica una función  
para filtrar registros

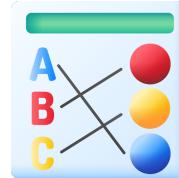


**Join:** Para  
enriquecer datos



**Map:** Añadir campos, eliminar  
campos, realizar búsquedas externas

- Transformaciones de Machine Learning:



**FindMatches ML:** Identifica registros duplicados o coincidentes en tu conjunto de datos

- Conversiones de formato: CSV, JSON, Avro, Parquet, ORC, XML



- Transformaciones de Apache Spark (ejemplo: K-Means):

- Puede convertir entre Spark DataFrame y Glue DynamicFrame



# AWS Glue ETL - ResolveChoice

- Es una función en AWS Glue ETL que maneja las ambigüedades en un *DynamicFrame*, como cuando hay múltiples campos con el mismo nombre o cuando los tipos de datos no son consistentes
- Opciones de transformación:
  1. **make\_cols**: Crea una nueva columna para cada tipo.  
Por ejemplo, dos campos con el mismo nombre
  2. **cast**: Convierte todos los valores al tipo especificado
  3. **make\_struct**: Crea una estructura que contiene cada tipo de dato
  4. **project**: Proyecta cada tipo a un tipo dado

```
"myList": [  
  {  
    "price": 100.00  
  },  
  {  
    "price": "$100.00"  
  }  
]
```

```
Dataframe1 = df.resolveChoice(choice = "make_cols")  
Dataframe2 = df.resolveChoice(specs = [("myList[]].price",  
"make_struct"), ("columnA", "cast:double")])
```

# Modificación del catálogo de datos

- Los scripts ETL pueden actualizar tu esquema y particiones si es necesario

1

## Añadir nuevas particiones

Volver a ejecutar el crawler, o

Hacer que el script use las opciones enableUpdateCatalog y partitionKeys

2

## Actualizar el esquema de la tabla

Volver a ejecutar el crawler, o

Usar enableUpdateCatalog / updateBehavior desde el script

# AWS Glue ETL - Modificación del catálogo de datos

3

## Actualizar el esquema de la tabla

Volver a ejecutar el crawler, o

Usar enableUpdateCatalog / updateBehavior desde el script

4

## Crear nuevas tablas

enableUpdateCatalog / updateBehavior con setCatalogInfo

5

## Restricciones

Solo S3

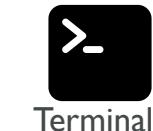
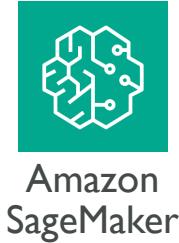
Solo json, csv, avro, parquet

Parquet requiere código especial

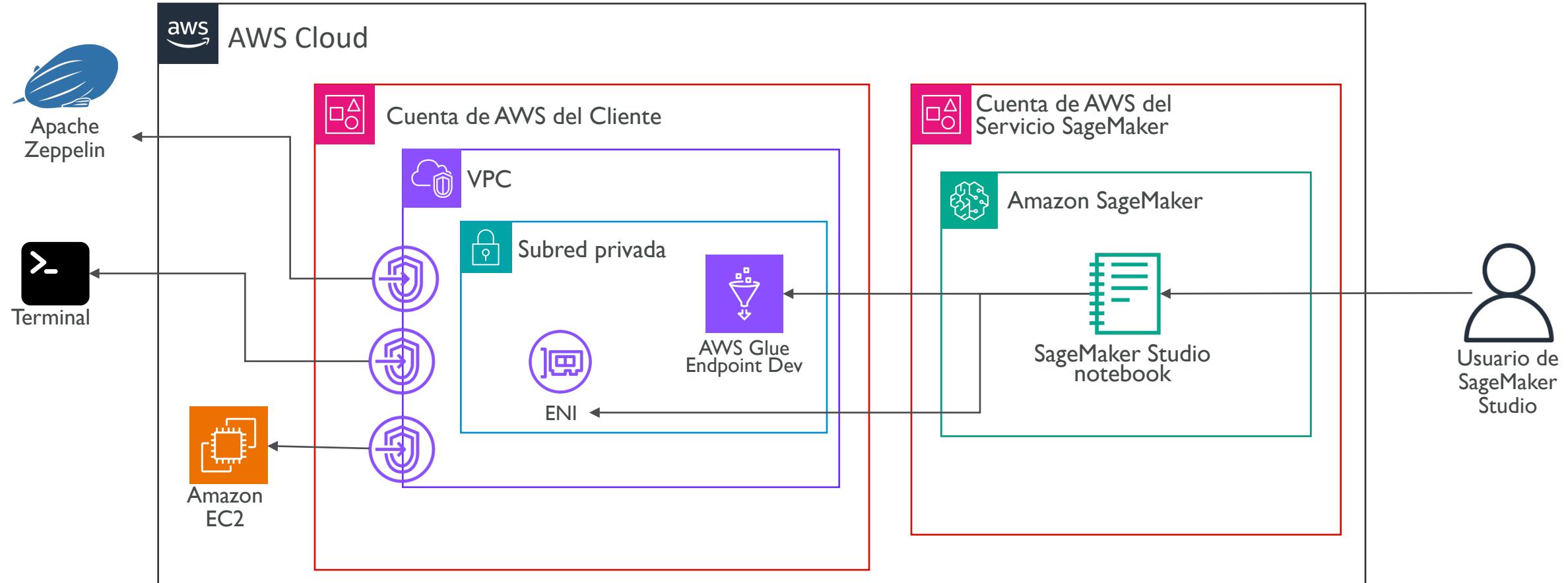
Los esquemas anidados no son soportados

# Puntos finales de desarrollo de AWS Glue

- **Desarrollo de scripts ETL en Notebooks:** AWS Glue permite desarrollar scripts ETL en entornos de notebooks, como:
  - Apache Zeppelin y SageMaker
- Una vez que el script está listo, se puede crear un trabajo ETL que lo ejecute usando Spark y Glue
- **Conexión al Endpoint en una VPC:** El endpoint de desarrollo está ubicado en una VPC y está controlado por grupos de seguridad. Puedes conectarte utilizando diferentes métodos:
  - Apache Zeppelin en tu máquina local
  - Servidor de notebook Zeppelin en EC2 (a través de la consola de Glue)
  - Notebook de SageMaker
  - Ventana de terminal
  - PyCharm edición profesional
  - Usa direcciones IP elásticas para acceder a una dirección de endpoint privada

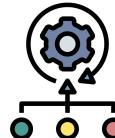


# Puntos finales de desarrollo de AWS Glue



# AWS Glue - Ejecución de trabajos

- Horarios basados en el tiempo (estilo cron)
- Marcadores de trabajo
- Eventos de CloudWatch



Persiste el estado desde la ejecución del trabajo



Previne el reprocesamiento de datos antiguos



Permite procesar solo datos nuevos al volver a ejecutar un trabajo en un horario



Funciona con fuentes S3 en una variedad de formatos



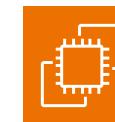
Funciona con bases de datos relacionales a través de JDBC (si las claves primarias están en orden secuencial)



Solo maneja nuevas filas, no filas actualizadas



Dispara una función Lambda o una notificación SNS cuando un trabajo ETL tiene éxito o falla



Invoca la ejecución de una instancia EC2, envía eventos a Kinesis, activa una función Step Function

# AWS Glue - Modelo de costos

- Facturación por segundo para crawlers y trabajos ETL
- El primer millón de objetos almacenados y accesos son gratuitos para el catálogo de datos de Glue
- Los endpoints de desarrollo para desarrollar código ETL se cobran por minuto



# AWS Glue Studio

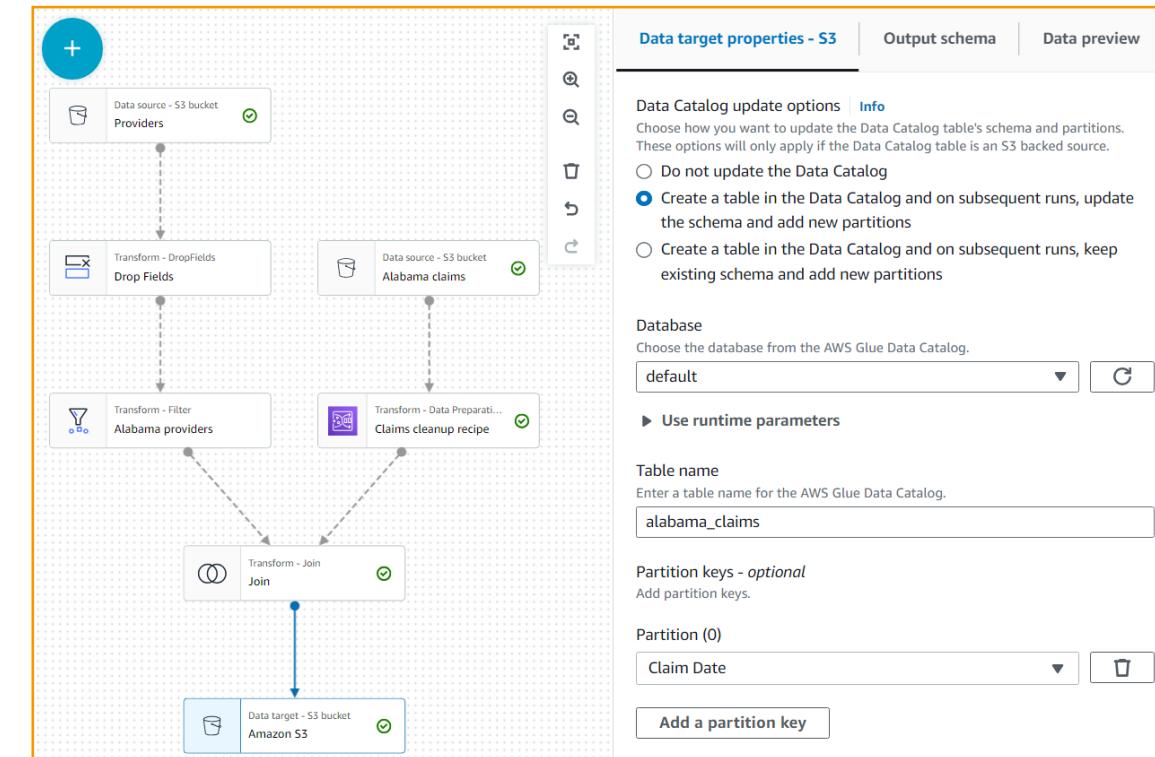
- **Interfaz visual para flujos de trabajo ETL**

- **Editor de trabajos visuales**

- Crear DAGs (grafos acíclicos dirigidos) para flujos de trabajo complejos
- Las fuentes incluyen S3, Kinesis, Kafka, JDBC
- Transformar / muestrear / unir datos, etc...
- Destino a S3 o Glue Data Catalog
- Soporta particionamiento

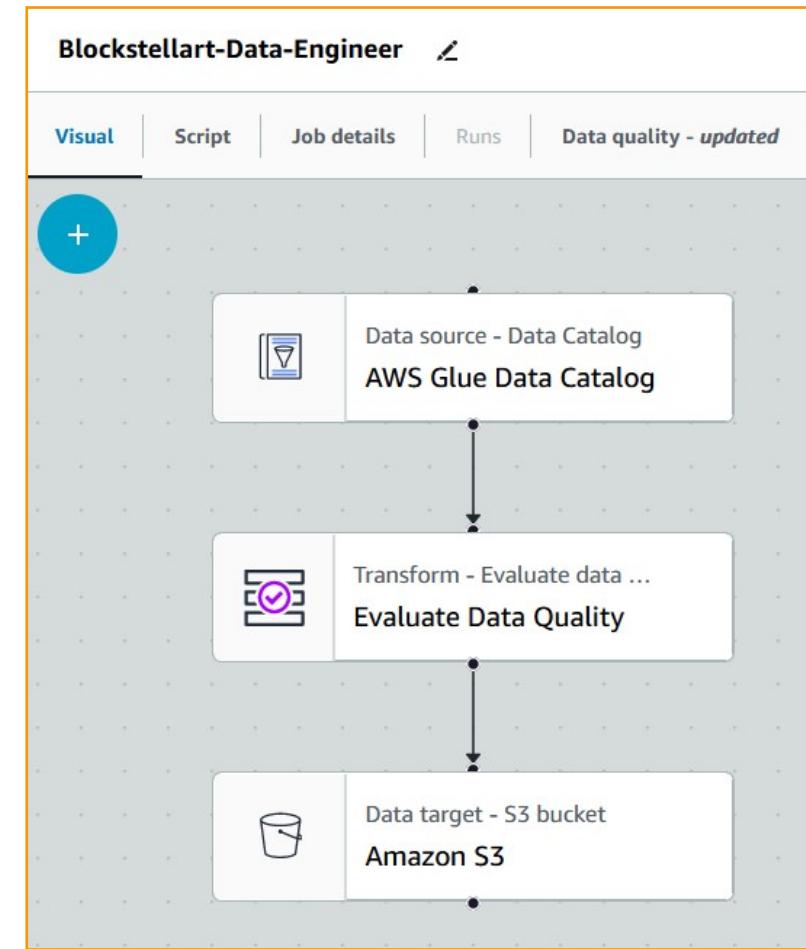
- **Panel de control visual de trabajos**

- Vistas generales, estado, tiempos de ejecución

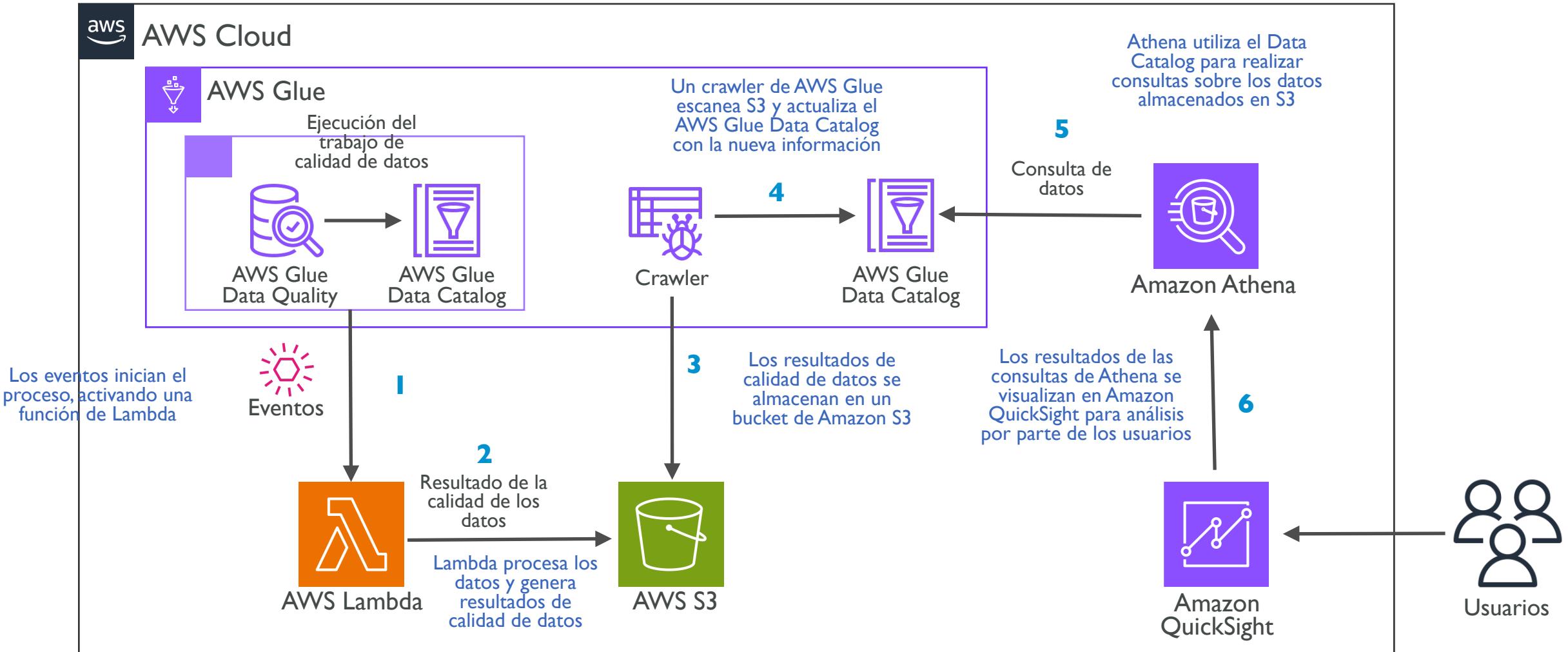


# Calidad de datos de AWS Glue

- Las reglas de calidad de datos pueden ser creadas manualmente o recomendadas automáticamente
- Se integra en los trabajos de Glue
- Utiliza el Data Quality Definition Language (DQDL)
- Los resultados pueden ser usados para fallar el trabajo, o simplemente ser reportados a CloudWatch



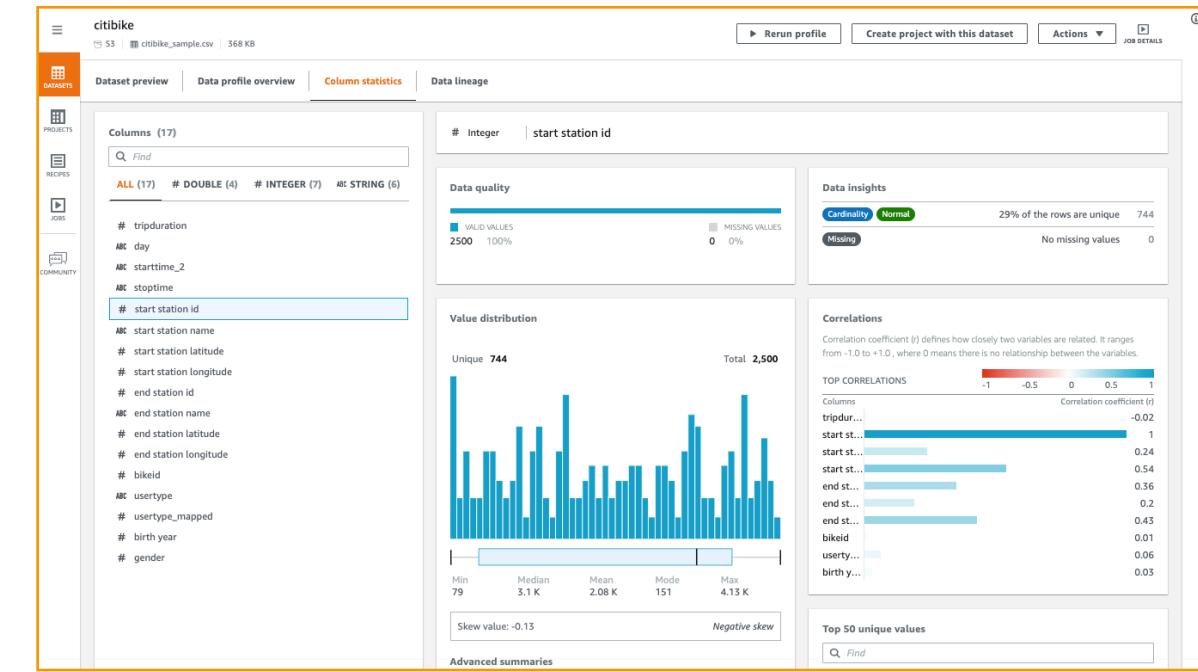
# Calidad de datos de AWS Glue



# AWS Glue DataBrew

AWS Glue DataBrew es una herramienta que facilita la **preparación y limpieza de datos** a través de una interfaz visual intuitiva.

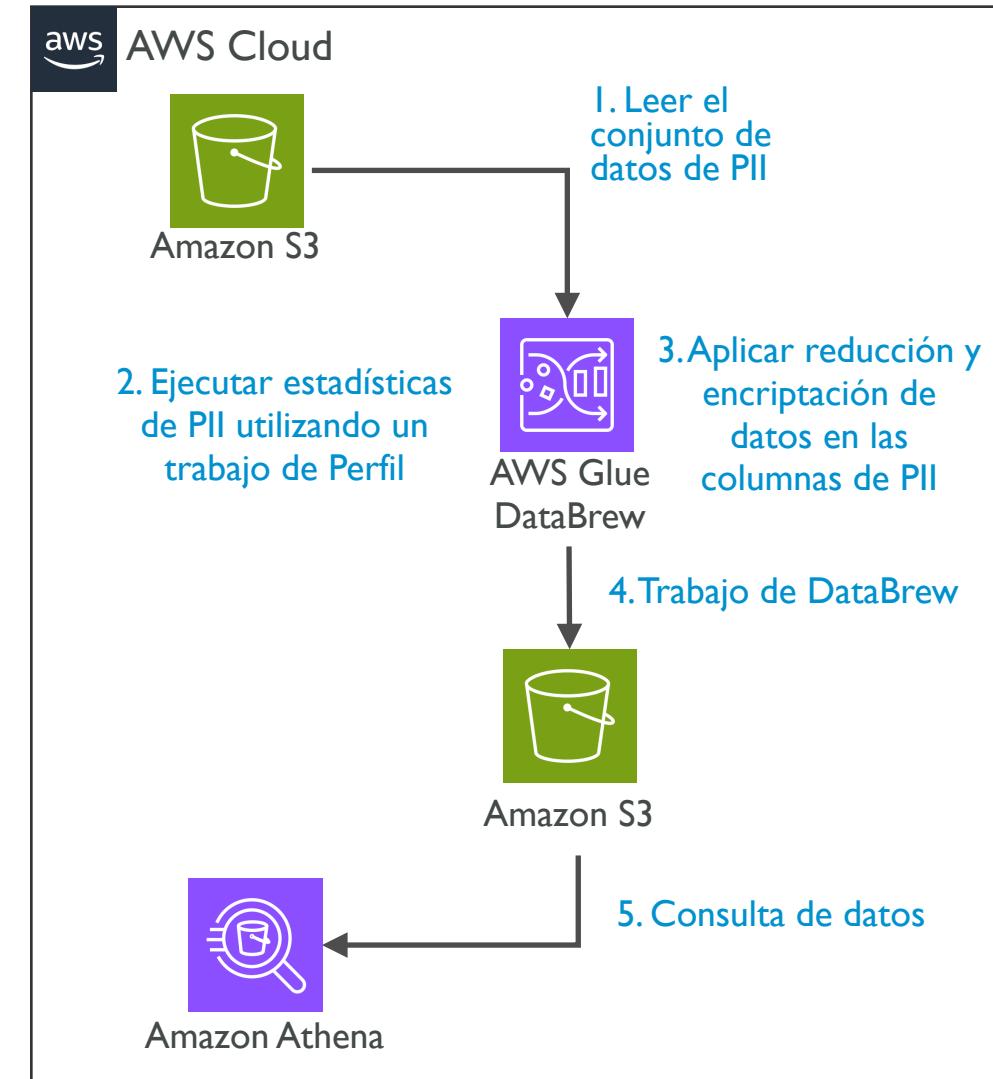
1. Transformaciones visuales (más de 250)
2. Recetas de transformación
3. Reglas de calidad de datos
4. Crear conjuntos de datos con SQL personalizado desde Amazon Redshift y Snowflake
5. Integración y seguridad
  - Puede integrarse con KMS (solo con claves maestras del cliente)
  - SSL en tránsito
  - IAM puede restringir quién puede hacer qué
  - CloudWatch y CloudTrail



# Manejo de Información Personalmente Identificable (PII) en Transformaciones de DataBrew

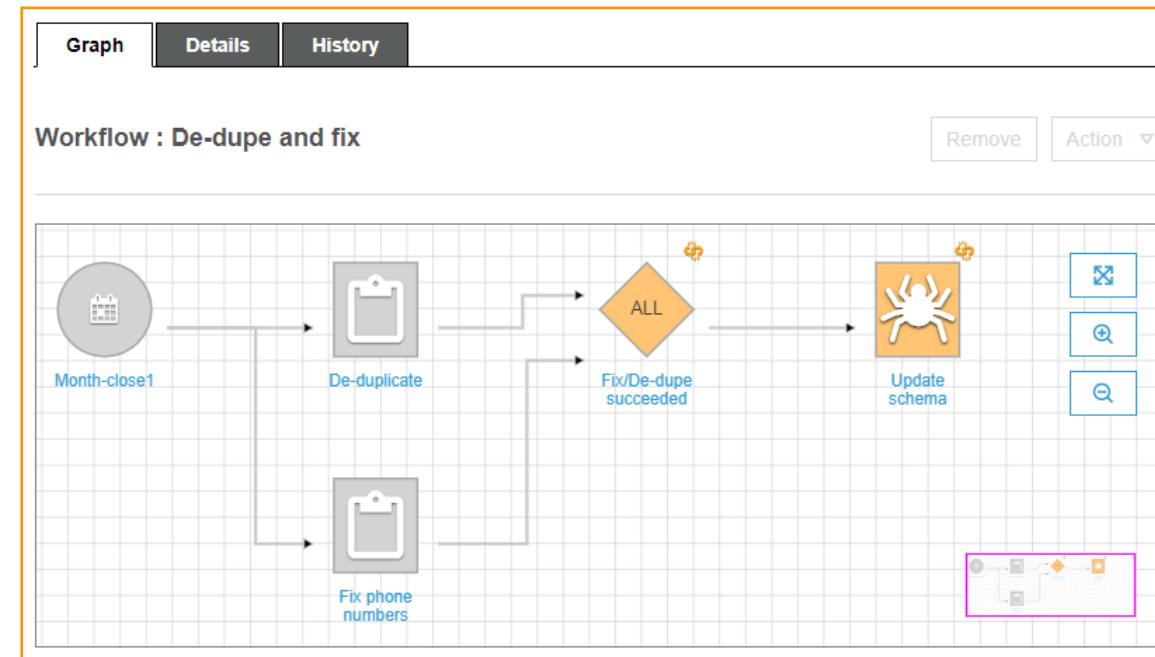
AWS Glue DataBrew proporciona diversas técnicas para manejar y proteger **Información Personalmente Identificable (PII)** durante las transformaciones de datos

1. Habilitar estadísticas de PII
2. Sustitución (REPLACE\_WITH\_RANDOM...)
3. Mezcla (SHUFFLE\_ROWS)
4. Encriptación determinística (DETERMINISTIC\_ENCRYPT)
5. Encriptación probabilística (ENCRYPT)
6. Desencriptación (DECRYPT)
7. Anulación o eliminación (DELETE)
8. Enmascaramiento (MASK\_CUSTOM, \_DATE, \_DELIMITER, \_RANGE)
9. Hashing (CRYPTOGRAPHIC\_HASH)



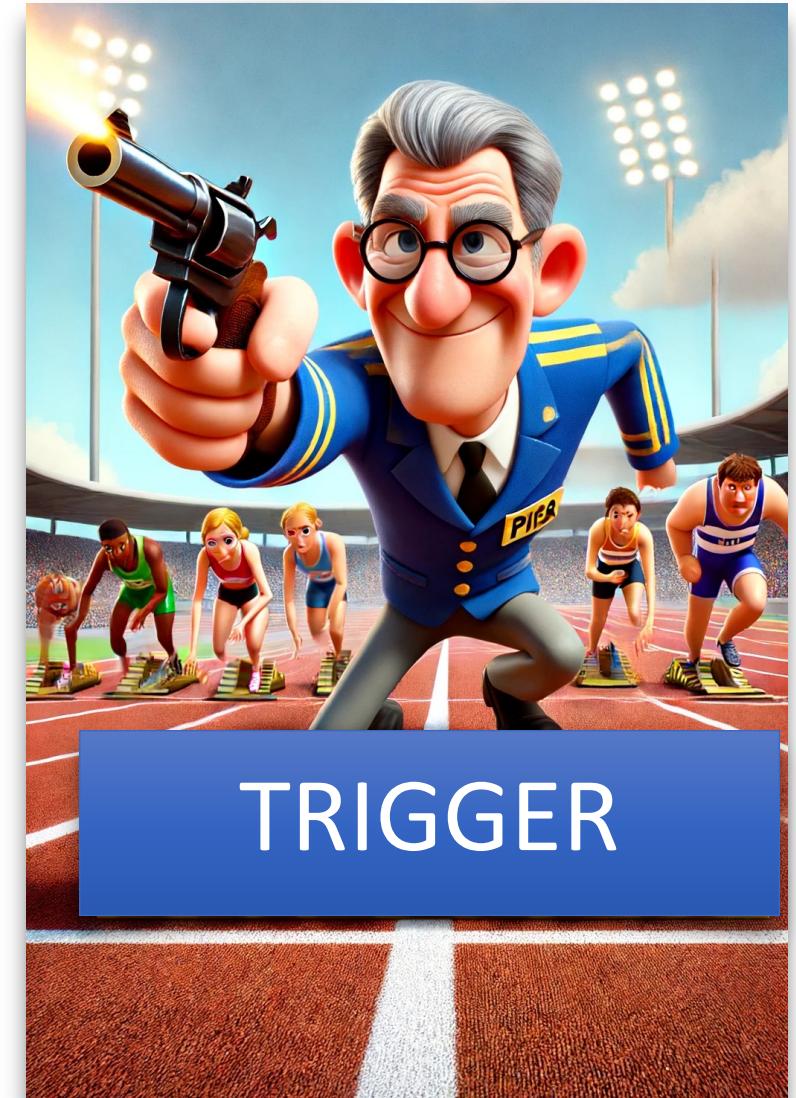
# Flujos de trabajo de AWS Glue

- Diseña procesos ETL con múltiples trabajos y múltiples crawlers que se ejecutan juntos
- Crea desde un modelo (blueprint) de AWS Glue, desde la consola o la API
- Esto es solo para orquestar operaciones ETL complejas utilizando Glue



# Triggers de flujo de trabajo de AWS Glue

- **Disparadores dentro de los flujos de trabajo** inician trabajos o crawlers
  - O pueden activarse cuando los trabajos o crawlers se completan
- **Programación** (Basado en una expresión cron)
- **A demanda**
- **Eventos de EventBridge**
  - Por ejemplo, la llegada de un nuevo objeto en S3
  - Inicia en un solo evento o en un lote de eventos
  - Condiciones opcionales para el lote
    - Tamaño del lote (número de eventos)
    - Ventana del lote (dentro de X segundos, el valor predeterminado es 15 minutos)



# AWS Lake Formation



AWS Lake Formation  
simplifica y acelera la  
**creación de un lago de**  
**datos** seguro en Amazon S3



# AWS Lake Formation



- Permite **configurar un lago de datos seguro** de manera eficiente



- Gestión de permisos granular para un acceso seguro y específico a los datos



- Soporte para transacciones ACID en múltiples tablas, optimizando el manejo y análisis de datos



- Cifrado y filtros de datos a nivel de columna, fila y celda para proteger la información sensible



- Compatible con AWS Glue, S3, Athena, y Redshift para una gestión de datos robusta

# AWS Lake Formation - Precios

- AWS Lake Formation **no tiene costo directo**, se puede usar la herramienta sin tarifas adicionales específicas por su uso
- Pero los **servicios subyacentes** tienen costos asociados



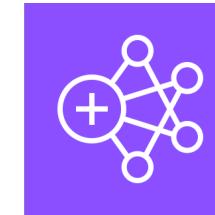
AWS Glue



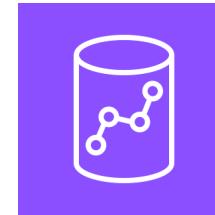
Amazon S3



Amazon Athena

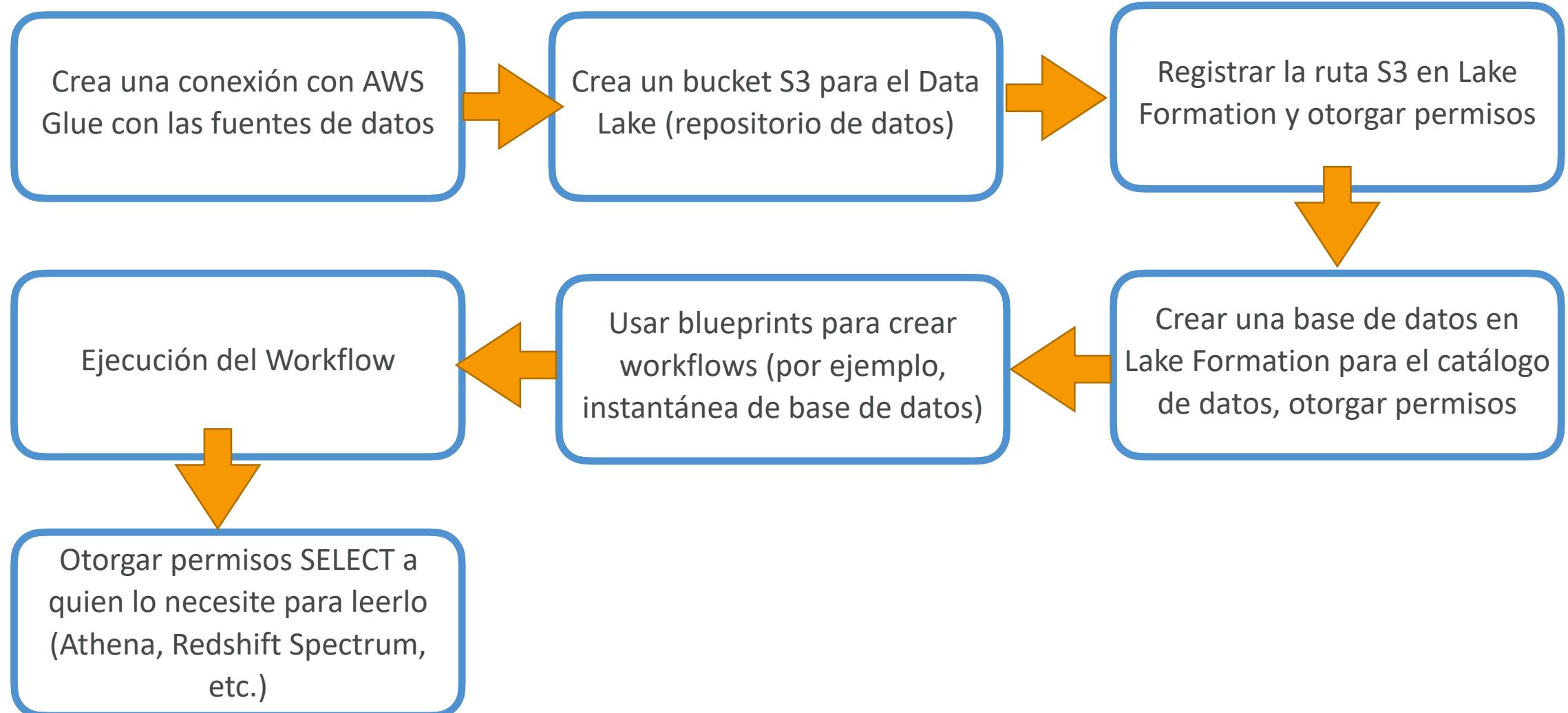


Amazon EMR



Amazon Redshift

# Construcción de un Data Lake en AWS



# AWS Lake Formation - Detalles importantes

- **Permisos entre cuentas en AWS Lake Formation**
  - El receptor debe estar configurado como administrador de un Data Lake
  - Se puede utilizar **AWS Resource Access Manager** (AWS RAM) para cuentas externas a tu organización
  - **Permisos IAM** para acceso entre cuentas
- Se necesitan permisos IAM en la clave de encriptación KMS para catálogos de datos cifrados en Lake Formation
- Se necesitan permisos IAM para crear flujos de trabajo (Workflows)



# AWS Lake Formation -Tablas gobernadas y seguridad

- Soporta **tablas gobernadas** que permiten transacciones ACID entre múltiples tablas



Nuevo tipo de tabla S3



No se puede cambiar la selección de gobernada posteriormente



Funciona con datos en streaming también (Kinesis)



Se pueden hacer consultas con Athena

- **Optimización de almacenamiento** con compactación automática
- **Control de acceso granular** con seguridad a nivel de fila y celda
  - Aplicable tanto para tablas gobernadas como S3
- Las **características mencionadas generan cargos adicionales** basados en el uso

# AWS Lake Formation - Permisos de datos



- Se pueden vincular a usuarios/roles IAM o cuentas AWS externas



- Se pueden seleccionar permisos específicos para tablas o columnas



- Se pueden usar etiquetas de políticas en bases de datos, tablas o columnas

# Integración de aplicaciones

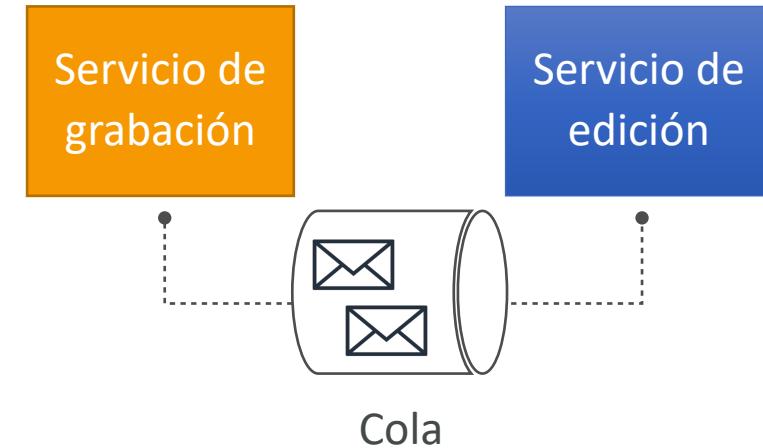
# Introducción a la mensajería

- Cuando empezamos a desplegar varias aplicaciones, es inevitable que tengan que comunicarse entre sí
- Existen dos **patrones de comunicación** entre aplicaciones

1) Comunicaciones sincrónicas  
(de aplicación a aplicación)



2) Asíncrono / basado en eventos  
(aplicación > cola > aplicación)



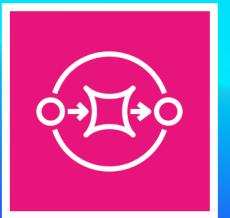
# Introducción a la mensajería

- La sincronización entre aplicaciones puede ser problemática si hay picos repentinos de tráfico
- ¿Qué pasa si de repente necesitas codificar 5000 vídeos pero normalmente son 50?
- En ese caso, es mejor **desacoplar** tus aplicaciones,
  - utilizando **SQS**: modelo de cola
  - utilizando **SNS**: modelo pub/sub
  - utilizando **Kinesis**: modelo de streaming en tiempo real
-  Estos servicios pueden escalar independientemente de nuestra aplicación

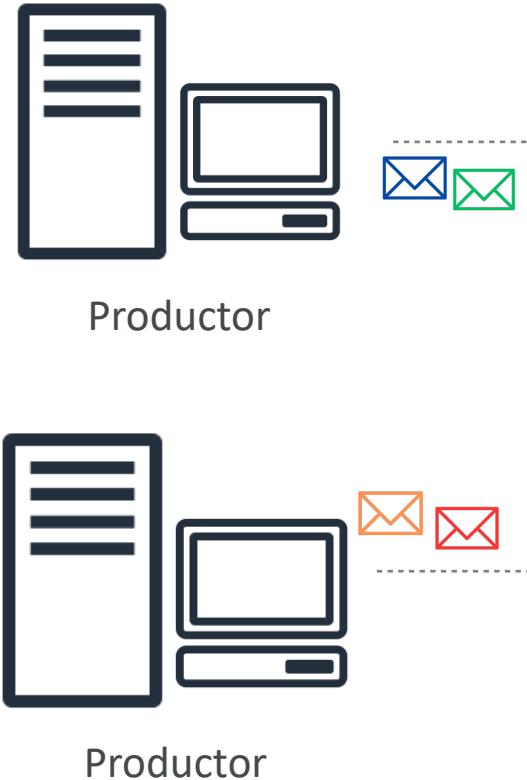


# Amazon SQS (Simple Queue Service)

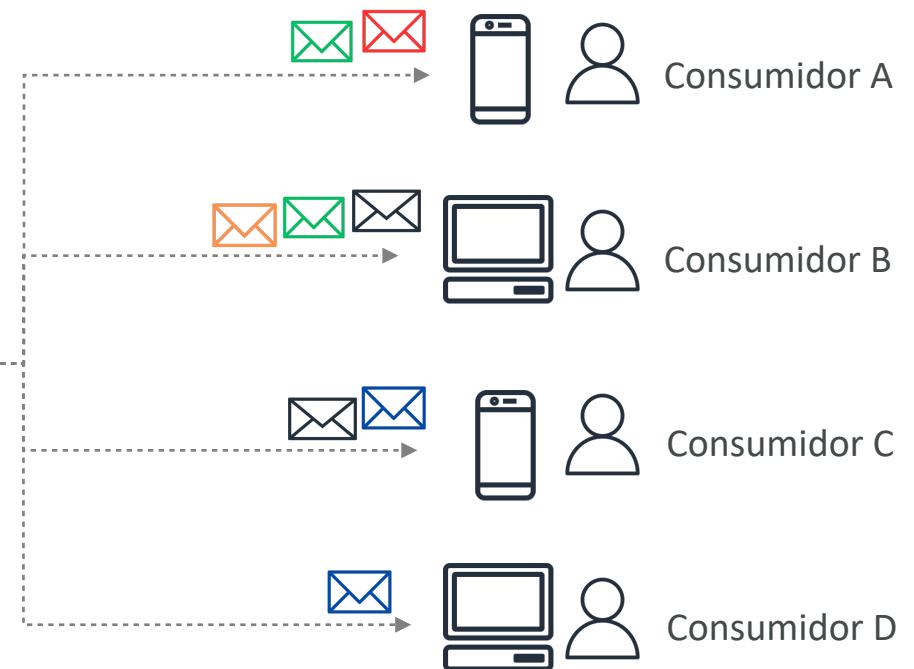
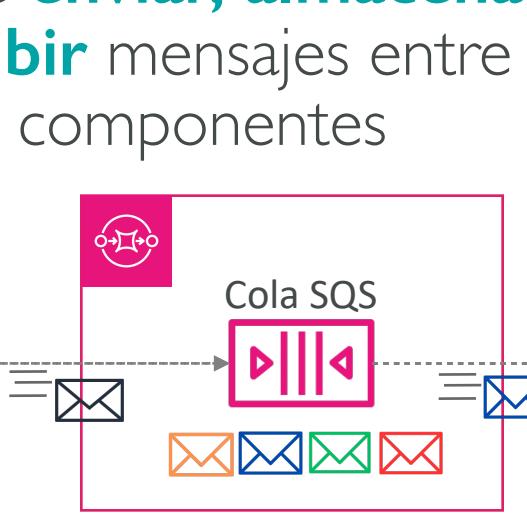
## ¿Qué es una cola?



Amazon SQS es un **servicio de mensajería** en la nube que facilita la comunicación entre componentes de software distribuidos y desacoplados

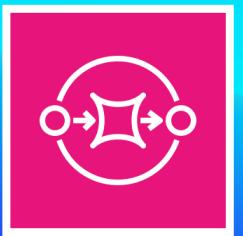


Permite **enviar, almacenar y recibir** mensajes entre componentes



# Visión general de Amazon SQS

## Cola estándar



- La oferta más antigua (más de 10 años)
- Servicio totalmente gestionado, utilizado para **desacoplar aplicaciones**
- Atributos:
  - Rendimiento ilimitado, número ilimitado de mensajes en cola
  - Retención de mensajes por defecto 4 días, máximo de 14 días
  - Baja latencia (<10 ms en publicación y recepción)
  - Limitación de 256 KB por mensaje enviado
- Puede haber mensajes duplicados (al menos una entrega, ocasionalmente)
- Puede haber mensajes fuera de orden (orden de mejor esfuerzo)

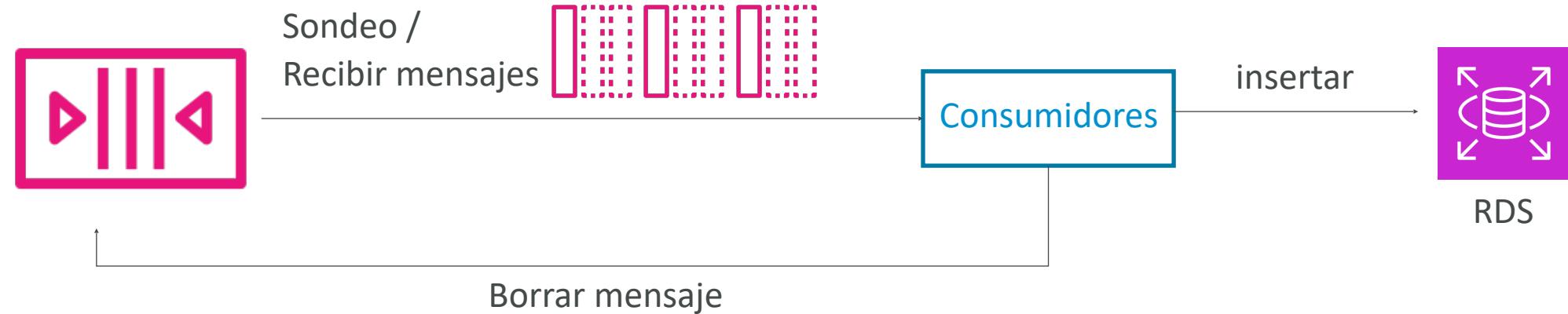
# SQS - Producción de mensajes

- El mensaje se **conserva** en SQS hasta que un consumidor lo elimina
- Retención del mensaje: por defecto 4 días, hasta 14 días (como máximo)
- Ejemplo: enviar un vídeo para ser editado
  - Id de vídeo
  - Id de curso
  - Los atributos que quieras
- Enviar los mensajes utilizando la API SendMessage

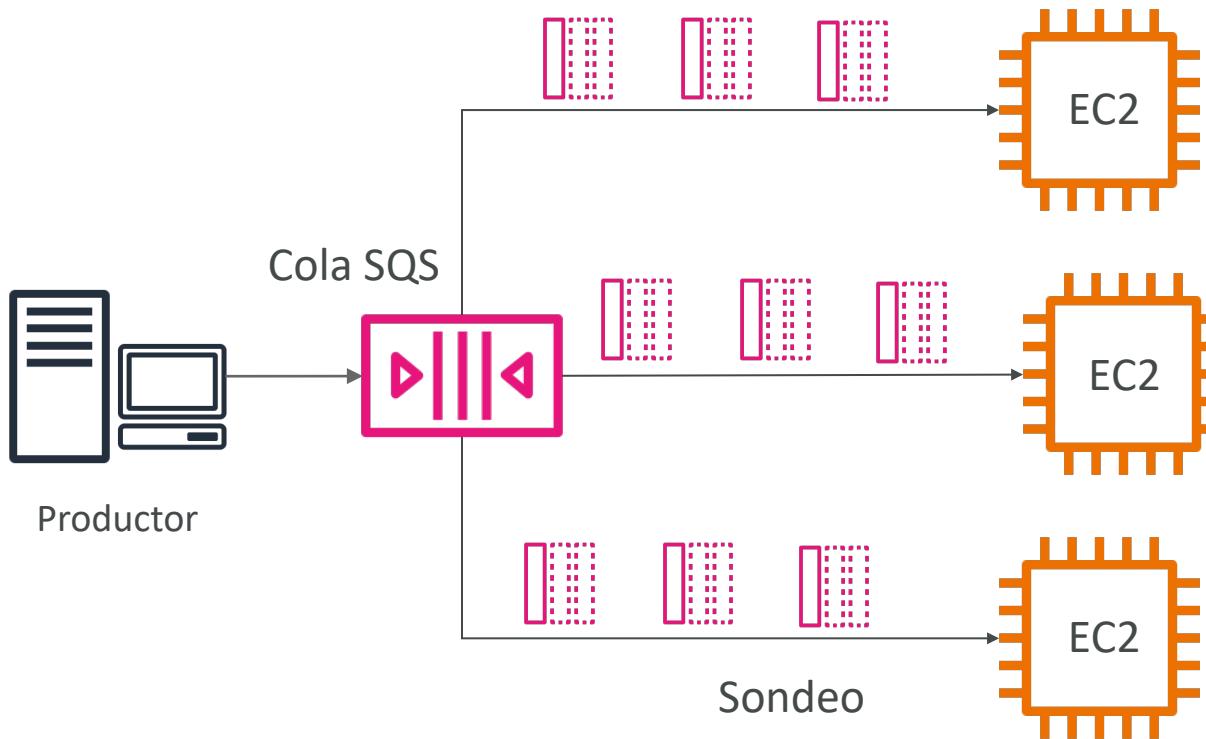


# SQS - Consumir mensajes

- **Consumidores:** ejecutándose en instancias EC2, servidores o AWS Lambda, etc
- **Sondeo (encuestas):** SQS en busca de mensajes (recibir hasta 10 mensajes a la vez)
- **Procesar los mensajes:** por ejemplo insertar el mensaje en una base de datos RDS
- Eliminar los mensajes utilizando la API DeleteMessage



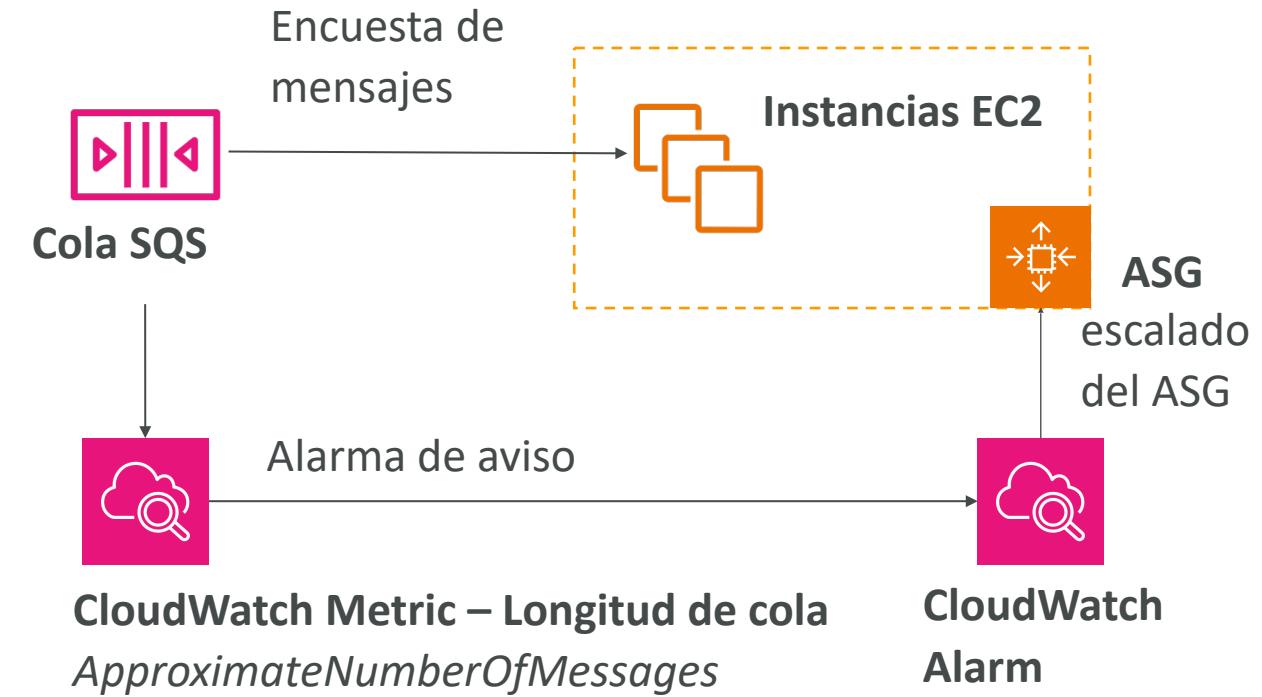
# SQS - Consumidores de múltiples instancias EC2



- Los consumidores reciben y procesan los mensajes en paralelo
- Al menos una entrega
- Ordenación de mensajes al mejor esfuerzo
- Los consumidores borran los mensajes después de procesarlos
- Podemos escalar los consumidores horizontalmente para mejorar el rendimiento del procesamiento

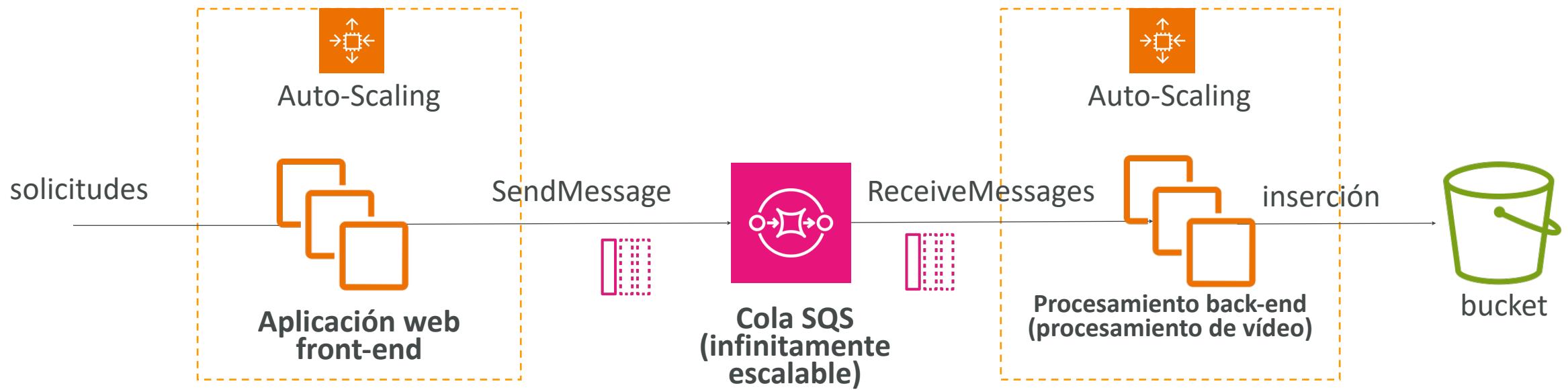
# SQS con Auto Scaling Group (ASG)

- **Cola SQS:** Recibe y almacena mensajes que necesitan ser procesados
- **Encuesta de mensajes:** Las instancias EC2 recuperan mensajes de la cola SQS para su procesamiento
- **CloudWatch Metric:** Monitorea la longitud de la cola usando la métrica `ApproximateNumberOfMessages`
- **CloudWatch Alarm:** Se activa una alarma basada en la longitud de la cola
- **Grupo de autoescalamiento (ASG):** La alarma desencadena el escalado automático del ASG para agregar o quitar instancias EC2 según la necesidad de procesamiento



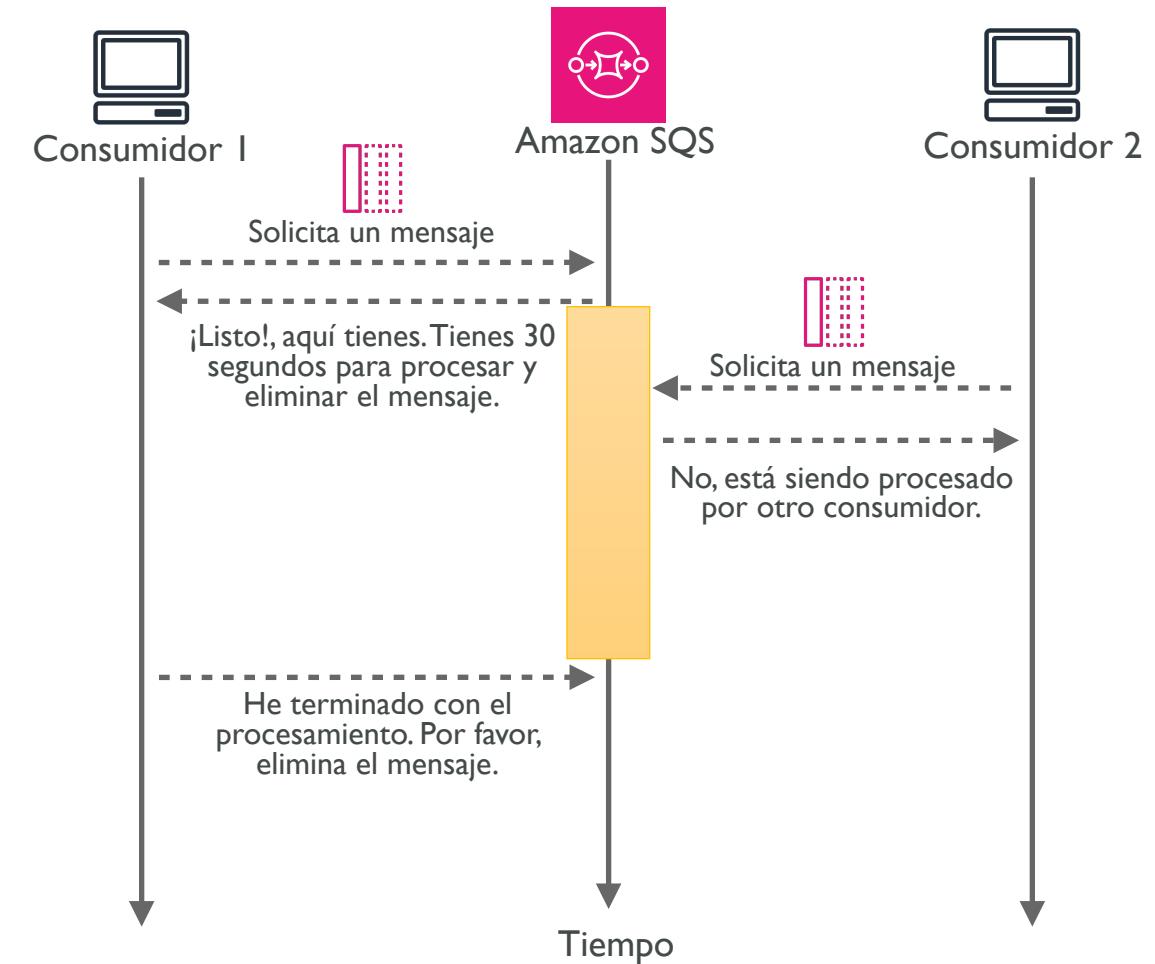
# SQS para desacoplar los niveles de aplicación

- **Front-end:** Recibe solicitudes de los usuarios y envía mensajes a la cola SQS
- **Cola SQS:** Actúa como intermediario que desacopla el front-end del back-end, almacenando mensajes hasta que el back-end los procese
- **Back-end:** Recupera mensajes de la cola SQS y realiza el procesamiento necesario
- **Bucket S3:** Almacena los resultados del procesamiento realizado por el back-end



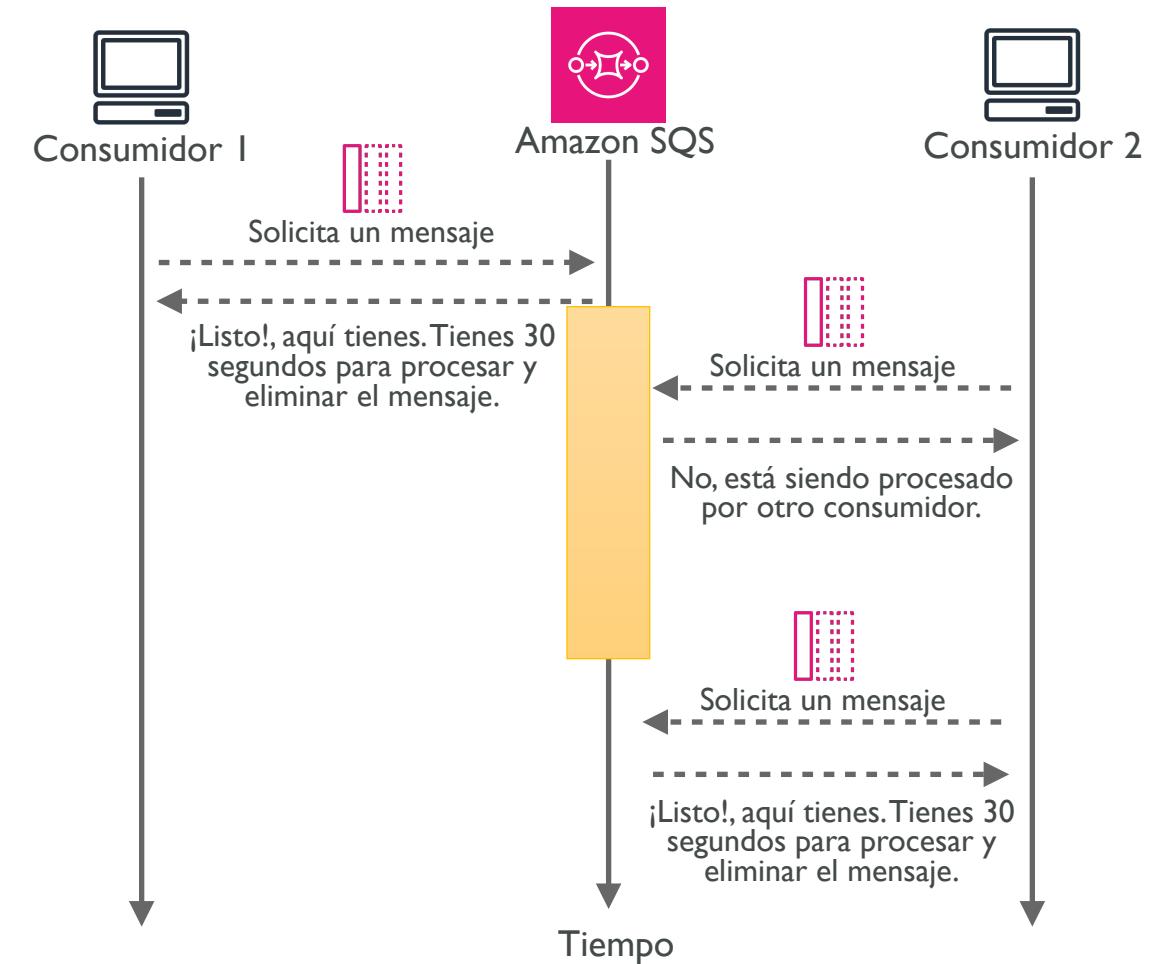
# SQS - Tiempo de espera de visibilidad de mensajes

- Después de que un consumidor sondee un mensaje, éste se vuelve **invisible** para los demás consumidores
- Por defecto, el "tiempo de visibilidad del mensaje" **es de 30 segundos**
- Esto significa que el mensaje tiene 30 segundos para ser procesado
- Una vez transcurrido el tiempo de espera, el mensaje es "visible" en SQS



# SQS -Tiempo de espera de visibilidad de mensajes

- Si un mensaje no se procesa dentro del tiempo de visibilidad, se procesará **dos veces**
- El consumidor puede llamar a la API **ChangeMessageVisibility** para obtener más tiempo
- Si el tiempo de espera de visibilidad es alto (horas) y el consumidor se bloquea, el reprocesamiento llevará tiempo
- Si el tiempo de espera de visibilidad es demasiado bajo (segundos), puede haber duplicados



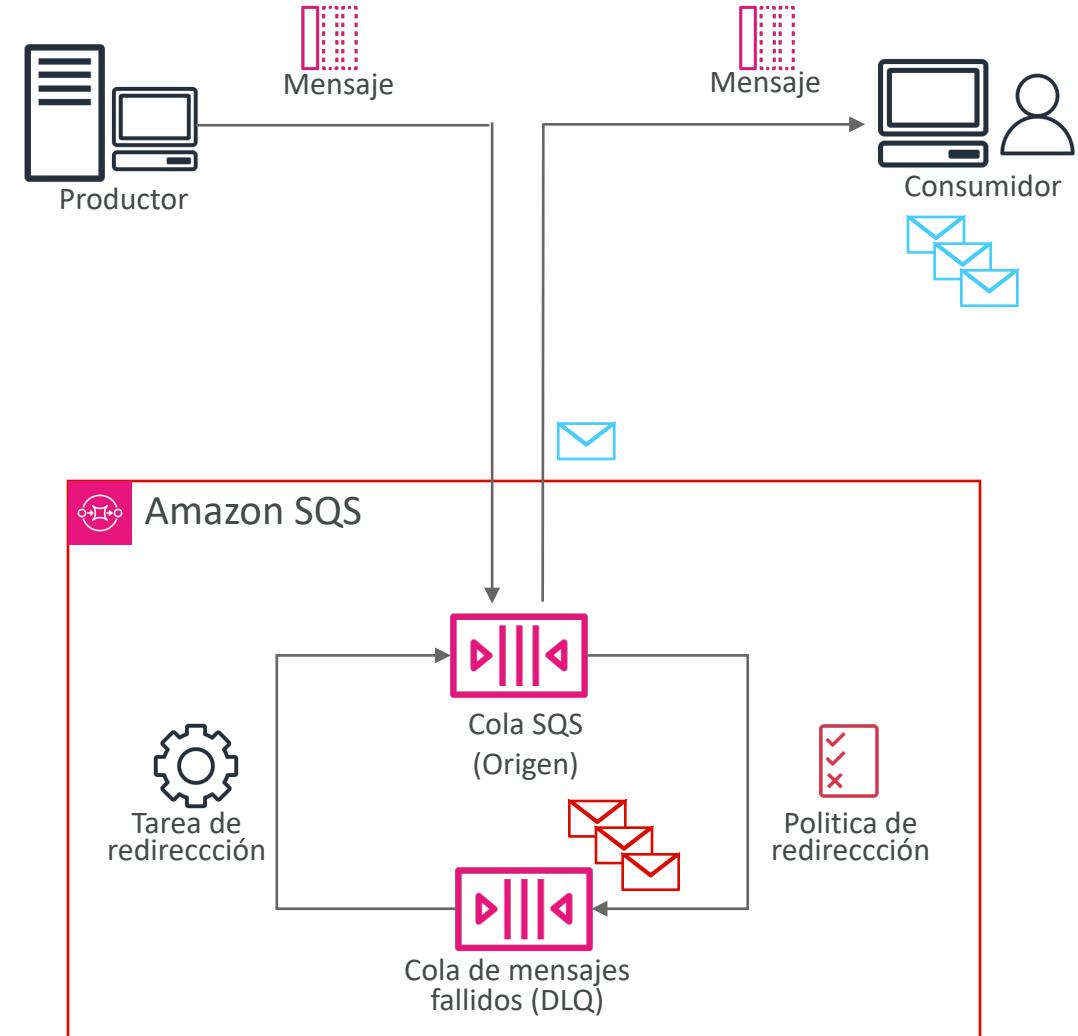
# Amazon SQS - Cola de mensajes fallidos (DLQ)

- Si un consumidor no procesa un mensaje dentro del tiempo de espera de visibilidad... ¡el mensaje vuelve a la cola!
  - Podemos **establecer un umbral** de cuántas veces puede volver un mensaje a la cola
  - Una vez superado el umbral de **MaximumReceives**, el mensaje pasa a una cola de mensajes fallidos (DLQ)
  - ¡Útil para depurar!
- 
- **DLQ de una cola FIFO debe ser también una cola FIFO**
  - **DLQ de una cola estándar también debe ser una cola estándar**
  - Asegúrate de procesar los mensajes de la DLQ antes de que caduquen:
    - Conviene fijar una retención de 14 días en la DLQ

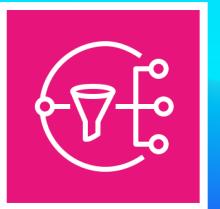


# SQS DLQ - Redirigir al origen

- Cuando nuestro código esté arreglado, podremos volver a conducir los mensajes de la DLQ a la cola de origen (o a cualquier otra cola) por lotes sin escribir código personalizado
- En la política de redirección se definen cuándo y cómo se deben transferir los mensajes a la DLQ
- La tarea de redirección es esencial para que los mensajes que requieren atención adicional no se pierdan ni interfieran con los procesos normales

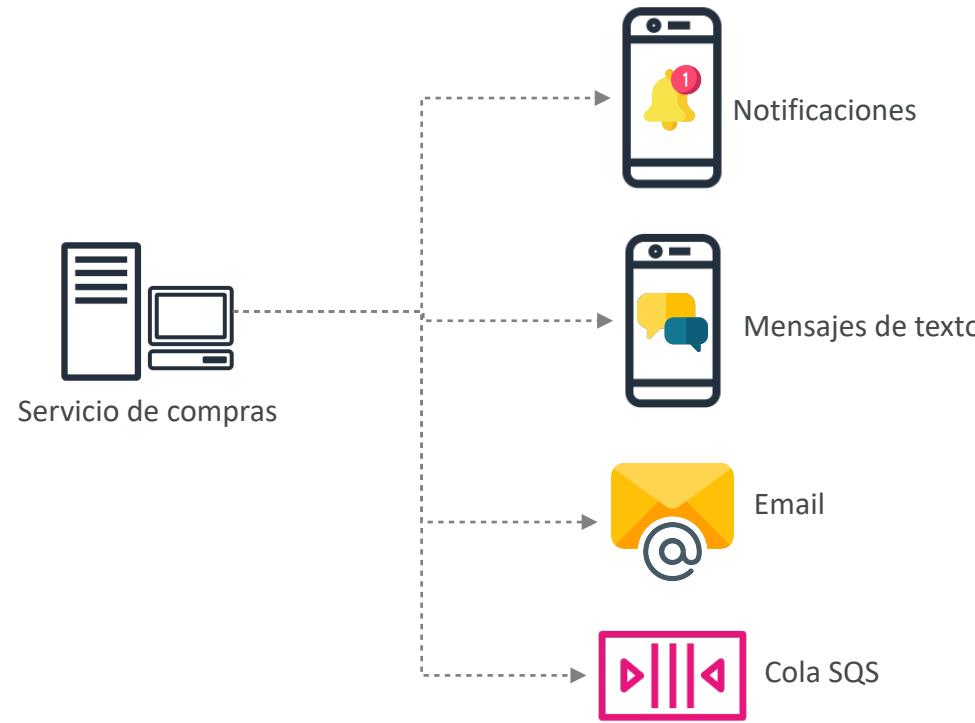


# Amazon SNS

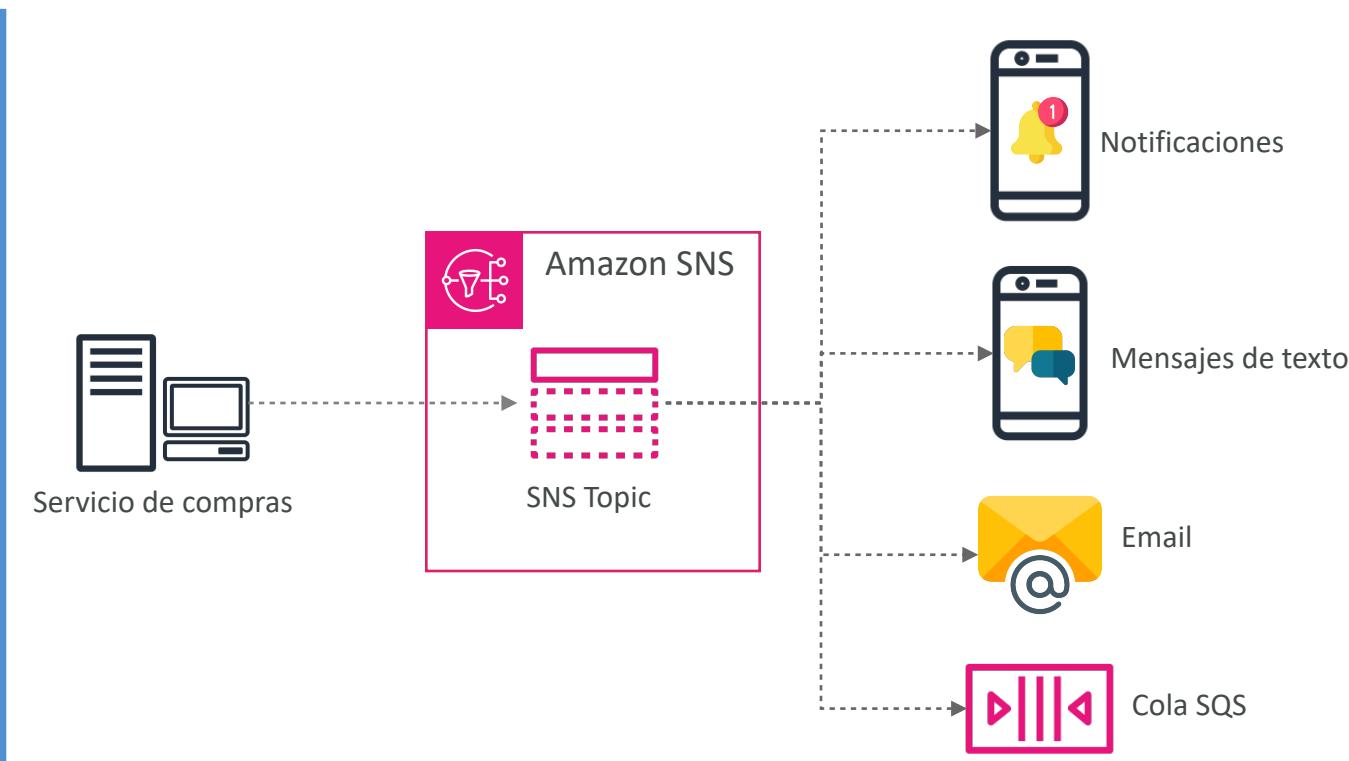


- ¿Y si quieres enviar un mensaje a muchos destinatarios?

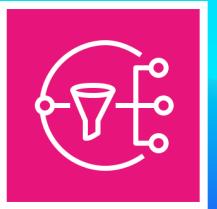
## Integración directa



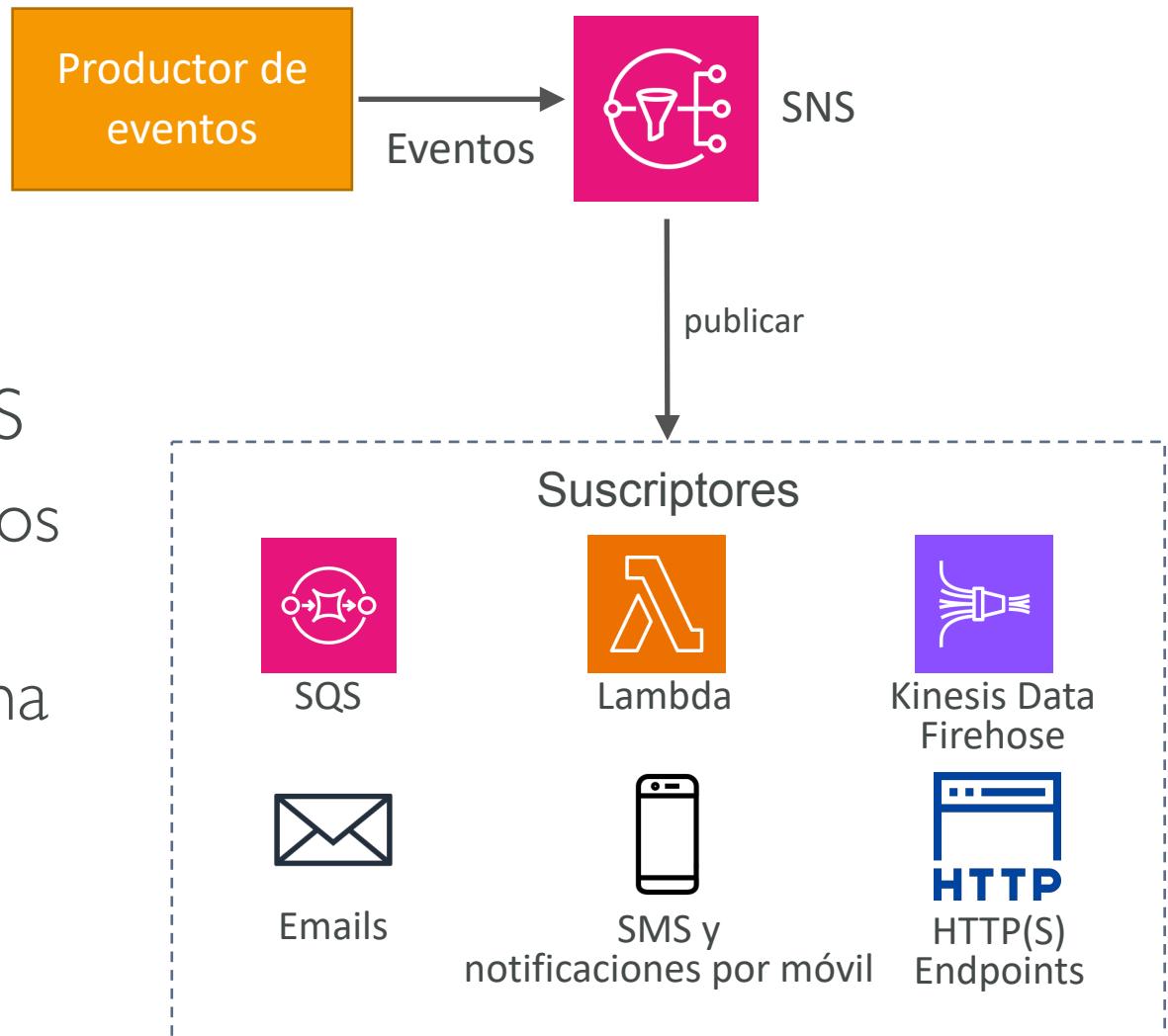
## Pub/Sub



# Amazon SNS

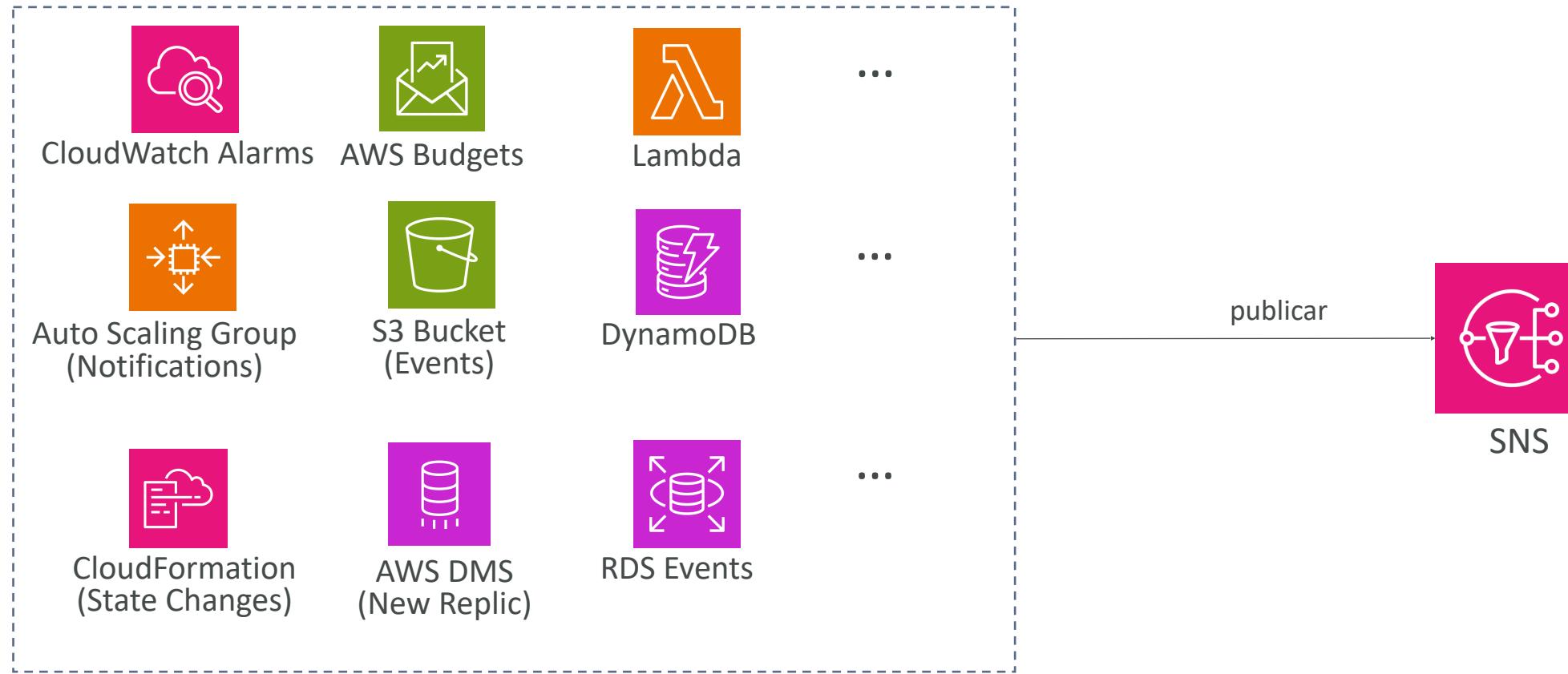


- El productor de eventos sólo envía mensajes a un tema SNS
- Tantos receptores de eventos (suscriptores) como queramos para escuchar las notificaciones del tema SNS
- Cada suscriptor al tema recibirá todos los mensajes (función para filtrar mensajes)
- Hasta 12.500.000 suscripciones por tema
- Límite de 100.000 temas



# SNS se integra con muchos servicios de AWS

- Muchos servicios de AWS pueden enviar datos directamente a SNS para notificaciones



# Amazon SNS - Cómo publicar

- **Publicación de temas (mediante el SDK)**

- Crear un tema (topic)
- Crea una suscripción (o varias)
- Publicar mensajes en el tema

- **Publicación directa (para aplicaciones móviles SDK)**

- Crear una aplicación de plataforma
- Crear un punto final de plataforma
- Publicar en el punto final de la plataforma
- Funciona con Google GCM, Apple APNS ...



# Amazon SNS - Seguridad

- **Cifrado:**

- Cifrado en vuelo mediante API HTTPS
- Cifrado en reposo mediante claves KMS
- Cifrado del lado del cliente si el cliente desea realizar el cifrado/ descifrado por sí mismo



- **Controles de acceso:** Políticas IAM para regular el acceso a la API SNS

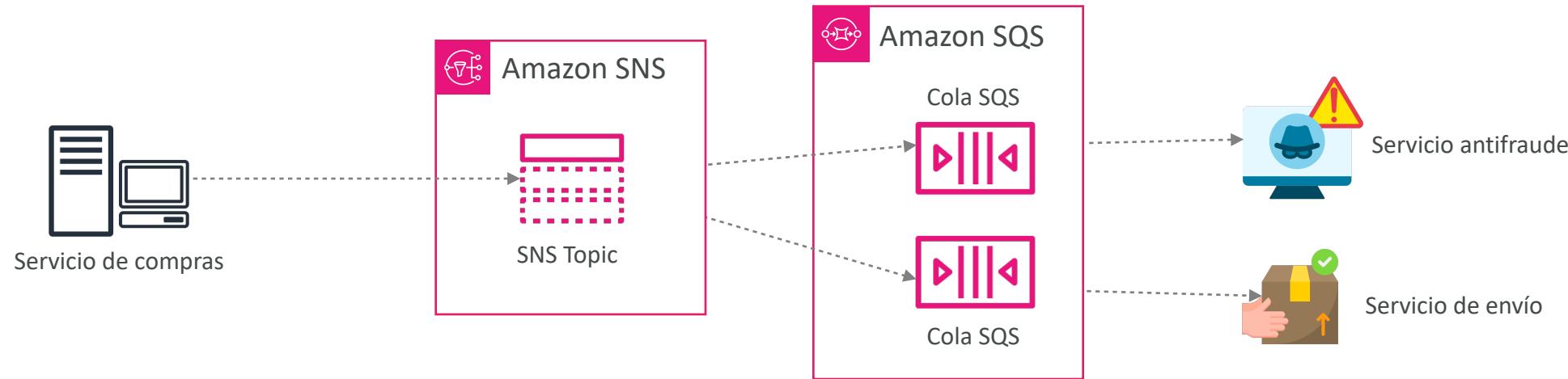


- **Políticas de acceso SNS** (similares a las políticas de bucket S3)

- Útil para el acceso entre cuentas a temas SNS
- Útil para permitir que otros servicios ( S3...) escriban en un tema SNS



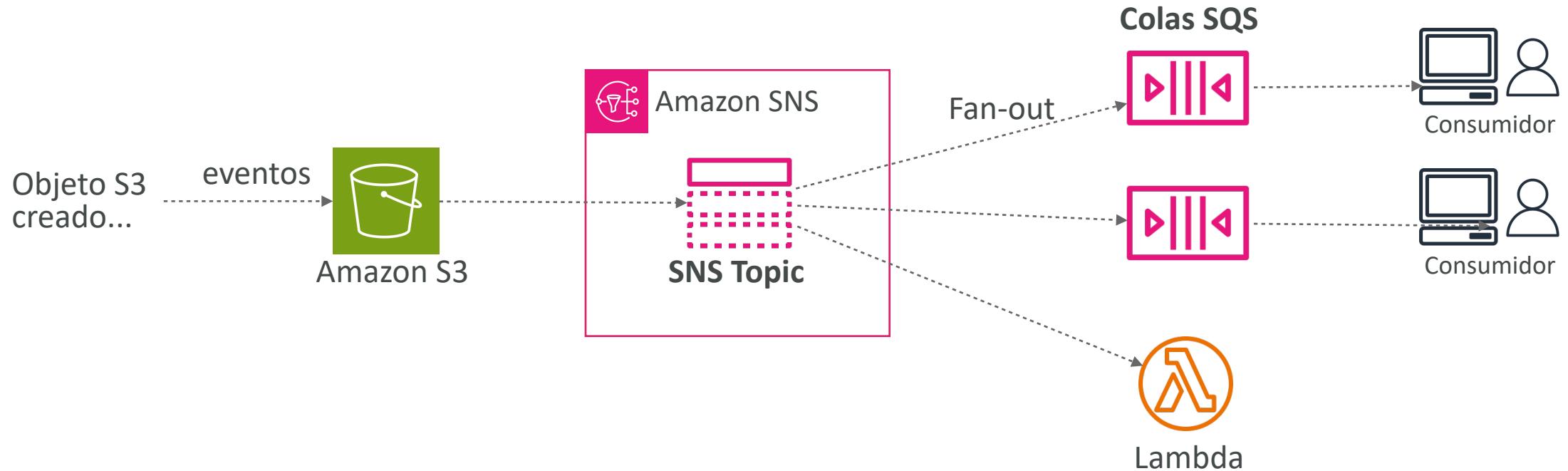
# SNS + SQS: Fan Out



- Haz el push una vez en SNS, recibe en todas las colas SQS que son suscriptores
- Totalmente desacoplado, sin pérdida de datos
- SQS permite: persistencia de datos, procesamiento diferido y reintentos de trabajo
- Posibilidad de añadir más suscriptores SQS con el tiempo
- Asegúrate de que la **política de acceso** a la cola SQS permite que SNS pueda escribir

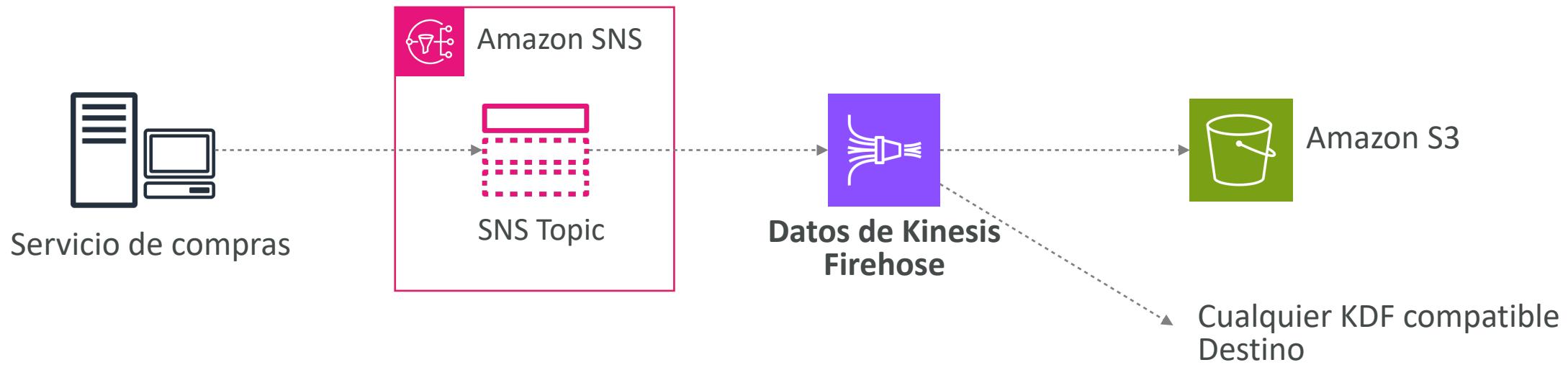
# Eventos S3 a múltiples colas

- Para la misma combinación de: **tipo de evento** (p.e. creación de objeto) y **prefijo** (p.e. imágenes/) sólo puedes tener una regla de Evento S3
- Si quieres enviar el mismo evento S3 a muchas colas SQS, utiliza fan-out



# SNS a Amazon S3 a través de Kinesis Data Firehose

- SNS puede enviar a Kinesis y por lo tanto podemos tener la siguiente arquitectura de soluciones:



# Amazon SNS - Tema (topic) FIFO

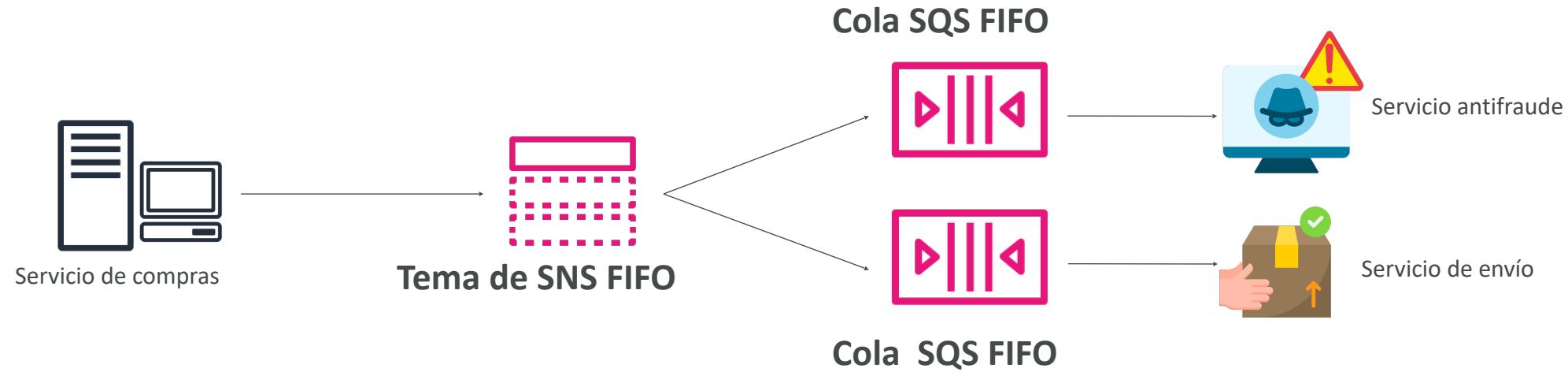
- FIFO = First In First Out (orden de los mensajes en el tema)



- Características similares a SQS FIFO:
  - **Ordenación** por ID de grupo de mensajes (se ordenan todos los mensajes del mismo grupo)
  - **Deduplicación** mediante ID de deduplicación o deduplicación basada en contenido
- Sólo puede tener colas SQS FIFO como suscriptores
- Rendimiento limitado (el mismo que SQS FIFO)

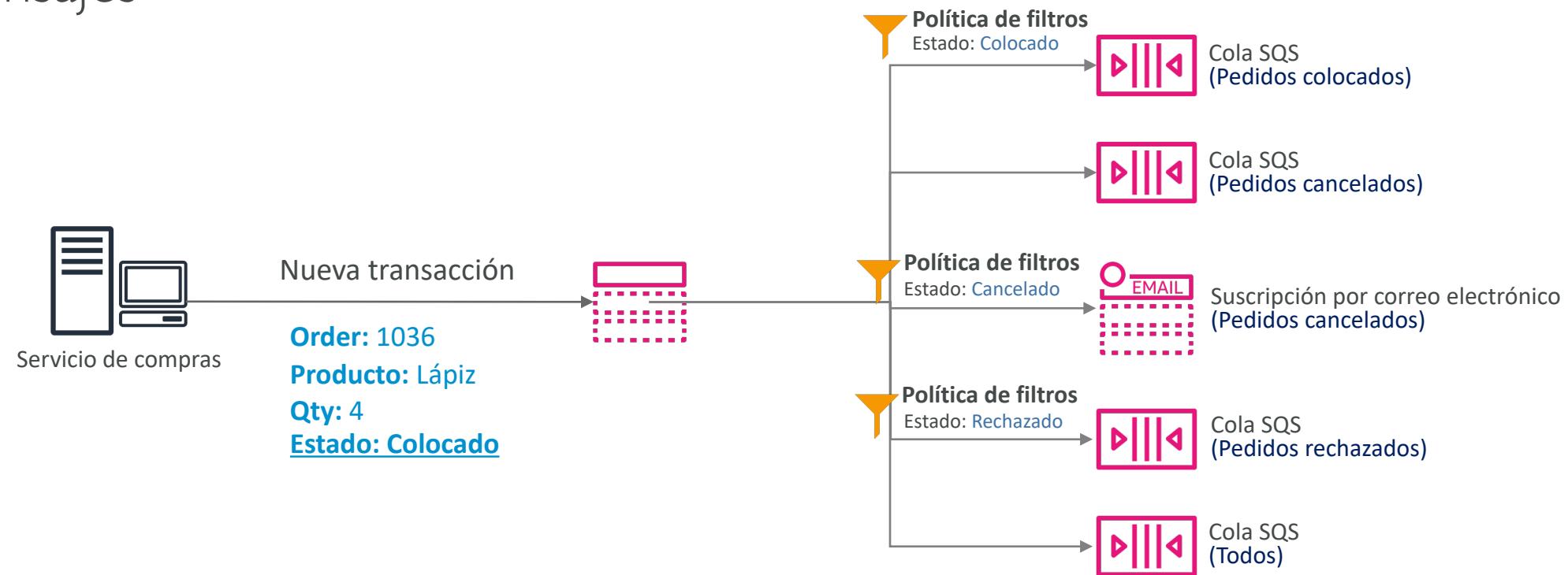
# SNS FIFO + SQS FIFO: Fan Out

- En caso de que necesites fan-out + ordenación + deduplicación



# SNS - Filtrado de mensajes

- Política JSON utilizada para filtrar los mensajes enviados a las suscripciones del tema SNS
- Si una suscripción no tiene una política de filtrado, recibe todos los mensajes





# Amazon Kinesis

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Visión general de Kinesis

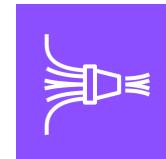


- Facilita la **recopilación**, el **procesamiento** y el **análisis** de datos de flujo continuo en tiempo real
- Ingesta de datos en tiempo real como: Registros de aplicaciones, métricas, secuencias de clics de sitios web, datos telemétricos de IoT...

**Kinesis Data Streams:** captura, procesa y almacena flujos de datos



**Amazon Data Firehose:** carga flujos de datos en almacenes de datos de AWS



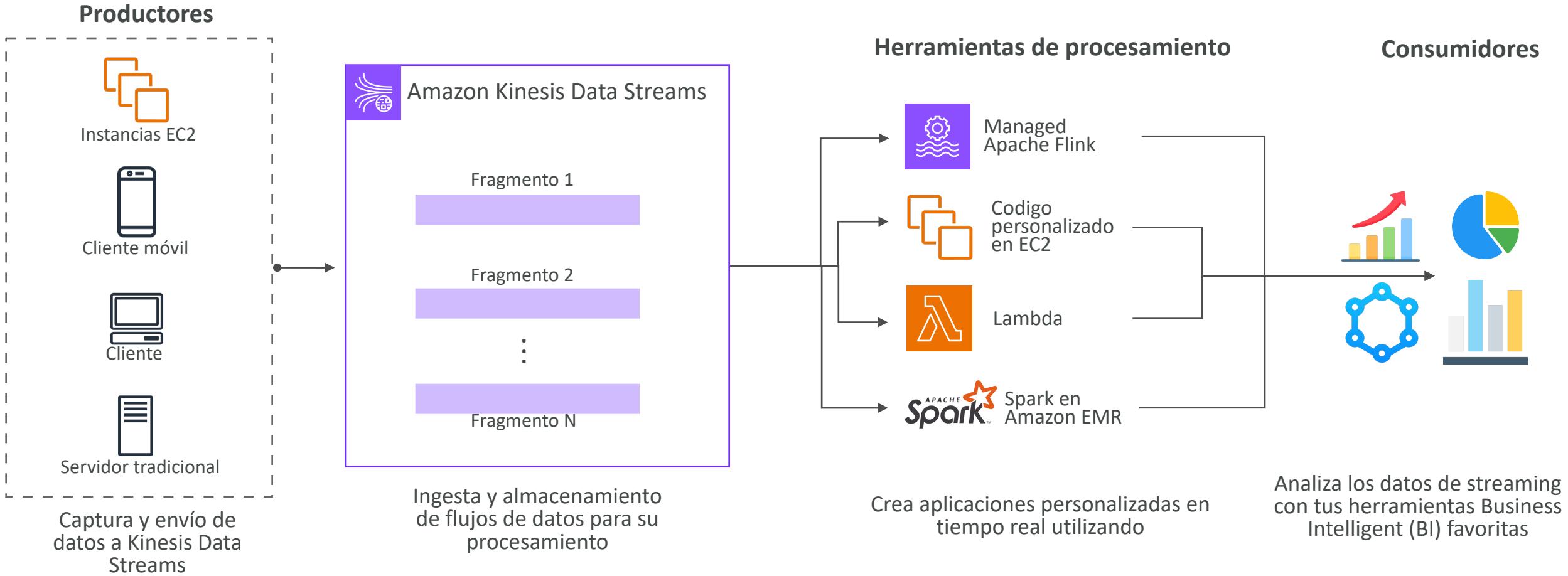
- **Managed Apache Flink:** (sucesor de Kinesis Data Analytics): analiza flujos de datos con SQL o Apache Flink



- **Kinesis Video Streams:** captura, procesa y almacena transmisiones de vídeo



# Kinesis Data Streams



# Kinesis Data Streams



- Retención entre 1 día y 365 días
  - Posibilidad de volver a procesar (reproducir) los datos
  - Una vez que los datos se insertan en Kinesis, no pueden borrarse (inmutabilidad)
  - Los datos que comparten la misma partición van al mismo fragmento (ordenación)
- 
- **Productores:**
    - SDK de AWS, biblioteca de productores de Kinesis (KPL), agente de Kinesis
  - **Consumidores:**
    - Escribe el tuyo propio: Kinesis Client Library (KCL), AWS SDK
    - Administrados: AWS Lambda, Amazon Data Firehose, Managed Apache Flink

# Kinesis Data Streams - Modos de capacidad

1

- **Modo aprovisionado**

- Tú eliges el número de shards aprovisionados, escala manualmente o usando API
- Cada fragmento recibe 1 MB/s (o 1000 registros por segundo)
- Cada fragmento recibe 2 MB/s de salida (consumo en fan-out clásico o mejorado)
- Se paga por cada fragmento aprovisionado por hora

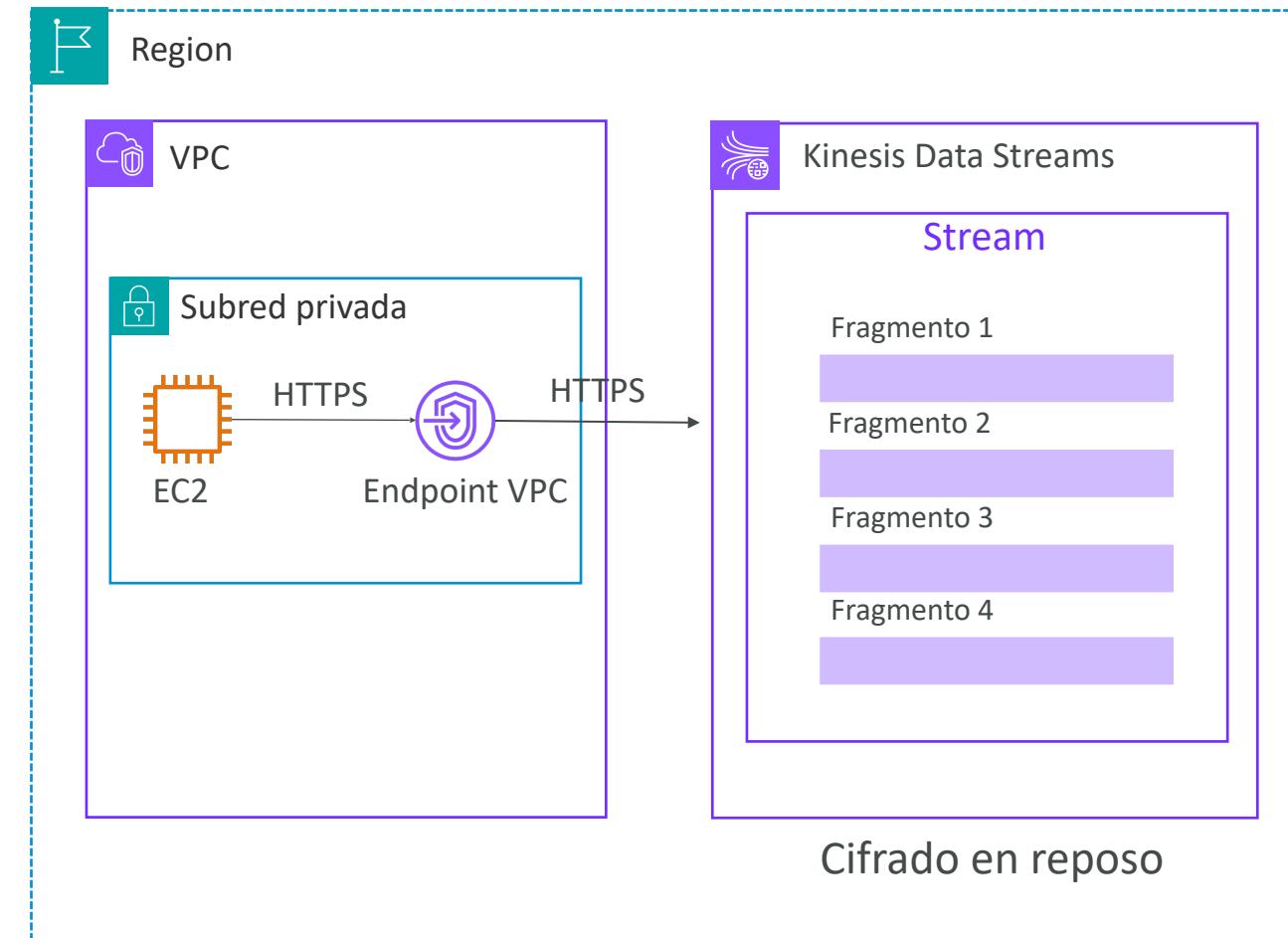
# Kinesis Data Streams - Modos de capacidad

2

- **Modo bajo demanda**
- No es necesario aprovisionar ni gestionar la capacidad
- Capacidad provisionada por defecto (4 MB/s de entrada o 4000 registros por segundo)
- Escala automáticamente en función del pico de rendimiento observado durante los últimos 30 días
- Pago por flujo por hora y entrada/salida de datos por GB

# Seguridad de Kinesis Data Streams

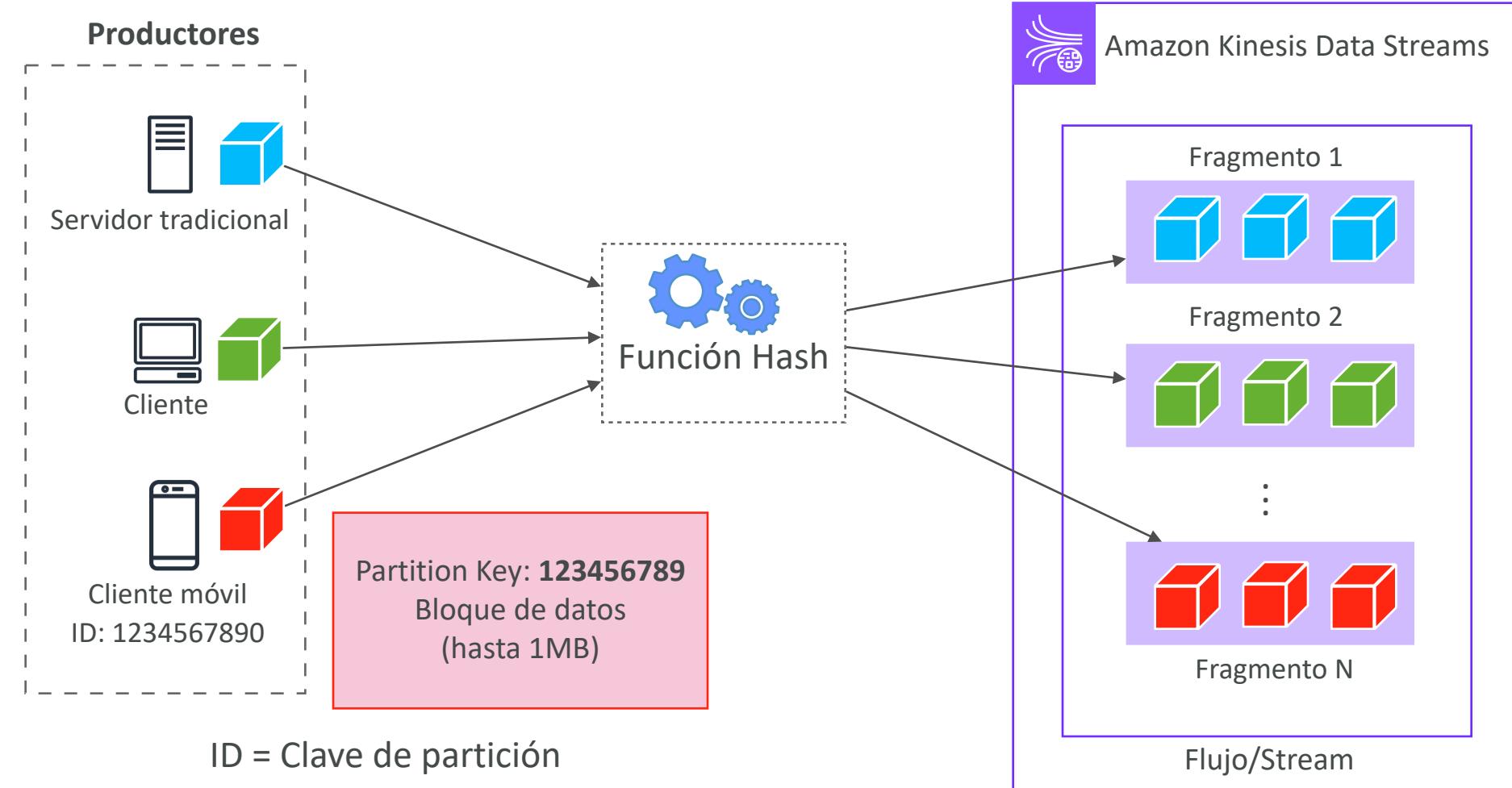
- Control de acceso / autorización mediante políticas IAM
- Cifrado en vuelo mediante endpoints HTTPS
- Cifrado en reposo mediante KMS
- Puedes implementar el cifrado/ descifrado de datos en el lado del cliente (más difícil)
- Endpoints VPC disponibles para que Kinesis acceda dentro de VPC



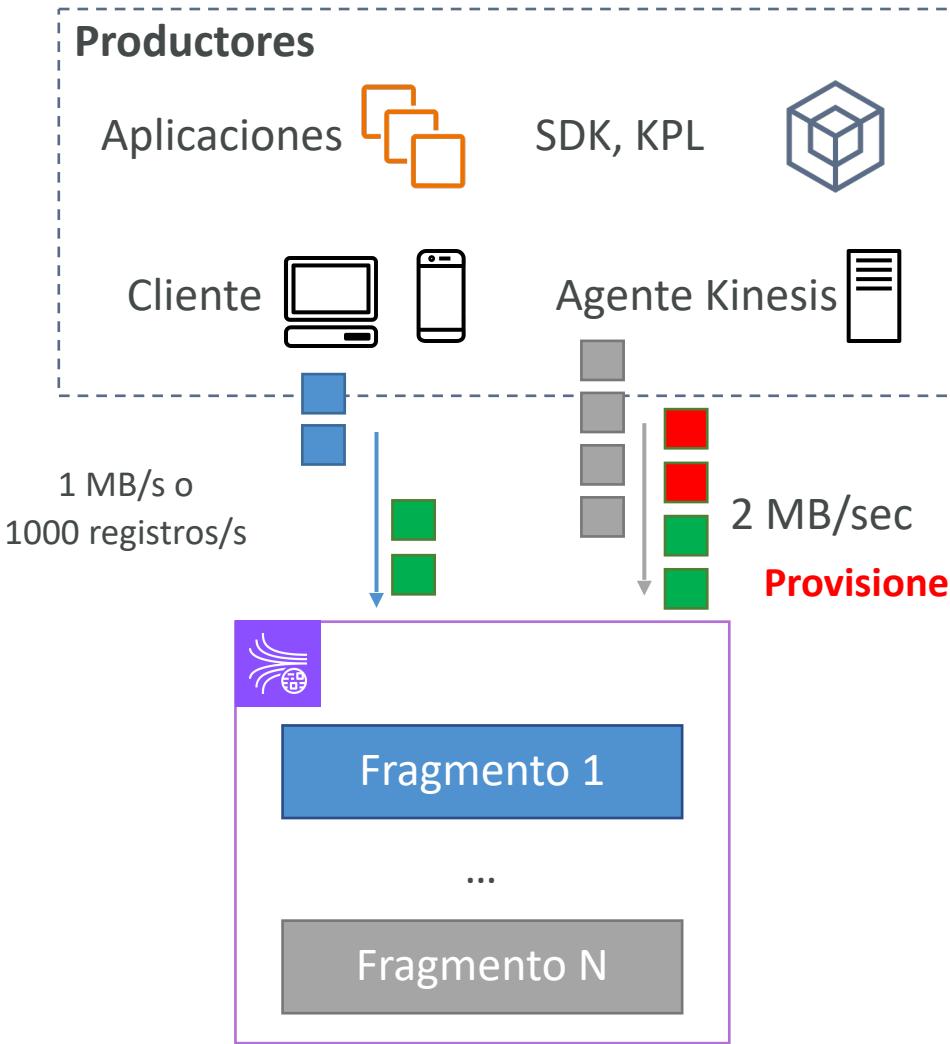
# Productores Kinesis

- Coloca registros de datos en streams de datos
- El registro de datos consta de:
  - Número de secuencia (único por clave de partición dentro del fragmento)
  - Clave de partición (debe especificarse al poner los registros en el stream)
  - Bloque de datos (hasta 1 MB)
- Productores:
  - **AWS SDK**: productor simple
  - **Biblioteca de productores Kinesis (KPL)**
  - **Agente Kinesis**: monitoriza los logs
- Velocidad de escritura: 1 MB/seg o 1000 registros/seg por fragmento
- Utiliza el procesamiento por lotes con la API PutRecords para reducir costes y aumentar el rendimiento

# Productores Kinesis



# Kinesis - ProvisionedThroughputExceeded



- El error:

**ReadProvisionedThroughputExceeded** se produce cuando Kinesis Data Streams estrangula las llamadas a GetRecords durante un periodo

## Solución:

- Utilizar clave de partición altamente distribuida
- Reintentos con backoff exponencial
- Aumentar los fragmentos (escalado)

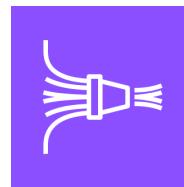
# Consumidores de Kinesis Data Streams



- AWS Lambda



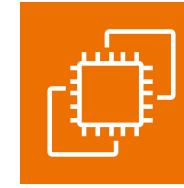
- Managed Apache Flink



- Amazon Data Firehose



- Consumidor personalizado  
(AWS SDK)



- Amazon EC2



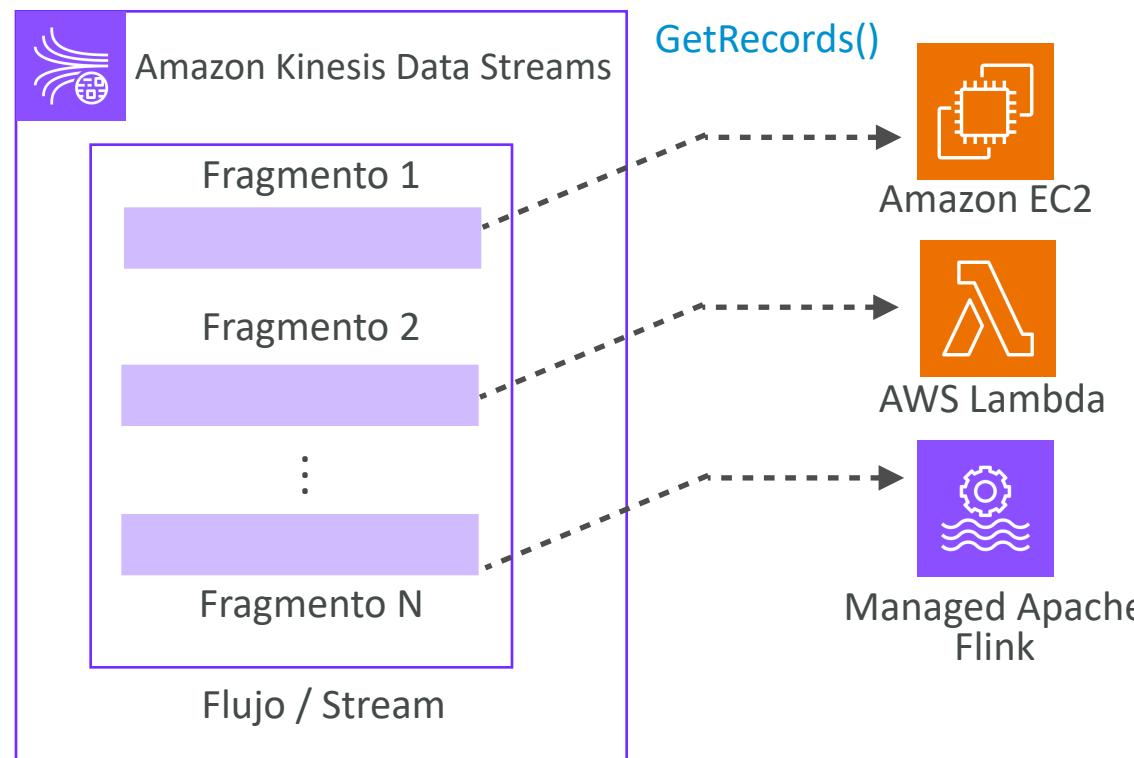
- Kinesis Client Library (KCL): biblioteca para simplificar la lectura del stream de datos



- AWS Glue

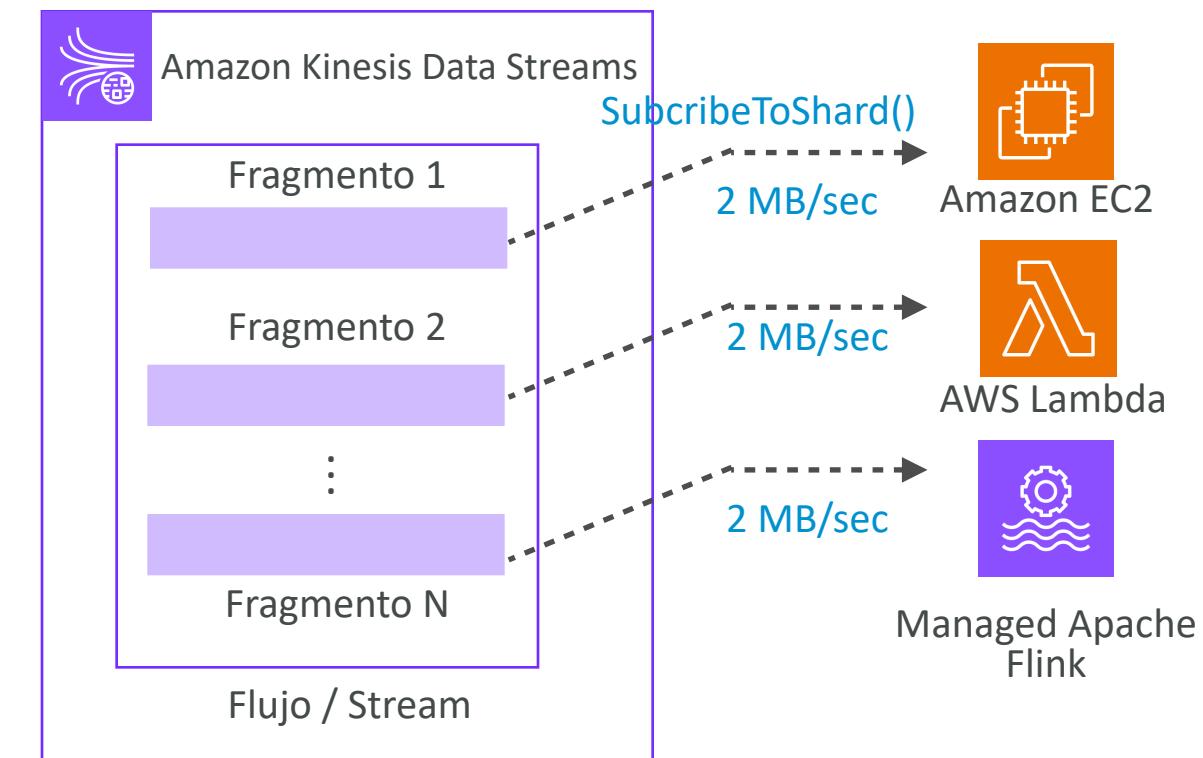
# Consumidores Kinesis - Consumidor personalizado

## Consumidor en Fan-out compartido (clásico)



2 MB/s por fragmento en todos los consumidores

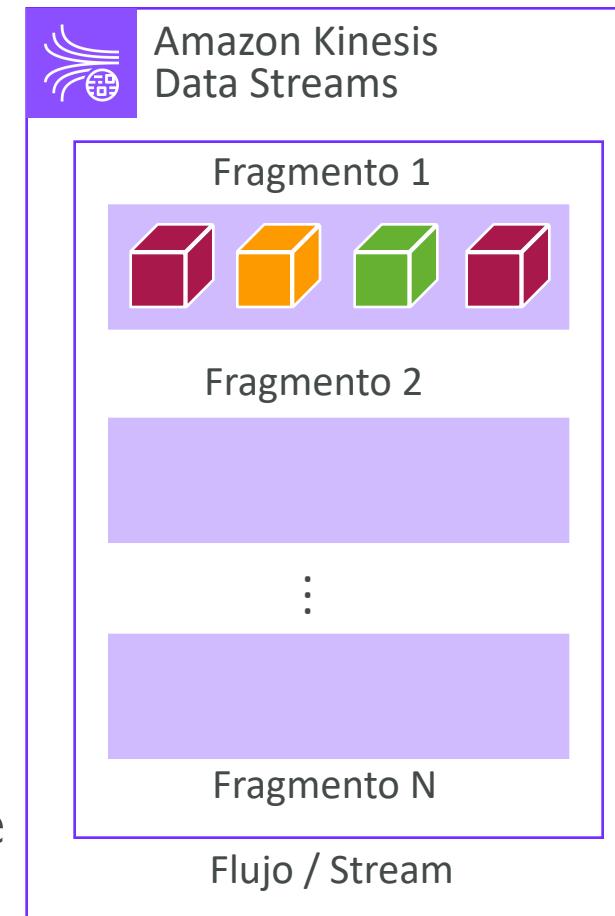
## Consumidor en Fan-out (mejorado)



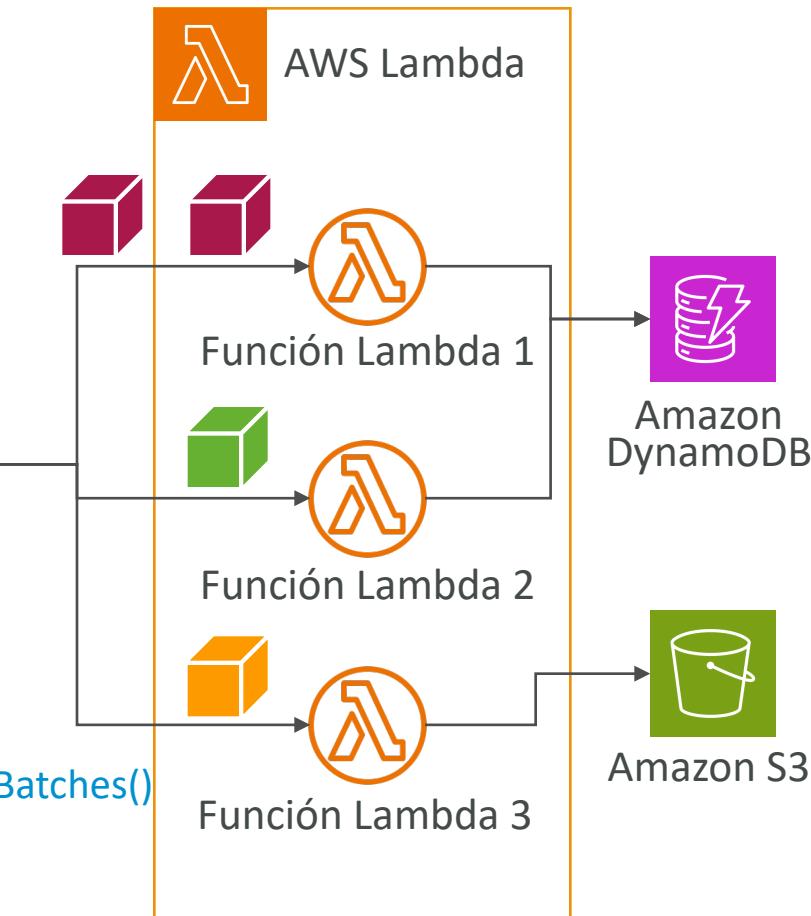
2 MB/s por consumidor por fragmento

# Consumidores Kinesis - AWS Lambda

- Soporta consumidores en Fan-Out clásico y mejorado
- Lee registros por lotes (batches)
- Puedes configurar el **tamaño del lote** y la **ventana del lote**
- Si se produce un error, Lambda reintenta hasta que tenga éxito o los datos caduquen
- Puede procesar hasta 10 lotes por fragmento simultáneamente



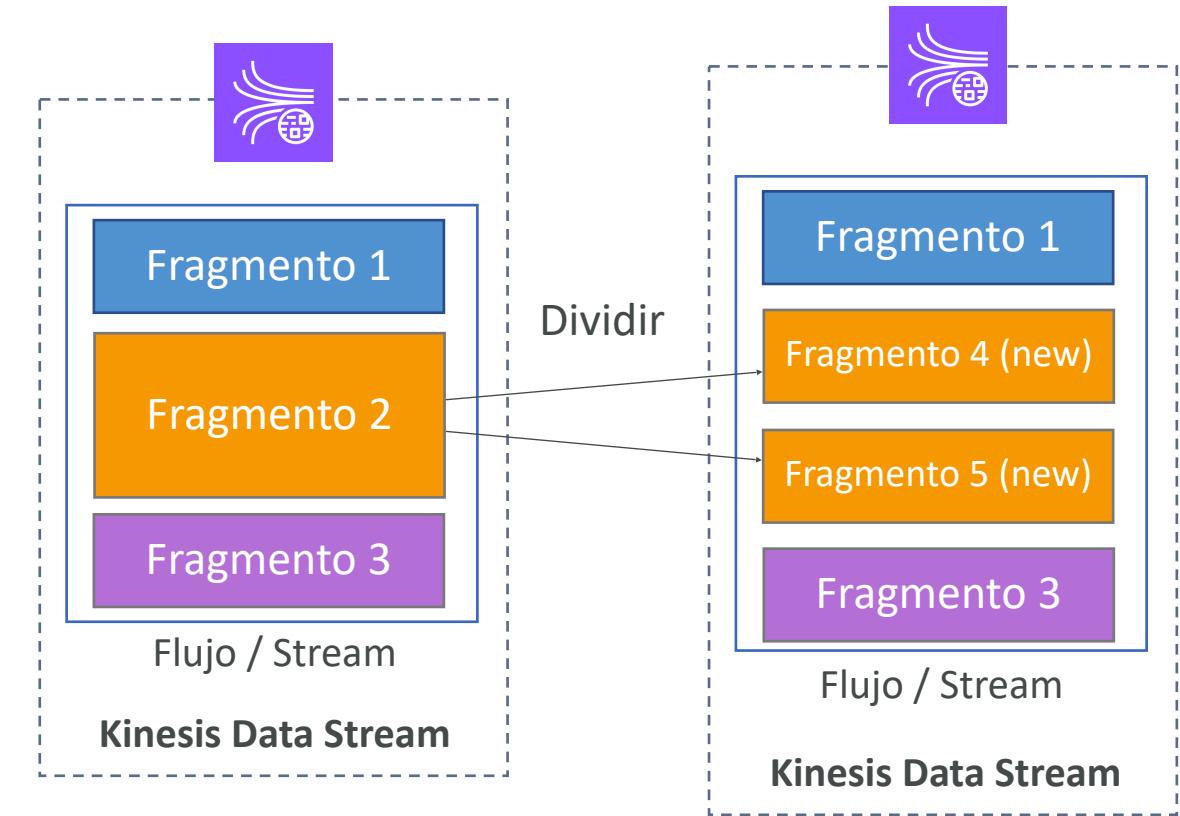
Procesar registros y guardarlos en DynamoDB



Procesar archivos y guardarlos en un bucket S3

# Operación Kinesis - Dividir fragmentos

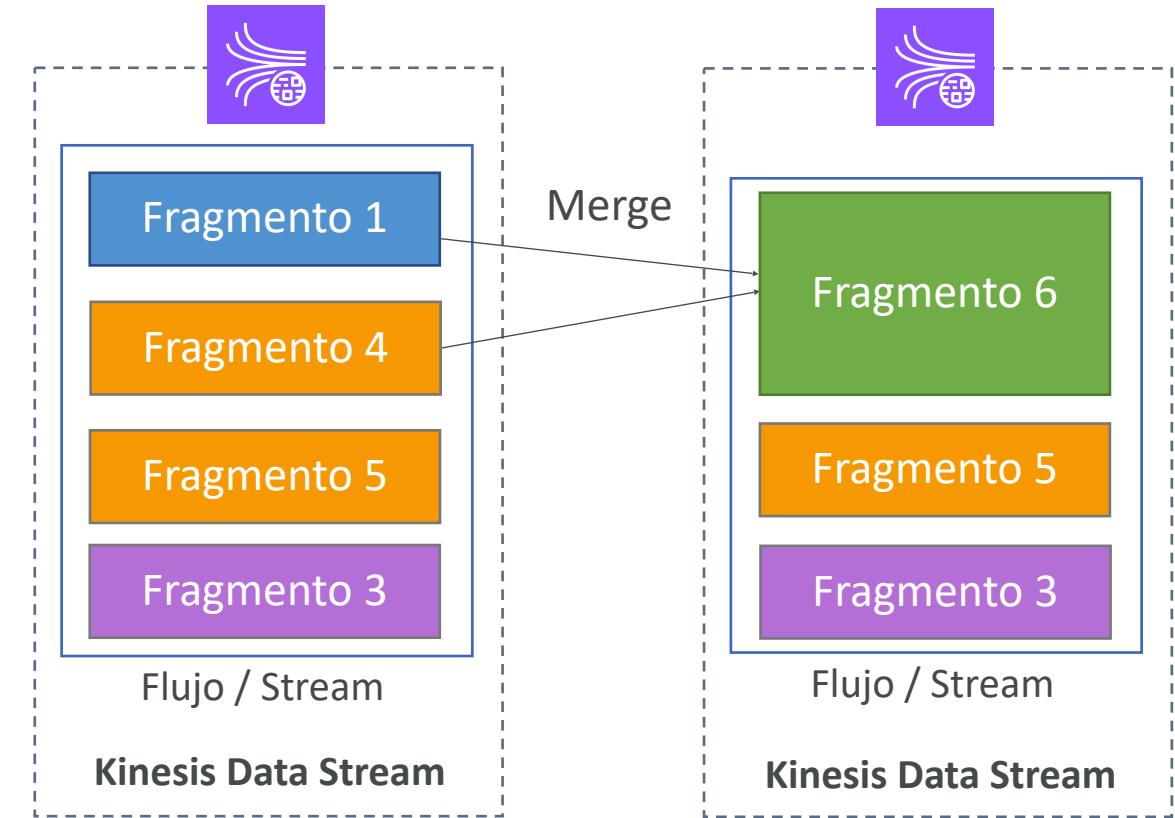
- Se utiliza para dividir un “fragmento caliente”
- El antiguo fragmento se cierra y se elimina cuando caducan los datos
- No hay escalado automático (aumenta/disminuye la capacidad manualmente)
- No se puede dividir en más de dos shards en una sola operación



Aumentar la capacidad y el coste

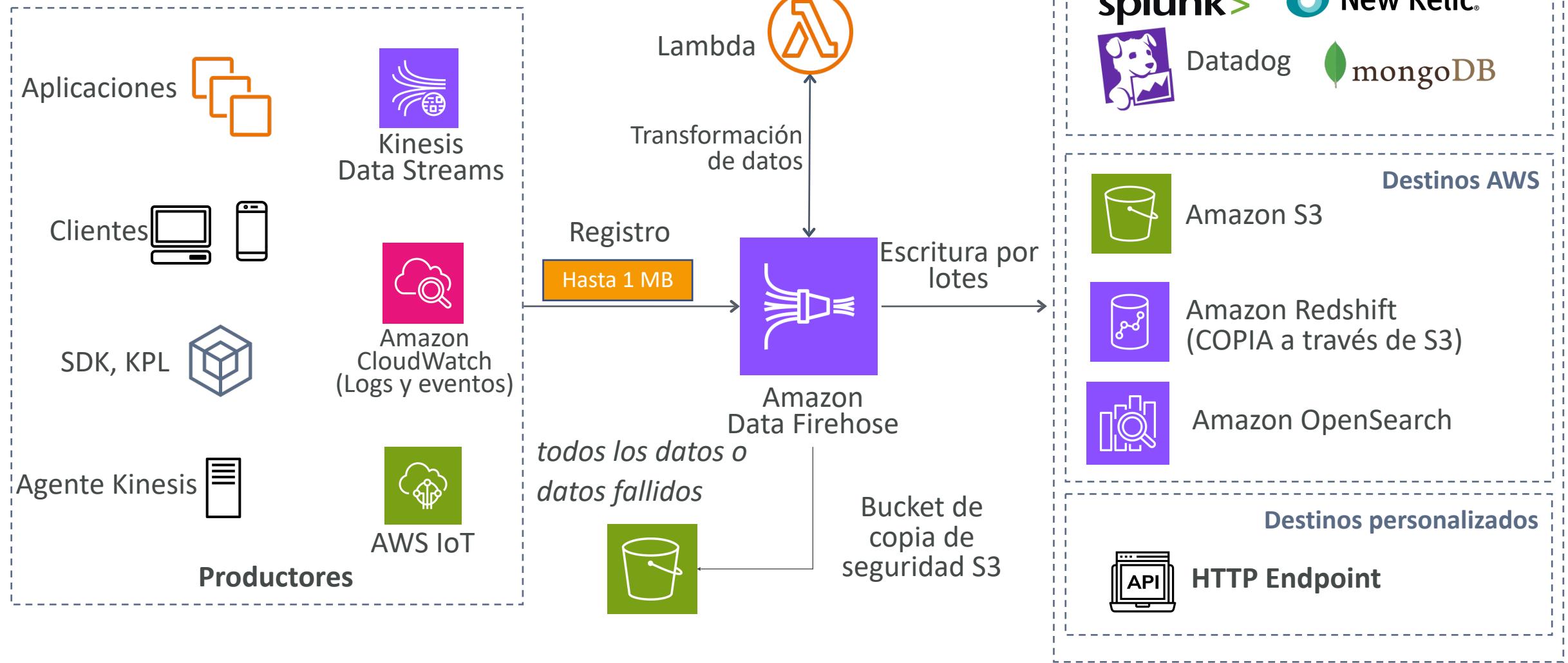
# Operación Kinesis - Merge ( fusión ) de fragmentos

- Reduce la capacidad del stream y ahorra costes
- Puede utilizarse para agrupar dos shards con poco tráfico (shards fríos)
- Los shards antiguos se cierran y se borrarán cuando caduquen los datos
- No se pueden fusionar más de dos shards en una sola operación

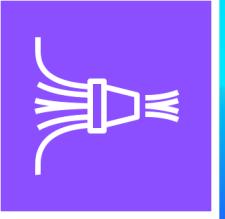


**Disminuir la capacidad y el coste**

# Amazon Data Firehose



# Amazon Data Firehose



- Servicio completamente gestionado, sin administración, escalado automático, sin servidor
  - AWS: Redshift / Amazon S3 / OpenSearch
  - Socio de terceros: Splunk / MongoDB / DataDog / NewRelic / ...
  - Personalizado: envía a cualquier endpoint HTTP
- Paga por los datos que pasan por Firehose
- **Casi en tiempo real**
  - Intervalo de buffer: de 0 segundos (sin buffer) a 900 segundos
  - Tamaño del buffer: mínimo 1MB
- Soporta muchos formatos de datos, conversiones, transformaciones, compresión
- Soporta transformaciones de datos personalizadas usando AWS Lambda
- Puede enviar datos fallidos o todos los datos a un bucket S3 de respaldo

# Kinesis Data Streams vs Amazon Data Firehose

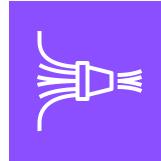


## Kinesis Data Streams

---

- Servicio de streaming para la ingesta a escala
- Escribir código personalizado (productor / consumidor)
- En tiempo real (~200 ms)
- Gestión del escalado (división/fusión de fragmentos)
- Almacenamiento de datos de 1 a 365 días
- Capacidad de reproducción

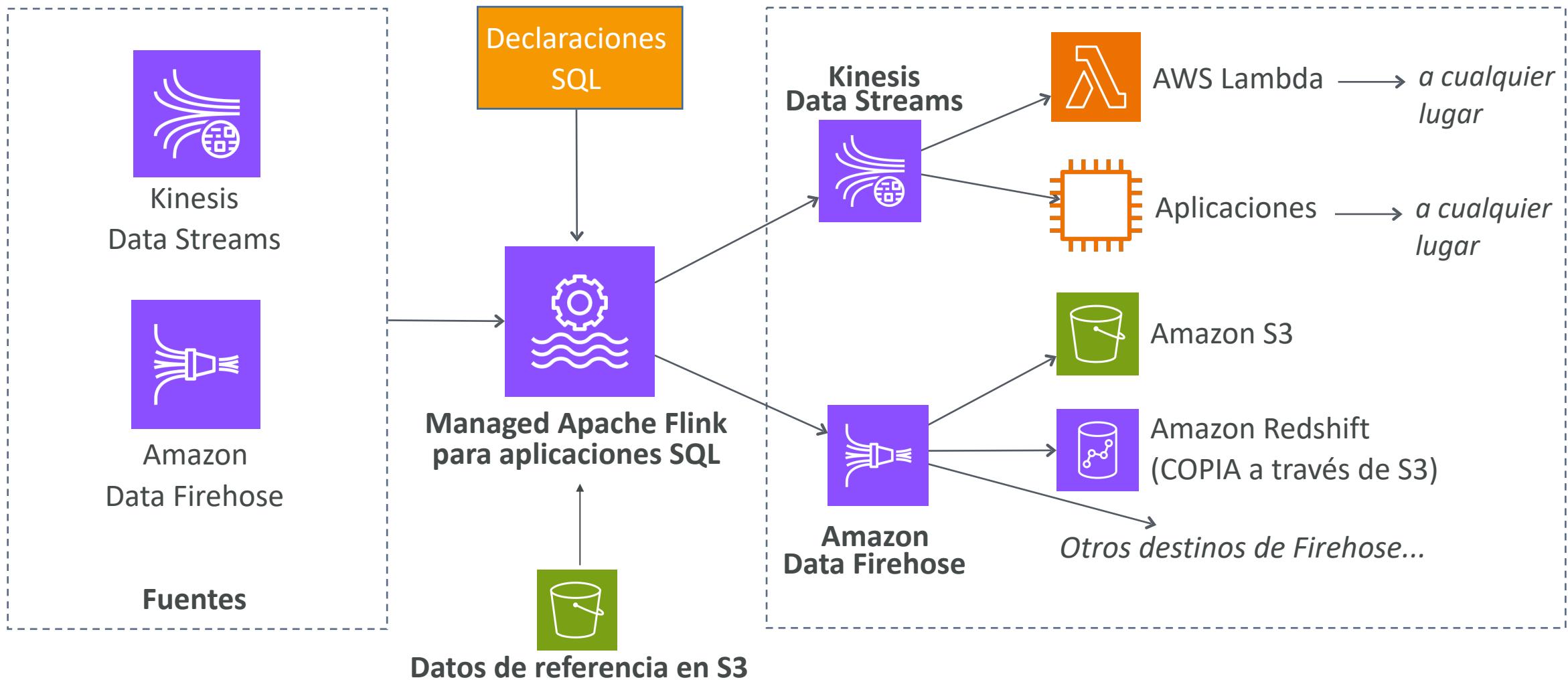
# Kinesis Data Streams vs Amazon Data Firehose

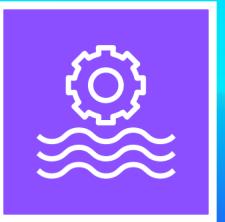


## Amazon Data Firehose

- 
- Carga de datos de streaming en S3 / Redshift / terceros / custom HTTP
  - Totalmente gestionado
  - Casi en tiempo real
  - Escalado automático
  - Sin almacenamiento de datos
  - No soporta capacidad de repetición

# Managed Apache Flink (antes Kinesis Data Analytics)





# Managed Apache Flink (aplicación SQL)

- Análisis en tiempo real en **Kinesis Data Streams & Amazon Data Firehose** mediante SQL
- Añadir datos de referencia de Amazon S3 para enriquecer los datos de streaming
- Totalmente administrado, sin servidores que aprovisionar
- Escalado automático
- Paga por la tasa de consumo real
- Salida:
  - Kinesis Data Streams: crea flujos a partir de las consultas analíticas en tiempo real
  - Amazon Data Firehose: envío de resultados de consultas analíticas a destinos
- Casos de uso:
  - Análisis de series temporales
  - Dashboards en tiempo real
  - Métricas en tiempo real



# Amazon Athena

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

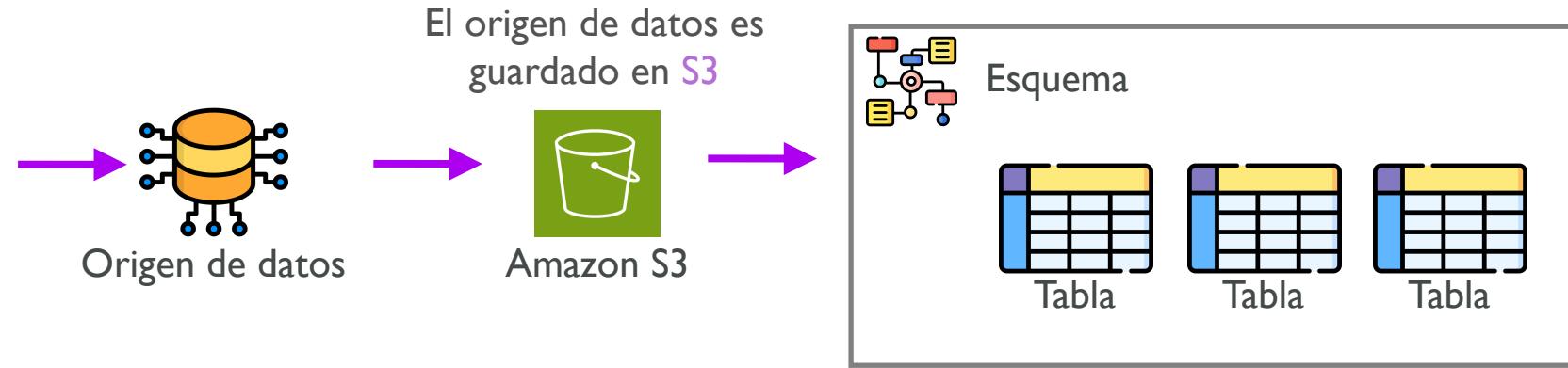


# ¿Qué es Amazon Athena?

- Servicio interactivo de consultas que permite analizar datos directamente en Amazon S3 utilizando SQL estándar sin necesidad de configurar servidores
  - No es necesario cargar los datos en Athena, permanecen en S3
- Servicio de consultas interactivas sin servidor
- Consultas ad-hoc sobre datos - paga solo por los datos consumidos
- **Esquema al leer** - traducción similar a una tabla
- Los datos originales **nunca se cambian** - permanecen en S3
- El esquema traduce los datos => similar a relacional cuando se leen
- Modelo de precio de pago por uso (Pay-as-you-go)
  - Ahorra mucho dinero usando formatos columnar (ORC, Parquet)



# Visión general de Amazon Athena



Las "tablas" se definen de antemano en un catálogo de datos y los datos se proyectan a través cuando se leen. Permite consultas similares a SQL sobre datos sin transformar los datos de origen

Admite formatos estándar de datos estructurados, semiestructurados y no estructurados



Athena puede leer directamente muchos formatos de datos de AWS como CloudTrail, registros de ELB y registros de flujo

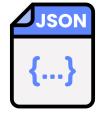
La salida se puede enviar a herramientas de visualización

# Formatos de datos y casos de uso

- Soporta muchos formatos de datos:



CSV, TSV  
(legibles por  
humanos)



JSON (legible  
por humanos)



ORC  
(columnar,  
divisible)



Parquet  
(columnar,  
divisible)



Avro  
(divisible)



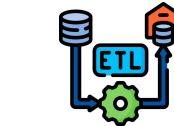
snappy

Zlib, LZO, Gzip

- Casos de uso:



Análisis de logs

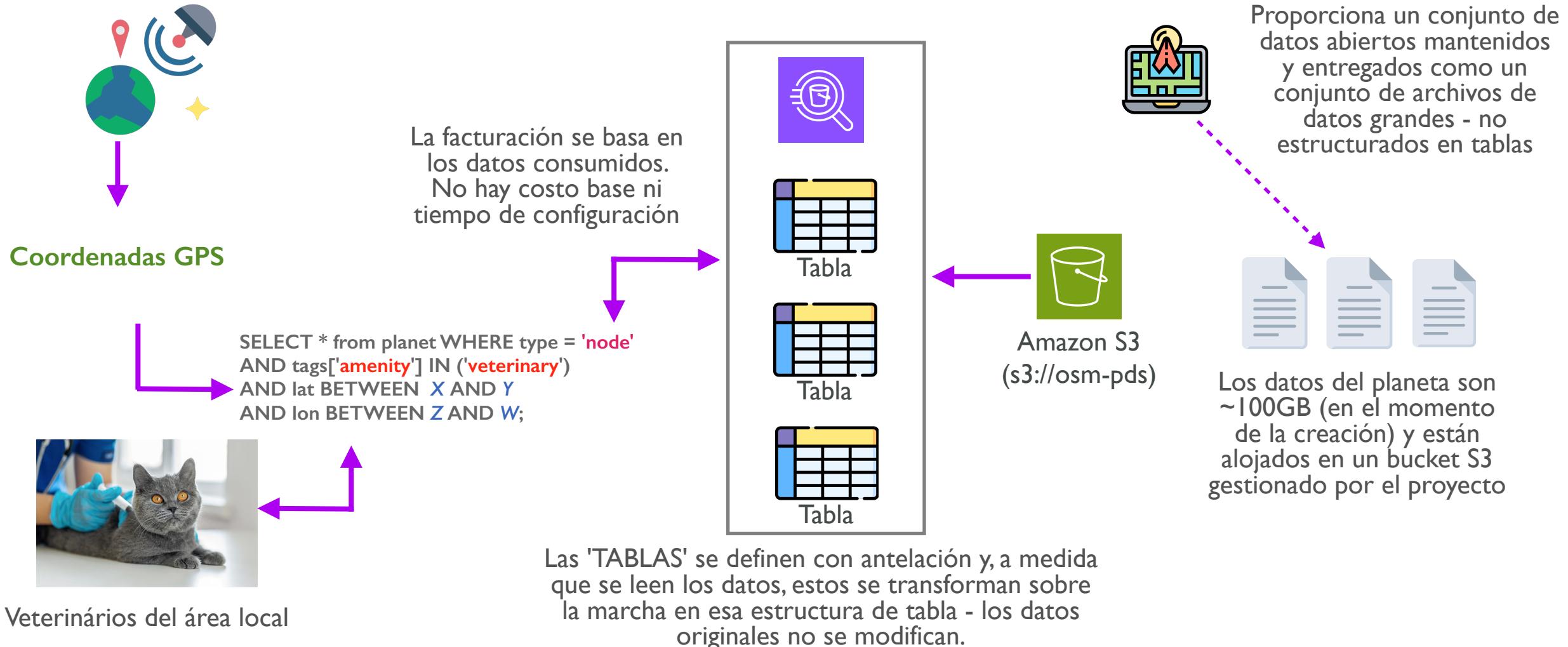


ETL (Extracción,  
Transformación y  
Carga)

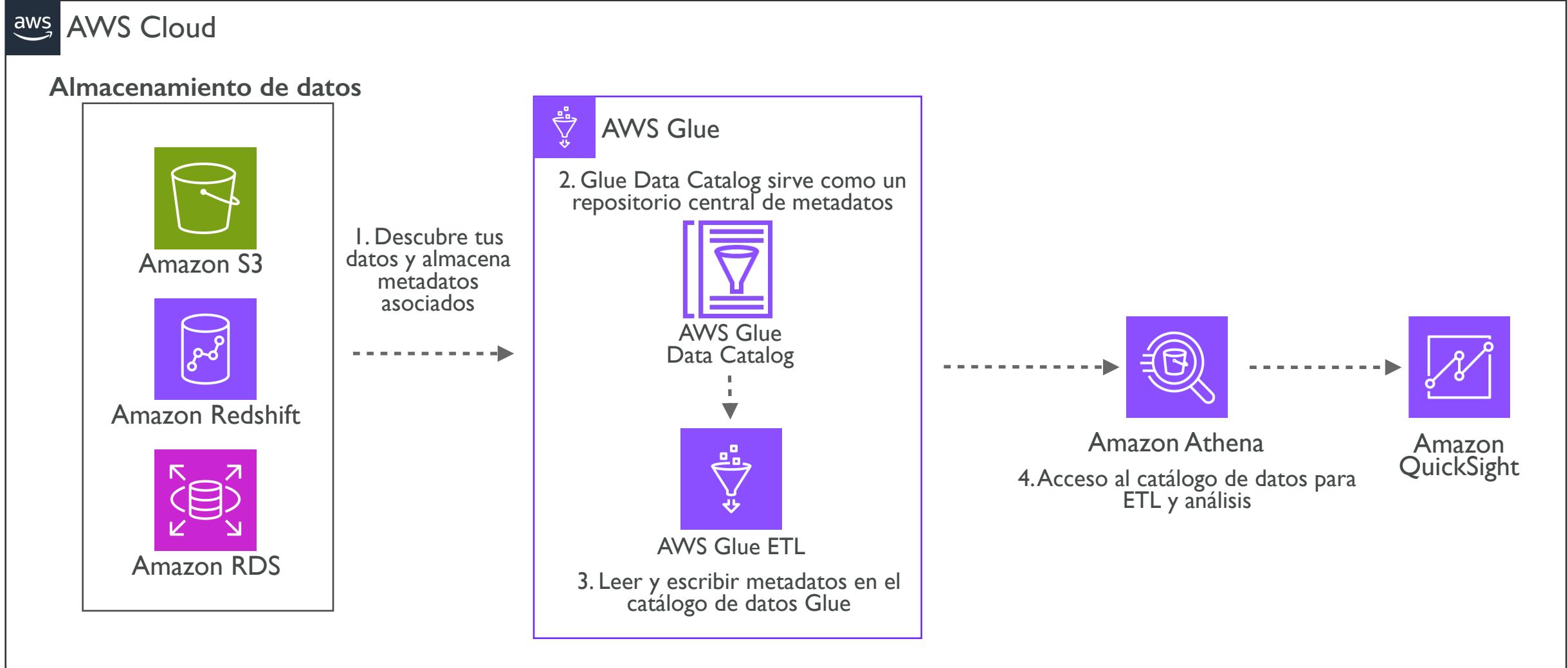


Lagos de datos

# [DEMO] Amazon Athena



# Amazon Athena + AWS Glue



# Grupos de trabajo de Athena

- Los grupos de trabajo de Athena se utilizan para controlar el acceso a recursos de consulta
- Cada cuenta tiene un grupo de trabajo principal y los permisos predeterminados permiten que todos los usuarios autenticados accedan a este grupo de trabajo
  - Control de acceso a las consultas y seguimiento de costos por grupo de trabajo
- Integración con IAM, CloudWatch, SNS
- Cada grupo de trabajo puede tener su propio:



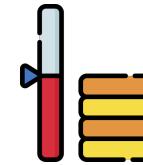
Historial de consultas



Políticas de IAM personalizadas



Configuraciones de cifrado



Límites de datos (puede limitar cuánto datos pueden escanear)

# Seguridad de Athena

- **Control de acceso**

- IAM, ACLs, políticas de buckets S3
- *AmazonAthenaFullAccess / AWSQuicksightAthenaAccess*

- Admite **opciones de cifrado** para conjuntos de datos y resultados de consulta en Amazon S3:

- Cifrado del lado del servidor con clave administrada por S3 (SSE-S3)
- Cifrado del lado del servidor con clave de KMS (SSE-KMS)
- Cifrado del lado del cliente con clave de KMS (CSE-KMS)

- **Cifrado en tránsito** (TLS) entre Athena y S3

- **Monitoreo y auditoría**

- Utiliza AWS CloudTrail para registrar todas las acciones realizadas en Athena y S3

- **Rotación de claves**

- Implementa políticas de rotación de claves en KMS para asegurar que las claves de cifrado se actualicen regularmente



# Consideraciones de Amazon Athena

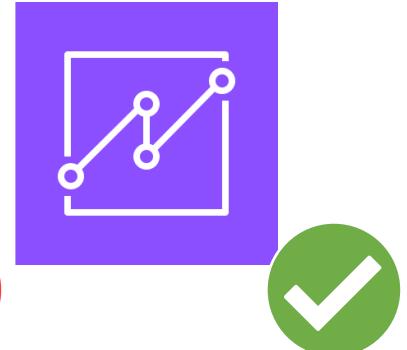
- **Informes altamente formateados / visualización**

- Amazon **Athena no está diseñado para la creación de informes** altamente formateados o visualizaciones complejas. Si tu objetivo es generar **dashboards interactivos o informes** visuales, **Amazon QuickSight** es la herramienta adecuada.

Athena



QuickSight



- **ETL (Extracción, Transformación y Carga)**

- Athena puede realizar algunas operaciones de transformación de datos, sin embargo no está optimizado para procesos ETL complejos. **AWS Glue** es la mejor opción.

Athena



Glue



# Amazon Athena - Optimización del rendimiento



- **Número de archivos grandes vs. archivos pequeños**

Tener un pequeño número de archivos grandes en lugar de muchos archivos pequeños mejora el rendimiento de las consultas en Athena



- **Uso de particiones**

Si estás gestionando particiones manualmente, utiliza el comando: `ALTER TABLE ADD PARTITION` para añadir nuevas particiones a la tabla



- **Usa datos columnar (ORC, Parquet)**

Los formatos de almacenamiento columnar como ORC y Parquet están optimizados para consultas de análisis.

# Transacciones ACID en Amazon Athena

\*ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)

- **Impulsadas por Apache Iceberg**

Solo añade `table_type = 'ICEBERG'` en tu comando `CREATE TABLE`

```
CREATE TABLE nombre_tabla(...)  
WITH (  
    table_type = 'ICEBERG', ...);
```

- **Tablas gobernadas en Lake Formation**



Esto es útil para organizaciones que requieren control de acceso granular y cumplimiento normativo junto con capacidades de transacción.

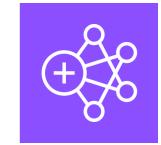
- **Modificaciones a nivel de fila**



Los usuarios pueden realizar modificaciones a nivel de fila sin comprometer la integridad de los datos

- **Compatibilidad**

Iceberg es compatible con AWS EMR, Apache Spark y cualquier otra plataforma que soporte el formato de tabla Iceberg



- **Compactación periódica**

Es un proceso que reescribe y organiza los datos para mejorar el rendimiento de las consultas y mantener la eficiencia del almacenamiento

```
OPTIMIZE table REWRITE DATA USING  
BIN_PACK WHERE catalog = 'c1';
```

- **Operaciones de viaje en el tiempo**

Permite a los usuarios recuperar estados anteriores

```
SELECT * FROM nombre_tabla FOR  
SYSTEM_TIME AS OF 'timestamp';
```

# Acceso detallado de Athena al catálogo de datos de AWS Glue

- **Seguridad a nivel de BD y tabla basada en IAM**

- Utilizar políticas de IAM para controlar el acceso a las bases de datos y tablas en el catálogo de datos

- **Limitaciones**

- Aunque la seguridad basada en IAM es amplia, no permite restricciones a versiones específicas de tablas. Esto significa que las políticas de IAM controlan el acceso a toda la tabla y no pueden restringir el acceso a versiones particulares de los datos

- **Política mínima requerida**

- Como mínimo, necesitas al menos una política que otorgue acceso a tu base de datos y al catálogo de datos en cada región



```
{  
  "Sid": "DatabasePermissions",  
  "Effect": "Allow",  
  "Action": [  
    "glue:GetDatabase",  
    "glue:GetDatabases",  
    "glue>CreateDatabase"  
,  
  "Resource": [  
    "arn:aws:glue:us-east-1:123456789012:catalog",  
    "arn:aws:glue:us-east-1:123456789012:database/  
default"  
,  
  ]  
}
```

# Acceso detallado de Athena al catálogo de datos de AWS Glue

- **Restricción de acceso a operaciones específicas**

- Esto proporciona un control detallado sobre quién puede realizar acciones críticas como:

*ALTER o CREATE DATABASE*

*CREATE TABLE*

*DROP DATABASE o DROP TABLE*

*MSCK REPAIR TABLE*

*SHOW DATABASES o SHOW TABLES*

```
{  
  "Effect": "Allow",  
  "Action": [  
    "glue:GetDatabase",  
    "glue:GetTable",  
    "glue:DeleteTable",  
    "glue:GetPartitions",  
    "glue:GetPartition",  
    "glue:DeletePartition"  
  ],  
  "Resource": [  
    "arn:aws:glue:us-east-1:123456789012:catalog",  
    "arn:aws:glue:us-east-1:123456789012:database/example_db",  
    "arn:aws:glue:us-east-1:123456789012:table/example_db/test"  
  ]  
}
```

- Documentación adicional: <https://docs.aws.amazon.com/athena/latest/ug/fine-grained-access-to-glue-resources.html>

# CREATE TABLE AS SELECT (CTAS)

## • Uso de CTAS en Amazon Athena

- La instrucción **CREATE TABLE AS SELECT (CTAS)** es una forma poderosa de crear nuevas tablas en Athena a partir de los resultados de una consulta SQL
- Es una práctica común en muchas bases de datos relacionales, no solo en Athena

## • Crear subconjuntos de tablas

- Puedes usar CTAS para crear una nueva tabla que contenga solo un subconjunto de los datos de otra tabla

## • Conversión de formatos de datos

- Por ejemplo, puedes convertir datos de una tabla almacenada en formato CSV a una tabla en formato Parquet u ORC

*Crear una nueva tabla en formato Parquet con compresión Snappy*

```
CREATE TABLE nueva_tabla
WITH (
    format = 'Parquet',
    write_compression = 'SNAPPY'
)
AS SELECT *
FROM tabla_antigua;
```

*Crear una nueva tabla en formato ORC y almacenarla en un lugar específico en S3*

```
CREATE TABLE mi_orc_tabla_ctas
WITH (
    external_location = 's3://mi_resultado_athena/
    mi_orc_tabla_stas/',
    format = 'ORC'
)
AS SELECT *
FROM tabla_antigua;
```

# Apache Spark

# Visión general de Apache Spark

- Motor de procesamiento de datos rápido y de código abierto



- Soporta APIs para Java, Scala, Python y R
- Ofrece procesamiento unificado para ETL, análisis interactivo, aprendizaje automático y procesamiento de gráficos
- Permite procesamiento en memoria, acelerando tareas hasta 100 veces más que Hadoop MapReduce
- Integra con Hadoop, bases de datos NoSQL y servicios en la nube

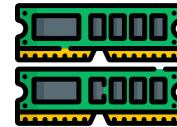
# Visión general de Apache Spark

- **Marco de procesamiento distribuido para big data**



Está diseñado para manejar grandes volúmenes de datos

- **Caché en memoria y ejecución de consultas optimizadas**



Permite realizar operaciones de datos a alta velocidad en comparación con MapReduce tradicional

- **Soporte multilinguaje**



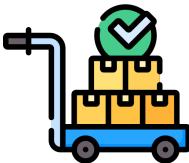
Soporta varios lenguajes de programación como Java, Scala, Python y R

- **Reutilización de código en múltiples tareas**



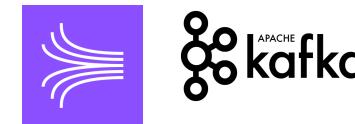
Procesamiento por lotes, consultas interactivas, análisis en tiempo real, aprendizaje automático y procesamiento de gráficos

- **No está destinado para OLTP**



Su enfoque principal es el procesamiento de datos en lotes y en tiempo real para análisis

- **Spark Streaming**



Procesamiento de datos en tiempo real. Se integra fácilmente con Amazon Kinesis y Apache Kafka

# ¿Cómo funciona Apache Spark?

- **Driver Program**

1. Declara las transformaciones y acciones
2. Crea el SparkContext

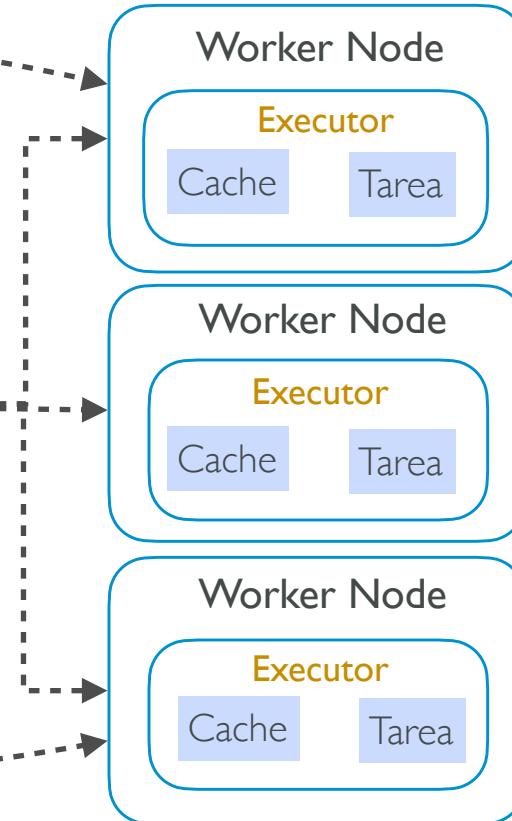


Solicitud de recursos para nodos trabajadores



- **SparkContext:**

1. Coordina la ejecución de la aplicación
2. Solicita recursos al Cluster Manager



- **Cluster Manager:**

1. Asigna recursos e instruye a los nodos trabajadores para ejecutar el trabajo

- **Executors**

1. Ejecutan las tareas asignadas
2. Almacenan los datos y resultados
3. Mantienen la caché en memoria

# Componentes de Apache Spark



Permite a los desarrolladores ejecutar consultas SQL en los datos almacenados en DataFrames y Datasets



Permite el procesamiento de flujos de datos en tiempo real

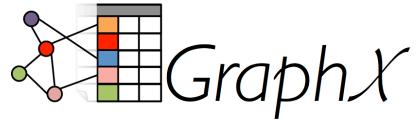


Proporciona varias implementaciones de algoritmos ML, incluyendo clasificación, regresión, clustering, filtrado colaborativo y minería de patrones



Permite a los usuarios de R aprovechar la capacidad de procesamiento distribuido de Spark para manipular y analizar grandes conjuntos de datos

# Componentes de Apache Spark



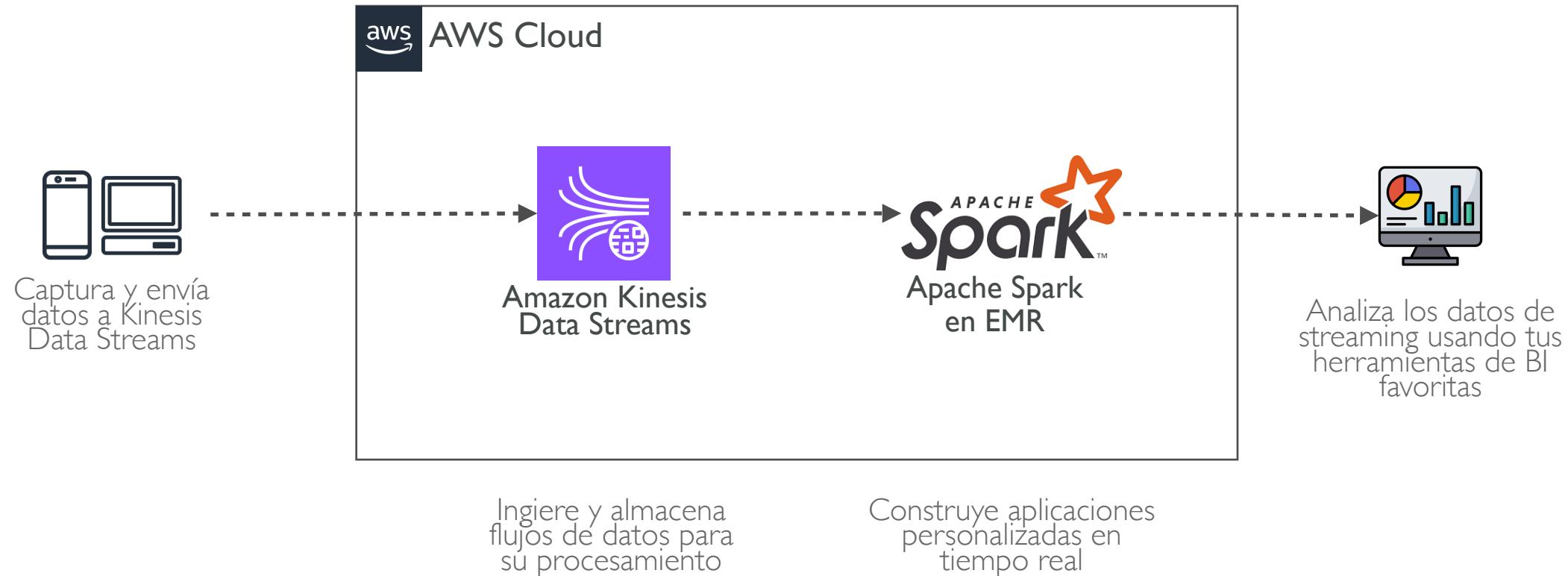
Permite a los desarrolladores realizar cálculos iterativos y consultas en gráficos grandes

## Apache Spark Core API



- Proporciona las funcionalidades básicas para la programación en Spark:
  - Gestión de memoria, la recuperación ante fallos, la programación y la monitorización de trabajos, etc

# Apache Spark + Amazon Kinesis

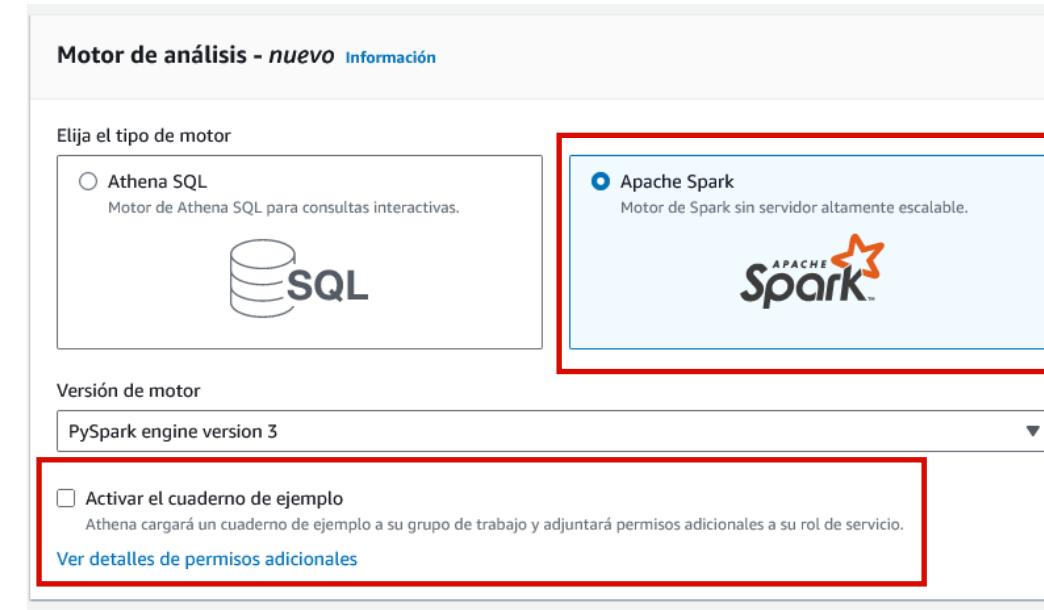


# Apache Spark + Amazon Redshift



# Apache Spark + Amazon Athena

- Seleccionable como un motor de análisis alternativo (vs. Athena SQL)
- Acceso programático mediante API / CLI
- Puedes ajustar los DPU's para el coordinador y los tamaños de los ejecutores
- Precio basado en el uso de cómputo y DPU por hora



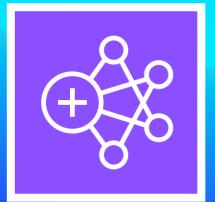


# Amazon EMR

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

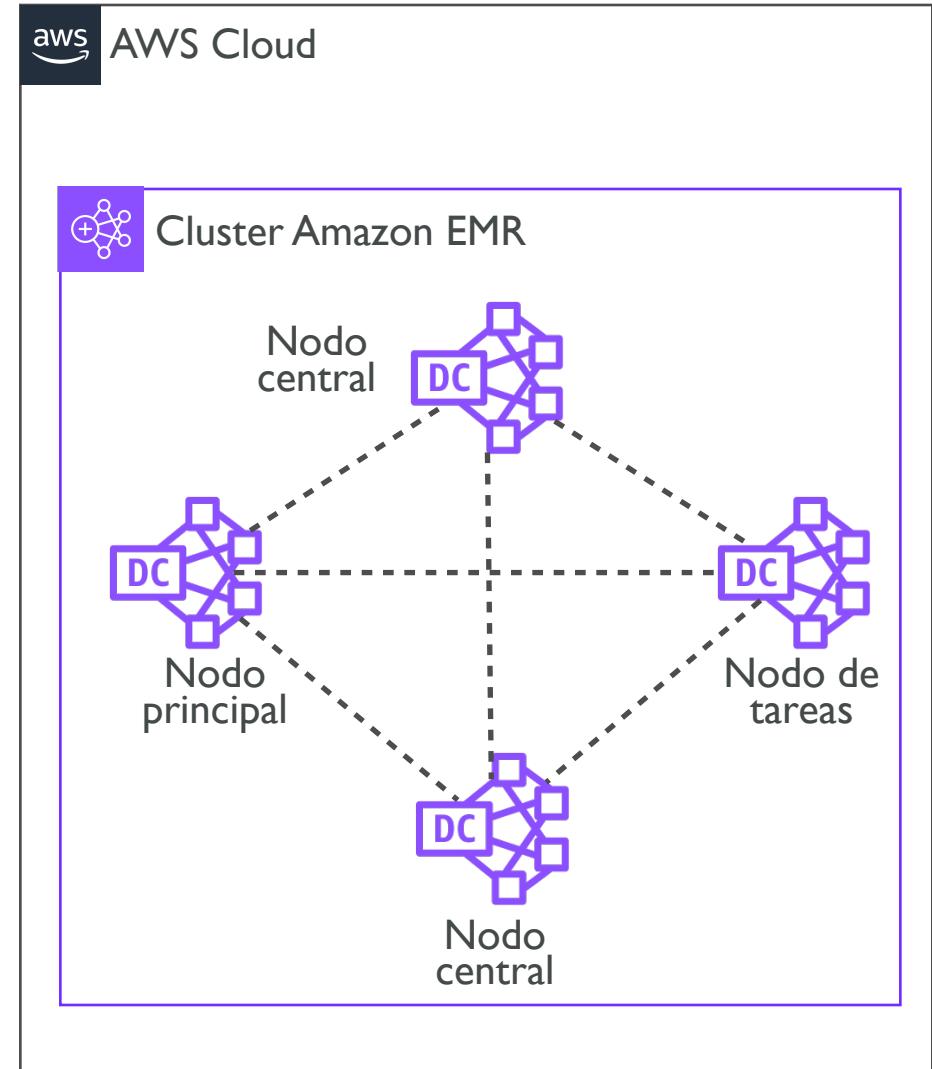
# Visión general de Amazon EMR



- **EMR = Elastic MapReduce**
- EMR ayuda a crear clusters Hadoop (**Big Data**) para analizar y procesar una gran cantidad de datos
- Los clusters pueden estar formados por **cientos de instancias EC2**
- También es compatible con Apache Spark, HBase, Presto, Flink...
- EMR se encarga de todo el aprovisionamiento y la configuración
- Autoescalado e integrado con instancias Spot
- Casos de uso: procesamiento de datos, Machine Learning, indexación web, big data...

# Clúster EMR

- **Nodo principal:** Gestiona el clúster
  - Monitorea el estado de las tareas y la salud del clúster
  - Instancia EC2 única (puede ser incluso un clúster de un solo nodo)
  - También conocido como "nodo líder"
- **Nodo central:** Aloja los datos en HDFS y ejecuta tareas
  - Puede ser escalado hacia arriba y hacia abajo
- **Nodo de tareas:** Ejecuta tareas, no aloja datos (\*Opcional)
  - No hay riesgo de pérdida de datos al removerlo
  - Buen uso de instancias de spot

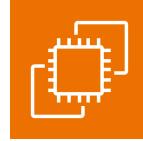


# Uso de EMR

- **Clústeres transitorios:** Se crean para realizar una tarea específica y luego se terminan una vez que todos los pasos se completan
  - Carga de datos, procesamiento, almacenamiento - luego se apagan
  - Ahorra dinero
- **Clústeres de larga duración:** Permanecen activos durante un período prolongado y deben ser terminados manualmente
  - Almacén de datos con procesamiento periódico en grandes conjuntos de datos
  - Pueden activar nodos de tareas utilizando instancias de spot para capacidad temporal
  - Pueden usar instancias reservadas en clústeres de larga duración para ahorrar dinero
  - Protección contra terminación activada por defecto, auto-terminación desactivada



# Integración EMR en AWS



**Amazon EC2**

Proporciona las instancias que componen los nodos del clúster



**Amazon VPC**

Configura la red virtual donde se lanzan las instancias



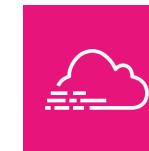
**Amazon S3**

Almacena los datos de entrada y salida



**Amazon CloudWatch**

Monitorea el rendimiento del clúster y configura alarmas



**Amazon CloudTrail**

Audita las solicitudes realizadas al servicio



**AWS IAM**

Configura permisos y seguridad



**AWS Data Pipeline**

Programa y arranca tus clústeres

# Almacenamiento EMR

- Uno de los aspectos clave de Amazon EMR es cómo maneja el almacenamiento de datos
- Hay varias opciones disponibles para almacenar y gestionar datos en un clúster de EMR:
  - **EBS (Elastic Block Store)**: EBS proporciona almacenamiento en bloque que puede ser utilizado por instancias EC2 dentro de un clúster de EMR
  - **HDFS (Hadoop Distributed File System)**: HDFS es el sistema de archivos distribuido utilizado por Hadoop para almacenar grandes volúmenes de datos en clústeres
  - **EMRFS (EMR File System)**: EMRFS es una implementación de Hadoop Compatible File System (HCFS) que permite a EMR acceder a Amazon S3 como si fuera un sistema de archivos Hadoop
  - **Discos Locales (Local Disks)**: Además de HDFS y S3, EMR puedes utilizar los discos locales de las instancias EC2
  - **Glue Data Catalog**: AWS Glue Data Catalog es un servicio que permite la gestión y catalogación de metadatos para los datos almacenados en S3 y otras fuentes

# Almacenamiento EMR - EBS

- **EBS (Elastic Block Store)**

- Permite el uso de EMR en tipos solo EBS (M4, C4)
- Los volúmenes EBS se eliminan cuando se termina el clúster
- Los volúmenes EBS solo se pueden adjuntar al lanzar un clúster
- Si se desmonta manualmente un volumen EBS, EMR lo trata como un fallo y lo reemplaza



# Almacenamiento EMR - HDFS

- **HDFS (Hadoop Distributed File System)**

- Múltiples copias almacenadas en instancias del clúster para redundancia
- Archivos almacenados como bloques (tamaño por defecto de 128 MB)
- Los datos en HDFS se pierden cuando el clúster se termina (efímeros)
- Útil para el almacenamiento intermedio o cargas de trabajo con I/O aleatorio significativo
- Hadoop intenta procesar los datos donde están almacenados en HDFS

# Almacenamiento EMR - EMRFS

- **EMRFS (EMR File System)**

- Acceso a S3 como si fuera HDFS
- Permite almacenamiento persistente después de la terminación del clúster
- Opcional para la consistencia de S3
  - Usa DynamoDB para rastrear la consistencia
  - Puede ser necesario ajustar la capacidad de lectura/escritura en DynamoDB

# Almacenamiento EMR - Discos locales

- **Discos locales**

- Además de HDFS y S3, Amazon EMR puede usar los discos locales de las EC2
- Útil para operaciones de E/S intensivas debido a la baja latencia y alto rendimiento de los discos locales
- Los datos en los discos locales también se pierden cuando la instancia se termina, similar a HDFS

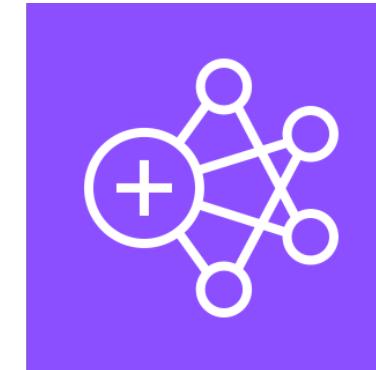


# Almacenamiento EMR - Glue Data Catalog

- **Glue Data Catalog**

- Amazon EMR puede integrarse con el catálogo de datos de AWS Glue para gestionar metadatos
- Permite la definición de esquemas y gestión de metadatos para datos almacenados en S3 y otras fuentes
- Facilita operar con otras herramientas de AWS y aplicaciones de análisis de datos

EMR



Glue



# Ventajas de EMR



- **Cobros por hora**

- EMR cobra por hora de uso
- Además, se aplican cargos de EC2



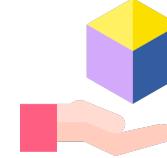
- **Provisión de nuevos nodos**

- Provisión de nuevos nodos automáticamente si falla un nodo central



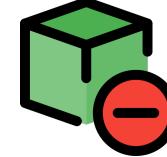
- **Gestión de nodos de tareas**

- Añadir y eliminar nodos de tareas sobre la marcha
- Aumenta la capacidad de procesamiento, pero no la capacidad de HDFS



- **Redimensionamiento de nodos centrales**

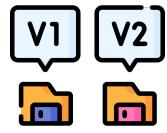
- Permite redimensionar los nodos centrales en ejecución
- Aumenta tanto la capacidad de procesamiento como la de HDFS



- **Adición y eliminación de nodos centrales**

- Los nodos centrales pueden ser añadidos o eliminados
- Eliminar nodos centrales conlleva el riesgo de pérdida de datos

# EMR Serverless



- **Elección de versión y runtime**

- Elige una versión de EMR y un runtime (Spark, Hive, Presto)



- **¿Por qué es importante?**

- No necesitas estimar cuántos workers se necesitan para tus cargas de trabajo; se provisionan automáticamente según sea necesario



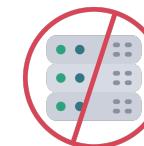
- **Envío de consultas / scripts**

- Envía consultas o scripts a través de solicitudes de ejecución de trabajos



- **Gestión de capacidad por EMR**

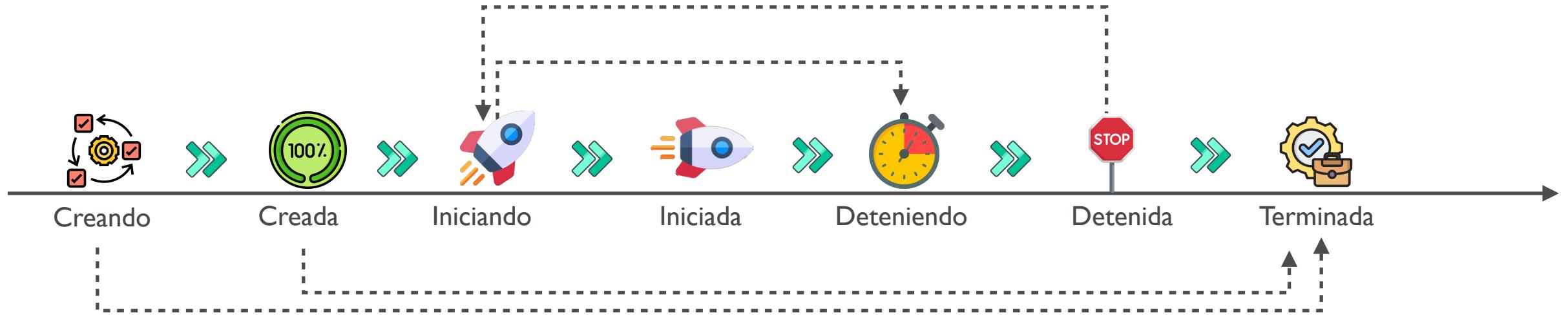
- EMR gestiona la capacidad subyacente automáticamente
- EMR calcula los recursos necesarios para tu trabajo y programa los trabajadores en consecuencia
- Todo dentro de una sola región (a través de múltiples zonas de disponibilidad)



- **¿Realmente es serverless?**

- Aunque es sin servidor, aún debes considerar los nodos de trabajadores y cómo se configuran

# Ciclo de vida de la app con EMR Serverless



**¡Esto no es todo automático!**

Debes llamar a las API, tales como:

- CreateApplication (Crear aplicación)
- StartApplication (Iniciar aplicación)
- StopApplication (Detener aplicación)
- Y, lo más importante, DeleteApplication (Eliminar aplicación) para evitar cargos excesivos

# Técnicas de escalado en EMR

## Escalado automático en EMR

- Basado en reglas de escalado personalizadas usando métricas de CloudWatch
- Solo soporta grupos de instancias



## Estrategia de escalado hacia abajo

- Reducir capacidad:
  - Primero elimina nodos de tareas, luego nodos centrales, sin pasar los límites mínimos establecidos
  - Las instancias spot siempre se eliminan antes que las instancias bajo demanda



## Escalado administrado en EMR

- Soporta grupos de instancias y flotas EC2
- Escala instancias spot, bajo demanda y en un plan de ahorro dentro del mismo clúster
- Disponible para cargas de trabajo en Spark, Hive y YARN

## Estrategia de escalado hacia arriba

- Añadir capacidad:
  - Primero añade nodos centrales, luego nodos de tareas, hasta alcanzar el límite máximo especificado





# Amazon QuickSight

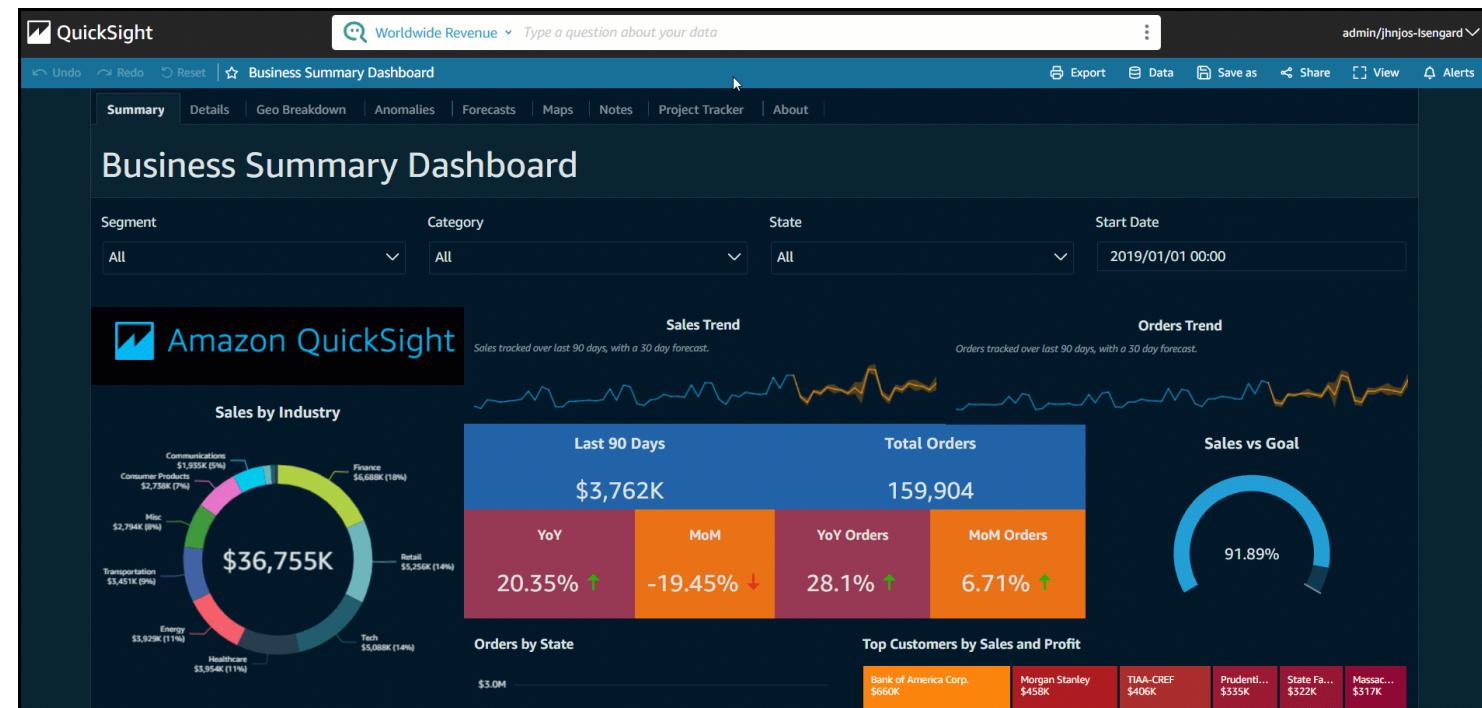
[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

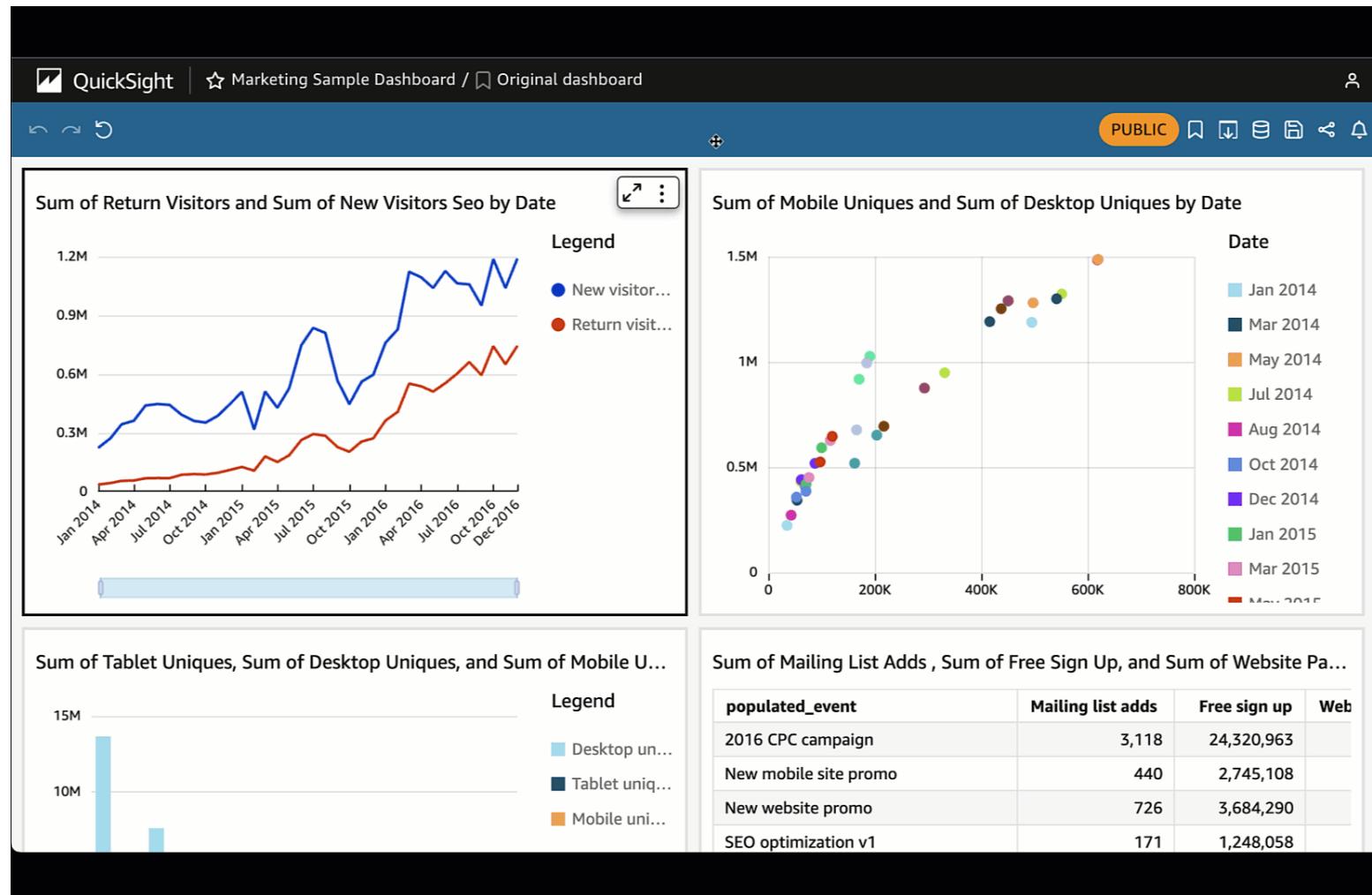
# Visión general de Amazon QuickSight



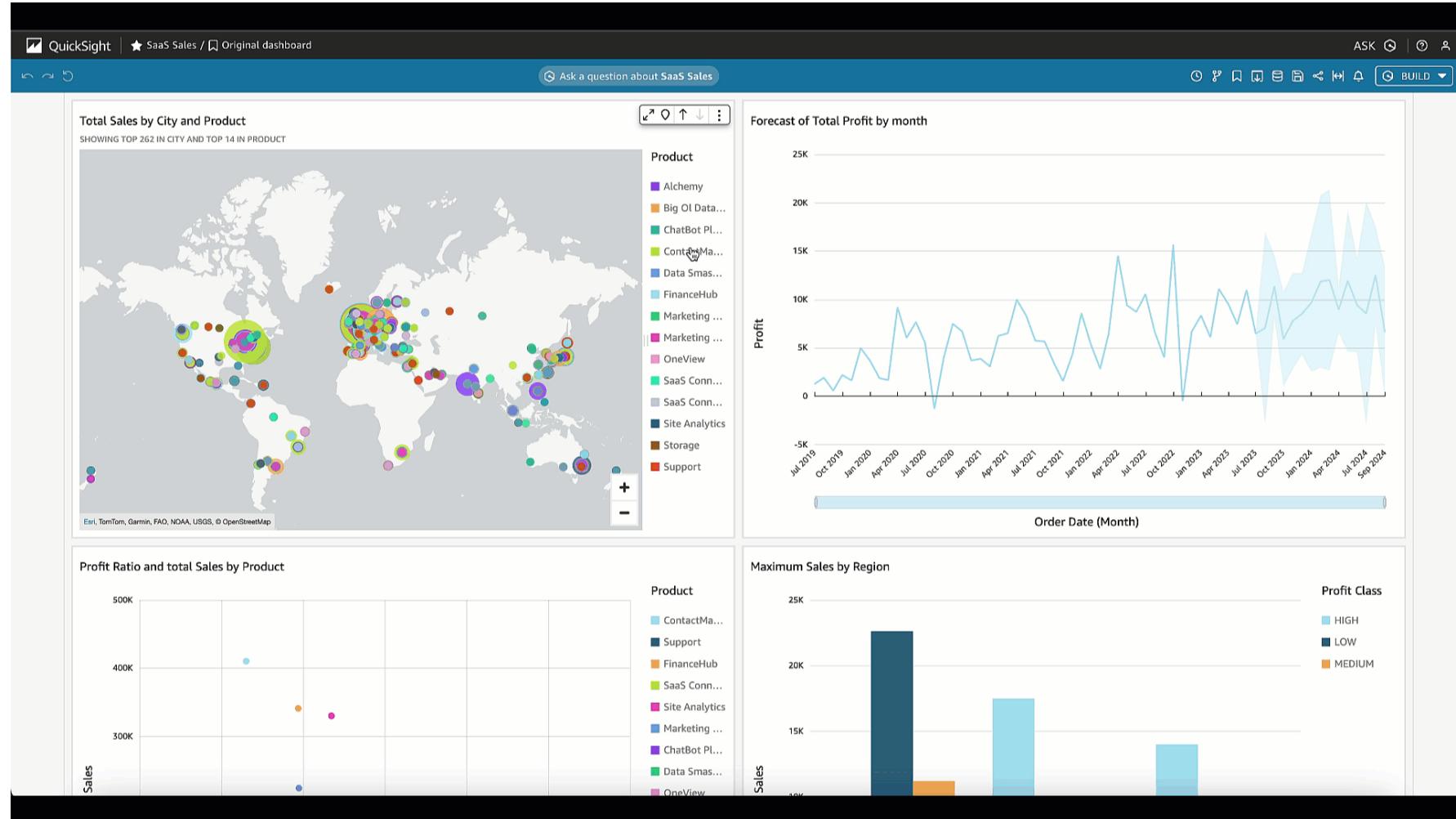
- Servicio de **análisis empresarial** en la nube que permite a los usuarios crear y compartir visualizaciones interactivas y dashboards de datos (sin servidor)
- QuickSight se integra de manera nativa con otras soluciones de AWS (S3, RedShift, RDS, etc.)
- Permite:
  - Crear visualizaciones
  - Realizar análisis ad-hoc
  - Recibir alertas sobre anomalías detectadas
  - Obtener rápidamente conocimientos empresariales a partir de datos
  - En cualquier momento, en cualquier dispositivo (navegadores, móviles, etc.)



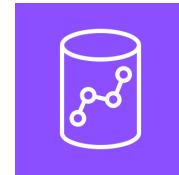
# Visión general de Amazon QuickSight



# Amazon QuickSight ❤️ Amazon Q



# Fuentes de datos de QuickSight



Amazon RedShift



Amazon Aurora



Amazon RDS



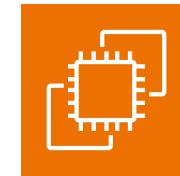
Amazon Athena



Amazon OpenSearch



AWS IoT Analytics



Bases de datos alojadas  
en Amazon EC2



S3

# Casos de uso de Amazon QuickSight

## Análisis de ventas y marketing



Visualiza métricas clave de rendimiento (KPI) y tendencias de ventas



Analiza campañas de marketing y su impacto



Cualquier fuente de datos JDBC/ODBC



Aplicaciones SaaS, como Salesforce



Bases de datos locales

## Análisis financiero



Monitorea ingresos, gastos y rentabilidad



Crea informes financieros detallados

## Gestión de recursos humanos



Analiza datos de empleados, como retención y rendimiento



Visualiza estadísticas de capacitación y desarrollo

## Operaciones y logística



Rastrea el inventario y la cadena de suministro



Optimiza rutas de entrega y tiempos de respuesta

## Monitoreo de TI y Seguridad



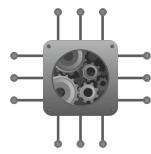
Visualiza métricas de rendimiento del sistema



Monitorea actividades de seguridad y cumplimiento

# SPICE: Motor de cálculo en memoria súper rápido y paralelo

## Conjuntos de datos importados a SPICE



Motor de cálculo en memoria, paralelo y súper rápido



Utiliza almacenamiento columnar, en memoria y generación de código máquina



Acelera consultas interactivas en grandes conjuntos de datos

## Beneficios de SPICE para los usuarios



Cada usuario obtiene 10GB de SPICE



Altamente disponible y duradero



Escalable a cientos de miles de usuarios



## Mejora el rendimiento de consultas

Puede acelerar consultas grandes que se agotarían en modo de consulta directa (accediendo a Athena directamente)



**Nota:** Si la importación de datos a SPICE toma más de 30 minutos, la operación se agotará

# Seguridad de QuickSight

- Acceso a recursos:
  - Debes **asegurarte de que QuickSight esté autorizado** para usar Athena y buckets de S3
  - Esto se **puede gestionar dentro de la consola de QuickSight**
- Acceso a datos:
  - Se **pueden crear políticas IAM para restringir qué datos en S3** pueden acceder los usuarios de QuickSight
-  **NOTA IMPORTANTE:**
  - Si QuickSight no tiene **permisos para descifrar** los datos almacenados en S3, no podrá acceder a los datos, incluso si puede listar / ver los metadatos del bucket

# Seguridad de QuickSight con Redshift

- **PROBLEMÁTICA:**

- Por defecto, QuickSight solo puede acceder a datos almacenados **EN LA MISMA REGIÓN** en la que QuickSight está ejecutándose
- Entonces, si QuickSight se ejecuta en una región y Redshift en otra, eso es un problema.

# Seguridad de QuickSight con Redshift

- **SOLUCIÓN:**

- Crea un nuevo grupo de seguridad con una regla de entrada que autorice el acceso desde el rango de IP de los servidores de QuickSight en esa región.
- Esos rangos están documentados en <https://docs.aws.amazon.com/quicksight/latest/user/regions.html>

# Precios de QuickSight Enterprise Edition

## Autores

Los autores de QuickSight pueden conectarse a los datos, crear dashboards e informes y compartir contenido con otros usuarios

### Autor

- 24 USD por usuario/mes
- 18 USD por usuario al mes con compromiso anual

### Autor Pro

- 50 USD por usuario/mes

## Lectores

Solo se les puede conceder acceso como **espectadores** a los dashboards. Pueden ver, exportar e imprimir el dashboard, pero no pueden guardarlo como un análisis. Pueden ver, filtrar y ordenar los datos del panel.

### Lector

- 3 USD por usuario/mes

### Lector Pro

- 20 USD por usuario/mes

# Precios de QuickSight Enterprise Edition

## Precios de capacidad para aplicaciones integradas

Los precios de capacidad permiten a los clientes de QuickSight comprar sesiones de lector o preguntas de Amazon Q en bloque, sin tener que aprovisionar usuarios individuales en QuickSight

### Capacidad de lector

- Desde 250 USD para 500 sesiones al mes

### Capacidad de preguntas de Amazon Q

- Desde 250 USD para 500 preguntas al mes

<https://aws.amazon.com/es/quicksight/pricing/>

# Precios de QuickSight Standard Edition

| Tipo                | Precio                | Incluye capacidad SPICE            |
|---------------------|-----------------------|------------------------------------|
| <b>Plan anual</b>   | \$ 9 por usuario/mes  | 10 GB por usuario                  |
| <b>Plan mensual</b> | \$ 12 por usuario/mes | 0,25 por GB de capacidad adicional |

<https://aws.amazon.com/es/quicksight/pricing/>

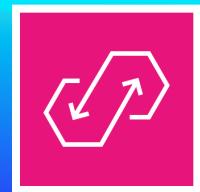


# Amazon AppFlow

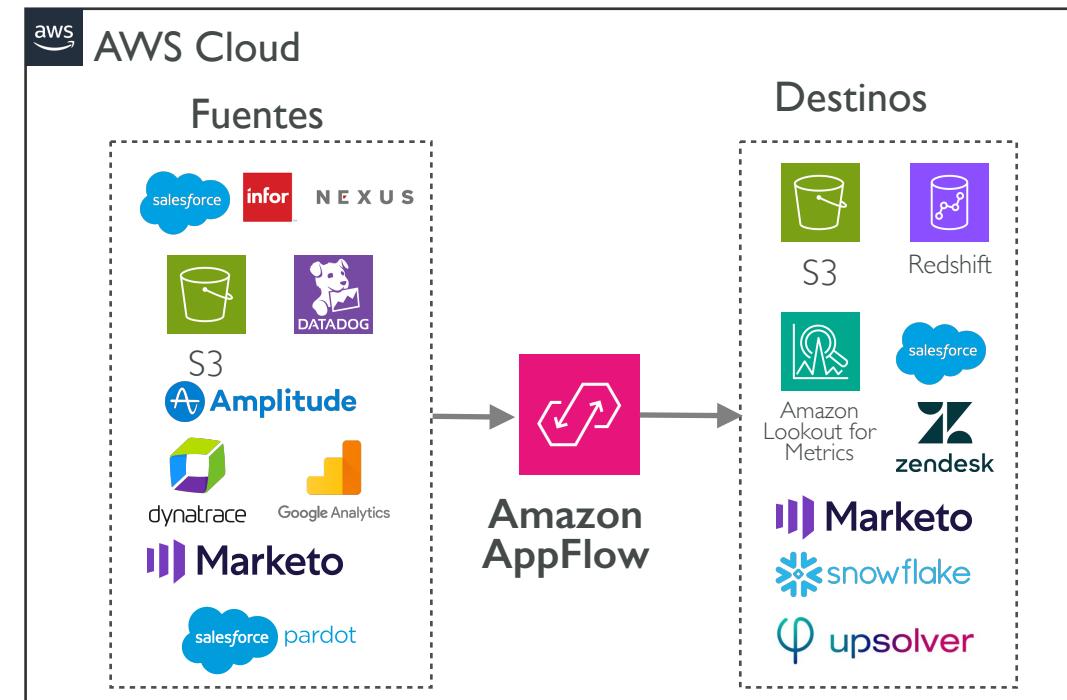
[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Visión general de Amazon AppFlow



- Facilita la **transferencia** segura y automática de datos **entre servicios AWS y aplicaciones SaaS**
- Elimina la necesidad de desarrollar y mantener integraciones personalizadas a través de una interfaz gráfica sencilla y amigable
- Ofrece capacidades integradas de **transformación de datos** (limpiar, filtrar y enriquecer datos durante su transferencia)
- Garantiza la seguridad de los datos en tránsito mediante cifrado y cumple con diversos estándares de cumplimiento (HIPAA y GDPR)
- Frecuencia de transferencia (programada, eventos, bajo demanda)



# Beneficios de Amazon AppFlow



**Seguridad y confiabilidad:** Garantía de que los datos se transfieren de manera segura y conforme a los estándares de AWS



**Facilidad de uso:** Elimina la necesidad de escribir y mantener integraciones personalizadas



**Flexibilidad y escalabilidad:** Permite ajustar y escalar las transferencias de datos según las necesidades del negocio



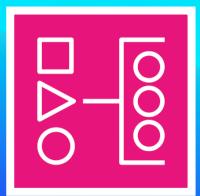
**Automatización:** Automatización de procesos entre aplicaciones con flujos de trabajo basados en eventos



# Amazon MWAA

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)



# Visión general de Amazon MWAA

- Amazon MWAA = Amazon Managed Workflows for Apache Airflow
- Creación, programación y monitoreo de **flujos de trabajo** complejos
- Interfaz fácil de usar para gestionar flujos de trabajo como código
- Los flujos de trabajo se definen como código Python que crea un **DAG** (Grafo Acíclico Dirigido)
- Gestiona la infraestructura necesaria, eliminando la necesidad de instalación y mantenimiento



Casos de uso:

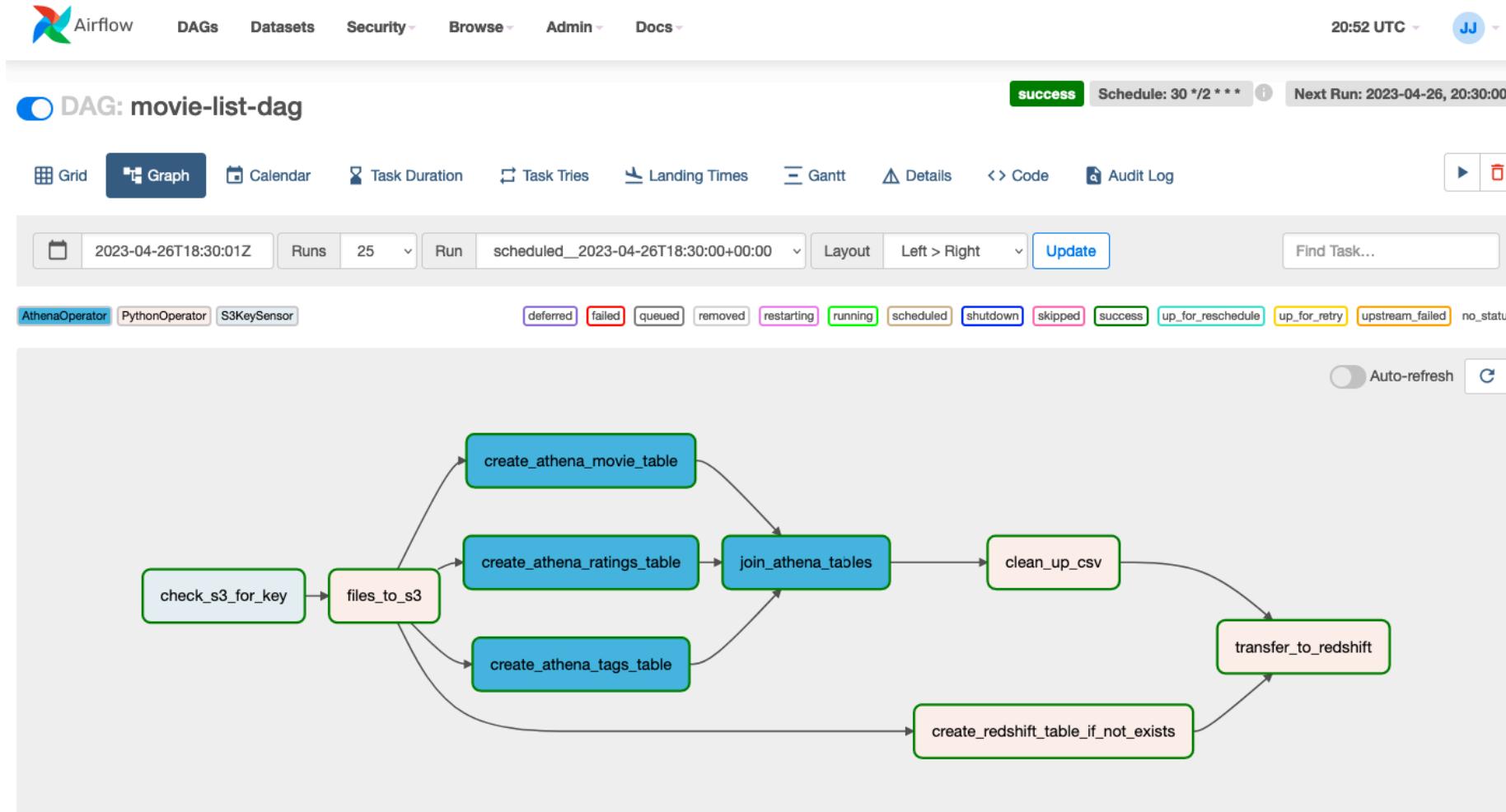
**Flujos de trabajo complejos**



**Coordina trabajos de ETL**

**Prepara datos de ML**

# Flujos de trabajo de Amazon MWAA



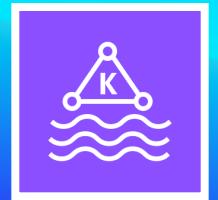


# Amazon MSK

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Visión general de Amazon MSK



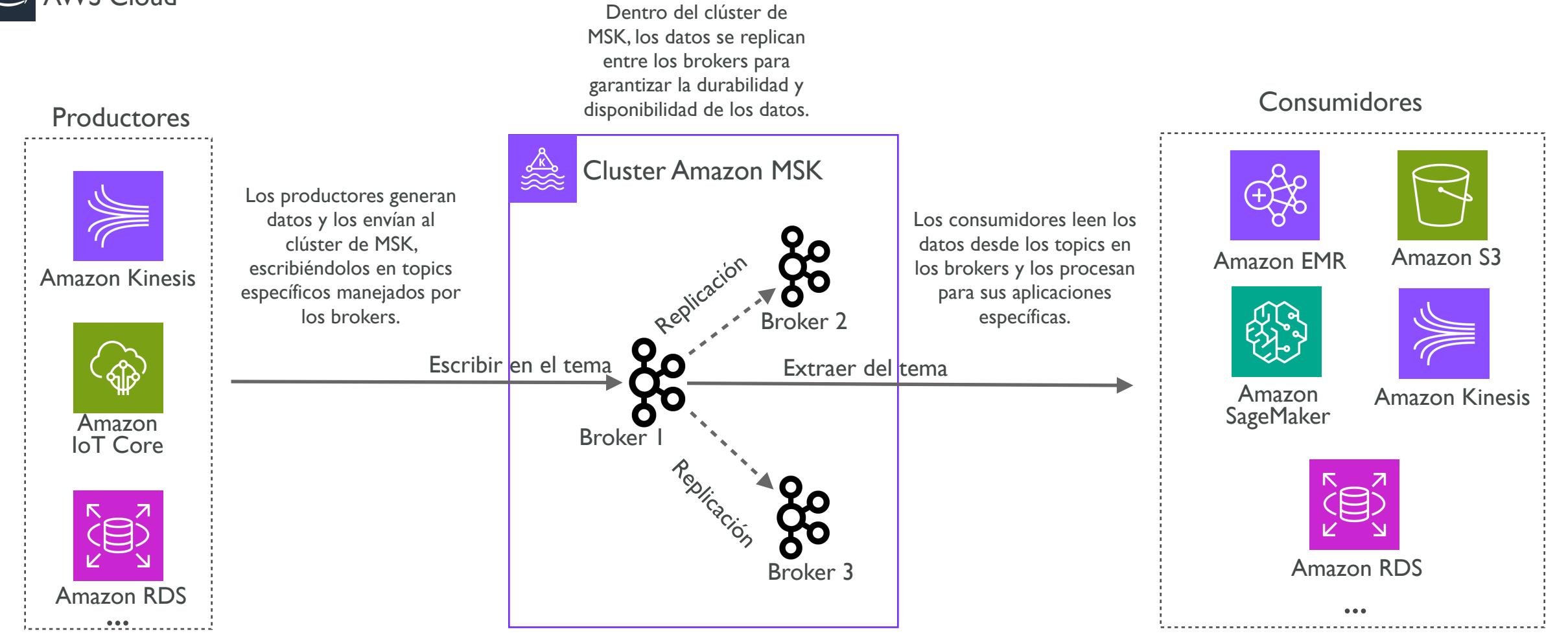
- Amazon MSK = Amazon Managed Streaming for Apache Kafka
- **Se encarga del aprovisionamiento, configuración y mantenimiento de clústeres de Apache Kafka**
- Puedes usar aplicaciones y herramientas existentes de Kafka sin cambios en el código
- Asegura una alta disponibilidad al reemplazar automáticamente los nodos no saludables y replicar datos a través de múltiples AZs
- Integración con Lambda, Amazon CloudWatch y Glue, facilitando la construcción y monitoreo de apps de streaming
- Precios:
  - Pago por uso, con cargos basados en el tiempo que las instancias están funcionando
  - Almacenamiento utilizado
  - Tarifas estándar de transferencia de datos



# Visión general de Apache Kafka

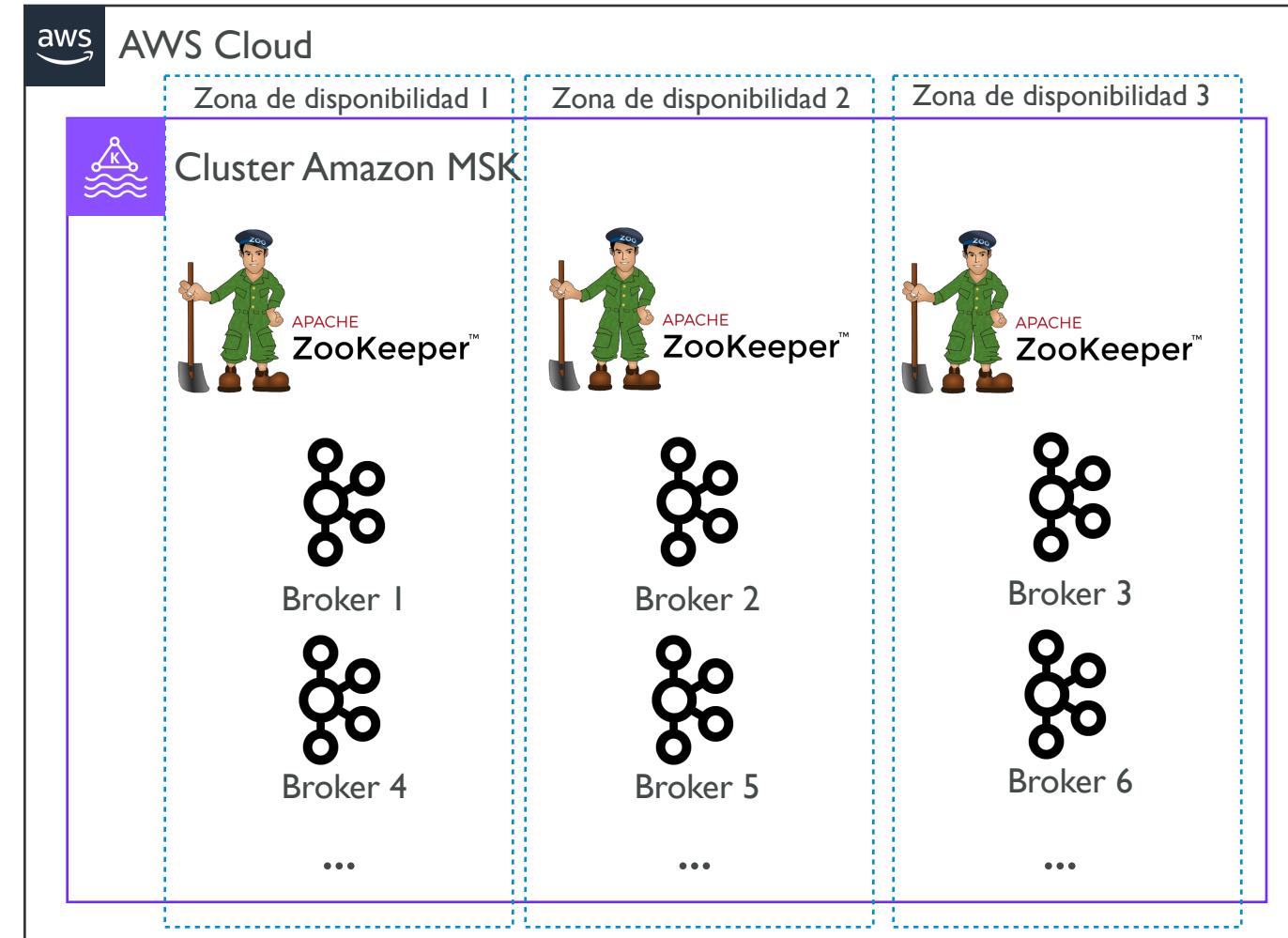


AWS Cloud



# Configuraciones del Clúster MSK

- Elige el número de Zonas de Disponibilidad (AZ)
  - Se recomienda 3
- Selecciona la VPC y subredes
- Tipo de instancia del broker
  - Ej: kafka.m5.large
- Número de brokers por AZ
  - Puedes añadir más brokers más tarde
- Tamaño de los volúmenes EBS
  - 1GB - 16TB



# Amazon MSK – Seguridad

- **Cifrado:**

- Opcional en tránsito utilizando TLS entre los brokers
- Opcional en tránsito utilizando TLS entre los clientes y brokers
- En reposo para tus volúmenes EBS utilizando KMS

- **Seguridad de red:**

- Autoriza grupos de seguridad específicos para tus clientes de Apache Kafka

- **Autenticación y autorización (importante):**

- Define quién puede leer/escribir en qué topics
- Mutual TLS (AuthN) + Kafka ACLs (AuthZ)
- SASL/SCRAM (AuthN) + Kafka ACLs (AuthZ)
- Control de acceso IAM (AuthN + AuthZ)



# Amazon MSK – Monitoreo

- **CloudWatch Metrics:** Utiliza Amazon CloudWatch para monitorear y recolectar métricas de rendimiento
  -  Monitoreo básico: Proporciona una visión general del rendimiento del clúster y los brokers
  -  Monitoreo mejorado: Ofrece métricas detalladas de los brokers, permitiendo un análisis más profundo
  -  Monitoreo a nivel de tema: Facilita el seguimiento de métricas específicas de los temas, lo que ayuda a identificar problemas a nivel granular
- **Entrega de Logs del Broker:** Permite la entrega de logs de los brokers a varios destinos para análisis y almacenamiento.



Amazon S3



Amazon Kinesis  
Data Streams



Amazon  
CloudWatch

- **Prometheus:** Herramienta de monitoreo de código abierto que se integra con Apache Kafka para exportar métricas



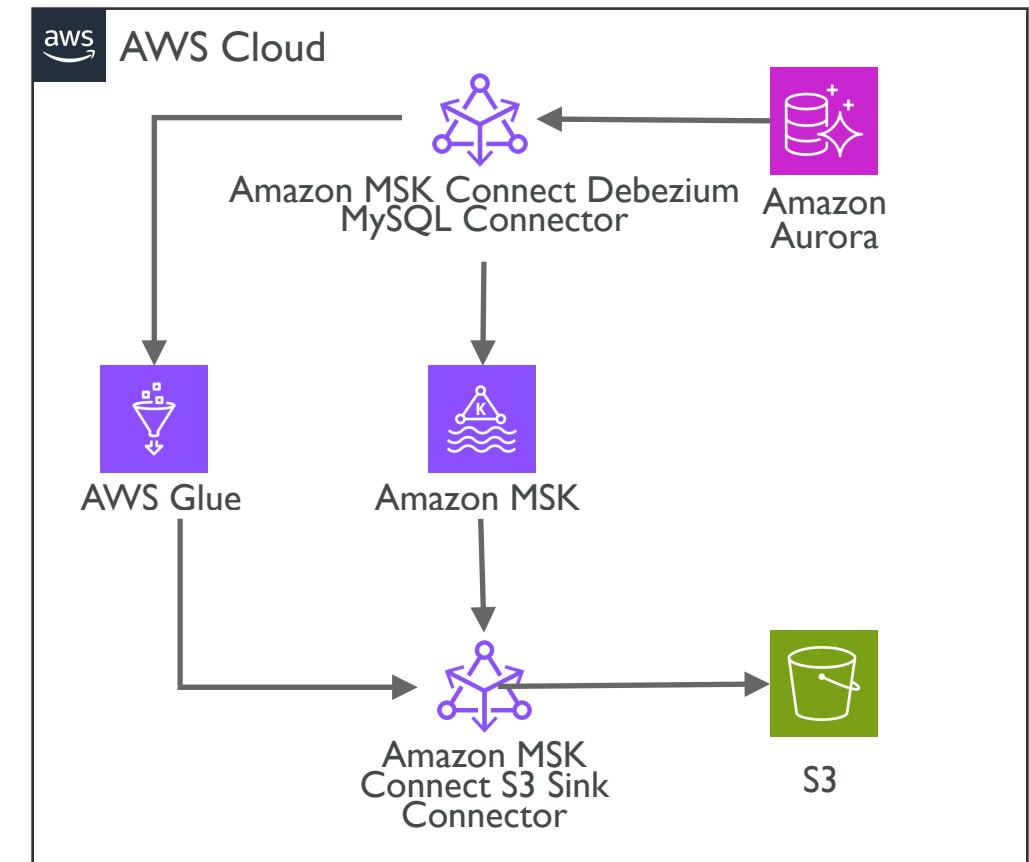
JMX Exporter: Captura métricas de Java Management Extensions (JMX) para monitorear la JVM



Node Exporter: Recolecta métricas de hardware y sistema operativo, como uso de CPU y disco

# Amazon MSK Connect

- Funcionalidad de Amazon MSK que **permite integrar** clústeres de Apache Kafka con múltiples sistemas de destino y origen de datos de manera eficiente
- Los trabajadores de Kafka Connect se gestionan completamente en AWS, simplificando la administración
- Escalado dinámico según la carga de trabajo para asegurar un rendimiento óptimo
- Puedes desplegar cualquier conector de Kafka Connect en MSK Connect como un plugin
  - Conectores disponibles: Amazon S3, Amazon Redshift, Amazon OpenSearch, Debezium, entre otros



# Amazon MSK Serverless

- MSK Serverless **elimina la necesidad de administrar** manualmente los recursos del clúster de Kafka
- Ajusta automáticamente los recursos según la demanda, asegurando un rendimiento óptimo
- Configuración sencilla, simplemente define los temas y particiones necesarios
- Utiliza AWS IAM para gestionar permisos y accesos de manera segura



# Kinesis Data Streams vs Amazon MSK



Amazon Kinesis Data Streams

- 1 MB límite de tamaño de mensaje
- Flujos de datos con fragmentos (Shards)
- División y fusión de fragmentos
- Cifrado en tránsito con TLS
- Seguridad:
  - Gestión de accesos y permisos mediante AWS IAM



Amazon MSK

- 1 MB por defecto, configurable (ej: 10 MB)
- Los datos se organizan en temas y particiones para facilitar el manejo y procesamiento
- Las particiones se añaden a temas ya existentes
- Cifrado en tránsito
- Seguridad:
  - Autenticación mutua y listas de control de acceso de Kafka



# Amazon OpenSearch Service

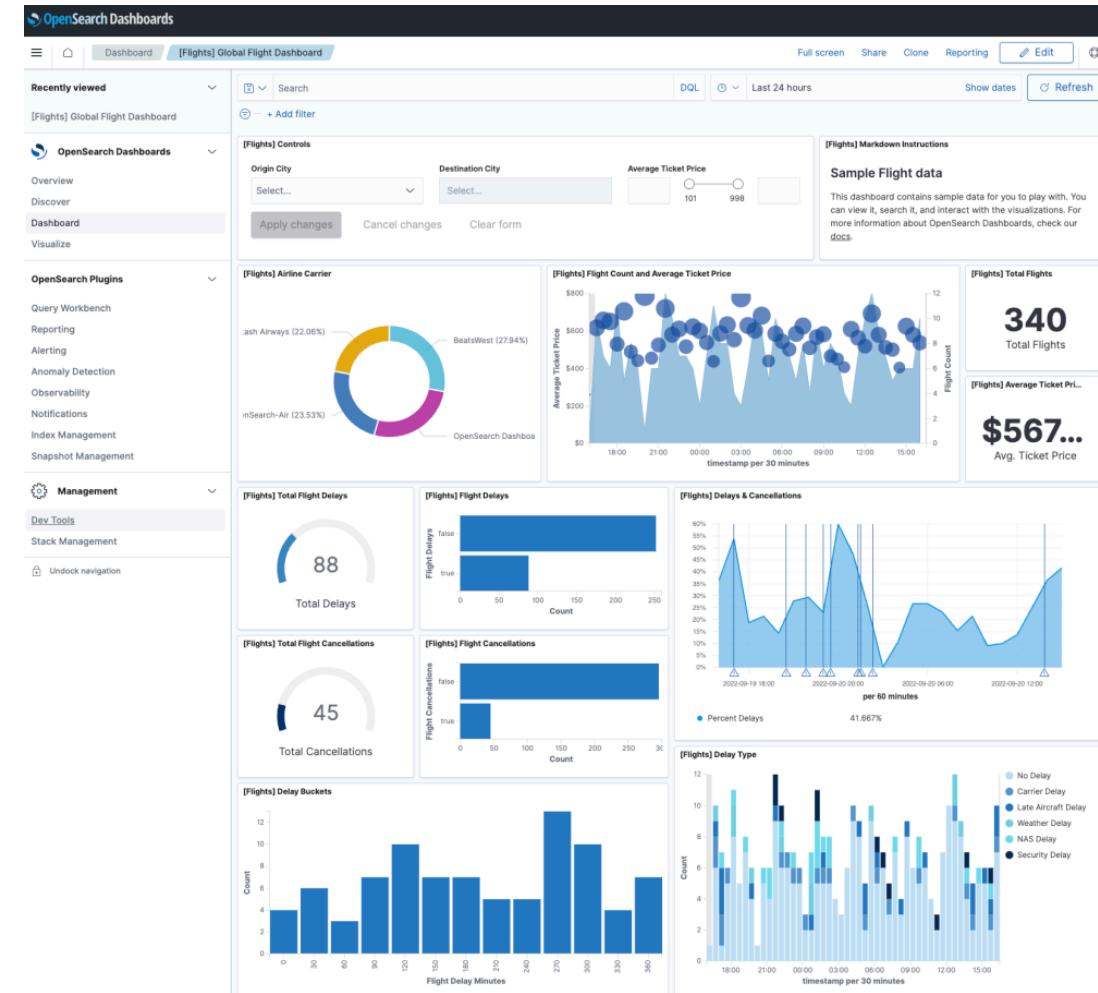
[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)



# Visión general de Amazon OpenSearch

- OpenSearch es un fork de Elasticsearch (iniciado por AWS)
- Servicio gestionado que facilita la **búsqueda, visualización y análisis** de grandes volúmenes de datos en tiempo real
- Incluye **OpenSearch Dashboards**, una herramienta de visualización integrada para analizar y monitorear datos
- Casos de uso:
  - Búsqueda de texto de forma rápida
  - Monitoreo de aplicaciones
  - Gestión de seguridad e información de eventos

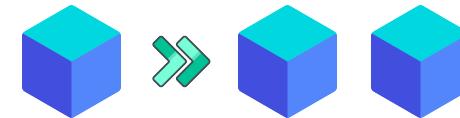


# Características principales de Amazon OpenSearch



## Búsqueda y análisis

Proporciona un motor de búsqueda potente y herramientas de análisis para diversas aplicaciones



## Escalabilidad

Permite la escalabilidad horizontal, facilitando la adición de nodos adicionales



## Seguridad

Características avanzadas de seguridad como encriptación, autenticación, autorización y auditoría



## Visualización

OpenSearch Dashboards ofrece una interfaz intuitiva para la visualización de datos

# Conceptos de OpenSearch [1/2]

## Documentos

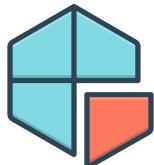


Son las unidades básicas de información en OpenSearch. Se almacenan en formato JSON y contienen datos estructurados que pueden ser buscados y analizados.

## Índices

- 1 Son colecciones de documentos. Los índices permiten organizar, buscar y analizar los documentos almacenados
- 2
- 3

## Shards (Fragmentos)



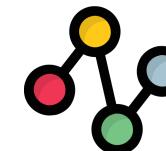
Los índices en OpenSearch se dividen en fragmentos (shards) para distribuir la carga de trabajo y mejorar el rendimiento.

## Clústeres



Un conjunto de uno o más nodos (servidores) que trabajan juntos para almacenar datos y proporcionar capacidades de búsqueda y análisis.

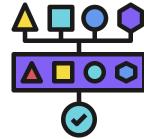
## Nodos



Es una instancia de OpenSearch que forma parte de un cluster. Los nodos pueden tener diferentes roles, como nodo principal, nodo de datos o nodo de coordinación, dependiendo de su configuración y propósito.

# Conceptos de OpenSearch [2/2]

## Pipeline de datos



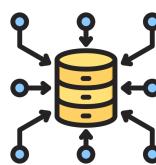
Se refiere a la secuencia de pasos y procesos que transforman y transportan datos desde su origen hasta su destino en OpenSearch

## OpenSearch Dashboards



Es una herramienta de visualización de datos que permite crear y compartir dashboards interactivos

## Agregaciones



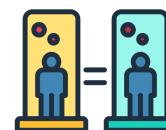
Son operaciones de análisis de datos que permiten calcular estadísticas y métricas, agrupar datos, y realizar operaciones avanzadas como análisis de series temporales

## Consulta (Query)



Son solicitudes de búsqueda realizadas al cluster de OpenSearch para recuperar y analizar datos

## Rélicas



Son copias adicionales de los fragmentos de un índice. Se utilizan para asegurar la disponibilidad de los datos y mejorar la tolerancia a fallos

# Anti-patrones de Amazon OpenSearch

- **OLTP (Procesamiento de Transacciones en Línea)**

- Sin transacciones: OpenSearch no está diseñado para manejar transacciones ACID completas
- RDS o DynamoDB es mejor: Para aplicaciones que requieren transacciones completas

- **Consultas ad-hoc de Datos**

- Athena es mejor: OpenSearch no está optimizado para consultas ad-hoc de datos

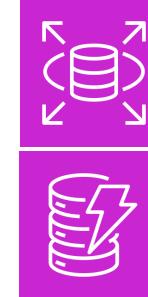
- **Recordar que OpenSearch es principalmente para búsqueda y análisis**

- OpenSearch está diseñado específicamente para proporcionar capacidades de búsqueda rápida y análisis de datos en tiempo real

OpenSearch



RDS



DynamoDB



OpenSearch



Athena



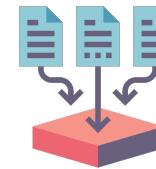
# Amazon OpenSearch Serverless

- Escalado automático bajo demanda



- Permite que el servicio escale automáticamente según las necesidades de carga de trabajo, sin intervención manual

- Funciona con "colecciones" en lugar de dominios aprovisionados



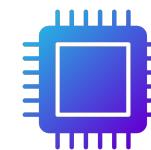
- Las colecciones pueden ser de tipo "búsqueda" o "series temporales"
- Facilita la organización y el acceso a los datos de manera más eficiente

- Siempre encriptado con tu clave KMS



- Define quién puede acceder a los datos
- Todos los datos almacenados están encriptados
- Permite establecer políticas de seguridad a través de múltiples colecciones

- Capacidad medida en Unidades de Cómputo de OpenSearch (OCUs)



- Puedes establecer un límite superior para controlar costos
- El límite inferior es siempre 2 OCUs para indexación y 2 OCUs para búsqueda, asegurando un mínimo de capacidad

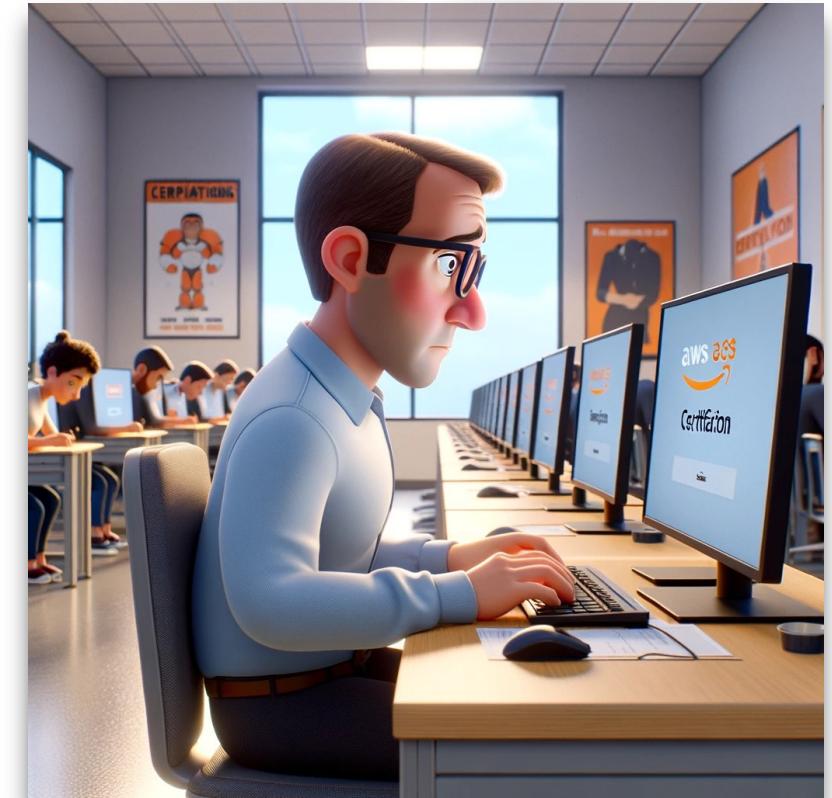
# Preparación del examen de certificación

# Tómate tu tiempo leyendo las preguntas

## Busca palabras clave sobre los requisitos

Una empresa de comercio electrónico desea **optimizar sus recomendaciones de productos** para los usuarios, utilizando un **modelo de aprendizaje automático** que se ajusta continuamente con datos de interacción. Aunque el volumen de datos es generalmente estable, se observan **picos significativos durante eventos de ventas** masivas como el Black Friday.

¿Qué arquitectura de sistema garantizaría una **adaptación eficiente y confiable** a estos cambios dinámicos en la carga de datos?



# Mantén el ritmo

- Tienes **130 minutos y 65 preguntas**, eso es solo 2 minutos por pregunta!
- Recomendaciones:
  - 🧘 Trata de **no estresarte**... eso es suficiente tiempo para leer y entender cada pregunta
  - 🤔 Asegúrate de entender bien cada alternativa antes de tomar una decisión
  - 🚫 Reduce tus opciones **descartando las respuestas claramente erróneas**
  - ⚡ Si no estás seguro, **marca la pregunta para revisarla más tarde** y avanza
  - ⏳ Si terminas antes de tiempo, utiliza los minutos restantes para revisar las preguntas marcadas y asegurarte de que no dejaste ninguna respuesta en blanco





# ¡Enhorabuena!

[www.blockstellart.com](http://www.blockstellart.com)

Todos los derechos reservados © BLOCKSTELLART [www.blockstellart.com](http://www.blockstellart.com)

# Rutas de certificaciones

## BÁSICO

Certificación basada en conocimientos para obtener conocimiento básico de la nube de AWS.

**No se necesita experiencia previa.**



## ASSOCIATE

Certificaciones basadas en roles que demuestran su conocimiento y habilidades de AWS y que construyen su credibilidad como profesional de la nube de AWS. **Se recomienda tener experiencia previa sólida en TI local o en la nube.**



## PROFESIONAL

Certificaciones basadas en roles que validan habilidades y conocimientos avanzados necesarios para diseñar aplicaciones seguras, optimizadas y modernas, y automatizar procesos en AWS. **Se recomienda tener 2 años de experiencia previa en la nube de AWS**



## ESPECIALIZACIÓN

Aprenda a profundidad y posíóngase como un asesor de confianza para las partes interesadas o clientes de estas áreas estratégicas. **Consulte las guías de examen en las páginas de exámenes para saber la experiencia recomendada.**

