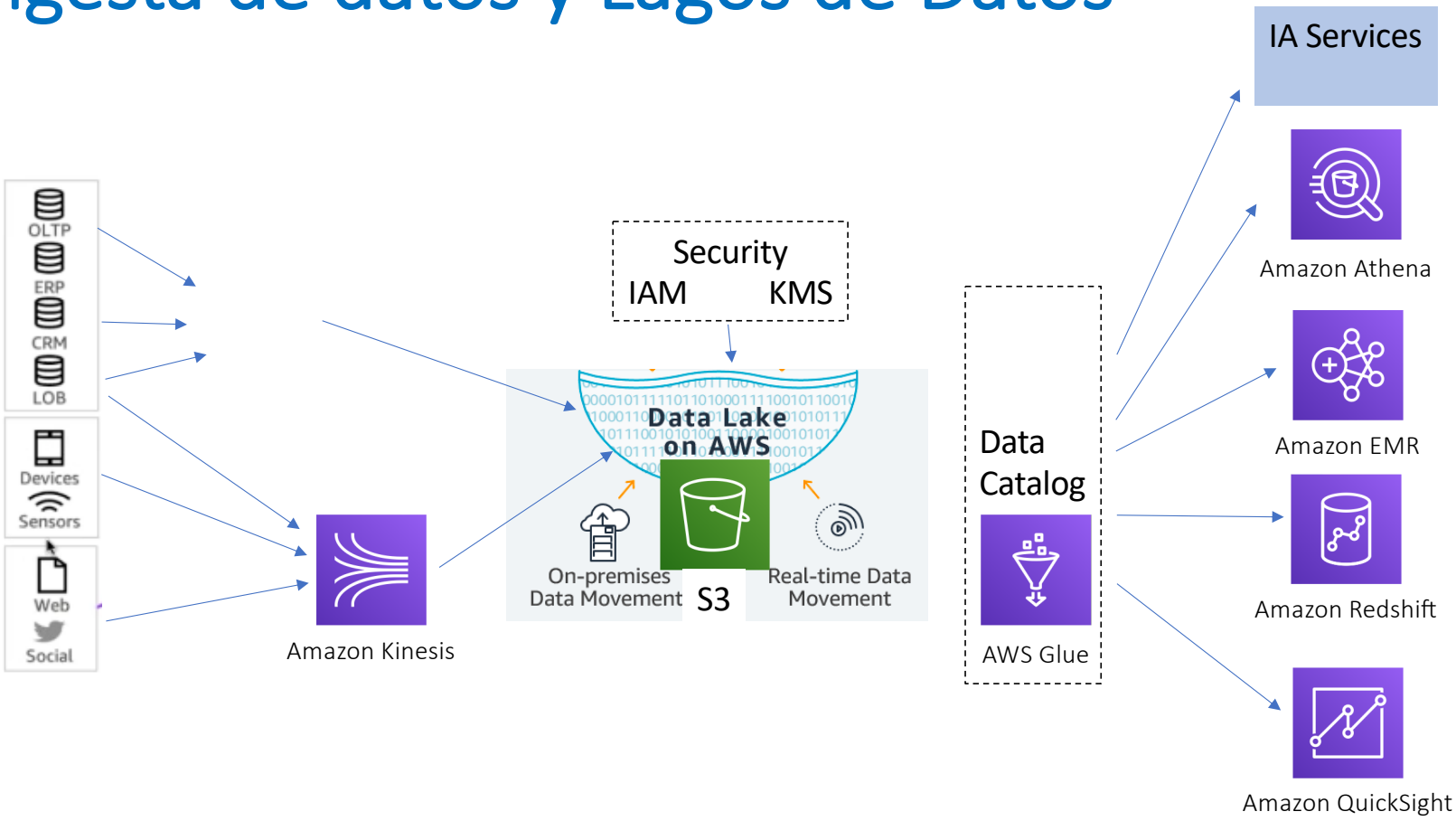
Datalake

Nueva tecnología de almacenamiento de datos de analítica

De Data warehouse hacia Datalakes.

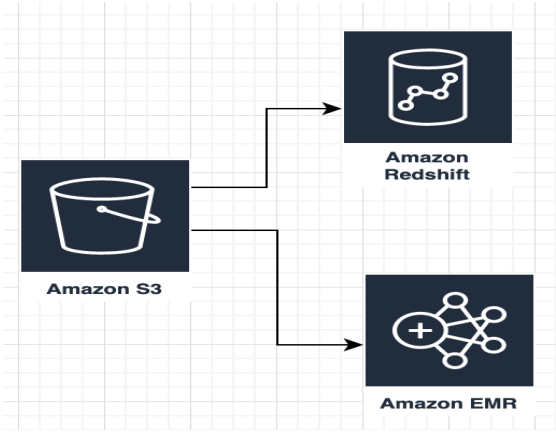
- Es una arquitectura para almacenar y analizar datos masivos y tipos heterogeneos de datos.
- Beneficios:
  - Todos los datos en un solo sitio
  - Rapida ingesta y transformación
  - Esquema en lectura

Ingesta de datos y Lagos de Datos



AWS S3 para datalake

- Los data lakes en nube, separan el ALMACENAMIENTO del PROCESAMIENTO (contrario a la idea original de hadoop/hdfs)
- S3 como una capa de almacenamiento, no como en hadoop ni DWH.
- Como almacenamiento separado tiene ventajas:
  - Arq de procesamiento temporal.
  - Lambda, athena, redshift, glue, etc.
- Algunos casos:
  - S3 como fuente para EMR
  - S3 como fuente para Redshift.



AWS S3

- Durable y confiable (11 9’s)
- Seguro
- Alto rendimiento
- Escalable
- Muchas interfases de acceso
- Integrable como muchos servicios amazon. (ej: EMR, athena, etc)

Storage – S3

- Buckets
  - Amazon S3 permite almacenar objetos (archivos) en ‘buckets’ (directorios)
  - Los buckets tienen nombres únicos globales (URI/URL)
  - Los buckets son a nivel de región

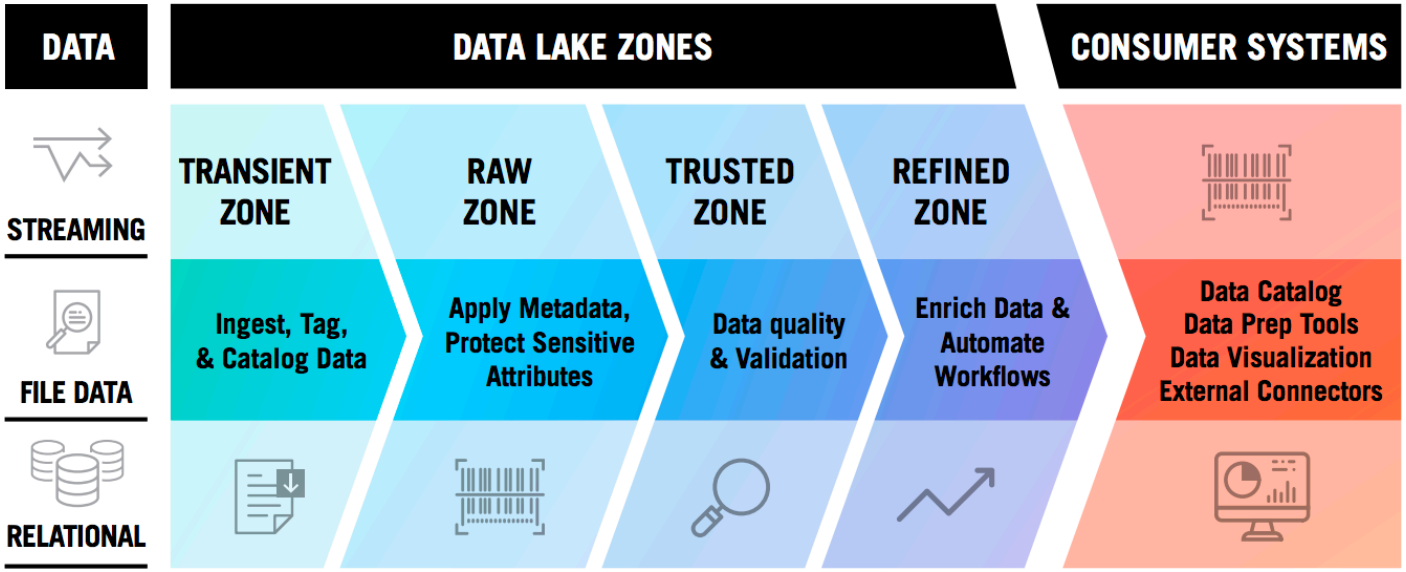
Ciclo de vida del data lake

- Ingesta
- Catalogo

- Ingesta:
  - Snowball & snowmobile
  - Starage gateway
  - Kinesis firehose
  - Direct connect
  - Many connectors: sqoop, flume, kafka, etc

Zonas

- Transcient
- Raw
- Trusted
- Refined



https://dzone.com/articles/data-lake-governance-best-practices

Catalogación en datalakes

- Contrario a un data warehouse, en un datalake se almacenan los datos SIN esquema.
- Se trabaja con Schema on Read
- Se requiere adicionar metadatos a los datos, estructurados, no estructurados y semi-estructurados.
  - Apache Atlas o AWS Glue nos da esta funcionalidad.

AWS Glue

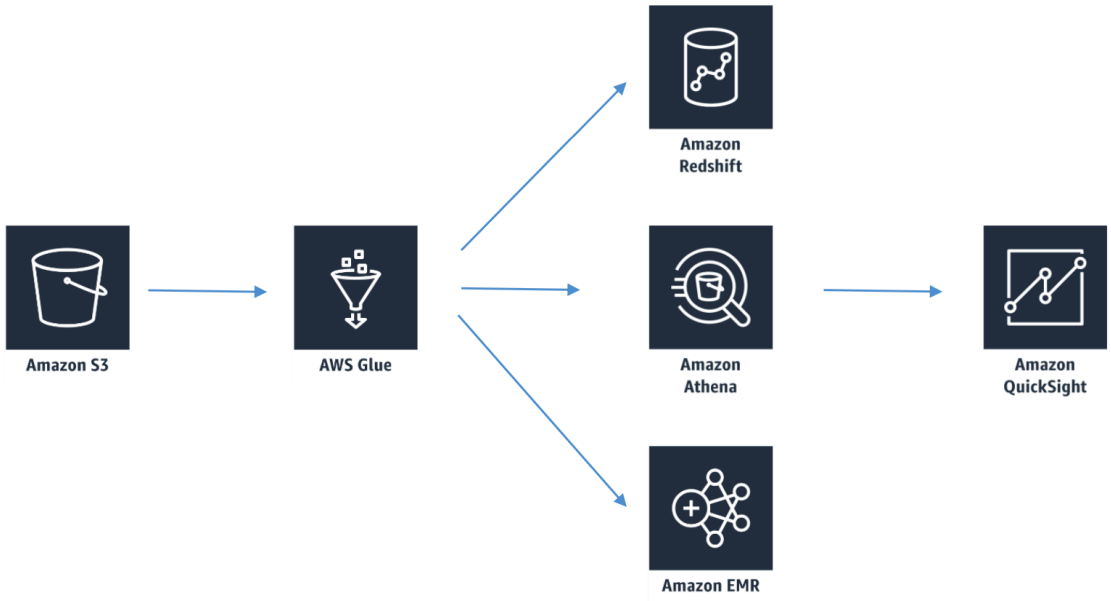
AWS Glue

Que es Glue

- Descubrimiento sin servidor y definición de esquemas y definiciones de tablas
  - S3 “data lakes”
  - RDS
  - Redshift
  - La mayoría de las otras bases de datos SQL
- Trabajos ETL personalizados
  - Impulsado por disparadores, en un horario o bajo demanda
  - Totalmente administrado

AWS Glue

Glue Crawler / Catálogo de datos



Glue y particiones

AWS Glue

- Glue crawler extraerá particiones en función de cómo se organizan los datos de S3
- Piense de antemano sobre cómo va a consultar su lago de datos en S3
- Ejemplo: los dispositivos envían datos del sensor cada hora
- ¿Consulta principalmente por intervalos de tiempo?
  - Si es así, organice sus buckets como yyyy/mm/dd/device
- ¿Consulta principalmente por dispositivo?
  - Si es así, organice su buckets como device/yyyy/mm/dd

AWS Glue ETL

AWS Glue

- Generación automática de código
- Scala o Python
- Cifrado
  - Server-side (at rest)
  - SSL (in transit)
- Puede estar controlado por eventos
- Puede aprovisionar “DPU’s” (data processing units) para aumentar el rendimiento de los trabajos subyacentes de Spark
- Errores notificados a CloudWatch

AWS Glue ETL

- Transformar datos, Limpiar datos, Enriquecer datos (antes de hacer análisis)
  - Generar código ETL en Python o Scala, puede modificar el código
  - Puede proporcionar sus propios scripts de Spark o PySpark
  - El destino puede ser S3, JDBC (RDS, Redshift) o en Glue Data Catalog
- Totalmente administrado, rentable, pagar sólo por los recursos consumidos
- Los trabajos se ejecutan en una plataforma Spark sin servidor
- Glue Scheduler para planificar los trabajos (jobs)
- Glue Triggers para automatizar las ejecuciones de trabajos basadas en "eventos"

Glue ETL- Transformaciones

AWS Glue

- Transformaciones agrupadas:
  - DropFields, DropNullFields – eliminar Campos (null)
  - Filter – especificar una función para filtrar registros
  - Join – para enriquecer los datos
  - Map - añadir campos, eliminar campos, realizar búsquedas externas
- Transformaciones Machine Learning:
  - FindMatches ML: identificar registros duplicados o coincidentes en el conjunto de datos, incluso cuando los registros no tienen un identificador único común y ningún campo coincide exactamente.
- Conversiones de formato: CSV, JSON, Avro, Parquet, ORC, XML
- Transformaciones Apache Spark (Ejemplo: K-Means)

Amazon Athena

AWS Athena

AWS Athena

Amazon Athena

- Serverless Interactive queries of S3 data (SQL)
  - No requiere cargar datos, permanecen en S3
- Sin servidor
- Soporta muchos formatos
  - CSV, JSON, ORC, Parquet, Avro
- Datos no-estructurados, semi-estructurados, estructurados

Usos

Amazon Athena

- Consultas ad-hoc sobre logs web
- Consultas ’temporales’ antes de cargar a redshift.
- Integración con jupyter, zeppelin, rstudio
- Integración con QuickSight
- Integración vía ODBC /JDBC

AWS Athena + Glue

Amazon Athena