Data Engineering - ML

Tuesday, 26 July 2022 11:53

Amazon S3

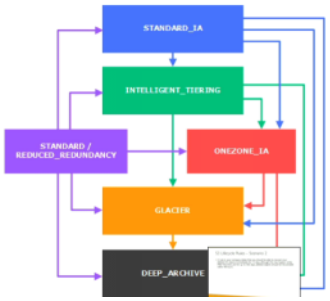
- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a **globally unique name**
- Objects (files) have a Key. The key is the **FULL** path:
  - <my_bucket>/my_file.txt
  - <my_bucket>/my_folder1/another_folder/my_file.txt
- This will be interesting when we look at **partitioning**
- Max object size is 5TB
- Object Tags (key / value pair – up to 10) – useful for security / lifecycle

AWS S3 for Machine Learning

- Backbone for many AWS ML services (example: SageMaker)
- Create a "Data Lake"
  - Infinite size, no provisioning
  - 99.999999999% durability
  - Decoupling of storage (S3) to compute (EC2, Amazon Athena, Amazon Redshift Spectrum, Amazon Rekognition, and AWS Glue)
- Centralized Architecture
- Object storage => supports any file format
- Common formats for ML: CSV, JSON, Parquet, ORC, Avro, Protobuf

Amazon S3 Lifecycle - Transition

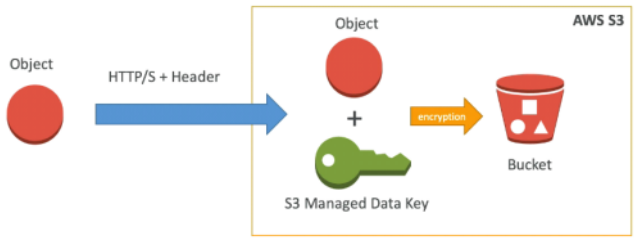
- You can transition objects between storage classes
- For infrequently accessed object, move them to STANDARD_IA
- For archive objects you don't need in real-time, GLACIER or DEEP_ARCHIVE
- Moving objects can be automated using a **lifecycle configuration**



Amazon S3 Encryption

- There are 4 methods of encrypting objects in S3
- SSE-S3: encrypts S3 objects using keys handled & managed by AWS
- SSE-KMS: use AWS Key Management Service to manage encryption keys
  - Additional security (user must have access to KMS key)
  - Audit trail for KMS key usage
- SSE-C: when you want to manage your own encryption keys
- Client Side Encryption
- From an ML perspective, SSE-S3 and SSE-KMS will be most likely used

SSE-S3



Amazon S3 Security

- User based
  - IAM policies - which API calls should be allowed for a specific user
- Resource Based
  - **Bucket Policies** - bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain

Amazon S3

S3 Durability and Availability

- Durability:
  - High durability (99.999999999%, 11 9's) of objects across multiple AZ
  - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
  - Same for all storage classes
- Availability:
  - Measures how readily available a service is
  - Varies depending on storage class
  - Example: S3 standard has 99.99% availability = not available 53 minutes a year

AWS S3 Data Partitioning

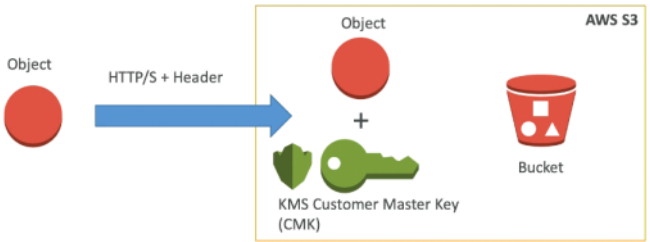
- Pattern for speeding up range queries (ex: AWS Athena)
- By Date: s3://bucket/my-data-set/year/month/day/hour/data_00.csv
- By Product: s3://bucket/my-data-set/product-id/data_32.csv
- You can define whatever partitioning strategy you like!
- Data partitioning will be handled by some tools we use (e.g. AWS Glue)

S3 Lifecycle rules

- **Transition actions:** It defines when objects are transitioned to another storage class.
  - Move objects to Standard IA class 60 days after creation
  - Move to Glacier for archiving after 6 months
- **Expiration actions:** configure objects to expire (delete) after some time
  - Access log files can be set to delete after a 365 days
  - Can be used to delete old versions of files (if versioning is enabled)
  - Can be used to delete incomplete multi-part uploads
- Rules can be created for a certain prefix (ex - s3://mybucket/mp3/*)
- Rules can be created for certain objects tags (ex - Department: Finance)

Amazon S3 Encryption

SSE-KMS



Amazon S3 Security

S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal: The account or user to apply the policy to

- Resource Based
  - Bucket Policies - bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain
  - Bucket Access Control List (ACL) – less common

- Actions: Set of API to Allow or Deny
- Effect: Allow / Deny
- Principal: The account or user to apply the policy to

- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

S3 Default Encryption vs Bucket Policies

- The old way to enable default encryption was to use a bucket policy and refuse any HTTP command without the proper headers:

```
"Condition": {
        "StringNotEquals": {
              "s3:x-amz-server-side-encryption": "AES256"
        }
}
```

- The new way is to use the "default encryption" option in S3

```
"Condition": {
        "Null": {
              "s3:x-amz-server-side-encryption": true
        }
}
```

S3- Security (EXAM)

- Networking - VPC Endpoint Gateway:
  - Allow traffic to stay within your VPC (instead of going through public web)
  - Make sure your private services (AWS SageMaker) can access S3
  - Very important for AWS ML Exam

- Logging and Audit:
  - S3 access logs can be stored in other S3 bucket
  - API calls can be logged in AWS CloudTrail

- Tagged Based (combined with IAM policies and bucket policies)
  - Example: Add tag Classification=PHI to your objects

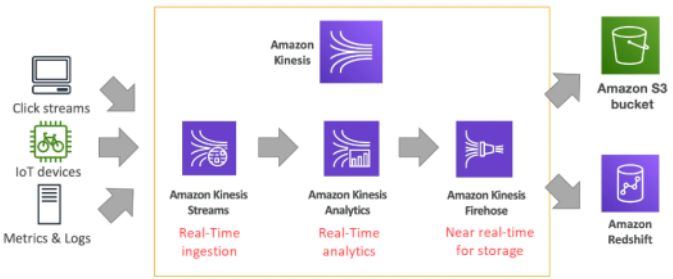
AWS Kinesis Overview

- Kinesis is a managed alternative to Apache Kafka
- Great for application logs, metrics, IoT, clickstreams
- Great for "real-time" big data
- Great for streaming processing frameworks (Spark, NiFi, etc…)
- Data is automatically replicated synchronously to 3 AZ

- Kinesis Streams: low latency streaming ingest at scale
- Kinesis Analytics: perform real-time analytics on streams using SQL
- Kinesis Firehose: load streams into S3, Redshift, ElasticSearch & Splunk
- Kinesis Video Streams: meant for streaming video in real-time

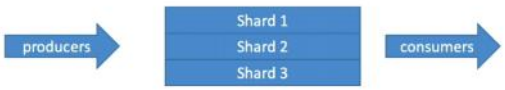
AWS Kinesis Overview

Architecture



Kinesis (Data) Streams Overview : Real-Time Streaming, Broadcast

- Streams are divided in ordered Shards / Partitions



- Data retention is 24 hours by default, can go up to 365 days
- Ability to reprocess / replay data
- Multiple applications can consume the same stream
- Once data is inserted in Kinesis, it can't be deleted (immutability)
- Records can be up to 1MB in size

Kinesis (Data) Streams - Capacity Mode: Provisioned, On-Demand

- Provisioned mode:
  - You choose the number of shards provisioned, scale manually or using API
  - Each shard gets 1MB/s in (or 1000 records per second)
  - Each shard gets 2MB/s out (classic or enhanced fan-out consumer)
  - You pay per shard provisioned per hour

- On-demand mode:
  - No need to provision or manage the capacity
  - Default capacity provisioned (4 MB/s in or 4000 records per second)
  - Scales automatically based on observed throughput peak during the last 30 days
  - Pay per stream per hour & data in/out per GB

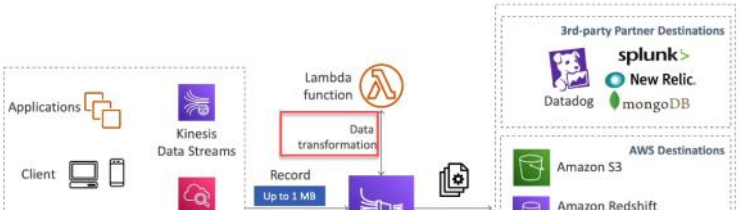
Kinesis Data Streams Limits

- Producer:
  - 1MB/s or 1000 messages/s at write PER SHARD
  - "ProvisionedThroughputException" otherwise
- Consumer Classic:
  - 2MB/s at read PER SHARD across all consumers
  - 5 API calls per second PER SHARD across all consumers
- Data Retention:
  - 24 hours data retention by default
  - Can be extended to 365 days

Kinesis Data Firehose: Near Real-Time Streamin, Ingest & Store, Auto-Scale

- Fully Managed Service, no administration
- Near Real Time (60 seconds latency minimum for non full batches)
- Data Ingestion into Redshift / Amazon S3 / ElasticSearch / Splunk
- Automatic scaling
- Supports many data formats
- Data Conversions from CSV / JSON to Parquet / ORC (only for S3)

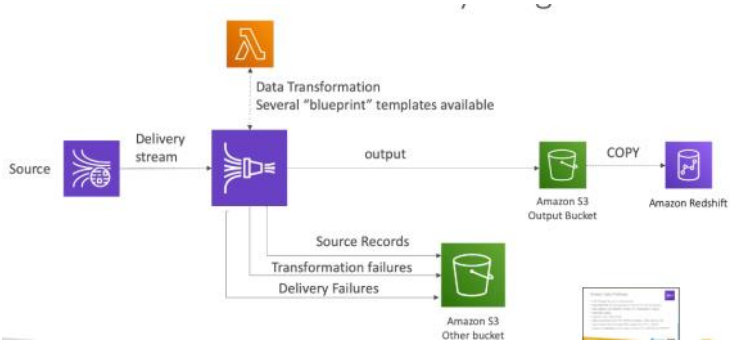
Kinesis Data Firehose - Architecture



- Automatic scaling
- Supports many data formats
- Data Conversions from CSV / JSON to Parquet / ORC (only for S3)
- Data Transformation through AWS Lambda (ex: CSV => JSON)
- Supports compression when target is Amazon S3 (GZIP, ZIP, and SNAPPY)
- Pay for the amount of data going through Firehose

Kinesis Data Streams vs Firehose

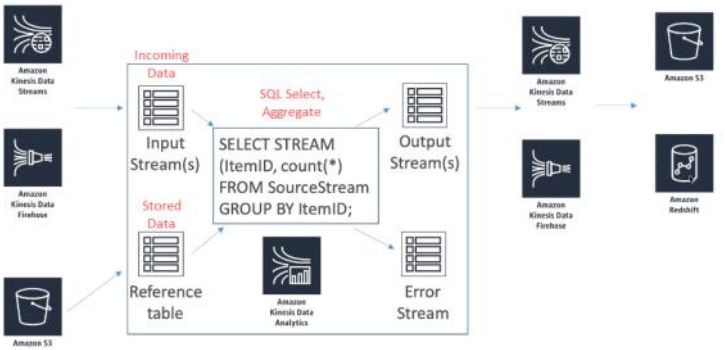
- Streams
  - Going to write custom code (producer / consumer)
  - Real time (~200 ms latency for classic, ~70 ms latency for enhanced fan-out)
  - Automatic scaling with On-demand Mode
  - Data Storage for 1 to 365 days, replay capability, multi consumers
- Firehose
  - Fully managed, send to S3, Splunk, Redshift, ElasticSearch
  - Serverless data transformations with Lambda
  - Near real time (lowest buffer time is 1 minute)
  - Automated Scaling
  - No data storage



Kinesis Data Analytics - SQL or Flink Live Analytics, Serverless

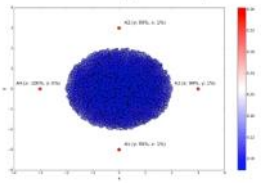
- Use cases
  - Streaming ETL: select columns, make simple transformations, on streaming data
  - Continuous metric generation: live leaderboard for a mobile game
  - Responsive analytics: look for certain criteria and build alerting (filtering)
- Features
  - Pay only for resources consumed (but it's not cheap)
  - Serverless; scales automatically
  - Use IAM permissions to access streaming source and destination(s)
  - SQL or Flink to write the computation
  - Schema discovery
  - Lambda can be used for pre-processing

Kinesis Data Analytics



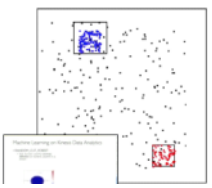
Machine Learning on KDA - RANDOM CUT FOREST: Anomaly RECENT HISTORY

- RANDOM_CUT_FOREST
  - SQL function used for anomaly detection on numeric columns in a stream
  - Example: detect anomalous subway ridership during the NYC marathon
  - Uses recent history to compute model



Machine Learning on KDA - HOTSPOTS, DENSE REGIONS

- HOTSPOTS
  - locate and return information about relatively dense regions in your data
  - Example: a collection of overheated servers in a data center

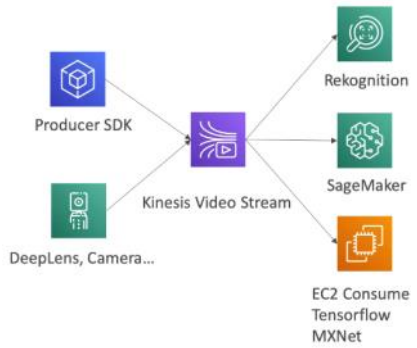


Kinesis Video Streams - 1 Stream per producer

- Producers:
  - security camera, body-worn camera, AWS DeepLens, smartphone camera, audio feeds, images, RADAR data, RTSP camera.
  - One producer per video stream
- Video playback capability
- Consumers
  - build your own (MXNet, Tensorflow)
  - AWS SageMaker
  - Amazon Rekognition Video
- Keep data for 1 hour to 10 years

- 1

Kinesis Video Streams - Architecture



Kinesis Summary

Use-Case Architecture

- Keep data for 1 hour to 10 years

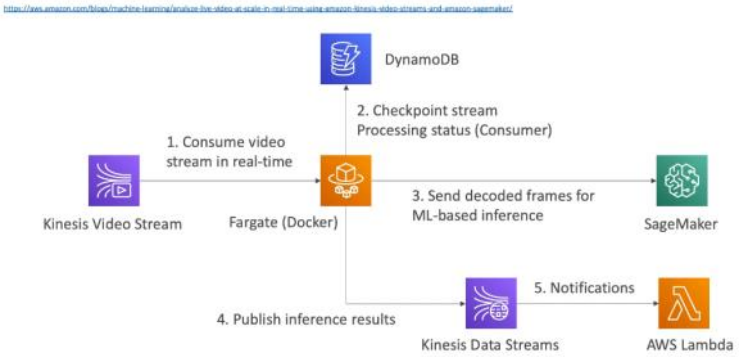
- 1

Kinesis Summary

- Kinesis Data Stream: create real-time machine learning applications
- Kinesis Data Firehose: ingest massive data near-real time
- Kinesis Data Analytics: real-time ETL / ML algorithms on streams
- Kinesis Video Stream: real-time video stream to create ML applications

MXNet

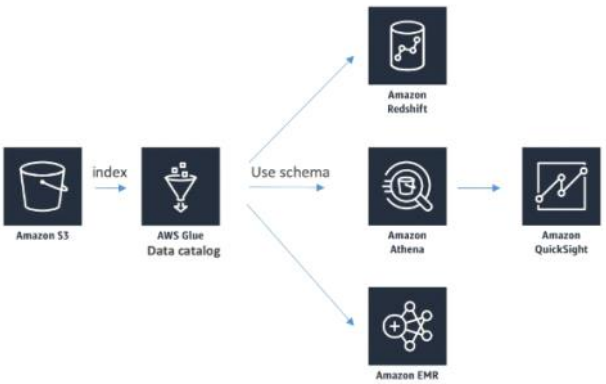
Use-Case Architecture



GLUE Data Catalog - Schema Metadata for Data Sources

- Metadata repository for all your tables
  - Automated Schema Inference
  - Schemas are versioned
- Integrates with Athena or Redshift Spectrum (schema & data discovery)
- Glue Crawlers can help build the Glue Data Catalog

GLUE Data Catalog



GLUE Crawlers - Infer Schema for Data Sources

- Crawlers go through your data to infer schemas and partitions
- Works JSON, Parquet, CSV, relational store
- Crawlers work for: S3, Amazon Redshift, Amazon RDS
- Run the Crawler on a Schedule or On Demand

- Need an IAM role / credentials to access the data stores

Glue and S3 Partitions

- Glue crawler will extract partitions based on how your S3 data is organized
- Think up front about how you will be querying your data lake in S3
- Example: devices send sensor data every hour
- Do you query primarily by time ranges?
  - If so, organize your buckets as s3://my-bucket/dataset/yyyy/mm/dd/device
- Do you query primarily by device?
  - If so, organize your buckets as s3://my-bucket/dataset/device/yyyy/mm/dd

GLUE ETL - Serverless ETL Spark Platform

- Transform data, Clean Data, Enrich Data (before doing analysis)
  - Generate ETL code in Python or Scala, you can modify the code
  - Can provide your own Spark or PySpark scripts
  - Target can be S3, JDBC (RDS, Redshift), or in Glue Data Catalog
- Fully managed, cost effective, pay only for the resources consumed
- Jobs are run on a serverless Spark platform

- Glue Scheduler to schedule the jobs
- Glue Triggers to automate job runs based on "events"

GLUE ETL Transformations - Bundled, ML, Format Conv., Spark

- Bundled Transformations:
  - DropFields, DropNullFields – remove (null) fields
  - Filter – specify a function to filter records
  - Join – to enrich data
  - Map - add fields, delete fields, perform external lookups
- Machine Learning Transformations:
  - FindMatches ML: identify duplicate or matching records in your dataset, even when the records do not have a common unique identifier and no fields match exactly.
- Format conversions: CSV, JSON, Avro, Parquet, ORC, XML
- Apache Spark transformations (example: K-Means)

Athena - Serverless Interactive SQL Query (Presto)

Covered later

AWS Data Stores for Machine Learning

- DynamoDB:
- S3:

AWS Data Stores for Machine Learning

OLAP Online Analytical Processing (Redshift) - Data Stored in Columns
OLTP Online Transaction Processing (RDS) - Data Stored in Rows

- DynamoDB:
  - NoSQL data store, serverless, provision read/write capacity
  - Useful to store a machine learning model served by your application

- S3:
  - Object storage
  - Serverless, infinite storage
  - Integration with most AWS Services

- Redshift:
  - Data Warehousing, SQL analytics (OLAP - Online analytical processing)
  - Load data from S3 to Redshift
  - Use Redshift Spectrum to query data directly in S3 (no loading)

- RDS, Aurora:
  - Relational Store, SQL (OLTP - Online Transaction Processing)
  - Must provision servers in advance

- OpenSearch (previously ElasticSearch):
  - Indexing of data
  - Search amongst data points
  - Clickstream Analytics

- ElastiCache:
  - Caching mechanism
  - Not really used for Machine Learning

AWS Data Pipelines - Pipeline Orchestrator with EC2

- ETL service to move data from one store to another
- Does not contain actual ETL code, orchestrate the code.

- Destinations include S3, RDS, DynamoDB, Redshift and EMR
- Manages task dependencies
- Retries and notifies on failures
- Data sources may be on-premises
- Highly available

AWS Glue vs Data Pipelines

- Glue:
  - Glue ETL - Run Apache Spark code, Scala or Python based, focus on the ETL
  - Glue ETL - Do not worry about configuring or managing the resources
  - Data Catalog to make the data available to Athena or Redshift Spectrum

- Data Pipeline:
  - Orchestration service
  - More control over the environment, compute resources that run code, & code
  - Allows access to EC2 or EMR instances (creates resources in your own account)

AWS Batch - Serverless, Docker based batch jobs

- Run batch jobs as Docker images
- Dynamic provisioning of the instances (EC2 & Spot Instances)
- Optimal quantity and type based on volume and requirements
- No need to manage clusters, fully **serverless**
- You just pay for the underlying EC2 instances

- Schedule Batch Jobs using CloudWatch Events
- Orchestrate Batch Jobs using AWS Step Functions

AWS Data Migration Service (DMS) - Database Migration

- Quickly and securely migrate databases to AWS, resilient, self healing
- The source database remains available during the migration
- Supports:
  - Homogeneous migrations: ex Oracle to Oracle
  - Heterogeneous migrations: ex Microsoft SQL Server to Aurora
- Continuous Data Replication using CDC
- You must create an EC2 instance to perform the replication tasks

AWS Data Pipelines - Architecture

AWS Glue vs Batch

- Glue:
  - Glue ETL - Run Apache Spark code, Scala or Python based, focus on the ETL
  - Glue ETL - Do not worry about configuring or managing the resources
  - Data Catalog to make the data available to Athena or Redshift Spectrum

- Batch:
  - For any computing job regardless of the job (must provide Docker image)
  - Resources are created in your account, managed by Batch
  - For any non-ETL related work, Batch is probably better

AWS Glue vs DMS

- Glue:
  - Glue ETL - Run Apache Spark code, Scala or Python based, focus on the ETL
  - Glue ETL - Do not worry about configuring or managing the resources
  - Data Catalog to make the data available to Athena or Redshift Spectrum

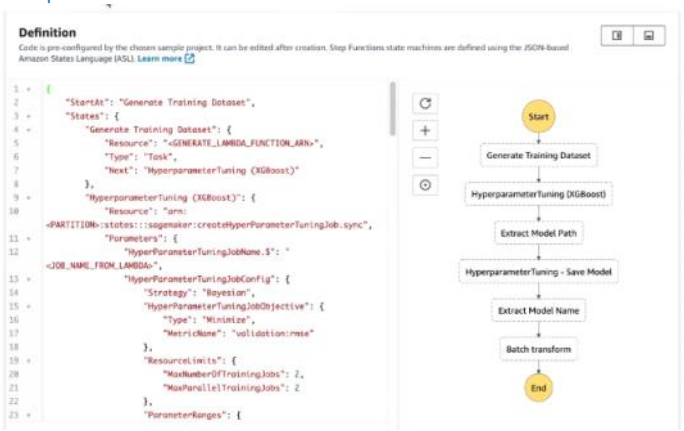
- AWS DMS:
  - Continuous Data Replication
  - No data transformation
  - Once the data is in AWS, you can use Glue to transform it

perform the replication tasks

AWS Step Functions - Visual Design Workflows (upto Wait 1 year)

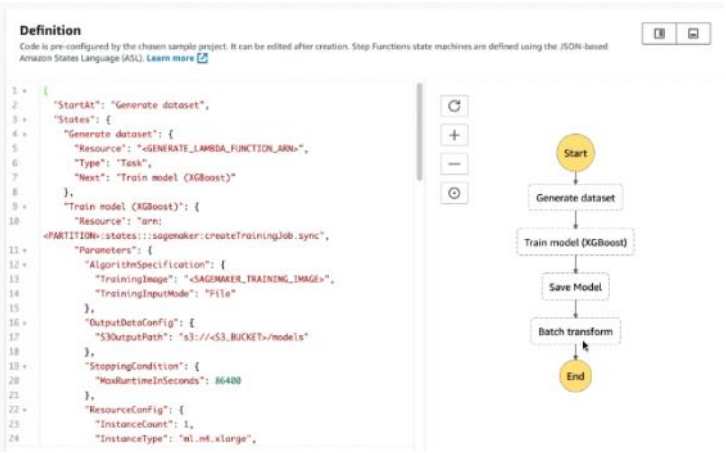
- Use to design workflows
- Easy visualizations
- Advanced Error Handling and Retry mechanism outside the code
- Audit of the history of workflows
- Ability to "Wait" for an arbitrary amount of time
- Max execution time of a State Machine is 1 year

Example - Tune ML Model



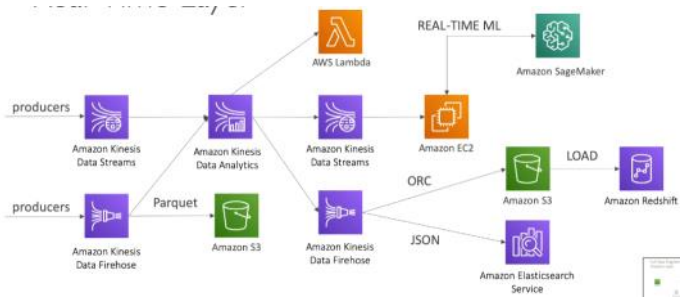
AWS Step Functions

Example - Train ML Model

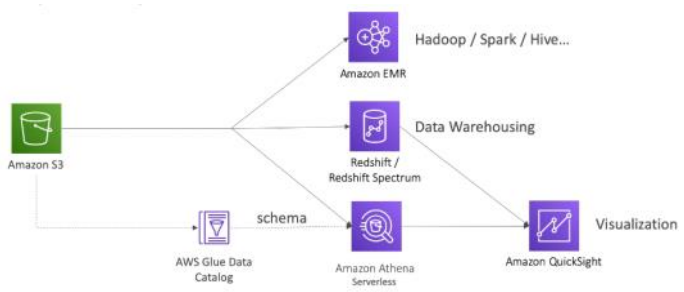


Full Data Engineering Pipelines

Real-Time Layer

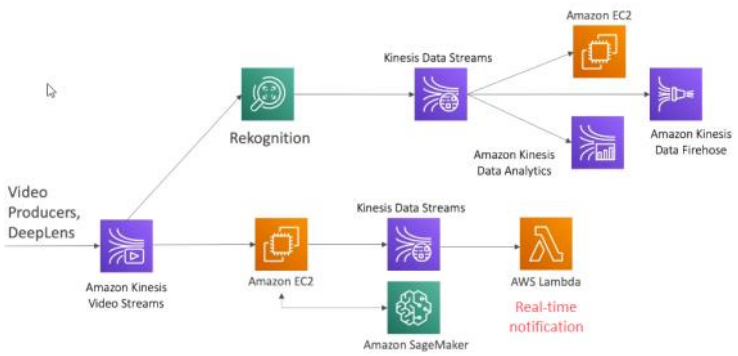


Analytics Layer

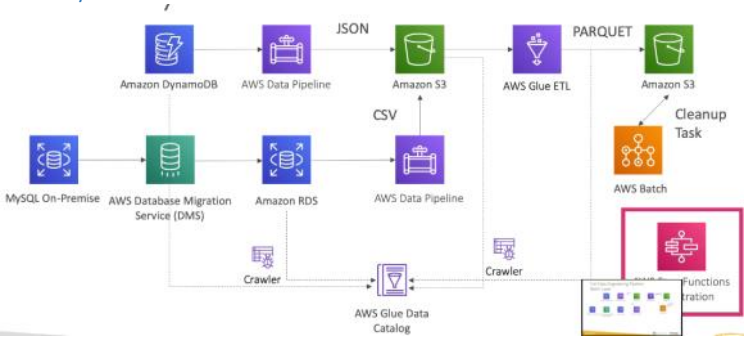


Full Data Engineering Pipelines

Video Layer



Batch Layer



Data Engineering Summary

Here's a quick summary of all the services we've mentioned

- **Amazon S3: Object Storage** for your data
- **VPC Endpoint Gateway: Privately access your S3 bucket** without going through the public internet
- **Kinesis Data Streams: real-time** data streams, need **capacity planning**, real-time applications
- **Kinesis Data Firehose: near real-time data ingestion** to **S3**, **Redshift**, **ElasticSearch**, **Splunk**

- **Kinesis Data Analytics: SQL transformations** on **streaming data**
- **Kinesis Video Streams: real-time video** feeds
- **Glue Data Catalog & Crawlers: Metadata repositories for schemas and datasets** in your account
- **Glue ETL: ETL Jobs as Spark programs**, run on a **serverless Spark Cluster**
- **DynamoDB: NoSQ**L store
- **Redshift: Data Warehousing for OLAP**, **SQL** language
- **Redshift Spectrum: Redshift on data in S3** (**without the need to load** it first in Redshift)
- **RDS / Aurora: Relational Data Store for OLTP**, **SQL** language
- **ElasticSearch: index for your data, search** capability, **clickstream** analytics
- **ElastiCache: data cache** technology
- **Data Pipelines: Orchestration of ETL jobs** between **RDS**, **DynamoDB**, **S3**. **Runs on EC2** instances
- **Batch: batch jobs run as Docker containers** - not just for data, manages EC2 instances for you
- **DMS: Database Migration Service, 1-to-1 CDC replication**, no ETL
- **Step Functions: Orchestration of workflows, audit**, **retry** mechanisms