Data Engineering Patterns

FIVE VS · INGESTION PATTERNS · GLUE ETL · KINESIS STREAMING · DATA LAKE & WAREHOUSE · EMR · ANALYTICS & VISUALIZATION

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

01 **Start from the business, weigh five Vs:** every data-infrastructure decision works **backward from the business outcome**. Judge data by its five Vs — **value, veracity, volume, velocity, variety** — considered *together*, not in sequence. Value and veracity decide whether insights are trustworthy; **volume and velocity together** drive the throughput and scaling your pipeline must handle.

02 **A pipeline has four layers: ingest → store → process → analyze**, with storage touching every layer. **Ingestion** copies data from a source store to a target store. **Homogeneous** ingestion moves data *as-is* (same format/engine, no transform); **heterogeneous** ingestion transforms it via **ETL** (transform *before* load → structured data → data warehouse) or **ELT** (load raw, transform later → unstructured data → data lake).

03 **Batch vs. streaming: batch** computes over *complete datasets* — high-volume, repetitive jobs run on a schedule, on demand, or on an event, off-peak, needing **high compute** (overnight sales reports). **Streaming** processes an *unbounded* sequence of small packets continuously for real-time analytics, needing **low compute** but reliable, low-latency networking (instant product recommendations). Pick by the use case, not the technology.

04 **Glue for batch, Kinesis for streams: AWS Glue** is the serverless data-integration/ETL engine — **Data Catalog** (metadata/schema only, never the data), **crawlers** (derive schema), **DataBrew** (no-code prep), **Data Quality**. The **Amazon Kinesis** family handles streams: **Data Firehose** (simplest, micro-batch delivery to S3/Redshift/OpenSearch, no code), **Kinesis Data Streams** (low latency, replay up to 365 days), **Managed Service for Apache Flink** (real-time analytics/anomaly detection), plus **Amazon MSK** (managed Kafka).

05 **One modern architecture, a data lake at the center:** store everything in a centralized, low-cost **data lake** (Amazon S3) and integrate a **data warehouse** and purpose-built stores around it with unified governance and seamless data movement. On AWS the lake = **S3** (storage) + **Lake Formation** (management/security/IAM) + **AWS Glue** (catalog/transform) + **Amazon Athena** (SQL access). Remember **schema-on-write = data warehouse**, **schema-on-read = data lake**.

02 KEY TERMS

TERM	DEFINITION
Five Vs of data	Value, veracity, volume, velocity, and variety — the data characteristics that, considered together, drive every infrastructure decision
Data ingestion	Extracting data from its source and loading it into the pipeline; copying data from a source data store to a target data store
Homogeneous ingestion	Moving data from source to destination as-is, keeping the same data format or storage-engine type — no transformation
ETL / ELT	Heterogeneous patterns: ETL transforms before loading (structured → data warehouse); ELT loads raw then transforms (unstructured → data lake)
Batch processing	Computes results over complete datasets in periodic, high-volume, repetitive jobs; high compute, often run off-peak
Streaming processing	Processes an unbounded, continuous, incremental sequence of small data packets for real-time analytics; low compute, low-latency network
AWS Glue	Serverless data-integration service that automates ETL for batch and streaming pipelines; includes Data Catalog, DataBrew, and Data Quality
Glue Data Catalog	Central store of metadata and schema for data sources and ETL targets — populated by crawlers; it stores no datasets, only definitions

TERM	DEFINITION
Data lake	Centralized, low-cost repository (Amazon S3) that stores all structured and unstructured data at any scale, as-is (schema-on-read)
Data warehouse	Database of curated, structured data as the single source of truth (Amazon Redshift); schema-on-write, for BI and reporting
Modern data architecture	Integrates a data lake, a data warehouse, and purpose-built stores with unified governance and seamless data movement — no silos
Parallel processing	Splitting a large dataset into parts, processing them concurrently, and aggregating results (Amazon EMR: Hadoop MapReduce, Apache Spark)

03 THE DATA PIPELINE, END TO END

storage touches every layer

01 Ingest — extract from source, load to target (AppFlow, DataSync, Data Exchange) → **02 Store** — data lake (S3), warehouse (Redshift), OLTP stores → **03 Process** — transform and aggregate (Glue, EMR, Firehose, Flink) → **04 Analyze** — query and visualize (Athena, QuickSight, OpenSearch)

04 BATCH VS. STREAMING PROCESSING

choose by the business use case

BATCH PROCESSING

Complete datasets, run periodically

- Computes results over **complete datasets**
- Runs on a schedule, on demand, or on an event — often off-peak
- Requires **high computing power**
- Deep analysis of large datasets; daily/weekly reporting
- Example: overnight sales reports sent to branches by morning

STREAMING PROCESSING

Unbounded packets, run continuously

- Processes an **unbounded**, continuous sequence of small packets
- Runs **continuously**; metrics update incrementally
- Requires **low compute**, low-latency networking
- Real-time analytics on a series of events
- Example: analyze immediately for a recommendation

05 AWS DATA SERVICES BY PIPELINE STAGE

one service per job — exam gold

STAGE	KEY AWS SERVICES	WHAT THEY DO
Ingest	AppFlow, DataSync, Data Exchange	SaaS apps; file shares to/from AWS; third-party data
Stream	Firehose, Kinesis Data Streams, Flink, Kinesis Video Streams, MSK	Ingest and process real-time streams
Process	AWS Glue, Amazon EMR, EMR Serverless	Batch ETL, cataloging, parallel big-data processing
Store	S3 (lake), Redshift (warehouse), Aurora/RDS, DynamoDB	Lake, warehouse, OLTP relational/NoSQL
Govern	Lake Formation, Glue Data Catalog	Central management, IAM access, schema/metadata
Analyze	Athena, QuickSight, OpenSearch Service	SQL queries, BI dashboards, real-time log analytics

06 STREAMING INGESTION SERVICES COMPARED

complexity vs. control vs. retention

SERVICE	BEST WHEN	RETENTION / REPLAY
Data Firehose	Simplest; plug-and-play delivery to approved destinations (S3, Redshift, OpenSearch); no code	Undelivered data max 24 h; no store or replay of delivered data
Kinesis Data Streams	Need more control or a non-Firehose destination; low latency; requires code (KPL/KCL)	Up to 365 days; multiple consumers can replay

SERVICE	BEST WHEN	RETENTION / REPLAY
Amazon MSK	Open-source Kafka to cut licensing; already know Kafka; most setup/engineering	Longer, configurable retention

07 EXAM TRAPS

where the knowledge check gets you

- × "Data ingestion always transforms the data." — **Homogeneous** ingestion moves data *as-is* with **no transformation** (same format/engine); only **heterogeneous** ingestion (ETL/ELT) transforms.
- × "ETL and ELT are interchangeable." — **ETL** transforms *before* loading and suits **structured data bound for a data warehouse**; **ELT** loads raw *then* transforms and suits **unstructured data bound for a data lake**.
- × "The Glue Data Catalog stores your datasets." — It stores **only metadata and schema** (table definitions, locations, attributes); the actual data stays in S3, RDS, and other stores.
- × "Firehose can replay delivered data." — Firehose **does not store or replay** delivered data (undelivered kept max 24 h). For replay, use **Kinesis Data Streams** (up to 365 days).
- × "A data lake needs a schema before you load data." — A data lake is **schema-on-read**; store data as-is. The **data warehouse** is **schema-on-write**.
- × "Streaming needs more compute than batch." — Reversed: **batch requires high compute** (whole datasets, off-peak); **streaming needs low compute** but reliable, low-latency networking.
- × "QuickSight, Athena, or Redshift can single-handedly analyze *and* visualize real-time logs." — Only **Amazon OpenSearch Service** does both in one service (Athena has no visualization; QuickSight needs a separate ingestion service; Redshift is a warehouse, not real-time).
- × "CSV is fine for large analytics files." — Convert **CSV → Parquet**: columnar, compressed, splittable for parallel processing — faster analytics and lower storage cost/time. CSV is inefficient beyond about 15 GB.

08 SELF-QUIZ

answers are upside-down below

Q1 Name the five Vs of data characteristics.

Q2 Which two Vs together drive a pipeline's expected throughput and scaling requirements?

Q3 In homogeneous ingestion, is the data transformed? Why or why not?

Q4 In ELT, is data transformed before or after loading, and what storage target does it suit?

Q5 Which serverless AWS service automates ETL and includes a Data Catalog, DataBrew, and Data Quality?

Q6 What does an AWS Glue crawler do?

Q7 Which streaming service retains data up to 365 days and supports replay by multiple consumers?

Q8 Which Kinesis-family service is the simplest, delivers to S3/Redshift/OpenSearch via micro-batches, and needs no code?

Q9 Does schema-on-write describe a data lake or a data warehouse?

Q10 Which single AWS service can analyze and visualize real-time streaming log data?

ANSWER KEY (rotate the page)

Q1 value, veracity, volume, velocity, variety **Q2** volume and velocity **Q3** No — the source and destination match, so data moves as-is **Q4** after loading (load raw first); it suits a data lake with unstructured data **Q5** AWS Glue **Q6** runs on data stores, derives a schema, and populates the Glue Data Catalog **Q7** Amazon Kinesis Data Streams **Q8** Amazon Data Firehose **Q9** a data warehouse (the data lake is schema-on-read) **Q10** Amazon OpenSearch Service (with OpenSearch Dashboards)