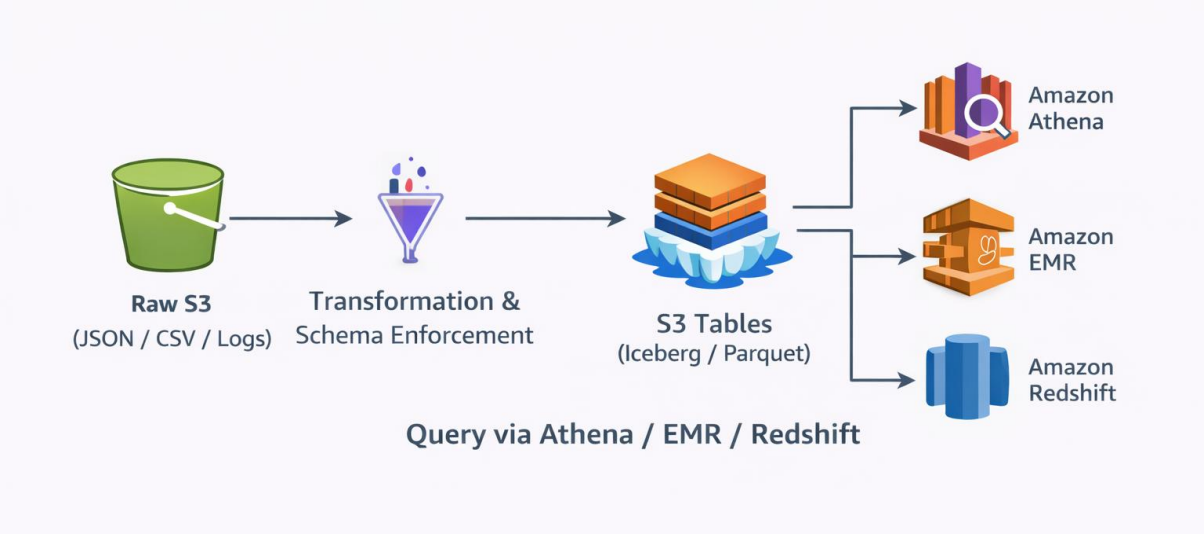
Amazon S3 Tables – Key Points

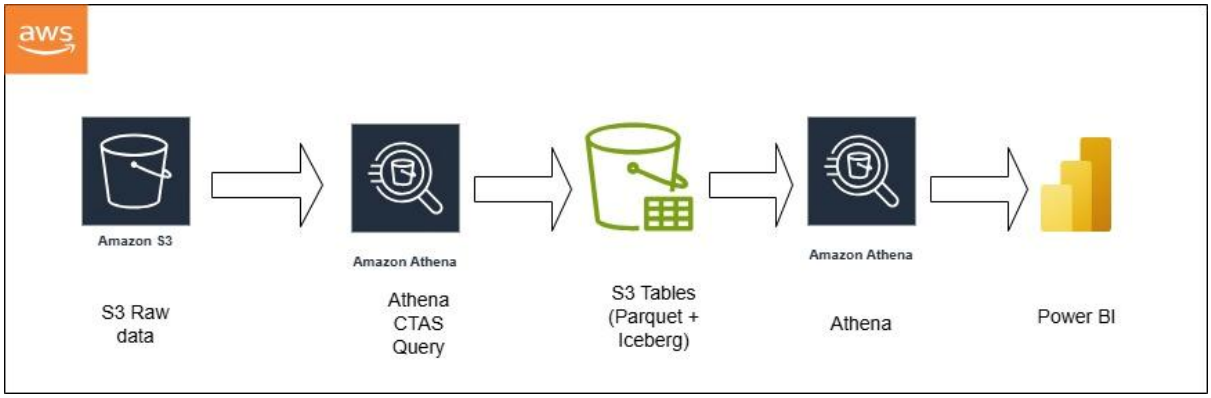
- **Amazon S3 Table Buckets** are purpose-built to store **Apache Iceberg tables** as first-class analytics objects.
- Unlike traditional S3 buckets, data is managed as **structured tables** with **automatic storage layout and metadata optimization**, eliminating manual tuning and maintenance.
- **Apache Iceberg** extends **Apache Parquet** by adding:
  - ACID-compliant transactions
  - Schema and partition evolution without data rewrites
  - Time-travel queries for historical analysis
  - Support for incremental updates
- Traditional Iceberg deployments required manual compaction, snapshot management, and cleanup of orphaned files.
- **Amazon S3 Tables automate table maintenance**, including compaction, snapshot lifecycle management, and metadata optimization.
- S3 Tables deliver **up to 3× faster query performance** and **up to 10× higher transaction throughput** compared to self-managed Iceberg tables.
- **Namespaces** provide structured organization and fine-grained access control at both namespace and table levels.
- Best suited for:
  - Streaming and near-real-time analytics
  - Change Data Capture (CDC) pipelines
  - Large-scale data lakes with frequent updates

Logical Flow



Data Injection Approaches

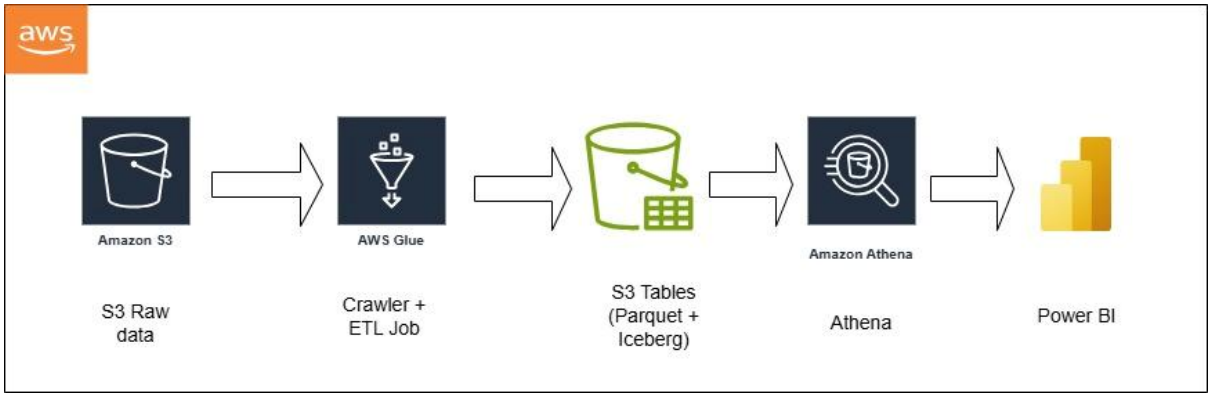
Approach 1: Athena CTAS (SQL-Only, Fastest Setup)



Process:

1. Raw data exists in S3 and is registered as an external table.
2. Athena reads raw data.
3. Athena writes transformed data as Parquet into an S3 Table.

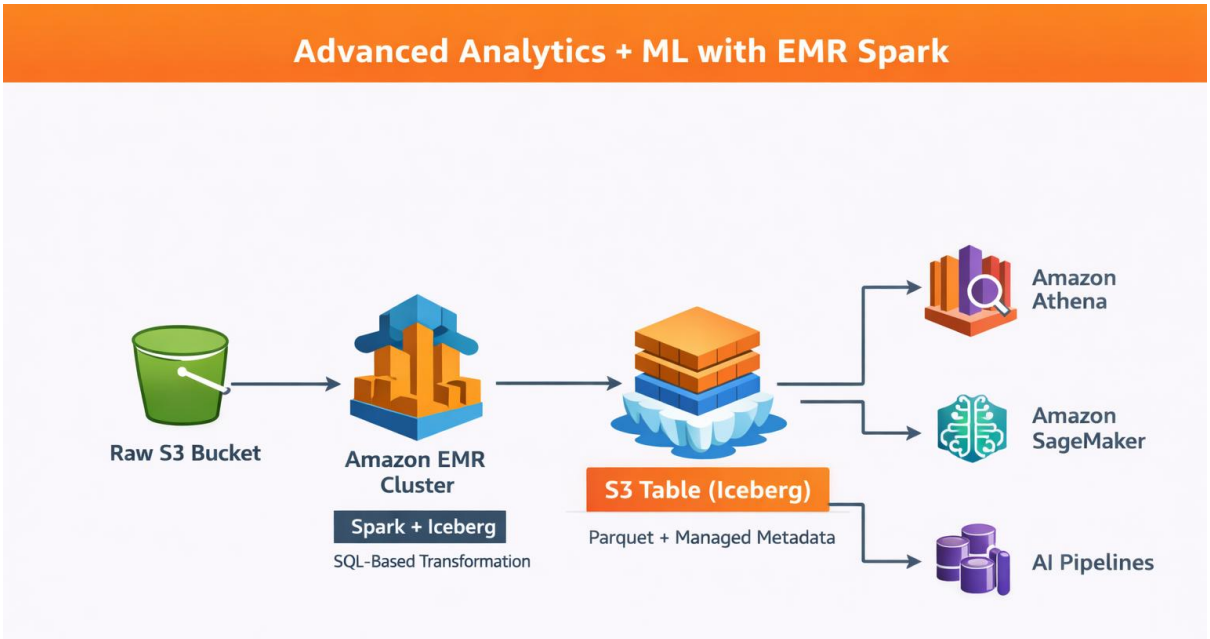
Approach 2: AWS Glue ETL (Enterprise Standard)



Process

1. Glue crawler catalogs raw S3 data.
2. Glue Spark job:
    - Reads raw S3
    - Cleans and normalizes data
    - Writes to S3 Tables (Iceberg)
3. S3 Tables automatically manage compaction and metadata.

Approach 3: EMR / Spark (High Control)



Process

- Spark job reads raw S3
- Writes Iceberg tables directly to S3 Tables

Cost Comparison: Data Injection Approaches (Raw S3 → S3 Tables)

Assumptions (for fair comparison):

- Data volume: **1 TB/day (30 TB/month)**
- Raw format: JSON / CSV
- Target format: Parquet (Iceberg)
- Region: Typical AWS pricing region
- Incremental daily loads (not full reloads)

Aspect	Athena CTAS	AWS Glue ETL	EMR / Spark
Cost Model	Pay per data scanned	Pay per DPU-hour	Pay per cluster runtime
Ingestion Cost (Monthly)	~$150	~$130–150	~$200–400
What You Pay For	Raw data scanned by SQL	Spark compute time	EC2 + EMR compute
Infrastructure	Fully serverless	Fully managed	User-managed cluster
Idle Cost	❌ None	❌ None	⚠ Possible if not stopped
Scalability Cost	Increases with data scan	Predictable	Increases with cluster size
Operational Overhead	Very low	Low	High
Transformation Complexity	Low–Medium	High	Very High
Cost Predictability	Medium	High	Low–Medium
Best Fit	POC / MVP	Production workloads	Advanced analytics / ML

Monthly Cost Breakdown (Illustrative)

Approach 1: Athena CTAS

- Athena scan cost: **$5 per TB**
- 1 TB/day × 30 days = 30 TB scanned
  ☞ **30 × $5 = ~$150/month**

✅ Lowest setup effort
❌ Cost grows with repeated raw scans

Approach 2: AWS Glue ETL

- Example job: 10 DPUs × 1 hour/day
- Cost: $0.44 per DPU-hour

10 × 1 × 30 × $0.44 ≈ $132/month

✅ Predictable cost
✅ No repeated raw scans
✅ Production standard

Approach 3: EMR / Spark

- Example: 6–10 node cluster
- 2 hours/day average runtime

≈ $200–400/month (depending on instance type)

⚠ Higher operational cost
⚠ Requires cluster lifecycle management