Data Warehousing and Beyond with AWS

Milo Kock
Amazon
What is the Data Warehouse?

- Centralized data storage
- Historical query and analysis
- Traditionally implemented with RDBMS
Typical DW Architecture

Transactions → ETL → Staging → ETL → DW → ETL → Front-End DB

- Logs
- Events
Case Study: Lightning Deals

- Promotional discount deals
- Time-bound
- Machine-learned
- Surface on Seller Central
DW-based Architecture

- Difficult to scale
- Inflexible
- Long-running queries block users
- Storage and compute tightly coupled

Sellers

Products

Offers

ETL

LD Data Mart

JDBC

ML-Scoring

Load

Seller-Facing DB
Rebuilding with AWS

- Collect
- Store
- Process
- Analyze
- Surface
<table>
<thead>
<tr>
<th>Collect &amp; Store</th>
<th>Process</th>
<th>Analyze</th>
<th>Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

?”
Storage

- S3
- SLA
  - 99.99% Availability
  - 99.999999999% Durability
- Integrated
- Scalable
- Cost-efficient
- Built-in lifecycle config
- Format-agnostic
- Store schema with data (Parquet, ORC)
- Apply at read-time
- Or, just CSV/JSON
AWS-Based Architecture

Collect & Store | Process | Analyze | Surface

S3
- Kinesis
- Streaming data platform
- Multiple applications can read same data in parallel
- Elastic – can scale to keep up with traffic spikes
- Up to a week retention period
- Triple-replicated
- Firehose - Kinesis "chunking" extension
- Configured via "hints"
  - Chunk size (1-128MB)
  - Chunk time interval (1-15min)
AWS-Based Architecture

Collect & Store | Process | Analyze | Surface

Kinesis | S3 (Raw)
- Elastic Map Reduce
- EC2 ETL compute grid
- Compute-agnostic
- Staging + aggregation
- Easy read/write to S3
AWS-Based Architecture

Collect & Store  Process  Analyze  Surface

Kinesis  S3 (Raw)  EMR  S3 (Processed)
- Elastic Map Reduce

- Spark

- Flink

- Hive

- Presto
AWS-Based Architecture

Collect & Store | Process | Analyze | Surface

Kinesis | S3 (Raw) | EMR | S3 (Processed) | EMR (SparkML) | S3 (Scored)
- DynamoDB
- NoSQL
- Low-latency
- Fully-managed
- Documents API
- Key-Value API
AWS-Based Architecture

Collect & Store
Kinesis → S3 (Raw)

Process
EMR → S3 (Processed)

Analyze
EMR (SparkML) → S3 (Scored)

Surface
DynamoDB (Front-End)
Orchestration

- AWS Glue
- Fully managed ETL
- Easy code generation to migrate data between sources
- Scheduling – event-based or scheduled
- Rich schema inference, tracking
- Automatically distributes ETL jobs on Apache Spark nodes
AWS-Based Architecture

Collect & Store
- Kinesis
- S3 (Raw)

Process
- EMR
- S3 (Processed)

Analyze
- EMR (SparkML)
- S3 (Scored)

Surface
- DynamoDB (Front-End)

AWS Glue
Thank you