DATA ENGINEERING | INTERMEDIATE | 3 DAYS

AWS Glue Deep Dive

Build a strong foundation in AWS Glue for production-grade ETL, from catalogs and crawlers to optimized Spark transformations. Learn how to build incremental pipelines, handle schema evolution, tune performance and cost, validate output quality, and troubleshoot real Glue job failures at scale.

Why AWS Glue ETL?

Faster ETL delivery

Build and run serverless data pipelines with minimal infrastructure work.

Improved metadata management

Glue Data Catalog enables consistent discovery and governance.

Reduced operational burden

Managed Spark reduces cluster setup, patching, and scaling concerns.

Cost-efficient processing

Pay for job execution time instead of always-on clusters.

Better integration

Seamlessly connect S3, Redshift, RDS, Athena, and modern lakehouse workflows.

What you'll be able to do

- Understand AWS Glue architecture and core components in detail.
- Build production-ready Glue ETL pipelines for batch data processing.
- Work confidently with Glue Data Catalog including databases, tables, crawlers, and schema updates.
- Use AWS Glue jobs effectively including Spark-based ETL, Glue Studio workflows (intro), and Python Shell jobs (overview).
- Transform data using Glue DynamicFrames and Spark DataFrames.
- Implement incremental processing patterns using partition-based loads and watermark strategy concepts.
- Handle schema drift and evolving datasets safely.
- Optimize Glue jobs for performance and cost using worker sizing, DPUs, partitioning, and pushdown predicates.
- Implement data quality and validation workflows inside Glue pipelines.
- Troubleshoot and debug Glue jobs using CloudWatch logs and practical techniques.

Who this is for

- Data Engineers
- Cloud Data Engineers (AWS)
- Analytics Engineers working with AWS data lake pipelines
- Data Platform Engineers
- DevOps/Cloud Engineers supporting Glue environments

Curriculum

01 AWS Glue Architecture and Core Components

- What AWS Glue is and where it fits in AWS data platforms
- Core components (Glue Data Catalog, Crawlers, Jobs, Triggers, Workflows)
- Execution model overview (Spark on AWS-managed infrastructure)
- IAM roles and permissions required for Glue pipelines
- **Hands-on:** Lab: Explore Glue console, identify core components, and map a Glue ETL flow end-to-end

02 Glue Data Catalog Deep Dive (Databases, Tables, Crawlers)

- Catalog structure (databases, tables, partitions)
- Crawlers configuration (data stores, classifiers, schedules)
- Schema inference behavior and common pitfalls
- Schema updates, versioning behavior, and safe update patterns
- **Hands-on:** Lab: Create a database + crawler, generate tables, and validate partition discovery

03 Building Production-Ready Glue ETL Pipelines (Batch Processing)

- ETL pipeline structure (raw → cleansed → curated)
- Job design patterns (modular transforms, reusable functions)
- Input and output formats (CSV, JSON, Parquet) and S3 layout strategy
- Job parameterization (paths, dates, environments)
- **Hands-on:** Lab: Build a batch Glue job to read raw data from S3, transform, and write curated Parquet

04 Glue Jobs Overview (Spark ETL, Glue Studio Intro, Python Shell Overview)

- Spark-based Glue jobs (scripts, bookmarks concept, job parameters)
- Glue Studio visual jobs overview (when to use, limitations)
- Triggers and scheduling patterns
- Python Shell jobs overview (lightweight orchestration and utilities)
- **Hands-on:** Lab: Create a Glue Spark job and run it with parameters; explore Glue Studio workflow UI

05 Transformations with DynamicFrames and Spark DataFrames

- DynamicFrames vs DataFrames (when to use each)
- Core DynamicFrame transforms (ApplyMapping, ResolveChoice, DropNullFields)
- Switching between DynamicFrame and DataFrame safely
- Common transformation patterns (join, dedup, aggregate)
- **Hands-on:** Lab: Implement cleansing transformations using DynamicFrames, then convert to DataFrame for advanced logic

06 Incremental Processing Patterns (Partitions and Watermarks)

- Full load vs incremental processing trade-offs
- Partition-based incremental loads (by date, by region, by source)
- Watermark strategy concepts (updated_at, ingestion timestamp)
- Handling late-arriving data concept
- **Hands-on:** Lab: Implement partition-based incremental loads and validate row counts across multiple runs

07 Handling Schema Drift and Evolving Datasets

- Why schema drift happens (new columns, type changes, nested structures)
- Crawler schema change behaviors and safe update settings
- ResolveChoice strategies and compatible type evolution patterns
- Backward/forward compatibility and data contract concepts
- **Hands-on:** Lab: Simulate schema changes and update Glue job logic to handle drift safely

08 Performance and Cost Optimization for Glue Jobs

- Understanding DPUs, workers, and job sizing
- Partitioning and file sizing to reduce shuffle and runtime
- Pushdown predicates and reading only necessary data
- Job tuning checklist (caching, broadcast joins concept, avoiding small files)
- **Hands-on:** Lab: Tune a slow job by adjusting workers, partitions, and pushdown predicates and compare runtimes

09 Data Quality and Validation in Glue Pipelines

- Data validation patterns (schema checks, null checks, duplicates, range checks)
- Fail-fast vs warn-only strategy
- Generating validation outputs (bad records, quality report)
- Publishing curated output only after validation passes
- **Hands-on:** Lab: Add a validation stage to a Glue job and generate a data quality report

10 Troubleshooting and Debugging Glue Jobs (CloudWatch + Practical Techniques)

- Reading CloudWatch logs and common error categories
- Debug techniques (sample runs, printing schemas, limiting data)
- Common failures (permissions, missing partitions, schema mismatch, memory issues)
- Operational best practices (retries, alarms concepts, runbooks)
- **Hands-on:** Lab: Fix a broken Glue pipeline scenario using CloudWatch logs and step-by-step debugging