Glue Slides

09:48 PM

AWS Glue

Table definitions and ETL

What is Glue?

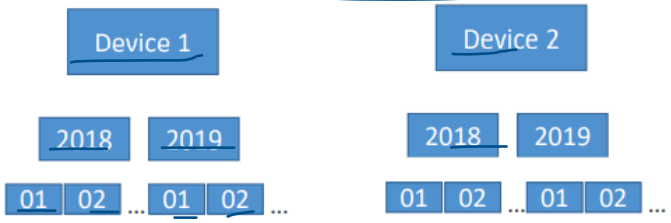
- Serverless discovery and definition of table definitions and schema
  - S3 “data lakes”
  - RDS
  - Redshift
  - DynamoDB
  - Most other SQL databases
- Custom ETL jobs
  - Trigger-driven, on a schedule, or on demand
  - Fully managed

Glue Crawler / Data Catalog

- Glue crawler scans data in S3, creates schema
- Can run periodically
- Populates the Glue Data Catalog
  - Stores only table definition
  - Original data stays in S3
- Once cataloged, you can treat your unstructured data like it’s structured
  - Redshift Spectrum
  - Athena
  - EMR
  - Quicksight

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 yyyy/mm/dd/device
- Do you query primarily by device?
  - If so, organize your buckets as device/yyyy/mm/dd



Glue + Hive

- Hive lets you run SQL-like queries from EMR
- The Glue Data Catalog can serve as a Hive "metastore"
- You can also import a Hive metastore into Glue

Glue ETL

- Automatic code generation
- Scala or Python
- Encryption
  - Server-side (at rest)
  - SSL (in transit)
- Can be event-driven
- Can provision additional "DPU's" (data processing units) to increase performance of underlying Spark jobs
  - Enabling job metrics can help you understand the maximum capacity in DPU's you need
- Errors reported to CloudWatch

- Could tie into SNS for notification

Glue ETL

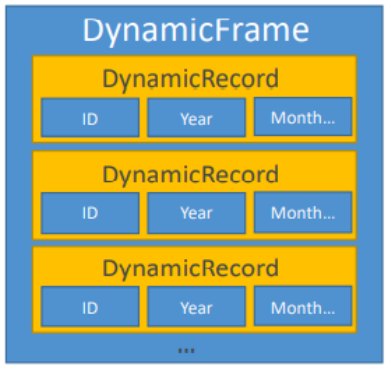
- Transform data, Clean Data, Enrich Data (before doing analysis)
  - Generate ETL code in Python or Scala, you can modify the code
  - Can provide your own Spark or PySpark scripts
  - Target can be S3, JDBC (RDS, Redshift), or in Glue Data Catalog
- Fully managed, cost effective, pay only for the resources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

- A DynamicFrame is a collection of DynamicRecords
  - DynamicRecords are self-describing, have a schema
  - Very much like a Spark DataFrame, but with more ETL stuff
  - Scala and Python APIs

```
val pushdownEvents = glueContext.getCatalogSource(
database = "githubarchive_month", tableName = "data")

val projectedEvents = pushdownEvents.applyMapping(Seq(
("id", "string", "id", "long"), ("type", "string", "type",
"string"), ("actor.login", "string", "actor", "string"),
("repo.name", "string", "repo", "string"),
("payload.action", "string", "action", "string"),
("org.login", "string", "org", "string"), ("year",
"string", "year", "int"), ("month", "string", "month",
"int"), ("day", "string", "day", "int") ))
```



Glue ETL - Transformations

- 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)
  - Can convert between Spark DataFrame and Glue DynamicFrame

Glue ETL: Modifying the Data Catalog

- ETL scripts can update your schema and partitions if necessary
- Adding new partitions
  - Re-run the crawler, or
  - Have the script use enableUpdateCatalog and partitionKeys options
- Updating table schema
  - Re-run the crawler, or
  - Use enableUpdateCatalog / updateBehavior from script
- Creating new tables
  - enableUpdateCatalog / updateBehavior with setCatalogInfo
- Restrictions
  - S3 only
  - Json, csv, avro, parquet only
  - Parquet requires special code
  - Nested schemas are not supported

AWS Glue Development Endpoints

- Develop ETL scripts using a notebook
  - Then create an ETL job that runs your script (using Spark and Glue)
- Endpoint is in a VPC controlled by security groups, connect via:
  - Apache Zeppelin on your local machine
  - Zeppelin notebook server on EC2 (via Glue console)
  - SageMaker notebook
  - Terminal window
  - PyCharm professional edition
  - Use Elastic IP's to access a private endpoint address

Running Glue jobs

- Time-based schedules (cron style)
- Job bookmarks
  - Persists state from the job run
  - Prevents reprocessing of old data
  - Allows you to process new data only when re-running on a schedule
  - Works with S3 sources in a variety of formats
  - Works with relational databases via JDBC (if PK's are in sequential order)

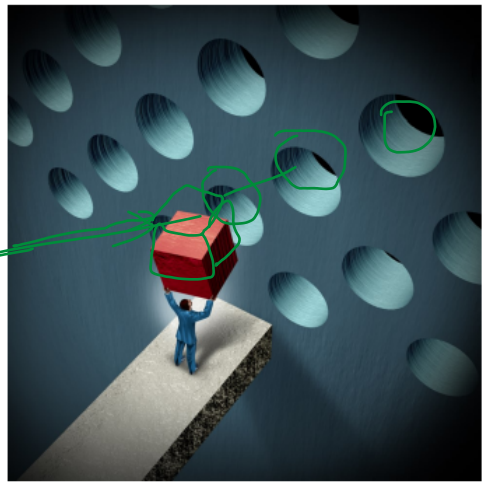
- Only handles new rows, not updated rows
- CloudWatch Events
  - Fire off a Lambda function or SNS notification when ETL succeeds or fails
  - Invoke EC2 run, send event to Kinesis, activate a Step Function

Glue cost model

- Billed by the minute for crawler and ETL jobs
- First million objects stored and accesses are free for the Glue Data Catalog
- Development endpoints for developing ETL code charged by the minute

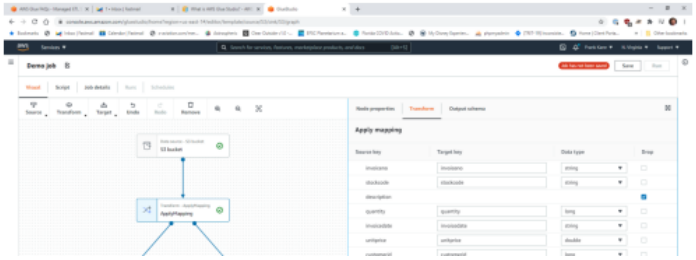
Glue Anti-patterns

- Multiple ETL engines
  - Glue ETL is based on Spark
  - If you want to use other engines (Hive, Pig, etc) Data Pipeline EMR would be a better fit.



AWS Glue Studio

- Visual interface for ETL workflows
- Visual job editor
  - Create DAG's for complex workflows
  - Sources include S3, Kinesis, Kafka, JDBC
  - Transform / sample / join data

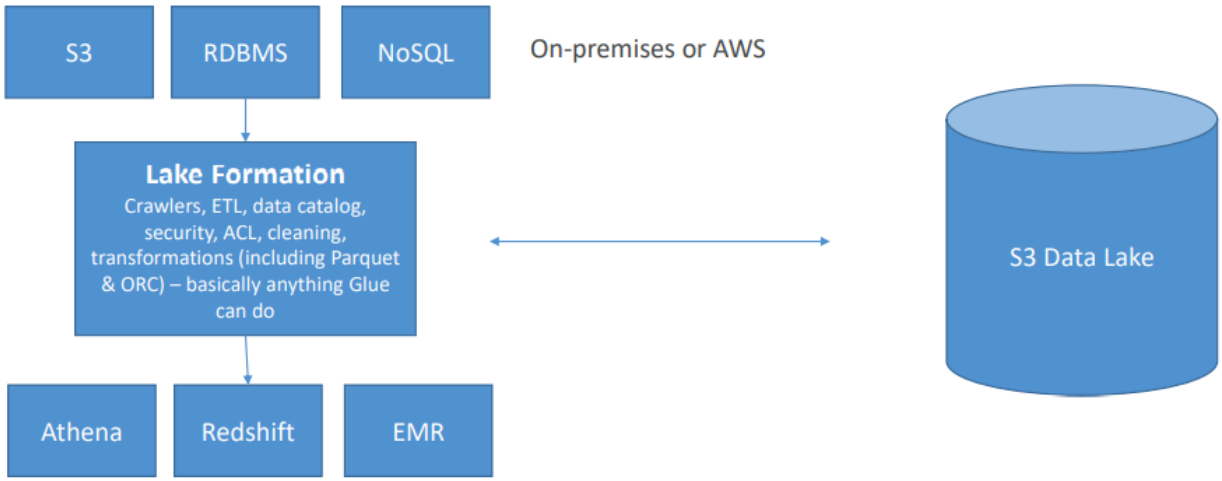


- Transform / Sample / Join data
  - Target to S3 or Glue Data Catalog
  - Support partitioning
- Visual job dashboard
  - Overviews, status, run times

AWS Glue DataBrew

- A visual data preparation tool
  - UI for pre-processing large data sets
  - Input from S3, data warehouse, or database
  - Output to S3
- Over 250 ready-made transformations
- You create “recipes” of transformations that can be saved as jobs within a larger project
- Security
  - Can integrate with KMS (with customer master keys only)
  - SSL in transit
  - IAM can restrict who can do what
  - CloudWatch & CloudTrail

AWS Lake Formation



AWS Lake Formation: Pricing

- No cost for Lake Formation itself
- But underlying services incur charges
  - Glue

- S3
- EMR
- Athena
- Redshift