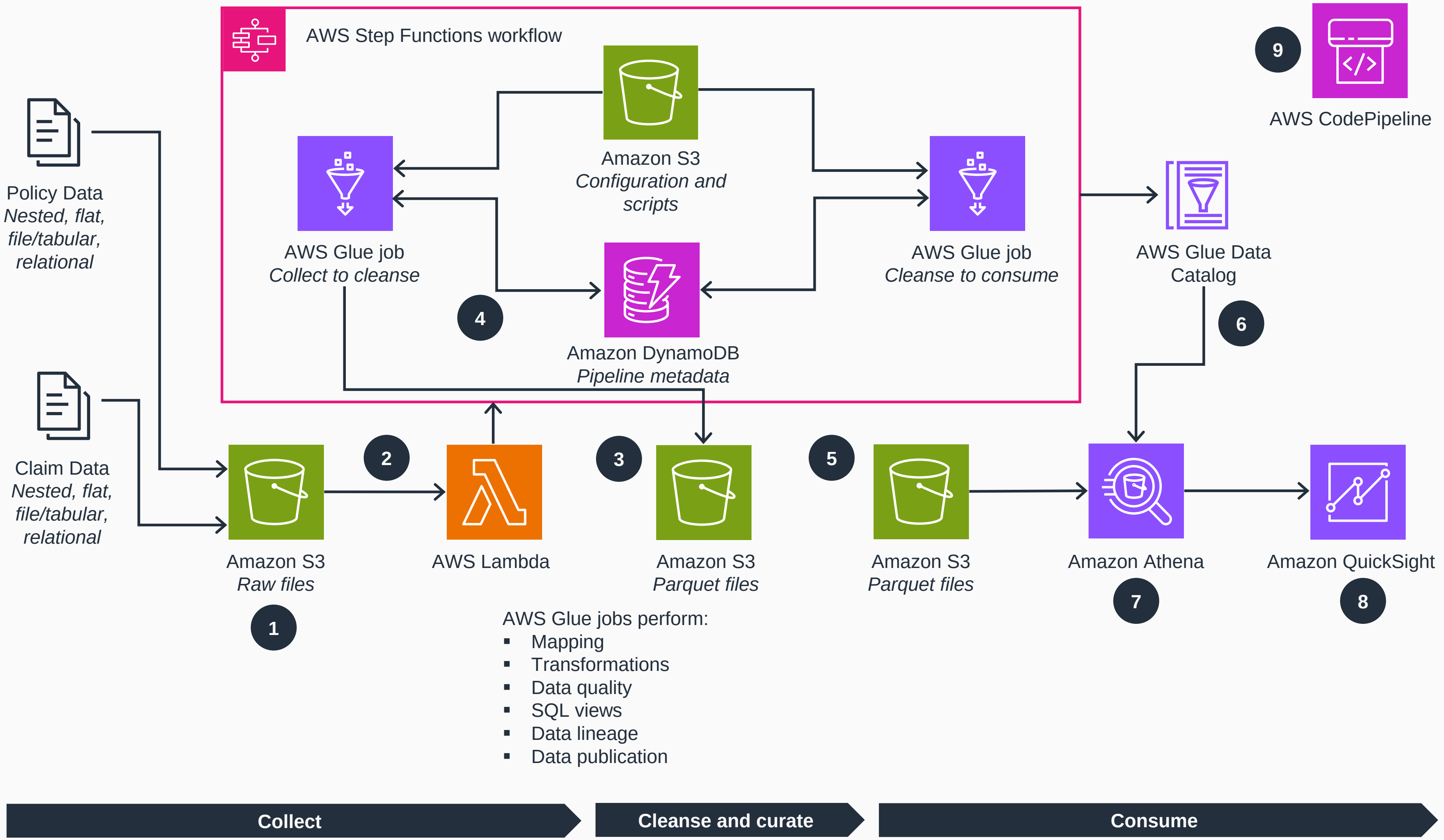
Guidance for Modern Insurance Data Lakes on AWS

This architecture diagram shows how to collect, cleanse, and consume insurance data with ETL processes and data storage.



AWS Glue jobs perform:

- Mapping
- Transformations
- Data quality
- SQL views
- Data lineage
- Data publication

1 Business analysts define the data pipeline operations using low-code configuration files stored in an **Amazon Simple Storage Service (Amazon S3)** bucket. Data sources upload source data files, such as policies and claims, to the *Collect* **S3** bucket.

2 An *ObjectCreated* event invokes an **AWS Lambda** function that reads metadata from the incoming source data, logs all actions, and starts the **AWS Step Functions** workflow.

3 **Step Functions** calls **AWS Glue** jobs that map the data to your predefined data dictionary. These jobs then perform the transformations and data quality checks for both the *Cleanse* and *Consume* layers.

4 **Amazon DynamoDB** contains lookup values used by the lookup and multi-lookup transforms; extract, transform, and load (ETL) metadata such as job audit logs, data lineage output logs, and data quality results are written here.

5 **AWS Glue** jobs store cleansed and curated data in **Amazon S3** as compressed, partitioned Apache Parquet files. **AWS Glue** jobs also create and update the **AWS Glue Data Catalog** databases and tables.

6 **AWS Glue** jobs store source data file validation failures in an **Amazon S3** *Quarantine* folder and **Data Catalog** table which can populate an exception queue dashboard that allows a human to review and take appropriate action.

7 **Amazon Athena** runs SQL queries using the **Data Catalog** databases and tables.

8 **Amazon QuickSight** dashboards and reports pull data from the data lake on a near real-time or scheduled basis.

9 **AWS CodePipeline** manages the full DevSecOps cycle for the infrastructure, application, and pipeline configuration.

Reviewed for technical accuracy June 27, 2024

AWS Reference Architecture