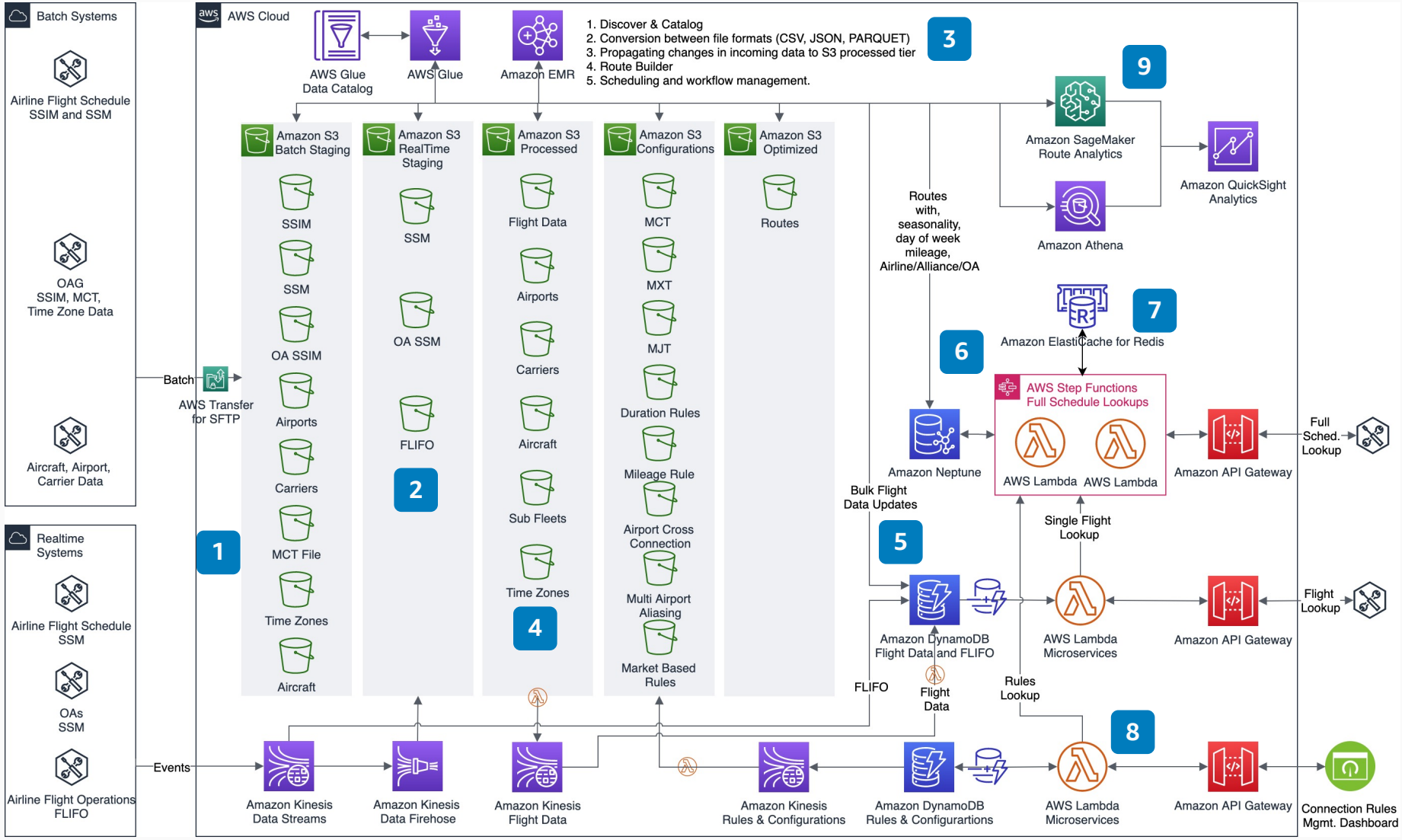
Airline Schedule Engine

Reduce the total cost of ownership using purpose built databases, serverless architecture, and data lakes to create a scalable, configurable, and fault tolerant tier-0 system for serving fast flight lookups and schedule lookups for an airline.



1. All batch inputs, like SSIM, are loaded into a batch staging bucket in **Amazon S3**.
2. All real-time data feeds, like SSM and FLIFO, are loaded into a real-time staging bucket in **Amazon S3**.
3. **AWS Glue** and **Amazon EMR** based processes are used to discover, catalog, process inputs, and create the processed data in S3. These processes combine batch and real-time data to create the processed flight data.
4. Flight data is created by converting the schedule files into individual flights and then loaded into **Amazon DynamoDB** for serving direct flight lookup.
5. Flight lookups are enabled through **AWS Lambda** and **Amazon DynamoDB**, with in-memory caching provided by **Amazon DynamoDB Accelerator (DAX)**. Additionally, FLIFO events can be ingested and served along with flight data.
6. Routes with mileage, day of week, seasonality, and airline type are processed and loaded into **Amazon Neptune** Graph database to provide route lookups by origin and destination.
7. Duration and mileage rules are applied, flights are retrieved for each route and combined with connection rules to create the full schedule. **Amazon ElastiCache for Redis** is used to improve performance.
8. Connection rules are maintained in **Amazon DynamoDB** for fast retrieval for usage in connection building. These rules can be managed with a connection rules management dashboard.
9. **Amazon SageMaker** can be used to improve route building and schedule lookup performance.

Reviewed for technical accuracy May 3, 2021

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture