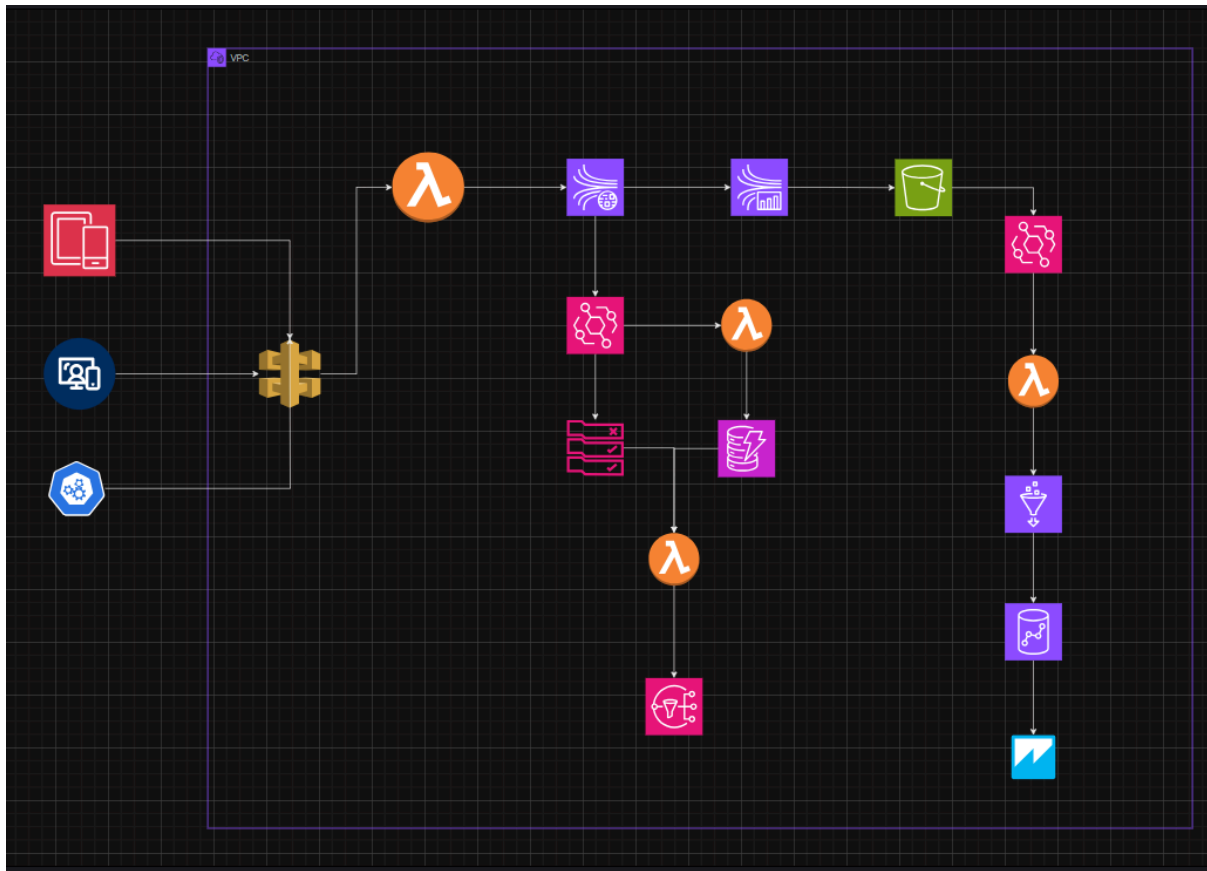


Designing a Real-time Data Ingestion Pipeline for a Flight Booking Application using AWS Services



User Interaction:

Users interact with the flight booking application through various platforms such as web, mobile, or API.

AWS API Gateway:

Serves as the entry point for user requests. Handles authentication, authorization, and routing of requests to the appropriate backend services.

AWS Lambda Functions:

Responsible for processing user requests and generating events for data ingestion.

Receives requests from the API Gateway and triggers corresponding Lambda functions based on the request type (e.g., new booking, cancellation, search query).

Amazon Kinesis Data Streams:

Streams are used to ingest real-time data generated by Lambda functions.

Lambda functions push events (e.g., new bookings, cancellations, search queries) to Kinesis Data Streams.

Amazon Kinesis Data Analytics:

Performs real-time analytics on the streaming data received from Kinesis Data Streams.

Analyzes, processes, and responds to incoming data streams in real-time.

Generates insights, such as real-time analytics and alerts on high-demand routes.

DynamoDB:

Store booking and cancellation events to a kinesis data stream and maintains current manifest(Like how many seats are filled and how many seats are available)

Amazon S3:

Stores the raw and processed data for long-term storage and further analysis.

Data ingested from Kinesis Data Streams is stored in S3 buckets.

Provides durability, scalability, and low-cost storage for large volumes of data.

Amazon Redshift:

Optionally used for data warehousing and analytics.

Allows querying and analysing large datasets with high performance and scalability.

Data from S3 can be loaded into Redshift for further analysis and reporting.

Amazon CloudWatch:

Monitors the health and performance of the entire pipeline.

Provides real-time monitoring and alerts for operational issues.

Alerts can be triggered based on predefined conditions or events detected in the data stream.

Amazon SNS:

Optionally used for sending alerts or notifications based on predefined conditions or events detected in the data stream.

Sends notifications to designated endpoints (e.g., email, SMS) based on specified triggers.

Amazon QuickSight:

Analysts and stakeholders can access the QuickSight dashboard to gain insights and perform analytics on the data stored in Redshift.

Note:

By deploying the entire architecture in a multi-zone setup, 'FlyHigh Airways' can ensure that their real-time data ingestion pipeline remains highly available, fault-tolerant, and resilient to AZ failures, providing a seamless experience for users of their flight booking application.

Assumptions:

Data volume: Assuming a high volume of real-time data generated by user interactions with the flight booking application.

Rate of data ingestion: Expecting a continuous stream of user bookings, cancellations, and search queries.

Scalability requirements: Anticipating the need for the system to handle increased traffic and data volume as 'FlyHigh Airways' expands its operations.

AWS-only architecture: Utilizing only AWS services for building the real-time data ingestion pipeline.

Scalability Justification:

Auto-scaling: AWS services like Kinesis Data Streams, Lambda, and S3 automatically scale to accommodate increasing data volume and traffic.

Built-in redundancy: AWS services are designed with built-in redundancy to ensure high availability and fault tolerance.

Managed services: Leveraging managed services minimizes operational overhead and allows for focusing on the application's business logic rather than infrastructure management.

Elasticity: AWS services can dynamically scale up or down based on demand, ensuring optimal performance and cost efficiency.

Horizontal scaling: Components like Kinesis Data Streams and Lambda support horizontal scaling, allowing for distributing the workload across multiple instances to handle peak loads efficiently.

Minimal latency: Utilizing services like Kinesis Data Streams and Lambda for real-time data processing ensures minimal latency, enabling timely insights and responses to user interactions.