AWS Step Functions for Workflow Automation

A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

in the Department of Computer Science & Engineering

Cloud Based AI/ML Speciality (22SDCS07A)

Submitted by
2210030123 : K. Poojitha

Under the guidance of

Ms. P. Sree Lakshmi

Department of Computer Science and Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075

March - 2025.

INTRODUCTION

This project implements a simple Amazon-like booking system that automates purchasing and refunding workflows using AWS Step Functions. Step Functions provide a serverless orchestration service that manages stateful executions of multiple AWS services, ensuring seamless automation of order processing and refunds. By integrating AWS services such as Lambda, DynamoDB, and S3, the system efficiently handles user transactions while ensuring reliability and scalability. The system ensures error handling through built-in retries and state transitions, minimizing failed transactions. The system also ensures seamless integration between purchasing and refunding workflows, enabling smooth transitions and reducing processing delays.

AWS Services

1. **AWS Step Functions:** Orchestrates the purchasing and refunding workflow with state management.
2. **Amazon Lambda:** Executes business logic for purchase handling, refund processing, and order finalization.
3. **Amazon S3:** Stores order-related data and logs for audit purposes.
4. **Amazon DynamoDB:** Manages order records, transaction details, and refund statuses.
5. **Amazon CloudWatch:** Monitors workflow execution, logs errors, and ensures system performance.

Project Purpose and Expected Outcome

The purpose of this project is to create an **automated booking system** that streamlines purchasing and refunding processes. By leveraging AWS Step Functions, businesses can minimize manual intervention, ensuring efficient transaction processing and improved user experience. Also, the system enhances data consistency by maintaining a structured workflow for order management. The expected outcomes include:

- **Automated Order Processing:** Step Functions handle purchasing and refunding workflows seamlessly.
- **Improved Scalability:** The serverless architecture ensures efficient handling of multiple transactions.
- **Real-time Monitoring:** CloudWatch provides insights into system performance and failure handling.
- **Secure and Reliable Transactions:** DynamoDB ensures data consistency, while Step Functions manage workflow reliability.

Methodology

Architecture and Workflow

The architecture of the AWS Step Functions-based booking system consists of multiple AWS services working together to handle purchasing and refunding. The workflow includes:

1. **Choice State (Purchase or Refund):** Users initiate either a purchase request or a refund request.
2. **Purchase Handler:** If the user selects purchase, Step Functions invoke a Lambda function that processes the order, updates the database, and logs the transaction.
3. **Refund Handler:** If the user requests a refund, Step Functions call a different Lambda function that verifies the request, processes the refund, and updates the order status.
4. **Result Handler:** After either purchase or refund execution, a final state processes the outcome and logs the result.
5. **End State:** The workflow completes after successfully processing the request.

AWS Services Interaction

AWS Step Functions interact with various AWS services to automate purchasing and refunding:

- **Amazon S3:** Stores logs of purchase and refund transactions, ensuring durability and secure access to historical records for auditing and reporting purposes.
- **AWS Lambda:** Handles purchase processing, refund verification, and order finalization by executing backend logic in a serverless environment, enabling rapid scaling and cost efficiency.
- **Amazon DynamoDB:** Stores transaction records, order details, and refund statuses with low-latency access, ensuring real-time updates and consistency across the system.
- **Amazon CloudWatch:** Monitors the workflow execution, logs system performance, and triggers alerts if failures occur.

Justification for AWS Service Selection

AWS Step Functions were chosen for their ability to efficiently orchestrate purchasing and refunding workflows while managing state transitions and error handling. The integration with Lambda ensures a serverless execution model**,** reducing infrastructure costs. DynamoDB provides fast**,** scalable transaction management, while CloudWatch ensures real-time monitoring and troubleshooting. Together, these services create a robust and efficient cloud-based booking system. Step Functions also provide built-in retry mechanisms, ensuring system reliability even in cases of transient failures. By automating the entire workflow, businesses can focus on enhancing customer experience rather than handling manual order processing.

Implementation

AWS Infrastructure Setup

To set up the Cloud-Based Booking System, the following AWS services are configured:

- **Create an Amazon S3 Bucket:** Stores order-related data and logs transaction details securely.
- **Deploy Amazon DynamoDB:** A NoSQL database is set up to manage order records, purchase transactions, and refund statuses with real-time updates.
- **Develop AWS Lambda Functions:** Lambda functions handle order processing, refund verification, and workflow execution, ensuring business logic is executed efficiently.
- **Configure AWS Step Functions:** Defines the workflow logic for order handling, including branching for purchase and refund requests.

Security Policies, IAM Roles, and Access Controls

- **IAM Roles & Policies:** Access is restricted based on roles, ensuring only authorized users and services can modify transactions.
- **S3 Bucket Permissions:** Order logs and transaction records are secured with restricted access to prevent unauthorized modifications.
- **Lambda Security:** IAM policies enforce strict execution permissions to limit function access and prevent unintended modifications.
- **DynamoDB Encryption:** Data is encrypted at rest and in transit to ensure compliance with security best practices.
- **Step Functions Permissions:** Access to Step Functions is restricted, ensuring only permitted workflows can trigger execution.

These security measures ensure data protection, controlled access, and system resilience against unauthorized modifications.

Automation

Automation in the **Cloud-Based Booking System** ensures efficient, real-time operations without manual intervention.

- **Automatic Lambda Triggering:** AWS Lambda automates purchase and refund processing by dynamically executing backend logic based on user actions. This eliminates the need for manual intervention, reducing processing delays and improving transaction efficiency.
- **Workflow Orchestration:** AWS Step Functions automate state transitions by defining a structured workflow, ensuring seamless coordination between purchasing and refunding

operations. This enhances system reliability by enforcing predefined execution paths and error handling mechanisms.

- **CloudWatch Monitoring:** Amazon CloudWatch continuously monitors workflow execution, capturing key performance metrics, error logs, and execution times. It provides real-time insights into system health, allowing proactive issue resolution and performance optimization.
- **Data Backup Strategy:** DynamoDB backups and S3 lifecycle policies automate data retention, ensuring critical order and refund records are securely stored. Versioning and scheduled backups enable rapid recovery from failures or accidental deletions, maintaining data integrity.