AWS Use Cases – Ransford Danso

Customer order software use case

Business Overview:

The Goal of this system is setting up an operating system that would be Fast and efficient for ordering transactions between customers employees and Managers.

Functional and Staff goals:

Service Staff

* Create a new order
* Close out an order

Kitchen Staff

* View all open orders
* Complete/fulfill an order

Managers

* View all orders
* Close out service
* Generate sales report’s
* Archive orders to McDonald’s regional headquarters
* Automatic delayed order detection
* View a specific sales report

Technical Architecture:

* AWS auto scaling can be used for scaling of the system as it can adjust its capacity to maintain the steadiness of the system.
* Read to write ratio of our data base should be 50/50 as our data will be mostly be receiving orders and responding to orders.
* Files and Data that would be stored in a S3 Glacier bucket system since it is designed for data archiving and easy to retrieve, PCI components would be stored in the system since we would be collecting a lot of customer data such as Debit/Credit cards and order’s that are archived for any future reference.

Additional Technical Architecture

* The storage data base will implement a main database from which all functions read/write to/from.
* IAM Groups will be implemented for all staff and employee’s with special permissions of the system.
* The operating system will also use AWS IAM for multi factor authentication
* AWS Cloud Trail will be used for event logs and activity logs.
* AWS Athena will be a quick and easy way of looking at analytics of your system because it is mainly serverless and an easy way to access our analytics.
* In case of emergency of any kind the system will use AWS backup for any type of data loss that can occur to the system.
* AWS CloudEndure will be used for any type of Disaster Recovery of our system.

Deployment Process

This CDK project will deploy a total of five (5) stacks into your AWS account using CloudFormation:

* Existing Resources - provisions an Amazon S3 bucket into your account that simulates/demonstrates using the project with existing infrastructure.
* SharedStack - provisions resources that are used across the following three stacks, including a central DynamoDB table (with configuration) and a Lambda layer for common database access functions (i.e. using a table scan to get all orders, archive the table, etc.).
* ServiceStaffStack - provisions resources that are used by service staff to access the system. Provides functions that allow users to create a new order and close out an order.
* KitchenStaffStack - provisions resources that are used by kitchen staff to access the system. Provides functions that allow users to view all open orders and mark a specific order as filled by the kitchen.
* ManagerStack - provisions resources that are used by one or more managers to access the system. Provides functions that allow users to view all orders (regardless of status), close-out the service for a day, and retrieve a specific report saved from a previous day's close-out.