

Reduce Food Wastage by the Unsold But Not Unwanted



Save & Save: Jaivignesh, Marcus, Niranjana, Yet Ying
CS5224 Cloud Computing
AY2021/22 Semester 2
Department of Computer Science
National University of Singapore



Motivation

- Over 1 billion tonnes of food waste is produced every year, even as more than 800 million people worldwide go to bed hungry every night.
- Supermarkets have been singled out as one of the major culprits behind food waste globally. In 2019, a local Supermarket, Fairprice, reported to have generated 2940 tonnes of food waste.
- Multiple solutions have been proposed over the years - donations to needy communities, discounted prices, etc. - however, there has not been a clear winner across the various supermarkets.

Objective

- Save & Save is a potential solution to this problem - a web platform for the various supermarket vendors to list and sell items that are expiring/to be cleared, at a discounted price
- The key objectives of Save & Save is to reduce edible food wastage. We will be targeting two major parties – the consumer and the seller.
- Promote and entice the unwanted food to potential consumers using Save & Save
- Provide sellers with analysis on items such as popularity, best period of time to sell certain products etc, to strategically improve sales performance

Workflow



Stores

- A store registers with the platform.
- Store uploads csv files of their products in a predefined format. CSV file contains necessary information about the product listings - name, category, new cost per unit, original cost per unit, offer expiry date and image links (optional).
- Store receives purchase order from Save & Save, and prepares products to be collected by the customers
- Store updates purchase order ‘status’ attribute after preparations are done / customer has collected them.
- Store reviews data insights at the end of the month - popular products, strategic timings etc.

Users



- A customer registers with the platform.
- Customer browses for products listed on Save & Save.
- Customer adds products into his/her cart and purchases them.
- Each purchase order will contain a ‘status’ attribute that will indicate if their items are ready for collection.
- Customer makes his/her way to the store to collect products.

Implementation

Overall SaaS Design

The services that we used were primarily from AWS. Since our product is a web application, this required a standard set of components - a **Backend system**, a **Frontend system**, a **Database system** and a **Storage system**.

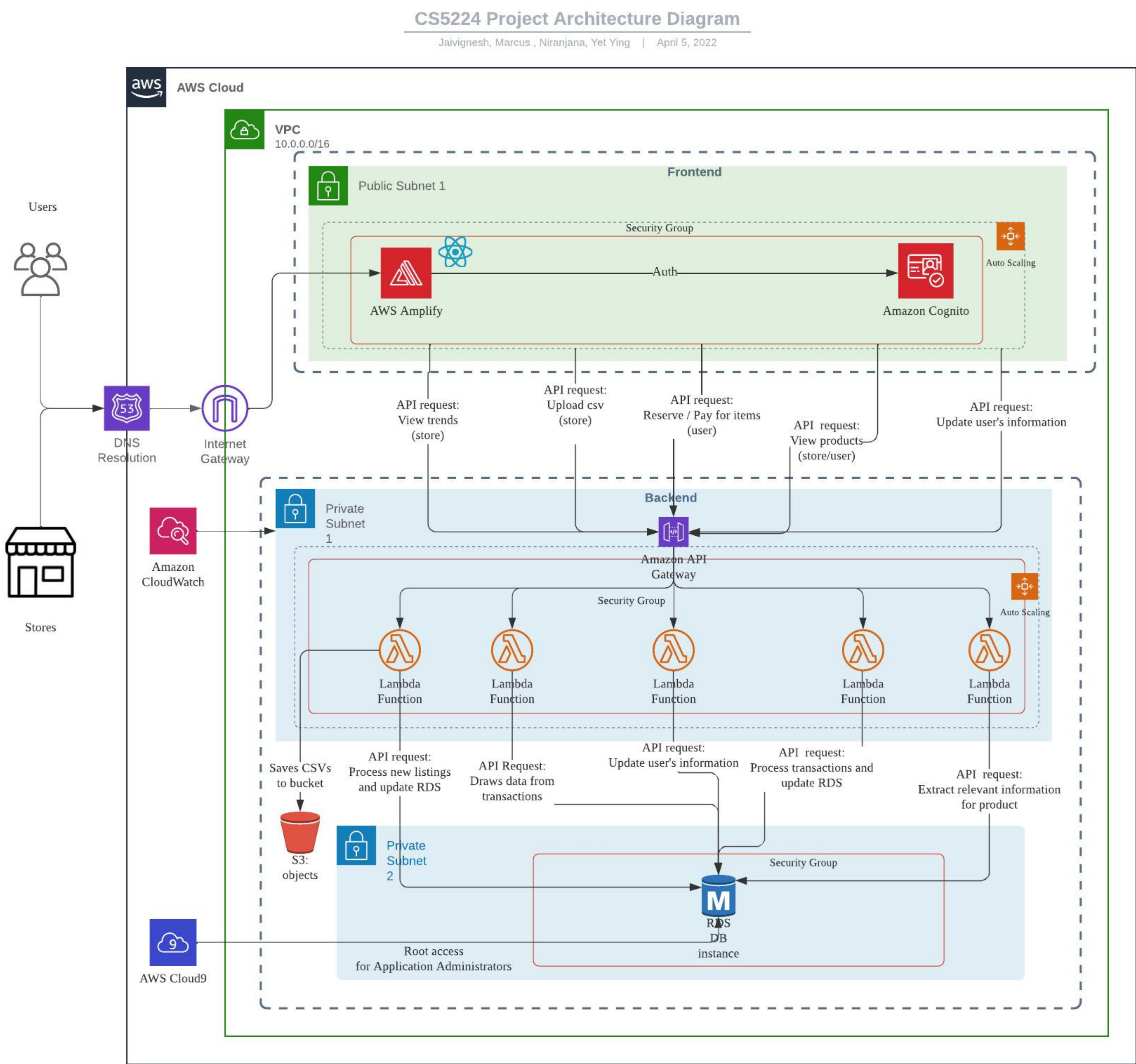
Components	SaaS Services
Frontend System	Amplify (build-in to configure CloudFormation, CloudFront, S3, AWS auth)
Backend System	AWS Lambdas
Database System	Amazon RDS (PostgresQL)
Storage (for CSVs)	S3
Monitoring	AWS CloudWatch

Components of Web Application

Lambda Functions as the Backbone of the Application

Lambda	Functionality
customer	Handles create, read, update and delete of platform user's (store/customer) information.
csv-handler	Handles creation of product listings via CSV files.
product	Handles create, read, update and delete of a single product listing.
products-query	Handles filtering of products based on keywords and offers expiry date.
purchase	Handles create, read of transactions between the platform and users.
purchase-product	Handles read of product listings and their quantities that were part of a transaction.
data-analytics-query	Handles queries such as 'popular listings', 'popular time to list' etc

Lambda functions and their responsibilities



Revenue Model

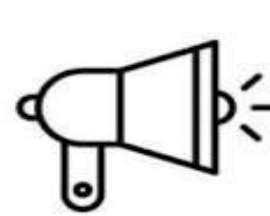


Free for buyers



Freemium Subscription for Supermarkets

Free mode
Monthly revenue < \$1000
Subscription mode
Monthly revenue >= \$1000



Advertisements

Paid Partnerships with Brands

Cost Breakdown

