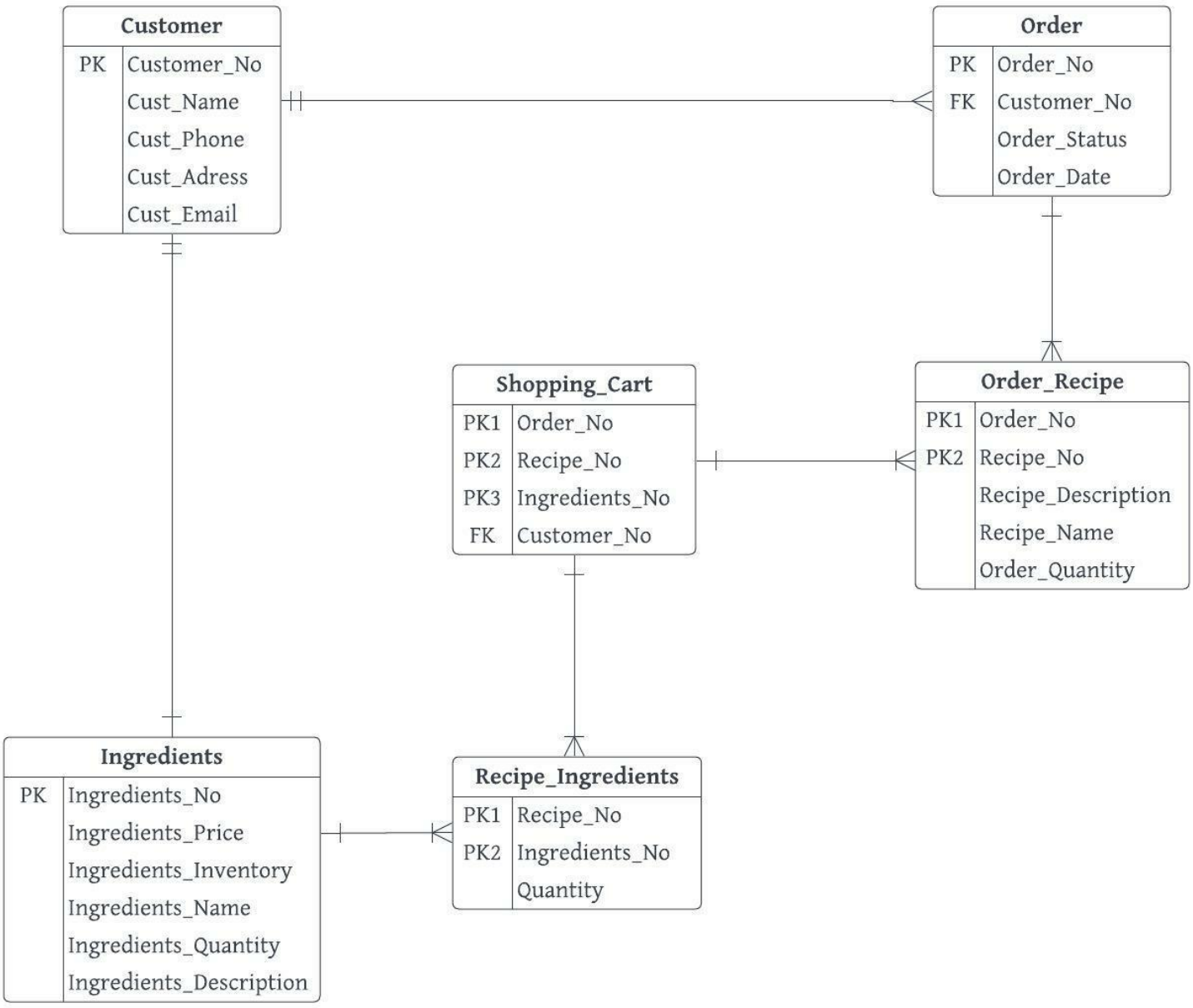
D4: Backend Design

4.1. Data Model - Entity-Relationship Diagram



Customer:

Customer_No	Cust_Name	Cust_Phone	Cust_Adress	Cust_Email

Order:

Order_No	Customer_No	Order_Status	Order_Date

Order_Recipe:

Order_No	Recipe_No	Recipe_Description	Recipe_Name	Order_Quantity

Shopping Cart:

Order_No	Recipe_No	Ingredients_No	Customer_No

Recipe_Ingredients:

Recipe_No	Ingredients_No	Quantity

Ingredients:

Ingredients_No	Ingredients_Price	Ingredients_Inventory	Ingredients_ Name	Ingredients _Quantity	Ingredients_ Description

Data Description :

Customer_No: Numeric Value, Up to 10 Characters; Cannot be NULL

Cust_Name: Alphanumeric Characters; Up to 30 Characters; Can be NULL

Cust_Phone: Numeric Value, Max of 10 Character length; Can be NULL

Cust_Address: Alphanumeric Characters; Up to 50 Characters; Can be NULL

Cust_Email: Alphanumeric Characters; Up to 80 Characters; Can be NULL

Order_No: Numeric Value, Up to 10 Characters; Cannot be NULL

Order_Status: Alphanumeric Characters; Up to 20 characters; Can be NULL

Order_Date: Date Value; Up to 9 Characters; Can be NULL

Order_Quantity: Numeric Value; Up to 10 Characters; Can be NULL

Recipe_No: Numeric Value; Up to 10 Characters; Cannot be NULL

Recipe_Description: Alphanumeric Characters; Up to 100 Characters; Can be NULL

Recipe_Name: Alphanumeric Characters; Up to 50 Characters; Can be NULL

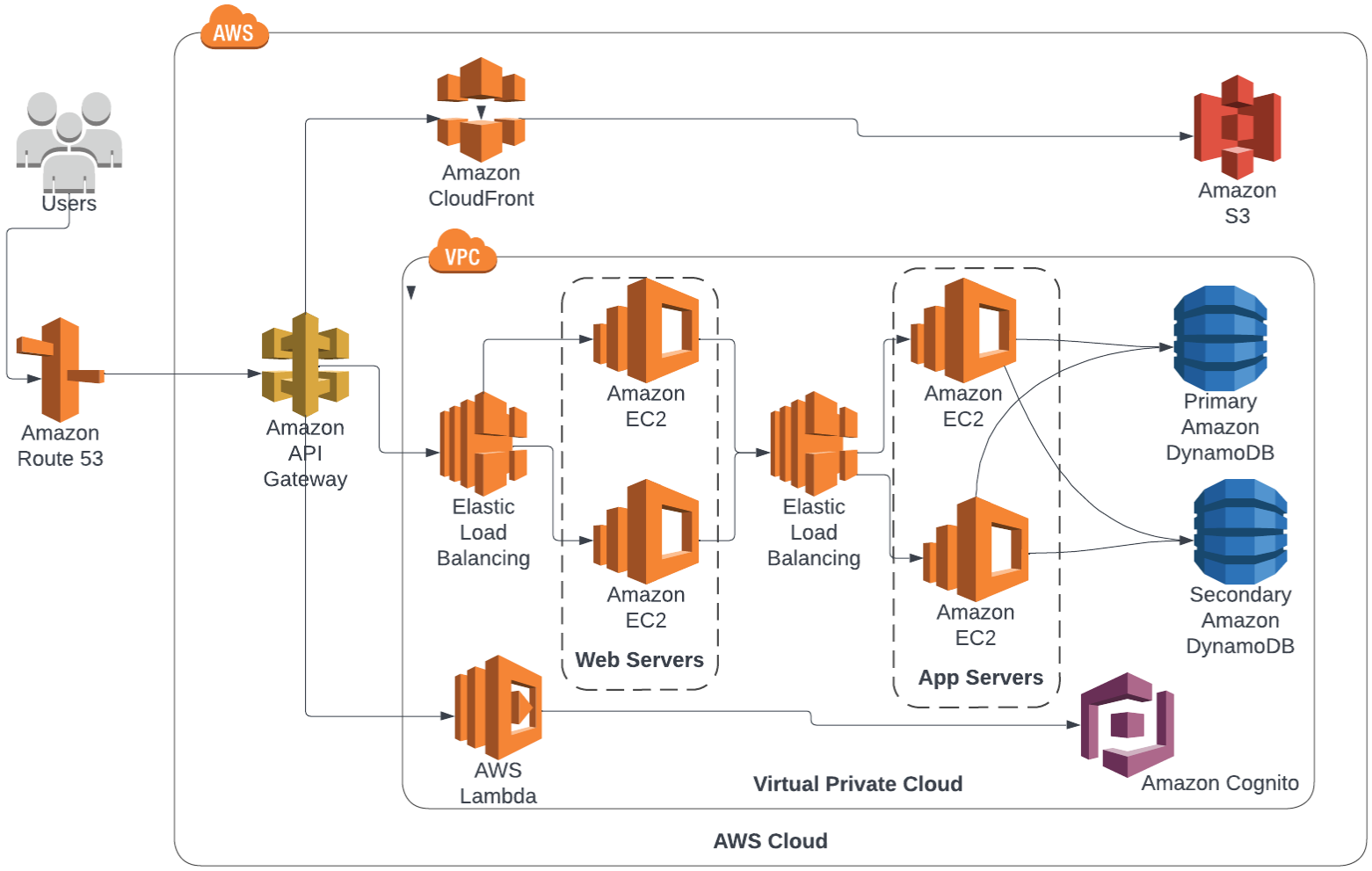
Ingredients_No: Numeric Value; Up to 10 Characters, Cannot be NULL

Ingredients_Price: Decimal Value; Up to 5 Characters ; Can be NULL

Ingredients_Inventory: Numeric Value; Up to 10 Characters; Can be NULL

Ingredients_Quantity: Numeric Value; Up to 10 Characters; Can be NULL

4.2. Cloud Architecture



1. Amazon Route 53: Particularly scalable DNS online web service. This route joins the users requests onto the applications that are either operating on AWS or on premises servers. Interprets URL into an IP address.
2. Amazon API Gateway: Allows you to fully leverage AWS, which assists in building, producing, maintaining, and monitoring APIs at essentially any scale. It is the “entrance” for applications to get access to web store’s data.
3. AWS Lambda: Serverless service, which allows you to run code for any type of application, while also automatically managing computing resources. This service assists customers with the login function, which grants them access to web store information with security.
4. Amazon Cognito: Allows customers/users to create an account, sign in, and gain access to the online web store. This creates a more fluent and secure user experience while browsing and making purchases.
5. Elastic Load Balancing: Allows for evenly distributed traffic to avert any failures that are caused by using a particular resource in excess.
6. Amazon EC2: A manageable data storage for websites. This makes overseeing static data collected on the website easier, and also collecting data which customers interact with, such as the web store's shopping selection.
7. Amazon DynamoDB: A way to manage data and utilizes NoSQL databases to run high performance applications at essentially any scale. This can store the web stores data pertaining to number of ingredients, customers order information, inventory quantities along with account info.
8. Amazon CloudFront: Accelerates the distribution of the contents, which helps deliver fast, secure performance of data transfers along with its convenience for developers.
9. Amazon S3: This is a simple storage, which can retrieve any amount of data at any time, and virtually anywhere. This can store queries on the web store to make customers requesting a preference or a certain category run smooth and efficiently.

1. Network & Web Tier: Manages external/internal network connection and configurations to handle the web/mobile requests) via AWS services such as Route 53, API Gateway, CloudFront etc. AWS route 53 routes end users to web servers by translating names into IP addresses. AWS-CloudFront is a middle-ware that provides cache for users to access recently used data from AWS and S3 data centers. This is helpful in the delivery of content services. CloudFront also speeds up distribution from S3 to users which results in a boost in performance and latency within the web server.

2. Application Server Tier: The application server tier is managed by Amazon EC2. When a user is in the process of choosing between recipes, meals, ingredients, quantity, and other customizations, their profile is stored by Amazon EC2 and updated in Primary Amazon DynamoDB and Secondary Amazon DynamoDB. This keeps updating as users make more actions as they progress through the purchasing process.

3. Database Tier: The database tier manages Database Servers and Data storage via AWS services such as RDS, DynamoDB, and S3. Amazon Relational Database Server RDS is used in order to store all data in a relational format. Amazon DynamoDB is used for transient data, which is managed by a Python program and Django web framework. This code manages data access and management tasks. This is a fully managed NoSQL database service which excels in storing data from any source at any time. S3 is also helpful for developers making web-scaling computations. These components all work together in order to efficiently store data, retrieve data, and query existing data.

4. Authentication: Due to security reasons and in order to better protect customer data, we will be implementing Amazon Cognito. Cognito, a customer identity and access management (CIAM) service will authenticate users during the login and checkout processes. Amazon Cognito will detect unauthorized users and allow only administrators to access the database. Lastly, Amazon Cognito is a user friendly way to reduce security risks and simplify the login process.
