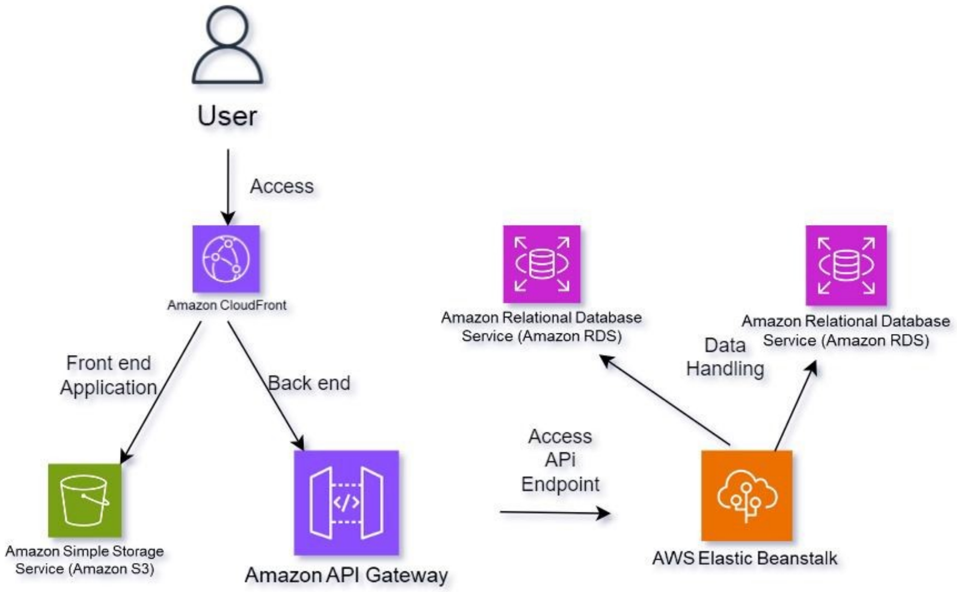
3-Tier AWS Architecture Description for Restaurant Reservation System

It has a 3-tier distributed architecture pattern in which it has the Presentation Tier, also known as Frontend, the Logic Tier, also known as Backend and the Data Tier, also known as Database. All the tiers are developed using AWS services to broaden capability, security and reliability.


User
Access
Amazon CloudFront
Front end Application
Back end
Amazon Simple Storage Service (Amazon S3)
Amazon API Gateway
Access APi Endpoint
AWS Elastic Beanstalk
Amazon Relational Database Service (Amazon RDS)
Data Handling
Amazon Relational Database Service (Amazon RDS)

1. Presentation Tier (Frontend)

AWS S3 (Simple Storage Service): The frontend React application is stored on AWS S3 catalogue as a static website. Such configuration also guarantees that the static assets required by the user such as HTML, CSS, Javascripts, and images have high availability and low latency.

AWS CloudFront (CDN): CloudFront is distribution service that simply caches static assets from S3 across multiple edge locations globally. This enhances efficiency and quickness in feeding content to the end-users thus reducing the lag time.

User Access via Browser: Frontend which is again in the frontend layer is accessed through web browsers by the end users. They make requests to the frontend, which controls the backend through application programming interfaces, shortened to APIs.

2. Logic Tier (Backend)

AWS API Gateway: API Gateway is used to build RESTful APIs to which the frontend can connect with the backend services. It receives all API requests and offers secure and scalable interfaces for handling those requests.

AWS Lambda (Optional): It is also possible to use a lambda function if there is a need for serverless processing of concrete actions for instance, reservations' validation or sending confirmation emails. This is done to have additional optional processing to work on some specific things to do in the backend.

AWS EC2 (Elastic Compute Cloud): Django application of the backend part is located on the EC2 instance. EC2 delivers elastic computing capacity for web and application servers delegated to performing the business logic, processing reservations, and interacting with databases.

AWS Elastic Load Balancer (ELB): An ELB is created to monitor the instances so that it can fan incoming traffic on multiple instances for the purpose of load balancing and fault tolerance. This makes it possible to also achieve high availability and reliability of the backend services.

3. Data Tier (Database)

AWS RDS (Relational Database Service): The PostgreSQL is located in AWS RDS. The database is handled by RDS for aspects such as Backups, Patching, Scaling and Data Durability and Availability. This tier needs to have all the important information in terms of reservations, menus and other information.

VPC (Virtual Private Cloud): By default, the RDS instance is configured in a private subnet within the VPC and hence it cannot be accessed through internet. Another one is the instance of EC2 from the same VPC connected securely with the RDS instance.

4. Security

Security Groups and NACLs (Network Access Control Lists): Security Groups for the EC2 instances and RDS have been created in-order to manage traffic coming into and out of the instances. NACLs are essentially applied at the Subnet level to give extra layer of security.

IAM Roles and Policies: IAM roles and policies are used to grant permission on the various AWS services. For instance, the EC2 instance can possess a policy that permits it interact with RDS and denies it access to the rest of AWS environment.

5. Monitoring and Logging

AWS CloudWatch: CloudWatch is applied for the monitoring of the EC2 instances, API Gateway and the majority of AWS solutions. It gathers logs, alerts on and present system performance and health status.

AWS CloudTrail: API calls as well as other activities occurring within an enterprise CloudTrail is used to log and track the actions.