Architecture Overview

High-Level Flow:

1. User calls support line (Amazon Connect).
2. The Connect flow invokes **Amazon Lex**, which determines user intent.
3. Lex forwards input to **AWS Lambda**.
4. Lambda processes the request and stores history in **DynamoDB**.
5. Based on the intent, user is either helped by the bot or transferred to a human agent.

Key AWS Services Used:

- **Amazon Connect** – voice interface
- **Amazon Lex** – natural language understanding
- **AWS Lambda** – backend logic
- **Amazon DynamoDB** – stores ticket history
- **IAM Roles & Policies** – permission management

Challenges Faced

- **Lex Bot Alias Not Appearing** in Amazon Connect: Had to ensure the bot version was published and a named alias (e.g., LiveBotAlias) was linked to it. Re-selecting the bot and alias inside the Connect flow helped resolve it.
- **Lambda Permissions**: Ensuring that Lambda had bedrock:InvokeModel, dynamodb:*, and logs:* access via IAM role.
- **Prompt Linking Confusion**: We had to make sure each block had both success and error paths connected to ensure flow publishing worked.
- **Publishing Failures**: Amazon Connect validation errors were resolved by assigning all required intents and matching aliases correctly.

Build Steps

Step-by-Step Setup Summary:

1. **Create a DynamoDB Table**

- Table name: CustomerHistory
- Partition Key: CustomerID (String)
- Sort Key: Timestamp (String)

2. **Create a Lambda Function**
   - Language: Python 3.x
   - Code: Handles user input, fetches/stores from DynamoDB, invokes Amazon Bedrock (or Lex)
3. **Create a Lex Bot**
   - Add intents like CheckHoursIntent, CreateTicketIntent
   - Publish the bot and create a new alias (LiveBotAlias)
4. **Set Up Amazon Connect**
   - Create an instance
   - Claim a number (toll-free/local)
   - Set up contact flow:
     - Play prompt
     - Get customer input (Lex)
     - Check contact attribute
     - Play fallback prompt
     - Transfer to agent queue
5. **Link Lex Bot to Connect**
   - Add Lex bot and alias under Connect Settings > Amazon Lex
   - Use in Get customer input block
6. **Deploy and Test**
   - Call the number, speak with the bot
   - If unrecognised input → fallback prompt → human queue