Creating and Integrating an Amazon Lex Bot for Dental Appointments

Overview

1. Create and test the bot using Amazon Lex.

2. Create and configure an AWS Lambda function.

3. Integrate the bot with the Lambda function.

4. Deploy the bot as a static website using Amazon S3.

Steps

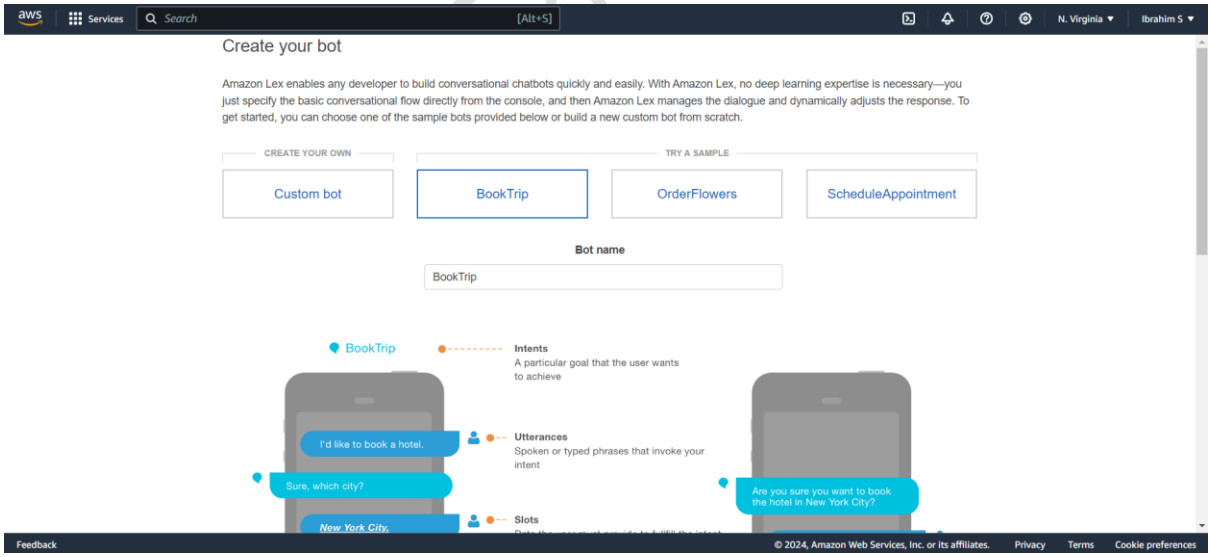
I. Create and Test the Bot

1. Navigate to Amazon Lex Console

   - Use Version 1 (V1) of Amazon Lex.

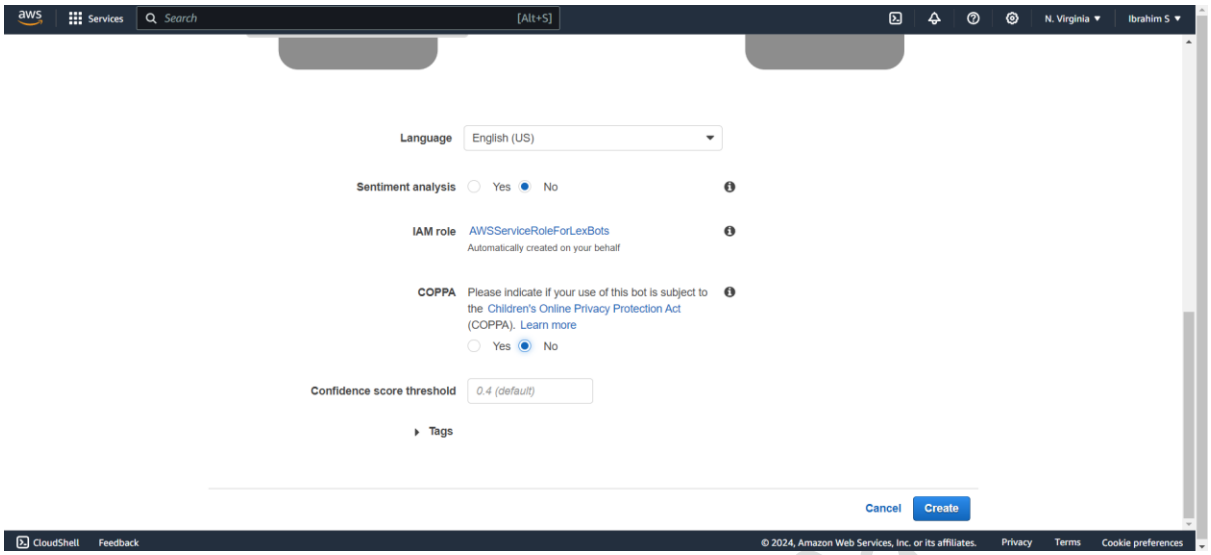
2. Select Blueprint

   - Use the `schedule appointment` blueprint.



3. Configure Basic Settings

- Configure the bot with necessary settings.



4. Understand Key Features

- Sample Utterances: Phrases that invoke the intent.
- Lambda Initialization and Validation: Business logic to validate user input.
- Context: Carries information from one intent to another.
- Output Tags: Set the output context for intents.

5. Build and Test the Bot

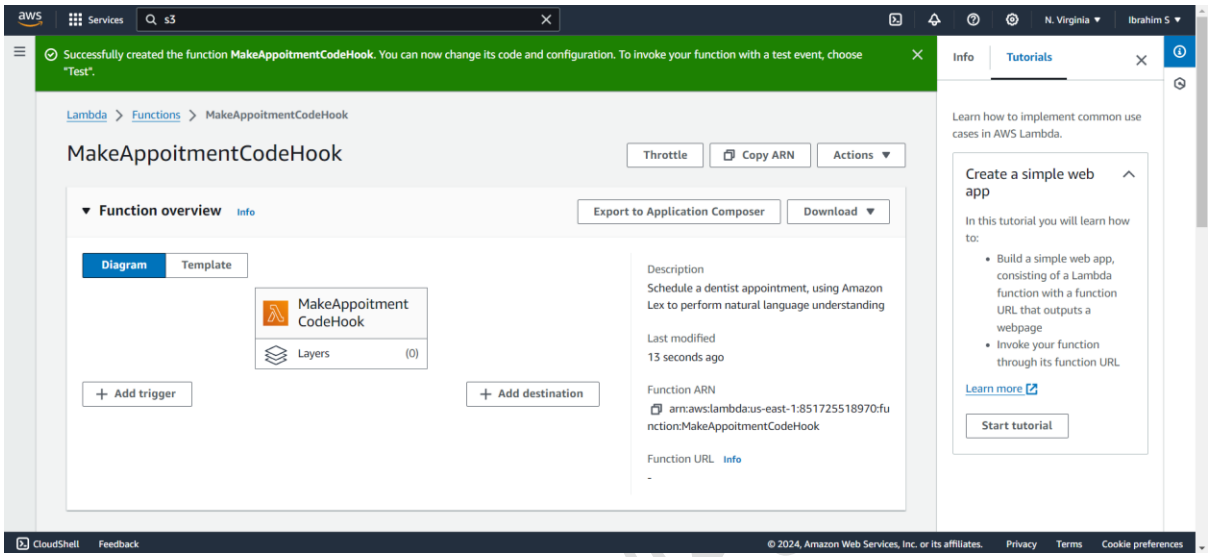
- Test the pre-built bot to understand its functionality.

II. Create and Configure an AWS Lambda Function

1. Create a Lambda Function

- Use the `make an appointment with Lex` blueprint.

- Name the function and create an IAM role (e.g., `myLexRole`).

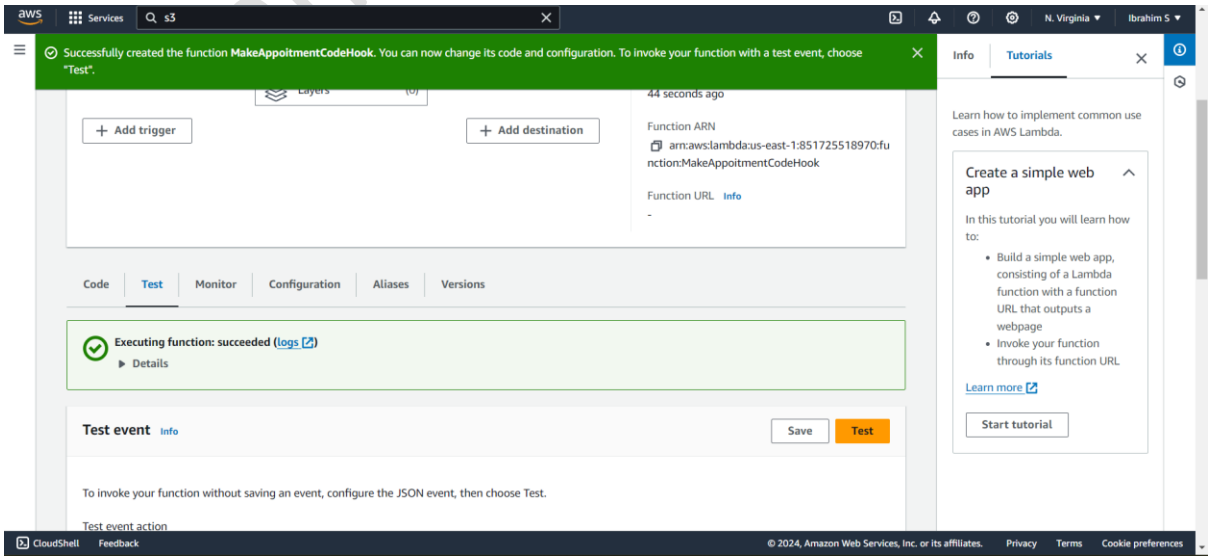


2. Automate Prompts

- The Lambda function automates prompts that were previously provided manually.

3. Test the Function

- Ensure the function executes successfully.

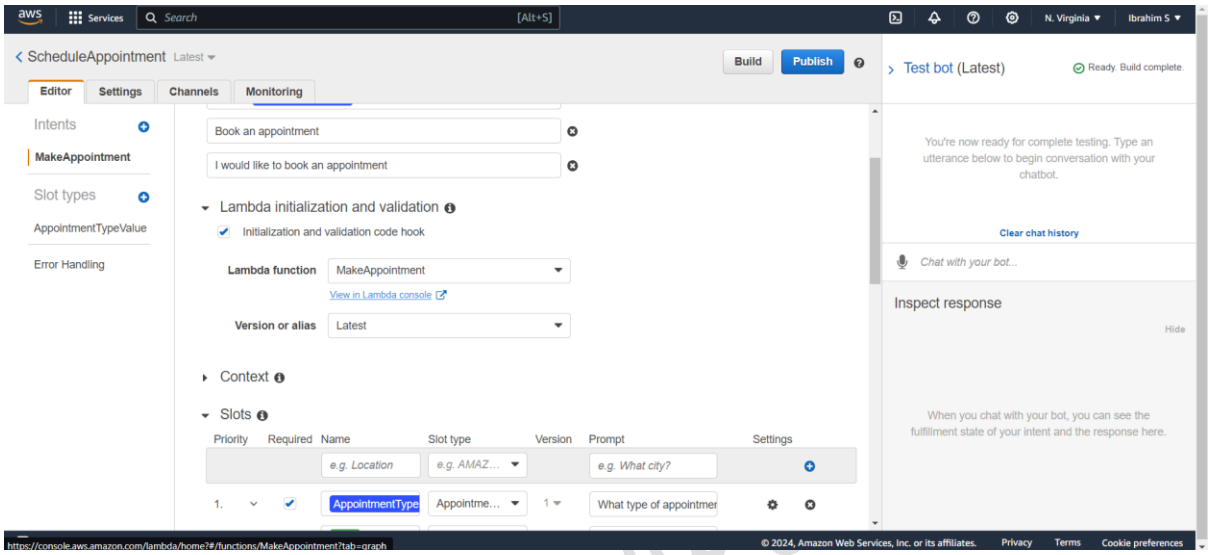


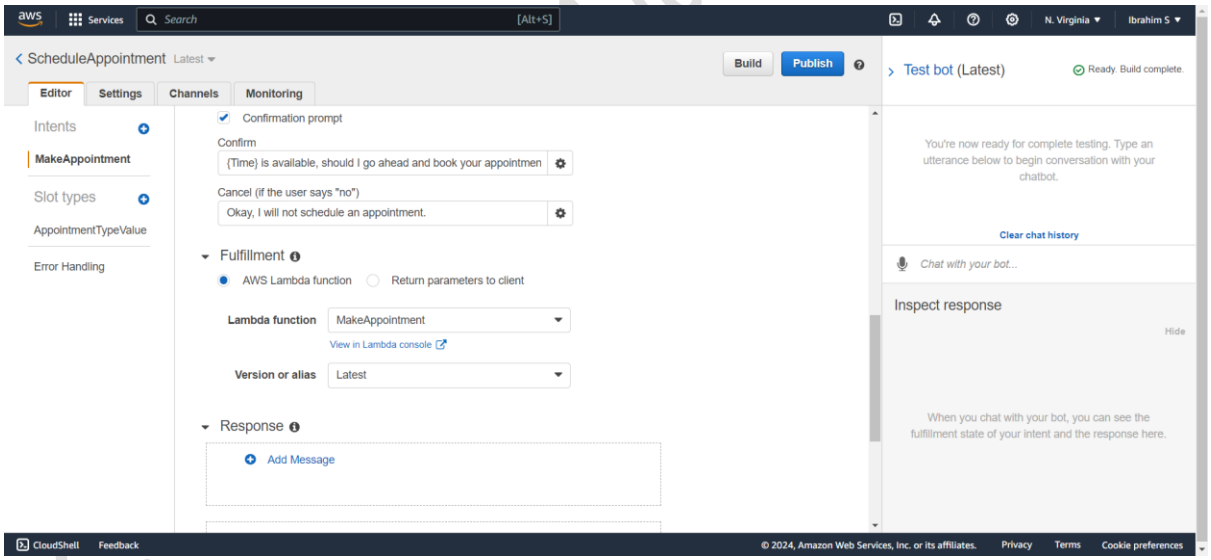
III. Integrate Lambda Function with Amazon Lex Bot

1. Link Lambda Function to Lex Bot

- In Amazon Lex console, set the Lambda function as the code hook for initialization and fulfilment.

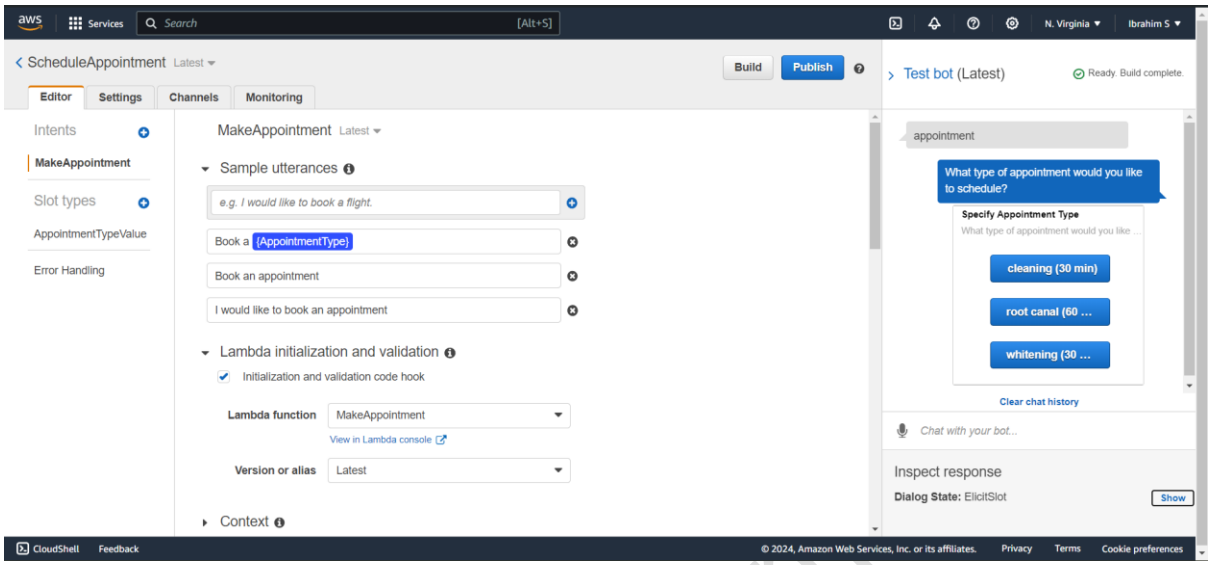
- Save the changes and rebuild the bot.





2. Test the Integrated Bot

- Confirm the bot now uses the Lambda function to provide options and schedule appointments automatically.

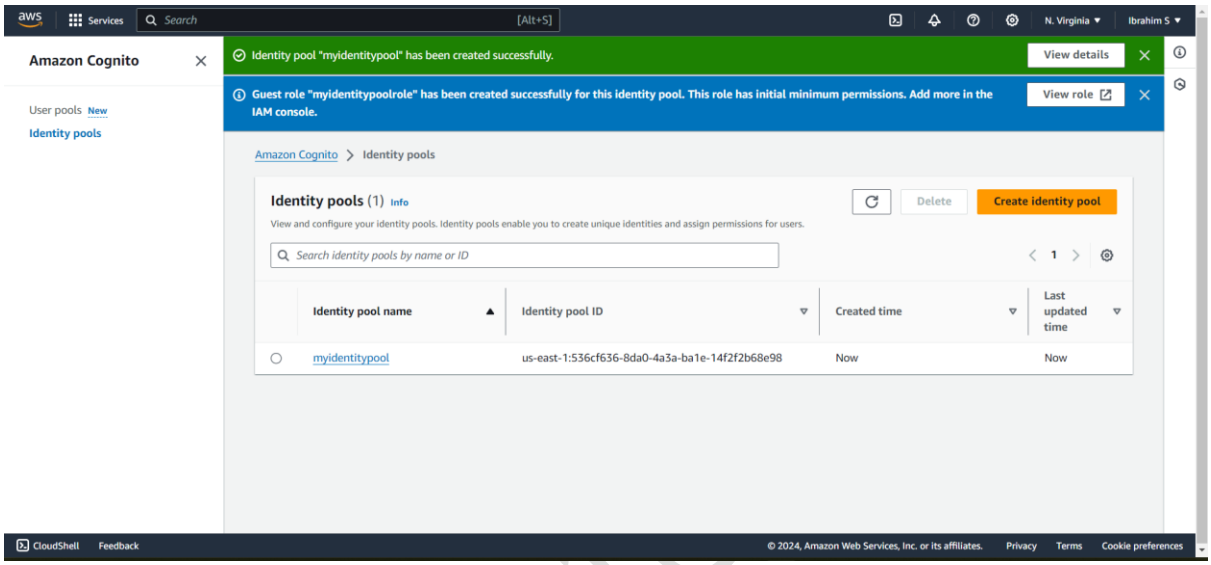


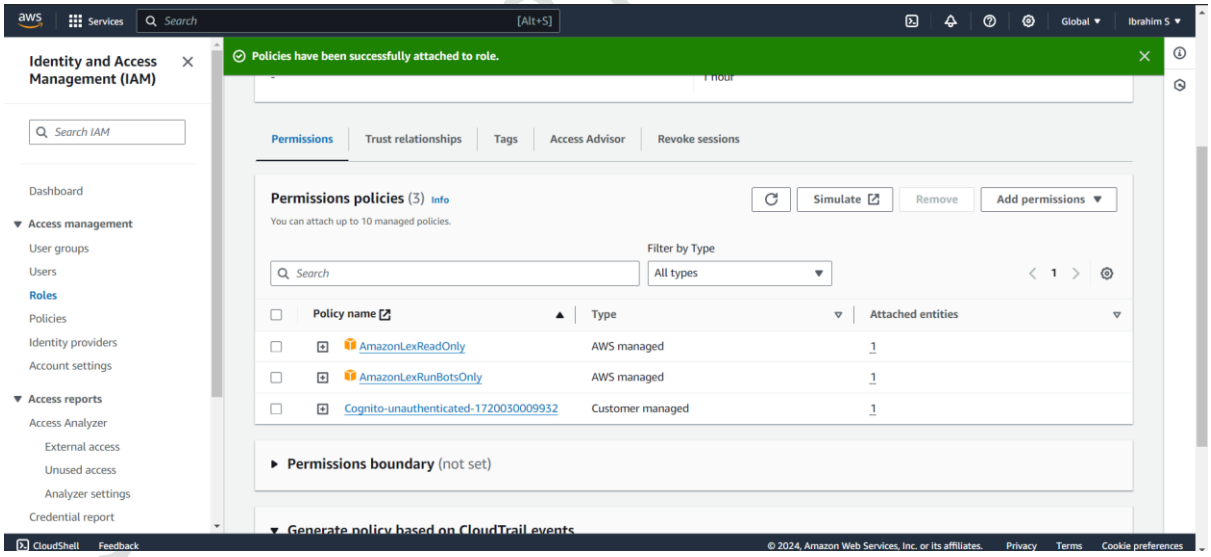
IV. Deploy as a Static Website using Amazon S3

1. Set Up Amazon Cognito for Authentication

- Create an identity pool with guest access.

- Configure an IAM role with necessary permissions (`AmazonLexReadOnly` and `AmazonLexRunBotsOnly`).

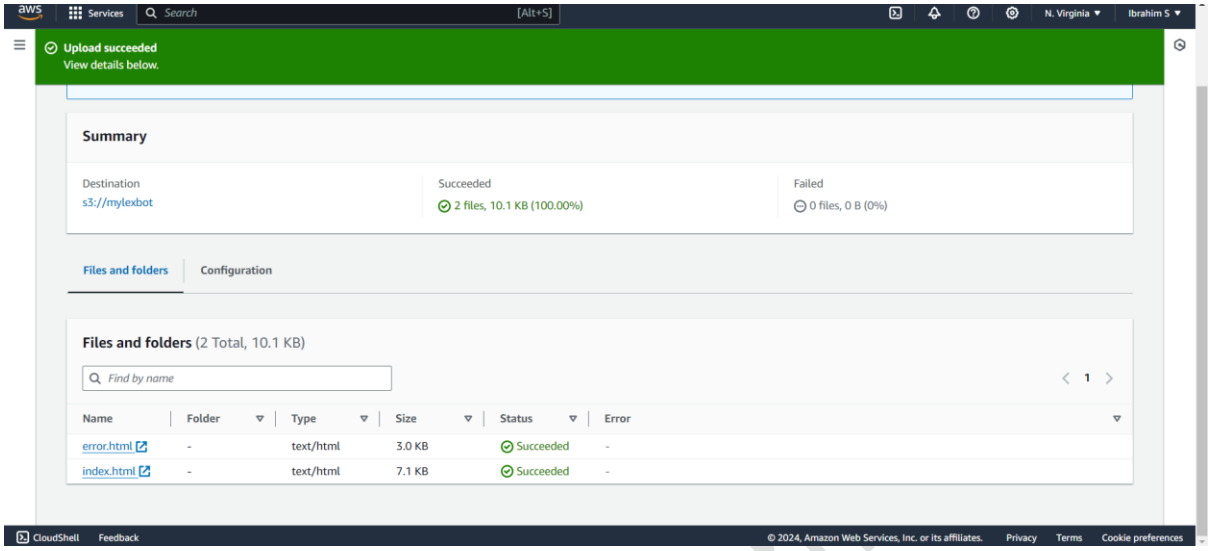




2. Create an S3 Bucket

- Name the bucket and configure it for static website hosting.

- Upload `index.html` and `error.html`.



3. Configure Bucket Policy

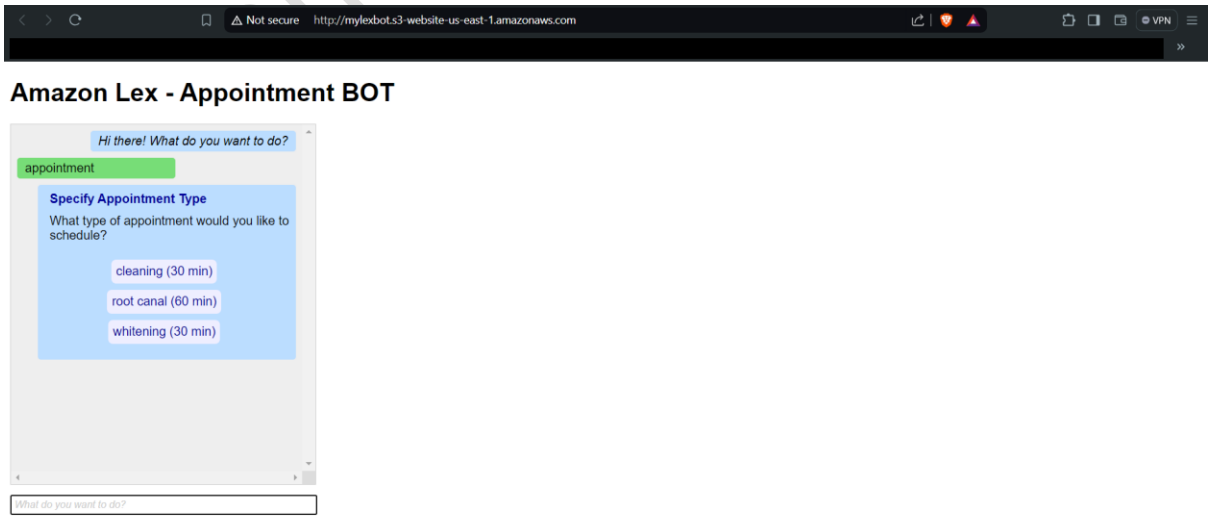
- Allow public read access to the objects in the bucket.

4. Host and Test the Static Website

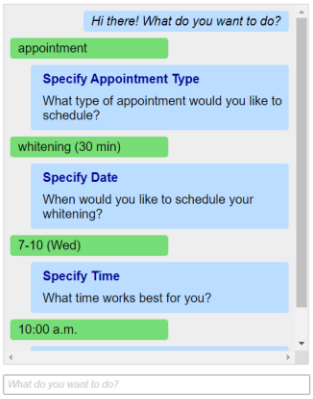
- Enable static website hosting in the bucket properties.

- Add bucket policy and specify `index.html` and `error.html` files.

- Access the static website via the provided URL and test the bot.



Amazon Lex - Appointment BOT



Conclusion

- Created an Amazon Lex Bot
- Configured and integrated a Lambda function
- Hosted the bot on Amazon S3 as a static website