Serverless Voice Assistant with AWS Polly and Lex

A Course Project Report Submitted in partial fulfillment of the course requirements for the award of grades in the subject of

CLOUD BASED AIML SPECIALITY
(22SDCS07A)

by

P HARISHITH CHOWDARY

(2210030050)

Under the esteemed guidance of

Ms. P. Sree Lakshmi
Assistant Professor,
Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K L Deemed to be UNIVERSITY

Aziznagar, Moinabad, Hyderabad,
Telangana, Pincode: 500075

April 2025

K L Deemed to be UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is Certified that the project entitled **"Serverless Voice Assistant with AWS Polly and Lex"** which is a experimental &/ theoretical &/ Simulation&/ hardware work carried out by P Harishith Chowdary (2210030050), in partial fulfillment of the course requirements for the award of grades in the subject of **CLOUD BASED AIML SPECIALITY**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

Ms.P.Sree Lakshmi

Course Coordinator

Dr. Arpita Gupta

Head of the Department

Ms. P. Sree Lakshmi

Course Instructor

CONTENTS

Page No.

1. INTRODUCTION

A serverless voice assistant is an AI-powered system that enables natural voice interactions without requiring dedicated server infrastructure. By leveraging AWS services like Amazon Polly (for text-to-speech conversion) and Amazon Lex (for building conversational interfaces), businesses can create scalable, cost-efficient, and intelligent voice assistants for various applications. The rise of artificial intelligence (AI) and cloud computing has enabled businesses to develop serverless voice assistants that provide natural language interactions and lifelike speech synthesis. AWS services like Amazon Polly and Amazon Lex offer powerful tools to build scalable, efficient, and cost-effective voice-enabled applications.

Amazon Polly is a cloud-based text-to-speech (TTS) service that converts text into realistic human-like speech. With multi-language support, neural voices, and cost-effectiveness, Polly enhances applications by improving engagement, accessibility, and user experience. It is widely used in industries such as e-learning, customer service, IoT, and accessibility solutions.

Amazon Lex is a conversational AI service that enables developers to build voice and text-based chatbots. Using automatic speech recognition (ASR) and natural language understanding (NLU), Lex processes user queries and generates intelligent responses. With serverless architecture, deep learning, and AWS integrations, it allows businesses to create advanced chatbots without requiring AI expertise.

By combining Amazon Polly and Amazon Lex, organizations can build intelligent voice assistants for customer service, healthcare, travel, retail, and hospitality. Companies like Marriott, MedWhat, and Fathom Tech leverage these technologies to automate guest services, provide medical information, and simplify corporate travel bookings. This paper explores real-world applications of AWS Polly and Lex, analyzing their impact on various industries and their role in enhancing automation and business efficiency.

2. AWS Services Used as part of the project

- **Amazon Lex –** for building the conversational interface (voice/chatbot).
- **Amazon Polly –** for converting text responses into lifelike speech.
- **Amazon CloudWatch –** for logging and monitoring Lambda functions and Lex interactions.
- **AWS IAM –** for managing permissions and secure access to services.

3. Steps involved in solving project problem statement

Steps involved in solving the project problem statement:

1. **Define Use Case and Workflow**
   - Decide what the voice assistant should do (e.g., answer FAQs, control devices, book appointments).

2. **Create a Lex Bot**
   - Define intents, utterances, and slots for user interaction.
   - Set fulfillment through AWS Lambda if logic is needed.

3. **Configure AWS Lambda Function**
   - Write backend logic for processing user input and generating dynamic responses.
   - Connect the Lambda function with the Lex bot.

4. **Integrate Amazon Polly**
   - Use Polly in the Lambda function to convert text responses to speech.

5. **Deploy API (Optional)**
   - Use API Gateway to expose endpoints for frontend or other applications.

6. **Build Frontend (Optional)**
   - Create a simple web or mobile interface to interact with the bot using audio input/output.

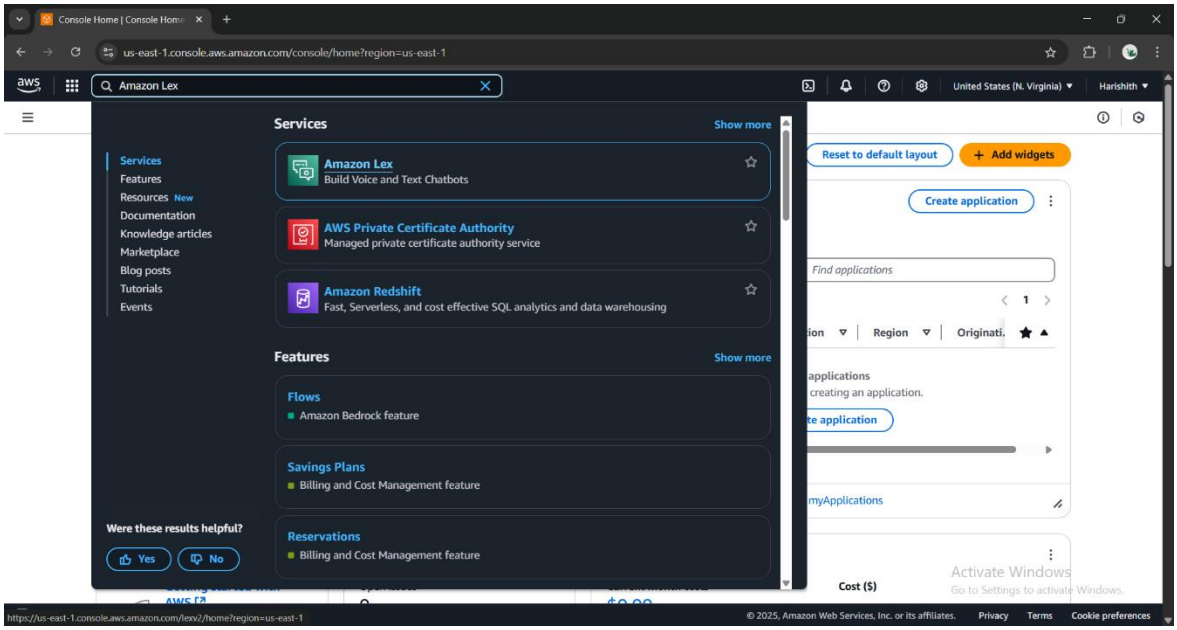
7. **Store/Log Data (Optional)**
    - Use Amazon S3 or DynamoDB for storing session data, logs, or transcripts.

8. **Test and Debug**
    - Test the bot through Lex console or frontend.
    - Use CloudWatch for monitoring and debugging.

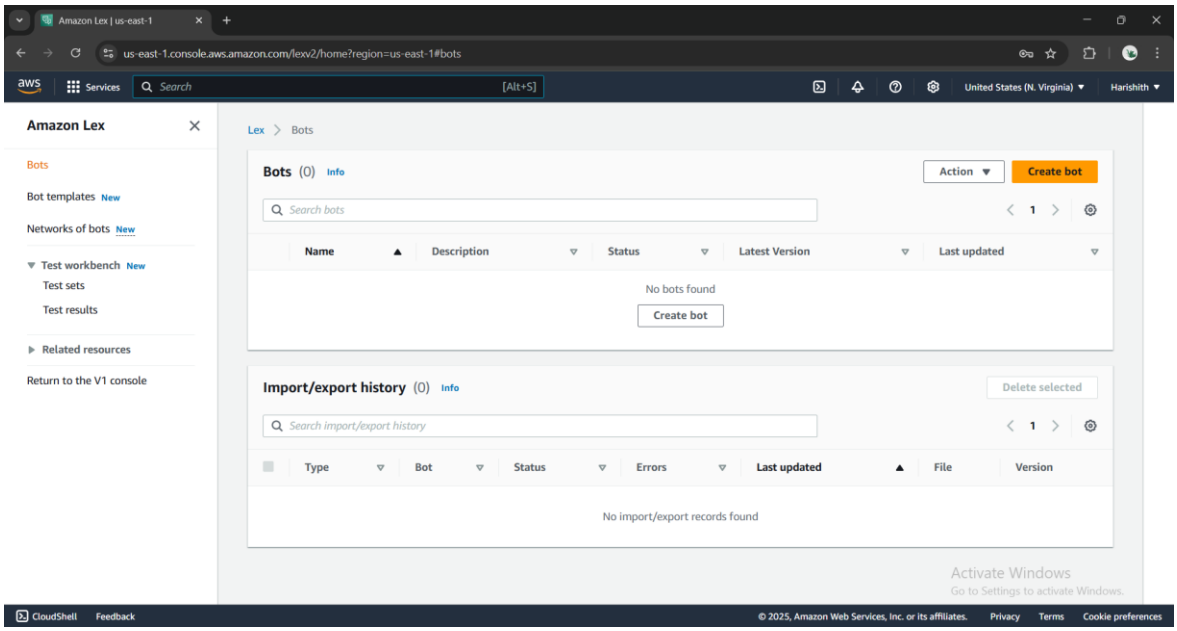
9. **Deploy and Monitor**
    - Deploy the solution and use CloudWatch for performance and error tracking.

4. Stepwise Screenshots with brief description

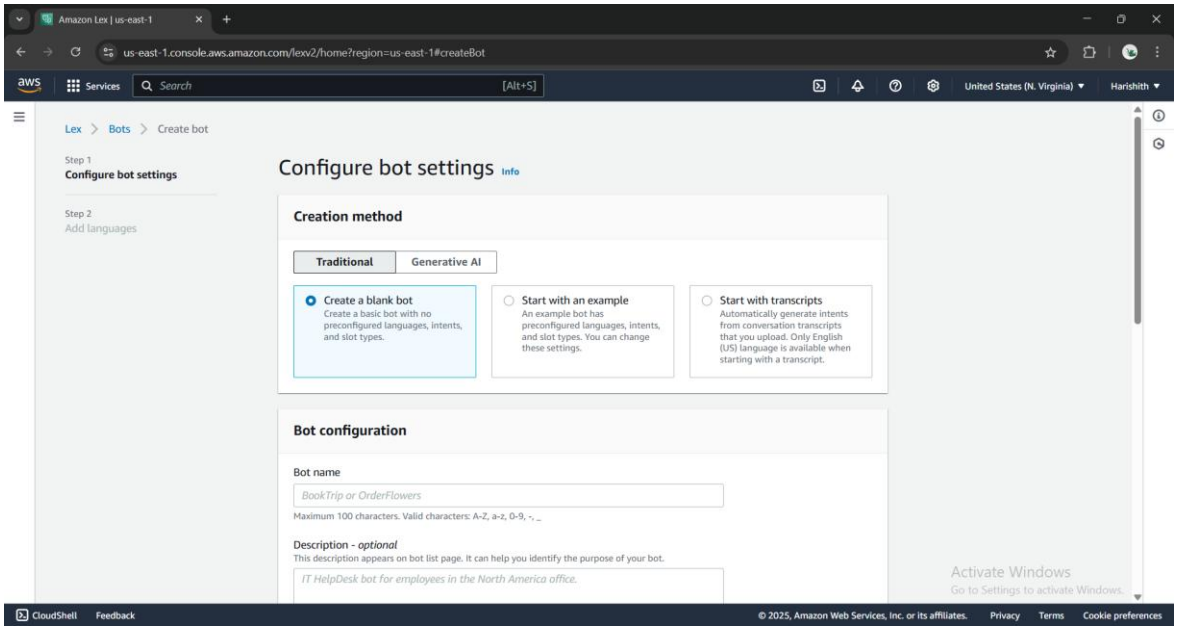
Step 1: Search for Amazon Lex service in aws search bar and click on Amazon Lex service.



Step 2: Now we have to Create a new Bot.

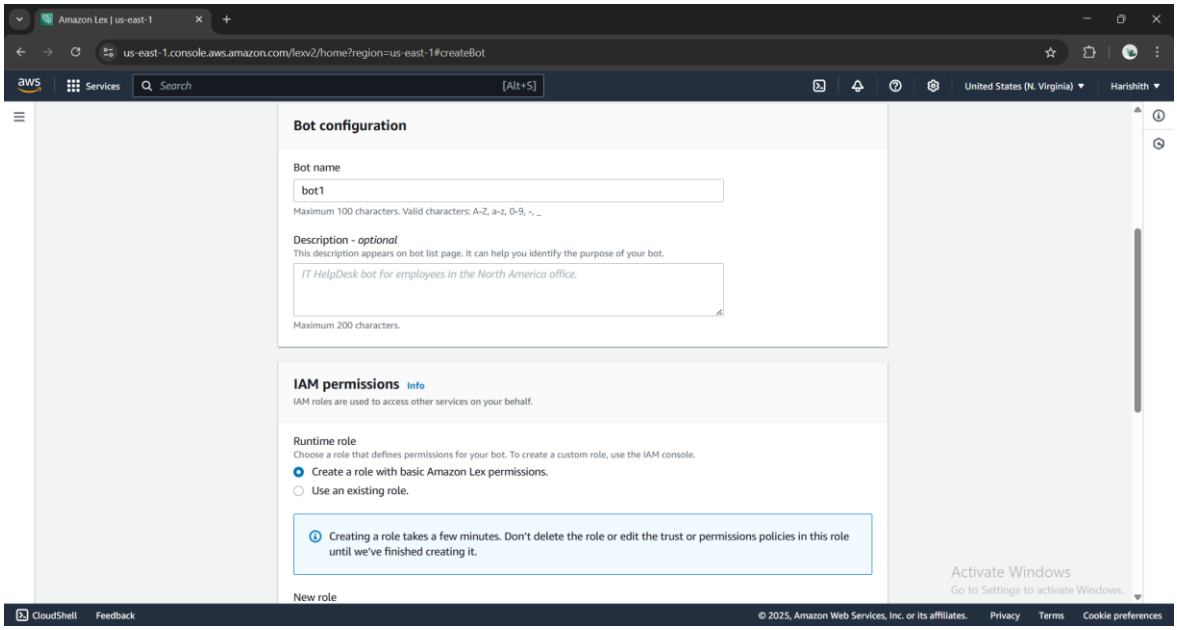


Step 3: Now we have to configure bot settings, here we are selecting a traditional bot with create a blank bot selection.

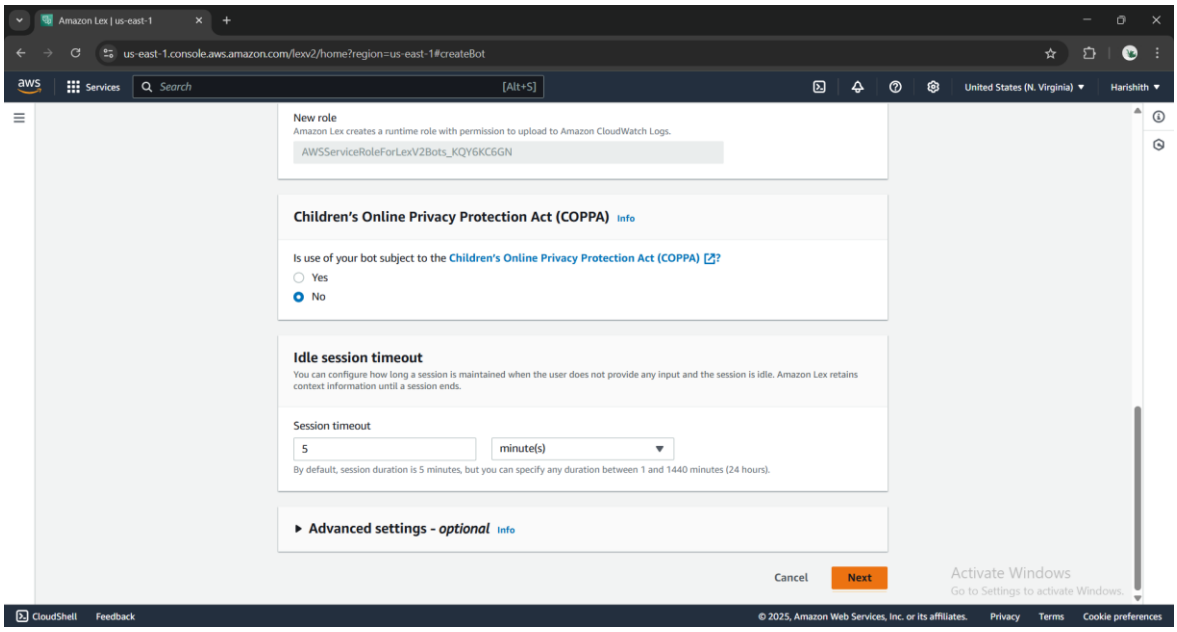


Step 4: Now set a name to your bot and in IAM permissions under Runtime role

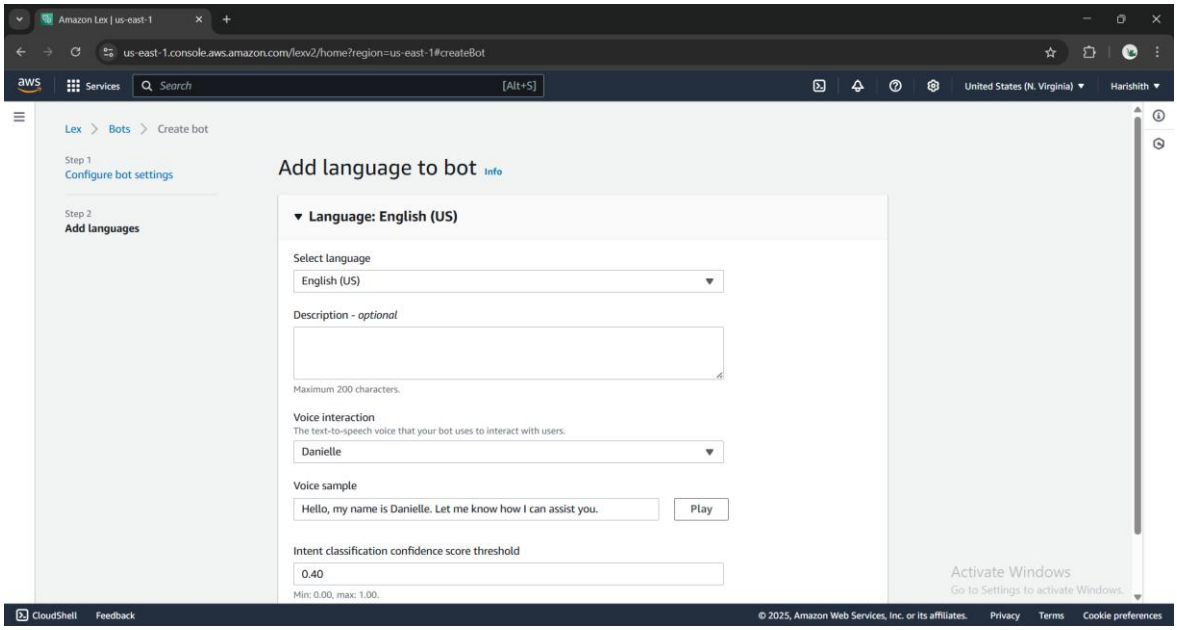
Choose a role that defines permissions for your bot. To create a custom role, use the IAM console. Create a role with basic Amazon Lex permissions.



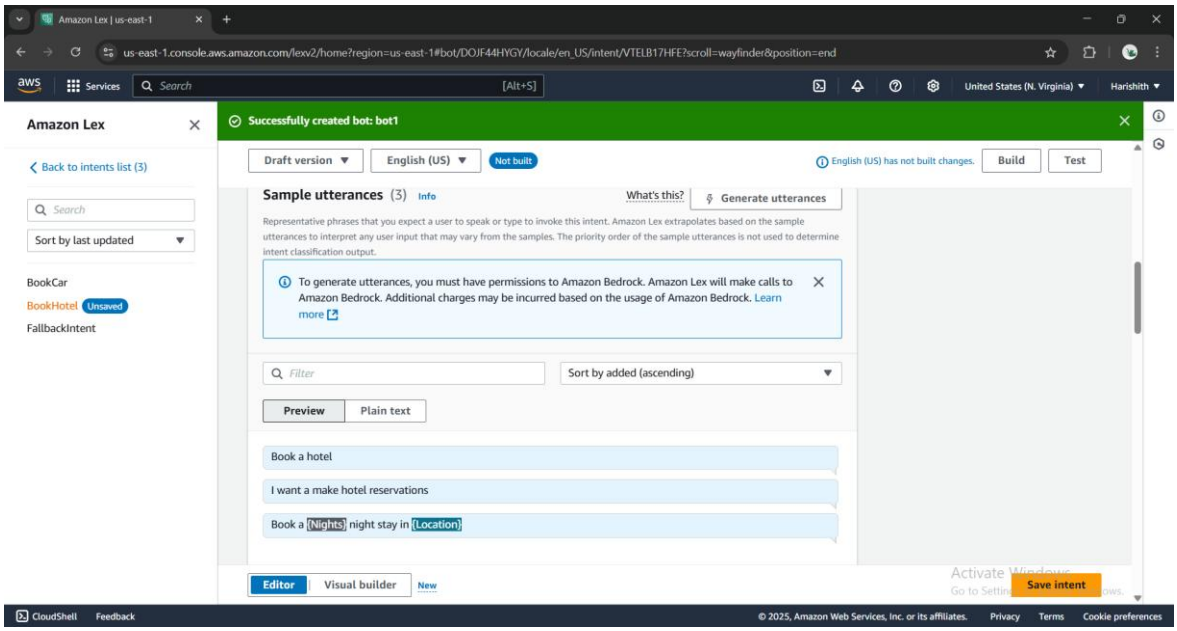
Step 5: Now in Children's Online Privacy Protection Act (COPPA) Info under Is use of your bot subject to the Children's Online Privacy Protection Act (COPPA)? Select No, and click on next.

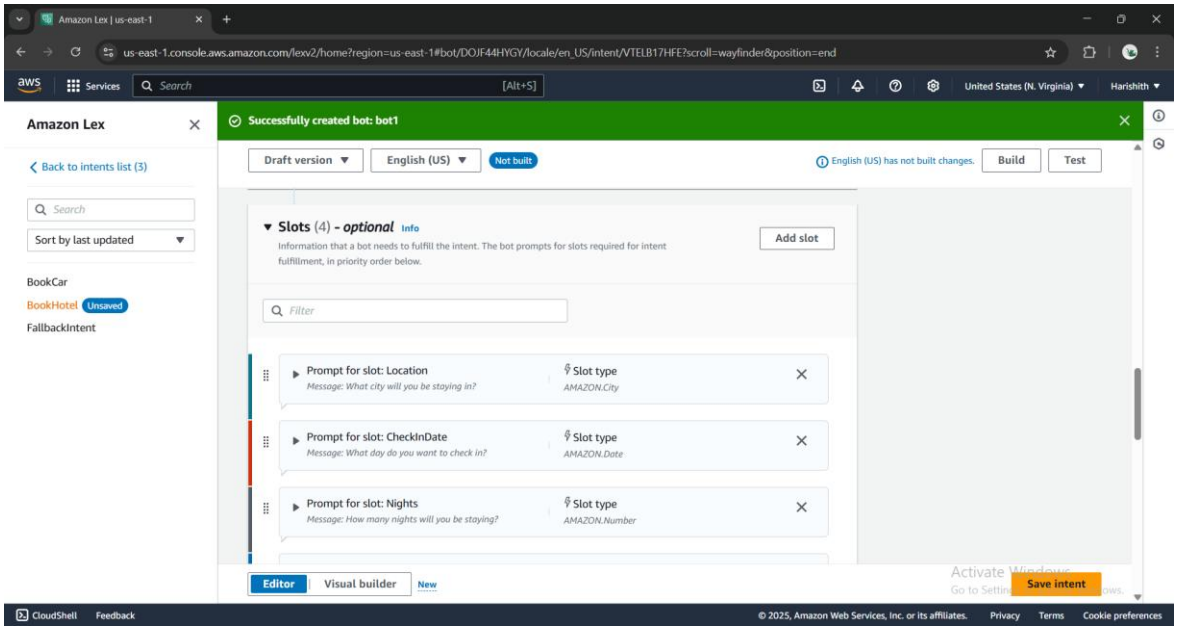


Step 6: Now you have to select your voice of the in which the bot will convey its message to you. And after selecting your bot voice then click on next.

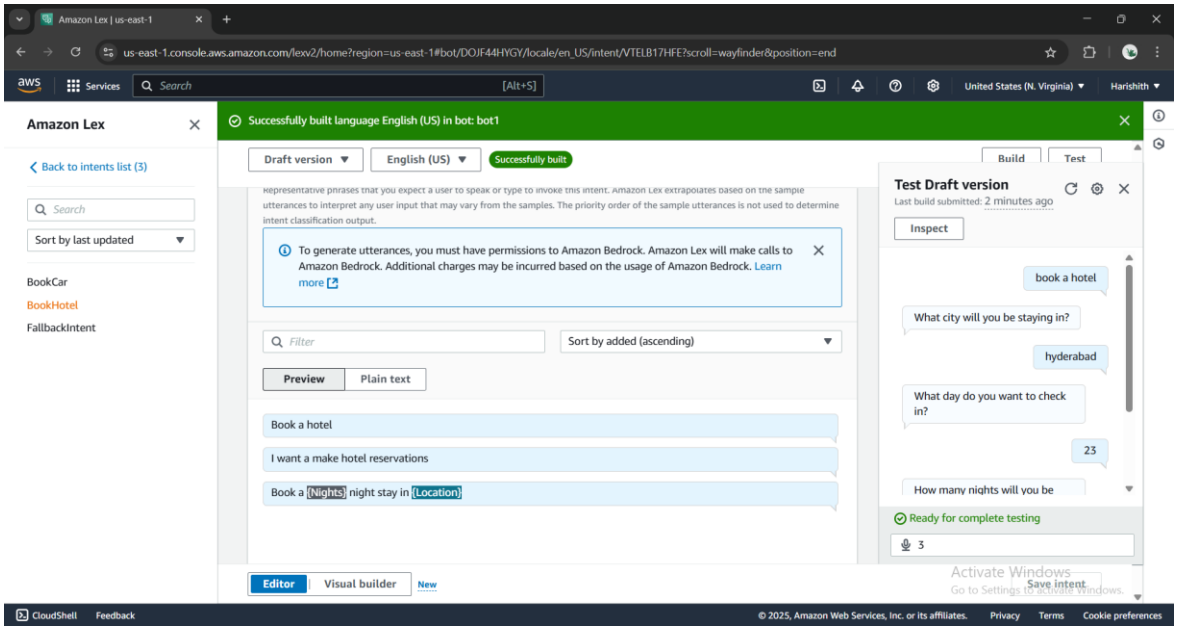


Step 7: Now add and fill fields like Sample utterances, Initial response, Slots, Confirmation, Fulfillment, Closing response. And after that click on save intent and click on build button to build your bot.





Step 8: Now you can test your final bot by clicking on test button.



5. Learning Outcomes:

- Gained hands-on experience with serverless architecture using AWS services.
- Learned to build and train conversational interfaces using Amazon Lex.
- Understood text-to-speech conversion using Amazon Polly.
- Developed skills in integrating AWS Lambda with Lex and Polly.
- Understood secure access control using IAM roles and policies.
- Learned how to monitor and troubleshoot serverless applications using CloudWatch.

6. Conclusion:

This project demonstrated the potential of building a Serverless Voice Assistant using AWS's suite of powerful cloud services, specifically Amazon Lex, Amazon Polly, and AWS Lambda. The core objective was to create an intuitive and scalable conversational interface that processes voice commands and responds with lifelike speech, utilizing the full capabilities of a serverless architecture.

By leveraging Amazon Lex, the voice assistant was able to recognize user intents, understand natural language, and engage in context-aware conversations. Integrating Amazon Polly into the system provided a dynamic way to convert text responses into high-quality speech, enhancing the overall user experience with lifelike voice output.

One of the most notable advantages of this solution is the serverless nature of the architecture. AWS Lambda executed the logic behind the interactions without the need for server management or provisioning, ensuring scalability and reducing operational overhead. The Lambda functions also seamlessly integrated with Amazon Lex to handle intents and interact with other AWS services, providing flexibility and extensibility to the system.

Amazon CloudWatch allowed for robust monitoring, helping identify and troubleshoot issues, and ensuring smooth functioning of the voice assistant. The use of IAM roles and policies added an essential layer of security, ensuring that each service interacted securely without exposing sensitive data or configurations.

Ultimately, this project serves as a foundation for building scalable, efficient, and easily maintainable voice-driven applications. The voice assistant can be extended to various real-world applications, including virtual customer support, smart home control, and personal assistants, demonstrating the power and flexibility of AWS services in building cloud-native applications. This architecture can be further optimized by introducing advanced features like multi-language support, natural language processing, or AI-driven analytics to enhance user interaction.

The project also underlines the importance of adopting serverless frameworks in modern cloud computing, as they significantly reduce infrastructure management burdens and promote cost-efficiency, making them an ideal choice for small to medium-scale applications. Future work can explore the integration of additional AWS services such as Amazon Transcribe for speech-to-text functionality, Amazon DynamoDB for session storage, and Amazon Rekognition for image-based interactions, providing a more comprehensive and interactive user experience.

7. References:

- Amazon Lex Documentation: Amazon Web Services, Inc. "What is Amazon Lex?" Accessed April 2025. Available at: https://docs.aws.amazon.com/lex/latest/dg/what-is.html
- Amazon Polly Documentation: Amazon Web Services, Inc. "What is Amazon Polly?" Accessed April 2025. Available at: https://docs.aws.amazon.com/polly/latest/dg/what-is.html
- AWS Lambda Documentation: Amazon Web Services, Inc. "AWS Lambda Overview." Accessed April 2025. Available at: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html
- AWS IAM Documentation: Amazon Web Services, Inc. "Introduction to AWS Identity and Access Management." Accessed April 2025. Available at: https://docs.aws.amazon.com/iam/latest/UserGuide/introduction.html
- Amazon CloudWatch Documentation: Amazon Web Services, Inc. "Amazon CloudWatch." Accessed April 2025. Available at: https://docs.aws.amazon.com/cloudwatch/
- AWS Serverless Application Model (SAM): Amazon Web Services, Inc. "What is AWS Serverless Application Model?" Accessed April 2025. Available at: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/what-is-sam.html
- Amazon Transcribe: Amazon Web Services, Inc. "What is Amazon Transcribe?" Accessed April 2025. Available at: https://aws.amazon.com/transcribe/
- Amazon Rekognition: Amazon Web Services, Inc. "What is Amazon Rekognition?" Accessed April 2025. Available at: https://aws.amazon.com/rekognition/