

Build a Chatbot with Amazon Lex



vishnu vardhan

The screenshot shows the AWS Lambda console interface. At the top, it displays the message "Successfully built language English (US) in bot: BankerBot". Below this, there are tabs for "Draft version" and "English (US)". A "Build" button is visible. On the left, a sidebar lists intents: "WelcomeIntent" and "FallbackIntent". The main area shows a preview of the bot's responses to various user inputs. A "Test Draft version" window is open, showing a conversation between a user ("Hi") and the bot ("Hello"). The bot also responds to "I need help" and "Can you help me?". A message input field at the bottom allows for testing additional messages. A "Ready for complete testing" button is present. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and a footer with copyright information.



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service that allows developers to build conversational interfaces for applications using voice and text.

How I used Amazon Lex in this project

I have used Amazon Lex to create a banker chat bot which helps customers to know bank balance and make transfers.

One thing I didn't expect in this project was...

i didnt expect one thing that its so easy to create our own custom chatbot.

This project took me...

it took me arround 30 mins

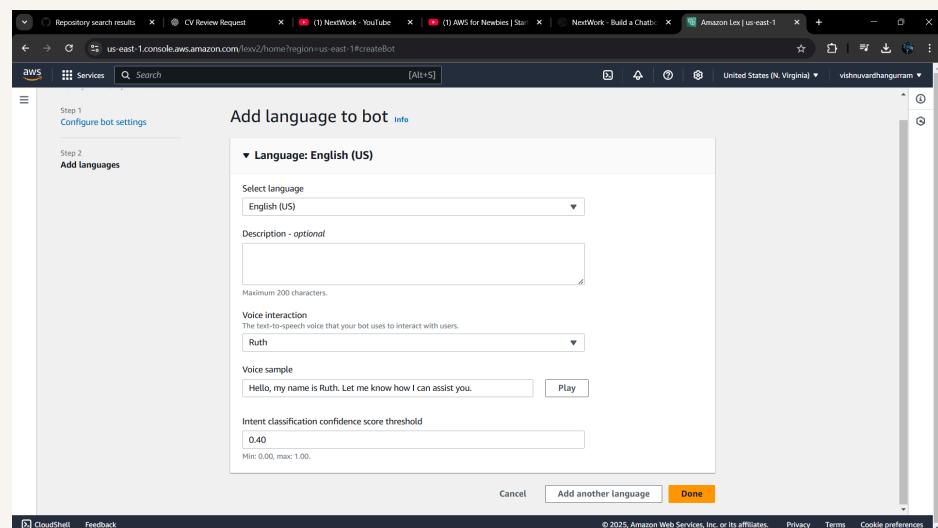


Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me 5 minutes to create the chatbot

While creating my chatbot, I also created a role with basic permissions because it should work as we specified.suppose like permision to access other resources , maximum time it should be idle for etc.

In terms of the intent classification confidence score, I kept the default value of 0.40. This means that the chat bot is 40 percent confident that i can understand the users chat and give response confidently.





Intents

An intent is what the user is trying to achieve in their conversation with the chatbot. For example, checking a bank account balance; booking a flight; ordering food.

'I created my first intent, WelcomeIntent , to Welcome a user when they say hello.

The screenshot shows the Amazon Lex console interface. The top navigation bar includes tabs for 'Editor', 'Visual builder', and 'New'. The main area displays the 'WelcomeIntent' configuration. It features a 'Preview' tab showing a conversation history with messages like 'Hi', 'Hello', 'I need help', and 'Can you help me?'. Below this, there is a text input field containing the message 'I want to book a flight' and a button labeled 'Add utterance'. A note below the input field states: 'You can provide messages to acknowledge the user's initial request. You can also configure next step in the conversation and branch based on conditions.' On the right side of the screen, a 'Test Draft version' window is open, showing a simulated conversation between a user ('hi') and the bot ('Hi I'm BB, the Banking Bot. How can I help you today?'). The bot's response 'help me' is followed by the user's message 'please'. A green checkmark indicates that the intent is 'Ready for complete testing'. At the bottom of the screen, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter hi,how are you,help me,please etc

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered... This error message occurred because Amazon Lex doesn't quite recognize your utterance

The screenshot shows the Amazon Lex console interface. On the left, there's a sidebar with 'WelcomeIntent' and 'FallbackIntent'. The main area displays a 'Test Draft version' window. It shows a conversation between a user and the bot. The user says 'hi', 'Hello', 'I need help', and 'Can you help me?'. The bot responds with 'Hi! I'm BB, the Banking Bot. How can I help you today?' and 'please'. Below this, a message box says 'Ready for complete testing' with a 'Type a message' input field. At the bottom, there are tabs for 'Editor', 'Visual builder', and 'New'.



Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when confidence score below 40% for all the intents you've defined.

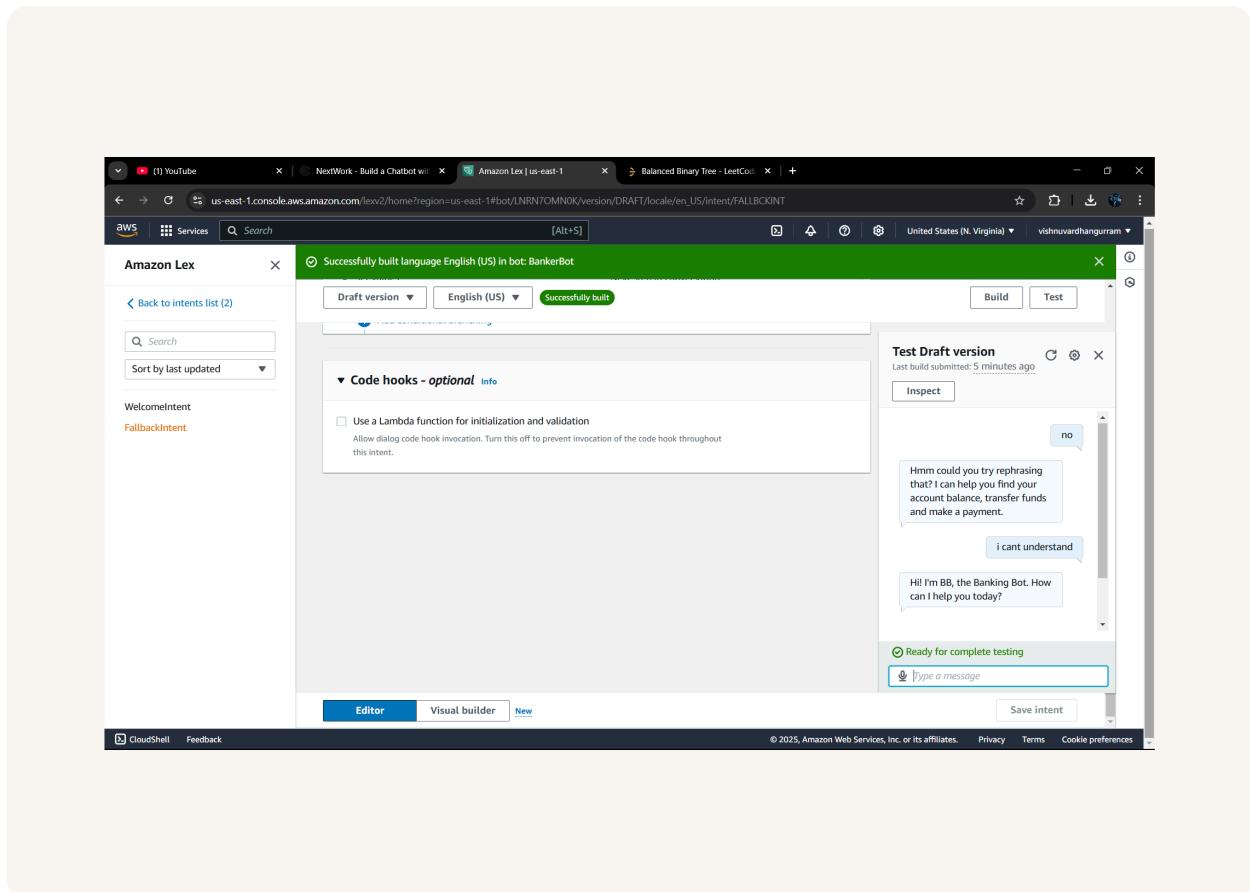
I wanted to configure FallbackIntent to customize error message that your chatbot will use to tell the user it doesn't understand their input.



Variations

To configure FallbackIntent, I Scroll down to Closing responses. Expand the arrow for Response sent to the user after the intent is fulfilled. and added the custom message

Variations are literally variations of the same Message in the main Message box. When Amazon Lex needs to return a Fallback response, it will randomly choose a message from the group and return that.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

