

NextWork.org

Build a Chatbot with Amazon Lex



jeraldarul1862004@gmail.com

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and various menu items like 'Services', 'CloudWatch Metrics', 'Logs', 'Metrics', 'CloudWatch Metrics Insights', and 'CloudWatch Metrics Insights'. Below the navigation bar, there's a breadcrumb trail: 'Lambda > Hello World' > 'Execution history'. On the left side, there's a sidebar with 'HelloWorld' selected. The main area displays the execution history for the 'HelloWorld' function. The first execution is highlighted with a green header that says 'Successfully built language English (US) in bot: bankerbot' and 'Draft version' with 'English (US)' selected. Below this, there are sections for 'Message group', 'Variations - optional', 'Set values', and 'Add conditional branching'. To the right of the message group section, there's a preview window showing a conversation between a user ('hiya') and a bot ('Hi! I'm BB, the Banking Bot. How can I help you today?'). The bot then responds with 'good morning'. A status message at the bottom right of the preview window says 'Intent FallbackIntent is fulfilled'. At the bottom of the main area, there are tabs for 'Editor', 'Visual builder', and 'New'. The bottom right corner of the main area has a button labeled 'Save intent'. At the very bottom of the page, there's a footer with links for 'CloudShell', 'Feedback', and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces using voice and text. It's powered by the same deep learning technologies as Alexa, making it ideal for creating chatbots, virtual assistants, and conversational applications.

How I used Amazon Lex in this project

i used amazon lex to create a banker bot with two intents one to welcome a user and another one is for fallback intent which also takes input in form of voice recognition

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

well i didnt expect creating a bot would be this easy

This project took me...

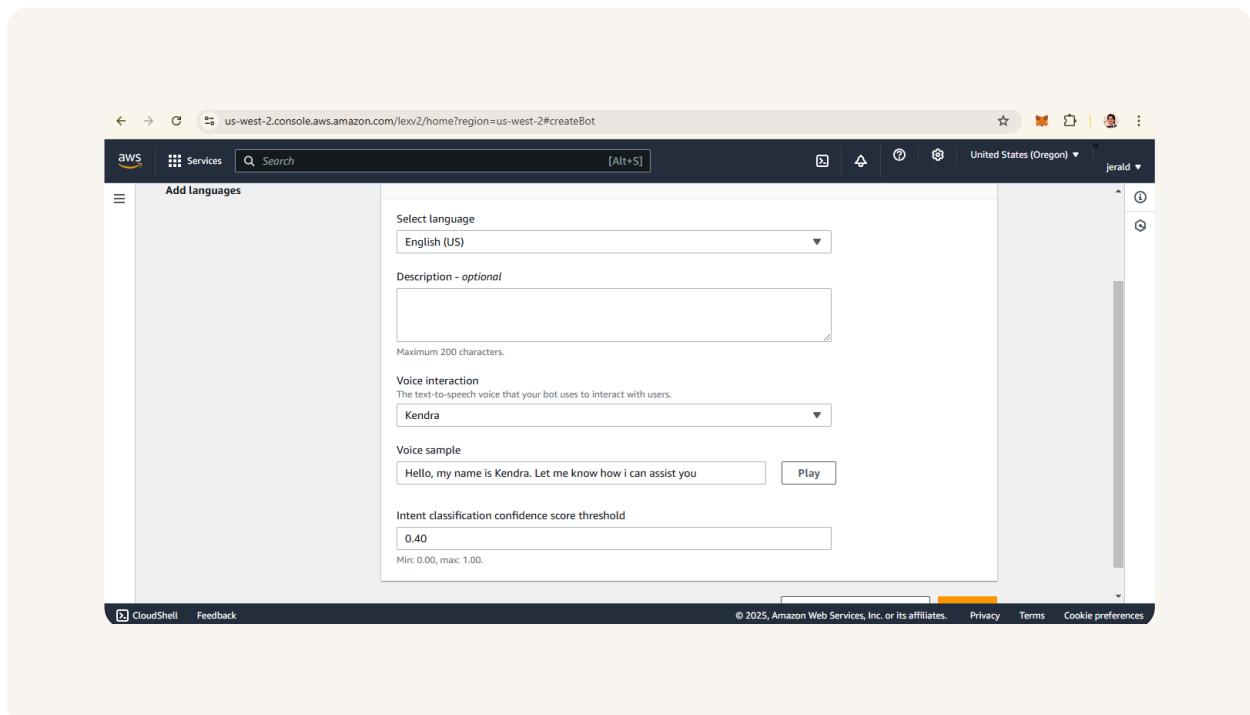
this project takes almost 1 hour because i played a little with all the possible inputs and created several welcome intent which is fun

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me about 2 minutes thanks to nextwork step by step documentation

While creating my chatbot, I also created a role with basic permissions because to allow other services like lambda to integrate with amazon lex

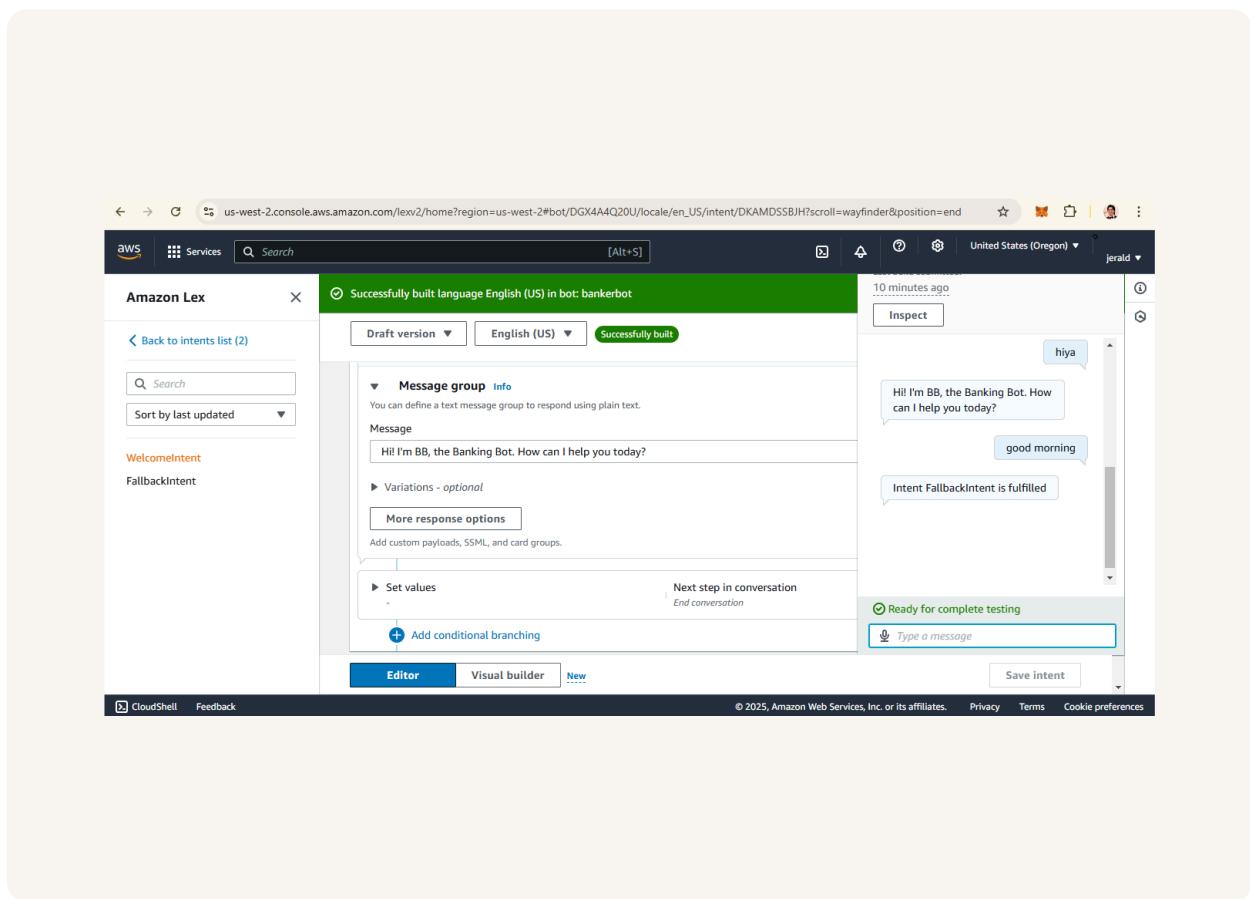
In terms of the intent classification confidence score, I kept the default value of 0.40. this threshold is like a minimum score for your chatbot to confidently understand what the user is trying to say.



Intents

intents are 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.

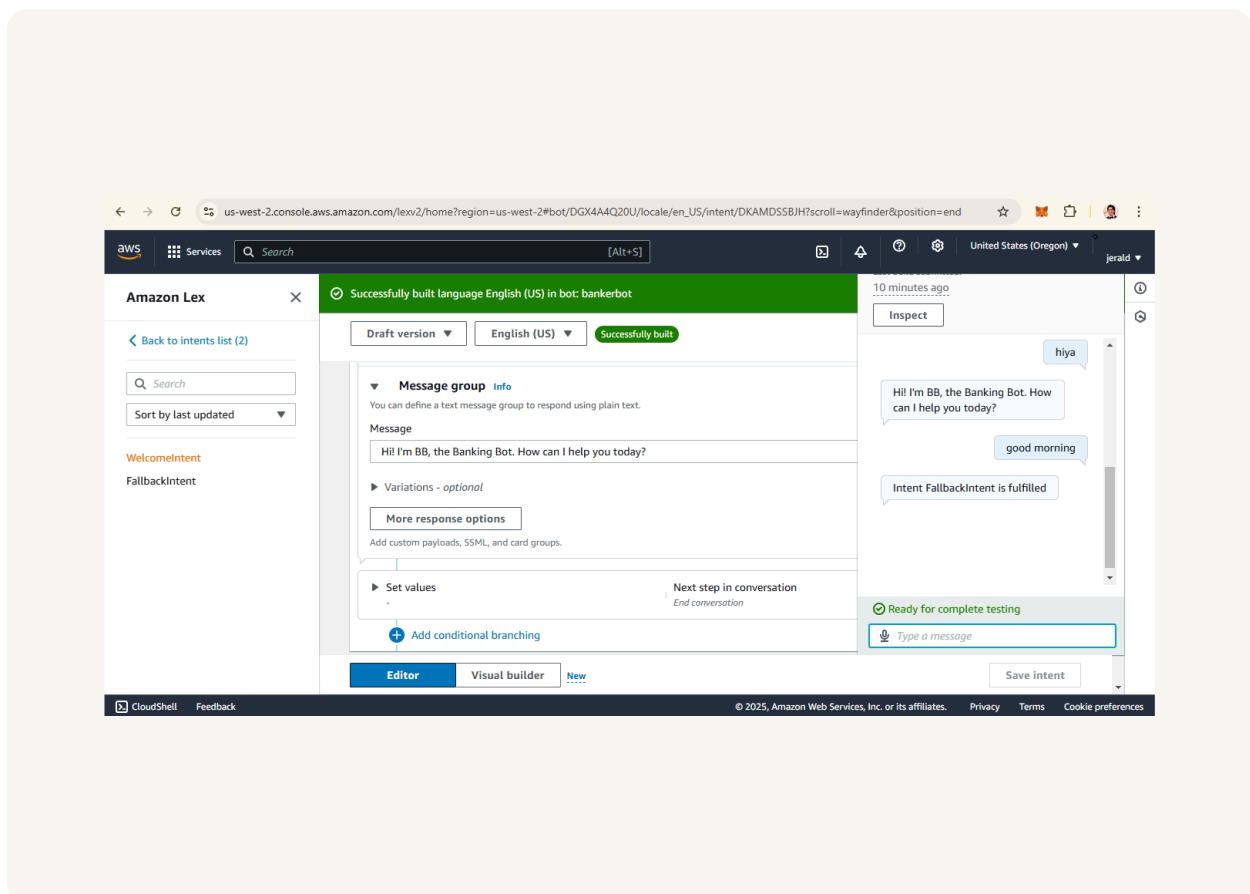
I created my first intent, WelcomeIntent, to Welcoming a user when they say hello.



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter Hi,hello,i need help,can you help me?

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered "good morning" This error message occurred because i didn't give this in the sample utterances so the bot won't understand



Configuring FallbackIntent

FallbackIntent is a default intent in every chatbot that gets triggered when the bot doesn't understand what you're saying think of it as a default error message

I wanted to configure FallbackIntent because i don't think the default message is very human friendly inorder to make the bot more conversational i gave several messages to make the user have a conversation with the bot

Variations

To configure FallbackIntent, I went to the fallbackintent section and inside the closing response added a message which will make the bot look more conversational

I also added variations! What this means for an end user is imagine getting the same kind of message everytime when the bot does not understand you inorder to make things fun by adding variations you'll be seeing different kind of messages

The screenshot shows the AWS Lambda console with the following details:

- Region:** United States (Oregon)
- Function Name:** lambda
- Last Execution:** 11 minutes ago
- Log Stream:** /aws/lambda/lambda-162004
- Logs:** The logs show multiple entries indicating failed API calls to 'https://api.sheety.co' with status codes 403 and 404.
- Code Summary:** The code summary indicates 10 lines of code and 1 file.
- Environment:** The environment variables listed include 'API_ID' and 'SHEETY_ID'.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

