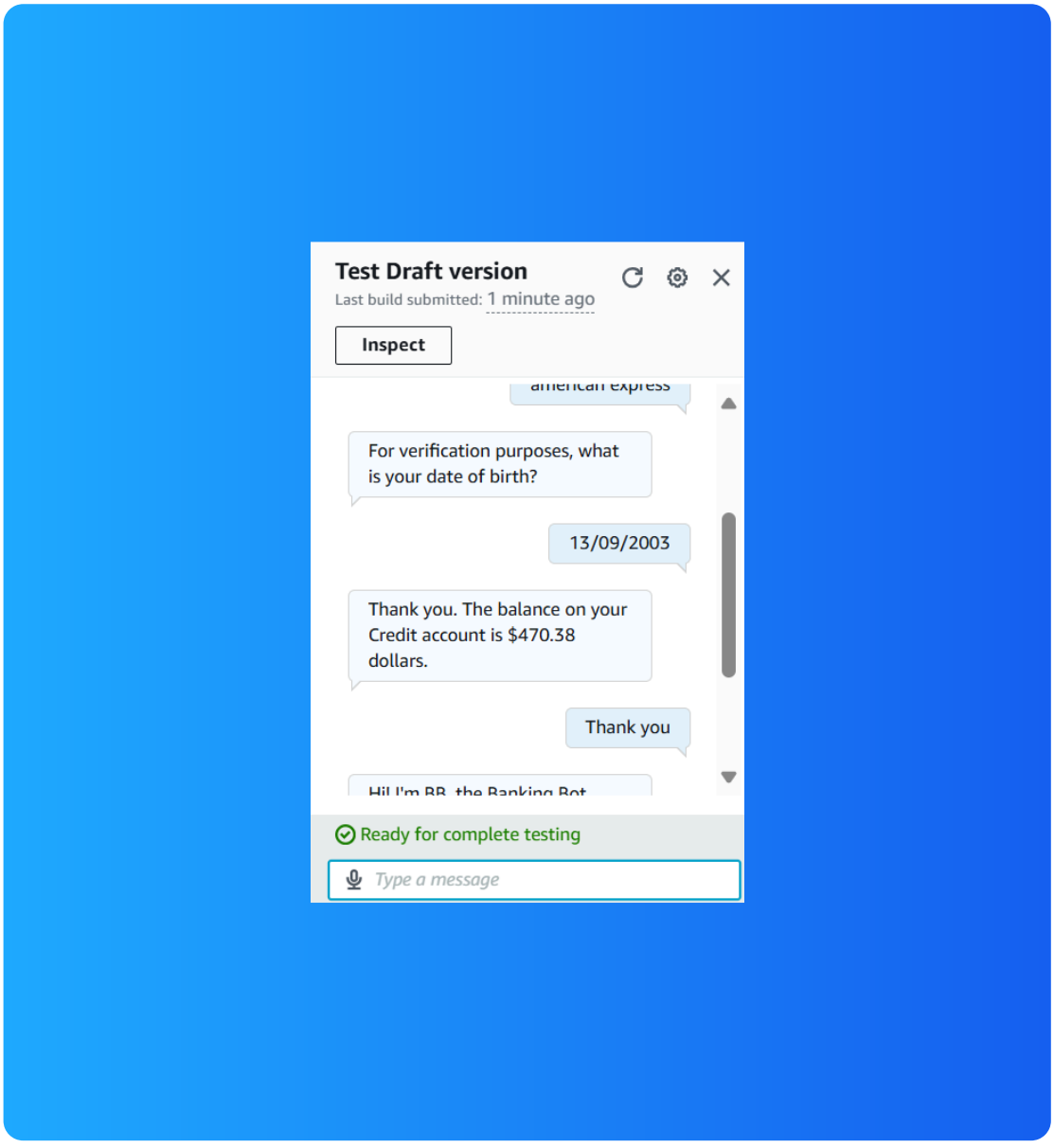
Connect a Chatbot with Lambda

Unmilan Mukherjee



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a chatbot service that is provided by Amazon. This can be linked with many other AWS services including AWS Lambda to enhance the experience across all the services.

How I used Amazon Lex in this project

I used Amazon Lex to ask a user for the account type and DOB and return a randomised account balance using a Lambda function and a random funky location. I also included some fallback intents for good measure.

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

I did not expect how easy it would be to link our AWS Lambda function to our Lex chatbot.

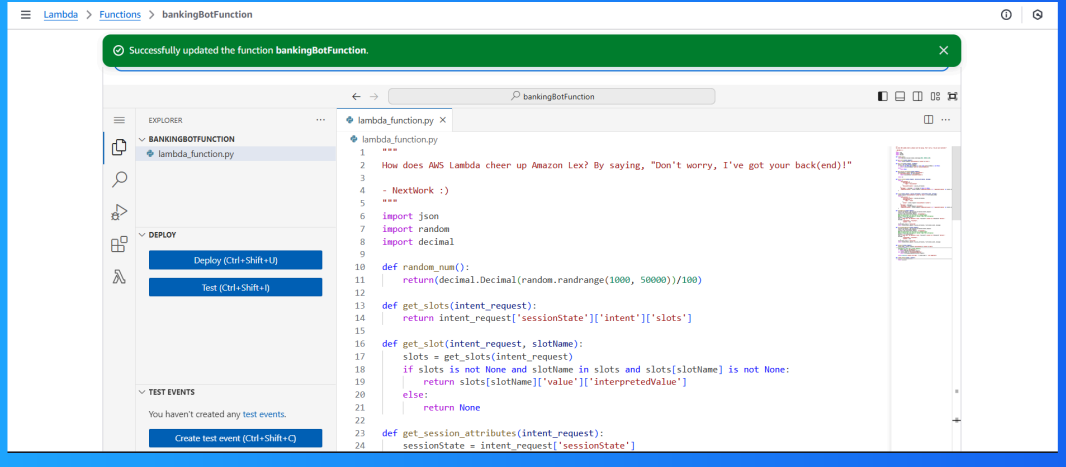
This project took me...

This project took me just under an hour.

AWS Lambda Functions

AWS Lambda is a service provided by Amazon that allows us to run some code without having to worry about maintaining a server or computer. It handles everything for us. It scales automatically from just a few requests a day to a thousand in seconds.

In this project, I created a Lambda function to take the user's account type from Lex and then use a random number generator function for testing purpose to generate a random account balance for the user and send it back to Lex.

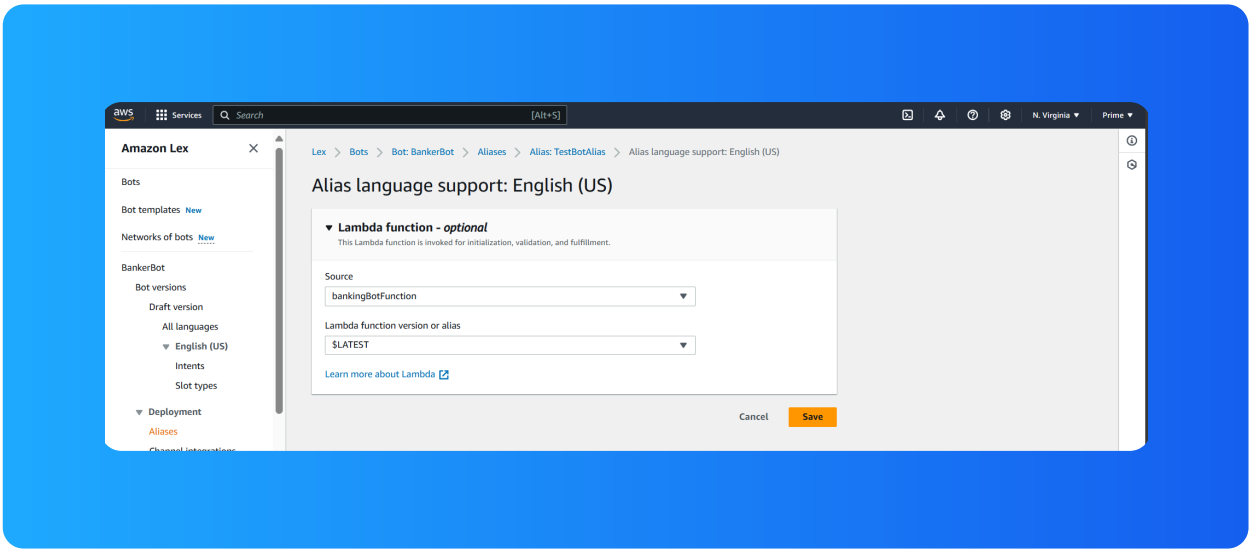


Chatbot Alias

An alias is a pointer that will allow other AWS services to connect to a specific version of our Lex chatbot. This saves developers a lot of time and reduces risk of errors. For our purpose I have put it to $LATEST.

TestBotAlias is a default Alias that AWS gives to our Lex chatbot that is intended to be used for test and development purposes. We can test the functionality of our bot in development using this alias.

To connect Lambda with my BankerBot, I visited my bot's TestBotAlias and clicked on the Language section, this opens the Lambda Function Panel where I can select the LambdaFunction I created earlier.

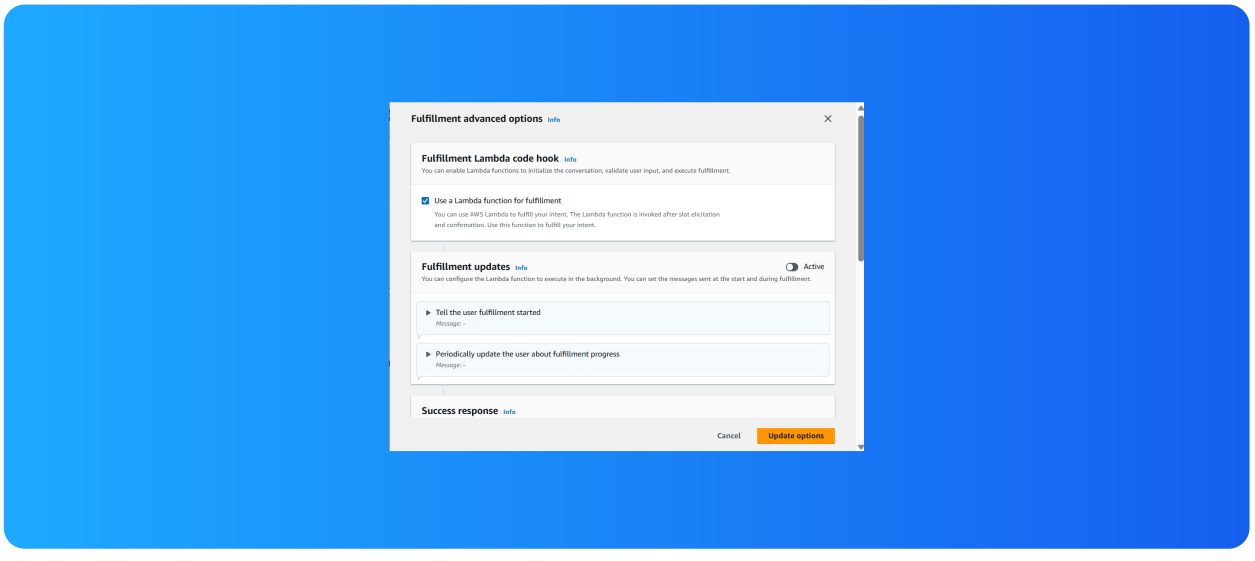


Code Hooks

A code hook connects our Lex chatbot to a specific Lambda function. This essentially makes our chatbot "smarter" as it can allow the bot to access backend or computative data which Lex cannot do on its own.

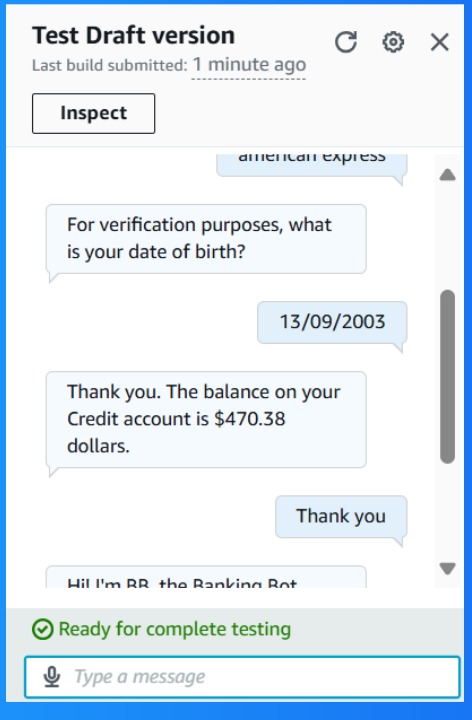
Even though I already connected my Lambda function with my chatbot's alias, I had to use code hooks because they enable dynamic responses and allow for more complex validation, logic, and processing of user inputs.

I could find code hooks at the very bottom of our intents configuration panel where we simply have to check a checkbox.



The final result!

I've set up my chatbot to trigger Lambda and return a random dollar figure when the user asks for their account balance after entering their account type and DOB.



Customizing the Lambda function

To level up the connection between Lambda and Lex, I added some funky different locations where our money could be and then made the code select any randomly. This customized message should be a more fun and engaging way to talk with our Lex chatbot.