



Connect Amazon Lex with Lambda



Gursimran Singh

The screenshot shows the Amazon Lex Test Draft version interface. At the top, there are 'Build' and 'Test' buttons. Below them, the title 'Test Draft version' and 'Last build submitted: Now' are displayed. There is an 'Inspect' button. The main area shows a conversation:

- User message: "For which account would you like your balance?"
- Bot response: "checking"
- User message: "For verification purposes, what is your date of birth?"
- Bot response: "10/8/1778"
- Bot message: "Thank you. The balance on your Checking account is \$474.55 dollars."

At the bottom, a green status bar says "Ready for complete testing" with a checkmark. A text input field says "Type a message" and a "Save intent" button is at the bottom right.



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service by AWS that helps you build chatbots and voice interfaces using natural language understanding. It is useful because it allows you to create smart, conversational bots that can interact with users naturally and efficiently.

How I used Amazon Lex in this project

In today's project, I used Amazon Lex to create a chatbot that responds to user greetings and balance inquiries. I set up intents, slots, and connected it with a Lambda function to return a random bank balance dynamically.

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

One thing I didn't expect in this project was how smoothly Amazon Lex integrates with Lambda. It was surprisingly easy to trigger backend logic and return custom responses without writing too much complex code.



Gursimran Singh
NextWork Student

nextwork.org

This project took me...

This project took me around 1 hour to complete. Most of the time went into setting up intents, linking Lambda, and testing the chatbot responses to make sure everything worked as expected.

AWS Lambda Functions

AWS Lambda is a serverless compute service that lets you run code in response to events without managing servers. It's great for connecting logic to services like Amazon Lex, so your chatbot can perform tasks like returning a bank balance.

My AWS Lambda function connects to my Amazon Lex bot and handles two intents: 'CheckBalance' and 'FollowupCheckBalance'. It generates a random balance for the user's account and sends it back to Lex, making the bot's response dynamic and realistic.

The screenshot shows the AWS Lambda Function Editor interface. The top bar displays the path 'Lambda > Functions > mybankerbotfunction' and a success message 'Successfully updated the function mybankerbotfunction.' Below the header is a code editor window titled 'lambda_function.py'. The code is as follows:

```
1  """
2  How does AWS Lambda cheer up Amazon Lex? By saying, "Don't worry, I've got your back(end)!"
3
4  ---
5
6  import json
7  import random
8  import decimal
9
10 def random_num():
11     return(decimal.Decimal(random.randrange(1000, 50000))/100)
12
13 def get_slots(intent_request):
14     return intent_request['sessionState']['intent']['slots']
15
16 def get_slot(intent_request, slotName):
17     slots = get_slots(intent_request)
18     if slots is not None and slotName in slots and slots[slotName] is not None:
19         return slots[slotName]['value']['interpretedValue']
20     else:
21         return None
22
23 def get_session_attributes(intent_request):
24     sessionState = intent_request['sessionState']
25     if 'sessionAttributes' in sessionState:
26         return sessionState['sessionAttributes']
27
```

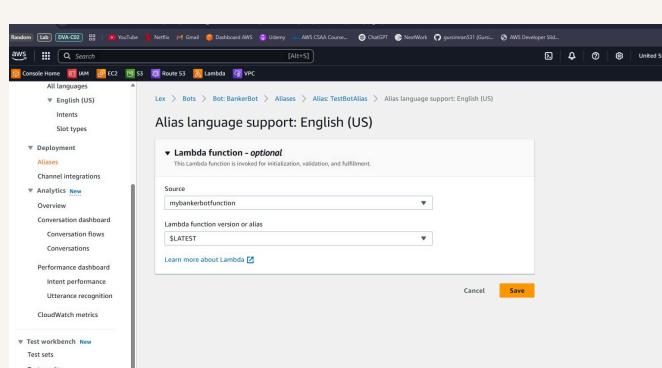
The code defines several functions: 'random_num' which returns a random decimal number between 1000 and 50000 divided by 100; 'get_slots' which returns the slots from the session state; 'get_slot' which returns the value for a specific slot; and 'get_session_attributes' which returns the session attributes. The code is annotated with comments explaining its purpose and how it interacts with the Lambda function and Amazon Lex.

Chatbot Alias

An Alias in Amazon Lex is like a version label for your bot. It lets you link different environments—like development, testing, or production—to specific versions of your bot, so you can manage updates without affecting live users.

TestBotAlias is the default alias Amazon Lex creates for testing your bot. It links to the latest draft version so you can try out changes in real time before publishing a final version for users.

To connect Lambda with my BankerBot, I visited my bot's TestBotAlias, went to the ***Language*** section, selected ***English***, and then chose my Lambda function to enable it for handling the backend logic of my bot's intents.

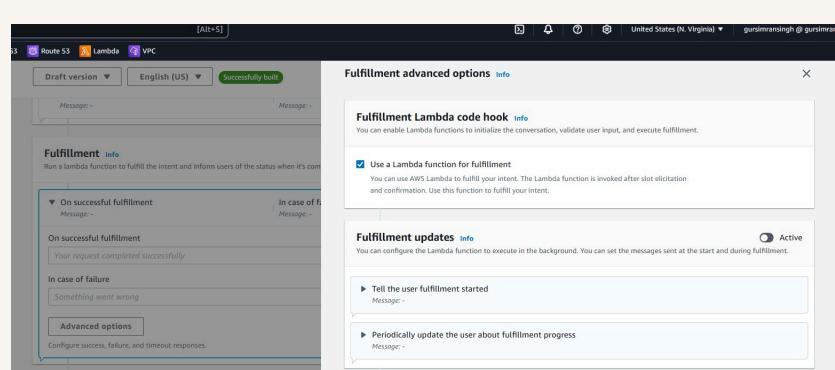


Code Hooks

A code hook is a way to connect your chatbot to backend code, like an AWS Lambda function. It lets the bot run custom logic—such as calculations or data lookups—before responding to the user.

Even though I already connected my Lambda function with my chatbot's alias, I had to use code hooks because they allow the chatbot to actually execute the Lambda code during the intent flow and return a dynamic, custom response like a bank balance.

I could find code hooks at the bottom of the intent editor in Amazon Lex, under the ***Code hooks*** section, where I enabled fulfillment using AWS Lambda for the CheckBalance intent.

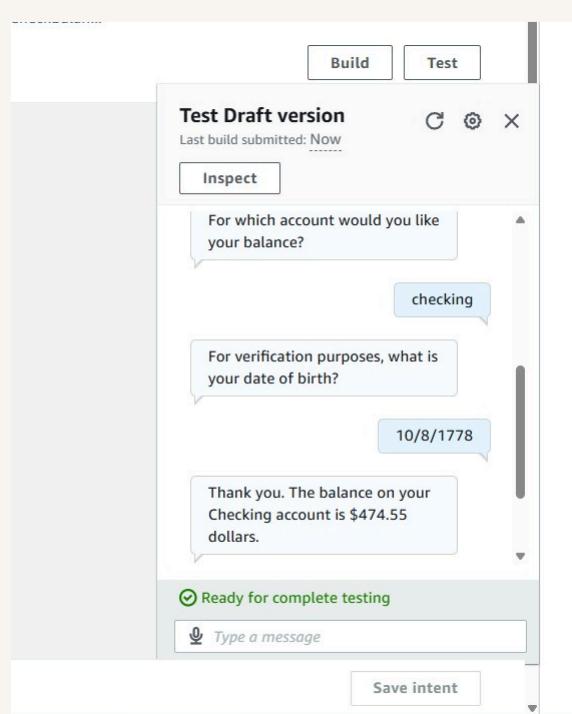


Gursimran Singh
NextWork Student

nextwork.org

The final result!

I've set up my chatbot to trigger Lambda and return a random dollar figure when a user activates the CheckBalance intent by asking about their account balance and selecting an account type.





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

