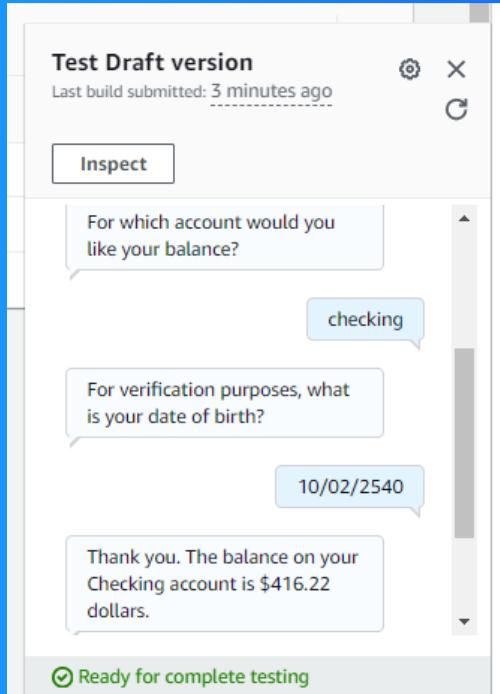




Connect a Chatbot with Lambda



sru anu





Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for creating conversational interfaces using voice and text. It's useful for building chatbots and virtual assistants that understand natural language and integrate seamlessly with other AWS services.

How I used Amazon Lex in this project

In today's project, I used Amazon Lex to build and configure a chatbot. I set up intents, slots, and responses to handle user queries, and integrated it with AWS Lambda for custom logic, enhancing the bot's functionality.

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

One thing I didn't expect in this project was the complexity of managing the `Alias` function. Handling different versions and ensuring the alias correctly pointed to the intended version for testing and production added an extra layer of configuration.

This project took me...

This project took me within an hour to complete, including setting up Amazon Lex, configuring intents and slots, and integrating it with Lambda for custom logic.

AWS Lambda Functions

AWS Lambda is a serverless computing service that lets you run code without provisioning or managing servers. It automatically scales and executes code in response to events, only charging for the compute time used.

In this project, I created a Lambda function to handle custom logic for my chatbot, such as processing user inputs, validating data, and returning dynamic responses based on specific conditions.

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The 'Code' tab is selected. Below the tabs is a toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and an Upload from dropdown. The main workspace displays a Python file named 'lambda_function.py'. The code is as follows:

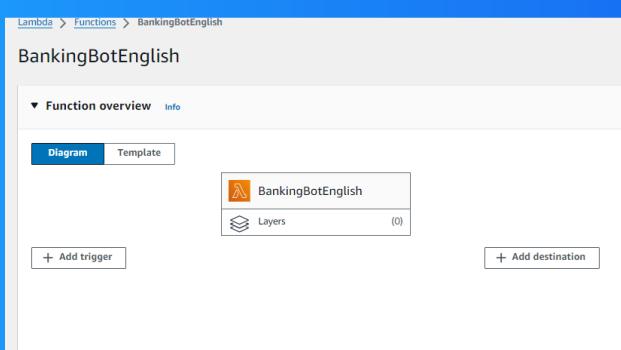
```
1 """
2 How does AWS Lambda cheer up Amazon Lex? By saying, "Don't worry, I've got your back(end)!"
3
4 Nextwork :
5
6 import json
7 import random
8 import decimal
9
10 def random_num():
11     return(decimal.Decimal(random.randrange(1000, 5000))/100)
12
13 def get_slots(intent_request):
14     return intent_request['sessionState']['intent']['slots']
15
16 def get_slot(intent_request, slotname):
17     slots = get_slots(intent_request)
18     if 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
28     return {}
29
30 def elicit_intent(intent_request, session_attributes, message):
31     return {
32         'sessionstate': {
33             'dialogAction': {
34                 'type': 'ElicitIntent',
35                 'message': message
36             }
37         }
38     }
```

Chatbot Alias

An alias is a pointer or reference to a specific version of an AWS resource, such as a Lambda function or Amazon Lex bot. It allows you to manage different versions and update them without changing the original resource name.

TestBotAlias is an alias that points to a specific version of a chatbot, typically used for testing purposes. It allows developers to test new configurations or versions of the bot without affecting the live production version.

To connect Lambda with my BankerBot, I visited my bot's TestBotAlias and specified the Lambda function under the fulfillment section. This allowed the bot to invoke the Lambda function for handling custom logic and generating dynamic responses.



sru anu
NextWork Student

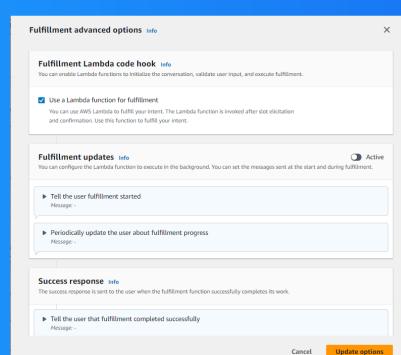
NextWork.org

Code Hooks

A code hook is a function, often written in AWS Lambda, that gets triggered during a chatbot's conversation to execute custom logic, such as validating user inputs or fetching data from external services before responding.

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 run custom logic at specific points in the conversation, such as validating user input or dynamically generating res

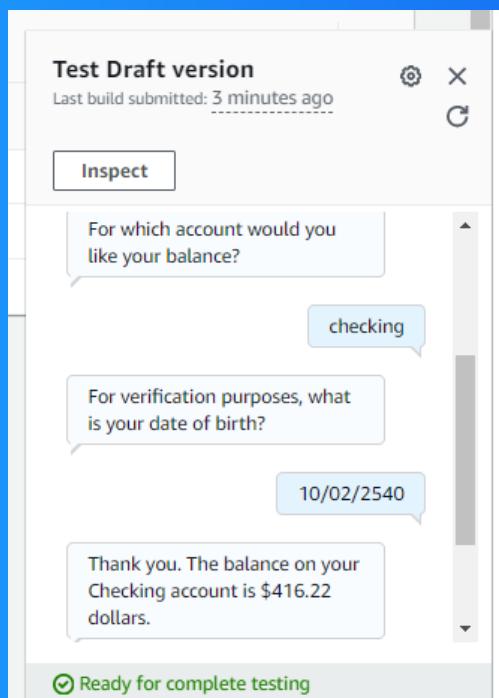
I could find code hooks at the fulfillment and validation stages of my chatbot's intents. They allow the bot to trigger custom logic when gathering user input or before completing an action, such as returning a bank balance.





The final result!

I've set up my chatbot to trigger Lambda and return a random dollar figure when the user requests their account balance and provides the necessary details, such as the account type, through the CheckBalance intent.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

