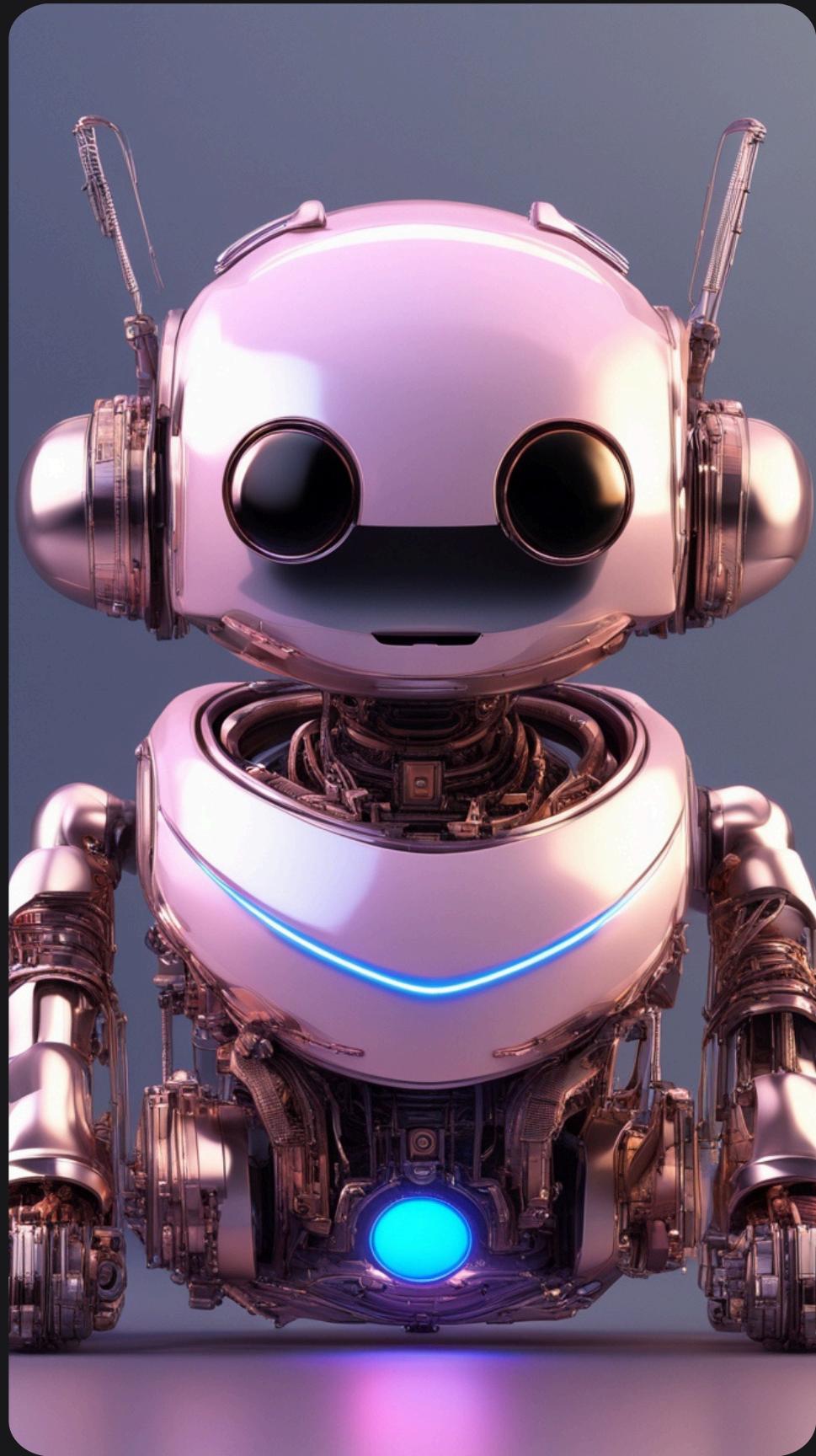
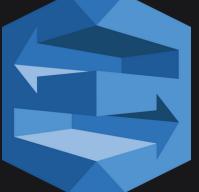


How I built a chatbot with Amazon Lex



Anitta Philip

 <https://www.linkedin.com/in/anittap/>



What is Amazon Lex?

What is Amazon Lex?

- Amazon Lex is a service for building conversational interfaces into applications using voice and text. It provides deep learning functionalities to recognize the intent of the text and respond appropriately.

What it does:

- The Amazon Lex chatbot, named "BankerBot," is designed to interact with users and provide banking-related services such as checking balances and transferring funds.

Why it's useful:

- The chatbot automates customer service tasks, reduces the need for human intervention, and provides quick and accurate responses to user queries.

How I'm using it in today's project:

- In this project, I used Amazon Lex to create a chatbot that can handle banking inquiries and transactions. The chatbot is integrated with AWS Lambda for enhanced functionality.



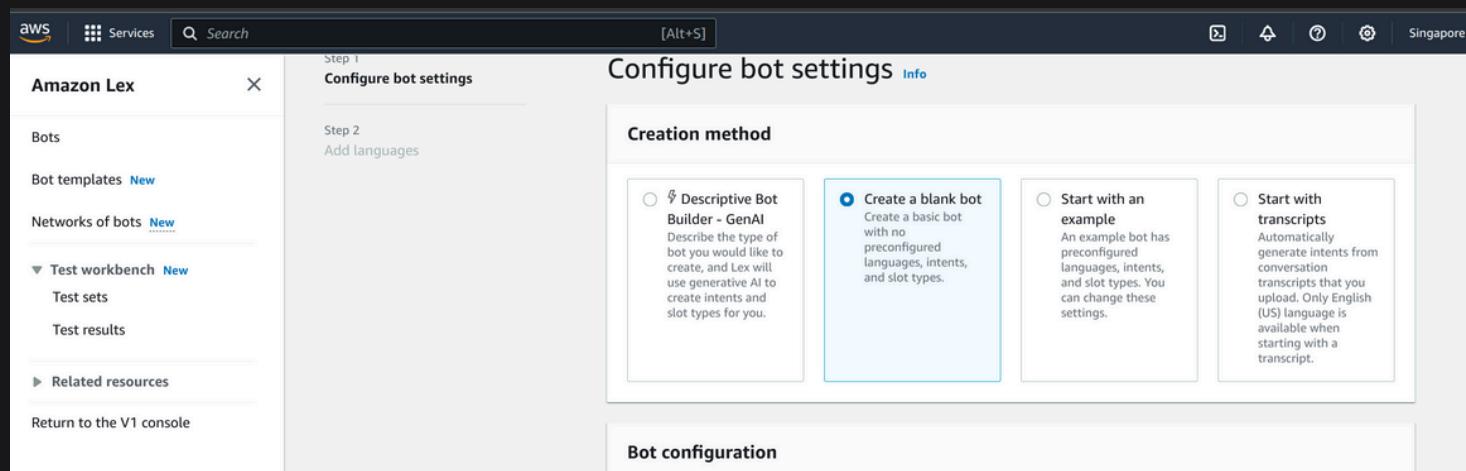
Anitta Philip

<https://www.linkedin.com/in/anittap/>

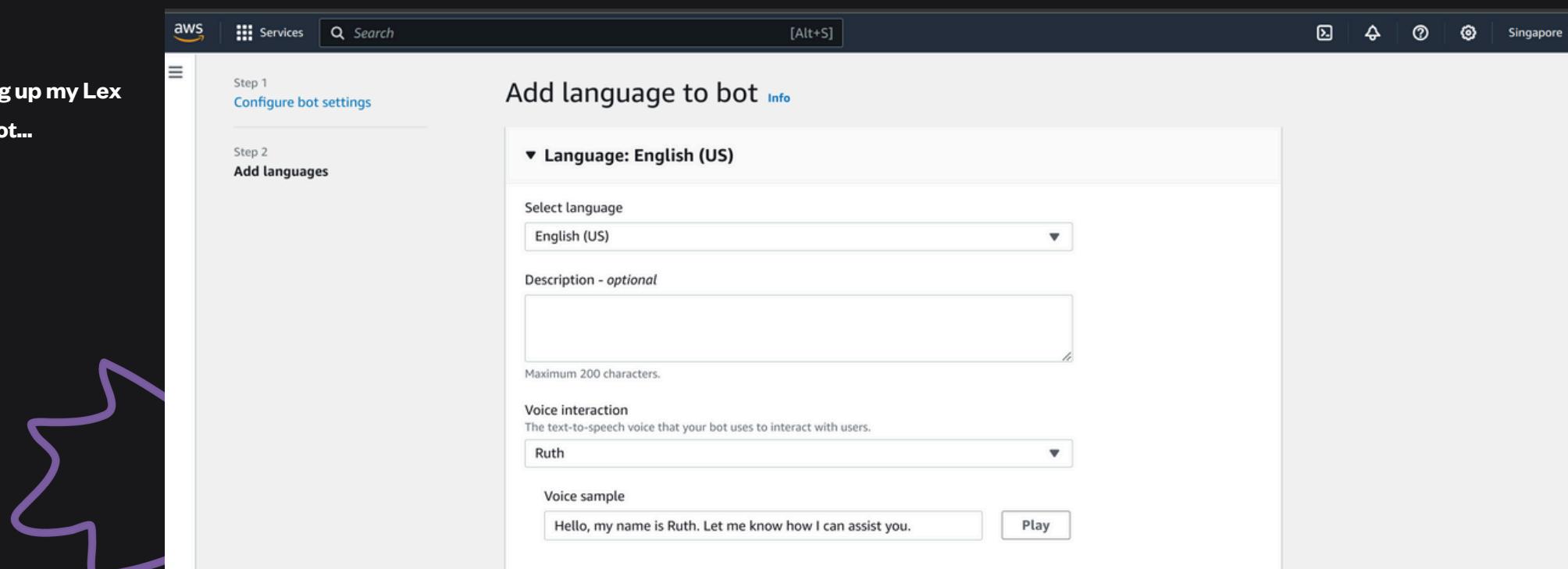


Set up a Lex chatbot

- I created BankerBot from scratch and used most default settings on Lex.



- The intent classification confidence score was kept at the default value of 0.40, which determines how confident the bot needs to be before responding to an intent.

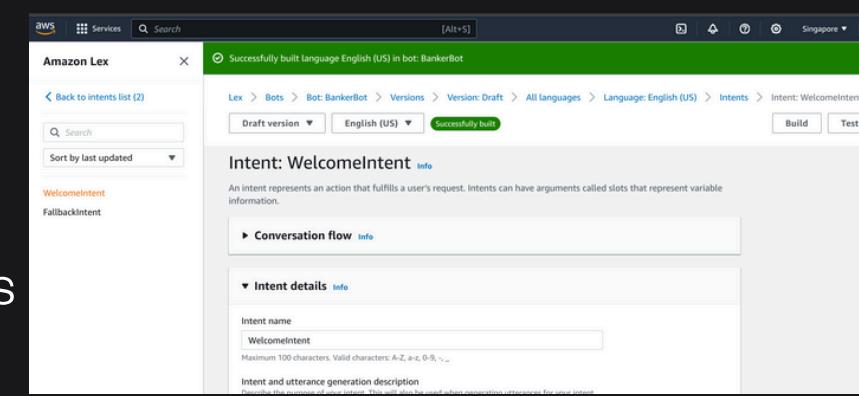


Anitta Philip

<https://www.linkedin.com/in/anittap/>

Create an intent in Lex

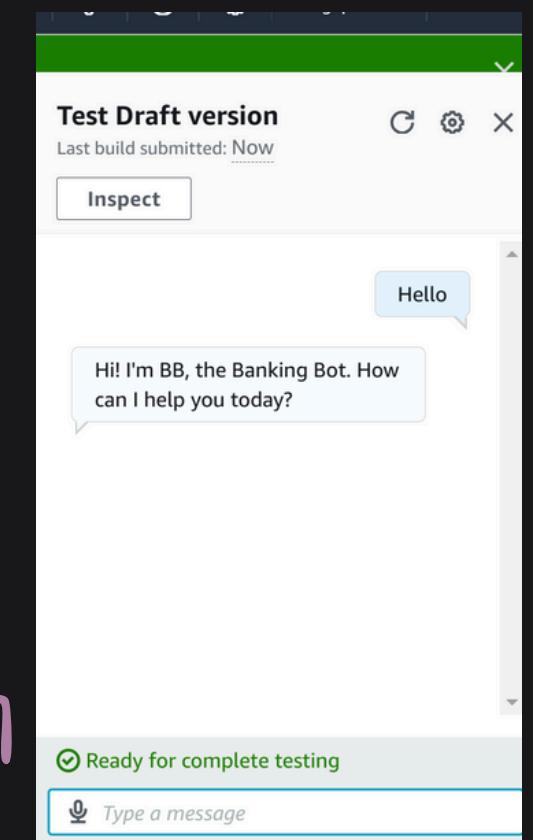
- Intents are predefined goals or actions that the chatbot is designed to fulfill.
- My first intent, "WelcomeIntent," was created to greet users and initiate interaction.



- **Steps to create "WelcomeIntent":**

1. Go to the Amazon Lex console.
2. Create a new intent named "WelcomeIntent."
3. Add sample utterances such as "Hello," "Hi," and "Good morning."
4. Save the intent and build the model.
5. Test the chatbot to ensure it responds correctly to the greetings.

My first test of the
chatbot



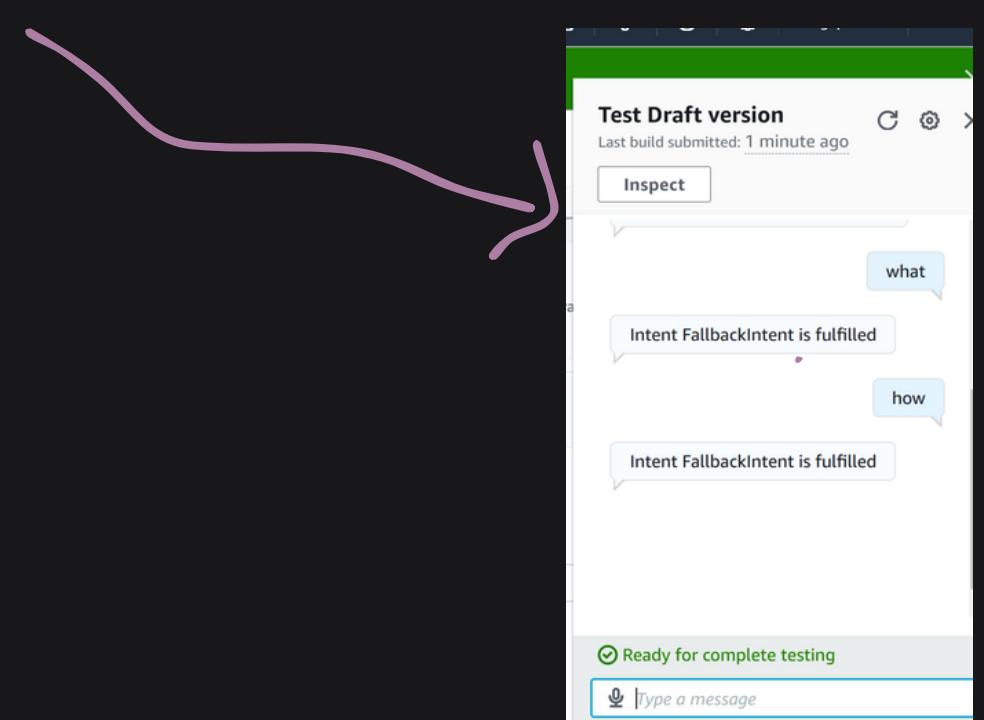
- **Error Handling:**

During testing, the chatbot returned the error message "Intent FallbackIntent is fulfilled" when certain inputs were entered. This error occurred because the input did not match any of the defined intents, triggering the FallbackIntent.



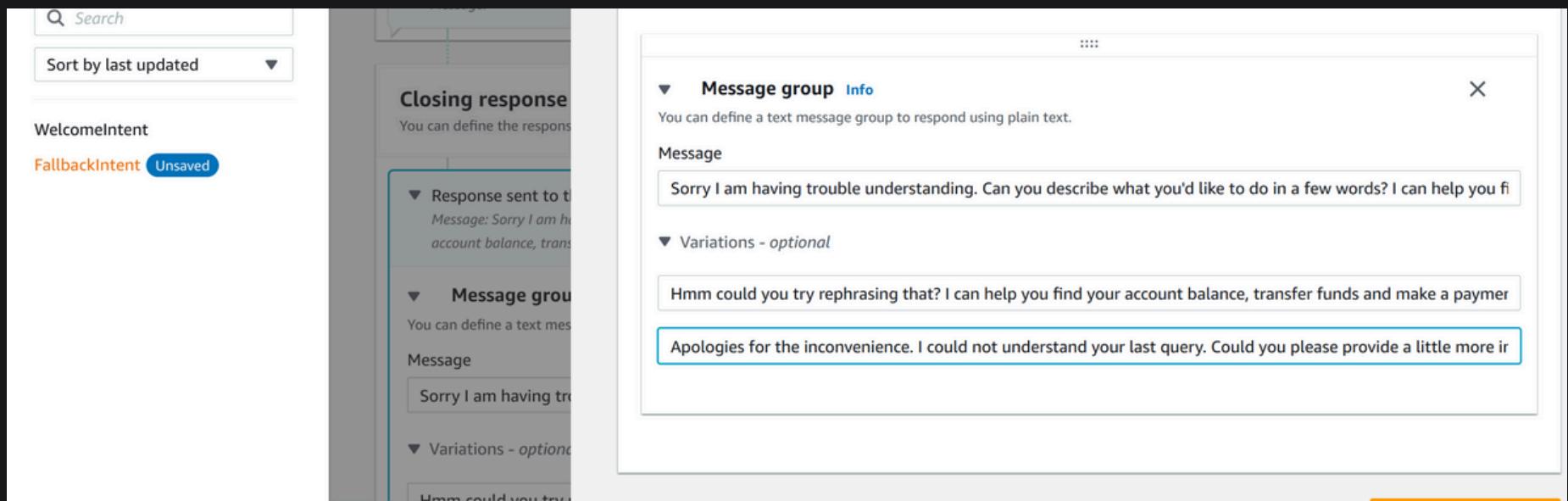
Anitta Philip

<https://www.linkedin.com/in/anittap/>

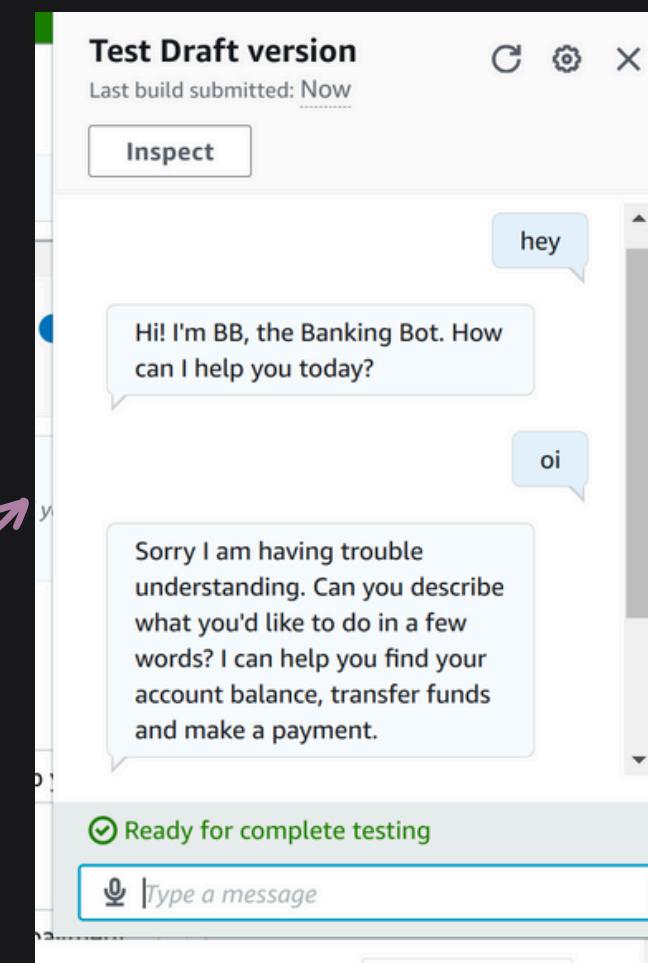


Manage FallbackIntent

- The FallbackIntent is a default intent that gets triggered when the chatbot cannot match an input to any existing intents.



- Steps to configure FallbackIntent:
 - Open the FallbackIntent in the Lex console.
 - Customize the response messages to provide clearer instructions to the user.
 - Add variations to the responses to improve user experience.

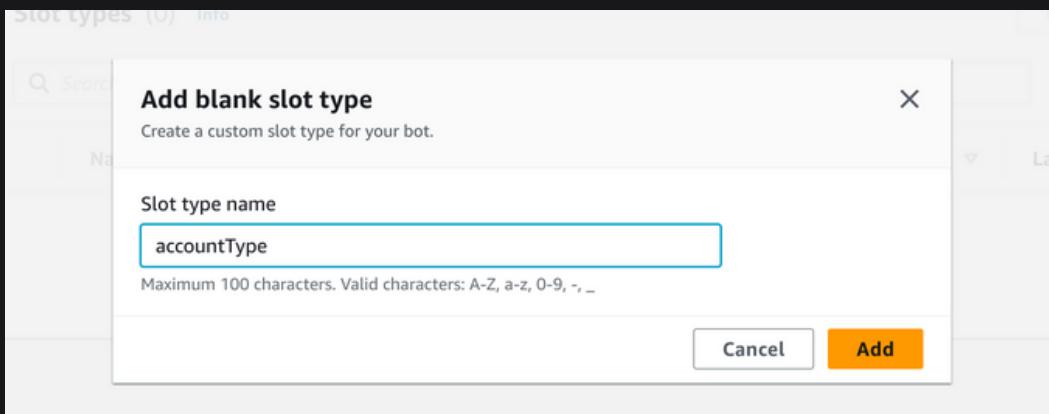


Anitta Philip

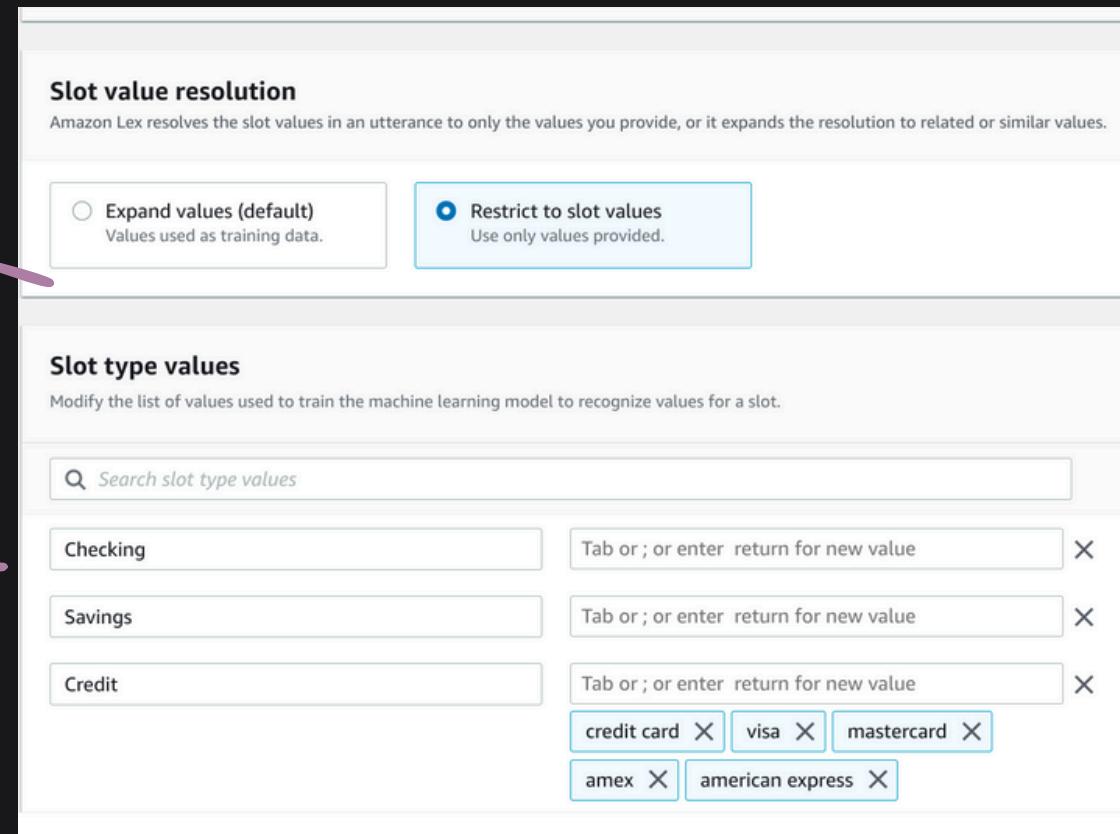
<https://www.linkedin.com/in/anittap/>

Create custom slots

- Slots are used to capture and store information provided by the user.



- Create a new slot type called "AccountType" with values like "savings," "checking," and "business."
- Associate the custom slot with the "CheckBalance" intent to capture the type of account the user wants to check.

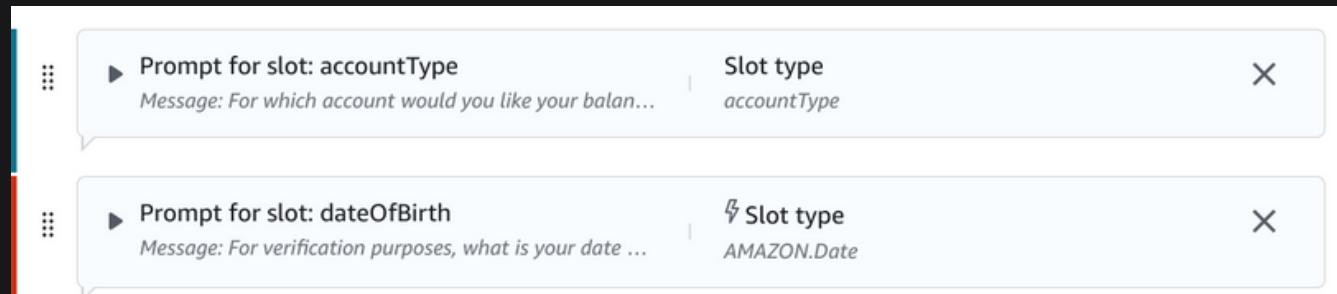
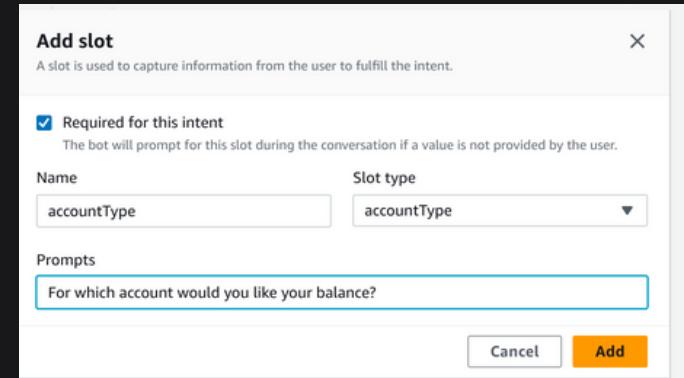


Anitta Philip

<https://www.linkedin.com/in/anittap/>

Simplifying the user experience

- Including slot values in the utterances can make the interaction smoother by reducing the number of questions the bot needs to ask.
- By adding the account type directly in the user's input, the chatbot can skip asking for it separately.



The screenshot shows the AWS Lambda Test console for a function named 'Test Draft version'. The 'Summary' tab is selected. On the left, the 'Intent' section shows 'CheckBalance'. In the 'Slots' section, 'accountType' is set to 'Savings' and 'dateOfBirth' is set to '2001-02-03'. The main pane displays a conversation: User says 'Hi! I'm BB, the Banking Bot. How can I help you today?', Bot replies 'I would like to check my savings account balance', User says 'For verification purposes, what is your date of birth?', Bot replies '02/03/2001'. Below the conversation, it says 'Intent CheckBalance is fulfilled' and 'Ready for complete testing'. A 'Type a message' input field is at the bottom.



Anitta Philip

<https://www.linkedin.com/in/anittap/>

Using AWS Lambda

- AWS Lambda is a service that allows you to run code without provisioning or managing servers.
- In this project, I created a Lambda function named "BankingBotEnglish NextWork.py" to handle backend processes that the chatbot alone cannot manage, such as accessing account balances.

A peek into the
Python code I
uploaded into
AWS Lambda!



The screenshot shows the AWS Lambda Code Editor interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs, there are sub-tabs for Code source and Info. On the left, there's a sidebar with File, Edit, Find, View, Go, Tools, Window, and a search bar labeled 'Go to Anything (Ctrl-P)'. The main area shows a file tree with a folder 'BankingBotEnglish' containing a file 'lambda_function.py'. The code editor displays the following Python code:

```
1 import json
2 import random
3 import decimal
4
5 def random_num():
6     return(decimal.Decimal(random.randrange(1000, 50000))/100)
7
8 def get_slots(intent_request):
9     return intent_request['sessionState']['intent']['slots']
10
11 def get_slot(intent_request, slotName):
12     slots = get_slots(intent_request)
13     if slots is not None and slotName in slots and slots[slotName] is not None:
14         return slots[slotName]['value']['interpretedValue']
15     else:
16         return None
17
```

At the bottom right of the editor, it says '5:18 Python Spaces: 4'.



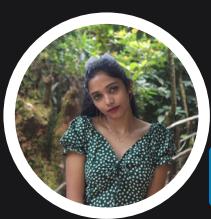
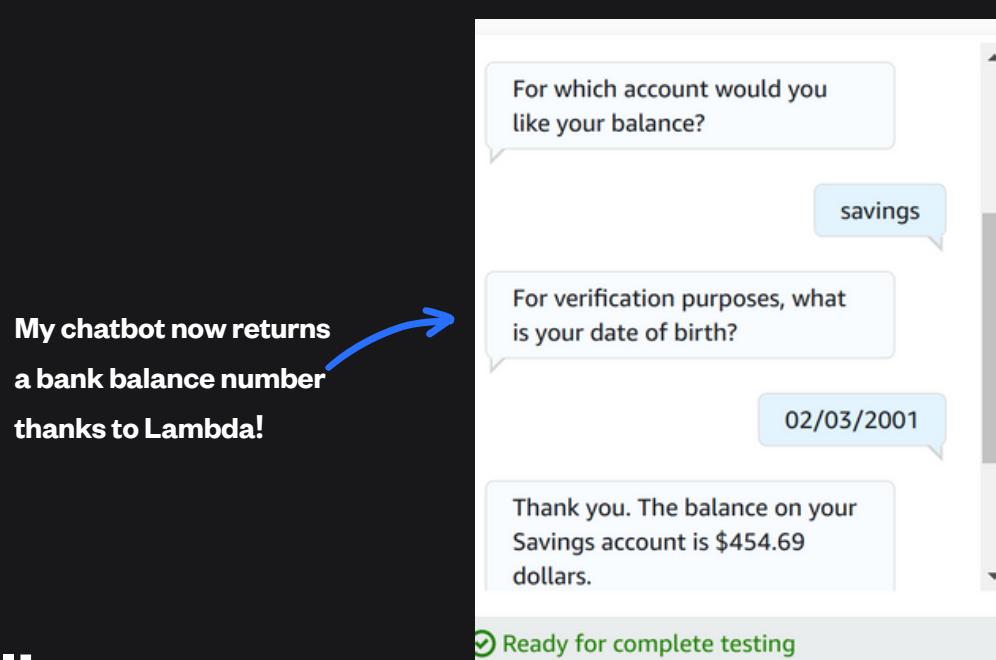
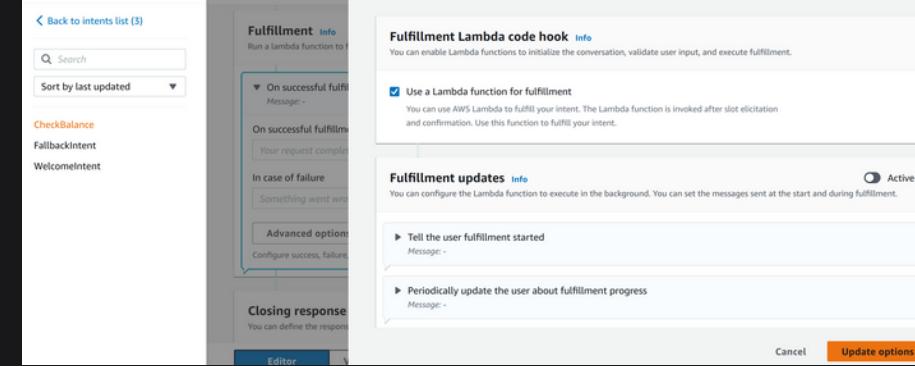
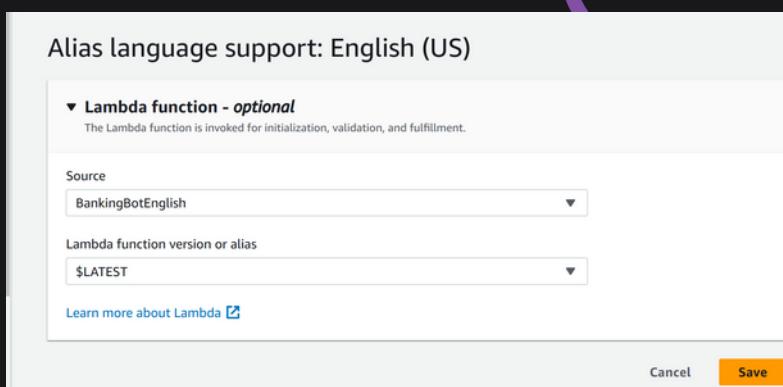
Anitta Philip

<https://www.linkedin.com/in/anittap/>

Connecting Lambda with Lex

To connect the Lambda function with the Lex chatbot, two main steps were required:

- Step 1:
 - a. Go to the Lex console and select the chatbot.
 - b. Navigate to the "Alias" section and associate the Lambda function with the chatbot alias.
- Step 2:
 - c. Configure code hooks in the intent settings.
 - d. Code hooks allow the chatbot to invoke the Lambda function during the conversation flow.



Anitta Philip

<https://www.linkedin.com/in/anittap/>

Context Tags

- Context tags are used to manage the conversation flow and maintain context between different intents.
- Types of context tags:
 - a. Output context tags: Set when an intent is fulfilled and passed to the next intent.
 - b. Input context tags: Required for an intent to be triggered.
- Example: I created an output context tag called "contextCheckBalance" in the "CheckBalance" intent.

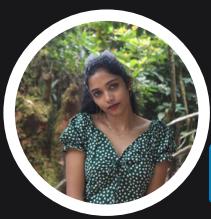
A look at output contexts

▼ Contexts - optional [Info](#)

Input contexts
[Choose contexts](#)

Output contexts
[Choose contexts](#)

contextCheckBalance X
Expires in 5 turns or 90s



Anitta Philip

 <https://www.linkedin.com/in/anittap/>

A Follow-Up Intent



- The "FollowupCheckBalance" intent was created to handle additional user queries after checking the balance.
- Relationship with "CheckBalance": This intent is triggered based on the context set by the "CheckBalance" intent, allowing the chatbot to maintain continuity in the conversation.

A look at input contexts

The screenshot shows the AWS Lambda function configuration interface for the 'FollowupCheckBalance' intent. On the left, a sidebar lists intents: 'FollowupCheckBalance' (highlighted in orange), 'CheckBalance', 'FallbackIntent', and 'WelcomeIntent'. The main panel displays the configuration for the 'FollowupCheckBalance' intent. It includes sections for 'Prompt for slot: dateOfBirth', 'Required for this intent' (checkbox checked), 'Name' (set to 'dateOfBirth'), 'Prompts' (message: 'For verification purposes, what is your date of birth?'), 'Confirmation' (helpful message: 'Prompts help to clarify whether the user wants to fulfill the intent or cancel the conversation.'), 'Set values' (button to 'Add conditional branching'), 'Dialog code hook' (info: 'You can enable Lambda functions to validate user input.', status: 'Active'), 'Lambda dialog code hook' (info: 'Invoke Lambda function: No'), and 'Default values - optional' (value: '#contextCheckBalance.dateOfBirth'). A note at the bottom says 'Provide a default value, #value for a context value, or [variable] for session variable.' with an example: 'San Diego, #ContextTag.SlotName, [SessionAttributeName]'.

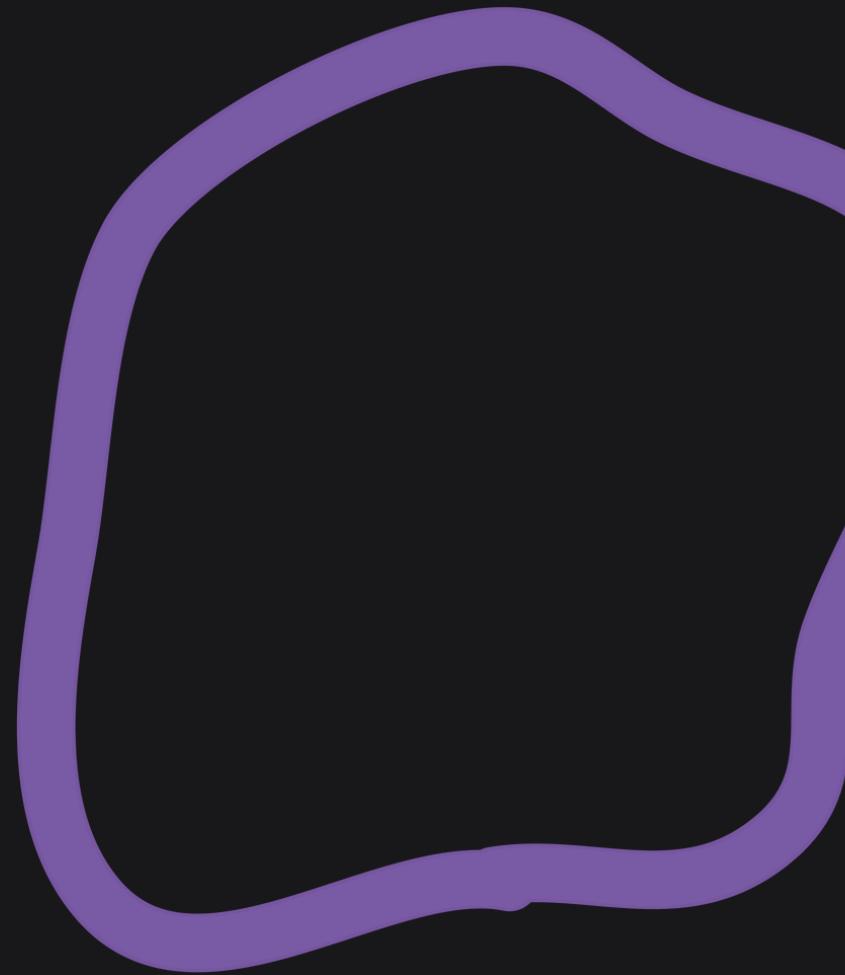


Anitta Philip

<https://www.linkedin.com/in/anittap/>

Context Tags in Action

- Conversation time! I built and tested my bot after creating the context tags and new intent.
- To see the context tags and the follow up in intent in action, I will have to trigger the intent by asking for another account balance the second time, to see if the bot will ask for the DOB again.
- Without setting the context, the bot would not understand the follow-up request.



Test Draft version C ⚙ X
Last build submitted: Now

Inspect

For verification purposes, what is your date of birth?

02/10/2021

My chatbot now carries over the user's date of birth to the next intent!

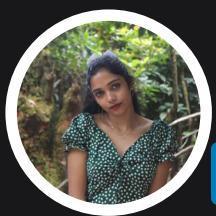
Thank you. The balance on your Savings account is \$136.41 dollars.

how about in credit

Thank you. The balance on your Credit account is \$474.82 dollars.

Ready for complete testing

Type a message

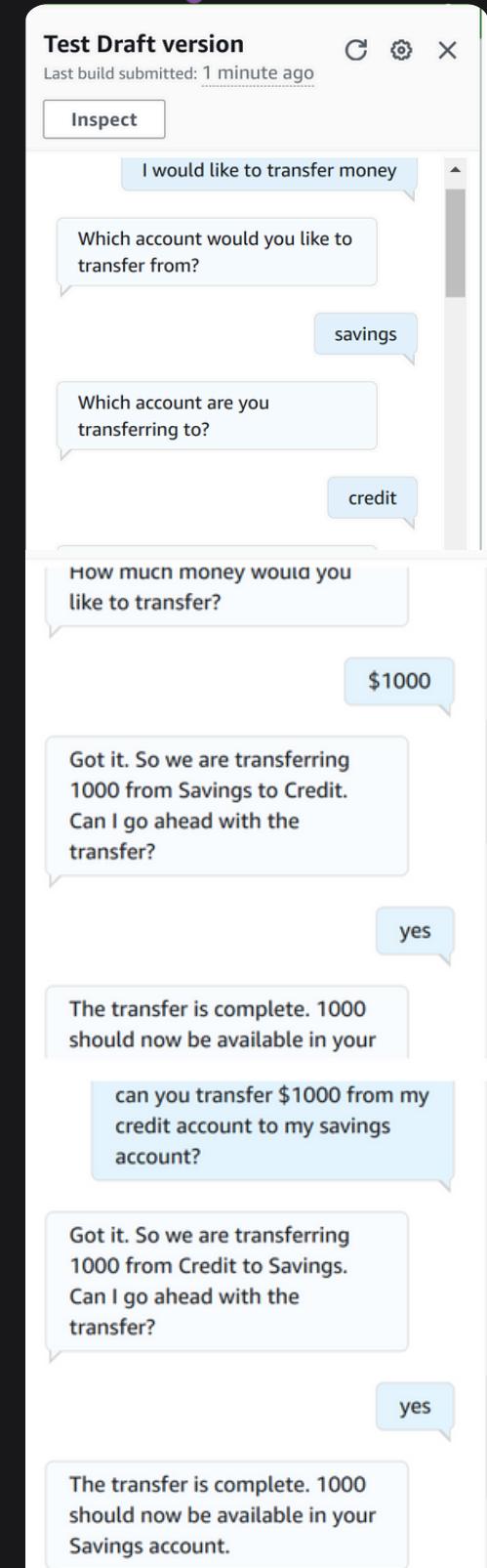


Anitta Philip

 <https://www.linkedin.com/in/anittap/>

More slots!

- Slots are used to capture and store information provided by the user.
- The final intent, "TransferFunds," uses multiple slot types to capture information about the transfer.
- For this intent, I had to use the same slot type twice. Multiple slots can have the same slot type, so it's super important that you're using clear slot names that make it easier to identify their differences.
- I also learned how to create confirmation prompts. Confirmation prompts are used to verify user input before performing actions.



A conversation demonstrating the two slots and the confirmation prompts in action!



Anitta Philip

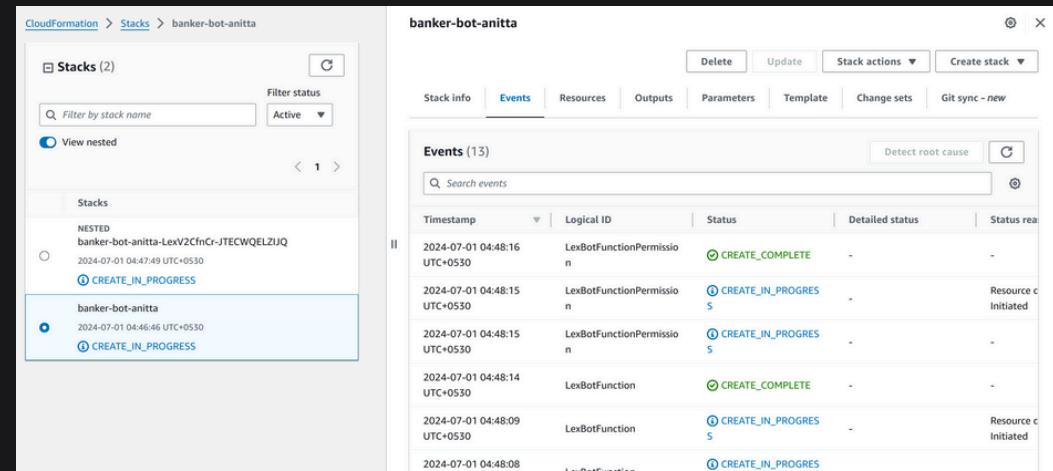
<https://www.linkedin.com/in/anittap/>

A LITTLE EXTRA...

Deploying with CloudFormation

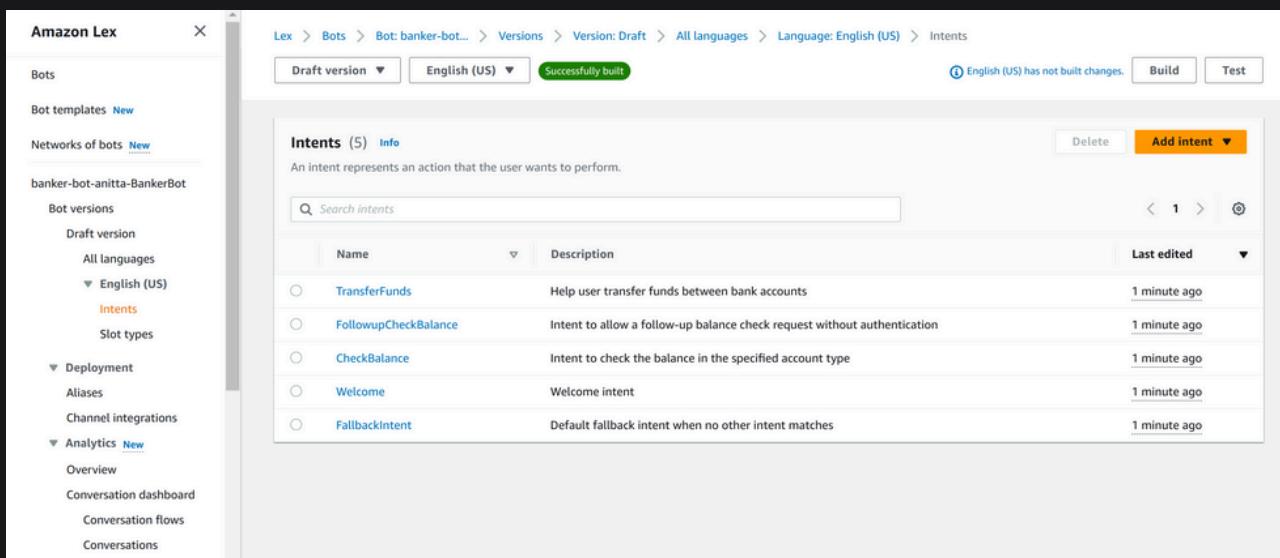
- AWS CloudFormation is a service that gives you an easy way to create and set up AWS resources. It is an infrastructure as code service - meaning you will create a CloudFormation template that describes all the resources you want to create and their dependencies as code.
- As an extension to this project, I learned how to deploy the entire BankerBot using a single CloudFormation stack.
- Doing this took me just around 10 minutes.
- Something I learned from deploying with CloudFormation was how easy and time-saving it is to just use cloudformation.

- We still have to do some troubleshooting after deploying with Cloudformation. But that's a story for another day. 



Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-01 04:48:16 UTC+0530	LexBotFunctionPermission	CREATE_COMPLETE	-	-
2024-07-01 04:48:15 UTC+0530	LexBotFunctionPermission	CREATE_IN_PROGRESS	-	Resource creation initiated
2024-07-01 04:48:15 UTC+0530	LexBotFunctionPermission	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:14 UTC+0530	LexBotFunction	CREATE_COMPLETE	-	-
2024-07-01 04:48:09 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	Resource creation initiated
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-
2024-07-01 04:48:08 UTC+0530	LexBotFunction	CREATE_IN_PROGRESS	-	-

CloudFront
deployed this
for me!



Name	Description	Last edited
TransferFunds	Help user transfer funds between bank accounts	1 minute ago
FollowupCheckBalance	Intent to allow a follow-up balance check request without authentication	1 minute ago
CheckBalance	Intent to check the balance in the specified account type	1 minute ago
Welcome	Welcome intent	1 minute ago
FallbackIntent	Default fallback intent when no other intent matches	1 minute ago



Anitta Philip

 <https://www.linkedin.com/in/anittap/>

My Key Learnings //

01

Amazon Lex: A service for building conversational interfaces using voice and text.

02

Intents: Predefined goals or actions that the chatbot fulfills.

03

AI/ML: Used in Lex to understand user input and respond appropriately.

04

FallbackIntent: A default intent triggered when no other intents match the input.

05

CloudFormation makes life easier 😊



Anitta Philip



<https://www.linkedin.com/in/anittap/>

Final thoughts...

- This project took me 2 hours to complete, and documentation took 1 more hours.
- **What's next?** In the next phase of this project, I'll be adding a new flow that lets users check their account balances and verify their identity with their birthday. I'll be creating a custom slot type to handle the different bank account types. Excited to bring this feature to life and make our BankerBot smarter and more interactive! 💡



Anitta Philip

 <https://www.linkedin.com/in/anittap/>

ERROR! 00

OMG!!!

An error I ran into was...

- Could not trigger "FollowupCheckBalance" intent after following all steps.
-  I noticed that I had accidentally added the context "contextCheckBalance" in the output context for this intent. I solved this error by changing it to an input context.



Anitta Philip

 <https://www.linkedin.com/in/anittap/>