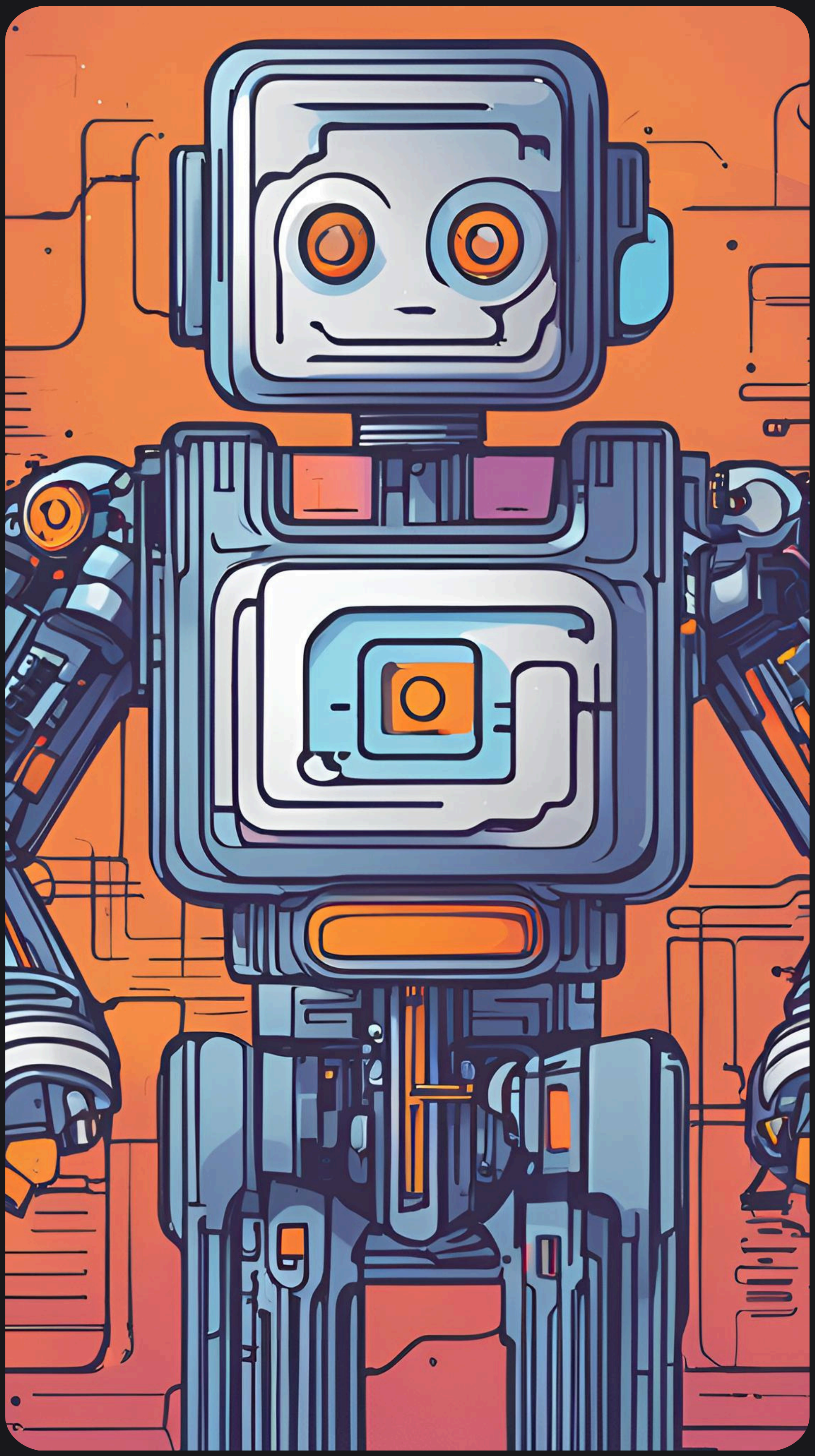
How I built a chatbot with Amazon Lex in 5 parts



AVRS Anand

in linkedin.com/in/avrsanand/

What is Amazon Lex?

What it does:

- Amazon Lex is an AWS service that enables the building of conversational chatbots into any application using voice and text.

Why it's useful:

- Its useful as it makes it easy to have a custom chatbot to interact with users, while also automating tedious repetitive simple tasks which would normally requirement human input.

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

- In this project I'm using Amazon Lex to create BankerBot, a chatbot that will ultimately help users find their account balance, transfer funds, and make payments

This project took me ...

- Around 2 hrs to complete as a total beginner, if you consider to document the process it may add another 2 hrs.

AVRS Anand

linkedin.com/in/avrsanand/

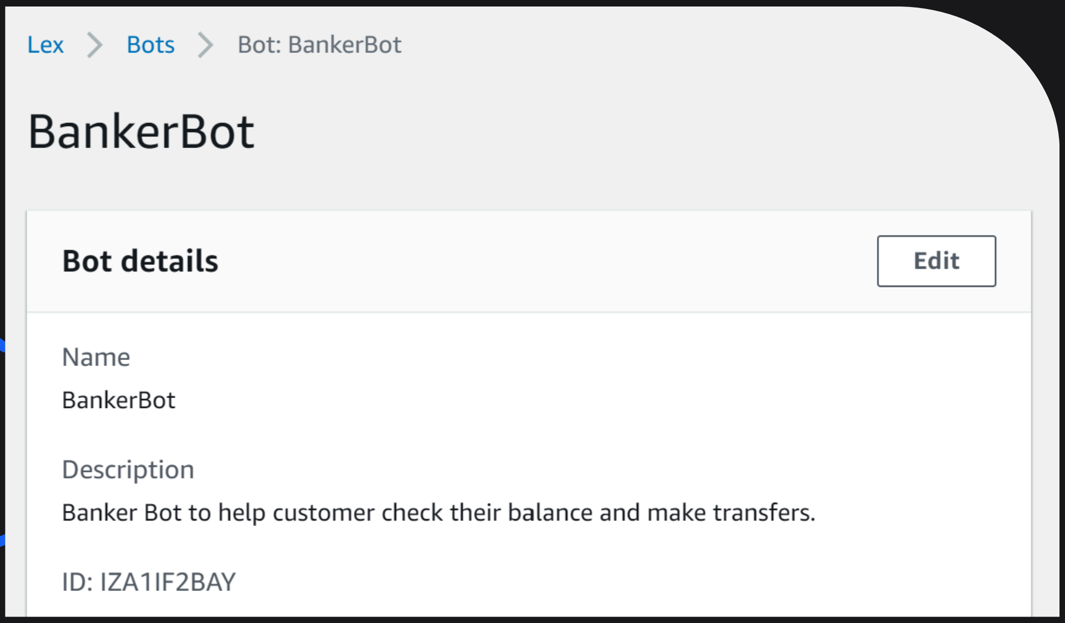
PART 1

Set up a Lex chatbot

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

- In terms of the **intent classification confidence score,** I kept the default value of 0.40. What this means for my chatbot is that my chatbot needs to be atleast 40% sure that it understands what the user is saying before it generates a response. This allows for a bit of ambiguity in prompts.

Setting up my Lex chatbot...



BankerBot

Bot details

Name
BankerBot

Description
Banker Bot to help customer check their balance and make transfers.

ID: IZA1IF2BAY

AVRS Anand

linkedin.com/in/avrsanand/

Create an intent in Lex

- Intents, like the title suggests, outline what a user is trying to achieve.

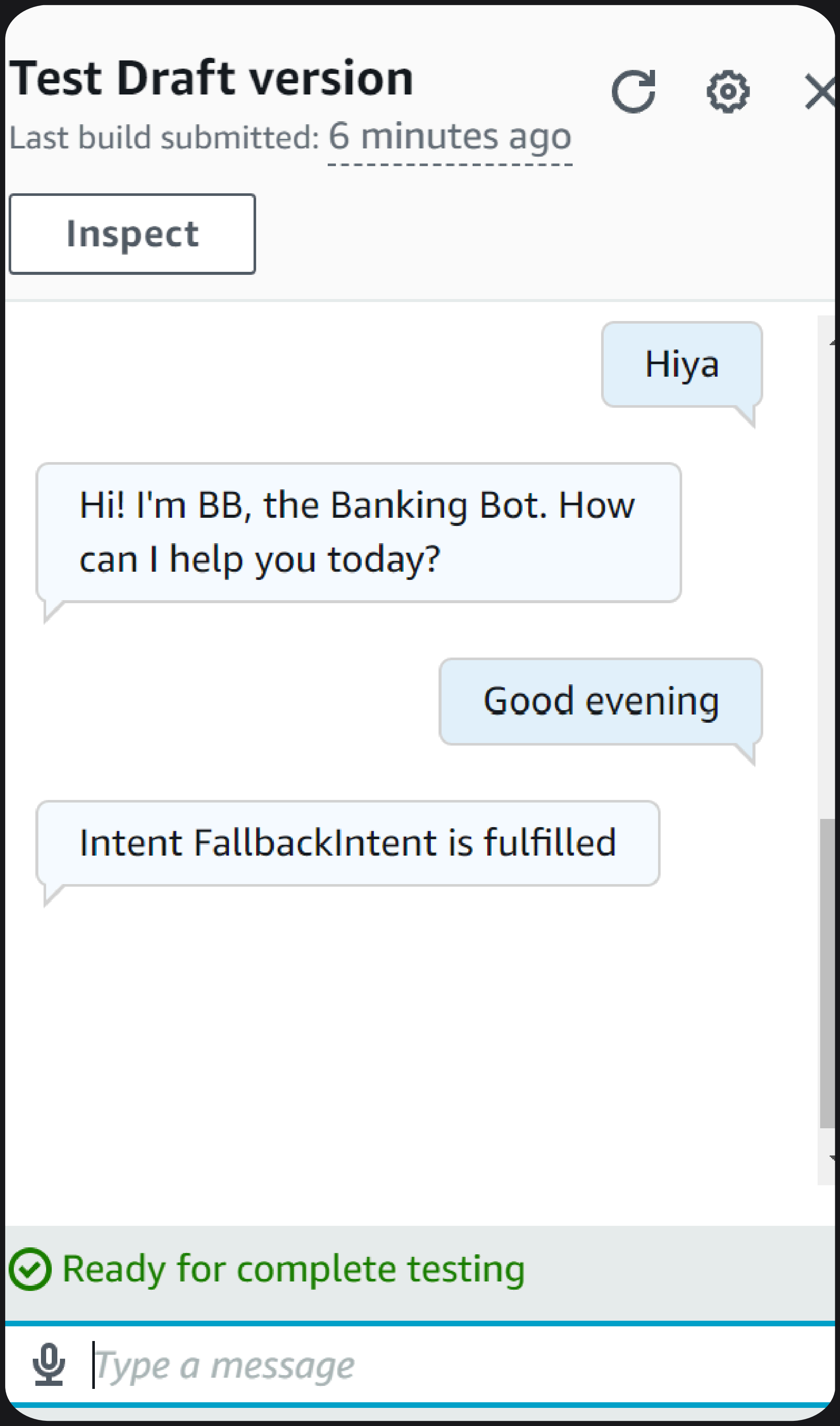
- My first intent, WelcomeIntent, was created to welcome users who are greeting or asking for help

- To set up this intent, I gave it the name WelcomeIntent, gave the intent a clear description, added statements as sample utterances, and added a closing response, which will be outputted if the user input matches any of the sample utterances.

- I launched and tested the chatbot, which could still respond if I enter messages which imply hi or suggest the need for help.

- However, the chatbot returned the error message “Intent FallbackIntent is fulfilled” when I entered messages my chatbot could not match to the utterance such as “Good evening”. This error message occurred because the chatbot was less than 40% confident it could match the input to an utteranace.

My first test of the chatbot



Test Draft version

Last build submitted: 6 minutes ago

Hiya

Hi! I'm BB, the Banking Bot. How can I help you today?

Good evening

Intent FallbackIntent is fulfilled

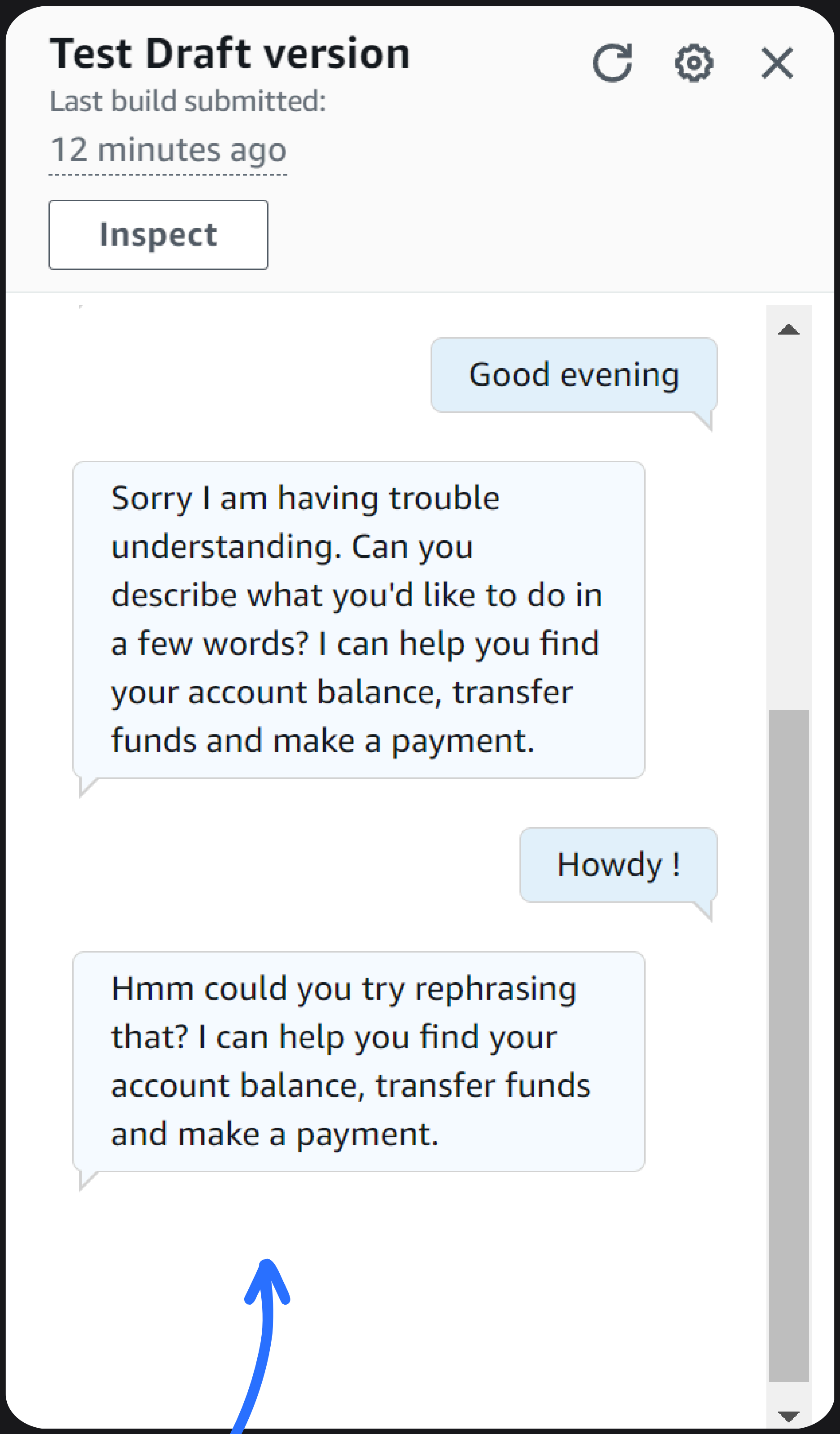
Ready for complete testing

AVRS Anand

linkedin.com/in/avrsanand/

Manage FallbackIntent

- FallbackIntent is a default intent in every chatbot that gets triggered when the input intent classification confidence score threshold is not met. Meaning the bot is not confident that the input can be matched to an utterance.
- I wanted to configure FallbackIntent because I wanted to give the user a clear message when the bot did not understand their input.
- To configure FallbackIntent, I had to select FallbackIntent, and enter a closing response message which will be displayed if the bot doesn't understand the user input.
- I also added variations! What this means for an end user is that the end user will receive multiple different responses instead of just one generic message, making the interaction a bit more conversational.



Test Draft version

Last build submitted:

12 minutes ago

Good evening

Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment.

Howdy !

Hmm could you try rephrasing that? I can help you find your account balance, transfer funds and make a payment.

Perfect, the error message is now much clearer, and there are variations too

AVRS Anand

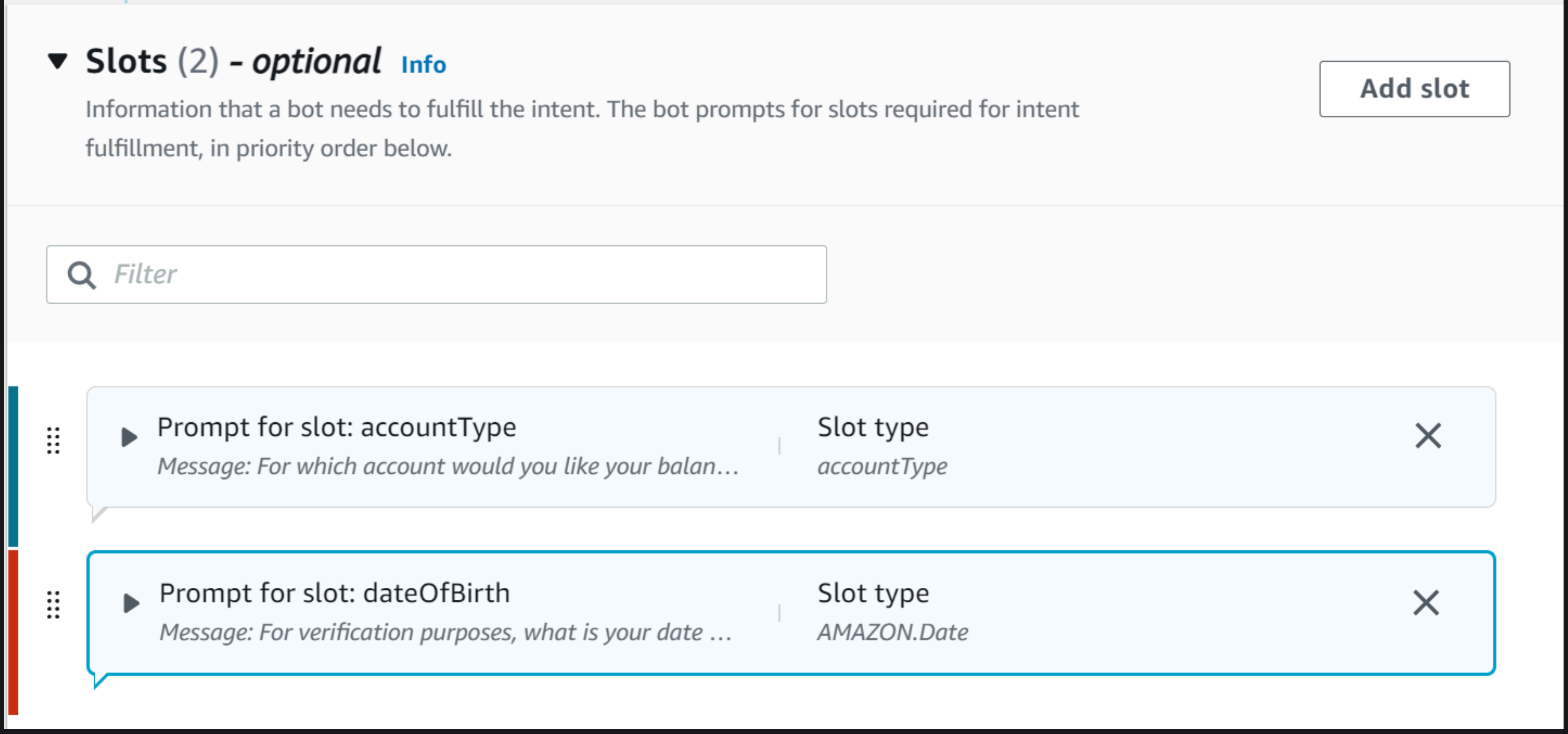
linkedin.com/in/avrsanand/

PART 2

Create custom slots

- **Slots** are pieces of information that a chatbot needs to complete a user's request.
- In this project, I created a custom slot to represent different bank account types: Checking, Credit and Savings and another slot to verify date of birth for verification of the user.

My custom slots



Slots (2) - *optional*

Information that a bot needs to fulfill the intent. The bot prompts for slots required for intent fulfillment, in priority order below.

Prompt for slot: accountType
Message: For which account would you like your balan…
Slot type
accountType

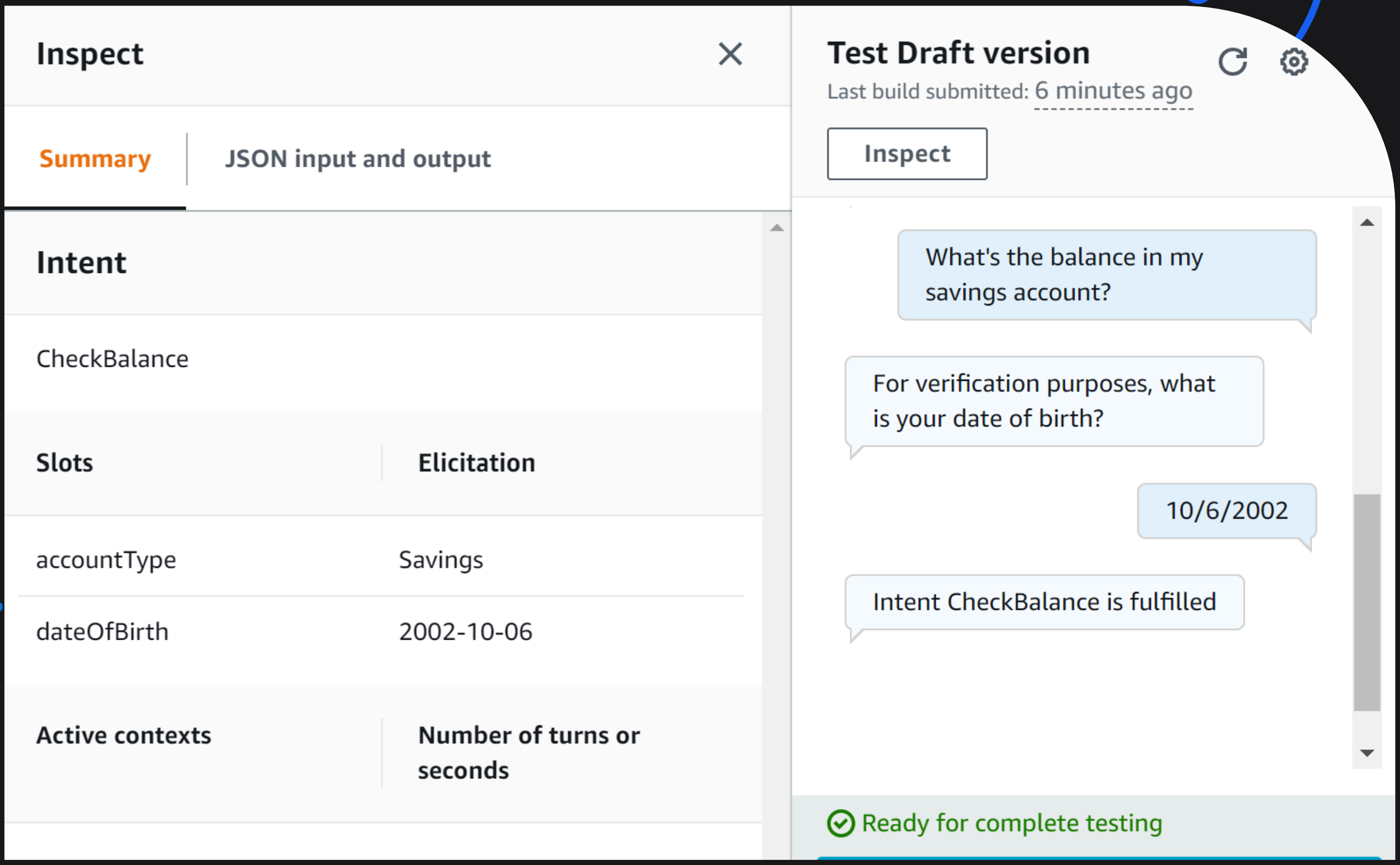
Prompt for slot: dateOfBirth
Message: For verification purposes, what is your date …
Slot type
AMAZON.Date

AVRS Anand

linkedin.com/in/avrsanand/

Simplifying the user experience

- I included slot values in some of the utterances(i.e user inputs)for this intent too. For example, I defined the utterance What's the balance in my {accountType} account? This is an example of an utterance that expects the slot accountType.
- By adding custom slots in utterances, the user experience is enhanced - the bot will automatically register which account type the user is trying to check, and will not ask for it again. This saves the users time and makes the conversation efficient.



Intent

CheckBalance

Slots	Elicitation
accountType	Savings
dateOfBirth	2002-10-06

Active contexts | Number of turns or seconds

What's the balance in my savings account?

For verification purposes, what is your date of birth?

10/6/2002

Intent CheckBalance is fulfilled

Ready for complete testing

Slot values getting recognised in a conversation

AVRS Anand

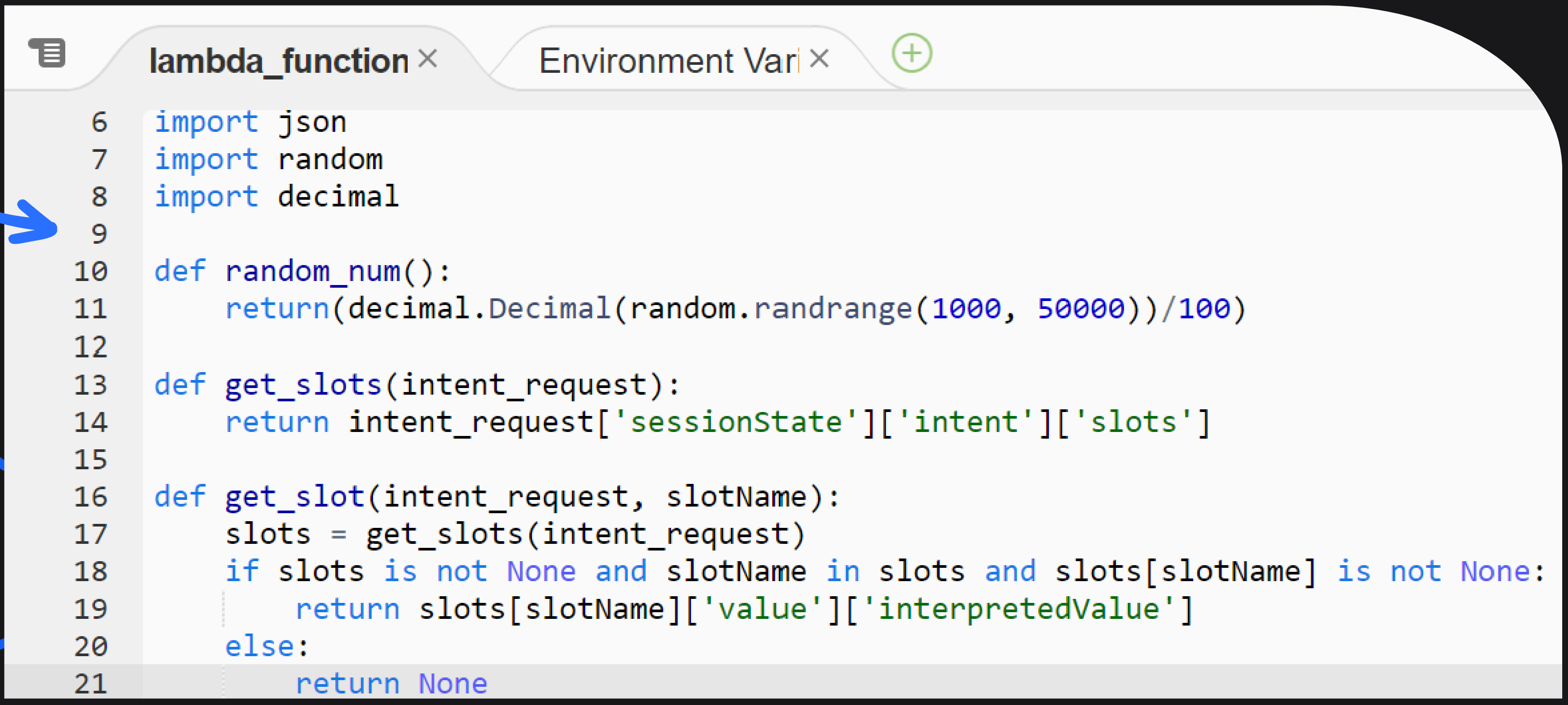
linkedin.com/in/avrsanand/

PART 3

Using AWS Lamba

- **AWS Lambda** is an AWS service that lets you run code without provisioning or managing servers.
- In this project, I created a Lambda function to generate the users' bank balance. A random number is generated, however in the real world the Lambda function can be used to extract users' bank balance from a database. Lex is not capable of doing this on its own.

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



```
import json
import random
import decimal

def random_num():
    return(decimal.Decimal(random.randrange(1000, 50000))/100)

def get_slots(intent_request):
    return intent_request['sessionState']['intent']['slots']

def get_slot(intent_request, slotName):
    slots = get_slots(intent_request)
    if slots is not None and slotName in slots and slots[slotName] is not None:
        return slots[slotName]['value']['interpretedValue']
    else:
        return None
```

AVRS Anand

in linkedin.com/in/avrsanand/

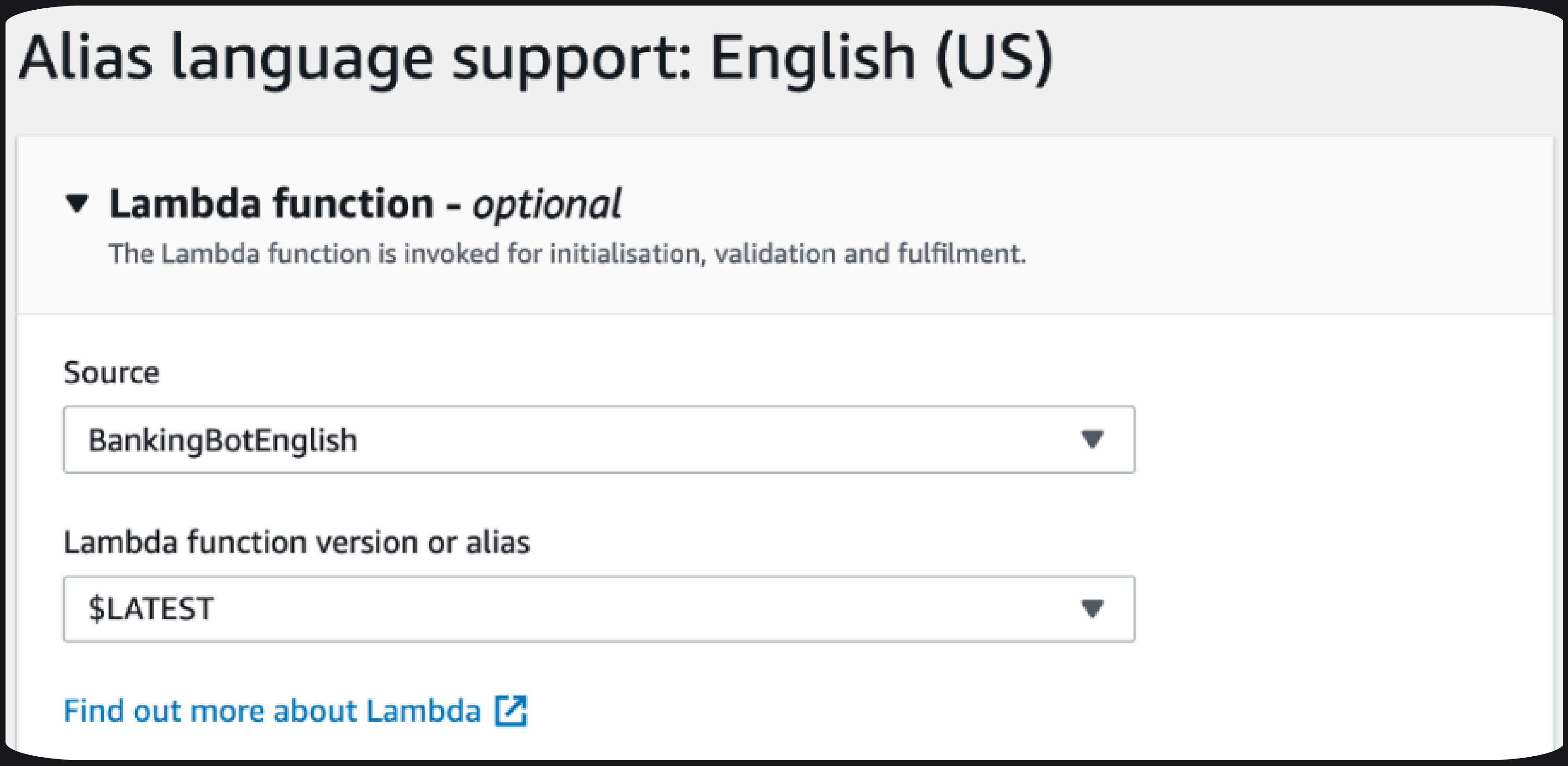
Connecting Lambda with Lex - 1

There were two steps involced to connect Lambda to Lex Chatbot -

Step 1 - Chatbot Alias

- An alias is a pointer for a specific version of your bot. TestBotAlias is a default version of your bot that's made for testing or development. To connect Lambda with my BankerBot, I visited my bot's TestBotAlias(my chatbot's default alisa) and connected it with the latest version of the AWS Lambda function defined.

Connecting latest Lambda function version



Alias language support: English (US)

▼ **Lambda function** - *optional*

The Lambda function is invoked for initialisation, validation and fulfilment.

Source: BankingBotEnglish

Lambda function version or alias: $LATEST

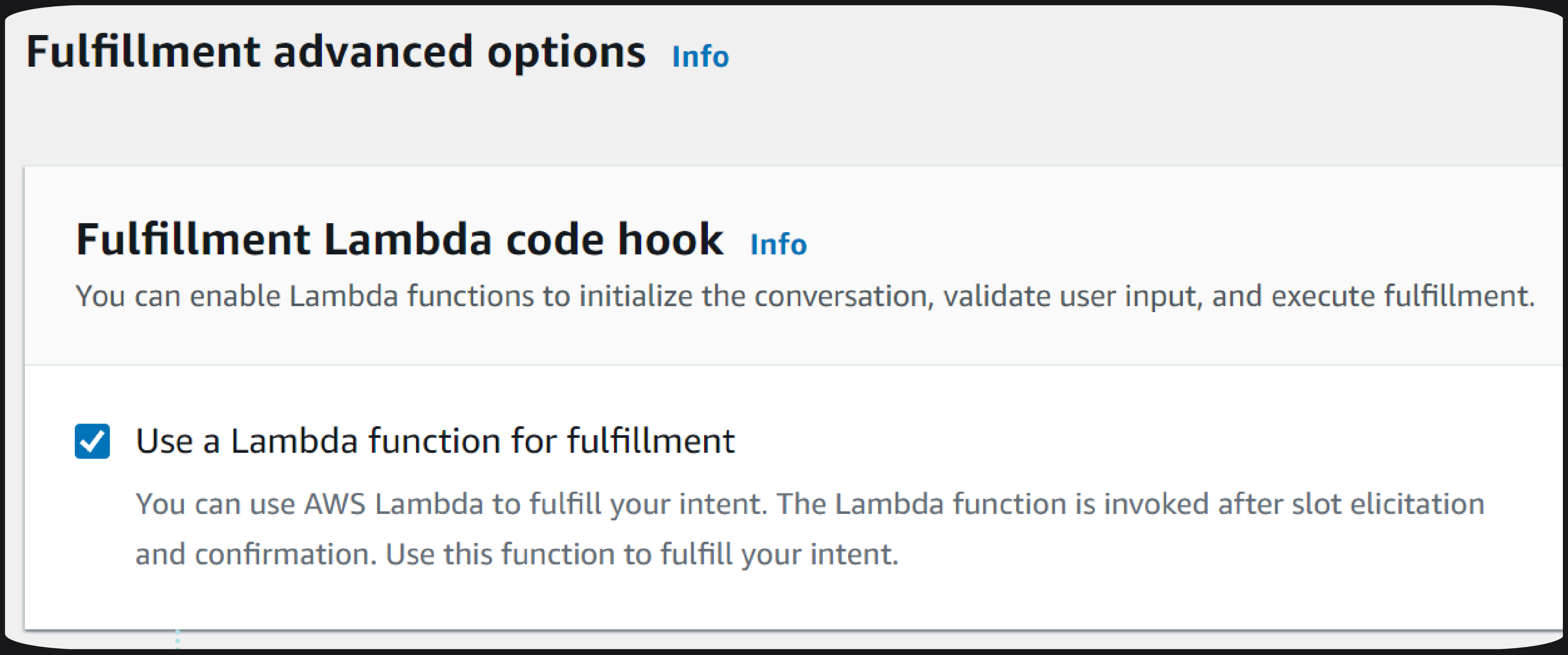
AVRS Anand

linkedin.com/in/avrsanand/

Connecting Lambda with Lex - 2

Step 2 - Code Hooks

- A code hook is a piece of code that can be connected to my chatbot to perform function/actions that my chatbot cannot do alone/by default.
- Even though I already connected my Lambda function with my chatbot's alias, I had to use code hooks because the chatbot is not able to calculate/return a bank balance figure on its own.
- I could find code hooks at CheckBalance intent in the Fulfillment module go to advanced settings to select the check box which specifies "Use a Lambda function for fullfillement"



Fulfillment advanced options Info

Fulfillment Lambda code hook Info

You can enable Lambda functions to initialize the conversation, validate user input, and execute fulfillment.

☑ Use a Lambda function for fulfillment

You can use AWS Lambda to fulfill your intent. The Lambda function is invoked after slot elicitation and confirmation. Use this function to fulfill your intent.

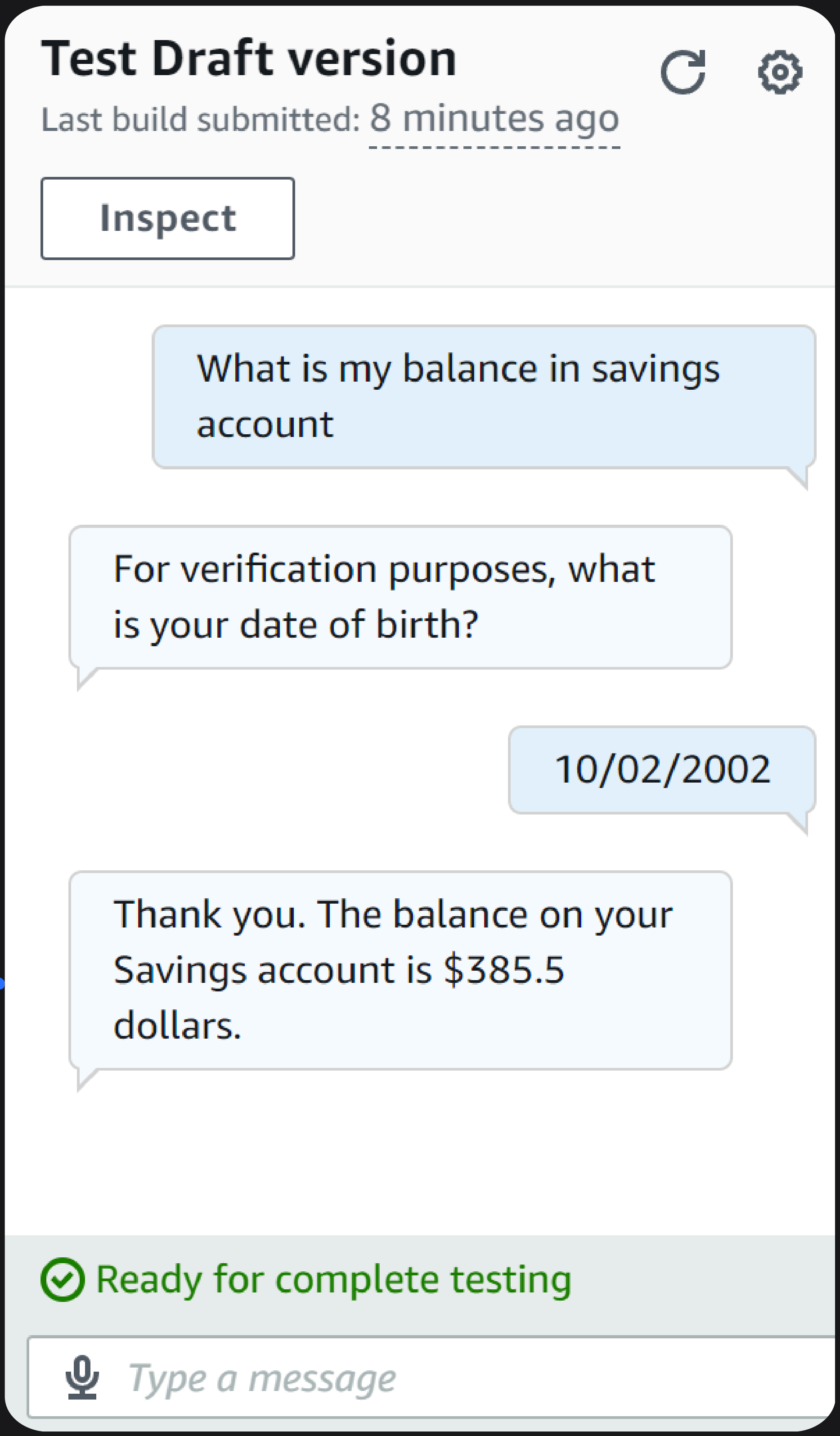
Using Lambda Function for fullfillment

AVRS Anand

linkedin.com/in/avrsanand/

Successful Connection with Lambda

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



Test Draft version

Last build submitted: 8 minutes ago

What is my balance in savings account

For verification purposes, what is your date of birth?

10/02/2002

Thank you. The balance on your Savings account is $385.5 dollars.

Ready for complete testing

My chatbot now returns a bank balance number thanks to Lambda!

AVRS Anand

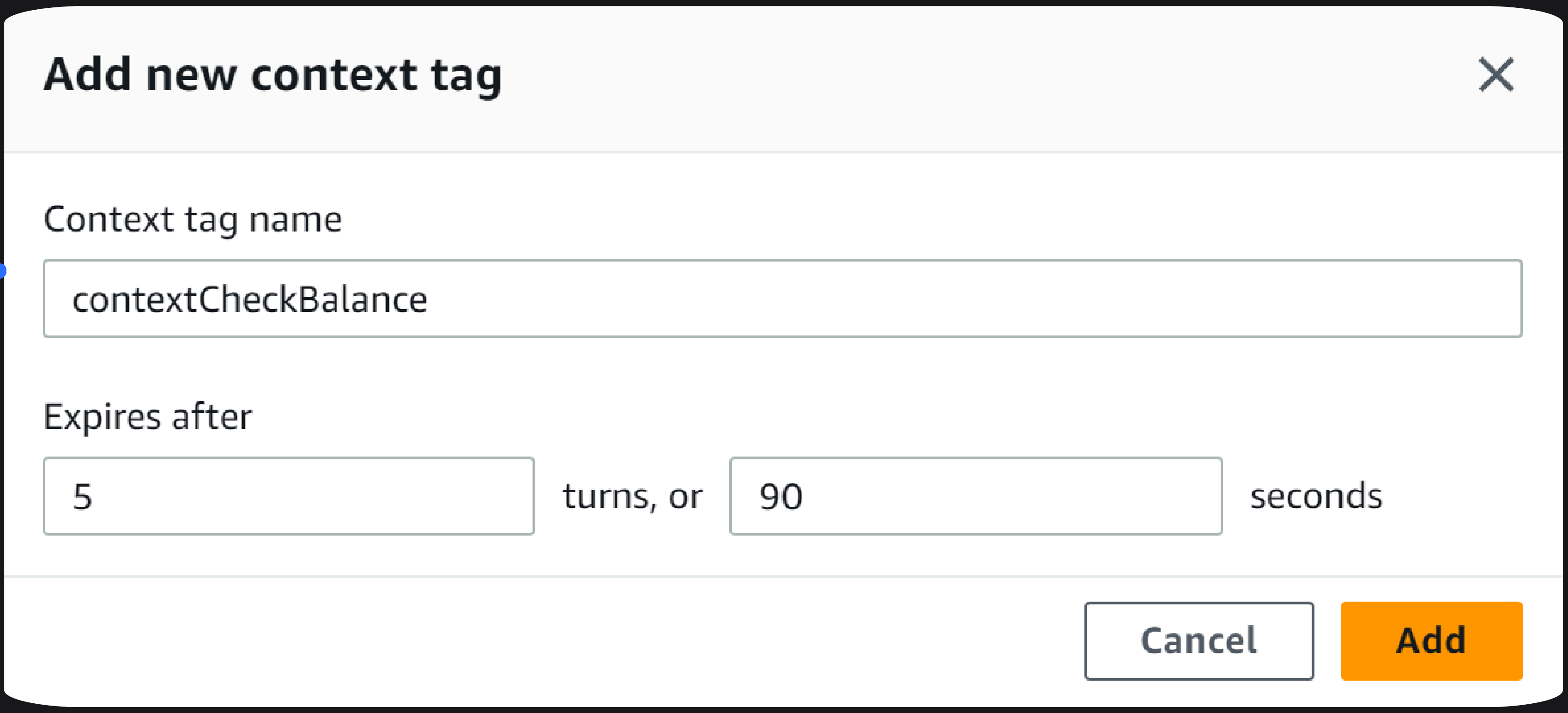
linkedin.com/in/avrsanand/

PART 4

Context Tags

- Context tags are ways for Amazon Lex to remember specific pieces of information gathered from a conversation and reuse that information throughout the session with its user.
- There are two types of context tags, they are output context tags, which allows the chatbot to remember details after an intent if finished, and an input context tag, which checks for readily available details before an intent even activates.
- I created an output context tag called **contextCheckBalance** in my CheckBalance intent so that BankerBot can remember the user 's date of birth for either 5 turns or 90 seconds (too long could cause a security risk).

Created Output context



Add new context tag

Context tag name

contextCheckBalance

Expires after

5 turns, or 90 seconds

AVRS Anand

linkedin.com/in/avrsanand/

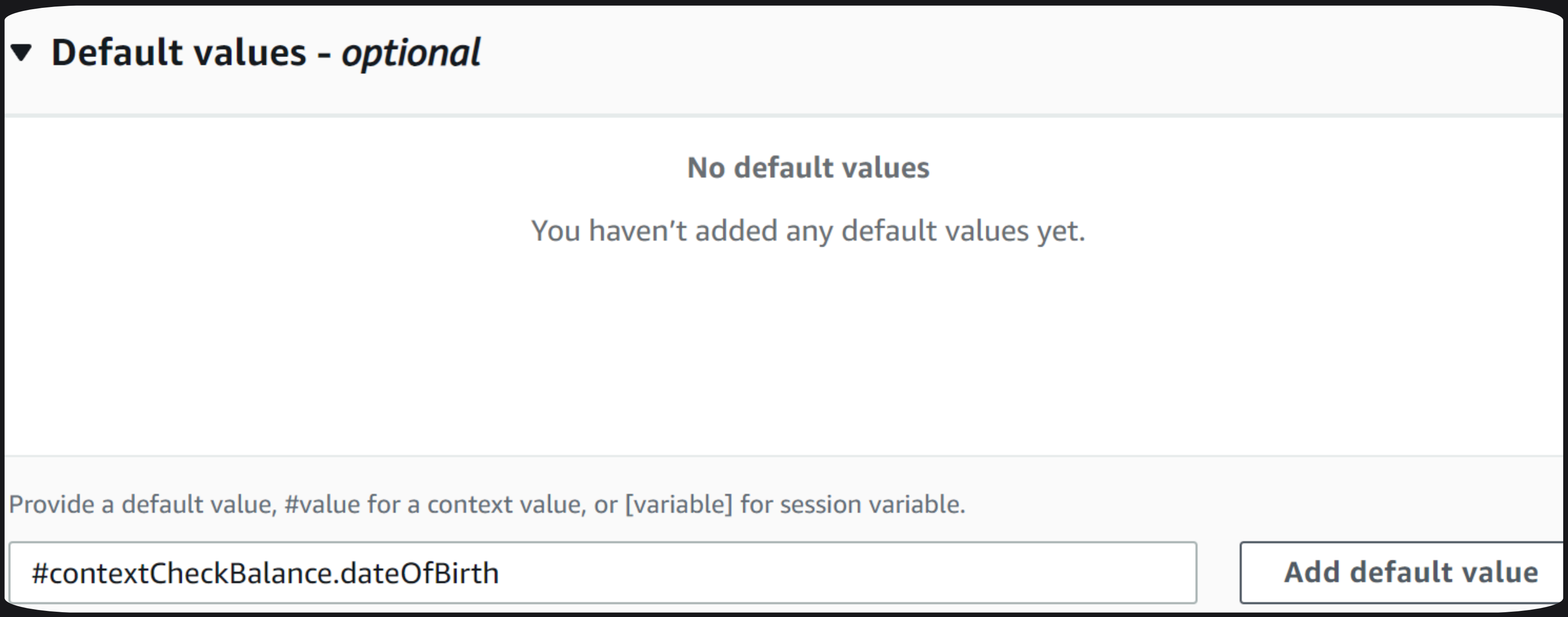
A Follow-Up Intent

- I created a new intent called **FollowupCheckBalance.** The purpose of this intent is to allow user to ask a follow up query without the need of date of birth verification again.

- This intent is related to the previous intent I made, **CheckBalance**, because if a user wants to check balance the chatbot asks for verification,. However the FollowupCheckBalance will help retaining the verification using the context tags.

- I created an input context, **contextCheckBalance**, that uses the value of the user's date of birth that was stored by the output context tag in CheckBalance. The default value for the date of birth slot in FollowupCheckBalance is configured to take in the date of birth from CheckBalance.

Created Input Context


▼ **Default values - *optional***

No default values

You haven't added any default values yet.

Provide a default value, #value for a context value, or [variable] for session variable.

#contextCheckBalance.dateOfBirth

AVRS Anand

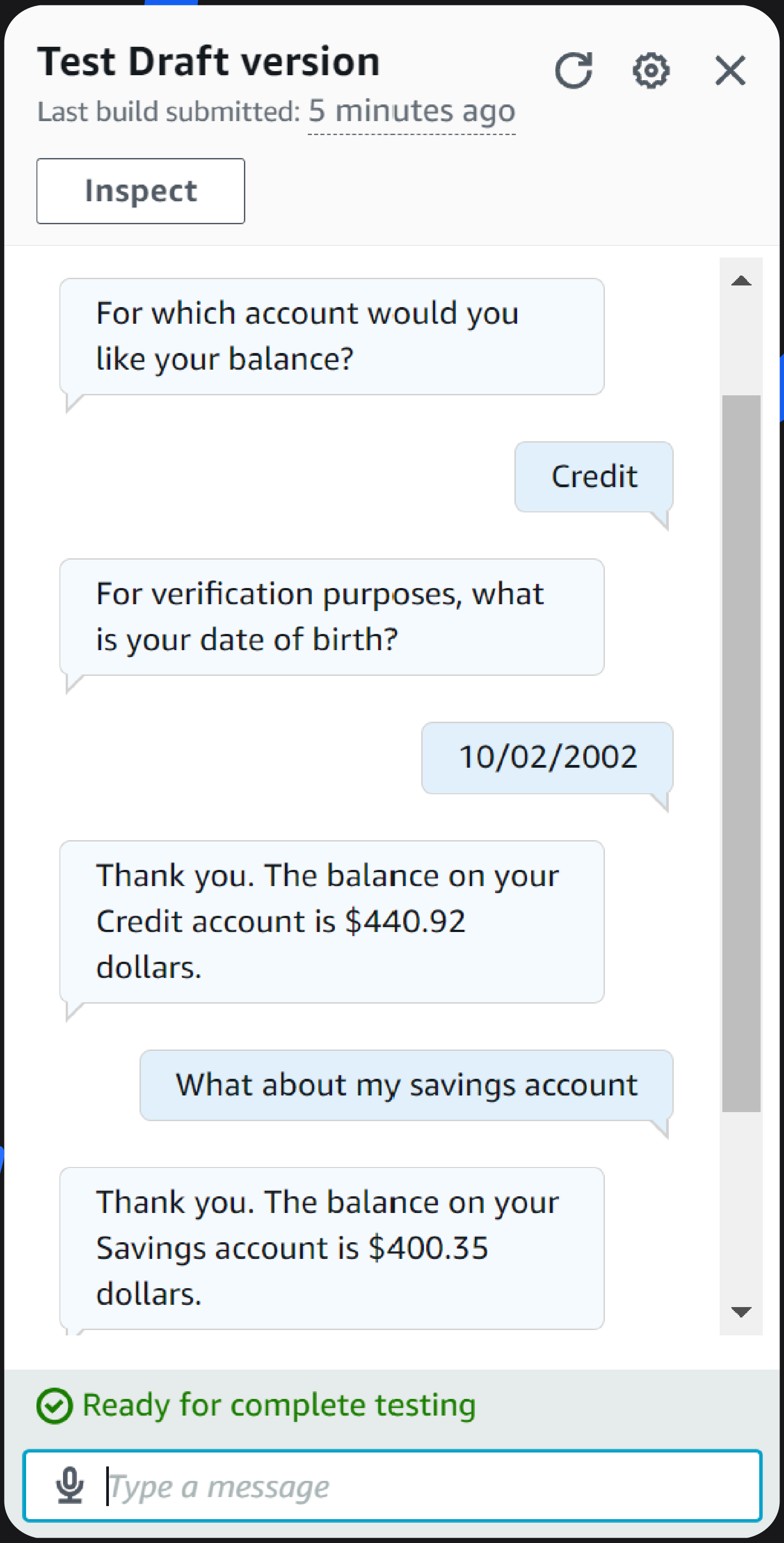
linkedin.com/in/avrsanand/

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 first triggered the CheckBalance intent, then I followed up with the utterance "What about savings account" to trigger FollowupCheckBalance.

- If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, my chatbot would not have the context needed to fulfill the conversation. As a result, it will return the FallbackIntent.



Test Draft version

Last build submitted: 5 minutes ago

Inspect

For which account would you like your balance?

Credit

For verification purposes, what is your date of birth?

10/02/2002

Thank you. The balance on your Credit account is $440.92 dollars.

What about my savings account

Thank you. The balance on your Savings account is $400.35 dollars.

Ready for complete testing

Type a message

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

AVRS Anand

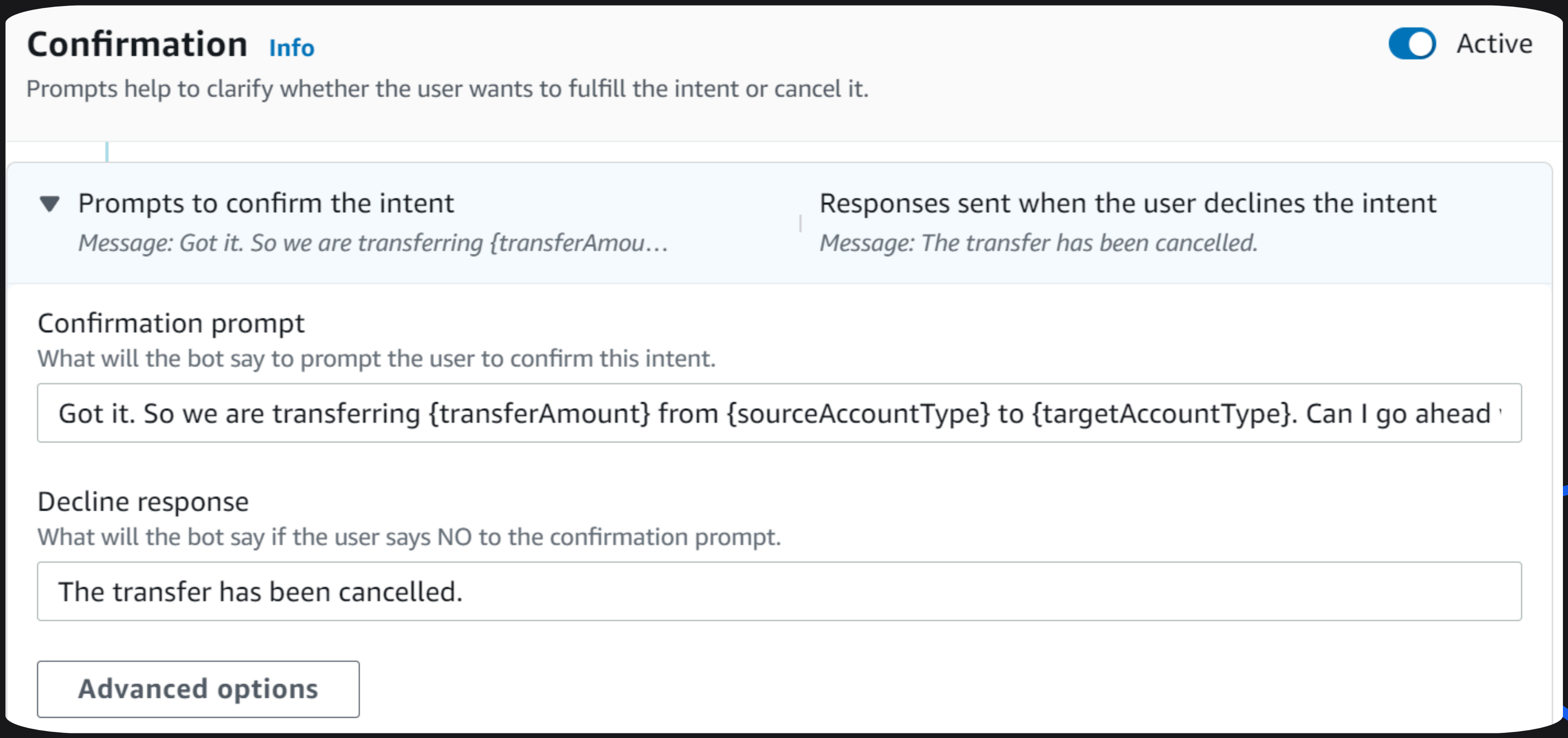
linkedin.com/in/avrsanand/

PART 5

More slots!

- Slots are pieces of information that a chatbot needs to complete a user ' s request.
- The final intent for my chatbot was TransferFunds, which will ask the user to enter the account from, the account to and the amount then use that into to make the transfer
- For this intent, I had to use the same slot type twice. This is because the TransferFunds intent involved two different accounts - the source account and the target account.
- I also learnt how to create confirmation prompts, which are prompts designed for the chatbot to confirm the user's intention to carry out the intent. In this project, a confirmation prompt was used to confirm if the user is transferring the money

Conformation Prompt



Confirmation Info

Active

Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

▼ Prompts to confirm the intent
Message: Got it. So we are transferring {transferAmou…

Responses sent when the user declines the intent
Message: The transfer has been cancelled.

Confirmation prompt
What will the bot say to prompt the user to confirm this intent.

Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I go ahead

Decline response
What will the bot say if the user says NO to the confirmation prompt.

The transfer has been cancelled.

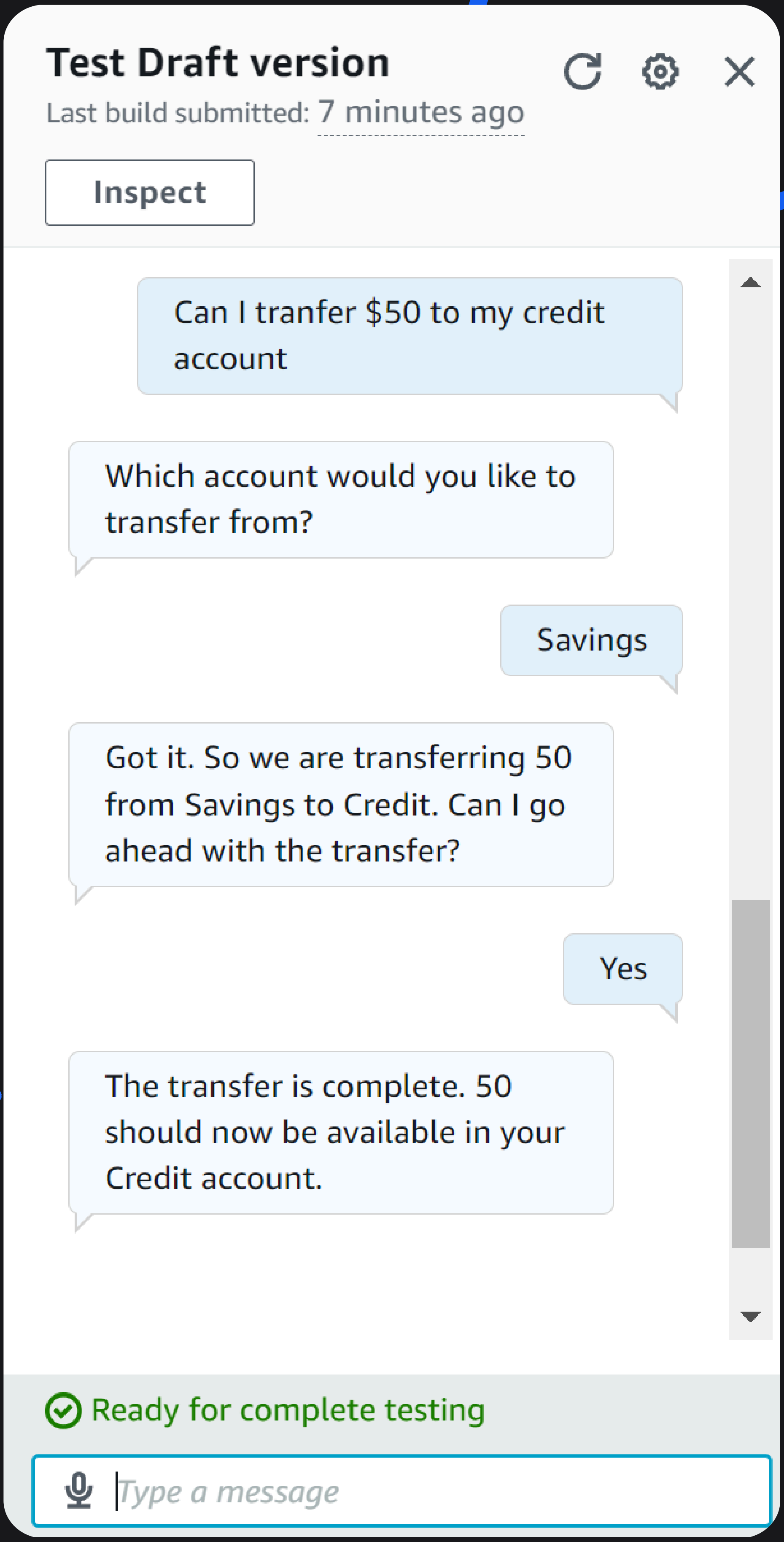
AVRS Anand

linkedin.com/in/avrsanand/

TransferFunds in Action

- The final intent “TransferFunds ” allow us to transfer amount from one account to another account.

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



Test Draft version

Last build submitted: 7 minutes ago

Can I tranfer $50 to my credit account

Which account would you like to transfer from?

Savings

Got it. So we are transferring 50 from Savings to Credit. Can I go ahead with the transfer?

Yes

The transfer is complete. 50 should now be available in your Credit account.

Ready for complete testing

AVRS Anand

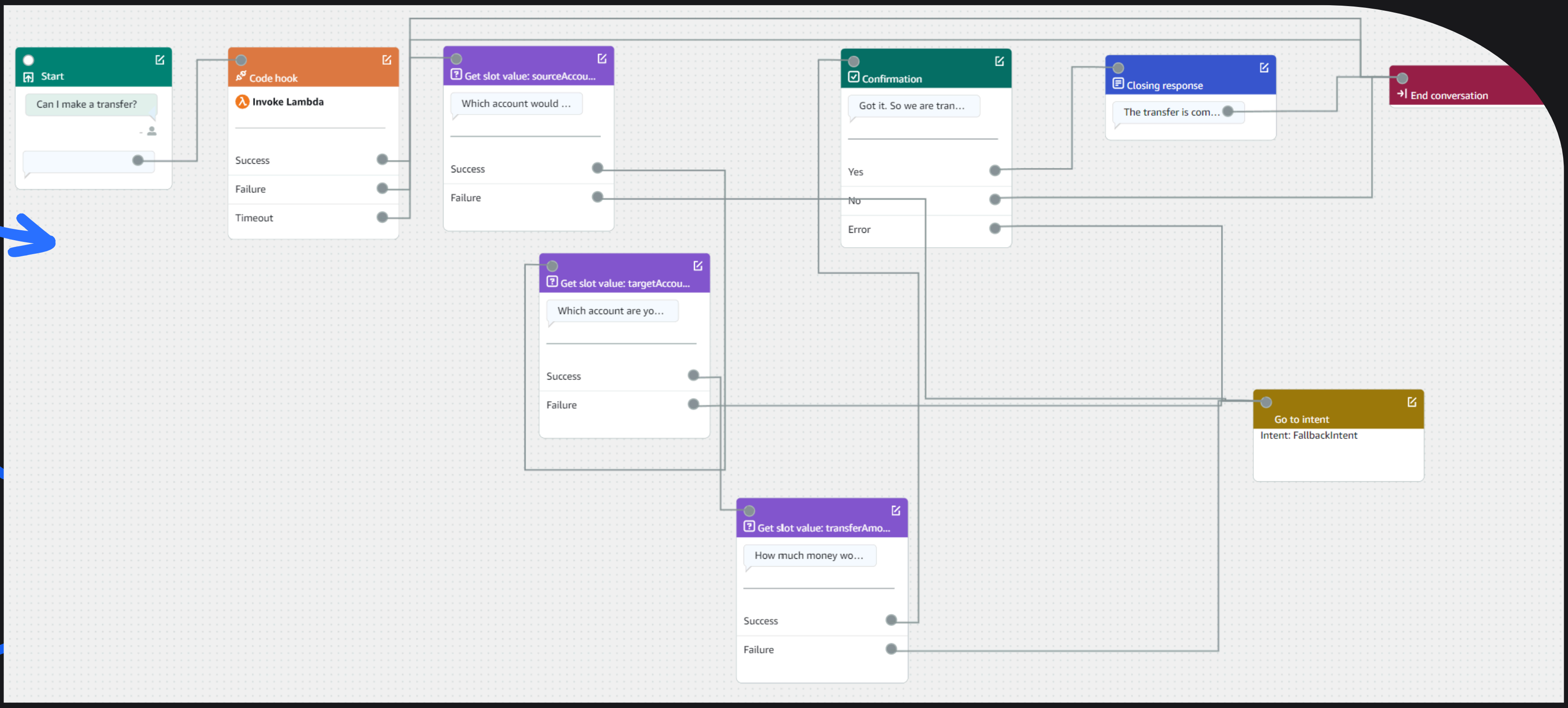
linkedin.com/in/avrsanand/

BONUS

Exploring LEX Features

- Conversation flow will update as you continue editing intent. It shows every step in a conversation in a logical, chronological order. You'll also see some blank 'ghost like' responses. These are recommendations you could add to your Intent.
- You could also set up your intent using a visual builder! A visual builder is a representation of the intent you have just built.

Visual Builder



AVRS Anand

in linkedin.com/in/avrsanand/

My Key Learnings

01 Amazon Lex is an AWS service that enables the building of conversational chatbots into any application using voice and text. Intents are fundamental components that represent the goal behind a user's input

02 Slots are like variables within an intent that captures specific pieces of information from the user's input necessary to fulfill the intent.

03 AWS Lambda is a serverless compute service provided by Amazon Web Services (AWS) that allows you to run code without provisioning or managing servers.

04 AWS Lambda is a serverless compute service provided by Amazon Web Services (AWS) that allows you to run code without provisioning or managing servers. You connect Lex with Lambda using alias & code hooks. This makes our chatbot smarter and more useful by allowing us automate tasks during chats.

05 Context tags are used to store and check specific information during a conversation to be used later. Confirmation prompts typically repeats back information for the user to confirm.

AVRS Anand

linkedin.com/in/avrsanand/

Final thoughts...

- This project took me around 2-3 hours writing the documentation took me another 1-2 hours.
- Delete EVERYTHING at the end! Let’s keep this project free :)
- In the next phase of this project, I’m excited to level up my AWS skills by using AWS CloudFormation to recreate my bot in seconds!

AVRS Anand

linkedin.com/in/avrsanand/

Find this helpful?

- Like this post yes!
- Leave a comment
- Save for later
- Let's connect!

AVRS Anand

linkedin.com/in/avrsanand/