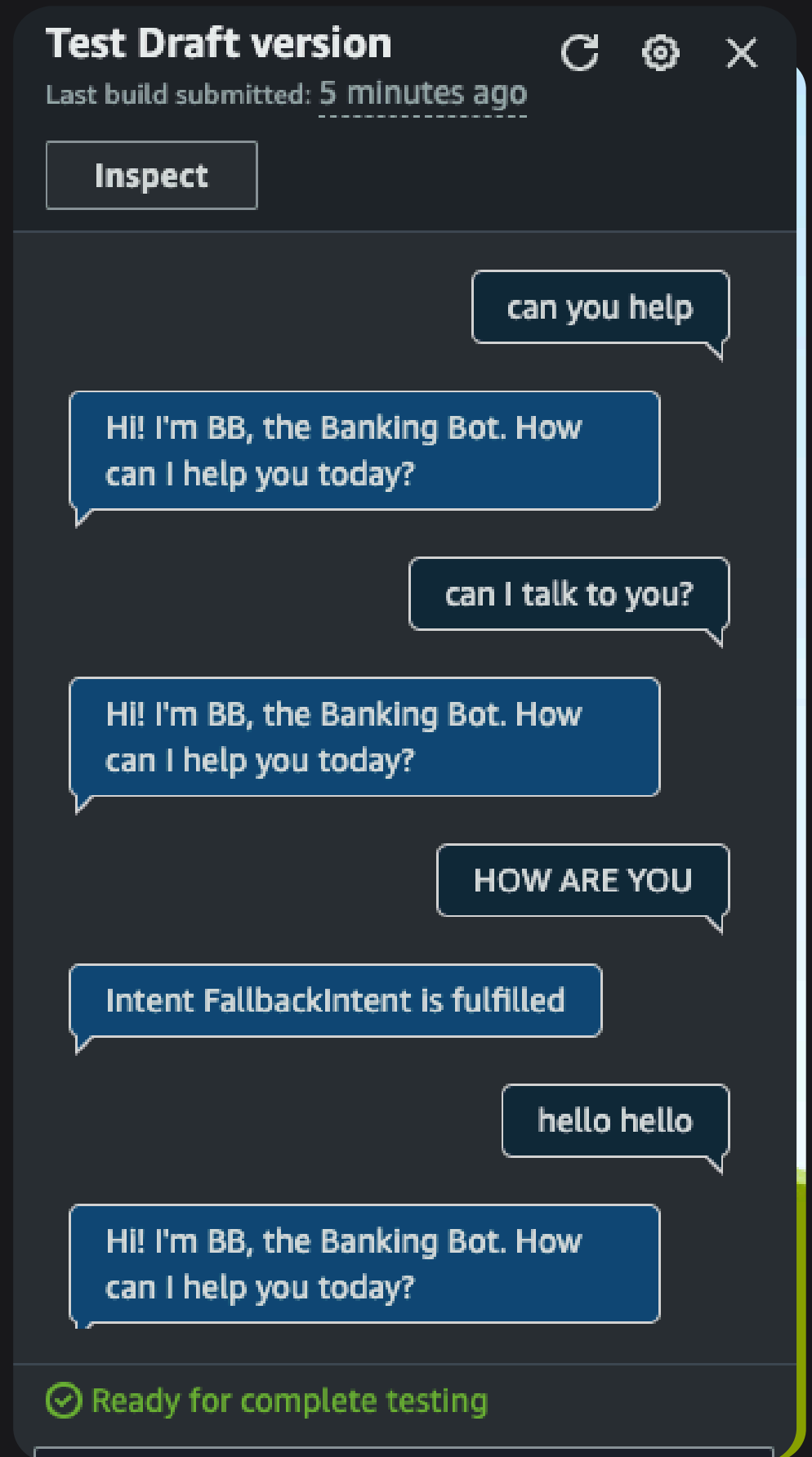



# How I built a chatbot with Amazon Lex



Rasha M

 @Badry2022

 [Rasha M.](#)

# TABLE OF CONTENTS

1

## What is Amazon Lex?

Amazon Lex is a tool that helps you create programs that can talk or chat with people, just like Siri or Alexa. This project is part ONE of five in this Amazon Lex chatbot series 🤖 In this intro, you'll learn how to set up a chatbot in Amazon Lex and make it respond to different user inputs.

## Create custom slots

In this second part, you'll set up a new intent that helps you check your account balance! This will be a fun level up where you'll learn to make custom slot types.

2

3

## AWS Lambda!

In this third part, you'll continue with building the intent that helps you check your account balance! This will be a fun level up where you'll learn to connect your Lex bot with AWS Lambda.

## Context Tags

In this fourth part, you'll learn to conduct a context carryover, which means bringing over things you've learnt about the user (e.g. their date of birth) to another intent!

4

5

## Multiple slots

In this final part, you'll create an intent that lets you transfer money between accounts

## Deploying with CloudFormation

learn about a handy service called AWS CloudFormation.

6

7

## French Version

This is the french version of the project (exact same steps)



# What is Amazon Lex?

## **What it does:**

- **Amazon Lex builds conversational interfaces using voice and text.**

## **Why it's useful:**

- It simplifies creating chatbots that understand natural language and can perform complex tasks.

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

- In this project, I'm using Amazon Lex to create BankerBot, a chatbot that helps customers check their account balances and transfer money.

# Set up a Lex chatbot

- I created BankerBot from scratch and used most default settings on Lex. My favorite voice option was Ruth .
- In terms of the intent classification confidence score, I kept the default value of 0.40. What this means for my chatbot is that it needs to be at least 40% confident in understanding user input to respond appropriately. This helps ensure the bot provides accurate and relevant responses.

Setting up my Lex  
chatbot...



The screenshot shows the configuration interface for a Lex chatbot. It includes a dropdown for 'Language: English (US)', a 'Select language' dropdown set to 'English (US)', a 'Description - optional' text area, a 'Voice Interaction' dropdown set to 'Ruth', a 'Voice sample' text input with the text 'Hello, my name is Ruth. Let me know how I can assist you. Don't bel' and a 'Play' button, and an 'Intent classification confidence score threshold' input set to '0.40'.

# Create an intent in Lex

Intents are specific goals users want to achieve with the chatbot, like checking a balance or making a transfer.

My first intent, WelcomeIntent, was created to greet users and assist them when they initiate a conversation.

To set up this intent:

- I Named it WelcomeIntent.
- Added sample utterances like "Hi," "Hello," and "I need help."
- Set a response message: "Hi! I'm BB, the Banking Bot. How can I help you today?"

I launched and tested the chatbot, which could still respond if I entered:

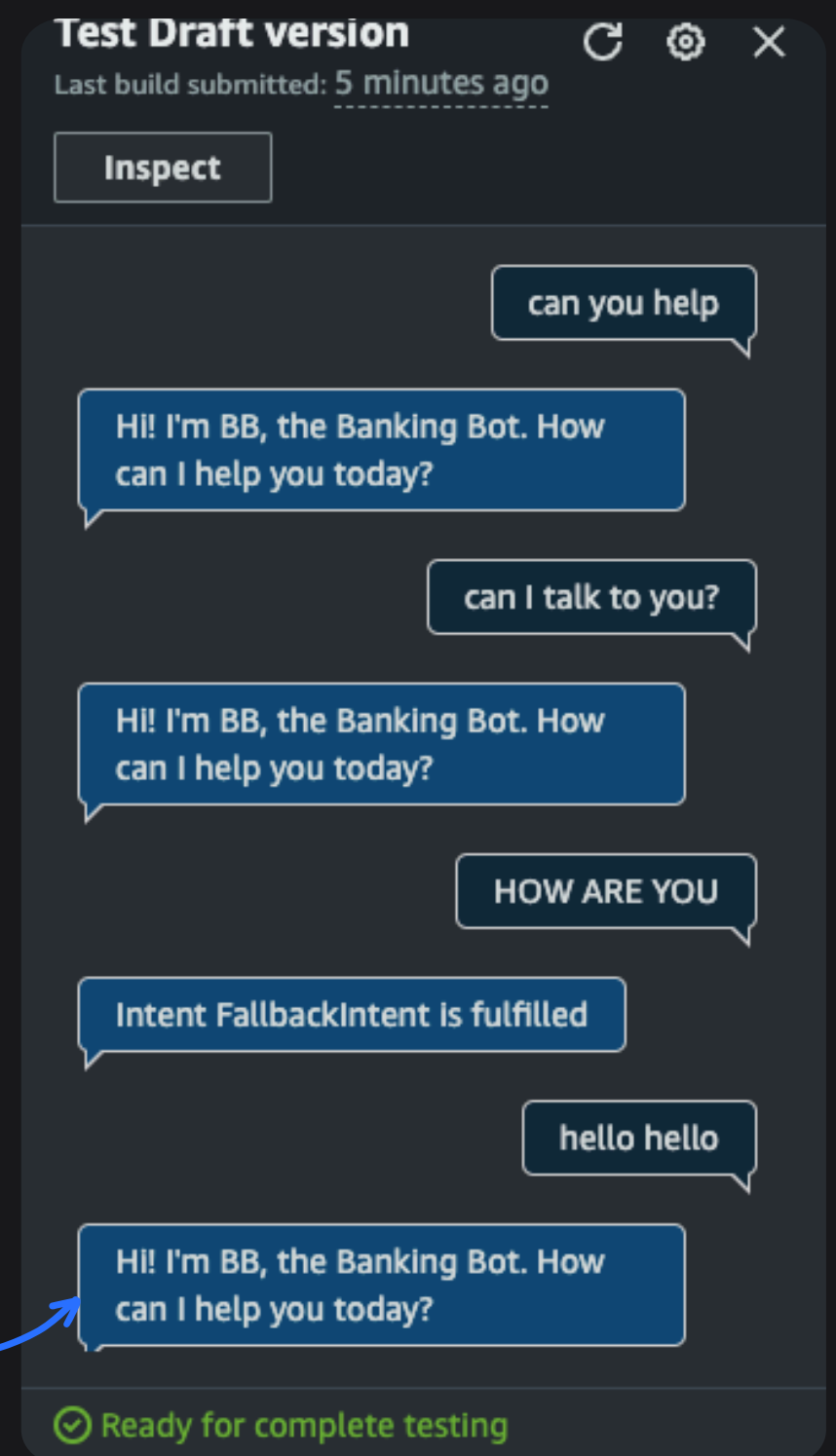
- "Help me"
- "Help m,e"
- "Hiya"

However, the chatbot returned the error message "Intent FallbackIntent is fulfilled" when I entered:

- "How are you"
- Any weird questions

This error message occurred because the chatbot's confidence score for these inputs was below the 0.40 threshold.

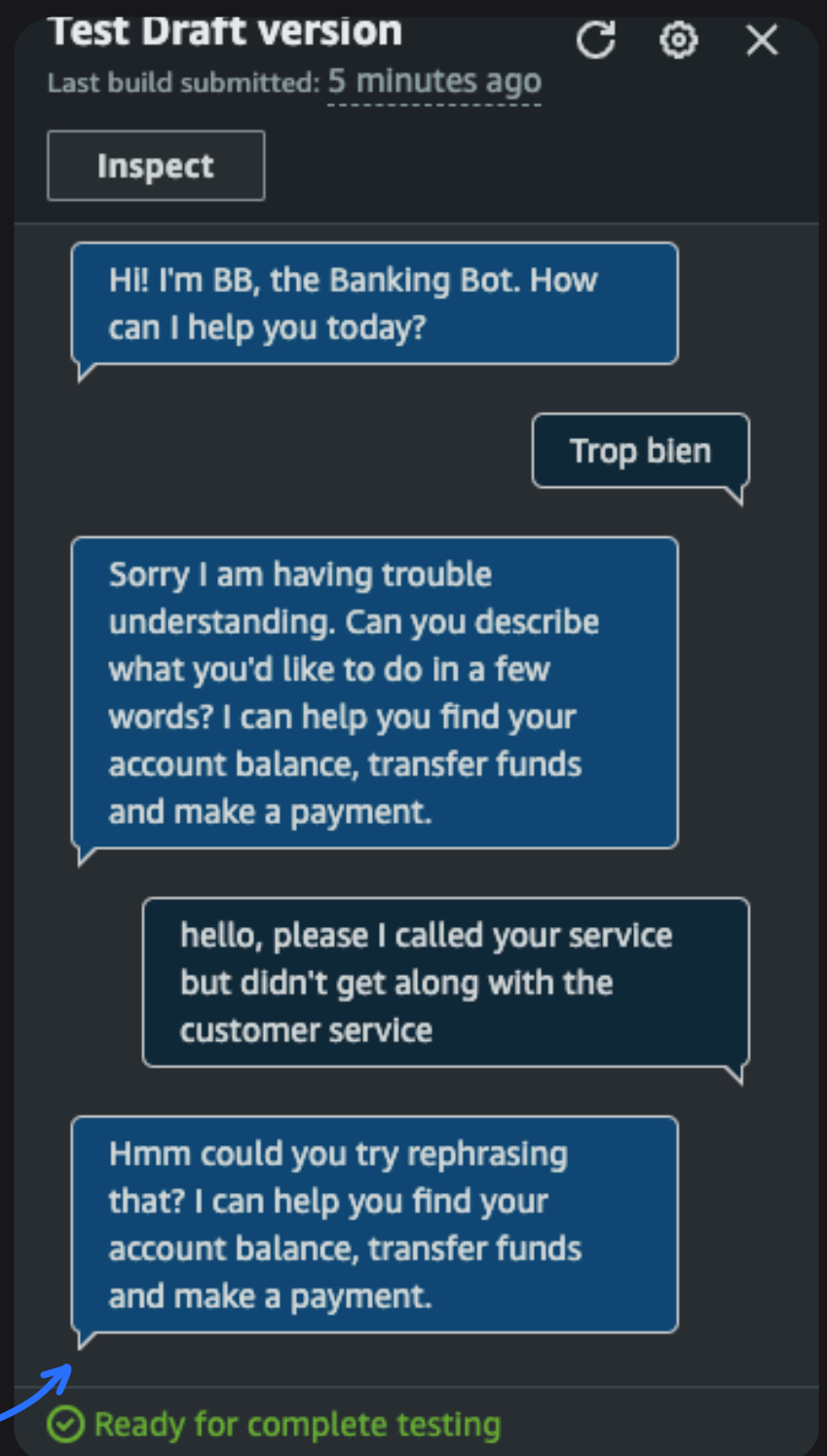
My first test of the  
chatbot



# Manage FallbackIntent

- FallbackIntent is a default intent in every chatbot that gets triggered when the confidence score is below the threshold (0.4 in our case).
- I wanted to configure FallbackIntent to provide clearer error messages when the chatbot doesn't understand the user's input.
- To configure FallbackIntent, I updated the closing response message to guide users on how to phrase their requests and added more context.
- I also added variations! What this means for an end user is a more dynamic and conversational experience when the chatbot doesn't understand their input.

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



# My Key Learnings

**01**

- Amazon Lex is a service that enables anyone to build conversational interfaces, or chatbots, using voice and text
- It's a no-code tool, which means, you don't need to be an AI expert to create a chatbot

**02**

Intents in Amazon Lex represent the goals or actions that users want to achieve through their interactions with the chatbot.

**03**

AI/ML is used in Amazon Lex to train the chatbot on recognizing intents and processing user inputs accurately.

**04**

FallbackIntent is employed for handling user inputs that the chatbot cannot confidently classify based on its training, ensuring a fallback response is provided for clarity.

**05**

Creating a chatbot with Amazon Lex highlighted how accessible it is for businesses to develop tailored conversational interfaces, enhancing customer interaction or service delivery.

# Final thoughts...

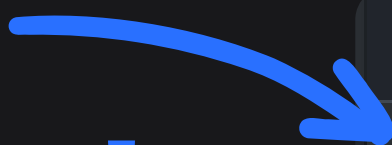
- This project took me approximately 1 hour to complete, and documentation took me about 1 hour.
- Delete EVERYTHING at the end! Let's keep this project free :)
- One thing I didn't expect was how straightforward it was to set up a functional chatbot using Amazon Lex.
- What's next? In the next phase 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! 🚀👁️



# Part 2 : How I built a chatbot with Amazon Lex



featuring  
custom slots!



**Inspect**

I want to check my account

For which account would you like your balance?

card

For verification purposes, what is your date of birth?

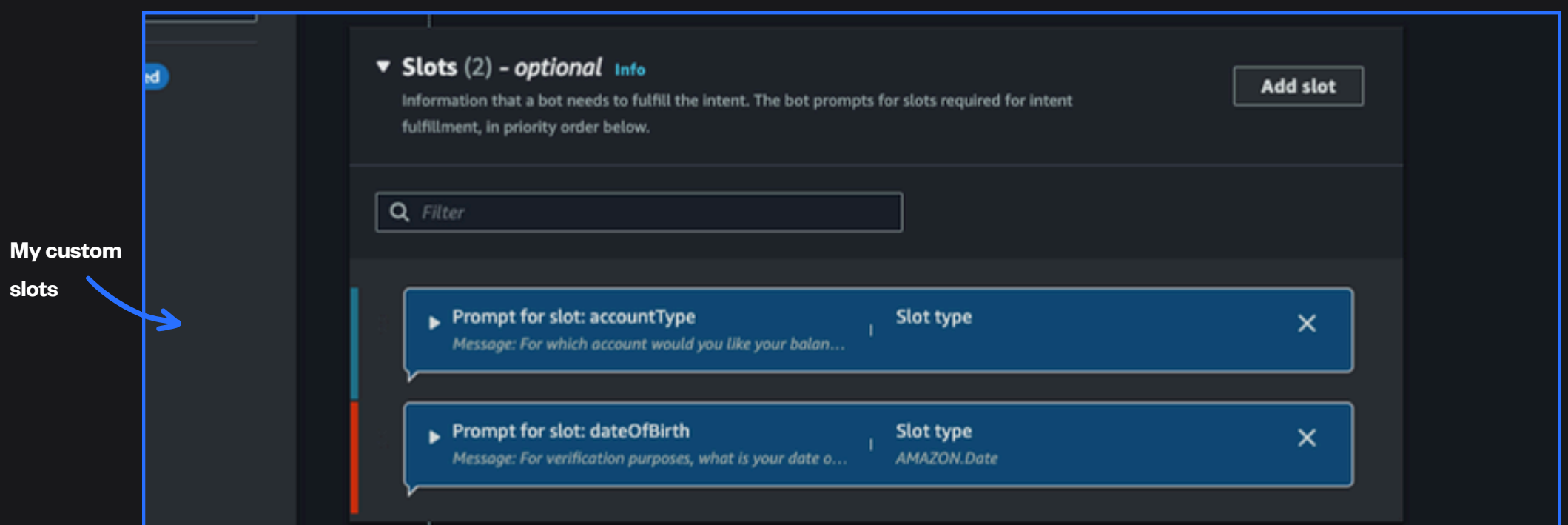
2019

Intent CheckBalance is fulfilled

✓ Ready for complete testing

# Create custom slots

- Slots are pieces of information that a chatbot needs to complete a user's request, like filling in blanks on a form or a date of birth.
- Amazon Lex supports built-in slot types that define how data in the slot is recognized and handled. You can create slots of these types in your intents. This eliminates the need to create enumeration values for commonly used slot data such as date, time, and location.
- In this project, I created a custom slot to represent different bank account types: Checking, Credit, and Savings.
- I then associated the custom slot with a new intent, CheckBalance, which allows users to inquire about the balance of a specific type of bank account, ensuring accurate and efficient responses.



# Simplifying the user experience

- I included slot values in some of the utterances (user inputs) for this intent too. For example, utterances like "What's the balance in my {accountType} account?" and "How much do I have in {accountType}?" include the slot value {accountType}.
- By adding custom slots in the utterance, Amazon Lex can automatically fill in the slot information if it matches the user's input. This saves the bot from having to ask for the account type separately, streamlining the interaction and saving the user from needing to declare it explicitly. This results in a more efficient and user-friendly experience.



Slot values getting recognised in a conversation

Intent	
CheckBalance	
Slots	Elicitation
accountType	Savings
dateOfBirth	1920-01-01

I want to check my balance please

For which account would you like your balance?

card

For verification purposes, what is your date of birth?

1920

# My Key Learnings

**01**

Slots are pieces of information that a chatbot needs to complete a user's request, similar to blanks in a form.

**02**

Slot types define the kind of data a slot can hold, such as dates, times, or custom values. Custom slot types are user-defined categories specific to particular needs.

**03**

I used a custom slot type in this project to represent different bank account types (Checking, Credit, Savings) to ensure accurate recognition and handling of these specific categories.

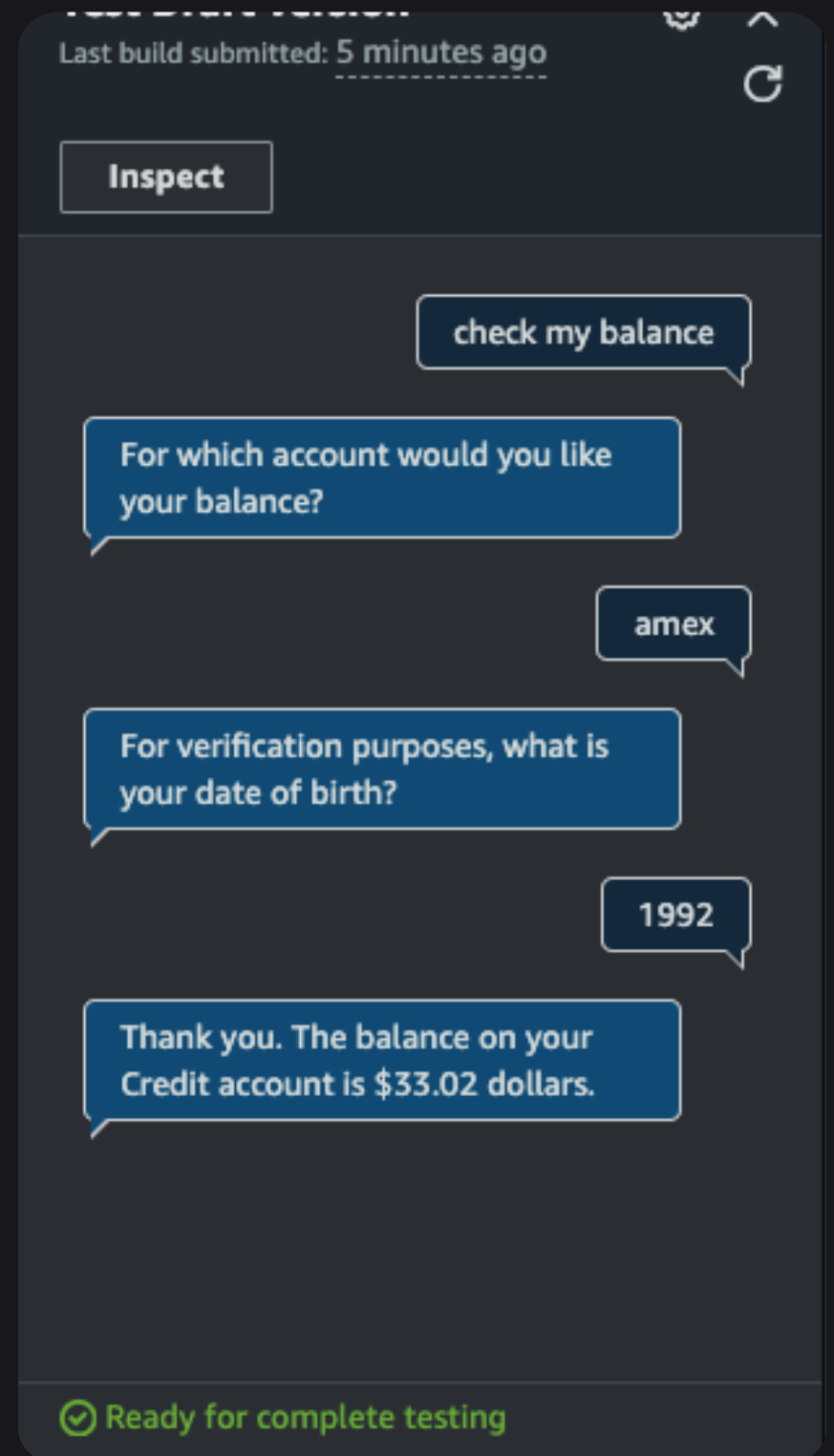
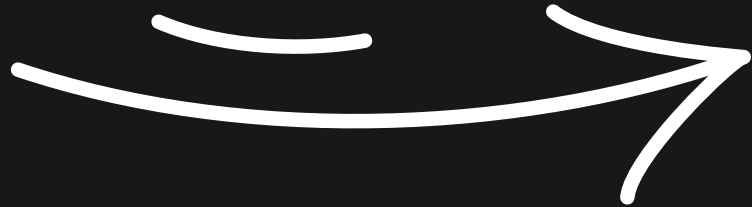
**04**

I parsed my custom slot in the utterance by including the slot value `{accountType}` in a prompt. For example, in "What's the balance in my `{accountType}` account?", Lex recognizes and fills in the account type from the user's input.

**05**

I also learned how to add and use another slot for the user's date of birth (`dateOfBirth`). This helps with verifying the user's identity before providing sensitive information like account balances. By incorporating the `AMAZON.Date` slot type for `dateOfBirth`, the chatbot can ask for and validate the user's birth date seamlessly, ensuring an extra layer of security in the interaction. I have the same interaction when I call my banker, that's amazing!

# Part 3 : How I connected my chatbot with **AWS** **Lambda**!





# Using AWS Lambda

## AWS Lambda :

- A Lambda function's execution role is an AWS Identity and Access Management (IAM) role that grants the function permission to access AWS services and resources. This is sometimes called Functions-as-a-Service (FaaS).
- AWS Lambda runs code without managing servers, scaling automatically.
- Useful for serverless applications, automation, and data processing.

## In this project :

- Created a Lambda function to handle CheckBalance for BankerBot.
- BankingBotEnglish NextWork.py generates and returns a random bank balance.
- This function adds dynamic data retrieval that the chatbot can't perform on its own.

```
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, 50000))/100)
12
13 def get_slots(intent_request):
14     return intent_request['sessionState']['intent']['slots']
15
16 def get_slot(intent_request, slotName):
```

# Connecting Lambda with Lex

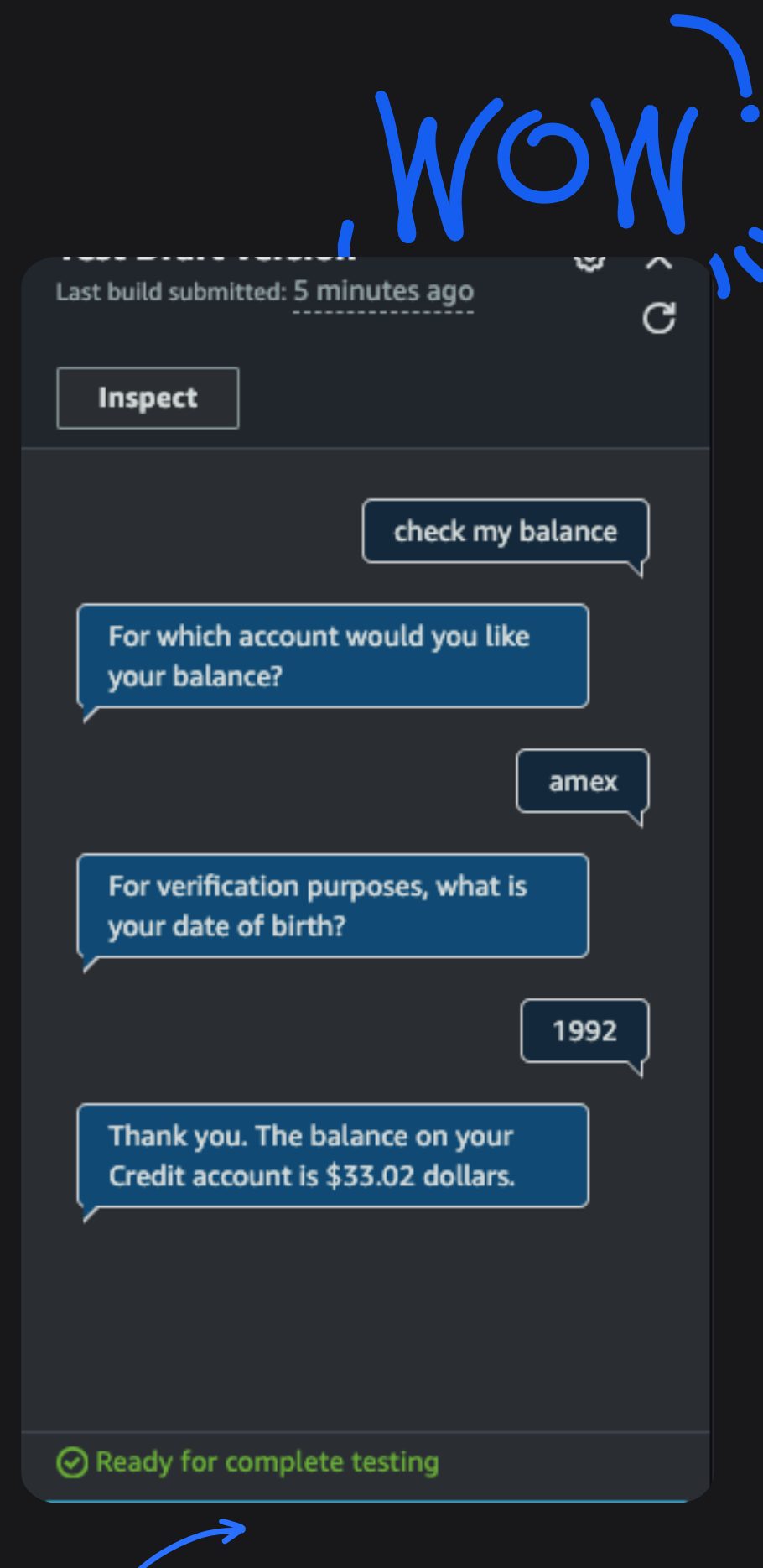
Step **1** To connect Lambda with my chatbot alias, I selected the Lambda function BankingBotEnglish in the Lex console under the TestBotAlias settings and saved the configuration.

Step **2** Another intent setting to configure is code hooks.

A code hook is a mechanism that allows the integration of custom logic or external services into a chatbot's workflow, enabling more sophisticated interactions.

In this project, I had to use code hooks because they facilitate the integration of the AWS Lambda function BankingBotEnglish with the CheckBalance intent in my Lex chatbot. This integration is necessary to fetch and return dynamic data, such as random bank balance figures, which the chatbot cannot generate on its own.

After connecting Lambda with my Lex bot and enabling the code hook, my chatbot could successfully retrieve and provide users with random bank balance figures in response to inquiries about their account balances.



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



# My Key Learnings

**01**

AWS Lambda is a service by Amazon that lets you run code without managing servers. It scales automatically and is used when you need to process data, connect to databases, or perform calculations based on user input.

**02**

You connect Amazon Lex with AWS Lambda when your chatbot needs to perform tasks beyond simple responses, such as accessing databases, performing calculations, or integrating with external APIs.

**03**

Set up a Lambda function in AWS Lambda console with a python code, In Amazon Lex console, configure your bot to use this Lambda function for specific intents

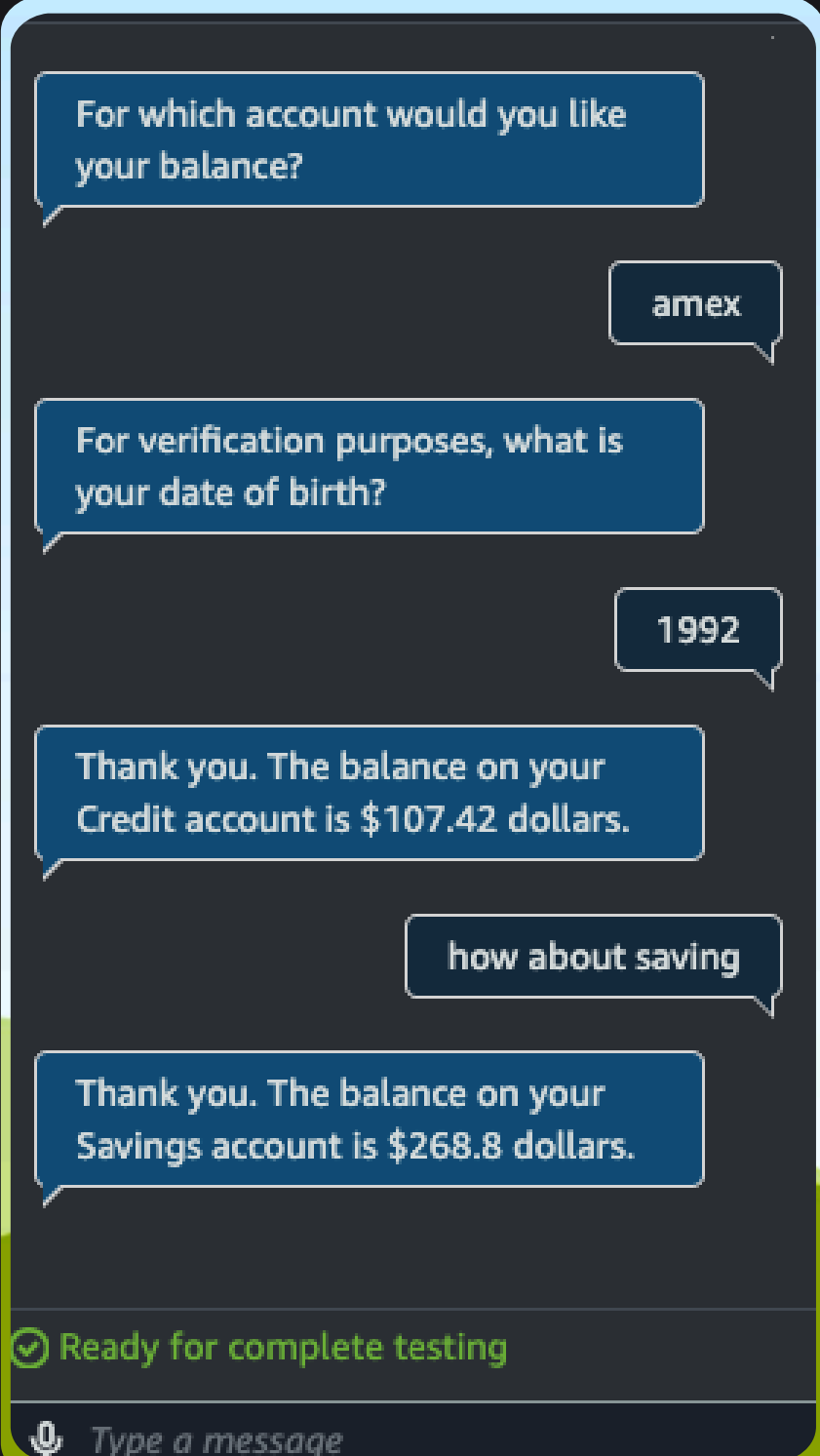
**04**

Understanding how to integrate AWS services like Amazon Lex and AWS Lambda has shown me the power of serverless architecture in building scalable and efficient applications.



# Part 4: How I built a chatbot with Amazon Lex

**that can remember user info!**



For which account would you like your balance?

amex

For verification purposes, what is your date of birth?

1992

Thank you. The balance on your Credit account is \$107.42 dollars.

how about saving

Thank you. The balance on your Savings account is \$268.8 dollars.

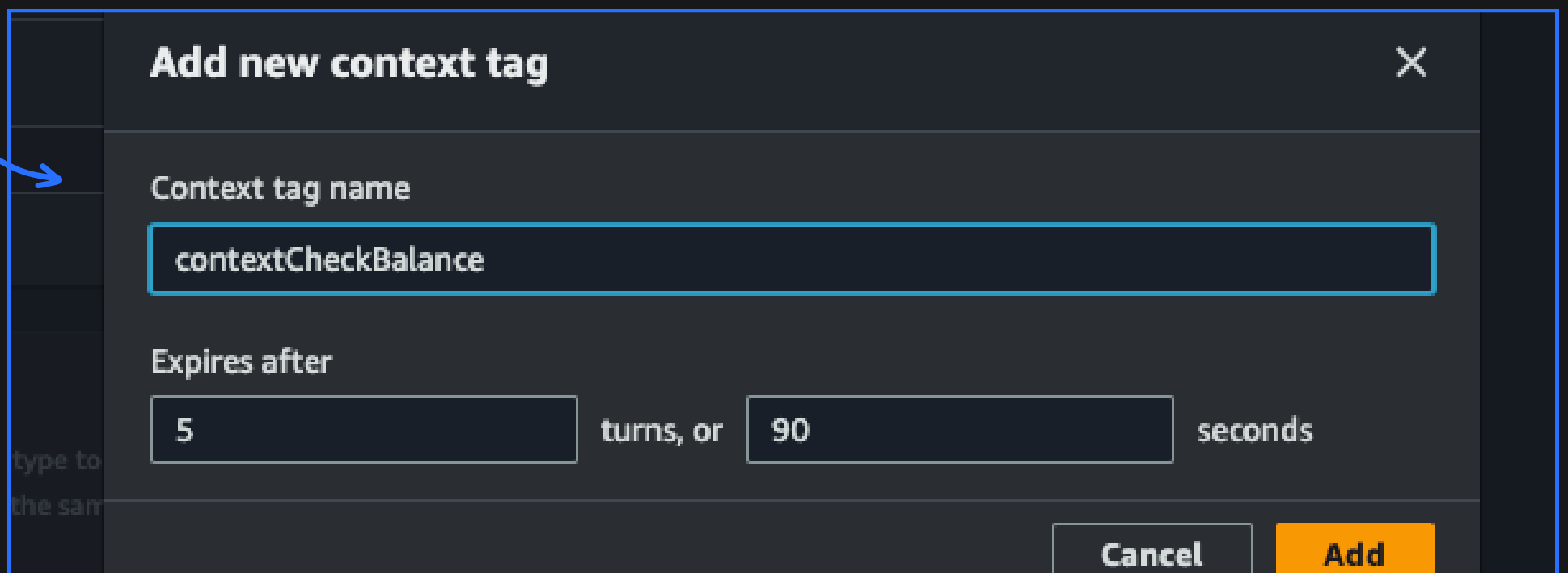
✓ Ready for complete testing

Type a message

# Context Tags

- Context tags in Amazon Lex manage information throughout a conversation, improving user experience by avoiding repetitive questions.
- There are two types of context tags :
- Output context tags: Store information after an intent is fulfilled, enabling other intents to access it later. For instance, saving account type details after a balance check.
- Input context tags: Check for specific information before activating an intent, ensuring that already provided details, like a user's birthday, aren't requested again.
- I created an output context tag called "contextCheckBalance" in the CheckBalance intent.

A look at output contexts



The screenshot shows a dialog box titled "Add new context tag" with a close button (X) in the top right corner. The dialog has a dark background with light text. It contains two main sections: "Context tag name" and "Expires after". The "Context tag name" section has a text input field containing "contextCheckBalance". The "Expires after" section has two input fields: the first contains "5" and the second contains "90", with the text "turns, or" between them and "seconds" at the end. At the bottom right, there are two buttons: "Cancel" and "Add". A blue arrow points from the text "A look at output contexts" to the "contextCheckBalance" input field. There is also some faint, partially visible text on the left side of the dialog that says "type to" and "the sam".

**Add new context tag**

Context tag name

contextCheckBalance

Expires after

5 turns, or 90 seconds

Cancel Add

# A Follow-Up Intent



- I created a new intent called FollowupCheckBalance. The purpose of this intent is to allow users to request a follow-up account balance check without re-entering previously provided information, such as their date of birth.
- This intent is related to the previous intent I made, CheckBalance, because FollowupCheckBalance is triggered after CheckBalance when the user wants to check another account balance. It utilizes the context stored from CheckBalance to avoid asking for the date of birth again.
- I created an input context, contextCheckBalance, that is connected to the output context tag from CheckBalance, allowing it to access stored information, such as the date of birth, ensuring a seamless user experience without redundant prompts.

A look at input contexts

▼ Default values - optional

#contextCheckBalance.dateOfBirth

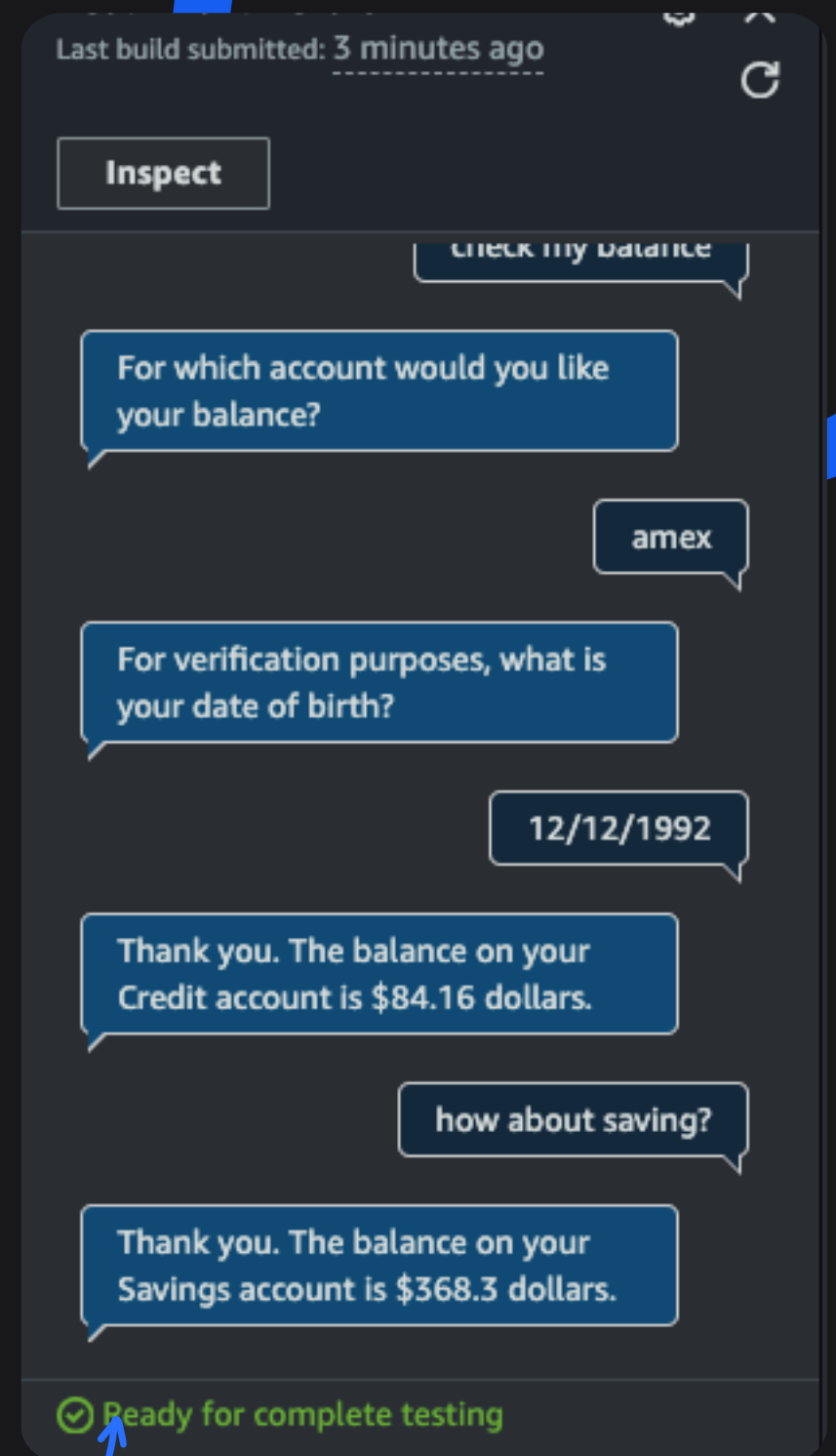
Provide a default value, #value for a context value, or [variable] for session variable.

San Diego, #ContextTag.SlotName, [SessionAttributeName]

Add default

# 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 started by asking the bot to "check my balance", "amex" and provided my birthday when prompted.
- Then, I said "how about saving" to trigger the follow-up intent for checking another account without needing to repeat my birthday.
- If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context...The bot wouldn't answer directly to "how about saving" because it wouldn't have my birthday information.
- It might ask for clarification or request to start over with "check balance" again.



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

# My Key Learnings

**01**

Context tags are used in Amazon Lex to store and retrieve specific information across different conversation parts. They help users avoid repeating information by remembering details from previous interactions.

**02**

Input context tags check for existing information before an intent, while output context tags store information for later use in the conversation.

**03**

I created the input context in FollowupCheckBalance by specifying 'contextCheckBalance' as the input context tag, ensuring the intent retrieves previously stored user details, such as the date of birth.

**04**

FollowupCheckBalance requires the presence of the 'contextCheckBalance' output context from a previous CheckBalance intent to retrieve stored user details such as the date of birth, thus it cannot function successfully without prior context

**05**

I've learned the importance of designing conversational flows that anticipate user needs and preferences, ensuring a smoother and more personalized interaction with the chatbot.

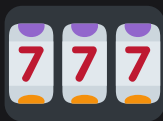
# Final thoughts...

- This project took me approximately 2 hours to complete. Writing documentation took me an additional 1 hour.
- Delete EVERYTHING at the end! Let's keep this project free :)
- One thing I didn't expect was...
- In the next phase of this project, I'm excited to level up my Lex bot one more time by creating an intent that can help users transfer funds between accounts. I'm also using AWS CloudFormation to recreate my bot in seconds!

# Part 5 : How I built a chatbot with Amazon Lex



with multiple slots!



I'd like to transfer money.

Which account would you like to transfer from?

checking

Which account are you transferring to?

savings

How much money would you like to transfer?

2000

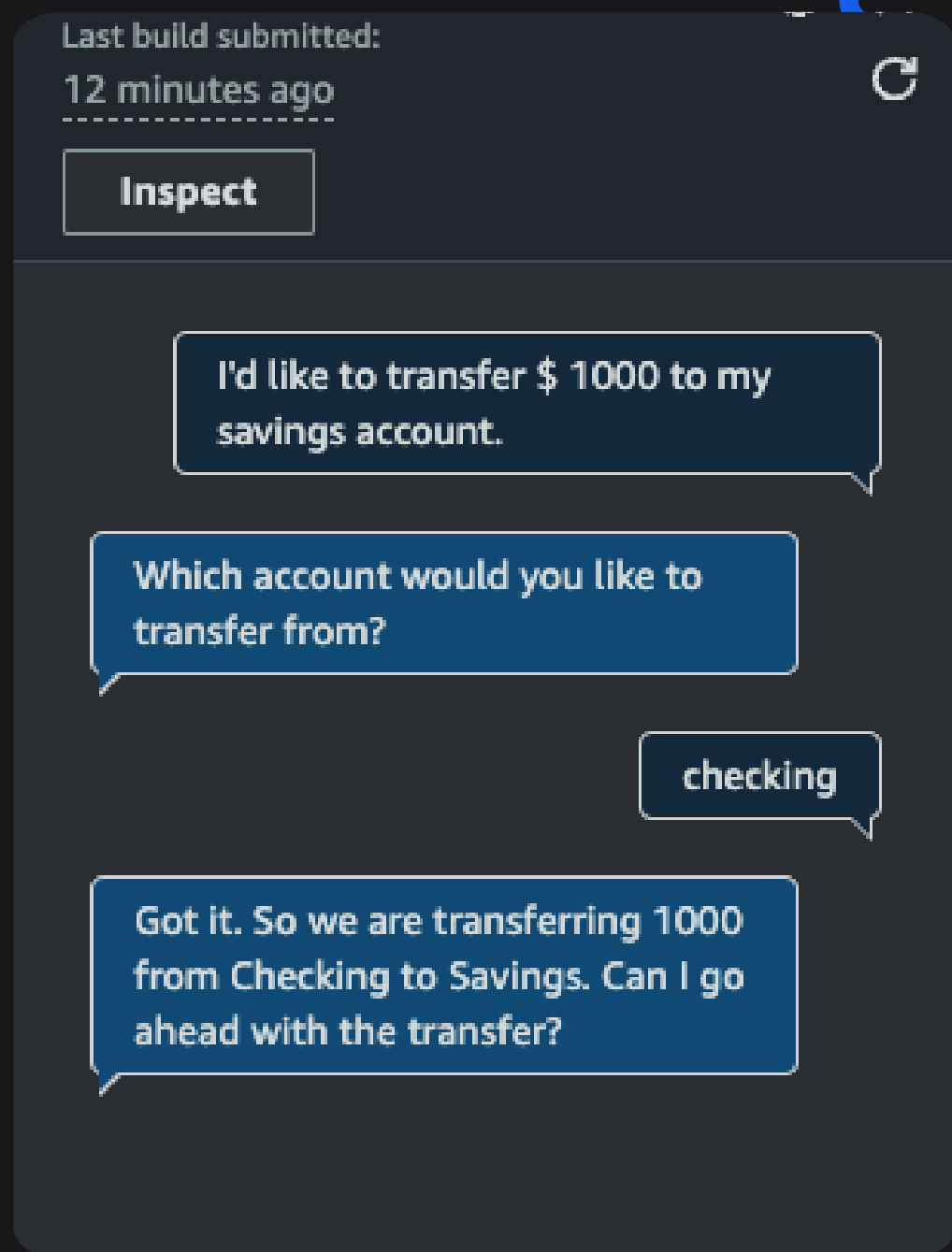
Got it. So we are transferring 2000

✓ Ready for complete testing

Type a message

# More slots!

- Slots in Amazon Lex serve as placeholders for specific user-provided information essential for executing actions or fulfilling requests within a chatbot interaction. For example, in BankerBot, the accountType slot categorizes different types of bank accounts such as Checking, Savings, or Credit.
- The final intent, TransferFunds, enables users to initiate and complete transfers of funds between specified accounts. This intent facilitates seamless transactions within the BankerBot, enhancing user convenience and banking functionality.
- In the TransferFunds intent, I utilized the accountType slot type twice : once for sourceAccountType and again for targetAccountType. This repetition ensures that the bot accurately identifies both the source from which funds will be transferred and the destination account.
- I also learnt how to create confirmation prompts. Confirmation prompts in TransferFunds are designed to verify transaction details before finalizing the transfer. They repeat back the specified transferAmount, sourceAccountType, and targetAccountType to the user, confirming the intended action. For instance, "Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I go ahead with the transfer?" This ensures clarity and reduces the risk of errors in financial transactions.



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

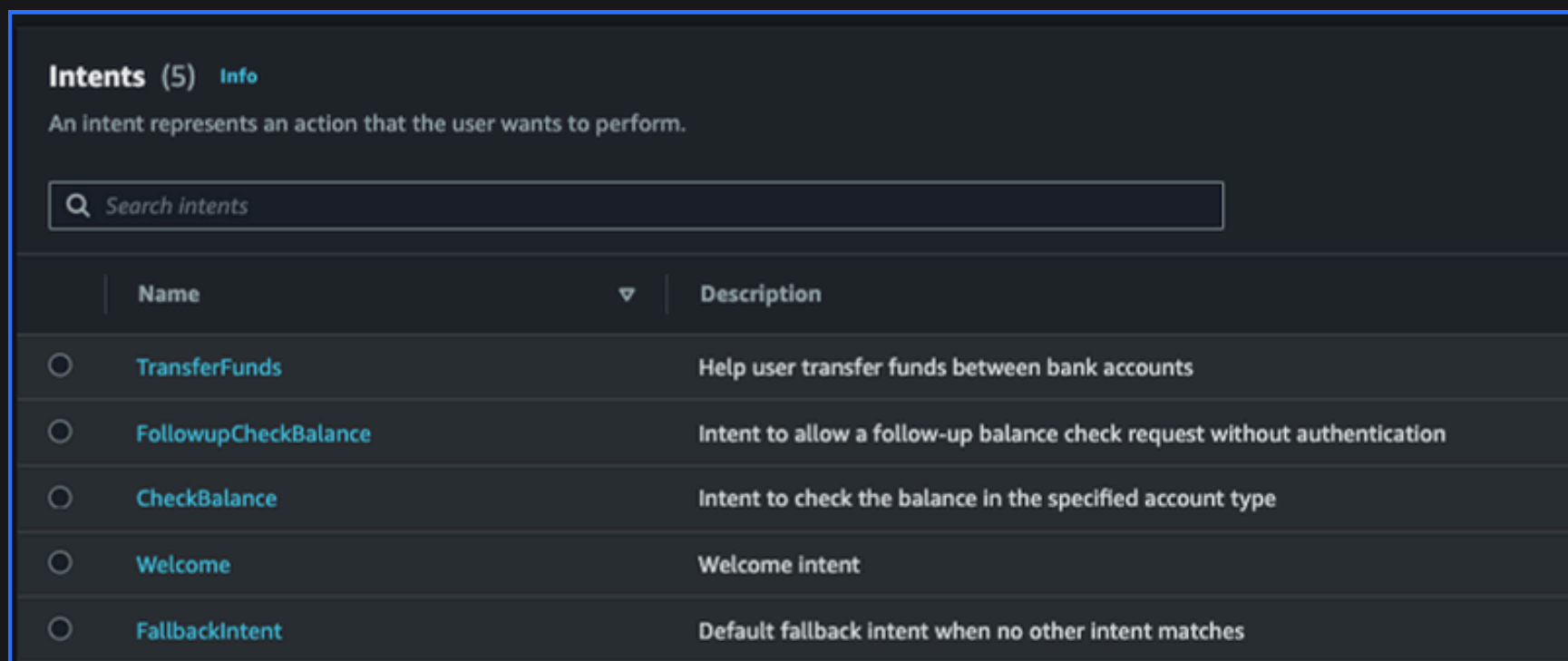


A LITTLE EXTRA...

# Deploying with CloudFormation

- AWS CloudFormation is a service that automates the provisioning and management of your AWS infrastructure.
- As an extension to this project, I learned how to deploy the entire BankerBot using a single CloudFormation stack.
- Doing this took me less time and effort compared to manually creating each resource.
- Something I learned from deploying with CloudFormation was the importance of infrastructure as code. It allows for a repeatable and consistent way to build and deploy my chatbot environment, but the YAML File isn't easy to write if you're not a programmer.

CloudFront  
deployed this  
for me!



Intents (5) <a href="#">Info</a>	
An intent represents an action that the user wants to perform.	
<input type="text" value="Search intents"/>	
Name	Description
<input type="radio"/> <a href="#">TransferFunds</a>	Help user transfer funds between bank accounts
<input type="radio"/> <a href="#">FollowupCheckBalance</a>	Intent to allow a follow-up balance check request without authentication
<input type="radio"/> <a href="#">CheckBalance</a>	Intent to check the balance in the specified account type
<input type="radio"/> <a href="#">Welcome</a>	Welcome intent
<input type="radio"/> <a href="#">FallbackIntent</a>	Default fallback intent when no other intent matches

# My Key Learnings

**01**

I used the `accountType` slot type twice to specify both the source and target accounts for transfers, ensuring clear distinctions for each step of the transaction.

**02**

Confirmation prompts are messages that verify user intent by restating information for confirmation, helping prevent errors. They ensure that the bot proceeds with accurate actions based on user input.

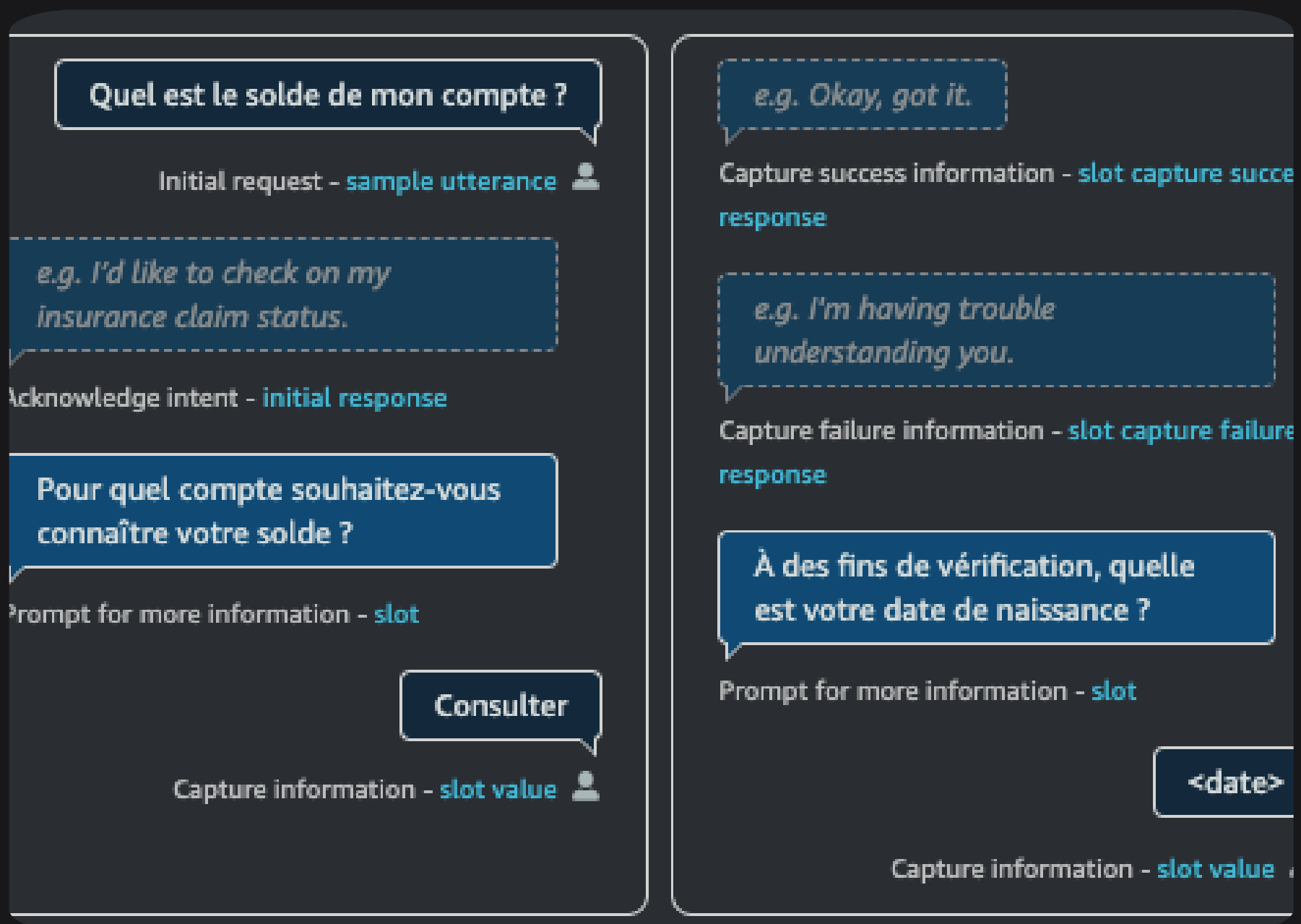
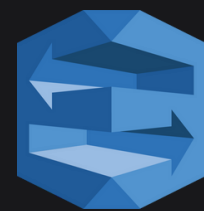
**03**

CloudFormation is an AWS service that automates the deployment of resources using templates. It created the entire BankerBot infrastructure, including intents, slots, and other configurations.

**04**

I learned the importance of using clear, distinct slot names and how structured automation can enhance deployment efficiency and consistency.

# How I built a French chatbot with Amazon Lex



## Step #1: Set up your Lex chatbot

**Creation method**

☒ **Create a blank bot**  
Create a basic bot with no preconfigured languages, intents, and slot types.

☐ **Start with an example**  
An example bot has preconfigured languages, intents, and slot types. You can change these settings.

☐ **Start with transcripts**  
Automatically generate intents from conversation transcripts that you upload. Only English (US) language is available when starting with a transcript.

**Bot configuration**

**Bot name**  
  
Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, \_

**Description - optional**  
This description appears on bot list page. It can help you identify the purpose of your bot.  
  
Maximum 200 characters.

If French, only one voice is available ...



**▼ Language: French (CA)**

**Select language**

**Description - optional**  
  
Maximum 200 characters.

**Voice interaction**  
The text-to-speech voice that your bot uses to interact with users.

**Voice sample**

**Intent classification confidence score threshold**  
  
Min: 0.00, max: 1.00.

Step #2: Create your first intent

Sample utterances (4) Info

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

Filter

Sort by added (ascending)

Preview

Plain text

Salut

Bonjour

J'ai besoin d'aide

Pouvez-vous m'aider ?

Closing response Info

You can define the response when closing the intent.

Active

Response sent to the user after the intent is fulfilled

Message: Bonjour ! Je suis BB, le Bot Bancaire. Comment puis-je vous aider aujourd'hui ?

Message group Info

You can define a text message group to respond using plain text.

Message

Bonjour ! Je suis BB, le Bot Bancaire. Comment puis-je vous aider aujourd'hui ?

Variations - optional

Bonjour ! Je suis votre assistant virtuel. Comment puis-je vous aider aujourd'hui ?

Bonjour ! Je suis votre assistant virtuel. Avez vous des questions concernant votre compte bancaire ou souhaitez vo

More response options

Add custom payloads, SSML, and card groups.

Nous contacter et prendre ren: x

mabanque.bnpparibas/fr/nous-contacter/nous-contacter

Particuliers Banque privée Professionnels Entreprises FR

BNP PARIBAS La banque d'un monde qui change

Devenir client Accéder à mes comptes

Gérer ses comptes Emprunter Assurer et sécuriser Épargner Investir en bourse Nous contacter Vous et vos besoins Ma banque et moi Ma banque s'engage

NOUS CONTACTER

PAR TÉLÉPHONE

VOUS AVEZ UNE QUESTION ? Sélectionnez la thématique

SUR NOS OFFRES ET SERVICES

CHATTEZ

Bonjour, je suis Telmi, votre assistant virtuel

Comment puis-je vous aider ?

Poser une question

Trouver une agence

Consulter mon solde

Écrivez un message...

To be honest, my bank has a mobile application, but I never use the chatbot available at the official webpage ...

👋 Step #3: Manage FallbackIntent

Closing responseInfo

Active

You can define the response when closing the intent.

▼ Response sent to the user after the intent is fulfilled

Message: Désolé, j'ai du mal à comprendre. Pouvez-vous décrire ce que vous aimeriez faire en quelques mots ? Je peux vous aider à trouver le solde de votre compte, faire un transfert et effectuer un paiement.

▼ Message groupInfo

You can define a text message group to respond using plain text.

Message

Désolé, j'ai du mal à comprendre. Pouvez-vous décrire ce que vous aimeriez faire en quelques mots ? Je peux vous a

▼ Variations - optional

Je n'arrive pas à trouver une réponse, merci de contacter votre agence ou d'envoyer un message à votre conseiller b

Afin de pouvoir vous aidez, assurez vous que votre message concerne bien le solde de votre compte, transférer des

More response options

Add custom payloads, SSML, and card groups.

► Set values

-

Next step in conversation

End conversation

+ Add conditional branching

▼ Code hooks - optionalInfo

## Step #4: Create a custom slot for account types

### Slot value resolution

Amazon Lex resolves the slot values in an utterance to only the values you provide, or it expands the resolution to related or similar values.

☐ Expand values (default)  
Values used as training data.

☒ Restrict to slot values  
Use only values provided.

### Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Consulter

Tab or ; or enter return for new value



consultez



Epargne

Tab or ; or enter return for new value



épargne



compte épargne



Crédit

Tab or ; or enter return for new value



credit



visa



mastercard



amex



american express



carte de crédit



carte bancaire



Value

Tab or ; or enter return for new value

Add value



## Step #5: Create the CheckBalance intent

▼ Intent details

Info

Intent name

CheckBalance

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, \_

Description - optional

Intention qui permet de vérifier le solde dans le type de compte spécifié.

Maximum 200 characters.

ID: ATSP4G21QB

Sample utterances (7)

Info

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

Filter

Sort by added (ascending) ▼

Preview

Plain text

Quel est le solde de mon compte ?

Vérifiez le solde de mon compte

Quel est le solde de mon compte {accountType} ?

Combien ai-je dans {accountType} ?

Je veux vérifier le solde

▼ Slots (2) - optional

Info

Add slot

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

Filter

Prompt for slot: accountType

Message: Pour quel compte souhaitez-vous connaître ...

Slot type

accountType

×

Prompt for slot: dateOfBirth

Message: À des fins de vérification, quelle est votre dat...

Slot type

AMAZON.Date

×



# Step #6 : Create Your AWS Lambda function

Lambda > Functions > Create function

Create function 

Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Container image

Select a container image to deploy for your function.

Basic information

Function name

Enter a name that describes the purpose of your function.

BankingBotFrench

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime 

Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.12

Architecture 

Info

Choose the instruction set architecture you want for your function code.

x86\_64

arm64

Permissions 

Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Change default execution role

Advanced settings

The code is the exact same but I have change some sentences in French

Successfully updated the function BankingBotFrench.

BankingBotFrench

Layers (0)

+ Add trigger

+ Add destination

Last modified 1 minute ago

Function ARN arn:aws:lambda:eu-west-2:381492025576:function:BankingBotFrench

Function URL 

Info

Code

Test

Monitor

Configuration

Aliases

Versions

Code source 

Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anywhere (Ctrl P)

Environment

BankingBotFrench

lambda\_function.py

67 def followupCheckBalance(intent\_request):

68 session\_attributes = get\_session\_attributes(intent\_request)

69 slots = get\_slots(intent\_request)

70 account = get\_slot(intent\_request, 'accountType')

71 #The account balance in this case is a random number

72 #Here is where you could query a system to get this information

73 balance = str(random\_num())

74 text = "Merci. Le solde de votre compte " + account + " est de " + balance + " dollars."

75 message = {

76 'contentType': 'PlainText',

77 'content': text

78 }

79 fulfillment state = "Fulfilled"

80

98.1 Python Spaces: 4

Code properties 

Info

Package size 987 byte

SHA256 hash Xv1XUaq70pocAz+d0eZfgn0HCFVRN2MMdxstFRa50M=


Last modified July 1, 2024 at 09:43 PM GMT+2

## Step #7: Connect AWS Lambda with Amazon Lex

Lex > Bots > Bot: BankerBot > Aliases

### Aliases (1) [Info](#)

An alias points to a specific version of your bot. With an alias, you can update the bot version that your client applications use.



	Alias name	Created
<input type="radio"/>	<a href="#">TestBotAlias</a>	5 hours ago

Lex > Bots > Bot: BankerBot > Aliases > [Alias: TestBotAlias](#) > Alias language support: French (CA)

## Alias language support: French (CA)

### ▼ Lambda function - *optional*


The Lambda function is invoked for initialization, validation, and fulfillment.

Source

▼

Lambda function version or alias

▼

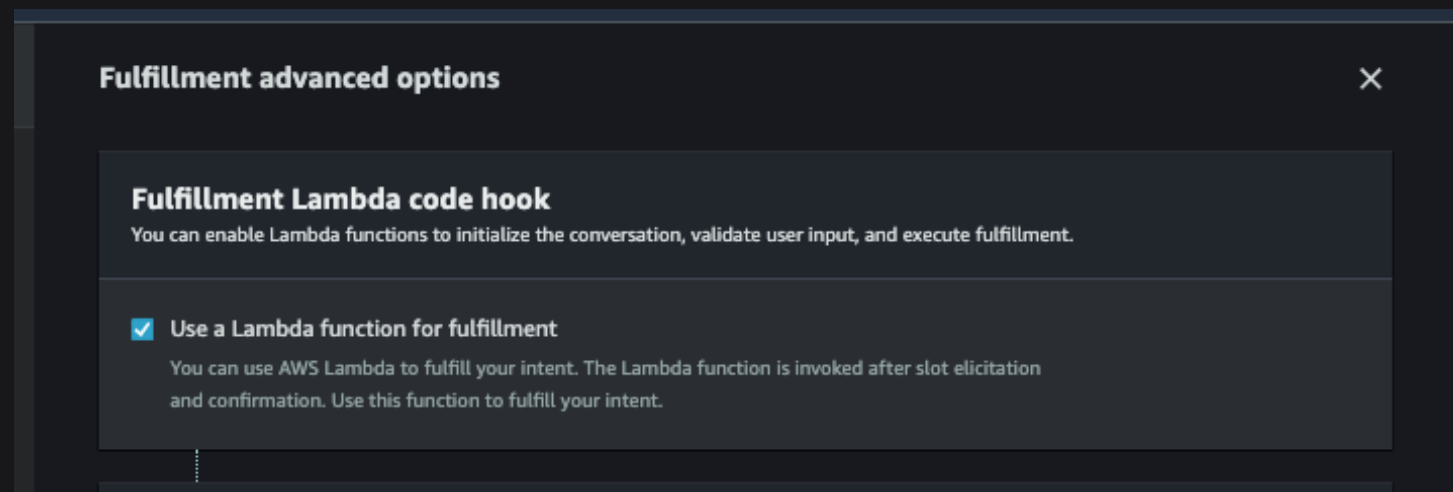
[Learn more about Lambda](#) 

Cancel

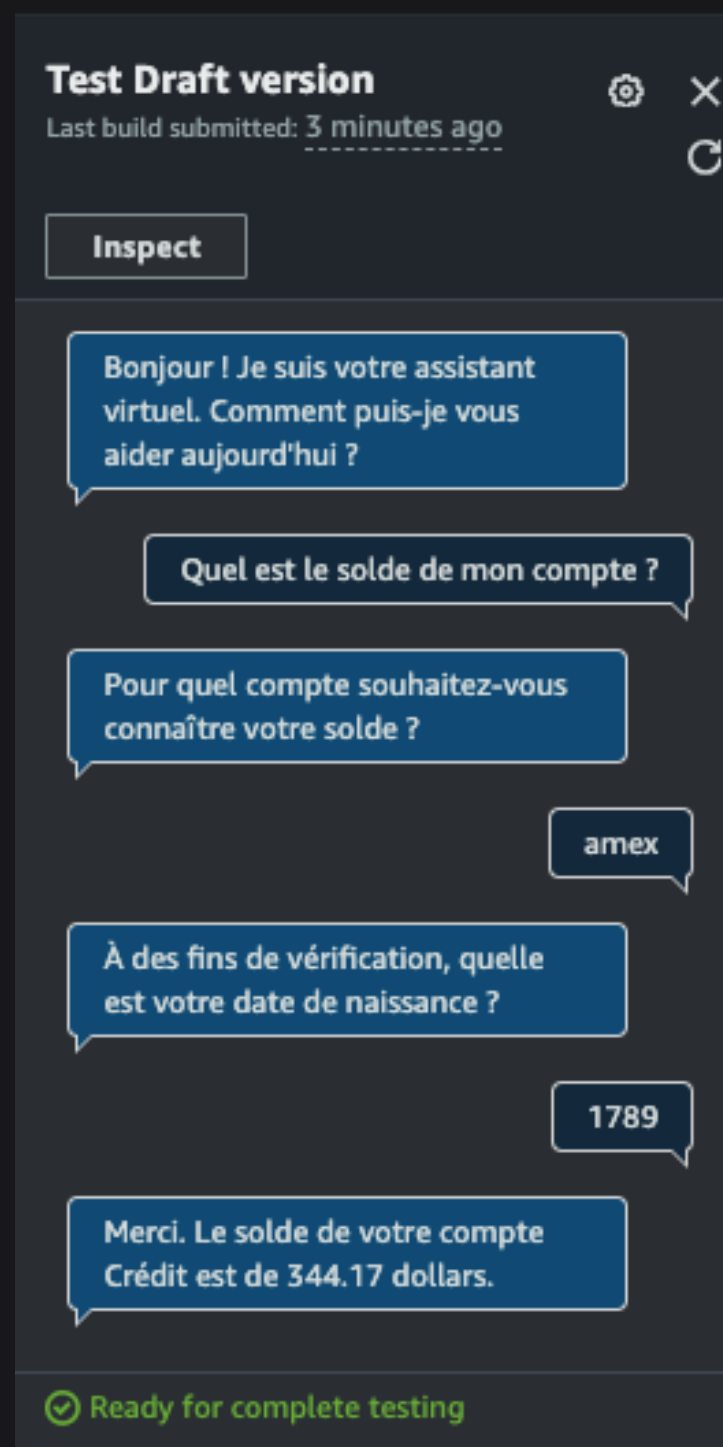
Save

## 🤝 Step #8: Connect your CheckBalance intent with your Lambda function

Navigate to your **CheckBalance** intent.  
Scroll down to your **Fulfilment** panel.



🇫🇷 The French bot is now able to return (random) bank balance figures!



## 🤝 Step #9 Create the new TransferFunds intent

Intent name

TransferFunds

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, \_

Description - *optional*

Aider les clients à effectuer des transferts entre deux comptes bancaires

Maximum 200 characters.

ID: ZPOSVOFPHJ

Sample utterances (7)

Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the samples. The priority order of the sample utterances is not used to determine intent classification output.

Q Filter

Sort by added (ascending) ▼

Preview

Plain text

Puis-je effectuer un virement ?

Je veux transférer des fonds

Je veux faire un virement

J'aimerais transférer {transferAmount} de {sourceAccountType} à {targetAccountType}

Puis-je transférer {transferAmount} vers mon {targetAccountType}

Pouvez-vous m'aider à faire un virement ?

J'ai besoin de faire un virement

▼ Slots (3) - *optional*

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

Add slot

Q Filter

▶ Prompt for slot: sourceAccountType

Message: De quel compte souhaitez-vous transférer ?

Slot type

accountType

×

▶ Prompt for slot: targetAccountType

Message: Sur quel compte souhaitez-vous effectuer le ...

Slot type

accountType

×

▶ Prompt for slot: transferAmount

Message: Quel montant souhaitez-vous transférer ?

Slot type

AMAZON.Number

×

## 🤝 Step #9 Create the new TransferFunds intent

### Confirmation

Active

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

▼ Prompts to confirm the intent

Message: *Merci. Nous allons donc transférer {transferAmount} de {sourceAccountType} vers {targetAccountType}. Puis-je pro*

Responses sent when the user declines the intent

Message: *Le transfert a été annulé.*

Confirmation prompt

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

Merci. Nous allons donc transférer {transferAmount} de {sourceAccountType} vers {targetAccountType}. Puis-je pro

Decline response

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

Le transfert a été annulé.

Advanced options

Configure confirmation prompts and decline responses.

### Closing response

Active

You can define the response when closing the intent.

▼ Response sent to the user after the intent is fulfilled

Message: *Le transfert est terminé. {transferAmount} devrait maintenant être disponible sur votre compte {targetAccountType}.*

▼ Message group

You can define a text message group to respond using plain text.

Message

ransfert est terminé. {transferAmount} devrait maintenant être disponible sur votre compte {targetAccountType}.

► Variations - optional

More response options

Add custom payloads, SSML, and card groups.

## 🤝 Step #9 Create the new TransferFunds intent

Puis-je effectuer un virement ?

De quel compte souhaitez-vous transférer ?

compte courant

Sur quel compte souhaitez-vous effectuer le transfère ?

compte épargne

Quel montant souhaitez-vous transférer ?

1233

Merci. Nous allons donc transférer 1233 de Crédit vers Epargne. Puis-je procéder au transfert ?

oui

Le transfert est terminé. 1233 devrait maintenant être disponible sur votre compte Epargne.

This time, when we ask for a new transfer, the bot won't ask to the user the birthday :

Le transfert est terminé. 1233 devrait maintenant être disponible sur votre compte Epargne.

Je veux transférer des fonds

De quel compte souhaitez-vous transférer ?

compte courant

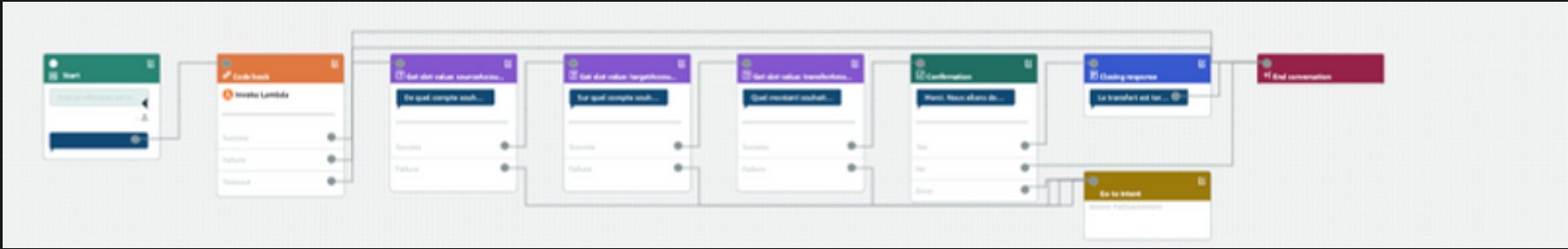
Sur quel compte souhaitez-vous effectuer le transfère ?

compte épargne

Quel montant souhaitez-vous transférer ?

# Final thought

Next time, I will try the visual builder



# Find this helpful?



Like this post

yes!



Leave a comment



Save for later



Let's connect!

**pssst...** if you want to get this free project guide and documentation template, **check out NextWork!**