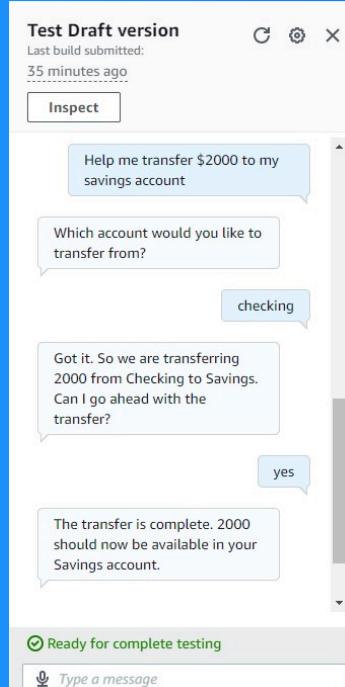




# Build a Chatbot with Multiple Slots



Nikhil Bhan



# Introducing Today's Project!

## What is Amazon Lex?

Amazon Lex is a service that allows users to develop chatbot applications that are capable of understanding and responding to human language, whether it's voice or text

## How I used Amazon Lex in this project

In this project I used Amazon Lex to create BankerBot and provided the user with the ability to transfer money from one account to another by using the same slot type; I learned how to create a CloudFormation Stack to provision multiple resources!

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

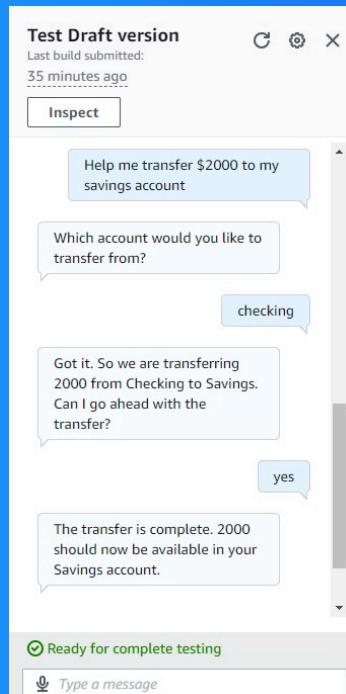
I wasn't expecting to get a permissions error when I was checking the account balance from my chatbot; this occurred while I was learning about CloudFormation and testing the resources from my Stack. I learned to do some troubleshooting from this!

## This project took me...

This project took me 1 hour and 30 minutes to complete, about 10 minutes to write up my documentation. The project took longer since I was troubleshooting the error I had with Lambda and my chatbot when I used CloudFormation to create my Stack.

# TransferFunds

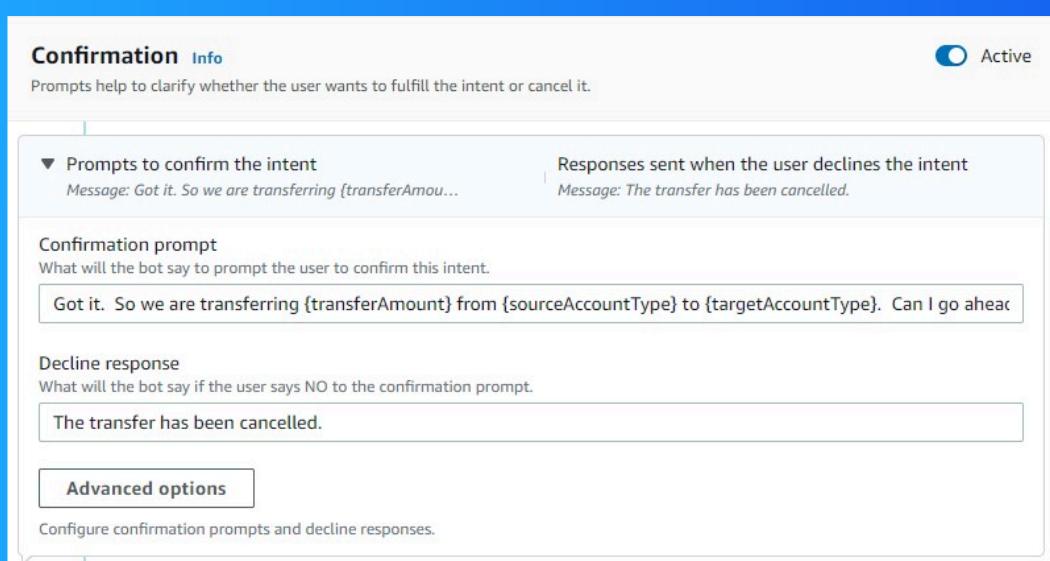
An intent I created for my chatbot was TransferFunds, which will transfer money from a bank account type to another account type that the user provides in the conversation.



# Using multiple slots

For this intent, I had to use the same slot type twice. This is because the user provides two bank accounts; one account is the source where the chatbot will take money and the other account is the target where the chatbot will transfer the money

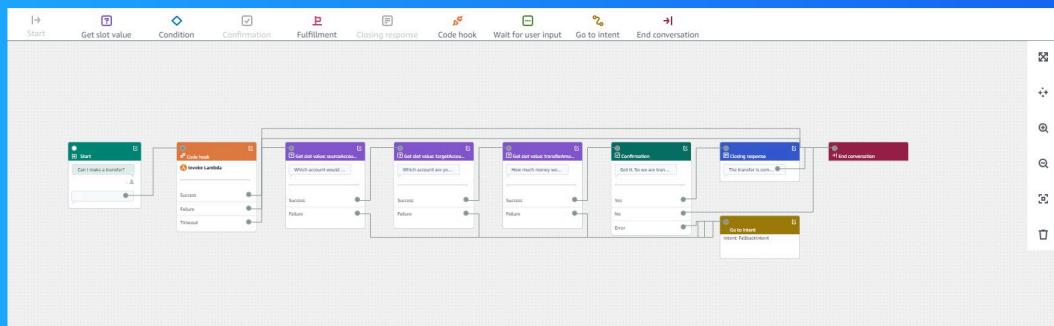
I also learnt how to create confirmation prompts, which are sent by the chatbot to verify with the user if they're sure they want the intent to be fulfilled or if they want to cancel it



# Exploring Lex features

Lex also has a special conversation flow feature that provides a visualization between the user and the chatbot. The user can navigate with different dialog nodes and the flow gets created automatically as changes are made in the intent.

You could also set up your intent using a visual builder! A visual builder shows the user a graphic interface where they have the ability to see their intent and modify the conversation flow by simply dragging and dropping blocks.



# AWS CloudFormation

AWS CloudFormation is a service that makes it fast and easy to set up AWS resources into a Stack; it's an Infrastructure as a Code service that uses a file that describes the resources and dependencies that need to be created.

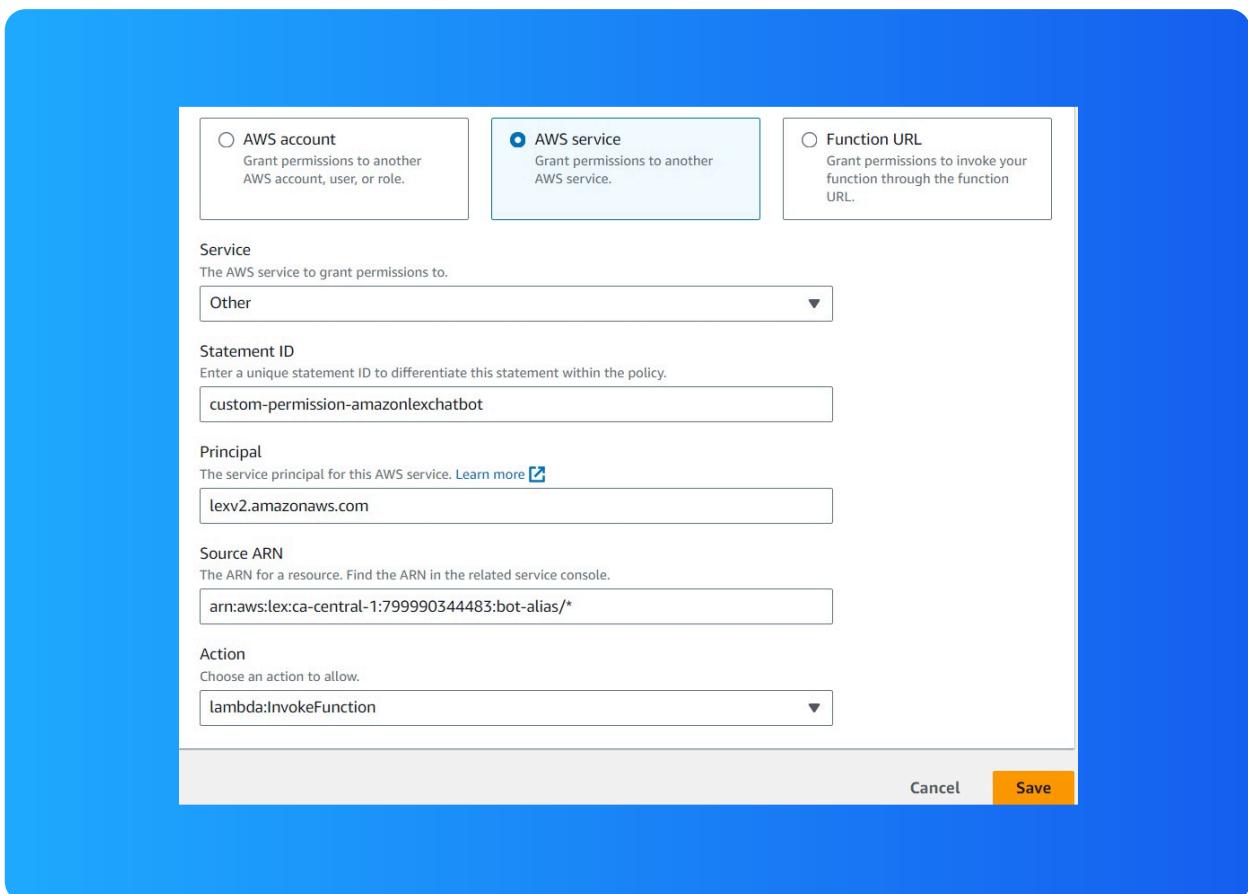
I used CloudFormation to deploy a chatbot and all the depending resources (aliases, intents, Lambda functions) by creating a Stack

Intents (5) <a href="#">Info</a>		
An intent represents an action that the user wants to perform.		
<input type="button" value="Delete"/> <input type="button" value="Add intent"/> <a href="#">▼</a>		
Name	Description	Last edited
TransferFunds	Help user transfer funds between bank accounts	4 minutes ago
FollowupCheckBalance	Intent to allow a follow-up balance check request without authentication	4 minutes ago
CheckBalance	Intent to check the balance in the specified account type	4 minutes ago
Welcome	Welcome intent	4 minutes ago
FallbackIntent	Default fallback intent when no other intent matches	4 minutes ago

# The final result!

Re-building my bot with CloudFormation took me less than 10 minutes to create

There was an error after I deployed my bot! I fixed this permission issue by creating a BankingBotEnglish function and associated it to the TestBotAlias. I also added an invokeFunction permission to allow my TestBotAlias to communicate with it.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

