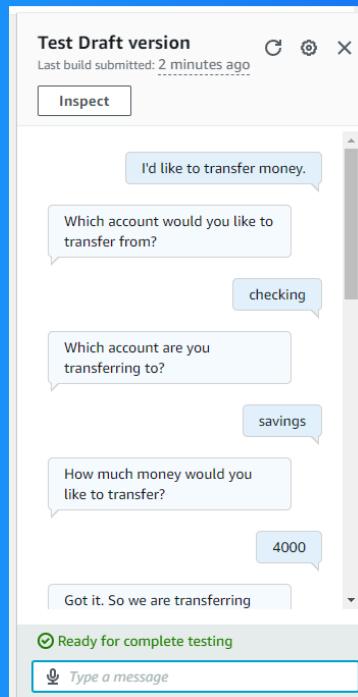




Build a Chatbot with Multiple Slots



Sai Prasanth Reddy



Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building chatbots using voice and text. It's useful for creating conversational interfaces, automating tasks, and integrating with AWS services, making it ideal for customer support and interactive applications.

How I used Amazon Lex in this project

I used Amazon Lex in today's project to build and deploy a chatbot for transferring funds and checking balances, configured intents and slots, integrated a Lambda function, and used CloudFormation to automate the deployment of the bot's resources.

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

One thing I didn't expect in this project was encountering a permission error after deploying the bot with CloudFormation, which required troubleshooting to update the IAM policy and grant the correct permissions for Lambda invocation.

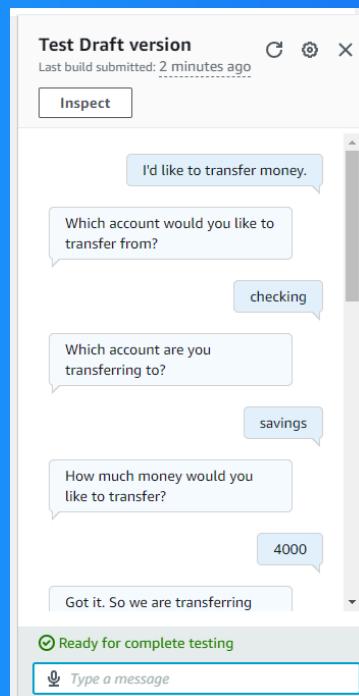
This project took me...

This project took me about an hour to complete, including setting up the chatbot, configuring intents, deploying resources with CloudFormation, troubleshooting issues, and testing the bot's functionality.



TransferFunds

An intent I created for my chatbot was TransferFunds, which will help users transfer funds between bank accounts by gathering the necessary details, such as the source and destination accounts, and the amount to transfer.

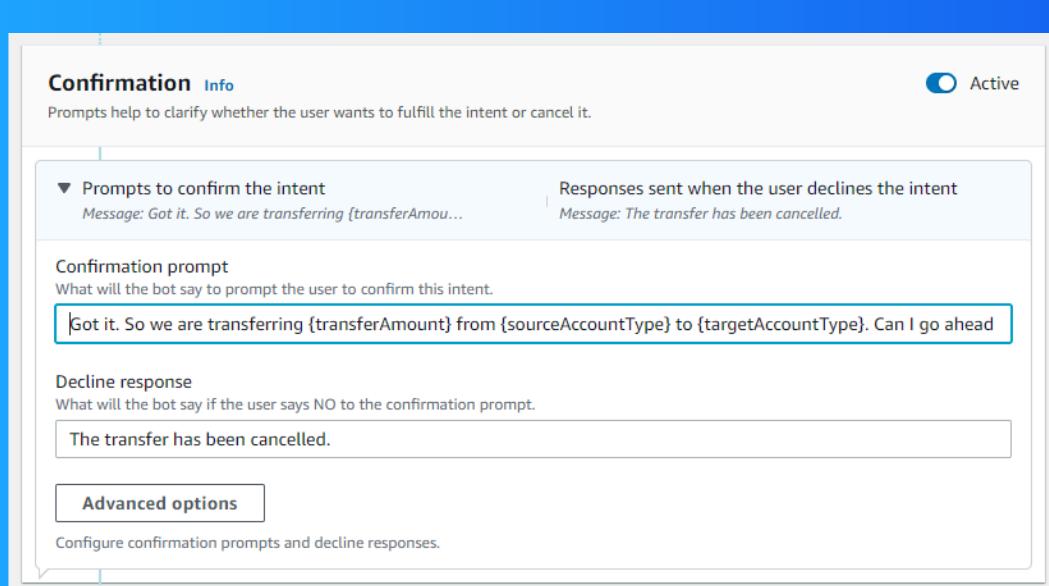




Using multiple slots

For this intent, I had to use the same slot type twice. This is because the chatbot needs to capture both the source and destination account types for the transfer, requiring the same slot type to differentiate between the two accounts.

I also learned how to create confirmation prompts, which are messages that ask users to confirm their action before proceeding, ensuring that the chatbot verifies key details like transfer amounts or account numbers before completing the request.

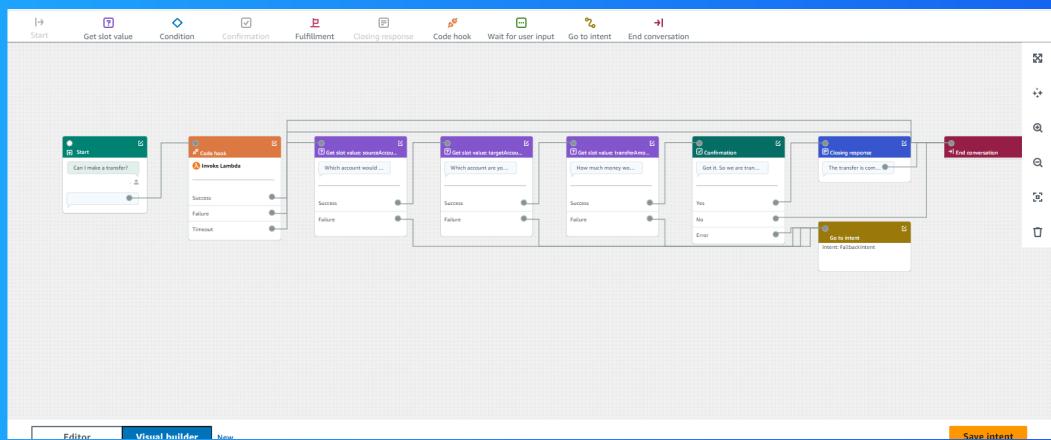




Exploring Lex features

Lex also has a special conversation flow feature that allows you to manage how a conversation progresses by guiding users through a structured interaction, ensuring that all necessary information is collected step-by-step before fulfilling an intent.

You could also set up your intent using a visual builder! A visual builder helps you create and manage your chatbot's conversation flow visually, making it easier to design and organize intents, slots, and responses without needing to write code.





AWS CloudFormation

AWS CloudFormation is a service that allows you to model and set up your AWS resources using templates, automating the process of provisioning and managing infrastructure. It simplifies resource management by treating infrastructure as code.

I used CloudFormation to deploy my BankerBot by automating the setup of all required resources, including the chatbot, intents, and Lambda functions, streamlining the deployment process through an infrastructure-as-code approach.

The screenshot shows a table of intents with the following data:

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



The final result!

Re-building my bot with CloudFormation took me about 30 minutes, as it involved creating and validating the template, then allowing CloudFormation to automatically provision the necessary resources for the bot.

There was an error after I deployed my bot! The error was a missing permission for the Lambda function. I fixed this by updating the IAM policy to grant the necessary permissions for the bot to invoke the Lambda function properly.

Add permissions

Edit policy statement

AWS account
Grant permissions to another AWS account, user, or role.

AWS service
Grant permissions to another AWS service.

Function URL
Grant permissions to invoke your function through the function URL.

Service
The AWS service to grant permissions to.
Other

Statement ID
Enter a unique statement ID to differentiate this statement within the policy.
my-custom-permission-amazonlexchatbot

Principal
The service principal for this AWS service. Learn more [↗](#)
lexv2.amazonaws.com

Source ARN
The ARN for a resource. Find the ARN in the related service console.
arn:aws:<service>:us-east-1:381491882143:<resource>

Action
Choose an action to allow.
lambda:InvokeFunction



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

