



NextWork.org

# Build a Chatbot with Multiple Slots



Kunal Parkhade

The screenshot shows the NextWork.org chatbot builder interface. At the top, a green banner indicates "Successfully built language English (US) in bot: BankerBot". Below this, the "Draft version" dropdown is set to "English (US)" and the status is "Successfully built". The main area displays a "Confirmation" intent card. It includes sections for "Prompts to confirm the intent" (with a message template "Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I") and "Responses sent when the user declines the intent" (with a message template "The transfer has been cancelled"). There are also "Confirmation prompt" and "Decline response" sections. On the right side, a "Test Draft version" window shows a conversation flow starting with "I want to transfer funds", followed by "Which account would you like to transfer from?", with options "Savings" and "Credit" shown. The next message is "Which account are you transferring to?", followed by "Ready for complete testing". A "Type a message" input field is at the bottom.



# Introducing Today's Project!

## What is Amazon Lex?

Amazon Lex is an AWS service to build chatbot that will take the user input in text or voice format and it will give response accordingly. It is useful because it provides end-to-end solution to build, publish, deploy and monitor chatbots.

## How I used Amazon Lex in this project

I used Amazon Lex to transfer the money

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

I didn't expect that I can rebuild my project in just 5 minutes using CloudFormation

## This project took me...

This project takes time around 1 hr



Kunal Parkhade  
NextWork Student

[NextWork.org](http://NextWork.org)

# TransferFunds

An intent I created for my chatbot was TransferFunds, which transfer money from one account to another account.

The screenshot shows the Microsoft Bot Framework Emulator interface. At the top, a green bar indicates "Successfully built language English (US) in bot: BankerBot". Below this, there are three tabs: "Draft version", "English (US)", and "Successfully built". The "Successfully built" tab is selected. On the left, a sidebar titled "Confirmation Info" contains sections for "Prompts to confirm the intent" (with a message template "Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I") and "Responses sent when the user declines the intent" (with a message template "The transfer has been cancelled"). Below these are "Confirmation prompt" (prompting the user to confirm the transfer) and "Decline response" (informing the user if they decline). At the bottom of the sidebar are "Advanced options", "Editor" (which is selected), "Visual builder", and "New...". On the right, a "Test Draft version" window is open, showing a conversation transcript. The user says "I want to transfer funds", followed by "Which account would you like to transfer from?", with a "Savings" button. Then, "Which account are you transferring to?", with a "Credit" button. A message at the bottom says "How much money would you like to transfer?". A green checkmark indicates "Ready for complete testing". There is also a "Type a message" input field and a "Save intent" button.



# Using multiple slots

For this intent, I had to use the same slot type twice. This is because I want account type for both source and target.

I also learnt how to create confirmation prompt, which are message after intent is fulfilled.

The screenshot shows the Microsoft Bot Framework builder interface. At the top, there is a green banner with the text "Successfully built language English (US) in bot: BankerBot". Below the banner, there are tabs for "Draft version" (selected), "English (US)", and a green button labeled "Successfully built". To the right, there is a status message "English (US) has not built changes." and two buttons: "Build" and "Test".

The main area is titled "Confirmation" with an "Info" link. It says "Prompts help to clarify whether the user wants to fulfill the intent or cancel it." A "Active" switch is turned on. Below this, there are two sections:

- Prompts to confirm the intent:** Message: Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I
- Responses sent when the user declines the intent:** Message: The transfer has been cancelled.

Below these sections, there are two more fields:

- Confirmation prompt:** What will the bot say to prompt the user to confirm this intent. Message: Got it. So we are transferring {transferAmount} from {sourceAccountType} to {targetAccountType}. Can I
- Decline response:** What will the bot say if the user says NO to the confirmation prompt. Message: The transfer has been cancelled.

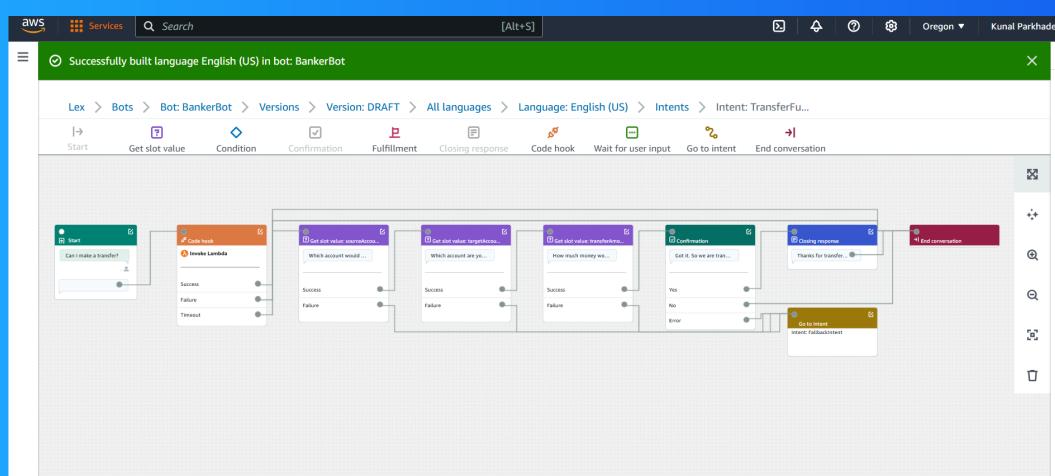
At the bottom, there is a "Advanced options" button with the sub-instruction "Configure confirmation prompts and decline responses."



# Exploring Lex features

Lex also has a special conversation flow feature that gives update as continue editing this intent. It shows every step in a conversation in a logical, chronological order.

You could also set up your intent using a visual builder! A visual builder is visual representation of the intent.





# AWS CloudFormation

AWS CloudFormation is service that gives an easy way to create and setup AWS resources.

I used CloudFormation to rebuild my Chatbot

The screenshot shows the AWS Lex console interface for managing intents. The top navigation bar includes links for Lex, Bots, Bot: Chatbot-Ba..., Versions, Version: Draft, All languages, Language: English (US), and Intents. Below the navigation are buttons for Draft version, English (US), Successfully built (highlighted in green), Build, and Test. A message indicates "English (US) has not built changes." The main content area is titled "Intents (5) Info" and contains a search bar and pagination controls. A table lists five intents:

Name	Description	Last edited
TransferFunds	Help user transfer funds between bank accounts	2 minutes ago
FollowupCheckBalance	Intent to allow a follow-up balance check request without authentication	2 minutes ago
CheckBalance	Intent to check the balance in the specified account type	2 minutes ago
Welcome	Welcome intent	2 minutes ago
FallbackIntent	Default fallback intent when no other intent matches	2 minutes ago



# The final result!

Re-building my bot with CloudFormation took me around 5 minutes.

There was an error after I deployed my bot! The error was system is not responding. I fixed this by setting resource based policy.

The screenshot shows the 'Add permissions' dialog box from the AWS Lambda console. The path in the top navigation bar is: Lambda > Functions > testfunctionDELETETHis > Add permissions. The main title is 'Add permissions'. Below it is a section titled 'Edit policy statement' with three radio button options:

- AWS account: Grant permissions to another AWS account, user, or role.
- AWS service: Grant permissions to another AWS service. This option is selected.
- Function URL: Grant permissions to invoke your function through the function URL.

Below these options is a 'Service' field labeled 'The AWS service to grant permissions to.' A dropdown menu is open, showing 'Other' as the selected option. To the right of the dropdown is a small downward arrow icon.

Further down, there are fields for 'Statement ID' (containing 'my-custom-permission-amazonlexchatbot'), 'Principal' (containing 'lexv2.amazonaws.com'), 'Source ARN' (containing 'arn:aws:lex:us-west-2:471112976395:bot-alias/\*'), and 'Action' (containing 'lambda:InvokeFunction').

At the bottom right of the dialog box are two buttons: 'Cancel' and a highlighted orange 'Save' button.



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

