CHATBOT WITH AMAZON LEX

SEPTEMBER 3

Devamadhav Kattimuttathu

Table of Contents

Build a Chatbot with Amazon Lex

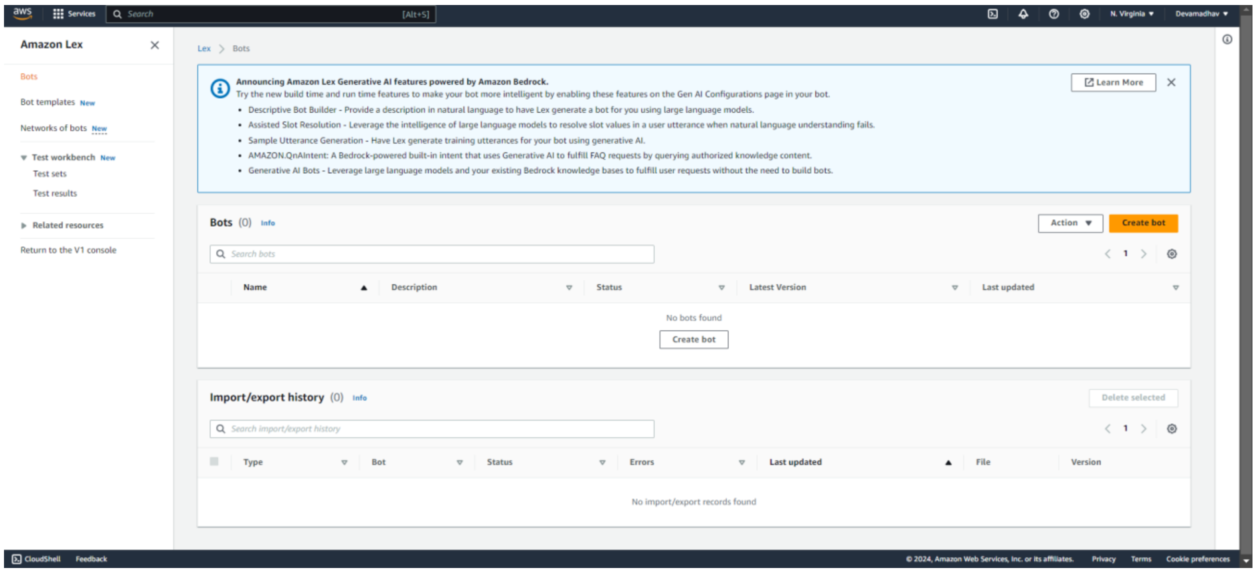
INTRODUCTION

In this lab, we are going to create and configure our very own Amazon Lex chatbot called BankerBot. Together, we'll walk through the entire process, from setting up the bot and defining its intents to integrating a Lambda function that will retrieve bank balance information. We'll start by building the bot from scratch, adding custom slot types, and fine-tuning the responses to ensure our bot can handle various user interactions smoothly. By the end of this lab, we'll have a fully functional chatbot capable of understanding user requests, handling errors, and even saving user details for more personalized conversations.

OBSERVATION AND SCREENSHOTS

Creating the BankerBot:

We started by logging into our AWS account and navigating to the Amazon Lex service. After switching to the Lex V2 console, we began the process of creating a new chatbot by selecting "Create bot" and choosing "Create a blank bot." We named our bot "BankerBot" and provided a brief description of its purpose. For IAM permissions, we opted to create a new role with basic Amazon Lex permissions, ensuring that our bot would be able to call Lambda functions later on. We confirmed that the bot did not fall under the Children's Online Privacy Protection Act (COPPA) and kept the default idle session timeout of 5 minutes. Moving forward, we selected the voice interaction settings, exploring different options before settling on a voice we liked. We kept the default intent classification confidence score threshold at 0.4, ensuring that our bot would only respond when it had a reasonable level of confidence in its understanding of user input. By the end of this step, we had successfully created the foundational settings for BankerBot.



Announcing Amazon Lex Generative AI features powered by Amazon Bedrock.
Try the new build time and run time features to make your bot more intelligent by enabling these features on the Gen AI Configurations page in your bot.

- Descriptive Bot Builder - Provide a description in natural language to have Lex generate a bot for you using large language models.
- Assisted Slot Resolution - Leverage the intelligence of large language models to resolve slot values in a user utterance when natural language understanding fails.
- Sample Utterance Generation - Have Lex generate training utterances for your bot using generative AI.
- AMAZON.QnAIntent: A Bedrock-powered built-in intent that uses Generative AI to fulfill FAQ requests by querying authorized knowledge content.
- Generative AI Bots - Leverage large language models and your existing Bedrock knowledge bases to fulfill user requests without the need to build bots.

Bots (0) Info

Name	Description	Status	Latest Version	Last updated

No bots found

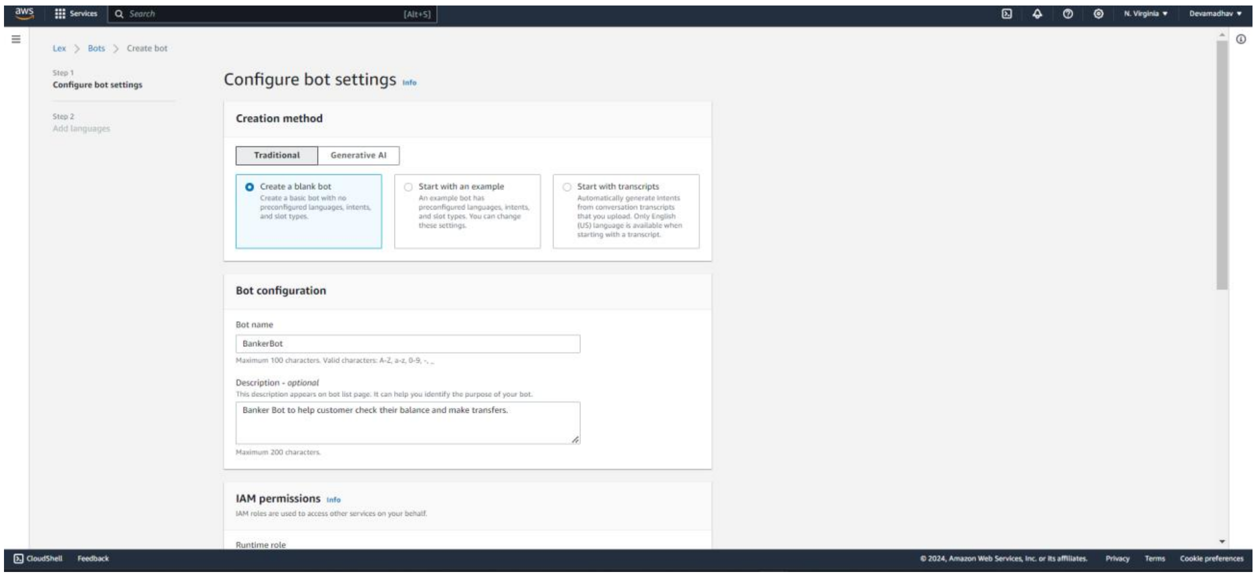
Import/export history (0) Info

Delete selected

Type	Bot	Status	Errors	Last updated	File	Version

No import/export records found

Figure 1 - Amazon Lex Console



Configure bot settings Info

Creation method

[✓ Traditional] Generative AI

[✓ 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

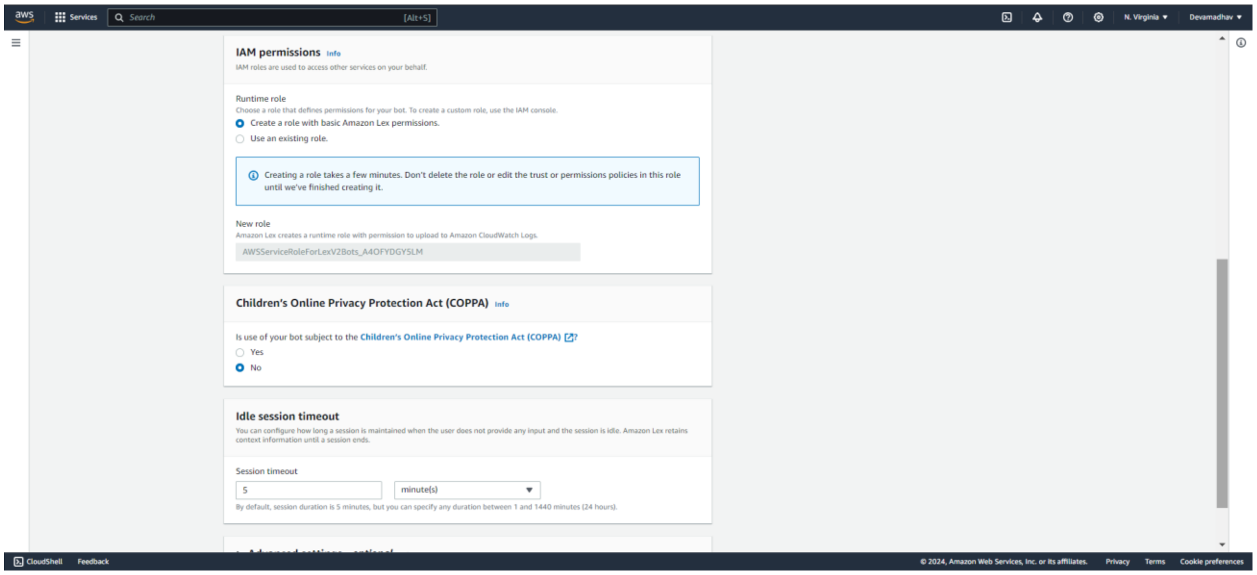
Bot name: BankerBot
Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

Description - *optional*
This description appears on bot list page. It can help you identify the purpose of your bot.
Banker Bot to help customer check their balance and make transfers.
Maximum 200 characters.

IAM permissions Info
IAM roles are used to access other services on your behalf.

Runtime role

Figure 2 - Creating a new chatbot


IAM permissions Info

IAM roles are used to access other services on your behalf.

Runtime role

Choose a role that defines permissions for your bot. To create a custom role, use the IAM console.

[✓ Create a role with basic Amazon Lex permissions.] Use an existing role.

Creating a role takes a few minutes. Don't delete the role or edit the trust or permissions policies in this role until we've finished creating it.

New role

Amazon Lex creates a runtime role with permission to upload to Amazon CloudWatch Logs.

AWSServiceRoleForLexV2Bots_A4OFYDGY5LM

Children's Online Privacy Protection Act (COPPA) Info

Is use of your bot subject to the Children's Online Privacy Protection Act (COPPA)?

Yes [✓ No]

Idle session timeout

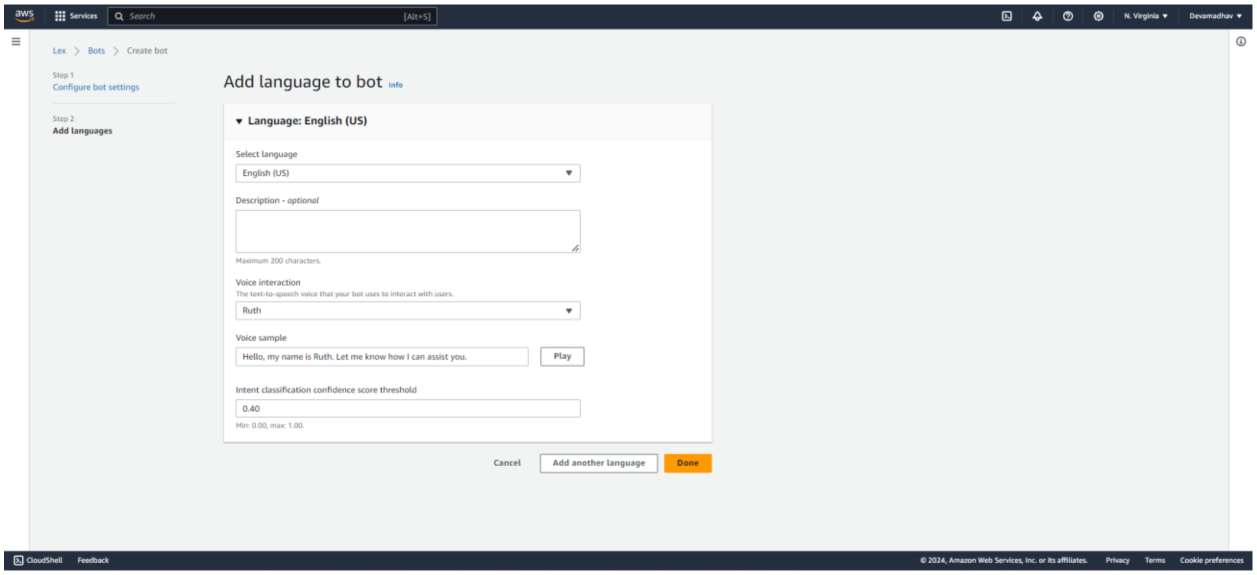
You can configure how long a session is maintained when the user does not provide any input and the session is idle. Amazon Lex retains context information until a session ends.

Session timeout: 5 minute(s)

By default, session duration is 5 minutes, but you can specify any duration between 1 and 1440 minutes (24 hours).

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 3 - Creating a chat bot- configuring bot settings


Lex > Bots > Create bot

Step 1 Configure bot settings

Step 2 Add languages

Add language to bot Info

Language: English (US)

Select language: English (US)

Description - *optional*

Maximum 200 characters.

Voice interaction

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

Ruth

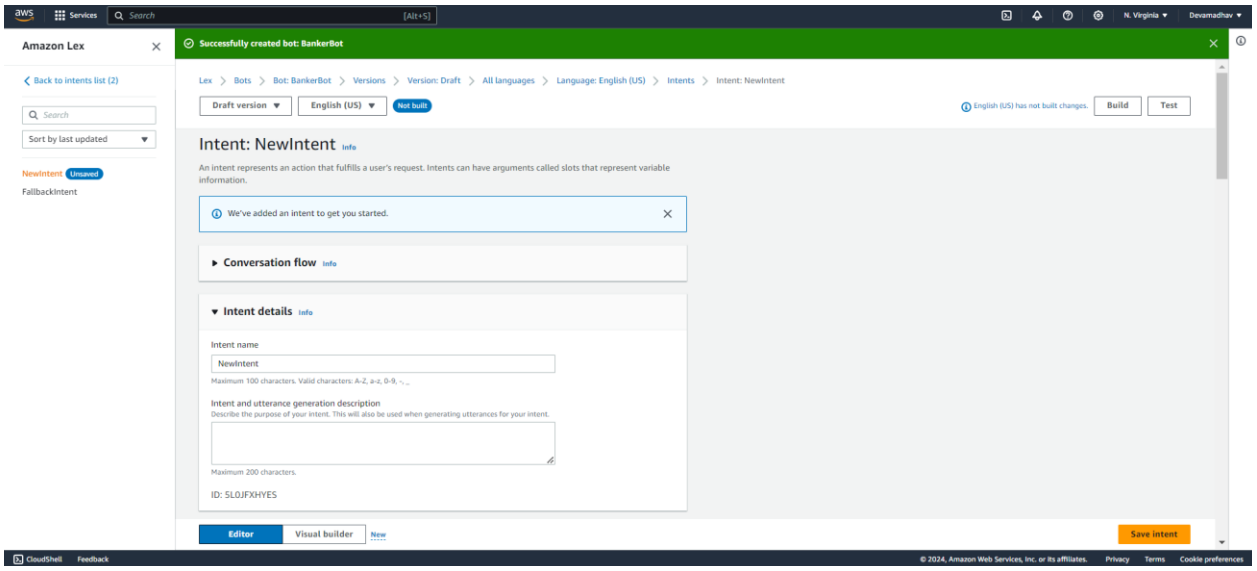
Voice sample: Hello, my name is Ruth. Let me know how I can assist you.

Intent classification confidence score threshold: 0.40

Min: 0.00, max: 1.00.

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 4 - Creating a chat bot- configuring bot settings



Successfully created bot: BankerBot

English (US) has not built changes.

Intent: NewIntent

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

We've added an intent to get you started.

Intent name: NewIntent

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

Intent and utterance generation description

Describe the purpose of your intent. This will also be used when generating utterances for your intent.

Maximum 200 characters.

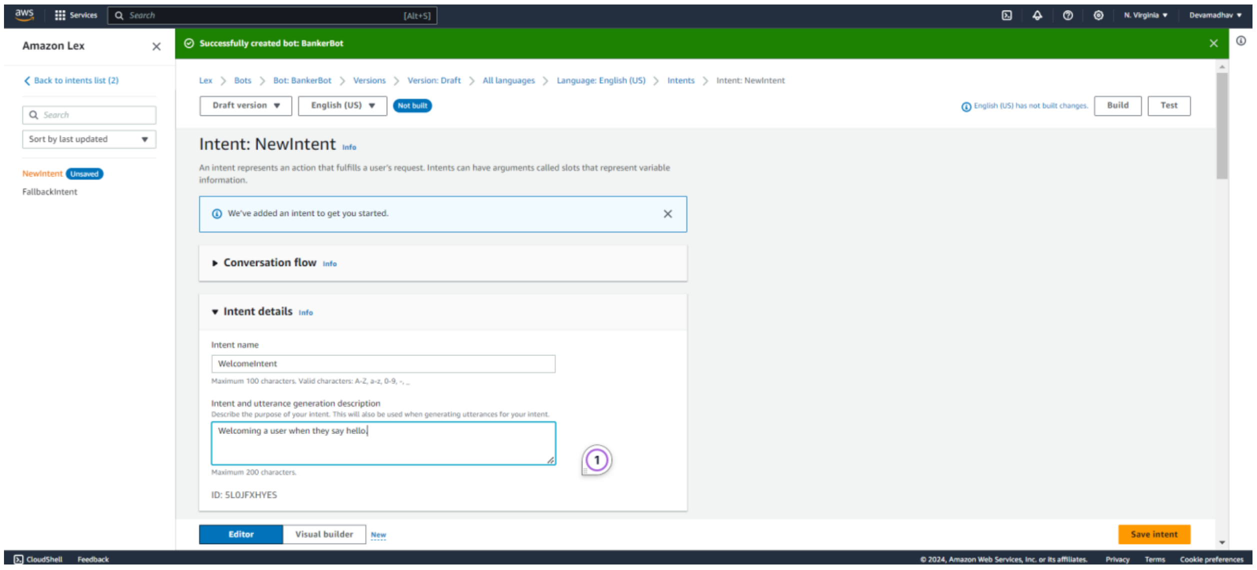
ID: 5L0JFXHYES

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 5 - Successfully created new bot

Setting Up the WelcomeIntent:

Understanding that there would be times when our bot might not understand a user's input, we customized the FallbackIntent. This intent serves as a catch-all for situations where the bot's confidence score is too low to match any defined intents. We navigated to the FallbackIntent and edited the default message to something more user-friendly and clear, such as "I'm sorry, I didn't understand that. Could you please rephrase?" We also added variations to this message to avoid repetitiveness, making the bot's responses feel more natural. By customizing the FallbackIntent, we aimed to improve the overall user experience, especially in cases where the bot couldn't accurately interpret the user's intent.



Successfully created bot: BankerBot

English (US) has not built changes.

Intent: NewIntent

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

We've added an intent to get you started.

Intent name: WelcomeIntent

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

Intent and utterance generation description

Describe the purpose of your intent. This will also be used when generating utterances for your intent.

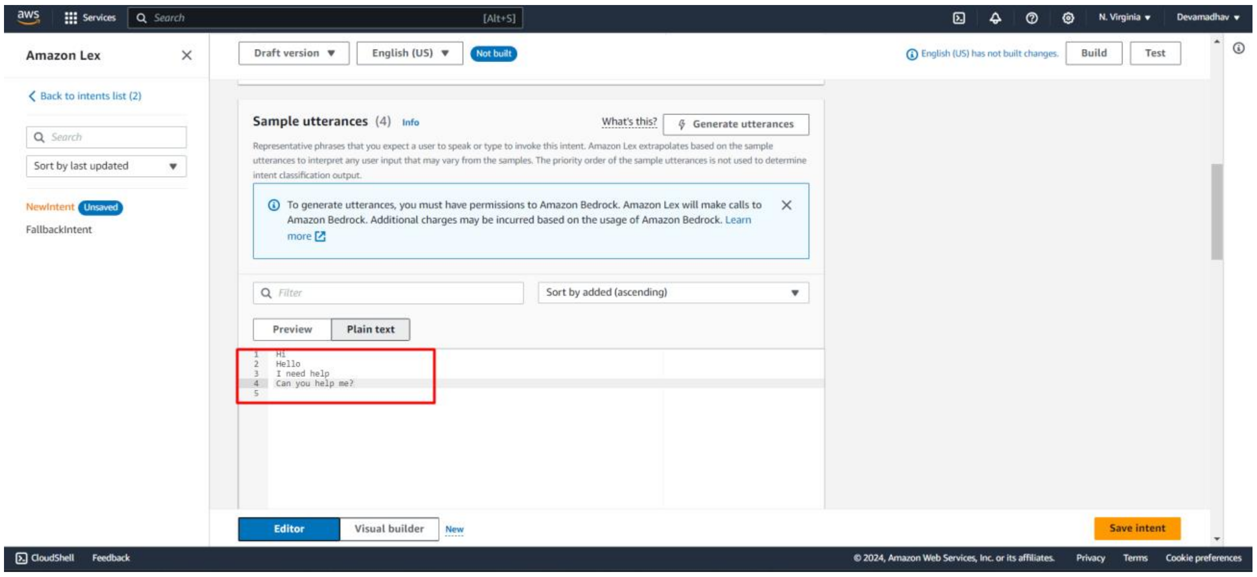
Welcoming a user when they say hello

Maximum 200 characters.

ID: 5L0JFXHYES

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 6 - Creating a new intent for the chat bot



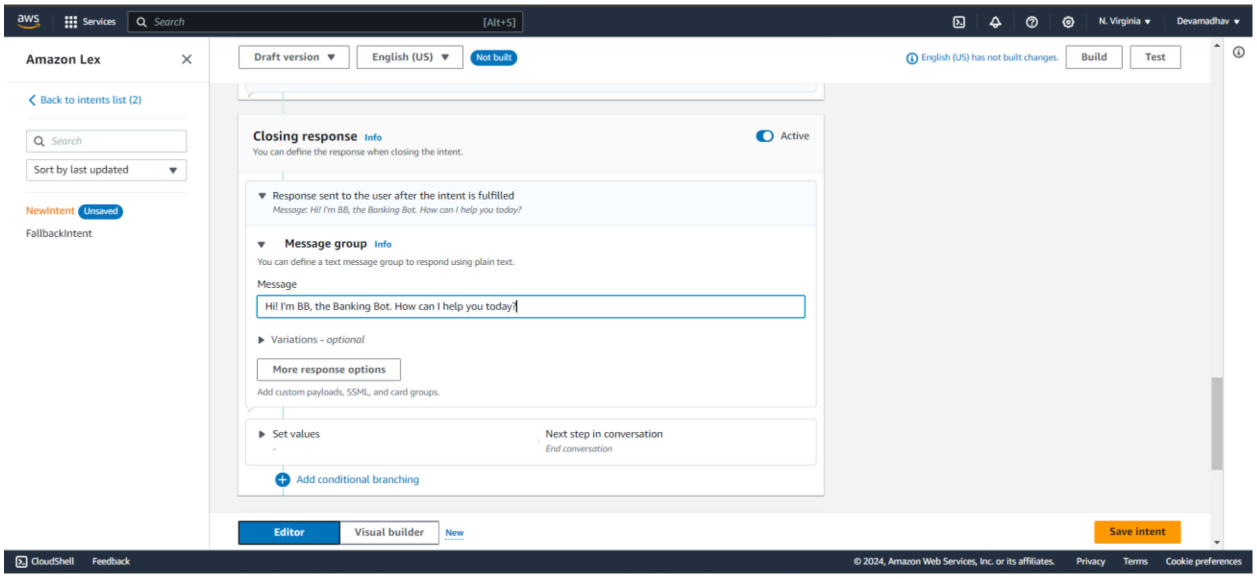
Sample utterances (4) Info

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.

To generate utterances, you must have permissions to Amazon Bedrock. Amazon Lex will make calls to Amazon Bedrock. Additional charges may be incurred based on the usage of Amazon Bedrock. Learn more

```
Hi
Hello
I need help
Can you help me?
```

Figure 7 - Configuring sample utterances



Closing response Info

You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled

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

Message group Info

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

Message

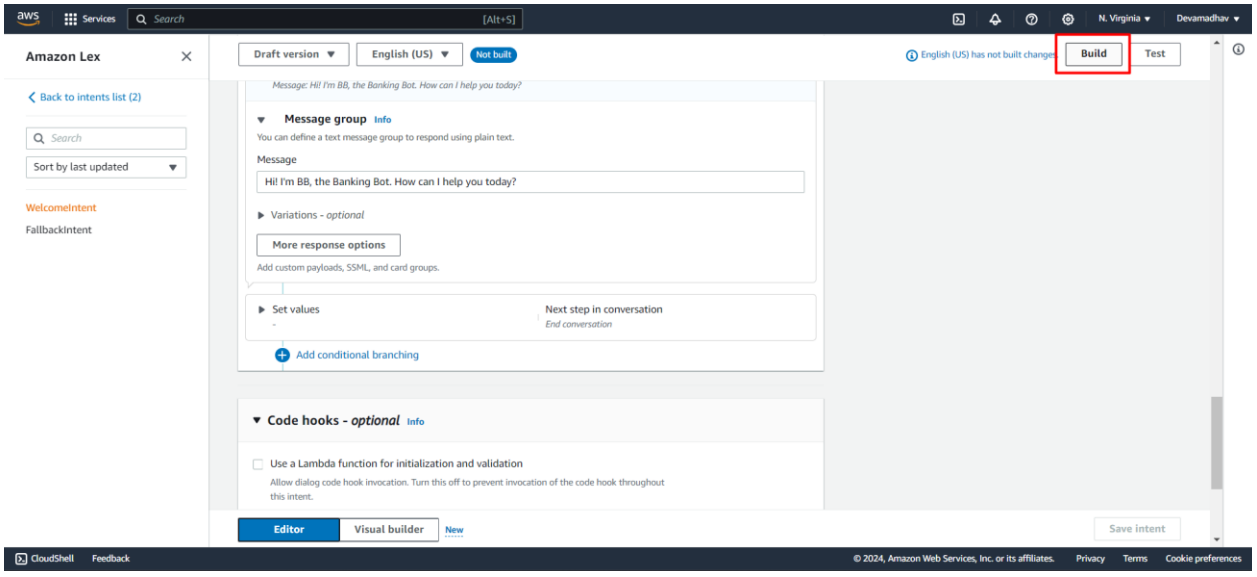
Hi! I'm BB, the Banking Bot. How can I help you today?

Add custom payloads, SSML, and card groups.

Set values: -

Next step in conversation: *End conversation*

Figure 8 - Configuring the closing response



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

Message group

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

Message

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

Variations - optional

Add custom payloads, SSML, and card groups.

Set values

Next step in conversation
End conversation

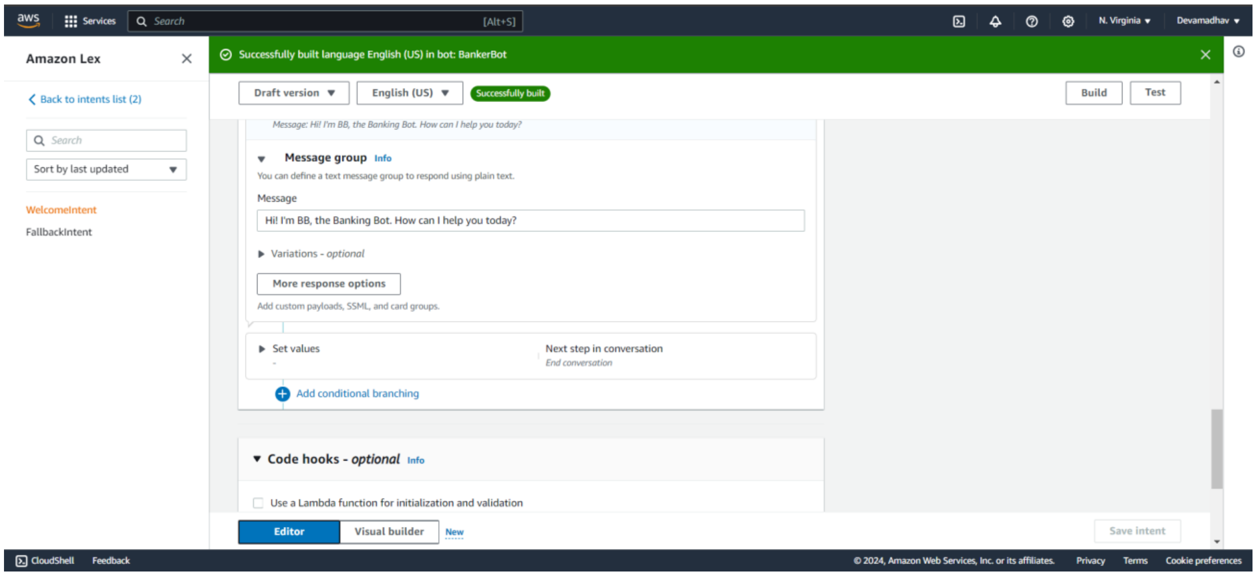
Code hooks - optional

Use a Lambda function for initialization and validation

Allow dialog code hook invocation. Turn this off to prevent invocation of the code hook throughout this intent.

English (US) has not built changes

Figure 9 -Configured the Message group and building the bot



Successfully built language English (US) in bot: BankerBot

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

Message group

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

Message

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

Variations - optional

Add custom payloads, SSML, and card groups.

Set values

Next step in conversation
End conversation

Code hooks - optional

Use a Lambda function for initialization and validation

Figure 10 - Successfully created the intent

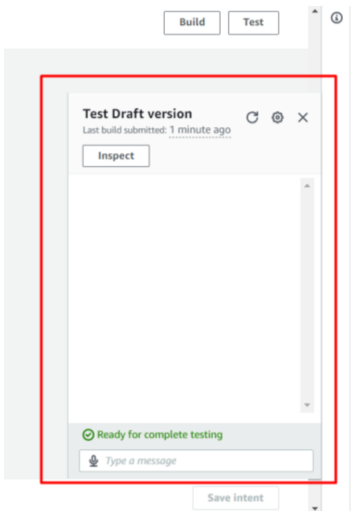
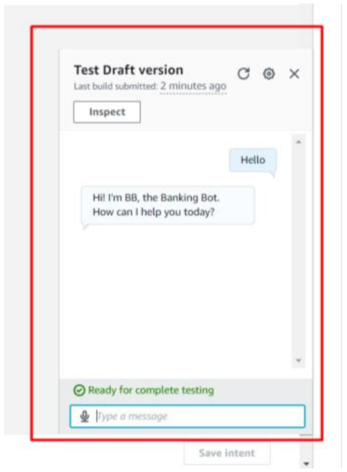

Build
Test
Test Draft version
Last build submitted: 1 minute ago
Inspect
Ready for complete testing
Type a message
Save intent

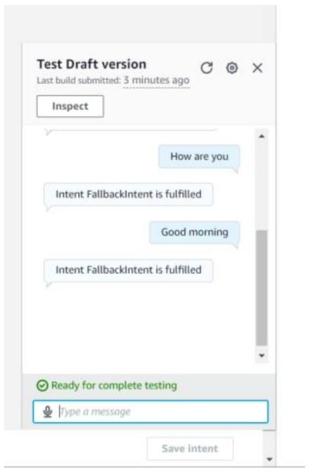
Figure 11 - Chatbot ready for testing


Test Draft version
Last build submitted: 2 minutes ago
Inspect
Ready for complete testing
Type a message
Save intent

Hello

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

Figure 12 Bot successfully responding to our utterance provided


Test Draft version
Last build submitted: 3 minutes ago
Inspect
Ready for complete testing
Type a message
Save intent

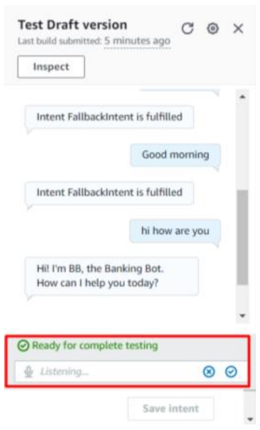
How are you

Intent FallbackIntent is fulfilled

Good morning

Intent FallbackIntent is fulfilled

Figure 13 - Negative reply when a new question is asked other than the provided utterance


Test Draft version

Last build submitted: 5 minutes ago

Inspect

Intent FallbackIntent is fulfilled

Good morning

Intent FallbackIntent is fulfilled

hi how are you

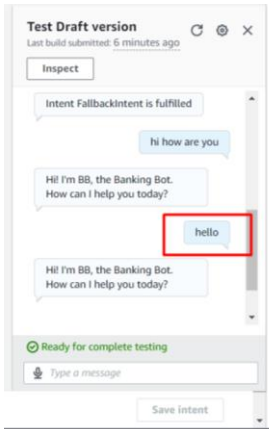
Hi! I'm BB, the Banking Bot.
How can I help you today?

Ready for complete testing

Listening...

Save intent

Figure 14 - Trying another utterances and used voice command option


Test Draft version

Last build submitted: 6 minutes ago

Inspect

Intent FallbackIntent is fulfilled

hi how are you

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

hello

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

Ready for complete testing

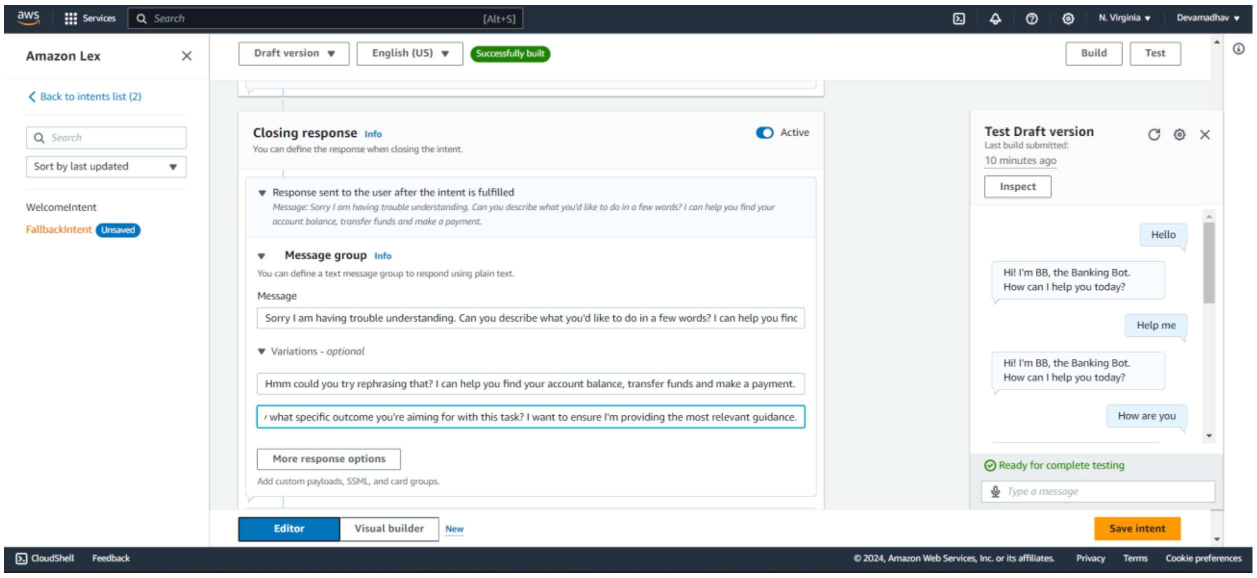
Type a message

Save intent

Figure 15 - Successful response for the voice command

Customizing the FallbackIntent:

Understanding that there would be times when our bot might not understand a user's input, we customized the FallbackIntent. This intent serves as a catch-all for situations where the bot's confidence score is too low to match any defined intents. We navigated to the FallbackIntent and edited the default message to something more user-friendly and clear, such as "I'm sorry, I didn't understand that. Could you please rephrase?" We also added variations to this message to avoid repetitiveness, making the bot's responses feel more natural. By customizing the FallbackIntent, we aimed to improve the overall user experience, especially in cases where the bot couldn't accurately interpret the user's intent.



Closing response Info — Active

You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled

Message: 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.

Message group Info

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

Message: Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you finc

Variations - *optional*

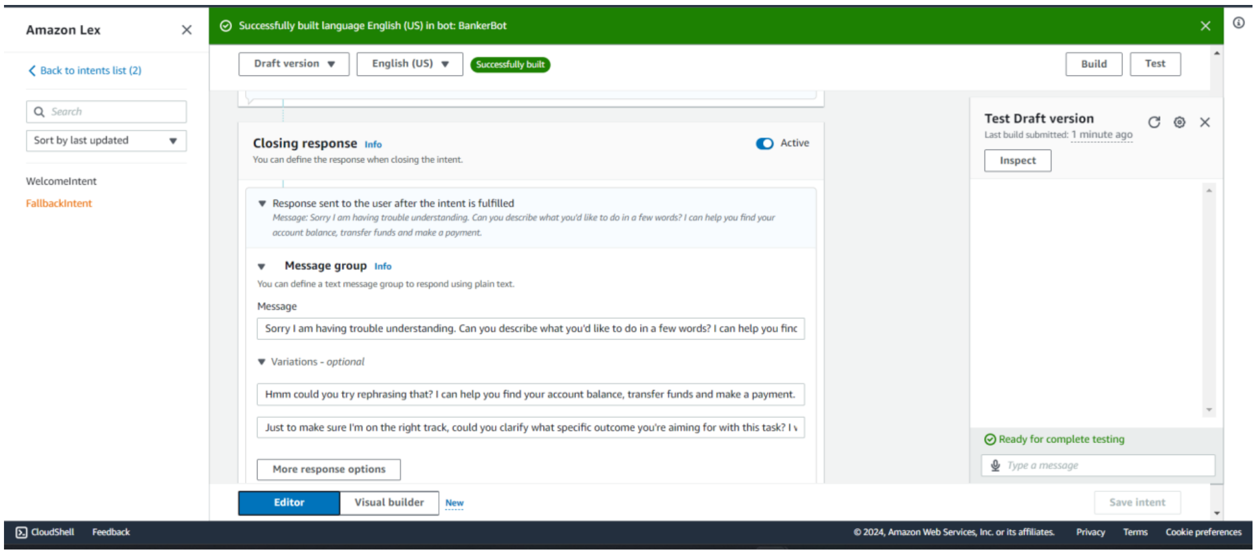
- Hmm could you try rephrasing that? I can help you find your account balance, transfer funds and make a payment.
- what specific outcome you're aiming for with this task? I want to ensure I'm providing the most relevant guidance.

Test Draft version — Last build submitted: 10 minutes ago

- Hello
- Hi! I'm BB, the Banking Bot. How can I help you today?
- Help me
- Hi! I'm BB, the Banking Bot. How can I help you today?
- How are you

Ready for complete testing

Figure 16 - Configuring the fallback indent



Successfully built language English (US) in bot: BankerBot

Closing response Info — Active

Response sent to the user after the intent is fulfilled

Message: 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.

Message: Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you finc

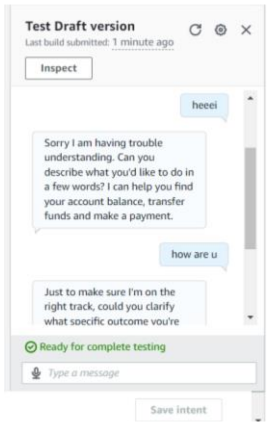
Variations - *optional*

- Hmm could you try rephrasing that? I can help you find your account balance, transfer funds and make a payment.
- Just to make sure I'm on the right track, could you clarify what specific outcome you're aiming for with this task? I v

Test Draft version — Last build submitted: 1 minute ago

Ready for complete testing

Figure 17 - Saved the changes by builting the bot



Test Draft version — Last build submitted: 1 minute ago

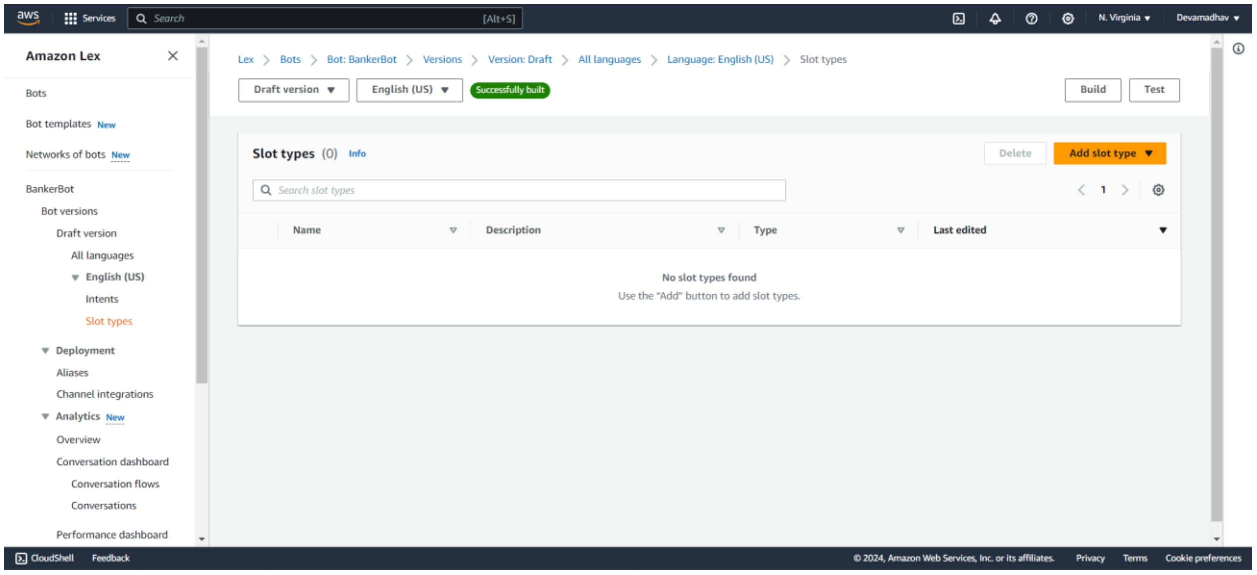
- heeei
- Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment.
- how are u
- Just to make sure I'm on the right track, could you clarify what specific outcome you're

Ready for complete testing

Figure 18 - Bot responding with alternative fallback intents

Creating the accountType Custom Slot:

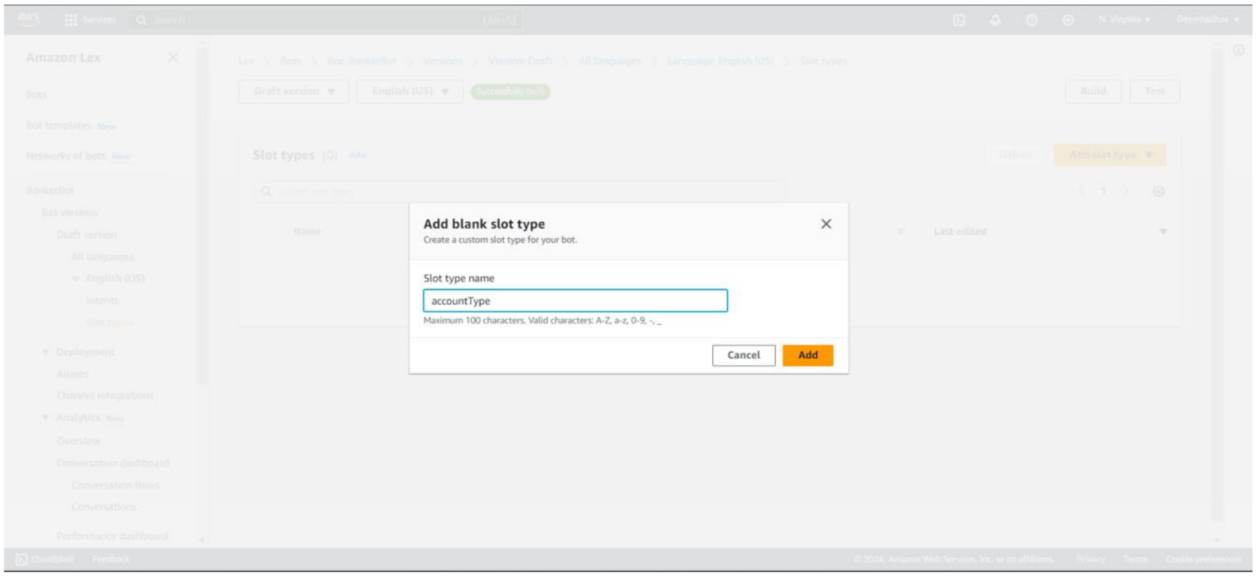
To enhance the bot's functionality, we needed it to recognize specific account types when users asked about their bank balances. To achieve this, we created a custom slot type called accountType. We selected "Add slot type" from the left-hand navigation panel and chose to add a blank slot type. We named the slot accountType and restricted its values to only those we specified, ensuring the bot would only accept valid account types. We defined three values: "Checking," "Savings," and "Credit," along with a few synonyms for each to account for different ways users might refer to their accounts. This customization allowed the bot to accurately capture and use the account type information provided by the user, enhancing its ability to handle balance inquiries effectively.



No slot types found

Use the "Add" button to add slot types.

Figure 19 - Displaying the slots available



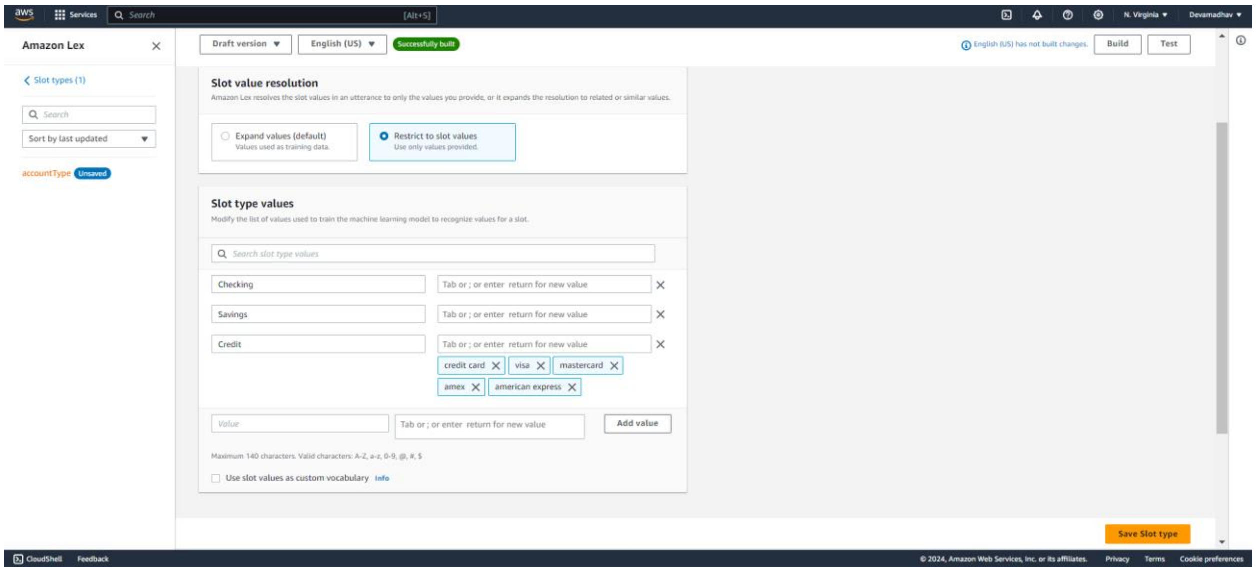
Add blank slot type

Create a custom slot type for your bot.

Slot type name: accountType

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

Figure 20 - Creating a new slot



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) [✓ Restrict to slot values]

Slot type values

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

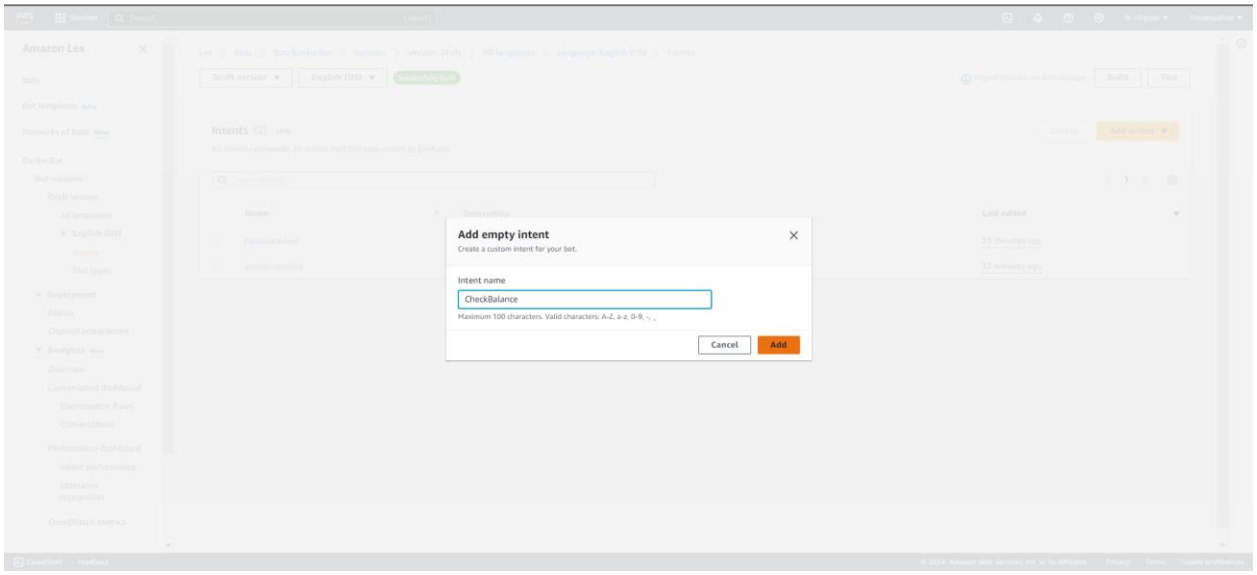
Checking	
Savings	
Credit	credit card, visa, mastercard, amex, american express

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

Figure 21 - Configuring the slot values

Developing the CheckBalance Intent:

With the accountType slot in place, we proceeded to create the CheckBalance intent, which would allow users to check their bank balances. We named the intent "CheckBalance" and provided a description explaining its purpose. In the sample utterances panel, we added various phrases users might say to check their balance, such as "What’s my {accountType} balance?" and "Show me my balance for {accountType}." These utterances included the {accountType} placeholder, enabling the bot to identify and extract the account type from the user's input. We then added a slot to the intent, using our custom accountType slot type. We provided prompts for the bot to ask if the user didn't specify their account type initially, ensuring the conversation flowed smoothly. After saving and building the intent, we tested it by asking the bot to check balances for different account types, confirming that it correctly prompted for and used the account type information.



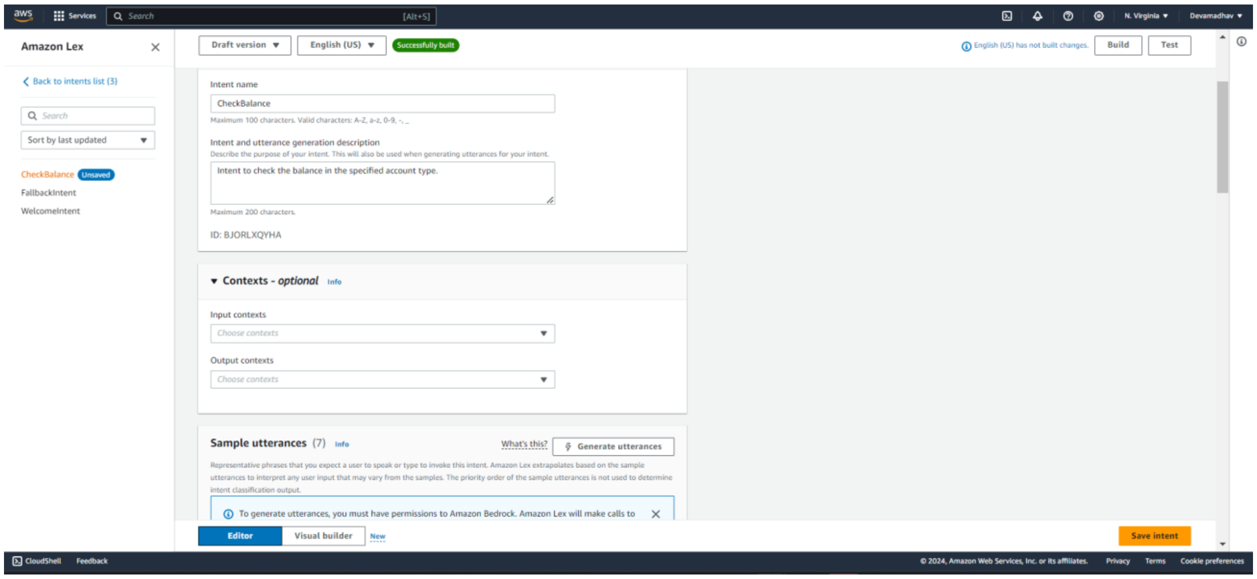
Add empty intent

Create a custom intent for your bot.

Intent name: CheckBalance

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

Figure 22 - Creating a new intent for checkBalance



Intent name: CheckBalance

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

Intent and utterance generation description

Describe the purpose of your intent. This will also be used when generating utterances for your intent.

Intent to check the balance in the specified account type.

Maximum 200 characters.

ID: BJORLXQYHA

Contexts - *optional*

Input contexts: Choose contexts

Output contexts: Choose contexts

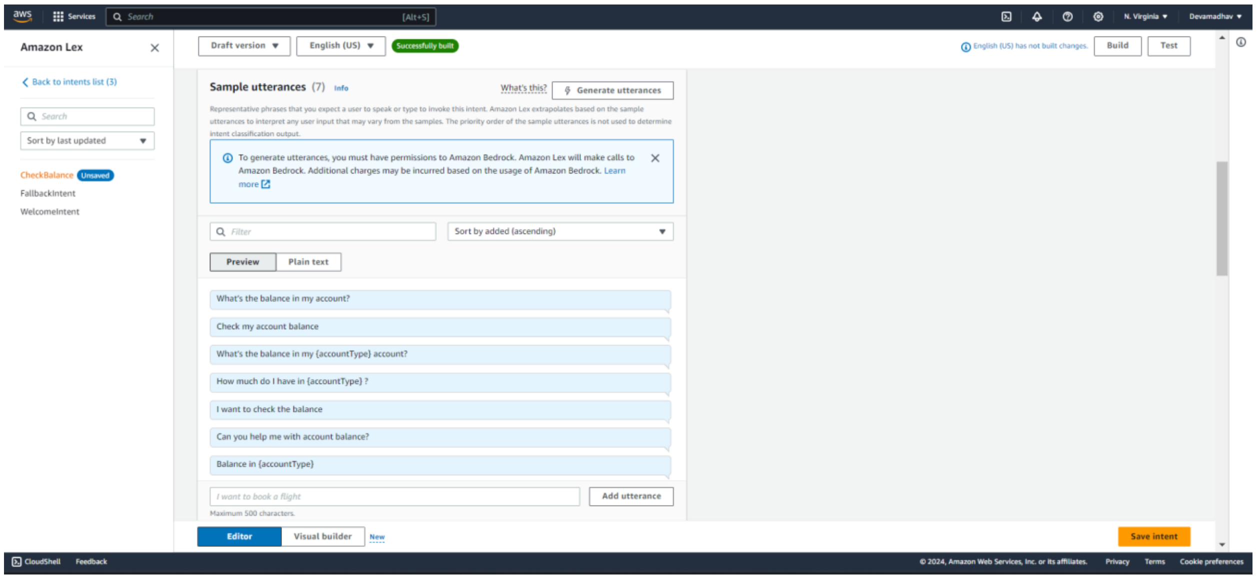
Sample utterances (7)

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.

To generate utterances, you must have permissions to Amazon Bedrock. Amazon Lex will make calls to

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 23 -Configuring the checkbalance intent



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.

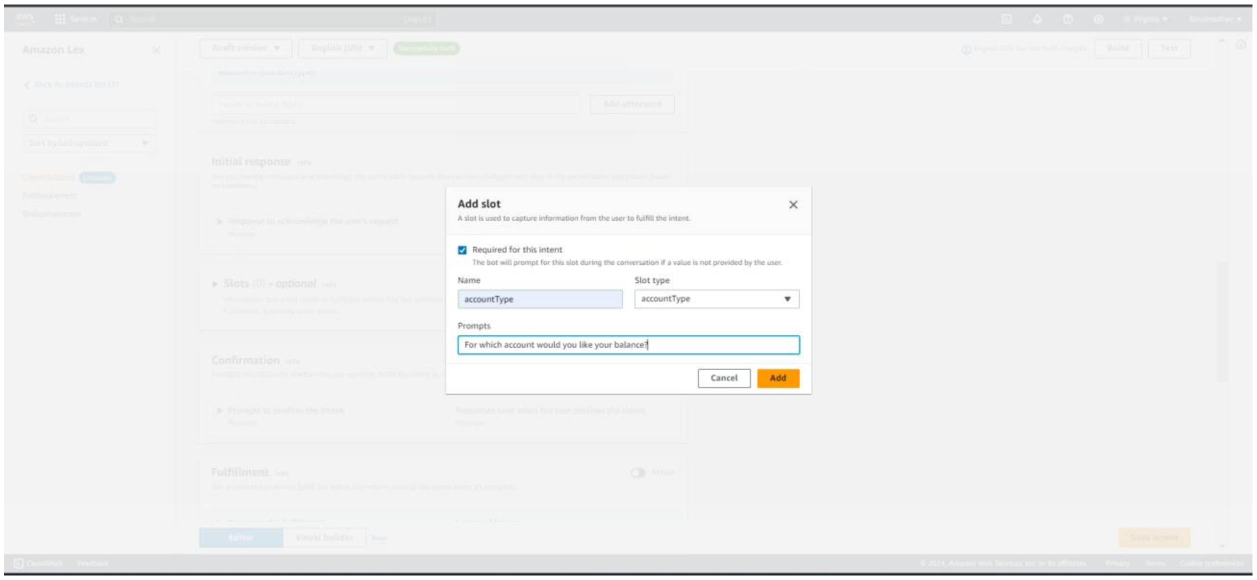
To generate utterances, you must have permissions to Amazon Bedrock. Amazon Lex will make calls to Amazon Bedrock. Additional charges may be incurred based on the usage of Amazon Bedrock. Learn more

Sort by added (ascending)

- What's the balance in my account?
- Check my account balance
- What's the balance in my {accountType} account?
- How much do I have in {accountType} ?
- I want to check the balance
- Can you help me with account balance?
- Balance in {accountType}

Maximum 500 characters.

Figure 24 - Configured the utterences for the checkBalance



Add slot

A slot is used to capture information from the user to fulfill the intent.

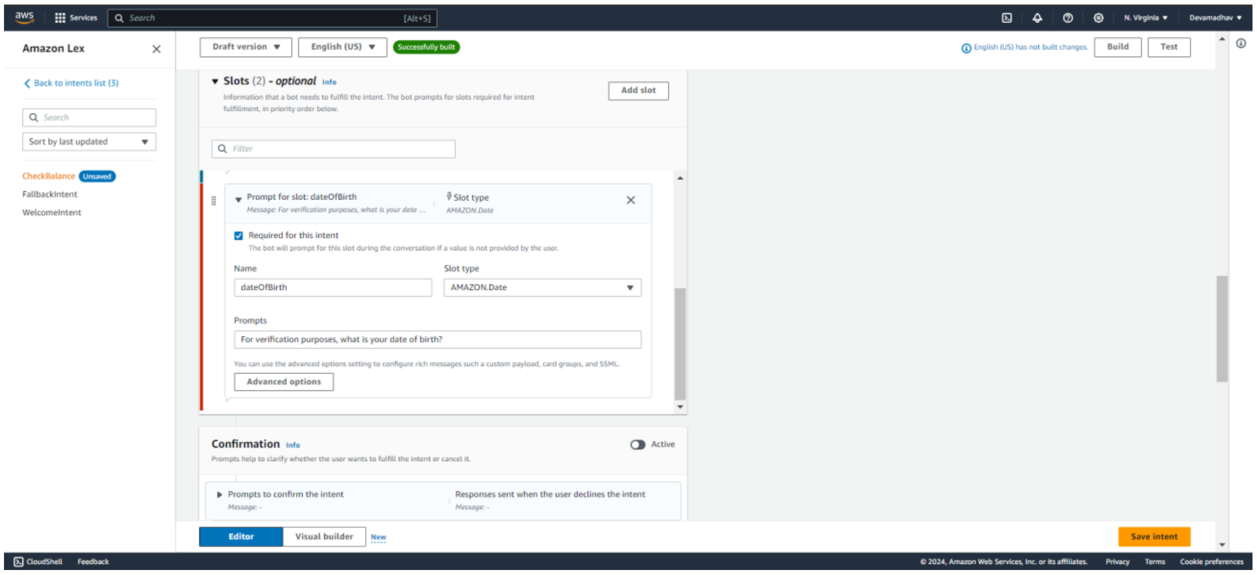
☑ Required for this intent

The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name	Slot type
accountType	accountType

Prompts: For which account would you like your balance?

Figure 25 - Selected the newly created slot for checkBalance


Amazon Lex

Back to intents list (3)

CheckBalance Unsaved
FallbackIntent
WelcomeIntent

Draft version | English (US) | Successfully built

English (US) has not built changes.

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

☑ Required for this intent
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: dateOfBirth
Slot type: AMAZON.Date

Prompts: For verification purposes, what is your date of birth?

You can use the advanced options setting to configure rich messages such a custom payload, card groups, and SSML.

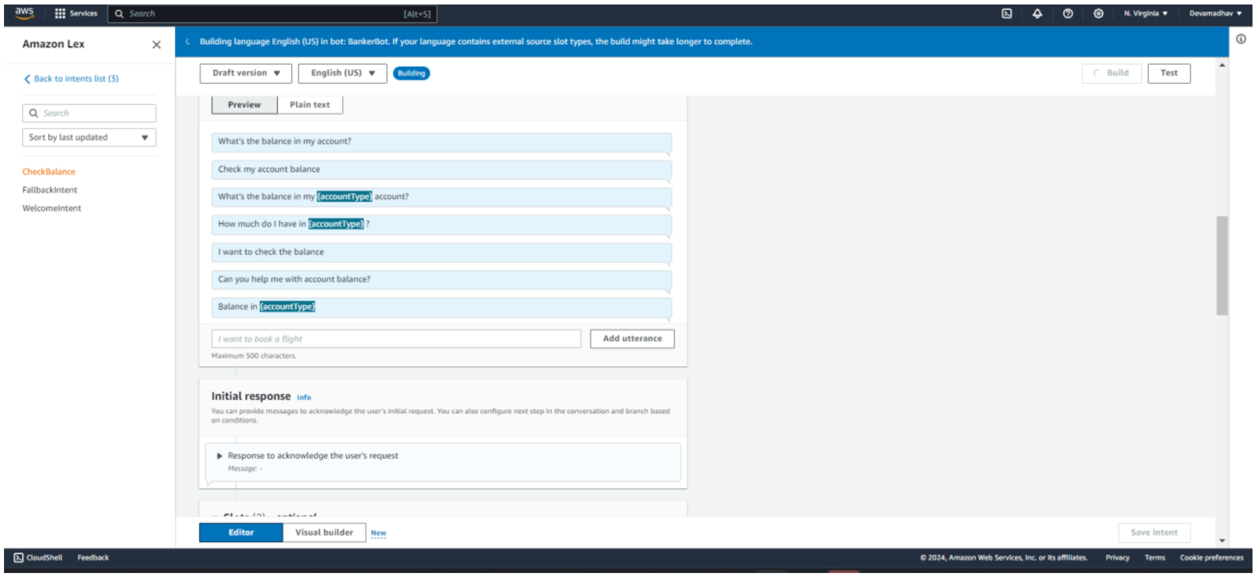
Confirmation

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

Prompts to confirm the intent — Message: -
Responses sent when the user declines the intent — Message: -

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 26 - Created another slot for date of birth


Building language English (US) in bot: BankerBot. If your language contains external source slot types, the build might take longer to complete.

Draft version | English (US) | Building

What's the balance in my account?
Check my account balance
What's the balance in my {accountType} account?
How much do I have in {accountType} ?
I want to check the balance
Can you help me with account balance?
Balance in {accountType}

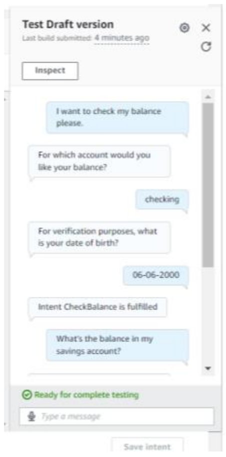
Maximum 500 characters.

Initial response

You can provide messages to acknowledge the user's initial request. You can also configure next step in the conversation and branch based on conditions.

Response to acknowledge the user's request — Message: -

Figure 27 - Building the bot


Test Draft version
Last build submitted: 4 minutes ago

I want to check my balance please.

For which account would you like your balance?

checking

For verification purposes, what is your date of birth?

06-06-2000

Intent CheckBalance is fulfilled

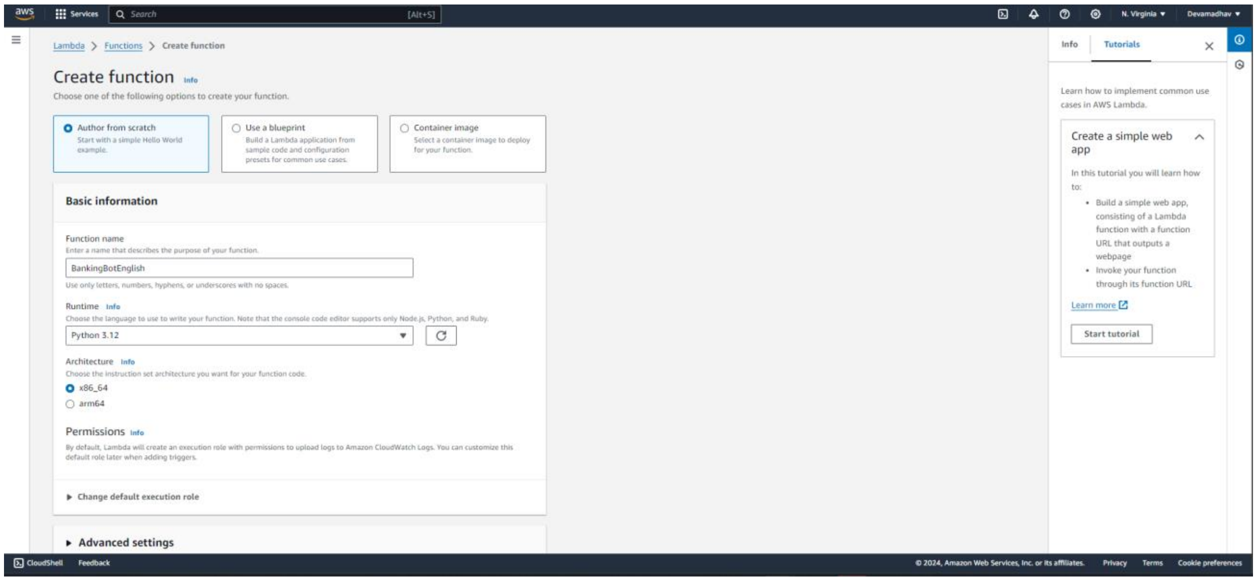
What's the balance in my savings account?

Ready for complete testing

Figure 28 - Testing the bot with checkbalance

Integrating AWS Lambda with the Bot:

To make our bot more dynamic, we integrated it with an AWS Lambda function that would generate random bank balance figures. We started by navigating to the AWS Lambda service and creating a new function from scratch. We named the function "BankingBotEnglish" and selected Python 3.12 as the runtime. After creating the function, we scrolled down to the Function code section and replaced the placeholder code with a Python script that generated random bank balances. We then deployed the function, making it ready to use. Returning to the Amazon Lex console, we connected this Lambda function to our CheckBalance intent by associating it with the TestBotAlias version of our bot. This integration allowed the bot to retrieve and display random balance figures when users asked about their account balances, adding a layer of realism to_our_chatbot.



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.

BankingBotEnglish

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

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

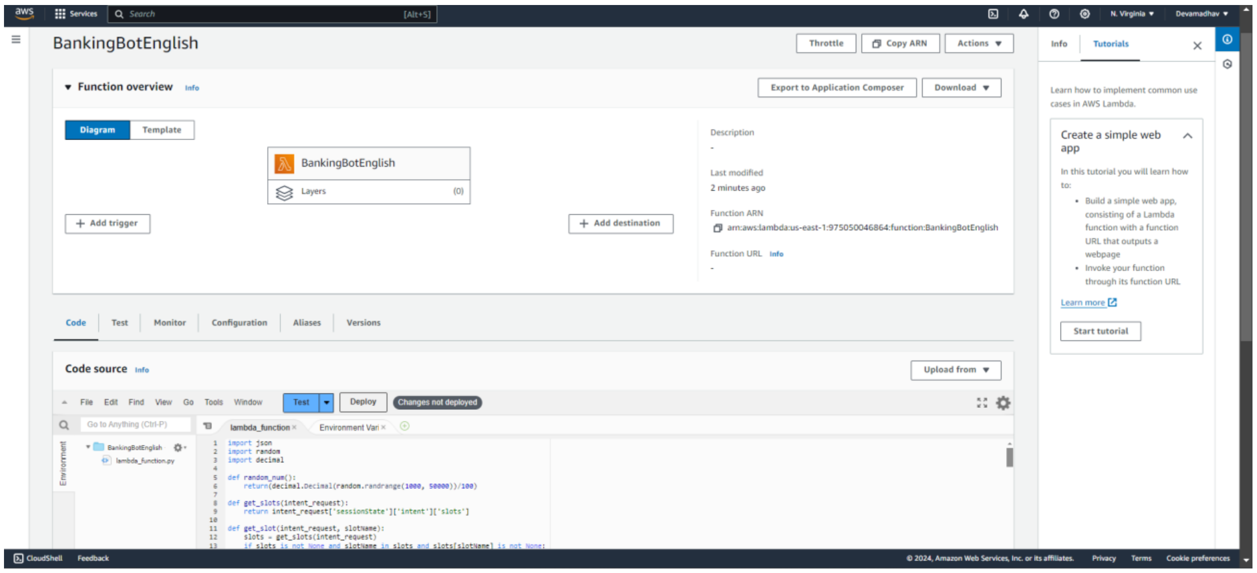
In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial

Figure 29 - Creating a new lamda function to generate random figures



BankingBotEnglish

Function overview

Description
-

Last modified
2 minutes ago

Function ARN
arn:aws:lambda:us-east-1:975050046864:function:BankingBotEnglish

Function URL
-

Code source

Changes not deployed

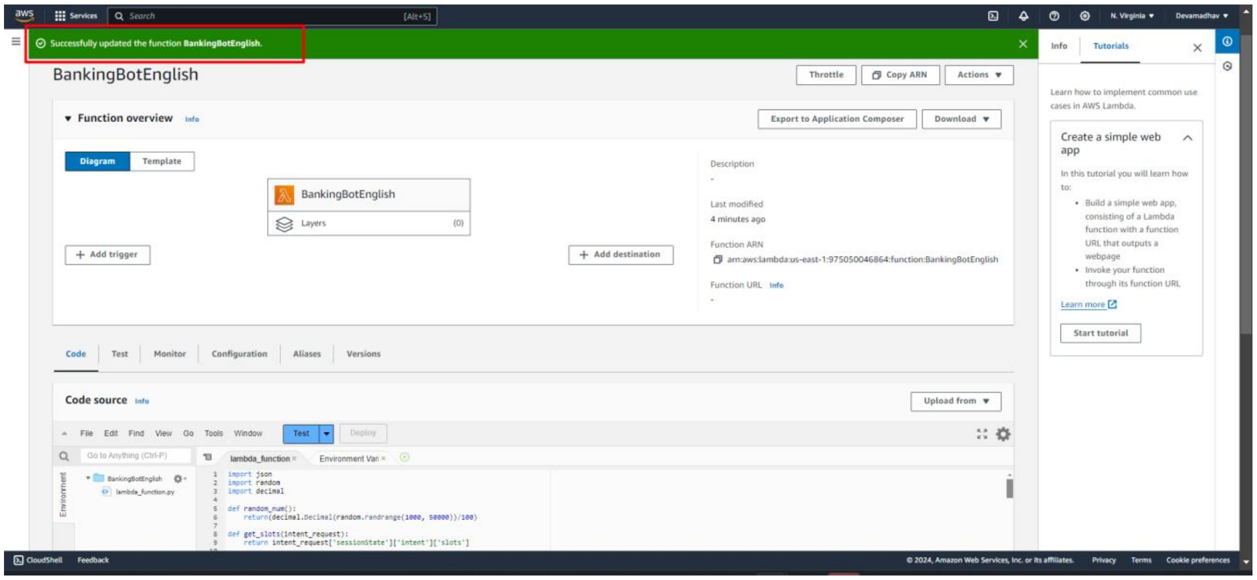
```
import json
import random
import decimal

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:
```

Figure 30 - Configured the code required



Successfully updated the function BankingBotEnglish.

BankingBotEnglish

Function overview

Description
-

Last modified
4 minutes ago

Function ARN
arn:aws:lambda:us-east-1:975050046864:function:BankingBotEnglish

Function URL
-

Code source

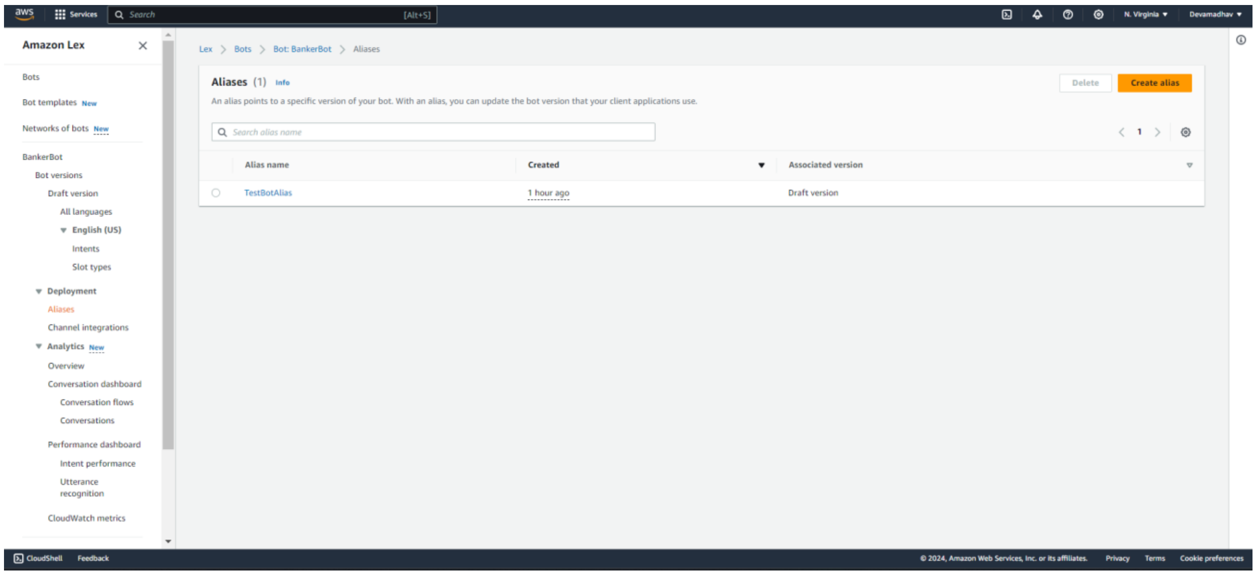
```
import json
import random
import decimal

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

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

Figure 31 - Successfully created the new function

Connect AWS Lambda with Amazon Lex

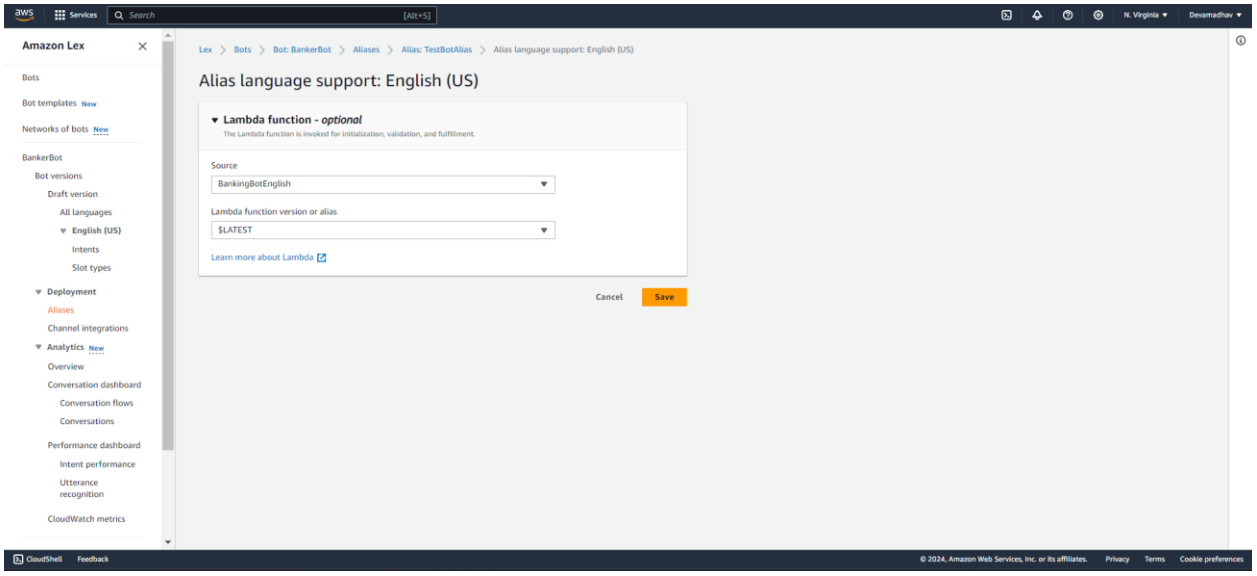

Lex > Bots > Bot: BankerBot > Aliases

Aliases (1)

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	Associated version
TestBotAlias	1 hour ago	Draft version

Figure 32 - Accessing Alias tab in Lex


Lex > Bots > Bot: BankerBot > Aliases > Alias: TestBotAlias > Alias language support: English (US)

Alias language support: English (US)

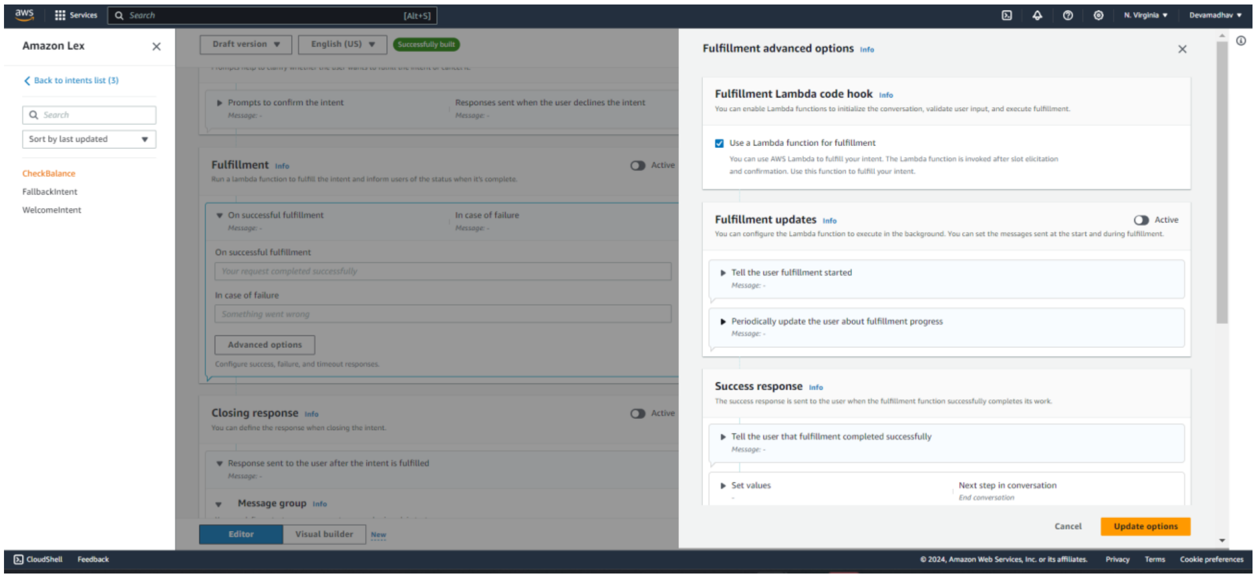
Lambda function - *optional*

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

Source: BankingBotEnglish

Lambda function version or alias: $LATEST

Figure 33 - Creating a new alias language support


Amazon Lex

Back to intents list (3)

Sort by last updated

CheckBalance
FallbackIntent
WelcomeIntent

Draft version | English (US) | Successfully built

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.

Fulfillment updates Info — Active
You can configure the Lambda function to execute in the background. You can set the messages sent at the start and during fulfillment.

Tell the user fulfillment started
Message: -

Periodically update the user about fulfillment progress
Message: -

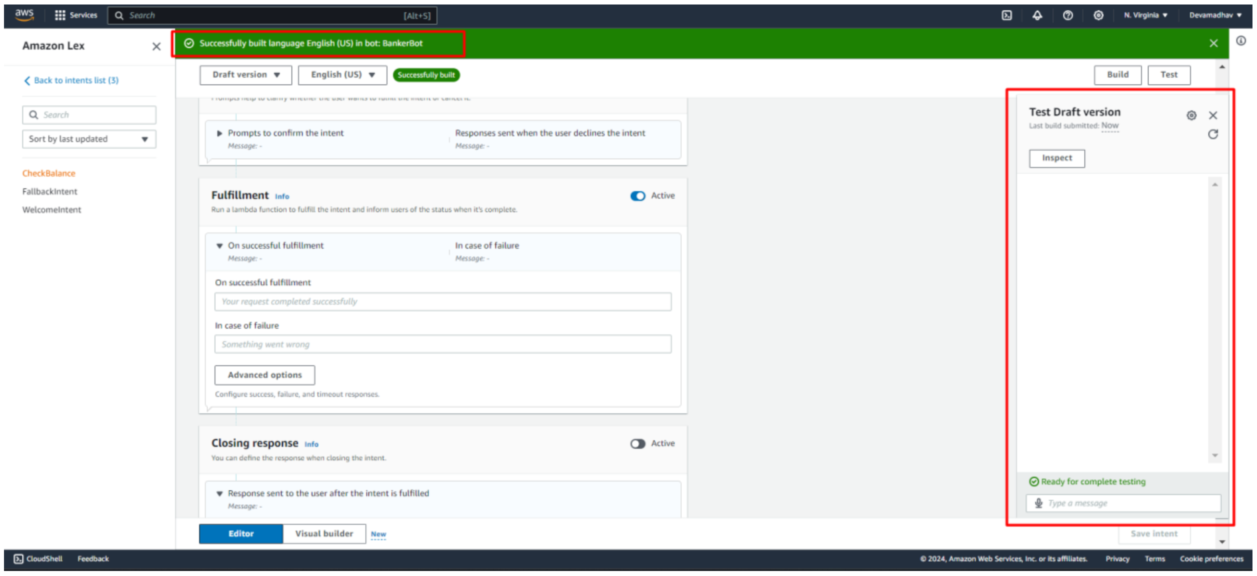
Success response Info
The success response is sent to the user when the fulfillment function successfully completes its work.

Tell the user that fulfillment completed successfully
Message: -

Set values | Next step in conversation: End conversation

Cancel | Update options

Figure 34 -selected the use lambda function for fullfillment


Successfully built language English (US) in bot: BankerBot

Draft version | English (US) | Successfully built | Build | Test

Prompts to confirm the intent — Message: -
Responses sent when the user declines the intent — Message: -

Fulfillment Info — Active
Run a lambda function to fulfill the intent and inform users of the status when it's complete.

On successful fulfillment — Message: -
In case of failure — Message: -

On successful fulfillment
Your request completed successfully
In case of failure
Something went wrong

Advanced options
Configure success, failure, and timeout responses.

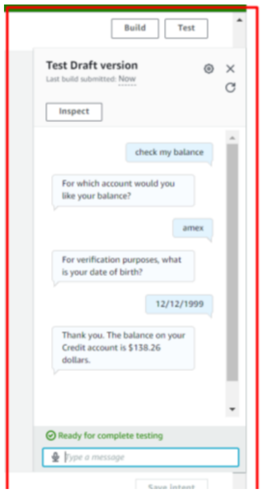
Closing response Info — Active
You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled — Message: -

Test Draft version
Last build submitted: Now
Inspect
Ready for complete testing
Type a message

Editor | Visual builder | Save intent

Figure 35 - Successfully built the language



check my balance

For which account would you like your balance?

amex

For verification purposes, what is your date of birth?

12/12/1999

Thank you. The balance on your Credit account is $138.26 dollars.

Figure 36 - Testing the bot

Create an output context from CheckBalance

In this step, we enhanced our BankerBot's ability to maintain the flow of conversation by creating an output context within the CheckBalance intent. The output context allows the bot to remember certain pieces of information, such as the user's selected account type, across multiple interactions within a defined period.

To do this, we navigated to the CheckBalance intent and scrolled down to the Context section. Here, we created a new output context and gave it a descriptive name, such as "AccountTypeContext." We set the context timeout to ensure that the bot retains this information for 5 turns or 90 seconds, whichever comes first. This configuration meant that if a user provided their account type, the bot could remember it for a short period, making subsequent interactions more efficient.

After saving the intent with the new context, we rebuilt the bot and conducted a series of tests. We observed that the bot correctly remembered the account type across multiple queries during the active context window. For instance, when we asked the bot to check the balance of a different account type within the same session, it no longer required us to specify the type again, demonstrating the successful implementation of the context feature.

This enhancement significantly improved the user experience by reducing redundant prompts and making the conversation with BankerBot feel more natural and intuitive.

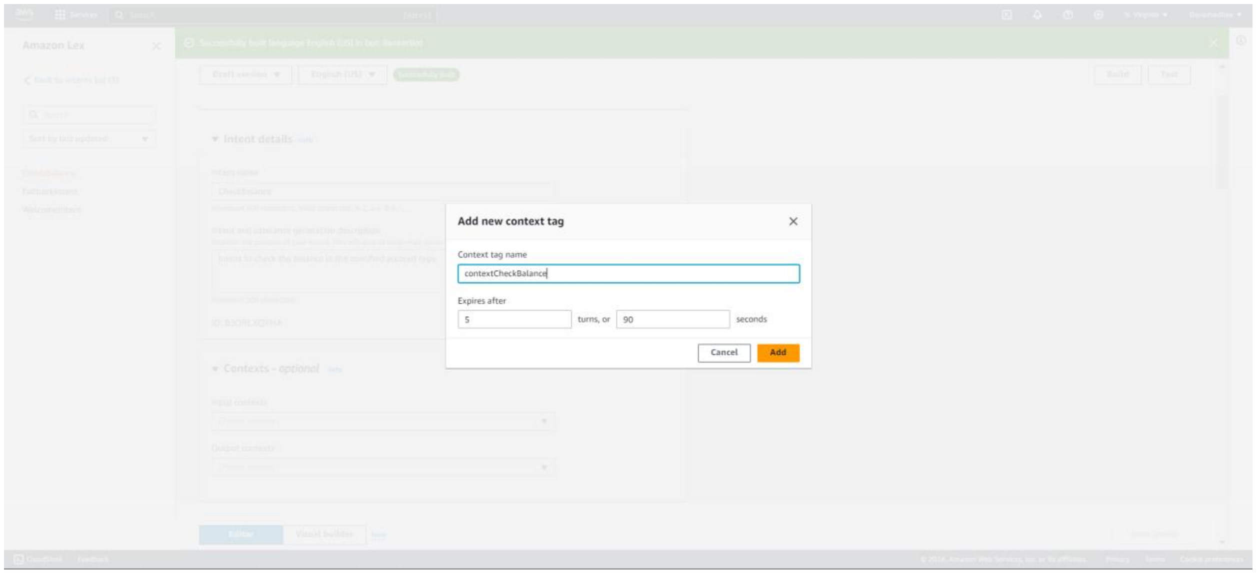


Figure 37 - Creating a new context tag contextCheck Balance

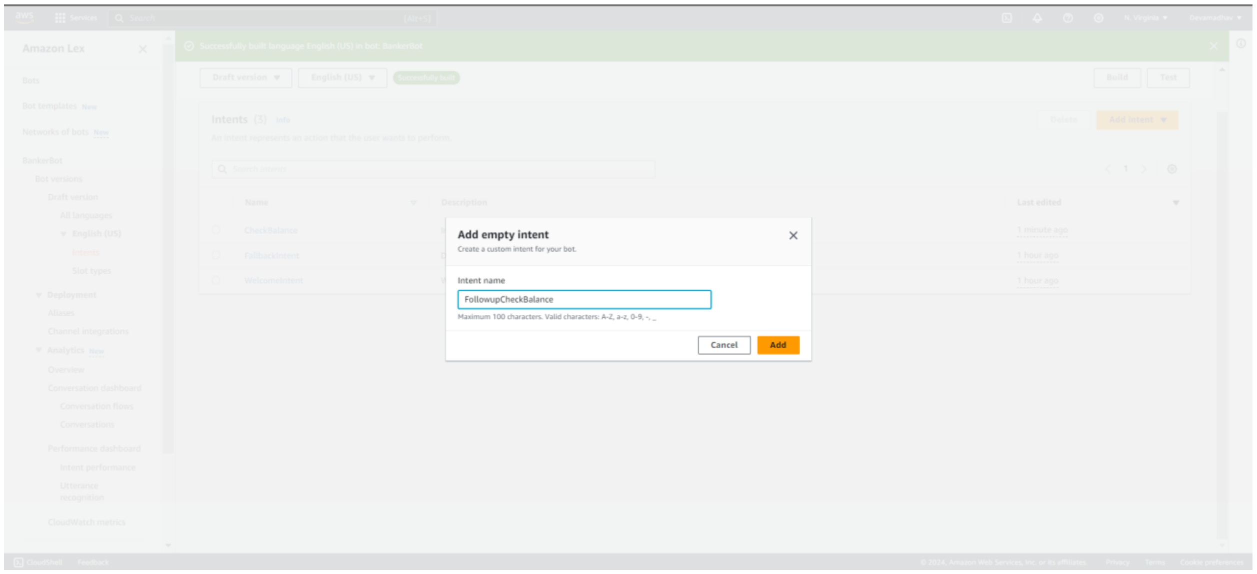
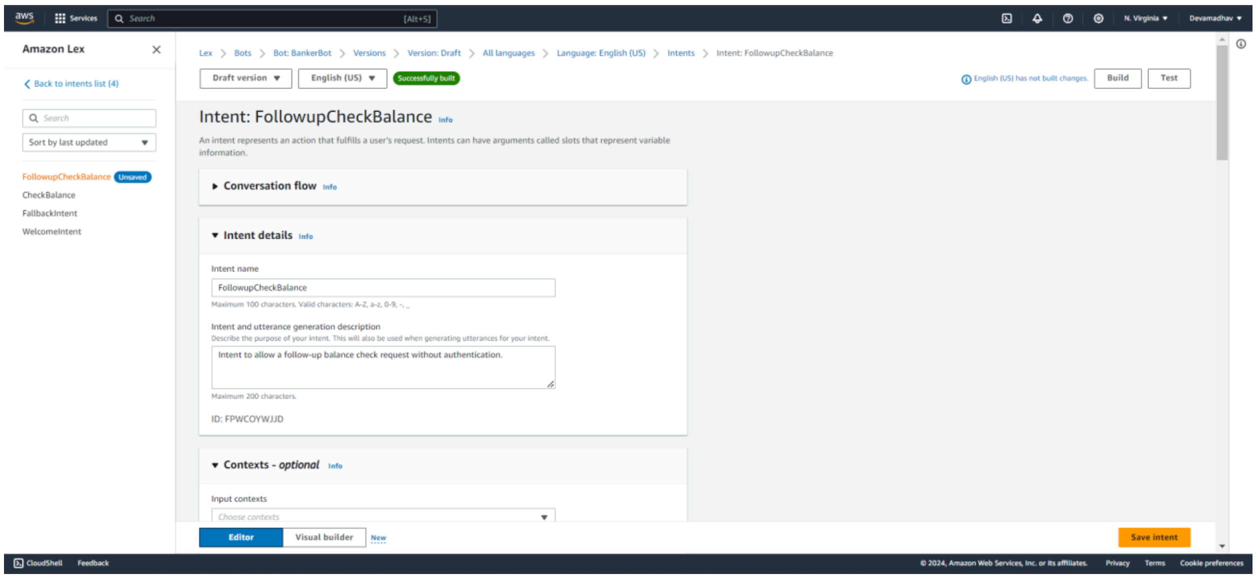

Add empty intent
Create a custom intent for your bot.
Intent name
FollowupCheckBalance
Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _
Cancel
Add

Figure 38 - Creating a new intent



Lex > Bots > Bot: BankerBot > Versions > Version: Draft > All languages > Language: English (US) > Intents > Intent: FollowupCheckBalance

Intent: FollowupCheckBalance

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent details

Intent name: FollowupCheckBalance

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

Intent and utterance generation description

Describe the purpose of your intent. This will also be used when generating utterances for your intent.

Intent to allow a follow-up balance check request without authentication.

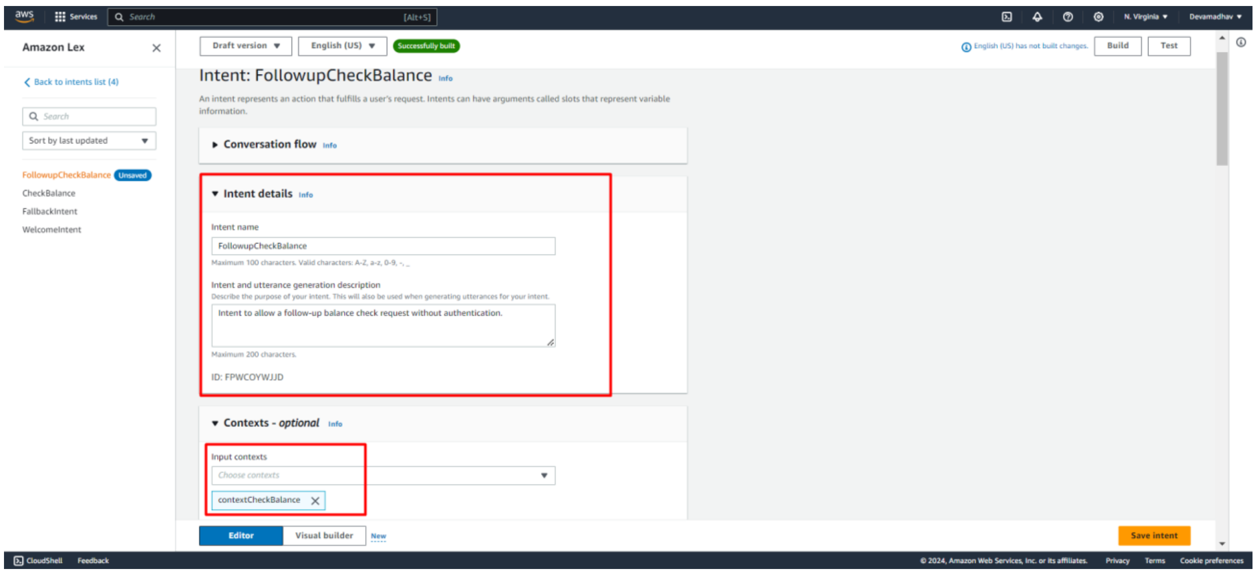
Maximum 200 characters.

ID: FPWCOYWJJD

Contexts - *optional*

Input contexts: Choose contexts

Figure 39 - Configuring the new intent



Intent: FollowupCheckBalance

An intent represents an action that fulfills a user's request. Intents can have arguments called slots that represent variable information.

Intent details

Intent name: FollowupCheckBalance

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

Intent and utterance generation description

Describe the purpose of your intent. This will also be used when generating utterances for your intent.

Intent to allow a follow-up balance check request without authentication.

Maximum 200 characters.

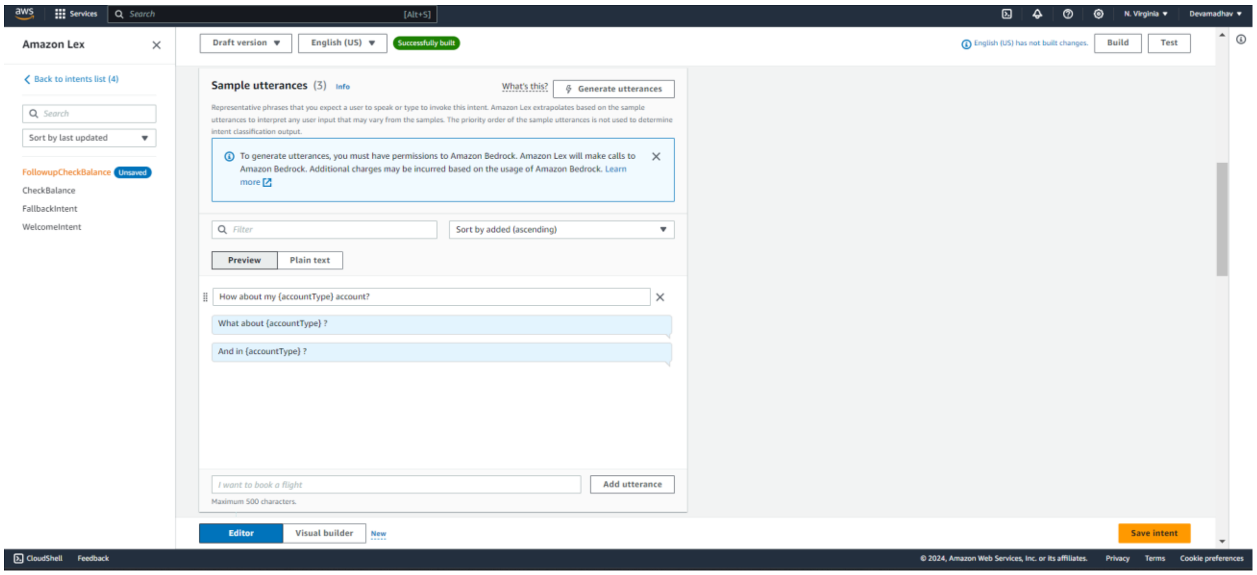
ID: FPWCOYWJJD

Contexts - *optional*

Input contexts: Choose contexts

contextCheckBalance

Figure 40 -Configuring context in followupcheckBalance



Sample utterances (3) 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.

To generate utterances, you must have permissions to Amazon Bedrock. Amazon Lex will make calls to Amazon Bedrock. Additional charges may be incurred based on the usage of Amazon Bedrock. Learn more

How about my {accountType} account?

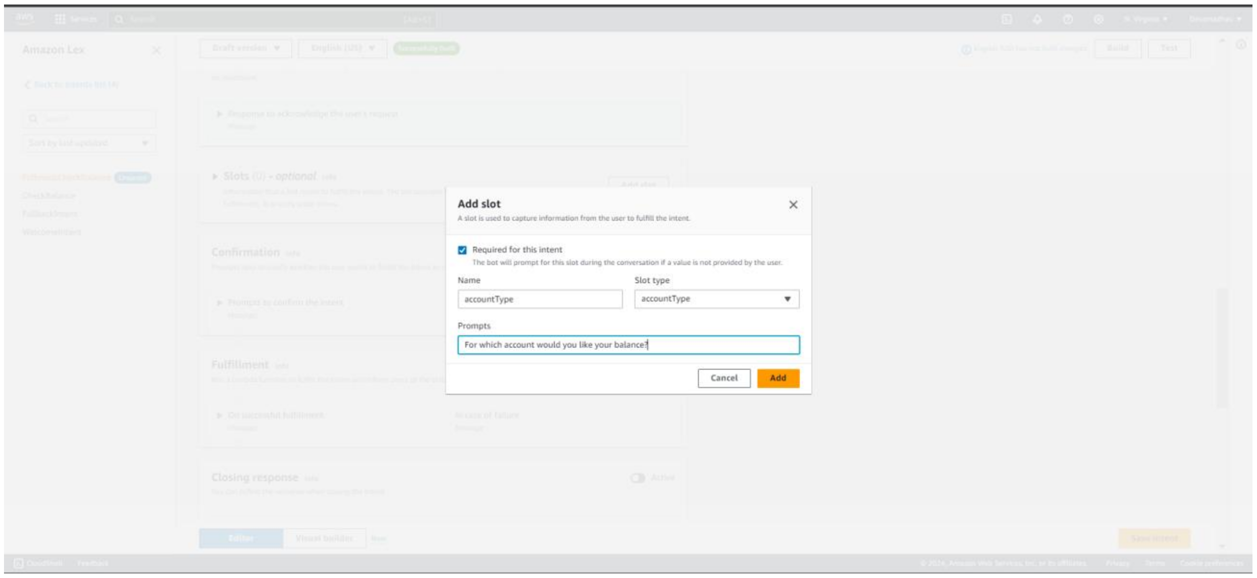
What about {accountType} ?

And in {accountType} ?

I want to book a flight

Maximum 500 characters.

Figure 41 -Configured utteramces



Add slot

A slot is used to capture information from the user to fulfill the intent.

[✓] Required for this intent

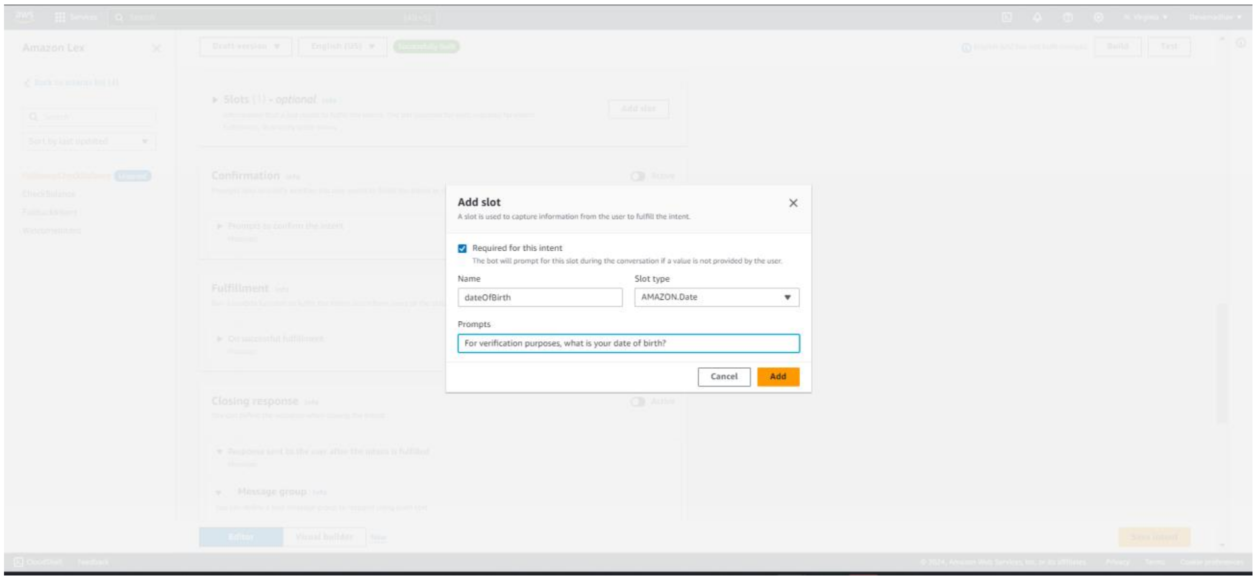
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: accountType

Slot type: accountType

Prompts: For which account would you like your balance?

Figure 42 – Adding a new slot for account type for the new intent


Add slot

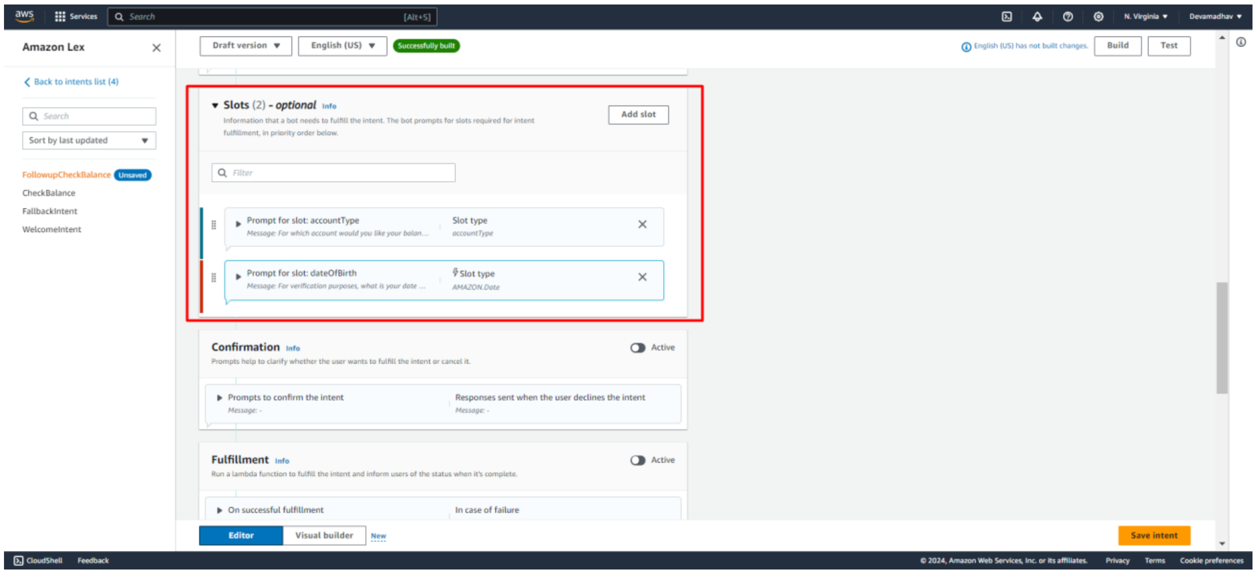
A slot is used to capture information from the user to fulfill the intent.

☑ Required for this intent

The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name	dateOfBirth
Slot type	AMAZON.Date
Prompts	For verification purposes, what is your date of birth?

Figure 43 -Adding a new slot for Date of Birth


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

Confirmation

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

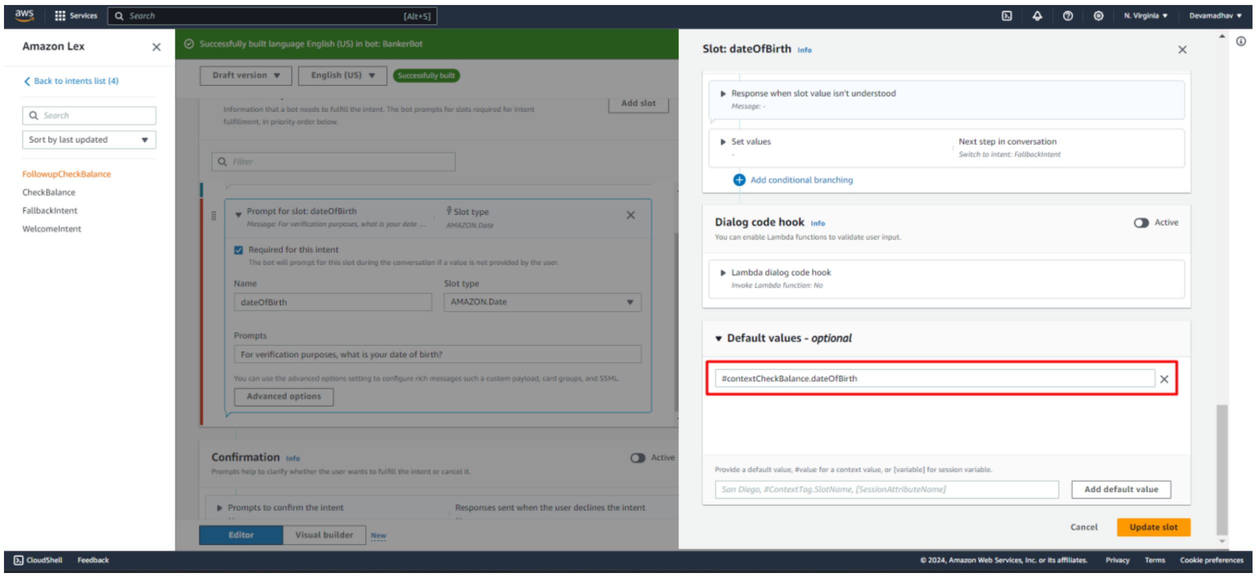
Prompts to confirm the intent Message: -	Responses sent when the user declines the intent Message: -

Fulfillment

Run a lambda function to fulfill the intent and inform users of the status when it's complete.

On successful fulfillment	In case of failure

Figure 44 - Displaying the newly added slots



Successfully built language English (US) in bot: BankerBot

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

Required for this intent
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: dateOfBirth
Slot type: AMAZON.Date

Prompts
For verification purposes, what is your date of birth?

You can use the advanced options setting to configure rich messages such a custom payload, card groups, and SSML.

Advanced options

Confirmation Info — Active
Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

Prompts to confirm the intent — Responses sent when the user declines the intent

Slot: dateOfBirth Info

Response when slot value isn't understood
Message: -

Set values — Next step in conversation
Switch to intent: FallbackIntent

Add conditional branching

Dialog code hook Info — Active
You can enable Lambda functions to validate user input.

Lambda dialog code hook
Invoke Lambda function: No

Default values - *optional*

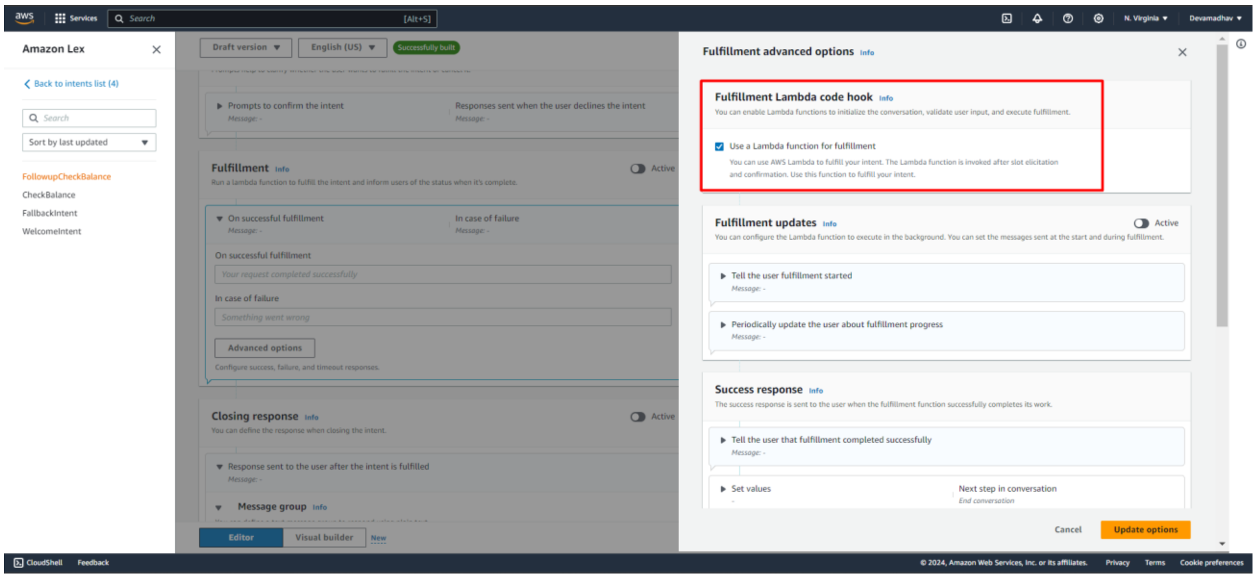
#contextCheckBalance.dateOfBirth

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

San Diego, #ContextTag.SlotName, [SessionAttributeName]

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 45 - Setting the default values



Prompts to confirm the intent — Responses sent when the user declines the intent
Message: - Message: -

Fulfillment Info — Active
Run a lambda function to fulfill the intent and inform users of the status when it's complete.

On successful fulfillment — In case of failure
Message: - Message: -

On successful fulfillment
Your request completed successfully

In case of failure
Something went wrong

Advanced options
Configure success, failure, and timeout responses.

Closing response Info — Active
You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled
Message: -

Message group Info

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.

Fulfillment updates Info — Active
You can configure the Lambda function to execute in the background. You can set the messages sent at the start and during fulfillment.

Tell the user fulfillment started
Message: -

Periodically update the user about fulfillment progress
Message: -

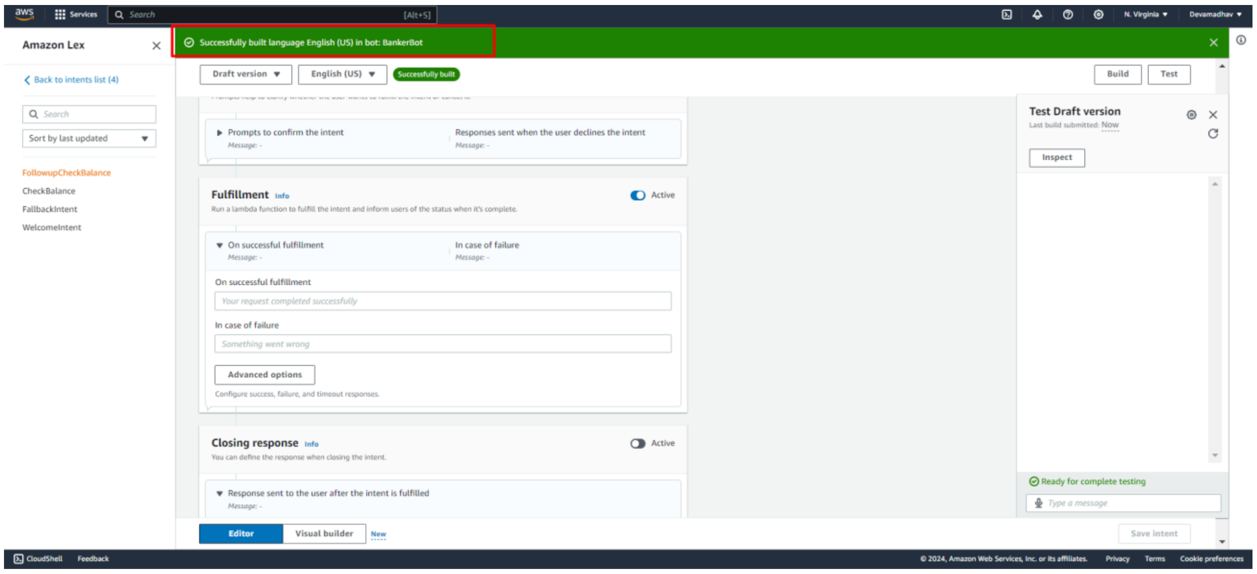
Success response Info
The success response is sent to the user when the fulfillment function successfully completes its work.

Tell the user that fulfillment completed successfully
Message: -

Set values — Next step in conversation
End conversation

© 2024, Amazon Web Services, Inc. or its affiliates.

Figure 46 - Using the Lamda functions for fullfillment


Successfully built language English (US) in bot: BankerBot

Amazon Lex
Back to intents list (4)
Sort by last updated
FollowupCheckBalance
CheckBalance
FallbackIntent
WelcomeIntent
Draft version
English (US)
Successfully built
Build
Test
Prompts to confirm the intent
Responses sent when the user declines the intent
Fulfillment
Active
Run a lambda function to fulfill the intent and inform users of the status when it's complete.
On successful fulfillment
In case of failure
Advanced options
Configure success, failure, and timeout responses.
Closing response
You can define the response when closing the intent.
Response sent to the user after the intent is fulfilled
Test Draft version
Inspect
Ready for complete testing
Type a message
Editor
Visual builder
Save intent

Figure 47 - Successfully built the new bot

Testing the Bot:

After setting up the Lambda function and connecting it to our bot, we thoroughly tested the entire system. We interacted with BankerBot by asking it to check the balance for different account types. During testing, we observed that the bot correctly recognized the account type, prompted for any missing information, and then retrieved and displayed a random balance figure using the Lambda function. This step was crucial in validating that all components were working together seamlessly and that the bot provided accurate and consistent responses to user queries.

Enhancing the Bot with Context Management:

To improve the user experience further, we added context management capabilities to BankerBot. We navigated back to the CheckBalance intent and created an output context tag. This tag allowed the bot to remember certain details, such as the user's account type, across multiple interactions. We set the context timeout to 5 turns or 90 seconds to balance usability with security concerns. After saving and building the intent with the new context tag, we tested the bot again to ensure it still operated correctly and that the context was being applied as expected. This enhancement made the bot more efficient by reducing the need for users to repeatedly provide the same information during a conversation.