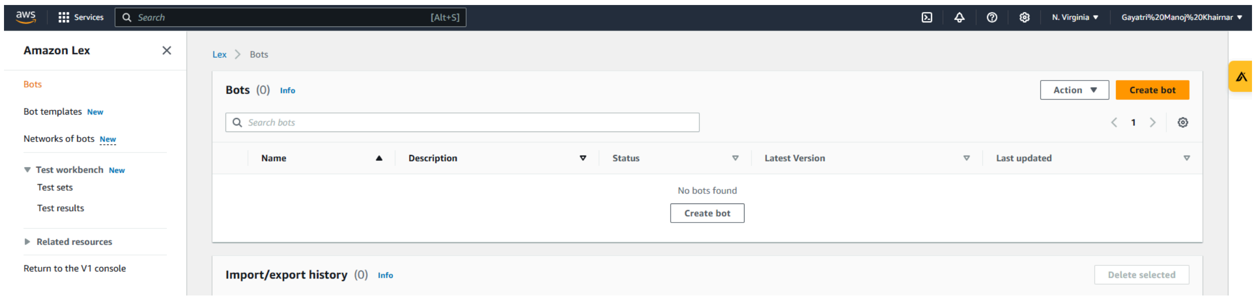
Creating a Conversational Bot with Amazon Lex

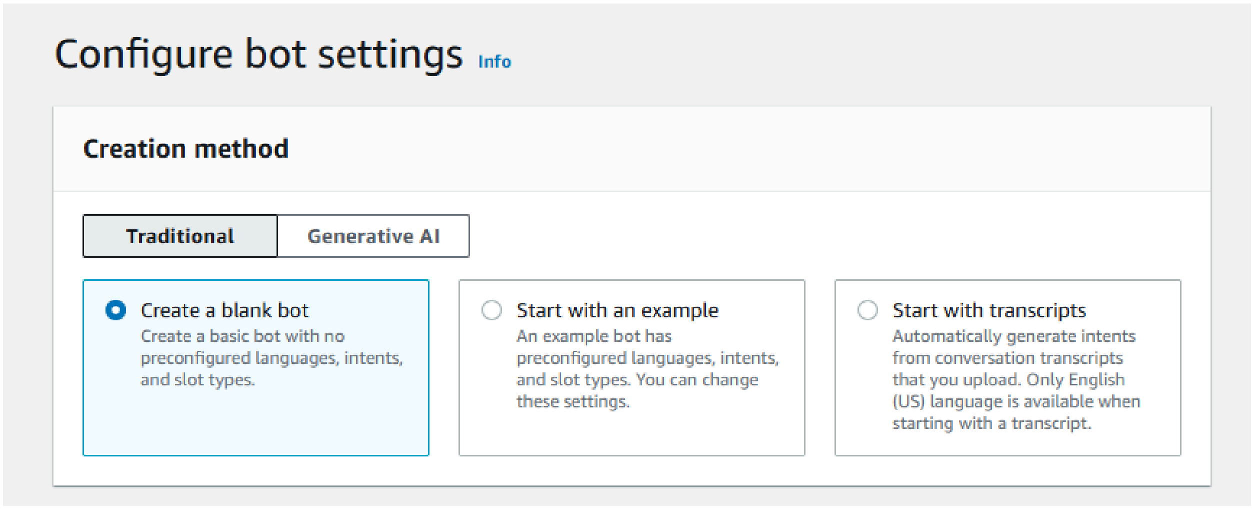
This guide outlines the step-by-step process for creating a conversational bot using Amazon Lex. It is designed for future reference and to assist others in replicating the process.

Table of Contents

Setting Up Amazon Lex

1. **Log In**:
    - Access your AWS account and navigate to Amazon Lex.
    - Ensure you are using the Lex V2 console (check for "lexv2" in the URL).
2. **Create a New Bot**:
    - Select "Create bot" and choose "Create a blank bot."




Configure bot settings Info

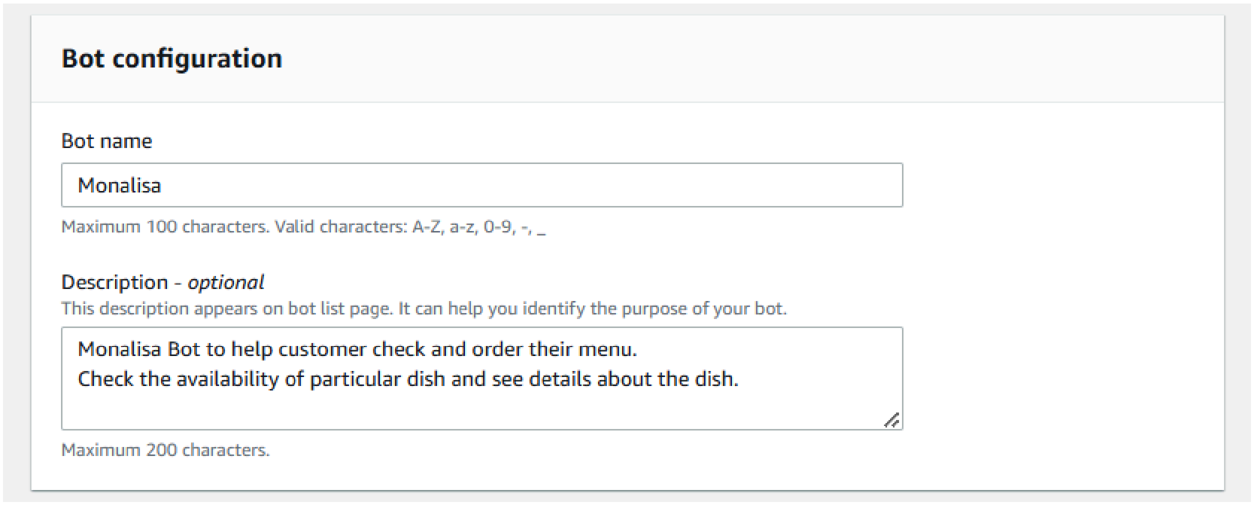
Creation method

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: Monalisa

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

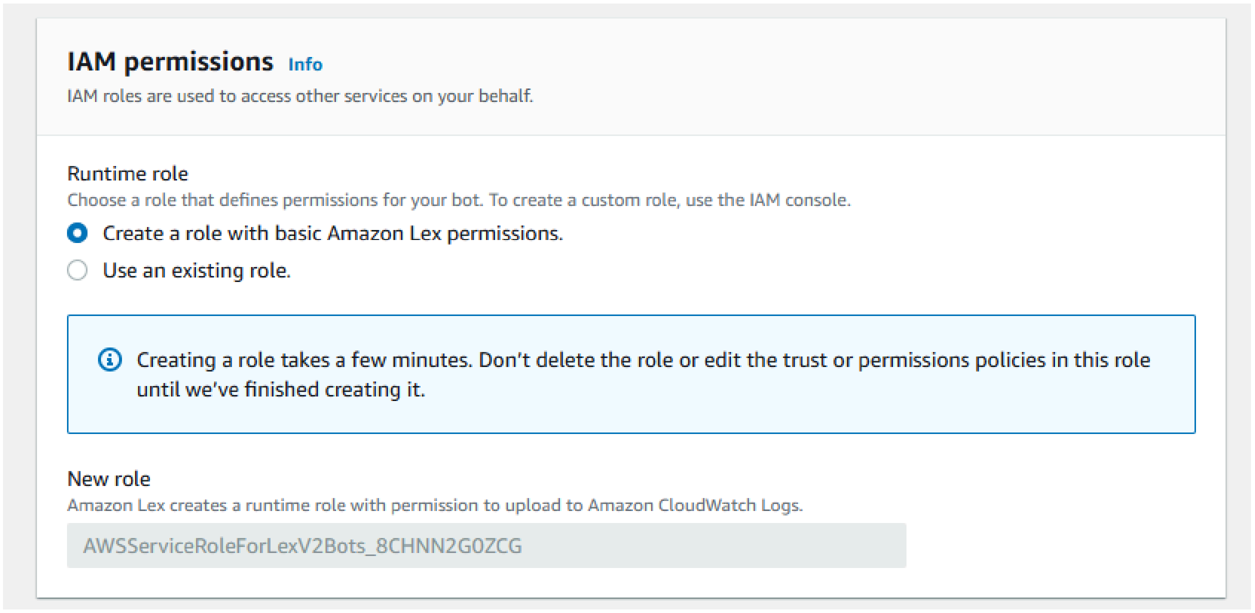
Description - *optional*

This description appears on bot list page. It can help you identify the purpose of your bot.

Monalisa Bot to help customer check and order their menu.
Check the availability of particular dish and see details about the dish.

Maximum 200 characters.

- Configure the following:
  - **IAM Permissions**: Create a role with basic Amazon Lex permissions.



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.

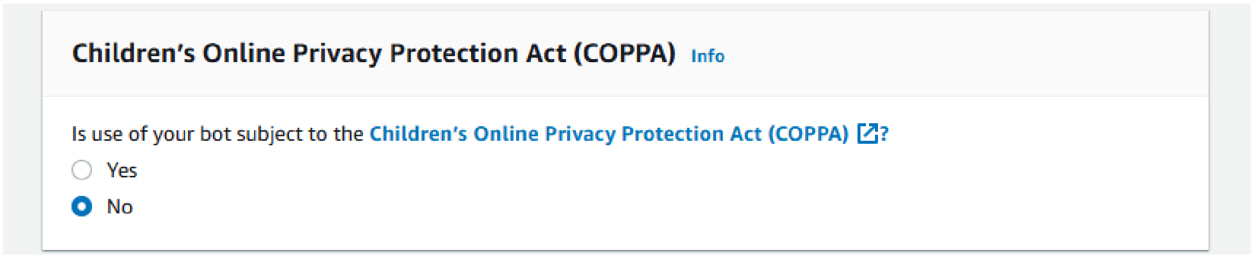
Creating a role takes a few minutes. Don't delete the role or edit the trust or permissions policies in this role until we've finished creating it.

New role

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

AWSServiceRoleForLexV2Bots_8CHNN2G0ZCG

- **COPPA Compliance**: Select "No."

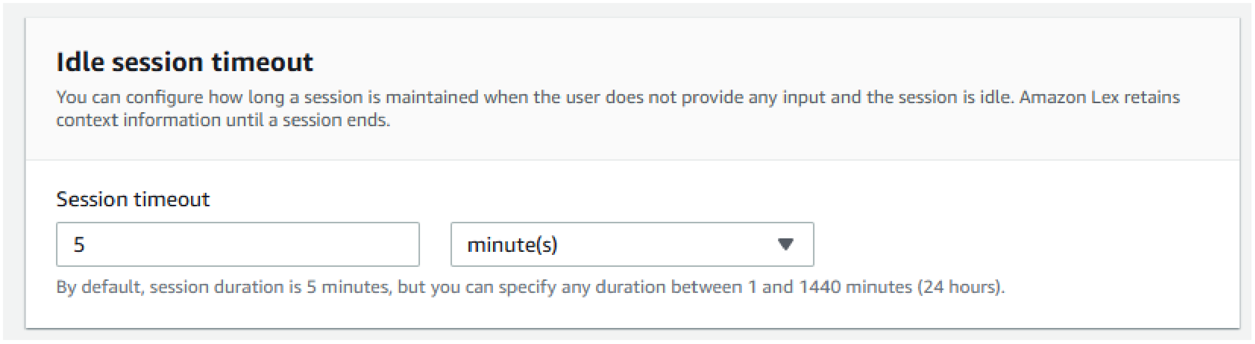


Children's Online Privacy Protection Act (COPPA) Info

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

Yes [✓ No]

- **Idle Session Timeout**: Keep the default of 5 minutes.



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.

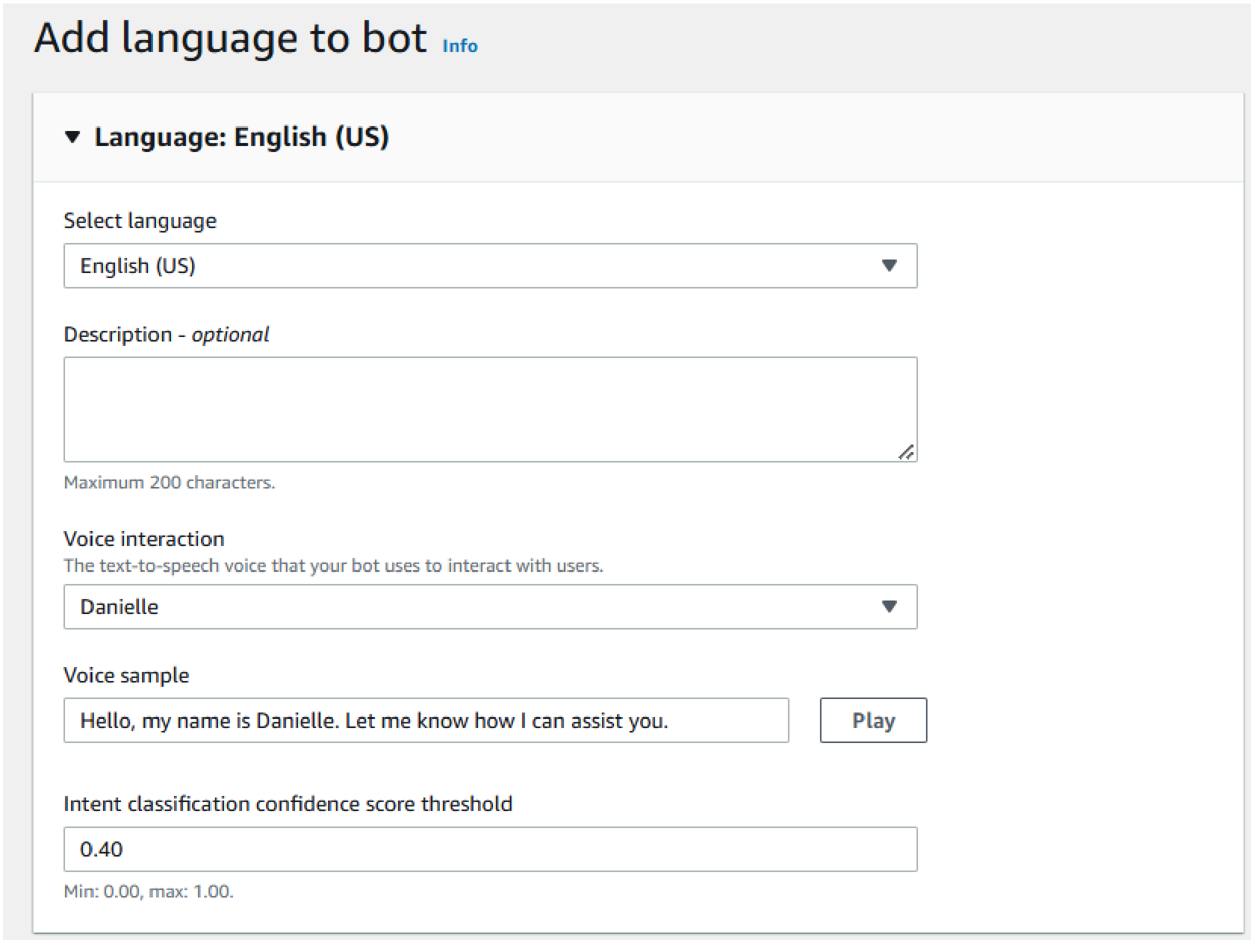
Session timeout

5 minute(s)

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

3. **Customize Voice Settings**:
    - Keep the language as English.
    - Choose a preferred voice from the dropdown (e.g., Gregory or Ruth).
4. **Set Intent Classification Confidence Threshold**:

- Use the default value of 0.40.



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.

Danielle

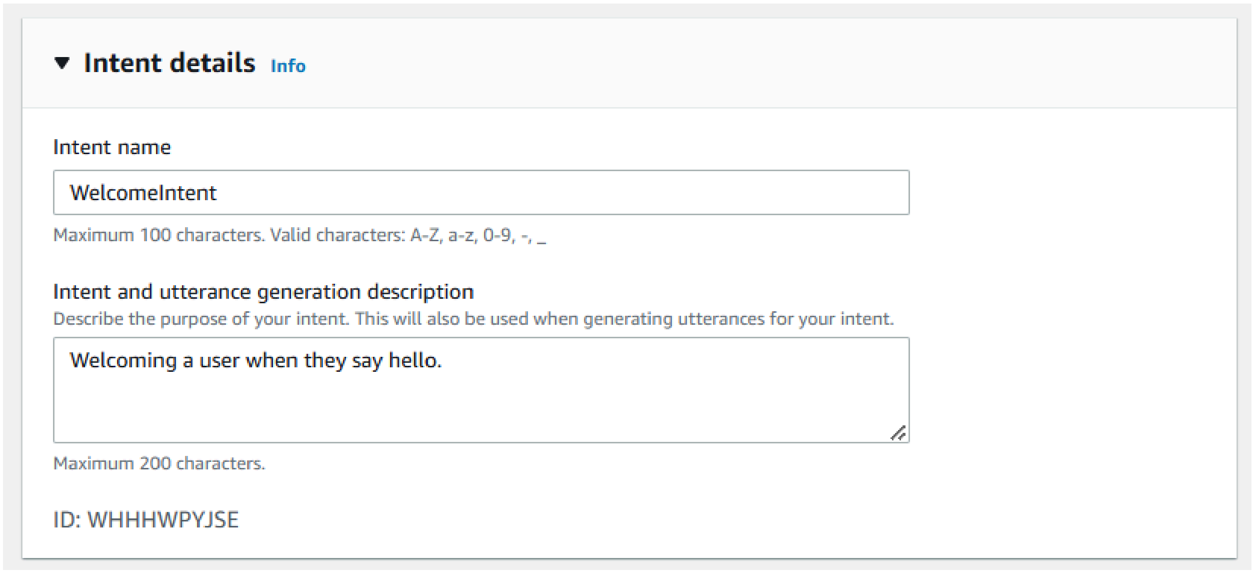
Voice sample: Hello, my name is Danielle. Let me know how I can assist you.

Intent classification confidence score threshold: 0.40

Min: 0.00, max: 1.00.

Creating Your First Intent

1. **Define the Intent**:
   - Rename "NewIntent" to "WelcomeIntent."
   - Add a description: "Welcoming a user when they say hello."



Intent details Info

Intent name

WelcomeIntent

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

Intent and utterance generation description

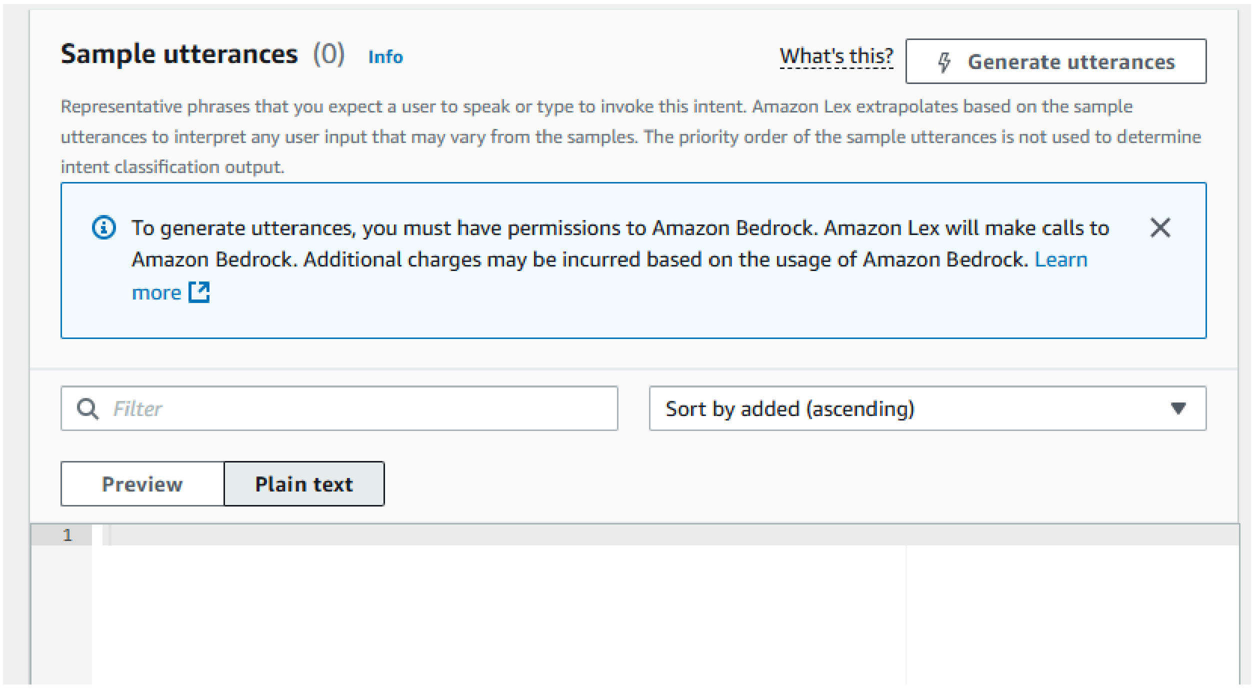
Describe the purpose of your intent. This will also be used when generating utterances for your intent.

Welcoming a user when they say hello.

Maximum 200 characters.

ID: WHHHWPYJSE

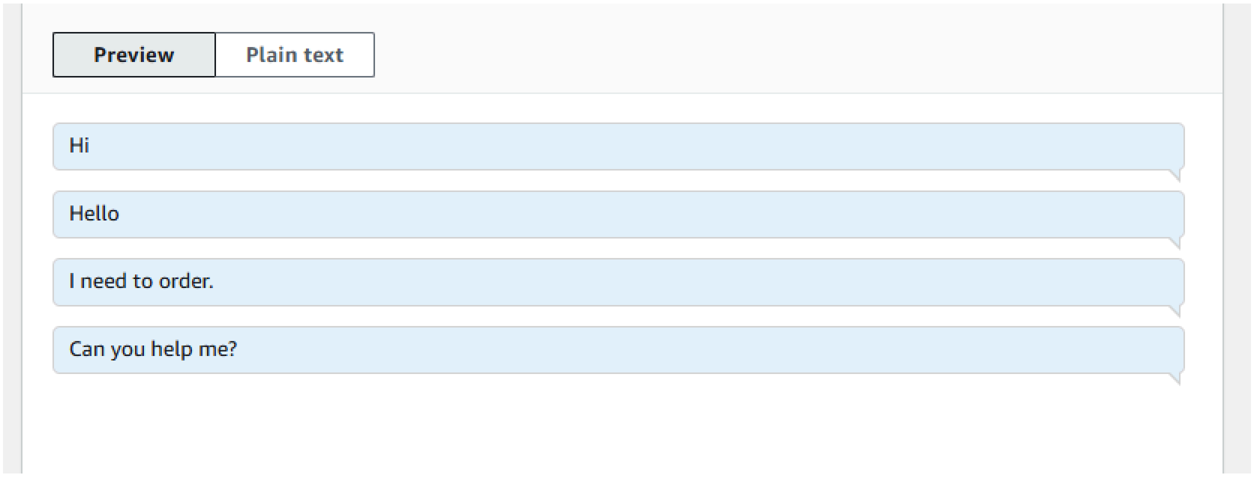
2. **Add Sample Utterances**:
   - Examples: "Hello," "Hi," "Good morning."



Sample utterances (0) 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



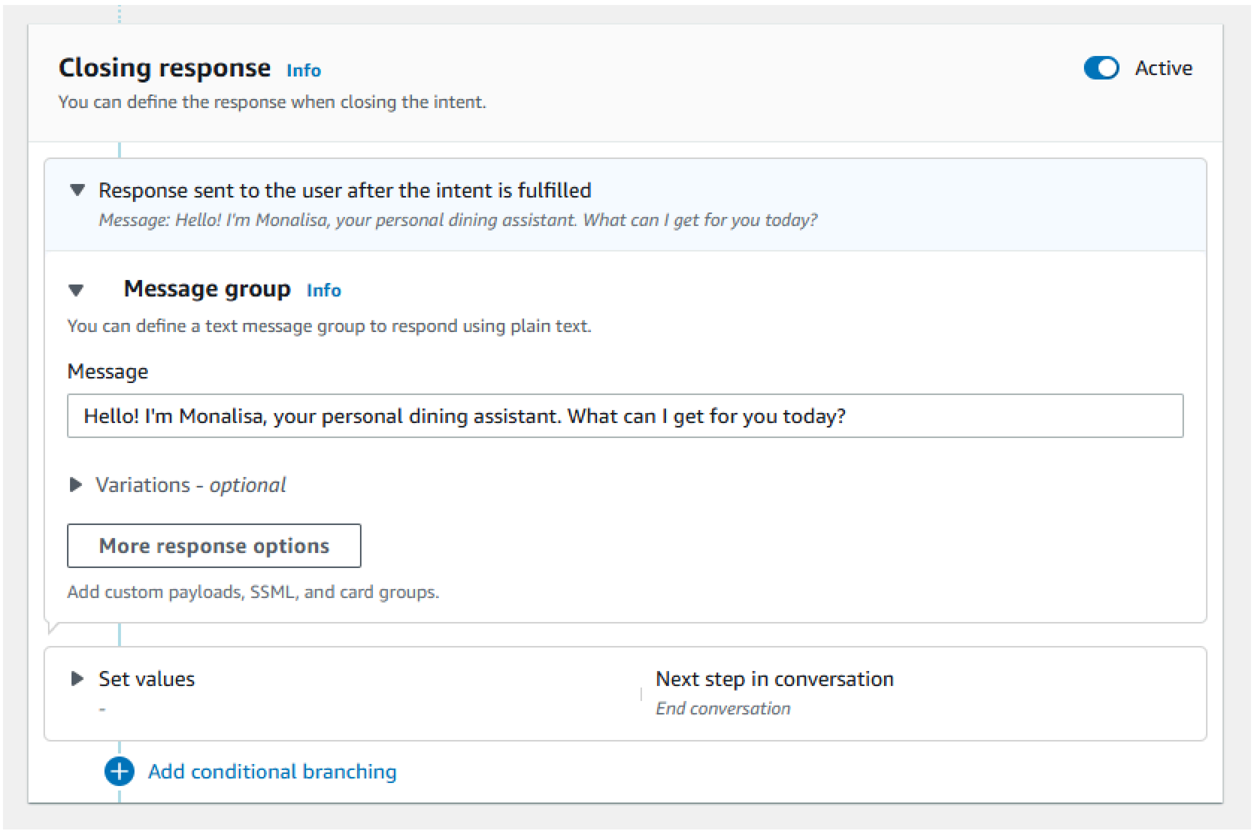
Hi

Hello

I need to order.

Can you help me?

3. **Set Responses**:
   - Closing response: "Hi! I'm your Dining Bot. How can I help you today?"



Closing response Info

You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled

Message: Hello! I'm Monalisa, your personal dining assistant. What can I get for you today?

Message group Info

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

Message

Hello! I'm Monalisa, your personal dining assistant. What can I get for you today?

Add custom payloads, SSML, and card groups.

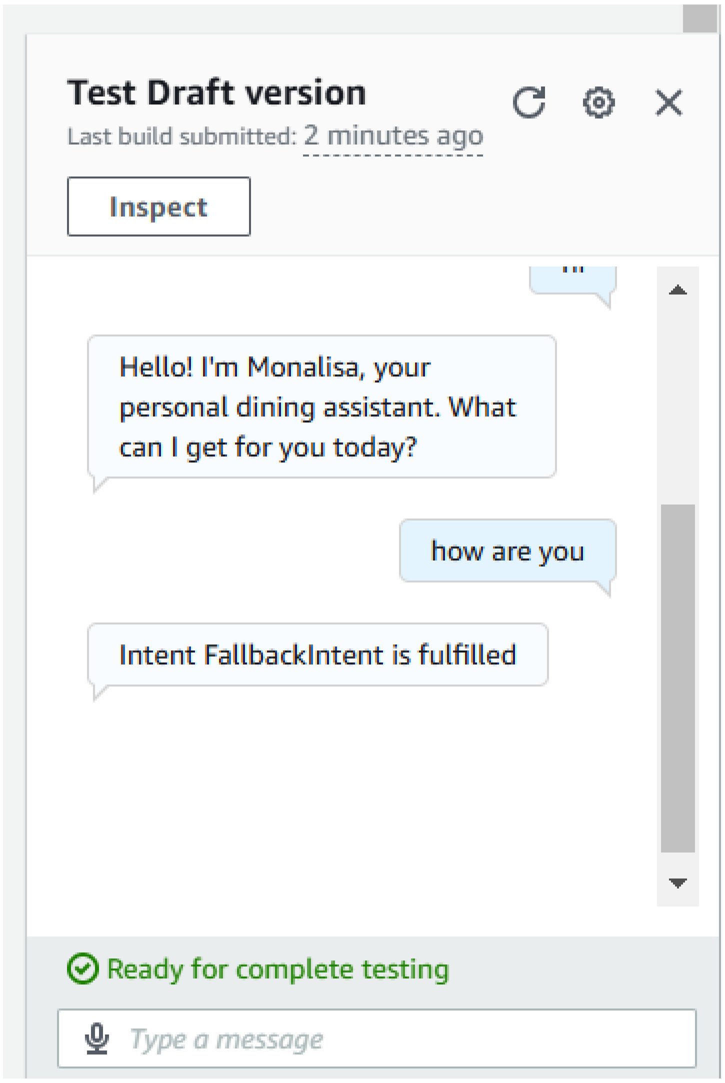
Set values

-

Next step in conversation

End conversation

4. **Build and Test**:
   - Save the intent and build the bot.
   - Use the test feature to try various greetings.


Test Draft version
Last build submitted: 2 minutes ago
Inspect

Hello! I'm Monalisa, your personal dining assistant. What can I get for you today?

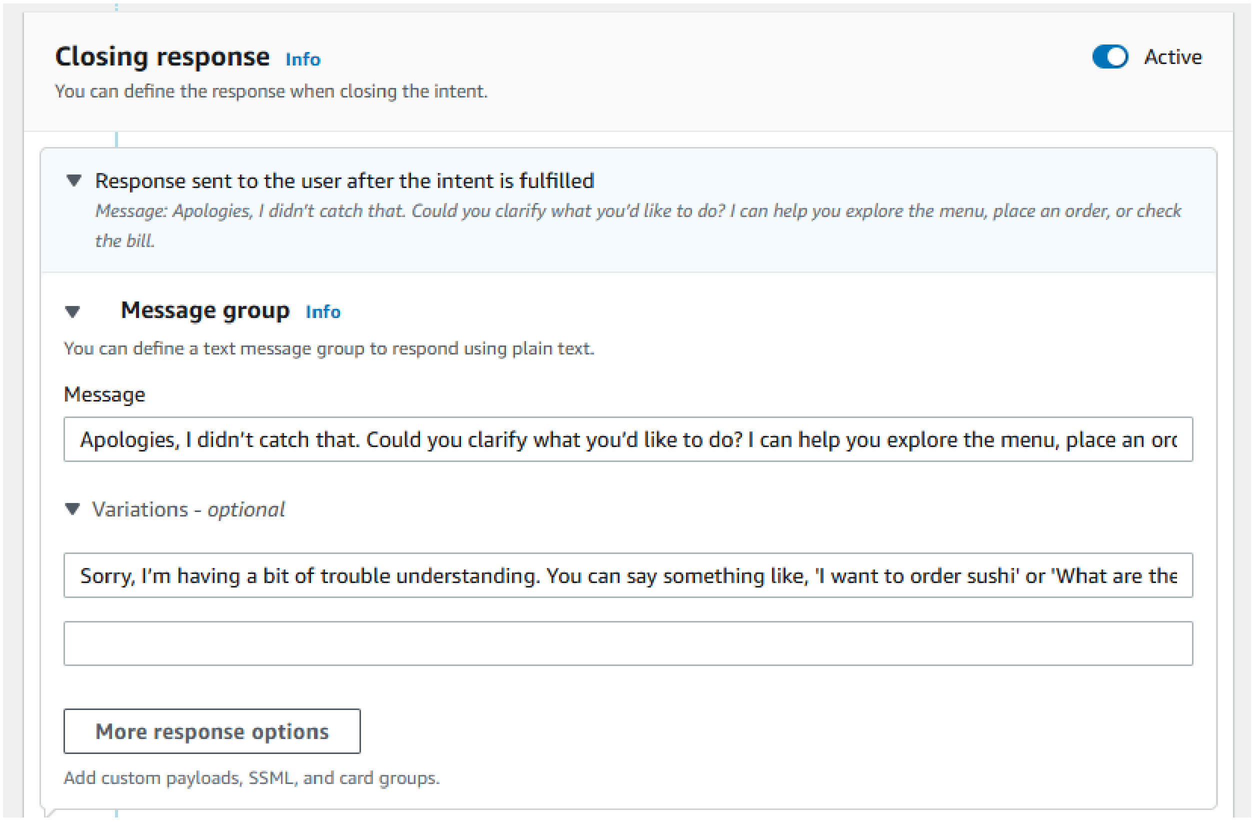
how are you

Intent FallbackIntent is fulfilled

Ready for complete testing
Type a message

Managing Fallback Intent

1. **Customize Fallback Responses**:
   - Navigate to the "FallbackIntent."
   - Replace the default message with user-friendly alternatives, e.g.,
     - "Sorry, I didn't understand that. Can you rephrase?"
     - "Hmm, could you try again? I can help with account balance, transfers, or payments."



Closing response Info

You can define the response when closing the intent.

Active

▼ Response sent to the user after the intent is fulfilled

Message: Apologies, I didn't catch that. Could you clarify what you'd like to do? I can help you explore the menu, place an order, or check the bill.

▼ Message group Info

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

Message

Apologies, I didn't catch that. Could you clarify what you'd like to do? I can help you explore the menu, place an orc

▼ Variations - *optional*

Sorry, I'm having a bit of trouble understanding. You can say something like, 'I want to order sushi' or 'What are the

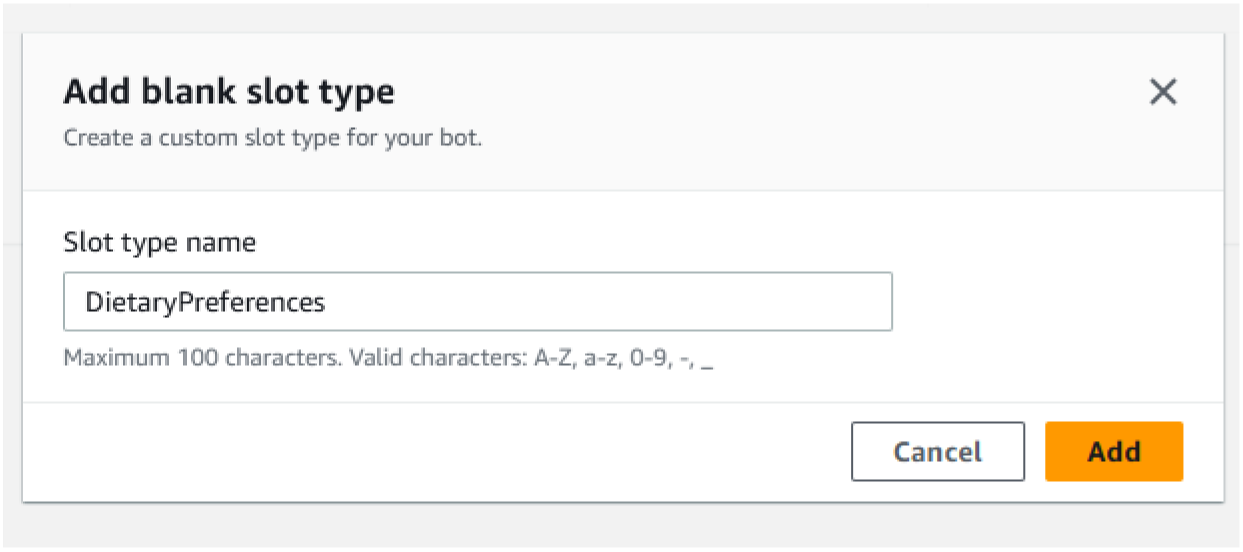
More response options

Add custom payloads, SSML, and card groups.

2. **Save and Build**:
   - Save the intent and rebuild the bot.

Creating Custom Slots

1. **Understand Slots**:
   - Slots store specific user-provided information (e.g., dietary preferences).
2. **Create a Slot Type**:
   - Navigate to "Slot types" and select "Add slot type."
   - Name it "dietaryPreference."


Add blank slot type

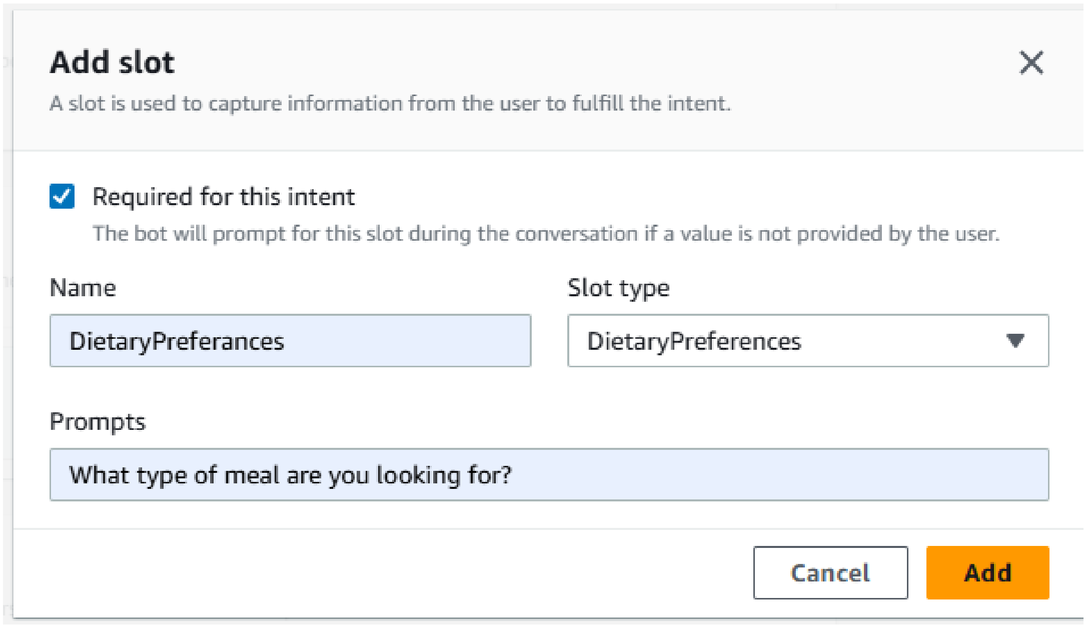
Create a custom slot type for your bot.

Slot type name: DietaryPreferences

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

- Add values and synonyms:
    - Veg (synonyms: Vegetarian)
    - Non-veg (synonyms: Meat)

3. **Use Slots in Intents**:
    - Add a slot to the "Order" intent:
        - Name: dietaryPreference
        - Prompt: "What are your meal preferences?"


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: DietaryPreferances

Slot type: DietaryPreferences

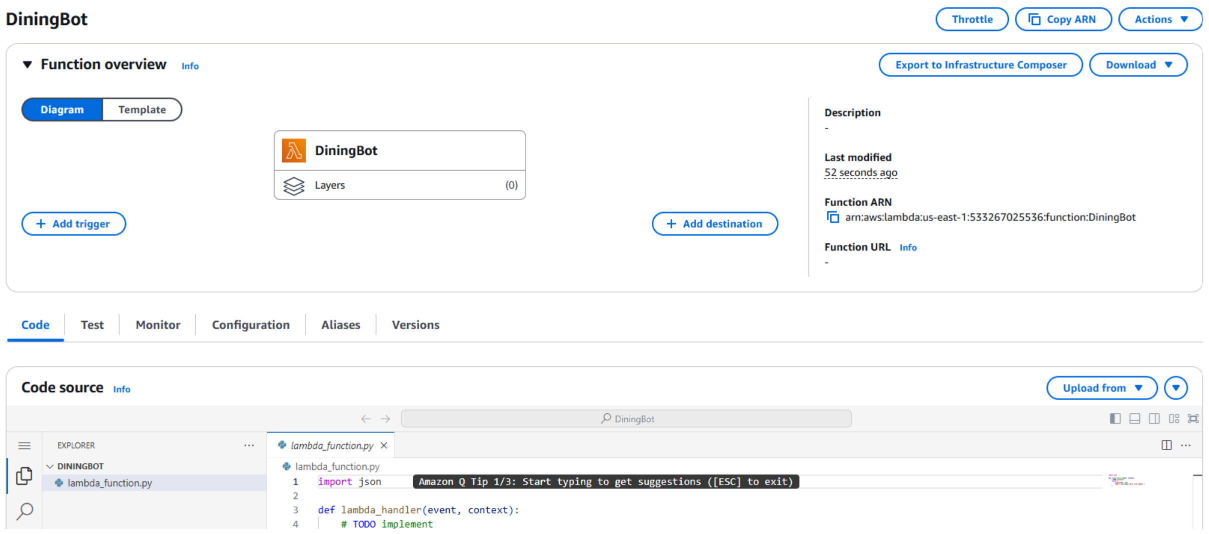
Prompts: What type of meal are you looking for?

4. **Save and Build**:

    - Save the slot type and rebuild the bot.

Integrating AWS Lambda

1. **Create a Lambda Function**:
    - Navigate to AWS Lambda and create a function:
        - Name: DiningBot
        - Runtime: Python 3.12 or later.



DiningBot

Function overview

Description

-

Last modified

52 seconds ago

Function ARN

arn:aws:lambda:us-east-1:533267025536:function:DiningBot

Function URL

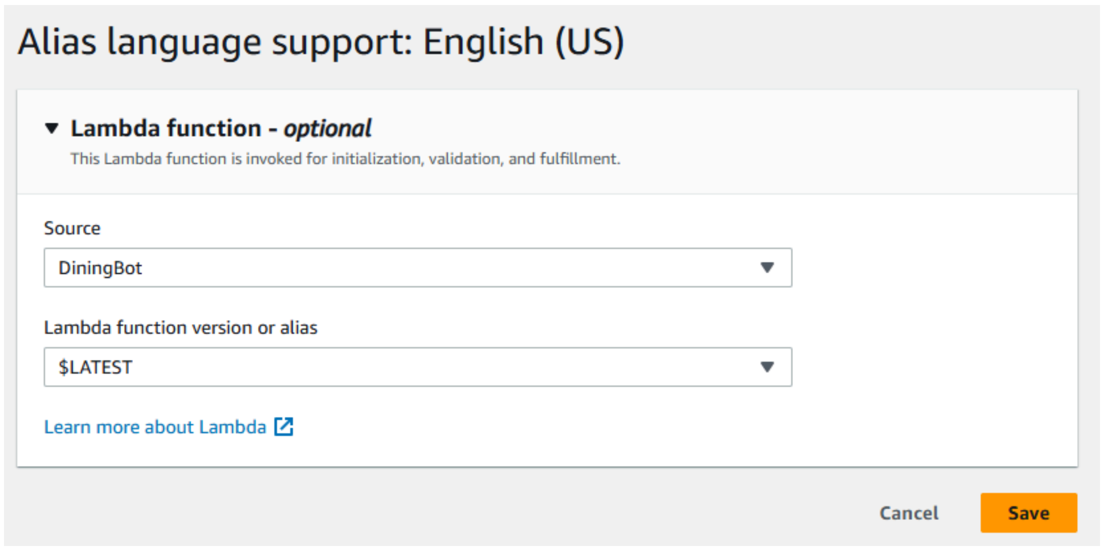
-

Code source

```
import json

def lambda_handler(event, context):
    # TODO implement
```

2. **Connect Lambda to Lex**:
    - In the Lex console, associate the Lambda function with the bot's alias.
    - Link the "Order" intent to the Lambda function under the Fulfillment settings.



Alias language support: English (US)

Lambda function - *optional*

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

Source

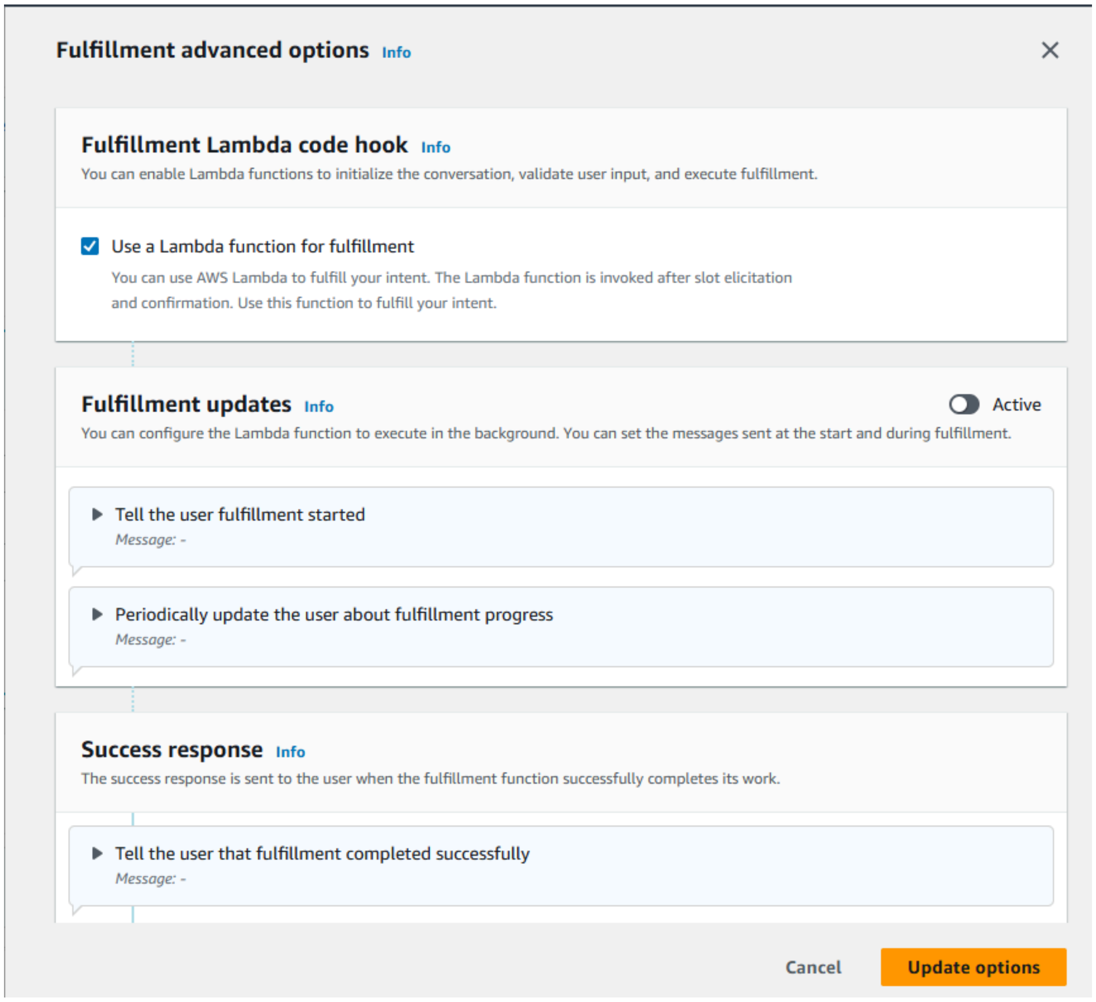
DiningBot

Lambda function version or alias

$LATEST

3. **Enhance Functionality**:

- Use Lambda to generate meal suggestions based on user preferences.


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.

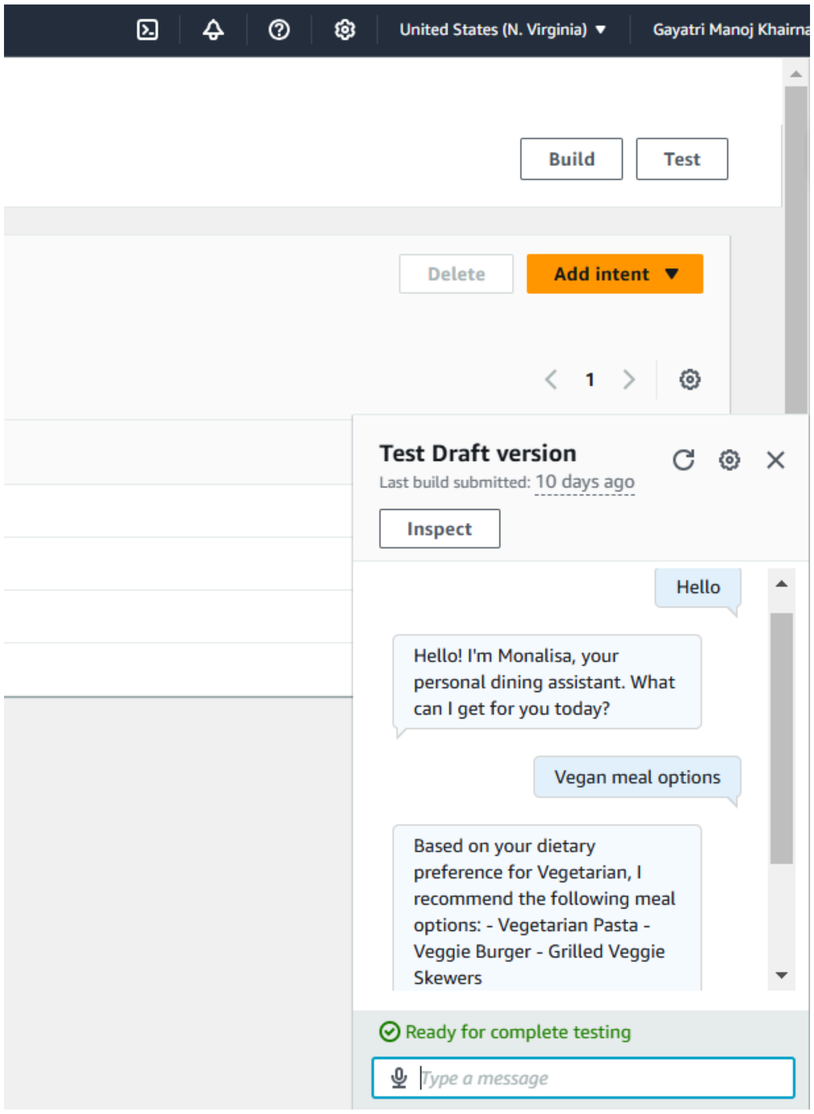
▶ Tell the user that fulfillment completed successfully
Message: -

Adding Contextual Follow-Up Intents

1. **Set Up a New Intent**:
   - Name: FollowupOrder
   - Description: "Intent for follow-up order checks without authentication."
2. **Add Slots**:
   - Slot Name: OrderId
   - Prompt: "What is your order ID?"
3. **Link Contexts**:
   - Use output contexts to pass information between intents.
4. **Save and Build**:
   - Save the intent and rebuild the bot.

Testing and Deployment

1. **Test the Bot**:
    - Use the test interface in Lex to simulate user interactions.
    - Verify the functionality of intents, slots, and fallback messages.
2. **Deploy the Bot**:
    - Publish the bot to a desired platform (e.g., web, mobile).
3. **Monitor and Improve**:
    - Use logs and analytics to refine the bot's performance.


United States (N. Virginia)
Gayatri Manoj Khairna
Build
Test
Delete
Add intent
1
Test Draft version
Last build submitted: 10 days ago
Inspect
Ready for complete testing
Type a message

Hello

Hello! I'm Monalisa, your personal dining assistant. What can I get for you today?

Vegan meal options

Based on your dietary preference for Vegetarian, I recommend the following meal options: - Vegetarian Pasta - Veggie Burger - Grilled Veggie Skewers

Conclusion

By following these steps, you can create a robust conversational bot using Amazon Lex. This guide serves as a reference for setting up intents, managing fallback responses, integrating Lambda, and deploying the bot effectively.