

Build a Hotel Reservation Agent



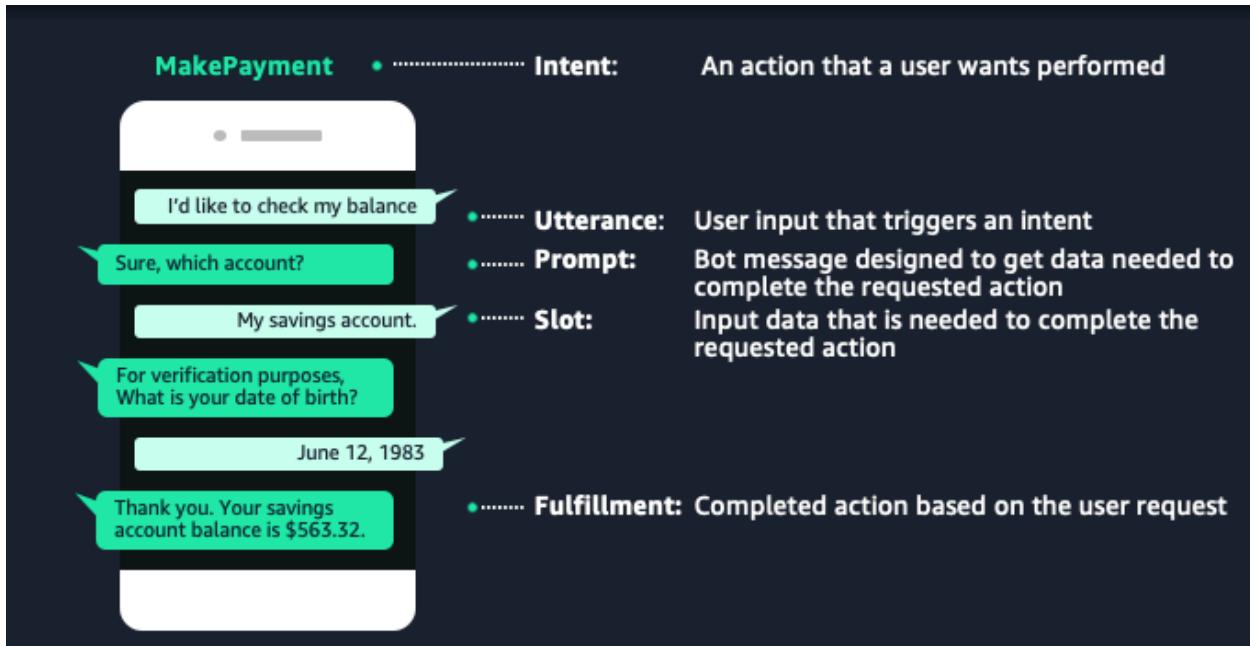
Welcome to the AWS NLP Summit! This is an introductory lab to walk through how to create a simple Hospitality-related chatbot that can provide hotel reservations. It will also show you how to integrate the chat bot with [AWS Lambda](#) functions, as well as putting data into DynamoDB, our Amazon native NoSQL database, and lastly integration with Amazon Connect and Facebook Messenger.



- **Lab Author:** Jake Wen, Solutions Architect
- **Co-Presenter:** Michelle Gibbs, Customer Success Manager
- **Expected duration:** 1.5 hours
- **Target audience:** Developers and Solutions Architects

Overview of Amazon Lex

Chatbot basic concepts:



1. **Intents** - this is an action that the user wants to perform. It enables the bot to understand and classify what task a user is trying to accomplish. A single bot can contain multiple different intents, allowing one bot to handle different - but most-likely related - requests from a user
2. **Utterances** - these are the textual representations of what a user has to type or say in order to trigger an intent. One intent can contain many different utterances, allowing users to trigger the bot using different phrases, potentially providing many different data parameters (or "slots") with their utterance
3. **Prompts** - Amazon Lex can define a different prompt for different stages of the conversation, such as when asking for a bank account number or a date for flight booking. They can also be overridden at run-time by your Lambda functions
4. **Slots** - these are the data items that the bot intent needs in order to be able to fulfill its task; e.g., for a flight booking intent you are likely to need slots for origin airport, destination airport, flight day, flight time, and booking class. Amazon Lex has a large number of built-in slot types, but you are free to create your own custom slot types for your use case
5. **Fulfillment** - this is the final state of the intent, normally with a closing message and some additional information for the user; e.g., for a flight booking you could confirm that the flight is booked, along with a brief summary of the route and the flight number that the user has been booked onto

Amazon Lex allows input using the following methods

- **Text** - a text string is submitted to Amazon Lex, taken from some client UI such as a Support chat-bot interface
- **Voice** - audio of a supported format is submitted to Amazon Lex

For both of these approaches it is strictly a one-to-one mapping of inputs and outputs - the user submits a request, Amazon Lex will process it and update the session state for that bot, and the result is returned. It is returned in the same fashion - text or voice. Voice responses are based upon an [Amazon Polly](#) voice that you have associated with your bot's language configuration. The response could be a request for more intent data, it could be a final fulfilment message, it could be a message dynamically created in your Lambda function or it could be, if the input request wasn't understood, some kind of static or dynamic clarification message.

Initial Bot Creation

Create the initial bot:

1. Before you login, make sure you follow the lab instruction and create the exact names listed on this document to avoid misconfiguration.
2. Login to the AWS Console using your event engine account and navigate to the Amazon Lex service, which can be done by typing Lex into the top search bar and select the Amazon Lex - Build Voice and Text Chatbots
3. Ensure that you are in the V2 console for Amazon Lex, which is indicated by the version number in the URL - in this example my current AWS Region is set to *N.Virginia (us-east-1)*.
4. Select the **Create bot** button on the top-right corner

The screenshot shows the 'Configure bot settings' page. It has two main sections: 'Creation method' and 'Bot configuration'. In the 'Creation method' section, the 'Create a blank bot' option is selected. In the 'Bot configuration' section, the 'Bot name' is set to 'HotelBot' and the 'Description - optional' field contains 'Example bot to provide Lex V2 functionality'.

Configure bot settings Info

Creation method

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
HotelBot
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.
Example bot to provide Lex V2 functionality
Maximum 200 characters.

5. Ensure that the **Create a blank bot** option is selected, and type in the bot's name **HotelBot** and an optional description. Note that the bot's name only accepts certain characters.

IAM permissions [Info](#)

IAM permissions 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.

i 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.

AWSRoleForLexV2Bots_K34UVAK6CF

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

6. For you **IAM permissions** you can select the **Create a role with basic Amazon Lex permissions** option - this will give create an IAM policy that gives Amazon Lex permissions to call other AWS services on your behalf, and it is just what we need for this workshop.
7. This bot has no COPPA implications, so you can just select **No** here.

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).

8. Amazon Lex will only maintain a session for a set length of time. If the user is idle and does not provide any input for this length of time then the session will end. For our workshop, we will leave this by default. Click **Next** to the next page.

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.

Ivy ▾

Voice sample

Hello, my name is Ivy. Let me know how I can assist you.

Play

Intent classification confidence score threshold

0.40

Min: 0.00, max: 1.00.

Cancel

Add another language

Done

9. You can select your language and your favorite voice here, for this workshop, we will leave it by default and click **Done**.

Create Your First Intent

Overview

We will work purely in English for this workshop, so we will not need to create any additional languages.

We will start off with the basic **Greeting** intent in this section, along with a simple **FallbackIntent**, and then increase the complexity of the bot by adding:

- Custom slot
- Custom intent
- Lambda function

Creating the Greeting Intent

Now you are in the **Intents** screen within Amazon Lex. You will see that the default **FallbackIntent** and **Newintent** has already been created, and that this initial version of the bot is the **Draft version**.

The screenshot shows the Amazon Lex console interface for creating a new intent. At the top, a green banner indicates "Successfully created bot: HotelBot". Below this, the "Intent details" panel is open, showing the intent name "Greeting" and a description "Helps users find their last devices". The "Sample utterances" panel is also visible, showing a list with one item: "No sample utterances". At the bottom, there are tabs for "Editor", "Visual builder", and "New...", and a prominent orange "Save Intent" button.

On the **Newintent** section, give it the name **Greeting** and click **Save Intent**.

This will rename the intent, so at this point add an optional description inside the **Intent details** panel, then scroll down to and open the **Sample utterances** panel, then click the **Plain text** button.

We are now going to add multiple utterances at once - when you are adding multiple utterances that you already have in a list this is the easiest method. Copy the text below, which represent the utterances that will trigger this intent, and paste it into the text window.

Hi
Hello
I need help
Can you help me?

Clicking on the **Preview** button will show these defined as four distinct utterances within this intent:

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.

Filter Sort by added (ascending) ▾

[Preview](#) [Plain Text](#)

Hi

Hello

I need help

Can you help me?

I want to book a flight [Add utterance](#)

Maximum 250 characters.

At this moment we have not defined any **Slots** for this intent because we are not intending to do any processing here, and we just want to open the conversation with a polite response.

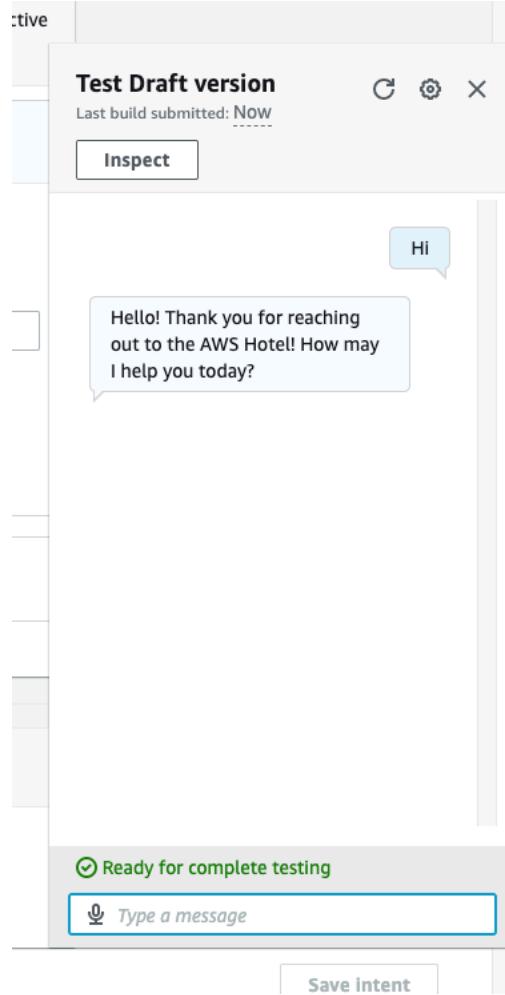
Now we need to create a closing message for the intent, which is what Amazon Lex will respond with if the intent processing is complete. Scroll down to **Closing responses**, expand the **Response sent to the user after the intent is fulfilled** section, and in the **Message** field enter the following message:

Hello! Thank you for reaching out to the AWS Hotel! How may I help you today?

The screenshot shows the AWS Lambda Functions console interface for creating a new intent. At the top right, there is an 'Active' status indicator. Below it, the section title 'Closing response' has an 'Info' link. A note says 'You can define the response when closing the intent.' To the right of the note is a blue circular icon with a white dot, labeled 'Active'. Under the 'Closing response' section, there is a collapsed section titled 'Response sent to the user after the intent is fulfilled' with a message example: 'Message: Hello! Thank you for reaching out to the AWS Hotel! How may I help you today?'. Below this is another collapsed section titled 'Message group' with an 'Info' link. It says 'You can define a text message group to respond using plain text.' Under this section, there is a 'Message' input field containing the same message as the previous section. Below the message input is a 'Variations - optional' section with a 'More response options' button. A note below it says 'Add custom payloads, SSML, and card groups.' In the bottom right corner of the main configuration area, there are two buttons: 'Set values' and 'Next step in conversation'. 'Set values' has a minus sign next to it, and 'Next step in conversation' has an 'End conversation' link. At the very bottom left of the configuration area is a blue plus sign button labeled 'Add conditional branching'.

Hit the **Save intent** button, followed by the **Build** button - this will then build the bot and enable you to test it. The process will take about 30 seconds.

Once built - assuming that there are no errors - you can test out your new greeting bot by selecting the **Test** button. The following dialog will pop up, and you can interact with the bot by entering your opening message.



Configure the Fallbackintent

The default message from the **FallbackIntent** is fairly simple - we need to update this so that the user is given a better message, which should be to ask them to re-phrase their question in a better way. Click on the **FallbackIntent** item and scroll down to **Closing responses**.

Expand the speech section for **Response sent to the user after the intent is fulfilled**, and in the **Message** field add the following text:

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

Closing response Info

You can define the response when closing the intent.

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?

▼ 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?

► Variations - *optional*

More response options

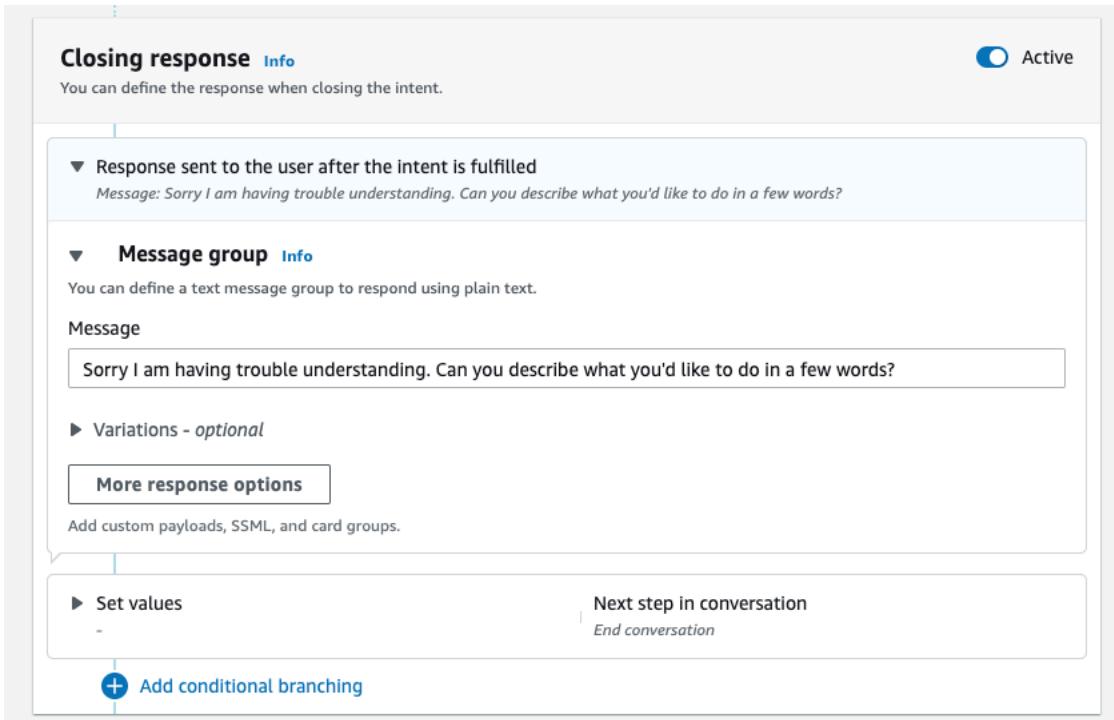
Add custom payloads, SSML, and card groups.

► Set values

-

Next step in conversation
End conversation

+ Add conditional branching



Hit the **Save intent** button, and click **Build**, once successfully built, then move to the next section.

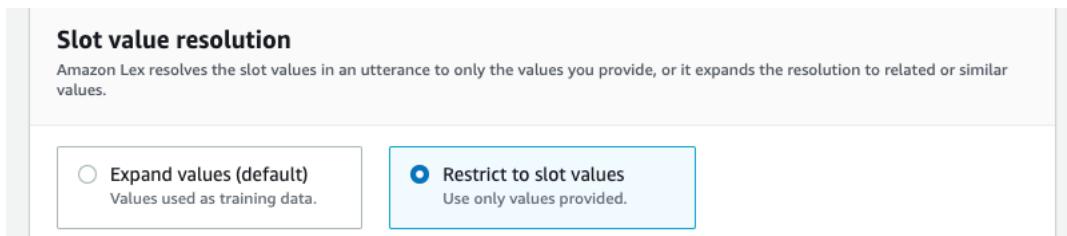
Create the HotelReservation Intent

Create a custom slot types

Now we are going to make this more powerful by adding the **HotelReservation** intent. This allows a user to make a hotel room reservation. We're going to support four different room types: **Standard, Queen, King and Double**. We can do this in one custom slot.

On the Amazon Lex Console, head to the **Slot types** panel. Click on the **Add slot type** button, and select the **Add blank slot type** - in the dialog that pops up please enter **HotelRoom** and click the **Add** button.

This will bring up a large **Slot types editor** panel, and the first thing to change is the **Slot value resolution**, which you want to set to **Restrict to slot values**.



This setting ensures that the only valid values possible are the ones that you supply in the next step. If you choose the **Expand values (default)** option then Amazon Lex will still honor the values that you supply, but if Amazon Lex's ML models see different values being given for these slots, then it will over time expand the resolution of slots to those values.

You can now add each of the hotel room types into the **Slot type values** panel - put each on one of the **Value** fields and click the **Add value** button.

Slot type: HotelRoom [Info](#)

A slot type is a list of values used to capture values for a slot.

► Slot type details

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)
Values used as training data.

Restrict to slot values
Use only values provided.

Slot type values

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

Search slot type values

King

Luxury Tab or ; for new value

X

Queen

Tab or ; for new value

X

Standard

Tab or ; for new value

X

Double

Two Bed Tab or ; for new value

X

Value

Tab or ; for new value

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

Use slot values as custom vocabulary [Info](#)

There are also synonyms for King and Double value. The idea here is now when a user enters any of these synonyms then Amazon Lex will correctly map it back to the correct value, which means you don't have to write a back-end integration code to convert multiple possible variations of this slot.

Now click on the **Save slot type** button and your first custom slot is saved, and it is available for any intents within this bot language.

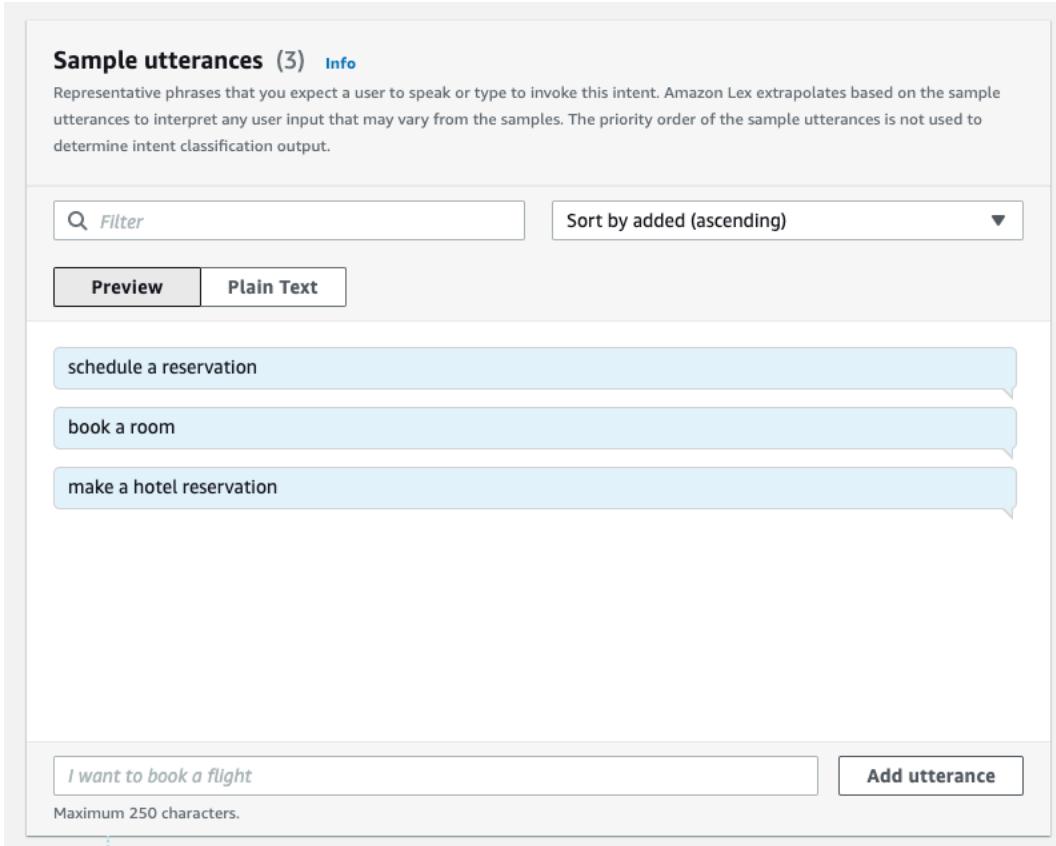
Create the HotelReservation intent

We are now ready to create the intent to make hotel reservation, the process here will be very similar to the **Greeting** intent.

Under **HotelBot** on the left-hand menu, click on **Intents**, then click the **Add intent** button, select *Add empty intent*, and for the **Intent Name** please enter **HotelReservation**, then click the **Add** button. Once the **Intent editor** panel appears, enter the following description in the **Intent details** pane: Intent to make hotel reservation

Scroll down to **Sample utterances**. switch to **Plain Text** and paste in the following utterances. Once entered, click the *Preview* button, then click on the **Save intent** button.

schedule a reservation
book a room
make a hotel reservation



The screenshot shows the 'Sample utterances' section of the Amazon Lex Intent Editor. It displays three utterances in a list:

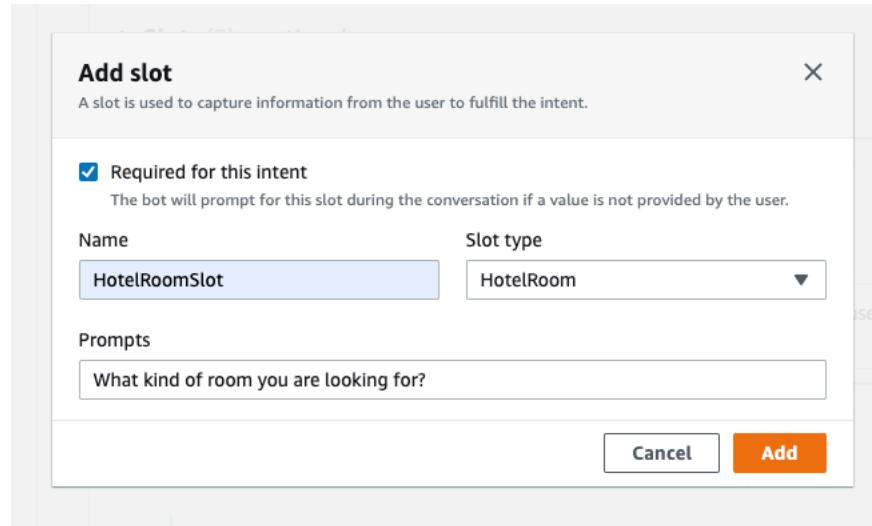
- schedule a reservation
- book a room
- make a hotel reservation

Below the list is a text input field containing "I want to book a flight" and a note indicating a maximum of 250 characters. There is also a "Add utterance" button.

Next you will scroll down to the **Slots** section. On this section, you will need to create couple slots for the bot to fulfill the required information.

Click **Add Slot**, and name the slot **HotelRoomSlot**. On the slot type section, choose the **HotelRoom** slot that we created earlier. and enter the following for the prompt:

What kind of room you are looking for?



Now that we have our custom slot, we want to repeat this entire process, but this time we will use one of the built-in slot types. Start by clicking the **Add slot** button, and this time use these values and then click the **Add** button:

- **Name** - HotelDate
- **Slot type** - AMAZON.Date
- **Prompts** - What day will you arrive?

Then we will repeat the same process to create another one:

- **Name** - HotelDuration
- **Slot type** - AMAZON.Duration
- **Prompts** - How long are you going to stay?

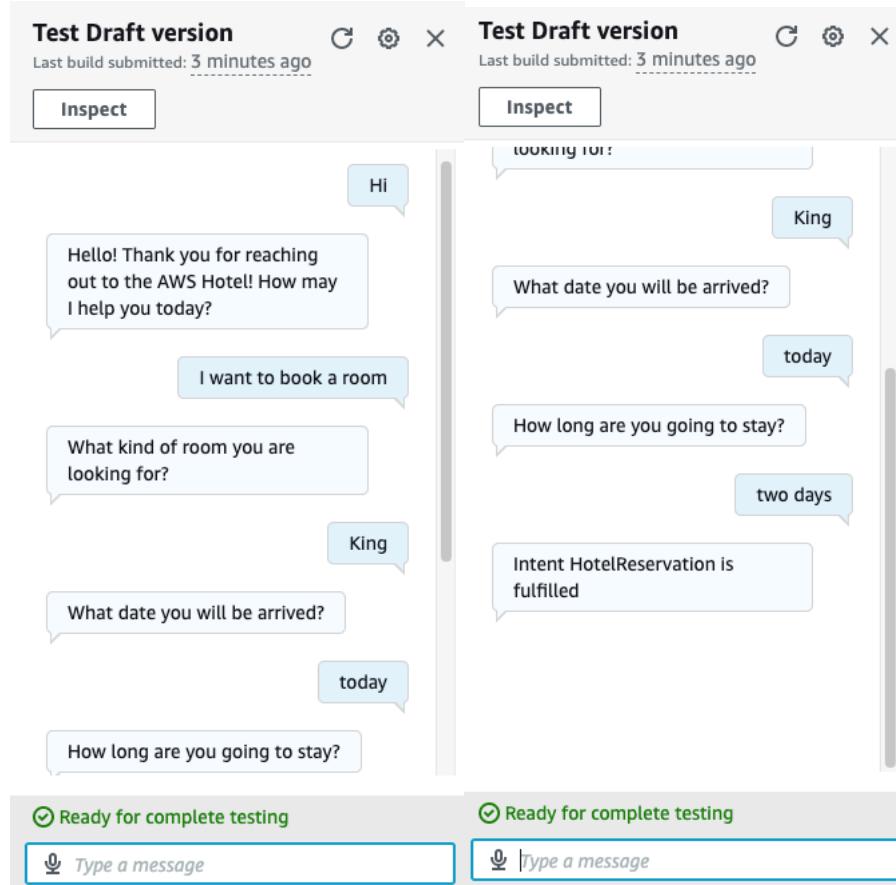
You should now see the following slots in the system. You are able to open up each one to edit the value.

The screenshot shows the 'Slots (3) - optional' section of the Amazon Lex configuration interface. It includes a 'Filter' search bar and a list of three slots:

- Prompt for slot: HotelRoomSlot**
Message: What kind of room you are looking for?
Slot type: HotelRoom
- Prompt for slot: HotelDate**
Message: What date you will be arrived?
Slot type: AMAZON.Date
- Prompt for slot: HotelDuration**
Message: How long are you going to stay?
Slot type: AMAZON.Duration

An 'Add slot' button is located in the top right corner.

Now you can go through the save / build / test cycle and let's try this out, use one of the utterances such as “Make hotel reservation”. Amazon Lex will then prompt you in order. It does this using the prompts that you have defined in the intent slots, and the result looks like this.



In this example we have created a bot to collect information from the users, in the next section we will be creating a lambda function to integrate with Dynamodb, so once the chatbot captures the data, it will automatically push the data to the lambda function and store the data in the Amazon NoSQL database DynamoDB.

Create the bot Lambda function

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code only when needed and scales automatically, from a few requests per day to thousands per second - all you need to do is supply your code in one of the languages that Lambda supports.

In Amazon Lex V2 you now create a single Lambda function per language per chatbot, which must be able to support both types of Amazon Lex interaction:

1. **Initialization/Validation** - Lambda is called at every turn of the conversation. This allows you to initialize, validate or override slot data values.
2. **Fulfillment** - Lambda is called to alter all slots that have been elicited and any confirmation messages that have been okayed. This is the time for Lambda to deliver "the result" of the Amazon Lex intent.

The Lambda function must also be able handle the required logic for each intent. Each time Amazon Lex calls a Lambda function it will include everything that the bot currently knows about the session - this includes, but is not limited to, all elicited slot values, the current user utterance, the intent and slot currently being elicited, and information from some prior intent triggers during this user session.

Creating the Lambda function

For this Workshop we are using a simple python file - please download the code at the following link to your local computer, as we will need it later in this section.

<https://github.com/jkwnamazon/aws-nlp-2022-lex.git>

Navigate within the AWS Console to the AWS Lambda service - type in **Lambda** in the console search bar and select **Lambda - Run Code without Thinking about Servers**. On the Lambda page click on the **Create function** button to begin the creation of your function.

We only need to set a few parameters on this screen, so enter the following pieces of data and click on the **Create function** button on the bottom-right of the screen

- Select **Author from scratch** on the radio-button bar
- **Function name - HotelRoomBooking**
- **Runtime - Python 3.9**

Leave everything else by default and click **Create Function**.

```

1 import json
2 import random
3 import decimal
4 import boto3
5
6 client = boto3.client('dynamodb')
7
8 def random_num():
9     return(decimal.Decimal(random.randrange(1000, 50000))/100)
10
11 def get_slots(intent_request):
12     return intent_request['sessionState']['intent']['slots']
13
14 def get_slot(intent_request, slotName):
15     slots = get_slots(intent_request)
16     if slots is not None and slotName in slots and slots[slotName] is not None:
17         return slots[slotName]['value']['interpretedValue']
18     else:
19         return None
20
21 def get_session_attributes(intent_request):
22     sessionState = intent_request['sessionState']
23     if 'sessionAttributes' in sessionState:
24         return sessionState['sessionAttributes']
25
26     return {}
27
28 def elicit_intent(intent_request, session_attributes, message):
29     return {
30         'sessionState': {
31             'dialogAction': {
32                 'type': 'ElicitIntent'
33             }
34         }
35     }

```

Now delete the default code, copy and paste the code from the python file from Github repo to the lambda function. Once you done, click **Deploy**.

General configuration	Execution role
Triggers	Role name HotelRoomBooking-role-grtgbv2l
Permissions	Edit
Destinations	

Now click the **Configuration** section and go to the **Permission**, click on the HotelRoomBooking-role-xxxxxx.

HotelRoomBooking-role-grtgbv2i

[Delete](#)

Summary

[Edit](#)

Creation date
September 18, 2022, 10:34 (UTC-07:00)

ARN
[arn:aws:iam::669793102817:role/service-role/HotelRoomBooking-role-grtgbv2i](#)

Last activity
None

Maximum session duration
1 hour

[Permissions](#)[Trust relationships](#)[Tags](#)[Access Advisor](#)[Revoke sessions](#)

Permissions policies (1)

You can attach up to 10 managed policies.

[Simulate](#)[Remove](#)[Add permissions](#)

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-80e5c7a4-6f75-4d90-b763-a412e43bb0cb	Customer managed	

Now you are in the IAM Role console, you will need to attach a DynamoDB access to this IAM Role, so the lambda function could be executed and store the data to the DynamoDB database. Click **Add permissions** and **Attach policies**.

Attach policy to HotelRoomBooking-role-grtgbv2i

▶ Current permissions policies (1)

Other permissions policies (Selected 1/770)

[Create policy](#)

<input type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/>	AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to DynamoDB streams and write permissions to CloudWatch logs.
<input type="checkbox"/>	AmazonDynamoDBReadOnlyAccess	AWS managed	Provides read only access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/>	AWSLambdaInvocation-DynamoDB	AWS managed	Provides read access to DynamoDB Streams.

Search DynamoDB on the search bar, choose the **AmazonDynamoDBFullAccess** and click **Attach policies**. Now your IAM role has the DynamoDB write access.

Configure bot Language configuration for Lambda

Now head back to the Amazon Lex Console, Lambda functions are defined on a per language basis, but they can be changed to different functions as your Amazon Lex bot evolves over time. Hence, you actually associate a Lambda function with an alias of your bot - this essentially represents a version of your bot. You will have noticed previous that your bot is the Draft version, and that is the version that we want to associate this with.

On the Amazon Lex console click on the **HotelBot**, then on the left-hand menu select the **Aliases** link, which will bring up a full list of aliases defined for your bot.

The screenshot shows the 'Aliases' page in the Amazon Lex console. The navigation path is Lex > Bots > Bot: HotelBot > Aliases. At the top right are 'Delete' and 'Create alias' buttons. Below is a descriptive text: 'An alias points to a specific version of your bot. With an alias, you can update the bot version that your client applications use.' A search bar labeled 'Search alias name' is present. The main table has columns: 'Alias name', 'Created', and 'Associated version'. One row is shown: 'TestBotAlias' (selected), '17 days ago', and 'Draft version'. Navigation controls <, 1, > and a refresh icon are at the bottom of the table.

Alias name	Created	Associated version
TestBotAlias	17 days ago	Draft version

From here you choose the default **TestBotAlias**, then from the *Languages* panel select **English (US)**. This will bring up a dialog that allows you to associate the Lambda function to this Language/version combination of your bot. Select your Lambda function **HotelRoomBooking** in the Source drop-down field, leave the **Lambda function version or alias** at the default **\$LATEST**. Click the **Save** button.

Alias language support: English (US)

▼ Lambda function - optional

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

Source

HotelRoomBooking

Lambda function version or alias

\$LATEST

[Learn more about Lambda](#)

Cancel Save

In Lambda you are able to create multiple versions of the same function, as well as aliased version names - your Amazon Lex bots can be associated with any of these specific versions, but if you use the `$LATEST` version then you will always use the most up-to-date version of the Lambda function - in development environments this is often the easiest way to work.

Enabling Lambda code hooks on the intents

The Lambda function is now ready to work on the **HotelBot** intents, but we still have to tell Amazon Lex which intents are to call the Lambda function and which tasks it should call upon to perform. Navigate to your intent by clicking on the **English (US)** link on the *Languages* pane, then on the far-left menu click on the *Intents* link to list your intents.

We will only associate the Lambda function with the **HotelReservation** intent - click on that intent and scroll down to the **Fulfillment** pane. Click on the arrow next to **On successful fulfillment**, followed by the **Advanced options** button, which will bring up a long list of fulfillment options. Go to the one marked *Fulfillment Lambda code hook* and click the checkbox to enable a fulfillment Lambda for this intent, then click on the **Update options** button at the bottom of the panel.

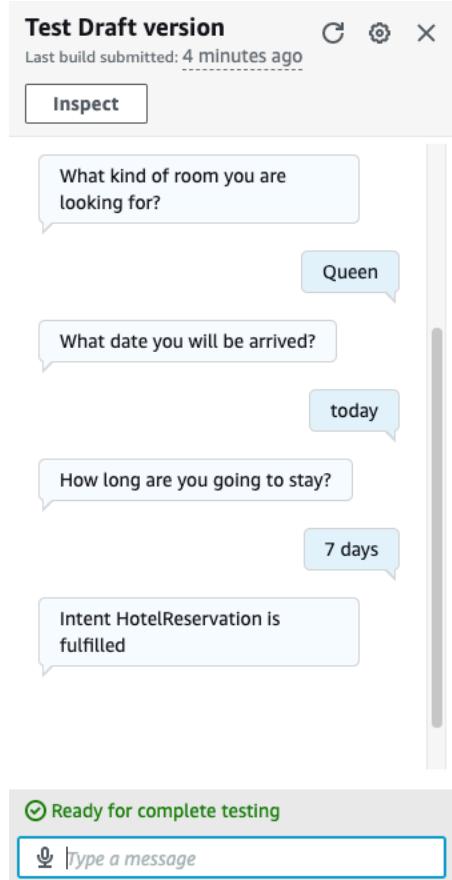
Fulfillment Lambda code hook Info

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.

Go through the save / build / test cycle again, and ask for make a hotel reservation - you should now see a result like the following test dialog.



Create a DynamoDB database

Amazon DynamoDB is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications at any scale. DynamoDB offers built-in security, continuous backups, automated multi-Region replication, in-memory caching, and data import and export tools.

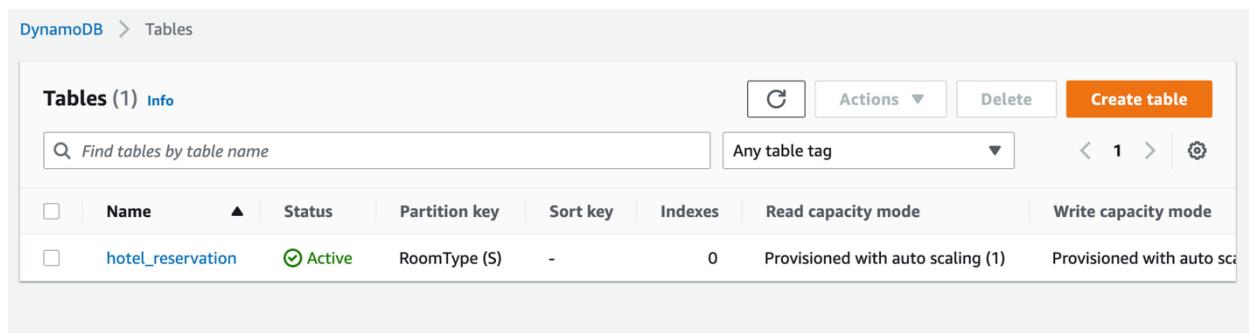
Head to the AWS console search bar, type DynamoDB and click it. Once you jump into the DynamoDB page, navigate to the **Tables** section.

Click **Create table**, enter the following information:

Table name: hotel_reservation

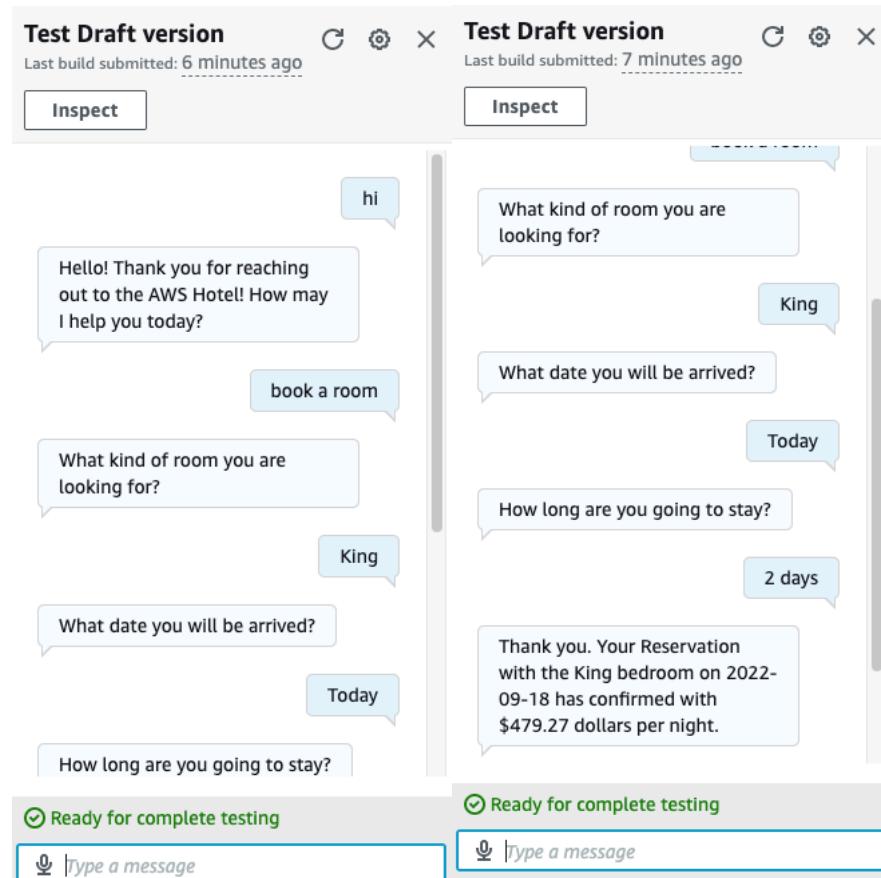
Partition key: RoomType

And select **String** for the partition key value. Then leave the Table setting as **default settings**. Scroll down and click **create table**.



The screenshot shows the AWS DynamoDB 'Tables' page. At the top, there is a header with 'DynamoDB > Tables'. Below the header, there is a search bar labeled 'Find tables by table name' and a dropdown menu labeled 'Any table tag'. On the right side of the header, there are buttons for 'Actions', 'Delete', and 'Create table'. The main area displays a table titled 'Tables (1) info'. The table has columns: Name, Status, Partition key, Sort key, Indexes, Read capacity mode, and Write capacity mode. A single row is shown for the table 'hotel_reservation', which is marked as 'Active' and has 'RoomType (S)' as its partition key. The 'Read capacity mode' is listed as 'Provisioned with auto scaling (1)' and the 'Write capacity mode' is listed as 'Provisioned with auto scaling'.

The database table status should be **Active** within a minute. Please verify your table name and the partition key are exactly the same as the one provided on the instruction. This will be used by the lambda function later.



Now head back to the Amazon Lex V2 and test the **HotelBot** again. This time because the Lex triggered the lambda function, so you will see the confirmation message from the lambda function indicate the thank you note, confirmation details and hotel pricing.

hotel_reservation

Autopreview View table details

▼ Scan/Query items

Scan/Query a table or index

Scan Query hotel_reservation ▾

▶ Filters

Run Reset

⌚ Completed Read capacity units consumed: 0.5

Items returned (1)

C Actions ▾ Create item

< 1 > ⚙️ ✖️

	RoomType	DateArrival	HotelDuration
<input type="checkbox"/>	King	2022-09-18	P2D

Great! You are almost there! Now you will head back to the DynamoDB database, click the database's name to get into the database. Then click the **Explore table items** option. Now you will see your reservation data stored in the DynamoDB table.

Create an Amazon Connect Instance

On the top of the AWS console, search **Amazon Connect** on the search bar and click it. Now you will be presented with an Amazon Connect introduction screen. Select **Create instance**.

The screenshot shows the 'Set identity' step of the 'Create Amazon Connect instance' wizard. On the left, a sidebar lists steps: Step 1 (Set identity), Step 2 (Add administrator), Step 3 (Set telephony), Step 4 (Data storage), and Step 5 (Review and create). The main area is titled 'Set identity' and contains the 'Identity management' section. It offers three options:

- Store users in Amazon Connect**: Create and manage users in Amazon Connect. You cannot share users with other applications.
- Link to an existing directory**: Amazon Connect uses an existing directory. You create users in the directory, and then add and configure them in Amazon Connect. You can only associate a directory with only one Amazon Connect instance. [Learn more](#)
- SAML 2.0-based authentication**: AWS supports identity federation with Security Assertion Markup Language (SAML 2.0). This feature enables single sign-on (SSO) so users can log into the AWS Management Console or call the AWS APIs without you having to create an IAM user for everyone in your organization. [Learn more](#)

Below this is the 'Access URL' section, which says 'Create a custom URL. Use this URL to log into this instance of Amazon Connect.' A text input field contains 'https://nlp2022-jw.my.connect.aws'. At the bottom right are 'Cancel' and 'Next' buttons.

For **Set identity** – Feel free to enter a unique instance name and choose **Next**.

Step 1

[Set identity](#)

Step 2

Add administrator

Step 3

[Set telephony](#)

Step 4

[Data storage](#)

Step 5

[Review and create](#)

Add administrator

Add administrator

Administrator - optional

Specify an administrator

Specify an administrator for this instance of Amazon Connect. The administrator will have full permissions to access all of Amazon Connect.

No administrator

First name

Jake

Last name

Wen

Username

jake.wen

Password

.....

Password (verify)

.....

Email

jkwn@amazon.com

[Cancel](#)

[Previous](#)

Next

Add administrator - Enter details for creating an administrator user for your new instance. Choose **Next** once done.

Step 1

[Set identity](#)

Step 2

[Add administrator](#)

Step 3

Set telephony

Step 4

Data storage

Step 5

Review and create

Set telephony

Telephony Options

Choose whether your contact center allows inbound calls, outbound calls, or both.

Allow incoming calls

Allow outgoing calls

[Cancel](#)

[Previous](#)

Next

Set telephony - Keep the default options selected to allow both incoming and outgoing calls for your instance. Choose **Next**.

Step 1

[Set identity](#)

Step 2

[Add administrator](#)

Step 3

[Set telephony](#)

Step 4

[Data storage](#)

Step 5

[Review and create](#)

Data storage

▼ Data storage

Call recordings, scheduled reports, and chat transcripts are stored in a S3 bucket that is created for you when you create an Amazon Connect instance. The stored data is encrypted by the AWS Key Management Service using a key specific to your Amazon Connect instance. Contact flow logs are stored in Amazon CloudWatch Logs in a log group created for you.

Amazon Connect permissions

By choosing Next, you are granting Amazon Connect permission to:

- Read and write to your S3 bucket.
- Read and write CloudWatch Logs.
- Encrypt your data.

Connect data

Your Connect data will be stored in this S3 bucket:

amazon-connect-38dda6337e6d/connect/nlp2022-jw

[Copy](#)

Contact flow logs

Your contact flow logs will be stored here in CloudWatch:

/aws/connect/nlp2022-jw

[Copy](#)

Enable Customer Profiles

Customer Profiles uses your customer data, including Connect contact history, to identify and help personalize contact flows and your agent's interactions with contacts. You can further customize your Customer Profile domain later, including adding more data sources and changing data encryption settings. [Learn more](#)

[Customize data storage \(advanced\)](#)

[Cancel](#)

[Previous](#)

[Next](#)

Data storage - Keep the default options selected for data storage. Choose **Next**.

Review and create - Review all settings and information before choosing **Create instance** to create your instance. The access URL for your instance is also shown on this screen.

Success!
Your `nlp2022-jw` instance has been created.
You can now choose phone numbers, accept calls, and engage with your customers.

Get started X

Amazon Connect > Instances

Amazon Connect virtual contact center instances

Instances	C	Delete	Add an Instance	
<input type="text"/> Find resources				
Instance alias	Access URL	Channels	Create date	Status
<input checked="" type="radio"/> nlp2022-jw	https://nlp2022-jw.my.connect.aws	Inbound, outbound telephony	9/18/2022	Active

Before you login in to the Amazon Connect page, click the **instance alias** and go to the **Contact flows** option.

Amazon Connect X

Amazon Connect > nlp2022-jw > Contact flows

Contact flows

Contact flow security keys

ⓘ Amazon Connect can encrypt sensitive data collected by contact flows using public-key cryptography. You can provide a X.509 certificate within your contact flow to encrypt the data that was captured using the Store customer input block. You must upload a signing key that is in PEM format to use this feature. The signing key will be used to verify the signature of the certificate used within the contact flow. Note: You may have up to two signing keys active at once to facilitate rotation. [Learn more](#)

Key ID	Creation date	Last accessed

Amazon Lex

Integrate Amazon Lex bots into your contact flows to take advantage of the same speech recognition and natural language understanding technology that powers Alexa. By adding Lex bots, you are granting Amazon Connect permission to interact with them. [Create a new Lex bot](#)

Region	Bot
US East: N. Virginia	HotelBot

Alias

TestBotAlias

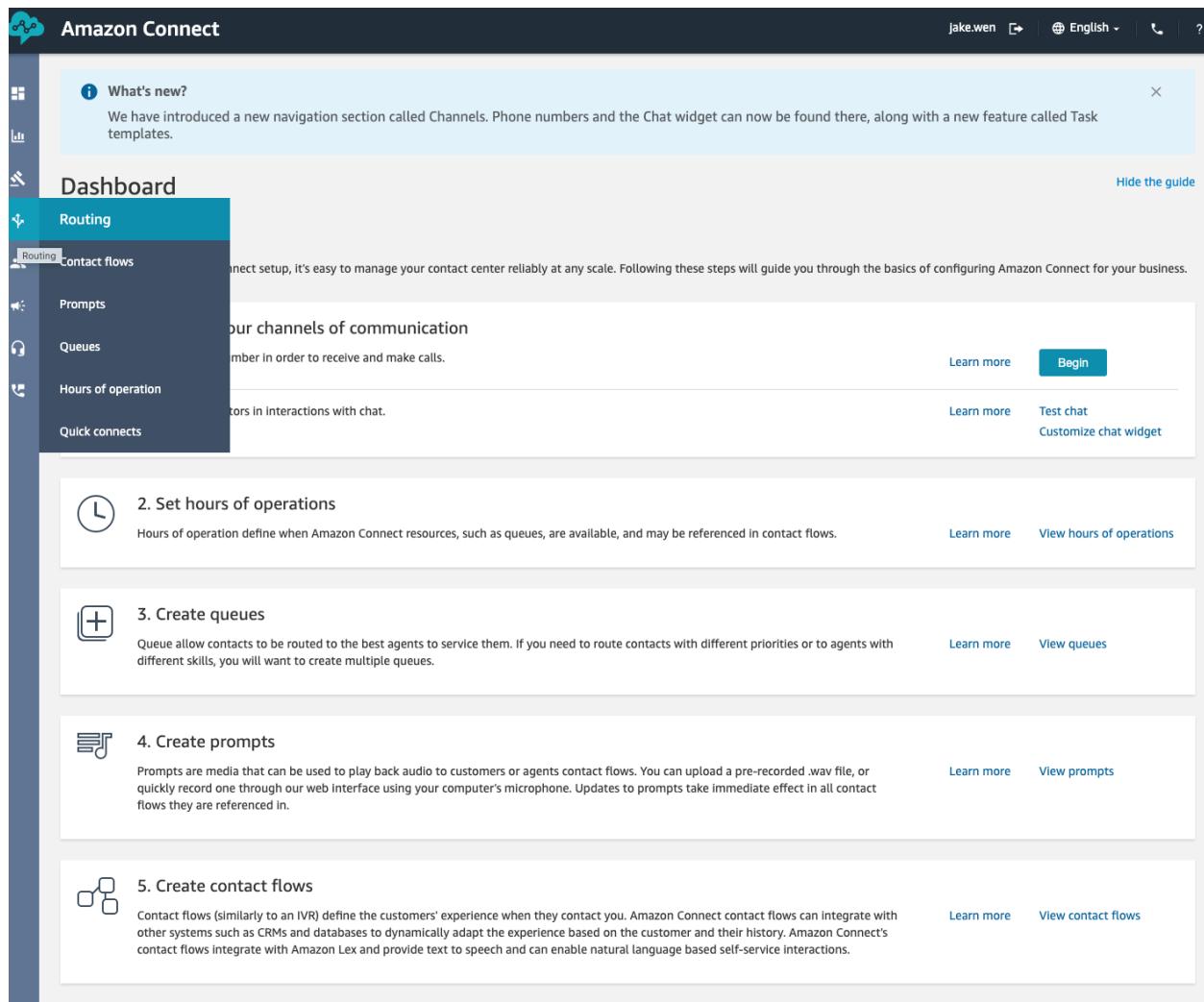
+ Add Amazon Lex Bot

Amazon Lex bot

On this page, choose the HotelBot we just created, picked the TestBotAlias and click **Add Amazon Lex Bot**.

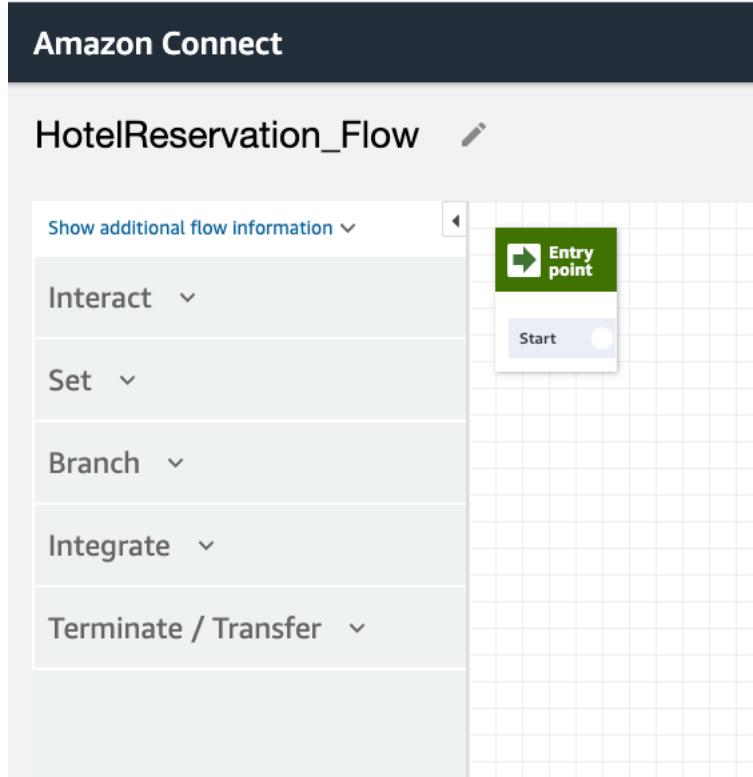
Build a contact flow to use Lex Bot

Open the Amazon Connect console and click the **Access URL**, enter your username and password we just created a moment ago to login to the Amazon Connect Console.

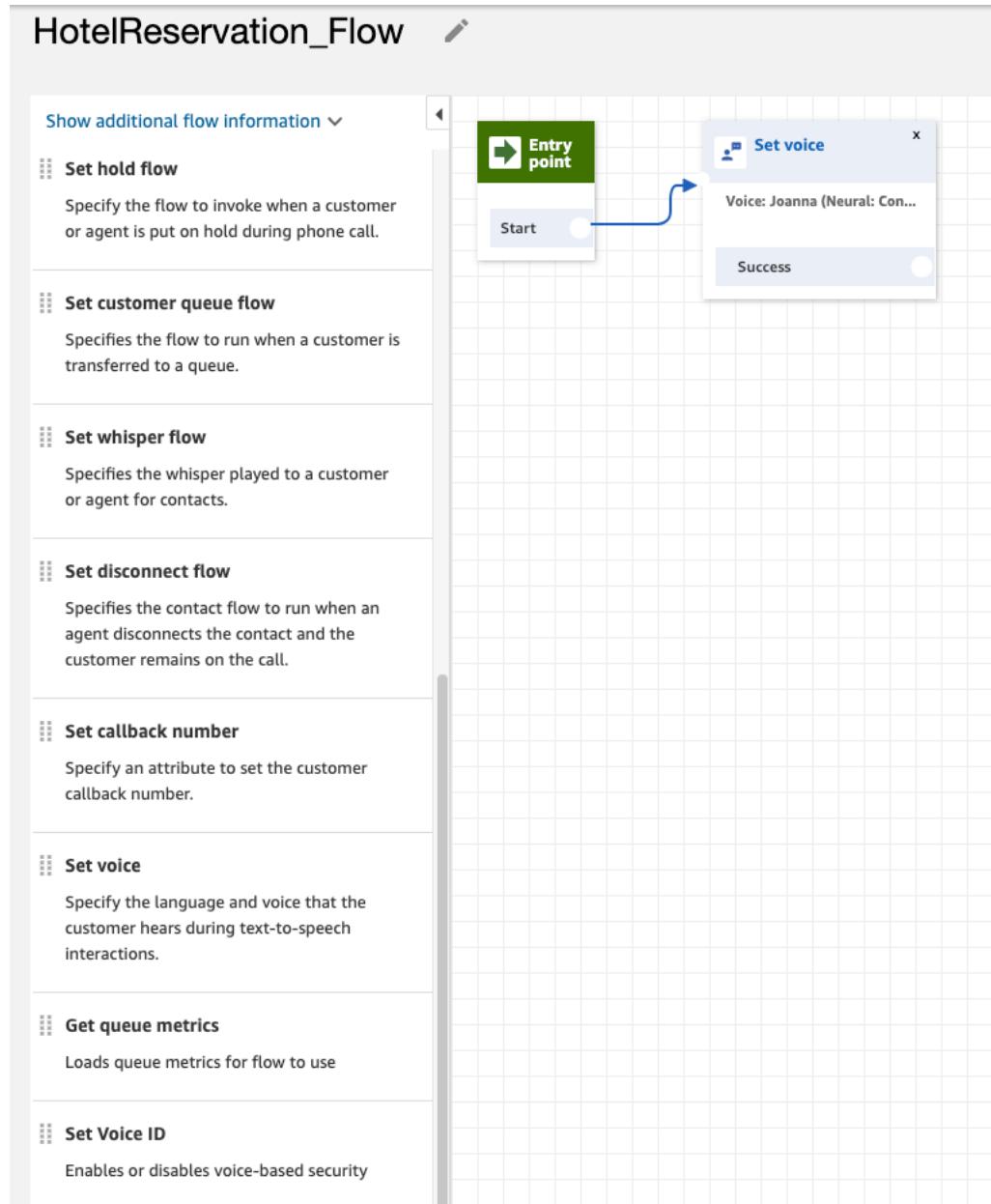


The screenshot shows the Amazon Connect Dashboard. On the left, there's a sidebar with icons for Channels, Routing, Contact flows, Prompts, Queues, Hours of operation, and Quick connects. The 'Routing' icon is highlighted with a teal bar. Under 'Routing', the 'Contact flows' option is also highlighted with a teal bar. The main content area has a header 'What's new?' with a message about the new 'Channels' navigation section. Below this, there are five numbered steps: 1. Set hours of operations, 2. Create queues, 3. Create prompts, 4. Create contact flows, and 5. Start a test chat. Step 1 is currently active. Step 5 includes a 'Begin' button and links to 'Test chat' and 'Customize chat widget'.

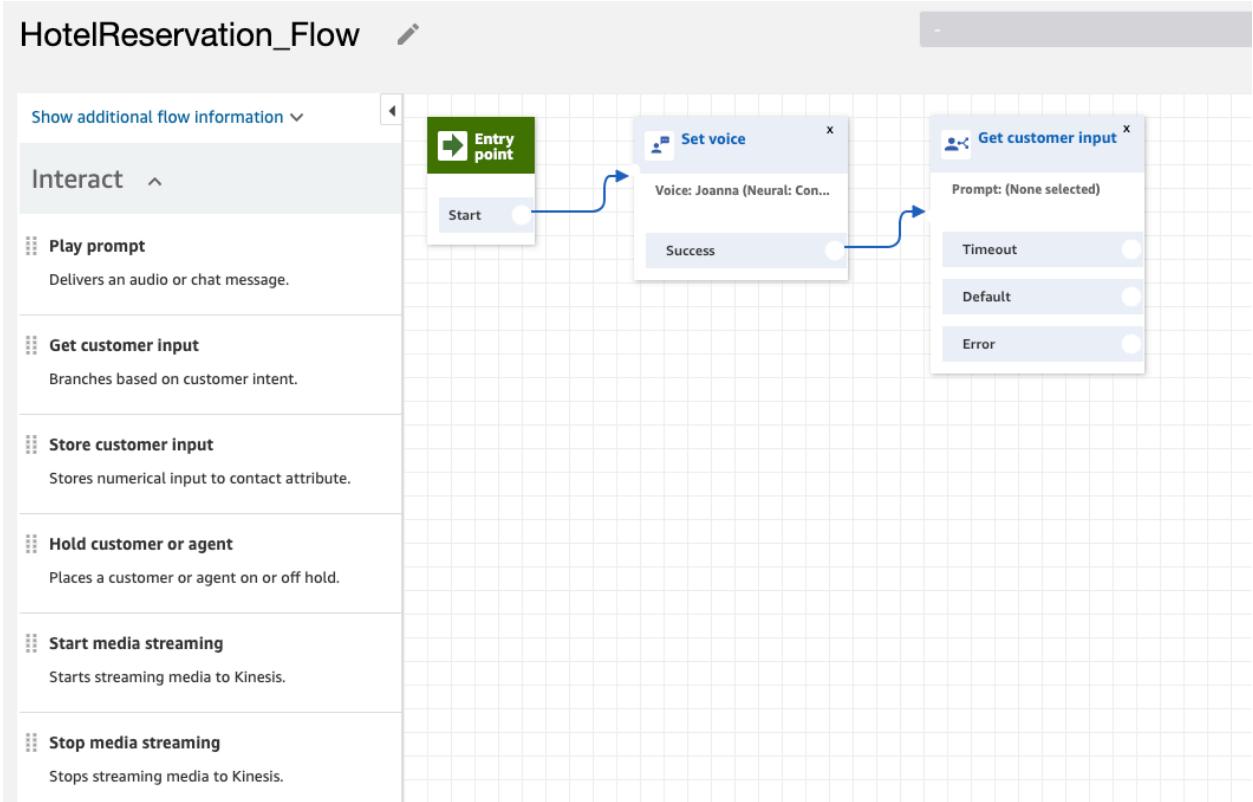
Once you login, click the **Contact Flows** option on the left pane under **Routing**. Then Choose on **Create contact flow** on the top right.



We have to give a name to this contact flow, now enter the name HotelReservation_Flow on the top left side.



In the **Set** section, bring a **Set voice** block onto the design pane and connect it to the Start output from the Entry point block.



Next in the **Interact** section, bring a **Get customer input** block onto the design pane and click the block.

The screenshot shows a contact flow in the AWS Step Functions console. A blue arrow points from the 'Get customer input' state to the configuration pane on the right.

Get customer input

Delivers an audio or chat message to solicit customer input.

Based on response, the contact flow branches. [Learn more](#)

Select from the prompt library (audio)
 Specify an audio file from an S3 bucket
 Text-to-speech or chat text
 Set manually

Hi, thank you so much for calling AWS hotel. How may I help you today?

Use attribute

Interpret as

Text

DTMF Amazon Lex

Plays an audio prompt and branches based on DTMF or Amazon Lex intents. The audio prompt is interruptible when using DTMF.

Lex bot

Select a Lex bot

Name

HotelBot (US East: N. Virginia)

Alias

TestBotAlias

Enter an ARN

Session attributes

Add an attribute

Intents

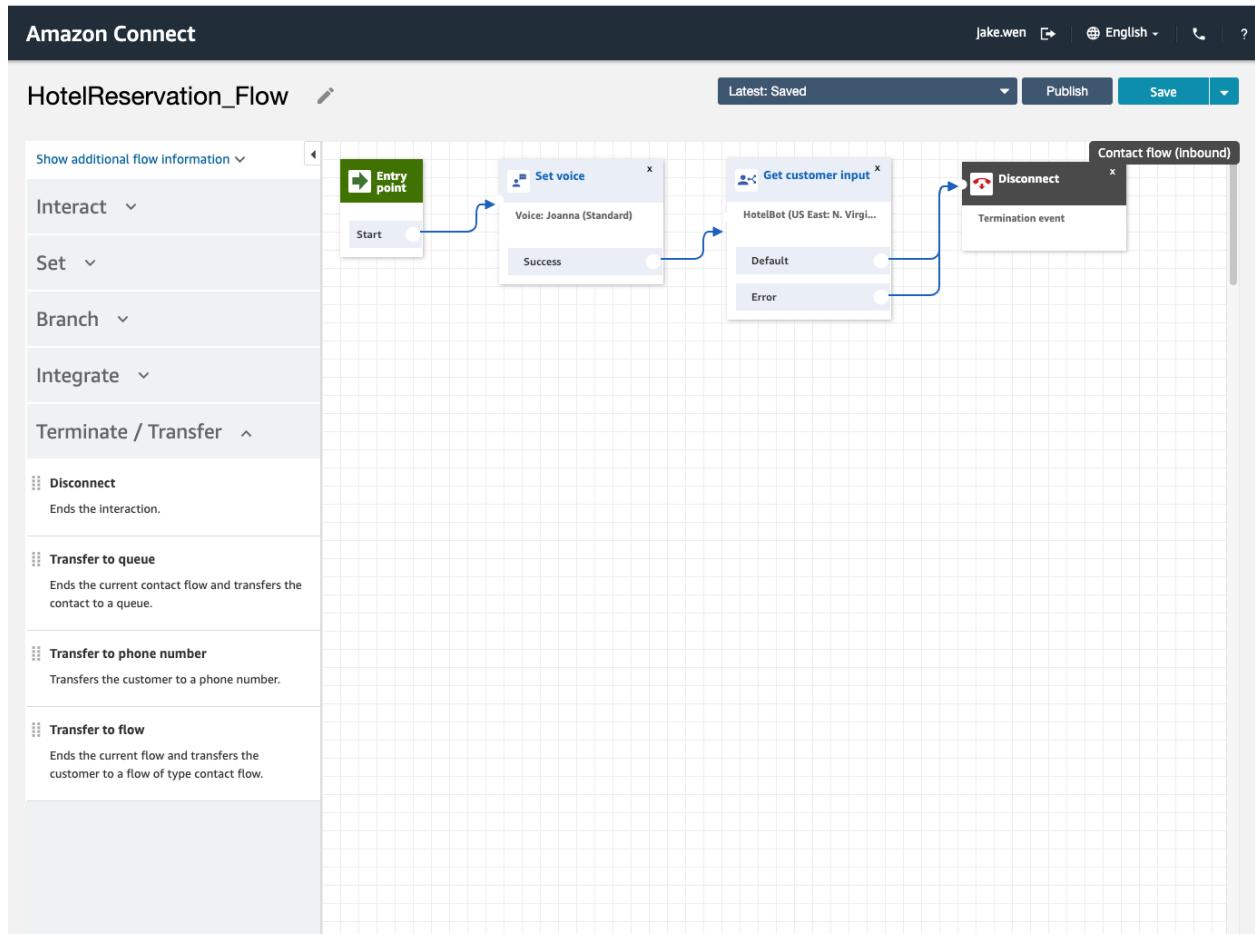
Add an intent

Cancel Save

Choose the Text-to-speech or chat text option, and enter the following greeting message:

Hi, thank you so much for calling AWS hotel. How may I help you today?

On the next section, choose Amazon Lex and select the HotelBot and the corresponding Alias. Then click **Save**.



On the Terminate/ Transfer option, pick **Disconnect** and save. Now you have built your very first contact flow from scratch. You can click Publish now to make the contact flow live.

i What's new?

We have introduced a new navigation section called Channels. Phone numbers and the Chat widget can now be found there, along with a new feature called Task templates.

X

Dashboard[Hide the guide](#)[Configuration guide](#)

Now that you have Amazon Connect setup, it's easy to manage your contact center reliably at any scale. Following these steps will guide you through the basics of configuring Amazon Connect for your business.

**1. Explore your channels of communication**

Claim a phone number in order to receive and make calls.

[Learn more](#)[Begin](#)

Engage more visitors in interactions with chat.

[Learn more](#)[Test chat](#)[Customize chat widget](#)

Now head back to the Amazon Connect home page and click **Begin** to claim a phone number.

Claim phone number**Claim your phone number**

Once you have claimed your phone number (from the list below), you can use the Contact Control Panel (CCP) to take calls.

Note: This is the phone number customers will call to reach your business. You can claim additional phone numbers from the Amazon Connect console later.

Select one from the list:

Country	Type	Phone number
+1	DID (Direct Inward Dialing)	+1 412-436-3624

[Skip for now](#)[Next](#)

For claim phone number, you can choose the country by preference. Then click **Next**.

Edit Phone number

+1 412-436-3624

Optional information

Description

First Phone Number

232 of 250 characters remaining.

Contact flow / IVR

HotelReservation_Flow

x ▾

Save

Cancel

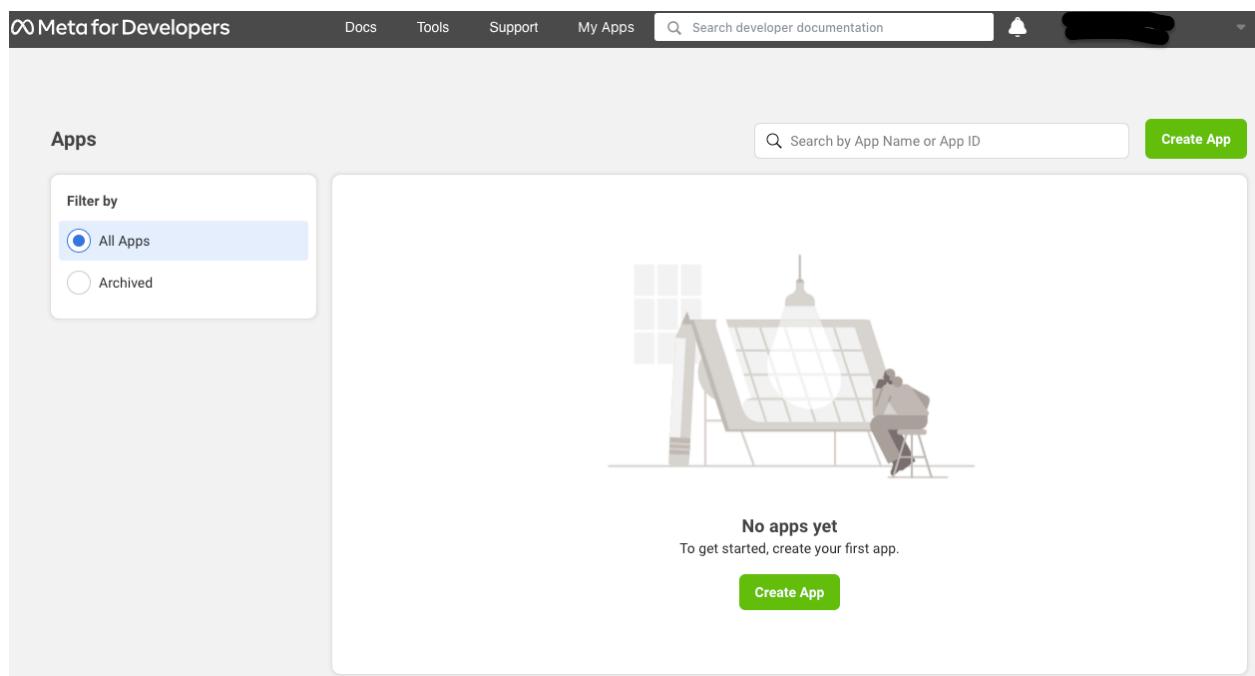
Once you successfully claimed a phone number, click the phone number into the **Edit Phone Number** page, attach the **HotelReservation_Flow** to this phone number and click **Save**.

Awesome! You made it! Now you can call the phone number you just claimed and try it out!

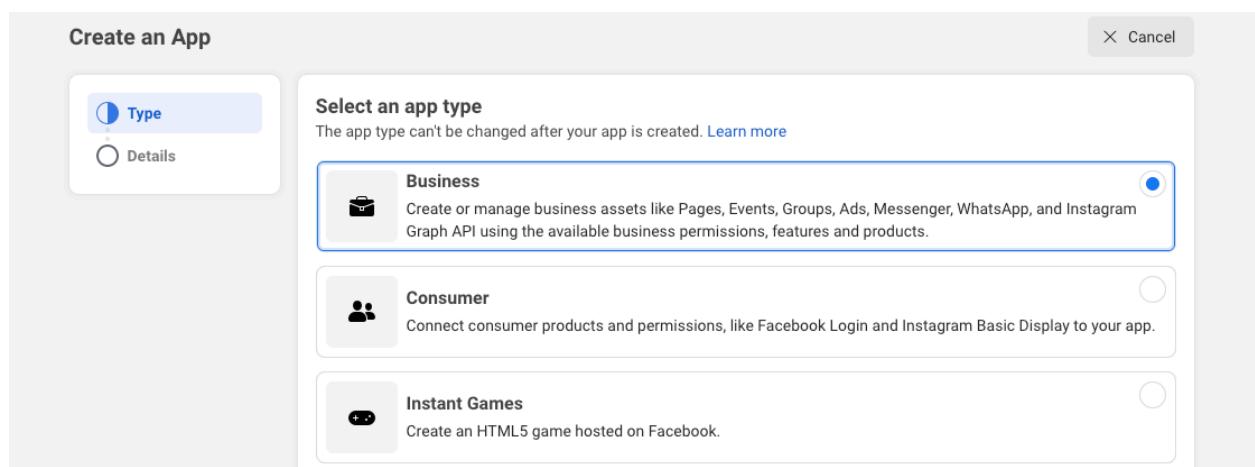
Optional: Integrate with Facebook Messenger

If you have a Meta developer (Facebook developer) account and want to test it out the Lex integration with Facebook Messenger this is how.

Visit this link: <https://developers.facebook.com/> and click **Get Started** on the right upper corner. Fill in the required information and click complete registration.



Once you completed the registration, click the **Create App** button.



Then choose the app type, for this demo, we will just use Business.

Create an App

Type
 Details

Provide basic information

Display name
This is the app name associated with your app ID. You can change this later.

16/30

App contact email
This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

Business Account · Optional
To access certain permissions or features, apps need to be connected to a Business Account.

By proceeding, you agree to the [Facebook Platform Terms](#) and [Developer Policies](#).

Previous
Create app

On the next page, put AWS London Hotel as the Display name and your personal email, then click Create app.

The screenshot shows the Facebook App Dashboard for the app "AWS London Hotel". The top navigation bar includes the app name, App ID (656567475660325), App type: Business, and a Help link. The left sidebar contains links for Dashboard, Settings, Roles, Alerts, App Review, Products (with an "Add Product" button), Activity Log, and an Activity Log icon. The main content area displays nine API integration boxes arranged in a grid:

- Fundraisers**: Create and manage fundraisers for charities. Buttons: Read Docs, Set up.
- Instagram Graph API**: Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts. Buttons: Read Docs, Set up.
- Jobs**: Post jobs to the Facebook platform and receive applications from users. Buttons: Read Docs, Set up.
- Marketing API**: Integrate Facebook Marketing API with your app. Buttons: Read Docs, Set up.
- Messenger**: Customize the way you interact with people on Messenger. Buttons: Read Docs, Set up.
- Web Payments**: Accept in-app payments through Facebook's secure payment system. Buttons: Read Docs, Set up.
- ThreatExchange**: Share and learn about potential threats to help everyone stay more secure. Buttons: Read Docs, Set up.
- Webhooks**: Subscribe to changes and receive updates in real time without calling the API. Buttons: Read Docs, Set up.
- WhatsApp**: Integrate with WhatsApp. Buttons: Read Docs, Set up.

Once the app is created, you will land on this page. Click Set Up under the Messenger box.

AWS London Hotel App ID: 656567475660325 App type: Business

Create ads to help more people discover your experience in Messenger.

Get started

Check out some resources that may help you in development: [Quick Start](#) (build a Messenger app in 10 minutes) and [Complete Documentation](#). Also join our [Messenger Platform Developers Community](#) on Facebook to get latest news and learn more!

Access Tokens

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Create new Page

No page permissions granted

You'll need to connect pages and grant them the required permissions in order for tokens to be generated.

Add or remove Pages

Webhooks

Click Create new Page.

Pages > Create a Page

Create a Page

Your Page is where people go to learn more about you. Make sure yours has all the information they may need.

Page name (required)
AWS London Hotel ✓

Use the name of your business, brand or organization, or a name that helps explain your Page. [Learn More](#)

Category (required)
Hotel ✓

Enter a category that best describes you.

Bio (optional)
Welcome to the best hotel in London

Tell people a little about what you do.

Desktop Preview

Intro

- 0 Followers
- Page · Hotel

Posts

Filters

Customize your page with your personal preference.

App ID: 656567475660325

App type: Business

Log in With Facebook

facebook.com/dialog/oauth?encrypted_query_string=AeDsII4TYZVY8s5pTy...

Jk Wen ▾

Get started

Check out some Documentation.

Access Tokens

Generate a Page

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

What Pages do you want to use with AWS London Hotel?

In the next step, you will determine what AWS London Hotel can do with the Pages you selected.

All Pages (2)

Select All

	AWS London Hotel	<input type="checkbox"/>
		100558606163083

Create new Page

Head back and choose the **Add or remove Pages**, and pick the **AWS London Hotel**.

Access Tokens

Create new Page

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Pages ↑	Tokens
AWS London Hotel 100558606163083	Token generated

Generate token

Add or remove Pages

Click **Generate token**.

Meta for Developers

Docs Tools Support My Apps

Search developer documentation

Jk Wen

AWS London Hotel App ID: 656567475660325 App type: Business

Help

Dashboard

Settings

Basic

Advanced

Roles

App ID: 656567475660325

App secret: *****

Show

Display name: AWS London Hotel

Namespace:

App domains:

Contact email: [\(i\)](#)

Copy the **App secret key** on the Basic section under Settings.

The screenshot shows the Amazon Lex interface. On the left, there's a sidebar with the following navigation:

- Bots
- HotelBot
 - Bot versions
 - Draft version
 - All languages
 - ▼ English (US)
 - Intents
 - Slot types
 - ▼ Deployment
 - Aliases
 - Channel integrations
 - ▼ Analytics
 - CloudWatch metrics
 - Utterances statistics

The main content area shows the "Channel integrations (0)" section. It includes a search bar labeled "Search channels", a table header with columns "Channel name", "Platform", and "Alias", and a message "No channels found". There is a prominent "Add channel" button at the bottom right.

Navigate to **Channel integrations** section, and click **Add Channel**.

Amazon Lex

Bots

HotelBot

Bot versions

Draft version

All languages

▼ English (US)

Intents

Slot types

▼ Deployment

Aliases

Channel integrations

▼ Analytics

CloudWatch metrics

Utterances statistics

► Related resources

Return to the V1 console

This is the key that will be used to protect your sensitive channel configuration information.

aws/lex

Integration configuration

Name

AWSHotel_FB

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

Description - optional

Customer Support

Maximum 200 characters.

Alias

TestBotAlias

Language

English (US)

Additional configuration

Alias

TestBotAlias

Page access token

EAAJVJQcoZAIUBAMJWgdE50Jc[REDACTED]

App secret key

3d3f96bb341c[REDACTED]

Cancel

Create

Paste the required information on the channel integration page and click create.

AWSHotel_FB

Delete

General configuration

Name	IAM role	Language
AWSHotel_FB	AWSServiceRoleForLexV2Chan...	English (US)
Platform	KMS key	
Facebook	alias/aws/lex	
Alias	Alias token	
TestBotAlias	TestBotAlias	
Description	Generated	
-	22 minutes ago	

 Use this generated callback URL to connect your channel to the alias listed.

[Learn more](#)

Callback URL

Endpoint

 Copied

<https://ls.lex.us-east-1.amazonaws.com/v2/facebook/webhook/af4a830d-87e5-4c41-1433f>

 Copy

Now copy the endpoint URL.

AWS London Hotel ▾

App ID: 656567475660325

App type: Business

Dashboard

Settings

Roles

Alerts

App Review

Products

Add Product

Facebook Login

Messenger**Settings**

Instagram settings

Activity Log

Activity Log

Check out some resources that may help you in development: [Quick Start](#) (build a Messenger app in 10 min) [Documentation](#). Also join our [Messenger Platform Developers Community](#) on Facebook to get latest news and updates.

Access Tokens

Generate a Page access token to start using the platform APIs. You will be able to generate an access token if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who are Page admins.

Pages ↑	Tokens
AWS London Hotel 100558606163083	—

Add or remove Pages ⓘ**Webhooks**

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Add Callback URL

Will your Page be affected by updates due to new privacy rules in Europe?

Click **Add Callback URL** and paste the endpoint URL there.

The screenshot shows the Meta for Developers dashboard with the following details:

- Top Bar:** Docs, Tools, Support, My Apps, Search developer documentation, Notifications, and User Profile (Jk Wen).
- Left Sidebar:** AWS London Hotel selected in the dropdown, App ID: 656567475660325, App type: Business.
- Central Content:** A modal window titled "Edit page subscriptions" for the "AWS London Hotel" app (ID: 100558606163083).
 - Subscription Fields:** A grid of checkboxes for various messaging events:

<input type="checkbox"/> messages	<input type="checkbox"/> messaging_postbacks	<input type="checkbox"/> messaging_optins
<input type="checkbox"/> messaging_optouts	<input type="checkbox"/> message_deliveries	<input type="checkbox"/> message_reads
<input type="checkbox"/> messaging_payments	<input type="checkbox"/> messaging_pre_checkouts	<input type="checkbox"/> messaging_checkout_updates
<input type="checkbox"/> messaging_account_linking	<input type="checkbox"/> messaging_referrals	<input type="checkbox"/> message_echoes
<input type="checkbox"/> messaging_game_plays	<input type="checkbox"/> standby	<input type="checkbox"/> messaging_handovers
<input type="checkbox"/> messaging_policy_enforcement	<input type="checkbox"/> message_reactions	<input type="checkbox"/> inbox_labels
<input type="checkbox"/> messaging_feedback	<input type="checkbox"/> messaging_customer_information	<input type="checkbox"/> whatsapp_messages
 - Buttons:** "Add subscriptions" (blue), "Cancel", and "Save".
 - Text Input:** "Enter Page IDs separated by ';' comma, ';' semicolon, or ' ' space."
- Bottom Navigation:** Settings, Instagram settings, Activity Log, Chat Plugin.

Click **Add subscription** and check the **messages** box and **Save**. Now navigate to your newly created Facebook page.



AWS London Hotel

0 likes · 0 followers

Posts

About

Mentions

Reviews

Followers

Switch into AWS London Hotel's Page to start managing posts.



Go to Ad Center to promote your Page

You'll have tools to create and manage ads for your page.

Promote

Intro



AWS London Hotel



...

Now

Hi! Thank you for contact AWS hotel!
How may I help you today?

book a room

What kind of room you are looking for?

King

What date you will be arrive?



Click the Messenger feature on your Facebook page and test it out!