

# User Guide for Audio Collection Framework

## About

*Author: Zeyuan Xu (@xuzeyuan)*

### What is Audio Collection Framework?

A framework that collects Human-Human and Human-Bot conversations in spoken/audio form, together with IVR data, on the platform of AWS Connect.

### Where can I find the code package?

The package is available at:

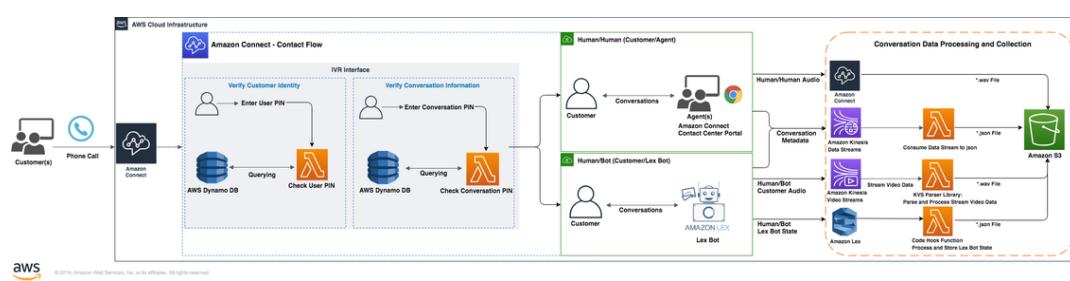
<https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/trees/mainline>

### What services does the framework use?

The framework uses multiple AWS services, including:

[AWS Connect](#), [AWS Lex](#), [AWS Lambda](#), [AWS Dynamo DB](#), [AWS S3](#), [AWS IAM](#), [AWS Kinesis Video/Data Streams](#), and [AWS Transcribe](#).

### How is the framework working?



## Document Overview

### 1. Deployment Guidance (Important: Please follow the exact sequence to deploy!)

- Deploy Audio Collection Framework on AWS
  - Download [framework code package](#) to local machine
  - AWS Account
  - AWS S3
  - AWS Lambda
  - AWS Lex
  - AWS Connect
    - AWS Kinesis Video/Data Streams
    - AWS Dynamo DB
    - Framework Configuration File
- Deploy Framework Web Interface
  - Local Machines
  - AWS EC2 (TODO)
- Deploy Framework Command Line Interface

### 2. Operational Activity Guidance

- Add a new Human/Human conversation domain
- Add a Lex bot

## Deployment Guidance

### Deploy Audio Collection Framework on AWS

#### Deploy Audio Collection Framework on AWS

The audio collection framework heavily relies on AWS services. Please follow the steps one by one for deployment.

[DOWNLOAD FRAMEWORK CODE PACKAGE TO LOCAL MACHINE](#)

```
cd <directory>
git clone ssh://git.amazon.com/pkg/AWSDeepSenseSpokenDataCollectionFramework
```

If you want to revise the code and publish a commit:

1. Create a brazil workspace and download the code
  - a. brazil workspace create --name AWSDeepSenseSpokenDataCollectionFramework --root <directory>/<workspace\_name> --versionset **AWSDeepSenseSpokenDataCollectionFramework/xuzeyuan**

```
cd <directory>/<workspace_name>
brazil workspace use --package AWSDeepSenseSpokenDataCollectionFramework --branch mainline
```
2. Build the package and run the test
  - a. cd <directory>/<workspace\_name>/src/AWSDeepSenseSpokenDataCollectionFramework
 

```
brazil-build
brazil-build test
```

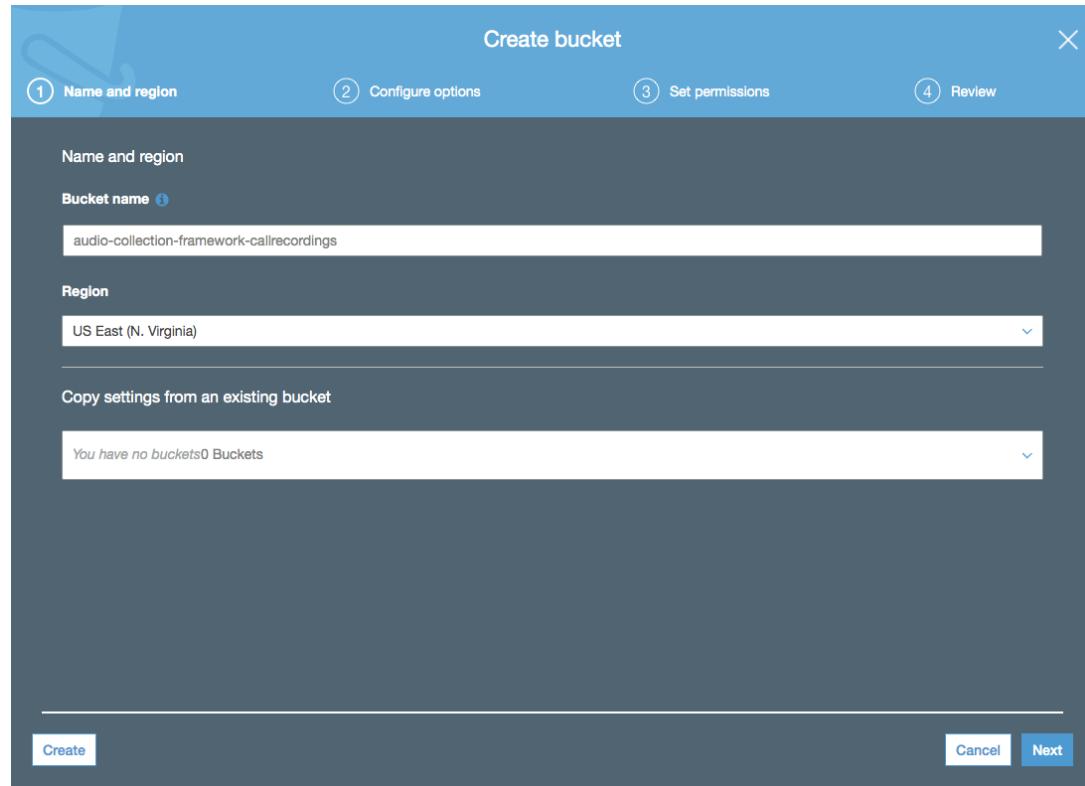
#### CREATE NEW AWS ACCOUNT

There are two ways of creating a new AWS account.

1. Please follow the instructions of the hyper-links given:
  - a. [Amazon Public AWS account](#)
  - b. [Amazon Internal Isengard AWS Account](#) (Please confirm that you have 'Admin Role' for the created account)
2. Obtain your AWS Secret Access Key and Access Key ID by following the [instructions](#)
3. Update the framework configuration file in local machines
  - a. Find the configuration file in local ([file in code.amazon.com](#))
  - b. Change it to the following format:
    - i. AWS\_ACCESS\_KEY\_ID <AWS Secret Access Key Id>
    - AWS\_ACCESS\_KEY <AWS Secret Access Key>
    - AWS\_REGION\_NAME <AWS Region Alias>

#### DEPLOY AWS S3

1. Go to [S3 Management Console](#)
2. Click 'Create bucket' → Bucket name: **audio-collection-framework-callrecordings**, Region: **US East (N. Virginia)** (**Note: S3 Bucket Name is globally unique**. You should use another bucket name different from the one below and choose a region that you prefer)



3. Click 'Next' → Use all default settings → Click 'Create bucket' → Done!

S3 buckets

Bucket name: audio-collection-framework-callrecordings

Access: All access types

Region: US East (N. Virginia)

Date created: Jul 30, 2019 10:13:11 AM GMT-0700

a. 4. Add one new line to the framework configuration file in local machine (Check your region alias [here](#))

a. CALL\_RECORDINGS\_BUCKET\_NAME <AWS S3 Bucket Name>

## DEPLOY AWS LAMBDA

1. Before deployment, you need to create a new IAM role to identify the function's access permission:

a. Go to [IAM Management Console](#)

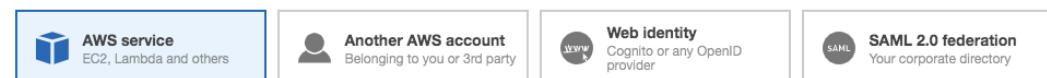
b. Choose the service that will use this role → Choose '**Lambda**' → Click '**Next: Permissions**'

i.

### Create role

1 2 3 4

#### Select type of trusted entity



Allows AWS services to perform actions on your behalf. [Learn more](#)

#### Choose the service that will use this role

##### EC2

Allows EC2 instances to call AWS services on your behalf.

##### Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway	Comprehend	ElastiCache	Lex	SMS
AWS Backup	Config	Elastic Beanstalk	License Manager	SNS
AWS Support	Connect	Elastic Container Service	Machine Learning	SWF
Amplify	DMS	Elastic Transcoder	Macie	SageMaker
AppSync	Data Lifecycle Manager	Elastic Load Balancing	MediaConvert	Security Hub
Application Auto Scaling	Data Pipeline	Forecast	Migration Hub	Service Catalog
Application Discovery Service	DataSync	Glue	OpsWorks	Step Functions
Batch	DeepLens	Greengrass	Personalize	Storage Gateway
CloudFormation	Directory Service	GuardDuty	RAM	Textract
CloudHSM	DynamoDB	Inspector	RDS	Transfer
CloudTrail	EC2	IoT	Redshift	Trusted Advisor
CloudWatch Application Insights	EC2 - Fleet	IoT Things Graph	Rekognition	VPC
	EC2 Auto Scaling	KMS	RoboMaker	WorkLink

\* Required

Cancel

Next: Permissions

c. Search and check '**AdministratorAccess**' (Use Admin Access for simplicity) → Click '**Next: Tags**'

i.

## Create role

1    2    3    4

### ▼ Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#)



Filter policies ▾		Q, AdministratorAccess	Showing 1 result	
	Policy name ▾	Used as	Description	
<input checked="" type="checkbox"/>	▶  AdministratorAccess	None	Provides full access to AWS services an...	

### ► Set permissions boundary

\* Required

[Cancel](#)

[Previous](#)

[Next: Tags](#)

d. Add tags if you want to → Click 'Next: Reviews'

e. Role name: **audio\_collection\_framework\_IAM\_role** (can use other names you prefer) → Click 'Create role'

i.

## Create role

1
2
3
4

### Review

Provide the required information below and review this role before you create it.

<b>Role name*</b> <input type="text" value="audio_collection_framework_IAM_role"/> <small>Use alphanumeric and '+,-,@-_ ' characters. Maximum 64 characters.</small>	
<b>Role description</b> <small>IAM role for lambda functions of audio collection framework.</small> <small>Maximum 1000 characters. Use alphanumeric and '+,-,@-_ ' characters.</small>	
<b>Trusted entities</b> AWS service: lambda.amazonaws.com	
<b>Policies</b> <a href="#">AdministratorAccess</a>	
<b>Permissions boundary</b> Permissions boundary is not set	
<i>No tags were added.</i>	

\* Required

[Cancel](#)

[Previous](#)

[Create role](#)

2. Go to [Lambda Console](#)

3. Get Started → Create a function

4. There are 5 lambda functions in total to create. Let's first deploy lambda function **checkPINcode**:

a. Choose '**Author from scratch**'

b. Function name: **checkPINcode**

c. Runtime: **Python 2.7**

d. Execution role: **Use an existing role**

e. Existing role: **audio\_collection\_framework\_IAM\_role** (based on the name of your previous created IAM role)

f. Click '**Create function**'

i.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is selected. The 'Function name' field contains 'checkPINcode'. The 'Runtime' dropdown is set to 'Python 2.7'. In the 'Permissions' section, it says 'Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.' A dropdown menu for 'Execution role' shows 'Use an existing role' selected, with 'audio\_collection\_framework\_IAM\_role' listed. At the bottom right, there are 'Cancel' and 'Create Function' buttons.

g. **Function Code** → replace the template code with the code in code package

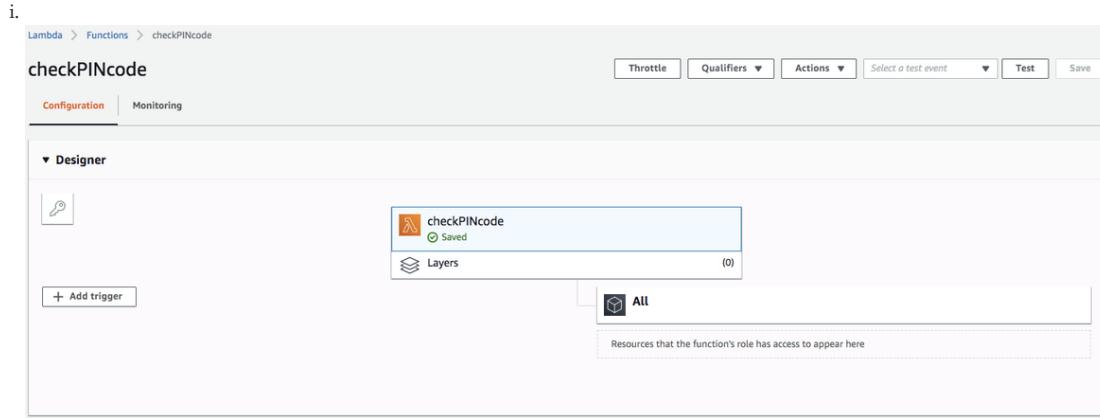
([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/aws\\_lambda\\_functions/checkPINcode.py](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/aws_lambda_functions/checkPINcode.py))

h. **Environment variables:**

(Check your region alias [here](#))

AWS\_REGION\_NAME: <AWS region alias>

i. Click 'Save' in the up right



ii.

```

# check_pin_code_lambda.py: This module is not directly run by the framework.
# It is deployed on AWS and will be called during the conversation.
# This module is to check if a PIN code, which entered by a customer,
# is valid, by querying AWS Dynamo DB.

import os
import json
import boto3
from boto3.dynamodb.conditions import Key

AWS_REGION_NAME = os.environ['AWS_REGION_NAME']
dynamodb = boto3.resource('dynamodb', region_name=AWS_REGION_NAME)
DYNAMODB_COLLECTION_REQUEST_TABLE_NAME = 'collectionSession'
DYNAMODB_COLLECTION_REQUEST_SECONDARY_INDEX = 'conversationPIN-index'
HUMANZHUMAN_MODE = 'human'
HUMANZBOT_MODE = 'bot'

def check_user_pin(user_pin):
    """
    Check if the user-input user PIN is valid
    :type user_pin: String
    :return: if the PIN is valid
    """
    table = dynamodb.Table('userAccount')
    session = table.get_item(Key={'PIN': str(user_pin)})
    response = {'response': 'False'}
    if 'Item' in session and session['Item']['type'] == 'customer':
        response = {'response': 'True'}
    return response

```

5. Repeat the steps of deploying **checkPINcode** to deploy:

a. **triggerLambdaFunction** (Ignore) Step i, No need to set up Environment Variables

i. Code link:

([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/aws\\_lambda\\_functions/triggerLambdaFunction.py](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/aws_lambda_functions/triggerLambdaFunction.py))

b. **consumeCtrStream**

i. Code link:

([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/aws\\_lambda\\_functions/consumeCtrStream.py](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/aws_lambda_functions/consumeCtrStream.py))

ii. Environment variables: (**Note:** should use your own AWS S3 bucket name)

iii.

CALL_RECORDINGS_BUCKET_NAME	audio-collection-framework-callrecordings	<a href="#">Remove</a>
Key	Value	<a href="#">Remove</a>

c. **storeLexConversation**

i. Code link:

([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/aws\\_lambda\\_functions/storeLexConversation.py](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/aws_lambda_functions/storeLexConversation.py))

ii. Environment variables: (**Note:** should use your own AWS S3 bucket name)

iii.

#### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

<input type="text" value="CALL_RECORDINGS_BUCKET_NAME"/>	<input type="text" value="audio-collection-framework-callrecordings"/>	<button>Remove</button>
<input type="text" value="Key"/>	<input type="text" value="Value"/>	<button>Remove</button>

▶ Encryption configuration

6. Deploy lambda function **KVSTranscribeStreamingLambda**, click ‘Create function’ in the Lambda Console:

a. Choose ‘Author from scratch’

b. Function name: **KVSTranscribeStreamingLambda**

c. Runtime: **Java 8**

d. Execution role: **Use an existing role**

e. Existing role: **audio\_collection\_framework\_IAM\_role** (based on the name of your previous created IAM role)

f. Click ‘Create function’

g. Function Code → Handler

h. com.amazonaws.kvstranscribestreaming.KVSTranscribeStreamingLambda::handleRequest

i. Function Code → **Function package**

([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/aws\\_lambda\\_functions/KVSTranscribeStreamingLambda/amazon-connect-realtime-transcription-lambda-function.zip](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/blobs/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/aws_lambda_functions/KVSTranscribeStreamingLambda/amazon-connect-realtime-transcription-lambda-function.zip))

i.

The screenshot shows the 'Function code' section of the Lambda function configuration. It includes fields for 'Code entry type' (Upload a .zip or .jar file), 'Runtime' (Java 8), and 'Handler' (Info: com.amazonaws.kvstranscribestreaming.KVSTranscribe). A file named 'amazon-connect-realtime-transcription-lambda-function.zip' (17.3 MB) is shown as uploaded. A note at the bottom states: 'For files larger than 10 MB, consider uploading using Amazon S3.'

j. Environment variables: (**Note:** 1. Please replace **us-east-1** with your own AWS region alias 2. Please replace **connect-callrecordings** with your own AWS S3 bucket name.)

i.

#### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

<input type="text" value="APP_REGION"/>	<input type="text" value="us-east-1"/>	<button>Remove</button>
<input type="text" value="CONSOLE_LOG_TRANSCRIPT_FLAG"/>	<input type="text" value="FALSE"/>	<button>Remove</button>
<input type="text" value="INPUT_KEY_PREFIX"/>	<input type="text" value="audio-file-input/"/>	<button>Remove</button>
<input type="text" value="RECORDINGS_BUCKET_NAME"/>	<input type="text" value="connect-callrecordings"/>	<button>Remove</button>
<input type="text" value="RECORDINGS_KEY_PREFIX"/>	<input type="text" value="Value"/>	<button>Remove</button>
<input type="text" value="RECORDINGS_PUBLIC_READ_ACL"/>	<input type="text" value="FALSE"/>	<button>Remove</button>
<input type="text" value="SAVE_PARTIAL_TRANSCRIPTS"/>	<input type="text" value="FALSE"/>	<button>Remove</button>
<input type="text" value="START_SELECTOR_TYPE"/>	<input type="text" value="EARLIEST"/>	<button>Remove</button>
<input type="text" value="TABLE_CALLER_TRANSCRIPT"/>	<input type="text" value="contactTranscriptSegments"/>	<button>Remove</button>
<input type="text" value="TRANSCRIBE_REGION"/>	<input type="text" value="us-east-1"/>	<button>Remove</button>
<input type="text" value="Key"/>	<input type="text" value="Value"/>	<button>Remove</button>

k. Basic settings: **Memory** → **1024 MB**, **Timeout** → **1 min**

i.

**Basic settings**

**Description**

**Memory (MB)** [Info](#)  
Your function is allocated CPU proportional to the memory configured.

1024 MB

**Timeout** [Info](#)  
1 min 0 sec

1. Click 'Save' in the up right

7. Finally, the **Lambda Management Console** should look like this:

a.

Function name	Description	Runtime	Code size	Last modified
KVSTokenizedStreamingLambda		Java 8	17.3 MB	2 minutes ago
triggerLambdaFunction		Python 2.7	659 bytes	33 minutes ago
checkNoise		Python 2.7	1.3 kB	49 minutes ago
consumeCloudStream		Python 2.7	851 bytes	32 minutes ago
storeLexConversation		Python 2.7	1.0 kB	32 minutes ago

8. Done!

#### DEPLOY AWS LEX

1. Go to [Lex Management Console](#)
2. Click **Get Started → Create bot**
3. Use sample template to create **BookTrip** bot → Click **Create**

**IAM role** AWSLambdaBasicExecutionRole  
Automatically created on your behalf

**COPPA** Please indicate if your use of this bot is subject to the Children's Online Privacy Protection Act (COPPA). [Learn more](#)

Yes  No

a.

4. Setup code hook function **for each intent**

- a. Click **BookCar** under **Intents** in the left sidebar
- b. Click **Lambda Initialization and Validation**
- c. Check Initialization and validation code hook
- d. Select **storeLexConversation** as the Lambda function, **Latest** as the Version or alias

i.

**BookTrip Latest**

**Editor** **Settings** **Channels** **Monitoring**

**Intents**

- BookCar** BookHotel

**Slot types**

**Error Handling**

**BookCar Latest**

**Sample utterances**

e.g. I would like to book a flight.  
Make a car reservation  
Reserve a car  
Book a car

**Lambda initialization and validation**

Initialization and validation code hook

**Lambda function** **storeLexConversation**

**Version or alias** **Latest**

e. Scroll down and click **Save Intent**f. Click **BookHotel** under **Intents** in the left sidebar

g. Repeat step b - e

h. Click **Build** on the top right (**Note:** This bot can only be used in Amazon Connect scenario. Trying to talk to the bot on the console will always fail.)i. Once **Build** is done, click **Publish** on the top right, enter **BookTrip** in **Create an alias**

**Publish BookTrip**

Your bot is published! You can now connect to your mobile app or continue to chatbot deployment.

<b>Bot Name</b>	BookTrip
<b>Bot Version</b>	1
<b>Alias</b>	BookTrip

**What to do next?**  
Here are some resources to help you progress once your bot is published.

**How to connect to your mobile app**  
Learn how to connect to your bot to your mobile app.  
[Download connection info](#)

**Integrate with Mobile hub.**  
Please create a project and choose the Conversational Bots feature in Mobile Hub

**How to deploy your bot to other services**  
Learn how to deploy your bot to other services like Facebook Messenger, Slack, Twilio, and Kik.  
[Go to channels](#)

j.

**Close**

5. Use sample template to create **OrderFlowers** bot → Click **Create**

CREATE YOUR OWN TRY A SAMPLE

[Custom bot](#) [BookTrip](#) [OrderFlowers](#) [ScheduleAppointment](#)

**Bot name**  
OrderFlowers

**IAM role** AWSServiceRoleForLexBots  
Automatically created on your behalf

**COPPA** Please indicate if your use of this bot is subject to the Children's Online Privacy Protection Act (COPPA). [Learn more](#)

Yes  No

a.

6. Repeat step 4 to deploy the **OrderFlowers** bot

**Publish OrderFlowers**

Your bot is published! You can now connect to your mobile app or continue to chatbot deployment.

<b>Bot Name</b>	OrderFlowers
<b>Bot Version</b>	1
<b>Alias</b>	OrderFlowers

**What to do next?**  
Here are some resources to help you progress once your bot is published.

**How to connect to your mobile app**  
Learn how to connect to your bot to your mobile app.  
[Download connection info](#)

**Integrate with Mobile hub.**  
Please create a project and choose the Conversational Bots feature in Mobile Hub

**How to deploy your bot to other services**  
Learn how to deploy your bot to other services like Facebook Messenger, Slack, Twilio, and Kik.  
[Go to channels](#)

[Close](#)

a.

7. Done!

**DEPLOY AWS CONNECT**

1. Go to [Connect Management Console](#)
2. Click **Get started**
3. Check **Store users within Amazon Connect**
4. **Name** your connect instance (Shall be different from the image below)

**Amazon Connect resource configuration**

Get up and running in a few easy steps

**Step 1: Identity management**

Step 2: Administrator  
Step 3: Telephony options  
Step 4: Data storage  
Step 5: Review and create

**Identity management**

Amazon Connect can be configured to manage your users directly or to leverage an existing directory. This cannot be changed once your instance is created. [Learn more](#)

**Store users within Amazon Connect**  
Users will be created and managed by you within Amazon Connect. Note: you will not be able to share users with other applications.

**Link to an existing directory**  
Amazon Connect will leverage an existing directory. You create users within the directory and then add and configure them within Amazon Connect. Note: you can only associate a directory with a single Amazon Connect instance. [Learn more](#)

**SAML 2.0-based authentication**  
AWS supports identity federation with **SAML 2.0** (Security Assertion Markup Language 2.0), an open standard that many identity providers (IdPs) use. This feature enables federated 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](#)

[Access URL](#)  [Edit](#)

[Cancel](#) [Previous](#) [Next step](#)

a.

5. **Create Admin accounts** for your connect instance (Shall be different from the image below)

## Create an Administrator

Specify an administrator for this instance of Amazon Connect; this could be you or someone else. You will be able to manage permissions and add more users from within Amazon Connect.

Add a new admin

First Name	admin
Last Name	admin
Username	admin
Password	*****
Password (verify)	*****
Email Address	your-alias@amazon.com

Skip this

[Cancel](#) [Previous](#) [Next step](#)

a.

### 6. Enable both Incoming calls and Outbound calls

#### Telephony Options

Amazon Connect offers the ability to accept inbound calls, make outbound calls, or both. You will claim a telephone number later.  
Note: You will not be able to place or receive phone calls if you don't select the corresponding telephony options.

#### Incoming calls

I want to handle **incoming** calls with Amazon Connect

#### Outbound calls

I want to make **outbound** calls with Amazon Connect

Note: You can set which users can place outbound calls in user permissions.

[Cancel](#) [Previous](#) [Next step](#)

a.

### 7. Data Storage Settings

a. Click **Customize Settings**

i.

## Data storage

Call recordings and scheduled reports are stored in an Amazon 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.

To successfully create an Amazon Connect instance, you need to use an AWS account that has access to both Amazon S3 and Amazon CloudWatch.

**Important:** By choosing **Next step** you are granting Amazon Connect the following permissions:

- Read and write access to your S3 bucket to save and manage your data
- Encrypt/decrypt permissions to encrypt data
- Read and write access to CloudWatch Logs

Your data will be encrypted and stored here connect-c92a66b38ade/connect/audioCollectionFrameworkDemo

Your Contact flow logs will be stored here /aws/connect/audioCollectionFrameworkDemo

[Customize settings](#)

[Cancel](#) [Previous](#) [Next step](#)

b. Click **Edit** under **Call Recordings**

c. Click **Select an existing S3 bucket**

d. Select previously created S3 bucket

e. Set **Path prefix** to **human2human**

f. Uncheck **Enable encryption**

i.

### Call recordings

<input checked="" type="checkbox"/> Enable call recording	<a href="#">Cancel</a>	<a href="#">Save</a>
<input type="radio"/> Create a new S3 bucket for me (recommended)		
<input checked="" type="radio"/> Select an existing S3 bucket		

Name	<input type="text" value="call-callrecordings"/>	<a href="#">?</a>
Path prefix	<input type="text" value="human2human"/>	<a href="#">?</a>

Location: call-callrecordings/connect/audioCollectionFrameworkDemo/human2human

### Encryption options for call recordings

Enable encryption

g. Click **Save**

h. Click **Edit** under **Exported reports**

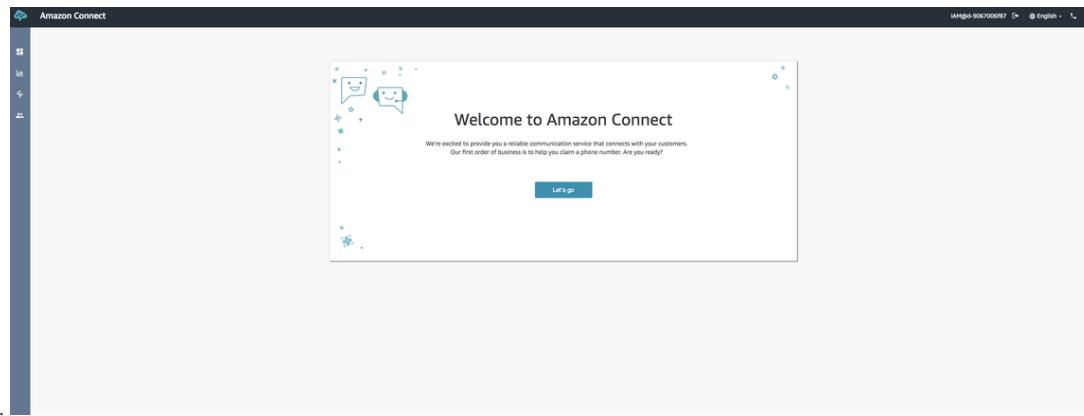
i. Repeat steps c - g

j. Click **Save**

8. Click **Next step**

9. Click **Create instance**

10. Click **Get started**, it should route you to another web page → Click **Let's go**



a.

11. Claim a phone number → Set **Country** to **+1 United State**, Set **Type** as **Direct Dial**, Select a random **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	Direct Dial	+1 217-262-6662 +1972-954-9239 +1470-235-4959 +1929-436-4614 +1629-209-3182

Skip for now **Next**

a.

12. Call Amazon Connect phone number for testing

### Claim phone number

**Make a call**

Dial +1 217-262-6662 from another phone and choose the menu item to connect with an agent. You can then use the Amazon Connect Contact Control Panel located to the side to accept the call. You can call this number back to experience more of what Amazon Connect can do.

**Continue** **Skip for now**

Change status **Available**

Dial number Quick connects

a.

13. Add new lines to the framework configuration file in local machine

a. CONNECT\_PHONE\_NUMBER <claimed\_phone\_number\_with\_no\_space>  
CONNECT\_CCP\_URL https://<instance\_name>.awsapps.com/connect/ccp#/

14. Add sample **travel** queue

- Click **left sidebar** → **Routing** → **Queues**
- Click **Add new queue**
- Name:** travel, **Outbound caller ID name:** Audio Collection Framework, **Outbound caller ID number:** <select the only one>, **Hours of operation:** Basic Hours

i.

Name  
travel

Description  
Queue for having conversations about travel  
207 of 250 characters remaining.

Hours of operation  
Basic Hours

Outbound caller ID name  
Audio Collection Framework  
The name that will show up on the customer's phone

Outbound caller ID number  
+1 217-262-6662

Outbound whisper flow (optional)  
Search for contact flow

Maximum contacts in queue (optional)  
 Set a limit

d. Click **Add new queue**e. Get **Queue ARN**i. Click back into the **travel** queue againii. Parse the url address to get **Queue ARN** (Queue\_ARN=url.split('/')[−1])

1.

iii. Keep it in a safe place. We will need this later.

15. Add two other queues (**enterprise**, **finance**) by following the same step a-e in 14

16. Add queue into routing profiles

a. Click **left sidebar → Users → Routing profiles**b. Edit **Basic Routing Profile**c. Remove **basic queue**, add **travel**, **finance**, **enterprise** queue in the profile

Name  
Basic Routing Profile

Description  
A simple routing profile.  
223 of 250 characters remaining.

Routing profile queues  
If no queue is added, the agent will only be able to make outbound calls. At least one queue is needed for inbound calls, and must not be a duplicate.

Name	Priority	Delay (in seconds)
enterprise	1	0
finance	1	0
travel	1	0

d. Get **Routing Profile ID**i. Parse the current url address to get **Routing Profile ID** (Routing\_Profile\_ID=url.split('/')[−1])

1.

ii. Add new lines to the framework configuration file in local machine

1. CONNECT\_ROUTING\_ID &lt;routing\_profile\_ID&gt;

f. Click **Save**17. Get **Security Profile ID** for agenta. Click **left sidebar → Users → Security profiles**b. Edit **Agent**c. Parse the url address to get **Security Profile ID** (Security\_Profile\_ID=url.split('/')[−1])

i.

- d. Add new lines to the framework configuration file in local machine
- CONNECT\_SECURITY\_ID <security\_profile\_ID>
18. Configure the connect instance
- Go to [Connect Management Console](#)
  - Click the only **Instance Alias**
  - Click **Overview** on the left sidebar
  - Get the connect instance id by parsing the Instance ARN (Instance ID=Instance\_ARN.split('/')[−1])

1.

- ii. Add new lines to the framework configuration file in local machine
- CONNECT\_INSTANCE\_ID <instance\_ID>
- d. Click **Data Storage** on the left sidebar
- Edit **Live media streaming**
  - Check **Enable live media streaming**, set **Prefix** as **Kinesis**, check **Select KMS key by name**, select **aws/kinesisvideo** for **KMS master key**, change **Data retention period** to **1 Years**

1.

Amazon Connect creates a group of Amazon Kinesis video streams. You can view the streams in the Kinesis Video Streams console. Provide a prefix to use for the stream names so you can easily identify them. If you change the Prefix or the KMS key, the change applies to future recordings only. [Learn more](#)

**Prefix**  -connect-audiocollectionframeworkdemo-contact- [i](#)  
Max length: 128 characters. Valid characters include: letters, numbers, underscore, hyphen.

#### Encryption

All data put into a Kinesis video stream is encrypted at rest

Select KMS key by name  Enter key ARN/ID

**KMS master key**  [i](#)

**ARN / ID**

**Description** Default master key that protects my Kinesis Video Streams data when no other key is defined

**Account** 843374839531

#### Data retention period

Kinesis video streams can store stream data for hours, days, or not at all for immediate consumption only. Cost is based on bandwidth and total storage used. You can modify data retention at any time.

No data retention

- iii. Click **Save**
  - iv. Click **Save** (Overall Save for **Data Storage**)
  - v. Click **Data Streaming** on the left sidebar
  - vi. Check **Enable data streaming**
  - vii. Check **Kinesis Stream**
  - viii. Click **Create a new Kinesis Stream**, it should route you to a new web page
- e. In the new page, click **Create Kinesis stream**
- i. Name **connect\_ctr\_streaming** for **Kinesis stream name** (You can name it whatever you prefer)
  - ii. Number of shards: 2

1.

**Kinesis stream name\*** connect\_ctr\_streaming  
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

**Shards**  
A shard is a unit of throughput capacity. Each shard ingests up to 1MB/sec and 1000 records/sec, and emits up to 2MB/sec. To accommodate for higher or lower throughput, the number of shards can be modified after the Kinesis stream is created using the API. [Learn more](#)

**Producers** → **Kinesis stream** → **Consumers**

▶ Estimate the number of shards you'll need

**Number of shards\*** 2  
You can provision up to 500 more shards before hitting your account limit of 500.  
[Learn more or request a shard limit increase for this account](#)

**Total stream capacity** Values are calculated based on the number of shards entered above.

<b>Write</b> 2 MB per second	<b>Read</b> 4 MB per second
2000 Records per second	

\* Required      Cancel      **Create Kinesis stream**

- iii. Click **Create Kinesis stream**
- f. Go back to [Connect Management Console Data Streaming](#) page, set **Kinesis Stream** to the one we just created

i.

## Data streaming

You can export Contact Trace Records (CTRs) and agent events from Amazon Connect in order to perform analysis on your data. Get started by enabling data streaming and utilizing [Amazon Kinesis Stream](#) or [Amazon Kinesis Firehose](#) to export your data. [Learn more](#).

- Enable data streaming

By enabling this feature, you are granting us the permission to put records to your Kinesis Stream or Kinesis Firehose.

### Contact Trace Records

Use one of your existing Amazon Kinesis Stream or Amazon Kinesis Firehose from the list below, or create a new one.

<input type="radio"/> Kinesis Firehose <small>(i)</small>	<input checked="" type="radio"/> Kinesis Stream <small>(i)</small>
<input style="width: 200px; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="text" value="connect_ctr_streaming"/> <span style="font-size: 10px;">▼</span>	
<a href="#">Create a new Kinesis Stream</a> <small>(i)</small>	

### Agent Events

Use your existing Amazon Kinesis Stream from the list below, or create a new one.

<input type="radio"/> Kinesis Stream <small>(i)</small>	<input style="width: 200px; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="text" value="Select"/> <span style="font-size: 10px;">▼</span>	<a href="#">Create a new Kinesis Stream</a> <small>(i)</small>
---------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------

[Cancel](#) [Save](#)

ii. Click **Save**

g. Go to [Lambda Management Console](#)

i. Edit **consumeCtrStream**

ii. Designer → Add trigger

iii. Select **Kinesis**, select **connect\_ctr\_streaming** for **Kinesis stream**, leave other settings unchanged

1.

The screenshot shows the 'Trigger configuration' section of the 'Add trigger' wizard. It is specifically configured for a Kinesis stream. The 'Kinesis' category is selected in the sidebar. The 'Kinesis stream' dropdown contains the value 'connect\_ctr\_streaming'. The 'Consumer' dropdown is set to 'No consumer'. The 'Batch size' input field contains the value '100'. The 'Starting position' dropdown is set to 'Latest'. A note below states: 'In order to read from the Kinesis trigger, your execution role must have proper permissions.' A checked checkbox labeled 'Enable trigger' has the sub-instruction: 'Enable the trigger now, or create it in a disabled state for testing (recommended)'. At the bottom right are 'Cancel' and 'Add' buttons.

iv. Click **Add**

h. Go back to [Connect Management Console](#), click **Contact flows** on the left sidebar

- i. Add Lex Bot (BookTrip, OrderFlowers)
- ii. Add Lambda Functions (all 5 that we created previously)

iii.

## 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.

Note: By adding Lex bots, you are granting Amazon Connect permission to interact with them [Create a new Lex bot](#)

Region: US East: N. Virginia

Bot: Select

No Lex bots available to add. [Create new](#).

**Lex bots**

BookTrip	<a href="#">Remove</a>
OrderFlowers	<a href="#">Remove</a>

## AWS Lambda

Amazon Connect can interact with your own systems and take different paths in IVR dynamically. To achieve this, invoke AWS Lambda functions in contact flows to interact with your own systems or other services, then build personalized and dynamic experiences based on data returned.

Note: By adding Lambda functions, you are granting Amazon Connect permission to invoke them [Create a new Lambda function](#)

Function: Select

No Lambda functions available to add for your account. [Create new](#).

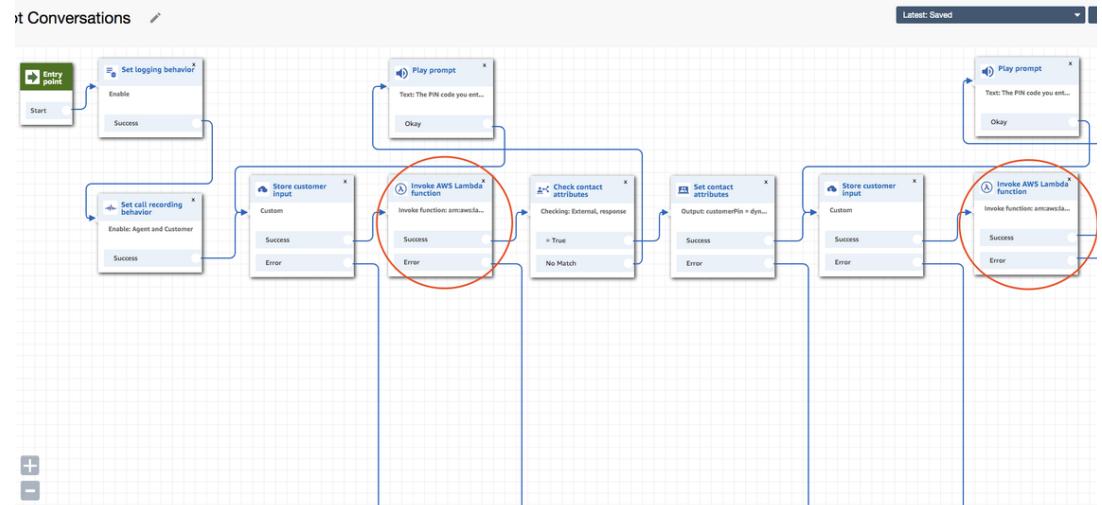
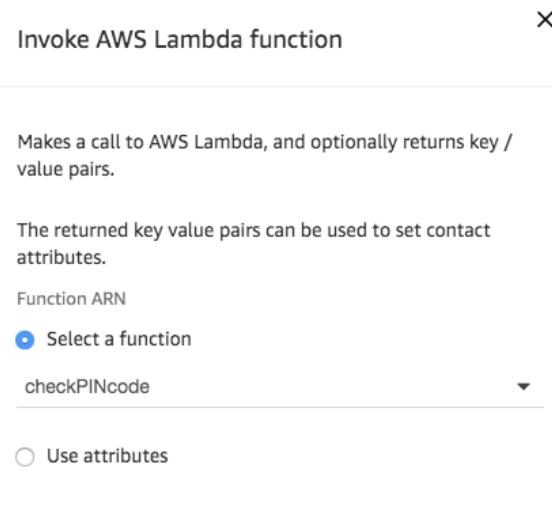
**Lambda Functions**

KVSTranscribeStreamingLambda	arn:aws:lambda:us-east-1:843374839531:function:KVSTranscribeStreamingLambda	<a href="#">Edit</a>	<a href="#">Remove</a>
triggerLambdaFunction	arn:aws:lambda:us-east-1:843374839531:function:triggerLambdaFunction	<a href="#">Edit</a>	<a href="#">Remove</a>
checkPINcode	arn:aws:lambda:us-east-1:843374839531:function:checkPINcode	<a href="#">Edit</a>	<a href="#">Remove</a>
consumeCtrStream	arn:aws:lambda:us-east-1:843374839531:function:consumeCtrStream	<a href="#">Edit</a>	<a href="#">Remove</a>
storeLexConversation	arn:aws:lambda:us-east-1:843374839531:function:storeLexConversation	<a href="#">Edit</a>	<a href="#">Remove</a>

### 19. Deploy contact flow

- Click left sidebar → Routing → Contact flows
- Click **Create contact flow**
- Click **Import flow (beta)** in the scroll down menu next to the save button
- Upload contact flow JSON file in the code package  
([https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/trees/mainline/-/src/aws\\_deep\\_sense\\_spoken\\_data\\_collection\\_framework/meta-data/Amazon Connect Contact Flow for Human-Human\\_Human-Bot Conversation.json](https://code.amazon.com/packages/AWSDeepSenseSpokenDataCollectionFramework/trees/mainline/-/src/aws_deep_sense_spoken_data_collection_framework/meta-data/Amazon Connect Contact Flow for Human-Human_Human-Bot Conversation.json))
- Set correct Lambda function
  - checkPINcode**
    - Select the **red-circled blocks**

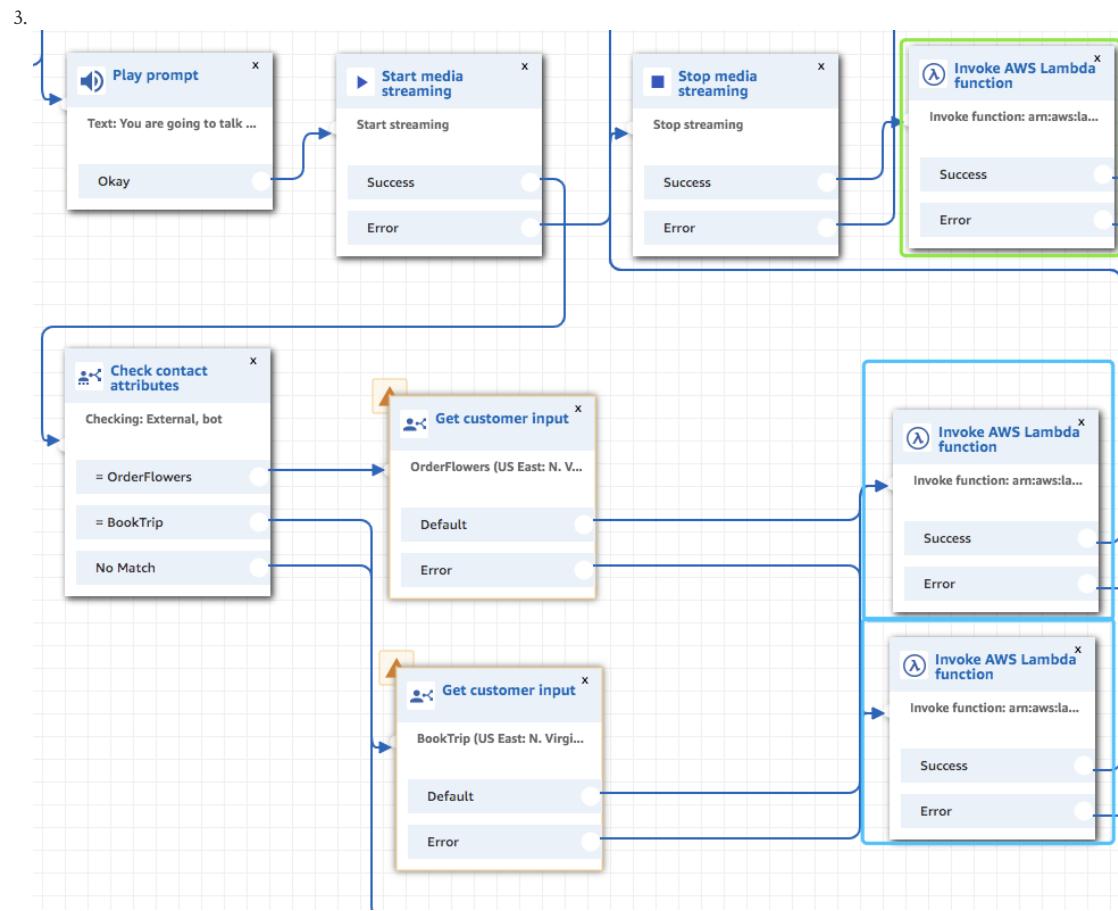
a.

2. set the lambda function to **checkPINcode**

a.

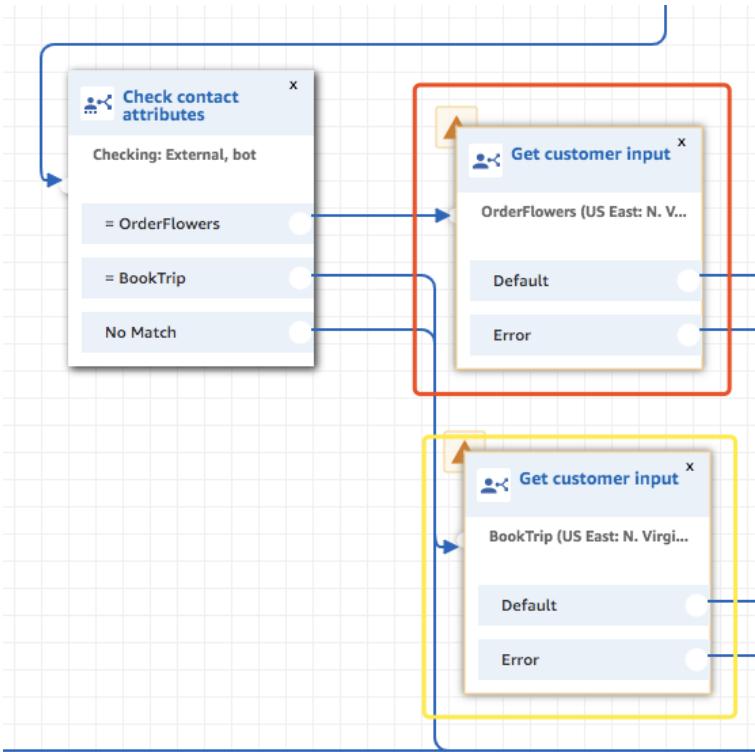
3. Click **Save**ii. **triggerLambdaFunction** and **storeLexConversation**

1. For blue-squared blocks, set the lambda function to **storeLexConversation**
2. For green-squared blocks, set the lambda function to **triggerLambdaFunction**



f. Set correct Lex bots

- For red-squared block, set the bot as OrderFlowers
- For yellow-squared block, set the bot as BookTrip
- Ignore the **alarm icon**



iv.

g. Click **Publish**

h. Click **left sidebar → Routing → Phone numbers**

- Click the only phone number that we claimed earlier

## ii. Set the Contact flow / IVR to Human-Human/Human-Bot Conversations

1.

iii. Click **Save**

20. Done!

**DEPLOY AWS DYNAMO DB**1. Go to [Dynamo DB Management Console](#)2. Click **Create table**3. Deploy **collectionBot** tablea. **Table name:** collectionBotb. **Primary Key:** bot

i.

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

**Table name\*** collectionBot [Tutorial](#) [?](#)

**Primary key\*** Partition key  
bot [String](#) [Add sort key](#)

**Table settings**

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

**Note:** You do not have the required role to enable Auto Scaling by default.  
Please refer to documentation.

+ Add tags [New!](#)

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

[Cancel](#) [Create](#)

c. Click **Create**d. Click **Items**, click **Create item**, fill in **BookTrip**, click Save

## Create item

The screenshot shows a 'Create item' interface. At the top, there's a 'Tree ▾' button and three small icons for sorting. Below that, a section labeled '▼ Item {1}' contains a plus sign icon and a key-value pair: 'bot String : BookTrip'. The value 'BookTrip' is highlighted with a yellow box.

i.

e. Repeat step d for creating item **OrderFlowers**

f. Finally, the table should look like this

The screenshot shows the AWS Lambda console. On the left, a sidebar has 'Create table' and 'Delete table' buttons. A search bar says 'Filter by table name' with 'collectionBot' selected. Below it, a 'Choose a table ...' dropdown and an 'Actions' dropdown are shown. A table with a single row 'collectionBot' is listed under 'Name'. The main area is titled 'collectionBot' with a 'Close' button. It has tabs for 'Overview', 'Items' (which is selected), 'Metrics', 'Alarms', and 'Logs'. Under 'Items', it says 'Scan: [Table] collectionBot: bot'. It shows a 'Scan' dropdown set to 'Scan', an 'Add filter' button, and a 'Start search' button. The results table lists two items: 'bot' with value 'BookTrip' and 'bot' with value 'OrderFlowers'.

g.

4. Deploy **collectionCategory** table

a. **Table name:** collectionCategory

b. **Primary Key:** category

c. Click **Create**

d. Click **Items**

i. Click **Create item**

ii. Set **category** as **enterprise**

iii. Append a new **queueArn** object, set the value to **enterprise Queue ARN** that we obtained earlier

### Create item



1.

iv. Click **Save**

v. Repeat step i - iv for adding item of **travel** and **finance**

e. Finally, the table should look like this:

The screenshot shows the AWS Lambda collectionCategory table. On the left, there is a list of tables: 'collectionBot' and 'collectionCategory'. 'collectionCategory' is selected. On the right, the table details are shown. The table name is 'collectionCategory'. The table has one item: 'collectionSession'. The item details show 'category' as 'enterprise' and 'queueArn' as '5d511325-1f2c-4e2d-93bf-e10f9ed68a62'. Below this, there are tabs for 'Overview', 'Items' (which is selected), 'Metrics', and 'Alarms'. A search bar at the top right shows 'Scan: [Table] collectionCategory: category'. The table list on the left shows 'collectionBot' and 'collectionCategory' again.

i.

5. Deploy **collectionSession** table

a. **Table name:** collectionSession

b. **Primary Key:** category

c. Uncheck **Use default settings** under **Table settings**

d. Click **+Add index** under **Secondary indexes**

e. **Secondary Index:** conversationPIN

**Add index**

**Primary key\*** Partition key  
 String

Add sort key

**Index name\*** conversationPIN-index

**Projected attributes** All   
 Create as Local Secondary Index

**Add index**

i.

## Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is mad

**Table name\*** collectionSession

**Primary key\*** Partition key  
 String   
 Add sort key

### Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or aft

Use default settings

### Secondary indexes

Name	Type	Partition key	Sort key	Projected Attributes	
conversa	GSI	conversation	-	ALL	

[+ Add index](#)

ii.

f. Click **Create**

6. Deploy **userAccount** table

a. **Table name:** userAccount

b. **Primary Key:** PIN

c. Click **Create**

7. Finally, the table overview should look like this:

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity	Auto Scaling	Encryption
collectionBot	Active	bot (String)	-	0	5	5	-	DEFAULT
collectionCategory	Active	category (String)	-	0	5	5	-	DEFAULT
collectionSession	Active	collectionPIN (String)	-	1	10	10	READ_AND_WRITE	DEFAULT
userAccount	Active	PIN (String)	-	0	5	5	READ_AND_WRITE	DEFAULT

a.

8. Done!

### FRAMEWORK CONFIGURATION FILE

Finally, the framework configuration file (`aws_config`) should be in the following format:

```
KEY|space|VALUE
KEY|space|VALUE
```

...

```
AWS_ACCESS_KEY_ID <AWS_secret_access_key_ID>
AWS_ACCESS_KEY <AWS_secret_access_key>
AWS_REGION_NAME <AWS_region_alias>
CALL_RECORDINGS_BUCKET_NAME <AWS_S3_bucket_name>
CONNECT_INSTANCE_ID <AWS_connect_instance_ID>
CONNECT_SECURITY_ID <AWS_connect_security_profile_id_for_agent>
CONNECT_ROUTING_ID <AWS_connect_routing_profile_id>
CONNECT_PHONE_NUMBER <AWS_connect_phone_number>
CONNECT_CCP_URL <AWS_connect_ccp_url>
```

## Deploy Framework Web Interface

### DEPLOY ON LOCAL MACHINES

1. Install dependencies (**Note:** Please ensure that **Python** Version is **3.4+**)

```
a. sudo -H pip3 install --upgrade pip
   sudo -H pip3 install django==2.0.7
   sudo -H pip3 install boto3
   sudo -H pip3 install botocore
   sudo -H pip3 install mock
   sudo -H pip3 install numpy
   sudo -H pip3 install pydub
   sudo -H pip3 install scipy
   sudo -H pip3 install selenium
   sudo -H pip3 install urllib3
```

2. Deploy Django Web Server

a. Create user for login

```
i. cd <directory>
cd src/AWSDeepSenseSpokenDataCollectionFramework/src/ivrFrameworkWebInterface
python3 manage.py createsuperuser
```

b. Run web server using local ports

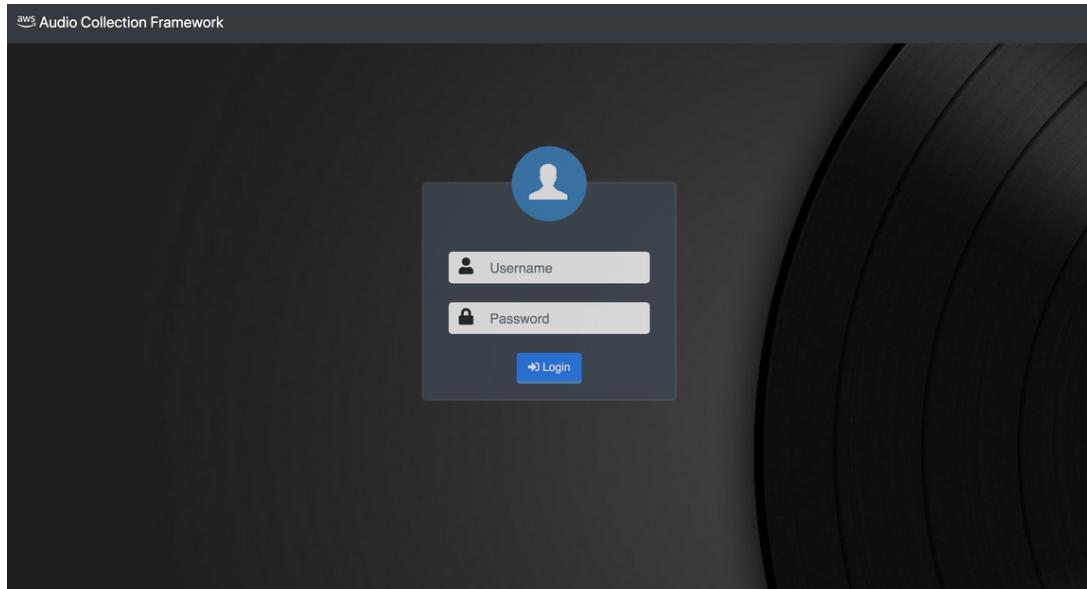
```
i. sudo python3 manage.py runserver
```

c. It should give you something like this

```
i.
38f9d356e78a:ivrFrameworkWebInterface xuzeyuan$ sudo python3 manage.py runserver
Password:
Watching for file changes with StatReloader
Performing system checks...
/usr/local/lib/python3.7/site-packages/pydub/utils.py:165: RuntimeWarning: Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
System check identified no issues (0 silenced).
July 30, 2019 - 18:04:55
Django version 2.2, using settings 'webApps.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

- d. Open web browser, go to <http://127.0.0.1:8000/>

i.



- e. Login using the administer user that we just created in command line

i.

The screenshot shows the AWS Audio Collection Framework interface. At the top, there is a navigation bar with links for 'Collection Requests', 'User Manage', and 'About'. On the right, there are user account details ('admin') and a 'Logout' link. Below the navigation bar, there are two main sections. The first section, titled 'Collection Requests', contains the text 'In the Collection Request portal, you can start new collection requests, and check all ongoing collection requests.' and a blue 'Enter' button. The second section, titled 'User Management', contains the text 'In the User Management portal, you can create new users, delete current users, and check all existing users.' and a blue 'Enter' button.

## Deploy Framework Command Line Interface

1. Install dependencies (**Note:** Please ensure that **Python** Version is **3.4+**)

```
a. sudo -H pip3 install --upgrade pip
   sudo -H pip3 install django==2.0.7
   sudo -H pip3 install boto3
   sudo -H pip3 install botocore
   sudo -H pip3 install mock
   sudo -H pip3 install numpy
   sudo -H pip3 install pydub
   sudo -H pip3 install scipy
   sudo -H pip3 install selenium
   sudo -H pip3 install urllib3
```

2. Print out the command line help message

```
a. cd <directory>
cd src/AWSDeepSenseSpokenDataCollectionFramework/src/aws_deep_sense_spoken_data_collection_framework
sudo python3 framework_runner.py --help
```

3. Follow the help message to use the interactive command line tool

```
38f9d356e78a:aws_deep_sense_spoken_data_collection_framework xuzeyuan$ python3 framework_runner.py -h
/usr/local/lib/python3.7/site-packages/pydub/utils.py:165: RuntimeWarning: Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
usage: framework_runner.py [-h] [-sc] [-gc] [-cs] [-lc] [-ec] [-ea] [-cu]
                           [-lu] [-op] [-du] [-da] [-dc] [-gt]

Framework for Spoken and IVR Data Collection

optional arguments:
  -h, --help            show this help message and exit
  -sc, --startCollection
                        start a new collection request
  -gc, --getCollection  get the information of an ongoing collection request
  -cs, --changeCollectionStatus
                        change the collection status of an onging collection
                        request
  -lc, --listCollection
                        list all ongoing collection requests
  -ec, --endCollection  end a collection request, release the resources
                        (Deprecated)
  -ea, --endAllCollection
                        end all collection requests (Deprecated)
  -cu, --createUser     create a new user as conversation role
  -lu, --listAllUser    list all users
  -op, --openConnectPortal
                        open the browser window for Amazon Connect Contact
                        portal
  -du, --deleteUser    delete a user
  -da, --deleteAllUser  delete all users
  -dc, --download       download call recordings and corresponding metadata
                        from AWS S3
  -gt, --getTranscribe  apply machine transcribe to call recordings for fast
                        benchmarking purpose

38f9d356e78a:aws_deep_sense_spoken_data_collection_framework xuzeyuan$
```

## Operational Activity Guidance

**Important:** Please make sure you finish deployment before you perform any operational activities

### Add a new Human/Human conversation domain

1. Go to [AWS Connect Management Console](#)

2. Go to the instance you want to add domain on

### Amazon Connect virtual contact center instances

Select a virtual contact center instance to manage its directory, administrator(s), telephony options, data storage, and advanced features.

Add an Instance	Remove				
Instance Alias	Access URL	Channels	Create Date	Status	
<input type="checkbox"/> audiocollectionframeworkdemo	<a href="https://audiocollectionframeworkdemo...">https://audiocollectionframeworkdemo...</a>	Inbound, outbound telephony	7/29/2019	Active	

a. Login to **Connect Center Portal**

a. Click **Overview** on the left sidebar, Click **Login as administrator**

i.

Amazon Connect > audiocollectionframeworkdemo

Overview	Instance ARN	arn:aws:connect:us-east-1:843374839531:instance/bbfc83e4-f1a9-40eb-a4c6-0f1b17b78624
Telephony	Directory	audiocollectionframeworkdemo
Data storage	Service-linked role	AWSServiceRoleForAmazonConnect_6uqW1xDeka6sFkuLH9 Learn more
Data streaming	Login URL	<a href="https://audiocollectionframeworkdemo.awsapps.com/connect/login">https://audiocollectionframeworkdemo.awsapps.com/connect/login</a>
Application integration		
Contact flows		

**Login as administrator**

b. You should be then routed to **Connect Center Portal**

i.

Amazon Connect

Dashboard

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. Claim a phone number
 

You must claim a phone number in order to receive and place calls.

[Learn more](#) [View phone numbers](#)
- 2. Set hours of operations
 

Hours of operation define when Amazon Connect resources, such as queues, are available, and may be referenced in contact flows.

[Learn more](#) [View hours of operations](#)
- 3. Create queues
 

Queues allow contacts to be routed to the best agents to service them. If you need to route contacts with different priorities or to agents with different skills, you will want to create multiple queues.

[Learn more](#) [View queues](#)
- 4. Create prompts
 

Prompts are media that can be used to play back audio to customers or agents contact flows. You can upload a pre-recorded .wav file, or quickly record one through our web interface using your computer's microphone. Updates to prompts take immediate effect in all contact flows they are referenced in.

[Learn more](#) [View prompts](#)
- 5. Create contact flows
 

Contact flows (similarly to an IVR) define the customers' experience when they contact you. Amazon

[Learn more](#) [View contact flows](#)
- 6. Create routing profiles
 

A Routing Profile is a collection of queues that an agent will service contacts from. [Routing profile](#)

[Learn more](#) [View routing profiles](#)
- 7. Configure users
 

User Management enables adding, managing, and deleting users. User specific settings like routing profile

[Learn more](#) [View users](#)

4. Add new queue for the new domain

a. Click **left sidebar → Routing → Queues**

b. Click **Add new queue**

c. **Name:** <domain\_name>, **Outbound caller ID name:** Audio Collection Framework, **Outbound caller ID number:** <select the only one>, **Hours of operation:** Basic Hours

i.

Name  
travel

Description  
Queue for having conversations about travel  
207 of 250 characters remaining.

Hours of operation  
Basic Hours

Outbound caller ID name  
Audio Collection Framework  
The name that will show up on the customer's phone

Outbound caller ID number  
+1 217-262-6662

Outbound whisper flow (optional)  
Search for contact flow

Maximum contacts in queue (optional)  
 Set a limit

d. Click **Add new queue**e. Get **Queue ARN**i. Click back into the **travel** queue againii. Parse the url address to get **Queue ARN** (`Queue_ARN=url.split('/')[−1]`)

1.

f. Add queue into routing profiles

i. Click left sidebar → Users → Routing profiles

ii. Edit **Basic Routing Profile**

iii. Add the new queue into the profile

iv.

Name  
Basic Routing Profile

Description  
A simple routing profile.  
225 of 250 characters remaining.

Name	Priority	Delay (in seconds)
enterprise	1	0
finance	1	0
travel	1	0

v. Click **Save**g. Update Queue Info in **Dynamo DB**i. Go to [AWS Dynamo DB Management Console](#)ii. Go to **collectionCategory** table

iii. Create Items

1. Click **Items**2. Click **Create item**3. Set **category** as <domain\_name>4. Append a new **queueArn** object, set the value to **Queue ARN** that we obtained earlier

## Create item

a.

5. Click **Save**

## Add a Lex bot

1. Go to [AWS Lex Management Console](#)
2. Create a bot
  - a. Click **Create**
  - b. Create the lex bot you want to add into the framework, name it as **<bot\_name>**
  - c. Setup code hook function **for each intent**
    - i. Click **BookCar** under **Intents** in the left sidebar
    - ii. Click **Lambda Initialization and Validation**
    - iii. Check Initialization and validation code hook
    - iv. Select **storeLexConversation** as the Lambda function, **Latest** as the **Version or alias**

1.

v. Scroll down and click **Save Intent**

vi. Repeat i - v for each intent

d. Click **Build**, and click **Publish**, enter **<bot\_name>** as in **Create an alias**

i.

The screenshot shows a modal dialog titled "Publish ScheduleAppointment". Inside, a message box says: "Your bot is published! You can now connect to your mobile app or continue to chatbot deployment." Below this, bot details are listed:

Bot Name	ScheduleAppointment
Bot Version	1
Alias	MakeAppointment

To the right, under "What to do next?", there are three sections with links:

- "How to connect to your mobile app" with a "Download connection info" button.
- "Integrate with Mobile hub." with a link to "Please create a project and choose the Conversational Bots feature in Mobile Hub".
- "How to deploy your bot to other services" with a link to "Learn how to deploy your bot to other services like Facebook Messenger, Slack, Twilio, and Kik." and a "Go to channels" button.

At the bottom right of the dialog is a "Close" button.

## 3. Update Queue Info in Dynamo DB

- Go to [AWS Dynamo DB Management Console](#)
- Go to **collectionBot** table
- Create Items
  - Click **Items**
  - Click **Create item**
  - Set **bot** as `<bot_name>`

The screenshot shows the "Create item" interface in the AWS DynamoDB console. At the top, it says "Create item". Below that is a toolbar with "Tree ▾" and navigation icons. The main area shows a single item with the key "bot" and value "String : MakeAppointments".

iv.

- Click **Save**

## 4. Update AWS Connect instance

- Go to [AWS Connect Management Console](#)
- Go to the instance you want to add domain on
  -

i.

**Amazon Connect virtual contact center instances**

Select a virtual contact center instance to manage its directory, administrator(s), telephony options, data storage, and advanced features.

Add an Instance	Remove	Instance Alias	Access URL	Channels	Create Date	Status
<input type="checkbox"/> audiocollectionframeworkdemo			<a href="https://audiocollectionframeworkdem...">https://audiocollectionframeworkdem...</a>	Inbound, outbound telephony	7/29/2019	Active

c. Click **Contact flows**d. Select the new created lex bot, click **+Add Lex Bot**

i.

**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.

Note: By adding Lex bots, you are granting Amazon Connect permission to interact with them [Create a new Lex bot](#)

Region: US East: N. Virginia    Bot: ScheduleAppointment    + Add Lex Bot

**Lex bots**

BookTrip	<a href="#">Remove</a>
OrderFlowers	<a href="#">Remove</a>

5. Login to **Connect Center Portal**a. Click **Overview** on the left sidebar, Click **Login as administrator**

i.

Amazon Connect > audiocollectionframeworkdemo

**Overview**

Instance ARN	arn:aws:connect:us-east-1:843374839531:instance/bbfc83e4-f1a9-40eb-a4c6-0f1b17b78624
Directory	audiocollectionframeworkdemo
Service-linked role	AWSServiceRoleForAmazonConnect_6uqW1xDeka6sFkuXhLH9 <a href="#">Learn more</a>
Login URL	<a href="https://audiocollectionframeworkdemo.awsapps.com/connect/login">https://audiocollectionframeworkdemo.awsapps.com/connect/login</a>

[Login as administrator](#)

b. You should be then routed to **Connect Center Portal**

i.

**Dashboard**

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.

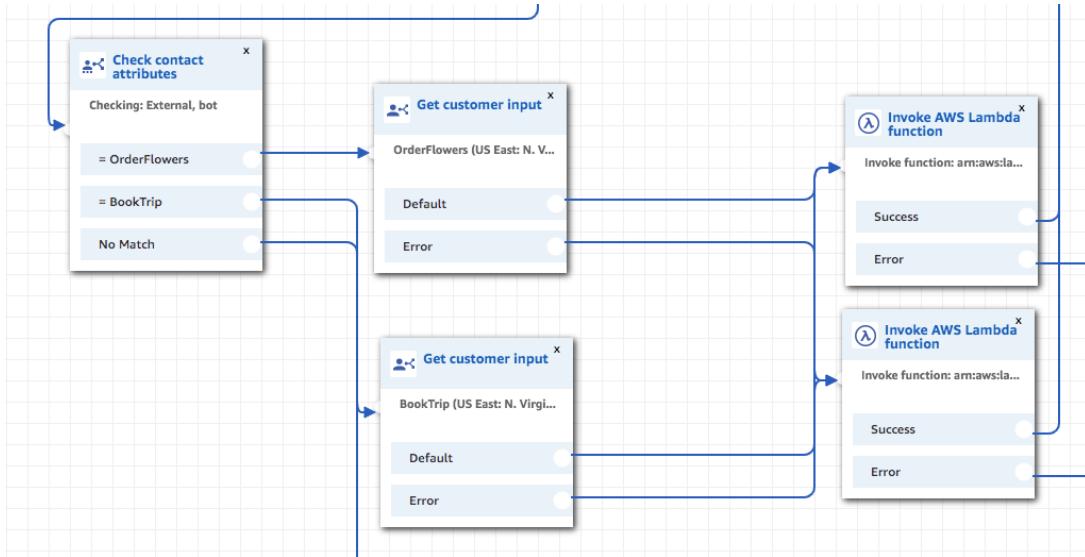
- 1. Claim a phone number**  
You must claim a phone number in order to receive and place calls.  
[Learn more](#) [View phone numbers](#)
- 2. Set hours of operations**  
Hours of operation define when Amazon Connect resources, such as queues, are available, and may be referenced in contact flows.  
[Learn more](#) [View hours of operations](#)
- 3. Create queues**  
Queues allow contacts to be routed to the best agents to service them. If you need to route contacts with different priorities or to agents with different skills, you will want to create multiple queues.  
[Learn more](#) [View queues](#)
- 4. Create prompts**  
Prompts are media that can be used to play back audio to customers or agents contact flows. You can upload a pre-recorded .wav file, or quickly record one through our web interface using your computer's microphone. Updates to prompts take immediate effect in all contact flows they are referenced in.  
[Learn more](#) [View prompts](#)
- 5. Create contact flows**  
Contact flows (similarly to an IVR) define the customers' experience when they contact you. Amazon
- 6. Create routing profiles**  
A Routing Profile is a collection of queues that an agent will service contacts from. [Routing profiles](#)
- 7. Configure users**  
User Management enables adding, managing, and deleting users. User specific settings like routing profiles

## 6. Deploy contact flow

a. Click **left sidebar → Routing → Contact flows**b. Edit contact flow **Human-Human/Human-Bot Conversations**

## c. Locate Lex Bot Branching

i.



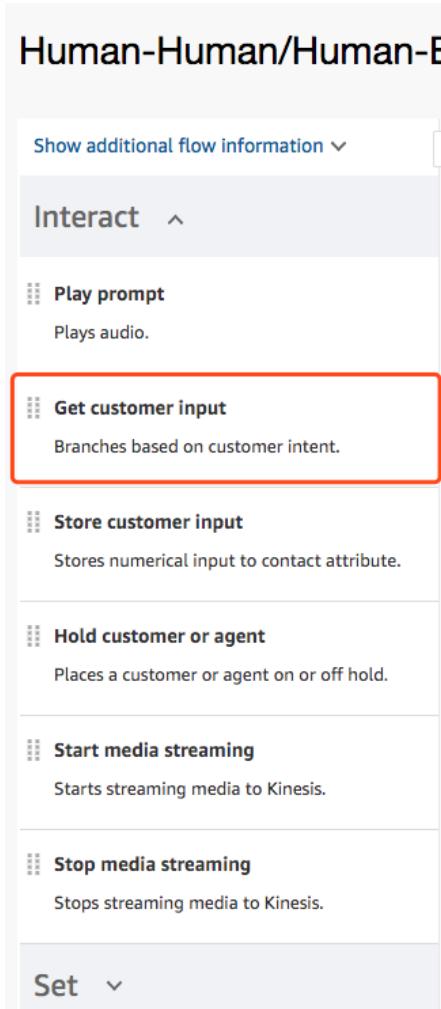
## d. Edit Check contact attributes

i. Click Add another condition

ii. Add Equals: &lt;bot\_name&gt;

iii. Click Save

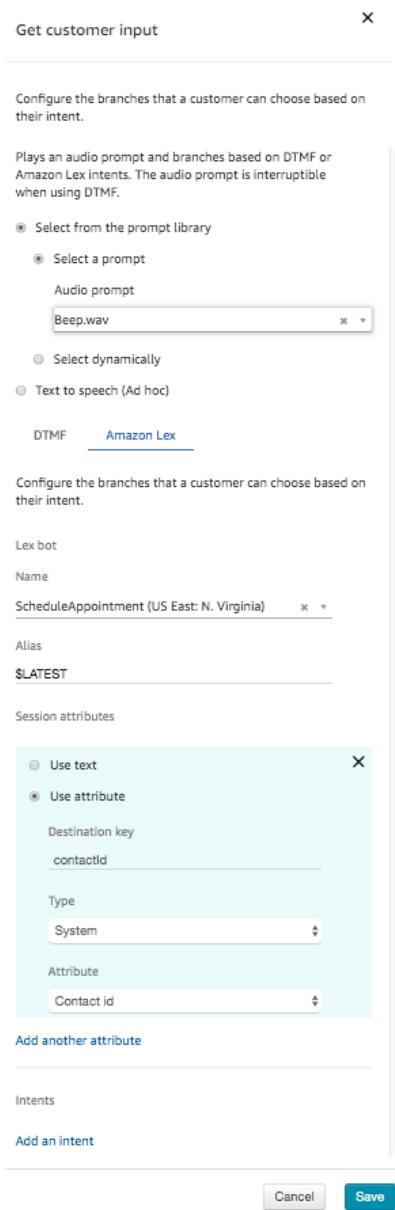
e. On the left toolkit, click Integrate, drag Get customer input block to the contact flow chart



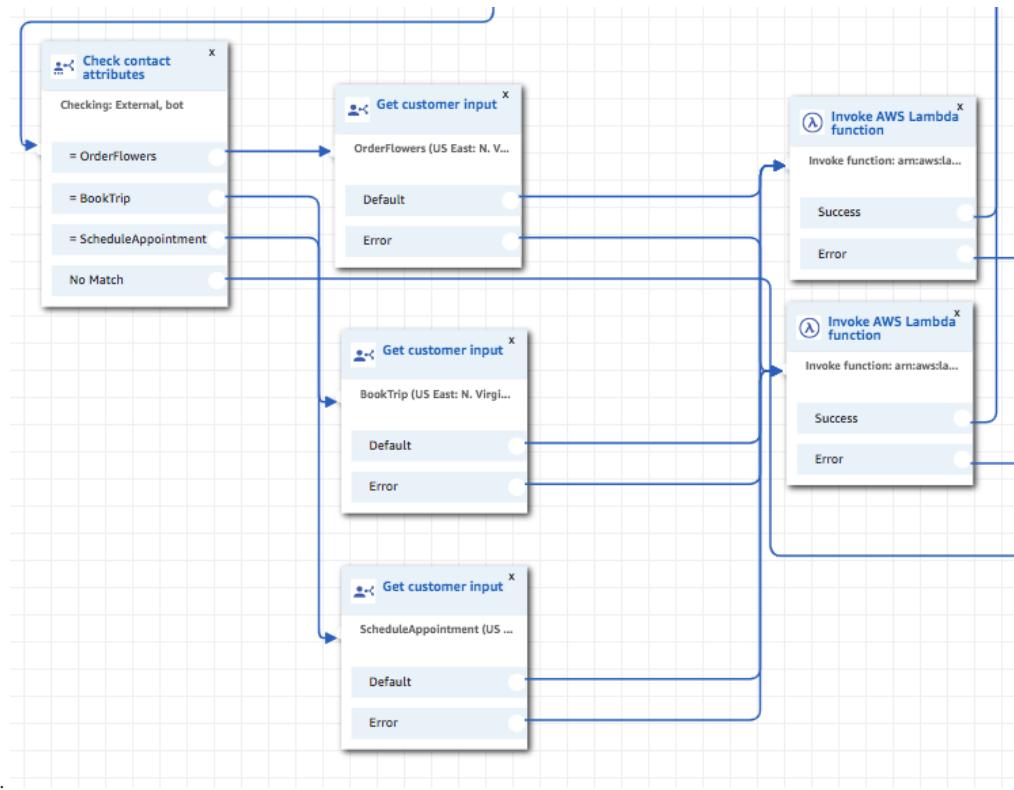
## f. Edit the new Get customer input block

i. Click the module to edit

- ii. Check **Select from the prompt library**
- iii. Check **Select a prompt**, select **Beep.wav**
- iv. Click **Amazon Lex**, select the new created Lex bot
- v. Click **Add an attribute**, click **Use attribute**
- vi. set **Destination key** as **contactId**
- vii. set **Type** as **System**
- viii. set **Attribute** as **Contact id**



- ix.
- x. Click **Save**
- g. Connect the **Get customer input** into the contact flow
- h. Finally, it should look like this



i.

- i. Click **Publish** on the top right