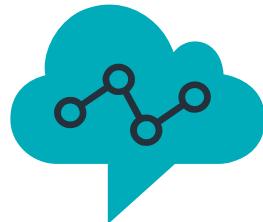




Amazon Connect Hands-On Guide (Level 100)



Instructor: Lili Chan
Sr Amazon Connect SA, ASEAN
Email: lilichan@amazon.com

May 2021 rev 2.2



Introduction

The purpose of this guide is to walk through several common use cases frequently requested by customers from different industrial verticals.

The exercises in this guide will slowly introduce concepts that build on top of each other. It is preferred to go sequentially through the modules start, but due to time constraints participants can skip to certain modules of their focus areas. Certain labs will present methods that use additional AWS services that compliment Amazon Connect. Ensure the account used to follow along have the necessary level of access for both Amazon Connect and these additional AWS services.

At the completion of this guide, you will learn how to:

- Create a new AWS account (under AWS Free Tier).
- Set access right/permission for Lambda within AWS (IAM – Identity and Access Management).
- Create a DynamoDB table to be used for the “Holiday” announcement lab.
- Create a Lambda function to be used for the “Holiday” announcement lab.
- Get started with Amazon Connect – create a new Amazon Connect instance.
- Configuration settings for Amazon Connect.
- Understand how to export and import a contact flow.
- Step through the Connect contact flow - Dynamic announcements/prompts, Configure business hours of operation, Queues, Agents management, Routing profile, etc.
- Create another contact flow for caller identification and personalization.
- Work with customer data that is available or that is entered during an existing call.
- Query and import external data to make decisions and cater the experience to callers.
- Create your own Amazon Lex Bot.
- Use Amazon Lex as a voice bot within your contact flow.
- Incorporate one-time pin 2FA SMS authentication using Amazon Pinpoint.



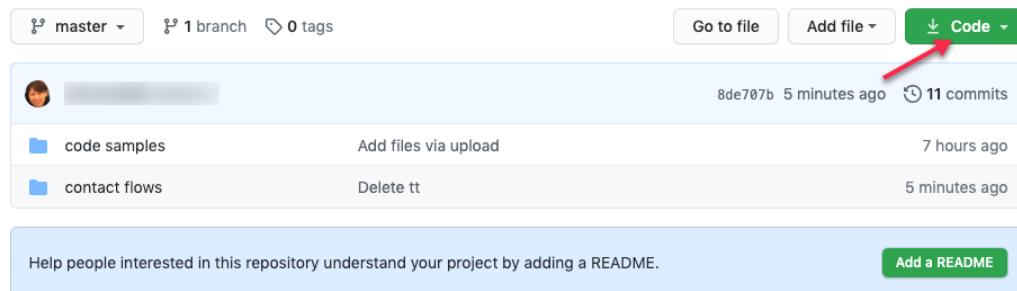
Revisions

Sep 2020	Version 1.0
Nov 2020	Version 2.0 Added section 8, 8.1 and 8.2 for caller identification and personalization. Create new DDB, lambda and contact flow. Added section 9, 9.1 for Amazon Lex voice bot.
May 2021	Version 2.1 Added section 10 for 2FA SMS Pin authentication using Amazon Pinpoint. Create new Lambda for oneTimePin. Version 2.2 Made some corrections. Tested SMS working.

Prerequisites

1. Your laptop / workstation needs to be multi-media enabled (speaker, mic).
2. Internet access is required.
3. Either Chrome or Firefox (FF) browser is supported (up to last 3 versions).
4. Download the following files required for the hands-on session using below URL link:
<http://bit.ly/lilichanworkshop>

Click on the bit.ly link, you should see a green button named “code” as follows, click on it and select the option “Download ZIP” to download all the contents as a ZIP file.



Throughout this hands-on lab guide, you will be referred back to the contents in this ZIP file.



*Part 1: Create your own AWS Account

***Note:** If you already have an AWS account and want to access the AWS management console, you can skip this Part 1 & go directly to Part 2.

Objective: Create an AWS instance

By the end of this section, you will have created an AWS account where you can access AWS services which you will be using throughout this session.

- 1) If you do not have a AWS account, navigate to <https://portal.aws.amazon.com/billing/signup#/start>. Go through the steps to have your own FREE account (AWS Free Tier).

The screenshot shows a web-based sign-up form titled "Create an AWS account". It contains four input fields: "Email address", "Password", "Confirm password", and "AWS account name". Below these fields is a yellow "Continue" button. At the bottom left of the form, there is a link "Sign in to an existing AWS account".

- 2) Click on “Professional”, enter in your company’s address and information.
- 3) Enter in your Credit Card information. Note that the \$1 charge will be removed once your credit card is verified.

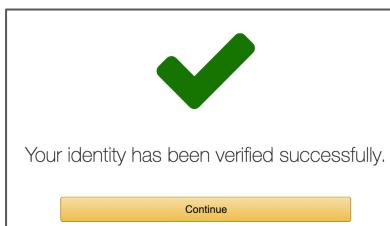


We use your payment information to verify your identity and only for usage in excess of the [AWS Free Tier Limits](#). We will not charge you for usage below the AWS Free Tier Limits. For more information, see the [frequently asked questions](#).



When you submit your payment information, we will charge \$1 USD/EUR to your credit card as a verification charge to ensure your card is valid. The amount may show as pending in your credit card statement for 3-5 days until the verification is completed, at which time the charge will be removed. You may be redirected to your bank website to authorize the verification charge.

- 4) Verification of your account via SMS or Phone Call.



- 5) Click on “Free”



- 6) Sign into the Management Console. Input the email address and password used.

Sign in

Email address of your AWS account
Or to sign in as an IAM user, enter your [account ID](#) or [account alias](#) instead.

Next

You are all set now!



Part 2: AWS Management Console

Objective: Getting yourself familiar with the AWS Management Console

If you have skip Part 1 and you want to login to AWS Management Console, go to:

<https://aws.amazon.com/> and click on [Sign In to the Console](#)

Below is the landing page you will see once you log into the AWS Management Console. From the top right-hand corner, you will see the Region being selected. Please note to have “Singapore” selected unless the AWS service you required is not available in the Singapore region, then you will need to select the country you require from the drop-down list.

You will also see the logged in AWS account name (see the 1st arrow below), and if you click on the “down arrow” next to it, you can see your AWS account number which will be useful if you want to raise a support case ticket, etc.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a dropdown menu for 'Services'. Below the search bar is a 'Recently visited services' section with links to DynamoDB, Lambda, Amazon Lex, S3, Amazon Kendra, Amazon Connect, Amazon Polly, IAM, and Billing. A large 'AWS services' sidebar on the left lists various service categories like Compute, Storage, Database, and more, each with a list of specific services. To the right of the sidebar is the main content area. On the far right, there are three boxes: 'Stay connected to your AWS resources on-the-go' (with a link to download the mobile app), 'Explore AWS' (listing services like Amazon S3 Glacier, Amazon SageMaker Autopilot, AWS CloudTrail, and AWS Training), and 'Have feedback?' (with a link to submit feedback). At the bottom, there are links for 'Feedback', 'English (US)', and 'Privacy Policy/Terms of Use'.

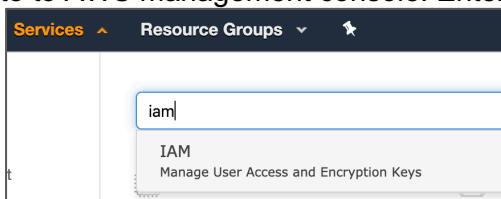
Part 3: IAM (Identity and Access Management)

Objective: Setting up the rights and permissions for your lambda function.

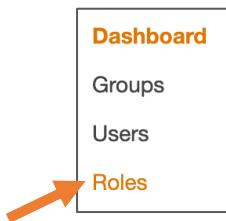
By the end of this part, you should be able to navigate and understand where to set up restricted rights for services in AWS.

Note: The example below grants administrator access to Lambda functions to call other AWS services on your behalf for Lab purposes. In a typical production environment, you might want to consider least privilege access.

Navigate to AWS management console. Enter in **IAM**.



- 1) On the left-hand column, click on **Roles**.



- 2) Click on the button in blue **Create Role**
- 3) Click on **Lambda** and click on **Next: Permission** button.



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	CodeDeploy	ElastiCache	Lambda	S3
AWS Backup	Comprehend	Elastic Beanstalk	Lex	SMS
AWS Chatbot	Config	Elastic Container Service	License Manager	SNS
AWS Support	Connect	Elastic Transcoder	Machine Learning	SWF
Amplify	DMS	Elastic Load Balancing	Macie	SageMaker
AppStream 2.0	Data Lifecycle Manager	Forecast	MediaConvert	Security Hub
AppSync	Data Pipeline	Global Accelerator	Migration Hub	Service Catalog
Application Auto Scaling	DataSync	Glue	OpsWorks	Step Functions
Application Discovery	DeepLens	Greengrass	Personalize	Storage Gateway

* Required

Cancel

Next: Permissions

- 4) Select AdministratorAccess and click Next: Tags.

Showing 569 results			
	Policy name	Used as	Description
<input checked="" type="checkbox"/>	AdministratorAccess	Permissions policy (1)	Provides full access to AWS services an...
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	None	Provide device setup access to AlexaFor...
<input type="checkbox"/>	AlexaForBusinessFullAccess	None	Grants full access to AlexaForBusiness r...
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	None	Provide gateway execution access to Al...
<input type="checkbox"/>	AlexaForBusinessNetworkProfileServicePolicy	None	This policy enables Alexa for Business to...
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	None	Provide read only access to AlexaForBu...
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	None	Provides full access to create/edit/delete...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	None	Provides full access to invoke APIs in A...

Set permissions boundary

* Required

Cancel Previous Next: Tags

- 5) No need for any tags, click on **Next: Review**

- 6) Give the role a name **lambdaAdmin**. And go ahead with the creation of the role.



Review

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

Role name* lambdaAdmin

Use alphanumeric and '+,.,@-_ ' characters. Maximum 64 characters.

- 7) All set for your permission setup for the lambda function which we are going to use later on!

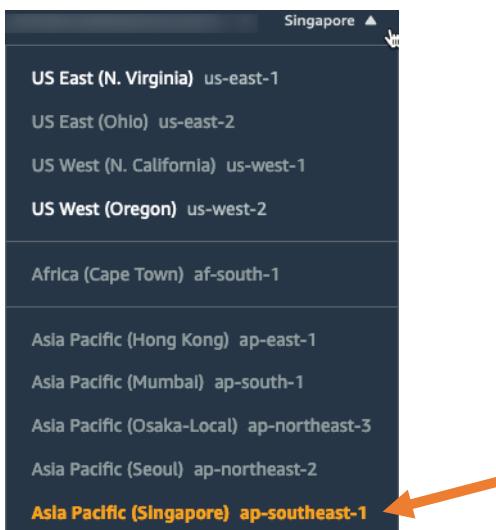
*Part 4: Create an Amazon Connect Instance

***Note:** If you already have created an Amazon Connect instance and want to use that instead, you can skip this Part 4 and go directly to Part 5.

Objective: Creating your very first Amazon Connect instance

By the end of this section, you will have created an Amazon Connect Instance on your AWS account. This instance will be used in future labs, so it is essential this section be completed in its entirety.

- 1) Check that you are in the **Asia Pacific (Singapore) ap-southeast-1** region.



- 2) Within the AWS Search bar, type **Amazon Connect** and click the header that appears.



Find Services
You can enter names, keywords or acronyms.

Amazon Connect

Amazon Connect
Amazon Connect is a contact center that enables engagement at any scale.

- 3) If this is your first time navigating to Amazon Connect, click **Getting Started**. Otherwise, you will see a list of instances that have already been built with an **Add an Instance** button. Click either **Add an Instance** or **Getting Started**.

The screenshot shows the Amazon Connect landing page. It features a blue icon of a speech bubble with a gear inside. Below the icon, the text "Amazon Connect" is displayed. A paragraph of text explains what Amazon Connect is and its benefits. At the bottom of the page is a blue "Get started" button, which has an orange arrow pointing towards it from the left side of the image.

- 4) Leave the **Store users within Amazon Connect** selected and name your Amazon Connect Instance. Click **Next** to continue.

NOTE: Amazon Connect Instances require globally unique names. Select a unique name else it will be rejected.

- 5) Enter your name and password.
The next screen asks for you to enter a local Administrator for your Amazon Connect instance.
For simplicity, enter the same credentials used for the AWS Account login. Click **Next Step**
- 6) Leave the Inbound and Outbound options checked and click **Next Step**.

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.

- 7) Click **Next Step**



The Data Storage screen allows you to customize where call reports and recording are to be stored. To explore the details of these settings, click **Customize Settings**, else move along and click **Next Step**.

- 8) Review the details listed on the final confirmation screen and click **Create Instance**.
- 9) Once complete, click **Getting Started** on the success page.
- 10) The creation process will begin and take about **2 minutes** to complete. Once complete you will have a fully enabled Amazon Connect contact center instance ready for configuration and calls.

Success!

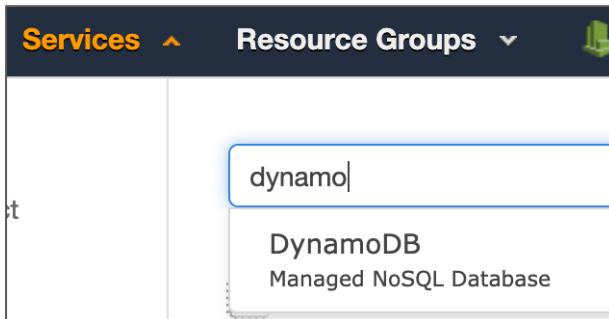
Your Amazon Connect has been created.
Now you can begin to choose phone numbers, accept calls, and engage with your customers.

[Get started](#)

Part 5: DynamoDB Table

Objective: Using a DynamoDB table to store holidays / special events / ad-hoc messages

- 1) Go back into your management console. Search for **DynamoDB**. Click into it.



- 2) Click on **Create table**
- 3) Enter in **holidayCalendar** for the table name (or any other table name you like), and **dateStart** for the primary key. Click on the blue Create table at the bottom of the page. The table will take a short while to be created.

Note: Names are cap-sensitive

Create DynamoDB table

Tutorial

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*	<input type="text" value="holidayCalendar"/>
Primary key*	Partition key
	<input type="text" value="dateStart"/>
<input type="checkbox"/> Add sort key	

- 4) Create new line items in the table. Click on **Create item**.

- 5) Enter in **dateStart**, **dateEnd** and **reason**. To add more fields, click on the plus sign and **Append**. Again, do take note that field names are cap sensitive.

Create item

The screenshot shows the AWS Lambda 'Create item' interface. A yellow box highlights the 'dateStart' field under 'Item {1}'. An orange arrow points to the 'Number' option in the dropdown menu that appears when the 'Append' button is clicked.

Your table should look similar to the following:

holidayCalendar [Close](#)

Overview	Items	Metrics	Alarms	Capacity	Index
Create item	Actions ▾				
Scan: [Table] holidayCalendar: dateStart ^ Scan <input type="button" value="▼"/> [Table] holidayCalendar: dateStart Add filter <input type="button" value="Start search"/>					
	dateStart ⓘ	dateEnd	reason		
<input checked="" type="checkbox"/>	1598976000	1599062399	Test Holiday One		
<input type="checkbox"/>	1599062400	1599148799	Test Holiday Two		
<input type="checkbox"/>	1600128000	1600214399	Test Holiday Three		
<input type="checkbox"/>	1608825600	1608911999	Christmas		
<input type="checkbox"/>	1609430400	1609516799	New Year		

- 6) If you want to edit or append more lines to your table, just select any of the row you want to edit or append to and click on the **Actions** button then click on **Edit**.

holidayCalendar [Close](#)

Scan: [Table] holidayCalendar

	dateStart	dateEnd	reason
<input checked="" type="checkbox"/>	1598976000	1599062399	Test Holiday One
<input type="checkbox"/>	1599062400	1599148799	Test Holiday Two
<input type="checkbox"/>	1600128000	1600214399	Test Holiday Three
<input type="checkbox"/>	1608825600	1608911999	Christmas
<input type="checkbox"/>	1609430400	1609516799	New Year

If you want to change the attribute value, just overwrite the existing value and click **Save**.

Edit item

Tree ▾

- ▼ Item {3}
 - + dateEnd Number : 1599062399
 - + dateStart Number : 1598976000
 - + reason String : Test Holiday One

- 7) Now, it's time to populate the table with data!

We will be using Christmas as an example.

Firstly, the dateStart and dateEnd are in Epoch timestamp format so we will need to convert the human-readable date/time to epoch timestamp using the following link:

<https://www.epochconverter.com/>

Below is the dateStart for Christmas in epoch format - **1608825600**

Yr Mon Day Hr Min Sec

2020 - 12 - 25 0 : 0 : 0 Local time ▾ Human date to Timestamp

Epoch timestamp: 1608825600

Timestamp in milliseconds: 1608825600000

Date and time (GMT): Thursday, 24 December 2020 16:00:00

Date and time (your time zone): Friday, 25 December 2020 00:00:00 GMT+08:00



8)

Below is the dateEnd for Christmas in epoch format – **1608911999**

Yr	Mon	Day	Hr	Min	Sec
2020	12	25	23	: 59	: 59

Local time ▾ Human date to Timestamp

Epoch timestamp: 1608911999

Timestamp in milliseconds: 1608911999000

Date and time (GMT): Friday, 25 December 2020 15:59:59

Date and time (your time zone): Friday, 25 December 2020 23:59:59 GMT+08:00

Using Epoch timestamp, you can have very granular date/time intervals up to seconds. You can also have more than one day timestamp interval for e.g., Chinese New Year.

9) Your table will look something as follows, taking Christmas as an example.

dateStart ⓘ	dateEnd	reason
1598976000	1599062399	Test Holiday One
1599062400	1599148799	Test Holiday Two
1600128000	1600214399	Test Holiday Three
1608825600	1608911999	Christmas

10) Do take note to input the holiday name or the message prompt or announcement as the reason attribute value can be used as a dynamic prompt later on within the Amazon Connect contact flow using Amazon Polly (Amazon Text-to-Speech engine).

Note: For other lab exercises, if no data type is being specified, then use the **String** data type as default.

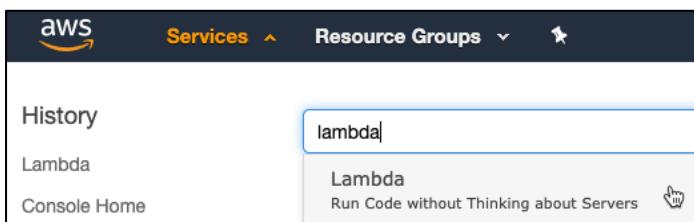
11) All set for your table! You can come back to DynamoDB to add more lines or edit this table.

Part 6: Lambda

Objective: Creating your lambda code to check whether current date/time stamp falls within the dateStart and dateEnd time intervals for the DynamoDB holiday table.

Lambda is an AWS service that lets you run serverless codes without a need to provision or manage any servers. This will be the main way for you to run codes, perform database dip etc., for the lab exercises.

- 1) Go to your Amazon management console. Search for **Lambda**.



- 2) Click on the orange button on the right **Create function**.
- 3) Enter in function name of **holidayCalendarRead**, or any name you prefer. Leave the runtime as **Node.js 12.x or above**.
- 4) Click on the arrow for the dropdown of Choose or create an execution role. Select Use an existing role.
- 5) Select **lambdaAdmin** in the IAM section above which you have previously created.



Basic information

Function name

Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function.

Permissions Info

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.

▼ Choose or create an execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

[View the lambdaAdminAccess role](#) on the IAM console.

6) Click on **Create function** button.

- 7) From the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named “code samples”, click into it and locate the holidayCalendarRead.js file.
- 8) Copy and paste all the codes in there into the new lambda function as shown below.
- 9) Save the lambda function for e.g., **holidayCalendarRead**

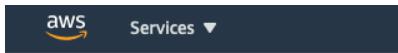
```
const AWS = require('aws-sdk');
const docClient = new AWS.DynamoDB.DocumentClient({region: 'ap-southeast-1'});

exports.handler = (event, context, callback) => {
    //var d = Date.now()
    //console.log(d);
    var seconds = Math.round(d.getTime() / 1000);
    console.log(seconds);
    let dateNow = new Date(seconds).toLocaleString('en-US', {timeZone: 'Asia/Singapore'});
    console.log(dateNow);
    //console.log('spore time: ' + (new Date(sporeTime)).toISOString());
    const scanningParams = {
        TableName: 'holidayCalendar',
        Limit: 10,
        FilterExpression: ":dateNow between :dateStart and :dateEnd",
        ExpressionAttributeValues: {
            ':dateNow': dateNow,
            ':dateStart': seconds
        }
    };
    callback(null, scanningParams);
};

Execution Result
Execution results
No execution results yet
```

10) Go on to management console, select **Amazon Connect**.

11) Click on **Contact Flows**



Amazon Connect > sgconnectinstance1

Overview
Telephony
Data storage
Data streaming
Application integration
Contact flows

- 12) Scroll down and search for **AWS Lambda**. Add in the **holidayCalendarRead** function you have just created. And click **+ Add Lambda Function**

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](#)

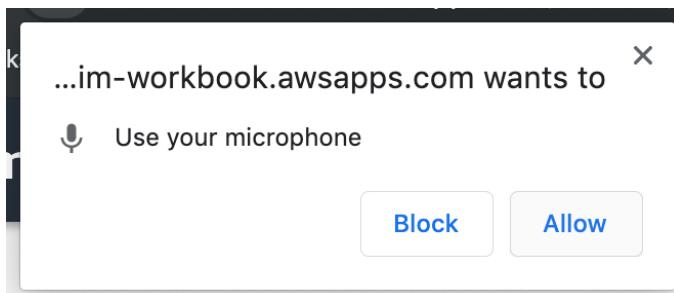
Function [+ Add Lambda Function](#)

Lambda Functions

Part 7: Configuring Amazon Connect

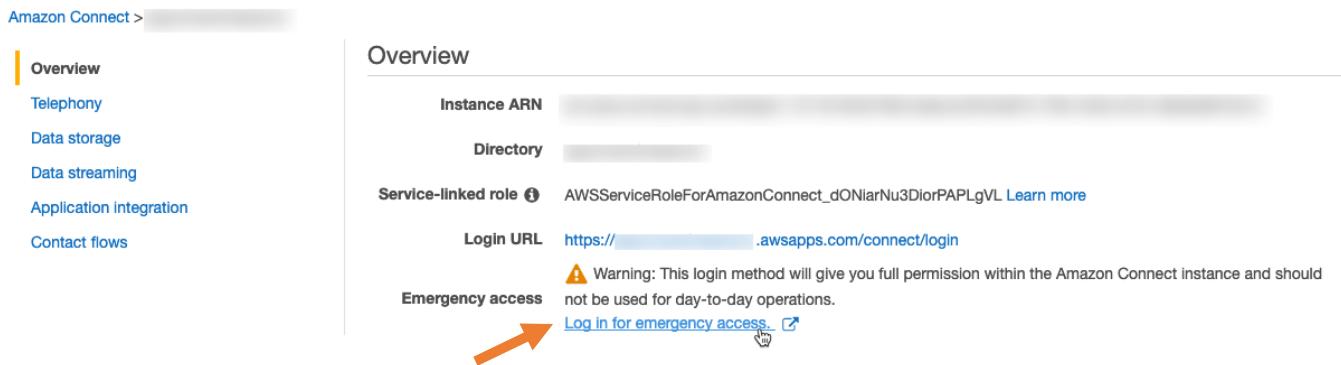
Objective: Setting up your Amazon Connect instance

NOTE: If the popup appears, please click on Allow. This is for the Amazon Connect CCP softphone to use your laptop/computer's microphone.



- 1) Go to Management Console. Click on Amazon Connect.
- 2) Navigate to the Emergency access URL link created for your Amazon Connect instance shown below and click to log in.

If you have created your own Amazon Connect administrative username and password earlier on, you can use that to login using the Access URL created as well.

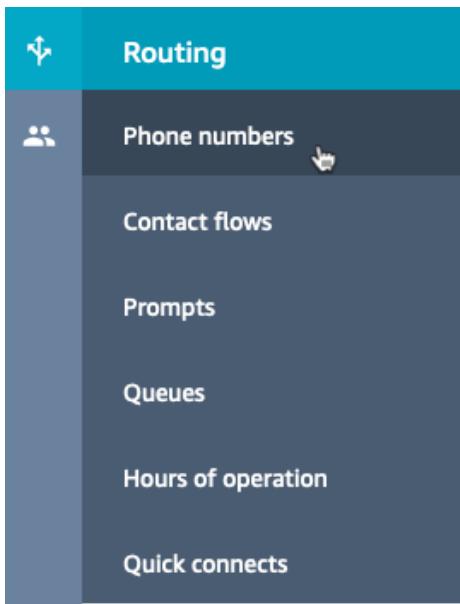


The screenshot shows the 'Overview' page of an Amazon Connect instance. On the left, there's a sidebar with links: Overview (which is selected and highlighted in yellow), Telephony, Data storage, Data streaming, Application integration, and Contact flows. The main area has a heading 'Overview' and several sections: 'Instance ARN' (redacted), 'Directory' (redacted), 'Service-linked role' (AWSRoleForAmazonConnect_dONiarNu3DiorPAPLgVL) with a 'Learn more' link, 'Login URL' (https://[redacted].awsapps.com/connect/login), and 'Emergency access'. The 'Emergency access' section contains a warning message: 'Warning: This login method will give you full permission within the Amazon Connect instance and should not be used for day-to-day operations.' Below the message is a blue link 'Log in for emergency access.' An orange arrow points to this blue link.

Part 7.1: Claim number

Amazon Connect supports numbers claimed directly for the service. Numbers claimed will become linked to a Contact flow once saved. Amazon Connect includes several default Contact flows which will be referenced, for now, until we are ready to build our own Contact flow for use.

- 1) Navigate to the Routing > Phone Number section



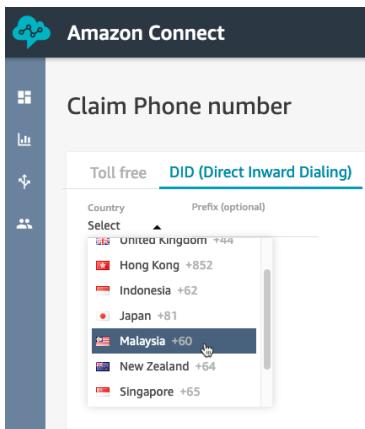
2) Click Claim a number.

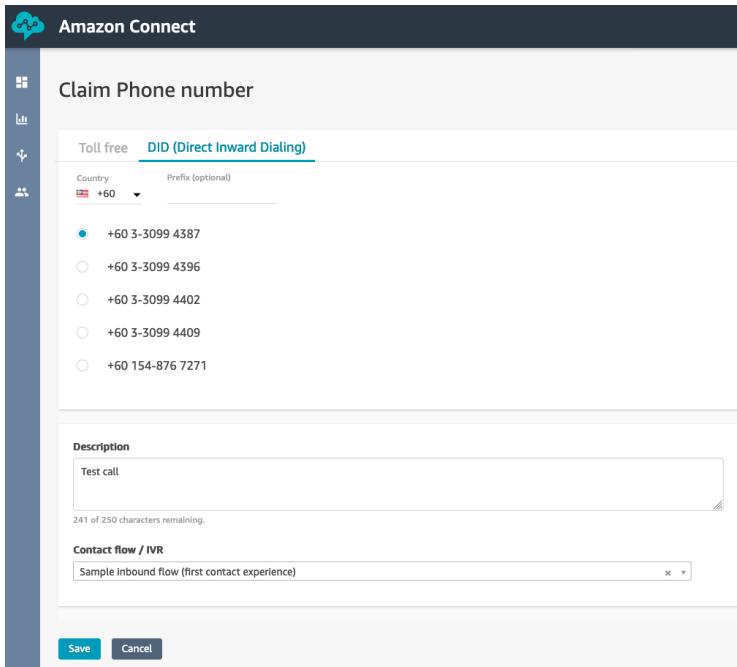
There are both **Toll Free** and **DID** options.

Select **DID** and look at the list of countries available.

You will notice that most of the ASEAN countries like Singapore, Thailand, Indonesia require you to raise a support case ticket to request for a number, this is because these countries have telecom regulatory requirements to ask for business registration information or individual photo-based identification info before approving to the request.

For the lab, you can select **Malaysia** for country. Choose a number listed.





You can map this chosen number to a sample contact flow already provided within Amazon Connect known as Sample inbound flow (first contact experience) and make your first test call.

Part 7.2: Queues

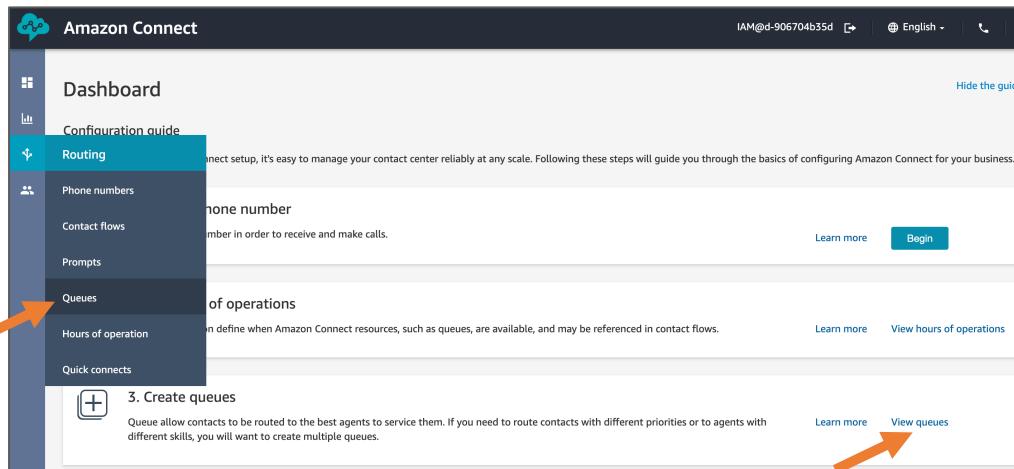
Objective: Creating Queues for calls to be routed to and queues for agents to work.

By the end of this lab, you will have created a series of queues that calls which enter Amazon Connect can be routed to. You will also have been introduced to the concept of Routing Profiles and better understand how these profiles are applied to agents and used when making call routing decisions.

Amazon Connect implements the concept of Routing Profiles.

A Routing Profile is a collection of created queues that are arranged based on priority and differing delay values. Once a routing profile is created, it becomes easy to assign this profile to an agent who inherits the queues contained within the profile. Pending calls become available for agents based on priority and delay values configured within the routing profile. Routing Profiles make it easy to make bulk changes that impact many users with a single configuration change. Sudden changes in call volume can be addressed by adjusting delay values that traditionally would require rebasing in order to compensate for the sudden increase in call volume. Lastly, routing profile values are limited to the discrete queues that have been created. This prevents significant overhead common in skills management and allows routing administrators and IVR administrators collaboration when designing a customer-centric contact center.

- 1) Navigate to the **Routing > Queues** section.



- 2) Notice the **Basic Queue** option. Amazon Connect comes preconfigured with a sample queue to help get started. Leave this alone and select **Add new queue**.
- 3) Queue names should reflect the skills required to solve problems. When naming, ensure the name aligns with the skill required for agents who work this queue. For example, if agents are expected to have knowledge in general credit card questions, name your queue **General Credit Card** questions. However, if you'd like to separate agents who have specialized skills, such as **Lost or Stolen cards**, name it accordingly.

In our example, we'll be building several queues.

Enter a **description** for your queue.

Hours of Operation are required; for now, select **Basic Hours**.

A callback number of the queue is also required in case configured to allow callbacks. Select this box and choose the number listed. This number is the one created in Part 7.1.

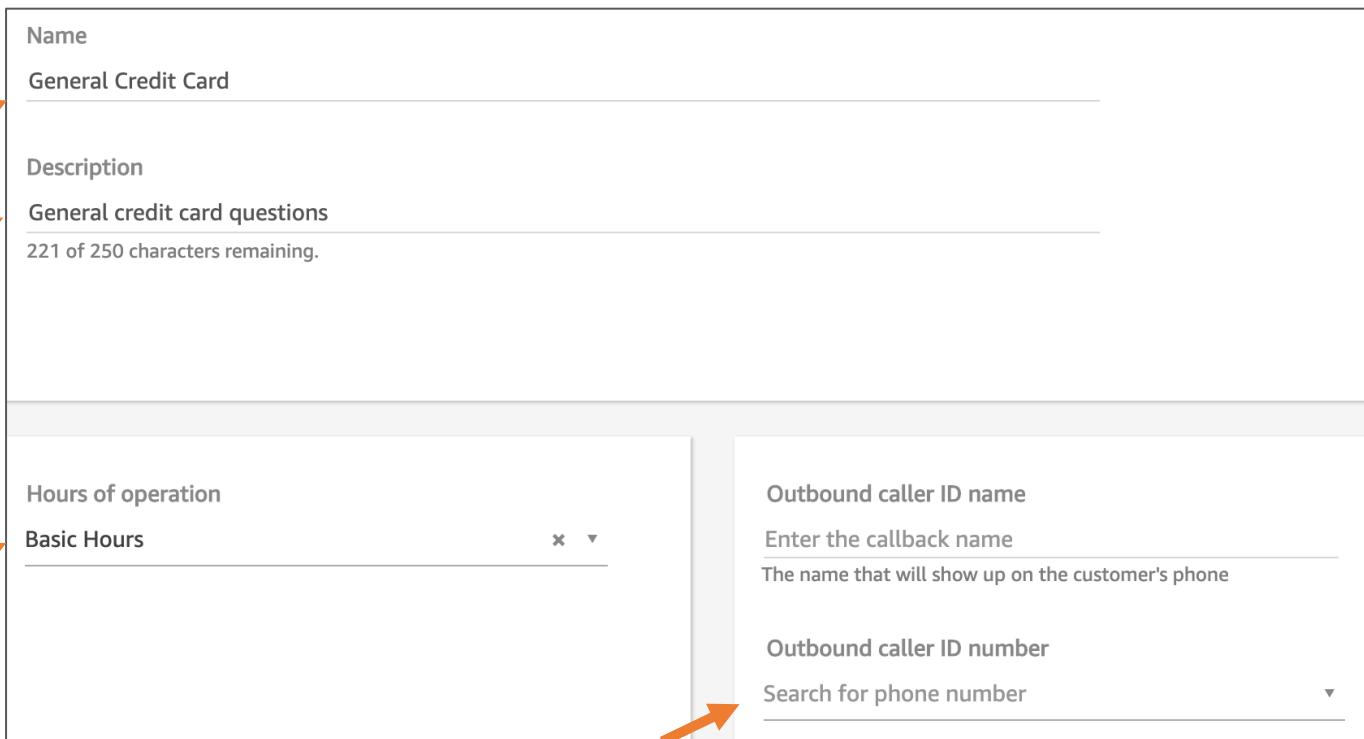
Name
General Credit Card

Description
General credit card questions
221 of 250 characters remaining.

Hours of operation
Basic Hours

Outbound caller ID name
Enter the callback name
The name that will show up on the customer's phone

Outbound caller ID number
Search for phone number



Leave the remaining options as they are and click **Add new queue**

Repeat this process for the queues listed below. By the end, you should have **3** discrete queues configured for use.

- 4) Repeat step 3. For 2 other queues with queue names below.
 - a. General Credit Card
 - b. Fraud Assistance
 - c. Technical Assistance

Part 7.3: Routing Profiles (Optional)

Objective: Configuring routing profiles

Gives an understanding of how to configure agents to the queues.

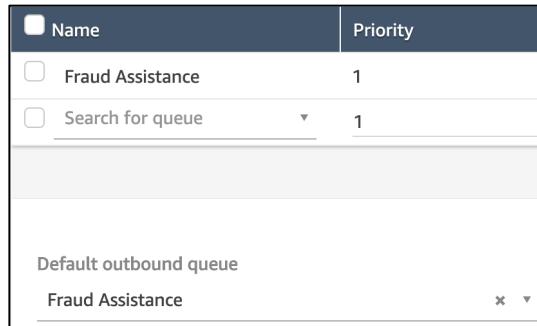
We'll create 3 different routing profiles with the below agent workflows:

- **Generalized** - Agents who are generalists and front-line support.
- **Specialized** - Agents who only work specialized calls.
- **Escalations** - Agents who primarily work escalations.

- 1) Navigate to Users > Routing Profiles

-
- 2) Notice the Basic Routing Profile included. Like with Basic Queue, Amazon Connect includes a Basic Routing Profile to assist getting started. Create your own profile by clicking **Add new profile**.
- 3) Enter a **Name** for the profile and **Description**. Start with **Generalized** and the description below.
- 4) Continue for the remaining two queues.

Queue Name	Routing Profile
General Credit Card	Generalized
Fraud Assistance	Escalations
Technical Assistance	Specialized



Name	Priority
Fraud Assistance	1
Search for queue	1

Default outbound queue

Fraud Assistance

Part 7.4: Users Management

Objective: Understand how to manage Users (Agents, Supervisors, etc.) and Security Profiles

By the end of this section, you would have created some users and assigned security profiles to them. One of the key responsibilities of the CC manager or supervisor would be to manage users, giving them their credentials, and assigning appropriate permissions so they can access the features needed to do their job.

Before you can add users to Amazon Connect, you need the following permissions assigned to your security profile: **Users - Create**.



Type	All	View	Edit	Create	Remove	Enable / Disable	Edit permission
Users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agent hierarchy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Security profiles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agent status	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

By default, the Amazon Connect **Admin** security profile has these permissions.

Add a user individually

- 1) Log in to the Amazon Connect console with an **Admin** account, or an account assigned to a security profile that has permissions to create users.
- 2) Choose Users, User management.
- 3) Choose Add new users.
- 4) Choose Create and set up a new user and then choose Next.
- 5) Enter the name, email address, and password for the user.
- 6) Choose a routing profile and a security profile.
- 7) Choose **Save**. If the Save button isn't active, it means you're logged in with an Amazon Connect account that doesn't have the required security profile permissions.

To fix this issue, log in with an account that is assigned to the Amazon Connect Admin security profile. Or, ask another Admin to help.

Edit users

Users to edit	Email address	
<input type="text"/> @amazon.com	<input type="button" value="reset password"/>	
Routing Profile:	Security Profiles:	Phone Type:
Basic Routing Profile	Agent	Soft phone <input type="checkbox"/> Auto-Accept Call
Agent Hierarchy:	No agent groups available.	After call work (ACW) timeout: 0
<input type="button" value="Save"/> <input type="button" value="Cancel"/>		

Note: Do take note of the Password criteria required - At least 8 characters with an uppercase letter, a lowercase letter, and a number.



Add users in bulk from a .csv file

Use these steps to add several users from a csv file such as an Excel spreadsheet"

- 1) Log in to the Amazon Connect console with an **Admin** account, or an account assigned to a security profile that has permissions to create users.
- 2) Choose **Users, User management**.
- 3) Choose **Add new users**.
- 4) Choose **Upload my users from a template (csv)** and then choose **Next**.
- 5) Choose **Download template**.
- 6) Add your users to the template and upload it to Amazon Connect.

If you get an error message, it usually indicates that one of the required columns is missing information, or there's a typo in one of the cells.

- We recommend checking the format of the phone number as a starting point in your investigation.
- If you get an error message that **Security profile is not found**, check whether there's a typo in one of the cells in the **security_profile_name_1** column.
- Update the .csv file and try uploading it again.



Errors detected in uploaded file.

Fix any errors, save the file, and go back to upload it again.

- Line number: 1 Invalid routing profile:
- Line number: 2 Invalid routing profile:

Part 7.5: Hours of Operation

Objective: Understand how to set the hours of operation and time zone for a queue

The first thing you need to do when you set up a queue is to specify the hours of operation and time zone. The hours may be referenced in contact flows. For example, when routing contacts to agents, you might use the [Check hours of operation](#) block first, and then route the contact to the appropriate queue.

To set the hours of operation and time zone for a queue

- 1) Choose **Routing, Hours of operation**.



-
- 2) To create a template, choose **Add new hours** and enter a name and a description.
 - 3) For **Time zone**, select a value.
 - 4) For **Add new**, set new hours.
 - 5) Choose **Save**.
 - 6) Now you can specify these the hours of operation when you [create a queue](#), and check them in the [Check hours of operation](#) block.

How to specify midnight

To specify midnight, enter 12:00AM.

For example, if you want to set your hours to 10:00AM to midnight, you would enter: 10:00AM to 12:00AM. Your call center would be open for 14 hours. Here's the math:

- 10:00AM-12:00PM = 2 hours
- 12:00PM-12:00AM = 12 hours
- Total = 14 hours

Examples

Schedule for 24x7

Day	Start	End
<input type="checkbox"/> Sunday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Monday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Tuesday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Wednesday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Thursday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Friday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Saturday	12 : 00 AM	12 : 00 AM

Schedule for Monday to Friday 9:00 AM to 5:00 PM

Remove Sunday and Saturday from the schedule.

Time zone: America/New_York Add new Remove Edit

Day	Start	End
<input checked="" type="checkbox"/> Sunday	12 : 00 AM	12 : 00 AM
<input type="checkbox"/> Monday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Tuesday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Wednesday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Thursday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Friday	09 : 00 AM	05 : 00 PM
<input checked="" type="checkbox"/> Saturday	12 : 00 AM	12 : 00 AM

The final schedule looks like this:

Time zone: America/New_York Add new Remove Edit

Day	Start	End
<input type="checkbox"/> Monday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Tuesday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Wednesday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Thursday	09 : 00 AM	05 : 00 PM
<input type="checkbox"/> Friday	09 : 00 AM	05 : 00 PM

Add lunch and other breaks

If your entire contact center were to close for lunch from 12-1, for example, then you'd enter hours to specify that, as in the following image:

Day	Start	End
<input type="checkbox"/> Monday	09 : 00 AM	12 : 00 PM
<input type="checkbox"/> Monday	01 : 00 PM	05 : 00 PM

In most contact centers breaks are staggered. While some agents are at lunch, for example, others are still available to handle contacts. Instead of specifying this in the hours of operation, you [add custom agent statuses](#) that appear in the agent's Contact Control Panel (CCP).

For example, you might create a custom status named **Lunch**. When the agent goes to lunch, they change their status in the CCP from **Available** to **Lunch**. During this time, no contacts are routed to them. When they return from lunch and are ready to take contacts again, they change their status back to **Available**.

Supervisors can change an agent's status using the real-time metrics report.



What happens during daylight savings time?

Amazon Connect uses the time zone to determine whether daylight savings time is in effect for the queues, and adjusts automatically. When a contact comes in, Amazon Connect looks at the hours and time zone to determine whether the contact can be routed to the given queue.

Use the Check Hours of Operation block

At the start of your contact flows, you can use the [Check hours of operation](#) block to determine whether your contact center is open, and to branch accordingly.

Part 7.6: Contact Flow

Objective: Understand contact flow creation and working with data

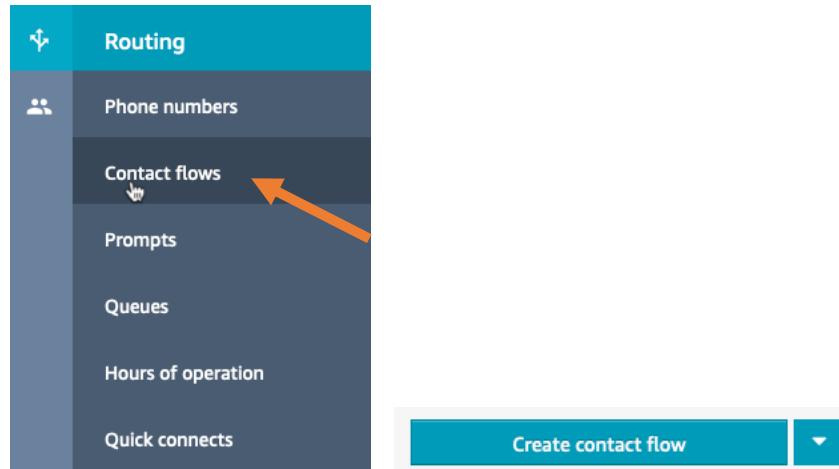
By the end of this section, you would have imported an existing contact flow file and gone through step by step this contact flow and understand how to leverage a lambda to data dip whenever needed.

In this section, we'll be looking at a customer contact flow that will check for Business Operation Hours to determine whether it's operating hours as well as lookup a holiday table and playback a custom prompt /announcement if it's a holiday. This applies for special events or any ad-hoc announcements which the business want to add/change/delete on the fly in a very dynamic manner. This serves as the basis of getting started with contact flow design in Amazon Connect.

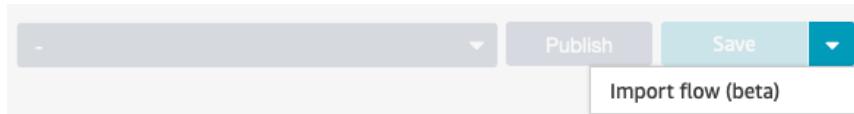
For this lab, we are going to import an existing contact flow and start from there.

- 1) First, we need to get the contact flow file. From the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named "contact flows", within it you will see a file named "**myFlow1**", this is the contact flow file which you will be importing.

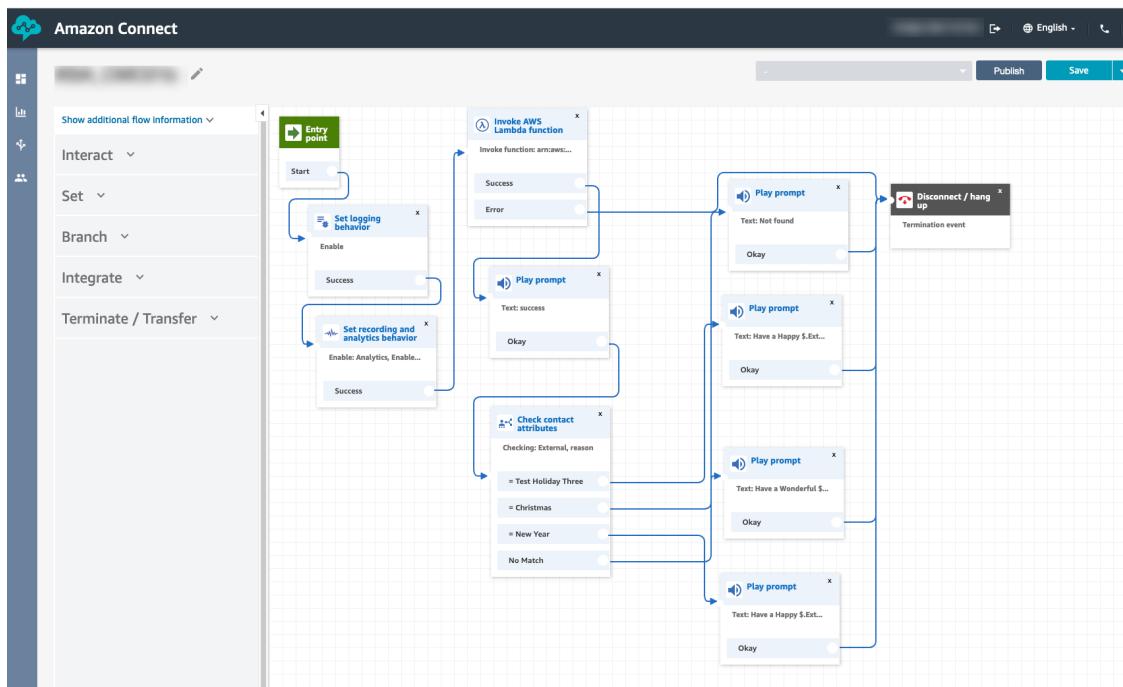
- 2) From Amazon Connect management console, navigate to **Routing > Contact flows** section.
And click on **Create contact flow**.



- 3) From the down arrow, click on Import flow (beta).



- 4) Click on that to select the contact flow which you have downloaded earlier.
- 5) You can choose to rename this contact flow to any name you require.
Your contact flow should look like the following (or a more organized version 😊).



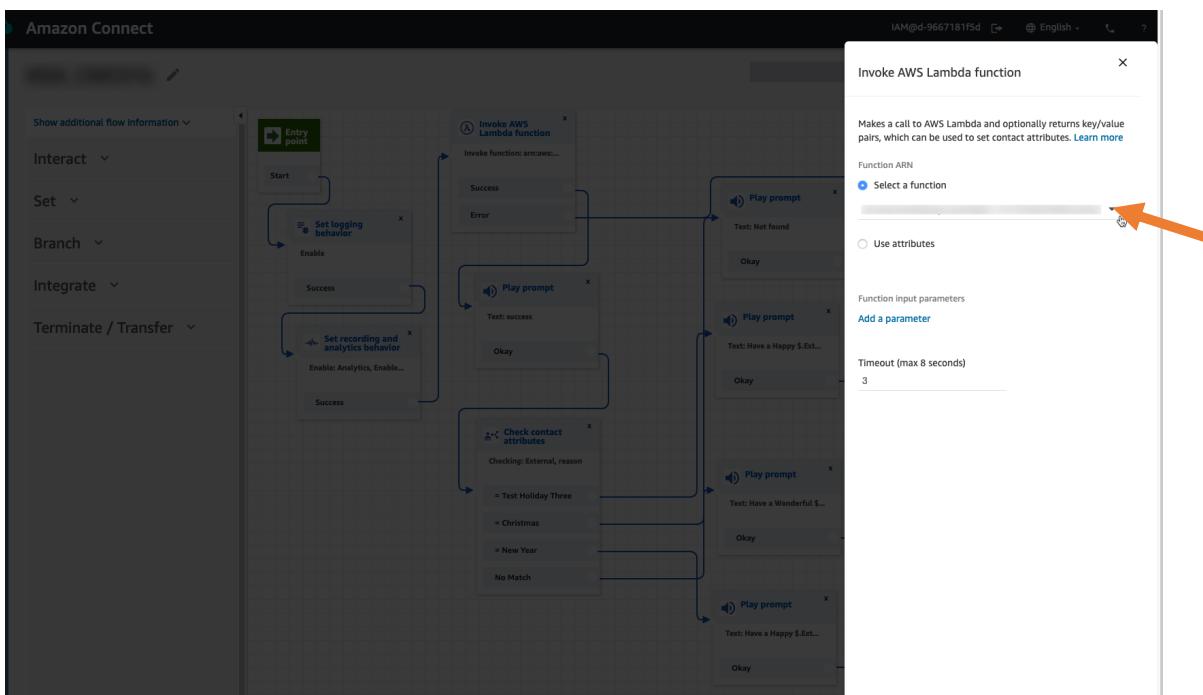
- 6) From the flow, the first block used after the Entry point is the Set logging behavior block which is to enable the contact flow log and you can download/view this from the respective S3 bucket.

After that, it's the Set recording and analytics behavior block which you can use it to turn on voice recording for customer, agent or both customer and agent.

Do take note that if you want to enable Contact Lens for Amazon Connect, you can do it from this block too, .. just check a box to enable it, it's as simple as that!

<Refer to [AWS Summit Online ASEAN 2020 Video - The Art of the Possible: Amazon Connect Customer Service Experience @ 17:50 for Amazon Connect demo](#)>

- 7) Next, click on the **Invoke Amazon Lambda function** block, and select the earlier lambda which you have created – holidayCalendarRead.



- 8) The next block which is a **Play Prompt** is just used for testing purpose.

- 9) The next block is a **Check contact attributes** block.

Contact attributes is one of the most important concept which you will need to know within Amazon Connect!

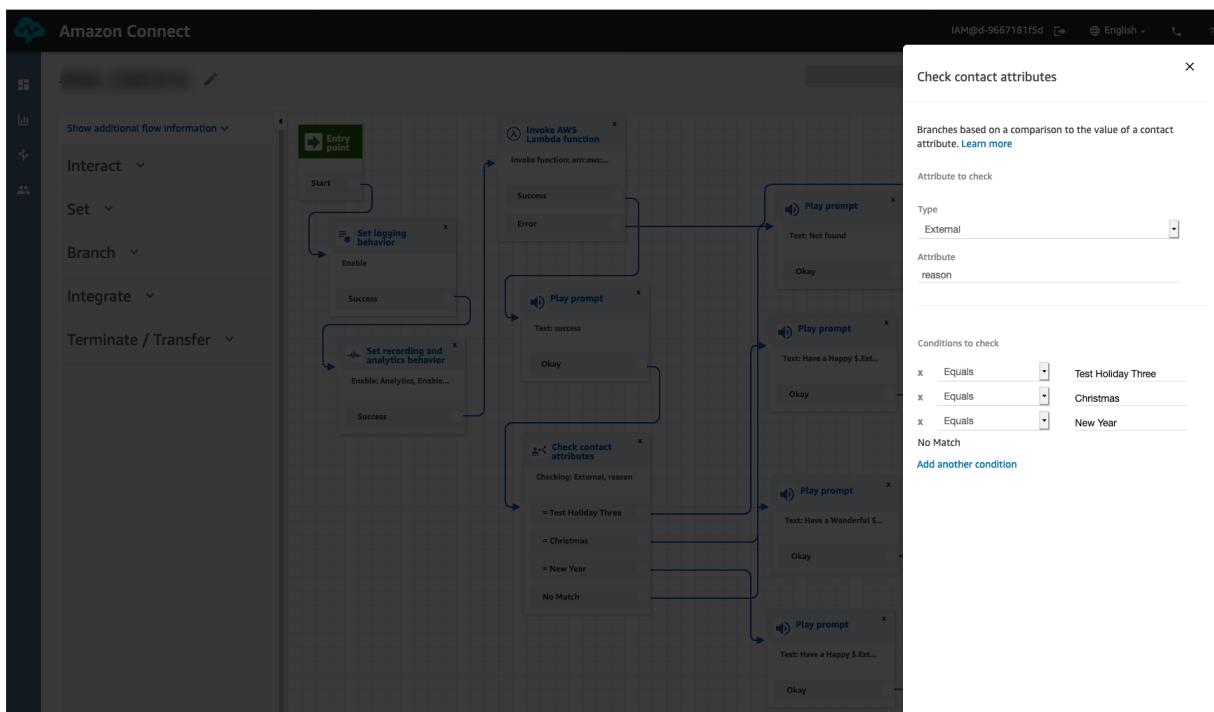
Please look through following sections of the Amazon Connect online documentation guide:
<https://docs.aws.amazon.com/connect/latest/adminguide/connect-contact-attributes.html>

In Amazon Connect, a contact is an interaction with a customer in your contact centre. The interaction can be a voice phone call, a chat, or an automated interaction using an Amazon Lex bot. Contact attributes in Amazon Connect refer to key-value pairs that contain data about a contact.

Using contact attributes, you can customize and personalize the experience customers have when they interact with your contact centre. Contact attributes let you store customer input or data about a customer, and then use it later in a contact flow. You can also check the values of contact attributes and use a condition to determine the branching behaviour of the contact flow based on the value.

Contact attributes let you pass data between Amazon Connect and other services, such as Amazon Lex and AWS Lambda. Contact attributes can be both set and consumed by each service.

For e.g., you could use a Lambda function to look up customer information, such as their name or order number, and use contact attributes to store the values returned to Amazon Connect. You could then reference those attributes to include the customer's name in messages using Amazon Polly Text-to-Speech, or store their order number so they do not have to enter it again.

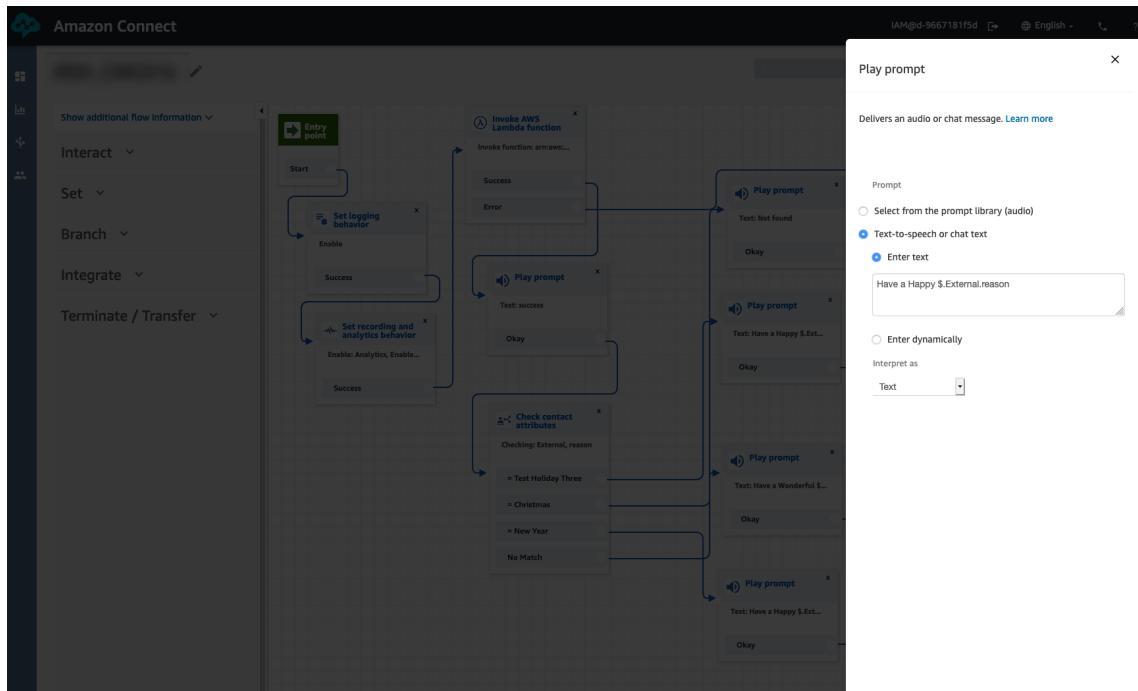


If you look at the lambda function code, you will see that it will return a JSON result consisting of multiple key-value pairs, we will have to check the output of the **reason** key-value.

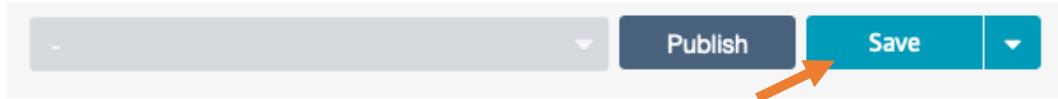
The lambda will check the current date/time stamp against the DynamoDB table to see if it falls within the corresponding holidays/events date/time intervals, and if it does it will return that item(s).

- 10) After that, we can use the Play prompt block to playback the message or prompt which you want the caller to hear.

For e.g., the prompt below will playback an announcement “Have a Happy” followed by whatever you have input in the **reason** value within the DynamoDB table using an external contact attribute defined by `$.External.<attribute name>`. This is using Amazon Polly Text-to-Speech engine.



11) If you want to save your contact flow, just click on the **Save** button at the top right-hand corner.

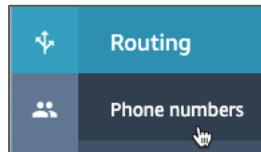


This will save your changes but will not publish (activate) the flow.

If you need to activate the flow, then click on **Publish** which will both save and activate this flow. If there are errors, check that all nodes are connected.

Note: You can import and export flows for sharing and archival purposes.

12) We need to now change our claimed phone number to use this contact flow. Navigate to **Routing > Phone Numbers**. Click onto the number claimed.



13) Under **Contact flow/ IVR**, change this setting to the name of our Contact flow.

14) Click **Save**.

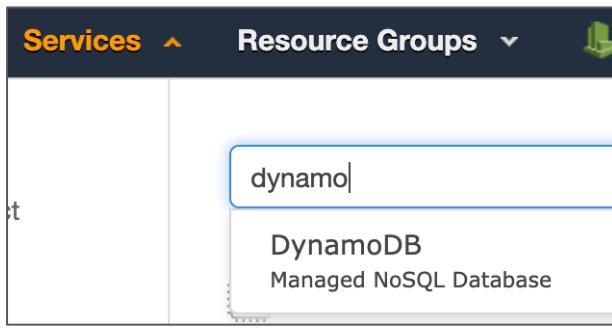
- 15) Dial this number and listen to the interaction. Check that the flow runs as what you have set it up to be.
- 16) Congratulations on your first successful contact flow!

Part 8: Personalization

Objective: Incorporating personalization into your contact flow

By the end of this part, you would have a greater understanding on how we are simulating data that exists in your existing backend system or any CRM system that you may have. We will be leveraging another DynamoDB table as a data dip point later on in this part.

- 1) Go back into your management console. Search for **DynamoDB**. Click into it.

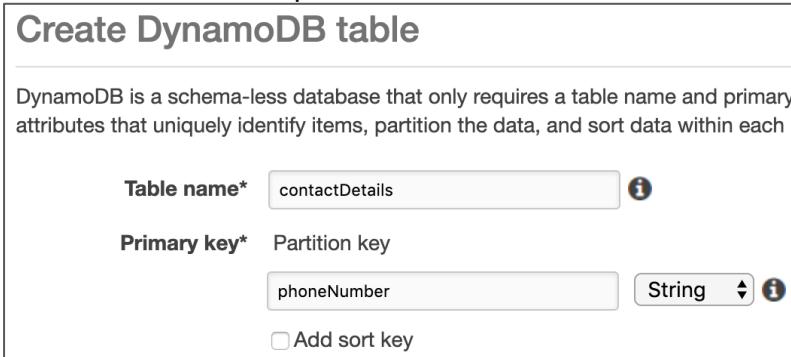


- 2) Click on

Create table

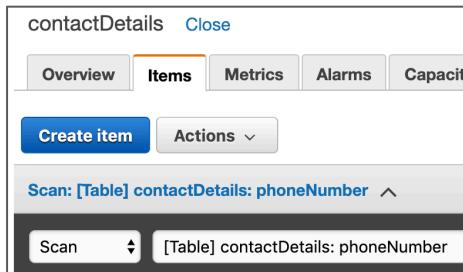
- 3) Enter in **contactDetails** for the table name, and **phoneNumber** for the primary key. Click on the blue Create table at the bottom of the page.

***NOTE:** Names are cap sensitive



Create DynamoDB table	
DynamoDB is a schema-less database that only requires a table name and primary attributes that uniquely identify items, partition the data, and sort data within each partition.	
Table name*	contactDetails
Primary key*	Partition key
phoneNumber	
String	
<input type="checkbox"/> Add sort key	

- 4) Create new line items in the table. Click on **Create item**.



- 5) Enter in your **phoneNumber**, **firstName** and **lastName**. To add more fields, click on the plus sign, Append and String.

Again, do take note that **field names are cap-sensitive**. Input the country code and phone number without spacing.

Your table should look like the table on the right.

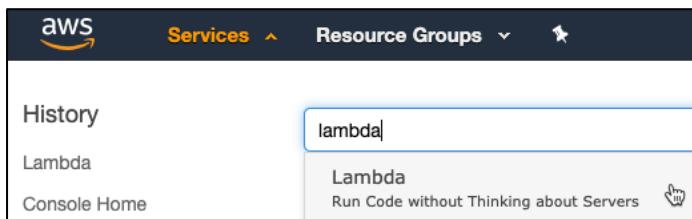
phoneNumber	firstName	lastName
+658XXXXXX	Lili	Chan
+659XXXXXX	Shannon	Lim

- 6) All set for your table! You can come back into DynamoDB to add more lines or edit this table.

Part 8.1: Lambda used for DDB Read

Pulling in custom attributes from DynamoDB table by using lambda. Lambda is a service that lets you run codes without a need to provision or manage any servers. This will be the main way for you to run codes, data dip etc.

- 7) Go to your Amazon management console. Search for **Lambda**.



- 8) Click on the orange button on the right **Create function**.

- 9) Enter in function name of **DDB_Read**. Or any name you prefer. Leave the runtime as **Node.js 10.x. or above**.

- 10) (Note: You can also choose Python if you prefer to code in this language).



- 11) Click on the arrow for the dropdown of Choose or create an execution role. Select Use an existing role.
- 12) Select **lambdaAdmin** in the IAM section above which you have previously created.
- 13) Click on Create function.
- 14) Go to the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named “contact samples”, open the [sample_dynamo_get_node.js](#) file using your favorite code editor tool. (Note: For Python, the file would be [sample_dynamo_get_python.py](#))
- 15) Copy and paste the codes from entire file.

16) Save the lambda function.

The screenshot shows the AWS Lambda function editor for a function named "DDB_Read". The "Function code" tab is selected. The code entry type is set to "Edit code inline". The runtime is "Node.js 10.x" and the handler is "index.handler". The code itself is a Node.js script named "index.js" containing logic to read data from a DynamoDB table using the AWS SDK. The code includes imports for AWS, a try-catch block for handling database errors, and a Lambda function logic block that logs the event and constructs a response object.

```
// Global Variables
var AWS = require("aws-sdk");
var docClient = new AWS.DynamoDB.DocumentClient();

// API calls:
async function readDDB(paramsQuery) {
  try {
    const getData = await docClient.query(paramsQuery).promise();
    return getData;
  } catch (err) {
    console.log('error' + err);
    throw err;
  }
}

// Lambda function logic:
exports.handler = async(event, context) => {
  console.log("Loading event" + JSON.stringify(event));
  const paramsQuery = {
    TableName: event.Details.Parameters.TableName,
    KeyConditionExpression: event.Details.Parameters.Key + " = :var1",
    ExpressionAttributeValues: { ":var1": event.Details.Parameters.KeyValue }
  };
  const getData = await readDDB(paramsQuery);
  if (getData.Count == 0) {
    let result = {
      "recordCount": "0"
    };
  }
};
```

- 17) Go on to management console, select **Amazon Connect**
- 18) Click on Contact Flows

The screenshot shows the Amazon Connect management console for a workspace named "shannlim-workbook". The left sidebar has a navigation tree with several categories: Overview, Telephony, Data storage, Data streaming, Application integration, and Contact flows. The "Contact flows" item is highlighted with a blue border, indicating it is the active or selected menu item.



- 19) Scroll down and search for **AWS Lambda**. Add in the **DDB_Read** function you have just created. And click **+ Add Lambda Function**.

The screenshot shows the AWS Lambda interface. At the top, there is a heading "AWS Lambda". Below it, a note states: "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." A note below says: "Note: By adding Lambda functions, you are granting Amazon Connect permission to invoke them [Create a new Lambda function](#)". A dropdown menu shows "Function" and "DDB_Read". A button labeled "+ Add Lambda Function" is highlighted with a mouse cursor.

Part 8.2: Create a contact flow to show Personalization

By the end of this part, you would have created another contact flow and understand how to leverage lambda to data dip whenever needed.

In this section, we'll create a basic customer contact flow that can be used for a phone number claimed on Amazon Connect. This flow will greet a caller, prompt for simple DTMF entry of digits, and end.

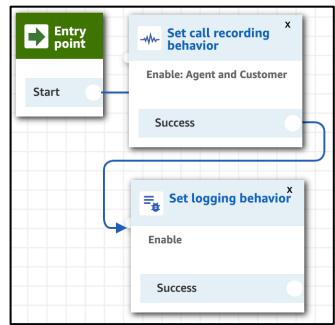
1) Go to Routing > **Contact Flow**

2) Click Create Contact Flow

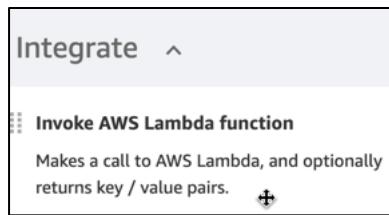
3) Enter a Name of **Entry – All Calls** or something similar. Enter a description of your choice and click **Save**.

The screenshot shows the "Create Contact Flow" interface. The title bar says "Entry - All Calls" with a pencil icon. Below it, a dropdown shows "Latest: Saved". There is a "Publish" button and a "Save" button highlighted with an orange arrow. A blue arrow points to the "Show additional flow information" link. The main area shows a canvas with a single green entry point block.

- 4) Notice the green **Entry Point** box now has a line added. Start adding blocks to build the contact flow from the left and connect the lines to blocks.
- 5) Expand the **Interact** group and click and drag the **Set call recording behavior** and **Set logging behavior** blocks to the canvas.



- 6) On the left column of options, drag out **Invoke AWS Lambda function**.



- 7) Select the lambda function you have created **DDB_Read**.
 8) Click on **Add a parameter**. Add in the destination key and value as below.



Invoke AWS Lambda function

X

Makes a call to AWS Lambda and optionally returns key/value pairs, which can be used to set contact attributes. [Learn more](#)

Function ARN

- Select a function

- Use attributes

Function input parameters

X

- Use text

Destination key

tableName 

Value

<Insert DDBTable Name here>

- Use attribute

- Use text

X

Destination key

key

Value

phoneNumber

- Use attribute

- Use text

- Use attribute

Destination key

keyValue

Type

System

Attribute

Customer Number

- 9) You should have 3 parameters as seen above. Note that the destination key and value is case sensitive.

TableName is the name of the DynamoDB table you have created in previous part above.
phoneNumber is the key name of your DynamoDB table.

And we are using the system attribute of incoming call number as the search key.

- 10) Drag over the **Get customer input** box.

- 11) Click on the box on the canvas and a sidebar of options will appear for this block. Select **Text to Speech (Ad hoc)**. Enter a greeting text of as follows.

Hello \$.External.firstName, how can we help you today? For general enquiries, please press 1. If you believe you are a victim of fraud, please press 2. For technical assistance, please press 3. For all other enquiries, please press 0.

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

Select from the prompt library (audio)
 Text to speech (Ad hoc)
 Enter text

Hello \$.External.firstName, how can we help you today?
 For general enquiries, please press 1. If you believe you
 are a victim of fraud, please press 2. For technical
 assistance, please press 3. For all other enquiries, please
 press 0.

Enter dynamically
 Interpret as
 Text

- 12) Do not close this box—scroll down within the block until you see the **DTMF** header and the link **Add another condition**. Click this link four times and enter **1** in the first line, **2** in the second, **3** in the third, **0** in the fourth. Click **Save**.

Set timeout (Minimum one second)
 5 seconds

Option:
 x 1

Option:
 x 2

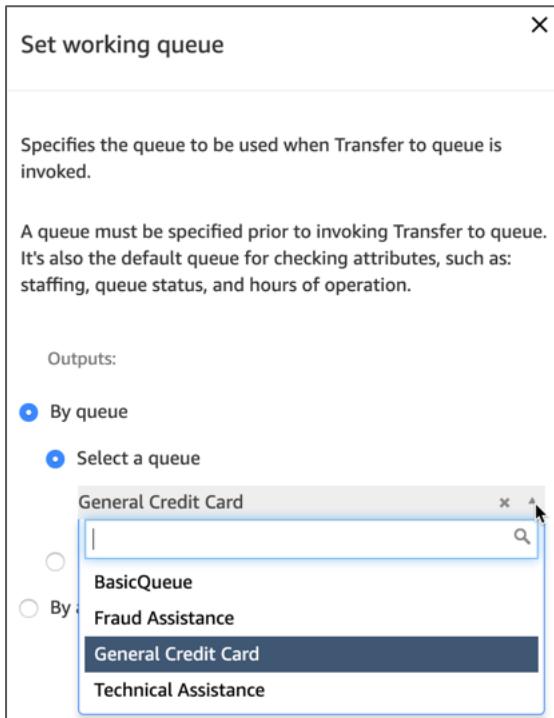
Option:
 x 3

Option:
 x 0

Take note of the **\$.External.attribute** section.

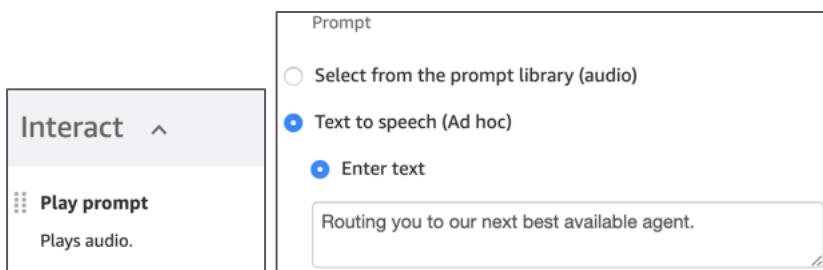
For more information, read here: <https://docs.aws.amazon.com/connect/latest/adminguide/using-contact-attributes.html>

- 13) On the left-hand side options, under the Set options. Drag and pull out the **Set working queue** block. Go ahead and set the block to a **General Credit Card** queue.



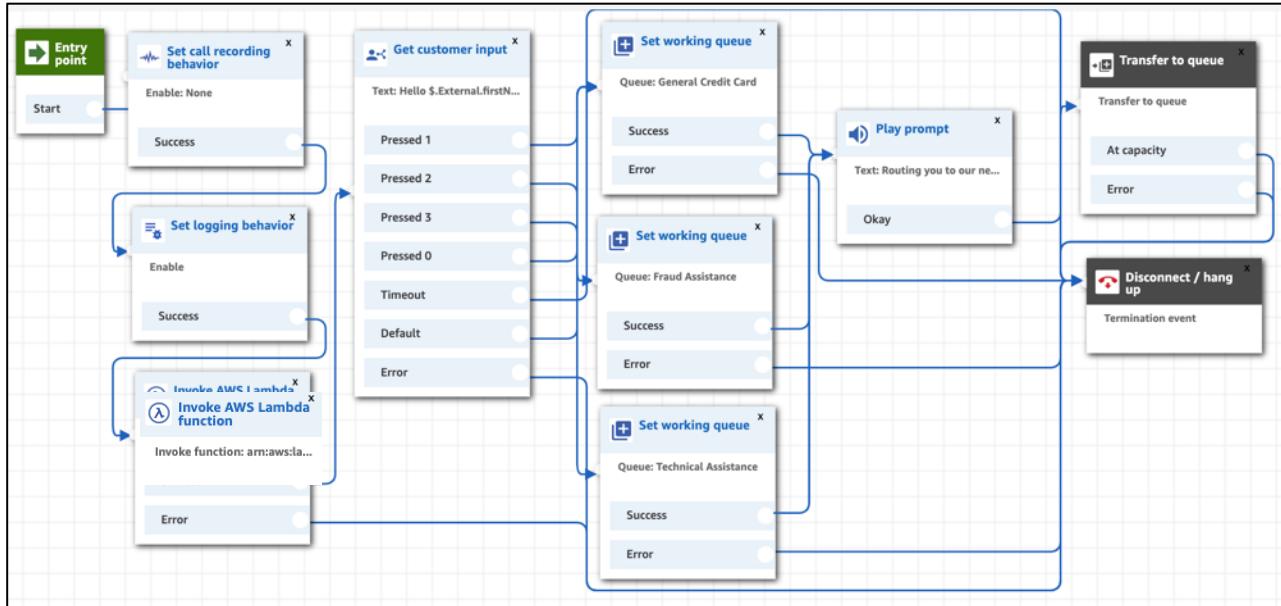
- 14) Repeat step 13 by pulling out two other **Set working queue** block and setting them respectively for **Fraud Assistance** and **Technical Assistance**.
- 15) Connect option 1 to the General Credit Card block.
 Connect option 2 to the Fraud Assistance block.
 Connect option 3 to the Technical Assistance block.
- 16) Drag the Play prompt block and enter in the following text.

Routing you to our next best available agent.



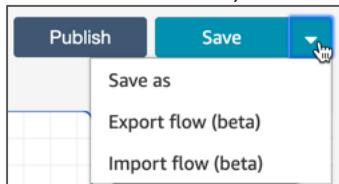
- 17) Connect the 3 **Set working queue** blocks to the **Play prompt** block you have just created.

- 18) Pull out a **Transfer to queue** block and have all three successful option from **Set working queue** block go to this **Transfer to queue** block.
- 19) For all other failure options or time out options, drag out and connect them to the **Disconnect / hang up** block.
- 20) Your contact flow should look like below.



- 21) We now need to publish our contact flow so it becomes live. Click the **drop-down arrow** and select **Save**. Select **Publish** if there are no errors.

If there are errors, check that all nodes are connected.



Note: You can import and export flows for sharing and archival purposes.

- 22) We need to now change our claimed phone number to use this contact flow. Navigate to **Routing > Phone Numbers**. Click onto the number claimed.



- 23) Under **Contact flow/ IVR**, change this setting to the name of our Contact flow (**Entry – All Calls**)



24) Click **Save**.

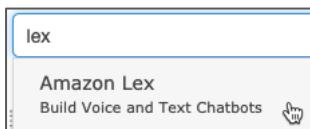
25) Dial this number and listen to the interaction. Check that flow is as you have set up. Notice how the call is put into queue for an agent. Since we have no agents working, it will remain in queue until you disconnect.

26) Congratulations on another successful contact flow!

Part 9: Creating an Amazon Lex Bot

Objective: Creating your first Lex bot

1) Go to your Amazon management console. Search for **Amazon Lex** (which is available from Singapore region).



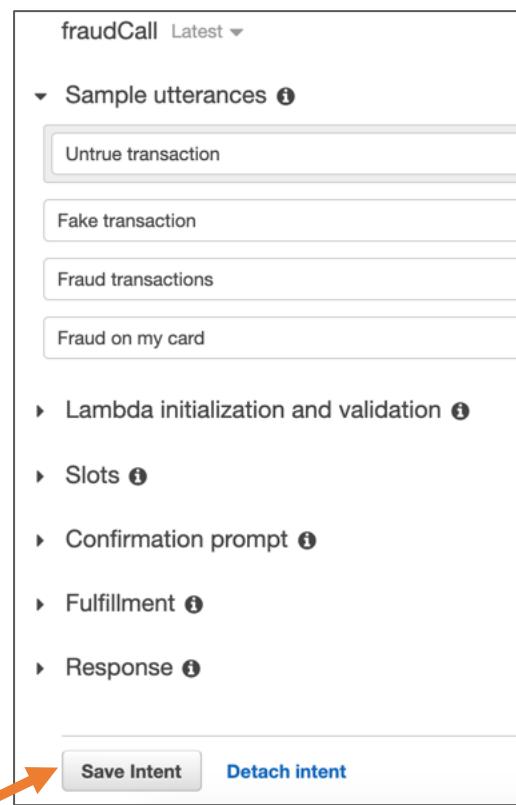
2) Go ahead and click on **Get Started**.

3) Select **Custom bot**. Enter in your bot's name. I have used the name **financeBot** here in this example. Use any output voice you prefer and a session timeout. Select **No** for COPPA. Then click on the Create button.

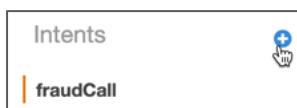
The screenshot shows the 'CREATE YOUR OWN' section of the Amazon Lex 'Get Started' page. It includes the following fields:

- Custom bot** (selected)
- TRY A SAMPLE** button
- Bot name**: financeBot
- Language**: English (US)
- Output voice**: Salli
- Type text here to hear a sample** input field
- Session timeout**: 10 min
- IAM role**: AWSLambdaRoleForLexBots (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

- 4) Click on the + Create Intent button to create your first intent. Select Create Intent.
- 5) We will be creating 3 intents in total. We want our caller to say what the call is about instead of using DTMF. This will replace the DTMP menu option in the contact flow we have created earlier. Enter in an intent name of **fraudCall**.
- 6) Under sample utterance. Enter in what caller would say if their intention is to say they are calling for Fraud. Below is a sample of what you can enter in. You can enter in more if you have more utterances!

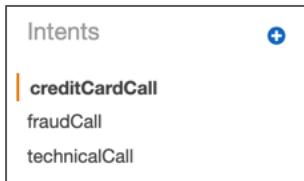


- 7) Do remember to **Save the Intent**.
- 8) Go ahead and create the other two intents.



- 9) Name the intent **technicalCall**. go ahead and do the same as step 6 above. Enter in what the caller would usually say if they are calling for technical help. Save it.

-
- 10) For the last intent, name it **creditCardCall**. And do the same as step 9.

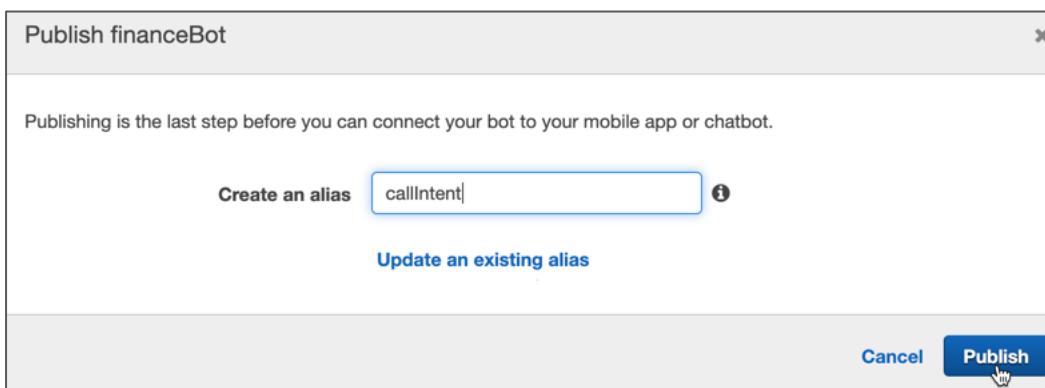


- 11) On the top right-hand side of the page. You can see both the Build and Public buttons.

Go ahead to **Build** the bot.



- 12) Then **Publish** the bot. It will ask for an alias. Enter in **callIntent**. Go ahead and click on the **Publish** button.



- 13) It will give you a successful message and you're all set!

Part 9.1: Leveraging your Amazon Lex Bot

Objective: Understanding and working with Amazon Lex, our voice bot

By the end of this section, you would have edited the DTMF contact flow you have created and leveraged on your Amazon Lex Bot.

Contact flow security keys

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

Add key Remove

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: [dropdown] Bot: financeBot (+ Add Lex Bot)

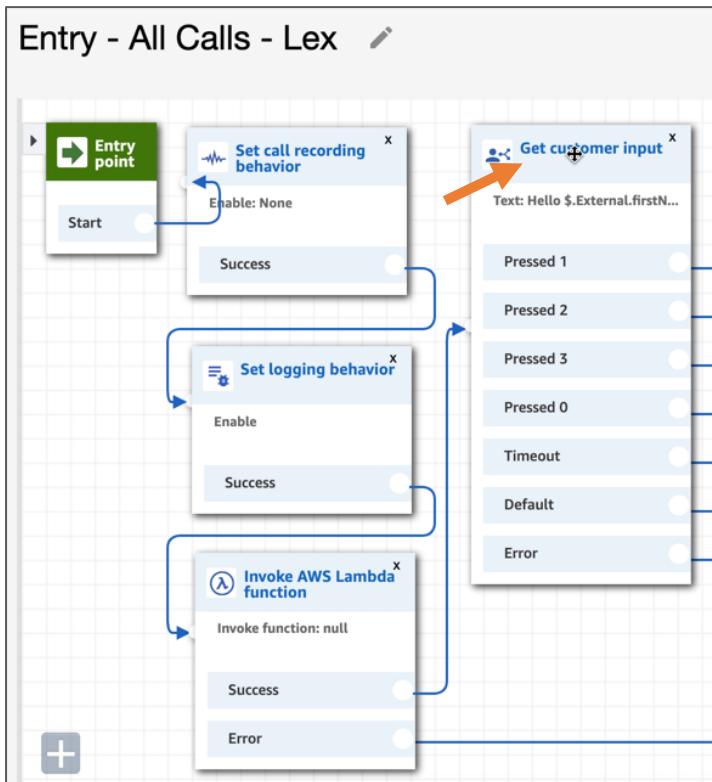
Lex bots

- 1) Go in to your management console. Click in to **Contact flows** as shown above. Add in the bot you have just published. And importantly, click on **+ Add Lex Bot**.
- 2) Go into your existing contact flow. Save this flow as a different name **Entry – All Calls – Lex** or some other name to differentiate it from your DTMF contact flow.
- 3) Click into your existing **Invoke AWS lambda** block. You would have to re-enter all the variables as you have done earlier.

- 4) In your first **Get customer input** block, click into it and change the text to:

Hello \$.External.firstName, how can we help you today? For general credit card enquiries, please say credit card. If you believe you are a victim of fraud, please say fraud transactions. For

technical assistance, please say technical help. For all other enquiries, please hold.



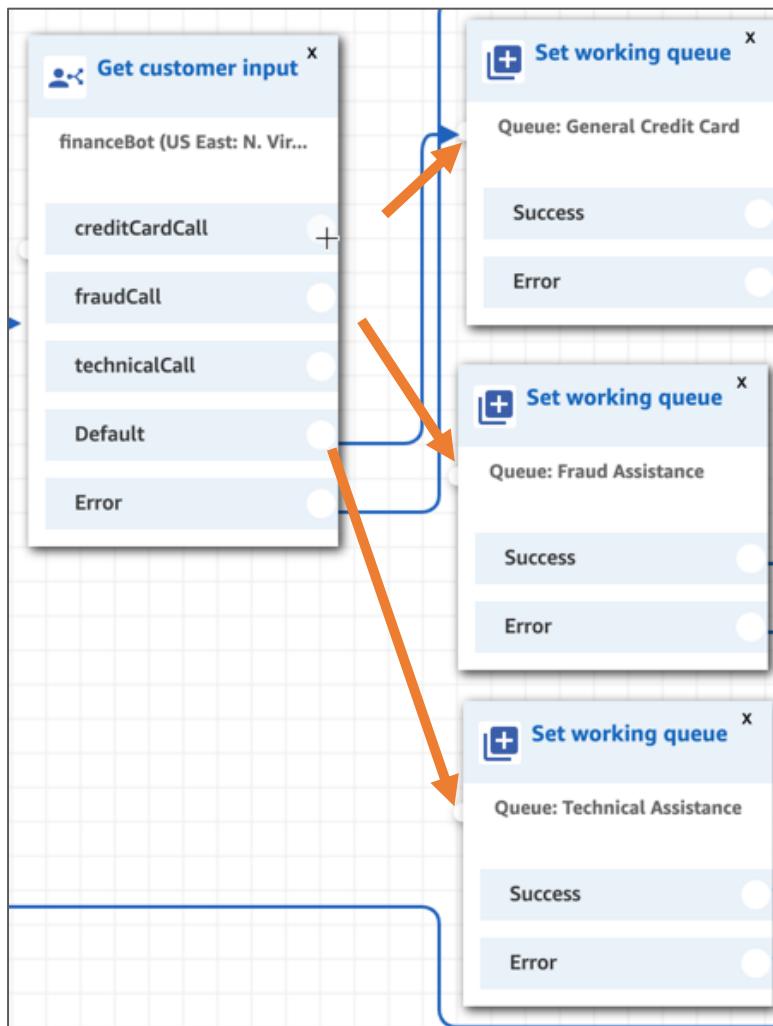
- 5) Scroll down the edit section of the block. You will see **Amazon Lex** as another tab beside DTMF. Click into it. Your bot should appear in the picklist. Select it.

You gave the intent an alias name earlier in the Lex section. Enter in **callIntent** if you did not change the guide's recommended alias.

Enter in the other names of the intents you have created earlier in Lex (**do note the names are case-sensitive**). Go ahead and save it.

Intents
X creditCardCall
X fraudCall
X technicalCall

- 6) Go ahead and connect the nodes to the right blocks as it was for DTMF.



7) Publish it. And test it out!



Part 10: One-Time Pin Authentication using Amazon Pinpoint

Amazon Pinpoint is a flexible and scalable outbound and inbound marketing communications service. You can connect with customers over channels like email, SMS, push, or voice.

Transactional messages

Send immediate, trigger-based customer communications across channels directly from your application, such as purchase confirmations, one-time passwords, or shipping notifications. With SMS, you can receive messages back from your customers.

Go to your Amazon management console. Search for **Amazon Pinpoint** (which is available from Singapore region).

Click on “Create a project” at the right hand corner.

Give a name for your Project for e.g., myOneTimePinxxxxx.



Create a project

X

Project name

myOneTimePin

Project names can include up to 64 alphanumeric characters.

Cancel

Create

Under SMS – click on “Configure”.

You've created myOneTimePin. Now add features to your project.

Pinpoint > All projects > myOneTimePin > Configure features

Configure features

Choose a feature to add to your project. You can add more features later.

Project features

Messaging channels and response metrics

- Email: Send personalized email messages to your customers. [Info](#) [Configure](#)
- SMS: Send SMS text messages from shared or reserved phone numbers. [Info](#) [Configure](#)
- Push notifications: Send push notifications to users of your mobile apps. [Info](#) [Configure](#)

Application analytics

- Mobile app analytics: Track usage metrics for mobile applications. [Info](#) [Configure](#)
- Web app analytics: Track usage metrics for web-based applications. [Info](#) [Configure](#)

[Skip this step](#)

Under “Set up SMS” section, click on the “Enable the SMS channel for this project”.

Select “Transactional” message type.

You can type in a Sender ID which will appear on the recipient's device upon receiving the SMS message.



Pinpoint > All projects > myOneTimePin > Configure features > Set up SMS

Set up SMS

General settings

Enable the SMS channel for this project

Account-level settings

The settings in this section apply to all SMS messages that you send from your AWS account, including messages that you send using other AWS services.

Default message type
The type of messages you plan to send from this project. [Info](#)

Transactional
Time-sensitive content, such as one-time passcodes.

Promotional
Non-critical content, such as marketing messages.

Account spending limit
The maximum amount of money, in USD, that you want to spend sending SMS messages each month. The limit for accounts in the sandbox is 1 (\$1.00). [Info](#)

1

The spend limit that you specify can't include decimals. The minimum value is 0, and the maximum value is 1.

(Optional) Account sender ID
The identity that appears on recipients' devices when they receive this message. Support varies by country or region. [Info](#)

PinAlert

Your sender ID can contain up to 11 alphanumeric characters. It has to contain at least one letter, and it can't consist only of numbers. Some countries and regions may have additional restrictions.

Advanced configurations - optional

Cancel **Save changes**

Click on “Save changes”.

From the Pinpoint console, go to “All Projects”, take note of the Project ID as you will be using this for your *OneTimePin* lambda later on.

Pinpoint

All projects

All projects (3)

Search

Project name Project ID Creation date

Project name	Project ID	Creation date
myOneTimePin	df212e2e78834c2390b833e10722e4e9	May 9th 2021, 04:24 PM, UTC

Create a project

Go back to your SMS project’s settings, and look at the Long Code number being assigned.



SMS

Quick start

Choose one of the options below to request a number or to register your company and campaign to use 10DLC.

Request phone numbers

Choose a long code or short code and assign that number to your AWS account. You can request a long code through [self-service](#) or [file a support request](#) for a short code.

Register your company and campaign purpose

Register your company and campaign to take advantage of 10DLC. This is only required if you're planning on using SMS with U.S. recipients. [Learn more about 10DLC](#)

- Register your company
- Create a 10DLC campaign

SMS settings

Channel status: Enabled

Default message type: Transactional

Account spending limit: \$1.00

(Optional) Account sender ID: PinAlert

Request a spending limit increase

Request a sender ID

Phone numbers (1)

Number	Type	Registered keyword	Country	SMS	Created Date	Status
+1 314-326-4875	Long code	-	US	Disabled	April 8th 2021, 05:49 AM, UTC	Active

Endpoints

Developers: Define endpoints to contact

An endpoint is a destination that you can send messages to such as a user's mobile device, email address, or phone number. Before you can send email, you have to add endpoints to your project. You can add endpoints by using the API or an AWS SDK.

- ▶ Use AWS Pinpoint SDK
- ▶ Use AWS CLI
- ▶ Import and Export from the Console

Now to test the SMS messaging, go to “Test Messaging” at the right side menu.

Enter the E164 mobile number (+659xxxxxxxx) which you want to send the SMS message to.

The Message Type is “Transactional”.

Type in a short test message and click on “Send Message”.



The screenshot shows the AWS Pinpoint console under the 'myOneTimePin' project. In the left sidebar, 'Test messaging' is selected. The main area is titled 'Test messaging'. Under 'Channel', 'SMS' is selected. In 'Destinations', a phone number '+65' is entered. The 'Message' section contains the text 'Test message from Liji Chan'. At the bottom right, there is an orange 'Send message' button.

You should see that a SMS message will be sent to the mobile number which you have indicated.

A green banner at the top of the screen displays the message 'Message sent Successfully sent SMS message.' Below the banner, the 'Test messaging' interface is visible, showing the same setup as the previous screenshot.

Next, you will be creating a new Lambda for generating a one-time pin as a SMS to be used in your Amazon Connect contact flow.

- 1) Go to your Amazon management console. Search for **Lambda**.

The screenshot shows the AWS Lambda service page. A search bar at the top contains the text 'lambda'. Below the search bar, a card for a Lambda function is displayed, showing the name 'Lambda' and the description 'Run Code without Thinking about Servers'.

- 2) Click on the orange button on the right **Create function**.



- 3) Enter in function name for e.g., OneTimePin, or any name you prefer.
Leave the runtime as **Node.js 12.x** or above.
- 4) Click on the arrow for the dropdown of Choose or create an execution role.
Select Use an existing role.
- 5) Select **lambdaAdmin** in the IAM section above which you have previously created.
- 6) Click on **Create function** button.
- 7) From the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named “code samples”, click into it and locate the **oneTimePin.js** file.
- 8) Copy and paste all the codes in there into the new lambda function as shown below.
- 9) Save the lambda function for e.g., **oneTimePin**

The screenshot shows the AWS Lambda console interface. At the top, the navigation bar includes 'Services' (dropdown), 'Search for services, features, marketplace products, and docs' (input field), and '[Option+S]' (button). Below the search bar, the path 'Lambda > Functions > OneTimePin' is displayed. The main area shows the 'OneTimePin' function details:

- Function overview**: Shows the function icon (Lambda), name 'OneTimePin', and 0 layers.
- Description**: Empty.
- Last modified**: 7 months ago.
- Function ARN**: arn:aws:lambda:ap-southeast-1:841392346146:function:OneTimePin

Below the overview, there are tabs for 'Throttle', 'Copy ARN', and 'Actions'. The 'Code' tab is selected, showing the code editor. The code editor has tabs for 'Code' (selected), 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The code editor window shows the 'index.js' file content:

```
const AWS = require('aws-sdk');
var pinpoint = new AWS.Pinpoint({ region: process.env.region });

exports.handler = (event, context, EmergencyDB) => {
  var phoneNumber = event.Details.Parameters.concernedNumber;
  //Make the PIN
  function makeId() {
    var text = "";
    var possible = "0123456789"; //The possible digits
    for (var i = 0; i < process.env.pinLength; i++) {
      text += possible.charAt(Math.floor(Math.random() * possible.length));
    }
    return text;
  }
  var PIN = makeId();

  //Check the number is a mobile
  var paramsMobile = {
    NumberValidateRequest: {
      //IsoCountryCode: 'STRING_VALUE',
      PhoneNumber: phoneNumber
    }
  };
  pinpoint.phoneNumberValidate(paramsMobile, function(err, data) {
    if (err) console.log(err, err.stack); // an error occurred
    else {
      console.log(phoneNumber, "is", data.NumberValidateResponse.PhoneType);
      if (data.NumberValidateResponse.PhoneType == "MOBILE") { // If phone number is a mobile send the PIN
        var message = JSON.stringify({ PIN: PIN, Length: process.env.pinLength, Phonetyp: data.NumberValidateResponse.PhoneType });
        EmergencyDB(null, JSON.parse(message));
        console.log(PIN);
      }
    }
  });
}
```

Click on “Deploy” to save the Lambda.



Screenshot of the AWS Lambda Code source interface. The top navigation bar includes tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Code tab is selected. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, a Test dropdown, Deploy, and a status message 'Changes deployed'. A red arrow points to the 'Changes deployed' button.

Next, click on the “Configuration” menu to add the environmental variables.

Screenshot of the AWS Lambda Functions page. The URL is Lambda > Functions > OneTimePin. The function name is OneTimePin. The configuration section shows a description, last modified (7 months ago), and Function ARN (arn:aws:lambda:ap-southeast-1:841392346146:function:OneTimePin). Below the configuration, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. The Configuration tab is highlighted with a red arrow. A status message 'Code source: Info' is at the bottom left, and 'Upload from' is at the bottom right.

Please note that the environmental variables are case-sensitive.

Add in the following 4 environmental variables and respective values.

- appId
- pinLength
- region
- senderId



aws Services ▾

Search for services, features, marketplace products, and docs [Option+S]

Lambda > Functions > OneTimePin

OneTimePin

Function overview Info

OneTimePin (0) Layers + Add trigger + Add destination Description - Last modified 7 months ago Function ARN arn:aws:lambda:ap-southeast-1:841392346146:function:OneTimePin

Code Test Monitor Configuration Aliases Versions

General configuration Triggers Permissions Destinations Environment variables Tags VPC Monitoring and operations tools Concurrency Asynchronous invocation Code signing Database proxies File systems State machines

Environment variables (4)

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
appId	df212e2e... (red arrow)
pinLength	6
region	ap-southeast-1
senderId	PinAlert (red arrow)

Edit

aws Services ▾

Search for services, features, marketplace products, and docs [Option+S]

Lambda > Functions > Pin2 > Edit environment variables

AWS Lambda

Dashboard Applications Functions Pin2 Edit environment variables Additional resources Code signing configurations Layers Related AWS resources Step Functions state machines

Edit environment variables

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](#)

Key	Value
appId	df212e2e... (red arrow)
pinLength	6
region	ap-southeast-1
senderId	PinAlert

Add environment variable

Encryption configuration

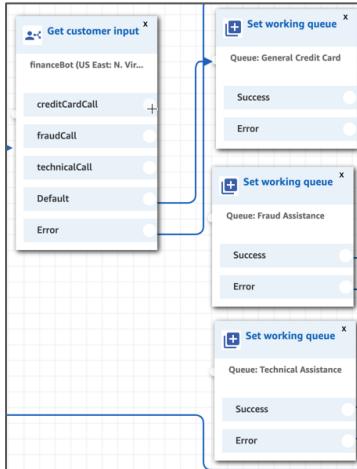
Cancel Save

Closer look above.



Now we can incorporate the one time pin verification into our contact flow.

Continuing from earlier section where you created your Lex Bot, you can insert in an additional flow to check for one time pin after the creditCardCall option, and before the “Set Working Queue” block.



Invoke AWS Lambda function

Makes a call to AWS Lambda and optionally returns key/value pairs, which can be used to set contact attributes. [Learn more](#)

Function ARN

Select a function
arn:aws:lambda:ap-southeast-1

Use attributes

Function input parameters

Use text
 Use attribute

Destination key: concernedNumber

Type: System

Attribute: Customer Number

Invoke function: arn:aws:lambda:ap-southeast-1

Success

Error

Check contact attributes X

Branches based on a comparison to the value of a contact attribute. [Learn more](#)

Attribute to check

Type: External

Attribute: PhoneType

Conditions to check:

- = MOBILE
- No Match

[Add another condition](#)

Check contact attributes X

Checking: External, Phon...

- = MOBILE
- No Match

Note: Set both condition to next block for testing purposes..

Store customer input X

Stores numerical input to contact attribute.

Plays an interruptible audio prompt and stores digits via DTMF as a contact attribute. [Learn more](#)

Prompt:

- Select from the prompt library (audio)
- Text-to-speech or chat text
- Enter text

Please enter our \$.External.Length digit PIN sent to your mobile phone.

Enter dynamically

Interpret as: Text

Customer input:

- Custom

Maximum Digits: 8	Timeout before first entry: 30
In seconds	

- Encrypt entry (recommended)
- Specify terminating keypress [Learn more](#).
- Disable cancel key [Learn more](#).
- Phone number

Store customer input X

Custom

- Success
- Error



Check contact attributes X

Branches based on a comparison to the value of a contact attribute. [Learn more](#)

Attribute to check

Type: System

Attribute: Stored customer input

Checking: System, Stored...

= \$.External.PIN (radio button)

No Match (radio button)

Conditions to check

x Equals (radio button) \$.External.PIN

No Match

Add another condition

Play prompt X

Delivers an audio or chat message. [Learn more](#)

Prompt

Select from the prompt library (audio)

Text-to-speech or chat text

Enter text

<speak>
You have been successfully authenticated, your current account balance is <say-as interpret-as="number"> 525

Enter dynamically

Interpret as: SSML

SSML Text:

<speak>

You have been successfully authenticated, please hold while we transfer you to our agent for details!

</speak>

If there's no match, then you might want to play following prompt.

Play prompt

Delivers an audio or chat message. [Learn more](#)

Prompt

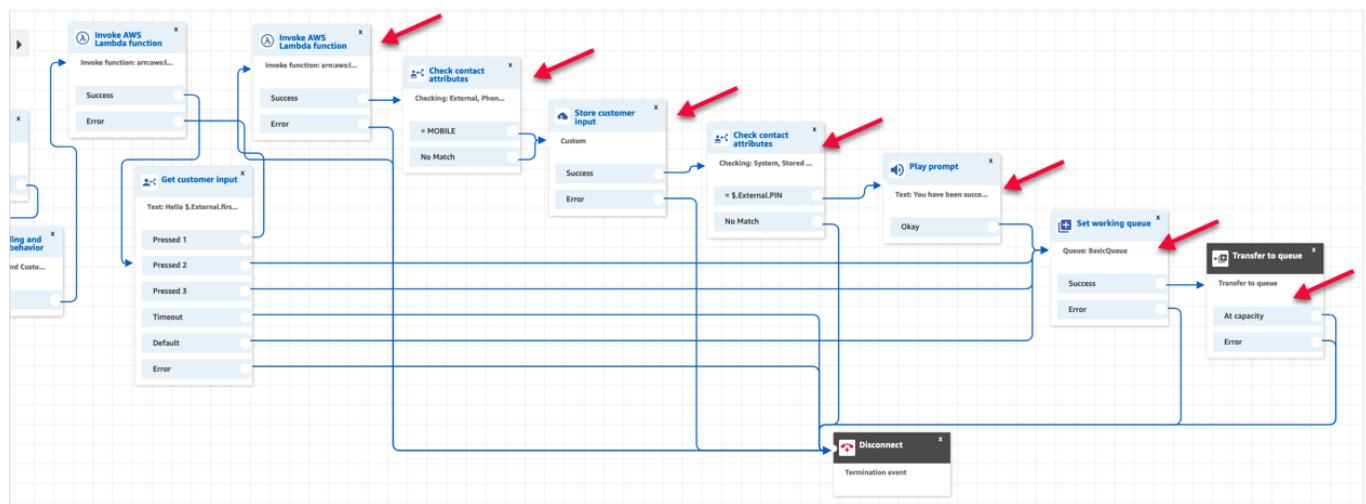
- Select from the prompt library (audio)
- Text-to-speech or chat text
- Enter text

Your PIN did not match. Goodbye.

- Enter dynamically

Interpret as

Text



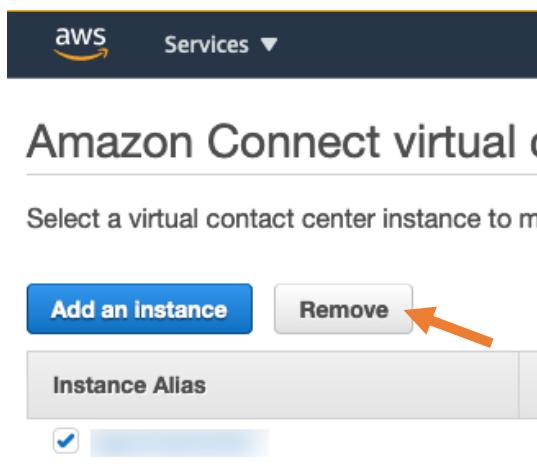
And time to test out the one-time pin generation SMS in the flow !

(Pls do note that the default spending limit for Amazon Pinpoint SMS is \$1... 😊).

Part 11: Deleting account

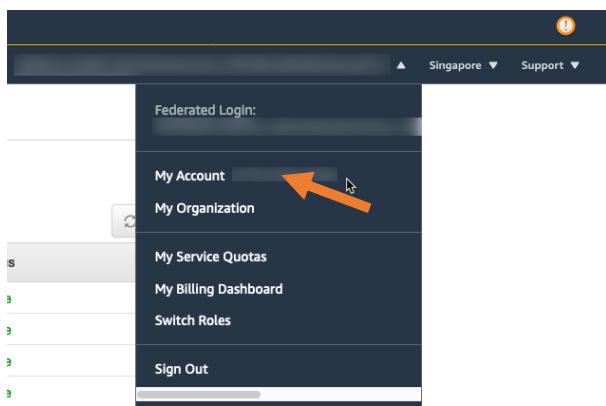
Objective: Delete your account to prevent any incidental costs.

- 1) Go into your management console. Search and click into **Amazon Connect**.
- 2) You will be able to delete your account from there. And you will need to type the name of the instance to confirm deletion.



Note: IF you would like to delete your entire AWS account.

- 3) On the right-hand side, click on your account name with the drop down of **My Account**.



- 4) Scroll to the very bottom of the **My Account** page. You will be able to close off your entire AWS account from here.

▼ Close Account

I understand that by clicking this checkbox, I am closing my AWS account. The closure of my AWS account serves as notice to AWS that I wish to terminate the AWS Customer Agreement or any other agreement with AWS that governs my AWS account, solely with respect to that AWS account.

Monthly usage of certain AWS services is calculated and billed at the beginning of the following month. If I have used these types of services this month, then at the beginning of next month I will receive a bill for usage that occurred prior to termination of my account. In addition, if I have any active subscriptions (such as a Reserved Instance for which I have elected to pay in monthly installments), then even after my account is closed I may continue to be billed for the subscription until the subscription expires or is sold in accordance with the terms governing the subscription.

I acknowledge that I may reopen my AWS account only within 90 days of my account closure (the "Post-Closure Period"). If I reopen my account during the Post-Closure Period, I may be charged for any AWS services that were not terminated before I closed my account. If I reopen my AWS account, I agree that the same terms will govern my access to and use of AWS services through my reopened AWS account.

If I choose not to reopen my account after the Post-Closure Period, any content remaining in my AWS account will be deleted. For more information, please see the the [Amazon Web Services Account Closure page](#).

I understand that after the Post-Closure Period I will no longer be able to reopen my closed account.

I understand that after the Post-Closure Period I will no longer be able to access the Billing Console to download past bills and tax invoices.

If you wish to [download any statements you can do so here](#). Select the month and expand the summary section to download the payment invoices and/or tax documents.

I understand that after the Post-Closure Period I will not be able to create a new AWS account with the email address currently associated with this account.

If you wish to update your e-mail address, [follow the directions here](#).

Close Account



Thank you for going through this hands-on guide with us! Keep on building!

