# Amazon Connect Hands-On Guide (Level 100)

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

Nov 2020 rev 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 important to start sequentially from Part 1 in order to best understand the presented material. 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.

# Revisions

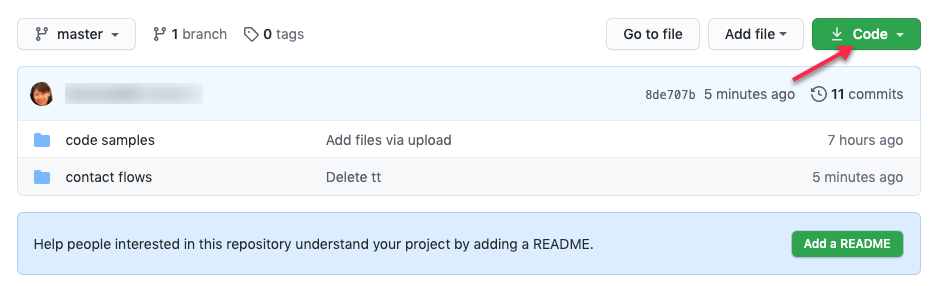
|  |  |
| --- | --- |
| Sep 2020 | Version 1.0 |
| Nov 2020 | Version 2.0  Added section 8, 8.1 and 8.2 for caller identification and personalization. Create new DDB, lambda and contact flow.  Added section 9, 9.1 for Amazon Lex voice bot. |
|  |  |
|  |  |
|  |  |

# 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.

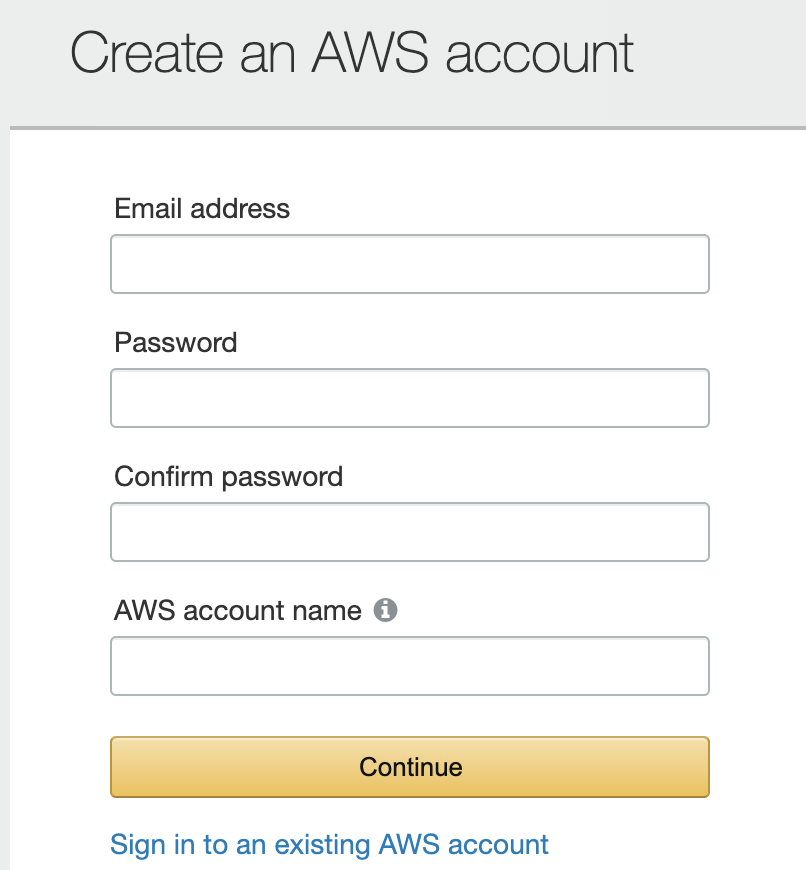
# \*Part 1: Create your own AWS Account

**\*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.

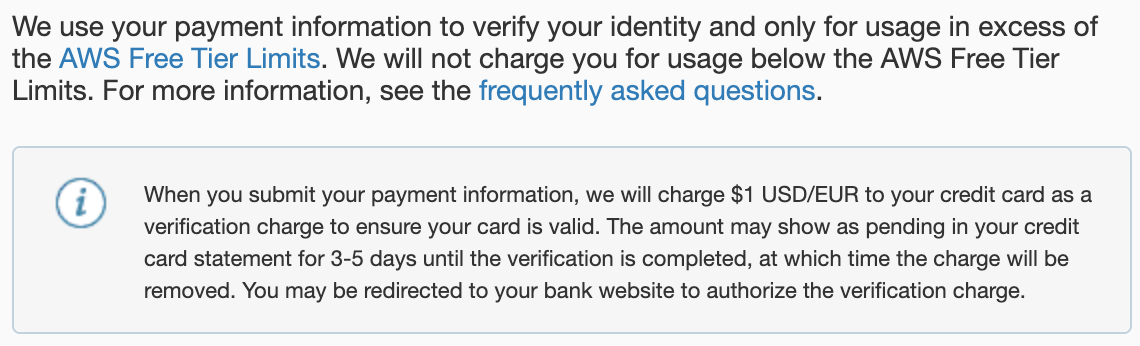
## Objective: Create an AWS instance

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.

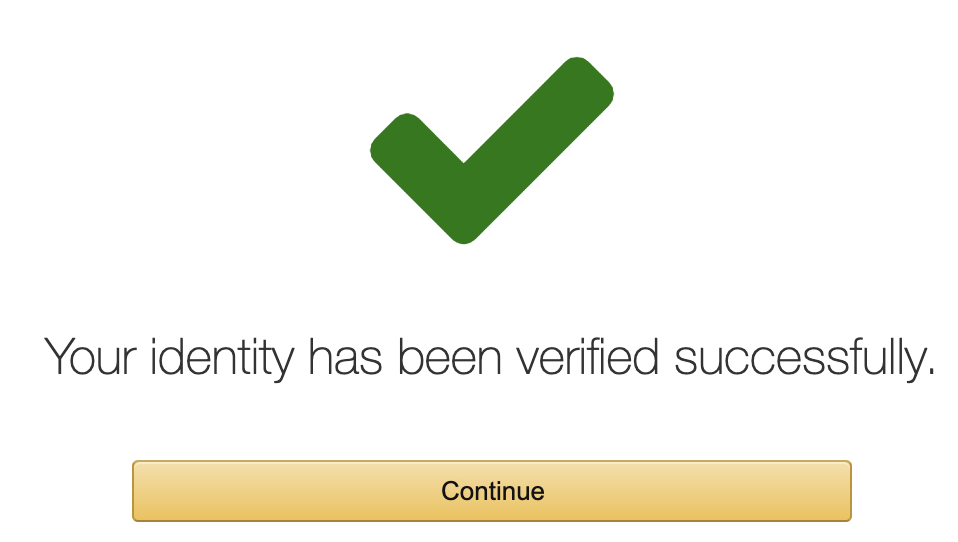
1. If you do not have a AWS account, navigate to <https://portal.aws.amazon.com/billing/signup#/start>. Go through the steps to have your own FREE account (AWS Free Tier).



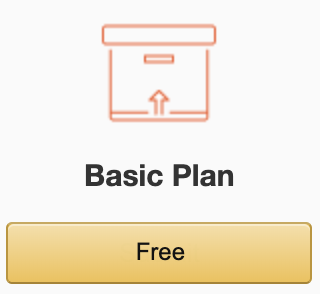
1. Click on “Professional”, enter in your company’s address and information.
2. Enter in your Credit Card information. Note that the $1 charge will be removed once your credit card is verified.



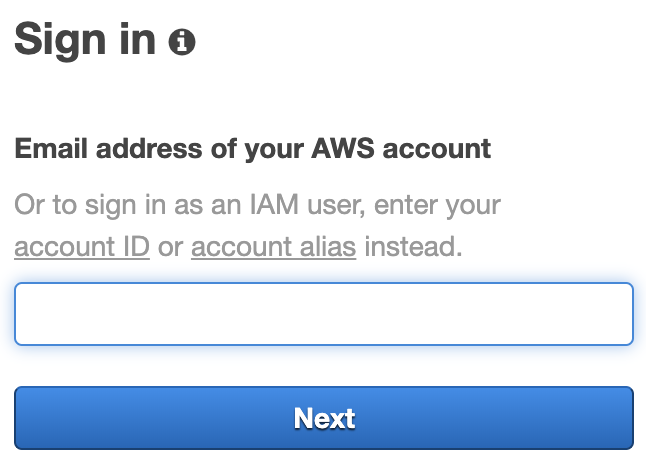
1. Verification of your account via SMS or Phone Call.



1. Click on “Free”



1. Sign into the Management Console. Input the email address and password used.



**You are all set now!**

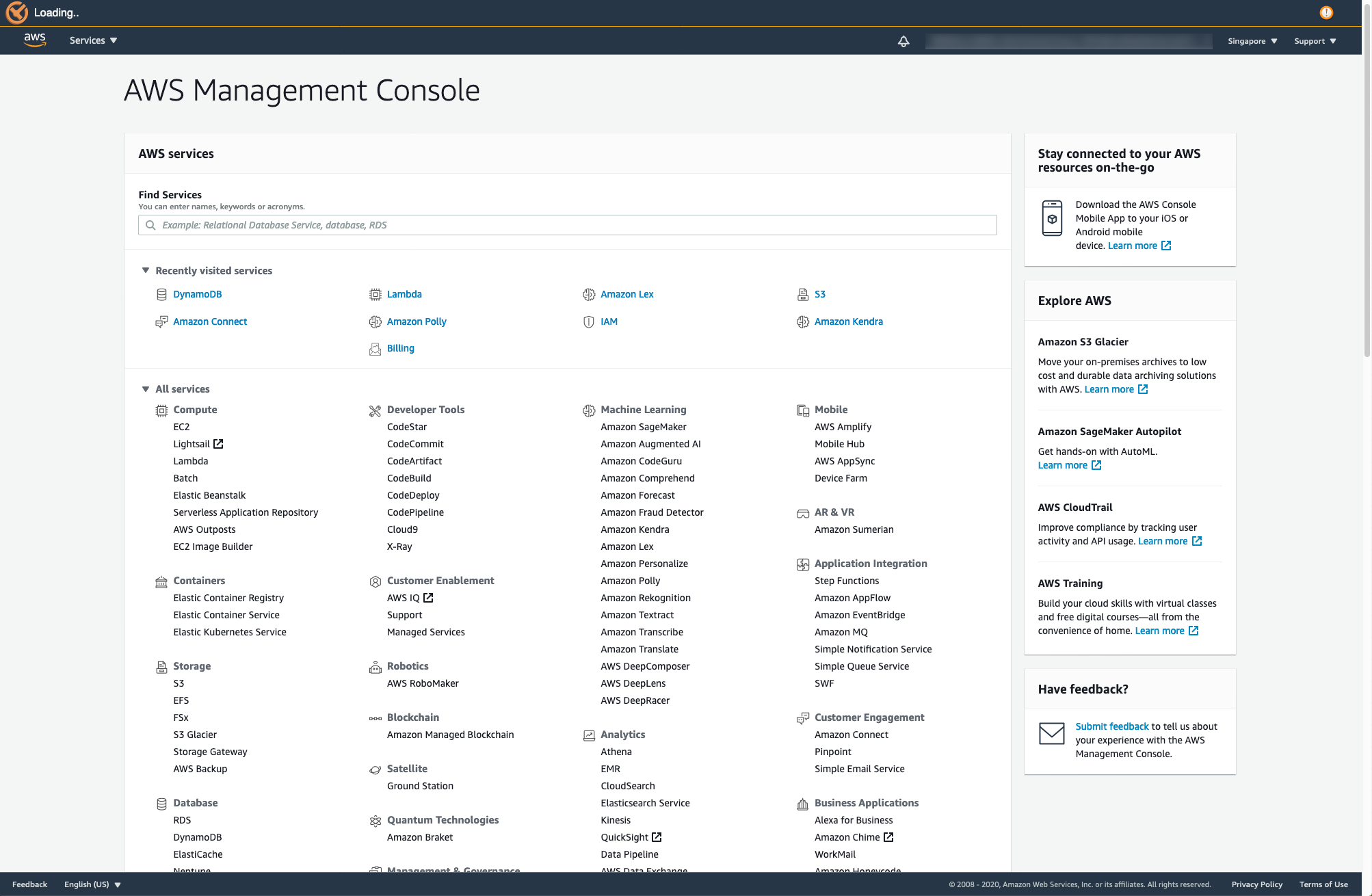
# Part 2: AWS Management Console

## 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  .  
  
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.



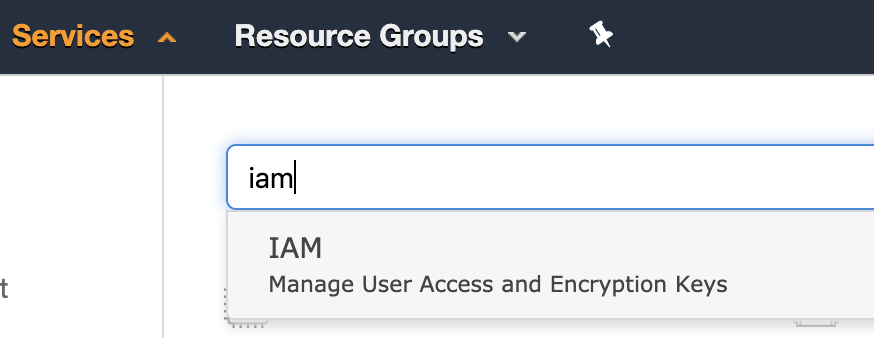
# 

# Part 3: IAM (Identity and Access Management)

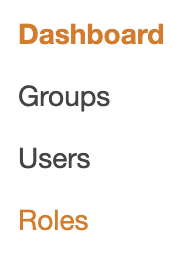
## Objective: Setting up the rights and permissions for your lambda function.

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.

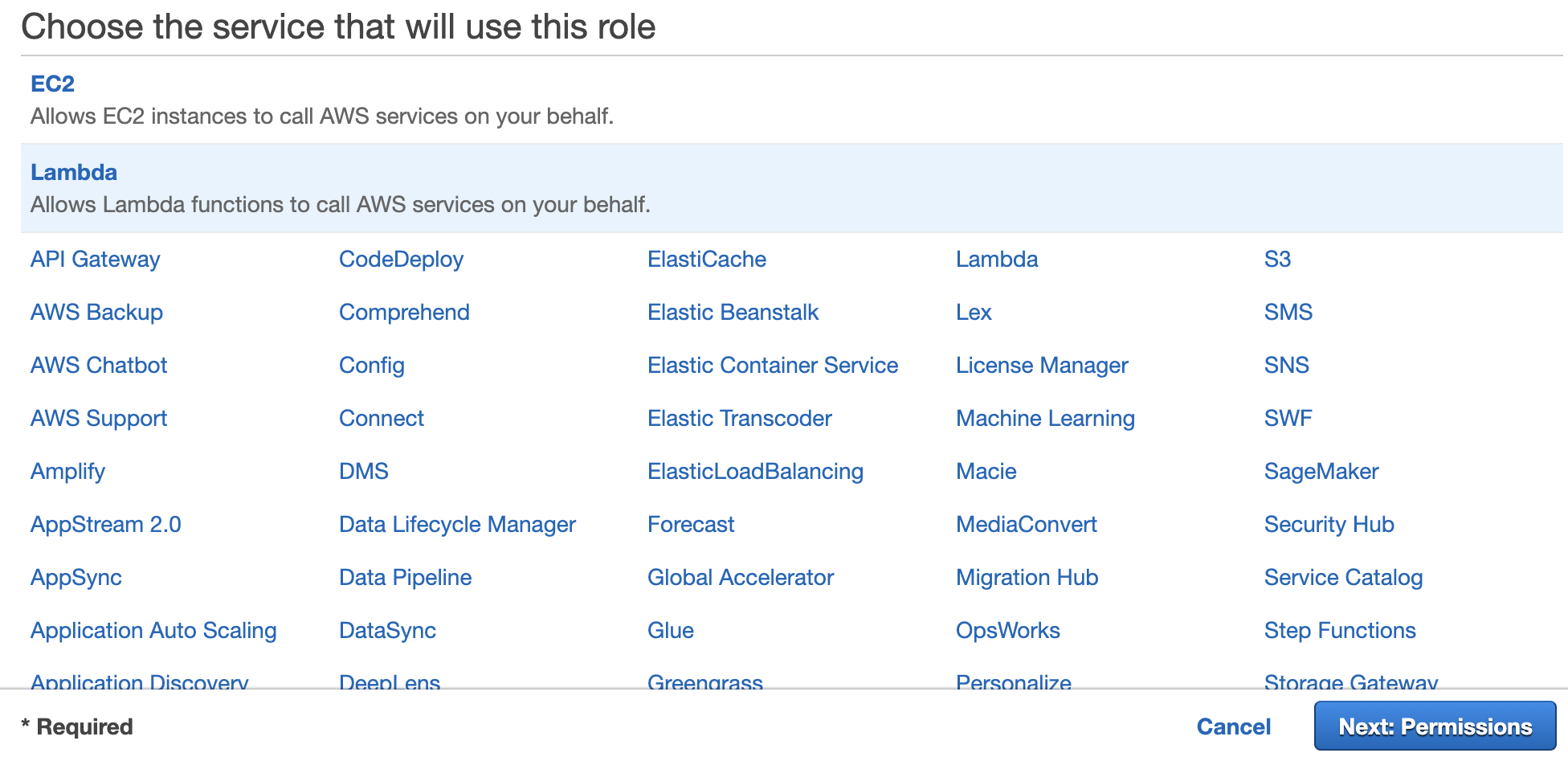
Navigate to AWS management console. Enter in **IAM.**



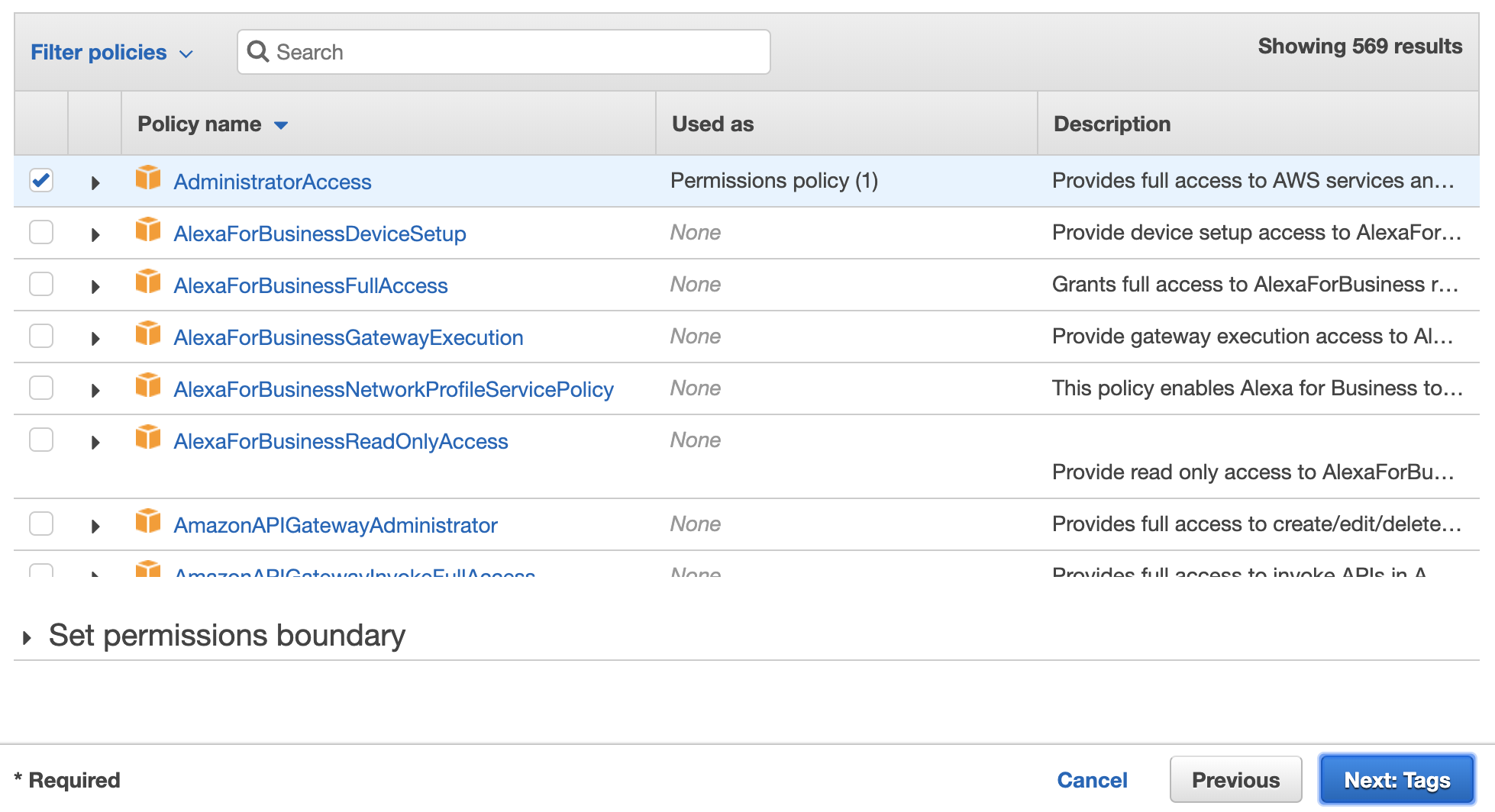
1. On the left-hand column, click on **Roles.**



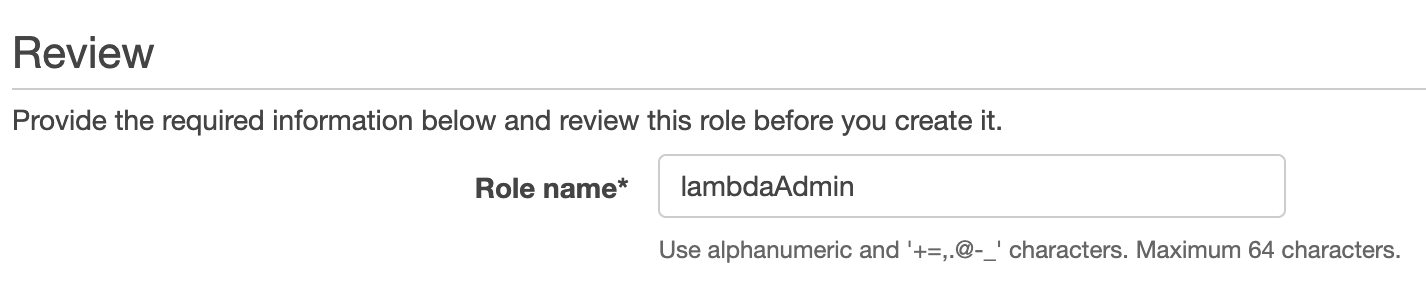
1. Click on the button in blue **Create Role**
2. Click on **Lambda** and click on **Next: Permission** button.

****

1. Select AdministratorAccess and click Next: Tags.



1. No need for any tags, click on **Next: Review**
2. Give the role a name **lambdaAdmin**. And go ahead with the creation of the role.



1. 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.

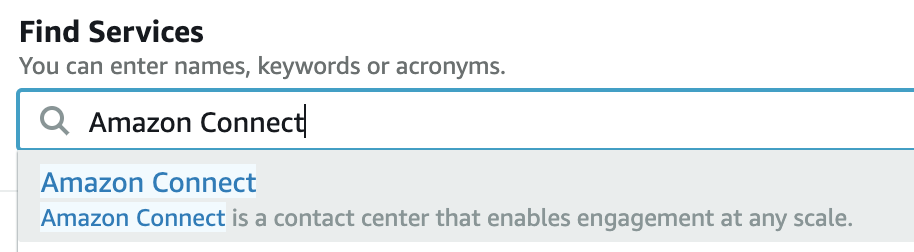
## Objective: Creating your very first Amazon Connect instance

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.

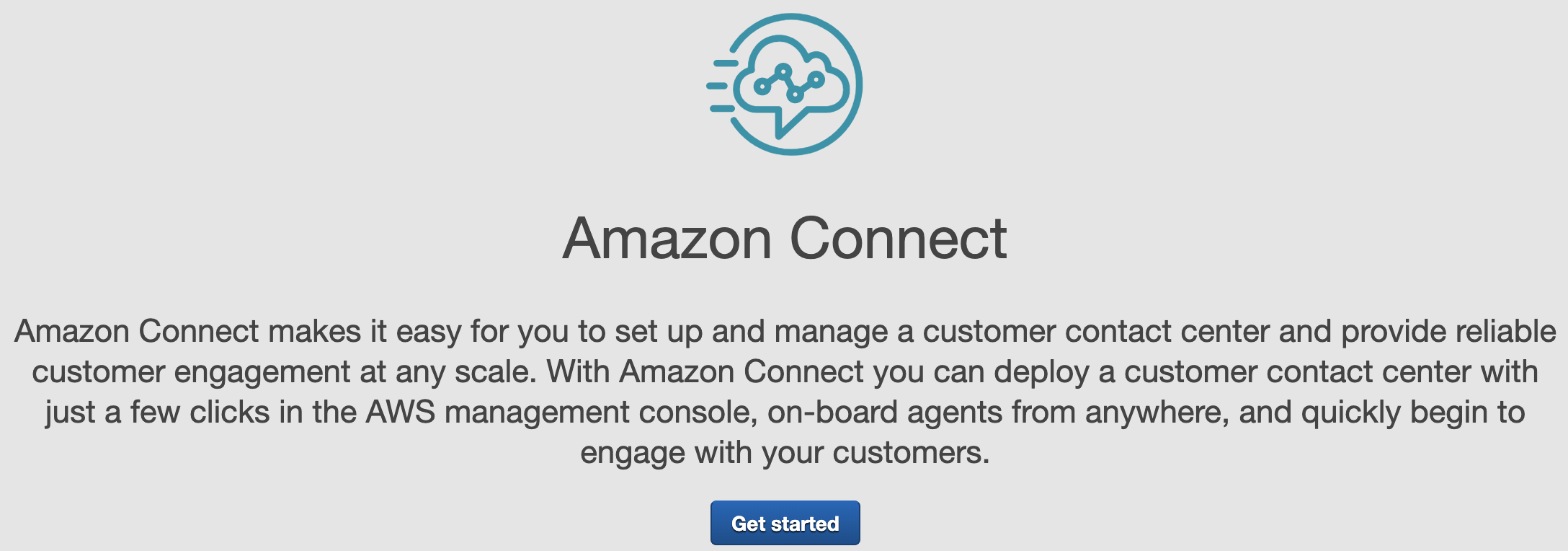
1. Check that you are in the **Asia Pacific (Singapore) ap-southeast-1** region.



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



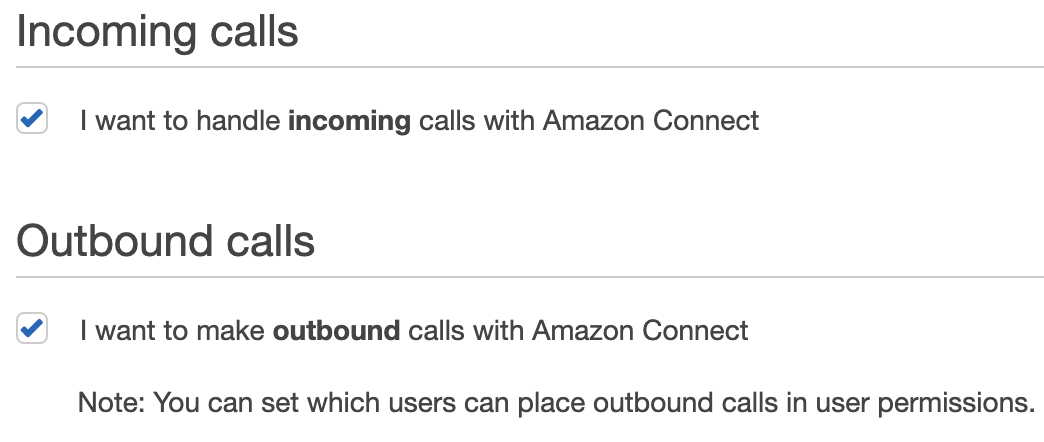
1. If this is your first time navigating to Amazon Connect, click **Getting Started**. Otherwise, you will see a list of instances that have already been built with an **Add an Instance** button. Click either **Add an Instance** or **Getting Started.**

****

1. 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.
2. 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**

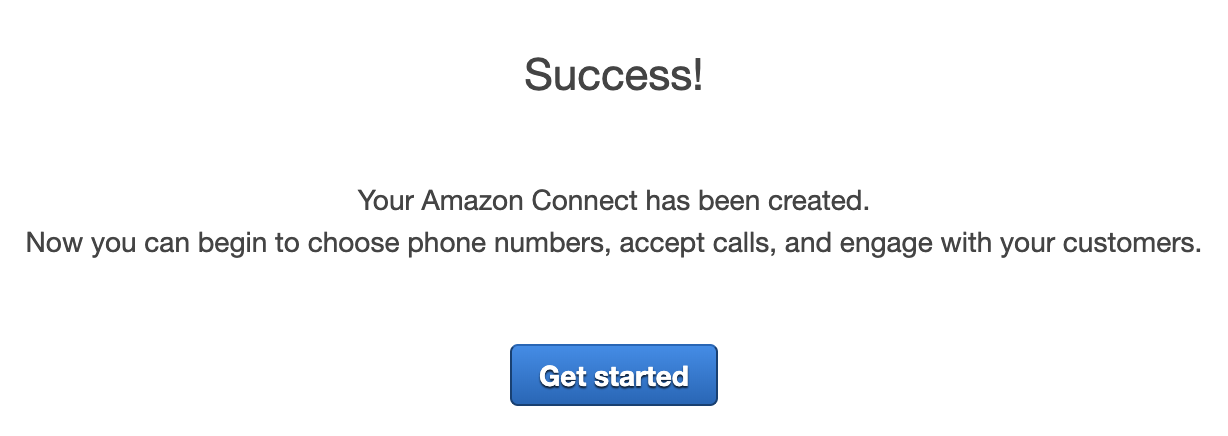
1. Leave the Inbound and Outbound options checked and click **Next Step**.



1. 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.**

1. Review the details listed on the final confirmation screen and click **Create Instance.**
2. Once complete, click **Getting Started** on the success page.
3. 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.



# Part 5: DynamoDB Table

## 

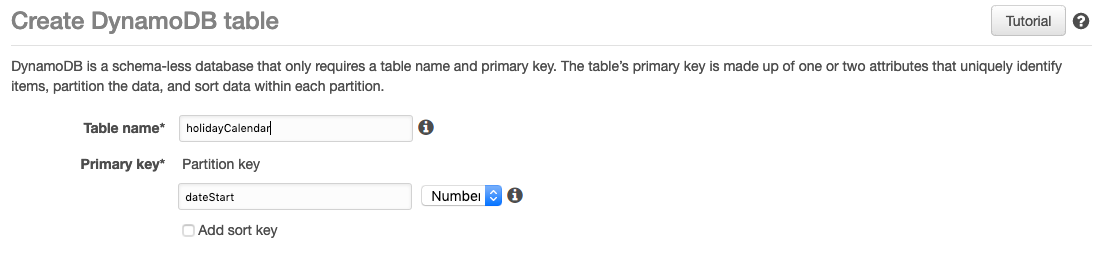
## Objective: Using a DynamoDB table to store holidays / special events / ad-hoc messages

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

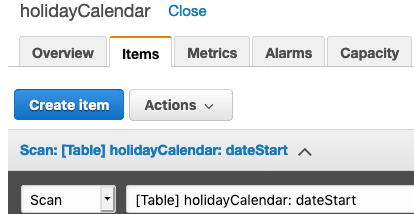


1. Click on
2. Enter in **holidayCalendar** for the table name (or any other table name you like), and **dateStart** for the primary key. Click on the blue Create table at the bottom of the page. The table will take a short while to be created.

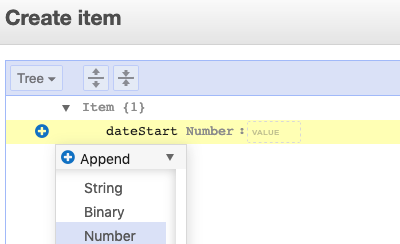
**Note:** Names are cap-sensitive



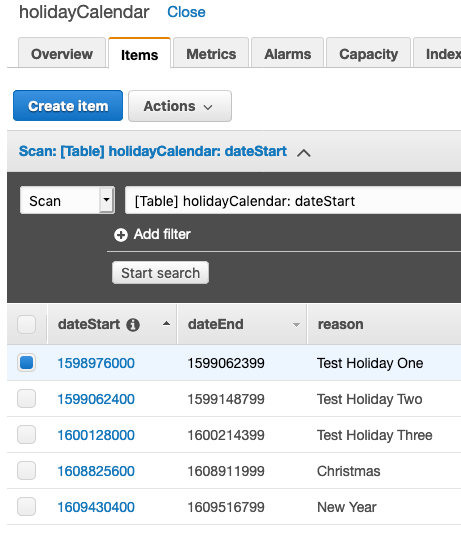
1. Create new line items in the table. Click on **Create item.**

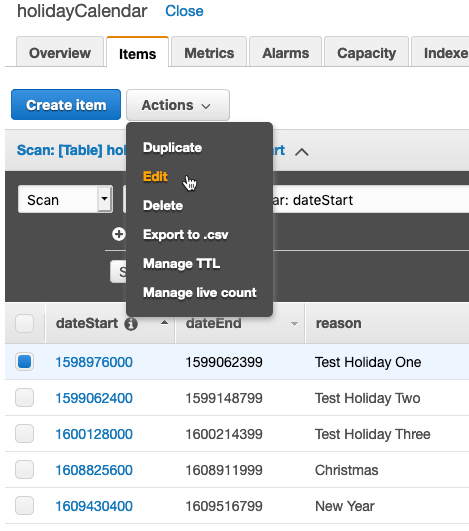


1. Enter in **dateStart**, **dateEnd** and **reason**. To add more fields, click on the plus sign and **Append**. Again, do take note that field names are cap sensitive.

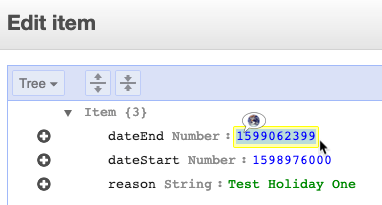


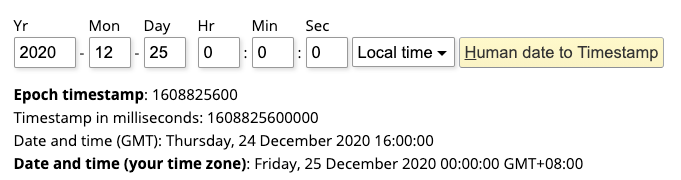
Your table should look similar to the following:

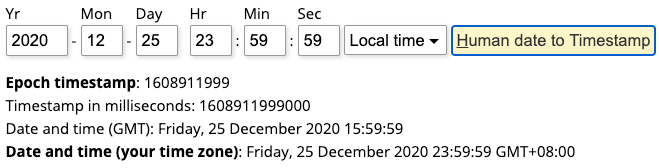


1. If you want to edit or append more lines to your table, just select any of the row you want to edit or append to and click on the **Actions** button then click on **Edit**.  
   

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



1. 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**  
   

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

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

1. 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.
2. All set for your table! You can come back to DynamoDB to add more lines or edit this table.

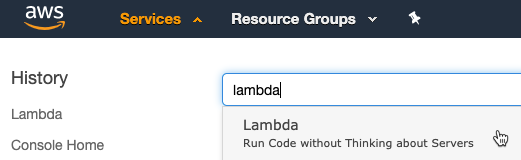
# 

# 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.**



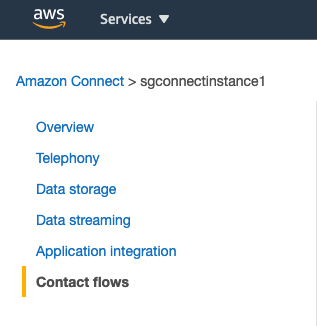
1. Click on the orange button on the right **Create function.**
2. Enter in function name of **holidayCalendarRead,** or any name you prefer.   
   Leave the runtime as **Node.js 12.x.**
3. Click on the arrow for the dropdown of Choose or create an execution role.   
   Select Use an existing role.
4.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



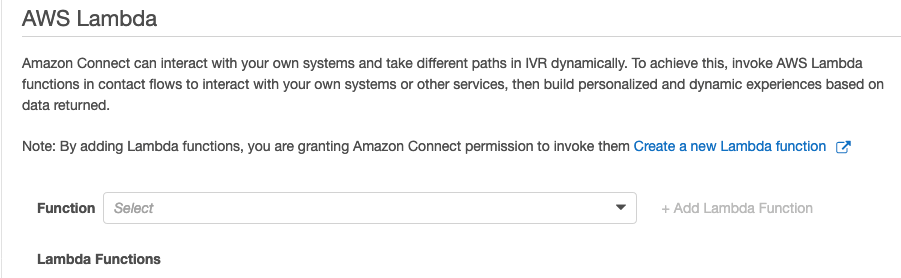
1. Click on **Create function** button.
2. From the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named “code samples”, click into it and locate the holidayCalendarRead.js file.
3. Copy and paste all the codes in there into the new lambda function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



1. Go on to management console, select **Amazon Connect.**
2. Click on Contact Flows



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

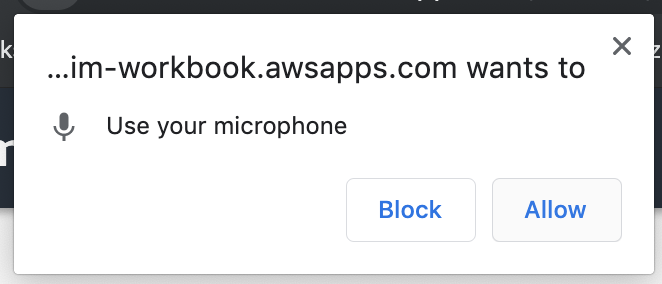


# Part 7: Configuring Amazon Connect

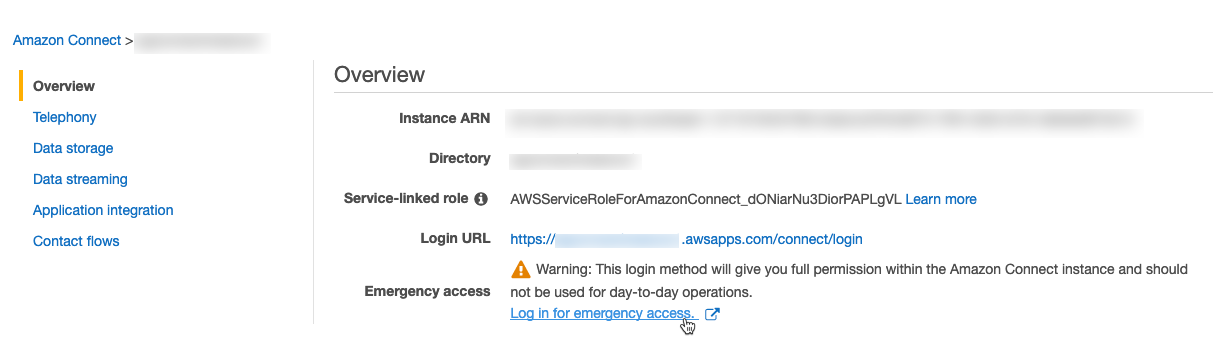
## 

## 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.



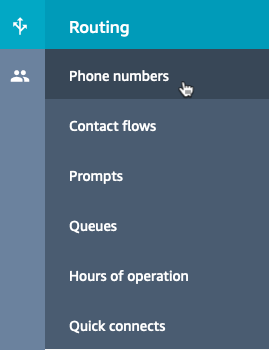
1. Go to Management Console. Click on Amazon Connect.
2. Navigate to the Emergency access URL link created for your Amazon Connect instance shown below and click to log in.   
     
   If you have created your own Amazon Connect administrative username and password earlier on, you can use that to login using the Access URL created as well.



# 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.

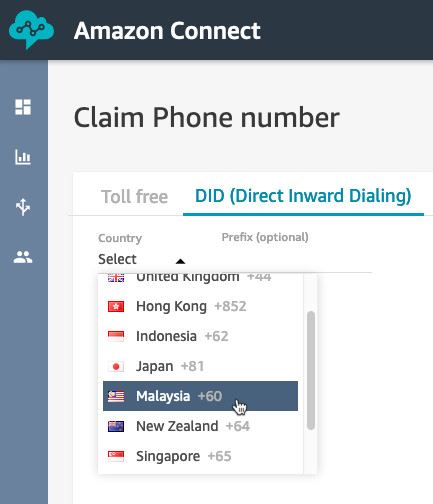
1. Navigate to the Routing > Phone Number section

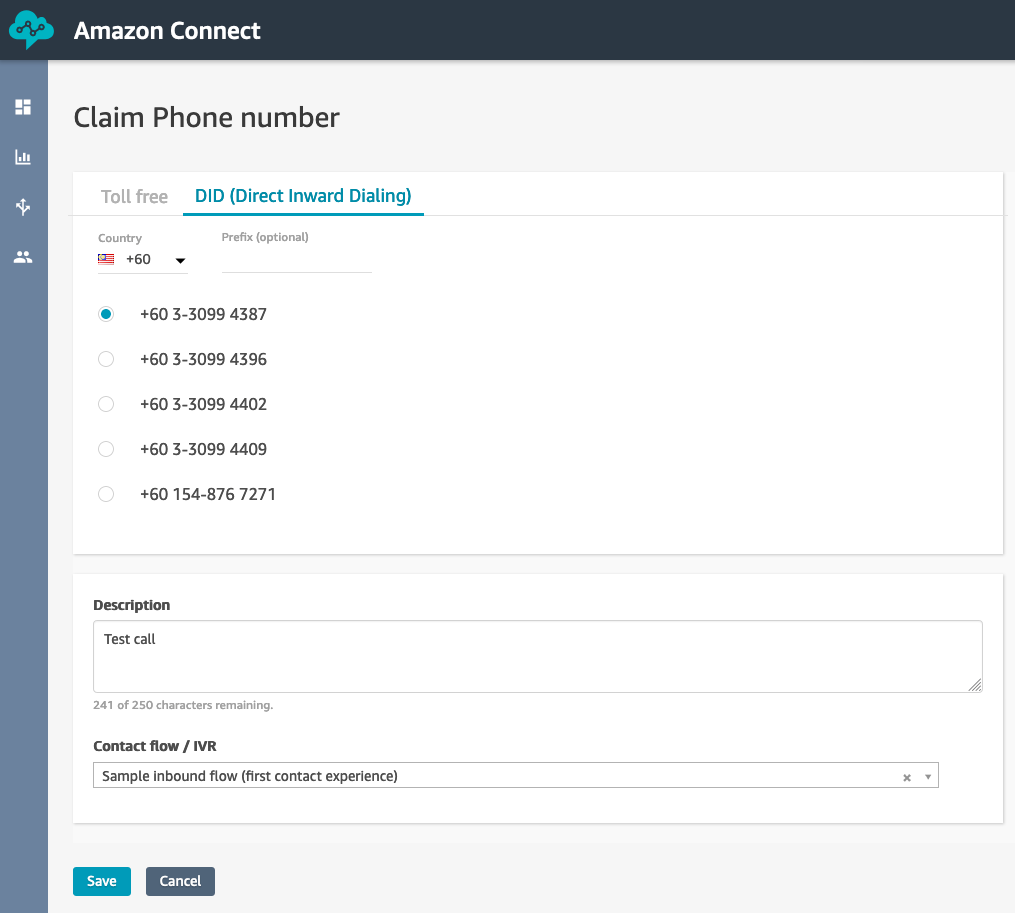


1. Click Claim a number.

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.

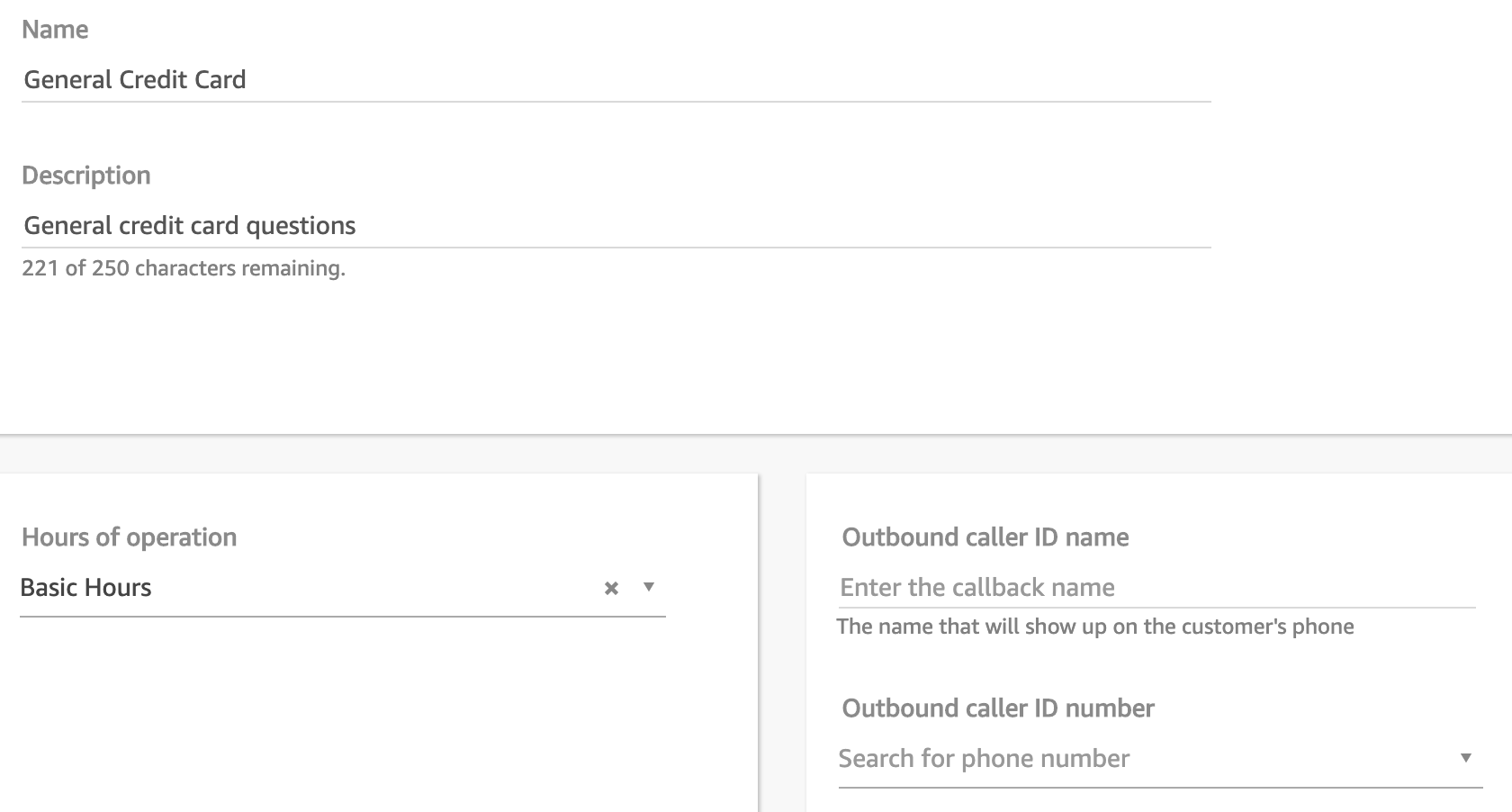


1. Notice the **Basic Queue** option. Amazon Connect comes preconfigured with a sample queue to help get started. Leave this alone and select **Add new queue**.
2. Queue names should reflect the skills required to solve problems. When naming, ensure the name aligns with the skill required for agents 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.



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.

1. Repeat step 3. For 2 other queues with queue names below.
   1. General Credit Card
   2. Fraud Assistance
   3. Technical Assistance

# Part 7.3: Routing Profiles (Optional)

## 

## Objective: Configuring routing profiles

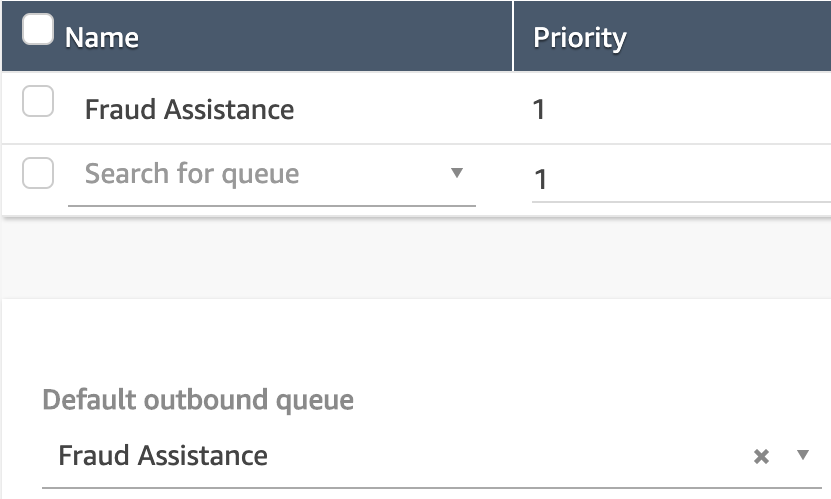
Gives an understanding of how to configure agents to the queues.

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 |



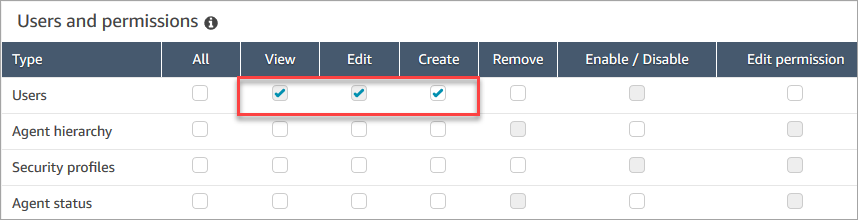
# Part 7.4: Users Management

## 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.

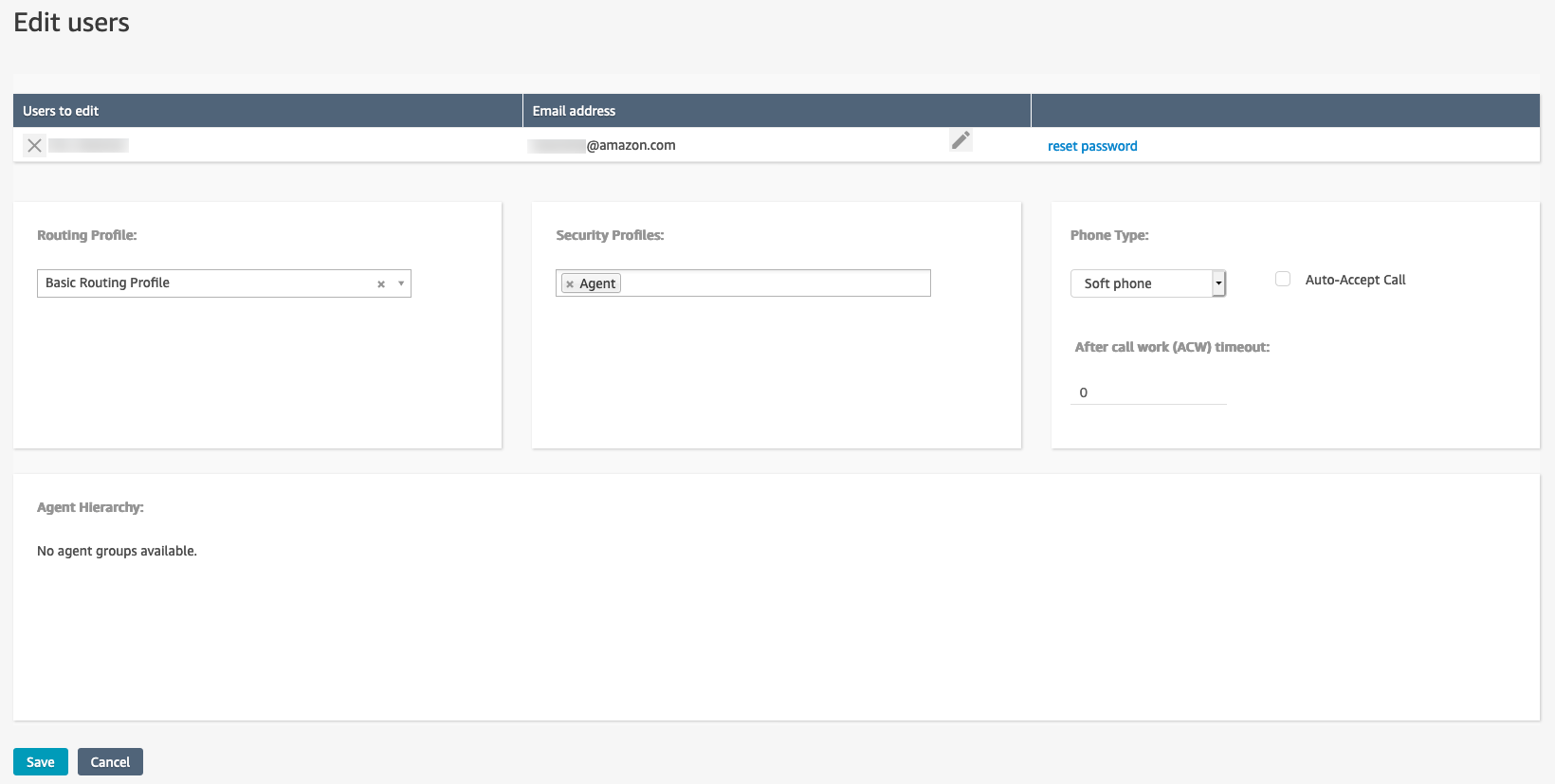
Before you can add users to Amazon Connect, you need the following permissions assigned to your security profile: **Users - Create**.



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.



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

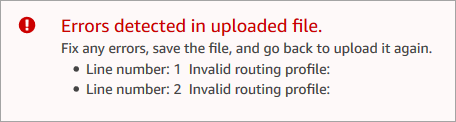
## Add users in bulk from a .csv file

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.



# 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](https://docs.aws.amazon.com/connect/latest/adminguide/check-hours-of-operation.html) 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](https://docs.aws.amazon.com/connect/latest/adminguide/create-queue.html), and check them in the [Check hours of operation](https://docs.aws.amazon.com/connect/latest/adminguide/check-hours-of-operation.html) block.

## How to specify midnight

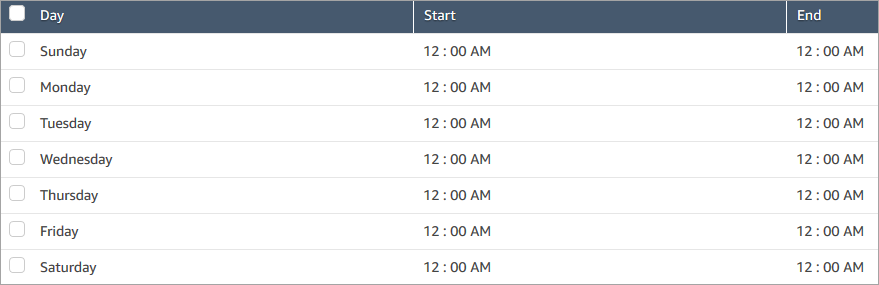
To specify midnight, enter 12:00AM.

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

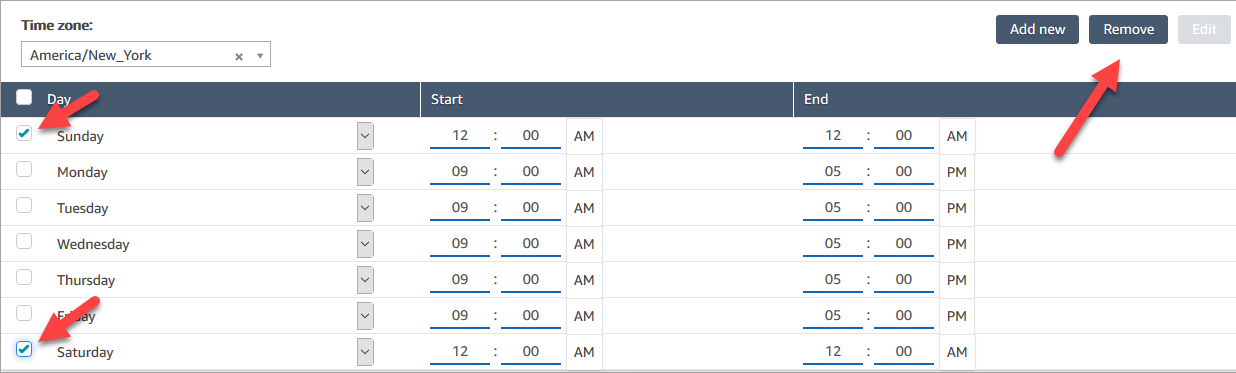
## Examples

**Schedule for 24x7**

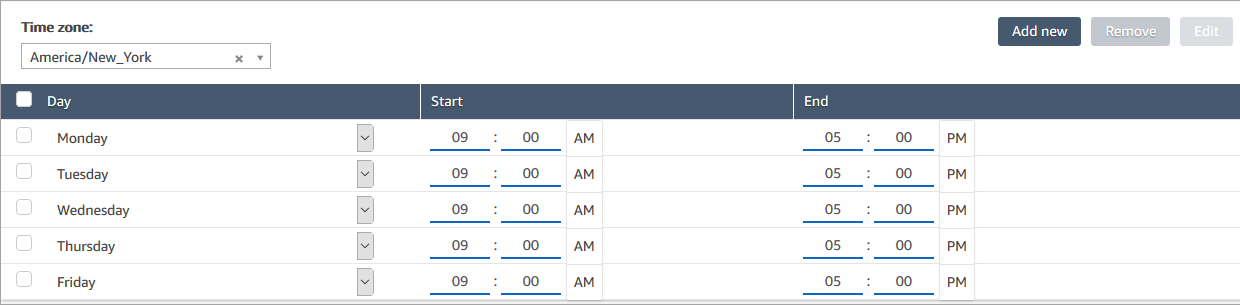


**Schedule for Monday to Friday 9:00 AM to 5:00 PM**

Remove Sunday and Saturday from the schedule.

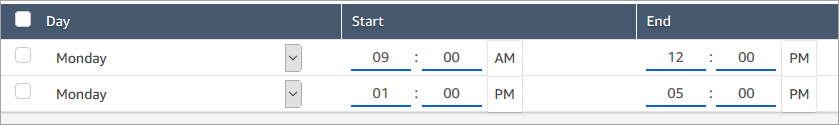


The final schedule looks like this:



## 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:



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](https://docs.aws.amazon.com/connect/latest/adminguide/agent-custom.html) 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](https://docs.aws.amazon.com/connect/latest/adminguide/check-hours-of-operation.html) 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

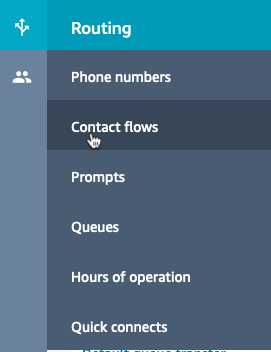
By the end of this section, you would have imported an existing contact flow file and gone through step by step this contact flow and understand how to leverage a lambda to data dip whenever needed.

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.

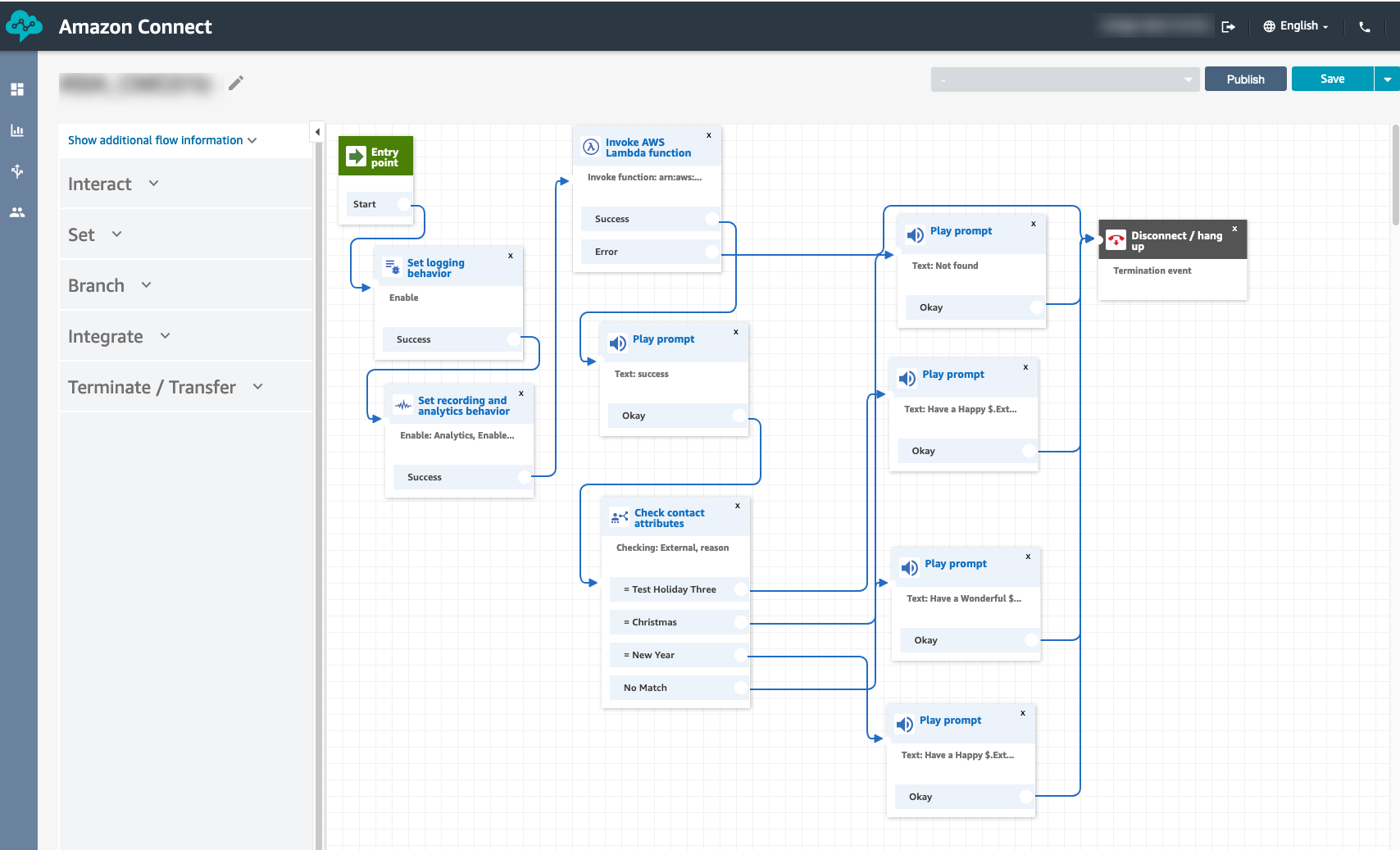
This serves as the basis of getting started with contact flow design in Amazon Connect.

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

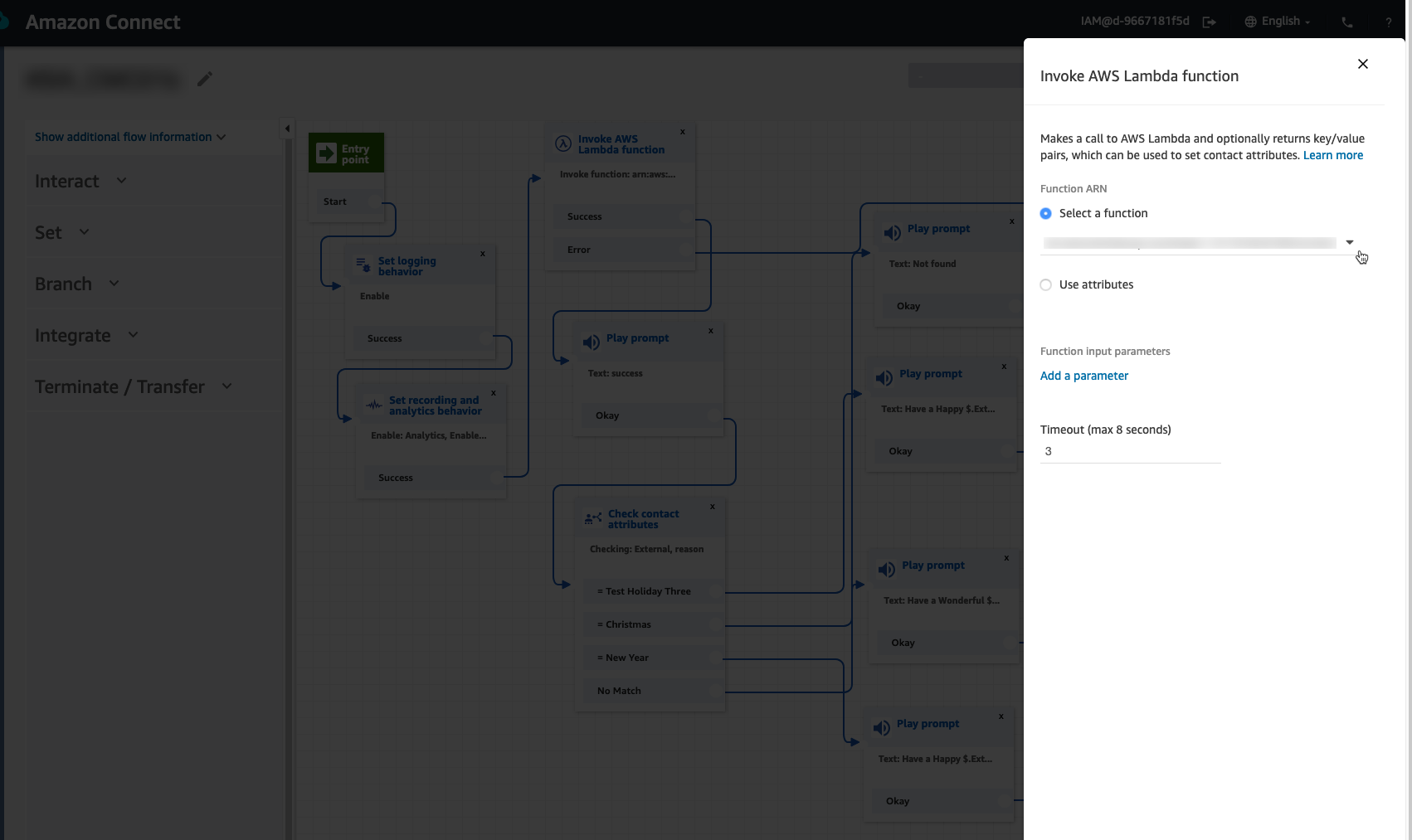
1. First, we need to get the contact flow file. From the contents of the ZIP file downloaded from <http://bit.ly/lilichanworkshop>, you should see a folder named “contact flows”, within it you will see a file named “myFlow1”, this is the contact flow file which you will be importing.
2. From Amazon Connect management console, navigate to **Routing > Contact flows** section.  
   And click on **Create contact flow**.

1. From the down arrow, click on Import flow (beta).  
     
   
2. Click on that to select the contact flow which you have downloaded earlier.
3. 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 ☺).



1. 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!  
     
   *< will be inserting a YouTube video to showcase Contact Lens for Amazon Connect here soon..>*
2. Next, click on the **Invoke Amazon Lambda function** block, and select the earlier lambda which you have created – holidayCalendarRead.

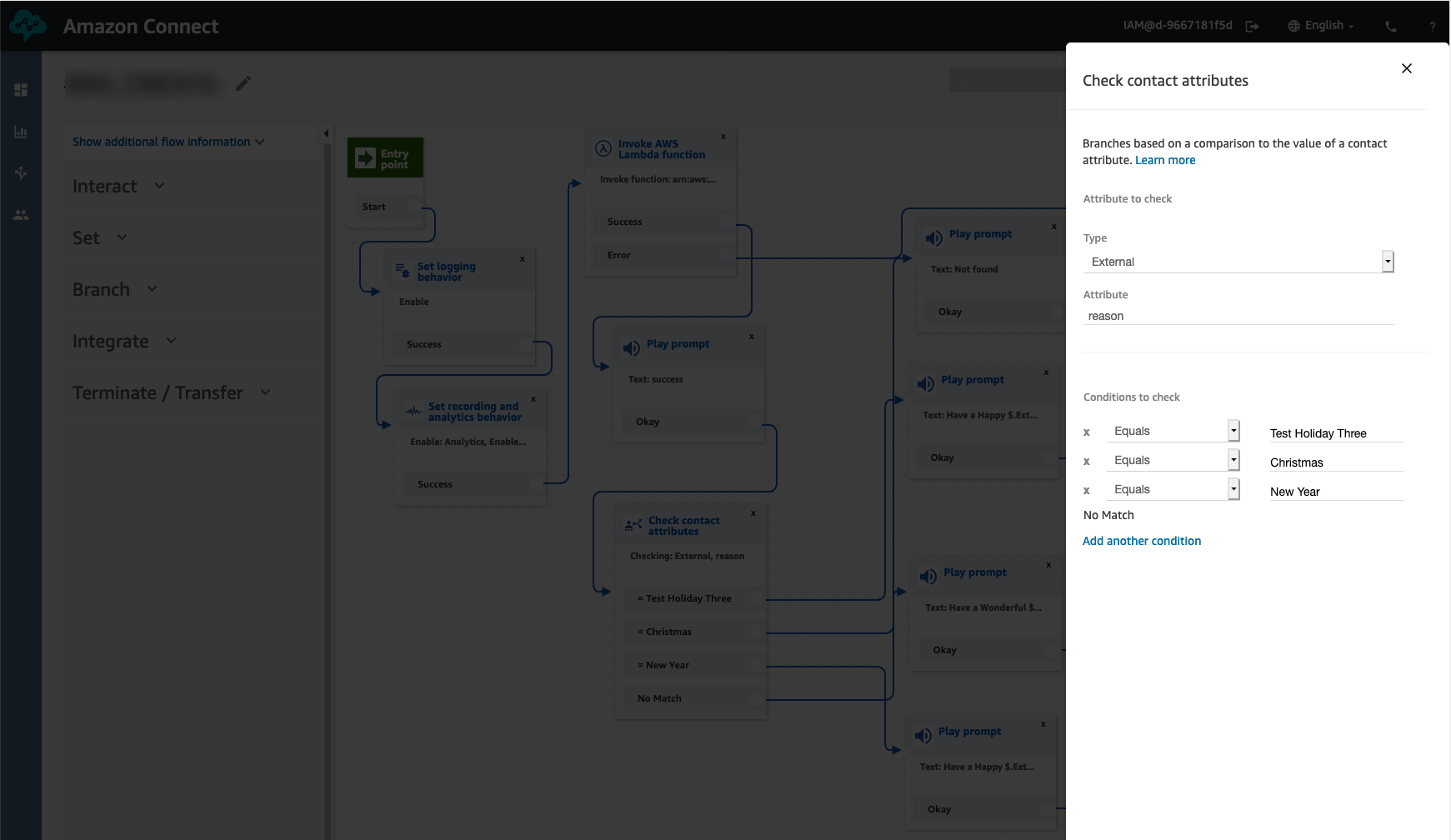


1. The next block which is a **Play Prompt** is just used for testing purpose.
2. The next block is a **Check contact attributes** block.  
     
   Contact attributes is one of the most important concept which you will need to know within Amazon Connect!  
     
   Pls look through following sections 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.

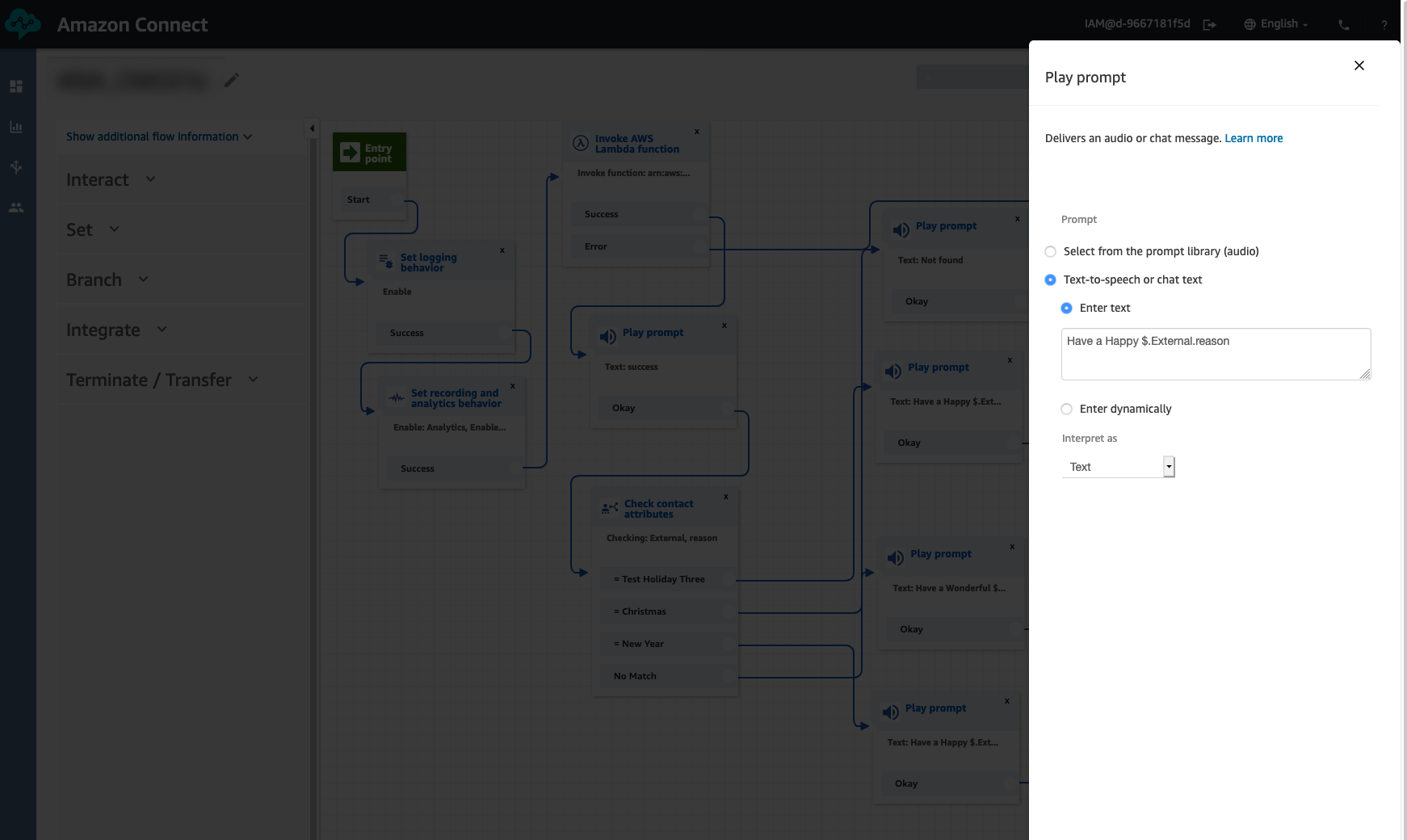
Using contact attributes, you can customize and personalize the experience customers have when they interact with your contact centre. Contact attributes let you store customer input or data about a customer, and then use it later in a contact flow. You can also check the values of contact attributes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

1. 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.  
     
   
2. If you want to save your contact flow, just click on the **Save** button at the top right-hand corner.  
     
   

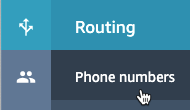
This will save your changes but will not publish (activate) the flow.

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.

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



1. Under **Contact flow/ IVR,** change this setting to the name of our Contact flow.
2. Click **Save.**
3. Dial this number and listen to the interaction. Check that the flow runs as what you have set it up to be.
4. Congratulations on your first successful contact flow!

# Part 8: Personalization

## Objective: Incorporating personalization into your contact flow

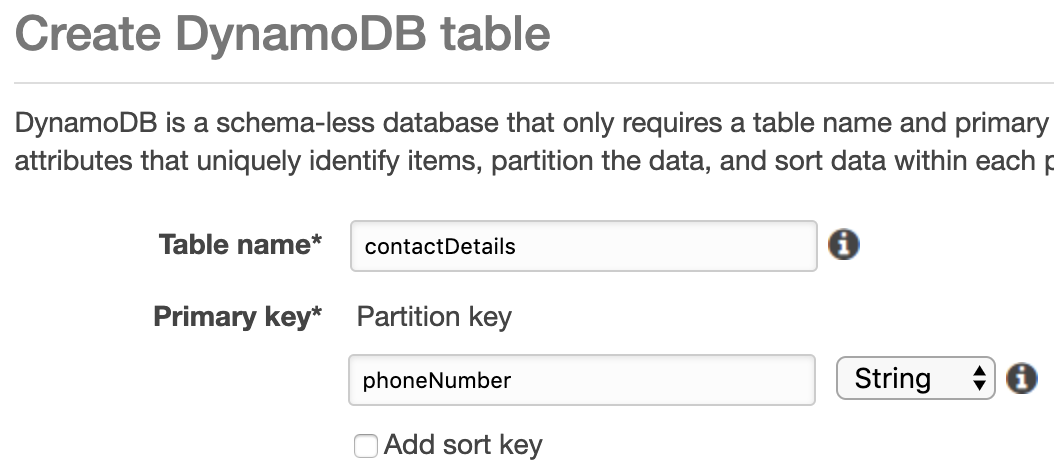
By the end of this part, you would have a greater understanding on how we are simulating data that exists in your existing backend system or any CRM system that you may have. We will be leveraging another DynamoDB table as a data dip point later on in this part.

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

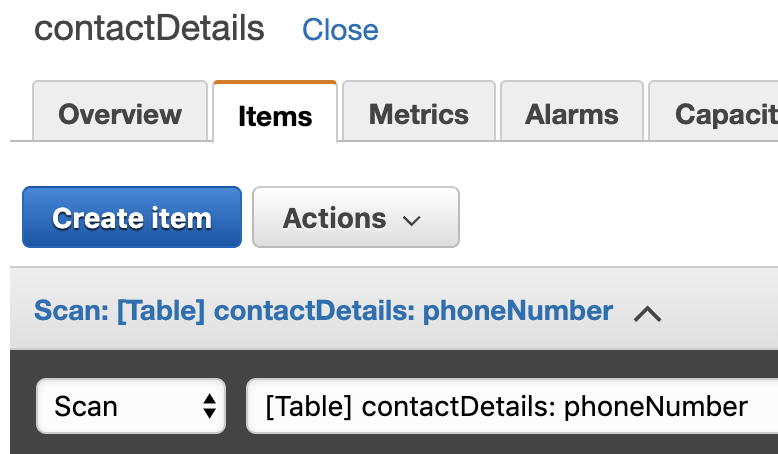


1. Click on
2. Enter in **contactDetails** for the table name, and **phoneNumber** for the primary key. Click on the blue Create table at the bottom of the page.

\*NOTE: Names are cap sensitive



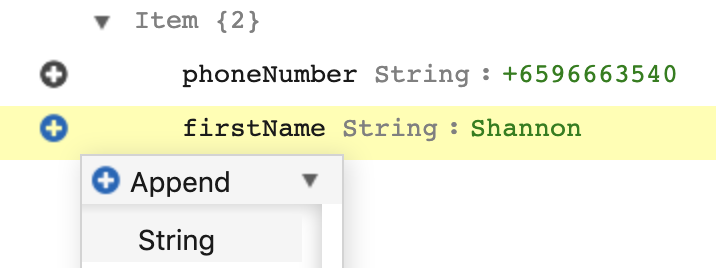
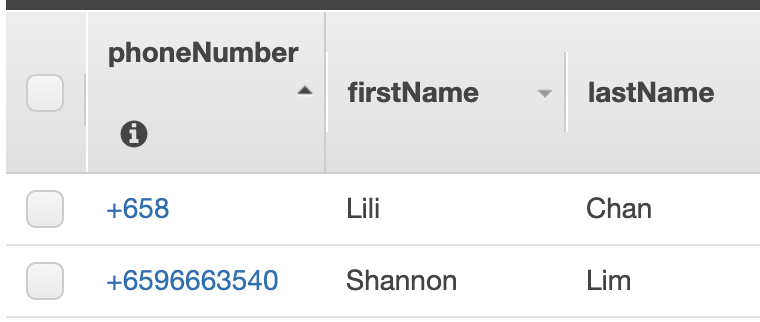
1. Create new line items in the table. Click on **Create item.**



1. 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.

You table should look like the table on the right.

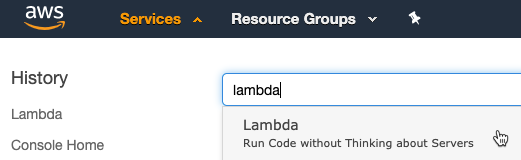


1. All set for your table! You can come back into DynamoDB to add more lines or edit this table.

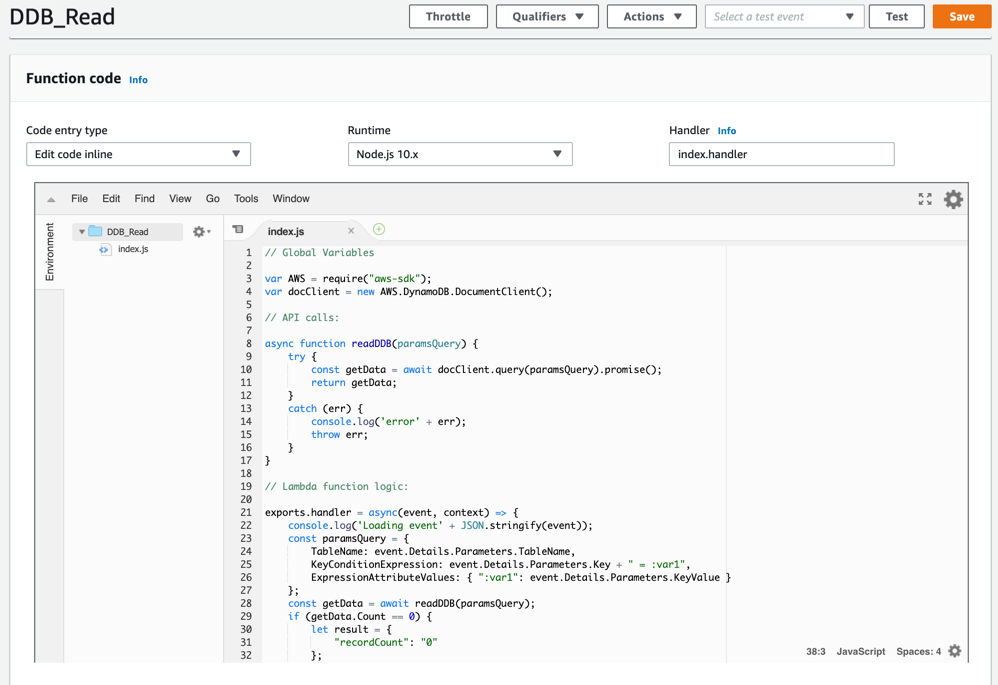
# Part 8.1: Lambda used for DDB Read

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.

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



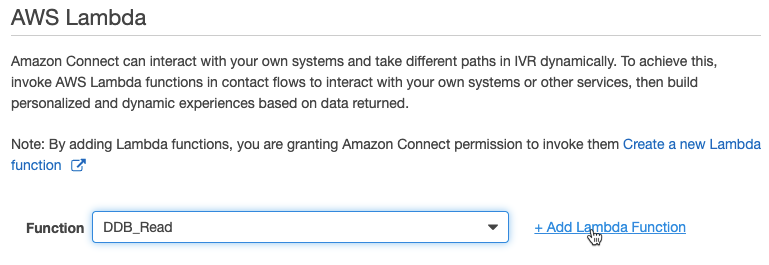
1. Click on the orange button on the right **Create function.**
2. Enter in function name of **DDB\_Read.** Or any name you prefer. Leave the runtime as **Node.js 10.x.** (Note: You can also choose Python if you prefer to code in this language).
3. Click on the arrow for the dropdown of Choose or create an execution role. Select Use an existing role.
4. Select **lambdaAdmin** in the IAM section above which you have previously created.
5. Click on Create function.
6. 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](https://github.com/lilichan888/workshop/blob/master/code%20samples/sample_dynamo_get_node.js) file using your favorite code editor tool. (Note: For Python, the file would be [sample\_dynamo\_get\_python.py](https://github.com/lilichan888/workshop/blob/master/code%20samples/sample_dynamo_get_python.py))
7. Copy and paste the codes from entire file.
8. Save the lambda function.



1. Go on to management console, select **Amazon Connect**
2. Click on Contact Flows



1. Scroll down and search for **AWS Lambda**. Add in the **DDB\_Read** function you have just created. And click **+ Add Lambda Function.**

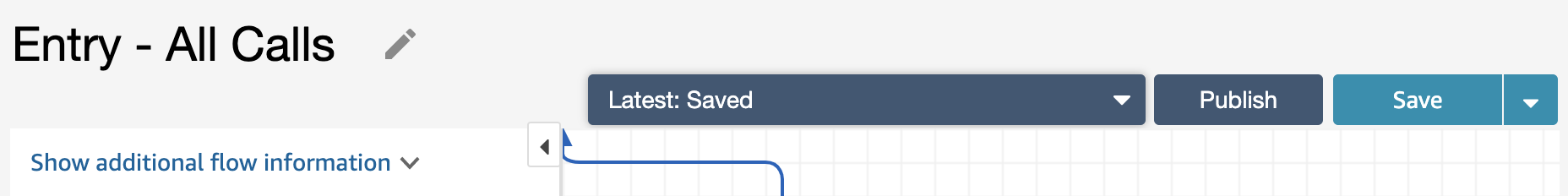


# Part 8.2: Create a contact flow to show Personalization

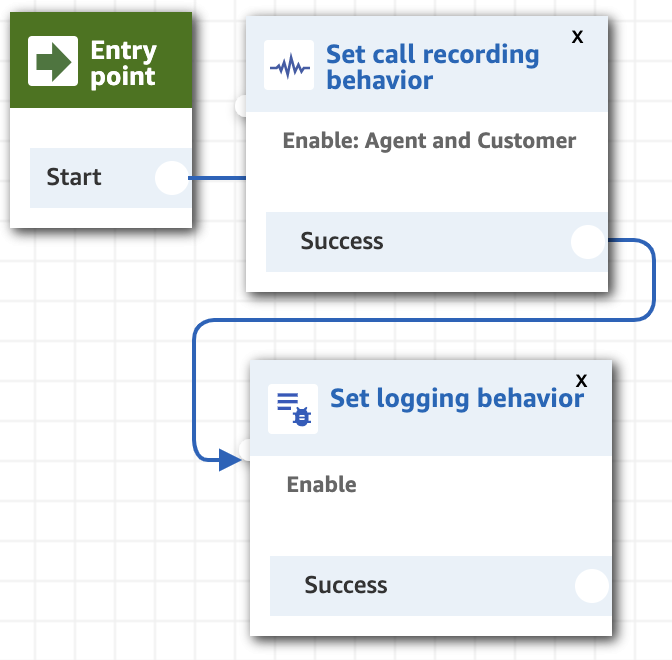
By the end of this part, you would have created another contact flow and understand how to leverage lambda to data dip whenever needed.

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.

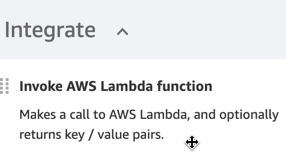
1. Go to Routing > **Contact Flow**
2. Click Create Contact Flow
3. Enter a Name of **Entry – All Calls** or something similar. Enter a description of your choice and click **Save**.



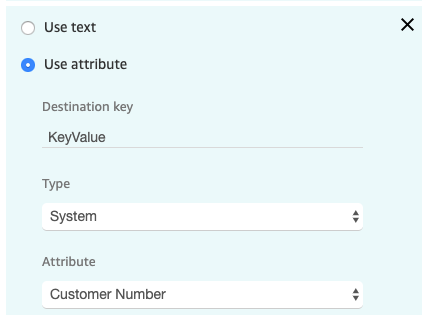
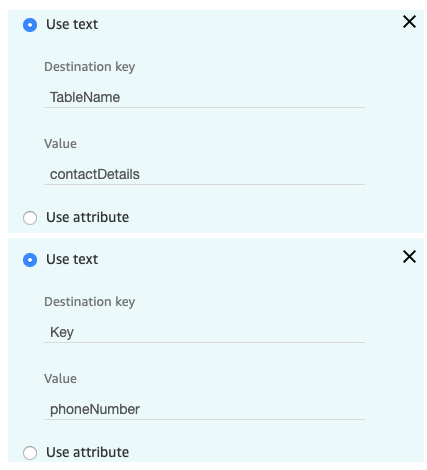
1. Notice the green **Entry Point** box now has a line added. Start adding blocks to build the contact flow from the left and connect the lines to blocks.
2. Expand the **Interact** group and click and drag the **Set call recording behavior** and **Set logging behavior** blocks to the canvas.



1. On the left column of options, drag out **Invoke AWS Lambda function**.



1. Select the lambda function you have created **DDB\_Read.**
2. Click on **Add a parameter**. Add in the destination key and value as below.



1. You should have 3 parameters as seen above. Note that the destination key and value is cap 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.

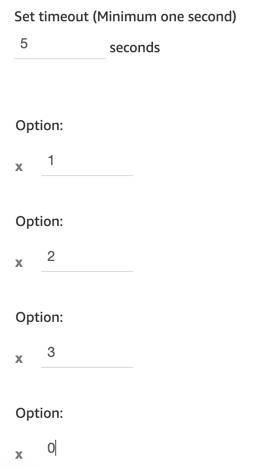
And we are using the system attribute of incoming call number as the search key.

1. Drag over the **Get customer input** box.
2. 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.***

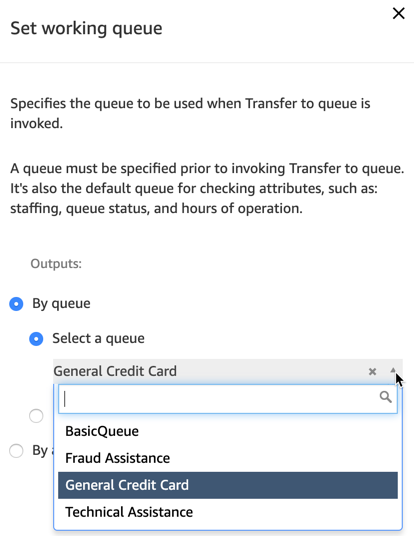


1. Do not close this box—scroll down within the block until you see the **DTMF** header and the link **Add another condition**. Click this link four times and enter **1** in the first line, **2** in the second, **3** in the third, **0** in the fourth. Click **Save**.



Take note of the $.External.attribute section.   
For more information, read here: <https://docs.aws.amazon.com/connect/latest/adminguide/using-contact-attributes.html>

1. 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.



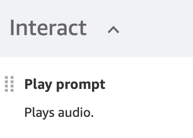
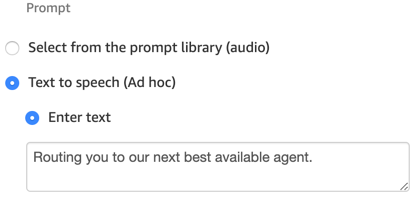
1. Repeat step 13 by pulling out two other **Set working queue** block and setting them respectively for **Fraud Assistance** and **Technical Assistance**.
2. Connect option 1 to the General Credit Card block.

Connect option 2 to the Fraud Assistance block.

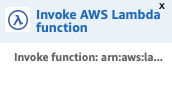
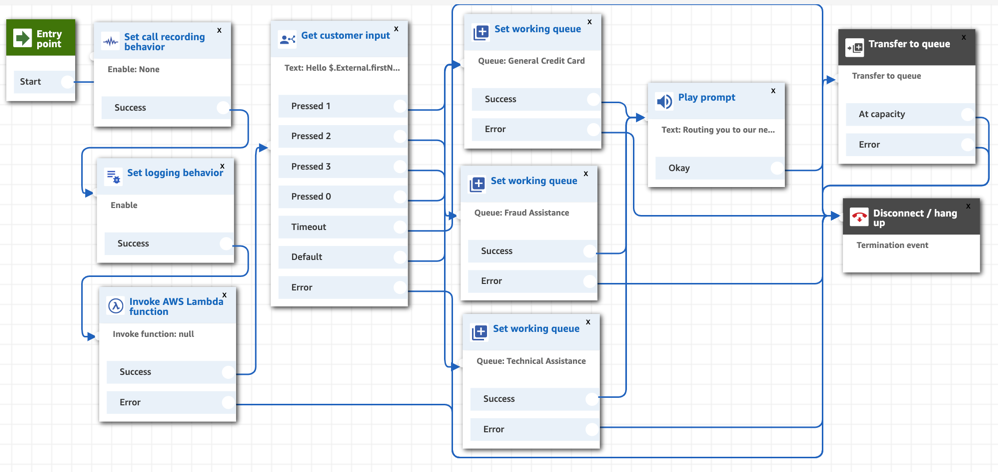
Connect option 3 to the Technical Assistance block.

1. Drag the Play prompt block and enter in the following text.

***Routing you to our next best available agent.***

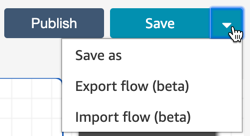
 

1. Connect the 3 **Set working queue** blocks to the **Play prompt** block you have just created.
2. Pull out a **Transfer to queue** block and have all three successful option from **Set working queue** block go to this **Transfer to queue** block.
3. For all other failure options or time out options, drag out and connect them to the **Disconnect / hang up** block.
4. Your contact flow should look like below.

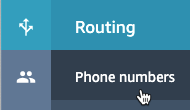


1. We now need to publish our contact flow so it becomes live. Click the **drop-down arrow** and select **Save.** Select **Publish** if there are no errors.

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

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

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



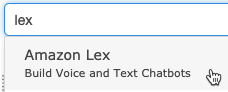
1. Under **Contact flow/ IVR,** change this setting to the name of our Contact flow (**Entry – All Calls**)
2. Click **Save.**
3. Dial this number and listen to the interaction. Check that flow is as you have set up. Notice how the call is put into queue for an agent. Since we have no agents working, it will remain in queue until you disconnect.
4. Congratulations on another successful contact flow!

# Part 9: Creating an Amazon Lex Bot

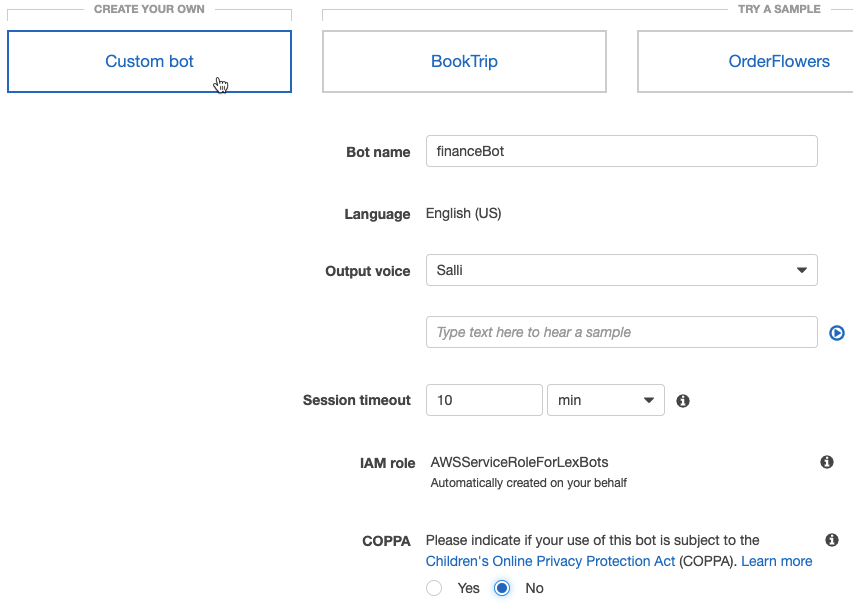
## 

## Objective: Creating your first Lex bot

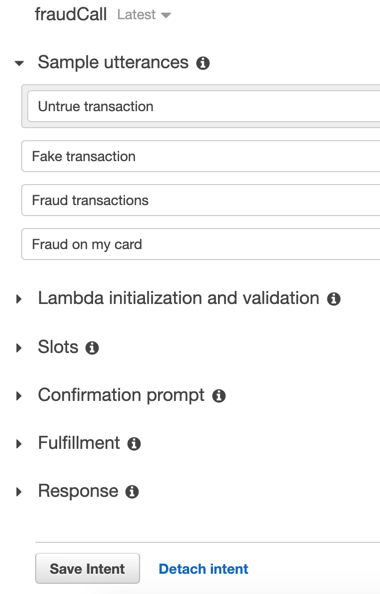
1. Go to your Amazon management console. Search for **Amazon Lex** (which is available from Singapore region).



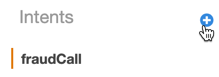
1. Go ahead and click on **Get Started.**
2. Select **Custom bot**. Enter in your bot’s name. I have used the name **financeBot** here in this example. Use any output voice you prefer and a session timeout. Select **No** for COPPA. Then click on the Create button.



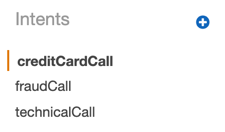
1. Click on the + Create Intent button to create your first intent. Select Create Intent.
2. We will be creating 3 intents in total. We want our caller to say what the call is about instead of using DTMF. This will replace the DTMP menu option in the contact flow we have created earlier. Enter in an intent name of **fraudCall.**
3. 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!



1. Do remember to **Save the Intent**.
2. Go ahead and create the other two intents.



1. 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.
2. For the last intent, name it **creditCardCall**. And do the same as step 9.

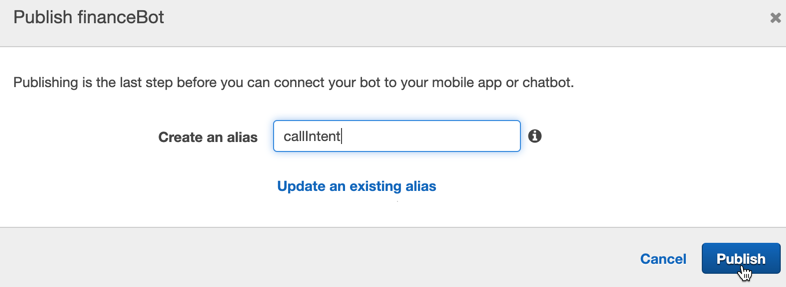


1. On the top right-hand side of the page. You can see both the Build and Public buttons.

Go ahead to **Build** the bot.



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



1. It will give you a successful message and you’re all set!

# Part 9.1: Leveraging your

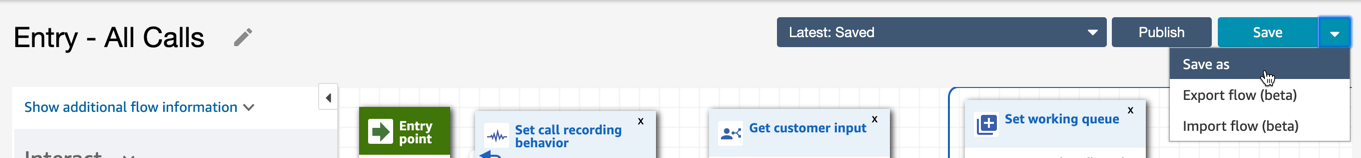
# Amazon Lex Bot

## 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.

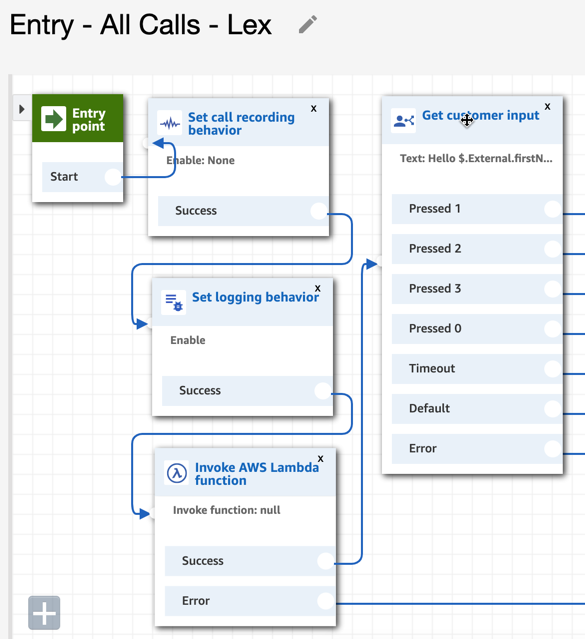


1. Go in to your management console. Click in to **Contact flows** as shown above. Add in the bot you have just published. And importantly, click on **+ Add Lex Bot**.
2. Go into your existing contact flow. Save this flow as a different name **Entry – All Calls – Lex** or some other name to differentiate 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.



1. In your first **Get customer input** block, click into it and change the text to:

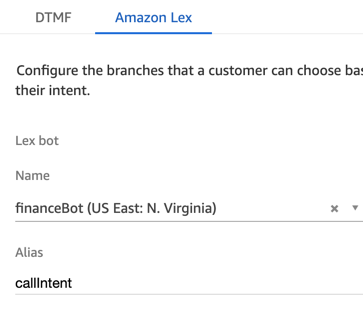
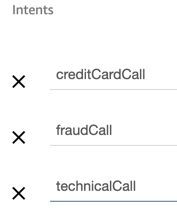
**Hello $.External.firstName, how can we help you today? For general credit card enquiries, please say credit card. If you believe you are a victim of fraud, please say fraud transactions. For technical assistance, please say technical help. For all other enquiries, please hold.**



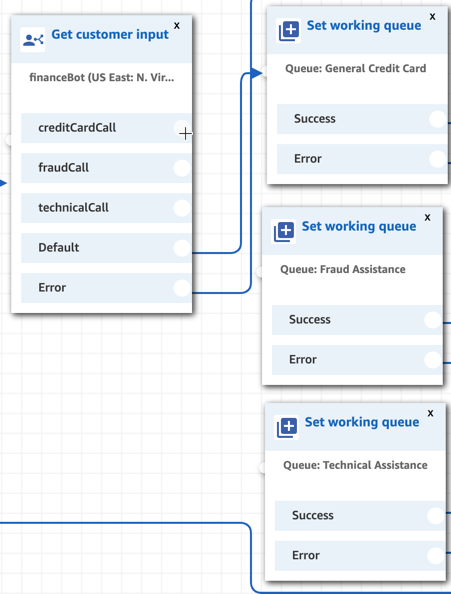
1. Scroll down the edit section of the block. You will see **Amazon Lex** as another tab beside DTMF. Click into it. Your bot should appear in the picklist. Select it.

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.



1. Go ahead and connect the nodes to the right blocks as it was for DTMF.



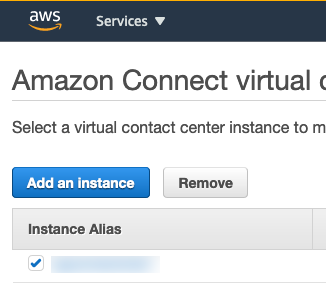
1. Publish it. And test it out!

# Part x: Deleting account

## 

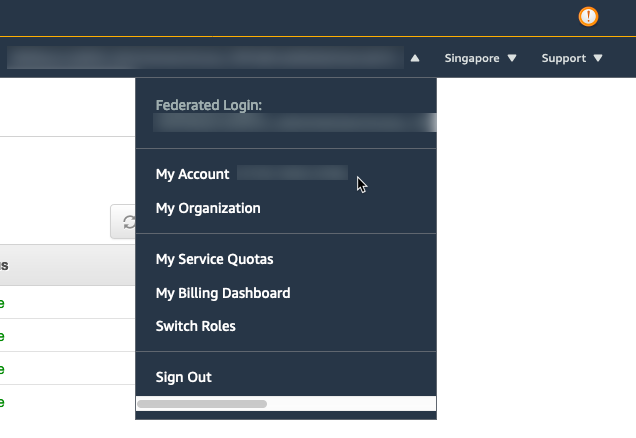
## Objective: Delete your account to prevent any incidental costs.

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.

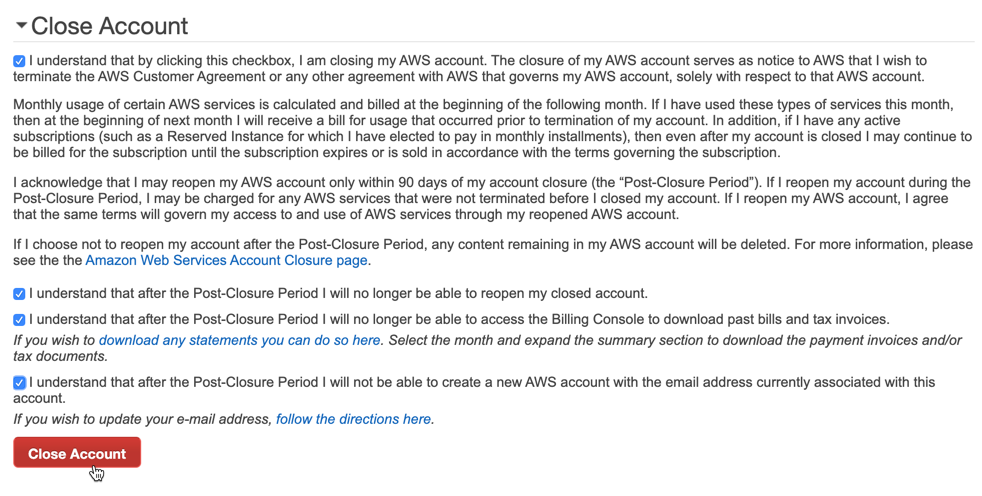


**Note:** IF you would like to delete your entire AWS account.

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



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



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