# Scheduling Callback in Amazon Connect - Technical Document

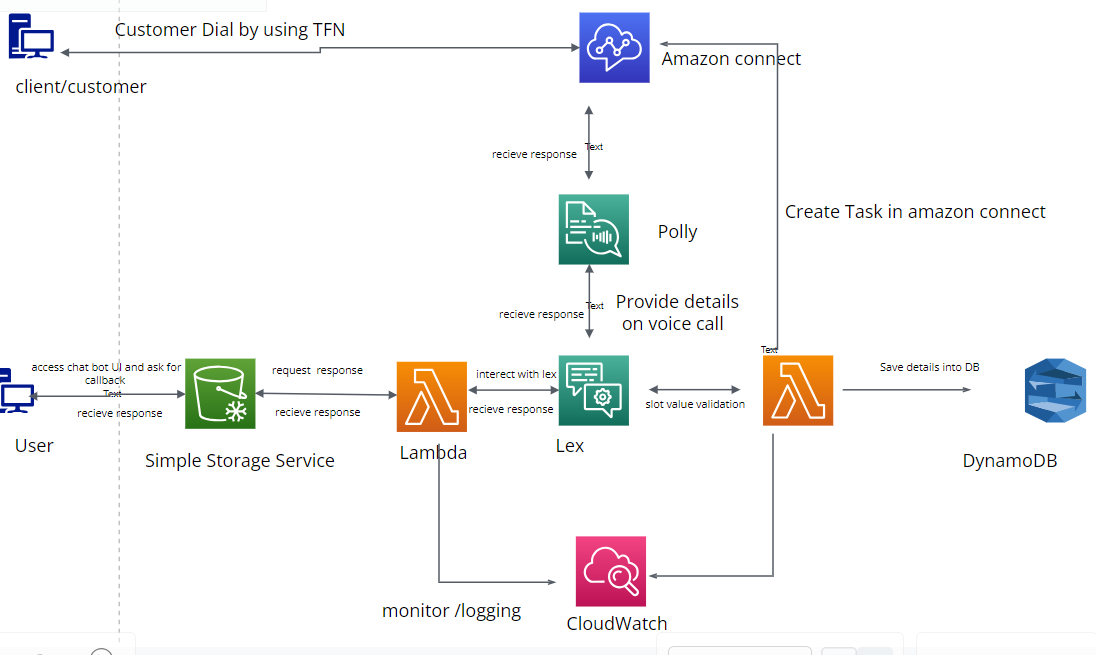
|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1.0** | **Mithilesh Singh** | **21/10/2024** | Initial Version |

|  |
| --- |
| Content |
| 1.Purpose |
| 2. Architecture |
| 3. Callback Implementation |
| 4. Process to hosting webpage in S3 |
| 5. Hosting Lex bot into Webpage |
| 6. Process to schedule, Receive callback |

# Purpose:

The purpose of this project is to enable customers to schedule a callback through Amazon Connect using Amazon Lex and AWS Lambda, with the callback request data being stored in DynamoDB. This setup aims to enhance customer service efficiency by allowing customers to request callbacks at their convenience.

# Architecture



## 2.1 Process of callback.

### 2.1.1Process on voice Call:

1. **Customer Interaction with Amazon Connect**:
   * The customer initiates a call to your Amazon Connect contact center.
   * The contact flow in Amazon Connect uses Amazon Lex to gather input from the customer, such as their preferred callback time and phone number.
2. **Input Validation with AWS Lambda**:
   * Amazon Lex captures the customer’s input and triggers an AWS Lambda function.
   * The Lambda function validates the input (e.g., checking if the preferred callback time is within business hours).
3. **Storing Details in DynamoDB**:
   * Once validated, the Lambda function stores the customer’s details (phone number, preferred callback time, etc.) in an Amazon DynamoDB table.
4. **Creating a Task in Amazon Connect**:
   * Another Lambda function can be used to create a task in Amazon Connect. This task will be scheduled to start at the customer’s preferred callback time.
   * The task can be configured to initiate a callback to the customer at the specified time.

Here’s a more detailed breakdown of each step:

**1. Customer Interaction with Amazon Connect**

* **Contact Flow**: Design a contact flow in Amazon Connect that uses Amazon Lex to interact with the customer.
* **Amazon Lex Bot**: Create a Lex bot with intents to capture the customer’s phone number and preferred callback time.

**2. Input Validation with AWS Lambda**

* **Lambda Function**: Write a Lambda function that validates the input from Lex. For example, it can check if the preferred callback time is within your business hours.
* **Integration**: Integrate the Lambda function with the Lex bot so that it gets triggered when the customer provides their input.

**3. Storing Details in DynamoDB**

* **DynamoDB Table**: Create a DynamoDB table to store the customer’s details.
* **Lambda Function**: Modify the Lambda function to save the validated details into the DynamoDB table.

**4. Creating a Task in Amazon Connect**

* **Task Creation**: Use another Lambda function to create a task in Amazon Connect. This task will be scheduled to start at the preferred callback time.
* **Task will start at given time and agent will accept the task , Copy phone number and make outbound call.**

### 2.1.2 Process With chat bot:

**Customer Interaction with the Chatbot**

* **Chat Interface**: Customers interact with the chatbot on your S3-hosted web page. The chatbot collects necessary details like phone number and preferred callback Date and time.

**Handling Responses with AWS Lambda**

* **Lambda Function**: a Lambda function that processes the input from the chat interface. This function will validate the input and invoke Amazon Lex for further processing and return response to UI.
* **Invoke Lex**: The Lambda function sends the customer’s input to Amazon Lex for intent recognition and further interaction.

**Validating Input with Amazon Lex and AWS Lambda**

* **Amazon Lex** a Lex bot with intents to capture and validate the customer’s phone number and preferred callback time.
* **Lambda Validation**: Use a Lambda function to validate the input received from Lex. For example, ensure the preferred callback time is within business hours.

**Storing Details in DynamoDB**

* **DynamoDB Table**: Set up a DynamoDB table to store the customer’s details.
* **Lambda Function**: Modify the Lambda function to save the validated details into the DynamoDB table.

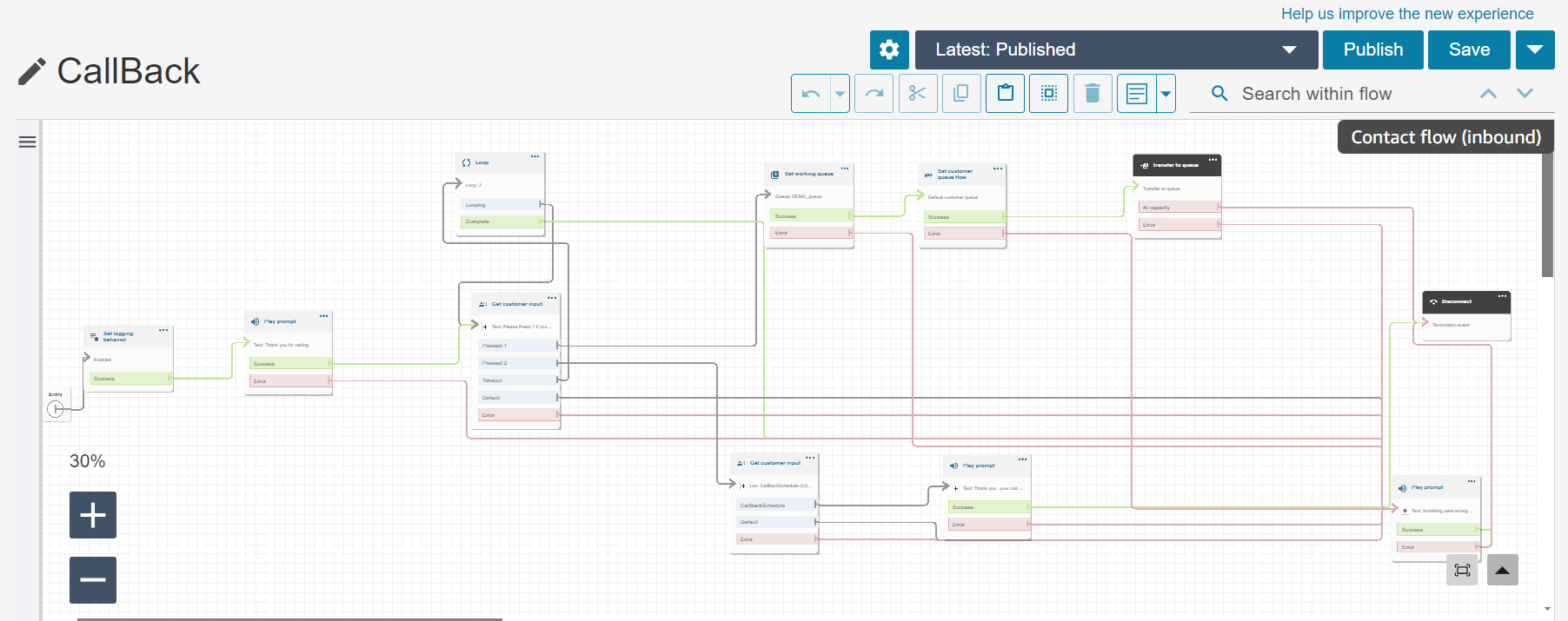
**Creating a Task in Amazon Connect**

* **Task Creation**: Use another Lambda function to create a task in Amazon Connect. This task will be scheduled to start at the customer’s preferred callback time.
* **Task will start at given time and agent will accept the task , Copy phone number and make outbound call.**

# Callback Implementation:

## 3.1Amazon Connect

1. **Create a Contact Flow**: Design a contact flow that includes a prompt for the caller to request a callback.



1. **Add Lex Bot**: Integrate the Lex bot into the contact flow.

Add lex bot into amazon flow.

A screenshot of a computer

Description automatically generated

Add bot name and alias into flows

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Select input from lex and select bot name and alias added earlier. Provide Prompt text and

add lex intent .

1. **Create Callback queue with callback ID.**

A screenshot of a computer

Description automatically generated

1. **Define routing profile.**

A screenshot of a computer

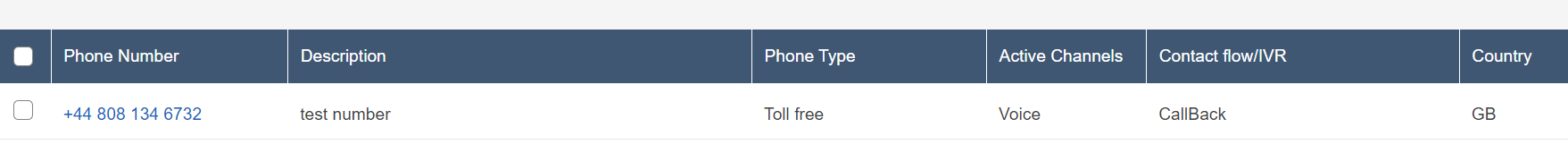
Description automatically generated

1. **Claim phone number**

A screenshot of a phone number

Description automatically generated

Claimed Number:



A screenshot of a computer

Description automatically generated

1. **Create quick connect with ca**llback queue.

A screenshot of a computer

Description automatically generated

## 3.2 Amazon Lex

1. **Create a Lex Bot**: Define a new bot with an intent (e.g., Schedule Callback).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Add utterence

A screenshot of a computer

Description automatically generated

1. **Define Slots**: Add slots for first name , last name Phone Number, Date and Preferred Time

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Confirmation**:

A screenshot of a computer

Description automatically generated

1. **Fulfillment**: Set the fulfillment to be handled by an AWS Lambda function.

## 3.3 AWS Lambda

1. **Create a Lambda Function**: Write a function to handle the Lex intent and save data to DynamoDB.

A screenshot of a computer

Description automatically generated

### 3.3.1 Amazon connect task API to create task in amazon connect.

A screenshot of a computer code

Description automatically generated

### 3.3.2Lambda code Saving details into DB:

A screen shot of a computer code

Description automatically generated

1. **Permissions**: Ensure the Lambda function has the necessary permissions to write to DynamoDB and amazon connect.

A screenshot of a computer

Description automatically generated

### 3.3.3 IAM Roles and Permissions

1. **Create IAM Roles**: Ensure that the Lambda function has the necessary permissions to interact with DynamoDB and that Lex can invoke the Lambda function and have amazon connect read and write permissions.

## 3.4 AWS DynamoDB

A screenshot of a computer

Description automatically generated

# 4. Process of Hosting a webpage into S3:

## **4.1**. Create an S3 Bucket

1. **Sign in to the AWS Management Console** and open the Amazon S3 console.
2. **Create a Bucket**:
   * Click on “Create bucket”.
   * Enter a unique bucket name (e.g., my-website-bucket).
   * Choose a region close to your users for better performance.
   * Click “Create bucket” to finalize.

## 4.2. Upload Your Website Files

1. **Select Your Bucket**: Click on the bucket name you just created.
2. **Upload Files**: Click on “Upload” and add your website files (e.g., index.html, styles.css).

## 4.3. Enable Static Website Hosting

1. **Go to Bucket Properties**: In the S3 console, select your bucket and go to the “Properties” tab.
2. **Enable Static Website Hosting**:
   * Scroll down to “Static website hosting” and click “Edit”.
   * Select “Use this bucket to host a website”.
   * Enter the name of your index document (e.g., index.html).
   * Optionally, specify an error document (e.g., error.html).
   * Click “Save changes”.

## 4.4. Configure Bucket Permissions

1. **Edit Block Public Access Settings**:
   * Go to the “Permissions” tab of your bucket.
   * Click “Edit” under “Block public access”.
   * Uncheck “Block all public access” and save changes.
2. **Set Bucket Policy**:
   * In the “Permissions” tab, scroll down to “Bucket policy”.
   * Add a policy to allow public read access:

**JSON**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::my-website-bucket/\*"

}

]

}

AI-generated code. Review and use carefully. [More info on FAQ](https://www.bing.com/new#faq).

* + Replace my-website-bucket with your bucket name and save the policy.

## 4.5. Access Your Website

1. **Website Endpoint**: After enabling static website hosting, you will see an endpoint URL in the “Static website hosting” section. This is the URL where your website is hosted (e.g., http://my-website-bucket.s3-website-us-east-1.amazonaws.com).

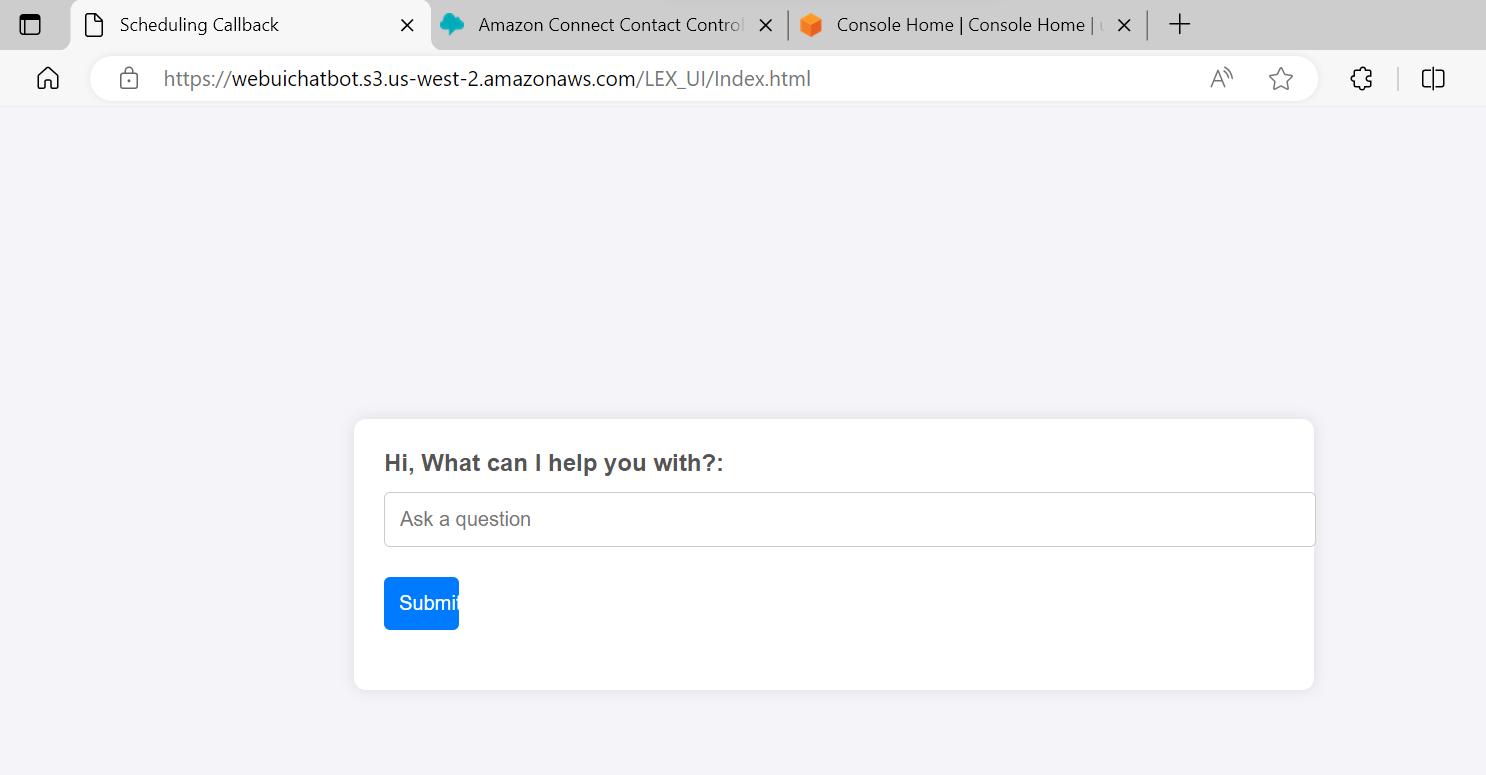
# Hosting Lex bot into webpage:

* 1. Create a webpage using Html, CSS and JS.
  2. Create a User in IAM in AWS console.
  3. Generate access key and secret access key
  4. Use this into web page and invoke lambda from web page.
  5. Create Recognized\_text API from lambda

**A screen shot of a computer code

Description automatically generated**

# Process to Schedule, receive callback and make Outbound to Customer:

Input details from this UI. 

Task start.

A screenshot of a computer

Description automatically generated

Task accept:

A screenshot of a computer

Description automatically generated

Copy the phone Number from Customer details and paste into number pad and click on call.

A screenshot of a computer

Description automatically generated