Amazon lex chat bot Project

(AWS)

J Raj kumar

Btech

Introduction

- In today's digital era, businesses are constantly striving to provide seamless and efficient customer support and engagement. One of the most effective ways to achieve this is through the use of chatbots. Chatbots have revolutionized customer interactions by offering automated conversational interfaces that can handle various user inquiries and tasks.
- In this project, we have leveraged the power of Amazon Lex, a sophisticated and highly scalable chatbot framework provided by Amazon Web Services (AWS). Amazon Lex utilizes advanced natural language processing (NLP) techniques to understand and interpret user inputs, enabling a more intuitive and human-like conversation.
- Let's explore how our Amazon Lex chatbot can transform the way we engage with our customers and elevate our customer support to new heights.

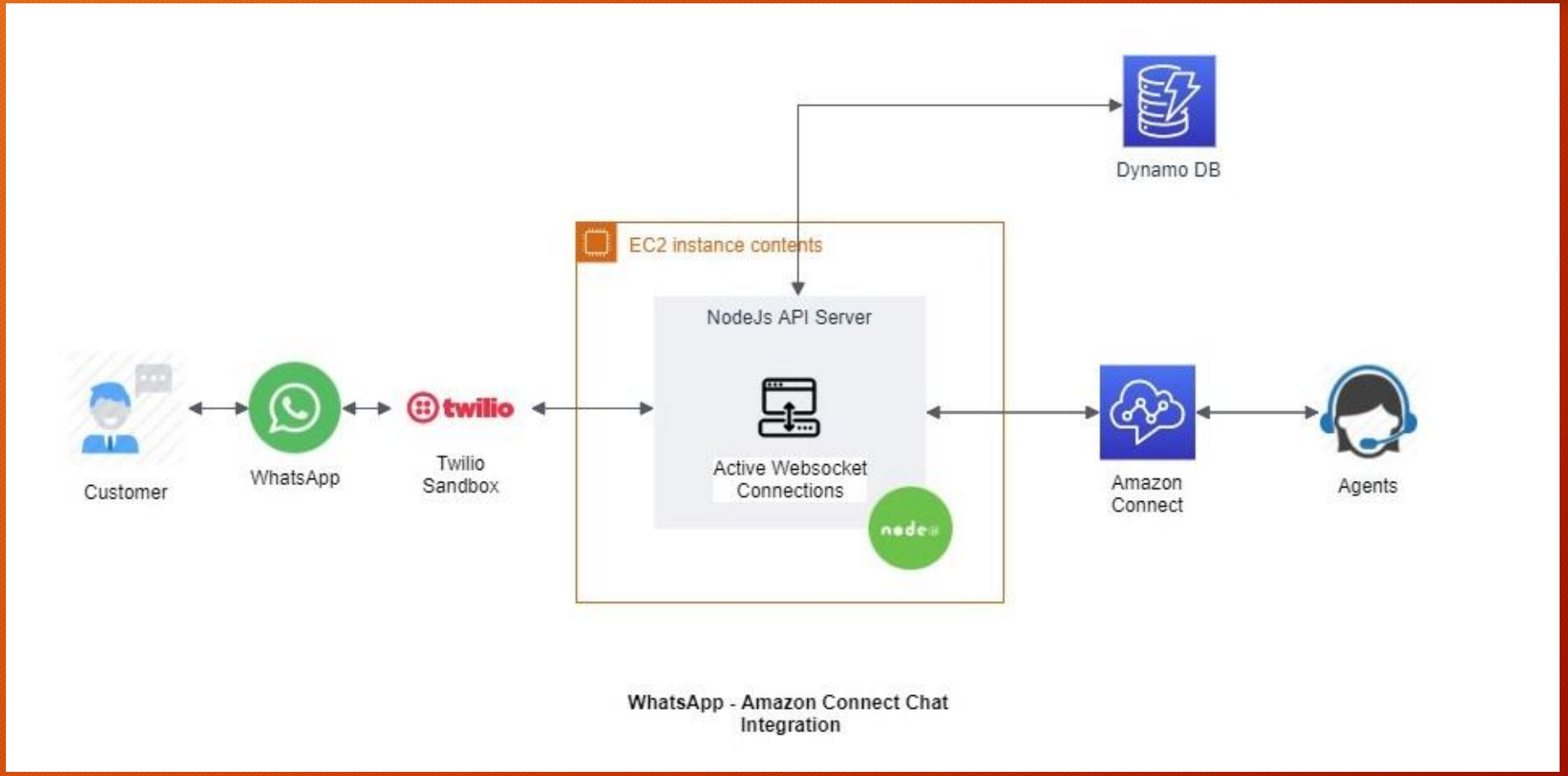
Objective

- The objective of our Amazon Lex chatbot project is to leverage the power of conversational AI to enhance customer support and engagement. We aim to achieve the following objectives:

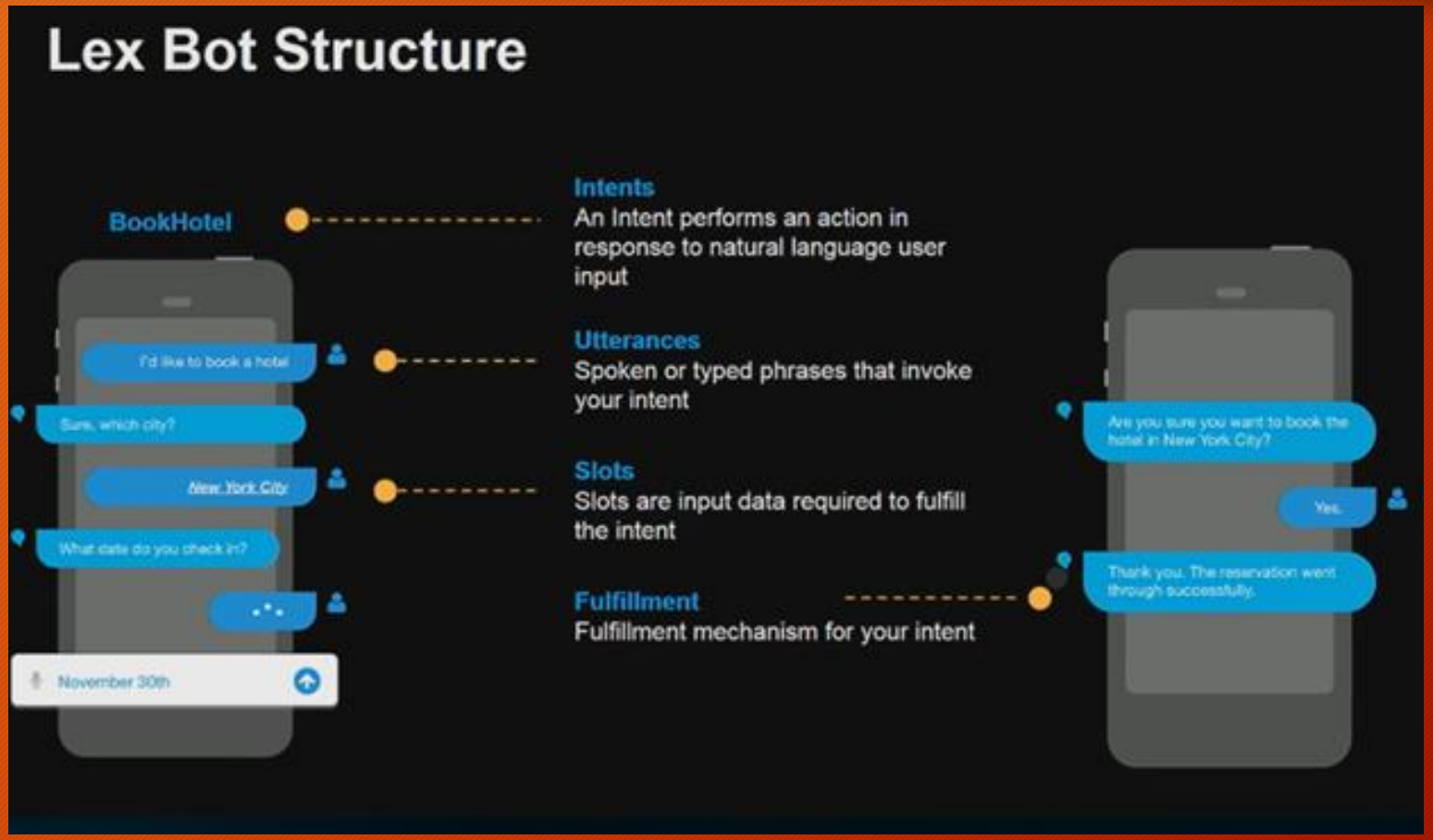
1. Improve customer experience through a user-friendly conversational interface.
2. Streamline support processes by automating routine tasks and providing instant responses.
3. Increase availability and responsiveness with 24/7 customer support.
4. Scale customer support to handle a larger volume of customer conversations.
5. Gather customer insights and feedback for continuous improvement.
6. Enable multichannel support across websites, messaging apps, and voice interfaces.

- These objectives collectively aim to enhance customer satisfaction, optimize support operations, and drive business growth by leveraging the capabilities of our Amazon Lex chatbot.

Integration with Whatsapp



Conversational AI Chatbots – The What, Why and Everything Else



Amazon Lex
What is intent?

- Within a chatbot, intent refers to the goal the customer has in mind when typing in a question or comment. While entity refers to the modifier the customer uses to describe their issue, intent is what they really mean.
- Can I order pizza
- I want to order a pizza

What is slot ?

- A slot is an information that Amazon lex needs to fulfill an intent. Each slot has a slot types.
- You can create your custom slot types or use built in slot types
- For example, the orderpizza intent requires slots such as pizza size and pizza type.

What is slotType?

- Each Slot has a type.
- You can create your slot type or use buit-in slot types.
- For example you might creates and use the following slot types for the order pizza intent.
- Size small, medium and large
- Crust thick or thin

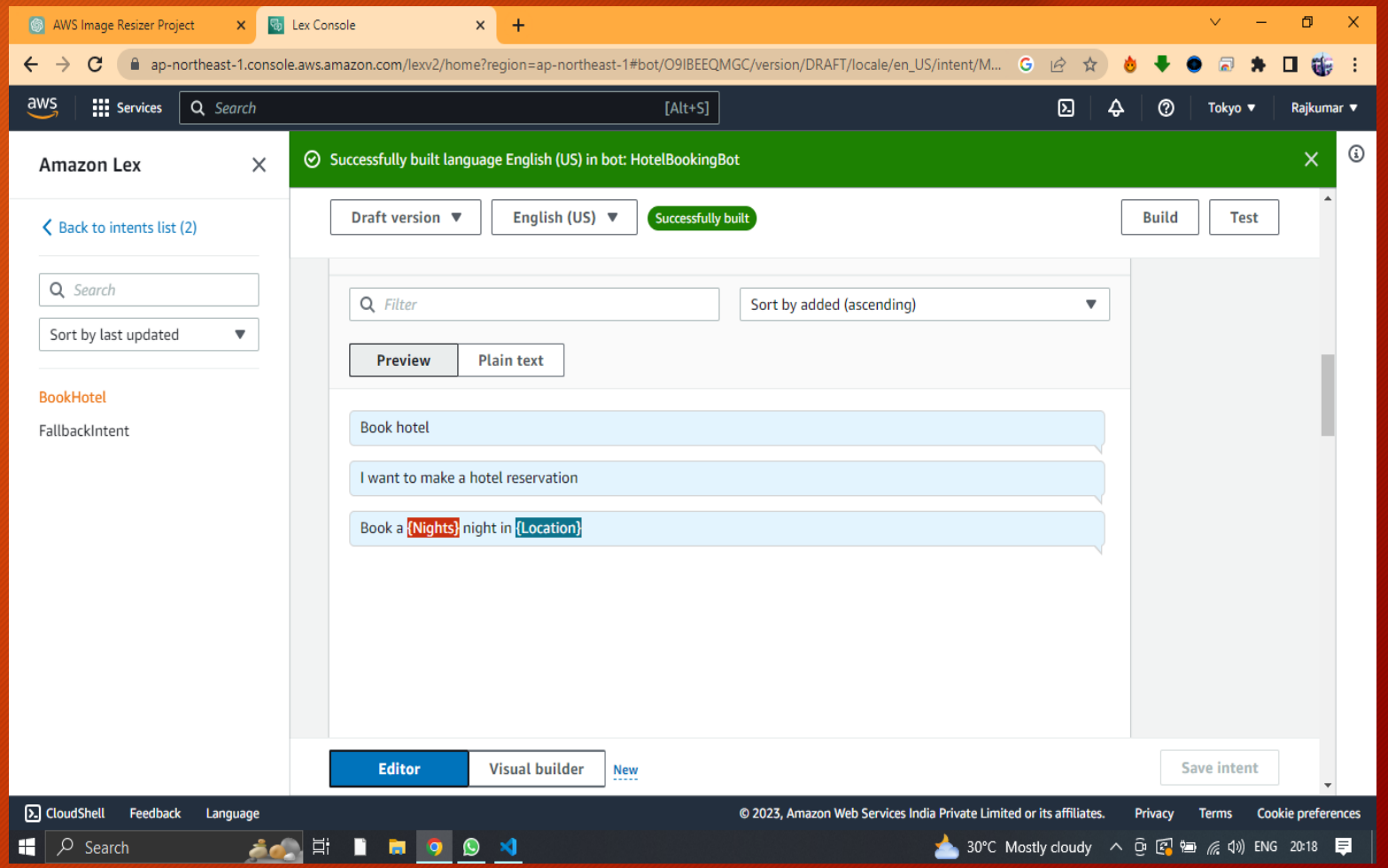
Hotel Booking Chatbot

- Intent:

  Book hotel

  I want to make a hotel reservation

  Book A {night} stay in {location}



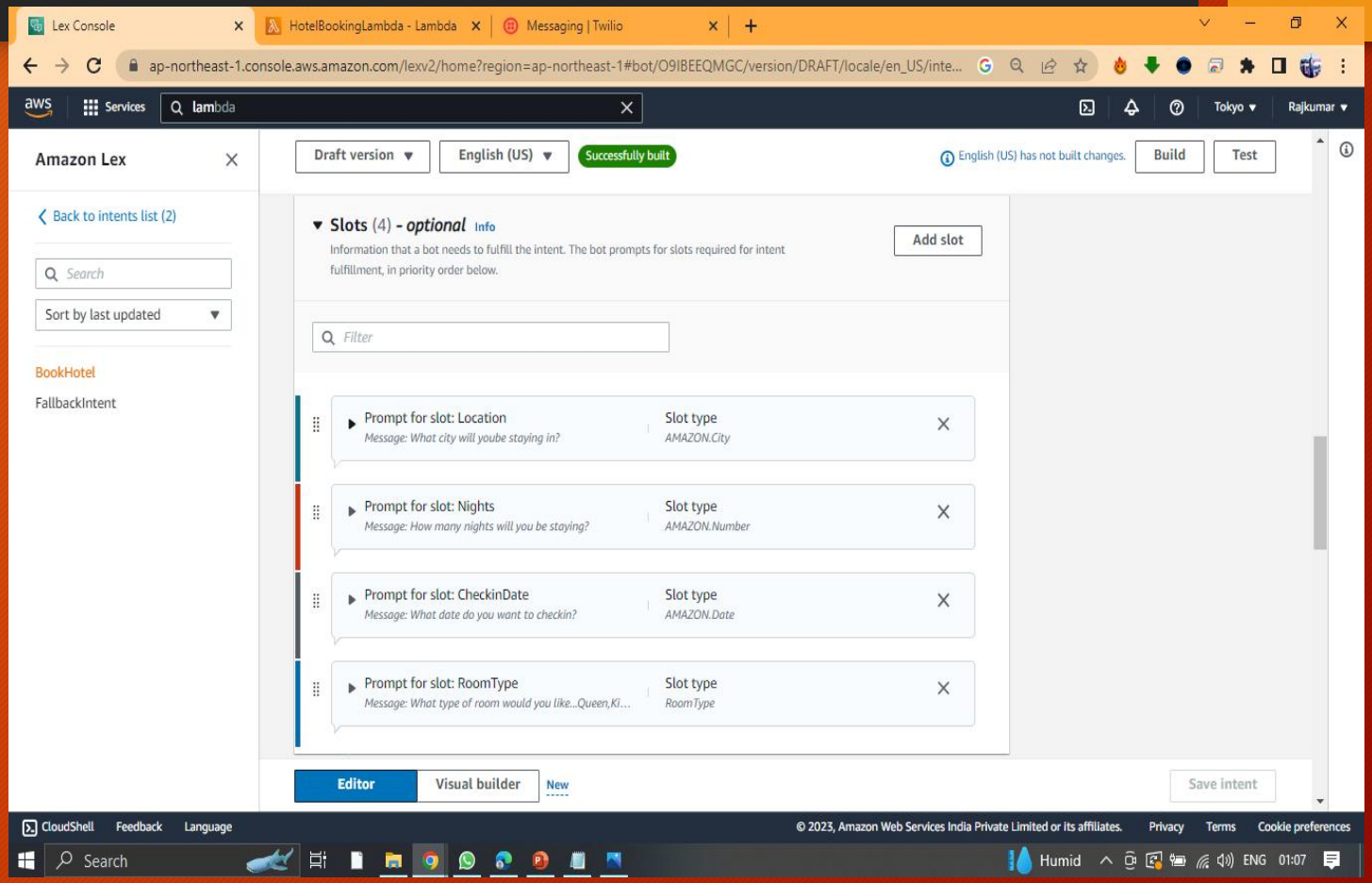
- **Slots:**

Location:

Prompts : What city will you be staying in?

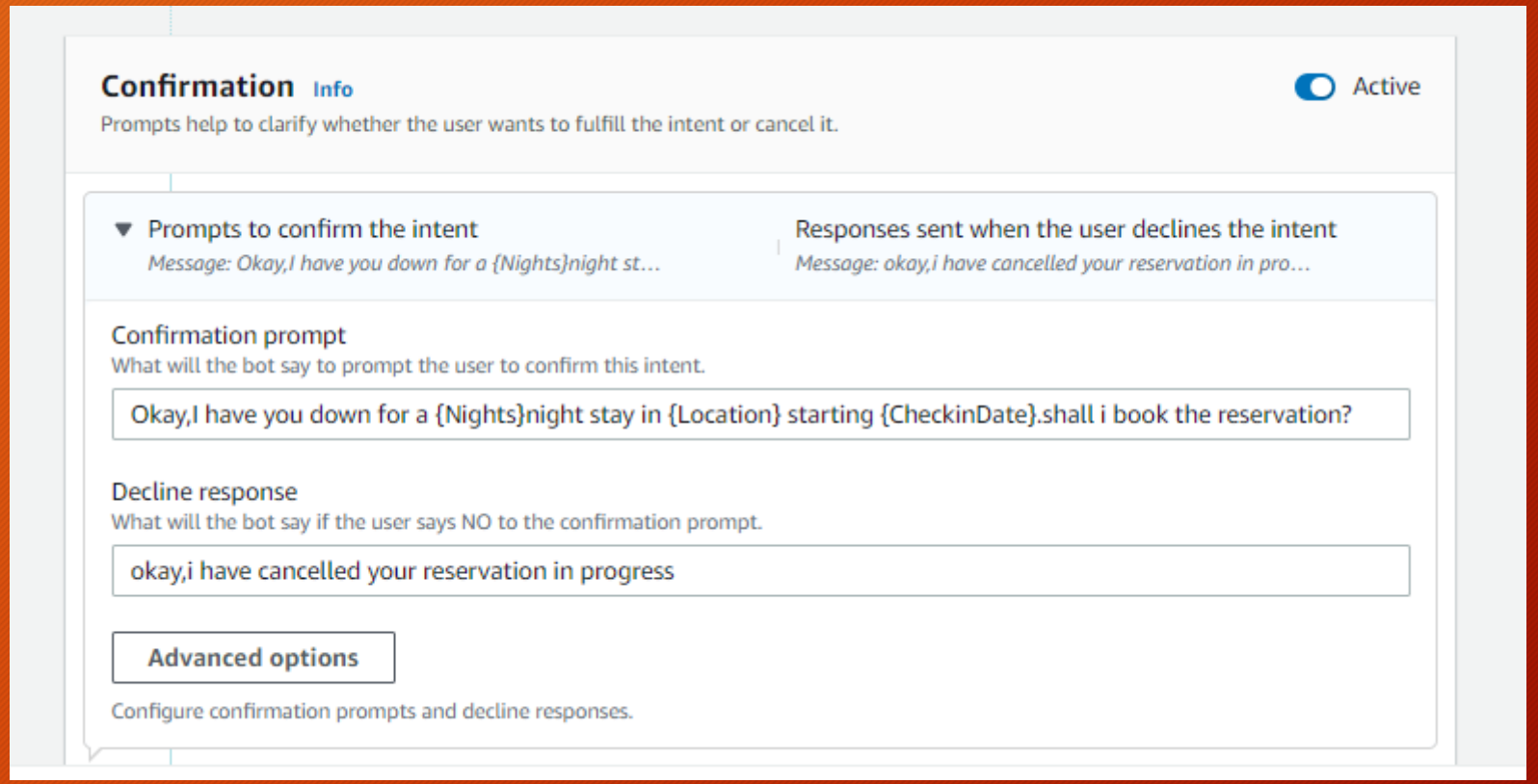
- **Checkin date:**
- **Prompts:**
- **What day do you want to check in?**

- **Nights:**
- **Prompts**
- **How many nights will you be staying?**
- **Roomtype:**
- **Prompts:**
- **What type of room would you like , Queen,King, Deluxe and Super Deluxe?**



Confirmation Prompt:

- Okay,I have youn down for a {Nights} night stay in{Location} starting {CheckIndate}.Shall I book the reservation?

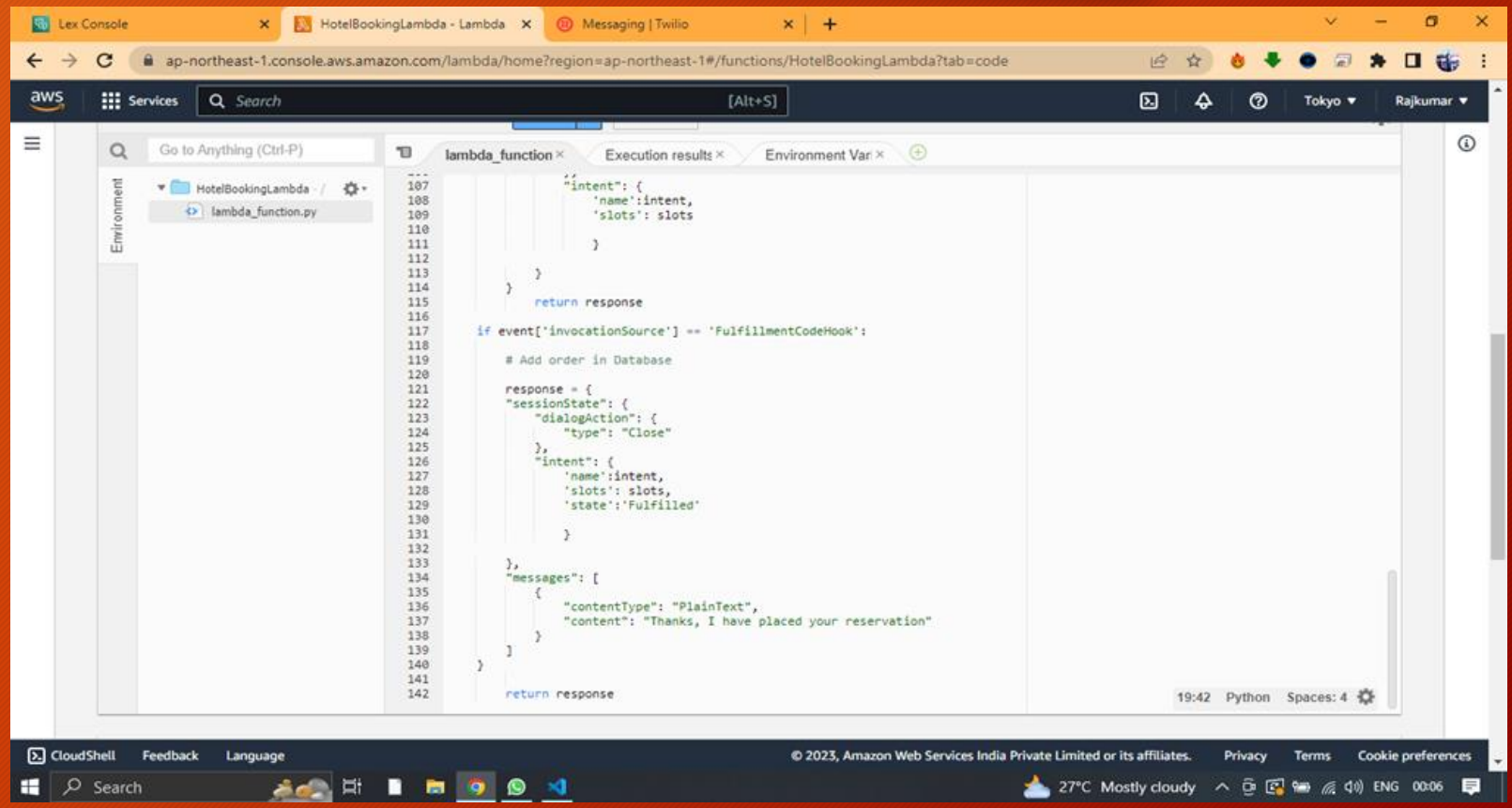


Code For Hotel Booking Chatbot

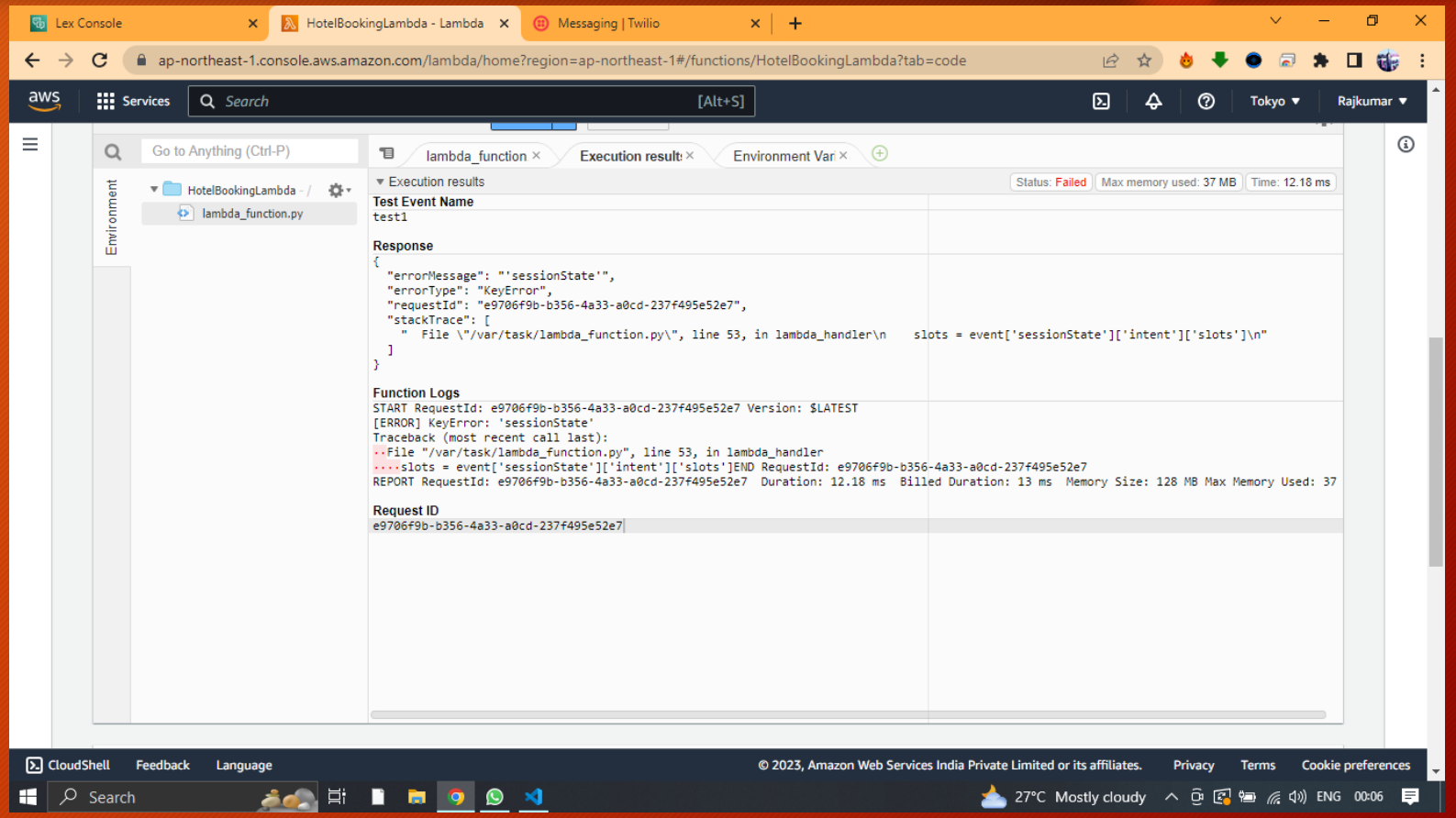
```
import json
import datetime
import time
def validate(slots):
    valid_cities = ['mumbai','delhi','banglore','hyderabad']
    if not slots['Location']:
        print("Inside Empty Location")
        return {
        'isValid': False,
        'violatedSlot': 'Location'
        }
    if slots['Location']['value']['originalValue'].lower() not in  valid_cities:
        print("Not Valide location")
        return {
        'isValid': False,
        'violatedSlot': 'Location',
        'message': 'We currently  support only {} as a valid destination.?'.format(", ".join(valid_cities))
        }
```

```
    if not slots['CheckInDate']:
        return {
        'isValid': False,
        'violatedSlot': 'CheckInDate',}
    if not slots['Nights']:
        return {
        'isValid': False,
        'violatedSlot': 'Nights'}
    if not slots['RoomType']:
        return {
        'isValid': False,
        'violatedSlot': 'RoomType'}
    return {'isValid': True}
def lambda_handler(event, context):
    slots = event['sessionState']['intent']['slots']
    intent = event['sessionState']['intent']['name']
    print(event['invocationSource'])
    print(slots)
    print(intent)
    validation_result = validate(event['sessionState']['intent']['slots'])
    if event['invocationSource'] == 'DialogCodeHook':
        if not validation_result['isValid']:
            if 'message' in validation_result:
                response = {
```

```
"sessionState": {
                "dialogAction": {
                    'slotToElicit':validation_result['violatedSlot'],
                    "type": "ElicitSlot"},
                "intent": {
                    'name':intent,
                    'slots': slots}},
            "messages": [
                {
                    "contentType": "PlainText",
                    "content": validation_result['message']}]
            }
        else:
            response = {
            "sessionState": {
                "dialogAction": {
                    'slotToElicit':validation_result['violatedSlot'],
                    "type": "ElicitSlot" },
                "intent": {
                    'name':intent,
                    'slots': slots}}}
            return response
        else:
            response = {
```

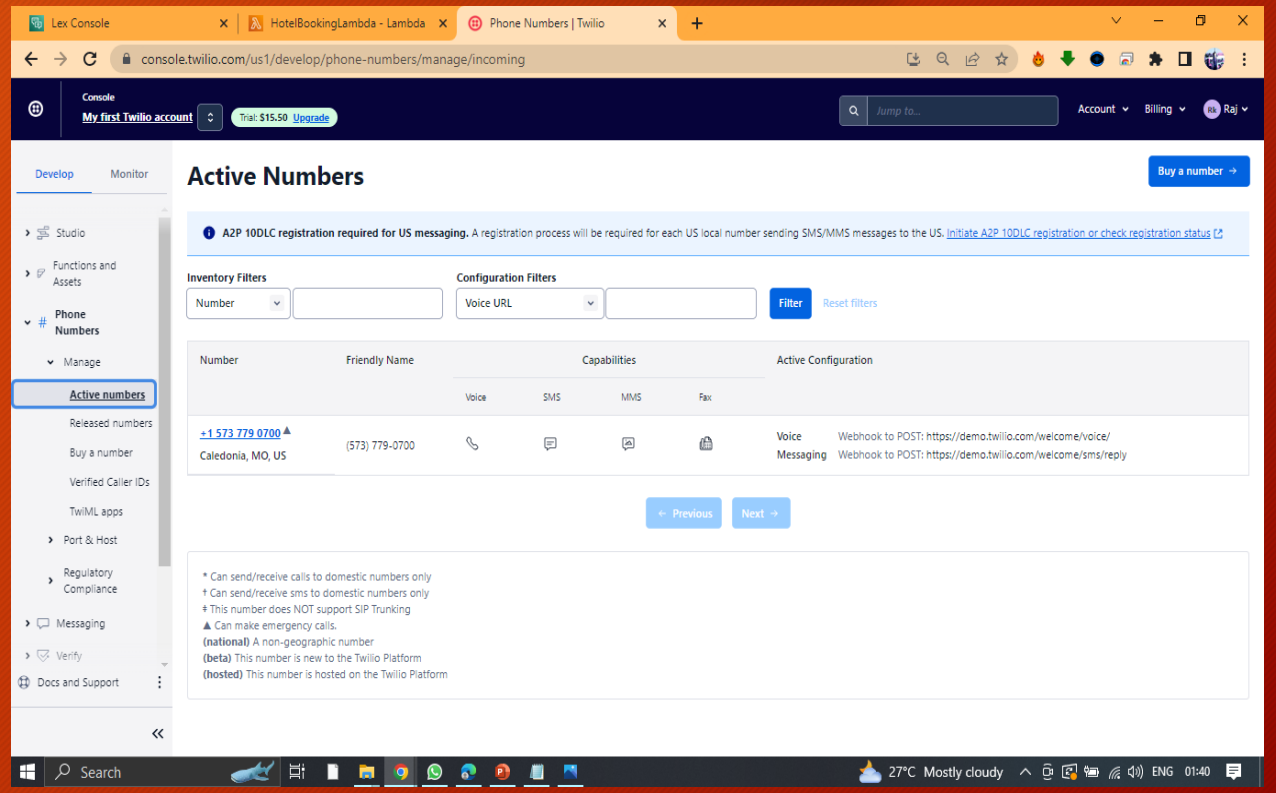


```
        "sessionState": {
            "dialogAction": {
                "type": "Delegate"
            },
            "intent": {
                'name':intent,
                'slots': slots } }}
        return response
    if event['invocationSource'] == 'FulfillmentCodeHook':
        response = {
        "sessionState": {
            "dialogAction": {
                "type": "Close"},
            "intent": {
                'name':intent,
                'slots': slots,
                'state':'Fulfilled' } },
        "messages": [
            {
                "contentType": "PlainText",
                "content": "Thanks, I have placed your reservation"
            }]}
        return response
```



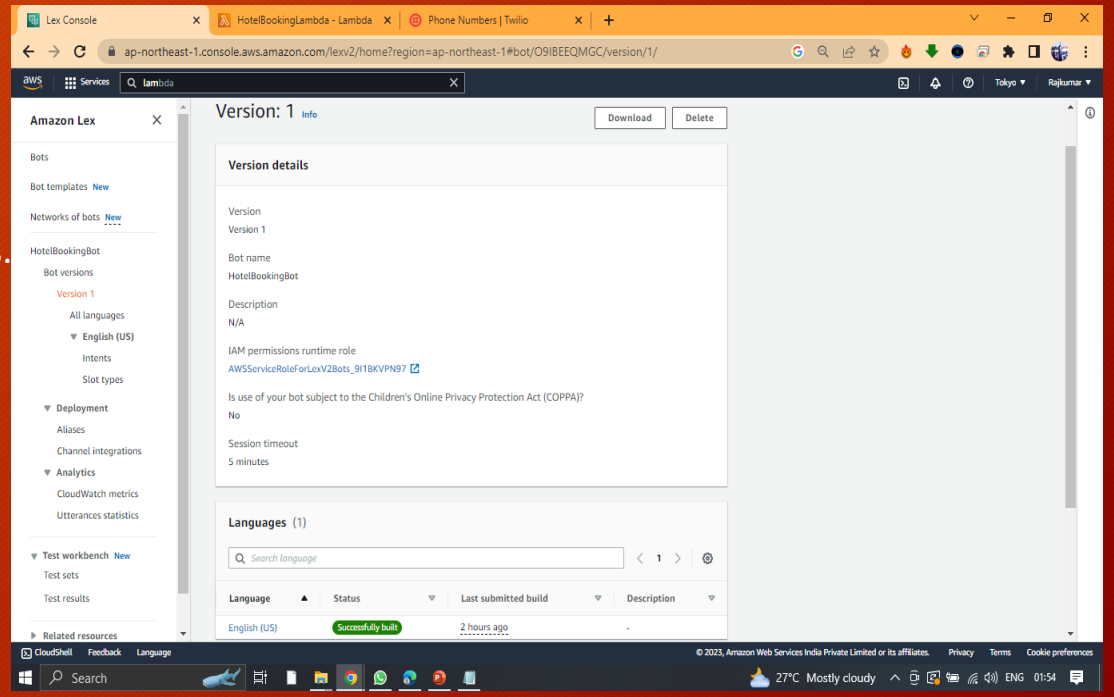
Twilio Account and number

- Twilio is a cloud communications platform that provides developers with a set of tools and APIs to easily integrate messaging, voice, and video capabilities into their applications. It enables businesses to build and scale robust communication solutions, including SMS and MMS messaging, voice calls, video chats, and more.



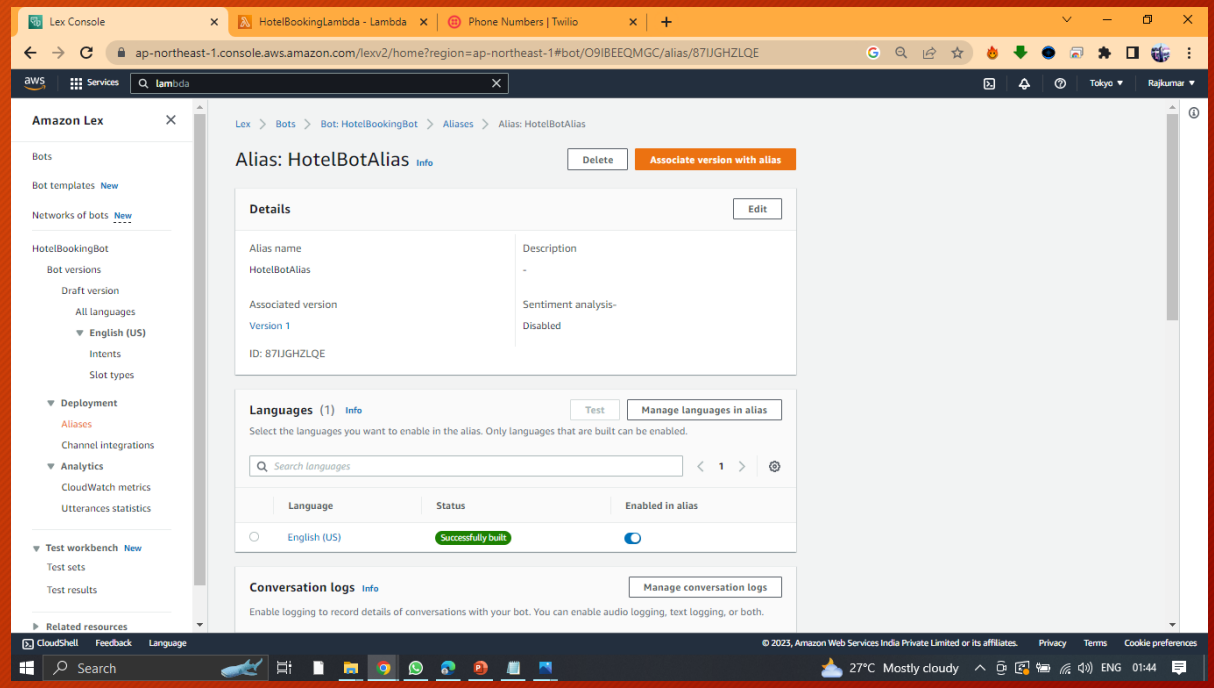
Create bot version and alias

- Bot Version: Twilio Communication Assistant
- Alias: TwilioBot
- Twilio Communication Assistant (TwilioBot) automates messaging tasks for businesses.
- It can handle SMS notifications, order updates, and appointment reminders.
- TwilioBot serves as an interactive voice response (IVR) system, allowing callers to interact with a menu.
- It can route calls, provide automated responses, and gather caller information.
- TwilioBot offers call routing and forwarding capabilities based on predefined rules.
- It enables two-factor authentication (2FA) via SMS or voice calls.
- TwilioBot incorporates natural language processing (NLP) to understand user queries.
- It supports message templating for personalized bulk messaging.
- TwilioBot provides analytics and reporting features for communication metrics.
- It integrates with third-party systems such as CRM and help desk software.
- TwilioBot is scalable, reliable, and built on a developer-friendly platform.



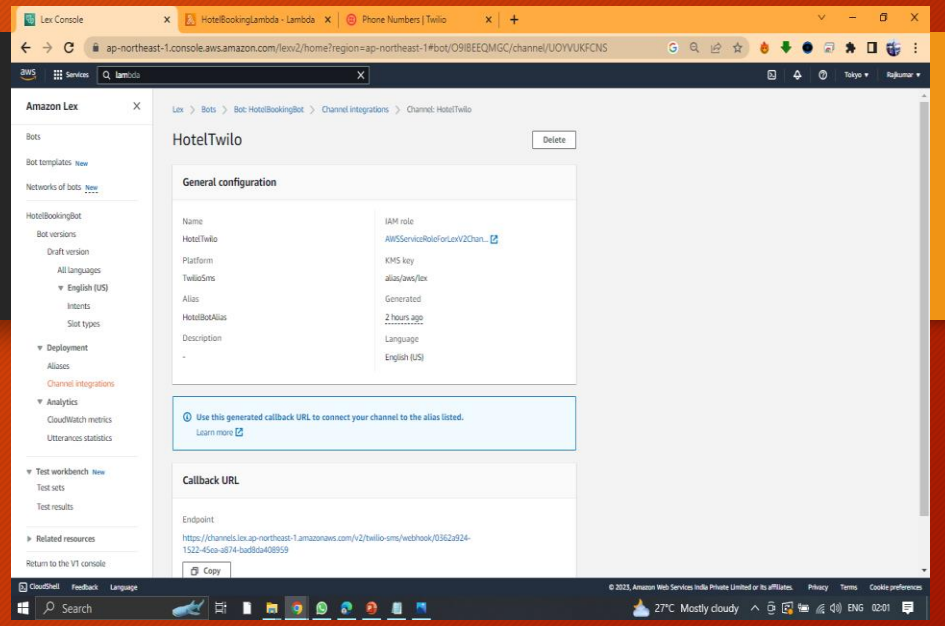
Create Amazon Lex with Twilio

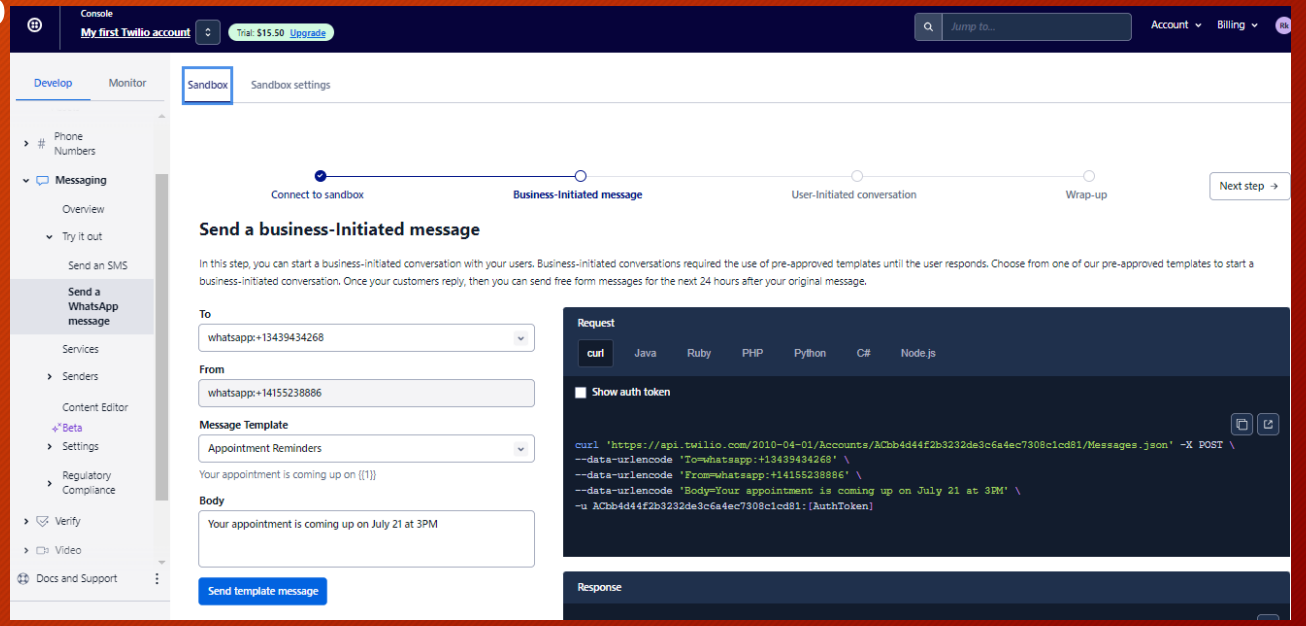
- Amazon Lex is a conversational AI service that enables developers to build chatbots and voice assistants.
- Twilio can be integrated with Amazon Lex to leverage its communication capabilities.
- By combining Amazon Lex and Twilio, businesses can create interactive chatbots and voice assistants that can communicate with users through SMS, MMS, voice calls, and other channels.
- Developers can use Amazon Lex to design conversational flows, define intents, and create responses for the chatbot or voice assistant.
- Twilio can handle the communication aspect, allowing the chatbot or voice assistant to send and receive messages or make and receive phone calls.
- Integration with Twilio enables the chatbot or voice assistant to engage with users in real-time, providing instant responses and support.



Integrate WhatsApp with Twilio

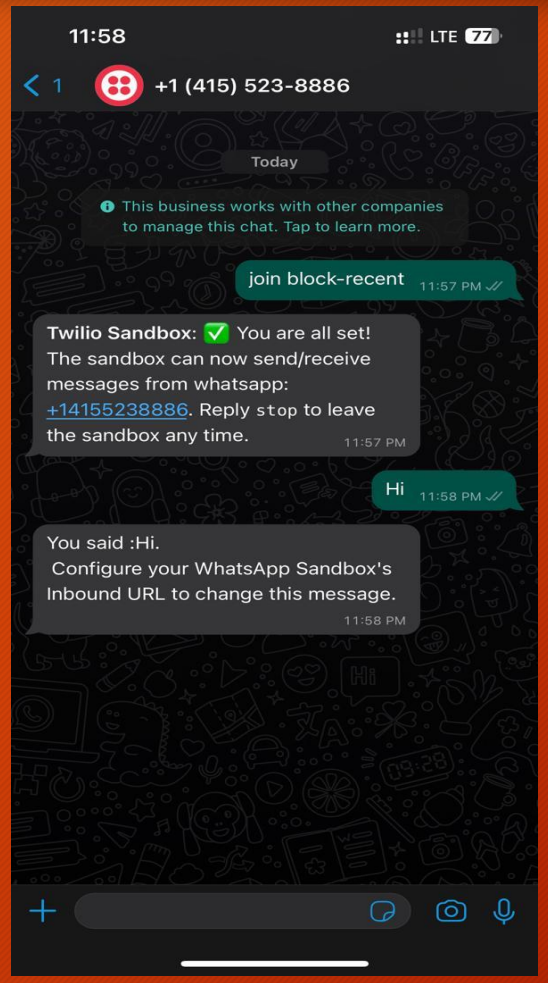
- Twilio enables businesses to integrate WhatsApp, a popular messaging platform, into their communication workflows.
- Integration with Twilio allows businesses to send and receive WhatsApp messages using the Twilio API.
- Through Twilio's WhatsApp integration, businesses can leverage the rich features of WhatsApp, including text messages, images, videos, documents, and location sharing.
- Twilio provides a unified interface to manage WhatsApp communication alongside other channels, such as SMS, voice calls, and chatbots, streamlining customer engagement and support.

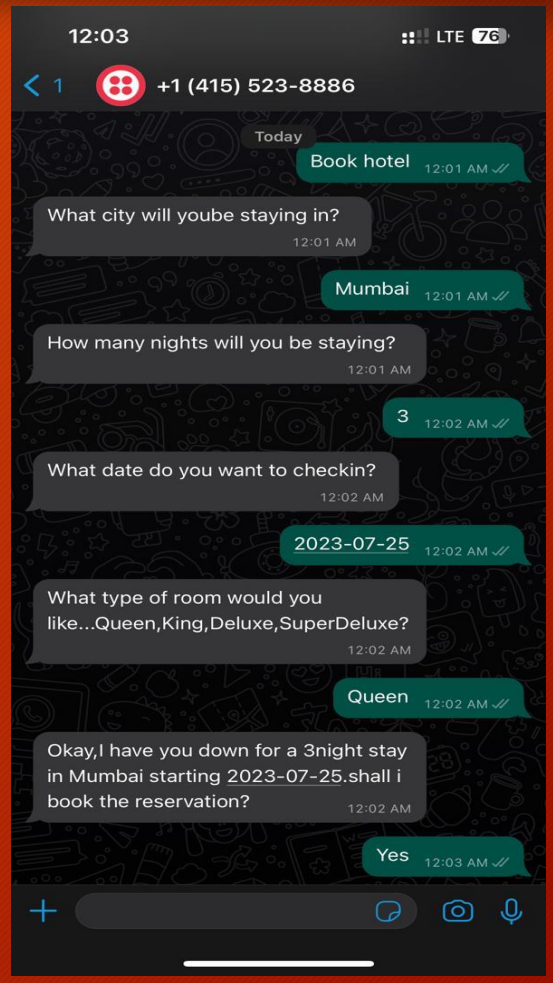




Interact from WhatsApp to Amazon Lex Chatbot

- By integrating WhatsApp with Amazon Lex chatbot, businesses can enable users to interact with the chatbot directly from their WhatsApp messaging platform.
- Users can initiate conversations with the Amazon Lex chatbot on WhatsApp, ask questions, seek assistance, and receive automated responses or personalized information.





Conclusion

- In conclusion, Amazon Lex chatbot is a versatile and user-friendly solution for businesses seeking to enhance their customer support and engagement. With its natural language understanding capabilities and seamless integration options, Amazon Lex allows businesses to create intelligent and interactive chatbot experiences across multiple channels. By automating customer interactions, providing instant responses, and offering personalized support, Amazon Lex enables businesses to streamline processes, improve efficiency, and deliver exceptional user experiences. With its developer-friendly platform and extensive integration capabilities, Amazon Lex empowers businesses to build and deploy chatbots quickly, making it an invaluable tool for delivering efficient and effective customer service.

Thank You