

Build and Deploy Chatbots using Amazon Lex

Creating a dental chatbot with **Amazon Lex, AWS Lambda, Cognito, IAM, and S3** involves multiple AWS services that work together to manage chatbot interactions, secure authentication, storage, and permissions.

Here's a step-by-step guide to get you started.

Step 1 : Set Up Amazon Lex for the Chatbot

1. **Create the Bot**:

- In the AWS Lex Console, create a new bot and give it a descriptive name (e.g., "DentalChatBot").
- Choose the preferred language and set up any additional options like voice if you need text-to-speech.

2. **Define Intents**:

- Intents are actions the chatbot can perform, like "ScheduleAppointment" or "GetDentalAdvice."

- For each intent, add sample **utterances** (phrases users may say) such as:
 - "I want to schedule a dental appointment"
 - "Tell me about dental hygiene"

3. **Set Up Slots**:

- Slots are variables or pieces of information required to fulfill the intent. For example, for scheduling an appointment, you might add slots for:

- **Appointment Type** (root canal)
- **Date** (`AMAZON.DATE` slot type)
- **Time** (`AMAZON.TIME`)

4. **Fulfillment**:

- Set up a Lambda function (explained in Step 2) to handle backend processing and scheduling logic for the chatbot.

5. **Test the Bot**:

- In the Lex console, test the bot to make sure it captures intents and slots correctly.

The screenshot shows the Lex V2 Console interface. On the left, a sidebar menu includes 'Amazon Lex' (selected), 'Bots' (highlighted in orange), 'Bot templates New', 'Networks of bots New', 'Test workbench New', 'Related resources', and 'Return to the V1 console'. The main content area has a header 'Bots (0) Info' with an 'Action' dropdown and a 'Create bot' button. Below is a search bar and a table with columns: Name, Description, Status, Latest Version, and Last updated. A message at the top right says 'No bots found' and 'Create bot'. At the bottom right, there's an 'Activate Windows' message with a link to settings.

Amazon Lex

Bots

Bot templates New

Networks of bots New

Test workbench New

Test sets

Test results

Related resources

Return to the V1 console

Bots (0) Info

Action Create bot

Search bots

Name	Description	Status	Latest Version	Last updated
No bots found				

Create bot

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

26°C Haze 01:35 27-10-2024

Amazon Lex

Bots

Intents

Slot types

Migration tool

Switch to the new Lex V2 console

Try the new Lex V2 Console

Build, deploy and manage your bots faster. The new console provides support for multiple languages in a bot, simplified versioning, interactive conversation flow, and other productivity tools. [Start building!](#)

Learn more.

Bots

Create Actions

Filter: Filter by Bot name

Name	Status	Locale	Last updated	Date
No records found.				

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

26°C Haze 01:35 27-10-2024

Amazon Lex enables any developer to build conversational chatbots quickly and easily. With Amazon Lex, no deep learning expertise is necessary—you just specify the basic conversational flow directly from the console, and then Amazon Lex manages the dialogue and dynamically adjusts the response. To get started, you can choose one of the sample bots provided below or build a new custom bot from scratch.

CREATE YOUR OWN

TRY A SAMPLE

Custom bot **BookTrip** **OrderFlowers** **ScheduleAppointment**

Bot name
ScheduleAppointment

Intents
A particular goal that the user wants to achieve

Utterances
Spoken or typed phrases that invoke your intent

I'd like to see the dentist.

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Haze 01:36 27-10-2024 © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon Lex us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-create:

Services Search [Alt+S] N. Virginia aaraf all

EC2 VPC S3

Language English (US)

Sentiment analysis Yes No

IAM role AWSServiceRoleForLexBots
Automatically created on your behalf

COPPA Please indicate if your use of this bot is subject to the [Children's Online Privacy Protection Act](#) (COPPA). Learn more
 Yes No

Confidence score threshold 0.4 (default)

Tags

Create Cancel Activate Windows Go to Settings to activate Windows.

CloudShell Feedback Type here to search 26°C Haze 01:36 27-10-2024 © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon Lex

us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=ScheduleAppointment

Build Publish

ScheduleAppointment Latest

Editor Settings Channels Monitoring

Intents +

MakeAppointment

Slot types +

AppointmentTypeValue

Error Handling

ScheduleAppointment build was successful

The build is now complete. You can now test the bot in the test window.

You're now ready for complete testing. Type an utterance below to begin conversation with your chatbot.

Clear chat history

Chat with your bot...

Inspect response

When you chat with your bot, you can see the fulfillment state of your intent and the response here.

Activate Windows

Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Haze 01:39 27-10-2024

Amazon Lex

us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=ScheduleAppointment

Build Publish

ScheduleAppointment Latest

Editor Settings Channels Monitoring

Intents +

MakeAppointment

Slot types +

AppointmentTypeValue

Error Handling

Lambda initialization and validation

Context

Slots

Priority	RequiredName	Slot type	Version	Prompt	Settings
1.	✓ Appointment	Appoin...	1	e.g. What type of appointment?	
2.	✓ Date	AMAZ...	Built-in	When should I sch...	
3.	✓ Time	AMAZ...	Built-in	At what time shou...	

Confirmation prompt

Fulfillment

AWS Lambda function Return parameters to client

Response

Test bot (Latest)

Ready. Build complete.

10 pm

22:00 is available, should I go ahead and book your appointment?

Clear chat history

Chat with your bot...

Inspect response

Dialog State: ConfirmIntent

Summary Detail

Intent: MakeAppointment

Slots (3/3)

AppointmentType root canal

Date 2024-10-25

Time 22:00

Activate Windows

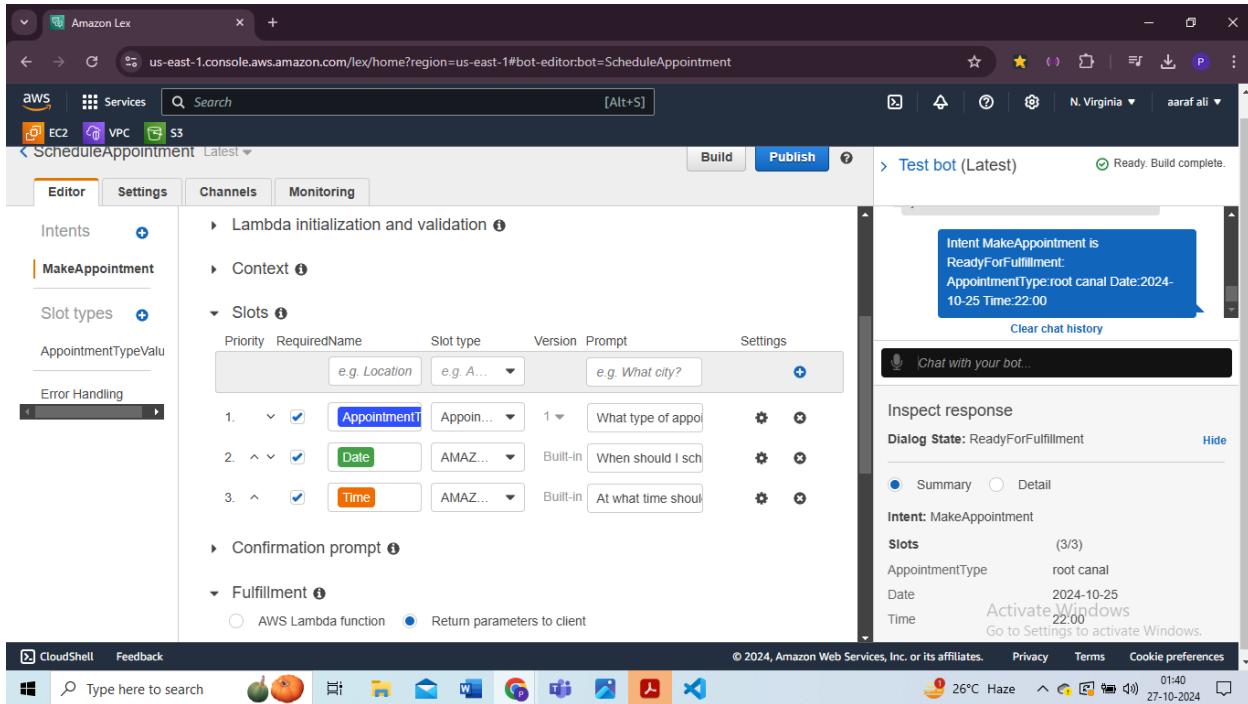
Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Haze 01:39 27-10-2024



Step 2 : Create the Lambda Function for Intent Fulfillment

1. **Create a Lambda Function**:

- In the AWS Lambda console, create a new function using Python as the runtime.
- Name it something like `MyChatBot`.

2. **Assign IAM Permissions**:

- Attach an IAM role to this function that allows it to interact with Lex, S3, and other AWS resources as needed.

3. **Write the Lambda Function Code**:

4. **Connect Lambda to Lex**:

- In the Lex console, for each intent, choose the Lambda function `MyChatBot` in the fulfillment section.

5. **Test the Lambda Integration**:

- Verify that Lex sends inputs to the Lambda function and receives appropriate responses.

The screenshot shows the AWS Lambda home page. At the top, there's a search bar and a navigation menu with tabs for EC2, VPC, and S3. A sidebar on the left lists Compute services. The main content area features a large heading "AWS Lambda" with the subtext "lets you run code without thinking about servers." Below this, a "Get started" box encourages users to author a function from scratch or choose from preconfigured examples, with a prominent orange "Create a function" button. A note below states: "You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration." The "How it works" section follows, with tabs for .NET, Java, Node.js, Python, Ruby, and Custom runtime. It includes a "Run" button and a link to "Next: Lambda responds to events".

AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

How it works

.NET Java Node.js Python Ruby Custom runtime

Run Next: Lambda responds to events

Activate Windows Go to Settings to activate Windows.

Create function

Choose one of the following options to create your function.

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

Basic information

Blueprint name: Hello world function nodejs18.x

Function name: myChatBot

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Activate Windows Go to Settings to activate Windows.

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is selected. In the 'Blueprint name' section, 'Make an appointment with Lex' is chosen with 'python3.10' selected as the runtime. The 'Function name' field contains 'myFunctionName'. A note states: 'Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).' On the right, there's an 'Activate Windows' message with a link to settings.

Basic information

Blueprint name: Make an appointment with Lex Runtime: python3.10

Function name: myFunctionName

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Activate Windows
Go to Settings to activate Windows.

The screenshot shows the code editor with the generated Python script. The script imports json, dateutil.parser, datetime, time, os, math, random, and logging. It defines a logger and sets its level to DEBUG. A docstring explains the purpose of the function. The main function, 'elicit_slot', takes session_attributes, intent_name, slots, slot_to_elicit, message, and response. It returns a dictionary with 'sessionAttributes' and 'dialogAction' keys.

```
1  #!/usr/bin/python3
2  # For INSTRUCTIONS on how to set up and test this bot, as well as additional samples,
3  # visit the Lex Getting Started documentation http://docs.aws.amazon.com/lex/latest/dg/getting-started.html
4  #
5  # This sample uses the Python AWS Lambda integration with Amazon Lex.
6  #
7  # This sample bot handles a simple appointment booking. It asks for the user's name, date, and time.
8  # It then provides a confirmation message and ends the session.
9  #
10 # Import required modules
11 import json
12 import dateutil.parser
13 import datetime
14 import time
15 import os
16 import math
17 import random
18 import logging
19
20 logger = logging.getLogger()
21 logger.setLevel(logging.DEBUG)
22
23 """
24     --- Helpers to build responses which match the structure of the necessary dialog actions
25
26
27 def elicit_slot(session_attributes, intent_name, slots, slot_to_elicit, message, response,
28                 return {
29                     'sessionAttributes': session_attributes,
30                     'dialogAction': {
```

1:1 Python Spaces:4

Create function

Activate Windows
Go to Settings to activate Windows.

The screenshot shows two browser windows side-by-side, both connected to the AWS Lambda service.

Top Window (Lambda Console):

- URL: us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myChatBot?newFunction=true&tab=testing
- Header: "myChatBot | Functions | Lambda"
- Message: "Successfully created the function myChatBot. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'."
- Tab: "Test" (selected)
- Content: "Executing function: succeeded (logs [2])" with a "Details" link.
- Buttons: "CloudWatch Logs Live Tail", "Save", and "Test".

Bottom Window (Lex Bot Editor):

- URL: us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=ScheduleAppointment
- Header: "ScheduleAppointment | Latest" (Editor tab selected)
- Content:
 - "Add permission to Lambda Function" dialog is open, stating: "You are about to give Amazon Lex permission to invoke your Lambda Function." with "OK" and "Cancel" buttons.
 - Left sidebar: "Intents" (MakeAppointment selected), "Slot types", "AppointmentTypeValue", and "Error Handling".
 - Right sidebar: "Build", "Publish", and "Test Chatbot" buttons.
 - Bottom status bar: "Activate Windows Go to Settings to activate Windows.", "26°C Haze", "01:42 27-10-2024".

The screenshot shows the Amazon Lex console interface for the 'myChatBot' function. The top half displays the 'ScheduleAppointment' intent configuration, including sample utterances like 'Book a flight' and 'Book an appointment'. It also shows Lambda initialization and validation settings. The bottom half shows the 'Test Chatbot' interface, where a confirmation prompt is displayed: 'When should I sch...'. The right side of the interface includes status indicators and links for 'Build' and 'Publish'.

Amazon Lex

us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=ScheduleAppointment

Services

Search [Alt+S]

EC2 VPC S3

ScheduleAppointment Latest

Editor Settings Channels Monitoring

Intents +

MakeAppointment

Slot types +

AppointmentTypeValue

Error Handling

Sample utterances

e.g. I would like to book a flight.

Book a **(AppointmentType)**

Book an appointment

I would like to book an appointment

Lambda initialization and validation

Initialization and validation code hook

Lambda function myChatBot

View in Lambda console

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

Amazon Lex myChatBot Functions Lambda

us-east-1.console.aws.amazon.com/lex/home?region=us-east-1#bot-editor:bot=ScheduleAppointment

Services

Search [Alt+S]

EC2 VPC S3

ScheduleAppointment Latest

Editor Settings Channels Monitoring

Intents +

MakeAppointment

Slot types +

AppointmentTypeValue

Error Handling

Confirmation prompt

2. Date AMAZ... Built-in When should I sch...

3. Time AMAZ... Built-in At what time shou...

Fulfillment

AWS Lambda function Return parameters to client

Lambda function myChatBot

View in Lambda console

Version or alias Latest

Response

Add Message

> Test bot (Latest) Ready. Build complete.

You're now ready for complete testing. Type an utterance below to begin conversation with your chatbot.

Clear chat history

Chat with your bot...

Inspect response

When you chat with your bot, you can see the fulfillment state of your intent and the response here.

Activate Windows Go to Settings to activate Windows.

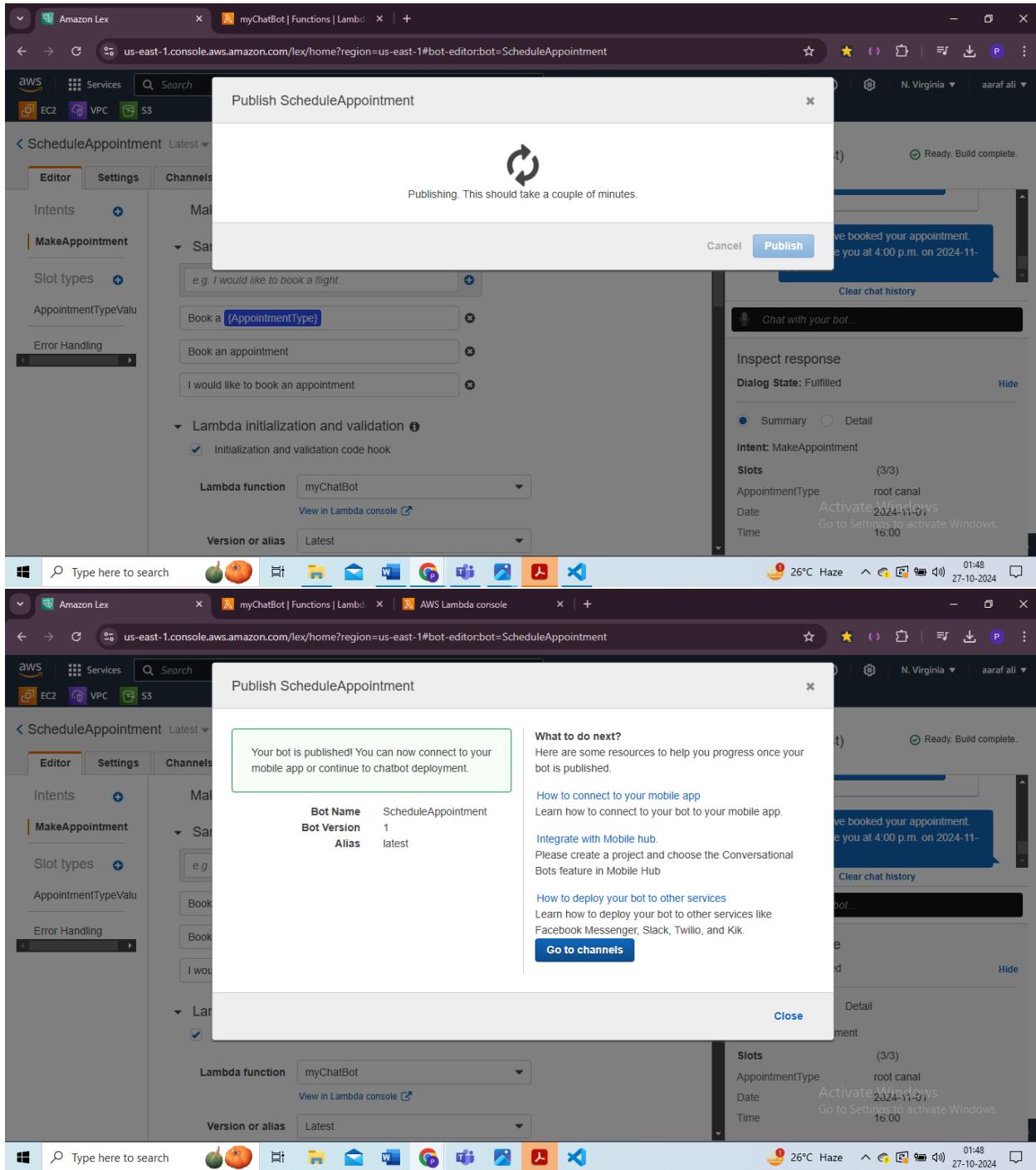
CloudShell Feedback

Type here to search

Activate Windows Go to Settings to activate Windows.

Air: Poor 01:45 27-10-2024

The screenshot shows the AWS Lambda and Amazon Lex console interface. The main focus is on the 'ScheduleAppointment' intent configuration under the 'Editor' tab. The 'Intents' section shows one intent named 'MakeAppointment'. The 'Sample utterances' section contains four examples: 'e.g. I would like to book a flight.', 'Book a [AppointmentType]', 'Book an appointment', and 'I would like to book an appointment'. The 'Lambda initialization and validation' section includes an 'Initialization and validation code hook' checkbox, which is checked, and a dropdown for the Lambda function set to 'myChatBot'. The 'Response' section is expanded, showing options for 'Add Message', 'Enable response card', and 'Wait for user reply'. A note states: 'If the user says "no," the following message will be presented.' At the bottom of the intent configuration, there are 'Save Intent' and 'Detach intent' buttons. To the right, the 'Test bot' panel shows the status 'Preparing build for express testing'. It includes a chat history placeholder 'Chat with your bot...', an 'Inspect response' section, and a note about viewing fulfillment state. The top navigation bar shows the region as 'N. Virginia' and the user as 'aaraf ali'. The bottom of the screen shows the Windows taskbar with various pinned icons.



Step 3: Use Amazon Cognito for User Authentication

1. **Create a Cognito Identity Pool**:

- Create an identity pool to enable authenticated access.
- Configure the identity pool to allow both authenticated and unauthenticated users if needed.

2. **Configure Authentication in Your Application**:

- Integrate Cognito in your front-end application (e.g., mobile or web) to handle user sign-in and sign-up.
- Upon authentication, Cognito will issue tokens that can be used for secure API calls.

The screenshot shows the AWS Lambda console with the function configuration page for 'myChatBot'. The 'Handler' dropdown is set to 'lambda-function-role-sh4y4lens'. The 'Code' section shows the path 'myChatBot.zip' and the 'Edit' button. The 'Environment' tab is selected, showing environment variables like 'AWS_LAMBDA_FUNCTION_NAME' and 'AWS_LAMBDA_FUNCTION_MEMORY_SIZE'. The 'Logs' tab is also visible.

The screenshot shows the 'Identity pools' section of the AWS Cognito console. It displays a table with columns: 'Identity pool name', 'Identity pool ID', 'Created time', and 'Last updated time'. A search bar at the top allows filtering by name or ID. A message indicates 'No identity pools' and provides a 'Create identity pool' button.

The screenshot shows the first step of the 'Create identity pool' wizard, titled 'Configure identity pool trust'. The left sidebar lists steps: Step 1 (Configure identity pool trust), Step 2 (Configure permissions), Step 3 (Configure properties), and Step 4 (Review and create). The main panel is titled 'Authentication' and describes configuring sources for generating identities and issuing credentials. It includes sections for 'User access' and 'Guest access'. A note at the bottom explains guest access distribution and directs users to learn more.

The screenshots illustrate the process of creating a new Cognito Identity Pool. In the first two steps, the user is configuring basic properties, including the identity pool name 'myIdentityPool'. In the third step, the user is setting up a guest role, choosing to 'Create a new IAM role' and naming it 'myIdentitypool'. The IAM role creation includes a note about initial minimum permissions and trust relationships.

Step 4: Set Up IAM Roles and Policies

1. **Create an IAM Role for Lambda**:

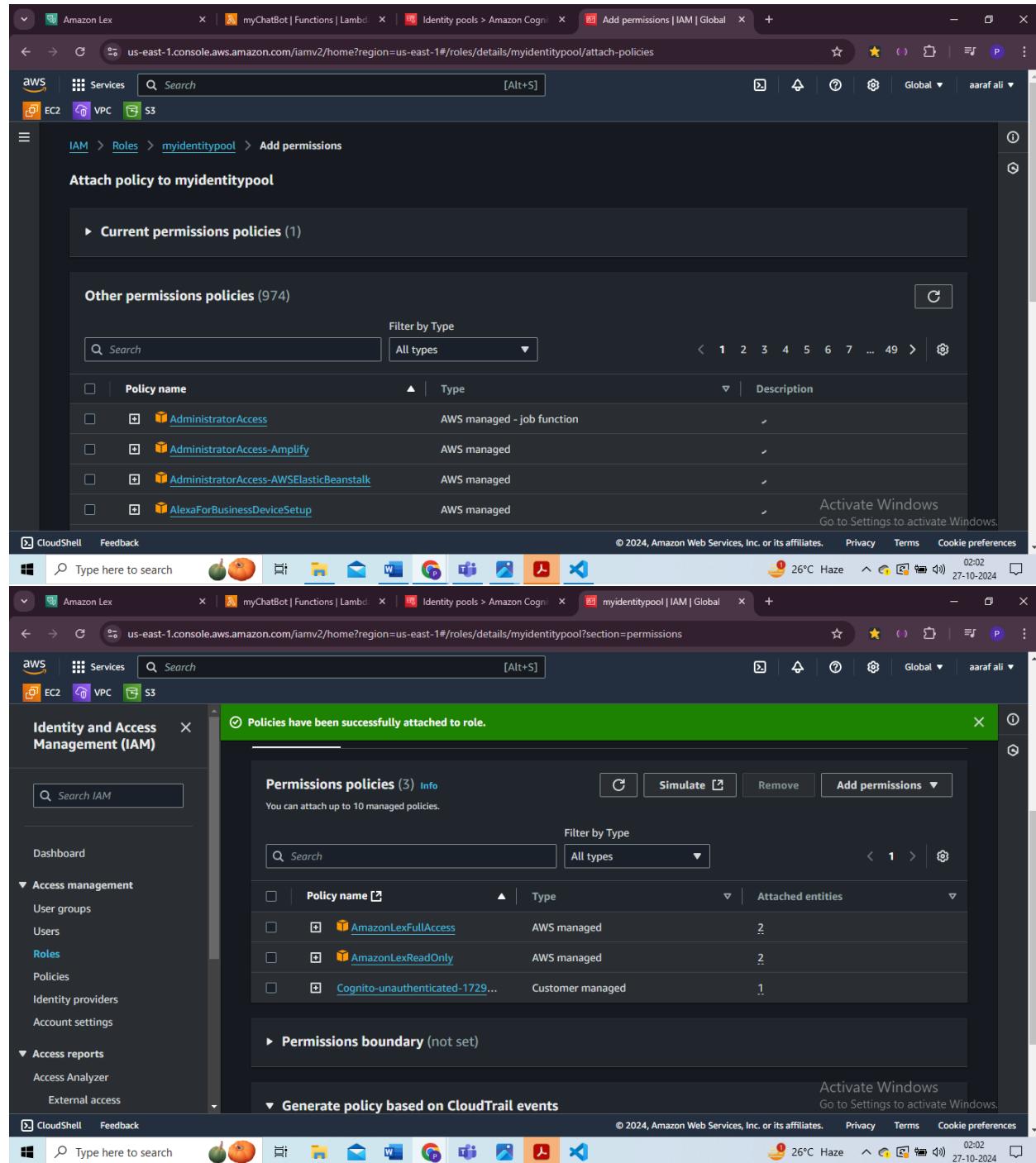
- Attach an IAM role to the Lambda function with permissions to interact with Lex, S3, and other required services.

2. **Create an IAM Role for Cognito**:

- Create separate IAM roles for authenticated and unauthenticated users in Cognito.
- Define policies that limit access to specific resources based on user roles.

3. **Set Permissions**:

- Ensure that your IAM policies provide the minimum necessary permissions, following the principle of least privilege.



The screenshot shows two instances of the AWS IAM console interface. The top instance is titled 'Add permissions | IAM | Global' and displays the 'Attach policy to myidentitypool' step. It lists 'Current permissions policies (1)' and 'Other permissions policies (974)'. The bottom instance is titled 'myidentitypool | IAM | Global' and shows the successful attachment of three policies: 'AmazonLexFullAccess', 'AmazonLexReadOnly', and 'Cognito-unauthenticated-1729...'. Both instances include standard browser navigation and status bars at the bottom.

Top Window: Add permissions | IAM | Global

Bottom Window: myidentitypool | IAM | Global

Success Message: Policies have been successfully attached to role.

Permissions policies attached:

Policy name	Type	Attached entities
AmazonLexFullAccess	AWS managed	2
AmazonLexReadOnly	AWS managed	2
Cognito-unauthenticated-1729...	Customer managed	1

Step 5: Store Data and Files in S3

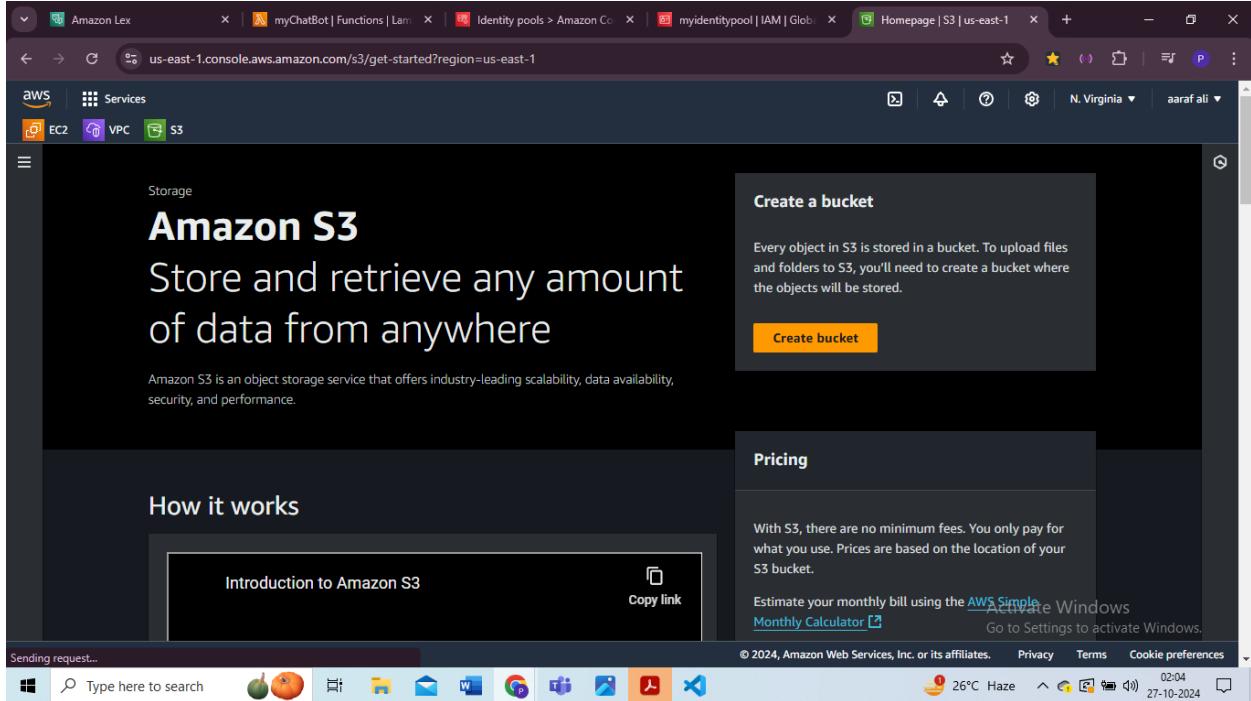
1. **Create an S3 Bucket**:

- In the S3 Console, create a bucket for storing files, such as appointment records or logs.

2. **Configure Permissions**:

- Set bucket policies or object policies to restrict access to the files only to authorized users (e.g., via Cognito roles).

3. **Enable Static Website Property**:



The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section is displayed, with the 'AWS Region' set to 'US East (N. Virginia) us-east-1'. The 'Bucket type' dropdown is open, showing 'General purpose' (selected) and 'Directory'. The 'Bucket name' field contains 'mydentalchatbot'. A note below the name field states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.' The status bar at the bottom right shows 'Activate Windows Go to Settings to activate Windows.'

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: General purpose

Bucket name: mydentalchatbot

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.

Activate Windows
Go to Settings to activate Windows.

The screenshot shows the 'Upload objects - S3 bucket' page in the AWS S3 console. It displays a summary of the upload results: 'Destination s3://mydentalchatbot' with 'Succeeded' status for '3 files, 227.5 KB (100.00%)' and 'Failed' status for '0 files, 0 B (0%)'. Below this, a table lists the uploaded files: 'dentist.jpg' (image/jpeg, 210.4 KB, succeeded), 'error.html' (text/html, 3.0 KB, succeeded), and 'index.html' (text/html, 14.0 KB, succeeded). The status bar at the bottom right shows 'Activate Windows Go to Settings to activate Windows.'

Upload succeeded

Succeeded: 3 files, 227.5 KB (100.00%)

Failed: 0 files, 0 B (0%)

Files and folders (3 Total, 227.5 KB)

Name	Folder	Type	Size	Status	Error
dentist.jpg	-	image/jpeg	210.4 KB	Succeeded	-
error.html	-	text/html	3.0 KB	Succeeded	-
index.html	-	text/html	14.0 KB	Succeeded	-

Activate Windows
Go to Settings to activate Windows.

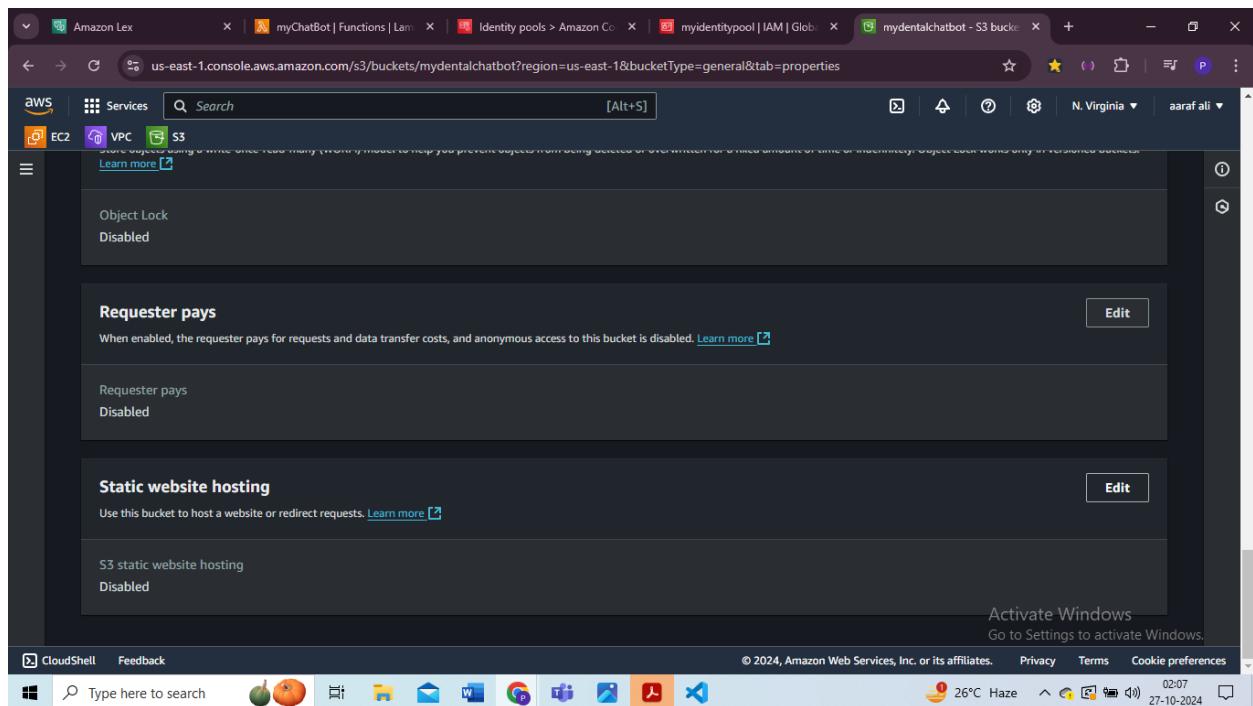
The screenshot shows the 'Buckets' page in the AWS S3 console, displaying the contents of the 'mydentalchatbot' bucket. The table shows three files: 'dentist.jpg' (image/jpeg, 210.4 KB, succeeded), 'error.html' (text/html, 3.0 KB, succeeded), and 'index.html' (text/html, 14.0 KB, succeeded). The status bar at the bottom right shows 'Activate Windows Go to Settings to activate Windows.'

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Haze 02:06 27-10-2024



Step 6: Test the Dental Chatbot End-to-End

1. **Test in the Lex Console**:

- Test the chatbot to ensure that it correctly recognizes intents, invokes Lambda, and retrieves information.

2. **Test Authentication and File Storage**:

- Verify that only authenticated users can schedule appointments and that S3 files are accessible only to users with proper permissions.

Amazon Lex - Dental Appointment BOT

Hi there! What do you want to do?

book a appointment

Specify Appointment Type
What type of appointment would you like to schedule?

root canal (60 min)

Specify Date
When would you like to schedule your root canal?

10-28 (Mon) 10-29 (Tue)
10-30 (Wed) 10-31 (Thu)
11-1 (Fri)

What do you want to do?



Activate Windows
GO TO ACTIVATION SITE TO ACTIVATE WINDOWS.

Amazon Lex - Dental Appointment BOT

Hi there! What do you want to do?

book a appointment

Specify Appointment Type
What type of appointment would you like to schedule?

root canal (60 min)

Specify Date
When would you like to schedule your root canal?

10-28 (Mon)

Specify Time
What time works best for you?

12:00 p.m. 4:00 p.m.

What do you want to do?

mydentalchatbot.s3-website-us-east-1.amazonaws.com/#



Activate Windows
GO TO ACTIVATION SITE TO ACTIVATE WINDOWS.

book a appointment

Specify Appointment Type
What type of appointment would you like to schedule?

root canal (60 min)

Specify Date
When would you like to schedule your root canal?

10-28 (Mon)

Specify Time
What time works best for you?

12:00 p.m.

12:00 p.m. is available, should I go ahead and book your root canal?

What do you want to do?

Type here to search

26°C Haze 02:28 27-10-2024

schedule?

root canal (60 min)

Specify Date
When would you like to schedule your root canal?

10-28 (Mon)

Specify Time
What time works best for you?

12:00 p.m.

12:00 p.m. is available, should I go ahead and book your root canal?

yes

Okay, I have booked your appointment. We will see you at 12:00 p.m. on 2024-10-28

What do you want to do?

Type here to search

26°C Haze 02:28 27-10-2024

Additional Features (Optional)

- **Multi-turn Dialogues**: Create a more engaging experience by adding follow-up questions.
- **Logging and Monitoring**: Use CloudWatch for logging, performance monitoring, and setting up alarms for any issues.

- **Reminders**: Set up reminders using Amazon SNS to send appointment reminders to patients.

This setup enables a secure, cloud-based dental chatbot that interacts with users, schedules appointments, and stores data safely in AWS.