Chatbot using amazon lex

MINI PROJECT REPORT

18CSC312J - Artificial Intelligence and Applications in Cloud Computing

(2018-regulation)

III Year/ VI Semester

Academic Year: 2022 -2023

By

Chinta Pradeep(RA2011028010059)
Silpi Kartheek Achari(RA2011028010068)

Under the guidance of

Dr. Vaishnavi Moorthy
Assistant Professor

Department of Networking and Communications

DEPARTMENT OF NETWORKING AND COMMUNICATIONS
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Kattankulathur, Kancheepuram
MAY 2023

BONAFIDE

Certified that this Mini project report titled “Chatbot using amazon lex” for the course 18CSC312J – ARTIFICIAL INTELLIGENCE AND APPLICATIONS IN CLOUD COMPUTING LABORATORY is the bonafide work of Chinta Pradeep(RA2011028010059), Silpi Kartheek Achari(RA2011028010068) who undertook the task of completing the project within the allotted time.

Signature

Dr. Vaishnavi Moorthy
Course Faculty
Professor
Department of NWC

Signature

M Dr. Annapurani Panaiyappan K
Head of the Department Assistant
Professor
Department of NWC

Abstract

Amazon Lex is a conversational interface built in the cloud that allows developers to create conversational interfaces for chatbots and other conversational apps. Developers may use Amazon Lex to construct chatbots that can interpret natural language inputand reply in a conversational manner. The service interprets user input and generates relevant answers using automated speech recognition (ASR) and natural language understanding (NLU) technology.

Amazon Lex comes with a pre-built collection of conversational components, or intents, that may be customised and enhanced to create a chatbot targeted to specific use cases. These intentions are intended to recognise typical user requests, such as booking a reservation, checking the status of an order, or offering product suggestions. To handle more sophisticated requests, developers can specify their own intents and slot kinds.

Amazon Lex works with other Amazon Web Services (AWS) technologies like AWS Lambda and Amazon DynamoDB to provide a full development environment for creating conversational apps. AWS Lambda may be used by developers to construct code that handles certain intents and slot kinds, while DynamoDB can be used to store data related to the chatbot's functionality.

A web-based dashboard for managing and testing chatbots is also provided by Amazon Lex. Developers may use the console to build and edit intents and slot kinds, test chatbot answers, and track use and performance.

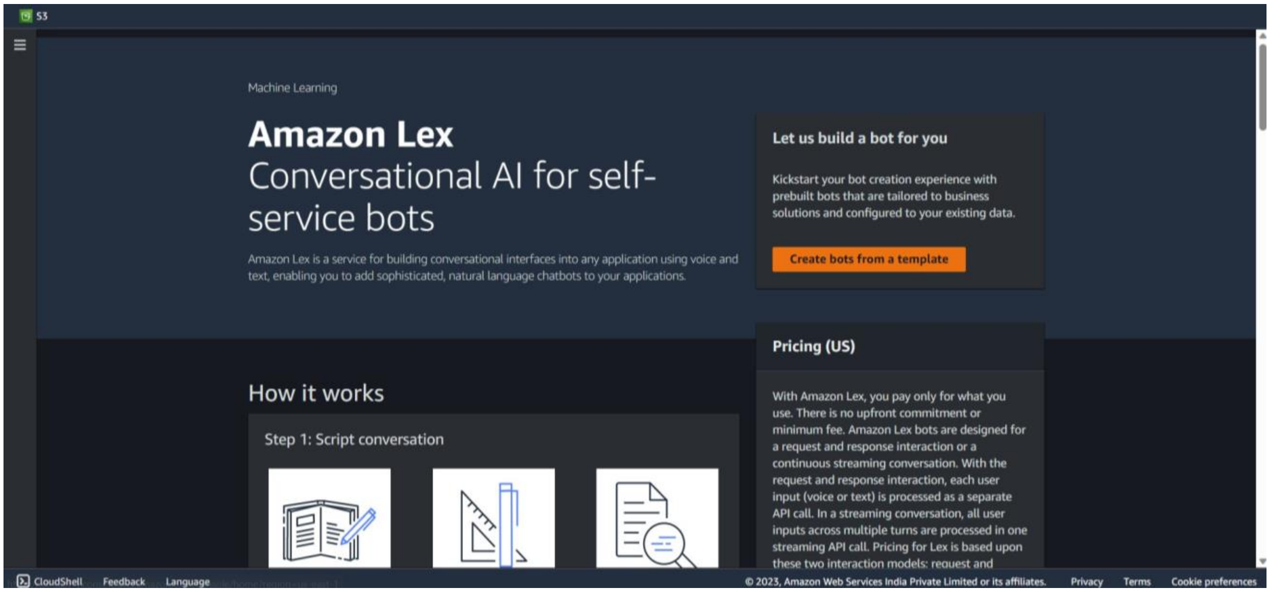
Scalability is one of Amazon Lex's primary features. As a cloud-based service, it can handle massive numbers of queries and dynamically scale up or down to suit demand. As a result, it is well suited for usage in applications requiring large levels of concurrency, such as customer care chatbots or virtual assistants.

Another benefit of Amazon Lex is its compatibility with other AWS services. Other AWS services, such as Amazon S3 or Amazon SageMaker, may be readily integrated by developers to increase the chatbot's capability. A chatbot for an e-commerce site.

In conclusion, Amazon Lex is an extremely powerful tool for creating conversational interfaces for chatbots and other conversational apps. It is a popular choice for developers wishing to design complex conversational interfaces because to its pre-built intents and slot kinds, scalability, and connection with other AWS services. Amazon Lexdelivers the tools and services you need to get started quickly and simply, whether you're developing a customer care chatbot, a virtual assistant, or a chatbot for an e-commerce site.

Introduction:-

Amazon Lex is a powerful tool for creating chatbots for a range of applications, including e-commerce. An e-commerce chatbot created using Amazon Lex may assistconsumers with a number of activities, including product discovery, order placement, shipment tracking, and customer service.



Amazon Lex

Conversational AI for self-service bots

Amazon Lex is a service for building conversational interfaces into any application using voice and text, enabling you to add sophisticated, natural language chatbots to your applications.

Let us build a bot for you

Kickstart your bot creation experience with prebuilt bots that are tailored to business solutions and configured to your existing data.

Pricing (US)

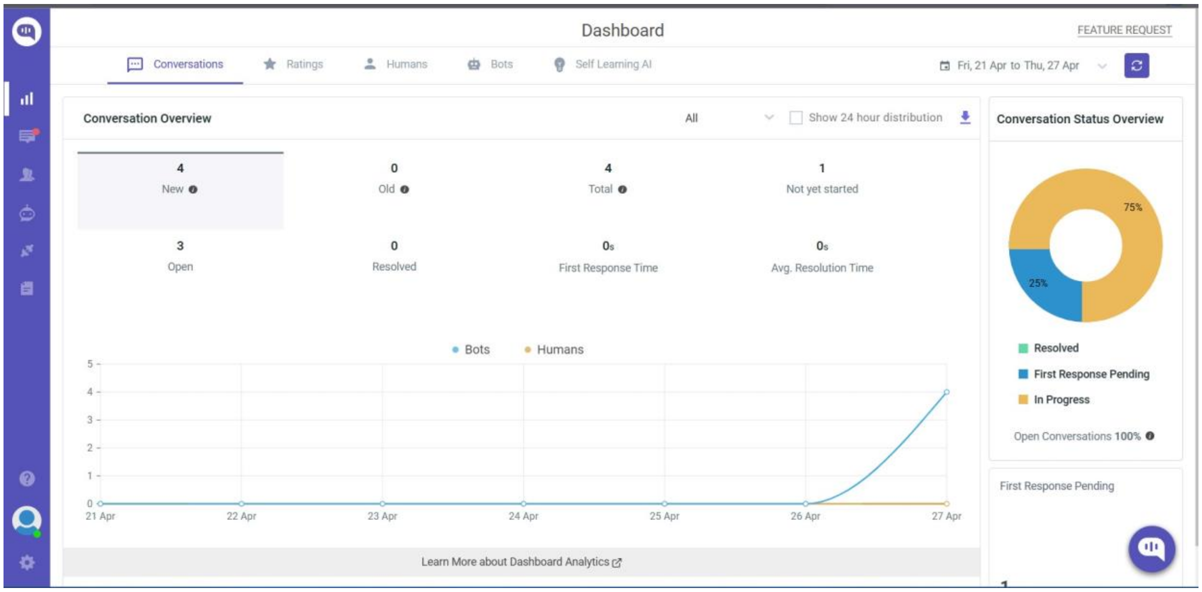
With Amazon Lex, you pay only for what you use. There is no upfront commitment or minimum fee. Amazon Lex bots are designed for a request and response interaction or a continuous streaming conversation. With the request and response interaction, each user input (voice or text) is processed as a separate API call. In a streaming conversation, all user inputs across multiple turns are processed in one streaming API call. Pricing for Lex is based upon these two interaction models: request and

How it works

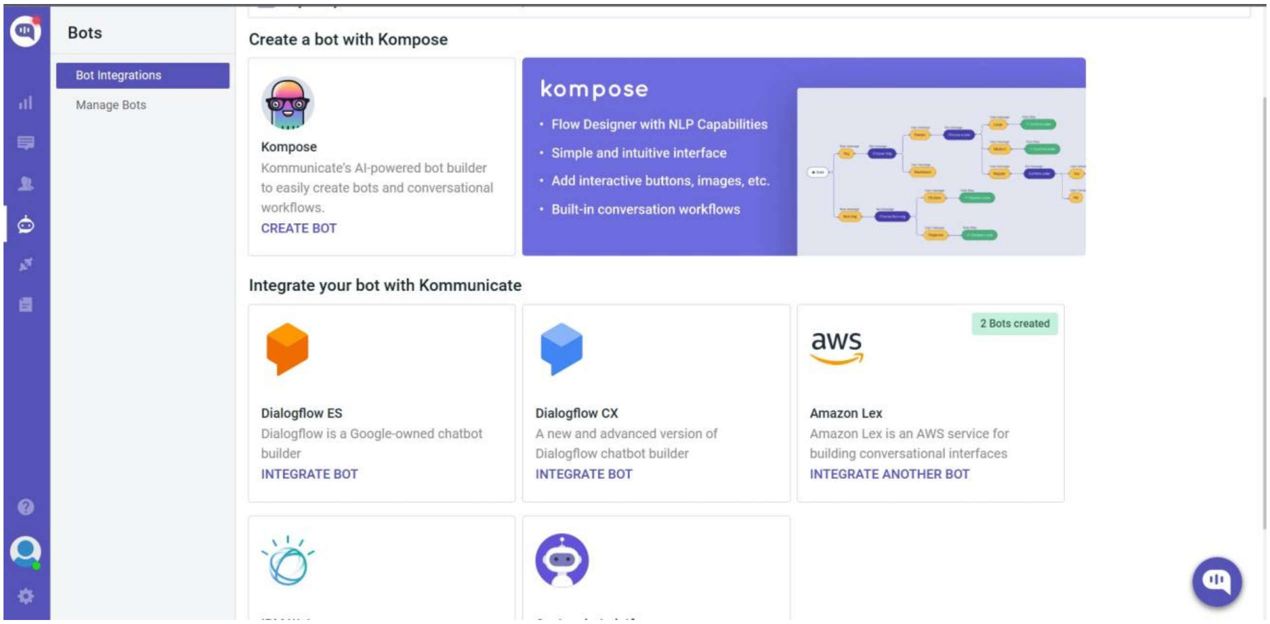
Step 1: Script conversation

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

This amazon lex is used to create a chatbot that is used to create a conversational chatbot that is generally used to integrate with third party softwares Kommunicate and integrate using telegram and other stuff.



4 New	0 Old	4 Total	1 Not yet started
3 Open	0 Resolved	0s First Response Time	0s Avg. Resolution Time



Bots

Bot Integrations

Manage Bots

Create a bot with Kompose

Kompose

Kommunicate's AI-powered bot builder to easily create bots and conversational workflows.

CREATE BOT

kompose

- Flow Designer with NLP Capabilities
- Simple and intuitive interface
- Add interactive buttons, images, etc.
- Built-in conversation workflows

Integrate your bot with Kommunicate

Dialogflow ES

Dialogflow is a Google-owned chatbot builder

INTEGRATE BOT

Dialogflow CX

A new and advanced version of Dialogflow chatbot builder

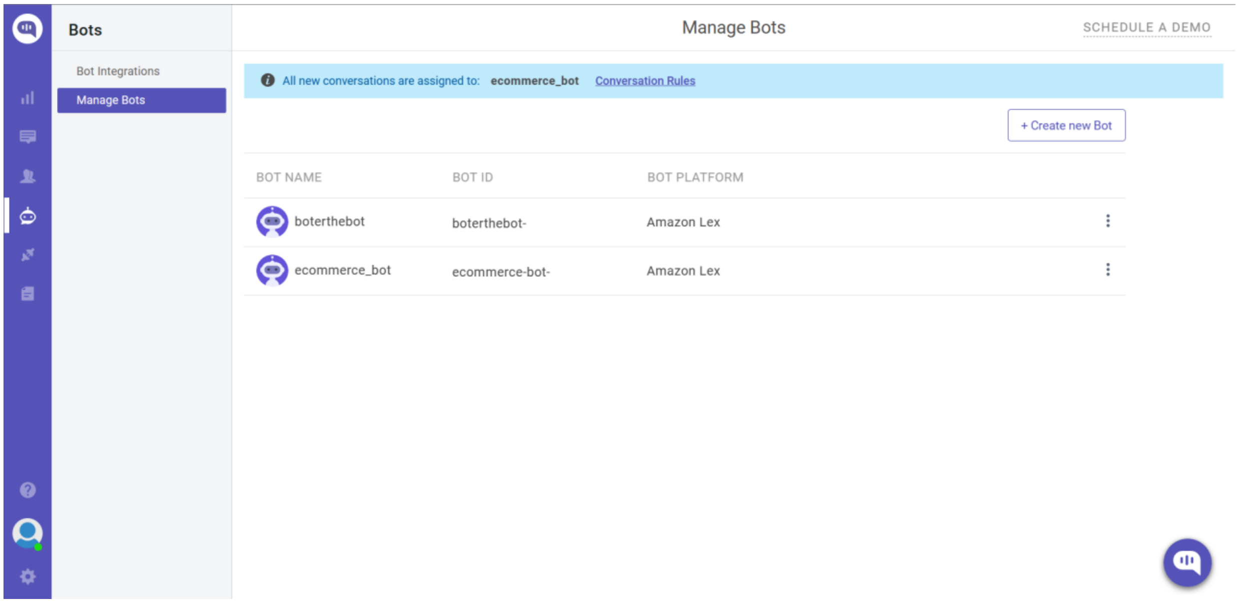
INTEGRATE BOT

2 Bots created

Amazon Lex

Amazon Lex is an AWS service for building conversational interfaces

INTEGRATE ANOTHER BOT



Bots

Bot Integrations

Manage Bots

Manage Bots

SCHEDULE A DEMO

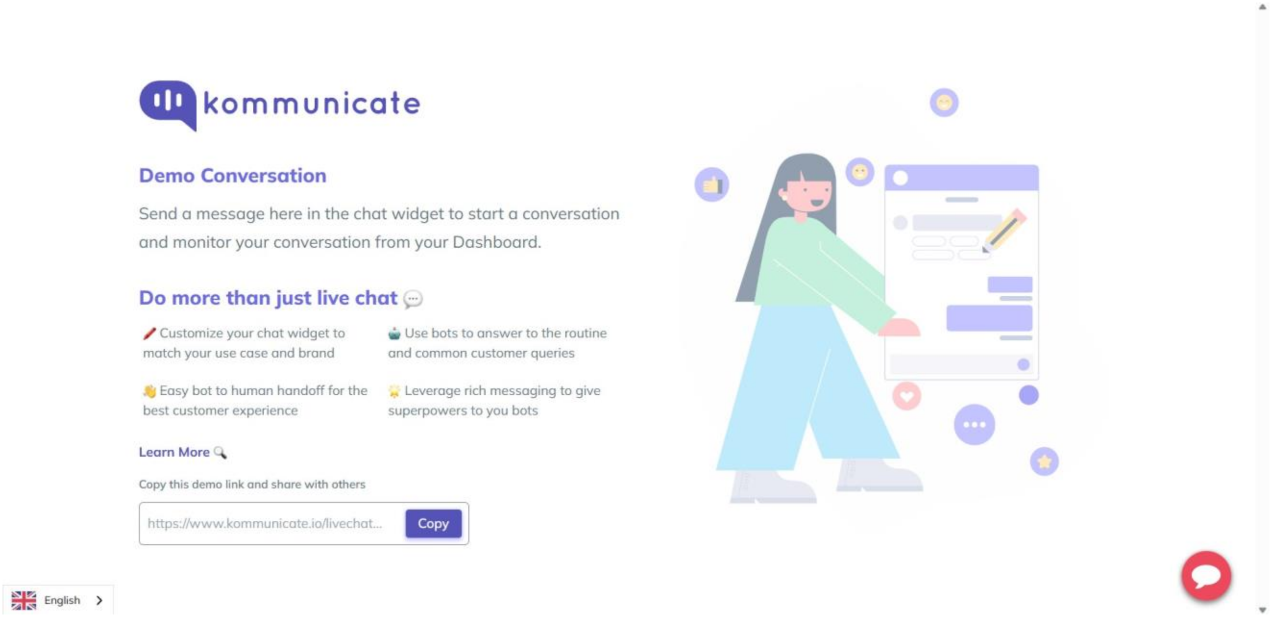
All new conversations are assigned to: **ecommerce_bot** Conversation Rules

+ Create new Bot

BOT NAME	BOT ID	BOT PLATFORM
boterthebot	boterthebot-	Amazon Lex
ecommerce_bot	ecommerce-bot-	Amazon Lex

METHODOLOGY USED:

We Both as a team used Amazon lex as a primary aws managed service to create a chatbot.We then integrated that into the kommunicate software using access id and secret id key.



Demo Conversation

Send a message here in the chat widget to start a conversation and monitor your conversation from your Dashboard.

Do more than just live chat 💬

- Customize your chat widget to match your use case and brand
- Use bots to answer to the routine and common customer queries
- Easy bot to human handoff for the best customer experience
- Leverage rich messaging to give superpowers to you bots

Learn More

Copy this demo link and share with others

https://www.kommunicate.io/livechat...

Implementation:

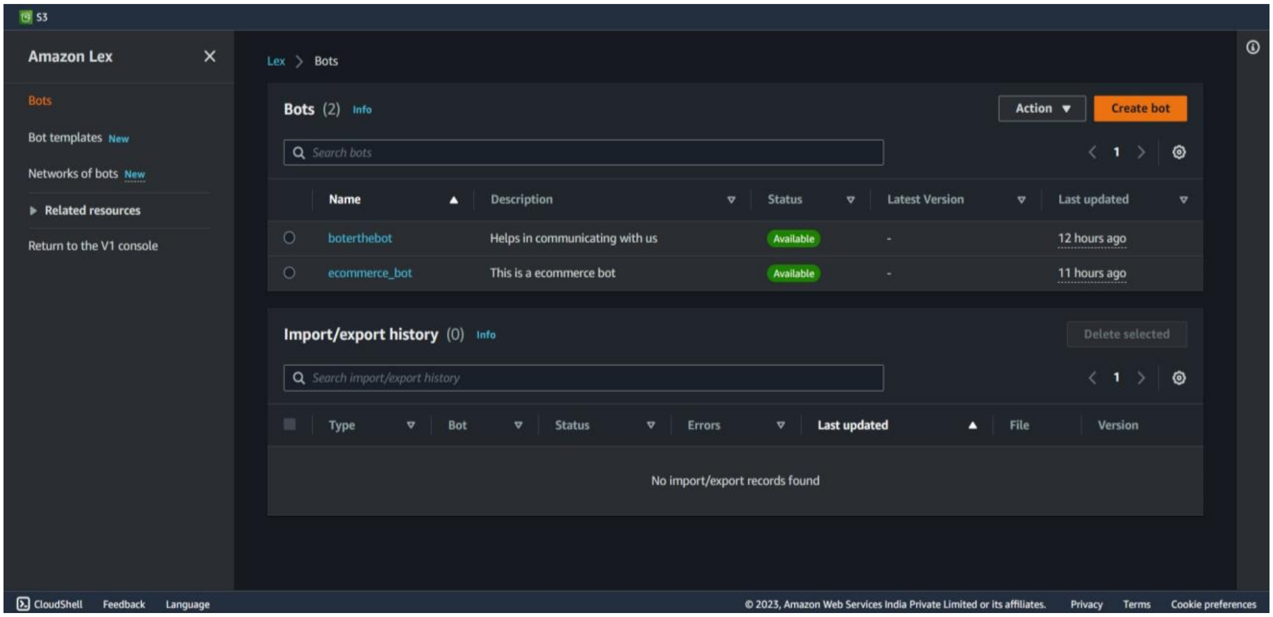
You may use Amazon Lex to create an e-commerce chatbot by following these steps:

1) Define the intents : The first step is to establish the intents that will be handled by your chatbot. You may have intents for looking for things, adding items to a basket, checking out, and monitoring orders.

2) Build the Bot : After you've specified your intentions, you can begin creating your chatbot. Amazon Lex has a visual editor that makes it simple to design your bot's conversational flow.

3) Connect to your Backend:To conduct activities like as looking for items and completing orders, your chatbot will need to connect to your e-commerce backend. AWS Lambda may be used to generate the backend functions that yourchatbot will need.

4) Test and Deploy:After you've constructed your chatbot, test it to ensure it functions properly. When you're happy with the findings, you may integrate it into your e-commerce website or mobile app.



Lex > Bots

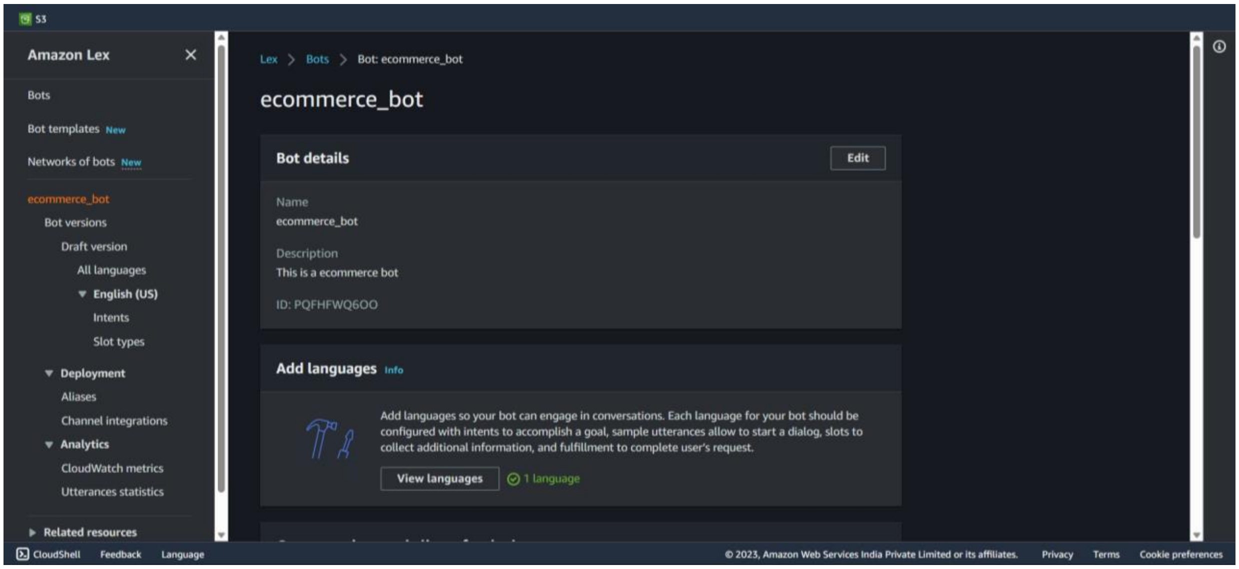
Bots (2) Info

Name	Description	Status	Latest Version	Last updated
botherthebot	Helps in communicating with us	Available	-	12 hours ago
ecommerce_bot	This is a ecommerce bot	Available	-	11 hours ago

Import/export history (0) Info

Type	Bot	Status	Errors	Last updated	File	Version

No import/export records found



Lex > Bots > Bot: ecommerce_bot

ecommerce_bot

Bot details

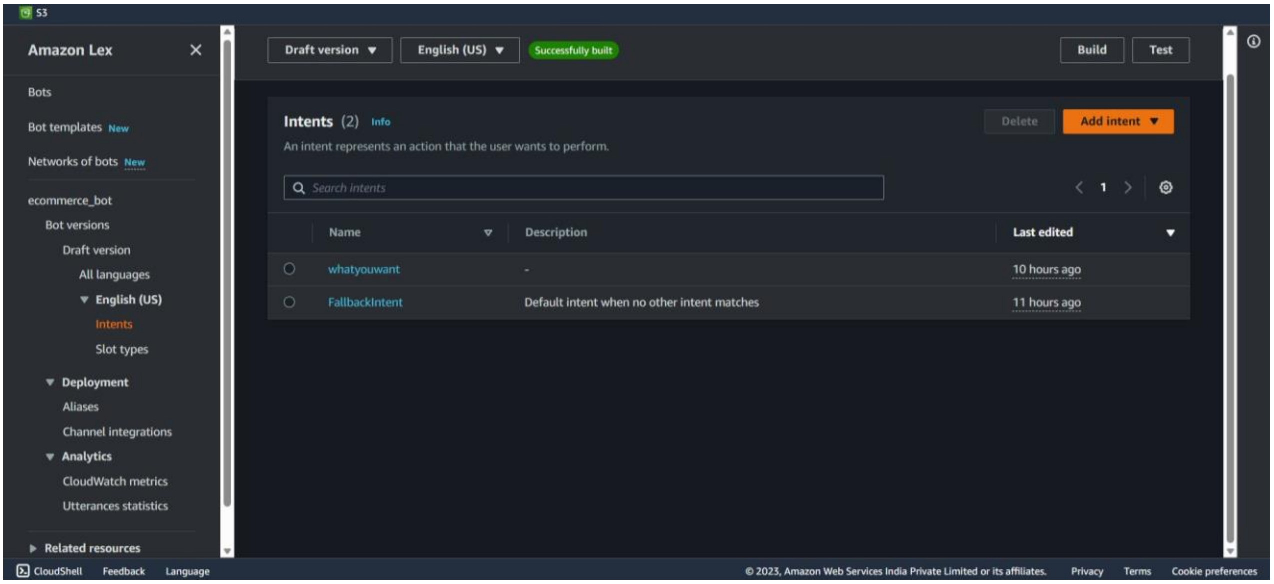
Name
ecommerce_bot

Description
This is a ecommerce bot

ID: PQFHFWQ6OO

Add languages Info

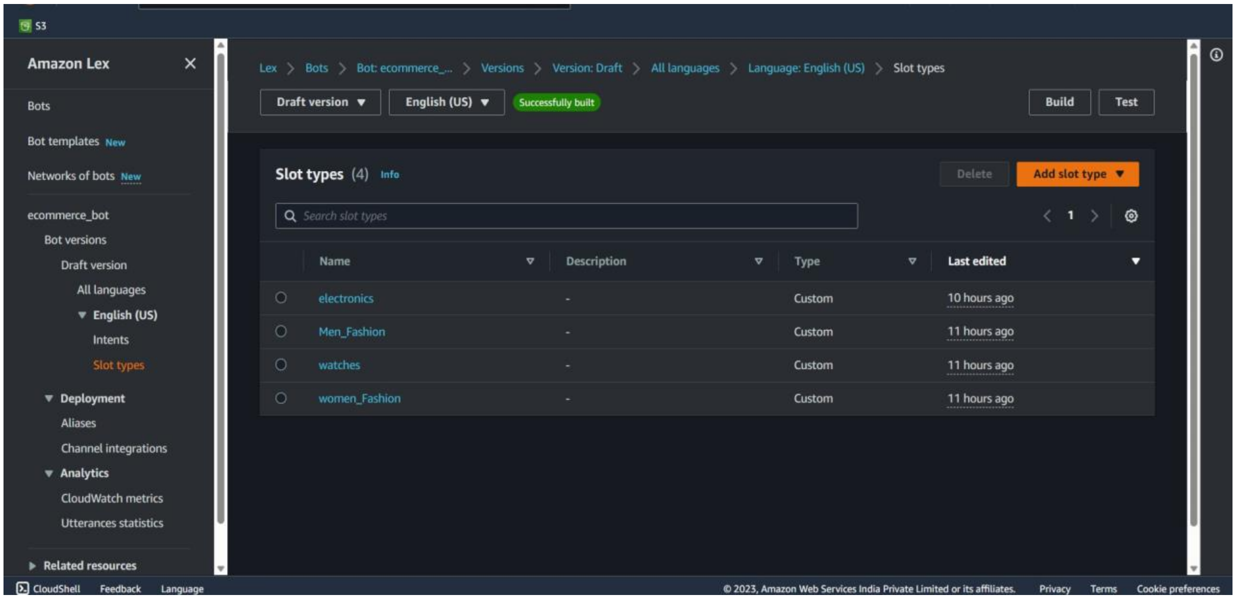
Add languages so your bot can engage in conversations. Each language for your bot should be configured with intents to accomplish a goal, sample utterances allow to start a dialog, slots to collect additional information, and fulfillment to complete user's request.



Intents (2) Info

An intent represents an action that the user wants to perform.

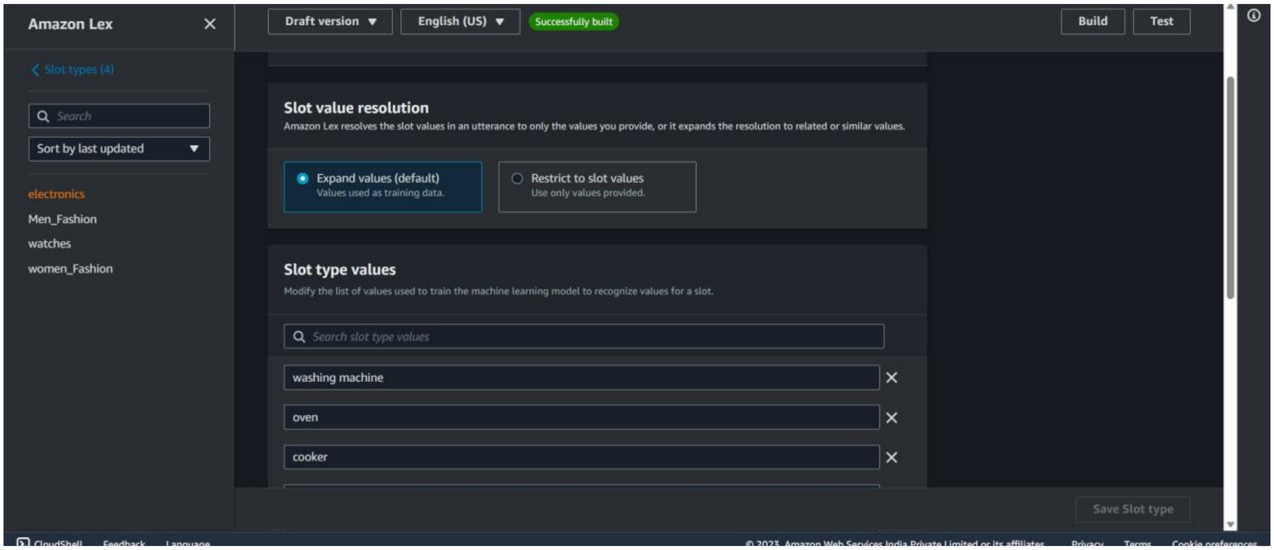
	Name	Description	Last edited
○	whatyouwant	-	10 hours ago
○	FallbackIntent	Default intent when no other intent matches	11 hours ago



Slot types (4) Info

	Name	Description	Type	Last edited
○	electronics	-	Custom	10 hours ago
○	Men_Fashion	-	Custom	11 hours ago
○	watches	-	Custom	11 hours ago
○	women_Fashion	-	Custom	11 hours ago

Slot-electronics:



Slot value resolution

Amazon Lex resolves the slot values in an utterance to only the values you provide, or it expands the resolution to related or similar values.

● Expand values (default)
Values used as training data.

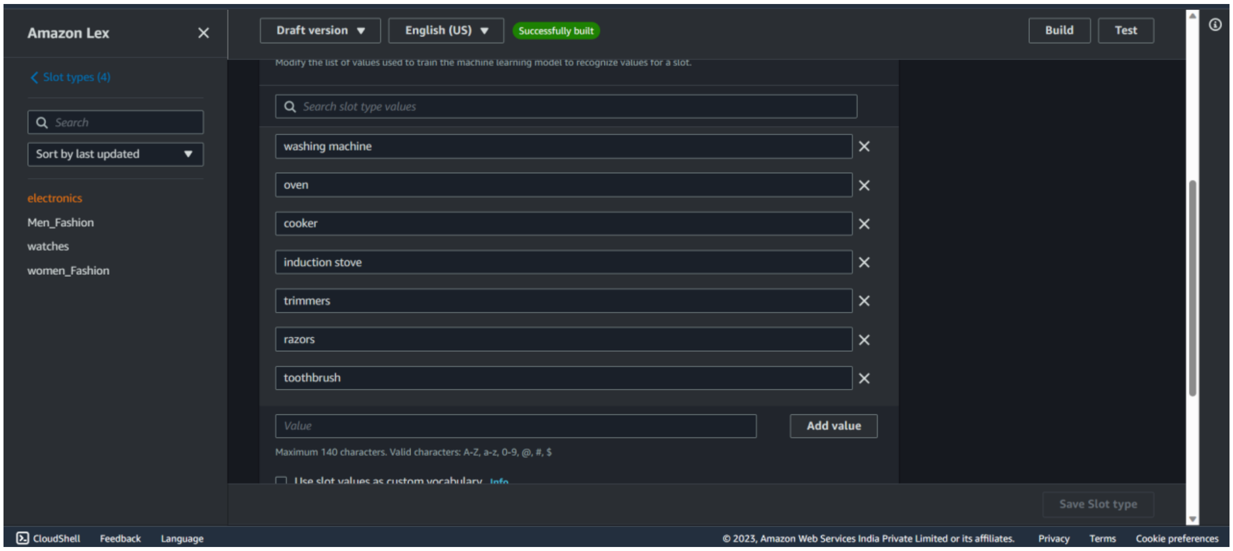
○ Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

- washing machine
- oven
- cooker



Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

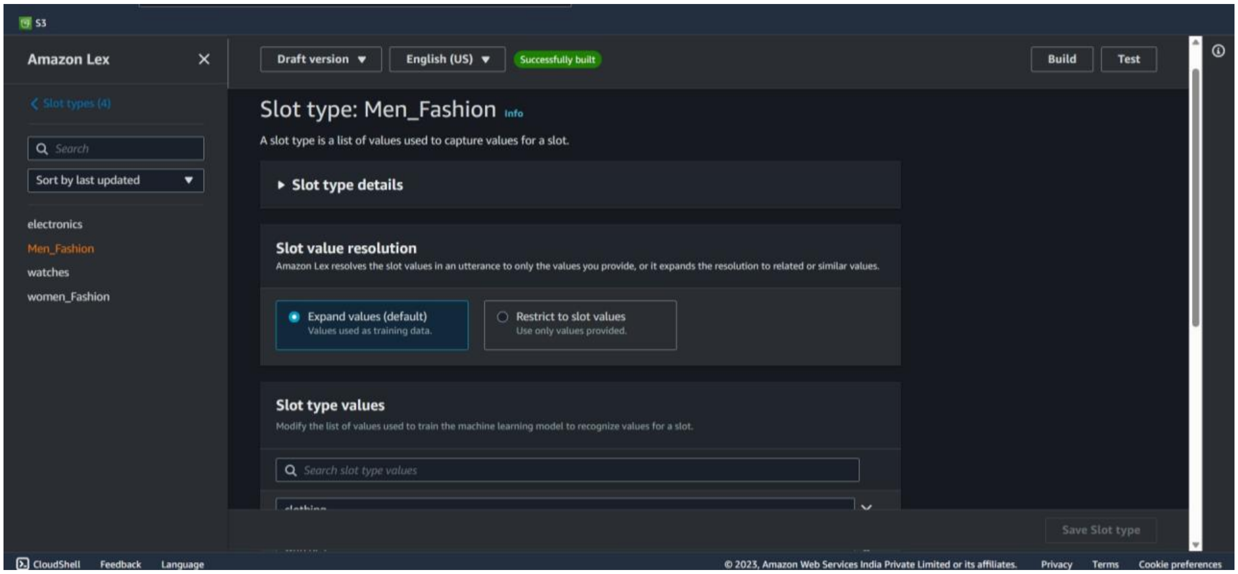
- washing machine
- oven
- cooker
- induction stove
- trimmers
- razors
- toothbrush

Value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

Slot-type : Men-Fashion



Slot type: Men_Fashion Info

A slot type is a list of values used to capture values for a slot.

▸ Slot type details

Slot value resolution

Amazon Lex resolves the slot values in an utterance to only the values you provide, or it expands the resolution to related or similar values.

● Expand values (default)
Values used as training data.

○ Restrict to slot values
Use only values provided.

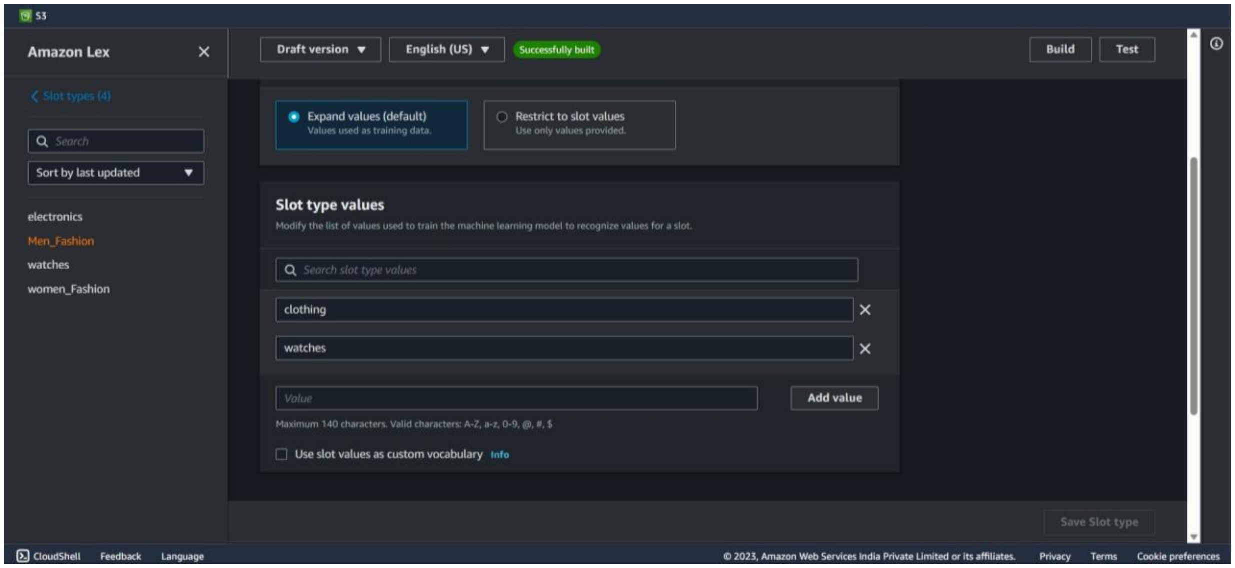
Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

clothing

© 2023, Amazon Web Services India Private Limited or its affiliates.



● Expand values (default)
Values used as training data.

○ Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

clothing ✕

watches ✕

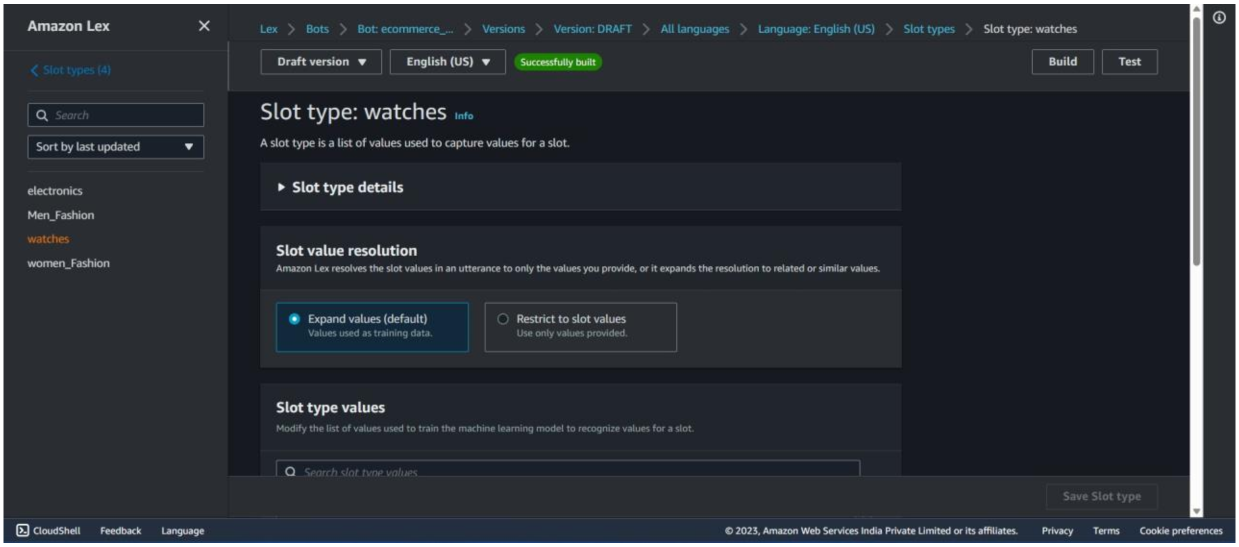
Value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

© 2023, Amazon Web Services India Private Limited or its affiliates.

Slot-type : watches



Slot type: watches Info

A slot type is a list of values used to capture values for a slot.

▸ Slot type details

Slot value resolution

Amazon Lex resolves the slot values in an utterance to only the values you provide, or it expands the resolution to related or similar values.

● Expand values (default)
Values used as training data.

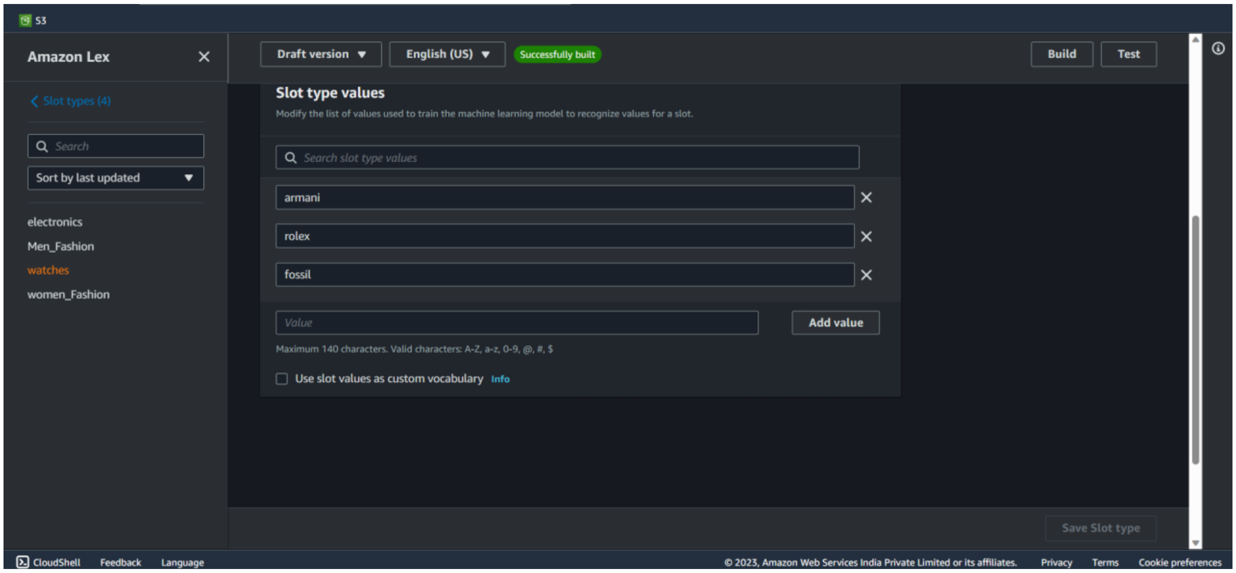
○ Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

© 2023, Amazon Web Services India Private Limited or its affiliates.



Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

Search slot type values

armani ✕

rolex ✕

fossil ✕

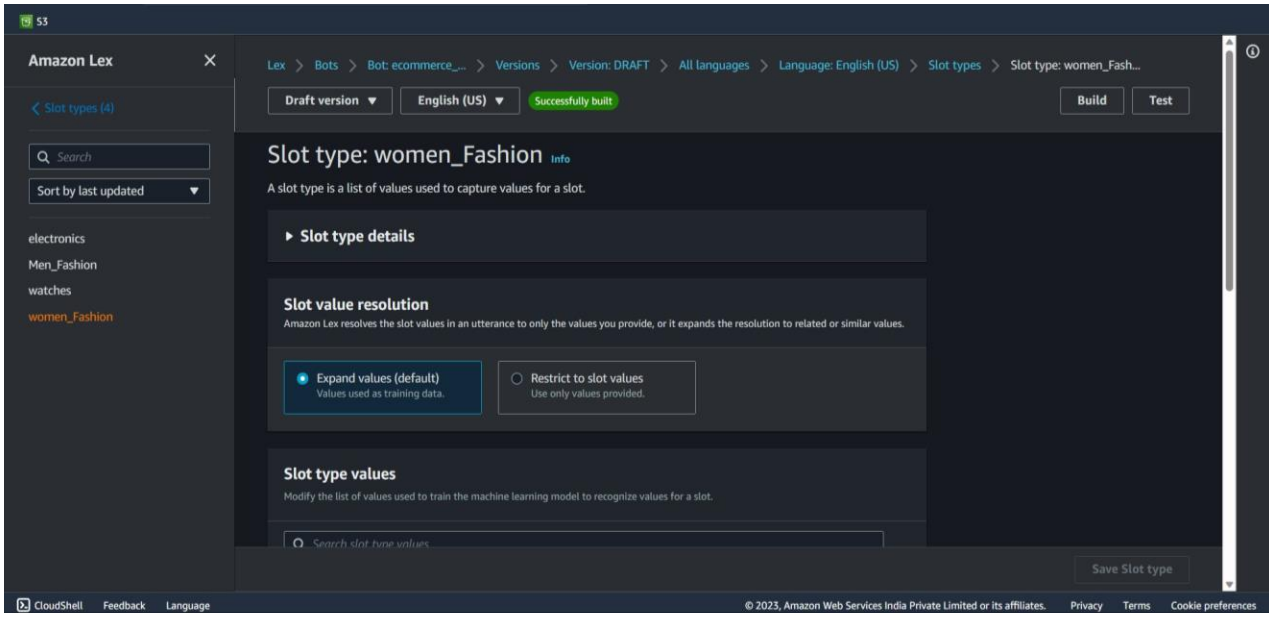
Value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

© 2023, Amazon Web Services India Private Limited or its affiliates.

Slot-type : women-fashion



Slot type: women_Fashion Info

A slot type is a list of values used to capture values for a slot.

▸ Slot type details

Slot value resolution

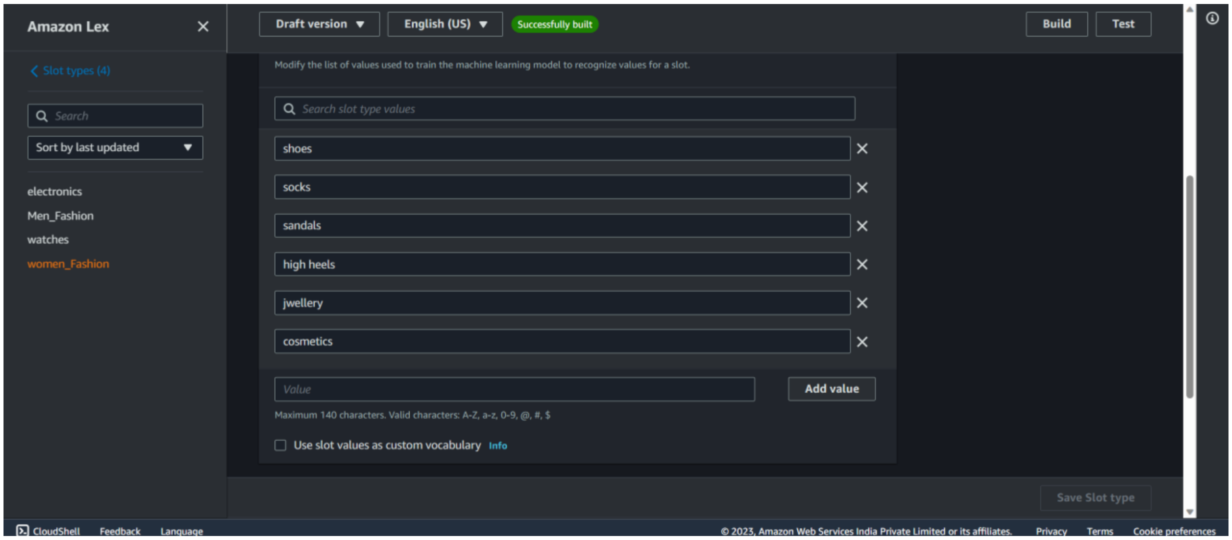
Amazon Lex resolves the slot values in an utterance to only the values you provide, or it expands the resolution to related or similar values.

● Expand values (default)
Values used as training data.

○ Restrict to slot values
Use only values provided.

Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.



Modify the list of values used to train the machine learning model to recognize values for a slot.

- shoes
- socks
- sandals
- high heels
- jwellery
- cosmetics

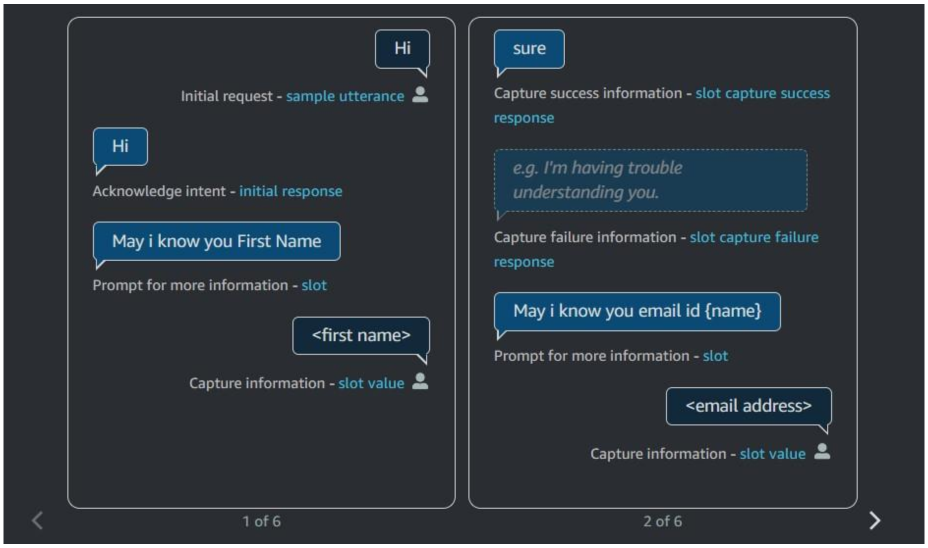
Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

After All setting done click on build button:

This will build the bot and if you want to test the bot after bot then click on test to run the bot but before that remove all errors and just check on conversational bot.

Conversational Flow:



Hi

Initial request - sample utterance

Hi

Acknowledge intent - initial response

May i know you First Name

Prompt for more information - slot

<first name>

Capture information - slot value

1 of 6

sure

Capture success information - slot capture success response

e.g. I'm having trouble understanding you.

Capture failure information - slot capture failure response

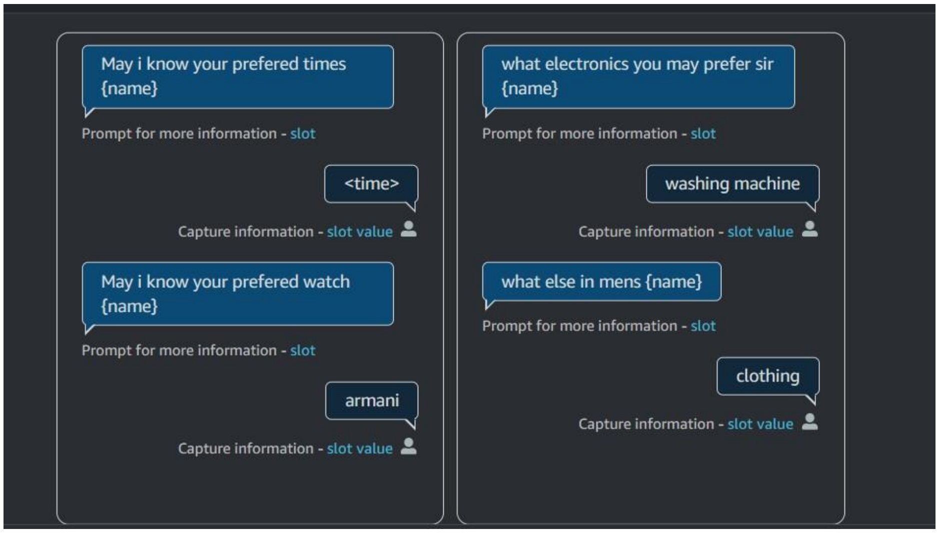
May i know you email id {name}

Prompt for more information - slot

<email address>

Capture information - slot value

2 of 6



May i know your prefered times {name}

Prompt for more information - slot

<time>

Capture information - slot value

May i know your prefered watch {name}

Prompt for more information - slot

armani

Capture information - slot value

what electronics you may prefer sir {name}

Prompt for more information - slot

washing machine

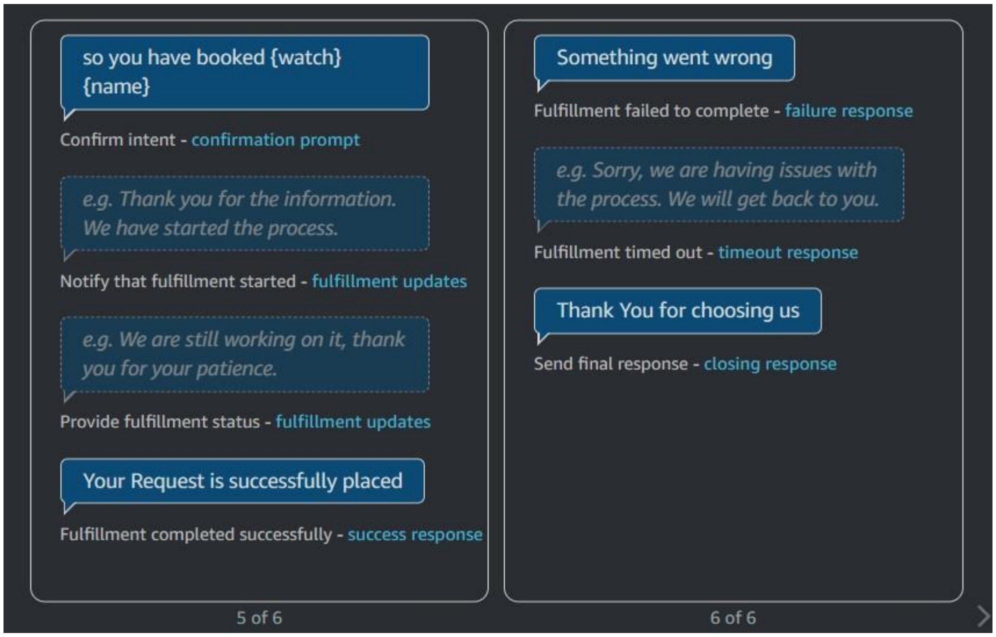
Capture information - slot value

what else in mens {name}

Prompt for more information - slot

clothing

Capture information - slot value



so you have booked {watch} {name}

Confirm intent - confirmation prompt

e.g. Thank you for the information. We have started the process.

Notify that fulfillment started - fulfillment updates

e.g. We are still working on it, thank you for your patience.

Provide fulfillment status - fulfillment updates

Your Request is successfully placed

Fulfillment completed successfully - success response

Something went wrong

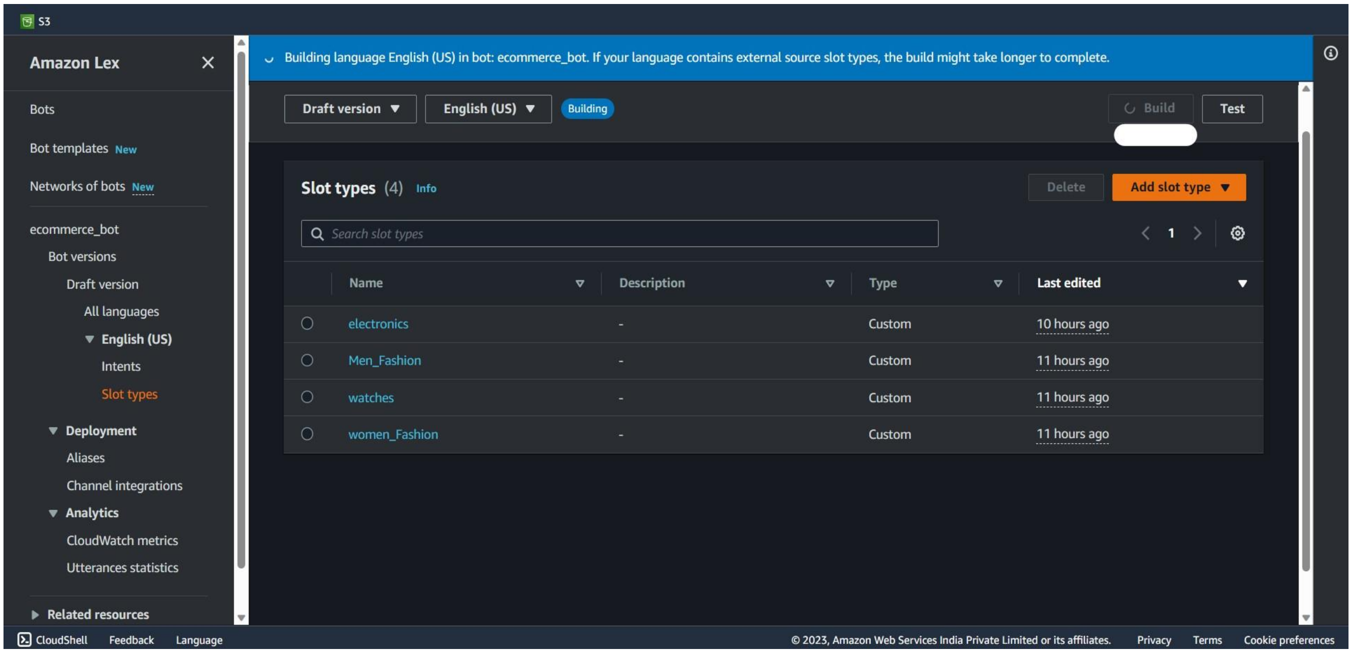
Fulfillment failed to complete - failure response

e.g. Sorry, we are having issues with the process. We will get back to you.

Fulfillment timed out - timeout response

Thank You for choosing us

Send final response - closing response



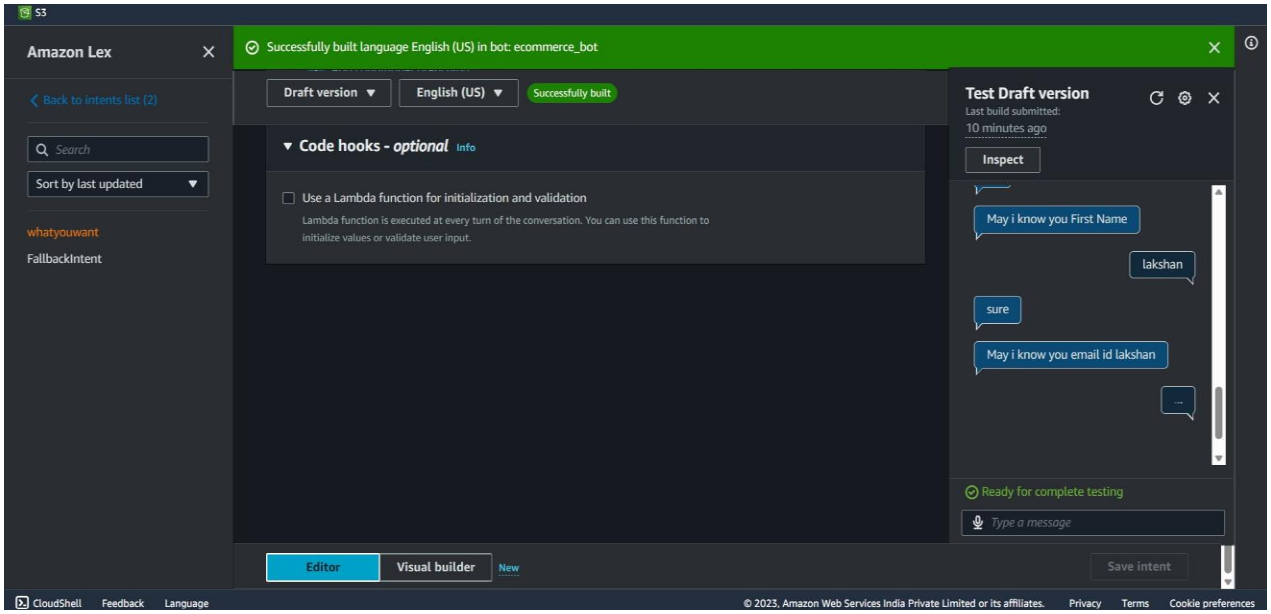
Building language English (US) in bot: ecommerce_bot. If your language contains external source slot types, the build might take longer to complete.

Slot types (4) Info

	Name	Description	Type	Last edited
○	electronics	-	Custom	10 hours ago
○	Men_Fashion	-	Custom	11 hours ago
○	watches	-	Custom	11 hours ago
○	women_Fashion	-	Custom	11 hours ago

© 2023, Amazon Web Services India Private Limited or its affiliates.

Audio::



Successfully built language English (US) in bot: ecommerce_bot

□ Use a Lambda function for initialization and validation

Lambda function is executed at every turn of the conversation. You can use this function to initialize values or validate user input.

May i know you First Name

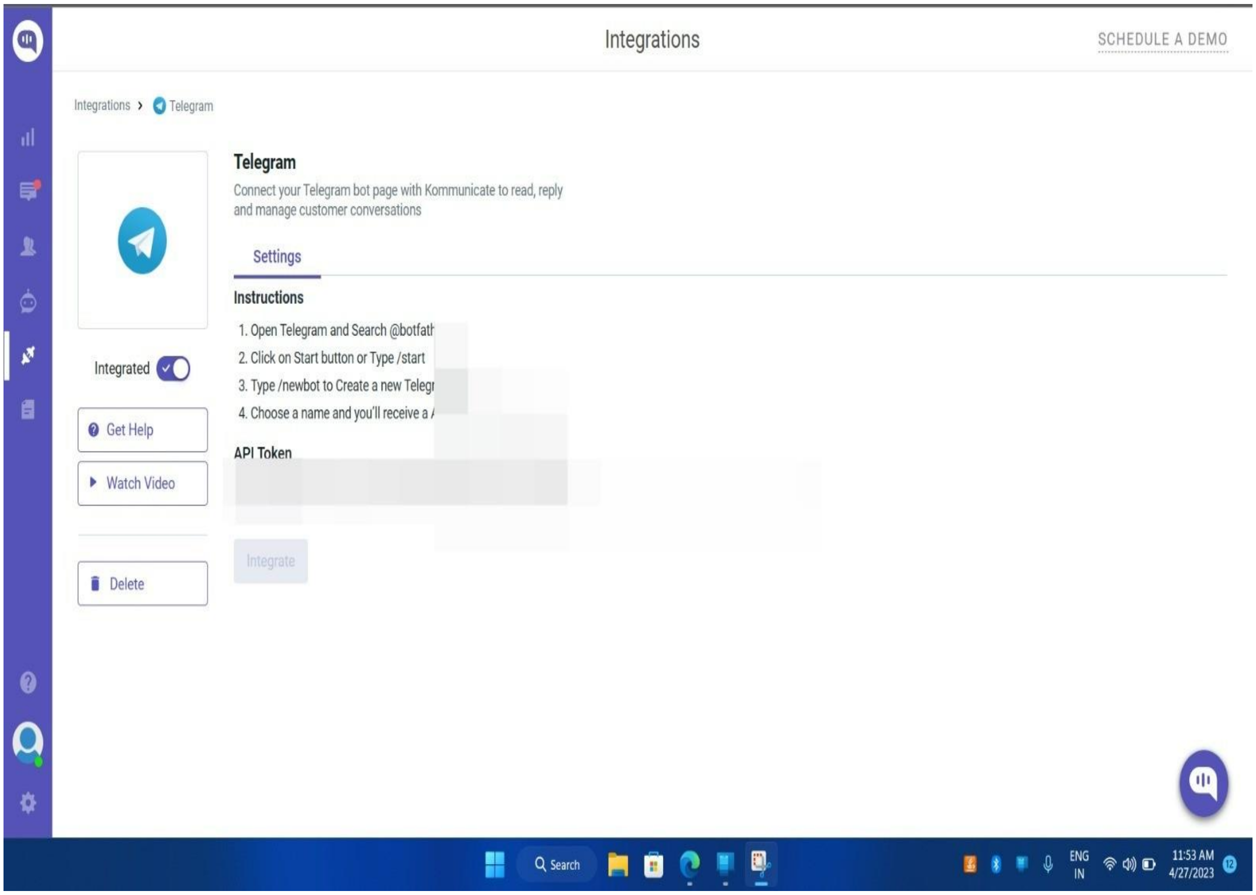
lakshan

sure

May i know you email id lakshan

...

Integration with telegram:



Telegram

Connect your Telegram bot page with Kommunicate to read, reply and manage customer conversations

Instructions

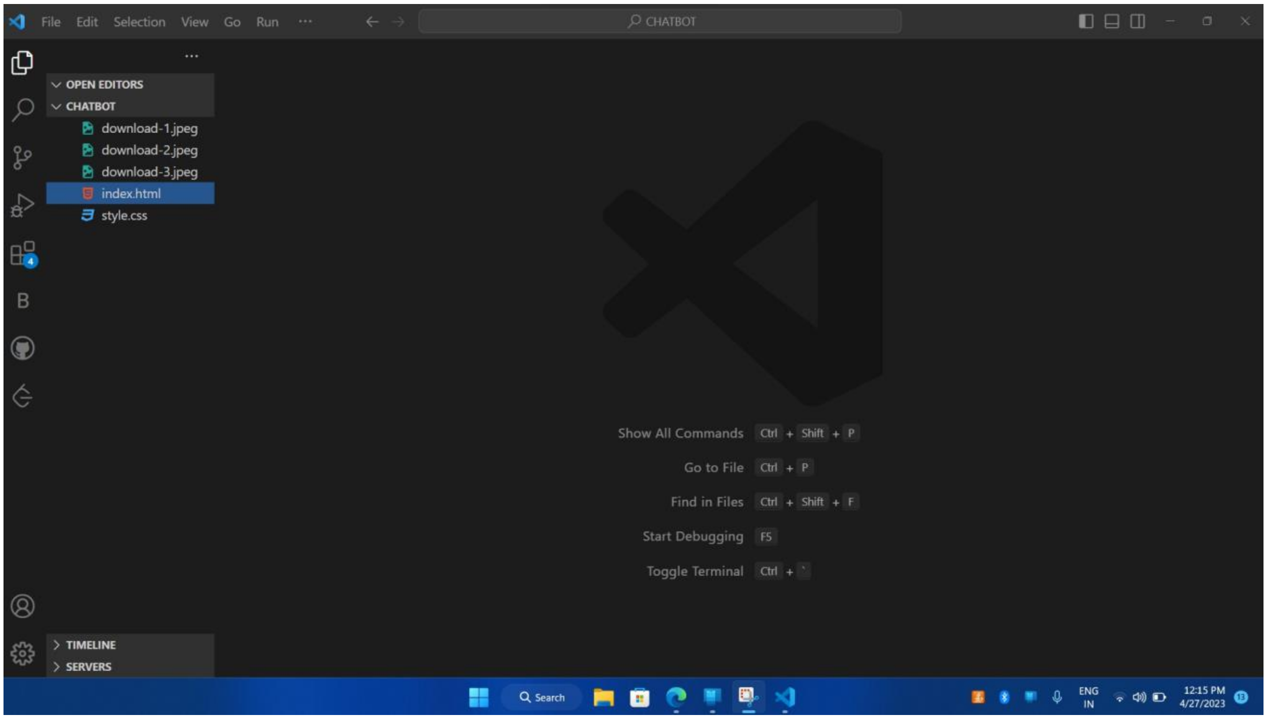
1. Open Telegram and Search @botfath
2. Click on Start button or Type /start
3. Type /newbot to Create a new Telegr
4. Choose a name and you'll receive a /

API Token

[redacted]

Integration with Our Website:

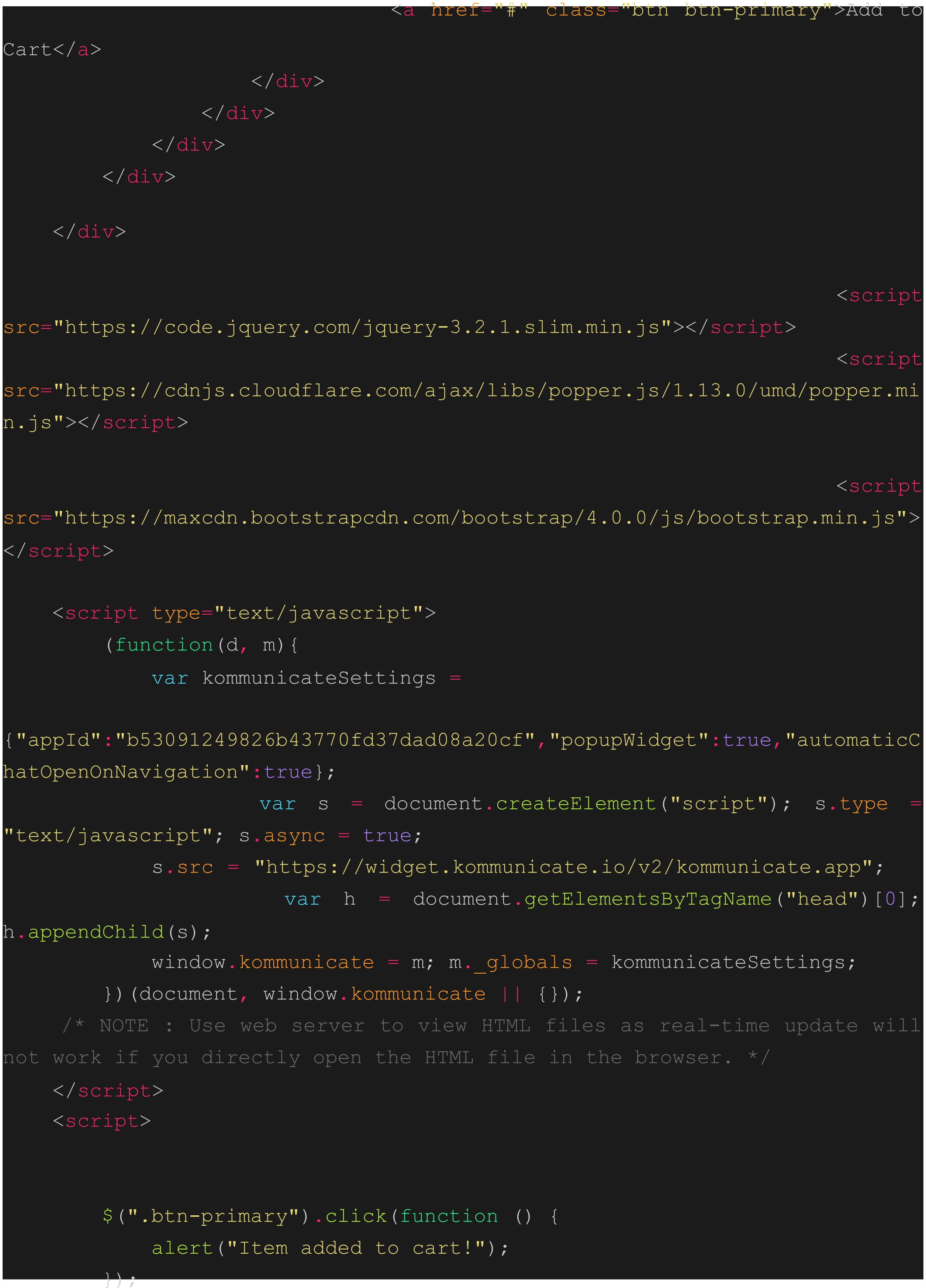
Project Directory:



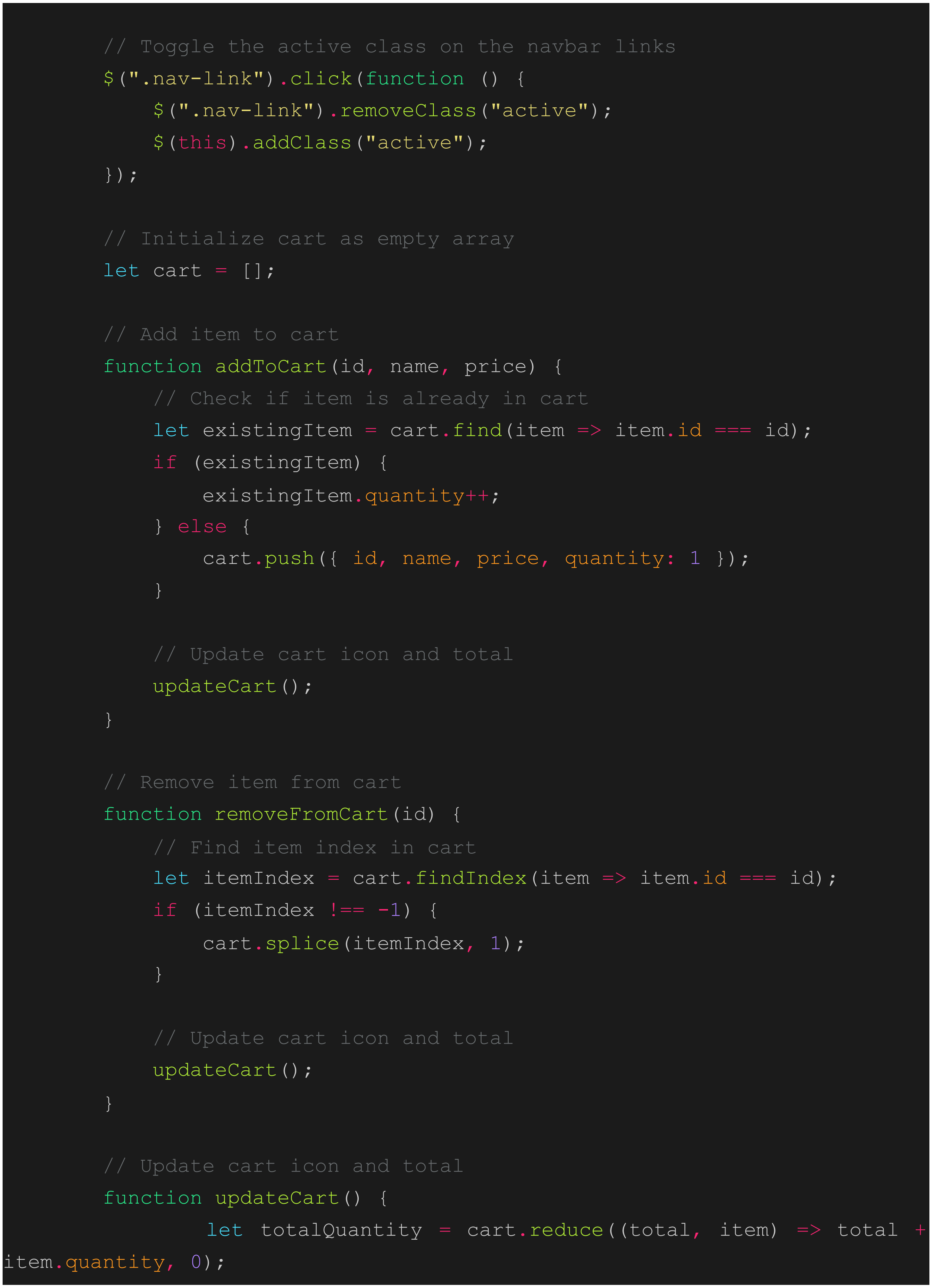
Index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Ecommerce Website</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="#">Ecommerce Website</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="#">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Shop</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Cart</a>
                </li>
            </ul>
        </div>
    </nav>

    <div class="container">
```

```
        <h1>Welcome to our Ecommerce Website!</h1>
        <p>Explore our collection of products.</p>

        <div class="row">
            <div class="col-md-4">
                <div class="card">
                        <img class="card-img-top" src="/download-1.jpeg"
alt="Product 1">
                    <div class="card-body">
                        <h5 class="card-title">Product 1</h5>
                            <p class="card-text">This is a description of
Product 1.</p>
                             <a href="#" class="btn btn-primary" onlick =
"addToCart()">Add to Cart</a>
                    </div>
                </div>
            </div>

            <div class="col-md-4">
                <div class="card">
                        <img class="card-img-top" src="/download-2.jpeg"
alt="Product 2">
                    <div class="card-body">
                        <h5 class="card-title">Product 2</h5>
                            <p class="card-text">This is a description of
Product 2.</p>
                               <a href="#" class="btn btn-primary">Add to
Cart</a>

                    </div>
                </div>
            </div>

            <div class="col-md-4">
                <div class="card">
                        <img class="card-img-top" src="/download-3.jpeg"
alt="Product 3">
                    <div class="card-body">
                        <h5 class="card-title">Product 3</h5>
                            <p class="card-text">This is a description of
Product 3.</p>
```



```
                               <a href="#" class="btn btn-primary">Add to
Cart</a>
                    </div>
                </div>
            </div>
        </div>

    </div>

                                                                   <script
src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
                                                                   <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.13.0/umd/popper.mi
n.js"></script>

                                                                   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>

    <script type="text/javascript">
        (function(d, m){
            var kommunicateSettings =

{"appId":"b53091249826b43770fd37dad08a20cf","popupWidget":true,"automaticC
hatOpenOnNavigation":true};
                    var s = document.createElement("script"); s.type =
"text/javascript"; s.async = true;
            s.src = "https://widget.kommunicate.io/v2/kommunicate.app";
                      var h = document.getElementsByTagName("head")[0];
h.appendChild(s);
            window.kommunicate = m; m._globals = kommunicateSettings;
        })(document, window.kommunicate || {});
     /* NOTE : Use web server to view HTML files as real-time update will
not work if you directly open the HTML file in the browser. */
    </script>
    <script>


        $(".btn-primary").click(function () {
            alert("Item added to cart!");
        });
```



```
        // Toggle the active class on the navbar links
        $(".nav-link").click(function () {
            $(".nav-link").removeClass("active");
            $(this).addClass("active");
        });

        // Initialize cart as empty array
        let cart = [];

        // Add item to cart
        function addToCart(id, name, price) {
            // Check if item is already in cart
            let existingItem = cart.find(item => item.id === id);
            if (existingItem) {
                existingItem.quantity++;
            } else {
                cart.push({ id, name, price, quantity: 1 });
            }

            // Update cart icon and total
            updateCart();
        }

        // Remove item from cart
        function removeFromCart(id) {
            // Find item index in cart
            let itemIndex = cart.findIndex(item => item.id === id);
            if (itemIndex !== -1) {
                cart.splice(itemIndex, 1);
            }

            // Update cart icon and total
            updateCart();
        }

        // Update cart icon and total
        function updateCart() {
                let totalQuantity = cart.reduce((total, item) => total +
item.quantity, 0);
```

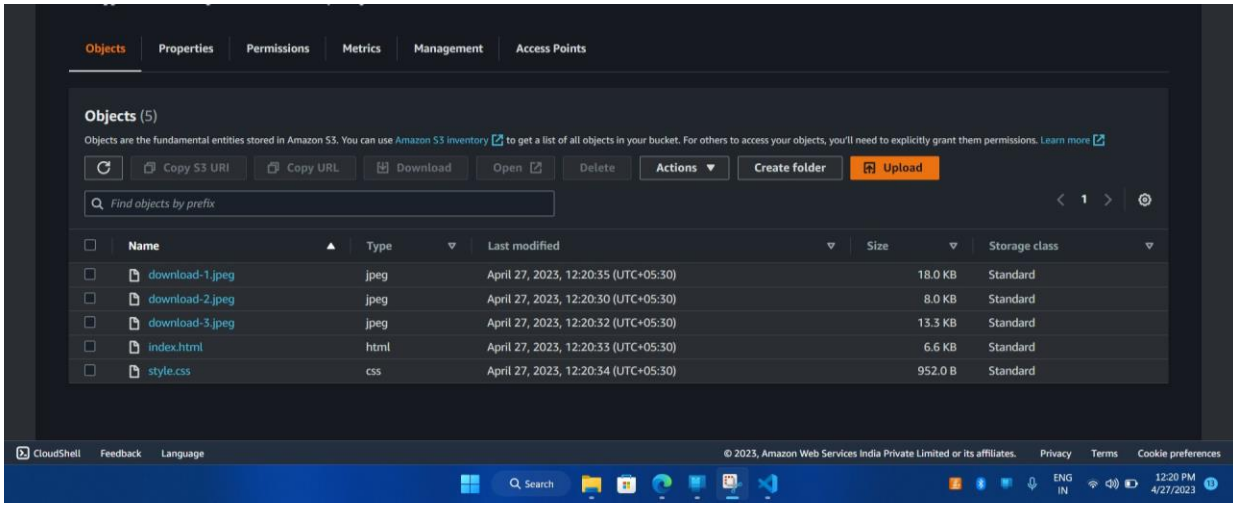
```
                  let totalPrice = cart.reduce((total, item) => total +
item.price * item.quantity, 0);

            // Update cart icon
            let cartIcon = document.querySelector("#cart-icon");
            cartIcon.dataset.quantity = totalQuantity;

            // Update cart total
            let cartTotal = document.querySelector("#cart-total");
            cartTotal.innerText = totalPrice.toFixed(2);

            // Update cart dropdown
            let cartDropdown = document.querySelector("#cart-dropdown");
            cartDropdown.innerHTML = "";
            if (cart.length === 0) {
                   cartDropdown.innerHTML = "<p class='dropdown-item'>Your
cart is empty</p>";
            } else {
                cart.forEach(item => {
                    let itemHtml = `
        <div class='dropdown-item'>
                     <button class='btn btn-danger btn-sm float-right'
onclick='removeFromCart(${item.id})'>Remove</button>
          <h6>${item.name}</h6>
          <p>${item.quantity} x ${item.price.toFixed(2)}</p>
        </div>
      `;
                    cartDropdown.innerHTML += itemHtml;
                });

                cartDropdown.innerHTML += `
      <div class='dropdown-divider'></div>
      <div class='dropdown-item'>
                                                    <h6>Total:      <span
class='float-right'>$${totalPrice.toFixed(2)}</span></h6>
      </div>
    `;
            }
        }

    </script>
```

```
</body>
</html>
```

Style.css

```
body {
    padding-top: 70px;
  }

  .card {
    margin-bottom: 20px;
    position: relative;
    overflow: hidden;
    cursor: pointer;
  }

  .card:hover .card-img-top {
    transform: scale(1.1);
    transition: transform 0.5s ease;
  }

  .card:hover .card-body {
    background-color: rgba(0, 0, 0, 0.7);
    transition: background-color 0.5s ease;
  }

  .card-img-top {
    height: 200px;
    object-fit: cover;
    transition: transform 0.5s ease;
  }

  .card-body {
    position: absolute;
    bottom: 0;
    left: 0;
    width: 100%;
    padding: 10px;
    color: #fff;
    background-color: rgba(0, 0, 0, 0.5);
    transition: background-color 0.5s ease;
  }

  .navbar-brand {
    font-size: 1.5rem;
  }

  .btn-primary {
    background-color: #007bff;
    border-color: #007bff;
  }

  .btn-primary:hover {
    background-color: #0069d9;
    border-color: #0062cc;
  }
```

Then Go to S3 bucket and put all files into s3 bucket

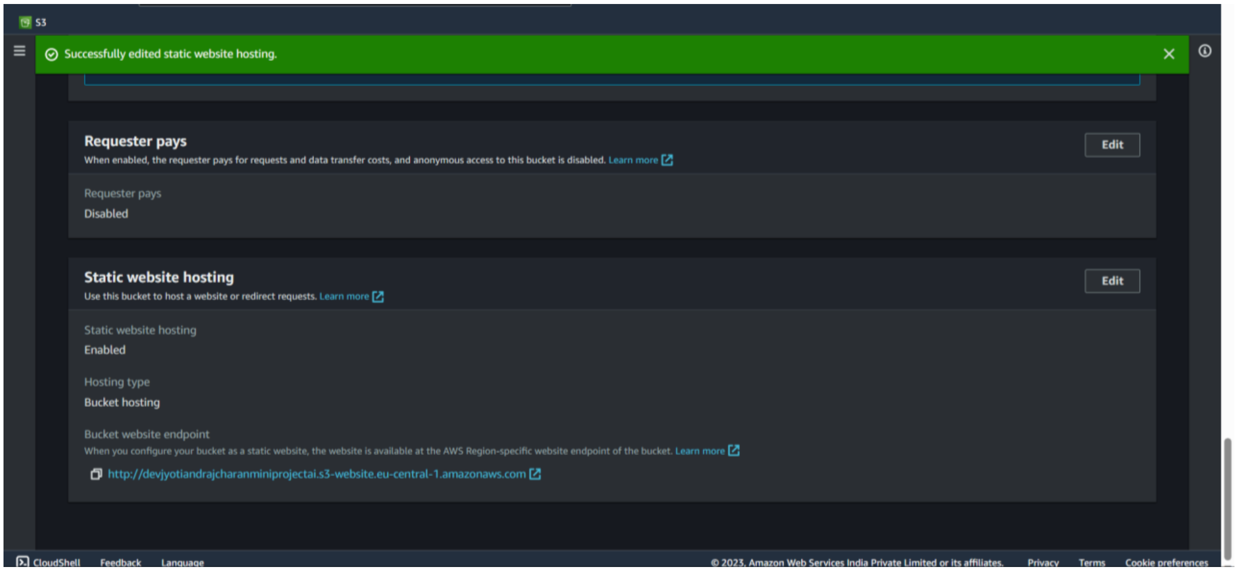


Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

☐	Name ▲	Type	Last modified	Size	Storage class
☐	download-1.jpeg	jpeg	April 27, 2023, 12:20:35 (UTC+05:30)	18.0 KB	Standard
☐	download-2.jpeg	jpeg	April 27, 2023, 12:20:30 (UTC+05:30)	8.0 KB	Standard
☐	download-3.jpeg	jpeg	April 27, 2023, 12:20:32 (UTC+05:30)	13.3 KB	Standard
☐	index.html	html	April 27, 2023, 12:20:33 (UTC+05:30)	6.6 KB	Standard
☐	style.css	css	April 27, 2023, 12:20:34 (UTC+05:30)	952.0 B	Standard

Then Go to permissions and enable static website hosting:



Successfully edited static website hosting.

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more

Requester pays
Disabled

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more

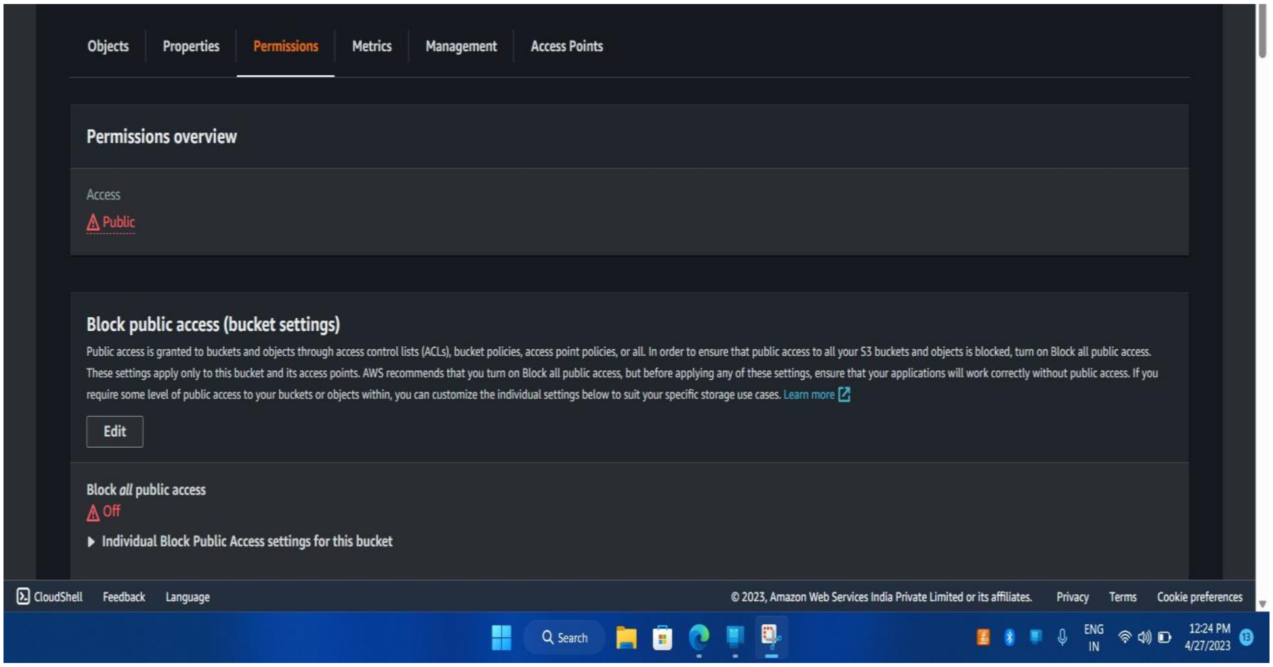
http://devjyotiandrajcharanminiprojectai.s3-website.eu-central-1.amazonaws.com

© 2023, Amazon Web Services India Private Limited or its affiliates.

Then Just go to permissions and remove block access for public and put bucket policy

Bucket policy in json by aws

```
{
  "Id": "Policy1682578402468",
  "Version": "2012-10-17",
  "Statement": [
    {Sid": "Stmt1682578400752",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::pradeepandkartheekminiprojectai",
      "Principal": "*"
    }
  ]
}
```



Permissions overview

Access

⚠ Public

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access

⚠ Off

▶ Individual Block Public Access settings for this bucket

© 2023, Amazon Web Services India Private Limited or its affiliates.

Conclusion and future enhancement

To summarize, Amazon Lex is an extremely powerful tool for creating chatbots and conversational interfaces. Its natural language processing (NLP) features allow developers to design bots that can recognise and interpret user input and provide relevant answers and actions.

Some potential future Amazon Lex chatbot features include:

1) Context awareness: Amazon Lex chatbots might be improved to better comprehend the context of a discussion, allowing them to give consumers with more personalised and accurate replies.

2) Handle for many languages: As organizations grow more global, the ability to handle different languages becomes increasingly crucial. Amazon Lex chatbots might be extended to handle other languages, allowing businesses to reach out to a larger audience.

3) Third-party platform integration: Amazon Lex chatbots might be expanded to interact with third-party platforms like as social media and messaging applications, allowing companies to communicate with consumers through their chosen channels.

4) augmented analytics: Amazon Lex chatbots might be augmented with sophisticated analytics capabilities, allowing businesses to get deeper insights into user interactions and behaviour, which they could then use to optimise their chatbot and improve the user experience.

5) Overall, Amazon Lex chatbots offer a lot of promise for organisations trying to improve customer interaction and optimise processes. They are expected to become even more important in the future as they continue to develop and improve.

References:

1) https://docs.aws.amazon.com/lex/index.html

2) Amazon Simple Storage Service Documentation

3) Developer Docs | Kommunicate · Configure Kommunicate's support platform for your product or site. Find integration and implementation guide along with code samples.