Building A ChatBot

With

Amazon Lex

1. **Chatbot Fundamentals**
   1. **What is a chatbot?**

**A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent.Today, most chatbots are accessed on-line via website popups, or through virtual assistants such as Google Assistant, Amazon Alexa, or messaging apps such as Facebook Messenger or WeChat.**



Image Courtesy: <https://expertsystem.com/chatbot/>

* 1. **Uses of chatbots:**

**Here are some of the use case of chatbot:-**

1. **Chatbots Answer Questions And Inquiries**
2. **Book Tickets To Events/Shows With Chatbots**
3. **Use Chatbots To Find Products, Check Inventory and Recommend Items**
4. **Chatbots To Build Remarkable Customer Experience**
5. **Chatbots Can Confirm Orders And Track Shipping**
6. **Chatbots Can Do Quizzes, Promotions, And Contests With Customers**
7. **Chatbots Become Personal Shopping Assistants**
   1. **Advantages of Chatbots**
8. **1- 24 hour availability**
9. **Keeping Up with the Trends: Being Present on Messaging Platforms**
10. **Improved Customer Service**
11. **Increased Customer Engagement**
12. **Cost Savings**
13. **Management of multiple clients**
14. **Gaining a deeper understanding of customers**

**What Is Amazon Lex?**

Amazon Lex is an AWS service for building conversational interfaces for applications using voice and text. With Amazon Lex, the same conversational engine that powers Amazon Alexa is now available to any developer, enabling you to build sophisticated, natural language chatbots into your new and existing applications. Amazon Lex provides the deep functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR) so you can build highly engaging user experiences with lifelike, conversational interactions, and create new categories of products.

Amazon Lex enables any developer to build conversational chatbots quickly. With Amazon Lex, no deep learning expertise is necessary—to create a bot, you just specify the basic conversation flow in the Amazon Lex console. Amazon Lex manages the dialogue and dynamically adjusts the responses in the conversation. Using the console, you can build, test, and publish your text or voice chatbot. You can then add the conversational interfaces to bots on mobile devices, web applications, and chat platforms (for example, Facebook Messenger).

Amazon Lex provides pre-built integration with AWS Lambda, and you can easily integrate with many other services on the AWS platform, including Amazon Cognito, AWS Mobile Hub, Amazon CloudWatch, and Amazon DynamoDB. Integration with Lambda provides bots access to pre-built serverless enterprise connectors to link to data in SaaS applications, such as Salesforce, HubSpot, or Marketo.

Some of the benefits of using Amazon Lex include:

* **Simplicity** – Amazon Lex guides you through using the console to create your own chatbot in minutes. You supply just a few example phrases, and Amazon Lex builds a complete natural language model through which the bot can interact using voice and text to ask questions, get answers, and complete sophisticated tasks.

* **Democratized deep learning technologies** – Powered by the same technology as Alexa, Amazon Lex provides ASR and NLU technologies to create a Speech Language Understanding (SLU) system. Through SLU, Amazon Lex takes natural language speech and text input, understands the intent behind the input, and fulfills the user intent by invoking the appropriate business function.

Speech recognition and natural language understanding are some of the most challenging problems to solve in computer science, requiring sophisticated deep learning algorithms to be trained on massive amounts of data and infrastructure. Amazon Lex puts deep learning technologies within reach of all developers, powered by the same technology as Alexa. Amazon Lex chatbots convert incoming speech to text and understand the user intent to generate an intelligent response, so you can focus on building your bots with differentiated value-add for your customers, to define entirely new categories of products made possible through conversational interfaces.

* **Seamless deployment and scaling** – With Amazon Lex, you can build, test, and deploy your chatbots directly from the Amazon Lex console. Amazon Lex enables you to easily publish your voice or text chatbots for use on mobile devices, web apps, and chat services (for example, Facebook Messenger). Amazon Lex scales automatically so you don’t need to worry about provisioning hardware and managing infrastructure to power your bot experience.

* **Built-in integration with the AWS platform** – Amazon Lex has native interoperability with other AWS services, such as Amazon Cognito, AWS Lambda, Amazon CloudWatch, and AWS Mobile Hub. You can take advantage of the power of the AWS platform for security, monitoring, user authentication, business logic, storage, and mobile app development.

* **Cost-effectiveness** – With Amazon Lex, there are no upfront costs or minimum fees. You are charged only for the text or speech requests that are made. The pay-as-you-go pricing and the low cost per request make the service a cost-effective way to build conversational interfaces. With the Amazon Lex free tier, you can easily try Amazon Lex without any initial investment.

**Amazon Lex: How It Works**

Amazon Lex enables you to build applications using a speech or text interface powered by the same technology that powers Amazon Alexa. Following are the typical steps you perform when working with Amazon Lex:

1. Create a bot and configure it with one or more intents that you want to support. Configure the bot so it understands the user's goal (intent), engages in conversation with the user to elicit information, and fulfills the user's intent.
2. Test the bot. You can use the test window client provided by the Amazon Lex console.
3. Publish a version and create an alias.
4. Deploy the bot. You can deploy the bot on platforms such as mobile applications or messaging platforms such as Facebook Messenger.

Before you get started, familiarize yourself with the following Amazon Lex core concepts and terminology:

* **Bot** – A bot performs automated tasks such as ordering a pizza, booking a hotel, ordering flowers, and so on. An Amazon Lex bot is powered by Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) capabilities, the same technology that powers Amazon Alexa.

Amazon Lex bots can understand user input provided with text or speech and converse in natural language. You can create Lambda functions and add them as code hooks in your intent configuration to perform user data validation and fulfillment tasks.

* **Intent** – An intent represents an action that the user wants to perform. You create a bot to support one or more related intents. For example, you might create a bot that orders pizza and drinks. For each intent, you provide the following required information:

* + Intent name– A descriptive name for the intent. For example, OrderPizza.
  + Sample utterances – How a user might convey the intent. For example, a user might say "Can I order a pizza please" or "I want to order a pizza".
  + How to fulfill the intent – How you want to fulfill the intent after the user provides the necessary information (for example, place order with a local pizza shop). We recommend that you create a Lambda function to fulfill the intent.

You can optionally configure the intent so Amazon Lex simply returns the information back to the client application to do the necessary fulfillment.

In addition to custom intents such as ordering a pizza, Amazon Lex also provides built-in intents to quickly set up your bot. For more information, see [Built-in Intents and Slot Types](https://docs.aws.amazon.com/lex/latest/dg/howitworks-builtins.html).

* **Slot** – An intent can require zero or more slots or parameters. You add slots as part of the intent configuration. At runtime, Amazon Lex prompts the user for specific slot values. The user must provide values for all *required* slots before Amazon Lex can fulfill the intent.

For example, the OrderPizza intent requires slots such as pizza size, crust type, and number of pizzas. In the intent configuration, you add these slots. For each slot, you provide slot type and a prompt for Amazon Lex to send to the client to elicit data from the user. A user can reply with a slot value that includes additional words, such as "large pizza please" or "let's stick with small." Amazon Lex can still understand the intended slot value.

* **Slot type** – Each slot has a type. You can create your custom slot types or use built-in slot types. For example, you might create and use the following slot types for the OrderPizza intent:

* + Size – With enumeration values Small, Medium, and Large.
  + Crust – With enumeration values Thick and Thin.

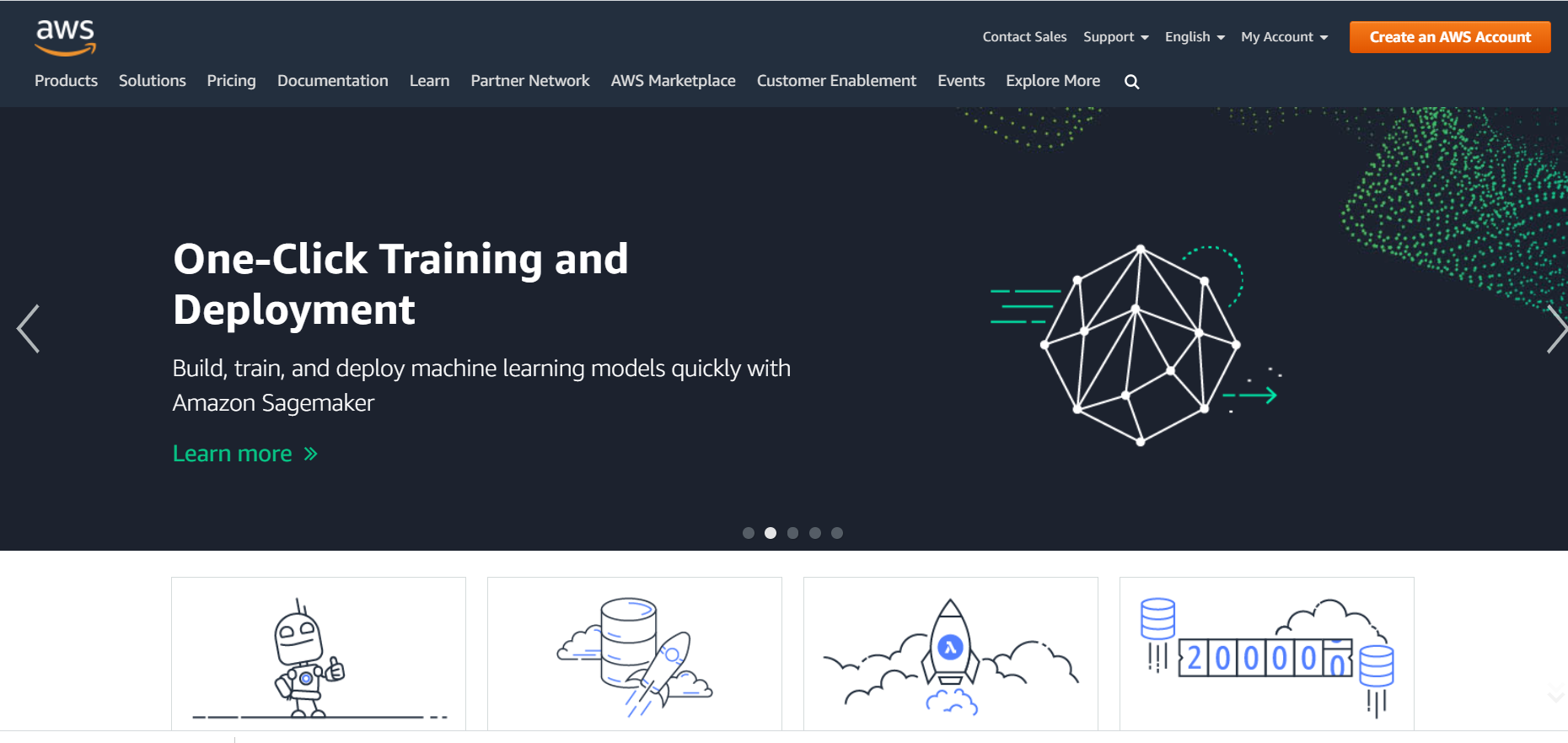
Amazon Lex also provides built-in slot types. For example, AMAZON.NUMBER is a built-in slot type that you can use for the number of pizzas ordered. For more information, see [Built-in Intents and Slot Types](https://docs.aws.amazon.com/lex/latest/dg/howitworks-builtins.html).

Currently, Amazon Lex supports only US English language. That is, Amazon Lex trains your bots to understand only US English.

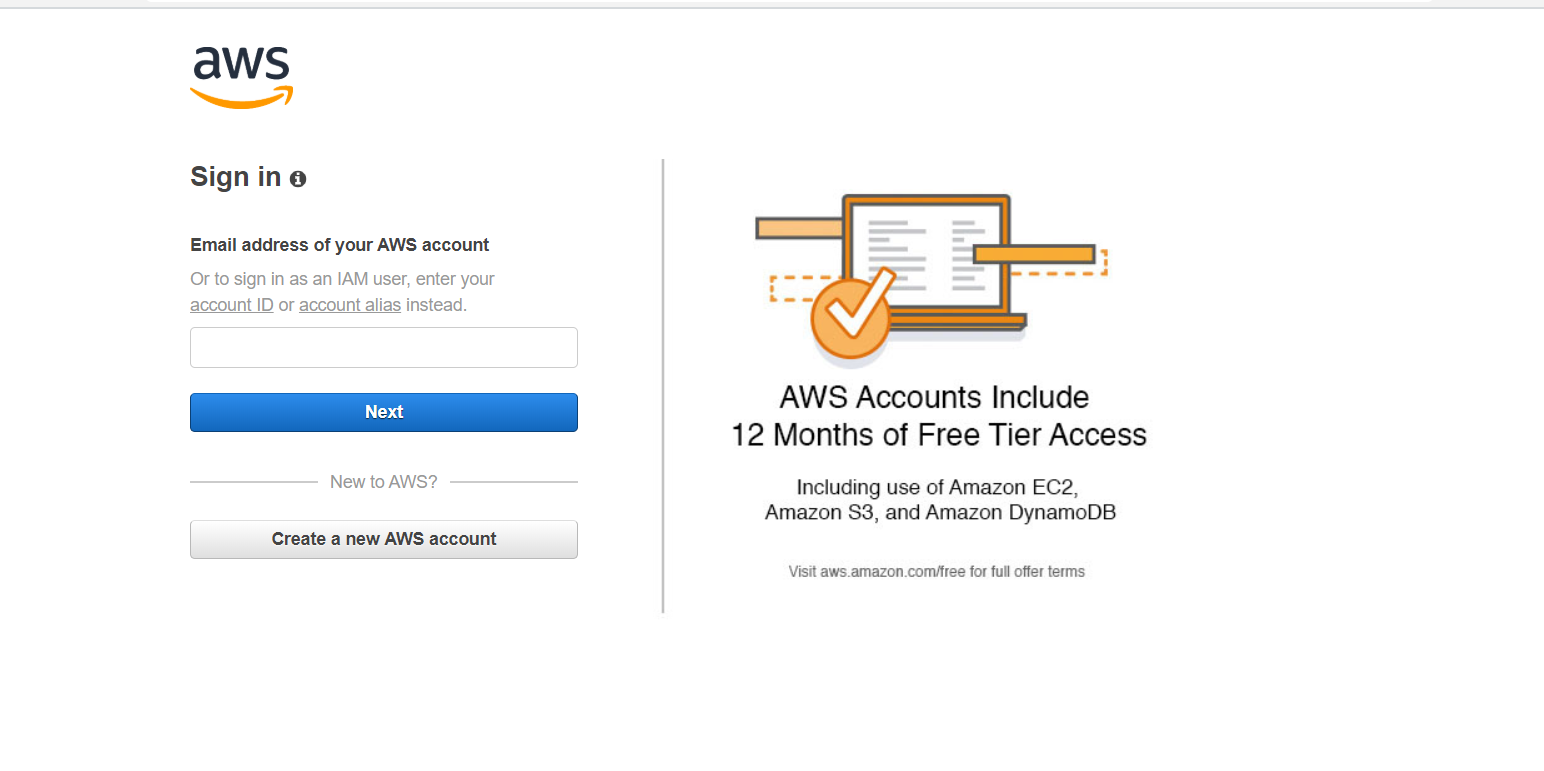
(Source: <https://docs.aws.amazon.com/lex/latest/dg/what-is.html>)

**Build your Chatbot:-**

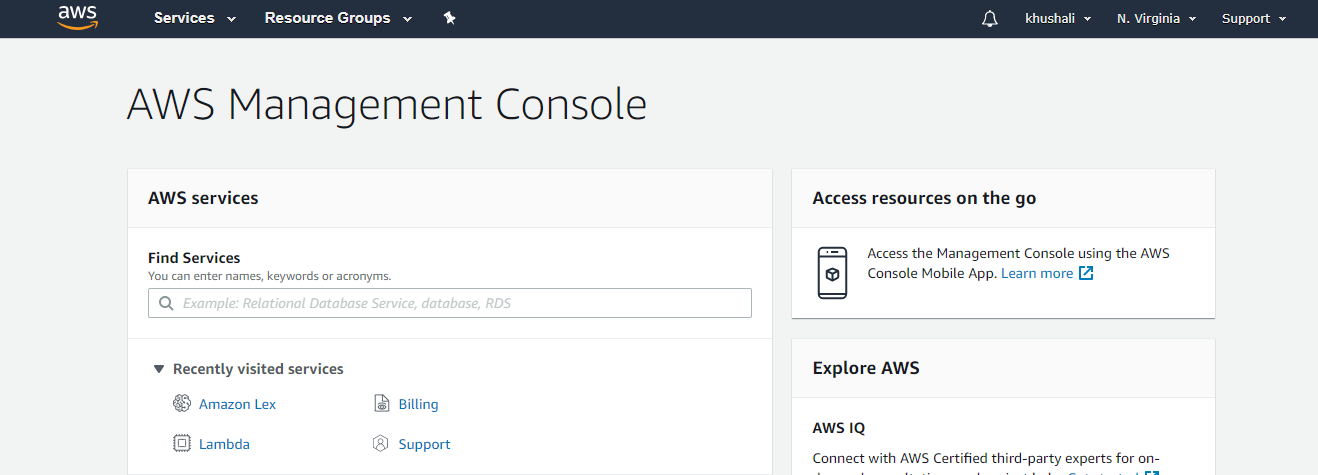
1) Login to AWS portal [**https://signin.aws.amazon.com/**](https://signin.aws.amazon.com/)



2) Click on “create an AWS Account”. Sign in with your credentials or create a new account.

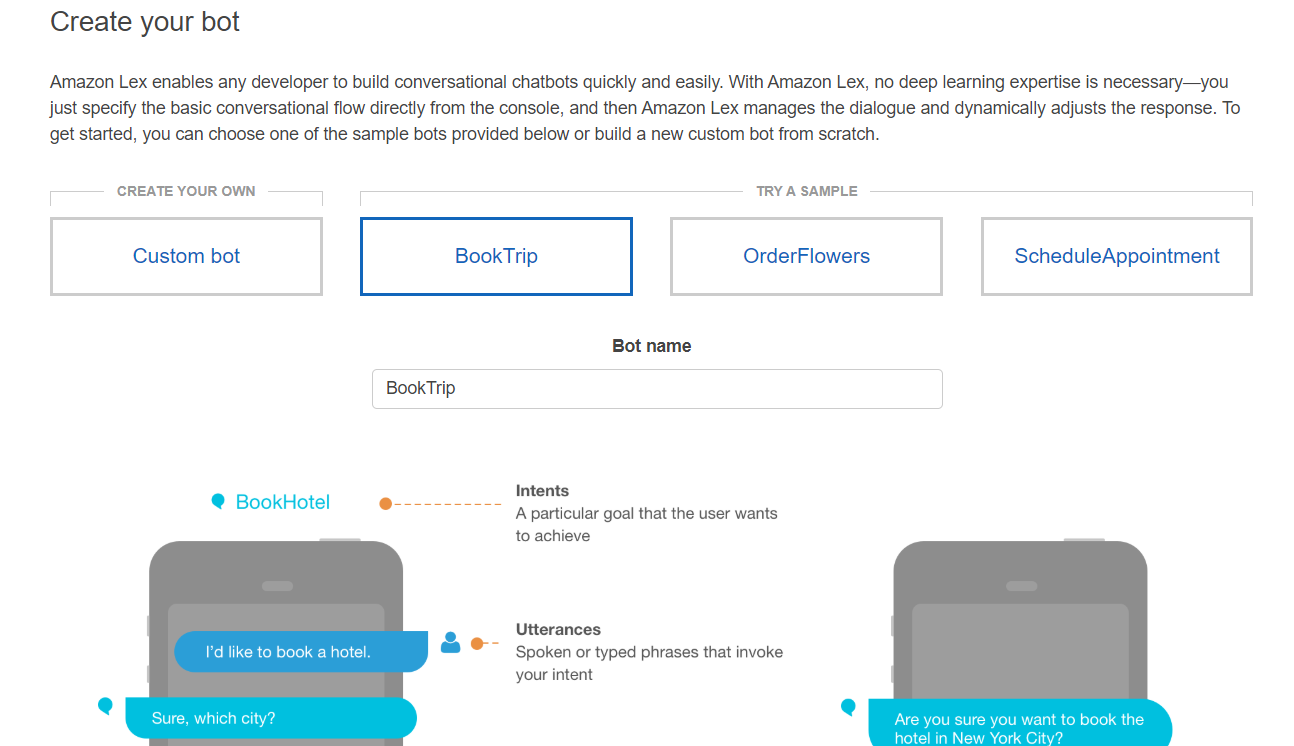


3) Once logged in, click on Amazon Lex to start creating your bot.



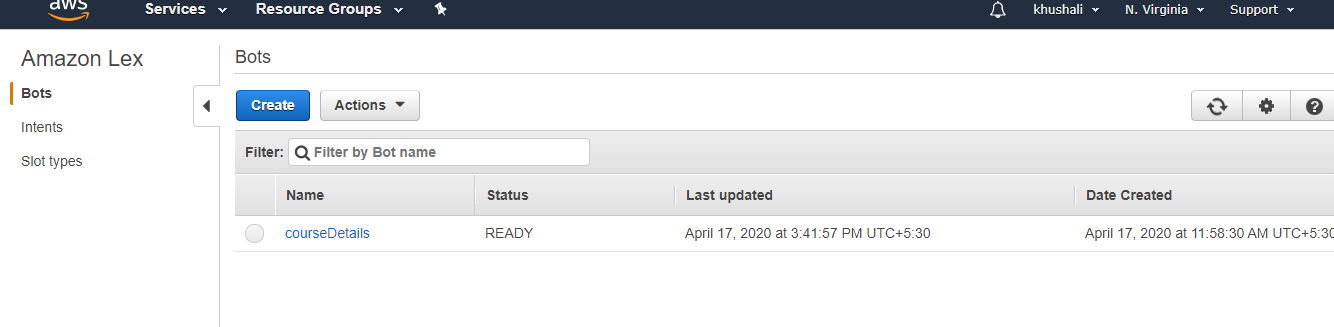
4) Click on create and create a new bot.

5) Once you click the create button, you will see a window like this. Click on the “custom bot” to make your own bot. The other bots are example bots and you can go through them to get a glance on some higher details.



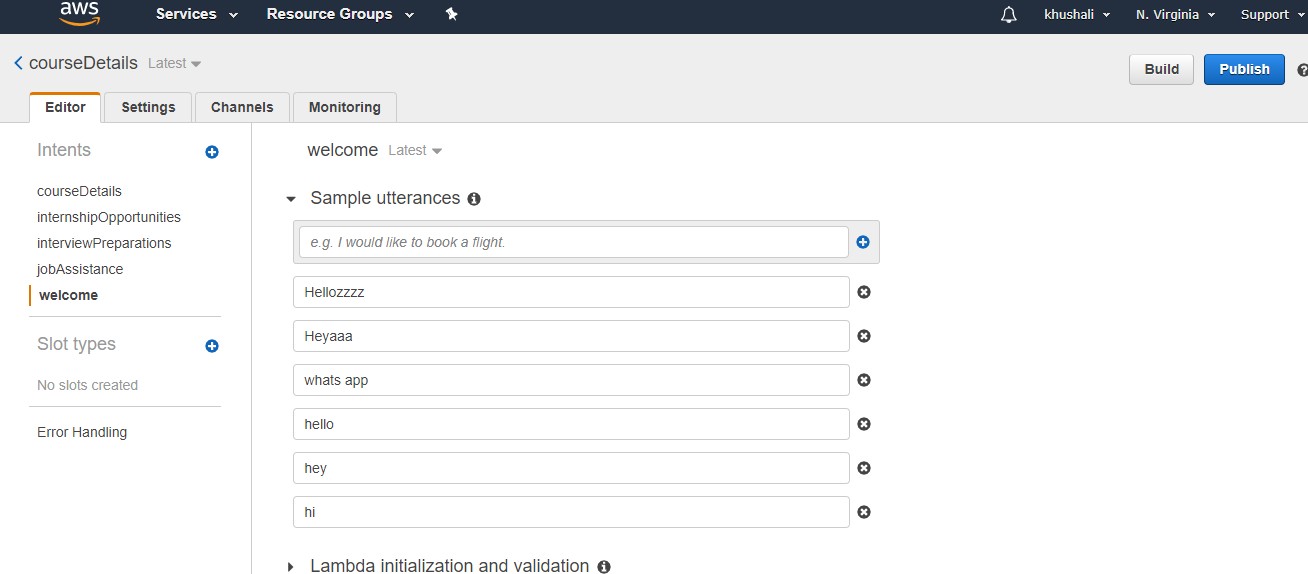
6) Fill up all the details and create on the create button.

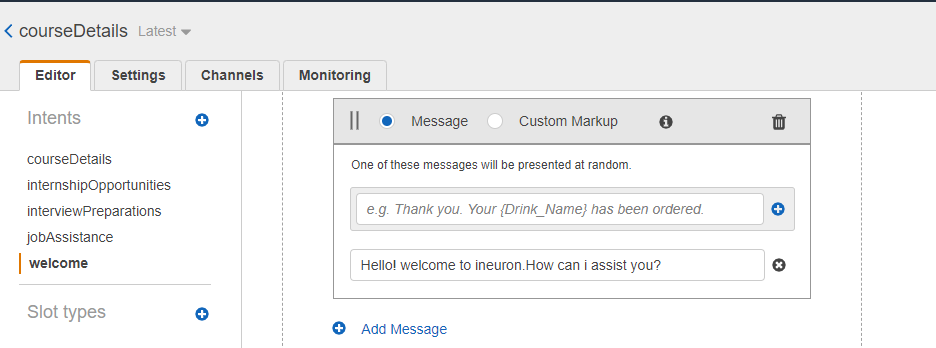
7) We have create a bot as follows:



8) We will start by creating intents for our bot. Click on the plus button next to intents.

We are going to create a “Welcome” intent which will greet the user on hearing/reading the defined utterances.

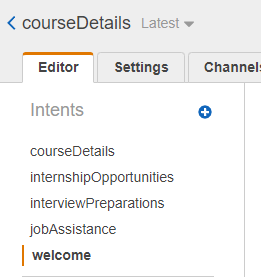




We are not filling the slots for this intent and keeping the “Fullfillment” as return parameters to client. We will see the use of slots and use of lambda function for the next intent.

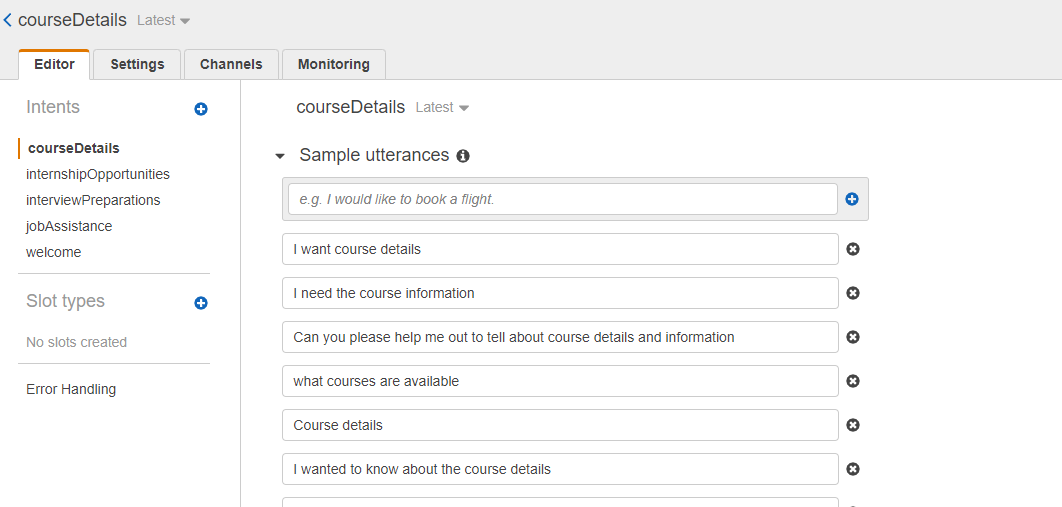
In the response section, we are giving the response that the bot is going to give after hearing all the utterances we have provided.

9) Similarly we are creating new intents as per our requirements that we need the bot to respond to.

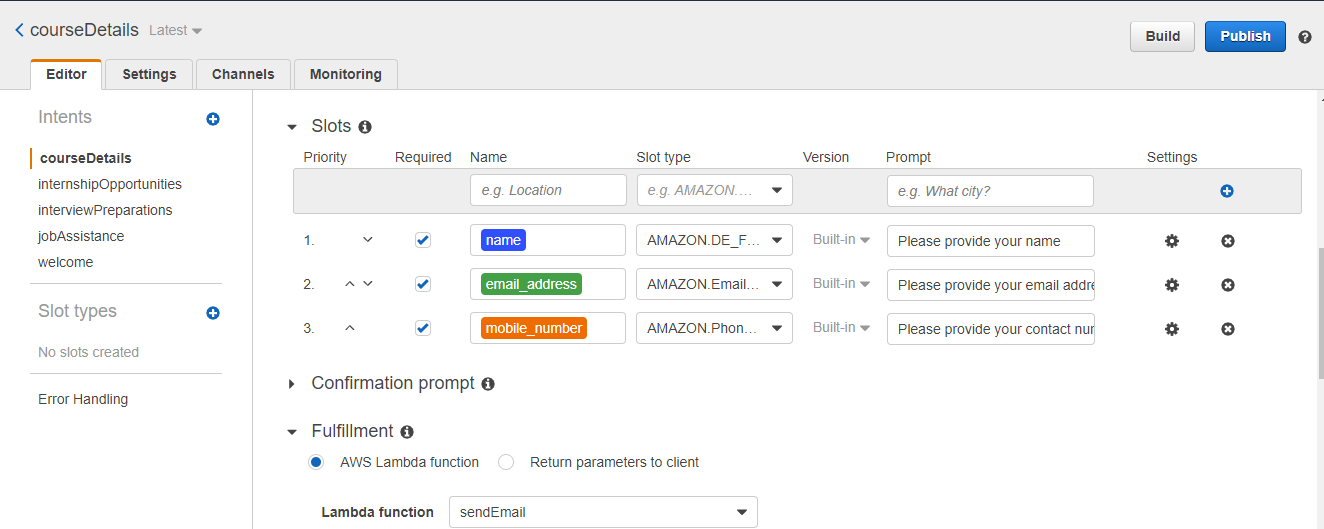


10) let’s see the Fulfillment using lambda function. We are creating one intent with sample utterances and once our bot receives that intent, it will ask the user for its details like name, mobile number and email. Then it will send a mail to the user with the course details attached and it will also send an email to the support team with all the details of the user requesting a callback.

Let’s see how we are going to do it.



Once we have filled up the sample utterances. Let’s fill the slots that we will be prompting the user for getting the user details.

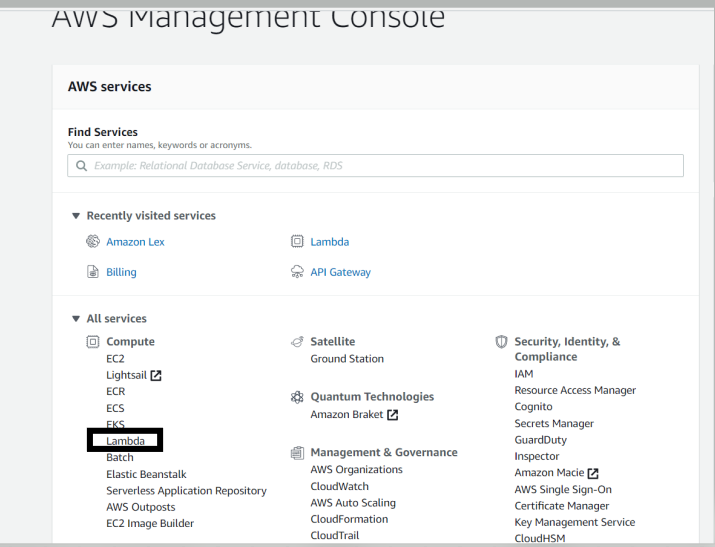


In the slot name we give the values that we want from the user. The slot\_type is the default validation parameter in Lex and we can use that to validate the values user gives against that slot. In the prompt we are going to prompt the user with the specific questions to get the slot values.

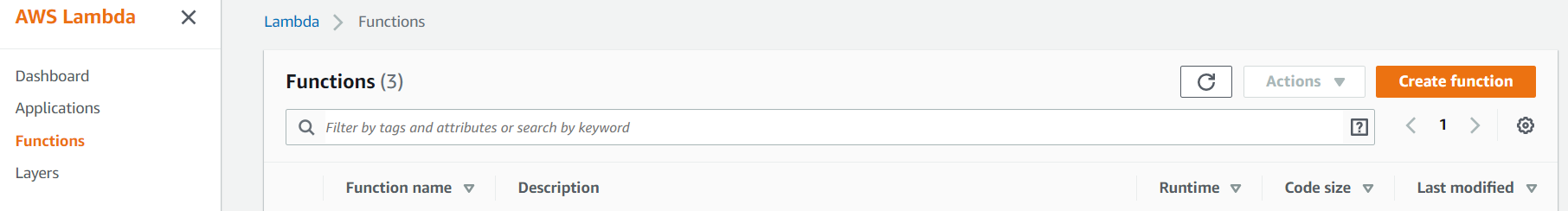
In the Fulfillment section, we are using Lambda function to send an email to the user on the given email id and to the support team with all the user details.

Let’s see how to create the lamda function.

Go to the AWS console and click on “Lambda” inside the ”compute” section.

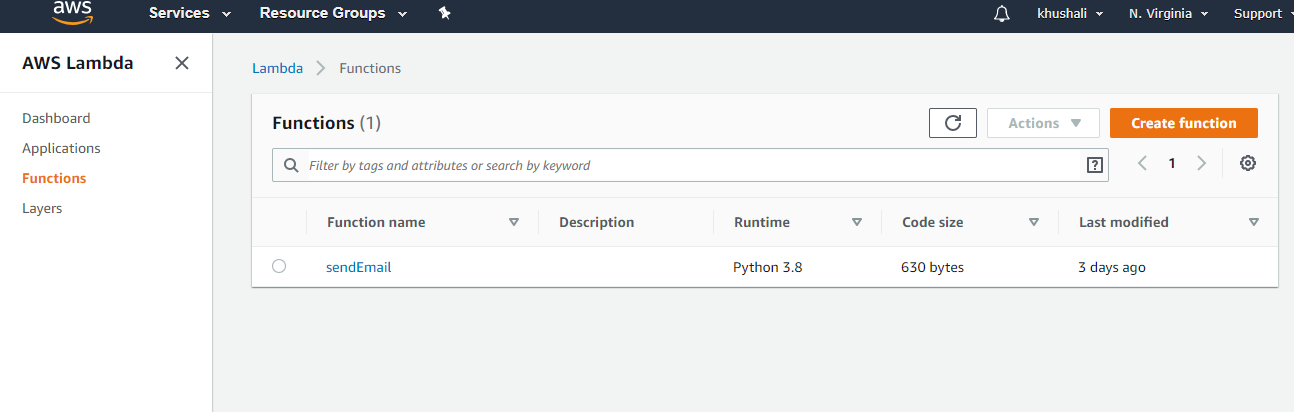


Click on the “create function” tab.

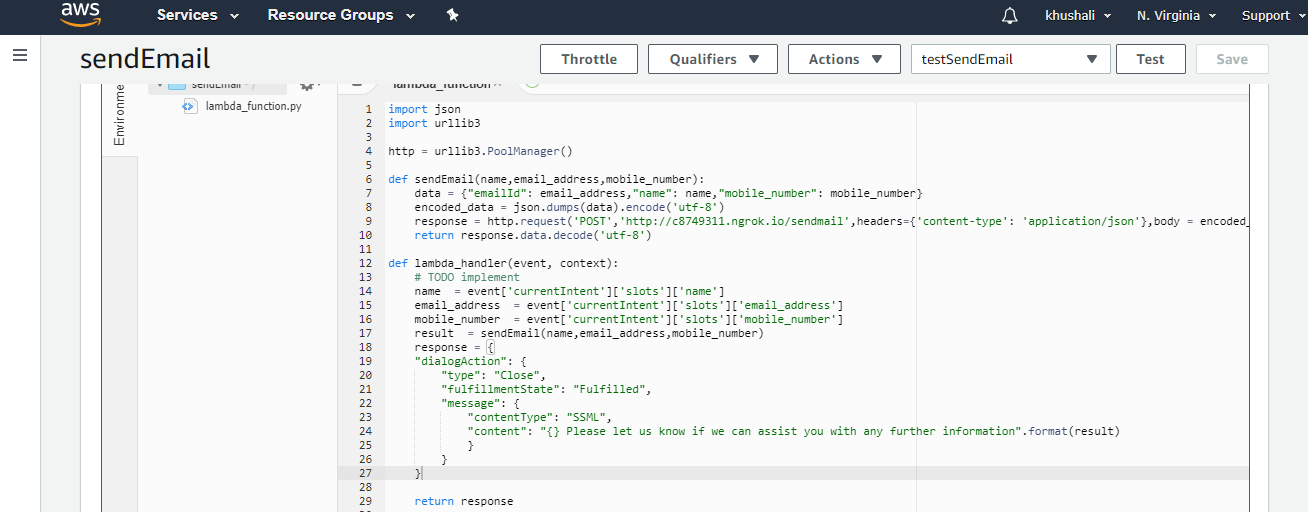


Given the name of the function and select the runtime as “python”, you can select other runtimes as well. Click on the create function.

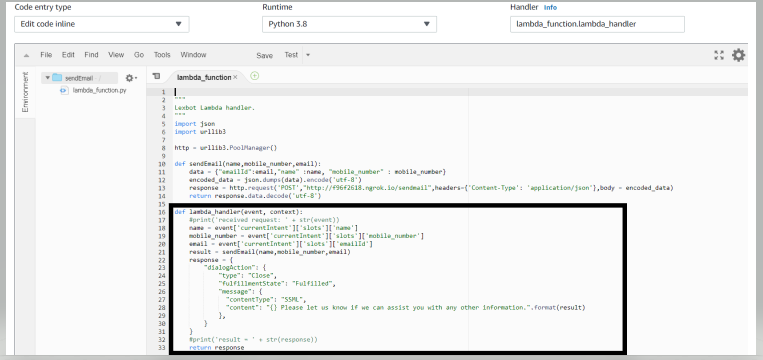
We have created a function named “sendEmail”. After clicking the “create function” button, you will see a page like this :



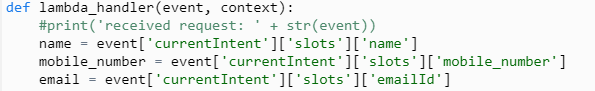
Go to the “Function code” section:



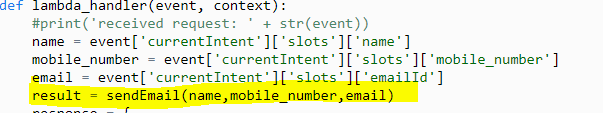
The function with name “lambda\_handler” is the function first called once an intent with fulfilment as lamda is called and the it is passed as the “event” argument in the function.



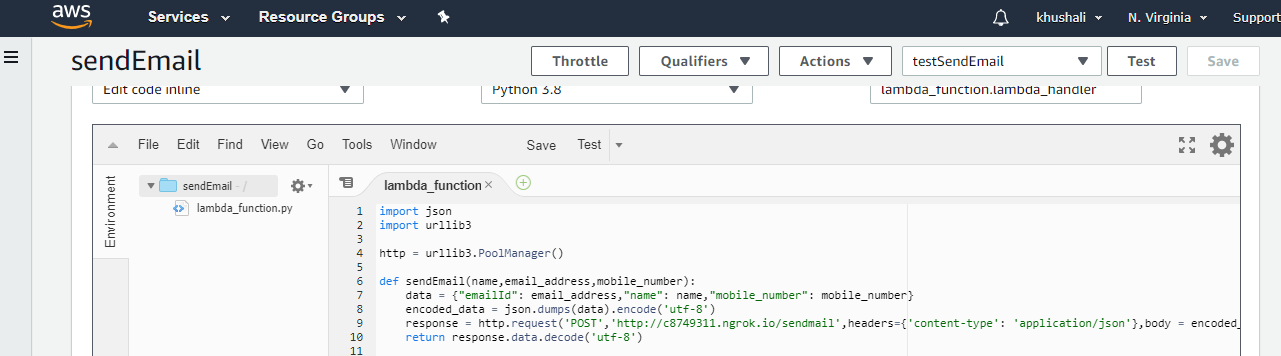
From the “event” argument we can derive our values in the slots like below :



Once we have got the values in our slots for the intent, we are going to pass these values as an argument in an another function we have written to send the mails.



Let’s see what does this function “sendEmail” does:

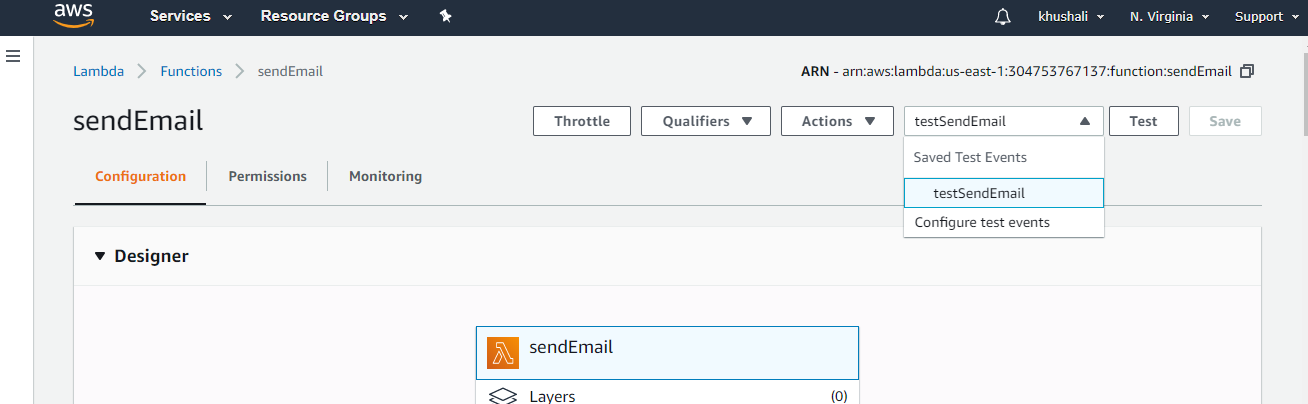


This function hits an “Api” using the post method. We pass all the user details in the api in a json format. The Api returns a message on successfully sending the mails.

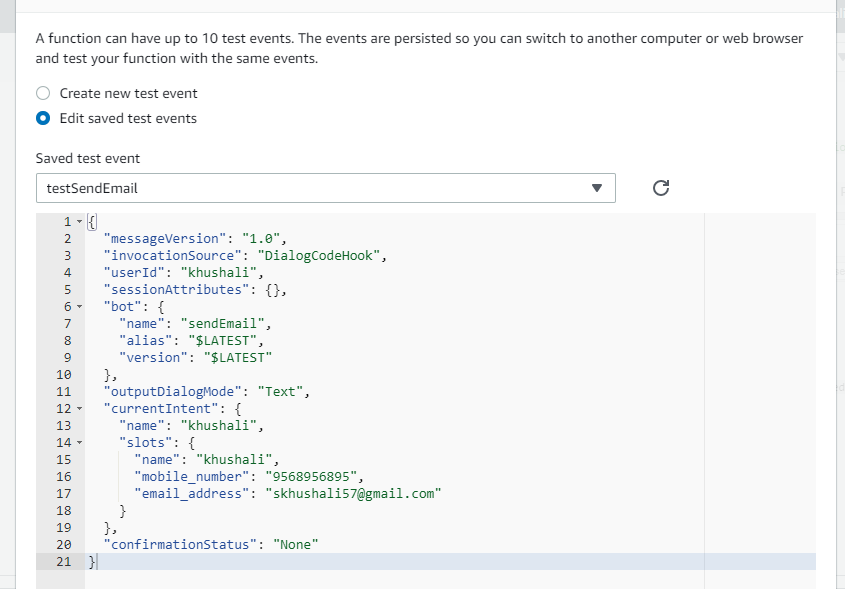
We have written a python code and deployed it as an “Api” which we are directly calling throught this “sendEmail” function.

We are keeping the Lambda function as simple as possible and we will not add any more functionality. Let’s test our function written in the lambda.

For running the test, firs save our lambda function, then click on the “configure test events”

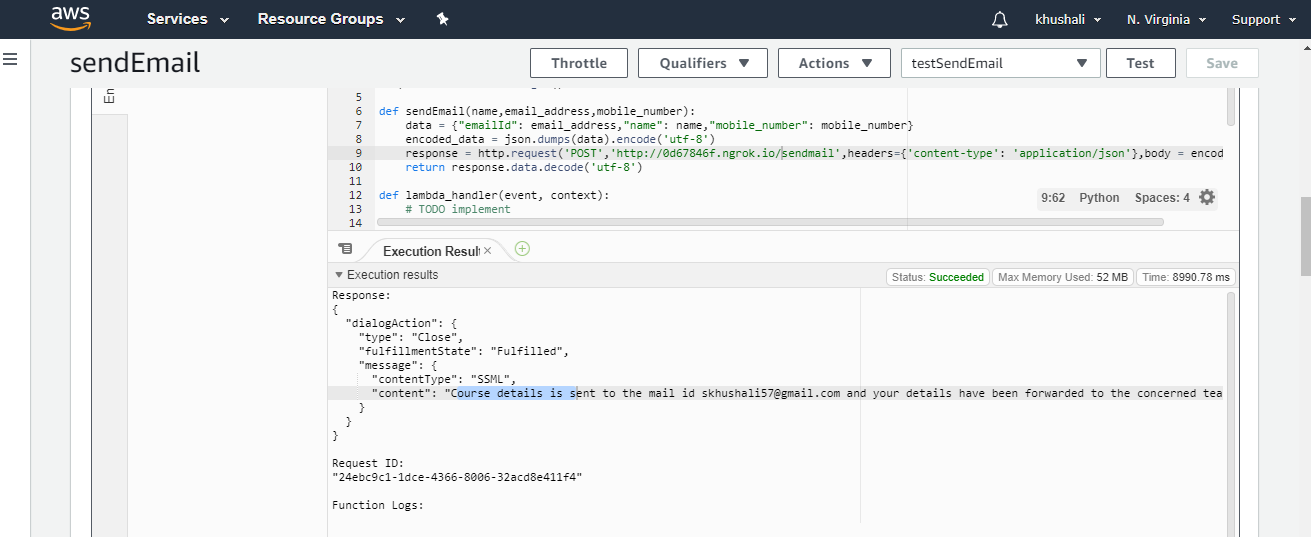


You will see fields like this, create a new test event and set default values for the slots for testing the lamda function:



Once the default values are set. Let’s test our bot. Click on the test bot.

You will see a success message if your test works like this:

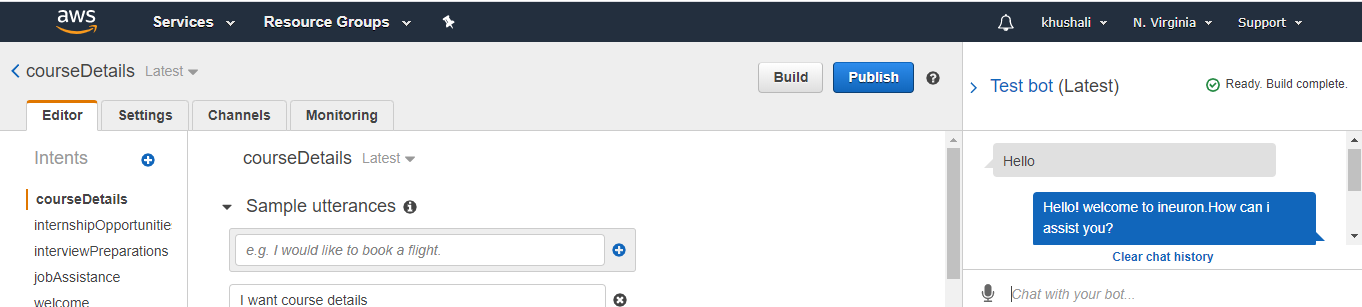


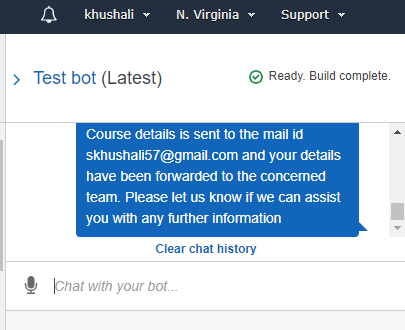
You will receive an error status if the function doesn’t work properly.

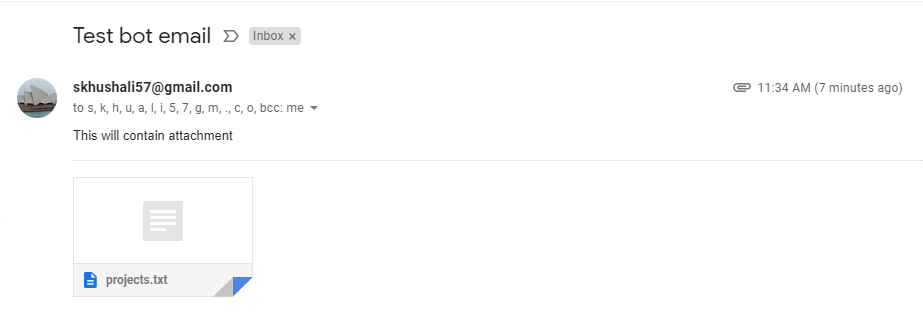
Now that we have created all of our intents. You can test your bot with the intents by “building” you bot.

Wait till you get the message that you bot is ready for testing.

Let’s test our bot in the “Test Bot” window on the right.

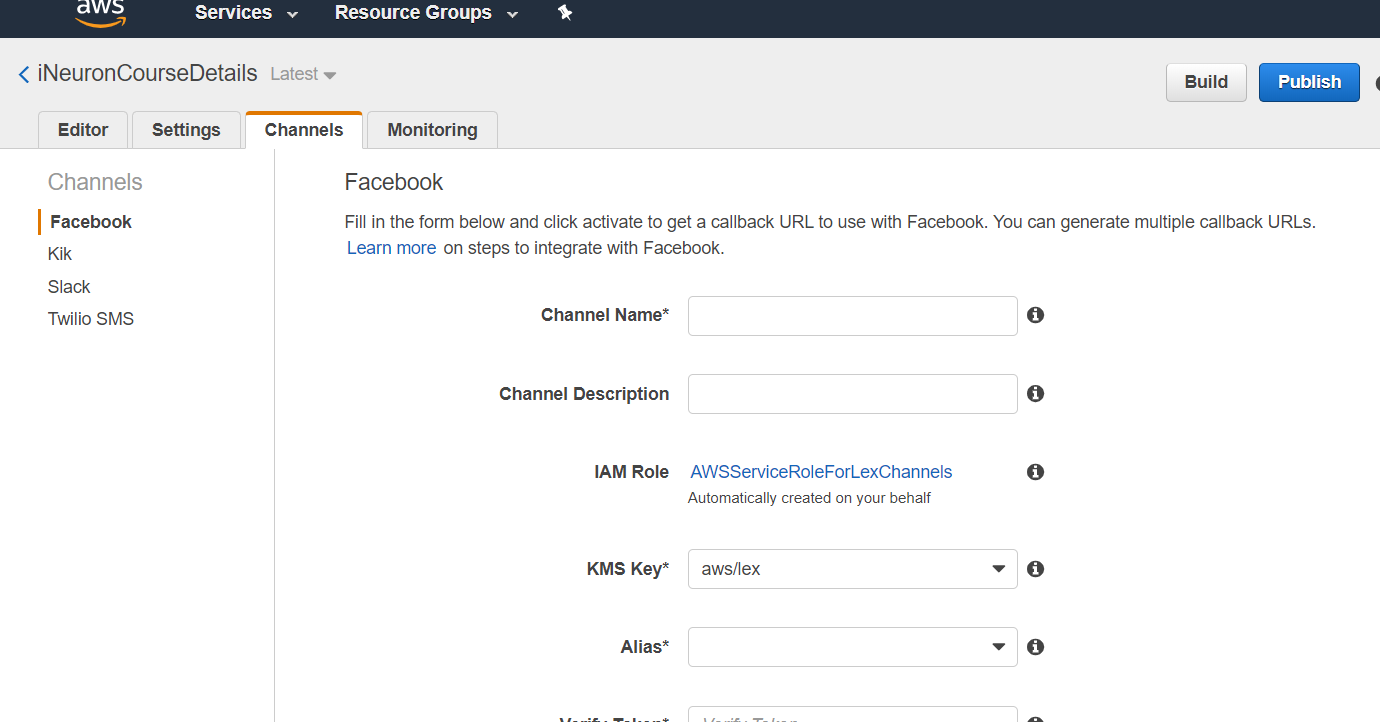


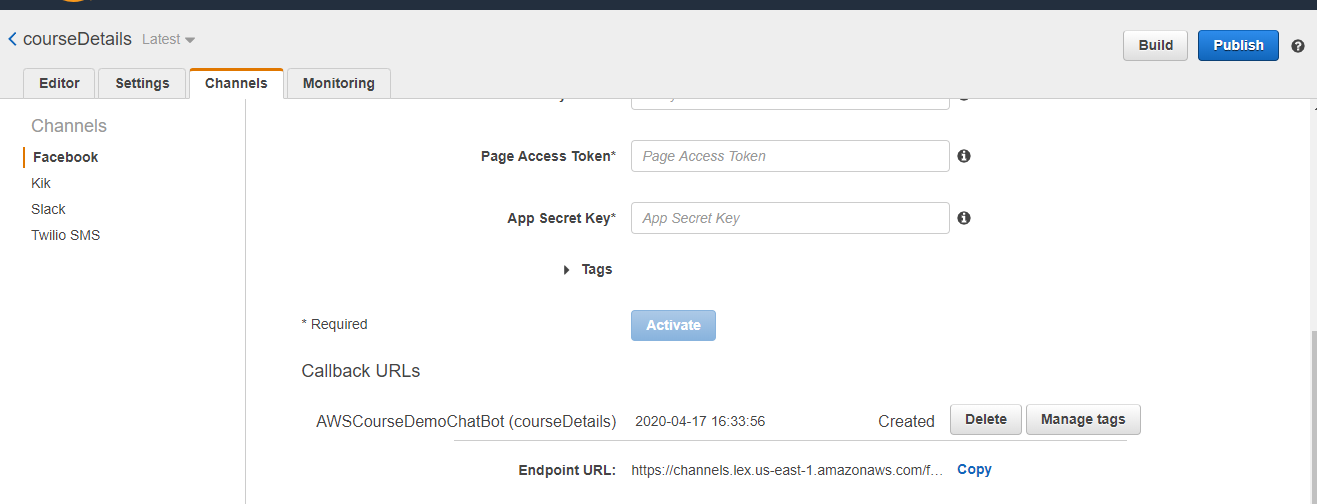




Now that we have built our bot and it’s working fine. Let’s go ahead with publishing the bot.

Before publishing the bot, go the settings tab and give an alias name to the bot. Again, build your bot.



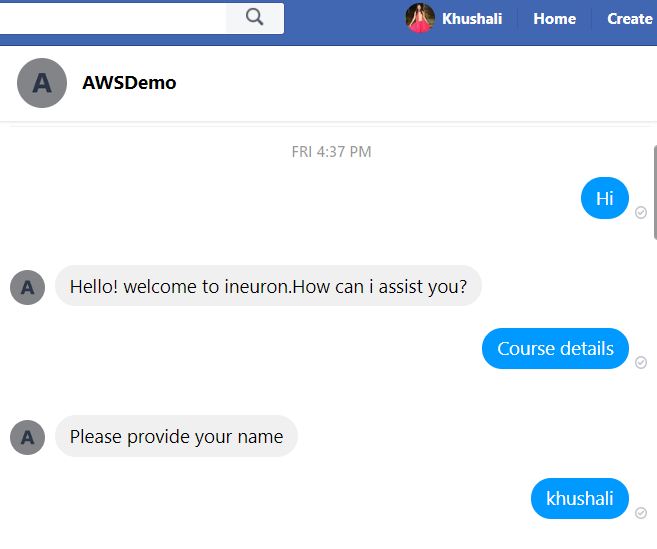


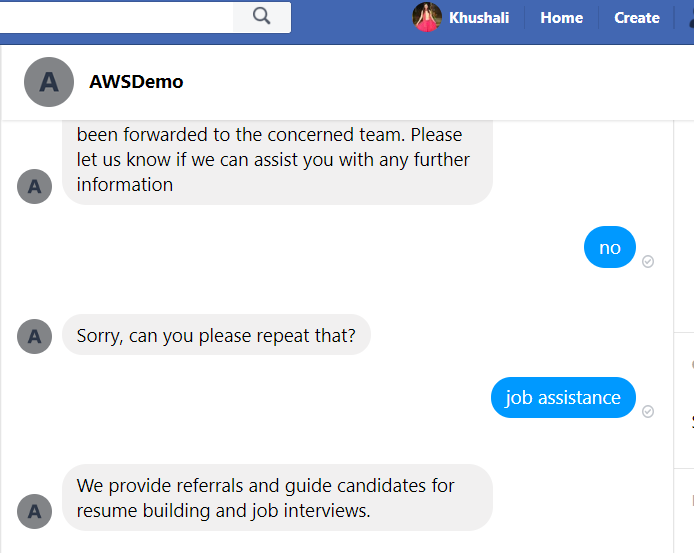
Click on the publish button and select the alias name that you have given.

Click on the publish button.

Now your bot is published. Let’s deploy our bot on Facebook. Clik on the “Go to Channels”.

Facebook Integration:-





**Note:- Here code has been deployed with ngrok server, so local app should be running to make chatbot working.**