Creating a Chatbot using Amazon Flex Service -Akhil Mittal

Table of Contents

Preface

This article is the submission against CodeProject's Slack API Challenge. The contest officially began on January 7, 2019, and ends on March 7, 2019.

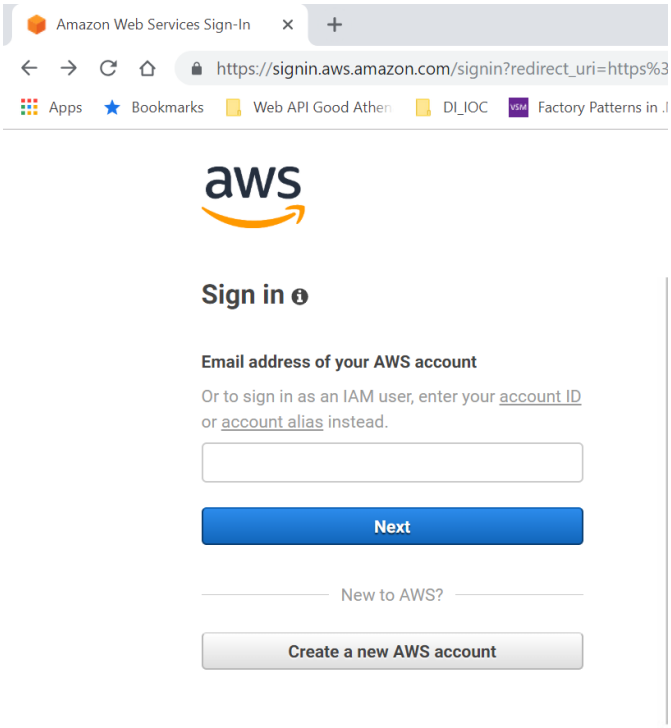
Introduction

Slack is no doubt the platform that every developer needs nowadays and is trending in a new way.
It is easy to use and manage, and not only this we can create our own slack app and add a lot of custom functionality to it. You can learn how to create your own slack app in this article. It is very easy and may hardly take 20 minutes to create your first app.

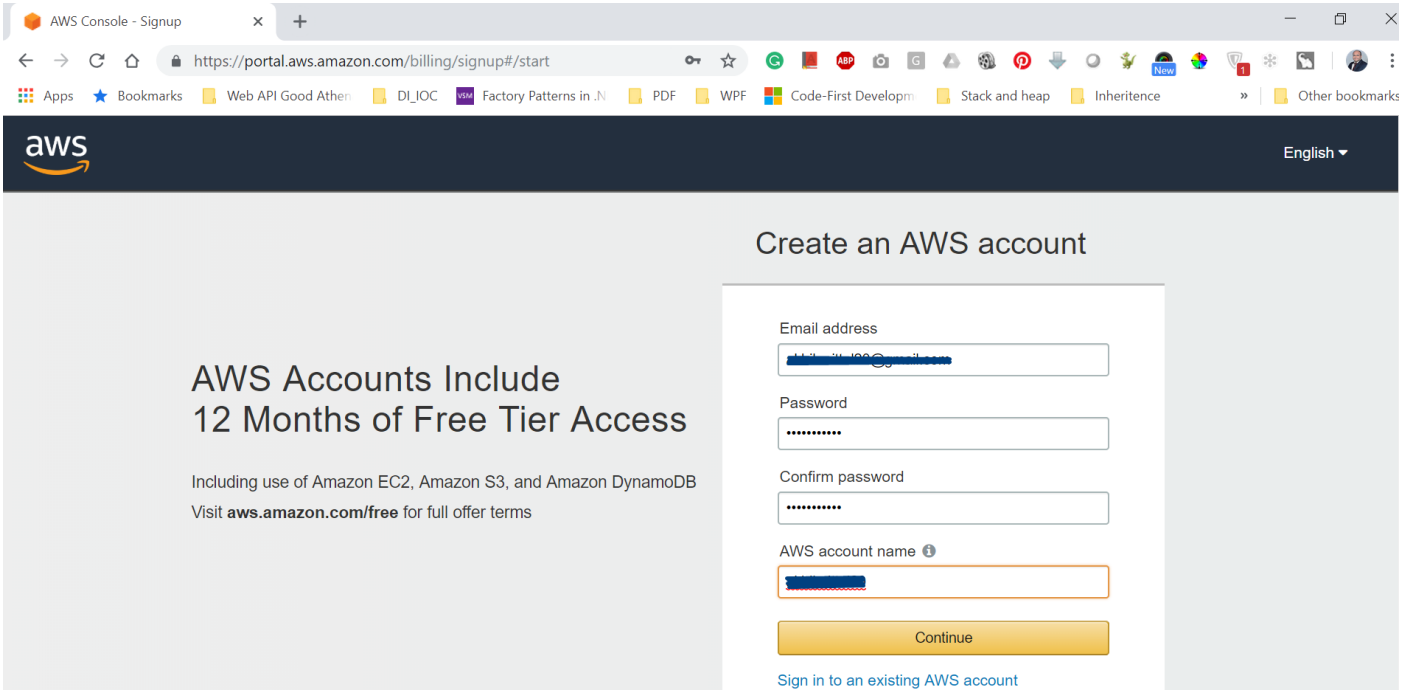
In this article, we'll discuss how to create a real-time chatbot using Amazon Flex service and deploy the bot over Facebook messenger. The reader of the article does not need to have prior knowledge of bot creation as the article will demonstrate the bot creation and deployment step by step using explanatory pictures. Trust me, it is fun and exciting to create your own bot and see it working over the thought of creating a bot. We'll setup AWS account to leverage Lex Services and Facebook developers account to deploy the bot on your own Facebook page and see it working. Let's jump directly into the tutorial.

Setup AWS account

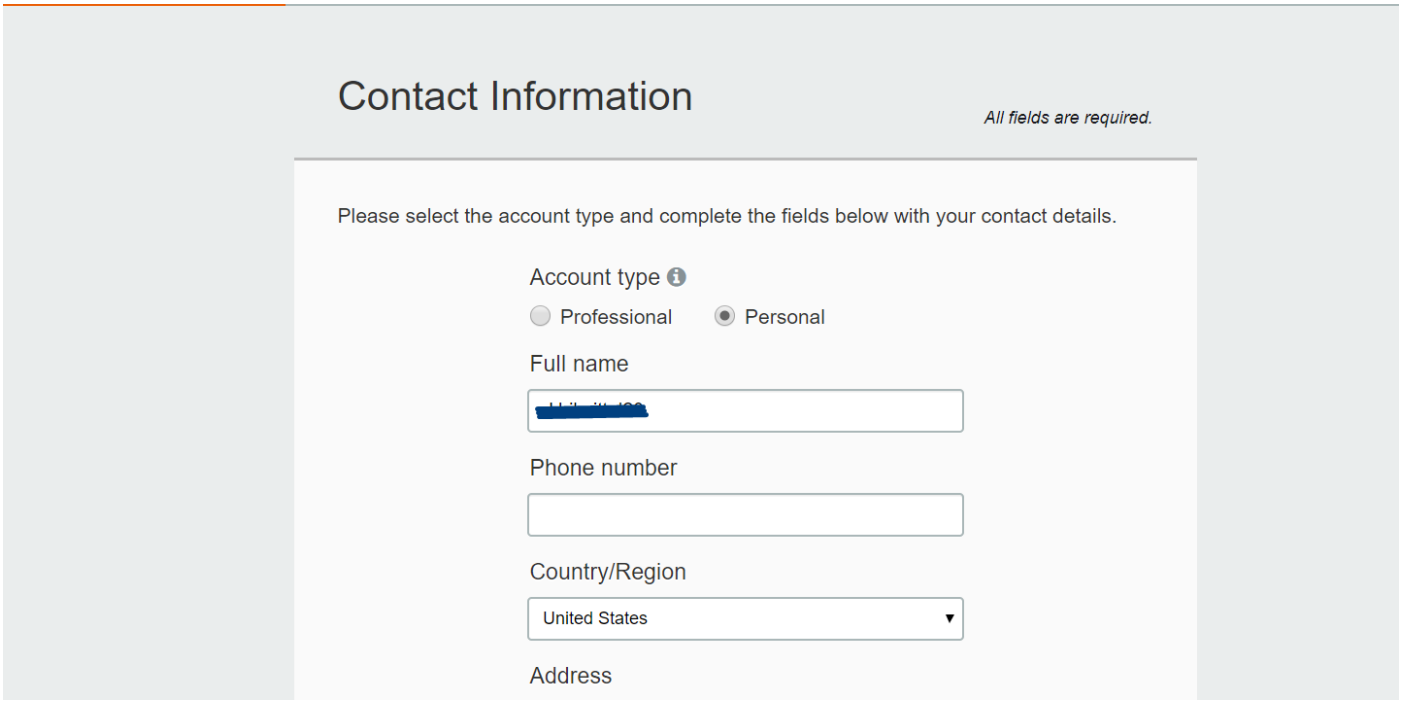
1. Go to aws.amazon.com and sign in using your email address if you have an account or create a new account.



2. Provide the relevant details like email address and password while creating a new account.

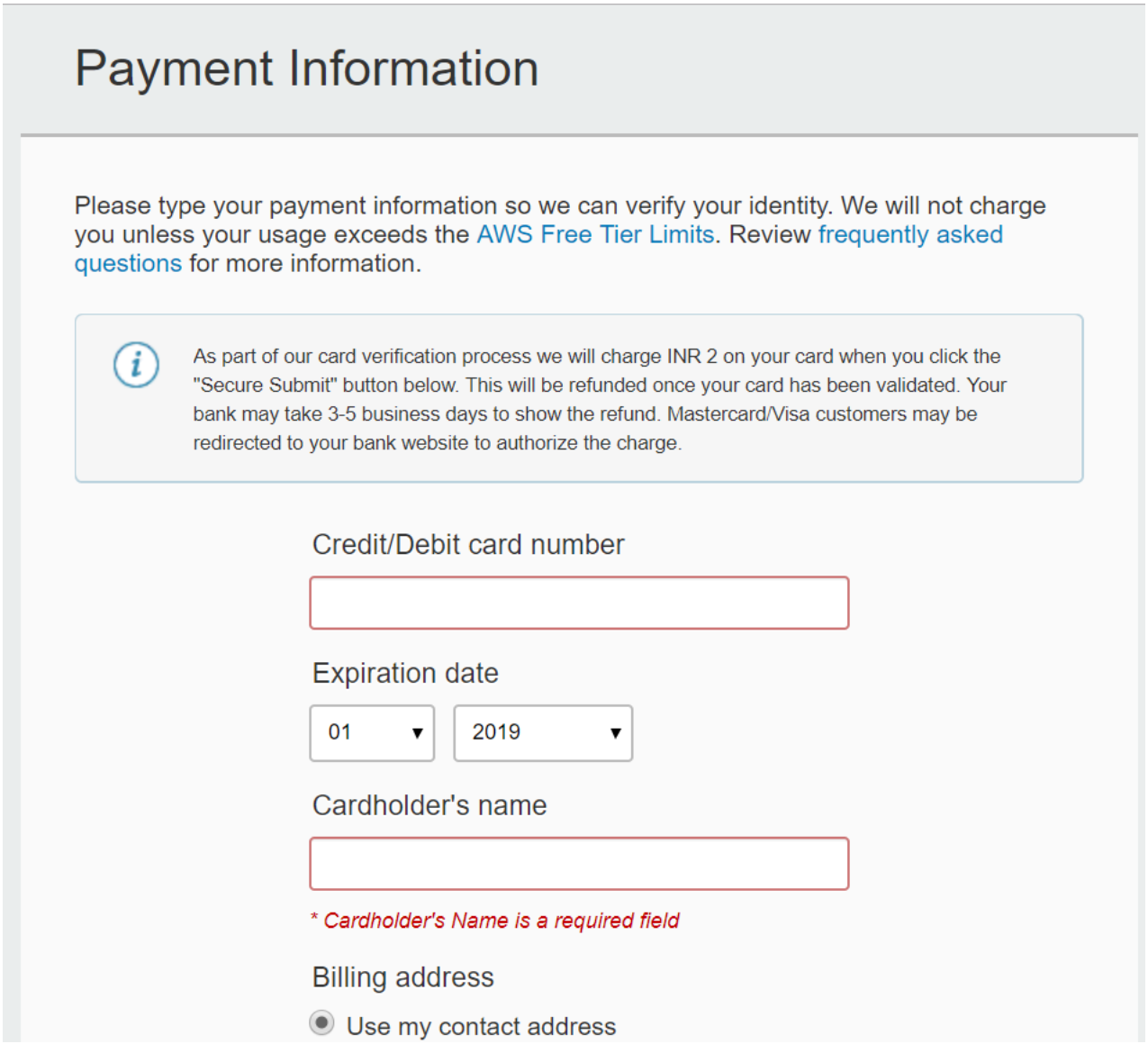


3. On the next step, you can choose to create a Personal type account and provide needed information there.

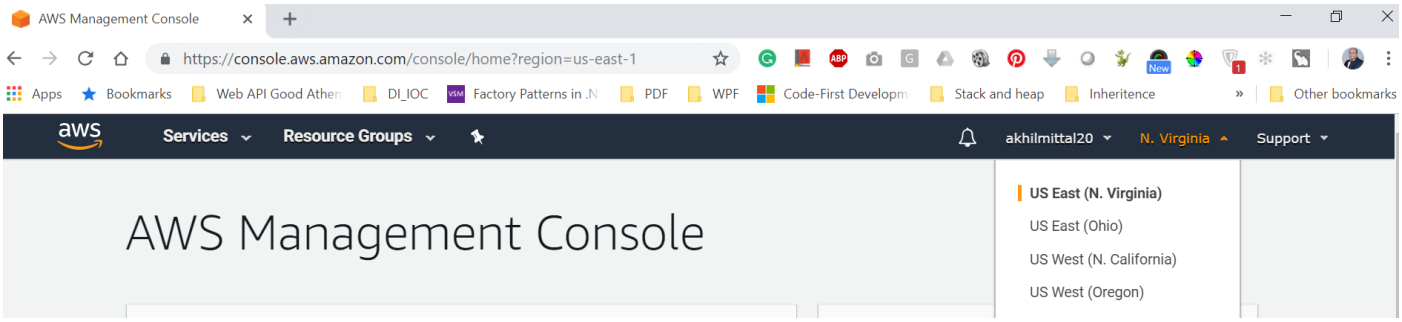


4. Next page is the payment information page where you need to provide the payment mode and its details. Nothing to worry about as when you enter your card details it deducts a minimal amount i.e. 2 INR which would be reverted to the account after successful verification of the card.
"With Amazon Lex, you pay only for what you use. You are charged based on the number of text or voice requests processed by your bot, at $0.004 per voice request, and $.00075 per text request. For example, the cost for 1,000 speech requests would be $4.00, and 1,000 text requests would cost $0.75. Your usage is measured in "requests processed", which are added up at the end of the month to generate your monthly charges.

   You can try Amazon Lex for free. From the date you get started with Amazon Lex, you can process up to 10,000 text requests and 5,000 speech requests per month for free for the first year." Read more about Lex pricing here.



5. Once successfully entered the console. Choose the region US East (N. Virginia).

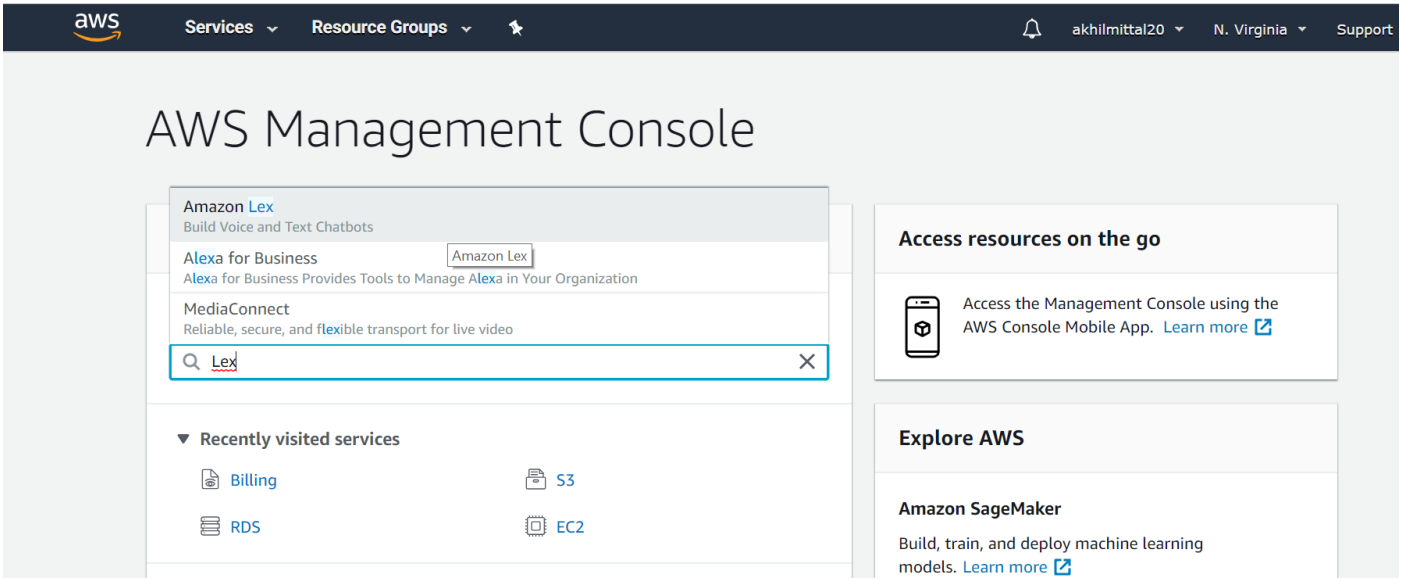


Create Bot using Amazon Lex

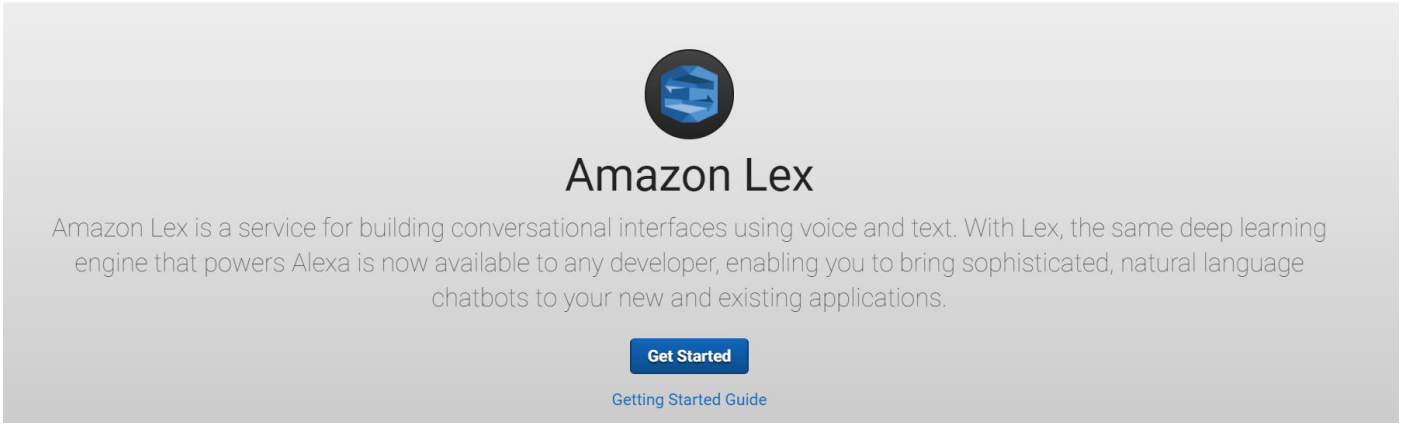
After setting up the AWS account, it is time to create the bot using Lex service. Let's get familiar with the terminology before we start.
A bot is a top-level component that that is deployed. It is a kind of container. The bot that we'll develop will have one or more intents. Each one of it represents a different topic that a bot can understand and perform an action to fulfill it.
A Slot is something that we define to get the kind of information we need from the bot user to fulfill the intent.

A Lambda function is simply a place to execute your code that doesn't require you to provision or manage a server, or serverless. Lambda supports multiple programming languages and is how you integrate your bot with other systems. Your AWS account includes 1 million free requests per month and up to 3. 2 million seconds of total execution time. Please visit the AWS Lambda pricing page for more details.

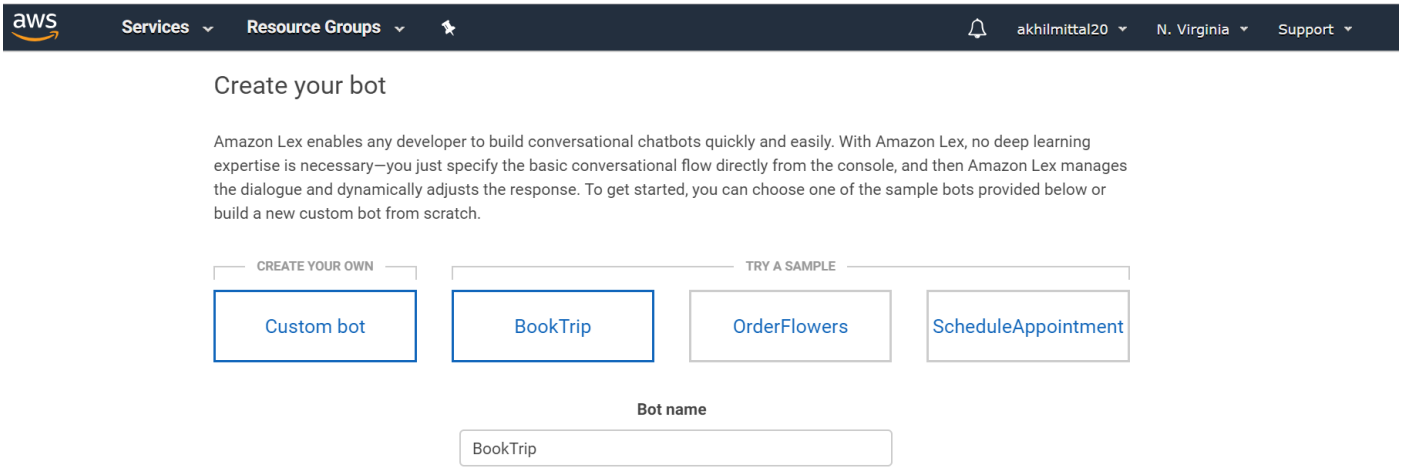
1. In the console, search for the service type Lex as shown in the following image.



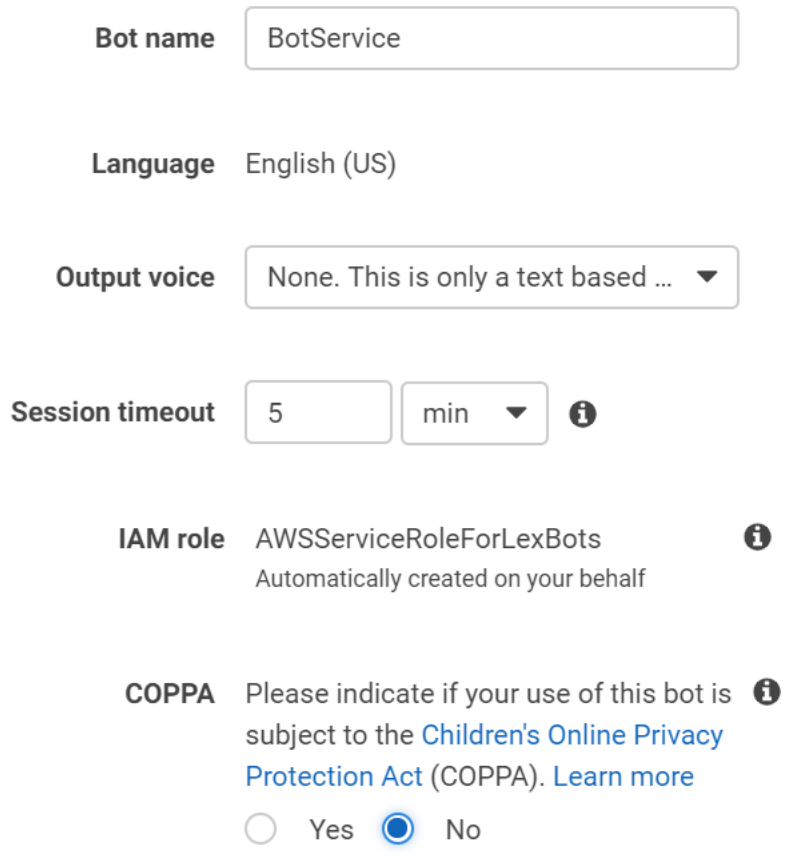
2. You'll be shown the Amazon Lex landing page. Click on Get Started to start creating the bot.



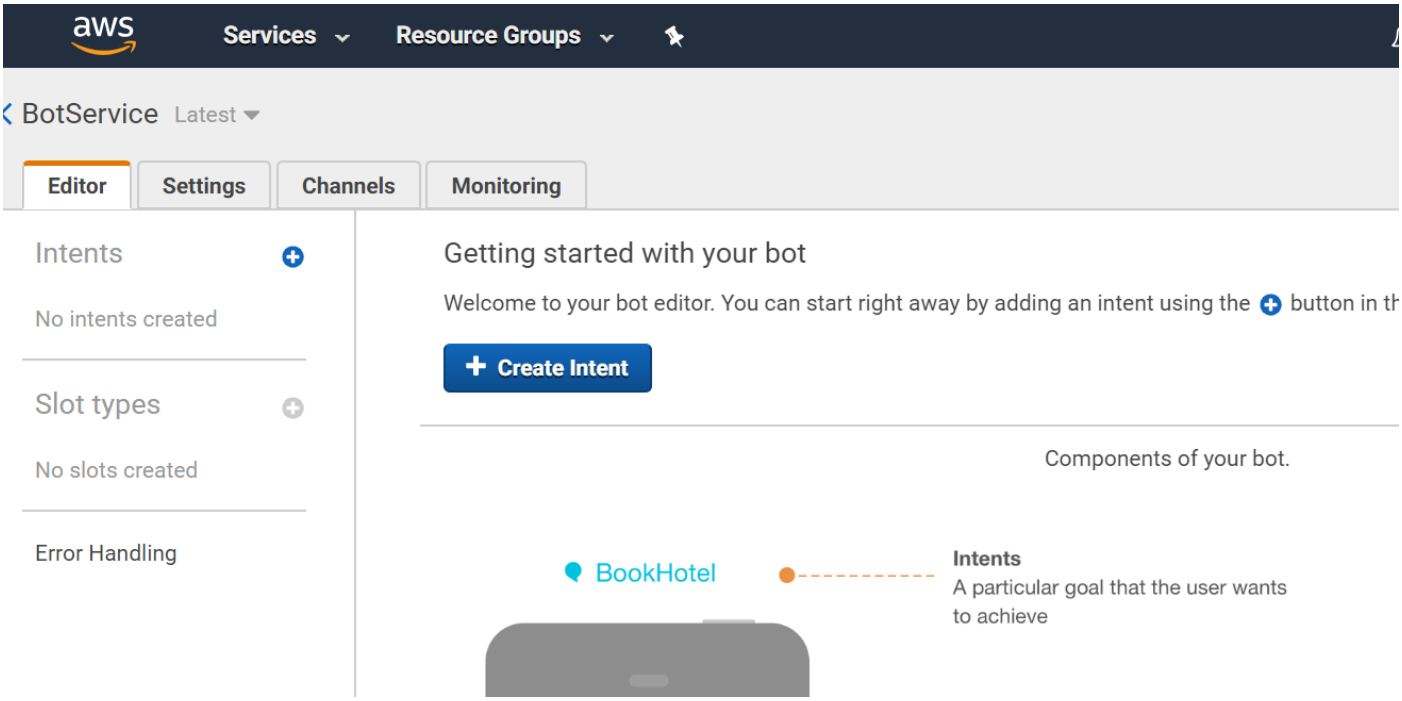
3. We can try the existing bot samples and can create our own as well. Since we are creating a bot from scratch, choose the option of the Custom bot and give it a name.



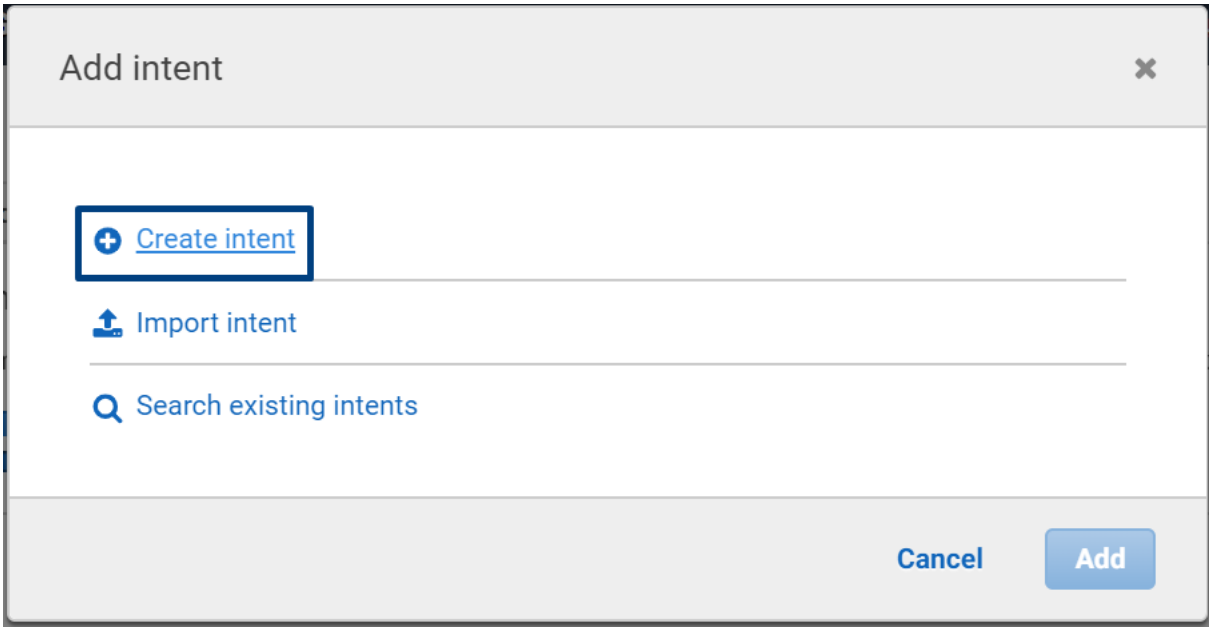
4. Provide details like bot name. For output voice, choose None as this will only be a text-based bot. Set the timeout to 5 minutes. The IAM role is created for us, so there is no action needed on this line. For COPPA we can choose No.



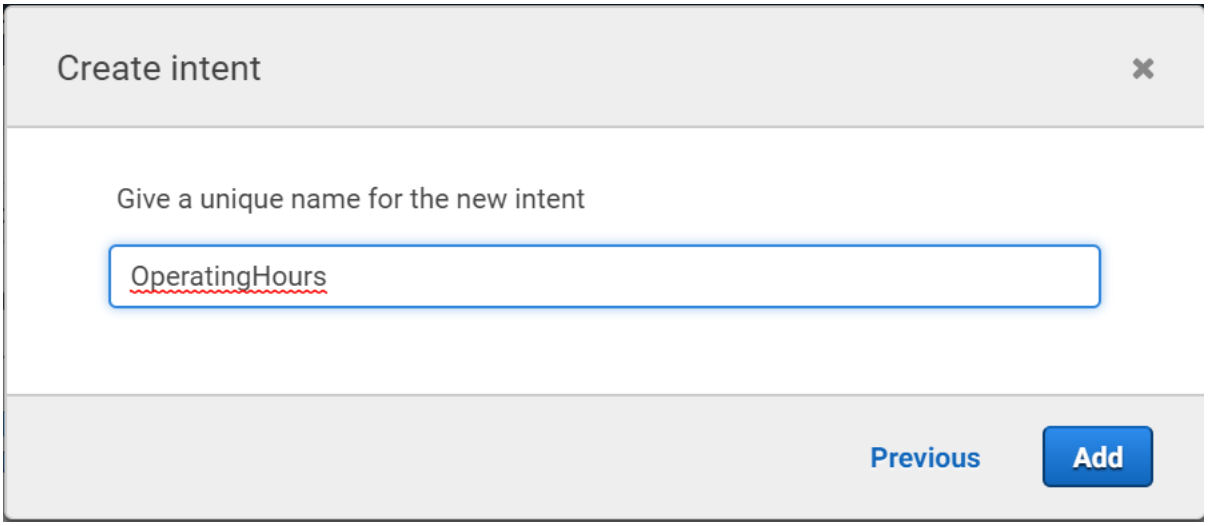
5. Once done, you'll see the BotService page. Now it's time to create the intent as the bot is only just a container and it needs intents. Click on create an intent button to create a new intent.



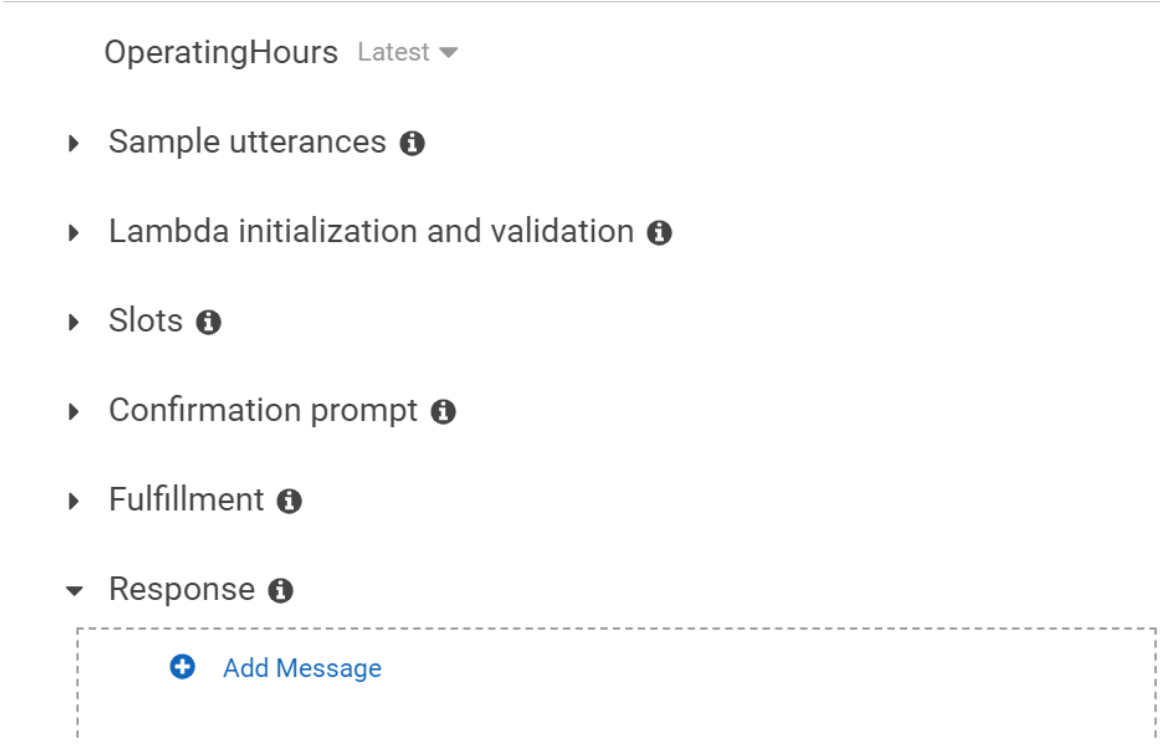
6. On the popup, it shows the option to search for existing intents, create new or import an intent. We'll select Create intent option to create a new intent.



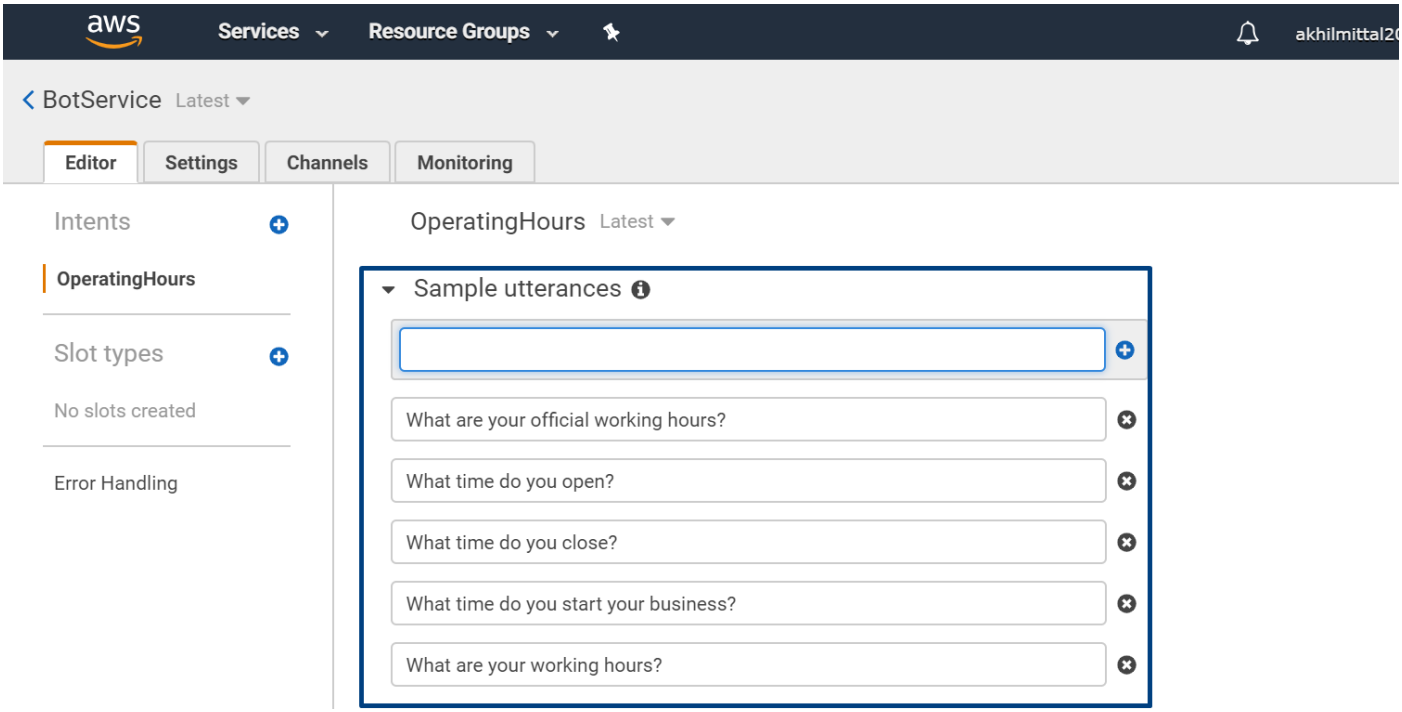
7. Give a unique name to the intent. In our case, it is "OperatingHours". The purpose of this intent to tell the user about the operating business hours of the company for which the user is seeking information. For e.g. when you reach out to a company's contact us via chat section, the bot welcomes you and if the user asks, "what your working hours are?" then the bot should respond with the company's working hours and tell that to the user.



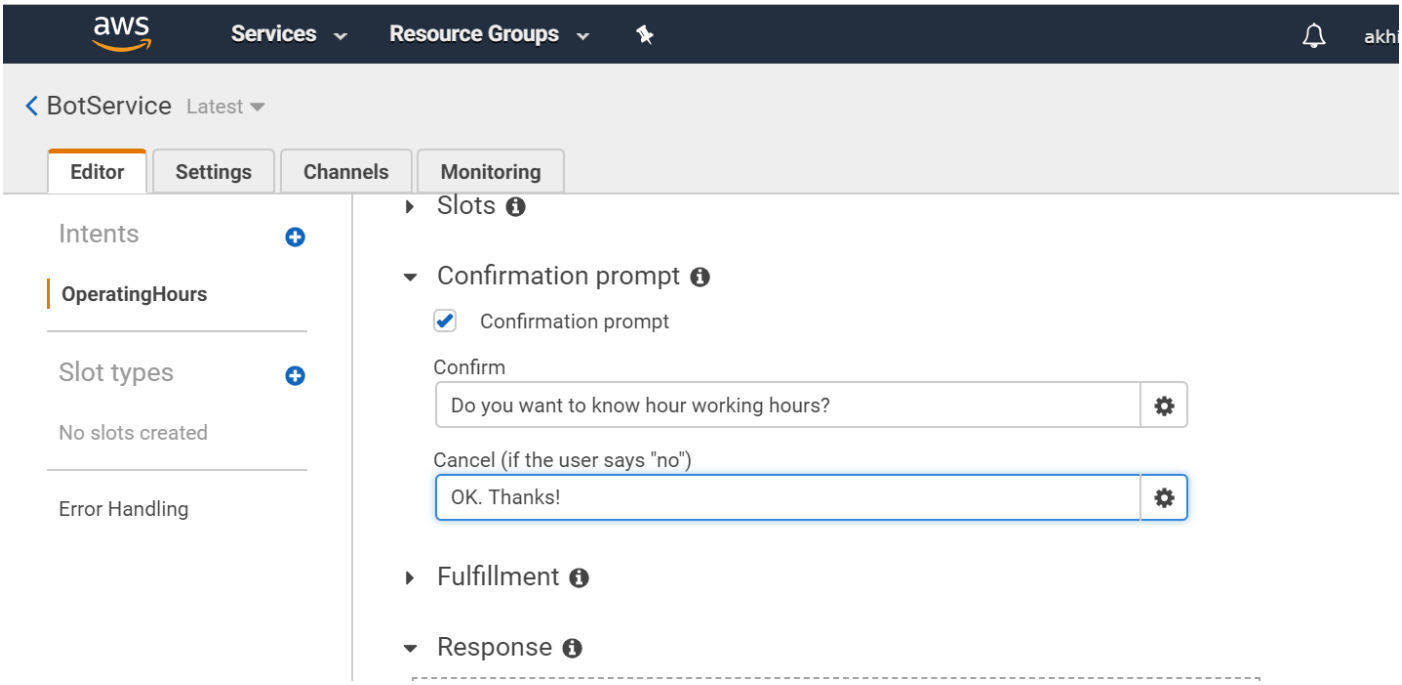
8. Adding intent needs some more actions to be performed. For e.g. Sample utterances i.e. the kind of questions a user can ask to bot for e.g. “what are your working hours?”, “what time do you open?”, “what time do you close?” etc. Note that we do not have to provide all possible examples. The built-in language model will recognize similar phrases. Next is Lambda initialization and validation. This is where we can link to the code in an AWS Lambda function to validate the user inputs if we are collecting information as part of the intent. Slots are how we collect data from the user to pass into validation or fulfillment. For this intent, we don't need any specific data from the user, so we will not define any Slots.



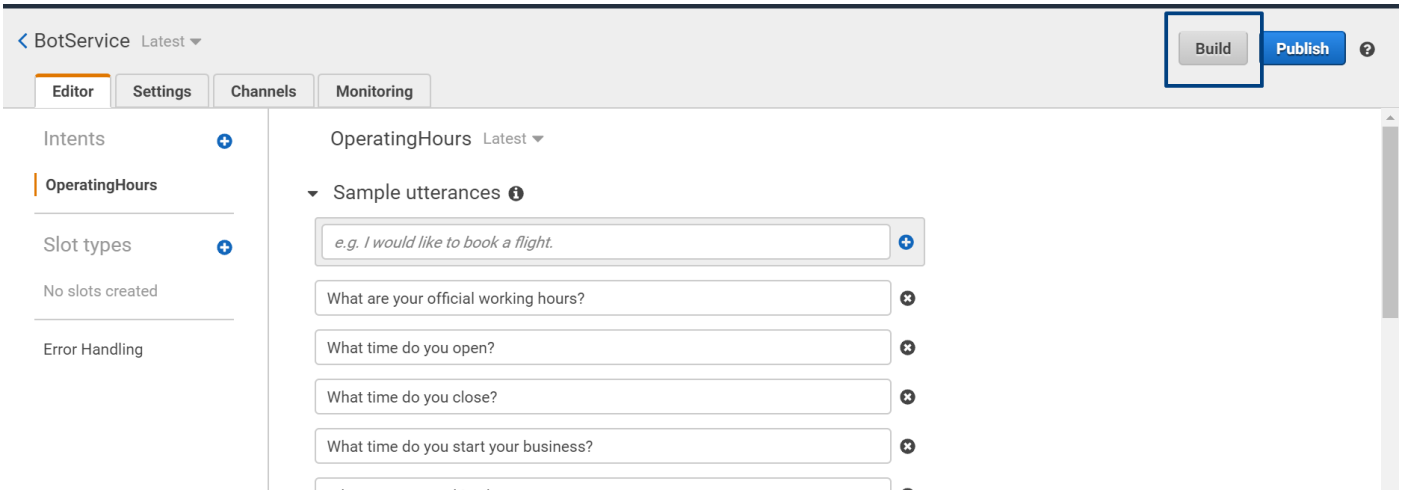
9. Provide some sample utterances as shown in the following picture.



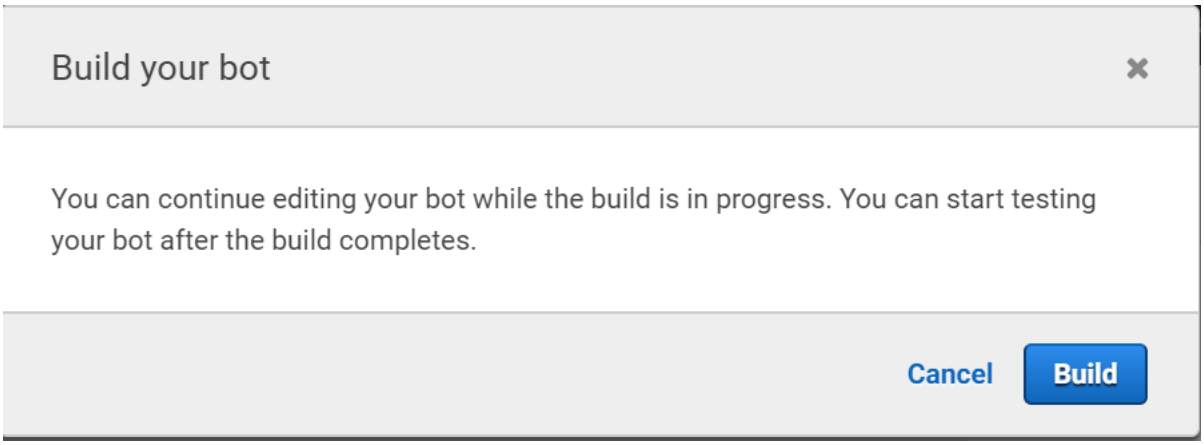
10. A confirmation prompt is for double checking with the user that we have everything understood correct from the user before the bit fulfills the intent. So, enable Confirmation prompt to see how it works. Since we didn't collect any data from the user, ask a simple question. If the user says yes, they move onto fulfillment. If they say no, then the message we put in this Cancel box will display, and the user will exit the intent.



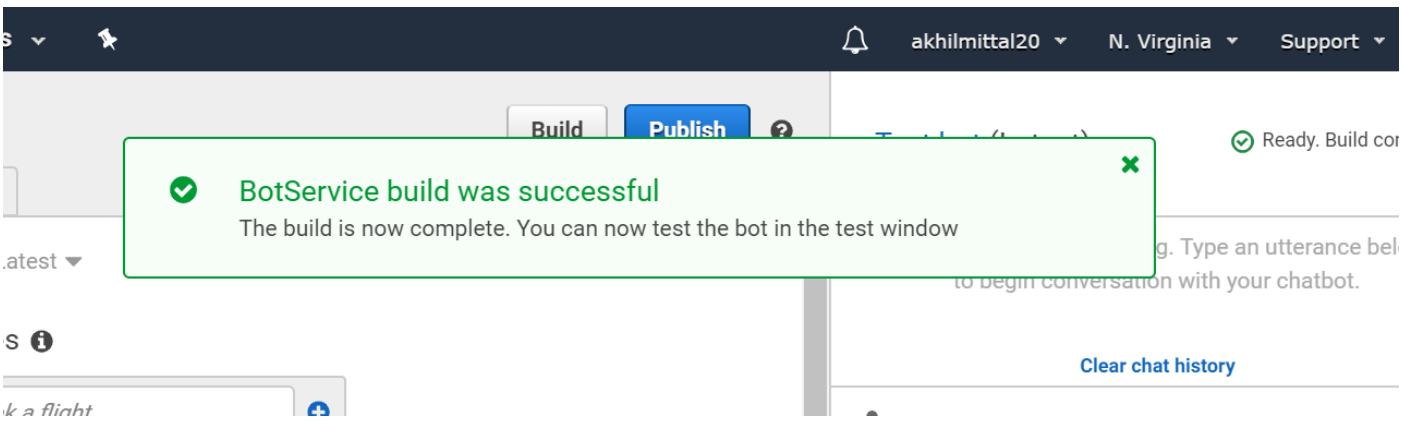
11. Finally, fulfillment is how the bot performs the result action. Typically, this is done by calling an AWS Lambda function to execute the appropriate business logic. There is also an option to return the parameters to the client. For now, you can choose that option as there is no lambda function for our bot.
12. Now, build the bot via Build the bot via Build option as shown in the following image.



13. Once you click built, you'll be prompted with a message to continue testing your bot while it is building. Click on Build.



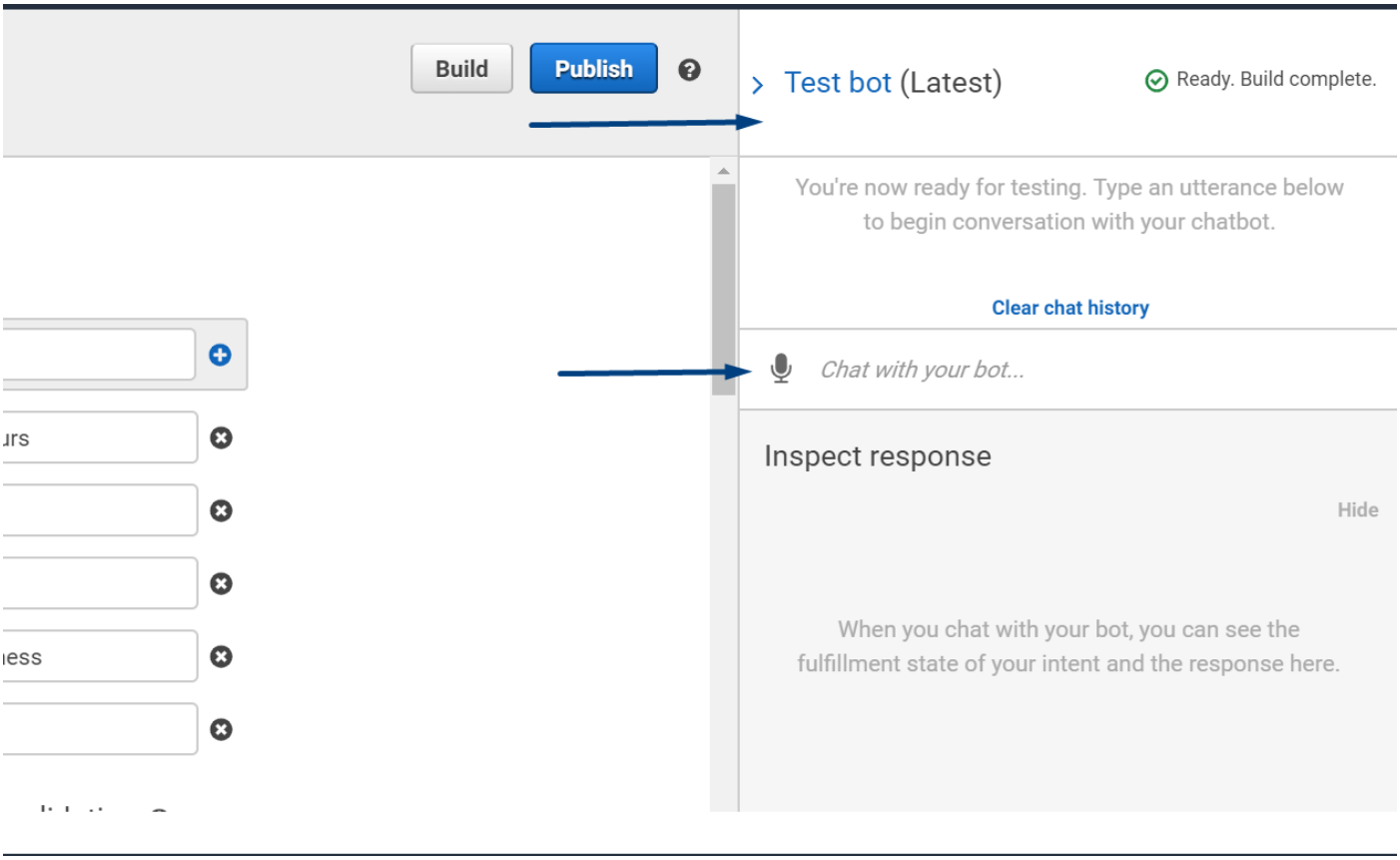
14. After a successful build, the prompt will be shown as shown in the following picture.



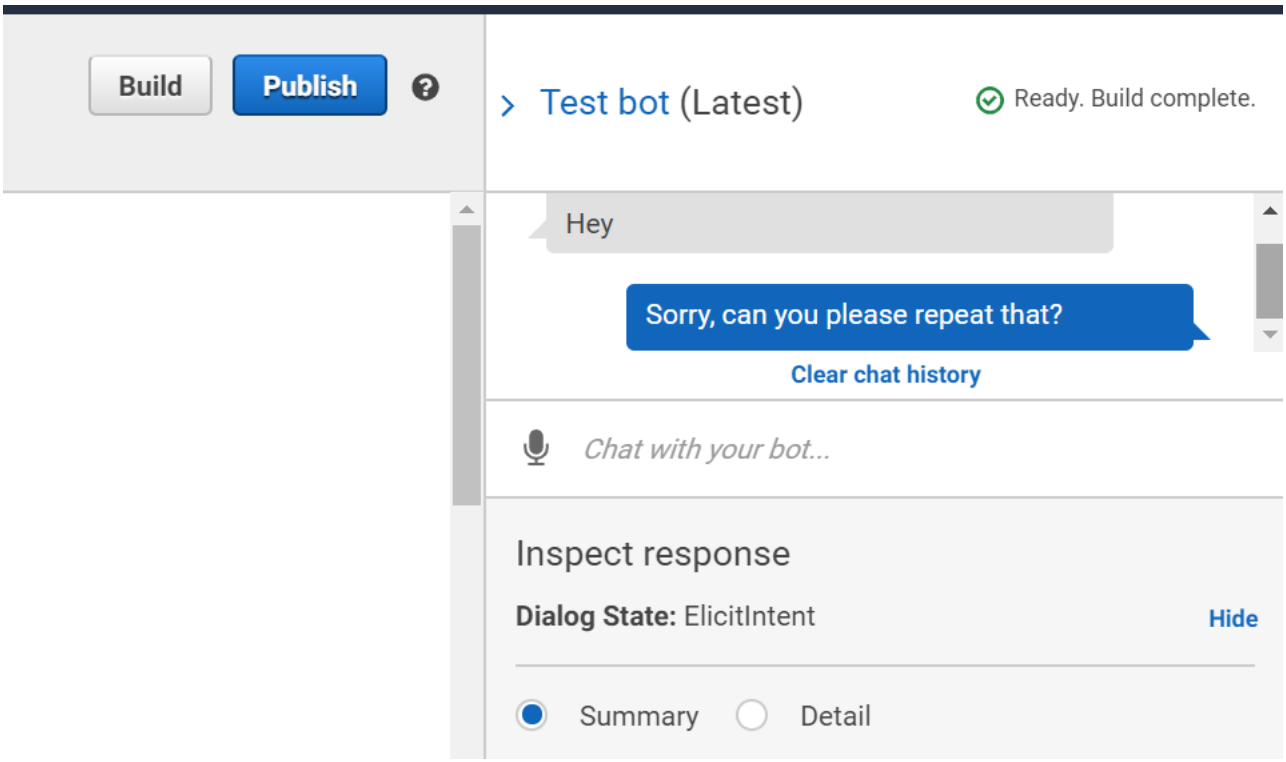
Test the Bot

Our first version of Bot is created now, and we can test it.

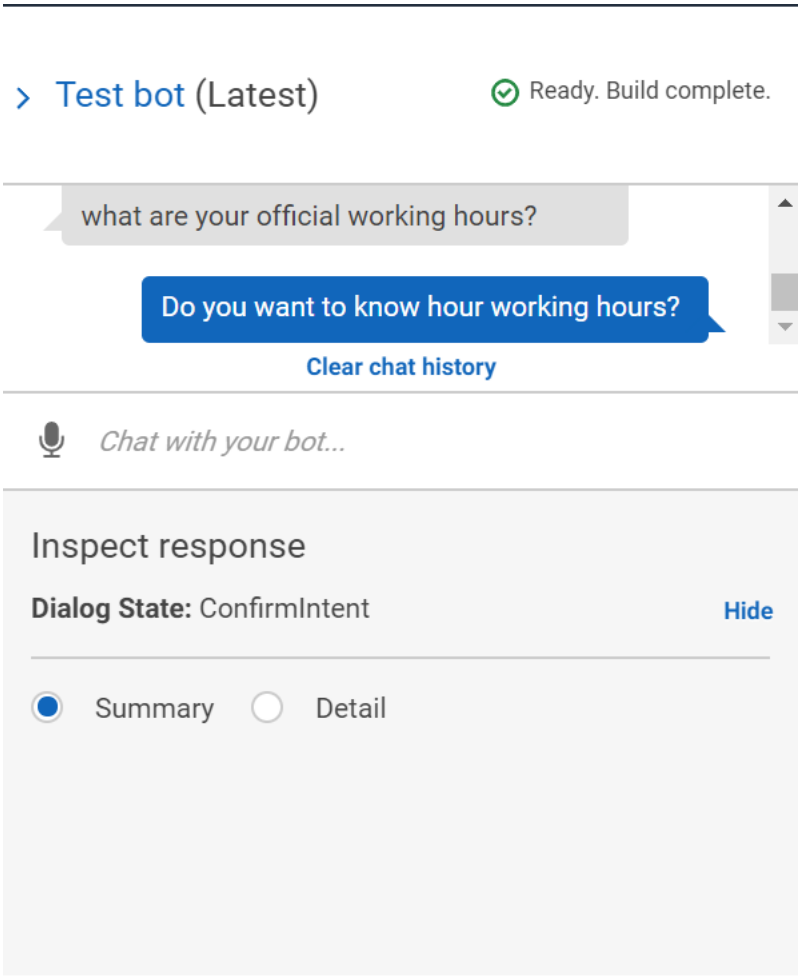
1. At the right side of the page, you can see the option to test the latest bot. Start typing in now 😊.



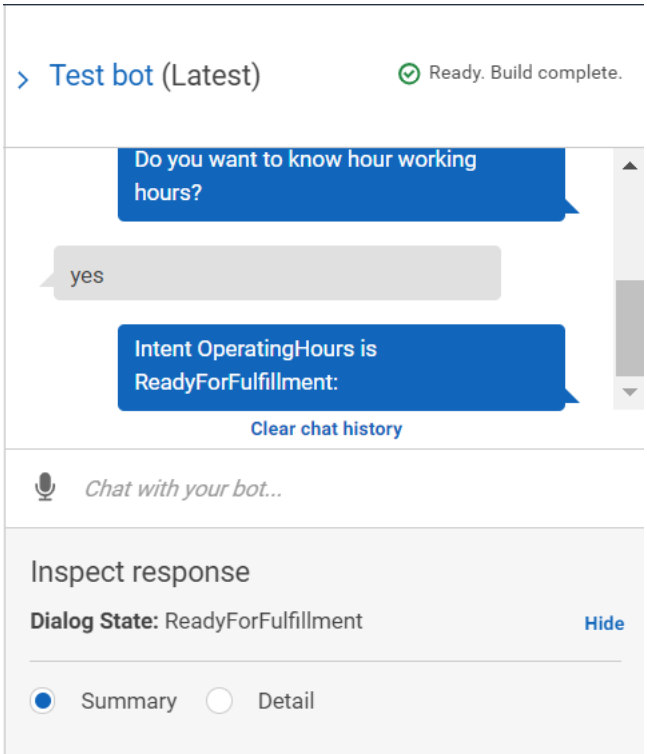
2. If you type anything other than the utterances defined in the intent, the bit doesn't recognizes that and gives a defined response as shown below.



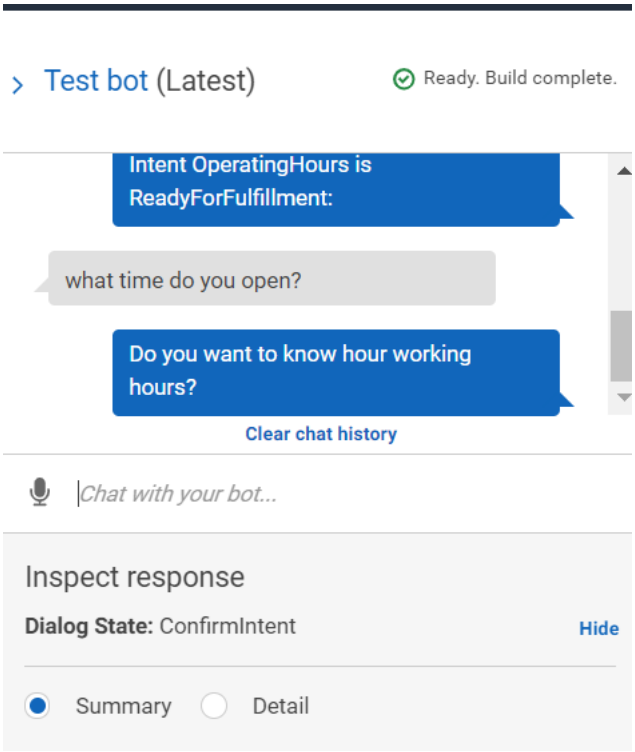
3. Ask your bot a question defined in utterances, for e.g. "what are your official working hours?" and it replies with the confirmation message that we provided in the intent.



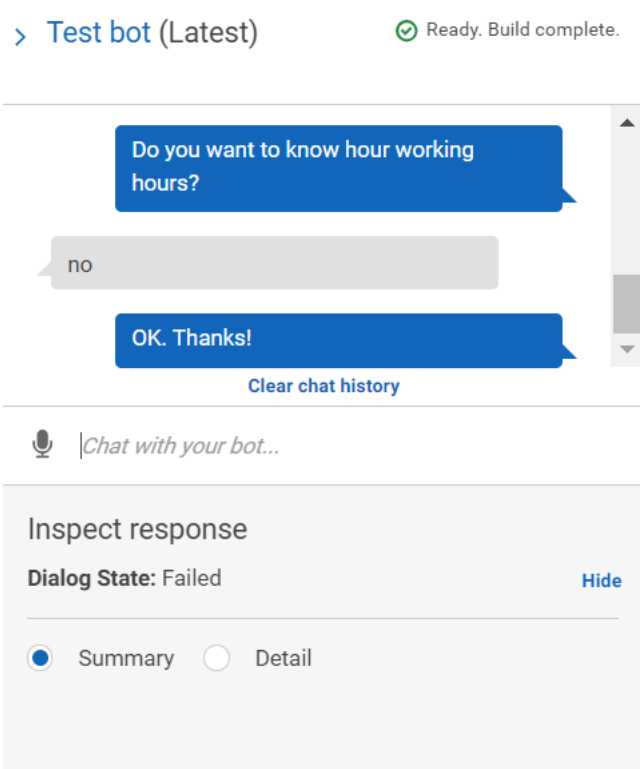
4. Once you say, yes, it navigates to fulfilling the intent, but since we did not provide any lambda it says it is ready to fulfillment.



5. Again, ask for the relevant question.

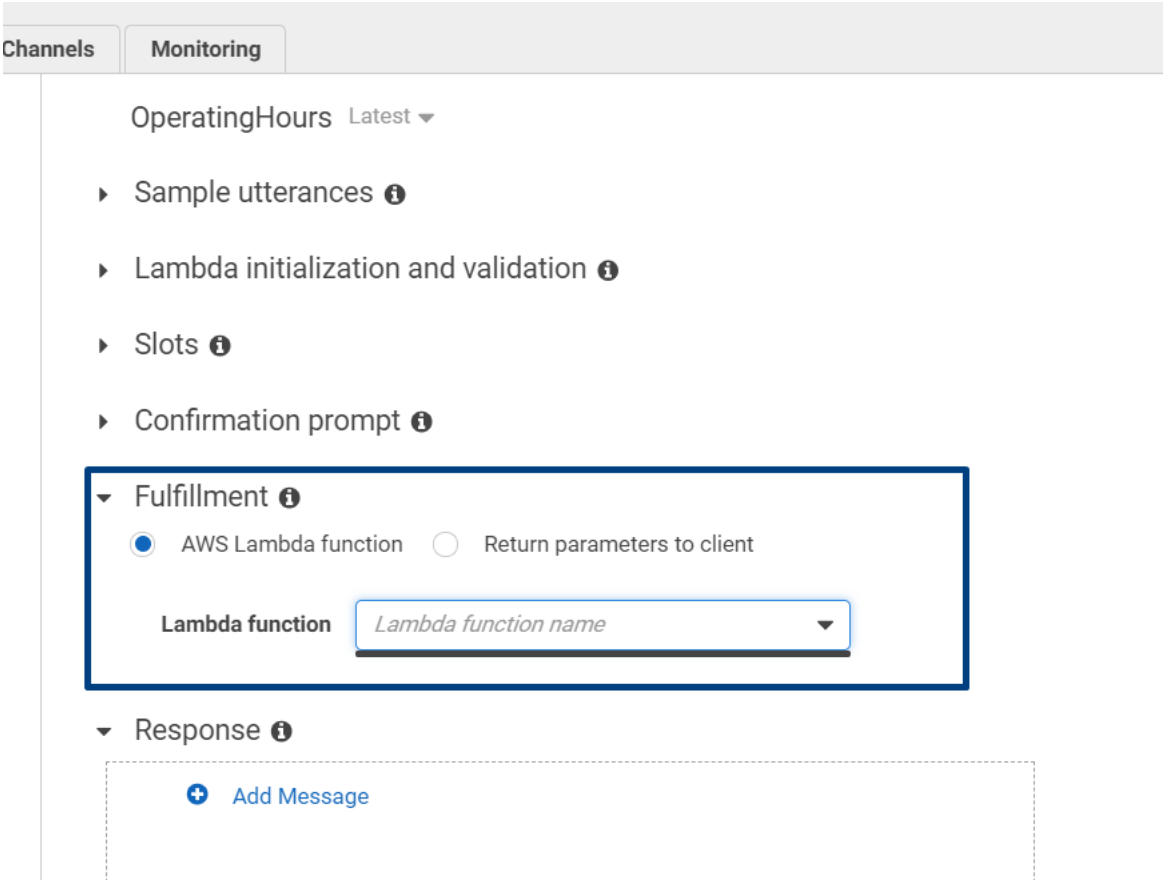


6. Now type no and we'll see that the bot responds with OK, Thanks! i.e. the text we provided in the confirmation for no.

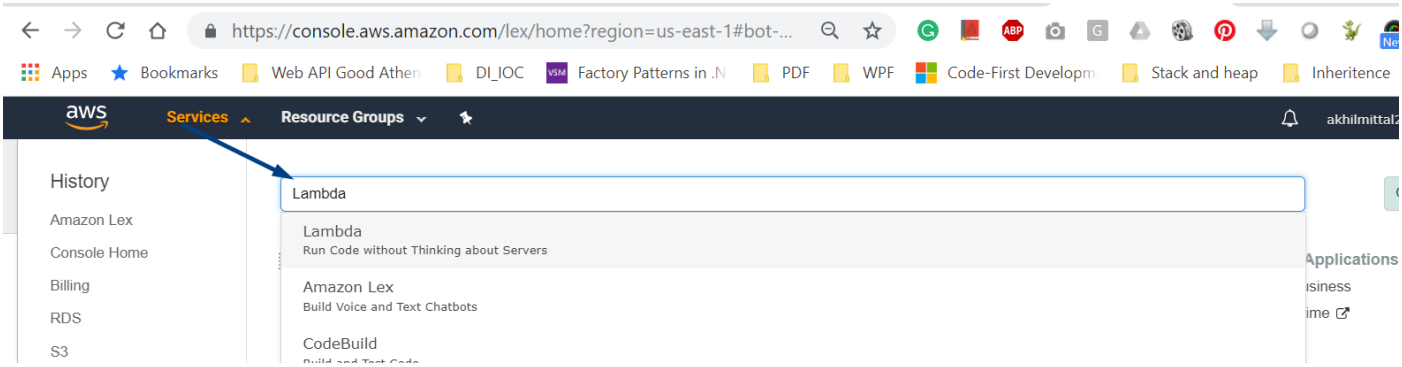


AWS Lambda Function

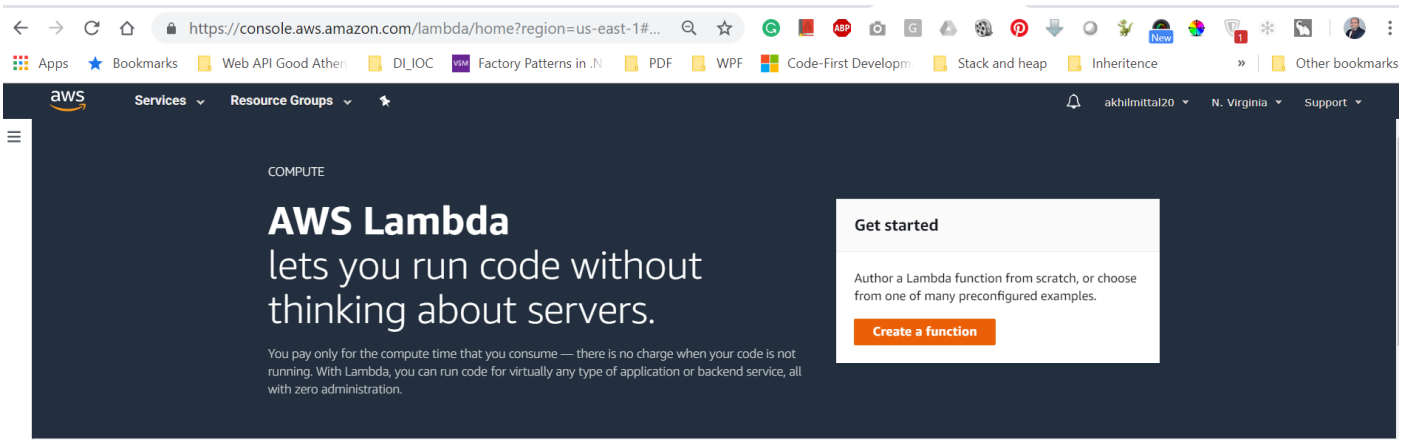
Let's enable our bot to use a lambda function i.e the defined business logic that a bot executes for fulfillment i.e. for the user when he types "yes". In the Fulfillment section of the intent, choose AWS Lambda function. This will ask you to provide a lambda function, but since we do not have one, we need to create it before we use it.



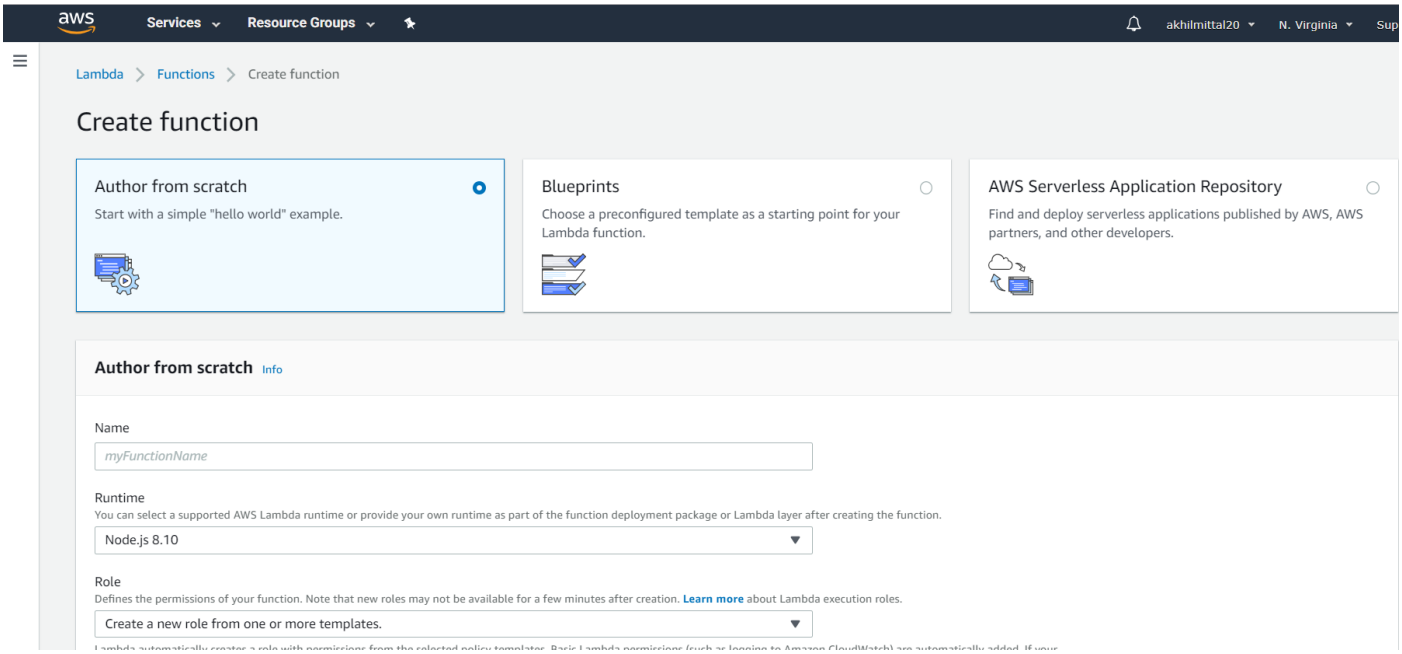
1. Go to Services search option and type Lambda. You'll get the suggestion options, choose the Lambda as shown.



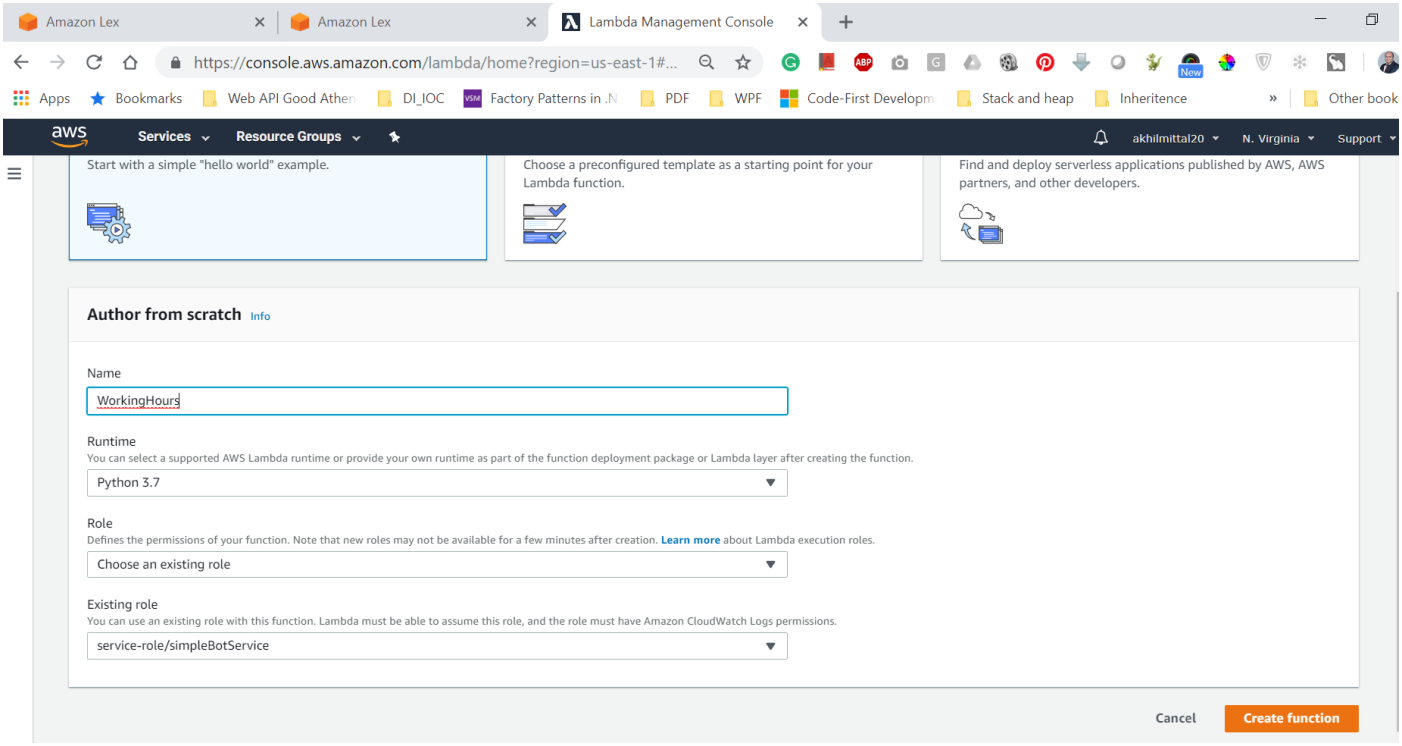
2. You'll be landed to AWS Lambda page. Click on Create a function to create a new lambda function for fulfillment.



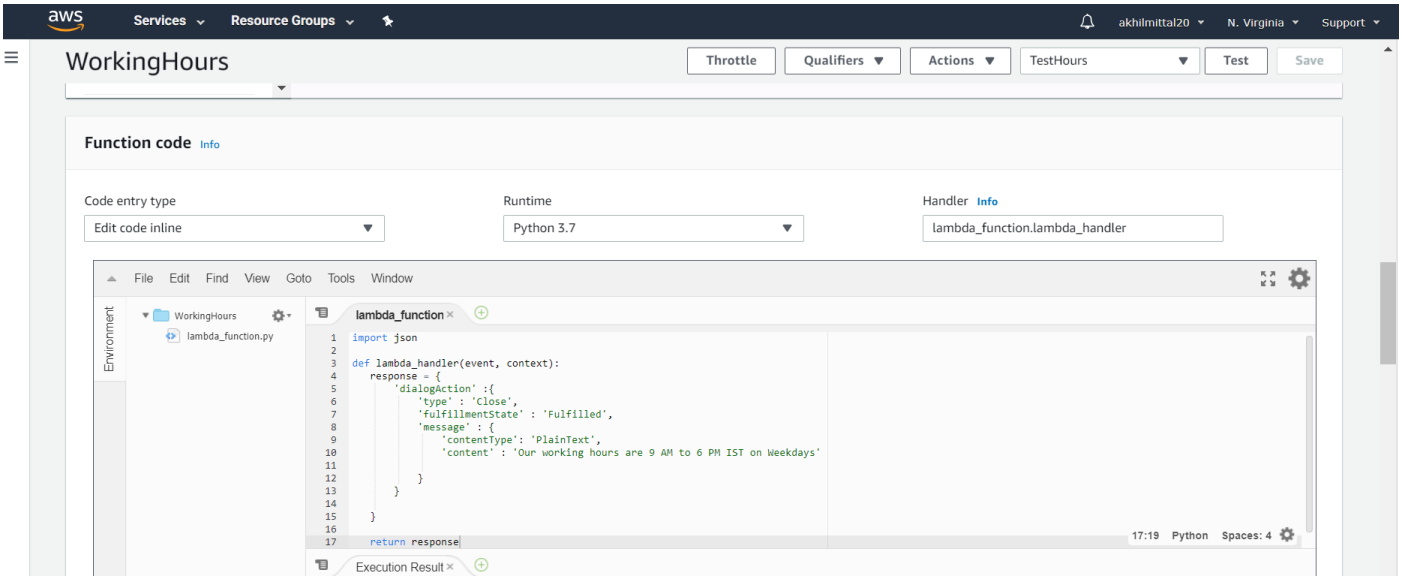
3. Select the option to author from scratch as shown below.



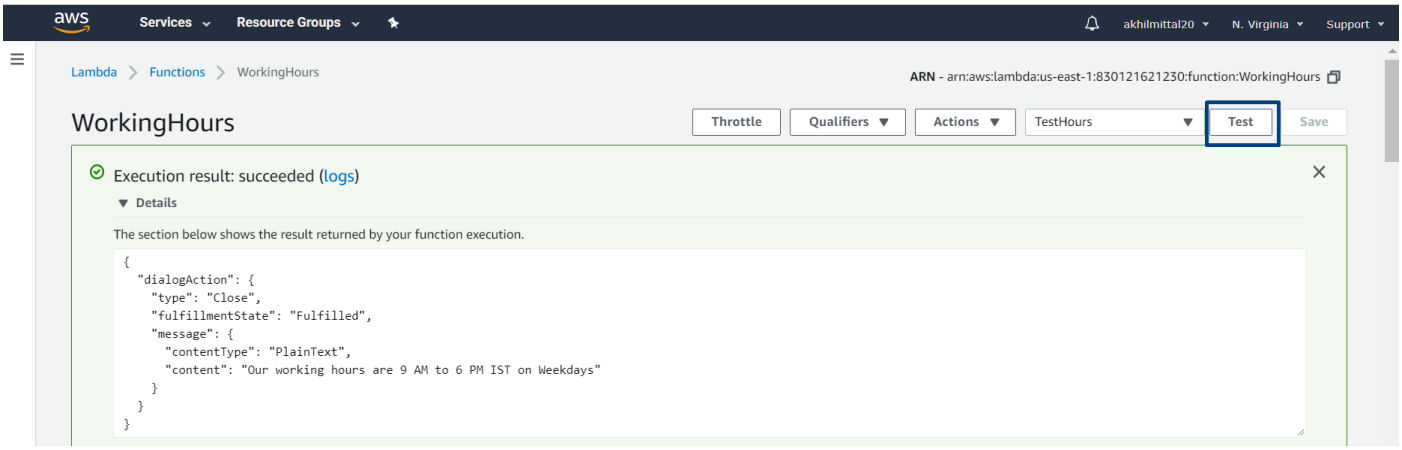
4. Provide the name to the function. You can choose any runtime you are comfortable with to write the code. I am choosing here Python 3.7 to write simple python code and return the result.



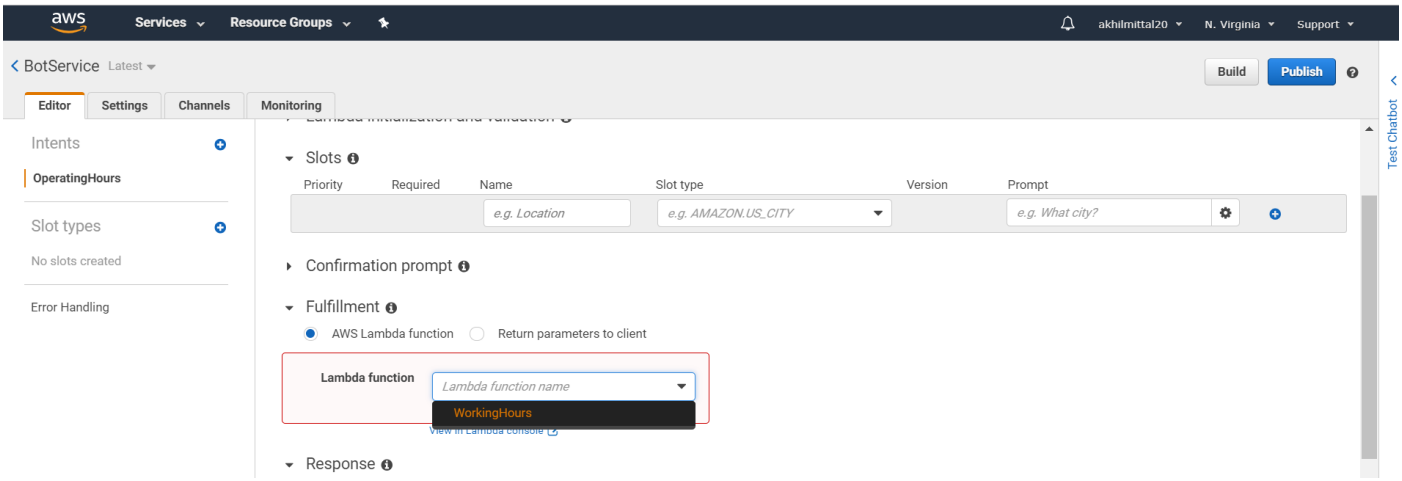
5. Click on Create function and you'll see the option to write the code. Choose "Edit code inline". The Handler details are pre-populated i.e. "**function name.handler name**". Write the function to return the text as "Our working hours are 9 AM to 6 PM IST on weekdays" as shown below.



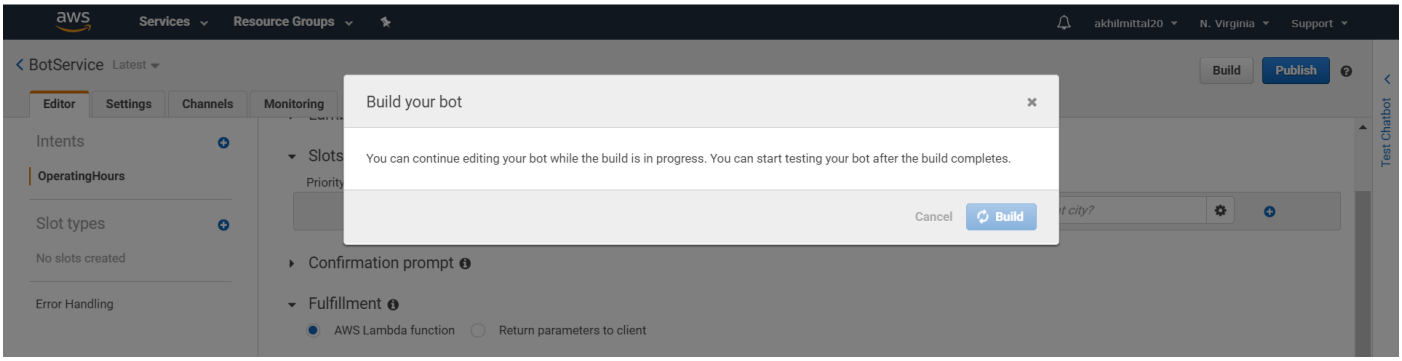
6. Test the function implementation by clicking the Test button and if everything is fine the test execution will show the result as succeeded.



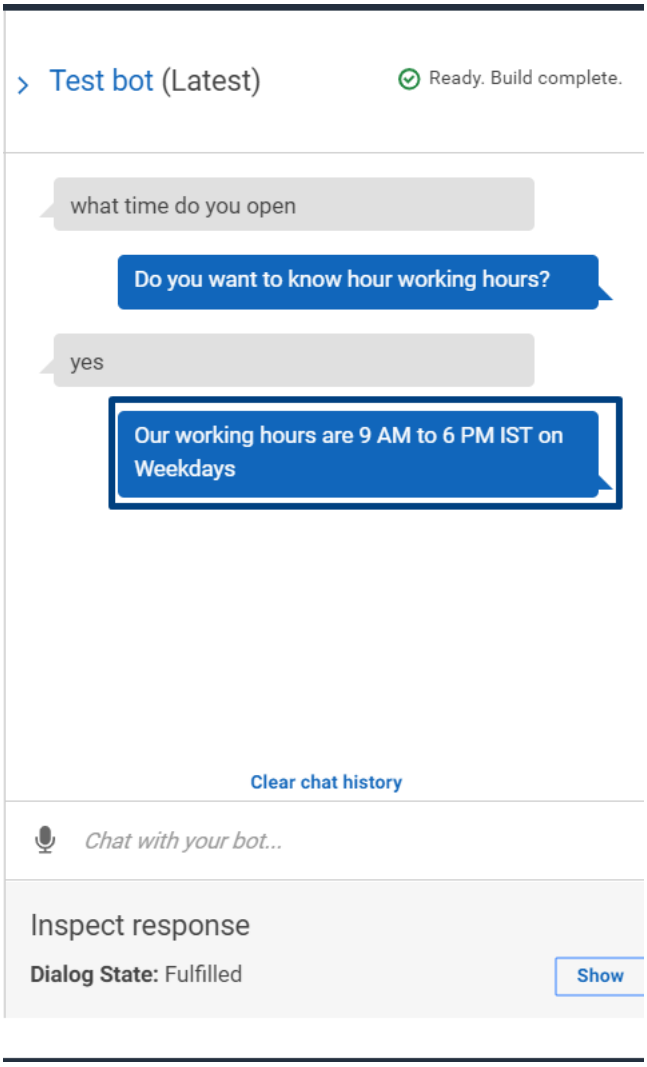
7. Now go back to the Intent and under fulfillment when AWS Lambda option is chosen, the recently created function is available. Choose that function.



8. Again, build your bot.



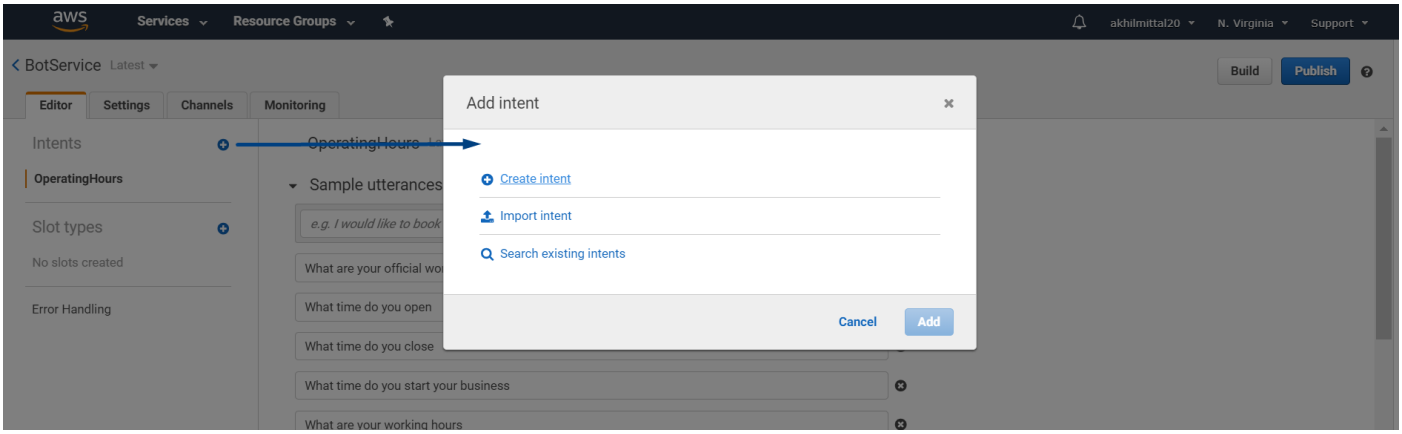
9. Now when you test the bot and after the confirmation of the bot if you type yes that means the fulfillment needs to be called and the bot here returns the text that we provided as a return text in the lambda function. So, our fulfillment and the lambda function work fine.



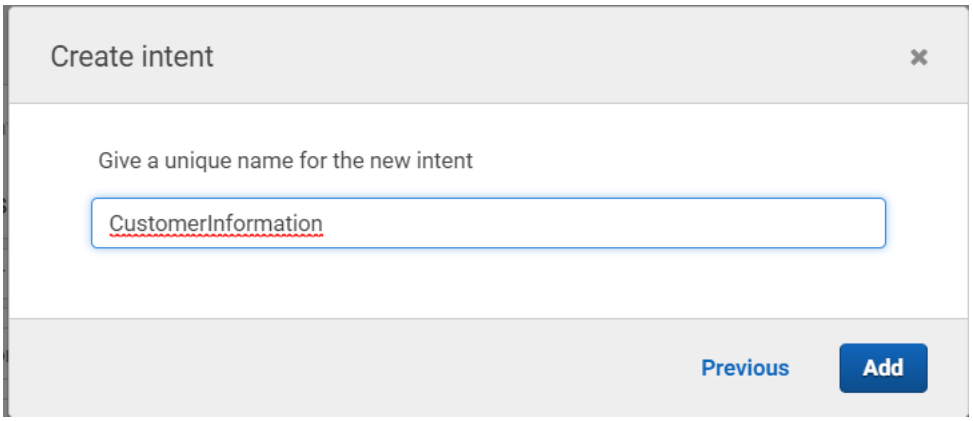
Slots

In this section, we'll see how to collect customer information using Slots.

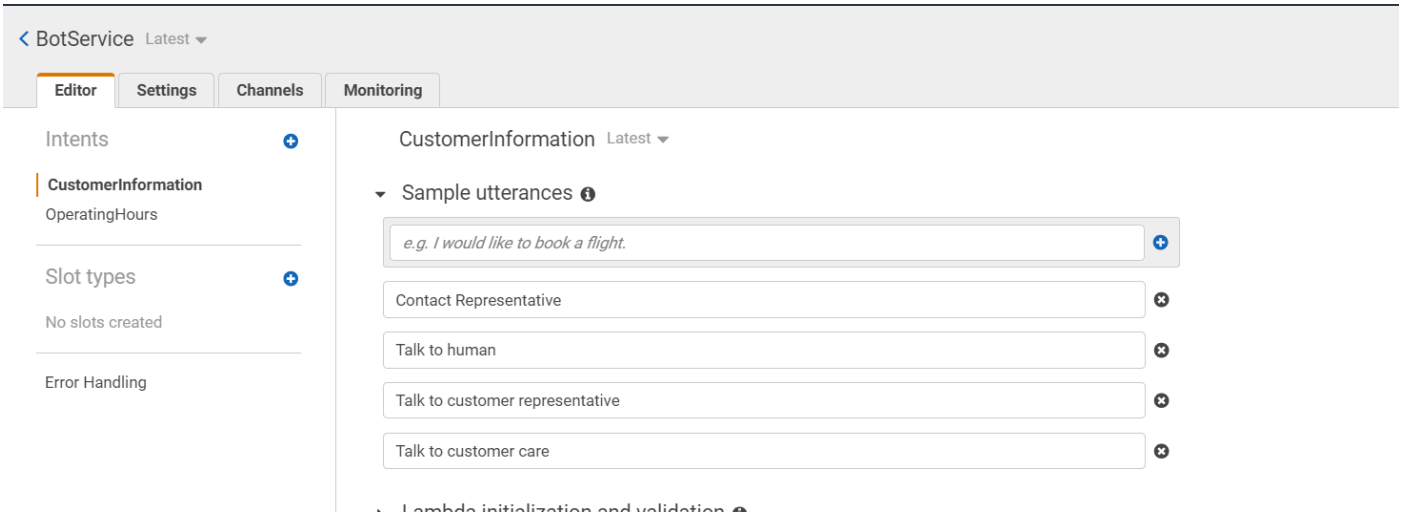
1. You can add a new intent to prompt a customer to speak to the representative.



2. Add the new intent and name it CustomerInformation.

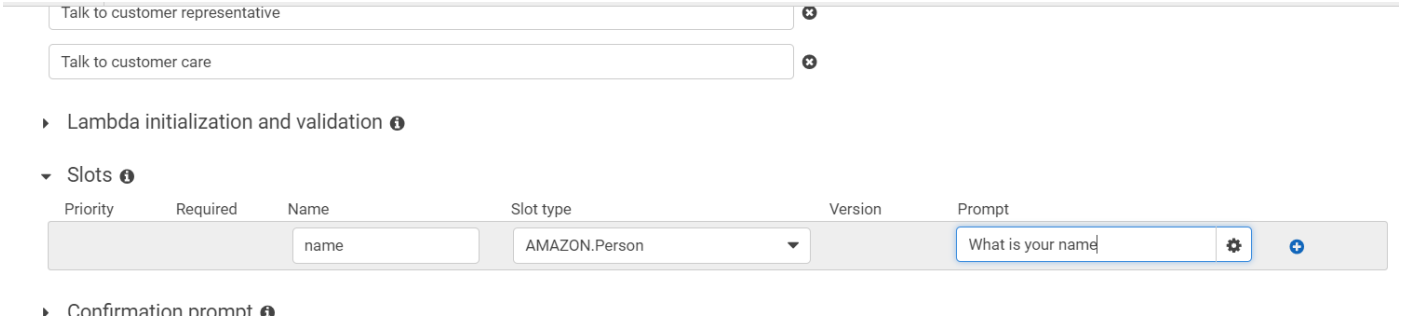


3. You can provide the sample utterances like “Contact Representative”, “Talk to human” etc.

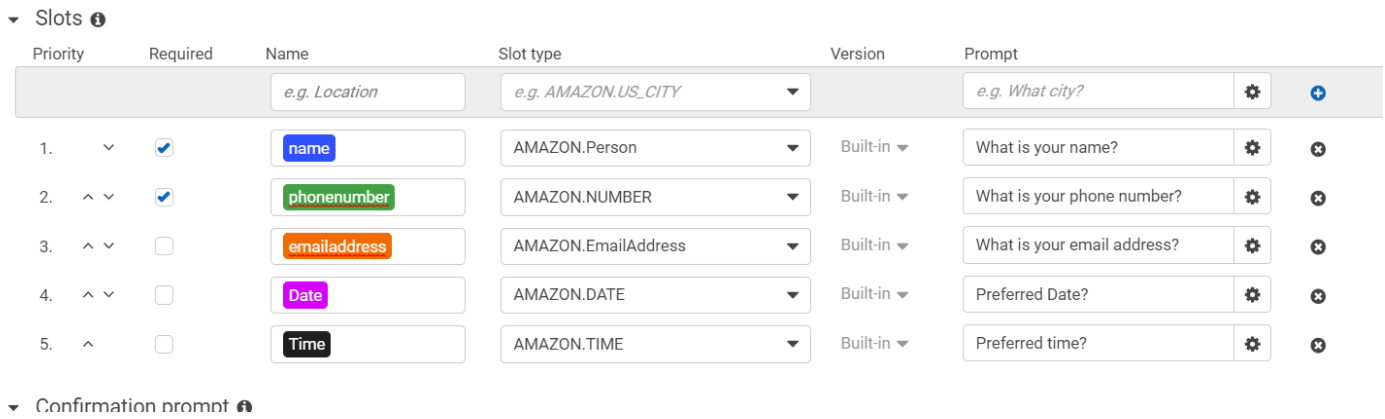


Coming down to slots section, provide the name of the slot, the type and the prompt. Lex service provides various types of slots, we’ll use those.

4. Choose slot Name as name, choose slot type as AMAZON.person and prompt as “What is your name”. You can explore more on slot types by checking in the dropdown on what information you need to seek from the customer.

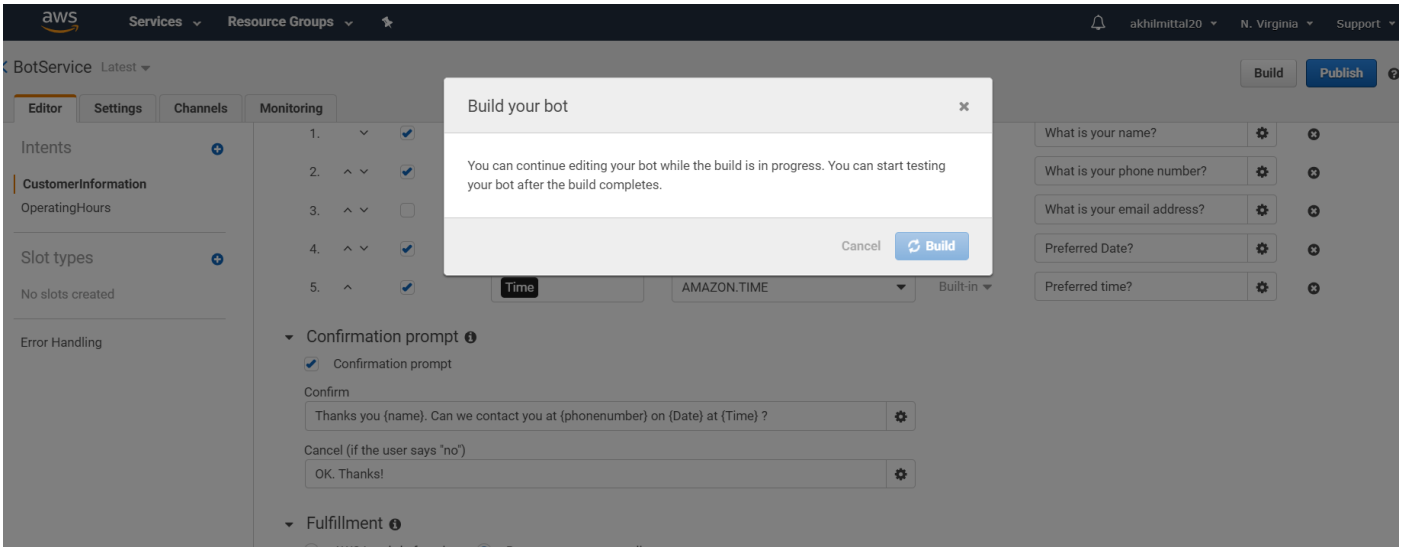


We'll add few more slots like phone number, email id, preferred date of calling to the customer, preferred time of calling to the customer. Check the following picture for the implementation.



Chatbot collects all the information from the user when the user intents to speak to the customer. You can check the Required checkbox to seek all the mandatory information.

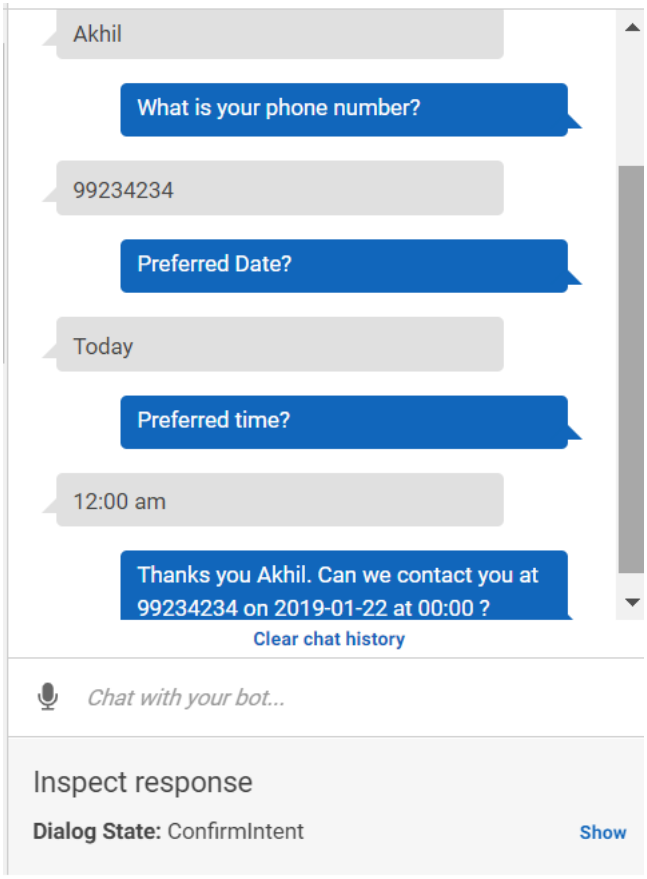
5. Now add the confirmation prompt to make sure that all the information provided by the customer is correct. In the Confirmation prompt, you can reference Slot values by surrounding the Slot name with curly braces. So, we will include name, phone number, Date, and Time in our Confirmation prompt. If the user decides they don't want us to call them and says no at the confirmation, we will just say "OK. Thanks!".
6. Now we can save, build, and test our bot.



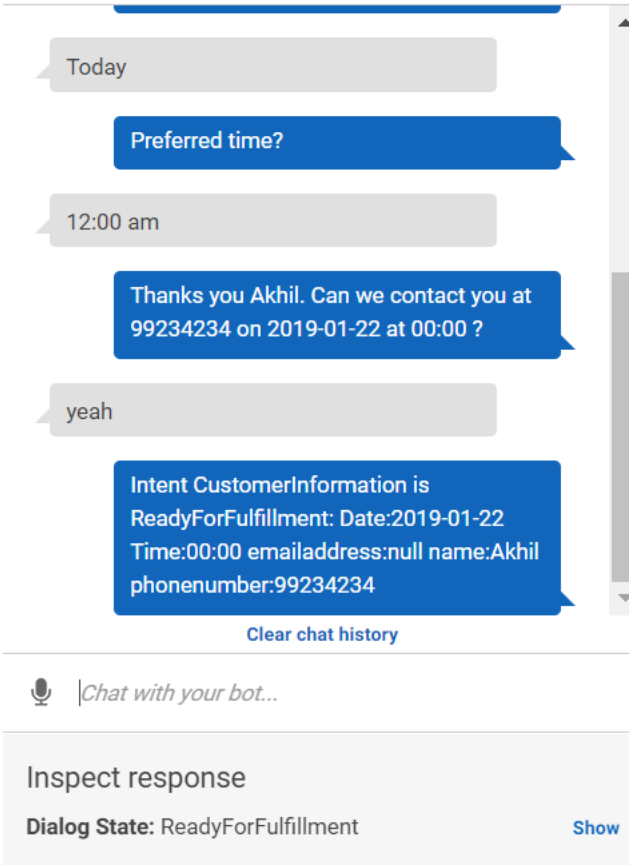
Testing Slots

After the bot is built again. You can test the intent and the slots again to check if the bot is working fine.

So, when you type the sample utterances phrase like “Speck to a representative”, the bot starts collecting information as asked in the slot.
It asks, “What is your name?”, “What is your phone number” etc whatever was provided in the slot.

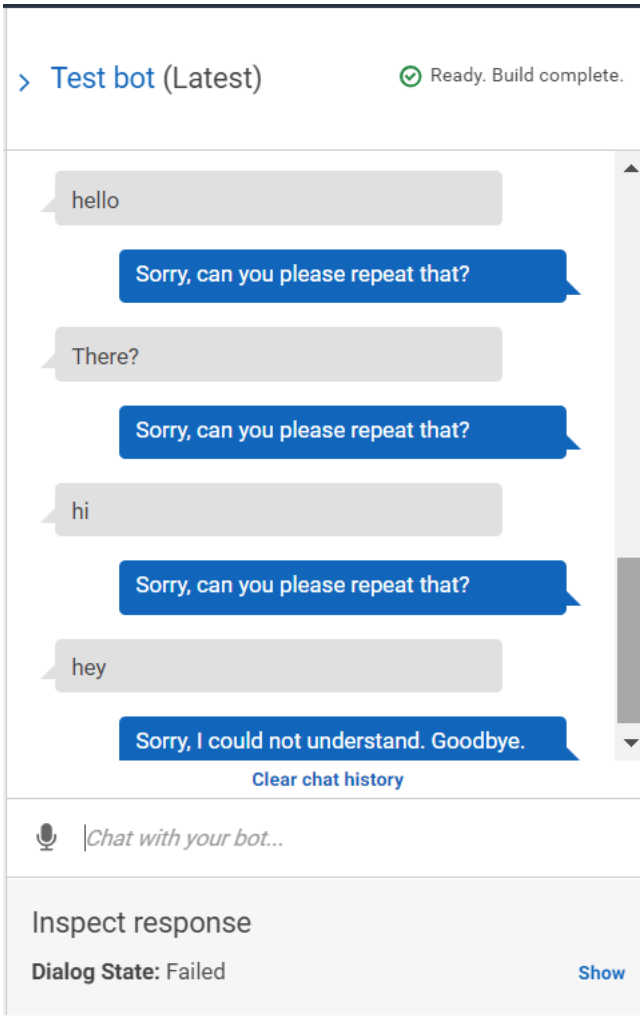


Once all the information is provided, it shows the confirmation prompt with the customer details that we provided I the prompt message for confirmation in curly braces.

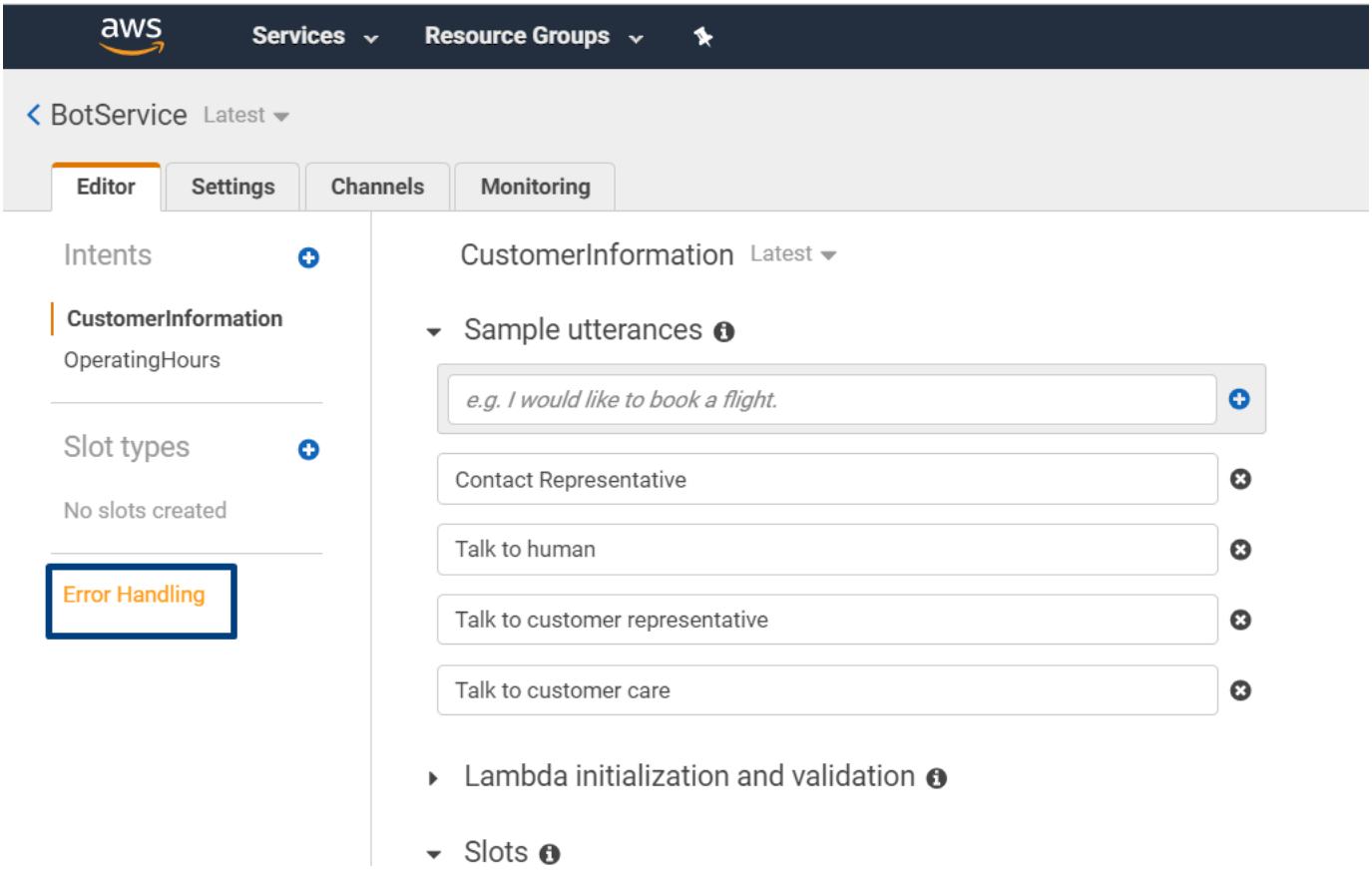


Error Handling

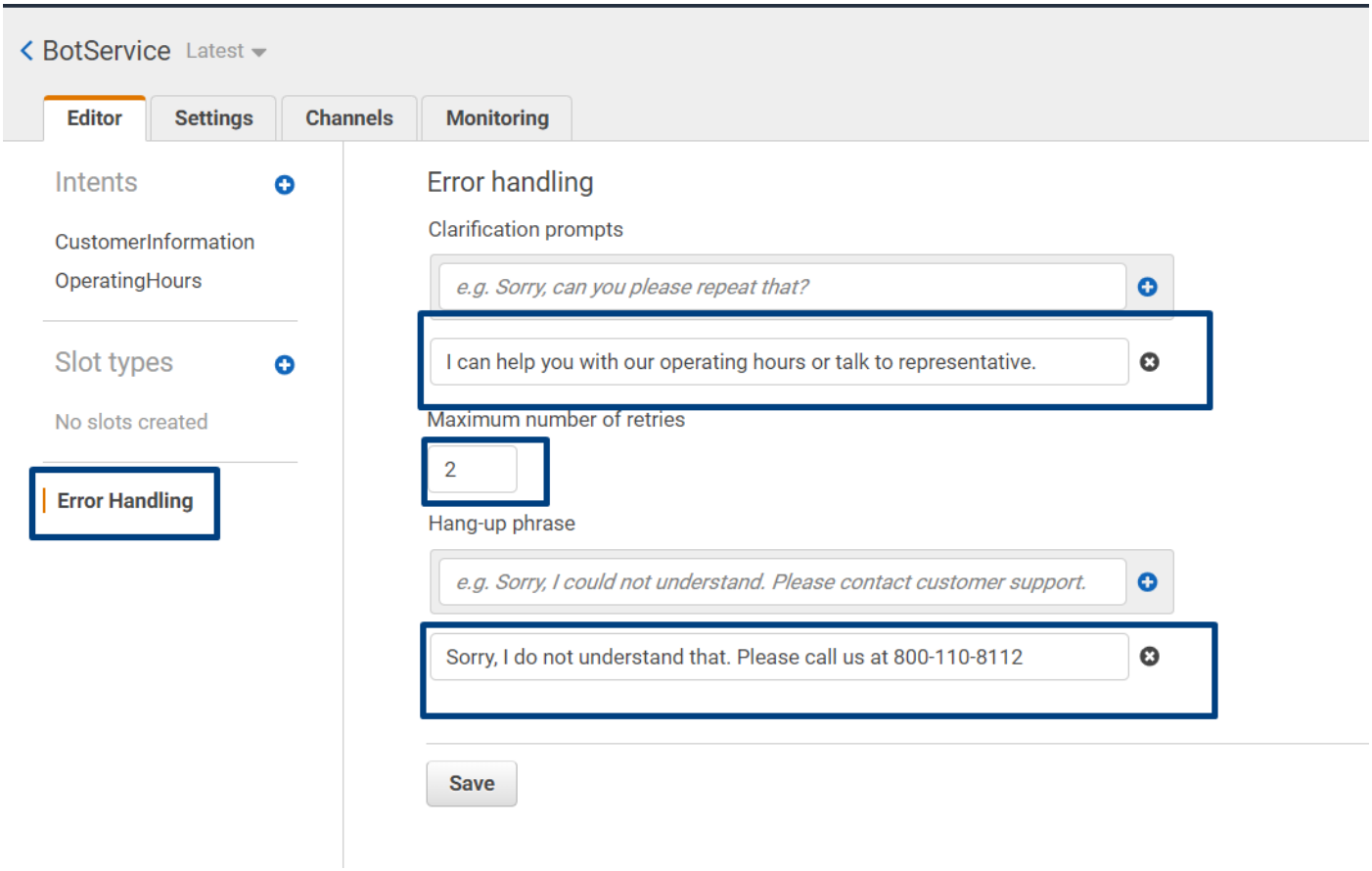
We saw that our bot does not respond well to the user saying “Hi” or “Hello” or something that is not mentioned in the intent utterances and it keeps on showing the defaulted message and after few prompts, it says that it does not understand what user is saying and goodbye. This is all configured for the bot as the default setting. We can change that as per our requirement using the Error Handling feature of Lex.



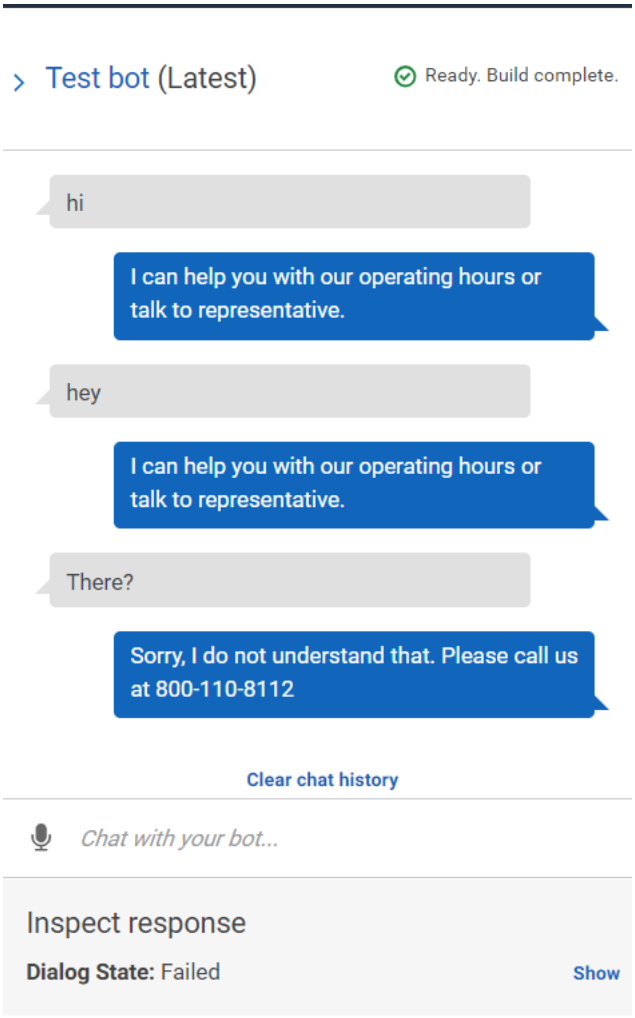
1. Click on the orange color link saying Error Handling on the BotService.



2. In the clarification prompt, give the prompt on what your bot is capable of so that it shows the meaningful information to the user if user types in anything that bot do not understand. You can limit a maximum number of tries as your leisure. I am limiting it to 2. After the two tries if still, user does not ask what the bot actually understands then give a meaningful hang-up phrase that user can calls the customer care at a particular number or something else meaningful. You can delete the defaulted clarification prompt and hang-up phrase now.



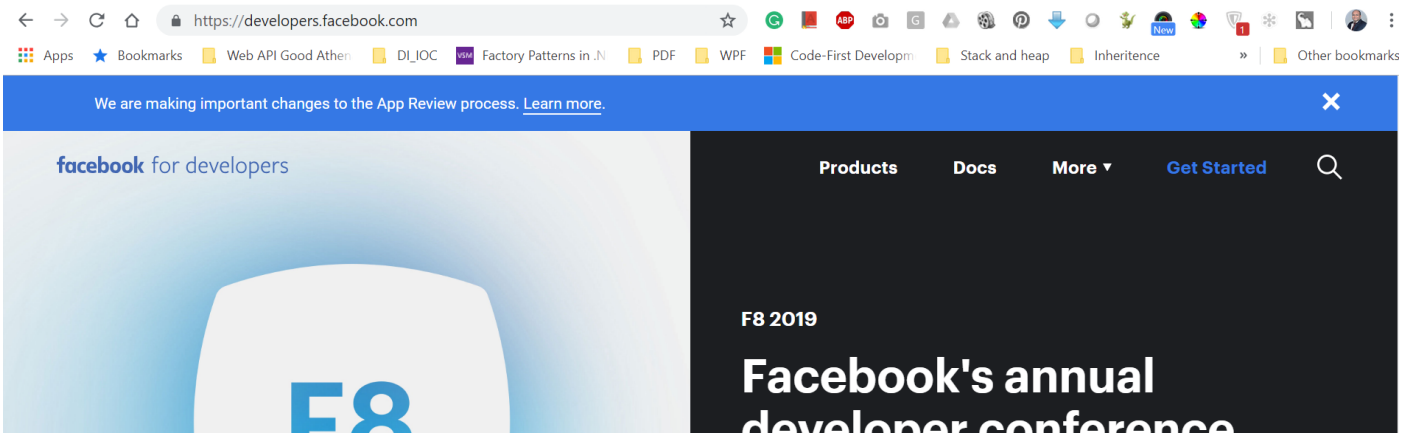
3. Now test the bot and you see that when we type phrases like “hi” or something else the bot prompts for two times on what it can help with and after that shows the hang-up phrase.



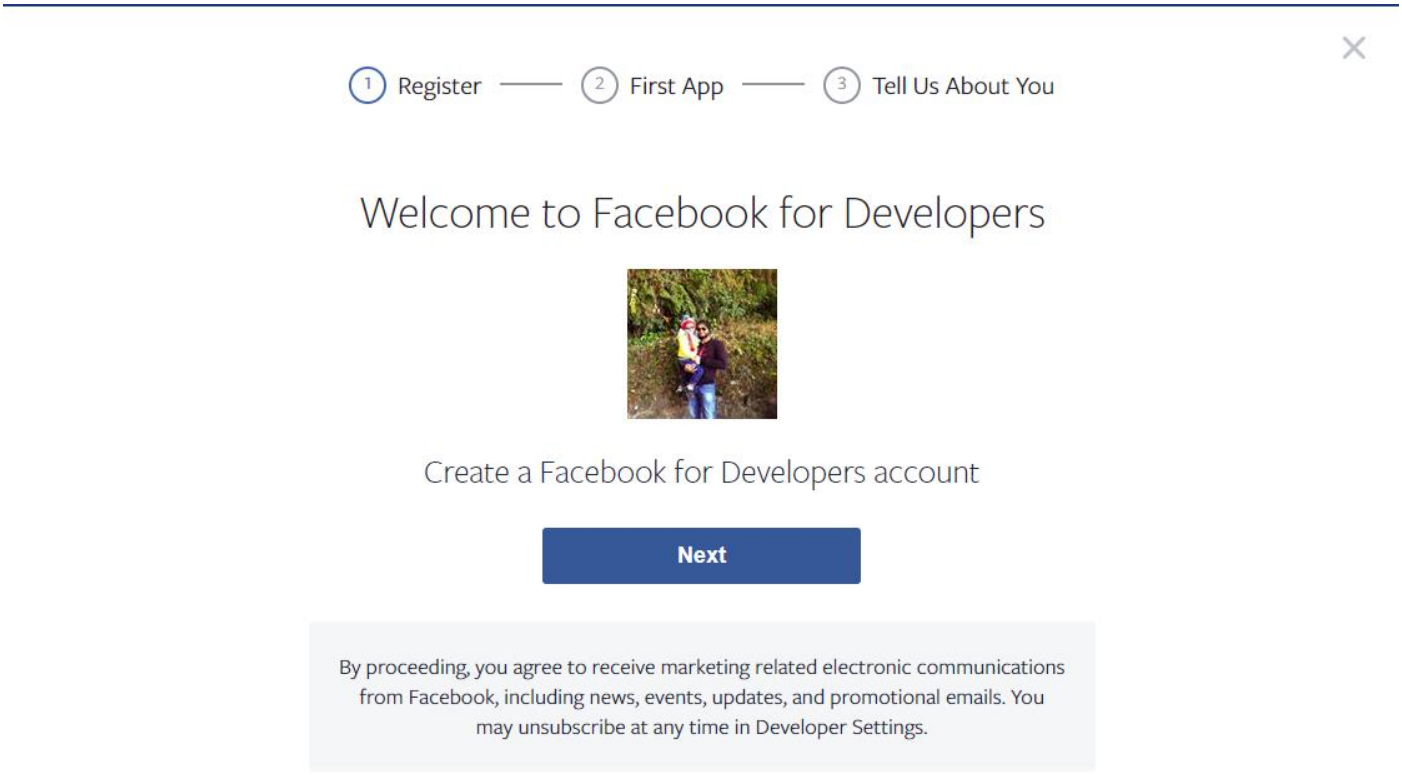
Deploy the Chatbot in Facebook Messenger

Our initial basic bot is completed, you can enhance it more as per your will and requirement. In this section, we'll see how to deploy the bot on Facebook messenger as a chatbot to your custom Facebook page.
So, the need here would be that suppose you have a Facebook page for your business or company and you need to deploy a feature of auto chatbot in that page so that any user can seek information or talk to that bot.
First things first, for deployment of the bot to the Facebook page, we need to have a Facebook page.

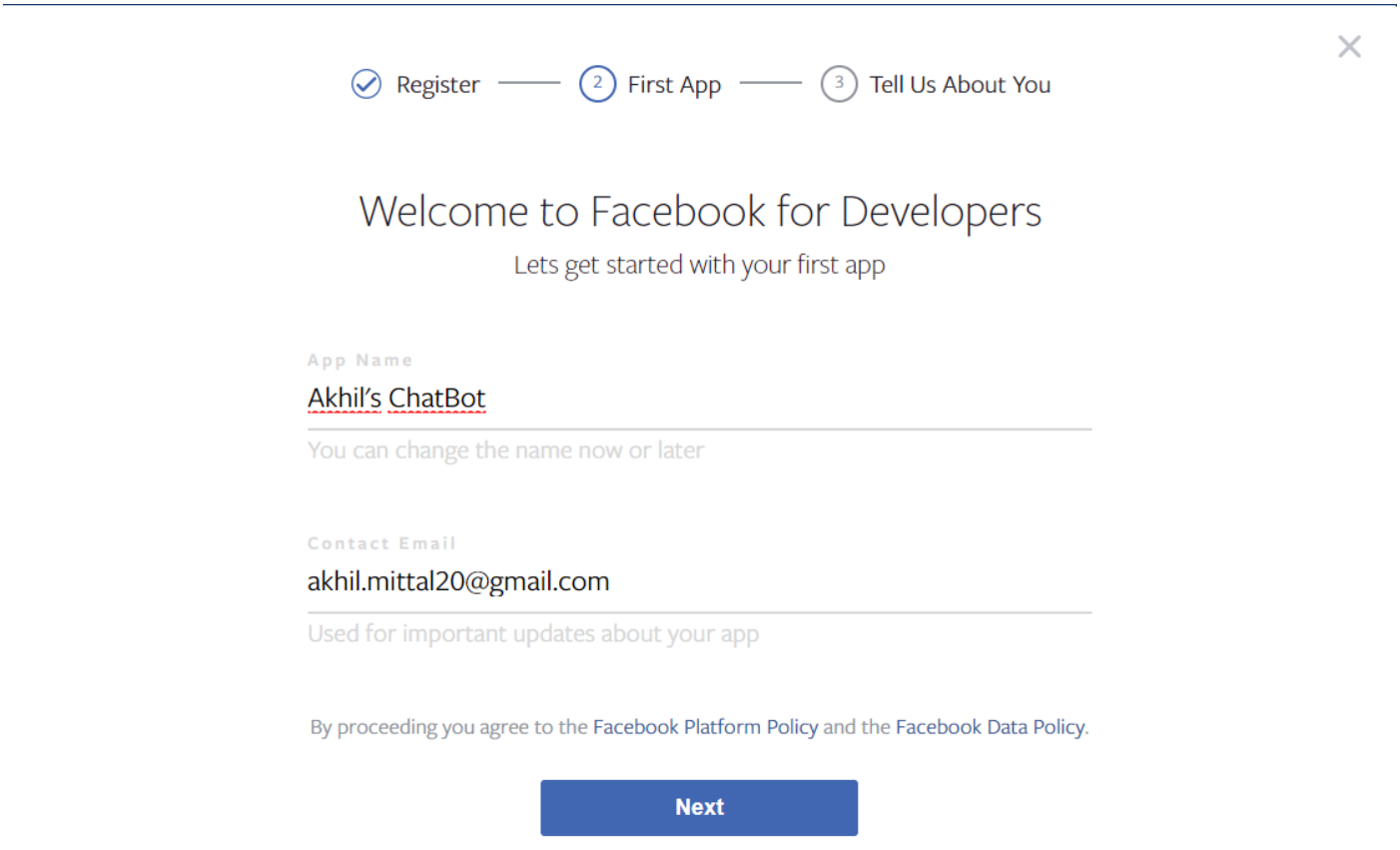
1. Open the URL https://developer.facebook.com and sign in with your existing Facebook credentials or create a new account



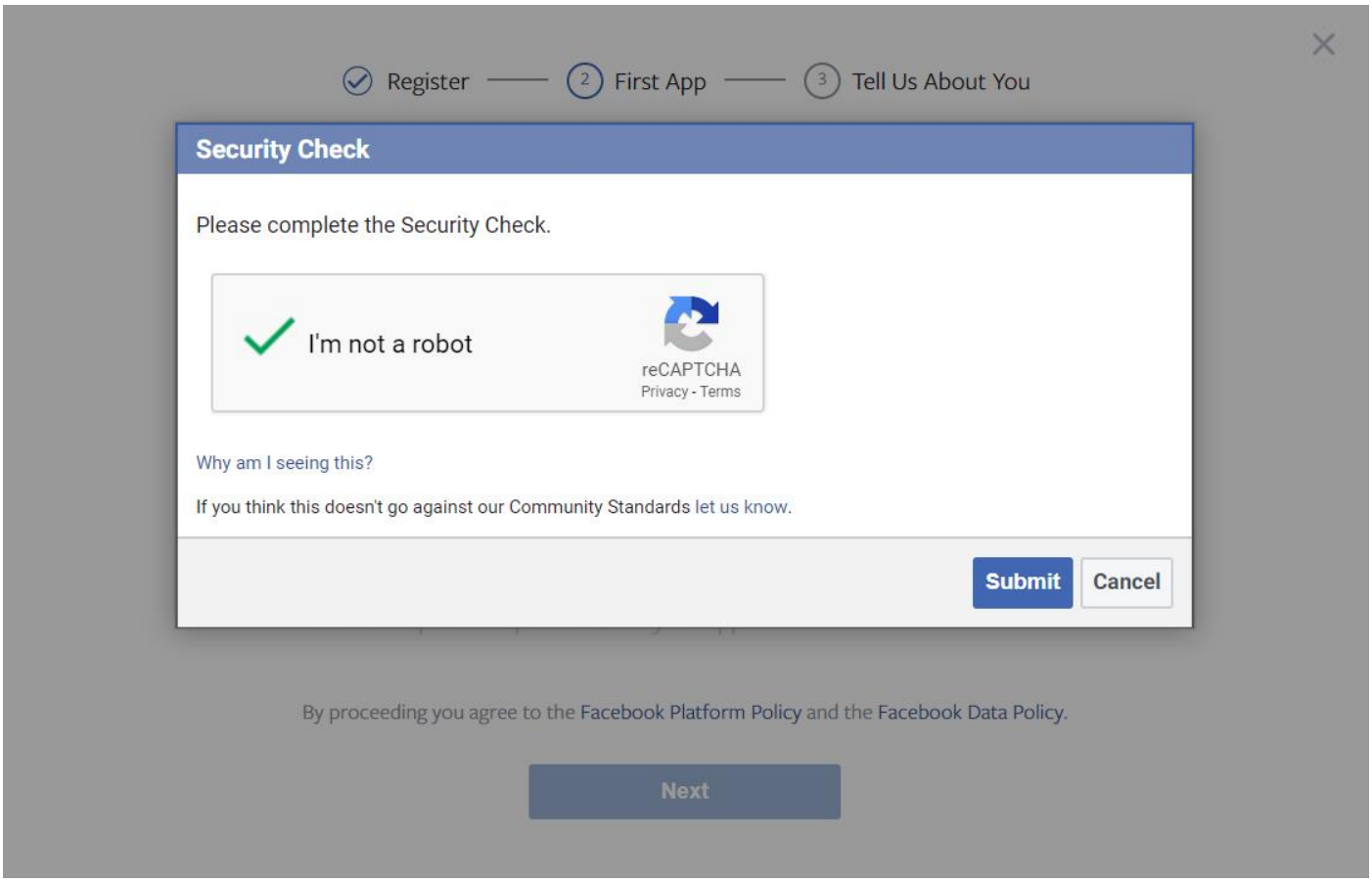
2. I am signing in with my existing Facebook account. It will ask to create a Facebook developers account, and yes we need to, so click Next.



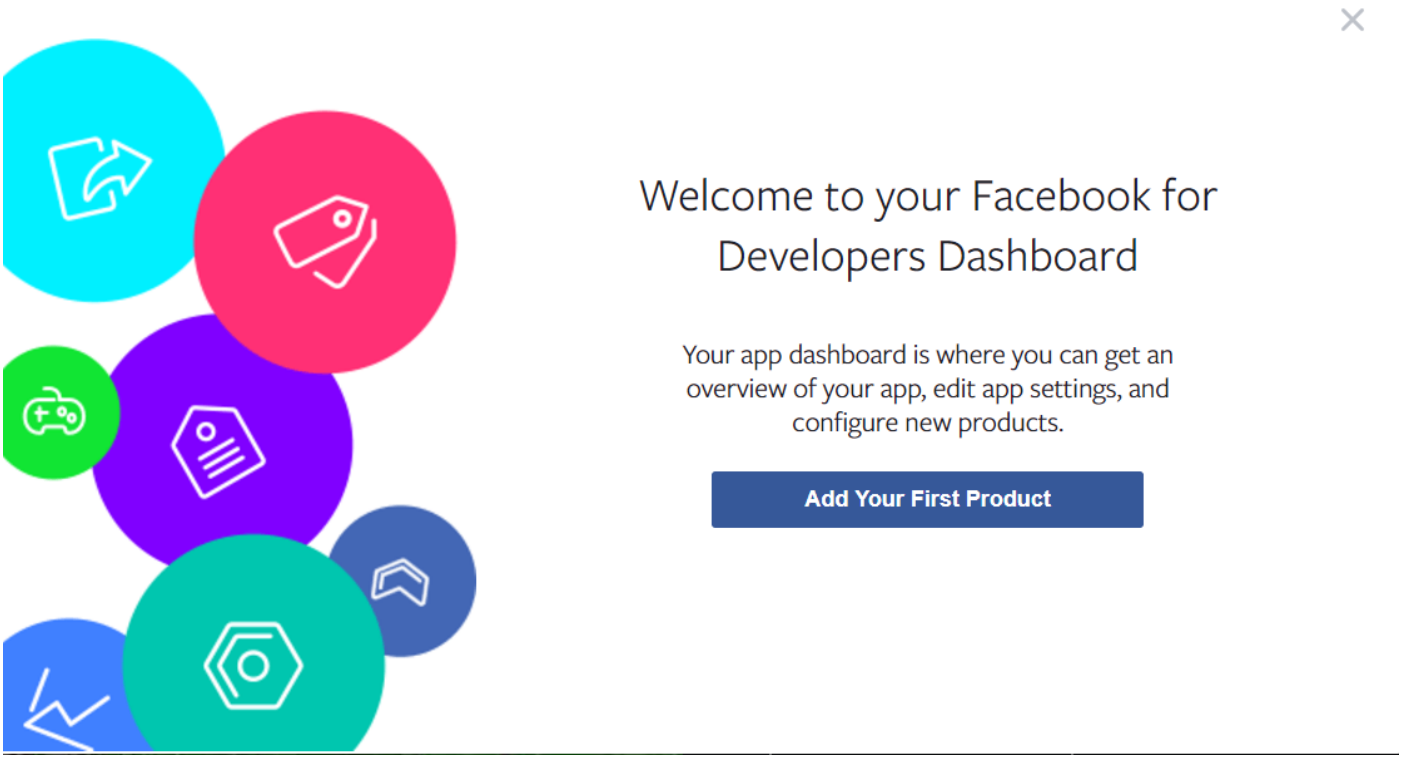
3. Provide the App Name (whatever you want) and contact email and click Next.



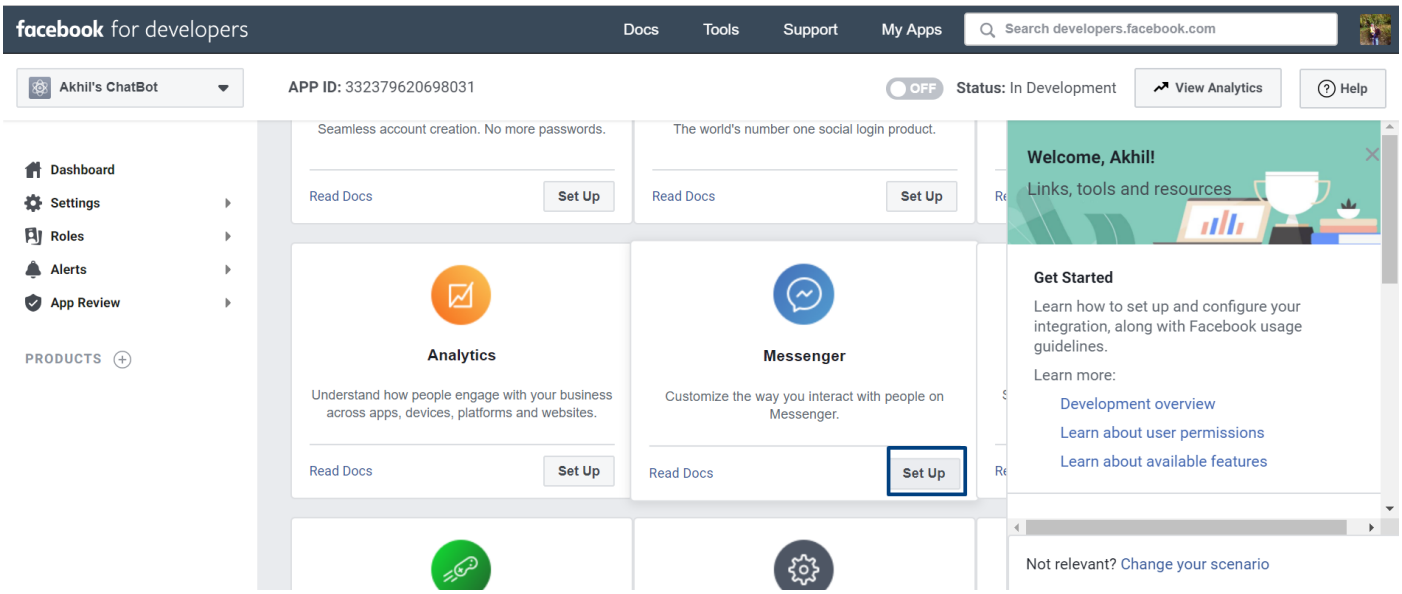
4. Pass the security check and click Submit.



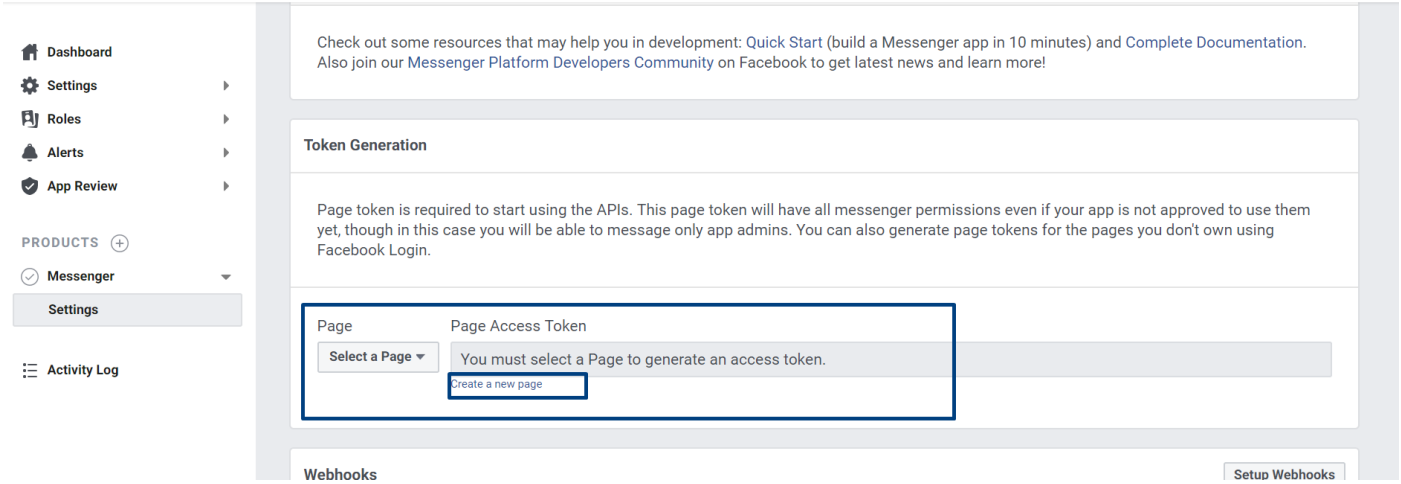
5. It will show the welcome message and ask to add your first product. Click on “Add Your First Product” button.



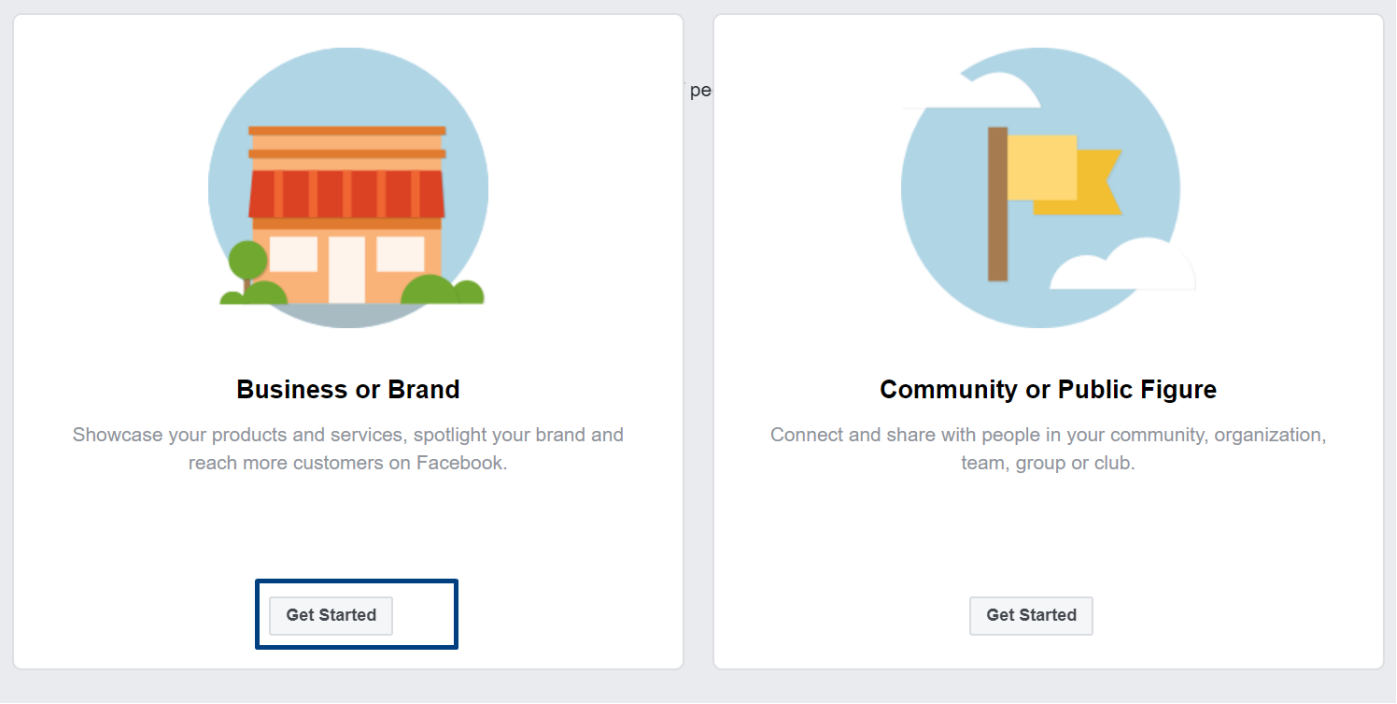
6. The next screen you'll see is the Facebook developer's dashboard. Since we need to set up the Facebook Messenger, click on the button to setup messenger as shown below.



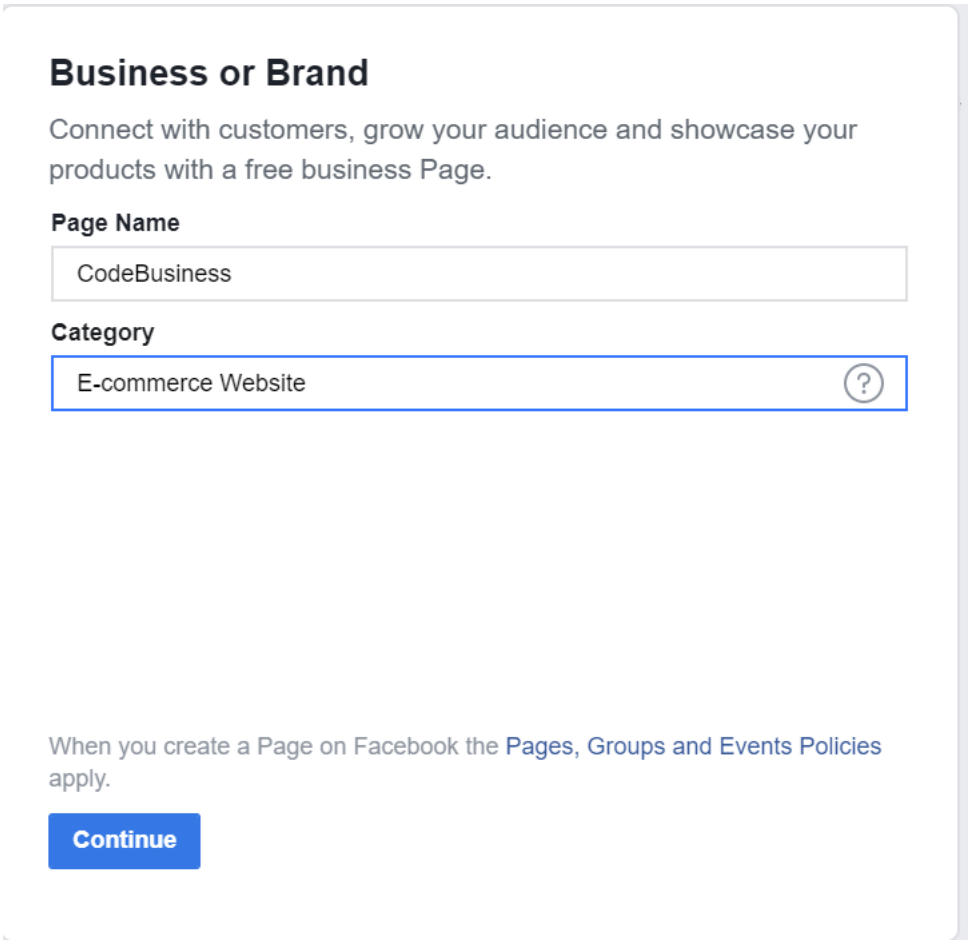
7. You need to have an access token and a secret key for that. So, navigate to Settings and scroll down to the section that says Token Generation. Select an existing page if you have or create a new one as per your wish.



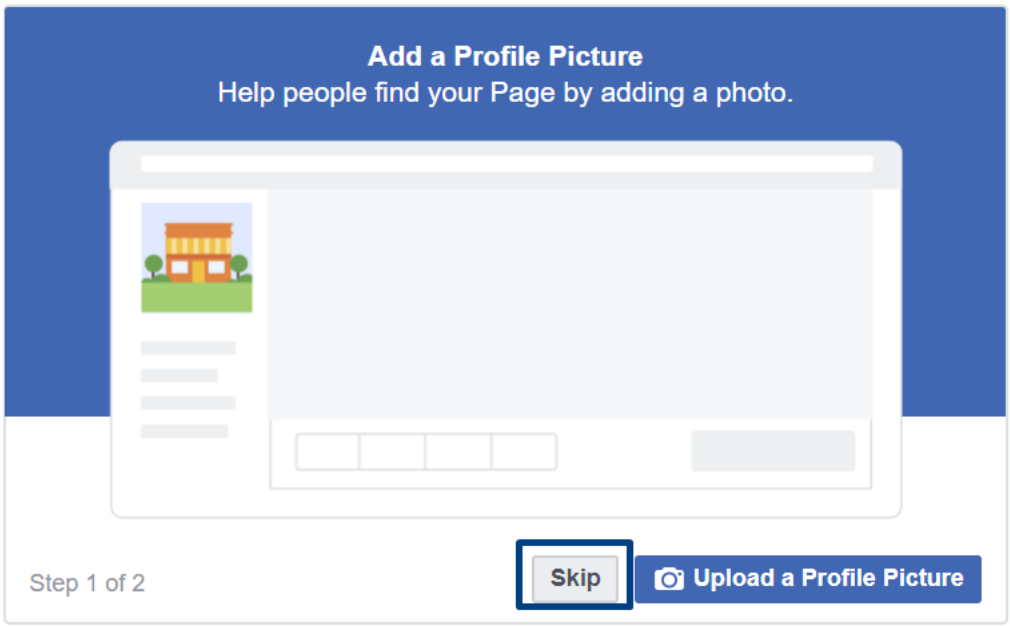
8. Select the type of page you want to create i.e. Community or Business.



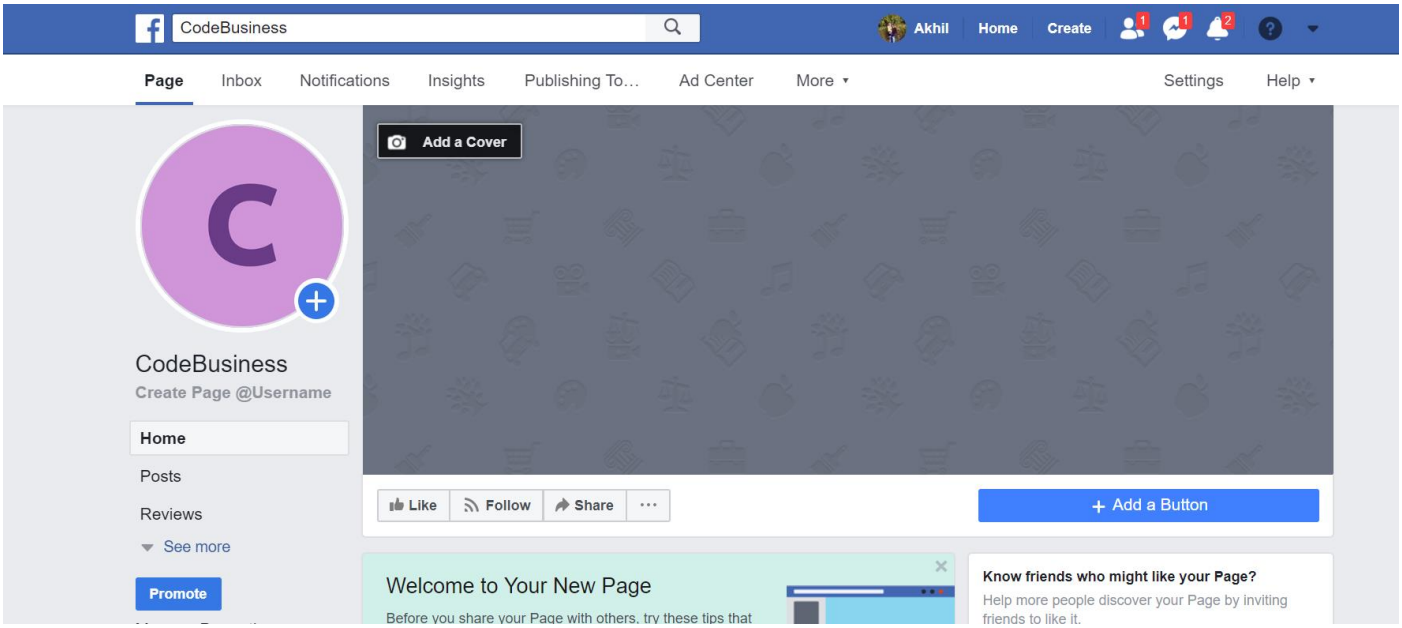
9. Provide the page name and the category.



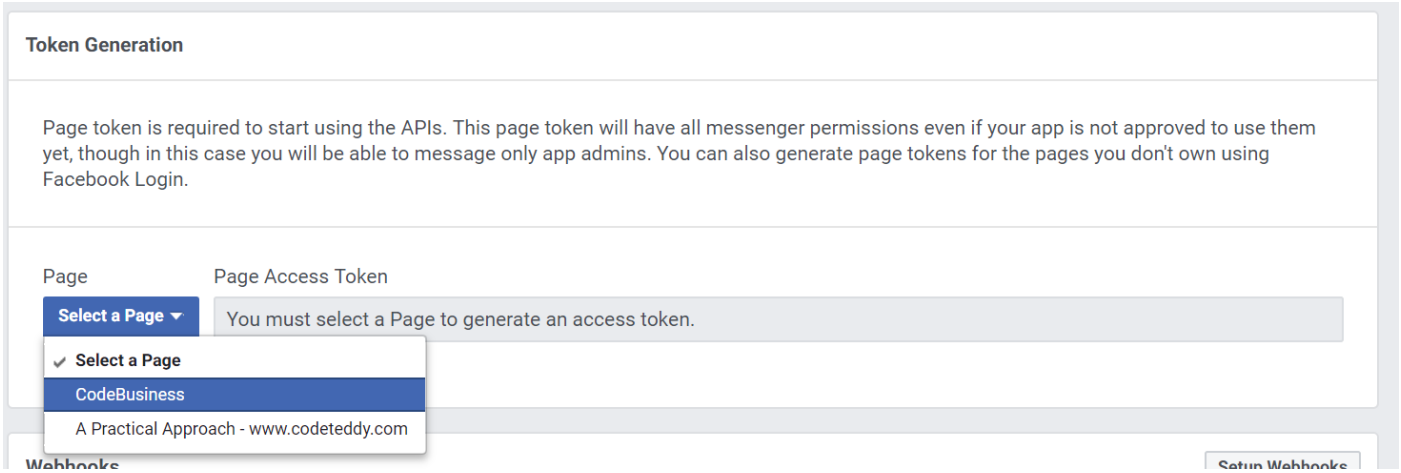
10. You can skip the steps like providing profile pic or choose to apply one.



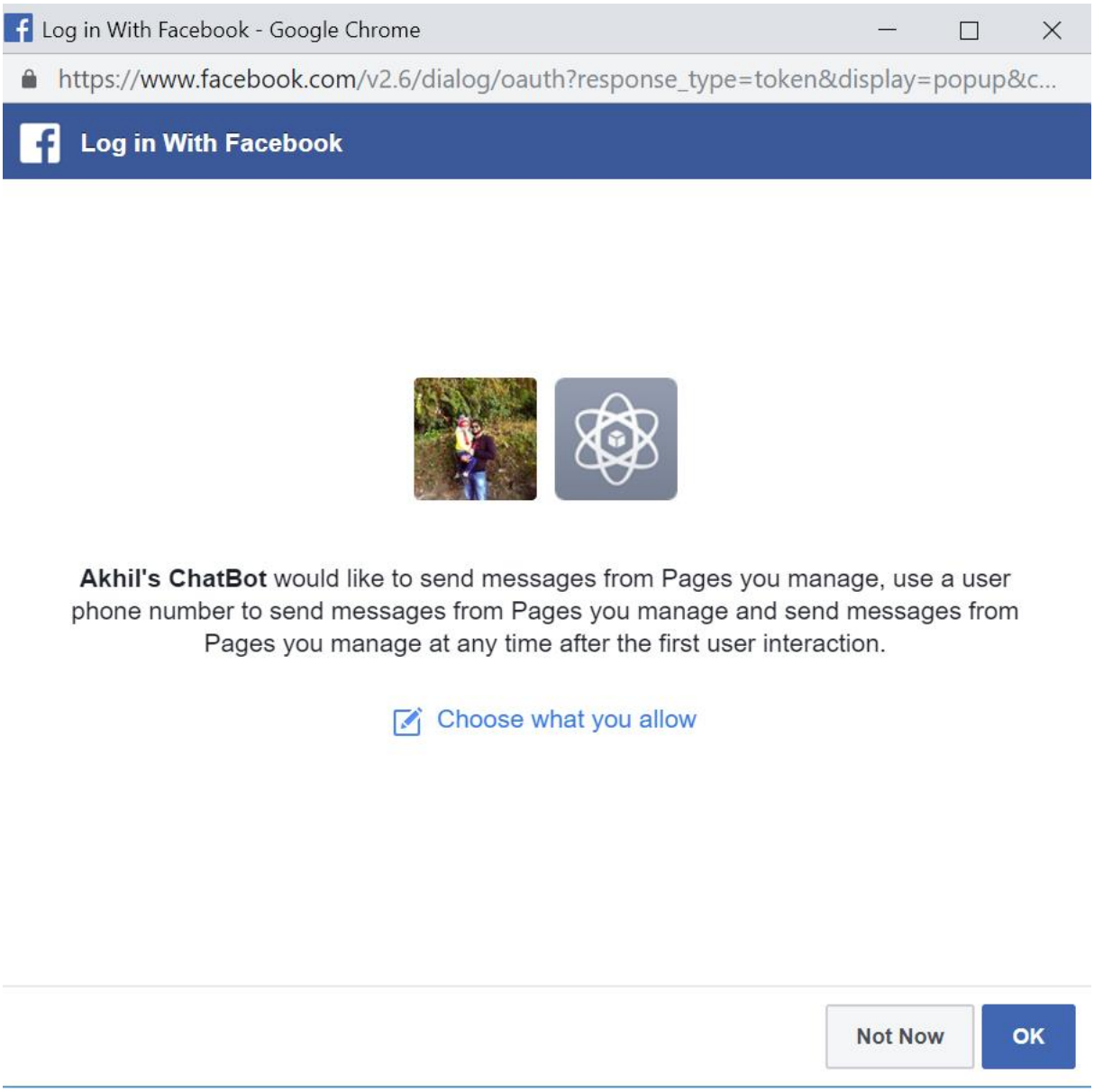
11. In the next step, you'll see the page created.



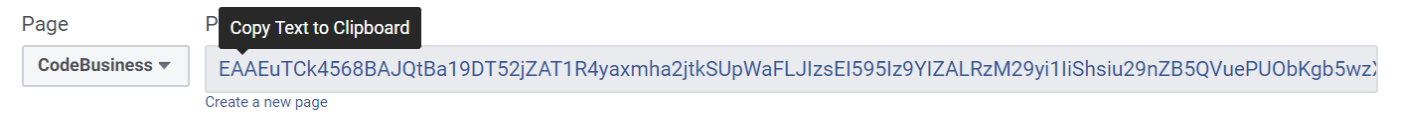
12. Now again go to settings and Token generation section and choose the newly created page.



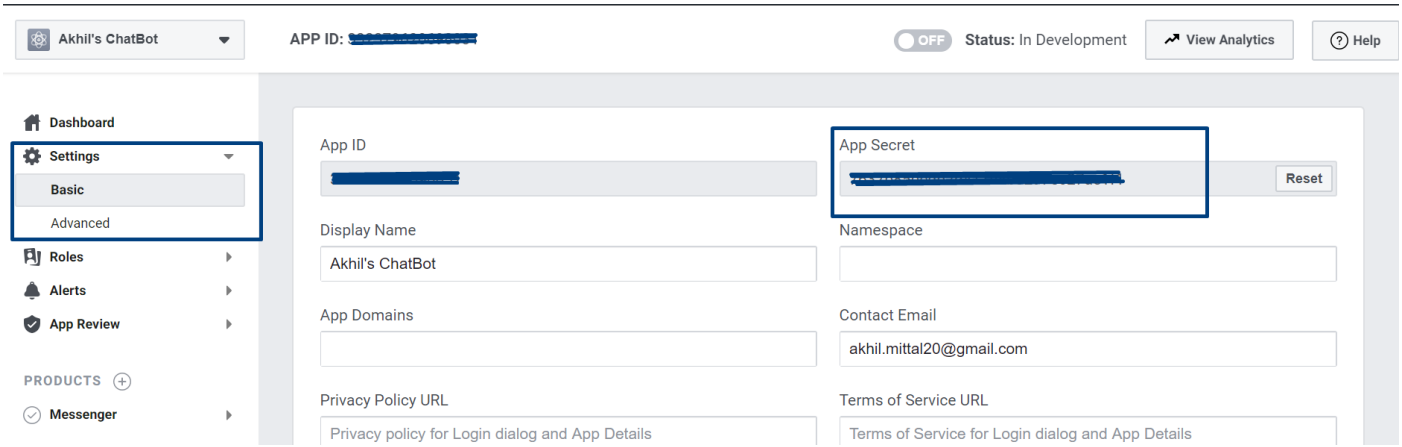
13. It will again prompt you for authentication. Do that.



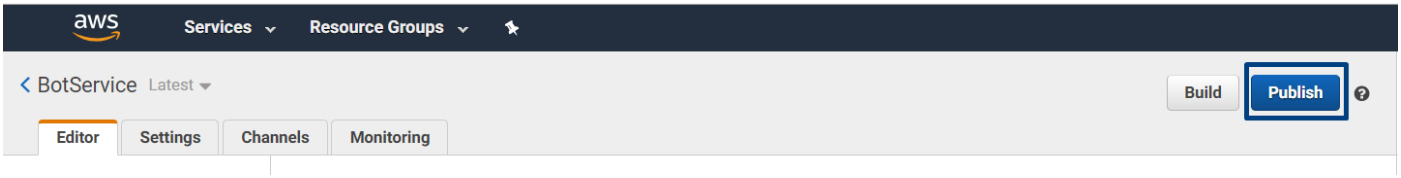
14. After the authentication, in the Token Generation section, we see now a Page Access token is generated. Copy that token and keep that at some location. We'll soon need that.



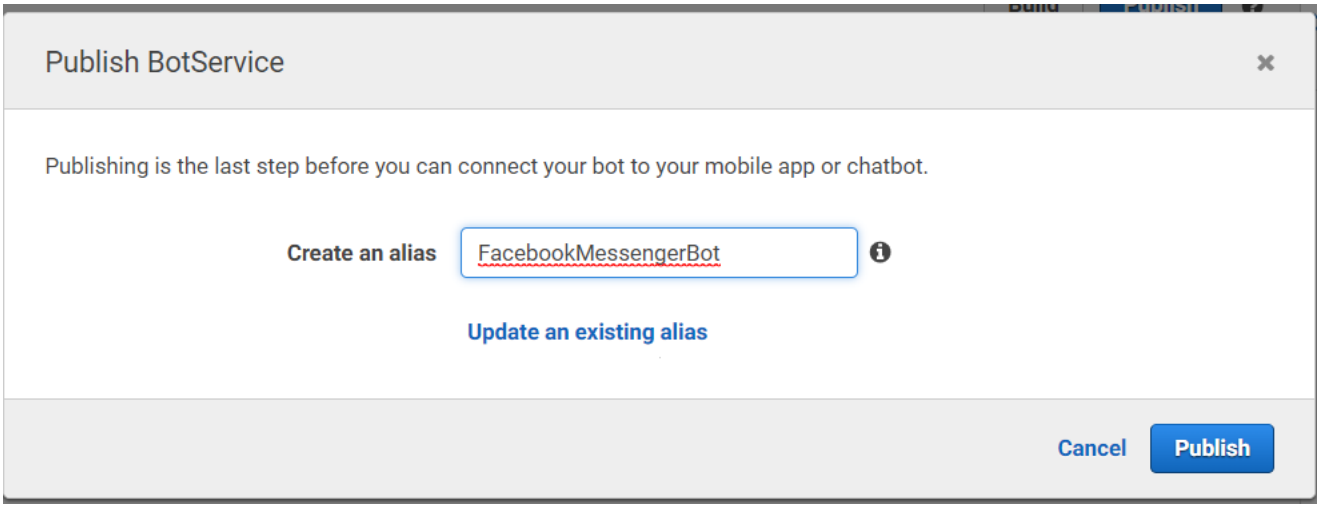
15. No, go to Settings->Basic to get the App Secret key as shown below. Copy and keep the App Secret key at some safe location, we'll need that soon.



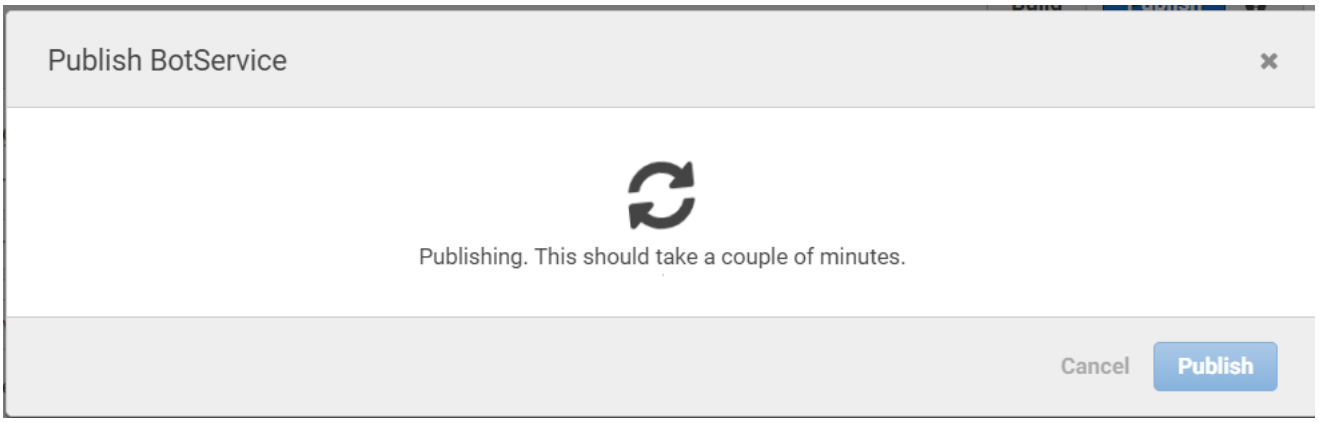
16. Go back to the Amazon Lex service, its time to publish your bot to the Facebook Messenger for the newly created Facebook page. Click on Publish on the BotService.



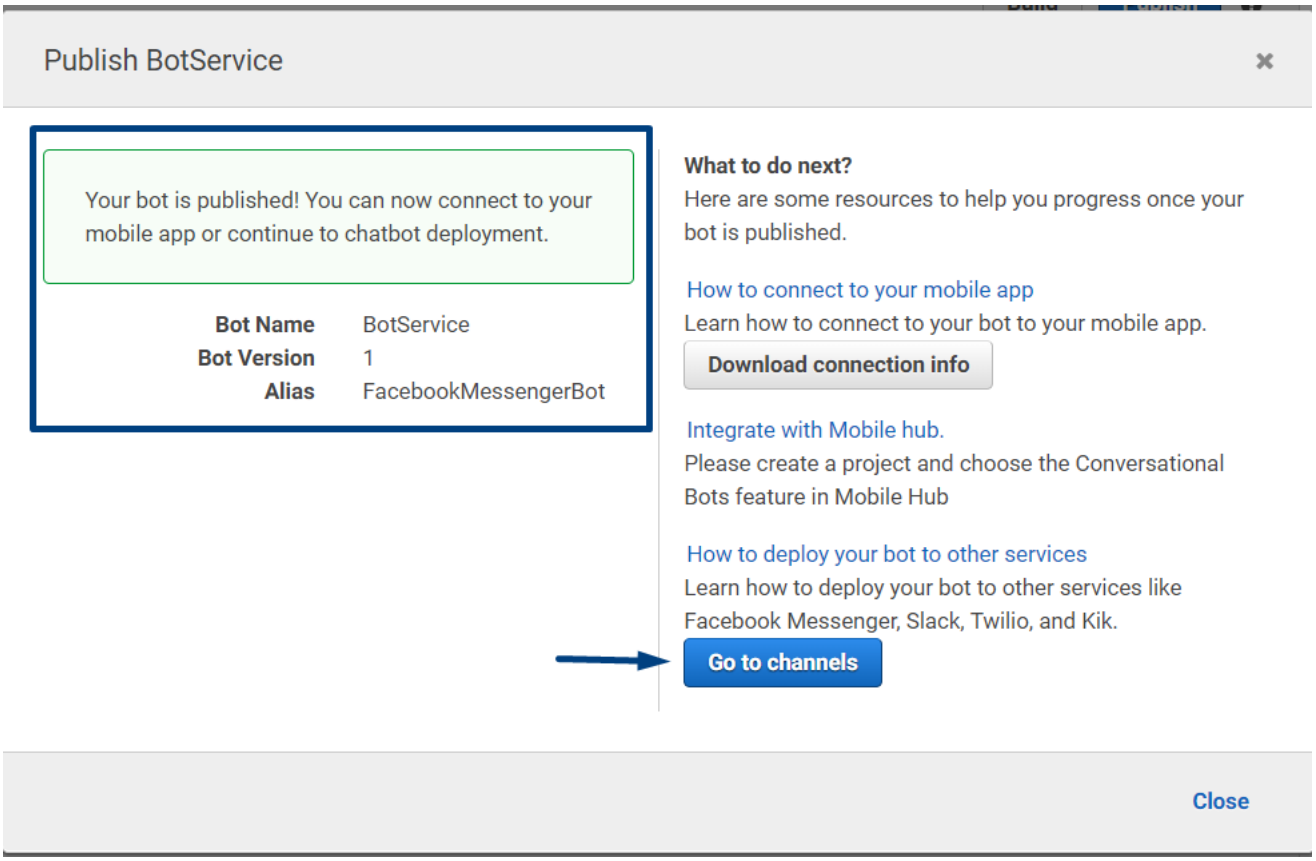
17. It will ask for the alias. Provide the alias name of your choice and remember it or store it somewhere.



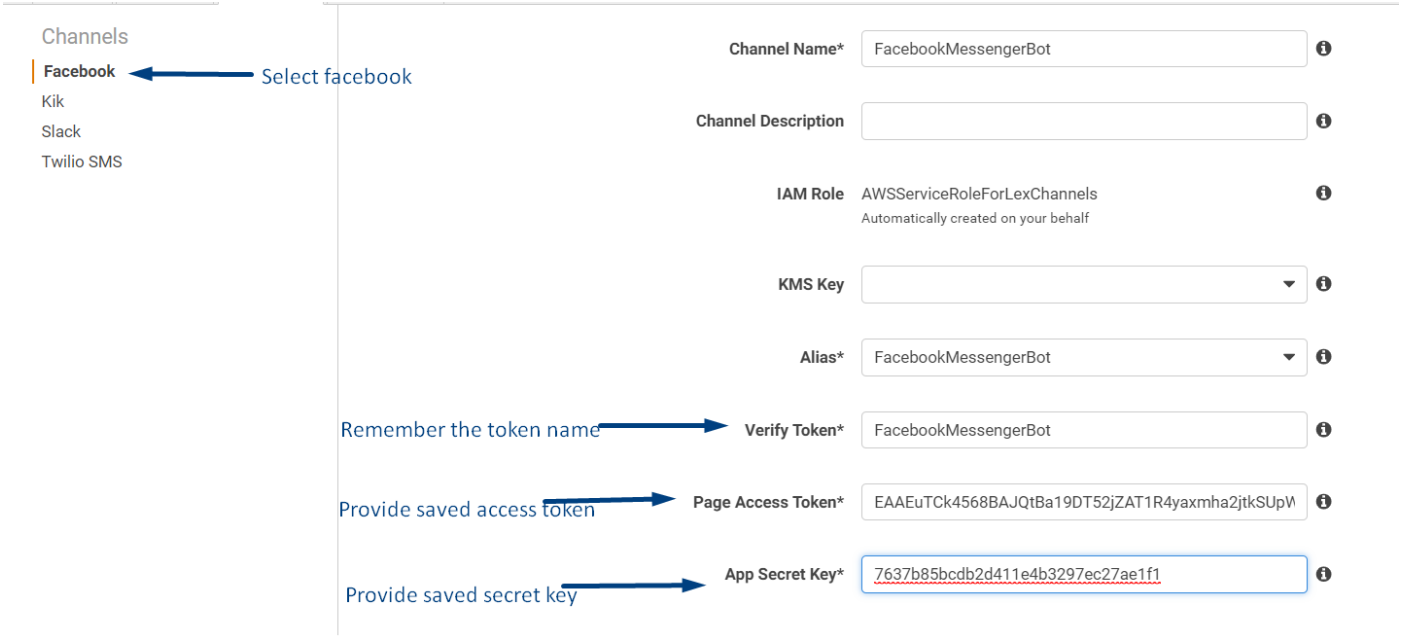
18. Click on publish, which may take some time.



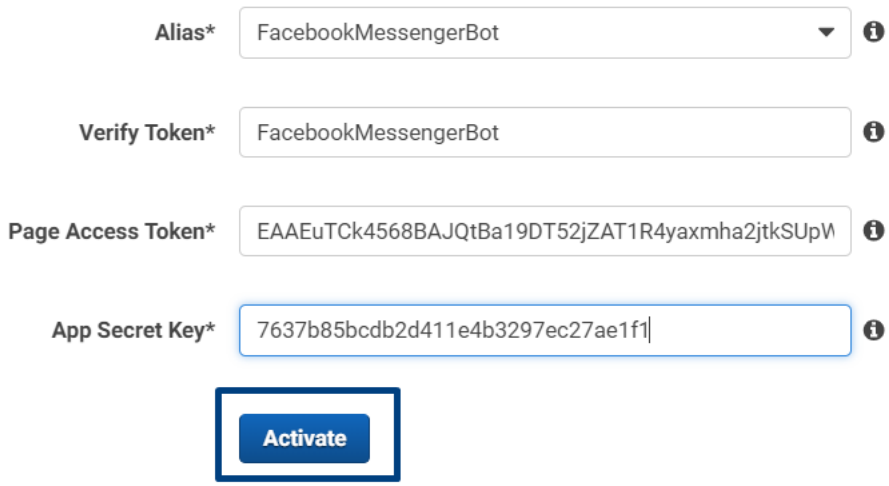
19. Once the bot is published, you'll see the confirmation message and a blue colored button which says, "Go to channels".



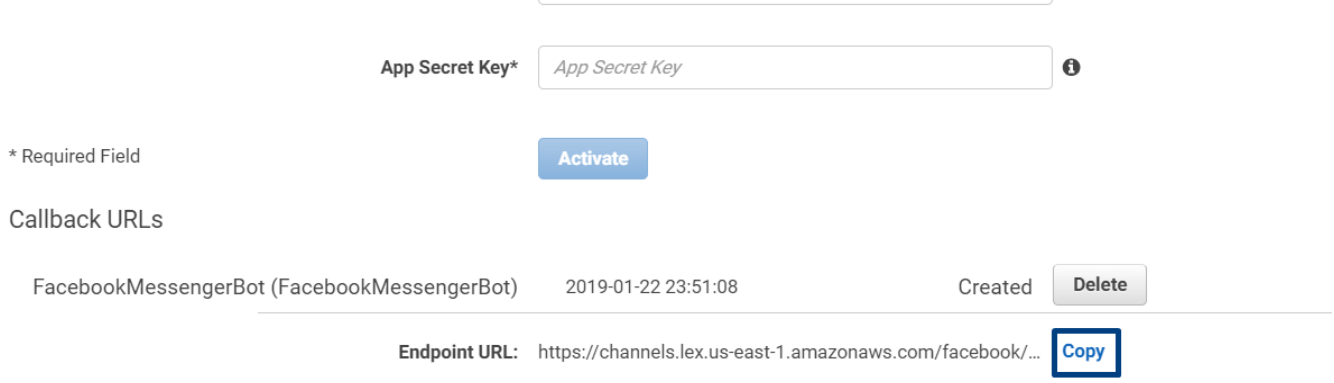
20. This will show up different channels like Facebook, Kik, Slack, and Twillo. Since we need to integrate that bot to Facebook, choose that. Provide the name of the channel, the alias that we created while publishing. Give a token name and remember it or store it. Provide the saved access token and the API secret key that we stored earlier.



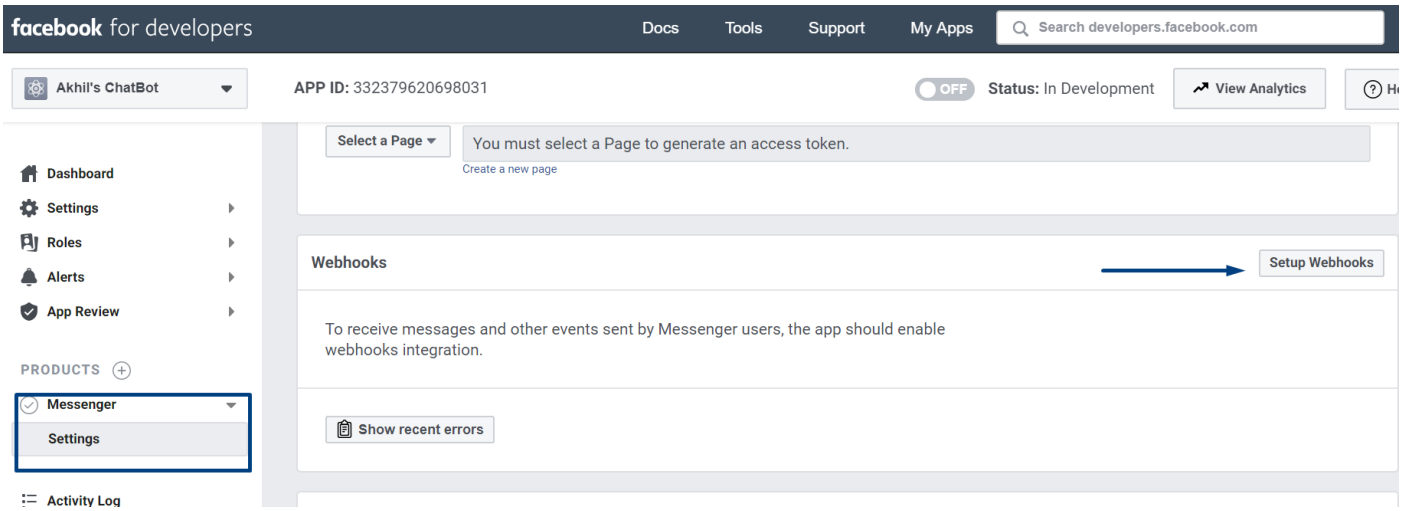
21. Click on Activate after providing all the information on that page.



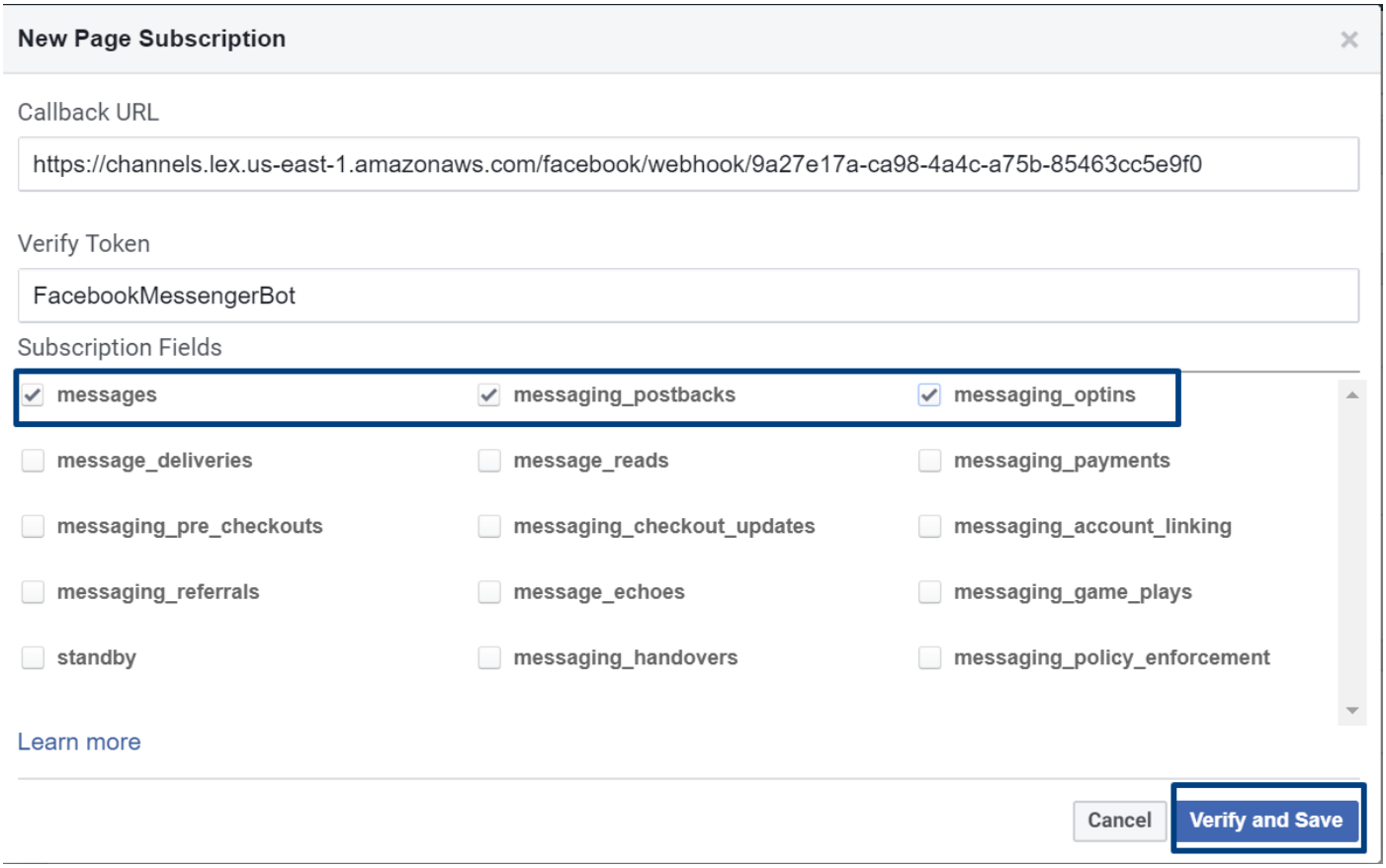
22. Once activated, you'll get the Callback URL at the bottom itself. Copy that URL and save it.



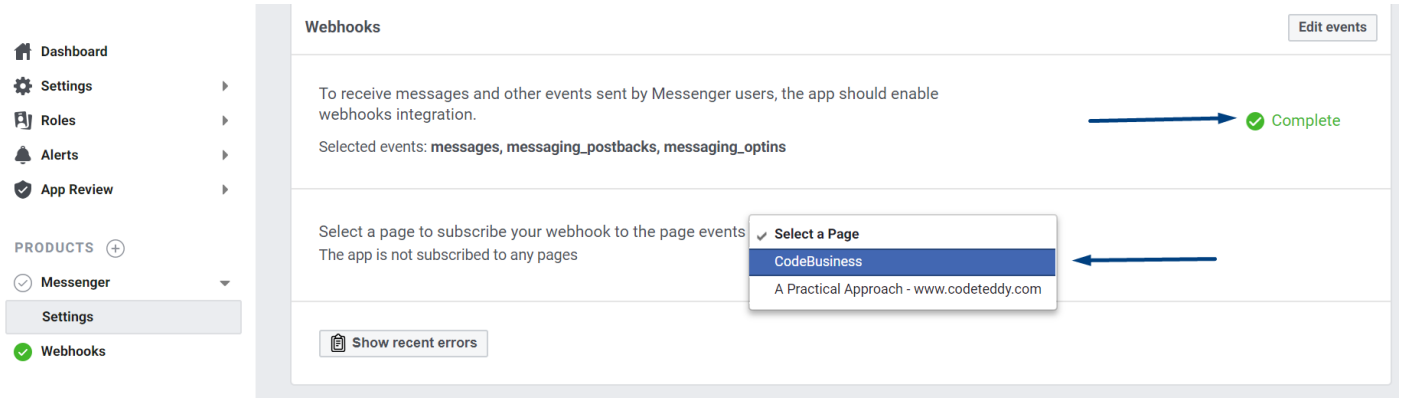
23. Again, go back to Facebook's developer's account and click Settings. Go to Webhooks section and click Setup Webhooks to provide the webhook to that copied call-back URL in the last step.



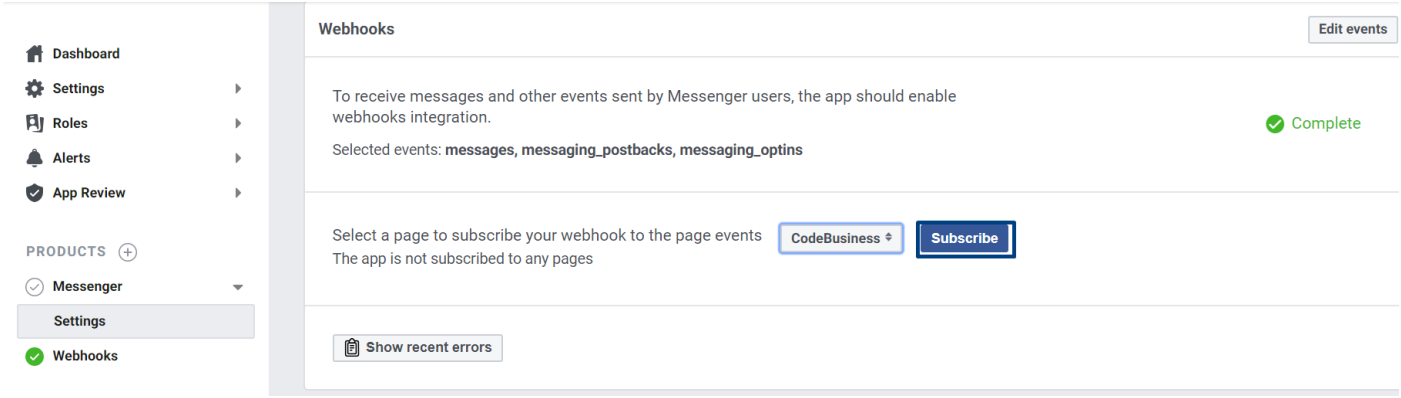
24. In the New Page Subscription i.e. the page opened after clicking setup webhooks, provide the Callback URL and verify token name that we stored earlier. Check the options like messages, postbacks, and optins as shown below. Click Verify and Save.



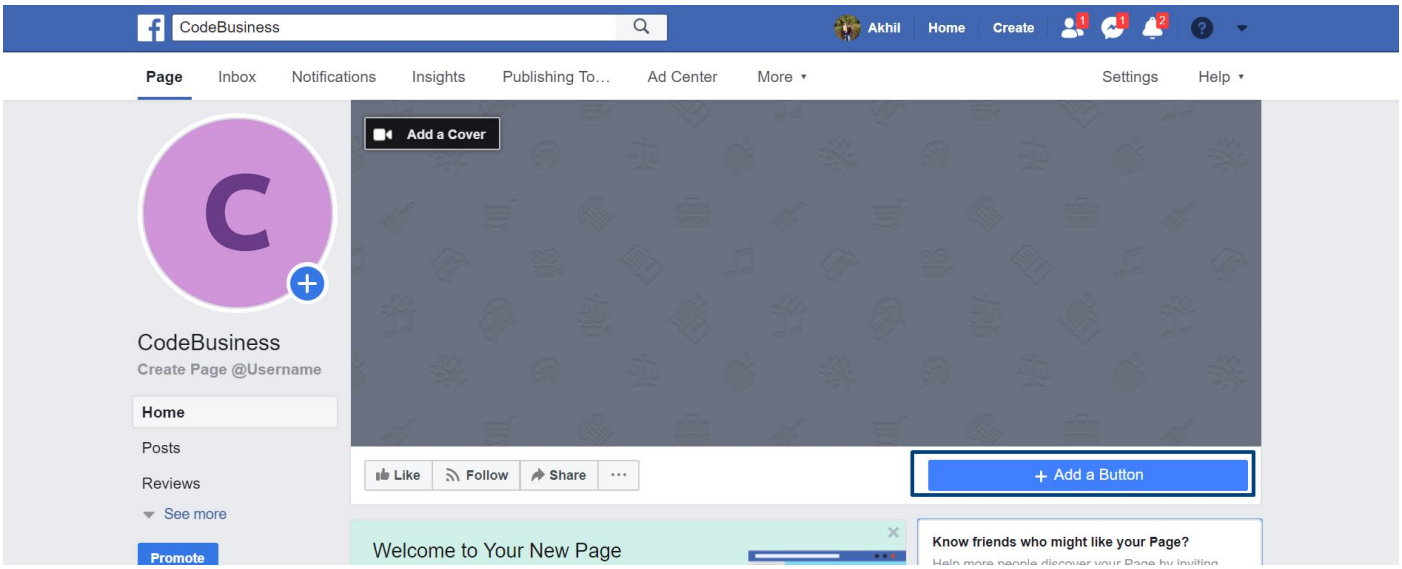
25. Once the webhook setup is complete, select the page to subscribe to webhook events. So select your Facebook page there.



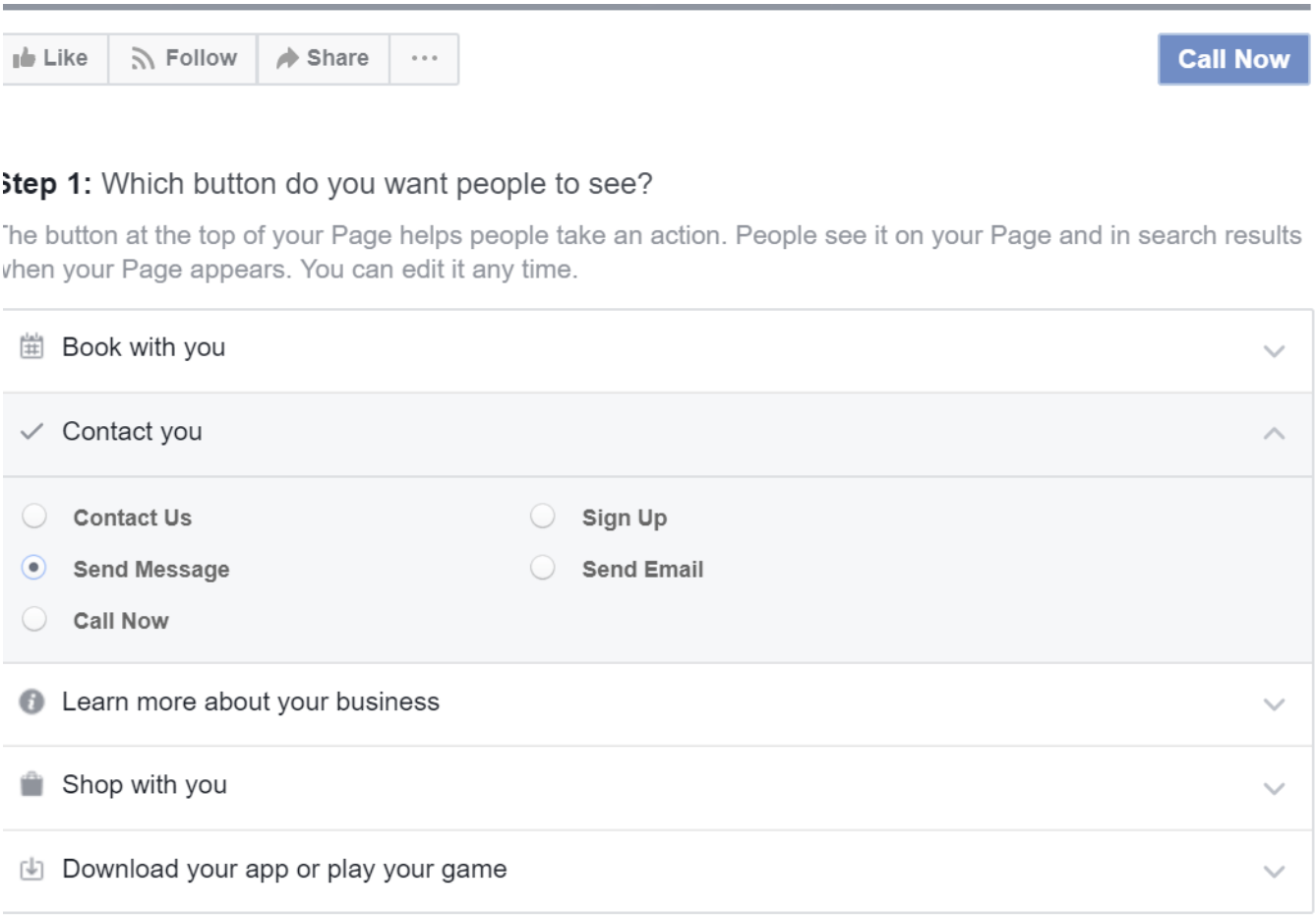
26. Click on the “Subscribe” button.



27. Once done, go back to the Facebook page and add a new button by clicking “Add a button” button as shown below.



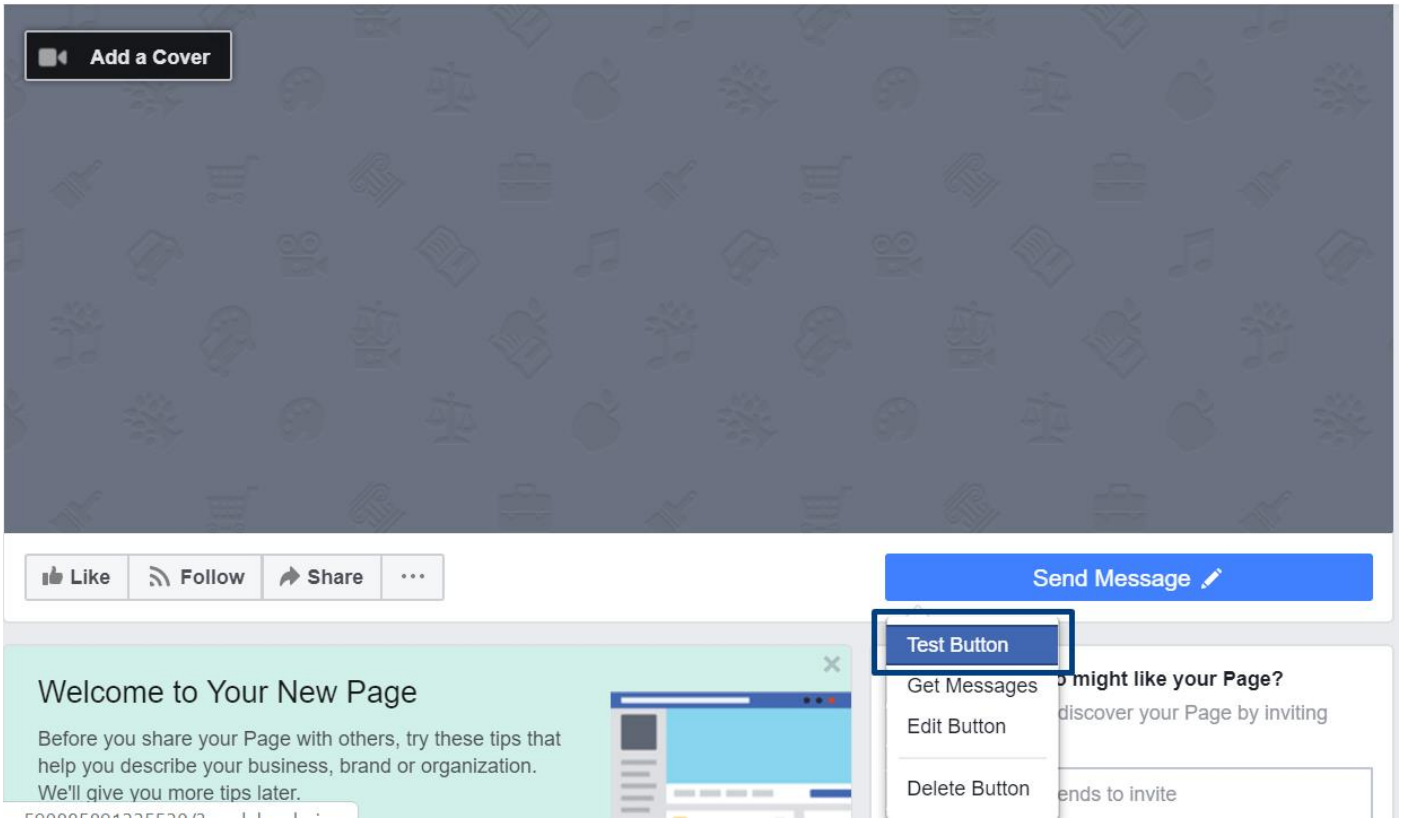
28. Provide details for all the steps i.e. selecting Contact you feature for my messenger type. Select the option to send messages.



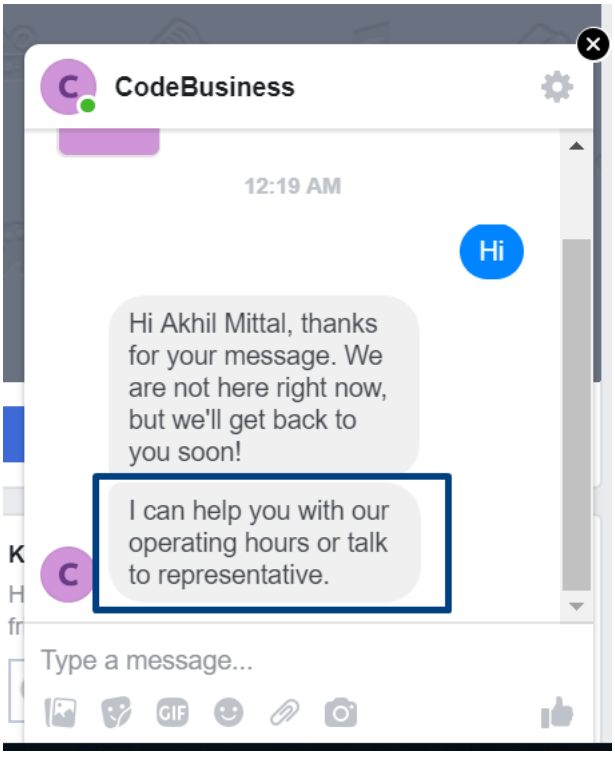
Live Chatbot on Facebook

After performing all the steps in the last section, we can say that our bot is successfully deployed and integrated with our Facebook page. Time to test it.

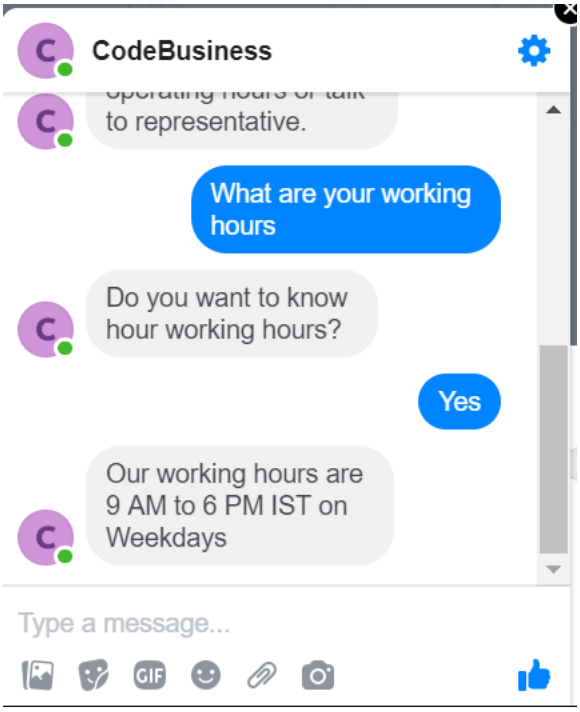
1. Click on newly added “Send Message” button and from the context menu click “Test Button”.



2. As soon as the Test Button link is clicked, the chatbot will open and note that the name of the chatbot is CodeBusiness. You type Hi, and it shows the message we provided in the error handling section.



3. You type the utterances in the first intent and it replies back with what we tested in the Bot Service.



Yippie. Our bot seems to be working well. Like-wise you can test everything that we tested on the BotService on Lex like talking to a human, seeing slot information and all. You can also provide the tester permissions to the user you want to see this bot when they see your page on Facebook. I find it very cool.

Conclusion

In this detailed article, we learned what a chatbot is, how to create your own bot using Amazon's Lex service. Deploy and integrate the bot with the Facebook messenger. The bot integration could also be done to other platforms like Slack, Twilio etc. Just a matter of exploring and playing with it.