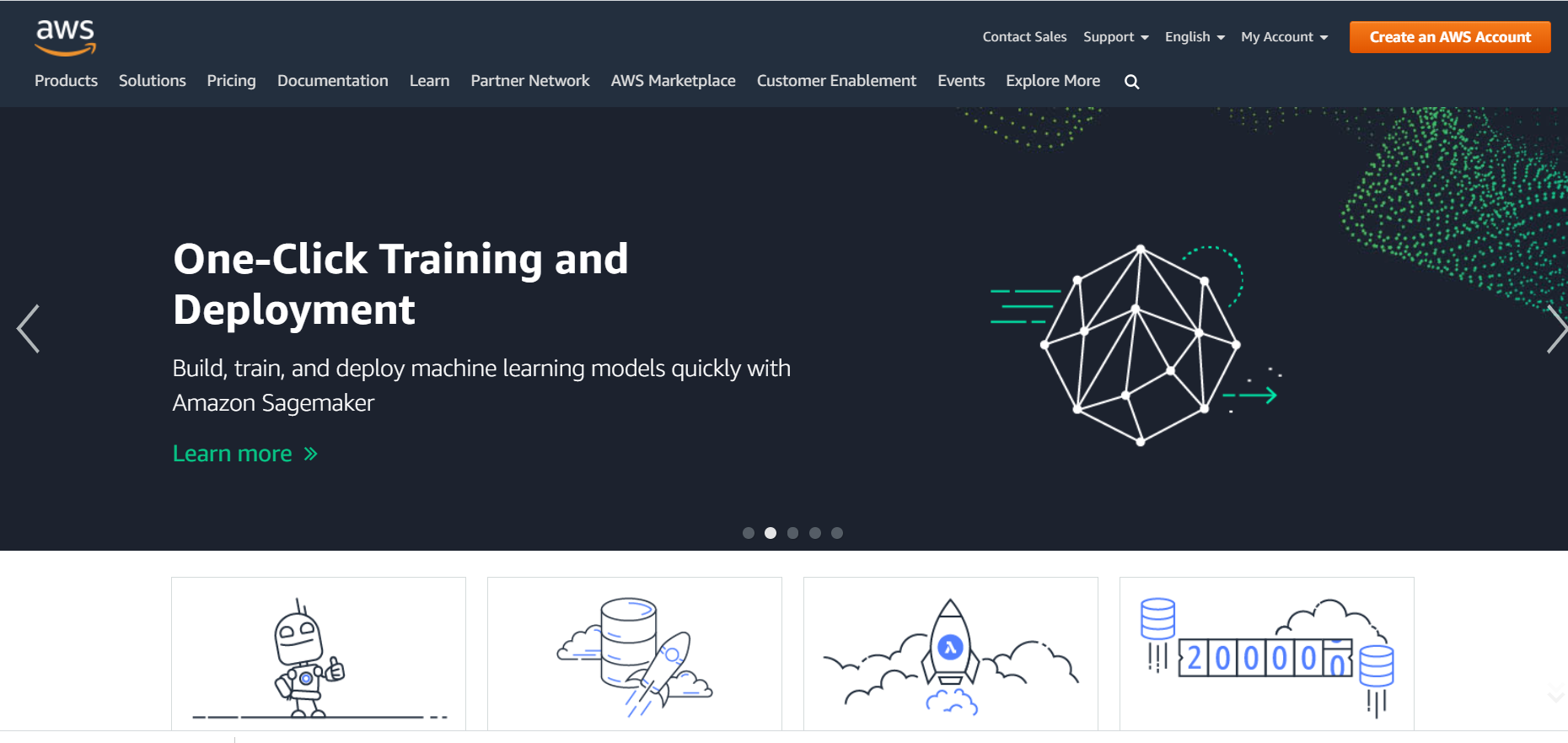
Building A ChatBot

With

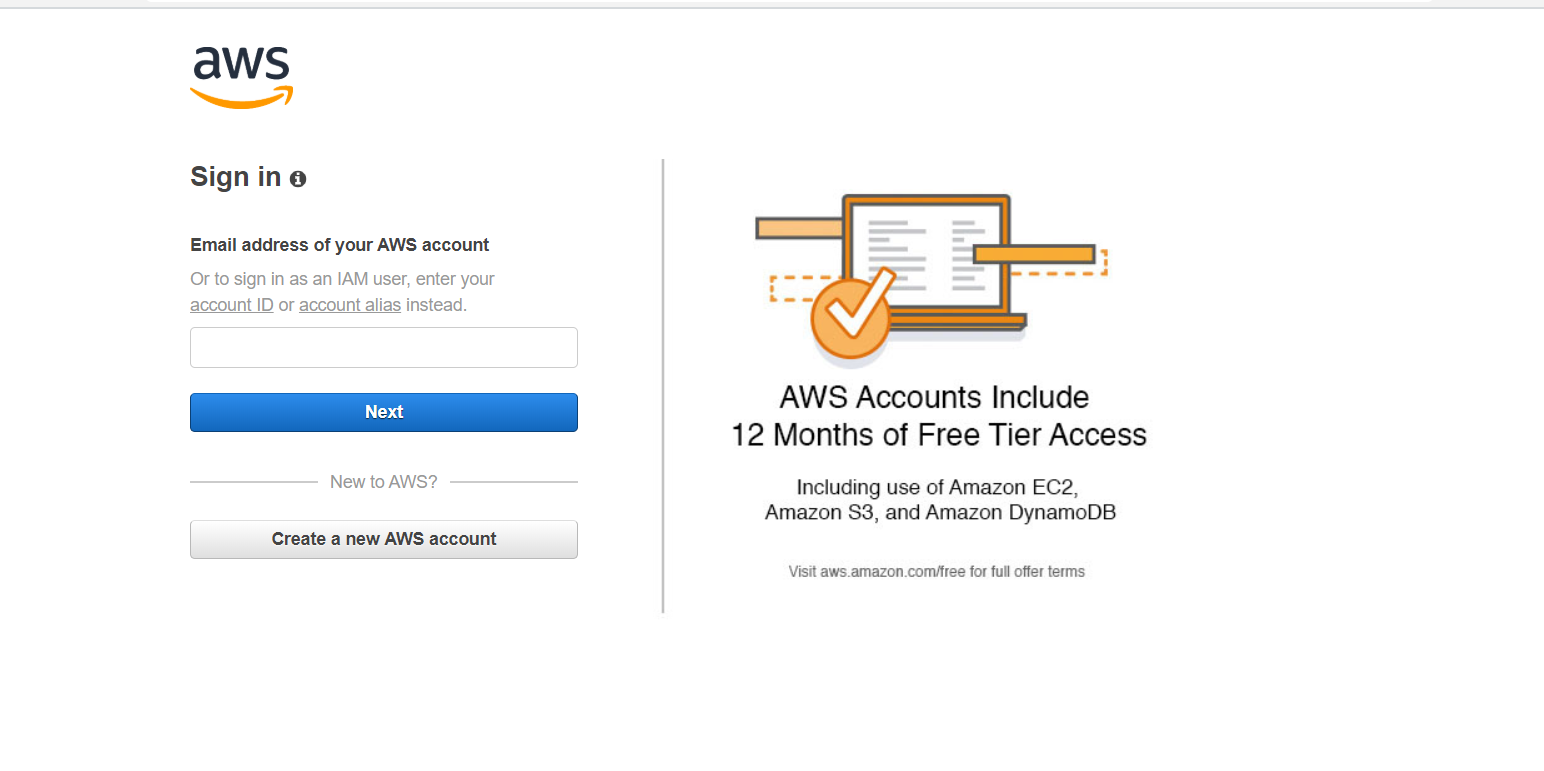
Amazon Lex

**Build your first chatbot**

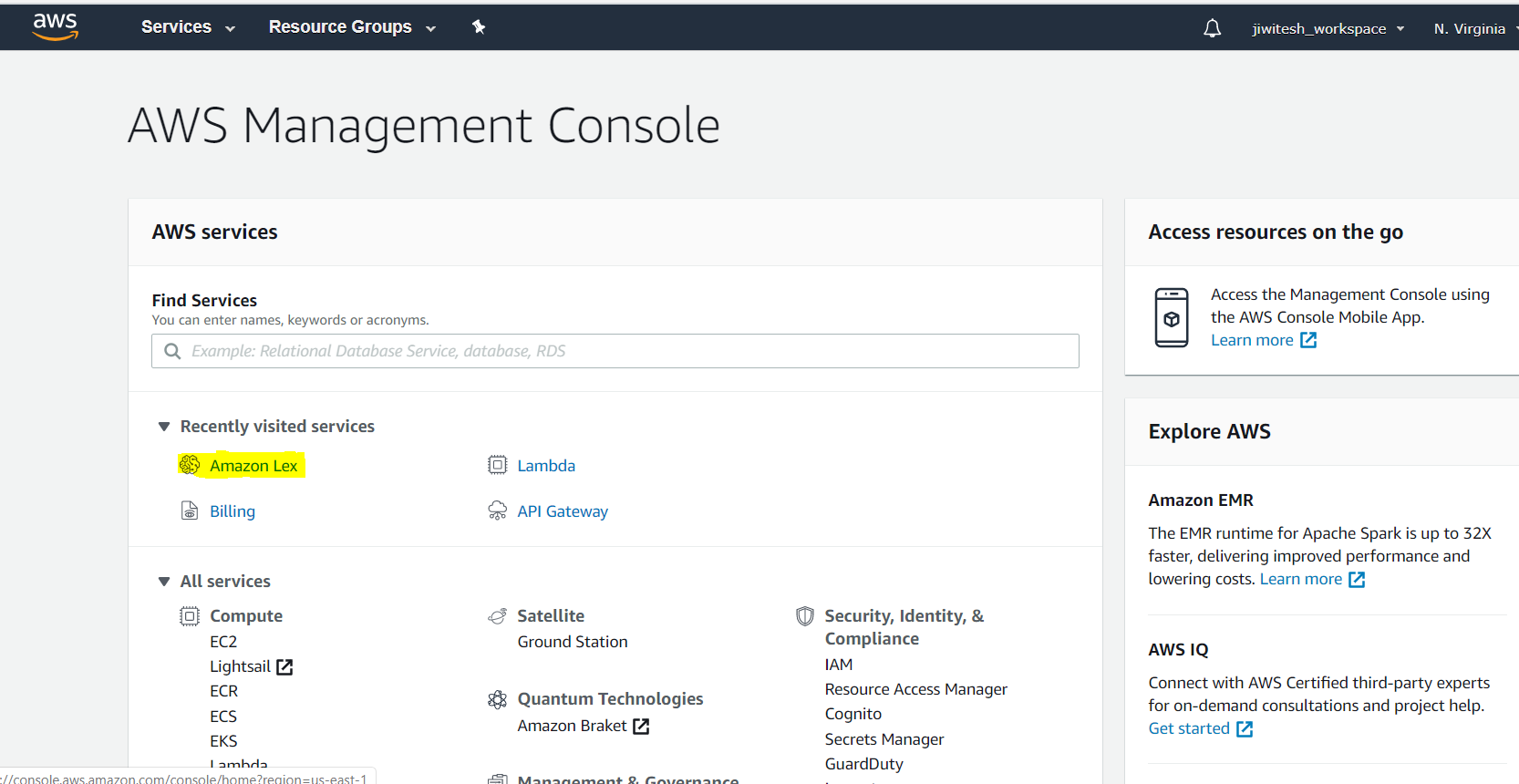
1) Login to AWS portal [**https://signin.aws.amazon.com/**](https://signin.aws.amazon.com/)



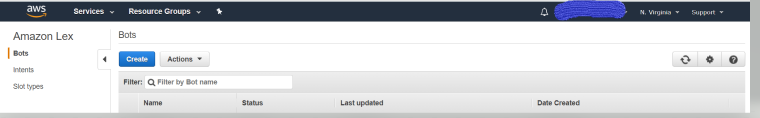
2) Click on “create an AWS Account”. Sign in with your credentials or create a new account.



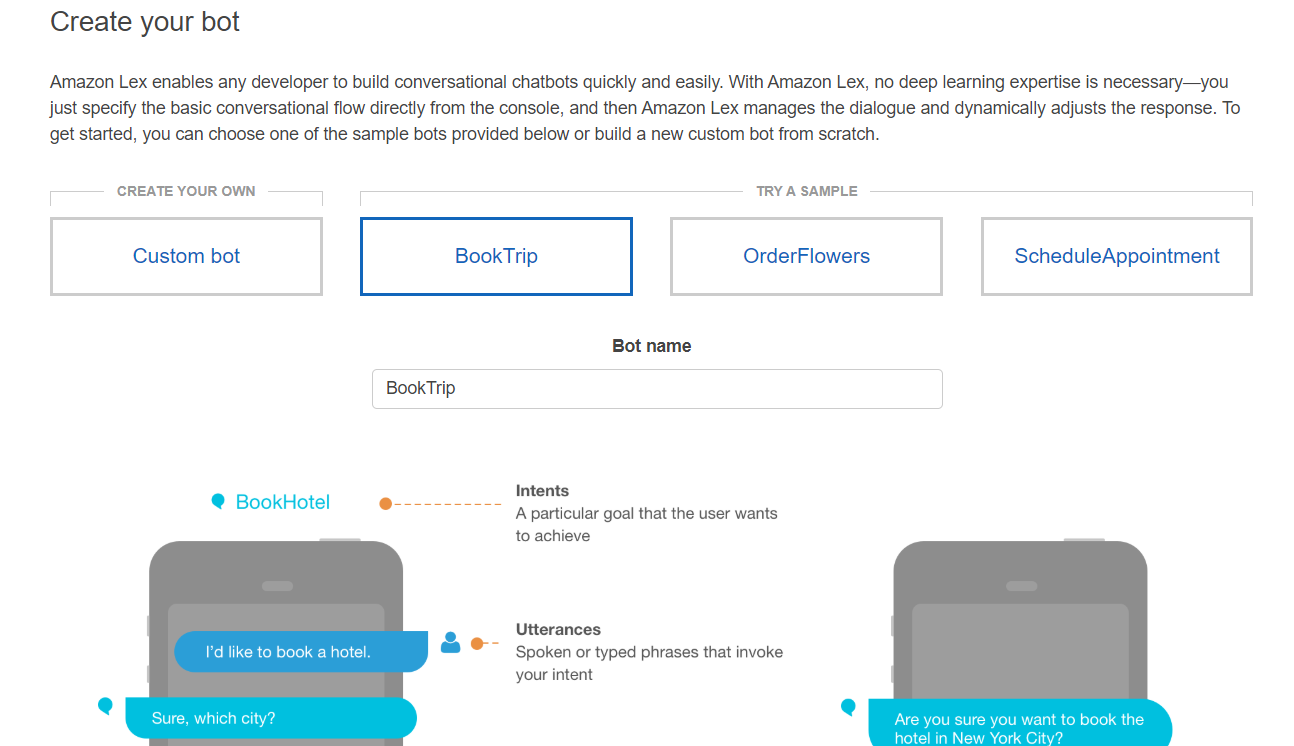
3) Once logged in, click on Amazon Lex to start creating your bot.



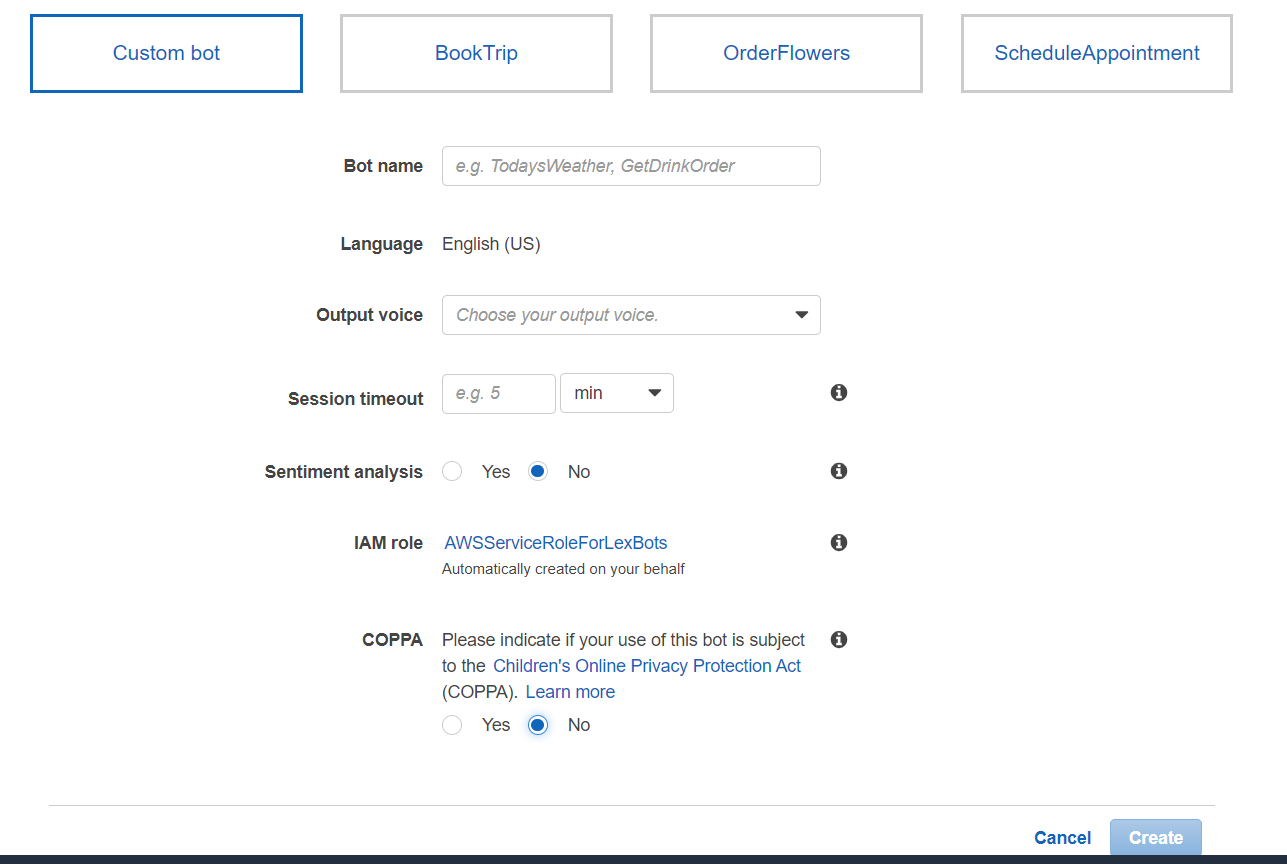
4) Click on create and create a new bot.



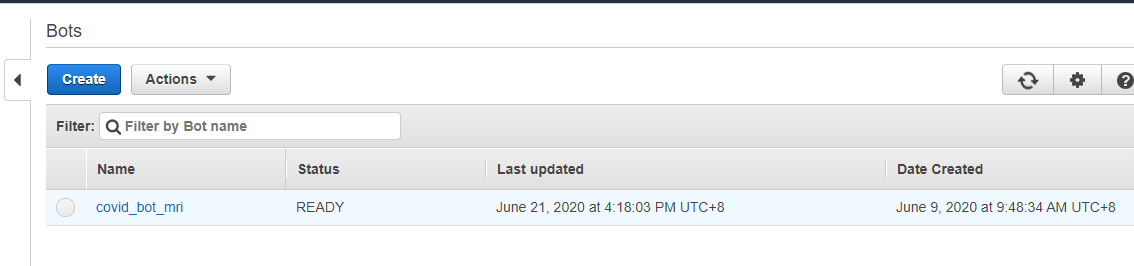
5) Once you click the create button, you will see a window like this. Click on the “custom bot” to make your own bot. The other bots are example bots and you can go through them to get a glance on some higher details.



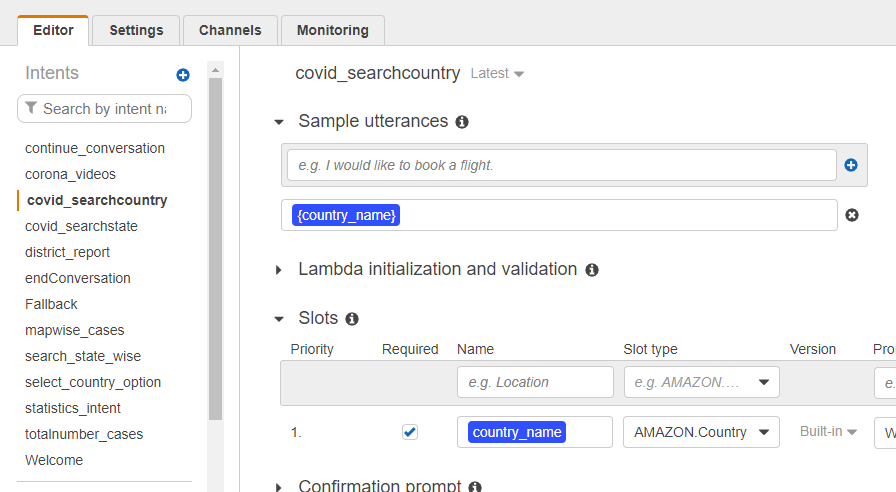
6) Fill up all the details and create on the create button.



7) We have created a bot as follows:

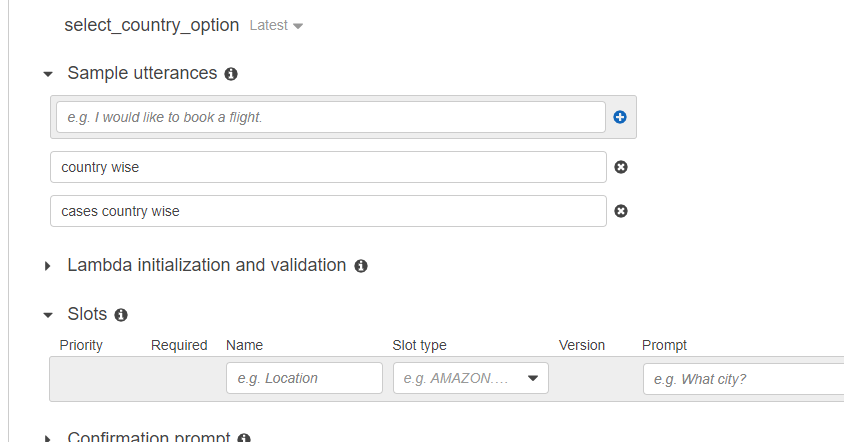


8) We will start by creating intents for our bot. Click on the plus button next to intents.



Give a name to your intent.

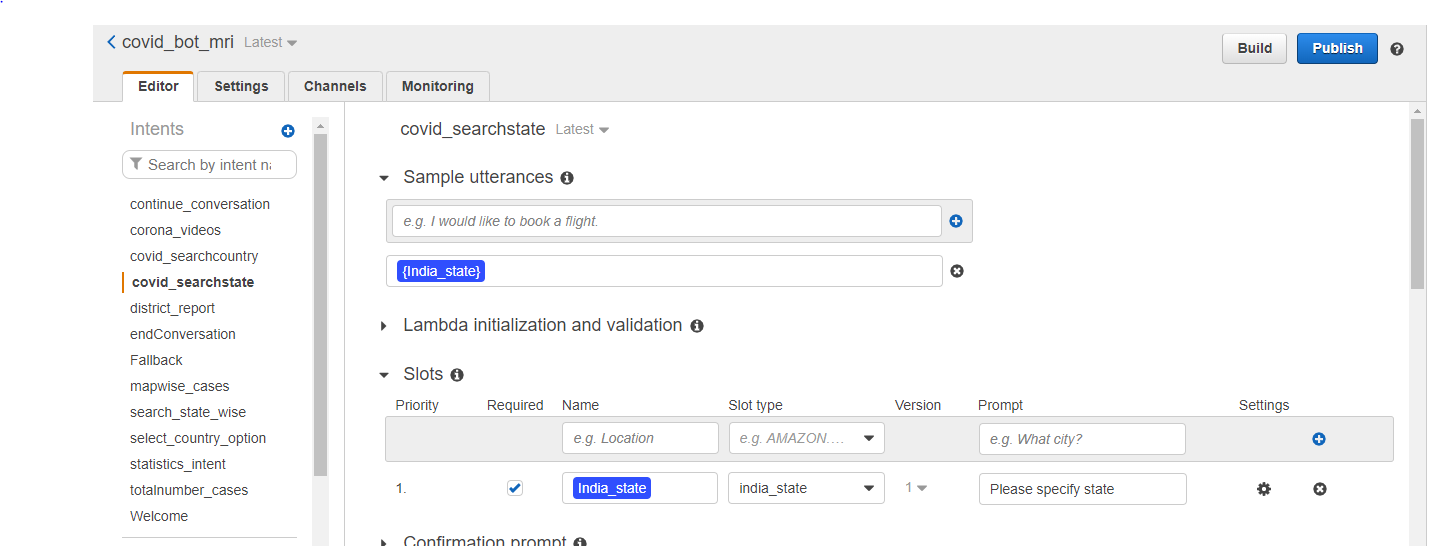
We are going to create a intent and defined utterances.



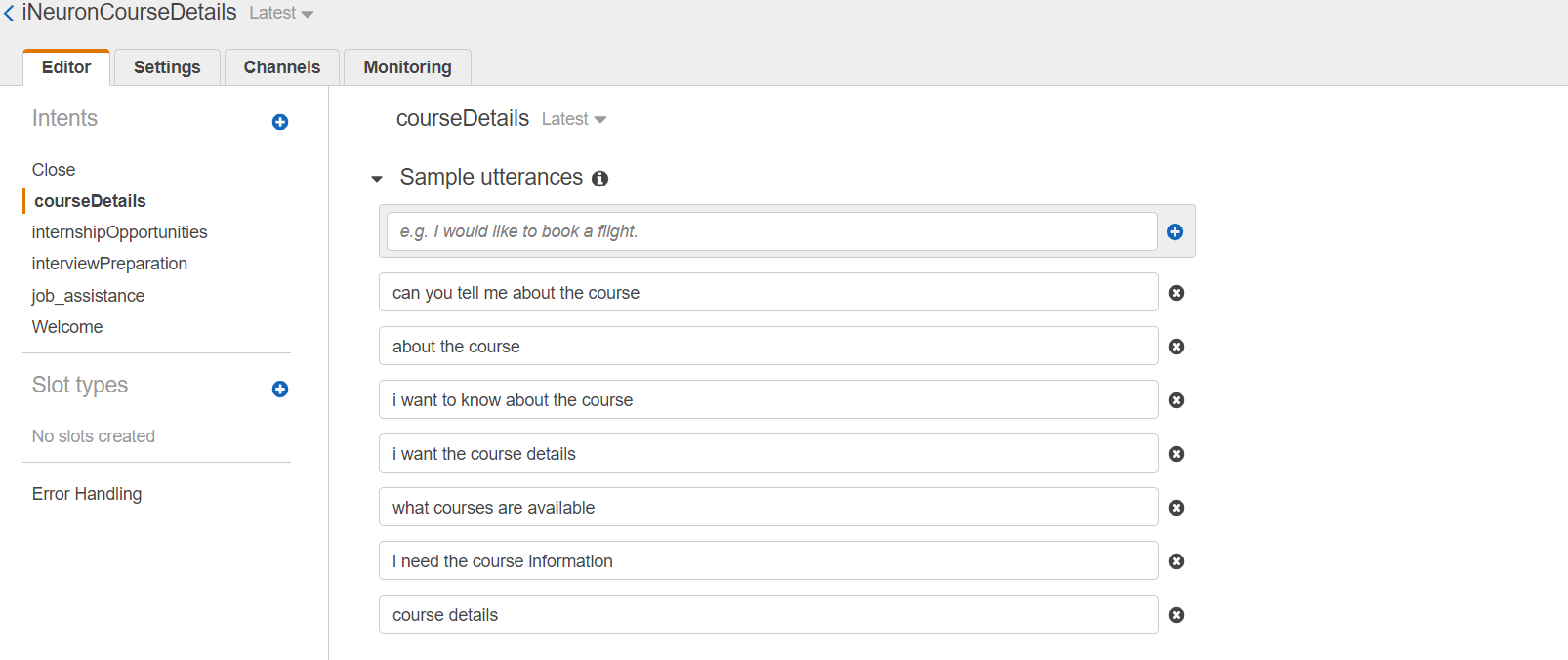
We are not filling the slots for this intent and keeping the “Fullfillment” as return parameters to client. We will see the use of slots and use of lambda function for the next intent.

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

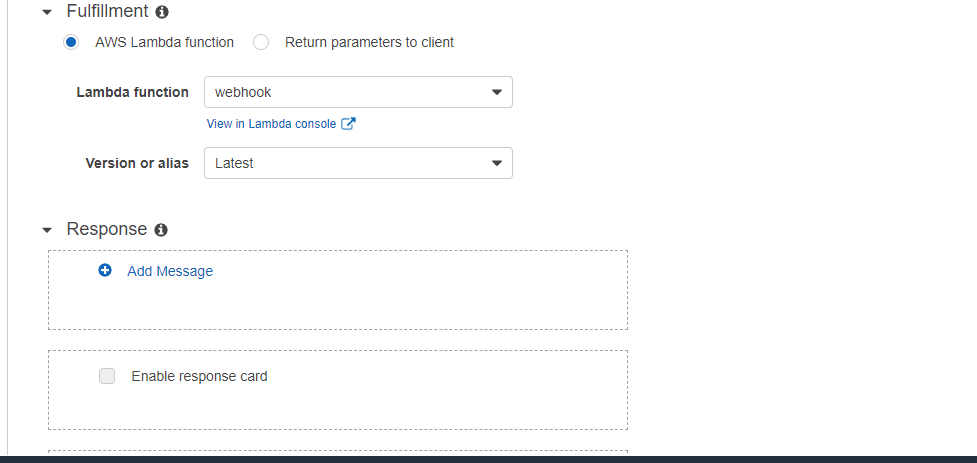
9) Similarly we are creating new intents as per our requirements that we need the bot to respond to.



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 covid case details and it will display the report India state and all the countries



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.

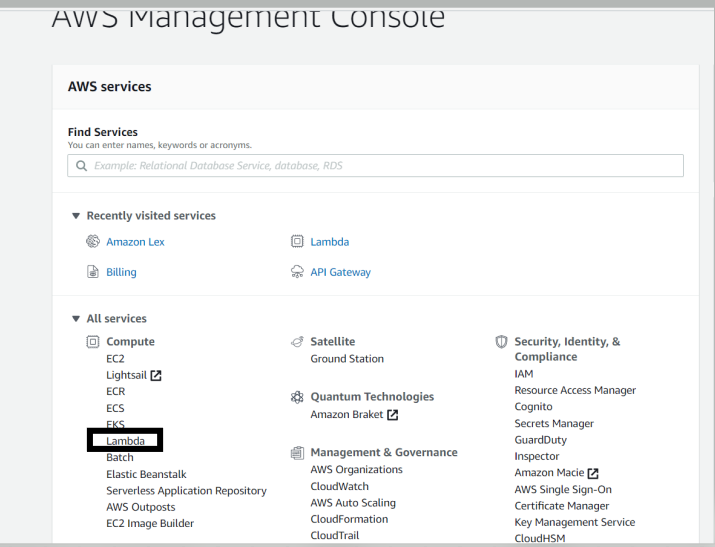


In the slot name we give the values that we want from the user. The slot\_type is the default validation parameter in Lex and we can use that to validate the values user gives against that slot. In the prompt we are going to prompt the user with the specific questions to get the slot values.

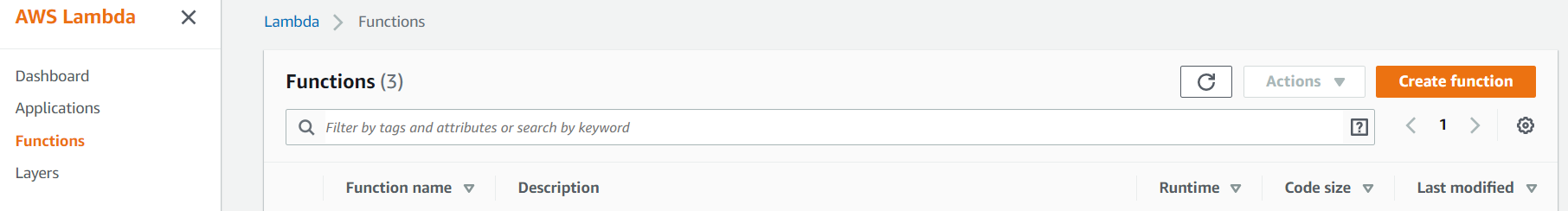
In the Fulfillment section, we are using Lambda function to send an email to the user on the given email id and to the support team with all the user details.

Let’s see how to create the lamda function.

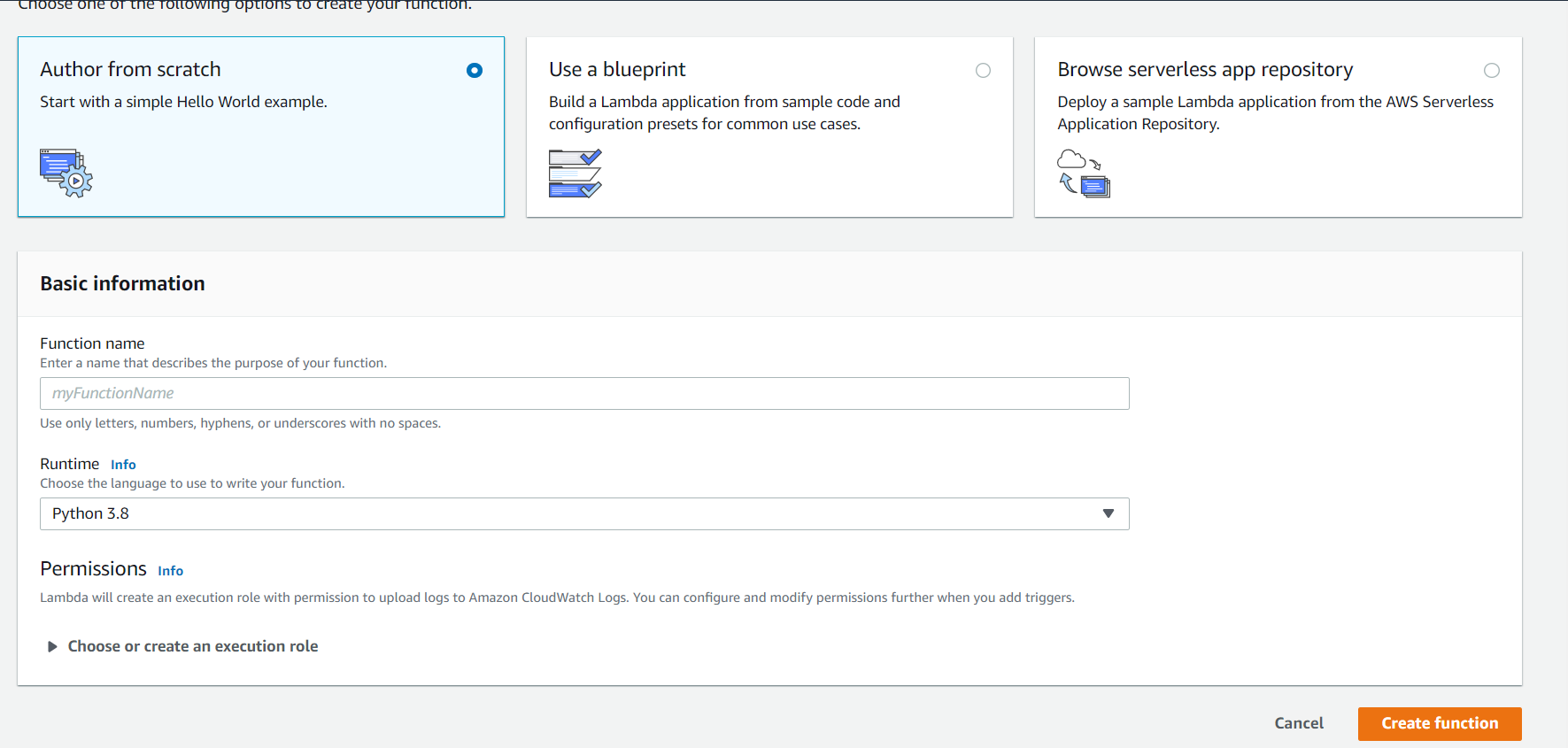
Go to the AWS console and click on “Lambda” inside the ”compute” section.



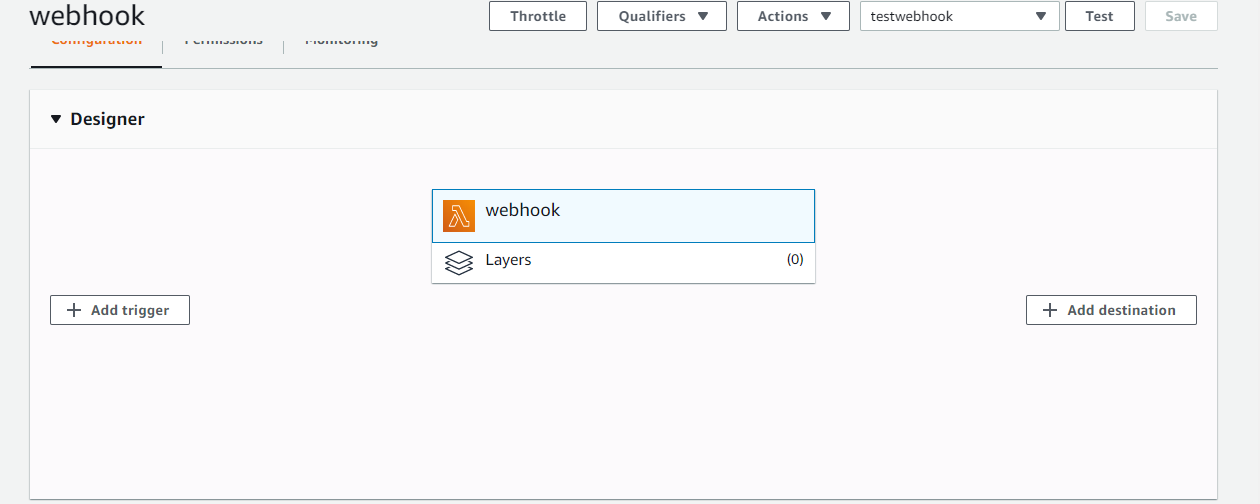
Click on the “create function” tab.



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 “webhook”. After clicking the “create function” button, you will see a page like this :



Go to the “Function code” section:

import json

import urllib3

http = urllib3.PoolManager()

def webhook(name1, curr\_slots):

data = {"Intent": name1, "curr\_slots" : curr\_slots}

encoded\_data = json.dumps(data).encode('utf-8')

response = http.request('POST',"http://1596fdf1ac92.ngrok.io/webhook", headers={'Content-Type': 'application/json'}, body = encoded\_data)

return response.data.decode('utf-8')

def lambda\_handler(event, context):

name = event['currentIntent']['name']

curr\_slots = event['currentIntent']['slots']

response = webhook(name, curr\_slots)

response = {

"dialogAction": {

"type": "Close",

"fulfillmentState": "Fulfilled",

"message" : {

"contentType" : "SSML",

"content" : response

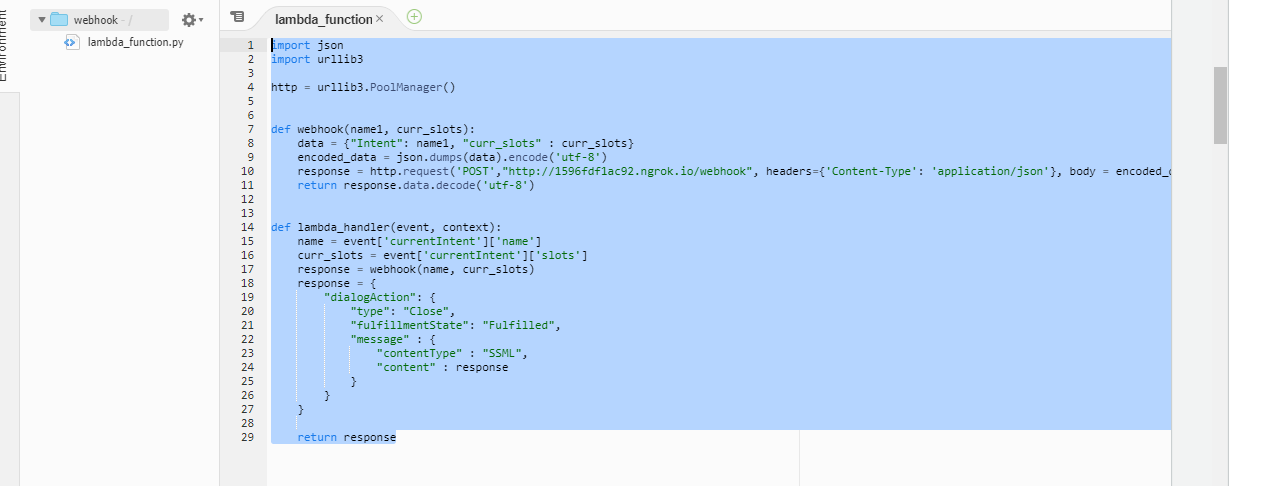
}

}

}

return response

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 :

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 “webhook” function. Let’s see the code written for creating the “Api” which is attached in the Github

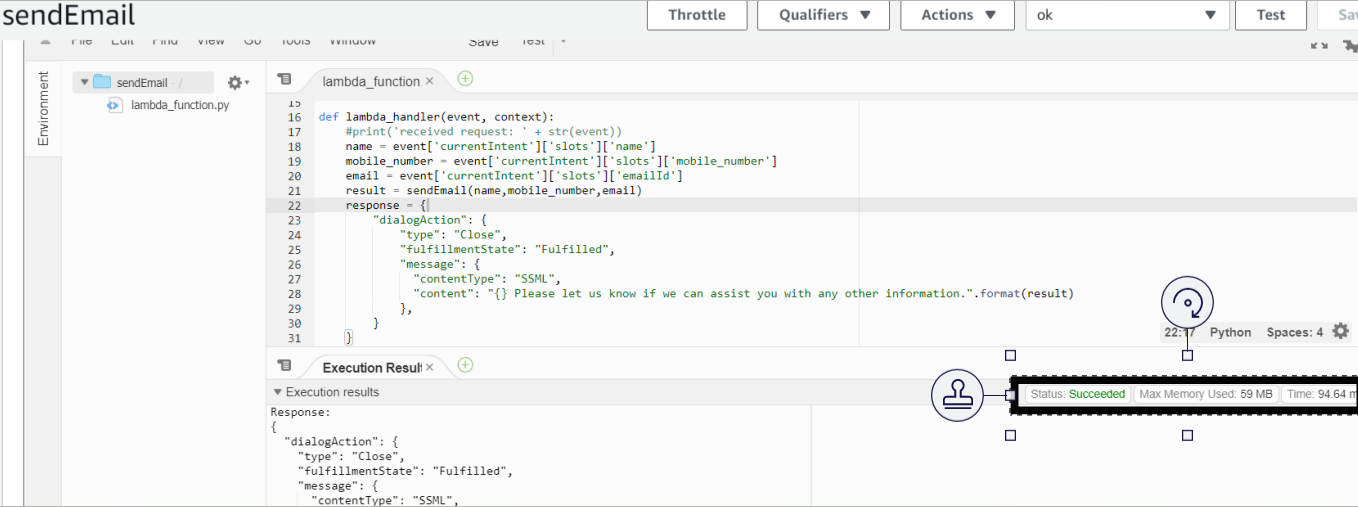
We are keeping the Lambda function as simple as possible and we will not add any more functionality. Let’s test our function written in the lambda.

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:

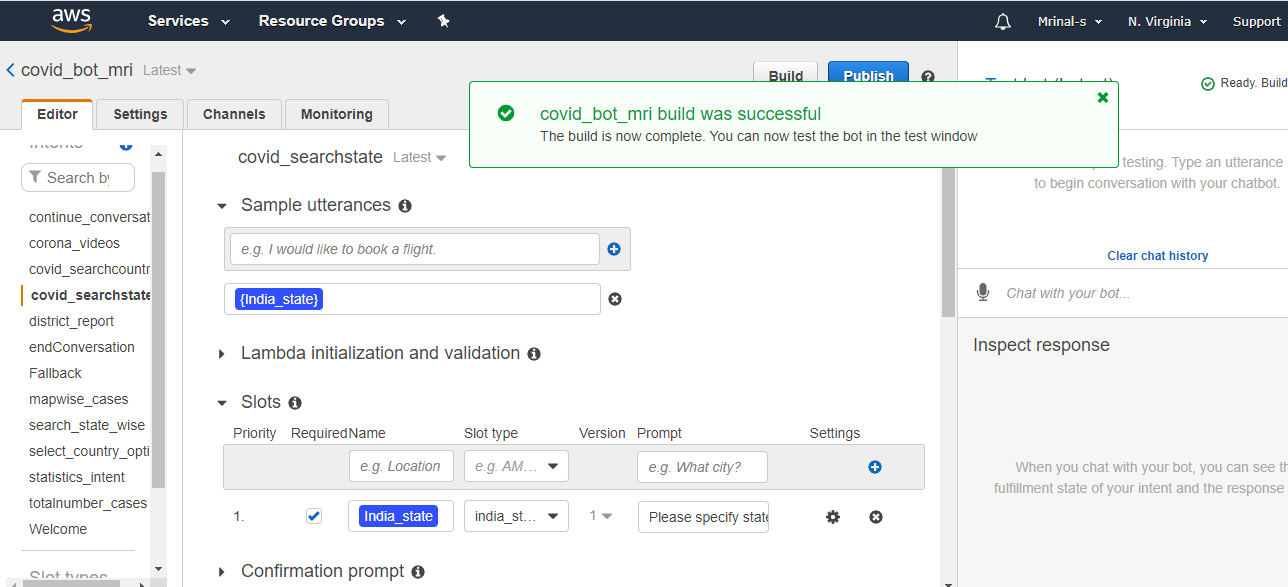


You will receive an error status if the function doesn’t work properly.

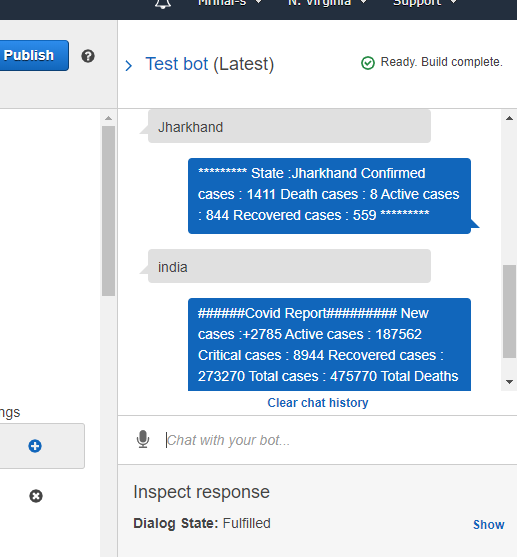
Now that we have created all of our intents. You can test your bot with the intents by “building” you bot.

Click on the “build” button to build the bot:

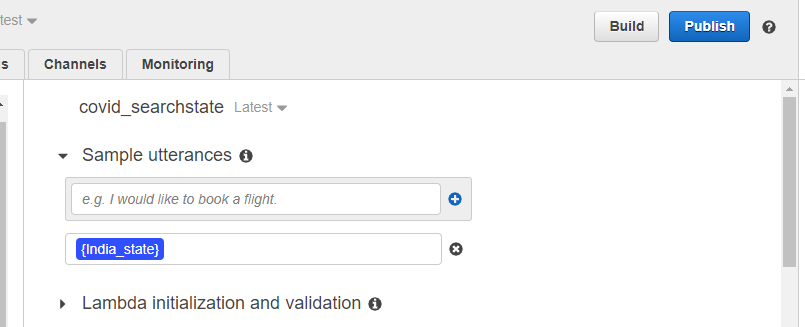
Wait till you get the message that you bot is ready for testing.



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.

We will talk about deployment after publishing the bot, we will use this channel tab then.