Alexa Document

Create account in

1. Amazon developer account and
2. AWS account with same credentials

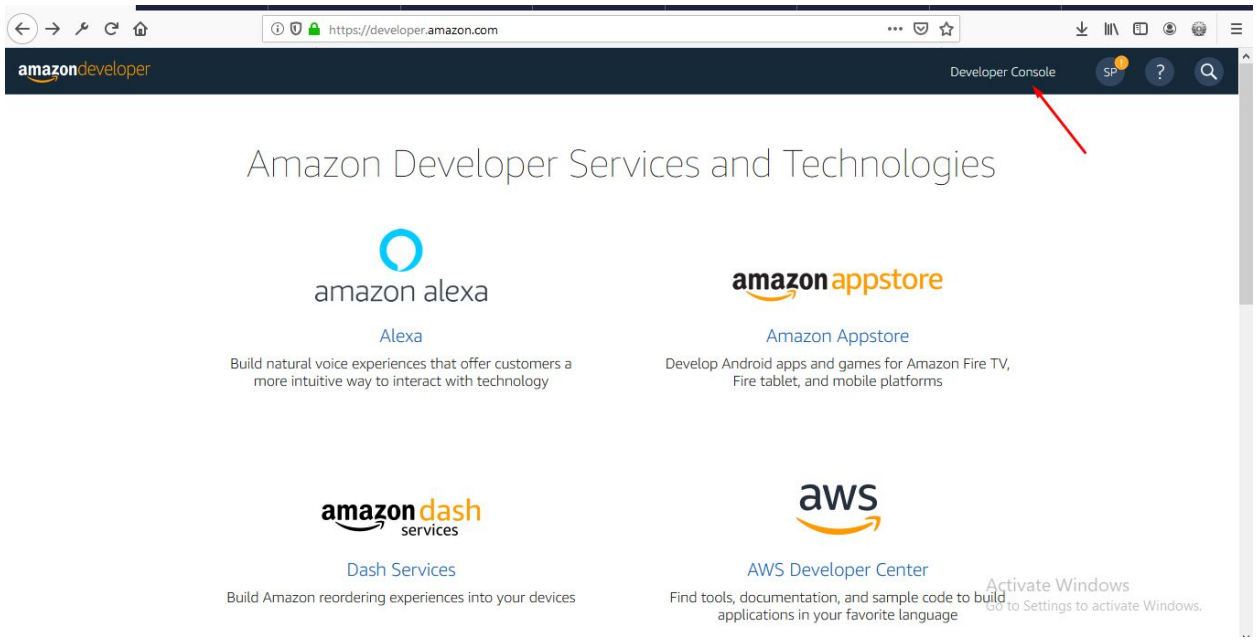
Alexa Skill :->

1) https://developer.amazon.com/docs/custom-skills/steps-to-build-a-custom-skill.html
2) https://developer.amazon.com/docs/devconsole/build-your-skill.html#custom-model

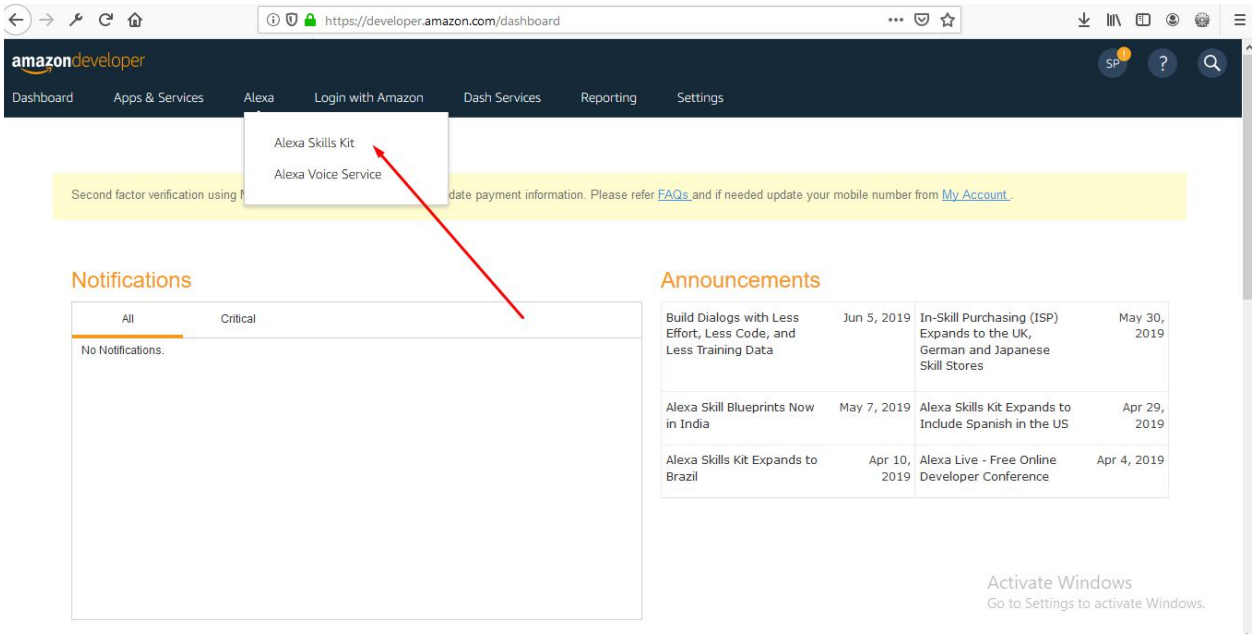
Lambda :->

1) https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html

1. Login to amazon.developer https://developer.amazon.com/
   Go to Developer Console



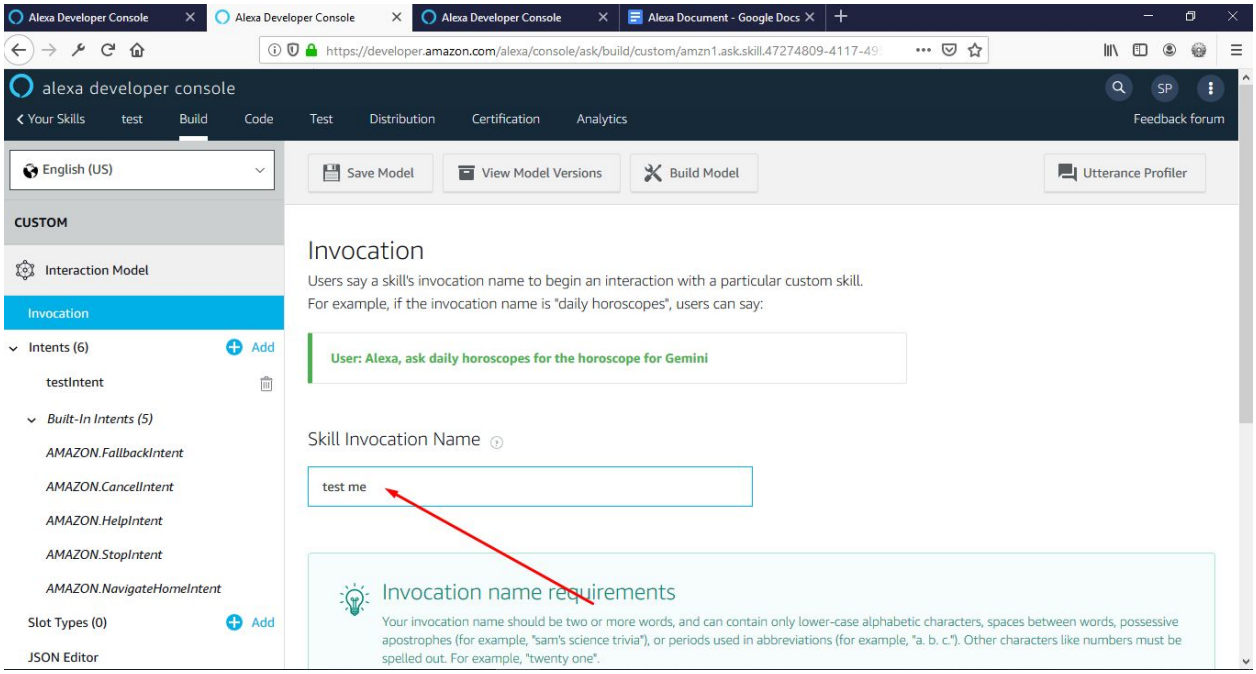
2. Then click Alexa->Alexa Skills Kit



1. Click **Create Skill.**
2. Enter the skill name and default language.
3. Click the model you want to include.
   (Choose Custom Model)
4. Choose a method to host your skill's backend Resources
   (Choose Provision your own)
5. Choose Template
   (Start from scratch)
   Click **Choose**

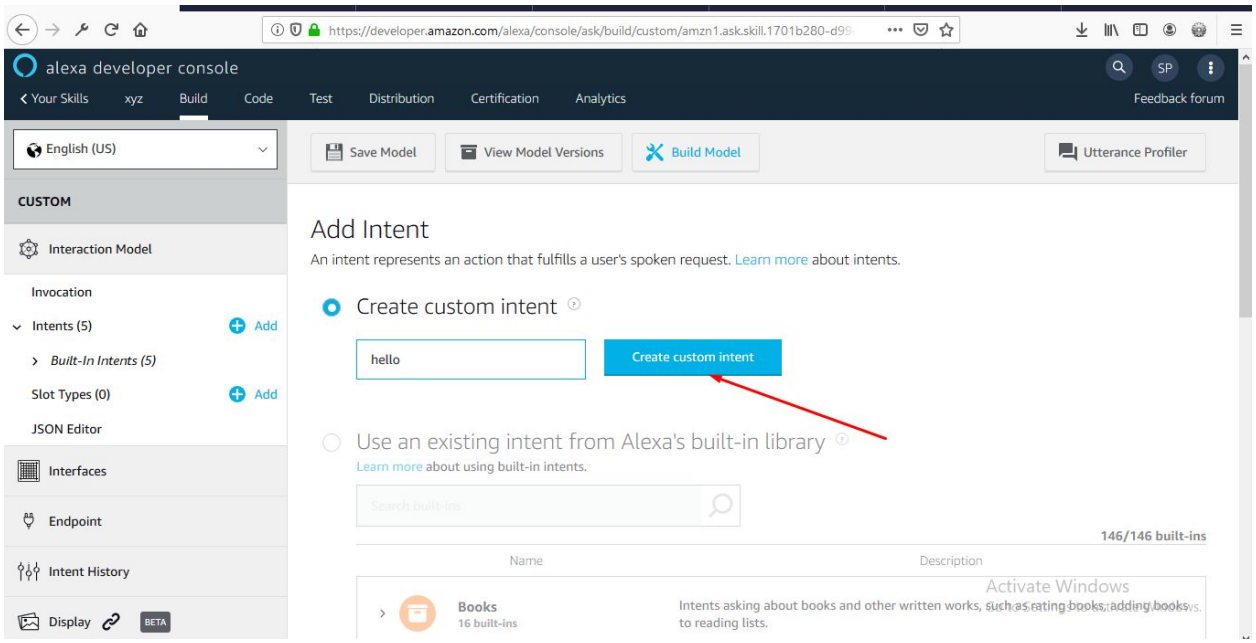
3. Once the developer console has created your new skill, you can start to configure it on the **Build** page. The specific information you need to provide depends on the model you select.

1.Add Skill Invocation Name: test me

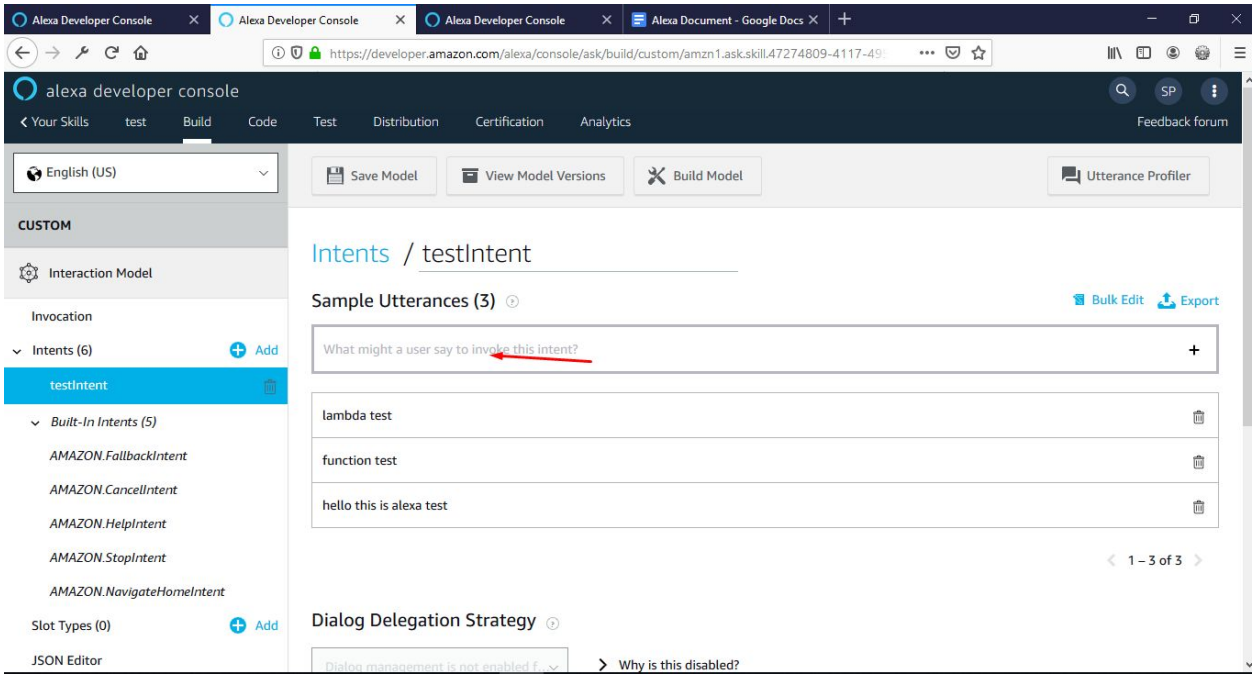


2. Click on intents form sidebar and Create Custom intent

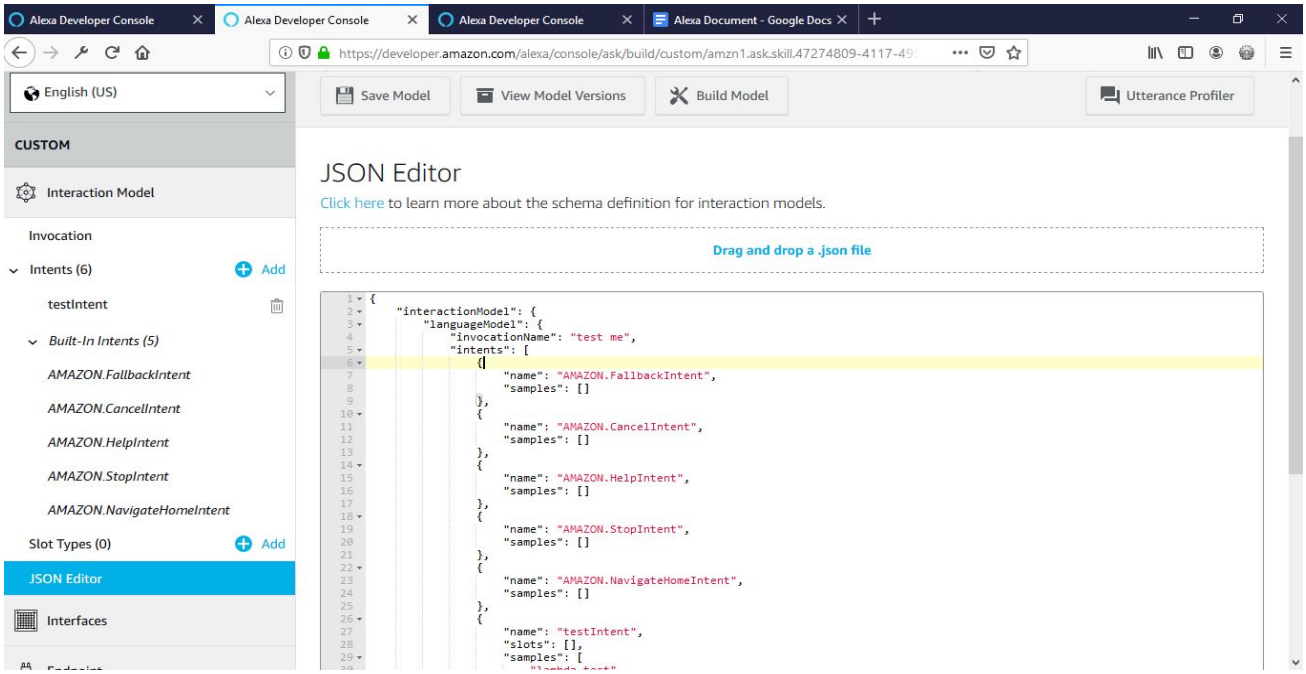
Intent->Add Intent(name:testIntent)



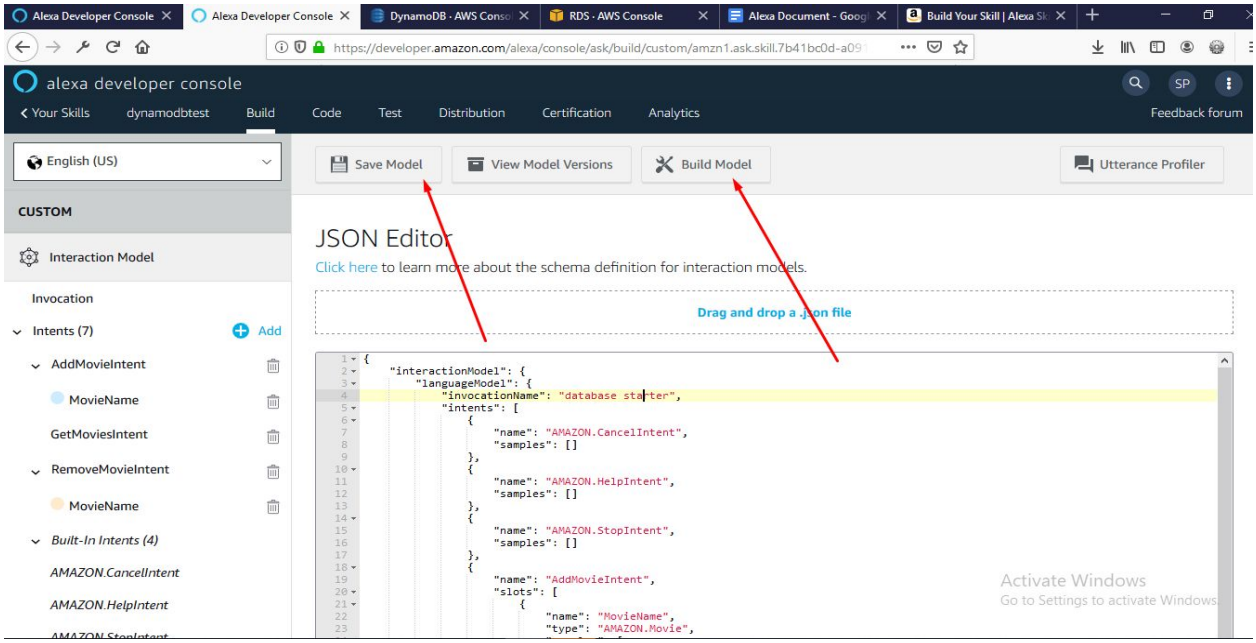
3. Add sample Utterances



You can see all intents in json editor and can change directly from here.

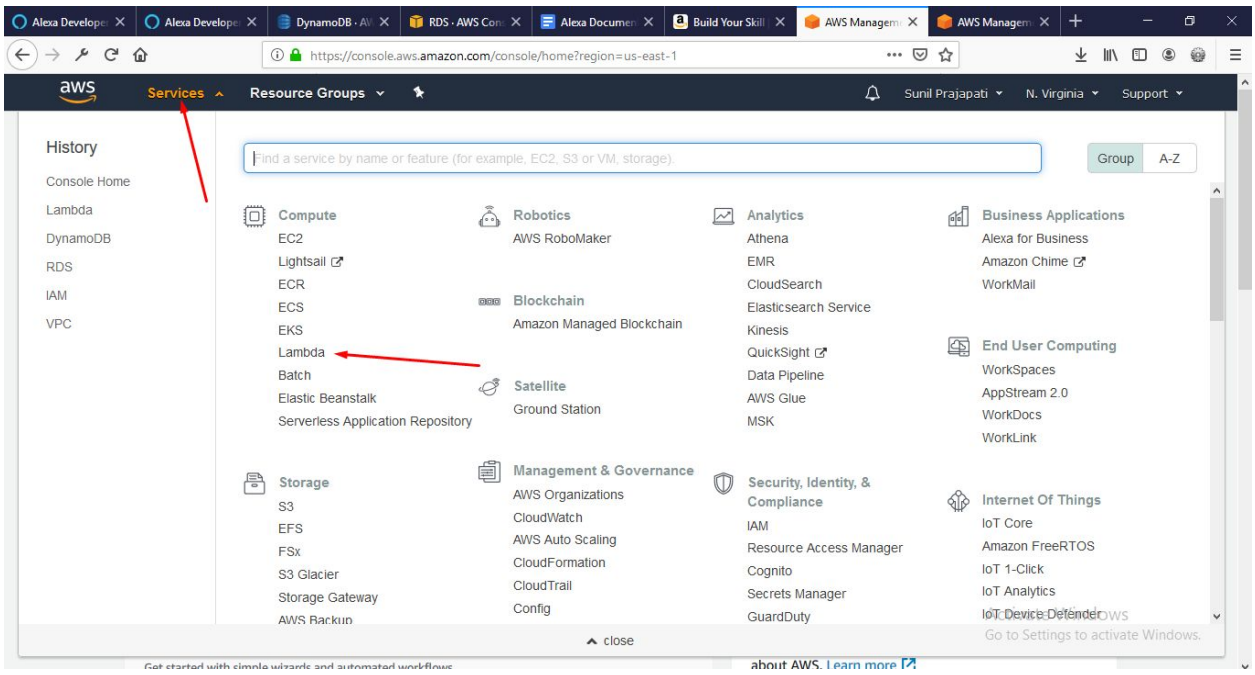


4. Click **Save Model** and then **Build Model**

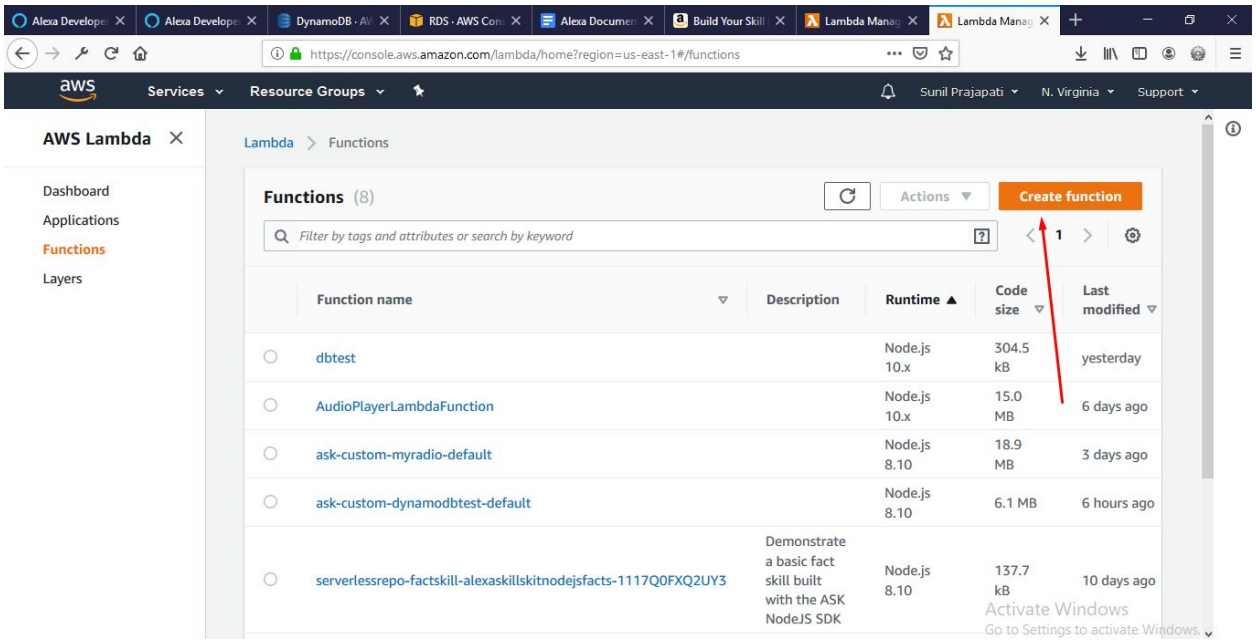


2. Login to https://console.aws.amazon.com/console/
On AWS Dashboard, Select *US East (N. Virginia)* region on top right – this is the only **region** with Alexa/Lambda free tier service.

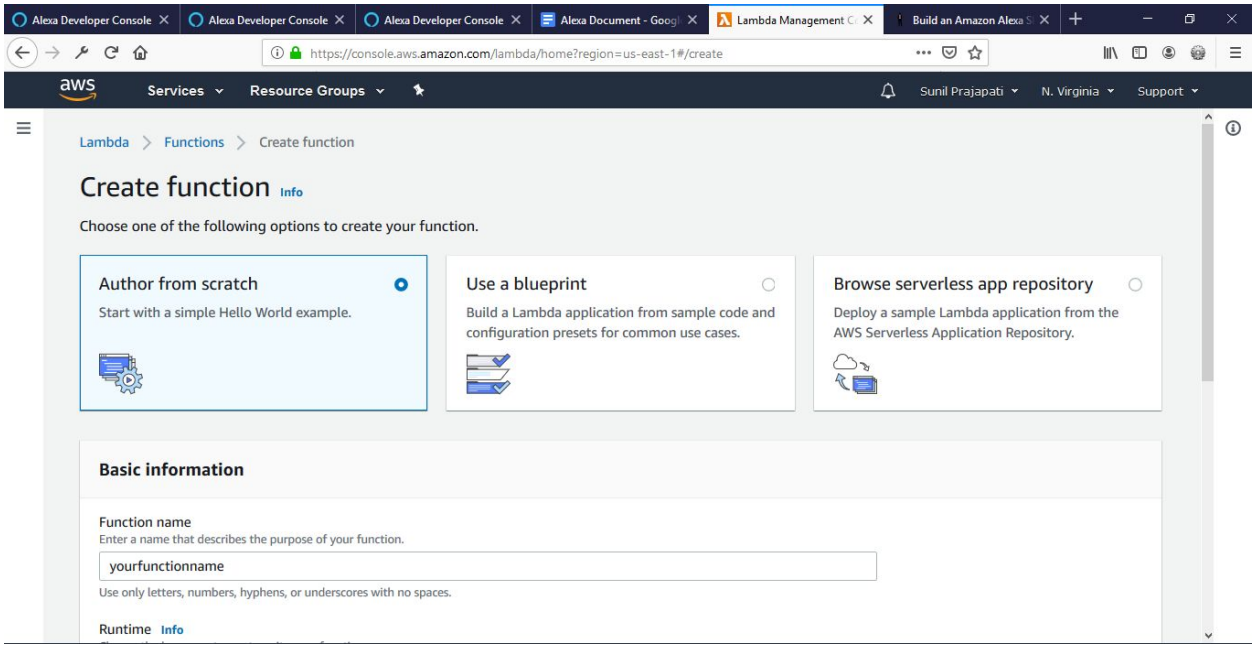
1. Go to Services then search for Lambda



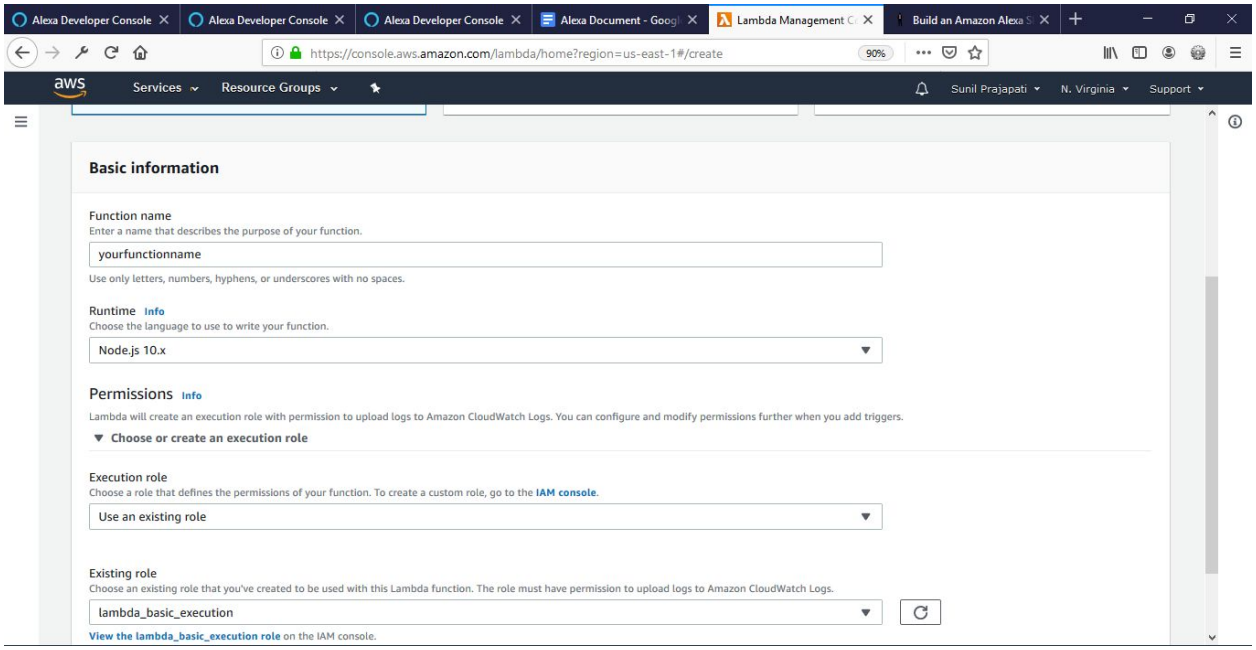
2.click **Create function**



3. Select **Author from scratch**



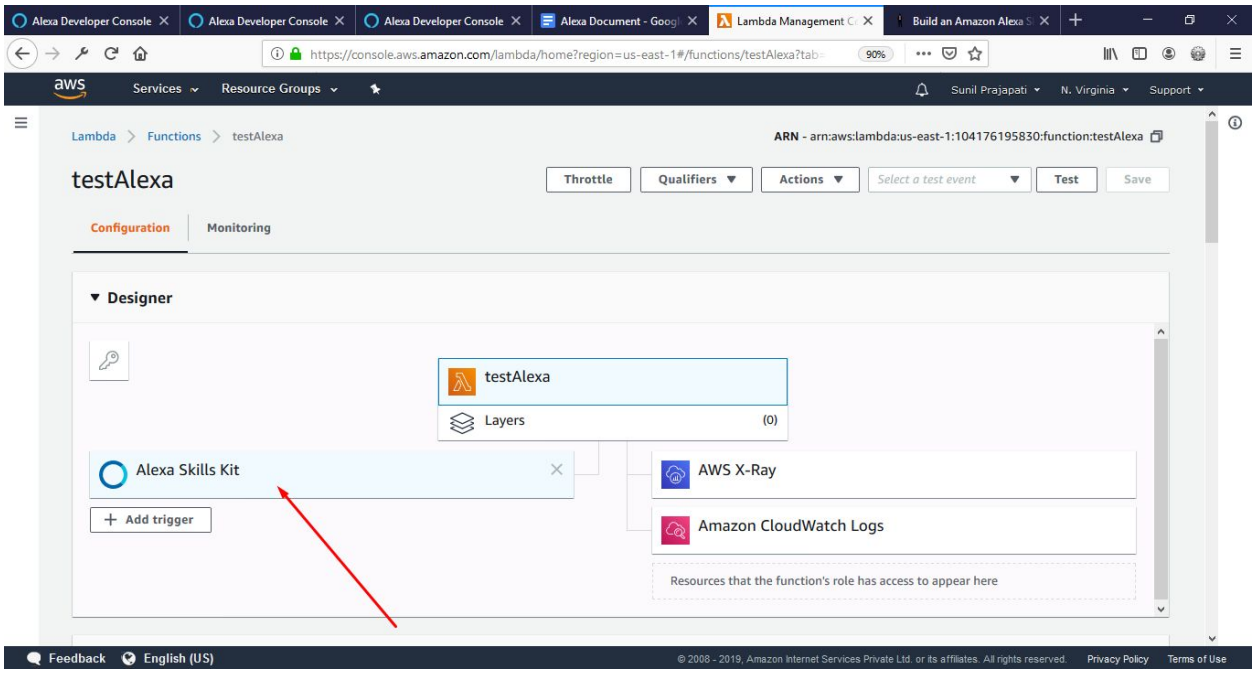
Basic information add below mentioned details and then **click** Create function



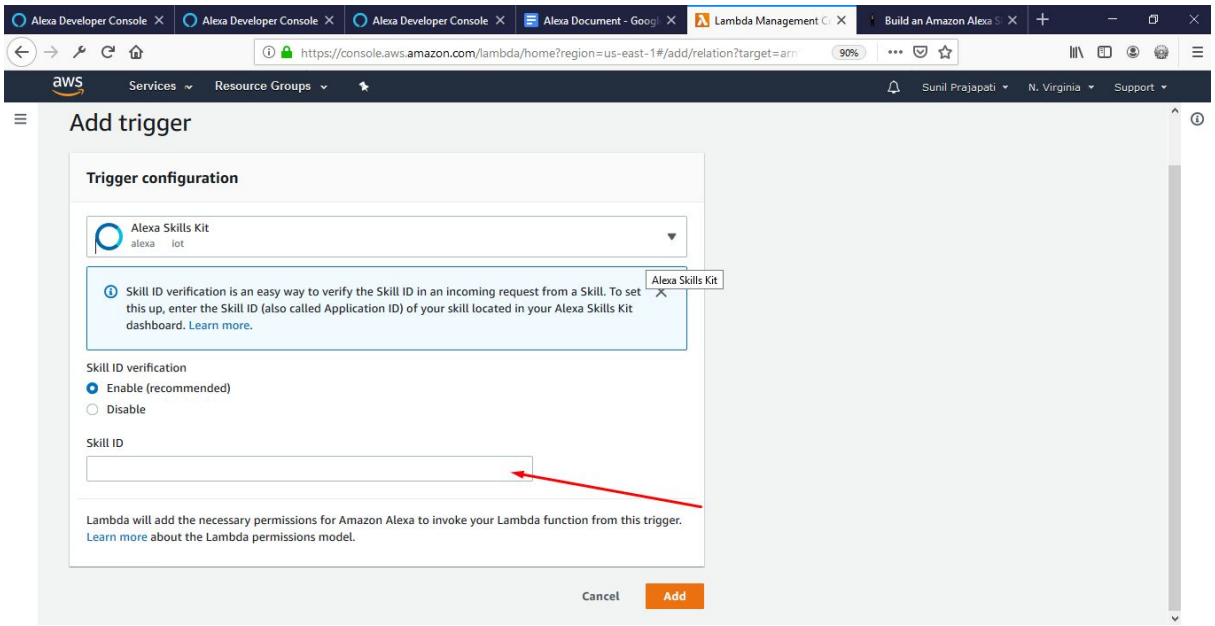
4. Click Add trigger

Choose Alexa Skills kit

Note the **ARN** on the top right *(We need this ARN information later)*



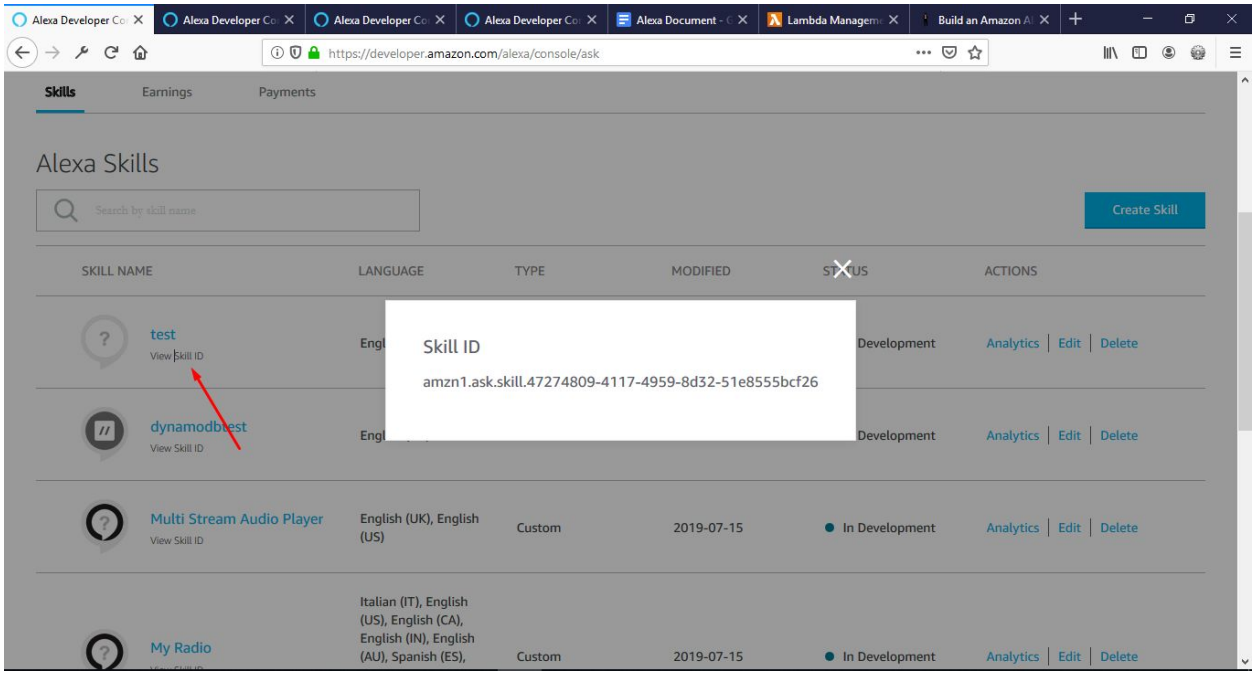
Add skill ID(you can find skill id from developer.amazon console)



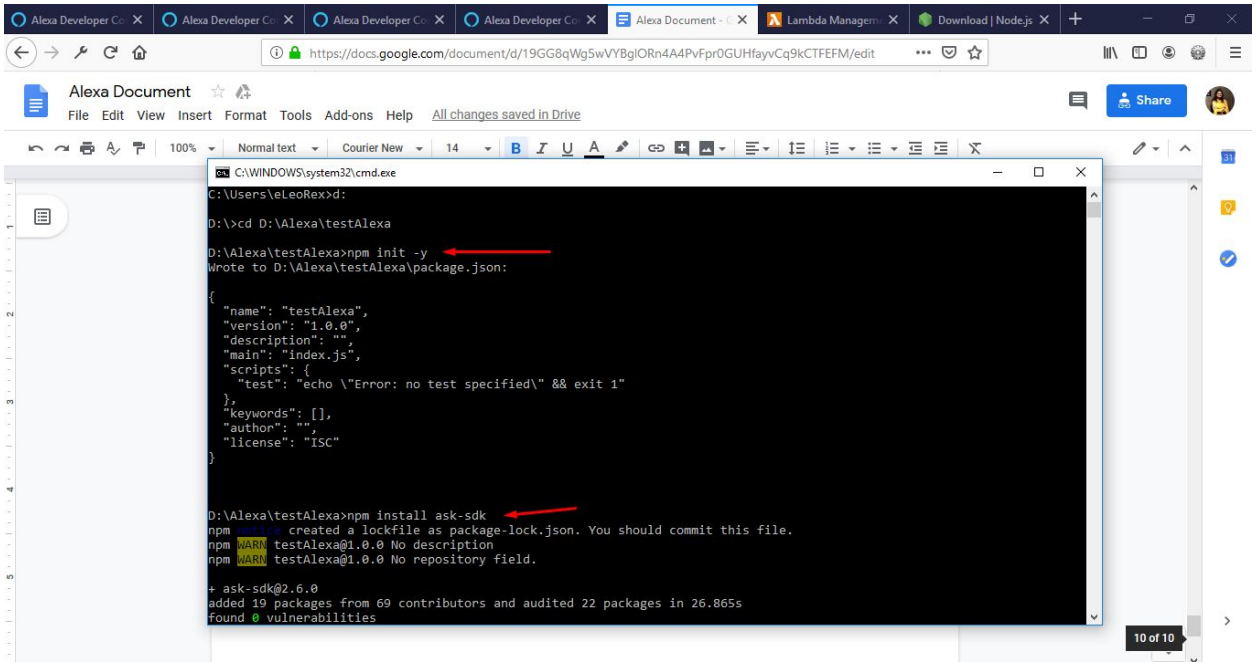
Go to developer.amazon console

Find your skill(in this case **test**)

Click View skill ID



5.Open any editor of your choice
(You should have installed node js and npm to create lambda function)
Set up basic node project using **npm install -y**
Install ask-sdk using **npm install ask-sdk**

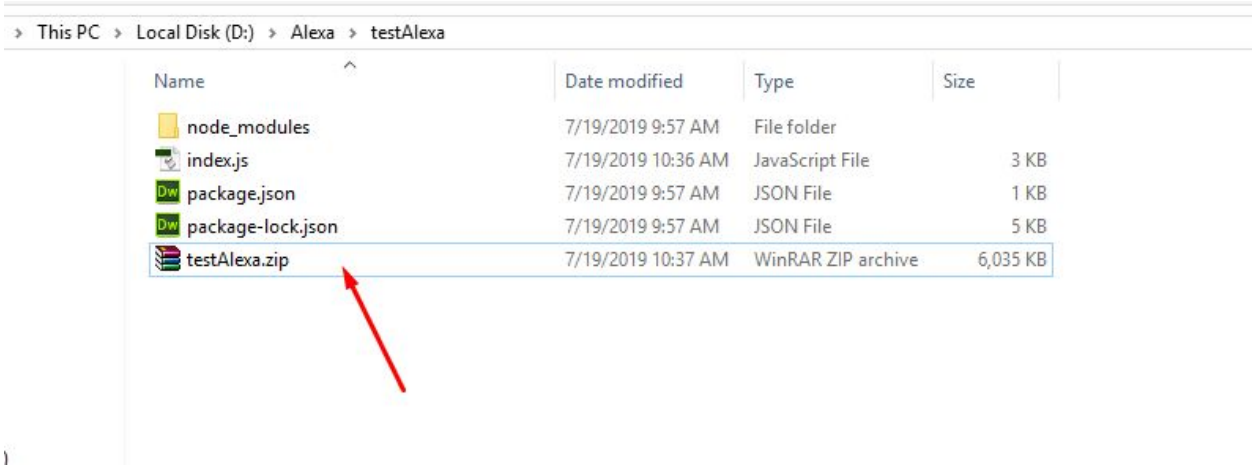


Now create index.js file and copy paste below mentioned code.

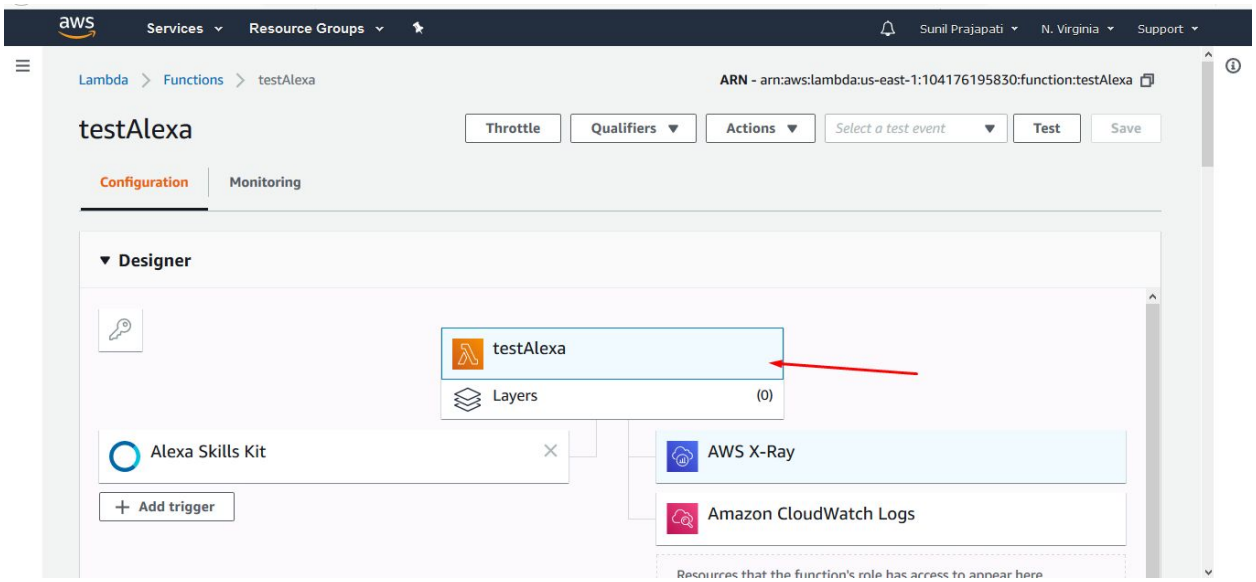
```
const Alexa = require('ask-sdk');

const TestHandler = {
 canHandle(handlerInput) {
   const request = handlerInput.requestEnvelope.request;
   return request.type === 'LaunchRequest'
     || (request.type === 'IntentRequest'
       && request.intent.name === 'testIntent');
 },
 handle(handlerInput) {


   return handlerInput.responseBuilder
     .speak('Hello congrats ,this is response of your skill from lambda')
     .getResponse();
 },
};

const HelpHandler = {
 canHandle(handlerInput) {
   const request = handlerInput.requestEnvelope.request;
   return request.type === 'IntentRequest'
     && request.intent.name === 'AMAZON.HelpIntent';
 },
 handle(handlerInput) {
   return handlerInput.responseBuilder
     .speak(HELP_MESSAGE)
     .reprompt(HELP_REPROMPT)
     .getResponse();
 },
};

const ExitHandler = {
 canHandle(handlerInput) {
   const request = handlerInput.requestEnvelope.request;
   return request.type === 'IntentRequest'
     && (request.intent.name === 'AMAZON.CancelIntent'
       || request.intent.name === 'AMAZON.StopIntent');
 },
 handle(handlerInput) {
   return handlerInput.responseBuilder
     .speak(STOP_MESSAGE)
     .getResponse();
 },
};

const SessionEndedRequestHandler = {
```

```
 canHandle(handlerInput) {
   const request = handlerInput.requestEnvelope.request;
   return request.type === 'SessionEndedRequest';
 },
 handle(handlerInput) {
   console.log(`Session ended with reason:
${handlerInput.requestEnvelope.request.reason}`);

   return handlerInput.responseBuilder.getResponse();
 },
};

const ErrorHandler = {
 canHandle() {
   return true;
 },
 handle(handlerInput, error) {
   console.log(`Error handled: ${error.message}`);

   return handlerInput.responseBuilder
     .speak('Sorry, an error occurred.')
     .reprompt('Sorry, an error occurred.')
     .getResponse();
 },
};



const HELP_MESSAGE = 'You can say test me';
const HELP_REPROMPT = 'What can I help you with?';
const STOP_MESSAGE = 'Goodbye!';



const skillBuilder = Alexa.SkillBuilders.standard();

exports.handler = skillBuilder
 .addRequestHandlers(
   TestHandler,
   HelpHandler,
   ExitHandler,
   SessionEndedRequestHandler
 )
 .addErrorHandlers(ErrorHandler)
 .lambda();
```

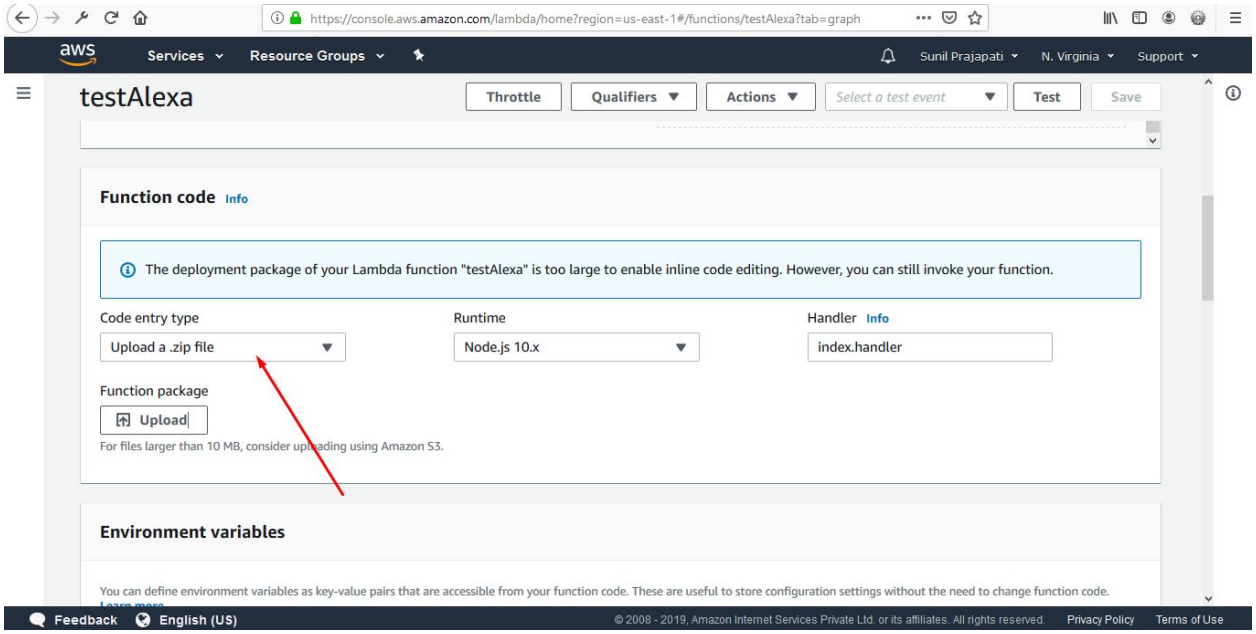
Save this and zip all files



Go to aws.amazon console
Click on lambda function
Scroll down

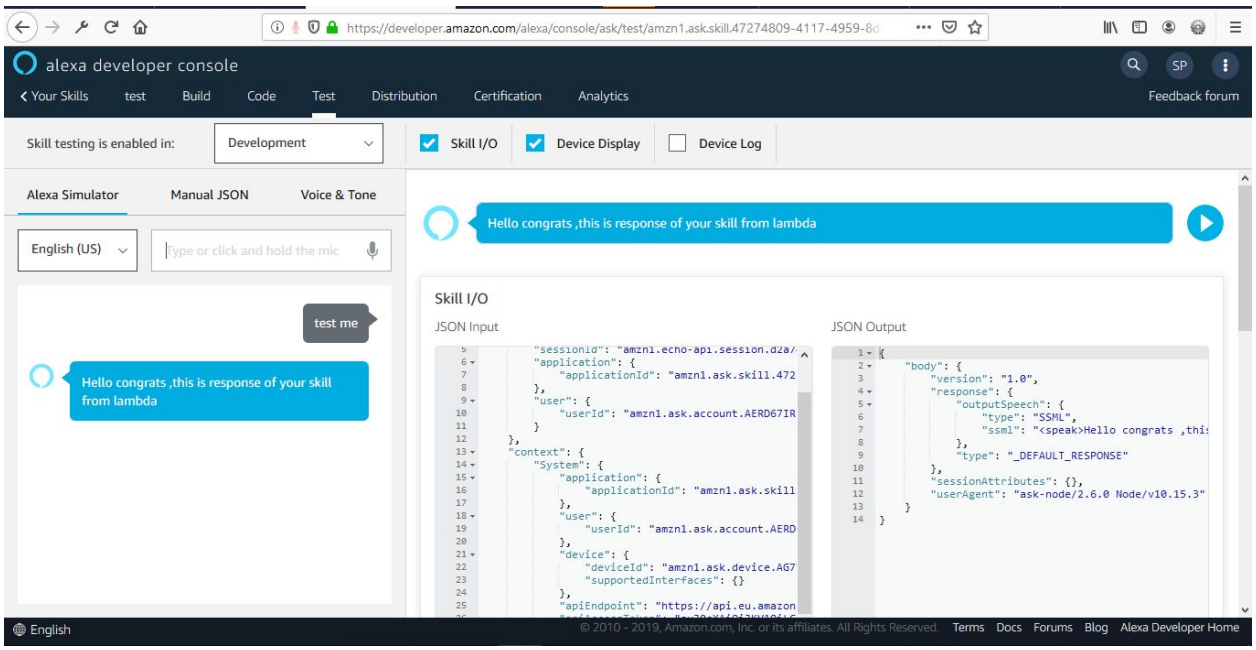


In function code tab
Choose code entry type:upload a zip(upload previously created zip folder here)



On Amazon Developer Portal, Go to **Alexa Skills Kit Developer Console**
On Endpoint, Choose AWS Lambda ARN as service endpoint type and add your ARN (Which we got from AWS Lambda) at Default Region.
(save & build)

Now test your skill using invocation name



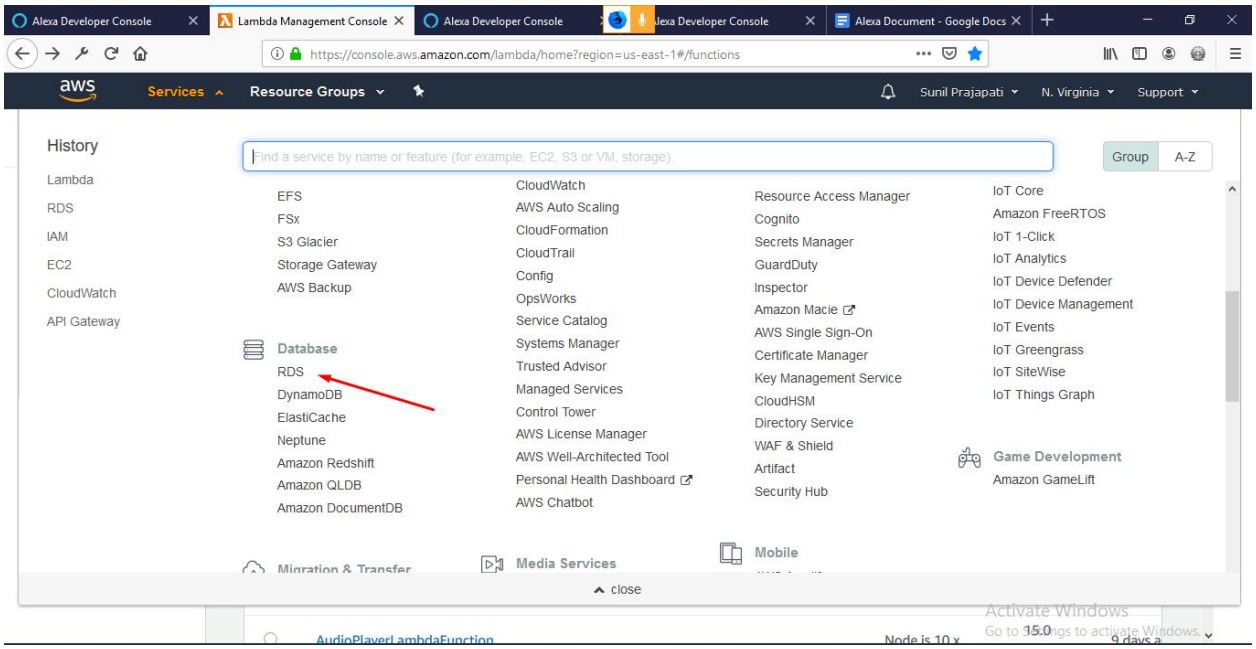
(You will be able to see JOSN input and JSON output)

For testing your own custom skill on your echo device Register your echo device with same email id (note:choose same language type for device and skill. For example if you have created skill using English(US) then also select English(US) for echo device).

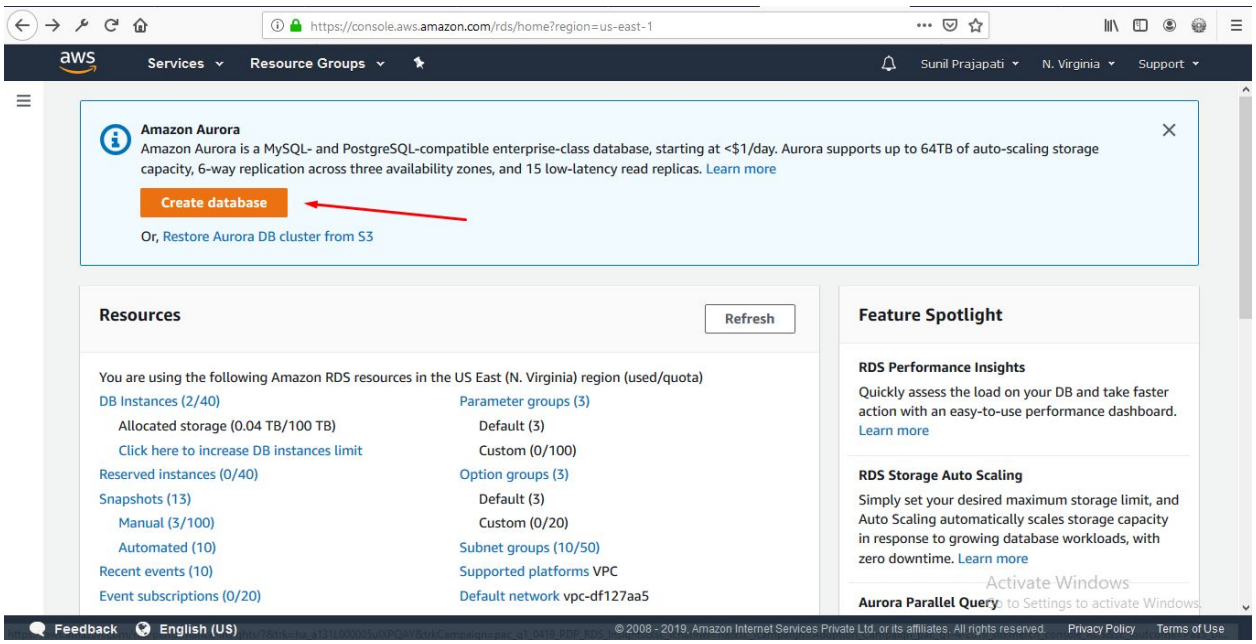
Connect Alexa to Database

Alexa=>LambdaFunction=>AWS RDS.

1.Click RDS(database) from Services tab



2.Click Create Database



Follow this steps to create Database

https://aws.amazon.com/getting-started/tutorials/create-mysql-db/

Note:Use sync-mysql to make synchronous queries to a mysql database