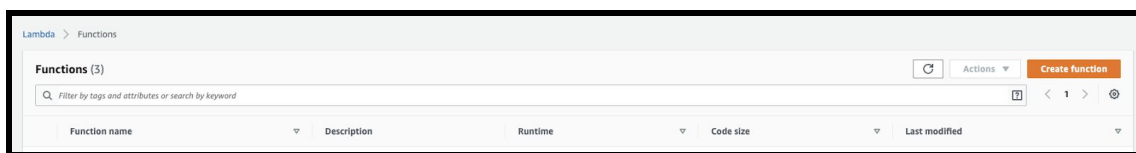
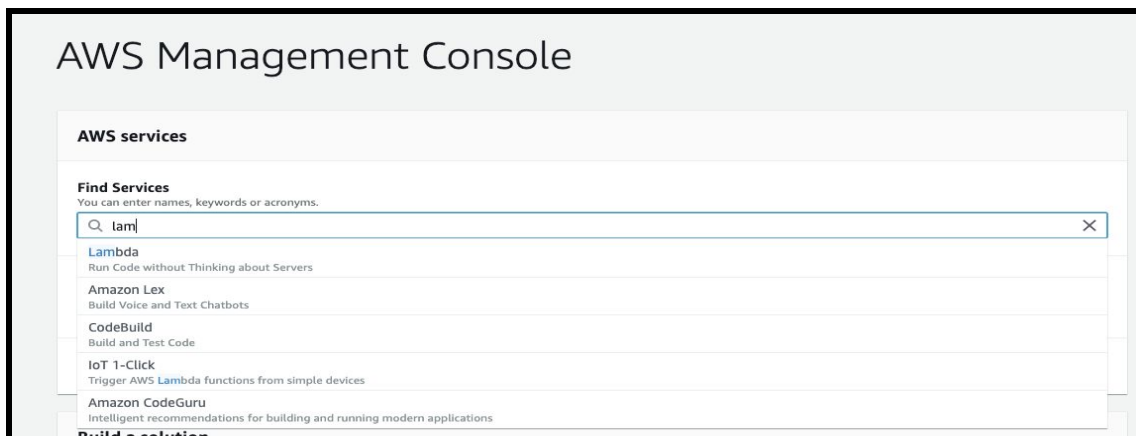


CONVO - AMAZON ALEXA GUIDE

Hope now you have downloaded the **convo.zip** after submitting your REST API documentation to our web application. Follow the following steps in order to deploy your custom skill to amazon alexa.

1. Create an aws account if you haven't created an account before using the following link.
https://console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin
2. After login in to the system under AWS services search for lamda. And click on it to navigate to create a new lambda function.



- Then click on the create function. And select the author from scratch option, then give your function a unique name and select python 3.7 as your run time. Then click create function.

The screenshot shows the 'Create function' page in the AWS Lambda console. At the top, there are three tabs: 'Author from scratch' (selected), 'Use a blueprint', and 'Browse serverless app repository'. Below these tabs is a 'Basic information' section with the following fields:

- Function name:** A text input field with the placeholder 'YourFunctionName'.
- Runtime:** A dropdown menu set to 'Python 3.7'.
- Permissions:** A section with a link to 'Choose or create an execution role'.

At the bottom right of the page, there are 'Cancel' and 'Create function' buttons.

- Then create a trigger for your lambda function. Click on the add trigger button. And select Alexa Skills Kit as the trigger. Then click on enable and leave the skill ID field empty for now.


The screenshot shows the 'Add trigger' page in the AWS Lambda console. The page title is 'Add trigger'. Below the title is a 'Trigger configuration' section with a search bar labeled 'Select a trigger'. A list of triggers is displayed below the search bar:

- API Gateway (api, application-services, aws, serverless)
- AWS IoT (aws, devices, iot)
- Alexa Skills Kit (alex, iot) - This option is highlighted with a grey background.
- Alexa Smart Home (alex, iot)
- Application Load Balancer (aws, load-balancing)
- CloudFront (aws, cdn, edge)
- CloudWatch Events/EventBridge (aws, events, management-tools)
- CloudWatch Logs (aws, logging, management-tools)

Lambda > Add trigger

Add trigger

Trigger configuration

 Alexa Skills Kit
alexa iot

Skill ID verification is an easy way to verify the Skill ID in an incoming request from a Skill. To set this up, enter the Skill ID (also called Application ID) of your skill located in your Alexa Skills Kit dashboard. [Learn more.](#)

Skill ID verification

☒ Enable (recommended)

☐ Disable

Skill ID

Lambda will add the necessary permissions for Amazon Alexa to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

5. Then log in to the alexa developer console using the same credentials. Using the following link. <https://developer.amazon.com/alexa/console/ask>.

6. Click on the create skill button.

alexa developer console

Skills Earnings Payments Hosting

Alexa Skills

Create Skill

SKILL NAME	LANGUAGE	TYPE	MODIFIED	STATUS	ACTIONS
------------	----------	------	----------	--------	---------

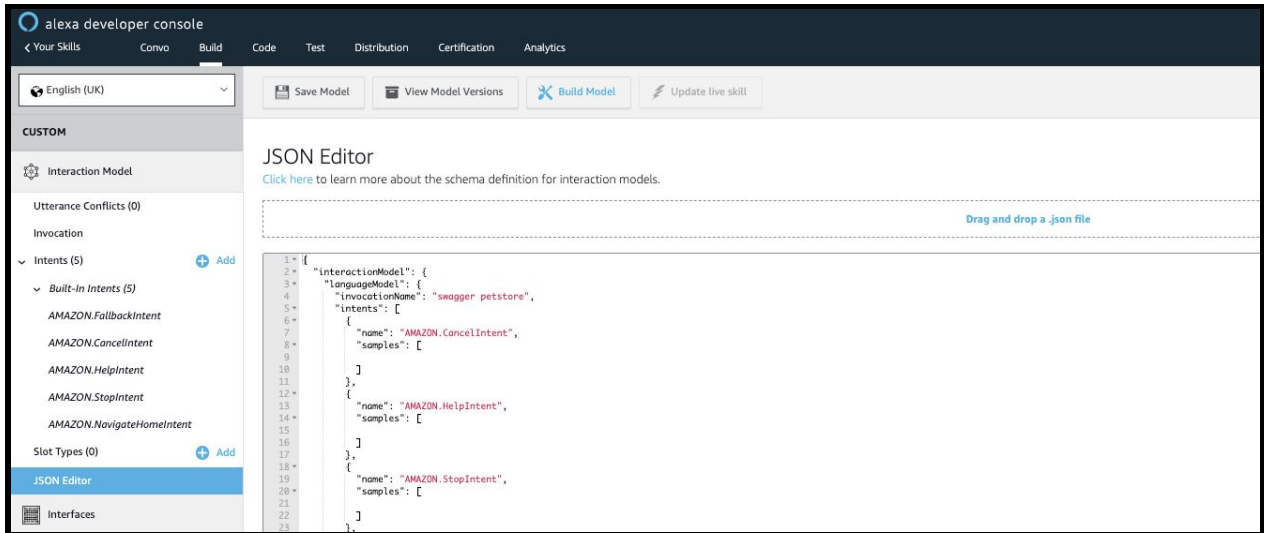
7. Enter a skill name and choose the options as in the following figure.

The screenshot shows the 'Create a new skill' page. At the top right are 'Cancel' and 'Create skill' buttons. Below the title, there's a 'Skill name' field with 'Convo' entered and a character count of 6/50. A 'Default language' dropdown is set to 'English (US)'. A note states: 'More languages can be added to your skill after creation'. Section 1, 'Choose a model to add to your skill', includes a sub-note: 'There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.' It features four options: 'Custom' (with a 'Tutorial' badge), 'Flash Briefing', 'Smart Home', and 'Video'. Each option has a description and a sample utterance box. Section 2, 'Choose a method to host your skill's backend resources', includes a sub-note: 'You can provision your own backend resources or you can have Alexa host them for you. If you decide to have Alexa host your skill, you'll get access to our code editor, which will allow you to deploy code directly to AWS Lambda from the developer console.' It offers three hosting methods: 'Provision your own' (with a 'Tutorial' badge), 'Alexa-Hosted (Node.js)', and 'Alexa-Hosted (Python)'.

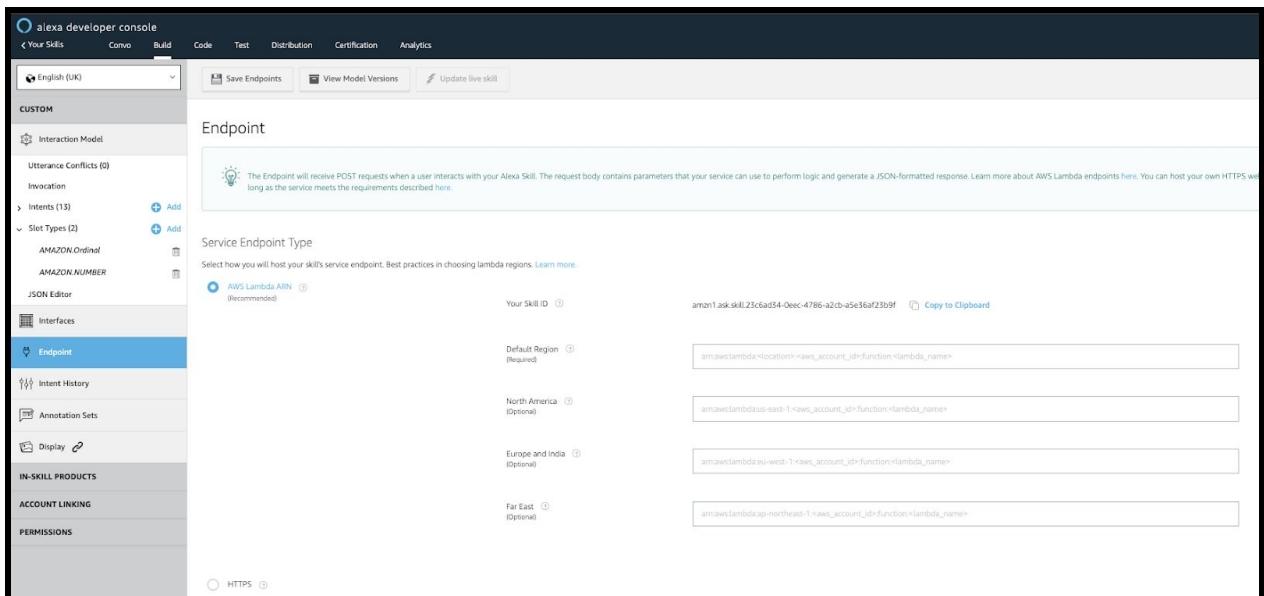
8. Click on create skill. Then choose **Start from scratch** template.

The screenshot shows the 'Choose a template' page. At the top right is a 'Choose' button. Below the title, it says: 'Select a quick start template to get started with a predefined skill or simply "Start from scratch"'. There are four template cards: 'Start from scratch' (with a 'Tutorial' badge), 'Fact Skill', 'Quiz Game Skill', and 'High-Low Game Skill'. Each card contains a brief description of the template's capabilities.

9. Now you will be directed to the alexa developer console. Click on the json editor and copy paste the content of the **interaction-model.json** that you have downloaded (inside the convo.zip) to the editor. (you can drag and drop too). Then click on save.



10. Then click on Endpoint.



- Copy your skill id. And paste it in the blank field that we have left in the aws.console. Then click Add.

The screenshot shows the 'Add trigger' configuration page in the AWS Lambda console. The 'Trigger configuration' section is active, showing the 'Alexa Skills Kit' trigger type. A blue information box explains Skill ID verification. The 'Skill ID verification' section has 'Enable (recommended)' selected. The 'Skill ID' field contains the value 'amzn1.ask.skill.23c6ad34-0eec-4786-a2cb-a5e36af23b9f'. At the bottom, there are 'Cancel' and 'Add' buttons.

Lambda > Add trigger

Add trigger

Trigger configuration

Alexa Skills Kit
alexa iot

Skill ID verification is an easy way to verify the Skill ID in an incoming request from a Skill. To set this up, enter the Skill ID (also called Application ID) of your skill located in your Alexa Skills Kit dashboard. [Learn more.](#)

Skill ID verification

☒ Enable (recommended)
☐ Disable

Skill ID

amzn1.ask.skill.23c6ad34-0eec-4786-a2cb-a5e36af23b9f

Lambda will add the necessary permissions for Amazon Alexa to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

- Now click on your function name. Then under that there will be a component that is showing some code. Delete that code and copy and paste the content of the lambda.py that you have downloaded.

The screenshot shows the AWS Lambda console for a function named 'YourFunctionName'. A green notification bar at the top states that the trigger was successfully added. The 'Configuration' tab is selected, showing the 'Designer' view with the 'Alexa Skills Kit' trigger. Below the designer, the 'Function code' section is visible, showing the 'Code entry type' as 'Edit code inline', the 'Runtime' as 'Python 3.7', and the 'Handler' as 'lambda_function.lambda_handler'. The code editor at the bottom shows the content of 'lambda_function.py'.

Lambda > Functions > YourFunctionName

ARN: arn:aws:lambda:us-east-1:931256542236:function:YourFunctionName

Throttle Qualifiers Actions Select a test event Test Save

The trigger 23c6ad34-0eec-4786-a2cb-a5e36af23b9f was successfully added to function YourFunctionName. The function is now receiving events from the trigger.

Configuration Permissions Monitoring

▼ Designer

YourFunctionName
Layers (0)

Alexa Skills Kit
+ Add trigger + Add destination

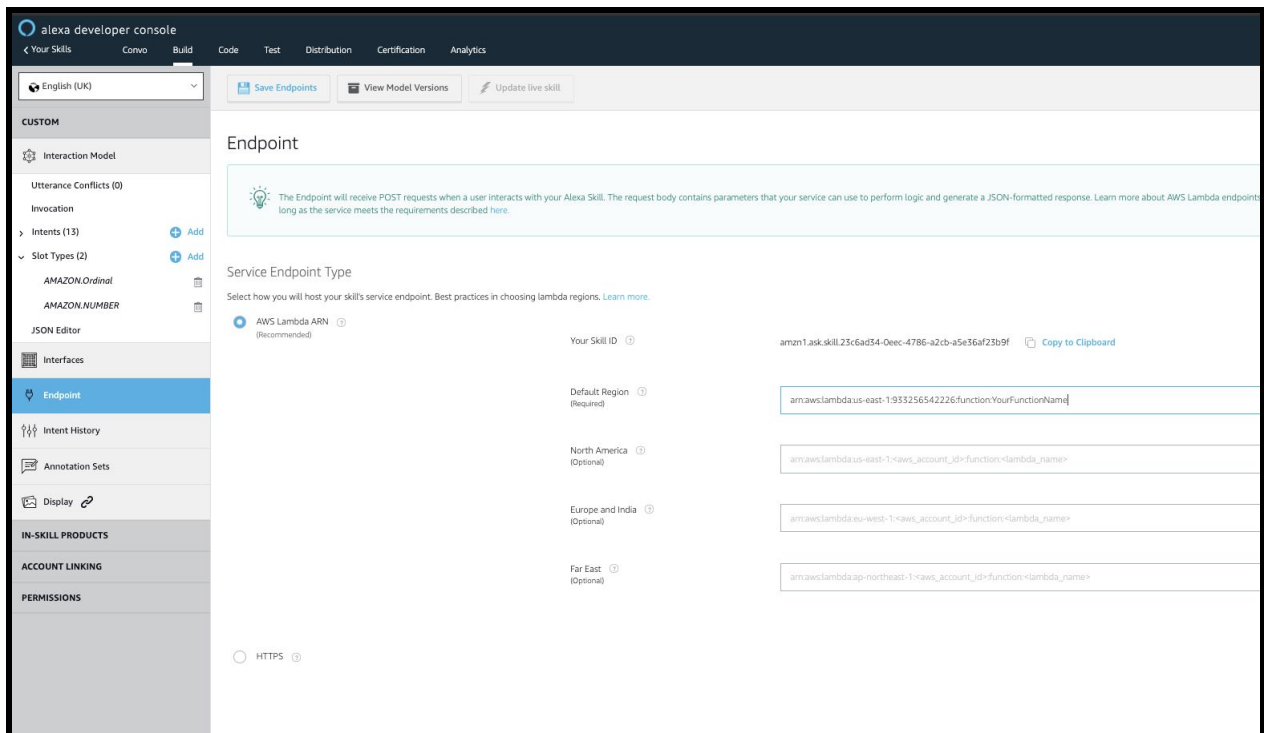
Function code [help](#)

Code entry type: Edit code inline Runtime: Python 3.7 Handler: lambda_function.lambda_handler

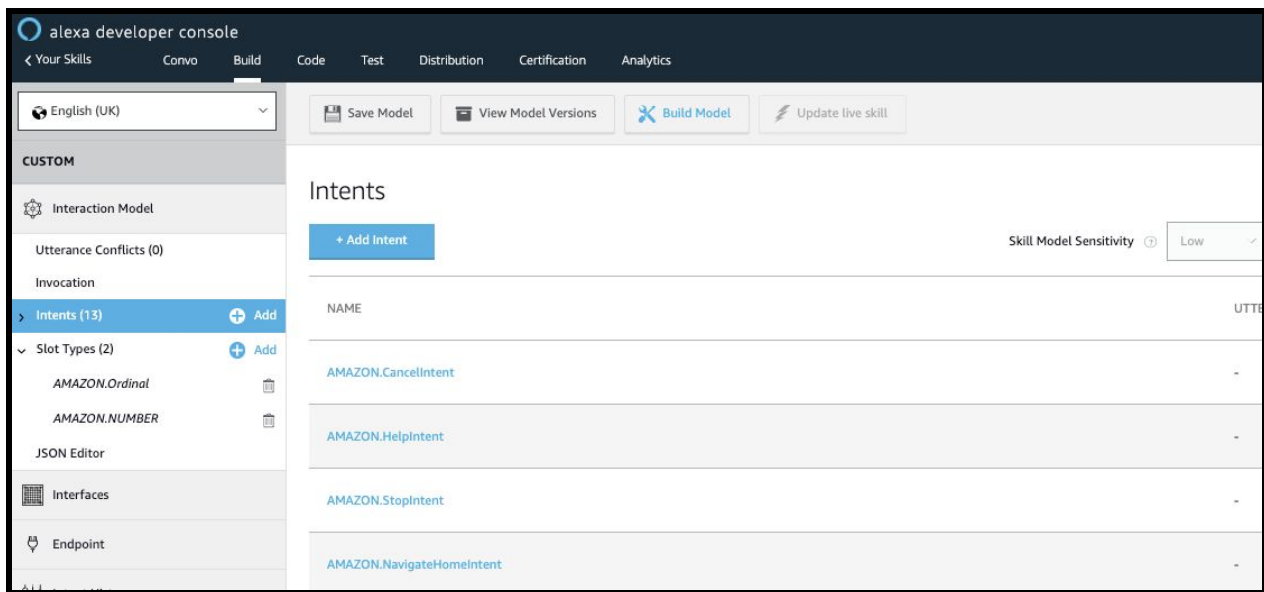
```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {}
6
7     "statusCode": 200,
8     "body": json.dumps("Hello from Lambda!")
9 }
```

13. Copy the ARN that is visible in the top right corner of the window.

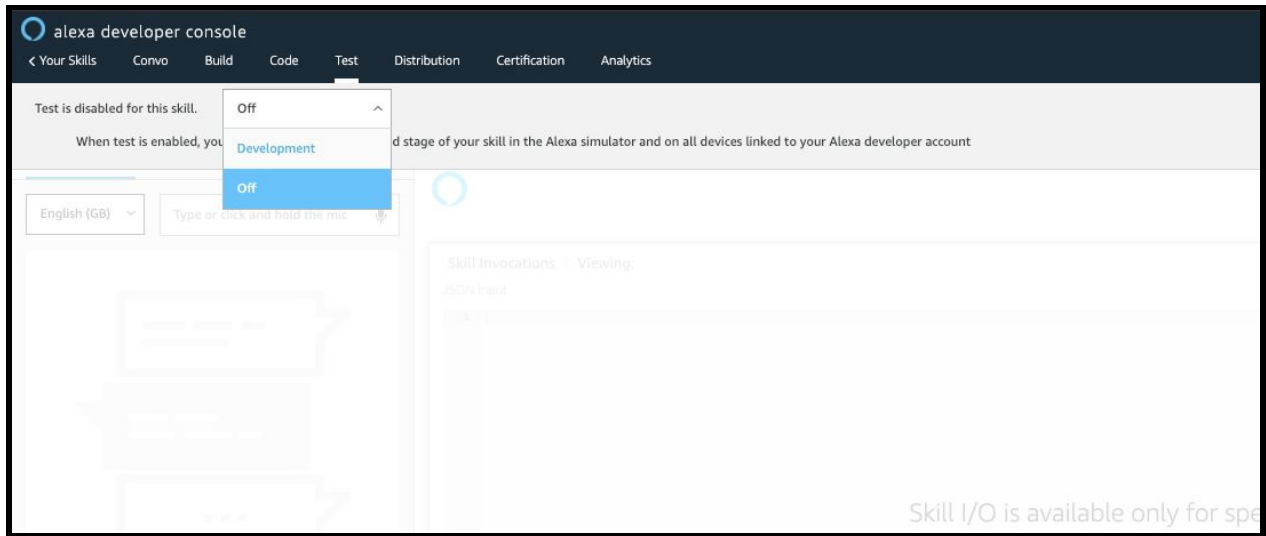
14. Paste the ARN in alexa developer console, in the default region option



15. Click on Save Endpoints. And click on the intents tab to view your intents.



16. Now click on the **Build Model**. To build your skill. After the build is successful click on the Test tab and enable testing by selecting development.



17. Now you can type your commands or use an inbuilt microphone to give commands to alexa.

