

Alexa voice service skill POC

Alexa Skills introduction :

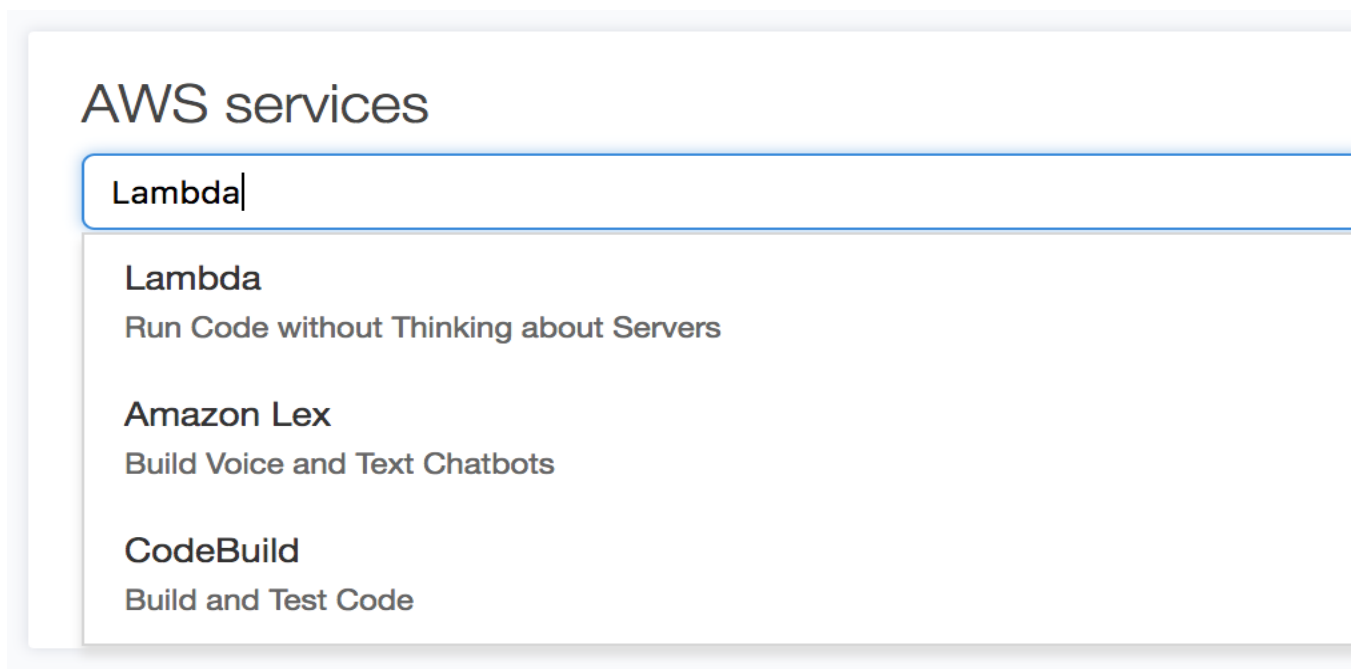
***Alexa** is Amazon's Voice Service, accessed through one of Alexa-enabled devices like **Amazon Echo**.*

***Alexa Skills** are like apps. Skills are voice-driven Alexa capabilities. In other words, Skills are the interface between users and Alexa.*

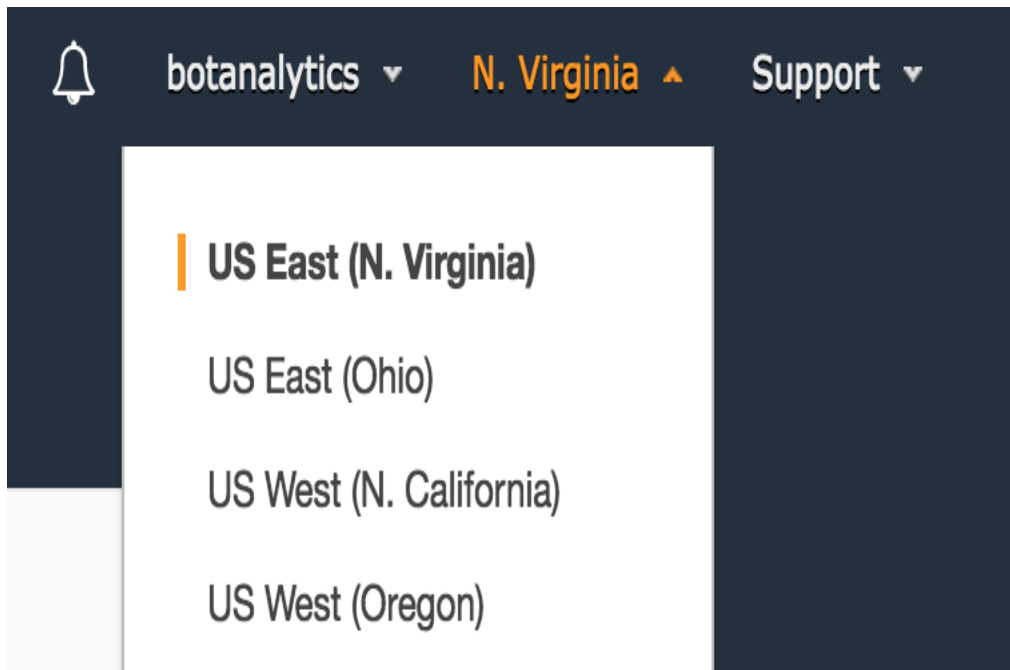
We'll handle the Alexa Skill development process under 3 parts. 1st part will be about signing in to Amazon Web Services (AWS) and creating a function and at 2nd part we'll use the Amazon Development Portal to configure our skill. 3rd part will be a connecting the skill and the lambda function configurations. Finally, we can test of our newly configured skill. You'll end with a simple but complete Alexa skill that can run on Amazon Echo or any device with Alexa.

Alexa voice service implementation steps :

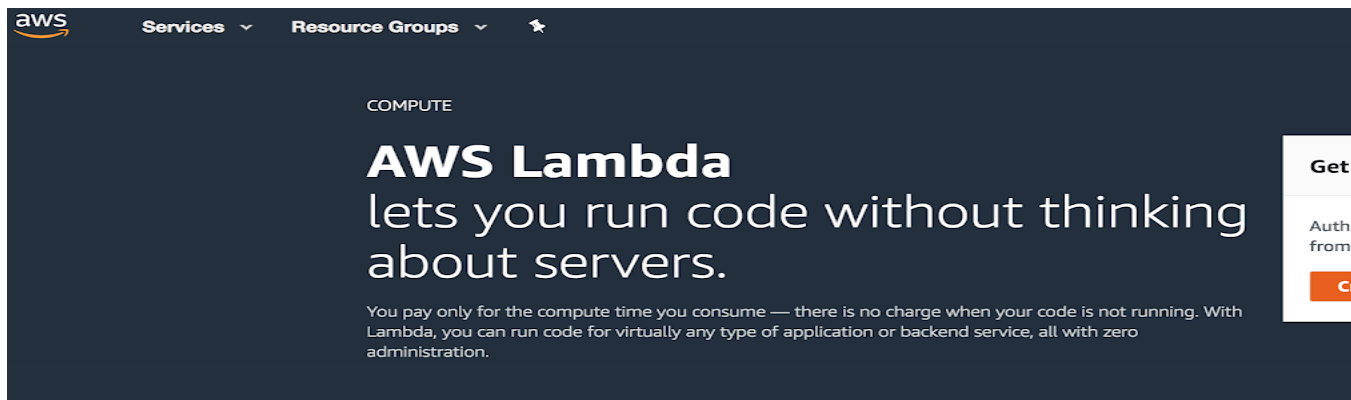
1. **Part 1 - AWS Lambda - Creating Lamba funtion on Amazon Web Services (AWS)**
 - Sign in to [AWS Management Console](#) , if you don't have any account then you can create a free account.
 - From the list of services, select **Lambda**.



1.
 - Click the region drop-down in the upper-right corner of the console and select **US East (N. Virginia)**, which is a supported region for Lambda functions used with the Alexa Skills Kit.



- Choose **Create a Function** to choose a blueprint for your new function.



- In the search filter box, type **alexa**. Select blueprint **alexa-skills-kit-color-expert-python** from the results.

Create function

Author from scratch ☐

Start with a simple "hello world" example.



Blueprints ☒

Choose a preconfigured template as a starting point for your Lambda function.



Blueprints [Info](#)

Add filter

keyword : alexa

[alex-skill-kit-sdk-factskill](#) ☐

Demonstrate a basic fact skill built with the ASK NodeJS SDK

nodejs6.10 · alexa

[alex-skill-kit-sdk-triviaskill](#) ☐

Demonstrate a basic trivia skill built with the ASK NodeJS SDK

nodejs6.10 · alexa

[alex-smart-home-skill-adapter](#) ☐

Provides the basic framework for a skill adapter for a smart home skill.

nodejs · iot · smart-home · alexa · light

[alex-expert-python](#) ☐

Demonstrates a basic skill built with the Amazon Alexa Skills Kit.

nodejs

[alex-skills-kit-color-expert-python](#) ☐

Demonstrates a basic skill built with the Amazon Alexa Skills Kit.

python2.7 · alexa

Name your function. We'll use **myTeamSkill**. Then, under the Lambda function handler and role, select the Create a custom role option.

Basic information [Info](#)

Name*

myTeamSkill

Role*

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. For more information, see [About Lambda execution roles](#).

Create a custom role



The custom role creation experience will open in a new tab. Ensure that popups are enabled to create a custom role.

2.
 - When the IAM role management console opens, click on the **Allow** to go back to the previous Lambda console.

AWS Lambda requires access to your resources

AWS Lambda uses an IAM role that grants your custom code permissions to access AWS resource

▼ Hide Details

Role Summary ?

Role

Lambda execution role permissions

Description

IAM Role

Create a new IAM Role

Role Name

lambda_basic_execution

▶ View Policy Document

-
- Scroll down and click on **Create Function**.

Lambda function code

Code is pre-configured by the chosen blueprint. You can configure it after you create the function.

Runtime

Python 2.7

```
183 """ Route the incoming request based on type (LaunchRequest
184 etc.) The JSON body of the request is provided in the event
185 """
186 print("event.session.application.applicationId=" +
187       event['session']['application']['applicationId'])
188
189 """
190 Uncomment this if statement and populate with your skill's
191 prevent someone else from configuring a skill that sends re
192 function.
193 """
194 # if (event['session']['application']['applicationId'] !=
195 #     "amzn1.echo-sdk-ams.app.[unique-value-here]"):
196 #     raise ValueError("Invalid Application ID")
197
198 if event['session']['new']:
199     on_session_started({'requestId': event['request']['requ
200                       event['session']})
201
202 if event['request']['type'] == "LaunchRequest":
203     return on_launch(event['request'], event['session'])
204 elif event['request']['type'] == "IntentRequest":
205     return on_intent(event['request'], event['session'])
206 elif event['request']['type'] == "SessionEndedRequest":
207     return on_session_ended(event['request'], event['sessio
208
```

* These fields are required.

- Copy the Amazon Resource Name (ARN) displayed in the upper-right corner of the console that starts with **arn:aws:lambda:us-east....**

Lambda > Functions > myTeamSkill


myTeamSkill Qualifiers


✔ Congratulations! Your Lambda function "myTeamSkill" has been successfully created and configured with as a trigger in a disabled state. We recommend that you enable the function.

Configuration **Monitoring**

▼ **Designer**

Add triggers
Click on a trigger from the list below to add it to your function.




myTeamSkill

2. Part 2 - Alexa Skills Kit - Configuring custom skill on Amazon Development Portal

- Sign in to [Amazon Development Portal](#) , if you don't have any account then you can create a free account.
- Go to [Create a New Alexa Skill](#) page
- Name your skill. This will be the name displayed to the users of the Alexa app. We'll use **Color Picker** for this example.

alexa developer console Feedback forum

Create a new skill Cancel Create skill

Skill name
Color Picker 1-20 characters

Default language
English (US) More languages can be added to your skill after creation.

Choose a model to add to your skill.
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.

Custom	Flash Briefing	Smart Home	Music	Video	Baby Activity
Design a unique experience for your users. A custom model enables you to create all of your skill's interactions.	Give users control of their news feed. This pre-built model lets users control what updates they follow to.	Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.	Give users complete control of their music. This pre-built model lets users search, pause, skip, or shuffle in your skill.	Let users find and consume video content. This pre-built model supports content searches and content suggestions.	Let users log and retrieve events for their infants. This pre-built model supports diaper changes, feedings, sleep, and weight.
	"Alexa, what's in the news?"	"Alexa, turn on the kitchen light?"	"Alexa, play music by Lady Gaga?"	"Alexa, play Interstellar?"	"Alexa, record a dirty diaper?"

Choose a method to host your skill's backend resources
You can self-host your backend resources or you can have Alexa host it 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.

Self Hosted	Alexa-Hosted (Beta)
Use your own AWS Lambda or another AWS FaaS endpoint to power your skill.	Alexa will host your skill on AWS or on the Free Tier limits, with access to an AWS Lambda endpoint, 5 GB of media storage with 10 GB of monthly data transfer, and a table for session persistence. Learn more

- ### Skill builder checklist

Complete these steps to be able to test your skill using the simulator in the test tab, or with your echo device.

REQ-APP	1. Invocation Name > Enter an invocation name for your skill	✓
REQ-APP	2. Intents, Samples, and Slots > Add at least one intent and one sample utterance	✓
REQ-APP	3. Build Model > Successfully build your interaction model	✓
REQ-APP	4. Endpoint > Set a web service endpoint to handle skill requests	✓
OPTIONAL	In-Skill Products > Create an in-skill product and add it to your skill	

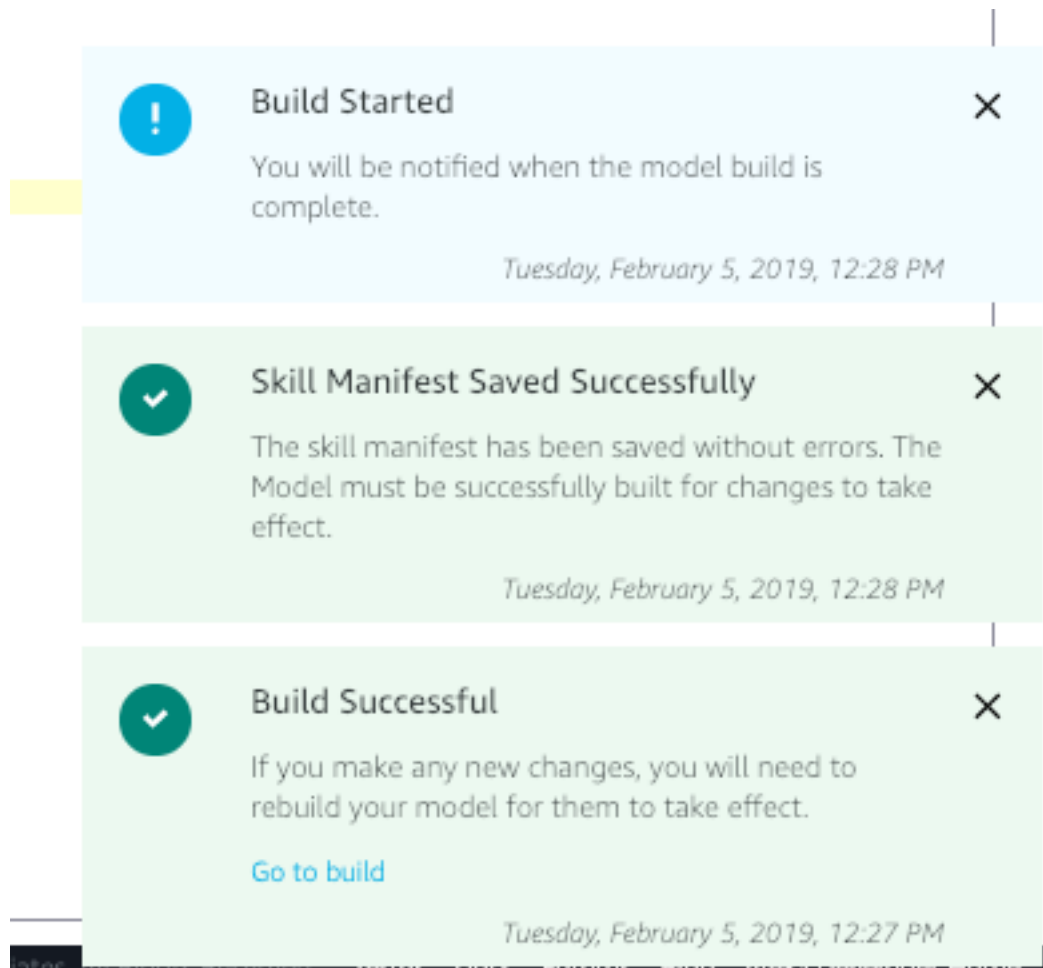
- The screenshot shows the AWS Skill Developer console interface. On the left is a navigation sidebar with categories like 'NEW SKILLS', 'MY SKILLS', 'IN-SKILL PRODUCTS', 'ACCOUNT SETTINGS', and 'PERMISSIONS'. The main area is titled 'Invocation' and contains a text input field for the 'Invocation Name'. Below this, a light green box lists 'Invocation name requirements'. The requirements state that the name should be lowercase with hyphens or underscores, avoid certain words, and be unique. A 'Build' button is visible in the top right corner of the main area.

- After [Create a new skill](#) above, now select **Custom** for the interaction model.
- Navigate to **Custom > Interaction Model > JSON Editor**.
- Delete the default code, then paste in the JSON shown below, then click **Save Model** and **Build Model**.
- Navigate to **Custom > Endpoint**, select **AWS Lambda ARN**, then enter or paste the ARN for your Lambda function in the **Default Region** box.

JSON version the interaction model:

```
{ "interactionModel": { "languageModel": { "invocationName":
"color picker", "intents": [ { "name": "MyColorIsIntent", "slots":
[ { "name": "Color", "type": "LIST_OF_COLORS" } ], "samples": [ "my
favorite color is {Color}" ] }, { "name": "WhatsMyColorIntent",
"slots": [], "samples": [ "what's my favorite color", "what is my
favorite color", "what's my color", "what is my color", "my
color", "my favorite color", "get my color", "get my favorite
color", "give me my favorite color", "give me my color", "what my
color is", "what my favorite color is", "yes", "yup", "sure", "yes
please" ] }, { "name": "AMAZON.HelpIntent", "samples": [] } ],
"types": [ { "name": "LIST_OF_COLORS", "values": [ { "name": {
"value": "green" } }, { "name": { "value": "red" } }, { "name": {
"value": "blue" } }, { "name": { "value": "orange" } }, { "name": {
"value": "gold" } }, { "name": { "value": "silver" } }, { "name": {
"value": "yellow" } }, { "name": { "value": "black" } }, { "name":
{ "value": "white" } } ] } ] } ] } }
```

- Click **Save Model** and wait until the interaction model finishes loading. It takes no more than a couple of seconds.
- Click **Build Model** and check the bottom right corner to see if its successful or check for any issues in compiling and building.



- From the left side navigation bar, go to Endpoint
- Select **North America** as your region and paste your ARN code from Part 1's last item
- For **Account Linking** select **No**, then click **Save Endpoints**.
- The final configurations for the new skill created and successfully built should be seen as follows -



English (US)



CUSTOM



Interaction Model

Invocation

▼ Intents (4)



Add

▼ MyColorsIntent



Color



WhatsMyColorIntent



▼ Built-In Intents (2)

AMAZON.HelpIntent

AMAZON.NavigateHomeIntent

▼ Slot Types (1)



Add

LIST_OF_COLORS



JSON Editor



Interfaces



Endpoint



Intent History

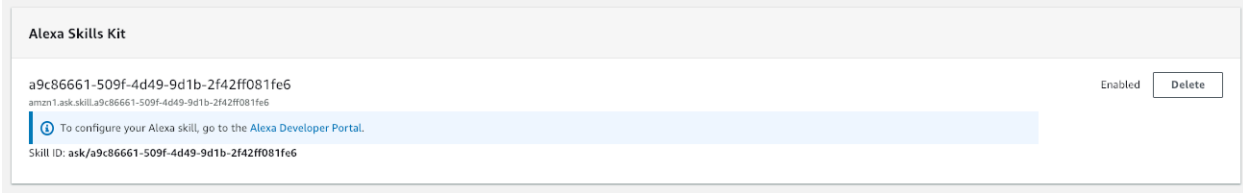
- From the Endpoint page, copy your skill set unique ID - **AWS Lambda ARN : Your Skill ID** for the newly created skill

3. Part 3 - Configuring new Alexa skill with AWS Lambda function

- From Part1's link, [AWS Management Console](#) , navigate to Lambda → Functions → Your custom function (**myTeamSkill**)
- Next navigate to Configuration section → Add Triggers → Click on Alexa Skills Kit (See below)

The screenshot displays the AWS Lambda console interface for configuring a function named 'myTeamSkill'. The top navigation bar shows 'Services' and 'Resource Groups'. The breadcrumb trail indicates the path: 'Lambda > Functions > myTeamSkill'. The function's ARN is displayed as 'arn:aws:lambda:us-east-1:857525326891:function:myTeamSkill'. The 'Configuration' tab is active, and the 'Add triggers' section on the left lists various services, with 'Alexa Skills Kit' selected. The main area shows a diagram of the function 'myTeamSkill' with a layer '(0)' and two triggers: 'Alexa Skills Kit' (with a 'Configuration required' icon) and 'Amazon CloudWatch Logs'. Below the diagram, the 'Configure triggers' section is visible, showing a notification about Skill ID verification and options to enable or disable it. The 'Skill ID' field is empty. At the bottom right, there are 'Cancel' and 'Add' buttons.

- Add the **AWS Lambda ARN : Your Skill ID**, from Part2's last item in the Configure triggers → Skill ID
- Save the changes to **myTeamSkill** and you should be able to see Alexa Skills Kit showing the new skill added to it



Test new Alexa skills by following 2 ways -

1. Test your skill on your Amazon Echo, Echo Dot, or any Alexa-enabled device by saying, "Alexa, open Color Picker."
2. Use the Service Simulator from the Test step of Color Picker development.
 1. [Amazon Development Portal](#) → Alexa → Alexa Skills Kit → Your Custom Skill (Color Picker) → Test
 2. Try entering **Open Color Picker** or **My favorite color is red** to see how Alexa responds.