

Lab 1: Hello Alexa

EE596B/LING580K Conversational Artificial Intelligence

Hao Fang

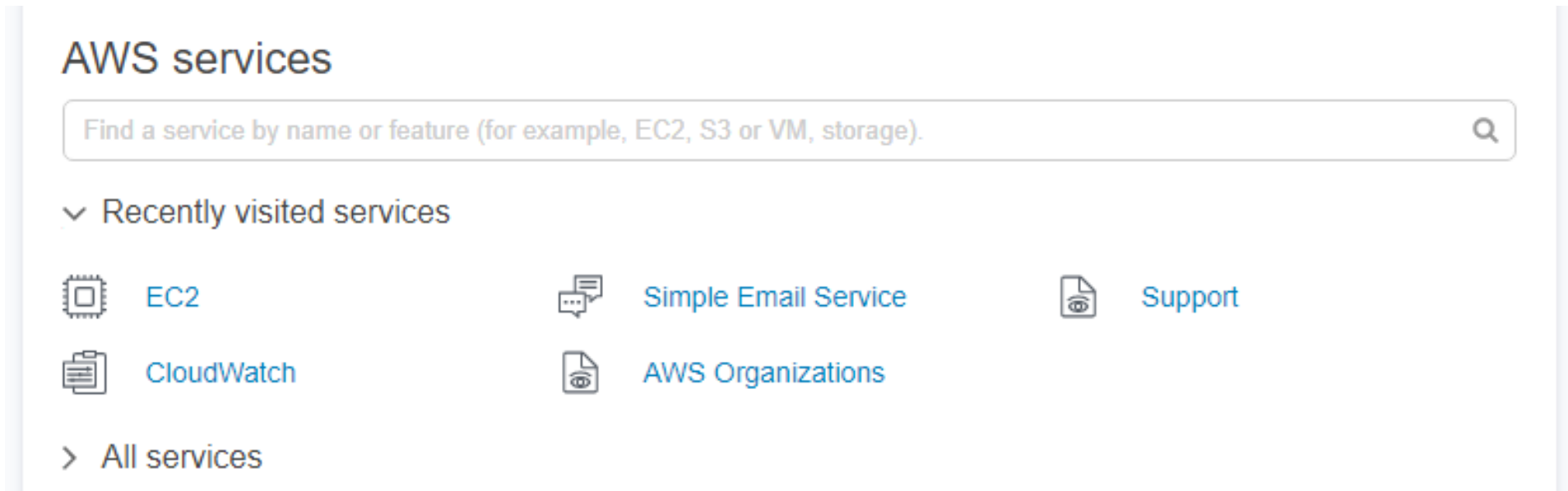
Outline

- Task 1: Create your first Alexa Skill
 - Step 1: Create an AWS Lambda function
 - Step 2: Create an Alexa Skill
 - Step 3: Test your skill using the Service Simulator
- Task 2: Create an AWS Lambda function from scratch
- Task 3: Deploy the AWS Lambda function using command-line

Step 1: Create an AWS Lambda
function

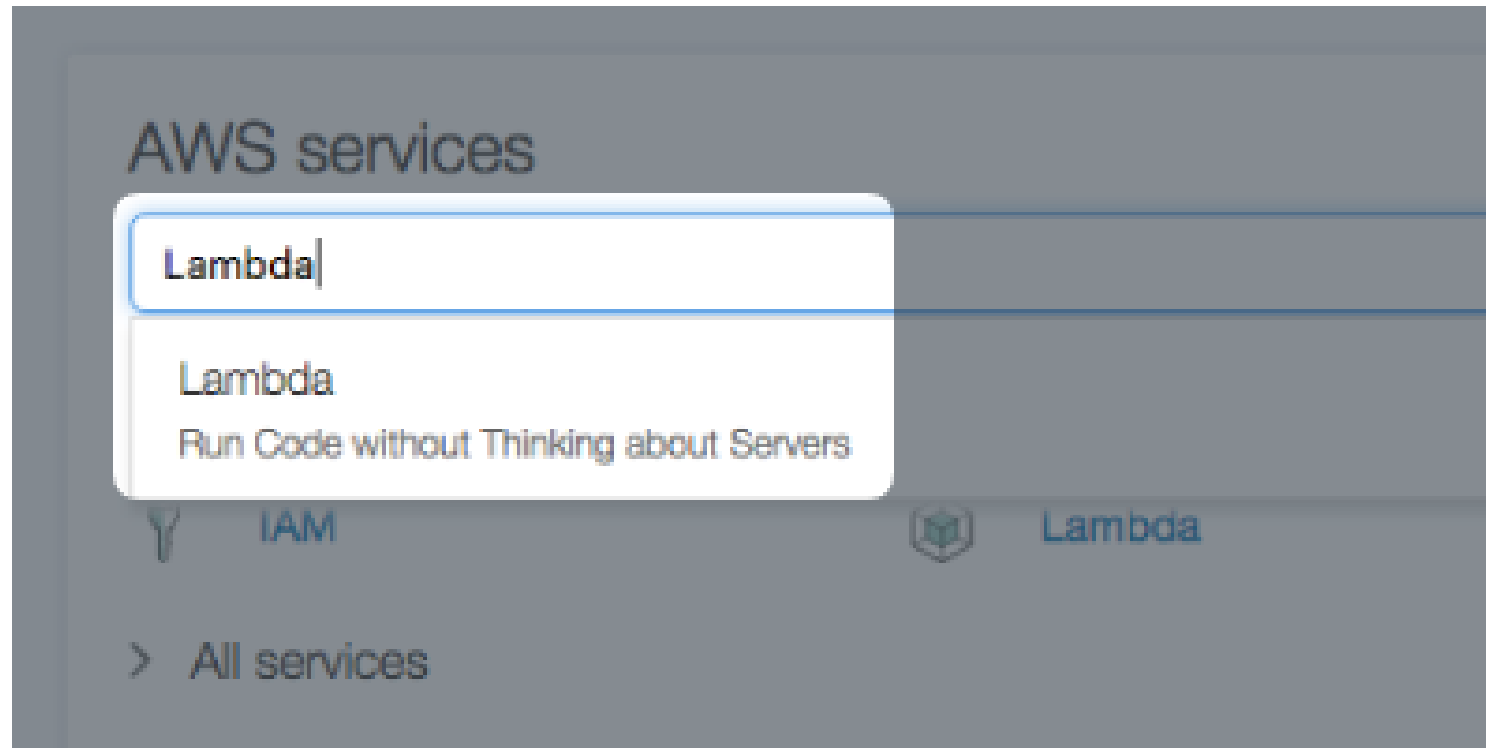
Step 1.1

- Log in to the [AWS Management Console](#). If you haven't done so already, you'll need to [create a free account](#).
 - <http://console.aws.amazon.com>



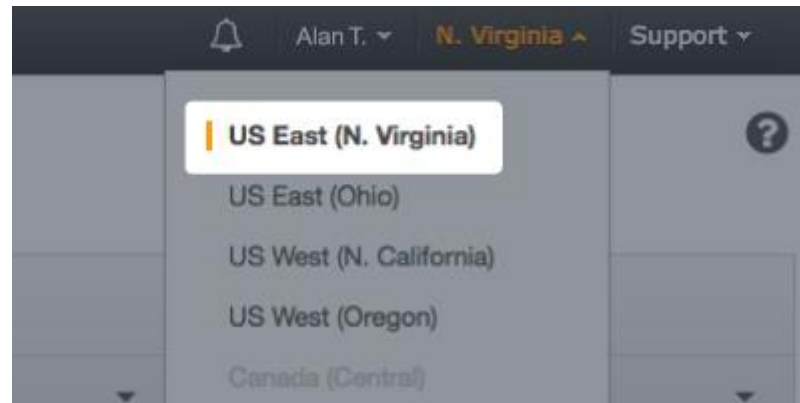
Step 1.2

- From the list of services, select **Lambda**.



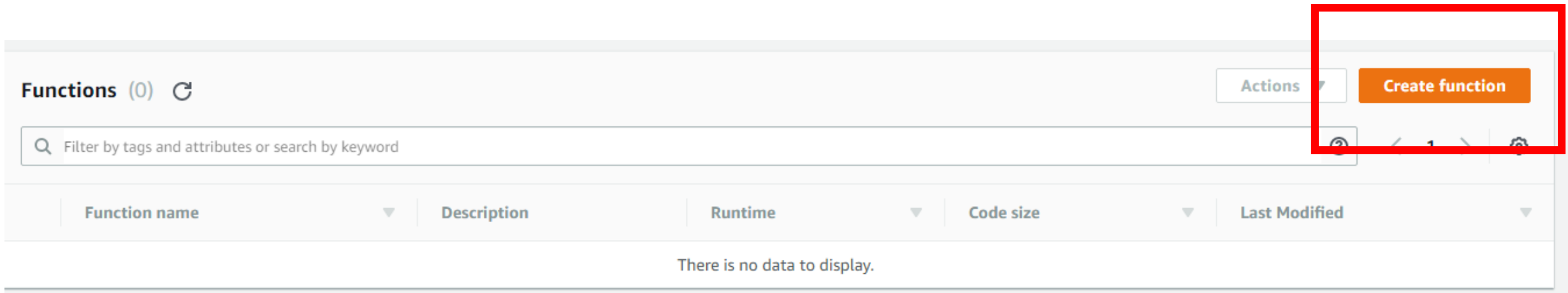
Step 1.3

- 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.



Step 1.4

- Choose **Create a Lambda Function**.



Step 1.5


- Choose a blueprint for your new function.

Lambda > Functions > Create function

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.



Serverless Application Repository

Find and deploy serverless apps published by developers, companies, and partners on AWS.



Blueprints [Info](#)

Filter by tags and attributes or search by keyword

Export

1 2 3 4 5 6 7 ... 12

kinesis-firehose-syslog-to-json

logicmonitor-send-cloudwatch-events

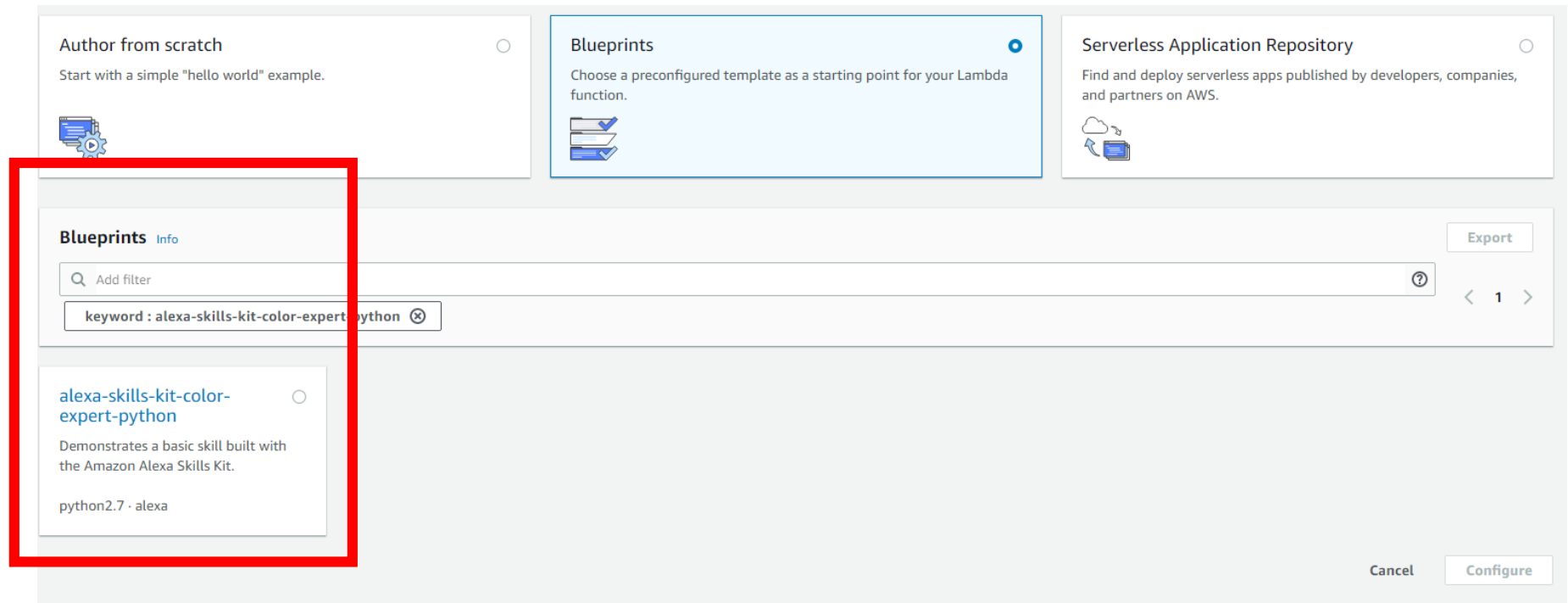
splunk-elb-application-access-logs-processor

alexa-skill-kit-sdk-factskill

batch-get-job-python27

Step 1.6

- Select blueprint **alexa-skills-kit-color-expert-python**. Make sure to choose the blueprint that ends in **-python**.
- Click “Configure”



Step 1.7

- Name your function. We'll use **myColorSkill** for this walkthrough.

Lambda > Functions > Create function > Using blueprint alexa-skills-kit-color-expert-python

Basic information [Info](#)

Name

Role

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

Choose an existing role ▼

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

▼

Step 1.8

- Under **Role**, select **Create a custom role**. A new webpage will be open.

Basic information [Info](#)

Name

myColorSkill

Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Create a custom role ▼

Choose an existing role

Create new role from template(s)

Create a custom role

alexas-skills-kit

Remove

Step 1.9

- When the IAM role management console opens, choose **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 resources.

▼ Hide Details

Role Summary ?

Role Lambda execution role permissions

Description

IAM Role

lambda_basic_execution ▼

Policy Name

Create a new Role Policy ▼

► View Policy Document

Cancel

Allow

Step 1.10

- Make sure the **Existing Role** is `lambda_basic_execution`.

Basic information [Info](#)

Name

Role

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

Choose an existing role ▼

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

lambda_basic_execution ▼

Step 1.11

- For simplicity, let's first **Disable** the Skill ID verification.
 - We will know the Skill ID in Step 2.22.

alexas-skills-kit

Remove

It is a best practice to enable Skill ID verification. If you have existing triggers with Skill ID verification 'Enable', you will not be able to create a new trigger with Skill ID verification 'Disable'. [Learn more](#).

Skill ID verification

☐ Enable (recommended)

☒ Disable

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

Step 1.12

- Click **Create Function**.

```
15 def build_speechlet_response(title, output, reprompt_text, should_end_session):  
16     return {  
17         'outputSpeech': {  
18             'type': 'PlainText',  
19             'text': output  
20         },  
21         'card': {  
22             'type': 'Simple',  
23             'title': "SessionSpeechlet - " + title,  
24             'content': "SessionSpeechlet - " + output  
25         },  
26         'reprompt': {  
27             'outputSpeech': {  
28                 'type': 'PlainText',  
29                 'text': reprompt_text  
30             },  
31             'text': reprompt_text  
32         }  
33     }
```

These fields are required.

Cancel

Previous

Create function

Step 1.13

- Your Lambda function “myColorSkill” is created!
- Now remember your ARN. We will need it in Step 2.21.

The screenshot shows the AWS Lambda console for the function 'myColorSkill'. At the top, the breadcrumb navigation is 'Lambda > Functions > myColorSkill'. The function's ARN, 'arn:aws:lambda:us-east-1:293726142454:function:myColorSkill', is displayed in a red-bordered box. Below the function name, there are buttons for 'Throttle', 'Qualifiers', 'Actions', 'Select a test event..', 'Test', and 'Save'. A green notification bar states: 'Congratulations! Your Lambda function "myColorSkill" has been successfully created and configured with as a trigger in a disabled state. We recommend testing the function behavior before enabling the trigger.' The 'Configuration' tab is selected, showing the 'Designer' section. On the left, a list of triggers includes 'API Gateway', 'AWS IoT', 'Alexa Skills Kit', 'Alexa Smart Home', and 'CloudFront'. The main area shows a diagram with a box for 'myColorSkill' connected to 'Alexa Skills Kit' and 'Amazon CloudWatch Logs'. Below these are dashed boxes for adding more triggers and resources.

Lambda > Functions > myColorSkill

myColorSkill

ARN - arn:aws:lambda:us-east-1:293726142454:function:myColorSkill

Throttle Qualifiers Actions Select a test event.. Test Save

✓ Congratulations! Your Lambda function "myColorSkill" has been successfully created and configured with as a trigger in a disabled state. We recommend testing the function behavior before enabling the trigger.

Configuration Monitoring

▼ Designer

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

API Gateway

AWS IoT

Alexa Skills Kit

Alexa Smart Home

CloudFront

myColorSkill

Alexa Skills Kit

Add triggers from the list on the left

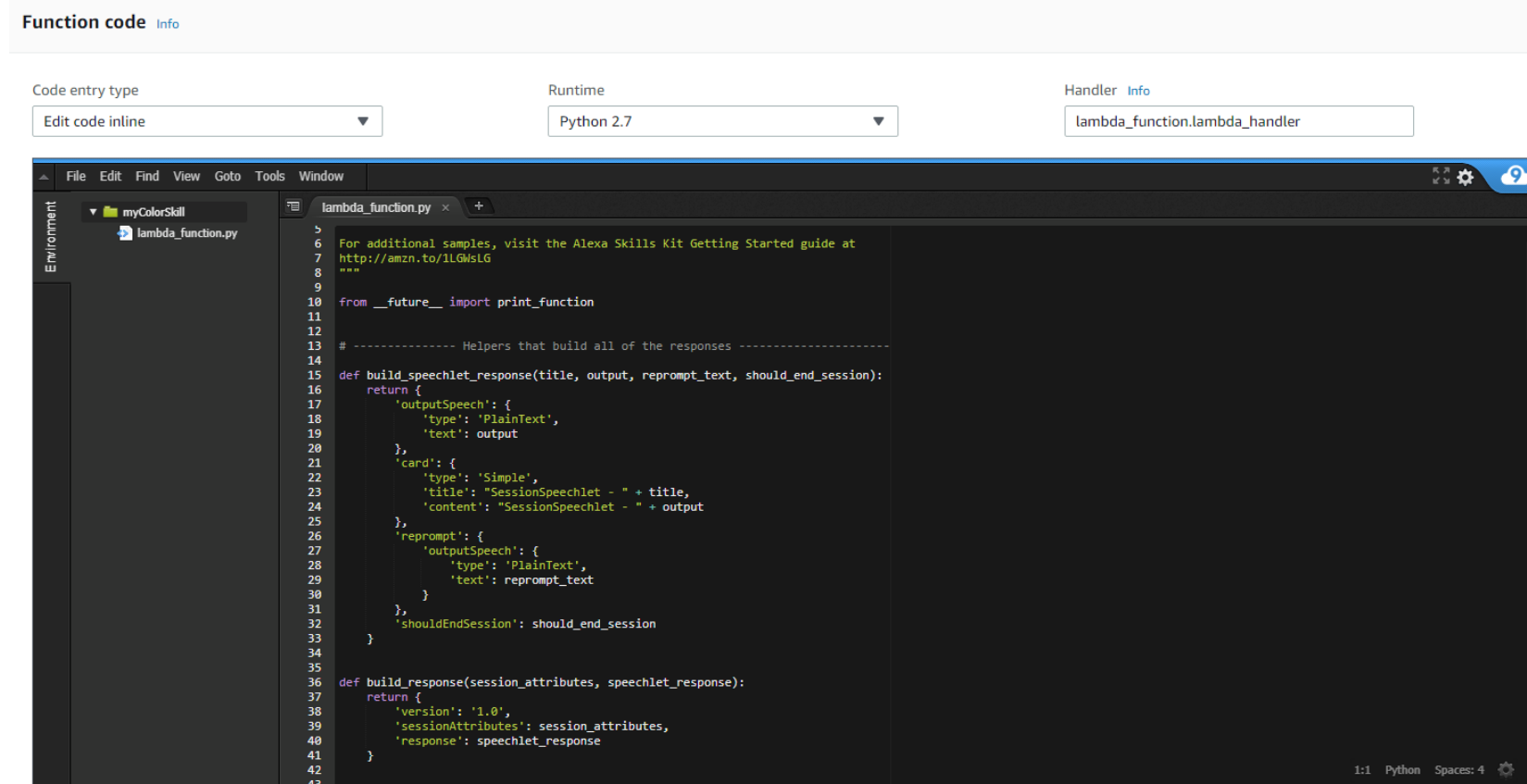
Amazon CloudWatch Logs

Resources the function's role has access to will be shown here

Function code Info

Step 1.14

- Review the **Function Code** section
 - What is the value in **Runtime**?
 - What is the value in **Handler**?



The screenshot displays the AWS Lambda console's 'Function code' section. At the top, there are three dropdown menus: 'Code entry type' set to 'Edit code inline', 'Runtime' set to 'Python 2.7', and 'Handler' set to 'lambda_function.lambda_handler'. Below these, a code editor shows the Python code for the lambda function. The code includes a docstring, imports, and two main functions: 'build_speechlet_response' and 'build_response'. The 'build_speechlet_response' function constructs a JSON response for a speechlet, including output speech, a card, and a reprompt. The 'build_response' function wraps the speechlet response in a standard Lambda response format.

```
>
6 For additional samples, visit the Alexa Skills Kit Getting Started guide at
7 http://amzn.to/1LGWslG
8 """
9
10 from __future__ import print_function
11
12
13 # ----- Helpers that build all of the responses -----
14
15 def build_speechlet_response(title, output, reprompt_text, should_end_session):
16     return {
17         'outputSpeech': {
18             'type': 'PlainText',
19             'text': output
20         },
21         'card': {
22             'type': 'Simple',
23             'title': "SessionSpeechlet - " + title,
24             'content': "SessionSpeechlet - " + output
25         },
26         'reprompt': {
27             'outputSpeech': {
28                 'type': 'PlainText',
29                 'text': reprompt_text
30             },
31         },
32         'shouldEndSession': should_end_session
33     }
34
35
36 def build_response(session_attributes, speechlet_response):
37     return {
38         'version': '1.0',
39         'sessionAttributes': session_attributes,
40         'response': speechlet_response
41     }
42
43
```

Step 1.15

- Review the **Execution Role** section and compare with Step1.9 & 1.10.

Execution role

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

Choose an existing role ▼

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

lambda_basic_execution ▼

Step 1.16

- Review the **Basic settings**
 - What is the **Memory (MB)** value?
 - What is the **Timeout** value?

Basic settings

Description

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

Memory (MB) [Info](#)

Your function is allocated CPU proportional to the memory configured.

128 MB

Timeout [Info](#)

0

min

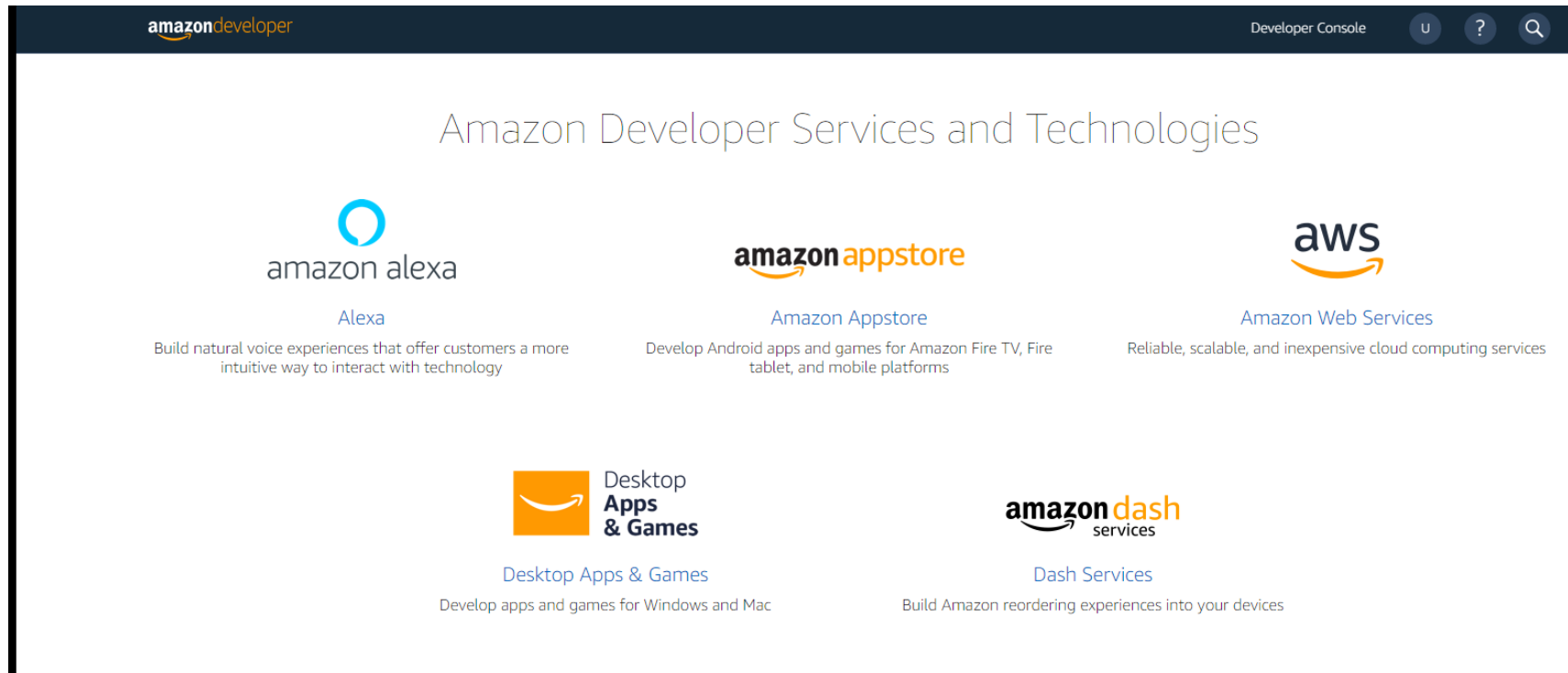
3

sec

Step 2: Create an Alexa Skill

Step 2.1

- Sign in to the [Amazon developer portal](https://developer.amazon.com/). If you haven't done so already, you'll need to create a free account.
 - <https://developer.amazon.com/>



Step 2.2

- Click **Developer Console**



on Developer Services and Technologies



Amazon Appstore

re Develop Android apps and games for Amazon Fire TV, Fire tablet, and mobile platforms



Amazon Web Services

Reliable, scalable, and inexpensive cloud computing services



op Apps & Games

id games for Windows and Mac

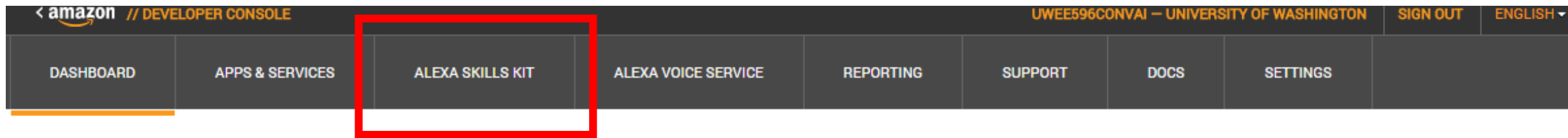


Dash Services

Build Amazon reordering experiences into your devices

Step 2.3

- Go to **ALEXA SKILL KIT**



Notifications

All	Critical
No Notifications.	

Announcements

Version 2 of the Alexa Skills Kit SDK for Java Is Now Available	Mar 29, 2018	The New Alexa Skills Kit Developer Console Is Now Generally Available	Mar 27, 2018
Build Alexa Skills for France Développez des Skills Alexa pour la France	Mar 12, 2018	Announcing the New Alexa Skills Kit Sound Library to Create More Engaging Skills	Mar 2, 2018
Announcing the New Alexa Skills Kit Developer Console (Beta) to Streamline Your Skill Development Process	Feb 15, 2017	AMAZON.YesIntent and AMAZON.NoIntent are Now Compatible with ASK Dialog Management Features	Feb 14, 2018


Step 2.4

- Click **Create Skill**

Welcome to the new Alexa Skills Kit Developer Console

Curious about what's new? [Watch the video overview](#) or [read about what's changed](#).

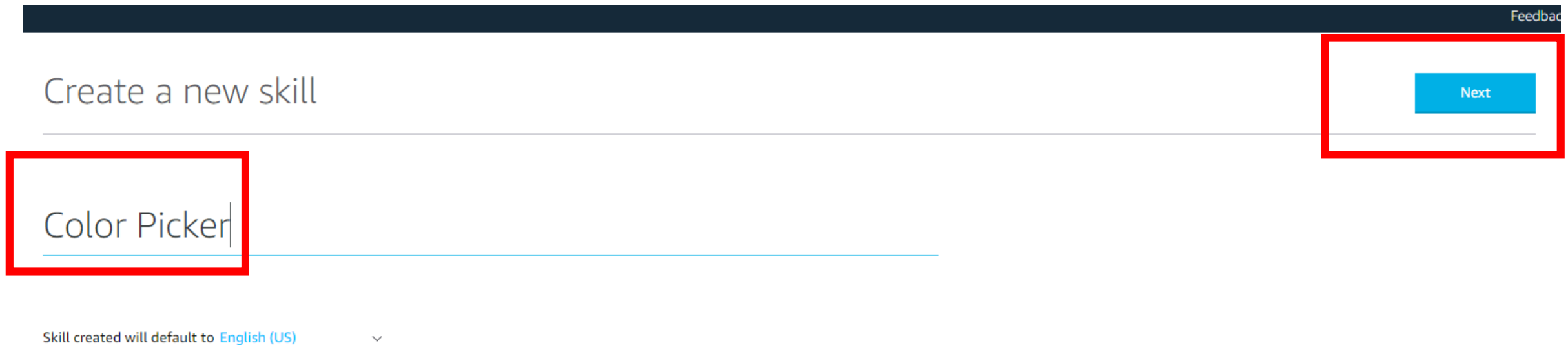
Alexa Skills

SKILL NAME	LANGUAGE	TYPE	MODIFIED	STATUS
<div><h3>Alexa Skills</h3><p>Create your first skill or learn more about Alexa Skills Kit</p><div>Create Skill</div></div>				

Create Skill

Step 2.5

- Name your skill. This is the name displayed to users in the Alexa app. For this example, we'll call it **Color Picker**. Click **Next**.



Feedback

Create a new skill

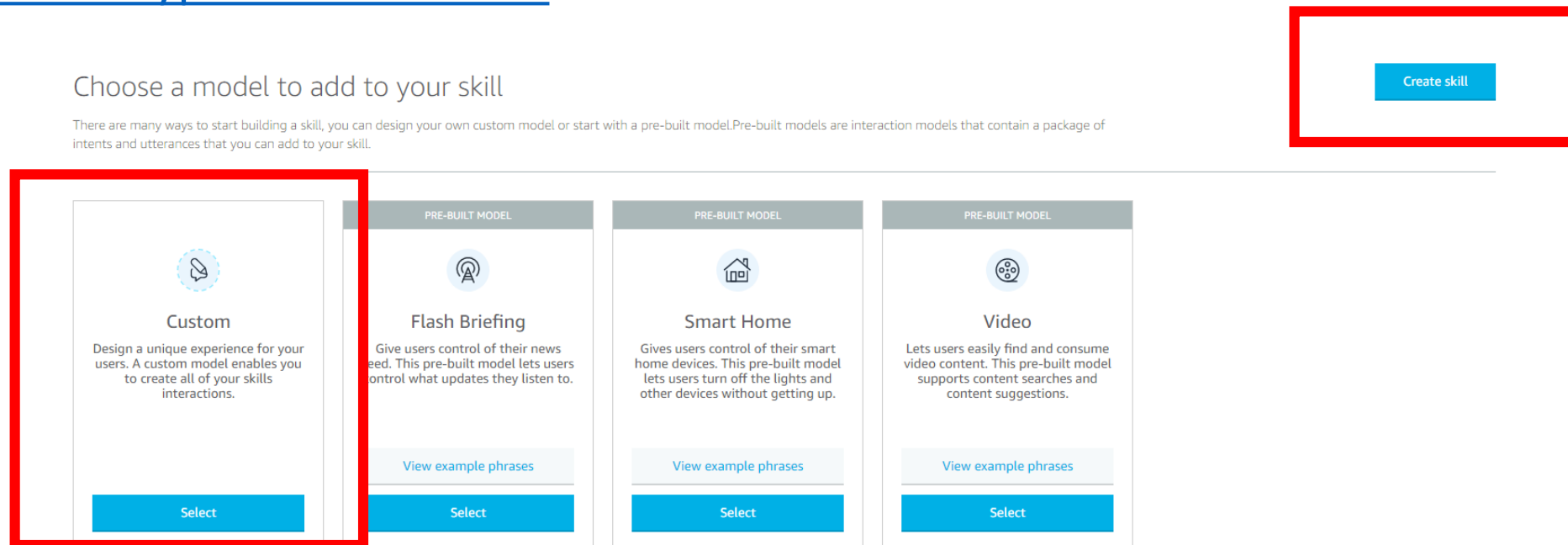
Color Picker

Next

Skill created will default to [English \(US\)](#) ▼

Step 2.6

- Choose **Custom**. Then click **Create skill**.
 - For different skill models, please read the official documents.
 - <https://developer.amazon.com/docs/ask-overviews/understanding-the-different-types-of-skills.html>



Step 2.7

- Check the video **How to get started** for a brief introduction.
 - <https://youtu.be/1pvR4aqwGhg>

amazon alexa

< Your Skills Color Picker Build Test Launch Measure

English (U.S.)

CUSTOM

Interaction Model

Invocation

Intents (3) + Add

Built-In Intents (3)

AMAZON.CancelIntent

AMAZON.HelpIntent

AMAZON.StopIntent

Slot Types (0) + Add

JSON Editor

Interfaces

Endpoint

ACCOUNT LINKING

PERMISSIONS

How to get started

Alexa Skills Kit Developer Console: Build

amazon alexa

Developer Console: Build

Resources

[Documentation](#)

Refer to our technical documents for detailed guides on building custom skills.

[Sample Alexa Projects](#)

Whatever your experience, you can get started quickly using one of our Alexa projects on GitHub.

[Weekly Office Hours](#)

Drop in with your questions and thoughts. We're here to help you.

[Alexa Developer Forums](#)

Visit our forums to get inspired, join our Alexa developer community.

Step 2.8

- Create an invocation name. Let's use the skill name **color picker**. Choose **Save Model** to continue to development for Color Picker.
 - Users will say, "Alexa, open **color picker**" to interact with your skill

The screenshot shows the Amazon Lexa console interface. On the left, a sidebar contains a language dropdown set to 'English (U.S.)', a 'CUSTOM' section, and an 'Interaction Model' section. The 'Invocation' tab is selected and highlighted in blue. Below it, a list of 'Intents (3)' is shown, including 'Built-In Intents (3)' with examples like 'AMAZON.CancelIntent', 'AMAZON.HelpIntent', and 'AMAZON.StopIntent'. At the bottom of the sidebar, 'Slot Types (0)' are listed. On the right, the 'Invocation' section is active, displaying a description: 'Users say a skill's invocation name to begin an interaction with a par... For example, if the invocation name is "daily horoscopes", users can s...'. Below this, a sample user input is shown: 'User: Alexa, ask daily horoscopes for the horoscope for Gemini'. The 'Skill Invocation Name' field is highlighted with a red box and contains the text 'color picker'. Above this field, the 'Save Model' button is also highlighted with a red box.

Step 2.9

- Review the three default intents. What are they?
 - Check the documents
 - <https://developer.amazon.com/docs/custom-skills/standard-built-in-intents.html>

The screenshot shows the Amazon Lex console interface. The top navigation bar includes tabs for < Your Skills, Color Picker, Build (selected), Test, Launch, and Measure. Below the navigation bar, there's a language selector set to 'English (U.S.)' and buttons for 'Save Model' and 'Build Model'. The left sidebar contains a tree view with 'CUSTOM' selected, showing 'Interaction Model', 'Invocation', 'Intents (3)' (highlighted with a red box), 'Slot Types (0)', 'JSON Editor', 'Interfaces', and 'Endpoint'. The 'Intents (3)' section lists 'Built-In Intents (3)': AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent. The right pane, titled 'Intents', features a '+ Add Intent' button and a table with columns 'NAME' and 'Intents'. The table lists three intents: AMAZON.StopIntent, AMAZON.HelpIntent, and AMAZON.CancelIntent.

NAME
AMAZON.StopIntent
AMAZON.HelpIntent
AMAZON.CancelIntent

Step 2.10

- Let's add a new intent.

The screenshot shows the AWS Lambda console interface for building an Alexa skill. The top navigation bar includes tabs for < Your Skills, Color Picker, Build (selected), Test, Launch, and Measure. Below the navigation bar, there's a language selector set to 'English (U.S.)' and two buttons: 'Save Model' and 'Build Model'. The left sidebar contains a tree view with sections: CUSTOM, Interaction Model, Invocation, Intents (3) (selected), Built-In Intents (3), Slot Types (0), JSON Editor, Interfaces, and Endpoint. The 'Intents (3)' section is expanded, showing a list of built-in intents: AMAZON.CancelIntent, AMAZON.HelpIntent, and AMAZON.StopIntent. A red box highlights the '+ Add' button next to the 'Intents (3)' section. Another red box highlights the '+ Add Intent' button in the 'Intents' section of the main content area. The main content area shows a table of intents with the following data:

NAME
AMAZON.StopIntent
AMAZON.HelpIntent
AMAZON.CancelIntent

Step 2.11

- The first custom intent is **WhatsMyColorIntent**.
 - Case sensitive

Add Intent

An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.

☒ Create custom intent [?]

WhatsMyColorIntent|

Create custom intent

Step 2.12

- Add the following sample utterances one by one.
 - 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

Intents / WhatsMyColorIntent

Sample Utterances (0) ?

what's my favorite color|


Step 2.13


- Let's add a new **Slot Type**.

The screenshot shows the Amazon Lex console interface. At the top, there is a language dropdown set to 'English (U.S.)' and two buttons: 'Save Model' and 'Build Model'. Below this is a sidebar with a 'CUSTOM' header and an 'Interaction Model' icon. The main content area is divided into sections: 'Invocation', 'Intents (4)' (with an '+ Add' button), and 'Built-In Intents (3)'. Under 'Built-In Intents', there are three entries: 'AMAZON.CancelIntent', 'AMAZON.HelpIntent', and 'AMAZON.StopIntent'. At the bottom of this section, there is a blue button labeled 'Slot Types (0)' with a '+ Add' icon, which is highlighted by a red rectangular box. Below this button is a 'JSON Editor' section. On the right side of the console, there is a 'Slot Types' section with a '+ Add Slot Type' button and a table with a single header row labeled 'NAME'.

Step 2.14


- Name the **Slot Type** as LIST_OF_COLORS.

 Save Model









 Build Model

Add Slot Type

Slot types define how data in an intent slot is recognized and handled. All intent slots must be assigned a slot type.

☒ Create custom slot type 


Create custom slot type


       

Step 2.15


- Add the following **Slot Values**.

- green
- red
- blue
- orange
- gold
- silver
- yellow
- black
- white

 Save Model


 Build Model

Slot Types / LIST_OF_COLORS

Slot Values (0) 

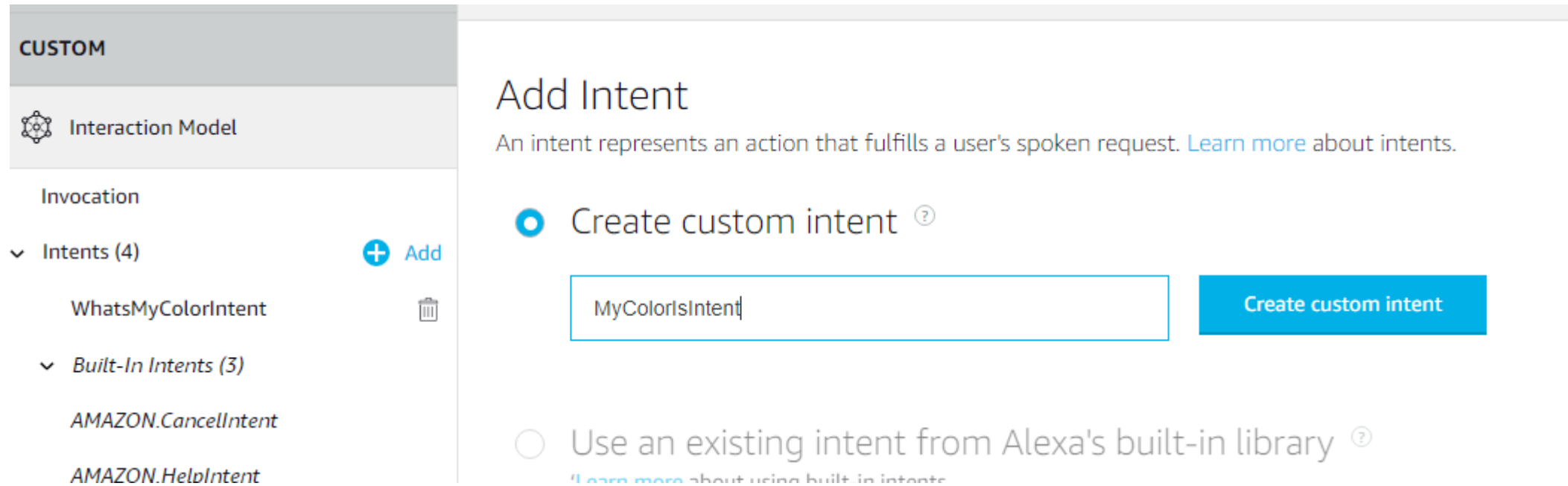
Search

green



Step 2.16

- Add a second custom intent **MyColorsIntent**



The screenshot shows the 'Add Intent' page in the Amazon Alexa Developer Console. On the left is a sidebar with a 'CUSTOM' header and an 'Interaction Model' icon. Below this are sections for 'Invocation' and 'Intents (4)'. The 'Intents (4)' section includes a '+ Add' button and a list of intents: 'WhatsMyColorIntent' (with a trash icon), and three built-in intents: 'AMAZON.CancelIntent', 'AMAZON.HelpIntent', and 'AMAZON.NavigateToMusicIntent'. The main content area is titled 'Add Intent' and includes a description: 'An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.' There are two radio button options. The first, 'Create custom intent', is selected and has a help icon. It features a text input field containing 'MyColorsIntent' and a blue 'Create custom intent' button. The second option, 'Use an existing intent from Alexa's built-in library', is unselected and also has a help icon. Below it is a link: '[Learn more](#) about using built-in intents'.

CUSTOM

Interaction Model

Invocation

Intents (4) + Add

- WhatsMyColorIntent 🗑️
- Built-In Intents (3)*
- AMAZON.CancelIntent
- AMAZON.HelpIntent
- AMAZON.NavigateToMusicIntent

Add Intent

An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.

☒ Create custom intent ?

MyColorsIntent Create custom intent

☐ Use an existing intent from Alexa's built-in library ?

[Learn more](#) about using built-in intents

Step 2.17

- Add the following sample utterance.
 - my favorite color is {Color}
- What the difference between the following two utterances?
 - my favorite color is Color
 - my favorite color is {Color}

[Intents](#) / [MyColorIsIntent](#)

Sample Utterances (0) ?

my favorite color is {Color}

Select an Existing Slot

No existing slots

OR

Create a new slot



Color

Add

Step 2.18

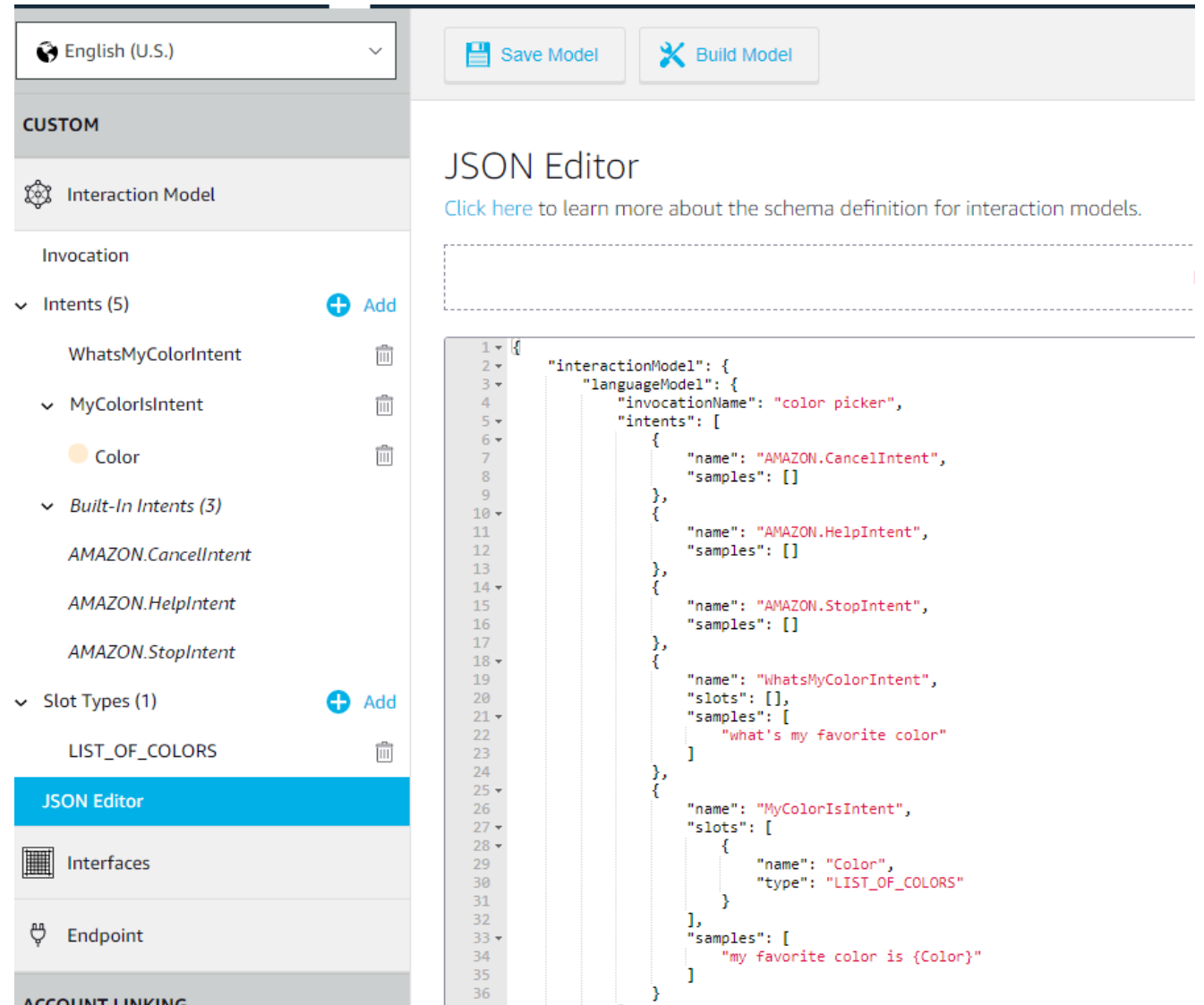
- Choose the **Slot Type** for the new Slot **Color**.
 - Use **LIST_OF_COLORS** which we created in Step 2.13-2.15.

Intent Slots (1) ?

ORDER ?	NAME ?	SLOT TYPE ?	ACTIONS
 1	 Color	<div>Select a slot type</div>	Edit Dialog Delete
2	Create a new slot	<div>+</div> <div>Select a slot type</div>	Edit Dialog Delete

Step 2.19

- Review the resulting JSON object of the **Interaction Model**.
- Can you understand what the JSON object represents?

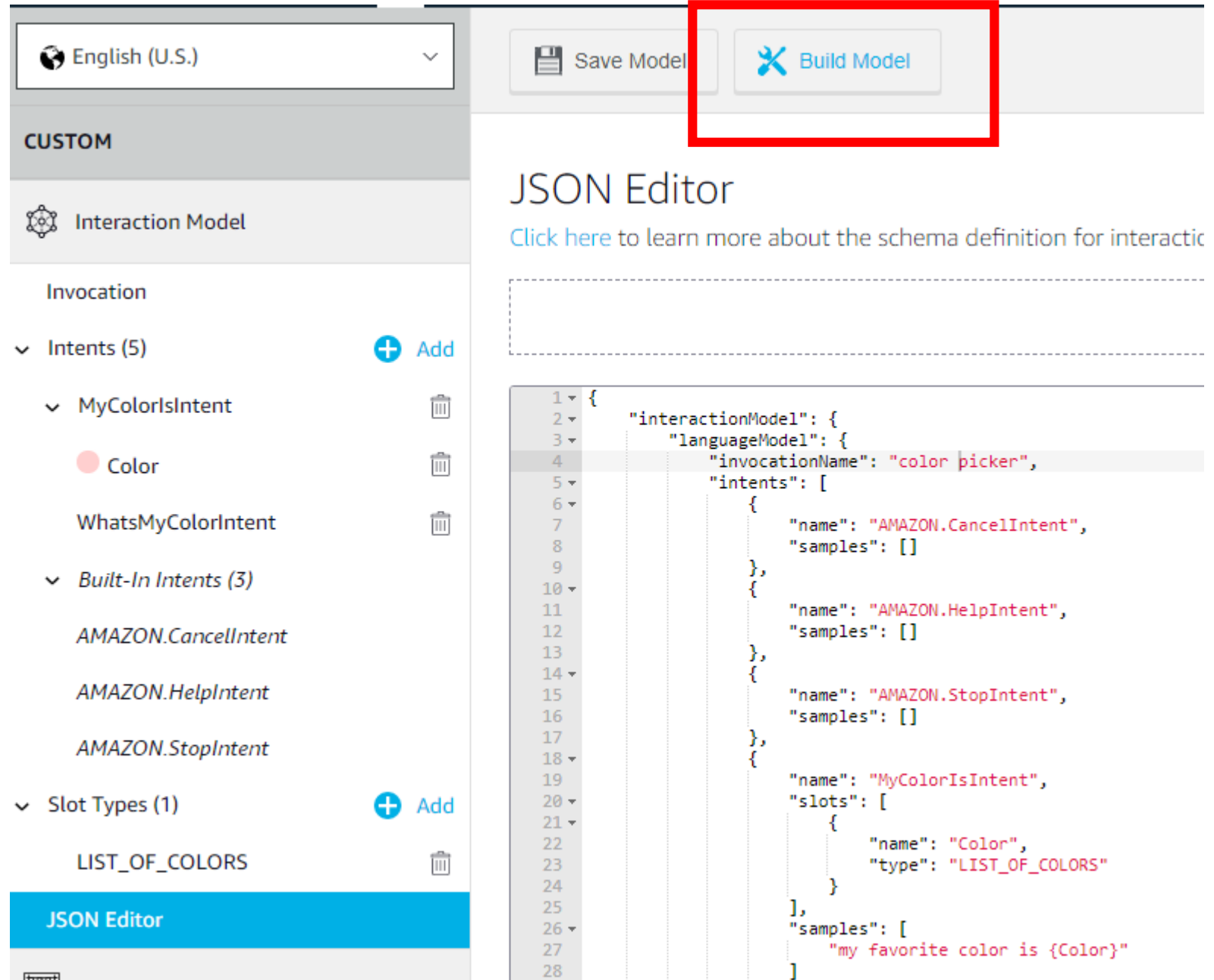


The screenshot displays the AWS Lambda console interface for configuring an Interaction Model. The top bar shows the language set to "English (U.S.)" and buttons for "Save Model" and "Build Model". The left sidebar contains a navigation menu with options: "CUSTOM", "Interaction Model" (selected), "Invocation", "Intents (5)" (with a sub-list of "WhatsMyColorIntent", "MyColorIsIntent" (containing "Color"), and "Built-In Intents (3)" (containing "AMAZON.CancelIntent", "AMAZON.HelpIntent", and "AMAZON.StopIntent"), "Slot Types (1)" (containing "LIST_OF_COLORS"), "JSON Editor" (highlighted in blue), "Interfaces", and "Endpoint". The main area is titled "JSON Editor" and includes a link to learn more about the schema definition. The JSON code is as follows:

```
1 {  
2   "interactionModel": {  
3     "languageModel": {  
4       "invocationName": "color picker",  
5       "intents": [  
6         {  
7           "name": "AMAZON.CancelIntent",  
8           "samples": []  
9         },  
10        {  
11          "name": "AMAZON.HelpIntent",  
12          "samples": []  
13        },  
14        {  
15          "name": "AMAZON.StopIntent",  
16          "samples": []  
17        },  
18        {  
19          "name": "WhatsMyColorIntent",  
20          "slots": [],  
21          "samples": [  
22            "what's my favorite color"  
23          ]  
24        },  
25        {  
26          "name": "MyColorIsIntent",  
27          "slots": [  
28            {  
29              "name": "Color",  
30              "type": "LIST_OF_COLORS"  
31            }  
32          ],  
33          "samples": [  
34            "my favorite color is {Color}"  
35          ]  
36        }  
37      ]  
38    }  
39  }  
40 }
```

Step 2.20

- Click **Build Model**.



The screenshot displays the Amazon Lex console interface. At the top, a language dropdown is set to 'English (U.S.)'. Below it, the 'CUSTOM' tab is selected, showing the 'Interaction Model' configuration. The left sidebar lists the model's components: 'Invocation', 'Intents (5)', and 'Slot Types (1)'. Under 'Intents', 'MyColorIsIntent' is expanded, showing a 'Color' slot. Under 'Slot Types', 'LIST_OF_COLORS' is listed. The 'JSON Editor' tab is active at the bottom of the sidebar. In the main area, the 'JSON Editor' title is visible, followed by a link to learn more about schema definitions. The JSON code is displayed with line numbers 1 through 28. The 'Build Model' button in the top right corner is highlighted with a red rectangle.

English (U.S.)

Save Model Build Model

CUSTOM

Interaction Model

Invocation

Intents (5) + Add

- MyColorIsIntent
 - Color
 - WhatsMyColorIntent
- Built-In Intents (3)
 - AMAZON.CancelIntent
 - AMAZON.HelpIntent
 - AMAZON.StopIntent
- Slot Types (1) + Add
 - LIST_OF_COLORS

JSON Editor

JSON Editor

[Click here](#) to learn more about the schema definition for interactive

```
1 {
2   "interactionModel": {
3     "languageModel": {
4       "invocationName": "color picker",
5       "intents": [
6         {
7           "name": "AMAZON.CancelIntent",
8           "samples": []
9         },
10        {
11          "name": "AMAZON.HelpIntent",
12          "samples": []
13        },
14        {
15          "name": "AMAZON.StopIntent",
16          "samples": []
17        },
18        {
19          "name": "MyColorIsIntent",
20          "slots": [
21            {
22              "name": "Color",
23              "type": "LIST_OF_COLORS"
24            }
25          ],
26          "samples": [
27            "my favorite color is {Color}"
28          ]
29        }
30      ]
31    }
32  }
```


Step 2.21

- Configure the Endpoint.
Copy ARN you found in Step 1.13 to Default Region. Click Save Endpoint.

English (U.S.)

Save Endpoints

Endpoint

The Endpoint will receive POST requests when a user interacts with your Alexa Skill. The request body contains parameters that your service can use to perform logic and generate a formatted response. Learn more about Lambda endpoints [here](#). You can host your own HTTPS web service endpoint as long as the service meets the requirements described in the [AWS Lambda documentation](#).

Service Endpoint Type

Select how you will host your skill's service endpoint.

☒ AWS Lambda ARN (Recommended)

Your Skill ID: `amzn1.ask.skill.d52b91fc-96c8-4a4c-802e-bf2b37bf9fa0` [Copy to Clipboard](#)

Default Region (Required): `us-east-1`

ARN: `arn:aws:lambda:us-east-1:293726142454:function:myColorSkill`

North America (Optional): `arn:aws:lambda:us-east-1:<aws_account_id>:function:<lambda_name>`

Europe and India (Optional): `arn:aws:lambda:eu-west-1:<aws_account_id>:function:<lambda_name>`

Save Failed
Error parsing the requested parameters. The request contains invalid values for the parameters. Error code: SkillManifestException

Step 2.22

- Review the Endpoint page. What is your Skill ID?

CUSTOM

- Interaction Model
- Invocation
- Intents (5) [+ Add](#)
 - MyColorIntent
 - Color
 - WhatsMyColorIntent
 - Built-In Intents (3)
 - AMAZON.CancelIntent
 - AMAZON.HelpIntent
 - AMAZON.StopIntent
- Slot Types (1) [+ Add](#)
 - LIST_OF_COLORS
- JSON Editor
- Interfaces
- Endpoint**

Endpoint

The Endpoint will receive POST requests when a user interacts with your Alexa Skill. The request body contains parameters that your service can use to provide a formatted response. [Learn more about Lambda endpoints here.](#) You can host your own HTTPS web service endpoint as long as the service meets the requirements.

Service Endpoint Type

Select how you will host your skill's service endpoint.

- ☒ AWS Lambda ARN [?](#)
(Recommended)

Your Skill ID [?](#)

amzn1.ask.skill.d52b91fc-96c8-4a4c-802e-bf2b37bf9fa0 [Copy to clipboard](#)

Default Region [?](#)
(Required)

arn:aws:lambda:us-east-1:293726142454:function:myColorSkill

North America [?](#)
(Optional)

arn:aws:lambda:us-east-1:<aws_account_id>:function:<lambda_name>

Europe and India [?](#)
(Optional)

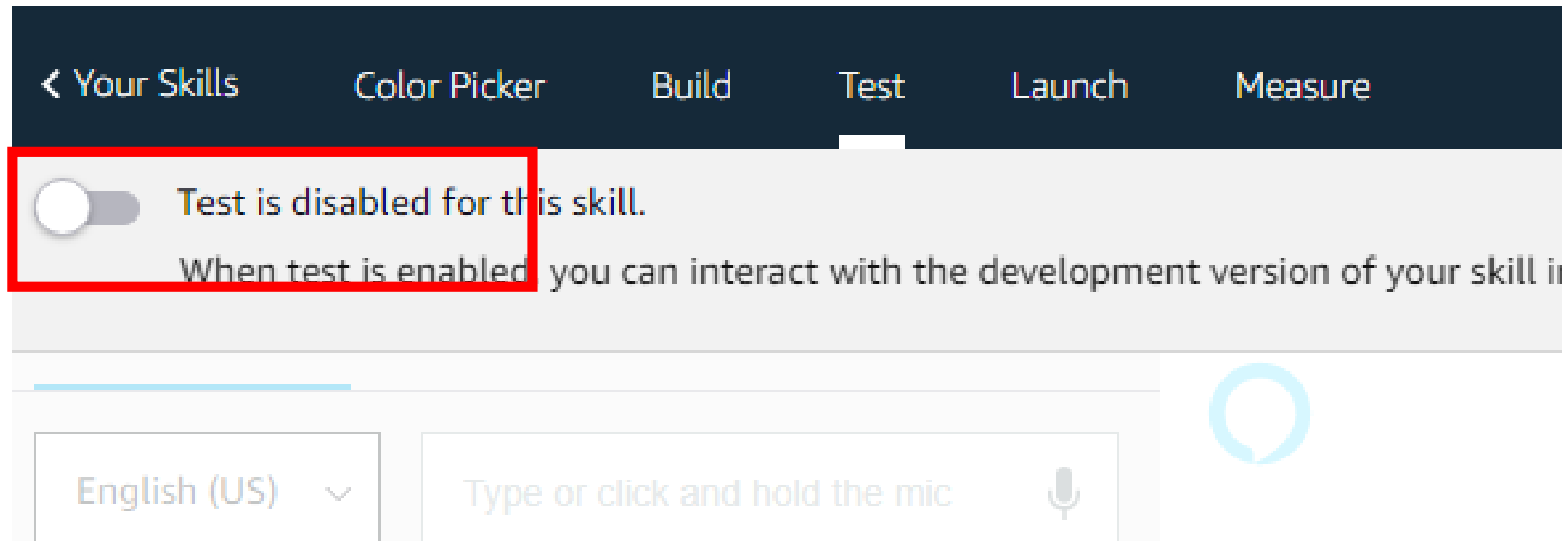
arn:aws:lambda:eu-west-1:<aws_account_id>:function:<lambda_name>

[Build Success](#)
If you make a

Step 3: Test your skill using the
Service Simulator

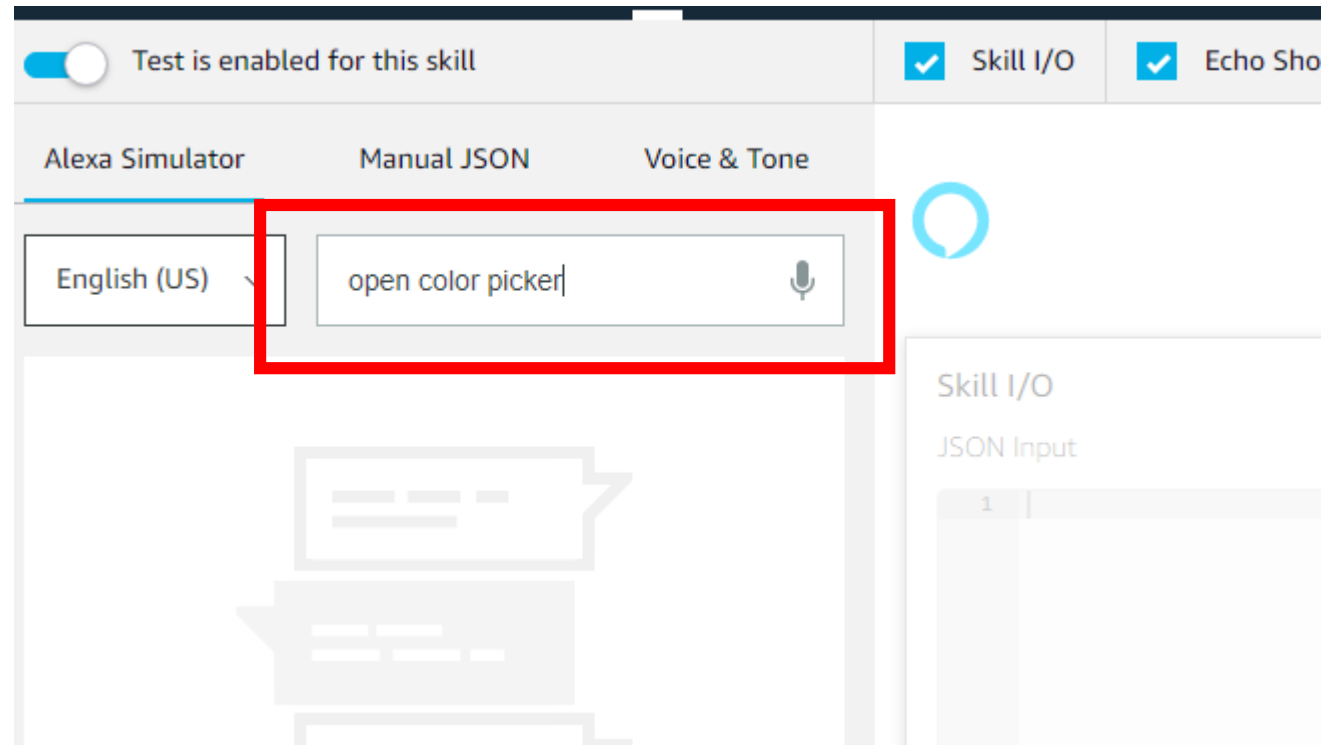
Step 3.1

- Go to the **Test** tab and Enable test.



Step 3.2

- Type “open color picker”



Step 3.3

- Reproduce the sample dialog.

Alexa Simulator

open color picker

Welcome to the Alexa Skills Kit sample. Please tell me your favorite color by saying, my favorite color is red

how are you

I'm not sure what your favorite color is. You can say, my favorite color is red.

my favorite color is red

I now know your favorite color is red. You can ask me your favorite color by saying, what's my favorite color?

what is my favorite color

Your favorite color is red. Goodbye.

Step 3.4

- Click the bot utterance and analyze the **JSON input** and **JSON output**.

The screenshot displays the Alexa Skills Kit (ASK) simulator interface. At the top, there are tabs for 'Alexa Simulator', 'Manual JSON', and 'Voice & Tone'. Below these, a language selector is set to 'English (US)'. The main chat area shows a sequence of interactions: a blue bot utterance, a grey user utterance, a white system response, another grey user utterance, and a final white system response. A red box highlights the first blue bot utterance. To the right of the chat, there are two panels: 'Skill I/O' and 'JSON Output', both also highlighted with red boxes. The 'Skill I/O' panel shows the 'JSON Input' for the selected utterance, and the 'JSON Output' panel shows the corresponding 'JSON Output'.

Test is enabled for this skill

☑ Skill I/O ☑ Echo Show Display ☐ Device Log

Alexa Simulator Manual JSON Voice & Tone

English (US) Type or click and hold the mic

open color picker

Welcome to the Alexa Skills Kit sample. Please tell me your favorite color by saying, my favorite color is red

my favorite color is red

I now know your favorite color is red. You can ask me your favorite color by saying, what's my favorite color?

what is my favorite color

Your favorite color is red. Goodbye.

Skill I/O

JSON Input

```
1 {
2   "version": "1.0",
3   "session": {
4     "new": true,
5     "sessionId": "amzn1.echo-api.session.3a2f47ec-d5ad-4b8c-a3d3-0ee1b3d",
6     "application": {
7       "applicationId": "amzn1.ask.skill.d52b91fc-96c8-4a4c-802e-bf2b37"
8     },
9     "user": {
10      "userId": "amzn1.ask.account.AEKNNPQKK7HX224WTJBL2ME5PDFCIPKWT7Z"
11    }
12  },
13  "context": {
14    "AudioPlayer": {
15      "playerActivity": "IDLE"
16    },
17    "Display": {},
18    "System": {
19      "application": {
20        "applicationId": "amzn1.ask.skill.d52b91fc-96c8-4a4c-802e-bf"
21      },
22      "user": {
23        "userId": "amzn1.ask.account.AEKNNPQKK7HX224WTJBL2ME5PDFCIPK"
24      },
25      "device": {
26        "deviceId": "amzn1.ask.device.AF400E55VVSYSYCYRD235XZW62K6A07"
27      },
28      "supportedInterfaces": {
29        "AudioPlayer": {},
30        "Display": {
31          "templateVersion": "1.0",
32          "markupVersion": "1.0"
33        }
34      }
35    }
36  }
37 }
```

JSON Output

```
1 {
2   "body": {
3     "version": "1.0",
4     "response": {
5       "outputSpeech": {
6         "type": "PlainText",
7         "text": "Welcome to the Alexa Skills Kit sample. Please tell me"
8       },
9       "card": {
10        "type": "Simple",
11        "title": "SessionSpeechlet - Welcome",
12        "content": "SessionSpeechlet - Welcome to the Alexa Skills Kit"
13      },
14       "reprompt": {
15        "outputSpeech": {
16          "type": "PlainText",
17          "text": "Please tell me your favorite color by saying, my"
18        }
19      },
20       "shouldEndSession": false
21    },
22    "sessionAttributes": {}
23  }
24 }
```

Task 2 & 3

Please follow instructions in https://github.com/hao-fang/ee596_spr2018_lab1