Alexa
Alexa Skills Kit
Alexa Voice Service
Alexa Fund

# Creating an AWS Lambda Function for a Custom Skill



## Introduction

The easiest way to build the cloud-based service for a custom Alexa skill is by using AWS Lambda, an Amazon Web Services offering that runs your code only when it's needed and scales automatically, so there is no need to provision or continuously run servers. You upload the code for your Alexa skill to a Lambda function and Lambda does the rest, executing it in response to Alexa voice interactions and automatically managing the compute resources for you.

## About Lambda Functions and Custom Skills

Using a Lambda function for your service eliminates some of the complexity around setting up and managing your own endpoint:

- You do not need to administer or manage any of the compute resources for your service.
- You do not need an SSL certificate.
- You do not need to verify that requests are coming from the Alexa service yourself. Access to execute your function is controlled by permissions within AWS instead.
- AWS Lambda runs your code only when you need it and scales with your usage, so there is no need to provision or continuously run servers.
- Alexa encrypts its communications with Lambda utilizing TLS. See also AWS security best practices.
- For most developers, the Lambda free tier is sufficient for the function supporting an Alexa skill. The first one million requests each month are free. Note that the Lambda free tier does not automatically expire, but is available indefinitely.

AWS Lambda supports code written in Node.js (JavaScript), Java, and Python. You can copy and edit JavaScript or Python code in the inline code editor in the AWS Lambda console or upload it in a zip file. For basic testing, you can invoke your function manually by sending it JSON requests in the Lambda console

**Alexa Skills Kit - Custom Skills**

- Understanding Custom Skills
- Get Started with Sample Code
- Voice Design
- Setting Up
- Developing
- Displaying Skill Cards
- Testing
- Publishing
- Reference

**Other Resources**

- Getting Started
- Smart Home Skills
- Flash Briefing Skills
- Glossary
- Alexa Skills Kit Forum
- Alexa Voice Service
- Alexa Fund

Web2PDF
converted by Web2PDFConvert.com

in the Lambda console.

For an overview of AWS Lambda, see What Is AWS Lambda?

The Lambda functions must adhere to the same service interface and handle the three types of requests sent by Alexa. For details about handling Alexa requests, see Handling Requests Sent by Alexa. For details about the JSON interface, see the JSON Interface Reference for Custom Skills.

> **Note:** Lambda functions for Alexa skills can be hosted in either the **US East (N. Virginia)** or **EU (Ireland)** region. These are the only regions the Alexa Skills Kit supports.

This document covers creating a new Lambda function for a *custom skill*. If you are using the Smart Home Skill API, you use Lambda to create a *skill adapter*. See Steps to Create an Alexa Smart Home Skill.

## Creating a Lambda Function for an Alexa Skill

You use the Lambda console to create a new Lambda function and do basic testing. You can use a *blueprint* to create a new function with code for a basic function that implements a simple skill in Node.js or Python. Alternatively, skip the blueprint step to create a Lambda function from scratch.

If you want to code your service in Java, but haven't yet set up a Java project, you may want to use one of the samples as a starting point for your function.

To create a new Lambda function:

1. If you do not already have an account on AWS, go to Amazon Web Services and create an account.
2. Log in to the AWS Management Console and navigate to AWS Lambda.
3. Click the region drop-down in the upper-right corner of the console and select either **US East (N. Virginia)** or **EU (Ireland)**.

   Lambda functions for Alexa skills *must* be hosted in either the **US East (N. Virginia)** or **EU (Ireland)** region.
4. If you have no Lambda functions yet, click **Get Started Now**. Otherwise, click **Create a Lambda Function**.
5. To start with sample code in Node.js or Python, select one of the Alexa Skills Kit blueprints:
   - alexa-skill-kit-sdk-factskill
   - alexa-skill-kit-sdk-triviaskill
   - alexa-skills-kit-color-expert
   - alexa-skill-kit-sdk-howtoskill
   - alexa-skills-kit-color-expert-python

   > **Tip:** Type 'alexa' in the **Filter** box to filter the list of blueprints.

   You can see more about these samples at the Alexa organization on GitHub.
6. When prompted to configure triggers, click the box and select **Alexa Skills Kit**, then click **Next**.
7. Enter a **Name** and **Description** for the function.
8. Select the language you want to use for the **Runtime** (Node.js, Java, or Python).
   - If you select Java, you need to also upload your Java code in a zip file.
   - Note that you cannot change the language for a

Web2PDF
converted by Web2PDFConvert.com

function once it is saved.

9. Select the **Role** for the function. This defines the AWS resources the function can access.
    - To use an existing role, select the role under **Use existing role**.
    - To create a new role, see Defining a new Role for the Function, below
10. On the **Review** page, make sure that the **Triggers** section includes **Alexa Skills Kit**.
11. Click **Create function** to save your new function.
12. See below to test the function in the console.

When you are ready to add your own code, edit the function and select the **Code** tab. From here, you can do any of the following:

- Write your code directly in the code editor in the Lambda console (*Node.js* or *Python*).
- Write your code offline and copy and paste it into the Lambda console editor (*Node.js* or *Python*).
- Write your code offline and upload it to the Lambda function in a zip file (*Node.js*, *Python*, or *Java*).
- Use the Eclipse IDE and the AWS Toolkit for Eclipse (*Java*). For details, see Using AWS Lambda with the Toolkit for Eclipse.

### Defining a New Role for the Function

The role specifies the AWS resources your function can access. To create a new role while configuring your function:

1. For **Role** (under **Lambda function handler and role**), select **Create new role from template(s)**.
2. Enter the **Role Name**.
3. From the **Policy templates** list, select **Simple Microservice permissions**.

## Test a Lambda Function in the Console

You can manually test a Lambda function in the Lambda console by sending sample JSON events formatted in the same way as requests sent by Alexa. Sample events are provided for testing within the console. You can use these events as a starting point and modify them to represent requests that the Alexa service would send to your own function. Note that you can also send events to a Lambda function using the AWS CLI.

See JSON Interface Reference for Custom Skills for details about the JSON interface.

1. In the Function list, click the function name to open the details for the function.
2. Click **Actions** and then **Configure test event**.
3. From the **Sample event template** list, select one of the sample Alexa requests:
    - Alexa Start Session
    - Alexa Intent - MyColors
    - Alexa End Session

    These three Alexa sample events send requests that correspond to the code provided in the **alexa-skills-kit-color-expert** and **alexa-skills-kit-color-expert-python** blueprints.

    You can send one of these requests as is, or use it as a starting point to test different intent and slot values.
4. Choose **Submit**

Web2PDF
converted by Web2PDFConvert.com

Once you have configured a test event, clicking **Test** sends that same request. To configure a different event, click **Actions** and then **Configure test event**.

> ☑ **Tip:** If you get an error, make sure your test event is a valid Alexa request. Click **Actions** and then **Configure test event**, and then select one of the Alexa events: `Alexa Start Session`, `Alexa Intent - MyColors`, or `Alexa End Session`. Selecting any other sample (such as `Hello World`) will not work.

After the function runs, the **Execution result** section shows the response returned by the function, in JSON format. You should see a response appropriate for the request pasted into the **Sample event** box. For example, for a `LaunchRequest`, the sample returns a response similar to this:

```
{
  "version": "1.0",
  "sessionAttributes": {},
  "response": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "Welcome to the Alexa Skills Kit sample, Please tell
    },
    "card": {
      "type": "Simple",
      "title": "SessionSpeechlet - Welcome",
      "content": "SessionSpeechlet - Welcome to the Alexa Skills K
    },
    "reprompt": {
      "outputSpeech": {
        "type": "PlainText",
        "text": "Please tell me your favorite color by saying, my
      }
    },
    "shouldEndSession": false
  }
}
```

The **Execution logs** section shows any log messages generated by the code. The sample writes a log message for each type of request, so you should see something like the following:

```
2015-05-18T23:53:22.357Z    0f885f98-fdb9-11e4-80af-1b9f8363b496
```

## Configuring the Alexa Skills Kit Trigger

Configuring the Alexa Skills Kit trigger grants Alexa the necessary *invocation permissions* for your function.

If you configured the trigger while creating the function, you can skip these steps.

1. Log in to the AWS Management Console and navigate to AWS Lambda.
2. Click your function in the list to open the configuration details.
3. Select the **Triggers** tab.
4. Click **Add trigger**.
5. Click the outlined box and choose **Alexa Skills Kit**.
6. Click **Submit**.

It is highly recommended that you limit invocation permissions to just Alexa to protect your function from malicious callers.

## Sample Interaction Model for the Color Expert Blueprint

If you created the Lambda function using either the **alexa-skills-kit-color-expert** or **alexa-skills-kit-color-expert-python** blueprint, you may want to

Web2PDF
converted by Web2PDFConvert.com

**expert** or **alexa skills kit color expert python** blueprint, you may want to test the sample skill with either an Alexa-enabled device or the Service Simulator. To do this, you need to register the skill in the developer portal and provide an interaction model for the skill.

A very simple interaction model for the sample is provided below. Copy this information into the **Intent Schema** and **Sample Utterances** boxes on the **Interaction Model** page of the developer portal when you register this new skill.

### Intent Schema

```
{
  "intents": [
    {
      "intent": "MyColorIsIntent",
      "slots": [
        {
          "name": "Color",
          "type": "LIST_OF_COLORS"
        }
      ]
    },
    {
      "intent": "WhatsMyColorIntent"
    },
    {
      "intent": "AMAZON.HelpIntent"
    }
  ]
}
```

### Custom Slot Type

Create a type called `LIST_OF_COLORS` and paste in the following values:

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

### Sample Utterances

```
WhatsMyColorIntent what's my favorite color
WhatsMyColorIntent what is my favorite color
WhatsMyColorIntent what's my color
WhatsMyColorIntent what is my color
WhatsMyColorIntent my color
WhatsMyColorIntent my favorite color
WhatsMyColorIntent get my color
WhatsMyColorIntent get my favorite color
WhatsMyColorIntent give me my favorite color
WhatsMyColorIntent give me my color
WhatsMyColorIntent what my color is
WhatsMyColorIntent what my favorite color is
WhatsMyColorIntent yes
WhatsMyColorIntent yup
WhatsMyColorIntent sure
WhatsMyColorIntent yes please
MyColorIsIntent my favorite color is {Color}
```

## Next Steps

- Next: Handling Requests Sent by Alexa
- Go back to: Registering and Managing Custom Skills in the Developer Portal
- Return to: Steps to Build a Custom Skill

For additional Node.js and Java samples you can use with Lambda, see:

- Using the Alexa Skills Kit Samples (Custom Skills)
- Deploying a Sample Custom Skill to AWS Lambda
- Alexa Skills Kit SDK for Node.js
- Alexa GitHub organization

Web2PDF
converted by Web2PDFConvert.com

SITEMAP

FOLLOW US

**Alexa**
Alexa Skills Kit
Alexa Voice Service
Alexa Fund

**Services & APIs**
Earn
Engage
Build

**Devices**
Fire Tablets
Amazon Fire TV
Dash Replenishment Service
Fire Phone
Amazon Echo
Amazon Tap

**Resources**
Platforms
Learning Center
Development Tools
Promotional Tools
Marketing Tips
Other Resources

**Blog**
Subscribe to the Blog
#Announcements
#AmazonUnderground
#Alexa
#AmazonFireTV
#FireTablets
#How To
#Monetization
#APIs
#Marketing
#Gamedev
#Developer Spotlight

**Support**
Submitting Your Apps
FAQs
Forums
Contact Us
App Distribution Agreement
Mobile Ad Network Publisher Agreement
Mobile Ad Network Program Participation Requirements
Advertise Your App With Amazon Agreement
Program Materials License Agreement
Trademark Guidelines
Terms of Use

© 2010-2016, Amazon.com, Inc. or its affiliates. All Rights Reserved.

Web2PDF
converted by Web2PDFConvert.com