

Getting started with Lambda

[PDF \(lambda-dg.pdf#getting-started\)](#) | [RSS \(lambda-updates.rss\)](#)

To get started with Lambda, use the Lambda console to create a function. In a few minutes, you can create a function, invoke it, and then view logs, metrics, and trace data.

Note

To use Lambda and other AWS services, you need an AWS account. If you don't have an account, visit aws.amazon.com and choose **Create an AWS Account**. For instructions, see [How do I create and activate a new AWS account?](http://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/)

As a best practice, create an AWS Identity and Access Management (IAM) user with administrator permissions, and then use that IAM user for all work that does not require root credentials. Create a password for console access, and create access keys to use command line tools. For instructions, see [Creating your first IAM admin user and group](https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html) in the *IAM User Guide*.

You can author functions in the Lambda console, or with an IDE toolkit, command line tools, or the AWS SDKs. The Lambda console provides a [code editor](#) for non-compiled languages that lets you modify and test code quickly. The [AWS Command Line Interface \(AWS CLI\)](#) gives you direct access to the Lambda API for advanced configuration and automation use cases.

You deploy your function code to Lambda using a deployment package. Lambda supports two types of deployment packages:

- A [.zip file archive](#) that contains your function code and its dependencies. For a tutorial, see [Create a Lambda function with the console](#).
- A [container image](#) that is compatible with the [Open Container Initiative \(OCI\)](#) specification.

Create a Lambda function with the console

In this getting started exercise, you create a Lambda function using the console. The function uses the default code that Lambda creates. The Lambda console provides a [code editor](#) ([./foundation-console.html#code-editor](#)) for non-compiled languages that lets you modify and test code quickly. For compiled languages, you must create a [.zip archive deployment package](#) ([./gettingstarted-package.html#gettingstarted-package-zip](#)) to upload your Lambda function code.

Create the function

You create a Node.js Lambda function using the Lambda console. Lambda automatically creates default code for the function.

To create a Lambda function with the console

1. Open the [Functions page](#) <https://console.aws.amazon.com/lambda/home#/functions> of the Lambda console.
2. Choose **Create function**.
3. Under **Basic information**, do the following:
 - a. For **Function name**, enter **my-function**.
 - b. For **Runtime**, confirm that **Node.js 14.x** is selected. Note that Lambda provides runtimes for .NET (PowerShell, C#), Go, Java, Node.js, Python, and Ruby.
4. Choose **Create function**.

Lambda creates a Node.js function and an [execution role](#) ([./lambda-intro-execution-role.html](#)) that grants the function permission to upload logs. The Lambda function assumes the execution role when you invoke your function, and uses the execution role to create credentials for the AWS SDK and to read data from event sources.

Invoke the Lambda function

Invoke your Lambda function using the sample event data provided in the console.

To invoke a function

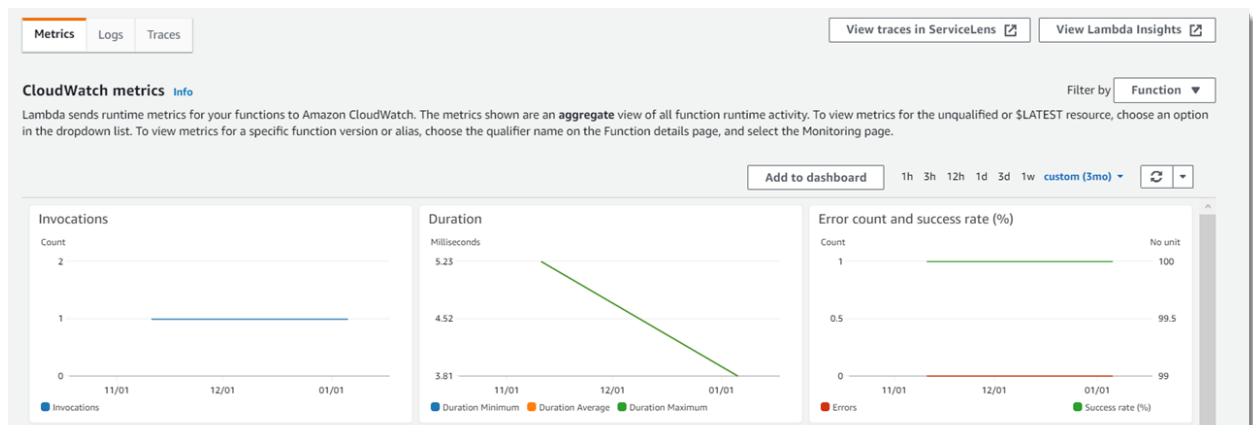
1. After selecting your function, choose the **Test** tab.
2. In the **Test event** section, choose **New event**. In **Template**, leave the default **hello-world** option. Enter a **Name** for this test and note the following sample event template:

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

3. Choose **Save changes**, and then choose **Test**. Each user can create up to 10 test events per function. Those test events are not available to other users.

Lambda runs your function on your behalf. The function handler receives and then processes the sample event.

4. Upon successful completion, view the results in the console.
 - The **Execution result** shows the execution status as **succeeded**. To view the function execution results, expand **Details**. Note that the **logs** link opens the **Log groups** page in the CloudWatch console.
 - The **Summary** section shows the key information reported in the **Log output** section (the *REPORT* line in the execution log).
 - The **Log output** section shows the log that Lambda generates for each invocation. The function writes these logs to CloudWatch. The Lambda console shows these logs for your convenience. Choose **Click here** to add logs to the CloudWatch log group and open the **Log groups** page in the CloudWatch console.
5. Run the function (choose **Test**) a few more times to gather some metrics that you can view in the next step.
6. Choose the **Monitor** tab. This page shows graphs for the metrics that Lambda sends to CloudWatch.



For more information on these graphs, see [Monitoring functions on the Lambda console](#) ([./monitoring-functions-access-metrics.html](#)) .

Clean up


If you are done working with the example function, delete it. You can also delete the log group that stores the function's logs, and the execution role that the console created.

To delete a Lambda function


1. Open the [Functions page](#) (<https://console.aws.amazon.com/lambda/home#/functions>) of the Lambda console.

2. Choose a function.
3. Choose **Actions, Delete**.
4. In the **Delete function** dialog box, choose **Delete**.

To delete the log group

1. Open the [Log groups page](https://console.aws.amazon.com/cloudwatch/home#logs:)  (<https://console.aws.amazon.com/cloudwatch/home#logs:>) of the CloudWatch console.
2. Select the function's log group (`/aws/lambda/my-function`).
3. Choose **Actions, Delete log group(s)**.
4. In the **Delete log group(s)** dialog box, choose **Delete**.

To delete the execution role

1. Open the [Roles page](https://console.aws.amazon.com/iam/home?#/roles)  (<https://console.aws.amazon.com/iam/home?#/roles>) of the AWS Identity and Access Management (IAM) console.
2. Select the function's role (`my-function-role-31exxmpl`).
3. Choose **Delete role**.
4. In the **Delete role** dialog box, choose **Yes, delete**.

You can automate the creation and cleanup of functions, log groups, and roles with AWS CloudFormation and the AWS Command Line Interface (AWS CLI). For fully functional sample applications, see [Lambda sample applications](#) ([./lambda-samples.html](#)) .