

Creating a Lambda function, versioning and aliases

## **Creating a Python Lambda Function**

You can run Python code in AWS Lambda. Lambda provides runtimes for Python that execute your code to process events. Your code runs in an Amazon Linux environment that includes AWS credentials from an AWS Identity and Access Management (IAM) role that you manage.

Read more about Python Lambda Functions at:

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-python.html>

Your Lambda function's handler is the method in your function code that processes events. When your function is invoked, Lambda runs the handler method. When the handler exits or returns a response, it becomes available to handle another event. Read about handlers here:

<https://docs.aws.amazon.com/lambda/latest/dg/python-handler.html>

To create a Python Lambda Function you need to write your code and then package it up into a deployment package. A deployment package is a ZIP archive that contains your function code and dependencies. You can upload the package directly to Lambda, or you can use an Amazon S3 bucket, and then upload it to Lambda. If the deployment package is larger than 50 MB, you must use Amazon S3.

Read about how to create a deployment package for a Python Lambda Function at:

<https://docs.aws.amazon.com/lambda/latest/dg/python-package.html>

Read about how to create an AWS Lambda function with the AWS Toolkit for Pycharm at:

<https://docs.aws.amazon.com/toolkit-for-jetbrains/latest/userguide/create-new-lambda.html>

Read about running and debugging an AWS Lambda function with the AWS Toolkit for Pycharm at:

<https://docs.aws.amazon.com/toolkit-for-jetbrains/latest/userguide/invoke-lambda.html>

## **Versioning and Aliases**

AWS Lambda versions and aliases can help to provide a lot of control when managing your function code. Make sure you understand their base concepts to best utilize these features.

You can publish a new version of your AWS Lambda function when you create new or update existing functions. Each version of a lambda function gets its own unique Amazon Resource Name (ARN). You can then use these ARNs to use different versions of the Lambda function for different purposes.

You also have the ability to create aliases for AWS Lambda functions. Aliases are essentially pointers to one specific Lambda version.

Each alias you create has a unique ARN. An alias can only point to one function version, not to another alias. You can update an alias to point to a new version of the function.

This is handy for a number of reasons, one being the following: consider event sources such as Amazon S3 which can be configured to invoke your Lambda function. These event sources maintain a mapping that identifies the function to invoke when events occur. If you specify a Lambda function alias in the mapping configuration, you don't need to update the mapping when the function version changes.

Read about versioning here: <https://docs.aws.amazon.com/lambda/latest/dg/configuration-versions.html>

Read about aliases here: <https://docs.aws.amazon.com/lambda/latest/dg/configuration-aliases.html>