* Understanding AWS Lambda

AWS Lambda is a serverless computing service that allows you to run code without provisioning or managing servers. It automatically scales your applications by running code in response to events, such as changes in data or system state, and manages the compute resources for you. This document provides an overview of AWS Lambda, its features, use cases, and how to get started.

* **What are Lambda Functions?**

AWS lambda are server-less compute functions are fully managed by the AWS where developers can run their code without worrying about servers. AWS lambda functions will allow you to run the code without provisioning or managing servers.

Once you upload the source code file into AWS lambda in the form of ZIP file then AWS lambda will automatically run the code without you provision the servers and also it will automatically scaling your functions up or down based on demand. AWS lambda are mostly used for the event-driven application for the data processing[Amazon S3 buckets](https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-s3/), or responding to HTTP requests.

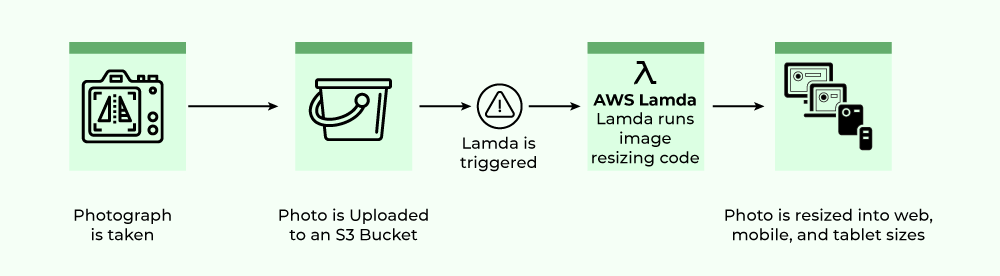
* **Features of AWS Lambda Functions**

The following are the some features which are provided by the AWS (Amazon Web Services):

1. **AutoScaling and High Availability:**AWS lambda will make sure that your application was highly available to the end users when there is sudden incoming traffic. High availability can be achieved by scaling the application.
2. **Serverless Execution:** There is no need for provisioning the servers manually in AWS. AWS lambda will provision the underlying infrastructure based on the triggers you are mentioned whenever a new file uploaded to a particular then AWS lambda will automatically trigger and takes care of the infrastructure.
3. **Pay-per-use-pricing:**AWS will charge you only for the time that time compute engine was active. AWS bills you based on the time taken to execute the code.
4. **Supports different programming languages:**AWS lambda function will support different programming languages. You can build the function with the language at your convenience.

* **Working of AWS Lambda Functions**

Start off by uploading the code to AWS Lambda. From there, set up the code to trigger from other AWS services, HTTP endpoints, or mobile apps. AWS Lambda will only run the code when it's triggered and will also only use the computing resources needed to run it. The user has to pay only for the compute time used



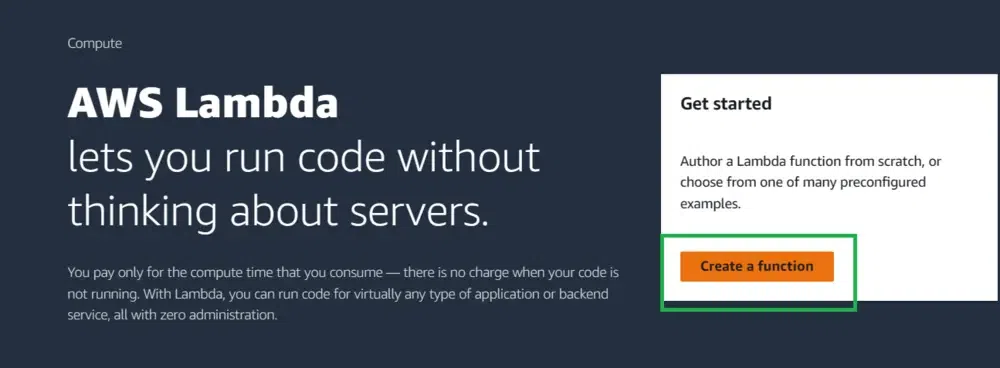
* **Steps for Creating AWS Lambda Functions Using AWS Console**

**Step 1:**Log in to your AWS console and search for Lambda. As shown in the following image.

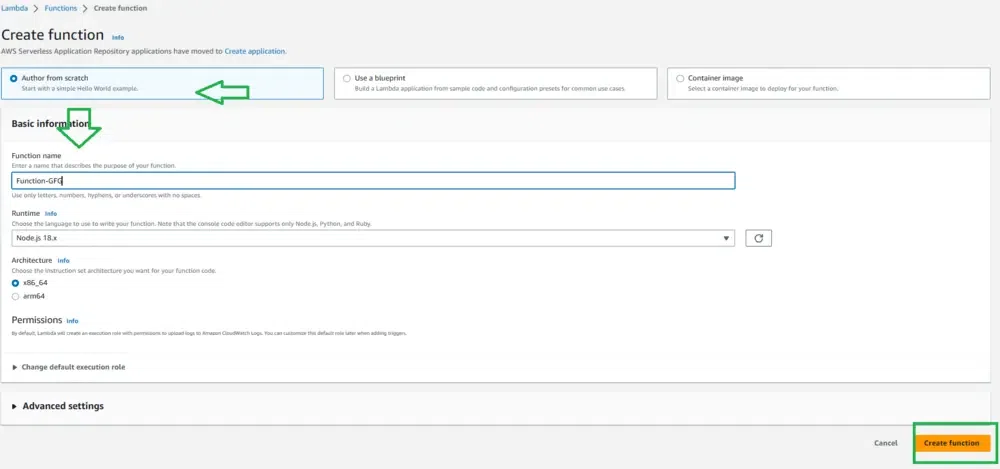
A screenshot of a computer

AI-generated content may be incorrect.

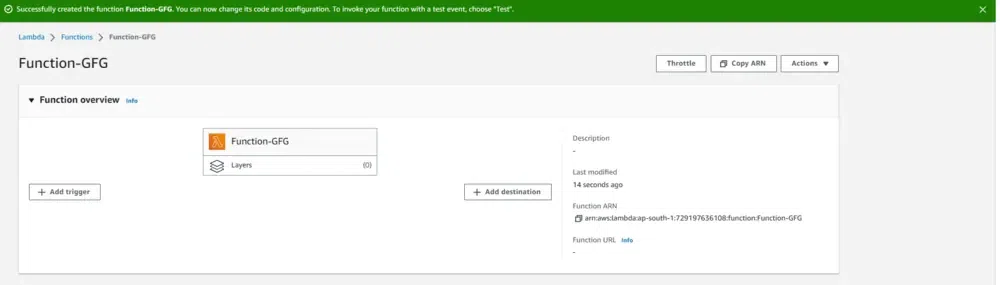
**Step 2:**Click on Create function.



**Step 3:**Here we are going to use a sample hello world program by using Author from scratch and configure the details according to your requirement



**Step 4:**Successfully our function is created.



* **Creating a Lambda Function**

To create a new Lambda function, you can use the following command:

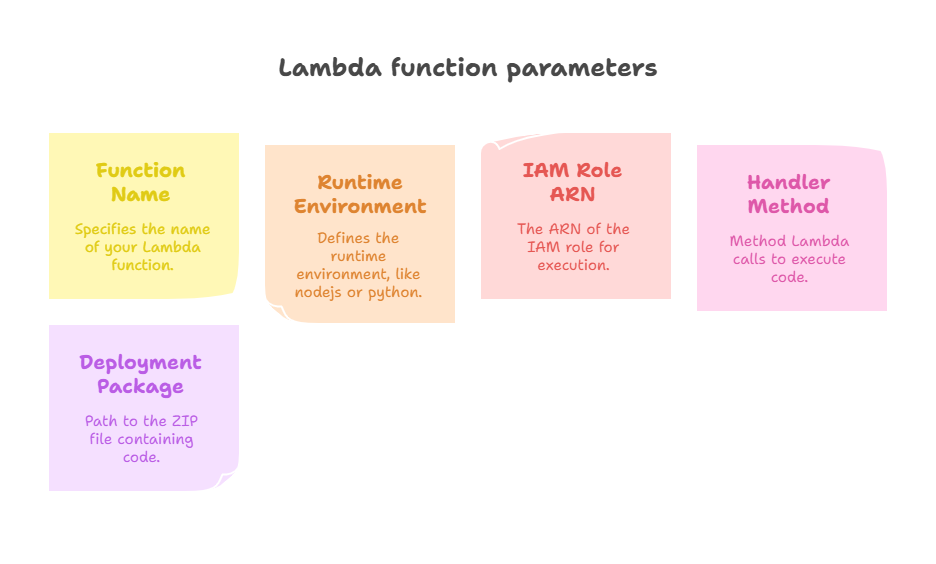
aws lambda create-function --function-name MyFunction \

--runtime nodejs14.x \

--role arn:aws:iam::account-id:role/service-role/MyRole \

--handler index.handler \

--zip-file fileb://function.zip



* **Updating a Lambda Function**

To update an existing Lambda function, use the following command:

**aws lambda update-function-code --function-name MyFunction \**

**--zip-file fileb://function.zip**

* **Deleting a Lambda Function**

**To delete a Lambda function, execute the following command:**

**aws lambda delete-function --function-name MyFunction**

* **Invoking a Lambda Function**

**To invoke a Lambda function, use the command below:**

**aws lambda invoke --function-name MyFunction output.txt**

**Parameters Explained**

* **--function-name: The name of the Lambda function you want to invoke.**
* **output.txt: The file where the output of the function will be stored.**
* **Listing Lambda Functions**

**To list all your Lambda functions, use:**

**aws lambda list-functions**

* **Syntax**

**The basic syntax of a lambda function is as follows:**

**lambda arguments: expression**

* **arguments: A comma-separated list of parameters that the function accepts.**
* **expression: A single expression that is evaluated and returned.**

**Examples**

* **Python**

**In Python, lambda functions are commonly used for operations like sorting, filtering, and mapping.**

**add = lambda x, y: x + y**

**print(add(2, 3)) # Output: 5**

**numbers = [1, 2, 3, 4, 5]**

**even\_numbers = list(filter(lambda x: x % 2 == 0, numbers))**

**print(even\_numbers) # Output: [2, 4]**

* **JavaScript**

**In JavaScript, lambda functions are often referred to as arrow functions. They provide a more concise syntax for writing functions.**

**// A simple arrow function that multiplies two numbers**

**const multiply = (x, y) => x \* y;**

**console.log(multiply(2, 3)); // Output: 6**

**// Using arrow functions with map**

**const numbers = [1, 2, 3, 4, 5];**

**const squared = numbers.map(x => x \* x);**

**console.log(squared); // Output: [1, 4, 9, 16, 25]**

* **Use Cases**

1. **Sorting: Lambda functions can be used as a key function in sorting operations.**
2. **Filtering: They are useful for filtering lists or collections based on specific criteria.**
3. **Mapping: Lambda functions can transform data in collections, such as applying a function to each element.**

* **Conclusion**

**Lambda commands provide a powerful and concise way to define small functions in various programming languages. They enhance code readability and allow for functional programming techniques. Understanding how to use lambda functions effectively can greatly improve your coding efficiency and style.**