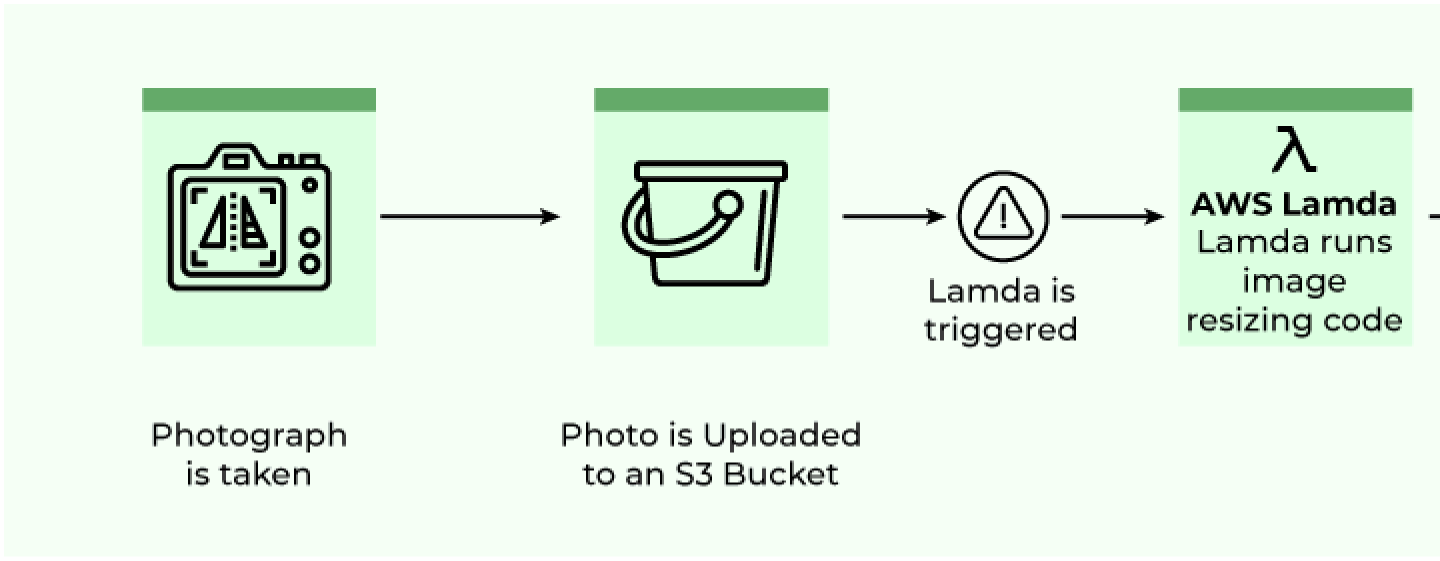
AWS Lambda

AWS Lambda is an Amazon serverless computing system that runs code and automatically manages the underlying computing resources like (EC2). It is an event-driven computing service. It lets a person automatically run code in response to many types of events, such as HTTP requests from the Amazon API gateway, table updates in Amazon DynamoDB, and state transitions. It also enables the person to extend to other AWS services with custom logic and even creates its own back-end services. For example, just write the code and then upload it as a .zip file or any container image. The service works by running code on high-availability computer infrastructure. It then performs all the administrative duties of that compute resource such as:


Photograph is taken
Photo is Uploaded to an S3 Bucket
Lamda is triggered
λ
AWS Lamda
Lamda runs image resizing code

providing maintenance on the server and operating system, auto-scaling and managing the person's capacity provisions, handling security patch deployment lambda extensions code, monitoring, logging, and concurrency function blueprints. AWS lambda is very helpful when you know how to write code but you don't know how to provision the underlying infrastructure in AWS. AWS lambda will scale up the applications rapidly when there is sudden incoming traffic and scale down to zero when the incoming traffic is reduced.

What Are Lambdas Functions?

AWS lambda are server-less compute functions are fully managed by the AWS where developers can run there code without worrying about servers.AWS lambda functions will allow you to run the code with out 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, or responding to HTTP requests.

Examples of AWS Lambdas

The following are the examples of AWS Lambda:

1. Process data from Amazon S3 buckets.
2. Respond to HTTP requests.

3. Build serverless applications.

Use Cases Of AWS (Amazon Web Services) Lambda Functions

You can trigger the lambda in so many ways some of which are mentioned below.

1. **File Processing:** AWS lambda can be triggered by using simple storage services (S3). Whenever files are added to the S3 service Lambda data processing will be triggered.
2. **Web Applications:** You can combine both web applications and AWS lambda which will scale up and down automatically based on the incoming traffic.
3. **IoT (Internet of Things) applications:** You can trigger the AWS lambda based on certain conditions while processing the data from the device which are connected to the IOT applications. It will analyze the data which are received from the IOT application.
4. **Stream Processing:** Lambda functions can be integrated with the Amazonn kinesis to process real-time streaming data for application tracking, log filtering, and so on.

AWS lambda will help you to focus more on your code than the underlying infrastructure. The infrastructure maintenance in AWS was taken care of by AWS lambda.

Features Of AWS (Amazon Web Services) Lambda Functions

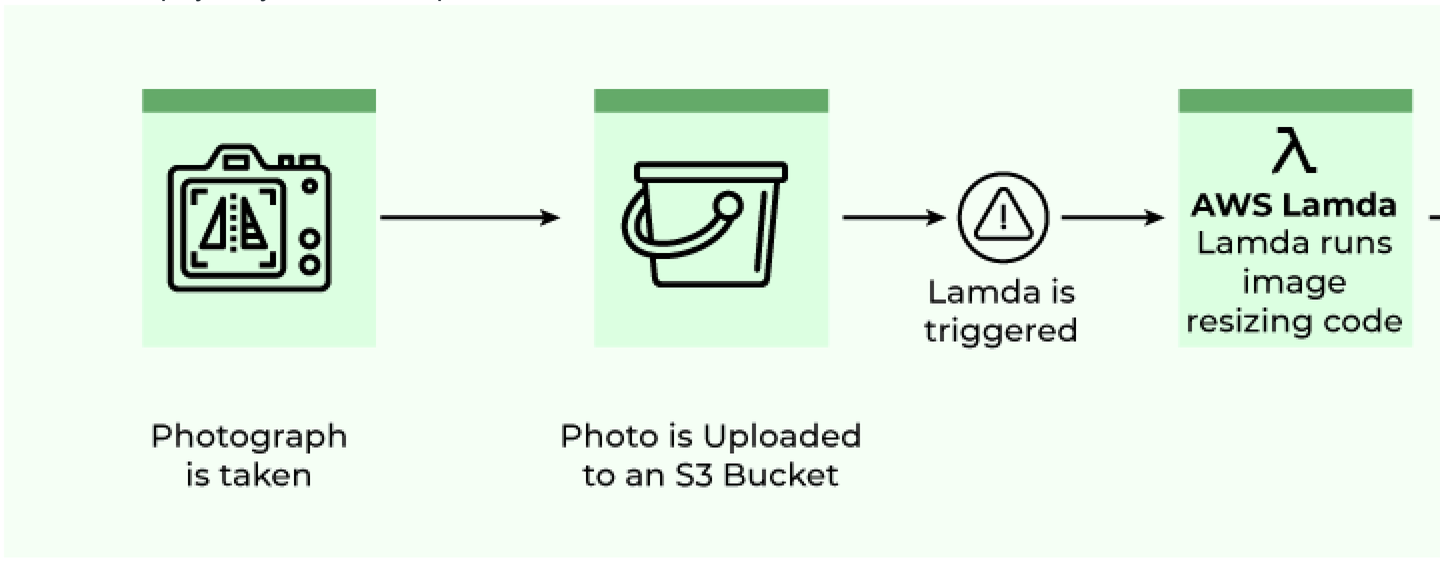
The following are the some features which are provided by the AWS (Amazon Web Services):

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. Following are some languages supported by AWS lambda:
   1. Python.
   2. Node.js.
   3. Java.
   4. C#.
   5. PowerShell.
   6. Go.
5. **Integrates with other AWS Services:** AWS lambda can be integrated with different AWS services like the following :
   1. API Gateway
   2. DynamoDB
   3. S3
   4. Step Functions
   5. SNS
   6. SQS.

6. **Versioning and Deployment:** AWS lambda function will maintain the different kinds of versions of the code by which you can change between the versions without any disruptions y based on the application performances.
7. **Security and Identity Management:** AWS lambda will leverage [AWS Identity and Access Management (IAM)]() to control the access to the functions which are built by using lambda You can define fine-grained permissions and policies to secure your functions and ensure that only authorized entities can invoke them.

How Does AWS (Amazon Web Services) Lambda Functions Work?

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.


Photograph is taken
Photo is Uploaded to an S3 Bucket
Lamda is triggered
λ
AWS Lamda
Lamda runs image resizing code

Pricing of AWS (Amazon Web Services) Lambda Function

AWS lambda pricing is based on the no.of requests made to your function and the time it has taken to execute the function. The following are the pricing factors to be considered for AWS Lambda. As previously mentioned, with AWS Lambda user only pays for what he uses, factoring in the number of requests and duration of the execution of the code. No charges are taken for creating lambda functions. Lambda considers a request to be each time it starts executing in response to a trigger such as an event notification or an invocation volume. The duration of the code is calculated from the moment the code begins executing until it returns or is terminated. If one is unsure about whether AWS Lambda is the right choice or not, there is a free tier option available to try. This option includes 1M free requests per month and 400,000 GB-seconds of compute time per month.

1. **Requests:** AWS lambda charges are made up of the no.of requests made to your AWS lambda function. Each request will be counted individually based on the time duration also aws lambda will calculate all the requests made per month. The pricing will be varied from the different regions and the total no.of requests made for your function.

2. **Duration:** AWS lambda will be charged based on the duration of time your code started executing and till the time it is terminated. And also while executing the AWS lambda you should allocate some memory to your function this factor is also counted while billing.
3. **Free requests:** You have one million free requests per month which is free for all users and when coming to the duration you have 400,000 GB-seconds of compute time per month.

Pricing Factor	Details
Requests	It will charge per request. - Each request is counted individually. - Pricing varies by region.
Duration	The charges are based on execution time from start to termination. - Memory allocation affects cost.
Free Tier	The 1 million free requests per month. - 400,000 GB-seconds of compute time per month.

Advantages of AWS (Amazon Web Services) Lambda Function

The following are the advantages of AWS Lambda function

1. **Zero Server Management:** Since AWS Lambda automatically runs the user's code, there's no need for the user to manage the server. Simply write the code and upload it to Lambda.
2. **Scalability:** AWS Lambda runs code in response to each trigger, so the user's application is automatically scaled. The code also runs in parallel processes, each triggered individually, so scaling is done precisely with the size of the workload.
3. **Event-Driven Architecture:** AWS Lambda function can be triggered based on the events happing in the aws other service like when the file or video is added to the S3 bucket you can trigger the AWS Lambda function.
4. **Automatic High Availability:** When there is high demand or high incoming traffic aws lambda function will automatically scale the servers.
5. **Affordable:** With AWS Lambda, one doesn't pay anything when the code isn't running. The user has to only be charged for every 100ms of code execution and the number of times his code is actually triggered.

Disadvantages of AWS (Amazon Web Services) Lambda Function

The following are the disadvantages of AWS Lambda function:

1. **Latency while starting:** While aws lambda is going to be activated after a long gap it will take some time to initialize the service which is required to deploy the application at that time end users will face latency issues.
2. **Limited control of infrastructure:** Behalf of your lambda function is going to take of underlying infrastructure so you will have very limited control over undelaying infrastructure.

3. **Time Limit:** AWS Lambda enforces a maximum execution time limit for functions, which is currently set to 900 seconds (15 minutes). If your function exceeds this time limit, it will be forcibly terminated.
4. **Vendor Lock-In:** If you want to execute the lambda function then you need the support of any cloud provider as here we are using AWS because it is widely used in the market.
5. **Limited Computing and Memory Options:** Limited configuration is there on the memory and CPU configuration. The predefined memory configuration was 120 MB to 120 GB and memory configuration determines the corresponding CPU power.

Benefits of AWS Lambda Functions

The following are the benefits of AWS Lambda Functions:

- **Cost Efficiency**: It only charges for the compute that is only for running known as pay-as-you-go model.
- **Automatic Scaling**: AWS Lambda automatically helps in scaling your applications by running code in response to each trigger.
- **Reduced Operational Compliance**: It allows the developers to focus on building your logic, the aws itself while take care of the infrastructure.
- **Integration with AWS Services**: It provides a seamlessly integration with other AWS services, enabling strong and scalable applications