**Why AWS Lambda?**

* It is an automated version of EC2. No worries about the underlying Architecture.
* For executing background tasks.

**AWS SDK?**

* One can simplify using aws services in ur applications with an API tailored to ur programming language or platform.
* AWS SDKs are called s/w development kits, which are APIs which are defined for a specific programming language.
* These APIs can be integrated with an IDE and thus can be used with ease.

**AWS SDKs with IDE ?**

Eclise and Visual Studio.

**Using AWS lambda with Eclipse ?**

**AWS Lambda Tutorial**

Today we’re going to talk about AWS Lambda. AWS Lambda is a compute service offered by Amazon. You must be curious as there are several other compute services from AWS, such as AWS EC2, AWS Elastic Beanstalk, AWS Opsworks etc., then why another compute service? In this AWS Lambda tutorial you will discover what is AWS Lambda, why it is used and in which use cases you should consider it.

## What is AWS Lambda?

Amazon explains, AWS Lambda (λ) as a ‘serverless’ compute service, meaning the developers, don’t have to worry about which AWS resources to launch, or how will they manage them, they just put the code on lambda and it runs, it’s that simple. Although , Lambda can only be used to execute background tasks.

## Where will I use AWS Lambda?

AWS Lambda executes your backend code, by automatically managing the AWS resources. When we say ‘manage’, it includes launching or terminating instances, health checkups, auto scaling, updating or patching new updates etc.

## So, how does it work?

The code that you want Lambda to run is known as a **Lambda function**. Now, as we know a function runs only when it is called, right? Here, **Event Source** is the entity which triggers a Lambda Function, and then the task is executed.

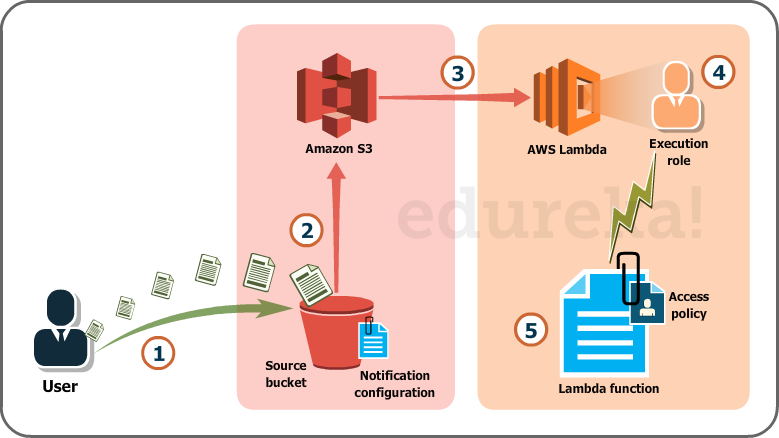
**Let’s take an example to understand it more clearly.**

Suppose you have an app for image uploading. Now when you upload an image, there are a lot of tasks involved before storing it, such as resizing, applying filters, compression etc.

So, this task of uploading of an image can be defined as an **Event Source** or the ‘trigger’ that will call the Lambda Function, and then all these tasks can be executed via the Lambda function.

In this example, a developer just has to define the event source and upload the code.

Let’s understand this example with real AWS resources now,



Over here we will be uploading images in the form of objects to an S3 bucket. This uploading an image to the S3 bucket, will become an event source or the ‘trigger’.

The whole process, as you can see in the diagram, is divided into 5 steps, let’s understand each one of them.

1. User uploads an image (object) to a source bucket in S3 which has notification attached to it, for Lambda.
2. The notification is read by S3 and it decides where to send that notification.
3. S3 sends the notification to Lambda, this notification acts as an invoke call of the lambda function.
4. Execution role in Lambda can be defined by using IAM (Identity and Access Management) to give access permission for the AWS resources, for this example here it would be S3.
5. Finally, it invokes the desired lambda function which works on the object which has been uploaded to the S3 bucket.

*If you were to solve this scenario traditionally, along with development, you would have hired people for managing the following tasks:*

* Size, provision and scale up group of servers
* Managing OS updates
* Apply security patches and
* Monitor all this infrastructure for performance and availability.

This would have been an expensive, tedious and tiresome task, therefore the need for AWS Lambda is justified. AWS Lambda is compatible with Node.JS, Python and Java, so you can upload your file in a zip, define an event source and you are set!

You can read more about [S3 AWS](https://www.edureka.co/blog/s3-aws-amazon-simple-storage-service/) here for a deeper understanding.

*We now know – How Lambda works and What Lambda does*.

Now, let’s understand-

.