Lambda

**What is Lambda in AWS?**

* AWS Lambda is a powerful **serverless computing service** that automatically runs code in response to events, without requiring you to manage the underlying infrastructure.
* It supports event-driven applications triggered by events such as HTTP requests, DynamoDB table updates, or state transitions.
* It automatically scales applications based on traffic, handling server management, auto-scaling, security patching, and monitoring.
* AWS Lambda is ideal for developers who want to focus on writing code without worrying about infrastructure management.

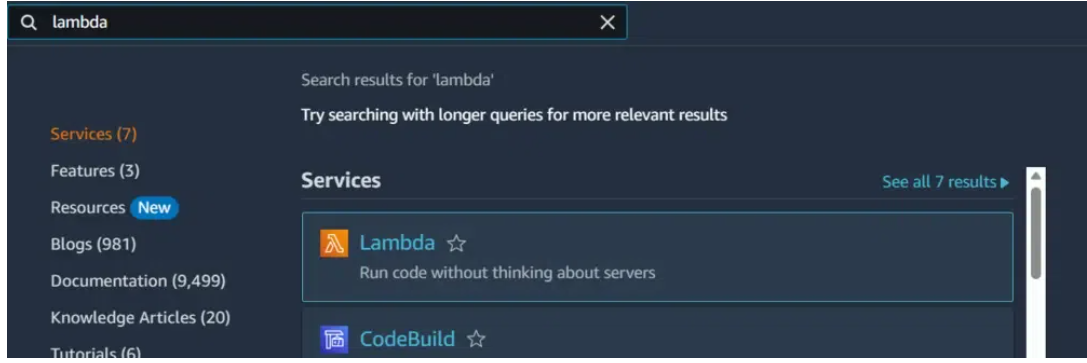
Functions of Lambda: -

1. AWS lambda is server-less compute functions are fully managed by the AWS where developers can run their code without worrying about servers.
2. AWS lambda functions will allow you to run the code without provisioning or managing servers.
3. 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 be scaling your functions up or down based on demand.
4. 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.

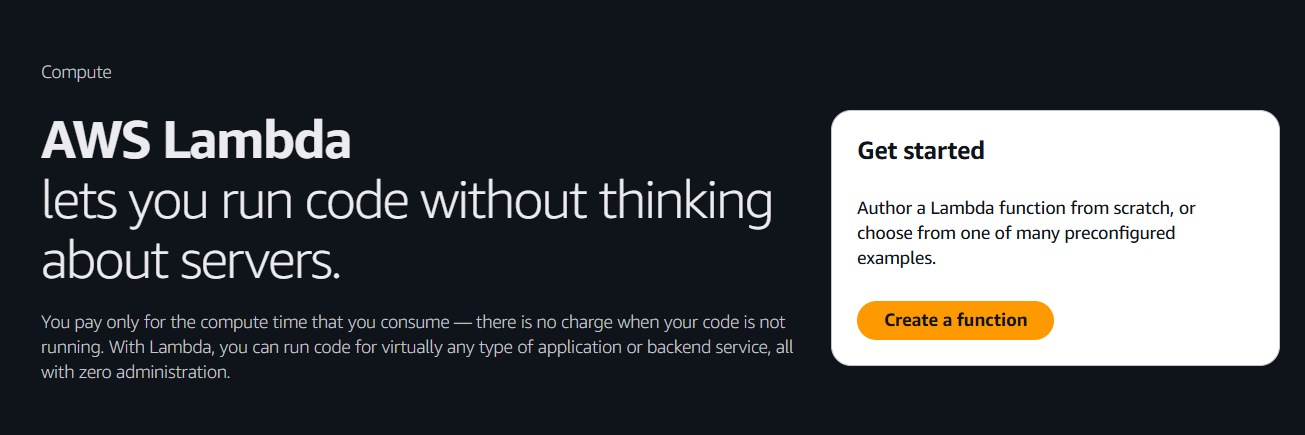
Steps for Creating AWS Lambda Functions Using AWS Console: -

**Task 1: Creating Lambda function**

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



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

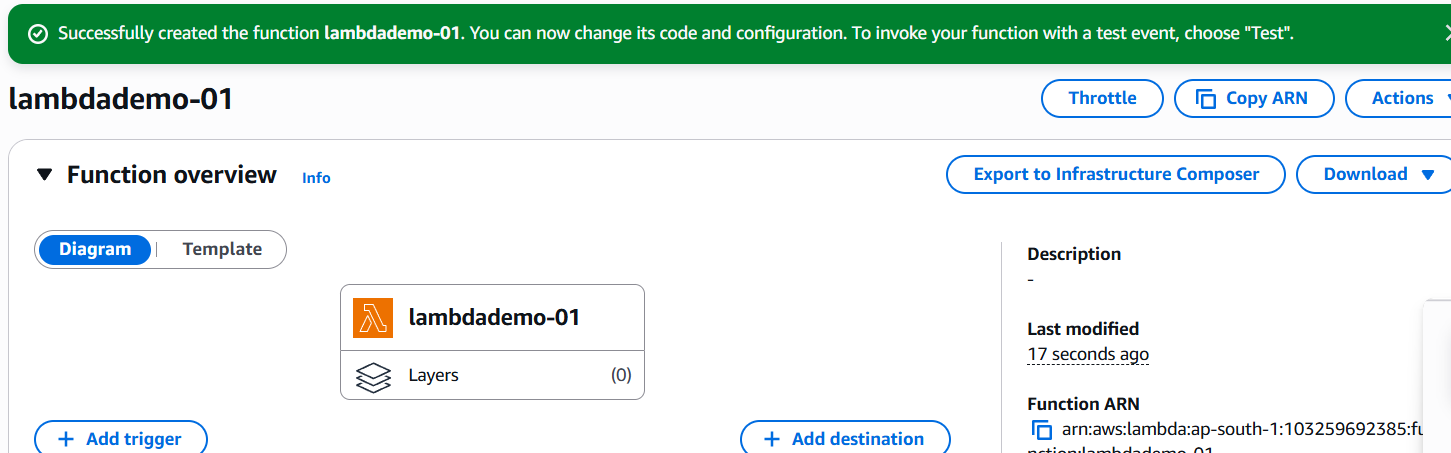
****

**Step 3:** Choose **Author from scratch.**

Enter your function name- **lambdademo-01.**

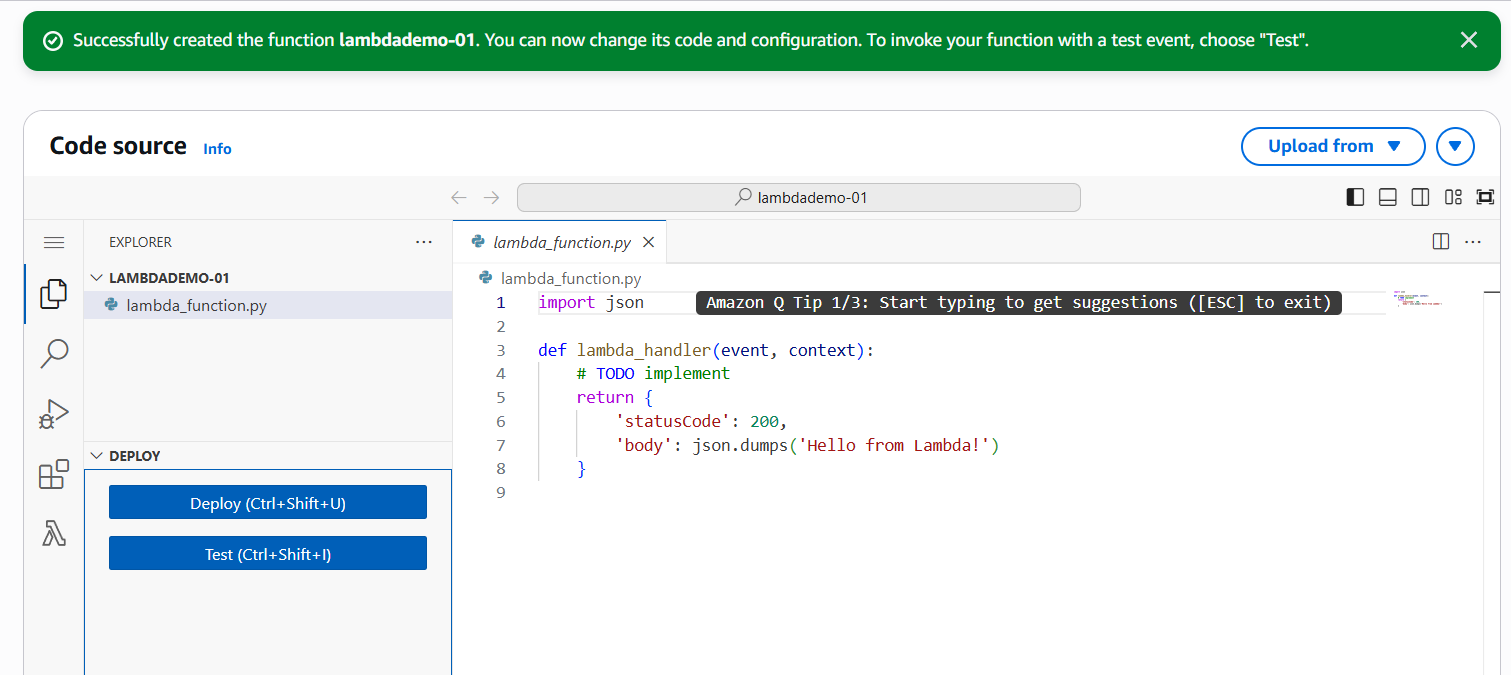
In runtime select **– Python3.13.**

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



**Task 2: Creating a test event**

**Step 1:** Scroll down then you see the code written there.



**Step 2:** Next you can run this code by simply clicking on **test**.

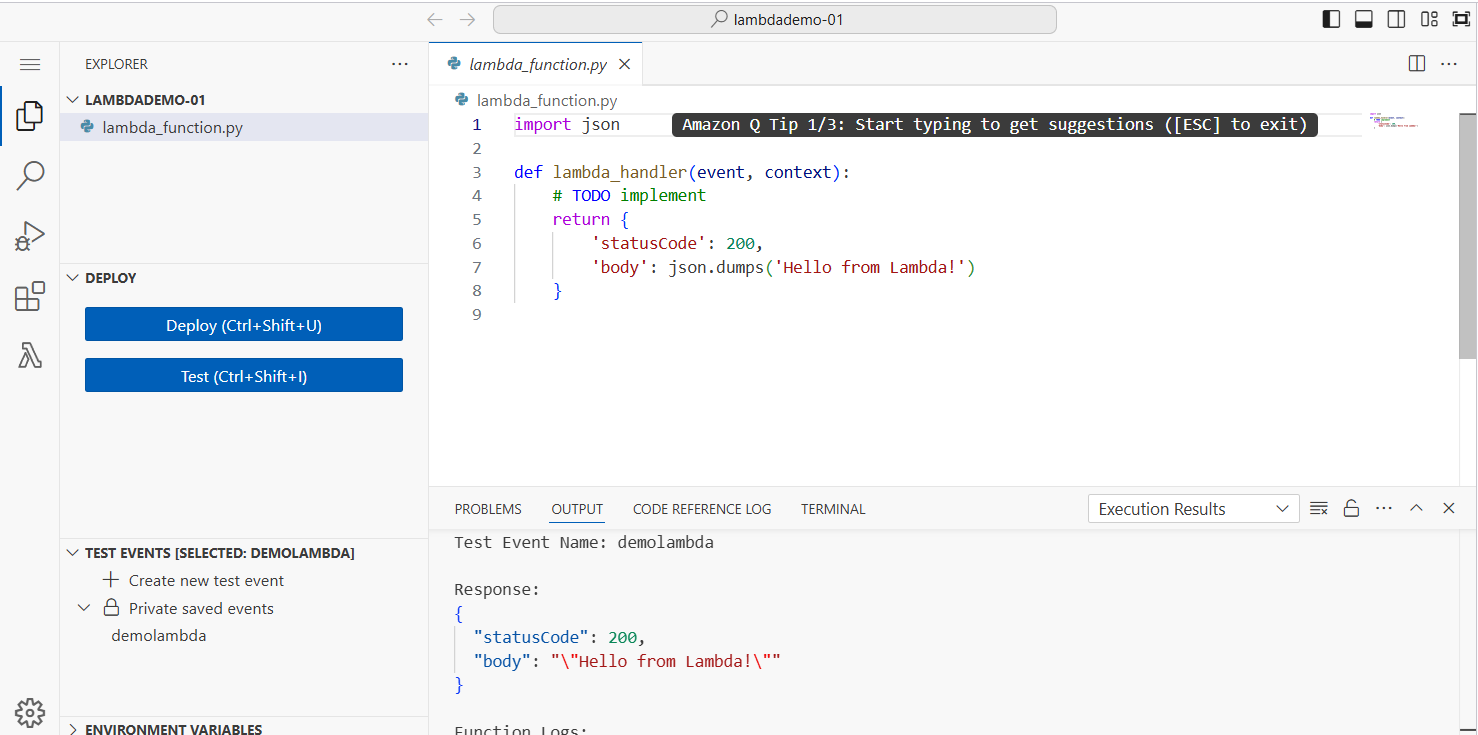
**Step 3:** When you click on test then this will show “**create new test**”, which means you don’t have any test event right now.

**Step 4:** Creating new test event. Click on test.

**Step 5:** Enter event name: **demolambda**

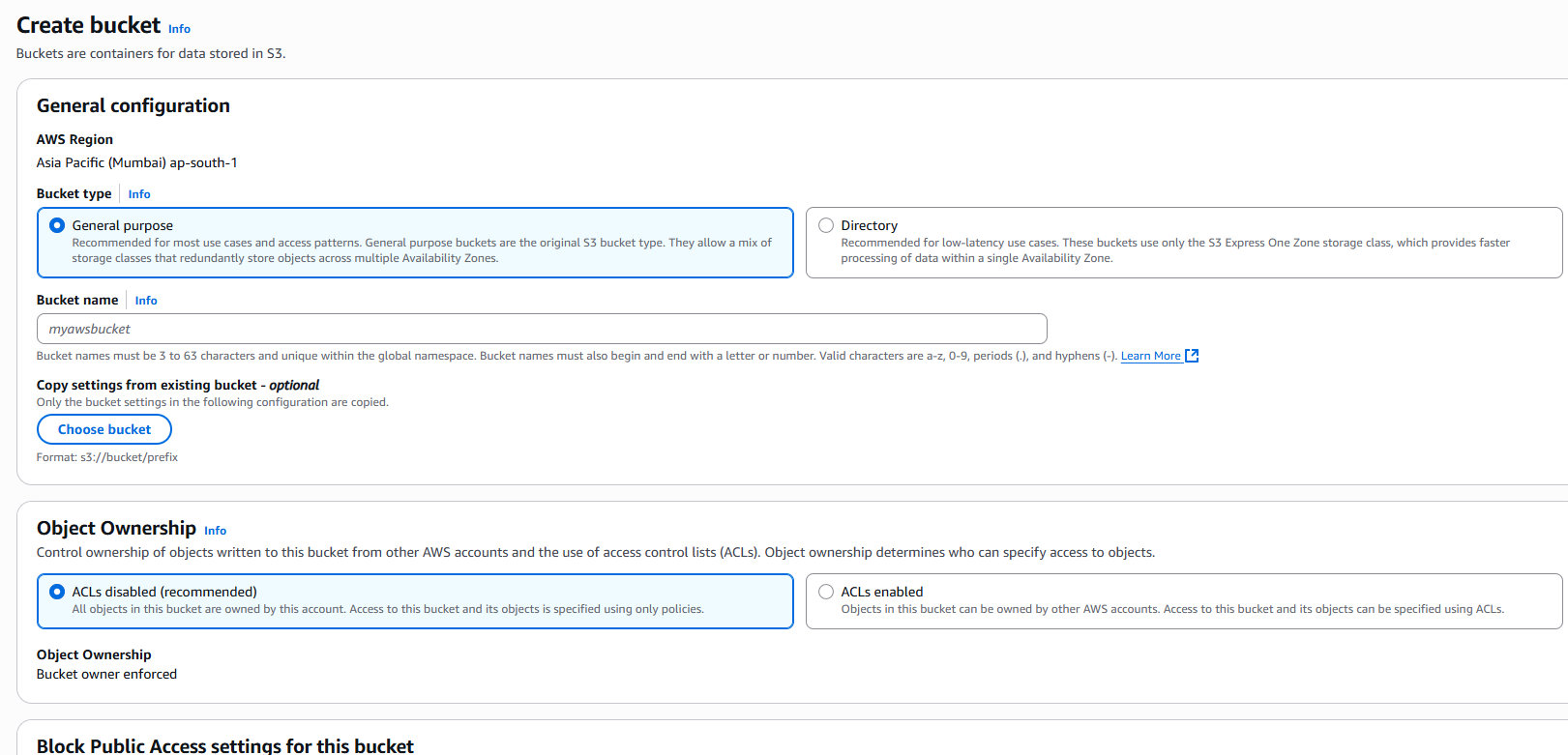
Step 6: Now click on **save.**

**Step 7**: Now click on test. Then your output is ready.



**Task 3: Create bucket**

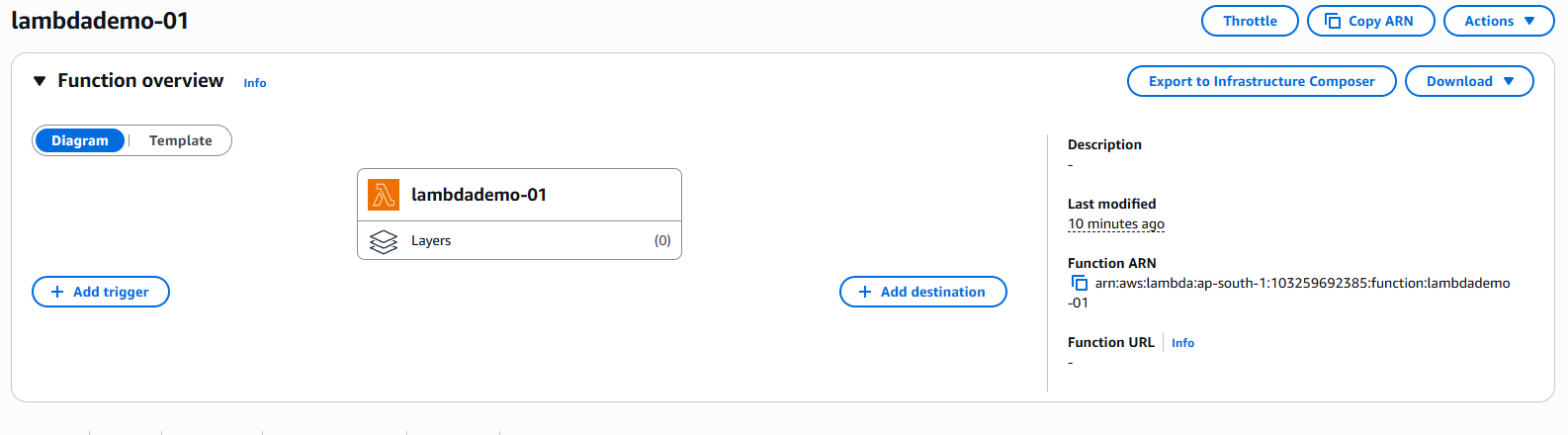
1. Navigate to aws S3 by clicking on services in the top left corner.
2. Click on create bucket.
3. Enter the bucket name: Mys3bucket.
4. Region: Select Asia Pacific(Mumbai) ap-south-1.



1. Leave all other settings as default and click on Create bucket.

**Task 4: Add S3 bucket as triggered.**

1. Go to lambda function and then click on add trigger written on left side.



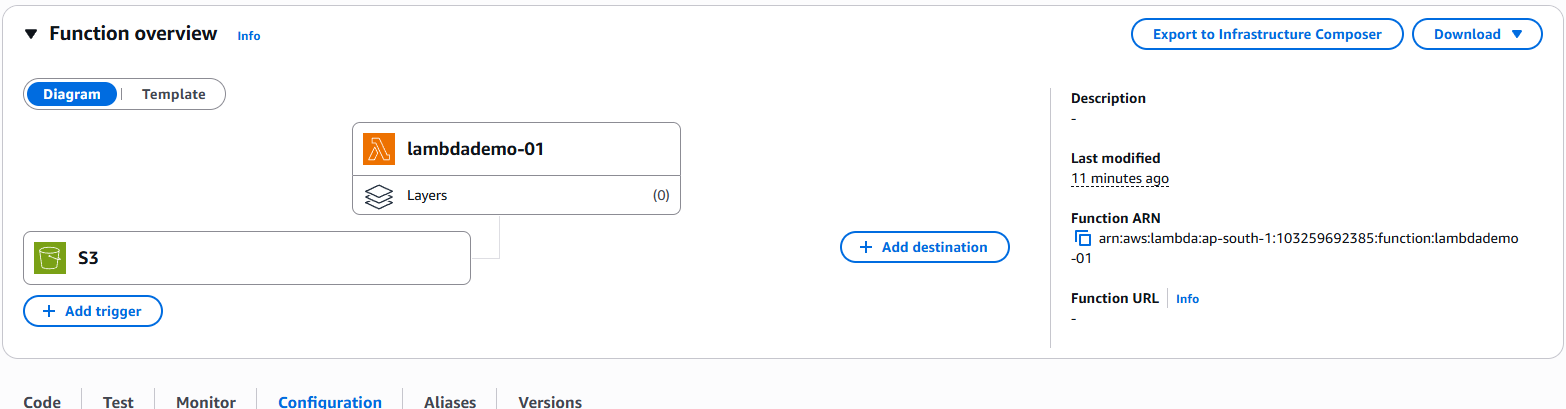
1. After clicking on add trigger, go to trigger configuration.

Select s3

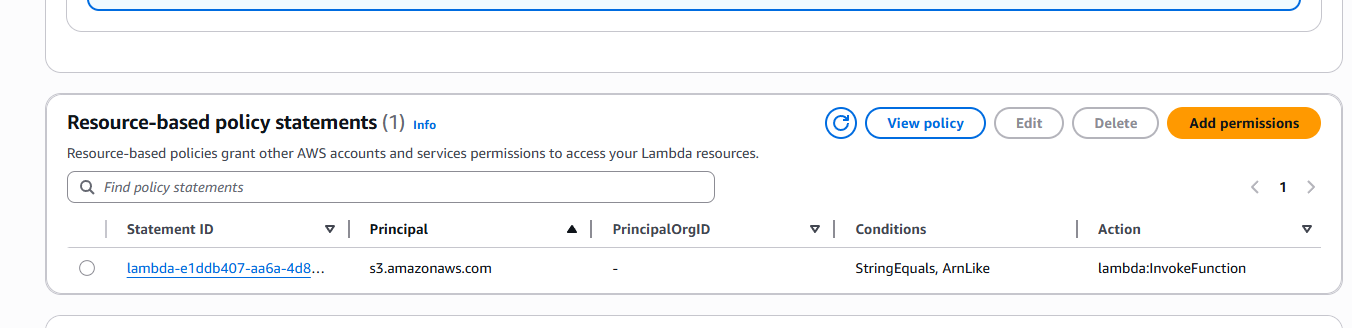
Next select the bucket which you have created in s3.

In recursive innovation – Check the box.

1. Then click on add.



1. Now scroll down and then click on permissions.



1. You can also check by clicking on the bucket name and then by clicking on properties.
2. Now, you can say that your lambda function is working properly.

**Use Cases of AWS 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)](https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-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 analyse the data which are received from the IOT application.
4. **Stream Processing:** Lambda functions can be integrated with the Amazon kinesis to process real-time streaming data for application tracking, log filtering, and so on.

**Features of AWS Lambda Functions: -**

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.

[Python](https://www.geeksforgeeks.org/introduction-to-python/)

[Node.js](https://www.geeksforgeeks.org/nodejs/)

[Java](https://www.geeksforgeeks.org/java/)

[C#](https://www.geeksforgeeks.org/csharp-programming-language/)

PowerShell

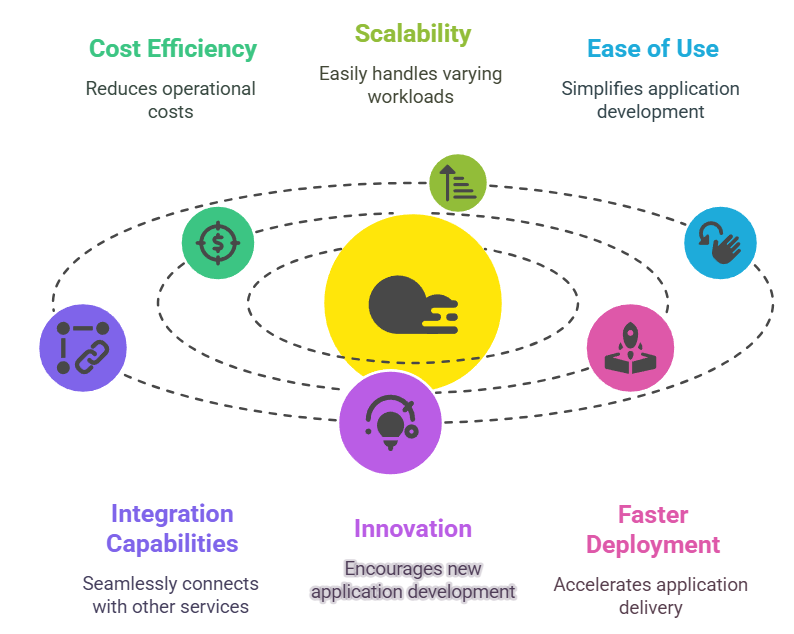
Go

1. **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.
2. **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.**

**Advantages of AWS 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 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](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/) configuration. The predefined memory configuration was 120 MB to 120 GB and memory configuration determines the corresponding CPU power.

**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,](https://www.geeksforgeeks.org/top-aws-services/) enabling strong and scalable applications.