
aws

AWS Lambda

Grow Data Skills

AWS Lambda is a serverless computing service that lets you run code without provisioning or managing servers. It automatically scales your application by running code in response to events, such as changes to data in Amazon S3 buckets or updates in an Amazon DynamoDB table.

Here are the primary features and properties of AWS Lambda:

- **Event-Driven**: AWS Lambda is designed to use events like changes to data in an S3 bucket or an update to a DynamoDB table to trigger the execution of code.
- **Scaling**: Lambda functions can scale automatically by running code in response to each trigger. Your trigger can be an uploaded image, a new log file, a new row in a database, etc.
- **Languages Supported**: As of my last update, AWS Lambda supports multiple programming languages. These include Node.js, Python, Ruby, Java, Go, .NET Core, and custom runtimes that you can provide.
- **Stateless**: By default, AWS Lambda is stateless, meaning each function execution is independent. If you need to maintain state, you would use an external service, like Amazon RDS or DynamoDB.
- **Short-lived**: Lambda functions are designed to be short-lived. Initially, there was a 5-minute max execution time, which later was extended to 15 minutes.
- **Resource Specification**: You can specify the amount of memory allocated to your Lambda function. AWS Lambda allocates CPU power linearly in proportion to the amount of memory configured.

AWS Lambda

- **Built-in Fault Tolerance**: AWS Lambda maintains compute capacity and infrastructure reliability, including monitoring, logging via Amazon CloudWatch, and automatic retries.
- **Deployment**: Code can be deployed as a Lambda function via a ZIP or JAR file. AWS also provides a blueprints feature to start off with sample code for common use cases.
- **Integrated with AWS Services**: It's integrated with many AWS services making it a flexible tool. For instance, you can trigger a Lambda function from changes in S3, updates in DynamoDB, endpoint requests in API Gateway, etc.
- **Layers**: Lambda Layers are a distribution mechanism for libraries, custom runtimes, and other function dependencies. Layers promote code sharing and separation of responsibilities.
- **Billing**: With AWS Lambda, you're billed for the compute time your code is running. You aren’t charged when your code isn’t running.
- **Event Source Mapping:** If a Lambda function is triggered by an event source, AWS Lambda takes care of the reading, retries, and deletion of the event, ensuring that each event is processed in order.
- **Concurrent Executions**: AWS Lambda scales functions in parallel. While it manages and scales these automatically, there is a default safety throttle for the number of concurrent executions across all functions in a given region.

AWS Lambda

Grow Data Skills