AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It allows developers to run code without provisioning or managing servers. With Lambda, you can build applications using a "serverless" architecture, where you only pay for the compute time used when the code is executed.

Key features of AWS Lambda:

1. Event-Driven: Lambda functions are triggered in response to specific events, such as changes in data stored in Amazon S3, updates to a database in Amazon DynamoDB, or HTTP requests via Amazon API Gateway.
2. Multiple Language Support: AWS Lambda supports multiple programming languages, including Node.js, Python, Java, Go, .NET Core, and more, allowing developers to write functions in their preferred language.
3. Automatic Scaling: Lambda automatically scales the execution environment to handle a large number of concurrent requests, ensuring high availability and performance.
4. Stateless: Lambda functions are stateless, meaning they do not retain any information between invocations. Any required state or data must be stored externally, such as in a database or S3.
5. Integration with AWS Services: Lambda integrates seamlessly with other AWS services, enabling you to create serverless architectures by connecting Lambda with services like S3, DynamoDB, SNS, and more.
6. Pay-As-You-Go: You are charged only for the compute time used to execute your Lambda function, measured in milliseconds. There is no charge when your code is not running.
7. Flexible Execution Models: Lambda functions can be executed synchronously or asynchronously, allowing for various use cases and integration patterns.
8. Versioning and Aliases: Lambda supports versioning and aliases, allowing you to manage and deploy different versions of your functions and control traffic routing.

Common use cases for AWS Lambda include:

* Backend for Mobile and Web Applications: Lambda can serve as a backend for mobile and web applications, processing API requests and interacting with databases and other services.
* Data Processing: Lambda can process data from various sources, such as logs, streams, or files, and perform real-time data transformations or analytics.
* Automation and Orchestration: Lambda can automate tasks and workflows by responding to events and triggering actions in other AWS services.
* Microservices: Lambda functions can be used to build individual microservices that collectively form a larger application architecture.
* Chatbots and Voice Assistants: Lambda can power chatbots or voice assistants by processing user requests and generating responses.

AWS Lambda's serverless architecture offers developers the flexibility to build scalable and event-driven applications without managing infrastructure. It helps reduce operational overhead and enables faster development and deployment of applications in the AWS cloud.