

Einführung in AWS Lambda

AWS Lambda ist ein Serverless-Computing-Service von Amazon Web Services (AWS), der Entwicklern die Möglichkeit gibt, Code auszuführen, ohne sich Gedanken über die zugrunde liegende Infrastruktur machen zu müssen. Stattdessen können Entwickler einfach Code schreiben und ihn als Funktion auf AWS Lambda hochladen. AWS Lambda kümmert sich dann um die zugrunde liegende Infrastruktur und führt den Code aus, wenn er benötigt wird.

Wie funktioniert AWS Lambda?

AWS Lambda verwendet eine ereignisgesteuerte Architektur (Event-Driven Architecture), bei der Funktionen nur dann ausgeführt werden, wenn sie durch ein Ereignis ausgelöst werden. Ein Ereignis kann zum Beispiel das Hochladen einer Datei in einen S3-Bucket, eine Nachricht in einer SQS-Warteschlange oder ein HTTP-Aufruf sein. Wenn ein Ereignis ausgelöst wird, startet AWS Lambda automatisch die zugehörige Funktion und führt den Code aus.

AWS Lambda ist auch sehr skalierbar, da AWS Lambda automatisch die Anzahl der ausgeführten Funktionen basierend auf der aktuellen Last und dem Bedarf anpassen kann. Dies bedeutet, dass AWS Lambda in der Lage ist, mit Lastspitzen umzugehen, ohne dass Entwickler sich Gedanken über die zugrunde liegende Infrastruktur machen müssen.

Vorteile von AWS Lambda

- **Skalierbarkeit:** AWS Lambda kann automatisch die Anzahl der ausgeführten Funktionen basierend auf der aktuellen Last und dem Bedarf anpassen.
- **Kostenersparnis:** Entwickler zahlen nur für die tatsächlich ausgeführte Zeit ihrer Funktionen, anstatt für die gesamte Infrastruktur.
- **Einfache Verwaltung:** AWS Lambda kümmert sich um die zugrunde liegende Infrastruktur und die Skalierung, so dass Entwickler sich auf die Entwicklung ihrer Funktionen konzentrieren können.
- **Schnelle Entwicklung:** Entwickler können Funktionen schnell und einfach schreiben und sie sofort testen, ohne sich Gedanken über die Infrastruktur machen zu müssen.

Laufzeitumgebungen

AWS Lambda unterstützt eine Vielzahl von Programmiersprachen, darunter Java, Go, PowerShell, Node.js, C#, Python, Ruby und mehr. Diese Sprachen werden als Laufzeitumgebungen bezeichnet und jede Laufzeitumgebung hat ihre eigenen Einschränkungen und Vorteile. Entwickler können die Laufzeitumgebung auswählen, die am besten zu ihren Anforderungen passt.

Trigger

Ein Trigger ist ein Ereignis, das eine AWS Lambda-Funktion auslöst. AWS Lambda unterstützt verschiedene Arten von Triggern, darunter:

- **API-Gateway:** Eine HTTP-API, die von anderen Anwendungen aufgerufen werden kann.
- **CloudWatch-Regeln:** Eine Möglichkeit, AWS Lambda-Funktionen auf Basis von CloudWatch-Ereignissen auszuführen.
- **DynamoDB:** Eine NoSQL-Datenbank, die eine vollständig verwaltete Schlüssel-Wert- und Dokumentdatenbank bereitstellt.

- **S3:** Ein Objektspeicher, der als Datenlake, als Backup- und Archivierungslösung sowie als Ziel für Datenverarbeitungsservices wie AWS Glue verwendet werden kann.

Konfigurationen

AWS Lambda-Funktionen können auf verschiedene Arten konfiguriert werden, um ihre Funktionalität anzupassen. Einige wichtige Konfigurationsmöglichkeiten sind:

- **Speicher:** AWS Lambda-Funktionen können bestimmte Mengen an Arbeitsspeicher zugewiesen werden, um ihre Leistung zu verbessern.
- **Timeout:** AWS Lambda-Funktionen haben standardmäßig eine maximale Ausführungszeit von 15 Minuten. Diese Zeit kann jedoch je nach Bedarf erhöht oder verringert werden.
- **IAM-Rollen:** AWS Lambda-Funktionen können Zugriff auf andere AWS-Services und Ressourcen benötigen. Hierfür können IAM-Rollen erstellt und mit der Funktion verknüpft werden.

Fehlerbehandlung

AWS Lambda bietet verschiedene Mechanismen zur Fehlerbehandlung, darunter:

- **Fehlersuche und Protokollierung:** Entwickler können AWS CloudWatch Logs verwenden, um Protokolldaten zu sammeln und Fehler zu diagnostizieren.
- **Retry-Logik:** AWS Lambda bietet eine integrierte Retry-Logik, die automatisch wiederholte Ausführungen von Funktionen ermöglicht, die aufgrund von Fehlern fehlgeschlagen sind.

Preise

AWS Lambda wird auf Pay-per-Use-Basis berechnet, was bedeutet, dass nur für die tatsächlich genutzte Rechenzeit bezahlt werden muss. Der Preis richtet sich nach der Laufzeit und der Anzahl der Ausführungen. **Die ersten 1 Million Anfragen und 400.000 GB-Sekunden an Rechenzeit pro Monat sind kostenlos.** Darüber hinaus sind die Preise je nach Region und Speicherkapazität unterschiedlich.

Einschränkungen

AWS Lambda hat einige Einschränkungen, die bei der Entwicklung und Ausführung von Funktionen berücksichtigt werden müssen:

- **Maximale Ausführungszeit:** Standardmäßig hat eine AWS Lambda-Funktion eine maximale Ausführungszeit von 15 Minuten. Dies kann jedoch mit einer speziellen Einstellung erhöht werden.
- **Speicherkapazität:** AWS Lambda-Funktionen haben Zugriff auf einen temporären Speicherplatz, der während der Ausführung bereitgestellt wird. Die maximale Speicherkapazität beträgt 3 GB.
- **Größe der Bereitstellungspakete:** AWS Lambda-Funktionen müssen innerhalb eines bestimmten Limits für die Größe der Bereitstellungspakete bleiben. Die Limits variieren je nach Sprache und werden in der AWS-Dokumentation beschrieben.
- **Einschränkungen für gleichzeitige Ausführungen:** Es gibt eine Obergrenze für die Anzahl der gleichzeitig ausgeführten Funktionen pro Konto. Diese Einschränkungen können jedoch erhöht werden, indem der AWS-Support kontaktiert wird.
- **Zugriff auf Ressourcen:** AWS Lambda-Funktionen haben Zugriff auf eine begrenzte Anzahl von Ressourcen, darunter Speicher, CPU-Zeit und Netzwerkbandbreite. Bei Bedarf können diese Ressourcen jedoch erweitert werden.

Fazit

AWS Lambda ist ein leistungsstarker und skalierbarer Service, der Entwicklern die Möglichkeit gibt, schnell und einfach Code auszuführen, ohne sich Gedanken über die zugrunde liegende Infrastruktur machen zu müssen. Mit AWS Lambda können Entwickler schnell auf Ereignisse reagieren und schnell auf Änderungen in ihrer Umgebung reagieren. AWS Lambda ist eine großartige Option für Entwickler, die serverlos arbeiten möchten und sich auf die Entwicklung von Funktionen konzentrieren möchten, anstatt sich um die zugrunde liegende Infrastruktur kümmern zu müssen.