AWS Architecture Blog

How to Design Your Serverless Apps for Massive Scale

by George Mao | on 25 JUN 2019 | in Advanced (300), Amazon API Gateway, Amazon Kinesis, Amazon RDS, Architecture, AWS Lambda, Serverless | Permalink | Share

Serverless is one of the hottest design patterns in the cloud today, allowing you to focus on building and innovating, rather than worrying about the heavy lifting of server and OS operations. In this series of posts, we'll discuss topics that you should consider when designing your serverless architectures. First, we'll look at architectural patterns designed to achieve massive scale with serverless.

Scaling Considerations

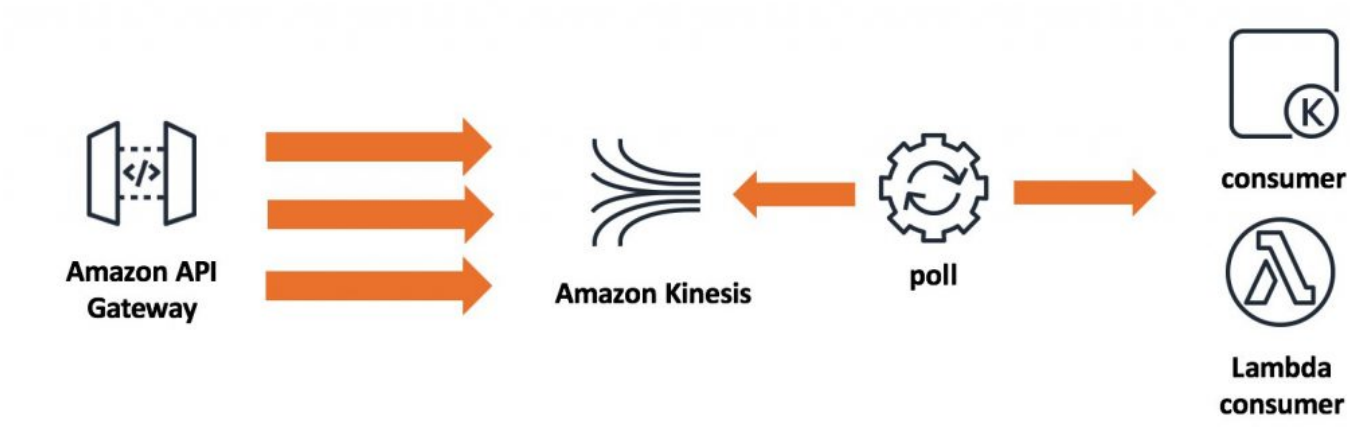
In general, developers in a "serverful" world need to be worried about how many total requests can be served throughout the day, week, or month, and how quickly their system can scale. As you move into the serverless world, the most important question you should understand becomes: "What is the concurrency that your system is designed to handle?"

The AWS Serverless platform allows you to scale very quickly in response to demand. Below is an example of a serverless design that is fully synchronous throughout the application. During periods of extremely high demand, Amazon API Gateway and AWS Lambda will scale in response to your incoming load. This design places extremely high load on your backend relational database because Lambda can easily scale from thousands to tens of thousands of concurrent requests. In most cases, your relational databases are not designed to accept the same number of concurrent connections.

This design risks bottlenecks at your relational database and may cause service outages. This design also risks data loss due to throttling or database connection exhaustion.

Cloud Native Design

Instead, you should consider decoupling your architecture and moving to an asynchronous model. In this architecture, you use an intermediary service to buffer incoming requests, such as Amazon Kinesis or Amazon Simple Queue Service (SQS). You can configure Kinesis or SQS as out of the box event sources for Lambda. In design below, AWS will automatically poll your Kinesis stream or SQS resource for new records and deliver them to your Lambda functions. You can control the batch size per delivery and further place throttles on a per Lambda function basis.



This design allows you to accept extremely high volume of requests, store the requests in a durable datastore, and process them at the speed which your system can handle.

Conclusion

Serverless computing allows you to scale much quicker than with server-based applications, but that means application architects should always consider the effects of scaling to your downstream services. Always keep in mind cost, speed, and reliability when you're building your serverless applications.

Our next post in this series will discuss the different ways to invoke your Lambda functions and how to design your applications appropriately.

TAGS: amazon api gateway, aws lambda, Kinesis Data Stream, RDS, scale, serverless, sqs

Resources

AWS Well-Architected
AWS Architecture Monthly
AWS Whitepapers
AWS Training and Certification
This Is My Architecture
AWS Answers

Follow

Twitter
Facebook
LinkedIn
Twitch
Email Updates

Related Posts

Orchestrating an application process with AWS Batch using AWS CDK

Scale your Remote Access VPN on AWS

Managing resources using AWS CloudFormation Resource Types

In-Depth Strategies from Infosys to Help Customers Re-Platform Mainframes on AWS

How to trigger AWS CodeBuild jobs for selective file changes in AWS CodeCommit

Running SQL on Amazon Athena to Analyze Big Data Quickly and Across Regions

How to analyze well drilling reports using natural language processing

Importing Azure Active Directory users and groups into Alexa for Business Directory using AWS Lambda