AWS Architecture Blog

Understanding the Different Ways to Invoke Lambda Functions

by George Mao | on 02 JUL 2019 | in Advanced (300), Architecture, AWS Lambda, Serverless | Permalink | Share

In our first post, we talked about general design patterns to enable massive scale with serverless applications. In this post, we'll review the different ways you can invoke Lambda functions and what you should be aware of with each invocation model.

Synchronous Invokes

Synchronous invocations are the most straight forward way to invoke your Lambda functions. In this model, your functions execute immediately when you perform the Lambda Invoke API call. This can be accomplished through a variety of options, including using the CLI or any of the supported SDKs.

Here is an example of a synchronous invoke using the CLI:

```bash
aws lambda invoke —function-name MyLambdaFunction —invocation-type RequestRespon
```

The Invocation-type flag specifies a value of "RequestResponse". This instructs AWS to execute your Lambda function and wait for the function to complete. When you perform a synchronous invoke, you are responsible for checking the response and determining if there was an error and if you should retry the invoke.

Many AWS services can emit events that trigger Lambda functions. Here is a list of services that invoke Lambda functions synchronously:

- Elastic Load Balancing (Application Load Balancer)
- Amazon Cognito
- Amazon Lex
- Amazon Alexa
- Amazon API Gateway

- Amazon CloudFront (Lambda@Edge)
- Amazon Kinesis Data Firehose

Asynchronous Invokes

Here is an example of an asynchronous invoke using the CLI:

```bash
aws lambda invoke —function-name MyLambdaFunction —invocation-type Event —payloa
```

Notice, the Invocation-type flag specifies "Event." If your function returns an error, AWS will automatically retry the invoke twice, for a total of three invocations.

Here is a list of services that invoke Lambda functions asynchronously:

- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Simple Email Service
- AWS CloudFormation
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CodeCommit
- AWS Config

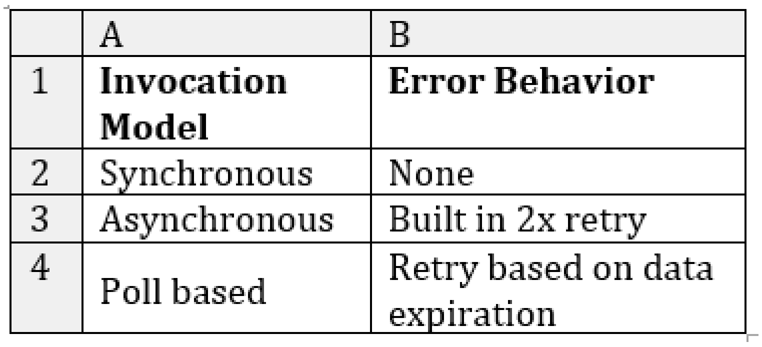
Asynchronous invokes place your invoke request in Lambda service queue and we process the requests as they arrive. You should use AWS X-Ray to review how long your request spent in the service queue by checking the "dwell time" segment.

Poll-Based Invokes

This invocation model is designed to allow you to integrate with AWS Stream and Queue based services with no code or server management. Lambda will poll the following services on your behalf, retrieve records, and invoke your functions. The following are supported services:

- Amazon Kinesis
- Amazon SQS
- Amazon DynamoDB Streams

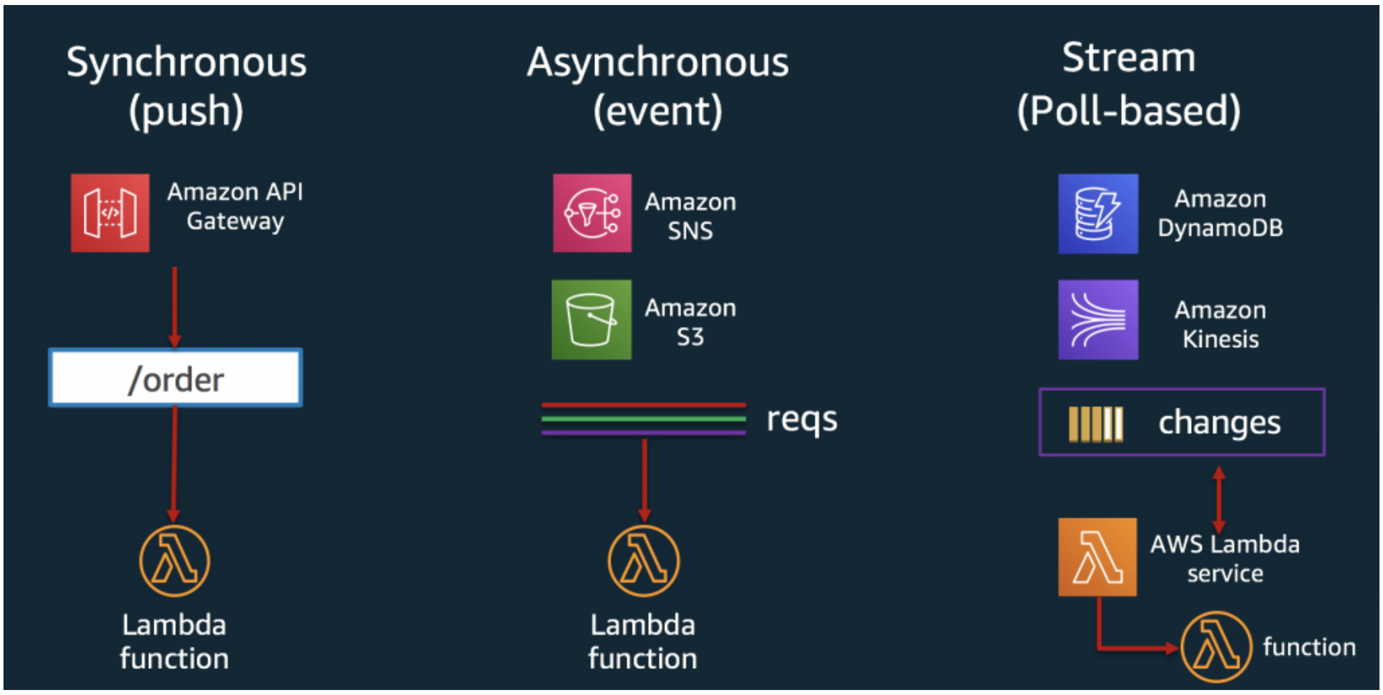
AWS will manage the poller on your behalf and perform Synchronous invokes of your function with this type of integration. The retry behavior for this model is based on data expiration in the data source. For example, Kinesis Data streams store records for 24 hours by default (up to 168 hours). The specific details of each integration are linked above.



	A	B
1	**Invocation Model**	**Error Behavior**
2	Synchronous	None
3	Asynchronous	Built in 2x retry
4	Poll based	Retry based on data expiration

Conclusion

In our next post, we'll provide some tips and best practices for developing Lambda functions. Happy coding!



TAGS: Amazon Alexa, Amazon Kinesis, amazon lambda, cloudformation, CloudWatch (Logs and Events / Rules, DynamoDB, Elastic Load Balancing, Kinesis Firehose, serverless

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