

AWS Lambda Push/Pull Model

There are multiple ways to invoke lambda functions. You can invoke the function directly by calling the invoke API as shown in the demos, or you can set up triggers to invoke the lambda function.

Read about how other AWS Services integrate with AWS Lambda here:

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.html>

When setting up a trigger, there are two main models to keep in mind. First is the push model. The push model, is where a trigger sends an event to lambda, which then invokes your lambda function. When the push model is used, the resource-based policy must allow the trigger to invoke the lambda function.

The second model is the pull model. Some AWS Services being configured as triggers for lambda functions do not push events to lambda, but instead are the source of events. AWS Lambda pulls the event from the event source, and then invokes the lambda function. A common example of a trigger that follows this model is Amazon Simple Queue Service (SQS). SQS allows messages to build up in a queue, and then your code would process those messages. Instead of having SQS invoke the lambda every time it gets a message, the messages build up in the queue and you define an Event Source Mapping as the trigger for the lambda function.

AWS Lambda will pull the events from SQS to the lambda function. You can configure how many messages to pull and at what frequency. Your code does not need to handle polling the queue for messages. Instead, the lambda service does that for you and passes the messages to your code through its event.

Read more about Event Source Mappings here:

<https://docs.aws.amazon.com/lambda/latest/dg/invoke-eventsourcemapping.html>

Asynchronous vs Synchronous Lambda Functions

When invoking a Lambda function via the CLI or the SDK, the type of invocation (synchronous or asynchronous) can be selected based on the API call used. When using an AWS service to trigger the Lambda function, the type of invocation is typically decided by the trigger configured. For example, S3 invokes Lambda asynchronously.

However, there are certain services that allow the type of invocation to be selected. Amazon API Gateway is one of those. The default setting is for API Gateway to invoke Lambda synchronously. For most REST API operations, the client will want a response from the Lambda function so it makes sense to have Lambda executed synchronously during those times. In the case of a long-latency operation, it would make sense to invoke it asynchronously and use another mechanism to let the client know that the operation is completed. That's why it's possible to choose how to invoke Lambda in API Gateway. However, this option is not available when using the Lambda proxy feature of API Gateway. Lambda must be configured with a custom integration. Then, the method execution can be configured to invoke the Lambda function asynchronously using the 'X-Amz-Invocation-Type' header set to 'Event' in the Integration Request. Or the client can decide which type of invocation to use by setting that header in the request to API Gateway. For more information, see:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-integration-async.html>

Security and Compliance with AWS Lambda

You must remember to apply the shared responsibility model when using AWS Lambda. The following documentation shows you how to configure Lambda to meet your security and compliance objectives:

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-security.html>