What is AWS Lambda?

AWS Lambda is a serverless computing service that allows you to run programs without having to worry about provisioning, maintaining, or waiting for servers to be built

AWS Lambda Features

Serverless: Run code without provisioning or maintaining a server.
Automatic Scaling: Scale your applications automatically as per the workload.
Pay per use: Billed per millisecond of use.
Consistency: Performance consistency is achieved by selecting the right memory size for the function.
Language support: AWS Lambda supports multiple programming languages.

Event Source

AWS Lambda Event Source is any AWS Service, custom application, a stream of data or queue which triggers a Lambda function to run.

Lambda Event Sources Type

Push-based model

- Synchronous
- Asynchronous

Pull-based model

- Stream
- Queue

Push-based model:

Other service directly triggers Lambda and tells it to activate when something happens.

- Synchronous: Lamba returns a response back to the event source.
- Asynchronous: For asynchronous invocations, Lambda does handle retries. After Lambda is invoked by the source, it will first place the event into a queue and immediately sends a success response back to the source.

Pull-based model:

Information is flowing through a stream or put in a queue that Lambda periodically polls, and it then goes into action when certain events are detected .

- Stream: Lambda will stop polling while it retries the message.
- Queue: Lambda will return the message to the queue if the invocation fails or times out but will also keep retrying until it’s either successful or expires.

Access Permissions

Security is crucial when dealing with Lambda because of its power to run code and take actions that affect other AWS services.
Two types of security permissions are needed for Lambda:
Invocation permissions and Execution roles.

- Invocation permissions are only needed for push event sources and are granted through an IAM resource policy automatically created when configuring an AWS service as an event source.
- Execution roles grant Lambda permissions to interact with other AWS services. They include an IAM policy defining what Lambda is allowed to do and a Trust policy allowing Lambda to assume the execution role and perform actions on the other service.

Adding the execution role to your Lambda function is the
final step in granting permissions and policies.