International Journal of Information Technology and Computer Engineering
ISSN: 2455-5290
Vol: 04, No. 02, Feb-Mar 2024
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.42.35.42

Exploring Usage of AWS Lambda in Data Processing

Shubhodip Sasmal*

**Senior Software Engineer, TATA Consultancy Services, Atlanta, Georgia, USA.*

*Corresponding Email: *shubhodipsasmal@gmail.com*

Received: 28 September 2023 **Accepted:** 17 December 2023 **Published:** 01 February 2024

Abstract: Cloud computing is changing game all around the industry. It's delivering computing resources, such as storage, processing power without investing in and managing physical infrastructure, offering a more flexible and cost-effective approach to meet computational demands. Serverless computing is a cloud computing execution model where cloud providers automatically manage the infrastructure needed to run applications and scale. This paper examines use of AWS Lambda in Data processing, the leading serverless computing service of Amazon Web Services (AWS). Our study will explore the architecture, integration of AWS Lambda with other AWS services, real-world use cases, proposing best practices for Data processing. This research will provide a complete understanding of AWS Lambda and findings here are a valuable resource for people looking to unleash the benefits of AWS Lambda in a scalable and high-performance cloud-native application architecture.

Keywords: Amazon S3, Cloud, SNS (Simple Notification Service), SQS (Simple Queue Service), KMS (Key Management System), Secrets, Redshift.

1. INTRODUCTION

AWS Lambda a key service for building scalable and cost-effective applications, was introduced by AWS back in 2014. Applications based on Lambda are event-driven architectures and can handle many complex stuffs like scaling and infrastructure management. It can also be useful in understanding the troubleshooting and monitoring tasks. Lambda mostly used in the 4 categories - web applications, web and mobile backends, data processing, parallelized computing tasks and Internet of Things (IoT) workloads. AWS Lambda free tire includes one million free requests per month, 400,000 GB-seconds of compute time per month and 100GiB of HTTP response streaming per month, beyond the first 6MB per request, which are free.

2. RELATED WORKS

There are multiple AWS serverless services are available like AWS Glue, Step Function, Lambda for data processing. Choose the right service is very important while designing the pipeline. Lambda can be useful while loading very small volume of data in a batch into target tables from source. It can be used to trigger AWS Glue- Extract, Transform, Load pipeline to transform and load huge batch data. Lambda function can be called from AWS Step function to load huge volume batch data in multiple loops. AWS Lambda can be integrated with lots of streaming services like Amazon Kinesis or AWS Dynamo DB Streams.

3. METHODOLOGY

3.1 Architecture of AWS Lambda

Few important pieces of AWS Lambda are Lambda function, triggers, IAM Access, logging and monitoring, deployment, concurrency, environment variables, VPC (Virtual Private Cloud) configuration, Lambda destination.

3.1.1 Lambda Function

Lambda function is the heart of AWS Lambda, it can be written in different supported programming language like Python, Node.js, Java and Go. It’s the place where business logic resides for AWS lambda. It can consume events as source, process it per business logic and publish it to other services if require.

3.1.2 Triggers

As Lambda is event-driven, the following are the most common events that can trigger Lambda:

- Amazon S3 event notification for file upload or deletion.
- A Rest API call
- Amazon Dynamo DB table streams
- Amazon Kinesis for real time streaming data
- Any other custom events like generated by Cloud watch

3.1.3 IAM Access

Lambda functions should have the required IAM access roles to perform all the duties. The role of Lambda function should have access of the all the services that it’s consuming from or publishing to.

3.1.4 AWS Lambda Logging and Monitoring

Lambda publishes logs to AWS CloudWatch. Lambda creates CloudWatch Log Group for each function and each invocation creates a log stream. One can get various metrics from CloudWatch like success, failure, performance etc. It can be integrated with AWS Xray for various other metrics and X-Ray tracing should be enabled.

3.1.5 Deployment

Most common way to build Lambda for deployment, is using deployment packages where all the codes and dependencies are packaged in a specific directory structure. In case of deployment packages, external dependencies can be packaged with Lambda code otherwise it can be added using Lambda Layers. Lambda can be deployed using AWS management console, CLI or CI/CD pipeline.

3.1.6 Concurrency

Lambda can scale horizontally by spinning up multiple instances in parallel. It helps to speed up the process and boosts performance. Lambda supports 3 kinds of concurrency:

- Reserved Concurrency: It's the maximum number of concurrent instances that Lambda function can run. This number is max 900 and configuring it is good practice always.
- Unreserved Concurrency: If there is no concurrency number is declared, Lambda function takes default setting of 1000 per account per region as concurrency.
- Provisioned concurrency: This is used to reduce latency and keep Lambda available in warm pool. This is to let AWS know before hand, how many lambda functions process needs at a certain point of time, so that numbers would be readily available whenever require and avoid the cold start time.

3.1.7 Environment Variables

This is very useful feature to pass any variables like KMS key for any encryption, Secret names for database credentials or Log group name to create CloudWatch log etc. You can specify these variables while deploying the Lambda.

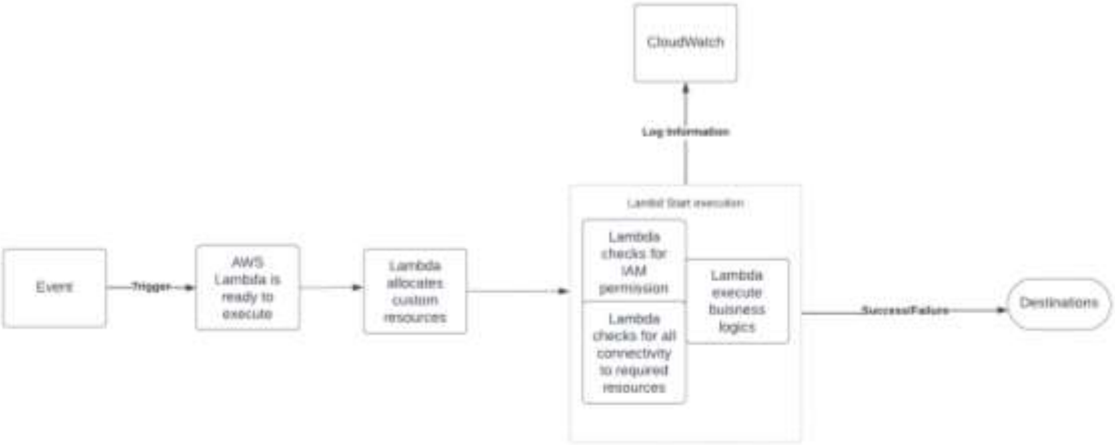
3.1.8 VPC (Virtual Private Cloud) Configuration

Lambda functions are deployed to a VPC. Lambda provides managed resources named Hyperplane ENIs, which it uses to connect from the Lambda VPC to an ENI (Elastic network interface) of a VPC.

3.1.9 Lambda Destinations

Lambda can publish its success or failure status to destinations without writing any code. The destination can be AWS SQS, AWS SNS, Event Bridge or another Lambda.

Lambda Architecture

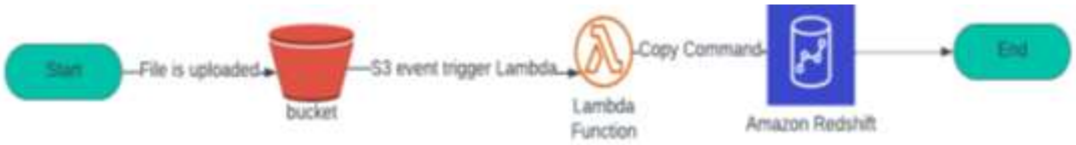

CloudWatch
Log Information
Lambd Start execution
Event
Trigger
AWS Lambda is ready to execute
Lambda allocates custom resources
Lambda checks for IAM permission
Lambda checks for all connectivity to required resources
Lambda execute buisness logics
Success/Failure
Destinations

4. RESULTS AND DISCUSSION

Lambda can be used below different ways in data processing.

1. Transform and Load Data into Database Using Lambda

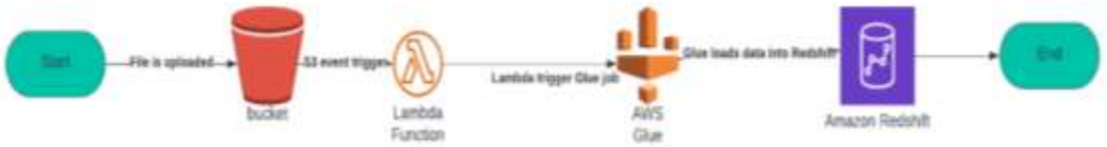
AWS Lambda can transform data per business logic written in lambda function. AWS Lambda can also be used to load small volume data into database like AWS Redshift. Lambda can execute COPY command to load data from S3 to database. It's quite easy to integrate Redshift with AWS Lambda once IAM access and database credentials are handled correctly. Lambda has a limitation of timing out in 900 seconds and due to that data processing or data loading volume shouldn't be too high so that it may cross 900 seconds. For larger volume batch Lambda can trigger AWS Glue job or can be called from AWS Step Function.



2. Invoke AWS Glue using Lambda

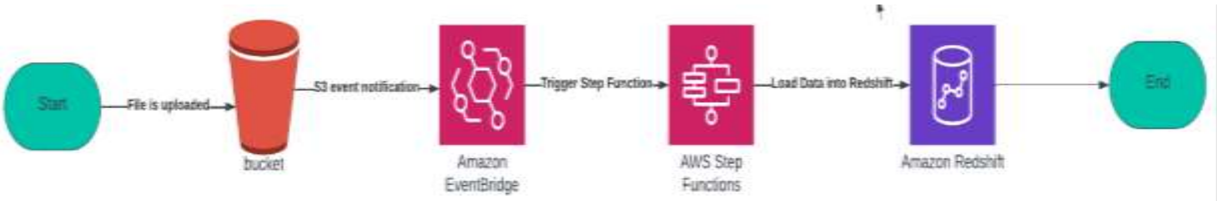
As Lambda has a timeout limitation, AWS Glue is used widely to process ETL (Extract, Transform, Load) pipeline. AWS Glue is a serverless, fully managed AWS offerings and it's able to discover, catalog, transform, and move data between data stores. It has some very exciting feature calls Crawler, Data Catalogs etc. Crawler can automatically discover, catalog metadata from sources and creating metadata tables in AWS Data Catalogs. Data Catalog supports S3, Redshift, RDS and other JDBC compatible databases as Source. AWS Lambda can be used to trigger Glue job based on some event. As an example, once file will be uploaded at S3, Lambda will be triggered and the Lambda will invoke a Glue job. Lambda can pass all the required information while calling Glue job as parameters. Here is the dummy python code to trigger Glue job from a Lambda function:

```
import boto3
def lambda_handler(event, context):
glue = boto3.client('glue')
job_name = 'dummy-glue-job-name'
response = glue.start_job_run(JobName=job_name)
return {
'statusCode': 200,
'body': f'Started AWS Glue job run: {response["JobRunId"]}'
}
```



3. Lambda and Step Function

AWS Step Functions provides serverless orchestration for modern applications. Orchestration centrally manages a workflow by breaking it into multiple steps, adding flow logic, and tracking the inputs and outputs between the steps. It is very popular service that can call AWS Lambda to load huge volume data in multiple loops. Step function can build serverless workflows by defining a series of state machines that represent the steps in your workflow. Each state of step function performs a task and it can decide next state based on condition. Step function has a choice state to decide next step based on previous step. So, Step function can call same Lambda function multiple times to load huge volume data. Suppose if a batch has 100,000 data and Lambda only can load 25,000 data into database, step function should call Lambda 4 times and each time 25,000 data would be loaded to database. Choice state should indicate that 4 Lambda iteration required to entire batch. For invocation Step function can be integrated with AWS Event Bridge to access any event notification like S3 event trigger and can start execution.

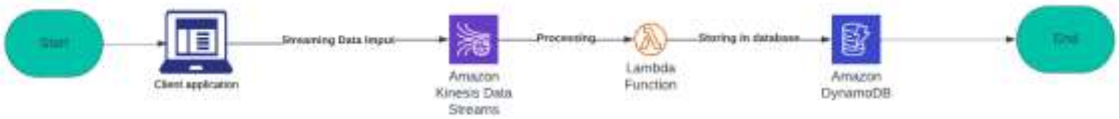


4. Streaming Data Processing using Lambda

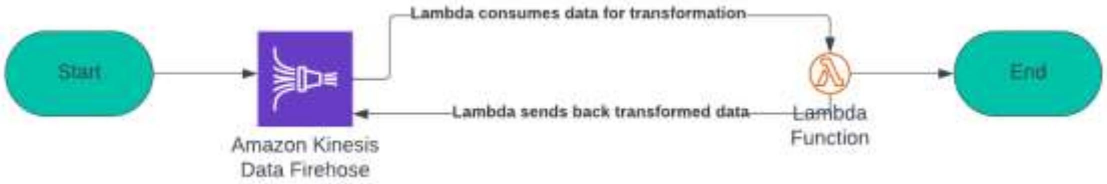
Lambda can be useful while processing real-time streaming data.

Lambda with Amazon Kinesis Data Stream

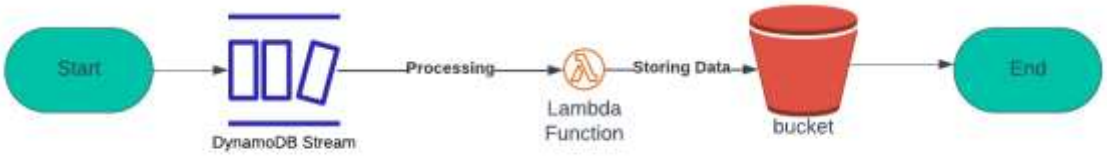
Amazon Kinesis Data Streams is a massively scalable, highly durable data ingestion and processing service optimized for streaming data. You can configure hundreds of thousands of data producers to continuously put data into a Kinesis data stream. Data will be available within milliseconds to your Amazon Kinesis applications, and those applications will receive data records in the order they were generated. AWS Lambda and Amazon Kinesis can process real-time streaming data like click stream, transaction order, IoT device, log stream etc. Lambda can read data from Kinesis Streams and store it to some database (Dynamo DB in below figure). Multiple processes can use this data from database per their need. The IAM role used for this purpose, should have all the required access of accessing Kinesis Data Streams and DynamoDB.



Lambda with Amazon Kinesis Data Firehose
Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service, Amazon OpenSearch Serverless, Splunk, and any custom HTTP endpoint or HTTP endpoints owned by supported third-party service providers, including Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Coralogix, and Elastic. Kinesis Data Firehose is part of the Kinesis streaming data platform, along with Kinesis Data Streams, Kinesis Video Streams, and Amazon Managed Service for Apache Flink.
Lambda can be invoked by Kinesis Data Firehouse for transforming incoming data and delivered to destination.


Start
Lambda consumes data for transformation
Amazon Kinesis Data Firehose
Lambda sends back transformed data
Lambda Function
End

DynamoDB Stream and Lambda Function
DynamoDB Streams capture a time-ordered sequence of item-level modifications in a DynamoDB table and can trigger events to respond to these changes. A DynamoDB stream is an ordered flow of information about changes to items in a DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table. It allows you to build applications that respond to changes in your DynamoDB table in real-time. Lambda can be triggered by Dynamo DB Stream, do the modification and store the data into S3 or some other places.



5. CONCLUSION

AWS Lambda is a revolutionary technology of serverless computing that offers unparalleled scalability, cost-efficiency, and flexibility to us. Throughout this research paper, we have looked into use of AWS Lambda in data processing. We have explored the batch processing and real-time data processing both. AWS Lambda is very cost effective due to it's pay as you go model and seamless integration with other AWS services enhances the overall capabilities of Lambda. AWS Lambda offers numerous advantages however it is essential to acknowledge some challenges like potential cold start latency and certain limitations on execution time and resource allocation. In conclusion, AWS Lambda is not just a technology choice but a strategic decision for organizations aiming for serverless computing. As cloud computing evolved, AWS Lambda is a front runner for its capabilities and one of the most useful service in serverless cloud computing.

6. REFERENCES

1. Amazon Web Services, "Processing streaming data using AWS Lambda," [Online]. Available: https://pages.awscloud.com/Processing-Streaming-Data-with-AWS-Lambda_2021_0408-SRV_OD.html
2. Amazon Web Services, "Lambda config for streaming services," [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/configuration-response-streaming.html
3. Amazon Web Services, "Lambda architecture," [Online]. Available: https://lumigo.io/learn/aws-lambda-architecture/
4. Amazon Web Services, "Lambda concurrency," [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html
5. Amazon Web Services, "Lambda stream processing," [Online]. Available: https://github.com/aws-samples/lambda-refarch-streamprocessing
6. Amazon Web Services, "Firehose," [Online]. Available: https://docs.aws.amazon.com/firehose/latest/dev/what-is-this-service.html
7. Amazon Web Services, "Streaming analytics," [Online]. Available: https://aws.amazon.com/blogs/compute/using-aws-lambda-for-streaming-analytics/
8. Amazon Web Services, "Kinesis data streams," [Online]. Available: https://aws.amazon.com/kinesis/data-streams/getting-started
9. Amazon Web Services, "Step Functions," [Online]. Available: https://aws.amazon.com/step-functions/features/
10. Amazon Web Services, "AWS Lambda"[Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html