PROJECT NAME - AWS BATCH DATA PIPELINE PROJECT.

OBJECTIVES OF THIS PROJECT- To design and implement a scalable data pipeline for smooth data flow from AWS S3 bucket to storing it into AWS DynamoDB using Lambda Trigger and Event Bridge. The Data is processed and Transformed through AWS glue before dumping into DynamoDB.

Below are the some of the goals or use cases that I defined when I started this project.

* Build and understand a data processing framework in AWS used for stream and batch data loading by companies.
* Setup and understand cloud components involved in data streaming and batch processing.
* Understand how to identify failures in a data processing pipelines and how to build systems that can handle the failures and errors better.
* Understand how to approach or build a data pipeline from scratch in AWS.

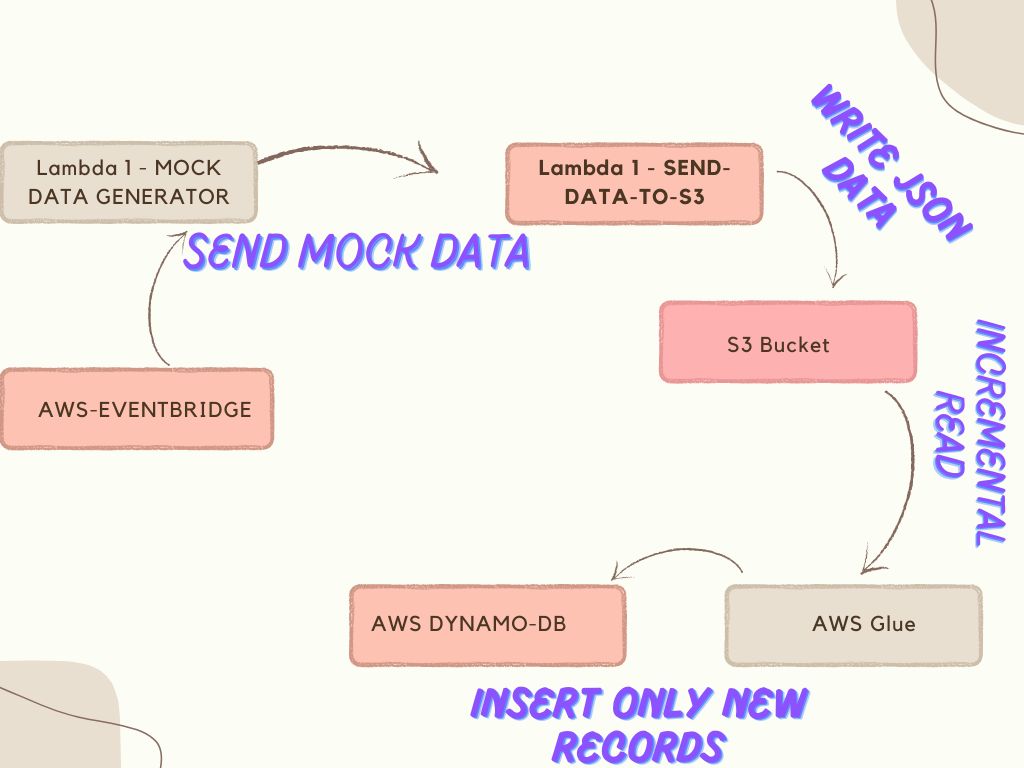
PROJECT WORKFLOW:- First of all I have created a AWS lambda function to generate some mock data. Then another lambda function is created for transferring the data to S3 bucket in the form of JSON. Then using AWS glue crawler I have tried to upload the data from S3 to DynamoDB with spark job only allowing to dump new records in the Database.

**AWS SERVICES USED IN THE PROJECT:-** I have used several AWS services for creating the data pipeline in this project. A brief overview of the services are:-

1. AWS Lambda:- AWS lambda is a server-less computing AWS service that executes our code in response to events and manages the underlying computing resources effortlessly. Lambda service comes handy when collecting the raw data is essential. Data Engineers can develop a lambda function to access an API endpoint, obtain the result,process the data and save it to other storing services.
2. AWS S3:- AWS S3 stands for Simple Storage service. It is a data lake that can store any volume of data from any part of the internet. It is an incredibly scalable,quick and affordable option. Data engineers can effectively create web based clouds solutions that expand automatically and have flexible setups owing to S3.
3. AWS Event Bridge:- [Amazon EventBridge](https://aws.amazon.com/eventbridge/" \t "https://medium.com/awesome-cloud/_blank) is a server-less, fully managed, and scalable event bus that enables integrations between AWS services, Software as a services (SaaS), and other applications. Event Bridge makes it easier to connect applications.We can ingest, filter, transform and deliver events without writing custom code.
4. AWS Glue :- AWS glue is a fully managed ETL service for easily and affordably processing, improving and migrating data between different data stores and data streams. Data engineers interactively analyze and process the data using AWS glue interactive sessions. We can visually develop, run and monitor ETL workflows in AWS glue studio with few clicks.
5. AWS DynamoDB :- AWS DynamoDB provides an alternative to relational database systems by using several data types such as document,graph, key-value, memory and search. They can use it store semi-structured data with a unique key.
6. Amazon IAM:- AWS identity and Access Management is another popular AWS service that enables us to control access to AWS resources. The IAM service offers features for managing authorizations for actions against AWS services like Amazon Sagemaker and S3.

PROJECT ARCHITECTURE:- The below diagram depicts the entire project workflow of the entire batch data processing pipeline using AWS services.

First of All, a **Lambda Function(Lambda-1)** is created which generates a mock data in the form of dictionary. This Lambda-1 function is triggered by **AWS Eventbridge** (Event generation service) which hits this lambda every 5 minutes.



Now another **lambda function(Lambda-2)** is created which is connected to the previous lambda function. The first lambda function basically sends the mock data to the second lambda function randomly which is further connected to a **S3 Bucket.** This lambda function writes JSON data to S3 bucket.

Now the S3 bucket is connected with the **AWS Glue Job** which performs the incremental read.And the Glue Job is finally connected to **DyanmoDB.** This glue job insert only new records to the DynamoDB table. A Spark application is created in AWS Glue which creates two dataframes. First dataframe is what is coming from S3 to Glue and the second dataframe is what is going from Glue to DyanmoDB. And the a left join operation is performed that finally insert only new records in the DynamoDB table.