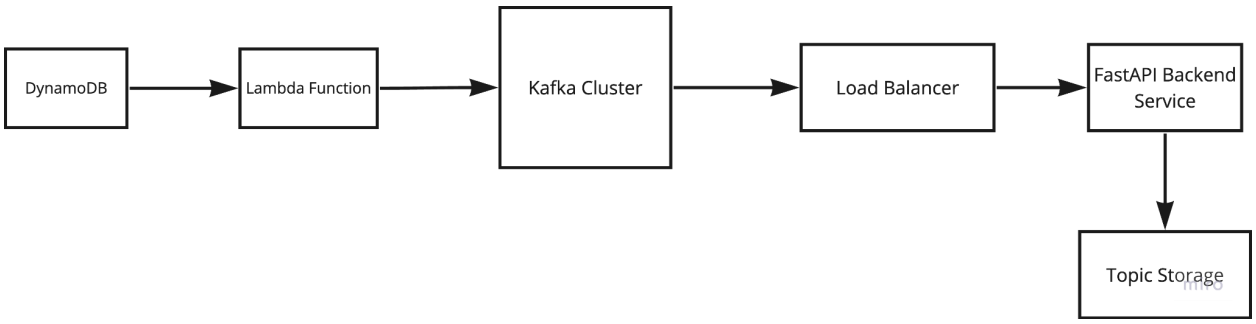
Topic Modeller Service Design Document

Description

This document proposes the design to operationalise the topic modeller service in order to scale for millions of URLs.



High-Level Design for Topic Modeller Service

Components

1. DynamoDB- We will use NoSQL storage to manage all URL storage.
2. Lambda Function - The Lambda function can be used to trigger on the basis of data updation events in dynamoDB.Its job will be to publish the new URLs to our Kafka Cluster.
3. Kafka Cluster - The Kafka Cluster will act as a messaging broker and will cater to the scale of millions of URLs.
4. FastAPI Backend - Our FastAPI-based backend service will consume the data from Kafka and extract the relevant topics after scraping the URLs received via Kafka.
5. Topic Storage - We can use any NoSQL-based storage to store the relevant topics extracted from the URL.

Scaling

1. AWS Lambda is designed to automatically scale based on the number of incoming triggers or events. When the number of triggers from DynamoDB increases to a substantial level, AWS Lambda can handle the increased workload by scaling

horizontally, which means it creates more instances (containers) of your Lambda function to process the events concurrently.
2. FastAPI backend service can be scaled horizontally by running a number of stateless replicas of the service in front of a load balancer to balance the load between each of them.
3. Kafka cluster scales by adding more Kafka broker nodes to the cluster to distribute the message processing and storage load.

Efficiency

1. We can increase the efficiency of our backend service by refining the process of parsing the HTML page
2. The efficiency of the LDA model can be increased by following steps
    a. Data Preprocessing
    b. Feature Selection
    c. Optimized Model Parameters
    d. Parallelization