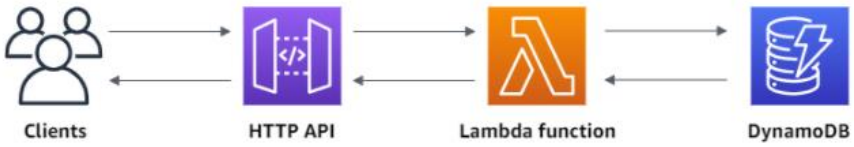
DynamoDB : (No SQL Database)

What is DynamoDB

- DynamoDB is fully managed NoSQL database service that supports key–value and document data structures
- It delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications
- Used where it require milli second read latency
- Scale to more than 10 trillion requests per day and can support peaks of more than 20 million requests per second
- Its serverless, so no server management is needed
- Stores data in the form of key-value and wide column
- It stores data in partitions
- No need to specify what amount of storage you need to store your data, it increases as it grows
- Launched in a region and automatically data replicated and stored in multiple AZs by default (3AZs) in a region
- Steams can be used to replicate data to a table in other region (Near Real Time)
- Global tables can be used to replicate data to another table in another region (Asynchronous)
- Doesn't live on VPC, VPC endpoints can be used to access privately
- Supported with data encryption at-rest using KMS and in-transit using SSL. By default DynamoDB encrypts all data
- Access is IAM based

Sample Architecture :



Key Components :

- **Tables :**
  - Table stores the data in a tabular format. Collections of data or items. For example example table called People that you could use to store personal contact information
- **Items :**
  - Items are the data stored in a table. For example in a people table, each item represents a person
- **Attributes :**
  - The fields or columns of each item, for example in a people table contains attributes called PersonID, LastName, FirstName, and so on
- **Primary Key :**

    - The primary key uniquely identifies each item in the table, so that two different items cannot have the same key. Dynamo has two different primary keys
        - **Partition Key :**
            - A simple primary key, composed of one attribute known as the partition key. For example for a Person table “PersonID” can be the primary key
            - Let’s assume you have a Music table and you want your users to be able to find information about songs and the artists who recorded them. One approach would be to use Artist as a partition key, to serve as the simple primary key for items in the table
        - **Partition Key and Sort Key :**
            - It is composed of two attributes, the first attribute is the partition key, and the second attribute is the sort key
            - In a table that has a partition key and a sort key, it's possible for two items to have the same partition key value. However, those two items must have different sort key values
            - For example, what if you wanted to store two different songs by the same Artist? Let's choose a composite primary key – one that uniquely identifies each song by the combination of Artist and SongTitle. We'll use SongTitle as a sort key, which enables you to query the Music table for all of the songs by a particular artist, and then retrieve those songs
- **DynamoDB Streams :**
    - It is an optional feature that captures data modification events in DynamoDB tables
    - Each event is represented by a stream record. If you enable a stream on a table, DynamoDB Streams writes a stream record whenever one of the following events occurs
        - A new item is added to the table
        - An item is updated
        - An item is deleted from the table

Example Solution :

- https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-dynamo-db.html