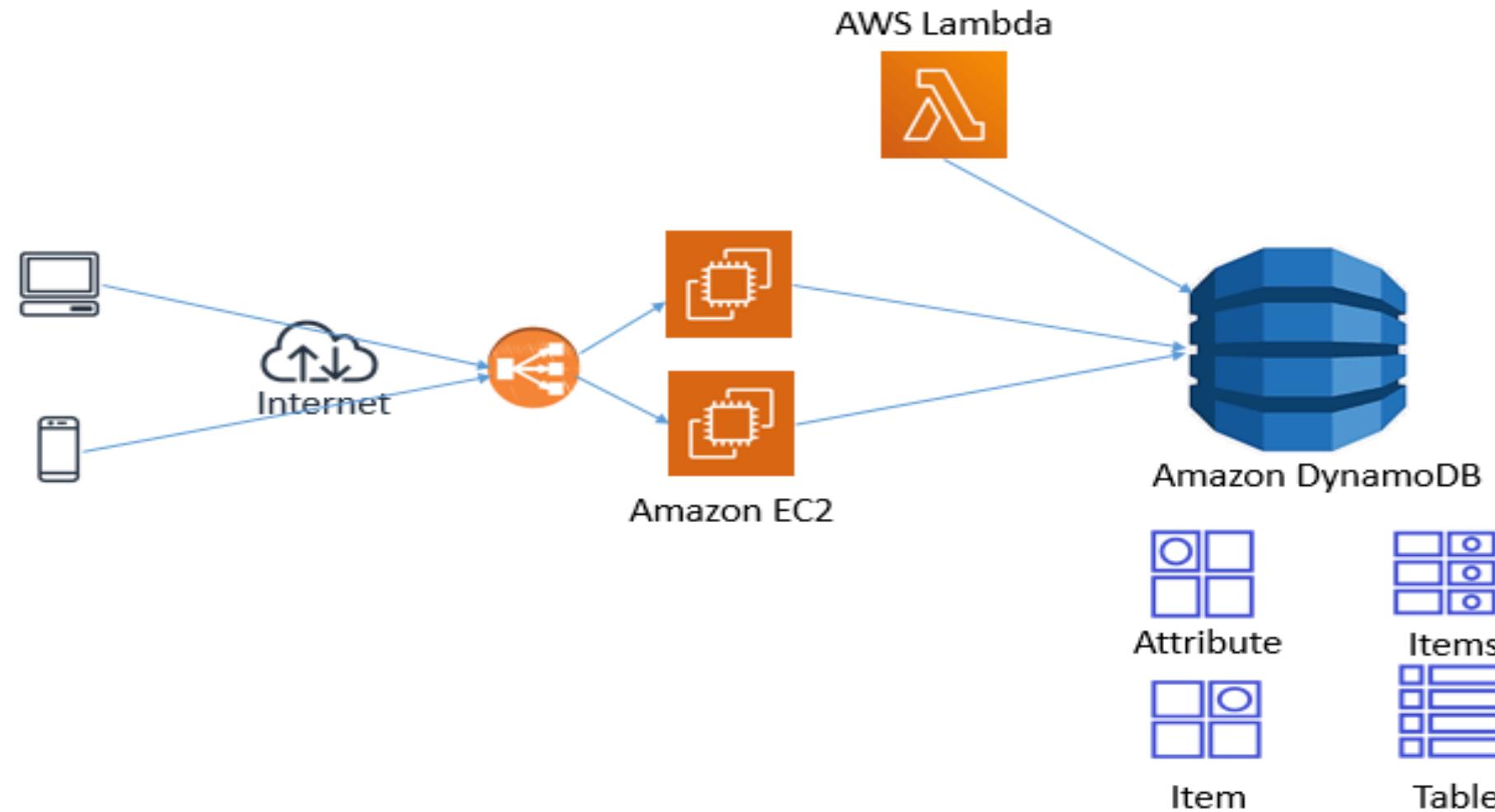


iamneo



Amazon DynamoDB

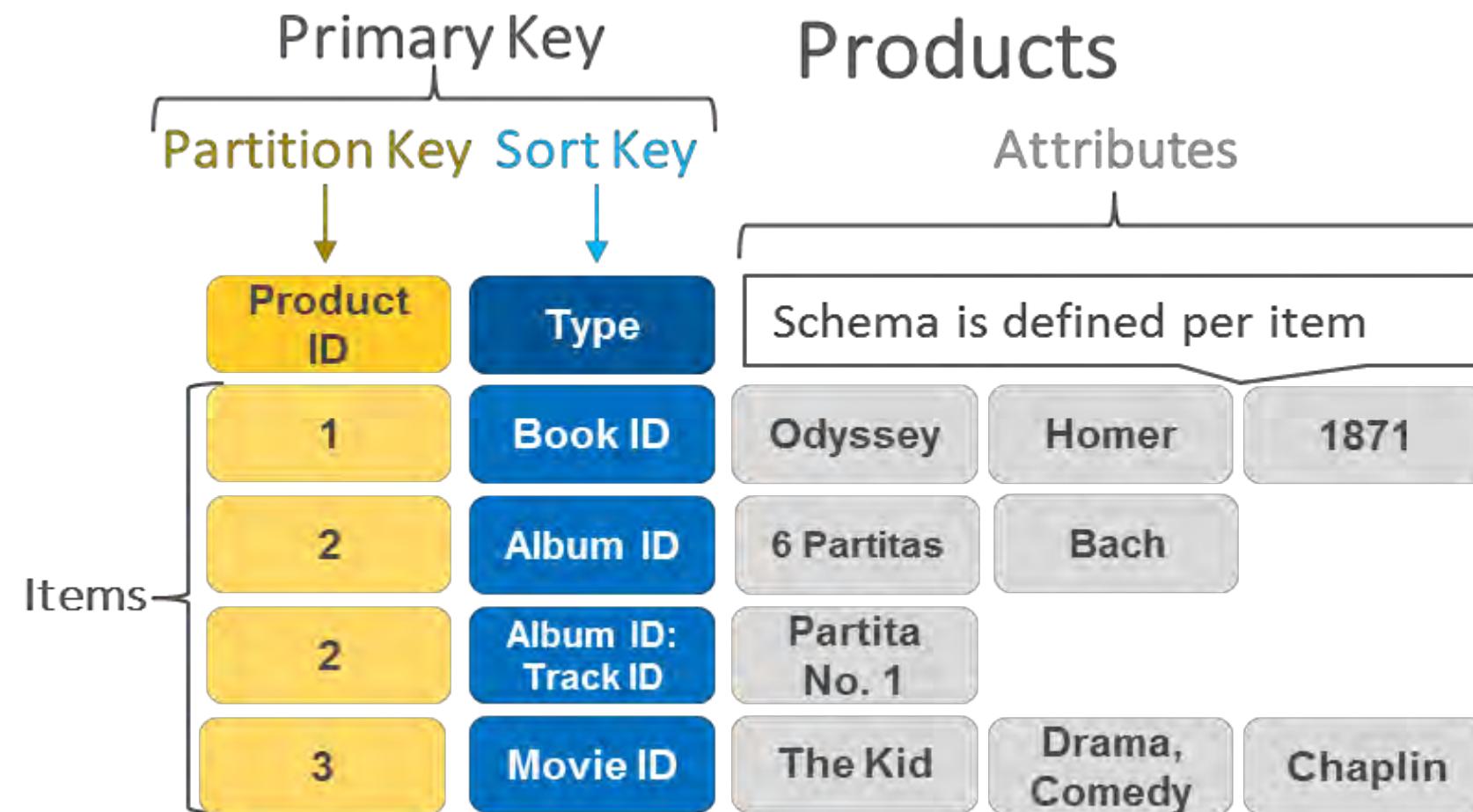
Introduction to Amazon DynamoDB



Introduction to Amazon DynamoDB

- AWS Dynamo DB is a No SQL Database which is built to support No SQL compatible database in cloud environment.
- AWS Dynamo DB is fully managed server less No SQL database service it means you do not need to take care of any server/infrastructure, AWS does take care it for you. Dynamo DB is highly scalable, available and durable database service. It is a key value pair database store.
- Dynamo DB can also store document.

Key-Value Data Model and Benefits

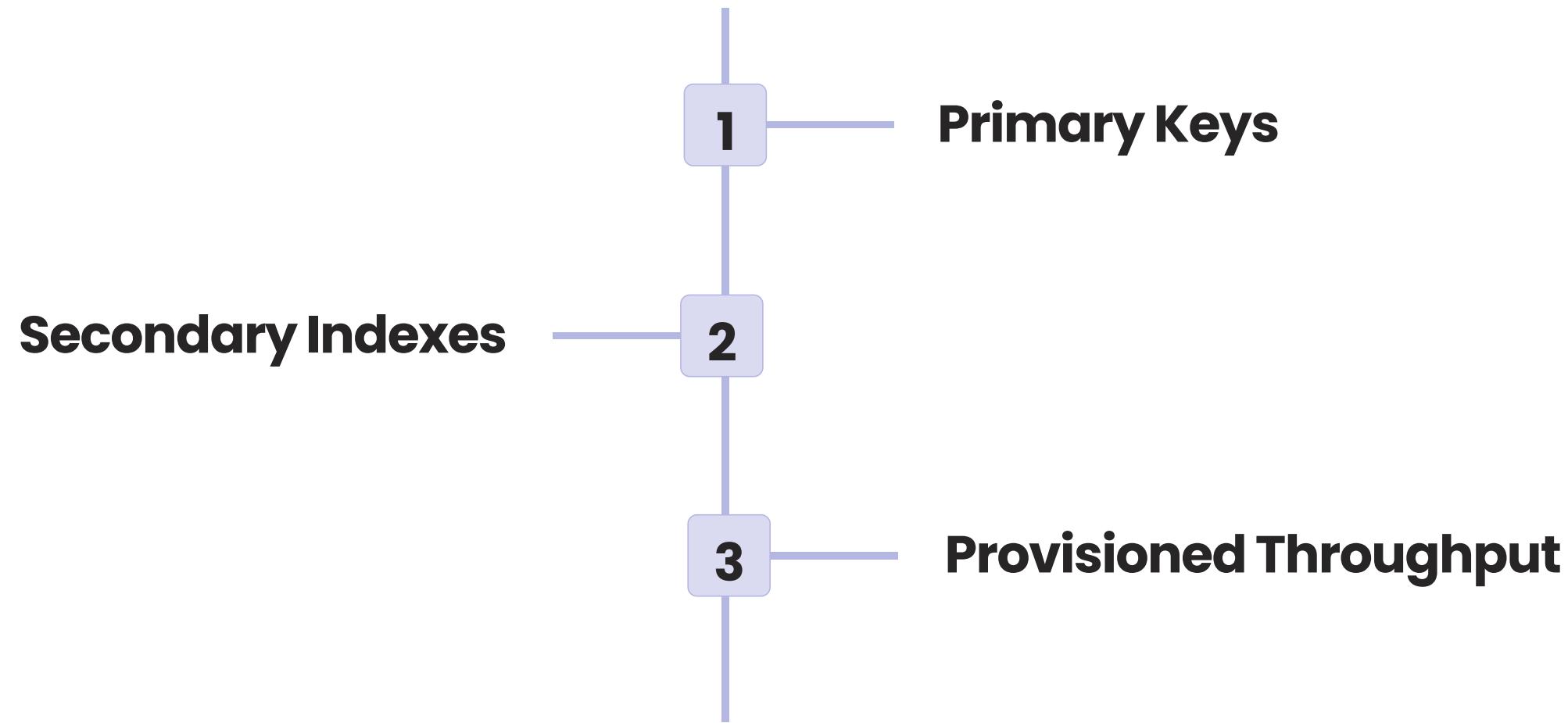


The primary key uniquely identifies each item in the table, and secondary indexes can be created.

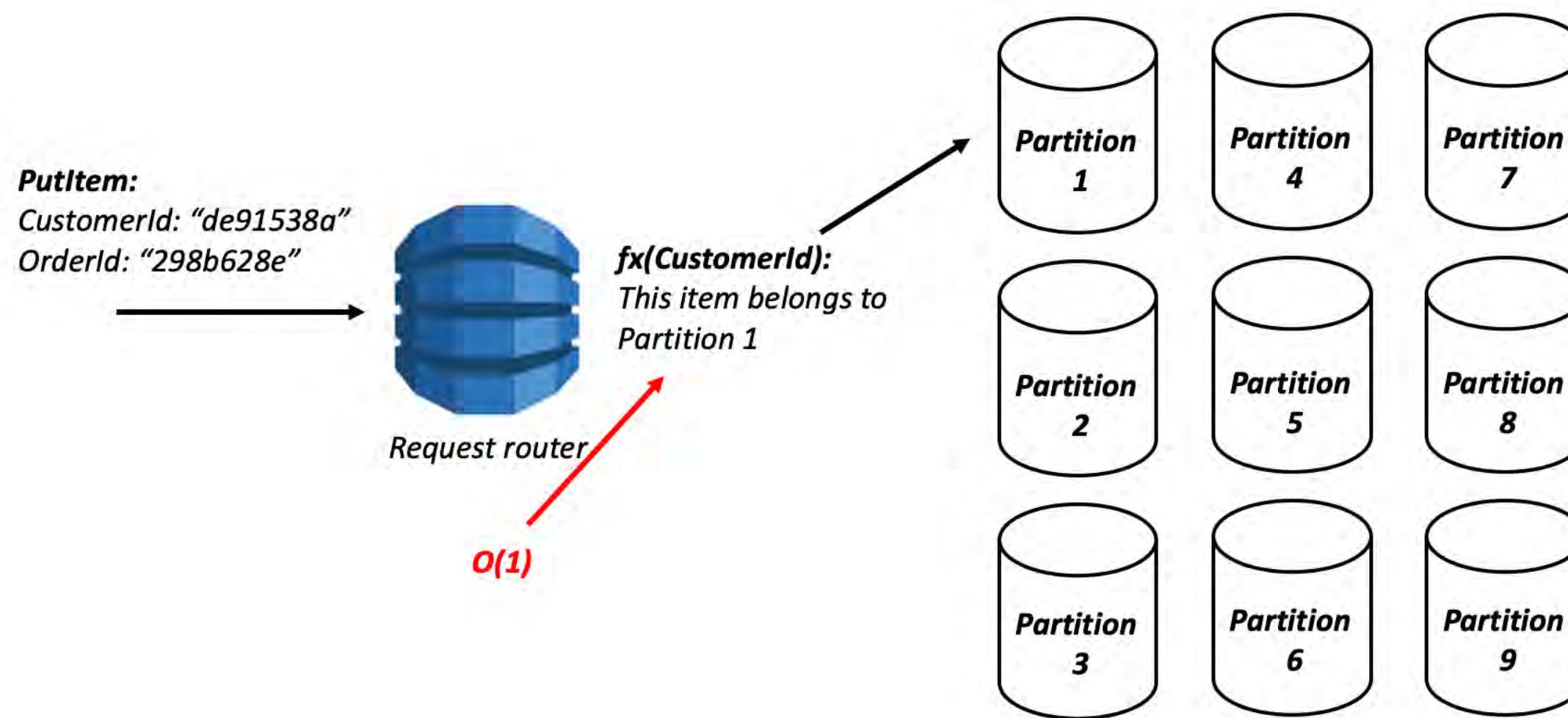
Benefits of DynamoDB

- High performance
- Fully managed
- On-demand pricing and automatic scaling
- Cost-effective
- Durable
- Highly available.

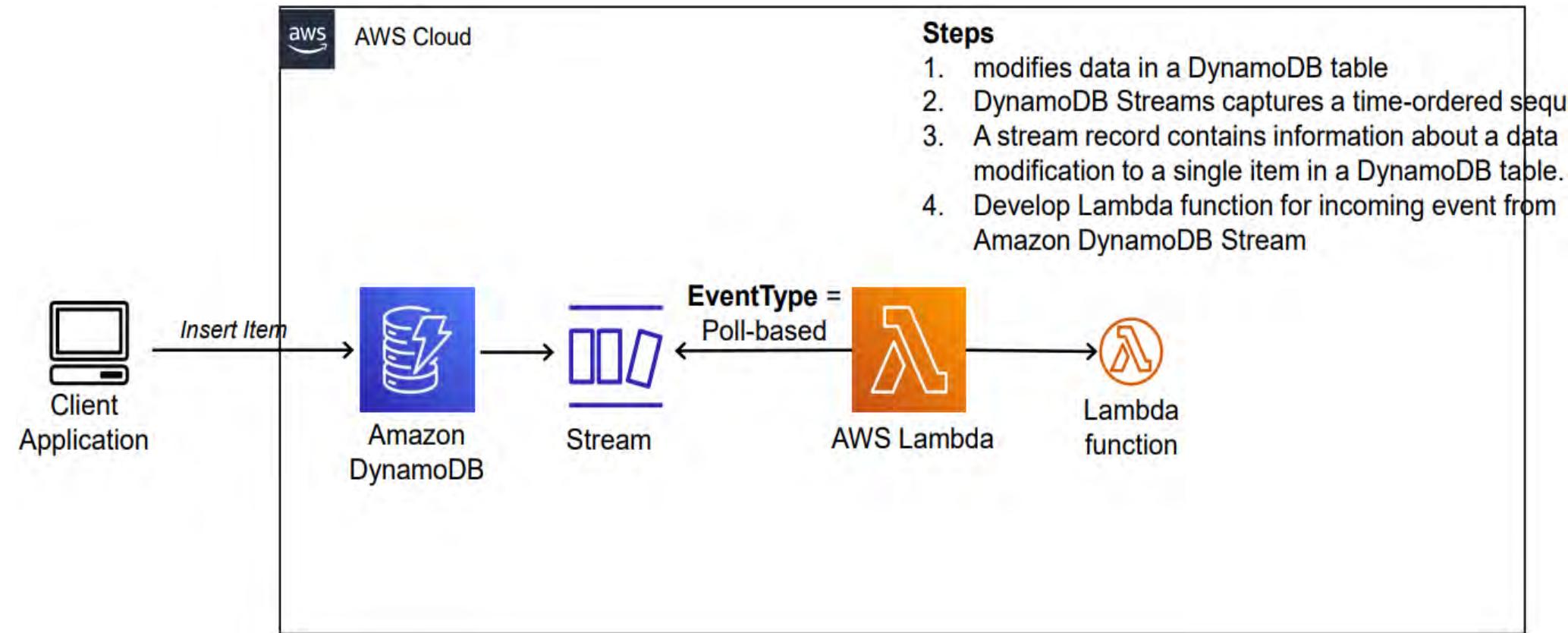
Designing DynamoDB Tables



Designing DynamoDB tables with proper primary keys



Using DynamoDB Streams for capturing real-time data changes



DynamoDB Streams – Using AWS Lambda to Process DynamoDB Streams for Change Data Capture of DynamoDB Tables.

Best Practices for Modeling Data in DynamoDB

Start with a clear understanding of your application

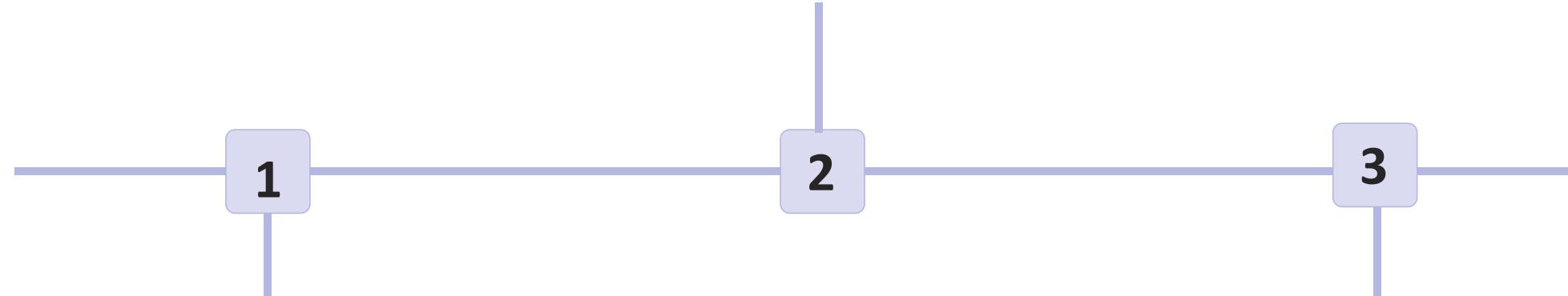
Use maximum capacity units efficiently

Optimize your table structure

Keep an eye on your costs

Querying DynamoDB

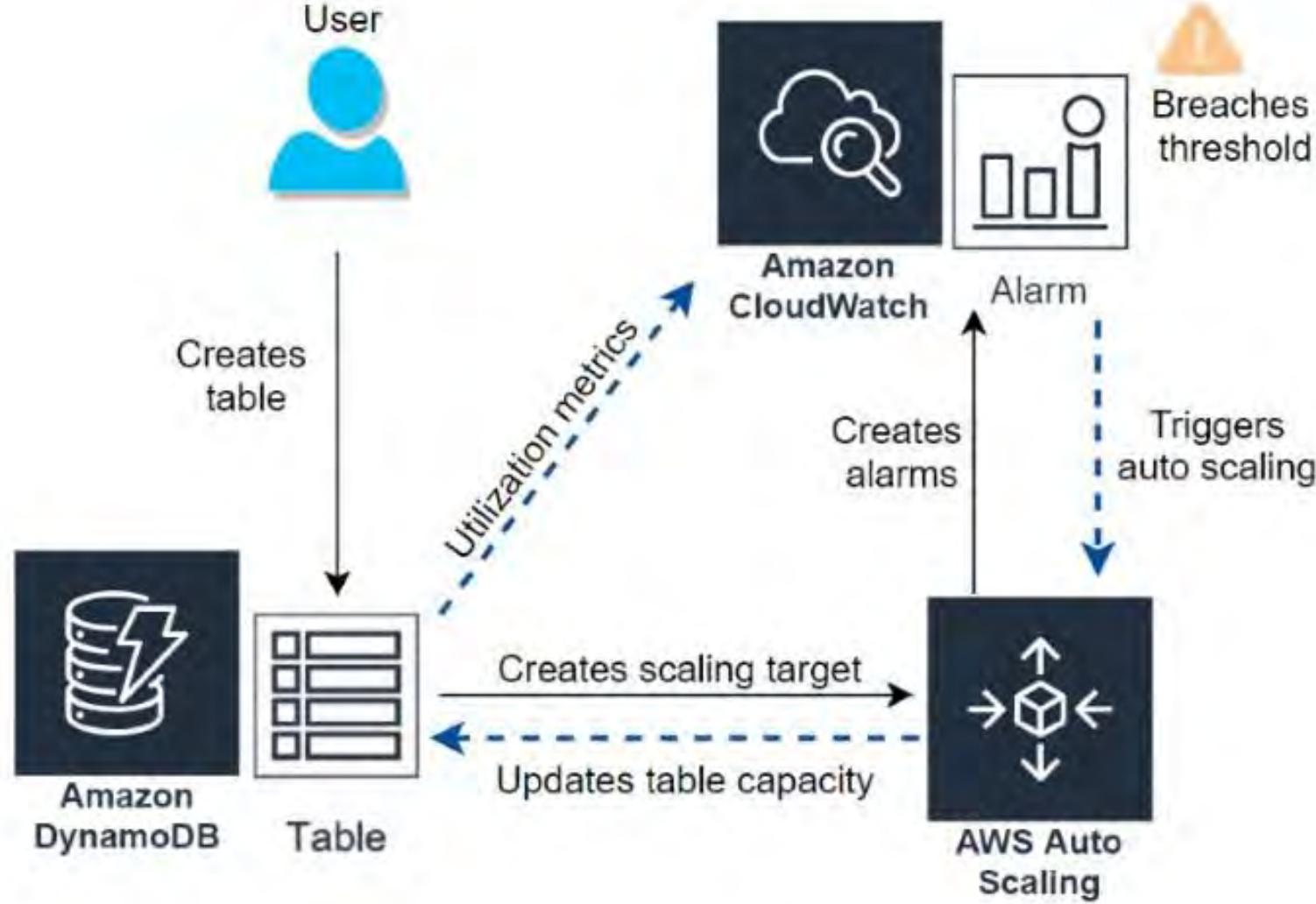
Range-based Operations



Key-based Operations

Filter Operations

Scaling DynamoDB



- **Partitioning**
- **Global Secondary Indexes**
- **Auto Scaling**

Performing CRUD Operations in DynamoDB

1 Create

Use PutItem to add a new item to your table, specifying its partition key and any additional attributes.

2 Read

Use GetItem and BatchGetItem to retrieve specific items by their primary keys.

3 Update

Use UpdateItem to modify an existing item or create a new one if it doesn't exist, specifying the attributes to update or delete.

4 Delete

Use DeleteItem and BatchWriteItem to remove items from your table, specifying their primary keys.

Querying Data in DynamoDB

- 1
- 2

Query Operations

Use Query to retrieve specific items by their partition and sort key values, and filter by any additional attributes.

Scan Operations

Use Scan to retrieve all items in a table or a specific set of items, and filter by any additional attributes.

Configuring auto scaling for DynamoDB tables based on capacity needs

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key String ⓘ

Add sort key

Table settings

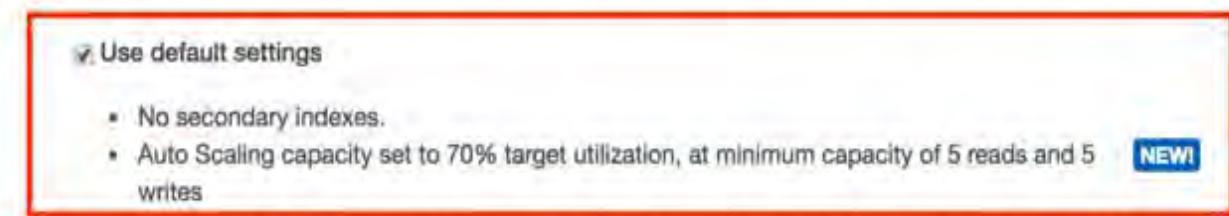
Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

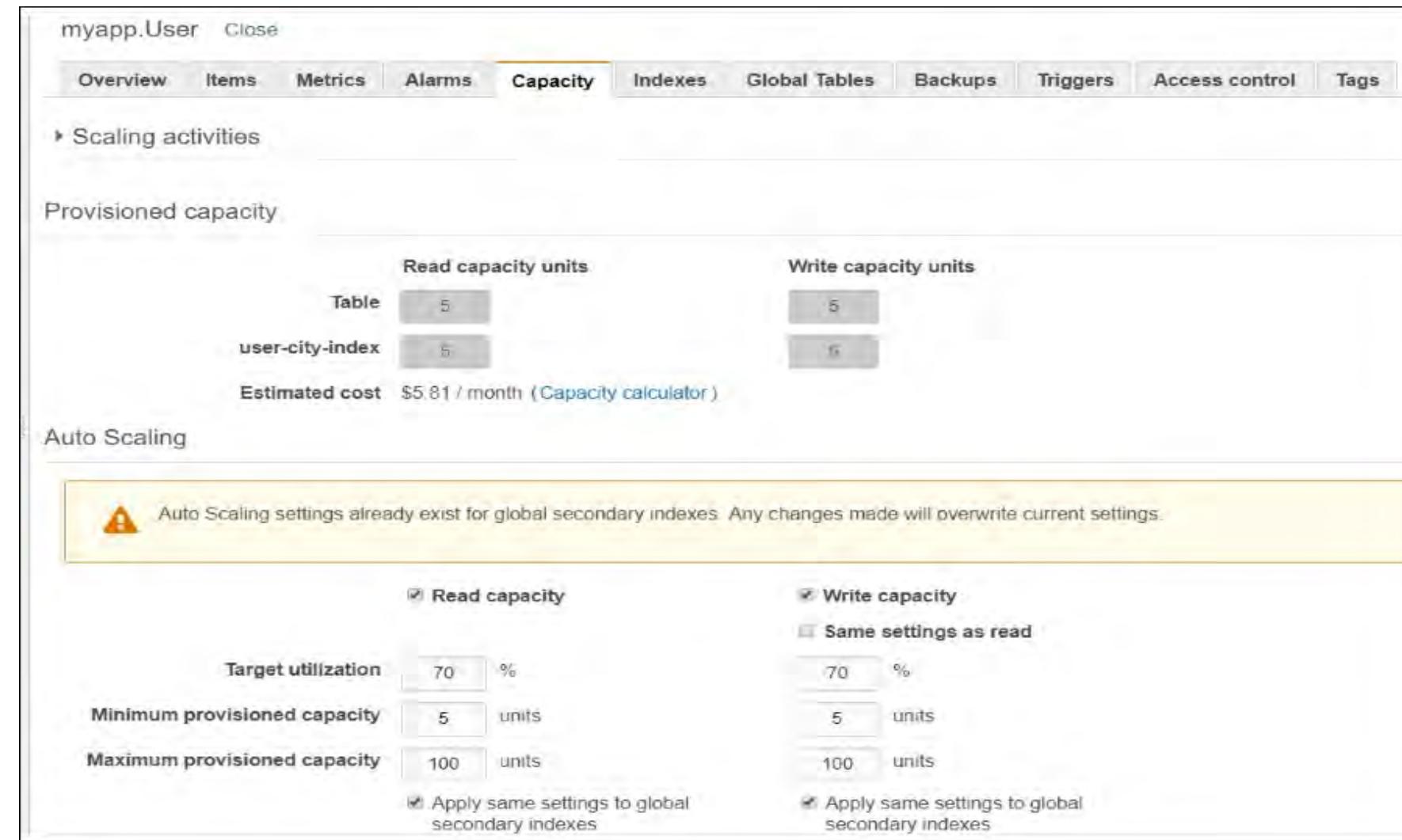
- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes NEW!

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

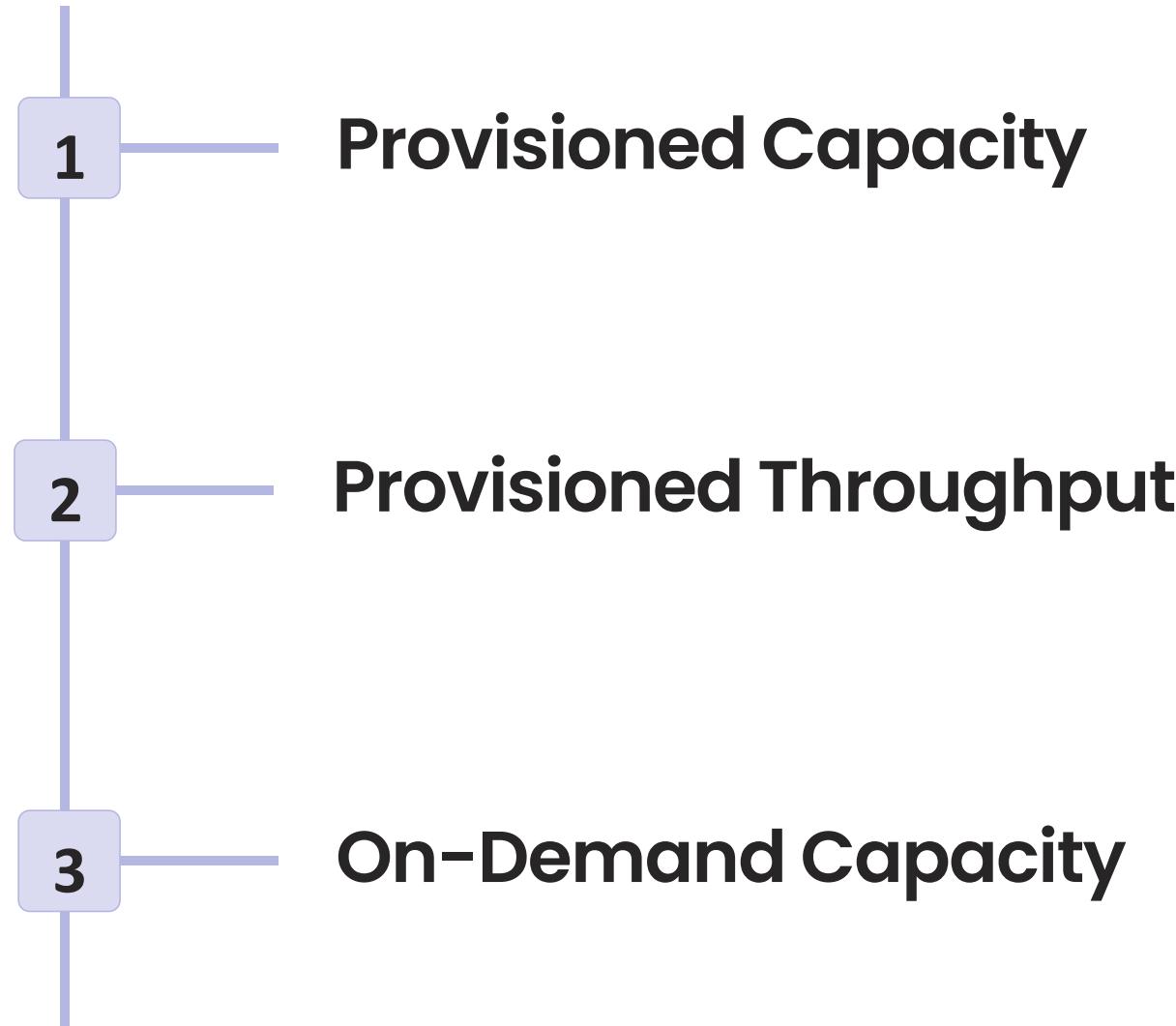
Cancel Create



Configuring auto scaling for DynamoDB tables based on capacity needs



Capacity Needs and Scaling Strategies



Utilizing Global Tables for Multi-Region Replication

What are Global Tables?

Global Tables enable automatic replication of DynamoDB tables across multiple AWS Regions.

Benefits of Multi-Region Replication

It helps prevent data loss from disasters, provides low-latency data access, and enables global reads and writes.

Global Scalability with Global Tables

Leverage Global Tables for scalable, multi-region access to your DynamoDB tables.

How to Configure Global Tables

