

Week 3 Notes and Resources

KEY TOPICS

Security, State, and Advanced Concepts

This week we focused on security, including encryption and access, expiration of items,

Security

Encryption and Compliance

All user data stored in Amazon DynamoDB is fully [encrypted at rest](#). DynamoDB encryption at rest provides enhanced security by encrypting all your data at rest using encryption keys stored in [AWS Key Management Service \(AWS KMS\)](#). This functionality helps reduce the operational burden and complexity involved in protecting sensitive data. With encryption at rest, you can build security-sensitive applications that meet strict encryption compliance and regulatory requirements.

VPC Endpoints

For security reasons, many AWS customers run their applications within an Amazon Virtual Private Cloud environment (Amazon VPC). With Amazon VPC, you can launch Amazon EC2 instances into a virtual private cloud, which is logically isolated from other networks—including the public Internet. With an Amazon VPC, you have control over its IP address range, subnets, routing tables, network gateways, and security settings. More information about using Amazon VPC endpoints for DynamoDB can be found [here](#).

Authentication and Access Control for Amazon DynamoDB

Access to Amazon DynamoDB requires credentials. Those credentials must have permissions to access AWS resources, such as an Amazon DynamoDB table or an Amazon Elastic Compute Cloud (Amazon EC2) instance. Details on DynamoDB authentication and access control can be found in the [developers guide](#).

State, and advanced concepts

Time to Live (TTL)

[Time To Live \(TTL\) for DynamoDB](#) allows you to define when items in a table expire so that they can be automatically deleted from the database.

TTL is provided at no extra cost as a way to reduce storage usage and reduce the cost of storing irrelevant data without using provisioned throughput. With TTL enabled on a table, you can set a timestamp for deletion on a per-item basis, allowing you to limit storage usage to only those records that are relevant.

TTL is useful if you have continuously accumulating data that loses relevance after a specific time period. For example: session data, event logs, usage patterns, and other temporary data. If you have sensitive data that must be retained only for a certain amount of time according to contractual or regulatory obligations, TTL helps you ensure that it is removed promptly and as scheduled.

Global tables

[Amazon DynamoDB global tables](#) provide a fully managed solution for deploying a multi-region, multi-master database, without having to build and maintain your own replication solution. When you create a global table, you specify the AWS regions where you want the table to be available. DynamoDB performs all of the necessary tasks to create identical tables in these regions, and propagate ongoing data changes to all of them.

DynamoDB Streams

Many applications can benefit from the ability to capture changes to items stored in a DynamoDB table, at the point in time when such changes occur.

[A DynamoDB stream](#) is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A *stream record* contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the "before" and "after" images of modified items.

Using DynamoDB Streams with Amazon Elasticsearch

You can use AWS Lambda to [send data to your Amazon Elastic Search domain from Amazon DynamoDB](#). New data that arrives in the database table triggers an event notification to Lambda, which then runs your custom code to perform the indexing. See below for more information about Amazon Elasticsearch. For a user example, see this [blog post](#).

Other Topics

Amazon Elasticsearch

[Amazon Elasticsearch Service \(Amazon ES\)](#) is a managed service that makes it easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Elasticsearch is a popular open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analysis. With Amazon ES, you get direct access to the Elasticsearch APIs; existing code and applications work seamlessly with the service.

WHAT YOU ACCOMPLISHED THIS WEEK

- You continued to build out your application using a session table and enabling TTL
- You secured your application following the least privilege principle, using AWS [IAM](#), [VPC](#), and an

[VPC Endpoint for DynamoDB.](#)