Amazon DynamoDB

## Service Overview

[Amazon DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html) is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.

In DynamoDB, tables, items, and attributes are the [core components](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html) that you work with. A table is a collection of items, and each item is a collection of attributes.

DynamoDB supports [eventually consistent and strongly consistent reads](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html).

DynamoDB comes in two [Read/Write capacity modes](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html), which should be chosen depending on application load and your budget.

For working with data, you can access Amazon DynamoDB using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the DynamoDB API.

## Use cases / Considerations

Whenever SQL database (like RDS) is not suitable for your needs (e.g., data structure is not the same across all items), DynamoDB is a great choice for such cases.

If your application requires [really fast reads](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.html#DAX.use-cases) (real-time bidding, social gaming, and trading applications) [DynamoDB Accelerator (DAX)](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.html) is a good choice, as it will cache query data from DynamoDB and your application will get the needed data much faster.

As DynamoDB has two Read/Write capacity modes, it’s important to understand, whether to use [On-Demand](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html#HowItWorks.OnDemand) or [Provisioned](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html#HowItWorks.ProvisionedThroughput.Manual) mode.

## Governance

DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs. For more information, see [On-Demand Backup and Restore for DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html).

You can create on-demand backups and enable point-in-time recovery for your Amazon DynamoDB tables. Point-in-time recovery helps protect your tables from accidental write or delete operations. With point-in-time recovery, you can restore a table to any point in time during the last 35 days. For more information, see [Point-in-Time Recovery: How It Works](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/PointInTimeRecovery_Howitworks.html).

DynamoDB allows you to delete expired items from tables automatically to help you reduce storage usage and the cost of storing data that is no longer relevant. For more information, see [Expiring Items By Using DynamoDB Time to Live (TTL)](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html).

Dynamo is also integrated with CloudWatch, allowing you to enable [logging and monitoring](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/MonitoringDynamoDB.html).

## Cautions

Dynamo has a [number of quotas](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html) you need to pay attention to. One of the most important limits is [Read/Write throughput](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html#default-limits-throughput-capacity-modes). If you exceed this limit, all queries and requests will be throttled. Setting up CloudWatch alerts to track such cases would be really nice.

## Pricing considerations

All info regarding DynamoDB pricing can be found in [AWS docs](https://aws.amazon.com/dynamodb/pricing/)

## More details

<https://www.youtube.com/watch?v=MF9a1UNOAQo>