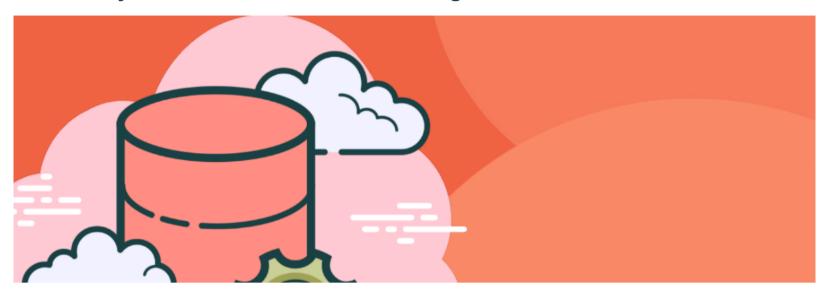**CHANDAN PATRA**   SEPTEMBER 15, 2022

BLOG   AMAZON WEB SERVICES

Amazon DynamoDB: What It Is and 10 Things You Should Know

What is Amazon DynamoDB?

Amazon DynamoDB is a managed NoSQL service with strong consistency and predictable performance that shields users from the complexities of manual setup.

NoSQL vs SQL Databases

Whether or not you've actually used a NoSQL data store yourself, it's probably a good idea to make sure you fully understand the key design differences between NoSQL (including Amazon DynamoDB) and the more traditional relational database (or "SQL") systems like MySQL.

First of all, **NoSQL** does not stand for "**No**t **SQL**", but "**N**ot **O**nly **SQL**". The two are not opposites, but complementary. NoSQL designs deliver faster data operations and can seem more intuitive, while not necessarily adhering to the ACID (atomicity, consistency, isolation, and durability) properties of a relational database.

There are many well-known NoSQL databases available, including MongoDB, Cassandra, HBase, Redis, Amazon DynamoDB, and Riak. Each of these was built for a specific range of uses and will therefore offer different features. We could group these databases into the following categories: *columnar* (Cassandra, HBase), *key-value store* (DynamoDB, Riak), *document-store* (MongoDB, CouchDB), and *graph* (Neo4j, OrientDB).

In this post, I'm going to focus on AWS DynamoDB — the giant of the NoSQL world. I believe it's become a giant because AWS built it for their own operations. Considering how much was at stake financially, anything less than complete reliability would simply not be tolerated. Software created in such a demanding environment and with the use of AWS-scale resources is bound to be epic. The result? Fantastic reliability and durability, with blazing fast service.

Like any other AWS product, Amazon DynamoDB was designed for failure (i.e., it has self-recovery and resilience built in). That makes DynamoDB a highly available, scalable, and distributed data store. Here are ten DynamoDB features that helped make this database service into a giant.

1. Amazon DynamoDB is a managed, NoSQL database service

With a managed service, users only interact with the running application itself. You don't need to worry about things like server health, storage, and network connectivity. With Amazon DynamoDB, AWS provisions and runs the infrastructure for you. Some of DynamoDB's critical managed infrastructure features include:

> Automatic data replication over three availability-zones in a single region.
> Infinitely scalable read-write I/O running on IOPS-optimized solid state drives.
> A provisioned-throughput model where read and write units can be adjusted at any time based on actual application usage.
> Data backed up to S3.
> Integrated with other AWS services like Elastic MapReduce (Amazon EMR), Data Pipeline, and Amazon Kinesis.
> Amazon DynamoDB pricing: Pay-per-use model – you never pay for hardware or services you're not actually using.
> Security and access control can be applied using AWS IAM service.
> Great enterprise-ready features such as a robust SLA, monitoring tools, and private VPN functionality.

2. Amazon DynamoDB has Predictable Performance

AWS claims that DynamoDB will deliver highly predictable performance. Considering Amazon's reputation for service delivery, we tend to take them at their word on this one. You can actually control the quality of the service you'll get by choosing between Strong Consistency (Read-after-Write) or Eventual Consistency. Similarly, if a user wants to increase or decrease the Read/Write throughput they'll experience, they can do it through simple API calls. Amazon DynamoDB also offers what they call **Provisioned Capacity**: briefly, DynamoDB allows you to "bank" up to five minutes of unused capacity, which, like the funds in an emergency bank account, you can use during short bursts of activity.

3. Amazon DynamoDB is designed for massive scalability

Being an AWS product, you can assume that Amazon DynamoDB will be extremely scalable. With their automatic partitioning model, as data volumes grow, DynamoDB spreads the data across partitions and raises throughput. This requires no intervention from the user.

4. Amazon DynamoDB data types

DynamoDB supports the following data types:

> **Scalar** – Number, String, Binary, Boolean, and Null.
> **Multi-valued** – String Set, Number Set, and Binary Set.
> **Document** – List and Map.

Scalar types are generally well understood. We'll focus instead on multi-valued and document types. Multi-valued types are sets, which means that the values in this data type are unique. For a *months* attribute you can choose a String Set with the names of all twelve months – each of which is, of course, unique.

Similarly, document types are meant for representing complex data structures in the form of Lists and Maps. See this example:

```
{
    Id = 100
    ProductName = "K3 Note"
```

```
    Color = [ "White", "Black" ]
    ProductCategory = "Mobile"
}
```

5. Amazon DynamoDB's Data Model

DynamoDB uses three basic data model units, Tables, Items, and Attributes. Tables are collections of Items, and Items are collections of Attributes.

Amazon DynamoDB attributes are basic units of information, like key-value pairs. Tables are like tables in relational databases, except that in DynamoDB, tables do not have fixed schemas associated with them. Items are like rows in an RDBMS table, except that DynamoDB requires a Primary Key. The Primary Key in DynamoDB must be unique so that it can find the exact item in the table. DynamoDB keys supports two kinds of Primary Keys:

Hash Type Primary Key: If an attribute uniquely identifies an item, it can be considered as Primary. DynamoDB builds a hash index on the attribute to facilitate the uniqueness. A Hash Key is mandatory in a DynamoDB table.

Hash and Range Type Primary Key: This type of Primary Key is built upon the hashed key and the range key in the table: a hashed index on the hash primary key attribute, and a range sort index on the range primary key attribute. This type of primary key allows for AWS's rich query capabilities.

6. Amazon DynamoDB indexes

There are two types of indexes in DynamoDB, a Local Secondary Index (LSI) and a Global Secondary Index (GSI). In an LSI, a range key is mandatory, while for a GSI you can have either a hash key or a hash+range key. GSIs span multiple partitions and are placed in separate tables. DynamoDB supports up to five GSIs. While creating a GSI, you need to carefully choose your hash key because that key will be used for partitioning.

Which is the right index type to use? Here are two considerations: LSIs limit item size to 10 GB, and GSIs offer only eventual consistency.

7. Amazon DynamoDB partitions

In DynamoDB, data is partitioned automatically by its hash key. That's why you will need to choose a hash key if you're implementing a GSI. The partitioning logic depends upon two things: table size and throughput.

The partition for a table is calculated by DynamoDB. Although it is transparent to users, you should understand the logic behind this.

(**Note**: Read Capacity Units – RCU – are measured in 4KB/sec. Write Capacity Units – WCU – are measured in 1KB/sec.)
According to this formula, if we have a table size of 16 GB and we have 6000 RCUs and 1000 WCUs, then:

# of partitions by throughput: 6000/3000+1000/1000 = 3

# of partitions by size: 16/10 = 1.6

So, the # of partitions in total: max(1.6, 3) = 3

Therefore, we will require three partitions. The RCUs and WCUs will be uniformly distributed across the partitions. Here, RCUs per partition will be 3000/3 = 1000. RCUs and the WCUs will be 1000/3 = 333 WCUs. The data per partition will be 16/3 = 5.4 GB per partitions.

8. Amazon DynamoDB streams

DynamoDB streams are like transactional logs for a table. According to the DynamoDB Developer's Guide:

> *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.*

Streams are applied only to tables, and each stream record appears exactly once in a stream. AWS maintains separate endpoints for DynamoDB and DynamoDB streams. There are all kinds of scenarios where streams can be useful, such as in a messaging application where a message or picture that is updated to a group must be reflected in the message boxes of all the group members, 0r for sending welcome messages to new customers when they sign up for your service.

9. Amazon DynamoDB integration with Amazon EMR and Redshift

NoSQL and Big Data technologies are often discussed together, because they both share the same distributed and horizontally scalable architecture, and both aim to provide high volume, structured, and semi-structured data processing. In a typical scenario, Elastic MapReduce (EMR) performs its complex analysis on datasets stored on DynamoDB. Users will often also use AWS Redshift for data warehousing, where BI tasks are carried out on data loaded from DynamoDB tables to Redshift.

10. Amazon DynamoDB JavaScript Web Shell

AWS has introduced a web-based user interface known as the DynamoDB JavaScript Shell for local development. You can download the tool here.

Steps:

Download and extract the appropriate file
Run following command:

```
java -Djava.library.path=./DynamoDBLocal_lib -jar DynamoDBLocal.jar
```

Access the console in a browser with the URL: **http://localhost:8000/shell**

The web page will look like this:

Click on the button to get some sample commands

For example the createTable API will run:

After running this, listTables will show you:

This is a great tool to perform syntax checking before actually going to production.

With DynamoDB, Amazon has done a great job providing a NoSQL service with strong consistency and predictable performance, while saving users from the complexities of a distributed system. One proof of their success is the many systems (like Riak) that chose to build on the DynamoDB design. With a strong ecosystem, Amazon DynamoDB is something to consider when you are building your next Internet-based scale application.

Ready to try it for yourself? Why not use Cloud Academy's AWS DynamoDB hands-on lab?

AWS   *DynamoDB*

WRITTEN BY   **CHANDAN PATRA**

Cloud Computing and Big Data professional with 10 years of experience in pre-sales, architecture, design, build and troubleshooting with best engineering practices. Specialities: Cloud Computing - AWS, DevOps(Chef), Hadoop Ecosystem, Storm & Kafka, ELK Stack, NoSQL, Java, Spring, Hibernate, Web Service

Subscribe to our blog

Get the latest posts in your email

OR SUBSCRIBE TO THE   RSS FEED

The cloud skills platform of choice for teams & innovators

Certification Learning Paths.
Hands-on Labs. Skill Validation.
Assess your team's tech skills today.

Tour the Product

CARLOS RIVAS	DECEMBER 1, 2023	ALANA LAYTON	DECEMBER 1, 2023	DANNY JESSEE	NOVEMBER 30, 2023	STUART SCOTT

AWS   *re:Invent*

New AWS re:Invent Announcements: Dr. Werner Vogels Keynote

Read more

AWS   *re:Invent*

New AWS re:Invent Announcements: AWS Partner Keynote with Dr Ruba Borno

Read more

AWS   *re:Invent*

New AWS re:Invent Announcements: Swami Sivasubramanian Keynote

Read more

AWS   *re:Invent*

AWS re:Invent announcemen Keynote

Read more

SUGGESTED POSTS

Cloud Academy's Black Friday Deal Is Here!

The Black Friday Early-Bird Deal Starts Now!

Cloud Academy Wins Again: Fall 2023 G2 Awards

Navigating the Challenges of Modernization with Cloud Academy

What is Cloud Migration? Strategy, Processes, Benefits & Risks

Google Cloud Certifications: Which is Right for You and Your Team?

The 12 AWS Certifications: Which is Right for You and Your Team?

Level Up Your Cloud Skills: Oracle Now in Cloud Academy's Training Library

Get 40% Off Individual Plans at Cloud Academy

Retaining Your Rockstars: How to Avoid Management Disruptions

5 Reasons Why Top Consulting Firms Fail at Multi-Cloud – And How to Fix It

Google Unveils Gemini AI

New AWS re:Invent Announcements: Dr. Werner Vogels Keynote

New AWS re:Invent Announcements: AWS Partner Keynote with Dr Ruba Borno

New AWS re:Invent Announcements: Swami Sivasubramanian Keynote

AWS re:Invent 2023: New announcements - Adam Selipsky Keynote

EXPLORE LIBRARY →

SOLUTIONS

Managing the Multi-Cloud
Migrate to the Cloud
Develop Tech Talent
Optimize Cloud Spend

CERTIFICATIONS

Cloud Certifications
AWS Certifications
Azure Certifications
Google Cloud Certifications

TRAINING LIBRARY

AWS Training
Azure Training
Google Cloud Training
Alibaba Cloud Training
Big Data Training
Business Management Training
AWS Certifications
Azure Certifications
GCP Certifications
Alibaba Cloud Certifications
Microsoft 365 Certifications
DevOps Certifications
Security Certifications
Cloud Computing Fundamentals
Cloud Migration Training
Containers Training
DevOps Training
Upcoming Training

RESOURCES

Cloud Webinars
Cloud Conferences
Cloud Case Studies
Whitepapers
Cloud Academy Blog
Press Releases
Courses
Release Notes

PAST EVENTS

COURSE INDEX

Amazon Web Services
Microsoft Azure
Google Cloud Platform
Oracle Cloud Infrastructure