



amazon  
DynamoDB

**Team 24** : Team\_JP

**Members** : Juhi Birju Patel (40190446)  
Pal Rakeshkumar Patel (40197897)

Dhruviben Jigneshkumar Modi (4389271294)  
Keta Chandreshbhai Thakkar (4389266784)

# Amazon DynamoDb

- It is completely managed by AWS.
- It is a NoSQL database.
- It provides API to implement operation like create, read, update and delete.
- It provides stable output performance with increase in data.



# Project Phases

**UPLOAD DATASET TO  
DYNAMODB**



**PERFORM OPERATIONS**

- Create - Read - Update - Delete - Query - Scan

**EXPLORE DISTRIBUTED  
CONCEPTS**

- Consistency and replication
- Coordination
- Security and compliance
- Resilience and Disaster Recovery

# Dataset - Used Vehicles

- The dataset contains information about used cars which are for sale in Craigslist.
- Craigslist is the world's largest Collection of used cars.
- This dataset includes every used vehicle entry within the United States on Craigslist.
- Source: <https://www.kaggle.com/austinreese/craigslist-carstrucks-data>
- Size: 1.3 GB



**DEMO**


# Consistency and Replication

## Consistency

- **Eventually Consistent Reads (DEFAULT)**
  - Maximize throughput
  - Eventual writes
  - Takes time to update
- **Strongly Consistent Reads**
  - Write operation is performed instantaneously
  - Up-to-Date data
  - Higher latency

## Replication

- Global Tables
- Multi-Region Replication

<b>Replicas (1)</b> Other AWS Regions to which you have replicated this table. You are using global tables version			
<div> <span>Delete replica</span> <span>Create replica</span></div>			
	Replication Region	Status	Provisioned read capacity
<input type="radio"/>	Europe (London)	✓ Active	1-10, auto scaling at 70%

# Coordination

- Transactional APIs provide ACID with no additional cost
- Covers longstanding needs for connected database operations
  - Extends scalability and performance to a broader set of workloads
- With transactions, you no longer have to worry about or struggle with rollback operations within the database.
- Transactions helps you maintain data integrity by coordinating actions across multiple items and tables.

# Security and Compliance

- Uses IAM to control authentication and authorization
- Provides AWS Lambda support to create triggers
  - Triggers respond automatically to stream events
  - Triggers can also respond to data modifications
- Fully managed database. AWS manages
  - OS
  - Firewalls
  - Database patches
  - Disaster recovery
- All user data is encrypted, and stores encryption keys in AWS KMS.



# Resilience and Disaster Recovery

## Resilience

### AWS Infrastructure is built around:

1. Availability Zones
2. AWS Regions

### AWS provides:

-Active-Active Cross-Region

## Disaster Recovery

### For Recovery:

- On-demand backups
- Point-in-time Recovery

#### Point-in-time recovery (PITR) [Info](#)

Point-in-time recovery provides continuous backups of your DynamoDB data for 35 days to help you protect against accidental write or deletes. Additional charges apply. See [Amazon DynamoDB pricing](#) [↗](#)

Edit

Restore

Status

✔ Enabled

Earliest restore point

December 18, 2021, 16:37:13  
(UTC-05:00)

Latest restore point

December 18, 2021, 16:37:13  
(UTC-05:00)

# Amazon DynamoDB

Juhi Birju Patel  
Concordia University  
Montreal, Quebec  
juhibirjupatel@gmail.com

Dhruviben Jigneshkumar Modi  
Concordia University  
Montreal, Quebec  
dhruvimodi2112@gmail.com

Pal Rakeshkumar Patel  
Concordia University  
Montreal, Quebec  
palpatel211@gmail.com

Keta Chandreshbhai Thakkar  
Concordia University  
Montreal, Quebec  
keta.arcane@gmail.com

## ABSTRACT

In an era where relational databases are mainly used for data storage and retrieval, modern web technologies posed a major challenge in the form of unstructured data, high scale data, enormous concurrency etc. NoSQL databases are highly flexible as they can store and combine any type of data, both structured and unstructured, unlike relational databases that can store data in a structured way only. Amazon DynamoDB is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications at any scale. It is completely managed by AWS. DynamoDB offers built-in security, continuous backups, automated multi-region replication, in-memory caching, and data export tools. It provides stable output performance with increase in data.



Figure 1: Amazon DynamoDB

## 1 INTRODUCTION

Amazon DynamoDB is a fully managed NoSQL database service that can scale on demand to support virtually unlimited concurrent read/write operations with response time in single-digit millisecond latency that can further be brought down to microseconds with caching services like DAX or DynamoDB Accelerator. With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

DynamoDB allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs. With point-in-time recovery, you can restore a table to any point in time during the last 35 days. It 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. Major companies including Disney+, Dropbox, Snapchat, and Zoom have employed DynamoDB to efficiently store and process the several terabytes of data they generate everyday.

## 2 CONCEPTS OF DISTRIBUTED SYSTEM IN DYNAMODB

### 2.1 Consistency

A database consistency model refers to the requirement that a successful write or update performed operation is reflected in a subsequent read operation of that same value. Amazon DynamoDB provides two consistency models to maintain consistent data throughout all the tables. They are Eventually Consistent Reads and Strongly Consistent reads.

- Eventually Consistent Reads : The eventual consistency option is used to maximize the read throughput. Consistency across all copies of data is usually reached within a second. Eventual read after a write option takes time to fetch the most recent writes. Repeating a read operation after a short time will fetch up-to-date data.
- Strongly Consistent Reads : A strongly consistent read returns a result that reflects all the writes that was received as a successful response prior to the read. However, it takes more resources to apply a strongly consistent read than a

eventually consistent read. Strongly consistent reads may have higher latency than eventually consistent reads.

The Eventually Consistent Read is set as default in Amazon DynamoDB. However, DynamoDB allows user to specify the desired consistency option for each read request within an application. Hence, while performing a read operation, user can specify whether they want the read to be eventually or strongly consistent.

## 2.2 Replication

With global tables, DynamoDB provides a fully managed solution to implement multi-region replication. Global tables allows user to specify the AWS Regions where they want the table to be deployed. DynamoDB performs all of the necessary tasks to create identical tables in these Regions and propagate ongoing data changes to all of them.

When a user creates a DynamoDB global table, it consists of multiple replica tables (one per AWS Region) that DynamoDB treats as a single unit. Every replica has the same table name and the same primary key schema. When an application writes data to a replica table in one Region, DynamoDB propagates the write to the other replica tables in the other AWS Regions automatically Figure 2.

## 2.3 Auto-Scaling

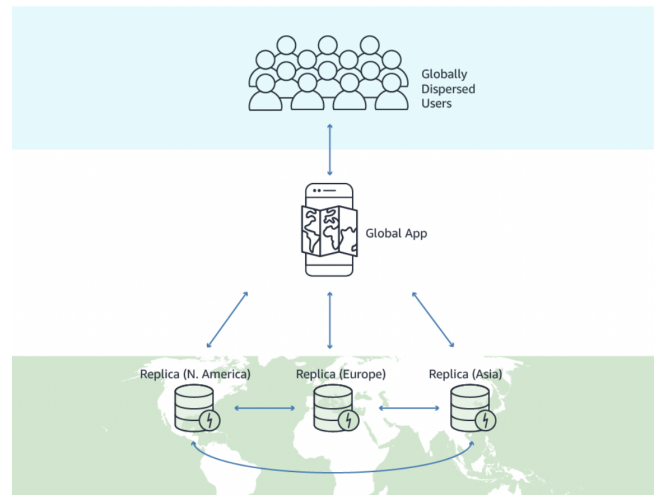
Sometimes, database workloads are cyclical in nature. That means, the database has more access requests during particular peak times and lesser requests during the rest of time. Consider a social media app that is used more by users during the daytime. This would mean that there should be more throughput during the day than at night. Taking another example of a game which is newly introduced in the market, there are chances of its installations shooting up rapidly within just few days of the launch. If the database is not ready to take in such numerous amounts of calls in a short time, the odds of the app crashing are high. Amazon DynamoDB supports Auto Scaling to adjust with the throughput requirements dynamically. This way, the database resources are neither wasted nor do they become insufficient when the request intensity increases, in response to the actual traffic patterns.

## 2.4 Coordination

DynamoDB transactional APIs, called TransactWriteItems and TransactGetItems simplify the developer experience by providing ACID for any item in any DynamoDB table with no additional cost. Transactions are enabled for all single-region DynamoDB tables by default and can be enabled on global tables optionally. It covers a longstanding need for connected database operations and extends the scale, performance, and enterprise-ready benefits of DynamoDB to a broader set of workloads. With transactions, you no longer have to worry about or struggle with rollback operations within the database. Transactions helps you maintain data integrity by coordinating actions across multiple items and tables.

## 2.5 Security and Compliance

AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Amazon DynamoDB provides a highly



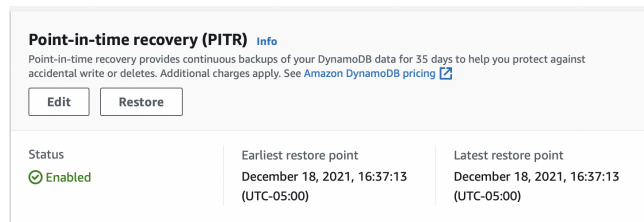
**Figure 2: An illustration of how DynamoDB tables are replicated among regions.**

durable storage infrastructure designed for mission-critical and primary data storage. Data is redundantly stored on multiple devices across multiple facilities in an Amazon DynamoDB Region. Identity and Access Management administrators control who can be authenticated (signed in) and authorized (have permissions) to use Amazon DynamoDB resources. The AWS Compliance Program helps customers to understand the robust controls in place at AWS to maintain security and compliance in the cloud.

DynamoDB is integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables. AWS handles basic security tasks like guest operating system (OS) and database patching, firewall configuration, and disaster recovery. DynamoDB encrypts at rest all user data stored in tables, indexes, streams, and backups using encryption keys stored in AWS Key Management Service (AWS KMS). This provides an additional layer of data protection by securing your data from unauthorized access to the underlying storage .

## 2.6 Resilience and Disaster Recovery

The AWS infrastructure is built around Availability Zones and AWS regions. This infrastructure is extremely secure, reliable, and extensive. AWS provides active-active cross-region support, that means, independent nodes in the networked server cluster have access to their own replicated database server. This capability increases the resilience of the database preventing a single database failure that can render an application inoperable. With DynamoDB, one can have full backups of tables for long term retrieval and archival. Using DynamoDB with AWS Backup, you can copy your on-demand backups across AWS accounts and regions. All on-demand backups are cataloged, discoverable, and retained until they are explicitly deleted. With point-in-time Recovery, you do not have to worry about creating and maintaining on-demand backups as it protects the DynamoDB tables from accidental write and delete operations.



**Figure 3: Point-in-time Recovery(PITR) in DynamoDB**

Let us suppose that a test script writes accidentally to a production DynamoDB table. With point-in-time recovery, we can restore that table to any point in time during the last 35 days as DynamoDB maintains incremental backups of your table. As shown in Figure 3, we have enabled Point-in-time Recovery for the table that is being used in the project.

### 3 PROJECT IMPLEMENTATION ASPECTS

#### 3.1 Used Car Analysis

**3.1.1 Introduction.** This is a Amazon DynamoDB based Used Car analysis project. It reads the file as an input for the datasource and retrieves data by querying the dataset.

**3.1.2 Dataset.** The dataset contains information about used cars which are for sale in Craigslist. This dataset includes every used vehicle entry within the United States on Craigslist. The size of the dataset used is 1.3GB.

### 4 TECHNIQUES IN THE IMPLEMENTATION OF DYNAMODB PROJECT

#### 4.1 DynamoDB Batch Write

The Amazon DynamoDB BatchWriteItem operation puts or deletes multiple items in one or more tables. A single call to BatchWriteItem can write up to 16 MB of data, which can comprise as many as 25 put or delete requests.

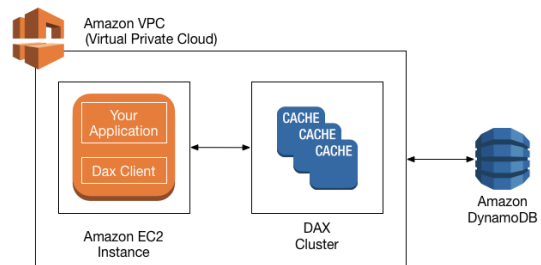
Amazon DynamoDB does not have a CSV write directly. We had a huge database of 1.3GB. So we first uploaded the dataset in S3 bucket. Then we performed BatchWriteItem operation with a batch size of 25 units on the dataset using lambda function and we were able to write the dataset in DynamoDB table.

#### 4.2 GetItem, Query and Scan Command

There are three operations available to query DynamoDB. They are GetItem, Query and Scan. The order of efficiency is GetItem> Query> Scan.

GetItem gets an item from the hash using the primary key. It is the fastest operation but it applies on a single element using the primary key. Here we pass id and retrieve all the information pertaining to that car like price, manufacturer, model, condition etc.

Query Returns the metaData of the table. It returns information about the table like the current status of the table, when it was created, primary key schema etc. We use this command to get information about the table.



**Figure 4: DAX: How it works.**

Scan is the most slowest one. It scans the entire dataset to fetch the results. A scan command in Amazon DynamoDB reads every item in the table. We have used this command to fetch cars which a price lower than a specified value, filtering cars on the basis on its condition, cars from a specific region.

#### 4.3 DAX

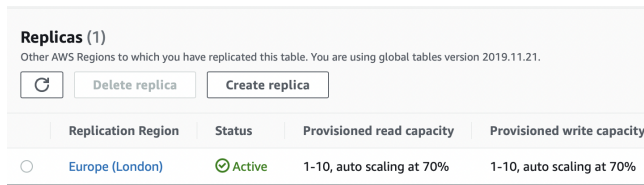
Amazon DynamoDB Accelerator (DAX) is an in-memory cache for Amazon DynamoDB. It delivers up to a 10 times performance improvement—from milliseconds to microseconds—even at millions of requests per second. It is fully managed by Amazon and is highly available. With DAX, an user does not have to be concerned with the caching, cluster management, data population anymore. DAX is designed to run within an Amazon Virtual Private Cloud (Amazon VPC) environment which defines a virtual network that closely resembles a traditional data center. One can launch a DAX cluster in a virtual network and can also control access to the cluster by using Amazon VPC security groups.

As shown in Figure 4, the DAX client would direct all your requests to the DAX cluster. If DAX is able to performs he requests, it does so. Otherwise it send to DynamoDB for the processing and response.

#### 4.4 Multi-region Replication

Amazon DynamoDB provides its users with a completely managed solution for the deployment of a multi-region database. This means, end users do not need to have their own replication solution for the data they are concerned with. With the provision of global tables, one can specify the AWS region where the table is supposed to be replicated. DynamoDB takes care of all the tasks to be done for creating identical tables in the mentioned regions.

It also makes sure that all the ongoing data changes are reflected across all the regions where the table is replicated. Also, according to the frequency of access to these tables in different regions, each replica can use a different table class to optimise performance as well as cost. For example, an infrequently-accessed table in a particular replica can use the DynamoDB Standard-Infrequent Access table class, otherwise, the Standard Table class can be used for the case of frequent accesses. As shown in Figure 5, we have replicated our table in the AWS Region Europe(London).



Replication Region	Status	Provisioned read capacity	Provisioned write capacity
Europe (London)	Active	1-10, auto scaling at 70%	1-10, auto scaling at 70%

**Figure 5: Multi-Region Replication in DynamoDB**

## 4.5 Backup and Recovery

The on-demand backup service of DynamoDB allows you to create full backups of your data, ranging from few megabytes to hundreds of terabytes with no impact on performance. These backups are capable of processing in seconds, regardless of the size of the tables in it. As mentioned in Section 2.5, Point-in-time Recovery (PITR) continuously backs up the tables. It stores the backup for the last 35 days when enabled, until it is disabled explicitly. One can perform backup and restore operations on DynamoDB tables with a single API call or with just a single click on the AWS console, without affecting the table performance or availability. DynamoDB also has a timestamp

## 5 CONCLUSION

Amazon DynamoDB provides consistency, multi-region replication replication through its global tables, coordination with its support for ACID properties for any item without any additional cost incurs, security with guest Operating Systems and database patching, firewall configurations, encryption for data, on-demand backups for any size range of tables, point-in-time recovery in case of accidental update or delete operations, in-memory caching, eliminating the need for end users or organisations to engineer a cloud infrastructure for these capabilities.

## REFERENCES

- [1] Amazon DynamoDB- Overview.  
<https://docs.aws.amazon.com/dynamodb/index.html>, Accessed December 17, 2021.
- [2] Amazon DynamoDB- Consistency.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/How-ItWorks.ReadConsistency.html>, Accessed December 17, 2021.
- [3] Amazon DynamoDB- Replication.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html>, Accessed December 17, 2021.
- [4] Amazon DynamoDB- Auto-Scaling.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html>, Accessed December 17, 2021.
- [5] Amazon DynamoDB- Coordination.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/transaction-apis.html>, Accessed December 17, 2021.
- [6] Amazon DynamoDB- Security and Compliance.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/security.html>, Accessed December 18, 2021.
- [7] Amazon DynamoDB- Resilience and Disaster Recovery.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/disaster-recovery-resiliency.html>, Accessed December 17, 2021.
- [8] Amazon DynamoDB- DynamoDB Batch Write.  
<https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/dynamodb-example-table-read-write-batch.html>, Accessed December 18, 2021.
- [9] Amazon DynamoDB- GetItem, Query and Scan Command.  
<https://dynamodb.dev/dynamodb-scan-vs-query/>, Accessed December 17, 2021.
- [10] Amazon DynamoDB- DAX.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.concepts.html>, <https://aws.amazon.com/dynamodb/dax/>, Accessed

December 18, 2021.

- [11] Amazon DynamoDB- Multi-region Replication.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html>, Accessed December 18, 2021.
- [12] Amazon DynamoDB- Backup and Recovery.  
<https://aws.amazon.com/dynamodb/backup-restore/>, Accessed December 19, 2021.