DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings

DAX
Clusters
Subnet groups
Parameter groups
Events

Database

Amazon DynamoDB

A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

How it works

Get started

Create a new table to start exploring DynamoDB.

Create table

Pricing

DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to turn on. DynamoDB has on-demand capacity mode and provisioned capacity mode, and these modes have pricing for processing reads and writes on your tables.

Learn more about pricing

Documentation

- What is Amazon DynamoDB?
- Getting started with Amazon DynamoDB
- Setting up DynamoDB
- Accessing DynamoDB
- Programming with DynamoDB

AWS DynamoDB Service # Yash Garg

DynamoDB stores data as items in tables, with each items represented as a json-like document consisting of key-value pairs.

→ No-SQL, fully managed, serverless database service by AWS.

→ Built for high availability, performance, & automatic scaling

→ Supports both key-value and document data models.

→ Ideal for applications that need low-latency data access.

Core Components

1. Tables

- Logical container for data
- Must have a primary key (partition key or partition + sort key).

2. Items

- Each row in a table is an item.
- Can have different attributes (schema-less).

3. Attributes

- A key-value pair with in an item.
- Supports:-

Scalar types:- String, number, binary, boolean, Null

Document types:- List, Map

Set types:- String Set, Number Set, Binary Set.

DynamoDB free Tier Offers & other

- 25 provisioned write capacity Units (WCUs) per month
- 25 provisioned Read capacity Units (RCUs) per month
- 25 GB of data storage
- 2.5 million read requests from DynamoDB Streams
- 1GB of outbound data transfer
- No time limits, available if you stay within the free tier.

AWS Educate (Student Credits)

- offers $25-100 in AWS credits usable for DynaDB & other AWS services.
- subject to eligible approved by AWS Education team.

AWS Activate (Startups)

- Provides $1000, - $100,000 AWS credits depending on program tier.
- Can be applied toward DynamoDB usage & other AWS resources.

DynamoDB Local

- Installable version of DynamoDB for local development & testing.
- Completely free with no AWS billing charge.

Summary Cost Table

Items	Free tier	Standard pricing
Write Capacity Units (WCU)	25 WCUs/month	$0.625/million WRUs (on-demand)
Read Capacity Units (RCU)	25 RCUs/month	$0.125/million RRUs (on-demand)
Provisioned WCU cost	Included in 25 RWCU	$0.00065 per WCU-hour
Provisioned RCU cost	Included in 25 WCU	$0.00013 per RCU-hour

	Yash	(73)
Storage	25 GB/month	$0.25/GB-month
(Stream Read Request Recunits) Stream SRRUs	2.5M free SRRUs /month	$0.02 per 100K SRRUs
On-demand Backup	–	$0.10/GB-month
(Point in time Recovery) PITR Backup	–	$0.20/GB-month
Restore Table	–	$0.15/GB
Global Table replication	–	$1.875/million replicated writes
change data capture	–	$0.10/million units
Data tranfer out.	1GB /month free	~ $0.09 per GB

Items

1. Write Capacity Units (WCU)
   - Amount of writes you can perform per second (used in provisioned mode).

2. Read Capacity Unit (RCU)
   - Amount of reads you can perform per second (used in provisioned mode).

3. Provisioned WCU & RCU
   - Reserved Read & Write throughput (pay pay per hour).

4. Storage
   - charge for data stores in your table. (GB)

5. SRRU (Streams) :-
   - Cost of reading table change record from DynamoDB streams.

6. On-Demand Backup :-
   - Cost to create manual backups of your table

7. Point-In-Time-Recovery :- Continuous backups features for recovering table to any second

8. Restore Table :- Cost to restore a table from a backups.

9. Global Table Replication :- Cost to replicate table data across multiple AWS region.

10. Change Data Capture :- Cost for tracking changes in your table.

11. Data Transfer Out :- Cost for transfering data out of AWS. (eg. to the Internet).

Steps :-

Step-1 :- DynamoDB

- Create a table 'contacts' with primary key as 'id' (Type String) as given in practical.

Step2 :- Create Access the policy for the DynamodB in the IAM Service.

Step-3 :- EC2 instance → Open terminal
sudo yum update -y
sudo yum install -y nodejs
npm install aws-sdk
nano app.js → given with practical
node app.js → Success & 3 connected.

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

dynamodb

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

id | String

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name | String

1 to 255 characters and case sensitive.

Table settings

- (•) **Default settings**
  The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.
- ( ) **Customize settings**
  Use these advanced features to make DynamoDB work better for your needs.

Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value	Editable after creation
Table class	DynamoDB Standard	Yes

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices Take a walkthrough Do not show me this message again.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

dynamoinstance

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible

ami-08a6efd148b1f7504 (64-bit (x86), uefi-preferred) / ami-0aaf509a1ebd95e61 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications

Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-08a6efd148b1f7504

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

Preview code

GNU nano 8.3 app.js Modified

```
const AWS = require('aws-sdk');
AWS.config.update({ region: 'us-east-1' }); // Change to your region

const dynamodb = new AWS.DynamoDB.DocumentClient();

const params = {
  TableName: 'Contacts',
  Item: {
    id: '1',
    name: 'Yash'
  }
};

dynamodb.put(params, function(err, data) {
  if (err) console.log('Error', err);
  else console.log('Success', data);
});
```

^G Help	^O Write Out	^F Where Is	^K Cut	^T Execute	^C Location	M-U Undo	M-A Set Mark	M-] To Bracket	M-B Previous	◂ Back	^◂ Prev Word	^A Home
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify	^/ Go To Line	M-E Redo	M-6 Copy	^B Where Was	M-F Next	▸ Forward	^▸ Next Word	^E End

i-04acaf68b4d6352b3 (dynamoinstance)

PublicIPs: 98.84.153.253 PrivateIPs: 172.31.26.223

```
[ec2-user@ip-172-31-26-223 ~]$ nano app.js
[ec2-user@ip-172-31-26-223 ~]$ node app.js
(node:26775) NOTE: The AWS SDK for JavaScript (v2) is in maintenance mode.
 SDK releases are limited to address critical bug fixes and security issues only.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the blog post at https://a.co/cUPnyil
(Use `node --trace-warnings ...` to show where the warning was created)
Success {}
[ec2-user@ip-172-31-26-223 ~]$
```

DynamoDB

Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings

DAX
Clusters
Subnet groups
Parameter groups
Events

Tables (1)

Any tag key

Any tag value

Find tables

1

dyanmodb

dyanmodb

Autopreview | View table details

Scan or query items

Scan | Query

Select a table or index: Table - dyanmodb

Select attribute projection: All attributes

Filters - *optional*

Run | Reset

Completed · Items returned: **1** · Items scanned: **1** · Efficiency: **100%** · RCUs consumed: **2**

Table: dyanmodb - Items returned (1)

Actions | Create item

Scan started on July 26, 2025, 14:42:01

1

id *(String)*	name
1	Yash

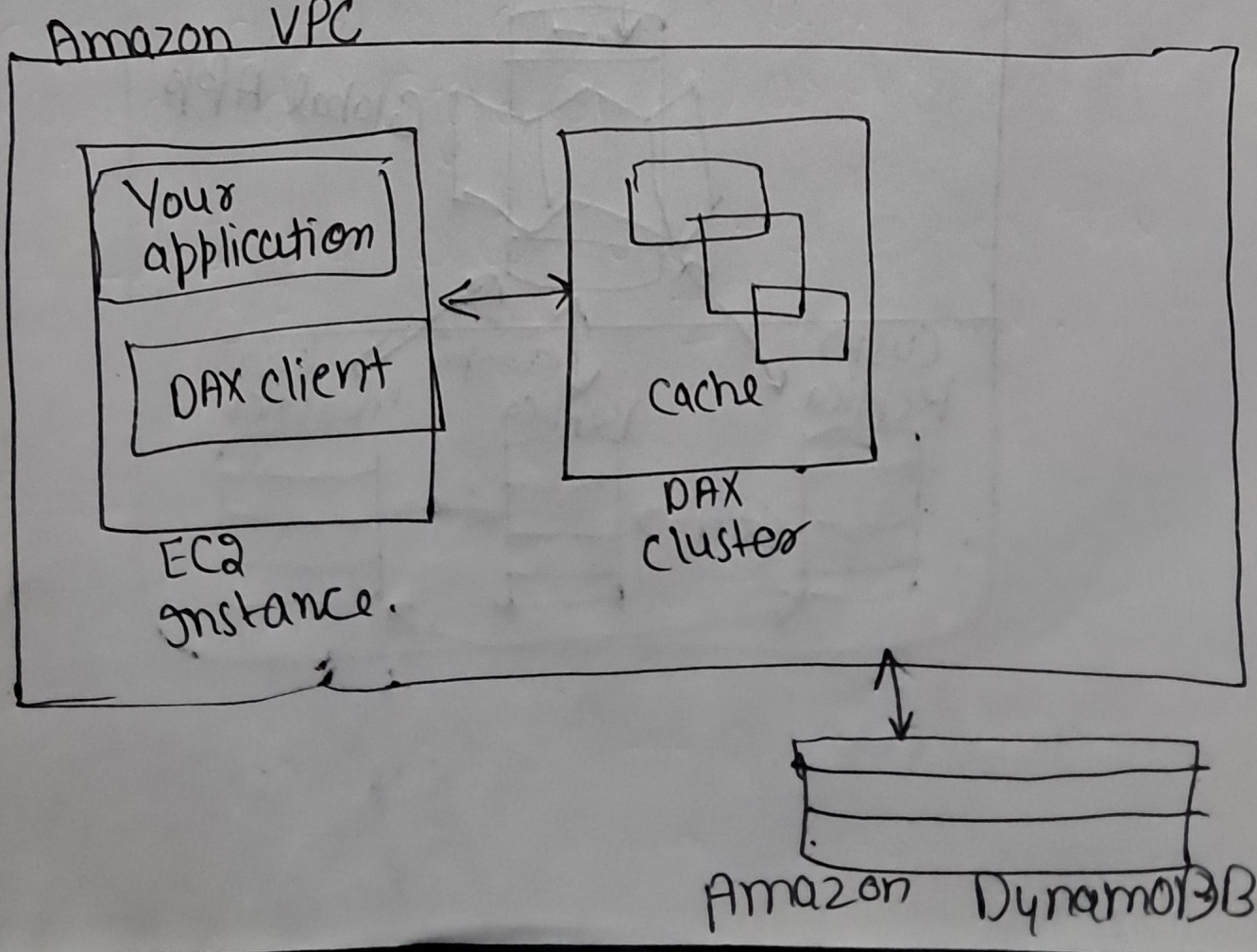
Features - points

- Serverless :- No need for server provisioning, software installation, maintenance, or patching
- Auto scaling :- Instantly scales up or down based on demand.
- Zero downtime :- Provide continuous availability without maintainence window.
- On demand pricing :- Pay only for the read/write request used, ideal for fluctuating workloads.
- Idle Cost Saving :- Scales down to zero during inactivity, so there's no cost when table have no traffic

Note:- It is a flexible data model & make reliable performance make dynamodb a great bit for mobile, web, gaming advertising technology, Internet of things, and other applications

DAX :→ DynamoDB Accelerator

- fully managed in-memory cache for DynamoDB
- DAX offers microsecond latency achieving up to 10X performance over standard dynamoDB queries.
- High Availability & Scalability: can be deployed in multiple AZs.
- DAX is only used for and is integrated with DynamoDB, while Elastic Cache can be used for other database.



Dynamo DB
Globabal table

1. Dynamo DB → dashboard → tables → table → click on it → Global table on 3-4 row → make replica in which region we want.

→ mostly used in industry.