

DynamoDB Table

The screenshot shows the AWS DynamoDB console interface. On the left is a navigation menu with options like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The main area is titled 'DynamoDB' and 'Explore items' for the 'UserActivity' table. It includes filters for 'Any tag key' and 'Any tag value', and a search bar 'Find tables'. The 'Scan or query items' section shows 'Scan' selected, with 'Table - UserActivity' chosen for the table and 'All attributes' for the attribute projection. A status bar indicates 'Completed' with 6 items returned, 6 items scanned, 100% efficiency, and 2 RCUs consumed. Below this, a table titled 'Table: UserActivity - Items returned (6)' shows the scan results.

	UserID (String)	ActivityTime (String)	ActivityType
<input type="checkbox"/>	2	20	football
<input type="checkbox"/>	1	30	cricket
<input type="checkbox"/>	6	30	snooker

Lambda Functions to use DynamoDB

The screenshot shows the AWS Lambda console for the 'queryUserActivity' function. The 'Code source' tab is active, displaying the JavaScript code for the function. The code imports the 'DynamoDBClient' and 'QueryCommand' from the '@aws-sdk/client-dynamodb' package, initializes a client and document client, and defines an async handler that sends a query command to the 'UserActivity' table. The handler filters by 'UserID' and sorts by 'ActivityTime' in descending order. The function is deployed and tested successfully, as shown in the 'TEST EVENTS' section.

```
1 import { DynamoDBClient } from "@aws-sdk/client-dynamodb"
2 import { QueryCommand, DynamoDBDocumentClient } from "@aws-sdk/lib-dynamodb"
3
4 const client = new DynamoDBClient({});
5 const docClient = DynamoDBDocumentClient.from(client);
6
7 export const handler = async (event) => {
8   const command = new QueryCommand({
9     TableName: "UserActivity",
10    KeyConditionExpression: "UserID = :userId",
11    ExpressionAttributeValues: {
12      ":userId": event.userId,
13    },
14    // Optional: Sort by time in descending order
15    ScanIndexForward: false,
16  });
17
18   try {
19     const response = await docClient.send(command);
20     console.log("Query result:", response.Items);
21   } catch (err) {
22     console.error("Error:", err);
23   }
24 }
```

TEST EVENTS [SELECTED: QUERYUSERACTIVITY...]

- Create new test event
- Private saved events
- QueryUserActivityByUserId

Execution Results

```
{
  "UserID": "2",
  "ActivityType": "football",
  "ActivityTime": "20"
}
```

aws [Search] [Option+S] United States (N. Virginia) SaurabGhimire

Lambda > Functions > insertIntoUserActivityTable

Code source info Upload from

EXPLORER

- INSERTINTOUSERACTIVITYTABLE
 - JS index.mjs

DEPLOY

Deploy (🔗U)

Test (🔗U)

TEST EVENTS [SELECTED: INSERTINTOUSERAC...

- Create new test event
- Private saved events
 - InsertIntoUserActivity

ENVIRONMENT VARIABLES

JS index.mjs X

```

1  import { DynamoDBClient } from "@aws-sdk/client-dynamodb"
2  import { PutCommand, DynamoDBDocumentClient } from "@aws-
3
4  const client = new DynamoDBClient({});
5  const docClient = DynamoDBDocumentClient.from(client);
6
7  export const handler = async (event) => {
8    const command = new PutCommand({
9      TableName: "UserActivity",
10     Item: {
11       UserID: event.userId,
12       ActivityType: event.activityType,
13       ActivityTime: event.activityTime, // ISO timestamp
14     },
15   });
16
17   try {
18     const response = await docClient.send(command);
19     console.log("Item added:", response);
20     return response;
21   } catch (error) {
22     console.error("Error adding item:", error);
23   }
24 }

```

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded

Test Event Name: InsertIntoUserActivity

Response:

```

{
  "Metadata": {
    "statusCode": 200,
    "requestId": "AD9CEPKI4TUJ8LUK24JAUUK97FV4KQNS05AEMVJF66Q9ASUAAJG",
    "attempts": 1,
    "totalRetryDelay": 0
  }
}

```

Amazon DocumentDB

aws [Search] [Option+S] United States (N. Virginia) SaurabGhimire

DocumentDB > Clusters > docdb-2025-04-12-16-29-36

Amazon DocumentDB <

Dashboard

- Clusters
- Performance Insights
- Snapshots
- zero-ETL integrations

Subnet groups

Parameter groups

Event Subscriptions

- Events
- Recommendations

No-code machine learning

What's New [?]

Tutorials

docdb-2025-04-12-16-29-36 Add instance Modify

▼ Summary

Engine version docdb 5.0.0	Instance status 3 / 3 instances are available	Memory optimized instances 3 instances	Cluster parameter group status in-sync
Total cluster instances 3/15	Cluster status Available	NVMe-backed instances 0 instances	Pending maintenance -

Connectivity & security Instances Configuration Monitoring Events & tags Maintenance & backups zero-ETL integrations

▼ Cluster details

Configurations and status ARN arn:aws:rds:us-east-1:103619411293:cluster:docdb-2025-04-12-16-29-36 Cluster identifier docdb-2025-04-12-16-29-36 (available) Cluster creation time 4/12/2025, 11:30:18 AM UTC-5 Cluster endpoint docdb-2025-04-12-16-29-36.cluster-cq402kqcat01.us-east-1.docdb.amazonaws.com	Backup Automated backups Enabled (1 day) Earliest restorable time 4/12/2025, 11:31:00 AM UTC-5 Latest restore time 4/12/2025, 11:59:26 AM UTC-5 Backup window 00:00-00:30 UTC (GMT)
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

aws [Search] [Option+S] United States (N. Virginia) SaurabGhimire

EC2 > Instances > i-0c5b7a3fc6e242017

EC2

- Dashboard
- EC2 Global View
- Events
- Instances**
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- Images**
 - AMIs
 - AMI Catalog
- Elastic Block Store**
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security**
 - Security Groups
 - Elastic IPs

Instance summary for i-0c5b7a3fc6e242017 (ec2) Info

Updated less than a minute ago

Instance ID i-0c5b7a3fc6e242017	Public IPv4 address 52.207.226.127 open address	Private IPv4 addresses 172.31.46.0
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-207-226-127.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-46-0.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-46-0.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.207.226.127 [Public IP]	VPC ID vpc-0d7818433017c2c0d	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-032a5248ece7d6acb	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:103619411293:instance/i-0c5b7a3fc6e242017	
Operator -		

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

▼ Instance details Info

AMI ID	Monitoring	Platform details
--------	------------	------------------

```
nodejs-full-i18n-1:18.20.6-1.amzn2023.0.2.x86_64
nodejs-libs-1:18.20.6-1.amzn2023.0.2.x86_64
nodejs-npm-1:10.8.2-1.18.20.6.1.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-46-0 ~]$ git clone https://github.com/Thao-V/aws-basic-apps

Cloning into 'aws-basic-apps'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 15 (delta 2), reused 11 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 86.10 KiB | 17.22 MiB/s, done.
Resolving deltas: 100% (2/2), done.
[ec2-user@ip-172-31-46-0 ~]$ curl -O https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 157k  100 157k    0     0  6028k      0 --:--:-- --:--:-- --:--:-- 6280k
[ec2-user@ip-172-31-46-0 ~]$ cp global-bundle.pem aws-basic-apps/node-docdb
[ec2-user@ip-172-31-46-0 ~]$ cd aws-basic-apps/node-docdb/
[ec2-user@ip-172-31-46-0 node-docdb]$ npm i

added 13 packages, and audited 14 packages in 2s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.3.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.3.0
npm notice To update run: npm install -g npm@11.3.0
npm notice
[ec2-user@ip-172-31-46-0 node-docdb]$ nano .env
[ec2-user@ip-172-31-46-0 node-docdb]$ nano .env
[ec2-user@ip-172-31-46-0 node-docdb]$ node app.js
(node:27396) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js
Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:27396) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js
Driver version 4.0.0 and will be removed in the next major version
DB connected
1 {
  acknowledged: true,
  insertedId: new ObjectId('67fa9b99718e1142aa27bb5f')
}
2 { _id: new ObjectId('67fa9b99718e1142aa27bb5f'), name: 'Thao' }

```

ElastiCache Redis

aws

Search

[Option+S]

United States (N. Virginia)

SaurabGhimire

ElastiCache

Valkey caches

redis-cluster

Amazon ElastiCache

<

Dashboard

▼ Resources

Valkey caches [New](#)

Memcached caches

Redis OSS caches

Global datastores

Reserved nodes

Backups

▼ Configurations

Subnet groups

Parameter groups

User management

User group management

Events

Service updates

ElastiCache cluster client

Documentation

Amazon MemoryDB

redis-cluster [Info](#)

Modify

Actions

▼ Cluster details

Cluster name redis-cluster	Description -	Node type cache.t2.micro	Status Available
Engine Valkey	Engine version 8.0.1	Global datastore -	Global datastore role -
Update status Up to date	Cluster mode Disabled	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Disabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Enabled	Parameter group default.valkey8	Outpost ARN -	Configuration endpoint -
Primary endpoint redis-cluster.8m2bhb.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint redis-cluster-ro.8m2bhb.ng.0001.use1.cache.amazonaws.com:6379	ARN arn:aws:elasticache:us-east-1:103619411293:replicationgroup:redis-cluster	Data migration No active migrations

Connectivity and security - newNodesMetricsLogsMaintenance and backupsService updatesTags

► Connect to your cache

Connect to your cache using AWS CloudShell, AWS CLI, GLIDE, or other client libraries.

Connect to cache

aws

Search

[Option+S]

United States (N. Virginia)

SaurabGhimire

EC2

Instances

i-0a7eab3301ad05fb3

EC2

<

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Instance summary for i-0a7eab3301ad05fb3 (ec2-for-redis) [Info](#)

Updated less than a minute ago

Connect

Instance state

Actions

Instance ID i-0a7eab3301ad05fb3	Public IPv4 address 54.197.33.120 open address	Private IPv4 addresses 172.31.30.43
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-197-33-120.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-30-43.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-30-43.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.197.33.120 [Public IP]	VPC ID vpc-0d7818433017c2c0d	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-067b942ce81fca362	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:103619411293:instance/i-0a7eab3301ad05fb3	
Operator -		

DetailsStatus and alarmsMonitoringSecurityNetworkingStorageTags

▼ Instance details [Info](#)

AMI ID	Monitoring	Platform details
--------	------------	------------------

```
Address: 172.31.0.2#53
Non-authoritative answer:
redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com canonical name = redis-cluster-002.8m2bbhb.0001.use1.cache.a
amazonaws.com.
Name: redis-cluster-002.8m2bbhb.0001.use1.cache.amazonaws.com
Address: 172.31.80.118

[ec2-user@ip-172-31-30-43 redis-stable]$ redis-cli -h redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com -p 6379
^C[ec2-user@ip-172-31-30-43 redis-stable]$ ^C
[ec2-user@ip-172-31-30-43 redis-stable]$ ^C
[ec2-user@ip-172-31-30-43 redis-stable]$ exit
logout
Connection to 54.197.33.120 closed.
➔ Downloads ssh -i key.pem ec2-user@54.197.33.120

      #_
    _\_\_#### Amazon Linux 2023
   ~~~\_#####
        \###|
        \|/
       ~~~~~V~'--->
           /
          /
         /
        /
       /m/'

Last login: Sat Apr 27 16:02:49 2025 from 205.221.160.10
[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com -p 6379
^C[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com -p 6379
redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> set count 4
(error) READONLY You can't write against a read only replica.
redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379>
[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com -p 6379
redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> set count 4
(error) READONLY You can't write against a read only replica.
redis-cluster-ro.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> exit
[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379 -p 6379
Could not connect to Redis at redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379: Name or service not known
not connected>
[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379
Could not connect to Redis at redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379: Name or service not known
not connected> exit
[ec2-user@ip-172-31-30-43 ~]$ redis-cli -h redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com -p 6379
redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> set count 4
OK
redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> get count
"4"
redis-cluster.8m2bbhb.ng.0001.use1.cache.amazonaws.com:6379> exit
[ec2-user@ip-172-31-30-43 ~]$
```