



Load and Query DynamoDB Tables

Gloria

Items returned (6)

	ID (Number)	Authors	Content Type	Difficulty	Price	Project Category	Publisher
<input type="checkbox"/>	3	[{"S": "Ne..."}]	Project	Easy peasy	0	AI/ML	true
<input type="checkbox"/>	2	[{"S": "Ne..."}]	Project	Easy peasy	0	Analytics	true
<input type="checkbox"/>	203		Video		0		
<input type="checkbox"/>	202		Video		0		
<input type="checkbox"/>	201		Video		0		
<input type="checkbox"/>	1	[{"S": "Nat..."}]	Project	Easy peasy	0	Storage	true

< 1 > | ⚙️ 🔍



Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service that provides fast, scalable, and flexible data storage. It's useful because it handles large amounts of data with high performance and requires no server management.

How I used Amazon DynamoDB in this project

In today's project, I used Amazon DynamoDB to create tables, load data, and manage items. I set up flexible, scalable tables like ContentCatalog, Forum, Post, and Comment, then added and updated data to organize and retrieve it efficiently.

One thing I didn't expect in this project was...

One thing I didn't expect in this project was how DynamoDB allows each item to have unique attributes, unlike RDS where data follows fixed rows and columns. This flexibility made DynamoDB feel more dynamic and adaptable and easy to use.

This project took me...

This project took me about 45 minutes to complete. It was a great hands-on experience learning how to create and manage DynamoDB tables, load data, and understand the flexibility it offers compared to relational databases.

Create a DynamoDB table

DynamoDB tables organize data using items and attributes. Items are like rows, each with a unique key, and attributes are like columns. Each item can have different attributes, making DynamoDB flexible compared to relational databases.

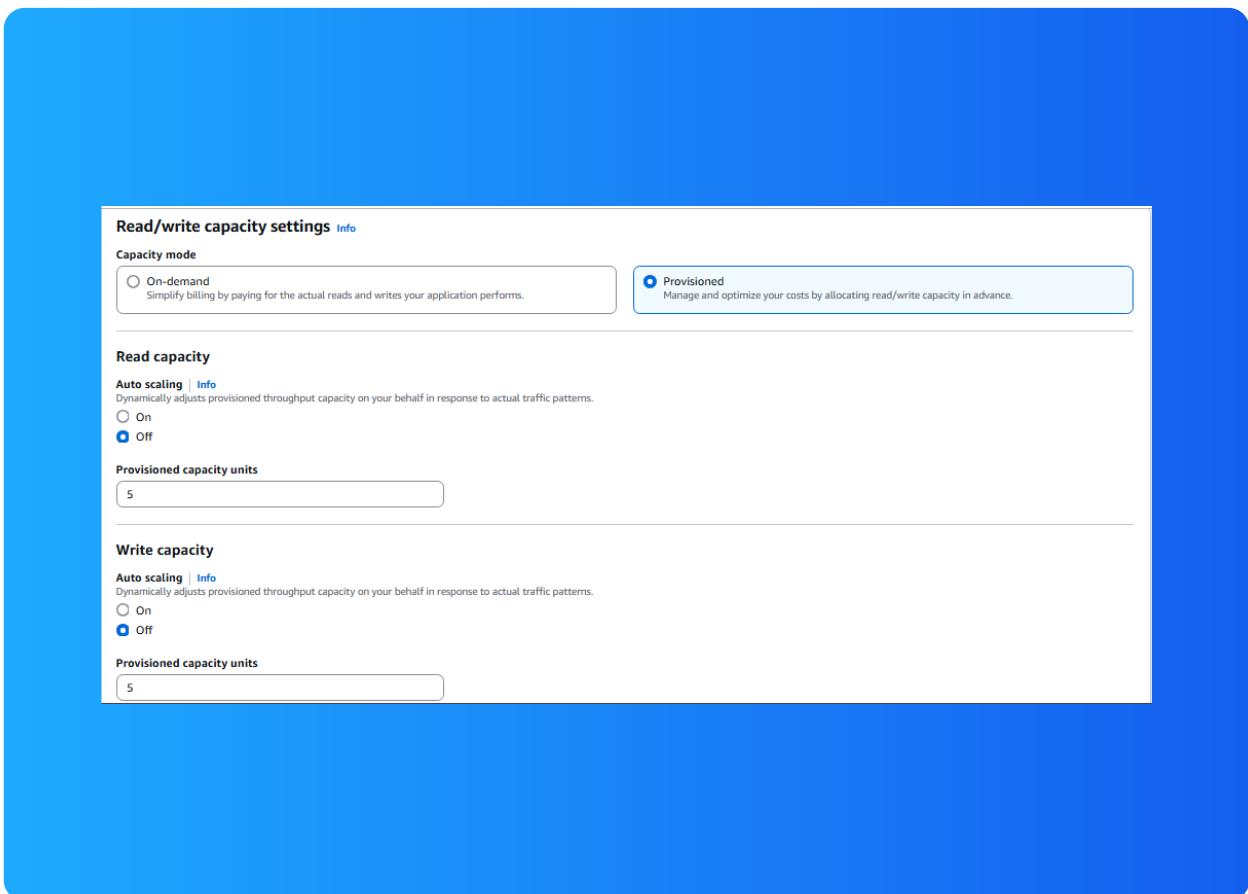
An attribute is a key-value pair that represents a piece of data in a DynamoDB item. It defines the properties of an item, such as a name, type, or value. For example, in a user table, Name can be an attribute with specific values for each item

The screenshot shows the AWS DynamoDB console interface. At the top, there are two dropdown menus: 'Select a table or index' containing 'Table - NextWorkStudents' and 'Select attribute projection' containing 'All attributes'. Below these are 'Filters' and 'Run' (highlighted in orange) and 'Reset' buttons. The main area displays the results of a query titled 'Items returned (1)'. The result table has two columns: 'StudentName (String)' and 'ProjectsComplete'. One item is listed: 'Nikko' with a value of '4'. To the right of the table are 'Actions' (with a dropdown arrow), 'Create item', and navigation controls (< 1 > and settings). A gear icon is also present.

Read and Write Capacity

Read capacity units (RCUs) and write capacity units (WCUs) are how DynamoDB measures data usage. RCUs show how many reads your table can handle per second, and WCUs show how many writes it can handle. These ensure your table runs smoothly.

Amazon DynamoDB's Free Tier covers 25 WCUs, 25 RCUs, and 25 GB of storage per month. I turned off auto-scaling because it adjusts capacity based on usage, which could exceed Free Tier limits and result in extra charges.





Using CLI and CloudShell

AWS CloudShell is a browser-based shell that lets you run AWS commands and manage resources directly in the AWS Management Console. It provides a pre-configured command-line environment, making it easy to interact with AWS services without setup.

AWS CLI is a command-line tool that allows you to interact with AWS services. It lets you manage resources, automate tasks, and run commands directly from your terminal or scripts, making AWS management faster and more efficient.

I ran a CLI command in AWS CloudShell that created four DynamoDB tables: ContentCatalog, Forum, Post, and Comment. Each table has specific attributes, partition keys, and sort keys defined for efficient data organization and querying.

```
[CloudShell]user@ip-10-136-37-216 ~]# ssm dynamodb create-table \
> {
>     "TableName": "Forum",
>     "AttributeDefinitions": [
>         {
>             "AttributeName": "Id",
>             "AttributeType": "N"
>         },
>         {
>             "AttributeName": "Key",
>             "AttributeType": "S"
>         }
>     ],
>     "ProvisionedThroughput": {
>         "ReadCapacityUnits": 1,
>         "WriteCapacityUnits": 1
>     },
>     "TableDescription": "TableStatus"
> }
[CloudShell]user@ip-10-136-37-216 ~]# ss dynamodb create-table \
> {
>     "TableName": "Forum",
>     "AttributeDefinitions": [
>         {
>             "AttributeName": "Id",
>             "AttributeType": "N"
>         },
>         {
>             "AttributeName": "Key",
>             "AttributeType": "S"
>         }
>     ],
>     "ProvisionedThroughput": {
>         "ReadCapacityUnits": 1,
>         "WriteCapacityUnits": 1
>     },
>     "TableDescription": "TableStatus"
> }
[CloudShell]user@ip-10-136-37-216 ~]# ss dynamodb create-table \
> {
>     "TableName": "Forum",
>     "AttributeDefinitions": [
>         {
>             "AttributeName": "Id",
>             "AttributeType": "N"
>         },
>         {
>             "AttributeName": "Key",
>             "AttributeType": "S"
>         }
>     ],
>     "ProvisionedThroughput": {
>         "ReadCapacityUnits": 1,
>         "WriteCapacityUnits": 1
>     },
>     "TableDescription": "TableStatus"
> }
[CloudShell]user@ip-10-136-37-216 ~]# ss dynamodb create-table \
> {
>     "TableName": "Forum",
>     "AttributeDefinitions": [
>         {
>             "AttributeName": "Id",
>             "AttributeType": "N"
>         },
>         {
>             "AttributeName": "Key",
>             "AttributeType": "S"
>         }
>     ],
>     "ProvisionedThroughput": {
>         "ReadCapacityUnits": 1,
>         "WriteCapacityUnits": 1
>     },
>     "TableDescription": "TableStatus"
> }
[CloudShell]user@ip-10-136-37-216 ~]# ss dynamodb create-table \
> {
>     "TableName": "Forum",
>     "AttributeDefinitions": [
>         {
>             "AttributeName": "Id",
>             "AttributeType": "N"
>         },
>         {
>             "AttributeName": "Key",
>             "AttributeType": "S"
>         }
>     ],
>     "ProvisionedThroughput": {
>         "ReadCapacityUnits": 1,
>         "WriteCapacityUnits": 1
>     },
>     "TableDescription": "TableStatus"
> }
```



Loading Data with CLI

I ran a CLI command in AWS CloudShell that used `batch-write-item` to load multiple items from JSON files into DynamoDB tables. The `--request-items` flag specified the local JSON file for each table: ContentCatalog, Forum, Post, and Comment.

```
[cloudshell-user@ip-10-136-37-216 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://ContentCatalog.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-136-37-216 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Forum.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-136-37-216 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Post.json
{
  "UnprocessedItems": {}
}
[cloudshell-user@ip-10-136-37-216 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Comment.json
{
  "UnprocessedItems": {}
}
```

Observing Item Attributes

Attributes			Add new attribute ▾
Attribute name	Value	Type	
Id - Partition key	1	Number	<button>Remove</button>
Authors	<button>Insert a field ▾</button>	List	<button>Remove</button>
ContentType	Project	String	<button>Remove</button>
Difficulty	Easy peasy	String	<button>Remove</button>
Price	0	Number	<button>Remove</button>
ProjectCategory	Storage	String	<button>Remove</button>
Published	<input checked="" type="radio"/> True <input type="radio"/> False	Boolean	<button>Remove</button>
Title	Host a Website on Amazon S3	String	<button>Remove</button>
URL	aws-host-a-website-on-s3	String	<button>Remove</button>

I checked a ContentCatalog item, which had the following attributes: Id, Authors, ContentType, Difficulty, Price, ProjectCategory, Published, Title, and URL. These define the project's details, like type, difficulty and where it was hosted

I checked another ContentCatalog item, which had a different set of attributes: Id , ContentType, Price, Services, Title, URL, and VideoType. These attributes describe a video, including its type, title, and link.

Benefits of DynamoDB

A benefit of DynamoDB over relational databases is flexibility because each item can have different attributes. You don't need fixed rows and columns, so you can easily store different types of data in one table.

Another benefit over relational databases is speed because DynamoDB uses partition keys to quickly find data and handles high-speed reads and writes without complex joins. This makes it much faster for handling large amounts of data.

Items returned (6)							
	Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory	Published
<input type="checkbox"/>	3	[{"S": "Ne...}	Project	Easy peasy	0	AI/ML	true
<input type="checkbox"/>	2	[{"S": "Ne...}	Project	Easy peasy	0	Analytics	true
<input type="checkbox"/>	203		Video		0		
<input type="checkbox"/>	202		Video		0		
<input type="checkbox"/>	201		Video		0		
<input type="checkbox"/>	1	[{"S": "Nat...}	Project	Easy peasy	0	Storage	true



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

