



# Load Data into DynamoDB



Benjamin Kofi Yankey

Items returned (6)						<input type="button" value="G"/>	Actions ▾	<input type="button" value="Create item"/>
<input type="checkbox"/>	Id (Number)	Authors	ContentType	Difficulty	Price			
<input type="checkbox"/>	<a href="#">3</a>	[{"S": "N..."}]	Project	Easy peasy	0			
<input type="checkbox"/>	<a href="#">2</a>	[{"S": "Ne..."}]	Project	Easy peasy	0			
<input type="checkbox"/>	<a href="#">203</a>		Video		0			
<input type="checkbox"/>	<a href="#">202</a>		Video		0			
<input type="checkbox"/>	<a href="#">201</a>		Video		0			
<input type="checkbox"/>	<a href="#">1</a>	[{"S": "Nat..."}]	Project	Easy peasy	0			



**Benjamin Kofi Yankey**

NextWork Student

[NextWork.org](http://NextWork.org)

---

# Introducing Today's Project!

## What is Amazon DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service that provides fast, scalable performance. It's useful for handling large amounts of dynamic data with high availability, automatic scaling, and minimal management.

## How I used Amazon DynamoDB in this project

I used Amazon DynamoDB in today's project to create a table, load data into it, and manage items with unique attributes. This allowed me to store and query data efficiently for my web app.

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

One thing I didn't expect in this project was how easy it was to scale DynamoDB automatically. The service seamlessly handled capacity adjustments, which simplified managing data for the web app.

## This project took me...

This project took me about a few hours, including setting up the DynamoDB table, loading data, and connecting it to my web app. The automation and user-friendly interface of AWS helped speed up the process.



# Create a DynamoDB table

DynamoDB tables organize data using items and attributes. Each item is a unique record identified by a primary key, which can be a single or composite key. Attributes are key-value pairs within each item.

An attribute is a piece of data about an item in DynamoDB. For instance, if the item is Nikko, an attribute could be the number of projects he completed. Each item can have a unique set of attributes.

A screenshot of the AWS DynamoDB console. At the top, a green success message box displays: "Completed. Read capacity units consumed: 0.5". Below this, a table titled "Items returned (1/1)" shows one item. The table has two columns: "StudentName (String)" and "ProjectsComplete". The item row contains a checked checkbox, the value "Nikko", and the value "4". To the right of the table are "Actions" and "Create item" buttons, along with navigation controls for pages and items.

StudentName (String)	ProjectsComplete
Nikko	4



# Read and Write Capacity

Read capacity units (RCUs) and write capacity units (WCUs) are measures used in Amazon DynamoDB to define the throughput capacity for reading and writing data. RCUs determine how many reads per second, while WCUs specify writes.

Amazon DynamoDB's Free Tier covers 25 GB of storage, 25 read capacity units (RCUs), and 25 write capacity units (WCUs) per month for the first 12 months. I turned off auto scaling because I wanted to manage my capacity manually.

**Read/write capacity settings** [Info](#)

**Capacity mode**

**Provisioned**  
Manage and optimize your costs by allocating read/write capacity in advance.

**On-demand**  
Simplify billing by paying for the actual reads and writes your application performs.

**Read capacity**

[Auto scaling](#) | [Info](#)  
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

**On**

**Off**

Minimum capacity units	Maximum capacity units	Target utilization (%)
1	10	70

**Write capacity**

[Auto scaling](#) | [Info](#)  
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

**On**

**Off**

Minimum capacity units	Maximum capacity units	Target utilization (%)
1	10	70



Benjamin Kofi Yankey

NextWork Student

[NextWork.org](http://NextWork.org)

# Using CLI and CloudShell

AWS CloudShell is a browser-based shell that allows users to manage AWS resources directly from their web browser. It provides a pre-configured environment with the AWS CLI and essential tools for easy access to AWS services.

AWS CLI is a command-line interface tool that allows users to interact with AWS services using commands in a terminal. It enables automation of tasks and management of resources without needing the AWS Management Console.

I ran a CLI command in AWS CloudShell that created a DynamoDB table, specifying the table name and defining the primary key attributes to organize and store data effectively.

```
> --attribute-definitions \
>   AttributeName=Id,AttributeType=N \
> -key-schema \
>   AttributeName=Id,KeyType=HASH \
> -provisioned-throughput \
>   ReadCapacityUnits=1,WriteCapacityUnits=1 \
> -query "TableDescription.TableStatus"
"CREATING"
[cloudshell-user@ip-10-130-92-73 ~]$ aws dynamodb create-table \
>   --table-name Forum \
>   --attribute-definitions \
>     AttributeName=Name,AttributeType=S \
> -key-schema \
>   AttributeName=Name,KeyType=HASH \
> --provisioned-throughput \
>   ReadCapacityUnits=1,WriteCapacityUnits=1 \
> -query "TableDescription.TableStatus"
"CREATING"
[cloudshell-user@ip-10-130-92-73 ~]$ aws dynamodb create-table \
>   --table-name Post \
>   --attribute-definitions \
>     AttributeName=ForumName,AttributeType=S \
>     AttributeName=Subject,AttributeType=S \
> -key-schema \
>   AttributeName=ForumName,KeyType=HASH \
```



**Benjamin Kofi Yankey**  
NextWork Student

[NextWork.org](https://NextWork.org)

# Loading Data with CLI

I ran a CLI command in AWS CloudShell that loaded data into my DynamoDB tables, specifying the items and attributes to ensure the tables were populated correctly for future queries and updates.

```
[cloudshell-user@ip-10-130-92-73 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://ContentCatalog.json
{
    "UnprocessedItems": {}
}
[cloudshell-user@ip-10-130-92-73 nextworksampleddata]$ ]
```



# Observing Item Attributes

Attributes

Add new attribute ▾

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

Cancel Save Save and close

I checked a ContentCatalog item, which had the following attributes: `ProjectID`, `ProjectName`, `Description`, `StartDate`, `EndDate`, and `Status`. These attributes provide detailed information about each project.

I checked another ContentCatalog item, which had a different set of attributes: `VideoID`, `Title`, `Duration`, `Resolution`, `UploadDate`, and `Category`. These attributes describe the video's details.



# Benefits of DynamoDB

A benefit of DynamoDB over relational databases is flexibility, because it allows each item to have a unique set of attributes. Unlike relational databases, which require fixed columns, DynamoDB can store different attributes for each item.

Another benefit over relational databases is speed, because DynamoDB uses a distributed, NoSQL architecture. It automatically partitions data across servers, enabling fast read/write performance even at scale.

Items returned (6)						
	Id (Number)	Authors	ContentType	Difficulty	Price	
<input type="checkbox"/>	<a href="#">3</a>	[{"S": "..."}]	Project	Easy peasy	0	
<input type="checkbox"/>	<a href="#">2</a>	[{"S": "Ne..."}]	Project	Easy peasy	0	
<input type="checkbox"/>	<a href="#">203</a>		Video		0	
<input type="checkbox"/>	<a href="#">202</a>		Video		0	
<input type="checkbox"/>	<a href="#">201</a>		Video		0	
<input type="checkbox"/>	<a href="#">1</a>	[ {"S": "Nat..."}]	Project	Easy peasy	0	



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

