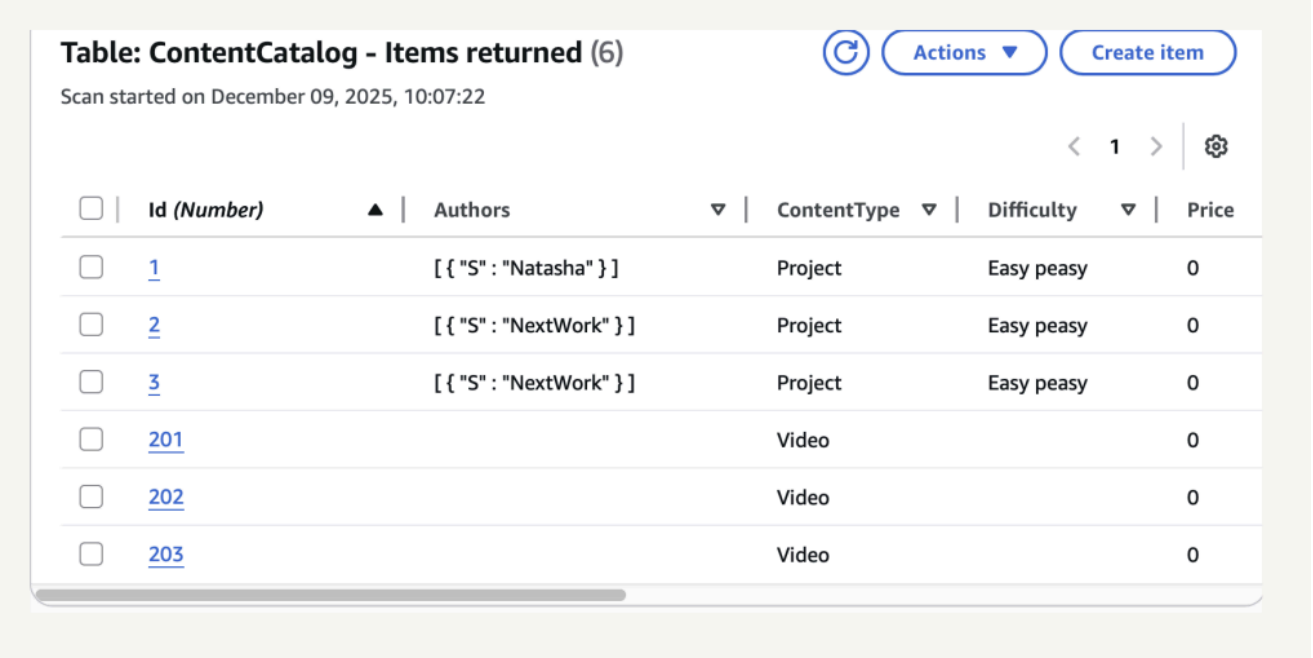
Load Data into a DynamoDB Table

Erik Gonzalez



Introducing Today's Project!

What is Amazon DynamoDB?

DynamoDB is a fully managed, serverless NoSQL database service from AWS that provides fast, predictable performance, seamless scalability, and high availability for applications needing to handle massive data volumes and traffic.

How I used Amazon DynamoDB in this project

In today's project, I used Amazon DynamoDB to create a table and store data and attributes into it.

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

One thing I didn't expect in this project is how each Id can have their own specific data/attributes.

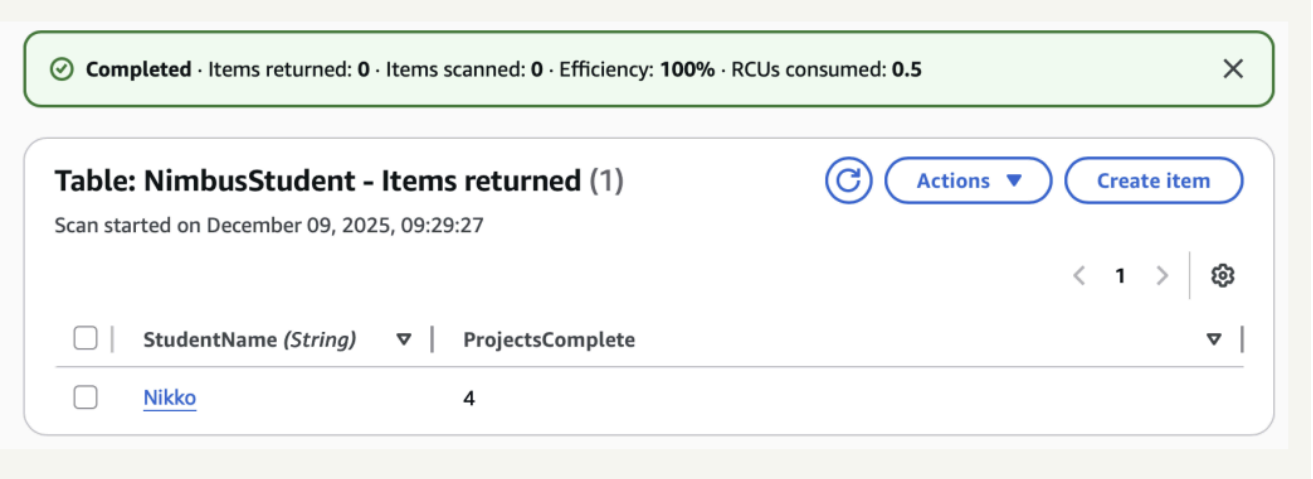
This project took me...

This project took me about an hour to complete.

Create a DynamoDB table

DynamoDB tables organize data using rows and columns, but it's actually quite different from traditional relational databases. Instead of looking at this as a spreadsheet, look at a DynamoDB table as a list of items (i.e. StudentNames, like Nikko), each with their own list of attributes (e.g. ProjectsComplete).

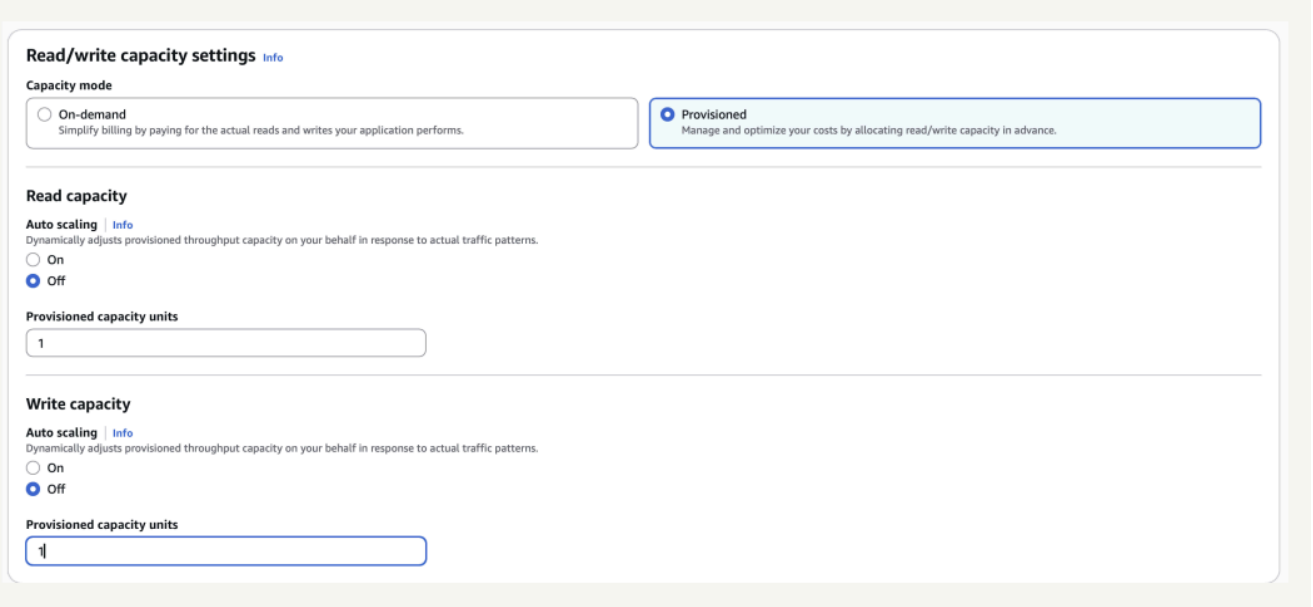
An attribute is like a piece of data about an item. In this case, our item is Nikko and the attribute is the number of projects Nikko completed.



Read and Write Capacity

In AWS DynamoDB, Read Capacity Units (RCUs) and Write Capacity Units (WCUs) determine how much throughput your table can handle. They control performance and cost when using Provisioned Capacity mode.

Amazon DynamoDB's Free Tier covers gives you 25GB of data storage, plus 25 Write and 25 Read Capacity Units (WCU, RCU). This is enough to handle 200M requests per month... all for free. I turned off auto scaling because I wanted to reduce costs, if at all in this project.



Using CLI and CloudShell

AWS CloudShell is a shell in your AWS Management Console, which means it's a space for you to run code! The awesome thing about AWS CloudShell is that it already has AWS CLI pre-installed.

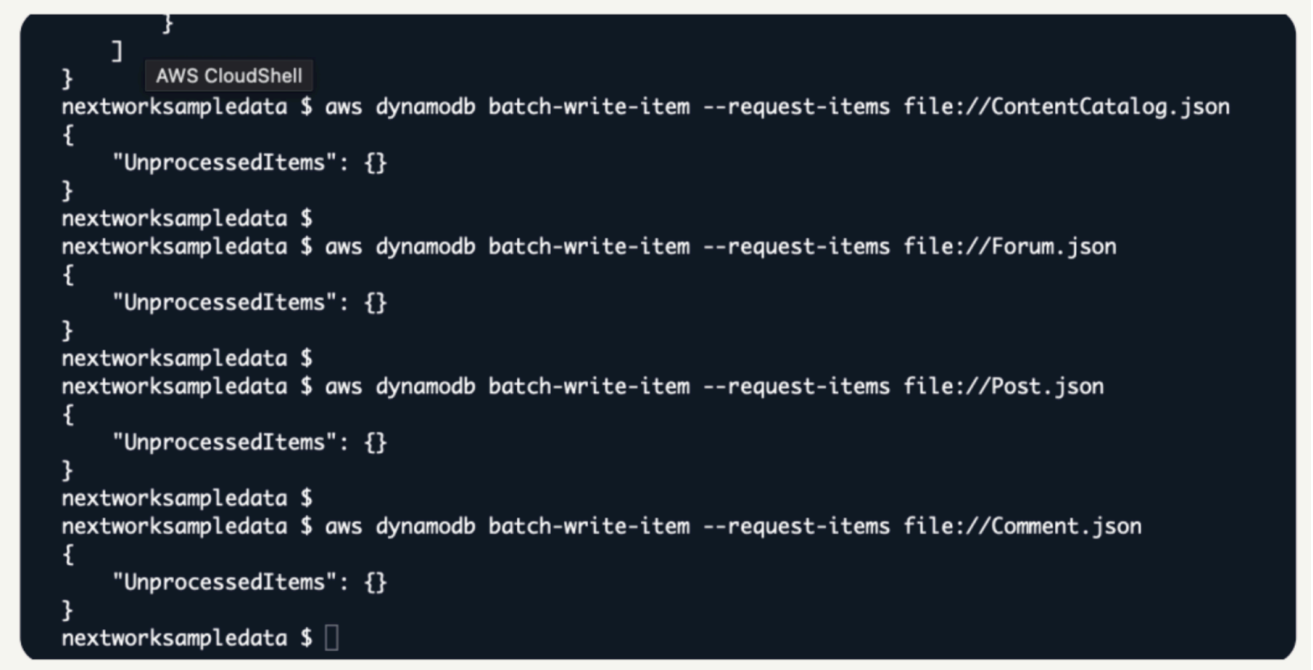
AWS CLI is a software that lets you create, delete and update AWS resources with commands instead of clicking through your console.

I ran a CLI command in AWS CloudShell that created four new tables in AWS DynamoDB, each with specific attributes and settin

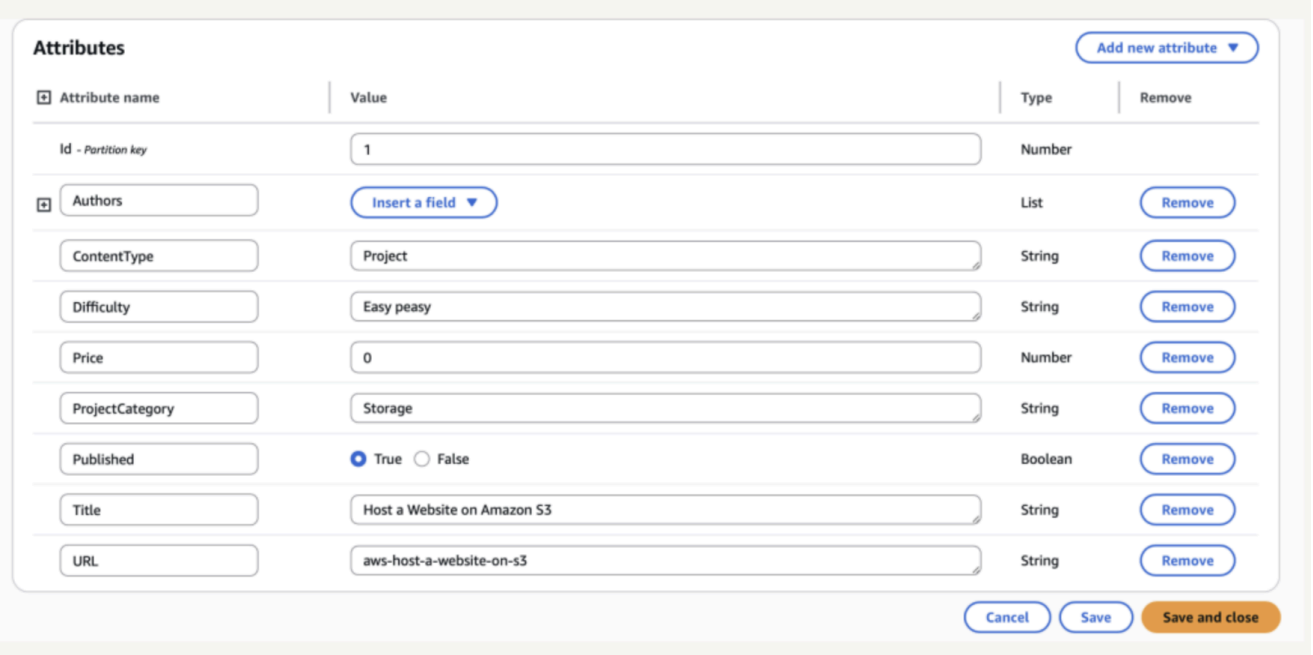

CloudShell
us-east-1
AWS CloudShell
> --provisioned-throughput \
> --attribute-definitions \
> AttributeName=Name,AttributeType=S \
> --key-schema \
> AttributeName=Name,KeyType=HASH \
> --provisioned-throughput \
> ReadCapacityUnits=1,WriteCapacityUnits=1 \
> --query "TableDescription.TableStatus"
"CREATING"
~ $ aws dynamodb create-table \
> --table-name Post \
> --attribute-definitions \
> AttributeName=ForumName,AttributeType=S \
> AttributeName=Subject,AttributeType=S \
> --key-schema \
> AttributeName=ForumName,KeyType=HASH \
> AttributeName=Subject,KeyType=RANGE \
> --provisioned-throughput \
> ReadCapacityUnits=1,WriteCapacityUnits=1 \
> --query "TableDescription.TableStatus"
"CREATING"
~ $ aws dynamodb create-table \
> --table-name Comment \
> --attribute-definitions \
> AttributeName=Id,AttributeType=S \
> AttributeName=CommentDateTime,AttributeType=S \
> --key-schema \
> AttributeName=Id,KeyType=HASH \
> AttributeName=CommentDateTime,KeyType=RANGE \
> --provisioned-throughput \
> ReadCapacityUnits=1,WriteCapacityUnits=1 \
> --query "TableDescription.TableStatus"
"CREATING"
~ $

Loading Data with CLI

I ran a CLI command in AWS CloudShell that is used to load or insert multiple items into DynamoDB tables!



Observing ItemAttributes



I checked a ContentCatalog item, which had the following attributes: Content Type, Difficulty, Price, Project Category, Published, Title, and URL.

I checked another ContentCatalog item, which had a different set of attributes as it does not have StudentProjects that I made for the other Id, but instead, it has VideoType and Services.

Benefits of DynamoDB

A benefit of DynamoDB over relational databases is flexibility, because every item having their own unique set of attributes is a huge advantage when items in a table could look different from each other. For example, e-commerce sites and shopping carts need to store different types of products with different attributes in the same place.

Another benefit over relational databases is speed, because DynamoDB tables can use partition keys to split up a table and quickly find the items they're looking for. Relational databases have to scan through the entire table to find data, which can slow down performance.

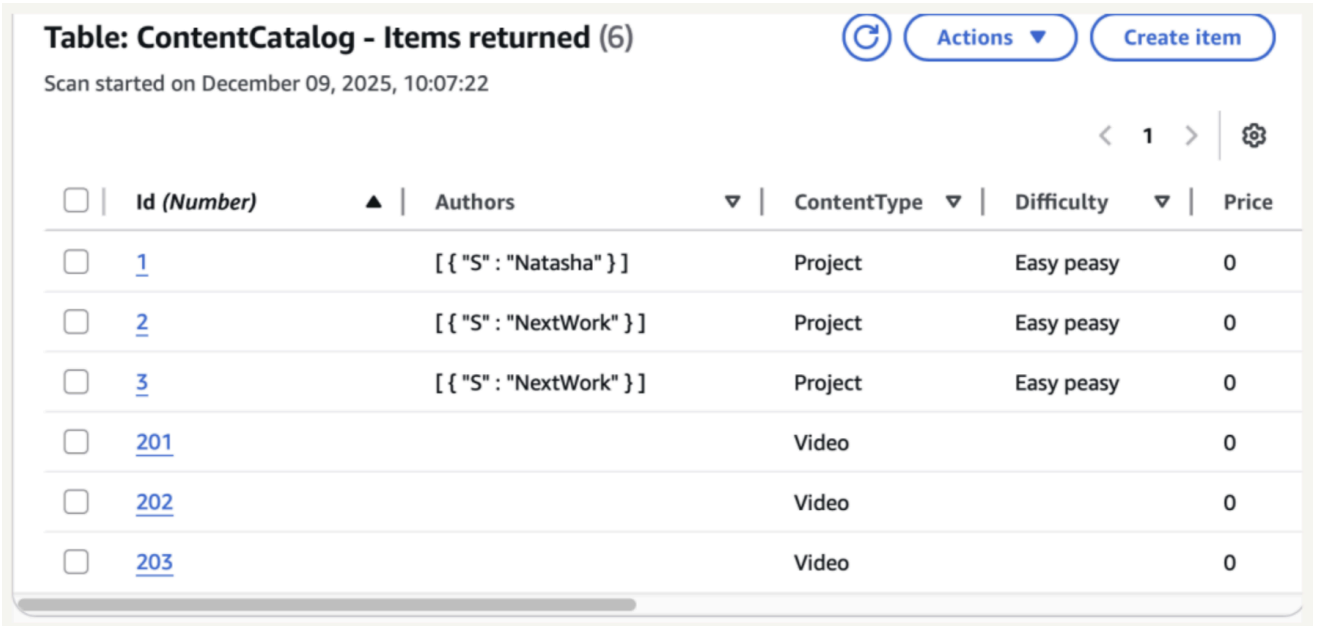


Table: ContentCatalog - Items returned (6)
Actions
Create item
Scan started on December 09, 2025, 10:07:22
1
Id (Number)
Authors
ContentType
Difficulty
Price
1
[{ "S" : "Natasha" }]
Project
Easy peasy
0
2
[{ "S" : "NextWork" }]
Project
Easy peasy
0
3
[{ "S" : "NextWork" }]
Project
Easy peasy
0
201
Video
0
202
Video
0
203
Video
0