



Load Data into DynamoDB



tahirgroot@gmail.com

Items returned (6)									Actions ▾	Create item
□	ID (Number)	Authors	Content Type	Difficulty	Price	Project Category	Published	Services		
□	3	[{"S": "Ne..."}]	Project	Easy peasy	0	AI/ML	true		< 1 >	
□	2	[{"S": "Ne..."}]	Project	Easy peasy	0	Analytics	true			
□	203		Video		0			[{"S": "Am..."}]		
□	202		Video		0			[{"S": "Am..."}]		
□	201		Video		0			[{"S": "Am..."}]		
□	1	[{"S": "Nat..."}]	Project	Easy peasy	0	Storage	true			

 TA

tahirgroot@gmail.com

NextWork Student

NextWork.org

Introducing Today's Project!

What is Amazon DynamoDB?

Amazon DynamoDB is a nonrelational database service with AWS that organizes data using key-value pairs.

How I used Amazon DynamoDB in this project

In today's Project We used Dynamo DB to create five tables! We used Cloud shell and CLI to load data into our tables quickly. We also compared Dynamo DB to a relational database by understanding how items and attributes work!

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

n/a

This project took me...

30 mins

Create a DynamoDB table

DynamoDB tables organises data using items and attributes! Every single item is recorded with a set of attributes. Items can have any number of attributes (minimum 1, for the partition key)

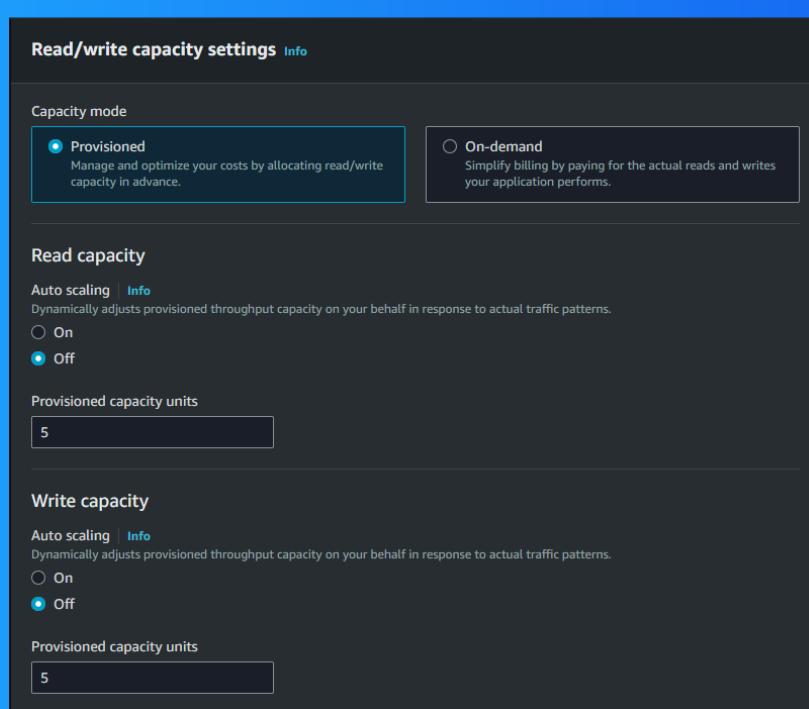
An attribute is like a piece of data about an item in the Dynamo DB table. For example, if an item was a specific student, attributes could be Student Name, Projects Completed, Email, and Signup Date. The number of attributes for each item is flexib!

Items returned (1)		Action	Actions	Create item
<input type="checkbox"/>	Groot279Name (String)	Project Complete	< 1 >	⚙️ 🔍
<input type="checkbox"/>	Nikko	4		

Read and Write Capacity

Read capacity units (RCUs) and write capacity units (WCUs) are units that measure my dynamo tables performance, DynamoDB pricing is based on RCUs and WCUs, so the more RCUs/WCUs consumed, the more expensive the project

Amazon DynamoDB's Free Tier covers 25 RCUs and WCUs monthly. I turned off auto-scaling because if Dynamo DB automatically scales up my operations (which charges more RCUs and WCUs), I can incur higher charges.



Using CLI and CloudShell

AWS CloudShell is an environment that lets us run code to interact with our AWS resources/services. Using AWS Cloud shell is handy because AWS CLI is already installed. so that we can run command right away.

AWS CLI (Command Line Interface) 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 commands to create four new tables in AWS DynamoDB, each with specific attributes and settings.

Tables (5) <small>Info</small>										
	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size	Table class
<input type="checkbox"/>	Comment	Active	Id (S)	CommentDateTime (S)	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	ContentCatalog	Active	Id (N)	-	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	Forum	Active	Name (S)	-	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	Groot279Students	Active	Groot279Name (S)	-	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	Post	Active	ForumName (S)	Subject (S)	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard

Loading Data with CLI

I ran a CLI command in AWS CloudShell that will load multiple pieces of data (i.e., multiple items) into the DynamoDB tables I set up in the previous step. This AWS CLI command is structured as 'aws dynamodb batch write-item --request-items file://...'

```
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://ContentCatalog.json
{
    "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Forum.json
{
    "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$ aws dynamodb batch-write-item --request-items file://Post.json
{
    "UnprocessedItems": {}
}
[cloudshell-user@ip-10-134-44-40 nextworksampleddata]$
[cloudshell-user@ip-10-134-44-40 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	Insert a field ▾	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>

Cancel Save Save and close

I checked a ContentCatalog item with the following attributes: id, Authors, Content Type, Difficulty Price, Project Category, Published, Title and URL!

I checked another ContentCatalog item, which had a different set of attributes: Services, Title, VideoType e.t.c

Benefits of DynamoDB

A benefit of DynamoDB over relational databases is flexibility because items can have their attributes. This is great for scenarios where a table has different types of data, and some attributes

Another benefit over relational databases is speed. Dynamo DB can find items quickly using partition keys, which is faster than relational databases, which need to scan entire tables to find specific pieces of data.

Items returned (6)								<input type="button" value="C"/>	Actions ▾	<input type="button" value="Create item"/>
	Id (Number)	Authors	ContentType	Difficulty	Price	ProjectCategory	Published	Services	< 1 > <input type="button" value=""/>	
<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			[{"S": "Am...}		
<input type="checkbox"/>	202		Video		0					
<input type="checkbox"/>	201		Video		0			[{"S": "Am...}		
<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

