



Amazon Dynamo DB with AWS CLI

1. In this lab we are going to work with Dynamo DB using AWS CLI. First, we will start with creating a Dynamo DB table then we'll write some data into this table. After that, we will read, update, and query the data in our Dynamo DB table.
2. For that, I have created some command text files which we will follow to work with the DynamoDB.
3. Now let's start by creating our Dynamo DB table first. For that, you need to run this command.
4. Below you can see that our table has been created.

```
○ PS D:\AWS Serveless\Amazon Dynamo DB with CLI> aws dynamodb create-table `>>   --table-name Order `>>   --attribute-definitions `>>     AttributeName=id,AttributeType=S `>>     AttributeName=status,AttributeType=S `>>   --key-schema `>>     AttributeName=id,KeyType=HASH `>>     AttributeName=status,KeyType=RANGE `>>   --provisioned-throughput `>>     ReadCapacityUnits=5,WriteCapacityUnits=5 `>>   --table-class STANDARD`>>TableDescription:>>  AttributeDefinitions:>>    - AttributeName: id `>>      AttributeType: S `>>    - AttributeName: status `>>      AttributeType: S`>>CreationDateTime: '2024-11-16T11:23:36.284000+05:30'`>>DeletionProtectionEnabled: false`>>ItemCount: 0`>>KeySchema:>>  - AttributeName: id
```

```
PS D:\AWS Serveless\Amazon Dynamo DB with CLI> aws dynamodb describe-table --table-name Order
>>
Table:
  AttributeDefinitions:
    - AttributeName: id
      AttributeType: S
    - AttributeName: status
      AttributeType: S
  CreationDateTime: '2024-11-16T11:23:36.284000+05:30'
  DeletionProtectionEnabled: false
  ItemCount: 0
  KeySchema:
    - AttributeName: id
      KeyType: HASH
    - AttributeName: status
      KeyType: RANGE
  ProvisionedThroughput:
    NumberOfDecreasesToday: 0
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
  TableArn: arn:aws:dynamodb:us-east-1:878893308172:table/Order
  TableClassSummary:
    TableClass: STANDARD
```

5. Also, if you visit the console and open Dynamo DB just to confirm then you will see your table there.

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	...
Order	Active	id (S)	status (S)	0	0	Off		

6. Now we are going to use a command to insert several items into our Order table. Below you can see that we are putting two items into our table.

```

windows:

aws dynamodb put-item \
--table-name Order \
--item '{
    "id": {"S": "1"}, 
    "status": {"S": "IN_PROGRESS"}, 
    "desc": {"S": "iphone order"}, 
    "orderDate": {"S": "2022-05-15"}
}'

aws dynamodb put-item \
--table-name Order \
--item '{
    "id": {"S": "1"}, 
    "status": {"S": "DELIVERY"}, 
    "desc": {"S": "samsung order"}, 
    "orderDate": {"S": "2022-05-19"}
}'
```

7. Use the bash terminal in VS Code to run these commands so that you can put items in your table.
8. I have used bash commands because I was unable to run them in the PowerShell terminal.

```

PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb put-item \
  --table-name Order \
  --item '{"id": {"S": "1"}, "status": {"S": "IN_PROGRESS"}, "desc": {"S": "iphone order"}, "orderDate": {"S": "2022-05-15"}}'

PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb put-item \
  --table-name Order \
  --item '{"id": {"S": "1"}, "status": {"S": "DELIVERY"}, "desc": {"S": "samsung order"}, "orderDate": {"S": "2022-05-19"}}'

PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$
```

9. Once your items are added go to the console to verify them.

10. Now we are going to read the items from our table using the CLI. So, use this command to read the data.

```
PS D:\AWS Serveless\Amazon Dynamo DB with CLI> aws dynamodb get-item ` 
>>   --consistent-read ` 
>>   --table-name Order ` 
>>   --key "{\"id\": {\"S\": \"1\"}, \"status\": {\"S\": \"DELIVERY\"}}"
● >>
Item:
  desc:
    S: samsung order
  id:
    S: '1'
  orderDate:
    S: '2022-05-19'
  status:
    S: DELIVERY

○ PS D:\AWS Serveless\Amazon Dynamo DB with CLI>
```

11. Our next step is to update the items in our Table. For that run the below command. Below you can see that I have updated the item into our table.
12. Here you can see that we have updated the description, id, order date, and the status.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb update-item \
    --table-name Order \
    --key '{"id": {"S": "1"}, "status": {"S": "DELIVERY"}}' \
    --update-expression "SET orderDate = :newval" \
    --expression-attribute-values '{":newval": {"S": "2022-06-06"}}' \
    --return-values ALL_NEW
Attributes:
  desc:
    S: samsung order
  id:
    S: '1'
  orderDate:
    S: '2022-06-06'
  status:
    S: DELIVERY
```

13. Below you can see that we have used the query command to check for the data in our Dynamo DB table.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb query \
    --table-name Order \
    --key-condition-expression "id = :id" \
    --expression-attribute-values '{":id": {"S": "1"}}'
● ConsumedCapacity: null
Count: 2
Items:
- desc:
    S: samsung order
  id:
    S: '1'
  orderDate:
    S: '2022-06-06'
  status:
    S: DELIVERY
- desc:
    S: iphone order
  id:
    S: '1'
  orderDate:
    S: '2022-05-15'
  status:
    S: IN_PROGRESS
ScannedCount: 2
```

14. In the end we are going to perform some PartiQL CRUD operation inside our Dynamo DB table by using AWS CLI.

15. The below command will create an item.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb execute-statement --statement "INSERT INTO \"Order\" VALUE {'id':'2','status':'IN_PROGRESS'}"
• Items: []
```

Completed. Read capacity units consumed: 0.5

Items returned (3)				
	id (String)	status (String)	desc	orderDate
	2	IN_PROGRESS		
	1	DELIVERY	samsung order	2022-06-06
	1	IN_PROGRESS	iphone order	2022-05-15

16. Then we used this command to retrieve this item.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb execute-statement --statement "SELECT * FROM \"Order\" WHERE id='2' AND status='IN_PROGRESS'"
Items:
- id:
  S: '2'
  status:
  S: IN_PROGRESS
```

17. By using the below command we have update the item.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
$ aws dynamodb execute-statement --statement "UPDATE \"Order\" SET description='updated order' WHERE id='2' AND status='IN_PROGRESS'"
• Items: []
```

Items returned (3)

	id (String) ▾	status (String) ▾	desc	description	orderDate
	2	IN_PROGRESS		updated order	
	1	DELIVERY	samsung order		2022-06-06
	1	IN_PROGRESS	iphone order		2022-05-15

18. In the end we used the delete command to delete this item.

Items returned (2)

	id (String)	status (String)	desc	orderDate
<input type="checkbox"/>	1	DELIVERY	samsung order	2022-06-06
<input type="checkbox"/>	1	IN_PROGRESS	iphone order	2022-05-15

19. Lastly, we used the delete command to delete our entire table.

```
PULKIT@LAPTOP-G2CAKBK8 MINGW64 /d/AWS Serveless/Amazon Dynamo DB with CLI
● $ aws dynamodb delete-table --table-name Order
  TableDescription:
    DeletionProtectionEnabled: false
    ItemCount: 0
    ProvisionedThroughput:
      NumberOfDecreasesToday: 0
      ReadCapacityUnits: 5
      WriteCapacityUnits: 5
    TableArn: arn:aws:dynamodb:us-east-1:878893308172:table/Order
    TableClassSummary:
      TableClass: STANDARD
    TableId: 8c6448d8-d9cc-44b0-abb4-81ce4b1a98f0
    TableName: Order
    TableSizeBytes: 0
    TableStatus: DELETING
```

DynamoDB > Tables

Tables (0) Info		Actions	Delete	Create table					
<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Re
You have no tables in this account in this AWS Region.									
Create table									