Big Data on AWS

Lab Four: DynamoDB

Table of Contents

[Lab Four: DynamoDB 3](#_Toc509607442)

[Task One: Create Tables using Console 3](#_Toc509607443)

[Create the ProductCatalog Table 3](#_Toc509607444)

[Create the Forum Table 3](#_Toc509607445)

[Create the Thread Table 3](#_Toc509607446)

[Create the Reply Table 4](#_Toc509607447)

[Task Two: Load Data into Tables 4](#_Toc509607448)

[Download the Sample Data File Archive 4](#_Toc509607449)

[Load the Sample Data Into DynamoDB Tables 5](#_Toc509607450)

[Verify Data Load 5](#_Toc509607451)

[Task 3: Query the Data and CLI 5](#_Toc509607452)

[Task 4: Explore DynamoDB API/Document API and GSI 8](#_Toc509607453)

[Task 5: Setup DynamoDB Local 8](#_Toc509607454)

[Task 6: Clean up 9](#_Toc509607455)

[To Delete the Sample Tables 9](#_Toc509607456)

# Lab Four: DynamoDB

The objective of this lab is to understand DynamoDB concepts like Tables, PartitionKeys, Sort Keys, querying DynamoDB tables, Document API, GSI, configuring DynamoDB Local and performing Batch Operations using Console, CLI or SDK.

Through this lab, we will be using AWS Management console to create table using console and load data using AWS cli.

* Task 1: Create Tables using Console
* Task 2: Batch Load Data to tables using aws cli
* Task 3: Query/BatchGet the Data from Console and aws cli
* Task 4: Work with DynamoDB API/Document API and GSI with Java
* Task 5: Setting up Dynamo Local
* Task 6: Cleanup

## Task One: Create Tables using Console

### Create the ProductCatalog Table

1. Open the DynamoDB console
2. Choose Create Table.
3. In the **Create DynamoDB table** screen, do the following:
4. In the **Table** name field, type ProductCatalog.
5. For the **Primary key**, in the **Partition key** field, type Id. Set the data type to **Number**.
6. When the settings are as you want them, choose **Create**.

### Create the Forum Table

1. Open the DynamoDB console
2. Choose Create Table.
   1. In the **Create DynamoDB table** screen, do the following:
   2. In the **Table** name field, type Forum.
3. For the **Primary key**, in the **Partition key** field, type Name. Set the data type to **String**.
4. When the settings are as you want them, choose **Create**.

### Create the Thread Table

Open the DynamoDB console

Choose Create Table.

In the **Create DynamoDB table** screen, do the following:

In the **Table name** field, type Thread.

For the **Primary key**, do the following:

In the **Partition key** field, type ForumName. Set the data type to **String**.

Choose Add sort key.

In the **Sort key** field, type Subject. Set the data type to **String**.

When the settings are as you want them, choose **Create**.

### Create the Reply Table

1. Open the DynamoDB console
2. Choose Create Table.
3. In the **Create DynamoDB table** screen, do the following:

In the **Table name** field, type Reply.

For the **Primary key**, do the following:

In the **Partition key** field, type Id. Set the data type to **String**.

Choose Add sort key.

In the **Sort key** field, type ReplyDateTime. Set the data type to **String**.

In the Table settings section, deselect Use default settings.

In the Secondary indexes section, choose Add index.

In the Add index window, do the following:

For the **Primary key**, do the following:

* + - * In the **Partition key** field, type PostedBy. Set the data type to **String**.
      * Select Add sort key.
      * In the **Sort key** field, type Message. Set the data type to **String**.

In the **Index name** field, type PostedBy-Message-Index.

Set the Projected attributes to All.

Choose Add index.

When the settings are as you want them, choose **Create**

## Task Two: Load Data into Tables

### Download the Sample Data File Archive

1. Download the sample data archive (sampledata.zip) using this link:

[sampledata.zip](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/samples/sampledata.zip" \t "_blank)

1. Extract the. json data files from the archive.
2. Copy the. json data files to your current directory.

### Load the Sample Data Into DynamoDB Tables

To load the ProductCatalog table with data, enter the following command:

aws dynamodb batch-write-item --request-items file://ProductCatalog.json

To load the Forum table with data, enter the following command:

aws dynamodb batch-write-item --request-items <file://Forum.json>

To load the Thread table with data, enter the following command:

aws dynamodb batch-write-item --request-items file://Thread.json

To load the Reply table with data, enter the following command:

aws dynamodb batch-write-item --request-items file://Reply.json

### Verify Data Load

You can use the AWS Management Console to verify the data that you loaded into the tables.

To verify the data using the AWS Management Console

Open the DynamoDB console

In the navigation pane, choose **Tables**.

In the list of tables, choose ProductCatalog.

Choose the **Items** tab to view the data that you loaded into the table.

To view an item in the table, choose its Id. (If you want, you can also edit the item.)

To return to the list of tables, choose **Cancel**.

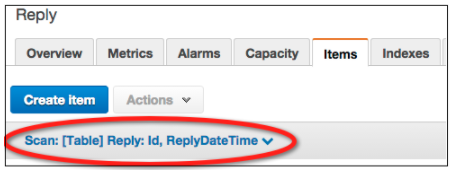
Repeat this procedure for each of the other tables you created:

* Forum
* Thread
* Reply

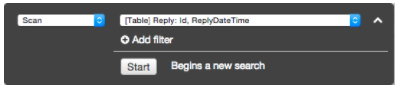
## Task 3: Query the Data and CLI

In this step, you will try some simple queries against the tables that you created, using the DynamoDB console.

1. Open the DynamoDB console
2. In the navigation pane, choose **Tables**.
3. In the list of tables, choose Reply.
4. Choose the **Items** tab to view the data that you loaded into the table.
5. Choose the data filtering link, located just below the **Create item** button.



When you do this, the console reveals a data filtering pane.



1. In the data filtering pane, do the following:
   1. Change the operation from **Scan** to **Query**.
   2. For the **Partition key**, enter the value Amazon DynamoDB#DynamoDB Thread 1.
   3. Choose **Start**. Only the items that match your query criteria are returned from the Reply table.
2. The Reply table has a global secondary index on the PostedBy and Message attributes. You can use the data filtering pane to query the index. Do the following:
   1. Change the query source from this:

[Table] Reply: Id, ReplyDateTime

to this:

[Index] PostedBy-Message-Index: PostedBy, Message

* 1. For the **Partition key**, enter the value User A.
  2. Choose **Start**. Only the items that match your query criteria are returned from PostedBy-Message-Index.

Take some time to explore your other tables using the DynamoDB console:

* ProductCatalog
* Forum
* Thread

1. Get an item

aws dynamodb get-item \

--table-name ProductCatalog \

--key '{"Id":{"N":"1"}}'

1. Projection and Consumed capacity parameters

aws dynamodb get-item \

--table-name ProductCatalog \

--key '{"Id":{"N":"1"}}' \

--consistent-read \

--projection-expression "Description, Price, RelatedItems" \

--return-consumed-capacity TOTAL

Observe the difference in the output of 1 and 2.

1. BatchGetItem

aws dynamodb batch-get-item \

--request-items file://request-items.json

Contents of request-items.json should be as below

{

"Thread": {

"Keys": [

{

"ForumName":{"S": "Amazon DynamoDB"},

"Subject":{"S": "DynamoDB Thread 1"}

},

{

"ForumName":{"S": "Amazon S3"},

"Subject":{"S": "S3 Thread 1"}

}

],

"ProjectionExpression":"ForumName, Subject, LastPostedDateTime, Replies"

}

}

1. BatchWriteItem

aws dynamodb batch-write-item \

--request-items file://request-items.json

Contents of request-items.json should be as below

{

"ProductCatalog": [

{

"PutRequest": {

"Item": {

"Id": { "N": "601" },

"Description": { "S": "Snowboard" },

"QuantityOnHand": { "N": "5" },

"Price": { "N": "100" }

}

}

},

{

"PutRequest": {

"Item": {

"Id": { "N": "602" },

"Description": { "S": "Snow shovel" }

}

}

}

]

}

## Task 4: Explore DynamoDB API/Document API and GSI

1. Now let’s Observe and perform some DDB operations using API/SDK.
   1. Connect to the DevBox.
   2. In Eclipse, open file AmazonDynamoDBSample.java
   3. Inside the main method, observe the code to create table, describe table, add items and scan.
   4. Run the application and notice that the table is created, data is added and scan result are returned.
2. Working with Global Secondary Indexes and Document API
   1. In Eclipse, open file DocumentAPIGlobalSecondaryIndexExample.java and go over the code. (Ask your instructor in case of queries)
   2. Run the application and notice the creation of Table with Indexes and data being loaded using document api.

## Task 5: Setup DynamoDB Local

1. Choose Oregon region.
2. Go to AWS DynamoDB Console and delete the **“my-favorite-movies-table”** table.
3. RDP to the DevBox (ask instructor for help)
4. Download and extract DDB Local from <https://s3.ap-south-1.amazonaws.com/dynamodb-local-mumbai/dynamodb_local_latest.zip>
5. Change directory to the ddb local directory
6. To start DynamoDB on your computer, open a command prompt window, navigate to the directory where you extracted DynamoDBLocal.jar, and type the following command:

java -Djava.library.path=./DynamoDBLocal\_lib -jar DynamoDBLocal.jar -sharedDb

1. Do not press Ctrl + C if you don’t intend to stop.
2. Open Powershell and run the following command to verify if DDB local is working with the DDB local endpoint. (Using AWS CLI)

aws dynamodb list-tables --endpoint-url http://localhost:8000

1. Open eclipse and create a copy of the **AmazonDynamoDBSample** and lets change that to make use of DDB Local rather then DDB Web Service endpoint.
   1. To use DDB Local programmatically, Comment out the below snippet in init method

dynamoDB = AmazonDynamoDBClientBuilder.standard()

.withCredentials(credentialsProvider)

.withRegion("us-west-2")

.build();

and replace with

dynamoDB = AmazonDynamoDBClientBuilder.standard().withEndpointConfiguration(

new AmazonDynamoDBClientBuilder.EndpointConfiguration("http://localhost:8000", "us-west-2"))

.build();

1. Go to AWS DynamoDB console and verify that the table is not created in DDB. So hence the operations took place with DDB Local.

**Note:** Observe in the correct region.

## Task 6: Clean up

### To Delete the Sample Tables

1. Open the DynamoDB console
2. In the navigation pane, choose **Tables**.
3. In the list of tables, choose ProductCatalog.
4. Choose **Delete Table**. You will be asked to confirm your selection.

Repeat this procedure for each of the other tables you created.