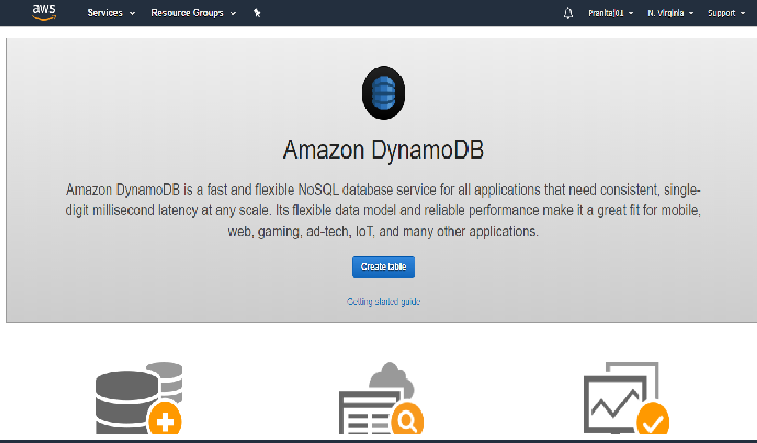
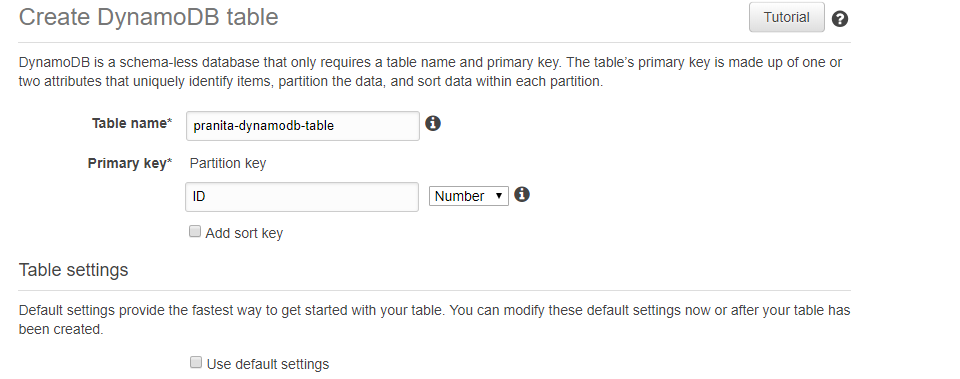
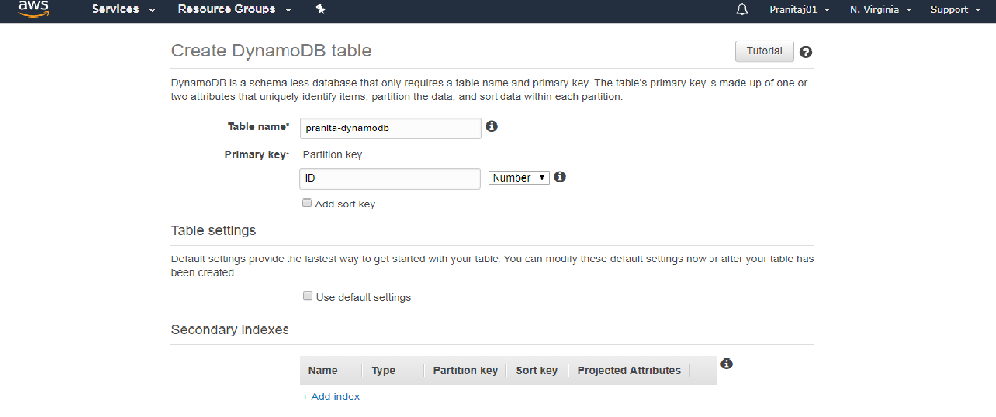
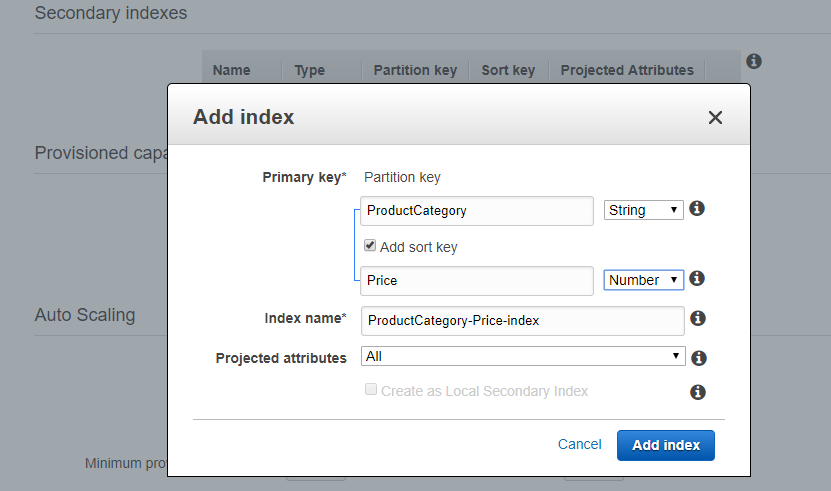
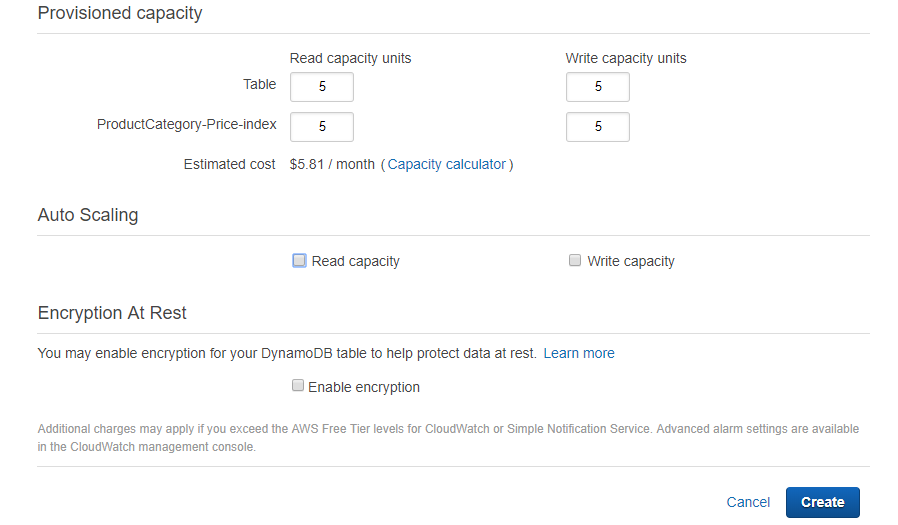
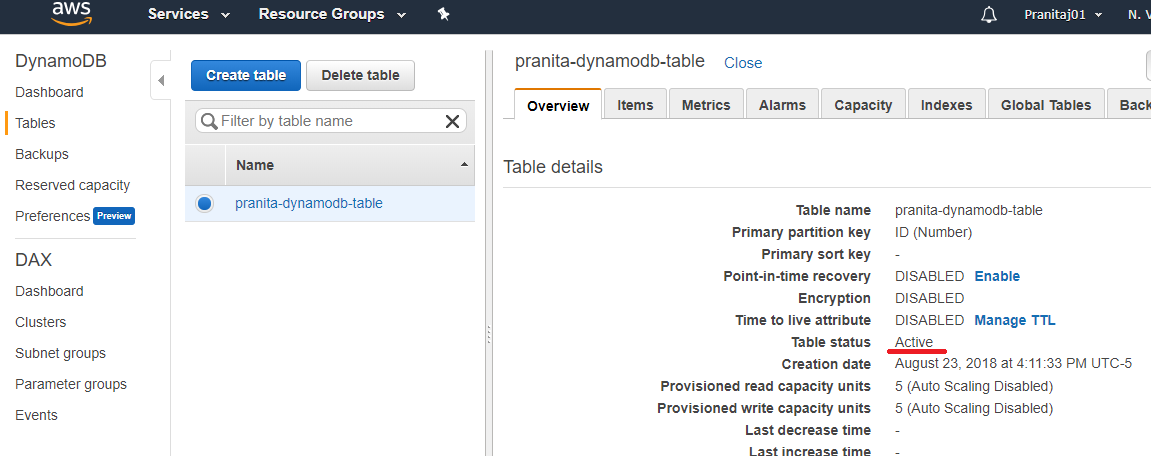
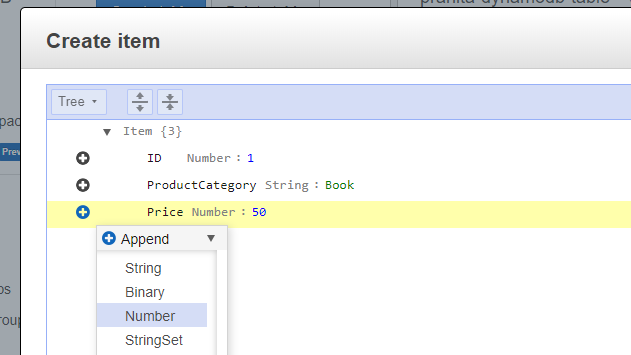
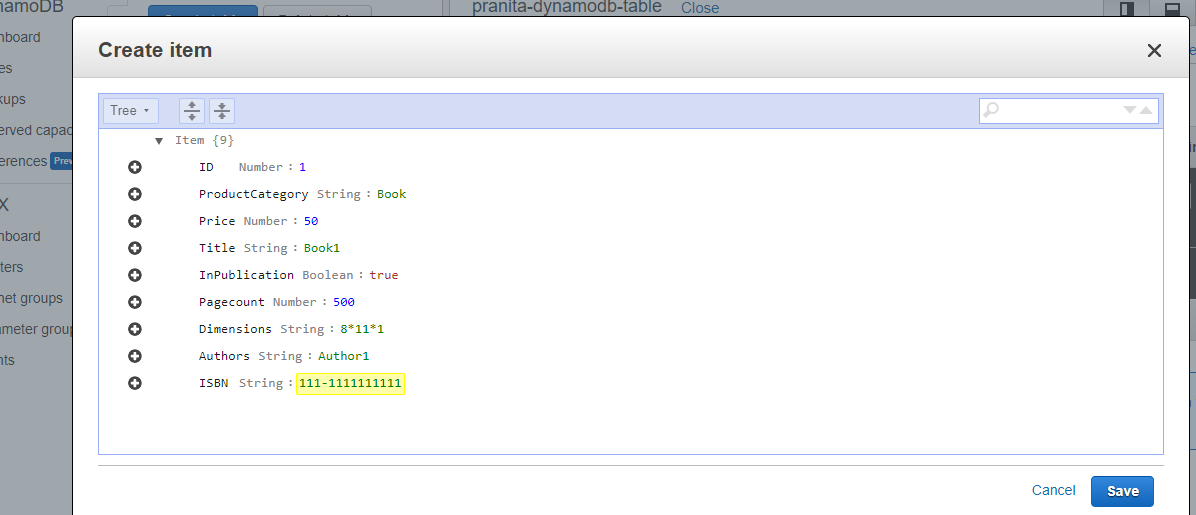
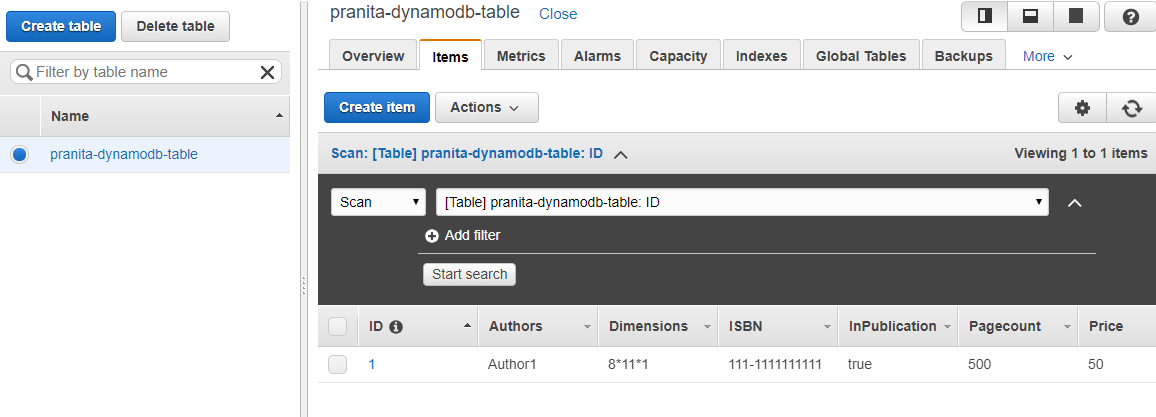
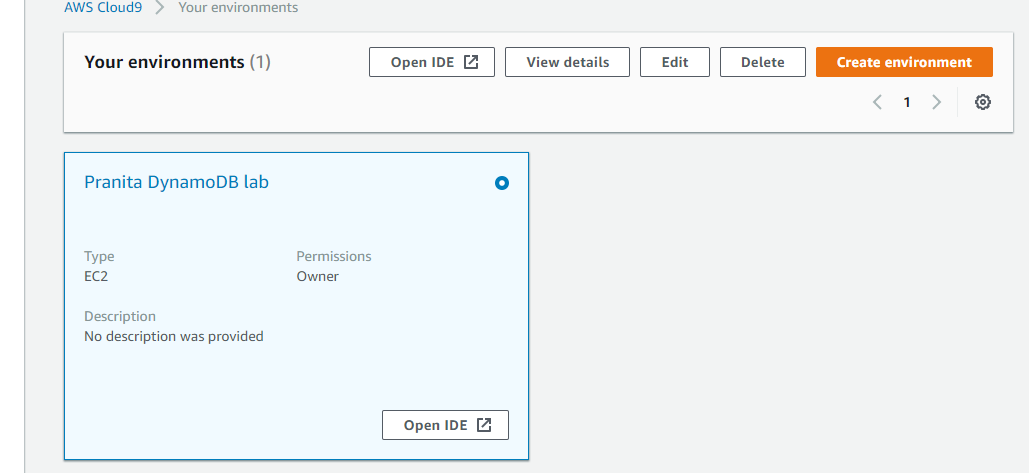
**Introduction:**  
 For obtaining broader sense of knowledge about using and implementing DyanmoDB service of AWS, here is what I did. Using AWS management console, I created a simple DynamoDB table and added items into it. Now when it comes to adding vast number of multiple items, it is almost too time consuming to add each item using the console.  
 Hence, I considered a NodeJS code to do the insertion of items into the DynamoDB table for me. Please continue reading to know how I achieved it.  
  
The following link helped me in understanding to AWS Javascript SDK:  
<https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/DynamoDB.html>  
  
I have also taken help of my trainers to prepare and review this document.

**Create DynamoDB table and add items to it using console:**  
  
On the management console, navigate **Services** -> **DynamoDB**  


Click **create table**Fill columns **table name, primary key.** Make sure you don’t leave space in any of your names.Uncheck **Use Default Settings**  
   
 Under **Secondary Indexes**, click **+ Add index**  
  


Enter the index details and click **Add index**  
   
I preferred to keep auto scaling disabled. Continue using other settings as default, and click **create**.  
****  
After a while, your table is created and you can check the **table status** to be **active**  
Now let’s add one item into the DynamoDB table using management console.  
Click **items -> create item.** Enter the values as shown below. Click on the ‘**+**’ and you can insert(a field to add before), or append(a field to add after) as many columns as you want.  
  
  
  
  
  
  
  
  
  
Please see below the final values inserted, then click **save**.  
  
  
You have successfully added one item to the table!  
  


**Import items into DynamoDB table using batchWriteItem**  
  
In this section, I will be using a NodeJS EC2 instance to import items from a JSON file into DynamoDB table. For this purpose, I used another AWS services, Cloud9 IDE.  
  
In the AWS management console, click **services -> cloud9 -> create environment.**Give your environment a name. Click **next step**.  
Enter the other details as per the below screenshot. Click **next step -> create environment.**  
This is how my environment looks like on my console  
  
  
Click **Open IDE** and paste the following code in it editor. The below code will do the necessary SDK settings and list your DynamoDB tables in the output.  
// Load the AWS SDK for Node.js

var AWS = require('aws-sdk');

/\*\* \* Don't hard-code your credentials!

\* Create an IAM role for your EC2 instance instead.

\* For development an IAM role is not required for Cloud9 \*/

// Set your region

AWS.config.region = 'us-east-1';

var db = new AWS.DynamoDB();

db.listTables(function(err, data) {

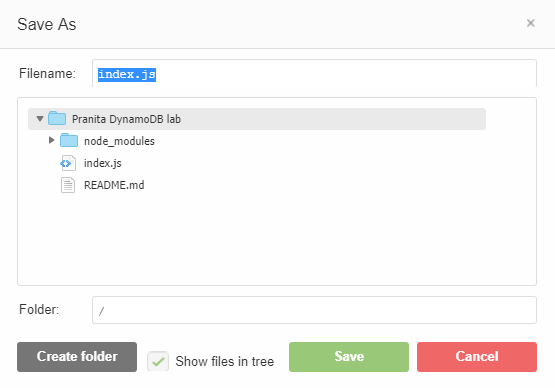
if (err) {

console.log(err, err.stack); // an error occurred

}

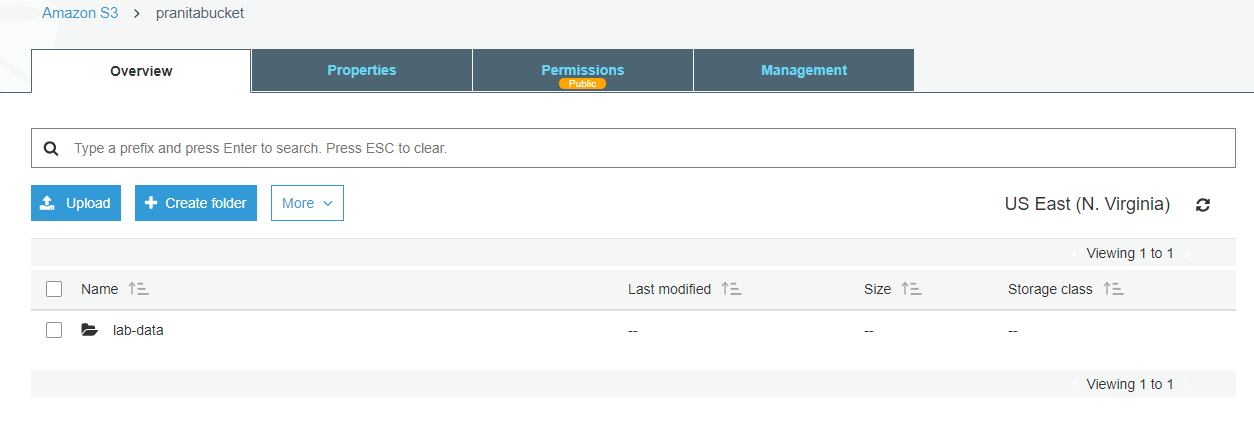
else {

console.log(data.TableNames); // successful response

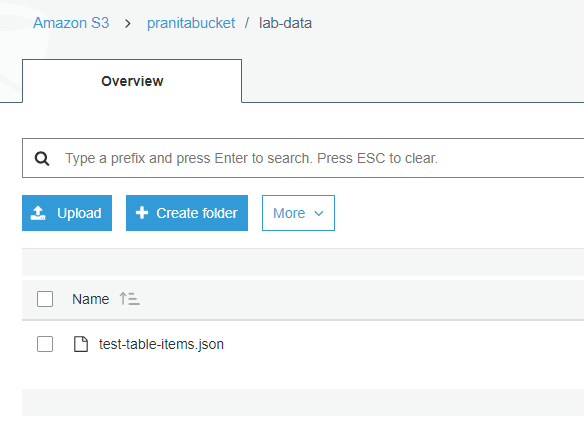
});  
  
  
  
  
  
  
Press **ctrl + s** and save the code as **index.js**  
  
From the bash console at the bottom of the screen of Cloud9 IDE, enter the following commands one after the other:  
  
To install the AWS SDK, hit command – **npm install aws-sdk**  
To run the app – **node index.js**

On running the app, the list of tables will be listed in the console.

Download the JSON file from the following link:-  
<https://github.com/pranitaj01/PranitaDynamoDB/blob/master/test-table-items.json>

On the AWS management console, click **services -> S3 -> create bucket**The bucket and folder can remain private if using Cloud9 IDE. If using an IDE remotely connected to EC2, you will require an EC2 role to access S3.  
Create “lab-data” folder in your bucket.  


Upload the JSON file into this folder. Make sure you give access to this file while uploading



Please refer the following code. You can paste the chunks of code one by one in your IDE and check for the respective results. *Donot forget to change bucket name, DnamoDB table name as per what you create.*  
<https://github.com/pranitaj01/PranitaDynamoDB/blob/master/NodeJS%20dynamodb%20batchWrite%20code.txt>