Serverless Development 101

**Module 05A – The Web Application using Serverless Computing and Dynamo DB**

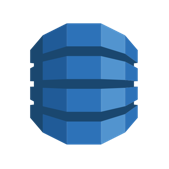
1/10/2019 Developed by Kevin Wang

1/10/2019 Checked by Clark Jason Ngo

1/10/2019 Tested by Tuan Khai and Minh Truong

2/11/2019 Revised by Sam Chung

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Learning Outcomes**

* Learn how to add a DynamoDB table through Serverless framework.
* Learn how to configure the permission to operate the table.
* Learn how to add some data to the table for testing.
* Learn how to deploy a fetching API with Lambda.

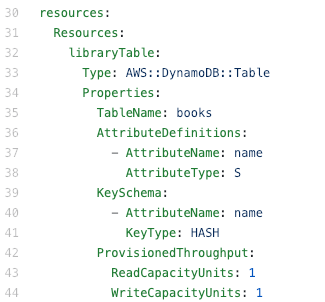
1. Open the VSCode and open the “**myproject**” project folder that we developed a web application using serverless computing in the previous module.

**Configuration – Function, Database, and IAM**

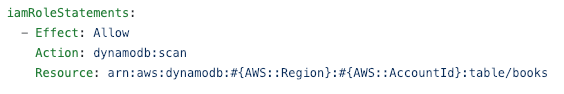
1. Open the “**serverless.yml**” file under the root folder and replace content with <http://bit.ly/2BZVfRo>   
   Note: There are several new things are added
2. A new function is defined

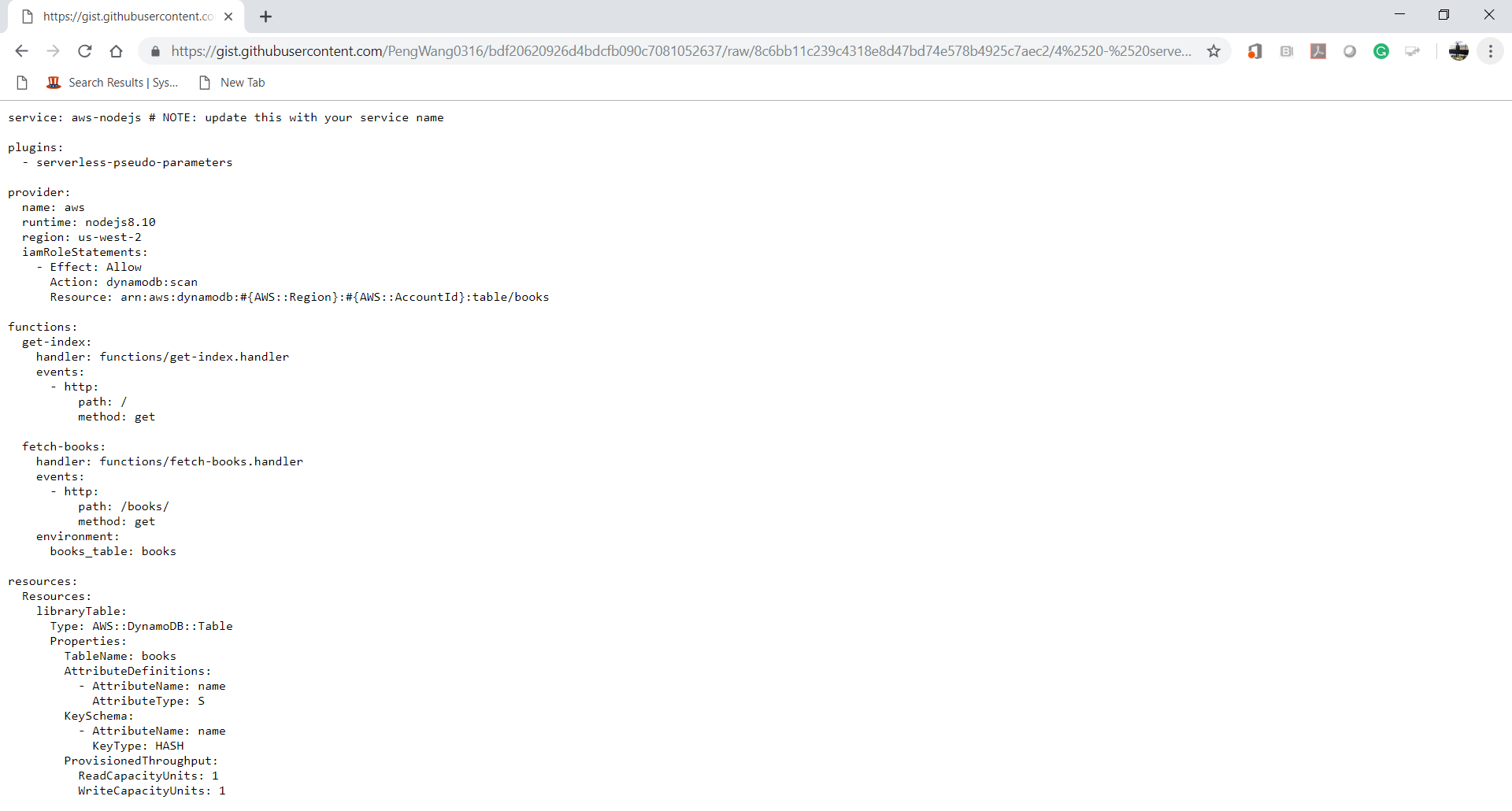


1. A DynamoDB table is defined



1. The scan operation permission is given to all function in this service

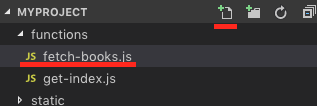
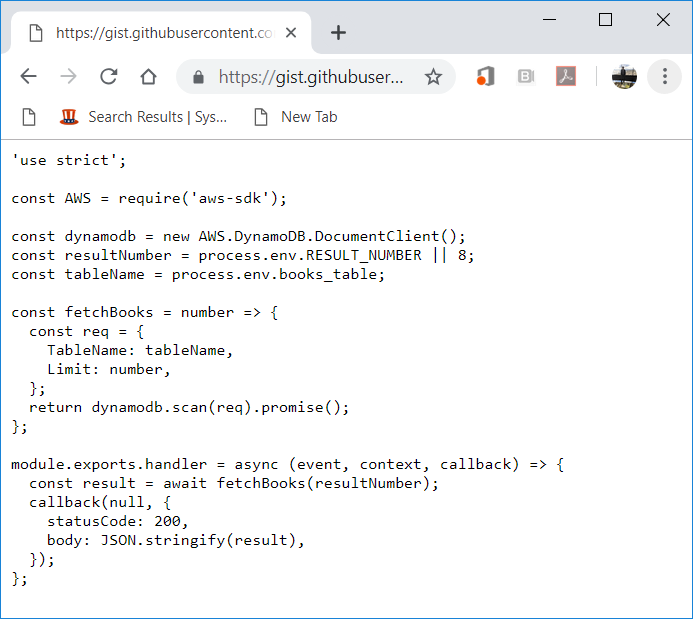


The updated “serverless.yml”

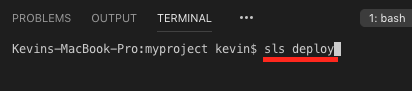
**Lambda Function**

1. Create a “**fetch-books.js**” file under the functions folder and copy the content from <http://bit.ly/2SDqWFn>. Save the file.

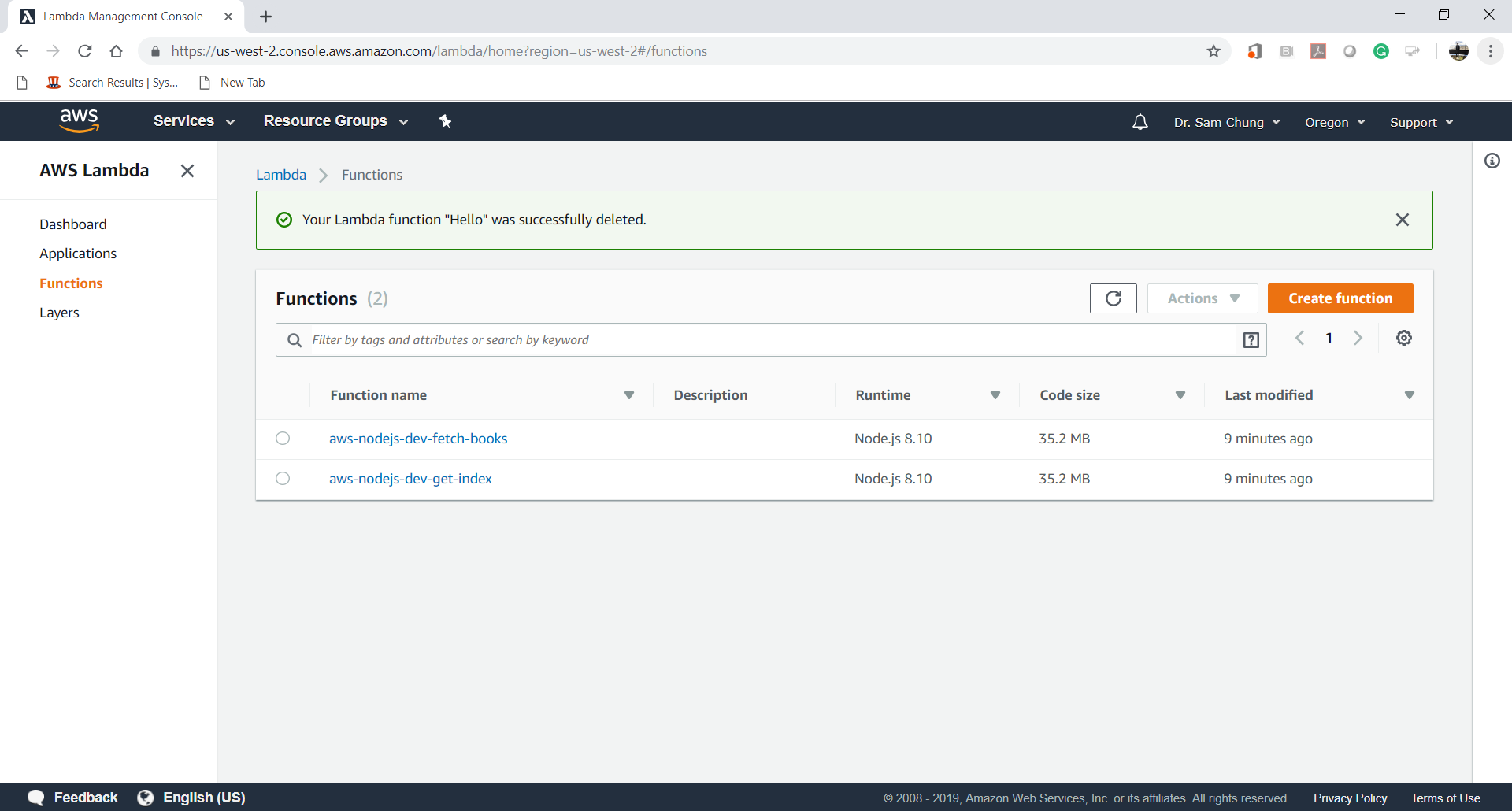
*Common Gotcha: Internal Server Error. Check your file-name. It should be fetch-****books****.js*

1. Open a terminal (press control and ~ keys) in VSCode and type “**sls deploy**” to deploy your new function. (**It may take a while based on your network speed**.)

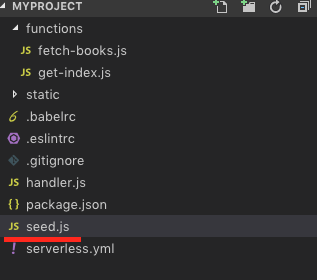
****

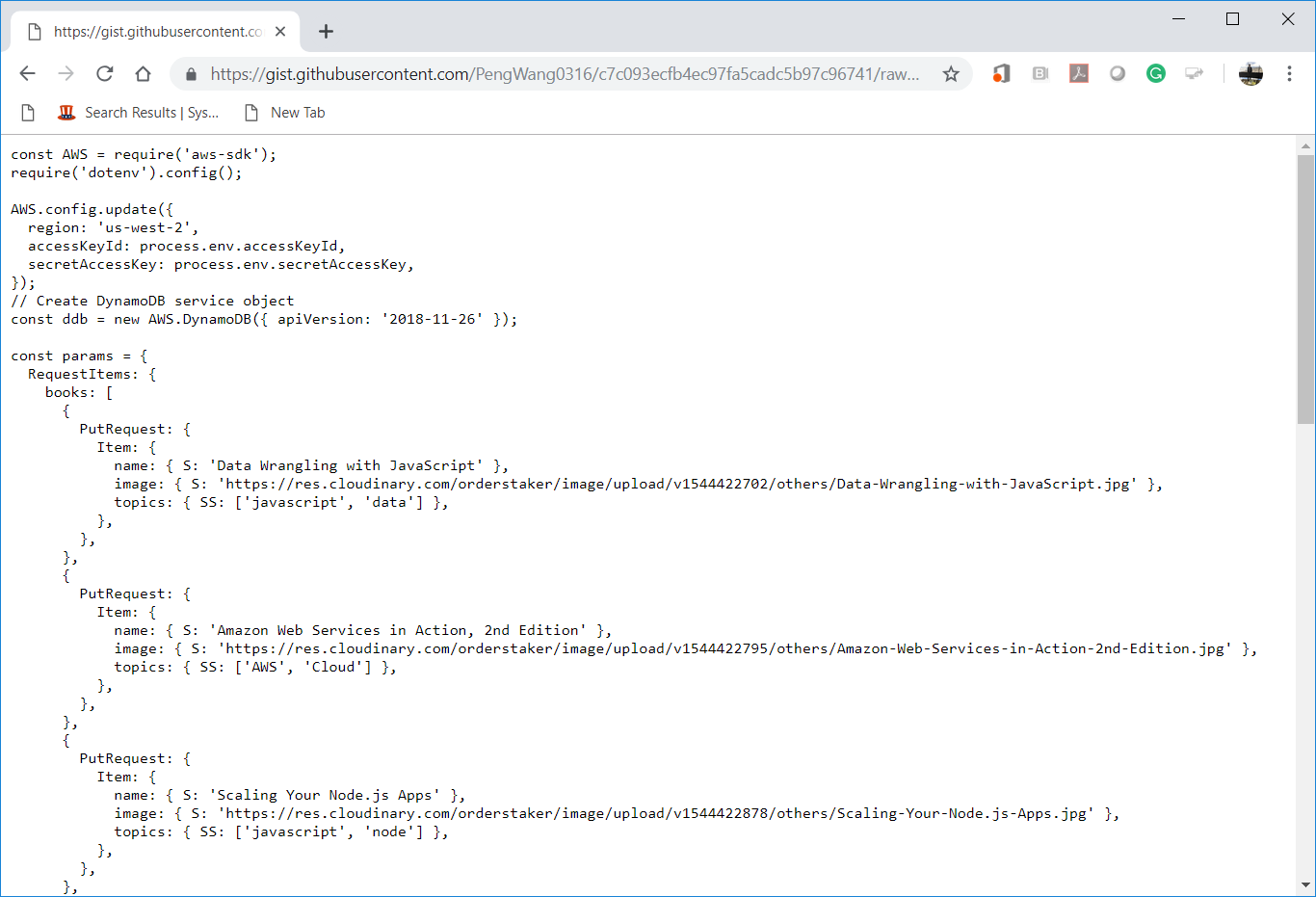
1. Access your “AWS Management Console” and visit your “Lambda” service.  
   Then, visit your Lambda function that you just created.



**Data Load**

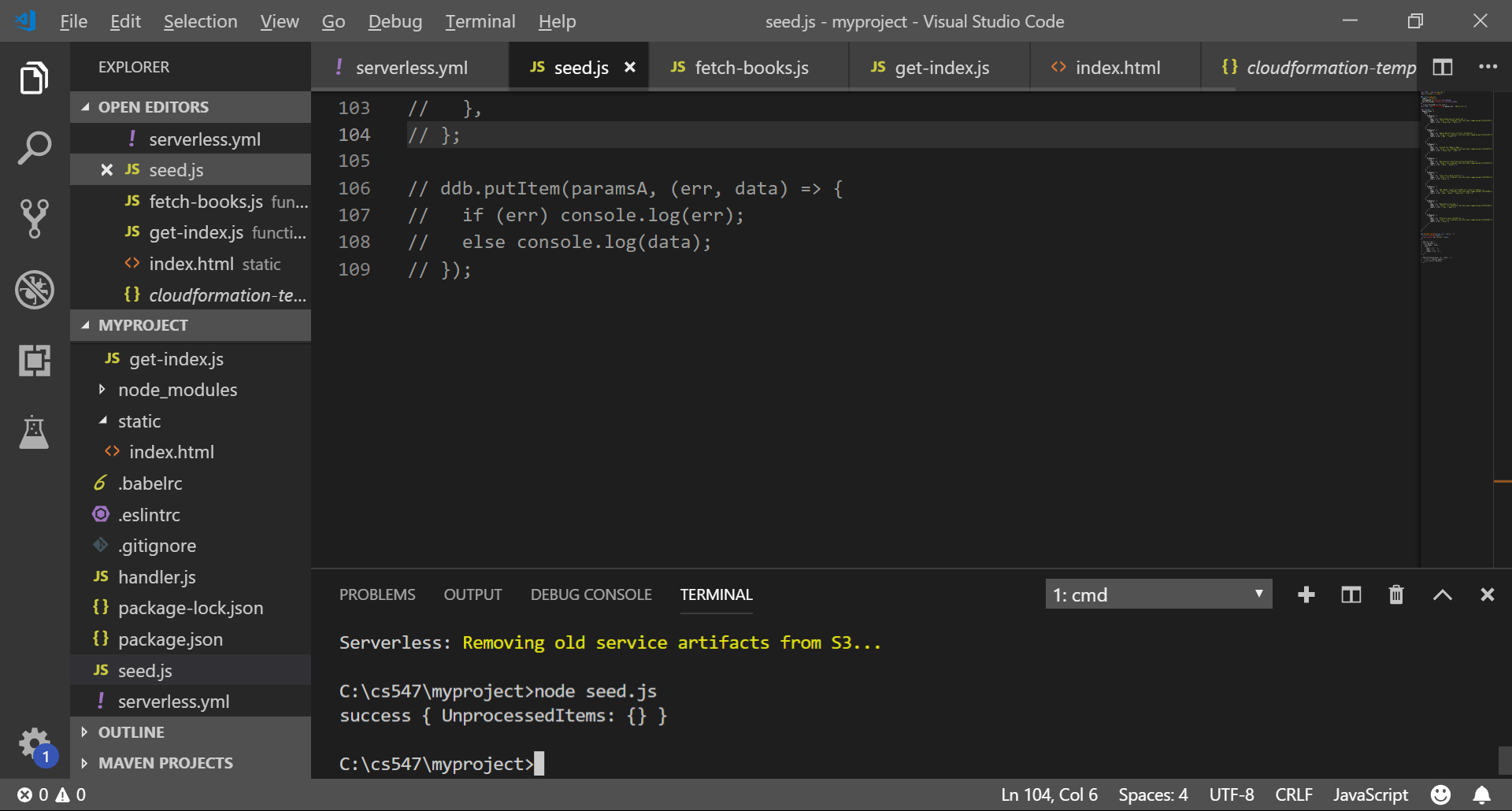
1. Create a “**seed.js**” file under the **root** folder and copy content from <http://bit.ly/2zRGuyn> to insert some fake data for testing. Save the file.  
   Eight (8) books are loaded into your “books” table.



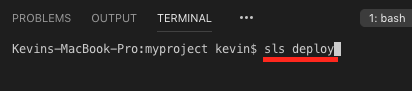


1. Open the terminal in the VSCode and type “**node seed.js**” to insert data

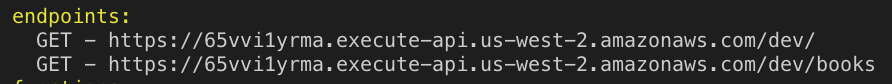
/var/folders/34/3jns11qs1nd6qd5csswy30mc0000gn/T/com.microsoft.Word/Content.MSO/3B36FFDA.tmp



1. Type “**sls deploy**” in the terminal to deploy the change

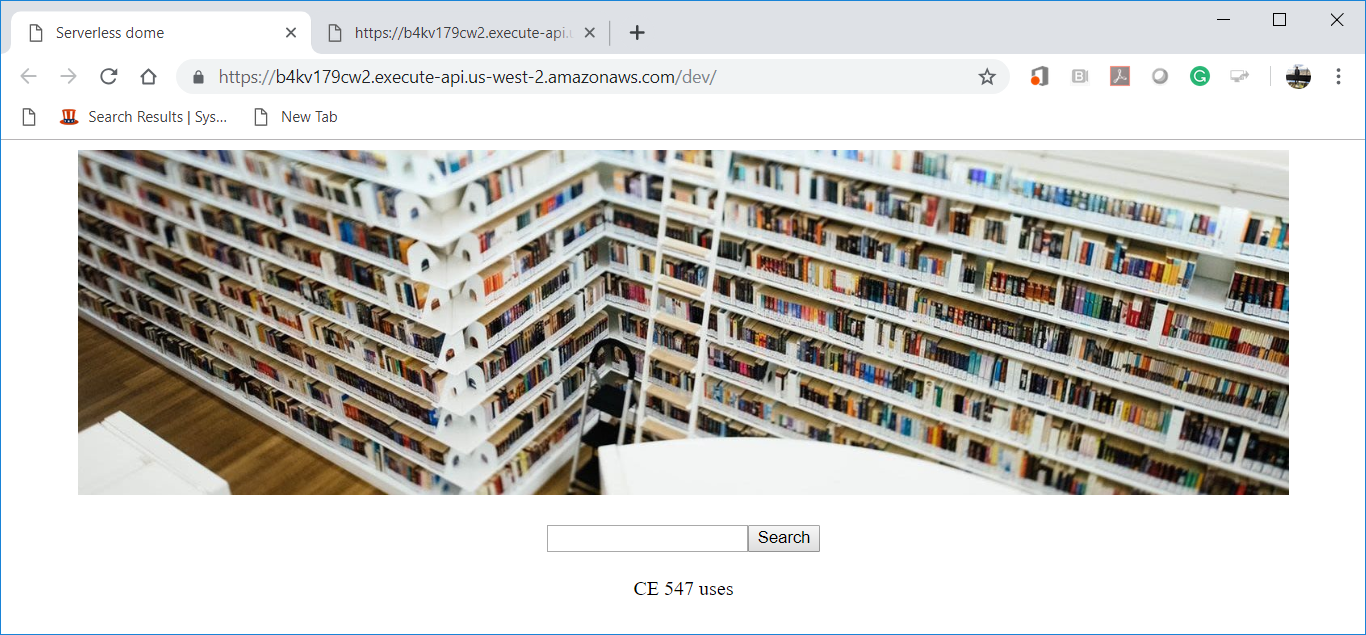
 

1. Copy the URL from “**endpoints**” and put it in the browser to invoke it or press “ctrl+click” to follow link.

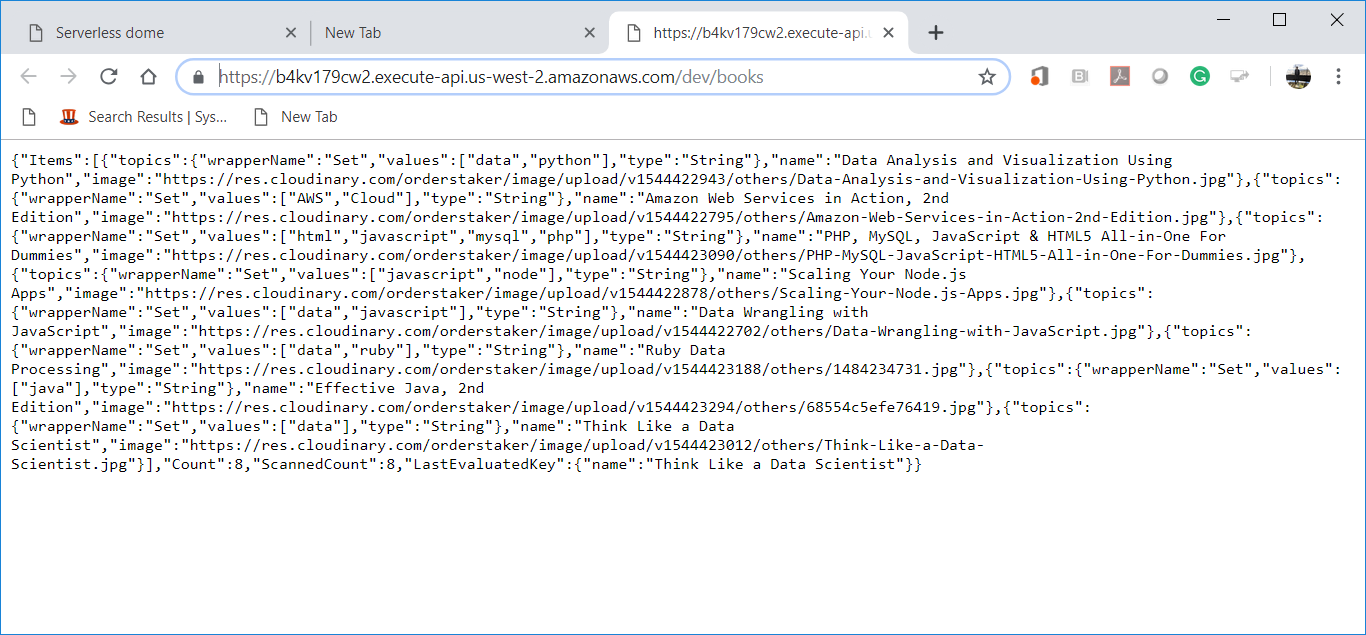
   
Note: you will have different URL endpoints.

Keep this endpoint and use it in the next step

Web Application

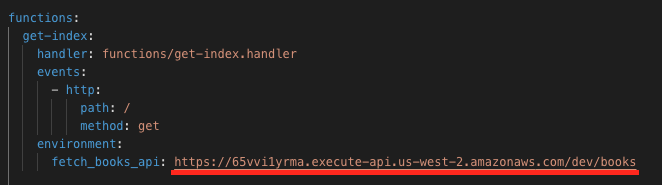


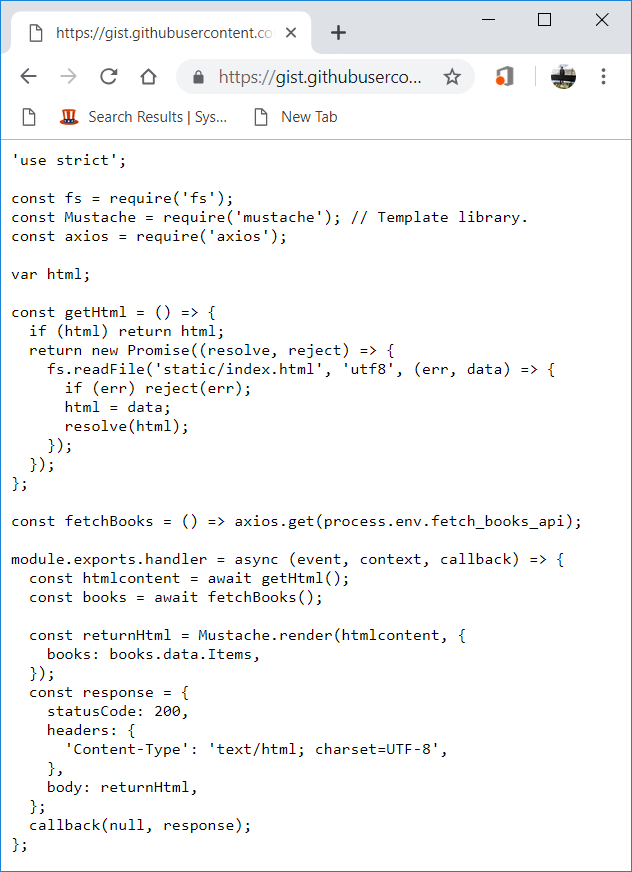
DynamoDB Data

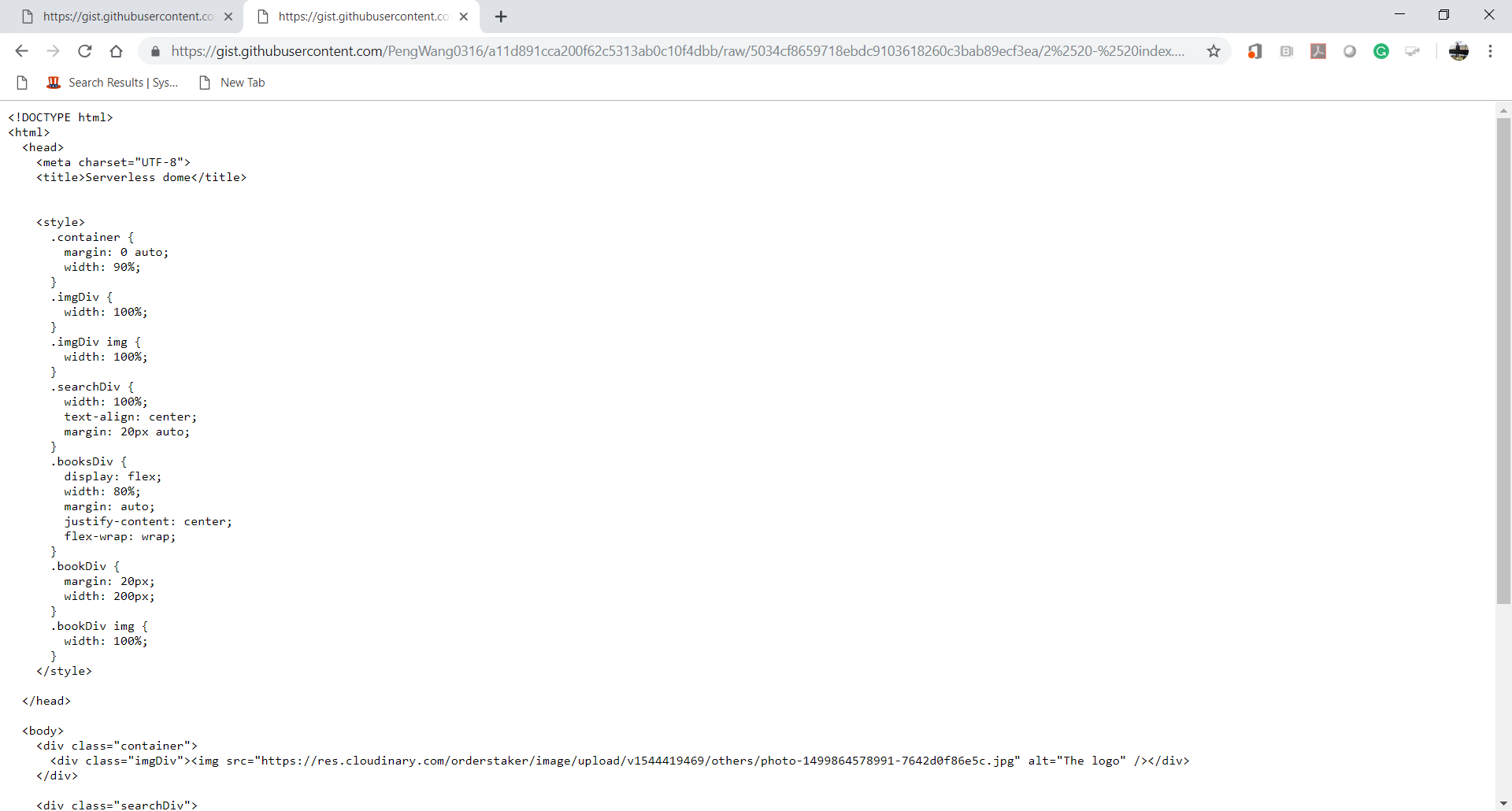


**Updating the Web Application**

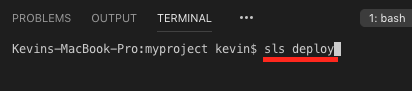
1. Open “**serverless.yml**” file under the root folder and add environment statement under the get-index function (**Replace the endpoint with your own that you get from last step**)

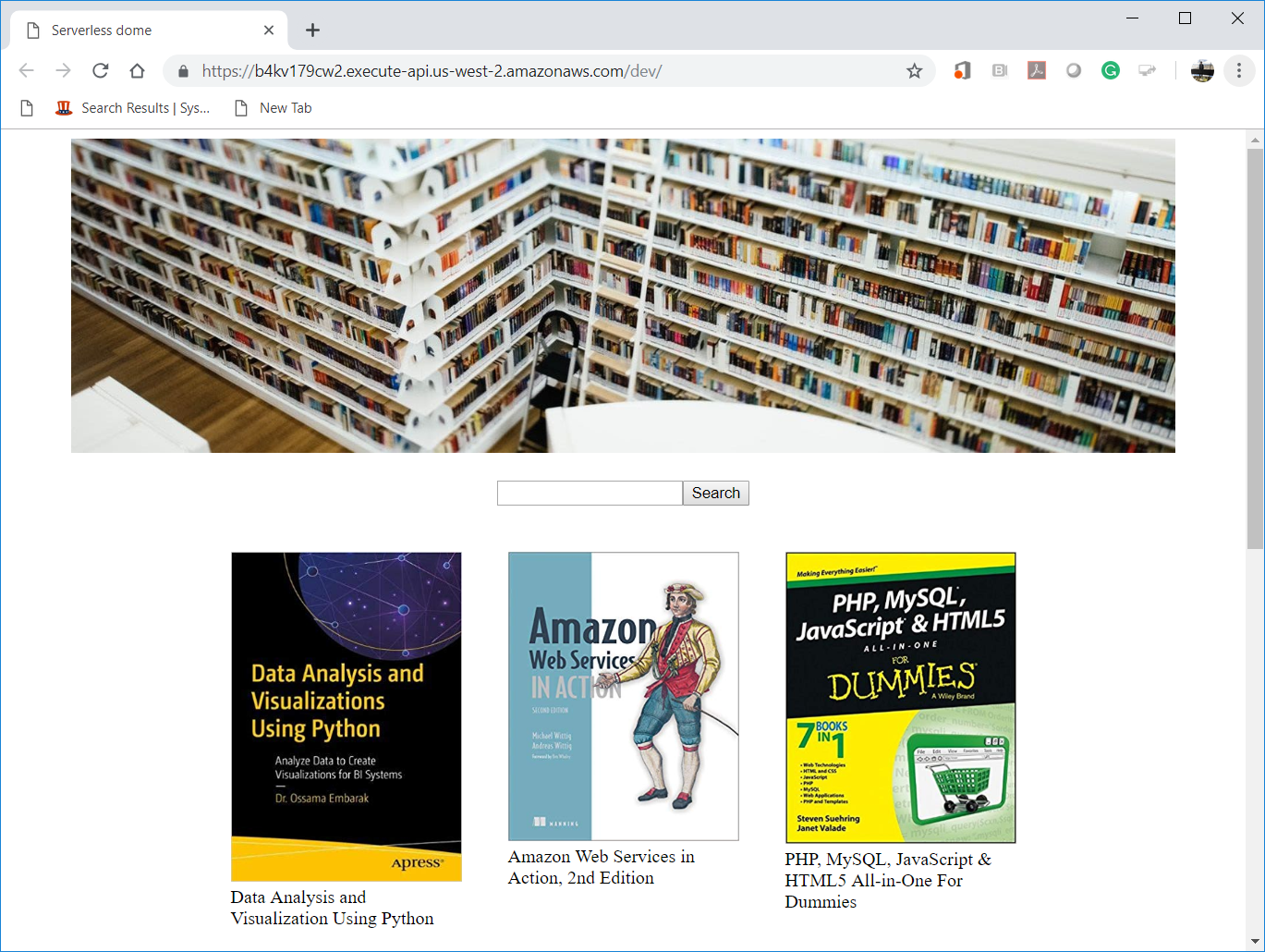
1. Open the “**get-index.js**” function.  
   Update it with <http://bit.ly/2PssDUd>.   
   Find which statements were added or modified.  
   
2. Open the “**index.html**” file and update the content with <http://bit.ly/2C1O2Ac>  
   Find which statements were added or modified.



1. Open the terminal and run the “**sls deploy**” command.

1. Open the “get-index” endpoint in your browser to see the new changes we made



**Questions:**

1. Explain a new function “**fetch-books.js**”?  
   (Answer)  
   We use “**fetch-books.js**” to get the json data from DynamoDB -> books\_table and return our result.
2. What is the DynamoDB?  
   (Answer)  
   DynamoDB is a NoSQL database that is fully managed by AWS
3. What are the pros and cons to use the DynamoDB?  
   (Answer)

**Pros**

* Flexible – it doesn’t have flexible schema when each data item may have a different number of attributes.
* Scalable – no limit on the amount of data to store and automatic data allocation
* Cost effective – allows more than 40 million database operations per month

**Cons**

* Limit on querying(1MB)
* Limit on row size(64KB)
* Deployable only on AWS

1. What is the purpose of the “seed.js” file?  
   Explain the table structure.  
   (Answer)

In this file, we create DynamoDB service object and putting books information it in.

1. What were changed in the “**serverless.yml**” file?  
   (Answer)  
   We added environment link for a fetch books api.

*environment:*

*fetch\_books\_api: https://wfi698quki.execute-api.us-west-2.amazonaws.com/dev/books*

1. What were changed in the “**index.html**” file?  
   (Answer)  
   We added “{{#books}}” to dynamically populate data on the page from JavaScript.
2. What were changed in the “**get-index.js**” file?  
   (Answer)

We added “Mustache” render to get html from the api.

1. Any other service that AWS offers can be used to replace DynamoDB?  
   (Answer)

There are other Database types that AWS is offering such as:

* Amazon Aurora, Amazon RDS, Amazon Redshift – relational
* Amazon ElastiCache – in memory
* Amazon DocumentDB – Document
* Amazon Keyspaces – Wide-Column
* Amazon Neptune – Graph
* Amazon Timestream – Time series
* Amazon Quantum Ledger Database - Ledger