AWS Lambda + OpenSearch

Step 1: Create an OpenSearch Service domain

---

see here for a step-by-step tutorial

1. Go to https://aws.amazon.com and sign in
2. Under **Analytics**, choose **Amazon OpenSearch Service**
3. choose **Create domain**
4. provide a name
5. choose **Standard Create**
6. Templates: **Dev/test**
7. Domain without standby and 1-AZ
8. For Data nodes - General purpose, t3.small-search (include previous generation instance types), 1 node
9. Network - public access
10. Fine-grained access control: enable, and create master user
11. Access policy - Only use fine-grained
12. Create

Copy Domain endpoint (IPv4) - https://search-bdbi-opensearch-dwohisg6gfus332e6xkt475svq.us-east-2.es.amazonaws.com

Step 2: Upload Data

---

from https://docs.aws.amazon.com/opensearch-service/latest/developerguide/search-example.html

1. run `curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-binary @sample-movies.bulk -H 'Content-Type: application/json'

Step 3: Create and deploy Lambda Function

Create the function

1. run `mkdir my-opensearch-function` in the terminal
2. `cd my-opensearch-function`
3. create a file called `opensearch-lambda.py`
4. run the following

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
pip3 install --target ./package requests_aws4auth
```

5. `cd package`
6. `zip -r ../my-deployment-package.zip .`
7. `cd ..`
8. `zip my-deployment-package.zip opensearch-lambda.py`

Deploy the function

1. navigate to the Lambda console: https://console.aws.amazon.com/lambda/home
2. choose **Create a function**
3. use the following fields:
    - function name: `opensearch-function`
    - runtime: Python 3.9
    - architecture: x86_64
4. in code source, choose **Upload from** and choose your .zip file
5. under **Runtime settings** choose **Edit** and rename your handler to `opensearch-lambda.lambda_handler`

Step 4: Create the API in API Gateway

Create the API

1. Navigate to API Gateway console: https://console.aws.amazon.com/apigateway/home
2. Choose **REST API** and **Build**
3. Make sure **New API** is selected
4. configure the following fields:
    - API name: opensearch-api

  - Description: Public API for searching an Amazon OpenSearch Service domain
  - endpoint type: regional
5. click **Create API**

Add GET method

1. Choose **Create method**
2. set **Method type** to **GET**
3. select your Lambda function and turn default timeout on
4. make sure **Use Lambda proxy integration** = true
5. click **Create method**
6. edit the **Method request settings**
   - authorization: none
   - request validator: validate query string parameters and headers
   - API key required: false
7. under **URL query string parameters**, add a query string with the following:
   - name: q
   - required: yes
8. save
9. Click **Deploy API**
10. **Deployment stage** - choose `*New Stage*` and name it `opensearch-api-test`
11. Click **Deploy**
12. **Edit stage details**
    - enable throttling: yes
    - rate: 1000
    - burst: 500
13. save

copy the URL

Step 5: Map the Lambda role

---

1. Navigate to the OpenSearch Dashboards URL
2. log in
3. select **Security, Roles** and `all_access` from main menu
4. choose **Mapped users, Manage mapping**

5. under backend roles, add the ARN of the Lambda role
    - should take form of arn:aws:iam::`123456789123`:role/service-role/`opensearch-lambda-role-1abcdefg`
6. select **Map**

Step 6: Run web application

---

1. unzip `sample-site.zip`
2. under `scripts/search.js` edit the `apigatewayendpoint` variable with your API Gateway endpoint and add a backslash to the end of the path
3. run `index.html` using Live Server in VS code
4. try different queries like `thor`, `house`, etc.

Step 7: Enable Cors

---

may need to enable cors (inspect the console on your website)

1. Enable CORS on the GET resource (Gateway API) - under **Advanced**, set **Access-Control-Allow-Credentials** to true
2. Redeploy your API in API Gateway
3. Delete and re-add your Lambda function trigger
    - trigger: API Gateway
    - API: opensearch-api
    - deployment stage: opensearch-api-test
    - security: open