



Lomba Kompetensi Siswa Sekolah Menengah Kejuruan Tingkat Provinsi Jawa Barat Tahun 2023

Modul 1 - Serverless Architecture

May 7, 2023

Bidang Lomba Cloud Computing

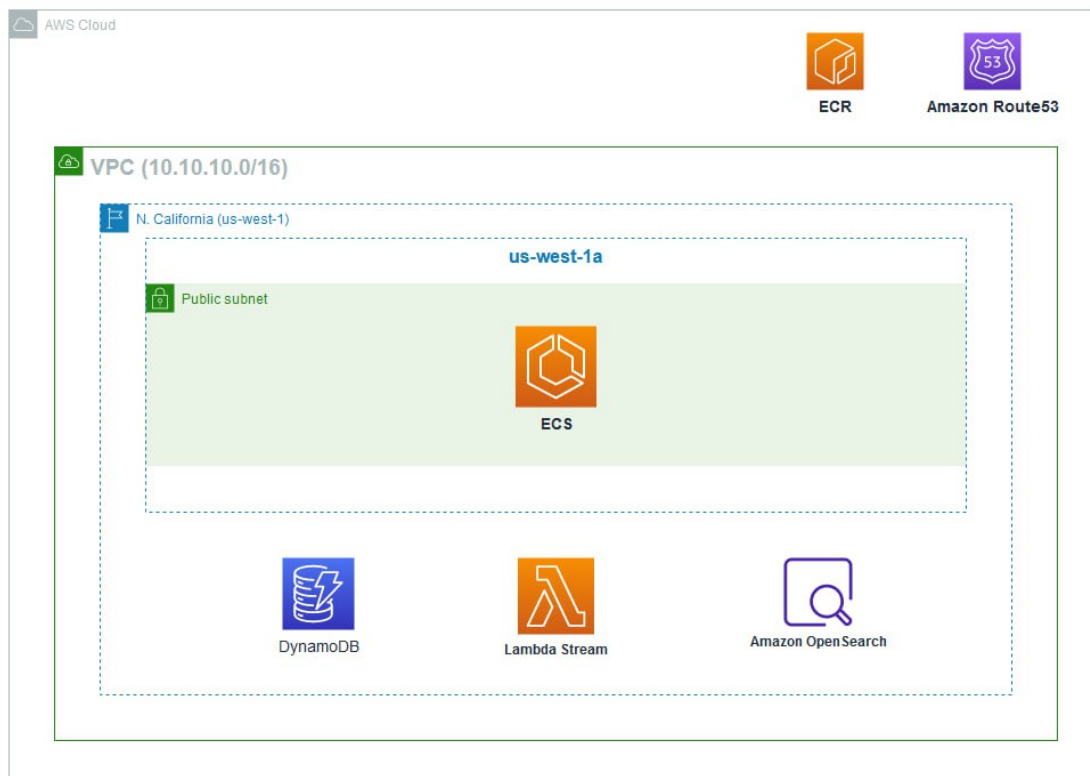
1. Overview

Data search methods have experienced rapid development. This feature has become one of the features that cannot be ignored in the development of websites and other applications. With the development of data storage technology and the large amount of existing data, a reliable search method is needed. Opensearch is one of the latest technologies that is widely used. AWS recently launched the AWS Opensearch Serverless Service which is an option from the existing AWS Opensearch.

2. General Rules

1. Failure to comply with the rules will result in immediate disqualification.
2. You have 3 hours to finish the tasks.
3. You may use AWS Console and AWS CLI to deploy the solutions. You may not use SAM, CloudFormation or CDK.
4. Between and after the event, you may not access your account. Any activity on AWS during this period is not allowed.
5. During the event, multiple login is not permitted.
6. If you have any questions, do not hesitate to ask.

3. Architecture



1. Through the provided website user can search, add, and delete data
2. When searching for data, the website will access the Opensearch Collection and then Opensearch will return data that matches the given keywords.
3. When adding new data the website will store the data to DynamoDB. After successfully adding data, DynamoDb will activate a trigger that will run a Lambda Function which will add the data to Opensearch.
4. When deleting data the website will delete the data in DynamoDB. After successfully deleting data, DynamoDb will activate a trigger that will run a Lambda Function which will remove the data on Opensearch.

4. Information

1. This solution must be deployed in **us-west-1** region. Deploying in another region will result in a major point reduction.

5. Task

Your task is to create solution from the Section 3

1. Create IAM User (if not exist)
2. Prepare a domain listed in Route53 Hosted Zones
3. Create VPC with following configuration
 - Name : vpc-modul1
 - IPV4 CIDR Block : 10.0.0.0/16
 - No IPv6 CIDR block
 - Add tag : Key=LKS-ID, Value=MODUL1-STREAM
4. Create 3 Subnets with following configuration
 - VPC ID : VPC ID from VPC you created
 - Name : subnet-modul-1-1, subnet-modul-1-2, subnet-modul-1-3 accordingly
 - Availability Zone : us-west-1a, us-west-1b, us-west-1c accordingly
 - IPv4 CIDR block : 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24 accordingly
 - Add Tags : Key=LKS-ID, Value=MODUL1-STREAM-1 on subnet-modul-1-1
 - Add Tags : Key=LKS-ID, Value=MODUL1-STREAM-2 on subnet-modul-1-2
 - Add Tags : Key=LKS-ID, Value=MODUL1-STREAM-3 on subnet-modul-1-3
5. Create Security Group with following configuration
 - VPC : VPC you created on previous step
 - Inbound Rules
 - Type : All TCP
 - Source : Custom
 - CIDR Block : 0.0.0.0/0

- Outbound Rules
 - Type : All TCP
 - Source : Custom
 - CIDR Block : 0.0.0.0/0
- 6. Create an Internet Gateway and attach it to VPC you created
- 7. Create a new routing table (if not exists) and add 0.0.0.0/0 on routing table to Internet Gateway
- 8. Make all subnets associate with the routing table you edited
- 9. Create Amazon OpenSearch Serveless Collection with following configuration
 - Step 1
 - Collection Name = **lks-id-collection**
 - Collection type : Search
 - Encryption : Use AWS owned key
 - Access Type : VPC
 - Create VPC Endpoint :
 1. Name : vpc-endpoint-modul-1
 2. VPC : use VPC created in previous step
 3. Subnet : subnet-modul-1-1 created in previous step
 4. Security Group : use default Security Group
 - Add tag : Key=LKS-ID, Value=MODUL1-STREAM
 - Step 2
 - Rule Name : **lks-id-os-rule-1**
 - Select Principal : add IAM User from user you use to log into console
 - Grant Permission : select all
 - Step 3
 - Create as a new data access policy
 - Access policy name : **lks-id-os-policy**
 - Review and Submit
 - Wait until collection created
- 10. Create Lambda Function with following configuration
 - Author from scratch
 - Function name : **message-trigger-opensearch**
 - Runtime : Node.js 16.x
 - Architecture : x86_64
- 11. Clone repo from https://github.com/darkbcx/lkscjabar2023modul1_stream,
 - Follow instruction given
 - Upload generated zip file as code for Lambda created in previous step
- 12. Open your Lambda created in previous step
 - Add tag : Key=LKS-ID, Value=MODUL1-STREAM

13. Open Opensearch Collection created in previous step

- Update data access policy
- Add principal : select role from Lambda created in previous step
- Save changes

14. Open Opensearch Dashboard URL from collection created

- Create index named **message** with following mappings :

```
{
  "id": {
    "type": "text"
  },
  "message": {
    "type": "text"
  },
  "title": {
    "type": "text"
  }
}
```

15. Create DynamoDb table with following configuration

- Table name = **message**
- Partition Key = **id**(String)
- Add tag : Key=LKS-ID, Value=MODUL1-STREAM
- Create a trigger and connect it with the Lambda Function created

16. Open your created DynamoDb table

- On **Export and Stream** tab, create a trigger and connect it with the Lambda Function created

17. Create EC2 instance with following configuration

- AMI : Amazon Linux 2023 AMI
- Architecture : 64-bit (x86)
- Instance Type : t3.small
- Key Pair : use any key pair available or you can create new one
- VPC : use VPC you created
- Subnet : choose one subnet linked to selected VPC
- Security Group : use Security Group you created

18. Take note of Public IP from EC2 instances created, and add it as record for your domain in Route53 Hosted Domain

19. Connect to your EC2 using SSH and do the following

- Install Git, NodeJs and PM2
- Add environment variables as follow :
 - INDEX_NAME = message
 - TABLE_NAME = message
 - BASE_URL = [your domain]
 - OPENSEARCH_URL = [your Opensearch endpoint]

- HOSTNAME: 0.0.0.0
- PORT : 80
- Clone repo from https://github.com/kensasongko/lksccejabar2023modul1_aplikasi
- `cd` into repo folder
- Run `npm run install`
- Run `npm run build`
- Run `pm2 -n my_service start sudo - npm run start`

6. References

- [AWS VPC Documentation](#)
- [AWS VPC Security Groups](#)
- [AWS VPC Internet Gateway](#)
- [AWS Opensearch Documentation](#)
- [AWS Lambda Documentation](#)
- [AWS DynamoDB Documentation](#)
- [AWS EC2 Documentation](#)
- [AWS Route53 Documentation](#)
- [Deploy Lambda Function](#)
- [Installing Git](#)
- [Installing NodeJs](#)
- [Installing PM2](#)
- [Set Environment Variables](#)
- [Configure Subnets](#)