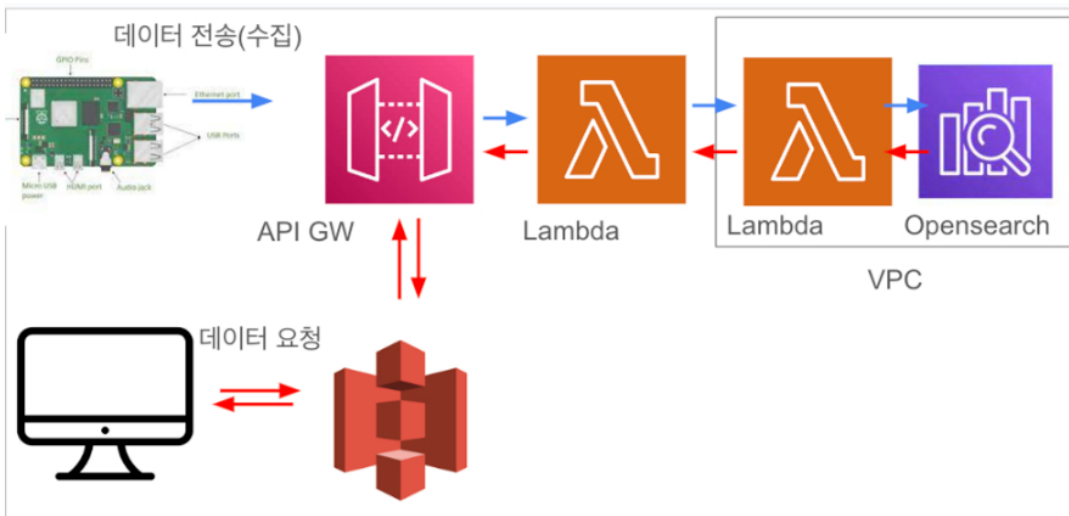


센서플랫폼VPC내용추가

보안강화를 위한 VPC설정 및 코드개선, 결과화면

기존 Opensearch는 vpc로 설정되어 있지않아 보안성이 떨어짐
이를 해결하고자 vpc를 적용하기로하였으며, 그 과정과 결과를 다룬다.

VPC를 통해서는 오직 API GW를 통해서만 람다, 오픈서치를 이용할 수 있도록 구성된다.



- VPC

아래와 같이 구성하였다.

Your VPCs (1/1) [Info](#) [Refresh](#) [Actions](#) [Create VPC](#)

[Name : crc-sensorplatform-vpc](#) [Clear filters](#) [1](#) [Settings](#)

<input checked="" type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6
<input checked="" type="checkbox"/>	crc-sensorplatform-vpc	vpc-0fdd6427e81be1195	Available	172.0.0.0/24	-

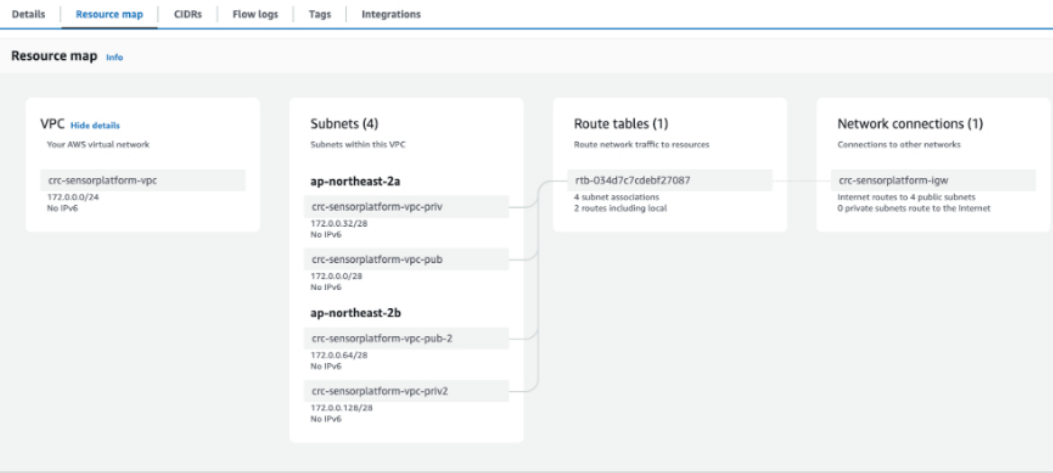
vpc-0fdd6427e81be1195 / crc-sensorplatform-vpc

[Details](#) [Resource map](#) [CIDRs](#) [Flow logs](#) [Tags](#) [Integrations](#)

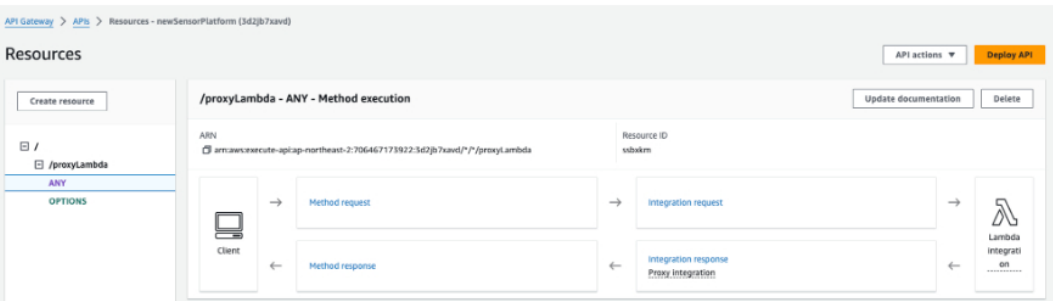
Details

VPC ID vpc-Ofdd6427e81be1195	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-b3a704d8	Main route table rtb-034d7c7cdebf27087	Main network ACL acl-0937b3241ded7c7a0
Default VPC No	IPv4 CIDR 172.0.0.0/24	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 706467173922	

vpc-Ofdd6427e81be1195 / crc-sensorplatform-vpc



- API GW 구성화면이다.



ANY를 통해서 POST와 GET의 요청을 통합실행하는 방법을 택했다.

다음은 해당 GW와 연결된 람다화면이다.

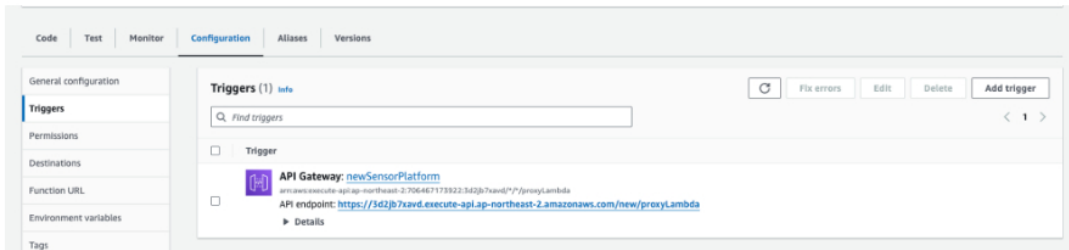
- Lambda (외부)

The Lambda console shows the configuration for the proxyLambda function. The function overview includes a diagram showing the function triggered by an API Gateway. The code source section shows the following code:

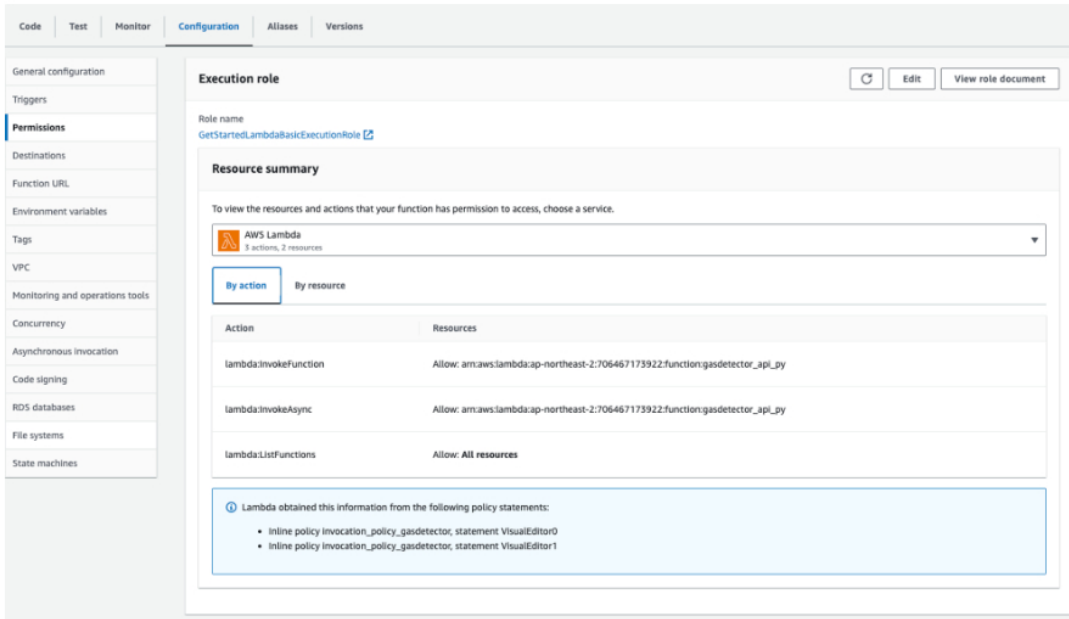
```

1 import boto3, json
2 client = boto3.client('lambda', region_name='ap-northeast-2')
3
4 def invokeFunction(payload):
5     p = json.loads(payload)
6     function_name = p['function_name']
7     response = client.invoke(FunctionName=function_name, Payload=json.dumps(p))
8     return response['Payload'].decode('utf-8')
  
```

다이어그램에서 보는 것과 같이 API GW와 연동되어있음을 확인할 수 있다.



configuration > Triggers 를 통해서도 확인할 수도 있다.



또한 이 람다함수는 오픈서치

람다함수의 코드이며

[<https://github.com/kmucrc/sensorplatform/blob/main/2022/aws/lambda/proxy-lambda-function.py>] 에서도 확인할 수 있다.

```
import boto3, json
client = boto3.client('lambda', region_name="ap-northeast-2")

def invokePrivateLambda(payload):
    r = client.invoke(
        FunctionName='gasdetector_api_py',
        InvocationType='RequestResponse',
        Payload=bytes(payload, encoding='utf-8')
    )
    return r['Payload'].read().decode('utf-8')

def lambda_handler(event, context):
    method = event['httpMethod']

    if method == 'POST':
        body = event['body']
        index = event['headers']['index']
        payload = json.dumps({'body': body, 'method': method, 'index': index})
        response = invokePrivateLambda(payload)

    return {
        'statusCode': 200,
        'headers': {
            'Access-Control-Allow-Headers': 'Content-Type',
```

```

        'Access-Control-Allow-Origin': '*',
        'Access-Control-Allow-Methods': 'OPTIONS,POST,GET'
    },
    'body': response
}

elif method == 'GET':
    index = event["queryStringParameters"]['index']
    query = event["queryStringParameters"]['query']
    payload = json.dumps({'query':query, 'method': "GET", 'index': index })
    response = invokePrivateLambda(payload)

    return {
        'statusCode': 200,
        'headers': {
            'Access-Control-Allow-Headers': 'Content-Type',
            'Access-Control-Allow-Origin': '*',
            'Access-Control-Allow-Methods': 'OPTIONS,POST,GET'
        },
        'body': response
    }

return {
    'statusCode': 400,
    'body': json.dumps('Invalid_Access')
}

```

- Lambda(VPC, 내부)

gasdetector_api_py

Throttle Copy ARN Actions

Function overview Info

Export to Application Composer Download

Diagram Template

gasdetector_api_py

Layers (2)

+ Add trigger + Add destination

Description

Last modified last year

Function ARN

arn:aws:lambda:ap-northeast-2:706467173:922:function:gasdetector_api_py

Function URL Info

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Monitoring and operations tools

Concurrency

Asynchronous invocation

VPC Info Edit

VPC

vpc-0fdd6427e81be1195 (172.0.0.0/24) | crc-sensorplatform-vpc

Subnets

- Allow IPv6 traffic = false
- subnet-00ad5d06c9b6b7c80 (172.0.0.32/28) | ap-northeast-2a, crc-sensorplatform-vpc-priv

Security groups

- sg-07bb9915e04e0fcf0 (crc-sensorplatform-sg)

Inbound rules Outbound rules

Security group ID	Protocol	Ports	Source
sg-07bb9915e04e0fcf0	All	All	sg-0062ae26c6c9e92ec

오픈서치서비스와 연결되는 람다함수는 위와 같이 VPC설정을 한다.

코드는 아래와 같으며, 링크는

(https://github.com/kmucrc/sensorplatform/blob/main/2022/aws/lambda/gasdetector_api_py.py)이다.

```
from pytz import timezone
import constants
from urllib import parse

client = boto3.client(
    'iot-data',
    region_name = 'ap-northeast-2',
    endpoint_url = 'https://a26pnpi3qj130t-ats.iot.ap-northeast-2.amazonaws.com'

def publish_to_iot_core(data):
    now = datetime.datetime.now(timezone('Asia/Seoul'))
    nowDatetime = now.strftime('%Y-%m-%d %H:%M:%S')
    data['date'] = nowDatetime;

    return data;

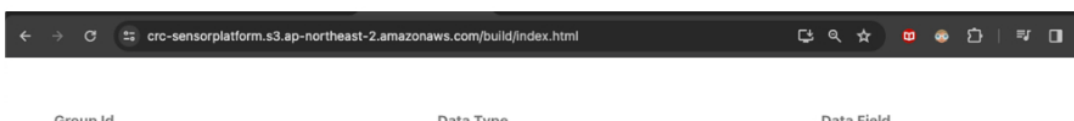
def lambda_handler(event, context):
    print
    headers = {
        # 'authorization': 'Basic a211Y3JjZGV2X29z0iFLbXVjcmMyMDIy',
        'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
        'Content-Type': 'application/json'
    };
    auth = HTTPBasicAuth('오픈서치아이디', '오픈서치비밀번호');

    if event['method'] == "GET":
        url = 'https://vpc-crc-sensorplatform-os-ncsx6wiaesylldgi3q462uvmaeq.ap-northeast-2.amazonaws.com/build/index.html'
        r = requests.get(url, headers=headers, auth=auth, data=event['query'])
        result = r.text

    if event['method'] == "POST":
        url = 'https://vpc-crc-sensorplatform-os-ncsx6wiaesylldgi3q462uvmaeq.ap-northeast-2.amazonaws.com/build/index.html'
        data = json.loads(event["body"]);
        publish_to_iot_core(data);
        result = requests.post(url = url, data = json.dumps(data), headers = headers)

    return {
        'statusCode': 200,
        'headers': {
            'Access-Control-Allow-Headers': 'Content-Type',
            'Access-Control-Allow-Origin': '*',
            'Access-Control-Allow-Methods': 'OPTIONS,POST,GET'
        },
        'body': result
    }
```

데이터요청 결과화면



Group No

1

Device Type

oximetry

Device Name

HbA1c(당화혈색소)

Created Time

Start Date2023-11-01

Start Time03:01 AM

End Date2024-02-29

End Time03:01 AM

Search

