

Lambda, Cloudwatch를 활용하여 ec2 stop/start 스케줄 설정하기

Lambda란 ?

서버리스 컴퓨팅 서비스로, 애플리케이션을 실행하기 위해 별도의 서버 셋업 없이 곧바로 코드를 실행해주는 서비스이다.

초 단위로 비용을 계산하는 ec2와는 별개로 Lambda는 1ms 당 요금을 계산해 정확히 사용한 만큼만 비용이 발생한다.

1. Lambda에 붙일 IAM 정책 및 역할 생성

**정책 및 역할을 연결하지 않고 Lambda를 바로 생성했을 경우에는 정책 및 역할이 자동 생성된다.

[Policy]

정책 이름 : Lambda_policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

[Role]

역할 이름 : Lambda_role

생성한 policy(Lambda_policy) 추가

2. Lambda 생성

Author from scratch 선택

Lambda Function 이름 : StartEC2Instance / StopEC2Instance

Runtime : Python3.8

Permissions - Use an existing role - 생성해두었던 role 선택

[Lambda Code]

```
import boto3
region = 'ap-northeast-2'
instances = []
ec2_r = boto3.resource('ec2')
ec2 = boto3.client('ec2', region_name=region)
//
for instance in ec2_r.instances.all():
    instances.append(instance.id)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

태그를 설정하여 특정 ec2에만 스크립트 적용될 수 있도록 할수도 있습니다.

```
import boto3
region = 'ap-northeast-2'
instances = []
ec2_r = boto3.resource('ec2')
ec2 = boto3.client('ec2', region_name=region)

for instance in ec2_r.instances.all():
    for tag in instance.tags:
        if tag['Key'] == 'auto-schedule':
            if tag['Value'] == 'auto':
                instances.append(instance.id)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('start your instances: ' + str(instances))
```

정상적으로 스크립트가 구동되는지 테스트를 해봅시다.

Lambda - Deploy - Test

구동이 제대로 안된다면,

1) IAM Policy, Lambda code 확인

2) Lambda-General configuration - Timeout 을 3(default) -> 15로 변경.

3) 지정한 region에 있는 ec2의 Instance state가 1EA라도 terminate 되어있는 상태라면 스크립트가 구동되지 않음.

3. CloudWatch 스케줄링 등록

Event - Rules - Go to Amazon EventBridge

step1)

규칙 이름 : StartEC2Instance

Event bus : default

Rule type : Schedule

step2)

GMT 기준 cron 표현식으로 원하는 시간 지정

step3)

target types : AWS service

Select a target : Lambda function

Function : Lambda Function 선택 (StartEC2Instance)

++ 타겟 추가 가능

step4)

태그 설정

(선택) 4. Lambda code 성공/실패 메일 알람

[SNS]

topic 생성

subscription 생성

Protocol : Email

Endpoint : 수신받을 이메일 지정

[Lambda]

Configuration - Destinations - Add destination

source : Asynchronous invocation

Condition : On Failure / On Success

Destination Type : SNS topic

Destination : 생성한 topic 선택