

Lambda를 활용한 EC2 자동화

EC2 automation using Lambda

황인성
전예한
임지훈



연세 IT미래교육원
YONSAI IT EDUCATION CENTER

AGENDA

01.주제 선정 이유

02.팀 구성 과 스케줄

03.EC2 자동화 개념

04.EC2 자동화 소개

05.자동화된 EC2 보안 사례 연구

06.결과 소회 및 자체 평가

The background features decorative geometric patterns in the corners, consisting of thin blue lines, dots, and circles. In the top-left, there are several parallel lines and a cluster of dots. In the top-right, a circle with a dot inside is connected to a line. In the bottom-left, there are more parallel lines and a circle with a dot. In the bottom-right, there are complex line patterns and a circle with a dot.

01

주제 선정 이유

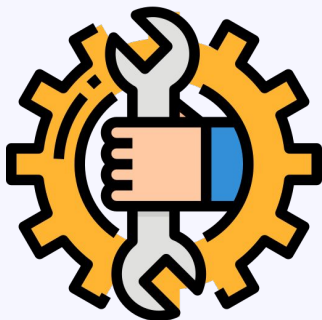
Reason for selecting topic

01.주제 선정 이유



1.보안위협 증가

보안위협은 시간이
지날수록 점점 증가하는
추세



2.수동 관리의 한계

수동으로 관리 할 경우
휴먼에러 발생 가능성 증가
수많은 인스턴스가 존재 할
경우 시간적 한계 발생



3.자동화의 장점

빠르고 효율적으로
일괄적인 보안정책을
적용 가능

02

팀 구성과 스케줄

Team composition and schedule

팀 구성 및 역할

황 인 성

【조 장】

Ec2 자동화 코드 구현
각 팀원의 업무 및 진행 상황 관리
발표준비 및 스케줄링
팀 회의 진행 및 의사결정 촉진

전 예 한

주제 관련 자료 검색 및 수집
자료 검증 및 분석
주요 내용 정리
PPT제작 보조

임 지 훈

태그, 디도스 공격 코드 구현
프로젝트 자료 관리
PPT 제작
최종 결과물 정리

Schedule

[1주차]

- AWS** 리소스 생성 및 자료수집
(계정생성, **ec2**, **s3**, **IAM** 등)
- 팀 구성 및 프로젝트 주제수립

[2주차]

- AWS** 리소스 작업과 자료수집

[3주차]

- boto3 python** 활용 코드 생성

[4주차]

- AWS boto3 python** 활용 코드
생성 및 적용

[5주차]

- 자료 조사 및 코드 구현
- ppt** 제작 및 피드백

[6주차]

- ppt** 내용 보완 및 발표 스크립트
- 발표 예행 연습 및 피드백

03

EC2 자동화 개념

Concept summary of
EC2 Automation

자동화 개념



AWS 리소스와 애플리케이션을 수동으로 관리하는 대신,
자동화 도구를 사용하여 자동으로 관리하는 것을 의미
자동화는 다음과 같은 장점을 존재

시간과 비용 절약: 수동 작업을 자동화하면

시간과 비용을 절약가능

오류 감소: 수동 작업은 오류를 유발할 가능성이 높습니다.

자동화하면 오류를 줄임

효율성 향상: 자동화는 작업을 더 효율적으로

수행할 수 있도록 도와줌

보안 강화: 일관된 보안 정책을 자동으로 적용하여 보안 위험
감소시킴

LAMBDA란?



서버리스 컴퓨팅 서비스

AWS가 실행을 위한 인프라를
완전히 관리 하기 때문에 서버를

*프로비저닝 하거나 관리할 필요가 없음

이벤트 기반

Lambda 함수는 다양한 AWS 서비스

(예: Amazon S3, Amazon SNS, Amazon CloudWatch 등)에서
발생하는 이벤트에 의해 자동으로 트리거

*프로비저닝: 시스템에 필요한 데이터, 리소스, 서비스를 준비/배포하는 과정

LAMBDA의 작동과정



add_keypair

▼ 함수 개요 정보

다이어그램

템플릿



add_keypair



Layers

(0)



SNS

+ 트리거 추가

Lambda 함수에 트리거를 연결

04

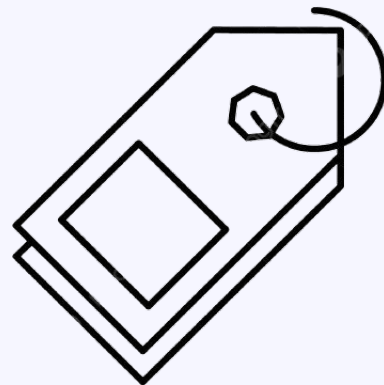
EC2 자동화 소개

Introducing EC2 Automation

EC2 자동화 소개



1. 출퇴근 시간에
맞춰 자동화된
인스턴스 시작과 종료



2. 태그 변환시
자동화된 보안정책 변경

1. 자동화된 인스턴스 시작과 종료



출퇴근 시간에 자동으로 EC2 ON/OFF 설정



비용 절감 달성

1. 자동화된 인스턴스 시작과 종료

```
filters = [{
```

```
if len(RunningInstances) > 0:  
    # perform the startup  
    AutoStarting = ec2.instances.filter(InstanceIds=RunningInstances).start()  
    print("AutoStarting")  
else:  
    print("Nothing to see here")
```

```
}
```

```
if len(RunningInstances) > 0:  
    # perform the startup  
    AutoStarting = ec2.instances.filter(InstanceIds=RunningInstances).start()  
    print("AutoStarting")  
else:  
    print("Nothing to see here")
```

1. 자동화된 인스턴스 시작과 종료

```
def lambda_handler(event, context):
    # All running EC2 instances.
    filters = [
        {
            'Name': 'tag:AutoStop',
            'Values': ['True']
        },
        {
            'Name': 'instance-state-name',
            'Values': ['running']
        }
    ]

    # Filter the instances which are stopped
    instances = ec2.instances.filter(Filters=filters)
    # Get all id of running EC2 Instances
    RunningInstances = [instance.id for instance in instances]
    # Print the instances for logging purposes
    # Print RunningInstances
    if len(RunningInstances) > 0:
        # Perform Shutdown EC2 Instances
        shuttingDown = ec2.instances.filter(InstanceIds=RunningInstances).stop()
        print(shuttingDown)
    else:
        print("Nothing to see here")
```


1. 자동화된 인스턴스 시작과 종료

EC2_Auto_Start

▼ 함수 개요 정보

다이어그램

템플릿



+ 트리거 추가



EventBridge (CloudWatch Events)

aws asynchronous schedule management-tools

Rule

Pick an existing rule, or create a new one.

☒ Create a new rule

☐ Existing rules

Rule name

Enter a name to uniquely identify your rule.

EC2AutoStart

Rule description

Provide an optional description for your rule.

평일 오후 12시 35분에 인스턴스를 시작합니다.

Rule type

Trigger your target based on an event pattern, or based on an automated schedule.

☐ Event pattern

☒ Schedule expression

Schedule expression

Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.

cron(35 03 ? * MON-FRI *)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

1. 자동화된 인스턴스 시작과 종료

EC2_Auto_Stop

▼ 함수 개요 정보

다이어그램

템플릿



EC2_Auto_Stop



Layers

(0)

+ 트리거 추가



EventBridge (CloudWatch Events)

aws asynchronous schedule management-tools

Rule

Pick an existing rule, or create a new one.

☒ Create a new rule

☐ Existing rules

Rule name

Enter a name to uniquely identify your rule.

EC2AutoStop

Rule description

Provide an optional description for your rule.

정일 오후 12시 39분에 인스턴스를 종료합니다.

Rule type

Trigger your target based on an event pattern, or based on an automated schedule.

☐ Event pattern

☒ Schedule expression

Schedule expression

Self-trigger your target on an automated schedule using [Cron](#) or [rate expressions](#). Cron expressions are in UTC.

cron(39 03 ? * MON-FRI *)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

1. 자동화된 인스턴스 시작과 종료

인스턴스 (1/3) 정보

Q Instance을 속성 또는 (case-sensitive) 태그로 찾기

<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형
<input type="checkbox"/>		i-043d57cac1ad358e0	⊖ 종료됨	t2.micro
<input checked="" type="checkbox"/>	Auto_Test	i-04c8558941e526a16	✔ 실행 중	t2.micro
<input type="checkbox"/>	Start_and_Stop	i-04543fda85df77014	✔ 실행 중	t2.micro

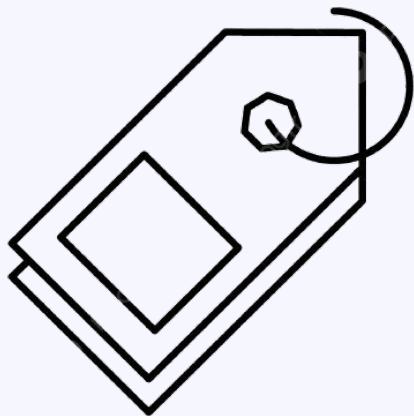
인스턴스 (3) 정보

Q Instance을 속성 또는 (case-sensitive) 태그로 찾기

<input type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형
<input type="checkbox"/>		i-043d57cac1ad358e0	⊖ 종료됨	t2.micro
<input type="checkbox"/>	Auto_Test	i-04c8558941e526a16	⊖ 중지됨	t2.micro
<input type="checkbox"/>	Start_and_Stop	i-04543fda85df77014	⊖ 중지됨	t2.micro

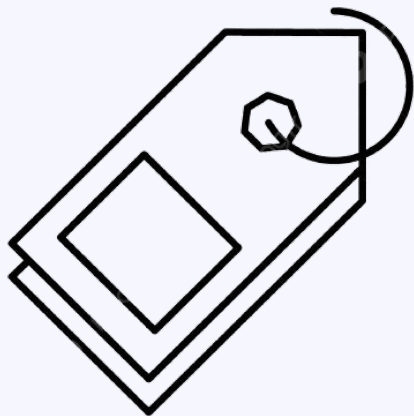


2.태그 변경시 보안정책 자동 변경



- 리소스에 태그를 지정하고 이를 기반으로 보안 정책을 자동으로 변경하는 기능
- 리소스를 "개발", "운영" 및 "테스트" 환경등으로 분류하여 관리하는 기능

2.태그 변경시 보안정책 자동 변경



- TEST환경에서는 다양한 포트를 개방하고 진행하는 경우가 많음
- OPERATION태그로 작성시 보안 정책을 자동으로 변경하여 보안 안전성 확보

2. 태그 변경시 보안정책 자동 변경

```
import boto3
import logging

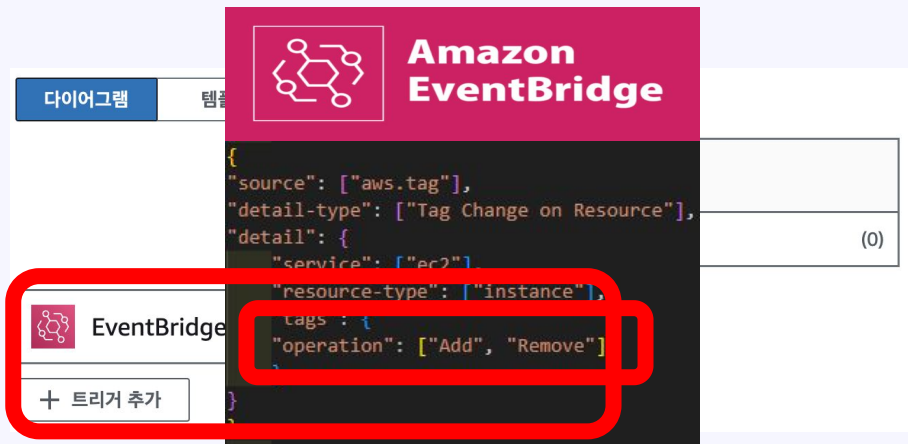
def lambda_handler(event, context):
    ec2_client = boto3.client('ec2')

    security_group_id = 'sg-0e3ad09268cacaac3'

    # 불필요한 포트를 제거하는 규칙
    ec2_client.revoke_security_group_ingress(
        GroupId=security_group_id,
        IpPermissions=[
            {'IpProtocol': 'tcp',
             'FromPort': 23,
             'ToPort': 23,
             'IpRanges': [{'CidrIp': '0.0.0.0/0'}]}
        ]
    )

    logging.getLogger().setLevel(logging.INFO)
    logging.info("Lambda function executed successfully.")
```

2. 태그 변경시 보안정책 자동 변경



이벤트 브릿지에 **operation(실무환경)**
태그가 **Add**됐을 때 이벤트 브릿지를 생성하는
JSON을 작성한 후에

2. 태그 변경시 보안정책 자동 변경

태그 관리 정보

태그는 AWS 리소스에 할당하는 사용자 지정 레이블입니다. 태그를 사용하여 인스턴스를 정리하고 식별할 수 있습니다.

키	값 - 선택 사항	
<input type="text" value="Name"/>	<input type="text" value="my_ec2"/>	<input type="button" value="제거"/>
<input type="text" value="operation"/>	<input type="text" value="Add"/>	<input type="button" value="제거"/>
<input type="button" value="새로운 태그 추가"/>		

최대 48개의 태그를 더 추가할 수 있습니다.

테스트 환경이 종료되었다고 가정하고
Operation 으로 태그를 변경하면

Operation 환경에서 불필요한
23번 telnet port가
Lambda에 의해 폐쇄

▼ 인바운드 규칙

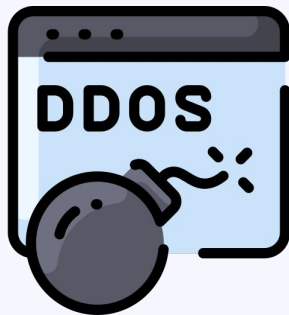
Q 필터 규칙			
이름	보안 그룹 규칙 ID	포트 범위	프로토콜
-	sgr-08086a66155a3db24	443	TCP
-	sgr-09453fc1a386b2998	1433	TCP
-	sgr-06beda6181b1490fd	22	TCP
-	sgr-05c38ea53a3ee3da6	80	TCP
-	sgr-05c38ea53a3ee3da6	80	TCP

05

자동화된 EC2 보안 사례 연구

Case study of EC2 security
automation

자동화된 EC2 보안 사례 연구



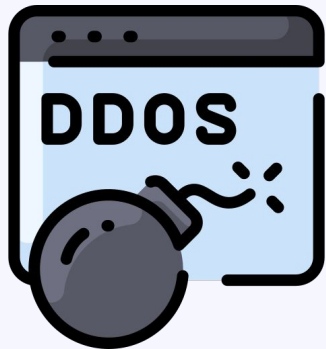
폭증하고 있는 디도스 공격과 피해를
빠르게 인지하고
피해를 최소화해야 할 필요성 증가

2021~2023년, 디도스 최대 공격량 매년 100% 이상 폭발적 증가해

이들간 유튜브·아프리카TV 인터넷방송 디도스 의심 공격 잇따라

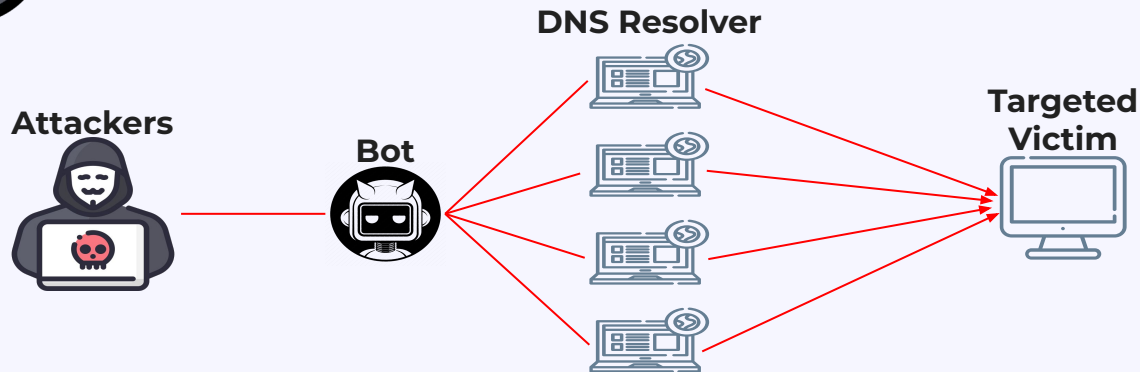
국내 웹사이트 무차별 공격 중인 중국 해커, 1만개 이상 IP 주소 텔레그램 공개

자동화된 EC2 보안 사례 연구



DDOS?

(분산 서비스 거부 공격, **Distributed Denial of Service**)
수많은 컴퓨터(좀비, 봇)를 이용하여
대량의 트래픽을 일으키고
정상적인 사용자의 서비스 이용을 방해하는 공격 방법



자동화된 EC2 보안 사례 연구

● ○ ○ ○ ○

경보 생성

BOTO3를 활용해
CloudWatch경보
생성

● ● ● ● ○

Lambda 실행

솔루션 역할의
람다 함수 실행

● ● ○ ○ ○

트리거 생성

AWS SNS를 활용해
람다의 트리거 역할과
보안관리자에게 매일
발송

● ● ● ● ●

Solution1

AutoScaling 그룹
생성 및 실행
->트래픽 분산

● ● ● ○ ○

DDOS공격 시연

명령어를 활용해
DDOS공격 시연

● ● ● ● ●

Solution2

2차 경보 발생 시
긴급상황용 인스턴스로
Route Redirecting

자동화된 EC2 보안 사례 연구

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cloud Security with one team</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>Welcome to CloudSecurity group 1</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
  <section class="hero">
    <h2>클라우드 보안은 1조와 함께</h2>
    <p>Discover the creativity with Yehan</p>
  </section>
  
    <p>&copy; 2024 web design</p>
  </footer>
</body>
</html>
```

EC2
보안

```
nav {
  background: #333;
}

nav ul {
  list-style-type: none;
  padding: 0;
}

nav ul li {
  display: inline;
  margin: 0 15px;
}

nav ul li a {
  color: white;
  text-decoration: none;
}

.hero {
  background: #eee;
  margin: 20px 0;
}

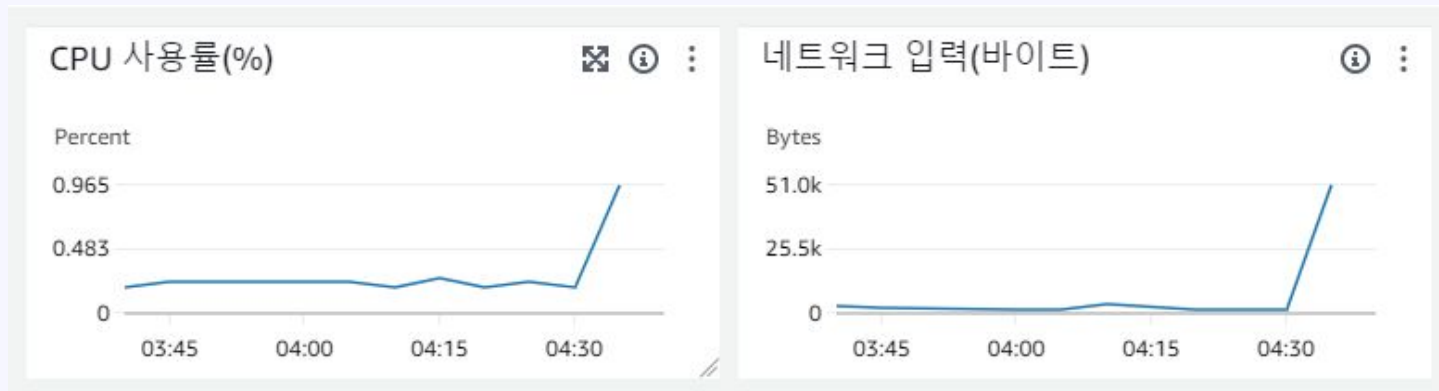
footer {
  background: #333;
  color: white;
}
```

상에
위한
성

모의 DDOS 공격 실행

`hping3 --flood --rand-source -S <IP 주소> -p <포트 번호> --syn --ack`

*타겟 IP주소에 지정한 포트로 --rand-source 랜덤한 소스로부터 Ping flood 유발



인스턴스 모니터링 창에서 **CPU**사용률과 네트워크 입력 증가 확인

boto3 활용 CloudWatch 경보 생성

```
import boto3

# CloudWatch 클라이언트 생성
cloudwatch_client = boto3.client('cloudwatch')

# 알람 생성 함수
def create_alarm(alarm_name, metric_name, namespace, statistic, comparison_operator,
                 threshold, evaluation_periods, period, alarm_actions, dimensions):
    try:
        response = cloudwatch_client.put_metric_alarm(
            AlarmName=alarm_name,
            MetricName=metric_name,
            Namespace=namespace,
            Statistic=statistic,
            ComparisonOperator=comparison_operator,
            Threshold=threshold,
            EvaluationPeriods=evaluation_periods,
            AlarmActions=alarm_actions,
            Period=period,
            Dimensions=dimensions
        )
        return response
    except Exception as e:
        print(e)
```

boto3 활용 CloudWatch 경보 생성

```
# 알람 설정 변수
alarm_name = 'HighNetworkInTrafficAlarm'
metric_name = 'NetworkIn' # EC2 네트워크 인 메트릭
namespace = 'AWS/EC2' # EC2 네임스페이스
statistic = 'Sum'
comparison_operator = 'GreaterThanOrEqualToThreshold'
threshold = 10000000 # 예를 들어, 10MB 이상의 데이터
evaluation_periods = 1
period = 300 # 5분 간격으로 메트릭 평가
alarm_actions = ['arn:aws:sns:ap-northeast-2:950890532950:DDOSALARM']
dimensions = [
    {
        'Name': 'InstanceId',
        'Value': 'i-0c63f2b53ad7cf8c4' # 모니터링하려는 EC2 인스턴스의 ID로 교체하세요
    }
]

# 알람 생성 함수 호출
create_alarm(alarm_name, metric_name, namespace, statistic, comparison_operator, threshold,
             evaluation_periods, period, alarm_actions, dimensions)
```


CloudWatch 경보 발생 및 SNS

CloudWatch > 경보

경보 (1)

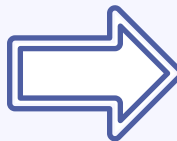
Q 검색

☐ 이름

▼ | 상태

☐ [awsec2-i-03618020c9464f84b-
GreaterThanOrEqualToThreshold-CPUUtilization](#)

⚠ 경보 상태




Amazon SNS

람다활용 오토스케일링 그룹 생성

```
import boto3
def lambda_handler(event, context):
    client = boto3.client('autoscaling')
    try:
        response = client.create_auto_scaling_group(
            AutoScalingGroupName='auto_scaling_group',
            LaunchTemplate={
                'LaunchTemplateId': 'lt-12345678',
                'Version': '1'
            },
            MaxSize=3,
            DesiredCapacity=2,
            AvailabilityZones=['ap-northeast-2a'],
        )
        print("successfully make AS ")
        return {
            'statusCode': 200,
            'body': 'Auto Scaling group created successfully.'
        }
    except Exception as e:
        return {
            'statusCode': 500,
            'body': f'Error creating Auto Scaling group: {str(e)}'
        }
```

람다활용 오토스케일링 그룹 생성

 **DDOS-second-alarm** <no-reply@sns.amazonaws.com>
나에게 ▾

You are receiving this email because your Amazon CloudWatch Alarm "awsec2-i-0c63f2b53ad7cf8 [37462.8 (08/02/24 01:59:00)]" was greater than or equal to the threshold (10000.0)." at "Thursday

View this alarm in the AWS Management Console:
<https://ap-northeast-2.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-northeast-2#al>

Alarm Details:

- Name: i-0c63f2b53ad7cf8c4-GreaterThanOrEqualToThreshold-NetworkIn
- Description: Alarm on instance i-0c63f2b53ad7cf8c4: Triggered when NetworkIn >= 100
- State Change: OK -> ALARM
- Reason for State Change: 1 datapoint [37462.8 (08/02/24 01:59:00)] was g
- Timestamp: Thursday 08 February, 2024 02:09:14 UTC

보안담당자
메일수신 확인,
로그를 통해
람다 정상 실행 확인

```
▶ 2024-02-08T11:13:52.662+09:00 INIT_START Runtime Version: python:3.12.v18 Runtime Version ARN: arn:aws:lambda:ap-northeast-2
▶ 2024-02-08T11:13:52.953+09:00 START RequestId: c82c8dcf-1604-41de-82e8-5ddddd078f60e Version: $LATEST
▶ 2024-02-08T11:13:55.839+09:00 successfully make AS
▶ 2024-02-08T11:13:55.874+09:00 END RequestId: c82c8dcf-1604-41de-82e8-5ddddd078f60e
▶ 2024-02-08T11:13:55.874+09:00 REPORT RequestId: c82c8dcf-1604-41de-82e8-5ddddd078f60e Duration: 2921.19 ms Billed Duration: 29
```

람다활용 오토스케일링 그룹 생성

리소스

EC2 Global view   

아시아 태평양 (서울) 리전에서 다음 Amazon EC2 리소스를 사용하고 있음:

인스턴스(실행 중)	2	로드 밸런서	1	배치 그룹	0
보안 그룹	21	볼륨	3	스냅샷	2
인스턴스	5	전용 호스트	0	키 페어	2
탄력적 IP	4	Auto Scaling 그룹	0		

project_ec2	i-0fa6523eeeb968312	✔ 실행 중	🔍 🔍	t2.micro	✔ 2/2개 검사 통과...
project2	i-0c63f2b53ad7cf8c4	✔ 실행 중	🔍 🔍	t2.micro	✔ 2/2개 검사 통과...
	i-0dd3a3fb90e5cc772	✔ 실행 중	🔍 🔍	t2.micro	🕒 초기화
	i-07df859bd3272c764	✔ 실행 중	🔍 🔍	t2.micro	🕒 초기화

람다 활용 ROUTE53 RE-routing

01

공격 지속

AS실행 이후에도
부하가 줄지
않는다면?

02

2차경보 설정

1차보다 기준이
높은
2차 클라우드 워치
경보 설정

03

백업 서버 활용

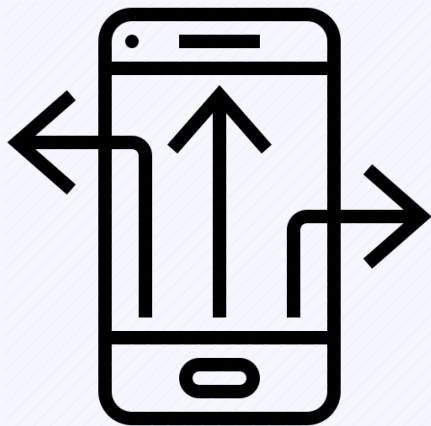
2차 경보가 울리면
람다를 활용해
백업 인스턴스로
라우트 리다이렉팅

04

부하분산 구현

부하분산을 통해
서비스 안정화 실현

람다 활용 ROUTE53 RE-routing



라우트 리다이렉팅?

사용자가 특정 도메인 또는 URL로 요청을 보냈을 때,
그 요청을 다른 IP 나 도메인 또는 URL로
자동으로 전송하는 과정

람다 활용 ROUTE53 RE-routing

Welcome to CloudSecurity group 1

레코드 (4) 정보



레코드 삭제

영역 파일 가져오기

레코드 생성

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

Q 속성 또는 값을 기준으로 레코드 필터링

유형 ▼

라우팅 정책 ▼

별칭 ▼

< 1 >



<input type="checkbox"/>	레코드 이름 ▼	유형 ▲	라우팅 ... ▼	차별화 ... ▼	별칭 ▼	값/트래픽 라우팅 대상 ▼	TTL(초)
<input type="checkbox"/>	cloudsimon.click	A	단순	-	아니요	15.165.0.70	300
<input type="checkbox"/>	www.cloudsimon.click	A	단순	-	아니요	15.165.0.70	60

람다 활용 ROUTE53 RE-routing

Name	인스턴스 ID	인스턴스 상태	인스턴...	퍼블릭 IPv4 주소
project_original	i-0c63f2b53ad7cf8c4	실행 중	t2.micro	15.165.0.70
project_backup	i-06eeadee5734ef65b	실행 중	t2.micro	3.34.102.90

```
record_set = {
    'Name': 'www.cloudsimon.click.',
    'Type': 'A',
    'TTL': 60,
    'ResourceRecords': [{"Value": '3.34.102.90'}]}

return response

except Exception as e:
    print("Route 53 record set 변경 요청 중 오류가 발생했습니다:", e)
    raise e

print("Route 53 record set 변경 요청 중 오류가 발생했습니다:", e)
raise e
```


람다 활용 ROUTE53 RE-routing

레코드 (4) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

속성 또는 값을 기준으로 레코드 필터링

레코드 이름	유형	라우팅 ...	차별...	별칭	값/트래픽 라우팅
<input type="checkbox"/> cloudsimon.click	A	단순	-	아니요	15.165.0.70
<input type="checkbox"/> www.cloudsimon.click	A	단순	-	아니요	3.34.102.90
<input type="checkbox"/> cloudsimon.click	NS	단순	-	아니요	ns-608.awsdns-12.net. ns-1203.awsdns-22.org. ns-469.awsdns-58.com.
<input type="checkbox"/> cloudsimon.click	SOA	단순	-	아니요	ns-1725.awsdns-23.co.uk. a...

백업 서버의
IP로 변환

람다 활용 ROUTE53 RE-routing

```
C:\Users\WYONSAI>nslookup  
기본 서버:  dns.google  
Address:  8.8.8.8
```

```
> www.cloudsimon.click  
서버:  dns.google  
Address:  8.8.8.8
```

```
권한 없는 응답:  
이름:  www.cloudsimon.click  
Address:  3.34.102.90
```

```
> cloudsimon.click  
서버:  dns.google  
Address:  8.8.8.8
```

```
권한 없는 응답:  
이름:  cloudsimon.click  
Address:  15.165.0.70
```

```
C:\Users\WYONSAI>nslookup  
기본 서버:  dns.google  
Address:  8.8.8.8
```

```
> www.cloudsimon.click  
서버:  dns.google  
Address:  8.8.8.8
```

```
권한 없는 응답:  
이름:  www.cloudsimon.click  
Address:  3.34.102.90
```

```
> cloudsimon.click  
서버:  dns.google  
Address:  8.8.8.8
```

```
권한 없는 응답:  
이름:  cloudsimon.click  
Address:  3.34.102.90
```

Nslookup 명령어로 Ip변화 확인

06

결과 소회 및 자체 평가

Reflection on the results and
self-assessment

결과 소회

"혼자서는 할 수 없는 일도 함께하면 할 수 있다." - 익명

"협력은 성공의 가장 빠른 방법이다." - 익명

프로젝트는 개인의 능력도 물론 중요하지만 팀워크가 더 중요합니다.
서로 협력하고 주제에 대해 고민하고 의견을 나누면서 결과물을
만들어가는 과정이 무엇보다 중요하다고 생각합니다.
클라우드 보안이라는 영역 안에 자동화라는 어려운 주제를 선택함에도
불구하고 끝까지 해낸 조원들에게 또 발표를 들으시는 분들에게
감사함을 표합니다.

The background features decorative geometric patterns in the corners, consisting of thin blue lines, dots, and circles. A horizontal line with dots at its ends is positioned below the '07' text.

07

질의 응답 Q&A

Reference

업무에 바로 쓰는 **AWS 입문(도서)**-aws 람다 파트

→7장 람다파트 p.190~206

AWS 사용 설명서-개념정리 및 자료 조사

-ec2 개념→https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/concepts.html

-자동화 개념→https://docs.aws.amazon.com/ko_kr/userguide/systems-manager-automation.h

-LAMBDA(AWS) 개념→<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

Bard(Ai)-ec2 자동화 코드 참조

→<https://g.co/bard/share/847ef96544d1>

네이버 뉴스-자동화 사례연구 기사 참조

→https://www.newsis.com/view/?id=NISX20230204_0002181054&cID=13005&pID=13100

클라우드 보안 실무가이드 (도서)-황치하 양지언 지음

→3장 클라우드 보안 서비스 122-192p

The background features abstract geometric patterns in the corners, consisting of thin blue lines, dots, and small circles, creating a modern, minimalist aesthetic.

감사합니다.