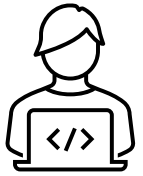


ELB

# ELB : AWS에서 제공하는 트래픽 분산 서비스

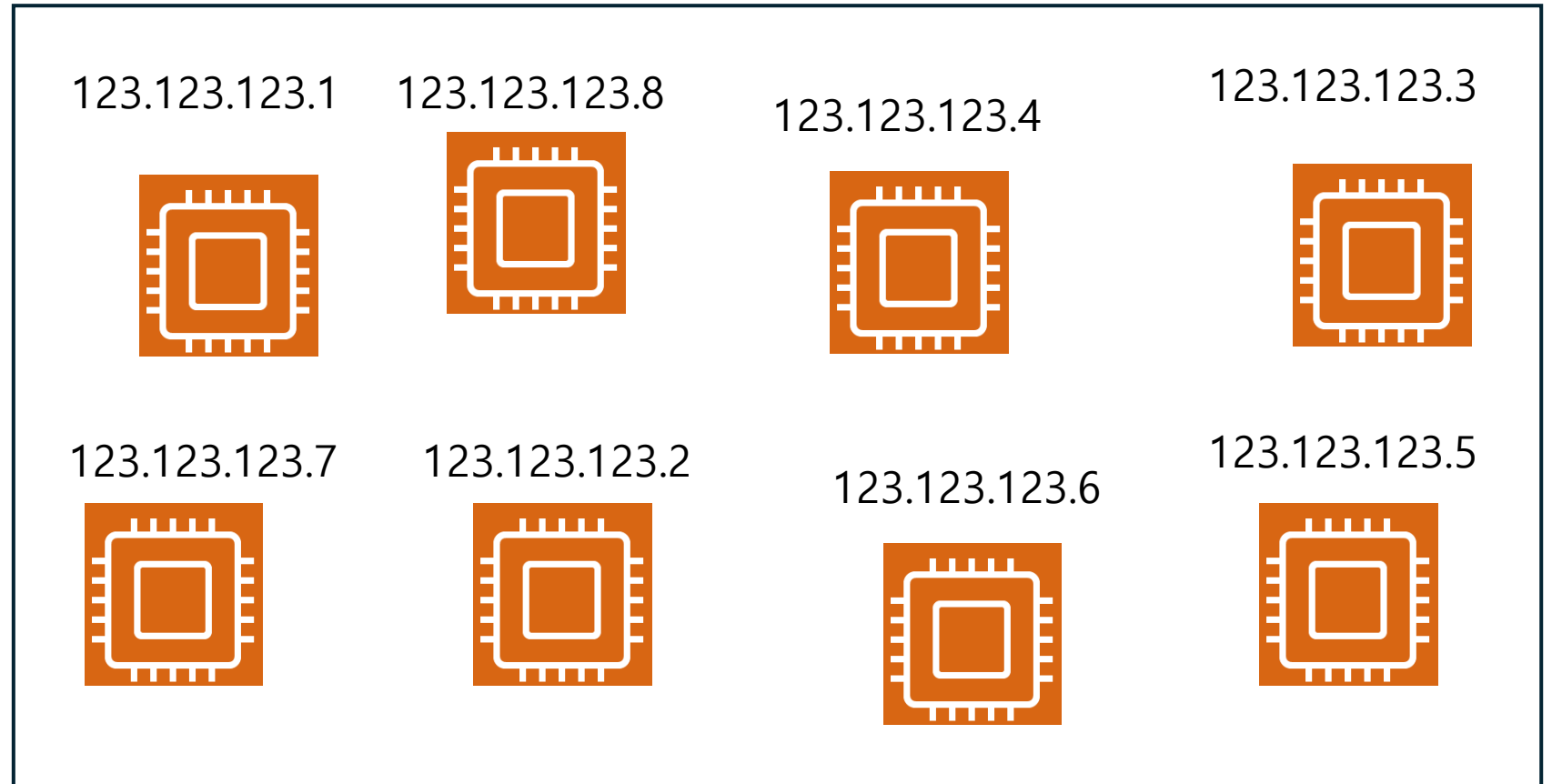
사용자가 EC2에 접근하려면 모든 EC2 IP 주소 알아야 함! 만약 EC2 한대가 종료된다면?



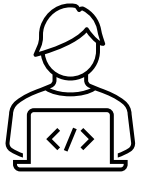
123.123.123.1

~

123.123.123.8



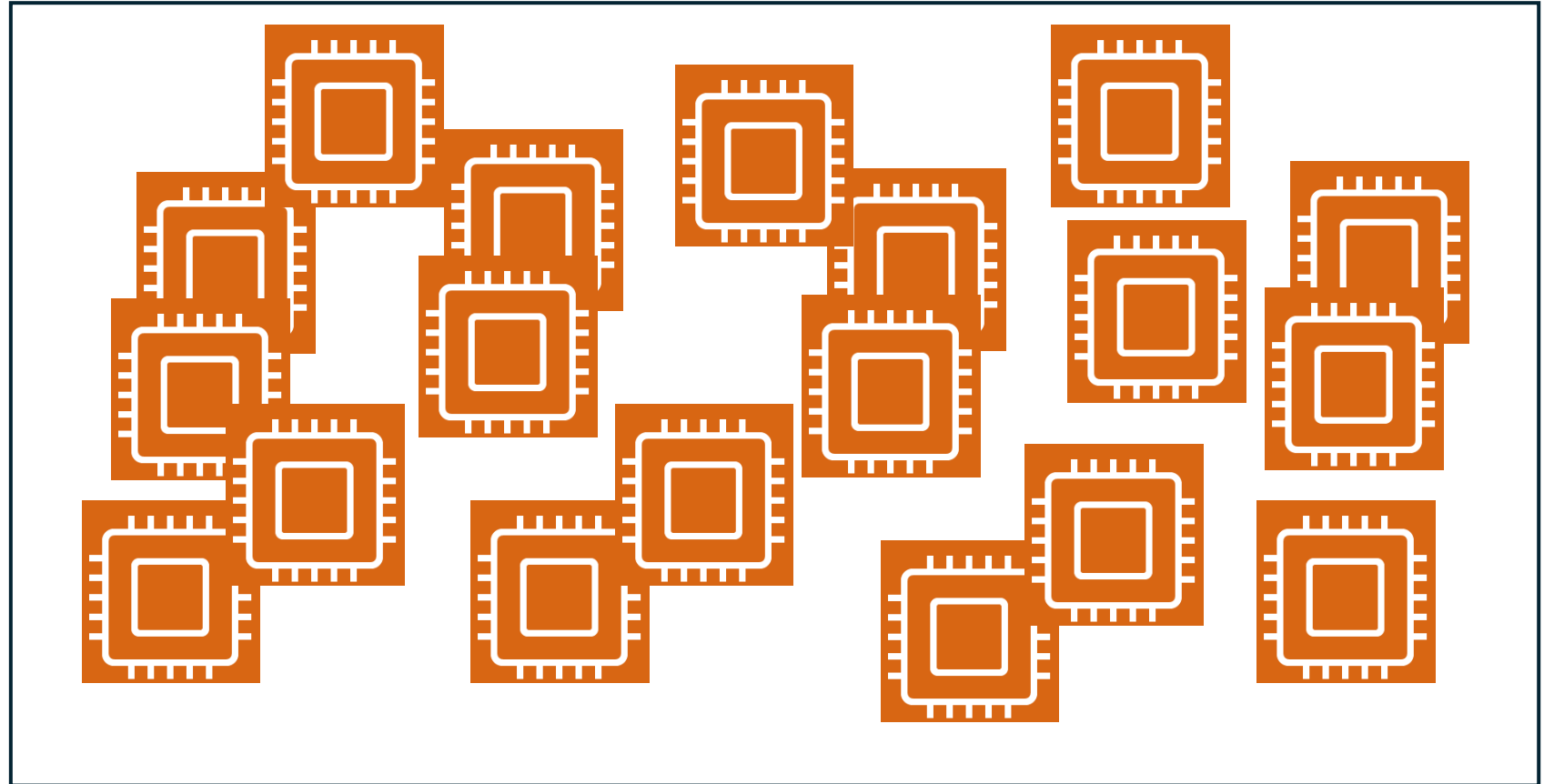
# 만약 사용자가 관리 할 수 없을 정도로 EC2가 많다면?



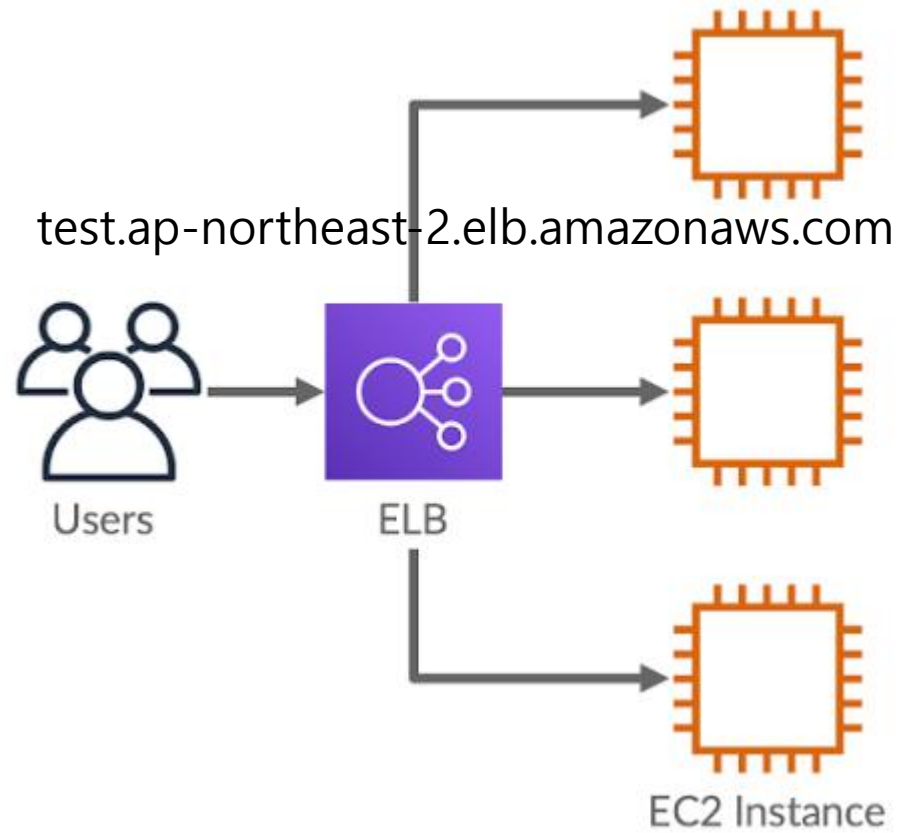
123.123.123.1

~

123.123.123.255



# 그래서 ELB!



# Elastic Load Balancer

- ELB는 들어오는 어플리케이션 트래픽을 Amazon 인스턴스, 컨테이너, IP 주소, 람다 함수와 같은 여러 대상에 자동 분산 시킴.
- ELB는 단일 가용영역, 여러 가용 영역에서 다양한 어플리케이션 부하를 처리 가능.
- ELB에서 제공하는 3가지 로드밸런서는 모두 어플리케이션의 내결합성에 필요한고가용성, 자동 확장/축소, 강력한 보안을 갖추고 있음

# ELB 특징

- 다수의 서비스에 트래픽을 분산 시켜주는 서비스
- Health Check: 직접 트래픽을 발생시켜 Instance가 살아있는지 체크
- Autoscaling 과 연동 가능
- 여러 가용영역에 분산 가능
- 지속적으로 IP 주소가 바뀌며 IP 고정 불가능: 항상 도메인 기반으로 사용
- 총 3가지 종류
  - Application Load Balancer
  - Network Load Balancer
  - Classic Load Balancer

# ELB 종류

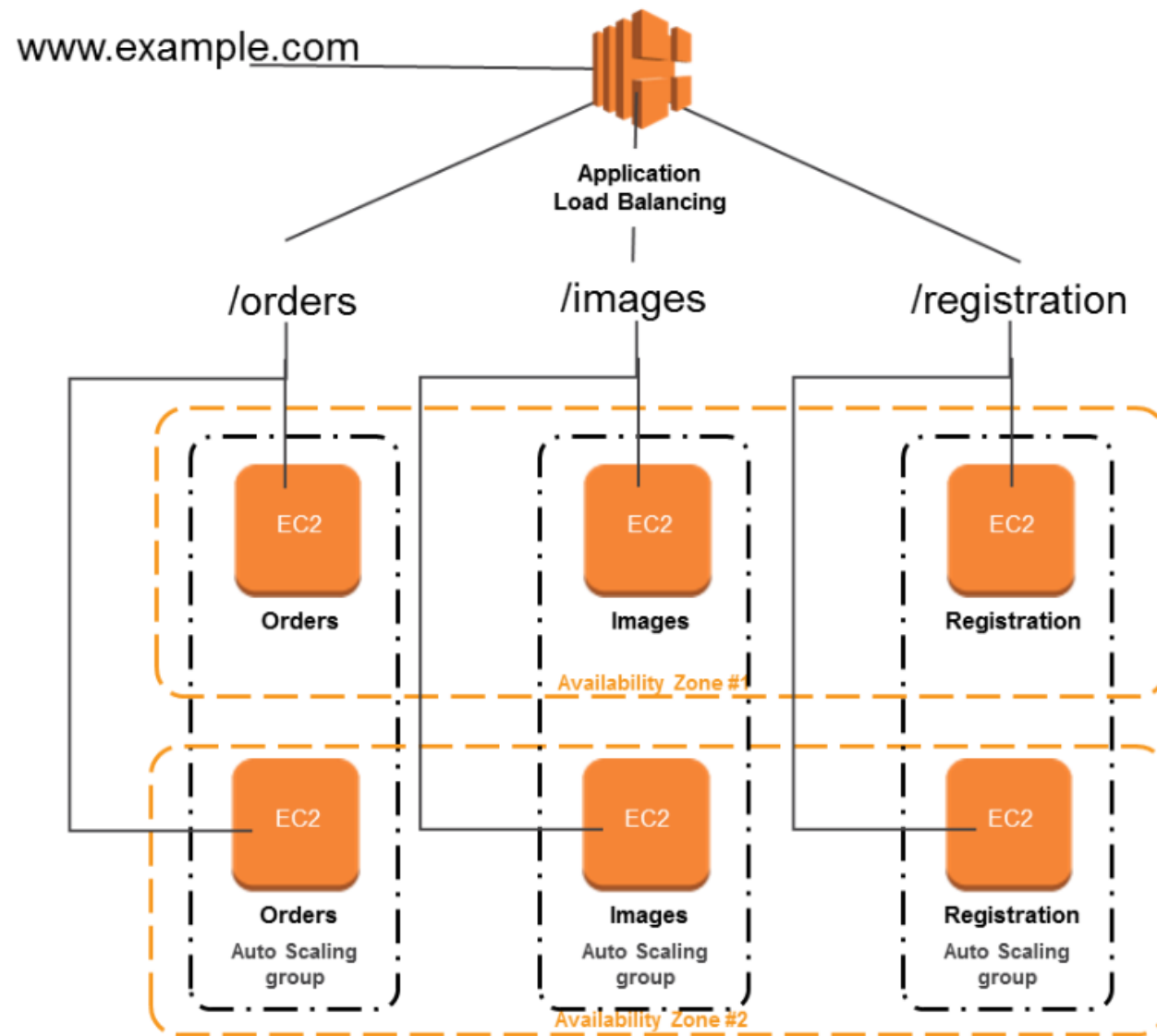
- Application Load Balancer : 똑똑함
  - 트래픽을 모니터링하여 라우팅 가능
  - image.test.com → 이미지 서버로, web.test.com → 웹 서버로 트래픽 분산
- Network Load Balancer : 빠름
  - TCP 기반 빠른 트래픽 분산
  - Elastic IP 할당 가능
- Classic Load Balancer : 옛날..
  - 지금 잘 사용하지 않음

# ALB - Target Group

- 타겟 그룹: ALB가 트래픽을 분산시킬 대상(Target)을 논리적으로 묶어놓은 그룹
- 타겟 그룹 기준으로 요청 분배
- EC2 인스턴스, IP 주소, Lambda 함수, ELB 타겟 그룹 가능
- 역할
  - ALB → 라우팅 규칙에 따라 타겟 그룹으로 요청 전달
  - 타겟 그룹은 내부에서 등록된 인스턴스 중 헬스체크를 통과한 대상에게 트래픽 전달
  - 타겟 그룹별로 포트, 프로토콜, 헬스체크 설정 가능



# ALB 구성 예

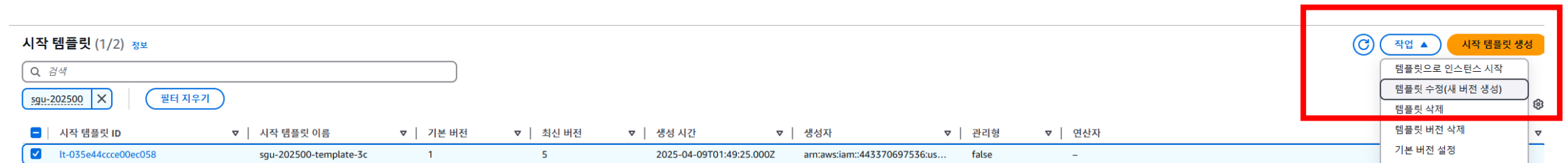


# 실습1) ALB 생성하기!

- 1. 시작 템플릿 수정!
- 2. 시작 템플릿으로 EC2 생성
- 3.Target 생성
- 4.ELB 생성

# 실습1) ALB – 시작템플릿 수정하기!

- 시작템플릿 → 작업 → 템플릿 수정(새버전생성)



The screenshot shows the AWS CloudFormation console. At the top, there's a search bar and a table of templates. The '작업' (Actions) button is highlighted with a red box, and its dropdown menu is open, showing options like '템플릿으로 인스턴스 시작' (Start instance with template), '템플릿 수정(새 버전 생성)' (Edit template (create new version)), '템플릿 삭제' (Delete template), '템플릿 버전 삭제' (Delete template version), and '기본 버전 설정' (Set default version). The '템플릿 수정(새 버전 생성)' option is the one to be selected.

시작 템플릿 ID	시작 템플릿 이름	기본 버전	최신 버전	생성 시간	생성자	관리형	연산자
lt-035e44ccce00ec058	sgu-202500-template-3c	1	5	2025-04-09T01:49:25.000Z	arn:aws:iam::443370697536:us...	false	-

- 고급세부정보 클릭 → 사용자 데이터 수정 → 템플릿 버전 생성

## 사용자 데이터 - 선택 사항 | 정보

사용자 데이터가 포함된 파일을 업로드하거나 필드에 입력합니다.

📎 파일 선택

```
#!/bin/bash
yum install httpd -y
systemctl enable httpd
systemctl start httpd
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
INSTANCE_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/instance-id)
echo "$INSTANCE_ID" > /var/www/html/index.html
```

템플릿 버전 생성

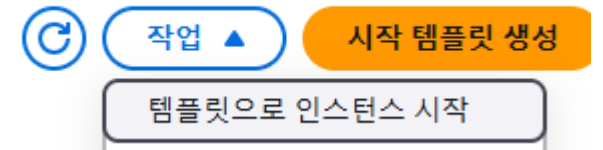
# 실습1) ALB – 시작 템플릿 수정하기!

```
#!/bin/bash
yum install httpd -y
systemctl enable httpd
systemctl start httpd
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
INSTANCE_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/instance-id)
echo "$INSTANCE_ID" > /var/www/html/index.html
```

인스턴스 ID가져오는 스크립트

# 실습1) ALB – 시작 템플릿으로 EC2 생성

- 시작 템플릿 → 작업 → 시작 템플릿으로 인스턴스 시작
- 최신버전으로 인스턴스 시작하기!



## 템플릿으로 인스턴스 시작

템플릿에서 시작하면 예전에 저장해 두었던 인스턴스 구성으로 시작할 수 있습니다. 이러한 저장된 구성을 재사용하거나 다른 사용자와 공유함으로써

### 시작 템플릿 선택

원본 템플릿

sgu-202500-template-3c  
ID: lt-035e44ccce00ec058



5

instance\_id



# 실습1) ALB – 시작 템플릿으로 EC2 생성

- 해당 EC2가 생성된 것을 확인해보자! → 확인완료 후, ec2 한대 더 생성하기!

← → ↺ ⚠ 주의 요함 52.78.207.51

☰

i-0541f59416eb5b722

- 아래와 같이 생성 후, 이름 바꾸기! 계정+ALB1, 계정+ALB2

<input type="checkbox"/>	sgu-202500-ALB1	i-0541f59416eb5b722	✔ 실행 중	🔍 🔍
<input checked="" type="checkbox"/>	sgu-202500-ALB2	i-02fa776c971ce156d	✔ 실행 중	🔍 🔍

# 실습1) ALB – 타켓 생성!

- 로드밸런싱 → 대상그룹 → 대상그룹생성

## ▼ 로드 밸런싱

로드밸런서

대상 그룹

Trust Store

Amazon EC2 Auto Scaling

## 그룹 세부 정보 지정

로드 밸런서는 요청을 대상 그룹의 대상으로 라우팅하고 대상에 대한 상태 확인을 수행합니다.

### 기본 구성

대상 그룹이 생성된 후에는 이 섹션의 설정을 변경할 수 없습니다.

### 대상 유형 선택

#### ☒ 인스턴스

- 특정 VPC 내의 인스턴스에 대한 로드 밸런싱을 지원합니다.
- [Amazon EC2 Auto Scaling](#)을 사용하여 EC2 용량을 관리하고 크기를 조정할 수 있습니다.

sgu-202500-3a-target

하이픈을 포함하여 최대 32자의 영숫자 문자를 사용할 수 있지만 이름이 하이픈으로 시작하거나 끝나지 않아야 합니다.

### 프로토콜 : 포트

트래픽을 라우팅할 로드 밸런서 유형에 해당하는 대상 그룹의 프로토콜을 선택합니다. 일부 프로토콜에는 이제 대상에 대한 이상 탐지 기능을 포함되어 있으며, 대상 그룹이 생성되면 수 있습니다.

HTTP

80

1-65535

### IP 주소 유형

표시된 IP 주소 유형의 대상만 이 대상 그룹에 등록할 수 있습니다.

#### ☒ IPv4

각 인스턴스에는 기본 프라이빗 IPv4 주소가 할당된 기본 네트워크 인터페이스(eth0)가 있습니다. 인스턴스의 기본 프라이빗 IPv4 주소는 대상에 적용되는 주소입니다.

#### ☐ IPv6

등록하는 각 인스턴스에는 할당된 기본 IPv6 주소가 있어야 합니다. 이는 인스턴스의 기본 네트워크 인터페이스(eth0)에서 구성됩니다. [자세히 알아보기](#)

### VPC

대상 그룹에 포함할 인스턴스가 있는 VPC를 선택합니다. 위에서 선택한 IP 주소 유형을 지원하는 VPC만 이 목록에서 사용할 수 있습니다.

shingu\_class

vpc-0352e69bb13e04b28

IPv4 VPC CIDR: 172.31.0.0/16

### 프로토콜 버전

#### ☒ HTTP1

HTTP/1.1을 사용하여 대상으로 요청을 전송합니다. 요청 프로토콜이 HTTP/1.1 또는 HTTP/2일 때 지원됩니다.

- 기본 구성: 인스턴스

- 대상그룹이름 : sgu-202500-3a-target

# 실습1) ALB - 타켓 생성 → 대상 등록

- 생성한 EC2 체크 → 아래에 보류 중인 것으로 포함 클릭 → 대상그룹 생성

## 대상 등록

이는 대상 그룹을 생성하기 위한 선택적 단계입니다. 그러나 로드 밸런서가 이 대상

2개 선택됨

### 사용 가능한 인스턴스 (2/5)

인스턴스 필터링

인스턴스 ID 이름

<input checked="" type="checkbox"/>	i-02fa776c971ce156d	sgu-202500-ALB2
<input checked="" type="checkbox"/>	i-0541f59416eb5b722	sgu-202500-ALB1

### 선택한 인스턴스를 위한 포트

선택한 인스턴스로 트래픽을 라우팅하기 위한 포트입니다.

80

1-65535(임의로 여러 포트 구분)

아래에 보류 중인 것으로 포함

### 대상 보기

대상 (2)

보류 중인 모든 항목 제거

< 1 > ⚙

대상 필터링

☐ 대기 중인 항목만 보기

인스턴스 ID	이름	포트	상태	보안 그룹	영역	프라이빗 IPv4 주소	서브넷 ID	시작 시간
i-02fa776c971ce156d	sgu-202500-ALB2	80	실행 중	sgu-202500-sgtest2	ap-northeast-2a	172.31.6.24	subnet-038db19e8a7e100e8	2025년 4월 14일, 20:52 (UTC+09:00)
i-0541f59416eb5b722	sgu-202500-ALB1	80	실행 중	sgu-202500-sgtest2	ap-northeast-2a	172.31.6.119	subnet-038db19e8a7e100e8	2025년 4월 14일, 20:27 (UTC+09:00)

2개 대기 중

취소

이전

대상 그룹 생성



# 실습1) 대상그룹 생성 된 것을 확인!

🟢 대상 그룹 sgu-202500-3a-target이(가) 생성되었습니다. 이상 탐지는 등록된 모든 대상에 자동으로 적용됩니다. 결과는 대상 탭에서 볼 수 있습니다.

sgu-202500-3a-target

작업 ▼

## 세부 정보

🔗 [arn:aws:elasticloadbalancing:ap-northeast-2:443370697536:targetgroup/sgu-202500-3a-target/85275f54592b933e](#)

대상 유형  
인스턴스

프로토콜 : 포트  
HTTP: 80

프로토콜 버전  
HTTP1

VPC

[vpc-0352e69bb13e04b28](#)

IP 주소 유형  
IPv4

로드 밸런서  
[연결된 항목 없음](#)

2  
대상 합계

🟢 0  
정상  
0 이상

🔴 0  
비정상

🕒 2  
사용되지 않음

🕒 0  
초기

🕒 0  
드레이닝

## ▶ 가용 영역별 대상 배포

아래의 등록된 대상 테이블에 적용된 해당 필터를 보려면 이 테이블에서 값을 선택합니다.

대상

모니터링

상태 검사

속성

태그

## 등록된 대상 (2) 정보

[이상 완화: 해당되지 않음](#)



등록 취소

대상 등록

대상 그룹은 지정한 프로토콜 및 포트 번호를 사용하여 등록된 개별 대상으로 요청을 라우팅합니다. 상태 확인은 대상 그룹의 상태 확인 설정에 따라 등록된 모든 대상에 대해 수행됩니다. 이상 탐지는 정상 대상이 3개 이상 있는 HTTP/HTTPS 대상 그룹에 자동으로 적용됩니다.

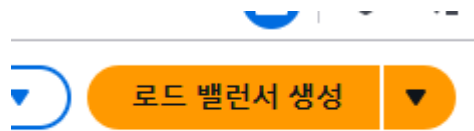
🔍 대상 필터링

< 1 > ⚙️

<input type="checkbox"/> 인스턴스 ID	이름	포트	영역	상태 확인	상태 확인 세부 정보	관리상 재정의	재정의 세부
<input type="checkbox"/> <a href="#">i-02fa776c971ce156d</a>	sgu-202500-A...	80	ap-northeast-...	🕒 Unused	Target group is not co...	-	-
<input type="checkbox"/> <a href="#">i-0541f59416eb5b722</a>	sgu-202500-A...	80	ap-northeast-...	🕒 Unused	Target group is not co...	-	-

# 실습1)로드밸런서 생성

- 로드밸런서 생성 버튼 클릭→ Application Load Balancer 생성



# 실습1) ALB 생성하기!

- 로그 밸런서 이름 : sgu-202500-3a-ALB
- 가용영역 다 선택, 보안그룹은 나의 보안그룹

## 기본 구성

### 로드 밸런서 이름

이름은 AWS 계정 내에서 고유해야 하며 로드 밸런서 생성 후에는 변경할 수 없습니다.

sgu-202500-3a-ALB

하이픈을 포함하여 최대 32자의 영숫자 문자를 사용할 수 있지만 이름이 하이픈으로 시작하거나 끝

### 체계 | 정보

로드 밸런서 생성 후에는 스키마를 변경할 수 없습니다.

#### 인터넷 경계

- 인터넷 경계 트래픽을 처리합니다.
- 퍼블릭 IP 주소가 있습니다.
- DNS 이름은 공개적으로 확인할 수 있습니다.
- 퍼블릭 서브넷이 필요합니다.

### 로드 밸런서 IP 주소 유형 | 정보

로드 밸런서에 할당할 프론트엔드 IP 주소 유형을 선택합니다. 이 로드 밸런서에 매핑된 VPC 및 서

#### IPv4

IPv4 주소만 포함합니다.

#### 듀얼 스택

IPv4 및 IPv6 주소를 포함합니다.

#### 퍼블릭 IPv4가 없는 듀얼 스택

퍼블릭 IPv6 주소와 프라이빗 IPv4 및 IPv6 주소를 포함합니다. 인터넷 연결 로드 밸런서와만 :

### 가용 영역 및 서브넷 | 정보

가용 영역을 2개 이상 선택하고 각 영역에 대해 서브넷을 선택합니

#### ☒ ap-northeast-2a (apne2-az1)

##### 서브넷

로드 밸런서 IP 주소 유형에 해당하는 CIDR 블록만 사용됩니

subnet-038db19e8a7e100e8  
IPv4 서브넷 CIDR: 172.31.0.0/20

#### ☒ ap-northeast-2b (apne2-az2)

##### 서브넷

로드 밸런서 IP 주소 유형에 해당하는 CIDR 블록만 사용됩니

subnet-0e84271f9f8e47d9a  
IPv4 서브넷 CIDR: 172.31.16.0/20

#### ☒ ap-northeast-2c (apne2-az3)

##### 서브넷

로드 밸런서 IP 주소 유형에 해당하는 CIDR 블록만 사용됩니

subnet-071f1b110ddb3becc  
IPv4 서브넷 CIDR: 172.31.32.0/20

#### ☒ ap-northeast-2d (apne2-az4)

##### 서브넷

로드 밸런서 IP 주소 유형에 해당하는 CIDR 블록만 사용됩니

subnet-0150a8faab1883eac  
IPv4 서브넷 CIDR: 172.31.48.0/20

## 보안 그룹 정보

보안 그룹은 로드 밸런서에 대한 트래픽을 제어하는 방화벽 규칙

## 보안 그룹

최대 5개의 보안 그룹 선택

sgu-202500-sgtest2

sg-0b2c3c51e042c9bb3 VPC: vpc-0352e69bb13e04b28

# 실습1) ALB 생성하기!

- 리스너 및 라우팅 : 80으로 들어온 것을 내가 생성한 대상그룹으로 전달! → 로드밸런서 생성 → 활성화까지 몇분 걸림

## 리스너 및 라우팅 정보

리스너는 사용자가 구성한 포트 및 프로토콜을 사용하여 연결 요청을 검사하는 프로세스입니다. 리스너에 대해 정의한 규칙에 따라 로드 밸런서가 등록된 대상으로 요청을 라우팅하는 방법이 결정됩니다.

### ▼ 리스너 HTTP:80

프로토콜

HTTP

포트

80

1-65535

기본 작업 | 정보

다음으로 전달:

sgu-202500-3a-target

대상 유형: 인스턴스, IPv4

HTTP

[대상 그룹 생성](#)

### 리스너 태그 - 선택 사항

리스너에 태그를 추가하는 것을 고려하십시오. 태그를 사용하면 AWS 리소스를 분류하

로드 밸런서 생성 완료: sgu-202500-3a-ALB

로드 밸런서가 완전히 설정되어 트래픽을 라우팅하려면 몇 분이 걸릴 수 있습니다. 대상도 등록 프로세스를 완료한 후 초기 상태 확인을 통과하려면 몇 분이 걸릴 수 있습니다.

이제 Application Load Balancer가 퍼블릭 IPv4 IP 주소 관리(IPAM)를 지원합니다.  
네트워크 매핑 섹션에서 IP 풀을 구성하여 이 기능을 시작할 수 있습니다.

IP 풀 편집

## sgu-202500-3a-ALB

### ▼ 세부 정보

로드 밸런서 유형  
애플리케이션

체계  
Internet-facing

상태  
프로비저닝 중

호스팅 영역  
ZWKZPGT148KDX

VPC  
[vpc-0352e69bb13e04b28](#)

가용 영역  
[subnet-071f1b110ddb3becc](#) ap-northeast-2c (apne2-az3)  
[subnet-038db19e8a7e100e8](#) ap-northeast-2a (apne2-az1)  
[subnet-0150a8faab1883eac](#) ap-northeast-2d (apne2-az4)  
[subnet-0e84271f9f8e47d9a](#) ap-northeast-2b (apne2-az2)

로드 밸런서 IP 주소 유형  
IPv4

생성된 날짜  
2025년 4월 14일, 21:07 (UTC+09:00)

# 실습1) ALB 생성하기!

- 프로비저닝 중 : 리소스를 할당하고 준비하고 있는 단계

## 로드 밸런서 (3)

Elastic Load Balancing은 수신 트래픽의 변화에 따라 자동으로 로드 밸런서 용량을 확장합니다.

Q 로드 밸런서 필터링

<input type="checkbox"/>	이름	▼	DNS 이름	▼	상태
<input type="checkbox"/>	sgu-202500-ALB		sgu-202500-ALB-13676801...		✓ 활성화
<input type="checkbox"/>	lucy-test		lucy-test-1676159921.ap-n...		✓ 활성화
<input type="checkbox"/>	sgu-202500-3a-ALB		sgu-202500-3a-ALB-13309...		⋮ 프로비저닝 중

# 실습1) ALB 테스트!

## • DNS 정보 복사!

sgu-202500-3a-ALB

작업 ▼

### ▼ 세부 정보

로드 밸런서 유형  
애플리케이션

체계  
Internet-facing

상태  
🟢 활성

호스팅 영역  
ZWKZPGT148KDX

VPC  
[vpc-0352e69bb13e04b28](#)

가용 영역  
[subnet-071f1b110ddb3becc](#) ap-northeast-2c (apne2-az3)  
[subnet-038db19e8a7e100e8](#) ap-northeast-2a (apne2-az1)  
[subnet-0150a8faab1883eac](#) ap-northeast-2d (apne2-az4)  
[subnet-0e84271f9f8e47d9a](#) ap-northeast-2b (apne2-az2)

로드 밸런서 IP 주소 유형  
IPv4

생성된 날짜  
2025년 4월 14일, 21:07 (UTC+09:00)

### 로드 밸런서 ARN

[arn:aws:elasticloadbalancing:ap-northeast-2:443370697536:loadbalancer/app/sgu-202500-3a-ALB/1ae599bde6af858a](#)

### DNS 이름 정보

[sgu-202500-3a-ALB-1330996514.ap-northeast-2.elb.amazonaws.com](#) (A 레코드)

⏪ ⏩ ↺ ⚠ 주의 요함 sgu-202500-3a-alb-1330996514.ap-northeast-2.elb.amazonaws.com



i-02fa776c971ce156d

## 실습2) 경로기반 리디렉션

- /s3 경로로 들어온 요청을 S3 버킷의 index.html로 보내는 실습!
- 1. S3 설정
- 2. 리스너 규칙 추가

# 실습2) 경로기반 리디렉션 : s3 설정 변경

- 버킷 속성 → 정적 웹 사이트 호스팅 활성화

sgu-202500-s3-3c 정보

객체

속성

권한

지표

관리

액세스 지정

객체 (6)

객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. [Amazon S3 인벤토리](#)를 사용하여 버킷0

Q 접두사로 객체 찾기

☐ 이름

☐ class/

☐ index.html

☐ index.php

☐ s3/

☐ test.log

☐ test.txt

정적 웹 사이트 호스팅

이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션하십시오. [자세히 알아보기](#)

정적 웹 사이트 호스팅에는 **AWS Amplify Hosting**을 사용하는 것이 좋습니다.  
AWS Amplify Hosting을 사용하여 빠르고 안전하며 안정적인 웹 사이트를 빠르게 배포하세요. [Amplify Hosting](#)에 대해 자세히 알아보거나 [기존 Amplify 앱을 확인하세요](#).

S3 정적 웹 사이트 호스팅  
비활성됨

txt

편집

Amplify 앱 생성



# 실습2) 경로기반 리디렉션 : s3 설정 변경

- 정적 웹사이트 호스팅 : 활성화 체크
- 인덱스문서 : index.html

정적 웹 사이트 호스팅 편집 [정보](#)

## 정적 웹 사이트 호스팅

이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션하십시오. [자세히 알아보기](#)

### 정적 웹 사이트 호스팅

☐ 비활성화

☒ 활성화

### 호스팅 유형

☒ 정적 웹 사이트 호스팅

버킷 엔드포인트를 웹 주소로 사용합니다. [자세히 알아보기](#)

☐ 객체에 대한 요청 리디렉션

요청을 다른 버킷 또는 도메인으로 리디렉션합니다. [자세히 알아보기](#)

**!** 고객이 웹 사이트 엔드포인트의 콘텐츠에 액세스할 수 있게 하려면 모든 콘텐츠를 공개적으로 읽기 가능하도록 설정해야 합니다. 이렇게 하려면, 버킷에 대한 S3 퍼블릭 차단 사용 [참조](#) 참조하십시오.

### 인덱스 문서

웹 사이트의 홈 페이지 또는 기본 페이지를 지정합니다.

index.html

### 오류 문서 - 선택 사항

오류가 발생하면 반환됩니다.

error.html

### 리디렉션 규칙 - 선택 사항

JSON으로 작성된 리디렉션 규칙은 특정 콘텐츠에 대한 웹 페이지 요청을 자동으로 리디렉션합니다. [자세히 알아보기](#)

# 실습2) 경로기반 리디렉션 : s3 설정 변경

## • 버킷 웹 사이트 엔드포인트 확인

### 정적 웹 사이트 호스팅

[편집](#)

이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션하십시오. [자세히 알아보기](#)

정적 웹 사이트 호스팅에는 **AWS Amplify Hosting**을 사용하는 것이 좋습니다.

[Amplify 앱 생성](#)

AWS Amplify Hosting을 사용하여 빠르고 안전하며 안정적인 웹 사이트를 빠르게 배포하세요. [Amplify Hosting](#)에 대해 자세히 알아보거나 [기존 Amplify 앱을 확인하세요](#).

### S3 정적 웹 사이트 호스팅

활성화됨

호스팅 유형

버킷 호스팅

### 버킷 웹 사이트 엔드포인트

버킷을 정적 웹 사이트로 구성하면, 해당 웹 사이트를 버킷의 AWS 리전별 웹 사이트 엔드포인트에서 사용할 수 있습니다. [자세히 알아보기](#)

<http://sgu-202500-s3-3c.s3-website.ap-northeast-2.amazonaws.com>

# 실습2) 경로기반 리디렉션 :ALB설정 변경

- 해당 ALB 클릭 → 리스너 규칙 추가

리스너 및 규칙 | 네트워크 매핑 | 리소스 맵 | 보안 | 모니터링 | 통합 | 속성 | 용량 | 태그

### 리스너 및 규칙 (1/1) 정보

리스너는 구성된 프로토콜 및 포트에서 연결 요청을 확인합니다. 리스너가 수신한 트래픽은 기본 작업 및 기타 추가 규칙에 따라 라우팅됩니다.

🔍 리스너 필터링

<input checked="" type="checkbox"/>	프로토콜: 포트 ▼	기본 작업 ▼	규칙 ▼	ARN ▼	보안 정책 ▼	기본 SSL/TLS ▼
<input checked="" type="checkbox"/>	<a href="#">HTTP:80</a>	대상 그룹으로 전달 <ul style="list-style-type: none"><li><a href="#">sgu-202500-3a-target</a> 1 (100%)</li><li>대상 그룹 고정성: 끄</li></ul>	<a href="#">1개 규칙</a>	ARN	해당되지 않음	해당되지 않음

규칙 관리 ▲ 리스너 관리

- 규칙 추가
- 규칙 편집
- 규칙 우선 순위 변경

# 실습2) 경로기반 리디렉션 :규칙 추가

- 이름 : redirection to S3

- 1단계  
● 규칙 추가
- 2단계  
○ 규칙 조건 정의
- 3단계  
○ 규칙 작업 정의
- 4단계  
○ 규칙 우선 순위 설정
- 5단계  
○ 검토 및 생성

## 규칙 추가 정보

규칙을 정의한 다음 이 리스너에 있는 다른 규칙의 컨텍스트에서 검토합니다.

▶ 리스너 세부 정보: HTTP:80

## 이름 및 태그 정보

태그를 사용하면 리소스를 관리, 식별, 구성, 검색 및 필터링할 수 있습니다.

Name

redirection to S3

[추가적인 태그 추가](#)

# 실습2) 경로기반 리디렉션 :조건 추가

- 조건추가버튼 클릭 → 경로 선택 → 경로 : /s3

## 규칙 조건 정의 정보

이 규칙에 도달하는 요청은 지정된 모든 조건과 일치해야 규칙을 적용할 수 있습니다. 하나 이상의 조건이 필요합니다.

▶ 리스너 세부 정보: HTTP:80

조건 (0)

조건 없음  
표시할 조건이 없습니다.

조건 추가

### 조건 추가 규칙 제한

#### 규칙 조건 유형

각 요청의 조건 유형에 따라 트래픽을 라우팅합니다. 각 규칙은 host-header, path, http-request-method 및 source-ip 조건 중 하나를 포함할 수 있습니다. 각 규칙은 http-header 및 query-string 조건 중 하나 이상을 포함할 수 있습니다.

경로

경로를 정의합니다. 대/소문자를 구분합니다.

확장하는 데

/s3

유효한 문자는 영문자, 숫자, 하이픈, 밑줄 문자입니다. 경로는 1~128자여야 합니다.

새로운 값 추가

이 규칙의 조건 값을 최대 4개 더 추가할 수 있습니다.

취소

확인

# 실습2) 경로기반 리디렉션 :규칙작업정의

- URL로 리디렉션 → 부분URL → 사용자 지정 호스트, 경로, 쿼리 체크→ 해당 S3 엔드포인트 →경로 /→ 상태코드 302 → 다음

**작업**

작업 유형  
라우팅 액션

☐ 대상 그룹으로 전달 ☒ URL로 리디렉션 ☐ 고정 응답 반환

**URL로 리디렉션 | 정보**

대상 URL에 지정된 URL은 클라이언트에게 리디렉션합니다. HTTPS를 HTTP로 리디렉션할 수 없습니다. 리디렉션할 때, 리디렉션 URL은 대상 URL과 동일하거나, 대상 URL의 기본 경로 구성 요소 중 하나 이상을 수정해야 합니다. 수정하지 않은 구성 요소는 원래 값을 유지합니다.

URI 부분  전체 URL

**프로토콜**  
클라이언트에서 로드 밸런서로 연결하기 위해 사용됩니다.

HTTP

**포트**  
로드 밸런서가 연결을 위해 수신 대기하는 포트입니다.

80

1~65535 또는 원래 포트를 유지하려면 #(port) 입력

☒ 사용자 지정 호스트, 경로, 쿼리를 사용하십시오...  
호스트, 경로 및 쿼리를 수정하려면 선택합니다. 아무런 변경 사항이 발생하지 않은 경우 요청 URL의 설정이 유지됩니다.

**호스트**  
원래 호스트를 유지하거나, 사용자 지정 호스트를 사용하여 리디렉션할 수 있습니다. 대/소문자를 구분하지 않습니다.

sgu-202500-s3-3cs3-website.ap-northeast-2.amazonaws.com

최대 128자까지 입력할 수 있습니다. 허용되는 문자는 a~z, A~Z, 0~9, 특수 문자 -, ., \*, ~, \_ 및 와일드카드(\*) 및 가입니다. 하나 이상의 '.'가 필요합니다. 마지막 '.' 문자 뒤에는 알파벳 문자만 사용할 수 있습니다.

**경로**  
(슬래시)를 사용하여 경로를 지정하거나 원래 경로를 유지합니다. 대/소문자를 구분하지 않습니다.

/

최대 128자까지 입력할 수 있습니다. 허용되는 문자는 a~z, A~Z, 0~9, 특수 문자 -, ., \*, ~, \_ 및 와일드카드(\*) 및 가입니다. 하나 이상의 '.'가 필요합니다. 마지막 '.' 문자 뒤에는 알파벳 문자만 사용할 수 있습니다.

**쿼리 - 선택 사항**  
#(query)를 사용하여 쿼리를 지정하거나 원래 쿼리를 유지합니다. 대/소문자를 구분하지 않습니다.

최대 128개 문자.

**상태 코드**

302 - 찾을 수 없음

취소 이전 **다음**

# 실습2) 경로기반 리디렉션 :규칙 우선 순위 설정

## • 우선 순위 : 1

### 규칙 우선 순위 설정 정보

각 규칙에는 우선 순위가 있습니다. 기본 규칙이 마지막에 평가되며 기본이 아닌 규칙의 우선 순위는 언제든지 변경할 수 있습니다. 기본 규칙의 우선 순위는 변경할 수 없습니다.

▶ 리스너 세부 정보: HTTP:80

#### 리스너 규칙 (2) 정보

규칙 제한

우선 순위 재설정

우선 순위 간 간격 추가 ▼

리스너가 수신한 트래픽은 기본 작업 및 추가 규칙에 따라 라우팅됩니다. 규칙은 가장 낮은 값에서 가장 높은 값까지 우선 순위에 따라 평가됩니다.

이름 태그	우선 순위	조건(인 경우)	작업(다음 수행)	ARN	태그
redirection to S3	<input type="text" value="1"/> <small>우선 순위 값은 1~50,000여야 합니다.</small>	경로 패턴은 /s3입니다	리디렉션 대상 HTTP://sgu-202500-s3-3c.s3-website.ap-northeast-2.amazonaws.com:80/s3? • 상태 코드: HTTP_302	대기 중	1개 태그
기본값	마지막(기본값)	다른 규칙이 적용되지 않는 경우	대상 그룹으로 전달 • <a href="#">sgu-202500-3a-target</a> : 1 (100%) • 대상 그룹 고정성: 끄	ARN	0개 태그

취소

이전

다음

# 실습2) 경로기반 리디렉션 생성!

## • 생성!

검토 및 생성

▶ 리스너 세부 정보: HTTP:80

규칙 세부 정보: redirection to S3

편집 ▼

우선 순위

1

조건(인 경우)

요청이 모두 일치하는 경우:  
경로 패턴은 /s3입니다

작업(다음 수행)

리디렉션 대상 HTTP://sgu-202500-s3-3c.s3-website.ap-northeast-2.amazonaws.com:80/s3?  
• 상태 코드: HTTP\_302

규칙 ARN

대기 중

규칙 태그 (1)

편집

태그를 사용하면 리소스를 관리, 식별, 구성, 검색 및 필터링할 수 있습니다.

키

값

Name

redirection to S3

▶ 서버 측 작업 및 상태

위 단계를 완료하고 제출하면 모든 서버 측 작업과 해당 상태를 모니터링할 수 있게 됩니다.

취소

이전

생성



# 실습 2) 생성확인!

리스너 HTTP:80에 규칙 'redirection to S3'을(를) 생성했습니다.

피드백 제공

HTTP:80 정보

작업 ▼

▼ 세부 정보

리스너는 사용자가 구성된 프로토콜 및 포트를 사용하여 연결 요청을 확인합니다. 생성한 기본 작업 및 추가 규칙에 따라 Application Load Balancer가 요청을 등록된 대상으로 라우팅하는 방법이 결정됩니다.

프로토콜: 포트

HTTP:80

로드 밸런서

[sgu-202500-3a-ALB](#)

기본 작업

대상 그룹으로 전달

- [sgu-202500-3a-target](#) 1 (100%)
- 대상 그룹 고정성: 끄

리스너 ARN

[arn:aws:elasticloadbalancing:ap-northeast-2:443370697536:listener/app/sgu-202500-3a-ALB/1ae599bde6af858a/bafaf05f546f2423](#)

규칙

속성

태그

리스너 규칙 (2) 정보

규칙 제한

작업 ▼

규칙 추가

리스너가 수신한 트래픽은 기본 작업 및 추가 규칙에 따라 라우팅됩니다. 규칙은 가장 낮은 값에서 가장 높은 값까지 우선 순위에 따라 평가됩니다.

규칙 필터링

⚙

<input type="checkbox"/>	이름 태그	우선 순위 ▲	조건(인 경우)	작업(다음 수행)	ARN	태그
<input type="checkbox"/>	<a href="#">redirection to S3</a>	1	경로 패턴은 /s3입니다	리디렉션 대상 HTTP://sgu-202500-s3-3c.s3-website.ap-northeast-2.amazonaws.com:80/s3? <ul style="list-style-type: none"><li>상태 코드: HTTP_302</li></ul>	<a href="#">ARN</a>	<a href="#">1개 태그</a>
<input type="checkbox"/>	<a href="#">기본값</a>	마지막(기본값)	다른 규칙이 적용되지 않는 경우	대상 그룹으로 전달 <ul style="list-style-type: none"><li><a href="#">sgu-202500-3a-target</a> 1 (100%)</li><li>대상 그룹 고정성: 끄</li></ul>	<a href="#">ARN</a>	<a href="#">0개 태그</a>

# 실습 2) 접근확인!

- sgu-202500-3a-ALB-1330996514.ap-northeast-2.elb.amazonaws.com/s3 입력 시, 버킷 index.html 가는지 확인

← → ↺ ⚠ 주의 요함 sgu-202500-s3-3c.s3-website.ap-northeast-2.amazonaws.com/s3/

## Amazon S3 정적 웹사이트 호스팅 실습

이 페이지는 S3 버킷에서 호스팅되고 있습니다.

업로드한 이미지:



## 실습3) Lambda

- Lambda 함수 생성
- /lambda 경로로 들어오는 요청 → 생성한 Lambda 함수로 연결

# Lambda란?

- AWS Lambda: 서버를 직접 관리하지 않고, 코드를 실행할 수 있게 해주는 서비스
  - 서버 필요 없음: EC2 인스턴스처럼 서버 띄우고 관리할 필요 X.
  - 초 단위 과금: 실제 코드가 돌아간 시간만 지불
  - 자동 확장: 호출이 많아지면 Lambda가 자동으로 인스턴스 확장
  - 이벤트 기반 실행: S3 파일 업로드, DynamoDB 변경, API Gateway 호출 등 다양한 이벤트로 동작 가능
  - 다양한 언어 지원 : Python, Node.js, Java, Go 등

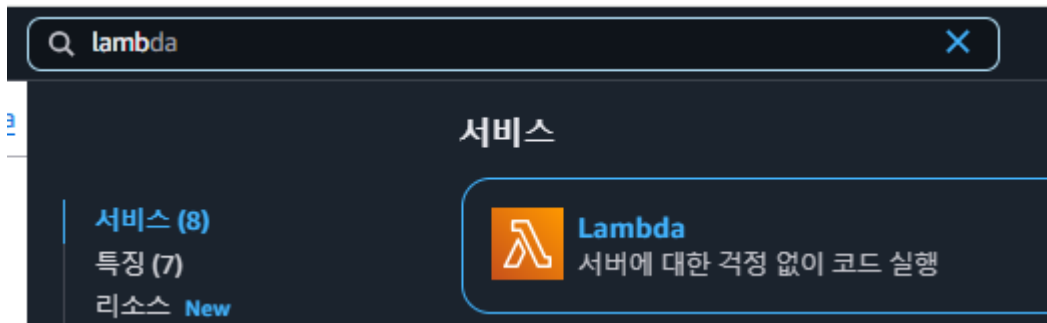
# 자바 Lambda는?

- 이것은 람다식!
- 코드 안에서 짧게 함수를 표현하는 방법
- 메모리나 서버, 인프라랑은 전혀 관계 없음

일반 함수 호출 (자바 기본)	람다식 호출 (Java Lambda)
<pre>public class Main {     public static void main(String[] args) {         greet("Haeri");     }      public static void greet(String name) {         System.out.println("Hello, " + name);     } }</pre>	<pre>@FunctionalInterface interface Greeting {     void sayHello(String name); }  public class Main {     public static void main(String[] args) {         Greeting greet = (name) -&gt;         System.out.println("Hello, " + name);         greet.sayHello("Haeri");     } }</pre>

# 실습3) lambda 생성

- lambda → 함수 생성 클릭



# 실습3) lambda 생성

- 함수이름 : sgu-202500-lambda\_handler / 런타임 : python 3.13
- 기존 역할 : 계정명 검색

**기본 정보**

**함수 이름**  
함수의 용도를 설명하는 이름을 입력합니다.  
  
함수 이름은 1~64자여야 하고, 리전에 고유해야 하며, 공백을 포함할 수 없습니다. 유효한 문자는 a~z, A~Z, 0~9, 하이픈(-), 밑줄(\_)입니다.

**런타임 정보**  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

**아키텍처 정보**  
함수 코드에 대해 원하는 명령 세트 아키텍처를 선택합니다.  
☒ x86\_64  
☐ arm64

**권한 정보**  
기본적으로 Lambda는 Amazon CloudWatch Logs에 로그를 업로드하는 권한을 가진 실행 역할을 생성합니다. 이 기본 역할은 나중에 트리거를 추가할 때 사용자 지정할 수 있습니다.

**▼ 기본 실행 역할 변경**

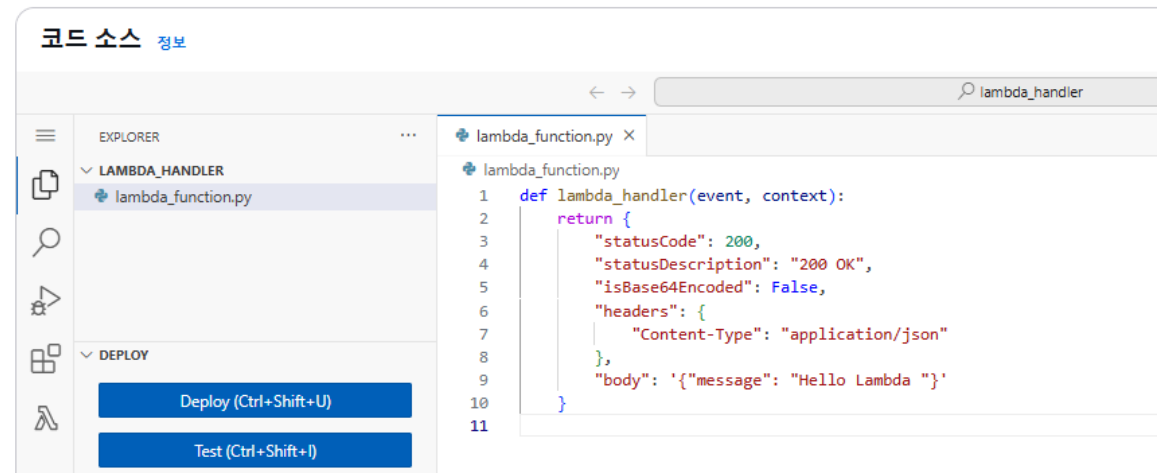
**실행 역할**  
함수에 대한 권한을 정의하는 역할을 선택합니다. 사용자 지정 역할을 생성하려면 [IAM 콘솔](#)로 이동하십시오.  
☐ 기본 Lambda 권한을 가진 새 역할 생성  
☒ 기존 역할 사용  
☐ AWS 정책 템플릿에서 새 역할 생성

**기존 역할**  
생성한 기존 역할 중에 이 Lambda 함수와 함께 사용할 역할을 선택합니다. 이 역할에는 Amazon CloudWatch Logs에 로그를 업로드할 수 있는 권한이 있어야 합니다.

[IAM 콘솔에서 SafeRoleForUser-sgu-202500 역할을 확인](#)합니다.

# 실습3) lambda 생성

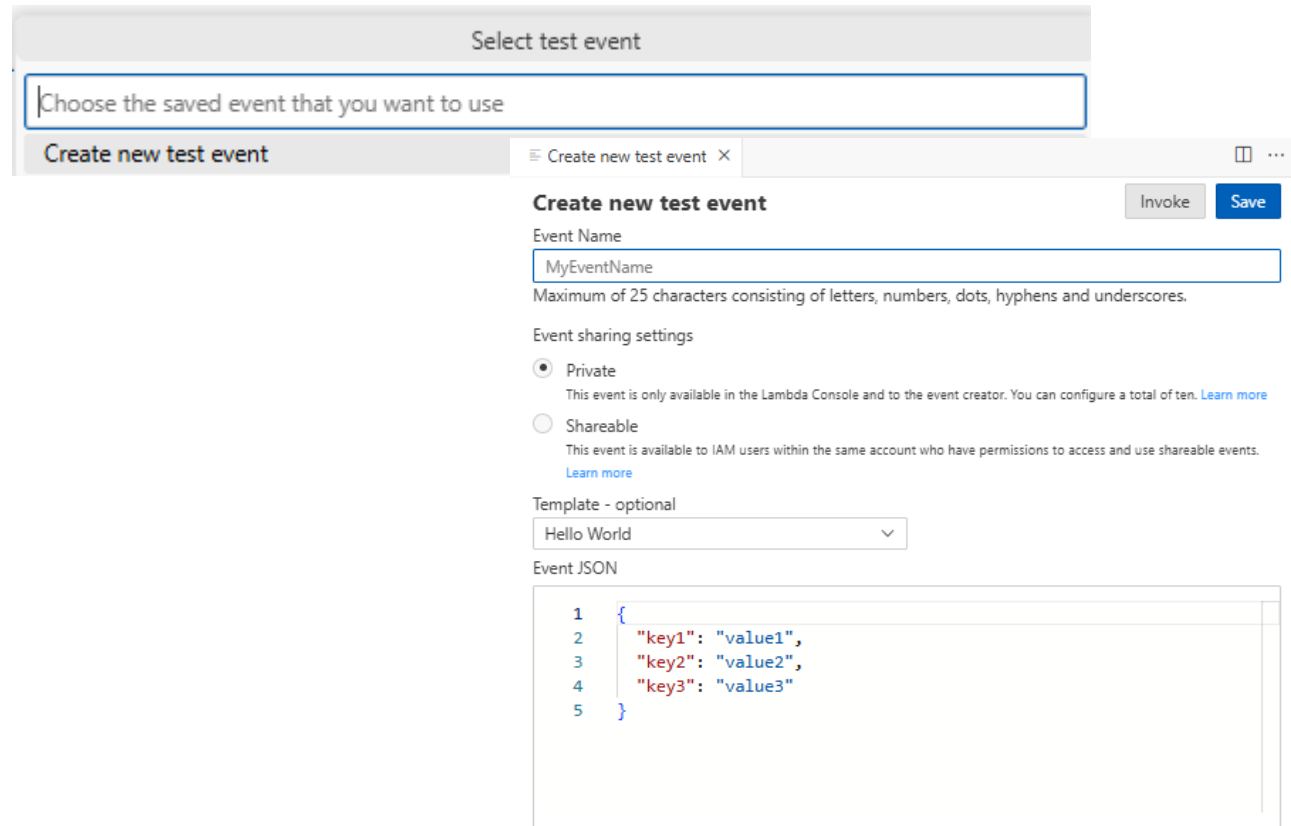
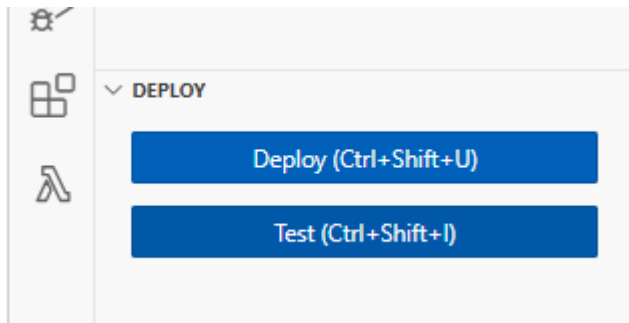
```
def lambda_handler(event, context):  
    return {  
        "statusCode": 200,  
        "statusDescription": "200 OK",  
        "isBase64Encoded": False,  
        "headers": {  
            "Content-Type": "application/json"  
        },  
        "body": '{"message": "Hello Lambda "'  
    }
```





# 실습3) lambda 테스트

- Test 클릭 → Create new test event → event name : sgu-202500-test

A screenshot of the 'Create new test event' form in the AWS Lambda console. The form is titled 'Select test event' and has a text input field with the placeholder 'Choose the saved event that you want to use'. Below this is a button labeled 'Create new test event'. To the right of this button is a tab labeled 'Create new test event' with a close icon. Further right are 'Invoke' and 'Save' buttons. The form contains several sections: 'Event Name' with a text input field containing 'MyEventName' and a note 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.'; 'Event sharing settings' with two radio buttons: 'Private' (selected) and 'Shareable'; 'Template - optional' with a dropdown menu showing 'Hello World'; and 'Event JSON' with a text area containing a JSON object: 

```
{  "key1": "value1",  "key2": "value2",  "key3": "value3"}
```

# 실습3) lambda 테스트

- invoke 클릭 → 정상확인 했으면 → Deploy 클릭!

The screenshot displays the AWS Lambda console interface during a test event creation. The 'Create new test event' dialog is open, showing the event name 'sgu-202500-test' and sharing settings set to 'Private'. The 'Invoke' and 'Save' buttons are highlighted with a red rectangle. The Explorer view on the right shows the 'LAMBDA\_HANDLER' folder containing 'lambda\_function.py'. Below the Explorer, the 'DEPLOY' section contains 'Deploy (Ctrl+Shift+U)' and 'Test (Ctrl+Shift+I)' buttons. The Output view shows the response from the function:

```
Response:
{
  "statusCode": 200,
  "statusDescription": "200 OK",
  "isBase64Encoded": false,
  "headers": {
    "Content-Type": "application/json"
  },
  "body": "{\"message\": \"Hello Lambda \""}
```

Function Logs:

```
START RequestId: aac8299d-3082-46ff-8802-c036467bab63 Version: $LATEST
END RequestId: aac8299d-3082-46ff-8802-c036467bab63
REPORT RequestId: aac8299d-3082-46ff-8802-c036467bab63 Duration: 1.94 ms Billed Duration: 2 ms
Duration: 83.48 ms
```

# 실습3) Lambda Target 생성!

- 로드밸런싱 → 대상 그룹 → 대상 그룹 생성
- 그룹 세부 정보 지정
- Lambda함수 체크!
- 대상그룹이름 : sgu-202500-lambda

## 그룹 세부 정보 지정

로드 밸런서는 요청을 대상 그룹의 대상으로 라우팅하고 대상에 대한 상태 확

### 기본 구성

대상 그룹이 생성된 후에는 이 섹션의 설정을 변경할 수 없습니다.

### 대상 유형 선택

#### ☐ 인스턴스

- 특정 VPC 내의 인스턴스에 대한 로드 밸런싱을 지원합니다.
- [Amazon EC2 Auto Scaling](#)을 사용하여 EC2 용량을 관리하고 크기

#### ☐ IP 주소

- VPC 및 온프레미스 리소스에 대한 로드 밸런싱을 지원합니다.
- 동일한 인스턴스에 있는 여러 IP 주소 및 네트워크 인터페이스로의 라
- 마이크로서비스 기반 아키텍처를 통한 유연성을 제공하여 애플케이
- IPv6 대상을 지원하여 종단 간 IPv6 통신 및 IPv4에서 IPv6로의 NAT를

#### ☒ Lambda 함수

- 단일 Lambda 함수로 라우팅을 지원합니다.
- Application Load Balancer에만 액세스할 수 있습니다.

#### ☐ Application Load Balancer

- Network Load Balancer가 특정 VPC 내에서 TCP 요청을 수락하고 라
- Application Load Balancer로 고정 IP 주소 및 PrivateLink를 손쉽게 사

### 대상 그룹 이름

sug-202500-lambda

하이픈을 포함하여 최대 32자의 영숫자 문자를 사용할 수 있지만 이름이 하이픈으로

# 실습3) Lambda 규칙 추가

- 해당 ALB 클릭 → 리스너 규칙 추가

리스너 및 규칙 | 네트워크 매핑 | 리소스 맵 | 보안 | 모니터링 | 통합 | 속성 | 용량 | 태그

### 리스너 및 규칙 (1/1) 정보

리스너는 구성된 프로토콜 및 포트에서 연결 요청을 확인합니다. 리스너가 수신한 트래픽은 기본 작업 및 기타 추가 규칙에 따라 라우팅됩니다.

🔍 리스너 필터링

<input checked="" type="checkbox"/>	프로토콜: 포트 ▼	기본 작업 ▼	규칙 ▼	ARN ▼	보안 정책 ▼	기본 SSL/TLS ▼
<input checked="" type="checkbox"/>	HTTP:80	대상 그룹으로 전달 <ul style="list-style-type: none"><li>sgu-202500-3a-target <a href="#">🔗</a>: 1 (100%)</li><li>대상 그룹 고정성: 끄</li></ul>	<a href="#">1개 규칙</a>	ARN	해당되지 않음	해당되지 않음

규칙 관리 ▲ 리스너 관리

- 규칙 추가
- 규칙 편집
- 규칙 우선 순위 변경

# 실습3) Lambda 규칙 추가

- 대상 등록 : 생성한 람다 함수 선택

## 대상 등록

이는 대상 그룹을 생성하기 위한 선택적 단계입니다. 그러나 로드 밸런서가 이 대상 그룹으로 트래픽을 라우팅하려면 대상을 등록해야 합니다.

### Lambda 함수

단일 Lambda 함수를 대상으로 지정할 수 있습니다.

☒ Lambda 함수 선택

☐ Lambda 함수 ARN 입력

### Lambda 함수

목록에서 Lambda 함수를 선택하거나 새 함수를 생성하고 목록을 새로 고침하여 선택합니다. [Lambda 콘솔에서 새 Lambda 함수를 생성합니다.](#)

sug-202500-lambda

### 버전

이 대상 그룹에서 사용할 선택한 Lambda 함수의 버전입니다.

\$LATEST

☐ 버전 대신 별칭 사용

함수 별칭은 Lambda 내에서 정의됩니다. 별칭은 단일 버전을 가리킬 수도, 두 버전에 대해 가중치를 지정할 수도 있습니다.

# 실습3) Lambda 규칙 추가

- 이름 : lambda-path-rule

## 규칙 추가 정보

규칙을 정의한 다음 이 리스너에 있는 다른 규칙의 컨텍스트에서 검토합니다.

### ▼ 리스너 세부 정보: HTTP:80

프로토콜: 포트  
HTTP:80

로드 밸런서  
[sgu-202500-alb](#)

#### 리스너 ARN

 arn:aws:elasticloadbalancing:ap-northeast-2:443370697536:listener/app/sgu-202500-alb/9b2c2ea6d37877ft

### ▼ 이름 및 태그 정보

태그를 사용하면 리소스를 관리, 식별, 구성, 검색 및 필터링할 수 있습니다.

#### Name

lambda-path-rule

[추가적인 태그](#)

### 조건 (0) 정보

이 규칙에 도달하는 요청은 지정된 모든 조건과 일치해야 규칙을 적용할 수 있습니다. 하나 이상의 조건이 필요합니다.

조건 없음  
표시할 조건이 없습니다.

[조건 추가](#)

# 실습3) Lambda 규칙 추가

조건 추가

규칙 제한

✕

규칙 조건 유형

각 요청의 조건 유형에 따라 트래픽을 라우팅합니다. 각 규칙은 host-header, path, http-request-method 및 source-ip 조건 중 하나를 포함할 수 있습니다. 각 규칙은 http-header 및 query-string 조건 중 하나 이상을 포함할 수 있습니다.

경로

▼

경로

경로를 정의합니다. 대/소문자를 구분합니다.

확장하는 데

/lambda

🗑

유효한 문자는 a~z, A~Z, 0~9 및 특수 문자입니다. 경로는 1~128자여야 합니다.

새로운 값 추가

이 규칙의 조건 값을 최대 4개 더 추가할 수 있습니다.

취소

확인

# 실습3) Lambda 규칙 추가

## 규칙 우선 순위 설정 정보




각 규칙에는 우선 순위가 있습니다. 기본 규칙이 마지막에 평가되며 기본이 아닌 규칙의 우선 순위는 언제든지 변경할 수 있습니다. 기본 규칙의 우선 순위는 변경할 수 없습니다.

▶ 리스너 세부 정보: HTTP:80

### 리스너 규칙 (2) 정보

규칙 제한  우선 순위 재설정 우선 순위 간 간격 추가 ▼

리스너가 수신한 트래픽은 기본 작업 및 추가 규칙에 따라 라우팅됩니다. 규칙은 가장 낮은 값에서 가장 높은 값까지 우선 순위에 따라 평가됩니다.

	이름 태그	우선 순위 	▼	조건(인 경우)	작업(다음 수행)	ARN	태그
⋮	lambda-path-rule	<div>1</div> <div>우선 순위 값은 1~50,000여야 합니다.</div>		경로 패턴은 /lambda입니다	대상 그룹으로 전달 <ul style="list-style-type: none"><li><a href="#">sgu-202500-LD</a> : 1 (100%)</li><li>대상 그룹 고정성: 끄</li></ul>	대기 중	1개 태그
	기본값	마지막(기본값)		다른 규칙이 적용되지 않는 경우	대상 그룹으로 전달 <ul style="list-style-type: none"><li><a href="#">sgu-202500-LD</a> : 1 (100%)</li><li>대상 그룹 고정성: 끄</li></ul>	 ARN	0개 태그

취소 이전 다음



# 실습3) Lambda 규칙 추가

경로 (1) [정보](#)



If  
경로

확장하는 데  
/lambda

작업 [정보](#)

이러한 작업은 규칙 조건과 일치하는 요청에 적용됩니다.

작업 유형  
라우팅 액션

☒ 대상 그룹으로 전달

☐ URL로 리디렉션

☐ 고정 응답 반환

대상 그룹으로 전달 [정보](#)

대상 그룹을 선택하고 라우팅 가중치를 지정하거나 [대상 그룹을 생성](#) 합니다.

대상 그룹

suq-202500-lambda  
대상 유형: Lambda, IPv4



가중치

1

0-999

백분율

100%

[대상 그룹 추가](#)

최대 4개의 대상 그룹을 더 추가할 수 있습니다.

대상 그룹 고정성 [정보](#)

로드 밸런서가 사용자 세션을 특정 대상 그룹에 바인딩할 수 있도록 합니다. 고정성을 사용하려면 클라이언트가 쿠키를 지원해야 합니다. 사용자 세션을 특정 대상에 바인딩하려면 대상 그룹 속성인 고정성을 켜세요.

☐ 대상 그룹 고정성 켜기

[취소](#)

[다음](#)

# 실습3) Lambda 규칙 추가

## 검토 및 생성

▶ 리스너 세부 정보: HTTP:80

### ▼ 규칙 세부 정보: lambda-path-rule

우선 순위  
1

조건(인 경우)  
요청이 모두 일치하는 경우:  
경로 패턴은 /lambda입니다

작업(다음 수행)  
대상 그룹으로 전달  
• [sgu-202500-LD](#) 1 (100%)  
• 대상 그룹 고정성: 끄

규칙 ARN  
대기 중

### 규칙 태그 (1)

편집

태그를 사용하면 리소스를 관리, 식별, 구성, 검색 및 필터링할 수 있습니다.

키

값

Name

lambda-path-rule

### ▶ 서버 측 작업 및 상태

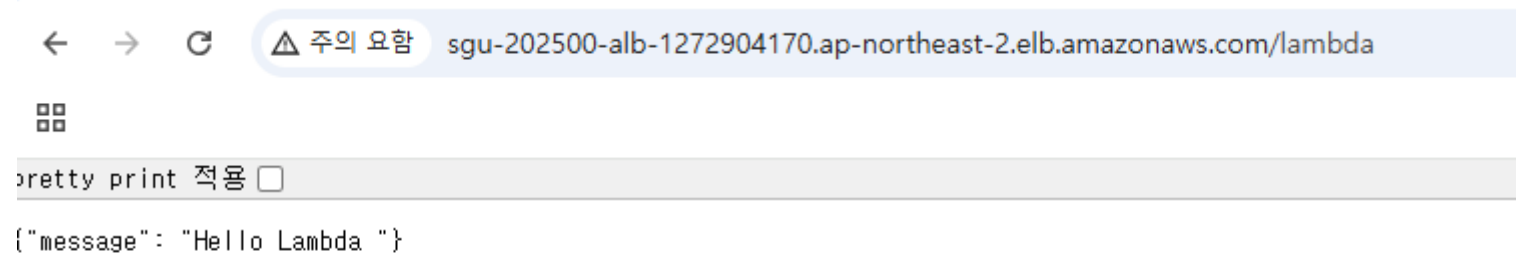
위 단계를 완료하고 제출하면 모든 서버 측 작업과 해당 상태를 모니터링할 수 있게 됩니다.

취소

이전

생성

# 실습3) 확인!



## 실습4) ALB + S3 + Lambda

- ALB도메인/upload 로 접근시, Lambda가 S3에 파일을 업로드 해보자!
- 1. Lambda생성!
- 2. 생성된 Lambda로 타겟 생성
- 3. 생성된 ALB에 리스너 규칙 추가
- 4. ALB도메인/upload 로 접근시, S3에 파일이 업로드 되게 규칙 추가

```
import json          # JSON 형식으로 응답을 구성하기 위한 모듈
import boto3         # AWS 서비스(S3 등)를 사용하기 위한 SDK
from datetime import datetime # 현재 시간 정보를 얻기 위한 모듈
```

```
def lambda_handler(event, context):
    # S3 클라이언트 생성 (S3에 접근하기 위한 boto3 객체)
    s3 = boto3.client('s3')

    # 업로드할 대상 버킷 이름
    bucket_name = 'sgu-202500-3b'

    # 파일을 저장할 경로
    #(S3에서는 디렉터리 개념이 아니라 key prefix)
    prefix = 'uploaded/'

    # 현재 시간을 문자열로 포맷 (파일 이름에 포함시킬 용도)
    now = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')

    # 파일 이름 생성: 예) uploaded/hello_2025-05-07_14-00-00.txt
    filename = f'{prefix}hello_{now}.txt'

    # 파일 내용 구성
    content = f'Hello Haeri! This file was created at {now}'
```

```
# S3에 파일 업로드
# Bucket: 업로드할 버킷 이름
# Key: 파일 경로 및 이름
# Body: 파일의 실제 내용
#(문자열을 UTF-8로 인코딩)
s3.put_object(
    Bucket=bucket_name,
    Key=filename,
    Body=content.encode('utf-8')
)
```

```
# ALB가 Lambda로부터 기대하는 응답 형식
return {
    "statusCode": 200,
    "statusDescription": "200 OK",
    "isBase64Encoded": False,
    바디 인코딩 여부 (파일이 아닌 경우 False)
    "headers": {
        "Content-Type": "application/json"
    },
    "body": '{"message": "업로드 완료"}'
}
```