

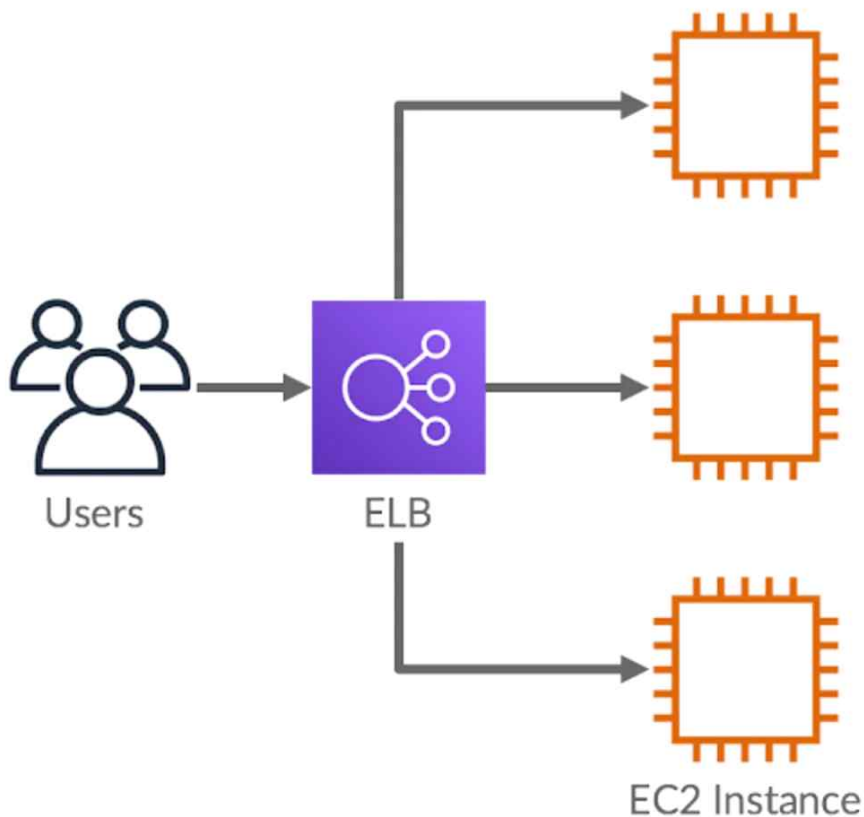
제 목	03장 HTTPS 연결하기 (ELB)	
상세내용	AWS ELB를 활용한 아키텍처 구성	

## 1. ELB란? / TLS, SSL과 HTTPS

### ✓ ELB(Elastic Load Balancer)란?

▶ 한 줄 요약 : 트래픽(부하)을 적절하게 분배해주는 장치이다.

트래픽(부하)를 적절하게 분배해주는 장치를 보고 전문적인 용어로 **로드밸런서(Load Balancer)**라고 부른다. 서버를 2대 이상 가용할 때 ELB를 필수적으로 도입하게 된다.



▶ 하지만 지금은 ELB의 로드밸런서 기능을 사용하지 않고, ELB의 부가 기능인 SSL/TLS(HTTPS)를 적용시키는 방법에 대해 학습할 것이다.

## ✓ SSL/TLS란 ?

SSL/TLS : 쉽게 표현하자면 **\*\*HTTP를 HTTPS로 바꿔주는 인증서\*\***이다.

위에서 말했다시피 **\*\*ELB\*\***는 **\*\*SSL/TLS 기능\*\***을 제공한다고 했다. **\*\*SSL/TLS 인증서를 활용해 HTTP가 아닌 HTTPS로 통신할 수 있게 만들어준다.\*\***

## ✓ HTTPS ?

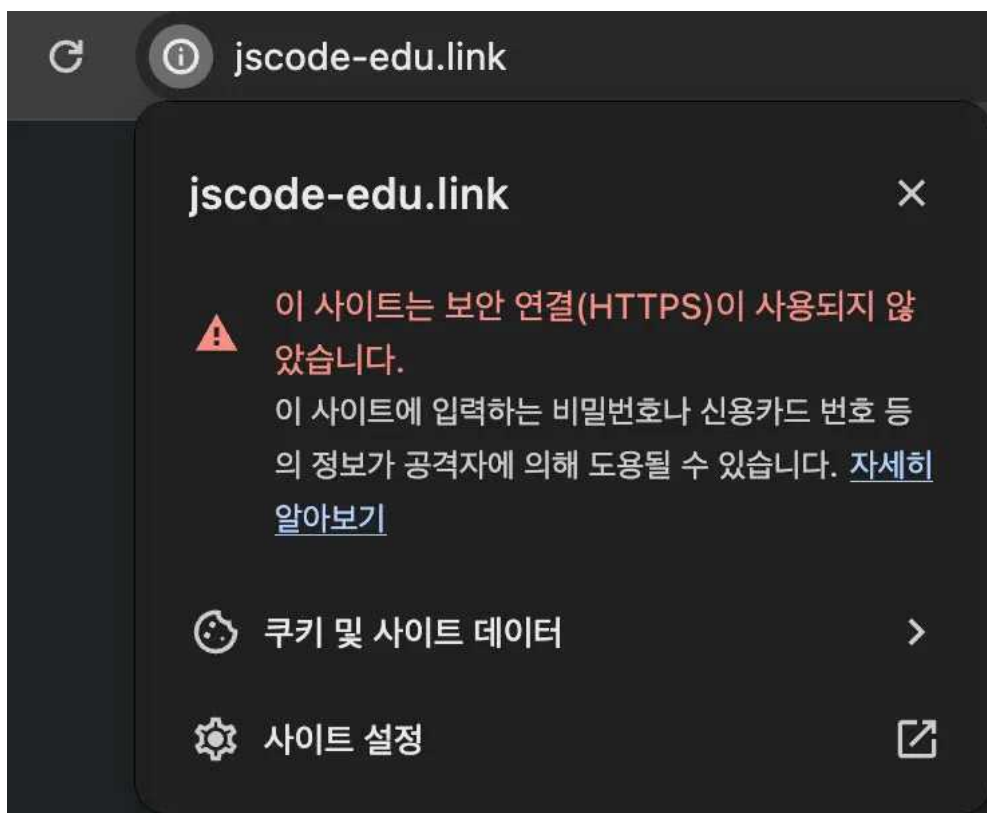
▶ HTTPS를 적용시켜야 하는 이유는 무엇일까?

### 1. 보안적인 이유

데이터를 서버와 주고 받을 때 암호화를 시켜서 통신을 한다. 암호화를 하지 않으면 누군가 중간에서 데이터를 가로채서 해킹할 수도 있다. 보안에 좋지 않다.

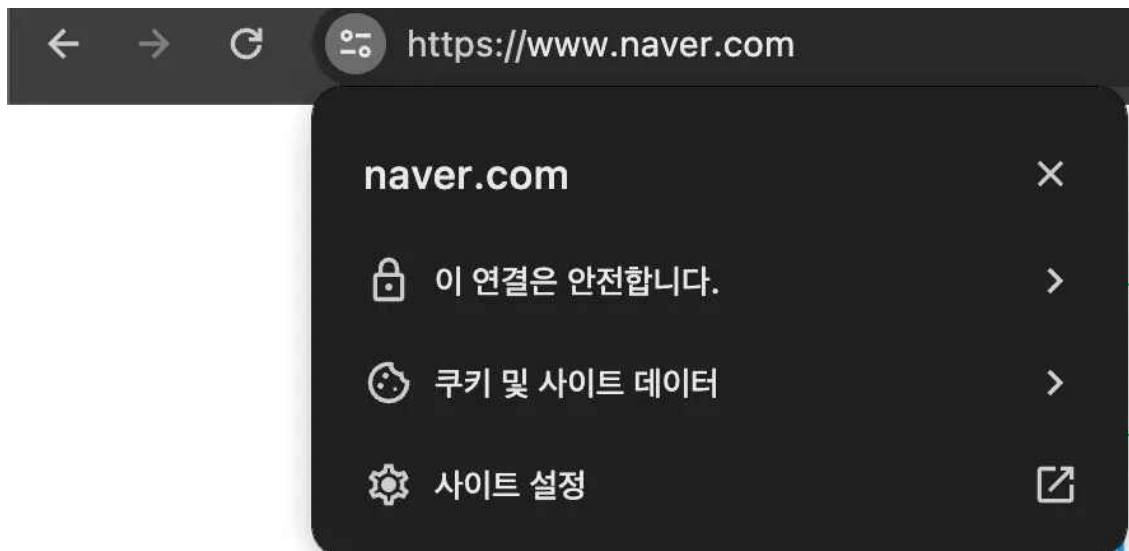
### 2. 사용자 이탈

어떤 사이트에 들어갔는데 아래와 같이 보인다면 왠지 믿음직스럽지 못한 사이트라고 생각할 것이다.



## ✓ 현업에서는 ?

대부분의 웹 사이트에서 HTTPS를 적용시킨다.



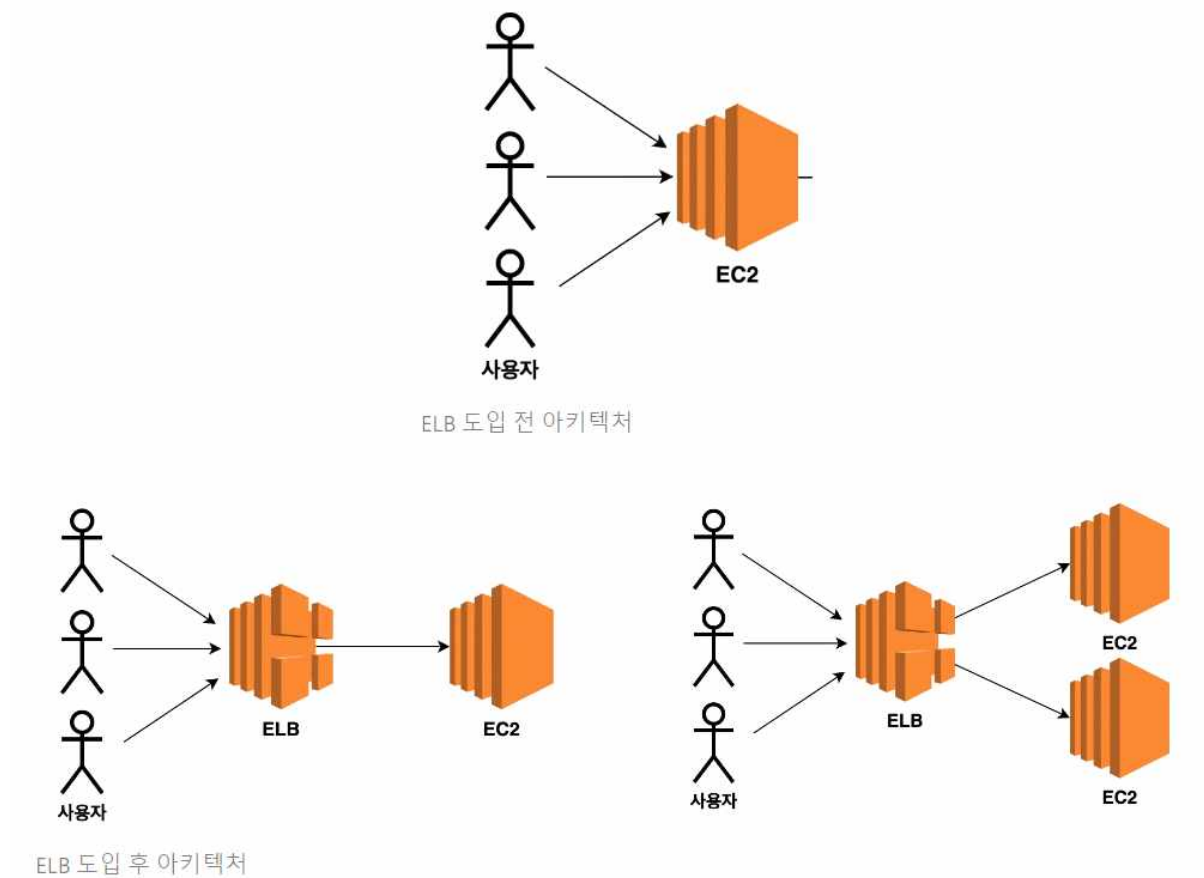
HTTPS 인증을 받은 웹 사이트가 백엔드 서버와 통신하려면, 백엔드 서버의 주소도 HTTPS 인증을 받아야 한다. 따라서 백엔드 서버와 통신할 때도 IP 주소로 통신하는 게 아니라, HTTPS 인증을 받은 도메인 주소로 통신을 한다.

주로 도메인을 구성할 때 아래와 같이 많이 구성한다.

- 웹 사이트 주소 : `https://jsgcode-edu.co.kr`
- 백엔드 API 서버 주소 : `https://api.jsgcode-edu.co.kr`

## 2. ELB를 활용한 아키텍처 구성

### ✓ ELB를 활용한 아키텍처 구성



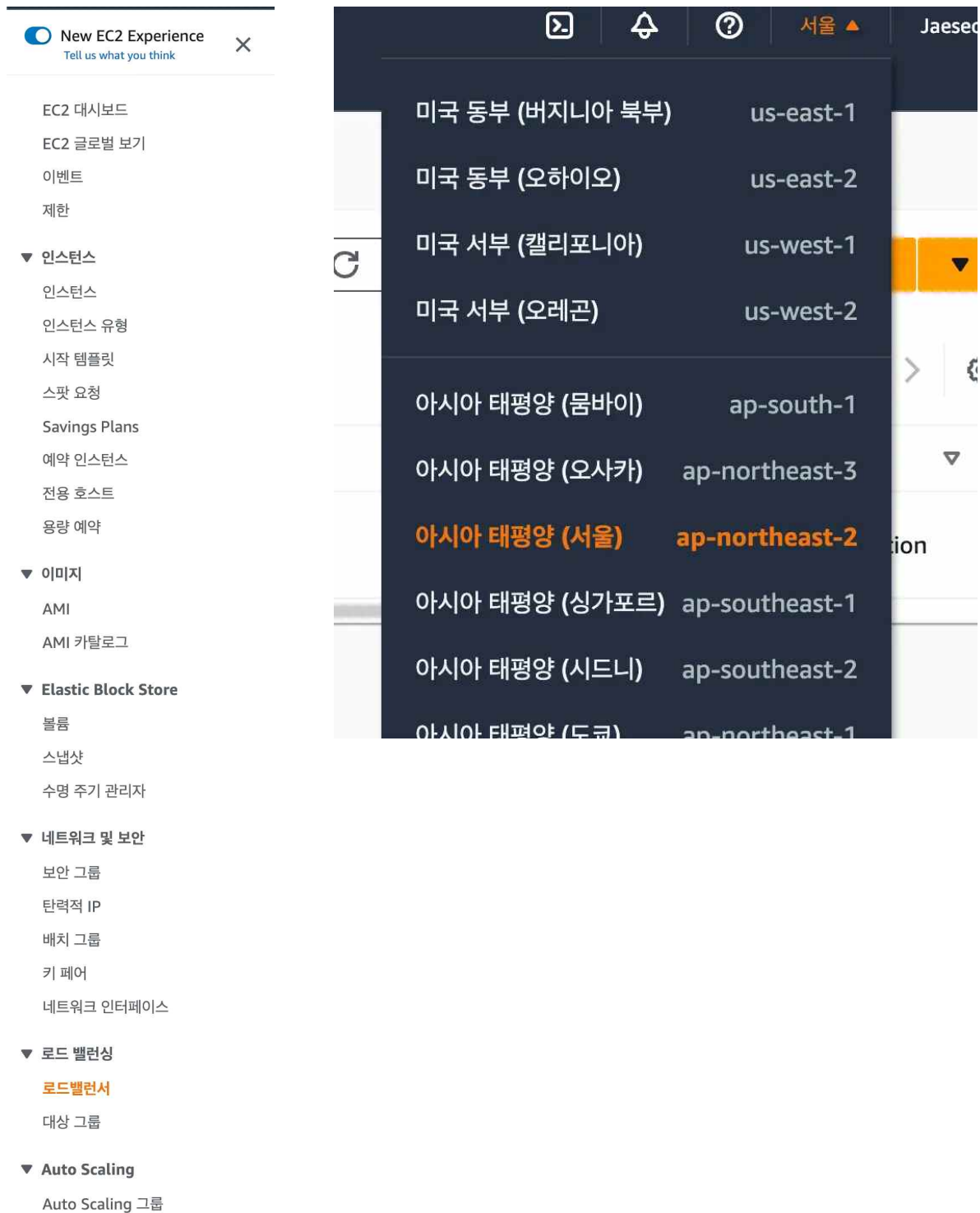
ELB를 사용하기 전의 아키텍처는 사용자들이 EC2의 IP 주소 또는 도메인 주소에 직접 요청을 보내는 구조였다.

하지만 ELB를 추가적으로 도입함으로써 사용자들이 EC2에 직접적으로 요청을 보내지 않고 ELB를 향해 요청을 보내도록 구성할 것이다. 그래서 EC2 달았던 도메인도 ELB에 달 것이고, HTTPS도 ELB의 도메인에 적용시켜 실습할 예정이다.

## [실습] 1. ELB 셋팅하기 - 기본 구성

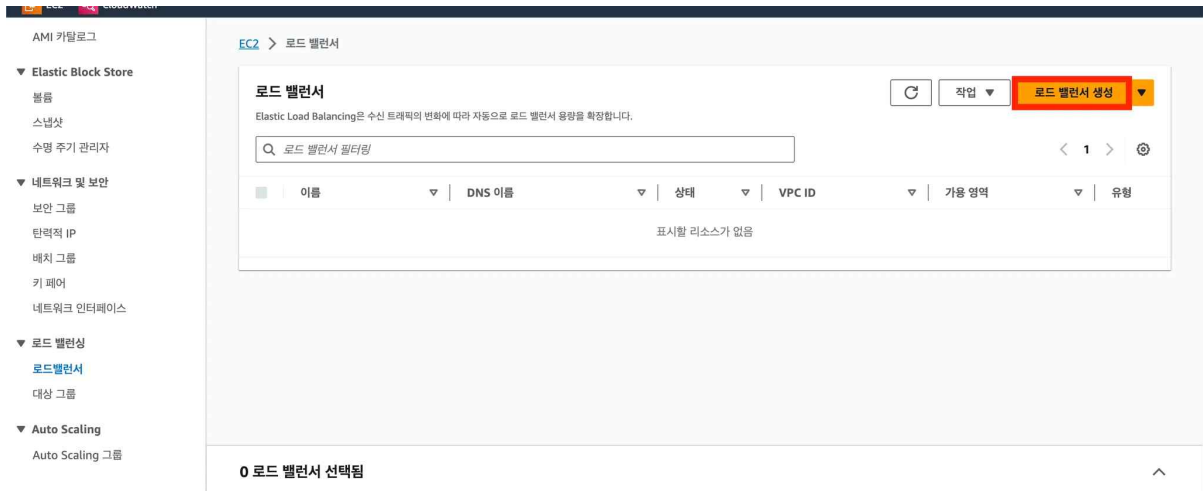
### ✓ 1. 리전 선택하기

**AWS EC2 로드밸런서** 서비스로 들어가서 리전(Region)을 선택해야 한다.



## ✓ 2. 로드 밸런서 유형 선택하기

### 2.1. 로드 밸런서 생성하기



### 2.2. 로드 밸런서 유형 선택하기

3가지 로드 밸런서 유형 중 Application Load Balancer(ALB)를 선택하면 된다

EC2 > 로드 밸런서 > 로드 밸런서 유형 비교 및 선택

### 로드 밸런서 유형 비교 및 선택

자세한 하이라이트와 함께 전체 기능별 비교도 제공됩니다. [자세히 알아보기](#)

#### 로드 밸런서 유형

#### Application Load Balancer 정보

HTTP 및 HTTPS 트래픽을 사용하는 애플리케이션을 위한 유연한 기능이 필요한 경우 Application Load Balancer를 선택합니다. 요청 수준에 따라 작동하는 Application Load Balancer는 마이크로서비스 및 컨테이너를 비롯한 애플리케이션 아키텍처를 대상으로 하는 고급 라우팅 및 표시 기능을 제공합니다.

생성

#### Network Load Balancer 정보

애플리케이션에 초고성능, 대규모 TLS 오프로딩, 중앙 집중화된 인증서 배포, UDP에 대한 지원 및 고정 IP 주소가 필요한 경우 Network Load Balancer를 선택합니다. 연결 수준에서 작동하는 Network Load Balancer는 안전하게 초당 수백만 개의 요청을 처리하면서도 극히 낮은 지연 시간을 유지할 수 있습니다.

생성

#### Gateway Load Balancer 정보

GENEVE를 지원하는 서드 파티 가상 어플라이언스 플릿을 배포 및 관리해야 할 경우 Gateway Load Balancer를 선택합니다. 이러한 어플라이언스를 사용하면 보안, 규정 준수 및 정책 제어를 개선할 수 있습니다.

생성

▶ Classic Load Balancer - 이전 세대

- 6 -

참고) **\*\*Application Load Balancer를 선택한 이유\*\***를 간단하게 들자면 **\*\*HTTP, HTTPS에 대한 특징을 활용하기 위함\*\***이다.

Application Load Balancer, Network Load Balancer, Gateway Load Balancer의 차이를 아는 건 AWS 입문자 입장에서 크게 중요한 부분이 아니다. 그러니 Application Load Balancer를 선택한 이유가 이해되지 않아도 넘어가도 괜찮다

### ✓ 3. 기본 구성

기본 구성

로드 밸런서 이름

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

practice

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

체계 정보

로드 밸런서 생성 후에는 스키마를 변경할 수 없습니다.

☒ 인터넷 경계
 

인터넷 경계 로드 밸런서는 인터넷을 통해 클라이언트의 요청을 대상으로 라우팅합니다. 퍼블릭 서브넷이 필요합니다. [자세히 알아보기](#)

☐ 내부
 

내부 로드 밸런서는 프라이빗 IP 주소를 사용하여 클라이언트의 요청을 대상으로 라우팅합니다.

IP 주소 유형 정보

서브넷이 사용하는 IP 주소 유형을 선택합니다.

☒ IPv4
 

내부 로드 밸런서에 권장합니다.

☐ 듀얼 스택
 

IPv4 및 IPv6 주소를 포함합니다.

- 인터넷 경계와 내부라는 옵션이 있다. 내부 옵션은 Private IP를 활용할 때 사용한다. 입문 강의에서는 VPC, Private IP에 대한 개념을 활용하지 않을 예정이라 인터넷 경계 옵션을 선택하면 된다.
- **\*\*IPv4\*\***와 **\*\*듀얼 스택\*\***이라는 옵션이 있다. IPv6을 사용하는 EC2 인스턴스가 없다면 **\*\*IPv4\*\***를 선택하면 된다. 우리가 만든 EC2 인스턴스는 전부 IPv4로 이루어져 있다.

#### ▶ 참고: IPv4와 IPv6의 차이

IPv4 주소는 `121.13.0.5`와 같은 IP 주소를 의미한다. 그런데 IPv4 주소가 고갈될 것으로 예측하고 IPv6을 추가로 만들어낸다. IPv6은 IPv4보다 훨씬 더 많은 주소 값을 만들어낼 수 있게 구성했다. IPv6의 형태는 `2dfc:0:0:0217:cbff:fe8c:0`와 같다.

## ✓ 4. 네트워크 매핑

로드 밸런서가 어떤 **\*\*가용 영역\*\***으로만 트래픽을 보낼 건지 제한하는 기능이다. 아직 가용 영역에 대한 개념을 배우지 않았다. AWS 입문자한테는 별로 중요한 개념이 아니다.

가용 영역에 제한을 두지 않고 모든 영역에 트래픽을 보내게 설정하자. 즉, **\*\*모든 가용 영역에 다 체크하자.\*\***

**네트워크 매핑**
정보

로드 밸런서는 IP 주소 설정에 따라 선택한 서브넷의 대상으로 트래픽을 라우팅합니다.

**VPC**
정보

대상에 대한 Virtual Private Cloud(VPC)를 선택하거나 [새로운 VPC를 생성할 수 있습니다](#). 인터넷 게이트웨이가 있는 VPC만 선택할 수 있습니다. 로드 밸런서 생성 후에는 선택한 VPC를 변경할 수 없습니다. 대상에 대한 VPC를 확인하려면 [대상 그룹](#)을 참조하세요.

-

vpc-0b80d71048ae406dd  
IPv4: 172.31.0.0/16

↻

**매핑**
정보

가용 영역을 2개 이상 선택하고 영역당 하나의 서브넷을 선택합니다. 로드 밸런서는 이러한 가용 영역의 대상으로만 트래픽을 라우팅합니다. 로드 밸런서 또는 VPC에서 지원하지 않는 가용 영역은 선택할 수 없습니다.

☒
**ap-northeast-2a (apne2-az1)**

서브넷

subnet-0ee07996981dd9ed4

IPv4 주소

AWS에서 할당

☒
**ap-northeast-2b (apne2-az2)**

서브넷

subnet-02c9c9ad11a33fa1d

IPv4 주소

AWS에서 할당

☒
**ap-northeast-2c (apne2-az3)**

서브넷

subnet-045c57a59ff4431be

IPv4 주소

AWS에서 할당

☒
**ap-northeast-2d (apne2-az4)**

서브넷

subnet-0dc902977bf0ee676

IPv4 주소

AWS에서 할당



## [실습] 2. ELB 셋팅하기 - 보안그룹

### ✓ 보안 그룹

#### 1. AWS EC2 보안 그룹에서 보안 그룹 생성하기

**보안 그룹 생성** 정보

보안 그룹은 인바운드 및 아웃바운드 트래픽을 관리하는 인스턴스의 가상 방화벽 역할을 합니다. 새 보안 그룹을 생성하려면 아래의 필드를 작성하십시오.

**기본 세부 정보**

**보안 그룹 이름** 정보

생성 후에는 이름을 편집할 수 없습니다.

**설명** 정보

**VPC** 정보

**인바운드 규칙** 정보

유형	프로토콜	포트 범위	소스	설명 - 선택 사항	작업
HTTP	TCP	80	Anywh... <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	삭제
HTTPS	TCP	443	Anywh... <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	삭제

**아웃바운드 규칙** 정보

유형	프로토콜	포트 범위	대상	설명 - 선택 사항	작업
모든 트래픽	전체	전체	사용자... <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	삭제

**태그 선택 사항**

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

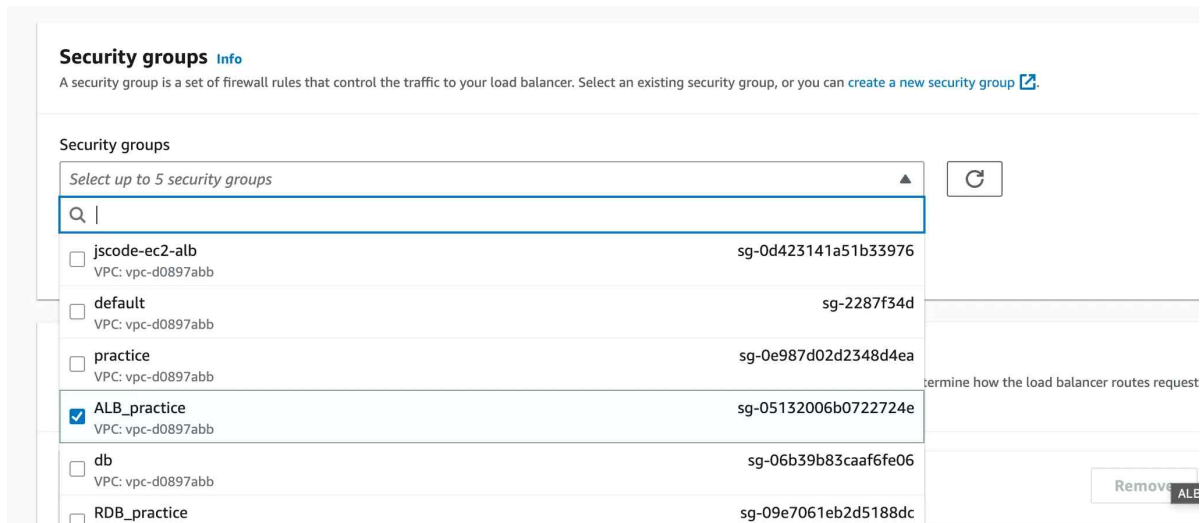
리소스와 연결된 태그가 없습니다.

최대 50개의 태그를 더 추가할 수 있습니다.

[취소](#)

- ELB의 특성상 인바운드 규칙에 **80(HTTP), 443(HTTPS) 포트**로 모든 IP에 대해 요청을 받을 수 있게 설정해야 한다.

## 2. ELB 만드는 창으로 돌아와서 보안 그룹 등록하기



## [실습] 3. ELB 셋팅하기 - 리스너 및 라우팅 / 헬스 체크

### ✓ 1. 대상 그룹(Target Group) 설정하기

리스너 및 라우팅 설정은 ELB로 들어온 요청을 어떤 EC2 인스턴스에 전달할 건지를 설정하는 부분이다.

#### 1. 리스너와 라우팅




ELB로 들어온 요청을 '어떤 곳'으로 전달해야 하는데, 여기서 '어떤 곳'을 **대상 그룹 (Target Group)**이라고 표현한다. 즉, ELB로 들어온 요청을 어디로 보낼 지 **대상 그룹**을 만들어야 한다.

## 2. 대상 유형 선택하기

**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.

☐ **IP 주소**

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

☐ **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 라우팅할 수 있는 유연성을 제공합니다.
- Application Load Balancer로 고정 IP 주소 및 PrivateLink를 손쉽게 사용할 수 있습니다.

EC2에서 만든 특정 인스턴스로 트래픽을 전달할 것이기 때문에 **\*\*인스턴스\*\*** 옵션을 선택한다.

### 3. 프로토콜, IP 주소 유형, 프로토콜 버전 설정

대상 그룹 이름

Practice-taget-group

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

프로토콜 : 포트

HTTP

80

1-65535

IP 주소 유형

표시된 IP 주소 유형의 대상만 이 대상 그룹에 포함될 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 대상에는 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.

-  
vpc-0b80d71048ae406dd  
IPv4: 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.

☐ HTTP2

HTTP/2를 사용하여 대상으로 요청을 전송합니다. 요청 프로토콜이 HTTP/2 또는 gRPC일 때 지원되지만 gRPC 전용 기능은 사용할 수 없습니다.

☐ gRPC

gRPC를 사용하여 대상으로 요청을 전송합니다. 요청 프로토콜이 gRPC일 때 지원됩니다.

ELB가 사용자로부터 트래픽을 받아 대상 그룹에게 어떤 방식으로 전달할 지 설정하는 부분이다. 위 그림은 **HTTP(HTTP1), 80번 포트, IPv4 주소**로 통신을 한다는 걸 뜻한다. 이 방식이 흔하게 현업에서 많이 쓰이는 셋팅 방법이다.

### 4. 상태 검사 설정하기

상태 검사

연결된 Load Balancer가 상태 테스트를 위해 등록된 대상에 아래 설정에 따라 요청을 주기적으로 전송합니다.

상태 검사 프로토콜

HTTP

상태 검사 경로

기본 경로 "/"를 사용하여 루트를 ping하거나 원하는 경우 사용자 지정 경로를 지정합니다.

/health

최대 1024자까지 허용됩니다.

▶ 고급 상태 검사 설정

ELB의 부가 기능으로 **\*\*상태 검사(= Health Check, 헬스 체크)\*\*** 기능이 있다. 이 기능은 굉장히 중요한 기능 중 하나이므로 확실하게 짚고 넘어가자.

실제 ELB로 들어온 요청을 대상 그룹에 있는 여러 EC2 인스턴스로 전달하는 역할을 가진다. ([**ELB**를 활용한 아키텍처 구성](https://www.notion.so/ELB-422ee00730b4445e84c3d1119d4d69c5?pvs=21)) 그런데 만약 특정 EC2 인스턴스 내에 있는 서버가 예상치 못한 에러로 고장났다고 가정해보자. 그럼 ELB 입장에서 고장난 서버한테 요청(트래픽)을 전달하는 게 비효율적인 행동이다.

이런 상황을 방지하기 위해 ELB는 주기적으로(기본 30초 간격) 대상 그룹에 속해있는 각각의 EC2 인스턴스에 요청을 보내본다. 그 요청에 대한 200번대(HTTP Status Code) 응답이 잘 날라온다면 서버가 정상적으로 잘 작동되고 있다고 판단한다. 만약 요청을 보냈는데 200번대의 응답이 날라오지 않는다면 서버가 고장났다고 판단해서, ELB가 고장났다고 판단한 EC2 인스턴스로는 요청(트래픽)을 보내지 않는다.

이러한 작동 과정을 통해 조금 더 효율적인 요청(트래픽)의 분배가 가능해진다.

위에서 설정한 값을 해석해보자면, 대상 그룹의 각각의 EC2 인스턴스에 **`GET /health`**(HTTP 프로토콜 활용)으로 요청을 보내게끔 설정한 것이다. 정상적인 헬스 체크 기능을 위해 EC2 인스턴스에서 작동하고 있는 백엔드 서버에 Health Check용 API를 만들어야 한다. 뒤에서 곧 만들 예정이다.

## 5. 대상 등록하기

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

사용 가능한 인스턴스 (1/1)

인스턴스 필터링

<input checked="" type="checkbox"/>	인스턴스 ID	이름	상태	보안 그룹	영역	프라이빗 IPv4 주소	서브넷 ID
<input checked="" type="checkbox"/>	i-07f254fb255a02954	instagram	🟢 실행 중	launch-wizard-1	ap-northeast-2c	172.31.39.19	subnet-045c57a59ff4431be

1개 선택됨

선택한 인스턴스를 위한 포트  
선택한 인스턴스로 트래픽을 라우팅하기 위한 포트입니다.  
80  
1-65535(필요로 여러 포트 구분)  
**아래에 보류 중인 것으로 포함**

1개의 선택 항목이 현재 아래에 보류 중입니다. 존재가 되면 대상을 더 포함하거나 등록하십시오.

**대상 보기**

대상 (1)

대상 필터링

대기 중인 항목만 보기

보류 중인 모든 항목 제거

제거	상태 확인	인스턴스 ID	이름	포트	상태	보안 그룹	영역	프라이빗 IPv4 주소	서브넷 ID	Launch time
X	대기 중	i-07f254fb255a02954	instagram	80	🟢 실행 중	launch-wizard-1	ap-northeast-2c	172.31.39.19	subnet-045c57a59ff4431be	2023년 11월 13일, 17:16 (UTC+09:00)

1개 대기 중

취소 이전 **대상 그룹 생성**

## 6. ELB 만드는 창으로 돌아와서 대상 그룹(Target Group) 등록하기

**Listeners and routing** [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol

Port

Default action

[Info](#)

HTTP ▼

:

80

1-65535

Forward to

Practice-target-group

HTTP ▼

↻

Target type: Instance, IPv4

Create target group [↗](#)

**Listener tags - optional**

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

위 설정을 해석하자면 ELB에 HTTP를 활용해 80번 포트로 들어온 요청(트래픽)을 설정한 대상 그룹으로 전달하겠다는 의미이다.

## 7. 로드 밸런서 생성하기

나머지 옵션들은 그대로 두고 로드 밸런서를 생성하면 된다.

기

필

선택

새 태그 추가

최대 49개의 태그를 추가할 수 있습니다.

**요약**

구성을 검토하고 확인합니다. [비용 예상](#) [↗](#)

<p><b>기본 구성</b> <a href="#">편집</a></p> <p>practice</p> <ul style="list-style-type: none"> <li>인터넷 경계</li> <li>IPv4</li> </ul>	<p><b>보안 그룹</b> <a href="#">편집</a></p> <ul style="list-style-type: none"> <li>instagram-elb-sg <a href="#">sg-017032463f0c448d5</a> <a href="#">↗</a></li> </ul>	<p><b>네트워크 매핑</b> <a href="#">편집</a></p> <p>VPC <a href="#">vpc-0b80d71048ae406dd</a> <a href="#">↗</a></p> <ul style="list-style-type: none"> <li>ap-northeast-2a <a href="#">subnet-0ee07996981dd9ed4</a> <a href="#">↗</a></li> <li>ap-northeast-2b <a href="#">subnet-02c9c9ad11a33fa1d</a> <a href="#">↗</a></li> <li>ap-northeast-2c <a href="#">subnet-045c57a59ff4431be</a> <a href="#">↗</a></li> <li>ap-northeast-2d <a href="#">subnet-0dc902977bf0ee676</a> <a href="#">↗</a></li> </ul>	<p><b>리스너 및 라우팅</b> <a href="#">편집</a></p> <ul style="list-style-type: none"> <li>HTTP:80 기본값: <a href="#">Practice-target-group</a> <a href="#">↗</a></li> </ul>
<p><b>추가 서비스</b> <a href="#">편집</a></p> <p>없음</p>	<p><b>태그</b> <a href="#">편집</a></p> <p>없음</p>		

**속성**

①

특정 기본 속성이 로드 밸런서에 적용됩니다. 로드 밸런서를 생성한 후 해당 속성을 보고 편집할 수 있습니다.

취소

로드 밸런서 생성

## ✓ 2. Health Check API 추가하기

### ▶ Node.js / express 서버 구축

- 깃허브: <https://github.com/JSCODE-EDU/elb-express-health-sample.git>

#### app.js

```
const express = require('express');
const app = express();
const port = 80;

app.get('/', (req, res) => {
  res.send(`Hello World!`);
})

// GET /health 요청에 대해 상태코드 200으로 응답하는 API
app.get('/health', (req, res) => {
  res.status(200).send("Success Health Check");
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

위의 샘플 프로젝트처럼 ELB의 **상태 검사(= Health Check, 헬스 체크)**에 응답할 수 있는 API를 추가하자. 그런 뒤에 EC2 인스턴스의 서버를 업데이트 시켜주자.

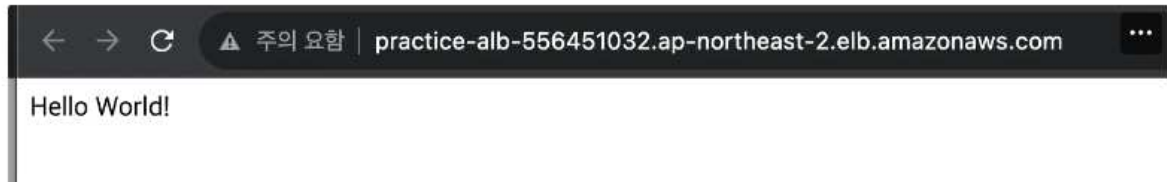
## ✓ 3. 로드밸런서 주소를 통해 서버 접속해보기

EC2 > Load balancers > practice

practice 🔄 Actions ▼

▼ Details

Load balancer type Application	Status 🟢 Active	VPC vpc-d0897abb	IP address type IPv4
Scheme Internet-facing	Hosted zone ZWKZPGT48KDX	Availability Zones subnet-6f48a514 ap-northeast-2b (apne2-az2) subnet-3a9dcc76 ap-northeast-2c (apne2-az3) subnet-1f3e6043 ap-northeast-2d (apne2-az4) subnet-35db365e ap-northeast-2a (apne2-az1)	Date created June 26, 2023, 08:49 (UTC+09:00)
Load balancer ARN arn:aws:elasticloadbalancing:ap-northeast-2:002177417362:loadbalancer/app/practice/e4a4cb9a4f1f7225	DNS name practice-146221799.ap-northeast-2.elb.amazonaws.com (A Record)		



## [실습] 4. ELB에 도메인 연결하기

### ✓ 1. Route 53에서 EC2에 연결되어 있던 레코드 삭제



### ✓ 2. Route 53에서 ELB에 도메인 연결하기

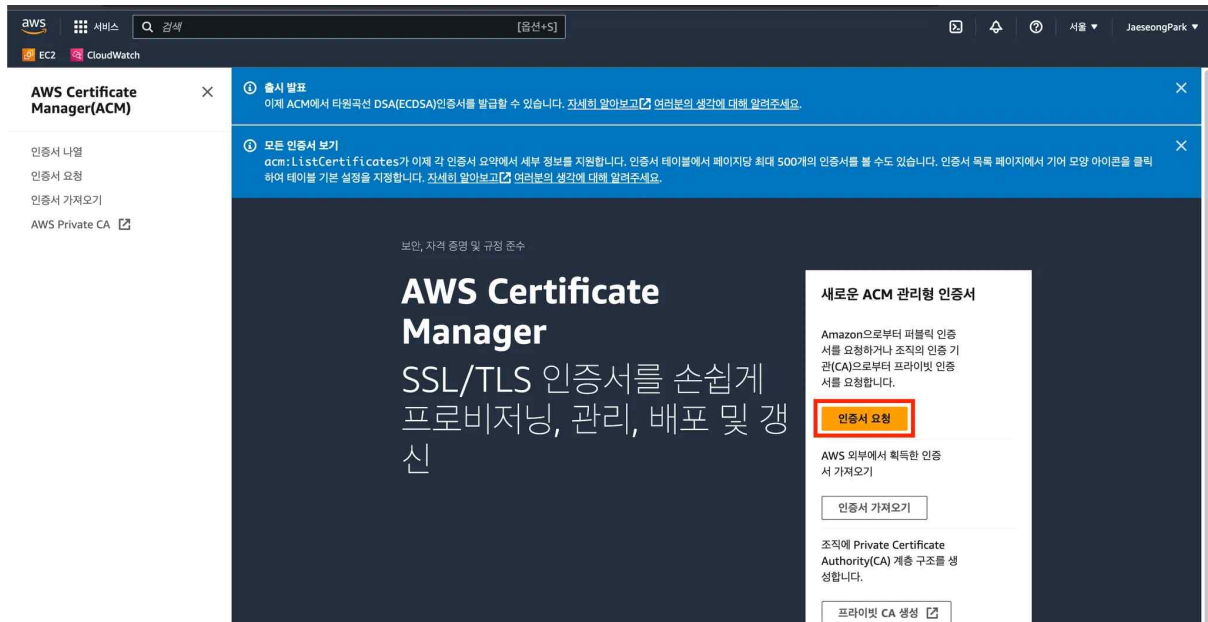




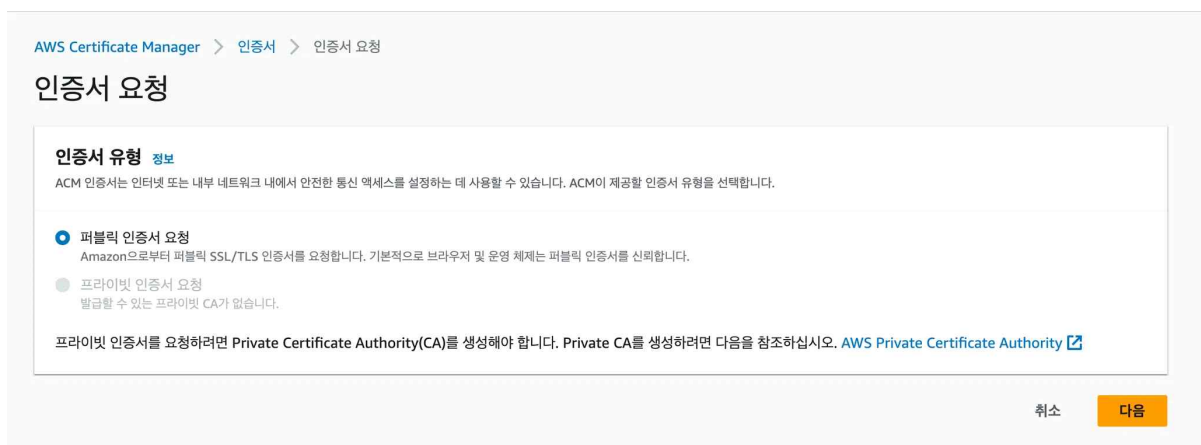
## [실습] 5. HTTPS 적용을 위해 인증서 발급받기

▶ HTTPS를 적용하기 위해서는 인증서를 발급받아야 한다.

✓ 1. AWS Certificate Manager 서비스로 들어가서 **인증서 요청** 버튼 누르기



✓ 2. 인증서 요청하기



AWS Certificate Manager &gt; 인증서 &gt; 인증서 요청 &gt; 퍼블릭 인증서 요청

## 퍼블릭 인증서 요청

## 도메인 이름

인증서에 대해 하나 이상의 도메인 이름을 제공합니다.

완전히 정규화된 도메인 이름 **정보**

practice123.o-r.kr

이 인증서에 다른 이름 추가

이 인증서에 이름을 추가할 수 있습니다. 예를 들어, 'www.example.com'에 대한 인증서를 요청하는 경우 고객이 두 이름 중 하나로 사이트에 접속할 수 있도록 'example.com'이라는 이름을 추가할 수 있습니다.

검증 방법 **정보**

도메인 소유권을 검증하기 위한 방법 선택

☒ DNS 검증 – 권장

인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한이 있는 경우 이 옵션을 선택합니다.

☐ 이메일 검증

인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한을 소유하지 않거나 획득할 수 없는 경우 이 옵션을 선택합니다.

키 알고리즘 **정보**

암호화 알고리즘을 선택합니다. 일부 알고리즘은 일부 AWS 서비스에서 지원되지 않을 수 있습니다.

☒ RSA 2048

RSA는 가장 널리 사용되는 키 유형입니다.

☐ ECDSA P256

암호화 강도는 RSA 3072와 동일합니다.

☐ ECDSA P384

암호화 강도는 RSA 7680와 동일합니다.

태그 **정보**

인증서를 쉽게 관리할 수 있도록 선택적으로 각 리소스에 고유한 메타데이터를 태그 방식으로 지정할 수 있습니다.

이 리소스와 연결된 태그가 없습니다.

태그 추가

태그를 50개 더 추가할 수 있습니다.

취소

이전

요청

### ✓ 3. 인증서 검증하기

내가 소유한 도메인이 맞는 지 검증하는 과정이다.

1.

AWS Certificate Manager > 인증서 > 061e748b-15b3-4c1d-bc08-21ea2176a8cf

#### 061e748b-15b3-4c1d-bc08-21ea2176a8cf

인증서 상태

식별자

061e748b-15b3-4c1d-bc08-21ea2176a8cf

상태

🕒 검증 대기 중 [정보](#)

ARN

arn:aws:acm:ap-northeast-2:961629799183:certificate/061e748b-15b3-4c1d-bc08-21ea2176a8cf

유형

Amazon 발급

도메인 (1)

Route 53에서 레코드 생성

CSV로 내보내기

도메인	상태	경신 상태	유형	CNAME 이름	CNAME 값
jscode-edu.link	🕒 검증 대기 중	-	CNAME	_53fd3cab4fbfac5662e196cdb68fb71.jscode-edu.link.	_9c2328aea1e992c2c14acb5d70a69576.mhbtspbnt.acm-validations.aws.

AWS Certificate Manager > 인증서 > Amazon Route 53에서 DNS 레코드 생성

#### Amazon Route 53에서 DNS 레코드 생성 (1/1)

도메인 검색

1 일치

검증 상태: 검증 대기 중 ✕

검증 상태: 실패 ✕

도메인이 Route 53에 있습니까?: 예 ✕

필터 지우기

<input checked="" type="checkbox"/>	도메인	검증 상태	유형	CNAME 이름	CNAME 값	도메인이 Route 53에 있습니까?
<input checked="" type="checkbox"/>	jscode-edu.link	🕒 검증 대기 중	CNAME	_53fd3cab4fbfac5662e196cdb68fb71.jscode-edu.link.	_9c2328aea1e992c2c14acb5d70a69576.mhbtspbnt.acm-validations.aws.	예

취소

레코드 생성

2. 검증 완료

**3분 정도** 기다렸다가 AWS Certificate Manager 창을 새로고침하면 아래와 같이 검증이 완료된다. (길게는 10분 정도 소요될 때도 있다.)

도메인 (1)

Route 53에서 레코드 생성

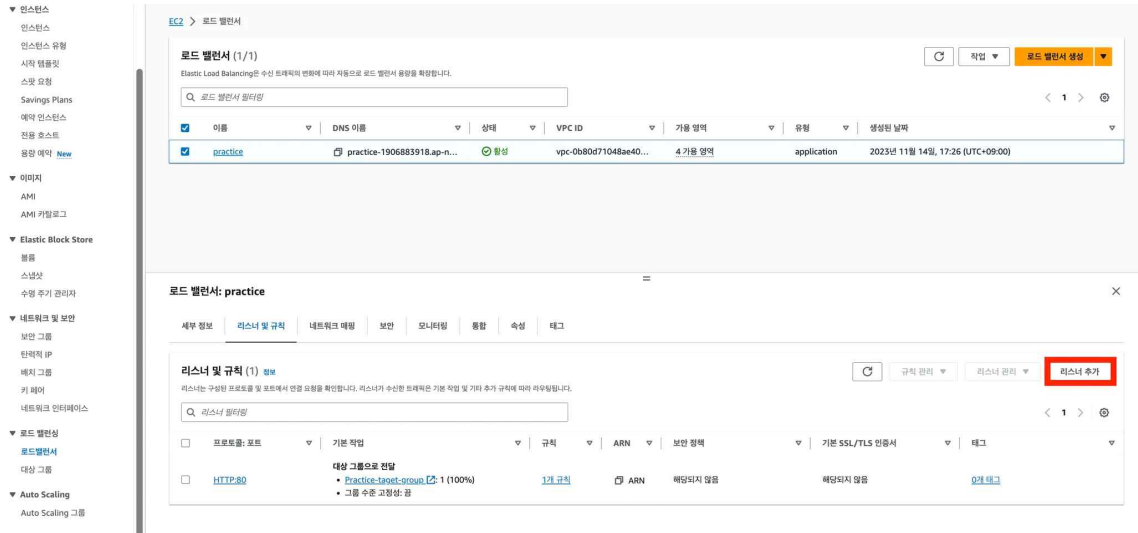
CSV로 내보내기

도메인	상태	경신 상태	유형	CNAME 이름	CNAME 값
practice123.o-r.kr	✅ 성공	-	CNAME	_1990974f1b39d86529694f60c69ef01d.practice123.o-r.kr.	_3d78c0c9b63e00ebd9dfda06bfa9977a.mrhcxwpsky.acm-validations.aws.

## [실습] 6. ELB에 HTTPS 설정하기

### ✓ 1. ELB의 리스너 및 규칙 수정하기

#### 1. HTTPS에 대한 리스너 추가하기



#### 리스너 세부 정보: HTTPS:443

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.

#### 리스너 구성

리스너는 프로토콜 및 포트 식별됩니다.

프로토콜	포트
클라이언트에서 로드 밸런서로 연결하기 위해 사용됩니다.	로드 밸런서가 연결을 위해 수신 대기하는 포트입니다.
HTTPS	443
	1-65535

#### 기본 작업 정보

다른 규칙이 적용되지 않는 경우 기본 작업이 사용됩니다. 이 리스너의 트래픽에 대해 기본 작업을 선택하세요.

#### 인증 정보

☐ OpenID 또는 Amazon Cognito 사용

OpenID Connect(OIDC) 또는 Amazon Cognito를 사용한 인증을 포함합니다.

#### 라우팅 액션

☒ 대상 그룹으로 전달
 ☐ URL 리디렉션
 ☐ 고정 응답 반환

#### 대상 그룹으로 전달 정보

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

#### 대상 그룹

대상 그룹	가중치	백분율
Practice-target-group 대상 유형: 인스턴스, IPv4	1 0-999	100%

[대상 그룹 추가](#)

최대 4개의 대상 그룹을 더 추가할 수 있습니다.

#### 그룹 수준 고정성 정보

대상 그룹에 고정성이 설정되어 있는 경우, 해당 그룹에 라우팅된 요청은 세션 기간 동안 대상 그룹에 유지됩니다. 개별 대상 고정성은 대상 그룹의 구성 옵션입니다.

☐ 그룹 수준 고정 활성화

**보안 리스너 설정** 정보

**보안 정책**  
로드 밸런스는 보안 정책이라고 하는 Secure Socket Layer(SSL) 협상 구성을 사용해 클라이언트와의 SSL 연결을 협상합니다. [보안 정책 비교](#)

ELBSecurityPolicy-TLS13-1-2-2021-06 (권장)

**Default SSL/TLS server certificate**  
클라이언트가 SNI 프로토콜 없이 연결되거나 일치하는 인증서가 없는 경우에 사용되는 인증서입니다. 이 인증서는 리스너 인증서 목록에 자동으로 추가됩니다.

**Certificate source**

☒ ACM에서 ☐ IAM에서 ☐ Import certificate

**Certificate (from ACM)**  
The selected certificate will be applied as the default SSL/TLS server certificate for this load balancer's secure listeners.

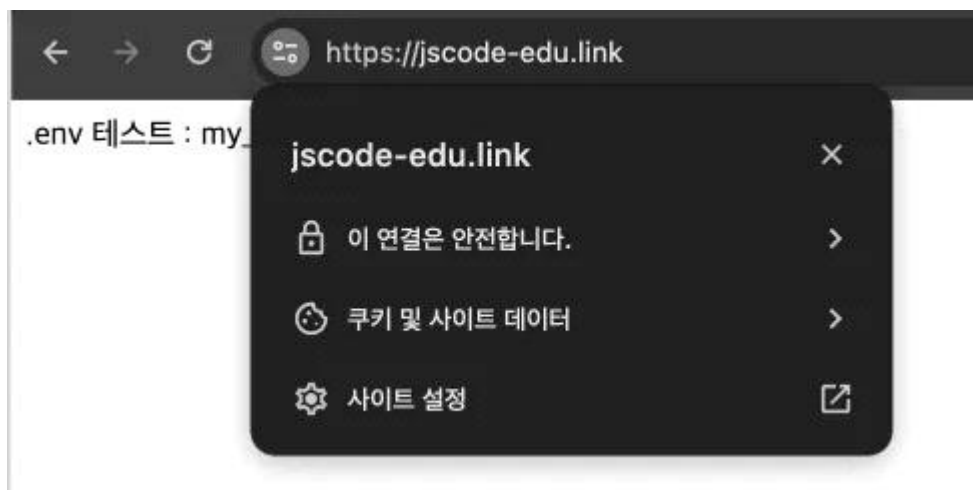
jscode-edu.link  
39758945-0190-44ff-9423-8417...

[새 ACM 인증서 요청](#)

위와 같이 설정하면 HTTPS가 한 5초 정도 있다가 바로 적용된다.

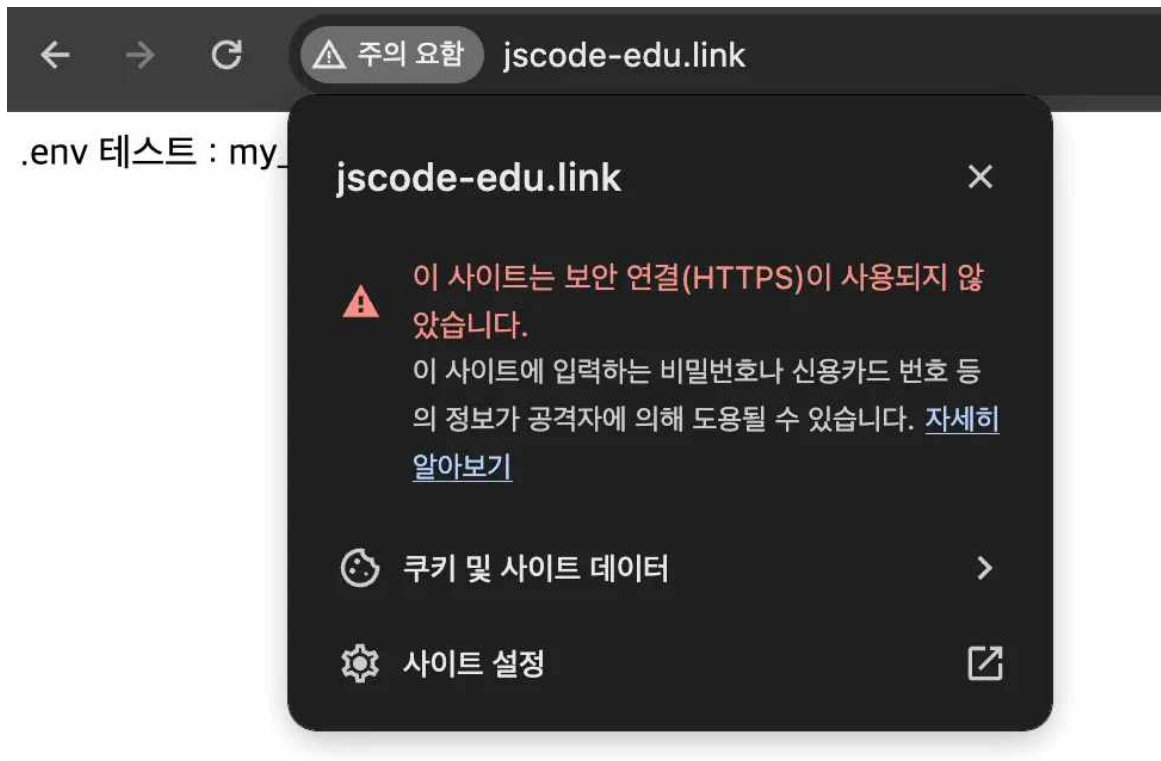
## ✓ 2. HTTPS가 잘 적용됐는 지 확인하기

구매한 도메인에 https를 붙여서 접속해보자.

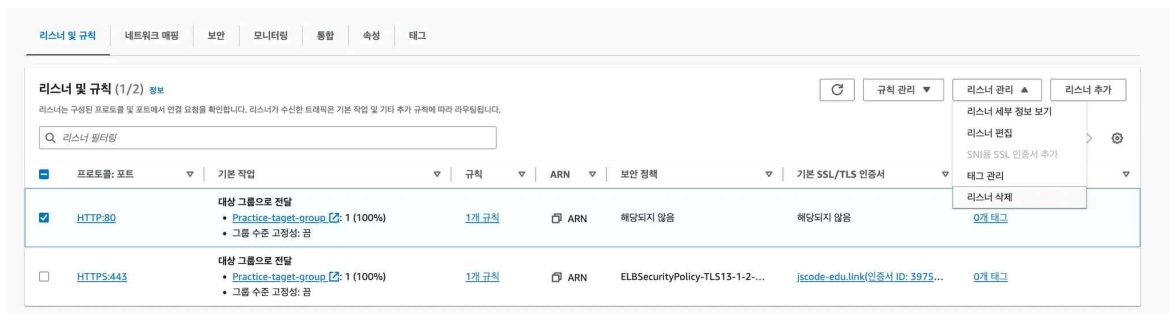


## ✓ 3. HTTP로 접속할 경우 HTTPS로 전환되도록 설정하기

아직까지 아쉬운 점은 http를 붙여서 접속할 경우 HTTPS를 사용하지 않고 접속이 가능하다는 점이다. http를 붙여서 접속하더라도 자동으로 HTTPS로 전환(Redirect)되도록 만들어보자.



## 1. 기존 HTTP:80 리스너를 삭제하기



## 2. 리스너 추가하기

**리스너 세부 정보: HTTP:80**

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.

**리스너 구성**

리스너는 프로토콜 및 포트로 식별됩니다.

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

HTTP

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

80

1-65535

**기본 작업** | 정보

다른 규칙이 적용되지 않는 경우 기본 작업이 사용됩니다. 이 리스너의 트래픽에 대해 기본 작업을 선택하세요.

**라우팅 액션**

☐ 대상 그룹으로 전달
☒ URL로 리디렉션
☐ 고정 응답 반환

**URL로 리디렉션** | 정보

특정 URL에서 다른 URL로 클라이언트 요청을 리디렉션합니다. HTTPS를 HTTP로 리디렉션할 수 없습니다. 리디렉션 루프를 방지하려면 프로토콜, 포트, 호스트 이름 또는 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.

HTTPS

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

443

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.

**상태 코드**

301 - 영구 이동됨

### ✓ 4. HTTP로 접속해도 HTTPS로 잘 전환되는 지 확인하기

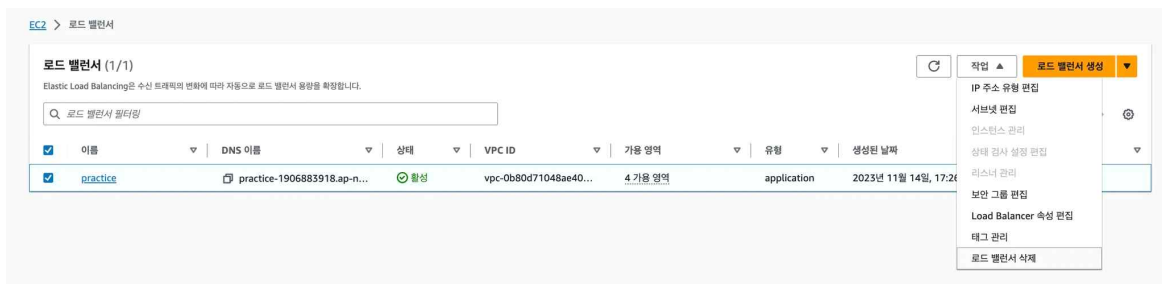


## [주의] 비용 나가지 않게 ELB 깔끔하게 종료하기

### ✔ 비용 나가지 않게 ELB 깔끔하게 종료하기

생성한 ELB와 ELB에 연결한 EC2만 깔끔하게 정리하면 요금이 나가지 않는다.

#### 1. ELB 종료하기



#### 2. EC2 종료하기

EC2 서버도 사용하지 않을 경우 종료한다.

## [보충 강의] HTTPS 연결 시 ELB vs Nginx, Certbot

### ✔ HTTPS 연결 시 ELB vs Nginx, Certbot

HTTPS를 연결하는 방법을 검색해보면 ELB 뿐만 아니라 Nginx, Certbot을 활용하는 방법도 많이 나온다.

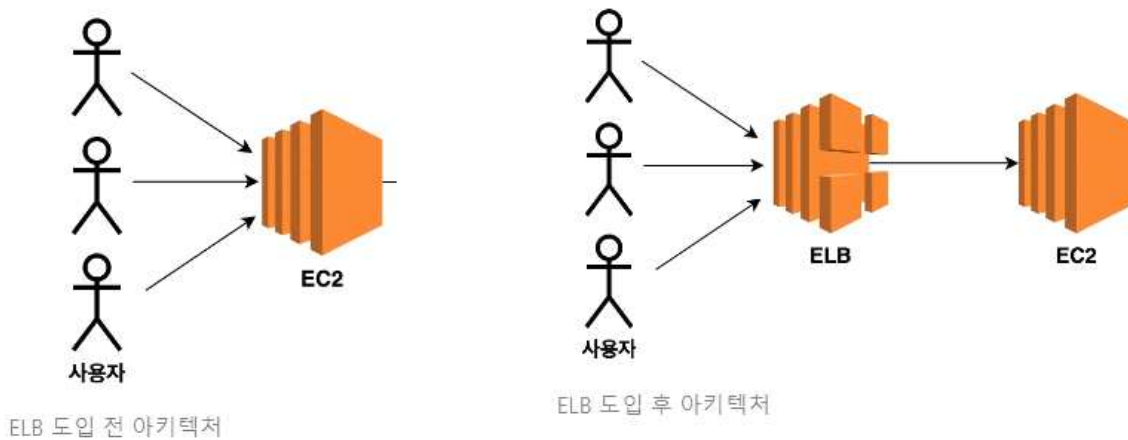
현업에서는 어떤 방법을 더 많이 사용할까?

**현업에서는 ELB를 활용해서 HTTPS 적용을 더 많이 시킨다.** HTTPS 설정도 쉬울 뿐더러 HTTPS 인증서의 만료기간 갱신도 자동으로 해주기 때문이다.

**그러면 Nginx는 왜 사용하는 걸까?**

Nginx와 Certbot을 활용해서 HTTPS를 적용시키는 **가장 큰 이유는 비용 때문**이다. HTTPS를 적용하는 데 일절 비용이 들지 않기 때문이다. 반면 ELB는 사용하는 것 자체로써 비용이 나간다. 따라서 비용이 부담스러운 기업 또는 학생 입장에서는 ELB를 사용하지 않고 백엔드 서버와 Nginx, Certbot를 하나의 EC2에 설치해서 사용하는 경우도 많다.





## [보충 강의] Nginx, Certbot을 활용해 HTTPS 연결하기

### ✓ 0. 사전 환경 셋팅하기

#### ▶ EC2 기본 세팅 및 Route53 도메인 연결

### ✓ 1. Express 서버의 실행 포트를 3000번으로 바꿔주기

Nginx를 80번 포트에서 실행시킬 것이기 때문에 Express 서버는 3000번에서 띄워주도록 하자.

**app.js**

```
require('dotenv').config();
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send(`.env 테스트 : ${process.env.DATABASE_NAME}`);
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

테스트를 위해 보안 그룹에서 3000번 포트를 추가적으로 열어주자.

## ✓ 2. Nginx 설치

```
$ sudo apt update
$ sudo apt install nginx
```

## ✓ 3. Nginx 잘 설치됐는 지 확인

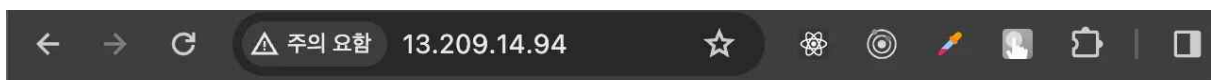
[리눅스 명령어로 확인]

```
$ sudo service nginx status
```

```
ubuntu@ip-172-31-13-138:~$ service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-11-24 06:11:55 UTC; 1min 41s ago
     Docs: man:nginx(8)
  Process: 3696 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 3697 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 3790 (nginx)
    Tasks: 2 (limit: 1121)
   Memory: 4.7M
      CPU: 25ms
  CGroup: /system.slice/nginx.service
          └─3790 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
            └─3793 "nginx: worker process"

Nov 24 06:11:55 ip-172-31-13-138 systemd[1]: Starting A high performance web server and a reverse proxy server...
Nov 24 06:11:55 ip-172-31-13-138 systemd[1]: Started A high performance web server and a reverse proxy server.
```

[EC2 IP로 접속해서 확인]



# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

**주의)** <https://<ip 주소>>가 아니라 반드시 <http://<ip 주소>>로 접속해야 한다.

#### ✓ 4. Certbot 설치하기

```
$ sudo snap install --classic certbot
$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

#### ✓ 5. SSL 인증서 발급받기

```
$ sudo certbot --nginx -d <도메인 주소>

# 예시
$ sudo certbot --nginx -d chungbuk.link
```

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): qkrwotjd1445@naver.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: Y
Account registered.

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/jscode-edu.link/fullchain.pem
Key is saved at: /etc/letsencrypt/live/jscode-edu.link/privkey.pem
This certificate expires on 2024-02-22.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for jscode-edu.link to /etc/nginx/sites-enabled/default
Congratulations! You have successfully enabled HTTPS on https://jscode-edu.link
We were unable to subscribe you the EFF mailing list because your e-mail address appears to be invalid. You can try again later by visiting https://a
ct.eff.org.

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
```

**\*\* 반드시 해당 EC2에 도메인을 연결한 뒤에 위 명령어를 쳐야 정상 작동한다.**

## ✓ 6. 리버스 프록시 설정해주기

```
$ sudo vi /etc/nginx/sites-available/default
```

/etc/nginx/sites-available/default

```
...
server {
    ...

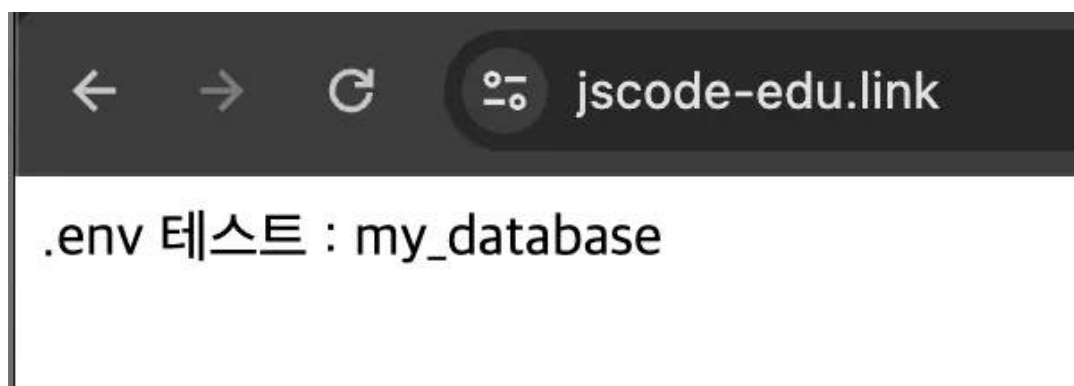
    server_name jscore.edu.link;

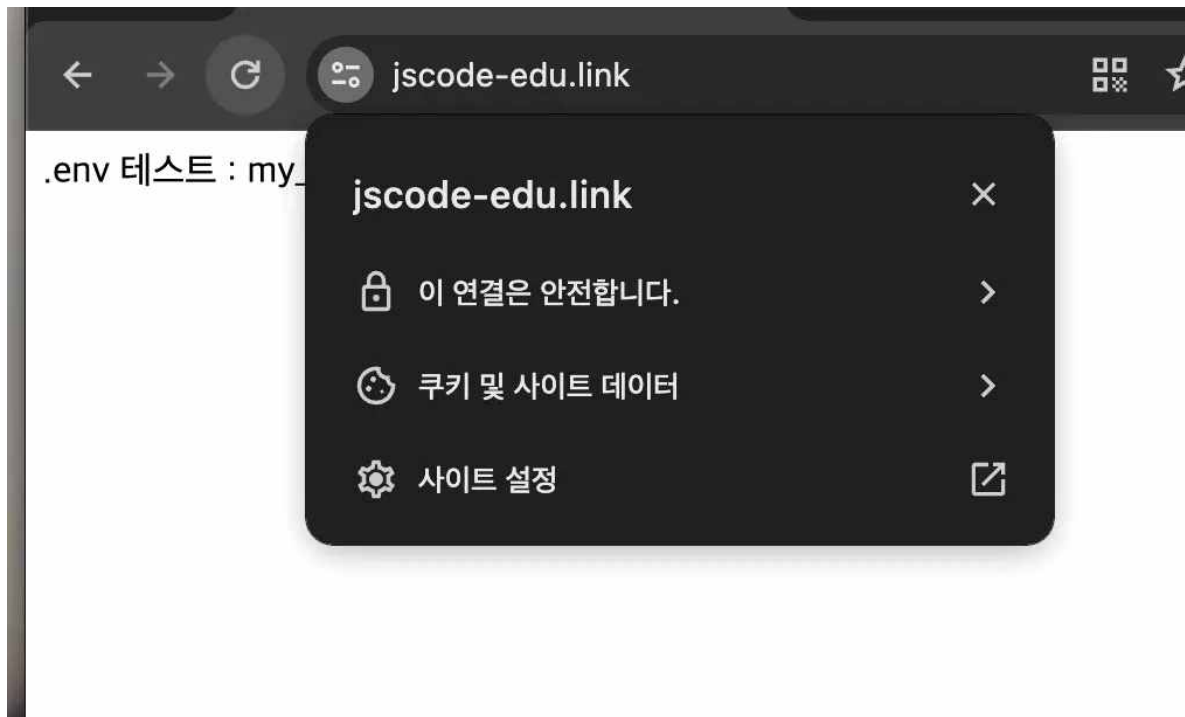
    location / {
        try_files $uri $uri/ =404;
        proxy_pass http://localhost:3000/;
    }
    ...
}
```

## ✓ 7. Nginx 재시작하기

```
$ sudo service nginx restart
```

## ✓ 8. 백엔드 서버에 HTTPS가 잘 적용되는 지 확인하기





### ✓ 참고 문서

- ▶ <https://tinyurl.com/2batxuu6>
- ▶ <https://certbot.eff.org/instructions?ws=nginx&os=ubuntufocal>