

# 주요 CSP NLB 서비스 및 인터페이스 분석

2022.05.12. CB-Spider Leader.

- AWS 중심으로 분석 후 타 CSP도 제공할 수 있는 공통 기능을 도출하기 위하여 GCP, Azure 분석
- GCP, Azure는 AWS NLB 개념 습득 후 Console을 통한 실습 중심의 차이점 위주 분석
- 분석 결과를 활용하여 CB-Spider NLB 규격 및 Driver API 정의

※ 참고: <https://github.com/cloud-barista/cb-spider/wiki/Network-Load-Balancer-and-Driver-API>

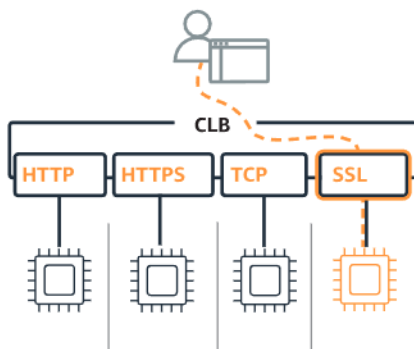
## 목 차

|           |    |
|-----------|----|
| ### AWS   | 2  |
| ### GCP   | 17 |
| ### Azure | 19 |

### ### AWS

- ELB(Elastic Load Balancing) 제공 기능 및 특징
  - Multi Target을 대상으로 input traffic 자동 분산 (in one or more AZ)
    - ◆ Target: **EC2 instance(VM)**, Container 및 IP Address 등
    - ◆ Target의 **동적 추가/삭제 가능**
  - 모니터링 the health of its registered Targets
  - Healthy Target으로 트래픽을 라우팅
    - ◆ App의 availability and fault tolerance 증가
  - Incoming traffic 변화에 따른 LB(Load Balancer) Scaling 제공
    - ◆ 대부분의 Workload에 맞는 auto scaling
- ELB가 제공하는 LB(Load Balancer) 종류

■ Classic LB: **Be retiring on 2022/04/15(금)**



- Application LB(**ALB**)
- **Network LB(NLB)** □ 연동 대상
- Gateway LB

## Elastic Load Balancing - Version 1

### User Guide for Classic Load Balancers

Use Classic Load Balancers with applications in the EC2-Classic network.

[HTML](#) | [PDF](#) | [Kindle](#) | [GitHub](#)

## Elastic Load Balancing - Version 2

## #VPC에 의존

### User Guide for Application Load Balancers

Use Application Load Balancers for HTTP and HTTPS traffic. The load balancer routes based on the content of the request.

[HTML](#) | [PDF](#) | [Kindle](#) | [GitHub](#)

### User Guide for Network Load Balancers

Use Network Load Balancers for TCP, UDP, and TLS traffic where extreme performance is required.

[HTML](#) | [PDF](#) | [GitHub](#)

### User Guide for Gateway Load Balancers

Use Gateway Load Balancers to deploy, scale, and manage virtual appliances, such as firewalls.

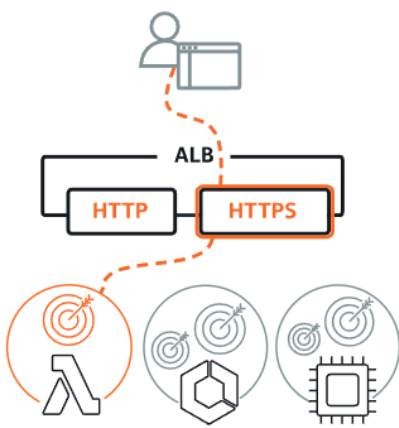
[HTML](#) | [PDF](#) | [GitHub](#)

ref) <https://docs.aws.amazon.com/elasticloadbalancing/index.html>

- 참고: 'Create Load Balancer' 메뉴 클릭 시 제공

### Load balancer types

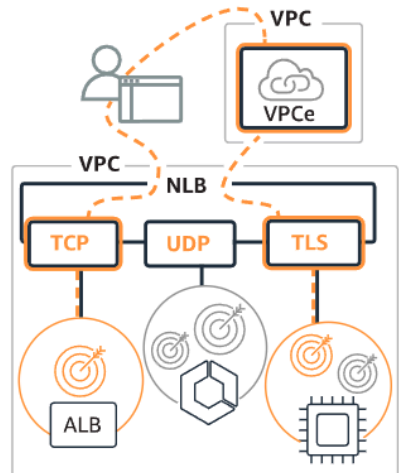
#### Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create


#### Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

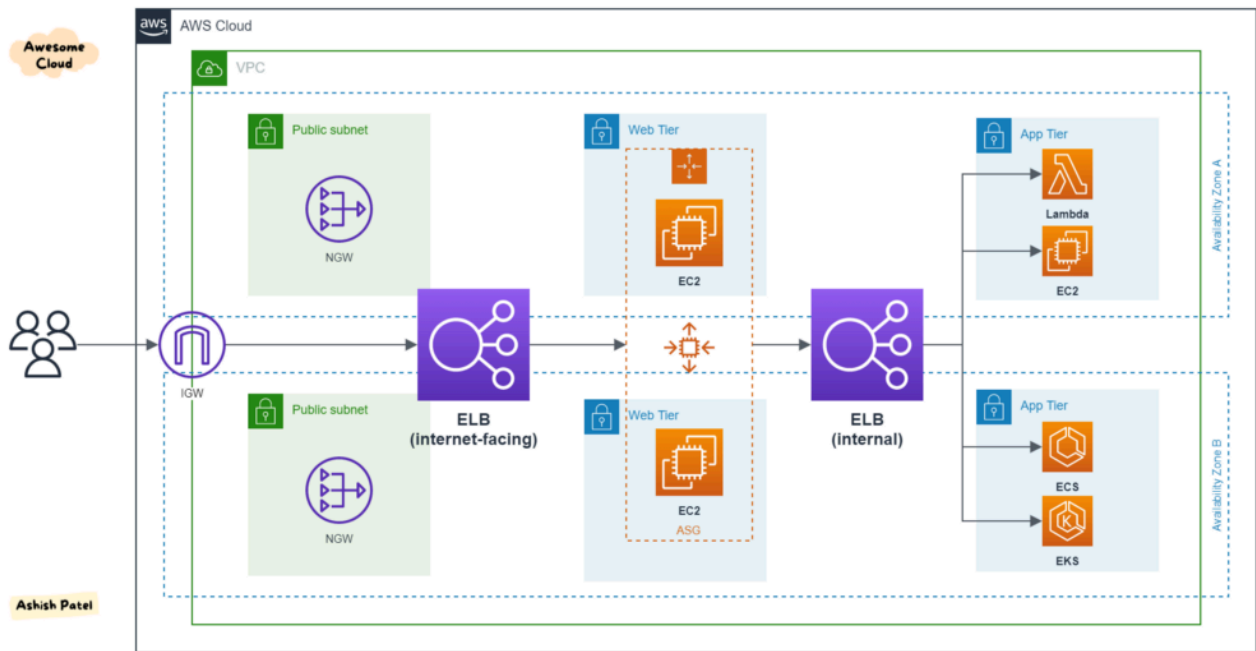
#### Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

ref) <https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#SelectCreateELBWizard:>



ref)

<https://medium.com/awesome-cloud/aws-elastic-load-balancer-elb-overview-introduction-to-aws-elb-alb-nlb-gwlb-e2820fe8fe27>

- LB(Load Balancer) 종류별 주요 기능 비교

| Feature                                | Application Load Balancer | Network Load Balancer                   | Gateway Load Balancer                    | Classic Load Balancer     |
|----------------------------------------|---------------------------|-----------------------------------------|------------------------------------------|---------------------------|
| Load Balancer type                     | Layer 7                   | Layer 4                                 | Layer 3 Gateway + Layer 4 Load Balancing | Layer 4/7                 |
| Target type                            | IP, Instance, Lambda      | IP, Instance, Application Load Balancer | IP, Instance                             |                           |
| Terminates flow/proxy behavior         | Yes                       | Yes                                     | No                                       | Yes                       |
| Protocol listeners                     | HTTP, HTTPS, gRPC         | TCP, UDP, TLS                           | IP                                       | TCP, SSL/TLS, HTTP, HTTPS |
| Reachable via                          | VIP                       | VIP                                     | Route table entry                        |                           |
| Supported network/Platforms            | VPC                       | VPC                                     | VPC                                      | EC2-Classic, VPC          |
| Health Checks                          | HTTP, HTTPS, gRPC         | TCP, HTTP, HTTPS                        | TCP, HTTP, HTTPS                         | TCP, SSL/TLS, HTTP, HTTPS |
| Support for fully private EKS clusters | ✓                         | ✓                                       |                                          |                           |

ref) [https://aws.amazon.com/elasticloadbalancing/features/#Product\\_comparisons](https://aws.amazon.com/elasticloadbalancing/features/#Product_comparisons)

- NLB 개요

- OSI 4 Layer 트래픽 제어
- LB 동작 개요

Client □ LB가 client connection request 수신 □ Target Group에서 Target 선택(w/ default rule)  
□ Listener 설정에 정의된 TCP:Port로 connection 시도

- Load Balancing 범위

- ◆ AZ Mode: 단일 AZ 상에 여러 Load Balancer Node 생성하여 traffic 분산 처리
- ◆ Cross-zone Mode: 여러 AZ에 걸쳐서 등록된 Target에 traffic 분산 처리

- [세부내용 참고](#)

- NI and static IP 자동 설정(by AWS)

- ◆ ELB는 사용자가 설정한 AZ에 NI(Network Interface)를 생성한다.
- ◆ AZ의 각 Load Balancer Node는 생성한 NI를 사용하여 Static IP를 할당한다.
  - Internet-facing LB의 경우, User는 Subnet 당 EIP를 할당할 수도 있다.

- Target Type과 Target 등록

- ◆ Target Group 생성시 Target Type을 선택 및 설정한다.
- ◆ Target Type에 따라 Target을 등록하는 방법이 결정된다.
- ◆ Target Type 종류
  - **Instance ID** Type: Client의 Source IP를 보관되어 User App에 제공된다.
  - **IP Address** Type: Source IP는 LB Node의 private IP 주소이다.
  - **ALB** Type: Client의 Source IP가 보관되어 User App에 제공된다.

- [세부내용 참고](#)

For TCP traffic, the load balancer **selects a target using a flow hash algorithm** based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number. The TCP connections from a client have **different source ports and sequence numbers, and can be routed to different targets**. Each individual TCP connection is **routed to a single target for the life of the connection**.

- 활용방법

- 생성 절차 및 API

1. Create a **load balancer** using [CreateLoadBalancer](#).
2. Create a **target group** using [CreateTargetGroup](#).
3. Register **targets** for the target group using [RegisterTargets](#).
4. Create one or more **listeners** for your load balancer using [CreateListener](#).

※ Listener 주요 업무

. Client로부터 연결 요청(request)을 check한다.

. User가 설정한 **Protocol, IP와 Port** 활용

. 요청 발생시 **Target Group**으로 전달한다.

※ Target Group 주요 업무

. 하나 이상의 등록된 **Target**으로 연결 요청을 전달한다.

. 하나의 **Target**은 여러 **Target Group**에 설정할 수 있다.

. 하나의 **Target Group**은 하나의 **NLB**에만 할당할 수 있다.

※ Health Check와 Target Group

. Health Check는 **Target Group** 대상으로 설정한다.

. Health Check 동작은 **Target Group**에 포함된 모든 **Target**에 대해 수행한다.

. Health Check 설정은 **Listener Rule(?)**로 정의한다.

- 삭제 절차 및 API

1. Delete the load balancer using [DeleteLoadBalancer](#).
2. Delete the target group using [DeleteTargetGroup](#).

ref) <https://docs.aws.amazon.com/elasticloadbalancing/latest/APIReference/Welcome.html> (2022/4/12)

- 실습 가이드

- ◆ Console 가이드: [Getting started with Network Load Balancers](#)

- ◆ CLI 가이드: [Tutorial: Create a Network Load Balancer using the AWS CLI](#)

- Console 기반 NLB 생성 실습

한페이지에 다음과 순서로 설정 후 생성 요청

- Basic configuration

- ◆ Scheme 선택: Internet-facing(External) | Internal

- ◆ IP address 타입 선택: IPv4 | Dualstack(IP4 and IP6)

### Basic configuration

**Load balancer name**  
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

powerkim-lb-test

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme**  
Scheme cannot be changed after the load balancer is created.

☒ Internet-facing  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal  
An internal load balancer routes requests from clients to targets using private IP addresses.

**IP address type** [Info](#)  
Select the type of IP addresses that your subnets use.

☒ IPv4  
Recommended for internal load balancers.

☐ Dualstack  
Includes IPv4 and IPv6 addresses.

- Network mapping

- ◆ VPC 선택, AZ 및 Subnet 1개 이상 선택

## Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

### VPC

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

(EKS)pwerkim-cluster

vpc-836f39ea

IPv4: 172.31.0.0/16



### Mappings

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

☒ **us-east-2a**

#### Subnet

subnet-bbe5c3d2

#### IPv4 settings

##### IPv4 address

Assigned by AWS

☐ **us-east-2b**

☐ **us-east-2c**

## ■ Listeners and routing

◆ Protocol 종류 선택: TCP | TCP\_UDP | TLS | UDP

## Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification.

▼ Listener **TCP:80**

Remove

Protocol

Port

Default action [Info](#)

TCP ▲

:

80

Forward to

Select a target group



TCP

1-65535

[Create target group](#)

TCP\_UDP

TLS

UDP

◆ Target Group 생성 | 선택



Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification.

▼ Listener TCP:80

Remove

Protocol

TCP ▼

Port

80

1-65535

Default action [Info](#)

Forward to

Select a target group ▲

Create target

Q |

k8s-default-hostname-bf64ba8561

Target type: Instance, IPv4

TCP

↺

Add listener

## ■ Tags – optional

► **Tags - optional**

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

## ■ Summary

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)

powerkim-lb-test

- Internet-facing
- IPv4

Network mapping [Edit](#)

VPC [vpc-836f39ea](#)

(EKS)pwerkim-cluster

- us-east-2a
- [subnet-bbe5c3d2](#)

Listeners and routing [Edit](#)

- TCP:80 defaults to [k8s-default-hostname-bf64ba8561](#)

Tags [Edit](#)

None

Attributes

ⓘ

Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

## ■ NLB 생성 후 제공 정보

### ◆ Table 목록 요약 정보 Field

- Name / DNS name / State / VPC ID / Availability Zones / Type / Create At / Monitoring

### ◆ 세부 정보

- Basic Configuration

Description




Listeners

Monitoring

Integrated services

Tags

## Basic Configuration

|                    |                                                                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name               | powerkim-lb-test                                                                                                                                                                |
| ARN                | arn:aws:elasticloadbalancing:us-east-2:635484366616:loadbalancer/net/powerkim-lb-test/786b56fedfa                                                                               |
| DNS name           | powerkim-lb-test-786b56fedfa9d2e7.elb.us-east-2.amazonaws.com <br>(A Record)                 |
| State              | Active                                                                                                                                                                          |
| Type               | network                                                                                                                                                                         |
| Scheme             | internet-facing                                                                                                                                                                 |
| IP address type    | ipv4<br><a href="#">Edit IP address type</a>                                                                                                                                    |
| VPC                | vpc-836f39ea                                                                                   |
| Availability Zones | subnet-bbe5c3d2 - us-east-2a <br>IPv4 address: Assigned by AWS<br><a href="#">Edit subnets</a> |
| Hosted zone        | ZLMOA37VPKANP                                                                                                                                                                   |
| Creation time      | April 14, 2022 at 8:57:37 PM UTC+9                                                                                                                                              |

- Attributes

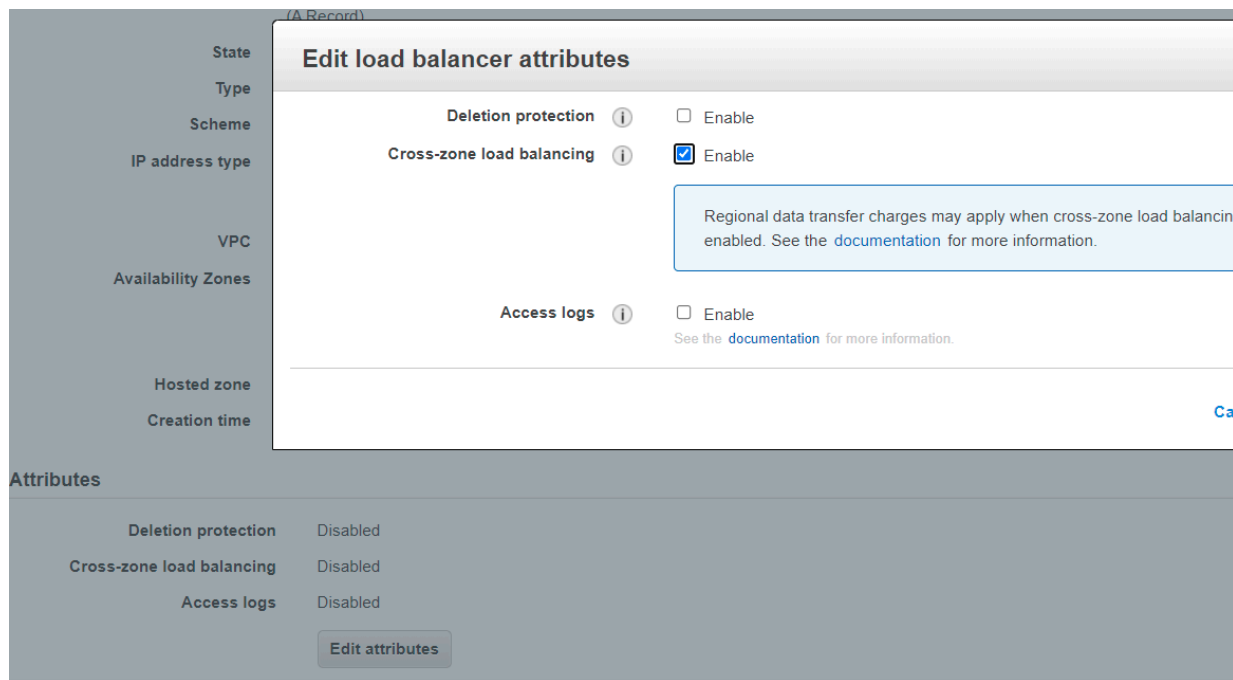
Cross-zone 선택 가능

## Attributes

|                           |          |
|---------------------------|----------|
| Deletion protection       | Disabled |
| Cross-zone load balancing | Disabled |
| Access logs               | Disabled |

[Edit attributes](#)

‘Edit attributes’ 클릭 후 Cross-zone 선택하면: Regional data transfer charge 경고



- Console 기반 Target group 생성 실습

한페이지에 다음과 순서로 설정 후 Next (Register targets)

- Basic configuration

- ◆ Target Type 선택: Instances | IP addresses | Lambda function | ALB

- Type 선택에 따라 Next 메뉴가 대상에 맞게 달라짐


## Basic configuration

Settings in this section cannot be changed after the target group is created.

### Choose a target type



#### Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#)  to manage and scale your EC2 capacity.



#### IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.



#### Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.



#### Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

### Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Port

HTTP ▼

:

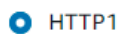
80

### VPC

Select the VPC with the instances that you want to include in the target group.

(EKS)pwerkim-cluster  
vpc-836f39ea  
IPv4: 172.31.0.0/16 ▼

### Protocol version



#### HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.



#### HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.



#### gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

◆ Protocol 선택: HTTP | HTTPS | TCP | TLS | UDP | TCP\_UDP | GENEVE

Target group name

powerkim-tg-test

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port

TCP ▲ : 80

HTTP

HTTPS

TCP

TLS

UDP

TCP\_UDP

GENEVE

with the instances that you want to include in the target group.

-cluster

/16

ks

and balancer periodically sends requests, per the settings below, to the registered targets to test their status.

## ■ Health checks

- ◆ Protocol 선택: TCP | HTTP | HTTPS

**Health checks**

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

TCP ▲

TCP

HTTP

HTTPS

Health check settings

- ◆ Protocol별 Advanced setting

- TCP Protocol

## Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

### Health check protocol

TCP ▼

### ▼ Advanced health check settings

Restore d

#### Port

The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

- ☒ Traffic port  
☐ Override

#### Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

3

2-10

#### Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

3

2-10

#### Timeout

The amount of time, in seconds, during which no response means a failed health check.

10

seconds

#### Interval

The approximate amount of time between health checks of an individual target

- ☐ 10 seconds  
☒ 30 seconds

10 or 30

- HTTP Protocol

## Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

### Health check protocol

HTTP ▼

### Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

### ▼ Advanced health check settings

Restore

#### Port

The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

☒ Traffic port

☐ Override

#### Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

3

2-10

#### Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

3

2-10

#### Timeout

The amount of time, in seconds, during which no response means a failed health check.

6

seconds

#### Interval

The approximate amount of time between health checks of an individual target

☐ 10 seconds

☒ 30 seconds

10 or 30

#### Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,201" or values (for example, "200-299").

200-399

- HTTPS Protocol (HTTP와 Timeout 값만 다름)

## Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

### Health check protocol

HTTPS ▼

### Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

### ▼ Advanced health check settings

Restore

#### Port

The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

☒ Traffic port

☐ Override

#### Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

3

2-10

#### Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

3

2-10

#### Timeout

The amount of time, in seconds, during which no response means a failed health check.

10

seconds

#### Interval

The approximate amount of time between health checks of an individual target

☐ 10 seconds

☒ 30 seconds

10 or 30

#### Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,201" or values (for example, "200-299").

200-399

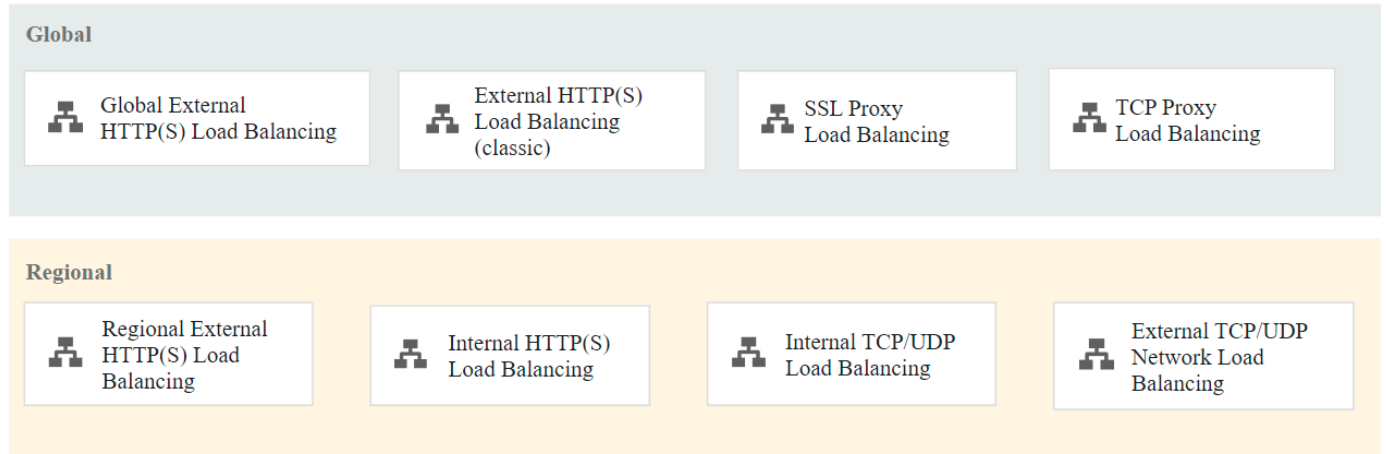
- Next button 클릭
- Register targets 설정 (Target Group Type 선택에 따라 설정 메뉴가 다름)





### ### GCP

- 개요



ref) <https://cloud.google.com/load-balancing/docs/load-balancing-overview?hl=ko>

- 특징

## HTTP(S) 부하 분산

HTTP(S) 부하 분산을 통해 다양한 리전의 여러 백엔드 인스턴스로 HTTP 및 HTTPS 트래픽을 분산시킬 수 있습니다. 단일 전역 IP 주소로 전체 앱을 사용할 수 있어 DNS 설정이 단순해집니다. HTTP(S) 부하 분산은 확장성과 내결함성을 갖추고 있으며 가동 준비 과정이 필요 없고 콘텐츠 기반 부하 분산을 지원합니다. HTTPS 트래픽의 경우 SSL 종료 및 부하 분산이 지원됩니다.

## 원활한 자동 확장

자동 확장을 통해 애플리케이션에서 트래픽 증가 처리가 원활해지고 리소스 수요가 줄면 비용을 절감할 수 있습니다. [자동 확장 정책](#)을 정의하기만 하면 자동 확장 처리에서 측정된 부하에 따라 자동 확장을 수행합니다. 가동 준비 과정이 필요 없으며 단 몇 초 만에 최대 범위까지 확장됩니다.

## 고급 기능 지원

Cloud Load Balancing에는 또한 IPv6 글로벌 부하 분산, WebSocket, 사용자 정의 요청 헤더, 비공개 VIP용 프로토콜 전달 등의 고급 지원 기능이 포함되어 있습니다.

## UDP 부하 분산

UDP 부하 분산은 Compute Engine 리전의 인스턴스 풀을 통해 UDP 트래픽을 분산시킵니다. 확장 가능하며 가동 준비 과정이 필요 없고 상태 확인을 통해 양호한 상태의 인스턴스만 트래픽을 수신하도록 보장합니다.

ref) <https://cloud.google.com/load-balancing?hl=ko>

- 특이점

- LB 종류별로 LB 생성 메뉴가 다름
- Backend에 Target Group(Pool) 개념이 없이 VM 단위 추가 또는 VM Group을 맵핑함.
- ◆ 다수 개의 Target Group 관리 지원하지 않음

- Console 위주의 실습: 문서화 생략

## Cloud Logging

부하 분산용 Cloud Logging이 부하 분산기에 전송된 모든 부하 분산 요청을 로깅합니다. 이 로그는 디버깅은 물론 사용자 트래픽 분석에 사용할 수 있습니다. 요청 로그를 조회하고 Cloud Storage, BigQuery 또는 Pub/Sub에 내보내 분석할 수 있습니다.

## SSL 오프로드

SSL 오프로드를 사용하면 중앙에서 SSL 인증서와 복호화를 관리할 수 있습니다. 부하 분산 레이어와 백엔드 사이의 암호화를 활성화하여 백엔드 처리에 대한 오버헤드를 추가해 보안을 최대한 강화할 수 있습니다.

## 어피니티

Cloud Load Balancing 어피니티를 통해 사용자 트래픽을 특정 백엔드 인스턴스로 전달해 고정할 수 있습니다.

## TCP/SSL 부하 분산

TCP 부하 분산은 Compute Engine 리전의 인스턴스 풀을 통해 TCP 트래픽을 분산시킬 수 있습니다. 확장 가능하며 가동 준비 과정이 필요 없고 상태 확인을 통해 양호한 상태의 인스턴스만 트래픽을 수신하도록 보장합니다. SSL 프록시가 부하 분산과 함께 HTTPS 외 트래픽에 대한 SSL 종료 기능을 제공합니다.

## 정확한 상태 확인

상태 확인은 새로운 연결의 부하가 이를 수신할 만큼 상태가 양호한 백엔드로만 분산되도록 합니다. 정확한 상태 확인이 가능하기 때문에 조사를 할 때 백엔드에 대한 실제 트래픽을 모방할 수 있습니다.

## Cloud CDN 통합

체크박스 클릭 한 번으로 [Cloud CDN](#)에 HTTP(S) 부하 분산을 사용 설정하여 사용자 애플리케이션 제공을 최적화할 수 있습니다.

### ### Azure

- Console 위주의 실습: 문서화 생략