Chapter 08

연결 제어 2: 분산제어

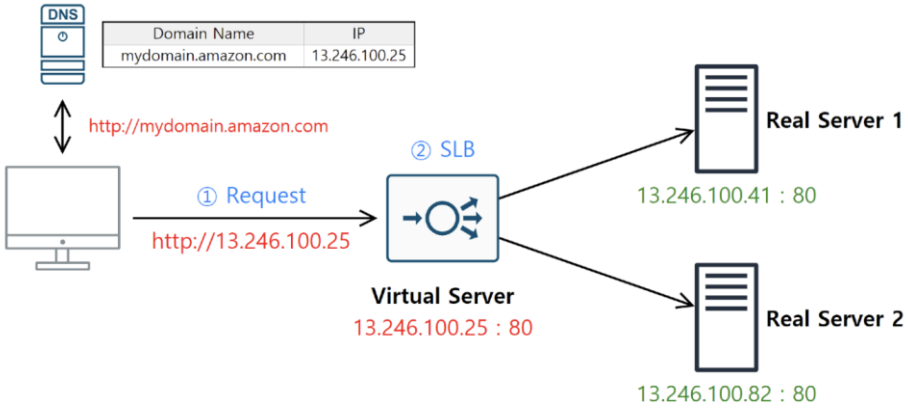
1. 서버 로드밸런싱(SLB) 개요

- 분산 제어란 클라이언트 요청을 수신한 서버가 그 요청 트래픽을 다시 여러 서버로 분산해 전달함을 의미한다.
- 그런데 트래픽을 왜 분산할까?
- 단일 서버의 처리량(Capacity)을 초과한 대량 요청 트래픽은 속도 저하나 서비스 지연 또는 장애를 유발한다.
- 이 상황을 대비해 부하(load)를 다수 서버로 분산(balancing)하는 것이다.
- 이것을 서버 로드밸런싱(Server Load Balancing, SLB)이라 한다.

1. 서버 로드밸런싱(SLB) 개요

온프레미스의 SLB 제어 : L4 스위치

- 부하를 분산한다는 것은 클라이언트 요청을 어디선가 수신하고 있다는 뜻이다.
- 그리고 그 요청을 여러 서버가 나눠 처리토록 배분해야 할 것이다.
- 이와 같이 클라이언트 트래픽을 가장 처음 수신하는 서버를 가상서버(Virtual Server)라 한다.
- 다음 그림은 온프레미스의 SLB 예시다.



- 가상 서버는 가상 IP(Virtual IP- vip)와 가상 포트(Virtual Port, vport)로 구성된다.
- 요청을 받은 가상 서버는 웹이 구동 중인 2개의 리얼 서버(Real Server)와 리얼 포트(Real Port, rport) 쌍으로 부하를 분산한다.
- 이때 리얼 서버 두 대는 웹 페이지 요청을 처리할 Nginx나 Apache 등의 웹 서버를 구동해 80 포트(report)를 리슨하고 있을 것이다.

1. 서버 로드밸런싱(SLB) 개요

온프레미스의 SLB 제어 : L4 스위치

- 서버의 부하 분산은 다음과 같이 해석할 수 있다.

> 13.246.100.25:80으로 요청이 들어오면
> → 13.246.100.41:80 또는 13.246.100.82:80으로 분산한다.

- 이같은 가상 서버의 포트 분산 기능은 4계층 이상에서 처리할 수 있으므로, 최소 Layer 4 또는 Layer 7 네트워크 장치가 필요하다.
- 이 장치를 L4스위치, L7스위치라 한다.
- 네트워크 상위 계층 장비는 하위 계층 기능을 수행할 수 있다.
- 따라서 부하 분산(L4)과 더불어 HTTP와 같은 애플리케이션 프로토콜(L7) 헤더 분석이 필요하면 L7 스위치를 사용하고 헤더 분석 필요없이 IP와 포트 쌍을 단순히 분산한다면 L4 스위치를 사용한다.
- 그러나 이 두 가지 모두 4계층 레벨을 기반으로 트래픽을 처리하므로 대표해 L4 스위치라 부르기도 한다.

1. 서버 로드밸런싱(SLB) 개요

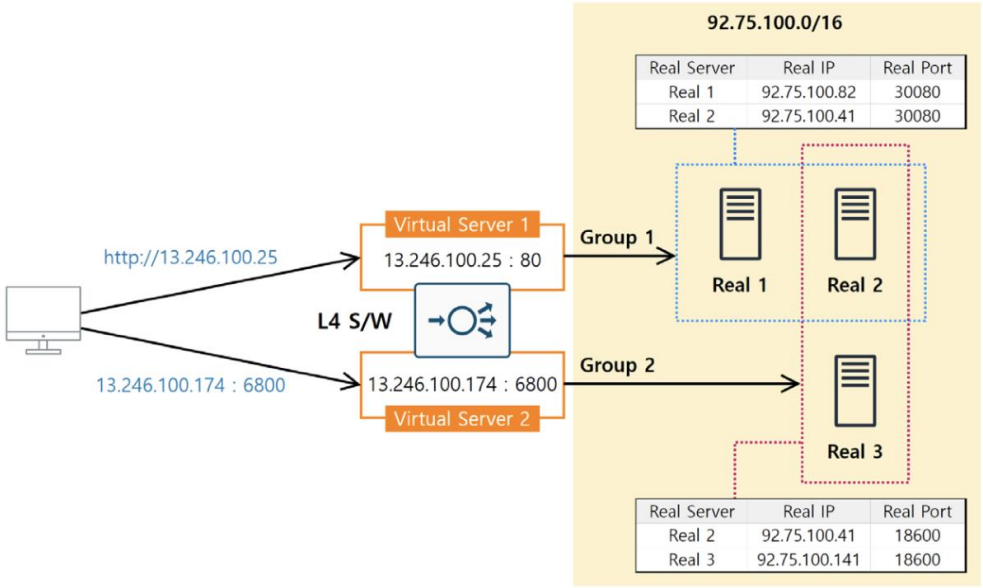
L4 스위치의 특징

- **L4 스위치 내부에 다수의 가상 서버를 생성할 수 있다.**
- **L4 스위치를 경계로 외부망(인터넷)과 내부망을 분리할 수 있다. 이는 보안과 직결된다. 쉽게 말해 L4 스위치만 퍼블릭 IP를 할당하고 리얼 서버는 프라이빗 IP만 할당해 서비스한다. 이 구성으로써 L4 스위치를 통과하지 않으면 인터넷의 리얼 서버 접근이 불가하므로 안전하다.**
- **가상 서버가 트래픽을 분산하는 리얼 서버의 모음을 대상 그룹(Target Group)이라 한다.**
- **vport와 rport가 서로 달라도 된다. vport로써 rport를 추측하기 어려워질수록 보안은 향상된다.**
- **가상 서버가 대상 그룹의 멤버(리얼 서버)로 트래픽을 분산하는 방식은 매우 다양하다. 또 대상 그룹마다 분산방식을 선택할 수 있다. 여기서는 3가지 방식을 설명한다.**
  1. 라운드 로빈(Round Robin): 대상 그룹에 포함된 리얼 서버에 순차적으로 분산하는 방식이다.
  2. IP 해시(Hash) : 클라이언트 IP 주소를 해시 함수의 변수로 활용하고, 고정된 리얼 서버로 분산하는 방식이다.
  3. Least Connection(최소 연결): 클라이언트의 신규 요청 시점에 리얼 서버별 기존 연결(분산) 수를 분석해 그 수가 가장 적은 서버(부하가 덜한 서버)로 분산하는 방식이다.

1. 서버 로드밸런싱(SLB) 개요

L4 스위치의 특징

- 다음 그림은 위의 특징을 반영한 예시다.



- 다음 그림의 트래픽 요청과 분산 경로는 다음과 같이 표현할 수 있다.

사용자 요청	Virtual Server IP	Virtual Port	요청 전달	Target Group	요청 분산	Real Server IP	Real Port
→	13.246.100.25	80	→	1	→	92.75.100.82	30080
					→	92.75.100.41	30080
→	13.246.100.174	6800	→	2	→	92.75.100.41	18600
					→	92.75.100.141	18600

1. 서버 로드밸런싱(SLB) 개요

L4 스위치 Config 예시: Alteon

- 대표 L4 스위치 중 하나인 Alteon 장비에 설정값을 입력해 앞장의 그림처럼 작동하는 환경을 구성해 보자.
- 장비 운영에 필요한 기본 설정(계정, SNMP, 물리적 포트, 인터페이스, 게이트웨이, VRRP 등)은 생략하고 SLB 설정만 다뤄본다.
- 설정값 입력 방법은 이렇다.
- 다음 설정 예시에 보이는 첫 슬래시(/)는 메뉴 최상위 경로를 뜻한다.
- 하위 메뉴로 이동하려면 메뉴명 다음에 슬래시(/)를 입력하고 원하는 메뉴에 진입하고 세부 설정값을 넣으면 된다.
- 전체 경로로써 한 번만에 메뉴로 진입할 수도 있다.
- 또 Enter를 여러 번 눌러도 된다.
  - (예) /c/slb/real 1 = 메인에서 c + Enter + slb + Enter + real 1

1. 서버 로드밸런싱(SLB) 개요

L4 스위치 Config 예시: Alteon

Config 입력 순서는 다음과 같다.

① SLB 설정 켜기

– CONFIG설명 c: configuration, slb: server load balancing

```
/c/slb
        on
```

② 리얼 서버 IP 지정

– 리얼 서버 3개를 정의하고 각 IP를 지정한다.

– CONFIG설명 real: 리얼 서버 정의, ena : 활성화, ipver v4: IPv4사용, rip : 리얼 서버 IP 지정

```
/c/slb/real 1
        ena
        ipver v4
        rip 92.75.100.82
/c/slb/real 2
        ena
        ipver v4
        rip 92.75.100.41
/c/slb/real 3
        ena
        ipver v4
        rip 92.75.100.141
```

1. 서버 로드밸런싱(SLB) 개요

L4 스위치 Config 예시: Alteon

- **Config 입력 순서는 다음과 같다.**

③ 대상 그룹 정의와 멤버 지정

- 1 번 그룹에 1 번과 2번 서버를 추가하고, 2번 그룹에는 2번과 3번 서버를 추가한다.
- CONFIG 설명 group: 대상 그룹 정의, metric: 분산 방식 선택, add : 그룹에 멤버 추가

```
/c/slb/group 1
        ipver v4
        metric roundrobin
        add 1
        add 2
/c/slb/group 2
        ipver v4
        metric roundrobin
        add 2
        add 3
```

1. 서버 로드밸런싱(SLB) 개요

L4 스위치 Config 예시: Alteon

- **Config 입력 순서는 다음과 같다.**

④ 가상 서버 설정

- 가상 서버 2개(virt 1,virt 2)를 각각 정의한다.
- 가상 서버마다 사용자 요청을 수신할 서비스 vport(http, 6800) 지정하고 vport로 들어온 요청을 분산할 대상 그룹과 rport를 지정한다.
- CONFIG 설명 virt : 가상서버 정의, vip: 가상서버의 IP 지정, service : 서비스 포트 지정, group: 분산대상 그룹 지정, rport: 분산 대상 리얼 포트 지정

```
/c/slb/virt 1
        ena
        ipver v4
        vip 13.246.100.25
/c/slb/virt 1/service http
        group 1
        rport 30080
/c/slb/virt 2
        ena
        ipver v4
        vip 13.246.100.174
/c/slb/virt 2/service 6800
        group 2
        rport 18600
```

2. AWS의 SLB 제어 : 로드밸런서(ELB)

- AWS도 온프레미스와 비슷한 용어를 사용한다.
- AWS의 SLB를 Elastic Load Balancing(탄력적 로드밸런싱 또는 로드밸런싱)이라 하고, 이 기능을 제공하는 서비스를 로드밸런서(Elastic Load Balancer, ELB)라 한다.

2. AWS의 SLB 제어 : 로드밸런서(ELB)

L4 스위치 vs. 로드밸런서(ELB)

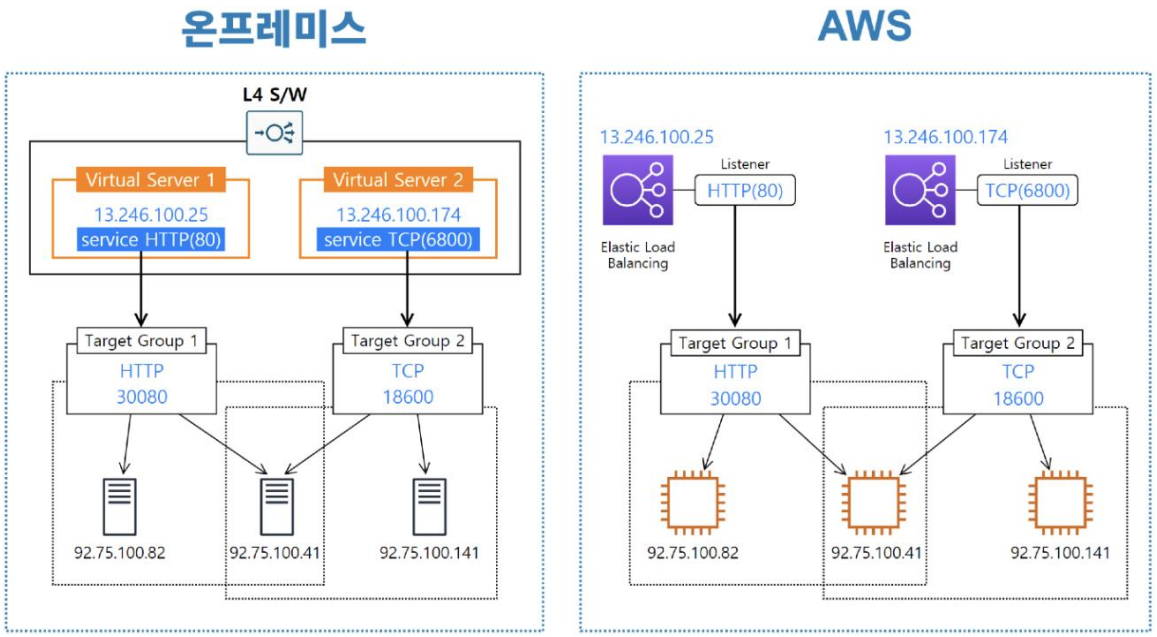
- 다음 표는 온프레미스의 L4 스위치와 ELB 각 구성 요소를 비교한 것이다.

구분 \ 제어 요소	L4 / L7 스위치	ELB
요청 수신	가상 서버 (Virtual Server)	ELB의 IP + 리스너(Listener)
가상 서버(리스너) 구성 요소	가상 IP + 프로토콜과 포트	리스너의 프로토콜과 포트
VIP 개수	가상 서버당 1개	ELB당 1개 도메인
포함 관계	가상 서버의 모음 ⊂ L4 스위치	리스너의 모음 ⊂ ELB
요청 전달 대상	대상 그룹(Target Group) + 대상 그룹의 프로토콜과 포트	
요청 분산 대상	리얼 서버 IP + 리얼 서버의 프로토콜과 포트	대상(Target) + 대상의 프로토콜과 포트
다중화	장비 다중화 가능	ELB 다중화 불가

2. AWS의 SLB 제어 : 로드밸런서(ELB)

L4 스위치 vs. 로드밸런서(ELB)

- 표에서 제시한 7가지 비교 항목을 다음 그림에서 확인해보자.



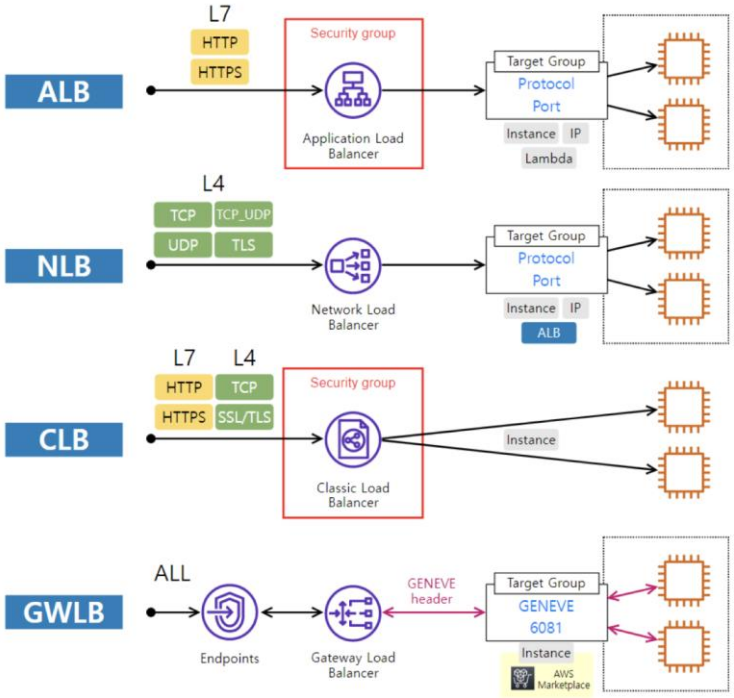
- L4 스위치와 ELB는 구조나 생김새가 매우 유사하다.
- 굳이 차이점을 꼽는다면 L4 스위치는 내부에 가상 서버(가상IP)를 여러 개 만들 수 있다는 점이다.
- AWS에서는 여러 ELB를 생성하는 것으로 대체한다.
- 하지만 가상 서버마다 다수 서비스(프로토콜과 포트)를 생성할 수 있듯이 ELB도 내부에 다수의 리스너를 생성할 수 있다.

2. AWS의 SLB 제어 : 로드밸런서(ELB)

ELB 유형 비교: ALB, NLB, CLB, GWLB

- **AWS는 다음 4개 ELB 유형을 제공한다.**
  - 애플리케이션 로드밸런서(Application Load Balancer, ALB)
  - 네트워크 로드밸런서(Network Load Balancer, NLB)
  - 클래식 로드밸런서(Classic Load Balancer, CLB)
  - 게이트웨이 로드밸런서(Gateway Load Balancer, GWLB)

- **다음 그림은 4개 유형을 비교하기 쉽게 표현한 토폴로지다.**



2. AWS의 SLB 제어 : 로드밸런서(ELB)

ELB 유형 비교: ALB, NLB, CLB, GWLB

- **4개의 ELB 유형의 특징은 다음과 같다.**
  - ALB 와 CLB 만 SG를 사용한다.
  - CLB는 대상그룹을 사용하지 않는다.
  - ALB 는 L7, NLB 는 L4, 그리고 CLB 는 L4 와 L7 모두 지원한다.
  - ALB, NLB, CLB의 목적은 ELB에 등록된 대상으로 로드밸런싱하는 것이다.
  - GWLB의 목적도 이와 같지만, 대상이 트래픽의 최종목적지는 아니다. GWLB는 IPS나 방화벽 같은 어플라이언스(Appliance)로 로드밸런싱한다. 즉, 로드밸런싱 대상(어플라이언스)은 트래픽 검사나 필터링을 위한 중간 경유지며, 트래픽 최종 목적지는 아니다. 따라서 모든 트래픽을 수용해야 한다.
  - ALB의 대상 그룹 유형은 인스턴스, IP, Lambda를 지원하고 NLB는 인스턴스, IP, ALB, 그리고 CLB 는 인스턴스만 지원한다.

2. AWS의 SLB 제어 : 로드밸런서(ELB)

ELB 유형 비교: ALB, NLB, CLB, GWLB

- 로드밸런서 유형별 특징 비교

특징 \ ELB 유형	ALB	NLB	CLB	GWLB
보안 그룹(SG) 사용	O	X	O	X
대상 그룹 사용	O	O	X	O
대상(그룹) 유형	인스턴스, IP, Lambda	인스턴스, IP, ALB	인스턴스	GENEVE 지원 어플라이언스
계층	L7	L4	L4 / L7	L3 / L4
리스너 프로토콜	HTTP, HTTPS	TCP, UDP, TCP_UDP, TLS	HTTP, HTTPS, TCP, SSL/TLS	IP

- ALB는 HTTP와 HTTPS 서비스에 최적화돼 있다.
- 4계층의 대용량, 고성능 트래픽을 분산하려면 NLB를 사용한다.
- CLB는 VPC가 아닌 EC2-Classic 플랫폼에서도 사용할 수 있으나 2013년 12월 이후 생성된 AWS 계정은 EC2-Classic 플랫폼 지원이 안되므로 참고한다.
- 추가참고 사항으로 EC2-Classic은 2022년 8월부로 공식 지원이 종료된다.

ELB 가용 영역과 노드

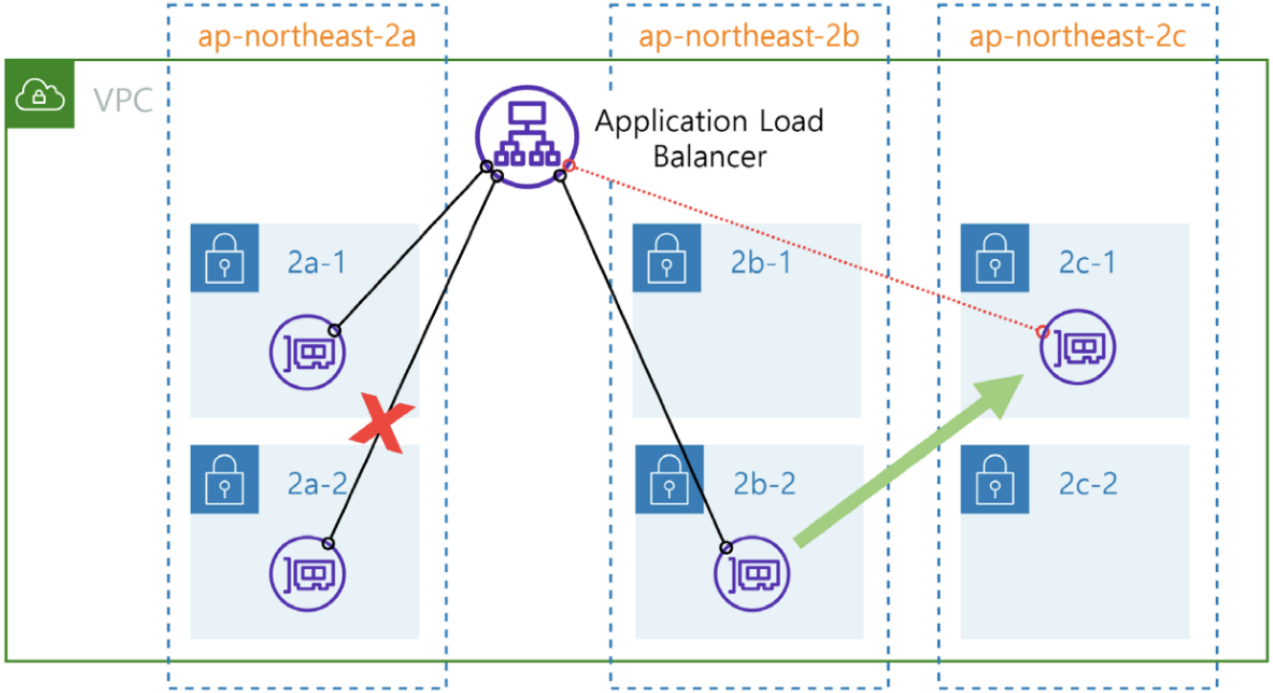
ELB의 특징

- ELB의 부모는 VPC다. 또 ELB 유형에 따라 수명 주기 동안 연결 특성이 달라진다. ALB와 CLB는 다른 요소에 연결할 수 없으며 NLB와 GWLB는 엔드포인트 서비스에 연결할 수 있다.
- ELB 생성과 동시에 ELB용 ENI가 생성된다. 이것을 로드밸런서 노드(Load Balancer Node)라 한다.
- 노드 생성 위치는 ELB 생성 시 선택한 가용 영역의 서브넷이다.
- 가용 영역별 1 개 서브넷만 선택할 수 있다. 즉, 가용 영역별 1 개 노드가 생성된다.
- ELB의 실제 역할은 노드가 수행한다. 노드는 각 가용 영역에서 활동하는 ELB의 특파원으로 비유할 수 있다. 클라이언트 요청을 받은 ELB는 각 가용 영역에 대기중인 노드로 명령을 내려 로드밸런싱을 수행토록 한다.
- 각 가용 영역의 노드는 가용 영역의 모든 서브넷으로 로드밸런싱할 수 있다.
- 모든 가용 영역을 선택하지 않아도 된다. 단, 선택한 가용 영역으로만 로드밸런싱을 할 수 있다.
- 클라이언트가 실제 접속하는 ELB의 IP는 엄밀히 말해 노드의 IP다.
- ALB는 반드시 2개 이상의 가용 영역을 선택해야 한다.

3. 로드밸런싱 처리부

ELB 가용 영역과 노드

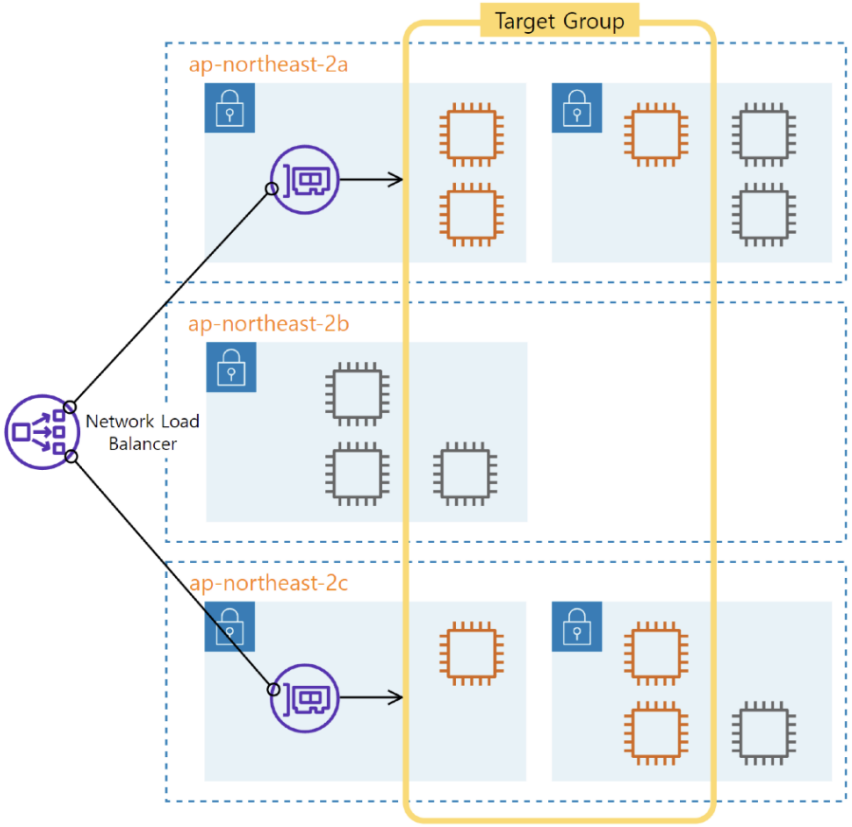
- 가용 영역 전체를 관할하는 ELB 노드



3. 로드밸런싱 처리부

ELB 중복 구현 : 교차 영역 로드밸런싱

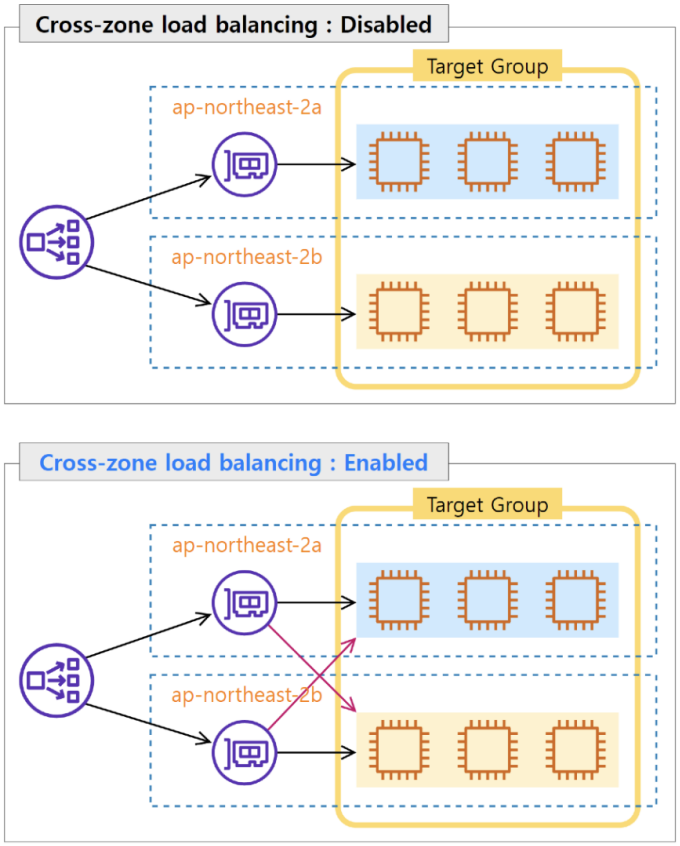
- 온프레미스에서는 L4 스위치 장비를 다중화해 장애를 대비하지만, ELB는 다중화가 불가능했다.
- 그럼 ELB 다중화는 어떻게 구현할까?
- 그림은 다중 가용 영역(2a, 2c)을 지정한 NLB의 모습이다.



3. 로드밸런싱 처리부

ELB 중복 구현 : 교차 영역 로드밸런싱

- 문제는 2a 노드가 장애면 2a에 놓인 인스턴스로는 로드밸런싱이 불가능하다는 것이다.
- 이 문제는 교차 영역 로드밸런싱(Cross-zone load balancing)으로 해결할 수 있다.
- 다음 그림은 교차 영역 로드밸런싱 속성을 활성화(아래)한 것과 그렇지 않은(위) NLB를 나타낸다.



3. 로드밸런싱 처리부

ELB 중복 구현 : 교차 영역 로드밸런싱

- 이처럼 교차 영역 로드밸런싱을 활성화하면 각 노드가 다른 가용 영역으로도 로드밸런싱하므로 단 하나의 ELB로 다중화를 구현할 수 있다.
- ELB 유형별 기능 차이는 다음 표와 같다.

특징 \ ELB 유형	ALB	NLB	CLB	GWLB
교차 영역 로드밸런싱 속성 기본값	활성화	비활성화	·API, CLI : 비활성화 ·콘솔 : 활성화	비활성화
속성 변경	불가	가능	가능	가능

3. 로드밸런싱 처리부

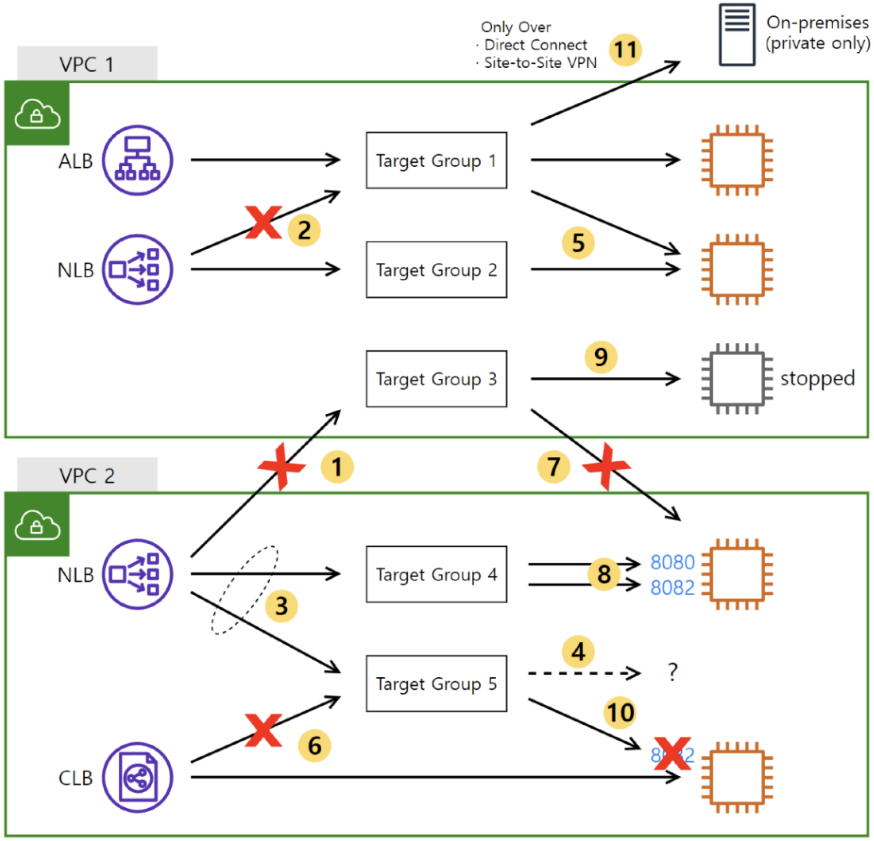
대상 그룹과 대상

- **대상 그룹(Target Group)은 ELB가 로드밸런싱하는 대상(Target)의 모음이다.**
- **대상 그룹의 특징은 다음과 같다.**
  - 대상 그룹의 패런트는 VPC 다. 수명 주기 동안 VPC 를 변경할 수 없다.
  - 대상 그룹은 수명 주기 동안 다른 요소에 연결할 수 있다. 연결 대상은 ELB다.
  - 연결 종속성으로 인해 기존 연결된 ELB를 해제해야 다른 ELB에 연결할 수 있다. 즉, 하나의 대상 그룹을 여러 ELB에 연결할 수 없다. 단일 ELB의 제어를 받기 때문이다.
  - 그러나 다중 연결 가능성(N : 1)이 있으므로 여러 대상 그룹을 하나의 ELB에 연결할 수 있다.

3. 로드밸런싱 처리부

대상 그룹과 대상

- 대상 그룹은 ENI와 유사하다.
- ENI는 인스턴스의 통제 하에 있기 때문에 ENI 1개를 여 러 인스턴스에 연결할 수는 없지만 여러 ENI를 하나의 인스턴스에 연결할 수는 있었다.
- 이 같은 대상 그룹의 특징을 다음 그림에 표현했다.



3. 로드밸런싱 처리부

대상 그룹과 대상

- **다음의 각 특징은 앞의 그림에 표시된 번호와 일치한다.**

① 대상 그룹의 부모는 VPC다. 따라서 다른 VPC의 ELB에 연결할 수 없다. 또 [Target Group 3]처럼 그 어떤 ELB에 연결되지 않고 홀로 존재할 수 있다.

② 대상 그룹과 ELB는 1:N 관계가 불가능하다. 단, 기존 ELB와 연결을 해제하면 다른 ELB에 연결할 수 있다.

③ 대상 그룹과 ELB는 N : 1 관계가 가능하다. 하나의 ELB가 여러 대상 그룹으로 라우팅할 수 있다.

④ 대상 그룹이 그 어떤 대상도 포함하지 않을 수 있다.

⑤ [Target Group 1] 포함하는 대상을 [Target Group 2]가 포함해도 된다.

⑥ CLB는 대상 그룹으로 로드밸런싱할 수 없다. 인스턴스로 직접 로드밸런싱한다.

⑦ [VPC 1]에 속한 [Target Group 2]이 [VPC 2]에 속한 대상 인스턴스를 포함할 수 없다. 단, VPC 간 피어링이 연결되면 인스턴스 ID 대신 IP 주소를 등록할 수 있다.

⑧ 동일 대상(인스턴스)의 다른 포트로 로드밸런싱을 할 수 있다.

⑨ 중지된 인스턴스도 대상으로 지정할 수 있다.

⑩ 포트가 중지 된 인스턴스도 대상으로 지정할 수 있다.

⑪ 온프레미스의 프라이빗 IP를 대상으로 지정할 수 있다. 단, IP 유형을 지원하는 ALB와 NLB만 가능하며 Direct Connect와 사이트 간 VPN 연결상에서만 대상 지정을 할 수 있다.

3. 로드밸런싱 처리부

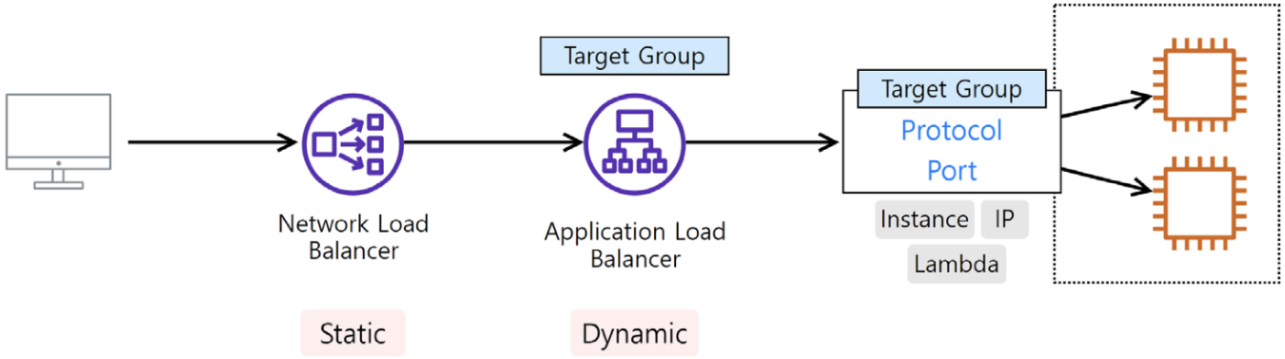
가변 노드를 장착한 ELB : ALB, CLB

- ELB는 유형에 따라 노드 생성 방식이 다르다.
- NLB 생성 시 3개(2a, 2b, 2c)의 가용 영역을 선택했다면 정확히 노드 3개(2a, 2b, 2c)가 생성된다.
- 그러나 ALB와 CLB는 그렇지 않다.
- 이 둘은 노드를 가변적으로 조정한다.
- 즉, 트래픽 부하나 로드밸런싱 대상상태에 따라 노드를 확장하고 삭제한다.
- 한편 가용 영역에 있던 기존 노드를 제거하고 새 노드를 만들어 교체하기도 한다.
- 교체 과정에서 동일한 가용 영역에 2개 이상의 노드가 공존할 때도 있다.
- 물론 일정 시간이 지나면 가용 영역별 단 하나의 노드만 남는다.
- 반면 NLB는 로드밸런싱 대상 유무와 관계없이 모든 가용 영역에 노드 1개씩을 생성해 둔다.
- 수명 주기 동안 노드 변경도 없다.
- 이 특성은 NLB에만 탄력적 IP를 할당할 수 있는 근거가 된다.
- 모든 노드(ENI)가 고정적이기 때문이다.
- 탄력적 IP는 프라이빗 IP 단위로 할당되므로, 프라이빗 IP가 ENI에서 해제되면 탄력적 IP 연결도 해제된다.
- 따라서 프라이빗 IP가 바뀌지 않는 NLB만 탄력적 IP를 연결하도록 설계했을 것이다.

3. 로드밸런싱 처리부

NLB 대상 그룹에 ALB 연결하기

- 가변 노드를 사용하는 ALB와 CLB는 탄력적 IP가 아닌 동적 퍼블릭 IP만 사용할 수 있다.
- ALB와 CLB의 DNS 이름(도메인)을 쿼리하면 상황에 따라 노드의 IP와 그 개수가 달라진 것을 확인할 수 있을 것이다.
- 따라서 접근 제어 적용 시 어려움을 겪는다.
- 이 같은 가변 노드의 단점을 보완하기 위해 2021년 9월부터 NLB에 ALB 유형 대상 그룹 연결을 지원하고 있다(그림).



4. 요청 수신부

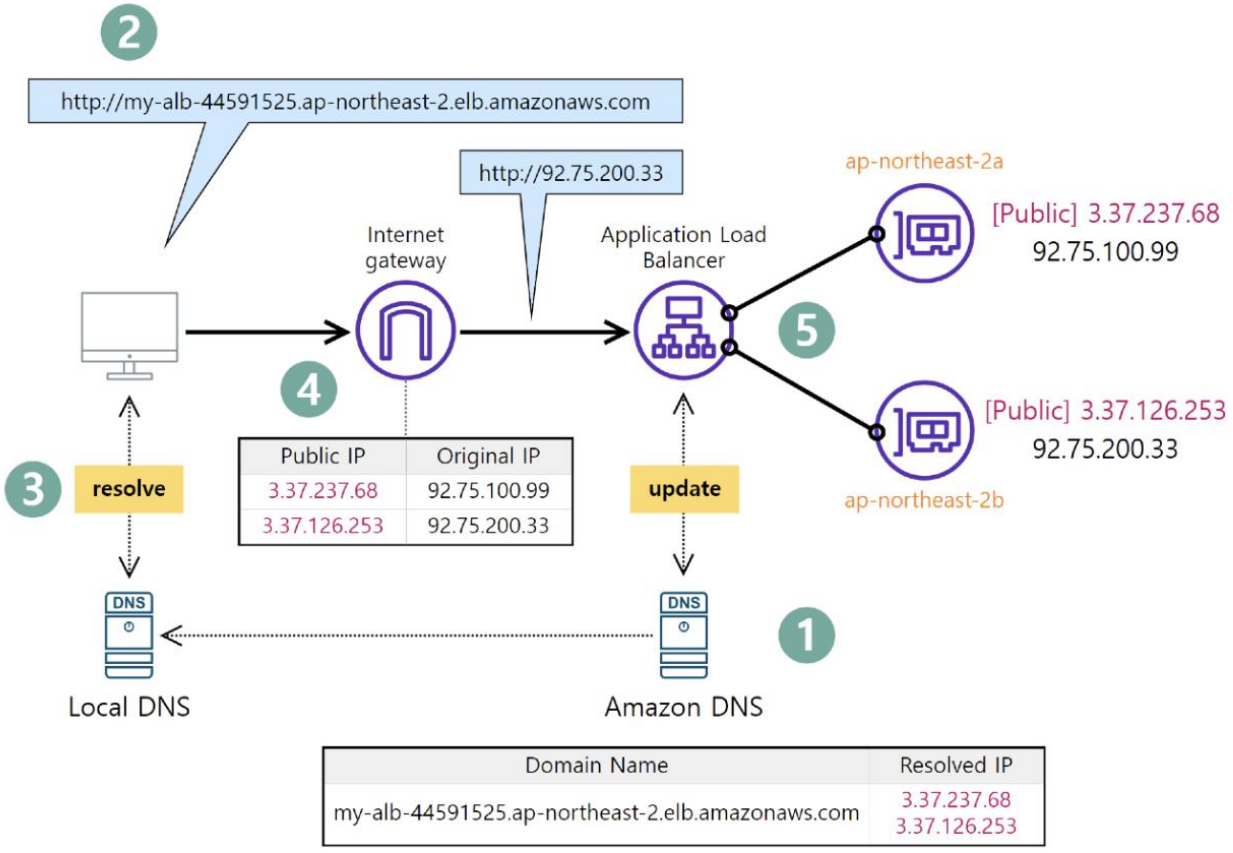
ELB 체계와 DNS 이름

- **무엇이든 ELB 요청 주체가 될 수 있다.**
- **이를테면 일반 사용자(PC, 모바일)나 온프레미스 시스템, 그리고 AWS 서비스 등 모든 컴퓨팅이 ELB 클라이언트다.**
- **이렇듯 ELB는 어떤 클라이언트라도 쉽게 액세스하도록 다음 기능을 제공한다.**
  - ELB 전용 DNS 이름(DNS name)을 사용한다. DNS 이름은 ELB마다 유일하며 변하지 않는다. ELB의 DNS 이름과 ELB가 소유한 모든 노드 IP 간 매핑 정보가 DNS에 저장되므로, 클라이언트는 모든 노드의 IP를 일일이 몰라도 DNS 이름만으로 ELB에 접속할 수 있다.
  - ELB는 2가지 요청 유형을 제공한다. 이를 체계(Scheme)라 한다.
    - 인터넷(Internet) 체계 : 인터넷상의 클라이언트가 IGW를 통과해 ELB로 접속 시 사용한다. 따라서 IGW의 NAT 테이블에 등록될 퍼블릭 IP(또는 탄력적 IP)가 노드에 할당돼 있어야 한다.
    - 내부(Internal) 체계 : 클라이언트가 IGW를 통과하지 않고 ELB로 접속할 때 사용한다. 그러므로 퍼블릭 IP(또는 탄력적 IP)는 없다. 클라이언트는 리전 위치와 무관하며 VPC 내부 또는 동일 계정의 다른 VPC나 다른 계정의 VPC에 존재할 수도 있다. 또 Direct Connect, VPN등 VGW를 경유해 접속할 때도 사용한다. 이처럼 VPC 네트워킹을 사용하지 않는 클라이언트도 내부 체계 ELB로 트래픽을 요청할 수 있다.
  - 이 2가지 체계의 구분 기준은 클라이언트 트래픽의 IGW 통과 여부다.
  - 그러나 DNS 이름은 이 체계와 관계없이 트래픽 요청 과정을 매우 단순하게 한다.

4. 요청 수신부

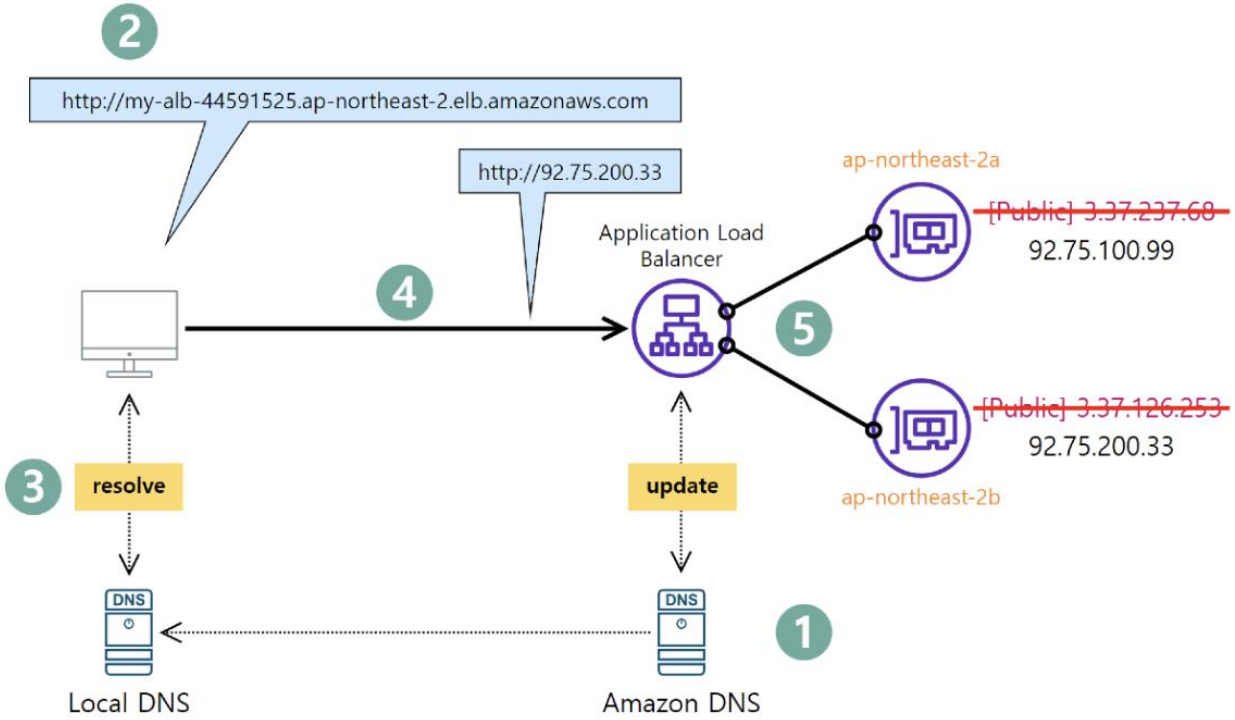
ELB 체계와 DNS 이름

- 다음 그림은 인터넷 체계 ALB의 트래픽 처리 과정이다.



4. 요청 수신부

- **ELB 체계와 DNS 이름**
  - 그럼 내부 체계는 어떨까?
  - 다음 그림은 내부 ALB로 웹 트래픽을 요청하는 과정을 나타낸다.



Domain Name	Resolved IP
my-alb-44591525.ap-northeast-2.elb.amazonaws.com	92.75.100.99 92.75.200.33

4. 요청 수신부

ELB 체계와 DNS 이름

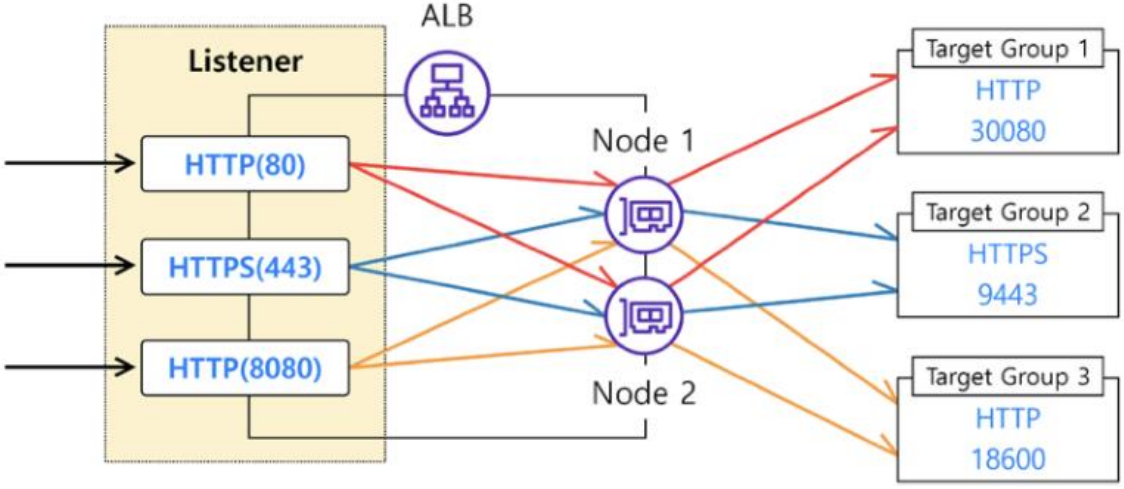
- 인터넷 체계 ELB를 사용한다는 것은 ELB 노드에 퍼블릭 IP(또는 탄력적 IP)가 할당됐음을 뜻한다.
- ALB, NLB, CLB는 인터넷 체계를 사용할 수 있으며 GWLB는 VPC 엔드포인트로만 접근하므로 내부 체계만 제공한다.
- 앞서 언급한대로 NLB는 고정 노드를 사용하므로 탄력적 IP를 연결할 수 있다.
- 반면 ALB나 CLB는 동적 퍼블릭 IP를 할당받아 사용한다.
- ELB 유형별 인터넷 체계 속성 비교

<table>
<tr><th>특징 \ ELB 유형</th><th>ALB</th><th>NLB</th><th>CLB</th><th>GWLB</th></tr>
<tr><td>인터넷 체계
사용 가능</td><td colspan="3">O</td><td rowspan="2">X</td></tr>
<tr><td>노드에
탄력적 IP
연결 가능</td><td>X</td><td>O</td><td>X</td></tr>
</table>

4. 요청 수신부

리스너

- 클라이언트 요청이 자극이라면 로드밸런싱은 반응이다.
- 이 자극을 기다리고 반응하는 것 모두 리스너의 역할이다.
- 리스너(Listener)는 포트를 열어 놓고 대기한다.
- 클라이언트 요청이 대기 중인 ELB 리스너의 IP 및 프로토콜(포트) 쌍과 일치하면 대상(그룹)으로 로드 밸런싱한다.
- 다음 그림은 웹 요청을 수신한 ALB가 대상 그룹으로 트래픽을 로드밸런싱하는 예를 보여준다.

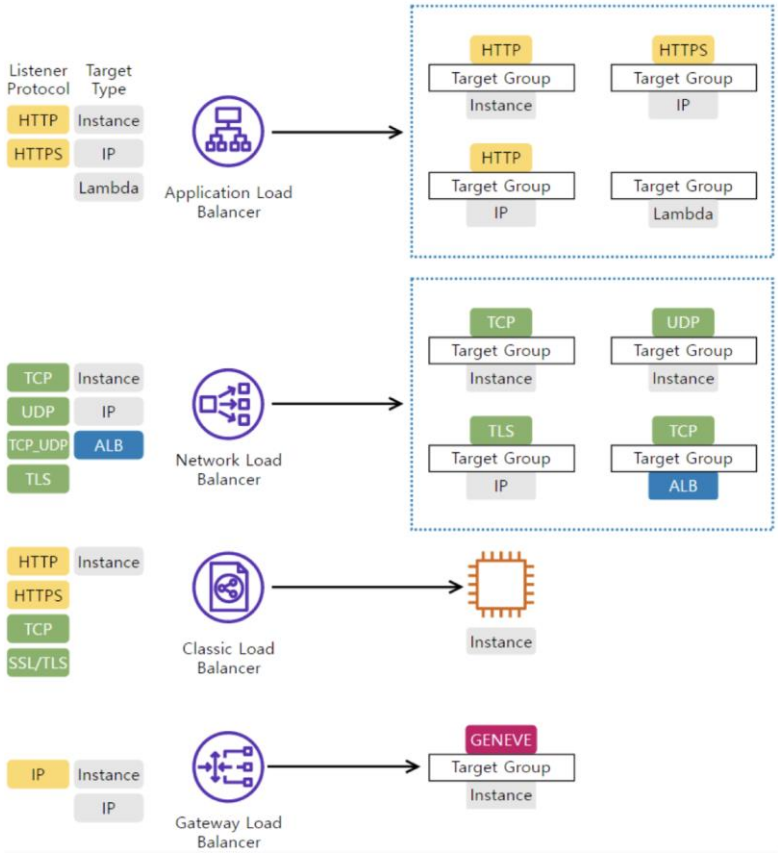


사용자 요청	리스너 프로토콜(포트)	로드밸런싱	대상 그룹의 프로토콜(포트)	대상 그룹
→	HTTP(80)	→	HTTP(30080)	1
→	HTTPS(443)	→	HTTPS(9443)	2
→	HTTP(8080)	→	HTTP(18600)	3

5. 요청 수신부터 로드밸런싱 처리까지

리스너와 대상 그룹

- 앞 장의 표를 보면 리스너 프로토콜(HTTP, HTTPS)과 대상 그룹의 프로토콜(HTTP, HTTPS)이 일치함을 알 수 있다.
- 즉, TCP나 UDP 대상 그룹은 연결할 수 없다.
- 다음 그림은 이를 보여주고 있다.



5. 요청 수신부터 로드밸런싱 처리까지

리스너와 대상 그룹

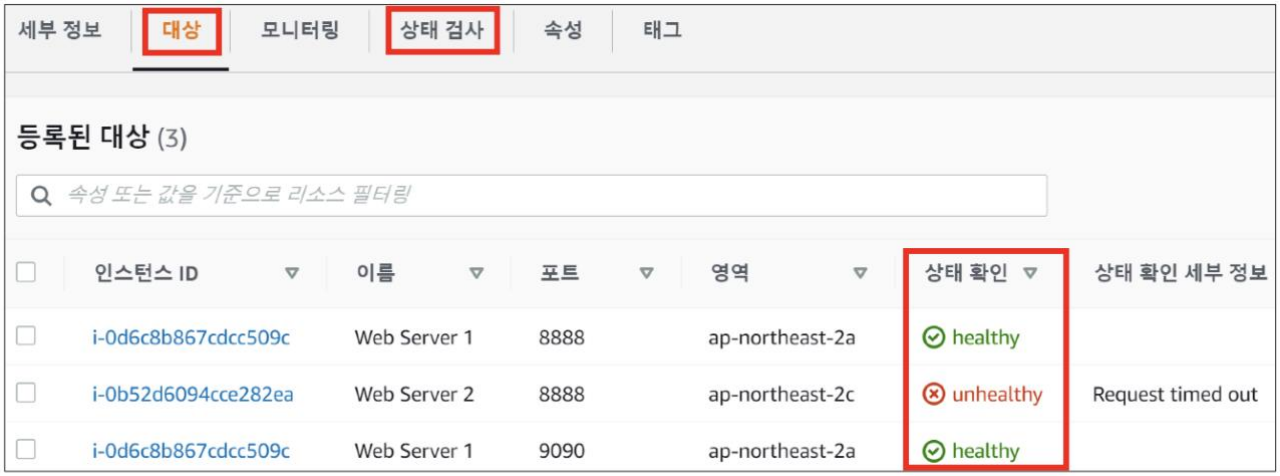
- ELB 유형별 사용 가능한 리스너 프로토콜은 다음 표와 같다.

특징 \ ELB 유형	ALB	NLB	CLB	GWLB
리스너 프로토콜 = 대상 그룹 프로토콜	HTTP, HTTPS	TCP, UDP, TCP_UDP, TLS	HTTP, HTTPS, TCP, SSL/TLS	IP
계층	L7	L4	L4 / L7	L3 / L4
대상(그룹) 유형	인스턴스, IP, Lambda	인스턴스, IP, ALB	인스턴스	GENEVE 지원 어플라이언스 (인스턴스)

5. 요청 수신부터 로드밸런싱 처리까지

상태 검사 결과가 ELB에 미치는 영향

- 로드밸런싱 대상의 정상(Healthy) 여부를 모니터링하는 기능이 있다.
- 이것을 상태 검사(Health Checks)라 한다.
- 대상 그룹을 사용하는 3가지 유형(ALB, NLB, GWLB)은 다음 그림처럼 서비스 > EC2 > 대상 그룹 메뉴의 대상 탭에서 상태 확인 속성을 볼 수 있다.
- 또 상태 검사 옵션은 상태 검사 탭에서 설정한다.



5. 요청 수신부터 로드밸런싱 처리까지

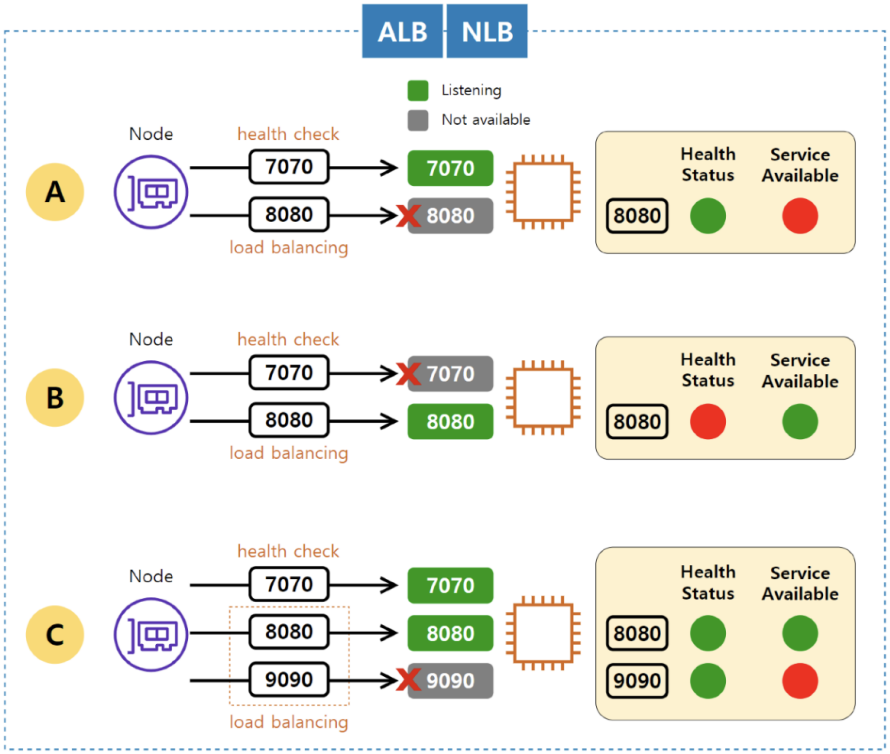
상태 검사 결과가 ELB에 미치는 영향

- 반면 대상 그룹이 없는 CLB는 다음 그림처럼 서비스 > EC2> 로드밸런서 메뉴의 인스턴스 탭에서 상태를 확인하고 상태 검사 탭에서 검사 옵션을 설정한다.

5. 요청 수신부터 로드밸런싱 처리까지

상태 검사 결과가 ELB에 미치는 영향

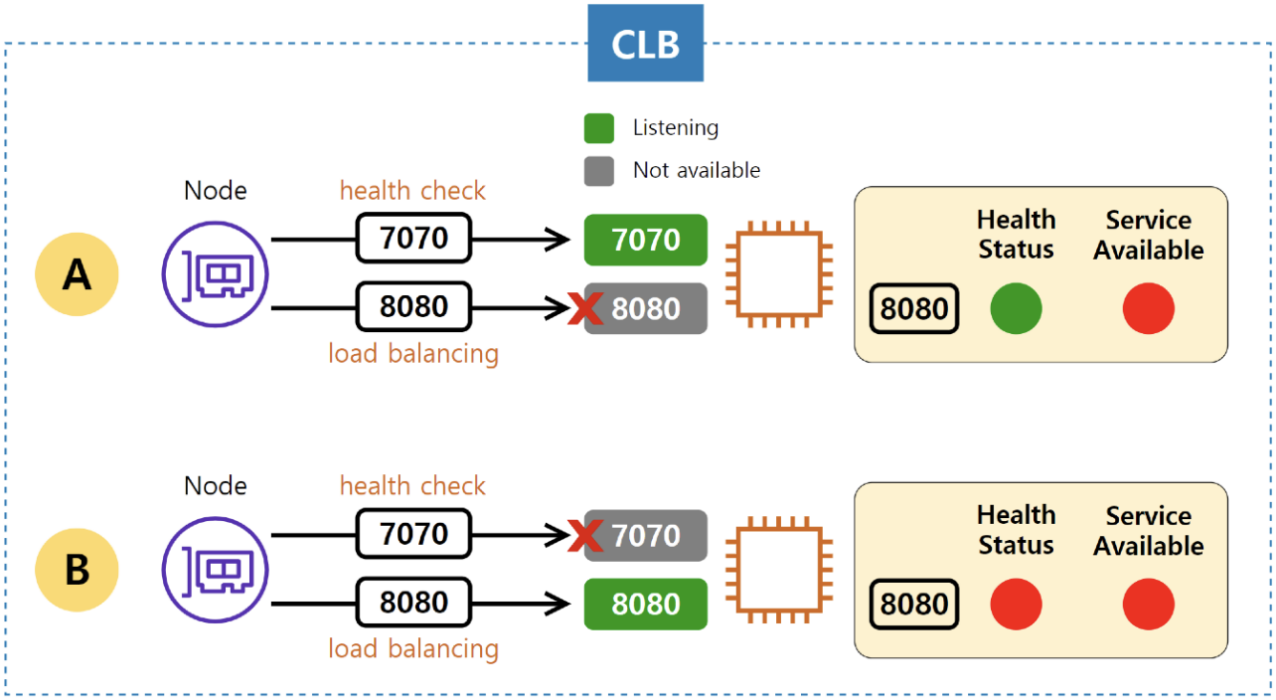
- 다음 그림은 ELB 노드가 대상 그룹으로 로드밸런싱하는 3개의 예시다.
- 편의상 대상 그룹에는 인스턴스 1개만 있다고 가정한다.
- 상태 검사 포트와 로드밸런싱(트래픽) 포트를 같게 하거나 예시처럼 다르게 지정할 수도 있다.
- 포트가 같다면 상태 검사 결과(Health Status)를 신뢰할 수 있으므로 별도로 표현하진 않았다.
- 또 상태 검사 및 로드밸런싱용 접근 제어는 설정됐다고 가정한다.



5. 요청 수신부터 로드밸런싱 처리까지

상태 검사 결과가 ELB에 미치는 영향

- 이처럼 상태 검사만으론 ELB 서비스 상태를 신뢰하기 어렵다.
- 다음 그림은 CLB 예시를 보여준다.



5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

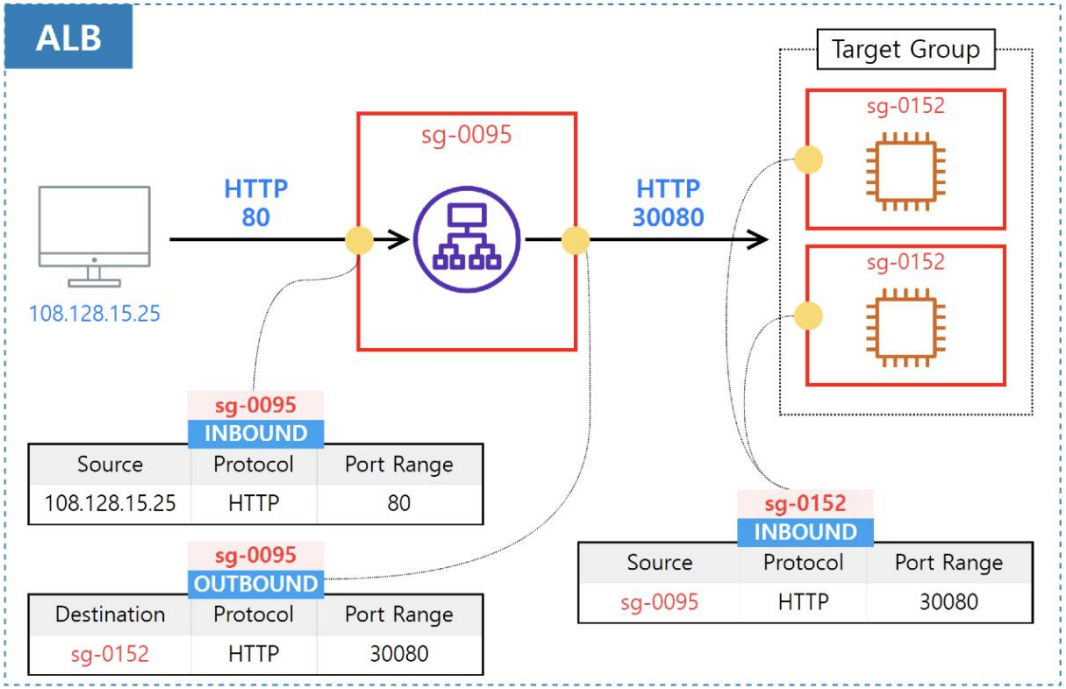
- **ENI 유형은 다음 2가지 차이점이 있었다.**
  - 소스/대상확인 : 라우팅 ENI는 소스/대상 확인 설정이 해제된 상태로 생성되며 트래픽을 수신하면 데이터 변경없이 포워딩한다.
  - SG 강제 적용 : 컴퓨팅 ENI는 SG를 사용하고, 라우팅 ENI는 사용하지 않는다.

- **ELB 노드가 컴퓨팅 ENI를 사용하면 컴퓨팅 노드를 정의하고, 라우팅 ENI를 사용하면 라우팅 노드라 정의하자.**
- **정리하면 다음처럼 분류할 수 있다.**
  - 컴퓨팅 노드 : ALB, CLB
  - 라우팅 노드 : NLB, GWLB

5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

- 다음 그림은 ALB와 라우팅 대상에 연결된 SG 모습이다.

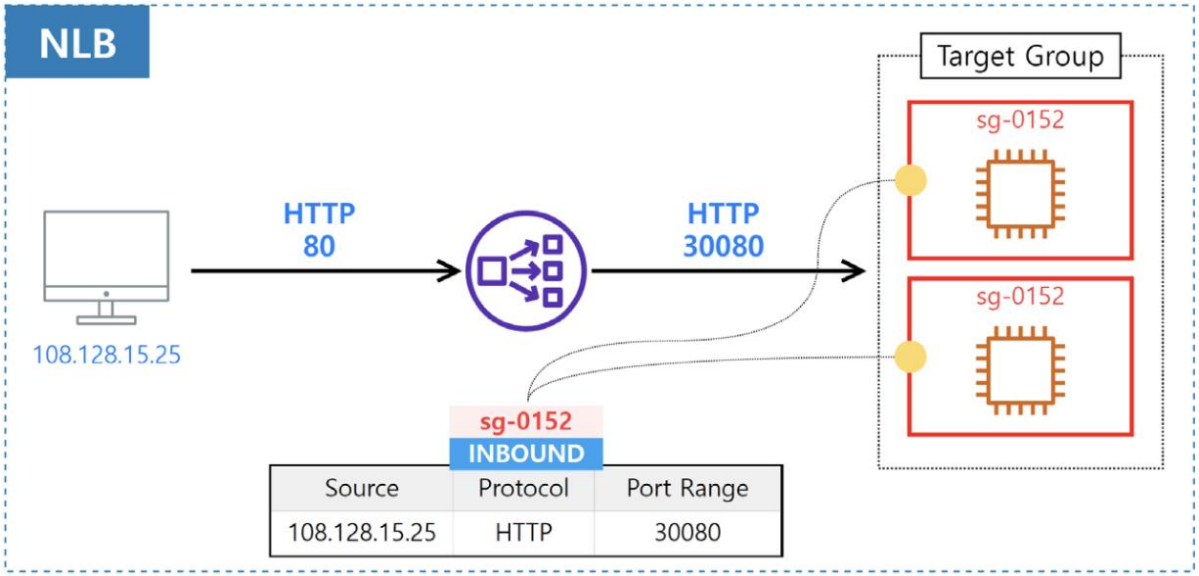


- **ELB에 연결된 SG는 다음 접근만 허용해야 한다.**
  - 인바운드 : 클라이언트 IP와 ELB 리스너 프로토콜(포트)
  - 아웃바운드 : ELB가 로드밸런싱할 대상 IP와 프로토콜(포트)

5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

- 다음 그림은 라우팅 노드를 사용하는 NLB의 모습이다.



- NLB는 ALB와 다르게 SG를 사용하지 않는다.
- 그러므로 로드밸런싱 대상 2개가 클라이언트 접근을 직접 제어한다.
- 예시처럼 인터넷 체계를 사용하면 각 대상들이 인터넷 클라이언트 접속을 직접 상대해야 하므로 접근 제어에 특히 유의해야 한다.

5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

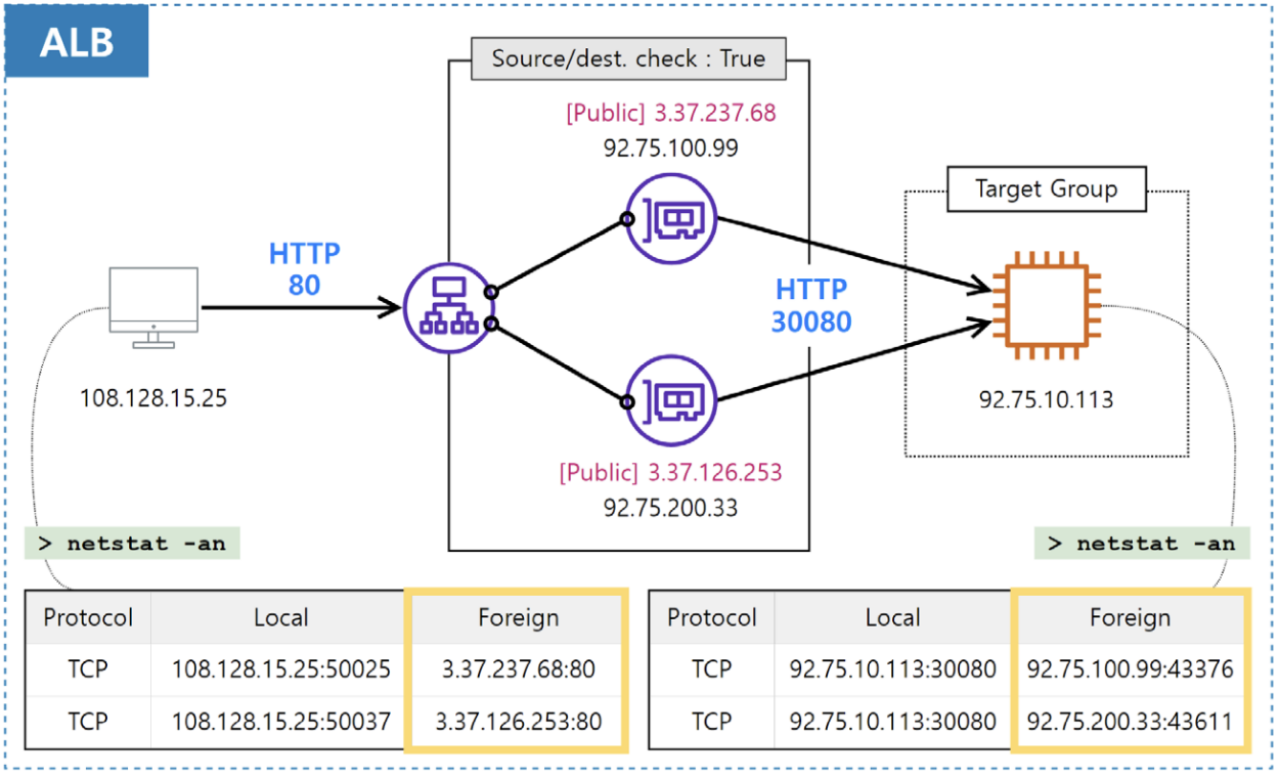
- 그럼 로드밸런싱만 관여할 것 같던 ALB는 어떤 데이터 처리 과정이 있을까?
- 다음 그림은 ALB와 NLB 리스너의 라우팅 방법을 선택하는 화면이다.

- 전달 대상 옵션만 선택할 수 있는 NLB와는 달리 ALB는 다음 4가지 옵션이 있다.
  - 전달 대상 : 앞서 학습한 일반적인 로드밸런싱이다.
  - 리디렉션 : 클라이 언트에게 다른 라우팅을 제시한다. (예) [https://92.75.200.33]요청을 [https://92.75.200.33]으로 변경 처리한다.
  - 고정 응답 반환 : 클라이언트 요청 데이터와 관계없이 단 하나의 응답만 제공한다. 정기 PM 작업, 리뉴얼 페이지 전환 안내 시 유용하다.
  - 인증 : Amazon Cognito나 OIDC(OpenID Connect) 사용자를 인증할 수 있다. 리스너 프로토콜이 HTTPS 일 때만 사용한다.

5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

- 다음 그림은 ALB에 맺어진 세션의 모습이다.



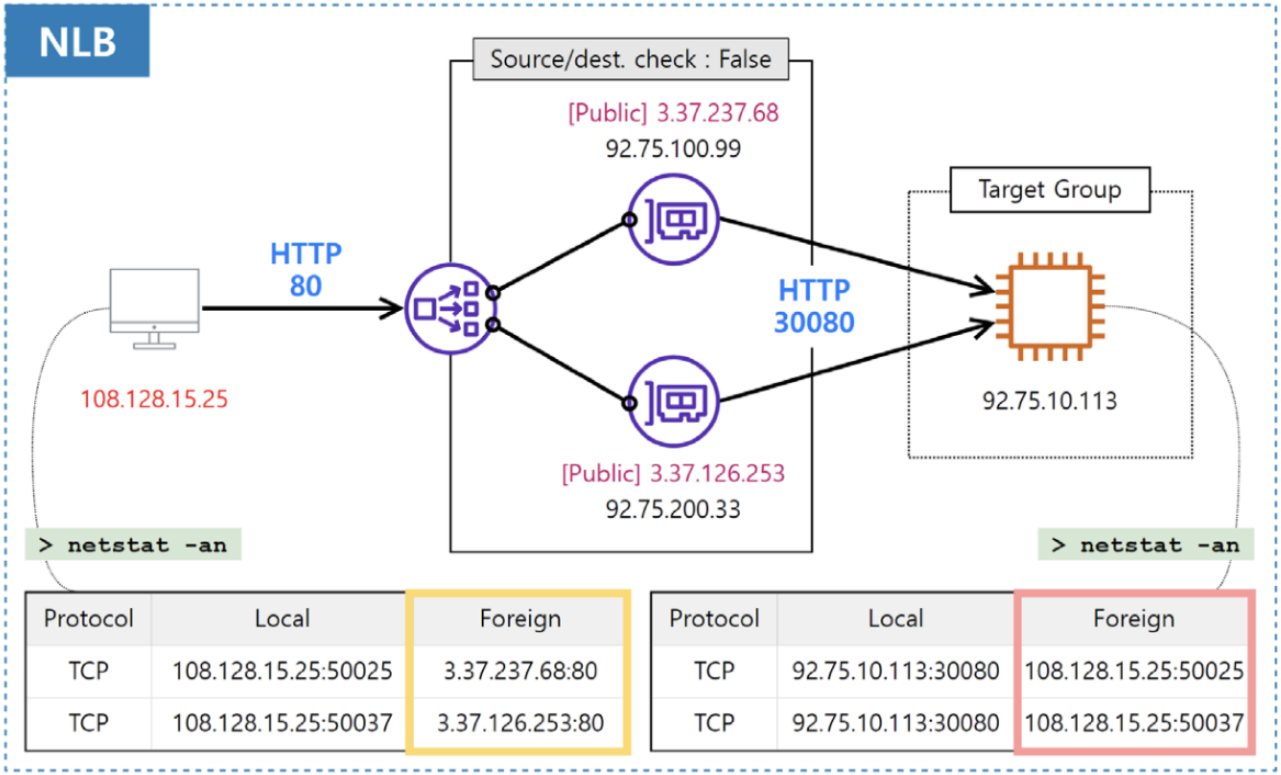
Protocol	Local	Foreign
TCP	108.128.15.25:50025	3.37.237.68:80
TCP	108.128.15.25:50037	3.37.126.253:80

Protocol	Local	Foreign
TCP	92.75.10.113:30080	92.75.100.99:43376
TCP	92.75.10.113:30080	92.75.200.33:43611

5. 요청 수신부터 로드밸런싱 처리까지

컴퓨팅 노드와 라우팅 노드 비교: ALB vs. NLB

- 다음 그림은 NLB에 접속한 클라이언트와 대상의 세션이다.



Protocol	Local	Foreign
TCP	108.128.15.25:50025	3.37.237.68:80
TCP	108.128.15.25:50037	3.37.126.253:80

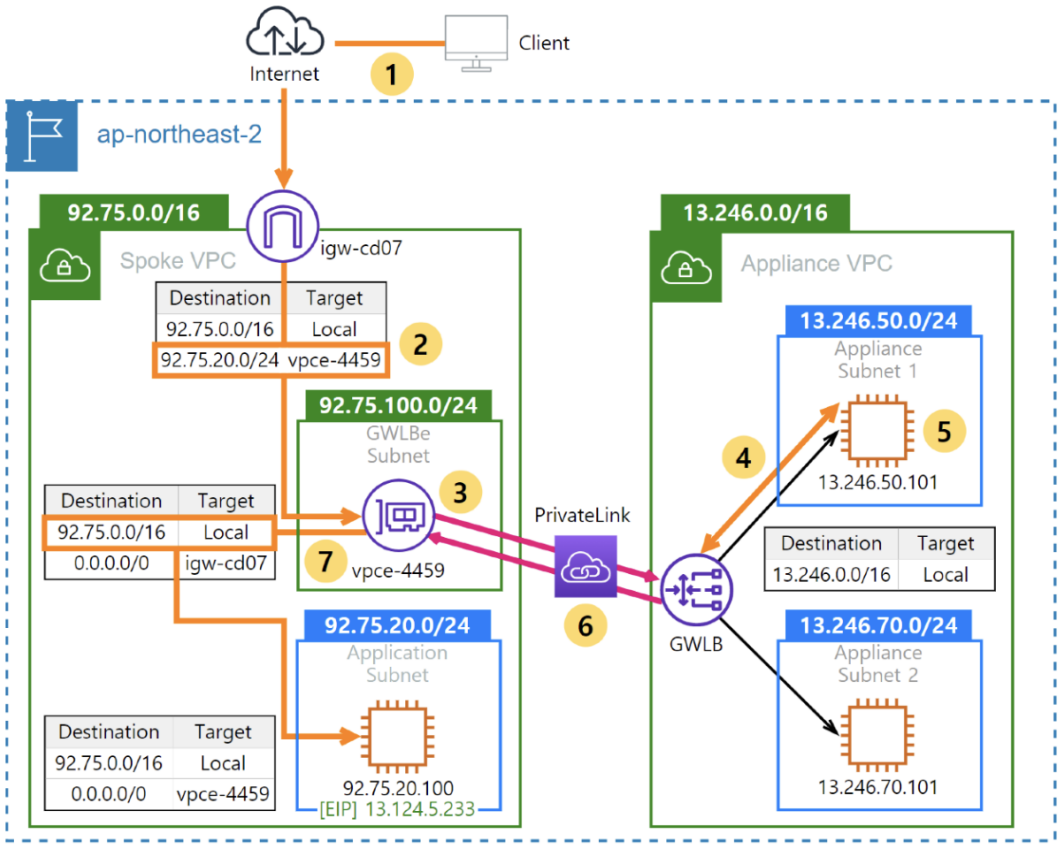
Protocol	Local	Foreign
TCP	92.75.10.113:30080	108.128.15.25:50025
TCP	92.75.10.113:30080	108.128.15.25:50037

6. 게이트웨이 로드밸런서(GWLB)

- GWLB도 로드밸런싱을 한다.
- 그러나 로드밸런싱 형태가 다른 3개 유형과 확연히 구분된다.
- 클라이언트가 이 3개 ELB(ALB, NLB, CLB)로 요청하면 분산 대상 중 하나가 클라이언트의 요청을 처리하고 회신한다.
- 다시 말해 대상이 클라이언트 트래픽의 최종 목적지다.
- 그러나 GWLB의 대상은 클라이언트의 최종 목적지가 아니다.
- 클라이언트의 요청 트래픽은 GWLB가 로드밸런싱하는 대상에 잠깐 들린 후 최종 목적지로 다시 이동한다.

6. 게이트웨이 로드밸런서(GWLB)

- 다음 그림은 GWLB의 트래픽 처리 과정이다.
- GWLB는 엔드포인트를 기반으로 하는 ELB다.

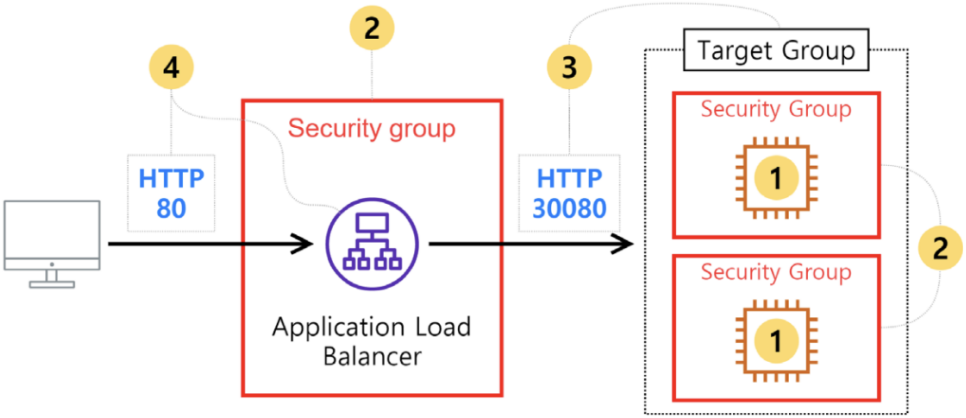


6. 게이트웨이 로드밸런서(GWLB)

- **GWLB를 사용하려면 다음 조건을 만족해야 한다.**
  - 로드밸런싱 대상은 GENEVE 프로토콜(UDP6081)을 지원하는 가상 어플라이언스여야 한다.
  - GWLB 엔드포인트 서비스(AWS PrivateLink)와 엔드포인트를 생성해야 한다.
  - 클라이언트 트래픽의 첫 진입점은 이 엔드포인트(그림의 vpce-4459)다.

7. 실습. ALB 생성과제

■ 그림의 표시 순서에 따라 실습해보자.



① **로드밸런싱 대상 인스턴스 2개를 만든다. 윈도우라면 서버 관리자에서 IIS를 설치하고 HTTP 페이지를 30080 포트로 바인딩한다.**

② **규칙없는 SG 2개를 만든다. ALB에 연결할 SG를 [sg-0123]이라고 하고 대상에 연결할 SG를 [sg-0987]이라 하자.**

- sg-0123 인바운드 : 소스는 클라이언트의 IP를 입력하고 포트는 [HTTP 80]을 입력한다.
- sg-0123 아웃바운드 : 목적지는 sg-0987을 입력하고 포트는 [HTTP 30080]을 입력한다.
- sg-0987 인바운드 : 소스는 sg-0123을 입력하고 포트는 [HTTP 30080]을 입력한다.

③ **서비스 > EC2 > 대상 그룹 메뉴에서 대상 그룹을 만든다.**

④ **서비스 > EC2 > 로드밸런서 메뉴에서 Application Load Balancer를 생성한다.**

Thank You