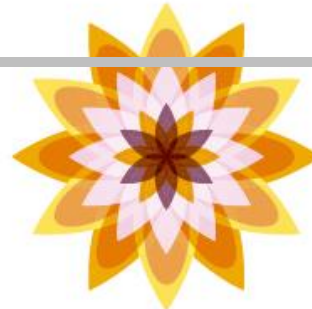
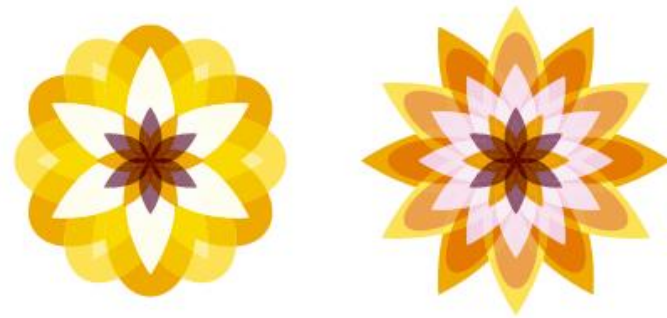


Chapter 07

연결 제어 1: VPC 통제 3요소



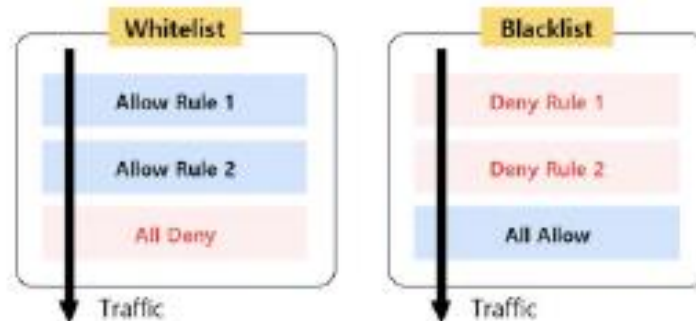
1. 접근 제어 : 보안 그룹과 네트워크 ACL

- 접근 제어(Access Control)는 컴퓨팅 서비스를 보호하는 안전 장치다.
- 쉽게 말해 필요한 트래픽만 허용하고 불필요하면 차단한다.
- 일반적으로 온프레미스 환경은 방화벽으로 접근을 제어한다.
- 인터넷 접점에서 외부 공격을 차단하는가 하면 서버망 전단에서 서버들을 보호하는 등 다양한 위치에서 트래픽을 허용하거나 차단하고 있다.
- VPC에서는 보안그룹과 네트워크 ACL이 방화벽 역할을 한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(1): Whitelist vs. Blacklist

- 방화벽의 접근 제어 방식은 크게 2가지다.
 - 화이트 리스트(Whitelist) 방식
 - 블랙 리스트(Blacklist) 방식
- 화이트 리스트 방식은 모든 트래픽을 기본 차단한 상태에서 접속이 필요한(화이트) 트래픽만 선별적으로 허용한다.
- 반면 블랙 리스트 방식은 모든 트래픽을 허용해 놓고 거부할(블랙) 트래픽만 선별해 차단하는 방식이다.
- 그림은 이 2가지 제어 방식을 나타낸다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

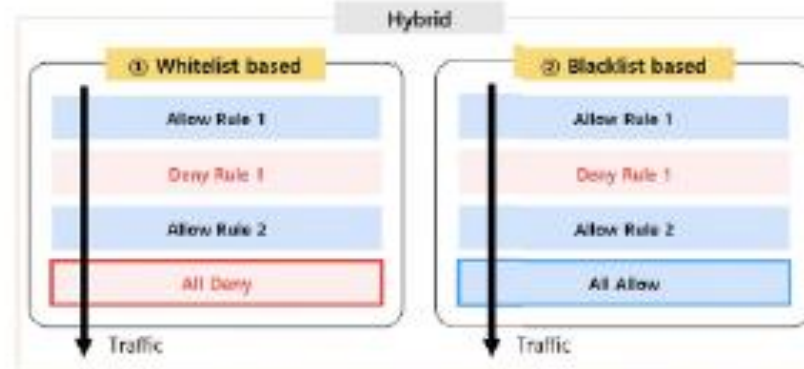
■ 접근 제어 방식 비교(1): Whitelist vs. Blacklist

- 화이트 리스트 정책에는 2개의 허용 규칙(화이트 리스트)과 모두 거부 규칙을 적용했다.
- 허용 규칙과 관련된 트래픽이 들어오면 통과시키고, 그 이외의 트래픽은 차단한다.
- 따라서 허용 규칙이 전혀 없다면 모든 트래픽은 차단될 것이다.
- 블랙 리스트 정책에는 2개의 거부 규칙(블랙 리스트)과 모두 허용 규칙을 적용했다.
- 거부 규칙과 관련된 트래픽이 들어오면 차단시키고, 그 이외의 트래픽은 허용한다.
- 따라서 거부 규칙이 전혀 없다면 모든 트래픽은 허용될 것이다.
- 그러나 현실적으로 허용과 차단 둘 다 필요하므로, 2개 방식을 결합한 형태로 사용한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(1): Whitelist vs. Blacklist

- 그림은 화이트 리스트 방식과 블랙 리스트 방식을 결합한 2가지 접근 제어를 나타낸다.



- 화이트 리스트 기반(Whitelist based) 결합 방식은 모두 거부 규칙을 최하단에 놓고 상단에 허용과 거부 규칙을 혼합 배치한다.
- 블랙 리스트 기반(Blacklist based) 결합 방식은 모두 허용 규칙을 최하단에 놓고 상단에 허용과 거부 규칙을 혼합 배치한다.
- 화이트 리스트 기반 결합 방식에서 거부 규칙(Deny Rule 1)이 없으면 완전한 화이트 리스트 방식이므로 화이트 리스트 방식은 화이트 리스트 기반 결합 방식의 일종이다.
- 블랙 리스트 방식도 마찬가지다.
- 따라서 하이브리드 여부와 무관하게 최하단에 모두 거부 규칙이 있으면 화이트 리스트 기반 결합 방식이고 모두 허용 규칙이 있으면 블랙 리스트 기반 결합 방식이라고 한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(1): Whitelist vs. Blacklist

- 혼동을 방지하고자 지금부터는 다음 4개 용어를 사용한다.
 - 화이트 리스트, 화이트 기반 결합
 - 블랙 리스트, 블랙 기반 결합
- 모든 트래픽을 기본 허용하는 2가지 블랙 방식은 관리자가 차단 대상을 모두 알고 있어야 하므로 관리가 까다로워 단독 사용이 어렵다.
- 따라서 화이트 리스트 방식과 이중 보안 체계를 구성하는 게 좋다.
- 단일로 사용하려면 블랙 리스트 방식이 아닌 화이트 리스트나 화이트 기반 결합 방식을 사용해야 한다.
- 화이트 리스트나 블랙 리스트 각 방식은 최하단 규칙을 제외한 모든 트래픽을 허용 또는 차단하므로, 규칙 적용순서가 중요하지 않다는 공통점이 있다.
- 그러나 결합 방식은 룰 적용 순서에 따라 허용 가능한 트래픽 범위가 달라진다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(1): Whitelist vs. Blacklist

- 그림은 화이트 기반 결함의 2가지 정책을 보여주고 있다.
- 왼쪽처럼 허용 규칙을 먼저 적용하면 [160.83.25.60]의 접근을 차단하지 못한다.
- 반면 오른쪽처럼 차단 규칙을 먼저 적용하면 [160.83.25.60]은 차단되고 나머지 IP만 접근할 수 있다.



- 이처럼 결함 방식은 규칙 적용 순서가 중요하므로 규칙마다 규칙 번호(Rule Number)가 있으며, 트래픽이 들어오면 낮은 번호부터 순차 적용된다.
- 온프레미스 방화벽은 규칙 번호를 시퀀스(SEQ)로 표현하기도 한다.
- AWS에서는 보안 그룹(Security Group, SG)이 화이트 리스트 방식을 사용하고 네트워크 ACL(Network Access Control List, NACL)이 결함 방식을 사용한다.
- 그러므로 규칙 순서가 중요한 NACL은 규칙 번호를 사용하고 규칙 순서가 무의미한 SG는 규칙 번호를 사용하지 않는다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

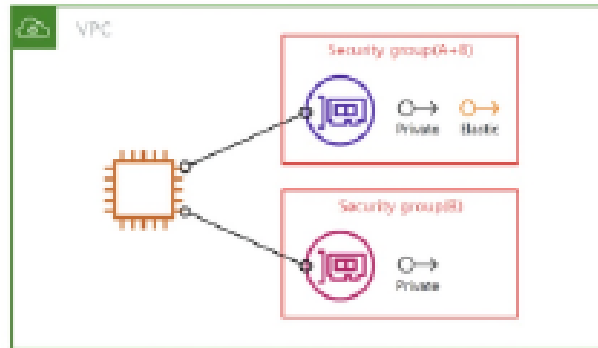
■ [SG] 표면적 특징과 다중 연결성(1 : N, N : 1)

- SG는 VPC의 보안통제 3요소중 하나로 ENI로 들어오거나 나가는 트래픽 접근을 제어한다.
- SG 특징은 다음과 같다.
 - SG의 부모는 VPC 이다.
 - SG의 연결 대상은 ENI이며, 수명 주기 동안 다른 ENI에 연결할 수 있다. 또 어떤 ENI에도 연결하지 않은 상태로 존재할 수 있다.
 - 반대로 컴퓨팅 ENI는 수명 주기 동안 반드시 SG와 연결돼 있어야 한다. 다시 말해 컴퓨팅 서비스 생성 시점에 SG를 지정해야 한다.
 - SG 는 두 가지 다중 연결 특징이 있다.
 - 1:N 연결성 : 1 개 SG를 여러 ENI에 연결할 수 있다. 역할마다 SG를 구분 생성하고 서비스 역할에 따라 관련 SG를 연결하면 유용하다.
 - N:1 연결성 : 여러 SG를 1개 ENI에 연결할 수 있다. 서비스 하나에 여러 역할이 필요할 때 유용하다.
 - 기본 VPC를 포함한 모든 VPC가 생성될 때 기본 SG도 함께 생성된다. 따라서 기본 SG 와 VPC 개수는 같다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 표면적 특징과 다중 연결성(1 : N, N : 1)

- 그림은 SG의 두 가지 연결 특징을 표현하고 있다.



- SG(B)가 기본 ENI와 추가 ENI에 각각 독립적으로 연결된 것은 SG와 ENI의 관계가 1 : N임을 나타낸다. 또 SG(A)와 SG(B)가 1개 ENI에 연결된 것으로 SG와 ENI의 관계가 N : 1인 것을 알 수 있다.
- SG의 부모는 VPC이므로, VPC 내의 모든 SG는 이 관계가 성립한다.
- 그럼 하나의 ENI에 여러 SG가 연결된 SG(A+B)는 어떤 방식으로 접근 제어를 할까?
- SG는 화이트 리스트 방화벽이므로 규칙 간 순서가 중요하지 않다고 한 바 있다.
- 따라서 SG(A)의 규칙과 SG(B)의 규칙 순서도 중요하지 않다.
- ENI에 SG(A)나 SG(B) 중 어떤 것을 먼저 연결해도 접근 제어 결과는 같다.
- 다시 말해 SG(A+B)는 SG(A)의 허용 규칙과 이의 허용 규칙을 순서없이 적용한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 규칙의 형태

- 온프레미스 방화벽의 규칙은 대개 그림의 형태를 띤다.

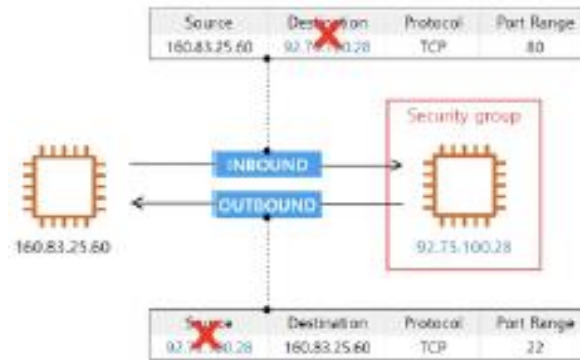
Allow/Deny	Source	Destination	Protocol	Port Range
Allow	160.83.25.60	92.75.100.28	TCP	80
Allow	92.75.100.28	160.83.25.60	TCP	22

- 출발지 IP와 목적지 IP, 프로토콜 유형과 포트 번호를 저장해 두고 유입 트래픽이 각 규칙과 일치하거나 겹치는 부분이 있으면 허용하거나 차단한다.
- 단, 4개 속성 모두 겹쳐야 한다.
- SG의 연결 대상은 ENI다.
- SG를 ENI에 연결한다는 것은 ENI에서 나가거나 들어오는 트래픽을 SG로 통제한다는 뜻이다.
- 그러므로 트래픽이 ENI로 들어올 땐 SG 규칙에 목적지가 필요없고, ENI에서 나갈 땐 출발지가 필요없다.
- 출발지와 목적지 모두 ENI의 IP이기 때문이다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 규칙의 형태

- 그림은 [92.75.100.28] 인스턴스에 연결된 SG 규칙이다.
- SG는 그림처럼 인바운드와 아웃바운드 규칙을 개별 관리한다.
- 인바운드 규칙에는 대상(목적지)이 인스턴스 자신이므로 대상을 지정할 필요가 없고, 반대로 아웃바운드 규칙에는 소스(출발지)가 인스턴스이므로 소스를 지정할 필요가 없다.
- 쉽게 말해 통신 대상(160.83.25.60)만 소스나 대상에 입력하면 된다.

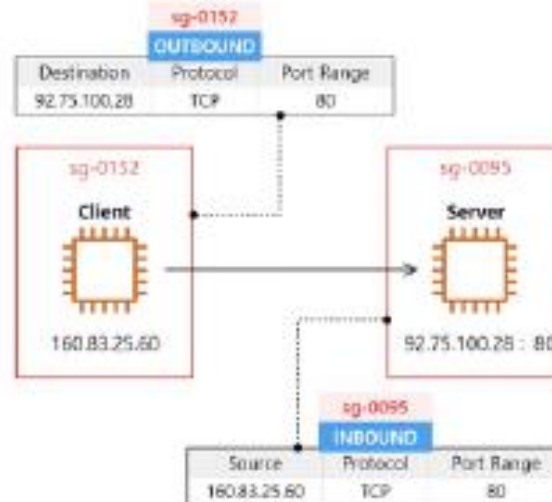


- 일반적으로 온프레미스 환경의 방화벽은 인바운드와 아웃바운드 규칙을 별도로 관리하진 않지만 소스와 대상 모두 규칙 하나에 입력해 트래픽을 제어한다는 점에서 SG와 차이가 있다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 소스/대상에 SG 허용

- 그림은 클라이언트가 서버로 접속하는 예시다.

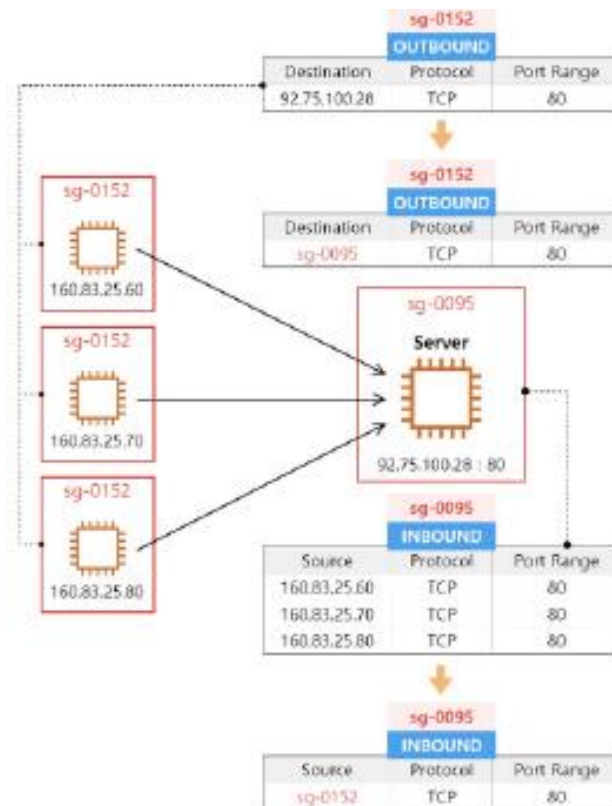


- 클라이언트(인스턴스)가 서버(인스턴스)에 접속하려면 클라이언트에 연결된 [sg-0152]에 서버(92.75.100.28)를 대상으로 하는 아웃바운드 규칙을 저장해야 한다.
- 반면 서버는 클라이언트의 접속을 허용해야 하므로 [sg-0095]에 클라이언트(160.83.25.60)를 소스로 하는 인바운드 규칙을 저장해야 한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 소스/대상에 SG 허용

- 서버에 접속할 클라이언트 수가 많아지면 접근 제어를 어떻게 해야 할까?
- 그림은 [sg-0152]를 사용하는 3개의 클라이언트가 서버에 접속하는 모습이다.
- 서버가 이 클라이언트들의 접속을 허용하려면 SG에 3개의 인바운드 규칙을 저장해야 한다.
- 클라이언트 수가 더 많아지면 규칙 수도 함께 늘어난다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

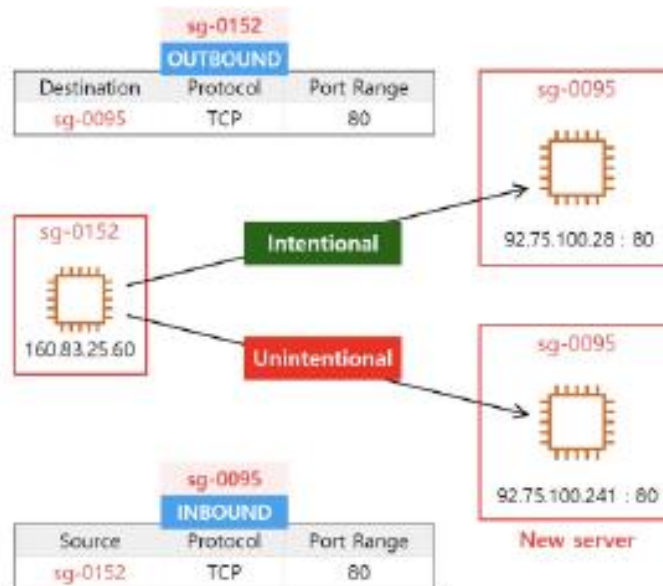
■ [SG] 소스/대상에 SG 허용

- AWS는 SG 규칙 관리 효율을 높이고자 소스와 대상에 SG를 지정할 수 있게 설계했다.
- 소스와 대상에 SG가 있으면 다음과 같이 해석한다.
 - 아웃바운드 규칙 대상의 SG : 해당 SG를 사용하는 컴퓨팅 서비스로 접속 허용
 - 인바운드 규칙 소스의 SG : 해당 SG를 사용하는 컴퓨팅 서비스의 접속 허용
- [sg-0152] 아웃바운드 대상을 [sg-0095]로 지정했으므로 [sg-0095]를 사용하는 [92.75.100.28]로 가는 트래픽을 허용한다.
- 또 [92.75.100.28]은 [sg-0152]의 인바운드 소스를 [sg-0152]로 지정해 [sg-0152]를 사용하는 3개 인스턴스를 허용하고 있다.
- 이처럼 SG의 소스와 대상에 SG를 지정하면 규칙 관리가 편리하며, 단순한 규칙만으로 다량의 IP를 허용할 수 있다.
- 그러나 문제는 SG를 사용하는 모든 서비스를 허용한다는 것이다.
- SG는 1 : N 연결 성질이 있기 때문이다.
- 컴퓨팅 ENI는 반드시 SG를 연결해야 한다.
- 인스턴스는 생성 단계에서 ENI를 장착하므로 SG 역시 인스턴스 생성 시 함께 선택, 연결토록 설계돼 있다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [SG] 소스/대상에 SG 허용

- 그림의 [92.75.100.241] 인스턴스를 생성할 때 이전에 생성된 [sg-0095]를 연결했다고 가정하면 새로운 허용 규칙을 입력하지 않아도 의도치 않은 허용 경로가 생긴다.
- 따라서 불필요 SG는 인스턴스에서 반드시 해제해야 하며, 사용하지 않는 SG는 다른 ENI에 연결할 수 없도록 주기적으로 확인하고 삭제해야 한다.
- 그러나 SG의 소스나 대상에 SG가 지정된 것은 그 SG를 사용하는 AWS 컴퓨팅 서비스를 허용한다는 뜻이므로 접속 대상도 AWS 내부로 한정된다.
- 그러므로 IP가 저장된 규칙보다 SG가 저장된 규칙에 좀 더 안심할 수는 있다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

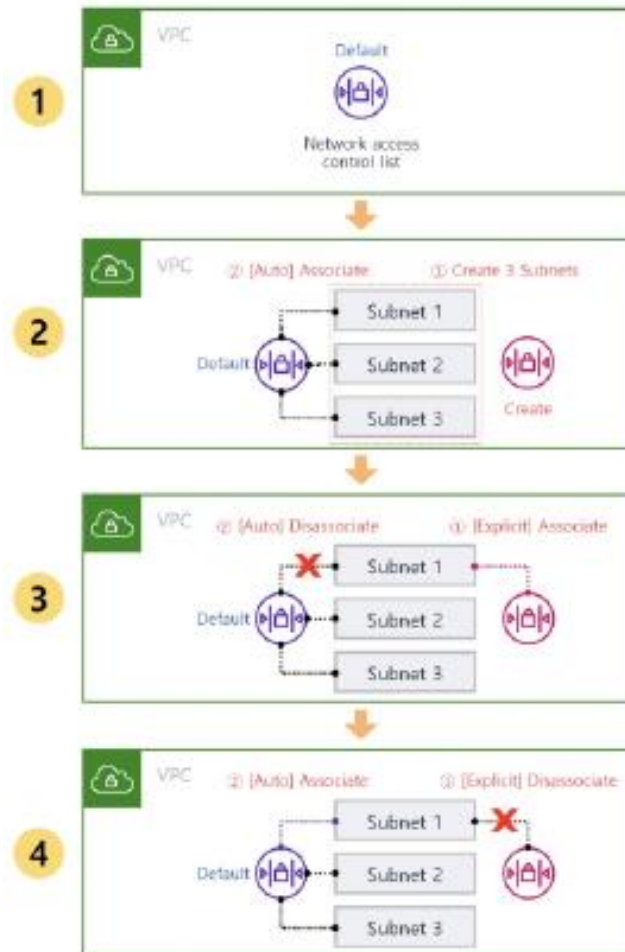
■ [NACL] 표면적 특징과 다중 연결성(1 : N)

- 네트워크 ACL(Network Access Control List, NACL)은 VPC의 보안 통제 3요소 중 하나로 서브넷을 통과하는 트래픽 접근을 제어한다.
- NACL의 특징은 다음과 같다.
 - NACL의 부모는 VPC 이다.
 - NACL의 연결 대상은 서브넷이며 수명 주기 동안 다른 서브넷에 연결할 수 있다. 또 어떤 서브넷에도 연결하지 않은 상태로 존재할 수 있다.
 - 반대로 서브넷은 수명 주기 동안 반드시 NACL과 연결돼 있어야 한다. 서브넷은 단 하나의 NACL을 사용하지만 다른 NACL로 바꿔 사용할 수도 있다.
 - 기본 VPC를 포함한 모든 VPC가 생성될 때 기본 NACL도 함께 생성된다. 따라서 기본 NACL과 VPC 개수는 같다.
 - 서브넷 생성 단계에서 서브넷에 연결할 NACL을 지정할 수 없다. 서브넷을 생성하면 무조건 기본 NACL에 자동 연결된다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [NACL] 표면적 특징과 다중 연결성(1 : N)

- 그림은 네트워크 ACL(NACL)의 특징을 보여준다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [NACL] 규칙의 형태

- NACL 규칙의 형태는 SG와 유사하다.
- 그러나 NACL은 허용과 거부 규칙을 결합한 제어 방식을 쓰므로 차단 규칙도 적용할 수 있다.
- 그림은 NACL의 규칙 예시다.
- NACL은 SG에게는 없는 규칙 번호와 허용/거부 속성이 있다.
- 결합 방식을 사용하므로 허용/거부 규칙이 나뉘져 있다.
- 또 규칙 적용 순서가 중요하므로 규칙 번호 순서에 따라 트래픽 접근을 제어한다.

규칙 번호	무엇	프로토콜	포트 범위	소스	허용/거부
100	HTTP(80)	TCP(6)	80	160.81.25.80/32	Allow
1	모든 트래픽	모두	모두	0.0.0.0/0	Deny

- NACL 최하단에는 삭제 불가능한 모두 차단 규칙이 적용돼 있다.
- 따라서 NACL의 기본 형태는 화이트 기반 결합 방식이다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ [NACL] 규칙의 형태

- 그럼 NACL을 블랙 기반 결함 방식으로 사용하려면 어떻게 해야 할까?
- 블랙 방식은 최하단에 모두 허용 규칙이 있어야 한다.
- NACL은 규칙 번호 순서로 트래픽을 제어하므로 그림처럼 모두 거부 규칙 상단에 모두 허용 규칙 [200번]을 적용하면 모두 거부 규칙은 무용지물이 되고 블랙 기반 결함 방식으로 활용할 수 있다.

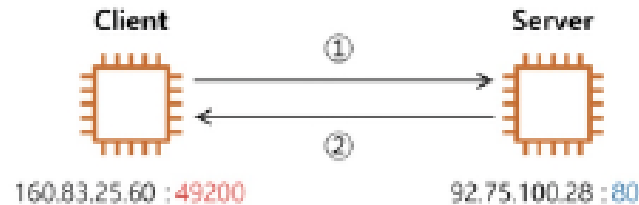
규칙 번호	유형	프로토콜	포트 범위	소스	허용/거부
100	HTTP(80)	TCP(80)	80	160.88.25.60/32	Deny
200	모든 트래픽	모두	모두	0.0.0.0/0	Allow
*	모든 트래픽	모두	모두	0.0.0.0/0	Deny

- 그림의 NACL 규칙 적용 순서는 [100] → [200] → *(모두 거부) 순이다.
- 여기서 200번 규칙은 최하단 모두 거부 규칙을 무력화하고, 100번에서 차단한 트래픽 외 모든 트래픽을 허용한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless

- TCP 통신은 3-way handshake 방식으로 논리적 세션을 형성하고, 클라이언트와 서버 사이 데이터를 주고 받는다.
- 이때 4가지 통신 요소가 필요한데, 그림은 이 4개 요소(클라이언트 IP와 PORT, 서버 IP와 PORT)를 나타낸다.



- 이로써 서버 입장에서 SG에 허용할 규칙은 다음 그림으로 요약할 수 있다.

INBOUND			OUTBOUND		
Source	Protocol	Port Range	Destination	Protocol	Port Range
160.83.25.60	TCP	80	160.83.25.60	TCP	49200

- 여기서 클라이언트가 운영체제에게 할당받는 [49200] 포트를 동적 포트(Dynamic Port) 또는 휘발성 포트(Ephemeral Port)라 한다.
- 클라이언트가 접속을 요청할 때마다 동적 포트 번호는 일정 범위 내에서 변한다.
- 운영체제 종류마다 이 기본 범위가 정해져 있으며, 그 범위를 변경할 수도 있다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless

- 그림은 윈도우에 기본 설정된 TCP 동적 포트 범위를 나타낸다.
- 49152부터 16384개를 사용할 수 있으므로 49152~65535 범위 중 하나를 클라이언트에게 할당할 것이다.

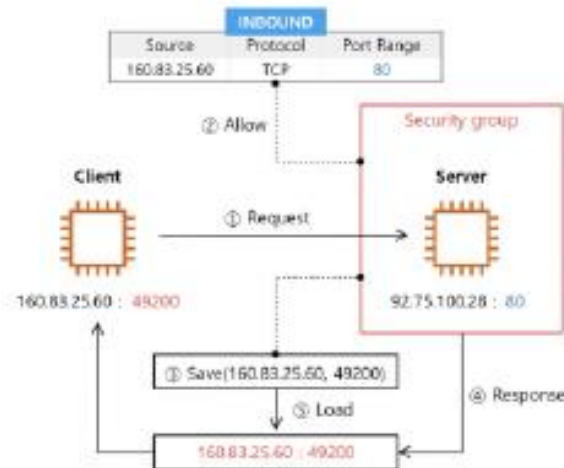
```
C:\Users\Administrator>netsh int ipv4 show dynamicport tcp  
Protocol tcp Dynamic Port Range  
-----  
Start Port      : 49152  
Number of Ports : 16384
```

- 그러나 SG는 최초 접속 규칙만 입력해도 통신할 수 있도록 설계돼 있다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless

- 클라이언트가 SG를 통과해 서버에 접속하는 과정을 그림에서 확인해보자.

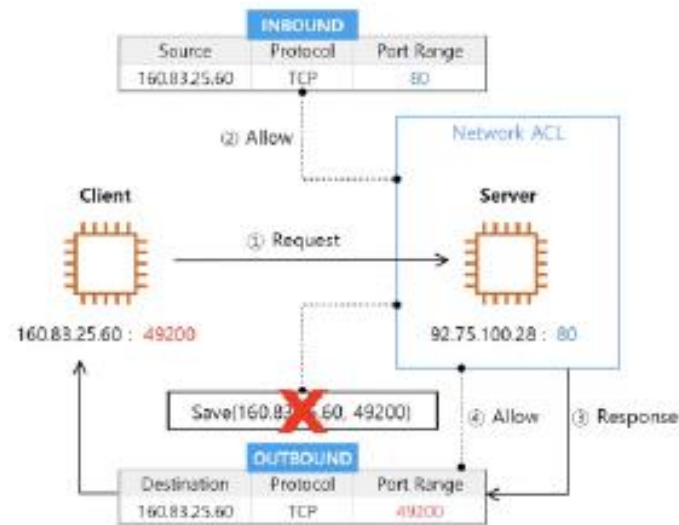


- 이처럼 클라이언트의 IP와 포트를 저장하는 기법을 상태 저장(Stateful) 방식이라고 하며 그 반 대의 경우를 상태 비저장(Stateless) 방식이라고 한다.
- SG는 상태 저장 방식을 사용하고 NACL은 상태 비저장 방식을 사용한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless


- 그림은 상태 비저장 방식을 사용하는 NACL의 패킷 허용 원리를 나타낸다.
- NACL은 클라이언트 정보를 저장하지 않으므로, 서버의 응답이 향하는 목적지(160.83.25.60)와 포트(49200)를 아웃바운드 규칙에 허용해 줘야 한다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless

- NACL에 인바운드 규칙만 저장돼 있으면 통신이 불가능할 것이다.
- 그럼 아웃바운드 규칙에는 어떤 포트를 허용해야 할까?
- 앞서 설명한 운영체제의 동적 포트를 그림과 같이 적용해야 한다.



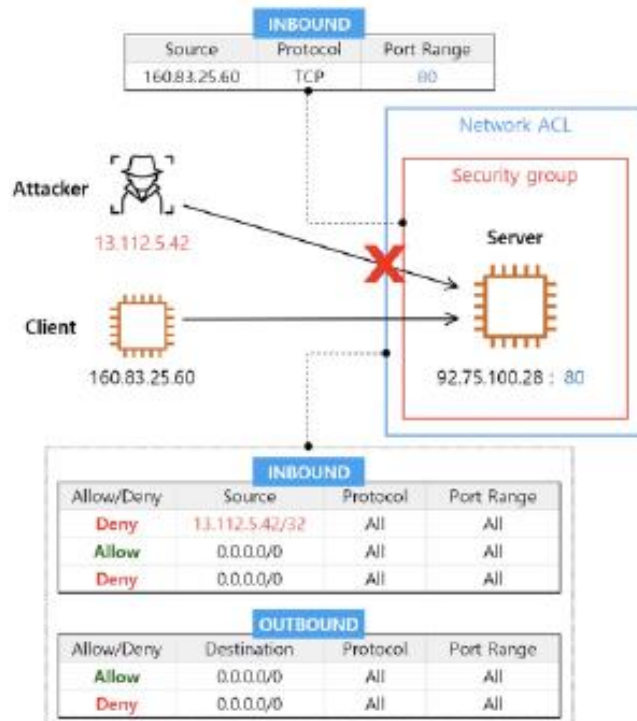
규칙 번호	유형	프로토콜	포트 범위	대상	허용/거부
100	사용자 지정 TCP	TCP(6)	49152 - 65535	160.85.25.60/32	Allow
*	모든 트래픽	모든	모든	0.0.0.0/0	Deny

- 하지만 그림은 특정 윈도우 버전에 한정된 규칙이다.
- 또 운영체제 종류마다 동적 포트 범위가 다르다.
- 서버는 자신에게 접속하는 클라이언트의 운영체제 종류를 모두 알 수 없으므로, 포트 범위를 제한하는 방법으로는 원활한 서비스 제공이 어렵다.
- 그러므로 모든 포트 허용을 권장한다.
- NACL을 화이트 방식으로 사용하면 다음과 같은 문제가 발생한다.
 - 인바운드 규칙에 허용된 IP를 아웃바운드 규칙에도 적용해야 한다.
 - 이때 클라이언트의 동적 포트를 허용해야 한다.
 - SG에 신규 허용 규칙을 등록할 때마다 NACL도 함께 등록해야 한다. 다시 말해 NACL은 서브넷에 속한 모든 서비스의 접근 제어에 관여해야 한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 접근 제어 방식 비교(2): Stateful vs. Stateless

- 이 문제를 해결하는 접근 제어 방법은 다음과 같다.
 - 개별 인스턴스의 접근 제어는 SG로 관리하고 서브넷 접근은 NACL로 관리한다.
 - 이때 NACL은 블랙 기반 결합 방식을 사용한다. NACL 최하단에 모두 허용 규칙을 적용한 후 블랙 방식 NACL로 변경하고, 서브넷에 속한 인스턴스가 공통으로 차단할 트래픽(13.112.5.42)만 NACL에 적용한다.
 - NACL을 화이트 기반 결합으로 꾸며 보안 수준을 보다 향상시킬 수도 있다. 단, 클라이언트에게 응답하는 규칙은 모든 포트를 지정해야 한다.
 - 화이트 기반 결합 방식은 왼쪽 그림의 NACL을 오른쪽 그림으로 대체하면 된다.



INBOUND			
Allow/Deny	Source	Protocol	Port Range
Deny	13.112.5.42/32	All	All
Allow	160.83.25.60/32	TCP	80
Deny	0.0.0.0/0	All	All

OUTBOUND			
Allow/Deny	Destination	Protocol	Port Range
Allow	160.83.25.60/32	TCP	All
Deny	0.0.0.0/0	All	All

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ SG vs. NACL 비교

- SG와 NACL의 부모는 VPC이며 인바운드와 아웃바운드 규칙을 개별 관리한다는 공통점이 있다.
- 그 외 다른 특징을 다음 표에서 비교해보자.

특징 \ 접근 제어	보안 그룹 (SG)	네트워크 ACL (NACL)
연결(동제) 대상	네트워크 인터페이스(ENI)	서브넷
다중 연결성	1:N, N:1	1:N
접근 제어 방식	화이트 리스트	블랙 기반 결합 또는 화이트 기반 결합
		허용 또는 거부
		○ (있음)
상태 저장	저장(Stateful)	비저장(Stateless)
소스/대상 허용	SG 허용 가능	NACL 허용 불가

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ SG와 NACL 바르게 사용하기

- 클라우드의 온프레미스와 달리 외부에 노출돼 있으며, 기업에 따라 차이는 있겠지만 별도 결재나 허가 없이도 쉽게 설정 변경을 할 수 있다.
- 그러므로 관리자의 보안 의식 수준이 매우 중요하다.
- 특히 VPC 통제 3요소의 관리 수준은 VPC 보안을 좌우한다.
- SG 관리 유의점
 - SG의 다중 연결성(1:N, N:1)은 VPC 네트워킹 구축과 관리 효율을 높여준다. 그러나 이 특징은 방화벽 정책 복제와 같은 맥락이므로, 반드시 필요한 서비스에만 연결해야 한다.
 - 서비스 생성 시점에 서비스와 무관한 SG를 연결할 수 있으므로, 과거에 생성한 미사용 SG는 주기적으로 확인, 삭제해야 한다. ENI가 사용하지 않는 모든 SG가 삭제 대상이다. API를 활용해 미사용 SG를 빠르게 추출할 수 있다.
 - 서비스에 불필요하게 연결된 SG를 점검한다. SG가 2개 이상 연결된 서비스는 부주의로 연결됐을 가능성이 크므로 더 세밀히 확인한다.
 - 1:N 다중 연결성은 반드시 필요할 때만 활용한다. 서비스마다 허용할 소스와 대상이 다른 데도 운영 편의성만 고려해 소수의 SG에 서비스의 허용 규칙을 다량 적용하면 안된다. 다시 말해 허용 규칙에 지정된 소스와 대상이 서비스와 일부라도 무관하면 반드시 새로운 SG를 만들어 연결해야 한다.
 - 소스와 대상이 SG로 지정되지 않은 규칙을 점검한다. 특히 접속 대상이 퍼블릭 CIDR이면 허용 범위의 적절성과 사용 목적을 확인해야 한다.
 - 소스와 대상에 SG를 지정한 규칙은 접속 대상 서비스 규모와 범위가 적절한지 확인한다.
 - SG는 상태 저장 방식(Stateful)이므로 응답 패킷용 허용 규칙은 불필요하다.
 - 특히 아웃바운드에 모두 허용 규칙이나 과도한 IP를 허용하는 모든 TCP, 모든 UDP 규칙은 반드시 검사해서 불필요하면 삭제한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ SG와 NACL 바르게 사용하기

■ 네트워크 ACL(Network ACL, NACL) 관리 유의점

- 서브넷 생성 시점에 자동 연결된 기본 NACL은 기존 등록된 규칙 때문에 불필요 트래픽을 허용할 수 있다. 따라서 서브넷 생성 직후 연결된 NACL의 규칙을 반드시 점검해야 한다.
- 서브넷마다 허용 또는 차단할 대상이 일부라도 다르면 반드시 새로운 NACL을 만들어 연결해야 한다.
- 서브넷의 접속 대상이 퍼블릭 CIDR이면 허용 범위의 적절성과 사용 목적을 확인해야 한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 서비스 > EC2 > 보안그룹 메뉴에서 보안그룹 생성 버튼을 클릭한다.

▼ 네트워크 및 보안

보안 그룹

탄력적 IP

배치 그룹

키 페어

네트워크 인터페이스

보안 그룹 (3) 정보

↺

작업 ▼

보안 그룹을 CSV로 내보내기 ▼

보안 그룹 생성

🔍

보안 그룹 필터링

<

1

>

⚙

<input type="checkbox"/>	Name ▼	보안 그룹 ID ▼	보안 그룹 이름 ▼	VPC ID ▼	설명 ▼	소유자
<input type="checkbox"/>	-	sg-e21ceeff	default	vpc-f4a4c989 🔗	default VPC security gr...	26266376
<input type="checkbox"/>	-	sg-08627bacf6e6fddc6	launch-wizard-1	vpc-f4a4c989 🔗	launch-wizard-1 create...	26266376
<input type="checkbox"/>	-	sg-03dcad201bfa5c653	launch-wizard-2	vpc-f4a4c989 🔗	launch-wizard created ...	26266376

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 보안 그룹 이름, 설명을 입력하고 VPC를 선택한다.
- Web Server용이므로 아웃바운드에 저장된 규칙을 삭제하고, 인바운드에서 규칙 추가 버튼을 클릭한다.

기본 세부 정보

보안 그룹 이름 정보

MyWebServerSG

생성 후에는 이름을 편집할 수 없습니다.

설명 정보

개발자에게 SSH 액세스 허용

VPC 정보

Q vpc-f4a4c989 X

인바운드 규칙 정보

이 보안 그룹에는 인바운드 규칙이 없습니다.

규칙 추가

아웃바운드 규칙 정보

유형 정보	프로토콜 정보	포트 범위 정보	대상 정보	설명 - 선택 사항 정보
모든 트래픽 ▼	전체	전체	사용자 ... ▼ Q	
0.0.0.0/0 X				<div>삭제</div>

규칙 추가

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 유형은 HTTP를 선택하고, 소스는 [160.83.25.60/32]를 입력한다.

The screenshot shows the '인바운드 규칙 정보' (Inbound Rule Information) form. It has five tabs: '유형 정보' (Type Information), '프로토콜 정보' (Protocol Information), '포트 범위 정보' (Port Range Information), '소스 정보' (Source Information), and '설명 - 선택 사항 정보' (Description - Optional Information). The '유형 정보' tab is active, showing a dropdown menu with 'HTTP' selected. The '소스 정보' tab shows a search box with '160.83.25.60/32' entered. The '규칙 추가' (Add Rule) button is highlighted with a red box.

- 아래 규칙 추가 버튼을 클릭해 이전과 같은 방법으로 HTTP 유형과 [175.100.203.22/32] 소스를 입력한 규칙을 더 생성한다.
- 완료되면 하단 보안 그룹생성 버튼을 클릭한다.

The screenshot shows the '인바운드 규칙 정보' (Inbound Rule Information) form with two rules listed. The first rule has 'HTTP' as the type and '160.83.25.60/32' as the source. The second rule, which is highlighted with a red box, also has 'HTTP' as the type and '175.100.203.22/32' as the source. The '규칙 추가' (Add Rule) button is visible at the bottom left.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 생성한 SG를 클릭해 화면 하단에 인바운드 규칙탭의 내용을 확인하고 SG ID(sg-0e84s04dda5d63c8d)를 복사한다.

sg-08ee1e77a33006d76 - MyWebServerSG 작업 ▼

세부 정보

보안 그룹 이름 MyWebServerSG	보안 그룹 ID sg-08ee1e77a33006d76	설명 Allow HTTP Traffic	VPC ID vpc-f4a4c989
소유자 262663767358	인바운드 규칙 수 2 권한 항목	아웃바운드 규칙 수 0 권한 항목	

인바운드 규칙 | 아웃바운드 규칙 | 태그

이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다. Reachability Analyzer 실행 ×

인바운드 규칙 (2) ↻ 태그 관리 인바운드 규칙 편집

Q 보안 그룹 규칙 필터

<input type="checkbox"/>	Name ▼	보안 그룹 규칙 ID ▼	IP 버전 ▼	유형 ▼	프로토콜 ▼	포트 범위
<input type="checkbox"/>	-	sgr-0c791e0018bac0d07	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-028a53d2ed62a88...	IPv4	HTTP	TCP	80

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 상기 과정을 반복해 Web Client용 SG(sg-0aed3a94e8aca98ac)를 추가 생성한다.
- 인바운드 규칙은 비워두고 아웃바운드 규칙 대상을 Web Server용 SG(sg-0e84a04dda5d63c8d)로 지정한다.

기본 세부 정보

보안 그룹 이름 정보

MyWebClientSG

정명 후에는 이름을 편집할 수 없습니다.

설명 정보

For Web Server Access

VPC 정보

Q vpc-f4a4c989 X

인바운드 규칙 정보

이 보안 그룹에는 인바운드 규칙이 없습니다.

규칙 추가

아웃바운드 규칙 정보

유형 정보

모든 트래픽 ▼

프로토콜 정보

전체

포트 범위 정보

전체

대상 정보

사용자 ... ▼

Q

0.0.0.0/0 X

설명 - 선택 사항 정보

삭제

규칙 추가

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 상기 과정을 반복해 Web Client용 SG(sg-0aed3a94e8aca98ac)를 추가 생성한다.
- 인바운드 규칙은 비워두고 아웃바운드 규칙 대상을 Web Server용 SG(sg-0e84a04dda5d63c8d)로 지정한다.

아웃바운드 규칙 정보

유형 정보	프로토콜 정보	포트 범위 정보	대상 정보	설명 - 선택 사항 정보
HTTP	TCP	80	사용자 ... sg-08ee1e77a33006d76	<input type="text"/> 삭제

규칙 추가

- 완료되면 하단 보안 그룹 생성 버튼을 클릭한다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- Web Server용 SG(sg-0e84a04dda5d63c8d)의 인바운드 규칙을 편집해 Web Client용 SG(sg-0aed3a94e8aca98ac) 를 소스로 지정한 규칙을 추가한다.

sg-08ee1e77a33006d76 - MyWebServerSG

세부 정보 | **인바운드 규칙** | 아웃바운드 규칙 | 태그

이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다. [Reachability Analyzer 실행](#)

인바운드 규칙 (2) [태그 관리](#) **인바운드 규칙 편집**

보안 그룹 규칙 필터

<input type="checkbox"/>	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위
<input type="checkbox"/>	-	sgr-0c791e0018bac0d07	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-028a53d2ed62a88...	IPv4	HTTP	TCP	80

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- Web Server용 SG(sg-0e84a04dda5d63c8d)의 인바운드 규칙을 편집해 Web Client용 SG(sg-0aed3a94e8aca98ac) 를 소스로 지정한 규칙을 추가한다.

인바운드 규칙 편집 정보

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

인바운드 규칙 정보

보안 그룹 규칙 ID	유형 <small>정보</small>	프로토콜 <small>정보</small>	포트 범위 <small>정보</small>	소스 <small>정보</small>	설명 - 선택 사항 <small>정보</small>	
sgr-0c791e0018bac0d07	HTTP	TCP	80	사용자 ...	175.100.203.22/32	삭제
sgr-028a53d2ed62a884c	HTTP	TCP	80	사용자 ...	160.83.25.60/32	삭제
-	HTTP	TCP	80	사용자 ...	sg-03b8a81f5929bb951	삭제

규칙 추가

취소 변경 사항 미리 보기 **규칙 저장**

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- ENI 생성 예제에서 만든 ENI를 확인한다.
- 서비스 > EC2 > 네트워크 인터페이스 메뉴에서 해당 ENI를 선택하고 화면 하단에 네트워크 인터페이스 세부 정보를 클릭한다.

네트워크 인터페이스 세부 정보		
네트워크 인터페이스 ID	이름	설명
eni-051674dbd2135bdfd	-	NEW ENI(Elastic Network Interface)
네트워크 인터페이스 상태	인터페이스 유형	보안 그룹
Available	interface	sg-06475b590714a117e (default)
VPC ID	서브넷 ID	가용 영역
vpc-0514155126c806195	subnet-04cb035c6534817fa	ap-northeast-2b

- ENI를 생성하면서 연결한 SG를 확인할 수 있다.
- 이번에 새로 만든 SG를 이 ENI에 추가 연결해본다.

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- ENI 생성 예제에서 만든 ENI를 확인한다.
- 서비스 > EC2 > 네트워크 인터페이스 메뉴에서 해당 ENI를 선택하고 화면 하단에 네트워크 인터페이스 세부 정보를 클릭한다.

▼ 네트워크 및 보안

보안 그룹

탄력적 IP

배치 그룹

키 페어

네트워크 인터페이스

네트워크 인터페이스: eni-0769ee14d88fc6038

세부 정보

플로우 로그

태그

이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다.

Reachability Analyzer 실행

▼ 네트워크 인터페이스 세부 정보

네트워크 인터페이스 ID

eni-0769ee14d88fc6038

네트워크 인터페이스 상태

Available

VPC ID

vpc-f4a4c989

소유자

262663767358

이름

-

인터페이스 유형

탄력적 네트워크 인터페이스

서브넷 ID

subnet-31633c10

요청자 ID

-

설명

New ENI

보안 그룹

sg-e21ceeff (default)

가용 영역

us-east-1a

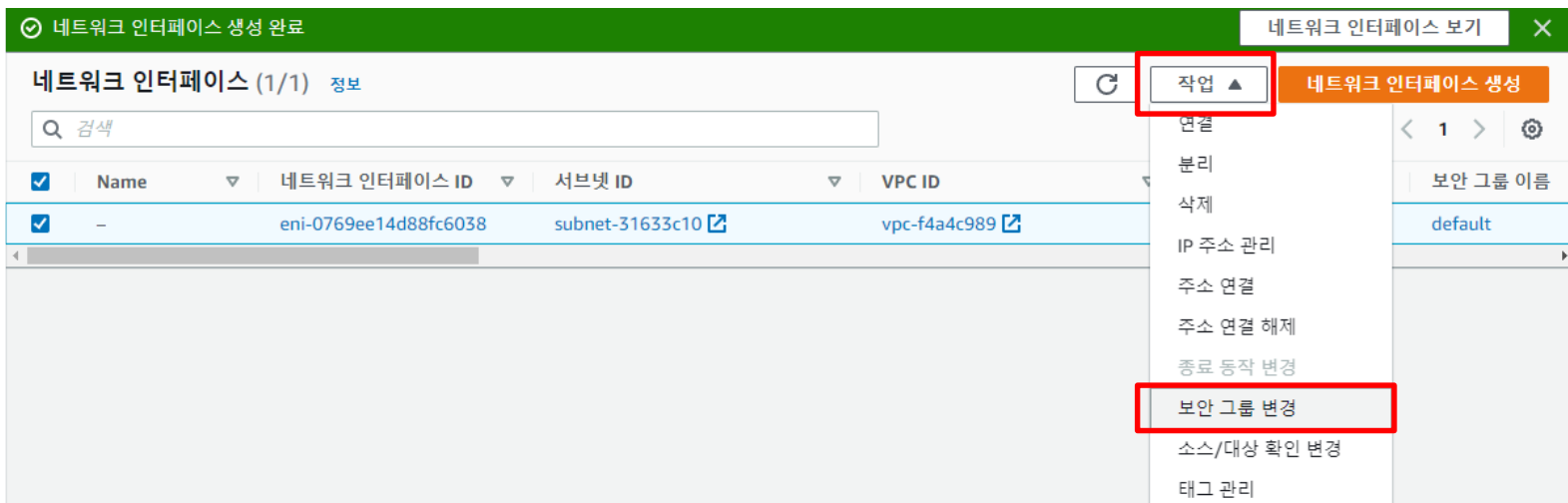
요청자 관리형

아니요

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- ENI를 생성하면서 연결한 SG를 확인할 수 있다.
- 이번에 새로 만든 SG를 이 ENI에 추가 연결해본다.
- ENI를 선택한 상태에서 작업 > 보안그룹 변경을 클릭한다.



1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 보안 그룹 선택을 클릭해 Web Server용 SG를 선택하고, 우측 보안 그룹 추가 버튼을 클릭한다.

EC2 > 네트워크 인터페이스 > eni-0769ee14d88fc6038 > 보안 그룹 변경

보안 그룹 변경 정보

Amazon EC2는 선택한 보안 그룹의 모든 규칙을 평가하여 인스턴스에서 송수신되는 인바운드 및 아웃바운드 트래픽을 제어합니다. 이 창을 사용하여 보안 그룹을 추가 및 제거할 수 있습니다.

네트워크 인터페이스 세부 정보

네트워크 인터페이스 ID
eni-0769ee14d88fc6038

연결된 보안 그룹

네트워크 인터페이스에 하나 이상의 보안 그룹을 추가합니다. 보안 그룹을 제거할 수도 있습니다.

MyWebClientSG (sg-03b8a81f5929bb951) MyWebClientSG	<input type="button" value="제거"/>
default (sg-e21ceeff) default	
launch-wizard-1 (sg-08627bacf6e6fddc6) launch-wizard-1	
MyWebServerSG (sg-08ee1e77a33006d76) MyWebServerSG	
launch-wizard-2 (sg-03dcad201bfa5c653) launch-wizard-2	

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- 연결 대상 SG 2개를 확인하고 하단 저장을 클릭한다.

EC2 > 네트워크 인터페이스 > eni-0769ee14d88fc6038 > 보안 그룹 변경

보안 그룹 변경 정보

Amazon EC2는 선택한 보안 그룹의 모든 규칙을 평가하여 인스턴스에서 송수신되는 인바운드 및 아웃바운드 트래픽을 제어합니다. 이 창을 사용하여 보안 그룹을 추가 및 제거할 수 있습니다.

네트워크 인터페이스 세부 정보

네트워크 인터페이스 ID
eni-0769ee14d88fc6038

연결된 보안 그룹

네트워크 인터페이스에 하나 이상의 보안 그룹을 추가합니다. 보안 그룹을 제거할 수도 있습니다.

✕ 보안 그룹 추가

네트워크 인터페이스와 연결된 보안 그룹(en-0769ee14d88fc6038)

보안 그룹 이름	보안 그룹 ID	
default	sg-e21ceeff	제거
MyWebServerSG	sg-08ee1e77a33006d76	제거

취소 저장

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. SG 생성 예제

- ENI에 연결된 2개 SG를 확인할 수 있다.

네트워크 인터페이스: eni-0769ee14d88fc6038

세부 정보

플로우 로그

태그

▼ 네트워크 인터페이스 세부 정보

네트워크 인터페이스 ID

eni-0769ee14d88fc6038

네트워크 인터페이스 상태

Available

VPC ID

vpc-f4a4c989

소유자

262663767358

이름

-

인터페이스 유형

탄력적 네트워크 인터페이스

서브넷 ID

subnet-31633c10

요청자 ID

-

설명

New ENI

보안 그룹

sg-e21ceff (default)

sg-08ee1e77a33006d76 (MyWebServerSG)

가용 영역

us-east-1a

요청자 관리형

아니요

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 서브넷 생성 예제에서 만든 서브넷 6개를 확인한다.
- 모두 기본 NACL과 연결돼 있다.
- 서비스 > VPC > 서브넷

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

캐리어 게이트웨이

DHCP 옵션 세트

탄력적 IP

관리형 접두사 목록

엔드포인트

엔드포인트 서비스

NAT 게이트웨이

피어링 연결

서브넷 (6) 정보					서브넷 생성
Q 서브넷 필터링					< 1 > ⚙
<input type="checkbox"/>	Name	서브넷 ID	IPv4 CIDR	네트워크 ACL	
<input type="checkbox"/>	PRI-WAS-2b-92.75.20	subnet-098a731873c43792e	92.75.20.0/24	acl-0631fd4d1c5ce0a60	
<input type="checkbox"/>	PUB-WEB-2b-92.75.200	subnet-04cb036c6534817fa	92.75.200.0/24	acl-0631fd4d1c5ce0a60	
<input type="checkbox"/>	PRI-WAS-2a-92.75.10	subnet-0b985a2d487fd363c	92.75.10.0/24	acl-0631fd4d1c5ce0a60	
<input type="checkbox"/>	PUB-WEB-2a-92.75.100	subnet-0765bc7c7bd1b3b09	92.75.100.0/24	acl-0631fd4d1c5ce0a60	
<input type="checkbox"/>	PRI-DB-2a-92.75.1	subnet-0a8f690278a26e95f	92.75.1.0/24	acl-0631fd4d1c5ce0a60	
<input type="checkbox"/>	PRI-DB-2b-92.75.2	subnet-0e556740d35b9b5b6	92.75.2.0/24	acl-0631fd4d1c5ce0a60	

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 그림에 나타난 NACL 링크를 클릭해 인바운드 규칙을 확인한다.

인바운드 규칙 (2)							인바운드 규칙 편집
Q 인바운드 규칙 필터링							< 1 > ⚙
규칙 번호	유형	프로토콜	포트 범위	소스	허용/거부		
100	모든 트래픽	모두	모두	0.0.0.0/0	✓	Allow	
*	모든 트래픽	모두	모두	0.0.0.0/0	✗	Deny	

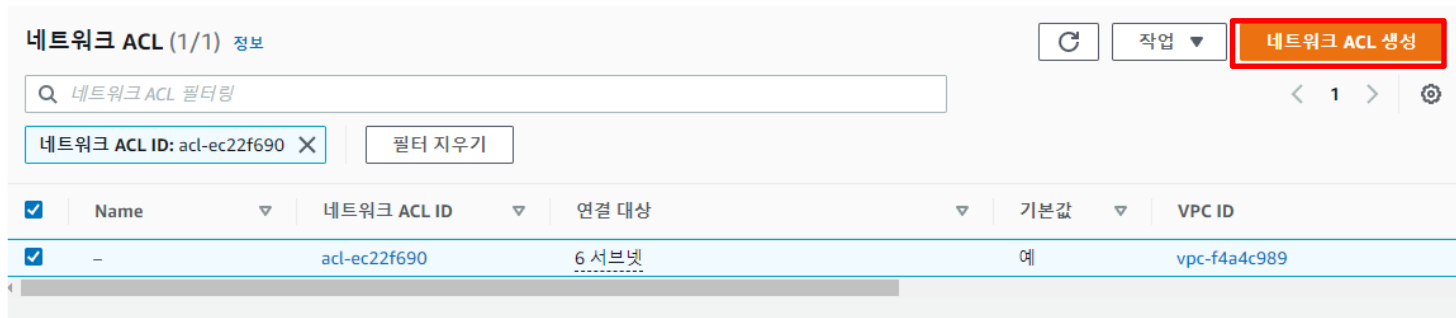
- 모두 허용 규칙이 저장돼 있으므로 블랙 기반 결함 방식이다.
- 아웃바운드 규칙도 이와 같다.

아웃바운드 규칙 (2)							아웃바운드 규칙 편집
Q 아웃바운드 규칙 필터링							< 1 > ⚙
규칙 번호	유형	프로토콜	포트 범위	대상	허용/거부		
100	모든 트래픽	모두	모두	0.0.0.0/0	✓	Allow	
*	모든 트래픽	모두	모두	0.0.0.0/0	✗	Deny	

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 우측 상단 네트워크 ACL 생성 버튼을 클릭한다.



네트워크 ACL (1/1) 정보

네트워크 ACL 필터링

네트워크 ACL ID: acl-ec22f690 ✕ 필터 지우기

<input checked="" type="checkbox"/>	Name	네트워크 ACL ID	연결 대상	기본값	VPC ID
<input checked="" type="checkbox"/>	-	acl-ec22f690	6 서브넷	예	vpc-f4a4c989

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 이름과 VPC를 입력한 뒤, 우측 하단의 네트워크 ACL 생성 버튼을 클릭한다.

VPC > 네트워크 ACL > 네트워크 ACL 생성

네트워크 ACL 생성 정보

네트워크 ACL은 서브넷 내부와 외부의 트래픽을 제어하기 위한 방화벽 역할을 하는 선택적 보안 계층입니다.

네트워크 ACL 설정

이름 - 선택 사항
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

Public-Subnet-NACL

VPC
이 네트워크 ACL에 사용할 VPC입니다.

vpc-f4a4c989

태그

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키

값 - 선택 사항

Q Name X Q Public-Subnet-NACL X 제거

새 태그 추가

49종(류) 태그.개 더 추가할 수 있습니다.

취소 **네트워크 ACL 생성**

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 신규 NACL과 기본 NACL/을 비교해본다.
- 새로 만든 NACL은 연결된 서브넷이 없으며 기본 NACL이 아님을 보여준다.

네트워크 ACL (1/2) 정보						작업 ▼	네트워크 ACL 생성
Q 네트워크 ACL 필터링						< 1 > ⚙	
<input type="checkbox"/>	Name ▼	네트워크 ACL ID ▼	연결 대상 ▼	기본값 ▼	VPC ID		
<input type="checkbox"/>	Public-Subnet-NACL	acl-01c19b985a9b79b1a	-	아니요	vpc-f4a4c989		
<input checked="" type="checkbox"/>	-	acl-ec22f690	6 서브넷	예	vpc-f4a4c989		

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 새로 생성한 NACL은 모두 거부 규칙만 최하단에 저장돼 있어 화이트 기반 결합 방식이다.
- 아웃바운드 규칙도 이와 같다.

acl-01c19b985a9b79b1a / Public-Subnet-NACL

세부 정보 | **인바운드 규칙** | 아웃바운드 규칙 | 서브넷 연결 | 태그

인바운드 규칙 (1) 인바운드 규칙 편집

Q 인바운드 규칙 필터링 < 1 > ⚙

규칙 번호	유형	프로토콜	포트 범위	소스	허용/거부
*	모든 트래픽	모두	모두	0.0.0.0/0	Deny

acl-01c19b985a9b79b1a / Public-Subnet-NACL

세부 정보 | 인바운드 규칙 | **아웃바운드 규칙** | 서브넷 연결 | 태그

아웃바운드 규칙 (1) 아웃바운드 규칙 편집

Q 아웃바운드 규칙 필터링 < 1 > ⚙

규칙 번호	유형	프로토콜	포트 범위	대상	허용/거부
*	모든 트래픽	모두	모두	0.0.0.0/0	Deny

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 인바운드와 아웃바운드 규칙을 편집해 클라이언트 접속을 허용한다.
- 우선 인바운드 규칙 편집 > 새 규칙 추가를 클릭한다.
- 그림과 같이 입력한 뒤 우측 하단 변경 사항 저장 버튼을 클릭한다.

인바운드 규칙		
규칙 번호 정보	유형 정보	프로토콜 정보
100	HTTP(B0)	TCP(6)
포트 범위 정보	소스 정보	허용/거부 정보
80	160.83.25.60/32	허용

- 아웃바운드 규칙도 ⑥과 같은 방법으로 편집한다.
- 유형값이 다르다는 점에 유의하자.

아웃바운드 규칙		
규칙 번호 정보	유형 정보	프로토콜 정보
100	모든 TCP	TCP(6)
포트 범위 정보	대상 정보	허용/거부 정보
모두	160.83.25.60/32	허용

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 변경한 규칙을 확인한다.

인바운드 규칙 (2)						
Q 인바운드 규칙 필터링						
규칙 번호	유형	프로토콜	포트 범위	소스	허용/거부	
100	HTTP(80)	TCP(S)	80	160.83.25.60/32	Allow	
*	모든 트래픽	모두	모두	0.0.0.0/0	Deny	

아웃바운드 규칙 (2)						
Q 아웃바운드 규칙 필터링						
규칙 번호	유형	프로토콜	포트 범위	대상	허용/거부	
100	모든 TCP	TCP(S)	모두	160.83.25.60/32	Allow	
*	모든 트래픽	모두	모두	0.0.0.0/0	Deny	

- 변경 완료한 NACL을 퍼블릭 서브넷에 연결해보자.
- 서브넷 연결 편집을 클릭한다.

네트워크 ACL (1/2) 정보				작업
Q 네트워크 ACL 필터링				네트워크
				세부 정보 보기
				인바운드 규칙 편집
				아웃바운드 규칙 편집
				서브넷 연결 편집
Name	네트워크 ACL ID	연결 대상		
Public-Subnet-NACL	acl-0f25698e011650066	-		

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- NACL을 변경할 서브넷을 선택한 후 변경 사항저장 버튼을 클릭한다.

이용 가능한 서브넷 (2/6)

Q 서브넷 검색 및 필터링

	이름	서브넷 ID	연결 대상	IPv4 CIDR
<input type="checkbox"/>	PRD-WAS-2b-92.75.20	subnet-098a731873c43792e	acl-0631fd4d1c5ce0a60	92.75.20.0/24
<input checked="" type="checkbox"/>	PUB-WEB-2b-92.75.200	subnet-04cb036c6534817fa	acl-0631fd4d1c5ce0a60	92.75.200.0/24
<input type="checkbox"/>	PRD-WAS-2a-92.75.10	subnet-0b985a2d487fe563c	acl-0631fd4d1c5ce0a60	92.75.10.0/24
<input checked="" type="checkbox"/>	PUB-WEB-2a-92.75.100	subnet-0765bc7c7bd1b3b09	acl-0631fd4d1c5ce0a60	92.75.100.0/24
<input type="checkbox"/>	PRD-LB-2a-92.75.1	subnet-0a3f690278a26e93f	acl-0631fd4d1c5ce0a60	92.75.1.0/24
<input type="checkbox"/>	PRD-LB-2b-92.75.2	subnet-0e556740d35b9b5b6	acl-0631fd4d1c5ce0a60	92.75.2.0/24

선택한 서브넷

subnet-0765bc7c7bd1b3b09 / PUB-WEB-2a-92.75.100 X subnet-04cb036c6534817fa / PUB-WEB-2b-92.75.200 X

1. 접근 제어 : 보안 그룹과 네트워크 ACL

■ 실습. NACL 생성 예제

- 2개의 퍼블릭 서브넷이 새로운 NACL에 연결됐다.
- 기본 NACL 연결은 해제돼 4개의 프라이빗 서브넷만 연결된 상태다.

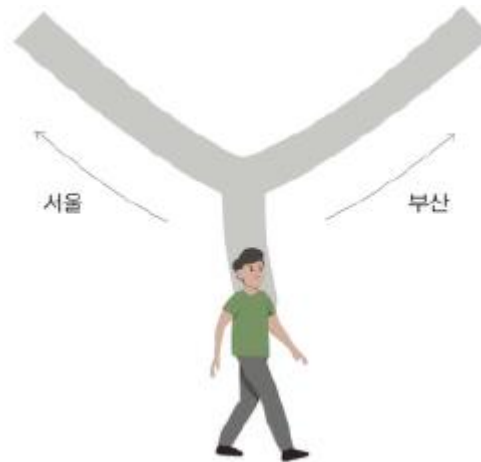
Name	네트워크 ACL ID	연결 대상	기본값
<input checked="" type="checkbox"/> Public-Subnet-NACL	acl-0f25698e011650066	2 서브넷	아니요
<input type="checkbox"/> -	acl-0631fd4d1c5ce0a60	4 서브넷	예

네트워크 ACL ID	연결 대상	기본값
acl-0f25698e011650066	2 서브넷	subnet-04cb036c6534817fa / PUB-WEB-2b-92.75.200 subnet-0765bc7c7bd1b3b09 / PUB-WEB-2a-92.75.100
소유자		
671559022704		

2. 경로 제어 : 라우팅 테이블

■ 라우팅이란?

- 부산으로 여행을 가던 중 갈림 길에 섰다.
- 어느 방향으로 가야 할지 몰라 두리번거리다 다행히 이정표를 발견했다.

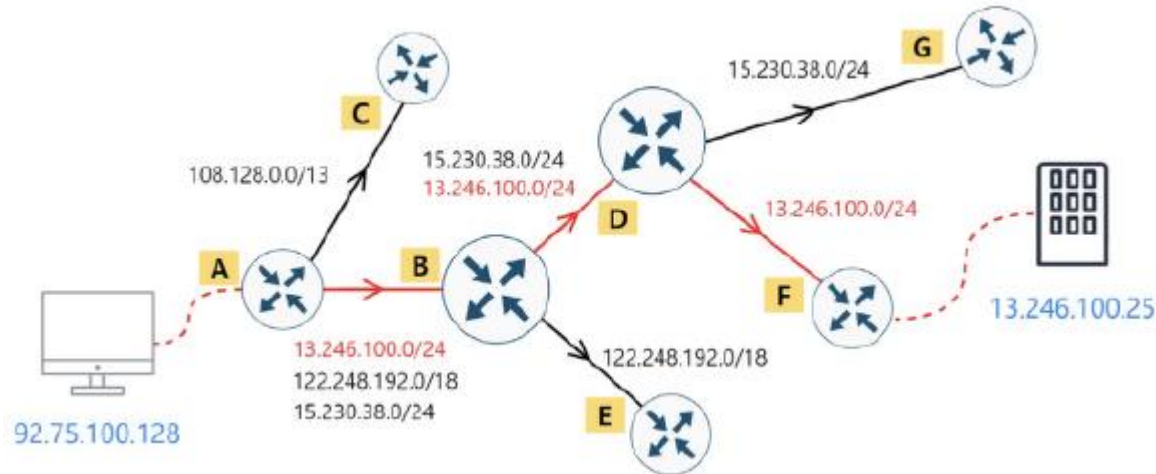


- 이정표에는 최종 목적지와 그 방향이 반드시 명기돼 있어야 한다.

2. 경로 제어 : 라우팅 테이블

■ 라우팅이란?

- 트래픽의 이정표는 라우팅이다.
- 트래픽은 라우팅에 명기된 목적지와 방향을 보고 다음 장소로 이동한다.
- 다음 그림은 PC(92.75.100.128)를 떠난 트래픽이 목적지(13.246.100.25)를 향해 가는 경로를 나타낸다.



- 그림에서 보는 것처럼 라우팅도 목적지와 방향이 있다.
- AWS에서는 목적지를 대상(Destination)이라 하고 방향을 타깃(Target) 또는 게이트웨이(Gateway)라 한다.
- Next Hop으로 쓸 때도 있다.
- 대상과 타깃으로 구성된 이 라우팅은 라우팅 테이블(Route table)에 쌓여 트래픽에게 경로를 안내한다.

2. 경로 제어 : 라우팅 테이블

■ 라우팅이란?

■ 라우팅 용어 주의

- AWS 콘솔을 '한국어'로 사용하면 라우팅의 대상(Destination)과 타깃(Target) 모두 '대상'으로 번역해 표시한다.
- AWS 한글 설명서에도 '대상'으로 번역돼 있으므로, 라우팅 테이블을 참고할 땐 영문 설명본과 비교 하길 바란다.

The image shows two screenshots of the AWS Route 53 console. The top screenshot is in English, showing a routing table with columns: Destination, Target, Status, and Propagated. The bottom screenshot is in Korean, showing the same routing table with columns: 대상 (Destination), 대상 (Target), 상태 (Status), and 전파됨 (Propagated). Both screenshots show a single route for destination 92.75.0.0/16, which is local, active, and not propagated. Red boxes highlight the 'Target' and '대상' columns in the Korean version, illustrating the translation of the term.

Destination	Target	Status	Propagated
92.75.0.0/16	local	Active	No

대상	대상	상태	전파됨
92.75.0.0/16	local	활성	아니요

2. 경로 제어 : 라우팅 테이블

■ 라우팅이란?

- 그림은 앞의 그림의 경로상에 놓인 네트워크 장치의 라우팅 테이블을 나타낸다.
- A, B, D는 장치 3개 각각에 연결된 인터페이스이며 트래픽이 다음 구간으로 이동할 때 여는 문(게이트웨이)이다.

A		B		D	
Destination	Target	Destination	Target	Destination	Target
13.246.100.0/24	B	15.230.38.0/24	D	13.246.100.0/24	F
122.248.192.0/18	B	13.246.100.0/24	D	15.230.38.0/24	G
15.230.38.0/24	B	122.248.192.0/18	E		
108.128.0.0/13	C				

- 각 장치의 구간 초입(인터페이스)에 도달한 트래픽은 라우팅 대상을 하나씩 훑어본다.
- 대상이 자신의 목적지와 일치하거나 목적지를 포함하면 타깃이 안내하는 방향으로 이동한다.
- 첫 번째 장치(A)의 라우팅은 다음과 같이 안내한다.
- 테이블에서 자신의 목적지를 찾지 못하면 트래픽은 소멸되고 여행은 종료된다.

대상(Destination)	타깃(Target)
목적지가 13.246.100.0/24 범위에 있으면	B로 이동하시오
목적지가 122.248.192.0/24 범위에 있으면	B로 이동하시오
목적지가 15.230.38.0/24 범위에 있으면	B로 이동하시오
목적지가 108.128.0.0/13 범위에 있으면	C로 이동하시오

2. 경로 제어 : 라우팅 테이블

■ 라우팅이란?

- 그림은 앞의 그림의 경로상에 놓인 네트워크 장치의 라우팅 테이블을 나타낸다.
- A, B, D는 장치 3개 각각에 연결된 인터페이스이며 트래픽이 다음 구간으로 이동할 때 여는 문(게이트웨이)이다.

A		B		D	
Destination	Target	Destination	Target	Destination	Target
13.246.100.0/24	B	15.230.38.0/24	D	13.246.100.0/24	F
122.248.192.0/18	B	13.246.100.0/24	D	15.230.38.0/24	G
15.230.38.0/24	B	122.248.192.0/18	E		
108.128.0.0/13	C				

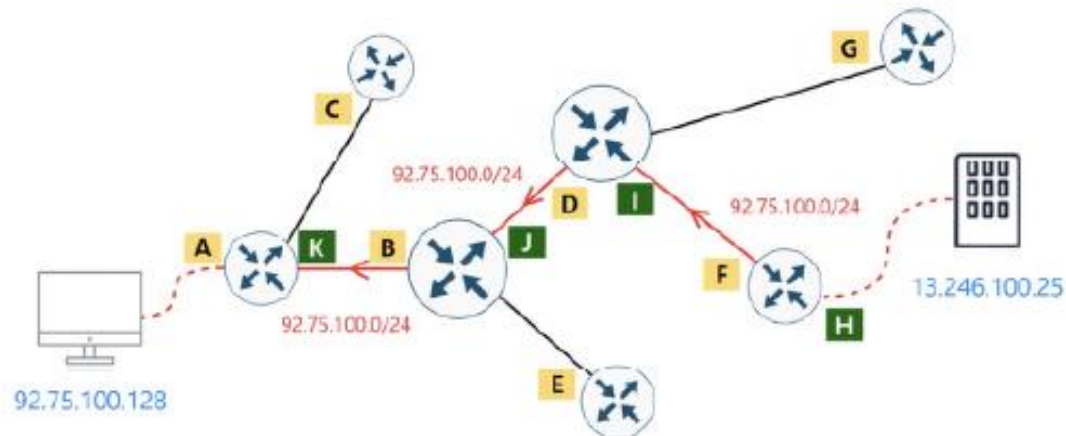
- 각 장치의 구간 초입(인터페이스)에 도달한 트래픽은 라우팅 대상을 하나씩 훑어본다.
- 대상이 자신의 목적지와 일치하거나 목적지를 포함하면 타깃이 안내하는 방향으로 이동한다.
- 첫 번째 장치(A)의 라우팅은 다음과 같이 안내한다.
- 테이블에서 자신의 목적지를 찾지 못하면 트래픽은 소멸되고 여행은 종료된다.

대상(Destination)	타깃(Target)
목적지가 13.246.100.0/24 범위에 있으면	B로 이동하시오
목적지가 122.248.192.0/24 범위에 있으면	B로 이동하시오
목적지가 15.230.38.0/24 범위에 있으면	B로 이동하시오
목적지가 108.128.0.0/13 범위에 있으면	C로 이동하시오

2. 경로 제어 : 라우팅 테이블

■ 반환 트래픽의 라우팅

- 위의 라우팅으로써 통신은 성공할까?
- PC가 보낸 트래픽이 목적지까지 도달하지만 통신은 실패한다.
- 이유는 무엇일까?
- 클라이언트가 서버로 TCP 통신을 요청한다고 가정하자.
- 이때 서버는 응답 패킷을 클라이언트로 반드시 회신해야 한다.
- 문제는 라우팅 테이블이 클라이언트로 돌아오는 경로를 안내하진 않는다는 점이다. 돌아오는 이 트래픽을 응답 트래픽(Response traffic) 또는 반환 트래픽(Return traffic)이라 한다.
- 그림에서 빨강선은 반환트래픽의 경로다.
- 초록표시(H, I, J, K)는 회신 방향에 연결된 라우팅 장치의 네트워크 인터페이스다.



2. 경로 제어 : 라우팅 테이블

■ 반환 트래픽의 라우팅

- 반환 트래픽의 시작점은 H다.
- [92.75.100.128]로 가는 트래픽을 I로 보내고 I와 J에서도 동일하게 안내한다.
- 이를 반영한 라우팅 테이블은 다음 그림과 같다.

A K	
Destination	Target
13.246.100.0/24	B
122.248.192.0/18	B
15.230.38.0/24	B
109.128.0.0/13	C

B J	
Destination	Target
15.230.38.0/24	D
13.246.100.0/24	D
122.248.192.0/18	E
92.75.100.0/24	K

D I	
Destination	Target
13.246.100.0/24	F
15.230.38.0/24	G
92.75.100.0/24	J

F H	
Destination	Target
92.75.100.0/24	I

2. 경로 제어 : 라우팅 테이블

■ 반환 트래픽의 라우팅

- 이 테이블에서 92.75.100.128(PC)이 13.246.100.25(서버) 접속에 필요한 라우팅만 모으면 다음 그림과 같이 정리할 수 있다.

A K	
Destination	Target
13.246.100.0/24	B

B J	
Destination	Target
13.246.100.0/24	D
92.75.100.0/24	K

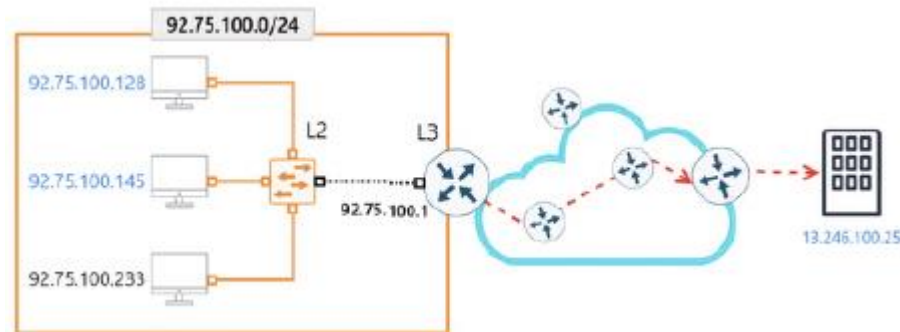
D I	
Destination	Target
13.246.100.0/24	F
92.75.100.0/24	J

F H	
Destination	Target
92.75.100.0/24	I

2. 경로 제어 : 라우팅 테이블

■ 서비스의 아지트: On-link(로컬) 라우팅

- PC가 네트워크(CIDR) 내부에서만 통신하려면 어떤 라우팅이 필요할까?
- 다음 그림의 [92.75.100.128]과 [92.75.100.145]는 같은 CIDR(92.75.100.0/24)의 멤버이므로 특별한 라우팅 없이도 서로 접속할 수 있다.



- PC에 IP를 할당하는 과정부터 살펴보자.
- 사용자는 CIDR 범위 내 미사용 IP를 찾아 다음 예시처럼 입력할 것이다.
- 물론 DHCP 방식으로 할당받을 수도 있다.
 - IP 주소: 92.75.100.128, 서브넷 마스크 : 255.255.255.0

2. 경로 제어 : 라우팅 테이블

■ 서비스의 아지트: On-link(로컬) 라우팅

- 이와 같이 IP를 할당하면 [92.75.100.128](PC)이 [92.75.100.0/24](CIDR)의 정식 멤버가 된 의미로, 다음 그림과 같은 라우팅을 PC에 자동 생성한다.

Destination	Target
92.75.100.0/24	On-link

- 이 라우팅을 PC에서 확인해보자.
- 명령 프롬프트에서 route print를 입력하면 다음 그림과 같은 모습을 볼 수 있다.

IPv4 경로 테이블					
=====					
활성 경로:					
네트워크 대상	네트워크 마스크	게이트웨이	인터페이스	메트릭	
92.75.100.0	255.255.255.0	연결됨	92.75.100.128	266	
92.75.100.128	255.255.255.255	연결됨	92.75.100.128	266	
92.75.100.255	255.255.255.255	연결됨	92.75.100.128	266	

IPv4 Route Table					
=====					
Active Routes:					
Network	Destination	Netmask	Gateway	Interface	Metric
	92.75.100.0	255.255.255.0	On-link	92.75.100.128	266
	92.75.100.128	255.255.255.255	On-link	92.75.100.128	266
	92.75.100.255	255.255.255.255	On-link	92.75.100.128	266

- 명시적으로 라우팅을 등록하지 않아도 CIDR 멤버끼리 접속이 가능한 이유는, 라우팅이 불필요해서가 아니라 자동 등록된 CIDR 라우팅이 있기 때문이다.
- 다시 말해 윈도우 PC에 IP 주소와 서브넷 마스크를 입력하는 행위 자체가 라우팅을 등록하는 과정이라 할 수 있다.

2. 경로 제어 : 라우팅 테이블

■ 서비스의 아지트: On-link(로컬) 라우팅

- 앞의 그림에서 빨강 표시된 라우팅을 직역해보자.

목적지가 **92.75.100.0/24** 범위에 있으면 On-link로 이동하시오.

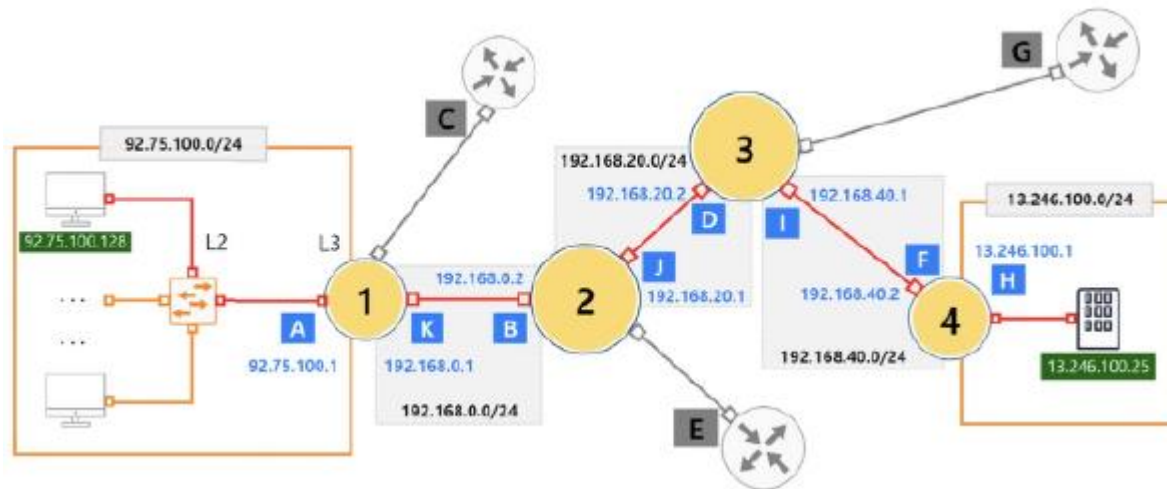
- 여기서 On-link(연결됨)란 PC에 연결된 인터페이스를 뜻한다.
- 즉, 빨강 박스 라우팅 오른쪽 끝에 보이는 [92.75.100.128] 인터페이스로 트래픽을 전달하라는 의미와 같다.
- 좀 더 쉽게 말하면 “내 구역(CIDR)은 내 선(인터페이스)에서 해결한다”는 뜻이다.
- 이로써 라우팅을 다음과 같이 바꿔 해석할 수 있다.

목적지가 **92.75.100.0/24** 범위에 있으면
내장된(연결된) **92.75.100.128** 인터페이스로 트래픽을 전달하시오.

2. 경로 제어 : 라우팅 테이블

■ 서비스의 아지트: On-link(로컬) 라우팅

- 이처럼 라우팅이 가능한 모든 장치는 On-link 라우팅에 저장돼 있다.
- 따라서 반환 트래픽 경로를 다음 그림으로 바꿔 표현할 수 있다.

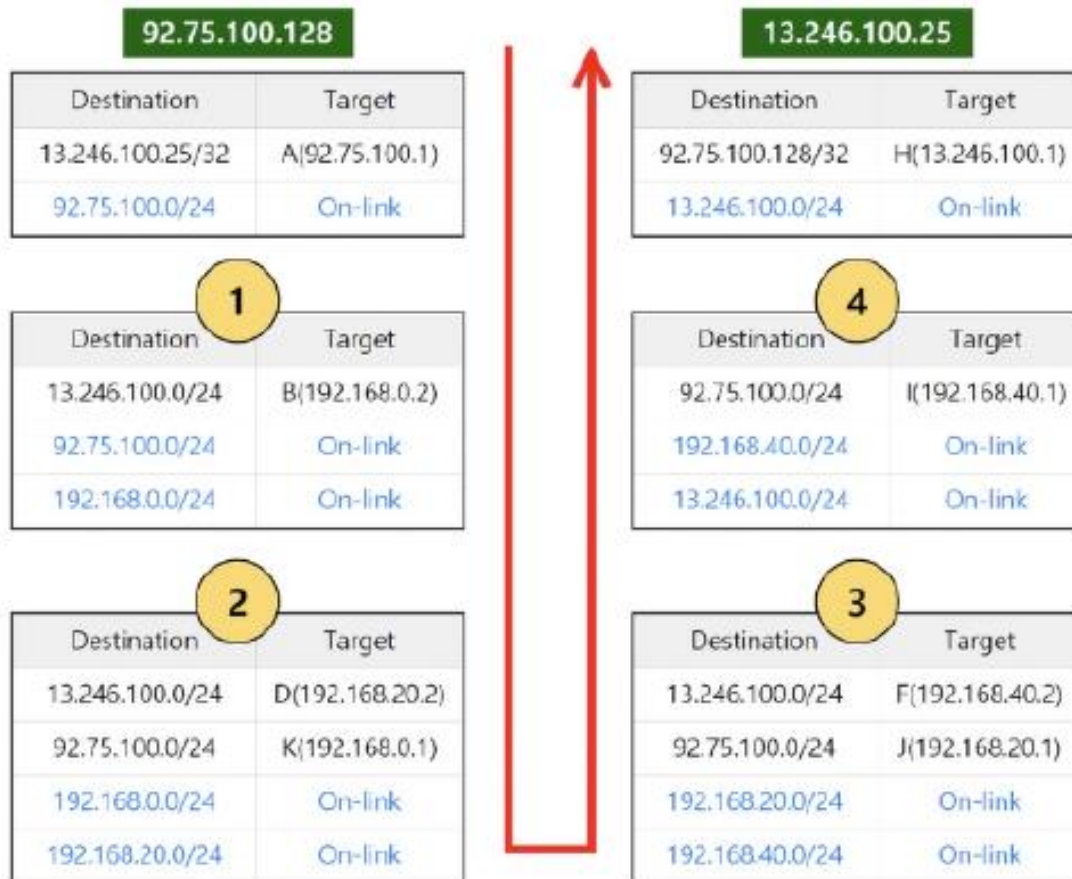


- 그림에서 트래픽 경로의 라우팅 장치(1~4)와 그 장치에 연결된 모든 인터페이스를 표시했다.
- 이 인터페이스들이 트래픽의 방향이자 게이트웨이이다.
- A, B, D, F는 가는 방향의 게이트웨이이고 H, I, J, K는 돌아오는 방향의 게이트웨이이다.
- ①~④번 각 장치에 연결된 인터페이스들도 IP가 할당돼 있다.
- 따라서 각 장치마다 Onlink 라우팅이 등록돼 있을 것이다.

2. 경로 제어 : 라우팅 테이블

■ 서비스의 아지트: On-link(로컬) 라우팅

- 다음 그림은 트래픽 경로상의 모든 라우팅 테이블이다.
- 테이블마다 요청 라우팅, 반환 라우팅, 그리고 On-link 라우팅까지 있으므로 완벽하다.
- 그럼 PC에서 트래픽을 다시 전송해보자.



2. 경로 제어 : 라우팅 테이블

■ VPC의 라우팅

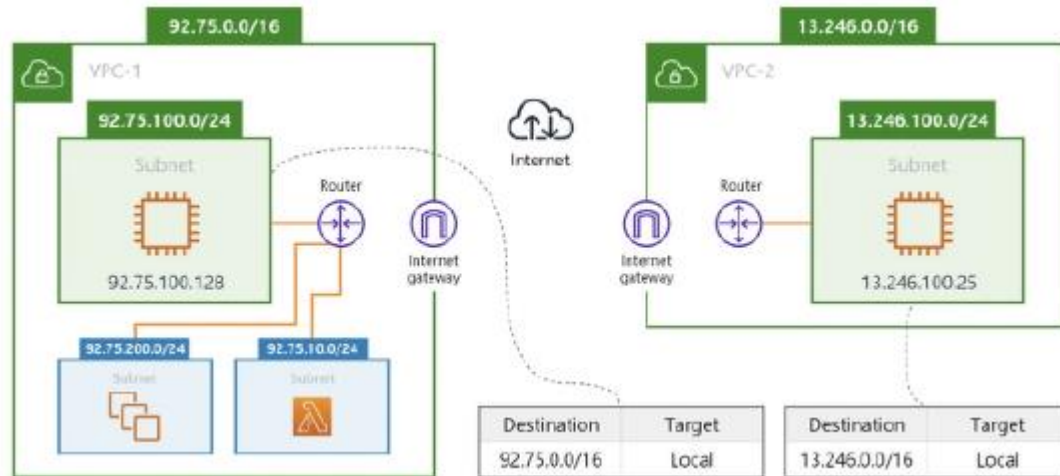
- VPC 라우팅은 온프레미스보다 쉽고 간편하며 다음과 같은 특징이 있다.
 - 라우팅(Routing) 기본 개념은 온프레미스와 같다.
 - VPC 라우팅은 라우팅 테이블(Route table)로 관리한다.
 - 라우팅 테이블은 서브넷 단위로 사용한다. 즉, 서브넷에 연결해 사용한다.
 - 네트워크 경로상에 존재하는 장치는 확인할 수 없을 뿐더러 그 라우팅은 신경쓰지 않아도 된다.
 - 인스턴스나 서비스 내부에 라우팅 설정을 하지 않는다. 서비스가 놓인 서브넷의 라우팅이 곧 서비스의 라우팅이다.
 - VPC 환경의 On-link는 Local로 표기하며 로컬 라우팅이라 한다. 로컬 라우팅 대상은 VPC CIDR이다.

특징 \ 라우팅 환경	VPC	온프레미스
라우팅 관리	라우팅 테이블	
라우팅 테이블 연결(관리) 대상	서브넷	경로상의 모든 라우팅 장치
중간 경로 라우팅	설정 불필요	장치별 설정 필요
On-link 라우팅	로컬(Local) 라우팅	On-link 라우팅
On-link 단위	VPC	라우팅 장치에 연결된 인터페이스의 CIDR

2. 경로 제어 : 라우팅 테이블

■ VPC의 라우팅

- 다음 그림은 온프레미스 환경을 VPC로 변환한 토폴로지다.

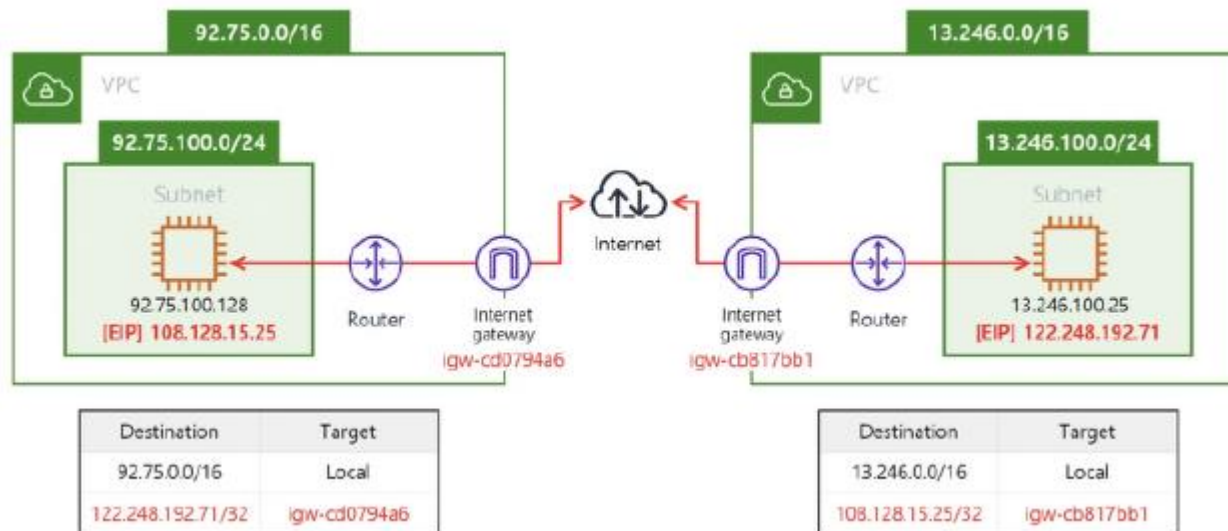


- VPC 토폴로지에서 Router와 라우팅 테이블을 볼 수 있다.
- Router는 가상 라우팅 장치며 라우팅 테이블 설정이 반영된다.
- [92.75.100.0/24]와 [13.246.100.0/24] 서브넷 모두 라우팅 테이블이 연결돼 있다.
- Local 라우팅 대상은 VPC의 CIDR 블록이다.
- 따라서 VPC-1의 3개 서브넷은 Local 라우팅만으로 서로 접속할 수 있다.

2. 경로 제어 : 라우팅 테이블

■ 인터넷 게이트웨이(IGW)와 NAT 테이블

- 인터넷 게이트웨이(Internet gateway, IGW)는 VPC 내부 서비스가 인터넷으로 접속하게 해주는 리소스다.
- 쉽게 말해 IGW를 통과한 트래픽은 인터넷으로 전송된다.
- IGW의 부모는 리전이고 연결 대상은 VPC이므로, 수명 주기 동안 리전 내부의 모든 VPC에 1 : 1로 연결하고 해제할 수 있다.
- VPC 서비스가 인터넷에 접속하려면 다음 요건을 만족해야 한다.
- 다음 그림을 보며 하나씩 확인한다.



2. 경로 제어 : 라우팅 테이블

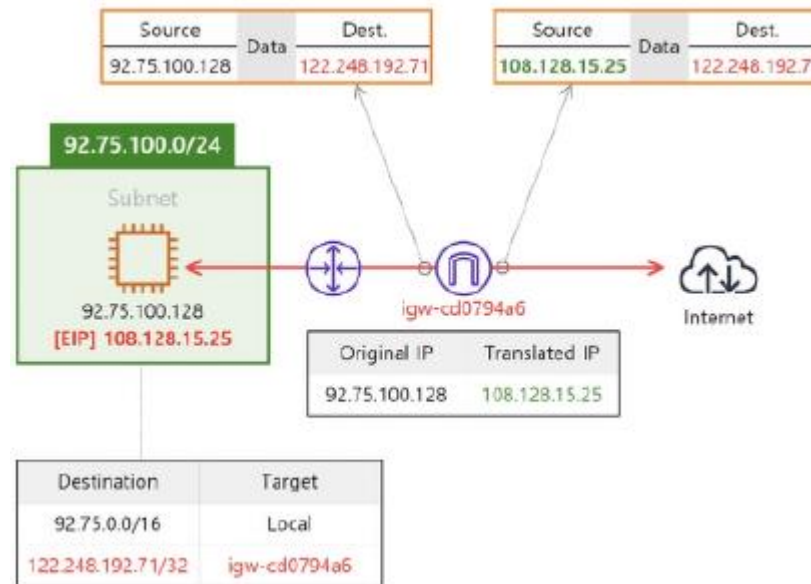
■ 인터넷 게이트웨이(IGW)와 NAT 테이블

- ① VPC에 IGW를 연결해야 한다.
 - 두 VPC에 [igw-cd0794a6], [igw-cb817bb1]을 각각 연결했다.
- ② 서비스가 놓인 서브넷 라우팅에 IGW를 타겟으로 설정해야 한다.
 - 각 서브넷 라우팅 테이블에 [igw-cd0794a6]과 [igw-cb817bb1]을 타겟으로 설정했다.
- ③ 서비스에 퍼블릭 IP 또는 탄력적 IP가 할당돼 있어야 한다.
 - 2개 인스턴스에 각각 탄력적 IP(108.128.15.25, 122.248.192.71)를 할당했다.

2. 경로 제어 : 라우팅 테이블

■ 인터넷 게이트웨이(IGW)와 NAT 테이블

- IGW를 이용한 [92.75.100.128](인스턴스)의 트래픽 전달 원리를 다음 그림에서 확인해보자.

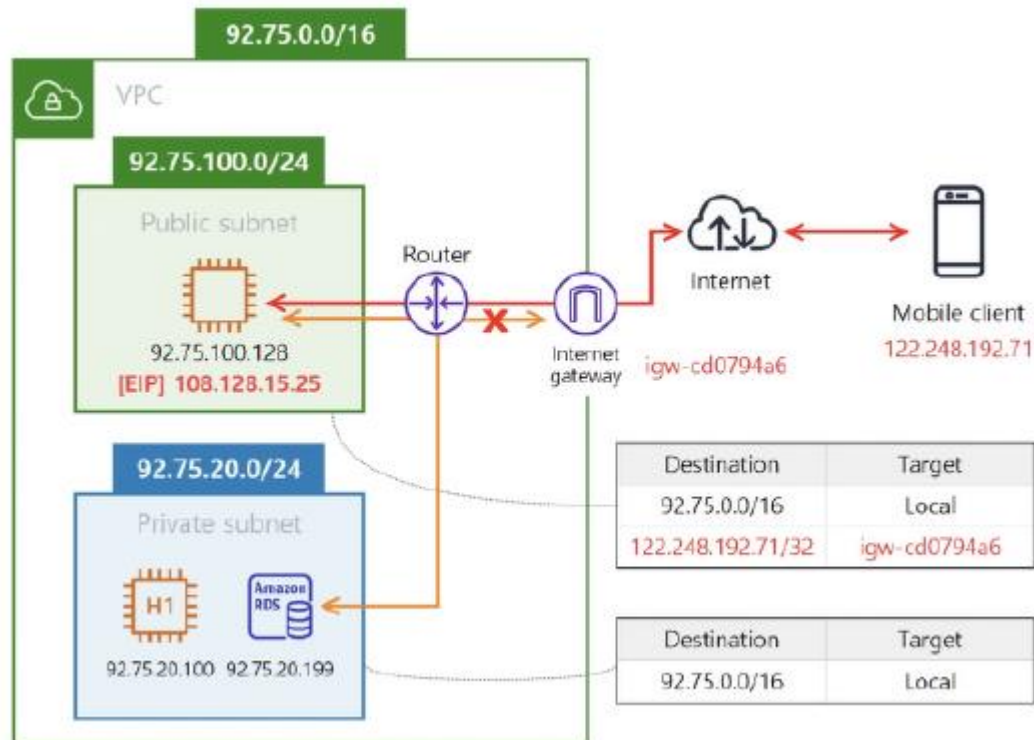


- 컴퓨팅 서비스가 퍼블릭 IP나 탄력적 IP와 연결되면 IGW는 NAT(Network Address Transition)테이블에 서비스의 주소 매핑 정보를 저장한다.
- 예컨대 그림의 인스턴스에 연결된 ENI의 프라이빗 IP(92.75.100.128)와 퍼블릭 IP(108.125.15.25)의 매핑 정보를 보관해 뒀다가, 서비스가 인터넷 접속을 시도하면 프라이빗 IP를 퍼블릭 IP로 변환해 인터넷으로 보낸다.
- 따라서 NAT 테이블에 프라이빗 IP와 매핑되는 퍼블릭 IP(또는 탄력적 IP)가 없으면 트래픽을 보낼 수 없다.

2. 경로 제어 : 라우팅 테이블

■ 퍼블릭과 프라이빗 서브넷의 경계: IGW

- IGW를 활용하면 인터넷으로 아웃바운드 요청을 보내거나 인터넷에서 들어오는 인바운드 요청을 직접 수신할 수 있다.
- 이렇게 양방향 요청이 가능한 서브넷을 퍼블릭 서브넷(Public subnet)이라 한다.
- 쉽게 말해 IGW가 연결된 서브넷은 퍼블릭 서브넷이다.
- 다음 그림은 퍼블릭 서브넷(92.75.100.0/24)과 프라이빗 서브넷(92.75.20.0/24)의 예시다.



2. 경로 제어 : 라우팅 테이블

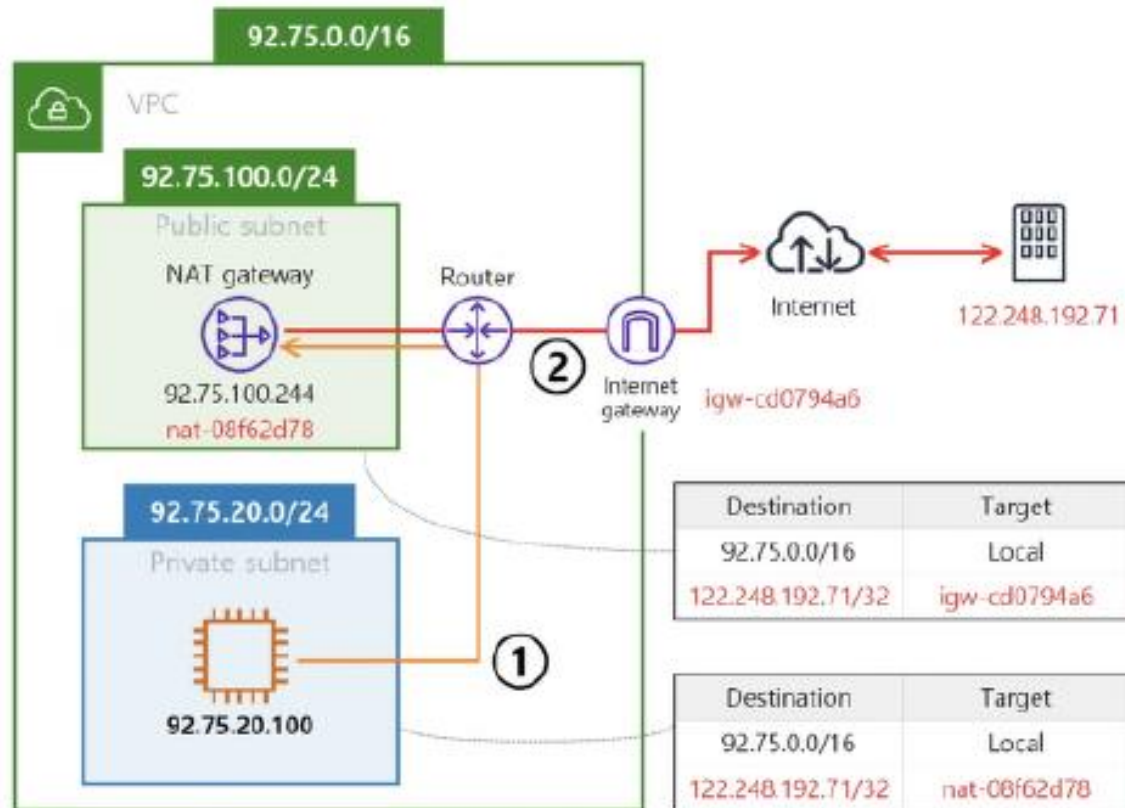
■ 퍼블릭과 프라이빗 서브넷의 경계: IGW

- 동영상 스트리밍 서비스를 예로 들어보자.
- 휴대폰(122.248.192.71)로 스트리밍 앱에 접속해 사용자 인증을 한 뒤, 보고 싶은 드라마를 선택하면 재생이 시작된다.
- 앱 UI 서버(108.128.15.25) 인터넷 사용자의 요청(인바운드 트래픽)을 처리해야 하므로 퍼블릭 서브넷에 구축돼 있을 것이다.
- 그러나 영상 콘텐츠 서버(92.75.20.100)나 영상 목록, 사용자 정보가 담긴 데이터베이스는 높은 보안 수준으로 관리해야 하고, 미인증 사용자가 직접 접근할 수 없어야 하므로 서브넷에 IGW를 연결하면 안 된다.
- 이런 용도의 서브넷을 프라이빗 서브넷(Private subnet)이라 한다.
- 사용자는 프라이빗 서브넷에 직접 접근할 수 없지만, 퍼블릭 서브넷은 프라이빗 서브넷과 서로 통신할 수 있다.
- 따라서 사용자 인증 절차는 이 둘 사이에 수행하며 퍼블릭 서브넷은 인증 결과를 인터넷 사용자에게 회신한다.

2. 경로 제어 : 라우팅 테이블

■ NAT 게이트웨이

- 그럼 프라이빗 인스턴스가 인터넷에 접속하려면 어떻게 해야 할까?
- 다음 그림은 프라이빗 인스턴스가 NAT 게이트웨이를 경유해 [122.248.192.71](인터넷 서버)에 접속하는 토폴로지다.



2. 경로 제어 : 라우팅 테이블

■ NAT 게이트웨이

■ NAT 게이트웨이의 특징은 다음과 같다.

- NAT 게이트웨이는 소스 IP 변환이 주목적이다. 퍼블릭 유형과 프라이빗 유형이 있다.
- 퍼블릭 유형은 프라이빗 IP만 소유한 서비스가 인터넷 접속이 필요할 때 사용하고, 프라이빗 유형은 인터넷 접속과 관계없이 소스 주소 변환의 목적으로만 사용한다.
- NAT 게이트웨이의 부모는 서브넷이다.
- NAT 게이트웨이는 라우팅 ENI를 사용하는 서비스다.
- 라우팅 ENI는 SG를 사용하지 않으며, 소스/대상 확인 옵션이 꺼져 있다. 즉, 트래픽을 단순 포워딩한다.
- NAT 게이트웨이의 ENI는 요청자 관리형이다. 그러므로 ENI 화면에서 옵션 변경이 불가능하다.
- 퍼블릭 유형의 NAT 게이트웨이를 생성하려면 인터넷 접속에 필요한 탄력적 IP를 연결해야 한다. 연결할 시점에 리전에 할당된 탄력적 IP가 없으면 NAT 게이트웨이 생성 단계에서 탄력적 IP 할당과 동시에 연결할 수도 있다.
- NAT 게이트웨이를 라우팅 타깃으로 설정하면 라우팅 대상의 모든 트래픽은 NAT 게이트웨이로 전달된다.
- 퍼블릭 유형의 NAT 게이트웨이로 진입한 트래픽은 탄력적 IP로써 인터넷에 전송된다.

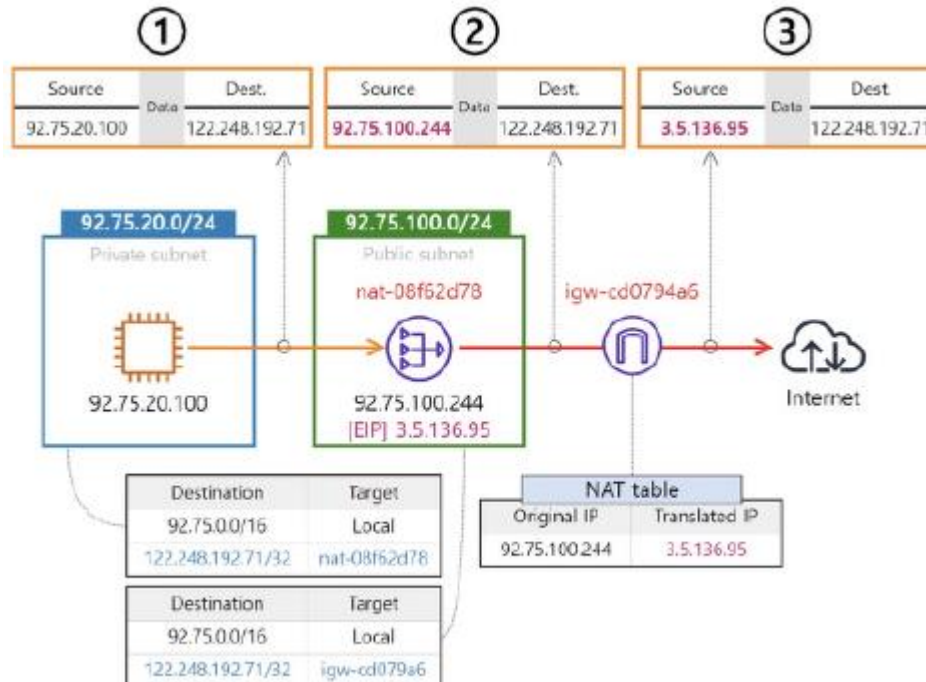
2. 경로 제어 : 라우팅 테이블

■ NAT 게이트웨이

- [92.75.20.100](프라이빗 인스턴스)의 인터넷 접속 순서는 다음과 같다.

- ① [92.75.20.0/24](프라이빗 서브넷) 라우팅 테이블은 [nat-08f62d78](NAT 게이트웨이)을 타겟으로 설정했다. [122.248.192.71]에 접속하는 트래픽은 [nat-08f62d78]을 전달된다.
- ② [92.75.100.0/24](퍼블릭 서브넷) 라우팅 테이블은 [igw-cd0794a6]을 타겟으로 설정했다. [122.248.192.71]에 접속하는 트래픽은 [igw-cd0794a6]으로 전달돼 인터넷으로 나간다.

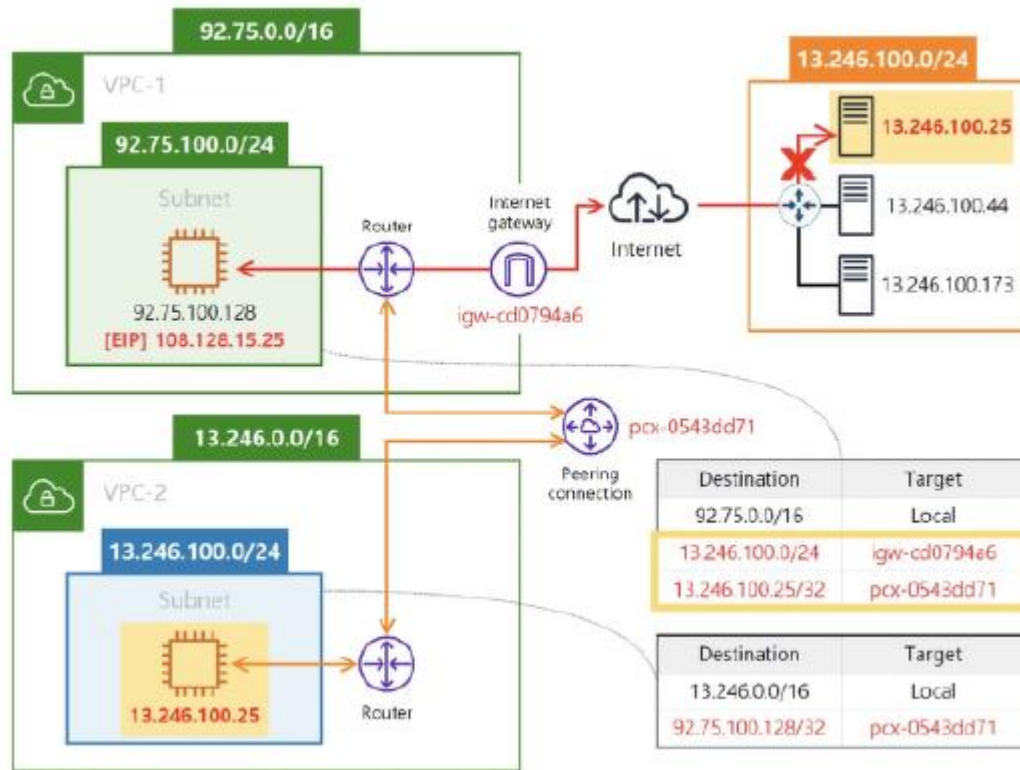
- ①~② 과정에서 트래픽 IP가 변환되는 모습은 다음 그림과 같다.



2. 경로 제어 : 라우팅 테이블

■ 라우팅의 솔로몬: Longest Prefix Match

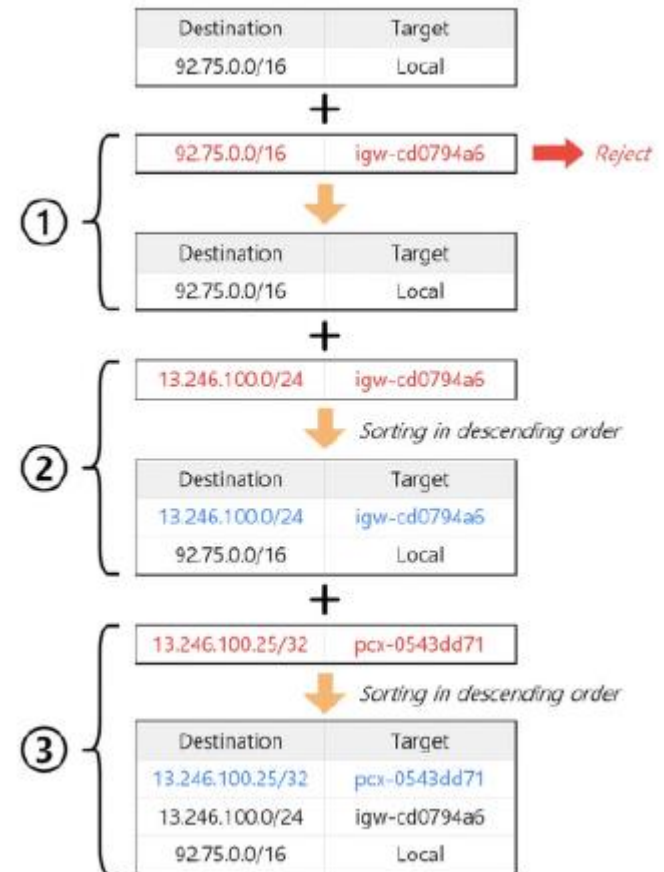
- 인스턴스의 접속 대상 IP가 인터넷과 VPC에 공존하는 상황을 가정해보자.
- 다음 그림은 인스턴스(92.75.100.128)가 접속할 대상(13.246.100.25)이 인터넷과 2번 VPC에 모두 존재하는 상황을 나타낸다.



2. 경로 제어 : 라우팅 테이블

■ 라우팅의 솔로몬: Longest Prefix Match

- 신규 라우팅 등록 요청이 들어오면 라우팅 테이블은 라우팅 저장 순서에 따른 오동작을 방지하고자 내부적으로 다음 그림과 같은 프로세스를 수행한다.
- Prefix란 라우팅 대상을 뜻한다.
- Longest Prefix Match는 가장 긴 서브넷 마스크 비트 순으로 라우팅을 정렬, 비교해 트래픽을 안내하는 것이다.



2. 경로 제어 : 라우팅 테이블

■ VPC 라우팅 구체화: East-West 트래픽 검사

- 그럼 Longest Prefix Match 기법은 로컬 라우팅에서도 통할까?
- 예컨대 [92.75.0.0/16]이 포함하는 [92.75.100.145/32]나 [92.75.100.0/24]를 라우팅 대상으로 적용하는 것이다.
- 다음 그림의 왼쪽 테이블은 VPC CIDR(92.75.0.0/16)보다 더 구체화된 라우팅 2개를 적용했고 오른쪽 테이블에는 VPC CIDR을 포함하는 더 넓은 범위의 라우팅 2개를 적용했다.

32, 24 \geq 16
(Equal to or more specific)

Destination	Target
92.75.100.145/32	eni-00cfb6fa
92.75.100.0/24	eni-0e8a7f1d
92.75.0.0/16	Local

13, 8 $<$ 16

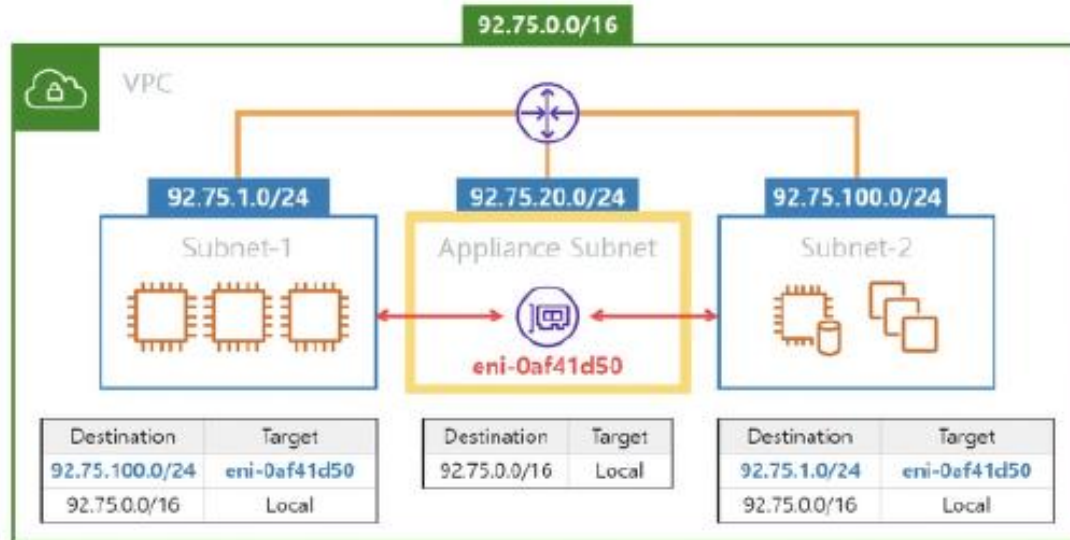
Destination	Target
92.75.0.0/16	Local
92.72.0.0/13	pcx-0543dd71
92.0.0.0/8	igw-cd0794a6

- 2021년 9월 이전에는 왼쪽 테이블처럼 라우팅 등록을 시도하면 오류가 발생했다.
- 다시 말해 VPC CIDR이 포함하는 영역은 라우팅 대상으로 허용하지 않았다.
- 반면 오른쪽 테이블은 [92.75.0.0/16]을 포함하는 더 넓은 영역(92.72.0.0/13, 92.0.0.0/8)이므로 변함없이 대상으로 적용할 수 있다.
- 트래픽의 목적지가 [92.5.12.108]이라면 Longest Prefix Match 기법에 따라 [igw-cd0794a6]으로 전송될 것이다.

2. 경로 제어 : 라우팅 테이블

■ VPC 라우팅 구체화: East-West 트래픽 검사

- East-West 트래픽이란 수평으로 흐르는 트래픽이다.
- 예컨대 VPC 내부 인스턴스간 통신이나 Transit Gateway, Direct Connect 등을 이용한 하이브리드 네트워킹도 East-West 트래픽으로 볼 수 있다.
- 여기서 주목할 만한 점은 VPC CIDR보다 구체화된 라우팅 적용을 할 수 있게 되면서, East-West 트래픽 검사가 가능하게 됐다는 점이다.
- 다음 그림은 VPC 내부 East-West 트래픽 라우팅 예시다.
- [Subnet-1] 인스턴스가 [Subnet-2] 서비스와 통신할 때 반드시 [eni-0af41d50]를 경유하도록 설계된 아키텍처다.

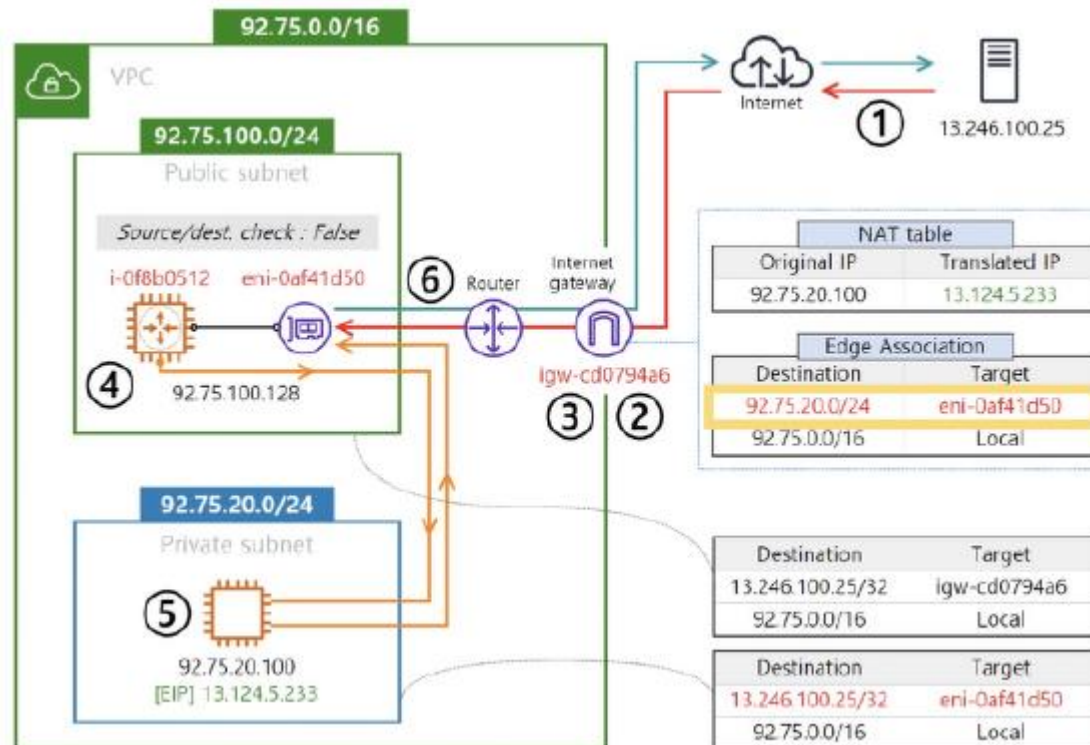


- 예시처럼 Appliance Subnet에 보안 어플라이언스나 모니터링 솔루션을 설치하면 트래픽이 최종목적지에 도달 전미리 검사를받고 반드시 필요한 패킷만중요 인스턴스로 전달하게 된다.

2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

- East-West와 반대되는 개념으로 North-South 트래픽이 있다.
- 네트워크 아키텍처상 수직으로 흐르는 트래픽을 생각하면 된다.
- IGW를 통과하는 트래픽이 그 예다.
- 다음 그림은 VPC CIDR보다 작은 규모의 CIDR(을 라우팅 대상에 등록한 예시다.
- 이처럼 North-South 트래픽도 VPC CIDR보다 구체화된 라우팅 적용을 할 수 있다.



2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

- ① 트래픽 전송: 인터넷서버 가 인스턴스로 트래픽을 전송한다.
- ② NAT 변환: IGW의 NAT 테이블은 트래픽의 목적지 IP를 변환 한다.
- ③ 엣지 라우팅 : IGW에 연결된 게이트웨이 라우팅 테이블을 확인한다. 목적지 |) 를 포함하는 라우팅 대상 을 찾고, 으로 트래픽을 전달한다. ENI를 소유한 B而 (인스턴스) 가수신한다.
- ④ 소스/대상 확인이 해제된 인스턴스의 트래픽 포워딩 : [인스턴스) 는소스/대상확인이 해제돼 있으므로 수신한 트래픽의 목적지 가자신 |) 과달라도트래 픽을수용한다 .
- 라우팅기능이설치된인스턴스는 서브넷의 로컬 라우팅에 따라 인스턴 스로 트래픽을 포워딩한다.

2. 경로 제어 : 라우팅 테이블

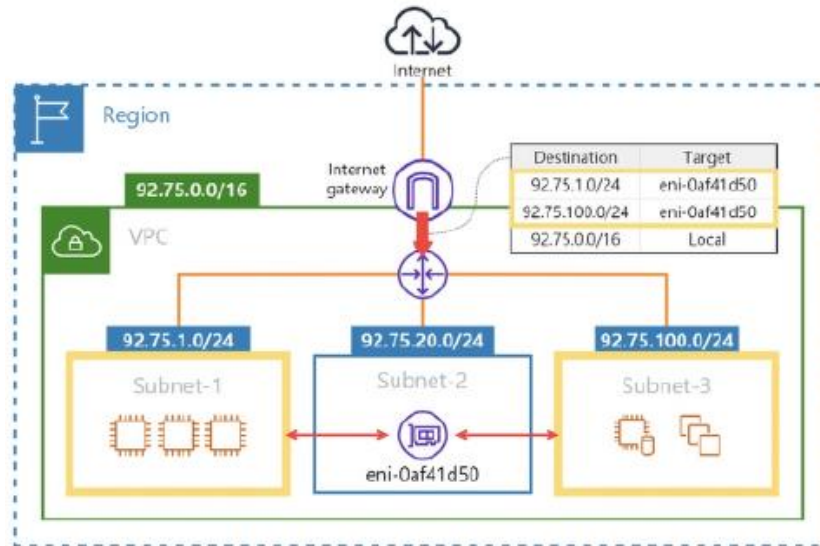
■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

- ① 트래픽 전송: 인터넷 서버(13.246.100.25)가 [13.124.5.233] 인스턴스로 트래픽을 전송한다.
- ② NAT 변환: IGW의 NAT 테이블은 트래픽의 목적지 IP를 변환(13.124.5.233 → 92.75.20.100)한다.
- ③ 엣지 라우팅 : IGW에 연결된 게이트웨이 라우팅 테이블을 확인한다. 목적지(92.75.20.100)를 포함하는 라우팅 대상(92.75.20.0/24)을 찾고, [eni-0af41d50]으로 트래픽을 전달한다. ENI를 소유한 i-0f8b0512(인스턴스)가 수신한다.
- ④ 소스/대상 확인이 해제된 인스턴스의 트래픽 포워딩 : i-0f8b0512(인스턴스)는 소스/대상 확인이 해제돼 있으므로 수신한 트래픽의 목적지(92.75.20.100)가 자신((92.75.100.128)과 달라도 트래픽을 수용한다. 라우팅 기능이 설치된 인스턴스는 [92.75.100.0/24] 서브넷의 로컬 라우팅에 따라 [92.75.20.100] 인스턴스로 트래픽을 포워딩한다.
- ⑤ 반환 트래픽 회신: 트래픽을 수신한 [92.75.20.100] 인스턴스는 [92.75.20.0/24] 서브넷의 첫 번째 라우팅에 따라 반환 트래픽을 [eni-0af41d50]으로 전달한다.
- ⑥ 트래픽 포워딩과 NAT 변환: 반환 트래픽을 수신한 인스턴스는 소스/대상 확인이 해제돼 있으므로 반환 트래픽의 목적지(13.246.100.25)가 자신(92.75.100.128)과 달라도 트래픽을 수용한다. 곧이어 서브넷의 첫 번째 라우팅에 따라 [igw-cd0794a6]으로 전달된다. 반환트래픽의 출발지 IP(92.75.20.100)는 NAT 테이블에서 소스 변환(13.124.5.233)돼 전송된다.

2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

- 게이트웨이 라우팅 테이블을 경유하는 인바운드 트래픽

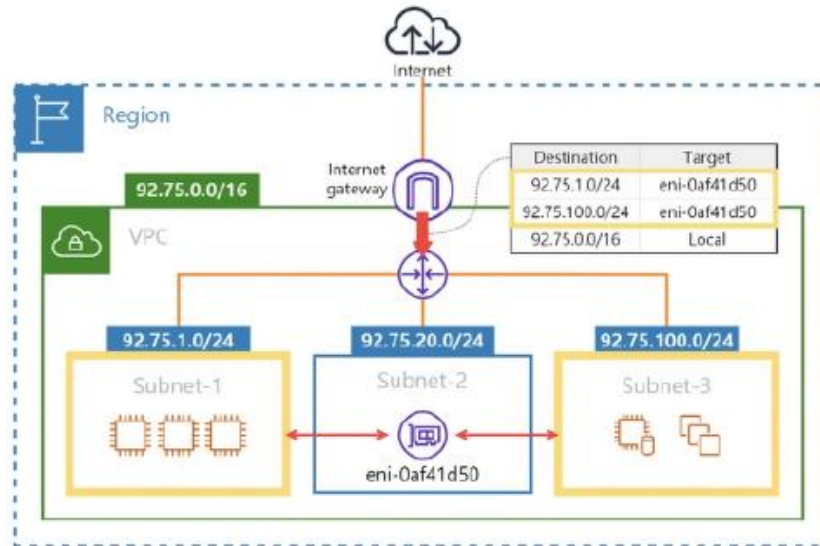


- 엣지(Edge)란 IGW와 가상 프라이빗 게이트웨이(Virtual Private Gateway, VGW)를 통틀어 지칭하는 용어다.
- 이 2가지 엣지의 부모는 리전이며 연결 대상은 VPC다. 따라서 엣지는 VPC에 연결돼 VPC 내부와 외부 사이 트래픽을 중계한다.
- 라우팅 테이블을 엣지(IGW, VGW)에 연결한 것을 엣지 연결(Edge association)이라 한다. 엣지연결은 선택이다.
- 라우팅 테이블 1개만 엣지에 연결할 수 있다. 서브넷에 연결된 라우팅 테이블도 엣지 연결을 할 수 있다.
- 엣지 연결 상태의 라우팅 테이블을 게이트웨이 라우팅 테이블(Gateway route table)이라 한다. 2가지 엣지(IGW, VGW) 모두 게이트웨이이기 때문일 것이다.

2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

- 게이트웨이 라우팅 테이블을 경유하는 인바운드 트래픽



- 게이트웨이 라우팅 테이블은 VPC 인바운드 트래픽만 관여한다. 그래서 이 테이블의 라우팅을 내부 라우팅(Ingress Routing)이라 부르기도 한다.
- 따라서 라우팅 대상도 VPC CIDR로 한정해야 하고, 라우팅 대상의 합도 VPC CIDR이어야 한다. 그래야 엣지로 유입되는 트래픽을 서브넷 곳곳에 빈틈없이 전달할 수 있다.
- 이로써 라우팅 타겟도 VPC 내부 리소스라 추측할 수 있다. 대개 서브넷 라우팅은 그 대상을 VPC CIDR로 한정하지 않으므로 VPC CIDR로 한정된 엣지용 라우팅 테이블은 서브넷 라우팅 테이블과 구분해서 사용해야 한다

2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

■ 게이트웨이 라우팅 테이블의 조건

- 라우팅 대상이 VPC CIDR인 라우팅은 삭제할 수 없으며 VPC CIDR 외 다른 라우팅을 새로 추가하려면 VPC 내부의 서브넷 단위로만 지정할 수 있다.
- 즉, 호스트 단위로는 적용할 수 없다.
- 아래 그림의 게이트웨이 라우팅 테이블을 보자.
- VPC CIDR(92.75.0.0/16)을 대상으로 지정한 기본 로컬 라우팅이 있다.
- 추가된 라우팅 2개는 VPC에 속한 서브넷(92.75.1.0/24, 92.75.100.0/24)을 대상으로 지정했다.
- 서브넷용 라우팅 테이블은 [92.75.0.0/16](VPC CIDR)을 포함하는 더 넓은 범위나 겹치지 않는 CIDR도 대상으로 지정할 수 있지만 엣지 연결의 라우팅 대상은 VPC 내부 서브넷으로 한정된다는 점에 유의하자.

X		O	
Destination	Target	Destination	Target
92.75.100.0/24	Local	92.75.0.0/16	Local
Destination	Target	Destination	Target
92.75.1.0/24	eni-0af41d50	92.75.0.0/16	eni-0af41d50
Destination	Target	Destination	Target
92.75.1.233/32	eni-0af41d50	92.75.100.0/24	eni-0af41d50
92.75.0.0/16	Local	92.75.0.0/16	Local
Destination	Target	Destination	Target
92.75.1.233/32	eni-0af41d50	92.75.1.0/24	eni-0af41d50
92.75.100.0/24	eni-0af41d50	92.75.100.0/24	eni-0af41d50
92.75.0.0/16	Local	92.75.0.0/16	Local

2. 경로 제어 : 라우팅 테이블

■ 엣지 연결과 Ingress Routing: North-South 트래픽 검사

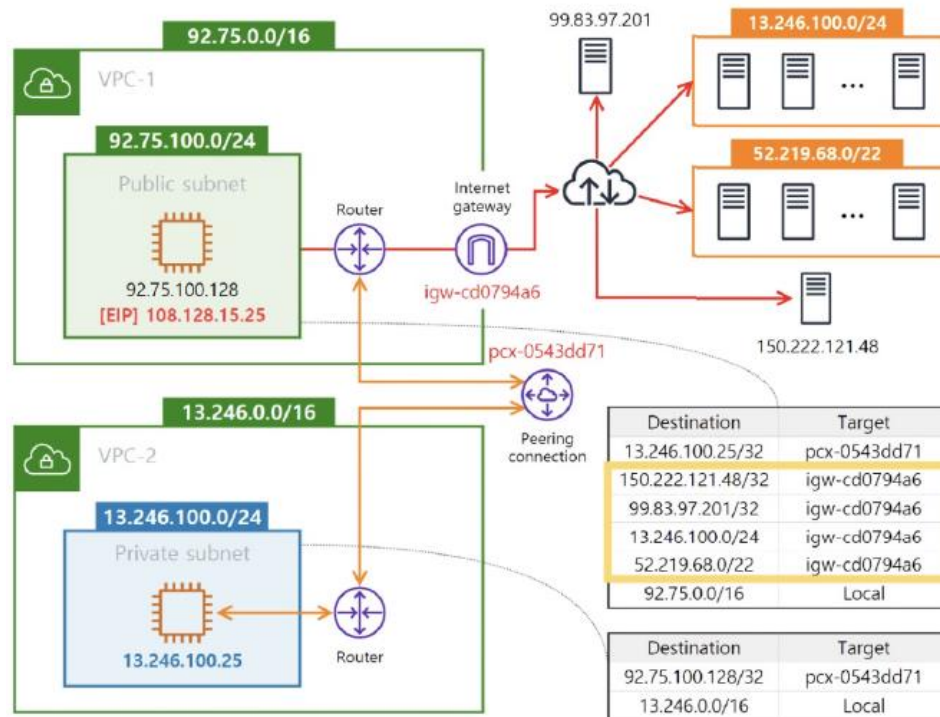
■ 게이트웨이 라우팅 테이블의 조건

- 라우팅 타킷은 다음 3종류만 가능하다.
 - ENI 또는 인스턴스
 - 게이트웨이 로드밸런서 엔드포인트
 - 로컬 (Local)

2. 경로 제어 : 라우팅 테이블

■ 기본 게이트웨이와 그 위험성

- 다음 그림처럼 VPC 인스턴스가 다수 인터넷 서버와 연동하는 상황을 가정해보자.



- [92.75.100.128] 인스턴스의 인터넷 접속 대상은 2개 대역(13.246.100.0/24, 52.219.68.0/22)과 2개 서버(99.83.97.201, 155.222.121.48)이다.
- 이에 필요한 모든 라우팅도 적용했다.

2. 경로 제어 : 라우팅 테이블

■ 기본 게이트웨이와 그 위험성

- 접속 대상이 이같이 한정돼 있다면 몰라도 IP 주소가 변경되거나 더 늘어나면 라우팅을 매번 변경하거나 신규로 등록해야 하므로 번거롭다.
- 라우팅을 수정해서 이 문제를 해결해보자.
- 다음 그림의 왼쪽 노랑 박스처럼 타깃(igw-cd0794a6)은 같지만 대상이 일정하지 않고 불규칙한 CIDR을 사용한다면 이를 모든 IP(0.0.0.0/0)로 바꿔 사용할 수 있다.
- 즉, 오른쪽 테이블처럼 (0.0.0.0/0)을 대상으로 지정한 라우팅의 타깃(igw-cd0794a6)을 기본 게이트웨이(default gateway)라 하며, 이 라우팅(0.0.0.0/0, igw-cd0794a6)을 기본 라우팅(default route)이라 한다.



Destination	Target
13.246.100.25/32	pcx-0543dd71
150.222.121.48/32	igw-cd0794a6
99.83.97.201/32	igw-cd0794a6
13.246.100.0/24	igw-cd0794a6
52.219.68.0/22	igw-cd0794a6
92.75.0.0/16	Local

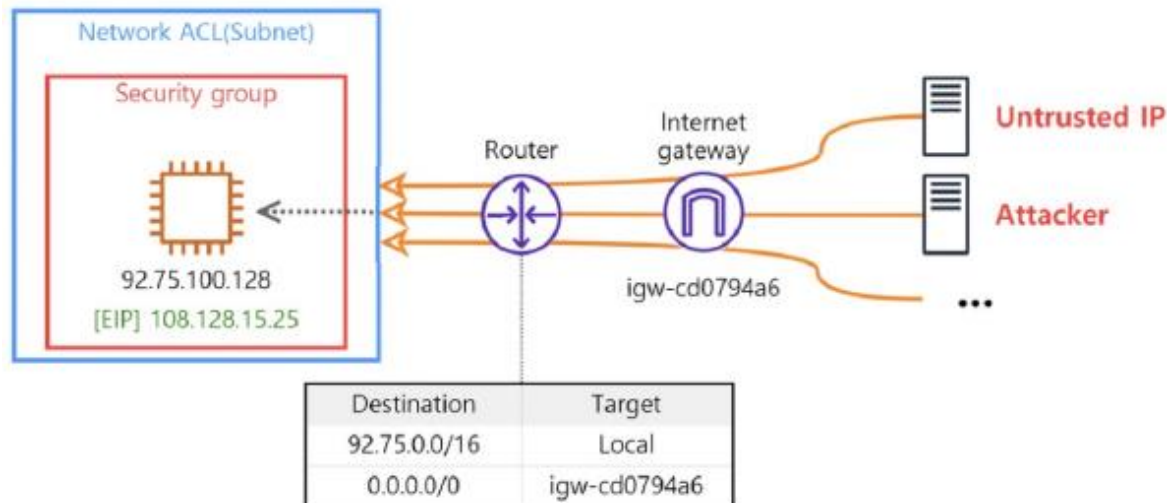
Destination	Target
13.246.100.25/32	pcx-0543dd71
92.75.0.0/16	Local
0.0.0.0/0	igw-cd0794a6

- 기본 라우팅은 가장 짧은 비트 마스크(/0)를 사용하므로 Longest Prefix Match 기법에 따라 그 어떤 라우팅보다 후순위로 탐색된다.
- 다시 말해 트래픽이 기본 게이트웨이를 타깃으로 채택했다면 앞선 모든 라우팅의 대상 매칭에 실패했다는 뜻이 된다.

2. 경로 제어 : 라우팅 테이블

■ 기본 게이트웨이와 그 위험성

- 이처럼 기본 라우팅을 사용하면 라우팅 개수를 현저히 줄일 수 있으므로 라우팅 탐색 시간을 절약해 VPC 네트워크 성능을 향상시킬 수 있다.
- 위의 예시처럼 피어링 연결(pcx-0543dd71)같은 예외 통신만 라우팅 테이블에 등록하고, 접속량이 많거나 주 경로로 사용하는 타깃을 기본 게이트웨이로 활용하면 효율적이다.
- 반대로 기본 라우팅은 다음 그림처럼 모든 IP와 통신할 수 있는 길을 열어 놓은 것이다.
- 이는 악성 트래픽이 서브넷이나 서비스 문 앞에서 자신을 허용(SG, NACL)해 주길 기다리고 있는 것과 같다.
- 따라서 접속 대상 CIDR이 명확하거나 마스크 비트를 한정할 수 있다면 기본 라우팅을 사용하면 안된다.



2. 경로 제어 : 라우팅 테이블

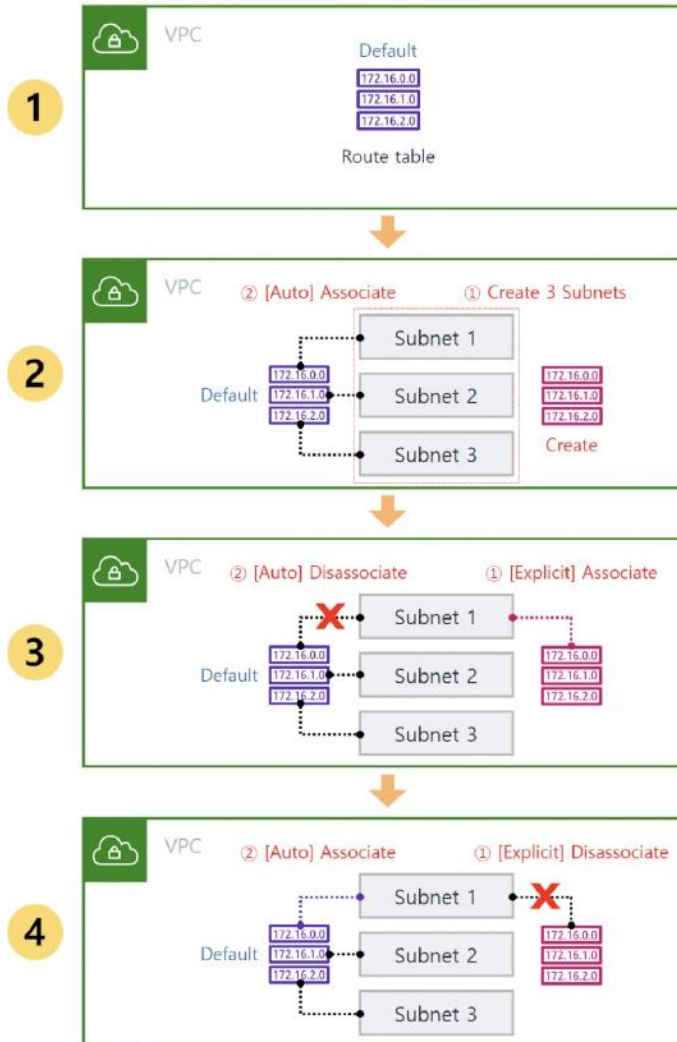
■ 라우팅 테이블의 표면적 특징과 다중 연결성 (1 : N)

- 서브넷 생성 시점에 기본 NACL이 연결되는 것처럼, 기본 라우팅 테이블도 서브넷에 자동 연결된다.
- 서브넷 연결 측면에서 라우팅 테이블과 NACL의 특징은 같다.
- 라우팅 테이블의 특징
 - 라우팅 테이블(Route table)의 부모는 VPC이다.
 - 라우팅 테이블의 연결 대상은 서브넷이며 수명 주기 동안 다른 서브넷에 연결할 수 있다. 또 어떤 서브넷에도 연결하지 않은 상태로 존재할 수 있다.
 - 반대로 서브넷은 수명 주기 동안 라우팅 테이블과 반드시 연결돼야 한다. 서브넷은 단 하나의 라우팅 테이블을 사용하지만 다른 라우팅 테이블로 바꿔 사용할 수도 있다.
 - 기본 VPC를 포함한 모든 VPC가 생성될 때 기본 라우팅 테이블도 함께 생성된다. 따라서 기본 라우팅 테이블과 VPC 개수는 같다.
 - VPC 내부에 서브넷을 생성하는 시점에, 연결 대상 라우팅 테이블을 바로 지정할 수 없다. 서브넷을 생성하면 일단 기본 라우팅 테이블에 자동 연결된다.

2. 경로 제어 : 라우팅 테이블

■ 라우팅 테이블의 표면적 특징과 다중 연결성 (1 : N)

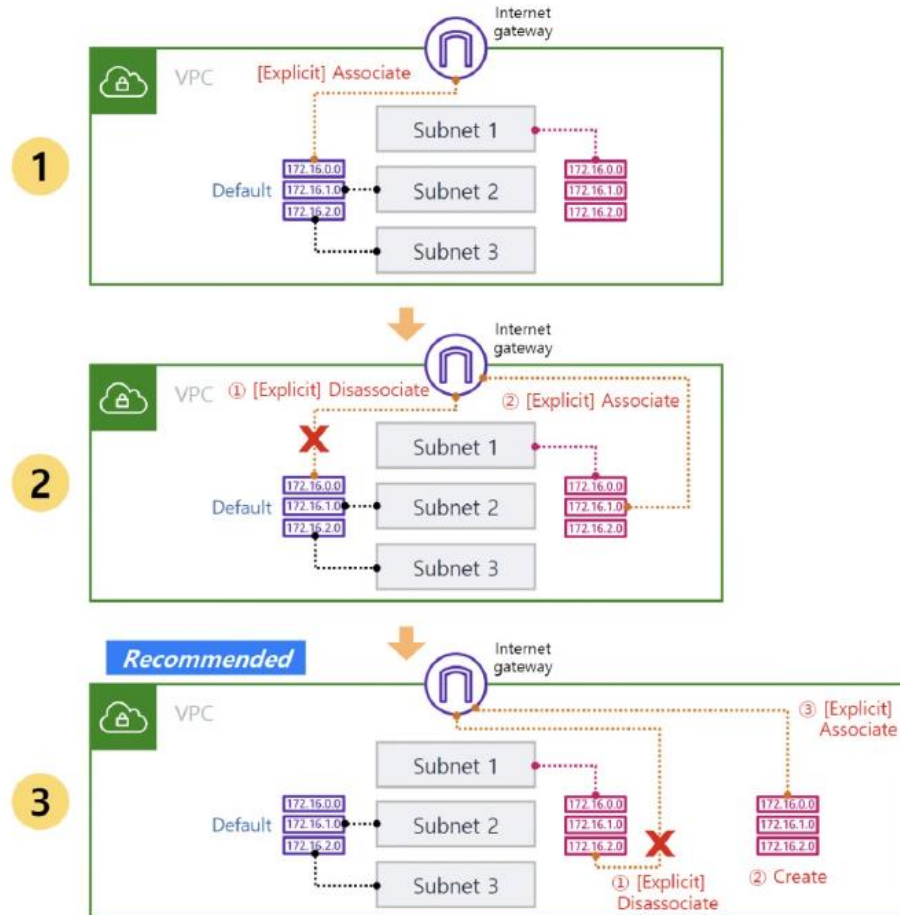
■ 라우팅 테이블의 특징



2. 경로 제어 : 라우팅 테이블

■ 라우팅 테이블의 표면적 특징과 다중 연결성 (1 : N)

■ 엣지 연결의 특징



2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- 서비스>VPC> 라우팅 테이블 메뉴에서 라우팅 테이블 생성을 클릭한다.

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

캐리어 게이트웨이

DHCP 옵션 세트

탄력적 IP

관리형 접두사 목록

엔드포인트

엔드포인트 서비스

NAT 게이트웨이

피어링 연결

라우팅 테이블 (1/1) 정보							🔄	작업 ▼	라우팅 테이블 생성
🔍 라우팅 테이블 필터링							< 1 > ⚙️		
<input checked="" type="checkbox"/>	Name ▼	라우팅 테이블 ID ▼	명시적 서브넷 연결	엣지 연결	기본 ▼	VPC			
<input checked="" type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989			

2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- 이름을 입력하고 라우팅 테이블을 생성할 VPC를 선택한 후, 우측 하단 라우팅 테이블 생성을 클릭한다.

VPC > 라우팅 테이블 > 라우팅 테이블 생성

라우팅 테이블 생성 정보

라우팅 테이블은 VPC, 인터넷 및 VPN 연결 내 서브넷 간에 패킷이 전달되는 방법을 지정합니다.

라우팅 테이블 설정

이름 - 선택 사항
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

Pub-RTB

VPC
이 라우팅 테이블에 대해 사용할 VPC입니다.

vpc-f4a4c989

태그

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키	값 - 선택 사항
<input type="text" value="Name"/>	<input type="text" value="Pub-RTB"/>

49을(를) 태그.개 더 추가할 수 있습니다.

취소

2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- 목록에서 라우팅 테이블을 확인한다.
- 새로 생성한 테이블과 기본 라우팅 테이블이 보인다.
- 명시적 서브넷 연결 필드에 아무런 값도 없으므로 모두 기본 라우팅 테이블에 연결돼 있음을 알 수 있다.

라우팅 테이블 (2) 정보							
<div>Q 라우팅 테이블 필터링</div>							
<input type="checkbox"/>	Name ▼	라우팅 테이블 ID ▼	명시적 서브넷 연결	엣지 연결	기본 ▼	VPC	
<input type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989	
<input type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	-	-	아니요	vpc-f4a4c989	

2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- Pub-RTB를 선택한 후 우측 상단 작업 메뉴에서 서브넷 연결 편집을 클릭한다

라우팅 테이블 (1/2) 정보

Q 라우팅 테이블 필터링

	Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본
<input type="checkbox"/>	-	rtb-65779f14	-	-	예
<input checked="" type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	-	-	아니오

작업 ▲

- 세부 정보 보기
- 기본 라우팅 테이블 설정
- 서브넷 연결 편집
- 엣지 연결 편집
- 라우팅 전파 편집
- 라우팅 편집
- 태그 관리
- 라우팅 테이블 삭제
- 문제 해결
- 네트워크 연결성 추적

2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- 퍼블릭 용도로 만든 서브넷 2개를 선택하고 하단의 연결 저장을 클릭한다.

서브넷 연결 편집

이 라우팅 테이블과 연결된 서브넷을 변경합니다.

이용 가능한 서브넷 (2/6)

< 1 > ⚙

<input type="checkbox"/>	이름	서브넷 ID	IPv4 CIDR	IPv6 CIDR	라우팅 테이블 ID
<input type="checkbox"/>		subnet-8c6206bd	172.31.48.0/20	-	기본 (rtb-65779f14)
<input checked="" type="checkbox"/>		subnet-cceebc93	172.31.32.0/20	-	기본 (rtb-65779f14)
<input type="checkbox"/>		subnet-31633c10	172.31.80.0/20	-	기본 (rtb-65779f14)
<input checked="" type="checkbox"/>		subnet-a98b6be5	172.31.16.0/20	-	기본 (rtb-65779f14)
<input type="checkbox"/>		subnet-5fb4bf51	172.31.64.0/20	-	기본 (rtb-65779f14)
<input type="checkbox"/>		subnet-fdefbe9b	172.31.0.0/20	-	기본 (rtb-65779f14)

선택한 서브넷

subnet-a98b6be5 ✕ subnet-cceebc93 ✕

취소 연결 저장

2. 경로 제어 : 라우팅 테이블

■ 실습 1. 라우팅 테이블 생성 및 서브넷 연결

- 명시적 서브넷 연결 필드에 '2서브넷' 표시가 나타난다.

라우팅 테이블 (2) 정보

Q 라우팅 테이블 필터링

작업 ▼ 라우팅 테이블 생성

<input type="checkbox"/>	Name ▼	라우팅 테이블 ID ▼	명시적 서브넷 연결	엣지 연결	기본 ▼	VPC
<input type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989
<input type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	2 서브넷	-	아니요	vpc-f4a4c989

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 서비스 > VPC > 인터넷게이트웨이 > (우측상단) 인터넷 게이트웨이 생성 > 이름 지정 > 인터넷 게이트웨이 생성 클릭

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

캐리어 게이트웨이

인터넷 게이트웨이 (1/1) 정보						작업 ▼	인터넷 게이트웨이 생성
Q 인터넷 게이트웨이 필터링						< 1 > ⚙	
☑	Name ▼	인터넷 게이트웨이 ID ▼	상태 ▼	VPC ID ▼	소유자		
☑	-	igw-671ccc1d	✔ Attached	vpc-f4a4c989	262663767358		

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 서비스 > VPC > 인터넷게이트웨이 > (우측상단) 인터넷 게이트웨이 생성 > 이름 지정 > 인터넷 게이트웨이 생성 클릭

VPC > 인터넷 게이트웨이 > 인터넷 게이트웨이 생성

인터넷 게이트웨이 생성 정보

인터넷 게이트웨이는 VPC를 인터넷과 연결하는 가상 라우터입니다. 새 인터넷 게이트웨이를 생성하려면 아래에서 게이트웨이 이름을 지정해야 합니다.

인터넷 게이트웨이 설정

이름 태그
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

태그 - 선택 사항

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키	값 - 선택 사항
<input type="text" value="Name"/> X	<input type="text" value="My-IGW"/> X

49글(을) 태그.개 더 추가할 수 있습니다.

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 연결할 IGW를 선택한 후 우측 상단 작업 메뉴에서 VPC에 연결을 클릭

인터넷 게이트웨이 (1/2) 정보

인터넷 게이트웨이 필터링

	Name	인터넷 게이트웨이 ID	상태	VPC ID
<input checked="" type="checkbox"/>	My-IGW	igw-046fedee45313ea51	Detached	-
<input type="checkbox"/>	-	igw-671ccc1d	Detached	-

작업 ▲

인터넷 게이트웨이 생성

세부 정보 보기

VPC에 연결

VPC에서 분리

태그 관리

인터넷 게이트웨이 삭제

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 연결할 VPC를 선택후 인터넷 게이트웨이 연결 클릭

VPC > 인터넷 게이트웨이 > VPC에 연결(igw-046fedee45313ea51)

VPC에 연결(igw-046fedee45313ea51) 정보

VPC

인터넷 게이트웨이를 VPC에 연결하여 인터넷과의 통신을 활성화합니다. 아래에서 연결하려는 VPC를 지정하십시오.

사용 가능한 VPC

인터넷 게이트웨이를 이 VPC에 연결합니다.

Q vpc-f4a4c989



▶ AWS Command Line Interface 명령

취소

인터넷 게이트웨이 연결

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- Pub-RTB 라우팅 테이블을 선택하고 하단 라우팅 탭과 라우팅 편집을 차례로 클릭한다.

라우팅 테이블 (1/2) 정보

Q 라우팅 테이블 필터링

< 1 > ⚙

	Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본	VPC
<input type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989
<input checked="" type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	2 서브넷	-	아니요	vpc-f4a4c989

rtb-03f99f1a28503ed63 / Pub-RTB

세부 정보 라우팅 서브넷 연결 엣지 연결 라우팅 전파 태그

라우팅 (1)

Q 라우팅 필터링

모두 < 1 > ⚙

라우팅 편집

대상	대상	상태	전파됨
172.31.0.0/16	local	🟢 활성화	아니요

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 라우팅 추가 버튼을 클릭한 후 대상을 클릭해서 인터넷 게이트웨이를 선택하면 VPC에 연결된 IGW를 볼 수 있다.
- IGW를 선택한 뒤 하단 변경 사항 저장을 클릭한다.

VPC > 라우팅 테이블 > rtb-03f99f1a28503ed63 > 라우팅 편집

라우팅 편집

대상	대상	상태	전파됨
172.31.0.0/16	Q local X	✔ 활성화	아니요
Q 0.0.0.0/0 X	Q 캐리어 게이트웨이 코어 네트워크 외부 전용 인터넷 게이트웨이 Gateway Load Balancer 엔드포인트 인스턴스 인터넷 게이트웨이 로컬	-	아니요

라우팅 추가

취소 미리 보기 **변경 사항 저장**

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 라우팅 테이블 목록이 추가된 것을 확인할 수 있다.

rtb-03f99f1a28503ed63 / Pub-RTB

세부 정보 | **라우팅** | 서브넷 연결 | 엣지 연결 | 라우팅 전파 | 태그

라우팅 (2) 라우팅 편집

Q 라우팅 필터링 모두 ▼ < 1 > ⚙

대상 ▼	대상 ▼	상태 ▼	전파됨 ▼
0.0.0.0/0	igw-046fedee45313ea51	✔ 활성화	아니요
172.31.0.0/16	local	✔ 활성화	아니요

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 라우팅을 그대로 둔 채 타깃 리소스를 삭제 또는 분리하면 블랙홀 상태로 변한다.
- IGW를 VPC에서 분리(Detach)한 뒤, 라우팅을 확인해보자.

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

인터넷 게이트웨이 (1/2) 정보

인터넷 게이트웨이 필터링

	Name	인터넷 게이트웨이 ID	상태	VPC ID
<input checked="" type="checkbox"/>	My-IGW	igw-046fedee45313ea51	Attached	vpc-f4a4c989
<input type="checkbox"/>	-	igw-671ccc1d	Detached	-

작업

세부 정보 보기

VPC에 연결

VPC에서 분리

태그 관리

인터넷 게이트웨이 삭제

인터넷 게이트웨이 생성

1 > ⚙

VPC에서 분리

인터넷 게이트웨이 `igw-046fedee45313ea51(My-IGW)`을(를) VPC `vpc-f4a4c989`에서 분리 하시겠습니까?

인터넷 게이트웨이를 분리하면 VPC의 리소스가 인터넷과 통신할 수 없습니다.

취소

인터넷 게이트웨이 분리

2. 경로 제어 : 라우팅 테이블

■ 실습 2. 라우팅 추가 및 Backhole 상태 확인

- 라우팅을 그대로 둔 채 타깃 리소스를 삭제 또는 분리하면 블랙홀 상태로 변한다.
- IGW를 VPC에서 분리(Detach)한 뒤, 라우팅을 확인해보자.
- 이처럼 트래픽을 수신할 수 없거나 라우팅보다 게이트웨이 리소스가 먼저 삭제되면 블랙홀 상태가 된다.

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

라우팅 테이블 (1/2) 정보						
Q 라우팅 테이블 필터링						
	Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본	VPC
<input type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989
<input checked="" type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	2 서브넷	-	아니요	vpc-f4a4c989

rtb-03f99f1a28503ed63 / Pub-RTB

세부 정보

라우팅

서브넷 연결

엣지 연결

라우팅 전파

태그

라우팅 (2)

Q 라우팅 필터링

모두

라우팅 편집

대상	대상	상태	전파됨
0.0.0.0/0	igw-046fedee45313ea51	⊗ 블랙홀	아니요
172.31.0.0/16	local	⊙ 활성	아니요

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- VPC에서 분리한 IGW를 다시 연결한다.

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트
웨이

인터넷 게이트웨이 (1/2) 정보

인터넷 게이트웨이 필터링

	Name	인터넷 게이트웨이 ID	상태	VPC ID
<input checked="" type="checkbox"/>	My-IGW	igw-046fedee45313ea51	Detached	-
<input type="checkbox"/>	-	igw-671ccc1d	Detached	-

작업

- 세부 정보 보기
- VPC에 연결**
- VPC에서 분리
- 태그 관리
- 인터넷 게이트웨이 삭제

인터넷 게이트웨이 생성

1 > ⚙

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- Pub-RTB 라우팅 테이블을 선택하고 작업 > 라우팅 편집을 클릭한다.

▼ Virtual Private Cloud

VPC

서브넷

라우팅 테이블

인터넷 게이트웨이

송신 전용 인터넷 게이트웨이

라우팅 테이블 (1/2) 정보

Q 라우팅 테이블 필터링

	Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본
<input type="checkbox"/>	-	rtb-65779f14	-	-	예
<input checked="" type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	2 서브넷	-	아니

작업 ▲

라우팅 테이블 생성

- 세부 정보 보기
- 기본 라우팅 테이블 설정
- 서브넷 연결 편집
- 엣지 연결 편집
- 라우팅 전파 편집
- 라우팅 편집
- 태그 관리

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- IGW 라우팅을 삭제하고 변경 사항 저장을 클릭한다.

VPC > 라우팅 테이블 > rtb-03f99f1a28503ed63 > 라우팅 편집

라우팅 편집

대상	대상	상태	전파됨
172.31.0.0/16	<input type="text" value="local"/>	✔ 활성화	아니요
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="igw-046fedee45313ea51"/>	✔ 활성화	아니요
<input type="button" value="라우팅 추가"/>			

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- Pub-RTB 라우팅 테이블이 선택된 상태에서 엣지 연결 탭과 엣지 연결 편집을 차례로 클릭한다.

라우팅 테이블 (1/2) 정보

라우팅 테이블 필터링

	Name	라우팅 테이블 ID	명시적 서브넷 연결	엣지 연결	기본	VPC
<input type="checkbox"/>	-	rtb-65779f14	-	-	예	vpc-f4a4c989
<input checked="" type="checkbox"/>	Pub-RTB	rtb-03f99f1a28503ed63	2 서브넷	-	아니요	vpc-f4a4c989

rtb-03f99f1a28503ed63 / Pub-RTB

세부 정보 | 라우팅 | 서브넷 연결 | **엣지 연결** | 라우팅 전파 | 태그

엣지 연결 편집

엣지 연결이 없습니다.

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- VPC에 연결된 IGW를 선택하고 변경 사항 저장을 클릭한다.

VPC > 라우팅 테이블 > rtb-03f99f1a28503ed63 > 엣지 연결 편집

엣지 연결 편집 (1/1)

라우팅 테이블 기본 세부 정보

라우팅 테이블 ID rtb-03f99f1a28503ed63	라우팅 테이블 이름 Pub-RTB	라우팅 테이블 VPC ID vpc-f4a4c989
-------------------------------------	-----------------------	--------------------------------

인터넷 게이트웨이 ☒

게이트웨이 ID
igw-046fedee45313ea51 / My-IGW [🔗](#)

상태
Attached

소유자
262663767358

취소

2. 경로 제어 : 라우팅 테이블

■ 실습 3. 엣지 연결

- VPC에 연결된 IGW를 선택하고 변경 사항 저장을 클릭하면 라우팅 테이블이 IGW에 연결된 모습을 볼 수 있다.

rtb-03f99f1a28503ed63 / Pub-RTB

세부 정보

라우팅

서브넷 연결

엣지 연결

라우팅 전파

태그

연결된 인터넷 게이트웨이 (1)

엣지 연결 편집

Q 인터넷 게이트웨이 찾기

< 1 > ⚙

ID	상태	VPC	소유자
igw-046fedee45313ea51 / My-IGW	🟢 연결됨	vpc-f4a4c989	262663767358



Thank You
