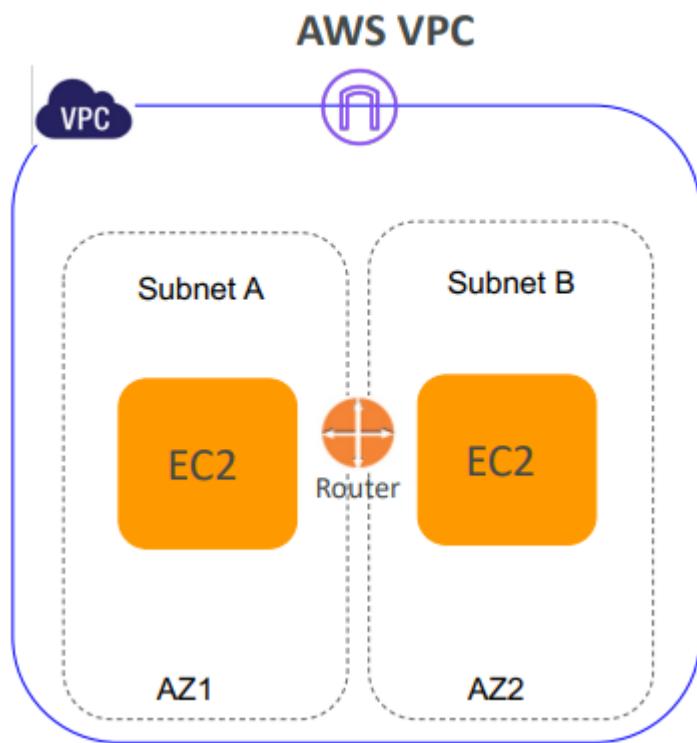


# ANS Study

## 1. Amazon VPC Fundamentals

### What is VPC?

- Amazon VPC는 AWS 클라우드에서 사용자가 정의한 가상 네트워크 환경으로, 온프레미스 네트워크와 유사한 구조를 제공합니다.
- AWS 클라우드의 리소스를 가상 네트워크에 배치하고, 네트워크 설정을 세부적으로 제어할 수 있습니다.

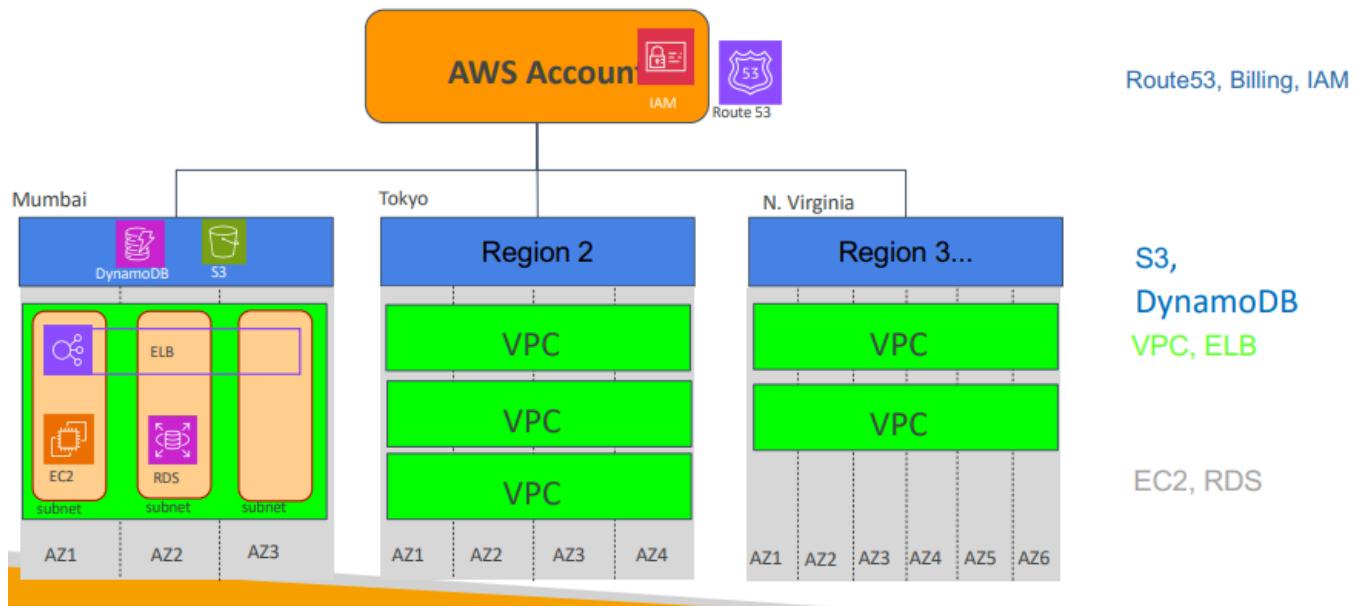


- 서비스의 특징
  - 가상 네트워크 구성:
    - CIDR(Classless Inter-Domain Routing)을 사용하여 IP 주소 범위를 정의.
    - 퍼블릭 및 프라이빗 서브넷을 사용하여 네트워크를 세분화.
  - 유연한 연결 옵션:
    - 인터넷 게이트웨이, NAT 게이트웨이, VPN, Direct Connect를 통해 외부와 통신 가능.
  - 보안 및 제어:
    - 보안 그룹과 네트워크 ACL을 통해 세부적인 보안 규칙 설정.
    - 로컬 트래픽과 외부 트래픽에 대해 라우팅 제어.

- VPC 활용 사례
  - 온프레미스 네트워크와 연결하여 하이브리드 클라우드 환경 구성.
  - AWS 리소스 간 안전한 네트워크 통신.
  - 외부 인터넷 액세스가 필요 없는 내부 애플리케이션 배치.
- 특장점:
  - 네트워크 설정의 유연성 제공.
  - 온프레미스 네트워크와 원활하게 통합 가능.
  - AWS의 확장성을 활용하면서 보안 제어 가능.
- 유의사항:
  - 네트워크 설계가 복잡할 수 있음.
  - 잘못된 설정은 보안 문제를 유발할 수 있음.

## Scope of VPC with respect to AWS Account, Region & AZ

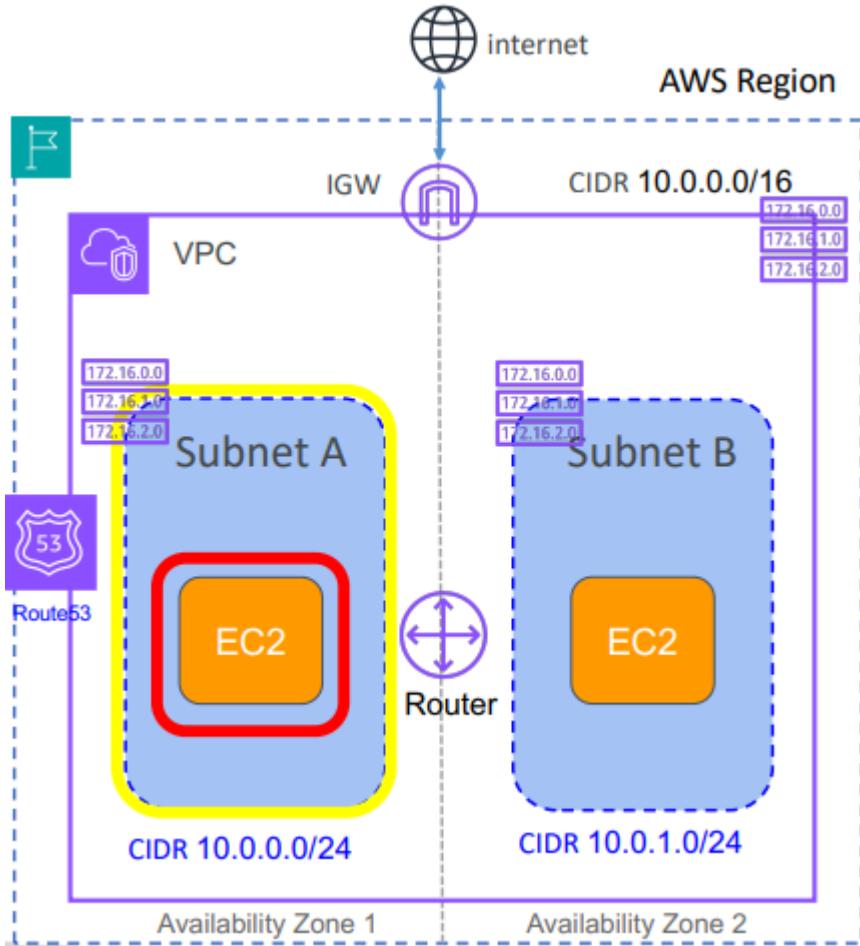
- AWS 서비스 범위
  - 글로벌 서비스: AWS 계정, IAM, Route 53과 같은 서비스는 글로벌로 작동하며, 리전 제한이 없습니다.
  - 리전 기반 서비스: EC2, RDS, VPC 등은 특정 리전 내에서만 작동하며, 데이터는 리전 간 자동 복제되지 않음(사용자 설정 필요).
  - 가용 영역(AZ): 리전 내 독립적으로 운영되는 데이터 센터. 장애 격리와 고가용성을 위해 여러 AZ 사용 권장.
  - VPC: 특정 리전 내에서 작동하며, 서브넷 및 라우트 테이블로 네트워크 트래픽 관리.



- 서비스 범위별 주요 사항
  - 글로벌 서비스:

- IAM: 사용자 및 권한 관리.
- Route 53: 글로벌 DNS 관리.
- CloudFront: 글로벌 콘텐츠 배포.
- 리전 기반 서비스:
  - EC2, RDS 등은 리전 내에서만 리소스를 실행.
  - 리전 간 연결 시 추가 설정 필요.
- AZ 기반 리소스:
  - 서브넷과 EC2 인스턴스는 특정 AZ에 위치.
  - 고가용성을 위해 다중 AZ 배포 권장.
- VPC:
  - 리전 내의 논리적 네트워크 격리.
  - 서브넷을 통해 AZ 단위로 네트워크 세분화.
- AWS 관련 서비스
  - 글로벌 서비스: IAM, Route 53, CloudFront.
  - 리전 서비스: EC2, RDS, VPC.
  - AZ 기반 서비스: 서브넷, EC2 인스턴스.
- Pros:
  - 서비스가 글로벌, 리전, AZ로 나뉘어 있어 설계가 유연함.
  - AZ를 활용한 고가용성 및 장애 격리 가능.
- Cons:
  - 리전 간 데이터 전송 시 추가 비용 발생.
  - 복잡한 리전 간 서비스 통합이 필요할 수 있음.

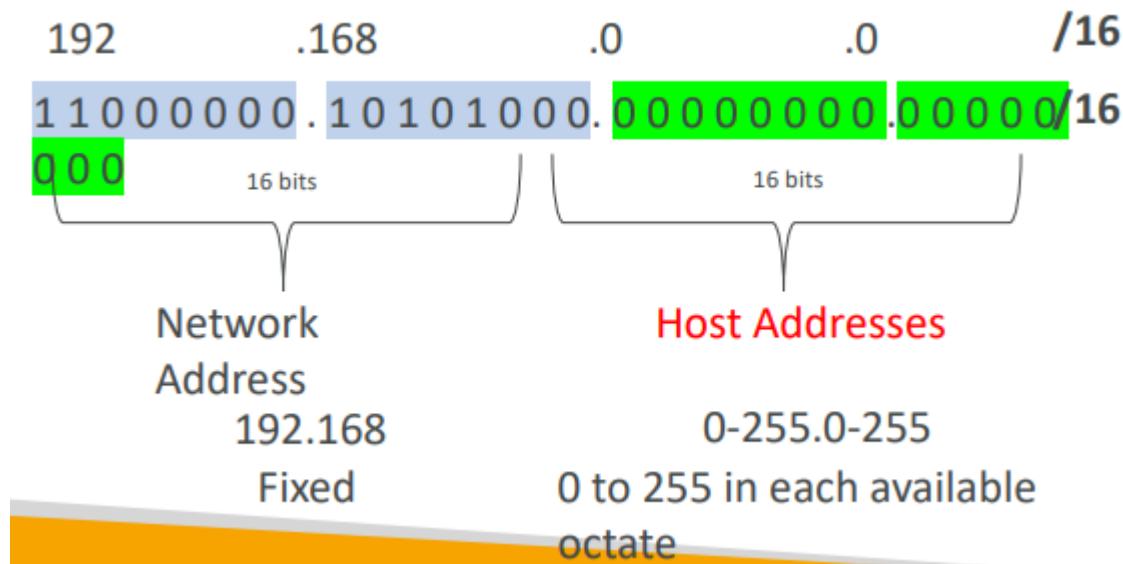
## VPC Building Blocks - Core Components



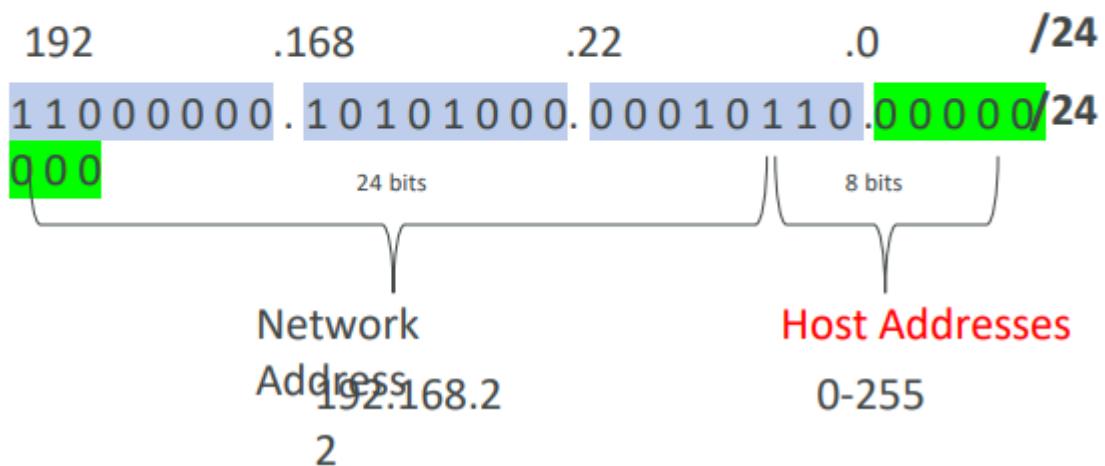
- CIDR (Classless Inter-Domain Routing)
  - VPC의 IP 주소 범위를 정의합니다. IPv4는 /16에서 /28 범위, IPv6는 /56 범위를 사용합니다.
  - AWS 권장 프라이빗 IP 범위:
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
- Subnets
  - VPC를 더 작은 서브넷으로 나누어 각 가용 영역(AZ)에 배치됩니다.
  - 서브넷은 퍼블릭과 프라이빗으로 나누며, 퍼블릭은 인터넷 게이트웨이를 통해 인터넷과 연결됩니다.
- Route Tables
  - 서브넷의 트래픽 라우팅을 정의합니다.
  - VPC는 기본 라우팅 테이블(Main Route Table)을 갖추고 있으며, 사용자 정의 라우팅 테이블을 추가할 수 있습니다.
- Internet Gateway (IGW)
  - VPC와 인터넷 간의 통신을 지원하는 가상 게이트웨이입니다.
- Security Groups

- EC2 인스턴스와 같은 자원에 대한 네트워크 트래픽을 제어합니다.
- 상태 저장 방식을 사용하며, 인바운드와 아웃바운드 규칙을 설정할 수 있습니다.
- Network Access Control List (NACL)
  - 서브넷 수준에서 적용되며 트래픽을 허용하거나 차단하는 규칙을 설정합니다.
  - 상태 비저장 방식을 사용하며, 명시적으로 아웃바운드 트래픽을 열어야 합니다.
- Domain Name Server (DNS)
  - VPC의 DNS 해석을 제공합니다.
  - 기본적으로 AWS 제공 DNS 서버(Route 53 Resolver)를 사용합니다.

## VPC Addressing (CIDR)



Example: IPv4 CIDR 192.168.22.0/24



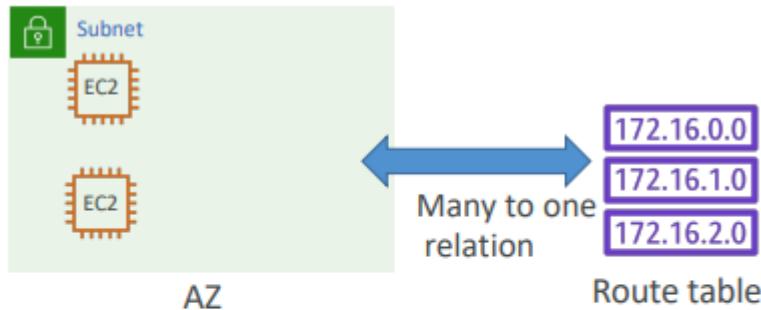
- CIDR (Classless Inter-Domain Routing):
  - IP 주소를 할당하는 체계로, 기존의 A/B/C 클래스 방식의 단점을 보완.
  - IPv4 주소는 네트워크 주소와 호스트 주소로 구분되며, 예시: 192.168.0.0/24는 256개의 IP를 포함.
  - IPv6 주소는 글로벌 고유 주소를 제공하며, AWS에서는 /56 또는 /64 블록 크기를 사용.
- IPv4 VPC CIDR:
  - VPC 내 IP 주소 범위는 /16에서 /28까지 지원.
  - AWS 권장 프라이빗 IP 범위:
    - 10.0.0.0/8 (16,777,216개 주소)
    - 172.16.0.0/12 (1,048,576개 주소)
    - 192.168.0.0/16 (65,536개 주소)
- IPv6 VPC CIDR:
  - IPv6 주소는 AWS에서 자동 할당되며, 서브넷 크기는 /64.
  - 글로벌 고유 주소로 인터넷 연결 가능.
  - IPv6는 AWS에서 기본적으로 퍼블릭 주소로 처리됨.
- 서브넷 주소 할당:
  - VPC CIDR에서 서브넷 CIDR은 /16에서 /28 범위로 정의.
  - 각 서브넷은 AWS가 5개의 IP를 예약:
    - 네트워크 주소, 라우터, DNS 매핑, AWS 예약, 브로드캐스트 주소.
- VPC CIDR의 유연성:
  - 멀티 VPC 환경에서 중복된 CIDR을 피하기 위해 신중한 설계 필요.
  - CIDR 크기는 필요한 호스트 수에 맞게 설정하되, AWS 예약 IP를 고려.
- 주요 포인트:
  - 시험에서 CIDR 블록의 크기 및 IP 예약 규칙과 관련된 문제가 자주 등장.

Feature	IPv4	IPv6
Address Size	32-bit (4 옥텟)	128-bit (8 블록)
Address Format	점으로 구분된 10진수 (예: 192.168.1.1)	콜론으로 구분된 16진수 (예: 2001:0db8::1)
Address Space	약 43억 개의 주소	거의 무한대 ( $2^{128}$ 주소)
Deployment	현재 가장 널리 사용됨	주로 신규 네트워크 및 확장용
Public/Private	공용/비공용 구분	공용만 사용 (비공용 개념 없음)
Subnet Size	가변적, /8 ~ /32 사용 가능	고정된 서브넷 크기 (/64)
Security	옵션으로 IPsec 지원	기본적으로 IPsec 포함
NAT 필요 여부	공용 IP 부족 문제로 NAT 필요	넉넉한 주소 공간으로 NAT 불필요

Feature	IPv4	IPv6
DNS Support	Amazon DNS를 통해 기본적으로 지원	Amazon DNS로 지원, 복잡성 증가
Header Size	20 bytes	40 bytes
Compatibility	기존 인터넷 프로토콜과 완전 호환	IPv4와 직접 호환되지 않음

## VPC Route Tables

Subnet A - 10.0.0.0/24



Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

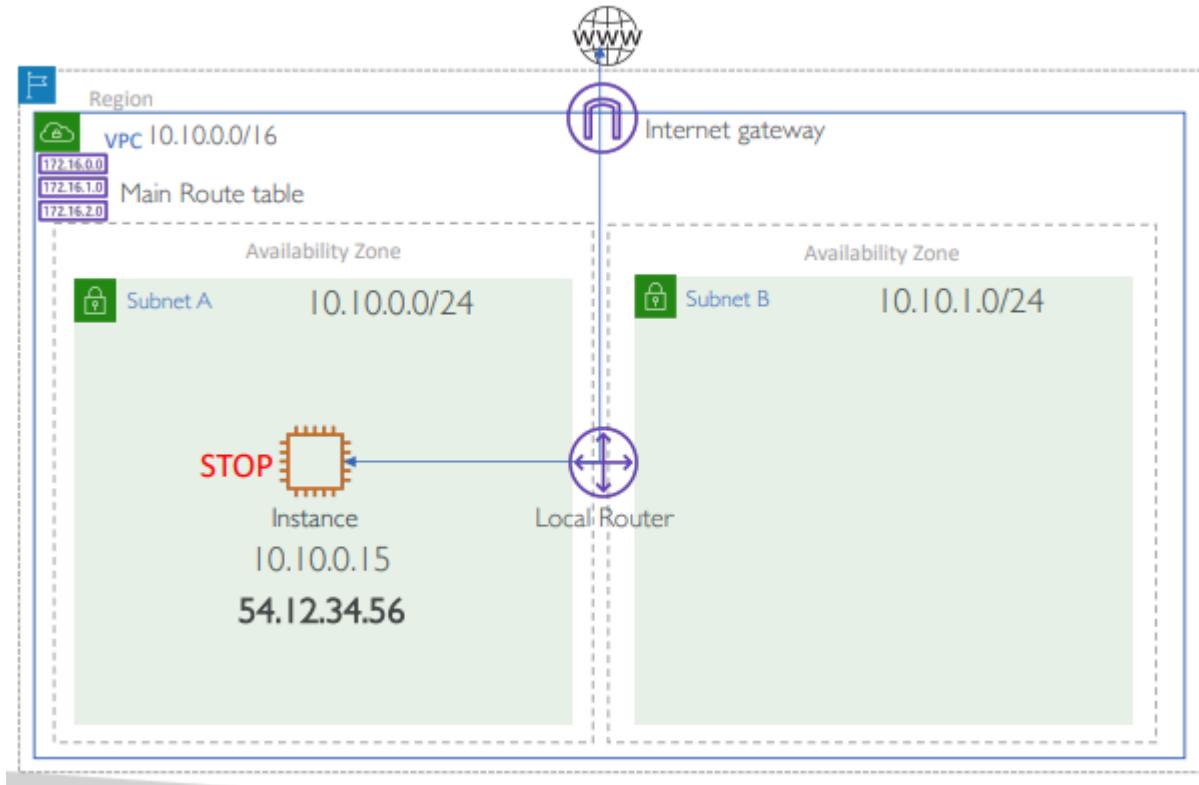
- 라우트 테이블이란?
  - VPC의 서브넷이나 게이트웨이를 통해 트래픽을 라우팅하는 데 사용됩니다.
  - 라우트 테이블은 여러 서브넷과 연결될 수 있습니다.
- 기본 특징
  - VPC 생성 시 기본(Main) 라우트 테이블이 자동으로 생성됩니다.
  - 모든 서브넷은 기본적으로 메인 라우트 테이블과 연결됩니다.
  - 각 라우트 테이블은 로컬 트래픽에 대한 기본 라우트(immutable local route)를 포함합니다.
- 커스텀 라우트 테이블
  - 서브넷별 커스텀 라우트 테이블을 생성해 특정 트래픽 흐름을 설정할 수 있습니다.
  - 예: 인터넷 게이트웨이를 추가하여 인터넷에 액세스 가능하도록 설정.
- 라우트 테이블 항목
  - 대상(Destination): 트래픽이 이동할 IP 범위 (예: 0.0.0.0/0).
  - 대상(Target): 트래픽이 이동할 게이트웨이, NAT, 네트워크 인터페이스 등.
- 예시

- 인터넷 트래픽용 라우트:
  - 대상: 0.0.0.0/0, 타겟: 인터넷 게이트웨이(IGW).
- 온프레미스 연결:
  - 대상: 특정 IP CIDR, 타겟: 가상 프라이빗 게이트웨이(VGW).
- 장점
  - 서브넷 수준에서 세밀한 트래픽 라우팅을 지원.
  - 다양한 AWS 게이트웨이(NAT Gateway, IGW, VGW)와 통합.
- 단점
  - 잘못된 라우트 설정 시 연결 오류 발생 가능.
  - 복잡한 네트워크 설계에서는 라우트 관리가 어려울 수 있음.

## Caution

- AWS는 각 서브넷에서 5개의 IP 주소(처음 4개와 마지막 1개)를 예약하며, 이는 인스턴스에 할당할 수 없습니다. 예를 들어, CIDR 블록이 10.0.0.0/24인 경우 다음 IP가 예약됩니다:
- 10.0.0.0: 네트워크 주소.
- 10.0.0.1: VPC 라우터에 의해 예약.
- 10.0.0.2: Amazon 제공 DNS와 매핑을 위해 예약.
- 10.0.0.3: 미래 사용을 위해 AWS에서 예약.
- 10.0.0.255: 네트워크 브로드캐스트 주소(브로드캐스트는 VPC에서 지원되지 않음).
- 시험 팁:
  - EC2 인스턴스에 필요한 IP 주소가 29개인 경우, 서브넷 크기로 /27(32개의 IP)을 선택하면 사용할 수 있는 IP가 27개밖에 되지 않아 적합하지 않습니다. 대신, /26(64개의 IP)을 선택해야 합니다.

## IP Addresses - Private vs Public vs Elastic & Allocation

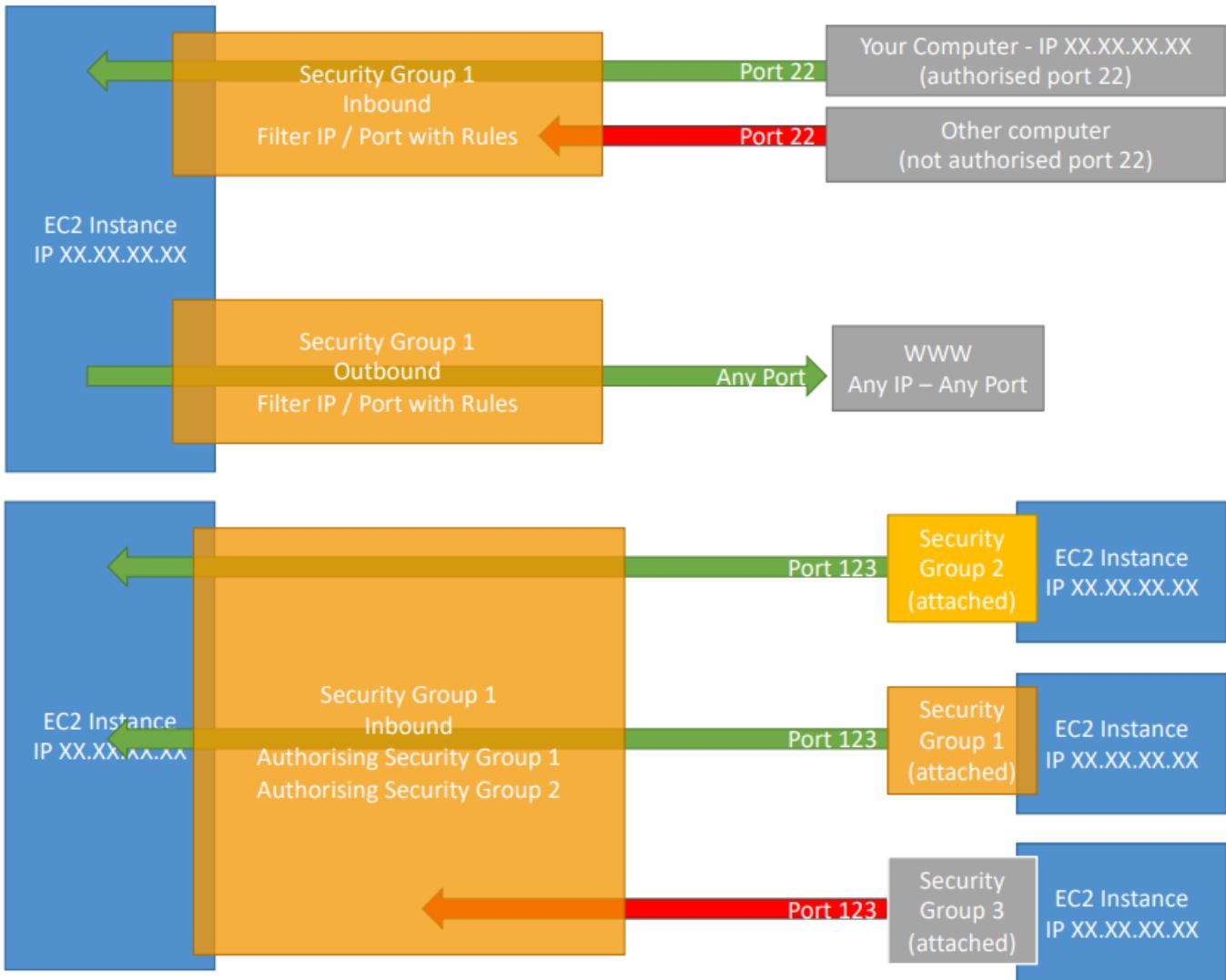


- Private IP
  - 정의: VPC의 내부 네트워크 통신용 IP 주소. 인터넷에서는 사용되지 않음.
  - 할당: EC2 인스턴스 생성 시 자동으로 할당.
  - 사용 사례: 애플리케이션 서버, 데이터베이스 서버와 같은 내부 통신 필요 리소스.
- Public IP
  - 정의: 인터넷 통신이 가능한 IP 주소. AWS IP 풀에서 할당.
  - 할당 조건: 퍼블릭 서브넷에 생성된 EC2 인스턴스가 퍼블릭 IP를 받을 수 있도록 설정.
  - 특징:
    - 인스턴스가 정지되거나 종료되면 IP가 해제.
    - 사용 사례: 웹 서버, 인터넷과 직접 통신이 필요한 리소스.
- Elastic IP (EIP)
  - 정의: 고정된 공용 IP 주소. 인스턴스에 할당하여 정적 IP로 유지.
  - 할당 및 특징:
    - 명시적으로 요청하여 AWS 계정에 할당.
    - 인스턴스가 종료되거나 정지되더라도 해제되지 않음.
    - 각 계정에 기본적으로 5개의 EIP 할당 제한.
  - 사용 사례: DNS 레코드가 변경되지 않도록 고정 IP 유지가 필요한 경우.
- 주소 재할당
  - Elastic IP는 다른 인스턴스에 재할당 가능.
  - Public IP는 인스턴스 재시작 시 새로운 주소로 변경.
- 장점

- Private IP는 비용 효율적이며 내부 네트워크 보안을 유지.
- Elastic IP는 고정 IP 제공으로 네트워크 설계 유연성 증가.
- 단점
  - Public IP는 정지/종료 시 변경되어 고정 IP 필요 시 Elastic IP를 사용해야 함.
  - Elastic IP는 사용하지 않을 경우에도 비용 발생.

Feature	Private IP	Public IP	Elastic IP
<b>Definition</b>	VPC 내부 통신용 비 공개 IP 주소	인터넷에서 직접 통신 가능한 공용 IP 주소	고정 공용 IP 주소, AWS 계정에 할당
<b>Allocation</b>	VPC 서브넷 CIDR 범위에서 자동 할당	퍼블릭 서브넷에서 자동 할당 가능	명시적으로 요청 후 AWS 계정에 할당
<b>Address Range</b>	RFC 1918 개인 네트워크 범위 내 주소	AWS가 제공하는 공용 IP 풀	AWS가 제공하는 공용 IP 풀
<b>Persistence</b>	인스턴스 정지 시 해제되지 않음	인스턴스 정지/종료 시 새 주소로 변경	인스턴스 정지/종료에도 유지
<b>Cost</b>	무료	과금 (since 2024.2.14 ~ )	사용하지 않을 경우 요금 발생
<b>Use Case</b>	내부 애플리케이션, 데이터베이스 서버	웹 서버, 인터넷과 통신하는 리소스	DNS 고정 및 고정된 IP가 필요한 서비스
<b>IP Changeability</b>	변경되지 않음	인스턴스 재시작 시 변경 가능	고정, 다른 인스턴스에 재할당 가능
<b>Internet Access</b>	NAT 또는 게이트웨이 필요	가능	가능

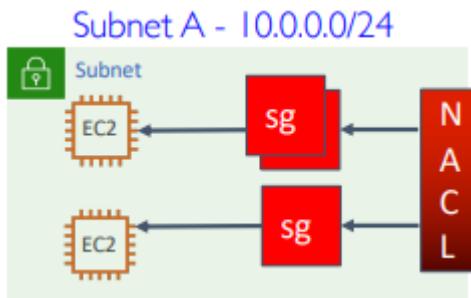
## VPC Firewall: Security Group



- Security Group이란?
  - EC2 인스턴스 및 리소스에 적용되는 가상 방화벽.
  - 인바운드(수신) 및 아웃바운드(송신) 트래픽 제어.
  - VPC 내의 인스턴스 수준에서 작동.
- 주요 특징
  - 상태 유지(Stateful):
    - 허용된 인바운드 트래픽은 자동으로 해당 연결의 아웃바운드 트래픽 허용.
    - 아웃바운드 트래픽도 동일하게 동작.
  - 기본 동작:
    - 모든 인바운드 트래픽은 차단.
    - 모든 아웃바운드 트래픽은 허용.
    - 허용 규칙만 적용 가능: 차단(Deny) 규칙은 지원하지 않음.
- 구성 요소
  - 프로토콜: TCP, UDP, ICMP 등.
  - 포트 범위: 특정 포트 또는 범위 설정.

- 소스/대상: CIDR, IP 주소, 또는 다른 Security Group.
- 예시
  - SSH 허용:
    - 프로토콜: TCP, 포트: 22, 소스: 203.0.113.0/24.
  - HTTP 트래픽 허용:
    - 프로토콜: TCP, 포트: 80, 소스: 0.0.0.0/0.
- 참조 가능
  - 다른 Security Group을 소스로 참조 가능하여 유연한 네트워크 정책 생성.
- 장점
  - 인스턴스 단위에서 세밀한 트래픽 제어 가능.
  - 동적 소스 참조를 통해 네트워크 구성 간소화.
- 단점
  - 블랙리스트(Deny) 정책 설정 불가능.
  - 서브넷 전체가 아닌 개별 리소스에만 적용.

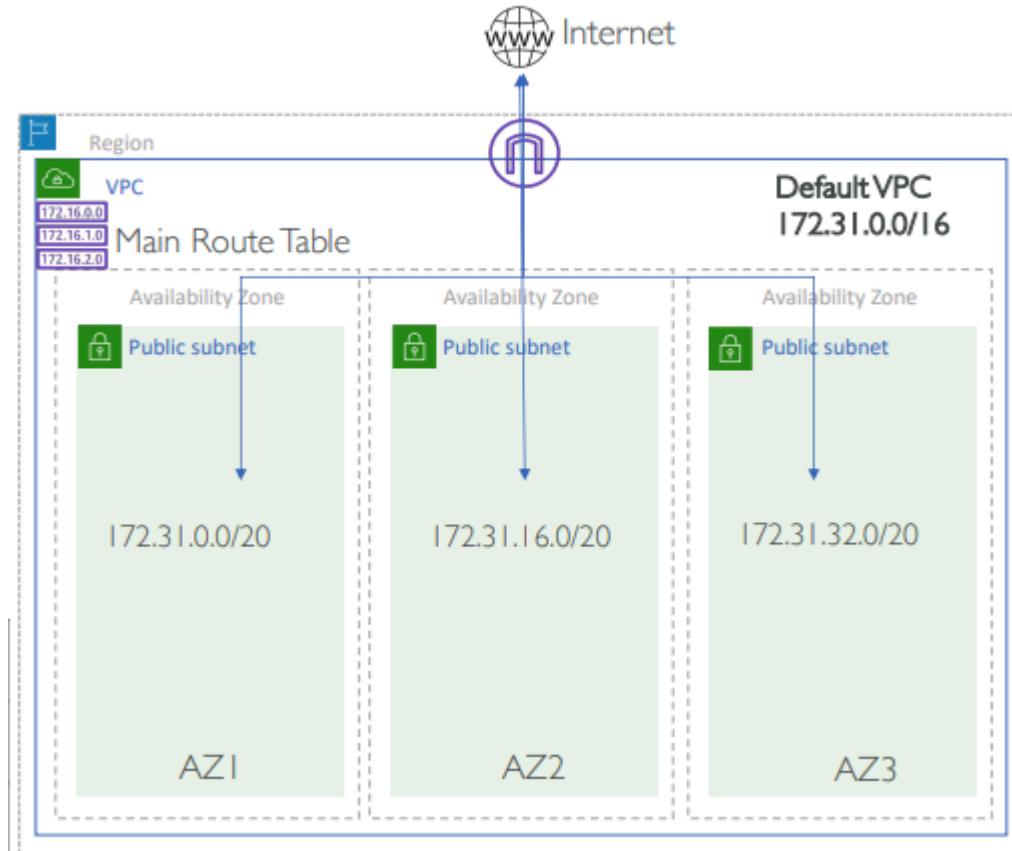
## VPC Firewall: Network Access Control List (NACL)



- NACL이란?
  - VPC의 서브넷 수준에서 트래픽을 제어하는 방화벽.
  - 서브넷에 적용되며, 서브넷에 연결된 모든 인스턴스에 영향을 미침.
  - 보안 그룹과는 달리 Stateless(상태 비저장) 동작.
- 주요 특징
  - 규칙 기반 접근 제어:
    - 허용(Allow)과 거부(Deny) 규칙 모두 지원.
    - 우선 순위 기반 규칙:
      - 규칙 번호가 낮을수록 우선 순위가 높음.
      - 규칙이 일치하면 해당 트래픽에 대해 허용 또는 거부 처리.
  - 기본 NACL 동작:
    - 새로 생성된 NACL은 모든 트래픽을 허용.
    - 기본 규칙은 삭제 가능.
- 구성 요소

- 규칙 번호: 각 규칙에 대한 우선 순위.
  - 타입: 트래픽 유형 (예: HTTP, HTTPS, SSH).
  - 프로토콜: TCP, UDP, ICMP 등.
  - 소스/대상: 특정 IP 주소, CIDR 블록.
  - 작업: 허용(Allow) 또는 거부(Deny).
- 예시
    - SSH 액세스 차단:
      - 규칙 번호: 100, 프로토콜: TCP, 포트: 22, 소스: 0.0.0.0/0, 작업: Deny.
    - HTTPS 허용:
      - 규칙 번호: 101, 프로토콜: TCP, 포트: 443, 소스: 0.0.0.0/0, 작업: Allow.
  - 장점
    - 서브넷 수준에서 전체 트래픽 차단 가능.
    - 특정 IP 주소나 범위를 기반으로 트래픽 제어.
  - 단점
    - Stateless 동작으로 인해 인바운드와 아웃바운드 규칙을 각각 정의해야 함.
    - 잘못된 규칙 구성 시 전체 서브넷의 트래픽 차단 가능.

## Default VPC



Default VPC는 AWS에서 자동으로 생성되는 기본 VPC 환경으로, 다음과 같은 특징을 가지고 있습니다:

- 각 AWS 리전마다 하나의 Default VPC가 생성됩니다.
- CIDR 범위는 **172.31.0.0/16**으로 설정됩니다.
- 가용 영역(AZ)마다 /20 크기의 서브넷이 기본적으로 생성됩니다.
- Default VPC는 이미 **인터넷 게이트웨이(Internet Gateway)**가 연결되어 있으며, 모든 서브넷이 기본적으로 퍼블릭 서브넷으로 설정되어 외부와 통신이 가능합니다.
- 사용자는 Default VPC를 수정하거나 삭제할 수 있으며, 삭제 후에는 콘솔을 통해 재생성할 수 있습니다.

## 장점

- AWS에서 기본적으로 제공하므로 별도의 설정 없이 VPC 환경을 빠르게 사용할 수 있음.
- 초보자와 테스트 환경에 적합.

## 단점

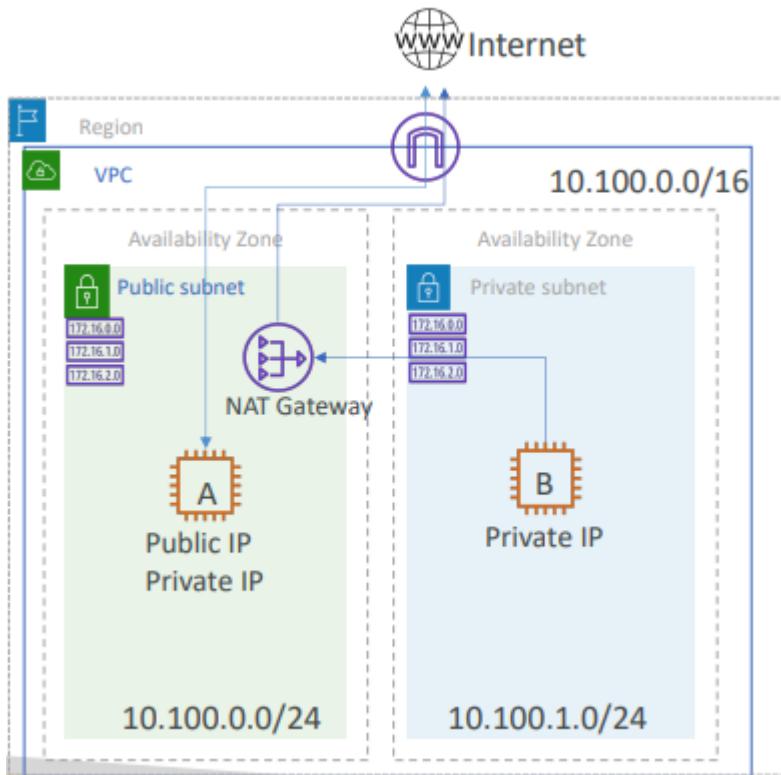
- 기본적으로 모든 서브넷이 퍼블릭 서브넷이므로 보안상의 위험이 있을 수 있음.
- 실제 프로덕션 환경에는 적합하지 않음.

## Nat Gateway

NAT Gateway는 Amazon VPC에서 사설 서브넷에 있는 리소스가 인터넷이나 다른 AWS 서비스에 안전하게 접근할 수 있도록 지원하는 관리형 네트워크 구성 요소입니다. NAT Gateway는 들어오는 트래픽을 차단하면서 아웃바운드 트래픽만 허용합니다.

## 특징

- **보안 강화**: 사설 서브넷의 리소스에 대해 들어오는 트래픽을 차단하고, 아웃바운드 인터넷 접근만 허용.
- **고가용성**: NAT Gateway는 기본적으로 AZ별로 고가용성을 제공하며, 장애 복구 없이 자동으로 동작.
- **확장성**: 고성능 네트워크 처리량(최대 45Gbps)을 지원하며, 트래픽 양에 따라 자동 확장.
- **간편성**: AWS에서 관리되며, 유지보수나 설정이 간단.



## Multi AZ 구성

NAT Gateway의 Multi-AZ 관련 주요 사항 - NAT Gateway는 특정 가용 영역(AZ)에 배포됩니다. - 고가용성을 위해 각 AZ에 NAT Gateway를 각각 생성해야 합니다. - 각 사설 서브넷은 동일한 AZ의 NAT Gateway와 연결되도록 라우팅 테이블을 구성해야 합니다. - NAT Gateway는 AZ 수준의 장애 복구를 보장하며, 한 AZ의 NAT Gateway가 중단되더라도 다른 AZ의 NAT Gateway에는 영향을 미치지 않습니다. - 다중 AZ를 구성하면 추가적인 비용이 발생하지만, 중요한 워크로드에서 높은 가용성을 보장합니다.

## 장점

- AWS 관리형 서비스로 구성 및 유지보수에 부담이 없음.
- 동시 연결 및 데이터 처리량에 제한이 적어 대규모 환경에 적합.
- IP 주소 자동 할당 및 관리 기능 제공.

## 단점

- NAT Gateway 사용에 따른 추가 비용 발생.
- 가용 영역별로 NAT Gateway를 구성해야 하므로 설정이 단순하지만 비용이 증가할 수 있음.

## 특이점

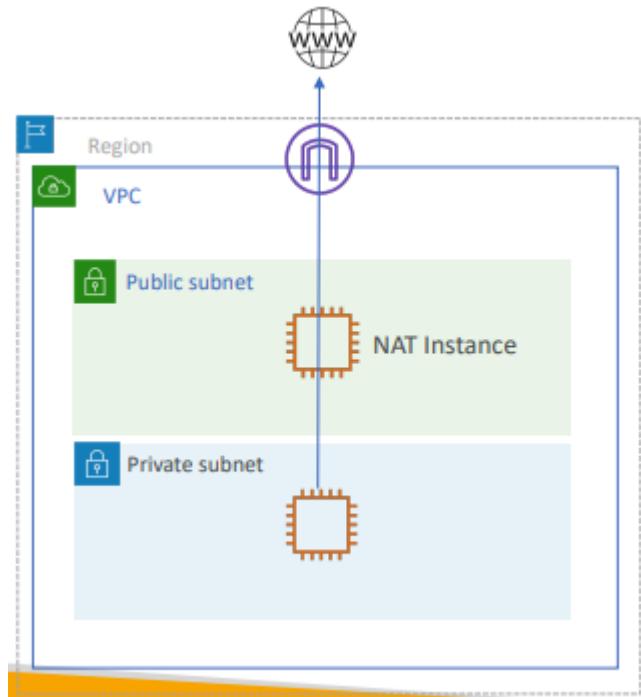
- NAT Gateway는 아웃바운드 트래픽만 처리하며, 들어오는 요청은 지원하지 않음.
- 가용 영역 단위로 배포되며, 다중 AZ 아키텍처에서는 각 AZ에 하나씩 생성하는 것이 권장됨.

- IPv6에서는 NAT Gateway를 사용할 수 없으며, 대체로 Egress-only Internet Gateway를 사용.

## Nat Instance (EC2 base NAT)

### 요약

NAT Instance는 Amazon EC2 인스턴스를 활용하여 VPC의 사설 서브넷에 위치한 리소스가 인터넷에 접근할 수 있도록 구성하는 방법입니다. NAT Gateway와 비교하여 더 많은 사용자 정의와 비용 효율성을 제공하지만, 수동 관리가 필요합니다.



### 특징

- **EC2 기반:** NAT 역할을 수행하기 위해 Amazon EC2 인스턴스를 사용.
- **트래픽 처리:** 사설 서브넷에서 인터넷으로 나가는 아웃바운드 트래픽만 허용.
- **보안 그룹:** EC2 보안 그룹을 설정하여 네트워크 트래픽을 제어 가능.
- **유지보수 필요:** 사용자가 직접 인스턴스 크기, 소프트웨어 업데이트 및 복구를 관리해야 함.

### 장점

- **비용 효율적:** 작은 규모의 트래픽 처리에 적합.
- **사용자 정의 가능:** 고급 네트워크 설정 및 사용자 지정 소프트웨어 설치 가능.
- **다양한 선택지:** EC2 인스턴스 유형을 사용자가 선택 가능.

### 단점

- 관리 부담: 사용자가 직접 설정 및 유지보수를 관리해야 함.
- 고가용성 부족: EC2 인스턴스 장애 시 트래픽이 중단될 수 있음.
- 성능 제한: EC2 인스턴스의 크기에 따라 트래픽 처리량이 제한됨.

## 비교

특징	NAT Gateway	NAT Instance
관리 주체	AWS 관리형 서비스	사용자가 EC2 인스턴스를 관리해야 함
성능	자동 확장 (최대 45Gbps)	EC2 인스턴스 크기에 따라 성능 제한
가용성	AZ별 고가용성 지원	인스턴스 장애 시 사용자가 수동으로 복구 필요
설치 및 설정	간단한 설정만으로 사용 가능	EC2 인스턴스와 관련 리소스 수동 구성 필요
비용	데이터 처리량에 따라 비용 발생	EC2 인스턴스 유형에 따라 비용 결정
IPv6 지원 여부	지원하지 않음 (IPv6는 Egress-only Gateway 사용)	지원하지 않음
보안 그룹	NAT Gateway에 직접 보안 그룹을 적용할 수 없음	EC2 인스턴스에 보안 그룹을 적용 가능
유지보수	AWS가 자동으로 유지보수	사용자가 직접 인스턴스를 관리 및 유지보수 필요
다중 AZ 구성	각 AZ에 개별 NAT Gateway를 생성해야 함	다중 AZ 구성 시 여러 NAT 인스턴스를 수동 구성
사용 사례	고성능, 대규모 네트워크 트래픽 처리	저비용, 소규모 또는 임시 네트워크 환경

## AWS 관련 서비스

- 인터넷 게이트웨이(IGW): 인터넷으로 트래픽 라우팅.
- 라우팅 테이블: 사설 서브넷에서 NAT Instance로 트래픽을 라우팅하도록 설정 필요.

## 특이점

- EC2 인스턴스에 퍼블릭 IP 주소를 할당해야 함.
- 고가용성을 위해 다중 AZ 구성 시 별도의 NAT Instance를 추가 설정해야 함.
- IP 마스커레이딩(IP Masquerading)을 통해 사설 IP 주소를 공용 IP 주소로 변환.

## VPC Exam Essentials

## 시험 준비를 위한 주요 내용

### 1. VPC의 기본 구성 요소:

- CIDR 블록의 범위와 서브넷 크기 계산.
- 기본 VPC와 커스텀 VPC의 차이점.

### 2. 네트워크 주소 지정:

- IPv4와 IPv6의 차이점.
- 사설 IP 주소와 공용 IP 주소의 역할.

### 3. 보안 구성:

- 보안 그룹(Security Group)의 상태 유지(stateful) 특징.
- 네트워크 ACL(Network ACL)의 상태 비유지(stateless) 특징.
- 두 보안 기능의 차이와 사용 사례.

### 4. 라우팅 및 게이트웨이:

- 라우팅 테이블 설정 및 NAT Gateway를 통한 인터넷 액세스 구성.
- NAT Gateway와 NAT Instance의 차이점.

### 5. 고가용성 아키텍처 설계:

- NAT Gateway의 다중 AZ 구성 방법.
- VPC에 고가용성을 추가하는 방법.

### 6. 기본 VPC 환경:

- Default VPC의 특성과 사용 사례.
- Default VPC 삭제 및 복구 방법.

## 2. Additional VPC Feature

### Extending VPC Address Space

#### 주요 개념

##### 1. CIDR 블록 추가:

- AWS에서는 VPC에 최대 5개의 추가 CIDR 블록을 할당할 수 있음.
- IPv4 CIDR 블록은 /16에서 /28 까지 지원.
- 추가된 CIDR은 기존 CIDR과 중복되지 않아야 함.

##### 2. IPv6 지원:

- AWS에서 할당된 IPv6 CIDR 블록은 /56로 고정.
- IPv6는 전역적으로 고유하며, 프라이빗 네트워크와 퍼블릭 네트워크 모두에서 사용 가능.

##### 3. 사용 사례:

- 리소스 증가로 기존 서브넷의 IP 주소가 부족할 때 CIDR 확장 필요.

- 새로운 워크로드를 지원하기 위해 추가 서브넷을 구성해야 할 경우.

#### 4. 제약 조건:

- 추가된 CIDR 블록은 원래의 VPC CIDR과 동일한 RFC 1918 범위여야 함.
- CIDR 블록 변경은 실시간 트래픽에 영향을 미치지 않음.

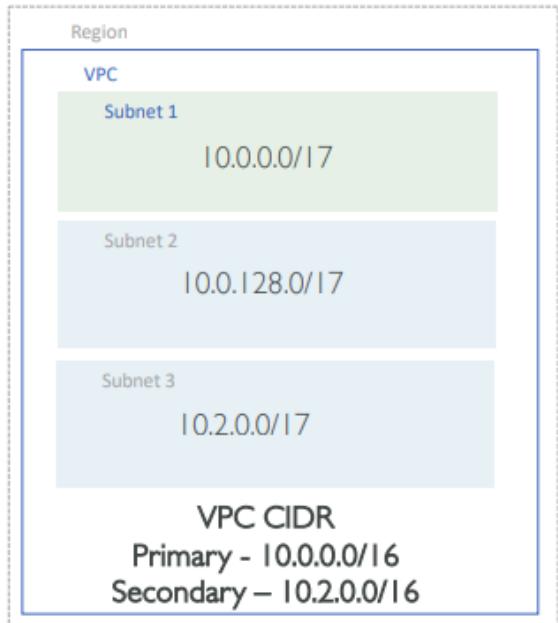
## 장점

- 기존 리소스를 중단하지 않고 VPC 주소 공간 확장 가능.
- 유연한 네트워크 설계를 통해 리소스 증가에 대응 가능.
- IPv6 CIDR 블록을 활용하여 향후 확장성 확보.

## 단점

- CIDR 충돌 방지 규칙을 수동으로 관리해야 함.
- 잘못된 CIDR 설계는 복잡성을 초래할 수 있음.

“ans/image/add\_vpc\_cidr.png” 을 찾지 못했습니다.



Main Route Table

Destination	Target
10.0.0.0/16	local
10.2.0.0/16	local

## ENI



## 요약

Elastic Network Interface (ENI)는 Amazon VPC의 가상 네트워크 카드로, EC2 인스턴스에 네트워크 연결을 제공하는 중요한 구성 요소입니다. ENI는 네트워크 트래픽을 유연하게 관리하고 고가용성을 지원하기 위한 다양한 기능을 제공합니다.

## 주요 개념

### 1. ENI의 정의 및 역할:

- ENI는 VPC 내부에서 IP 주소와 네트워크 연결을 관리하는 네트워크 어댑터.
- 인스턴스와 분리 가능하며, 한 ENI를 다른 EC2 인스턴스에 연결할 수 있음.

### 2. ENI의 주요 구성 요소:

- 기본 NIC:** EC2 인스턴스 생성 시 자동으로 생성되며 기본 NIC로 설정.
- 보안 그룹:** ENI 수준에서 보안 그룹을 설정하여 트래픽 제어.
- Private IP 및 Elastic IP:** ENI는 하나 이상의 프라이빗 IP를 가질 수 있으며, EIP와 연결 가능.

### 3. ENI의 활용 사례:

- 다중 네트워크 구성:** 단일 인스턴스에 여러 ENI를 추가하여 다중 네트워크 연결 구성.
- 고가용성 및 장애 복구:** 장애 시 ENI를 다른 인스턴스로 연결하여 빠른 복구 지원.
- 애플리케이션 분리:** 한 인스턴스에서 여러 네트워크 구성을 분리하여 운영.

### 4. 고급 사용 사례:

- ENI Trunking:** 고밀도 워크로드에서 다중 ENI를 사용하여 IP 주소 확장.
- ENI의 이동성:** ENI를 하나의 인스턴스에서 다른 인스턴스로 이동 가능.

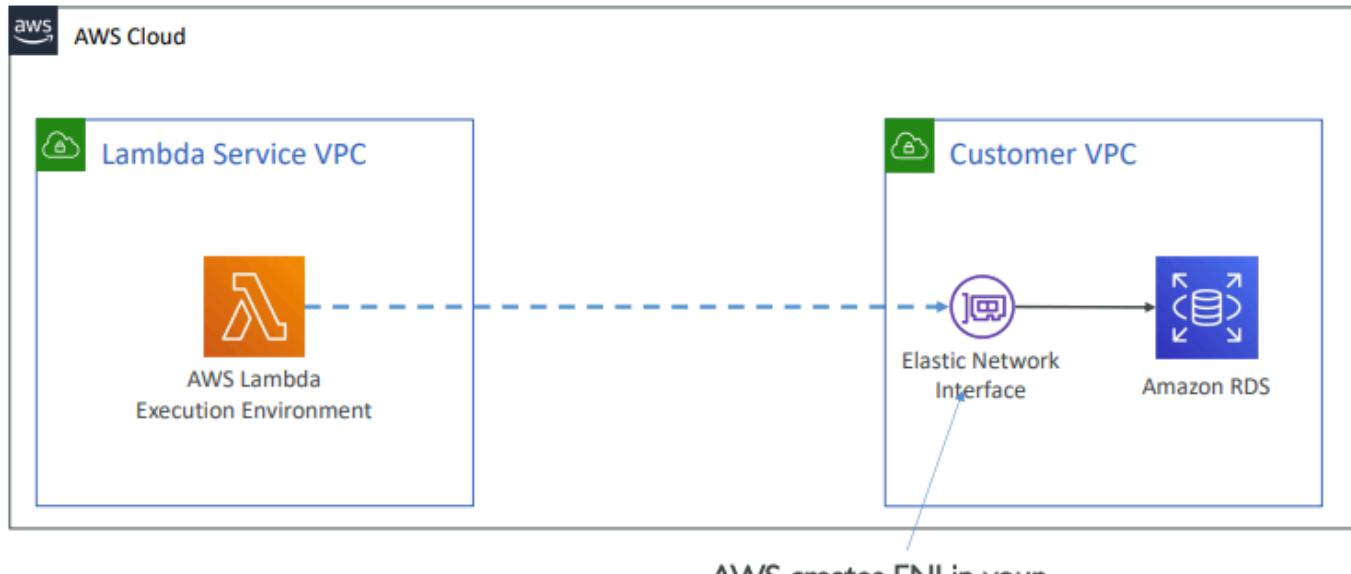
## 장점

- 네트워크 구성을 유연하게 변경할 수 있음.
- 다중 네트워크를 통해 분리된 트래픽을 처리 가능.
- ENI의 분리 및 이동성을 통해 장애 복구 및 유지보수에 용이.

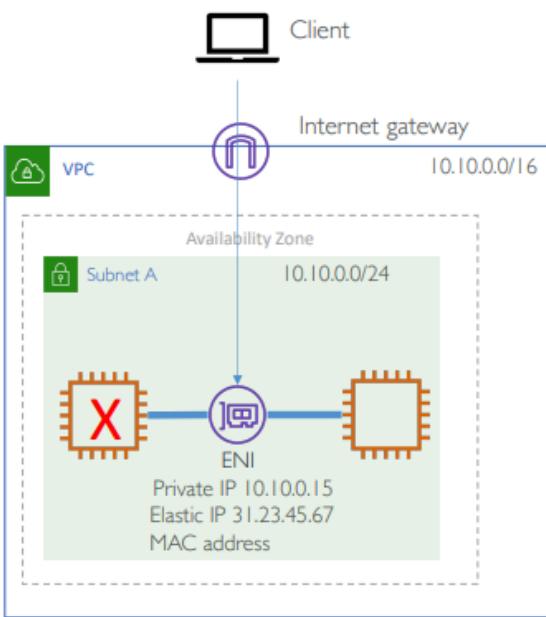
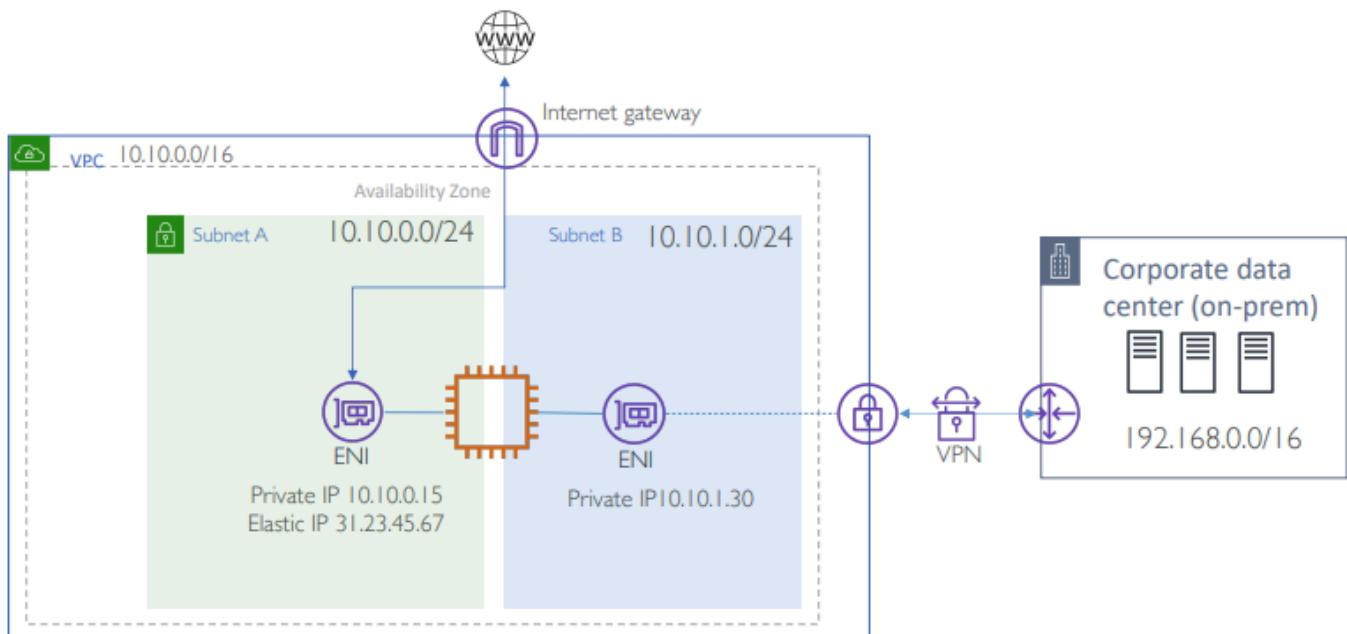
## 단점

- ENI의 설정과 관리는 사용자가 직접 수행해야 하므로 복잡성이 증가할 수 있음.
- 네트워크 처리량은 EC2 인스턴스 유형에 따라 제한될 수 있음.

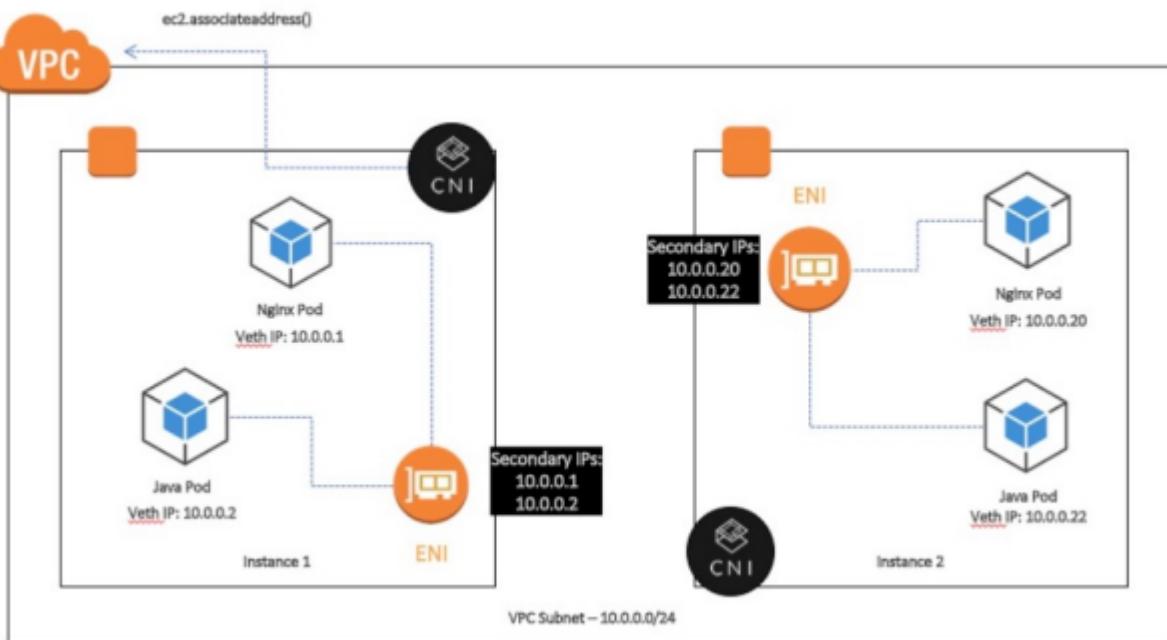
Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
a1.medium	2	4	4
a1.large	3	10	10
a1.xlarge	4	15	15
a1.2xlarge	4	15	15
a1.4xlarge	8	30	30
a1.metal	8	30	30
c1.medium	2	6	IPv6 not supported
c1.xlarge	4	15	IPv6 not supported
c3.large	3	10	10



AWS creates ENI in your  
VPC



- Attach the ENI to hot-standby instance in case primary instance fails
- No changes in routing or DNS configuration
- Brief loss of connectivity may be experienced



# Bring Your Own IP (BYOIP)

## 요약

**Bring Your Own IP (BYOIP)**는 고객이 소유한 IP 주소를 AWS로 가져와 AWS 서비스와 통합해 사용할 수 있게 해주는 기능입니다. 이 기능은 클라우드로의 전환 시 네트워크 변경을 최소화하고 고객이 기존 IP 주소를 유지할 수 있도록 돕습니다.

## 주요 개념

### 1. BYOIP의 정의:

- 고객이 소유한 IPv4 또는 IPv6 주소를 AWS로 이전하여 Amazon VPC 및 AWS Global Accelerator와 같은 서비스에서 사용 가능.
- IP 주소는 AWS 리전에 매핑되어 AWS 네트워크에서 활용.

### 2. 주요 사용 사례:

- 기존 IP 주소를 사용하는 고객들에게 클라우드 전환 간소화.
- 기업이 자체 IP 주소를 유지하여 DNS 변경 없이 마이그레이션 가능.
- 규제 요건 충족을 위해 특정 IP 주소 유지.

### 3. 프로세스:

- 검증: 고객의 IP 주소 소유권을 AWS에 검증.
- 광고: AWS의 네트워크에서 IP 주소가 광고되도록 설정.
- 사용: AWS 서비스에서 IP 주소를 사용(예: Elastic Load Balancer, EC2).

### 4. BYOIP의 주요 제약 조건:

- AWS가 고객의 IP 소유권을 확인하기 위한 검증 절차 필요.
- IPv4는 AWS 리전에서 BYOIP를 지원하며, IPv6은 글로벌 네트워크에서 지원.
- IP 주소가 AWS 외부에서 사용 중인 경우 일부 제한 가능.

## 장점

- 클라우드로 전환 시 기존 네트워크 설정을 유지 가능.
- IP 주소의 소유권과 유연성을 유지하며 AWS 리소스를 활용.
- 네트워크 구성의 변경이 최소화되어 다운타임 감소.

## 단점

- BYOIP 설정 및 소유권 검증에는 시간이 걸릴 수 있음.

- AWS 환경 외부에서 사용하는 IP와의 통합은 추가 작업이 필요할 수 있음.

## 3. VPC DNS and DHCP

### Amazon VPC DNS 옵션

#### 요약:

- **Amazon VPC의 기본 DNS 기능:**
  - Amazon VPC는 기본적으로 DNS 해석 및 호스트 이름 생성 기능을 제공합니다.
  - 퍼블릭 및 프라이빗 DNS 옵션을 활성화하여 클라우드 환경에서 네트워크 자원을 효과적으로 관리.
- **DNS 호스트 이름 및 DNS 해석기:**
  - VPC 내에서 EC2 인스턴스에 자동으로 DNS 호스트 이름을 할당.
  - 기본적으로 `ec2.internal` 형식의 이름을 제공하며, 퍼블릭 IP가 활성화된 경우 퍼블릭 DNS 이름도 제공.
- **DNS 해석 옵션:**
  - 퍼블릭 DNS 해석을 허용하거나, 프라이빗 DNS만 허용하는 설정 가능.
  - AWS에서는 Route 53 Resolver를 통해 외부 DNS 쿼리를 처리할 수 있음.

#### 장점:

- 사용자가 별도로 관리할 필요 없이 자동으로 DNS가 할당.
- 프라이빗 DNS를 활용해 보안이 강화된 네트워크 환경 제공.

#### 단점:

- 기본 DNS 옵션은 커스텀 도메인 요구 사항에 적합하지 않을 수 있음.
- 퍼블릭 DNS를 활성화하면 잠재적인 보안 문제가 발생할 가능성.

### AWS에서 관련 서비스:

- **Route 53 Resolver:** VPC와의 통합을 통해 DNS 쿼리를 처리.
- **DHCP 옵션 세트:** VPC 내 DNS 서버 주소를 동적으로 구성.

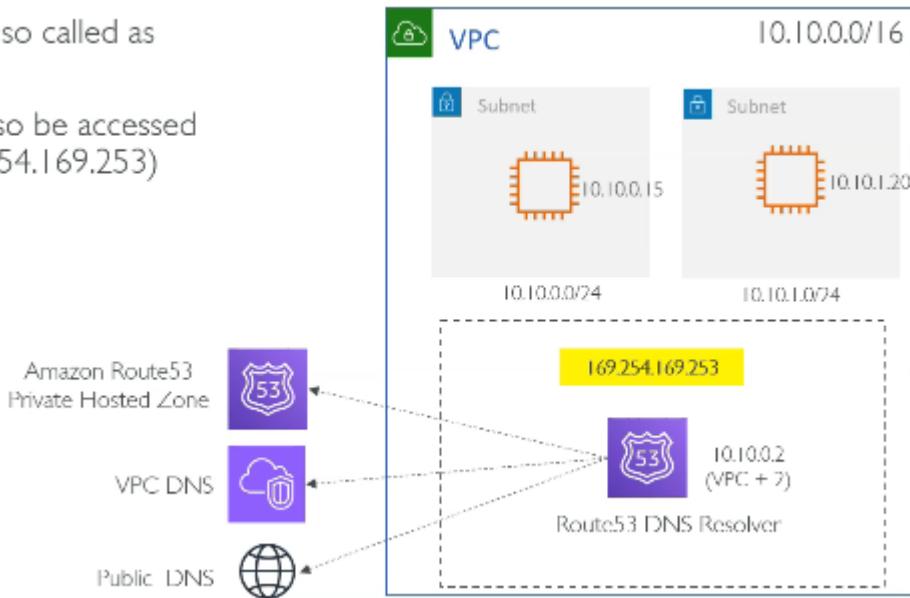
#### 주의사항:

- 퍼블릭 DNS 이름을 활성화하려면 VPC와 서브넷의 인터넷 게이트웨이가 올바르게 구성되어 있어야 함.
- 프라이빗 DNS를 사용할 때는 자체 DNS 서버를 설정하거나 AWS 제공 DNS를 사용해야 함.

# DNS VPC DNS Server (Route 53 Resolver)

Iso called as

so be accessed  
54.169.253)



## Route 53 Resolver와의 통합

### 요약:

- **Route 53 Resolver란?**
  - Amazon Route 53 Resolver는 VPC 내에서 DNS 쿼리를 처리하고 온프레미스 네트워크 와의 통합을 지원하는 DNS 해석기입니다.
  - VPC와 온프레미스 네트워크 간 DNS 쿼리를 원활히 전송할 수 있도록 인바운드 및 아웃바운드 엔드포인트를 제공.
- **주요 기능:**
  - **인바운드 엔드포인트:**
    - 온프레미스 DNS 서버에서 AWS VPC로 DNS 쿼리를 전달.
  - **아웃바운드 엔드포인트:**
    - VPC 내 DNS 쿼리를 온프레미스 DNS 서버로 전달.
- **활용 사례:**
  - 온프레미스 네트워크와 AWS 간의 하이브리드 네트워크에서 통합된 DNS 관리.
  - 여러 VPC와 연결된 복잡한 네트워크 아키텍처에서 DNS 쿼리를 중앙 관리.

### 장점:

- 하이브리드 네트워크에서 DNS 쿼리를 쉽게 통합 가능.
- VPC와 온프레미스 네트워크 간 상호작용을 단순화.
- 기존 DNS 서버를 유지하면서 AWS 네트워크를 통합할 수 있음.

## 단점:

- 엔드포인트 설정 및 운영에 대한 추가 비용 발생.
- 잘못된 설정으로 인해 DNS 쿼리 흐름에 문제가 발생할 수 있음.

## AWS에서 관련 서비스:

- **Route 53 Resolver:** 기본 제공 DNS 해석 및 인바운드/아웃바운드 엔드포인트.
- **VPC Peering:** 연결된 VPC 간 DNS 쿼리 처리.
- **Transit Gateway:** 여러 VPC와 온프레미스 네트워크 간 트래픽 관리.

## 주의사항:

- 인바운드와 아웃바운드 엔드포인트는 보안 그룹을 통해 접근을 제어해야 함.
- VPC와 온프레미스 간 IP 주소 충돌이 없는지 확인 필수.
- 온프레미스 네트워크와 VPC 간 경로 설정이 정확해야 함.

## 시험에서 자주 등장하는 패턴:

1. 하이브리드 네트워크에서 DNS 엔드포인트 구성 문제.
2. DNS 쿼리를 인바운드 및 아웃바운드 엔드포인트로 라우팅하는 시나리오.
3. Route 53 Resolver의 동작 원리를 묻는 문제.

## ### DHCP 옵션 세트

### 요약:

- **DHCP 옵션 세트란?**
  - DHCP 옵션 세트를 통해 Amazon VPC에서 동적으로 IP 주소 및 네트워크 설정을 제공 합니다.
  - 기본적으로 각 VPC는 하나의 DHCP 옵션 세트와 연결되며, 여러 네트워크 매개변수 (예: DNS 서버, 도메인 이름 등)를 정의할 수 있음.
- **구성 가능한 옵션:**
  1. **Domain Name Servers (DNS):**
    - 네트워크의 기본 및 보조 DNS 서버를 지정.
  2. **Domain Name:**
    - 네트워크의 기본 도메인 이름을 설정.
  3. **NTP Servers:**
    - 네트워크 시간 동기화를 위한 NTP 서버 주소를 정의.
  4. **NetBIOS 옵션:**

- Windows 기반 네트워크를 위한 NetBIOS 서버 및 관련 옵션을 설정.
- **활용 사례:**
  - VPC에서 커스텀 DNS 서버 사용.
  - 특정 NTP 서버로 시간 동기화.
  - 다양한 환경에 맞게 네트워크 매개변수를 동적으로 설정.

## 장점:

- VPC 내 인스턴스가 네트워크 설정을 자동으로 수신.
- 다양한 네트워크 환경에서 쉽게 설정 변경 가능.
- 네트워크 관리 자동화로 설정 오류 최소화.

## 단점:

- 모든 매개변수가 모든 네트워크 환경에 적용되지는 않음.
- 잘못된 옵션 설정 시 네트워크 접속 문제 발생 가능.

## AWS에서 관련 서비스:

- **Amazon Route 53:** 커스텀 DNS 서버 설정 시 활용.
- **VPC Subnet:** 서브넷별로 DHCP 옵션 세트를 관리.

## 주의사항:

- 기본 DHCP 옵션 세트는 삭제할 수 없으며, 필요한 경우 새 옵션 세트를 생성하여 대체.
- DHCP 옵션 세트는 VPC 단위로만 적용 가능(서브넷 단위는 불가).
- 커스텀 DNS 서버를 지정할 경우, 해당 서버가 가용한 상태인지 확인 필요.

## 시험에서 자주 등장하는 패턴:

1. DHCP 옵션 세트를 사용하여 VPC 내 네트워크 설정 변경 문제.
2. 커스텀 DNS 서버를 사용한 구성 시나리오.
3. DHCP 옵션 세트와 기본 VPC 네트워크 매개변수의 관계를 묻는 질문.

## 4. Network Performance and Optimization

### 기초용어

- **Bandwidth :** 최대 통신 가능 속도 (ex: 10mb/s, 10GB/s)
- **Latency :** 지연시간
  - 단순 지연이 아닌 신호에 의한 지연일 수도 있음. 또는 연결점이 많아서 지연이 발생할

수 있음.

- **Jitter** : 패킷간 지연이 변동된다.
- **Throughput** : 성공적 트래픽 처리량 (ex: 10mb/s)
  - Bandwidth가 1GB/s라 해서 무조건 처리량이 1GB/s가 될 수 없으며, Latency, Jitter 등 여러가지 요소에 따라 처리량에 영향이 있을 수 있음
- **Packet per Seconds (PPS)** : 초당 패킷 처리량

## Basics of Network performance - Bandwidth, Latency, Jitter, Throughput, PPS, MTU

- MTU : 일반적으로 1500Bytes (인터넷 표준)
- Jumbo Frame : MTU보다 더 큰 네트워크 패킷을 보낼 수 있음. (최대 9000bytes)
  - 장점
    - 패킷 수 감소: 점보 프레임은 표준 이더넷 프레임(1500바이트)보다 큰 패킷(최대 9000바이트)을 전송하므로, 동일한 데이터 양을 전송할 때 패킷 수가 줄어듭니다.
    - CPU 및 네트워크 부하 감소: 패킷 수가 줄어들면, 패킷당 발생하는 헤더 처리, 인터럽트, 복사 작업이 감소해 CPU와 네트워크 장치의 부하가 줄어듭니다.
    - Throughput(처리량) 증가: 패킷 수가 감소하므로, 패킷당 오버헤드가 줄어들어 네트워크의 처리량이 증가합니다. 특히 PPS(Packet Per Second) 제한이 있는 환경에서 효과적입니다.
  - 단점
    - 호환성 문제: 모든 네트워크 장치가 점보 프레임을 지원하는 것은 아닙니다. 경로상의 장치가 점보 프레임을 지원하지 않으면 MTU(Maximum Transmission Unit) 불일치로 인해 패킷이 드롭되거나 분할됩니다.
    - 프래그멘테이션 및 재조립 비용: MTU 불일치로 인해 패킷이 쪼개지면 성능 저하가 발생합니다. 재조립 과정에서 CPU 부하가 증가할 수 있습니다.
    - 메모리 사용량 증가: 큰 프레임을 처리하려면 네트워크 카드와 드라이버가 더 많은 메모리를 필요로 합니다.
    - 에러 재전송 비용 증가: 점보 프레임에서 에러가 발생하면 큰 데이터 블록을 다시 전송해야 하므로, 재전송 비용이 증가합니다.
  - Use Case
    - Instance A -> Route 1 -> Route 2 -> Instance B
    - Instance A -(MTU 1500 DF=1)> Route 1 (OK) -(MTU 1500)> Route 2 (Not Accept 1500, Change 1000 with ICMP)
  - **MTU Path Discovery**에선 ICMP가 반드시 열려 있어야 한다.
  - AWS에선 9001 MTU를 지원하며 VPC 레벨에서 기본으로 동작한다.
  - 단, IGW, VPC Peering에선 Jumbo Frames를 지원하지 않는다. (MTU 1500)
  - DX에선 9001 MTU 지원함.

- EC2 Cluster placement Group은 물리적으로 붙어 있어서 점보 프레임을 지원하고 최대 네트워크 Throughput을 제공한다.
- 점보 프레임은 VPC를 떠나는 트래픽에 대해서 주의해서 사용해야 함. 패킷이 1500바이트를 초과하면 조각화되거나 IP 헤더에 조각화 안 함 플래그가 설정되어 있으면 삭제됨.
- 점보 프레임은 ENI에서 정의됨. (tracepath amazon.com)
- MTU 확인 하려면 **ip link show {eth name}**
- MTU 설정을 수동으로 하려면 \*\*sudo ip link set dev {eth name} mtu {byte number}\*\*
- tracepath시 ec2에 public ip로 호출할 때와 private ip로 호출할 때 MTU에 대한 이슈로 네트워크 성능차가 발생할 수 있음.
- AWS 내 MTU
  - VPC 레벨 : Jumbo Frames 지원 (9001 MTU)
  - VPC Endpoint : MTU 8500 byte 지원
  - IGW를 벗어나면 1500 MTU로 감소됨
  - VPC Peering이라고 하더라도 같은 리전이면 MTU 9001 지원
  - VPC Peering이 다른 리전이면 MTU 1500
- On-Premise
  - VGW를 통한 VPN : MTU 1500
  - TGW를 쓰는 STS VPN : MTU 1500
  - DX : 9001 MTU 지원
  - DX via TGW : MTU 8500 for VPC Attachment Over Direct Connect

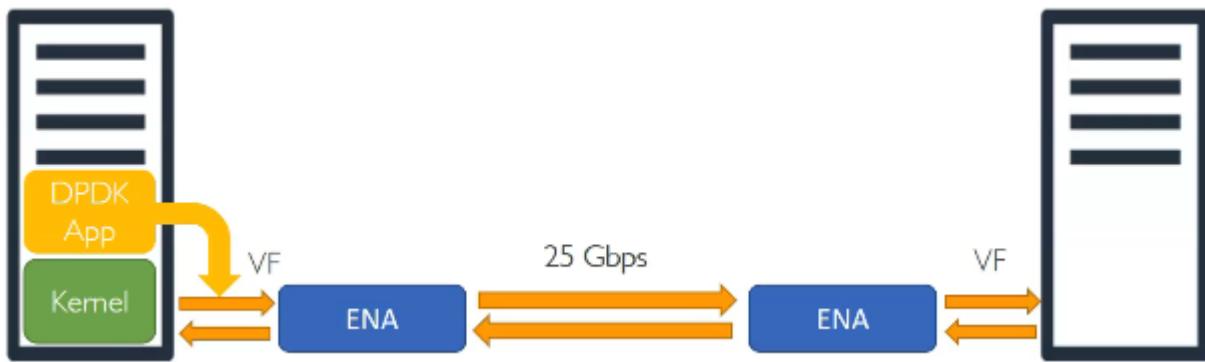
## ENA

- 100만 이상의 PPS Performance
- 인스턴스간 Latency를 줄이는 기술
- 하이퍼바이저 out과 일관성 있는 성능을 위해 SR-IOV with PCI passthrough 적용
- Intel ixgbevf driver 또는 Elastic Network Adapter(ENA)를 사용
- SR-IOV with PCI passthrough는 가상화 수단이며 이 기술을 통해 한 개의 NIC에 여러 개의 네트워크가 설정된 것처럼 가상함수를 적용해서 여러 개의 가상 NIC를 만들 수 있음.
- PCI가 ENI와 같은 역할을 하여 PCI상에서 여러 개의 서버에 통신을 가상의 직렬형태로 연결 할 수 있다.
- 조합시 대기시간이 최소화 되고 데이터 전송 속도가 증가한다.
- 전제 조건
  - 인스턴스 형식에 따라 다름
    - ixgbevf driver를 지원해야 함 (지원 시 10Gbps 까지)
    - Elastic Network Adapter (ENA)를 지원해야 함 (지원 시 100Gbps 까지)
  - 두 가지 모두 충족해야 함
  - 지원 타입

- 100Gbps: ENA 지원
  - A1, C5, C5a, C5d, C6g, F1, G3, G4, H1, I3, I3en, etc
- 10Gbps: Intel 82599 Virtual Function 지원
  - C3, C4, D2, M4 (m4.16xlarge는 예외), R3, etc
- EC2 Cluster Placement Group에 포함되어야 사용 가능한 옵션
- Cluster Placement Group이 아니라면 5Gbps로 내려감.

## DPDK and Elastic Fabric Adapter (EFA)

- DPDK : Intel Data Plane Development Kit
  - SR-IOV와 향상된 네트워킹이 인스턴스와 하이퍼바이저간에 패킷 처리 부하를 감소 시킨다.
  - DPDK는 OS 내에서 벌어지는 packet 처리 부하를 감소시킨다.

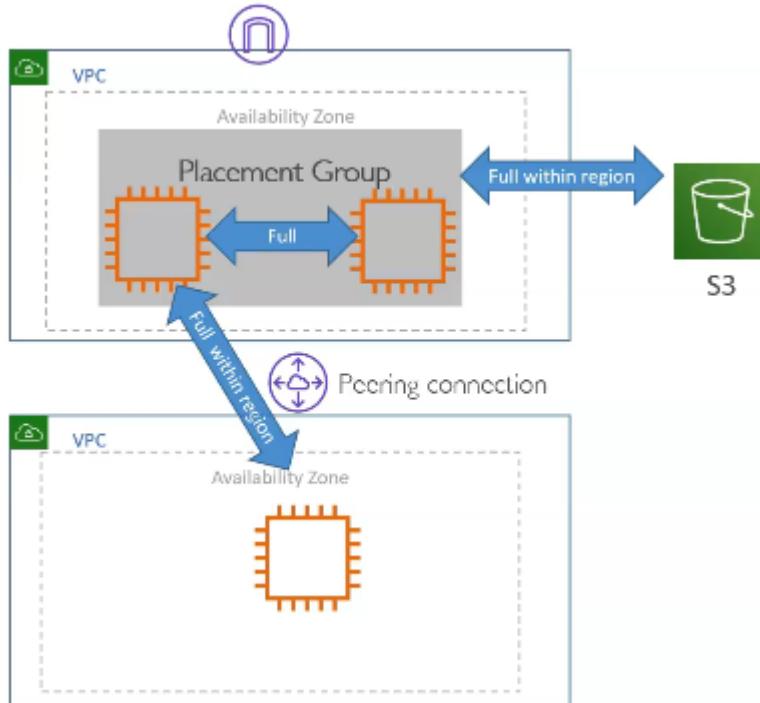


- EFA : ENA에 추가 되어 사용된다.
  - 낮은 Latency와 높은 throughput
  - Linux OS 전용
  - **Windows는 ENA에서 수행한다.**
  - MPI : Message Passing Interface (모든 노드 사이에서 병렬 통신이 가능하게 하여 네트워크 처리량 증가하게 한다)
  - c5n.18xlarge, p3dn.24xlarge

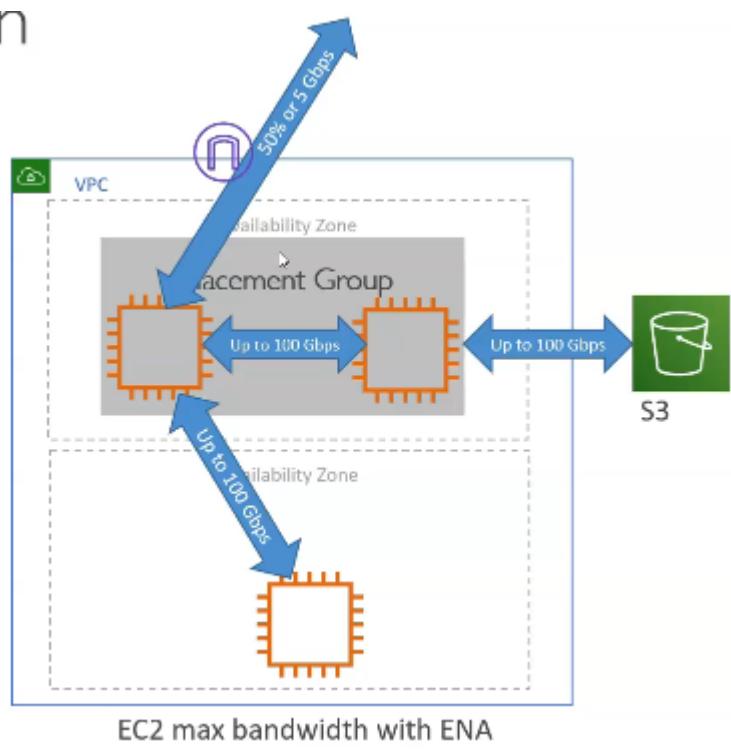
## Bandwidth Limits inside and outside of VPC

- NAT Gateway는 현재 최대 45Gbps까지 지원하고 있음
- 더 크게 사용한다면 멀티 NAT Gateway를 사용해야 하며 존을 넘지 않도록 할 것. 존을 넘어가면 추가 데이터 전송요금이 발생하게 됨.
- EC2의 경우 여러 요소에 의해 인스턴스 자체 대역폭을 결정하게 됨.
  - 최소
    - 인스턴스 패밀리
    - vCPU 크기
    - 트래픽 목적지

- 네트워크 최적화 여부
- Nitro 기반 인스턴스 여부
- **가급적이면 인스턴스를 최신세대로 정하라.**
- 인스턴스에서 네트워킹 대역폭도 설정할 수 있음.

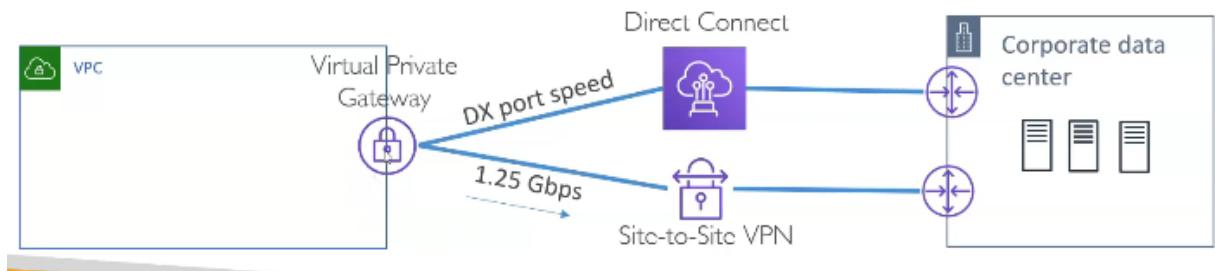


- 인터넷 또는 DX로 연결되는 경우 EC2가 제공하는 네트워크 대역폭의 50%만 사용 가능
- 인스턴스 vCPU가 32보다 작으면 5GBps로 제한됨.
- 트래픽이 igw 또는 S3 또는 다른 존으로 이동하면 50% 또는 5GBps로 대역폭이 내려간다.
- **최대**
  - Intel Virtual Function을 쓰면 최대 대역폭 10GBps를 사용 가능.
  - EC2 Cluster Placement 그룹을 통해 ENA를 사용한다면 최대 대역폭 100GBps를 사용 가능



EC2 max bandwidth with ENA

- S3와 데이터 통신을 빠르게 하기 위해 S3 Endpoint를 쓰게 되면 속도를 높일 수 있다.
- VPN and DX Bandwidth
  - VPG의 경우 총 대역폭은 25GBps이며 여러 VPN연결을 가질 수 있음.
  - 여러개와 상관없이 VPG의 총 대역폭은 25GBps임.
  - 만일 TGW라면 ECMP를 활성화하고 대역폭을 두배로 올릴 수 있음.
  - 직접 연결의 경우 포트 스피드에 의해 대역폭이 결정됨
  - **DX의 경우 1, 10, 100GBps와 50, 100, 500MBps 같은 하위 속도 옵션도 제공.**
  - TGW의 경우 최대 대역폭이 50Gbps.



## Network I/O Credit

- Network 또한 EC2 처럼 credit 시스템을 가지고 운영됨.
- R4, C5 패밀리는 network I/O Credit Mechanism으로 동작하고 있음.
- 특히 벤치마크 상황등에 해당 상황이 발생될 수 있으므로 속도가 느려지면 요런 요소도 고려 할 것.

## Summary

- 빠른 네트워크 대역폭을 처리하기 위해서 아래의 옵션을 고려할 것.
  - Jumbo Frame
  - EC2 Enhanced Networking
  - Placement groups
  - EBS Optimized Instance
  - DPDK
- Instance 레벨에서의 네트워크 최적화는
  - Enhance Networking (SR-IOV, ENA, Intel VF 82599)
  - Placement Groups
  - EBS Optimized Instance
- OS 레벨에서의 네트워크 최적화는 DPDK
- EFA는 HPC상에서 향상된 Network 성능을 위해 제공하는 OS-Bypass해주는 ENA의 추가 기능

## Exam Essential

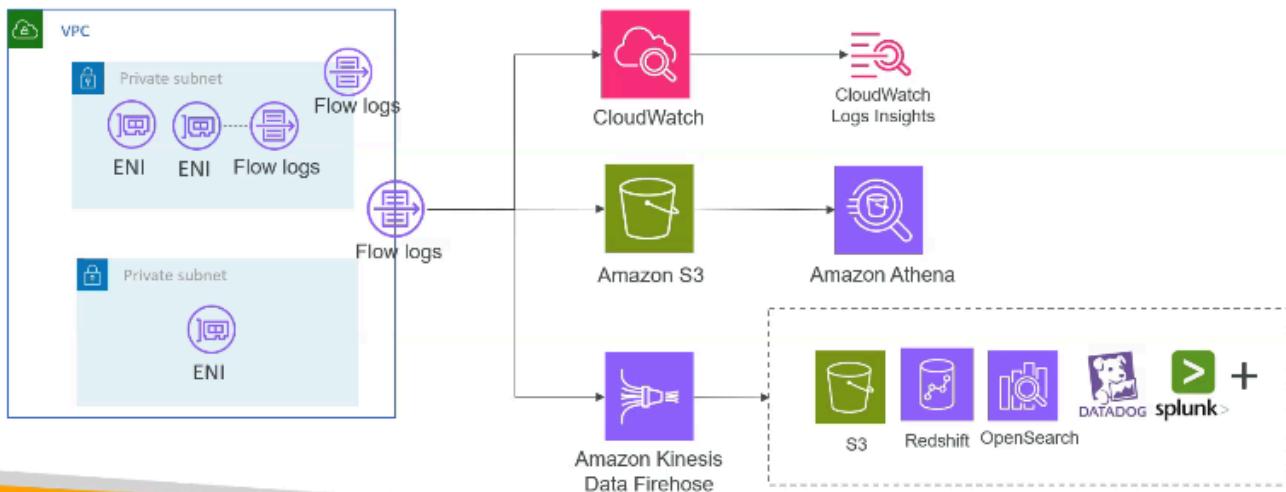
- VPC 내에서 MTU 사이즈는 **점보 프레임**을 통해 9001 byte까지 지원한다.
- **MTU가 1500 Byte만 될때는**
  - igw를 타는 경우
  - vpc-peering이 외부 리전인 경우
  - VPN 커넥션과 연결되는 경우
- 만일 PPS가 병목에 걸려 처리량이 떨어지는 경우 MTU를 늘려서 병목을 없애야 한다.
- 네트워크 속도 향상을 위해선 **Enhanced Networking** 또는 **Placement Groups**를 사용할 수 있음
  - **Enhanced Networking**은 Instance <-> Hypervisor간 대기 시간을 낮춰주는 것.
  - **DPDK**를 통해서 **OS레벨에서의 패킷 프로세싱**을 더 빠르게 처리한다.
  - 인스턴스 패밀리별로 별도의 대역폭을 가지고 있음.
    - vCPU, Network Throughput, Disk I/O, Enhanced Networking 지원 여부까지도.
- 대역폭은 **다중 플로우에 걸쳐 집계된 대역폭**이다.
  - 리전 내
    - EC2의 최대 bandwidth 사용 가능
    - 리전 내에서 EC2 인스턴스 간 또는 EC2 <-> S3에 대해서 최대 100GBps까지 사용 가능 (Multi Flow)
  - 리전 밖 (igw, DX)
    - EC2의 최대 대역폭에서 50%가 떨어짐.
    - 또한 인스턴스의 vCPU가 32가 위여야 50%만 떨어지고, 32 밑이라면 50%가 추가적으로 더 떨어짐.
    - ex> 최대 20 -> igw (10) -> c5.xlarge (4vcpu) -> total 5GBps

- 같은 존간의 대역폭이 5Gbps일때 Placement Groups에 배치하면 인스턴스간 10Gbps까지 네트워크 속도를 올릴 수 있다. 집계 대역폭은 100Gbps로 제한 (ENA)
- Endpoint와 EC2 인스턴스간에 얻을 수 있는 최대도 100Gbps (ENA)

## VPC Traffic Monitoring, Trouble Shooting & Analysis

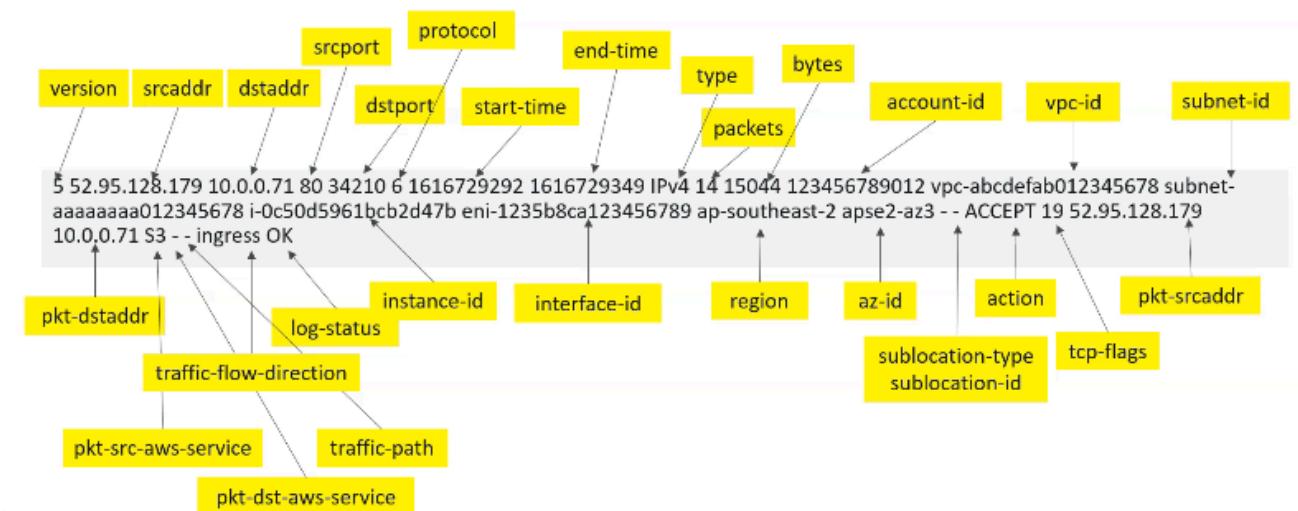
### VPC Flow Logs

- ENI의 내외부 트래픽 캡쳐
  - VPC, Subnet, ENI Level에서 감시
  - 모니터 및 Troubleshooting 목적
  - S3, Cloudwatch Logs, Kinesis Data firehose로도 보낼 수 있음
- 관리형 서비스에서 오는 네트워크 정보도 취합함.
  - ELB, RDS, ElastiCache, RedShift, Amazon Workspace, NAT Gateway, TGW 등등



- flow log version : 기본 2, 최신 5
- Flow Log는 사용자 요구에 따라 커스텀처리 할 수 있음

### Custom flow logs example



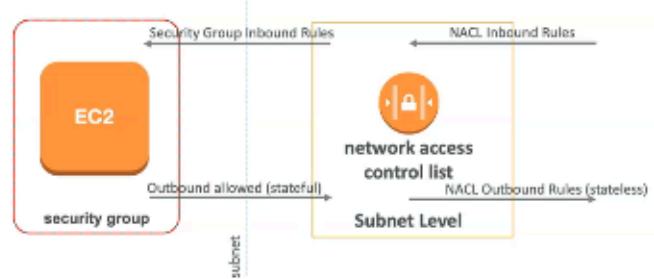
- 트러블 슈팅 시나리오

#### "ACTION" Field

##### For incoming requests

Inbound REJECT: NACL or SG

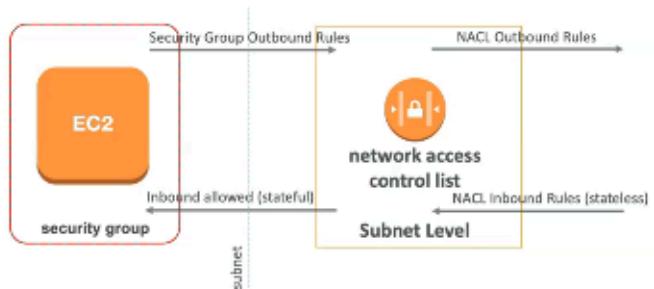
Inbound ACCEPT, outbound REJECT: NACL



##### For outgoing requests

Outbound REJECT: NACL or SG

Outbound ACCEPT, inbound REJECT: NACL



- VPC Flow Log limit
  - DNS 서버 로그는 저장되지 않음
  - EC2 Meta Data, Time Sync, DHCP, Windows Activation, Mirroring Traffic
- 예전 툴
  - wireshark (tcpdump)
  - traceroute
  - telnet
  - nslookup : host name resolve
  - ping

## VPC Traffic Mirroring

### 개요

- VPC Traffic Mirroring:** AWS VPC 내에서 네트워크 트래픽을 캡처하고 복제해 모니터링 및 보안 분석에 활용하는 기능.
- 대상:** ENI(Elastic Network Interface)에 대한 트래픽을 복제.
- 주요 활용 사례:**
  - 보안 위협 탐지 및 침입 분석
  - 네트워크 성능 모니터링 및 트러블슈팅
  - 규정 준수 및 감사

---

### 장점

- **비침투적 트래픽 캡처**: 애플리케이션 성능에 영향 없이 트래픽을 모니터링.
  - **세분화된 트래픽 복제**: 특정 트래픽 필터링 가능(예: 포트, 프로토콜, IP).
  - **네이티브 AWS 서비스**: 추가적인 하드웨어 구성 없이 AWS 내에서 직접 실행.
  - **보안 강화**: 실시간으로 네트워크 침입 감지 및 보안 분석 가능.
  - **확장성**: 필요에 따라 트래픽 복제 대상을 쉽게 확장.
- 

## 단점

- **비용**: 트래픽 양에 따라 비용이 증가.
  - **복잡성**: 초기 설정 및 필터 구성에 대한 이해도가 필요.
  - **대상 제한**: ENI 단위로만 트래픽을 복제 가능.
  - **성능 부담**: 대규모 트래픽 복제 시 일부 성능 저하 가능.
- 

## 구동 방법

1. **Traffic Mirror Target 생성**: 트래픽을 전송할 대상(예: ENI, NLB, EC2 인스턴스 등) 생성.
  2. **Traffic Mirror Filter 설정**: 복제할 트래픽의 조건(IP, 포트, 프로토콜 등)을 정의.
  3. **Traffic Mirror Session 생성**:
    - 복제 대상 ENI 선택
    - 트래픽 필터 및 타겟 연결
    - 우선순위 설정
  4. **트래픽 분석**: 복제된 트래픽을 보안 도구(Splunk, Wireshark 등)에서 분석.
- 

## 추가 팁

- 비용 절감을 위해 특정 포트 및 IP 범위만 복제하도록 필터링 설정 권장.
- NIDS(Network Intrusion Detection System)와 연동해 보안 강화 가능.

## VPC Traffic Mirroring – 알아두면 좋은 사항

- VPC 내부 및 외부로 흐르는 트래픽을 미러링하여 네트워크 분석 도구로 전송하고, 잠재적인 네트워크 및 보안 이상 징후를 감지할 수 있도록 함.
- 미러 소스는 ENI(Elastic Network Interface).
- 미러 타겟은 다른 ENI 또는 NLB(Network Load Balancer, UDP 포트 4789)일 수 있음.

- 미러 필터 – 필요한 트래픽만 캡처하도록 설정 가능.
  - 프로토콜, 소스/목적지 포트 범위, CIDR 블록을 지정해 트래픽 필터링.
  - 번호가 매겨진 규칙을 정의하여 해당 트래픽을 적절한 대상에 전송.
- 트래픽 미러 소스와 미러 타겟(모니터링 장치)은 동일한 VPC에 있거나,
  - 동일 리전의 VPC 피어링 또는 tgw를 통해 연결된 다른 VPC에 위치할 수 있음.
- 소스와 목적지는 서로 다른 AWS 계정에 있을 수 있음.

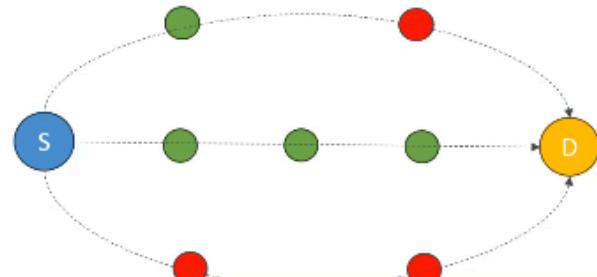
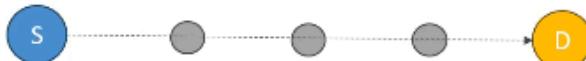
## VPC Features for Network Analysis



Reachability Analyzer



Network Access Analyzer



항목	Reachability Analyzer	Network Access Analyzer
목적	특정 소스에서 대상까지 네트워크 경로의 연결 여부 분석	네트워크 경로에서 잠재적인 과도한 접근 권한을 분석 및 식별
작동 방식	소스에서 대상까지 경로 추적 및 연결 상태 확인	다양한 경로를 분석하여 과도한 네트워크 접근을 시각화
주요 기능	<ul style="list-style-type: none"> <li>- 특정 경로에 대한 도달 가능성 확인</li> <li>- 네트워크 구성 문제 식별</li> </ul>	<ul style="list-style-type: none"> <li>- 접근 가능한 경로 분석 및 평가</li> <li>- 보안 정책 위반 가능성 감지</li> </ul>
장점	<ul style="list-style-type: none"> <li>- 직관적인 네트워크 경로 시각화</li> <li>- 특정 경로에 대한 세부 분석</li> </ul>	<ul style="list-style-type: none"> <li>- 광범위한 네트워크 분석 가능</li> <li>- 보안 및 접근 권한을 통합적으로 검토</li> </ul>
단점	<ul style="list-style-type: none"> <li>- 특정 소스/대상 경로에 한정</li> <li>- 네트워크 접근 범위 분석은 불가능</li> </ul>	<ul style="list-style-type: none"> <li>- 과도한 경로 식별 시 분석 시간이 길어질 수 있음</li> <li>- 네트워크 구성 변경 사항 추적 어려움</li> </ul>
사용 사례	<ul style="list-style-type: none"> <li>- 두 인스턴스 간 통신 문제 해결</li> <li>- 특정 인스턴스가 접근 불가능한 경우 경로 추적</li> </ul>	<ul style="list-style-type: none"> <li>- 과도한 권한 부여된 네트워크 구성 감지</li> <li>- 대규모 네트워크 접근 정책 감사 및 리포트</li> </ul>

항목	Reachability Analyzer	Network Access Analyzer
	- 방화벽 규칙 및 보안 그룹 구성 검토	- 네트워크 보안 태세 개선 및 규정 준수 확인
탐지 가능한 리소스	<ul style="list-style-type: none"> <li>- ENI(Elastic Network Interface)</li> </ul>	<ul style="list-style-type: none"> <li>- ENI(Elastic Network Interface)</li> </ul>
	<ul style="list-style-type: none"> <li>- EC2 인스턴스</li> </ul>	<ul style="list-style-type: none"> <li>- EC2 인스턴스</li> </ul>
	<ul style="list-style-type: none"> <li>- 인터넷 게이트웨이</li> </ul>	<ul style="list-style-type: none"> <li>- 인터넷 게이트웨이</li> </ul>
	<ul style="list-style-type: none"> <li>- NAT 게이트웨이</li> </ul>	<ul style="list-style-type: none"> <li>- NAT 게이트웨이</li> </ul>
	<ul style="list-style-type: none"> <li>- Transit Gateway</li> </ul>	<ul style="list-style-type: none"> <li>- Transit Gateway</li> </ul>
	<ul style="list-style-type: none"> <li>- VPC 피어링</li> </ul>	<ul style="list-style-type: none"> <li>- VPC 피어링</li> </ul>
	<ul style="list-style-type: none"> <li>- 보안 그룹(Security Group)</li> </ul>	<ul style="list-style-type: none"> <li>- 보안 그룹(Security Group)</li> </ul>
	<ul style="list-style-type: none"> <li>- 네트워크 ACL</li> </ul>	<ul style="list-style-type: none"> <li>- 네트워크 ACL</li> </ul>
	<ul style="list-style-type: none"> <li>- AWS PrivateLink</li> </ul>	<ul style="list-style-type: none"> <li>- AWS PrivateLink</li> </ul>
	<ul style="list-style-type: none"> <li>- 로드 밸런서(ALB/NLB)</li> </ul>	<ul style="list-style-type: none"> <li>- 로드 밸런서(ALB/NLB)</li> </ul>
		<ul style="list-style-type: none"> <li>- S3 버킷 정책</li> </ul>
		<ul style="list-style-type: none"> <li>- Lambda 함수 정책</li> </ul>
		<ul style="list-style-type: none"> <li>- IAM 정책</li> </ul>

## 5. VPC Peering

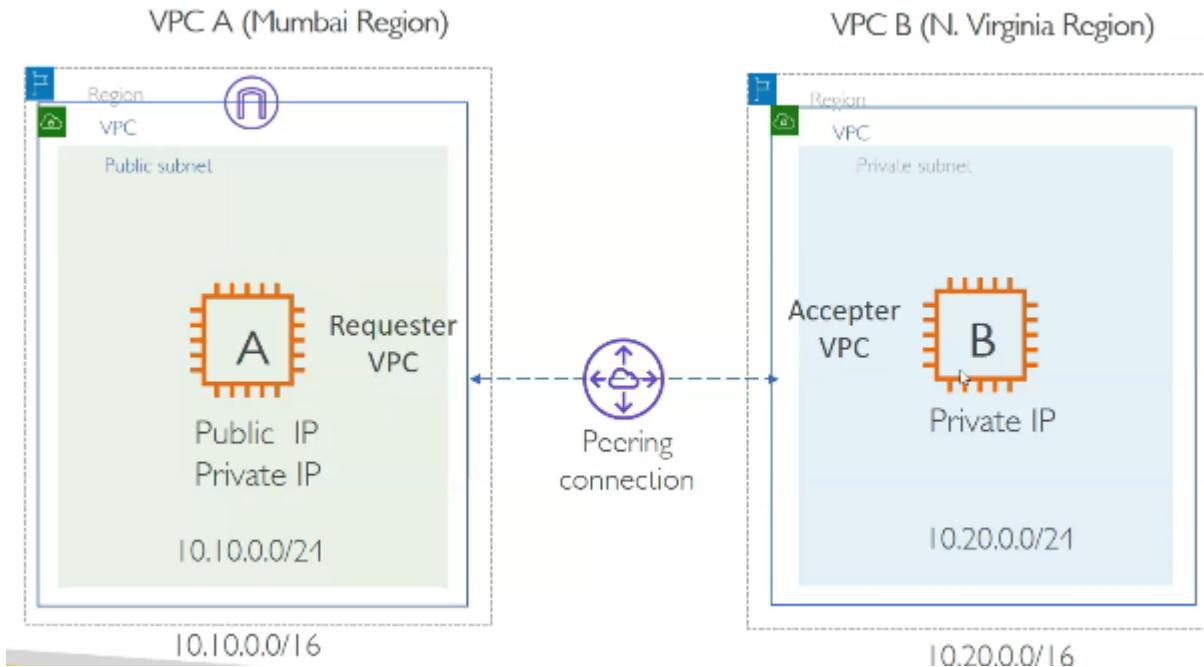
### VPC Peering is?

- AWS 네트워크를 이용해 VPC 2개를 연결하는 과정이며 동일한 네트워크를 사용하는 효과를 가지게 하는 것이다.
- AWS의 다른 계정에도 사용할 수 있고 다른 리전의 VPC에도 연동이 된다.
- CIDR이 겹치지 않아야 한다.

### Step

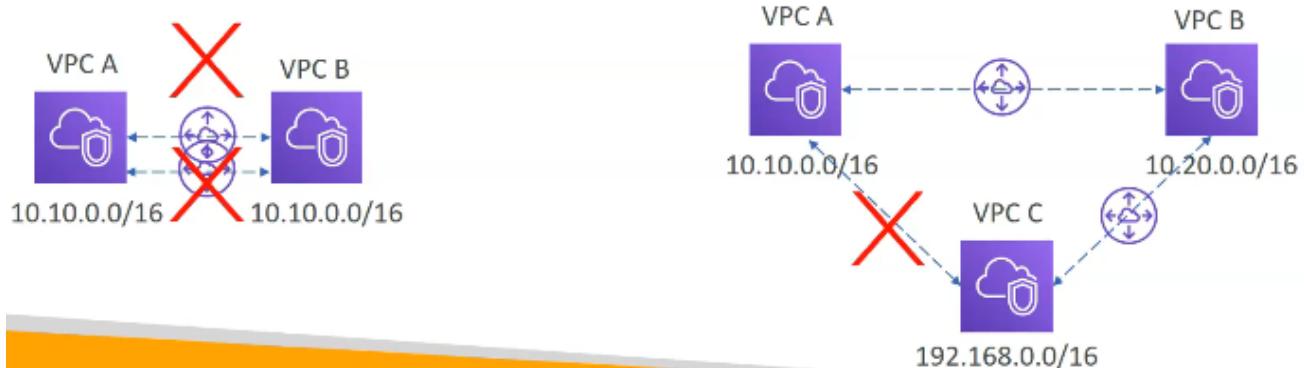
1. Create VPC-A (10.10.0.0/16)
2. Attach igw with VPC-A
3. Create Public Subnet in VPC-A (10.10.0.0/24)
4. Create EC2 in Public Subnet and Assign IP and Open Port 22
5. Create VPC-B (10.20.0.0/16)
6. Create Private Subnet in VPC-B (10.20.0.0/24)
7. Create EC2 in Private Subnet and Open Port 22, ICMP open in VPC-A CIDR

8. Create VPC Peering VPC-A to VPC-B
9. Accept Connect request in VPC-B
10. Modify Route Table in both side for Traffic on other VPC
11. Login EC2-A Instance from EC2-B (ping or SSH)

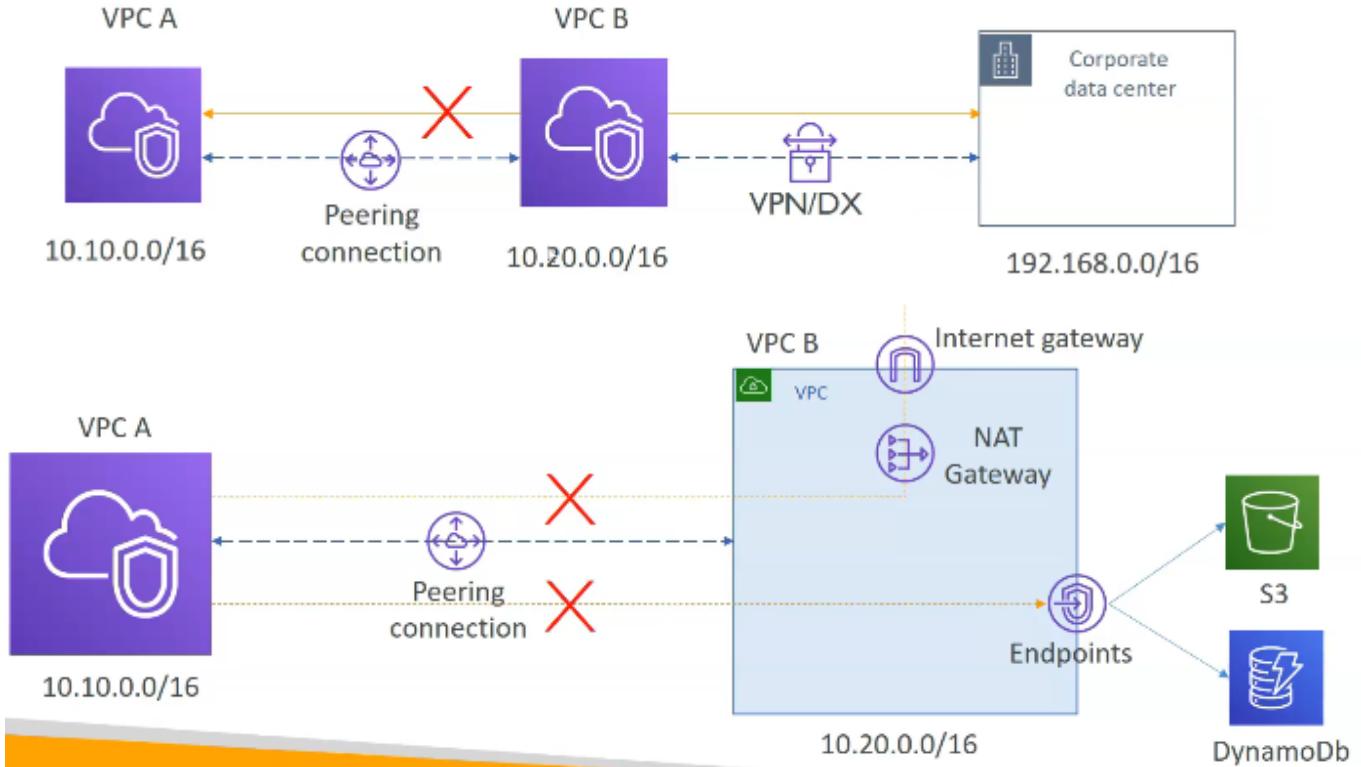


## Limit

- CIDR 겹치면 안됨
- VPC Peering은 2개의 VPC 피어링에 대해서만 허용한다.
- 링크 형태로 물려있다고 해서 통신이 되진 않는다.
- VPC당 최대 25개의 Peering Connection만 허용된다.



## VPC Peering으로 통신이 안되는 사례

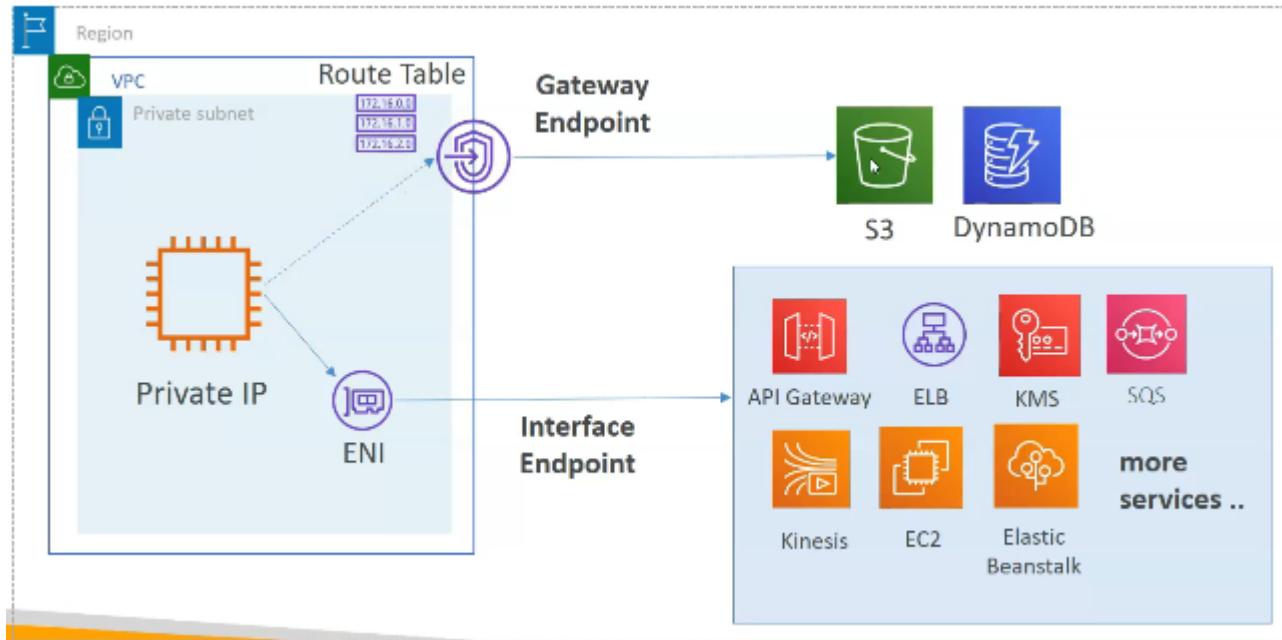


## 6. VPC Gateway Endpoint

### 개요

- Private Network 상황에서 다른 AWS Service를 사용할 수 있게 한다.
- AWS 타 서비스 연결 시, IGW, NAT Gateway가 필요하지 않다.
- 인터넷 연결을 피함으로서 네트워크 요금을 감면할 수 있다.
- 고가용성이며 별도의 트래픽 제한이 없다.
- Gateway Endpoint : Route Table을 통해서 연결이 가능 (S3, DynamoDB)**

- Interface Endpoint : ENI를 통해서 연결이 가능 (기타 AWS Service)



## Gateway Endpoint

- S3, DynamoDB가 VPC의 Private connect에서 연결될 수 있도록 한다.
- Route Table에서 S3와 DynamoDB의 Gateway Endpoint가 연결될 수 있도록 해야 한다.
- S3 엔드포인트가 만들어지면 prefix 리스트가 VPC에 생성된다.
- prefix 목록은 S3를 사용하는 IP의 집합 (pl-xxxxxx 형태)
- 라우팅 테이블 및 Security Groups에서도 사용이 가능하다.

Destination	Target	Status
10.0.0.0/16	local	active
pl-78a54011 (com.amazonaws.ap-south-1.s3, 52.219.62.0/23, 3.5.212.0/23, 3.5.208.0/22, 52.219.64.0/22)	vpce:0e89398e630ab0bcf	active
0.0.0.0/0	nat-085190f27ccf2ca3e	active

Security Group: sg-17147973			
Edit			
Type	Protocol	Port Range	Destination
HTTPS	TCP	443	pl-68a54001
HTTP	TCP	80	pl-68a54001

- 보안
  - VPC Endpoint 정책에 특정 Bucket만 접근 허용도 가능하다.
  - IAM과 권한 처리가 유사함.
  - 특정하는 S3 Bucket, DynamoDB Table 모두 가능

“Pasted image 20250105115618.png” 을 찾지 못했습니다.

- 정책

- 리소스 기반정책 설정이 가능

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::my_secure_bucket",
        "arn:aws:s3:::my_secure_bucket/*"
      ]
    }
  ]
}
```

Restrict access to S3 Bucket

```
{
  "Statement": [
    {
      "Sid": "AccessToSpecificTable",
      "Principal": "*",
      "Action": [
        "dynamodb:BatchGet",
        "dynamodb:Delete*",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Update*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/StockTable"
    }
  ]
}
```

Restrict access to DynamoDB table

- S3에선 이렇게 설정이 가능.

- aws:sourceVpce 형태로 vpc endpoint가 접근하는 부분에 대해서 제어가 가능함.
- aws:sourceVpc의 경우, 특정 VPC만 차단함.
- 필요한 경우 Public IP로도 제한이 가능하고, Private IP는 처리할 수 없음.

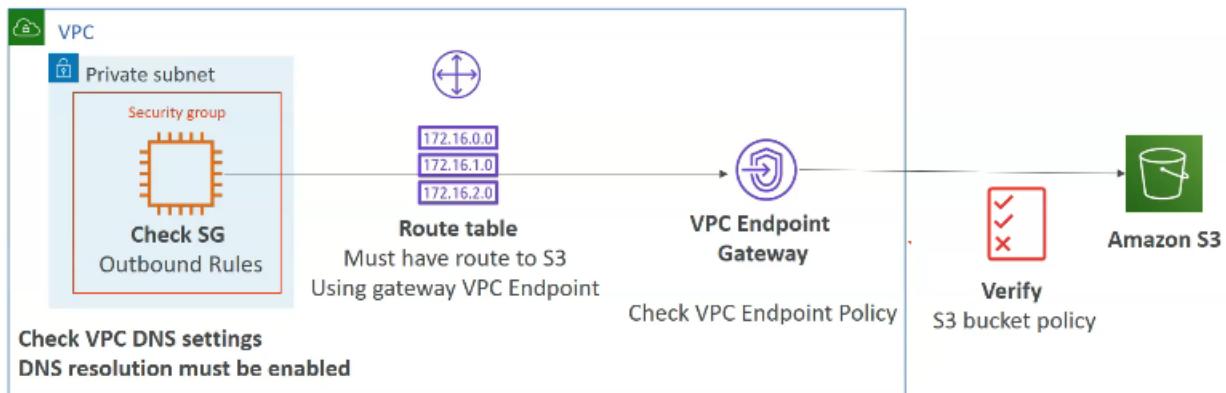
```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::my_secure_bucket",
        "arn:aws:s3:::my_secure_bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-1a2b3c4d"
        }
      }
    }
  ]
}
```

- Steps

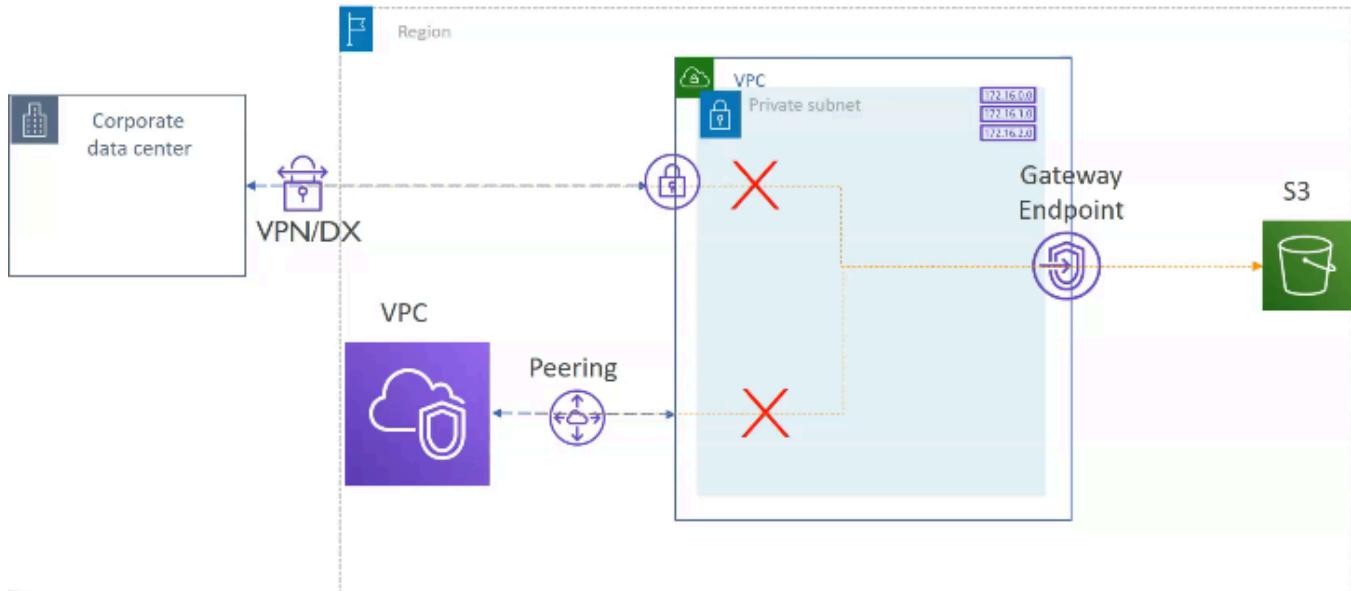
1. VPC 생성 후 Public/Private Subnet 생성
2. 각 서브넷에 EC2 생성
3. Private EC2의 IAM role에 S3 접근 권한 부여
4. VPC Gateway Endpoint for S3 생성
5. Private Subnet의 Route Table에 S3 Gateway Endpoint 수정
6. Public EC2에서 ssh로 Private EC2로 접근
7. S3 파일이 업로드 되는지 확인

- Case

- EC2 -> S3로 데이터를 보낼 때 막힌다?
- IAM Role에 막힌 게 있는지 확인할 것.
- Security Group에 VPC Endpoint가 열려 있는지 확인할 것.
- Route Table에 VPC Endpoint가 있는지 확인할 것.
- VPC Endpoint 정책에 문제가 있는지 확인할 것.
- S3 Bucket Policy에 문제가 있는지 확인할 것.



## VPC Gateway Endpoint Access From Remote Network



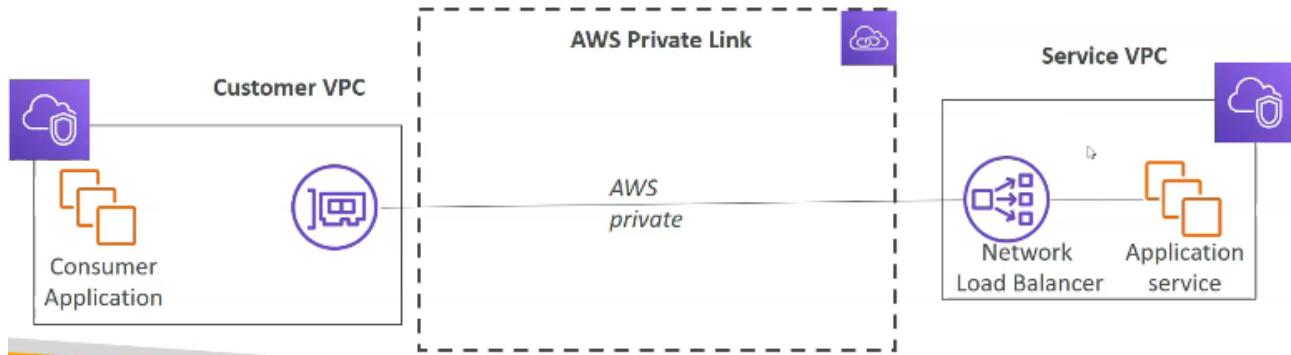
# 7. VPC Interface Endpoint and PrivateLink

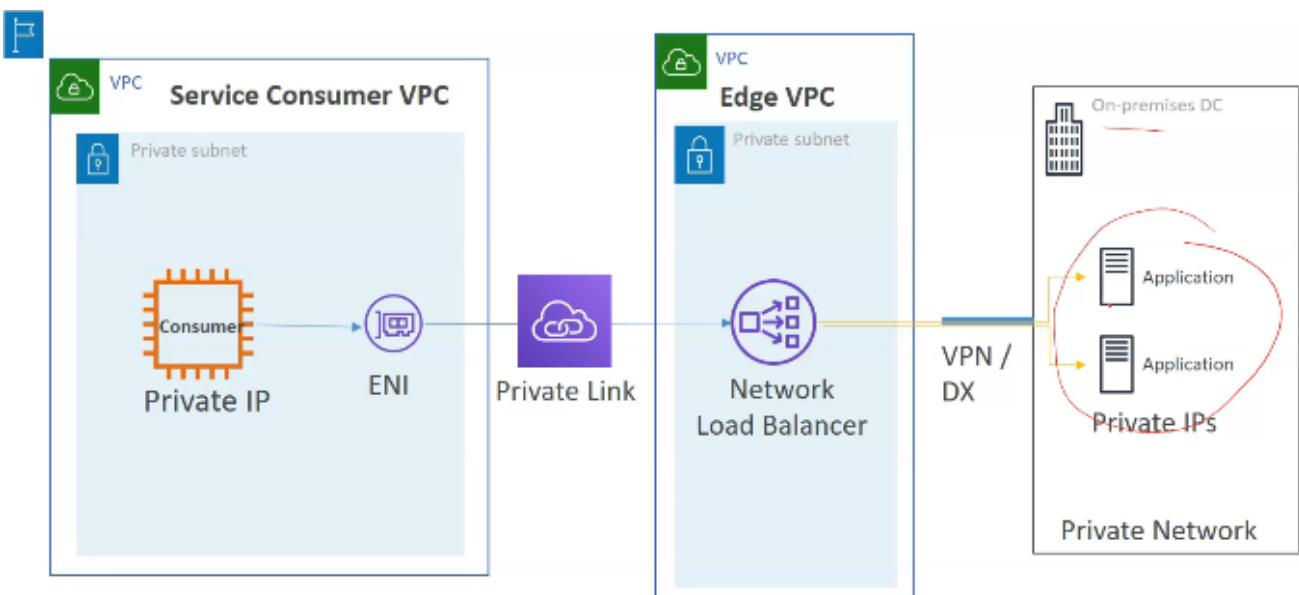
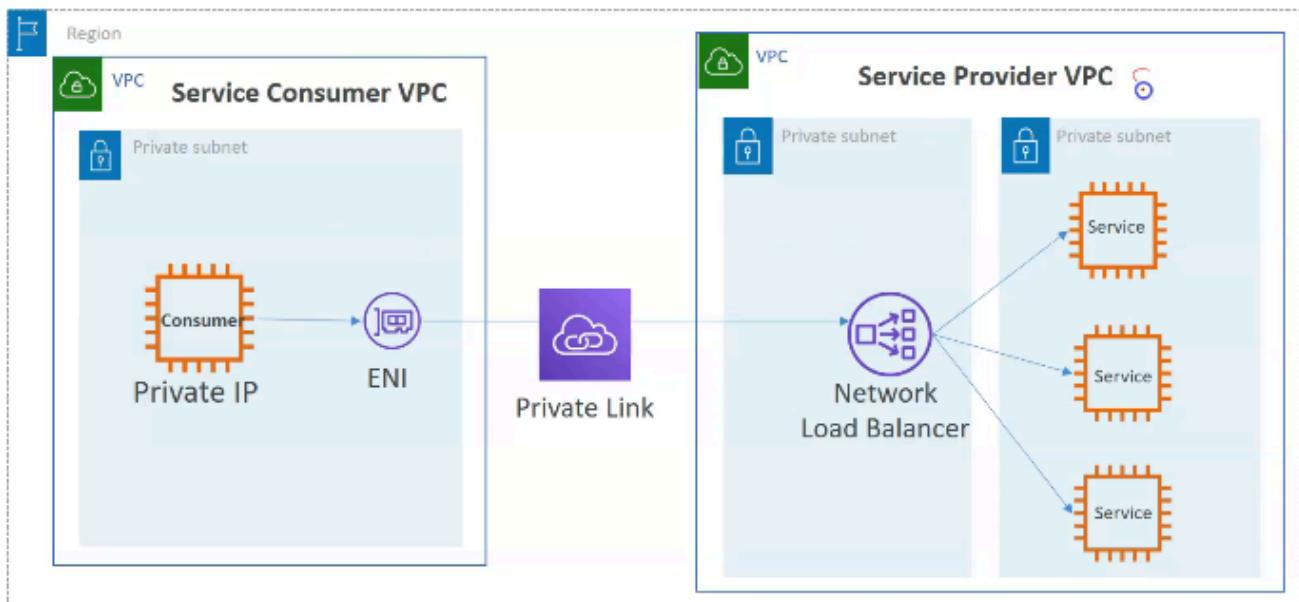
## VPC Interface Endpoint

- ENI를 이용해서 VPC에 접근
- Interface Endpoint는 AZ 별로 배치해야 한다.
- **0.01\$/hr per AZ**
- **0.01\$/GB**
- Security Group을 사용하므로 inbound rule을 항상 체크해야 함.
- 지역별 DNS를 사용하여 IP 선택할 수 있지만 특정 ENI를 선택은 불가능하다.
- 현재는 IPv4만 지원하고 있고 프로토콜은 TCP만 지원 가능.
- Route 53이 UDP를 지원하므로 Interface Endpoint를 통해 DNS를 체크하는 것이 아니기 때문에 TCP만 제공해도 현재는 큰 문제가 없음.

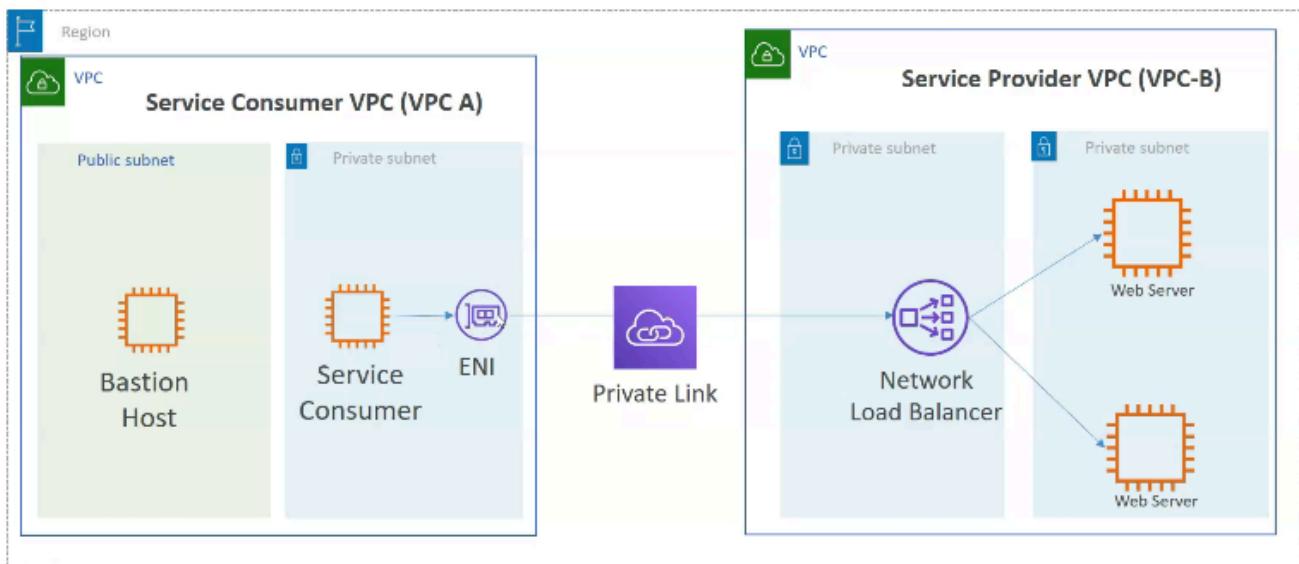
## VPC PrivateLink

- 보안에 좋고 1000개 가까운 VPC에 연동이 가능 (다른 계정 포함)
- VPC Peering, igw, NAT gw, route table 모두 불필요함.
- 대신 NLB (Service VPC)와 ENI (Customer VPC)가 필요함





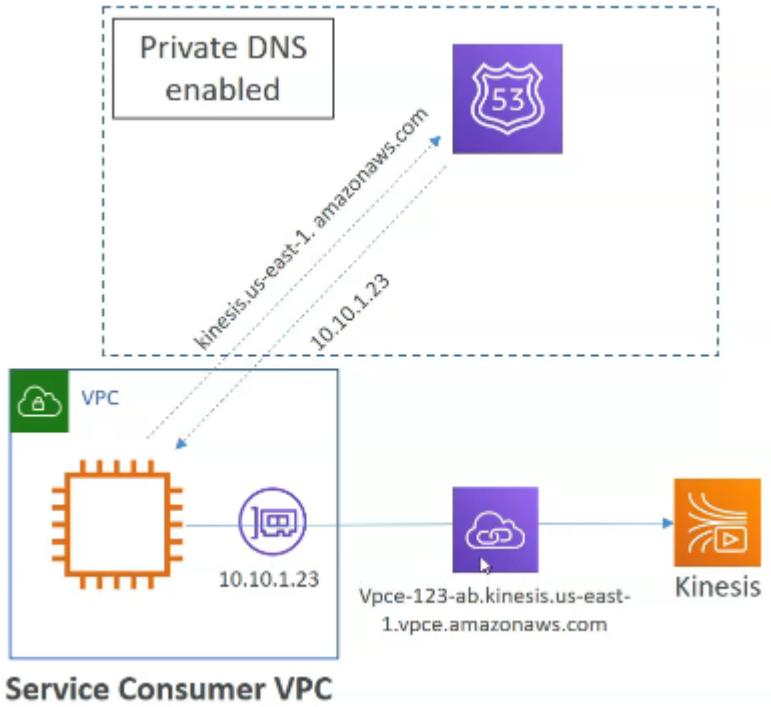
## Accessing Customer Service - Demo



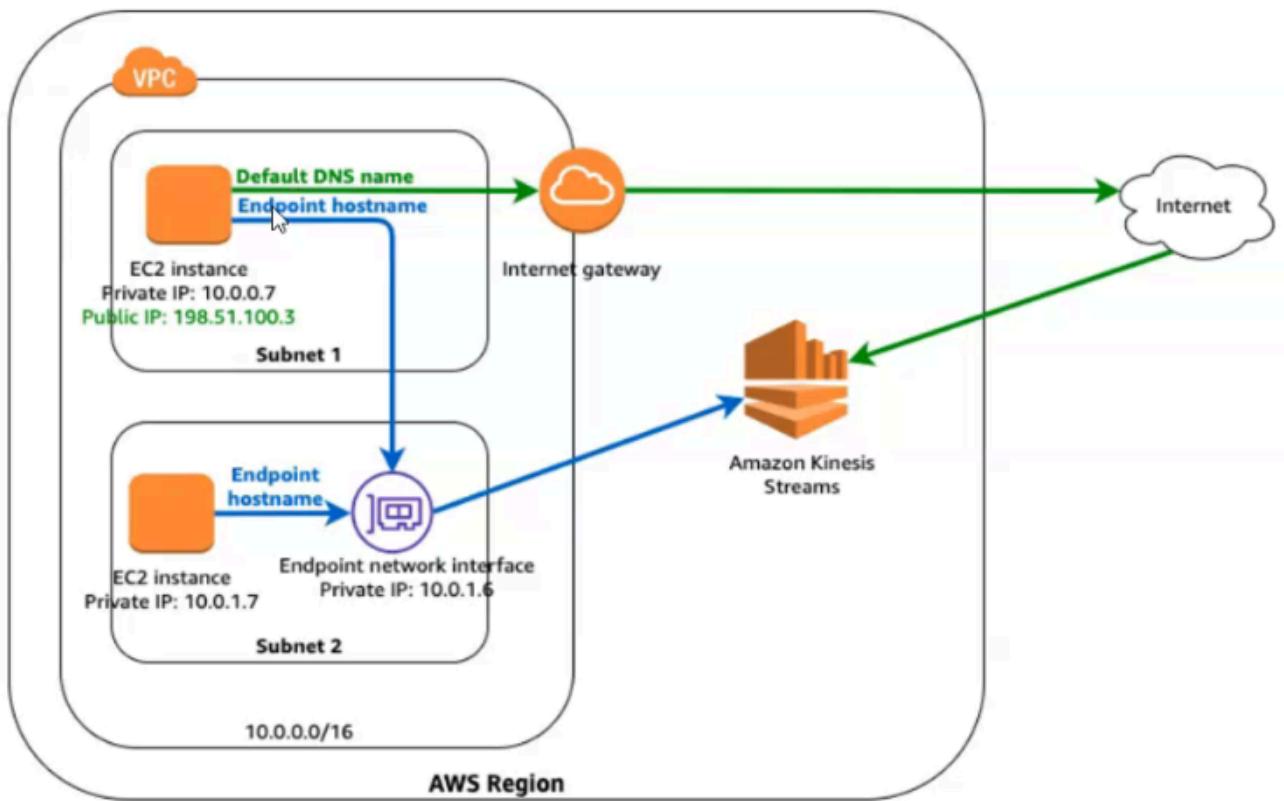
- 사전 준비 사항 – httpd 웹 서버를 포함한 EC2 AMI를 생성합니다. 이를 사용하여 VPC-B의 Private EC2 인스턴스를 실행해 더미 서비스를 호스팅합니다.
- VPC-B를 생성하고 2개의 Private 서브넷을 설정합니다.
- 이전에 생성한 AMI를 사용하여 Private 서브넷에 EC2 인스턴스를 실행합니다.
- 다른 Private 서브넷에 NLB(Network Load Balancer)를 생성하고, NLB 뒤에 EC2 인스턴스를 등록합니다.
- VPC-B에서 VPC Endpoint Service를 생성하고 NLB를 연결합니다.
- VPC-A와 VPC-B가 서로 다른 AWS 계정에 있는 경우, VPC-A의 AWS 계정을 허용 목록 (Whitelist)에 추가합니다.
- Public 및 Private 서브넷을 포함한 Service Consumer VPC(VPC-A)를 생성합니다.
- VPC-A에서 VPC Endpoint를 생성하고, 위에서 생성한 Endpoint Service를 검색합니다.
- Consumer VPC의 Private EC2 인스턴스에 로그인하고 VPC Endpoint DNS에 접근합니다.

## VPC Interface Endpoint - DNS

- private endpoint interface hostname을 가진 ENI를 배포
- interface endpoint용 private DNS 세팅
  - 서비스의 public hostname이 private hostname으로 IP Resolve 해줄 것이다.
  - VPC 세팅: "Enable DNS Hostnames"와 "Enable DNS Support"가 꼭 켜져 있어야 한다.
- Private DNS가 켜져 있으면, 소비자 VPC에는
  - Public Pattern: servicename.region.amazonaws.com
  - Private Pattern: vpce-12345.ab.ec2.us-east-1.vpce.amazonaws.com

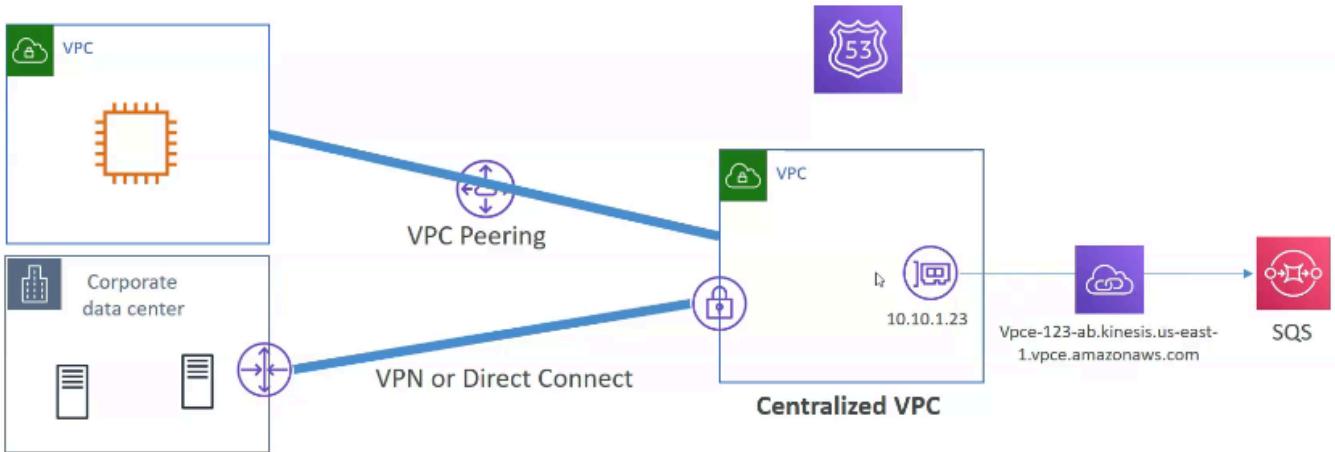


Service Consumer VPC



- Subnet 1 -> kinesis.us-east-1.amazonaws.com -> igw
- Subnet 1 (no igw) -> vpce-123-ab.kinesis.us-east-1.vpce.amazonaws.com -> interface endpoint
- Subnet 2 (no igw) -> <<Interface Endpoint DNS를 쓴다는 조건>> interface endpoint -> kinesis

## VPC Interface Endpoint from 원격 네트워크



- VPC에는 당연히 정상적으로 동작함.
- On-Premise는 직접 해석이 불가능 하므로 Custom Route 53 Resolver를 사용하여 해결해야 함.

## VPC PrivateLink vs VPC Peering

	VPC Peering	VPC PrivateLink
연결성	Peered된 VPC간의 여러 리소스와 연동이 가능하다.	타 VPC의 특정 어플리케이션만 연결할 수 있다.
연결 제한	CIDR이 중복된 경우, 연결할 수 없다.	CIDR이 중복되더라도 상관없이 연결할 수 있다.
연결 max	최대 125개 연결 Peering	제한 없음.
양방향 통신	Peering이 되고 나면 양방향 통신이 허용됨	단 방향만 통신 가능. 즉 A->B는 되지만, B->A는 되지 않음.

## Summary

- **VPC 피어링**은 동일 리전 또는 서로 다른 리전의 두 VPC 간 통신을 가능하게 합니다.
- **VPC 엔드포인트**는 인터넷 게이트웨이, NAT 장치, VPN 연결 또는 AWS Direct Connect 연결 없이, PrivateLink를 통해 지원되는 AWS 서비스 및 기타 서비스를 VPC와 안전하게 연결할 수 있도록 합니다.
- VPC 내 인스턴스는 서비스 내 리소스와 통신하기 위해 공인 IP 주소가 필요하지 않습니다.
- VPC와 다른 서비스 간 트래픽은 Amazon 네트워크를 벗어나지 않습니다.
- **게이트웨이 VPC 엔드포인트**는 서비스를 연결할 수 있게 해주지만 VPC 내 네트워크 구성 요소로 존재하지는 않습니다.
- **인터페이스 VPC 엔드포인트**는 탄력적 네트워크 인터페이스, 비공개 IP 주소, DNS 이름으로 구성되며, 이를 통해 VPC 내부에서 AWS 클라우드 서비스를 안전하게 액세스할 수 있습니다.

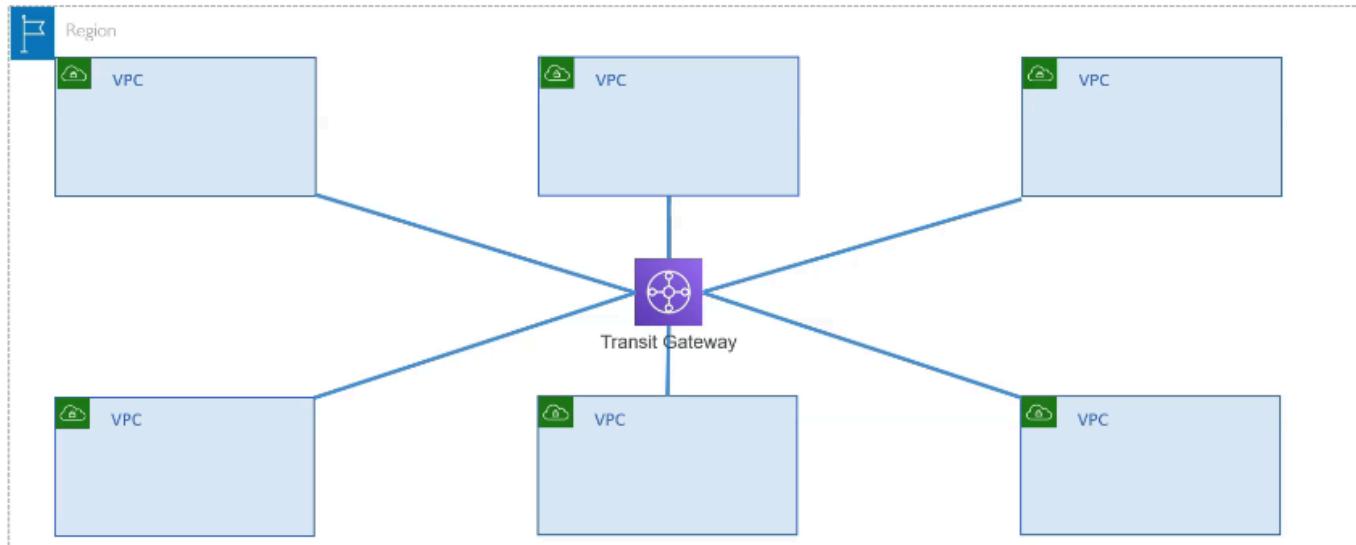
다.

- **AWS PrivateLink**는 인터페이스 VPC 엔드포인트의 확장 기능으로, 사용자가 자신의 엔드포인트를 생성하거나 다른 사용자가 생성한 엔드포인트를 사용할 수 있도록 합니다.

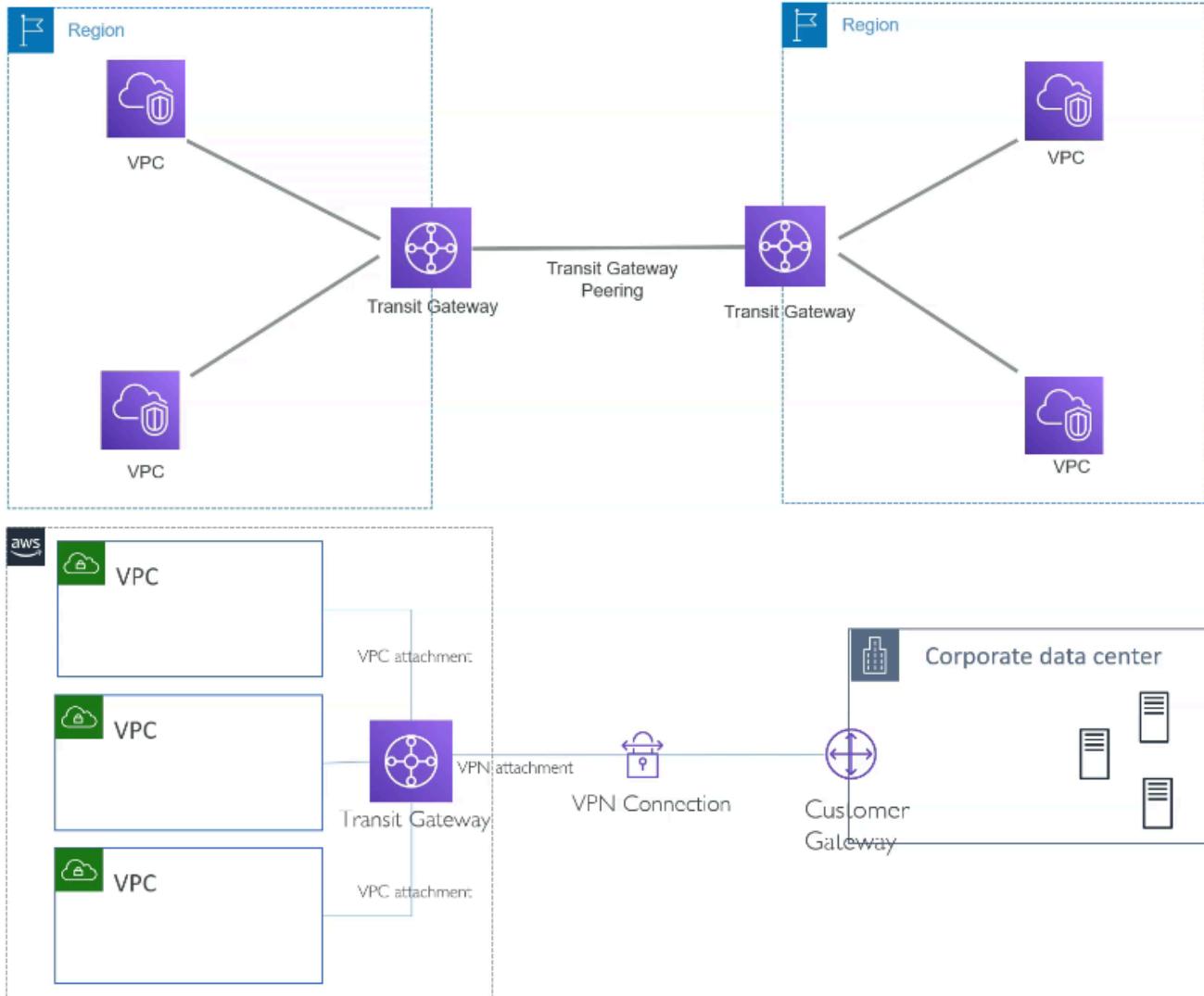
## Exam Essential

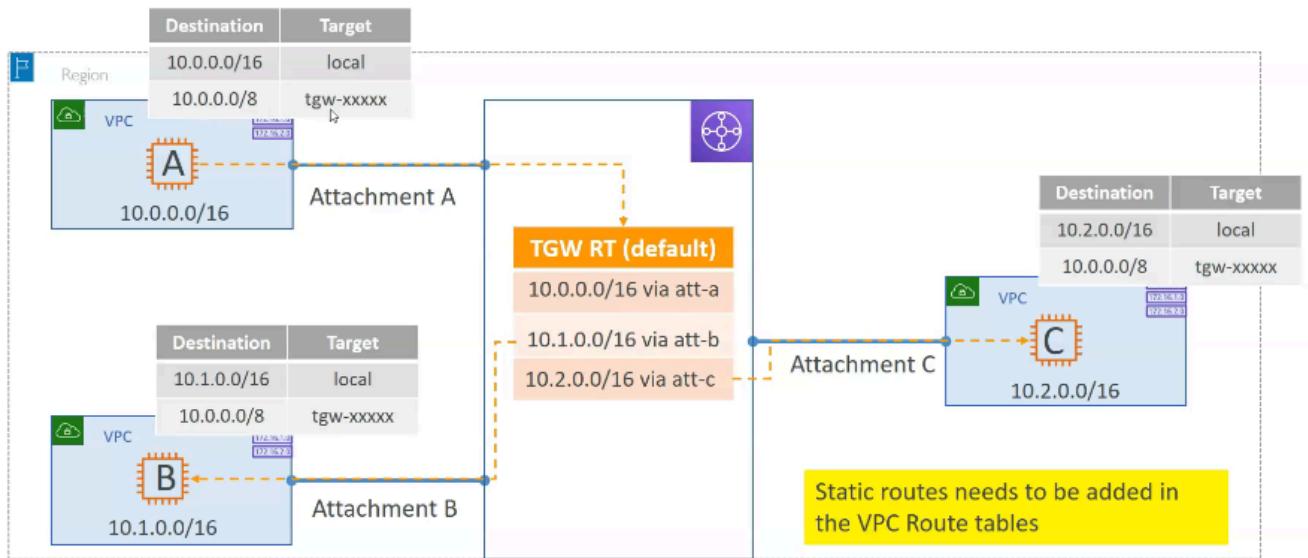
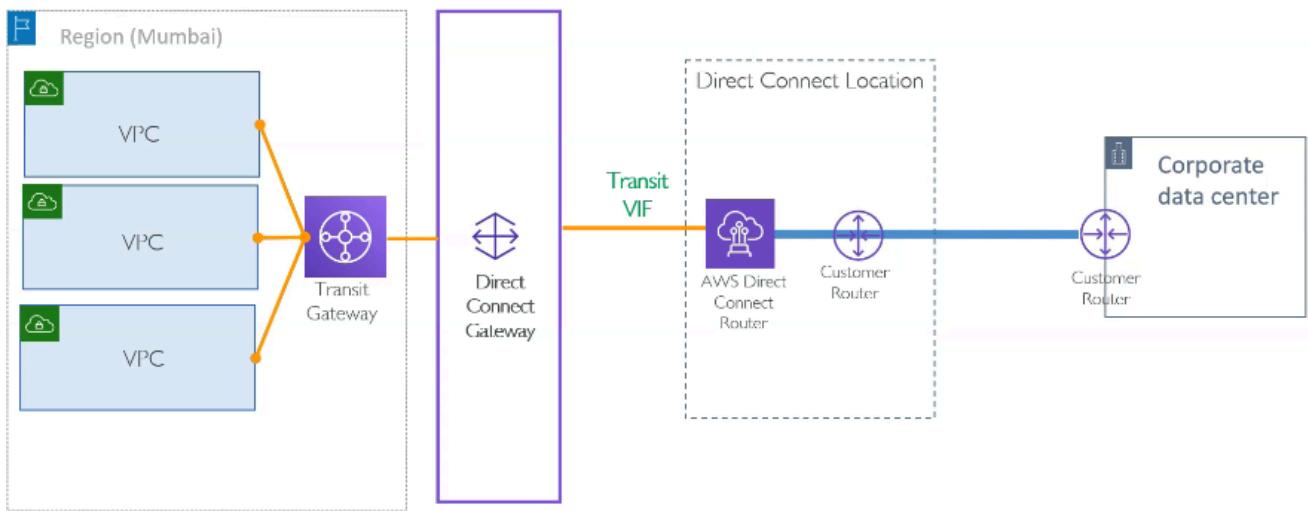
- **VPC 피어링**은 전달 라우팅(transitive routing)을 지원하지 않습니다. 따라서 피어링 연결을 통해 다른 VPC의 IGW(인터넷 게이트웨이)나 NAT에 접근할 수 없습니다.
- VPC 피어링은 다른 VPC의 보안 그룹(Security Group)을 인바운드 규칙의 소스로 사용할 수 있습니다.
- 최대 125개의 VPC 피어링 연결을 생성할 수 있습니다.
- **VPC Gateway Endpoint**는 IGW(인터넷 게이트웨이)나 NAT 없이 동일 AWS 리전에서 S3 또는 DynamoDB와의 비공개 연결을 가능하게 합니다.
- VPC Gateway Endpoint를 통해 트래픽을 라우팅하려면 필요한 서브넷의 라우팅 테이블을 수정해야 합니다.
- VPC Gateway Endpoint는 Direct Connect/VPN 또는 VPC 피어링을 통해 액세스할 수 없습니다.
- **VPC 인터페이스 엔드포인트**는 서브넷에 ENI(탄력적 네트워크 인터페이스)를 생성합니다.
- 인터페이스 엔드포인트는 리전 및 영역 DNS 이름을 수신합니다.
- **Route53 프라이빗 호스팅 존**을 사용하여 인터페이스 DNS에 대한 Alias 레코드를 활용해 커스텀 DNS를 사용할 수 있습니다.
- 인터페이스 엔드포인트는 **Direct Connect 연결**, AWS 관리형 VPN, 그리고 VPC 피어링 연결을 통해 액세스할 수 있습니다.
- 트래픽은 VPC 내 리소스에서 시작되며, 엔드포인트 서비스는 요청에만 응답할 수 있습니다. 엔드포인트 서비스가 요청을 시작할 수는 없습니다.

## 8. Transit Gateway

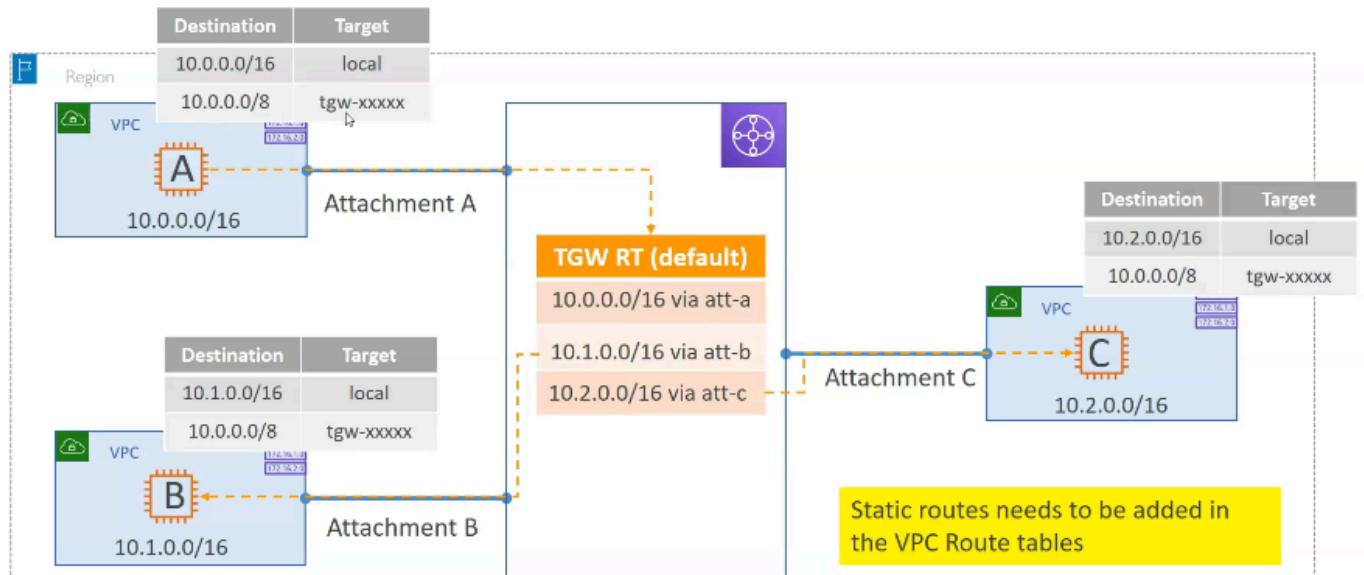


- 천개 이상의 VPC 또는 On-Premise 네트워크와 연동 가능
  - 1개 이상의 VPC
  - TGW와도 연동 가능
  - SD-WAN/3rd party network appliance와도 연동 가능
  - VPN
  - Direct Connect Gateway
- Multicast Support, MTU, Appliance Mode, AZ 고려, 계정 간 TGW 공유



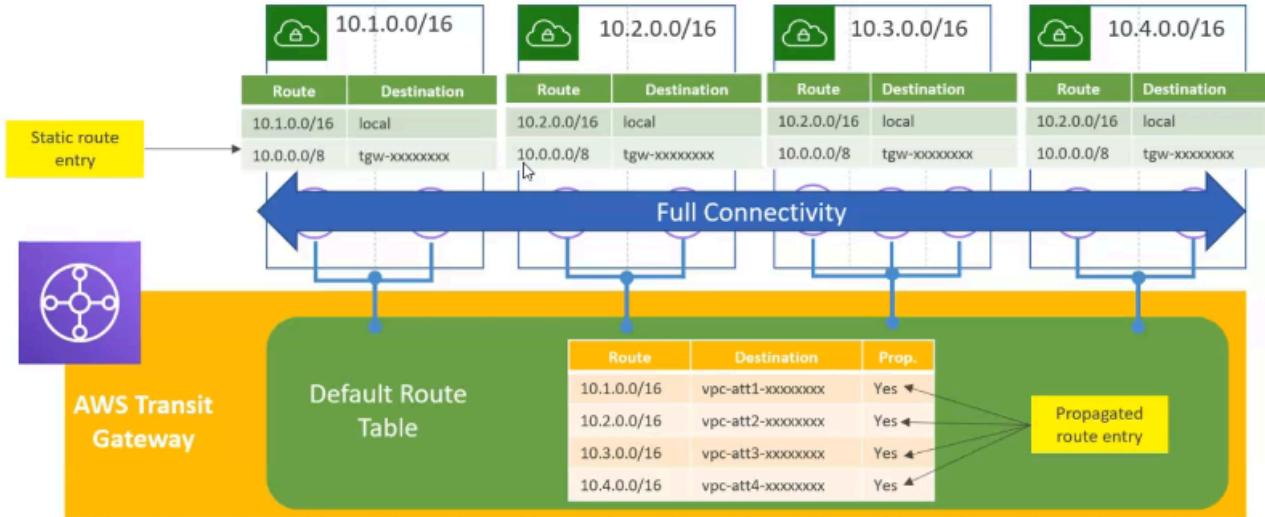


## TGW Attachments

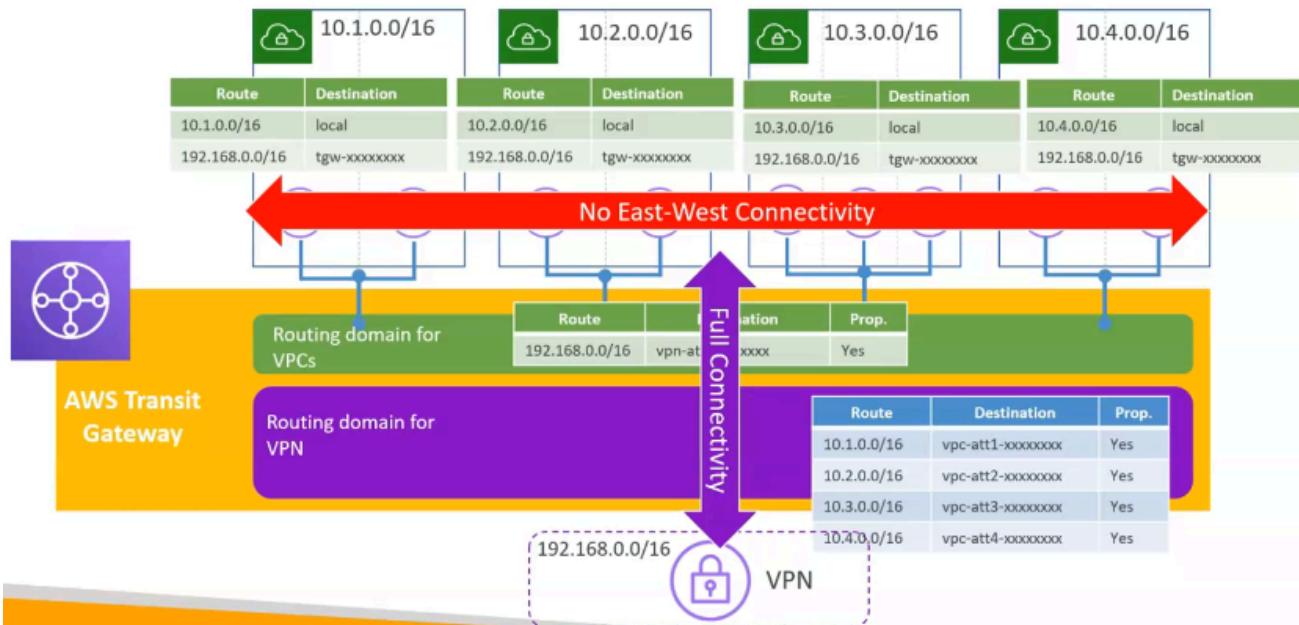


## TGW VPC Network Pattern

- **Flat Network** : TGW에 VPC가 평면으로 붙은 hub & spoke의 전형적인 모형
  - TGW Routing table의 경우 전체 연결이 가능하며 Routing 도메인이 하나다.



- **Segmented Network**: Flat 구조 이외에 VPN이 붙는다던지 한다면 기본 AWS 네트워크와 VPN 연결을 구성한다.
  - VPC간은 통신을 안하지만 모든 VPC가 VPN과 통신이 가능하고 VPN측 추가 라우팅에서 VPC와 IP대역을 연결하면 통신이 가능하게 한다.

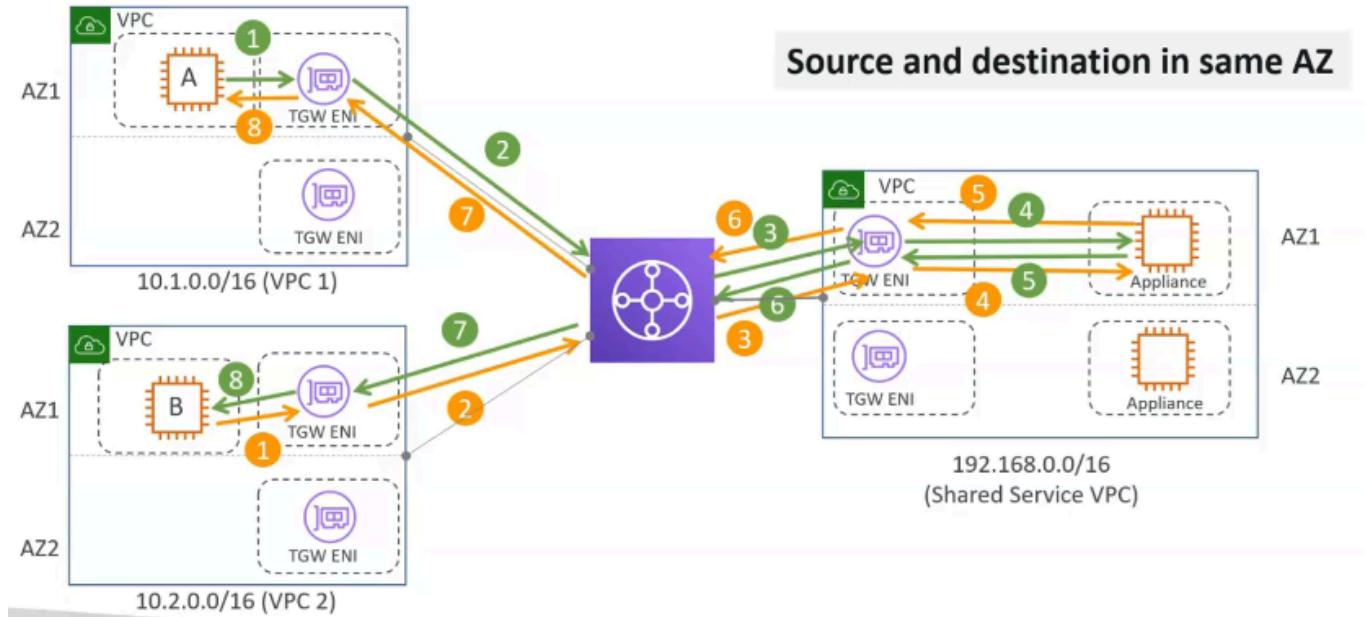


## VPC & Subnet Design for TGW

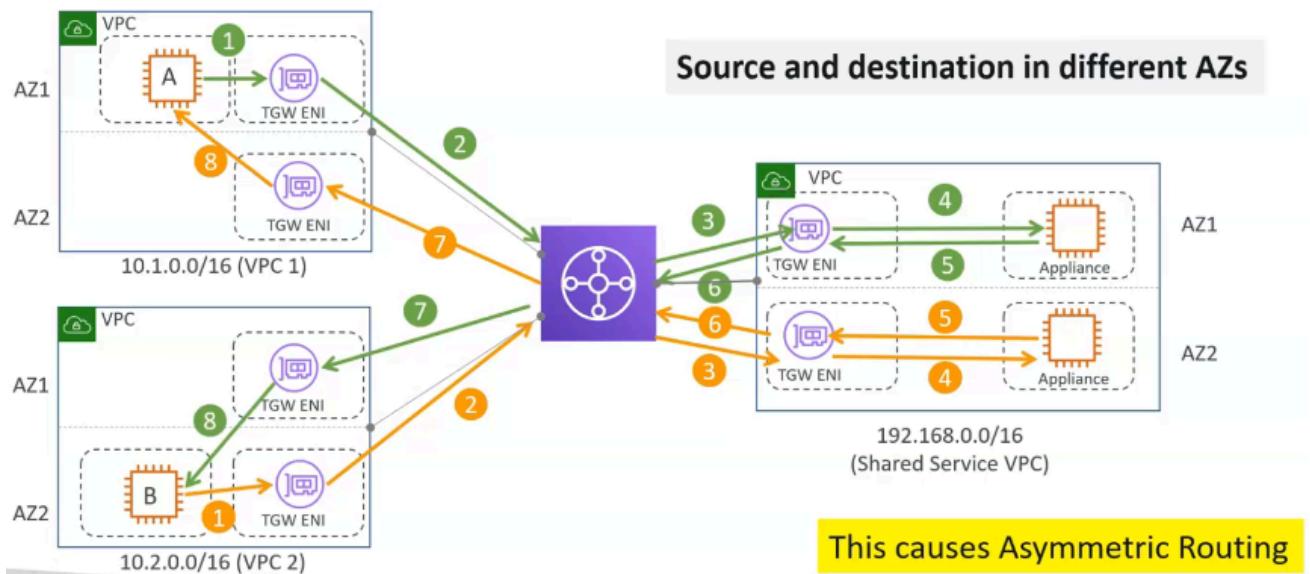
- VPC를 TGW에 연결할 때, TGW가 VPC 서브넷의 리소스로 트래픽을 라우팅할 수 있도록 하거나 이상의 가용 영역(Availability Zone)을 활성화해야 합니다.
- 각 가용 영역을 활성화하려면 정확히 하나의 서브넷을 지정해야 하며(일반적으로 워크로드 서브넷을 위해 IP를 절약할 수 있는 /28 범위 사용),
- TGW는 해당 서브넷에 네트워크 인터페이스를 배치하며 서브넷에서 하나의 IP 주소를 사용합니다.

- 가용 영역을 활성화한 후에는 해당 영역의 모든 서브넷으로 트래픽을 라우팅할 수 있으며, 지정된 서브넷에만 한정되지 않습니다.
- TGW 연결이 없는 가용 영역에 위치한 리소스는 TGW에 접근할 수 없습니다.

## TGW AZ Affinity & Appliance Mode



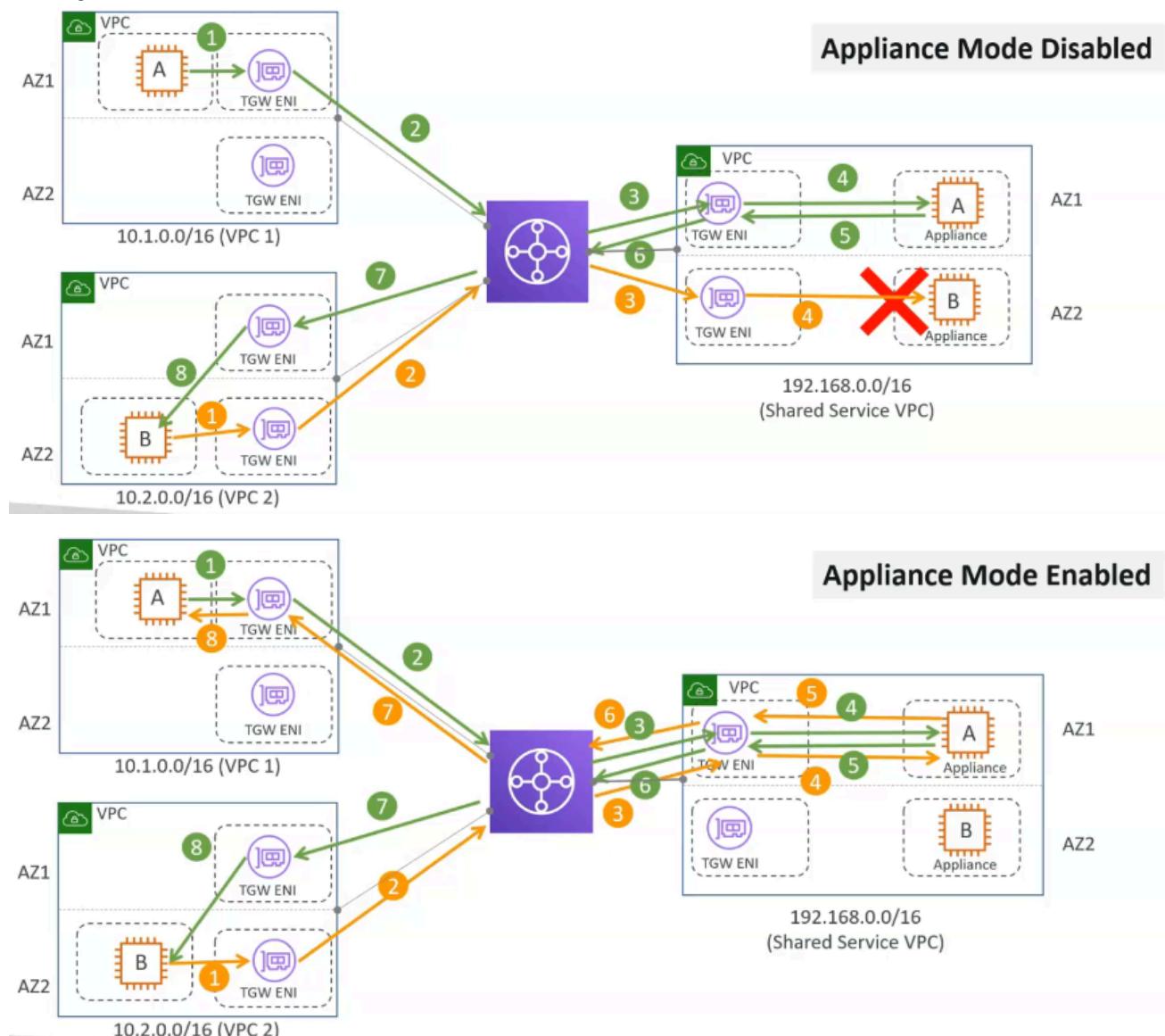
- TGW는 목적지 트래픽에 도달할 때까지 같은 AZ간의 통신으로 유지하려고 한다.
  - 요금 절감
  - 트래픽 이동 적음
  - 만일 AZ가 다운되도 피해가 적음.



- 만일 통신하는 존이 다르다면 비대칭 라우팅이라 한다.
- 네트워크 어플라이언스 상태가 달라서 어플리케이션에서 트래픽 일치 여부를 체크할 경우 문제가 될 수 있다. 아래처럼 어플라이언스에서 트래픽을 삭제처리 할 수 있으므로 어플라이언스 모드를 활성화해야 함.

- Flow Hash Algorithm을 이용해서 반환 트래픽에 대해서 Appliance 서버의 AZ가 달라도 그 방향으로 넣어주도록 조정한다.

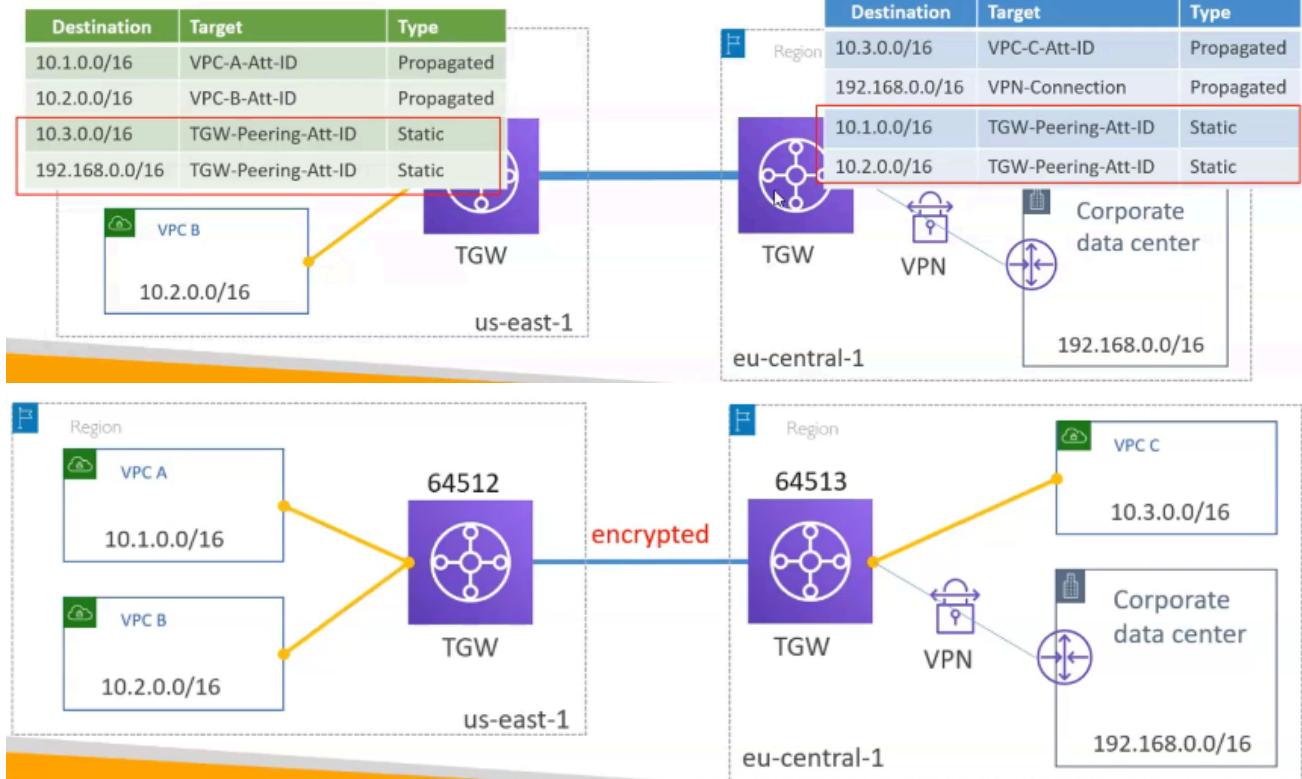
- 5 tuples: Source Port, IP, Dest Port IP, Protocol을 이용함.



## TGW Peering

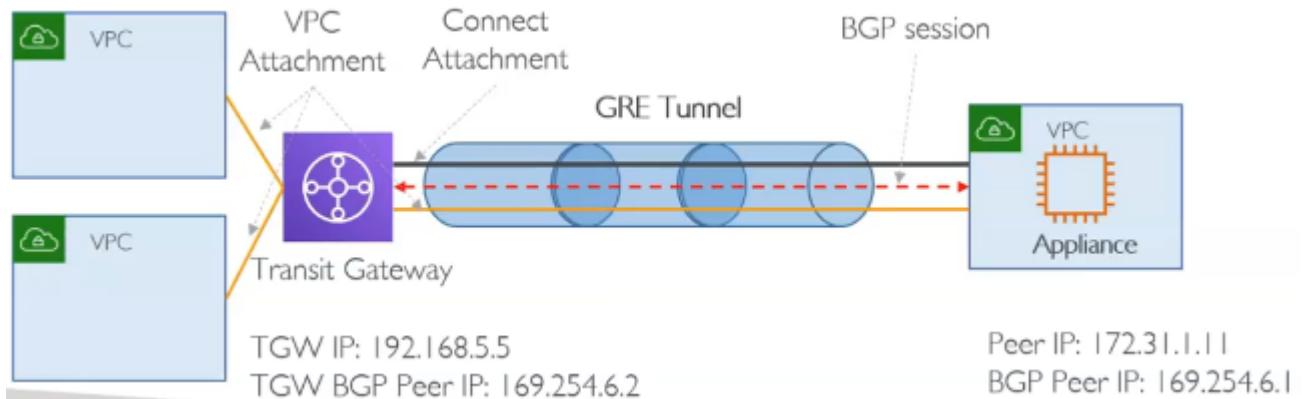
- TGW는 리전 기반 라우터로, 동일한 리전 내에서 VPC를 연결할 수 있습니다.
- 리전 간 네트워크 연결을 위해 트랜짓 게이트웨이를 서로 피어링할 수 있습니다.
- 피어링 연결을 위해 **Static Routes**를 추가해야 하며, **BGP는 지원되지 않습니다.**
- 리전 간 트래픽은 암호화되어 AWS 글로벌 네트워크를 통해 전송되며, 공용 인터넷에 노출되지 않습니다. 최대 **50 Gbps** 대역폭을 지원합니다.

- 피어링된 TGW에는 고유한 ASN을 사용하는 것이 좋습니다 (가능한 경우).

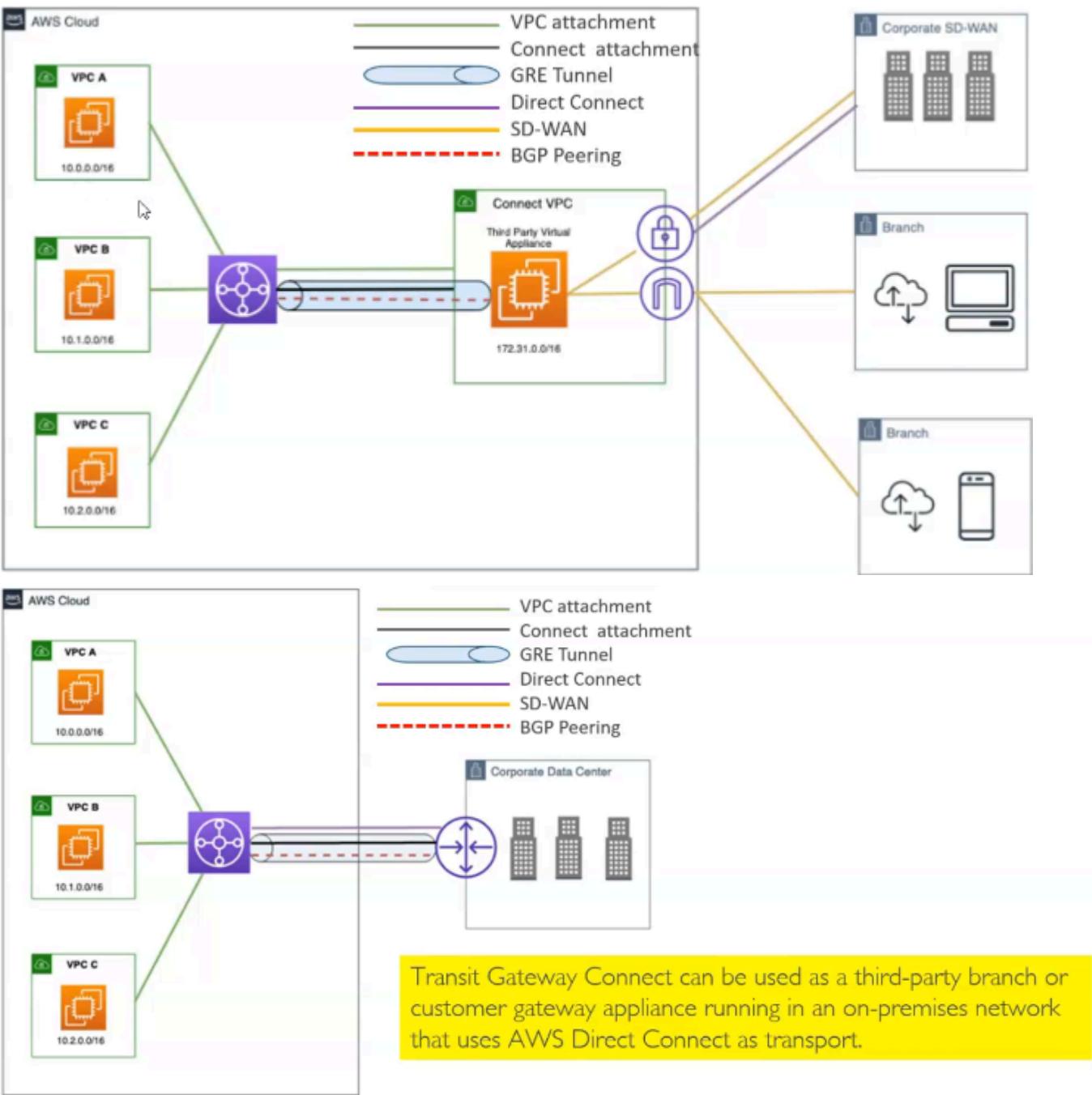


## TGW Connect Attachment

- Transit Gateway Connect Attachment**를 생성하여 트랜짓 게이트웨이와 VPC에서 실행 중인 서드파티 가상 어플라이언스(SD-WAN 어플라이언스 등) 간의 연결을 설정할 수 있습니다.
- Connect Attachment**는 기존 VPC 또는 AWS Direct Connect Attachment를 기본 전송 메커니즘으로 사용합니다.
- GRE(Generic Routing Encapsulation) 터널 프로토콜**을 통해 고성능 전송을 지원하며, **BGP(Border Gateway Protocol)**를 사용해 동적 라우팅을 설정합니다.



## TGW Connect Attachment over the transport Attachment



- TGW Connect Attachment는 DX를 이용해서 On-Premise 네트워크에서 통신하는 third-party 혹은 CGW Appliances에도 연결될 수 있다.

## TGW Connect Attachment Summary

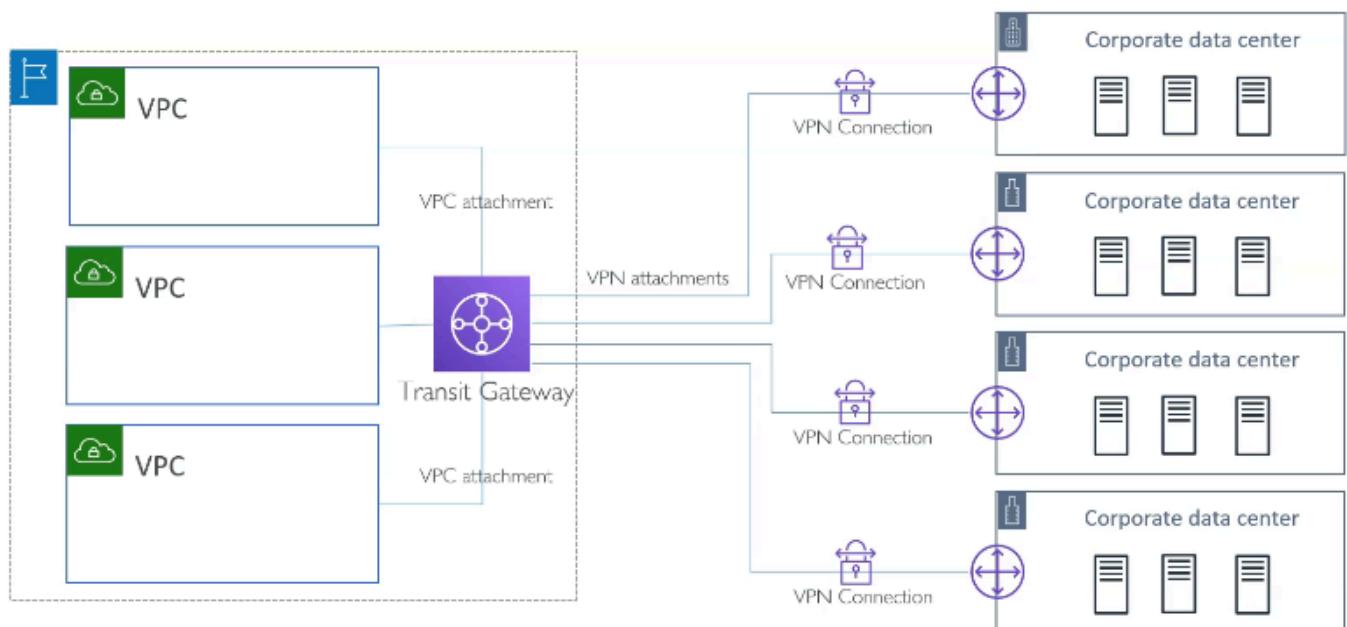
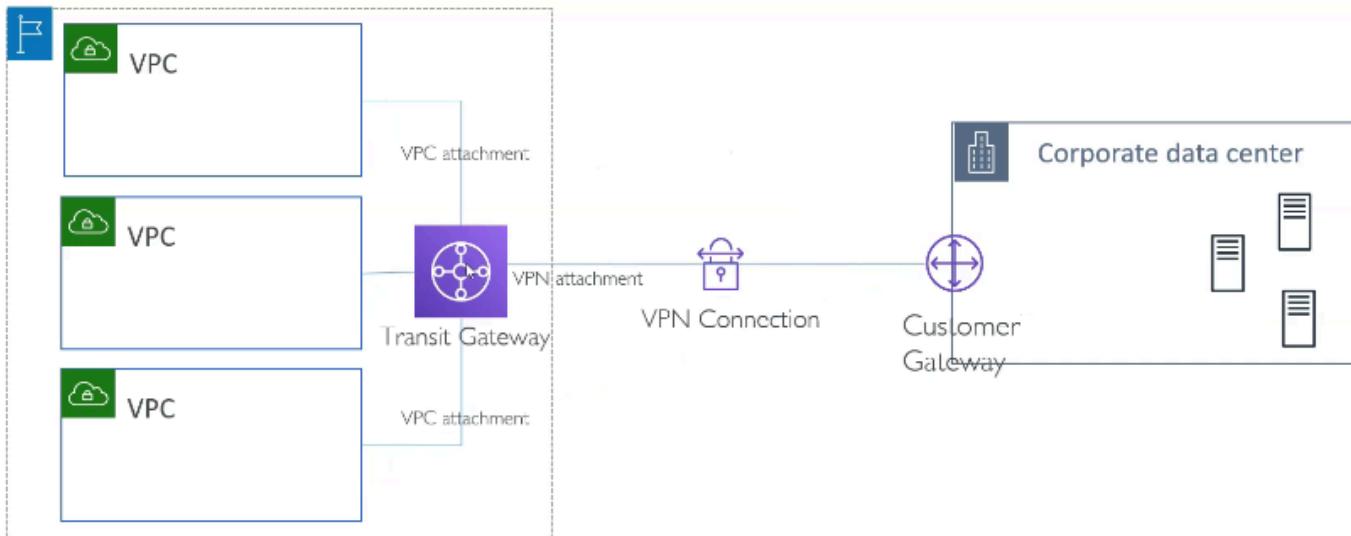
- Connect Attachments**는 정적 라우트를 지원하지 않습니다. **BGP**는 Transit Gateway Connect의 최소 요구 사항입니다.
- Transit Gateway Connect**는 GRE 터널당 최대 **5 Gbps**의 대역폭을 지원합니다. 5 Gbps 이상의 대역폭은 동일한 Connect Attachment에 대해 여러 Connect Peer(GRE 터널)에서 동일한 프리픽스를 광고(Advertise)함으로써 달성됩니다.
- 각 Connect Attachment는 최대 **4개의 Connect Peer**를 지원하며, 이를 통해 연결당 총 **20**

**Gbps**의 대역폭을 제공합니다.

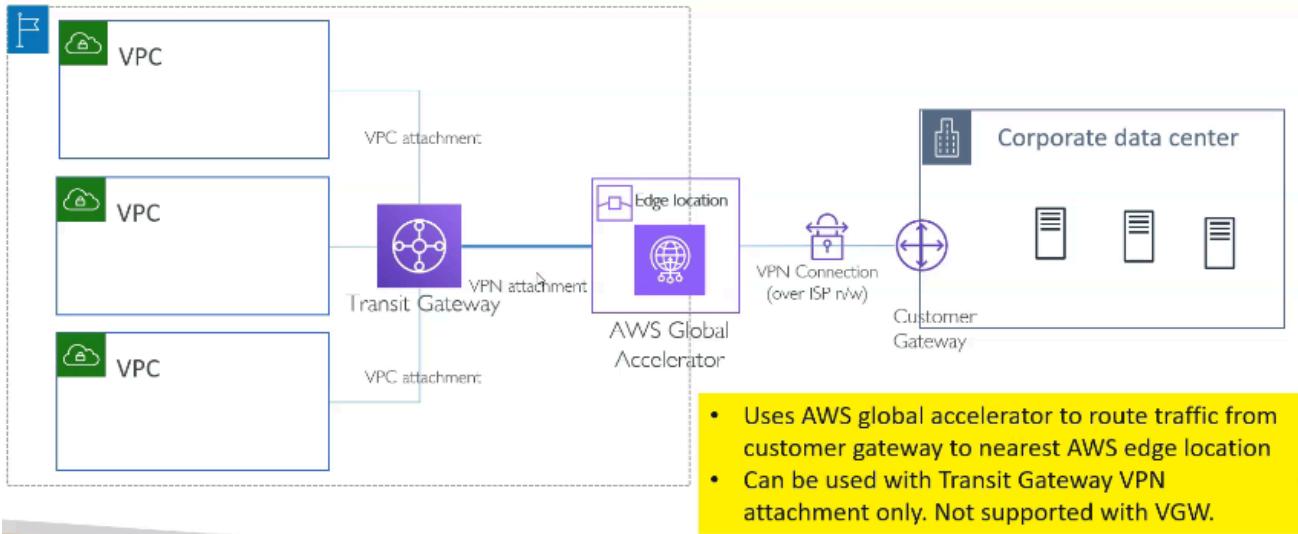
- 추가 설명:

- 정적 라우트 비지원: Connect Attachment에서는 정적 라우트 대신 **BGP**(Border Gateway Protocol)를 사용하여 동적으로 경로 정보를 교환합니다.
- GRE 터널 대역폭 확장:**
  - 기본적으로 GRE 터널당 최대 5 Gbps를 사용할 수 있지만, 여러 GRE 터널을 생성하고 동일한 경로 정보를 공유함으로써 대역폭을 확장할 수 있습니다.
  - 예: 4개의 GRE 터널을 사용하면 20 Gbps까지 대역폭을 확장 가능.
- BGP 필수:** GRE 터널과 함께 사용되는 동적 라우팅 프로토콜로 BGP는 연결된 네트워크 간에 경로 정보를 효과적으로 교환합니다.

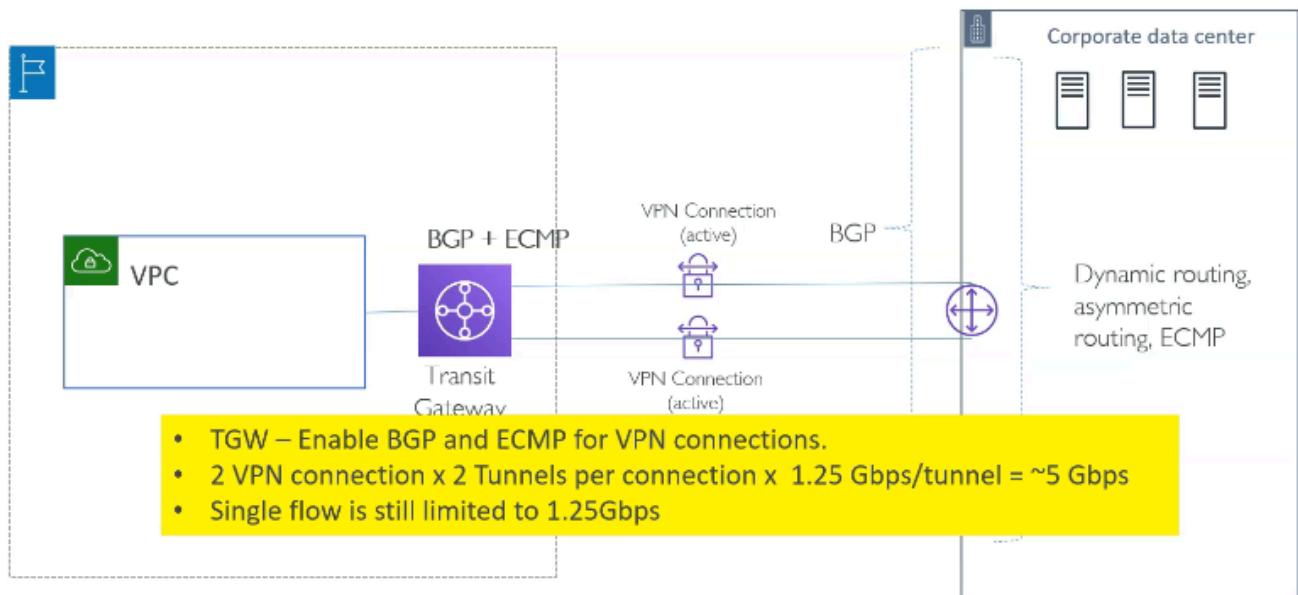
## TGW with VPN



- AWS Global Accelerator는 고객 게이트웨이에서 가장 가까운 AWS 엣지 로케이션으로 트래픽을 라우팅하는 데 사용됩니다.
- 이 설정은 **Transit Gateway VPN Attachment**와 함께 사용할 수 있으며, **Virtual Private Gateway(VGW)**와는 지원되지 않습니다.



- VPC to On-Premise 간 연결에서 추가적인 대역폭을 가지려면 다중 연결이 가능하다고 생각 할 수 있겠지만, VGW당 최대 25GB의 집계 대역폭을 가짐. 터널당 최대 1.25GB의 대역폭을 가지며 다중 연결 트래픽을 로드밸런싱 하려면 ECMP가 필요함.



#### • **Transit Gateway의 지원 연결 유형**

- VPC 연결
- Transit Gateway 피어링
- VPN 연결

#### • **VPN 연결 개요**

- Transit Gateway를 사용해 온프레미스 네트워크를 AWS로 연결 가능.
- AWS에서 제공하는 **Site-to-Site IPsec VPN**을 사용.

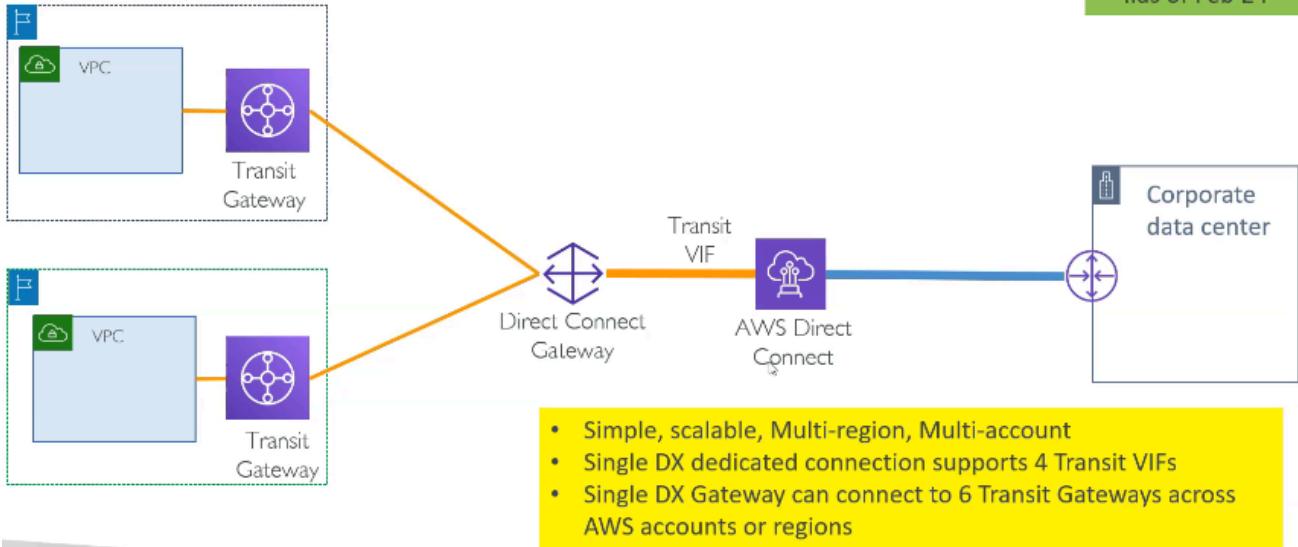
- VPC에 Virtual Private Gateway(VPN Gateway)를 연결하고, 고객 측에서는 고객 라우터를 사용하여 Site-to-Site VPN을 설정.
- **기존 아키텍처의 한계**
  - VPN 연결은 VPC 및 고객 게이트웨이마다 개별적으로 설정 필요.
  - 다수의 VPC 또는 브랜치 오피스를 연결할 경우 관리 복잡성이 증가.
- **Transit Gateway를 활용한 단순화**
  - Transit Gateway에 모든 VPC와 VPN 연결을 통합.
  - 다수의 VPN 연결을 Transit Gateway에 종결시켜 하이브리드 연결(온프레미스 ↔ AWS VPC)을 간소화.
  - VPN 연결 시 **AWS Global Accelerator**를 활용해 네트워크 경로 최적화:
    - 트래픽이 가까운 엣지 로케이션을 거쳐 AWS 백본 네트워크로 전송됨.
    - Virtual Private Gateway에서는 사용할 수 없음.
- **집계 대역폭 향상**
  - Virtual Private Gateway의 대역폭 한계:
    - VPN 터널당 최대 대역폭: 1.25 Gbps.
    - ECMP(동일 비용 다중 경로) 미지원 → 다중 연결에서도 대역폭 증가 불가.
  - Transit Gateway를 활용한 집계 대역폭:
    - ECMP와 동적 라우팅(BGP) 구성.
    - 최대 4개의 터널을 통해 집계 대역폭을 최대 5 Gbps까지 확장 가능.
    - **단일 흐름(Per-flow) 대역폭은 여전히 1.25 Gbps로 제한되나, 다중 파일 전송 시 총 대역폭 증가 가능.**
- **고급 라우팅 설정**
  - BGP 라우팅 및 ECMP를 통해 다중 VPN 연결 간 트래픽 분산.
  - CIDR 범위에 따라 특정 VPN 연결에 트래픽을 전달하도록 경로 설정.
  - VPN 연결 중 하나가 비활성화될 경우, 다른 활성 연결로 트래픽을 재라우팅 가능.
- **결론**
  - Transit Gateway와 VPN 연결을 활용하면 네트워크 아키텍처의 복잡성을 줄이고 대역폭을 확장하며, 하이브리드 연결 환경을 최적화 가능.
  - BGP 라우팅 필수, 정적 라우팅으로는 ECMP 구성 불가.

## TGW with Direct Connect

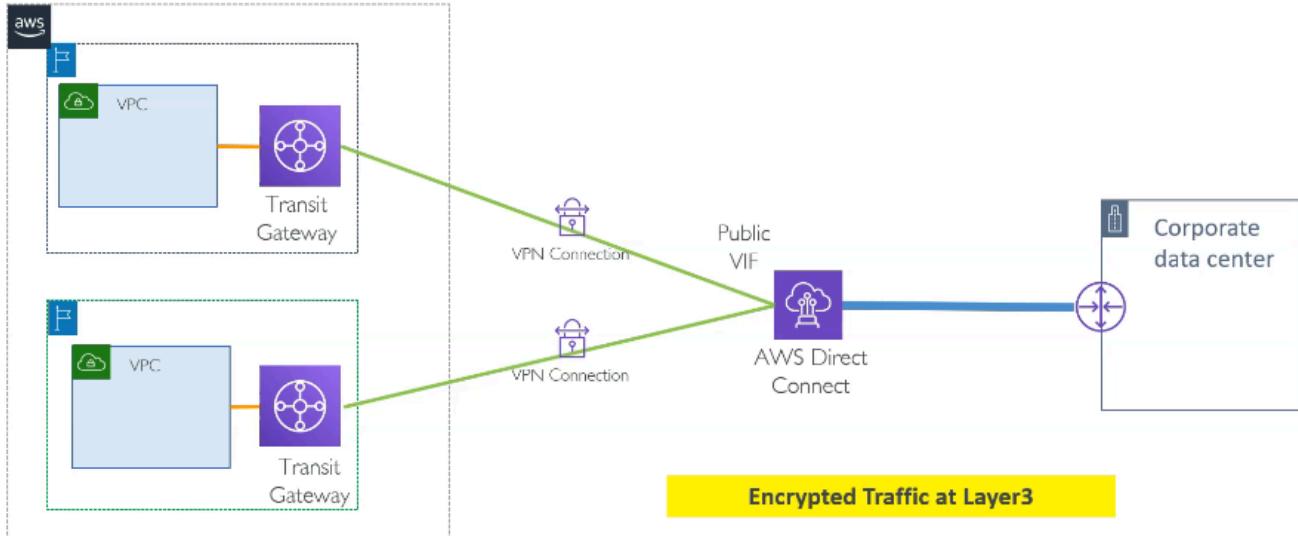
- AWS와 온프레미스 데이터센터 간의 물리적 연결을 제공.
- 안정적이고 빠른 네트워크 연결을 통해 데이터 전송.
- **Transit Gateway 없이 Direct Connect 사용**
  - 일반적으로 **Private VIF(Virtual Interface)**를 사용하여 Direct Connect Gateway에 연결.
  - Direct Connect Gateway는 **최대 10개의 VPC**만 연결 가능.

- 수백 개의 VPC 연결이 필요한 경우 확장성 한계에 직면.
- **Transit Gateway와 함께 Direct Connect 사용**
  - **Transit VIF** 사용:
    - Direct Connect에 Transit VIF를 생성하고 Direct Connect Gateway에 연결.
    - Direct Connect Gateway를 Transit Gateway와 연결하여 다수의 VPC를 지원.
  - **확장성:**
    - 하나의 Transit Gateway는 **수백, 수천 개의 VPC** 연결 가능.
    - 하나의 Direct Connect Gateway는 **최대 6개의 Transit Gateway** 연결 지원.
    - 하나의 Direct Connect 연결에서는 **최대 4개의 Transit VIF** 생성 가능.
    - 결과적으로, 하나의 Direct Connect 연결로 **최대 24개의 Transit Gateway**를 연결 가능.
- **다중 계정 지원**
  - Direct Connect와 Transit Gateway는 서로 다른 AWS 계정에 있을 수 있음:
    - 예: Direct Connect는 계정 A에, Transit Gateway와 VPC는 계정 B에.
- **비용 고려**
  - **Transit Gateway 데이터 처리 요금:**
    - Transit Gateway를 사용할 경우 전송되는 데이터에 대해 **GB당 요금**이 발생.
    - 데이터 전송량이 많을 경우 비용이 높아질 수 있으므로 주의 필요.
    - 대규모 데이터 전송에는 Transit Gateway 사용이 비효율적일 수 있음.
- **보안 강화: IP 레벨 암호화**
  - 기본 Direct Connect 아키텍처는 데이터 트래픽이 **암호화되지 않음**.
  - 일부 기업에서는 Layer 3 또는 Layer 4 암호화를 요구:
    - **IPSec VPN**을 활용하여 트래픽 암호화 가능.
    - **Public VIF**를 생성해 AWS의 Public IP와 연결:
      - Transit Gateway의 VPN 엔드포인트(Public IP)를 통해 IPSec VPN 연결.
      - Direct Connect와 Transit Gateway 간 트래픽 암호화 지원.
- **추천 아키텍처 요약**
  - **확장성과 간소화 필요:**
    - 다수의 VPC 연결 시 Transit Gateway와 Transit VIF 사용.
  - **보안 요구:**
    - 암호화가 필요한 경우 IPSec VPN과 Public VIF 사용.
  - **비용 관리:**
    - 대규모 데이터 트래픽에는 Transit Gateway 사용을 신중히 고려.
- **결론**
  - Transit Gateway는 확장성과 간소화를 제공하지만, **비용과 보안 요구**를 함께 고려해야 함.
  - 특정 요구 사항에 따라 Direct Connect와 Transit Gateway를 결합한 적절한 아키텍처를 선

택.



- 이 경우 L3, L4 Layer는 암호화를 지원하지 않기 때문에 아래의 방법으로 하면 IPsec VPN을 이용해 네트워크를 암호화할 수 있다.



## Multicast with TGW

멀티캐스트란?

항목	설명
멀티캐스트	하나의 메시지를 여러 대상에게 동시에 전송하는 방식.
유니캐스트	한 소스 → 한 대상.
IP 주소	Class D 범위: 224.0.0.0 ~ 239.255.255.255 .
프로토콜	UDP 기반, 단방향 통신.
활용 사례	OTT 플랫폼, TV 방송, 주식 거래, 그룹 채팅.

## AWS Transit Gateway의 멀티캐스트 지원

항목	설명
멀티캐스트 도메인	멀티캐스트 그룹이 작동하는 기본 구성 요소.
ENI	ENI를 그룹에 추가하여 멤버십 구성.
IGMP	동적 그룹 참여 및 탈퇴 지원.
주요 기능	IPv4/IPv6 지원, 외부 애플리케이션 통합 가능.

### 멀티캐스트 설정 절차

단계	설명
Transit Gateway 생성	멀티캐스트 활성화 필요.
멀티캐스트 도메인 생성	서브넷을 도메인에 참여시킴.
멀티캐스트 그룹 생성	ENI 추가 및 Class D IP 주소 지정.
그룹 멤버십 관리	정적(AWS CLI) 또는 동적(IGMP)으로 설정.

### 하이브리드 멀티캐스트 통합

항목	설명
지원 환경	Direct Connect 및 VPN에서 멀티캐스트 미지원.
해결 방법	GRE 터널을 통해 온프레미스와 AWS 연결.
트래픽 전달	온프레미스 → VPC → Transit Gateway → 멀티캐스트 그룹.

### 보안 및 네트워크 ACL 구성

항목	설명
IGMP 트래픽	수신: 224.0.0.1/32 , 발신: 224.0.0.2/32 .
멀티캐스트 트래픽	멀티캐스트 그룹 IP로부터 수신 및 발신 허용.
보안 그룹 및 ACL 설정	IGMP 및 멀티캐스트 트래픽 허용 필요.

NACL Inbound		
Protocol	Source	Description
IGMP(2)	0.0.0.0/32	IGMP query sent to 224.0.0.1/32
UDP	Remote host sending multicast traffic	For inbound multicast traffic sent to multicast group IP

Security Group Inbound		
Protocol	Source	Description
IGMP(2)	0.0.0.0/32	IGMP query
UDP	Remote host sending multicast traffic	For inbound multicast traffic

NACL Outbound		
Protocol	Destination	Description
IGMP(2)	224.0.0.2/32	IGMP Leave
IGMP(2)	Multicast group IP address	IGMP Join
UDP	Multicast group IP address	For outbound multicast traffic

Security Group Outbound		
Protocol	Destination	Description
IGMP(2)	224.0.0.2/32	IGMP Leave
IGMP(2)	Multicast group IP address	IGMP Join
UDP	Multicast group IP address	For outbound multicast traffic

## 추가 멀티캐스트 고려 사항

항목	설명
서브넷 제한	한 서브넷은 하나의 멀티캐스트 도메인만 참여 가능.
ENI 그룹 멤버십	동일 도메인 내 여러 멀티캐스트 그룹에 참여 가능.
IGMPv2 지원	Join/Leave 메시지로 그룹 동적 참여/탈퇴 지원.
StaticSourcesSupport	특정 멤버만 메시지 송신자로 지정 가능.
Nitro 제한	Non-Nitro 인스턴스는 송신 불가. 수신 시 Source/Destination 체크 비활성화 필요.

## 제약 사항

항목	설명
멀티캐스트 라우팅 제한	Direct Connect, Site-to-Site VPN, Peering 등 미지원.
멀티캐스트 공유	AWS RAM(Resource Access Manager)로 도메인 공유 가능.

## 시험 대비 주요 사항

항목	설명
IGMPv2 프로토콜	동적 멤버십 지원.
VPC 및 계정 간 통합 가능	멀티캐스트 도메인 공유 가능.
하이브리드 통합	GRE 터널 필요.

# 결론

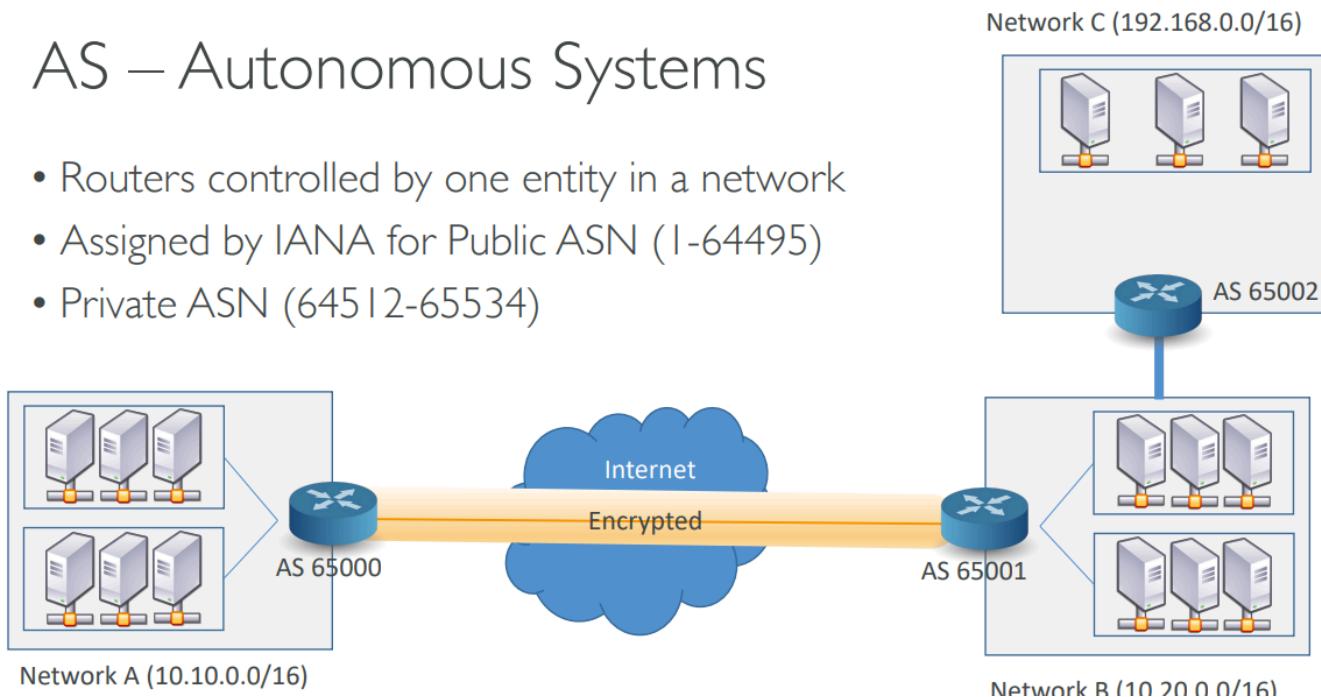
AWS Transit Gateway의 멀티캐스트 기능은 대규모 트래픽 배포를 지원하며, VPC 간 통신 및 하이브리드 환경 통합을 통해 효율적인 네트워크 아키텍처를 제공합니다. 🚀

## 9. Hybrid Network Basic

### VPN Routing - Static vs Dynamic

#### AS – Autonomous Systems

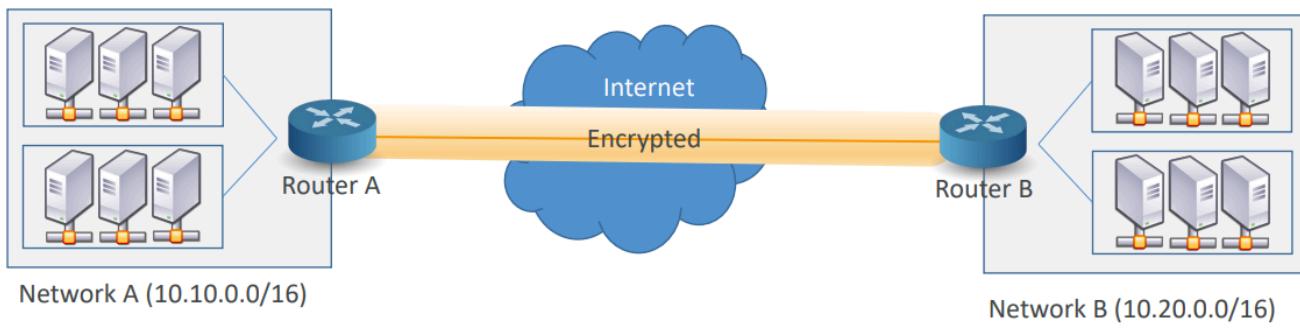
- Routers controlled by one entity in a network
- Assigned by IANA for Public ASN (1-64495)
- Private ASN (64512-65534)



- 네트워크 고유 식별자를 **AS**라 부르며 다른 네트워크와 경로를 교환할 수 있는 모든 라우터다.
- 식별자의 ID를 **ASN**이라 부른다.
- 공용ASN이라면 IANA에서 부여한다. (**1 - 64495**)
- 다른 네트워크간 통신이 되길 원한다면 사설ASN도 사용할 수 있다 (**64512 - 65534**)
- 사설 연결로 구성하면 BGP도 사용할 수 있다.

### Static Routing (고정 라우팅)

Network A		Network B	
Destination	Target	Destination	Target
10.10.0.0/16	Local	10.20.0.0/16	Local
10.20.0.0/16	Router B	10.10.0.0/16	Router A



## Static Routing

Network A		Network B	
Destination	Target	Destination	Target
10.10.0.0/16	Local	10.20.0.0/16	Local
10.20.0.0/16	Router B	10.10.0.0/16	Router A
		10.30.0.0/16	Router C

The diagram shows a more complex static routing setup. Network A (10.10.0.0/16) has two server groups connected to Router A. Network B (10.20.0.0/16) has two server groups connected to Router B. Router A and Router B are connected to the "Internet" cloud. Router C is also connected to the "Internet" cloud and is connected to Router B. The connection between Router A and Router B is highlighted in yellow and labeled "Encrypted".

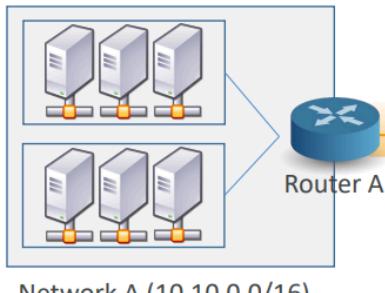
- 단점 : 만일 A -> C로 보내고 싶은데, A는 C가 어딘지 모르므로 트래픽을 Drop해버릴 수 있다.
- 해결 : A Route Table에서 10.30.0.0/16을 Router C로 수동 등록해야 함.

## Dynamic Routing (동적 라우팅)

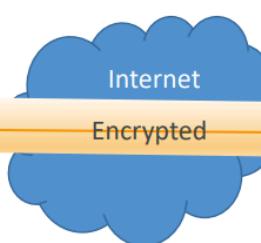
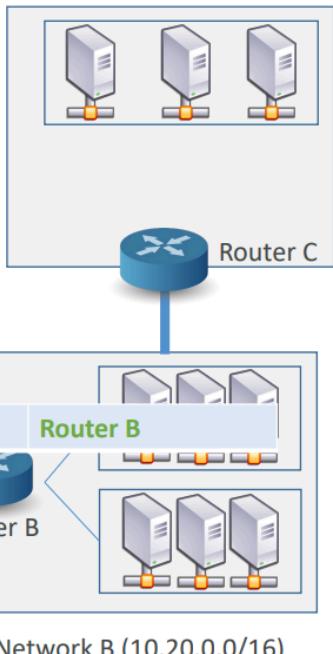
# Dynamic Routing

Network A	
Destination	Target
10.10.0.0/16	Local
10.20.0.0/16	Router B

In dynamic routing the routes get propagated automatically



Network C (10.30.0.0/16)



10.30.0.0/16

Encrypted



Router B

Router A

Network B (10.20.0.0/16)

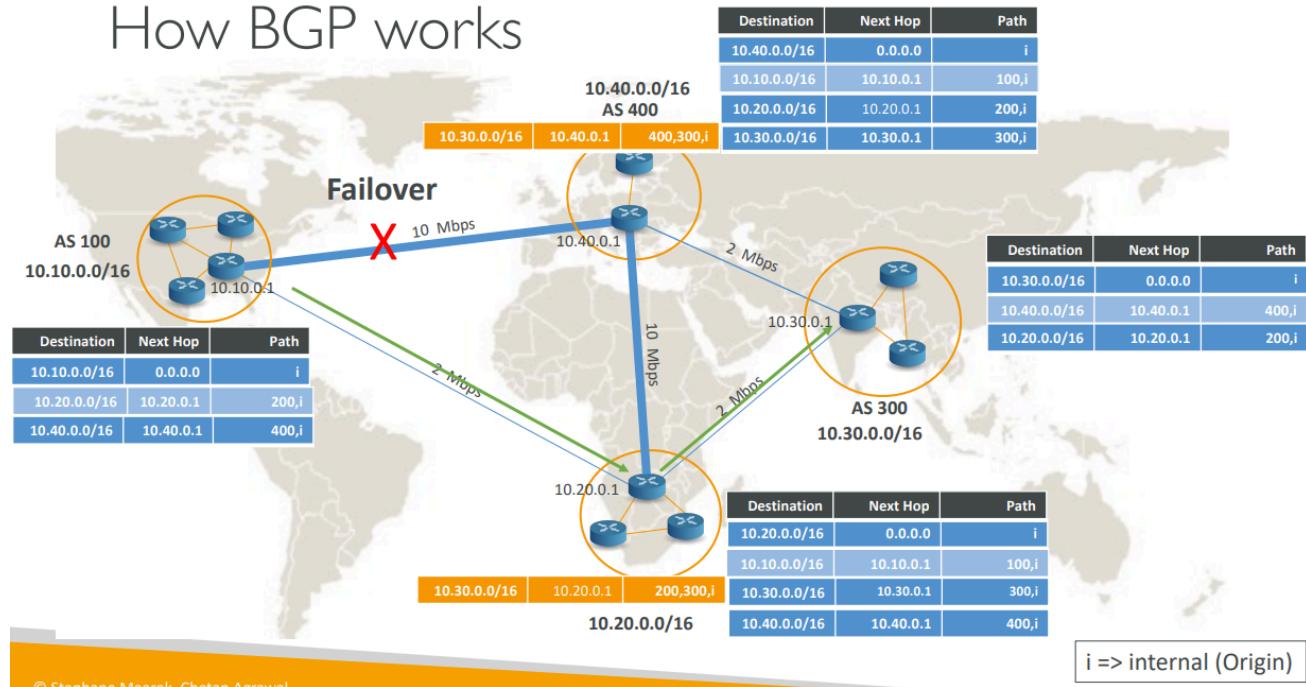
- 네트워크 B에서 C가 연결된 내용이 확인되면 **자동으로 A 네트워크에 프로토콜을 통해 공유 한다.**

## BGP (Border Gateway Protocol)

- Path-Vector 프로토콜을 사용하는 동적 라우팅에서는 피어(peer) 또는 자율 시스템(AS, Autonomous System) 간에 목적지로 가는 최적 경로를 교환합니다.
  - iBGP:** AS 내에서의 라우팅.
  - eBGP:** AS 간의 라우팅.
- 라우팅 결정은 다음에 의해 영향을 받습니다:
  - Weight:** Cisco 라우터에서 사용하는 AS 내에서만 작동하는 값.
  - ASPATH:** 경로를 따라 이동해야 하는 AS들의 순서 (AS 간에서 작동).
  - Local Preference (LOCAL\_PREF):** AS 내에서 작동.

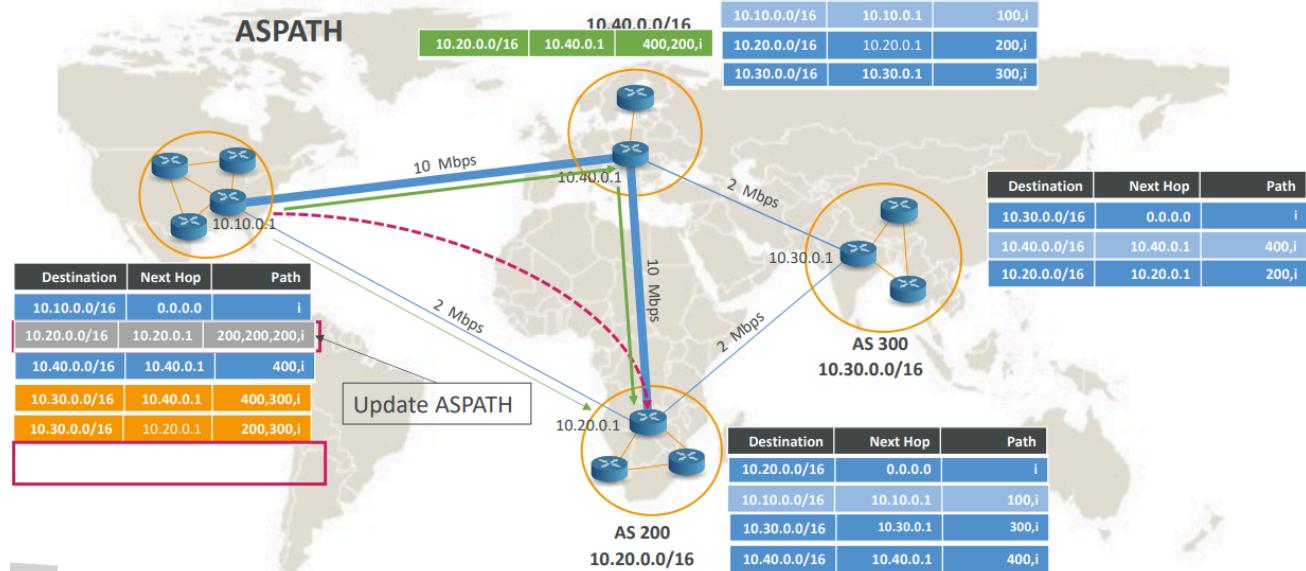
- MED (Multi-Exit Discriminator): AS 간에서 작동.

## How BGP works



- 모든 AS에는 라우터 테이블이 있고 **BGP 테이블**이라고 불리운다.
- BGP에서 통신 경로의 순위를 매길때 목적지, next hop, 경로를 선택해야 하는데 이를 **PATH**라 한다.
- 표에 나와 있는 Path에서 보면 ASN,i (internal) 형태로 Path가 추가된다.
- BGP가 하는 일은 **BGP 이웃들이** 경로를 이해하거나 학습하거나 해서 다른 AS에 정보를 전달하는 것.

## How BGP works

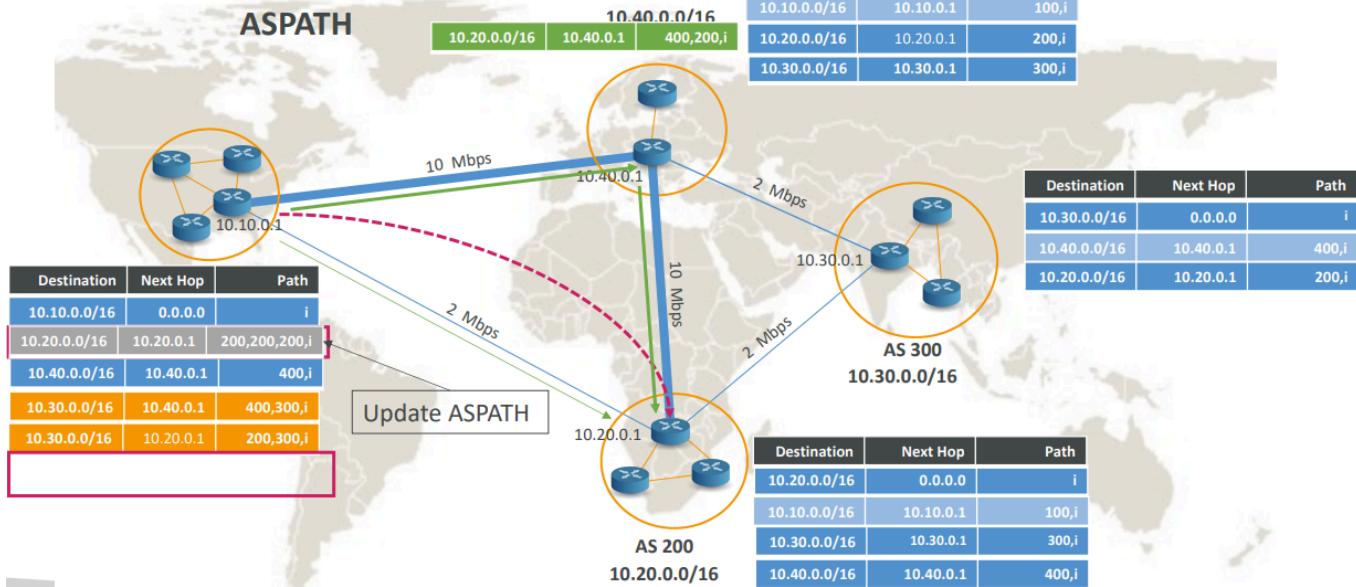


- 그래서 경로 학습이 끝나면 400, 300,i 와 200,300,i 가 선택된다.
- 만일 100 -> 200으로의 통신이 다운되었다고 가정하면
  - AS 100 -> 300으로 갈때 400,300,i가 연결 경로가 된다.
  - 통신 장애 극복에 이용된다.**

# BGP Route Selection (ASPATH, LOCAL\_PREF, MED)

## 1. AS\_PATH

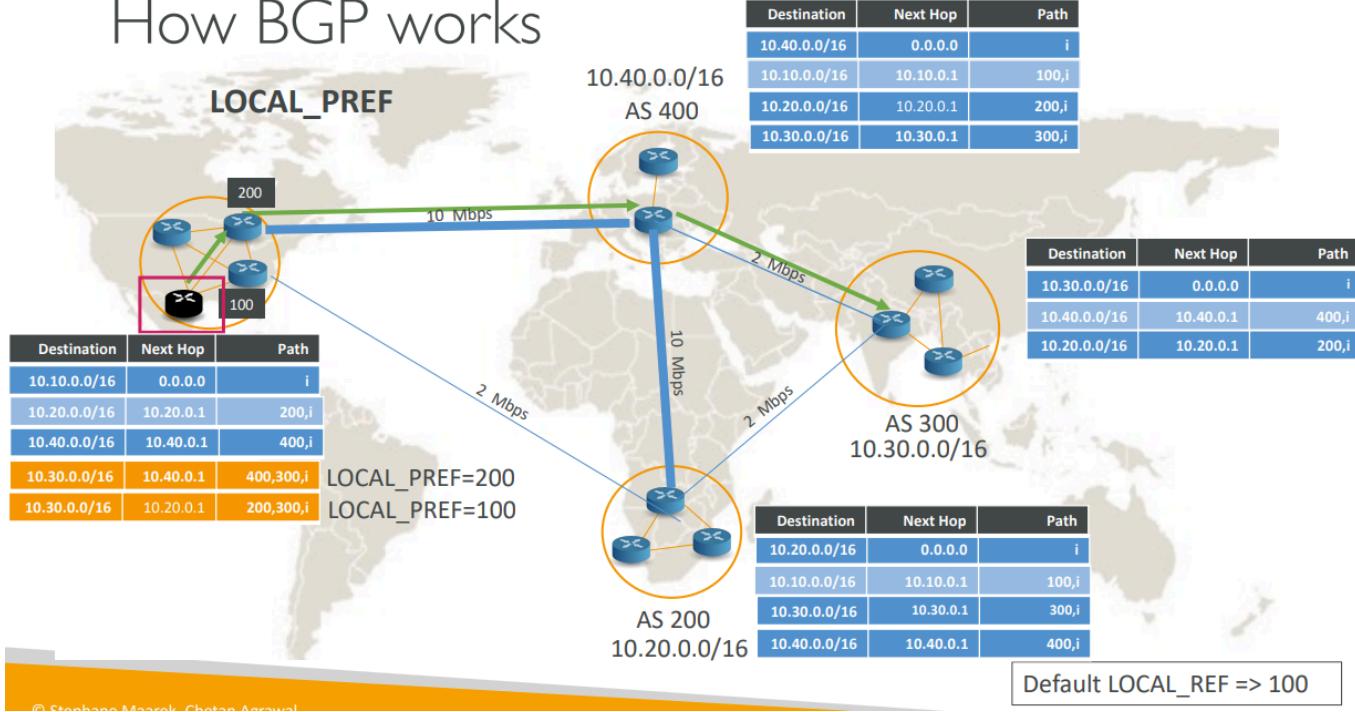
### How BGP works



- 역할:** AS 간의 최단 경로를 계산하며, 짧은 AS\_PATH 값을 가진 경로가 더 선호됨.
- 동작 원리:**
  - 각 AS는 BGP 업데이트 메시지에 자신을 AS\_PATH에 추가.
  - AS\_PATH는 목적지 Network까지의 경로를 나타냄.
- 속도를 빠르게 하는 기술**
  - AS\_PATH Prepending**을 이용하면 강제로 흡을 더 끼워 넣어서 네트워크 경로를 조절 할 수 있다.
  - 예시
    - 100 -> 200 (2Mbps), AS\_PATH 더 짧아서 기본적으로 우선순위 높음 (200,i)
    - 100 -> 400 -> 200 (10Mbps) 도 방법이지만, AS\_PATH상에선 우선순위 낮음 (400,200,i)
    - AS\_PATH Prepending을 이용해서 강제로 200에 링크를 삽입 (200,200,200,i)
    - 100 -> 200 연결 시도시 길이가 길어져 100 -> 400 -> 200이 더 우선시 된다.

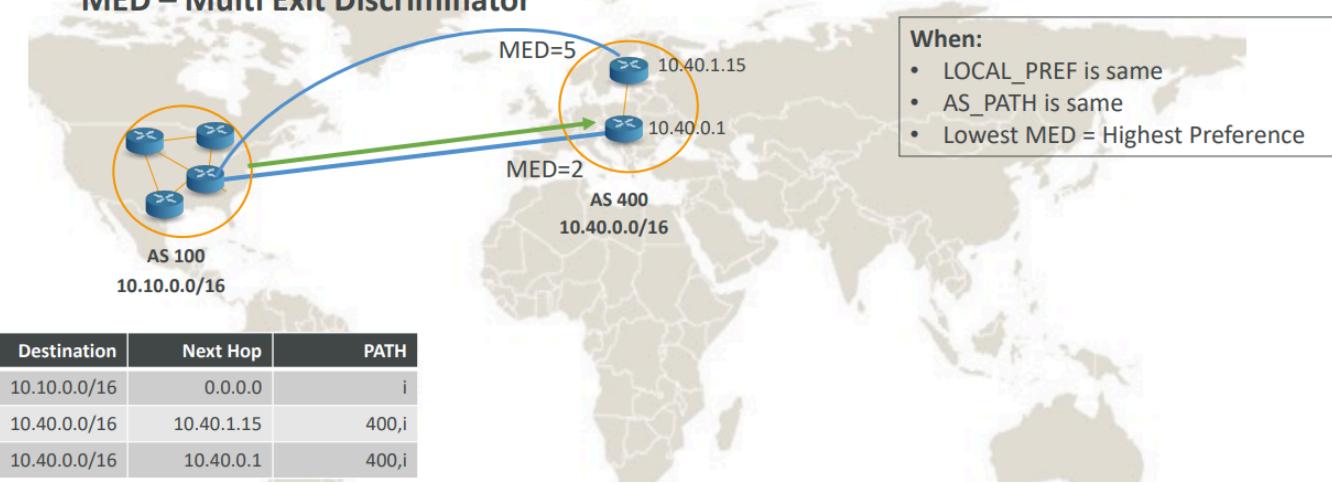
## 2. LOCAL\_PREF

# How BGP works



- 정의:** BGP에서 AS 내부에서 사용되는 경로 선택 기준으로, 가장 높은 Local Preference 값이 우선.
  - 역할:** AS 내부의 트래픽 경로를 제어.
  - 적용 범위:** AS 내부에서만 유효하며, 다른 AS로 전파되지 않음.
  - 기본값:** Local Preference 값은 기본적으로 100으로 설정됨.
  - 그림 설명**
    - AS 200 (10.20.0.0/16)**
      - Local Preference = 200 (우선 경로)
      - 경로 10.20.0.1을 통해 외부 네트워크로 연결.
    - AS 400 (10.40.0.0/16)**
      - Local Preference = 100 (낮은 우선순위)
      - 경로 10.40.0.1 사용.
    - AS 300 (10.30.0.0/16)**
      - AS 내부에서 두 개의 외부 경로를 통해 다른 네트워크로 연결:
        - AS 200 → Local Preference = 200 (우선).
        - AS 400 → Local Preference = 100 (후순위).
- ### 3. MED (Multi Exit Discriminator)

## MED – Multi Exit Discriminator



- **역할:** AS 간의 경로 선택에서 우선순위를 부여하며, 값이 낮을수록 높은 우선순위를 가짐.
- **적용 범위:** BGP의 선택 과정에서 **Local Preference**, **AS Path**가 동일한 경우 적용됨.
- **동작 방식:** AS 간 경로에 대해 특정 진입점을 우선시하거나, 트래픽이 특정 링크를 통해 유입 되도록 경로 선택을 유도.
- **그림 예제 설명**

### 1. 네트워크 구성:

- AS 100 (10.10.0.0/16)과 AS 400 (10.40.0.0/16)이 두 개의 BGP 경로로 연결됨.
- AS 400은 두 개의 라우터(10.40.1.15와 10.40.0.1)를 통해 AS 100으로 연결.
- **MED 값:**
  - 10.40.1.15 → MED=5
  - 10.40.0.1 → MED=2 (더 낮은 값)

### 2. BGP 경로 선택 과정:

- **Local Preference:** 동일 (그림에서 LOCAL\_PREF가 같음).
- **AS\_PATH 길이:** 동일 (둘 다 "400" 경로 사용).
- **MED 값 비교:**
  - 10.40.0.1 (MED=2)이 10.40.1.15 (MED=5)보다 낮으므로 더 높은 우선순위를 가짐.

### 3. 결과:

- AS 100은 10.40.0.1을 통해 AS 400으로 트래픽을 보냄.
- **경로:** 10.40.0.1 이 선호되므로 초록색 경로를 통해 트래픽이 유입.

특성	AS_PATH	LOCAL_PREF	MED
정의	경로를 거친 AS의 순서를 나타냄	AS 내부에서 경로 우선순위를 결정	AS 외부에서 들어오는 경로의 우선순위를 결정
주 사용 목적	경로 길이에 따라 짧은 경로를 선호	AS 내부 트래픽 흐름을 최적화	외부 AS로부터 들어오는 트래픽의 선호 경로

특성	AS_PATH	LOCAL_PREF	MED
			지정
기본값	각 경로에 대해 자동 업데이트	기본값 = 100	기본값 = 0
조작 가능 여부	<b>Prepending</b> 으로 길이 조작 가능	사용자가 원하는 값으로 조정 가능	낮은 값으로 조정 가능
적용 범위	AS 간 전파	AS 내부에서만 유효	다른 AS로 전파되지 않음
선호 경로 결정	경로 길이가 짧을수록 선호	값이 높을수록 선호	값이 낮을수록 선호
루프 방지 역할	루프 감지 및 제거	역할 없음	역할 없음
설정 위치	온프레미스 또는 BGP 라우터에서 설정	온프레미스 또는 BGP 라우터에서 설정	온프레미스 또는 BGP 라우터에서 설정
AWS 내 설정 여부	직접 설정 불가 (온프레미스에서만 가능)	직접 설정 불가 (온프레미스에서만 가능)	직접 설정 불가 (온프레미스에서만 가능)
일반적인 사용 사례	짧은 경로를 선호하도록 조정	특정 경로 우선 지정	외부에서 낮은 값을 부여해 경로 선호 조정

## BGP Route Selection 순서와 속성

### Highest Weight

- Cisco 라우터에서 사용하는 특정 속성
- AS 내부에서만 작동하며, 경로를 생성한 라우터에 의해 설정됨
- 외부 BGP 라우터 간에는 교환되지 않음

### 1. Local Preference

- 외부 BGP(eBGP) 경로 중 아웃바운드 경로 선택
- 내부 BGP(iBGP) 라우터 간 우선순위 설정
- 외부 BGP 라우터 간에는 교환되지 않음
- 기본값은 100

### 2. AS Path

- AS 경로가 가장 짧은 경로를 선호
- AS Path는 AS Path Prepend를 사용하여 더 길게 만들어 우선순위를 낮출 수 있음

### 3. Multi Exit Discriminator (MED)

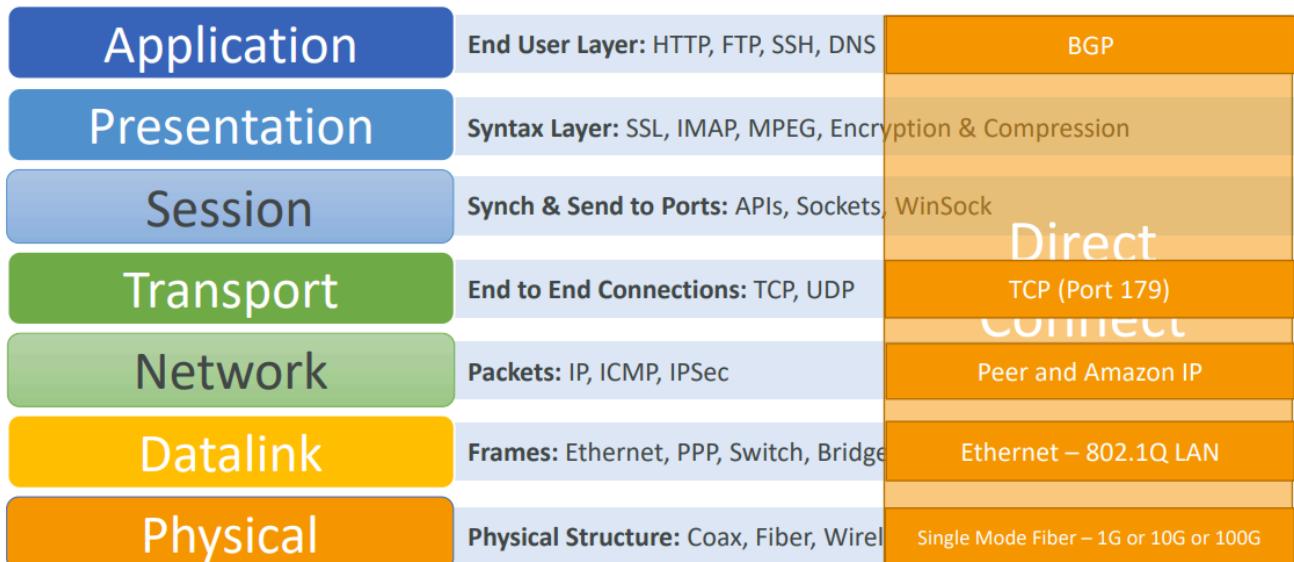
- 두 AS 간 여러 경로가 있을 때 사용
- MED 값이 낮을수록 더 높은 우선순위를 가짐
- AS 간에 MED 값이 교환됨

## 10. AWS Direct Connect (DX)

### OSI 7 Layer

- Application : End User Layer (http, ftp, ssh, dns)
- Presentation : Syntax Layer (SSL, IMAP, MPEG, Encryption & Compression)
- Session : Sync & Send to Ports : APIs, Socket, WinSock
- Transport : End to End Connection (TCP, UDP)
- Network : Packets (IP, ICMP, IPSec)
- Datalink : Frames (Ethernet, PPP, Switch, Bridge)
- Physical : Physical Structure (Coax, Fiber, Wireless, Hubs, Repeaters)
- 외우기 쉬운 방법?
  - "응용 프로그램에서 표준화된 세션을 통해 전송된 네트워크 데이터는 물리적으로 전달된다."
  - "All People Seem To Need Data Processing"

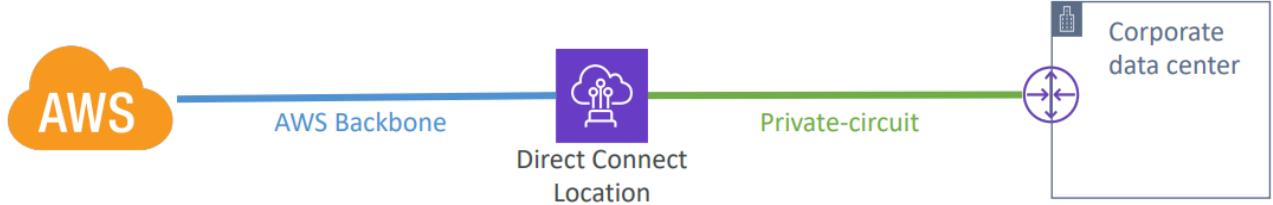
Direct Connect & OS



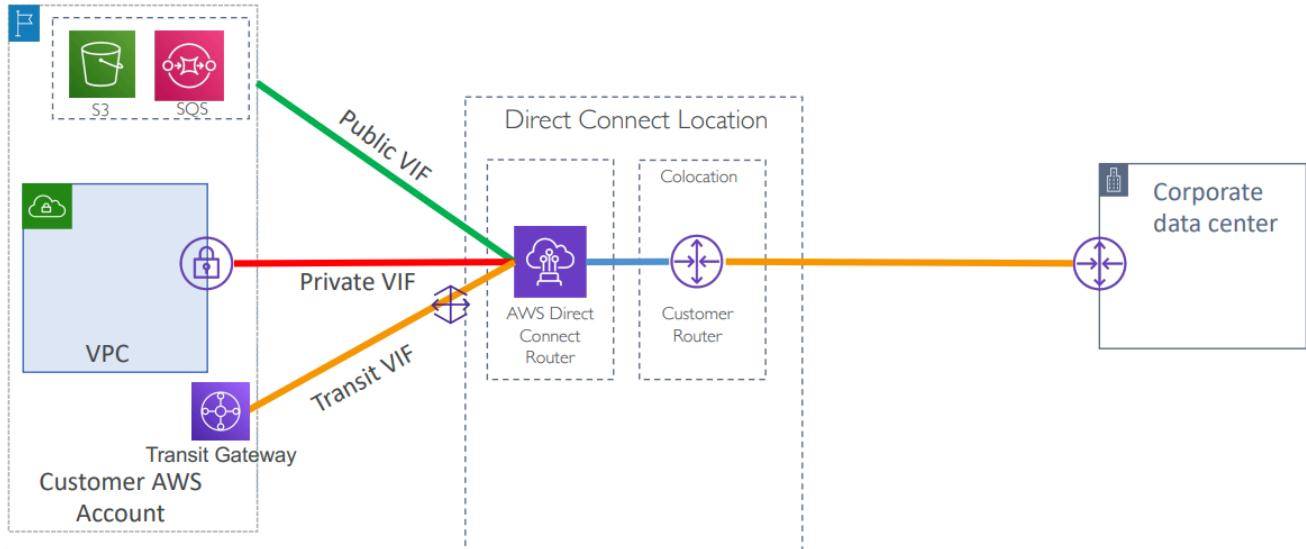
### 기본 설명

- 온프레미스에서 AWS로의 전용 네트워크 연결
- AWS <-> DirectConnect 위치 <-> 온프레미스 데이터 센터

- 낮은 지연 시간과 일관된 대역폭
- 데이터 전송 비용 절감
  - 모든 AWS Direct Connect 위치에서 데이터 전송은 기가바이트당 0.00 USD입니다.
  - Direct Connect에서 다른 리전으로 통신할 경우 요금이 부과될 수 있음. 하지만 igw, nat gw보단 저렴.
  - <https://aws.amazon.com/ko/directconnect/pricing/>
- AWS 프라이빗 네트워크(VPC) 및 AWS 퍼블릭 서비스 엔드포인트(예: S3, DynamoDB) 접근



- AWS 글로벌 네트워크 백본을 활용
- DX(Direct Connect) 위치는 신뢰할 수 있는 제3자 제공업체가 제공 -> Equinix, Cdfi...
- 전 세계 31개 AWS 리전에 걸쳐 **115개의 DX 위치** 보유 (현재 기준)
- 종단 간 프로비저닝 시간은 **4~12주** 소요
- 전용 연결(Dedicated connection)을 통해 **1, 10 또는 100 Gbps** 대역폭 제공
- AWS Direct Connect APN 파트너를 활용하여 **1Gbps 미만의 대역폭(50/100/200/400/500 Mbps, 1, 10 Gbps)** 제공



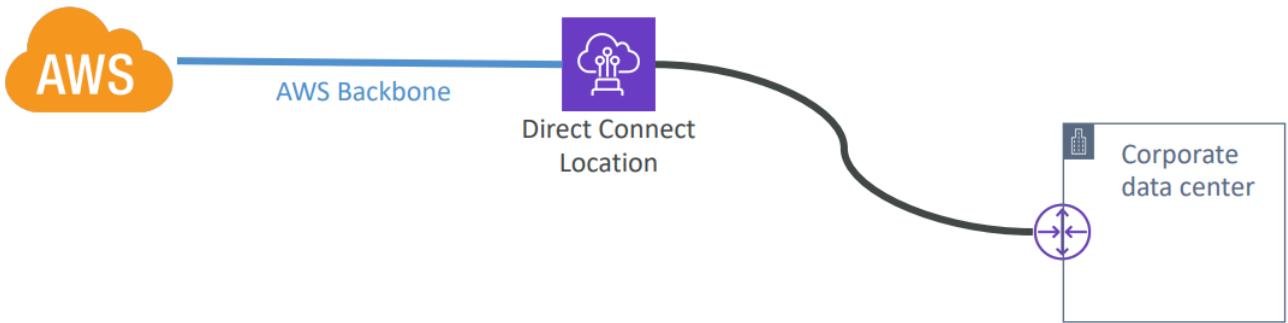
## Direct Connect Requirement

- 요약
  - 싱글 모드 광케이블(Single-mode fiber)
    - 1Gbps 용 1000BASE-LX (1310 nm) 트랜시버
    - 10Gbps 용 10GBASE-LR (1310 nm) 트랜시버

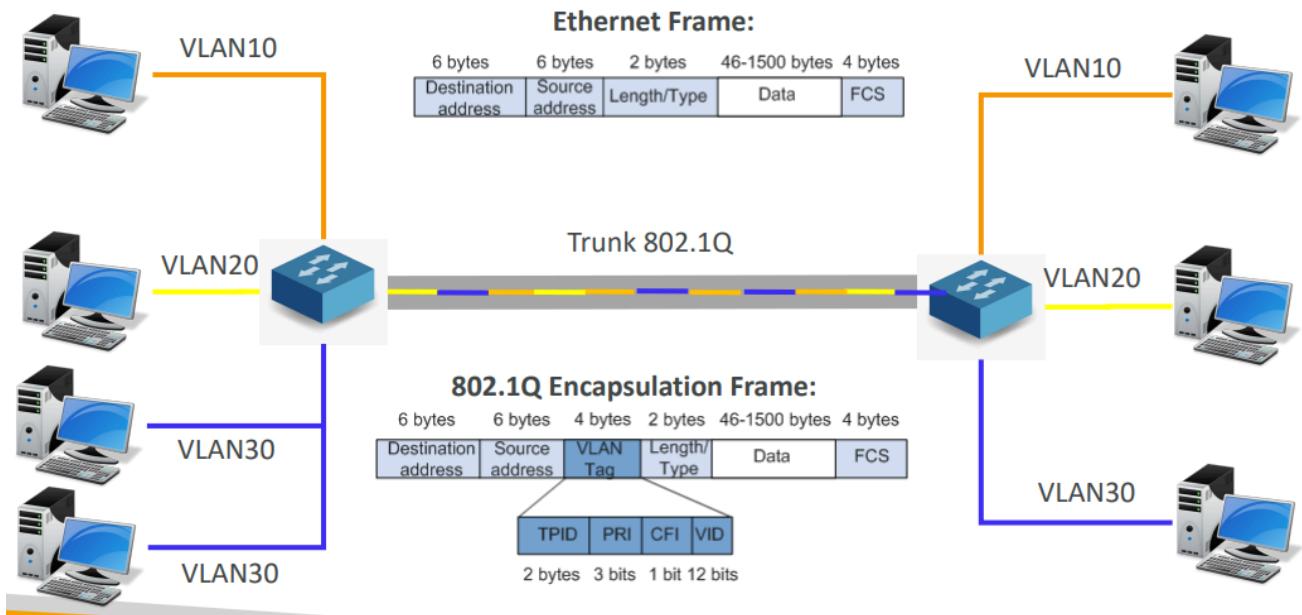
- 100Gbps 용 100GBASE-LR4 트랜시버
- 802.1Q VLAN 캡슐화를 지원해야 함
- 1Gbps 이상의 포트 속도의 경우, 포트 자동 협상(Auto-negotiation)을 비활성화해야 함
- 고객 측 라우터(온프레미스)는 BGP(Border Gateway Protocol) 및 BGP MD5 인증을 지원해야 함
- (선택 사항) 양방향 전달 감지(Bidirectional Forwarding Detection, BFD)

## Direct Connect Requirement 상세

- 싱글 모드 광케이블(Single-mode fiber)
  - 1Gbps 용 1000BASE-LX (1310 nm) 트랜시버
  - 10Gbps 용 10GBASE-LR (1310 nm) 트랜시버
  - 100Gbps 용 100GBASE-LR4 트랜시버

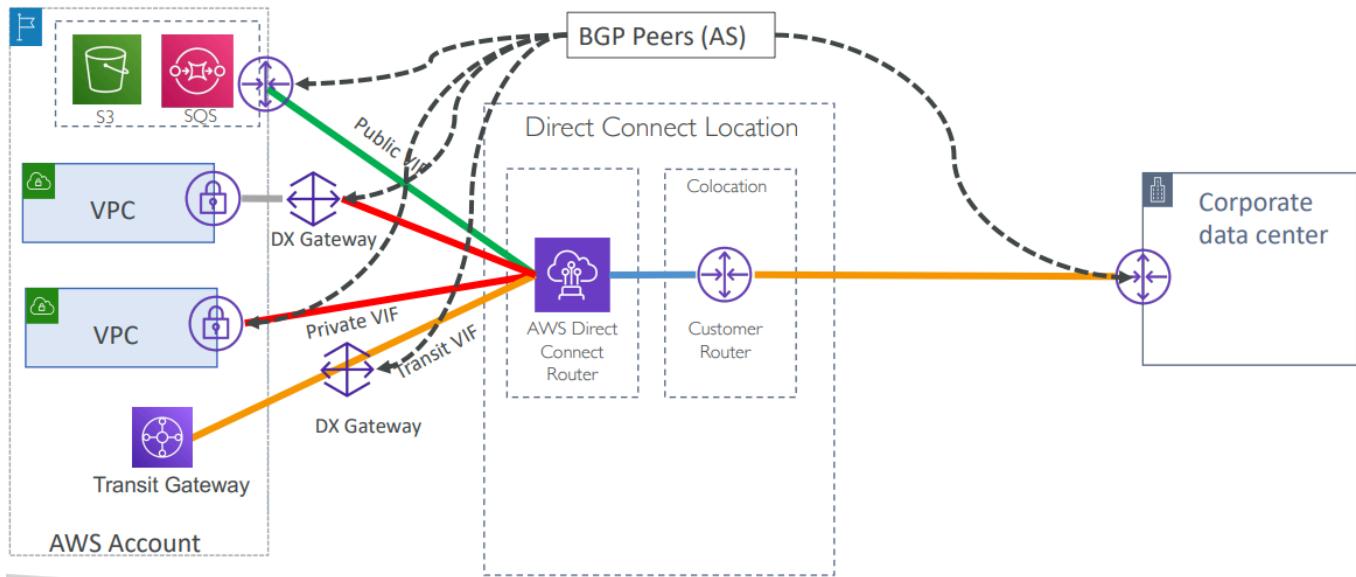


- 802.1Q VLAN 캡슐화를 지원해야 함



- 1Gbps 이상의 포트 속도의 경우, 포트 자동 협상(Auto-negotiation)을 비활성화해야 함
  - Auto-negotiation은 네트워크 인터페이스가 다른 네트워크 인터페이스와 자동으로 연결 매개변수(속도와 듀플렉스 모드)를 조정하는 기능입니다.
  - **속도(Speed):** 네트워크 연결의 전송 속도 (예: 10Mbps, 100Mbps, 1Gbps 등)
  - **듀플렉스 모드(Duplex):** 데이터 전송 방향 (예: 반이중(Half-duplex), 전이중(Full-duplex))
  - 자동 협상은 두 장치가 최적의 속도와 듀플렉스 모드를 자동으로 선택할 수 있도록 도와 줍니다.
  - 1Gbps 미만의 경우, 라우터 공급 업체에 따라 상이함.
- 고객 측 라우터(온프레미스)는 BGP(Border Gateway Protocol) 및 BGP MD5 인증을 지원 해야 함
  - 경로-벡터 프로토콜(Path-Vector protocol)을 사용한 **동적 라우팅(Dynamic Routing)**은 피어(peer) 또는 자율 시스템(AS) 간에 목적지까지의 최적 경로를 교환합니다.
  - **TCP 기반 프로토콜로, 포트 179**를 사용합니다.
  - **iBGP:** 동일한 AS 내부에서의 라우팅
  - **eBGP:** 서로 다른 AS 간의 라우팅
  - 네트워크 경로 선택은 **BGP 라우팅 매개변수**에 의해 영향을 받습니다.
    - **AS\_PATH:** 경로 상의 AS 목록 (AS 간에 작동)
    - **LOCAL\_PREF(로컬 선호도):** 동일한 AS 내에서의 경로 우선순위 결정
    - **MED(Multi-Exit Discriminator):** AS 간의 경로 선택에 사용
- (선택 사항) 양방향 전달 감지(Bidirectional Forwarding Detection, BFD)
  - 간단한 Hello 네트워크 프로토콜
  - 이웃 피어(neighboring peers) 간의 네트워크 장애 감지 시간을 단축합니다.
  - 1초 미만의 빠른 장애 감지 시간을 제공합니다.

# BGP AS & ASN



- 이게 맞는 걸까?
  - On Premise -> AWS Public Service : BGP Peer가 다이렉트로 붙는다.
  - Direct Connect Gateway가 연결된 경우 Direct Connect Gateway로 연결된다.
  - Direct Connect Gateway 없이 연결하면 BGP Peer가 다이렉트로 붙는다.

## Direct Connection Type

### 전용 연결(Dedicated Connections):

- 1Gbps, 10Gbps, 100Gbps 용량
- 고객 전용 물리적 이더넷 포트 제공
- AWS에 먼저 요청한 후, AWS Direct Connect 파트너가 완료
- 고객의 네트워크 제공자 또는 AWS Direct Connect 파트너가 설정 가능

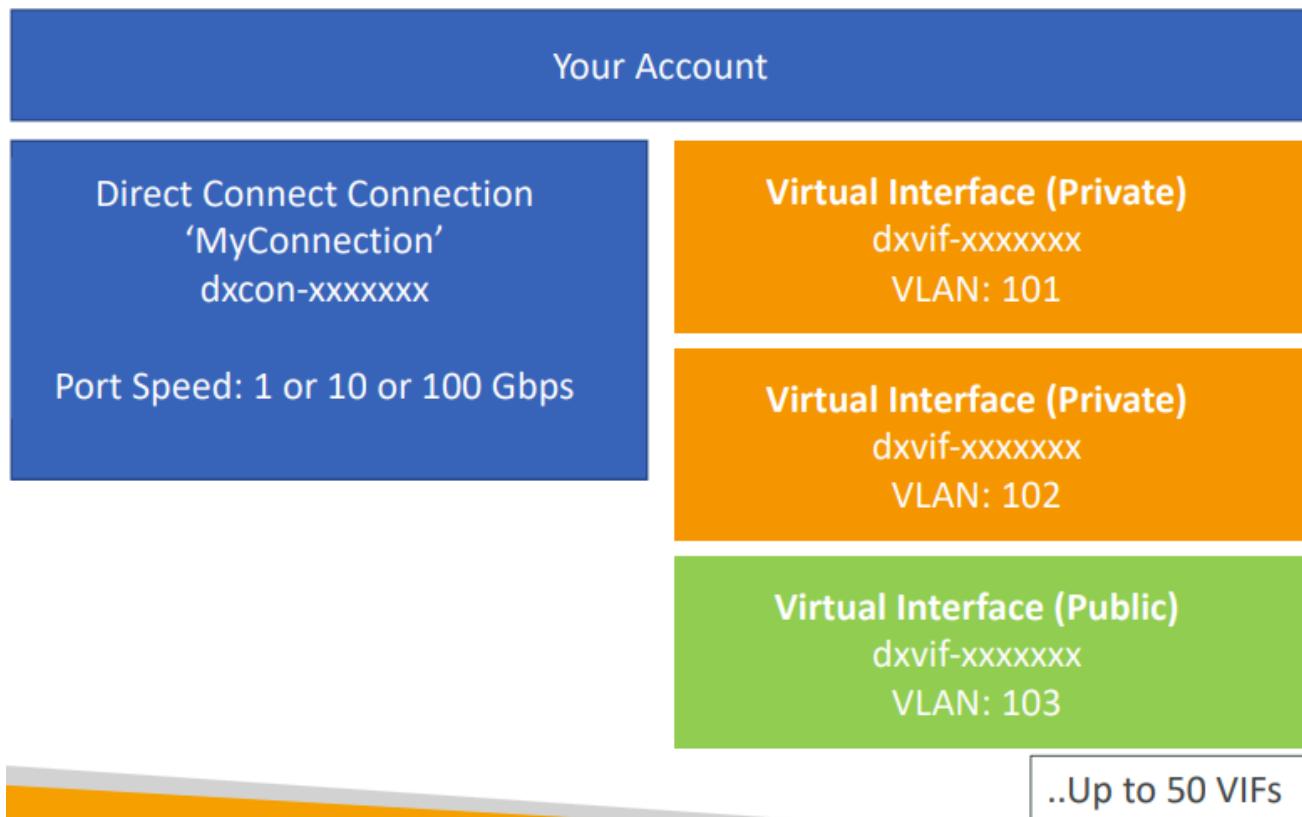
### 호스팅 연결(Hosted Connections):

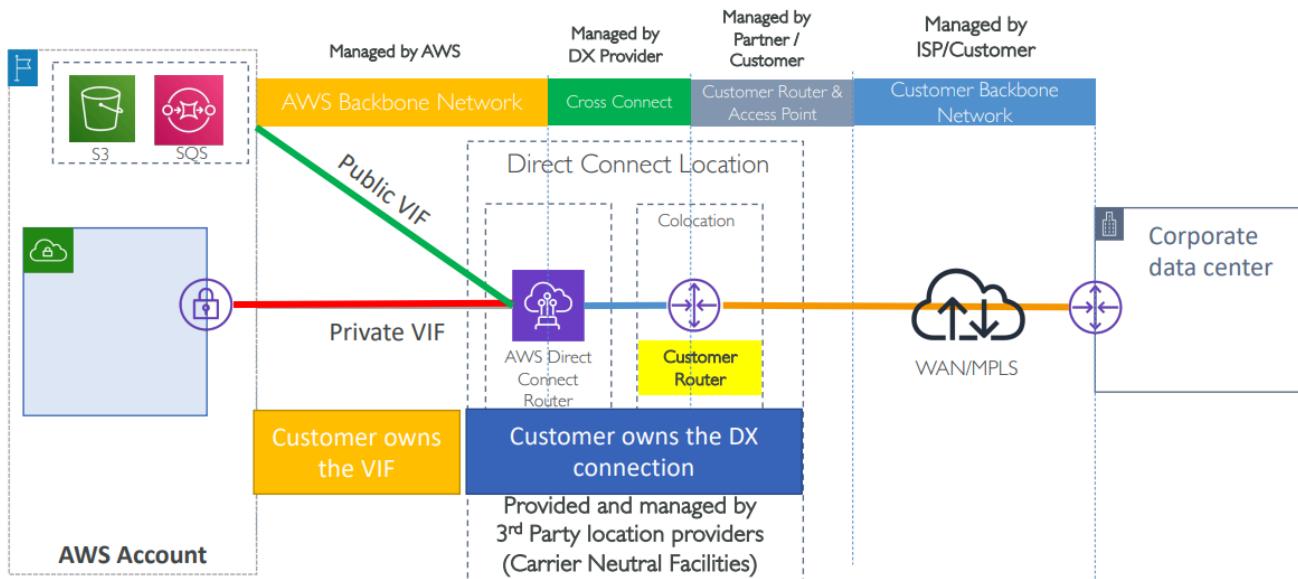
- 50, 100, 200, 300, 400, 500 Mbps 및 1Gbps, 2Gbps, 5Gbps, 10Gbps 용량
- 연결 요청은 AWS Direct Connect 파트너를 통해 이루어짐
- 1, 2, 5, 10Gbps는 일부 AWS Direct Connect 파트너에서만 제공
- AWS는 호스팅 연결에 트래픽 폴리싱(traffic policing)을 적용하여, 초과 트래픽은 폐기(dropped)됨

항목	전용 연결 (Dedicated Connections)	호스팅 연결 (Hosted Connections)
용량	1Gbps, 10Gbps, 100Gbps	50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, 500Mbps, 1Gbps, 2Gbps, 5Gbps, 10Gbps
포트 유형	고객 전용 물리적 이더넷 포트	AWS Direct Connect 파트너가 제공하는 공유 포트
요청 및 설정	AWS에 요청 후, AWS Direct Connect 파트너가 완료	AWS Direct Connect 파트너를 통해 요청 및 설정
설정 주체	고객의 네트워크 제공자 또는 AWS Direct Connect 파트너	AWS Direct Connect 파트너
대역폭 가용성	모든 AWS Direct Connect 위치에서 사용 가능	일부 AWS Direct Connect 파트너에서만 사용 가능
트래픽 관리	트래픽 폴리싱 없음	트래픽 폴리싱 적용 (초과 트래픽은 폐기됨)
적합한 사용 사례	높은 대역폭과 전용 리소스가 필요한 경우	낮은 또는 중간 대역폭이 필요하고 유연성이 요구되는 경우

## Dedicated Connection Structure

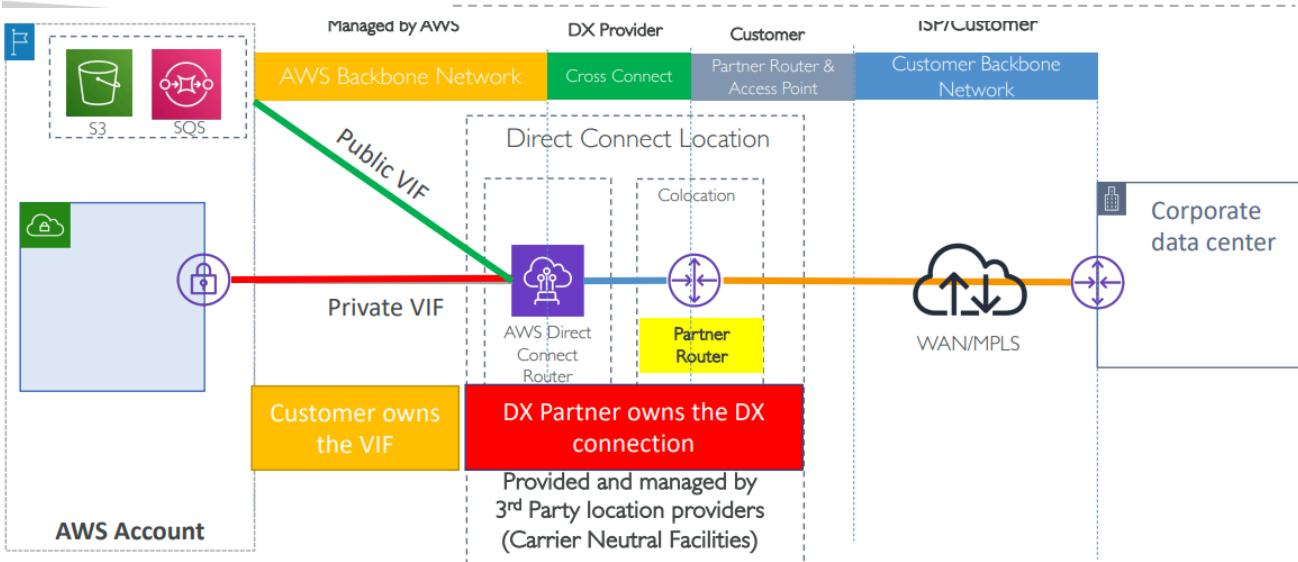
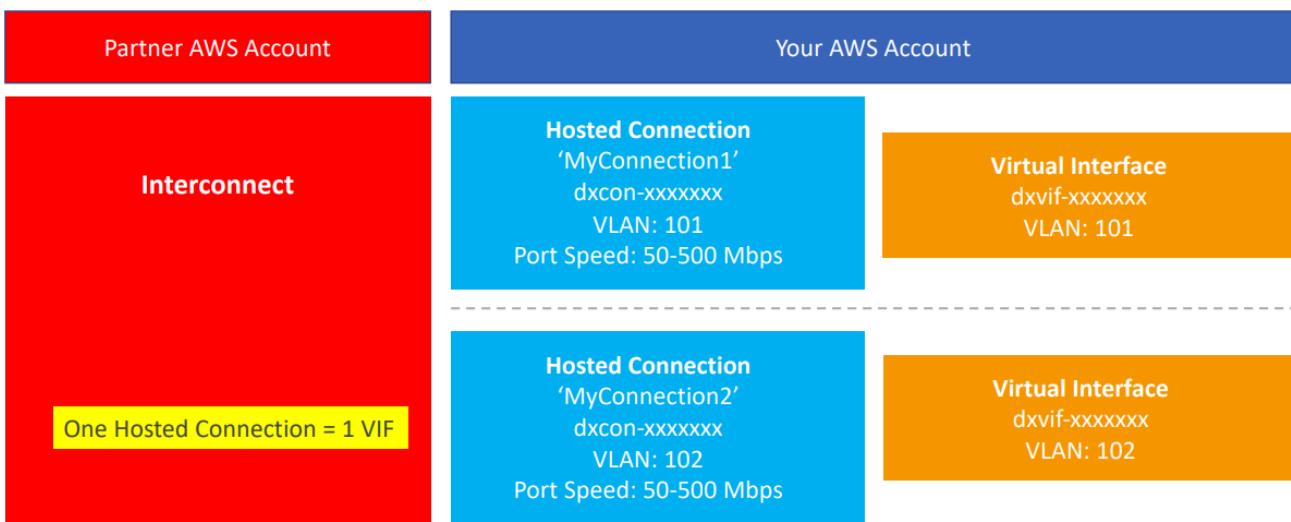
- 커넥션 생성 후 최대 50개까지의 VIF를 연결할 수 있음.

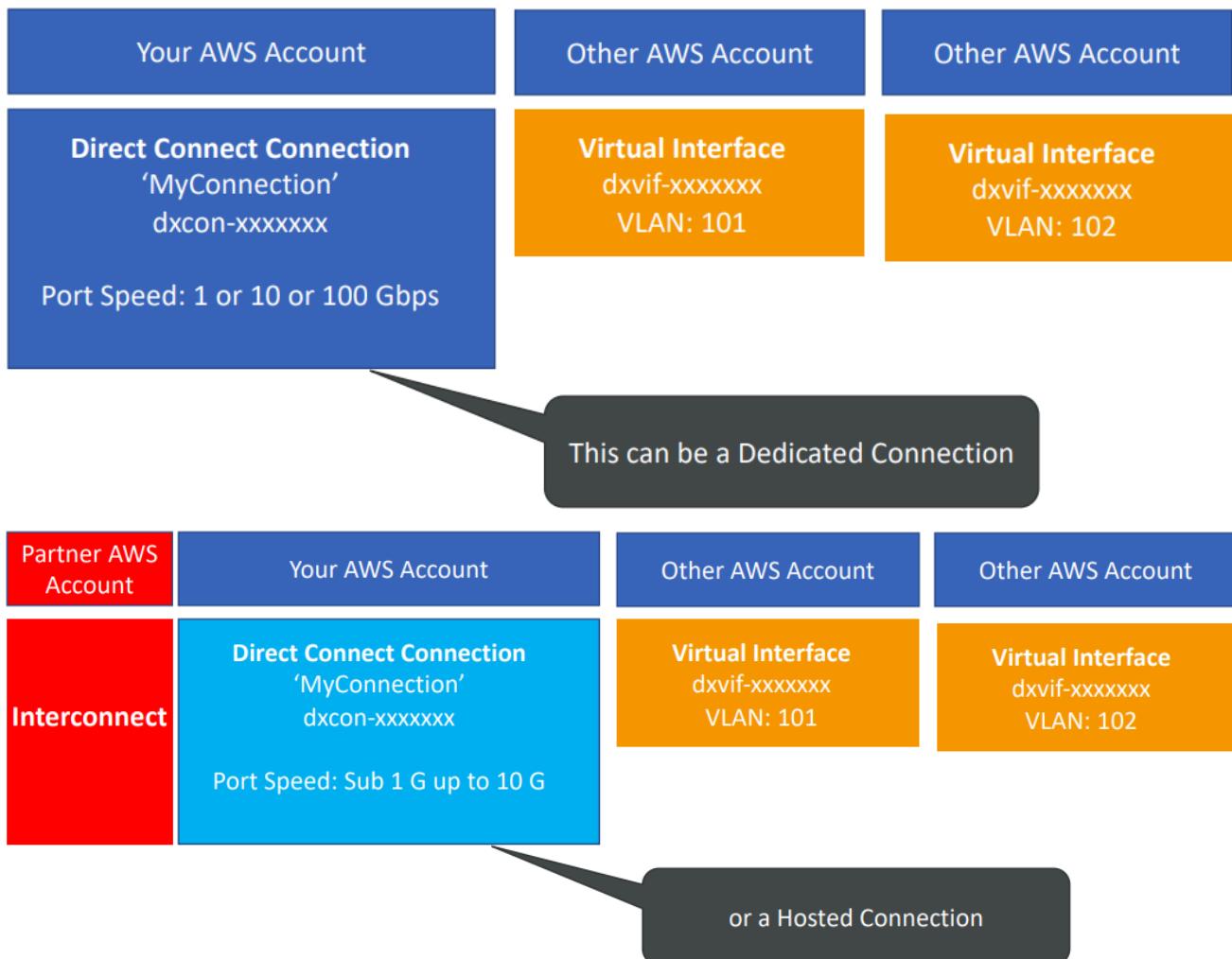




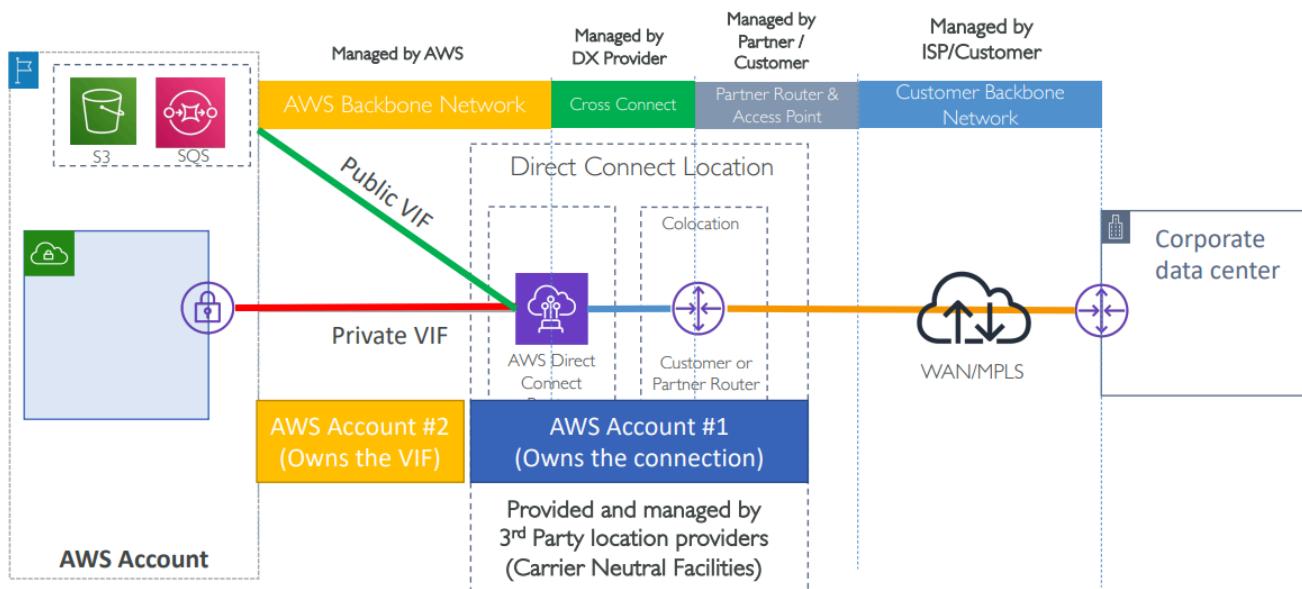
## Hosted Connection Structure

- Dedicated와는 다르게 1 Connection당 1 VIF만 제공하므로 추가 연결이 필요하면 DX를 여러개 만들어야 한다.





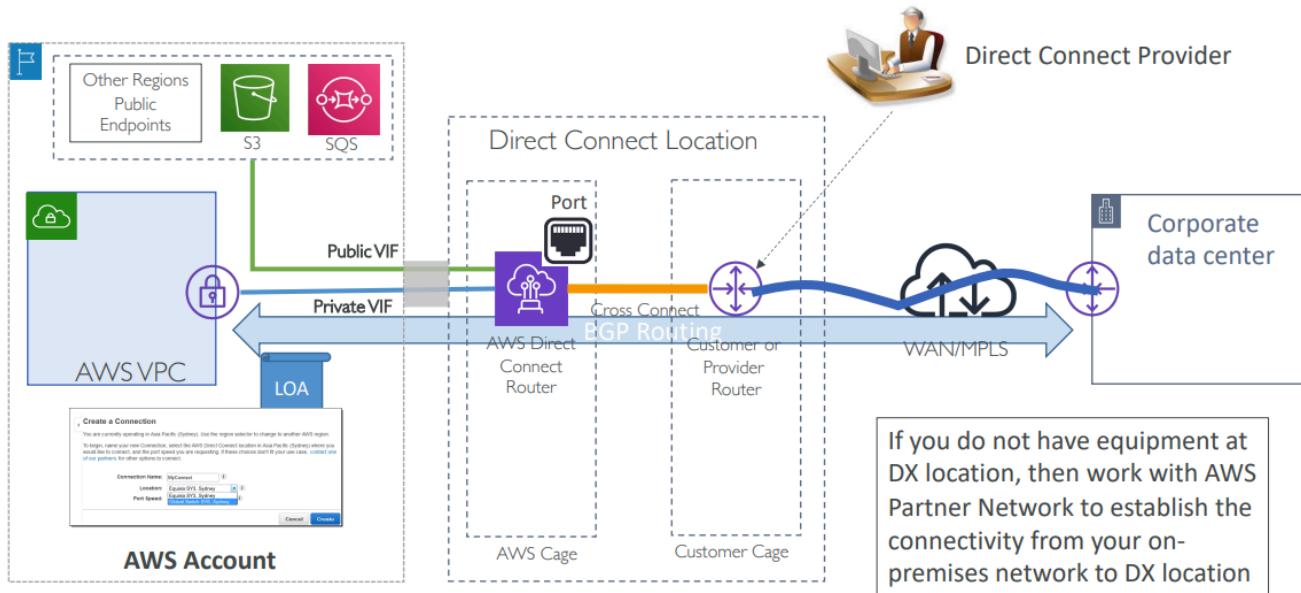
- Dedicated Hosted VIF는 고객이 온전히 제어가 가능하며, Organization 내 모든 연결을 설정 할 수 있다.
- Hosted의 경우, Partner가 일부 제어를 하고 있어 완전 제어는 불가능하지만, Organization 내 모든 연결을 설정할 수 있다.



## Direct Connect 설치 절차

## Dedicated

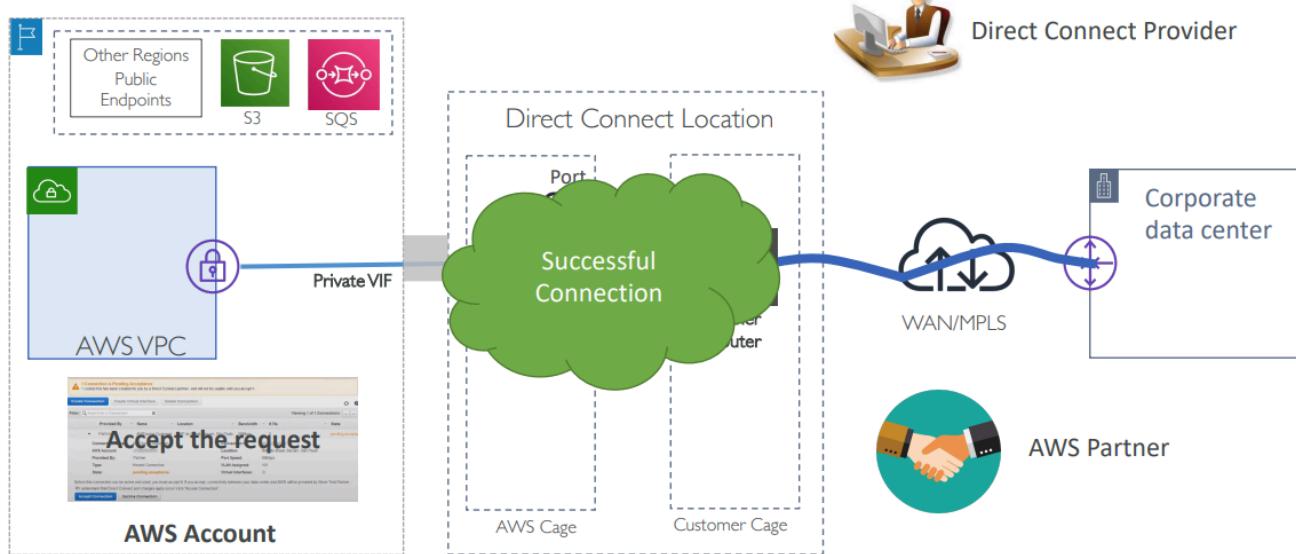
1. AWS 리전, DX(Direct Connect) 위치를 선택하고, AWS 콘솔, CLI 또는 API를 통해 연결 요청을 제출합니다.
2. AWS는 72시간 이내에 포트를 프로비저닝하고, LOA-CFA(Letter of Authorization – Connection Facility Assignment)를 제공합니다.
  - LOA-CFA는 승인서 및 연결 시설 할당서를 의미합니다.
3. LOA에는 해당 시설 내에서 할당된 포트의 경계(demarcation) 정보가 포함되어 있습니다.
4. 귀하의 조직이 DX 위치에 물리적으로 존재하는 경우, 시설 내에서 크로스-커넥트(cross-connect)를 요청하여 AWS 장치에 연결할 수 있습니다.
5. 그렇지 않은 경우, LOA 사본을 DX APN 파트너에게 제공하고, 파트너가 크로스-커넥트를 주문합니다.
6. 연결이 완료되면, 귀하의 장비에서 Tx/Rx 광 신호를 수신합니다. (-18 to -25db 적정)
7. 이제 프라이빗 또는 퍼블릭 가상 인터페이스(Virtual Interface)를 생성하여 VPC 또는 AWS 퍼블릭 서비스에 연결할 수 있습니다.



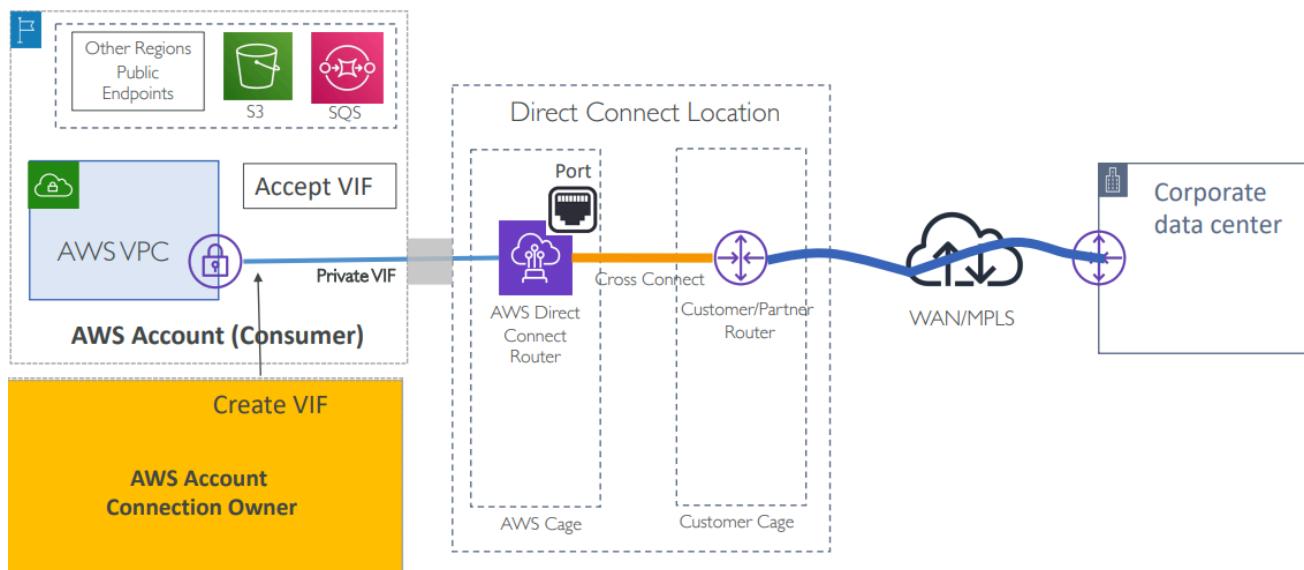
## Hosted

1. 호스팅 연결(Hosted Connection)을 주문하려면 LOA(Letter of Authorization)가 필요하지 않습니다. Direct Connect 파트너에 직접 연락하여 연결을 주문할 수 있습니다.
2. 12자리 AWS 계정 번호를 파트너에게 제공합니다.
3. 파트너는 호스팅 연결을 설정하며, 이 연결은 귀하의 AWS 계정(지정된 리전)에서 수락할 수 있게 됩니다.
4. 연결을 수락하면, 관련 포트 사용 시간 및 데이터 전송 요금에 대한 청구가 시작됩니다.
  - 연결 요청 후 연결 완료를 하지 않으면 AWS는 최대 90일을 대기하고 90일이 넘어가면 사

용료에 대한 청구할 수 있음.



- 호스팅 연결(Hosted Connection)과 혼동하지 마세요.
- VIF(Virtual Interface)를 생성할 때 "다른 AWS 계정"을 선택할 수 있습니다.
  - 이 경우, Direct Connect 연결의 소유자는 여전히 귀하이지만, VIF는 다른 AWS 계정에 생성되며, 해당 계정이 이를 수락해야 합니다.
- 프라이빗 VIF의 경우, 다른 계정은 VIF를 VGW(Virtual Private Gateway)와 연결해야 합니다.
- 1Gbps 미만의 연결은 하나의 가상 인터페이스만 지원합니다.
- 이 방식은 일반적으로 중앙 네트워크 팀이 DX 연결을 관리하고, 비즈니스 계정을 위해 VIF를 프로비저닝하는 시나리오에서 사용됩니다.



- LOA-CFA가 나올 수 있는데, Dedicated 설정시에만 필요하다는 것을 인지하고 있을 것.

## DX Virtual Interfaces

- VIF는 DX 연결을 위해 필수적인 설정입니다.

- 주로 802.1Q VLAN으로 구성됨.
- Public VIF**는 퍼블릭 서비스, **Private VIF**는 VPC, **Transit VIF**는 Transit Gateway와의 연결에 각각 사용됩니다.
- 이를 통해 AWS와 온프레미스 환경 간의 안정적이고 효율적인 네트워크 연결이 가능합니다.

### 1. Public VIF

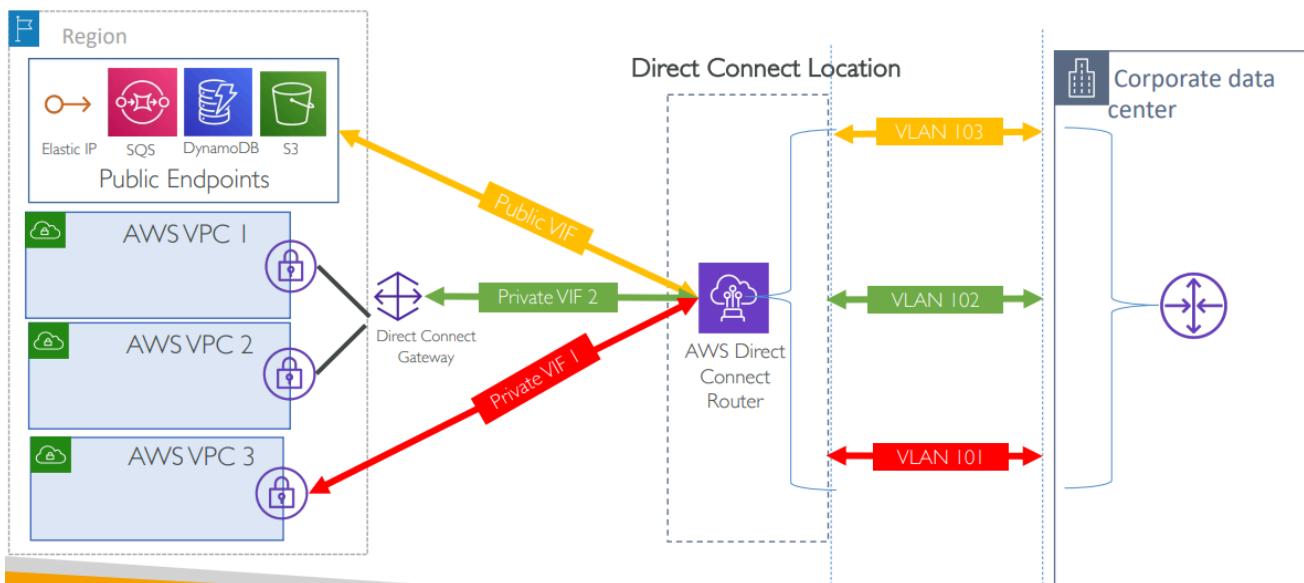
- **기능**: 모든 AWS 퍼블릭 IP 주소(예: S3, DynamoDB)에 연결할 수 있게 해줍니다.
- **용도**: AWS의 퍼블릭 서비스와 통신할 때 사용됩니다. (Elastic IP 포함)

### 1. Private VIF

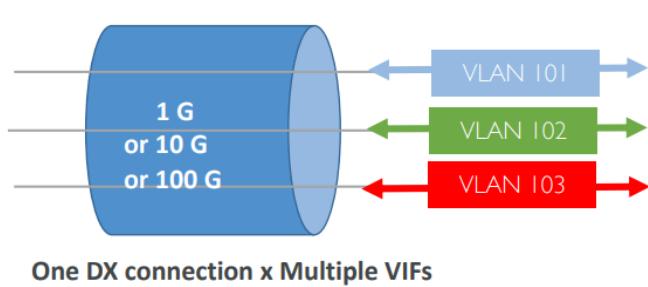
- **기능**: VPG 또는 Direct Connect Gateway를 통해 **VPC**에 연결할 수 있게 해줍니다.
- **용도**: 프라이빗 리소스(예: EC2 인스턴스)와 통신할 때 사용됩니다.

### 2. Transit VIF

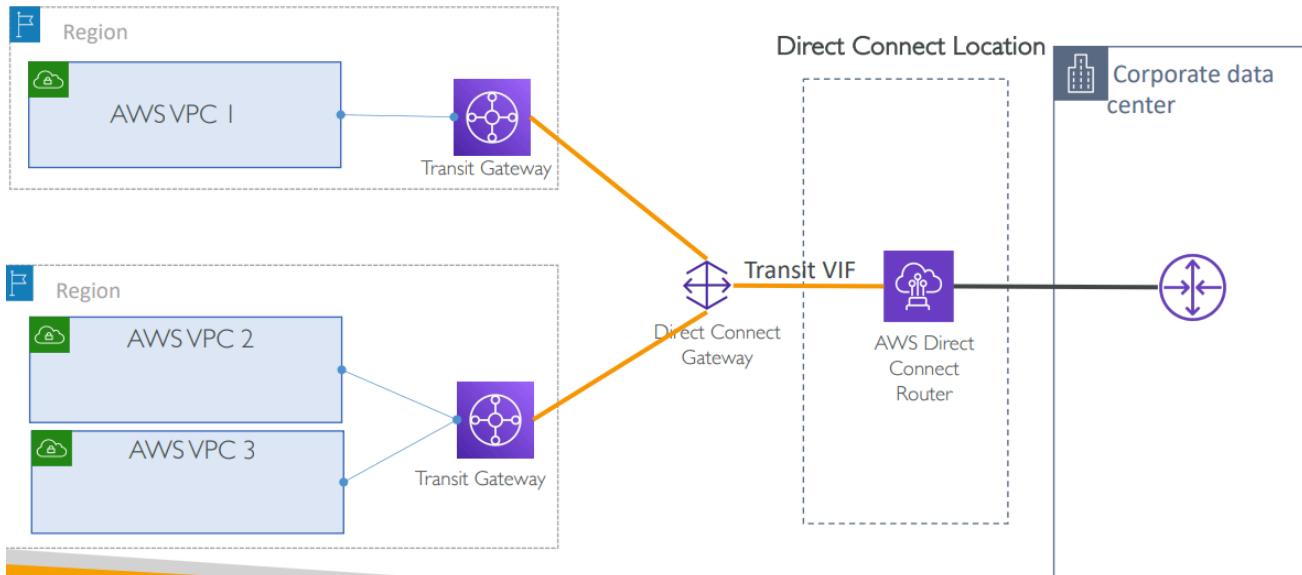
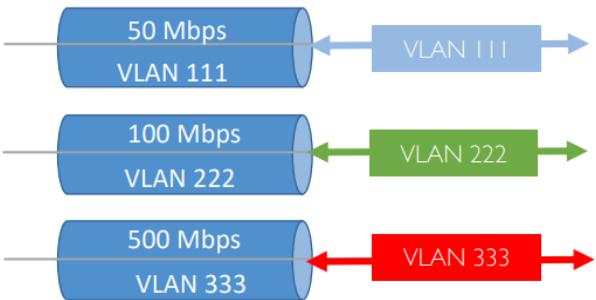
- **기능**: Direct Connect Gateway를 통해 **Transit Gateway**에 연결할 수 있게 해줍니다.
- **용도**: 여러 VPC와 온프레미스 네트워크를 중앙에서 연결할 때 사용됩니다.



## Dedicated Connection



## Hosted Connection



## VIF parameter

### 1. Connection:

- DX 연결 또는 LAG(Link Aggregation Group)를 사용하여 네트워크 연결을 구성합니다.
- LAG는 여러 물리적 연결을 하나의 논리적 연결로 묶어 대역폭과 안정성을 높입니다.

### 2. VIF 설정:

- VIF Type:** Public, Private, Transit 중 선택.
- VIF Name:** 임의로 지정 가능.
- VIF Owner:** 본인 계정 또는 다른 계정(호스팅된 VIF).

### 3. Gateway Type (Private VIF만 해당):

- Virtual Private Gateway 또는 Direct Connect Gateway 선택.

### 4. VLAN:

- VLAN ID는 1-4094 범위 내에서 중복 없이 설정.
- Hosted Connection의 경우 파트너가 VLAN ID를 미리 구성.

### 5. BGP 설정:

- IPv4:**

- **Public VIF:** 사용자가 퍼블릭 IP(/30)를 할당.
- **Private VIF:** AWS가 기본적으로 169.254.0.0/16 범위의 프라이빗 IP 제공.
- **IPv6:** AWS가 자동으로 /125 IPv6 CIDR를 할당하며, 사용자가 직접 지정 불가.

## 6. BGP ASN (Autonomous System Number):

- **퍼블릭 ASN:**
  - 고객이 소유해야 하며, **IANA**에서 할당받은 ASN을 사용해야 합니다.
- **프라이빗 ASN:**
  - 사용자가 직접 설정할 수 있으며, 범위는 다음과 같습니다:
    - **16-bit ASN:** 64512 ~ 65534
    - **32-bit ASN:** 1 ~ 2147483647

## 7. BGP MD5 인증 키:

- BGP 세션의 보안을 위해 MD5 인증 키를 설정할 수 있습니다.
- 사용자가 키를 제공하지 않으면 AWS가 자동으로 생성합니다.

## 8. 광고할 Prefixes (Public VIF만 해당):

- **Public VIF**를 사용할 경우, BGP를 통해 광고할 퍼블릭 **IPv4 경로** 또는 **IPv6 경로**를 지정해야 합니다.
- 이는 AWS와의 BGP 세션을 통해 네트워크 경로를 공유하는 데 사용됩니다.

## 9. Jumbo Frames (Private 및 Transit VIF만 해당):

- **Private VIF:**
  - **9001 MTU**를 지원하지만, 이는 전파된 경로(propagated routes)에만 적용됩니다.
  - 기본 MTU 값은 **1500**입니다.
- **Transit VIF:**
  - **8500 MTU**를 지원합니다.

## ip-ranges.json

- AWS에서 사용하는 IP 주소를 json형태로 제공하고 있다.
- <https://ip-ranges.amazonaws.com/ip-ranges.json>
- 서비스, 리전, ipv4, 6까지 필터링이 가능하다.

- AWS Public IP를 제어하는 데 효과적이다.

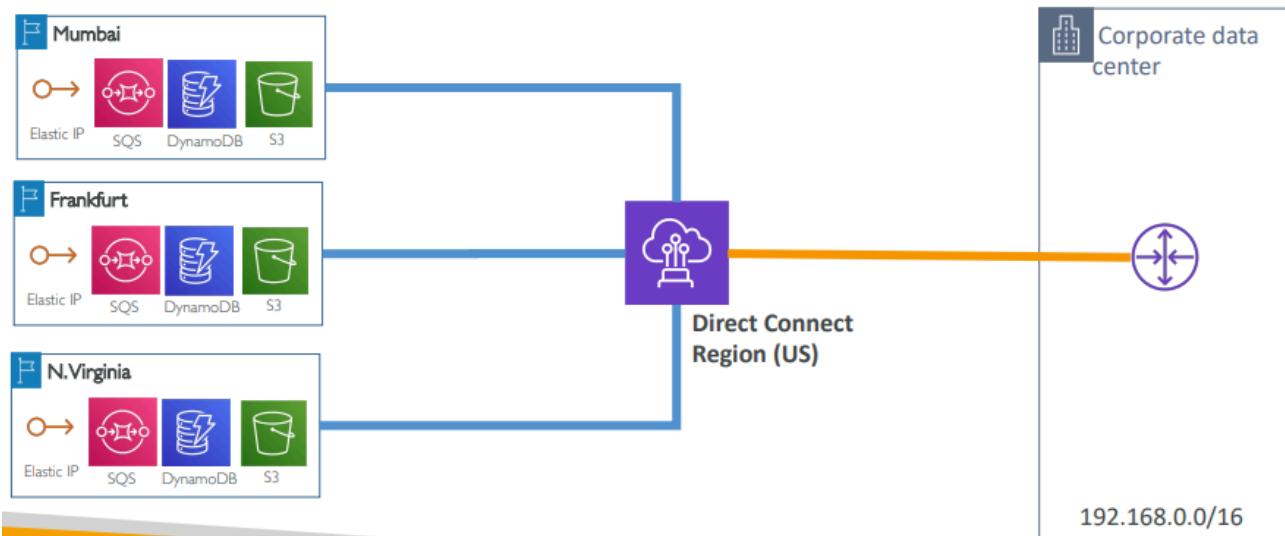
The syntax of ip-ranges.json:

```
{
  "syncToken": "0123456789",
  "createDate": "yyyy-mm-dd-hh-mm-ss",
  "prefixes": [
    {
      "ip_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ],
  "ipv6_prefixes": [
    {
      "ipv6_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ]
}
```

```
{
  "ip_prefix": "3.5.140.0/22",
  "region": "ap-northeast-2",
  "service": "S3",
  "network_border_group": "ap-northeast-2"
},
{
  "ip_prefix": "52.219.170.0/23",
  "region": "eu-central-1",
  "service": "S3",
  "network_border_group": "eu-central-1"
},
{
  "ip_prefix": "15.181.232.0/21",
  "region": "us-east-1",
  "service": "EC2",
  "network_border_group": "us-east-1-iah-1"
},
{
  "ip_prefix": "142.4.160.136/29",
  "region": "us-east-1",
  "service": "EC2",
  "network_border_group": "us-east-1-msp-1"
},
```

## Public VIF

- AWS의 전체 AWS Public IP를 사용할 수 있도록 하기 위함이다.



### • AWS 퍼블릭 IP와 글로벌 연결 지원

- 사용자의 네트워크가 AWS의 모든 퍼블릭 IP에 연결 가능함.
- S3, SQS, DynamoDB 같은 퍼블릭 엔드포인트에 접근 가능.

### • 퍼블릭 VIF 생성 요건

- AWS 라우터 및 사용자의 라우터 퍼블릭 IP 필요 (CIDR /30).
- 퍼블릭 IP가 없을 경우 AWS에서 /31 범위를 제공받을 수 있음.

### • IP 주소 광고 조건

- 광고할 IPv4 주소 프리픽스를 명시해야 함.
- AWS는 인터넷 등록 기관을 통해 사용자의 IP 소유권을 검증.

### • BGP를 통한 AWS 프리픽스 광고

- AWS는 BGP 세션을 통해 EC2, S3, Amazon.com 등의 프리픽스를 광고.
- Amazon 프리픽스 외의 네트워크에는 접근 불가.

- **프리픽스 관리**
  - 최신 Amazon 프리픽스는 `ip-ranges.json`에서 확인 가능.
  - 고객 라우터의 방화벽을 이용해 특정 Amazon 프리픽스 접근 제한 가능.
- **BGP 세션의 광고 제한**
  - 고객 라우터에서 AWS로 BGP 세션당 최대 **1000개의 경로 프리픽스만 광고 가능.**

## 요약

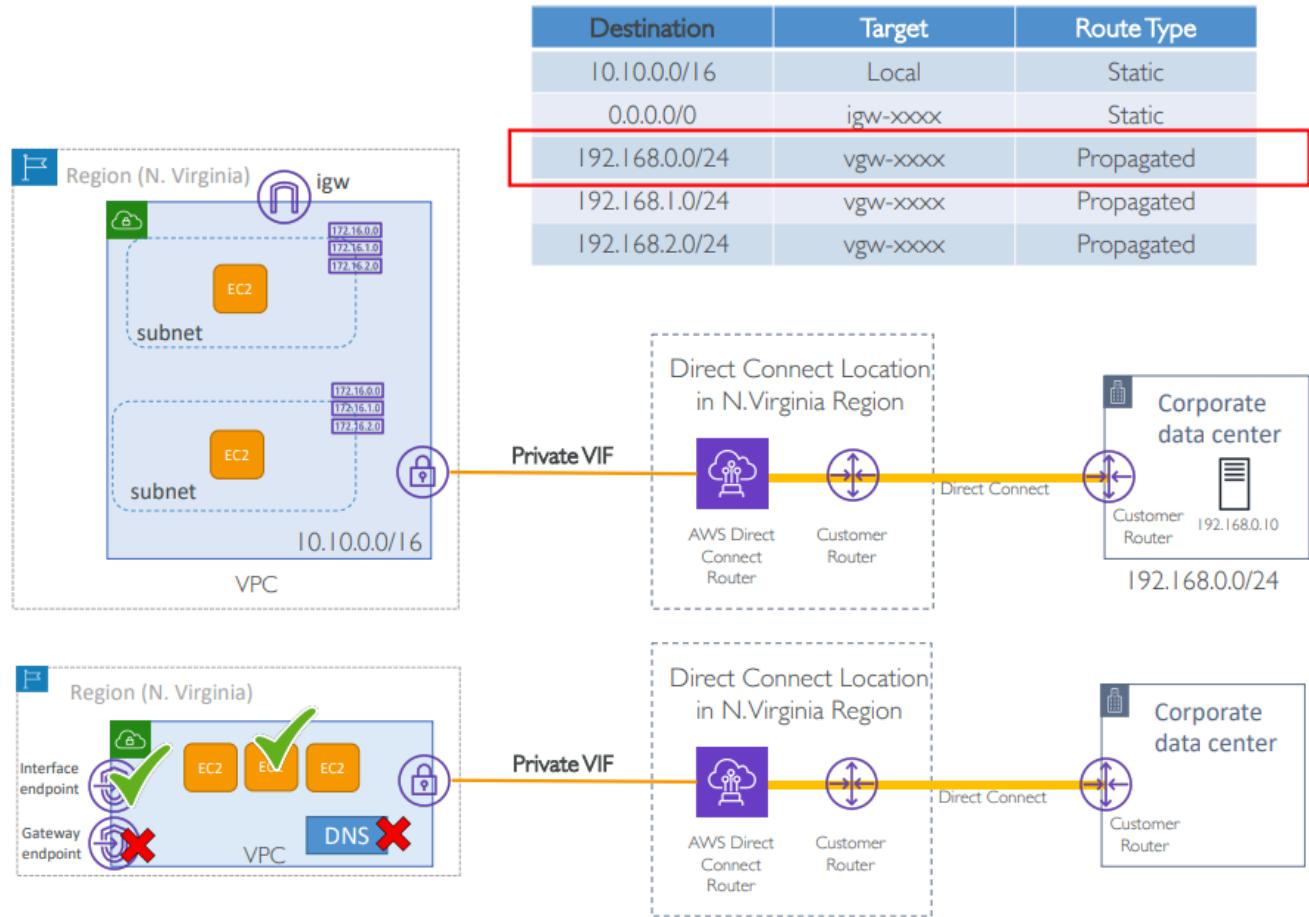
- Public VIF를 사용하면 AWS 퍼블릭 IP와 글로벌 연결 가능.
- Public VIF 생성 시 IPv4 /30 CIDR의 AWS 및 고객 라우터 IP 필요.
- BGP 세션을 통해 AWS가 Amazon 관련 프리픽스를 광고하며, 비-Amazon 프리픽스는 접근 불가.
- 최대 1000개의 경로 프리픽스를 BGP 세션당 광고 가능.
- 퍼블릭 IP가 없는 경우, AWS에서 /31 범위를 제공받을 수 있음.
- AWS는 IP 프리픽스 광고 시, 사용자의 소유권을 인터넷 등록 기관을 통해 검증.

## Private VIF

### 요약

- Private VIF는 VPC 내부 리소스(EC2, RDS, Redshift 등)에 Private IP로 연결하는 용도.
- VPC → VGW → Private VIF로 연결해야 하며, VGW와 Private VIF는 같은 리전에 존재해야 함.
- BGP 세션에서 VPC의 모든 프리픽스를 고객 라우터로 전달.
- 최대 100개의 프리픽스를 AWS에 광고할 수 있으며, 서브넷 라우트 테이블에 자동 전파 가능.
- 100개 초과 시 프리픽스를 요약(summarization)하여 개수를 줄여야 함.
- 전파된 경로는 IGW(인터넷 게이트웨이) 경로보다 우선순위가 높음.
- MTU 기본값은 1500이며, 전파된 경로는 9001까지 지원 가능.

- VPC DNS 리졸버(Base + 2)와 VPC 게이트웨이 엔드포인트에 접근 불가.



## 구성 및 제한사항

- VPC → VGW(Virtual Private Gateway) → Private VIF로 연결해야 하며, VGW와 Private VIF는 같은 AWS 리전에 있어야 함.
- VPC DNS 리졸버(Base + 2) 및 VPC 게이트웨이 엔드포인트에 접근할 수 없음.

## Transit VIF

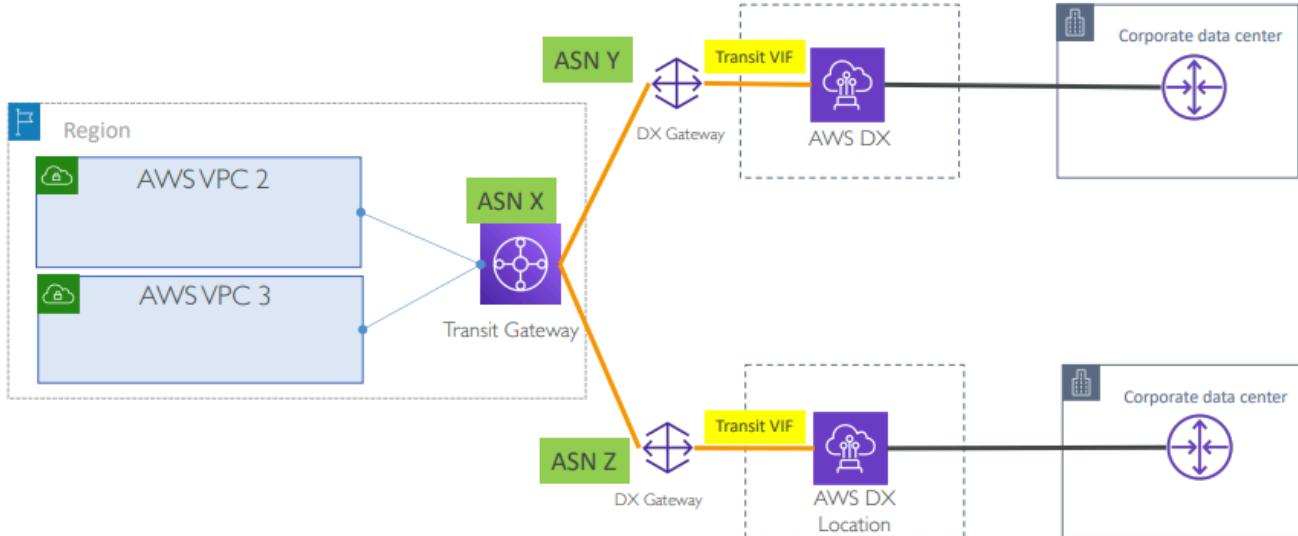
### 기능 및 구성

- Transit VIF는 Direct Connect와 Transit Gateway를 연결하는 역할을 함.
- Transit VIF → Direct Connect Gateway(DX Gateway) → Transit Gateway 순으로 연결됨.
- Direct Connect Gateway에 여러 개의 Transit Gateway를 연결할 수 있음.
- 하나의 Transit Gateway에 여러 개의 DX Gateway를 연결할 수 있음.

### 기술적 제한 및 요구사항

- MTU는 기본 1500, Jumbo Frames의 경우 최대 8500까지 지원.
- DX Gateway와 Transit Gateway의 ASN(Autonomous System Number)은 서로 달라야 함.

- 둘 다 기본 ASN(64512)을 사용할 경우, 연결이 실패함.



## DX Gateway with Private VIF and VPG

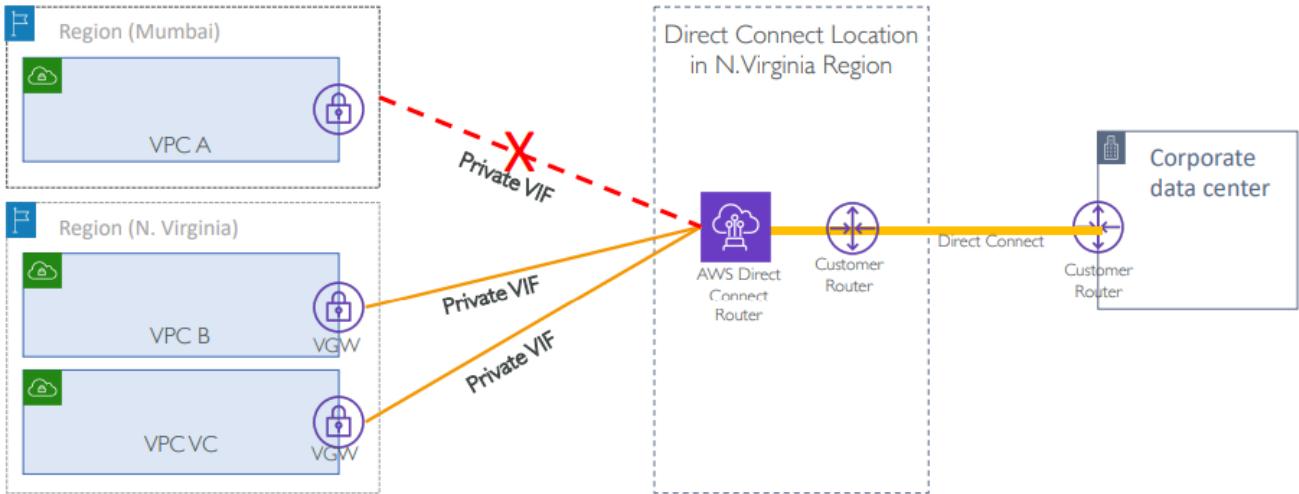
- **문제점:** 동일한 Direct Connect 연결을 통해 여러 개의 VPC에 접근하고 싶습니다.
- **해결책:** AWS Direct Connect Gateway를 사용합니다.
  - **하나의 Direct Connect Gateway**를 통해 여러 개의 VPC(같은 리전 또는 다른 리전, 같은 계정 또는 다른 계정의 VPC)와 연결할 수 있습니다.
  - **연결 방식:** Virtual Private Gateway(VGW) 또는 Transit Gateway(TGW) 사용.
- Direct Connect Gateway는 글로벌 네트워크 장치이며 모든 AWS 리전에서 접근 가능합니다.
- Direct Connect는 Private VIF 또는 Transit VIF를 통해 통합됩니다.
- DX Gateway는 VPC ↔ 온프레미스 연결용이며, 퍼블릭 엔드포인트 연결에는 사용할 수 없습니다.
- Private VIF 또는 Transit VIF와 Direct Connect Gateway는 같은 AWS 계정에서 소유해야 합니다.
  - 그러나 연결할 VPC(VGW) 또는 Transit Gateway는 같은 계정이거나 다른 계정일 수 있습니다.

## 기능 및 구성

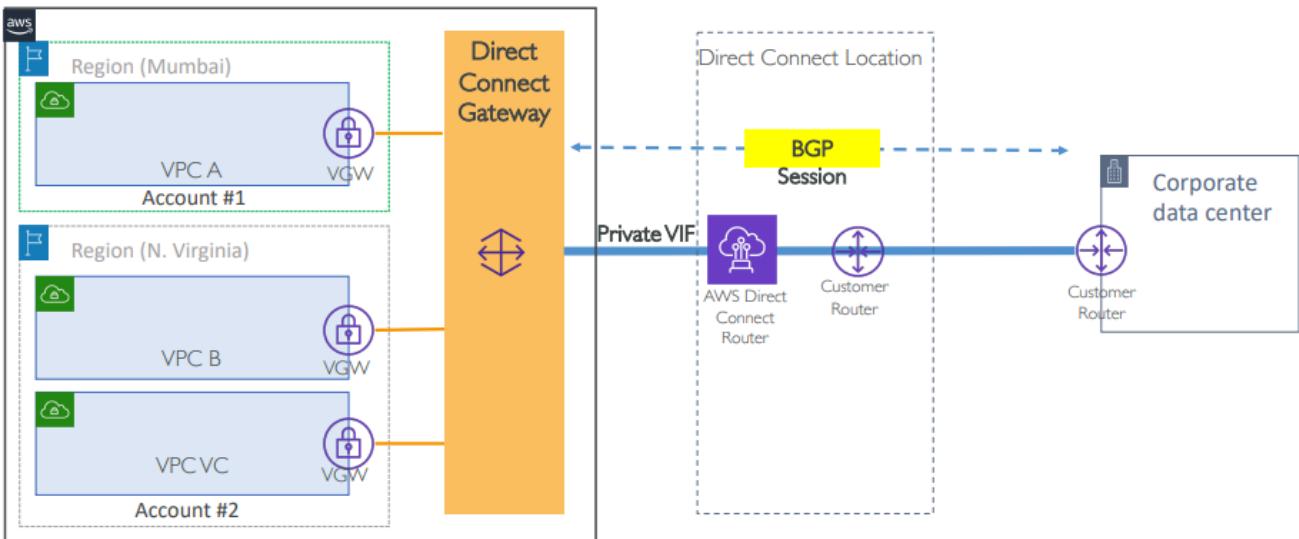
- DX Gateway를 사용하면 하나의 Direct Connect 연결을 통해 여러 VPC와 연결 가능.
- VPC는 같은 리전 또는 다른 리전에 있을 수 있으며, 같은 AWS 계정 또는 다른 AWS 계정에 속할 수 있음.
- Direct Connect는 Private VIF 또는 Transit VIF를 통해 연결됨.
- DX Gateway는 글로벌 서비스이며, 모든 AWS 리전에서 사용 가능.

## 제한 사항

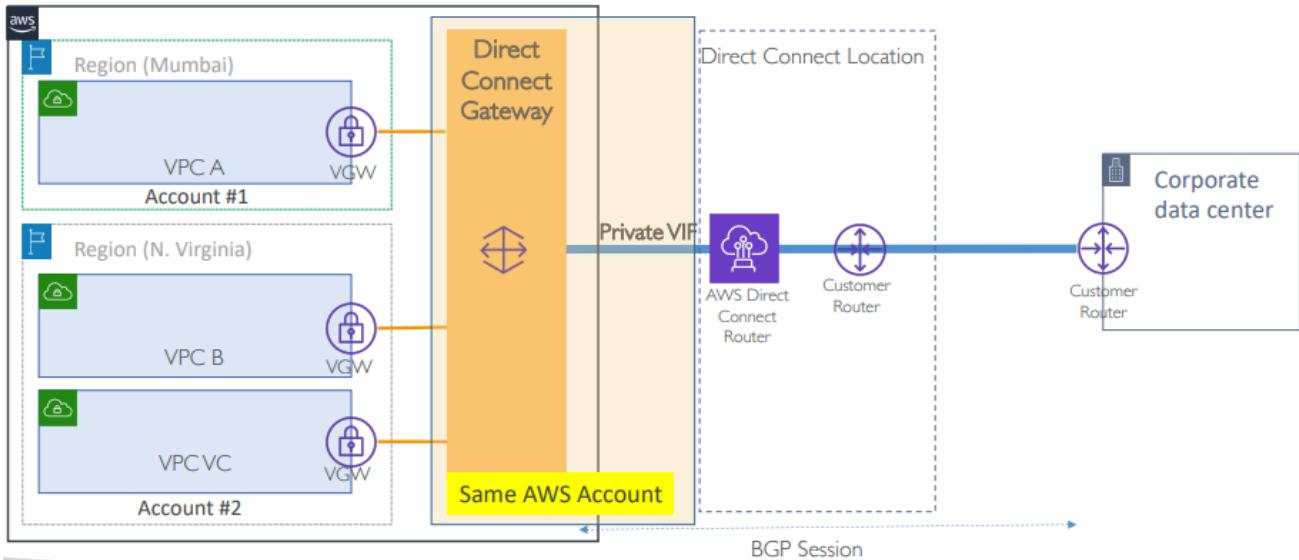
- DX Gateway는 온프레미스 ↔ VPC 연결용이며, 퍼블릭 엔드포인트 연결 불가.
- Private VIF 또는 Transit VIF 및 DX Gateway는 같은 AWS 계정에서 소유해야 함.
  - 그러나 연결되는 VPC(VGW) 또는 Transit Gateway는 같은 계정 또는 다른 계정에 속할 수 있음.
- 현재 Direct Connect Gateway는 최대 20 VGW (VPC)까지만 처리됨.



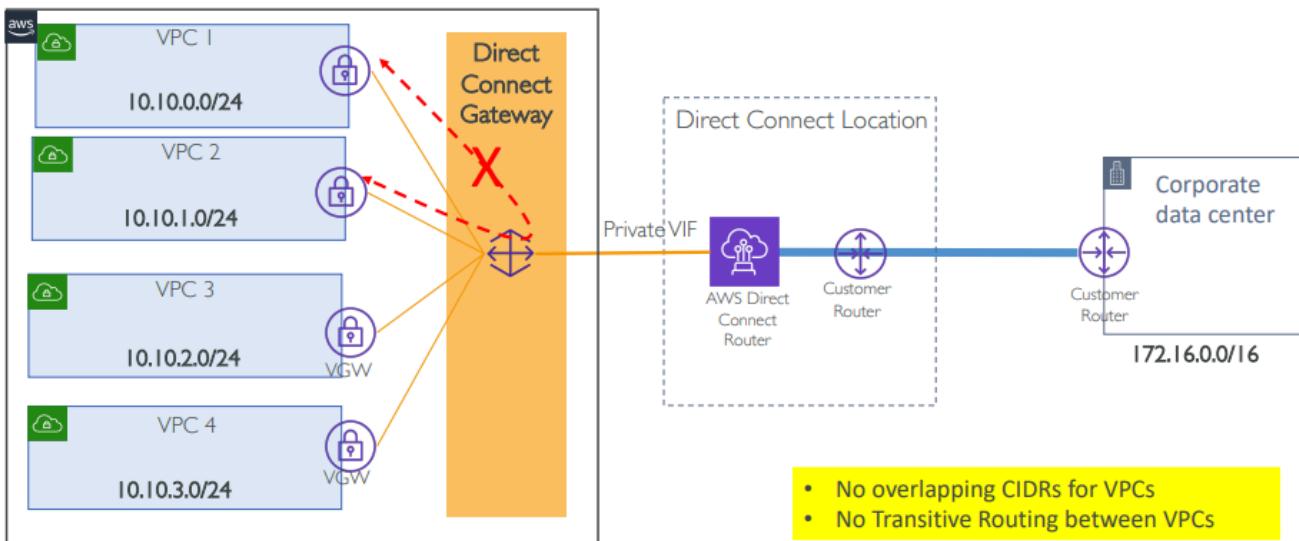
- Virginia에 생성된 DX가 같은 리전에는 Private VIF는 연결하지만 다른 리전에는 연결하지 못하는 이슈



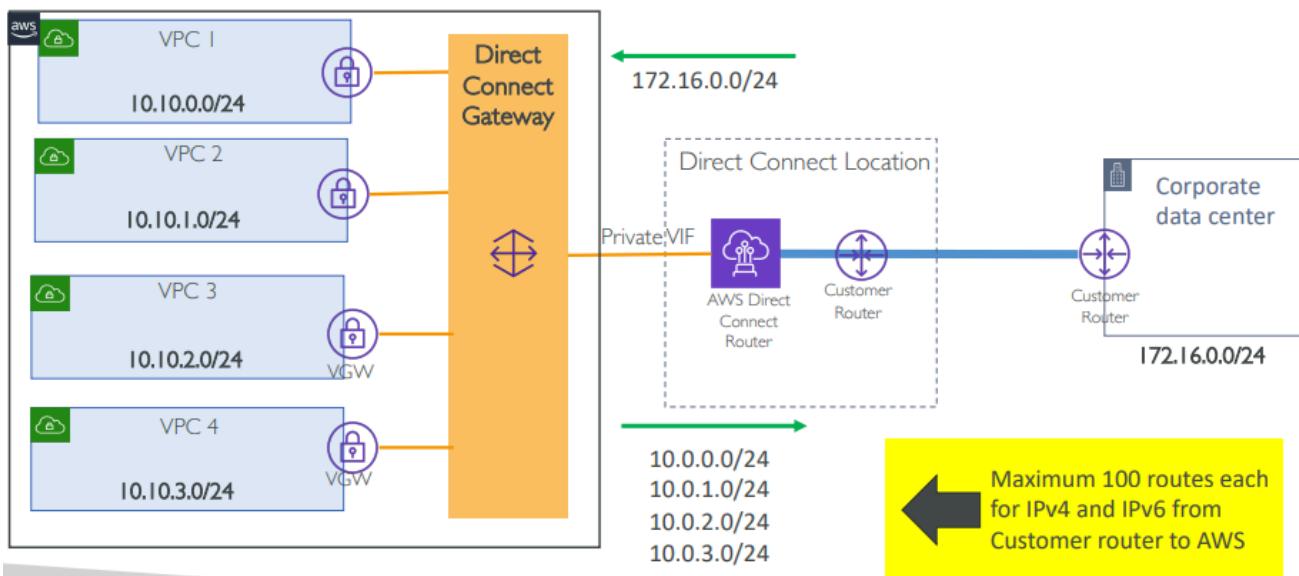
- DX Gateway를 이용해 모든 리전을 통합 관리할 수 있음.



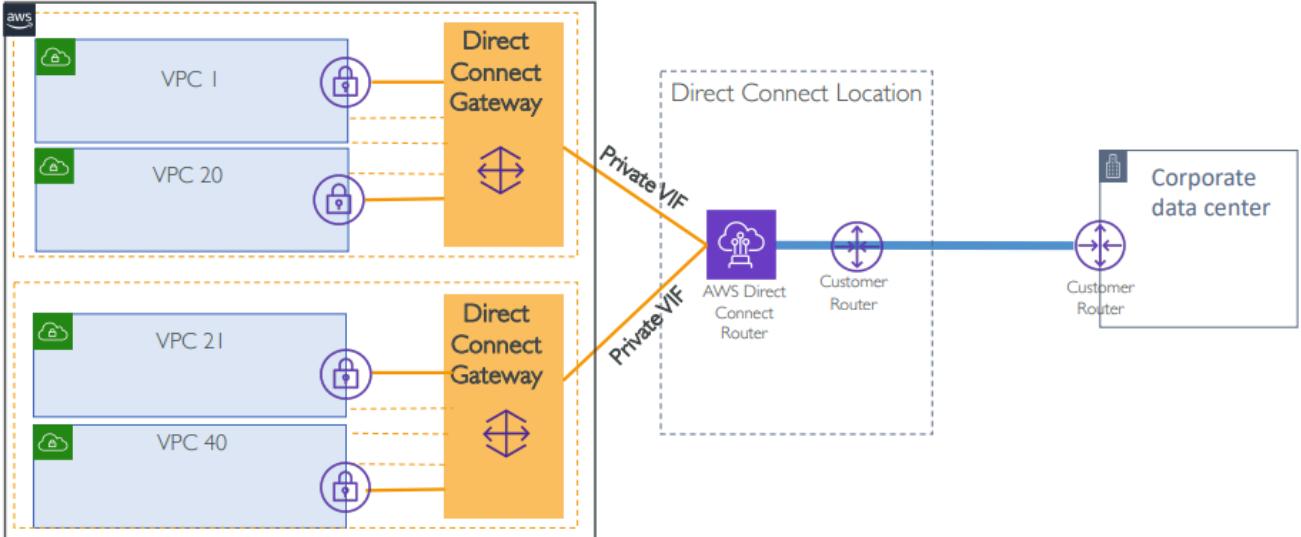
- DX Gateway와 Private VIF는 같은 계정에 들어 있어야 함.



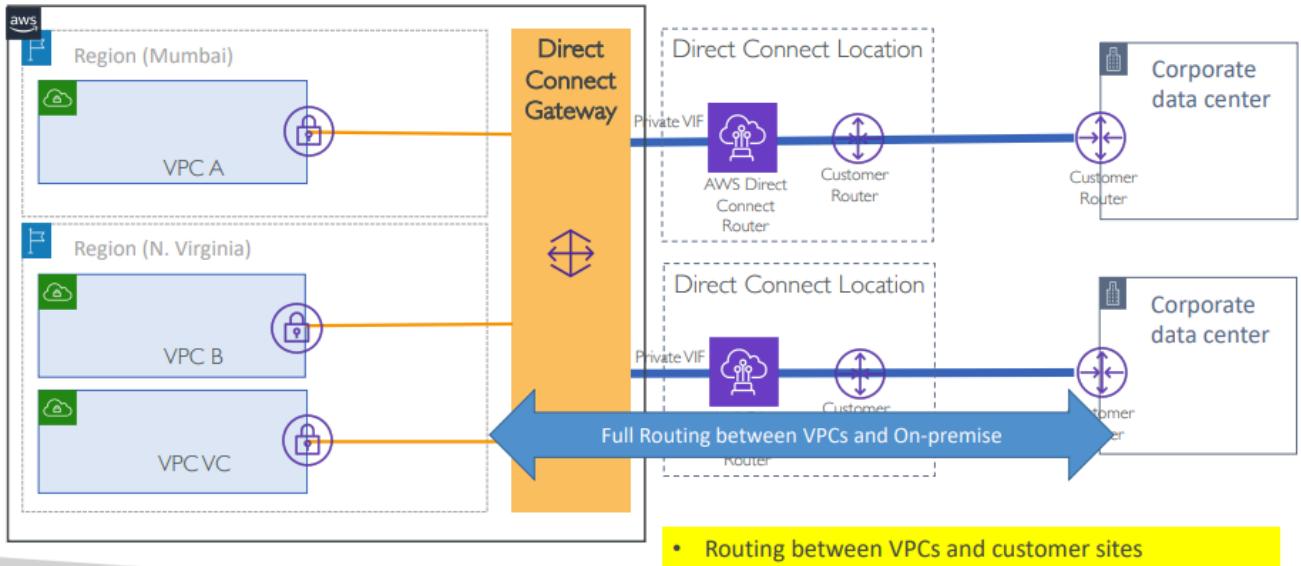
- VPC간 라우팅을 연동해주는 용도가 아니므로 거쳐가는 트래픽은 지원하지 않음.



- 최대 100개 (ipv4, ipv6 각각)의 라우트를 광고할 수 있다. (Private VIF, Transit VIF 모두 동일)



- 20개의 VPG연결을 피하려면 이렇게도 가능



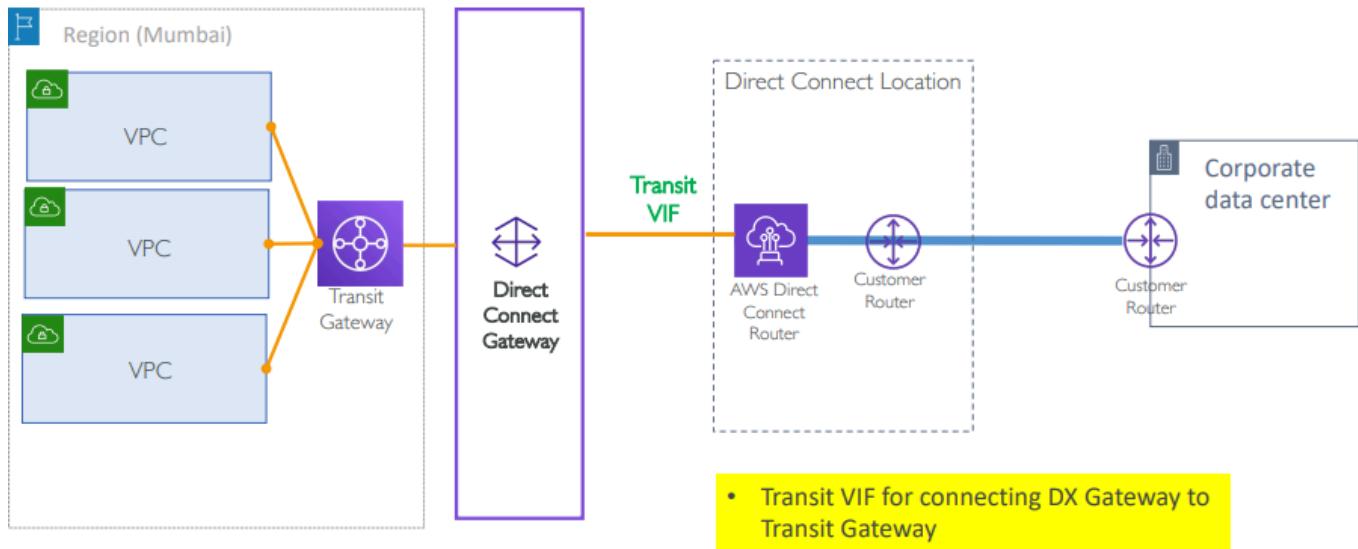
- 글로벌 리전인 경우 이런 방식으로도 할 수 있다.

## 요약

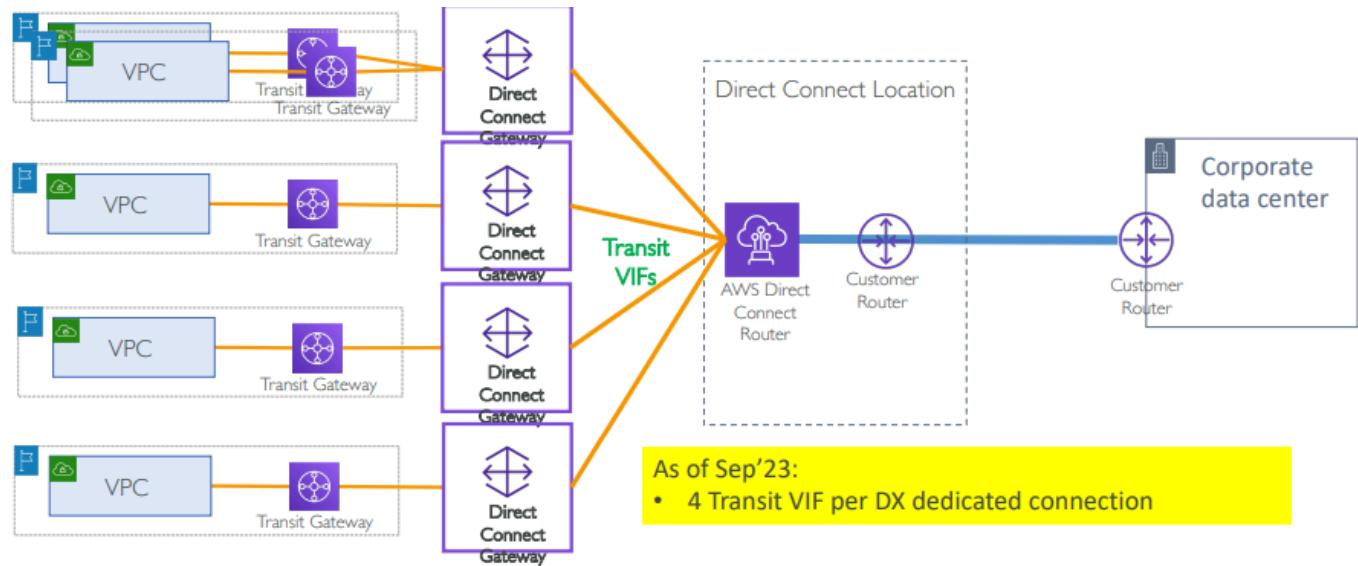
- 글로벌하게 접근 가능
- 온프레미스 네트워크와 AWS VPC 간의 프라이빗 연결을 제공
- 하나의 DX Gateway는 최대 20개의 VGW(VPC)와 연결 가능 (AWS 리전 및 계정 간 연결 가능)
  - 이 제한은 변경될 수 있으므로 최신 AWS 문서를 참고하세요.
- VPC 간 CIDR은 겹치면 안 됨
- DX Gateway를 통한 VPC 간 통신(VPC ↔ VPC)은 허용되지 않음
- DX Gateway와 Private VIF는 같은 AWS 계정에서 생성해야 함
- DX Gateway 사용에는 추가 비용이 없음
  - 단, 데이터 송출(egress) 요금은 원격 AWS 리전 및 DX 포트 사용 시간에 따라 부과됨.

# DX Gateway with Transit VIF and Transit Gateway

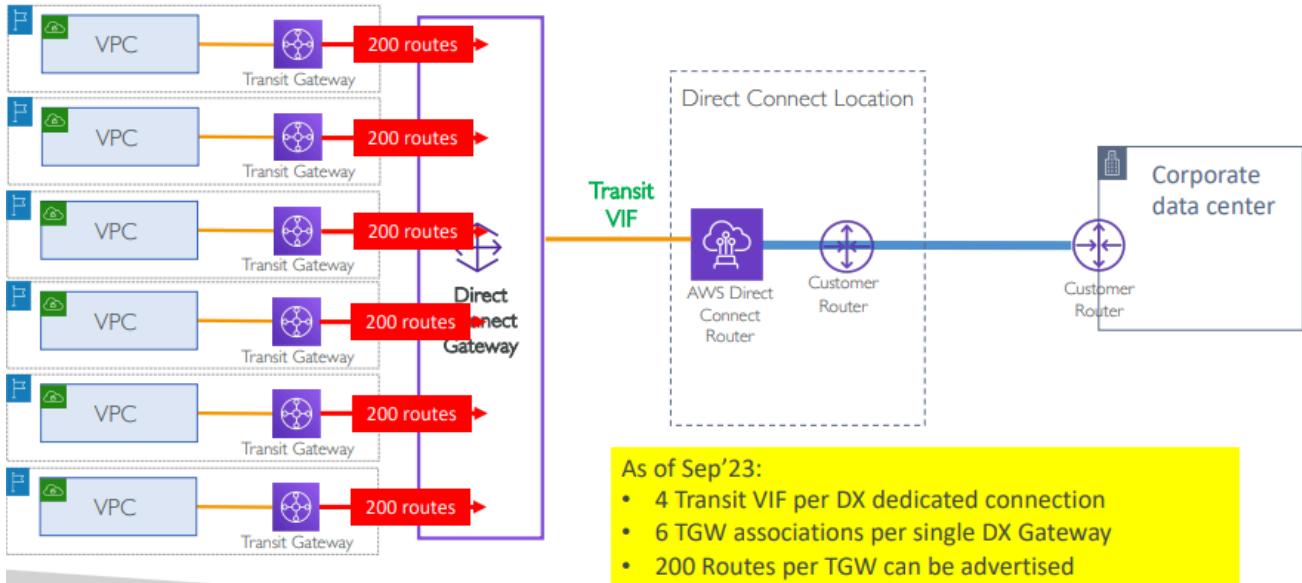
## 기본 구성



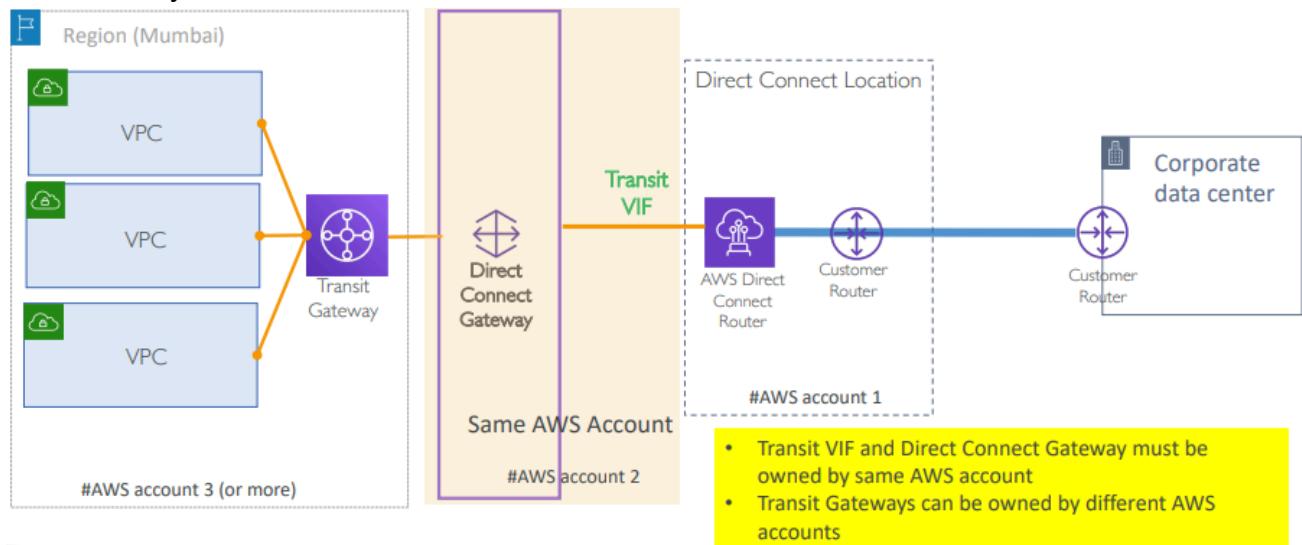
## Multi Transit VIF



- 현재 전용 연결은 4개의 Transit VIF가 최대임

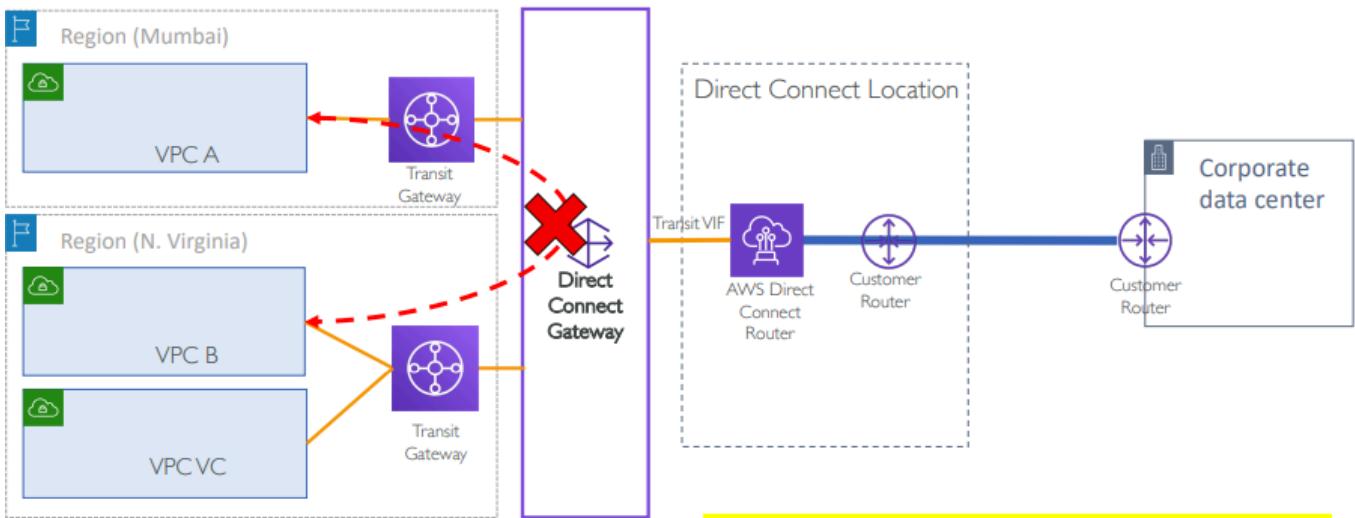


- DX Gateway당 6개의 TGW 연결이 가능함.

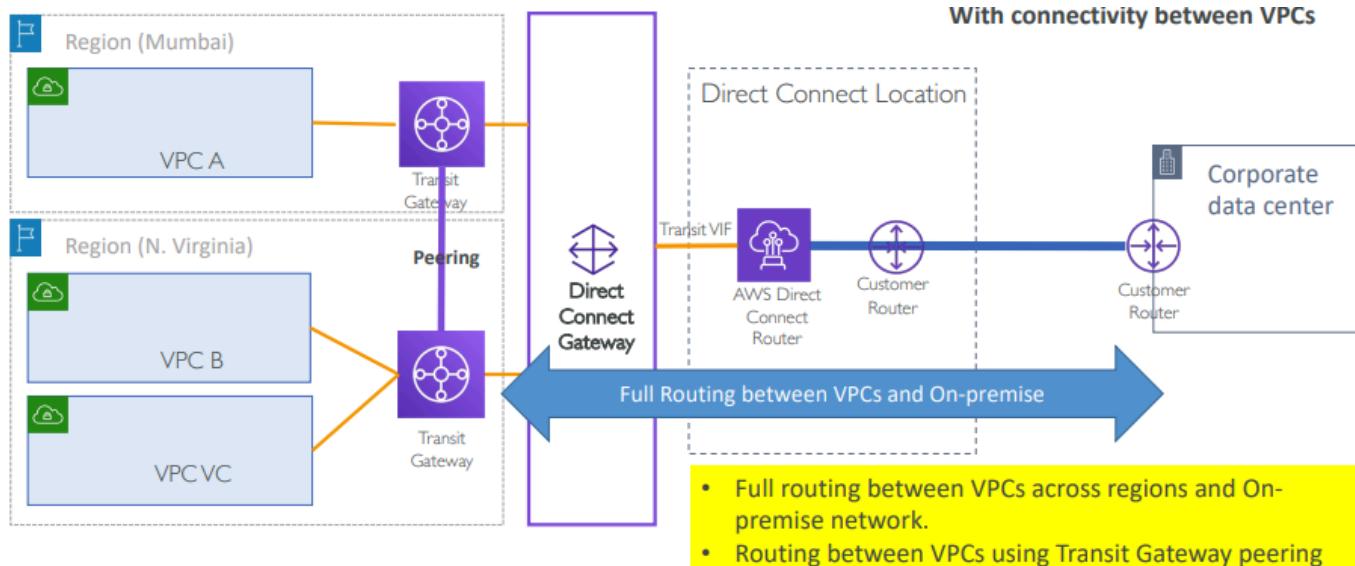


- Transit VIF와 DX Gateway는 같은 계정이어야 한다는 사실을 잊으면 안됨.

## Case1

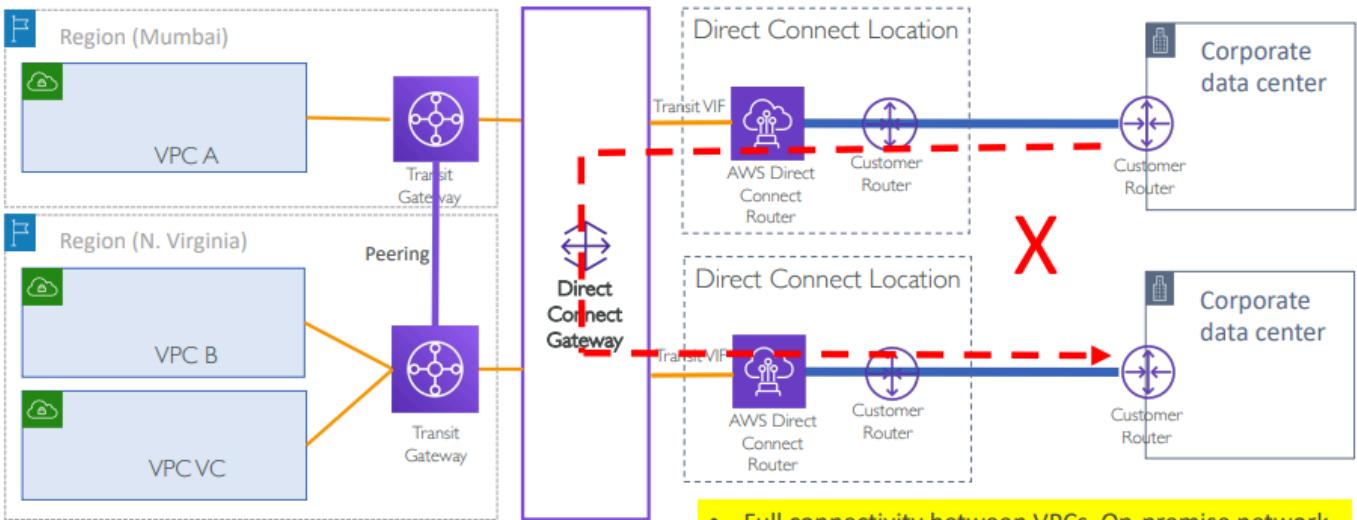


- No Transitive Routing between VPCs

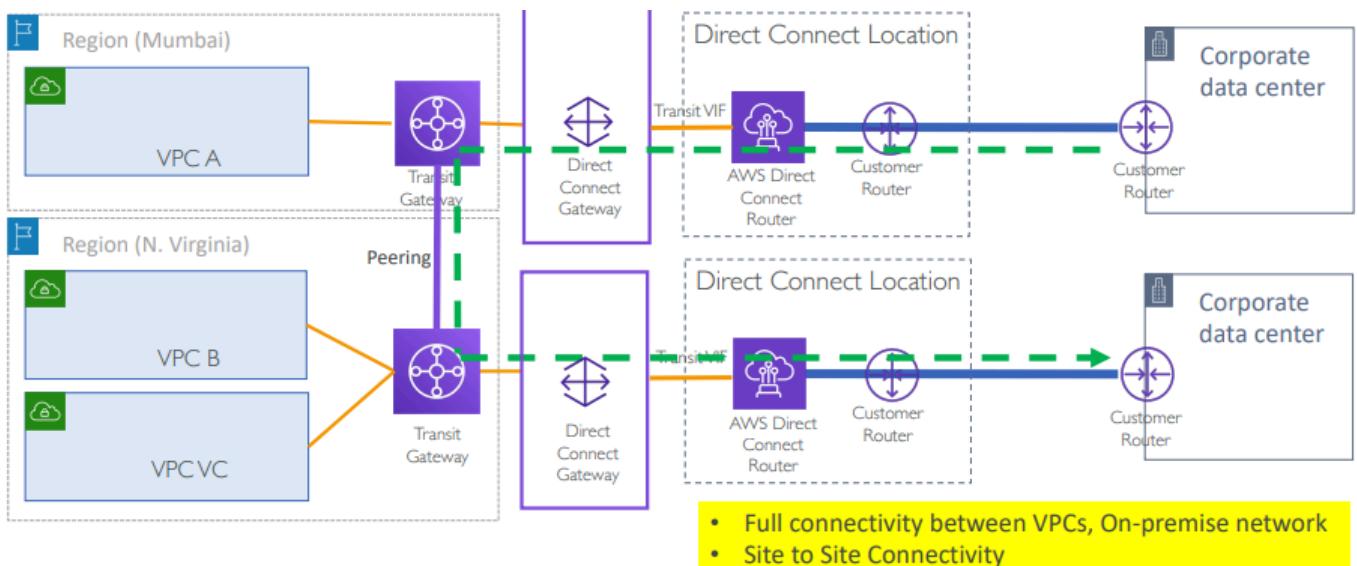


- Full routing between VPCs across regions and On-premise network.
- Routing between VPCs using Transit Gateway peering

## Case2



- Full connectivity between VPCs, On-premise network
- No connectivity between customer sites because DX Gateway does not support transitive routing



- Full connectivity between VPCs, On-premise network
- Site to Site Connectivity

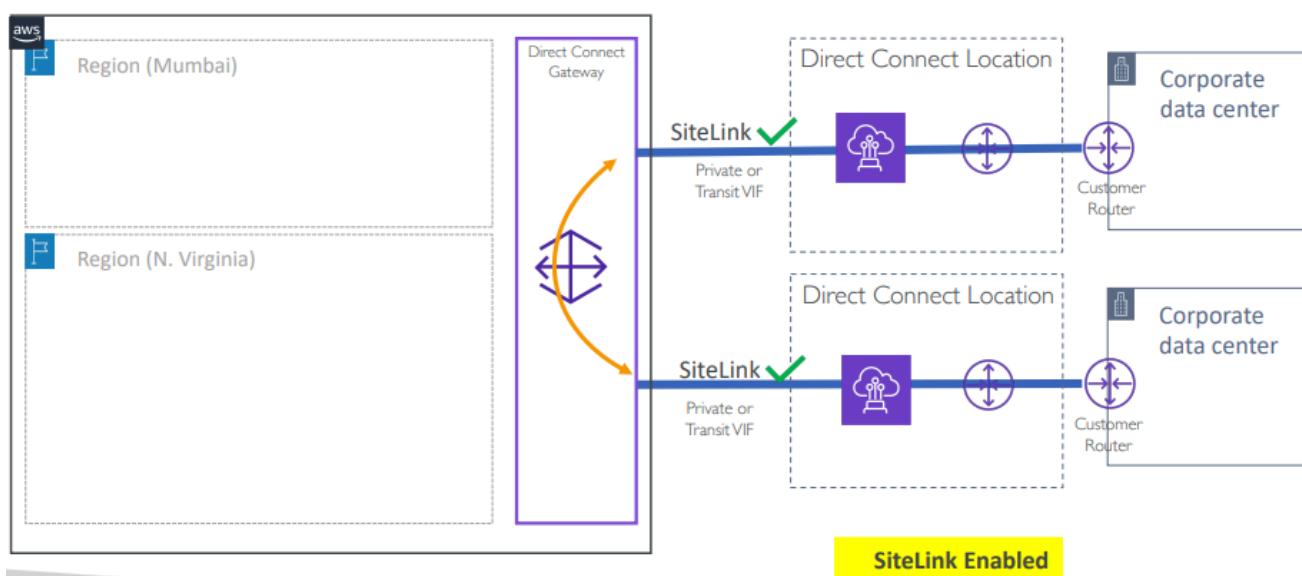
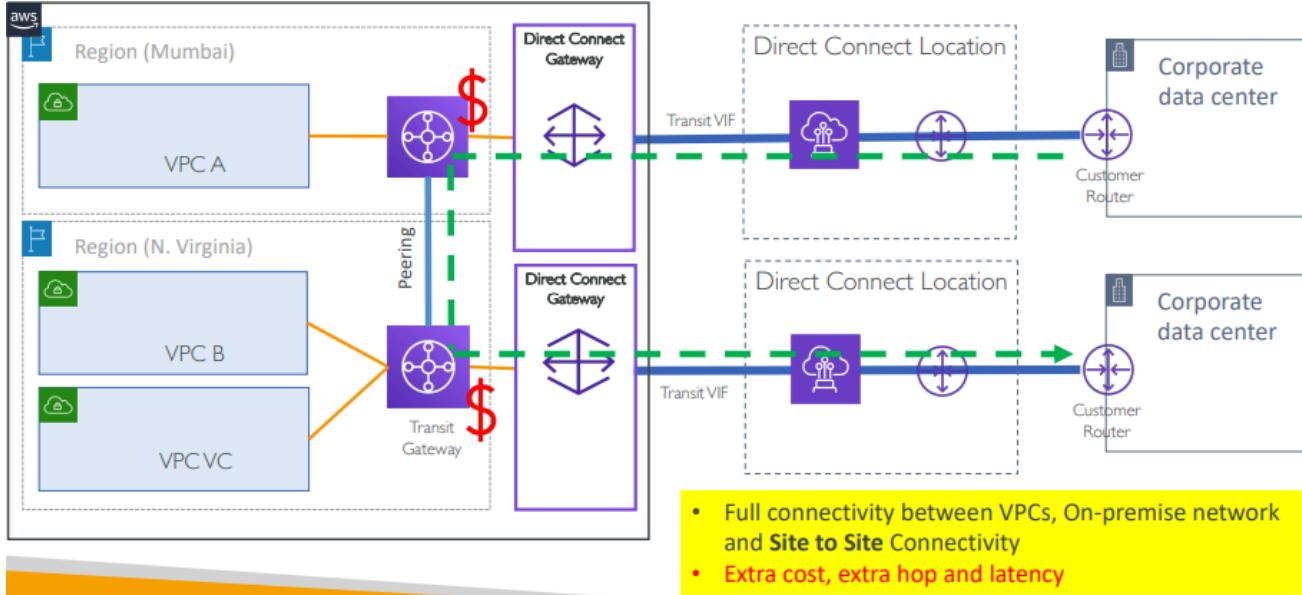
## 요약

- 하나의 Direct Connect 전용 연결(Direct Connect Dedicated Connection)당 최대 4개의 Transit VIF를 연결할 수 있음.
- 하나의 Direct Connect 호스팅 연결(Direct Connect Hosted Connection)당 최대 1개의 Transit VIF만 연결할 수 있음.
- Transit Gateway(TGW)는 Direct Connect Gateway(DX Gateway)와 연결됨.
- 하나의 DX Gateway에 최대 6개의 TGW를 연결할 수 있음.
- TGW 간 피어링을 통해 AWS 리전 간 VPC 간 통신이 가능함.

## AWS Direct Connect – SiteLink

- SiteLink는 Private VIF 또는 Transit VIF에서 활성화할 수 있음.
- 전용(Dedicated) 또는 호스팅(Hosted) Direct Connect 연결과 다양한 포트 속도에서 지원됨.

- 트래픽은 AWS 글로벌 네트워크를 통해 가장 짧은 경로로 전송됨.
- 몇 분 안에 SiteLink를 켜거나 끌 수 있음.
- IPv4 및 IPv6 라우팅 프리픽스와 트래픽을 지원함.
- 고객 위치 간 풀 메쉬(Full Mesh) 또는 격리된 네트워크 연결을 지원함.



## 기능 및 장점

- Private VIF 및 Transit VIF에서 SiteLink 활성화 가능.
- 전용(Dedicated) 및 호스팅(Hosted) DX 연결 모두 지원, 다양한 포트 속도에서 사용 가능.
- 트래픽은 AWS 글로벌 네트워크에서 가장 짧은 경로를 선택하여 최적화됨.
- 몇 분 안에 SiteLink 기능을 활성화하거나 비활성화 가능.
- IPv4 및 IPv6 트래픽과 라우팅 프리픽스를 지원.
- 고객 네트워크 간 풀 메쉬(Full Mesh) 또는 개별적으로 격리된 연결 구성 가능.

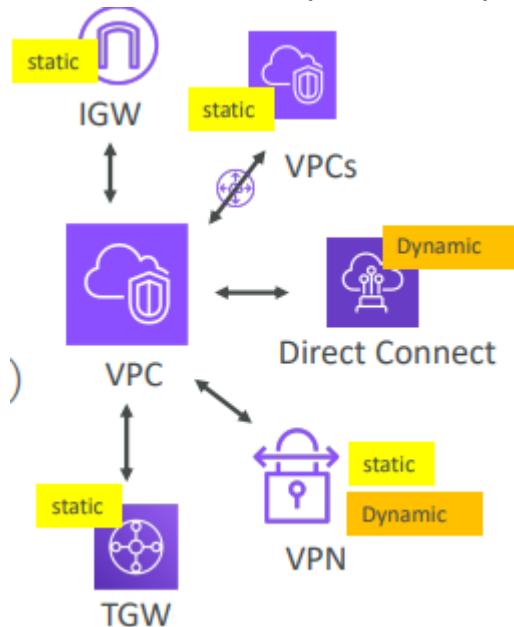
## 제한 사항

- SiteLink를 사용하려면 Private VIF 또는 Transit VIF가 필요함.
- 전용(Dedicated) 또는 호스팅(Hosted) DX 연결 유형에 따라 설정이 달라질 수 있음.

## Direct Connect Routing Policies and BGP Communities

### VPC에서 나가는 트래픽의 라우팅 우선순위

1. 가장 긴 프리픽스(Longest Prefix Match)가 우선 적용됨.
  - 예: 10.10.2.15/32 가 10.10.2.0/24 보다 우선순위가 높음.
2. 정적(Static) 라우트가 전파(Propagated)된 라우트보다 우선함.
3. 전파된(Propagated) 라우트가 적용됨.
  1. Direct Connect BGP 라우트 (동적 라우트)
  2. VPN 정적(Static) 라우트
  3. VPN BGP 라우트 (동적 라우트)



- 라우팅 정책은 Direct Connect 연결을 통해 트래픽이 흐를 때 라우팅 결정에 영향을 미침.
- 인바운드(Inbound) 라우팅 정책: 온프레미스 → AWS 방향의 트래픽을 제어하는 정책.
- 아웃바운드(Outbound) 라우팅 정책: AWS → 온프레미스 방향의 트래픽을 제어하는 정책.

- 라우팅 정책과 BGP 커뮤니티는 Public VIF와 Private/Transit VIF에서 다르게 동작함.



- For path selection (route priority) use BGP attributes:
    - Longest prefix
    - AS\_PATH
  - For route propagation scope, use BGP community tags
    - Inbound: 7224:9100, 7224:9200, 7224:9300
    - Outbound: 7224:8100, 7224:8200, No tag
    - NO\_EXPORT
- For path selection (route priority) use BGP attributes:
    - Longest prefix
    - AS\_PATH
    - Local preference BGP community Tags:
      - 7224:7100 (Low)
      - 7224:7200 (Medium)
      - 7224:7300 (High)

## Public VIF

### 1. 경로 선택(Path Selection) 우선순위 (BGP 속성):

- Longest prefix (가장 긴 프리픽스 우선).
- AS\_PATH (BGP 경로 속성).

### 2. 경로 전파 범위(Route Propagation Scope):

- Inbound**(온프레미스 → AWS)
  - BGP 커뮤니티 태그: 7224:9100, 7224:9200, 7224:9300 .
- Outbound**(AWS → 온프레미스)
  - BGP 커뮤니티 태그: 7224:8100, 7224:8200, 또는 **No tag**.
- NO\_EXPORT**: 해당 경로를 외부로 전파하지 않음.

## Private/Transit VIF

### 1. 경로 선택(Path Selection) 우선순위 (BGP 속성):

- Longest prefix (가장 긴 프리픽스 우선).
- AS\_PATH (BGP 경로 속성).

### 2. 로컬 우선순위(Local Preference) BGP 커뮤니티 태그:

- Low Priority**: 7224:7100 .
- Medium Priority**: 7224:7200 .
- High Priority**: 7224:7300 .

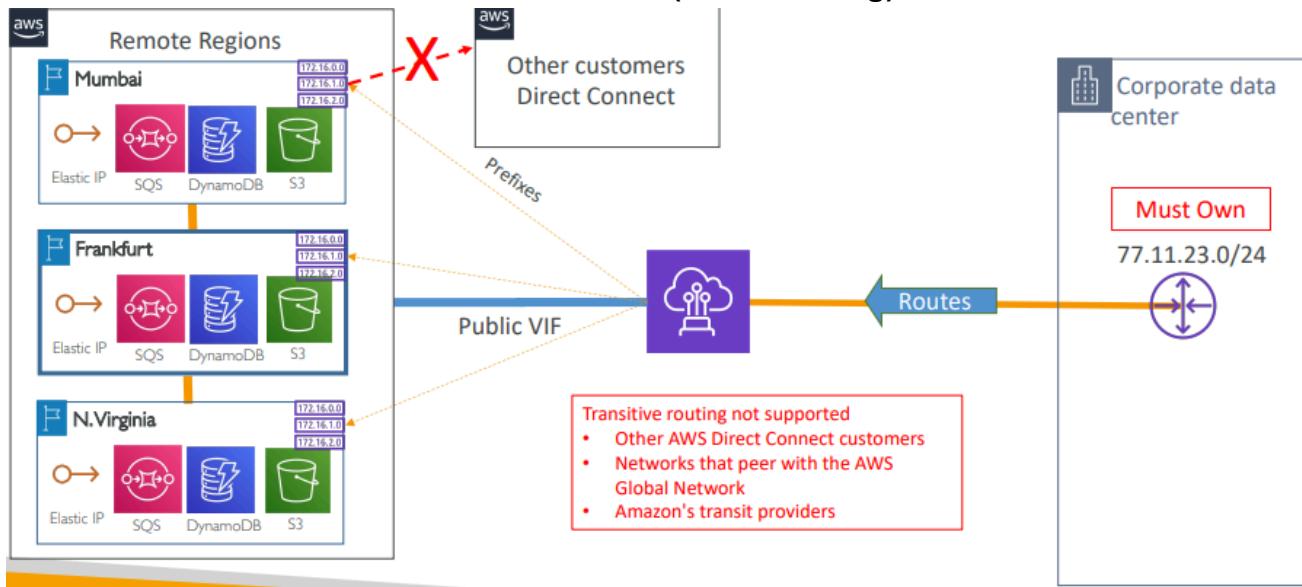
## Public VIF Routing Policies

## Inbound (온프레미스 → AWS)

- 광고할 공용 프리픽스를 명확히 지정해야 하며, 소유권과 인터넷 등록 필요.
- 트래픽은 Amazon 공용 프리픽스로만 전달 가능.
- 연결 간 전이적 라우팅(Transitive Routing) 불가.
- AWS는 인바운드 트래픽에 대해 패킷 필터링 수행.

## Outbound (AWS → 온프레미스)

- Longest Prefix Match와 AS\_PATH를 사용하여 라우팅 제어 가능.
- AWS는 모든 로컬 및 원격 리전의 공용 프리픽스를 광고하며, NO\_EXPORT BGP 커뮤니티 태그와 함께 제공.
- 추가적인 BGP 커뮤니티 태그: 7224:8100, 7224:8200.
- 광고된 프리픽스는 Direct Connect 연결을 넘어서는 네트워크로 전파될 수 없음.
- 여러 Direct Connect 연결 간 트래픽 부하 분산(Load Sharing)을 위해 ECMP를 활용 가능.



## Multiple DX connections traffic routing scenarios using Routing policies for Public VIF

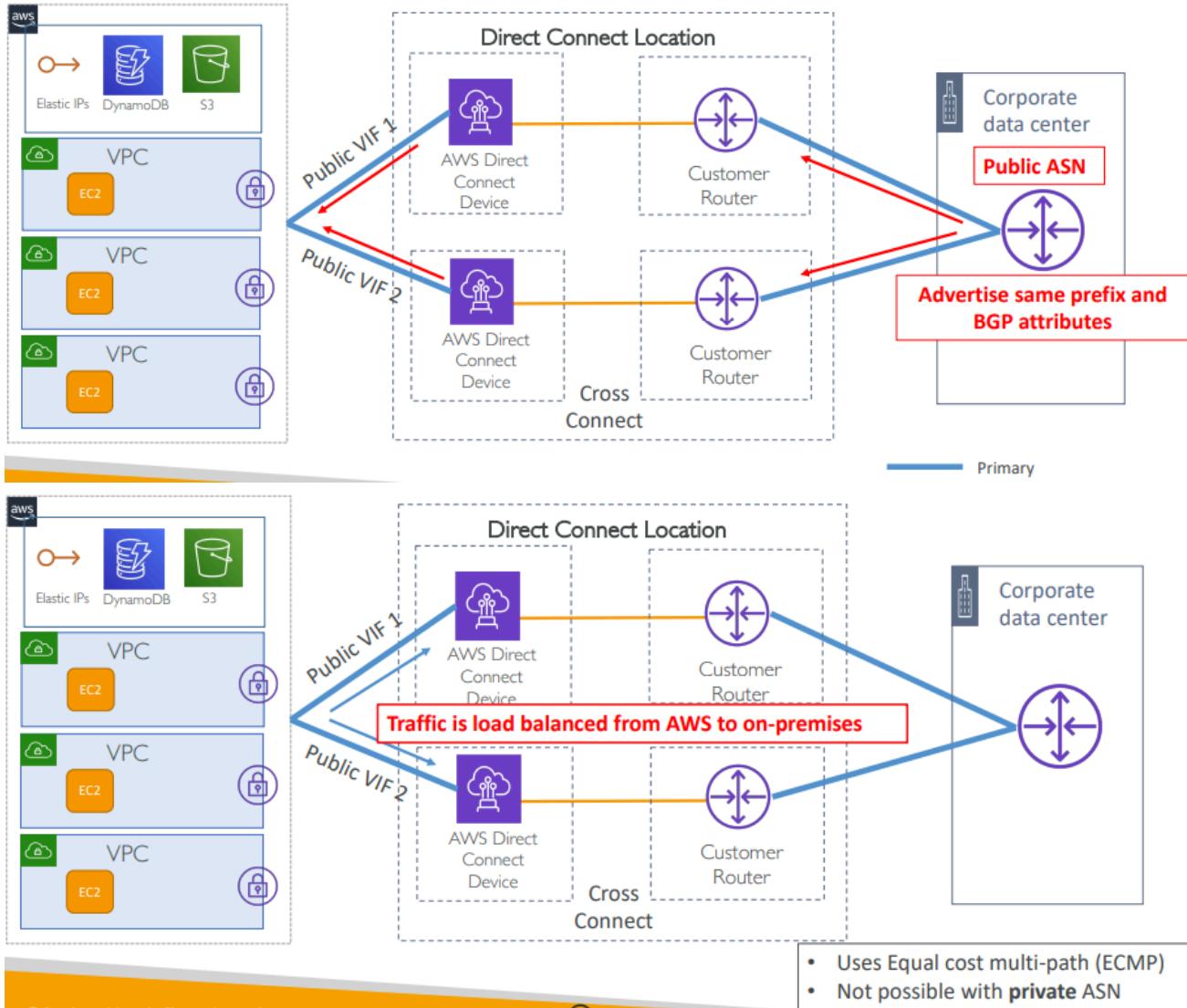
### Active-Active connection using Public VIF

#### Public ASN 사용 시

- 고객 게이트웨이는 동일한 BGP 속성으로 동일한 프리픽스를 두 개의 Public Virtual Interface(Public VIF)에 광고해야 함.
- 이 구성은 두 개의 Public VIF를 통해 트래픽을 부하 분산(Load Balancing)함.

#### Private ASN 사용 시

- Public VIF에서 부하 분산(Load Balancing)은 지원되지 않음.



## Active-Passive Connection using Public VIF

### Public ASN 사용 시

- 고객 게이트웨이는 두 BGP 세션에서 동일한 프리픽스(소유한 공용 IP 또는 네트워크)를 광고해야 합니다.
- 보조 연결(Secondary Connection)에서는 추가 AS\_Path prepends를 사용하여 온프레미스 공용 프리픽스를 광고합니다.
- Local Preference(local-pref)를 증가시켜 온프레미스 라우터가 항상 올바른 경로(Primary Connection)를 선택하도록 보장합니다.

### Private ASN 사용 시

- Primary Connection에서 더 긴 프리픽스를 사용합니다.  
 - 예: Primary Connection에서 두 개의 프리픽스 (x.x.x.0/25 및 x.x.x.128/25)를 광고하

고, Secondary Connection에서는 하나의 프리픽스( x.x.x.0/24 )를 광고합니다.

