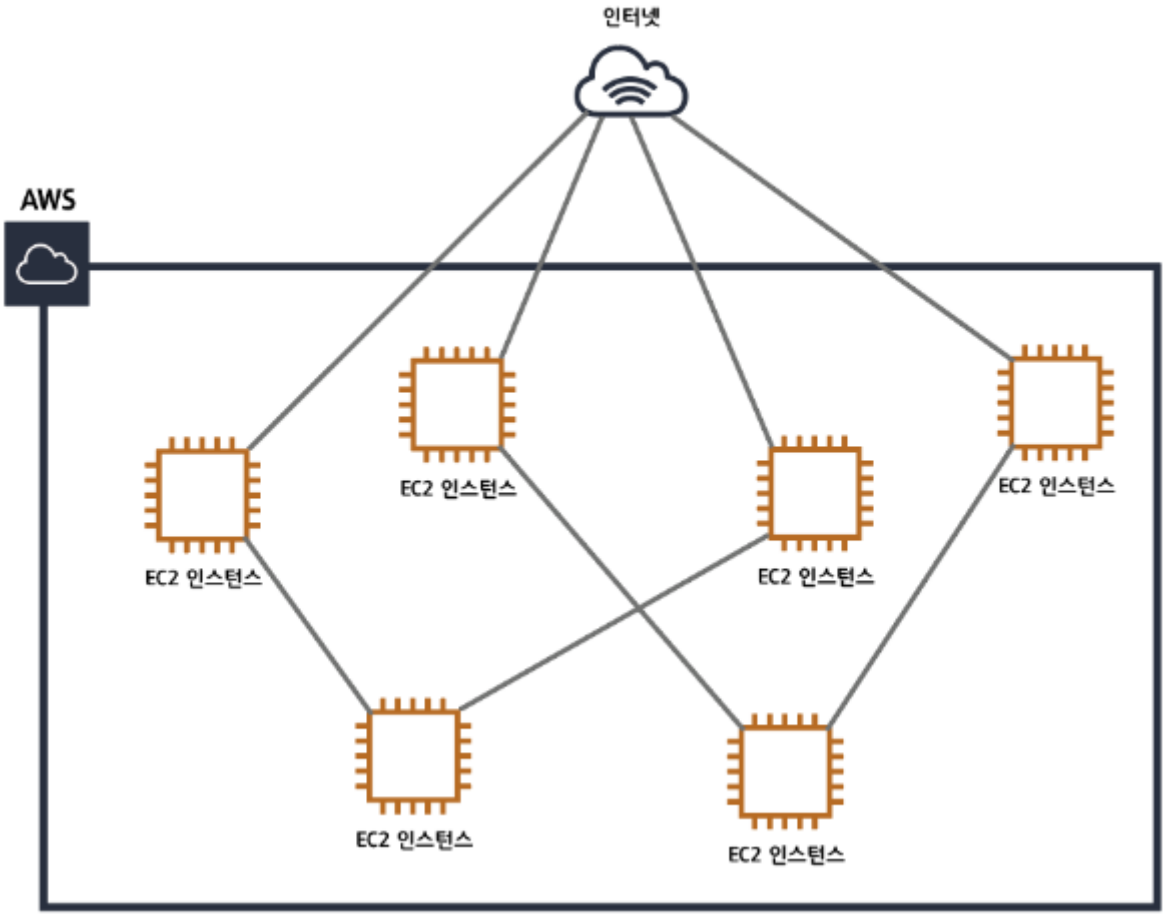
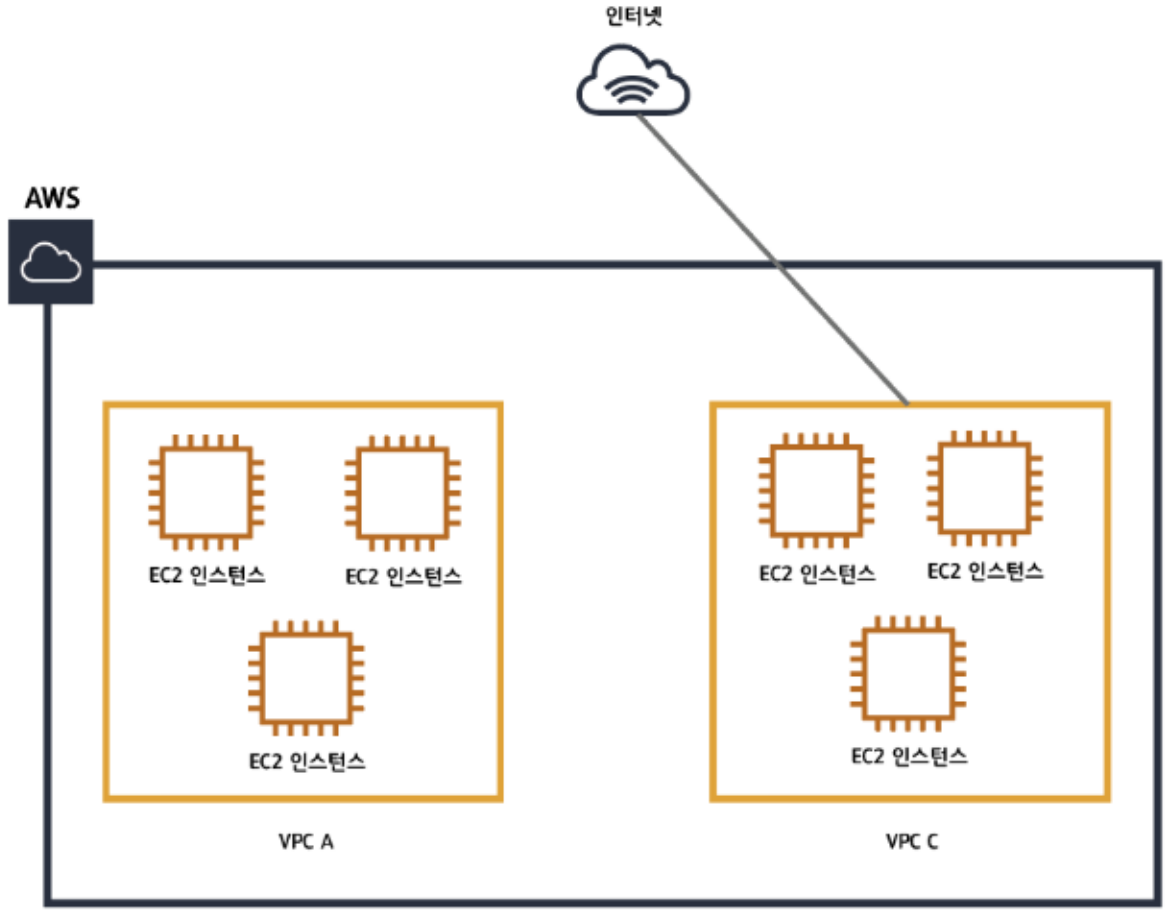
[AWS] 4. VPC (Virtual Private Cloud)

- References
  - AWS 공인 솔루션스 아키텍트 스터디 가이드 - 어소시에이트 3/e - 4장
  - https://inpa.tistory.com/entry/WEB-IP-클래스-서브넷-마스크-서브넷팅-총정리 (IP / Subnet)
  - https://inpa.tistory.com/entry/WEB-🌐-CIDR-이-무얼-말하는거야-⇛-개념-정리-계산법 (CIDR 개념)
  - https://inpa.tistory.com/entry/AWS-📚-아마존-웹-서비스-구조-Region-AZ-Edge-Location-Cache-완벽-정리#엣지_로케이션_(Edge_Location)
  - https://medium.com/harrythegreat/aws-가장쉽게-vpc-개념잡기-71eef95a7098 (VPC 쉬운 개념 정리)
  - https://inpa.tistory.com/entry/AWS-📚-NAT-Gateway-NAT-Instance-대체해서-비용-절약?category=947440 (NAT Gateway → NAT Instance)



VPC 가 없는 구조



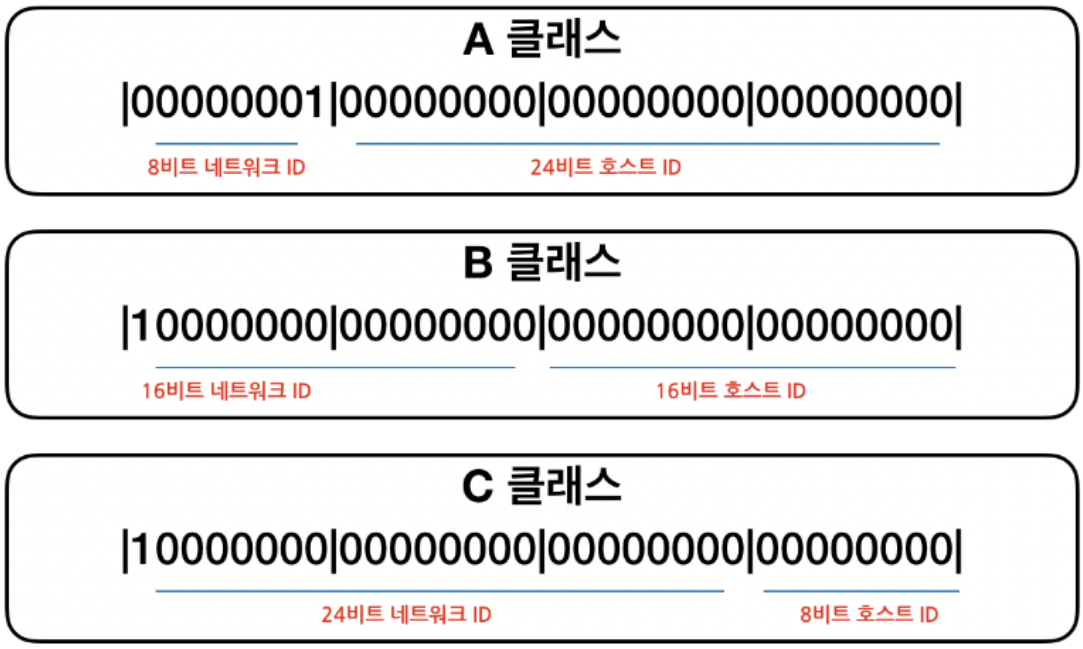
VPC가 있는 구조

※ VPC의 기본 개념

- **VPC (Virtural Private Cloud) 개요**
  - AWS 사용자 전용 가상 네트워크로, 개인 네트워크 망 데이터 센터로 이해하면 된다.
  - 모든 서비스에 VPC를 적용하도록 강제 하고 있다.
  - 인스턴스를 하나의 VPC로 네트워크를 구분하여, VPC 별로 필요한 설정을 통해 인스턴스에 네트워크 설정을 적용할 수 있게 되었다.
  - 위 그림처럼 VPC 없는 구조에서 인스턴스 추가 삭제에 따른 네트워크 문제를 해결하기 위해 등장한 가상 네트워크 개념

※ IP / Subnet / CIDR 기본 개념

- **IP**
  - IP 란 장치를 식별 할 수있는 고유 주소
  - 네트워크 ID + 호스트 ID 로 구성된다.
  - IPv4
    - IP 주소를 8비트씩 4등분, 각각을 옥텟 이라고 부른다.
    - 옥텟 별로 IP를 A, B, C 클래스로 나눈다.



    - 가장 첫번째 호스트 주소, 마지막 주소는 사용할 수 없다 (네트워크, 브로드캐스트 주소)

- **서브넷 (Subnet)**
  - 클래스로 나누어 할당하는 것이 비효율 적이기에 네트워크 장치수에 따라 효율적으로 사용 가능한 서브넷 개념이 등장
  - 서브넷이란 하나의 네트워크를 분할시켜 나눈 작은 네트워크

- 이런 서브넷 네트워크 만들기 위해 분할하는 것을 서브네팅 이라 한다.
- 이 서브네팅은 서브넷 마스크를 통해 계산되어 수행 된다.

- **서브넷 마스크 (Subnet Mask)**
  - 32비트 2진수로 표현되며, 연속된 1과 연속된 0 으로 구성된다.
  - 11111111.11111111.11111100.00000000 처럼 1이 연속되거나 아닌 형태만 가능하다.
  - A클래스의 디폴트 서브넷 마스크 255.0.0.0
  - B클래스의 디폴트 서브넷 마스크 255.255.0.0
  - C클래스의 디폴트 서브넷 마스크 255.255.255.0

- **prefix 표현**
  - 서브넷 마스크를 간소화 하여 표현하기 위한 방법
  - 예를들어 /24 라는 뜻은 앞에서부터 1의 갯수가 24개라는 뜻 (/0 ~/32 까지 가능하다)

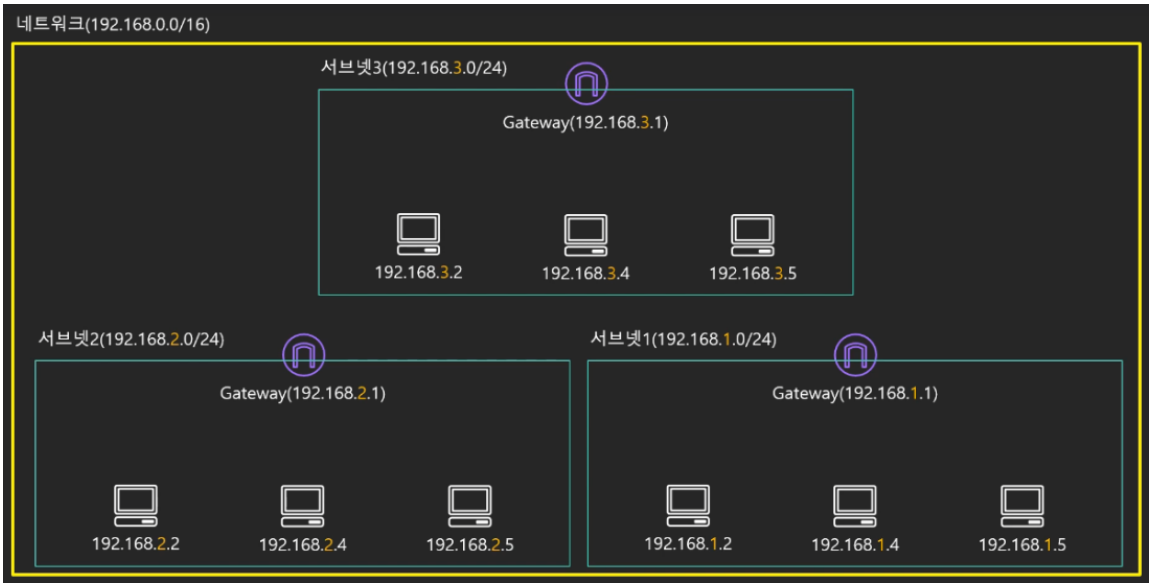
- **서브네팅**
  - 서브넷 구분비트를 호스트 ID 비트의 제일 왼쪽 비트부터 할당 하여 호스트를 나눔

- **슈퍼네팅**
  - 서브네팅의 반대의 개념으로 네트워크를 합치는 것이다.
  - 서브넷 마스크를 이동시키면 된다 = Prefix를 감소 시킨다.
  - Prefix 숫자가 감소한다는 것은 호스트ID 갯수가 늘어난다는 뜻

- **CIDR (사이더)**
  - Classless Inter-Domain Routing 으로 클래스 없는 도메인 간 라우팅 기법 이다.
  - 서브네팅과 유사하지만 서브네팅의 상위 개념
  - CIDR는 서브네팅 슈퍼니팅 등 IP를 나누고 합치는 기법을 모두 일컫는다.

- **CIDR 표기법**
  - 네트워크 정보를 여러개로 나누어진 Sub-Network들을 모두 나타낼 수 있는 하나의 Network로 통합해서 보여주는 방법이다
  - 한줄 표기법으로 네트워크 범위를 추측 또는 측정 할 수 있다.
  - 예) **192.168.10.70/26**
    - 서브넷 마스크 255.255.255.192
    - 256 - 192 = 64 - 2(네트워크, 브로드캐스트 주소 제외) 개의 호스트 ID를 가질 수 있다.
    - 256 / 64 = 4개의 서브넷 네트워크를 가진다.
    - 각 62개씩 호스트 ID를 지닌 4개의 네트워크로 분리되어
    - 192.168.10.70 은 두번째 네트워크인 (192.168.10.64 ~ 192.168.10.127) 에 속해 있다는 것을 알 수 있다.

- **VPC 의 CIDR (IPv4)**
  - VPC CIDR 지정 시 어떤 IP 범위라도 사용할 수 있지만, 다른 퍼블릭 인터넷 주소와 충돌을 피하기 위해 RFC 1918 범위를 사용할 것을 권장한다.
    - 10.0.0.0~10.255.255.255 (10.0.0.0/8)
    - 172.16.0.0~172.32.255.255 (172.16.0.0/12)
    - 192.168.0.0~192.168.255.255 (192.168.0.0/16)
  - VPC의 CIDR 범위는 /16 에서 /28 까지 가능하다.
  - VPC 생성 후 기본 CIDR 블록은 변경할 수 없으므로 VPC를 생성하기 전에 주소 요구사항을 신중히 검토 해야 한다.



- AWS 에서 보조 CIDR 블록이라 일컫는 CIDR 블럭은 일종의 서브넷 이다. (위 예시)
  - 네트워크 (VPC CIDR 블록) : 192.168.0.0/16 으로 기본 CIDR 블록 지정
  - 서브넷 1,2,3 (보조 CIDR 블록)

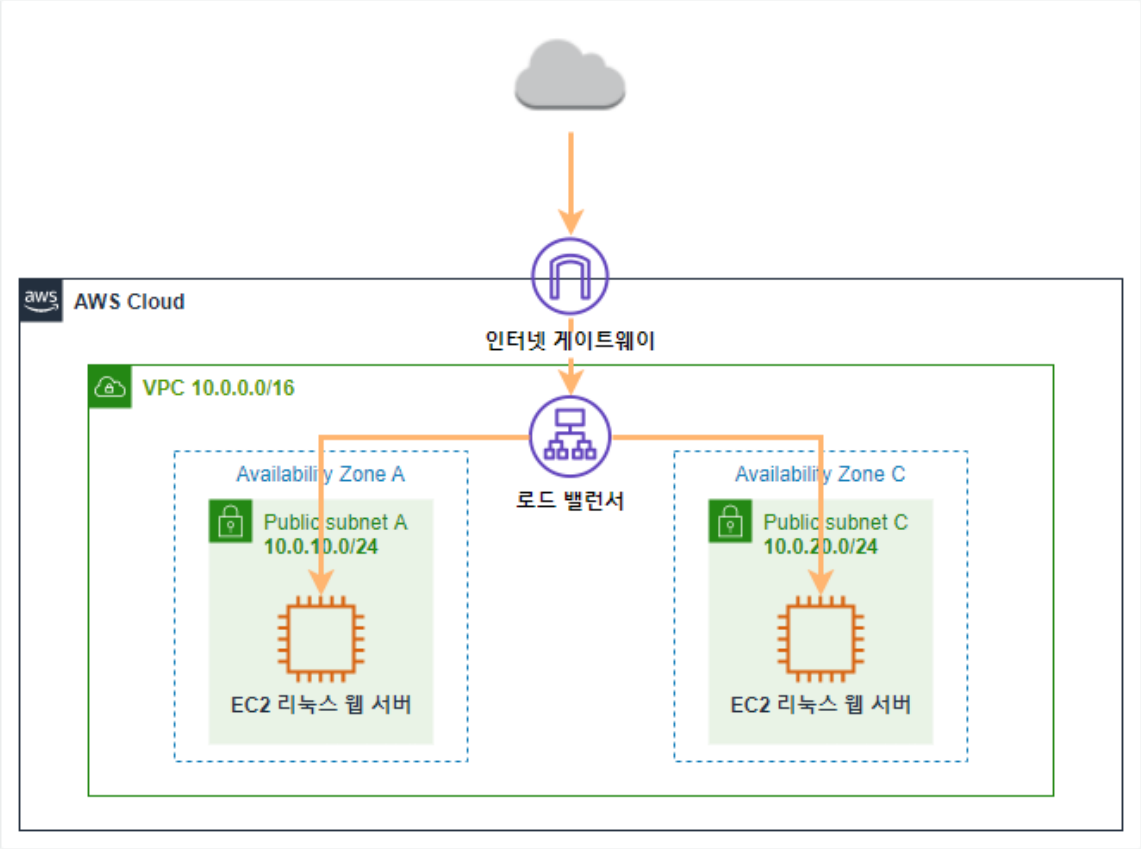
- AWS 에서는 자체 클라우드에서 설정해 사용하고 있는 IP 가 있어 총 5개의 IP가 자동으로 할당 된다.
  - 첫번째 주소 : 네트워크 주소
  - 두번째 주소 : AWS VPC 라우터 용으로 예약 (Default Gateway)
  - 세번째 주소 : DNS 서버 주소, DNS 서버의 IP 주소는 기본 VPC 네트워크 범위에 2를 더한 주소이다. CIDR 블록이 여러개인 VPC인 경우 DNS 서버의 IP 주소가 기본 CIDR 에 위치한다.
  - 네번째 주소 : AWS 에서 앞으로 사용하려고 예약한 주소 (예비 주소)
  - 마지막 주소 : 네트워크 브로드캐스트 주소

- **VPC 의 IPv6 CIDR 블록**
  - IPv4의 기본 CIDR 지정과 달리, IPv6에서는 CIDR 를 지정할 수 없다.
  - 대신 AWS에 요청 시 VPC에 IPv6 CIDR를 할당하며 이렇게 할당받은 CIDR는 글로벌 유니케스트 IPv6 주소공간에서 퍼블릭 라우팅이 가능한 IP 프리픽스로 사용될 수 있다.

- IPv6 VPC 기본 CIDR 의 프리픽스 길이는 항상 /56 이다.
- IPv6 서브넷의 접두사 길이는 /64로 고정되어 있다.
- IPv6만 사용 할 계획이라도 서브넷에는 반드시 IPv4 CIDR 블록을 할당해야 한다.

- **VPC 의 서브넷**
  - VPC 안 로직 컨테이너
  - 인스턴스간 트래픽 유입 유출을 제어하고 기능별로 조직화 할 수 있다.
  - 인터넷 접속이 가능한 퍼블릭 웹서버용 서브넷을 하나 만들고, 웹 인스턴스만 접속 가능한 데이터베이스 전용 서브넷을 추가하는 구성이 가능하다.
  - 서브넷은 전통적인 가상 LAN(VLAN) 과 유사한 개념의 네트워크 요소
  - 서브넷에 인스턴스를 생성후 다른 서브넷으로 옮길 수 없다.
    - 만약 옮겨야 할 때는 기존 인스턴스의 EBS 볼륨 스냅샷 생성, AMI 생성 그리고 해당 AMI를 사용하여 원하는 서브넷에서 새 인스턴스를 시작해야 한다.

- **AZ, Availability Zone (가용 영역)**
  - 리전 내 개별 데이터센터와 비슷한 개념
  - 서브넷은 하나의 가용영역내에서만 존재할 수 있다.
  - AWS 리전의 가용영역은 서로 연결되어 있으며, 하나의 가용영역에 장애가 발생하더라도 다른 영역에 그 영향이 미치지 않도록 설계되어 있다.



- 위 그림 처럼 하나의 VPC 아래 두개의 서브넷을 가용영역 A, C에 구성 한다면, A 영역에서 장애가 발생하더라도 B영역의 EC2 웹서버는 정상적으로 사용이 가능하다. (고가용성)

- **ENI (Elastic Network Interface)**
  - Elastic Network Interface 란 인스턴스가 AWS 서비스 등 다른 네트워크 리소스와 통신 할 수 있도록 한다.
  - SSH, RDP(Remote Desktop Protocol) 등을 이용해 인스턴스에서 실행되는 OS 와도 통신 가능.
  - 물리적 서버의 네트워크 인터페이스와 같은 기능을 제공한다.
  - 모든 인스턴스는 ENI를 가져야 하며, 이 인터페이스는 하나의 서브넷에만 연결 된다.
  - 각 인스턴스는 기본 프라이빗 IP 주소를 지니며, 인스턴스의 기본 ENI와 연결된다, 삭제 변경 불가능
  - 각 인스턴스에 기본 프라이빗 IP 주소 외에 보조 프라이빗 IP를 할당하여 사용 가능하며, 보조 프라이빗 IP 주소는 기본 ENI와 연결된 서브넷 범위 내에 있어야 한다.

- ENI 는 인스턴스와 독립적으로 존재할 수 있으며, ENI를 생성한 뒤 인스턴스에 부착할 수 있다.
- ‘종료 시 삭제’ 속성을 비활성화 하면 인스턴스 삭제 후에도 ENI가 삭제되지 않는다.

- **성능강화 네트워크 (Enhanced Networking)**
  - ENI 에 비해 고속의 네트워크 처리속도 및 저지연성을 제공한다.
  - 단일 루트 입출력 가상화 (SR-IOV) 기법을 사용
    - SR-IOV 기법 : 동일한 물리적 서버에서 호스팅 되는 다수의 인스턴스가 하이퍼바이저를 우회할 수 있도록 하여 좀더 낮은 CPU 활성화 수준 및 좀 더 높은 네트워크 성능을 제공

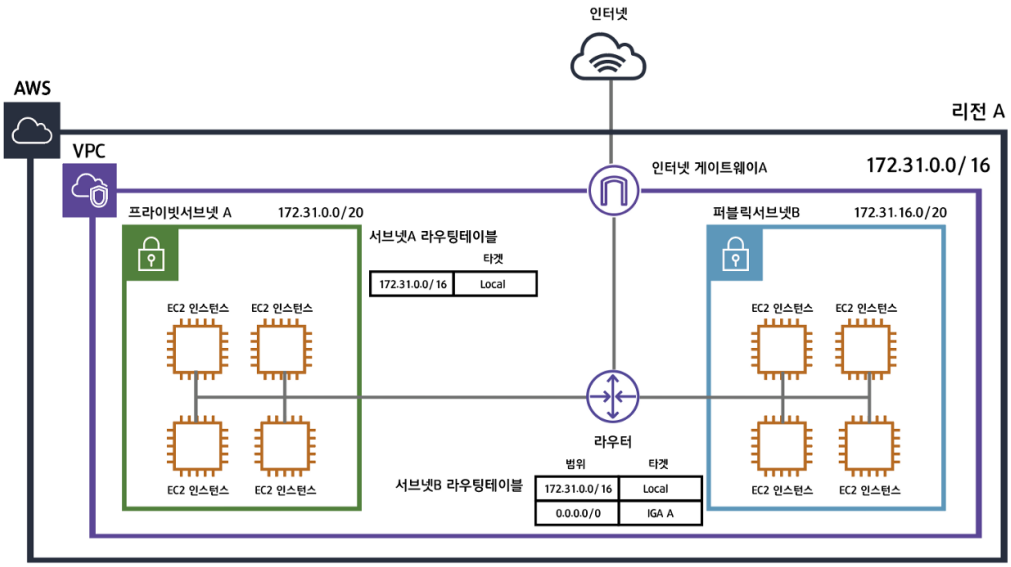
Elastic Network Adapter ENA	100 Gbps 처리속도 제공, 대부분의 인스턴스 타입 지원
Intel 82599 Virtual Function Interface	10 Gbps 처리속도 제공, ENA를 지원하지 않는 일부 인스턴스 타입을 지원한다.

  - 성능 강화 네트워크를 사용하려면 OS에 성능강화 네트워크를 지원할 수 있는 드라이버가 설치 되어 있어야 하며, Amazon Linux 및 Ubuntu HVM AMI 에는 ENA 지원 기능이 기본적으로 탑재되어 있다.

- **라우팅 테이블과 라우터**
  - VPC 내의 트래픽의 유입, 유출, 이동을 제어하기 위해 라우트 테이블에 저장된 라우트를 사용
  - 사용자에 의한 환경설정이 필요한 기존 물리적 혹은 가상 라우터와 달리 VPC 아키텍처는 IP 라우팅을 소프트웨어 함수로 구현한 내재된 라우터 (Implied Router) 의 특징을 지닌다.
  - VPC에는 인터페이스 IP 주소를 설정할 가상 라우터, BGP와 같은 동적 라우팅 프로토콜도 없다.
  - 서브넷은 라우트 테이블 연결 없이 존재할 수 없으며, 서브넷을 커스텀 라우트 테이블에 명시적으로 연결하지 않으면, AWS가 암묵적으로 해당 서브넷을 기본 라우트 테이블에 연결한다.
  - 라우트는 라우트 테이블과 연결된 서브넷 내에서의 트래픽 유입 및 유출을 결정
  - 라우트를 생성시에 대상(destination) 주소 IP 프리픽스, 타겟 리소스 요소를 반드시 설정

- 내재된 라우터는 트래픽을 어디로 보낼지 결정할 때 가장 일치도가 높은 범위를 선택하며, 라우트 순서에는 상관하지 않는다.

- **인터넷 게이트웨이**
  - VPC와 인터넷을 연결해주는 Gateway
  - 온프레미스에 설치하는 인터넷 라우터와 유사
  - 차이점은 기존 네트워크에서 코어 라우터의 기본 라우터가 인터넷 라우터의 IP를 가리키도록 되어 있지만, 인터넷 게이트웨이에는 IP주소나 네트워크 인터페이스가 없어, AWS 리소스 ID를 식별용으로 할당 (igw- 로 시작 하는 ID) (→ 아래 그림의 서브넷 라우팅테이블 0.0.0.0/0 IGA A 에 해당한다)



  - 인터넷 게이트웨이를 사용 하려면, 라우트 테이블에 인터넷 게이트웨이를 타겟으로 하는 기본 라우트를 생성해야 한다.

- **보안그룹 (= 방화벽)**
  - 인스턴스 ENI에 대한 트래픽의 유입 또는 유출 여부를 결정한다.
  - 모든 ENI는 최소 하나 이상의 보안그룹에 연결 되어 있어야 한다.

- 하나의 보안그룹을 다수의 ENI에 연결 가능하다.
- 하나의 인스턴스에는 여러개의 ENI 여러개의 보안그룹이 연결 될 수 있다.

	필수 요소	허용 및 규칙 방식
인바운드 규칙	소스, 프로토콜, 포트 범위	기본 거부 규칙을 사용 (Default-deny), 보안 그룹의 규칙 순서에는 영향이 없다.
아웃바운드 규칙	대상 주소, 프로토콜, 포트 범위	기본 허용 규칙을 사용, 규칙 제거 시 보안그룹은 인스턴스가 외부 다른 어떤 요소에도 접근하지 못하도록 막는다.

- 소스 및 대상 주소에는 CIDR 블록 또는 보안그룹의 리소스 ID가 될 수 있다.
- 소스 보안 그룹은 다른 AWS 계정에 있어도 무방하며, 이 때 소스 보안 그룹에 해당 계정 소유자 ID를 지정하면 된다.
- 상태 저장 방화벽 (Stateful Firewall) 기능을 제공한다.
  - 보안 그룹이 트래픽을 한방향으로 전달하도록 허용한 뒤, 반대 방향의 응답 트래픽을 기억해 두었다 지능적으로 허용하는 것을 의미한다.
  - 보안 그룹은 연결 추적 기능을 이용하여 응답 트래픽의 허용 여부를 결정한다.
- 모든 VPC에는 삭제 불가능한 기본 보안그룹이 포함되어 있다.

- **네트워크 접속 제어 목록 (NACL)**
  - Network Access Control List 는 보안그룹과 마찬가지로 방화벽의 기능을 수행한다.
  - 각 VPC에는 삭제 불가능한 기본 NACL이 있다.
  - NACL은 ENI 가 아닌 서브넷에 연결된다.
  - 서브넷 내 인스턴스간 트래픽 제어에는 보안그룹을 사용해야 한다.
  - 서브넷에는 하나의 NACL만 연결 가능하다.
  - 하나의 NACL은 여러개의 서브넷에 연결이 가능하다.
  - 기본 NACL을 수정하거나 새 NACL을 만들어 연결한다.
  - 보안그룹이 Statefule 속성을 지니는 반면 NACL은 Stateless 속성을 지닌다.
  - NACL을 통과하는 연결상태를 추적하지 않고, 응답트래픽을 자동으로 허용하지 않는다는 점에서 전통적인 스위치, 라우터의 ACL과 유사하다.
  - NACL과 보안그룹이 충돌 시 보안그룹이 더 높은 우선순위를 갖는다.

	필수 요소	허용 및 규칙 방식
인바운드 규칙	규칙번호, 프로토콜, 포트 범위, 소스 CIDR, 동작 (Allow/Deny)	규칙은 규칙 번호의 오름차순으로 처리한다, 규칙번호는 가장 작은 숫자부터 우선된다
아웃바운드 규칙	규칙번호, 프로토콜, 포트 범위, 소스, 동작	stateless 속성에 따라 응답트래픽을 자동으로 허용하지 않는다. 즉 명시적으로 허용해야 한다.

- **Public IP (퍼블릭 IP 주소)**
  - 퍼블릭 인터넷으로 접속 가능한 주소
  - VPC에 인터넷 게이트웨이를 연결해야 한다.
  - 퍼블릭 IP 주소는 기본적으로 DHCP(?) 이므로 인스턴스 재시작 시 새로운 퍼블릭 IP가 할당된다.

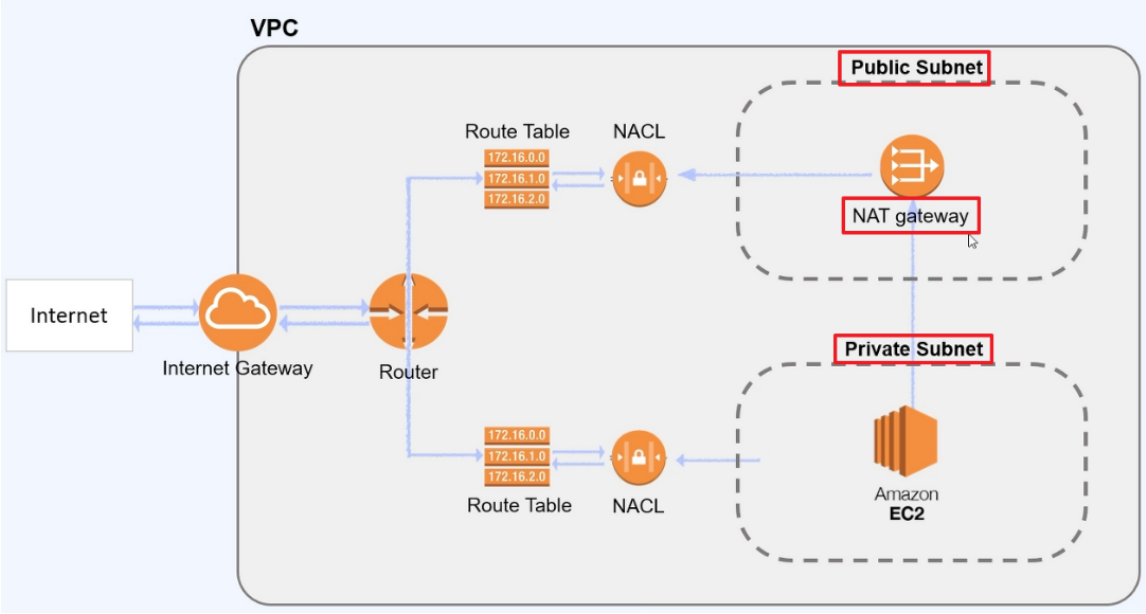
- **Elastic IP (탄력적 IP 주소)**
  - 사용자의 요청에 따라 AWS 가 사용자의 계정에 할당하는 퍼블릭 IP 주소
  - EIP를 처음 생성하면 인스턴스와 연결되지 않은 상태이므로 (독립적), 직접 ENI와 연결후 ENI를 인스턴스와 연결하여 사용하여야 한다.
  - EIP를 ENI와 연결한 뒤에는 ENI를 삭제하거나 EIP를 해제하지 않는 한 연결이 지속
  - EIP는 리전 단위로 제공된다.
  - EIP를 AWS 계정이 보유한 퍼블릭 주소로 전달 가능하다. (BYOIP - Bring your own IP address 라 부르며 리전당 최대 다섯개의 주소목록을 가져올 수 있다.)

- **AWS 글로벌 엑셀레이터**
  - AWS의 여러 리전에 리소스가 분리되어 있다면 리전별 여러개의 EIP를 관리하기 위해 사용
  - AWS 글로벌 엑셀레이터는 어디에든 연결할 수 있는 두개의 정적 IPv4 주소를 제공하여 이를 통해 어떤 리전에 있는 리소스와도 연결 가능하다.
  - 애니캐스트 주소 anycast address 라고도 부른다.
  - 접속 포인트 (POP) 를 연결한 정적 주소 체계

- 특정 POP이 작동하지 않을 시 트래픽을 자동으로 다른 POP로 라우팅 하므로 전체 서비스엔 영향이 없다.

- **네트워크 주소 변환 (NAT)**
  - 사설 네트워크에 속한 여러 대의 호스트가 하나의 공인 IP 주소를 사용하여 인터넷에 접속하기 위함
  - 인터넷 게이트웨이가 퍼블릭 IP주소를 ENI의 프라이빗 IP 주소로 매핑하는 과정
  - NAT 작업은 인스턴스가 퍼블릭 IP를 지닌 경우, 인터넷 게이트웨이에서 자동으로 이루어 지며, 사용자가 변경 불가능하다.

- **NAT 게이트웨이**
  - 인터넷 접속이 가능한 Public 서브넷에 NAT 게이트웨이를 생성하고, Private 서브넷이 외부 인터넷으로 나아갈 경우에만 라우팅을 추가



  - NAT Gateway 는 내부에서 외부로의 접속만 가능하며, 외부에서 NAT Gateway를 이용하여 접속하는 것은 불가능 하다.
  - 외부 인터넷 연결에 대한 보안 문제 해결
  - NAT Gateway 는 인터넷 게이트웨이 없이 동작할 수 없다.
  - NAT Gateway는 기본적으로 트래픽이 나갈때 데이터 처리 요금이 발생하고, 계속 돌려놓고 있어도 시간당 요금이 청구되는 꽤 비싼 서비스이다.

- NAT Gateway를 생성할 때 EIP를 할당해서 연결해야 하며, 퍼블릭 서브넷에 생성한다.

- **NAT 인스턴스**
  - NAT Gateway의 비용을 절약하기 위해 EC2를 NAT로 사용하는 기술
  - NAT Gateway와 달리 대역폭 요구가 증가하더라도 자동으로 확장되지 않는다.
  - 적절한 성능을 갖춘 인스턴스 유형을 선택하는것이 중요하다.
  - NAT 인스턴스는 ENI를 지니고 보안그룹을 적용해야 하므로, 직접 퍼블릭 IP 주소도 할당해야 한다.
  - 직접 NAT 인스턴스의 ENI에서 소스/대상 주소 옵션을 비활성화 해야한다.
  - NAT 인스턴스의 이점은 배스티온 호스트 (Bastion Host) 또는 점프 호스트 (Jump Host) 로 사용하여 퍼블릭 IP가 없는 인스턴스에 연결할 수 있다는 점이다. (NAT 게이트웨이로는 불가능)
  - 인스턴스나 AZ 장애시 치명적 (다른 NAT 인스턴스를 가리키도록 하는 것이 불가능)

NAT Instance	NAT Gateway
단일 인스턴스	AWS에서 제공하는 서비스
= EC2	= 서비스
꺼지면 죽음	꺼져도 죽지않음(고가용성 보장)
보안그룹 영향 받음	보안그룹 영향 받지않음
예전 스타일	요즘 스타일
Source/Destination을 해제해야 함	
Bastion을 겸할 수O	Bastion을 겸할 수X
Public Subnet에 있어야 함	Public Subnet에 있어야 함

- **VPC 피어링**
  - 프라이빗 AWS 네트워크를 통해 하나의 VPC에 포함된 인스턴스가 다른 VPC에 포함된 인스턴스와 소통할 수 있다.
  - Point to Point 연결로, 두 VPC간 하나의 피어링만 설정 가능하며 CIDR 블록은 겹치지 않아야 한다.
  - 인스턴스간 통신만 허용하며, 인터넷게이트웨이 NAT 디바이스는 공유할 수 없지만 NLB(Network Load Balancer) 는 공유할 수 있다.
  - 트래픽이 양방향으로 소통되도록 두 VPC에 새로운 라우팅 규칙이 추가된다.
  - 각 라우트 대상 주소 프리픽스는 대상주소 VPC의 범위내에 있어야 하며, 각 라우트 타겟은 pcx- 로 시작하는 피어링 연결 ID로 한다. (각 VPC의 타겟은 같은 피어링 ID로 동일)
  - 일부 리전간 VPC 피어링을 사용할 수 없고, IPv6를 지원하지 않는다.
  - 리전 간 피어링 연결의 최대 전송 단위는 1,500바이트

- **하이브리드 클라우드 네트워킹**
  - 프라이빗 속성을 지니고 인터넷과 연결성 없이, AWS의 온프레미스와 VPC의 프라이빗 연결 서비스는 다음과 같다
    - VPN
    - AWS Transit Gateway
    - AWS Direct Connect

- **VPN**
  - VPG(Virtual Private Gateway)라 부르는 VPC 리소스를 구성한 뒤, 온프레미스 라우터 또는 방화벽 등 고객 게이트웨이를 구성하게 되며, 이를통해 VPG로 암호화 VPN 터널이 생성된다.
  - VPG는 AES 256 비트 및 AES 128 비트 암호를 지원한다.
  - 대량의 VPC를 온프레미스 네트워크에 연결 시, 또는 다수의 온프레미스 네트워크를 하나의 VPC에 연결 시 VPN은 많은 수작업을 필요로 하고 실수가 발생할 확률

이 높아 이런 경우 AWS Transit Gateway를 사용하는 것이 좋다.

- **AWS Transit Gateway**
  - Direct Connect 링크와 VPN을 사용해서 다수의 VPC 및 다수의 온프레미스 네트워크를 연결할 수 있도록 해주는 고가용성 서비스
  - Transit Gateway 라우트 테이블을 통해 부착된 요소를 서로 연결
  - 주요 용도

중앙화 라우터	중앙화 라우터로 모든 VPC 및 온프레미스 트래픽을 제어
격리라우터	하나의 Transit Gateway에 다수의 격리 VPC를 생성하여 VPC간 격리성을 유지
공유서비스	하나의 VPC에서 Active Directory, LLDP 등의 공유 서비스를 호스팅 하는 경우, Transit Gateway를 이용하여 격리 보안이 유지된 상태에서 공유환경 구성 가능
피어링	서로 다른 리전간 피어링이 가능하다.
멀티캐스트	VPC간 멀티캐스트 지원
블랙홀 라우트	특정 라우트를 차단하고 싶을 때 Transit Gateway 라우트 테이블에 블랙홀 엔트리를 추가하여 트래픽을 차단.

- **AWS Direct Connect**
  - AWS 리소스에 대한 프라이빗, 저지연성 연결을 제공
  - 인터넷을 우회해서 접속하는 방법을 제공하여, 문제발생 가능성을 낮추고 광대역 인터넷을 사용할 수 있도록 해준다.
  - 대량의 데이터 또는 실시간 데이터 전송, 퍼블릭 인터넷으로 데이터를 전송해서 안될때 유용하다.

전용(Dedicated)	물리적 단일 연결, Direct Connect 지점에 자체 장비를 추가해야 한다. 1Gps ~ 10Gps 연결속도를 선택 가능하다.
호스트(Hosted)	50Mbps ~ 10 Gbps 연결을 지원하는 호스트 연결타입은 자체장비를 추가할 여력이 없거나 1Gps 미만의 연결속도로 충분할 때 사용, 데이터센터 또는 사무실과 Direct Connect 지점을 잇는 라스트 마일 연결을 제공한다.

- **Direct Connect Gateways**
  - 리전 내 여러 VPC를 하나의 연결지점에서 접속할 수 있도록 해주는 글로벌 리소스
  - AWS 측에서는 Transit Gateway 또는 VPG가 Direct Connect Gateway 역할
  - 사용자 측에서는 Direct Connect Gateway가 온프레미스 장비로 BGP 세션을 유지하며 IPv4, IPv6 라우트 프리픽스를 전파 및 수신한다.

- **가상 인터페이스**
  - Direct Connect 연결 방식에 따라 가상 인터페이스를 생성해서 사용하며, 세가지의 가상 인터페이스를 제공한다

프라이빗 가상 인터페이스	단일 VPC 내 EC2 또는 RDS 인스턴스 등과 같은 리소스의 프라이빗 IP 주소에 연결할 수 있다.
퍼블릭 가상 인터페이스	퍼블릭 엔드포인트를 지닌 S3 또는 DynamoDB 와 같은 AWS 서비스의 퍼블릭 IP 주소에 연결할 수 있다, 온프레미스 애플리케이션을 퍼블릭 엔드포인트를 이용해서 AWS 서비스에 연결하려는 경우 유용
트랜싯 가상 인터페이스	하나 이상의 AWS Transit Gateway에 연결한다. 1Gbps 이상의 속도를 제공

- **고성능 컴퓨팅 (HPC - High Perfomance Computing)**
  - 고성능 컴퓨팅은 집약적인 워크로드를 다수의 인스턴스 (HPC 클러스터) 를 이용하여 동시에 병렬적으로 처리하는 연산 패러다임
  - HPC는 보통 긴밀하게 연결된 HPC 클러스터를 의미하며, 보통 고속, 저지연성, 고신뢰성 네트워크 연결을 기본 속성으로 한다.

Loosely Coupled (느슨한 연결 클러스터)	개별 인스턴스가 독립적으로 처리할 수 있도록 다시 세분화, 이미지 프로세싱등의 업무에 주로 활용되며, 하나의 인스턴스는 다른 인스턴스와 완전히 별개의 요소로 작동하며, 고속의 통신 등을 필요로 하지 않아 별개의 클러스터 플레이스먼트 그룹에 배치 가능하다.
Tightly Coupled (긴밀한 연결 클러스터)	여러개의 인스턴스가 단일 슈퍼컴퓨터와 같이 작동, 인스턴스는 고속의 네트워크로 서로 연결, 동일 클러스터 플레이스먼트 그룹에 배치하는 작업이 필요하다, 하나의 변수가 다른 변수에 영향을 미치는 복합적인 시뮬레이션 작업에 사용, 머신러닝, 기상예측 등

- **일래스틱 패브릭 어댑터 (EFA)**
  - Elastic Fabric Adapter 는 전통적인 TCP/IP 네트워크 연결성을 지원하는 특수한 형태의 ENA
  - Libfabirc API 를 이용해 OS 의 기본 TCP/IP 스택을 우회해 EFA 에 직접 접속할 수 있도록 해주므로 HPC 애플리케이션을 위한 높은 처리량 및 저지연성을 제공
  - ENA란? Elastic Network Adapter의 약자로 최신의 프로세스에 최적화 해 대역폭 및 성능을 향상(호스트의 프로세스의 부하 감소)시킨 네트워크 아답터 모듈이다.
  - 고가의 인스턴스 타입만 지원하고 하나의 인스턴스에는 하나의 EFA만 부착할 수 있다.

- **AWS ParallelCluster**
  - 리눅스 기반 HPC 클러스터를 자동으로 관리하며, 클러스터 인스턴스 프로비저닝 작업을 수행하고, 15GB 공유 파일시스템을 자동으로 생성한다.
  - 공유 파일시스템은 마스터 인스턴스의 EBS 볼륨에 저장되며, NFS를 통해 다른 인스턴스에 공유
  - NFS 외에도 Amazon EFS, Amazon FSx 를 공유 파일시스템으로 활용 가능하다.
  - ParallelCluster는 AWS Batch 를 이용해 배치 스케줄러를 생성한다. 사용자가 배치 스케줄러에 HPC 컴퓨팅 잡을 제출하면, ParallelCluster는 작업에 맞춰 자동으로 클러스터 확장 또는 축소를 수행한다.