03. AWS 네트워킹 서비스

3.1. 네트워킹이란

3.1.1. 네트워킹 정의

네트워킹(networking)은 '서로 연결한다'는 뜻으로 서로 간에 의사소통(communication)을 하는 환경이다. 일상에서도 자주 사용하지만, IT에서는 IT 자원 간 연결하여 통신하는 환경을 의미한다. 어떤 IT 서비스를 제공하고자 한다면, 그 전에 먼저 다양한 IT 자원이 통신할 수 있는 네트워킹 환경이 구성되어야 한다. 이런 측면에서 네트워킹은 IT 영역에서 필수이자 모든 IT 환경의 밑마탕이 되는 초석이라고 볼 수 있다.

3.1.2. 네트워킹 요소

IT 자원들이 통신할 때는 다양한 네트워킹 요소 간에 복잡한 과정을 거친다. 이런 복잡한 과정을 좀 더 쉽게 이해할 수 있도록 계층별로 분류하는 모델이 여러 개 있다. 그중 대표적인 모델은 국제표준화기구(ISO)에서 개발한 OSI 7계층 모델로, 네트워킹 통신 구조를 계층 일곱 개로 분류하는 방식이다.

- 1계층(물리 계층, physical layer)
  네트워크 하드웨어 전송 기술을 이루는 계층으로, 물리적으로 연결된 매체가 서로 데이터를 송수신할 수 있게 연결하고 유지하는 역할을 한다.
- 2계층(데이터링크 계층, datalink layer)
  물리 계층에서 송수신되는 정보의 오류와 흐름을 제어한다.
- 3계층(네트워크 계층, network layer)
  데이터를 목적지까지 전달하는 계층으로, 최적의 통신 경로를 찾는다.
- 4계층(전송 계층, transport layer)
  종단의 대상 간에 데이터 전송을 다루는 계층으로, 데이터 전송의 유효성과 효율성을 보장한다.
- 5계층(세션 계층, session layer)
  종단의 대상 간 응용 프로세스 통신을 관리하는 방법으로, 데이터 통신을 위한 논리적인 연결을 담당한다.
- 6계층(표현 계층, presentation layer)
  데이터 형식에 차이가 있을 때 데이터를 서로 이해할 수 있는 형태로 변환하는 역할을 한다.
- 7계층(응용 계층, application layer)
  응용 프로세스와 직접 연계하여 실제 응용 프로그램을 서비스하는 역할을 한다.

IP 주소와 서브넷

IP 주소
IP(Internet Protocol) 주소는 인터넷상에서 IT 자원을 식별하는 고유한 주소이다. IP 버전에는 IPv4와 IPv6 두 가지가 있으며, 일반적으로 IPv4를 더 많이 사용한다. IPv4는 10진수 또는 2진수 네 자리로 되어 있으며, 각 자리는 온점(.)으로 구분해서 표현한다.

퍼블릭 IP 주소와 프라이빗 IP 주소
IP 주소는 통신 용도에 따라 퍼블릭 IP 주소와 프라이빗 IP 주소로 분류된다. 퍼블릭 IP 주소는 실제 인터넷에서 사용하려고 인터넷 서비스 공급자(ISP)에서 제공하는 유일한 공인 IP 주소이다. 반면 프라이빗 IP 주소는 인터넷이 아닌 독립된 네트워크 내부에서만 사용하려고 네트워크 관리자가 제공하는 사설 IP 주소이다.

프라이빗 IP 주소는 세 가지 클래스로 범위가 정해져 있다.
- 클래스 A: 10.0.0.0 ~ 10.255.255.255
- 클래스 B: 172.16.0.0 ~ 172.31.255.255
- 클래스 C: 192.168.0.0 ~ 192.168.255.255

고정 IP 주소와 유동 IP 주소
IP 주소는 할당하는 방식에 따라 고정 IP 주소와 유동 IP 주소로 분류된다. 고정 IP 주소는 네트워크 관리자가 수동으로 할당하는 방식이며, 유동 IP 주소는 특정 서버가 IP 주소 범위에 따라 동적으로 할당하는 방식이다.
유동 IP 주소는 DHCP(Dynamin Host Configuration Protocol) 프로토콜을 통해 주소를 제공하는 서버와 주소를 할당받는 클라이언트로 구성되며, IP 주소를 임대(lease)하는 형태를 취한다.

서브넷과 서브넷 마스크
모든 네트워크는 하나의 네트워크로만 구성되어 있지 않다. 주체와 목적에 따라 부분 네트워크로 나뉘고, 서로 연결하여 거대한 네트워크 환경을 이루고 있다. 여기에서 서브넷(subnet)은 부분 네트워크를 의미하며, 다양한 서브넷이 연결되어 거대한 네트워크 환경을 이루고 있다고 이해하면 된다. 그렇다면 부분 네트워크인 서브넷은 어떻게 구분하고 식별할 수 있을까? 이때 사용하는 것이 서브넷 마스크(subnet mask)이다.
서브넷 마스크는 IP 주소와 동일한 32비트 구조에 네트워크 ID와 호스트 ID로 구성되어 있다. 여기서 네트워크 ID는 서브넷을 구분하는 기준 값이고, 호스트 ID는 동일 서브넷 내에서 대상을 구분하는 기준 값이다.

라우팅과 라우터

라우팅(routing)은 네트워킹 통신을 수행할 때 목적지 경로를 선택하는 작업을 의미하며, 이를 수행하는 장비를 라우터(router)라고 한다. 이런 라우터는 라우팅 테이블이라는 서브넷의 경로 리스트를 가지고 목적지 네트워크에 대한 최적 경로를 선택해서 전달하는 역할을 한다.

TCP와 UDP

OSI 7계층의 전송 계층에서 사용하는 프로토콜에는 대표적으로 TCP(Transmission Control Protocol)와 UDP(User Diagram Protocol)가 있으며, 데이터 전송을 담당한다.
TCP는 송수신 대상 간 연결을 맺고 데이터 전송 여부를 하나씩 확인하며 전송하는 연결형 프로토콜로, 신뢰성 있는 데이터 전송을 보장한다. UDP는 TCP와 반대로 송수신 간 연결 없이 전달하는 비연결형 프로토콜이다.
TCP와 UDP를 사용하는 응용 서비스는 서로 구분할 수 있도록 포트 번호를 사용한다.

3.2. AWS 네트워킹 소개

AWS 네트워킹 서비스는 AWS 글로벌 인프라에서 생성된 다양한 자원의 워크로드를 수행하는 네트워킹 서비스이다. AWS 네트워킹 서비스를 이용하여 최적의 안정성과 보안성, 성능을 보장받는 애플리케이션을 실행할 수 있다.

3.2.1. AWS 리전 네트워크 디자인

리전은 전 세계 주요 도시의 데이터 센터를 군집화(clustering)하는 물리적 위치를 의미한다. AWS 리전 내부에는 트랜짓 센터(transit center)와 가용 영역이 서로 연결되어 네트워크 환경을 이루고 있으며, 네트워킹 측면으로 어떤 대상과 연결되었는지에 따라 다음과 같이 분류할 수 있다.

Intra-AZ 연결

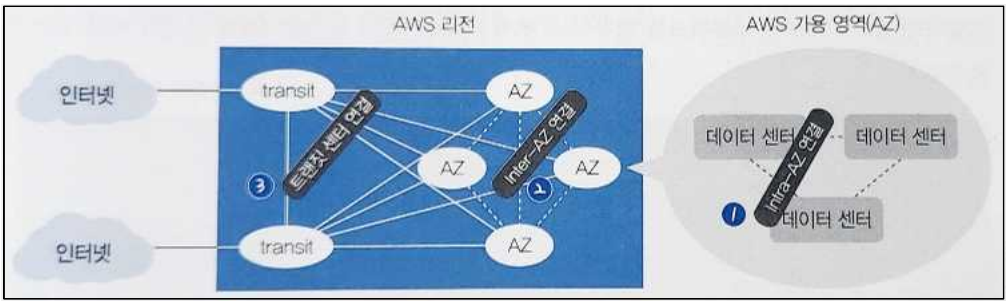
리전 내부에는 논리적인 데이터 센터의 집합인 가용 영역이 여럿 존재한다. 이런 데이터 센터 간 연결을 Intra-AZ 연결이라고 한다.

Inter-AZ 연결

리전 내부에 위치하는 가용 영역은 실제 100km 이내로 떨어져 있다. 지리적으로 떨어져 있는 가용 영역끼리 연결되어 네트워킹 환경을 구성하고 있으며, 이런 가용 영역 간 연결을 Inter-AZ 연결이라고 한다.

트랜짓 센터 연결

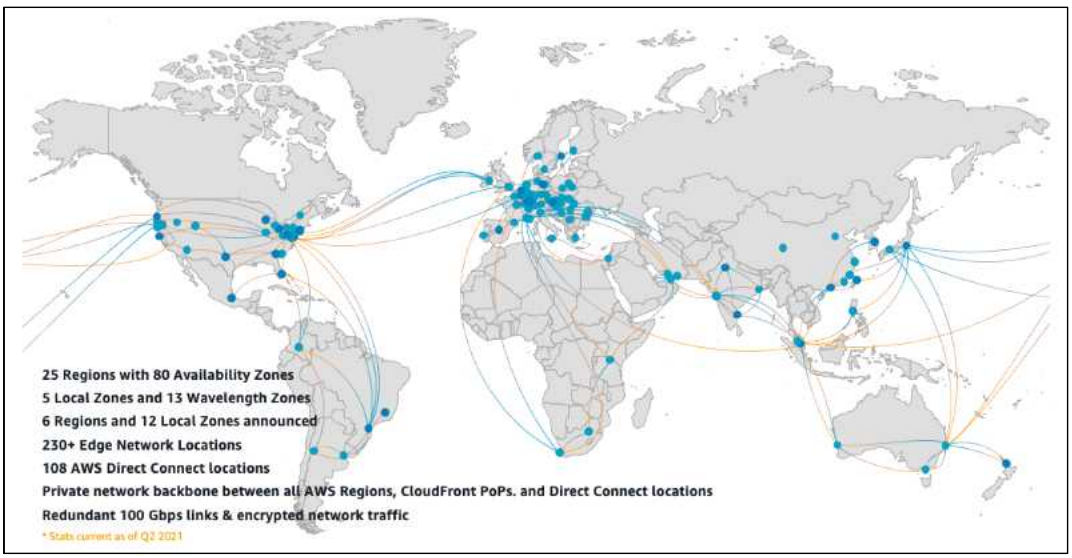
리전에서 외부 인터넷 구간과 통신이 필요할 때는 트랜짓 센터를 통해 통신한다. 내부에 있는 가용 영역들은 외부 인터넷 통신을 위해 트랜짓 센터와 연결되어 네트워킹 환경을 구성하며, 이런 가용 영역 간 연결을 트랜짓 센터 연결(transit center connection)이라고 한다.



3.2.2. AWS 글로벌 네트워크와 엣지 POP

엣지 POP(edge Point Of Presence)는 AWS 글로벌 네트워크라는 전용망을 활용하여 안정적으로 고성능 서비스를 제공하는 센터이다. 엣지 POP을 통해 사용자에게 글로벌 서비스 콘텐츠를 빠르게 제공할 수 있다.

엣지 POP은 엣지 로케이션(edge location)과 리전별 엣지 캐시(regional edge cache)로 구성되며, 전 세계 300개 이상의 엣지 로케이션과 13개의 리전별 엣지 캐시가 서로 연결된 AWS 글로벌 네트워크를 구성하고 있다.



3.2.3. AWS 네트워킹 서비스 소개

AWS의 다양한 자원이 서로 원활하게 통신할 수 있도록 도와주는 것이 AWS 네트워킹 서비스이다.

- VPC: 사용자 전용 가상의 프라이빗 클라우드 네트워크로, 네트워크 자원을 탄력적으로 활용하는 서비스를 제공한다.
- Transit Gateway: 중앙 허브 개념처럼 VPC와 온프레미스 네트워크를 연결하는 게이트웨이 역할의 서비스를 제공한다.
- Route 53: AWS에서 제공하는 관리형 DNS 서비스로 도메인 등록, 라우팅, 상태 확인 등 서비스를 제공한다.
- Global Accelerator: AWS 글로벌 네트워크를 통해 애플리케이션을 빠르고 안정적으로 사용할 수 있도록 가용성 및 성능을 보장하는 서비스를 제공한다.
- Direct Connect: 온프레미스 환경에서 AWS와 전용 네트워크 연결 서비스를 제공한다.
- Site-to-Site VPN: IPsec VPN 연결을 생성하여 암호화된 네트워크를 구성하는 서비스를 제공한다.

3.3. Amazon VPC 소개

Amazon VPC(Virtual Private Cloud)는 사용자 정의로 구성된 가상의 프라이빗 클라우드 네트워크 이다. 사용자는 Amazon VPC에서 제공하는 다양한 네트워킹 요소를 이용하여 가상의 클라우드 네트워크를 구성할 수 있다.

3.3.1. Amazon VPC 기본 구성 요소

리전과 VPC

Amazon VPC는 리전마다 독립적으로 구성되어 있다. 예를 들어 서울 리전에 VPC를 생성했다면 생성한 VPC는 서울 리전에만 있으며, 다른 리전에는 없다. 또한 리전 내에는 다수의 VPC를 생성할 수 있으며, 각 VPC는 서로 독립적으로 분리된다.

서브넷과 가용 영역

Amazon VPC라는 하나의 독립된 클라우드 네트워크에도 서브넷을 이용하여 분리된 네트워크로 구성할 수 있다. 서브넷은 VPC 내 별도로 나누어진 네트워크라고 생각하면 된다. 그리고 서브넷은 반드시 하나의 가용 영역에 종속적으로 위치한다.

서브넷은 VPC 네트워크 환경 구성에 따라 퍼블릭 서브넷과 프라이빗 서브넷으로 분류할 수 있다. 퍼블릭 서브넷은 인터넷 구간과 연결되어 있어 외부 인터넷 통신이 가능한 네트워크 영역이고, 프라이빗 서브넷은 인터넷 구간과 연결되지 않은 폐쇄적인 네트워크 영역이다.

IP CIDR(Classless Inter-Domain Routing)

IP CIDR은 네트워크에 할당할 수 있는 IP 주소 범위를 표현하는 방법이다. VPC라는 큰 네트워크의 IP CIDR에서 서브넷이라는 작은 네트워크의 IP CIDR이 분할되어 있다. 결론적으로 서브넷에 생성되는 자원은 IP CIDR 범위 안에 있는 IP 주소를 할당받을 수 있다.

10.0.0.0/16을 VPC의 CIDR로 설정했다고 가정하면 “/”뒤의 16이라는 숫자는 총 32비트로 구성되는 IP에서 앞에 16비트는 고정한다는 의미를 갖고 있다. 즉, 뒤에 16비트로 다양한 IP를 구성할 수 있고 결과적으로 65536개(2의 16승)의 IP를 구성할 수 있다. => 이 VPC는 65536개의 Private IP를 가질 수 있다.

가상 라우팅과 라우팅 테이블

Amazon VPC를 생성하면 기본적으로 네트워크 경로를 확인하여 트래픽을 전할하는 목적의 가상 라우터가 생성된다. 이렇게 생성된 가상 라우터는 기본 라우팅 테이블을 보유하고 있으며, 라우팅 테이블을 통해 네트워크 경로를 식별할 수 있다. 물론 기본 라우팅 테이블 외에 별도의 라우팅 테이블을 생성할 수 있고, 생성된 라우팅 테이블은 서브넷과 연결(attach)하여 서브넷마다 라우팅 테이블을 가질 수 있다.

보안 그룹과 네트워크 ACL

Amazon VPC는 보안 그룹(security group)과 네트워크 ACL(Access Control List) 같은 가상의 방화벽(firewall) 기능을 제공하여 서브넷과 생성된 자원에 대한 트래픽을 보호한다. 트래픽 접근을 통제하는 것이 주된 목적이며, IP CIDR 블록, 프로토콜, 포트 번호 등을 정의하여 허용(allow)과 거부(deny)를 결정하는 보안 규칙을 만든다.

보안 규칙에는 방향성이 있고 가상 방화벽을 기준으로 들어오는 인바운드(inbound) 규칙과 빠져나가는 아웃바운드(outbound) 규칙이 있기 때문에 트래픽 흐름을 고려한 보안 규칙을 세워야 한다.

트래픽 접근 제어 대상
보안 그룹과 네트워크 ACL의 가장 큰 차이점은 접근 제어 대상이 서로 다르다는 것이다. 보안 그룹은 인스턴스와

같은 자원 접근을 제어하며, 네트워크 ACL은 서브넷 접근을 제어한다.

스테이트풀과 스테이트리스
보안 그룹은 이전 상태 정보를 기억하고 다음에 그 상태를 활용하는 스테이트풀(stateful) 접근 통제를 수행하며, 네트워크 ACL은 이전 상태 정보를 기억하지 않아 다음에 그 상태를 활용하지 않는 스테이트리스(stateless) 접근 통제를 수행한다.

허용 및 거부 정책
보안 그룹의 정책 테이블은 허용 규칙만 나열하며 허용 규칙에 해당하지 않으면 자동 거부된다. 네트워크 ACL의 정책 테이블은 허용 규칙과 거부 규칙이 모두 존재하여 규칙을 순차적으로 확인하고 허용과 거부를 판단한다.

보안 그룹 => 자원 접근 제어 => 스테이트풀 접근 통제 => 허용 규칙만 나열
네트워크 ACL => 서브넷 접근 제어 => 스테이트리스 접근 통제 => 허용 규칙과 거부 규칙이 모두 존재

3.3.2. Amazon VPC와 다른 네트워크 연결

인터넷 게이트웨이

Amazon VPC는 프라이빗 클라우드 네트워크 환경으로, 외부 인터넷 구간과 연결되지 않은 독립적인 네트워크이다. 외부 인터넷 구간과 연결이 필요하면 게이트웨이라는 네트워킹 자원을 생성한 후 Amazon VPC와 연결하여 외부 인터넷과 통신한다. 여기에서 인터넷 게이트웨이는 VPC와 인터넷 구간의 논리적인 연결이자 인터넷으로 나가는 관문이 되는 네트워킹 자원이다.

NAT 게이트웨이

NAT 게이트웨이(Network Address Translation gateway)는 프라이빗 서브넷에서 외부 인터넷으로 통신하는 관문 역할을 한다. NAT는 IP 주소를 변환하는 기능을 제공하며, 프라이빗 IP 주소를 퍼블릭 IP 주소로 변환하여 외부 인터넷 구간 통신 환경을 만든다.

VPC 피어링

VPC 피어링은 서로 다른 VPC를 연결하는 기능이다. 동일 리전뿐만 아니라 다른 리전에 위치한 VPC와 연결할 수도 있고, 다른 계정에 위치한 VPC까지 연결할 수 있다.

전송 게이트웨이

전송 게이트웨이(transit gateway)는 다수의 VPC나 온프레미스를 단일 지점으로 연결하는 중앙 집중형 라우터이다. 단일 지점 연결이라 네트워크 구성이 간소화되고 비용도 절감되는 효과가 있다.

가상 프라이빗 게이트웨이

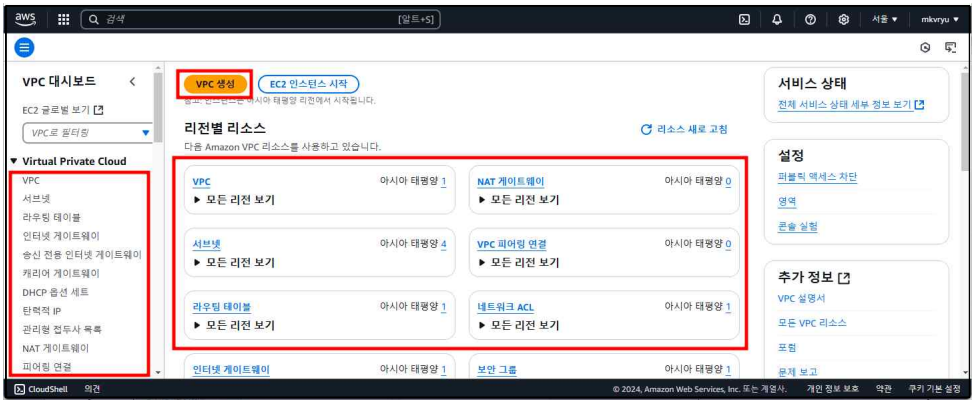
가상 프라이빗 게이트웨이(virtual private gateway)는 관리형 AWS Site-to-Site VPN을 연결하거나 AWS Direct Connect로 온프레미스 환경을 연결한다.

3.4. Amazon VPC로 퍼블릭 및 프라이빗 서브넷 구성하기

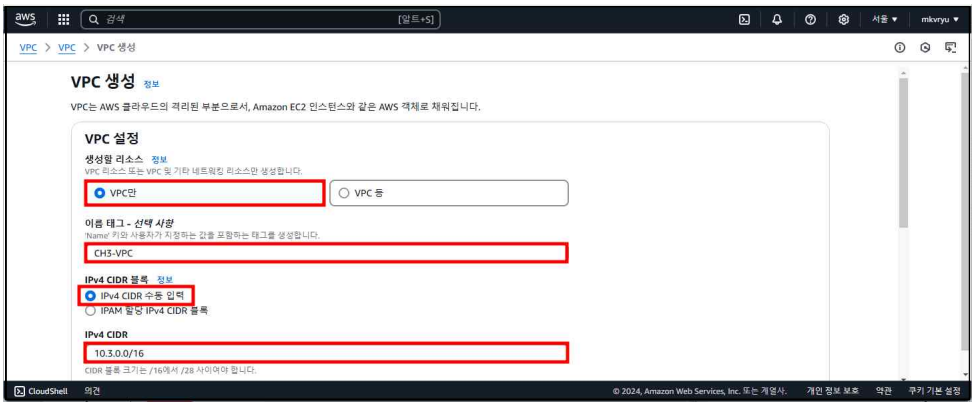
3.4.1. 사용자 VPC 생성하기

Amazon VPC는 생성 주체에 따라 기본 VPC(default VPC)와 사용자 VPC(custom VPC)로 구분한다.
기본 VPC는 리전마다 한 개씩 가지고 있으며, AWS 네트워킹 리소스를 미리 정해서 생성되어 있다. 반면 사용자 VPC는 사용자에 따라 수동으로 정의하는 VPC로 리전마다 최대 다섯 개까지 생성할 수 있다.

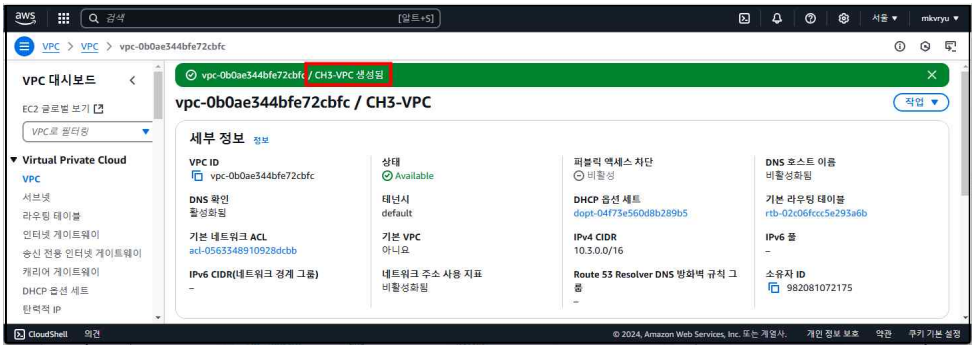
“AWS에 로그인” => “서비스” => “네트워킹 및 콘텐츠 전송” => “VPC” 선택
VPC 대시보드가 나타나고 VPC 대시보드를 살펴보면, 왼쪽에는 VPC 관련 메뉴가 나열되어 있고 가운데에는 리전마다 생성된 VPC 리소스를 확인할 수 있다. “VPC 생성”을 누른다.



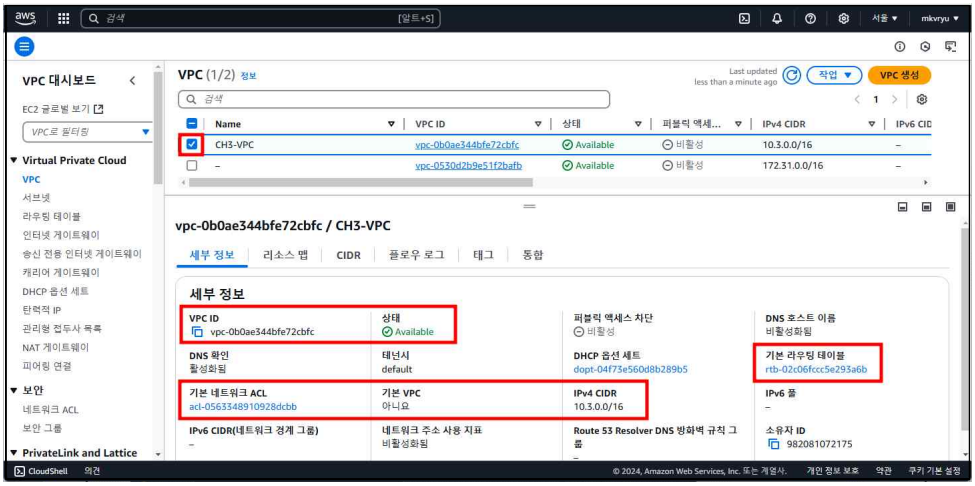
VPC 생성 페이지에서 다음과 같이 설정하고 아래쪽에 있는 VPC 생성을 누른다.
· 생성할 리소스 => “VPC만”
· 이름 태그 => “CH3-VPC”
· IPv4 CIDR 블록 => “IPv4 CIDR 수동 입력”
· IPv4 CIDR => “10.3.0.0/16”



가장 위쪽에 VPC가 생성되었다는 메시지가 보이고 설정한 세부 정보를 출력한다. 왼쪽 메뉴에서 “VPC”를 선택하면 현재 리전에 존재하는 VPC 정보를 확인할 수 있다.



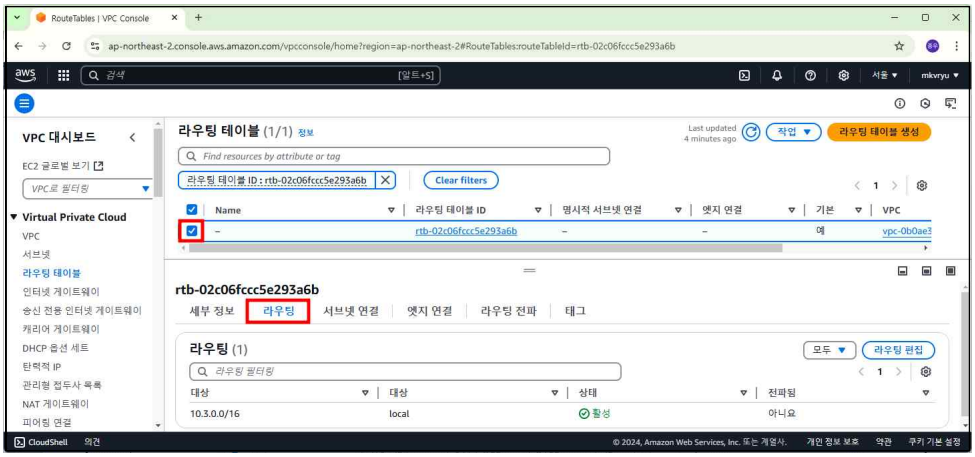
VPC 리스트를 보면 기본 VPC와 새로 생성한 사용자 VPC 두 개가 있다. 이번에 생성한 “CH3-VPC”에 체크한다. VPC의 세부 정보가 모두 표시되지 않으면 VPC 새로 고침을 누른다.



VPC의 세부 정보에서 몇 가지 사항을 확인해 보면 다음과 같다.

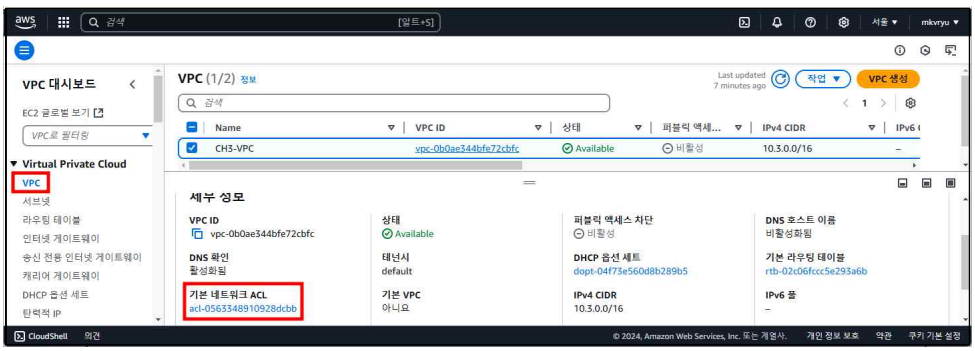
- VPC ID: VPC마다 고유한 ID로 자동 생성된다. 길고 복잡한 구조라서 VPC 식별은 이름 태그를 활용한다.
- 상태: 현재 VPC는 Available로 사용할 수 있는 상태이다.
- 기본 라우팅 테이블: VPC에서 사용하는 기본 라우팅 테이블이다.
- 기본 네트워크 ACL: VPC 네트워크에 대해 접근 통제를 수행하는 보안 정책이다.
- 기본 VPC: 기본 VPC가 아닌 사용자 VPC이다.
- IPv4 CIDR: VPC에서 사용하는 IP 대역이다.

기본 라우팅 테이블을 확인하기 위해 “기본 라우팅 테이블의 링크”를 클릭한다. 표시된 라우팅 테이블에 체크하고 아래쪽의 “라우팅 탭”을 클릭한다.

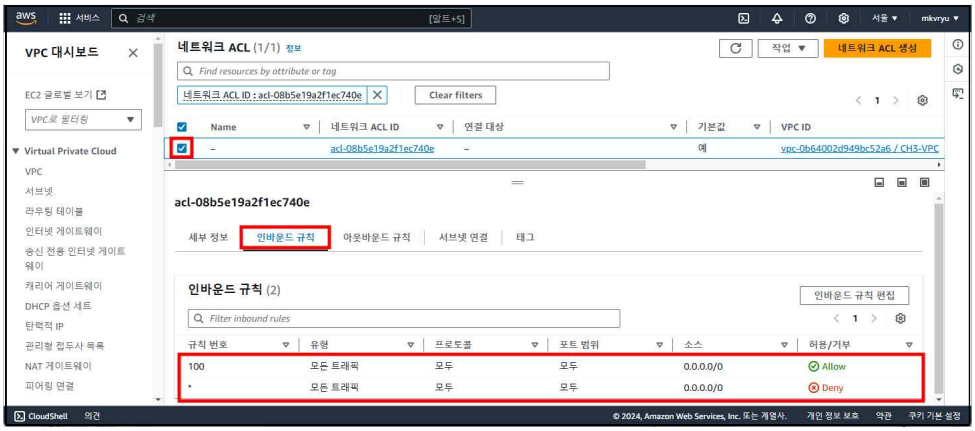


앞의 그림과 같이 자동으로 기본 라우팅 테이블의 생성되며, 테이블 정보를 확인해 보면 IP 대역 “10.3.0.0/16”에 대한 타깃 대상은 “local”이다. 여기에 라우팅 테이블이 존재하는 것으로 가상 라우터가 있다는 것을 알 수 있다.

다시 VPC 페이지로 이동한 후 네트워크 ACL을 확인하기 위해 기본 네트워크 ACL 링크를 클릭한다.



표시된 네트워크 ACL에 체크한 후 아래쪽 “인바운드 규칙” 탭을 클릭한다.



앞의 그림과 같이 자동으로 기본 네트워크 ACL이 생성되며, 정보를 확인해 보면 모든 트래픽 유형과 모든 IP 범위(0.0.0.0/0)를 허용하는 규칙이 앞에 있다.

네트워크 ACL은 여러 규칙 리스트를 순차적으로 확인하므로 첫 번째 규칙에 따라 모든 트래픽과 모든 IP를 허용한다. 물론 필요에 따라 원하는 트래픽만 허용하거나 거부하는 규칙을 편집할 수 있다. 아웃바운드 규칙 탭도 동일하므로 확인해 보자.

사용자 VPC 생성을 위해 태그 이름과 IPv4 CIDR만 설정했는데 다양한 네트워킹 자원이 자동을 생성되었다. 사용자 VPC를 생성하면 기본 라우팅 테이블을 사용하는 가상 라우터가 생성되고 로컬 통신이 가능하다. 그리고 기본 네트워크 ACL이 생성되는데, 모든 트래픽과 IP 대역을 허용하는 기본 규칙이 있어 제약 없이 통신이 가능하다.

3.4.2. 퍼블릭 서브넷 생성하기

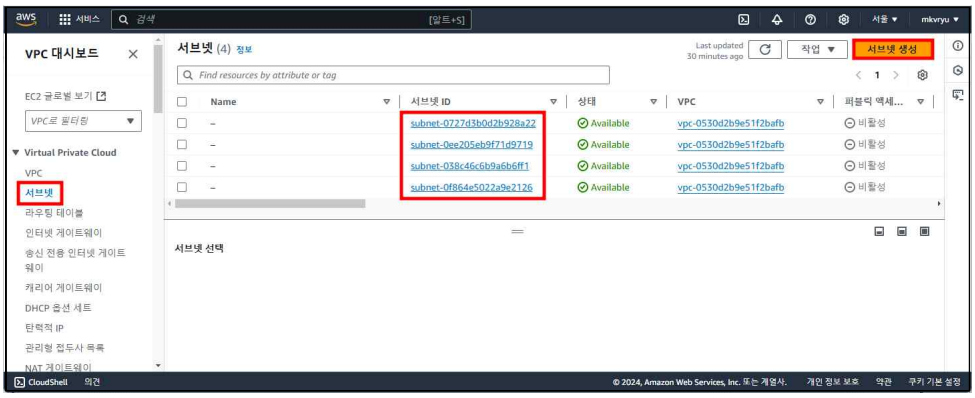
새로 생성한 사용자 VPC는 10.3.0.0/16 하나의 네트워크이다. 여기에 외부 인터넷과 자유롭게 통신할 수 있는 퍼블릭 서브넷을 생성한다.

서브넷 생성하기

서브넷은 분리된 네트워크로 VPC의 IP CIDR을 정의하며, 하나의 가용 영역에 종속되어 있다. 이런 서브넷을 생성할 때는 기본적으로 다음 사항을 정의해야 한다.

- VPC ID(연결산 대상의 IP)
- 서브넷 이름
- 가용 영역
- IPv4 CIDR 블록(VPC의 IPv4 CIDR에서 나누어진 대역)

VPC 메뉴에서 “서브넷”을 선택한다. 서브넷 페이지를 보면 현재 기본 VPC로 자동 생성된 서브넷이 4개 있다. 우리는 사용자 정의로 서브넷을 생성할 것이니 오른쪽 위의 “서브넷 생성”을 누른다.



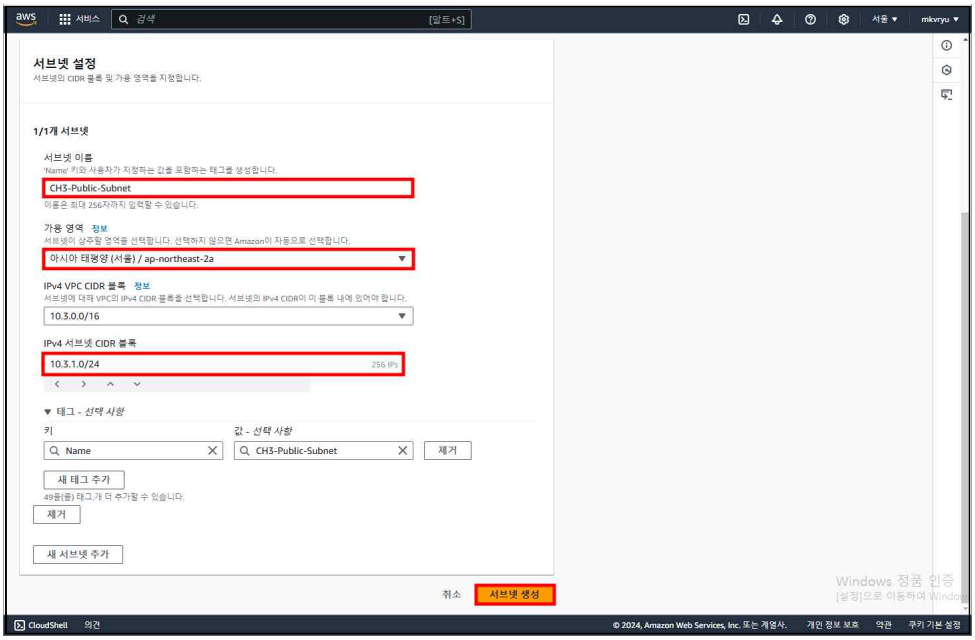
서브넷 생성 페이지에서 다음과 같이 설정하고 “서브넷 생성”을 누른다.

· VPC ID => “CH3-VPC”

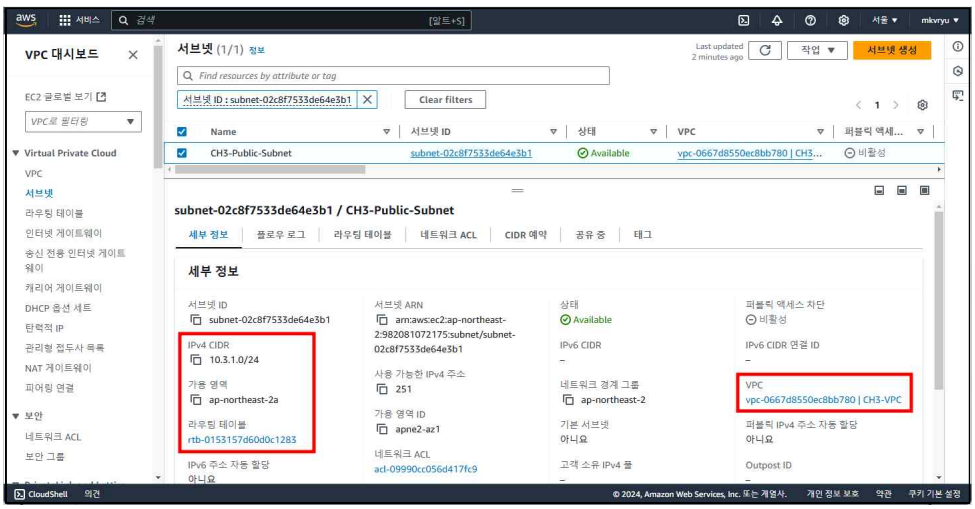
서브넷 이름 => “CH3-Public-Subnet”

가용 영역 => “아시아 태평양(서울) / ap-northeast-2a”

“IPv4 서브넷 CIDR 블록 => “10.3.1.0/24”



생성한 서브넷을 클릭하고 정보를 확인한다.



생성한 서브넷의 몇 가지 사항을 확인해 보면 다음과 같다.

· IPv4 CIDR: 서브넷에서 사용하는 IP 대역이다.

· 가용 영역: 서브넷이 위치한 가용 영역이다.

· VPC: 서브넷이 속한 VPC 정보이다.

· 라우팅 테이블: 서브넷이 사용하는 라우팅 테이블로 현재 기본 라우팅 테이블을 사용한다.

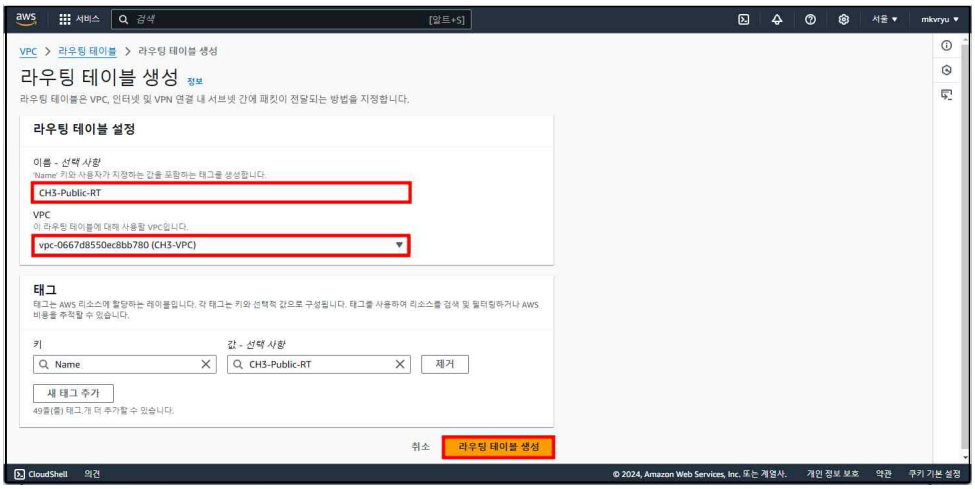
라우팅 테이블 생성하기

새로 생성한 서브넷은 기본 라우팅 테이블을 사용한다. 이번에는 서브넷마다 라우팅 테이블을 생성해서 연결한다. 이런 라우팅 테이블을 생성할 때는 기본적으로 다음 사항을 정의해야 한다.

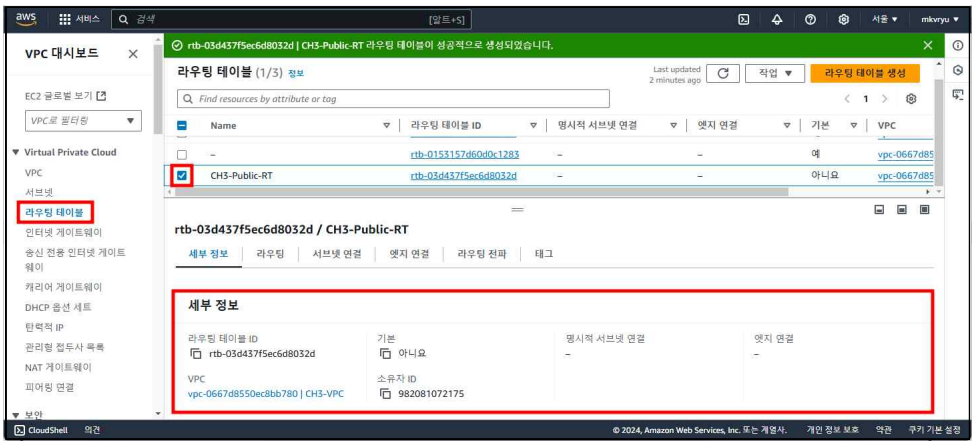
- 라우팅 테이블 이름
- VPC(연결한 대상의 VPC)
- 서브넷 연결

VPC 메뉴에서 “라우팅 테이블”을 선택한다. 라우팅 테이블 페이지가 표시되며 현재 기본 VPC의 기본 라우팅 테이블과 사용자 VPC의 기본 라우팅 테이블이 있다. 사용자 VPC의 사용자 정의로 라우팅 테이블을 생성할 것이니 오른쪽 위의 “라우팅 테이블 생성”을 누른다.

라우팅 테이블 생성 페이지에서 다음과 같이 설정하고 “라우팅 테이블 생성”을 누른다.
이름 => “CH3-Public-RT”
VPC => “CH3-VPC”

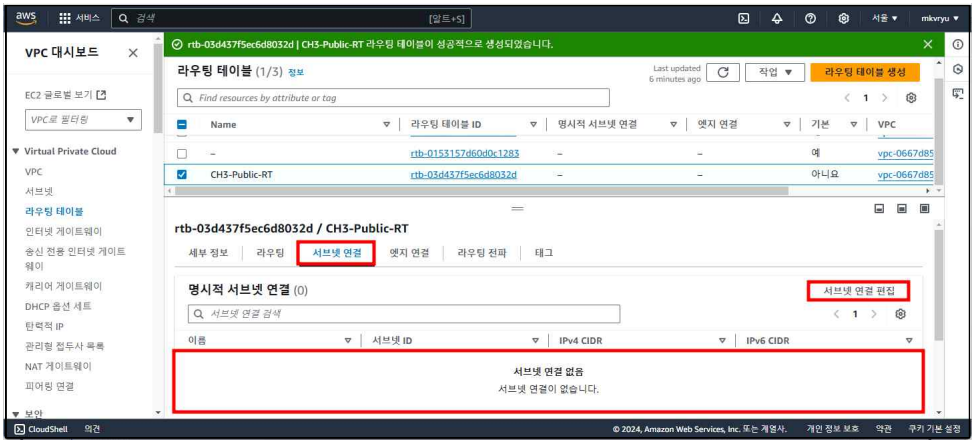


다시 라우팅 테이블 메뉴로 들어가서 새로 생성한 라우팅 테이블을 체크하면 세부 정보를 확인할 수 있다.

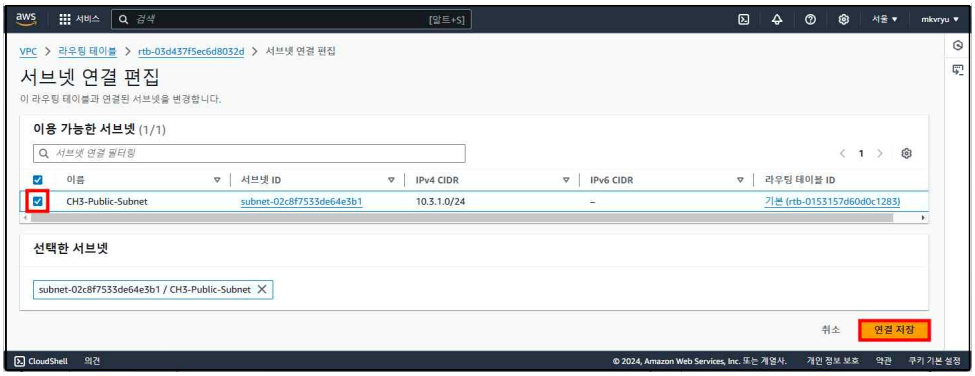


현재 라우팅 테이블은 기본 라우팅 테이블이 아닌 일반 라우팅 테이블이며, 서브넷과 연결되지 않고 덩그러니 있다. 생성된 라우팅 테이블을 서브넷과 연결하는 작업이 필요하다.

생성된 라우팅 테이블 아래쪽에서 “서브넷 연결 탭”을 클릭한다. 현재 연결된 서브넷이 없는 상태로 오른쪽에 있는 “서브넷 연결 편집”을 누른다.



라우팅 테이블에 새로 생성한 서브넷을 체크하고, 아래쪽에 연결 저장을 누른다.



라우팅 테이블 페이지에서 대상 라우팅 테이블 선택 후 “서브넷 연결 탭”에서 서브넷과 연결된 것을 볼 수 있다.

인터넷 게이트웨이 생성하기

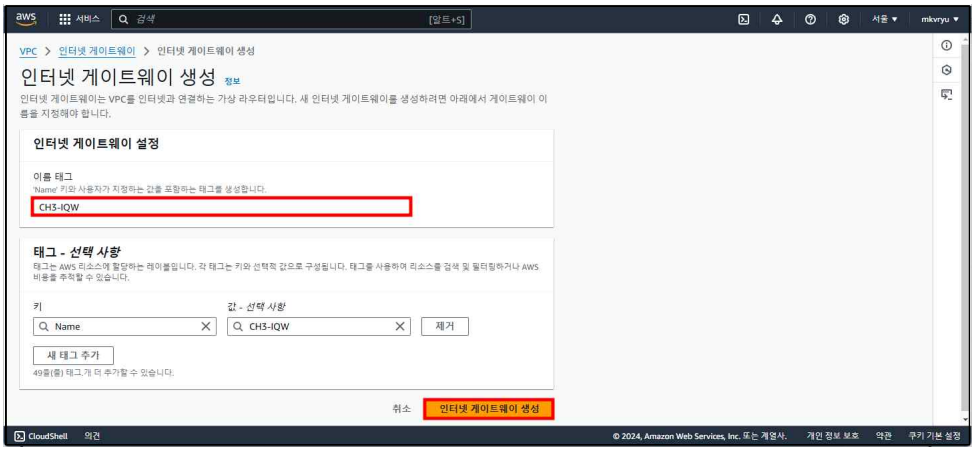
퍼블릭 서브넷은 외부 인터넷 구간과 자유롭게 사용할 수 있는 환경으로, 이 환경을 만들려면 인터넷 게이트웨이가 필요하다. 이런 인터넷 게이트웨이를 생성할 때는 기본적으로 다음 사항을 정의해야 한다.

- 이름 태그(인터넷 게이트웨이 이름)
- VPC 연결

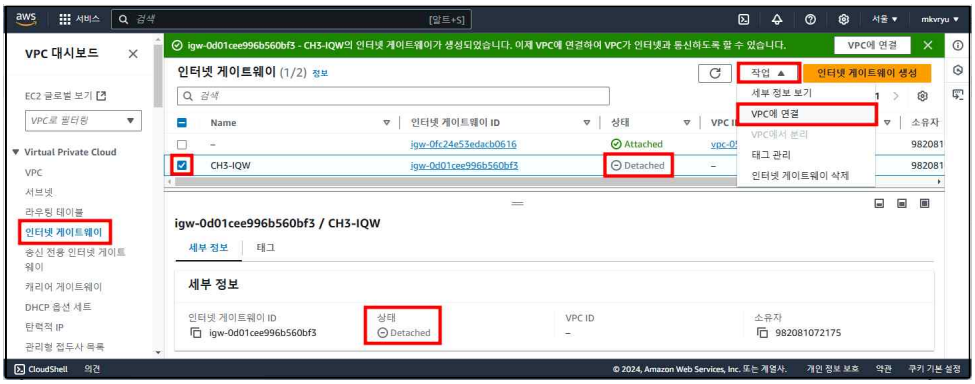
VPC 메뉴에서 “인터넷 게이트 웨이”를 선택한다. 인터넷 게이트웨이 페이지가 나타나며 현재 기본 VPC의 인터넷 게이트웨이가 존재한다. 신규 인터넷 게이트웨이를 생성하기 위해 위쪽의 “인터넷 게이트웨이 생성”을 누른다.

인터넷 게이트웨이 생성 페이지에서 다음과 같이 설정하고 “인터넷 게이트웨이 생성”을 누른다.

· 이름 태그 => “CH3-IGW”



인터넷 게이트웨이 페이지로 들어가면 신규 인터넷 게이트웨이가 생성된 것을 확인할 수 있다. 하지만 현재는 “Detached”(분리)로 뜨는데 VPC와 연결되지 않은 상태이다. VPC에 연결할 수 있도록 방금 생성한 인터넷 게이트웨이를 체크하고 위쪽의 “작업”을 클릭하여 “VPC에 연결”을 선택한다.



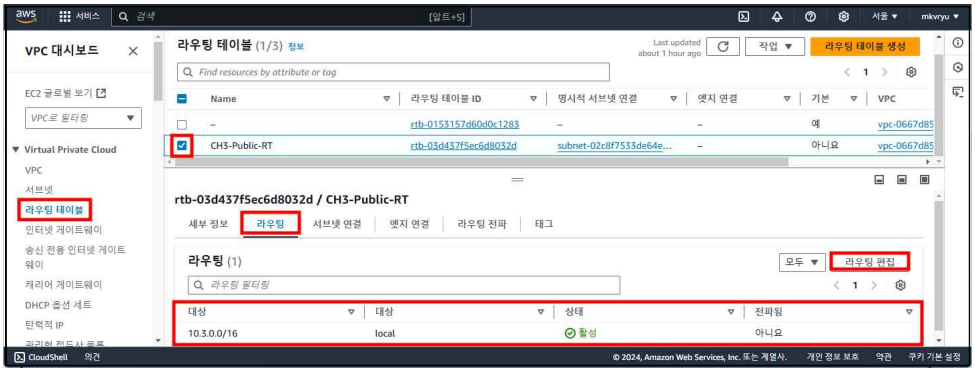
VPC에 연결 페이지가 표시되면 연결할 VPC를 선택하고 아래쪽의 “인터넷 게이트웨이 연결”을 누른다.

다시 인터넷 게이트웨이 페이지로 들어가면 상태 정보가 “Detached”에서 “Attached”(연결)로 변경된 것을 확인할 수 있다.

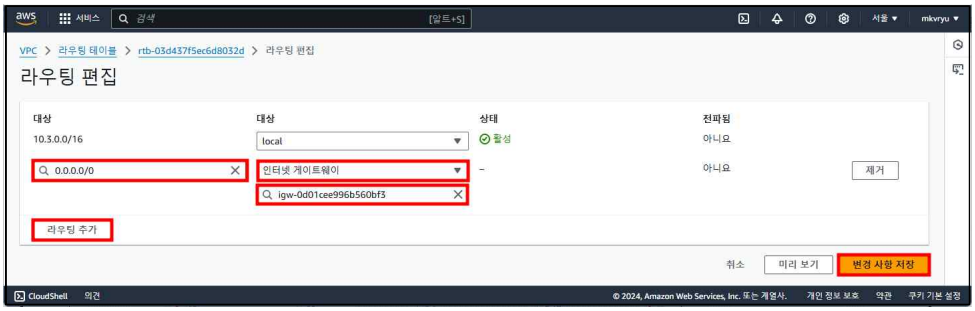
라우팅 테이블 편집하기

퍼블릭 서브넷 환경을 통해 인터넷 게이트웨이까지 생성했다. 하지만 퍼블릭 서브넷의 라우팅 테이블은 로컬 통신 경로만 있을 뿐 외부 인터넷 구간으로 갈 수 있는 경로 정보는 없다. 라우팅 테이블 수정 작업을 수행한다.

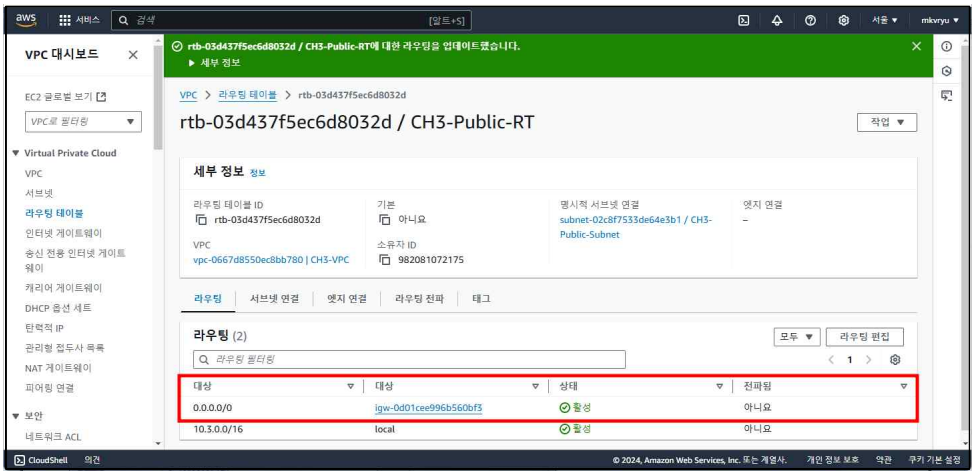
라우팅 테이블로 들어가서 생성한 라우팅 테이블을 체크한 후 “라우팅 탭”을 클릭한다. 라우팅 테이블 정보를 확인해 보면 “10.3.0.0/16” 대상에 대한 타깃 대상은 local로, 로컬 통신만 가능하다. 오른쪽에 있는 “라우팅 편집”을 누른다.



라우팅 편집 페이지에서 “라우팅 추가”를 눌러 다음과 같이 설정하고 “변경 사항 저장”을 누른다.
IP CIDR 대상 => “0.0.0.0/0”(모든 IP 대역)
타깃 대상 => “인터넷 게이트웨이”를 선택하고 생성한 인터넷 게이트웨이 ID를 선택



편집된 라우팅 정보를 확인한다.



0.0.0.0/0 모든 IP 대역에 대해 타깃 대상을 인터넷 게이트웨이로 설정했다. 여기까지 VPC를 생성하고 퍼블릭 서브넷 환경 구성을 완료했다. 퍼블릭 서브넷에 EC2 자원을 생성하고 통신을 해보자.

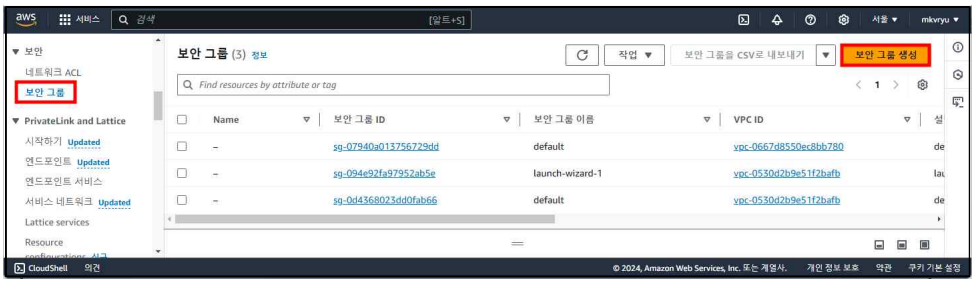
3.4.3. 퍼블릭 서브넷 통신 확인하기

퍼블릭 서브넷 환경을 구성 했으므로 퍼블릭 서브넷에 EC2 인스턴스를 생성하여 통신을 확인한다.

보안 그룹 생성하기

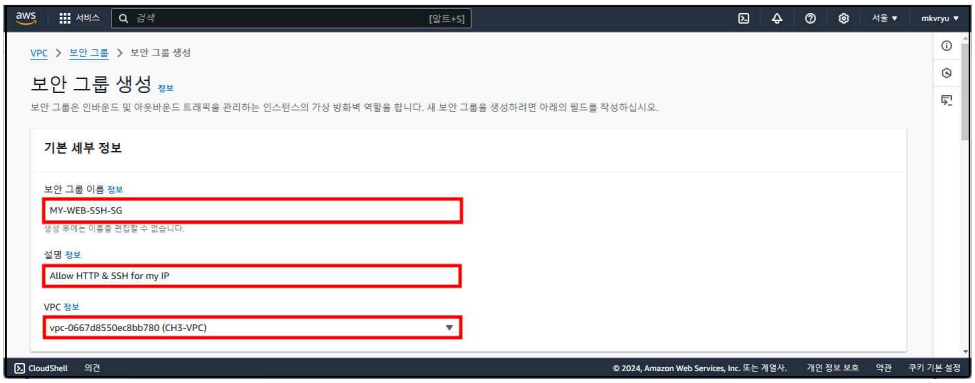
EC2 인스턴스는 기본적으로 한 개의 보안 그룹으로 접근을 통제한다. EC2 인스턴스를 설정할 때 별도의 보안 그룹을 지정하지 않으면 신규 보안 그룹이 자동으로 생성된다. 이번에는 수동으로 보안 그룹을 생성한다.

VPC 메뉴에서 “보안 그룹”을 선택한다. 새로운 보안 그룹을 만들기 위해 “보안 그룹 생성”을 누른다.



보안 그룹 생성 페이지에서 보안 그룹 이름과 VPC 정보를 다음과 같이 입력하고 “보안 그룹 생성”을 누른다.

- 보안 그룹 이름 => “MY-WEB-SSH-SG”
- 설명 => 에 영문이나 숫자로(예: Allow HTTP & SSH for my IP) 입력
- VPC => 새로 생성한 VPC(CH3-VPC)를 선택

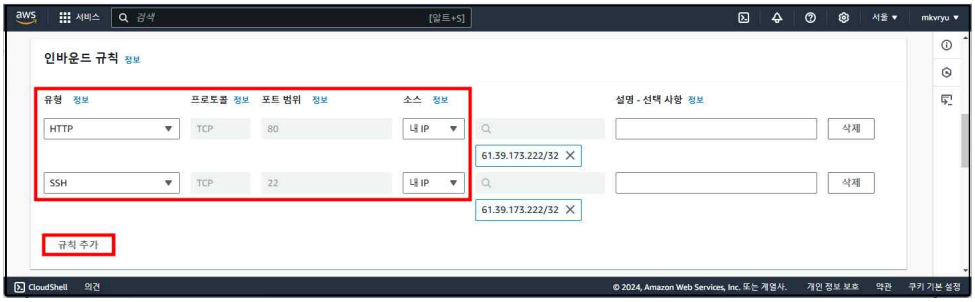


인바운드 규칙을 추가하기 위해 “규칙 추가”를 누른다.

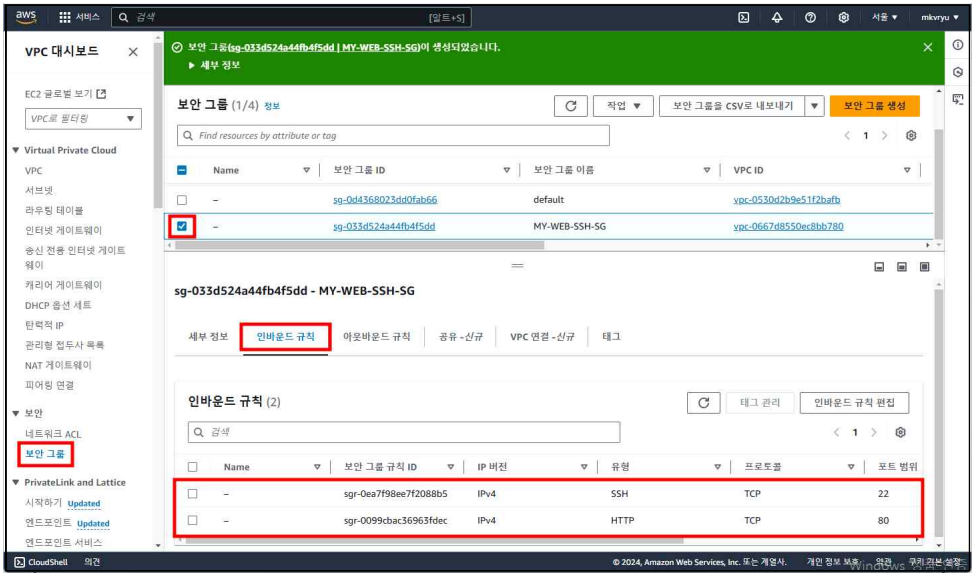
- 유형 => “HTTP”
- 소스 => “내 IP”

다시 “규칙 추가”를 누른다.

- 유형 => “SSH”
- 소스 => “내 IP”



다시 보안 그룹 페이지로 들어가서 생성한 보안 그룹을 체크하고 아래쪽에 있는 “인바운드 규칙 탭”을 클릭한다.



생성한 보안 그룹은 HTTP와 SSH 프로토콜에 대한 특정 IP 주소만 허용하는 보안 정책이다. 여기에서 특정 IP 주소는 현재 설정한 내 PC의 공인 IP 주소를 의미한다.

EC2 인스턴스 생성하기

서비스 => 컴퓨팅 => EC2 => 인스턴스 메뉴 => “인스턴스 시작”을 누르고 다음과 같이 설정한 후 “인스턴스 시작” 누르기

· 이름 및 태그 => “CH3-Public-EC2”

· 키 페어(로그인) => 기존에 생성한 키 페어 파일(ongja_key.pem) 선택

네트워크 설정에서 “편집” 누르기

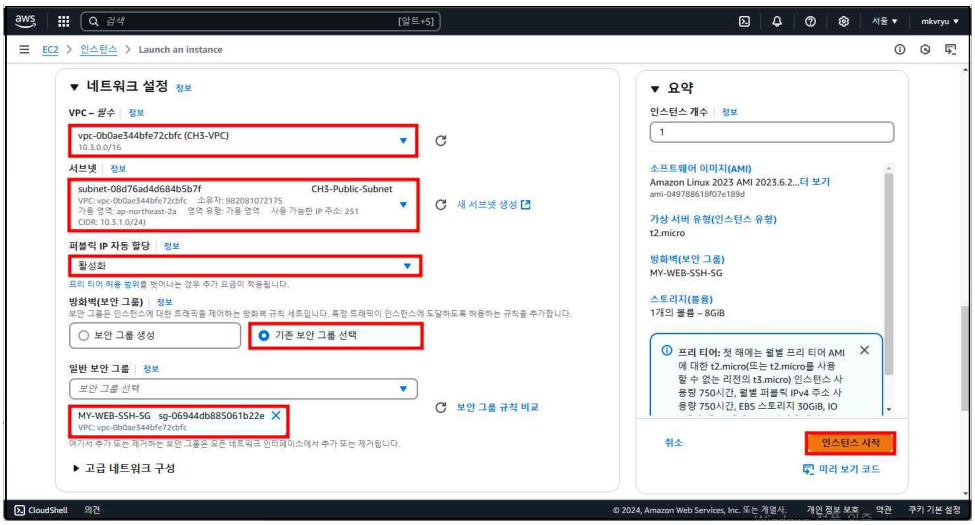
· VPC => “CH3-VPC”

· 서브넷 => “CH3-Public-Subnet”

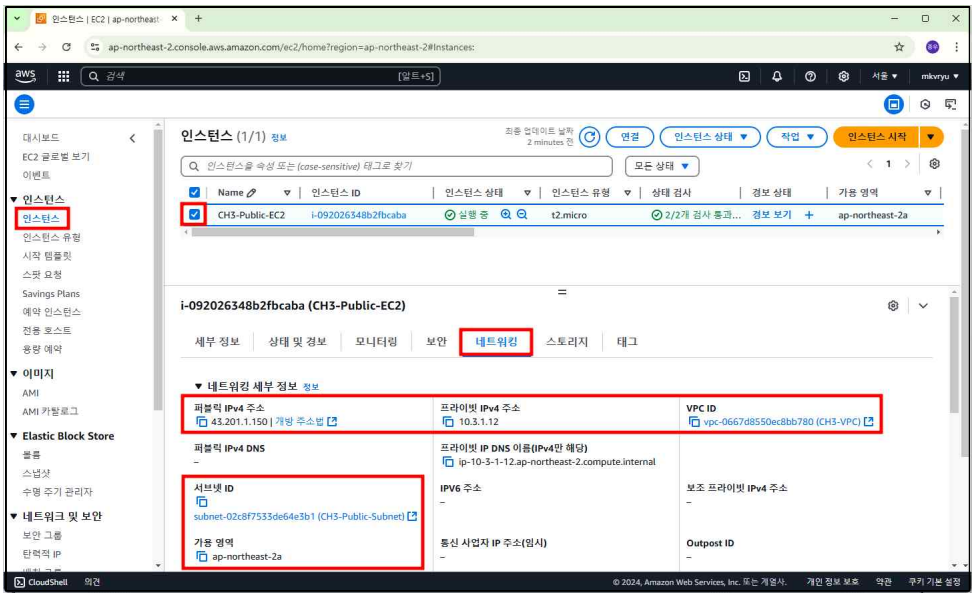
· 퍼블릭 IP 자동 할당 => “활성화”

· 방화벽(보안 그룹) => “기존 보안 그룹”

· 일반 보안 그룹 => “MY-WEB-SSH-SG”



"모든 인스턴스 보기" 누르기를 누르고 왼쪽 메뉴에서 "인스턴스"를 선택하면 생성된 EC2 인스턴스를 확인할 수 있다. 생성된 EC2 인스턴스에 체크하고 아래쪽에 있는 "네트워킹 탭"을 클릭한다.



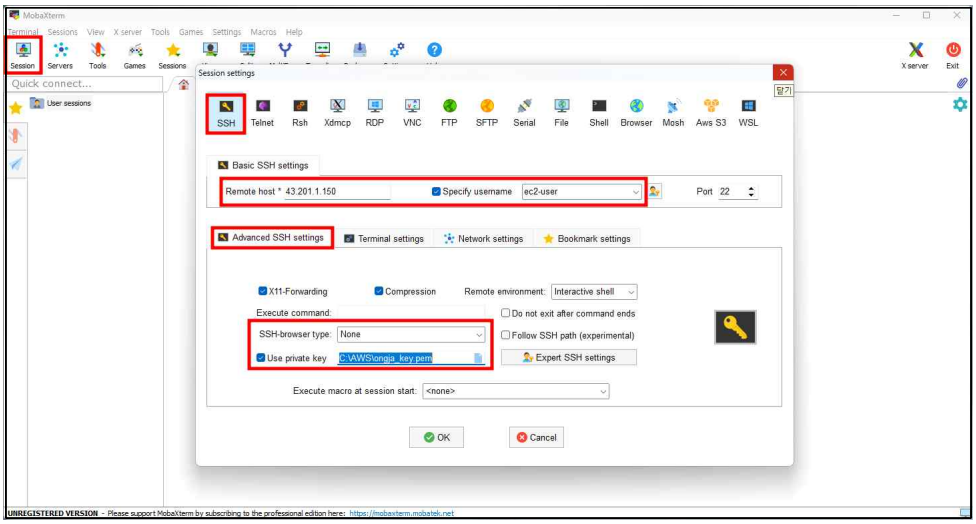
생성한 EC2 인스턴스의 네트워킹 세부 정보를 확인하면 다음과 같다.

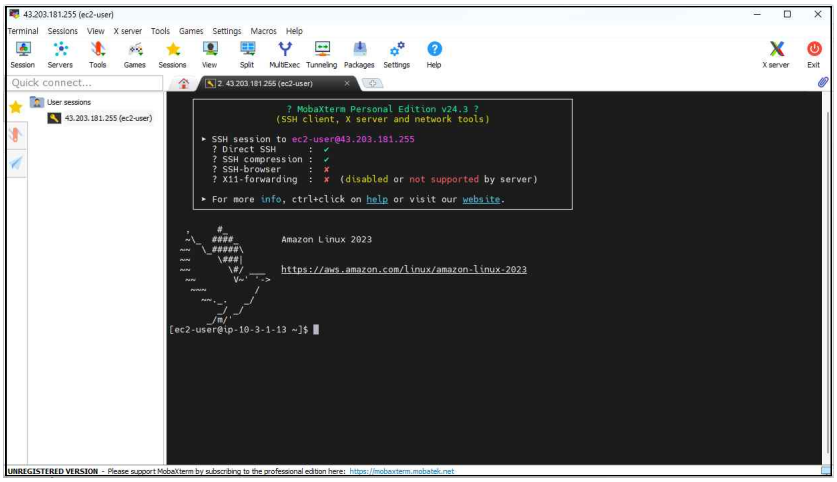
- 퍼블릭 IPv4 주소: EC2 인터페이스가 사용할 공인 IP 주소이다.
- 프라이빗 IPv4 주소: EC2 인스턴스가 내부 통신을 할 때 사용할 사설 IP 주소이다.
- VPC ID: 앞서 생성한 사용자 VPC 이다.
- 서브넷 ID: 퍼블릭 서브넷 용도로 생성한 서브넷에 위치한다.
- 가용 영역: 퍼블릭 서브넷에 설정한 가용 영역이다.

EC2 인스턴스 실습 환경 설정하기

생성된 EC2 인스턴스의 퍼블릭 IP 주소로 SSH 접근을 한다.

- 메인 메뉴에서 "Session"을 클릭하여 Session settings 창 열기
- Session setting 창에서 "SSH 탭"을 클릭하여 SSH 설정 정보 표시
- Remote host에 EC2 인스턴스 퍼블릭 IP 주소 붙여넣기
- "Specify unsename"에 체크한 후 "ec2-user" 입력
- 아래 "Advanced SSH settings"를 선택해서 상세 설정 정보 표시
- SSH-browser type은 "None"으로 선택
- "Use private key"에 체크한 후 내려받은 키 페어 파일 지정
- "OK" 누르기

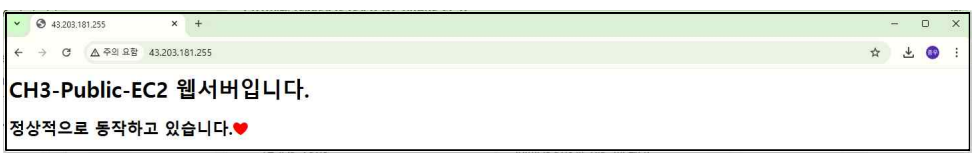




SSH 터미널에서 EC2 인스턴스에 웹 서비스를 설치한다.

```
# MyFirstEC2 인스턴스에 SSH 접속하기
# 슈퍼 유저로 변경
[ec2-user@ip-172-31-3-192 ~]$ sudo su -
# http 데몬 설치
[root@ip-172-31-3-192 ~]# yum install httpd -y
# http 데몬 실행
[root@ip-172-31-3-192 ~]# systemctl start httpd
# 웹 서비스 최초 페이지 내려받기
[root@ip-172-31-3-192 ~]# curl -L https://bit.ly/afbtest02 > /var/www/html/index.html
```

퍼블릭 서브넷의 통신을 확인하는 모든 설정이 완료되었다.



EC2 인스턴스에서 외부 인터넷으로 통신 확인하기

SSH 터미널에서 간단하게 외부 인터넷 통신을 확인한다.

```
# SSH 터미널 접속
```

```
# 외부 인터넷 구간으로 ping 테스트
[root@ip-10-3-1-12 ~]# ping google.com

PING google.com (172.217.161.206) 56(84) bytes of data.
64 bytes from kix07s03-in-f14.1e100.net (172.217.161.206): icmp_seq=1 ttl=48 time=33.5 ms
64 bytes from kix07s03-in-f14.1e100.net (172.217.161.206): icmp_seq=2 ttl=48 time=33.6 ms
64 bytes from kix07s03-in-f14.1e100.net (172.217.161.206): icmp_seq=3 ttl=48 time=33.4 ms
...
crtl + c

# 외부 인터넷 구간으로 HTTP 접근 테스트
[root@ip-10-3-1-12 ~]# curl google.com

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

퍼블릭 서브넷에 위치한 EC2 인스턴스는 외부 인터넷 구간의 대상과 정상적으로 통신한다.

외부 인터넷에서 EC2 인스턴스로 통신 확인하기

1. PC에서 웹 서비스에 접근 수행: 웹 서비스 접근 가능
2. PC에서 터미널 창을 열고 ping 테스트 수행: ping 통신 불가
3. 스마트폰에서 와이파이 연결을 해제하고 웹 서비스 접근: 웹 서비스 접근 불가
4. 스마트폰에서 와이파이를 연결하고 웹 서비스 접근: 웹 서비스 접근 가능

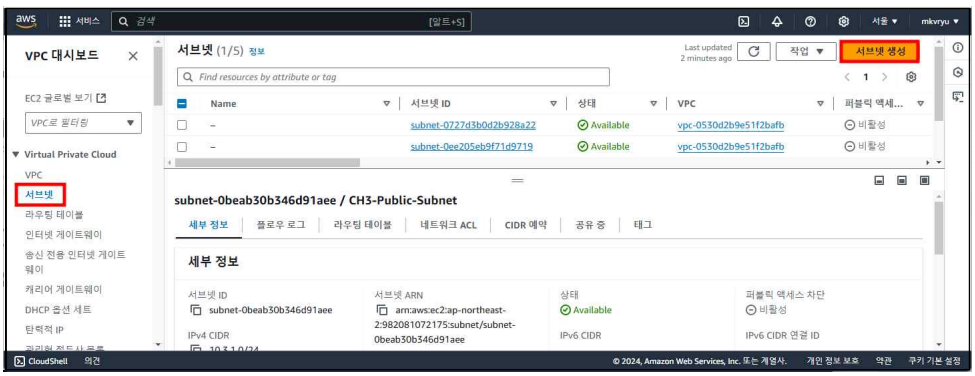
퍼블릭 서브넷 환경은 외부 인터넷 구간과 통신 제약이 없지만 왜 통신이 불가능한 경우가 발생할까? 그 이유는 보안 그룹 설정에 있다. 바로 보안 그룹 설정 때문에 HTTP와 SSH 프로토콜에 대해 내 IP(작업 중인 PC의 IP)만 허용한다. 스마트폰에서 집 무선 공유기에 와이파이를 연결하면 공유기의 NAT 기능을 이용하여 작업 중인 PC와 동일한 공인 IP 주소로 통신한다.

3.3.4. 프라이빗 서브넷 생성하기

프라이빗 서브넷 용도의 서브넷을 생성한다.

서브넷 생성하기

서비스 => 네트워킹 및 콘텐츠 전송 => VPC => 서브넷 => “서브넷 생성” 누르기



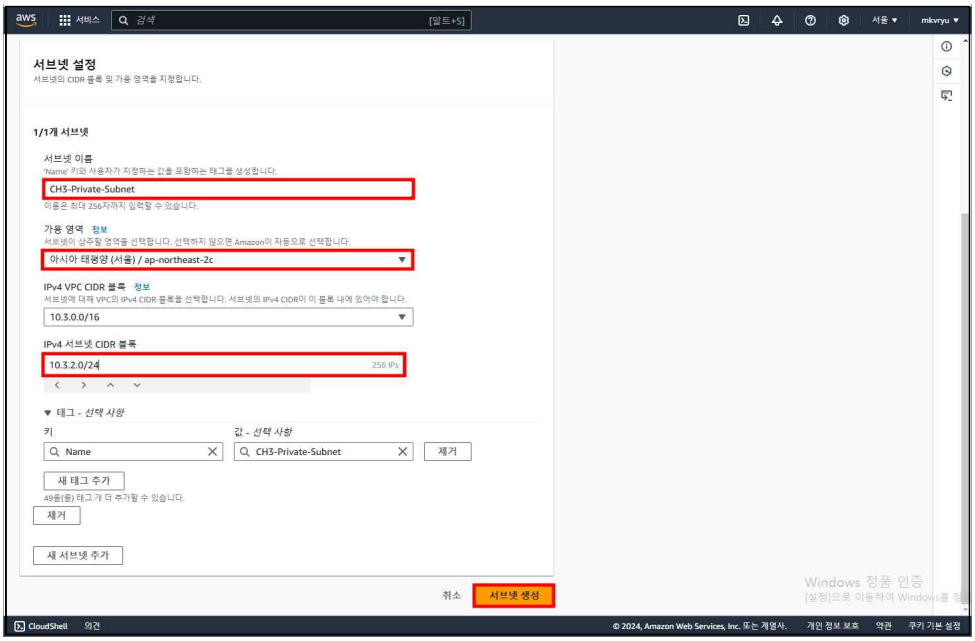
다음과 같이 입력하여 서브넷을 설정하고 “서브넷 생성”을 누른다.

· VPC ID => “CH3-VPC”

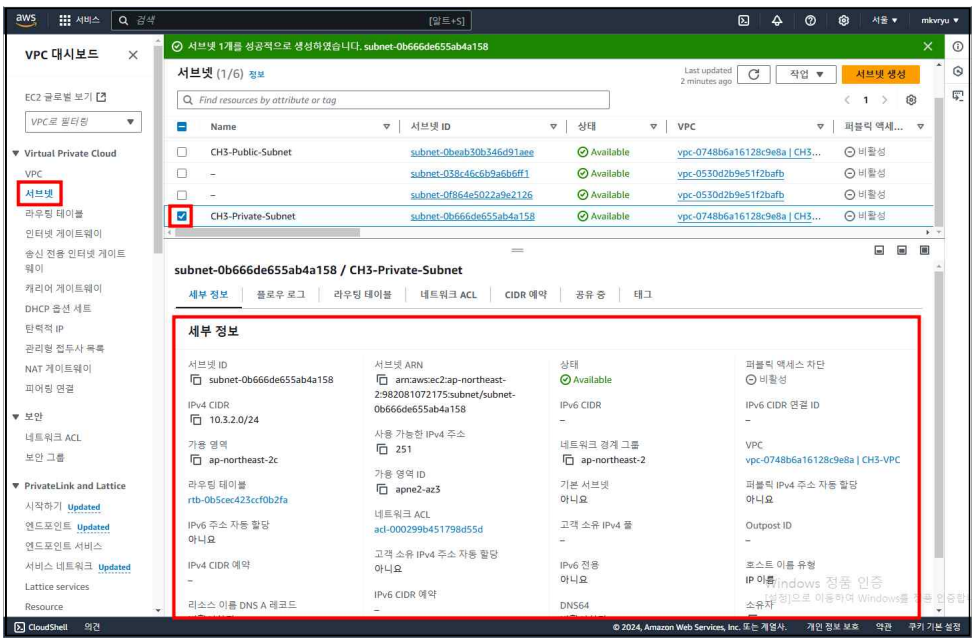
· 서브넷 이름 => “CH3-Private-Subnet”

· 가용 영역 => “아시아 태평양(서울) / ap-northeast-2c”

· IPv4 서브넷 CIDR 블록 => “10.3.2.0/24”

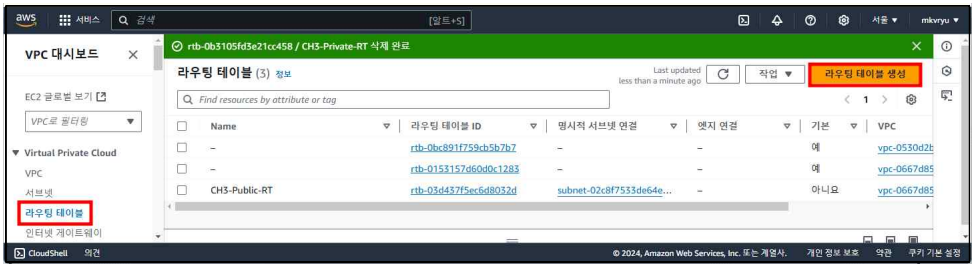


다시 “서브넷” 메뉴를 선택하여 생성한 서브넷을 체크한 후 정보를 확인한다.

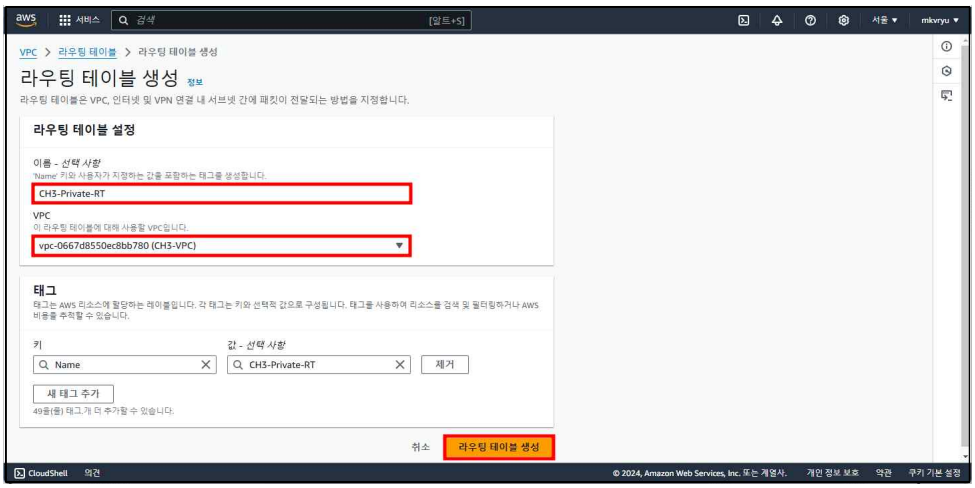


라우팅 테이블 생성하기

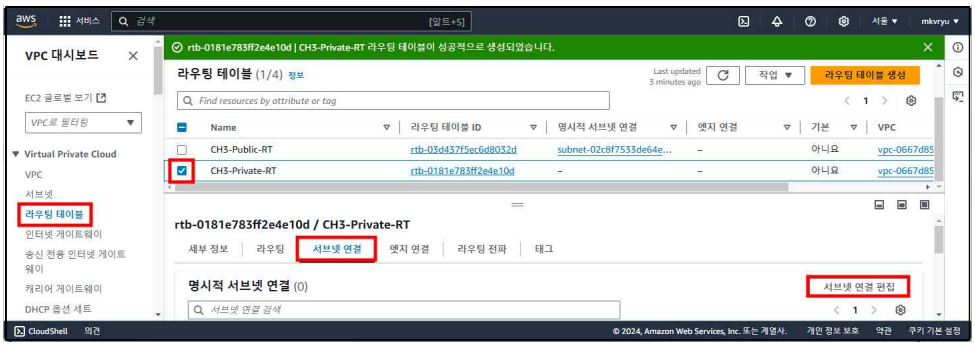
서비스 => 네트워킹 및 콘텐츠 전송 => VPC => 라우팅 테이블에서 “라우팅 테이블 생성”을 누른다.



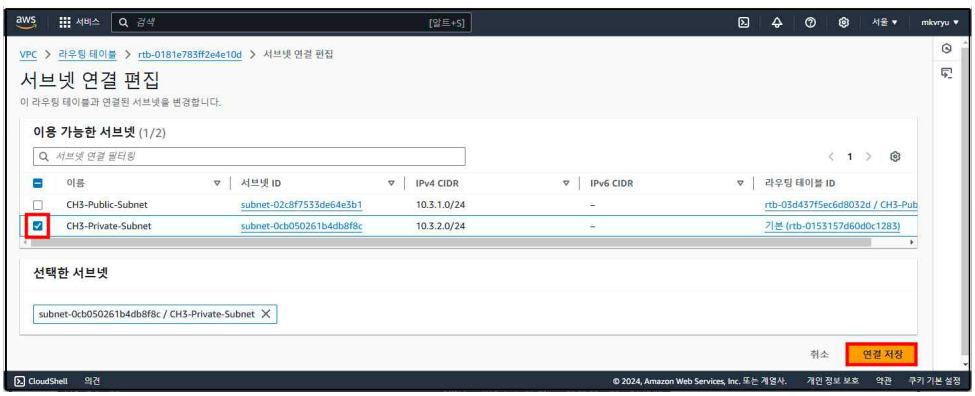
라우팅 테이블 생성에서 이름에 “CH3-Private-RT” 입력, VPC는 “CH3-VPC”로 선택하고 “라우팅 테이블 생성” 누르기



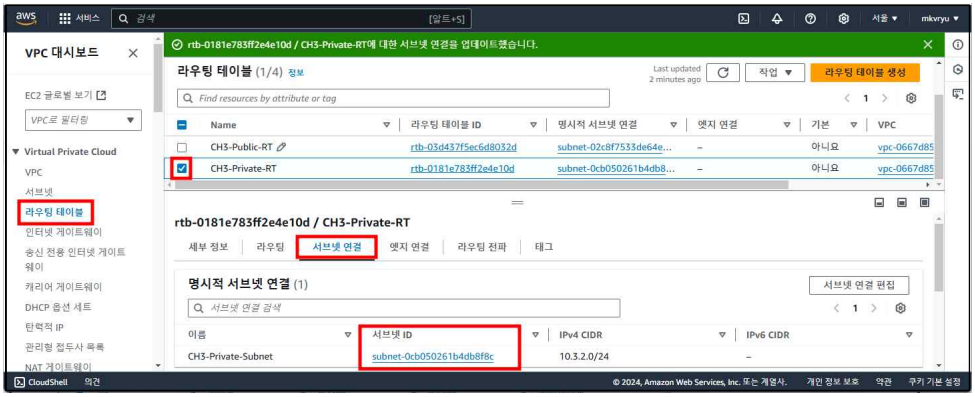
라우팅 테이블 메뉴에서 생성한 라우팅 테이블을 선택하고 “서브넷 연결 탭” 클릭 후 “서브넷 연결 편집” 누르기



“CH3-Private-Subnet”에 체크하고 “연결 저장” 누르기



라우팅 테이블에 들아가서 생성된 라우팅 테이블의 서브넷 연결 정보를 확인한다.



NAT 게이트웨이 생성하기

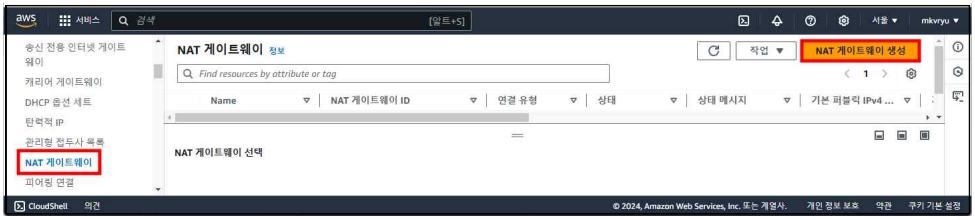
프라이빗 서브넷은 독립된 네트워크에서만 내부 통신하는 환경이지만, 외부 인터넷 통신을 위해 NAT 게이트웨이를 활용할 수 있다. 이런 NAT 게이트웨이를 생성할 때는 기본적으로 다음 사항을 정의해야 한다.

- 이름
- 서브넷
- 연결 유형(퍼블릭/프라이빗)
- 탄력적 IP 할당 ID

NAT 게이트웨이 연결 유형은 퍼블릭과 프라이빗으로 나누어진다. 퍼블릭은 NAT 게이트웨이를 통해 IP 주소를 변환하여 인터넷 구간과 통신하는 연결 유형이고, 프라이빗은 NAT 게이트웨이를 통해 IP 주소를 변환하여 다른 VPC나 온프레이미스 네트워크와 연결하는 유형이다.

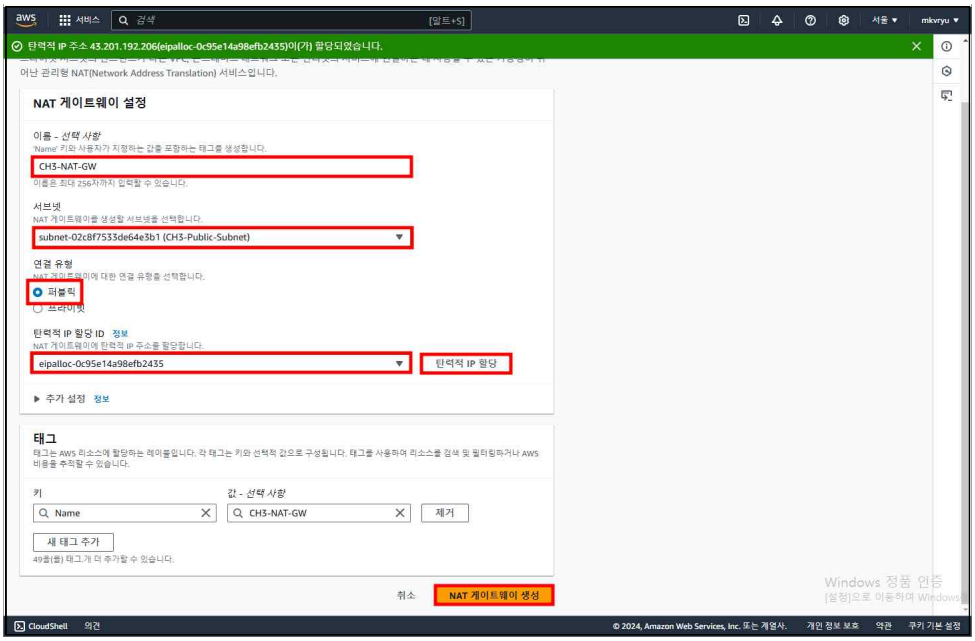
그리고 탄력적 IP 주소는 고정된 공인 IP 주소를 의미한다. 기본적으로 AWS에서 부여하는 퍼블릭 IP 주소는 유동적으로 주소를 관리한다. 예를 들어 EC2 인스턴스에 퍼블릭 IP 주소를 할당했다고 가정하면, EC2 인스턴스를 중지한 후 다시 시작할 때 할당받은 퍼블릭 IP 주소는 다른 주소로 변경된다. 이를 해결할 수 있는 주소가 탄력적 IP 주소로, 특정 이벤트가 있어도 할당받은 주소를 그대로 유지하는 특징이 있다.

왼쪽 VPC 메뉴에서 “NAT 게이트웨이”를 선택한다. NAT 게이트웨이 페이지가 나타나며, 새로 NAT 게이트웨이를 생성하기 위해 위쪽에 있는 “NAT 게이트웨이 생성”을 누른다.



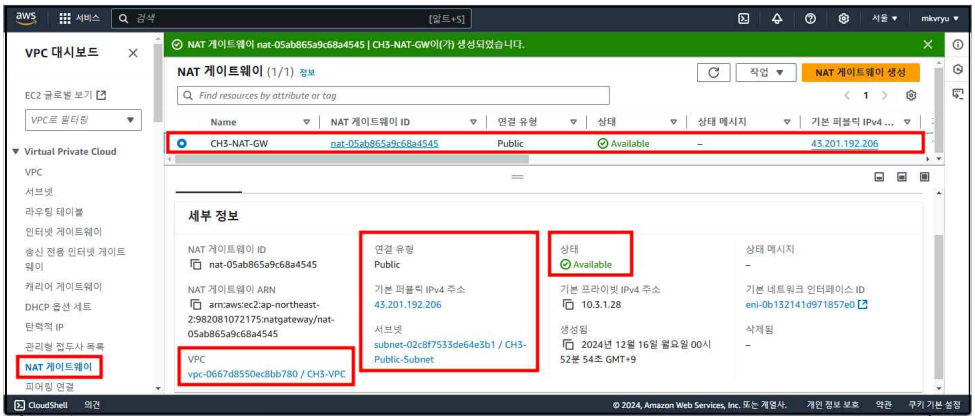
NAT 게이트웨이 생성 페이지에서 다음과 같이 설정하고 “NAT 게이트웨이 생성”을 누른다.

- 이름 => “CH3-NAT-GW”
- 서브넷 => “CH3-Public-Subnet”
- 연결 유형 => “퍼블릭”
- “탄력적 IP 할당”을 누른다.



NAT 게이트웨이 설정에서 서브넷은 “퍼블릭 서브넷”으로 선택해야 한다. 프라이빗 서브넷의 외부 인터넷 구간 통신을 하기 위해 NAT 게이트웨이를 사용하는 측면에서 프라이빗 서브넷을 선택해야 한다고 착각할 수 있다. 하지만 NAT 게이트웨이가 위치하는 서브넷을 의미하므로 인터넷 게이트웨이가 연결된 퍼블릭 서브넷을 선택해야 한다.

다시 “NAT 게이트웨이” 메뉴로 들어가면 신규 NAT 게이트웨이가 생성된 것을 확인할 수 있다.



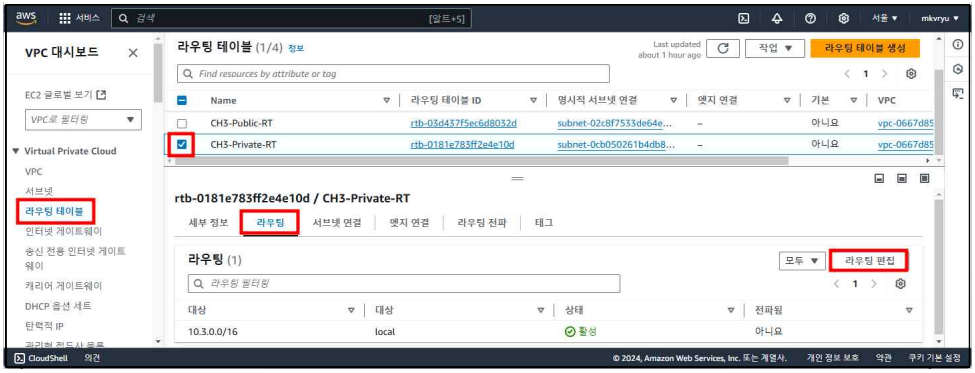
하지만 상태 정보를 확인하면 현재 대기(pending)이다. NAT 게이트웨이는 바로 생성되는 것이 아니라 약간의 대기 시간이 필요하다. 잠시 기다리면 사용 가능한 상태로 변환된다.

- 연결 유형: 외부 인터넷 구간과 연결하는 퍼블릭 유형
- 상태: 일정 시간이 지나면 “Available” 상태가 됨
- 기본 퍼블릭 IPv4 주소: NAT 게이트웨이가 사용할 고정된 공인 IP 주소
- 서브넷: NAT 게이트웨이가 생성된 서브넷으로, 우리가 생성한 퍼블릭 서브넷에 위치
- VPC: NAT 게이트웨이가 생성된 VPC

라우팅 테이블 편집하기

프라이빗 서브넷의 외부 인터넷 구간 통신을 위해 NAT 게이트웨이까지 생성했다. 하지만 프라이빗 서브넷의 라우팅 테이블은 로컬 통신 경로만 있을 뿐 외부 인터넷 구간으로 갈 수 있는 정보가 없기 때문에 라우팅 테이블 수정 작업을 한다.

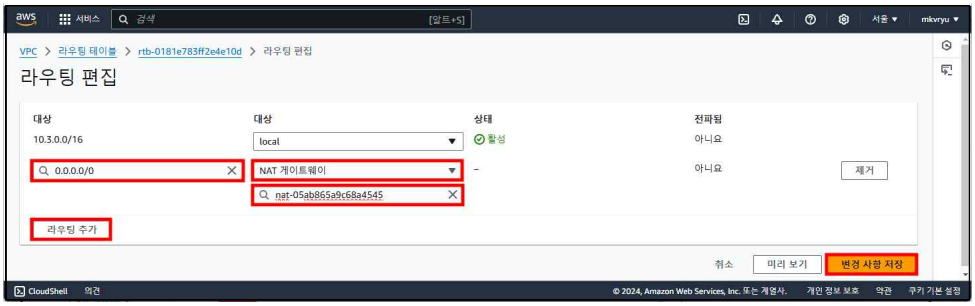
“라우팅 테이블”로 들어가서 생성한 프라이빗 라우팅 테이블 체크 후 “라우팅 탭”을 클릭하고 “라우팅 편집”을 누른다.



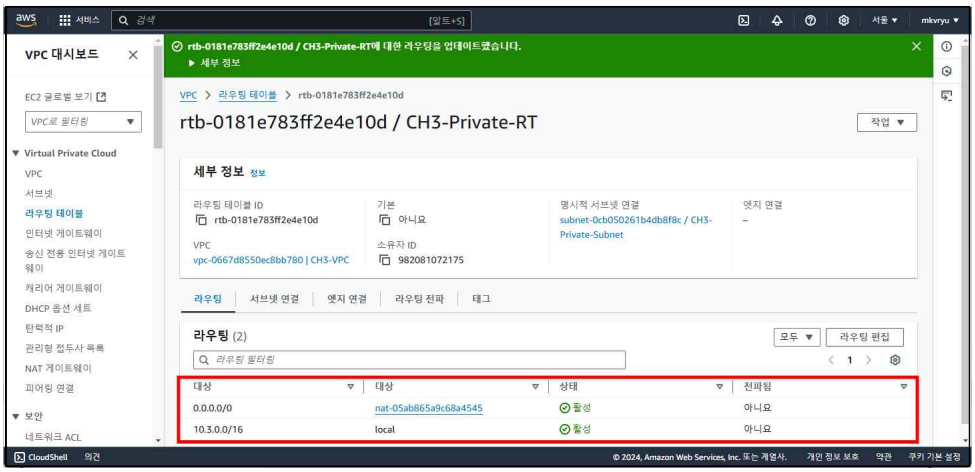
라우팅 편집 페이지에서 라우팅 추가"를 누르고 다음과 같이 설정한 후 "변경 사항 저장" 누르기

· IP CIDR 대상 => "0.0.0.0/0"(모든 IP 대역)

· 타깃 대상 => "NAT 게이트웨이"를 선택하고 생성한 "NAT 게이트웨이 ID" 선택



편집된 라우팅 정보를 확인한다.



0.0.0.0/0 모든 IP 대역에 대해 타깃 대상을 NAT 게이트웨이로 설정했다.

프라이빗 서브넷 환경 구성을 완료했다. 이번에는 프라이빗 서브넷에 EC2 자원을 생성하고 통신을 확인한다.

3.4.5. 프라이빗 서브넷 통신 확인하기

프라이빗 서브넷 환경을 구성했다. 이런 프라이빗 서브넷에 EC2 인스턴스를 생성하여 통신을 확인한다.

EC2 인스턴스 생성하기

서비스 => 컴퓨팅 => EC2 => 인스턴스 메뉴 => "인스턴스 시작"을 누르고 다음과 같이 설정한 후 "인스턴스 시작" 누르기

· 이름 및 태그 => "CH3-Private-EC2"

· 키 페어(로그인) => 기존에 생성한 키 페어 파일 선택

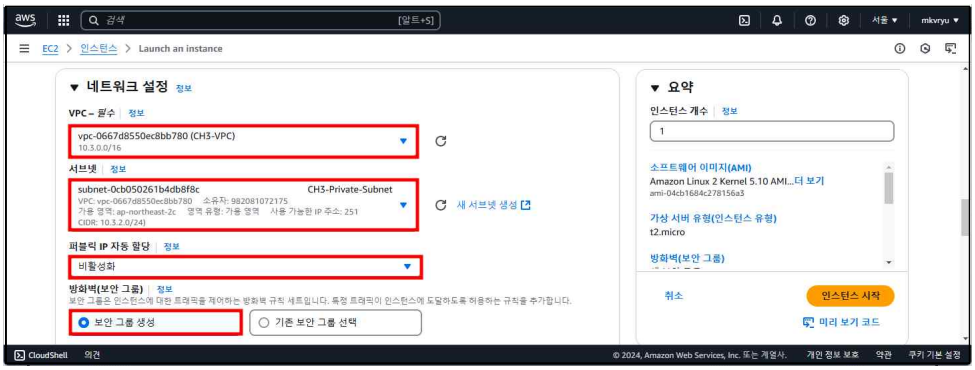
네트워크 설정에서 “편집” 누르기

· VPC => “CH3-VPC”

· 서브넷 => “CH3-Private-Subnet”

· 퍼블릭 IP 자동 할당 => “비활성화”

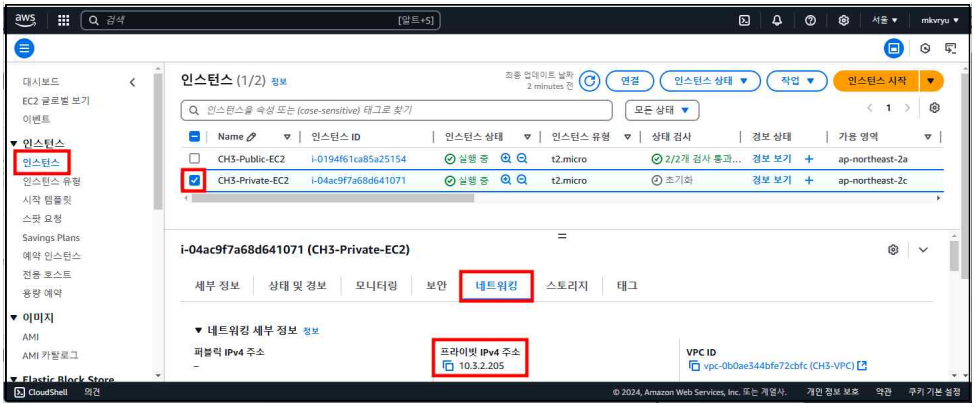
· 방화벽(보안 그룹) => “보안 그룹 생성”



고급 세부 정보에서 사용자 데이터는 다음 코드 입력(SSH 설정)

```
#!/bin/bash
(
echo "qwe123"
echo "qwe123"
) ¦ passwd --stdin ec2-user
sed -i "s/^PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config
systemctl restart sshd
```

“인스턴스” 메뉴로 들어가서 생성된 EC2 인스턴스에 체크하고 아래쪽에 “네트워킹 탭”을 클릭한다.



프라이빗 서브넷에 위치한 EC2 인스턴스에는 퍼블릭 IP 주소가 없다. 다음 SSH 접근을 위해 프라이빗 IP 주소를 메모한다.

EC2 인스턴스에서 외부 인터넷 통신 확인하기

생성된 EC2 인스턴스는 퍼블릭 IP 주소가 없는 프라이빗 서브넷에 위치한다. 현재 작업 중인 PC에서 SSH 접근을 바로 수행할 수 없는 관계로, 퍼블릭 서브넷의 EC2 인스턴스에 SSH로 접근 한 후 생성된 EC2 인스턴스의 프라이빗 IP 주소로 다시 접근한다.

```
# 프라이빗 서브넷의 EC2 인스턴스로 SSH 접근
[root@ip-10-3-1-13 ~]# ssh ec2-user@10.3.2.49 => CH3-Private-EC2의 프라이빗 IPv4 입력
The authenticity of host '10.3.2.49 (10.3.2.49)' can't be established.
ED25519 key fingerprint is SHA256:p4oALolaMnhcDOnXWfqBxrgHTgQzukTDTTuJ5UTdb1s.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes => 입력
Warning: Permanently added '10.3.2.49' (ED25519) to the list of known hosts.
ec2-user@10.3.2.49's password: => qwe123 입력, 입력하는 비밀번호는 화면에 표시되지 않는다.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-3-2-49 ~]$
```

```
# 프라이빗 EC2 인스턴스 SSH 터미널 접속
[ec2-user@ip-10-3-2-49 ~]$ ping google.com
PING google.com (172.217.25.174) 56(84) bytes of data.
64 bytes from kix06s19-in-f14.1e100.net (172.217.25.174): icmp_seq=1 ttl=104 time=35.4 ms
64 bytes from syd09s13-in-f14.1e100.net (172.217.25.174): icmp_seq=2 ttl=104 time=34.5 ms
64 bytes from syd09s13-in-f14.1e100.net (172.217.25.174): icmp_seq=3 ttl=104 time=35.2 ms
64 bytes from syd09s13-in-f14.1e100.net (172.217.25.174): icmp_seq=4 ttl=104 time=34.6 ms

ctrl + c
```

```
[ec2-user@ip-10-3-2-49 ~]$ curl ipinfo.io/ip
13.124.34.143
```

3.4.6. 실습을 위해 생성된 모든 자원 삭제하기

Amazon EC2 인스턴스 삭제하기

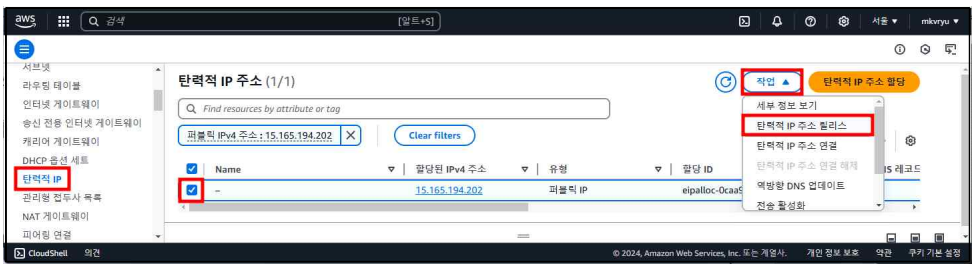
서비스 => 컴퓨팅 => EC2 => 인스턴스 => 삭제할 인스턴스 선택 => 인스턴스 상태 => 인스턴스 종료(삭제)

NAT 게이트웨이 삭제하기

서비스 => 네트워킹 및 콘텐츠 전송 => VPC => NAT 게이트웨이 => 삭제할 NAT 게이트웨이 선택 => 작업 => NAT 게이트웨이 삭제

탄력적 IP 삭제하기

서비스 => 네트워킹 및 콘텐츠 전송 => VPC => 탄력적 IP => 삭제할 탄력적 IP 선택 => 작업 => 탄력적 IP 릴리즈



VPC 삭제하기

서비스 => 네트워킹 및 콘텐츠 전송 => VPC => VPC => 삭제할 VPC 선택 => 작업 => VPC 삭제