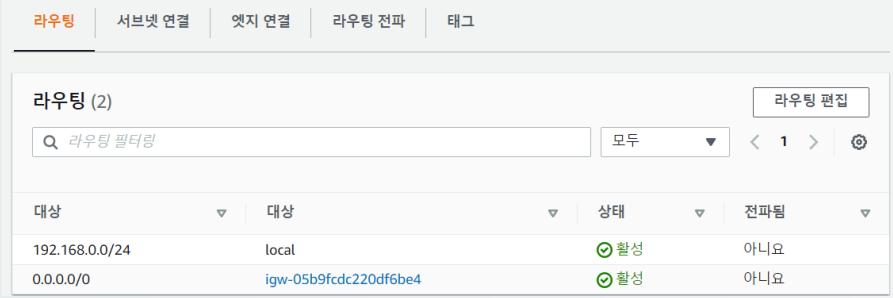
AWS VPC(Virtual Private Cloud)

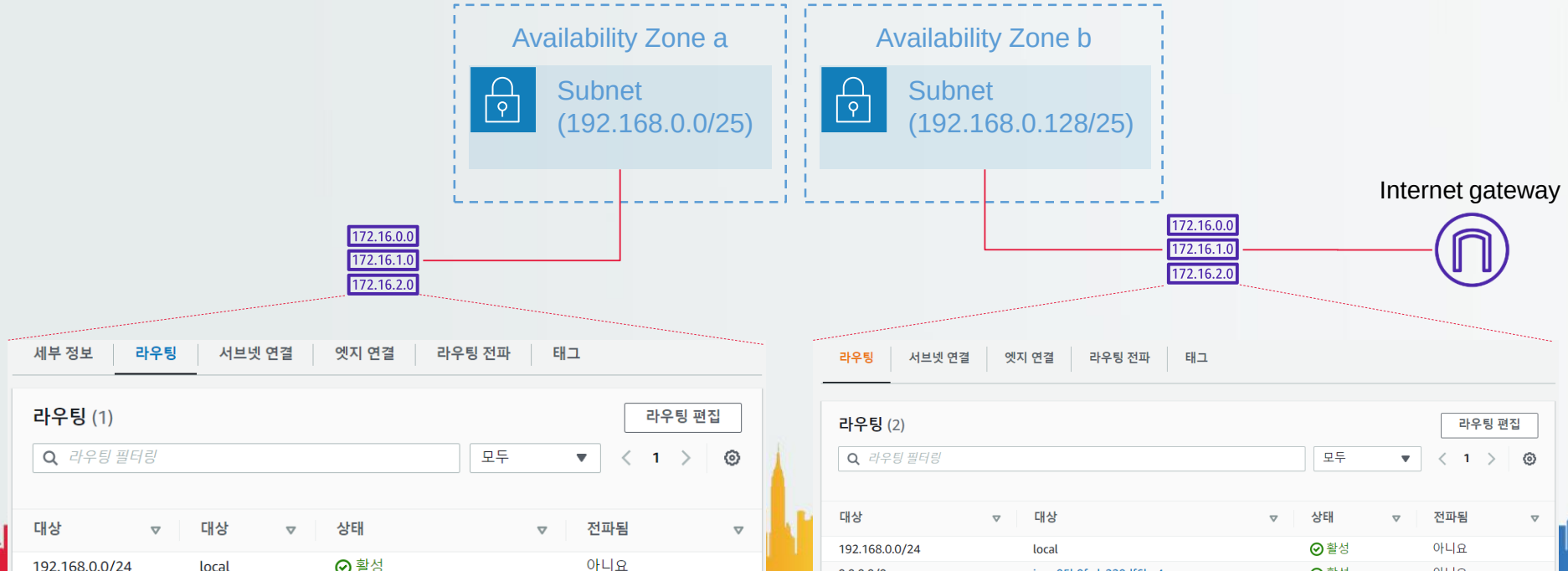
- 라우팅 테이블(Route Table)
  - 라우터 역할을 수행하며, VPC 내부에 위치한 EC2 인스턴스에서 대상(Target) 혹은 목적지(Destination)를 찾기 위한 경로를 저장한 테이블
  - 기본 라우팅 테이블: VPC에는 기본 라우팅 테이블이 자동 생성되며, VPC 내의 서브넷에 별도 생성한 라우팅 테이블을 연결하지 않으면, 기본 라우팅 테이블에 자동 연결됨



  - VPC의 IP CIDR Block이 192.168.0.0/24 이면, 위 Route Table은 해당 IP 대역에 위치한 대상들은 VPC 내에서 찾도록 지정하고, 0.0.0.0/0을 통해 그 외의 대상들은 인터넷 게이트웨이를 통해 인터넷 망에서 찾도록 함

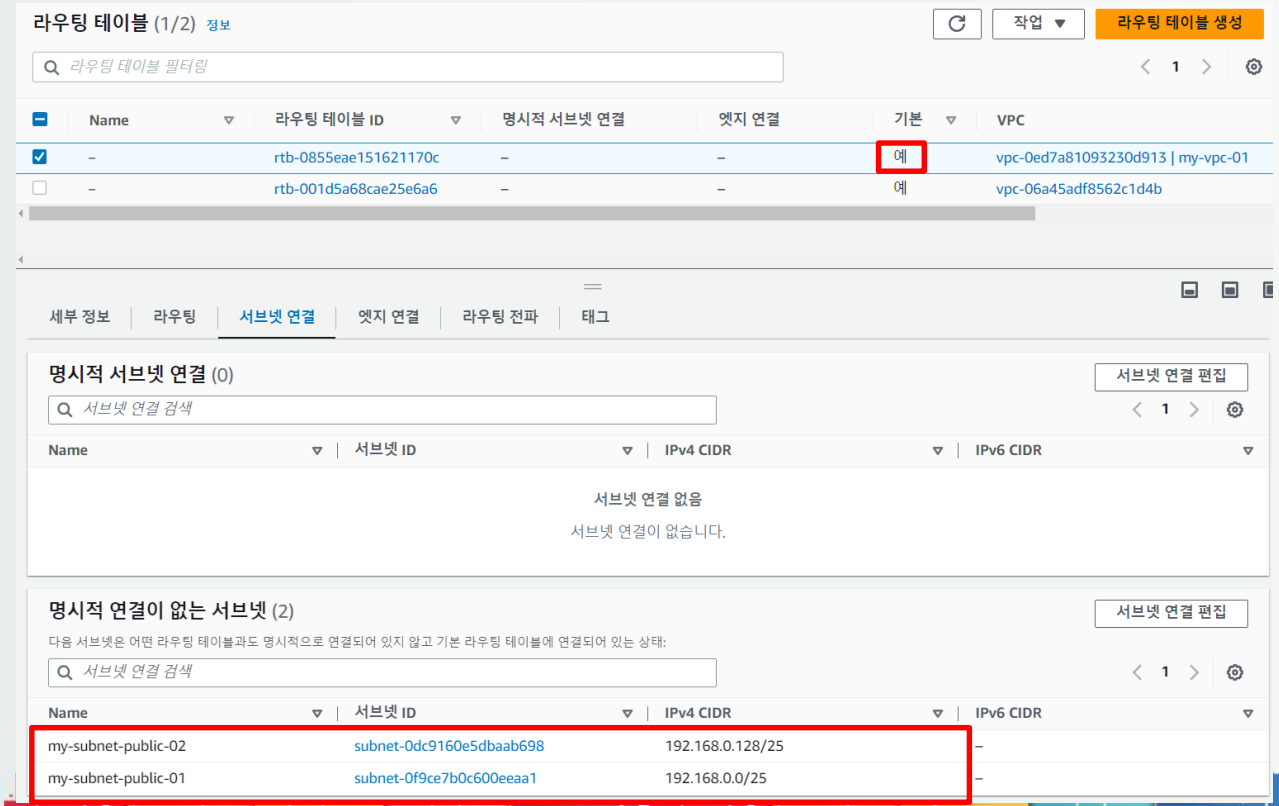
AWS VPC(Virtual Private Cloud)

- **라우팅 테이블(Route Table)**
  - **라우팅 테이블은 서브넷 별로 다르게 설정하는 것이 가능**
  - **단, VPC 내부에 대한 라우팅 규칙은 모든 라우팅 테이블에 항상 기본으로 존재**



라우팅 테이블 실습

- **라우팅 테이블(Route Table)**
  - **명시적 서브넷 연결**
    - 서브넷과 라우팅 테이블 직접 연결
  - **묵시적 서브넷 연결**
    - 명시적 연결이 없는 경우 기본 라우팅 테이블에 연결



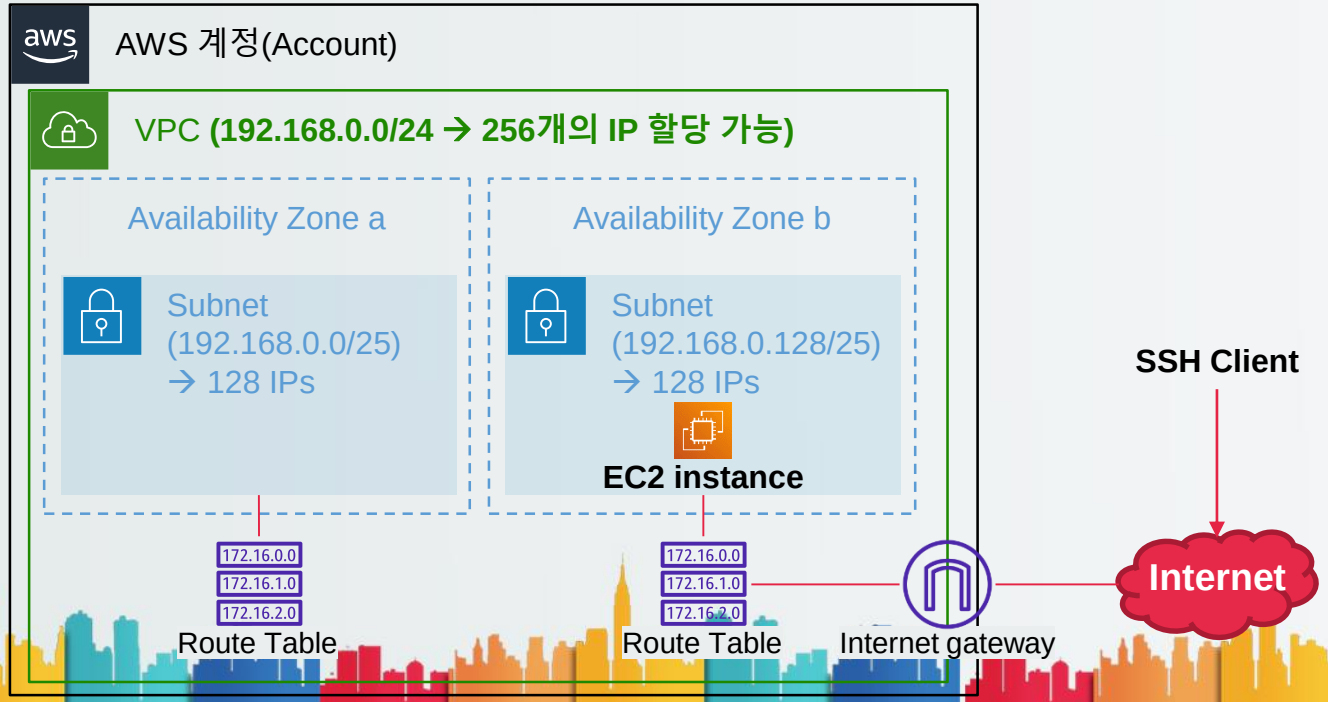
Public Subnet vs Private Subnet

- 퍼블릭 서브넷(Public Subnet)과 프라이빗 서브넷(Private Subnet)
  - 퍼블릭 서브넷이란?
    - 라우팅 테이블을 통해 인터넷 게이트웨이와 연결된 서브넷
  - 프라이빗 서브넷이란?
    - 인터넷 게이트웨이와 연결되지 않은 서브넷
    - 즉, 인터넷을 통해 외부에서 접근이 불가능한 서브넷

- 왜 프라이빗 서브넷이 필요한가?
  - 인터넷에 연결된 장치(PC 혹은 스마트폰 등)는 항상 공격의 대상이 됨
  - 인터넷망에 직접 연결되지 않아도 되는 서버는 가급적 네트워크에서 격리된 곳에 위치시키는 것이 보안 원칙

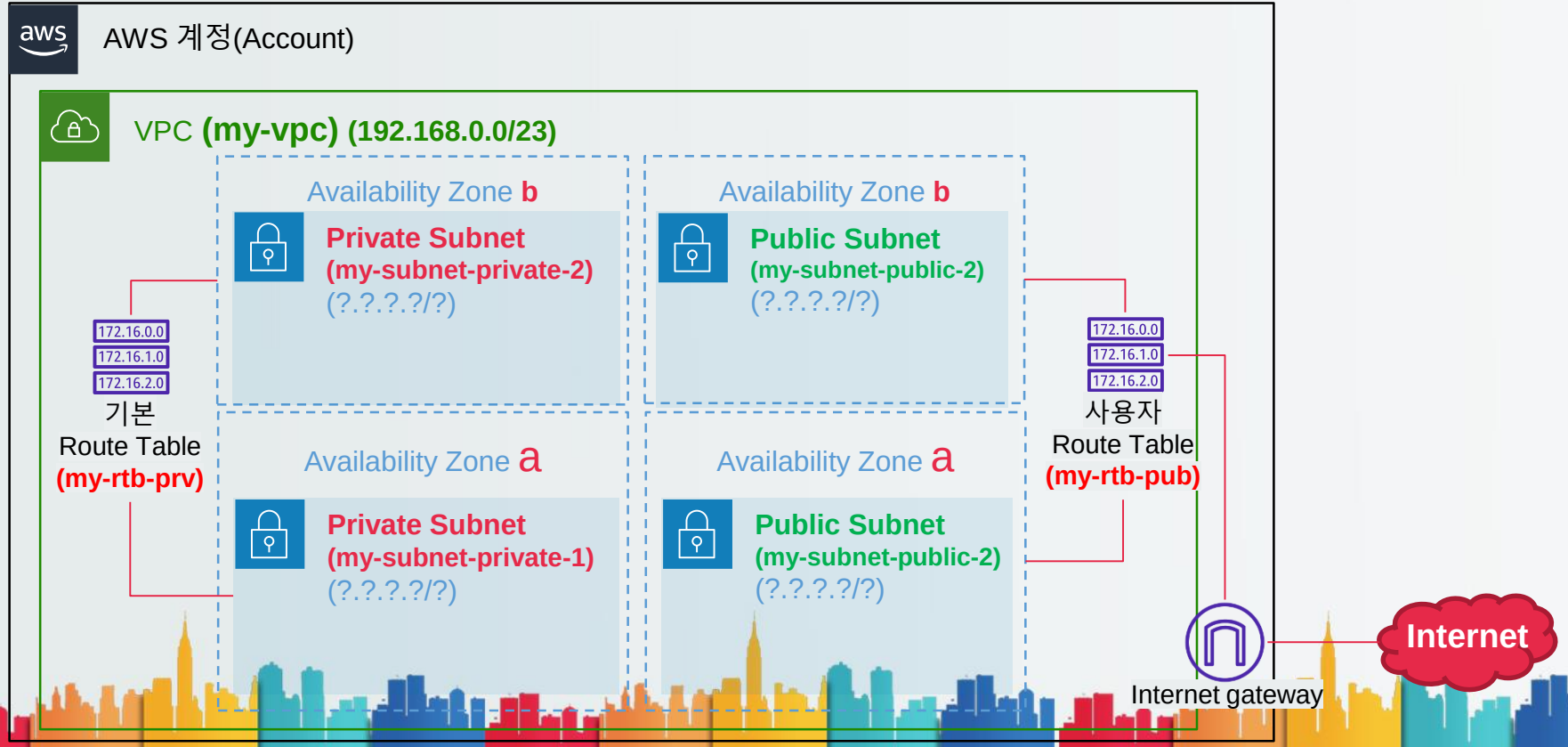
Public Subnet vs Private Subnet

- **퍼블릭/프라이빗 서브넷 생성 방법**
  - **라우팅 테이블을 서로 다르게 설정 → 특정 서브넷의 라우팅 테이블에만 인터넷 게이트웨이 연결**
  - **라우팅 테이블은 기본으로 VPC 내의 Subnet 간에는 통신이 가능하도록 설정됨**



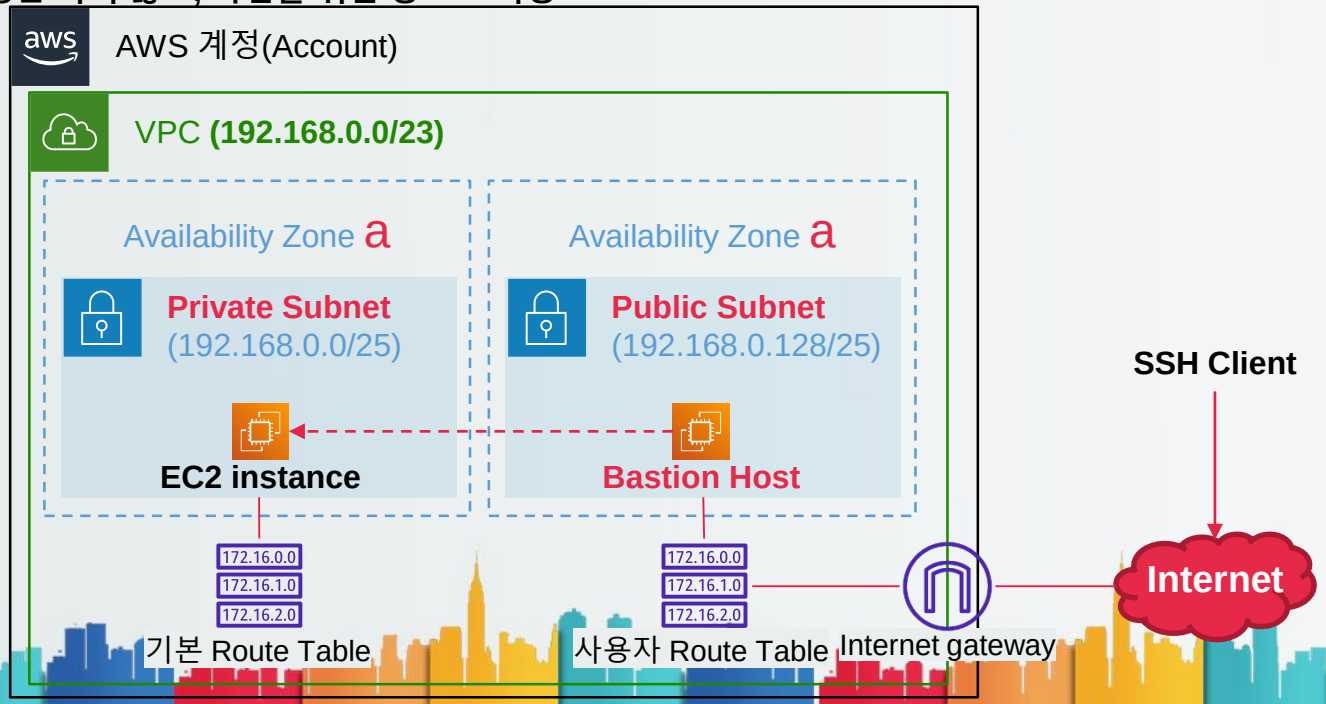
Public Subnet vs Private Subnet

- 퍼블릭/프라이빗 서브넷 생성 실습 → 같은 크기를 갖는 4개의 서브넷 생성



Public Subnet vs Private Subnet

- 프라이빗 서브넷에 위치한 인스턴스에 접근하는 방법 실습
  - Bastion Host
    - 외부에서 접근 가능한 EC2 인스턴스로서, 내부 네트워크에 위치한 서버에 접속하기 위한 접속 전용 호스트
    - 서비스 운영은 하지 않고, 작업을 위한 용도로 사용



Public Subnet vs Private Subnet

- 프라이빗 서브넷에 위치한 인스턴스에 접근하는 방법 실습
  - Bastion Host
    - 1. Bastion Host에 private key 복사

```
scp -i ~/.ssh/labsuser.pem ~/.ssh/labsuser.pem ec2-user@Public IP:/home/ec2-user/
```

    - 2. Bastion Host에 접속

```
ssh -i ~/.ssh/labsuser.pem ec2-user@Public IP
```

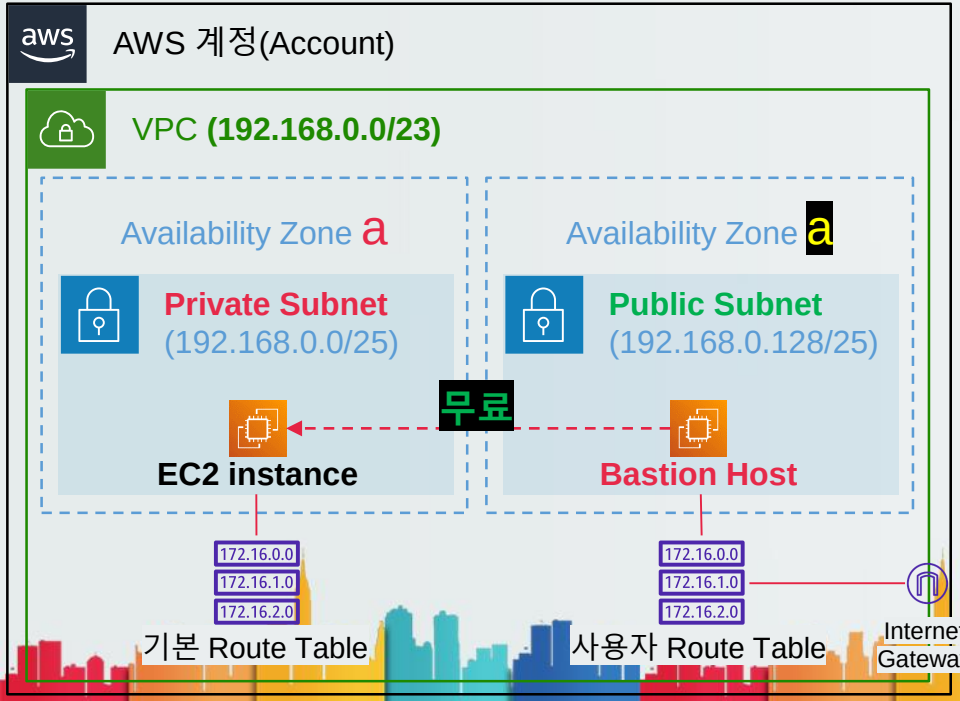
    - 3. Bastion Host에서 프라이빗 서브넷의 서버에 접속

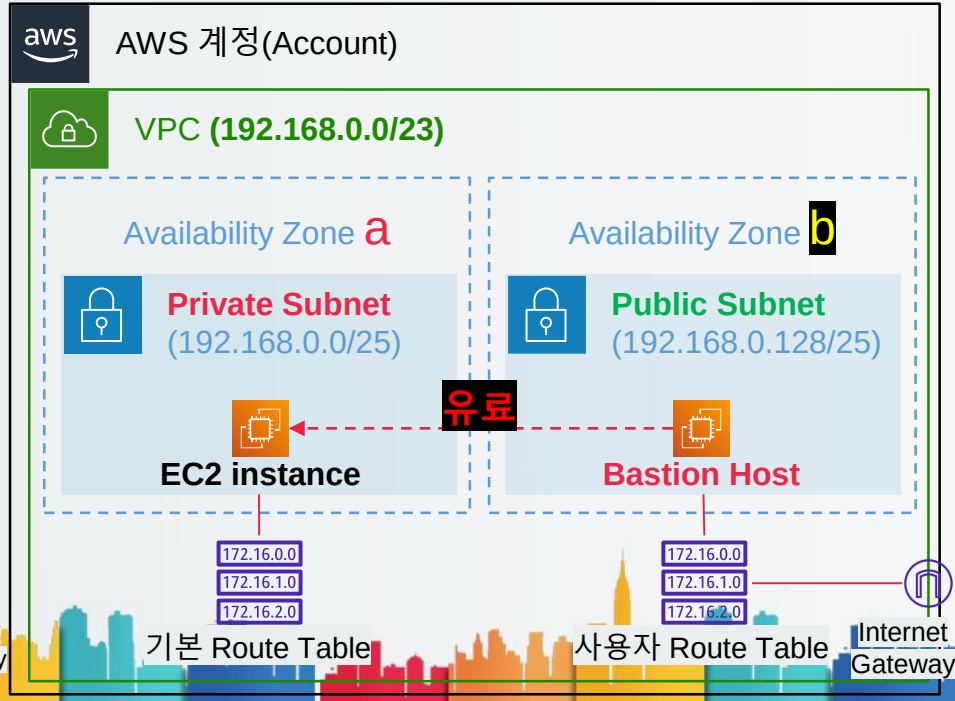
```
ssh -i labsuser.pem ec2-user@Private IP
```

    - 이 방법은 bastion host에 대한 SSH key를 강탈당할 경우, bastion host에 저장된 키 역시 모두 강탈당할 수 있어 위험성이 존재
    - 이 절차를 손쉽게 또 더욱 보안성이 뛰어나게 처리하는 SSH agent forwarding 이라는 방법이 존재

Public Subnet vs Private Subnet

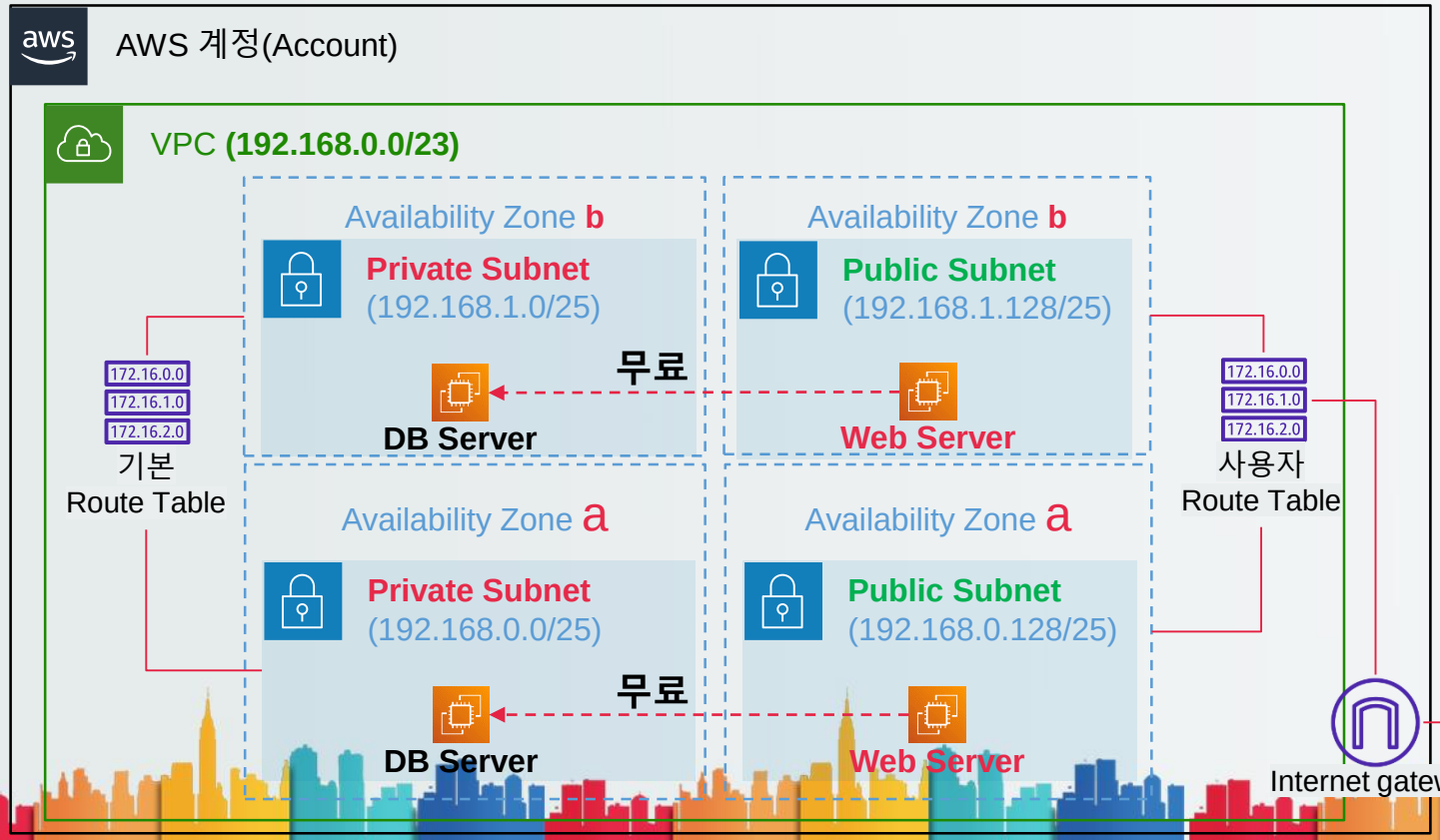
- 퍼블릭 서브넷과 프라이빗 서브넷을 같은 가용영역(AZ)으로 설정한 이유
  - AWS에서 가용영역은 데이터 센터를 의미
  - 가용영역 간에 통신은 데이터 전송량에 따라 비용이 발생
  - 같은 가용영역이면 서브넷이 달라도 비용이 발생하지 않음





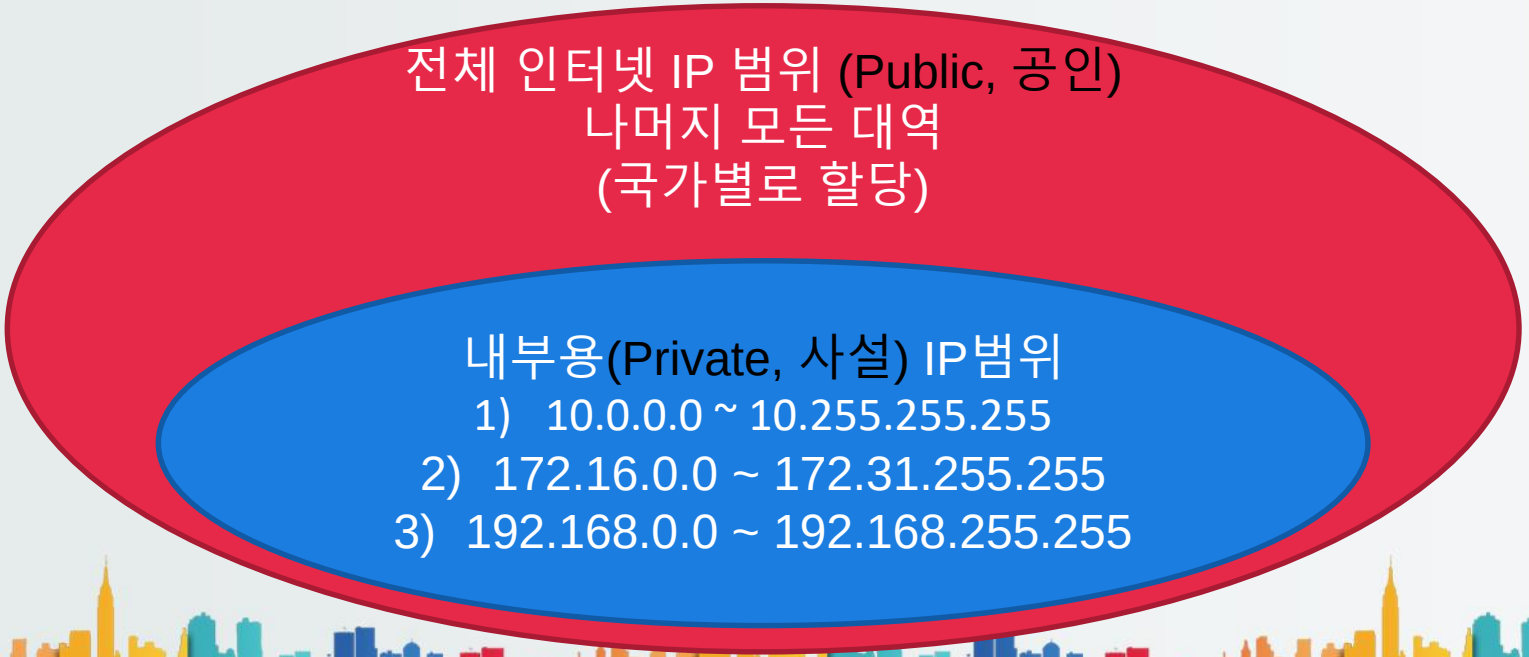
Public Subnet vs Private Subnet

- 비용효율적으로 고가용성(High Availability)을 확보하기 위한 구조



인터넷과 공인 IP

- **공인 IP(Public IP)**
  - **네트워크에서 장치를 특정하는 방법은 IP**
  - **인터넷은 거대한 하나의 IP 범위를 갖는 네트워크**

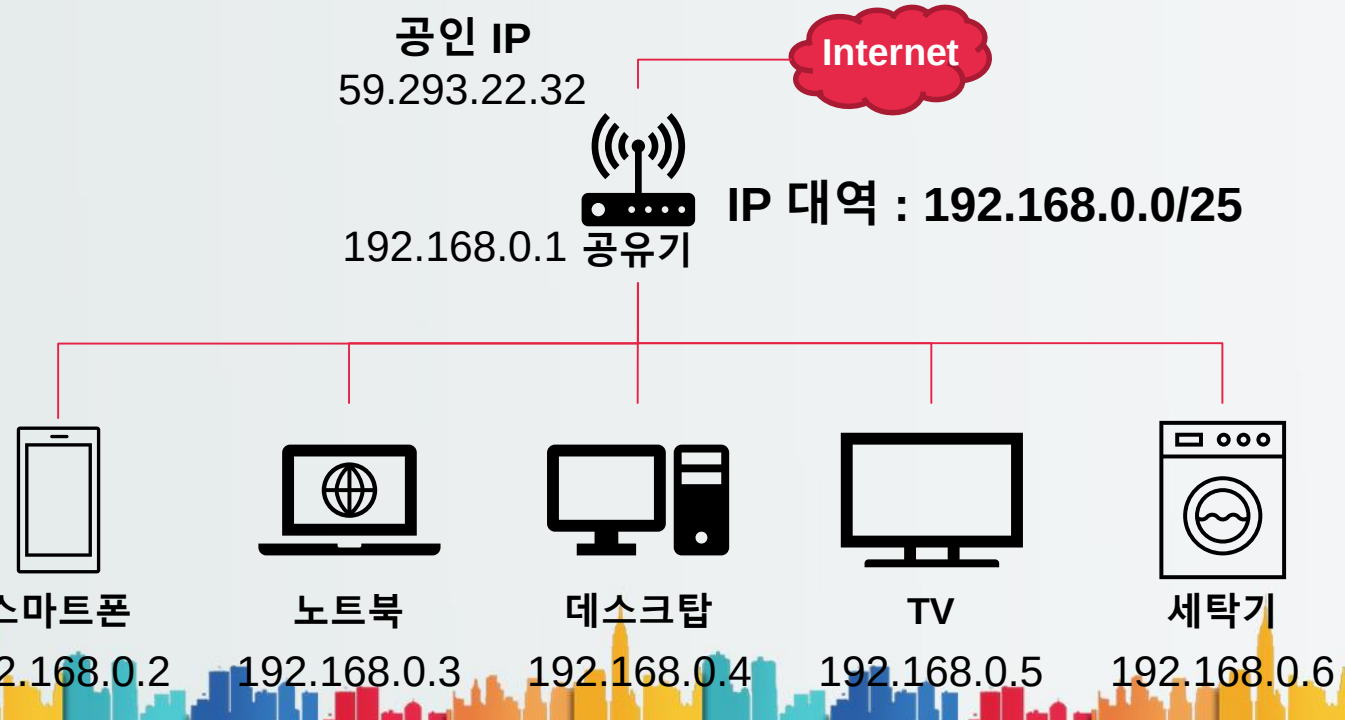


인터넷과 공인 IP

- 공인 IP(Public IP)
  - 인터넷의 IP는 국제기구에서 관리
    - ICANN 이라는 국제 기구에서 총괄하며, KISA(한국인터넷진흥원)에서 한국의 IP관리를 위임 받아 처리
    - KISA에서 관리하는 IP를 ISP(Internet Service Provider)에서 실제 사용자에게 할당
      - ISP는 SK브로드밴드, LGU+, KT 등의 인터넷 사업자를 의미
  - 인터넷에 연결하기 위해서 장치들은 인터넷으로 연결되는 경로와 인터넷에서 사용할 공인 IP가 반드시 있어야 함
    - AWS에서는 EIP나 자동할당 IP를 통해 공인 IP를 할당
    - 인터넷 게이트웨이와 라우팅 테이블을 통해 인터넷 연결 경로 생성

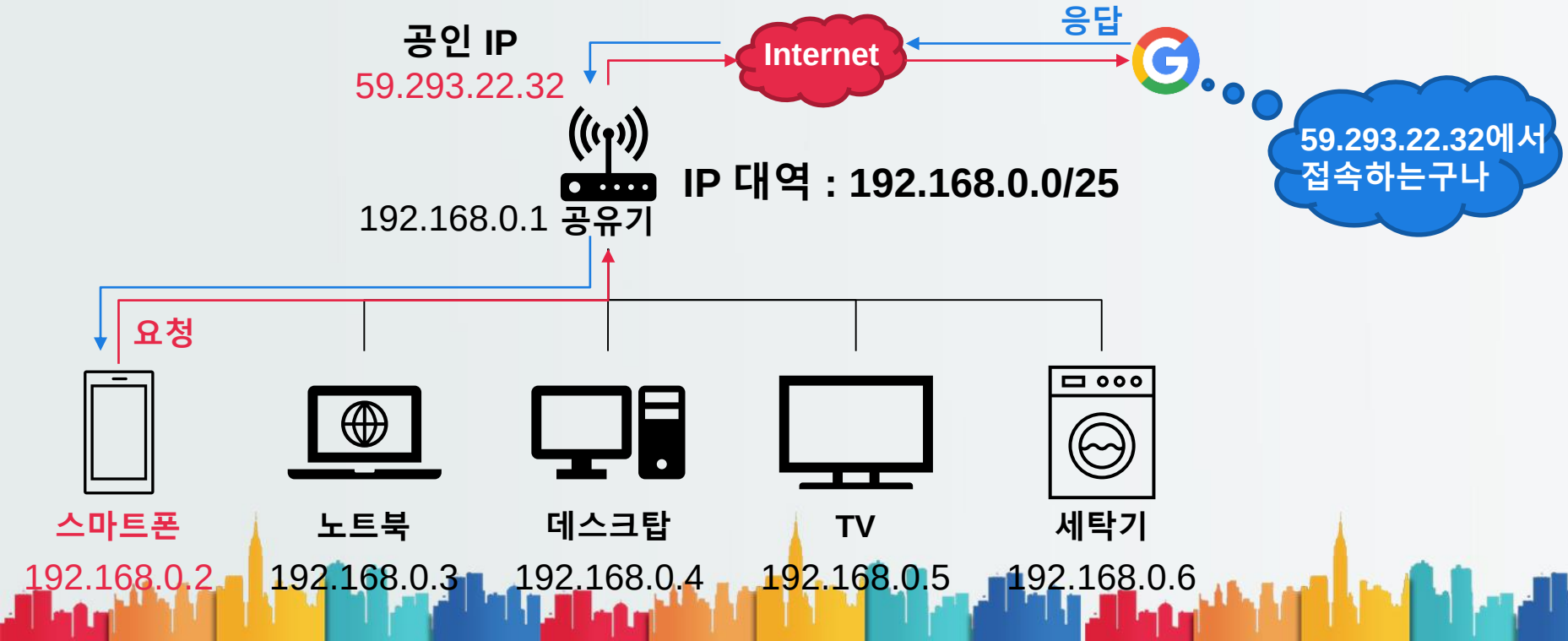
인터넷과 공인 IP

- 가정에서의 공유기를 통한 인터넷 연결
  - ISP에서 공유기에 공인 IP를 할당
  - 공유기에 연결된 기기는 공인 IP를 받지 않는데 어떻게 인터넷을 사용할 수 있는가? → NAT



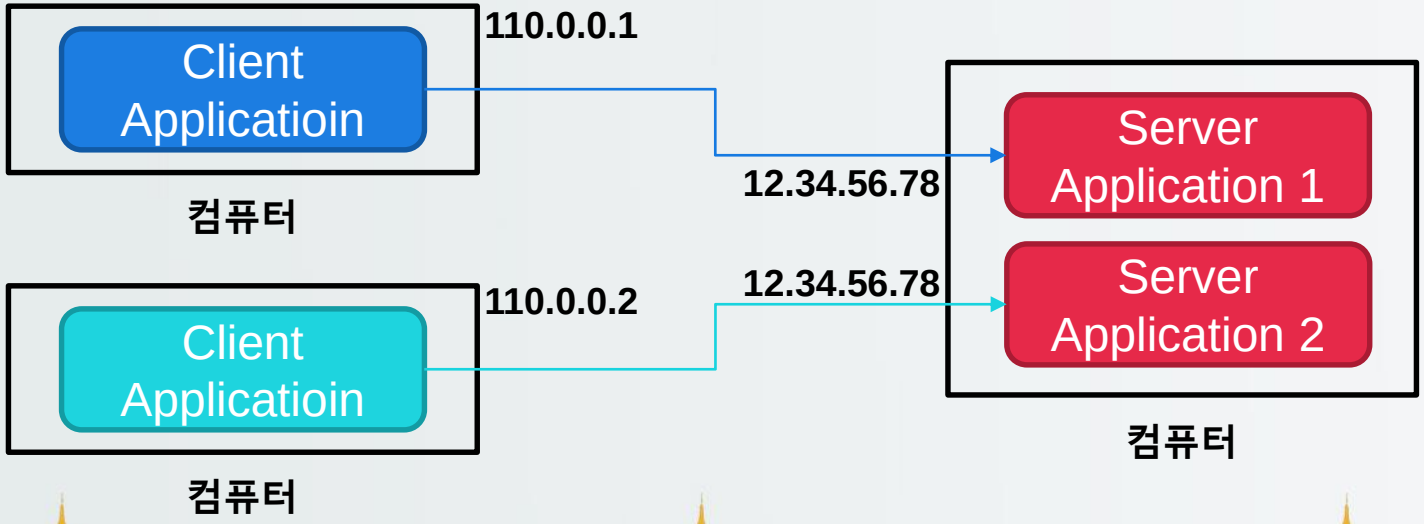
인터넷과 공인 IP

- **NAT(Network Address Translation)**
  - 공유기에서 내부에 연결된 장치들이 인터넷으로 접근할 때, 공유기에 할당된 공인 IP를 이용하도록 지원



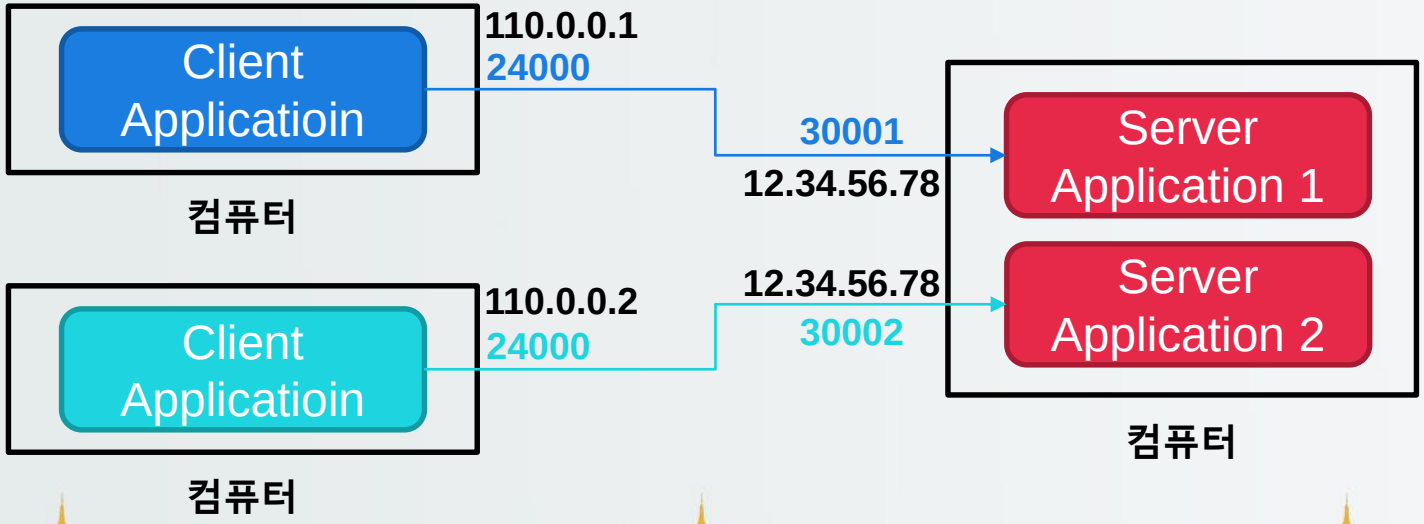
인터넷과 공인 IP

- **IP와 Port**
  - IP는 네트워크에 연결된 대상(컴퓨터)을 특정할 수 있음
  - 그렇다면 그 컴퓨터 내에서 여러 개의 프로그램이 네트워크를 사용하려 한다면 각각은 어떻게 구분하는가?



인터넷과 공인 IP

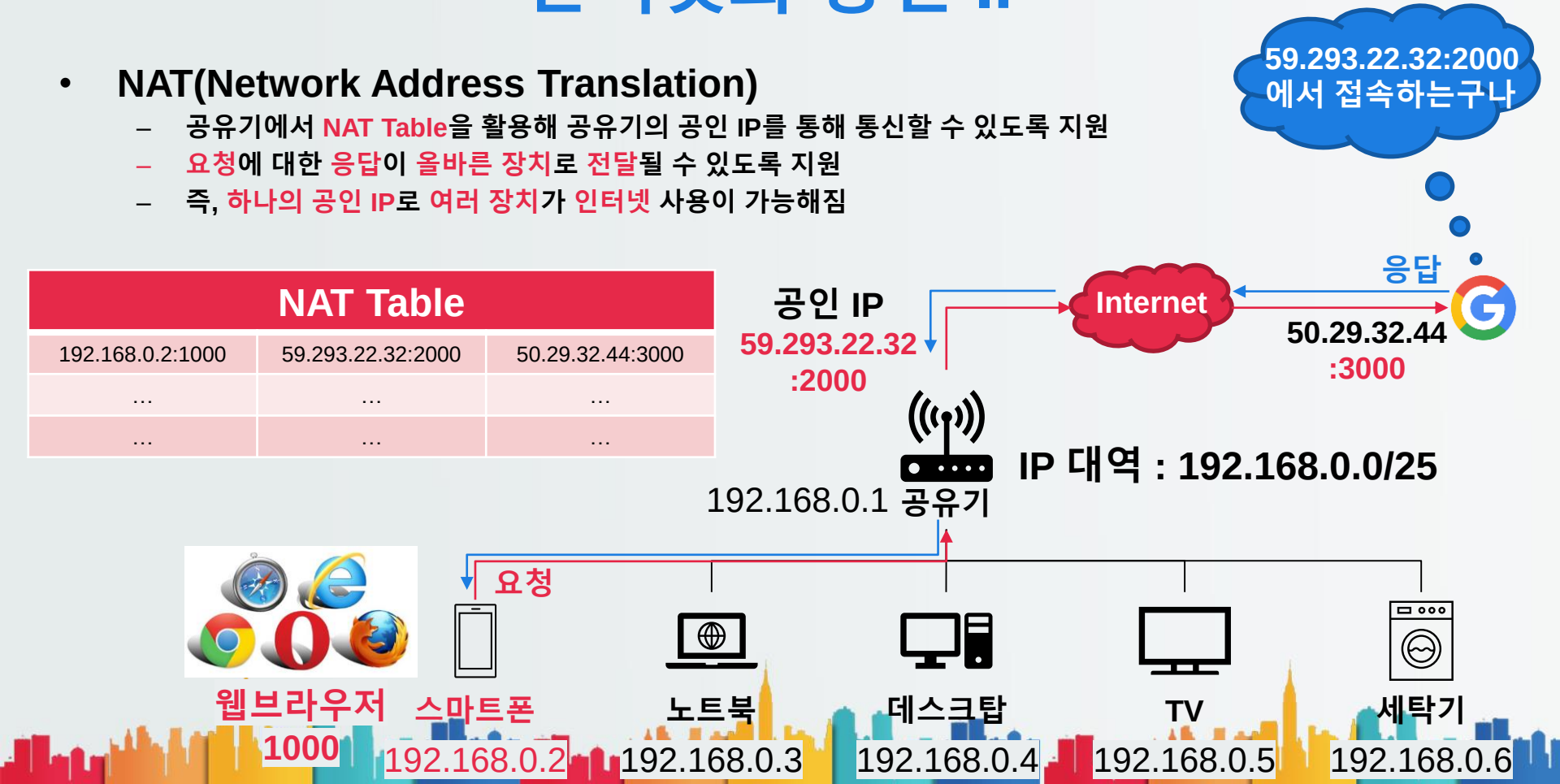
- **IP와 Port**
  - 포트(Port)
    - 하나의 컴퓨터에 대해서 네트워크를 사용하는 어플리케이션이 여러 개일 경우 이들을 구분하기 위한 ID 역할을 수행
    - 포트는 하나의 컴퓨터 내에서 유일하게 어플리케이션을 구분 (0 ~ 65535까지 사용 가능)



인터넷과 공인 IP

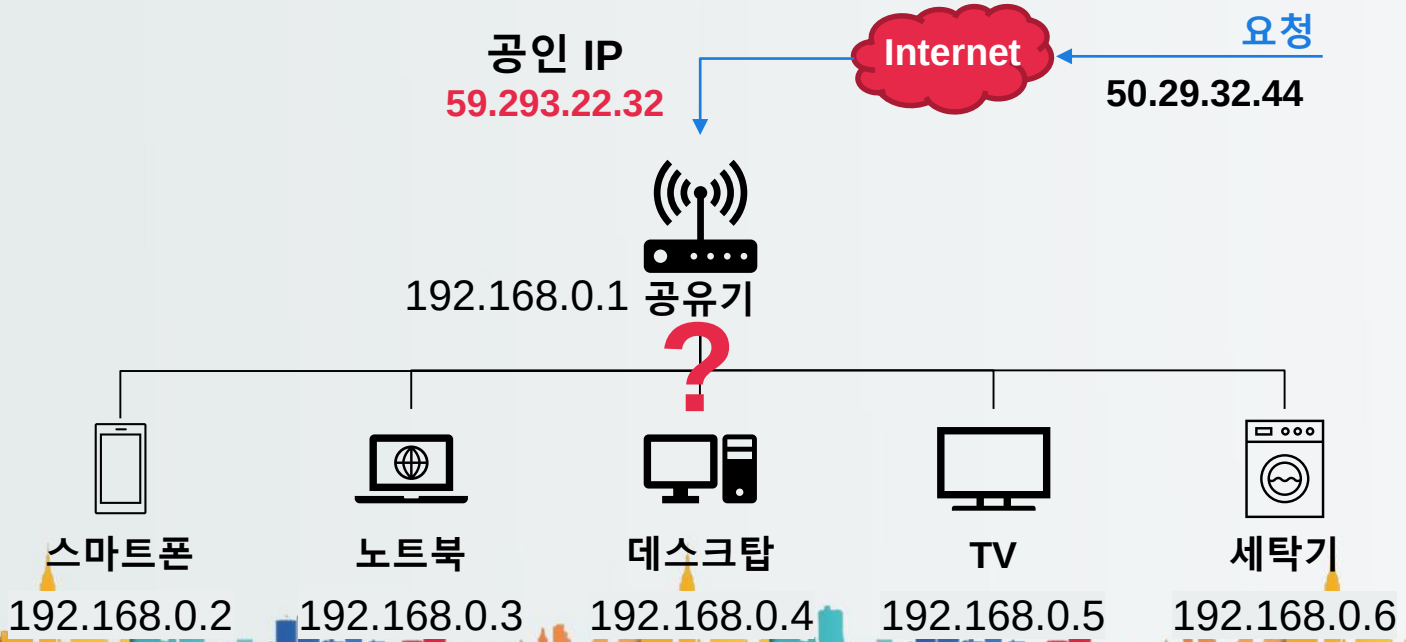
- **NAT(Network Address Translation)**
  - 공유기에서 NAT Table을 활용해 공유기의 공인 IP를 통해 통신할 수 있도록 지원
  - 요청에 대한 응답이 올바른 장치로 전달될 수 있도록 지원
  - 즉, 하나의 공인 IP로 여러 장치가 인터넷 사용이 가능해짐

NAT Table		
192.168.0.2:1000	59.293.22.32:2000	50.29.32.44:3000
…	…	…
…	…	…



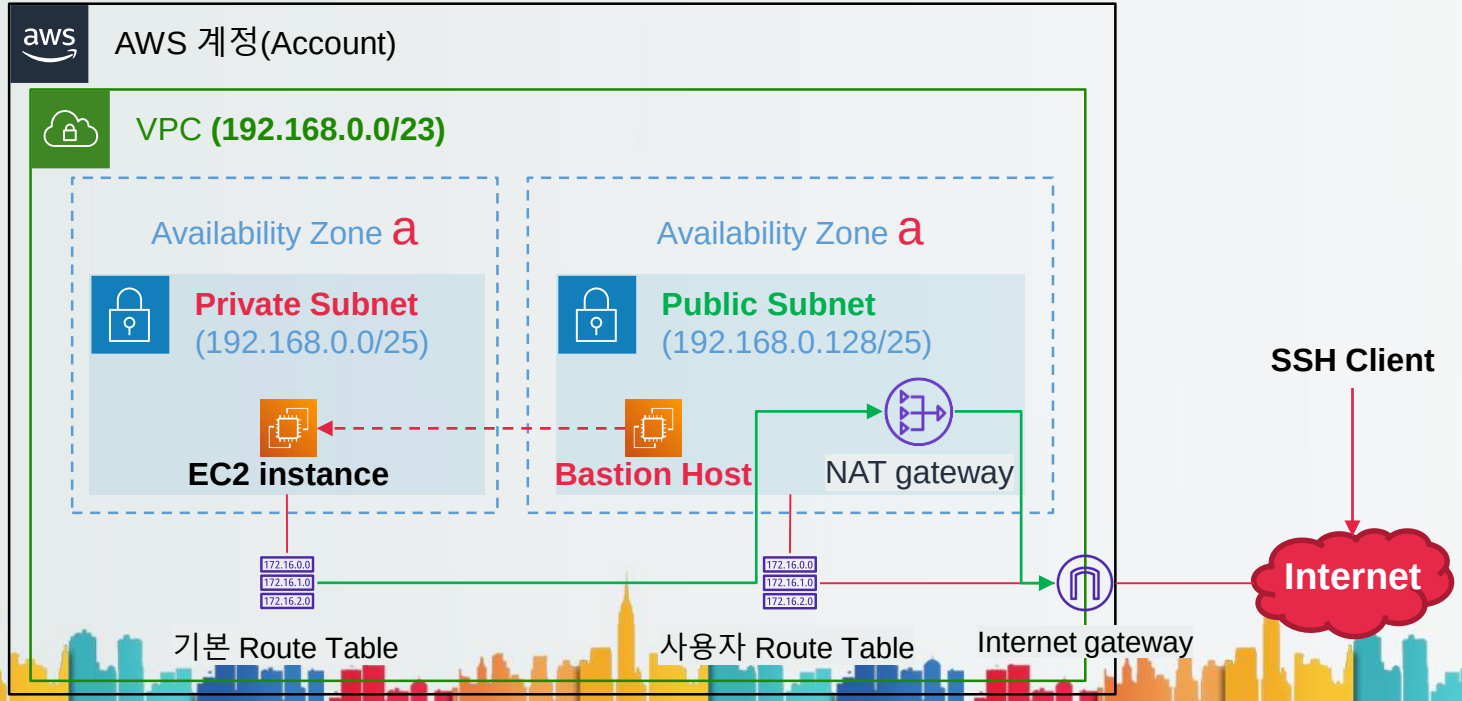
인터넷과 공인 IP

- **NAT(Network Address Translation)**
  - 단, 인터넷에 있는 다른 장치에서는 공유기에 연결된 장치에 직접 접근 불가
  - 공인 IP는 공유기에만 할당되어 있으므로, 특정할 수 있는 장치는 공유기 뿐



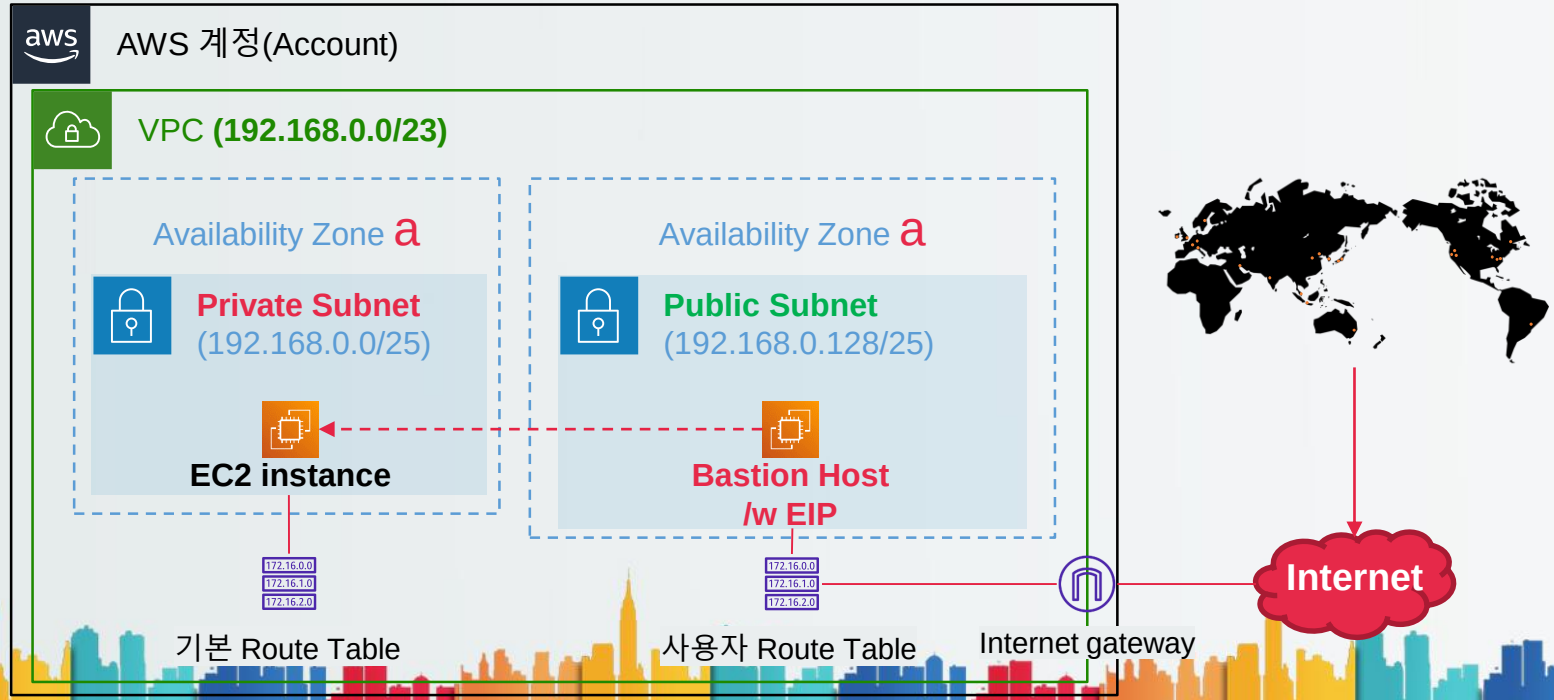
NAT를 이용한 인터넷 사용

- **NAT를 이용한 프라이빗 서브넷에서의 인터넷 접근**
  - 프라이빗 서브넷에서 인터넷으로 접근 가능
  - 인터넷에서 NAT를 통해 내부로 접근 불가능



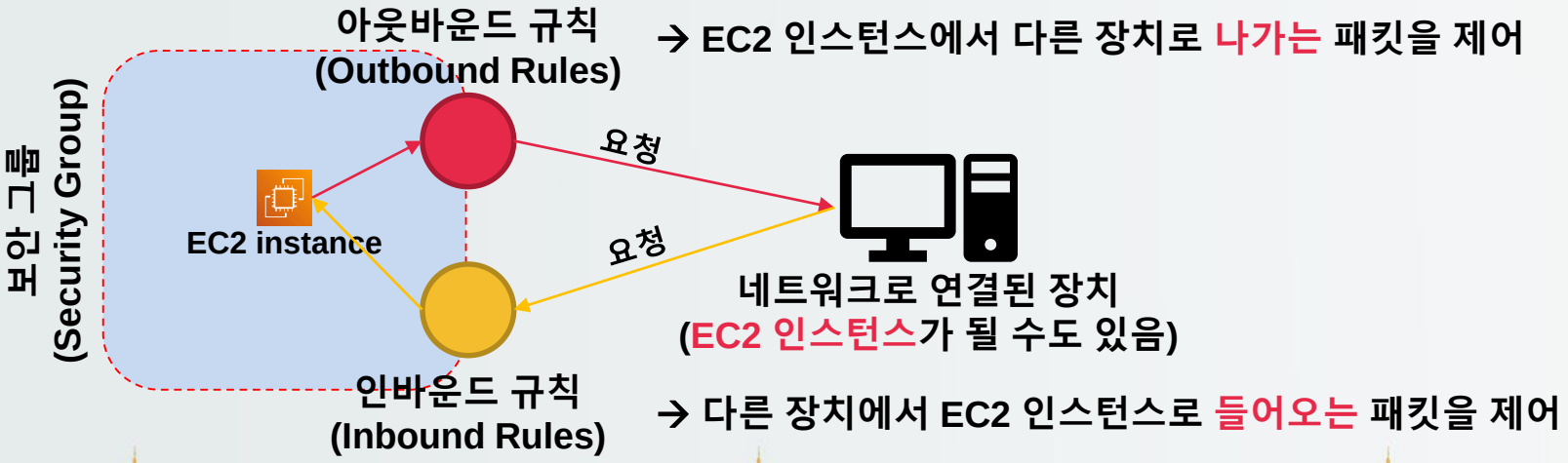
네트워크에 대한 접근 제어

- 특정 EC2 인스턴스에 접근할 수 있는 네트워크 대역을 한정하여, 공격의 위험성을 제거
- 접근하는 대상을 제한하지 않으면 전세계 어디에서든 공인 IP를 보유한 EC2 인스턴스에 접근 가능



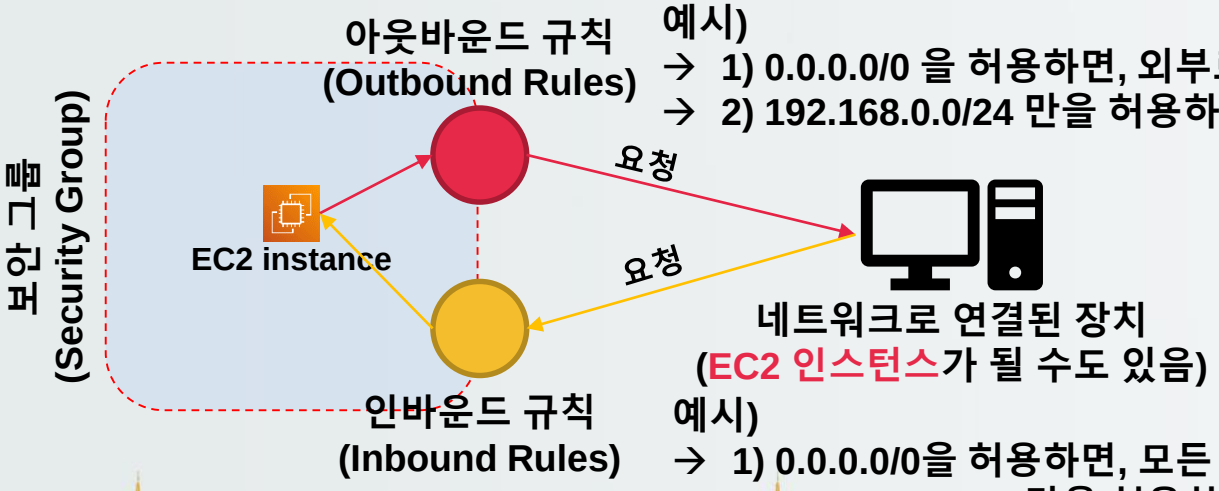
네트워크에 대한 접근 제어

- 보안 그룹(Security Group)을 통해 EC2 인스턴스에 접근 가능한 네트워크 영역을 제한
  - 아웃바운드/인바운드 규칙을 통해 들어오고 나가는 트래픽을 제어
    - 트래픽: 네트워크를 통해 교환되는 패킷의 흐름
    - 패킷: 네트워크를 통해 장치 간에 교환되는 데이터



네트워크에 대한 접근 제어

- **보안 그룹(Security Group) 설정**
  - 아웃바운드와 인바운드에 **아무런 규칙이 없으면** 어떤 **트래픽**도 **허용하지 않음**
  - 아웃바운드 규칙에서 EC2 인스턴스에서 접근할 수 있는 외부의 네트워크 대역을 설정
  - 인바운드 규칙에서 EC2 인스턴스로 접근할 수 있는 외부의 네트워크 대역을 설정
  - 외부의 네트워크 대역은 인터넷이나 혹은 또 다른 네트워크일 수 있음



예시)
→ 1) 0.0.0.0/0 을 허용하면, 외부로 나가는 모든 트래픽 허용
→ 2) 192.168.0.0/24 만을 허용하면, 해당 대역으로만 트래픽 허용

예시)
→ 1) 0.0.0.0/0을 허용하면, 모든 네트워크에서의 접근을 허용
→ 2) 192.168.0.0/24만을 허용하면, 해당 대역에서의 접근만 허용

네트워크에 대한 접근 제어

- 보안 그룹(Security Group) 설정
  - EC2 인스턴스에서 아웃바운드 허용된 곳으로 보낸 요청에 대한 응답은 인바운드 규칙과 상관없이 허용
  - 단, 외부에서 시작된 요청은 인바운드 규칙에 의해 통제됨
    - 아래 예시에서 192.168.1.0/24에서 EC2 인스턴스로 접근 불가