

## Lab7. Create Private Subnet, NAT Gateway and Network Connection Test

### 목적

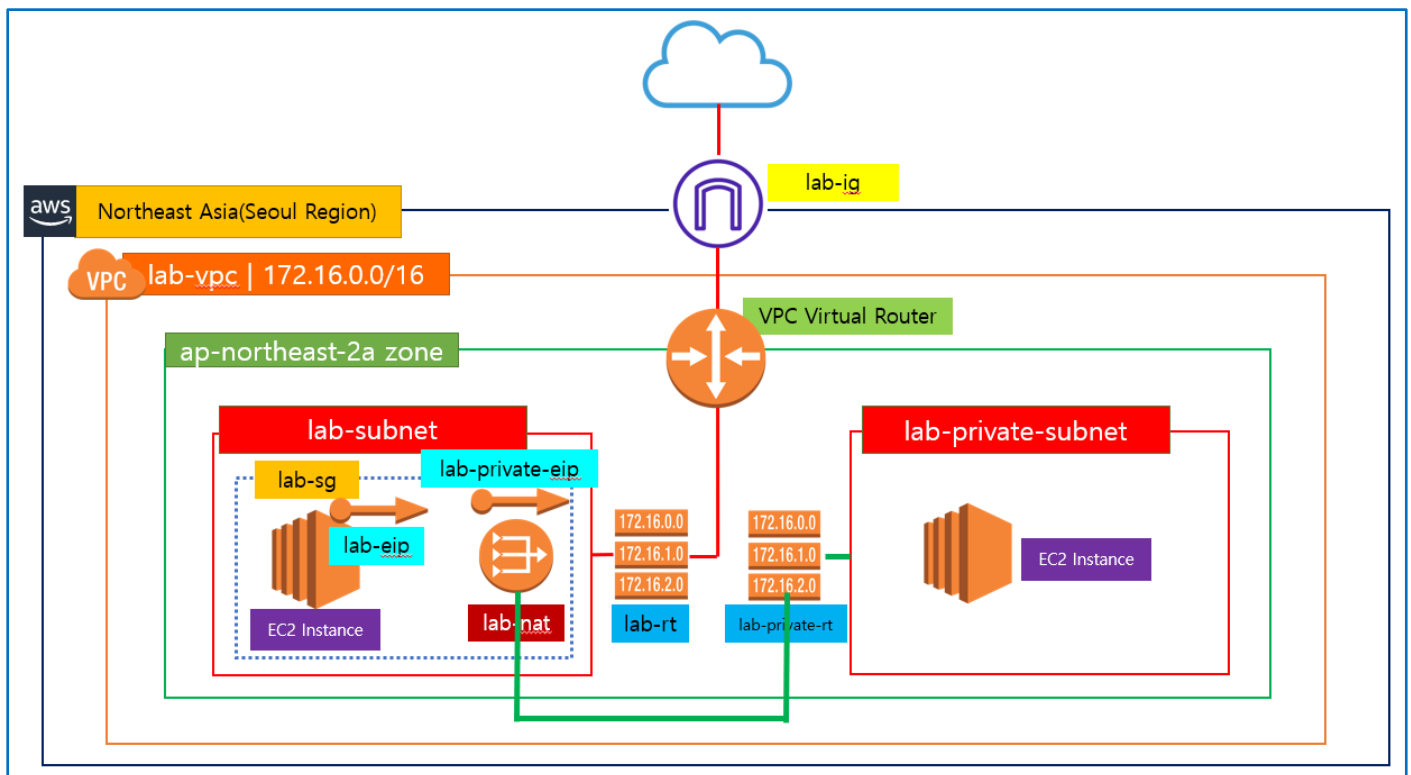
이번 실습에서는 Private Subnet에서 NAT Gateway를 통해 외부 인터넷을 사용가능 하도록 설정한다. 먼저 NAT Gateway에서 사용할 EIP를 생성하고, Private Subnet을 생성하며, NAT Gateway를 생성한다. 그 후, Private Subnet을 위한 Routing Table을 생성한다. 생성한 Routing Table을 Private Subnet에 연결하는데, 이 Routing Table은 NAT Gateway와 연결되어야 한다. 이렇게 해서 Private Subnet의 EC2 인스턴스가 외부 인터넷을 사용하기 위해 Public Subnet에 위치하고 있는 NAT Gateway를 사용하는 실습을 진행한다.

### 사전 준비물

AWS Free-Tier 계정

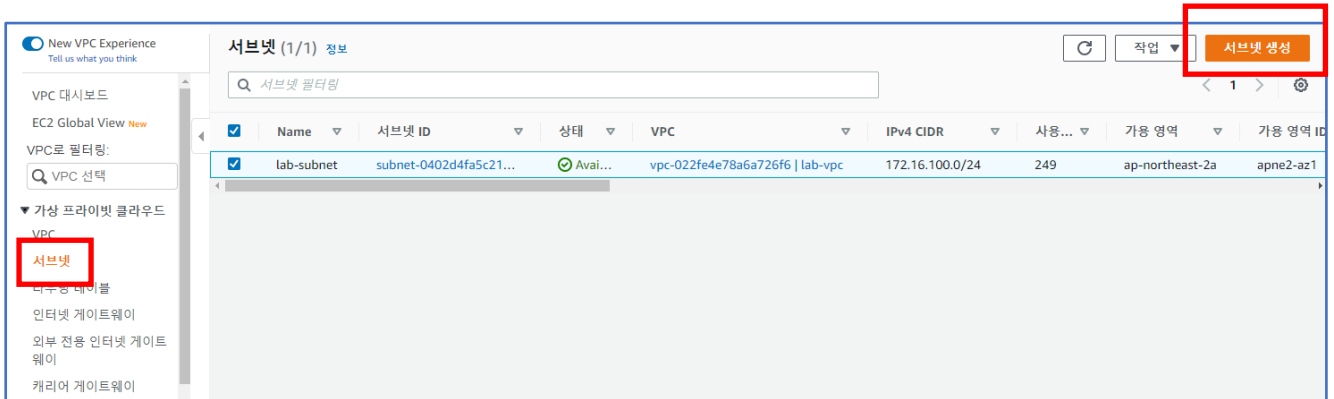
lab-vpc, lab-subnet, lab-eni, lab-ig

lab-rt, lab-nacl, lab-sg

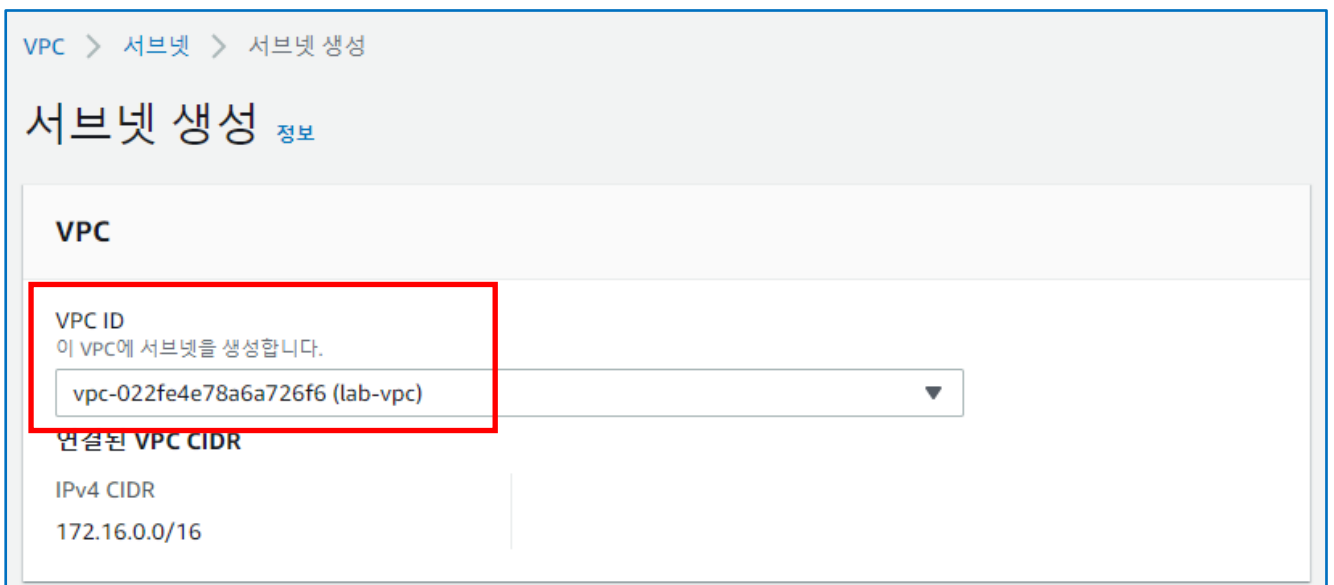


# Allocate EIP & Public IP

1. [서비스] > [VPC] > [가상 프라이빗 클라우드] > [서브넷]을 클릭하여 [서브넷] 페이지로 이동한다. 현재 앞의 Lab에서 생성한 **lab-subnet**이 보인다. 새로 **Private Subnet**을 생성하기 위해 우측 상단의 [서브넷 생성] 버튼을 클릭한다.



2. [서브넷 생성] 페이지에서 [VPC]는 **lab-vpc**로 설정한다.



3. [서브넷 설정] 페이지에서 다음과 같이 각각의 값을 설정한 후, [서브넷 생성] 버튼을 클릭한다. 이번 프라이빗 서브넷은 앞 Lab2에서 생성한 lab-subnet과 달리 다른 가용영역에 설치하기로 한다.

A. [서브넷 이름] : lab-private-subnet

B. [가용 영역] : ap-northeast-2c

C. [IPv4 CIDR] : 172.16.200.0/24

D. [키] : Name

E. [값] : lab-private-subnet

### 서브넷 설정

서브넷의 CIDR 블록 및 가용 영역을 지정합니다.

1/1개 서브넷

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

이름은 최대 256자까지 입력할 수 있습니다.

가용 영역 정보  
서브넷이 상주할 영역을 선택합니다. 선택하지 않으면 Amazon이 자동으로 선택합니다.

아시아 태평양 (서울) / ap-northeast-2c

IPv4 CIDR 블록 정보

▼ 태그 - 선택 사항

키	값 - 선택 사항	
<input type="text" value="Name"/>	<input type="text" value="lab-private-subnet"/>	<input type="button" value="제거"/>

49줄(줄) 태그 개 더 추가할 수 있습니다.

취소

4. 프라이빗 서브넷이 잘 생성 되었다.

서브넷 1개를 성공적으로 생성하였습니다. subnet-04da304b96cdf9e6

서브넷 (2) 정보

서브넷 필터링

< 1 >

	Name	서브넷 ID	상태	VPC	IPv4 CIDR	사용...	가용 영역	가용 영역
<input type="checkbox"/>	lab-private-subnet	subnet-04da304b96cdf9e6	Available	vpc-022fe4e78a6a726f6   lab-vpc	172.16.200.0/24	251	ap-northeast-2c	apne2-a
<input type="checkbox"/>	lab-subnet	subnet-0402d4fa...	Available	vpc-022fe4e78a6a726f6   lab-vpc	172.16.100.0/24	249	ap-northeast-2a	apne2-a

5. 방금 생성한 프라이빗 서브넷의 상세 페이지로 가보면 이 서브넷은 아직 기본 라우팅 테이블을 사용하고 있음을 알 수 있다.

VPC > 서브넷 > subnet-04da304b96cdf9e6

subnet-04da304b96cdf9e6 / lab-private-subnet

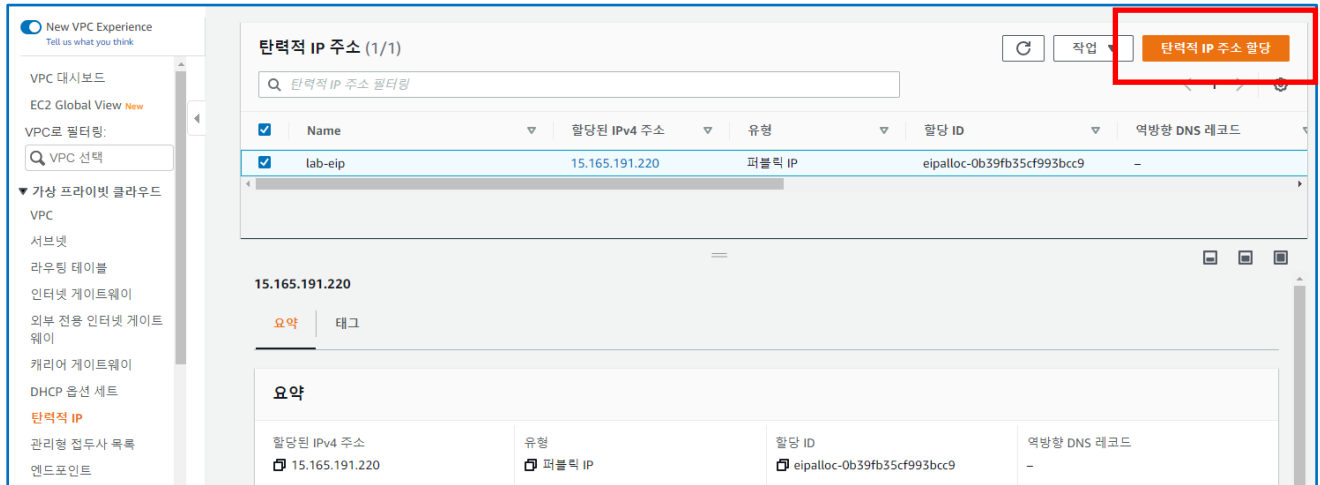
작업

세부 정보

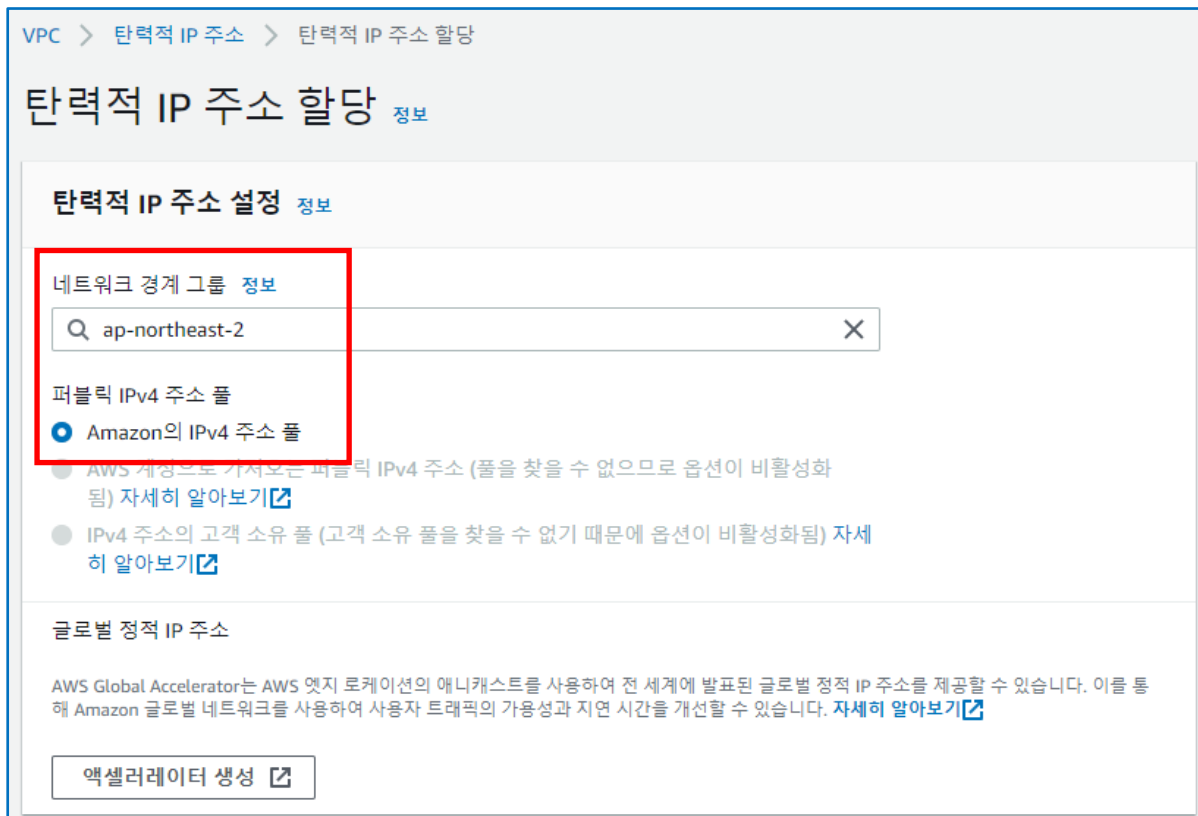
서브넷 ID subnet-04da304b96cdf9e6	서브넷 ARN arn:aws:ec2:ap-northeast-2:789534828835:subnet/subnet-04da304b96cdf9e6	상태 Available	IPv4 CIDR 172.16.200.0/24
사용 가능한 IPv4 주소 251	IPv6 CIDR -	가용 영역 ap-northeast-2c	가용 영역 ID apne2-az3
네트워크 경계 그룹 ap-northeast-2	VPC vpc-022fe4e78a6a726f6   lab-vpc	라우팅 테이블 rtb-06d685e6a373527a7	네트워크 ACL acl-056fec0dcb376282d
기본 서브넷 아니요	퍼블릭 IPv4 주소 자동 할당 아니요	IPv6 주소 자동 할당 아니요	고객 소유 IPv4 주소 자동 할당 아니요
고객 소유 IPv4 풀 -	Outpost ID -	IPv4 CIDR 예약 -	IPv6 CIDR 예약 -
IPv6 전용 아니요	호스트 이름 유형 IP 이름	리소스 이름 DNS A 레코드 비활성화됨	리소스 이름 DNS AAAA 레코드 비활성화됨
DNS64 비활성화됨	소유자 789534828835		

# NAT Gateway 생성하기

1. NAT 게이트웨이를 생성하기 전에 탄력적 IP를 생성해야 한다. [서비스] > [VPC] > [가상 프라이빗 클라우드] > [탄력적 IP]를 클릭하여 해당 페이지로 이동한다. 앞 Lab에서 생성한 lab-eip가 확인된다. 새로 생성하기 위해 우측 상단의 [탄력적 IP 주소 할당]을 클릭한다.



2. [탄력적 IP 주소 할당] 페이지에서 기본 값 그대로 사용하기로 한다. 페이지를 스크롤다운한다.



3. [태그] 섹션에서 [새로운 태그 추가]를 클릭하여, [키]를 Name, [값]을 lab-private-eip로 입력한 후, [할당] 버튼을 클릭한다. 탄력적 IP가 private이 되지 않지만 이름을 구별하기 위해 private 단어를 넣었다.

**태그 - 선택 사항**

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 값(선택 사항)으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키

값 - 선택 사항

Q Name X

Q lab-private-eip X

제거

새로운 태그 추가

최대 49개의 태그를 더 추가할 수 있습니다.

취소 할당

4. 새로 탄력적 IP 주소가 할당되었다.

탄력적 IP 주소가 할당되었습니다.  
탄력적 IP 주소 3.39.111.172 / lab-private-eip

이 탄력적 IP 주소 연결 X

탄력적 IP 주소 (2)

Q 탄력적 IP 주소 필터링

<input type="checkbox"/>	Name	할당된 IPv4 주소	유형	할당 ID	역방향 DNS 레코드
<input type="checkbox"/>	lab-eip	15.165.191.220	퍼블릭 IP	eipalloc-0b39fb35cf993bcc9	-
<input type="checkbox"/>	lab-private-eip	3.39.111.172	퍼블릭 IP	eipalloc-01addb21d216e56a0	-

5. 프라이빗 서브넷에서 외부 인터넷 구간 통신을 하려면 NAT 게이트웨이를 생성하고 VPC와 연결해야 한다. [서비스] > [VPC] > [가상 프라이빗 클라우드] > [NAT 게이트웨이] 를 클릭하여 NAT 게이트웨이 페이지로 이동한다. 우측 상단의 [NAT 게이트웨이 생성]을 클릭한다.

New VPC Experience  
Tell us what you think

VPC 대시보드  
EC2 Global View New

VPC로 필터링:  
Q VPC 선택

가상 프라이빗 클라우드

- VPC
- 서브넷
- 라우팅 테이블
- 인터넷 게이트웨이
- 외부 전용 인터넷 게이트웨이
- 캐리어 게이트웨이
- DHCP 옵션 세트
- 탄력적 IP
- 관리형 접두사 목록
- 엔드포인트
- NAT 게이트웨이**
- 하이퍼 콘솔

NAT 게이트웨이 정보

Q NAT 게이트웨이 필터링

Name	NAT 게이트웨이 ID	연결 유형	상태	상태 메시지	탄력적 IP 주소	프라이빗 IP 주소
------	--------------	-------	----	--------	-----------	------------

NAT 게이트웨이 선택

NAT 게이트웨이 생성

6. **[NAT 게이트웨이 생성]**페이지에서 다음의 각각의 값을 설정한다. 여기서 중요한 것은 지금 생성하는 **NAT 게이트웨이**의 위치는 **퍼블릭 서브넷**에 위치해야 한다는 것이다. 이렇게 하면 **프라이빗 서브넷**에서 **퍼블릭 서브넷**에 속해있는 **NAT 게이트웨이**를 통해 인터넷으로 나갈 수 있기 때문이다.

- A. **[이름]** : lab-nat
- B. **[서브넷]** : lab-subnet
- C. **[연결 유형]** : 퍼블릭
- D. **[탄력적 IP 할당 ID]** : lab-private-eip

VPC > NAT 게이트웨이 > NAT 게이트웨이 생성

## NAT 게이트웨이 생성 정보

프라이빗 서브넷의 인스턴스가 다른 VPC, 온프레미스 네트워크 또는 인터넷의 서비스에 연결하는 데 사용할 수 있는 가용성이 뛰어난 관리형 NAT(Network Address Translation) 서비스입니다.

### NAT 게이트웨이 설정

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

lab-nat

이름은 최대 256자까지 입력할 수 있습니다.

**서브넷**  
NAT 게이트웨이를 생성할 서브넷을 선택합니다.

subnet-0402d4fa5c211af22 (lab-subnet)

**연결 유형**  
NAT 게이트웨이에 대한 연결 유형을 선택합니다.

☒ 퍼블릭

☐ 프라이빗

**탄력적 IP 할당 ID 정보**  
NAT 게이트웨이에 탄력적 IP 주소를 할당합니다.

eipalloc-01addb21d216e56a0 (lab-private-eip)

탄력적 IP 할당

7. **[태그]** 섹션에서 자동으로 설정된 값을 확인하고 **[NAT 게이트웨이 생성]** 버튼을 클릭한다.

### 태그

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

키

Q Name X

값 - 선택 사항

Q lab-nat X

제거

새 태그 추가

49줄(줄) 태그 개 더 추가할 수 있습니다.

취소

**NAT 게이트웨이 생성**

8. 성공적으로 **NAT 게이트웨이**가 생성되었다.

VPC > NAT 게이트웨이 > nat-0eb75a96fa15a6f52

nat-0eb75a96fa15a6f52 / lab-nat

삭제

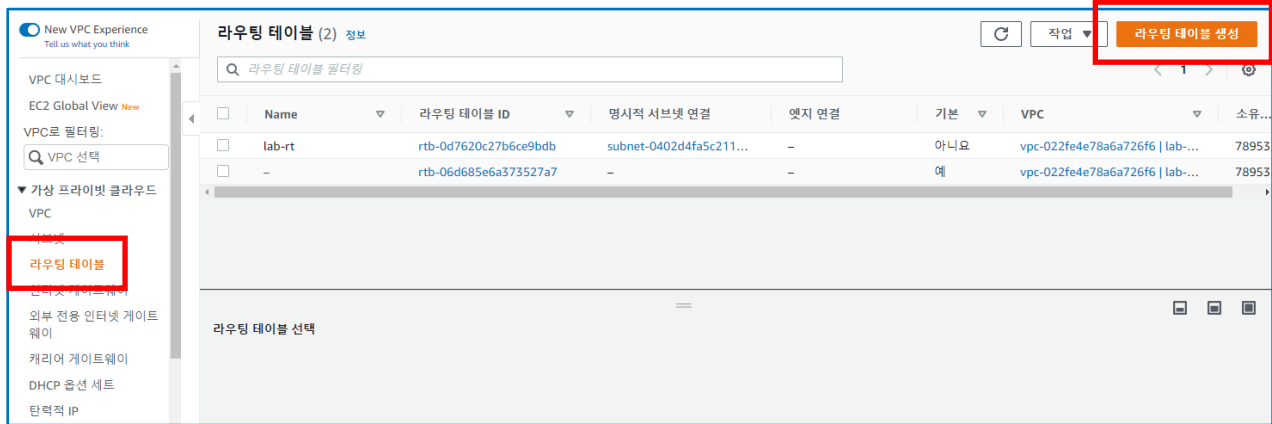
세부 정보 정보

<div>NAT 게이트웨이 ID</div> <div> nat-0eb75a96fa15a6f52</div>	<div>연결 유형</div> <div>Public</div>	<div>상태</div> <div> Available</div>	<div>상태 메시지 정보</div> <div>-</div>
<div>NAT gateway ARN</div> <div> arn:aws:ec2:ap-northeast-2:789534828835:natgateway/nat-0eb75a96fa15a6f52</div>	<div>탄력적 IP 주소</div> <div>3.39.111.172</div>	<div>프라이빗 IP 주소</div> <div>172.16.100.158</div>	<div>네트워크 인터페이스 ID</div> <div>eni-0e124116f77054fd7 </div>
<div>VPC</div> <div>vpc-022fe4e78a6a726f6 / lab-vpc</div>	<div>서브넷</div> <div>subnet-0402d4fa5c211af22 / lab-subnet</div>	<div>생성됨</div> <div> 2022년 4월 29일 금요일 17시 20분 1초 GMT+9</div>	<div>삭제됨</div> <div>-</div>



## Private Routing Table 생성 및 Subnet 연결

1. 기존에 생성한 lab-vpc에 프라이빗 라우팅 테이블을 생성하기 위해 [서비스] > [VPC] > [가상 프라이빗 클라우드] > [라우팅 테이블]을 클릭하여 라우팅 테이블 페이지로 이동한다. 우측 상단에 있는 [라우팅 테이블 생성] 버튼을 클릭한다.



2. [라우팅 테이블 생성] 페이지에서 다음과 같이 각각의 값을 설정한 후 [라우팅 테이블 생성] 버튼을 클릭한다.

- A. [이름] : lab-private-rt
- B. [VPC] : lab-vpc
- C. [태그] : Name/lab-private-rt

VPC > 라우팅 테이블 > 라우팅 테이블 생성

### 라우팅 테이블 생성 정보

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

#### 라우팅 테이블 설정

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

lab-private-rt

VPC  
이 라우팅 테이블에 대해 사용할 VPC입니다.

vpc-022fe4e78a6a726f6 (lab-vpc)

#### 태그

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

키

Q Name X

값 - 선택 사항

Q lab-private-rt X 제거

새 태그 추가

49줄(들) 태그.개 더 추가할 수 있습니다.

취소 **라우팅 테이블 생성**

3. 라우팅 테이블 생성 후, 연결된 서브넷이 없기 때문에 서브넷에 연결하기 위해 [서브넷 연결] 탭의 [서브넷 연결 편집]을 클릭한다.

VPC > 라우팅 테이블 > rtb-0f2def16868e2b904

## rtb-0f2def16868e2b904 / lab-private-rt

작업 ▼

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

### 세부 정보 정보

라우팅 테이블 ID rtb-0f2def16868e2b904	기본 아니요	명시적 서브넷 연결 -	엣지 연결 -
VPC vpc-022fe4e78a6a726f6   lab-vpc	소유자 ID 789534828835		

라우팅 **서브넷 연결** 엣지 연결 라우팅 전파 태그

### 명시적 서브넷 연결 (0)

서브넷 연결 검색

서브넷 연결 편집

서브넷 ID	IPv4 CIDR	IPv6 CIDR
서브넷 연결 없음 서브넷 연결이 없습니다.		

4. [서브넷 연결 편집] 페이지에서 이용 가능한 서브넷 목록 중 방금 생성한 lab-private-subnet을 체크하고 [연결 저장]을 클릭한다.

VPC > 라우팅 테이블 > rtb-0f2def16868e2b904 > 서브넷 연결 편집

## 서브넷 연결 편집

이 라우팅 테이블과 연결된 서브넷을 변경합니다.

### 이용 가능한 서브넷 (1/2)

서브넷 연결 필터링

	이름	서브넷 ID	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/>	lab-private-subnet	subnet-04da304b96cdf9e6	172.16.200.0/24	-
<input type="checkbox"/>	lab-subnet	subnet-0402d4fa5c211af22	172.16.100.0/24	-

### 선택한 서브넷

subnet-04da304b96cdf9e6 / lab-private-subnet ✕

취소 **연결 저장**

5. 이렇게 해서 프라이빗 서브넷은 기본 라우팅 테이블이 아닌 프라이빗 라우팅 테이블과 연결이 되었다. 하지만 라우팅 정보를 보면 아직 외부 인터넷 구간 통신을 위한 라우팅 경로가 없는 것을 확인할 수 있다. [라우팅 편집] 버튼을 클릭한다.

rtb-Of2def16868e2b904 / lab-private-rt

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

### 세부 정보 정보

라우팅 테이블 ID rtb-Of2def16868e2b904	기본 아니요	명시적 서브넷 연결 subnet-04da304b96cfe9e6 / lab-private-subnet	옛지 연결 -
VPC vpc-022fe4e78a6a726f6   lab-vpc	소유자 ID 789534828835		

**라우팅** | 서브넷 연결 | 옛지 연결 | 라우팅 전파 | 태그

### 라우팅 (1)

라우팅 필터링

대상	대상	상태	전파됨
172.16.0.0/16	local	활성	아니요

라우팅 편집

6. [라우팅 편집] 페이지에서 [라우팅 추가]를 클릭하여 다음과 같이 값을 설정한 후, [변경 사항 저장]을 클릭한다.

A. [대상] : 0.0.0.0/0

B. [대상] : NAT 게이트웨이 > lab-nat

VPC > 라우팅 테이블 > rtb-Of2def16868e2b904 > 라우팅 편집

### 라우팅 편집

대상	대상	상태	전파됨
172.16.0.0/16	local	활성	아니요
0.0.0.0/0	nat-0eb75a96fa15a6f52	-	아니요

라우팅 추가

취소 | 미리 보기 | **변경 사항 저장**

7. 프라이빗 라우팅 테이블에 NAT 게이트웨이를 통해 인터넷과 통신할 수 있는 라우팅 경로를 추가하였다.

세부 정보 정보

라우팅 테이블 ID  
rtb-0f2def16868e2b904

VPC  
vpc-022fe4e78a6a726f6 | lab-vpc

기본  
아니요

소유자 ID  
789534828835

명시적 서브넷 연결  
subnet-04da304b96cdfe9e6 / lab-private-subnet

엣지 연결  
-

라우팅

서브넷 연결

엣지 연결

라우팅 전파

태그

라우팅 (2)

라우팅 편집

라우팅 필터링

모두

< 1 >

대상	대상	상태	전파됨
172.16.0.0/16	local	확성	아니요
0.0.0.0/0	nat-0eb75a96fa15a6f52	활성	아니요

## EC2 인스턴스 생성하여 인터넷 통신 검증하기

- 다음과 같이 **al-webserver-ec2**를 생성했다. 주의할 점은 퍼블릭 IP 자동 할당은 기본값 **비활성화** 그대로 사용한다.
  - [OS] : Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volume Type, 64비트
  - [인스턴스 유형] : t2.micro
  - [네트워크 설정] : lab-vpc, lab-private-subnet
  - [사용자 데이터] : userdata.txt 참조
  - [EBS] : 없음 SSD(gp2) 30GiB
  - [태그] : Name/al-webserver-ec2
  - [보안 그룹] : 기존 보안 그룹 선택/lab-sg
  - [키 페어] : 키 페이 생성하지 않음.

고급 세부 정보

Enclave ⓘ

☐ 활성화

메타데이터 액세스 가능 ⓘ

활성화됨

메타데이터 버전 ⓘ

V1 및 V2(토큰 선택 사항)

메타데이터 토큰 응답 홉 제한 ⓘ

1

Allow tags in metadata ⓘ

활성화됨

⚠ Allow tags in metadata enabled

Any instance tags that you add will be available via instance metadata. To prevent this, under **Advanced details**, choose **Disable** from **Allow tags in metadata**.

사용자 데이터 ⓘ

☒ 텍스트로 ☐ 파일로 ☐ 입력이 이미 base64로 인코딩됨

```
echo "qwer1234"
)| passwd --stdin root
sed -i "s/^PasswordAuthentication no/PasswordAuthentication yes/g"
/etc/ssh/sshd_config
sed -i "s/^#PermitRootLogin yes/PermitRootLogin yes/g" /etc/ssh/sshd_config
service sshd restart
```

2. 아래 그림에서 보면 방금 생성한 EC2 인스턴스의 퍼블릭 IPv4 주소가 없다는 것이다. 프라이빗 IPv4 주소만 확인할 수 있다. 그렇기 때문에 방금 생성한 인스턴스에 PING 테스트나 SSH를 통한 연결을 불가능하다.

EC2 > 인스턴스 > i-010a3aa2e5e9442b8

**i-010a3aa2e5e9442b8 (al-webserver-ec2)에 대한 인스턴스 요약** 정보

less than a minute 전에 업데이트됨

인스턴스 ID i-010a3aa2e5e9442b8 (al-webserver-ec2)	퍼블릭 IPv4 주소 -	프라이빗 IPv4 주소 172.16.200.32
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv4 DNS -
호스트 이름 유형 IP 이름: ip-172-16-200-32.ap-northeast-2.compute.internal	프라이빗 IP DNS 이름(IPv4만 해당) ip-172-16-200-32.ap-northeast-2.compute.internal	프라이빗 리소스 DNS 이름 응답 IPv4(A)
인스턴스 유형 t2.micro	탄력적 IP 주소 -	자동 할당된 IP 주소 -
VPC ID vpc-022fe4e78a6a726f6 (lab-vpc)	AWS Compute Optimizer 찾기 권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다. 자세히 알아보기	IAM 역할 -
서브넷 ID subnet-04da304b96cdf9e6 (lab-private-subnet)	Auto Scaling Group name -	

3. 그래서 퍼블릭 서브넷에 속해있는 Lab6에서 생성했던 인스턴스를 SSH로 연결해서 로컬 통신을 통해 방금 생성한 프라이빗 EC2에 연결해보자. 로컬 통신은 문제 없이 PING 테스트 성공이라는 것을 확인할 수 있다.

```

1 Ubuntu Web Server x +
ubuntu@ip-172-16-100-109:~$
ubuntu@ip-172-16-100-109:~$
ubuntu@ip-172-16-100-109:~$ ping -c 4 172.16.200.32
PING 172.16.200.32 (172.16.200.32) 56(84) bytes of data.
64 bytes from 172.16.200.32: icmp_seq=1 ttl=255 time=1.18 ms
64 bytes from 172.16.200.32: icmp_seq=2 ttl=255 time=1.09 ms
64 bytes from 172.16.200.32: icmp_seq=3 ttl=255 time=1.08 ms
64 bytes from 172.16.200.32: icmp_seq=4 ttl=255 time=1.10 ms

--- 172.16.200.32 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.076/1.110/1.178/0.039 ms
ubuntu@ip-172-16-100-109:~$

```

4. 퍼블릭 서브넷에서 프라이빗 서브넷으로 SSH 연결하려면 추가작업을 해야 한다. 먼저 Lab5에서 생성했던 lab-nacl의 인바운드 규칙을 다음과 같이 설정한다.

**인바운드 규칙 편집** 정보

인바운드 규칙은 VPC에 도달할 수 있는 수신 트래픽을 제어합니다.

규칙 번호 정보	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	허용/거부 정보
100	모든 ICMP - IPv4	ICMP(1)	모두	0.0.0.0/0	허용
200	SSH(22)	TCP(6)	22	0.0.0.0/0	허용
300	HTTP(80)	TCP(6)	80	0.0.0.0/0	허용
400	모든 TCP	TCP(6)	모두	0.0.0.0/0	허용

5. 새로 lab-private-nacl을 생성하여 lab-private-subnet과 연결한 다음, 다음과 같이 인바운드 규칙을 생성한다.

인바운드 규칙 편집 정보						
인바운드 규칙은 VPC에 도달할 수 있는 수신 트래픽을 제어합니다.						
규칙 번호 정보	유형 정보	프로토콜 정보	포트 범위 정보	소스 정보	허용/거부 정보	
100	모든 ICMP - IPv4	ICMP(1)	모두	0.0.0.0/0	허용	
200	SSH(22)	TCP(6)	22	0.0.0.0/0	허용	

6. 아웃바운드 규칙은 일반적인 규칙대로 생성한다.

아웃바운드 규칙 편집 정보						
아웃바운드 규칙은 VPC에서 나갈 수 있는 발신 트래픽을 제어합니다.						
규칙 번호 정보	유형 정보	프로토콜 정보	포트 범위 정보	대상 정보	허용/거부 정보	
100	모든 ICMP - IPv4	ICMP(1)	모두	0.0.0.0/0	허용	
200	모든 TCP	TCP(6)	모두	0.0.0.0/0	허용	
300	모든 UDP	UDP(17)	모두	0.0.0.0/0	허용	

7. SSH 연결을 해보자. 인스턴스 생성시 지정했던 사용자 데이터에서 패스워드를 **qwer1234**로 설정했었다.

A. \$ ssh [root@172.16.200.32](#)(프라이빗 IP)

B. Are you sure you want to continue connecting (yes/no/[fingerprint])? **yes**

C. password : qwer1234

```
1 Ubuntu Web Server x +
ubuntu@ip-172-16-100-109:~$
ubuntu@ip-172-16-100-109:~$ ssh root@172.16.200.32
The authenticity of host '172.16.200.32 (172.16.200.32)' can't be established.
ECDSA key fingerprint is SHA256:6kD9aBrBa6Nuyco1DFzZy0RCPFMaUUB6RjBIKfkjcJE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.200.32' (ECDSA) to the list of known hosts.
root@172.16.200.32's password:

  _ | _ | _ )
  _ | ( _ /   Amazon Linux 2 AMI
  __| \__|_|_|

https://aws.amazon.com/amazon-linux-2/
12 package(s) needed for security, out of 28 available
Run "sudo yum update" to apply all updates.
[root@ip-172-16-200-32 ~]#
```

8. 로컬 통신을 통해 새로 생성한 인스턴스에 잘 연결되었다. 인터넷 연결이 되는지 확인해 보자. 아래의 그림과 같이 구글에 대한 PING 테스트가 성공적이다. 즉, **프라이빗 서브넷**에 있는 **al-webserver-ec2** 인스턴스는 외부에서는 접근할 수 없지만, **NAT 게이트웨이**를 통해 인터넷 통신이 가능함을 확인할 수 있다.

```
1 Ubuntu Web Server * +
[root@ip-172-16-200-32 ~]#
[root@ip-172-16-200-32 ~]# ping -c 4 google.com
PING google.com (172.217.175.14) 56(84) bytes of data:
64 bytes from nrt20s18-in-f14.1e100.net (172.217.175.14): icmp_seq=1 ttl=104 time=33.1 ms
64 bytes from nrt20s18-in-f14.1e100.net (172.217.175.14): icmp_seq=2 ttl=104 time=32.2 ms
64 bytes from nrt20s18-in-f14.1e100.net (172.217.175.14): icmp_seq=3 ttl=104 time=32.2 ms
64 bytes from nrt20s18-in-f14.1e100.net (172.217.175.14): icmp_seq=4 ttl=104 time=32.2 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 32.207/32.464/33.124/0.402 ms
[root@ip-172-16-200-32 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 0a:82:74:43:a7:f6 brd ff:ff:ff:ff:ff:ff
    inet 172.16.200.32/24 brd 172.16.200.255 scope global dynamic eth0
        valid_lft 3160sec preferred_lft 3160sec
    inet6 fe80::882:74ff:fe43:a7f6/64 scope link
        valid_lft forever preferred_lft forever
[root@ip-172-16-200-32 ~]#
```