AWS에서 C8000v 고가용성 구성 배포

목차

소개

이 문서에서는 Amazon Web Services 클라우드에서 Catalyst 8000v 라우터로 고가용성 환경을 설정하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 항목에 대해 미리 알고 있는 것이 좋습니다.

- AWS Console 및 해당 구성 요소에 대한 일반적인 지식
- Cisco IOS® XE 소프트웨어 이해
- HA 기능에 대한 기본 지식

사용되는 구성 요소

이 컨피그레이션 예에서는 다음 구성 요소가 필요합니다.

- 관리자 역할의 Amazon AWS 계정
- Cisco IOS® XE 17.15.3a 및 Ubuntu 22.04 LTS VM 1개를 실행하는 C8000v 디바이스 2개 같은 지역의 AMI

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

토폴로지

네트워크 요구 사항에 따라 다양한 HA 구축 시나리오가 있습니다. 이 예에서는 HA 이중화가 다음 설정으로 구성됩니다.

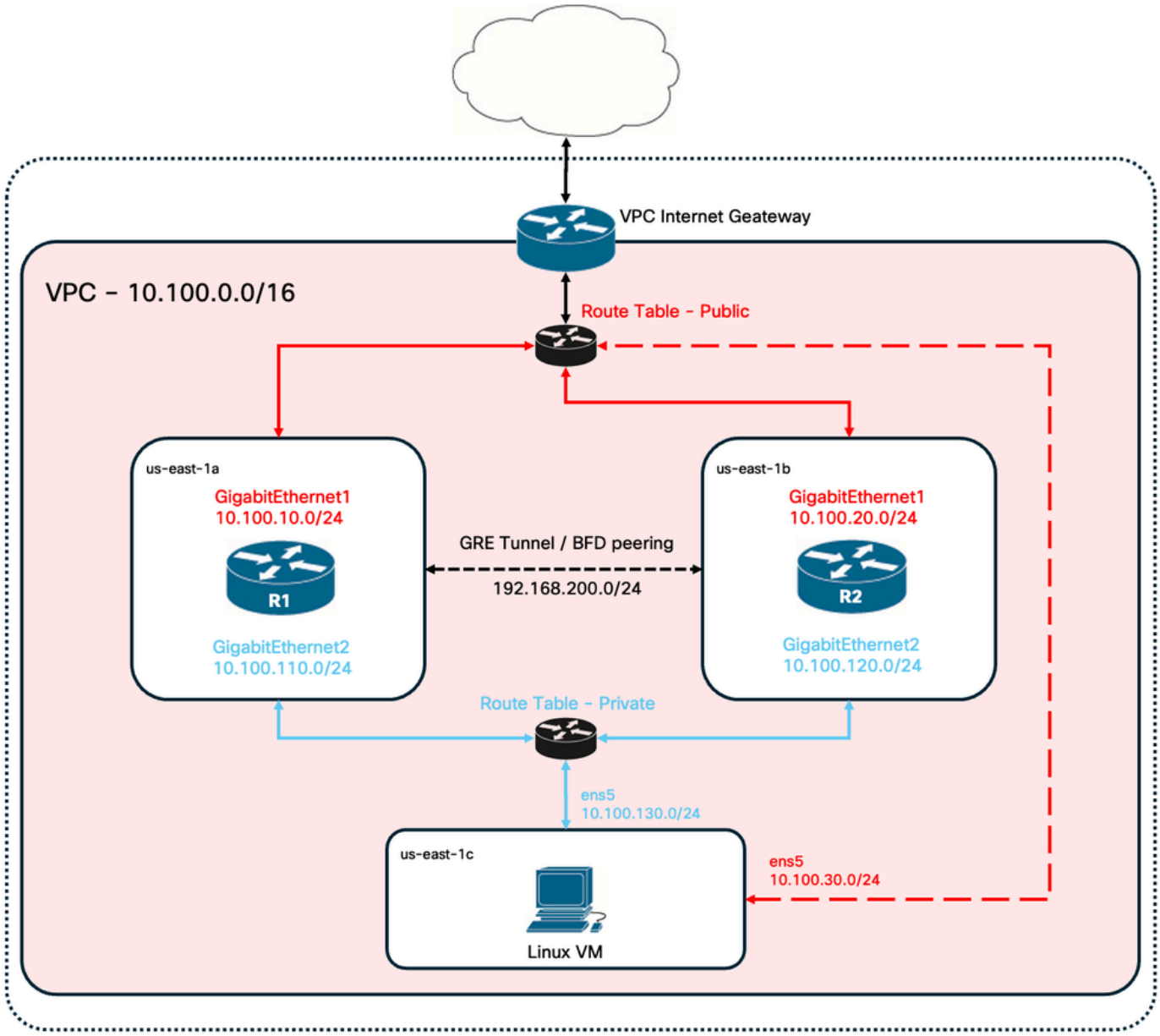
- 1x - 지역
- 1x - VPC
- 3x - 가용 영역
- 6x - 네트워크 인터페이스/서브넷(3x 공용/3x 사설)
- 2x - 경로 테이블(퍼블릭 및 프라이빗)
- 2x - C8000v 라우터(Cisco IOS® XEenali 17.15.3a)
- 1x - VM(Linux/Windows)

HA 쌍에는 서로 다른 두 가용 영역에 있는 2개의 C8000v 라우터가 있습니다. 각 가용 영역을 별도의 데이터 센터로 간주하여 추가적인 하드웨어 복원력을 제공합니다.

세 번째 영역은 사설 데이터 센터의 디바이스를 시뮬레이션하는 VM입니다. 지금은 공용 인터페이스를 통해 인터넷 액세스가 활성화되어 VM에 액세스하고 구성할 수 있습니다. 일반적으로 모든 일반 트래픽은 전용 경로 테이블을 통과해야 합니다.

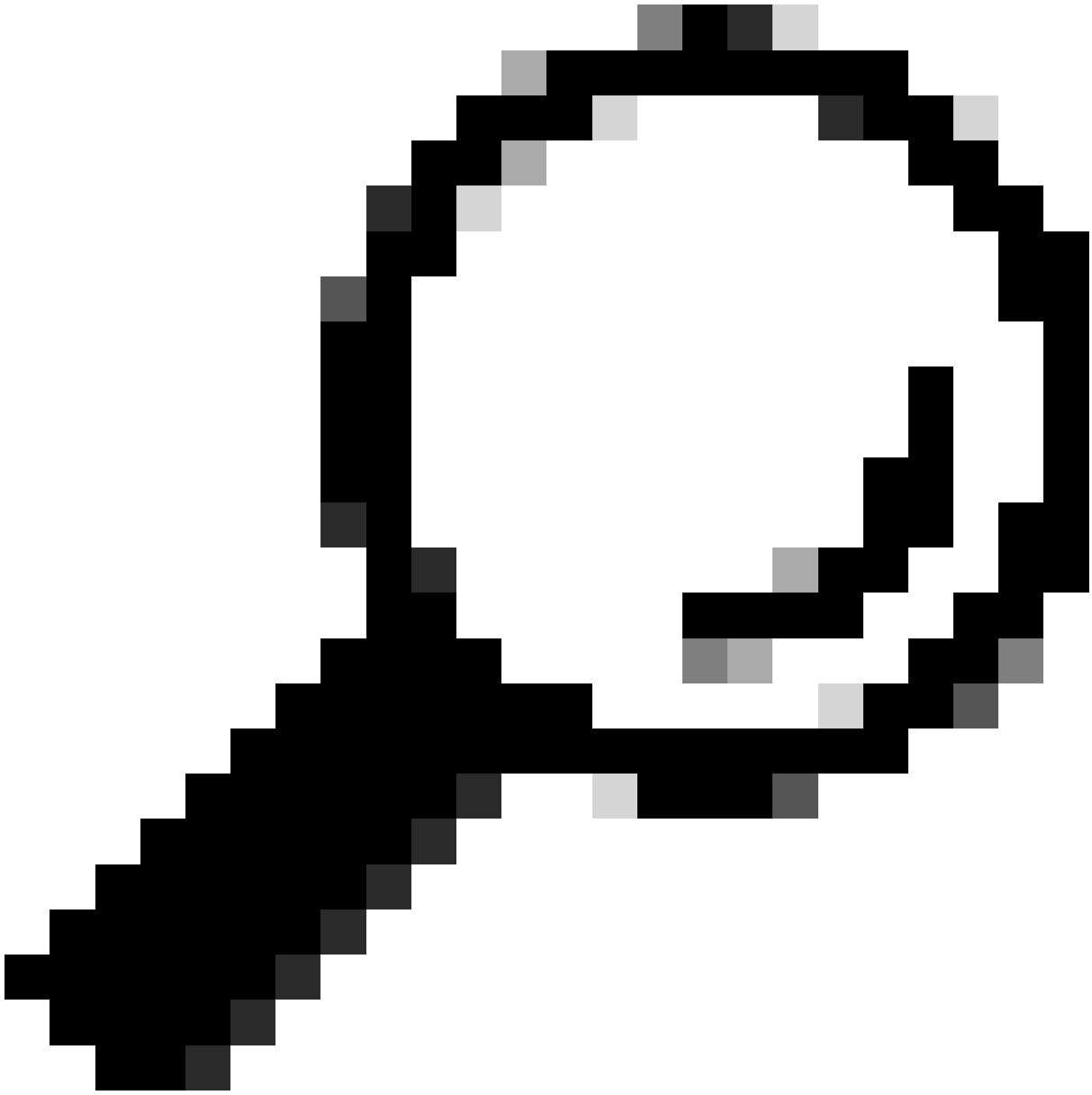
트래픽을 시뮬레이션하려면 가상 머신의 프라이빗 인터페이스에서 ping을 시작하여 프라이빗 경로 테이블을 R1을 통해 8.8.8에 도달합니다. 장애 조치가 수행되는 경우 프라이빗 경로 테이블이 R2 라우터의 프라이빗 인터페이스를 통해 트래픽을 라우팅하도록 자동으로 업데이트되었는지 확인합니다.

네트워크 다이어그램



테이블 요약

토폴로지를 요약하면, 다음은 실습의 각 구성 요소에서 가장 중요한 값을 제공하는 표입니다. 이 표에 제공된 정보는 이 실습에서만 제공됩니다.



팁: 이 표를 사용하면 가이드 전반에서 주요 변수의 개요를 쉽게 확인할 수 있습니다. 이 형식으로 정보를 수집하면 프로세스를 간소화할 수 있습니다.

디바이스	가용 영역	인터페이스	IP 주소	RTB	에니
R1	미국 동부 1a	기가비트 이더넷1	10.100.10.254	rtb-0d0e48f25c9b00635(퍼블릭)	eni-0645a881c13823696
		기가비트 이더넷2	10.100.110.254	rtb-093df10a4de426eb8(전용)	eni-070e14fbfde0d8e3b
R2	미국-동부-1b	기가비트 이더넷1	10.100.20.254	rtb-0d0e48f25c9b00635(퍼블릭)	eni-0a7817922ffbb317b
		기가비트 이더넷2	10.100.120.254	rtb-093df10a4de426eb8(전용)	eni-0239fda341b4d7e41
Linux	미국-	ENS5	10.100.30.254	rtb-0d0e48f25c9b00635(퍼	eni-0b28560781b3435b1

VM	동부-1c			블릭)	
		ens6	10.100.130.254	rtb-093df10a4de426eb8(전용)	eni-05d025e88b6355808

제한 사항

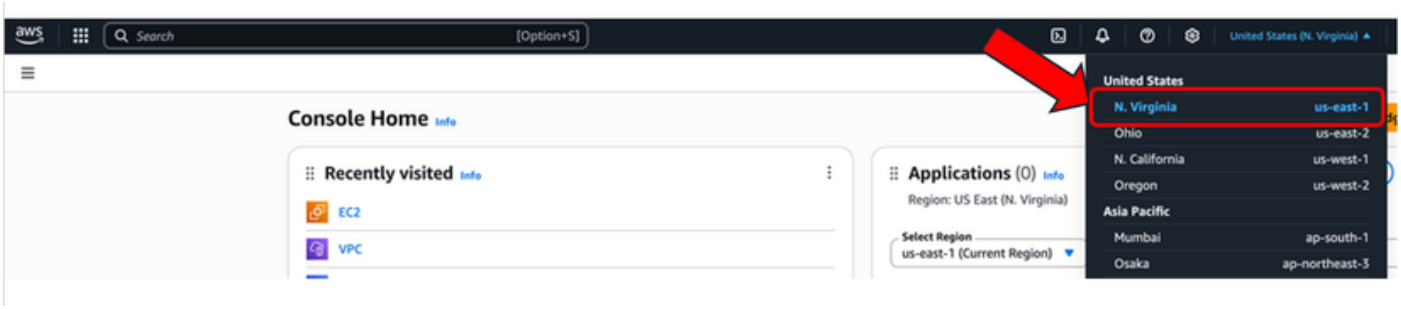
- 생성된 서브넷에서 해당 서브넷의 사용 가능한 첫 번째 주소를 사용하지 마십시오. 이러한 IP 주소는 AWS 서비스에서 내부적으로 사용됩니다.
- VRF 내에서 C8000v 디바이스의 공용 인터페이스를 구성하지 마십시오. HA가 설정되어 있으면 제대로 작동하지 않습니다.

설정

일반적인 컨피그레이션 흐름은 요청된 VM을 적절한 영역에 생성하고 각 VM의 경로 및 인터페이스와 같은 가장 구체적인 컨피그레이션으로 이동하기 위해 중점을 둡니다. 그러나 먼저 토폴로지를 이해하고 원하는 순서로 구성하는 것이 좋습니다.

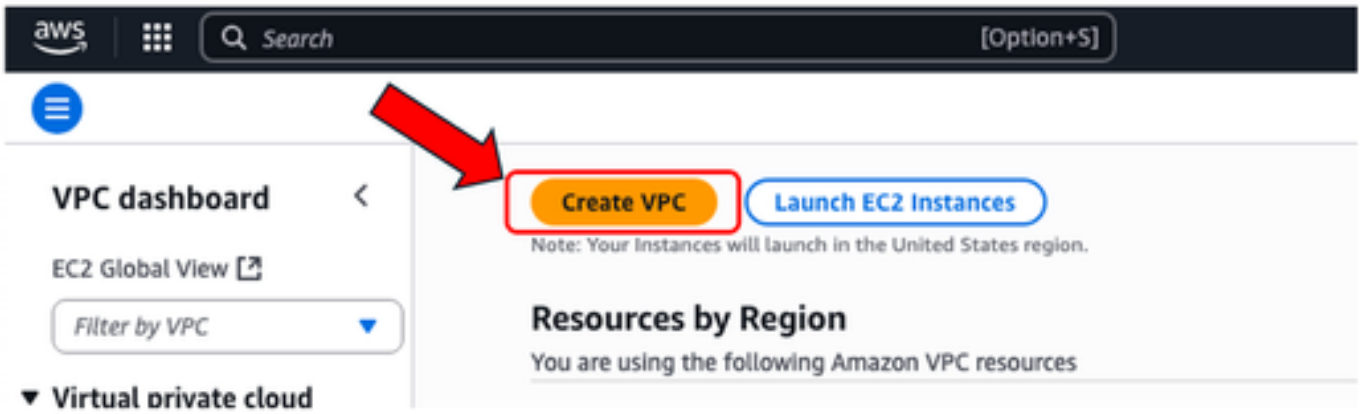
1단계. 지역 선택

이 구축 가이드에서는 미국 서부(버지니아 북부) - 미국 동부-1 지역을 VPC 지역으로 선택합니다.



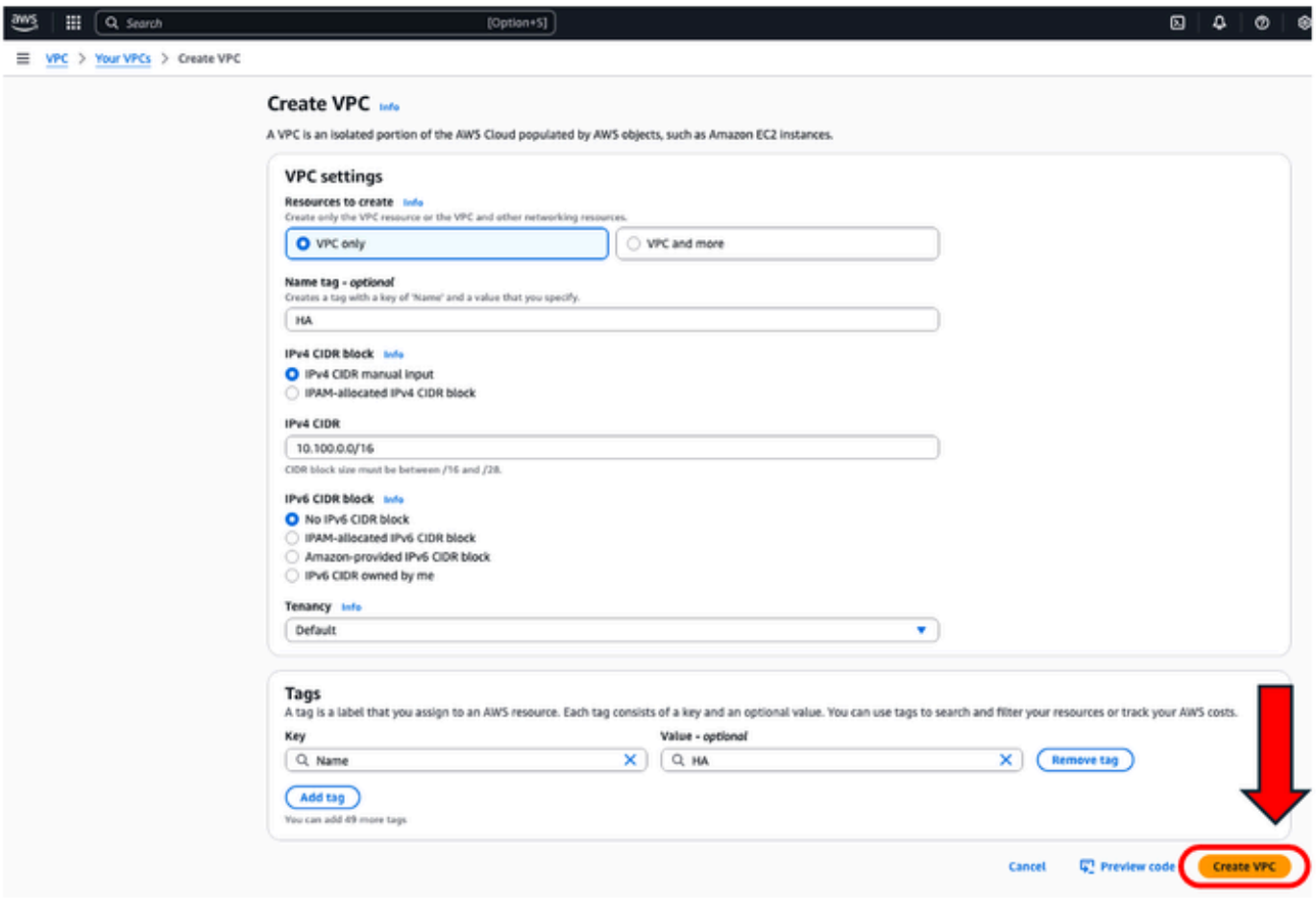
2단계. VPC 생성

AWS Console에서 VPC > VPC Dashboard > Create VPC로 이동합니다.

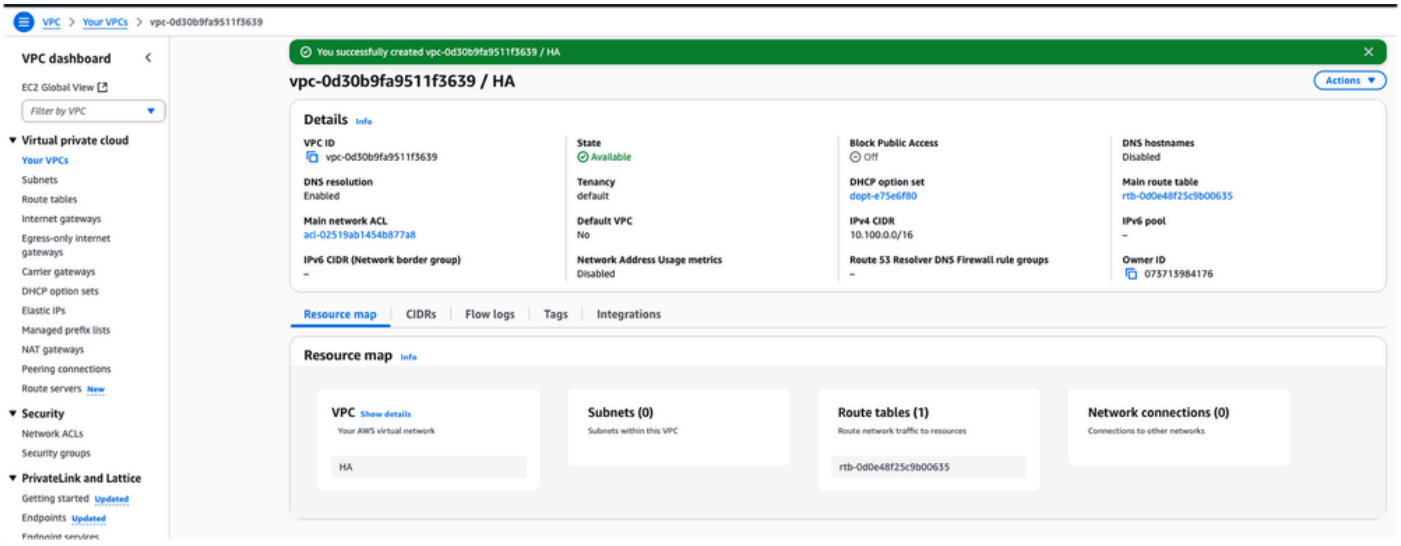


VPC를 생성할 때 VPC 전용 옵션을 선택합니다. 원하는 대로 사용할 /16 네트워크를 할당할 수 있습니다.

이 구축 가이드에서는 10.100.0.0/16 네트워크를 선택합니다.

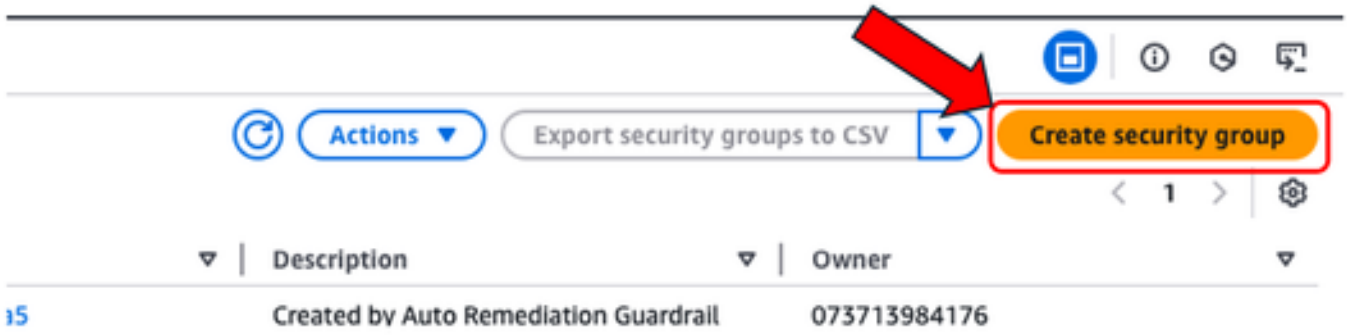


Create VPC(VPC 생성)를 클릭하면 HA 태그가 있는 VPC-0d30b9fa9511f3639가 생성됩니다.

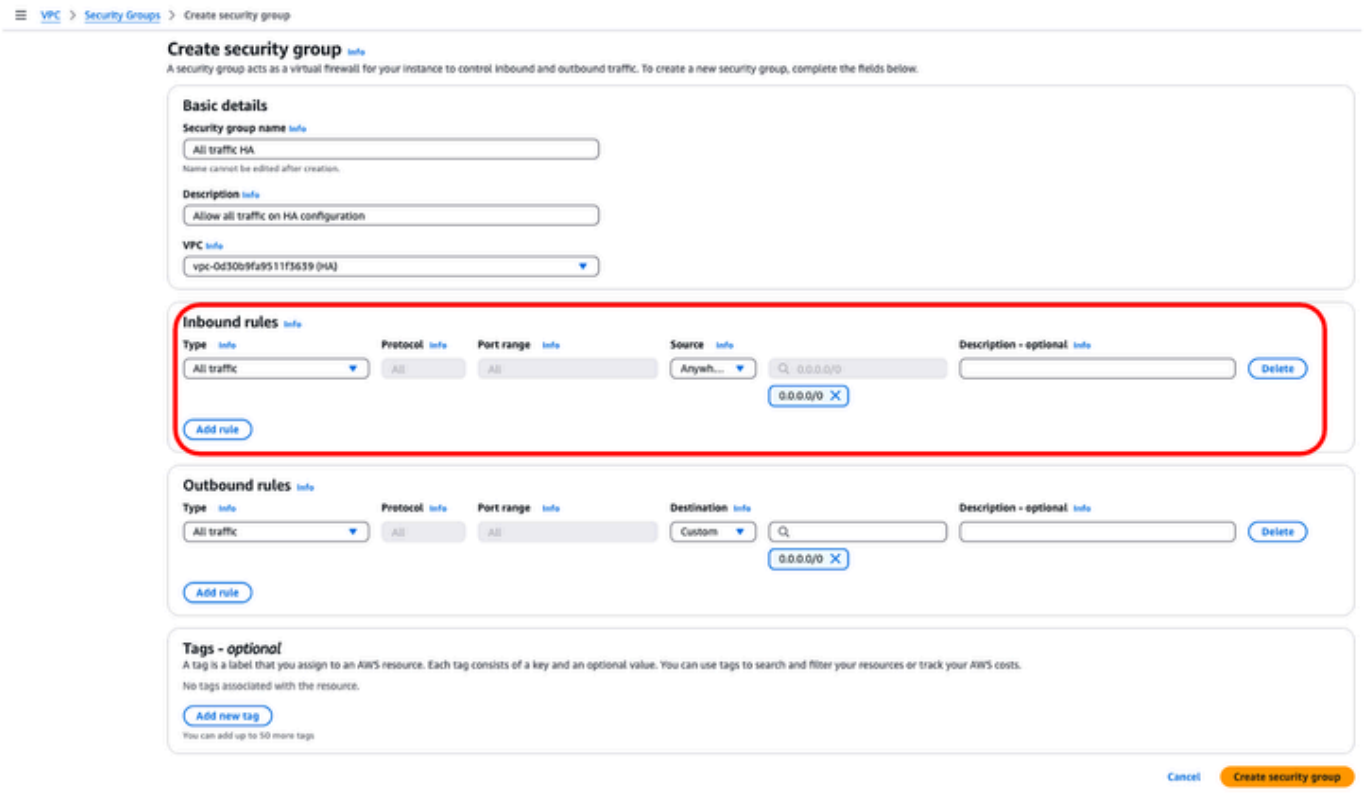


3단계. VPC에 대한 보안 그룹 생성

AWS에서 보안 그룹은 ACL과 같이 작동하여 VPC 내에서 구성된 VM에 대한 트래픽을 허용하거나 거부합니다. AWS 콘솔에서 VPC > VPC Dashboard > Security > Security Groups 섹션으로 이동하고 Create security group을 클릭합니다.



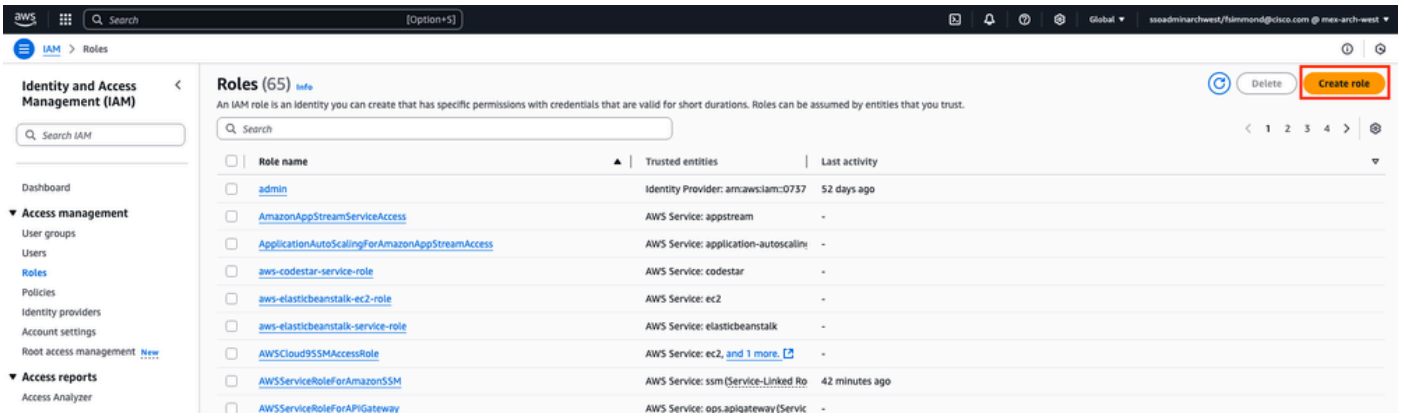
Inbound Rules(인바운드 규칙)에서 허용할 트래픽을 정의합니다. 이 예에서는 0.0.0.0/0 네트워크를 사용하여 All Traffic을 선택합니다.



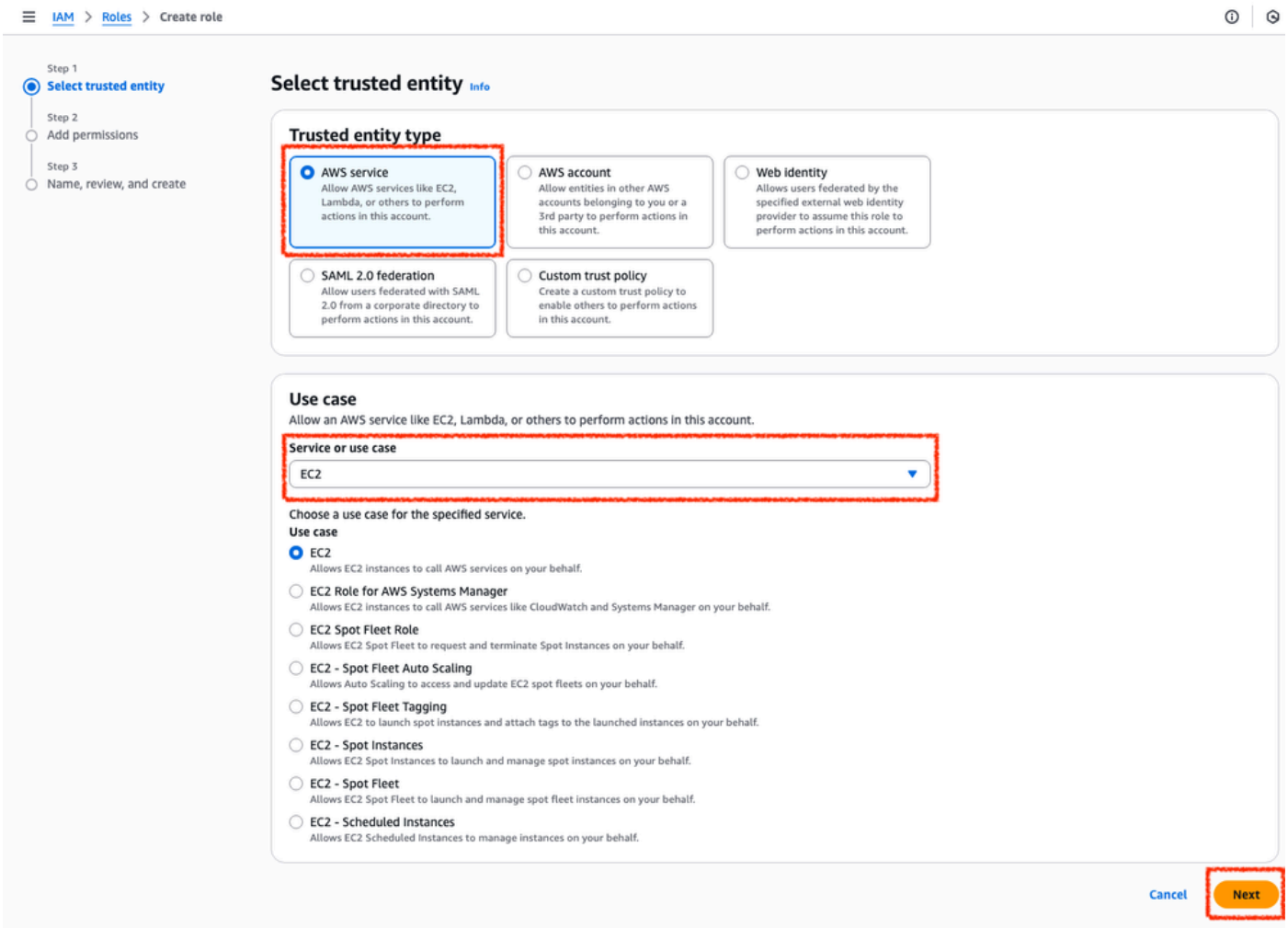
4단계. 정책으로 IAM 역할 생성 및 VPC에 연결

IAM은 Amazon API에 필요한 액세스 권한을 AMI에 부여합니다. C8000v는 AWS에서 경로 테이블을 수정하기 위해 AWS API 명령을 호출하는 프록시로 사용됩니다. 기본적으로 EC2 인스턴스는 API에 액세스할 수 없습니다. 따라서 AMI 생성 중에 적용할 새 IAM 역할을 만들어야 합니다.

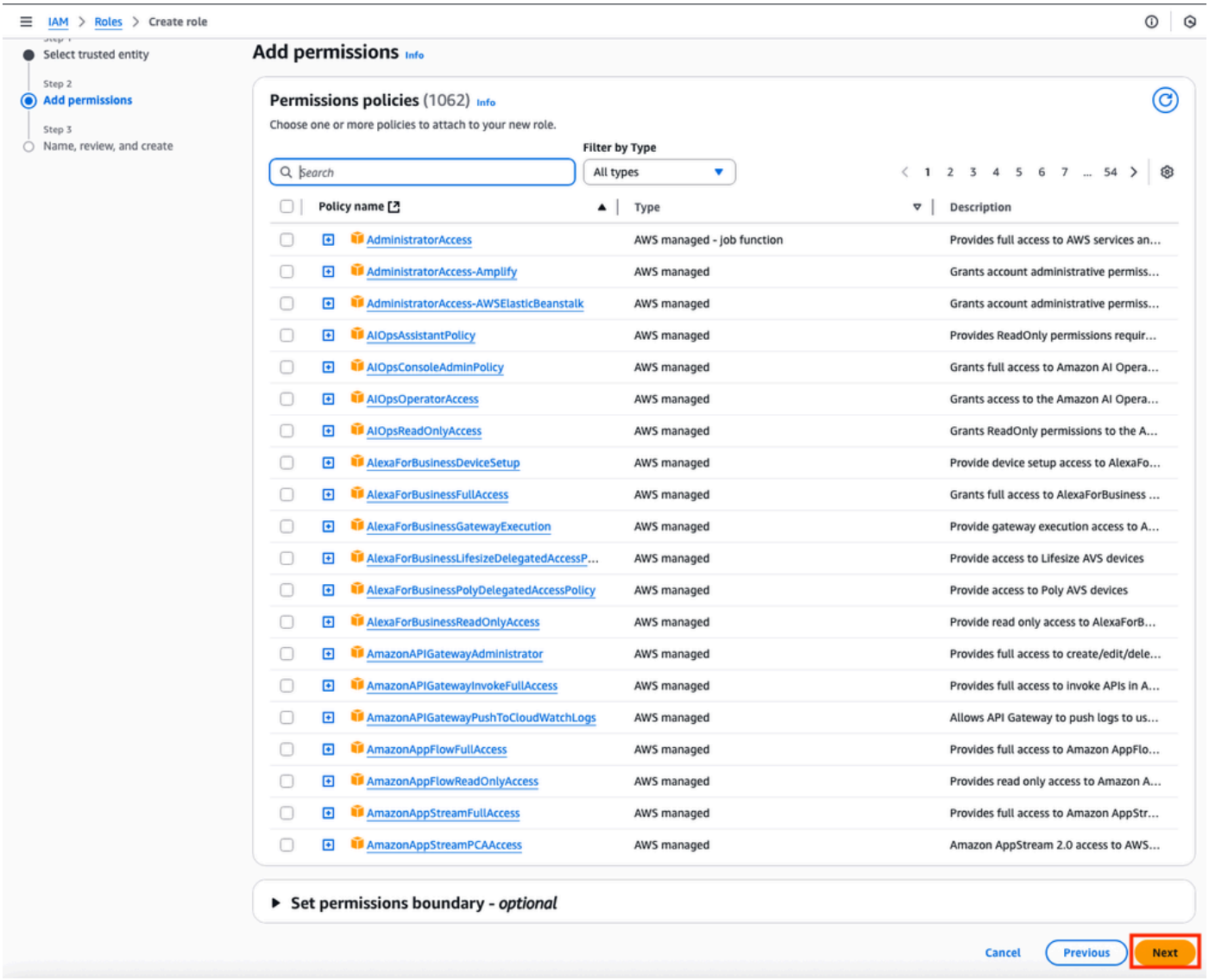
IAM 대시보드로 이동하고 Access Management(액세스 관리) > Roles(역할) > Create Role(역할 생성)로 이동합니다. 이 프로세스는 3단계로 구성됩니다.



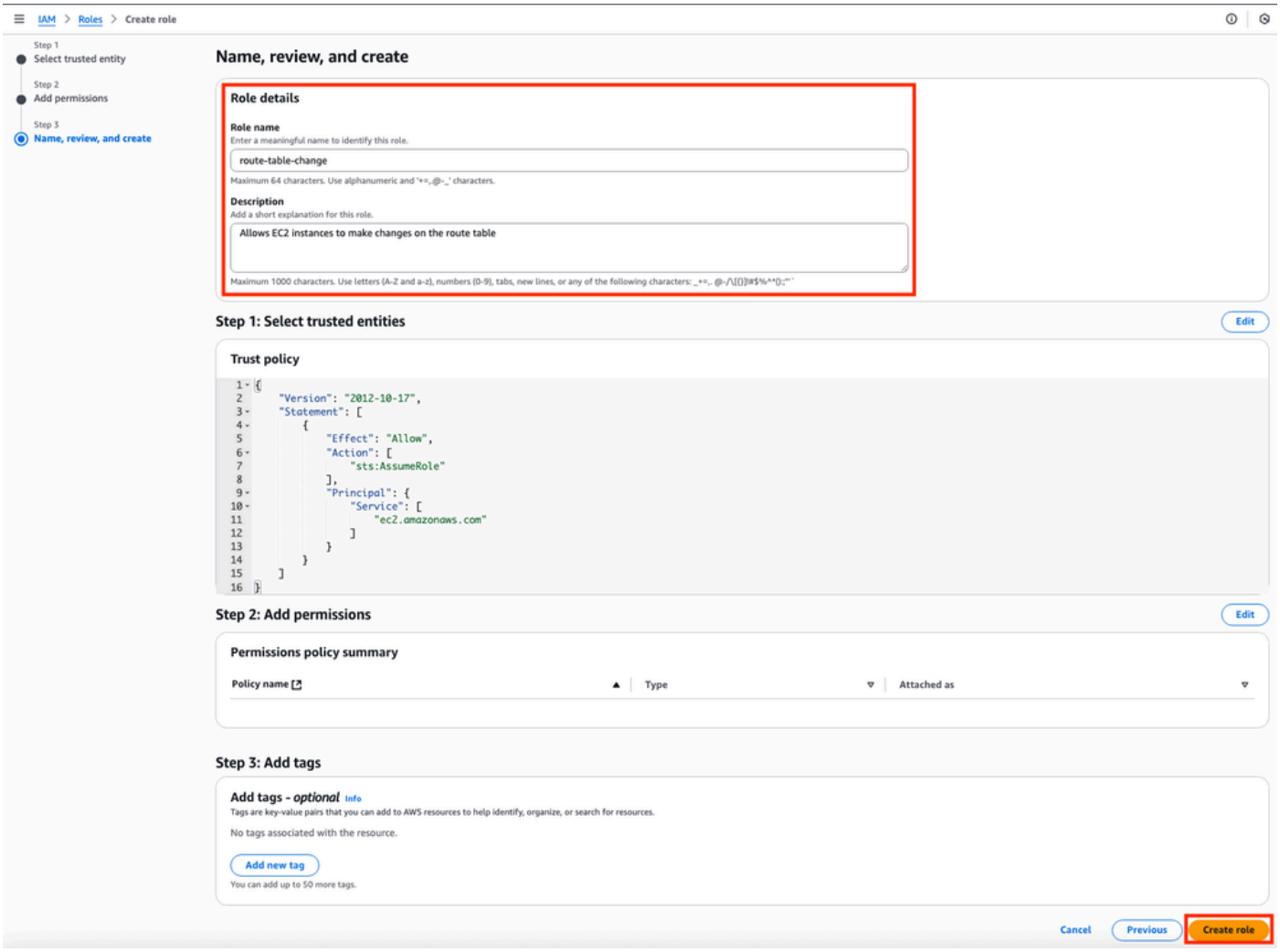
먼저, Trusted entity type(신뢰할 수 있는 엔터티 유형) 섹션에서 AWS Service(AWS 서비스) 옵션을 선택하고 이 정책에 할당된 서비스로 EC2를 선택합니다.



완료되면 다음을 클릭합니다.

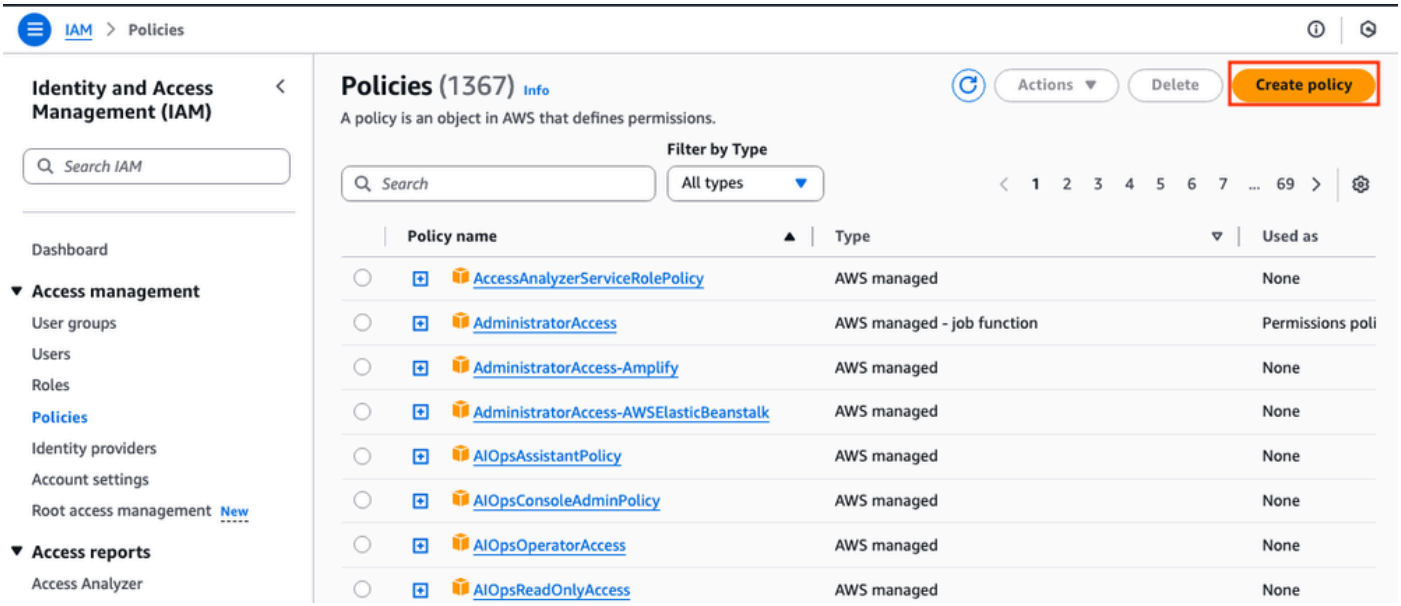


마지막으로, Role Name(역할 이름)을 설정하고 Create Role(역할 생성) 버튼을 클릭합니다.

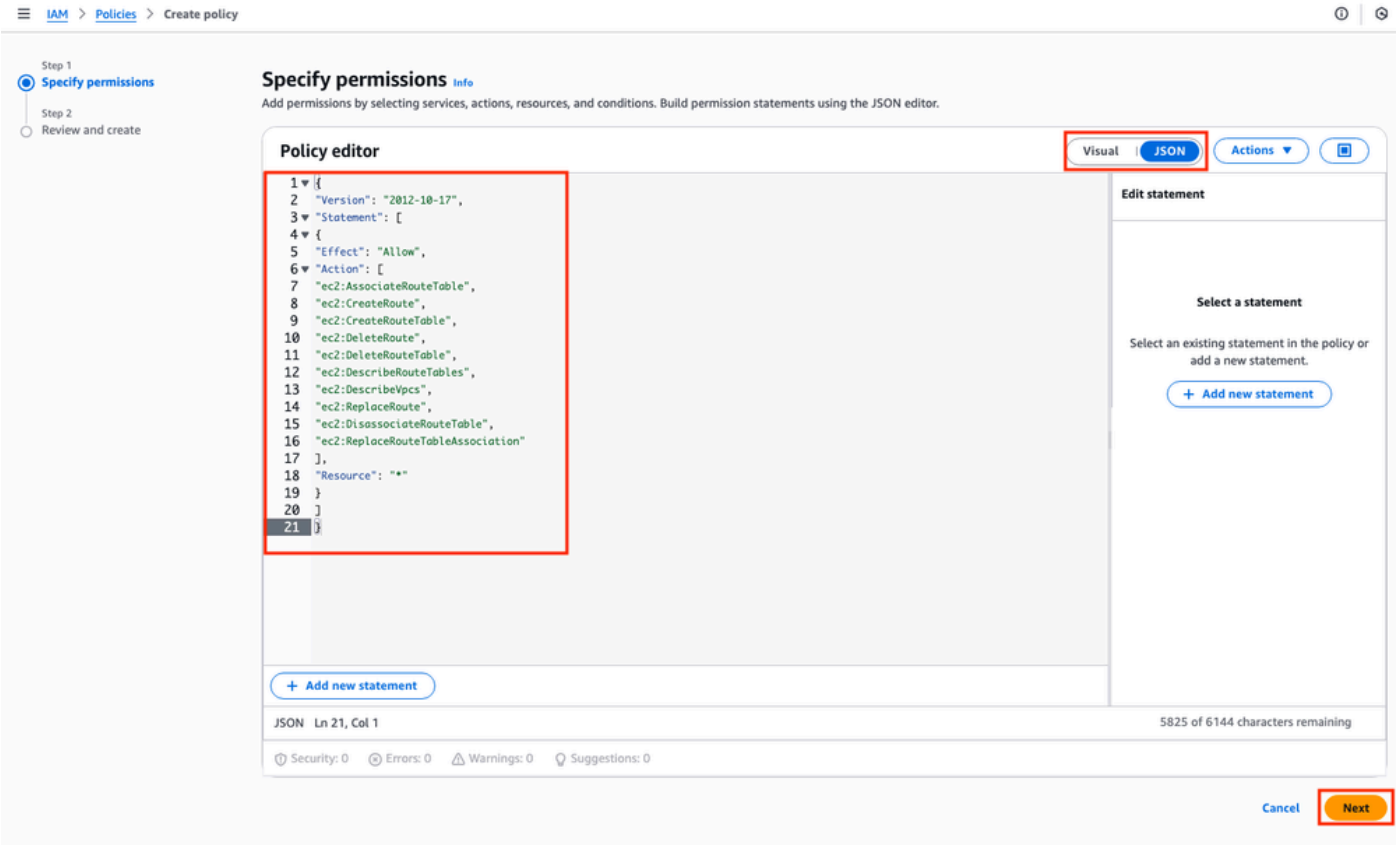


5단계. IAM 역할에 트러스트 정책 생성 및 연결

역할이 생성되면 필요할 때 AWS 라우팅 테이블을 수정하는 기술을 습득할 수 있도록 트러스트 정책을 만들어야 합니다. IAM 대시보드의 Policies 섹션으로 이동합니다. Create Policy(정책 생성) 버튼을 클릭합니다. 이 프로세스는 다음 2단계로 구성됩니다.



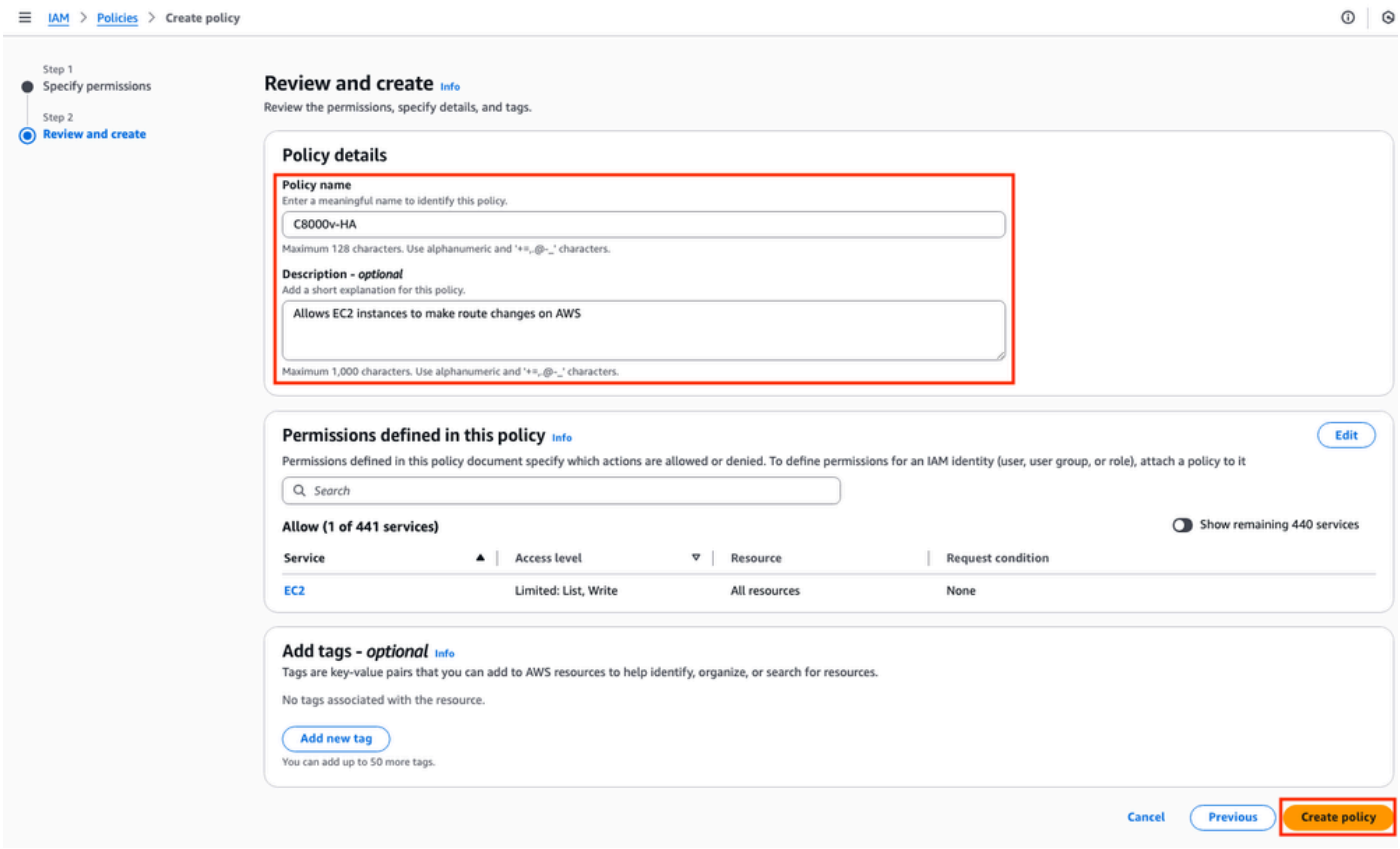
먼저, 정책 편집기가 JSON을 사용하고 있는지 확인하고 아래에 표시된 명령을 적용합니다. 구성이 완료되면 Next(다음)를 클릭합니다.



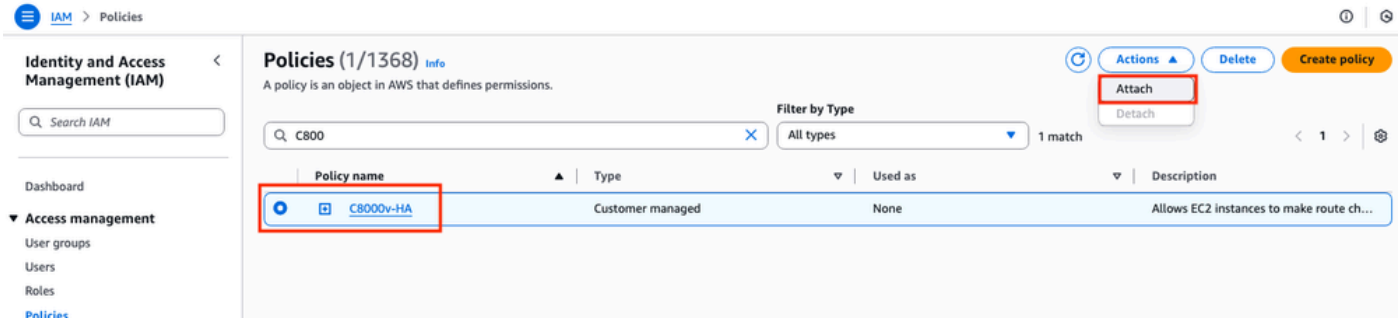
이미지에 사용되는 텍스트 코드입니다.

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"ec2:AssociateRouteTable",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
"ec2:DeleteRoute",
"ec2:DeleteRouteTable",
"ec2:DescribeRouteTables",
"ec2:DescribeVpcs",
"ec2:ReplaceRoute",
"ec2:DisassociateRouteTable",
"ec2:ReplaceRouteTableAssociation"
],
"Resource": "*"
}
]
}
```

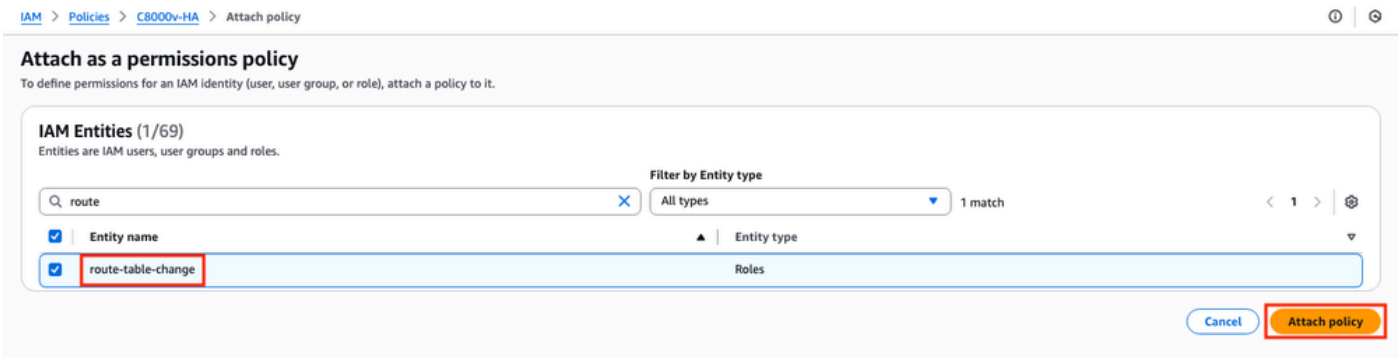
나중에 Policy Name(정책 이름)을 설정하고 Create Policy(정책 생성)를 클릭합니다.



정책이 생성되면 정책을 필터링하고 선택한 다음 Actions 드롭다운 메뉴에서 Attach 옵션을 클릭합니다.



새 창이 열립니다. IAM Entities(IAM 엔터티) 섹션에서 생성된 IAM 역할을 필터링하고 선택한 후 Attach policy(정책 연결)를 클릭합니다.

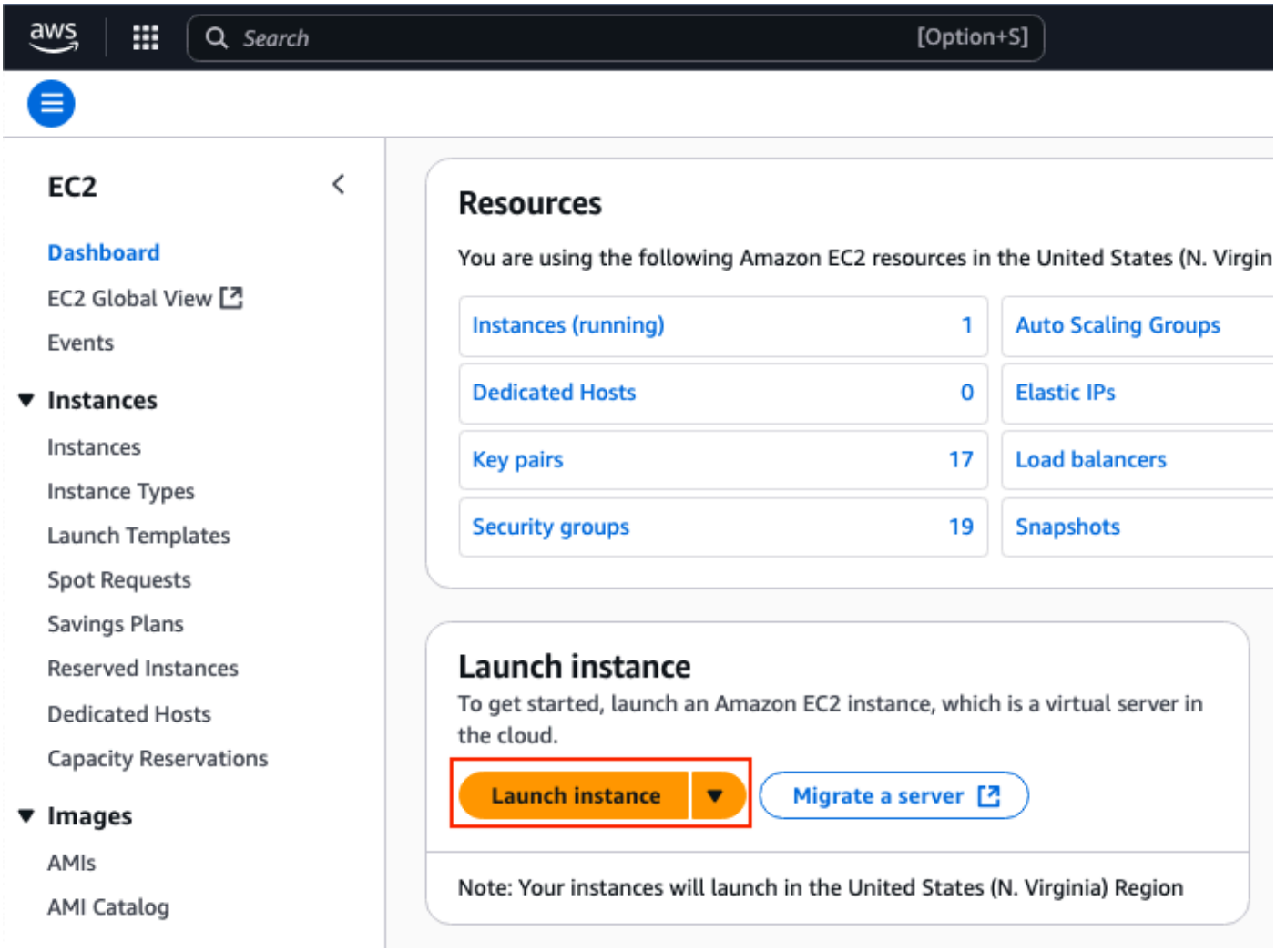


6단계 C8000v 인스턴스 구성 및 실행

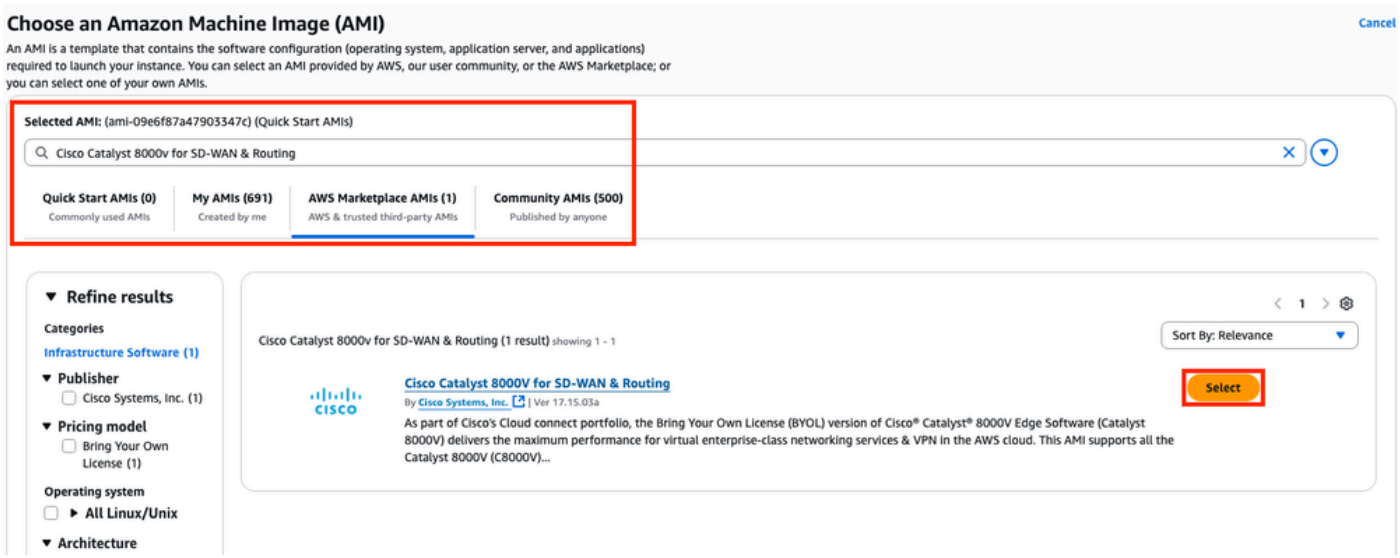
각 C8000v 라우터는 2개의 인터페이스(퍼블릭 1개, 프라이빗 1개)를 갖게 되며 자체 가용 영역에

생성됩니다.

EC2 Dashboard(EC2 대시보드)에서 Launch Instances(인스턴스 실행)를 클릭합니다.

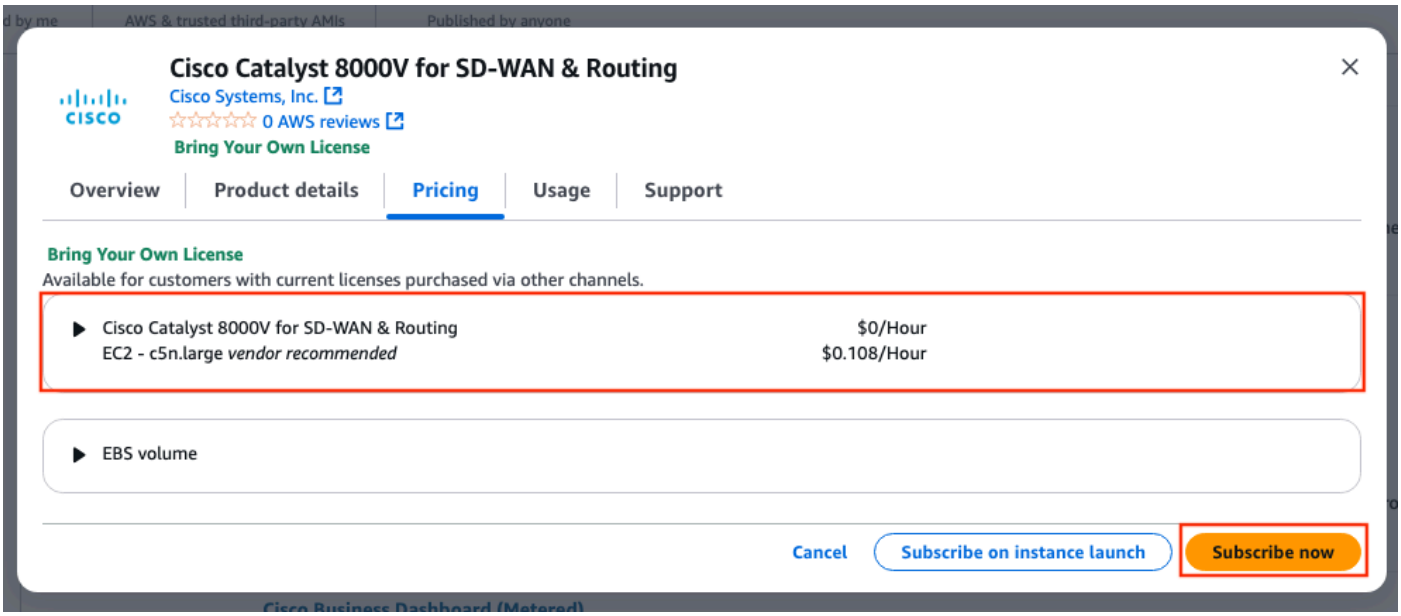


AMI 데이터베이스를 Cisco Catalyst 8000v for SD-WAN & Routing이라는 이름으로 필터링합니다. AWS Marketplace AMI 목록에서 Select(선택)를 클릭합니다.



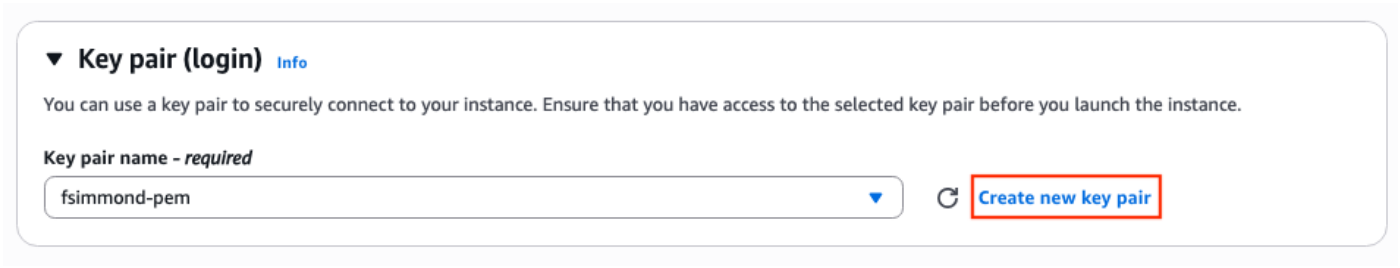
AMI의 해당 크기를 선택합니다. 이 예에서는 c5n.large size가 선택됩니다. 이는 네트워크에 필요한

용량에 따라 달라질 수 있습니다. 선택한 후에는 지금 가입을 클릭합니다.

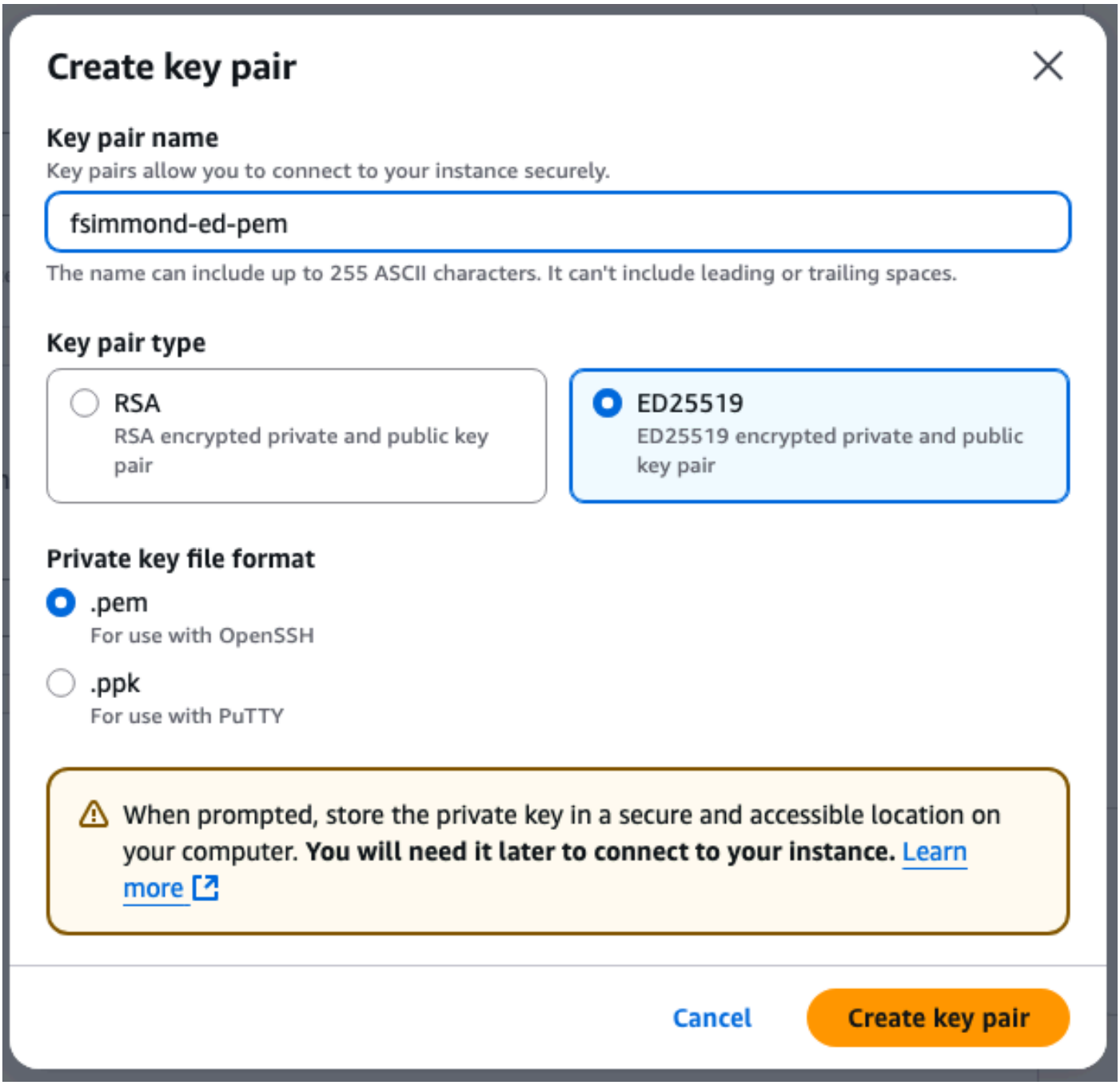


6.1단계. 원격 액세스를 위한 키 쌍 구성

AMI에 가입하면 여러 옵션이 있는 새 창이 표시됩니다. Key pair (login)(키 쌍(로그인)) 섹션에서 키 쌍이 없는 경우 Create new key pair(새 키 쌍 생성)를 클릭합니다. 생성된 모든 디바이스에 단일 키를 재사용할 수 있습니다.



새 팝업 창이 표시됩니다. 이 예에서는 ED25519 암호화를 사용하는 .pem 키 파일이 생성됩니다. 모든 것이 설정되면 키 쌍 만들기를 클릭합니다.



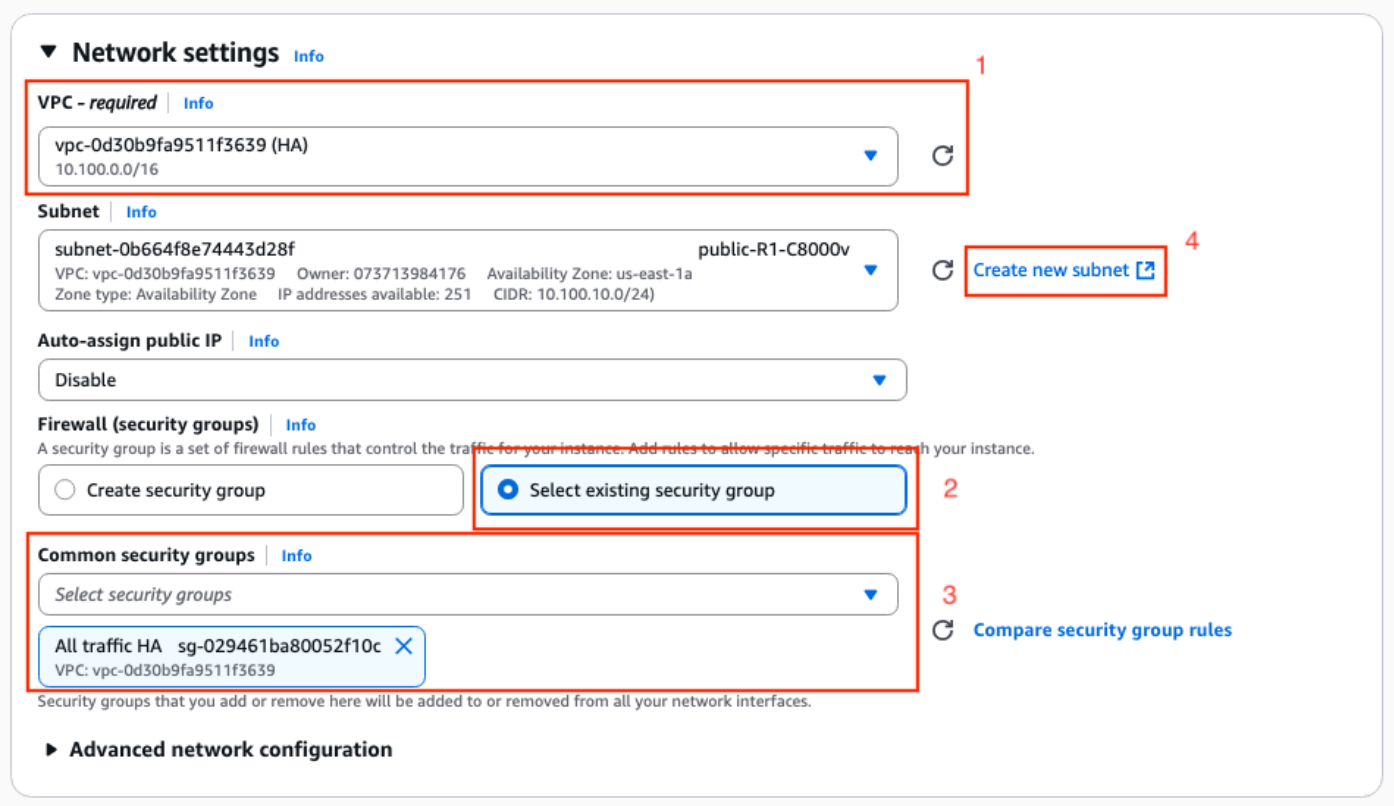
6.2단계. AMI용 서브넷 생성 및 구성

Network Settings(네트워크 설정) 섹션에서 Edit(수정)를 클릭합니다. 섹션 내의 일부 새 옵션을 사용할 수 있습니다.
1. 이 작업에 대해 원하는 VPC를 선택합니다. 이 예에서는 HA라는 VPC가 선택됩니다.

2. Firewall (security groups)(방화벽(보안 그룹)) 섹션에서 기존 보안 그룹 선택을 선택합니다.

3. 옵션 2를 선택하면 공통 보안 그룹 옵션을 사용할 수 있습니다. 원하는 보안 그룹을 필터링하고 선택합니다. 이 예에서는 All traffic HA 보안 그룹이 선택됩니다.

4. (선택 사항) 이러한 디바이스에 대한 서브넷이 생성되지 않은 경우 Create new subnet(새 서브넷 생성)을 클릭합니다.



웹 브라우저의 새 탭이 열려 있으며 이 탭을 통해 서브넷 만들기 섹션으로 이동합니다.

1. 드롭다운 목록에서 이 구성에 해당하는 VPC를 선택합니다.

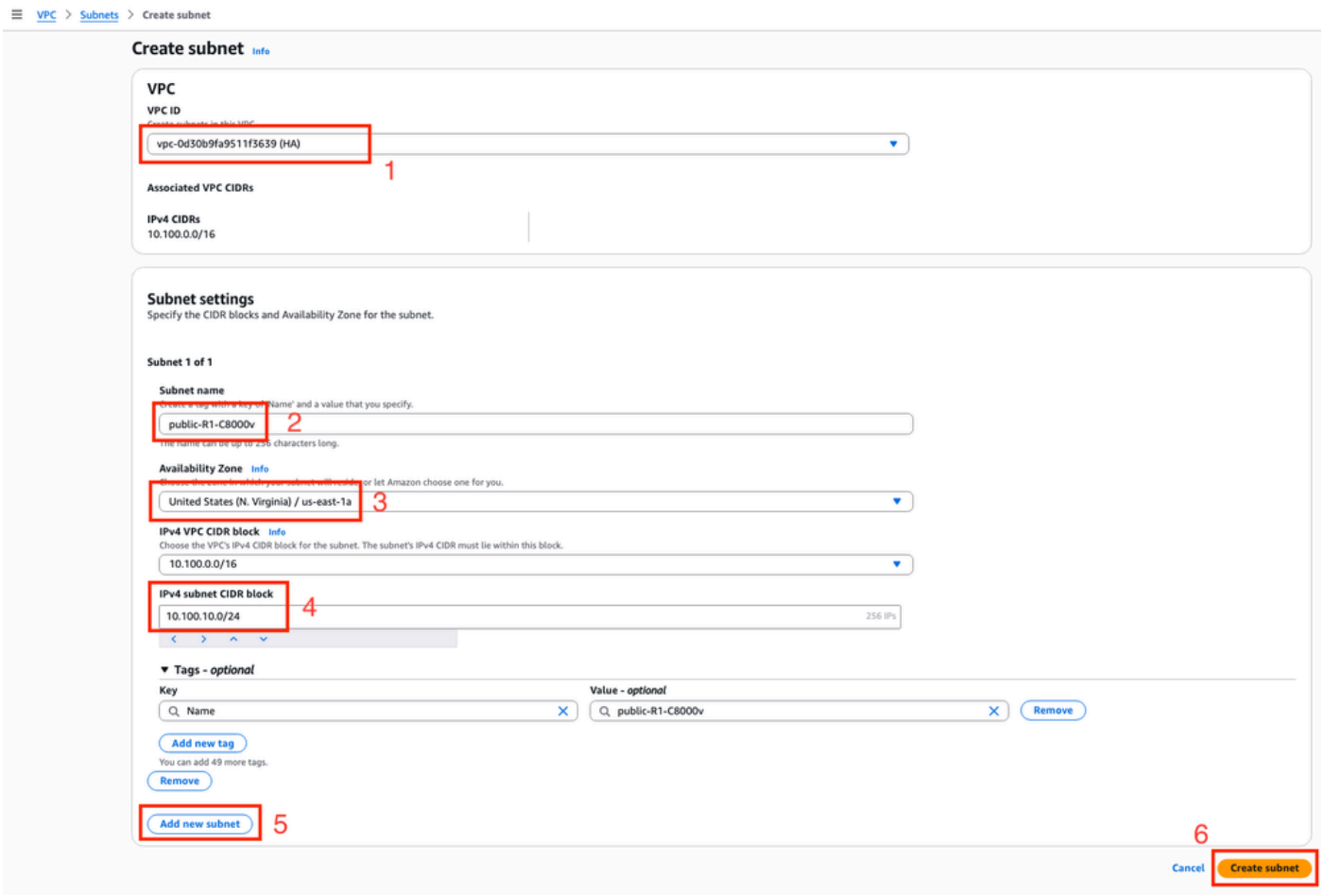
2. 새 서브넷의 이름을 설정합니다.

3. 이 서브넷의 가용 영역을 정의합니다. (설정에 대한 자세한 내용은 이 문서의 토폴로지 섹션을 참조하십시오.)

4. VPC CIDR 블록에 속하는 서브넷 블록을 설정합니다.

5. 또한 새 서브넷 추가 섹션을 클릭하고 각 서브넷에 대해 2~4단계를 반복하여 사용할 모든 서브넷을 생성할 수 있습니다.

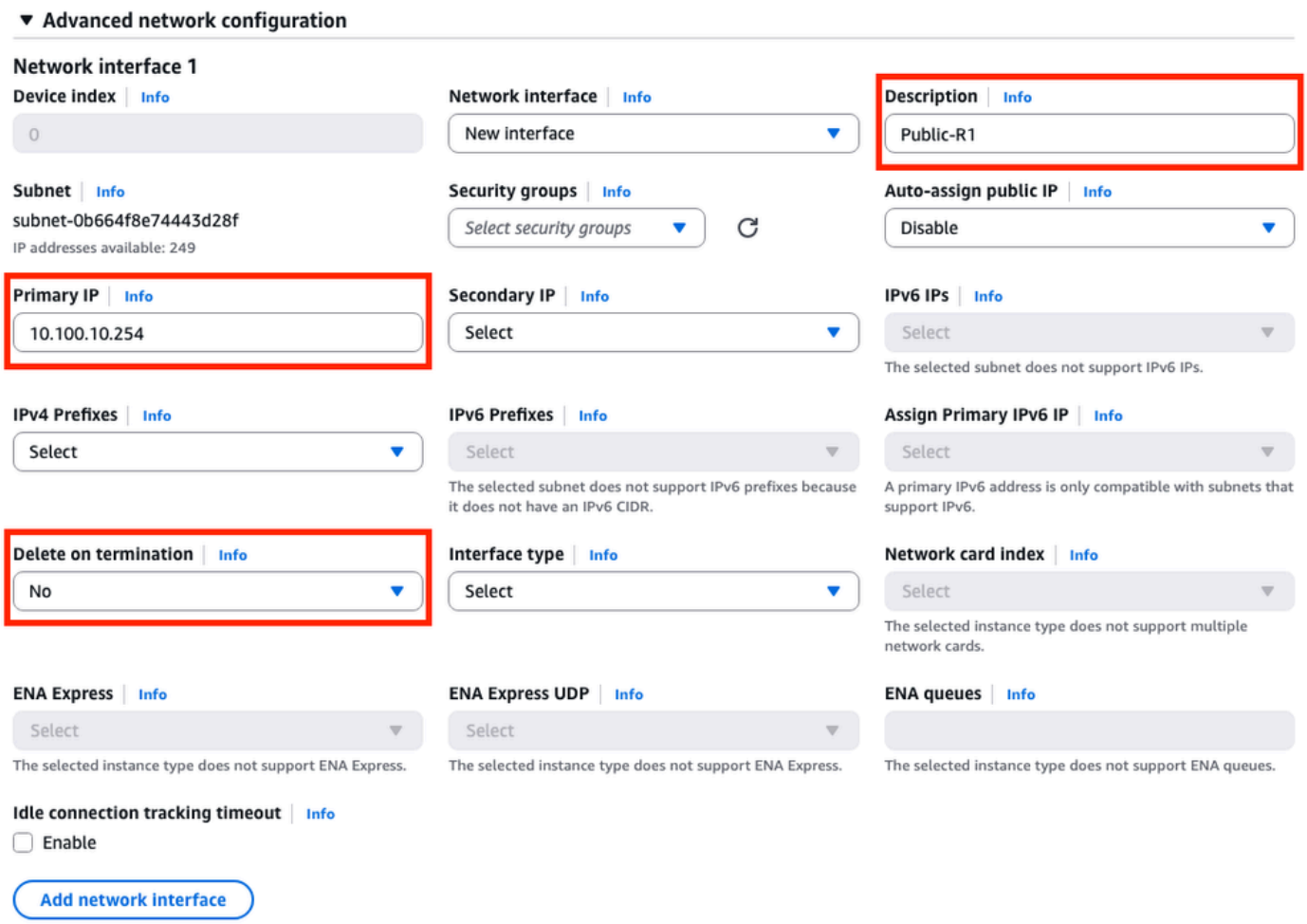
6. 완료되면 서브넷 만들기를 클릭합니다. 이전 페이지로 이동하여 설정을 계속 진행합니다.



네트워크 설정 섹션의 서브넷 하위 섹션에서 새로 고침 아이콘을 클릭하여 드롭다운 목록에서 생성된 서브넷을 가져옵니다.

6.3단계. AMI 인터페이스 구성

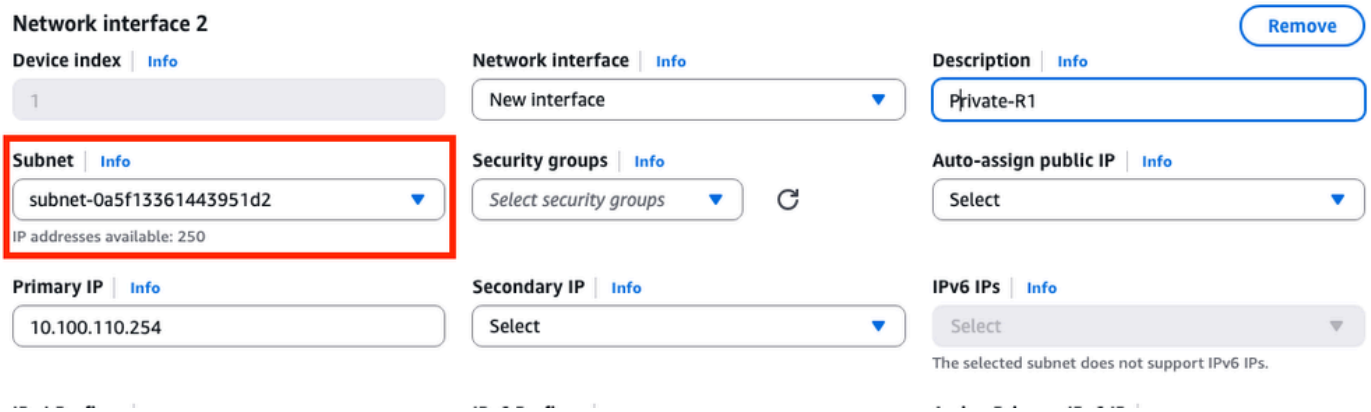
Network Settings(네트워크 설정) 섹션에서 Advanced Network configuration(고급 네트워크 컨피그레이션) 하위 섹션을 확장합니다. 다음 옵션이 표시됩니다.



이 메뉴에서 Description(설명), Primary IP(기본 IP), Delete on termination(종료 시 삭제) 매개변수를 설정합니다.
Primary IP 매개 변수에는 서브넷의 사용 가능한 첫 번째 주소를 제외한 모든 IP 주소를 사용합니다. 이는 AWS에서 내부적으로 사용됩니다.

이 예에서 Delete on termination(종료 시 삭제) 매개변수는 No(아니요)로 설정됩니다. 그러나 환경에 따라 yes로 설정할 수 있습니다.

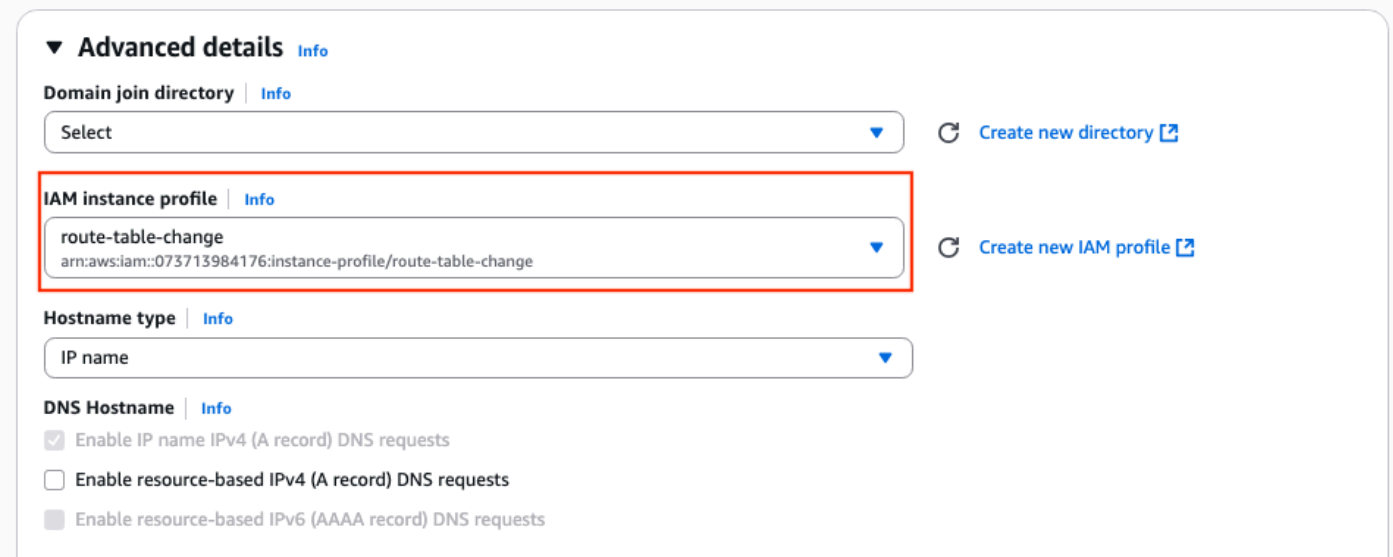
이 토폴로지로 인해 프라이빗 서브넷에 두 번째 인터페이스가 필요합니다. Add network interface(네트워크 인터페이스 추가)를 클릭하면 이 프롬프트가 표시됩니다. 그러나 인터페이스는 이번에는 서브넷을 선택할 수 있는 옵션을 제공합니다.



모든 매개변수가 네트워크 인터페이스 1에 설정된 대로 설정되면 다음 단계를 계속 진행합니다.

6.4단계. IAM 인스턴스 프로필을 AMI로 설정합니다.

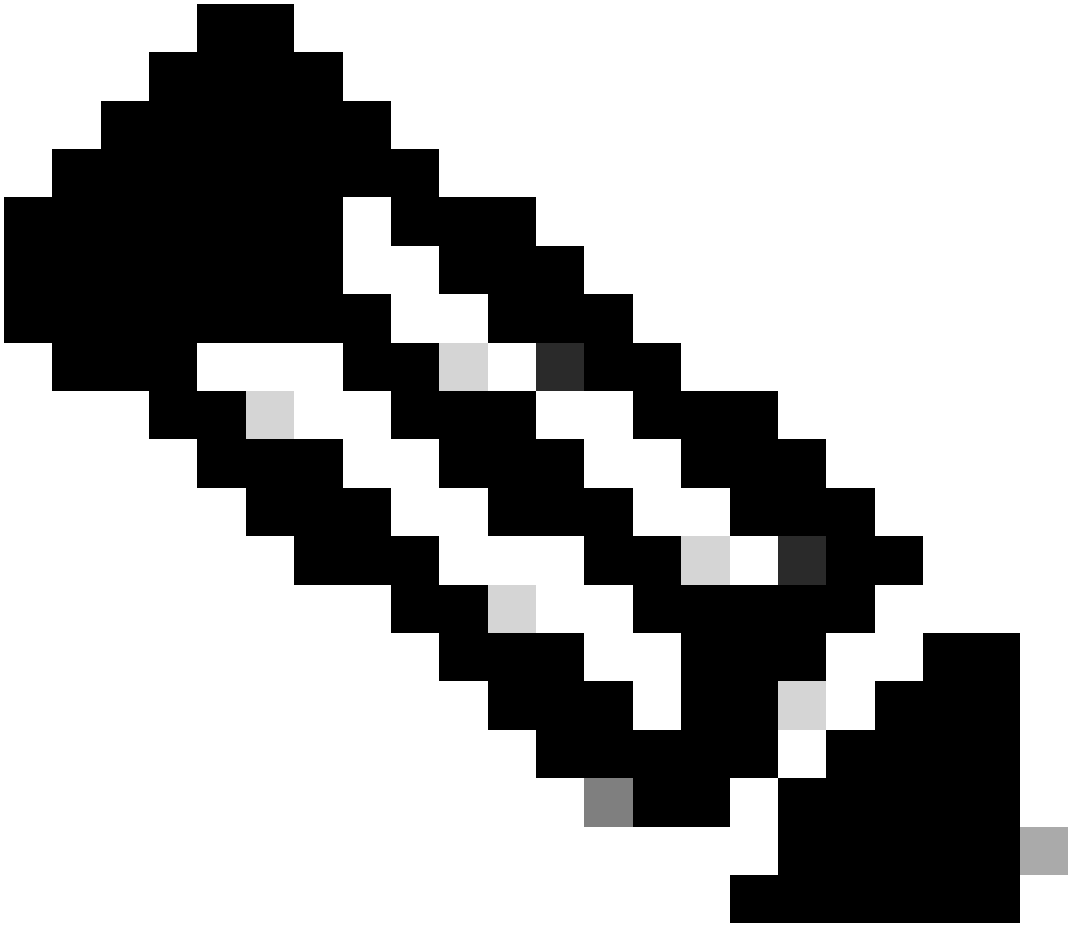
Advanced details(고급 세부 정보) 섹션에서 IAM 인스턴스 프로필 매개 변수에서 생성된 IAM 역할을 선택합니다.



6.5단계. (선택 사항) AMI에서 자격 증명을 설정합니다

Advanced details 섹션에서 User data - optional(사용자 데이터 - 선택 사항) 섹션으로 이동한 다음 이 설정을 적용하여 인스턴스가 생성되는 동안 사용자 이름 및 비밀번호를 설정합니다.

```
ios-config-1="username <username> priv 15 pass <password>"
```



참고: AWS가 C8000v에 SSH하기 위해 제공한 사용자 이름은 root로 잘못 나열될 수 있습니다. 필요한 경우 이를 ec2-user로 변경합니다.

6.6단계. 인스턴스 컨피그레이션 완료

모든 것이 구성되면 인스턴스 실행을 클릭합니다.

Summary

Number of instances | Info

1

Software Image (AMI)
Cisco Catalyst 8000V for SD-WA...read more
ami-03cc286883c62bdee

Virtual server type (instance type)
c5n.large

Firewall (security group)
All traffic HA

Storage (volumes)
1 volume(s) - 16 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

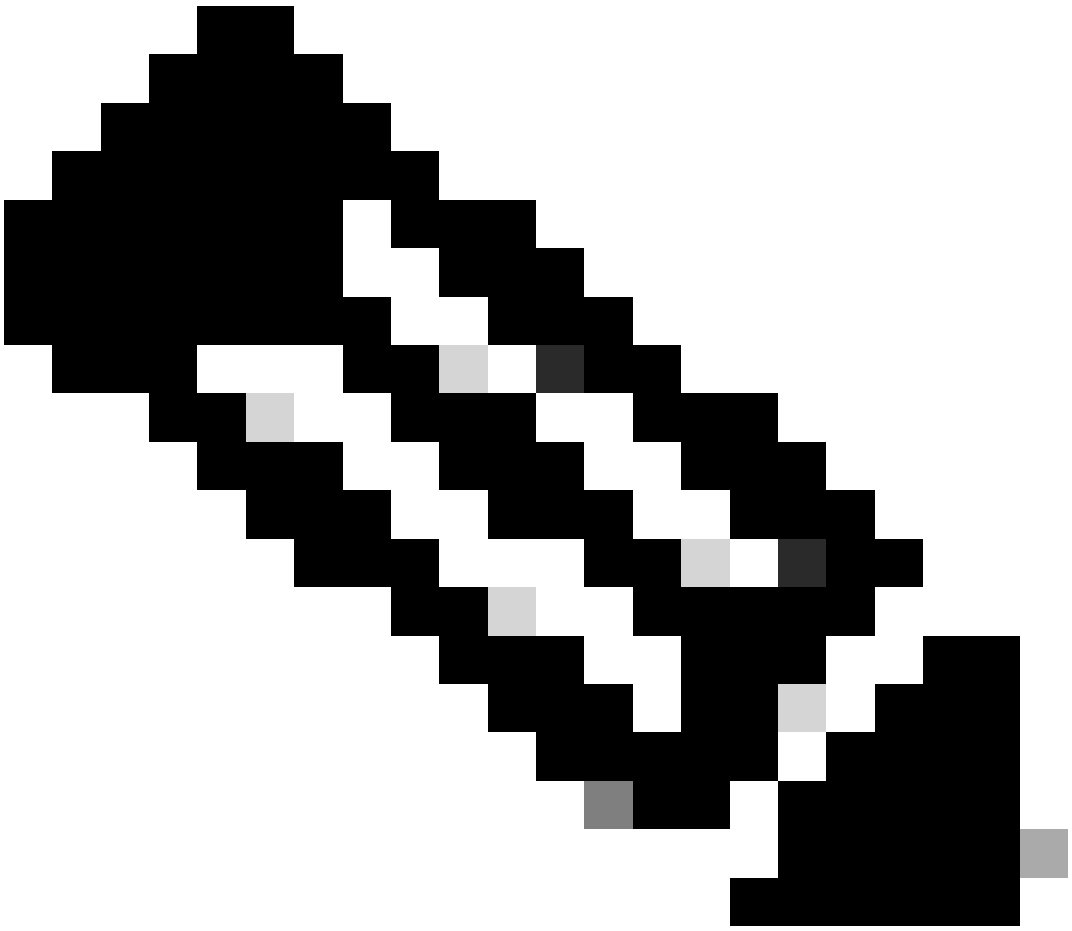
Cancel

Launch instance

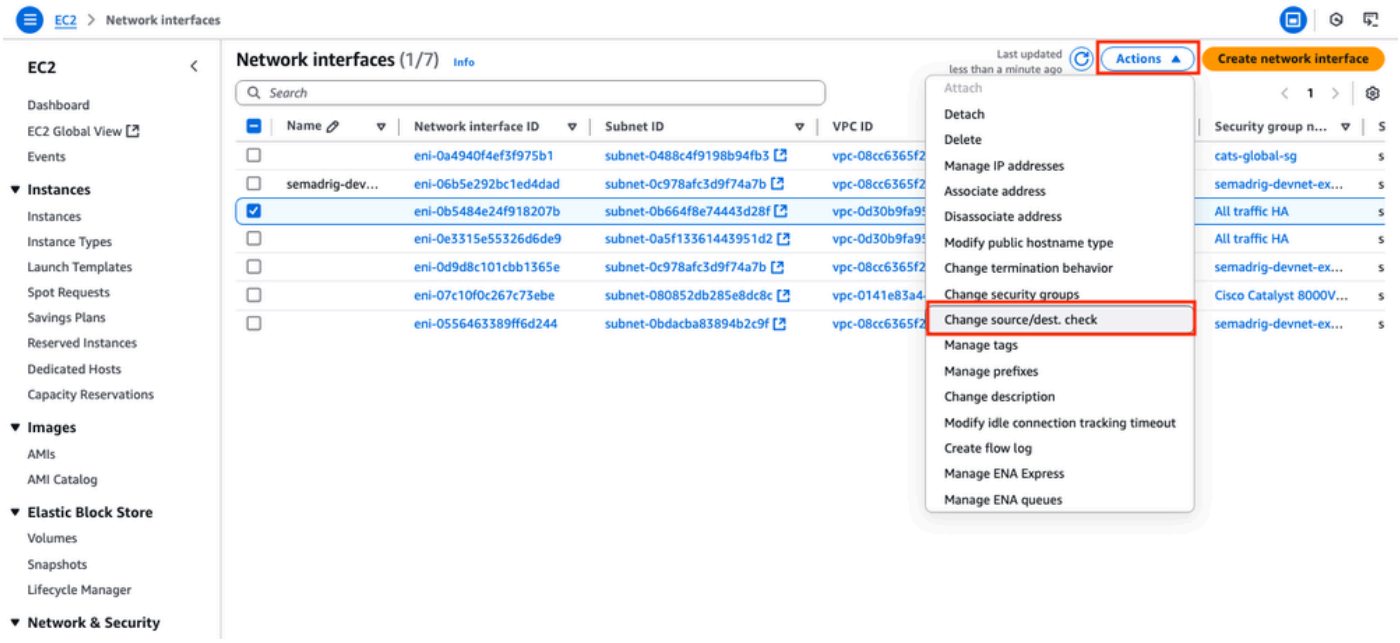
Preview code

6.7단계. ENI에서 Source/Destination Check 비활성화

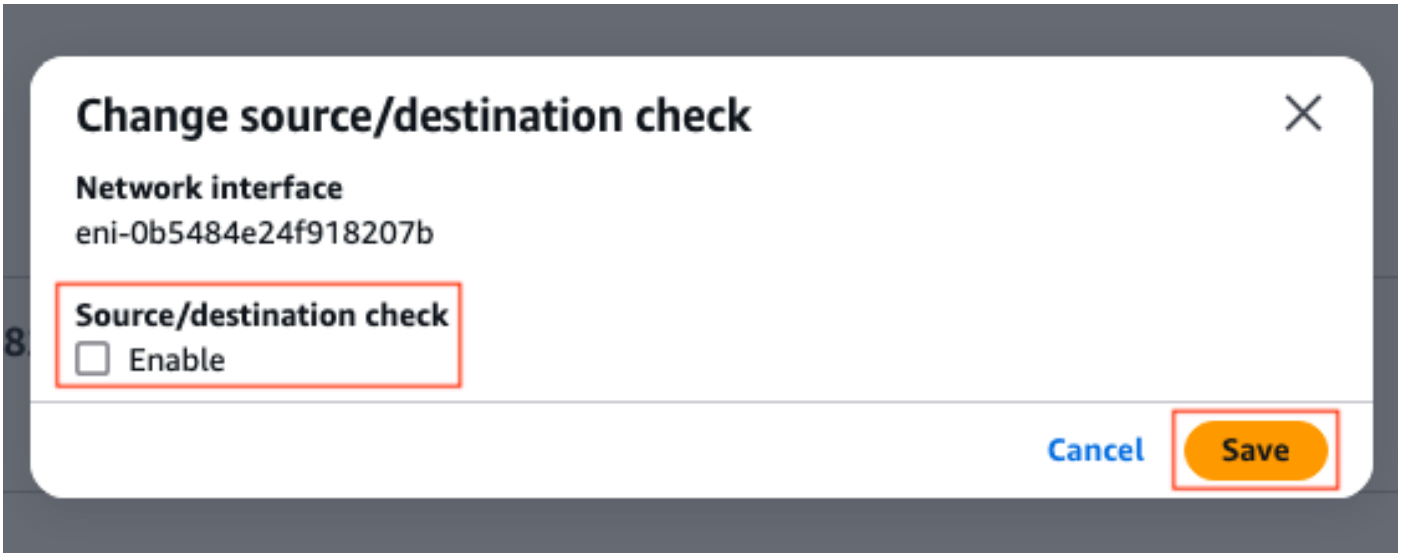
인스턴스가 생성되면 AWS에서 src/dst 검사 기능을 비활성화하여 동일한 서브넷의 인터페이스 간 연결을 가져옵니다. EC2 Dashboard(EC2 대시보드) > Network & Security(네트워크 및 보안) > Network interfaces(네트워크 인터페이스) 섹션에서 ENIs(ENI)를 선택하고 Actions(작업)를 클릭합니다 > 소스/대상 변경 확인합니다.



참고: 이 옵션을 사용하려면 ENI를 하나씩 선택해야 합니다.



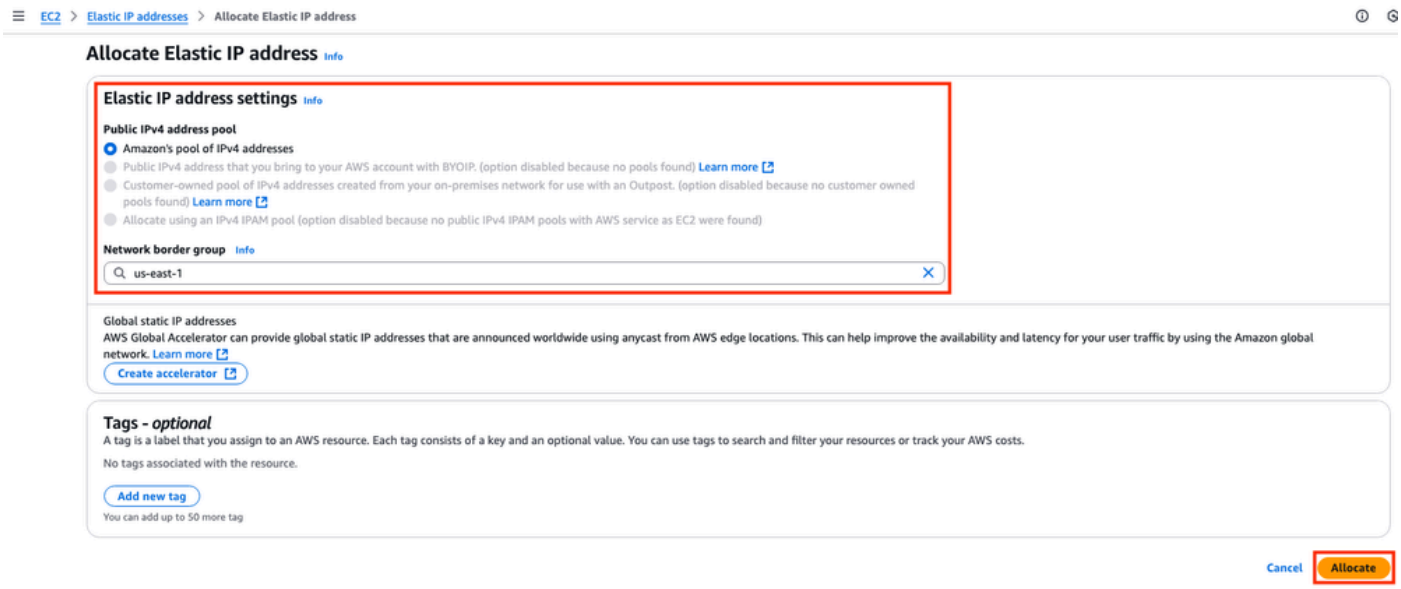
새 창이 표시됩니다. 새 메뉴에서 사용 확인란을 비활성화하고 저장을 클릭합니다.



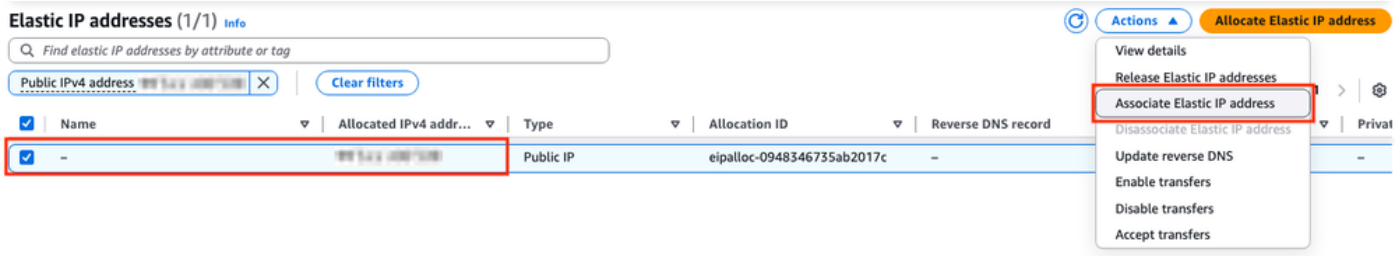
6.8단계. Elastic IP를 생성하고 인스턴스의 Public ENI에 연결

EC2 Dashboard(EC2 대시보드) > Network & Security(네트워크 및 보안) > Elastic IPs(탄력적 IP) 섹션에서 Allocate Elastic IP address(탄력적 IP 주소 할당)를 클릭합니다.

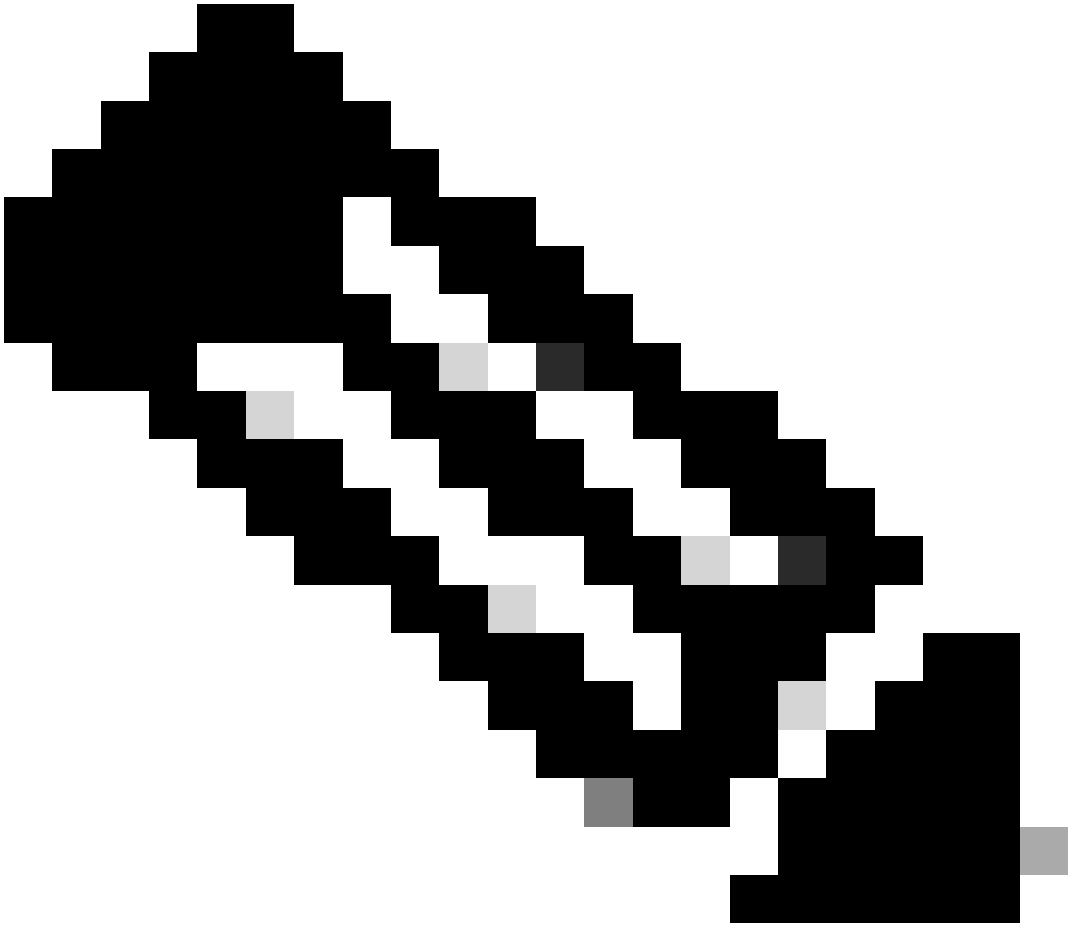
이 페이지는 다른 섹션으로 연결됩니다. 이 예에서는 가용성 영역 us-east-1과 함께 IPv4 주소의 Amazon 풀 옵션이 선택됩니다. 완료되면 [할당]을 클릭합니다.



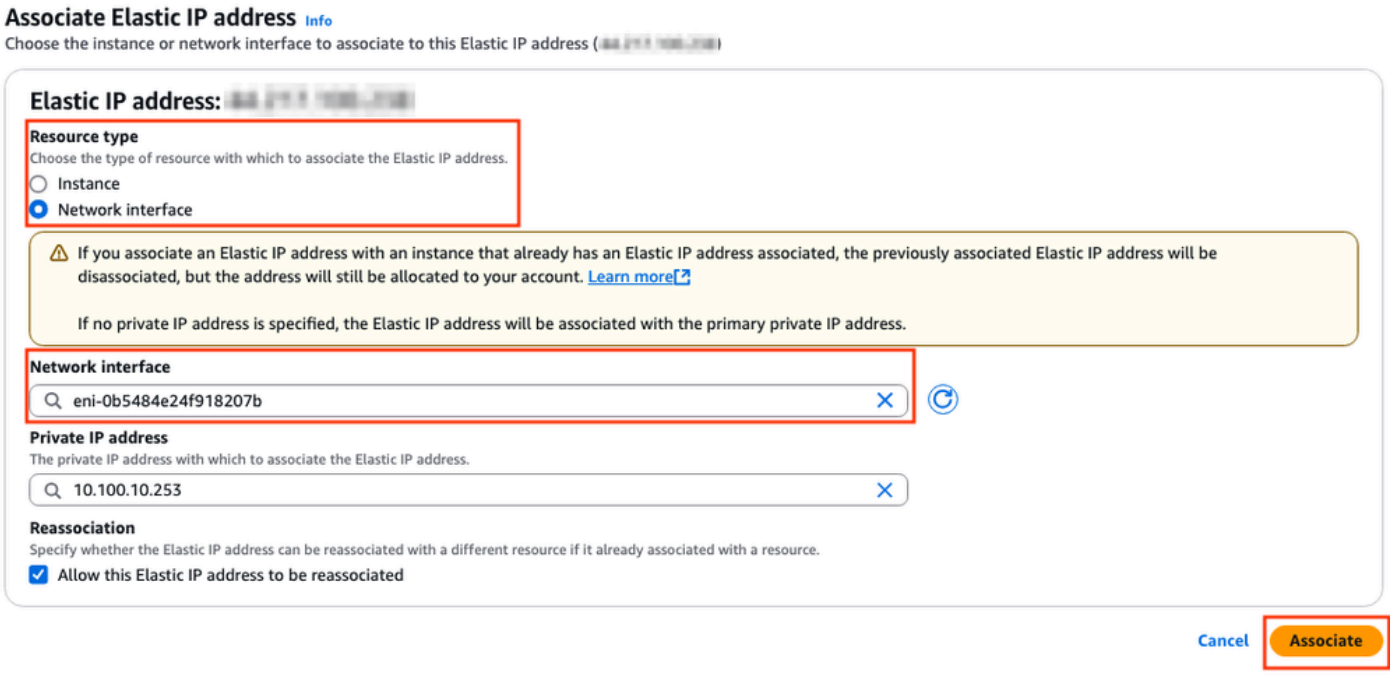
IP 주소가 생성되면 인스턴스의 공용 인터페이스에 IP 주소를 할당합니다. EC2 Dashboard(EC2 대시보드) > Network & Security(네트워크 및 보안) > Elastic IPs(탄력적 IP) 섹션에서 Actions(작업) > Associate Elastic IP address(탄력적 IP 주소 연결)를 클릭합니다.



이 새 섹션에서 Network interface 옵션을 선택하고 해당 인터페이스의 공용 ENI를 찾습니다. 해당 공용 IP 주소를 연결하고 Associate(연결)를 클릭합니다.



참고: 적절한 ENI ID를 가져오려면 EC2 Dashboard(EC2 대시보드) > Instances(인스턴스) 섹션으로 이동합니다. 그런 다음 인스턴스를 선택하고 Networking(네트워킹) 섹션을 선택합니다. 동일한 행에서 ENI 값을 가져오려면 공용 인터페이스의 IP 주소를 찾습니다.



7단계. 6단계를 반복하여 HA용 두 번째 C8000v 인스턴스를 생성합니다

각 인터페이스에 대한 해당 정보를 보려면 이 문서의 토폴로지 섹션을 참조하고 6.1에서 6.6까지 동일한 단계를 반복하십시오.

8단계. 6단계를 반복하여 AMI Marketplace에서 VM(Linux/Windows)을 생성합니다.

이 예에서는 AMI Marketplace에서 Ubuntu 서버 22.04.5 LTS가 내부 호스트로 선택됩니다.

ens5는 기본적으로 공용 인터페이스에 대해 생성됩니다. 이 예에서는 프라이빗 서브넷에 대한 두 번째 인터페이스(디바이스에서 ens6)를 생성합니다.

<#root>

```
ubuntu@ip-10-100-30-254:~$ sudo apt install net-tools
...
ubuntu@ip-10-100-30-254:~$ ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet
```

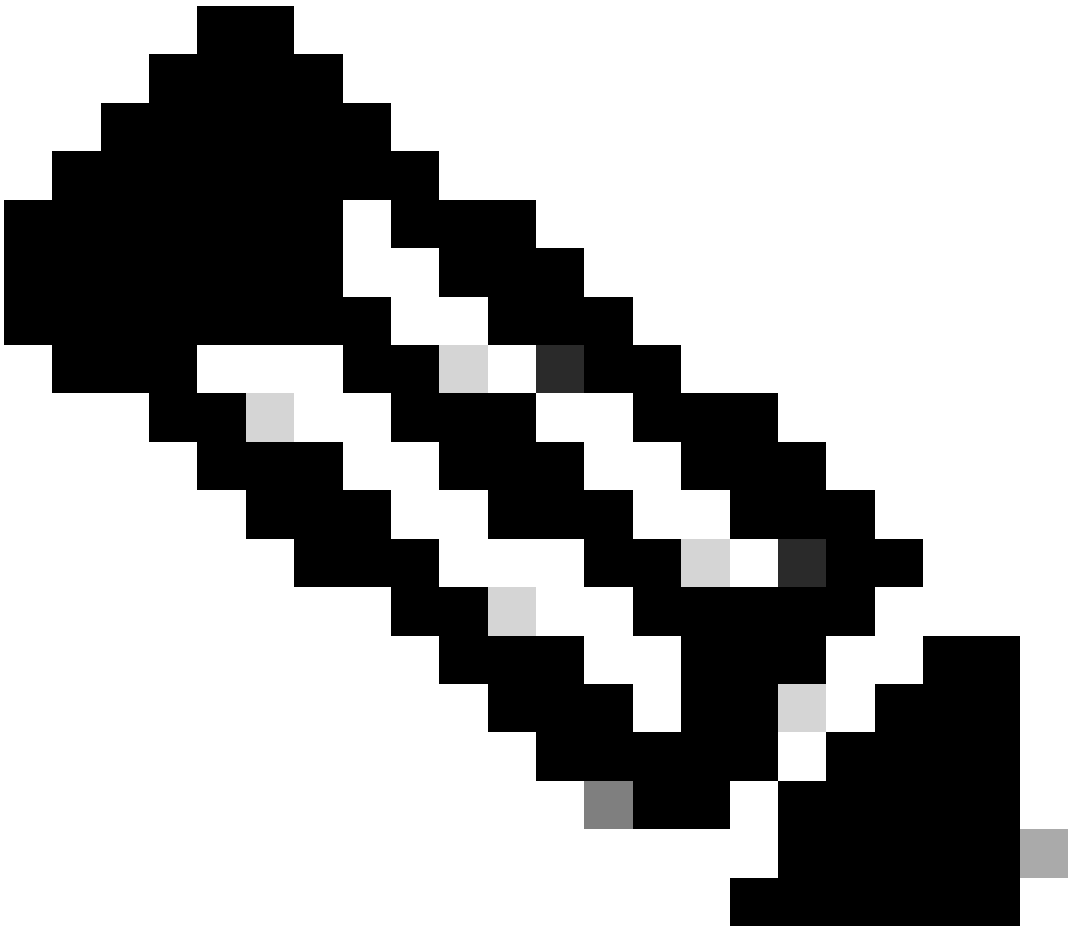
10.100.30.254

```
 netmask 255.255.255.0 broadcast 10.100.30.255
inet6 fe80::51:19ff:fea2:1151 prefixlen 64 scopeid 0x20<link>
ether 02:51:19:a2:11:51 txqueuelen 1000 (Ethernet)
RX packets 1366 bytes 376912 (376.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1417 bytes 189934 (189.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
inet
```

10.100.130.254

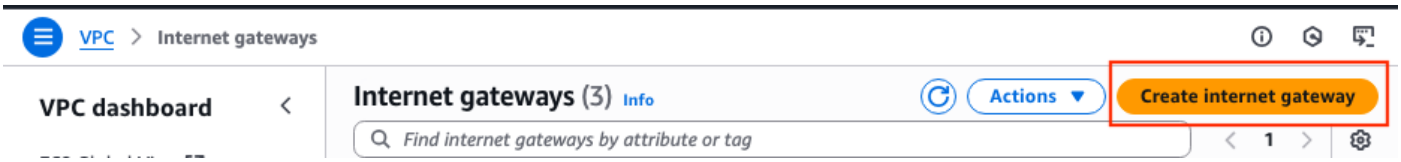
```
 netmask 255.255.255.0 broadcast 10.100.130.255
inet6 fe80::3b:7eff:fead:dbe5 prefixlen 64 scopeid 0x20<link>
ether 02:3b:7e:ad:db:e5 txqueuelen 1000 (Ethernet)
RX packets 119 bytes 16831 (16.8 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 133 bytes 13816 (13.8 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



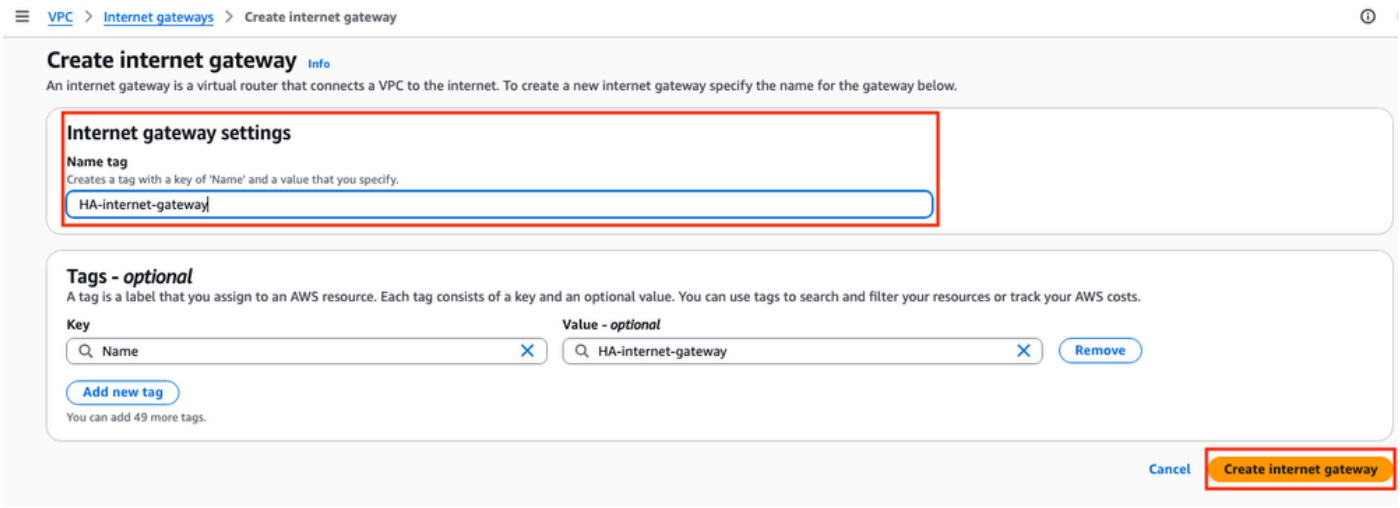
> 참고: 인터페이스에 변경 사항이 있는 경우, 인터페이스를 플랩(flap)하거나 VM을 다시 로드하여 변경 사항을 적용합니다.

9단계. VPC에 대한 IGW(인터넷 게이트웨이) 생성 및 구성

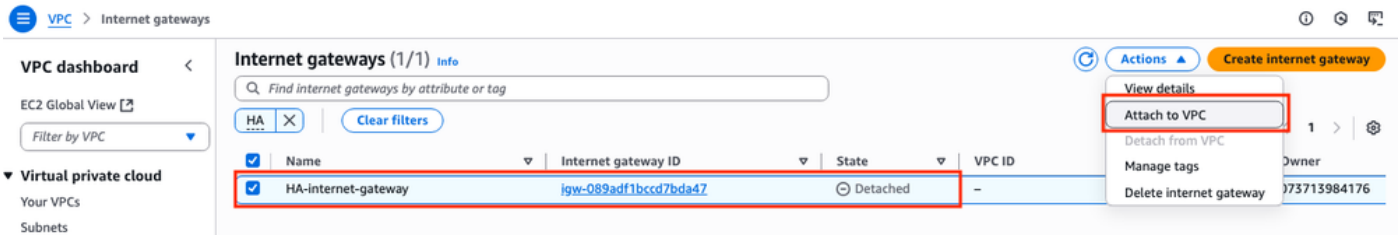
VPC Dashboard(**VPC 대시보드**) > **Virtual private cloud(가상 프라이빗 클라우드)** > **Internet gateways(인터넷 게이트웨이) 섹션**에서 Create **internet gateway(인터넷 게이트웨이 생성)**를 클릭합니다.



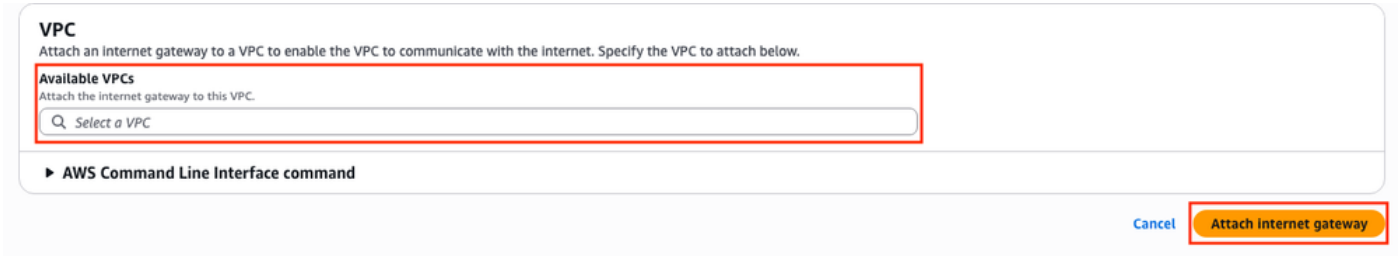
이 새 섹션에서 이 게이트웨이의 **이름 태그**를 만들고 **인터넷** 게이트웨이 만들기를 **클릭합니다.**



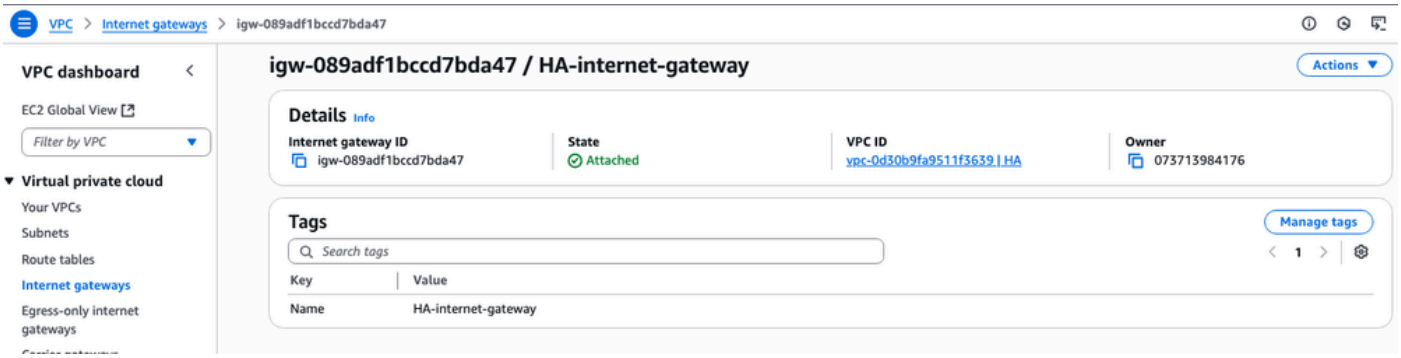
IGW가 생성되면 해당 VPC에 연결합니다. VPC Dashboard(VPC **대시보드**) > **Virtual Private Cloud(가상 프라이빗 클라우드)** > **Internet Gateway(인터넷 게이트웨이) 섹션**으로 이동하여 해당 IGW를 선택합니다. Actions(작업) **> Attach to VPC(VPC에 연결)를 클릭합니다.**



이 새 섹션에서 **HA**라는 VPC를 선택합니다. 이 예에서는 **인터넷 게이트웨이 연결을** 클릭합니다.



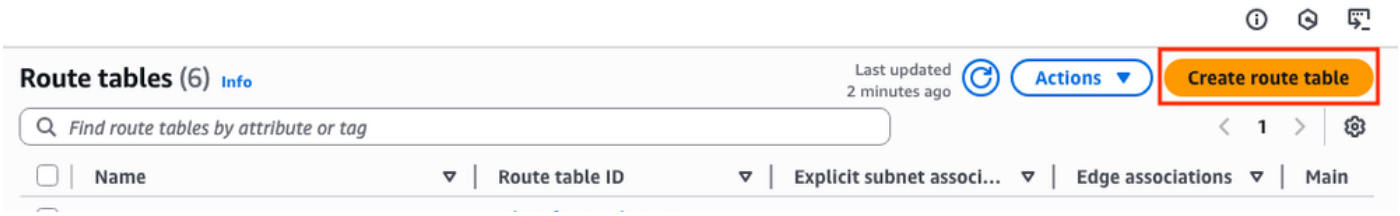
IGW는 다음과 같이 Attached 상태를 나타내야 합니다.



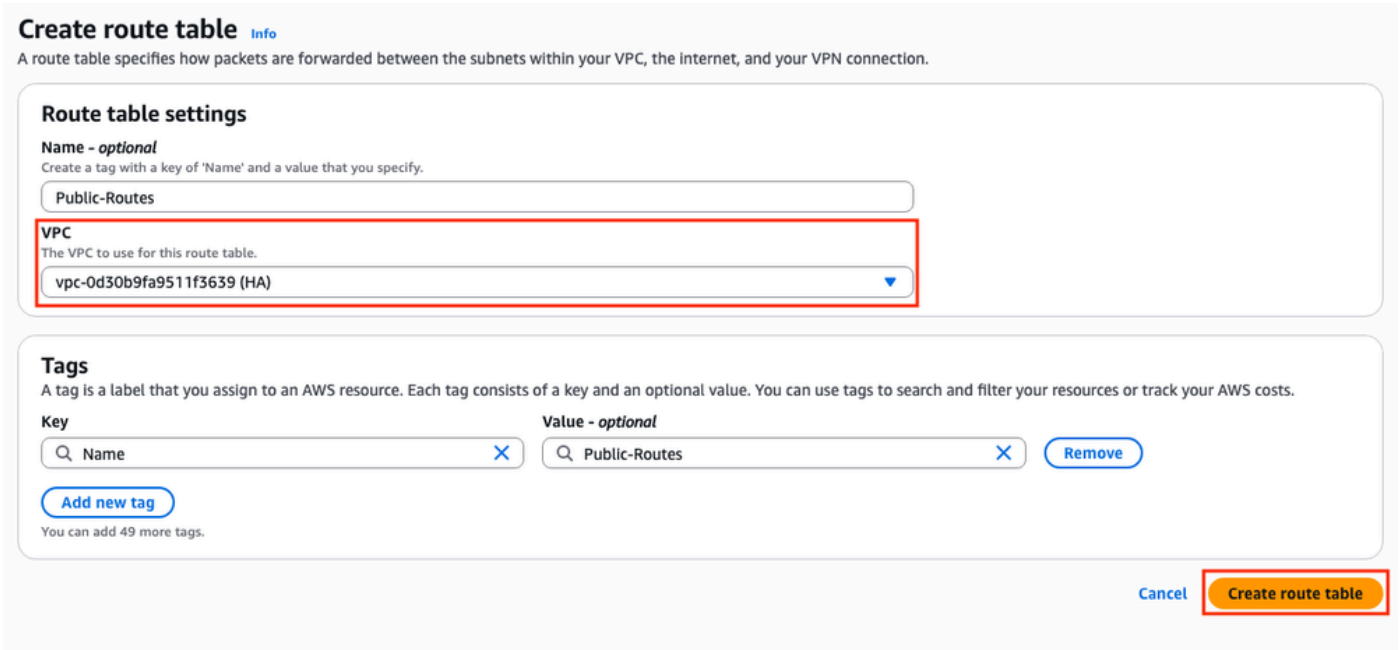
10단계. AWS에서 퍼블릭 및 프라이빗 서브넷에 대한 경로 테이블 생성 및 구성

10.1단계. 공용 경로 테이블 생성 및 구성

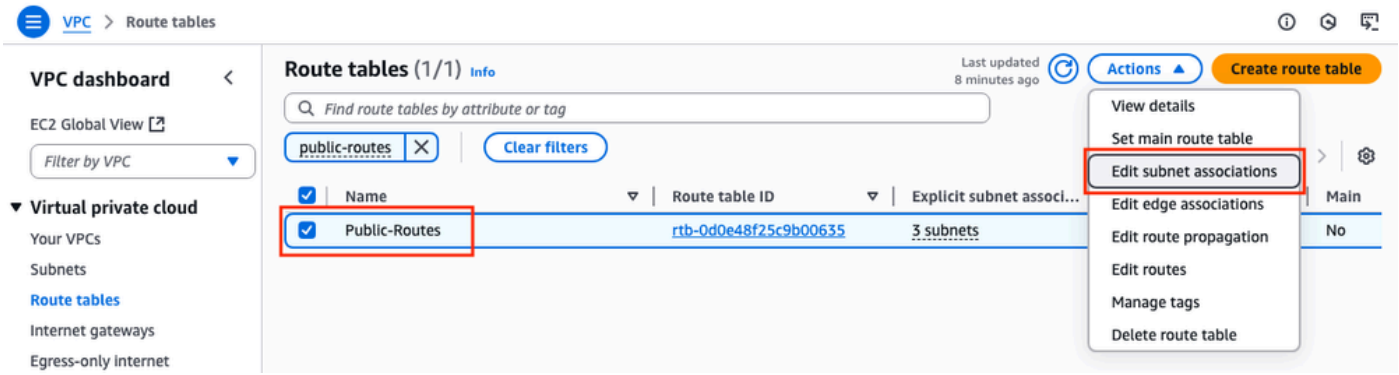
이 토폴로지에 HA를 설정하려면 해당 경로 테이블에 있는 모든 공용 및 사설 서브넷을 연결합니다. VPC Dashboard(VPC 대시보드) > Virtual Private Cloud > Route tables(경로 테이블) 섹션에서 Create route table(경로 테이블 생성)을 클릭합니다.



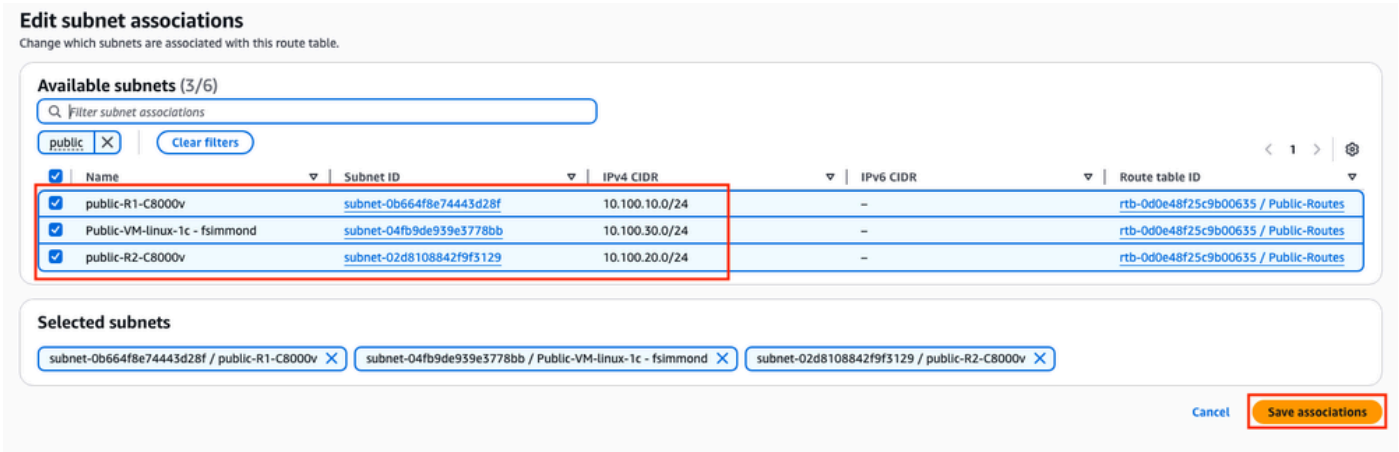
새 섹션에서 이 토폴로지에 해당하는 VPC를 선택합니다. 선택한 후 Create route table(경로 테이블 생성)을 클릭합니다.



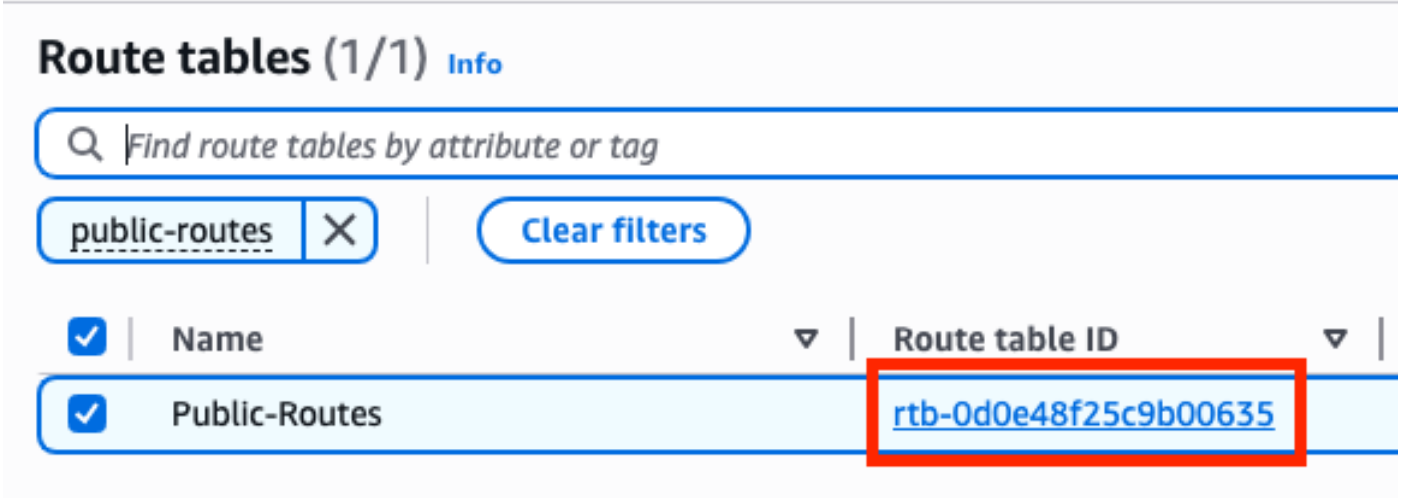
Route tables(라우팅 테이블) 섹션에서 생성된 테이블을 선택하고 Actions(작업) > Edit Subnet associations(서브넷 연결 수정)를 클릭합니다.



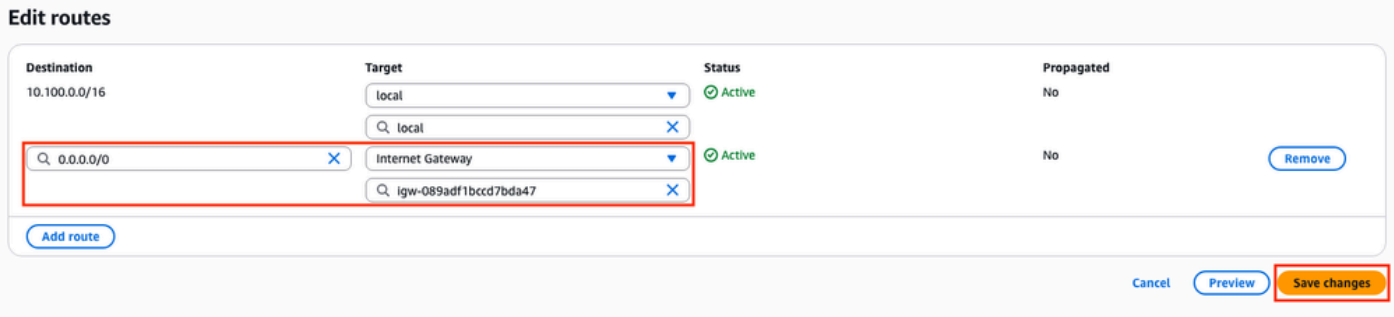
그런 다음 해당 서브넷을 선택하고 연결 저장을 클릭합니다.



서브넷이 연결되면 Route table ID(경로 테이블 ID) 하이퍼링크를 클릭하여 테이블에 알맞은 경로를 추가합니다. 그런 다음 Edit Routes(경로 수정)를 클릭합니다.

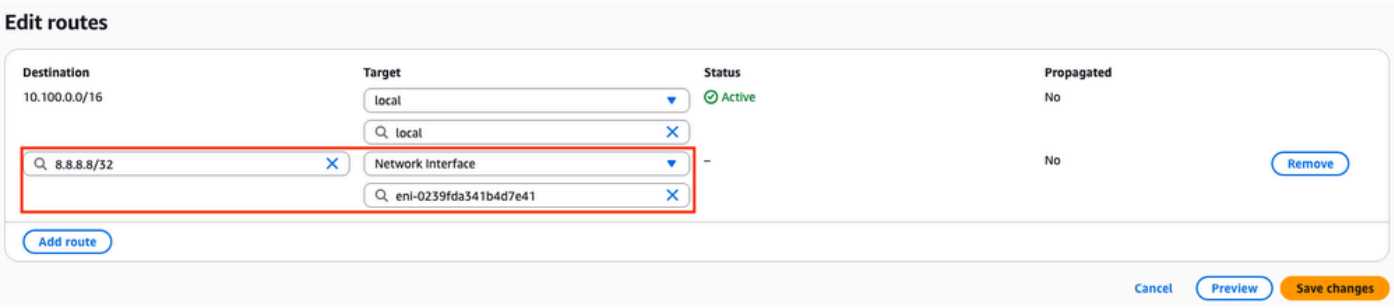


인터넷 액세스를 얻으려면 Add route(경로 추가)를 클릭하고 이 공용 경로 테이블을 9단계에서 생성된 IGW와 이러한 매개변수로 연결합니다. 선택한 후 변경 사항 저장을 클릭합니다.



10.2단계. 프라이빗 경로 테이블 생성 및 구성

이제 공용 경로 테이블이 생성되었으므로 해당 경로에 인터넷 게이트웨이를 추가하는 것을 제외하고 프라이빗 경로 및 프라이빗 서브넷에 대해 10단계를 복제합니다. 이 예에서는 8.8.8.8에 대한 트래픽이 사설 서브넷을 통과해야 하므로 라우팅 테이블이 다음과 같습니다.



11단계. 기본 네트워크 구성, NAT(Network Address Translation), BFD 및 라우팅 프로토콜을 사용하는 GRE 터널을 확인하고 구성합니다

AWS에서 인스턴스 및 해당 라우팅 구성이 준비되면 디바이스를 구성합니다.

C8000v R1 구성:

```
interface Tunnel1
ip address 192.168.200.1 255.255.255.0
bfd interval 500 min_rx 500 multiplier 3
tunnel source GigabitEthernet1
tunnel destination <Public IPv4 address of C8000v R2>
!
interface GigabitEthernet1
ip address 10.100.10.254 255.255.255.0
ip nat outside
negotiation auto
!
interface GigabitEthernet2
ip address 10.100.110.254 255.255.255.0
ip nat inside
negotiation auto
!
router eigrp 1
bfd interface Tunnel1
network 192.168.200.0
passive-interface GigabitEthernet1
!
```

```
ip access-list standard 10
10 permit 10.100.130.0 0.0.0.255
!
ip nat inside source list 10 interface GigabitEthernet1 overload
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 10.100.10.1
ip route 10.100.130.0 255.255.255.0 GigabitEthernet2 10.100.110.1
```

C8000v R2 구성:

```
interface Tunnel1
ip address 192.168.200.2 255.255.255.0
bfd interval 500 min_rx 500 multiplier 3
tunnel source GigabitEthernet1
tunnel destination<Public IPv4 address of C8000v R1>
!
interface GigabitEthernet1
ip address 10.100.20.254 255.255.255.0
ip nat outside
negotiation auto
!
interface GigabitEthernet2
ip address 10.100.120.254 255.255.255.0
negotiation auto
!
router eigrp 1
bfd interface Tunnel1
network 192.168.200.0
passive-interface GigabitEthernet1
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 10.100.20.1
ip route 10.100.130.0 255.255.255.0 GigabitEthernet2 10.100.120.1
!
ip access-list standard 10
10 permit 10.100.130.0 0.0.0.255
!
ip nat inside source list 10 interface GigabitEthernet1 overload
!

ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 10.100.20.1
ip route 10.100.130.0 255.255.255.0 GigabitEthernet2 10.100.120.1
```

12단계. 고가용성 구성(Cisco IOS® XE Denali 16.3.1a 이상)

VM 간의 이중화 및 연결이 설정되었으므로 라우팅 변경 사항을 정의하기 위해 HA 설정을 구성합니다. BFD 피어와 같은 AWS HA 오류 후 설정해야 하는 Route-table-id, Network-interface-id 및 CIDR 값을 설정합니다.

```
Router(config)# redundancy
Router(config-red)# cloud provider aws (node-id)
bfd peer <IP address of the remote device>
route-table <Route table ID>
```

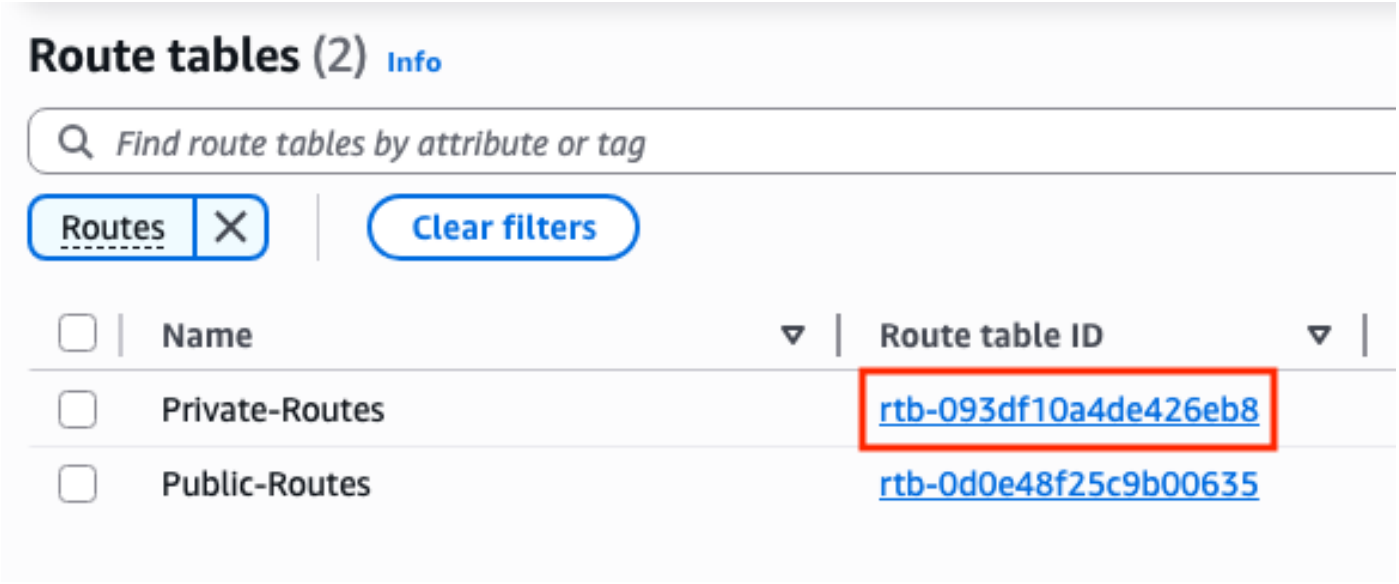
```
cidr ip <traffic to be monitored/prefix>
eni <Elastic network interface (ENI) ID>
region <region-name>
```

bfd 피어 매개변수는 터널 피어 IP 주소와 관련이 있습니다. show bfd neighbor 출력을 사용하여 이를 확인할 수 있습니다.

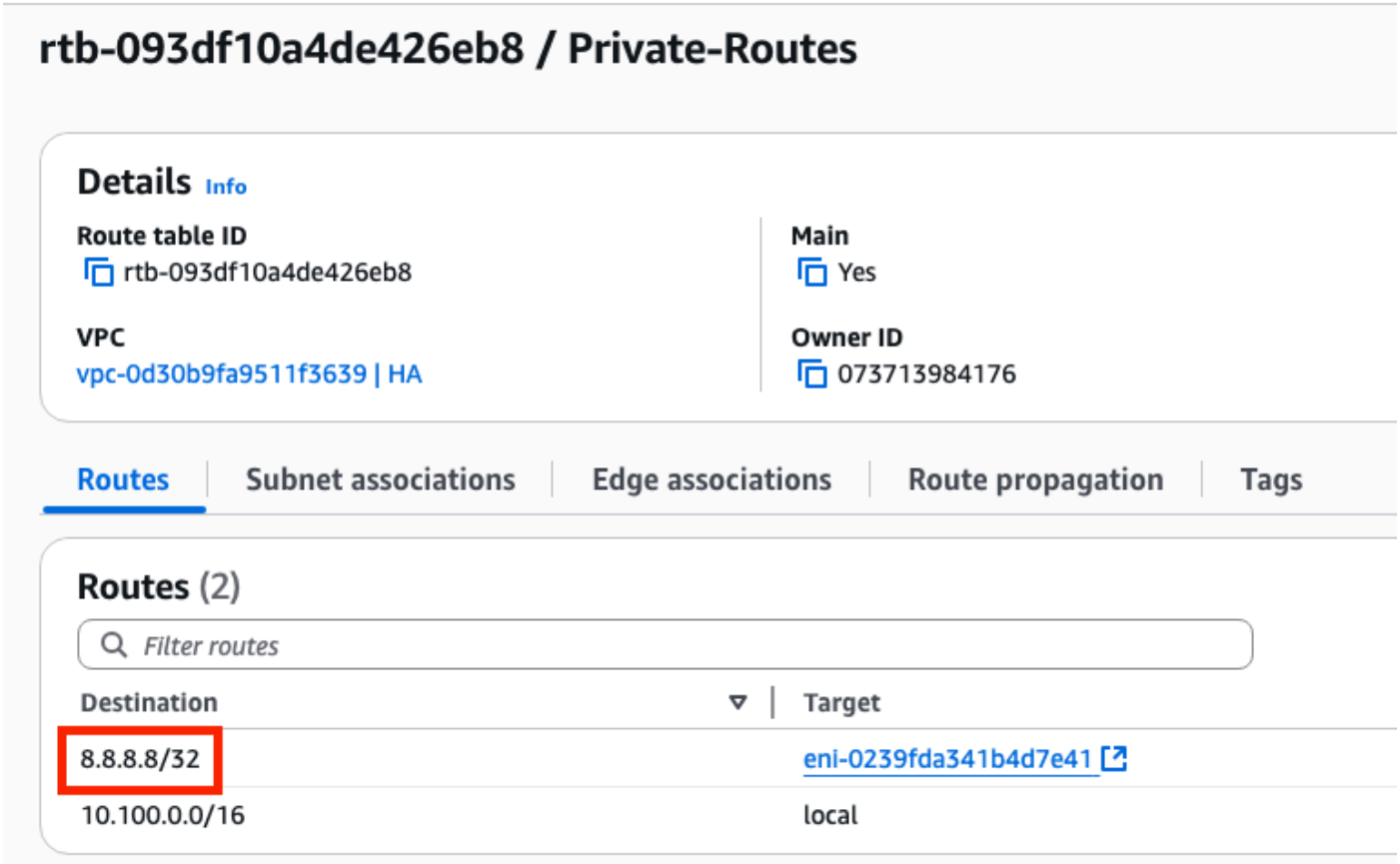
```
R1(config)#do sh bfd neighbors

IPv4 Sessions
NeighAddr LD/RD RH/RS State Int
192.168.200.2 4097/4097 Up Up Tu1
```

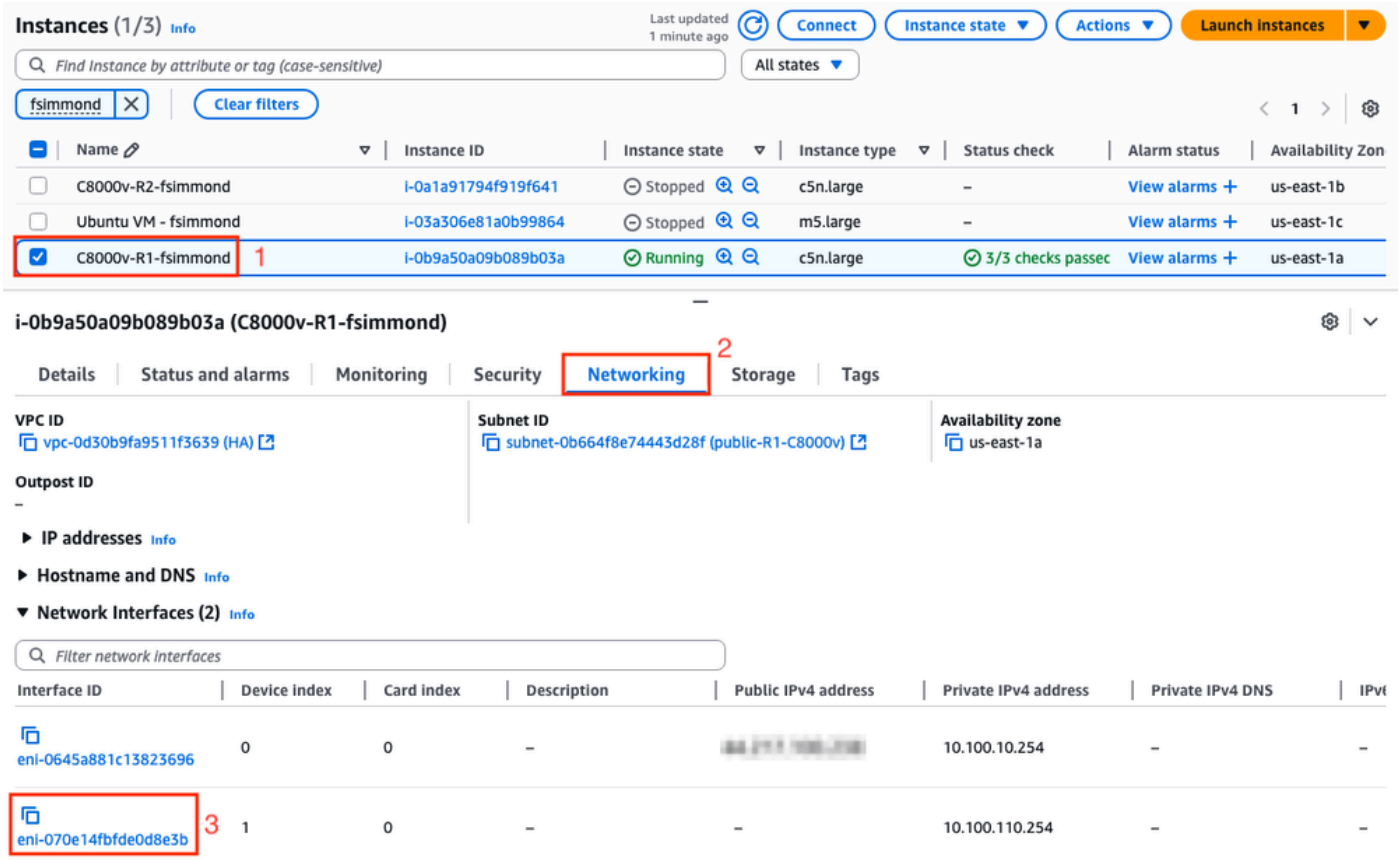
route-table 매개변수는 VPC Dashboard(VPC 대시보드) > Virtual Private Cloud(가상 프라이빗 클라우드) > Route Tables(경로 테이블) 섹션에 있는 프라이빗 경로 테이블 ID와 관련됩니다. 해당 경로 테이블 ID를 복사합니다.



cidr ip 매개변수는 Private Route Table(10.2단계에서 생성된 경로)에 추가된 경로 접두사와 관련이 있습니다.



eni 매개변수는 구성 중인 인스턴스의 해당 개인 인터페이스의 ENI ID와 관련이 있습니다. 이 예에서는 인스턴스의 인터페이스 GigabitEthernet2의 ENI ID가 사용됩니다.



region 매개변수는 VPC가 있는 리전의 AWS 설명서에 있는 코드 이름과 관련이 있습니다. 이 예에서는 us-east-1 지역이 사용됩니다.

그러나 이 목록은 변경되거나 확장될 수 있습니다. 최신 업데이트를 찾으려면 AmazonRegion and Availability Zone 문서를 방문하십시오.

이 모든 정보를 고려하여 VPC의 각 라우터에 대한 컨피그레이션 예는 다음과 같습니다.

C8000v R1의 구성 예:

```
redundancy
 cloud provider aws 1
  bfd peer 192.168.200.2
  route-table rtb-093df10a4de426eb8
  cidr ip 8.8.8.8/32
  eni eni-070e14fbfde0d8e3b
  region us-east-1
```

C8000v R2의 구성 예:

```
redundancy
 cloud provider aws 1
  bfd peer 192.168.200.1
  route-table rtb-093df10a4de426eb8
  cidr ip 8.8.8.8/32
  eni eni-0239fda341b4d7e41
  region us-east-1
```

확인

1. C8000v R1 인스턴스의 상태를 확인합니다. 터널 및 클라우드 이중화가 작동 및 실행 중인지 확인합니다.

```
R1#show bfd neighbors

IPv4 Sessions
NeighAddr LD/RD RH/RS State Int
192.168.200.2 4097/4097 Up Up Tu1
```

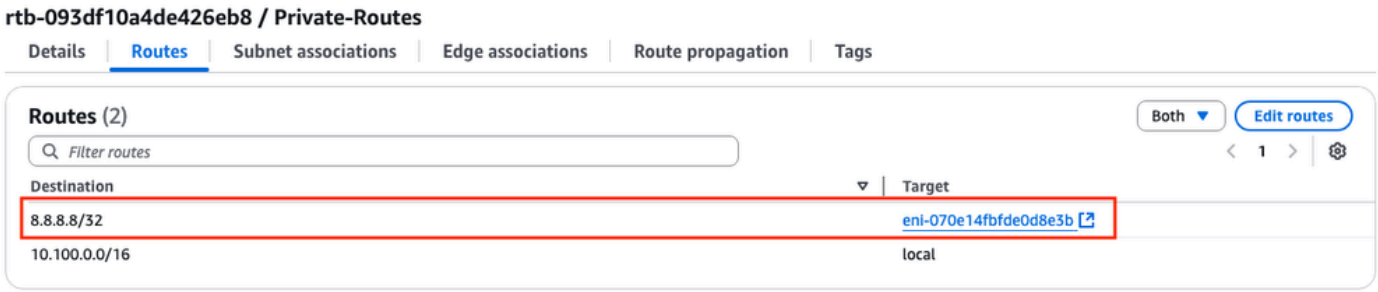
```
R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num
0 192.168.200.2 Tu1 10 00:16:52 2 1470 0 2
```

```
R1#show redundancy cloud provider aws 1
Provider : AWS node 1
BFD peer = 192.168.200.2
BFD intf = Tunnel1
route-table = rtb-093df10a4de426eb8
cidr = 8.8.8.8/32
eni = eni-070e14fbfde0d8e3b
region = us-east-1
```

2. 라우터 뒤에 있는 호스트 VM에서 8.8.8.8로 ping을 계속 실행합니다. Ping이 사설 인터페이스를 통과하는지 확인하십시오.

```
ubuntu@ip-10-100-30-254:~$ ping -I ens6 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 10.100.130.254 ens6: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=1.36 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=1.30 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=1.34 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=114 time=1.28 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=114 time=1.31 ms
```

3. AWS WebGUI를 열고 라우팅 테이블의 상태를 확인합니다. 현재 ENI는 R1 인스턴스의 전용 인터페이스에 속합니다.



4. R1 인스턴스에서 Tunnel1 인터페이스를 종료하여 경로 변경을 트리거하여 HA 장애 조치 이벤트를 시뮬레이션합니다.

```
R1#config t
R1(config)#interface tunnel1
R1(config-if)#shut
```

5. AWS의 route table에서 다시 확인하십시오. ENI ID가 R2 프라이빗 인터페이스 ENI ID로 변경되었습니다.

문제 해결

다음은 배포를 다시 만드는 동안 종종 잊거나 잘못 구성되는 공통점입니다.

- 리소스가 연결되었는지 확인합니다. VPC, 서브넷, 인터페이스, 경로 테이블 등을 생성할 때 이들 중 상당수는 자동으로 서로 연결되지 않습니다. 그들은 서로에 대해 전혀 모른다.
- Elastic IP 및 모든 Private IP가 올바른 인터페이스와 연결되어 있는지 확인합니다. 올바른 서브넷은 올바른 라우트 테이블에 추가되고 올바른 라우터에 연결되며 올바른 VPC 및 영역은 IAM 역할 및 보안 그룹과 연결됩니다.

- ENI별로 Source/Dest(소스/대상) 검사를 비활성화합니다.
  이 섹션에서 설명한 모든 사항을 이미 확인했으며 문제가 아직 해결되지 않은 경우, 이러한 출력을 수집하고 가능한 경우 HA 장애 조치를 테스트한 후 Cisco TAC에 케이스를 여십시오.

```
show redundancy cloud provider aws <node-id>
debug redundancy cloud all
debug ip http all
```

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.