

---

# Amazon Virtual Private Cloud

사용 설명서



## Amazon Virtual Private Cloud: 사용 설명서

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

## Table of Contents

Amazon VPC란 무엇인가? .....	1
Amazon VPC 개념 .....	1
VPC 및 서브넷 .....	1
지원되는 플랫폼 .....	2
기본 VPC와 기본이 아닌 VPC .....	2
인터넷 액세스 .....	2
회사 또는 홈 네트워크에 액세스 .....	3
처음으로 Amazon VPC 사용하기 .....	3
Amazon VPC를 다른 AWS 서비스와 함께 사용 .....	3
Amazon VPC에 액세스 .....	4
Amazon VPC 가격 .....	5
Amazon VPC 제한 .....	5
PCI DSS 준수 .....	5
시작하기 .....	6
Amazon VPC 시작하기 .....	6
1단계: VPC 생성 .....	7
2단계: 보안 그룹 만들기 .....	9
3단계: VPC에서 인스턴스 시작 .....	10
4단계: 인스턴스에 탄력적 IP 주소 할당 .....	11
5단계: 정리 .....	11
IPv6 시작하기 .....	12
1단계: VPC 생성 .....	12
2단계: 보안 그룹 만들기 .....	14
3단계: 인스턴스 시작 .....	15
시나리오 및 예시 .....	17
시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC .....	17
개요 .....	18
라우팅 .....	19
보안 .....	19
시나리오 1 구현 .....	21
시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC(NAT) .....	23
개요 .....	24
라우팅 .....	25
보안 .....	26
시나리오 2 구현 .....	29
NAT 인스턴스를 사용하여 시나리오 2 구현 .....	32
시나리오 3: 공인 및 사설 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC .....	34
개요 .....	34
라우팅 .....	35
보안 .....	37
시나리오 3 구현 .....	40
시나리오 4: 사설 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC .....	44
개요 .....	44
라우팅 .....	45
보안 .....	46
시나리오 4 구현 .....	47
예: AWS CLI를 사용하여 IPv4 VPC 및 서브넷 생성 .....	49
1단계: VPC와 서브넷 만들기 .....	49
2단계: 서브넷을 퍼블릭으로 만들기 .....	50
3단계: 서브넷에서 인스턴스 시작 .....	51
4단계: 정리 .....	53
예: AWS CLI를 사용하여 IPv6 VPC 및 서브넷 생성 .....	54
1단계: VPC와 서브넷 만들기 .....	54
2단계: 퍼블릭 서브넷 구성 .....	55

3단계: 외부 전용 프라이빗 서브넷 구성 .....	57
4단계: 서브넷의 IPv6 주소 지정 동작 변경 .....	58
5단계: 퍼블릭 서브넷에서 인스턴스 시작 .....	58
6단계: 프라이빗 서브넷으로 인스턴스를 시작 .....	59
7단계: 정리 .....	61
VPC 및 서브넷 .....	62
VPC 및 서브넷 기본 사항 .....	62
VPC 및 서브넷 크기 .....	63
IPv4의 경우, VPC 및 서브넷 크기 조정 .....	63
IPv6의 경우, VPC 및 서브넷 크기 조정 .....	64
서브넷 라우팅 .....	64
서브넷 보안 .....	65
로컬 네트워크 및 기타 VPC와의 연결 .....	65
VPC 및 서브넷 관련 작업 .....	66
VPC 만들기 .....	66
IPv6 CIDR 블록을 VPC와 연결 .....	67
VPC에 서브넷 추가 .....	67
IPv6 CIDR 블록을 서브넷에 연결 .....	68
서브넷에서 인스턴스 시작 .....	68
VPC 또는 서브넷에 연결된 IPv6 CIDR 블록을 분리 .....	68
서브넷 삭제 .....	69
VPC 삭제 .....	69
CLI 개요 .....	70
기본 VPC 및 기본 서브넷 .....	72
기본 VPC의 기본 사항 .....	72
가용성 .....	72
구성 요소 .....	73
기본 서브넷 .....	73
지원되는 플랫폼 및 기본 VPC 보유 여부 확인 .....	74
콘솔을 사용하여 플랫폼 지원 확인 .....	74
명령줄을 사용하여 플랫폼 지원 확인 .....	74
기본 VPC로 EC2 인스턴스 시작 .....	74
콘솔을 사용하여 EC2 인스턴스 시작 .....	74
명령줄을 사용하여 EC2 인스턴스 시작 .....	75
기본 서브넷과 기본 VPC 삭제 .....	75
IP 주소 지정 .....	76
프라이빗 IPv4 주소 .....	77
퍼블릭 IPv4 주소 .....	78
IPv6 주소 .....	78
서브넷에 대한 IP 주소 지정 동작 .....	79
IP 주소 작업 .....	79
서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 .....	80
서브넷의 퍼블릭 IPv6 주소 지정 속성 수정 .....	80
인스턴스 시작 시 퍼블릭 IPv4 주소 배정 .....	80
인스턴스 시작 시 IPv6 주소 배정 .....	81
인스턴스에 IPv6 주소 할당 .....	82
인스턴스에 할당된 IPv6 주소 해제 .....	82
API 및 명령 개요 .....	83
IPv6로 마이그레이션하기 .....	83
예: 퍼블릭 및 프라이빗 서브넷이 있는 VPC에서 IPv6 사용 .....	84
보안 .....	94
보안 그룹 및 네트워크 ACL 비교 .....	95
보안 그룹 .....	95
보안 그룹 기본 사항 .....	95
VPC의 기본 보안 그룹 .....	96
보안 그룹 규칙 .....	97
EC2-Classical 및 EC2-VPC용 보안 그룹의 차이점 .....	98

보안 그룹 작업 .....	99
API 및 CLI 개요 .....	102
네트워크 ACL .....	102
네트워크 ACL 기본 사항 .....	103
네트워크 ACL 규칙 .....	103
기본 네트워크 ACL .....	103
사용자 지정 네트워크 ACL .....	105
취발성 포트 .....	109
네트워크 ACL 작업 .....	109
예: 서브넷 내 인스턴스에 대한 액세스 제어 .....	112
API 및 명령 개요 .....	114
VPC에 권장되는 네트워크 ACL 규칙 .....	115
시나리오 1을 위한 권장 규칙 .....	115
시나리오 2를 위한 권장 규칙 .....	118
시나리오 3을 위한 권장 규칙 .....	123
시나리오 4를 위한 권장 규칙 .....	128
Controlling Access .....	130
AWS CLI 또는 SDK용 예제 정책 .....	130
콘솔용 예제 정책 .....	137
VPC 흐름 로그 .....	144
흐름 로그 기본 사항 .....	144
흐름 로그 제한 .....	145
흐름 로그 레코드 .....	145
흐름 로그에 대한 IAM 역할 .....	146
흐름 로그 작업 .....	147
문제 해결 .....	149
API 및 CLI 개요 .....	150
예: 흐름 로그 레코드 .....	150
예: 흐름 로그에 대한 CloudWatch 지표 필터 및 경보 만들기 .....	152
VPC 네트워킹 구성 요소 .....	153
네트워크 인터페이스 .....	153
라우팅 테이블 .....	154
라우팅 테이블 기본 사항 .....	154
라우팅 우선순위 .....	156
라우팅 옵션 .....	157
라우팅 테이블 작업 .....	160
API 및 명령 개요 .....	163
인터넷 게이트웨이 .....	165
인터넷 액세스 활성화 .....	165
인터넷 게이트웨이로 VPC 생성 .....	166
외부 전용 인터넷 게이트웨이 .....	170
외부 전용 인터넷 게이트웨이 기본 사항 .....	170
외부 전용 인터넷 게이트웨이 작업 .....	170
API 및 CLI 개요 .....	172
NAT .....	172
NAT 게이트웨이 .....	173
NAT 인스턴스 .....	182
NAT 인스턴스 및 NAT 게이트웨이 비교 .....	188
DHCP 옵션 세트 .....	189
DHCP 옵션 세트 개요 .....	190
Amazon DNS 서버 .....	191
DHCP 옵션 변경 .....	191
DHCP 옵션 세트를 사용한 작업 .....	192
API 및 명령 개요 .....	193
DNS .....	194
DNS 호스트 이름 .....	194
VPC에서 DNS 지원 .....	194

EC2 인스턴스의 DNS 호스트 이름 보기 .....	195
VPC에 대한 DNS 지원 업데이트 .....	196
프라이빗 호스팅 영역 사용 .....	197
VPC 피어링 .....	197
탄력적 IP 주소 .....	197
탄력적 IP 주소 기본 사항 .....	198
탄력적 IP 주소 작업 .....	198
API 및 CLI 개요 .....	200
VPC 엔드포인트 .....	200
엔드포인트 기본 사항 .....	201
엔드포인트 사용 제어 .....	203
서비스에 대한 액세스 제어 .....	204
Amazon S3에 대한 엔드포인트 .....	204
엔드포인트 작업 .....	208
API 및 CLI 개요 .....	211
.....	211
VPN 연결 .....	213
하드웨어 가상 프라이빗 게이트웨이 .....	214
VPN의 구성 요소 .....	214
VPN 구성의 예 .....	215
VPN 라우팅 옵션 .....	215
VPN 연결을 위해 필요한 항목 .....	215
VPN 연결을 위해 2개의 VPN 터널 구성 .....	216
중복 VPN 연결을 사용하여 장애 조치 제공 .....	216
VPN 연결 설정 .....	217
인스턴스의 종단 간 연결 테스트 .....	219
손상된 자격 증명 교체 .....	220
VPN 연결의 고정 경로 편집 .....	220
VPN 연결 삭제 .....	220
API 및 CLI 개요 .....	221
VPN CloudHub .....	223
제한 .....	224
VPC 및 서브넷 .....	224
탄력적 IP 주소 (IPv4) .....	225
흐름 로그 .....	225
게이트웨이 .....	225
네트워크 ACL .....	226
네트워크 인터페이스 .....	226
라우팅 테이블 .....	226
보안 그룹 .....	227
VPC 피어링 연결 .....	228
VPC 엔드포인트 .....	228
VPN 연결 .....	228
문서 기록 .....	229
AWS Glossary .....	232

# Amazon VPC란 무엇인가?

Amazon Virtual Private Cloud(Amazon VPC)을 사용하면 정의한 가상 네트워크에서 Amazon Web Services(AWS) 리소스를 시작할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용하는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.

## 항목

- [Amazon VPC 개념 \(p. 1\)](#)
- [처음으로 Amazon VPC 사용하기 \(p. 3\)](#)
- [Amazon VPC를 다른 AWS 서비스와 함께 사용 \(p. 3\)](#)
- [Amazon VPC에 액세스 \(p. 4\)](#)
- [Amazon VPC 가격 \(p. 5\)](#)
- [Amazon VPC 제한 \(p. 5\)](#)
- [PCI DSS 준수 \(p. 5\)](#)

## Amazon VPC 개념

Amazon VPC를 처음 사용하려면 이 가상 네트워크의 핵심 개념을 이해하고 기존의 네트워크와 비슷한 점은 무엇이고, 다른 점은 무엇인지를 파악해야 합니다. 이 단원에서는 Amazon VPC의 핵심 개념에 대해 간단히 설명합니다.

Amazon VPC는 Amazon EC2의 네트워킹 계층입니다. Amazon EC2를 처음 사용할 경우 [Amazon EC2란?](#)(Linux 인스턴스용 Amazon EC2 사용 설명서를 참조하여 개요를 파악하십시오.

## VPC 및 서브넷

Virtual Private Cloud(VPC)는 사용자의 AWS 계정 전용 가상 네트워크입니다. VPC는 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 분리되어 있습니다. Amazon EC2 인스턴스와 같은 AWS 리소스를 VPC에서 실행할 수 있습니다. VPC를 구성할 수 있으며 그 IP 주소 범위를 선택하고 서브넷을 생성하고 라우팅 테이블, 네트워크 게이트웨이 및 보안 설정을 구성할 수 있습니다.

서브넷은 VPC의 IP 주소 범위입니다. 선택한 서브넷에서 AWS 리소스를 시작할 수 있습니다. 인터넷에 연결되어야 하는 리소스에는 퍼블릭 서브넷을 사용하고, 인터넷에 연결되지 않는 리소스에는 프라이빗 서브넷을 사용하십시오. 퍼블릭 서브넷과 프라이빗 서브넷에 대한 자세한 내용은 [VPC 및 서브넷 기본 사항 \(p. 62\)](#)을 참조하십시오.

각 서브넷의 AWS 리소스를 보호하기 위해 보안 그룹 및 네트워크 액세스 제어 목록(ACL)을 비롯한 여러 보안 계층을 사용할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

## 지원되는 플랫폼

Amazon EC2 오리지널 버전은 다른 고객과 공유하며 EC2-Classic 플랫폼이라고 하는 일반적인 단일 네트워크를 지원했습니다. 이전 AWS 계정은 이 플랫폼을 계속 지원하며, EC2-Classic 또는 VPC에서 인스턴스를 시작할 수 있습니다. 2013년 12월 4일 이후에 만든 계정은 EC2-VPC만 지원합니다. 자세한 내용은 [지원되는 플랫폼 및 기본 VPC 보유 여부 확인 \(p. 74\)](#) 단원을 참조하십시오.

EC2-Classic 대신 VPC에서 인스턴스를 시작하면 다음의 장점이 있습니다.

- 인스턴스의 시작/중지에 상관 없이 유지되는 고정 IPv4 주소 할당
- IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 인스턴스에 할당하는 옵션을 사용할 수 있습니다.
- 인스턴스에 여러개의 IP 주소 할당이 가능합니다.
- 네트워크 인터페이스를 정의하고, 하나 혹은 그 이상의 네트워크 인터페이스를 VPC 인스턴스에 설치 가능합니다.
- 인스턴스가 실행중이라도 상관없이, 인스턴스의 보안 그룹 멤버십 변경이 가능합니다.
- 인스턴스의 인바운드 트래픽 제어(인그레스 필터링) 뿐만 아니라 아웃바운드 트래픽도 제어(이그레스 필터링) 가능합니다.
- 네트워크 액세스 제어 리스트(ACL)를 통해, 인스턴스에대한 액세스 보안이 한단계 더 강화되었습니다.
- 단일 테넌트 하드웨어에서 인스턴스 실행

## 기본 VPC와 기본이 아닌 VPC

EC2-VPC 플랫폼만 지원하는 계정일 경우 각 가용 영역에 기본 서브넷이 있는 기본 VPC가 제공됩니다. 기본 VPC는 EC2-VPC가 제공하는 고급 기능의 장점을 가지고 있으며 바로 사용할 수 있게 설정되어 있습니다. 기본 VPC가 있고 서브넷을 지정하지 않을 경우 인스턴스를 시작할 때 기본 VPC로 인스턴스가 시작됩니다. Amazon VPC에 대한 지식이 전혀 없어도 기본 VPC로 인스턴스를 시작할 수 있습니다.

계정에서 지원하는 플랫폼과 상관없이 VPC를 직접 생성할 수 있으며 필요에 따라 구성할 수 있습니다. 이를 기본이 아닌 VPC라고 합니다. 기본이 아닌 VPC에 만든 서브넷과 기본 VPC에 만든 추가 서브넷은 기본이 아닌 서브넷이라고 합니다.

## 인터넷 액세스

VPC에서 시작한 인스턴스가 VPC 외부의 리소스를 어떻게 액세스할지를 제어할 수 있습니다.

기본 VPC에는 인터넷 게이트웨이가 포함되며, 각각의 기본 서브넷은 퍼블릭 서브넷입니다. 기본 서브넷에서 시작한 각 인스턴스에는 프라이빗 IPv4 주소와 퍼블릭 IPv4 주소가 있습니다. 이러한 인스턴스는 인터넷 게이트웨이를 통해 인터넷과 통신할 수 있습니다. 인터넷 게이트웨이를 통해 인스턴스는 Amazon EC2 네트워크 엣지를 통해 인터넷에 연결할 수 있습니다.

기본적으로 기본이 아닌 서브넷에서 시작한 각 인스턴스에는 프라이빗 IPv4 주소가 있으며, 시작 시 특별히 지정하거나 서브넷의 퍼블릭 IP 주소 속성을 수정하지 않는 한 퍼블릭 IPv4 주소는 없습니다. 이러한 인스턴스는 서로 통신할 수는 있지만 인터넷에 액세스할 수는 없습니다.

기본이 아닌 서브넷에서 시작한 인스턴스에 대해 해당 VPC에 인터넷 게이트웨이를 추가하고(해당 VPC가 기본 VPC가 아닐 경우) 인스턴스에 탄력적 IP 주소를 연결하여 인터넷 액세스를 가능하게 할 수 있습니다.

또는 VPC의 인스턴스가 인터넷으로 아웃바운드 연결을 시작할 수 있도록 하되 인터넷으로부터의 원치 않는 인바운드 연결은 차단하려면 IPv4 트래픽용 네트워크 주소 변환(NAT) 디바이스를 사용하면 됩니다. NAT는 여러 개의 프라이빗 IPv4 주소를 하나의 퍼블릭 IPv4 주소에 매핑합니다. NAT 디바이스는 탄력적 IP 주소를 가지며, 인터넷 게이트웨이를 통해 인터넷에 연결됩니다. 프라이빗 서브넷의 인스턴스를 NAT 디바이스를 통해 인터넷에 연결할 수 있으며, 이렇게 하면 인스턴스의 트래픽이 인터넷 게이트웨이로 라우팅되고, 모든 응답은 인스턴스로 라우팅됩니다.

자세한 내용은 [NAT \(p. 172\)](#) 단원을 참조하십시오.



Amazon 제공 IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 인스턴스에 할당할 수도 있습니다. 인스턴스는 인터넷 게이트웨이를 통해 IPv6로 인터넷에 접속할 수 있습니다. 또는 인스턴스는 외부 전용 인터넷 게이트웨이를 사용하여 IPv6를 통해 인터넷에 대한 아웃바운드 연결을 시작할 수 있습니다. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 섹션을 참조하십시오. IPv6 트래픽은 IPv4 트래픽과 분리되어 있으므로, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다.

## 회사 또는 홈 네트워크에 액세스

원할 경우 IPsec 하드웨어 VPN 연결을 사용하여 VPC를 회사의 데이터 센터에 연결함으로써 회사 데이터 센터를 AWS 클라우드로 확장할 수 있습니다.

VPN 연결은 VPC에 추가된 가상 프라이빗 게이트웨이와, 데이터 센터에 위치하는 고객 게이트웨이로 구성됩니다. 가상 프라이빗 게이트웨이는 VPN 연결에서 Amazon 측에 있는 VPN 집선기입니다. 고객 게이트웨이는 VPN 연결에서 고객 측에 있는 물리적 디바이스 또는 소프트웨어 애플리케이션입니다.

자세한 내용은 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#) 단원을 참조하십시오.

## 처음으로 Amazon VPC 사용하기

Amazon VPC에 대해 실습해 보려면 [시작하기 \(p. 6\)](#)의 연습을 완료하십시오. 이 연습은 퍼블릭 서브넷에 기본이 아닌 VPC를 만들고, 귀사의 서브넷에서 인스턴스를 시작하는 절차를 안내해 줍니다.

기본 VPC가 있고 VPC에 대해 추가 구성을 하지 않고 VPC에서 인스턴스를 시작하려면 [기본 VPC로 EC2 인스턴스 시작 \(p. 74\)](#)을 참조하십시오.

Amazon VPC의 기본 시나리오에 대한 자세한 내용은 [시나리오 및 예시 \(p. 17\)](#)을 참조하십시오. VPC와 서브넷을 필요에 맞게 다른 방식으로 구성할 수 있습니다.

다음 표에는 이 서비스를 사용할 때 유용하게 참조할 수 있는 관련 리소스가 나와 있습니다.

Resource	설명
<a href="#">Amazon Virtual Private Cloud(VPC) 연결 오류</a>	네트워크 연결 옵션에 대한 개요를 설명하는 백서입니다.
<a href="#">Amazon VPC forum</a>	Amazon VPC 항목과 관련된 기술적 질문에 대해 토론할 수 있는 커뮤니티 기반 포럼입니다.
<a href="#">AWS 개발자 리소스</a>	설명서, 코드 샘플, 릴리스 정보 및 AWS를 사용하여 혁신적인 애플리케이션을 생성하는 데 도움이 되는 기타 정보를 한 자리에서 찾을 수 있습니다.
<a href="#">AWS Support Center</a>	AWS Support 홈 페이지입니다.
<a href="#">연락처</a>	AWS 결제, 계정 및 이벤트와 관련된 문의를 처리하는 중앙 문의처입니다.

## Amazon VPC를 다른 AWS 서비스와 함께 사용

Amazon VPC는 다른 수많은 AWS 서비스와 통합되며, 더 나아가 어떤 서비스에서는 특정 기능을 수행하기 위해 계정에 VPC가 필요합니다. 다음은 Amazon VPC를 사용하는 서비스의 예입니다.

서비스	관련 항목
AWS Data Pipeline	<a href="#">VPC에서 파이프라인을 위한 리소스 시작</a>

서비스	관련 항목
Amazon EC2	<a href="#">Amazon EC2 및 Amazon VPC</a>
Auto Scaling	<a href="#">Auto Scaling과 Amazon VPC</a>
Elastic Beanstalk	<a href="#">Amazon VPC에서 AWS Elastic Beanstalk 사용</a>
Elastic Load Balancing	<a href="#">Elastic Load Balancing 설정</a>
Amazon ElastiCache	<a href="#">Amazon VPC에서 ElastiCache 사용</a>
Amazon EMR	<a href="#">클러스터의 서브넷 선택</a>
AWS OpsWorks	<a href="#">VPC에서 스택 실행</a>
Amazon RDS	<a href="#">Amazon RDS 및 Amazon VPC</a>
Amazon Redshift	<a href="#">VPC에서 클러스터 관리</a>
Amazon Route 53	<a href="#">프라이빗 호스팅 영역 사용</a>
Amazon WorkSpaces	<a href="#">VPC 생성 및 구성</a>

계정에 있는 VPC, 서브넷 및 기타 VPC 리소스와 상호 간의 관계를 자세히 보려면 AWS Config 서비스를 사용하면 됩니다. 자세한 내용은 [What is AWS Config? 섹션](#)(출처: AWS Config Developer Guide) 단원을 참조하십시오.

## Amazon VPC에 액세스

Amazon VPC는 웹 기반 사용자 인터페이스인 Amazon VPC 콘솔을 제공합니다. AWS 계정에 가입했을 경우 AWS Management Console에 로그인한 후 콘솔 홈 페이지에서 [VPC]를 선택하여 Amazon VPC 콘솔에 액세스할 수 있습니다.

명령줄 인터페이스를 선호하는 고객의 경우 다음과 같은 옵션이 있습니다.

### AWS명령줄 인터페이스(CLI)

다양한 AWS 제품에 사용되며 Windows, Mac, Linux/UNIX에서 지원되는 명령을 제공합니다. 시작하려면 [AWS Command Line Interface 사용 설명서](#) 단원을 참조하십시오. Amazon VPC의 명령에 대한 자세한 내용은 [ec2](#)를 참조하십시오.

### Windows PowerShell용 AWS 도구

PowerShell 환경에서 스크립트 작업을 선호하시는 고객님께서 다양한 AWS 제품을 관리할 수 있도록, 명령줄 도구를 제공합니다. 시작하려면 [Windows PowerShell용 AWS 도구 사용 설명서](#) 단원을 참조하십시오.

Amazon VPC에서는 Query API를 제공합니다. 이 리퀘스트들은, HTTP나 HTTPS의 메시지 교환 방식인 GET이나 POST이며, 미리 정해진 이름인 "Action"을 쿼리 변수로 사용합니다. Amazon VPC에 관련된 API 작업에 대한 자세한 내용은 [작업](#)(Amazon EC2 API Reference)을 참조하십시오.

HTTP나 HTTPS 리퀘스트를 직접 보내는 대신, 각 언어가 제공하는 고유의 API를 사용하여 애플리케이션을 빌드하는 것을 선호하는 개발자 고객님을 위해 AWS는, 라이브러리, 샘플 코드, 자습서 및 기타 리소스를 제공합니다. 이러한 라이브러리는 요청에 암호화 서명, 요청 재시도, 오류 응답 처리 등과 같은 작업을 자동으로 관리하는 기본 기능을 제공하므로, 쉽게 시작할 수 있습니다. AWS SDK를 다운로드하는 방법은 [AWS SDK 및 도구](#)를 참조하십시오.

## Amazon VPC 가격

Amazon VPC 사용에 따르는 추가 요금은 없습니다. 사용하는 인스턴스와 기타 Amazon EC2 기능에 대한 표준 요금만 지불하면 됩니다. 하드웨어 VPN 연결 및 NAT 게이트웨이를 사용하면 요금이 부과됩니다. 자세한 내용은 [Amazon VPC 요금](#) 및 [Amazon EC2 요금](#)을 참조하십시오.

## Amazon VPC 제한

프로비저닝할 수 있는 Amazon VPC 구성 요소의 수에는 제한이 있습니다. 사용자는 이러한 제한을 높이도록 요청할 수 있습니다. 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

## PCI DSS 준수

Amazon VPC에서는 전자 상거래 웹사이트 운영자 또는 서비스 공급자에 의한 신용 카드 데이터의 처리, 저장 및 전송을 지원하며, Payment Card Industry(PCI) Data Security Standard(DSS) 준수를 검증 받았습니다. AWS PCI 규정 준수 패키지의 사본을 요청하는 방법 등 PCI DSS에 대해 자세히 알아보려면 [PCI DSS 레벨 1](#)을 참조하십시오.

# 시작하기

다음 주제는 기본이 아닌 VPC를 신속하게 설정하는 데 도움이 됩니다. 기본 VPC가 이미 있으며 이 VPC에서 인스턴스를 시작하려는 경우(새로운 VPC를 만들거나 구성하지 않을 경우), [기본 VPC로 EC2 인스턴스 시작](#) 단원을 참조하십시오.

VPC의 리소스가 IPv6를 통해 통신하도록 하고 싶다면 연결된 IPv6 CIDR 블록이 있는 VPC를 설정할 수 있습니다.

## 항목

- [Amazon VPC 시작하기 \(p. 6\)](#)
- [Amazon VPC용 IPv6 시작하기 \(p. 12\)](#)

## Amazon VPC 시작하기

이 연습에서는 IPv4 CIDR 블록이 있는 VPC와 IPv4 CIDR 블록이 있는 서브넷을 만든 후, 해당 서브넷에서 퍼블릭 인스턴스를 시작합니다. 인스턴스는 인터넷과 통신할 수 있으며, SSH(Linux 인스턴스인 경우) 또는 원격 데스크톱(Windows 인스턴스인 경우)을 사용하여 로컬 컴퓨터에서 이러한 인스턴스에 액세스할 수 있어야 합니다. 실제 환경에서는 이 시나리오를 사용하여 블로그 호스팅과 같은 퍼블릭 웹 서버를 만들 수 있습니다.

### Note

이 연습은 기본이 아닌 VPC를 직접 신속하게 설정하는 과정을 안내해 줍니다. 기본 VPC가 이미 있으며 이 VPC에서 인스턴스를 시작하려는 경우(새로운 VPC를 만들거나 구성하지 않을 경우), [기본 VPC로 EC2 인스턴스 시작](#) 단원을 참조하십시오. IPv6를 지원하는 기본이 아닌 VPC의 설정을 시작하려면 [Getting Started with IPv6 for Amazon VPC](#) 단원을 참조하십시오.

이 연습을 완료하려면 다음 작업을 수행하십시오.

- 단일 퍼블릭 서브넷이 포함된 기본이 아닌 VPC를 만듭니다. 서브넷을 사용하면 보안 및 운영상의 필요에 따라 인스턴스를 그룹화할 수 있습니다. 퍼블릭 서브넷은 인터넷 게이트웨이를 통해 인터넷에 액세스할 수 있는 서브넷입니다.
- 특정 포트를 통해서만 트래픽을 허용하는 인스턴스의 보안 그룹을 만듭니다.
- Amazon EC2 인스턴스를 서브넷에서 시작합니다.

- 인스턴스와 탄력적 IP 주소 연결. 이렇게 하면 인스턴스가 인터넷에 액세스할 수 있습니다.

Amazon VPC를 처음 사용할 경우, 먼저 Amazon Web Services (AWS)에 가입해야 합니다. 가입 시 AWS 계정은 Amazon VPC를 포함해 AWS의 모든 서비스에 자동으로 등록됩니다. AWS 계정을 아직 만들지 않은 경우 <http://aws.amazon.com>으로 이동한 후 무료 계정 생성을 선택합니다.

#### Note

이 연습에서는 해당 계정에서 EC2-VPC 플랫폼만 지원하는 것으로 가정합니다. 계정이 이전 EC2-Classic 플랫폼도 지원할 경우 이 연습의 단계를 그대로 진행해도 됩니다. 하지만 이 경우 해당 계정에 기본이 아닌 VPC와 비교할 기본 VPC가 없습니다. 자세한 내용은 [지원되는 플랫폼](#)을 참조하십시오.

#### 목차

- 1단계: VPC 생성 (p. 7)
- 2단계: 보안 그룹 만들기 (p. 9)
- 3단계: VPC에서 인스턴스 시작 (p. 10)
- 4단계: 인스턴스에 탄력적 IP 주소 할당 (p. 11)
- 5단계: 정리 (p. 11)

## 1단계: VPC 생성

이 단계에서는 Amazon VPC 콘솔의 Amazon VPC 마법사를 사용하여 VPC를 생성합니다. 마법사는 다음 단계를 수행합니다.

- IPv4 CIDR 블록이 /16인 VPC(프라이빗 IP 주소가 65,536개인 네트워크)를 생성합니다. VPC 크기 조정 및 CIDR 표기법에 대한 자세한 내용은 [VPC 단원](#)을 참조하십시오.
- 인터넷 게이트웨이를 VPC에 연결합니다. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이](#)를 참조하십시오.
- VPC에 크기가 /24인 IPv4 서브넷(256개 프라이빗 IP 주소)을 생성합니다.
- 사용자 지정 라우팅 테이블을 만들고 서브넷에 연결하여 서브넷과 인터넷 게이트웨이 간에 트래픽이 전달될 수 있도록 합니다. 라우팅 테이블에 대한 자세한 내용은 [라우팅 테이블 단원](#)을 참조하십시오.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

#### Note

이 연습에서는 VPC 마법사의 첫 번째 시나리오를 다룹니다. 다른 시나리오에 대한 자세한 내용은 [Scenarios for Amazon VPC](#) 단원을 참조하십시오.

Amazon VPC 마법사를 사용하여 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 오른쪽 상단의 탐색 모음에서 VPC를 생성하려는 리전을 기록해 둡니다. 다른 리전의 VPC에서 인스턴스를 시작할 수 없으므로 이 연습의 나머지 부분에서는 같은 리전에서 작업을 계속해야 합니다. 리전에 대한 자세한 내용은 [리전 및 가용 영역](#)을 참조하십시오.
3. 탐색 창에서 [VPC dashboard]를 선택한 다음 [Start VPC Wizard]를 선택합니다.

#### Note

탐색 창에서 [Your VPCs]를 선택하지 마십시오. 이 페이지에서 VPC 마법사에 액세스할 수 없습니다.

4. 첫 번째 옵션인 [VPC with a Single Public Subnet]을 선택한 후 [Select]를 선택합니다.

5. 구성 페이지에서 [VPC name] 필드에 VPC 이름을 입력합니다. 예를 들어, `my-vpc`를 입력하고 [Subnet name] 필드에 서브넷 이름을 입력합니다. 이렇게 하면 VPC와 서브넷을 만든 후 Amazon VPC 콘솔에서 이들을 식별하는 데 도움이 됩니다. 이 연습에서는 페이지 구성 설정의 나머지 부분을 그대로 두고 [Create VPC]를 선택합니다.

(선택 사항) 원할 경우 구성 설정을 다음과 같이 수정한 다음 Create VPC를 선택합니다.

- IPv4 CIDR block에는 VPC(`10.0.0.0/16`)에 사용할 IPv4 주소 범위가 표시되고, Public subnet's IPv4 CIDR 필드에는 서브넷(`10.0.0.0/24`)에 사용할 IPv4 주소 범위가 표시됩니다. 기본 CIDR 범위를 사용하지 않으려는 경우 직접 지정할 수 있습니다. 자세한 내용은 [VPC 및 서브넷 크기 조정](#)을 참조하십시오.
  - 서브넷을 생성할 가용 영역은 [Availability Zone] 목록에서 선택할 수 있습니다. [No Preference]로 두면 AWS가 가용 영역을 선택합니다. 자세한 내용은 [리전 및 가용 영역](#)을 참조하십시오.
  - Service endpoints 섹션에서 같은 리전의 Amazon S3에 VPC 엔드포인트를 생성할 서브넷을 선택할 수 있습니다. 자세한 내용은 [VPC 엔드포인트](#)를 참조하십시오.
  - [Enable DNS hostnames] 옵션을 [Yes]로 설정하면 VPC에서 시작되는 인스턴스가 DNS 호스트 이름을 수신합니다. 자세한 내용은 [VPC에서 DNS 사용하기](#) 섹션을 참조하십시오.
  - [Hardware tenancy] 옵션을 사용하면 VPC에서 시작되는 인스턴스가 공유 하드웨어에서 실행되는지 아니면 전용 하드웨어에서 실행되는지를 선택할 수 있습니다. 전용 테넌시를 선택하면 추가 비용이 발생합니다. 하드웨어 차용에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Dedicated Instances](#)를 참조하십시오.
6. 상태 창에 진행 중인 작업이 표시됩니다. 작업이 끝나면 [OK]를 선택하여 상태 창을 닫습니다.
  7. [Your VPCs] 페이지에 방금 생성했던 VPC 및 기본 VPC가 표시됩니다. 생성했던 VPC는 기본이 아닌 VPC이므로 [Default VPC] 옆에 [No]라고 표시됩니다.

## VPC에 대한 정보 보기

VPC를 생성했으면 서브넷, 인터넷 게이트웨이, 라우팅 테이블의 정보를 볼 수 있습니다. 생성한 VPC에는 두 개의 라우팅 테이블이 있습니다. 하나는 모든 VPC에 기본적으로 들어 있는 기본 라우팅 테이블이고 다른 하나는 마법사를 통해 생성한 사용자 지정 라우팅 테이블입니다. 사용자 지정 라우팅 테이블은 서브넷에 연결되어 있습니다. 즉, 이 테이블의 라우팅이 서브넷의 트래픽 흐름 방식을 결정합니다. VPC에 새 서브넷을 추가할 경우 기본적으로 기본 라우팅 테이블을 사용합니다.

### VPC에 대한 정보 보기

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다. 생성한 VPC의 이름과 ID([Name] 및 [VPC ID] 열 확인)를 기록해 둡니다. 이 정보를 사용하여 VPC와 연결되는 구성 요소를 식별할 수 있습니다.
3. 탐색 창에서 [Subnets]를 선택합니다. VPC를 생성할 때 생성된 서브넷이 콘솔에 표시됩니다. 서브넷을 [Name] 열에서 이름으로 식별하거나, 전 단계에서 얻은 VPC 정보를 참조하여 [VPC] 열에서 살펴볼 수 있습니다.
4. 탐색 창에서 [Internet Gateways]를 선택합니다. [VPC] 열을 참조하여 VPC에 연결된 인터넷 게이트웨이를 확인할 수 있습니다. 이 열에는 VPC의 ID와 이름(해당하는 경우)이 표시됩니다.
5. 탐색 창에서 [Route Tables]를 선택합니다. VPC와 연결된 라우팅 테이블은 2개가 있습니다. 사용자 지정 라우팅 테이블([Main] 열에 [No]라고 표시됨)을 선택한 후 [Routes] 탭을 선택하여 세부 정보 창에서 라우팅 정보를 조회합니다.
  - 테이블의 첫 번째 행은 로컬 경로로 VPC 내에 있는 인스턴스의 통신을 가능하게 합니다. 이 라우팅은 기본적으로 모든 라우팅 테이블에 있으며 삭제할 수 없습니다.
  - 두 번째 행에는 VPC 외부의 IPv4 주소(`0.0.0.0/0`)로 향하는 트래픽이 서브넷에서 인터넷 게이트웨이로 전송될 수 있도록 하기 위해 Amazon VPC 마법사가 추가한 경로가 표시됩니다.
6. 기본 라우팅 테이블을 선택합니다. 기본 라우팅 테이블에는 로컬 경로만 있으며 그 외 다른 경로는 없습니다.

## 2단계: 보안 그룹 만들기

보안 그룹은 가상 방화벽 역할을 하여 관련 인스턴스에 대한 트래픽을 제어합니다. 보안 그룹을 사용하려면 인스턴스로 수신되는 트래픽을 제어할 인바운드 규칙과, 인스턴스에서 발신되는 트래픽을 제어하는 아웃바운드 규칙을 추가합니다. 보안 그룹을 인스턴스와 연결하려면 인스턴스를 시작할 때 보안 그룹을 지정합니다. 보안 그룹에 규칙을 추가 및 삭제할 경우 변경 사항은 보안 그룹과 관련된 인스턴스에 자동으로 적용됩니다.

VPC는 기본 보안 그룹과 함께 제공됩니다. 시작 시 별도의 보안 그룹과 연결되지 않은 모든 인스턴스는 기본 보안 그룹과 연결됩니다. 이 연습에서는 새로운 보안 그룹인 `WebServerSG`를 생성하고, VPC에서 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.

항목

- [WebServerSG 보안 그룹 규칙 \(p. 9\)](#)
- [WebServerSG 보안 그룹 만들기 \(p. 9\)](#)

## WebServerSG 보안 그룹 규칙

다음 표에서는 `WebServerSG` 보안 그룹의 인바운드 규칙과 아웃바운드 규칙을 설명합니다. 인바운드 규칙은 직접 추가합니다. 아웃바운드 규칙은 모든 아웃바운드 통신을 허용하는 기본 규칙이므로, 이 규칙은 직접 추가할 필요가 없습니다.

인바운드			
소스 IP	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 액세스도 모두 허용
0.0.0.0/0	TCP	443	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 액세스도 모두 허용.
홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	홈 네트워크에서 Linux/UNIX 인스턴스로의 인바운드 SSH 액세스를 허용합니다.
홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	홈 네트워크에서 Windows 인스턴스로의 인바운드 RDP 액세스를 허용합니다.
아웃바운드			
목적지 IP	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 아웃바운드 IPv4 통신을 허용하는 기본 아웃바운드 규칙

## WebserverSG 보안 그룹 만들기

Amazon VPC 콘솔을 사용하여 보안 그룹을 만들 수 있습니다.

WebServerSG 보안 그룹을 만들어 규칙을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]을 선택합니다.



4. [Group name] 필드에 보안 그룹의 이름으로 `WebServerSG`를 입력하고 설명을 제공합니다. 필요에 따라 [Name tag] 필드를 사용하여 키가 `Name`인 보안 그룹에 대한 태그를 생성하거나 지정한 값을 사용할 수 있습니다.
5. [VPC] 메뉴에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.
6. 방금 생성한 `WebServerSG` 보안 그룹을 선택합니다. [Group Name] 옆에서 이름을 볼 수 있습니다.
7. [Inbound Rules] 탭에서 [Edit]를 선택하고 다음과 같이 인바운드 트래픽에 대한 규칙을 추가한 후 완료되면 [Save]를 선택합니다.
  - a. [Type] 목록에서 HTTP를 선택한 다음, `0.0.0.0/0`를 [Source] 필드에 입력합니다.
  - b. [Add another rule]을 선택한 후 [Type] 목록에서 [HTTPS]를 선택하고 [Source] 필드에 `0.0.0.0/0`을 입력합니다.
  - c. [Add another rule]을 선택합니다. Linux 인스턴스를 시작할 경우 [Type] 목록에서 [SSH]를 선택합니다. Windows 인스턴스를 시작할 경우에는 [Type] 목록에서 [RDP]를 선택합니다. 네트워크의 공인 IP 주소 범위를 Source(원본) 필드에 입력합니다. 주소 범위를 모르는 경우 이 연습에서 `0.0.0.0/0`을 사용할 수 있습니다.

#### Caution

`0.0.0.0/0`을 사용하는 경우 모든 IP 주소에서 SSH 또는 RDP를 사용하여 인스턴스에 액세스할 수 있습니다. 따라서 연습에서는 잠시 사용해도 되지만 프로덕션 환경에서 사용하는 것은 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

## 3단계: VPC에서 인스턴스 시작

VPC에서 EC2 인스턴스를 시작할 때 해당 인스턴스를 시작할 서브넷을 지정해야 합니다. 이 경우, 생성한 VPC의 퍼블릭 서브넷에서 인스턴스를 시작합니다. Amazon EC2 콘솔에서 Amazon EC2 시작 마법사를 사용하여 인스턴스를 시작합니다.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

VPC에서 EC2 인스턴스를 시작하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 오른쪽 상단의 탐색 모음에서 VPC 및 보안 그룹을 생성했던 리전과 동일한 리전을 선택합니다.
3. 대시보드에서 [Launch Instance]를 선택합니다.
4. 마법사의 첫 페이지에서, 사용하려는 AMI를 선택합니다. 이 연습에서는 Amazon Linux AMI 또는 Windows AMI를 선택하는 것이 좋습니다.
5. [Choose an Instance Type] 페이지에서는 시작할 인스턴스의 하드웨어 구성과 크기를 선택할 수 있습니다. 마법사는 사용자가 선택한 AMI를 기반으로 하여 첫 번째로 사용 가능한 인스턴스 유형을 기본적으로 선택합니다. 기본 선택을 그대로 두고 [Next: Configure Instance Details]를 선택합니다.
6. [Configure Instance Details] 페이지의 [Network] 목록에서 생성한 VPC를 선택하고, [Subnet] 목록에서 서브넷을 선택합니다. 나머지 기본 설정을 그대로 두고 [Add Tags] 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
7. [Add Tags] 페이지에서는 인스턴스에 `Name`을 사용하여 태그를 지정할 수 있습니다(예: `Name=MyWebServer`). 이렇게 하면 인스턴스를 시작한 후 Amazon EC2 콘솔에서 해당 인스턴스를 식별하는 데 도움이 됩니다. 모두 마쳤으면 [Next: Configure Security Group]을 선택합니다.
8. [Configure Security Group] 페이지에서 마법사는 사용자가 인스턴스에 연결할 수 있도록 마법사 시작 x 보안 그룹을 자동으로 정의합니다. 대신, [Select an existing security group] 옵션을 선택하고, 이전에 생성한 [WebServerSG] 그룹을 선택한 후 [Review and Launch]를 선택합니다.
9. [Review Instance Launch] 페이지에서 인스턴스의 세부 정보를 확인한 다음 [Launch]를 선택합니다.
10. Select an existing key pair or create a new key pair(기존 키 쌍 선택 또는 새 키 쌍 만들기) 대화 상자에 서 기존 키 쌍을 선택하거나 새 키 쌍을 만들 수 있습니다. 새 키 페어를 만들 경우, 파일을 다운로드한 후



안전한 위치에 저장해야 합니다. 인스턴스를 실행한 후 인스턴스에 연결하려면 개인 키 콘텐츠가 필요합니다.

인스턴스를 시작하려면 승인 확인란을 선택한 후 [Launch Instances]를 선택합니다.

11. 확인 페이지에서 [View Instances]를 선택하여 [Instances] 페이지에서 해당 인스턴스를 확인합니다. 인스턴스를 선택하고 [Description] 탭에서 세부 정보를 확인합니다. [Private IPs] 필드에는 서브넷의 IP 주소 범위에서 인스턴스에 할당된 프라이빗 IP 주소가 표시됩니다.

Amazon EC2 시작 마법사에서 사용할 수 있는 옵션에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

## 4단계: 인스턴스에 탄력적 IP 주소 할당

이전 단계에서, 인터넷 게이트웨이로 라우팅되는 서브넷인 퍼블릭 서브넷에서 인스턴스를 시작했습니다. 그러나 이 서브넷의 인스턴스는 인터넷과 통신하기 위해 퍼블릭 IPv4 주소도 필요로 합니다. 기본이 아닌 VPC의 인스턴스에는 퍼블릭 IPv4 주소가 할당되지 않도록 기본 설정되어 있습니다. 이 단계에서는 탄력적 IP 주소를 계정에 할당한 후 인스턴스와 연결합니다. 탄력적 IP 주소에 대한 자세한 정보는 [탄력적 IP 주소](#)를 참조하십시오.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

탄력적 IP 주소를 할당 및 지정하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. [Allocate New Address]를 선택한 다음 [Yes, Allocate]를 선택합니다.

### Note

해당 계정이 EC2-Classic을 지원할 경우, 우선 [Network platform] 목록에서 [EC2-VPC]를 선택합니다.

4. 목록에서 탄력적 IP 주소와 [Actions], [Associate Address]를 차례로 선택합니다.
5. 대화 상자의 [Associate with] 목록에서 [Instance]를 선택하고, [Instance] 목록에서 해당 인스턴스를 선택합니다. 작업을 마치면 [Yes, Associate]를 선택합니다.

이제 인스턴스를 인터넷에서 액세스할 수 있습니다. 또한 홈 네트워크에서 SSH 또는 원격 데스크톱을 사용하여 탄력적 IP 주소를 통해 인스턴스에 연결할 수 있습니다. Linux 인스턴스에 연결하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [\[Connecting to Your Linux Instance\]](#)를 참조하십시오. Windows 인스턴스에 연결하는 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [\[Connect to Your Windows Instance Using RDP\]](#)를 참조하십시오.

이것으로 연습을 마쳤습니다. 이제 VPC에서 인스턴스를 계속 사용하거나, 인스턴스가 더 이상 필요하지 않을 경우 인스턴스를 종료하고 탄력적 IP 주소를 릴리스하여 비용이 발생하지 않도록 할 수 있습니다. VPC를 삭제할 수도 있습니다. 단, 서브넷과 라우팅 테이블 등 이 연습에서 만든 VPC 구성 요소 및 VPC에 대해서는 요금이 청구되지 않습니다.

## 5단계: 정리

VPC를 삭제하기 전에 VPC 내에 실행하고 있는 모든 인스턴스를 종료해야 합니다. VPC 콘솔을 사용하여 VPC를 삭제하면 서브넷, 보안 그룹, 네트워크 ACL, DHCP 옵션 세트, 라우팅 테이블, 인터넷 게이트웨이 등 VPC와 관련된 리소스 또한 삭제됩니다.

인스턴스를 종료하려면 탄력적 IP 주소를 릴리스하고 VPC를 삭제하십시오.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]와 [Terminate]를 차례로 선택합니다.
4. 대화 상자에서 [Release attached Elastic IPs] 섹션을 확장하고 탄력적 IP 주소 옆에 있는 확인란을 선택합니다. [Yes, Terminate]를 선택합니다.
5. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
6. 탐색 창에서 Your VPCs를 선택합니다.
7. VPC를 선택하고 [Actions]를 선택한 후 [Delete VPC]를 선택합니다.
8. 확인 메시지가 나타나면 [Yes, Delete]를 선택합니다.

## Amazon VPC용 IPv6 시작하기

이 연습에서는 IPv6 CIDR 블록이 있는 VPC와 IPv6 CIDR 블록이 있는 서브넷을 만든 후, 해당 서브넷에서 퍼블릭 인스턴스를 시작합니다. 인스턴스는 IPv6를 통해 인터넷과 통신할 수 있으며, SSH(Linux 인스턴스인 경우) 또는 원격 데스크톱(Windows 인스턴스인 경우)을 사용하여 로컬 컴퓨터에서 IPv6를 통해 이러한 인스턴스에 액세스할 수 있어야 합니다. 실제 환경에서는 이 시나리오를 사용하여 블로그 호스팅과 같은 퍼블릭 웹 서버를 만들 수 있습니다.

이 연습을 완료하려면 다음 작업을 수행하십시오.

- IPv6 CIDR 블록과 단일 퍼블릭 서브넷이 포함된 기본이 아닌 VPC를 만듭니다. 서브넷을 사용하면 보안 및 운영상의 필요에 따라 인스턴스를 그룹화할 수 있습니다. 퍼블릭 서브넷은 인터넷 게이트웨이를 통해 인터넷에 액세스할 수 있는 서브넷입니다.
- 특정 포트를 통해서만 트래픽을 허용하는 인스턴스의 보안 그룹을 만듭니다.
- 서브넷에서 Amazon EC2 인스턴스를 실행하고, 시작하는 과정에서 IPv6 주소를 인스턴스에 연결합니다. IPv6 주소는 전역적으로 고유하며 인스턴스가 인터넷과 통신할 수 있게 해줍니다.

IPv4 및 IPv6 주소 지정에 대한 자세한 내용은 [VPC의 IP 주소 지정](#)을 참조하십시오.

Amazon VPC를 처음 사용할 경우, 먼저 Amazon Web Services (AWS)에 가입해야 합니다. 가입 시 AWS 계정은 Amazon VPC를 포함해 AWS의 모든 서비스에 자동으로 등록됩니다. AWS 계정을 아직 만들지 않은 경우, <http://aws.amazon.com>으로 이동한 후 Create a Free Account를 선택합니다.

목차

- 1단계: VPC 생성 (p. 12)
- 2단계: 보안 그룹 만들기 (p. 14)
- 3단계: 인스턴스 시작 (p. 15)

### 1단계: VPC 생성

이 단계에서는 Amazon VPC 콘솔의 Amazon VPC 마법사를 사용하여 VPC를 생성합니다. 마법사는 다음 단계를 수행합니다.

- IPv4 CIDR 블록이 /16인 VPC를 생성하여 크기가 /56인 IPv6 CIDR 블록을 VPC와 연결합니다. 자세한 내용은 [VPC 및 서브넷](#) 단원을 참조하십시오. IPv6 CIDR 블록의 크기는 고정되어 있고(/56) IPv6 주소의 범위는 Amazon의 IPv6 주소 풀에서 자동으로 할당됩니다(범위를 직접 선택할 수는 없음).
- 인터넷 게이트웨이를 VPC에 연결합니다. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이](#)를 참조하십시오.
- VPC에 IPv4 CIDR 블록이 /24이고 IPv6 CIDR 블록이 /64인 서브넷을 생성합니다. IPv6 CIDR 블록의 크기는 고정되어 있습니다(/64).
- 사용자 지정 라우팅 테이블을 만들고 서브넷에 연결하여 서브넷과 인터넷 게이트웨이 간에 트래픽이 전달될 수 있도록 합니다. 라우팅 테이블에 대한 자세한 내용은 [라우팅 테이블](#) 단원을 참조하십시오.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

#### Note

이 연습에서는 VPC 마법사의 첫 번째 시나리오를 다룹니다. 다른 시나리오에 대한 자세한 내용은 [Scenarios for Amazon VPC](#) 단원을 참조하십시오.

Amazon VPC 마법사를 사용하여 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 오른쪽 상단의 탐색 모음에서 VPC를 생성하려는 리전을 기록해 둡니다. 다른 리전의 VPC에서 인스턴스를 시작할 수 없으므로 이 연습의 나머지 부분에서는 같은 리전에서 작업을 계속해야 합니다. 리전에 대한 자세한 내용은 [리전 및 가용 영역](#)을 참조하십시오.
3. 탐색 창에서 VPC dashboard를 선택한 다음, Start VPC Wizard를 선택합니다.

#### Note

탐색 창에서 Your VPCs를 선택하지 마십시오. 이 페이지에서 VPC 마법사에 액세스할 수 없습니다.

4. 첫 번째 옵션인 VPC with a Single Public Subnet을 선택한 후 Select를 선택합니다.
5. 구성 페이지에서 VPC name에 VPC 이름을 입력합니다. 예를 들면, my-vpc를 입력하고 Subnet name에 서브넷 이름을 입력합니다. 이렇게 하면 VPC와 서브넷을 만든 후 Amazon VPC 콘솔에서 이들을 식별하는 데 도움이 됩니다.
6. IPv4 CIDR block에 대해서는 기본 설정(10.0.0.0/16)을 그대로 두거나 원하는 대로 설정할 수 있습니다. 자세한 내용은 [VPC 및 서브넷 크기](#) 단원을 참조하십시오.  
  
IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다.
7. Public subnet's IPv4 CIDR에 대해서는 기본 설정을 그대로 두거나 원하는 대로 설정할 수 있습니다. Public subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 기본 16진수 페어 값을 그대로 둘 수 있습니다(00).
8. 페이지에서 기본 구성의 나머지 부분은 그대로 두고 Create VPC를 선택합니다.
9. 상태 창에 진행 중인 작업이 표시됩니다. 작업이 끝나면 [OK]를 선택하여 상태 창을 닫습니다.
10. Your VPCs 페이지에 방금 생성했던 VPC 및 기본 VPC가 표시됩니다.

## VPC에 대한 정보 보기

VPC를 생성했으면 서브넷, 인터넷 게이트웨이, 라우팅 테이블에 대한 정보를 볼 수 있습니다. 생성한 VPC에는 두 개의 라우팅 테이블이 있습니다. 하나는 모든 VPC에 기본적으로 들어 있는 기본 라우팅 테이블이고 다른 하나는 마법사를 통해 생성한 사용자 지정 라우팅 테이블입니다. 사용자 지정 라우팅 테이블은 서브넷에 연결되어 있습니다. 즉, 이 테이블의 라우팅이 서브넷의 트래픽 흐름 방식을 결정합니다. VPC에 새 서브넷을 추가할 경우 기본값으로 기본 라우팅 테이블을 사용합니다.

VPC에 대한 정보 보기

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다. 생성한 VPC의 이름과 ID([Name] 및 [VPC ID] 열 확인)를 기록해 둡니다. 이 정보를 사용하여 VPC와 연결된 구성 요소를 식별할 수 있습니다.
3. 탐색 창에서 [Subnets]를 선택합니다. VPC를 생성할 때 생성된 서브넷이 콘솔에 표시됩니다. 서브넷을 [Name] 열에서 이름으로 식별하거나, 전 단계에서 얻은 VPC 정보를 참조하여 [VPC] 열에서 살펴볼 수 있습니다.
4. 탐색 창에서 [Internet Gateways]를 선택합니다. [VPC] 열을 참조하여 VPC에 연결된 인터넷 게이트웨이를 확인할 수 있습니다. 이 열에는 VPC의 ID와 이름(해당하는 경우)이 표시됩니다.
5. 탐색 창에서 [Route Tables]를 선택합니다. VPC와 연결된 라우팅 테이블은 2개가 있습니다. 사용자 지정 라우팅 테이블([Main] 열에 [No]라고 표시됨)을 선택한 후 [Routes] 탭을 선택하여 세부 정보 창에서 라우팅 정보를 조회합니다.

- 테이블의 첫 번째 두 행은 로컬 경로로서, VPC 내에 있는 인스턴스가 IPv4 및 IPv6를 통해 통신할 수 있게 해줍니다. 이 경로는 제거할 수 없습니다.
  - 그 다음 행에는 VPC 외부의 IPv4 주소(0.0.0.0/0)로 향하는 트래픽이 서브넷에서 인터넷 게이트웨이로 전송될 수 있도록 하기 위해 Amazon VPC 마법사가 추가한 경로가 표시됩니다.
  - 그 다음 행에는 VPC 외부의 IPv6 주소(:::0)로 향하는 트래픽이 서브넷에서 인터넷 게이트웨이로 전송되게 해주는 경로가 표시됩니다.
6. 기본 라우팅 테이블을 선택합니다. 기본 라우팅 테이블에는 로컬 경로만 있으며 그 외 다른 경로는 없습니다.

## 2단계: 보안 그룹 만들기

보안 그룹은 가상 방화벽 역할을 하여 관련 인스턴스에 대한 트래픽을 제어합니다. 보안 그룹을 사용하려면 인스턴스로 수신되는 트래픽을 제어할 인바운드 규칙과, 인스턴스에서 발신되는 트래픽을 제어하는 아웃바운드 규칙을 추가합니다. 보안 그룹을 인스턴스와 연결하려면 인스턴스를 시작할 때 보안 그룹을 지정합니다.

VPC는 기본 보안 그룹과 함께 제공됩니다. 시작 시 별도의 보안 그룹과 연결되지 않은 모든 인스턴스는 기본 보안 그룹과 연결됩니다. 이 연습에서는 새로운 보안 그룹인 `WebServerSG`를 생성하고, VPC에서 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.

항목

- [WebServerSG 보안 그룹 규칙 \(p. 14\)](#)
- [WebserverSG 보안 그룹 만들기 \(p. 15\)](#)

### WebServerSG 보안 그룹 규칙

다음 표에서는 `WebServerSG` 보안 그룹의 인바운드 규칙과 아웃바운드 규칙을 설명합니다. 인바운드 규칙은 직접 추가합니다. 아웃바운드 규칙은 모든 아웃바운드 통신을 허용하는 기본 규칙이므로, 이 규칙은 직접 추가할 필요가 없습니다.

인바운드			
소스 IP	프로토콜	포트 범위	설명
:::0	TCP	80	모든 IPv6 주소에서 이루어지는 인바운드 HTTP 액세스를 허용
:::0	TCP	443	모든 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽을 허용
홈 네트워크의 IPv6 주소 범위	TCP	22 또는 3389	홈 네트워크에서 IPv6 주소 범위로부터 Linux/UNIX 인스턴스로 이루어지는 인바운드 SSH 액세스(포트 22) 허용. 인스턴스가 Windows 인스턴스인 경우, RDP 액세스(포트 3389)를 허용하는 규칙이 필요합니다.
아웃바운드			
목적지 IP	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 아웃바운드 IPv4 통신을 허용하는 기본 아웃바운드 규칙. 이 규칙은 연습 용도이므로 수정할 필요가 없음.

::/0	모두	모두	모든 아웃바운드 IPv6 통신을 허용하는 기본 아웃바운드 규칙. 이 규칙은 연습 용도이므로 수정할 필요가 없음.
------	----	----	--

#### Note

IPv4 트래픽에 대해 웹 서버 인스턴스를 사용하려면, IPv4를 통한 액세스를 가능케 하는 규칙을 추가해야 합니다. 이 경우 모든 IPv4 주소(0.0.0.0/0)에서 발신되는 HTTP 및 HTTPS 트래픽과 홈 네트워크의 IPv4 주소 범위로부터 이루어지는 SSH/RDP 액세스를 말합니다.

## WebserverSG 보안 그룹 만들기

Amazon VPC 콘솔을 사용하여 보안 그룹을 만들 수 있습니다.

WebServerSG 보안 그룹을 만들어 규칙을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security Groups와 Create Security Group을 차례로 선택합니다.
3. Group name에 보안 그룹의 이름인 WebServerSG를 입력하고 설명을 제공합니다. 필요에 따라 [Name tag] 필드를 사용하여 키가 Name인 보안 그룹에 대한 태그를 생성하거나 지정한 값을 사용할 수 있습니다.
4. [VPC] 메뉴에서 VPC의 ID를 선택한 다음 [Yes, Create]를 선택합니다.
5. 방금 생성한 WebServerSG 보안 그룹을 선택합니다. [Group Name] 열에서 이름을 볼 수 있습니다.
6. Inbound Rules 탭에서 Edit를 선택하고, 다음과 같이 인바운드 트래픽에 대한 규칙을 추가한 후 완료되면 Save를 선택합니다.
  - a. Type에서 HTTP를 선택한 다음, ::/0를 Source 필드에 입력합니다.
  - b. Add another rule을 선택한 후 Type에서 HTTPS를 선택하고 Source 필드에 ::/0을 입력합니다.
  - c. [Add another rule]을 선택합니다. Linux 인스턴스를 시작할 경우, Type에서 SSH를 선택합니다. Windows 인스턴스를 시작할 경우에는 RDP를 선택합니다. Source 필드에 네트워크의 퍼블릭 IPv6 주소 범위를 입력합니다. 주소 범위를 모르는 경우 이 연습에서 ::/0을 사용할 수 있습니다.

#### Caution

::/0을 사용하는 경우, 모든 IPv6 주소에서 SSH 또는 RDP를 사용하여 인스턴스에 액세스할 수 있습니다. 따라서 연습에서는 잠시 사용해도 되지만 프로덕션 환경에서 사용하는 것은 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

## 3단계: 인스턴스 시작

VPC에서 EC2 인스턴스를 시작할 때 해당 인스턴스를 시작할 서브넷을 지정해야 합니다. 이 경우, 생성한 VPC의 퍼블릭 서브넷에서 인스턴스를 시작합니다. Amazon EC2 콘솔에서 Amazon EC2 시작 마법사를 사용하여 인스턴스를 시작합니다.

인스턴스를 인터넷에서 접속할 수 있도록 하려면 시작하는 과정에서 서브넷 범위에 속하는 IPv6 주소를 인스턴스에 지정해야 합니다. 이렇게 하면 인스턴스가 IPv6를 통해 인터넷과 통신할 수 있습니다.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

VPC에서 EC2 인스턴스를 시작하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 오른쪽 상단의 탐색 모음에서 VPC 및 보안 그룹을 생성했던 리전과 동일한 리전을 선택합니다.

3. 대시보드에서 [Launch Instance]를 선택합니다.
4. 마법사의 첫 페이지에서, 사용하려는 AMI를 선택합니다. 이 연습에서는 Amazon Linux AMI 또는 Windows AMI를 선택하는 것이 좋습니다.
5. [Choose an Instance Type] 페이지에서는 시작할 인스턴스의 하드웨어 구성과 크기를 선택할 수 있습니다. 마법사는 사용자가 선택한 AMI를 기반으로 하여 첫 번째로 사용 가능한 인스턴스 유형을 선택하도록 기본 설정되어 있습니다. 그 기본 선택을 그대로 두고 Next: Configure Instance Details를 선택합니다.
6. Configure Instance Details 페이지의 Network 목록에서 생성한 VPC를 선택하고, Subnet 목록에서 서브넷을 선택합니다.
7. Auto-assign IPv6 IP에 대해 Enable을 선택합니다.
8. 나머지 기본 설정을 그대로 두고 [Add Tags] 페이지에 도달할 때까지 마법사의 다음 페이지로 이동합니다.
9. [Add Tags] 페이지에서는 인스턴스에 Name을 사용하여 태그를 지정할 수 있습니다(예: Name=MyWebServer). 이렇게 하면 인스턴스를 시작한 후 Amazon EC2 콘솔에서 해당 인스턴스를 식별하는 데 도움이 됩니다. 모두 마쳤으면 [Next: Configure Security Group]을 선택합니다.
10. [Configure Security Group] 페이지에서 마법사는 사용자가 인스턴스에 연결할 수 있도록 마법사 시작 x 보안 그룹을 자동으로 정의합니다. 대신, [Select an existing security group] 옵션을 선택하고, 이전에 생성한 [WebServerSG] 그룹을 선택한 후 [Review and Launch]를 선택합니다.
11. Review Instance Launch 페이지에서 인스턴스의 세부 정보를 확인한 다음 Launch를 선택합니다.
12. Select an existing key pair or create a new key pair(기존 키 쌍 선택 또는 새 키 쌍 만들기) 대화 상자에서 기존 키 쌍을 선택하거나 새 키 쌍을 만들 수 있습니다. 새 키 페어를 만들 경우, 파일을 다운로드한 후 안전한 위치에 저장해야 합니다. 인스턴스를 실행한 후 인스턴스에 연결하려면 개인 키 콘텐츠가 필요합니다.

인스턴스를 시작하려면 승인 확인란을 선택한 후 Launch Instances를 선택합니다.

13. 확인 페이지에서 [View Instances]를 선택하여 [Instances] 페이지에서 해당 인스턴스를 확인합니다. 인스턴스를 선택하고 [Description] 탭에서 세부 정보를 확인합니다. Private IPs 필드에는 서브넷의 IPv4 주소 범위에서 인스턴스에 할당된 프라이빗 IPv4 주소가 표시됩니다. IPv6 IPs 필드에는 서브넷의 IPv6 주소 범위에서 인스턴스에 할당된 IPv6 주소가 표시됩니다.

Amazon EC2 시작 마법사에서 사용할 수 있는 옵션에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

홈 네트워크에서 SSH 또는 원격 데스크톱을 사용하여 IPv6 주소를 통해 인스턴스에 접속할 수 있습니다. Linux 인스턴스에 연결하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [\[Connecting to Your Linux Instance\]](#)를 참조하십시오. Windows 인스턴스에 연결하는 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [\[Connect to Your Windows Instance Using RDP\]](#)를 참조하십시오.

#### Note

인터넷, SSH 또는 RDP를 통해 IPv4 주소로 인스턴스에 액세스할 수 있게 하고 싶다면, 인스턴스에 탄력적 IP 주소(고정 퍼블릭 IPv4 주소)를 연결하고 IPv4를 통해 액세스를 허용하도록 보안 그룹 규칙을 조정해야 합니다. 이렇게 하려면 [시작하기 \(p. 6\)](#)의 절차를 따르십시오.

## 시나리오 및 예시

이 단원에서는 Amazon VPC 콘솔에서 VPC 마법사를 사용하는 방법에 대한 시나리오를 비롯한 VPC 생성 및 구성의 예를 제시합니다.

시나리오	사용량
시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC (p. 17)	블로그나 간단한 웹 사이트 같은 단일 티어의 퍼블릭 웹 애플리케이션을 실행하기 위해 VPC 마법사를 사용하여 VPC를 생성합니다.
시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC(NAT) (p. 23)	퍼블릭 웹 애플리케이션을 실행하기 위해 VPC 마법사를 사용하여 VPC를 생성하는 한편, 두 번째 서브넷에서 공개적으로 액세스가 불가능한 백 엔드 서버는 계속 유지합니다.
시나리오 3: 공인 및 사설 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC (p. 34)	데이터 센터를 클라우드로 확장하기 위해 VPC 마법사를 사용하여 VPC를 생성하고 VPC에서 직접 인터넷에 액세스합니다.
시나리오 4: 사설 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC (p. 44)	데이터 센터를 클라우드로 확장하기 위해 VPC 마법사를 사용하여 VPC를 생성하고 네트워크가 인터넷에 노출되지 않는 Amazon의 인프라를 최대한 활용합니다.
예: AWS CLI를 사용하여 IPv4 VPC 및 서브넷 생성 (p. 49)	AWS CLI를 사용하여 VPC와 퍼블릭 및 프라이빗 서브넷을 만듭니다.
예: AWS CLI를 사용하여 IPv6 VPC 및 서브넷 생성 (p. 54)	AWS CLI를 사용하여 연결된 IPv6 CIDR 블록이 있는 VPC를 만들고, 연결된 IPv6 CIDR 블록이 있는 퍼블릭 및 프라이빗 서브넷을 만듭니다.

## 시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC

이 시나리오의 구성에는 단일 퍼블릭 서브넷을 가진 Virtual Private Cloud(VPC)와 인터넷을 통해 통신할 수 있게 해주는 인터넷 게이트웨이가 포함됩니다. 블로그나 간단한 웹 사이트 같은 단일 티어의 퍼블릭 웹 애플리케이션을 실행해야 할 경우 이 구성을 권장합니다.



이 단원에서는 Amazon VPC 콘솔의 VPC 마법사를 사용하여 VPC를 생성한다고 가정합니다.

이 시나리오를 IPv6에 맞게 구성할 수도 있습니다. 즉 VPC 마법사를 이용해 VPC, 그리고 연결된 IPv6 CIDR 블록이 있는 서브넷을 만들 수 있습니다. 퍼블릭 서브넷에서 시작한 인스턴스는 IPv6 주소를 받아 IPv6를 사용해 통신할 수 있습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

#### 항목

- [개요 \(p. 18\)](#)
- [라우팅 \(p. 19\)](#)
- [보안 \(p. 19\)](#)
- [시나리오 1 구현 \(p. 21\)](#)

## 개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.

#### Note

[시작하기 \(p. 6\)](#) 연습을 완료했다면 Amazon VPC 콘솔에서 VPC를 사용하여 이 시나리오를 이미 구현한 것입니다.

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

- IPv4 CIDR 블록의 크기가 /16(예: 10.0.0.0/16)인 Virtual Private Cloud(VPC). 이것은 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- IPv4 CIDR 블록의 크기가 /24(예: 10.0.0.0/24)인 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다.
- 인터넷 게이트웨이. VPC를 인터넷 및 다른 AWS 서비스에 연결합니다.
- 인스턴스가 VPC의 다른 인스턴스와 통신할 수 있게 해주는 서브넷 범위(예: 10.0.0.6)의 프라이빗 IPv4 주소와 인터넷에서 인스턴스에 액세스할 수 있게 해주는 퍼블릭 IPv4 주소인 탄력적 IPv4 주소(예: 198.51.100.2)가 있는 인스턴스.
- 서브넷과 연결된 사용자 정의 라우팅 테이블. 라우팅 테이블 항목은 서브넷의 인스턴스가 IPv4를 사용하여 VPC 내의 다른 서브넷과 통신할 수 있도록 해주며, 또한 인터넷을 통해 직접 통신할 수 있게 해줍니다. 인터넷 게이트웨이로 이어지는 경로가 있는 라우팅 테이블과 연결된 서브넷을 퍼블릭 서브넷이라고 합니다.

서브넷에 대한 자세한 정보는 [VPC 및 서브넷 \(p. 62\)](#) 단원을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이 \(p. 165\)](#)를 참조하십시오.

## IPv6 개요

이 시나리오에 IPv6를 사용할 수도 있습니다. 위에 나열된 구성 요소뿐 아니라 다음 요소도 구성에 포함됩니다.

- VPC와 연결된 /56 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/56). Amazon은 CIDR을 자동 할당하므로 범위를 직접 선택할 수는 없습니다.
- 퍼블릭 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. 서브넷 IPv6 CIDR 블록의 크기는 선택할 수 없습니다.
- 서브넷 범위에서 인스턴스에 할당된 IPv6 주소(예: 2001:db8:1234:1a00::123).
- VPC의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있도록 해주고, 또한 인터넷을 통해 직접 통신할 수 있게 해주는 사용자 정의 라우팅 테이블의 라우팅 테이블 항목.



## 라우팅

VPC에는 라우터가 내재되어 있습니다(위 구성 다이어그램 참조). 이 시나리오에서 VPC 마법사는 VPC 외부 주소로 경로가 지정된 모든 트래픽을 인터넷 게이트웨이로 라우팅하는 사용자 정의 라우팅 테이블을 생성하고, 이 라우팅 테이블을 서브넷과 연결합니다.

다음 표는 위 구성 다이어그램의 예시에 대한 라우팅 테이블을 보여줍니다. 첫 번째 항목은 VPC의 로컬 IPv4 라우팅에 대한 기본 항목으로서, 이 VPC의 인스턴스가 서로 통신할 수 있게 해줍니다. 두 번째 항목은 기타 IPv4 서브넷 트래픽을 모두 인터넷 게이트웨이(예: `igw-1a2b3c4d`)로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
0.0.0.0/0	igw-id

## IPv6에 대한 라우팅

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다. 다음 표는 VPC에서 IPv6 통신을 사용하기로 한 경우, 이 시나리오에 대한 사용자 지정 라우팅 테이블을 정리한 것입니다. 두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다. 네 번째 항목에서는 기타 IPv6 서브넷 트래픽을 모두 인터넷 게이트웨이로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	igw-id
::/0	igw-id

## 보안

AWS는 VPC의 보안을 강화하기 위해 사용할 수 있는 두 가지 기능, 보안 그룹과 네트워크 ACL을 제공합니다. 보안 그룹은 인스턴스용 인바운드 및 아웃바운드 트래픽을 제어하고, 네트워크 ACL은 서브넷용 인바운드 및 아웃바운드 트래픽을 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 맞출 수 있지만, 원하는 경우 네트워크 ACL을 사용하여 VPC에 보안 계층을 더 추가할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

이 시나리오의 경우, 네트워크 ACL이 아니라 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 1을 위한 권장 규칙 \(p. 115\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 96\)](#)과 함께 제공됩니다. 인스턴스를 시작하는 동안 다른 보안 그룹을 지정하지 않는 경우, VPC에서 시작되는 인스턴스는 적절한 기본 보안 그룹과 자동 연결됩니다. 기본 보안 그룹에 규칙을 추가할 수 있지만, 그 규칙이 VPC에서 시작하는 다른 인스턴스에 적합하지 않을 수 있습니다. 그 대신 웹 서버용 사용자 지정 보안 그룹을 생성하는 것이 좋습니다.

이 시나리오의 경우, `WebServerSG`라는 보안 그룹을 생성하십시오. 생성된 보안 그룹에는 인스턴스에서 나가는 모든 트래픽을 허용하는 하나의 아웃바운드 규칙만 있습니다. 필요에 따라 인바운드 트래픽을 사용하고 아웃바운드 트래픽을 제한하려면 규칙을 수정해야 합니다. VPC에서 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.

다음은 WebServerSG 보안 그룹에 대한 IPv4 트래픽의 인바운드 및 아웃바운드 규칙입니다.

인바운드			
소스	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	(Linux 인스턴스) IPv4를 통한 네트워크로부터의 인바운드 SSH 액세스 허용. <a href="http://checkip.amazonaws.com">http://checkip.amazonaws.com</a> 과 같은 서비스를 사용하여 로컬 컴퓨터의 퍼블릭 IPv4 주소를 얻을 수 있습니다. 고정 IP 주소 없이 ISP 또는 방화벽을 경유하여 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 알아내야 합니다.
네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	(Windows 인스턴스) IPv4를 통한 네트워크로부터의 인바운드 RDP 액세스 허용
보안 그룹 ID(sg-xxxxxxx)	모두	모두	(선택 사항) 이 보안 그룹과 연결된 다른 인스턴스의 인바운드 트래픽 허용. 이 규칙은 VPC용 기본 보안 그룹에 자동으로 추가됩니다. 생성한 사용자 지정 보안 그룹에 대해서는 이러한 유형의 통신을 허용하는 규칙을 수동으로 추가해야 합니다.
아웃바운드(선택 사항)			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 IPv4 주소에 대한 아웃바운드 액세스를 모두 허용하는 기본 규칙. 웹 서버가 예를 들어 소프트웨어 업데이트 등을 받기 위해 아웃바운드 트래픽을 시작하게 하려면 기본 아웃바운드 규칙을 그대로 둡니다. 아니면 이 규칙을 제거할 수 있습니다.

## IPv6의 보안

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 보안 그룹에 별도의 규칙을 추가하여 웹 서버 인스턴스에 대한 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다. 이 시나리오에서 웹 서버는 IPv6를 통해 인터넷 트래픽을 모두 수신할 수 있고, IPv6를 통해 로컬 네트워크로부터 SSH 또는 RDP 트래픽을 수신할 수 있습니다.

다음은 WebServerSG 보안 그룹에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

인바운드			
소스	프로토콜	포트 범위	설명

::/0	TCP	80	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용
::/0	TCP	443	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
네트워크의 IPv6 주소 범위	TCP	22	(Linux 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 SSH 액세스 허용
네트워크의 IPv6 주소 범위	TCP	3389	(Windows 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 RDP 액세스 허용
아웃바운드(선택 사항)			
목적지	프로토콜	포트 범위	설명
::/0	모두	모두	모든 IPv6 주소에 대한 아웃바운드 액세스를 모두 허용하는 기본 규칙. 웹 서버가 예를 들어 소프트웨어 업데이트 등을 받기 위해 아웃바운드 트래픽을 시작하게 하려면 기본 아웃바운드 규칙을 그대로 둡니다. 아니면 이 규칙을 제거할 수 있습니다.

## 시나리오 1 구현

시나리오 1을 구현하려면 VPC 마법사를 사용하여 VPC를 생성하고 WebServerSG 보안 그룹을 생성 및 구성한 다음, VPC에서 인스턴스를 시작합니다.

이 절차에는 VPC용 IPv6 통신을 활성화 및 구성하기 위한 옵션 절차가 포함됩니다. VPC에서 IPv6를 사용하고 싶지 않다면 이 절차를 수행할 필요가 없습니다.

VPC를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. 첫 번째 옵션인 [VPC with a Single Public Subnet]을 선택한 후 [Select]를 선택합니다.
4. VPC name과 Public subnet name에 대해 VPC와 서브넷에 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다. VPC 및 서브넷에 대해 고유한 IPv4 CIDR 블록 범위를 지정하거나 기본값(10.0.0.0/16 및 10.0.0.0/24)을 그대로 둘 수 있습니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. Public subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택하고 서브넷에 16진수 페어 값을 지정하거나 기본값(00)을 그대로 둡니다.
6. 기본 설정의 나머지 부분은 그대로 두고 Create VPC를 선택합니다.

WebServerSG 보안 그룹을 만들려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]을 선택합니다.
4. 보안 그룹 이름으로 WebServerSG를 지정하고 설명을 제공합니다. [VPC] 메뉴에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.

5. 앞에서 만든 WebServerSG 보안 그룹을 선택합니다. 세부 정보 창은 인바운드 규칙 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 정보 탭을 포함합니다.
6. [Inbound Rules] 탭에서 [Edit]를 선택하고 다음을 수행합니다.
  - [Type] 목록에서 HTTP를 선택한 다음, 0.0.0.0/0를 [Source] 필드에 입력합니다.
  - [Add another rule]을 선택한 후 [Type] 목록에서 [HTTPS]를 선택하고 [Source] 필드에 0.0.0.0/0을 입력합니다.
  - [Add another rule]을 선택한 다음, [Type] 목록에서 [SSH] (Linux용) 또는 [RDP](Windows용)를 선택합니다. 네트워크의 공인 IP 주소 범위를 Source(원본) 필드에 입력합니다. (이 주소 범위를 모를 경우, 테스트 용도로 0.0.0.0/0을 사용할 수 있습니다. 실제 환경에서는 특정 IP 주소나 주소 범위만 해당 인스턴스에 액세스할 수 있도록 허가합니다.)
  - (선택 사항) [Add another rule]을 선택한 다음, [Type] 목록에서 [ALL traffic]을 선택합니다. [Source] 필드에 WebServerSG 보안 그룹의 ID를 입력합니다.
  - (선택 사항, IPv6 전용) Add another rule을 선택한 후 Type 목록에서 HTTP를 선택하고 Source 필드에 ::/0을 입력합니다.
  - (선택 사항, IPv6 전용) Add another rule을 선택한 후 Type 목록에서 HTTPS를 선택하고 Source 필드에 ::/0을 입력합니다.
  - (선택 사항, IPv6 전용) Add another rule을 선택한 다음, Type 목록에서 SSH(Linux용) 또는 RDP(Windows용)를 선택합니다. Source 필드에 네트워크의 IPv6 주소 범위를 입력합니다. (이 주소 범위를 모를 경우, 테스트 용도로 ::/0을 사용할 수 있습니다. 실제 환경에서는 특정 IPv6 주소나 주소 범위만 해당 인스턴스에 액세스할 수 있도록 허가합니다.)
7. [Save]를 선택합니다.
8. (선택 사항) [Outbound Rules] 탭에서 [Edit]를 선택합니다. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택한 후 [Save]를 선택합니다.

#### VPC에서 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. 마법사의 지침대로 진행합니다. AMI를 선택하고 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.

#### Note

IPv6 통신 용도로 인스턴스를 사용하고자 하는 경우, 지원되는 인스턴스 유형(예: T2)을 선택해야 합니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#) 단원을 참조하십시오.

4. [Configure Instance Details] 페이지의 [Network] 목록에서 1단계에서 만들었던 VPC를 선택한 후 서브넷을 지정합니다.
5. (선택 사항) 기본이 아닌 VPC에서 시작되는 인스턴스에는 퍼블릭 IPv4 주소가 할당되지 않도록 기본 설정되어 있습니다. 인스턴스에 연결할 수 있도록 하기 위해, 지금 퍼블릭 IPv4 주소를 지정하거나, 탄력적 IP 주소를 할당하고 인스턴스를 시작한 후 이를 인스턴스에 지정할 수 있습니다. 지금 퍼블릭 IPv4 주소를 지정하려면 Auto-assign Public IP 목록에서 Enable을 선택해야 합니다.

#### Note

인덱스가 eth0인 새로운 단일 네트워크 인터페이스에는 자동 할당 퍼블릭 IP 기능만 사용할 수 있습니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 80\)](#) 단원을 참조하십시오.

6. (선택 사항, IPv6 전용) 서브넷 범위 내에서 인스턴스에 IPv6 주소를 자동 할당할 수 있습니다. Auto-assign IPv6 IP에 대해 Enable을 선택합니다.
7. 마법사의 다음 두 페이지에서 인스턴스의 스토리지를 구성하고 태그를 추가할 수 있습니다. [Configure Security Group] 페이지에서 [Select an existing security group] 옵션을 선택하고, 2단계에서 만든 [WebServerSG] 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.

8. 선택한 설정을 검토합니다. 필요한 사항을 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.
9. 5단계에서 인스턴스에 퍼블릭 IPv4 주소를 지정하지 않은 경우, IPv4를 통해 인스턴스에 연결할 수 없습니다. 인스턴스에 탄력적 IP 주소를 할당합니다:
  - a. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
  - b. 탐색 창에서 [Elastic IPs]를 선택합니다.
  - c. Allocate new address를 선택합니다.
  - d. [Allocate]를 선택합니다.

#### Note

계정이 EC2-Classic을 지원할 경우 우선 [VPC]를 선택합니다.

- e. 목록에서 탄력적 IP 주소와 [Actions], [Associate Address]를 차례로 선택합니다.
- f. 주소를 연결할 인스턴스를 선택한 다음 [Associate]를 선택합니다.

이제 VPC의 인스턴스에 연결할 수 있습니다. Linux 인스턴스 연결 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#)을 참조하십시오. Windows 인스턴스 연결 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

## 시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC(NAT)

이 시나리오의 구성에는 퍼블릭 서브넷과 프라이빗 서브넷이 있는 Virtual Private Cloud(VPC)가 포함됩니다. 이 시나리오는 백엔드 서버에 대한 공개적인 액세스를 차단하면서 퍼블릭 웹 애플리케이션을 실행하려는 경우에 권장됩니다. 일반적인 예로 웹 서버는 퍼블릭 서브넷에 두고 데이터베이스 서버는 프라이빗 서브넷에 두는 다중 계층 웹 사이트가 있습니다. 웹 서버가 데이터베이스 서버와 통신할 수 있도록 보안 및 라우팅을 설정할 수 있습니다.

퍼블릭 서브넷의 인스턴스는 아웃바운드 트래픽을 인터넷으로 직접 전송할 수 있는 반면, 프라이빗 서브넷의 인스턴스는 그렇게 할 수 없습니다. 그 대신, 프라이빗 서브넷의 인스턴스는 퍼블릭 서브넷에 상주하는 네트워크 주소 변환(NAT) 게이트웨이를 사용하여 인터넷에 액세스할 수 있습니다. 소프트웨어 업데이트 시 NAT 게이트웨이를 사용하여 데이터베이스 서버를 인터넷에 연결할 수 있지만, 인터넷에서 데이터베이스 서버 연결을 설정할 수 없습니다.

#### Note

VPC 마법사를 사용하여 NAT 인스턴스로 VPC를 구성할 수도 있지만, NAT 게이트웨이를 사용하는 것이 좋습니다. 자세한 내용은 [NAT 게이트웨이 \(p. 173\)](#) 단원을 참조하십시오.

이 단원에서는 Amazon VPC 콘솔의 VPC 마법사를 사용하여 VPC와 NAT 게이트웨이를 생성한다고 가정합니다.

이 시나리오를 IPv6에 맞게 구성할 수도 있습니다. 즉 VPC 마법사를 이용해 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 만들 수 있습니다. 서브넷에서 시작한 인스턴스는 IPv6 주소를 받아 IPv6를 사용해 통신할 수 있습니다. 프라이빗 서브넷의 인스턴스는 외부 전용 인터넷 게이트웨이를 사용하여 IPv6를 통해 인터넷에 연결하지만, 인터넷은 IPv6를 통해 프라이빗 인스턴스에 대한 연결을 설정할 수 없습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

#### 항목

- [개요 \(p. 24\)](#)
- [라우팅 \(p. 25\)](#)

- [보안 \(p. 26\)](#)
- [시나리오 2 구현 \(p. 29\)](#)
- [NAT 인스턴스를 사용하여 시나리오 2 구현 \(p. 32\)](#)

## 개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

- IPv4 CIDR 블록의 크기가 /16(예: 10.0.0.0/16)인 VPC. 이것은 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- IPv4 CIDR 블록의 크기가 /24(예: 10.0.0.0/24)인 퍼블릭 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다. 퍼블릭 서브넷은 인터넷 게이트웨이로 이어지는 경로가 있는 라우팅 테이블과 연결된 서브넷입니다.
- IPv4 CIDR 블록의 크기가 /24(예: 10.0.1.0/24)인 프라이빗 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다.
- 인터넷 게이트웨이. VPC를 인터넷 및 다른 AWS 서비스에 연결합니다.
- 서브넷 범위에서 프라이빗 IPv4 주소가 있는 인스턴스(예: 10.0.0.5, 10.0.1.5). 이 경우 인스턴스가 서로 통신할 수 있으며 VPC의 다른 인스턴스와 통신할 수 있습니다.
- 탄력적 IPv4 주소(예: 198.51.100.1)가 있는 퍼블릭 서브넷의 인스턴스. 이 경우 탄력적 IPv4 주소는 인터넷을 통해 접근할 수 있게 해주는 퍼블릭 IPv4 주소입니다. 인스턴스는 탄력적 IP 주소 대신 퍼블릭 IP 주소가 실행 시 지정될 수 있습니다. 프라이빗 서브넷의 인스턴스는 인터넷에서 수신되는 트래픽을 수락할 필요가 없는 백 엔드 서버이므로 퍼블릭 IP 주소가 없지만, NAT 게이트웨이를 사용하여 인터넷으로 요청을 전송할 수 있습니다(다음 글머리표 항목 참조).
- 자체 탄력적 IPv4 주소를 가진 NAT 게이트웨이. 이를 통해 프라이빗 서브넷의 인스턴스는 IPv4를 통해 인터넷에 요청을 전송할 수 있습니다(예: 소프트웨어 업데이트 요청).
- 퍼블릭 서브넷과 연결된 사용자 지정 라우팅 테이블. 이 라우팅 테이블에는 서브넷의 인스턴스가 IPv4를 통해 VPC의 다른 인스턴스와 통신할 수 있게 하는 항목과, 서브넷의 인스턴스가 IPv4를 통해 인터넷과 직접 통신할 수 있게 하는 항목이 들어 있습니다.
- 프라이빗 서브넷과 연결된 기본 라우팅 테이블. 라우팅 테이블에는 서브넷의 인스턴스가 IPv4를 통해 VPC의 다른 인스턴스와 통신할 수 있게 해주는 항목과, 서브넷의 인스턴스가 IPv4로 NAT 게이트웨이를 통해 인터넷과 통신할 수 있게 해주는 항목이 들어 있습니다.

서브넷에 대한 자세한 정보는 [VPC 및 서브넷 \(p. 62\)](#) 단원을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이 \(p. 165\)](#)를 참조하십시오. NAT 게이트웨이에 대한 자세한 내용은 [NAT 게이트웨이 \(p. 173\)](#)를 참조하십시오.

## IPv6 개요

이 시나리오에 IPv6를 사용할 수도 있습니다. 위에 나열된 구성 요소뿐 아니라 다음 요소도 구성에 포함됩니다.

- VPC와 연결된 /56 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/56). Amazon은 CIDR을 자동 할당하므로 범위를 직접 선택할 수는 없습니다.
- 퍼블릭 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. VPC IPv6 CIDR 블록의 크기는 선택할 수 없습니다.
- 프라이빗 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a01::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. 서브넷 IPv6 CIDR 블록의 크기는 선택할 수 없습니다.
- 서브넷 범위에서 인스턴스에 할당된 IPv6 주소(예: 2001:db8:1234:1a00::1a).
- 외부 전용 인터넷 게이트웨이. 이를 통해 프라이빗 서브넷의 인스턴스는 IPv6를 통해 인터넷에 요청을 전송할 수 있습니다(예: 소프트웨어 업데이트 요청). 프라이빗 서브넷의 인스턴스가 IPv6를 통해 인터넷과의

통신할 수 있게 하려면 외부 전용 인터넷 게이트웨이가 필요합니다. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

- 퍼블릭 서브넷의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있게 해주며, 또한 인터넷을 통해 직접 통신할 수 있게 해주는 사용자 정의 라우팅 테이블의 라우팅 테이블 항목.
- 프라이빗 서브넷의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있도록 해주며, 또한 외부 전용 인터넷 게이트웨이를 통해 인터넷과 통신할 수 있게 해주는 기본 라우팅 테이블의 라우팅 테이블 항목.

## 라우팅

이 시나리오에서 VPC 마법사는 프라이빗 서브넷에 사용되는 기본 라우팅 테이블을 업데이트하고 사용자 정의 라우팅 테이블을 만들어 이를 퍼블릭 서브넷에 연결합니다.

이 시나리오에서는, 각 서브넷에서 AWS(예: Amazon EC2 또는 Amazon S3 엔드포인트)로 향하는 모든 트래픽이 인터넷 게이트웨이를 거칩니다. 프라이빗 서브넷의 데이터베이스 서버는 탄력적 IP 주소가 없기 때문에 인터넷에서 직접 트래픽을 수신할 수 없습니다. 하지만 데이터베이스 서버는 퍼블릭 서브넷의 NAT 디바이스를 통해 인터넷 트래픽을 전송하고 수신할 수 있습니다.

추가로 만든 모든 서브넷은 기본적으로 기본 라우팅 테이블을 사용하며, 따라서 기본적으로 프라이빗 서브넷입니다. 퍼블릭 서브넷으로 만들고 싶다면 연결된 라우팅 테이블을 변경하면 됩니다.

다음 표에서는 이 시나리오의 라우팅 테이블에 대해 설명합니다.

### 기본 라우팅 테이블

첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목으로서, VPC의 인스턴스가 서로 통신할 수 있게 해줍니다. 두 번째 항목에서는 기타 서브넷 트래픽을 모두 NAT 게이트웨이(예: nat-12345678901234567)로 전송합니다.

목적지	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gateway-id

### 사용자 정의 라우팅 테이블

첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목으로서, 이 VPC의 인스턴스가 서로 통신할 수 있게 해줍니다. 두 번째 항목에서는 인터넷 게이트웨이(예: igw-1a2b3d4d)를 통해 기타 서브넷 트래픽을 모두 인터넷으로 라우팅합니다.

목적지	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

## IPv6에 대한 라우팅

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다. 다음 표는 VPC에서 IPv6 통신을 사용하기로 한 경우, 이 시나리오에 대한 라우팅 테이블을 정리한 것입니다.

### 기본 라우팅 테이블



두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다. 네 번째 항목은 기타 IPv6 서브넷 트래픽을 모두 외부 전용 인터넷 게이트웨이로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	nat-gateway-id
::/0	egress-only-igw-id

#### 사용자 정의 라우팅 테이블

두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다. 네 번째 항목에서는 기타 IPv6 서브넷 트래픽을 모두 인터넷 게이트웨이로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	igw-id
::/0	igw-id

## 보안

AWS는 VPC의 보안을 강화하기 위해 사용할 수 있는 두 가지 기능, 보안 그룹과 네트워크 ACL을 제공합니다. 보안 그룹은 인스턴스용 인바운드 및 아웃바운드 트래픽을 제어하고, 네트워크 ACL은 서브넷용 인바운드 및 아웃바운드 트래픽을 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 맞출 수 있지만, 원하는 경우 네트워크 ACL을 사용하여 VPC에 보안 계층을 더 추가할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

시나리오 2에서는 네트워크 ACL이 아닌 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 2를 위한 권장 규칙 \(p. 118\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 96\)](#)과 함께 제공됩니다. 인스턴스를 시작하는 동안 다른 보안 그룹을 지정하지 않는 경우, VPC에서 시작되는 인스턴스는 적절한 기본 보안 그룹과 자동 연결됩니다. 이 시나리오의 경우, 기본 보안 그룹을 수정하는 대신 다음과 같은 보안 그룹을 생성하는 것이 좋습니다.

- WebServerSG: 퍼블릭 서브넷의 웹 서버를 시작할 때 이 보안 그룹을 지정합니다.
- DBServerSG: 프라이빗 서브넷의 데이터베이스 서버를 시작할 때 이 보안 그룹을 지정합니다.

보안 그룹에 지정된 인스턴스는 서로 다른 서브넷에 있을 수 있습니다. 하지만 이 시나리오에서 각 보안 그룹은 인스턴스가 수행하는 역할 유형과 일치하며, 각 역할은 인스턴스가 특정 서브넷에 있을 것을 요구합니다. 따라서 이 시나리오에서 한 보안 그룹에 지정된 모든 인스턴스는 동일한 서브넷에 있습니다.

아래 표에서는 WebServerSG 보안 그룹에 권장되는 규칙을 설명합니다. 이 규칙은 웹 서버가 인터넷 트래픽을 수신하고, 네트워크에서 발생하는 SSH 및 RDP 트래픽을 수신할 수 있도록 허용합니다. 웹 서버는 또한 프라이빗 서브넷의 데이터베이스 서버에 대해 읽기 및 쓰기 요청을 시작하고, 인터넷으로 트래픽을 전송합니다(예: 소프트웨어 업데이트 받기). 웹 서버는 다른 아웃바운드 통신을 시작하지 않기 때문에 기본 아웃바운드 규칙은 제거됩니다.



Note

이러한 권장 사항에는 SSH와 RDP 액세스 그리고 Microsoft SQL Server와 MySQL 액세스가 포함됩니다. 상황에 따라 Linux(SSH 및 MySQL) 또는 Windows(RDP 및 Microsoft SQL Server)용 규칙만 필요할 수 있습니다.

WebServerSG: 권장 규칙

인바운드			
소스	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	홈 네트워크에서 Linux 인스턴스로의 인바운드 SSH 액세스 허용(인터넷 게이트웨이를 통해). <a href="http://checkip.amazonaws.com">http://checkip.amazonaws.com</a> 과 같은 서비스를 사용하여 로컬 컴퓨터의 퍼블릭 IPv4 주소를 얻을 수 있습니다. 고정 IP 주소 없이 ISP 또는 방화벽을 경유하여 연결하는 경우에는 클라이언트 컴퓨터가 사용하는 IP 주소의 범위를 알아내야 합니다.
홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	홈 네트워크에서 Windows 인스턴스로 인바운드 RDP 액세스 허용(인터넷 게이트웨이를 통해).
아웃바운드			
목적지	프로토콜	포트 범위	설명
DBServerSG 보안 그룹의 ID	TCP	1433	DBServerSG 보안 그룹에 지정된 데이터베이스 서버에 대해 아웃바운드 Microsoft SQL Server 액세스 허용
DBServerSG 보안 그룹의 ID	TCP	3306	DBServerSG 보안 그룹에 지정된 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용
0.0.0.0/0	TCP	80	모든 IPv4 주소에 대한 아웃바운드 HTTP 액세스를 허용
0.0.0.0/0	TCP	443	모든 IPv4 주소에 대한 아웃바운드 HTTPS 액세스를 허용

아래 표에서는 DBServerSG 보안 그룹에 권장되는 규칙에 대해 설명합니다. 이 규칙은 웹 서버에서 전송되는 데이터베이스 읽기 또는 쓰기 요청을 허용합니다. 데이터베이스 서버는 인터넷으로 향하는 트래픽을 시작할 수도 있습니다. 라우팅 테이블에 따라 이 트래픽은 NAT 게이트웨이로 전송된 후 인터넷 게이트웨이를 통해 인터넷에 전달됩니다.

DBServerSG: 권장 규칙

인바운드
------

소스	프로토콜	포트 범위	설명
WebServerSG 보안 그룹의 ID	TCP	1433	WebServerSG 보안 그룹과 연결된 웹 서버에서 인바운드 Microsoft SQL Server 액세스가 가능하도록 허용
WebServerSG 보안 그룹의 ID	TCP	3306	WebServerSG 보안 그룹과 연결된 웹 서버에서 인바운드 MySQL Server 액세스가 가능하도록 허용
<b>아웃바운드</b>			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	IPv4를 통해 인터넷에 접근하는 아웃바운드 HTTP 액세스 허용(예: 소프트웨어 업데이트 요청)
0.0.0.0/0	TCP	443	인터넷에 대한 아웃바운드 HTTPS 액세스 허용 액세스 허용(예: 소프트웨어 업데이트 요청)

(선택 사항) VPC의 기본 보안 그룹에는 지정된 인스턴스가 서로 통신할 수 있도록 자동 허용하는 규칙이 있습니다. 사용자 지정 보안 그룹에 이러한 유형의 통신을 허용하려면 다음과 같은 규칙을 추가해야 합니다.

인바운드			
소스	프로토콜	포트 범위	설명
보안 그룹의 ID	모두	모두	이 보안 그룹에 지정된 다른 인스턴스로부터의 인바운드 트래픽 허용.
<b>아웃바운드</b>			
목적지	프로토콜	포트 범위	설명
보안 그룹의 ID	모두	모두	이 보안 그룹에 지정된 다른 인스턴스에 대해 아웃바운드 트래픽 허용

## IPv6의 보안

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, WebServerSG 및 DBServerSG 보안 그룹에 별도의 규칙을 추가하여 인스턴스에 대한 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다. 이 시나리오에서 웹 서버는 IPv6를 통해 인터넷 트래픽을 모두 수신할 수 있고, IPv6를 통해 로컬 네트워크로부터 SSH 또는 RDP 트래픽을 수신할 수 있습니다. 또한 인터넷으로의 아웃바운드 IPv6 트래픽을 시작할 수 있습니다. 데이터베이스 서버는 인터넷으로의 아웃바운드 IPv6 트래픽을 시작할 수 있습니다.

다음은 WebServerSG 보안 그룹에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

인바운드			
소스	프로토콜	포트 범위	설명
:::0	TCP	80	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용

::/0	TCP	443	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
네트워크의 IPv6 주소 범위	TCP	22	(Linux 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 SSH 액세스 허용
네트워크의 IPv6 주소 범위	TCP	3389	(Windows 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 RDP 액세스 허용
아웃바운드			
목적지	프로토콜	포트 범위	설명
::/0	TCP	HTTP	임의의 IPv6 주소에 대한 아웃바운드 HTTP 액세스 허용
::/0	TCP	HTTPS	임의의 IPv6 주소에 대한 아웃바운드 HTTPS 액세스 허용

다음은 DBServerSG 보안 그룹에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

아웃바운드			
목적지	프로토콜	포트 범위	설명
::/0	TCP	80	임의의 IPv6 주소에 대한 아웃바운드 HTTP 액세스 허용
::/0	TCP	443	임의의 IPv6 주소에 대한 아웃바운드 HTTPS 액세스 허용

## 시나리오 2 구현

VPC 마법사를 사용하여 VPC, 서브넷 및 NAT 게이트웨이를 만들 수 있습니다. 외부 전용 인터넷 게이트웨이를 만들 수도 있습니다. NAT 게이트웨이에 탄력적 IP 주소를 지정해야 합니다. 주소가 없는 경우, 먼저 계정에 주소를 할당해야 합니다. 기존 탄력적 IP 주소를 사용하려면 해당 주소가 다른 인스턴스 또는 네트워크 인터페이스와 현재 연결되어 있지 않은지 확인합니다. NAT 게이트웨이는 VPC의 퍼블릭 서브넷에서 자동으로 생성됩니다.

이 절차에는 VPC용 IPv6 통신을 활성화 및 구성하기 위한 옵션 절차가 포함됩니다. VPC에서 IPv6를 사용하고 싶지 않다면 이 절차를 수행할 필요가 없습니다.

(선택 사항) NAT 게이트웨이(IPv4)에 사용할 탄력적 IP 주소를 할당하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. [Allocate New Address]를 선택합니다.
4. [Yes, Allocate]를 선택합니다.

### Note

계정이 EC2-Classic을 지원할 경우 우선 [Network platform] 목록에서 [EC2-VPC]를 선택합니다.

## VPC를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. 두 번째 옵션인 [VPC with Public and Private Subnets]을 선택하고 [Select]를 선택합니다.
4. VPC name, Public subnet name 및 Private subnet name에 대해 VPC와 서브넷 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다. VPC 및 서브넷에 대해 고유한 IPv4 CIDR 블록 범위를 지정하거나 기본값을 그대로 둘 수 있습니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. Public subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택하고 서브넷에 16진수 페어 값을 지정하거나 기본값을 그대로 둡니다. Private subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 16진수 페어 값을 지정하거나 기본값을 그대로 둡니다.
6. Specify the details of your NAT gateway 섹션에서 계정의 탄력적 IP 주소에 할당 ID를 지정합니다.
7. 페이지에서 나머지 기본값은 그대로 두고 Create VPC를 선택합니다.

WebServerSG 및 DBServerSG 보안 그룹은 서로를 참조하기 때문에 이러한 보안 그룹에 규칙을 추가하기 전에 이 시나리오에 필요한 모든 보안 그룹을 만듭니다.

## WebServerSG 및 DBServerSG 보안 그룹을 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]와 [Create Security Group]을 차례로 선택합니다.
3. 보안 그룹 이름으로 WebServerSG를 지정하고 설명을 제공합니다. [VPC]에 대해 생성한 VPC의 ID를 선택하고 [Yes, Create]를 선택합니다.
4. [Create Security Group]을 다시 선택합니다.
5. 보안 그룹 이름으로 DBServerSG를 지정하고 설명을 제공합니다. [VPC]에 대해 VPC의 ID를 선택한 다음 [Yes, Create]를 선택합니다.

WebServerSG 보안 그룹에 규칙을 추가하려면 다음을 수행합니다.

1. 앞에서 만든 WebServerSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
2. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type], [HTTP]를 선택합니다. [Source]에 0.0.0.0/0을 입력합니다.
  - b. [Add another rule], [Type], [HTTPS]를 차례로 선택하고 [Source]에 0.0.0.0/0을 입력합니다.
  - c. [Add another rule], [Type], [SSH]를 선택합니다. Source에 네트워크의 퍼블릭 IPv4 주소 범위를 입력합니다.
  - d. [Add another rule], [Type], [RDP]를 선택합니다. Source에 네트워크의 퍼블릭 IPv4 주소 범위를 입력합니다.
  - e. (선택 사항, IPv6 전용) Add another rule, Type, HTTP를 차례로 선택합니다. [Source]에 ::/0을 입력합니다.
  - f. (선택 사항, IPv6 전용) Add another rule, Type, HTTPS를 차례로 선택합니다. [Source]에 ::/0을 입력합니다.
  - g. (선택 사항, IPv6 전용) Add another rule, Type, SSH (Linux용) 또는 RDP(Windows용)를 선택합니다. Source에 네트워크의 IPv6 주소 범위를 입력합니다.
  - h. Save를 선택합니다.
3. [Outbound Rules] 탭에서 [Edit]를 선택한 후 다음과 같이 아웃바운드 트래픽에 대한 규칙을 추가합니다.
  - a. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택합니다.

- b. [Type], [MS SQL]을 선택합니다. [Destination]에 DBServerSG 보안 그룹의 ID를 지정합니다.
- c. [Add another rule], [Type], [MySQL]을 선택합니다. [Destination]에 DBServerSG 보안 그룹의 ID를 지정합니다.
- d. [Add another rule], [Type], [HTTPS]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
- e. [Add another rule], [Type], [HTTP]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
- f. (선택 사항, IPv6 전용) Add another rule, Type, HTTPS를 차례로 선택합니다. [Destination]에 [::/0]을 입력합니다.
- g. (선택 사항, IPv6 전용) Add another rule, Type, HTTP를 차례로 선택합니다. [Destination]에 [::/0]을 입력합니다.
- h. [Save]를 선택합니다.

#### DBServerSG 보안 그룹에 권장 규칙을 추가하려면

1. 앞에서 만든 DBServerSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
2. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type], [MS SQL]을 선택합니다. [Source]에 WebServerSG 보안 그룹의 ID를 지정합니다.
  - b. [Add another rule], [Type], [MYSQL]을 선택합니다. [Source]에 WebServerSG 보안 그룹의 ID를 지정합니다.
  - c. Save를 선택합니다.
3. [Outbound Rules] 탭에서 [Edit]를 선택한 후 다음과 같이 아웃바운드 트래픽에 대한 규칙을 추가합니다.
  - a. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택합니다.
  - b. [Type], [HTTP]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
  - c. [Add another rule], [Type], [HTTPS]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
  - d. (선택 사항, IPv6 전용) Add another rule, Type, HTTP를 차례로 선택합니다. [Destination]에 [::/0]을 입력합니다.
  - e. (선택 사항, IPv6 전용) Add another rule, Type, HTTPS를 차례로 선택합니다. [Destination]에 [::/0]을 입력합니다.
  - f. [Save]를 선택합니다.

이제 VPC에서 인스턴스를 시작할 수 있습니다.

#### 인스턴스를 시작하려면(웹 서버 또는 데이터베이스 서버)

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.

#### Note

IPv6 통신 용도로 인스턴스를 사용하고자 하는 경우, 지원되는 인스턴스 유형(예: T2)을 선택해야 합니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#) 단원을 참조하십시오.

4. [Configure Instance Details] 페이지의 [Network]에서, 앞에서 만들었던 VPC를 선택한 후 서브넷을 선택합니다. 예를 들어 웹 서버는 퍼블릭 서브넷에서 시작하고, 데이터베이스 서버는 프라이빗 서브넷에서 시작합니다.
5. (선택 사항) 기본이 아닌 VPC에서 시작되는 인스턴스에는 퍼블릭 IPv4 주소가 할당되지 않도록 기본 설정되어 있습니다. 퍼블릭 서브넷의 인스턴스에 연결하려면 지금 퍼블릭 IPv4 주소를 지정하거나, 탄력적 IP 주소를 할당하고 인스턴스를 시작한 후 이를 인스턴스에 지정할 수 있습니다. 지금 퍼블릭 IPv4 주소를 지정하려면 Auto-assign Public IP 목록에서 Enable을 선택해야 합니다. 프라이빗 서브넷의 인스턴스에는 퍼블릭 IP 주소를 지정할 필요가 없습니다.

## Note

디바이스 인덱스가 eth0인 새로운 단일 네트워크 인터페이스에는 자동 할당 퍼블릭 IPv4 기능만 사용할 수 있습니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 80\)](#) 단원을 참조하십시오.

6. (선택 사항, IPv6 전용) 서브넷 범위 내에서 인스턴스에 IPv6 주소를 자동 할당할 수 있습니다. Auto-assign IPv6 IP에 대해 Enable을 선택합니다.
7. 마법사의 다음 두 페이지에서 인스턴스의 스토리지를 구성하고 태그를 추가할 수 있습니다. [Configure Security Group] 페이지에서 [Select an existing security group] 옵션을 선택한 후 앞에서 만든 보안 그룹 중 하나(웹 서버의 경우 [WebServerSG] 또는 데이터베이스 서버의 경우 [DBServerSG])를 선택합니다. [Review and Launch]를 선택합니다.
8. 선택한 설정을 검토합니다. 필요한 사항을 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

5단계에서 퍼블릭 서브넷의 인스턴스에 퍼블릭 IPv4 주소를 지정하지 않은 경우, 인스턴스에 연결할 수 없습니다. 퍼블릭 서브넷의 인스턴스에 액세스하려면 먼저 해당 인스턴스에 탄력적 IP 주소를 지정해야 합니다.

인스턴스에 탄력적 IP 주소를 할당하고 지정하려면 (IPv4)

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. [Allocate]를 선택합니다.

## Note

계정이 EC2-Classic을 지원할 경우 우선 [VPC]를 선택합니다.

5. 목록에서 탄력적 IP 주소를 선택한 다음, [Actions], [Associate Address]를 선택합니다.
6. 네트워크 인터페이스 또는 인스턴스를 선택합니다. [Private IP]에서 탄력적 IP 주소와 연결할 주소를 선택하고 [Associate]를 선택합니다.

이제 VPC의 인스턴스에 연결할 수 있습니다. Linux 인스턴스 연결 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#)을 참조하십시오. Windows 인스턴스 연결 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

## NAT 인스턴스를 사용하여 시나리오 2 구현

NAT 게이트웨이 대신 NAT 인스턴스를 사용하여 시나리오 2를 구현할 수 있습니다. NAT 인스턴스에 대한 자세한 내용은 [NAT 인스턴스 \(p. 182\)](#)를 참조하십시오.

위와 동일한 절차를 따를 수 있지만, VPC 마법사의 NAT 섹션에서 [Use a NAT instance instead]를 선택하고 NAT 인스턴스에 대한 세부 정보를 지정합니다. 또한 NAT 인스턴스가 프라이빗 서브넷의 인스턴스에서 인터넷 바운드 트래픽을 수신하고 네트워크에서 SSH 트래픽을 수신할 수 있도록 NAT 인스턴스(NATSG)에 대한 보안 그룹이 필요합니다. 또한 NAT 인스턴스는 인터넷으로 트래픽을 전송할 수 있으며 따라서 프라이빗 서브넷의 인스턴스는 소프트웨어 업데이트를 받을 수 있습니다.

NAT 인스턴스를 사용하여 VPC를 만든 후 NAT 인스턴스와 연결된 보안 그룹을 새 NATSG 보안 그룹으로 변경해야 합니다. 기본적으로 NAT 인스턴스는 기본 보안 그룹을 사용하여 시작됩니다.

NATSG: 권장 규칙

인바운드

소스	프로토콜	포트 범위	설명
10.0.1.0/24	TCP	80	프라이빗 서브넷의 데이터베이스 서버로부터의 인바운드 HTTP 트래픽 허용
10.0.1.0/24	TCP	443	프라이빗 서브넷의 데이터베이스 서버로부터의 인바운드 HTTPS 트래픽 허용
네트워크의 퍼블릭 IP 주소 범위	TCP	22	네트워크로부터 NAT 인스턴스에 대한 인바운드 SSH 액세스 허용(인터넷 게이트웨이를 통해)
아웃바운드			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	인터넷으로의 아웃바운드 HTTP 액세스 허용(인터넷 게이트웨이를 통해)
0.0.0.0/0	TCP	443	인터넷으로의 아웃바운드 HTTPS 액세스 허용(인터넷 게이트웨이를 통해)

#### NATSG 보안 그룹을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]와 [Create Security Group]을 차례로 선택합니다.
3. 보안 그룹 이름으로 NATSG를 지정하고 설명을 제공합니다. [VPC]에 대해 VPC의 ID를 선택한 다음 [Yes, Create]를 선택합니다.
4. 앞에서 만든 NATSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
5. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type], [HTTP]를 선택합니다. [Source]에 프라이빗 서브넷의 IP 주소 범위를 입력합니다.
  - b. [Add another rule], [Type], [HTTPS]를 선택합니다. [Source]에 프라이빗 서브넷의 IP 주소 범위를 입력합니다.
  - c. [Add another rule], [Type], [SSH]를 선택합니다. [Source]에 네트워크의 퍼블릭 IP 주소 범위를 입력합니다.
  - d. Save를 선택합니다.
6. [Outbound Rules] 탭에서 [Edit]를 선택한 후 다음과 같이 아웃바운드 트래픽에 대한 규칙을 추가합니다.
  - a. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택합니다.
  - b. [Type], [HTTP]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
  - c. [Add another rule], [Type], [HTTPS]를 선택합니다. [Destination]에 [0.0.0.0/0]을 입력합니다.
  - d. [Save]를 선택합니다.

VPC 마법사에서 NAT 인스턴스를 시작했을 때 VPC에 기본 보안 그룹이 사용되었습니다. NAT 인스턴스에 기본 보안 그룹 대신 NATSG 보안 그룹을 연결해야 합니다.

#### NAT 인스턴스의 보안 그룹을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.

3. 목록에서 NAT 인스턴스에 대한 네트워크 인터페이스를 선택한 후 [Actions], [Change Security Groups]를 선택합니다.
4. [Change Security Groups] 대화 상자의 [Security groups]에 대해 만든 NATSG 보안 그룹(보안 (p. 26) 참조)을 선택한 후 [Save]를 선택합니다.

## 시나리오 3: 공인 및 사설 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC

이 시나리오의 구성에는 퍼블릭 서브넷과 프라이빗 서브넷이 있는 Virtual Private Cloud(VPC)와, IPsec VPN 터널을 통해 귀사의 네트워크와 통신하기 위한 가상 프라이빗 게이트웨이가 포함됩니다. 네트워크를 클라우드로 확장하고 VPC로부터 직접 인터넷에 액세스하려는 경우 이 시나리오를 사용하는 것이 좋습니다. 이 시나리오를 사용하면 확장 가능한 웹 프론트 엔드가 있는 멀티 티어 애플리케이션을 퍼블릭 서브넷에서 실행하고, IPsec VPN 연결을 통해 네트워크에 연결된 프라이빗 서브넷에 데이터를 보관할 수 있습니다.

이 단원에서는 Amazon VPC 콘솔의 VPC 마법사를 사용하여 VPC와 VPN 연결을 생성한다고 가정합니다.

이 시나리오를 IPv6에 맞게 구성할 수도 있습니다. 즉 VPC 마법사를 이용해 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 만들 수 있습니다. 서브넷에서 시작한 인스턴스는 IPv6 주소를 받을 수 있습니다. 현재 Amazon은 VPN 연결을 통한 IPv6 통신을 지원하지 않습니다. 하지만 VPC의 인스턴스들은 IPv6를 통해서도 통신할 수 있고, 퍼블릭 서브넷의 인스턴스들은 IPv6를 통해 인터넷으로 통신할 수 있습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

### 항목

- 개요 (p. 34)
- 라우팅 (p. 35)
- 보안 (p. 37)
- 시나리오 3 구현 (p. 40)

## 개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.

### Important

이 시나리오를 위해 [Amazon VPC 네트워크 관리자 안내서](#)에서는 귀사의 VPN 연결에서 네트워크 관리자가 Amazon VPC 고객 게이트웨이를 구성하기 위해 해야 할 사항을 설명합니다.

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

- IPv4 CIDR의 크기가 /16(예: 10.0.0.0/16)인 Virtual Private Cloud(VPC). 이것은 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- IPv4 CIDR의 크기가 /24(예: 10.0.0.0/24)인 퍼블릭 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다. 퍼블릭 서브넷은 인터넷 게이트웨이로 이어지는 경로가 있는 라우팅 테이블과 연결된 서브넷입니다.
- IPv4 CIDR의 크기가 /24(예: 10.0.1.0/24)인 VPN 전용 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다.
- 인터넷 게이트웨이. 인터넷 게이트웨이는 VPC를 인터넷 및 다른 AWS 제품에 연결합니다.
- VPC와 네트워크 사이의 VPN 연결. VPN 연결은 VPC 연결의 Amazon 측에 있는 가상 프라이빗 게이트웨이와 VPN 연결의 사용자 측에 있는 고객 게이트웨이로 구성됩니다.
- 프라이빗 IPv4 주소가 서브넷 범위(예: 10.0.0.5 및 10.0.1.5)에 있는 인스턴스는 서로 통신할 수 있고 VPC의 다른 인스턴스와도 통신이 가능합니다.



- 탄력적 IP 주소(예: 198.51.100.1)가 있는 퍼블릭 서브넷의 인스턴스. 이 경우 탄력적 IP 주소는 인터넷을 통해 접근할 수 있게 해주는 퍼블릭 IPv4 주소입니다. 이 인스턴스는 실행 시 탄력적 IP 주소 대신에 퍼블릭 IPv4 주소가 지정될 수 있습니다. VPN 전용 서브넷의 인스턴스는 인터넷으로부터의 수신 트래픽은 허용할 필요가 없지만 귀사의 네트워크와 트래픽을 주고받을 수 있는 백엔드 서버입니다.
- 퍼블릭 서브넷과 연결된 사용자 지정 라우팅 테이블. 이 라우팅 테이블에는 서브넷의 인스턴스가 VPC의 다른 인스턴스와 통신할 수 있게 하는 항목과, 서브넷의 인스턴스가 인터넷과 직접 통신할 수 있게 하는 항목이 들어 있습니다.
- VPN 전용 서브넷과 연결된 기본 라우팅 테이블. 이 라우팅 테이블에는 서브넷의 인스턴스가 VPC의 다른 인스턴스와 통신할 수 있게 하는 항목과, 서브넷의 인스턴스가 귀사의 네트워크와 직접 통신할 수 있게 하는 항목이 들어 있습니다.

서브넷에 대한 자세한 내용은 [VPC 및 서브넷 \(p. 62\)](#) 및 [VPC의 IP 주소 지정 \(p. 76\)](#)을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이 \(p. 165\)](#)를 참조하십시오. VPN 연결에 대한 자세한 내용은 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#)을 참조하십시오. 고객 게이트웨이 구성에 대한 자세한 내용은 [Amazon VPC 네트워크 관리자 안내서](#)를 참조하십시오.

## IPv6 개요

이 시나리오에 IPv6를 사용할 수도 있습니다. 위에 나열된 구성 요소뿐 아니라 다음 요소도 구성에 포함됩니다.

- VPC와 연결된 /56 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/56). AWS는 CIDR을 자동 할당하므로 범위를 직접 선택할 수는 없습니다.
- 퍼블릭 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. IPv6 CIDR의 크기는 선택할 수 없습니다.
- VPN 전용 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a01::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. IPv6 CIDR의 크기는 선택할 수 없습니다.
- 서브넷 범위에서 인스턴스에 할당된 IPv6 주소(예: 2001:db8:1234:1a00::1a).
- 퍼블릭 서브넷의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있게 해주며, 또한 인터넷을 통해 직접 통신할 수 있게 해주는 사용자 정의 라우팅 테이블의 라우팅 테이블 항목.
- VPN 전용 서브넷의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있게 해주는 기본 라우팅 테이블의 라우팅 테이블 항목.

## 라우팅

VPC에는 라우터가 내재되어 있습니다(이 시나리오의 구성 다이어그램 참조). 이 시나리오의 경우, VPC 마법사는 VPN 전용 서브넷에 사용되는 기본 라우팅 테이블을 업데이트하고 사용자 정의 라우팅 테이블을 만들어 이를 퍼블릭 서브넷에 연결합니다.

VPN 전용 서브넷의 인스턴스는 인터넷에 직접 액세스할 수 없습니다. 인터넷 바운드 트래픽은 먼저 가상 프라이빗 게이트웨이를 통과해야 귀사의 네트워크에 액세스하며, 이러한 트래픽에는 귀사의 방화벽과 기업 보안 정책이 적용됩니다. 인스턴스에서 AWS 바운드 트래픽(예: Amazon S3 또는 Amazon EC2 API에 대한 요청)을 전송할 경우 이 요청은 가상 프라이빗 게이트웨이를 거쳐야 귀사의 네트워크에 도달할 수 있으며 따라서 AWS에 도달하기 전에 인터넷에 액세스합니다. 현재 VPN 연결에는 IPv6를 지원하지 않습니다.

### Tip

귀사의 네트워크에서 퍼블릭 서브넷의 인스턴스에 대한 탄력적 IP 주소로 향하는 트래픽은 인터넷을 거치며 가상 프라이빗 게이트웨이를 통과하지 않습니다. 이렇게 하지 않고, 네트워크에서 나오는 트래픽이 가상 프라이빗 게이트웨이를 거쳐 퍼블릭 서브넷으로 향하도록 라우팅과 보안 그룹 규칙을 설정할 수 있습니다.

VPN 연결은 고정 라우팅 VPN 연결 또는 동적 라우팅 VPN 연결(BGP 사용)로 구성됩니다. 고정 라우팅을 선택하는 경우, VPN 연결을 생성할 때 네트워크의 IP 접두사를 수동으로 입력하라는 메시지가 표시됩니다. 동

적 라우팅을 선택할 경우 BGP를 사용하는 VPC에 대한 가상 프라이빗 게이트웨이에 IP 접두사가 자동으로 알려집니다.

다음 표에서는 이 시나리오의 라우팅 테이블에 대해 설명합니다.

#### 기본 라우팅 테이블

첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목으로서, VPC의 인스턴스가 IPv4를 통해 서로 통신할 수 있게 해줍니다. 두 번째 항목에서는 가상 프라이빗 게이트웨이(예: `vgw-1a2b3c4d`)를 통해 프라이빗 서브넷에서 해당 네트워크로 기타 IPv4 서브넷 트래픽을 모두 라우팅합니다.

목적지	Target
10.0.0.0/16	local
0.0.0.0/0	vgw-id

#### 사용자 정의 라우팅 테이블

첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목으로서, VPC의 인스턴스가 서로 통신할 수 있게 해줍니다. 두 번째 항목에서는 인터넷 게이트웨이(예: `igw-1a2b3c4d`)를 통해 퍼블릭 서브넷에서 인터넷으로 기타 IPv4 서브넷 트래픽을 모두 라우팅합니다.

목적지	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

## 대체 라우팅

프라이빗 서브넷의 인스턴스가 인터넷에 액세스하게 하려면, 퍼블릭 서브넷에 네트워크 주소 변환(NAT) 게이트웨이 또는 인스턴스를 생성하고 서브넷의 인터넷 바운드 트래픽이 NAT 디바이스로 가도록 라우팅을 설정하면 됩니다. 이렇게 하면 VPN 전용 서브넷의 인스턴스가 인터넷 게이트웨이를 통해 요청(예: 소프트웨어 업데이트)을 전송할 수 있습니다.

NAT 디바이스를 수동으로 설정하는 방법에 대한 자세한 내용은 [NAT \(p. 172\)](#)를 참조하십시오. VPC 마법사를 사용하여 NAT 디바이스를 설정하는 것에 대한 자세한 내용은 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\) \(p. 23\)](#)를 참조하십시오.

프라이빗 서브넷의 인터넷 바운드 트래픽을 NAT 디바이스로 경로 지정하려면 기본 라우팅 테이블을 다음과 같이 업데이트해야 합니다.

#### 기본 라우팅 테이블

첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목입니다. 두 번째 행의 항목은 고객 네트워크(이 경우 로컬 네트워크의 IP 주소는 `172.16.0.0/12`)로 가는 서브넷 트래픽을 가상 프라이빗 게이트웨이로 라우팅합니다. 세 번째 항목에서는 기타 서브넷 트래픽을 모두 NAT 게이트웨이로 전송합니다.

목적지	Target
10.0.0.0/16	local
172.16.0.0/12	vgw-id
0.0.0.0/0	nat-gateway-id

## IPv6에 대한 라우팅

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다. 다음 표는 VPC에서 IPv6 통신을 사용하기로 한 경우, 이 시나리오에 대한 라우팅 테이블을 정리한 것입니다.

기본 라우팅 테이블

두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	vgw-id

사용자 정의 라우팅 테이블

두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다. 네 번째 항목에서는 기타 IPv6 서브넷 트래픽을 모두 인터넷 게이트웨이로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	igw-id
::/0	igw-id

## 보안

AWS는 VPC의 보안을 강화하기 위해 사용할 수 있는 두 가지 기능, 보안 그룹과 네트워크 ACL을 제공합니다. 보안 그룹은 인스턴스용 인바운드 및 아웃바운드 트래픽을 제어하고, 네트워크 ACL은 서브넷용 인바운드 및 아웃바운드 트래픽을 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 맞출 수 있지만, 원하는 경우 네트워크 ACL을 사용하여 VPC에 보안 계층을 더 추가할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

시나리오 3의 경우, 네트워크 ACL이 아닌 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 3을 위한 권장 규칙 \(p. 123\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 96\)](#)과 함께 제공됩니다. 인스턴스를 시작하는 동안 다른 보안 그룹을 지정하지 않는 경우, VPC에서 시작되는 인스턴스는 적절한 기본 보안 그룹과 자동 연결됩니다. 이 시나리오의 경우, 기본 보안 그룹을 수정하는 대신 다음과 같은 보안 그룹을 생성하는 것이 좋습니다.

- WebServerSG: 퍼블릭 서브넷의 웹 서버를 시작할 때 이 보안 그룹을 지정합니다.
- DBServerSG: VPN 전용 서브넷의 데이터베이스 서버를 시작할 때 이 보안 그룹을 지정합니다.

보안 그룹에 지정된 인스턴스는 서로 다른 서브넷에 있을 수 있습니다. 하지만 이 시나리오에서 각 보안 그룹은 인스턴스가 수행하는 역할 유형과 일치하며, 각 역할은 인스턴스가 특정 서브넷에 있을 것을 요구합니다. 따라서 이 시나리오에서 한 보안 그룹에 지정된 모든 인스턴스는 동일한 서브넷에 있습니다.

아래 표에서는 WebServerSG 보안 그룹에 권장되는 규칙을 설명합니다. 이 규칙은 웹 서버가 인터넷 트래픽을 수신하고, 네트워크에서 발생하는 SSH 및 RDP 트래픽을 수신할 수 있도록 허용합니다. 웹 서버는 또한

VPN 전용 서브넷의 데이터베이스 서버에 대해 읽기 및 쓰기 요청을 시작하고, 인터넷으로 트래픽을 전송합니다(예: 소프트웨어 업데이트 받기). 웹 서버는 다른 아웃바운드 통신을 시작하지 않기 때문에 기본 아웃바운드 규칙은 제거됩니다.

**Note**

이 그룹에는 SSH 및 RDP 액세스와 Microsoft SQL Server 및 MySQL 액세스가 포함됩니다. 상황에 따라 Linux(SSH 및 MySQL) 또는 Windows(RDP 및 Microsoft SQL Server)용 규칙만 필요할 수 있습니다.

**WebServerSG: 권장 규칙**

인바운드			
소스	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
네트워크의 퍼블릭 IP 주소 범위	TCP	22	네트워크에서 Linux 인스턴스로 인터넷 게이트웨이를 거쳐 인바운드 SSH 액세스 허용.
네트워크의 퍼블릭 IP 주소 범위	TCP	3389	네트워크에서 Windows 인스턴스로 인터넷 게이트웨이를 거쳐 인바운드 RDP 액세스 허용.
아웃바운드			
DBServerSG 보안 그룹의 ID	TCP	1433	DBServerSG에 지정된 데이터베이스 서버에 대해 아웃바운드 Microsoft SQL Server 액세스 허용.
DBServerSG 보안 그룹의 ID	TCP	3306	DBServerSG에 지정된 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용.
0.0.0.0/0	TCP	80	인터넷에 대한 아웃바운드 HTTP 액세스 허용.
0.0.0.0/0	TCP	443	인터넷에 대한 아웃바운드 HTTPS 액세스 허용.

다음 표에서는 DBServerSG 보안 그룹에 권장되는 규칙을 설명합니다. 이 규칙은 웹 서버의 Microsoft SQL Server 및 MySQL 읽기/쓰기 요청과 네트워크에서 발생하는 SSH 및 RDP 트래픽을 허용합니다. 또한 데이터베이스 서버는 인터넷에 대한 트래픽 바운드를 시작할 수 있습니다. 라우팅 테이블에서는 이 트래픽을 가상 프라이빗 게이트웨이를 통해 전송합니다.

**DBServerSG: 권장 규칙**

인바운드			
소스	프로토콜	포트 범위	설명
WebServerSG 보안 그룹의 ID	TCP	1433	WebServerSG 보안 그룹과 연결된 웹 서버에서 인바운드 Microsoft SQL Server 액세스가 가능하도록 허용

WebServerSG 보안 그룹의 ID	TCP	3306	WebServerSG 보안 그룹과 연결된 웹 서버에서 인바운드 MySQL Server 액세스가 가능하도록 허용
네트워크의 IPv4 주소 범위	TCP	22	네트워크에서 Linux 인스턴스로의 인바운드 SSH 트래픽 허용(가상 프라이빗 게이트웨이 거침).
네트워크의 IPv4 주소 범위	TCP	3389	네트워크에서 Windows 인스턴스로의 인바운드 RDP 트래픽 허용(가상 프라이빗 게이트웨이 거침).
아웃바운드			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	가상 프라이빗 게이트웨이를 통해 인터넷에 대한 아웃바운드 IPv4 HTTP 액세스 허용(예: 소프트웨어 업데이트).
0.0.0.0/0	TCP	443	가상 프라이빗 게이트웨이를 통해 인터넷에 대한 아웃바운드 IPv4 HTTPS 액세스 허용(예: 소프트웨어 업데이트)

(선택 사항) VPC의 기본 보안 그룹에는 지정된 인스턴스가 서로 통신할 수 있도록 자동 허용하는 규칙이 있습니다. 사용자 지정 보안 그룹에 이러한 유형의 통신을 허용하려면 다음과 같은 규칙을 추가해야 합니다.

인바운드			
소스	프로토콜	포트 범위	설명
보안 그룹의 ID	모두	모두	이 보안 그룹에 지정된 다른 인스턴스로부터의 인바운드 트래픽 허용.
아웃바운드			
목적지	프로토콜	포트 범위	설명
보안 그룹의 ID	모두	모두	이 보안 그룹에 지정된 다른 인스턴스에 대해 아웃바운드 트래픽 허용

## IPv6의 보안

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, WebServerSG 및 DBServerSG 보안 그룹에 별도의 규칙을 추가하여 인스턴스에 대한 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다. 이 시나리오에서 웹 서버는 IPv6를 통해 인터넷 트래픽을 모두 수신할 수 있고, IPv6를 통해 로컬 네트워크로부터 SSH 또는 RDP 트래픽을 수신할 수 있습니다. 또한 인터넷으로의 아웃바운드 IPv6 트래픽을 시작할 수 있습니다. 데이터베이스 서버는 인터넷으로의 아웃바운드 IPv6 트래픽을 시작할 수 없으므로, 추가 보안 그룹 규칙은 필요 없습니다.

다음은 WebServerSG 보안 그룹에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

인바운드			
소스	프로토콜	포트 범위	설명

::/0	TCP	80	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTP 액세스 허용
::/0	TCP	443	어떤 IPv6 주소에서든 웹 서버로의 인바운드 HTTPS 액세스 허용
네트워크의 IPv6 주소 범위	TCP	22	(Linux 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 SSH 액세스 허용
네트워크의 IPv6 주소 범위	TCP	3389	(Windows 인스턴스) IPv6를 통한 네트워크로부터의 인바운드 RDP 액세스 허용
아웃바운드			
목적지	프로토콜	포트 범위	설명
::/0	TCP	HTTP	임의의 IPv6 주소에 대한 아웃바운드 HTTP 액세스 허용
::/0	TCP	HTTPS	임의의 IPv6 주소에 대한 아웃바운드 HTTPS 액세스 허용

## 시나리오 3 구현

시나리오 3을 구현하려면 고객 게이트웨이 관련 정보를 얻은 후 VPC 마법사를 사용하여 VPC를 생성합니다. VPC 마법사는 고객 게이트웨이 및 가상 프라이빗 게이트웨이와의 VPN 연결을 생성해줍니다.

이 절차에는 VPC용 IPv6 통신을 활성화 및 구성하기 위한 옵션 절차가 포함됩니다. VPC에서 IPv6를 사용하고 싶지 않다면 이 절차를 수행할 필요가 없습니다.

고객 게이트웨이를 준비하려면

1. 고객 게이트웨이로 사용할 디바이스를 결정합니다. 테스트를 통해 검증된 디바이스에 대한 자세한 내용은 [Amazon VPC FAQ](#)를 참조하십시오. 고객 게이트웨이 요구 사항에 대한 자세한 내용은 [Amazon VPC 네트워크 관리자 안내서](#)를 참조하십시오.
2. 고객 게이트웨이 외부 인터페이스의 인터넷 라우팅 가능 IP 주소를 얻습니다. 주소는 고정 주소여야 하며 네트워크 주소 변환(NAT)을 수행하는 디바이스를 사용할 수 있습니다.
3. 고정 라우팅된 VPN 연결을 생성하고자 하는 경우, VPN 연결을 통해 가상 프라이빗 게이트웨이로 알려야 하는 내부 IP 범위의 목록(CIDR 표기법)을 얻어야 합니다. 자세한 내용은 [VPN 라우팅 옵션 \(p. 215\)](#) 단원을 참조하십시오.

VPC 마법사를 사용하여 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 Start VPC Wizard를 선택합니다.
3. 세 번째 옵션인 VPC with Public and Private Subnets and Hardware VPN Access를 선택한 후 Select를 선택합니다.
4. VPC name, Public subnet name 및 Private subnet name에 대해 VPC와 서브넷 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다. VPC 및 서브넷에 대해 고유한 IPv4 CIDR 블록 범위를 지정하거나 기본값을 그대로 둘 수 있습니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. Public subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택하고 서브넷에 16진수 페어 값

을 지정하거나 기본값을 그대로 둡니다. Private subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 16진수 페어 값을 지정하거나 기본값을 그대로 둡니다.

6. Next를 선택합니다.
  7. Configure your VPN 페이지에서 다음을 수행한 후 Create VPC를 선택합니다.
    - Customer Gateway IP에서 VPN 라우터의 퍼블릭 IP 주소를 지정합니다.
    - 선택적으로, 고객 게이트웨이와 VPN 연결의 이름을 지정합니다.
    - Routing Type에서 다음과 같은 라우팅 옵션 중 하나를 선택합니다.
      - VPN 라우터가 BGP(Border Gateway Protocol)를 지원할 경우 Dynamic (requires BGP)을 선택합니다.
      - VPN 라우터에서 BGP를 지원하지 않으면 Static을 선택합니다. IP Prefix에서 CIDR 표기법으로 네트워크에 대한 각각의 IP 범위를 추가합니다.
- 자세한 내용은 [VPN 라우팅 옵션 \(p. 215\)](#) 단원을 참조하십시오.
8. 마법사가 완료되면 탐색 창에서 [VPN Connections]를 선택합니다. 마법사가 생성한 VPN 연결을 선택하고 [Download Configuration]을 선택합니다. 대화 상자에서 해당되는 고객 게이트웨이 공급업체, 플랫폼, 소프트웨어 버전을 선택하고 [Yes, Download]를 선택합니다.
  9. VPN 구성을 포함하는 텍스트 파일을 저장하여 본 가이드 [Amazon VPC 네트워크 관리자 안내서](#)와 함께 네트워크 관리자에게 제공합니다. 네트워크 관리자가 고객 게이트웨이를 구성할 때까지는 VPN이 작동하지 않습니다.

WebServerSG 및 DBServerSG 보안 그룹을 생성합니다. 이들 보안 그룹은 서로를 참조하므로, 먼저 보안 그룹을 생성한 후에 이 보안 그룹에 규칙을 추가해야 합니다.

#### WebServerSG 및 DBServerSG 보안 그룹을 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]을 선택합니다.
4. [Create Security Group] 대화 상자에서 보안 그룹의 이름을 WebServerSG로 지정하고 설명을 입력합니다. [VPC] 목록에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.
5. [Create Security Group]을 다시 선택합니다.
6. [Create Security Group] 대화 상자에서 보안 그룹의 이름을 DBServerSG로 지정하고 설명을 입력합니다. [VPC] 목록에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.

WebServerSG 보안 그룹에 규칙을 추가하려면 다음을 수행합니다.

1. 앞에서 만든 WebServerSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
2. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type] 목록에서 HTTP를 선택한 다음, 0.0.0.0/0를 [Source] 필드에 입력합니다.
  - b. [Add another rule]을 선택한 후 [Type] 목록에서 [HTTPS]를 선택하고 [Source] 필드에 0.0.0.0/0을 입력합니다.
  - c. [Add another rule]을 선택한 후, [Type] 목록에서 [SSH]를 선택합니다. 네트워크의 공인 IP 주소 범위를 Source(원본) 필드에 입력합니다.
  - d. Add another rule을 선택한 후, Type 목록에서 RDP를 선택합니다. 네트워크의 공인 IP 주소 범위를 Source(원본) 필드에 입력합니다.
  - e. (선택 사항, IPv6 전용) Add another rule, Type, HTTP를 차례로 선택합니다. [Source]에 ::/0을 입력합니다.
  - f. (선택 사항, IPv6 전용) Add another rule, Type, HTTPS를 차례로 선택합니다. [Source]에 ::/0을 입력합니다.



- g. (선택 사항, IPv6 전용) Add another rule, Type, SSH (Linux용) 또는 RDP(Windows용)를 선택합니다. Source에 네트워크의 IPv6 주소 범위를 입력합니다.
  - h. Save를 선택합니다.
3. [Outbound Rules] 탭에서 [Edit]를 선택한 후 다음과 같이 아웃바운드 트래픽에 대한 규칙을 추가합니다.
- a. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택합니다.
  - b. [Type] 목록에서 [MS SQL]을 선택합니다. [Destination] 필드에 DBServerSG 보안 그룹의 ID를 지정합니다.
  - c. [Add another rule]을 선택한 후 [Type] 목록에서 [MySQL]을 선택합니다. [Destination] 필드에 DBServerSG 보안 그룹의 ID를 지정합니다.
  - d. [Add another rule]을 선택한 후 [Type] 목록에서 [HTTPS]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 입력합니다.
  - e. [Add another rule]을 선택한 후, [Type] 목록에서 [HTTP]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 입력합니다.
  - f. [Save]를 선택합니다.

#### DBServerSG 보안 그룹에 권장 규칙을 추가하려면

1. 앞에서 만든 DBServerSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
2. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type] 목록에서 [SSH]를 선택한 후 [Source] 필드에 네트워크의 IP 주소 범위를 입력합니다.
  - b. [Add another rule]을 선택한 후 [Type] 목록에서 [RDP]를 선택하고 [Source] 필드에 네트워크의 IP 주소 범위를 입력합니다.
  - c. [Add another rule]을 선택한 후, [Type] 목록에서 [MS SQL]을 선택합니다. [Source] 필드에 WebServerSG 보안 그룹의 ID를 지정합니다.
  - d. [Add another rule]을 선택한 후 [Type] 목록에서 [MYSQL]을 선택합니다. [Source] 필드에 WebServerSG 보안 그룹의 ID를 지정합니다.
  - e. [Save]를 선택합니다.
3. [Outbound Rules] 탭에서 [Edit]를 선택한 후 다음과 같이 아웃바운드 트래픽에 대한 규칙을 추가합니다.
  - a. 모든 아웃바운드 트래픽을 허용하는 기본 규칙을 찾아 [Remove]를 선택합니다.
  - b. [Type] 목록에서 [HTTP]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 입력합니다.
  - c. [Add another rule]을 선택한 후 [Type] 목록에서 [HTTPS]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 입력합니다.
  - d. [Save]를 선택합니다.

네트워크 관리자가 고객 게이트웨이를 구성한 후 VPC로 인스턴스를 시작할 수 있습니다.

#### 인스턴스를 시작하려면(웹 서버 또는 데이터베이스 서버)

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. 마법사의 지침대로 진행합니다. AMI를 선택하고 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.

#### Note

IPv6 통신 용도로 인스턴스를 사용하고자 하는 경우, 지원되는 인스턴스 유형(예: T2)을 선택해야 합니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#) 단원을 참조하십시오.



4. [Configure Instance Details] 페이지의 [Network] 목록에서, 앞에서 만들었던 VPC를 선택한 후 서브넷을 선택합니다. 예를 들어 웹 서버는 퍼블릭 서브넷에서 시작하고, 데이터베이스 서버는 프라이빗 서브넷에서 시작합니다.
5. (선택 사항) 기본이 아닌 VPC에서 시작되는 인스턴스에는 퍼블릭 IPv4 주소가 할당되지 않도록 기본 설정되어 있습니다. 퍼블릭 서브넷의 인스턴스에 연결하려면 지금 퍼블릭 IPv4 주소를 지정하거나, 탄력적 IP 주소를 할당하고 인스턴스를 시작한 후 이를 인스턴스에 지정할 수 있습니다. 지금 퍼블릭 IP 주소를 지정하려면 [Auto-assign Public IP] 목록에서 [Enable]를 선택해야 합니다. 프라이빗 서브넷의 인스턴스에는 퍼블릭 IP 주소를 지정할 필요가 없습니다.

#### Note

인덱스가 eth0인 새로운 단일 네트워크 인터페이스에는 자동 할당 퍼블릭 IP 주소 기능만 사용할 수 있습니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 80\)](#) 단원을 참조하십시오.

6. (선택 사항, IPv6 전용) 서브넷 범위 내에서 인스턴스에 IPv6 주소를 자동 할당할 수 있습니다. Auto-assign IPv6 IP에 대해 Enable을 선택합니다.
7. 마법사의 다음 두 페이지에서 인스턴스의 스토리지를 구성하고 태그를 추가할 수 있습니다. [Configure Security Group] 페이지에서 [Select an existing security group] 옵션을 선택한 후, 생성한 보안 그룹 중 하나(웹 서버 인스턴스의 경우 [WebServerSG], 데이터베이스 서버 인스턴스의 경우 [DBServerSG])를 선택합니다. [Review and Launch]를 선택합니다.
8. 선택한 설정을 검토합니다. 필요한 사항을 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

VPN 전용 서브넷에서 실행하는 인스턴스의 경우 네트워크에서 해당 인스턴스에 대해 ping을 실행하여 연결을 테스트할 수 있습니다. 자세한 내용은 [인스턴스의 종단 간 연결 테스트 \(p. 219\)](#) 단원을 참조하십시오.

5단계에서 퍼블릭 서브넷의 인스턴스에 퍼블릭 IPv4 주소를 지정하지 않은 경우, 인스턴스에 연결할 수 없습니다. 퍼블릭 서브넷의 인스턴스에 액세스하려면 먼저 해당 인스턴스에 탄력적 IP 주소를 지정해야 합니다.

콘솔을 사용하여 인스턴스에 탄력적 IP 주소를 할당하고 지정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. [Allocate]를 선택합니다.

#### Note

계정이 EC2-Classical을 지원할 경우 우선 [VPC]를 선택합니다.

5. 목록에서 탄력적 IP 주소를 선택한 다음, [Actions], [Associate Address]를 선택합니다.
6. 네트워크 인터페이스 또는 인스턴스를 선택합니다. 해당 [Private IP] 목록에서 탄력적 IP 주소와 연결할 주소를 선택한 다음 [Associate]를 선택합니다.

시나리오 3에서는 퍼블릭 서브넷이 인터넷의 서버와 통신할 수 있도록 허용하는 DNS 서버가 필요하며, VPN 전용 서브넷이 네트워크의 서버와 통신할 수 있도록 허용하는 또 다른 DNS 서버가 필요합니다.

VPC의 DHCP 옵션은 자동으로 domain-name-servers=AmazonProvidedDNS으로 설정됩니다. 이것은 Amazon이 VPC에 있는 어떤 퍼블릭 서브넷이든 인터넷 게이트웨이를 통해 인터넷과 통신할 수 있도록 하기 위해 제공하는 DNS 서버입니다. 해당 DNS 서버를 제공하고 이를 VPC에서 사용하는 DNS 서버 목록에 추가해야 합니다. DHCP 옵션 설정은 수정할 수 없으므로, 해당 DNS 서버와 Amazon DNS 서버를 모두 포함하는 DHCP 옵션 설정을 만들고, 이러한 새로운 DHCP 옵션 설정을 사용하도록 VPC를 업데이트해야 합니다.

DHCP 옵션을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 [DHCP Options Sets]를 선택합니다.
3. [Create DHCP options set]를 선택합니다.
4. [Create DHCP options set] 대화 상자의 [Domain name servers] 상자에 Amazon DNS 서버 (AmazonProvidedDNS)의 주소와 해당 DNS 서버의 주소(예: 192.0.2.1)를 쉼표로 구분하여 지정한 후 [Yes, Create]를 선택합니다.
5. 탐색 창에서 [Your VPCs]를 선택합니다.
6. VPC를 선택한 다음, [Actions], [Edit DHCP Options Set]를 선택합니다.
7. [DHCP options set] 목록에서 새로운 옵션 설정의 ID를 선택하고 [Save]를 선택합니다.
8. (선택 사항) VPC는 이제 이 새로운 DHCP 옵션 설정을 사용하므로 두 DNS 서버에 모두 액세스할 수 있습니다. 원한다면, VPC가 사용한 원래의 옵션 세트를 삭제할 수 있습니다.

이제 VPC의 인스턴스에 연결할 수 있습니다. Linux 인스턴스 연결 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#)을 참조하십시오. Windows 인스턴스 연결 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

## 시나리오 4: 사설 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC

이 시나리오의 구성에는 단일 프라이빗 서브넷이 있는 Virtual Private Cloud(VPC)와, IPsec VPN 터널을 통해 귀하의 네트워크와 통신하기 위한 가상 프라이빗 게이트웨이가 포함됩니다. 인터넷을 통한 통신을 지원하는 인터넷 게이트웨이가 없습니다. 네트워크를 인터넷에 노출하지 않고 Amazon의 인프라를 사용하여 네트워크를 [클라우드](#)로 확장하려는 경우 이 시나리오를 사용하는 것이 좋습니다.

이 단원에서는 Amazon VPC 콘솔의 VPC 마법사를 사용하여 VPC와 VPN 연결을 생성한다고 가정합니다.

이 시나리오를 IPv6에 맞게 구성할 수도 있습니다. 즉 VPC 마법사를 이용해 VPC, 그리고 연결된 IPv6 CIDR 블록이 있는 서브넷을 만들 수 있습니다. 서브넷에서 시작한 인스턴스는 IPv6 주소를 받을 수 있습니다. 현재 Amazon은 VPN 연결을 통한 IPv6 통신을 지원하지 않습니다. 하지만 VPC의 인스턴스들은 IPv6를 통해서도 통신할 수 있습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

### 항목

- [개요 \(p. 44\)](#)
- [라우팅 \(p. 45\)](#)
- [보안 \(p. 46\)](#)
- [시나리오 4 구현 \(p. 47\)](#)

## 개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.

### Important

이 시나리오에 대해, [Amazon VPC 네트워크 관리자 안내서](#)에서는 네트워크 관리자가 VPN 연결의 사용자 쪽에 Amazon VPC 고객 게이트웨이를 구성하기 위해 수행해야 할 작업을 설명합니다.

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

- CIDR의 크기가 /16(예: 10.0.0.0/16)인 가상 프라이빗 클라우드(VPC). 이는 65,536개의 프라이빗 IP 주소를 제공합니다.

- CIDR의 크기가 /24(예: 10.0.0.0/24)인 VPN 전용 서브넷. 이는 256개의 프라이빗 IP 주소를 제공합니다.
- VPC와 네트워크 사이의 VPN 연결. VPN 연결은 VPC 연결의 Amazon 측에 있는 가상 프라이빗 게이트웨이와 VPN 연결의 사용자 측에 있는 고객 게이트웨이로 구성됩니다.
- 프라이빗 IP 주소가 서브넷 범위(예: 10.0.0.5, 10.0.0.6 및 10.0.0.7)에 있는 인스턴스는 서로 통신할 수 있고 또한 VPC의 다른 인스턴스와도 통신이 가능합니다.
- 서브넷과 연결된 사용자 정의 라우팅 테이블. 이 라우팅 테이블에는 서브넷의 인스턴스가 VPC의 다른 인스턴스와 통신할 수 있게 하는 경로와, 서브넷의 인스턴스가 네트워크와 직접 통신할 수 있게 하는 항목이 들어 있습니다.

서브넷에 대한 자세한 내용은 [VPC 및 서브넷 \(p. 62\)](#) 및 [VPC의 IP 주소 지정 \(p. 76\)](#)을 참조하십시오. VPN 연결에 대한 자세한 내용은 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#)을 참조하십시오. 고객 게이트웨이 구성에 대한 자세한 내용은 [Amazon VPC 네트워크 관리자 안내서](#)를 참조하십시오.

## IPv6 개요

이 시나리오에 IPv6를 사용할 수도 있습니다. 위에 나열된 구성 요소뿐 아니라 다음 요소도 구성에 포함됩니다.

- VPC와 연결된 /56 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/56). AWS는 CIDR을 자동 할당하므로 범위를 직접 선택할 수는 없습니다.
- VPN 전용 서브넷과 연결된 /64 크기의 IPv6 CIDR 블록(예: 2001:db8:1234:1a00::/64). VPC에 할당된 범위 내에서 서브넷의 범위를 선택할 수 있습니다. IPv6 CIDR의 크기는 선택할 수 없습니다.
- 서브넷 범위에서 인스턴스에 할당된 IPv6 주소(예: 2001:db8:1234:1a00::1a).
- 프라이빗 서브넷의 인스턴스가 IPv6를 사용하여 서로 통신할 수 있도록 해주는 사용자 지정 라우팅 테이블의 라우팅 테이블 항목.

## 라우팅

VPC에는 라우터가 내재되어 있습니다(이 시나리오의 구성 다이어그램 참조). 이 시나리오에서 VPC 마법사는 VPC 외부 주소로 경로가 지정된 모든 트래픽을 VPN 연결로 라우팅하는 라우팅 테이블을 생성하고, 이 라우팅 테이블을 서브넷과 연결합니다.

아래에서는 이 시나리오의 라우팅 테이블에 대해 설명합니다. 첫 번째 항목은 VPC의 로컬 라우팅에 대한 기본 항목으로서, 이 VPC의 인스턴스가 서로 통신할 수 있게 해줍니다. 두 번째 항목에서는 기타 서브넷 트래픽을 모두 가상 프라이빗 게이트웨이(예: `vgw-1a2b3c4d`)로 라우팅합니다.

목적지	Target
10.0.0.0/16	로컬
0.0.0.0/0	vgw-id

VPN 연결은 고정 라우팅 VPN 연결 또는 동적 라우팅 VPN 연결(BGP 사용)로 구성됩니다. 고정 라우팅을 선택하는 경우, VPN 연결을 생성할 때 네트워크의 IP 접두사를 수동으로 입력하라는 메시지가 표시됩니다. 동적 라우팅을 선택할 경우 BGP를 통해 VPC에 IP 접두사가 자동으로 알려집니다.

VPC의 인스턴스는 인터넷에 직접 액세스할 수 없습니다. 인터넷 바운드 트래픽은 먼저 가상 프라이빗 게이트웨이를 통과해야 귀사의 네트워크에 액세스하며, 이러한 트래픽에는 귀사의 방화벽과 기업 보안 정책이 적용됩니다. 인스턴스에서 AWS 바운드 트래픽(예: Amazon S3 또는 Amazon EC2에 대한 요청)을 전송할 경우 이 요청은 가상 프라이빗 게이트웨이를 거쳐 귀사의 네트워크에 도달한 다음에 인터넷에 도달한 후 AWS에 도달해야 합니다. 현재 VPN 연결에는 IPv6를 지원하지 않습니다.

## IPv6에 대한 라우팅

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함됩니다. 아래에서는 이 시나리오의 사용자 지정 라우팅 테이블에 대해 설명합니다. 두 번째 항목은 IPv6를 통한 VPC의 로컬 라우팅에 자동으로 추가된 기본 경로입니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	vgw-id

## 보안

AWS는 VPC의 보안을 강화하기 위해 사용할 수 있는 두 가지 기능, 보안 그룹과 네트워크 ACL을 제공합니다. 보안 그룹은 인스턴스용 인바운드 및 아웃바운드 트래픽을 제어하고, 네트워크 ACL은 서브넷용 인바운드 및 아웃바운드 트래픽을 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 맞출 수 있지만, 원하는 경우 네트워크 ACL을 사용하여 VPC에 보안 계층을 더 추가할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

시나리오 4의 경우, 네트워크 ACL이 아닌 VPC에 대한 기본 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 4를 위한 권장 규칙 \(p. 128\)](#)을 참조하십시오.

VPC에는 기본 보안 그룹이 있으며, 이 보안 그룹의 초기 설정은 인바운드 트래픽을 모두 거부하고, 아웃바운드 트래픽을 모두 허용하며, 보안 그룹에 지정된 인스턴스 간의 모든 트래픽을 허용합니다. 이 시나리오의 경우에는 네트워크에서 발생하는 SSH 트래픽(Linux) 및 원격 데스크톱 트래픽(Windows)을 허용하는 인바운드 규칙을 기본 보안 그룹에 추가하는 것이 좋습니다.

### Important

기본 보안 그룹은 배정된 인스턴스가 서로 통신하도록 자동으로 허용하므로, 이를 허용하기 위한 규칙을 추가할 필요가 없습니다. 다른 보안 그룹을 사용하는 경우 이를 허용하는 규칙을 추가해야 합니다.

다음 표에서는 VPC에 대한 기본 보안 그룹을 추가해야 하는 인바운드 규칙을 설명합니다.

### 기본 보안 그룹: 권장 규칙

인바운드			
소스	프로토콜	포트 범위	설명
네트워크의 프라이빗 IPv4 주소 범위	TCP	22	(Linux 인스턴스)네트워크로부터의 인바운드 SSH 트래픽 허용.
네트워크의 프라이빗 IPv4 주소 범위	TCP	3389	(Windows 인스턴스)네트워크로부터의 인바운드 RDP 트래픽 허용.

## IPv6의 보안

IPv6 CIDR 블록을 VPC 및 서브넷에 연결하는 경우, 보안 그룹에 별도의 규칙을 추가하여 인스턴스에 대한 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다. 이 시나리오에서 데이터베이스 서버는 IPv6를 사용하는 VPN 연결을 통해 접속할 수 없습니다. 따라서 추가 보안 그룹 규칙은 필요 없습니다.

## 시나리오 4 구현

시나리오 4를 구현하려면 고객 게이트웨이 관련 정보를 얻은 후 VPC 마법사를 사용하여 VPC를 생성합니다. VPC 마법사는 고객 게이트웨이 및 가상 프라이빗 게이트웨이와의 VPN 연결을 생성해줍니다.

고객 게이트웨이를 준비하려면

1. 고객 게이트웨이로 사용할 디바이스를 결정합니다. 테스트를 통해 검증된 디바이스에 대한 자세한 내용은 [Amazon VPC FAQ](#)를 참조하십시오. 고객 게이트웨이 요구 사항에 대한 자세한 내용은 [Amazon VPC 네트워크 관리자 안내서](#)를 참조하십시오.
2. 고객 게이트웨이 외부 인터페이스의 인터넷 라우팅 가능 IP 주소를 얻습니다. 주소는 고정 주소여야 하며 네트워크 주소 변환(NAT)을 수행하는 디바이스를 사용할 수 있습니다.
3. 고정 라우팅된 VPN 연결을 생성하고자 하는 경우, VPN 연결을 통해 가상 프라이빗 게이트웨이로 알려야 하는 내부 IP 범위의 목록(CIDR 표기법)을 얻어야 합니다. 자세한 내용은 [VPN 라우팅 옵션 \(p. 215\)](#) 단원을 참조하십시오.

VPC 마법사를 사용하여 VPC 및 VPN 연결을 생성합니다.

VPC 마법사를 사용하여 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. 네 번째 옵션인 [VPC with a Private Subnet Only and Hardware VPN Access]를 선택한 후 [Select]를 선택합니다.
4. 마법사의 첫 페이지에서 VPC 및 프라이빗 서브넷에 대한 세부 정보를 확인합니다. VPC와 서브넷에 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. Private subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 16진수 페어 값을 지정하거나 기본값을 그대로 둡니다(00).
6. [Next]를 선택합니다.
7. Configure your VPN 페이지에서 다음을 수행한 후 Create VPC를 선택합니다.
  - Customer Gateway IP에서 VPN 라우터의 퍼블릭 IP 주소를 지정합니다.
  - 선택적으로, 고객 게이트웨이와 VPN 연결의 이름을 지정합니다.
  - Routing Type에서 다음과 같은 라우팅 옵션 중 하나를 선택합니다.
    - VPN 라우터가 BGP(Border Gateway Protocol)를 지원할 경우 Dynamic (requires BGP)을 선택합니다.
    - VPN 라우터에서 BGP를 지원하지 않으면 Static을 선택합니다. IP Prefix에서 CIDR 표기법으로 네트워크에 대한 각각의 IP 범위를 추가합니다.

자세한 내용은 [VPN 라우팅 옵션 \(p. 215\)](#) 단원을 참조하십시오.
8. 마법사가 완료되면 탐색 창에서 [VPN Connections]를 선택합니다. 마법사가 생성한 VPN 연결을 선택하고 [Download Configuration]을 선택합니다. 대화 상자에서 고객 게이트웨이 공급업체, 플랫폼, 소프트웨어 버전을 선택하고 [Yes, Download]를 선택합니다.
9. VPN 구성을 포함하는 텍스트 파일을 저장하여 본 가이드 [Amazon VPC 네트워크 관리자 안내서](#)와 함께 네트워크 관리자에게 제공합니다. 네트워크 관리자가 고객 게이트웨이를 구성할 때까지는 VPN이 작동하지 않습니다.

이 시나리오 4의 경우, 기본 보안 그룹을 네트워크에서의 SSH 및 원격 데스크톱(RDP) 액세스를 허용하는 새로운 인바운드 규칙으로 업데이트해야 합니다. 인스턴스가 아웃바운드 통신을 시작하기를 원하지 않는 경우, 기본 아웃바운드 규칙을 제거할 수도 있습니다.

### 기본 보안 그룹에 대한 규칙을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택한 후 VPC의 기본 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
3. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
  - a. [Type] 목록에서 [SSH]를 선택한 후 [Source] 필드에 네트워크의 프라이빗 IP 주소 범위(예: 172.0.0.0/8)를 입력합니다.
  - b. [Add another rule]을 선택한 후 [Type] 목록에서 [RDP]를 선택하고 [Source] 필드에 네트워크의 프라이빗 IP 주소 범위를 입력합니다.
  - c. [Save]를 선택합니다.
4. (선택 사항) [Outbound Rules] 탭에서 [Edit]를 선택하고, 모든 아웃바운드 트래픽을 활성화하는 기본 규칙을 찾고, [Remove]를 선택한 다음, [Save]를 선택합니다.

네트워크 관리자가 고객 게이트웨이를 구성한 후 VPC로 인스턴스를 시작할 수 있습니다. VPC 외부의 인스턴스를 시작하는 방법이 이미 익혔다면, 인스턴스를 VPC로 시작하기 위해 알아야 할 내용을 대부분 이미 알고 있는 셈입니다.

인스턴스를 시작하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 [Launch Instance]를 선택합니다.
3. 마법사의 지침대로 진행합니다. AMI를 선택하고 인스턴스 유형을 선택한 후 [Next: Configure Instance Details]를 선택합니다.

#### Note

IPv6 통신 용도로 인스턴스를 사용하고자 하는 경우, 지원되는 인스턴스 유형(예: T2)을 선택해야 합니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#) 단원을 참조하십시오.

4. [Configure Instance Details] 페이지의 [Network] 목록에서, 앞서 생성한 VPC를 선택한 후 서브넷을 선택합니다. [Next: Add Storage]를 선택합니다.
5. 마법사의 다음 두 페이지에서 인스턴스의 스토리지를 구성하고 태그를 추가할 수 있습니다. [Configure Security Group] 페이지에서 [Select an existing security group] 옵션을 선택하고, 기본 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.
6. 선택한 설정을 검토합니다. 필요한 사항을 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

시나리오 4에서는 네트워크의 서버와 통신하려면 VPN 전용 서브넷을 사용하는 DNS 서버가 필요합니다. DNS 서버를 포함하는 새로운 DHCP 옵션 세트를 생성한 후 그 옵션 세트를 사용하도록 VPC를 구성해야 합니다.

#### Note

VPC에는 domain-name-servers=AmazonProvidedDNS를 포함한 DHCP 옵션 세트가 자동으로 생성됩니다. 이것은 Amazon이 VPC에 있는 어떤 퍼블릭 서브넷이든 인터넷 게이트웨이를 통해 인터넷과 통신할 수 있도록 하기 위해 제공하는 DNS 서버입니다. 시나리오 4에는 어떤 퍼블릭 서브넷도 없으므로, 이 DHCP 옵션 세트가 필요하지 않습니다.

### DHCP 옵션을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [DHCP Options Sets]를 선택합니다.
3. [Create DHCP Options Set]를 선택합니다.

4. [Create DHCP Options Set] 대화 상자의 [Domain name servers] 상자에 DNS 서버의 주소를 입력한 후 [Yes, Create]를 선택합니다. 이 예제에서 DNS 서버는 192.0.2.1입니다.
5. 탐색 창에서 [Your VPCs]를 선택합니다.
6. VPC를 선택한 후 [Summary] 탭에서 [Edit]를 선택합니다.
7. [DHCP options set] 목록에서 새로운 옵션 설정의 ID를 선택하고 [Save]를 선택합니다.
8. (선택 사항) VPC는 이제 이 새로운 DHCP 옵션 세트를 사용하므로 사용자의 DNS 서버를 사용합니다. 원한다면, VPC가 사용한 원래의 옵션 세트를 삭제할 수 있습니다.

이제 SSH 또는 RDP를 사용하여 VPC의 인스턴스에 연결할 수 있습니다. Linux 인스턴스 연결 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#)을 참조하십시오. Windows 인스턴스 연결 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

## 예: AWS CLI를 사용하여 IPv4 VPC 및 서브넷 생성

다음 예에서는 AWS CLI 명령을 사용하여 IPv4 CIDR 블록이 있는 기본이 아닌 VPC를 만들고, 해당 VPC에 퍼블릭 및 프라이빗 서브넷을 만듭니다. VPC와 서브넷을 만든 후, 퍼블릭 서브넷에서 인스턴스를 시작하고 여기에 연결할 수 있습니다. 시작하려면 먼저 AWS CLI를 설치하고 구성해야 합니다. 자세한 내용은 [AWS 명령줄 인터페이스로 설정](#)을 참조하십시오.

### 항목

- 1단계: VPC와 서브넷 만들기 (p. 49)
- 2단계: 서브넷을 퍼블릭으로 만들기 (p. 50)
- 3단계: 서브넷에서 인스턴스 시작 (p. 51)
- 4단계: 정리 (p. 53)

## 1단계: VPC와 서브넷 만들기

첫 번째 단계는 VPC와 서브넷 두 개를 만드는 것입니다. 이 예에서는 VPC에 대해 CIDR 블록 10.0.0.0/16을 사용하지만, 다른 CIDR 블록을 선택할 수 있습니다. 자세한 내용은 [VPC 및 서브넷 크기](#) (p. 63) 단원을 참조하십시오.

AWS CLI를 사용하여 VPC와 서브넷을 만들려면

1. CIDR 블록이 10.0.0.0/16인 VPC를 만듭니다.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
```

반환된 출력에 표시된 VPC ID를 메모해 둡니다.

```
{
  "Vpc": {
    "VpcId": "vpc-2f09a348",
    ...
  }
}
```

2. 이전 단계에서 메모해 둔 VPC ID를 사용하여 CIDR 블록이 10.0.1.0/24인 서브넷을 만듭니다.

```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.1.0/24
```

3. CIDR 블록이 10.0.0.0/24인 VPC에 두 번째 서브넷을 만듭니다.



```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.0.0/24
```

## 2단계: 서브넷을 퍼블릭으로 만들기

VPC와 서브넷을 만든 후, VPC에 인터넷 게이트웨이를 연결하고, 사용자 지정 라우팅 테이블을 만들고, 서브넷이 인터넷 게이트웨이로 라우팅되도록 구성하여 서브넷 중 하나를 퍼블릭 서브넷으로 만들 수 있습니다.

서브넷을 퍼블릭 서브넷으로 만들려면

1. 인터넷 게이트웨이 만들기.

```
aws ec2 create-internet-gateway
```

반환된 출력에 표시된 인터넷 게이트웨이 ID를 메모해 둡니다.

```
{
  "InternetGateway": {
    ...
    "InternetGatewayId": "igw-1ff7a07b",
    ...
  }
}
```

2. 이전 단계에서 메모해 둔 ID를 사용하여 VPC에 인터넷 게이트웨이를 연결합니다.

```
aws ec2 attach-internet-gateway --vpc-id vpc-2f09a348 --internet-gateway-id igw-1ff7a07b
```

3. VPC에 대해 사용자 지정 라우팅 테이블 만듭니다.

```
aws ec2 create-route-table --vpc-id vpc-2f09a348
```

반환된 출력에 표시된 라우팅 테이블 ID를 메모해 둡니다.

```
{
  "RouteTable": {
    ...
    "RouteTableId": "rtb-c1c8faa6",
    ...
  }
}
```

4. 라우팅 테이블에 모든 트래픽(0.0.0.0/0)이 인터넷 게이트웨이를 가리키는 경로를 만듭니다.

```
aws ec2 create-route --route-table-id rtb-c1c8faa6 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-1ff7a07b
```

5. 경로가 만들어졌고 활성 상태인지 확인하기 위해, 라우팅 테이블을 설명하고 그 결과를 볼 수 있습니다.

```
aws ec2 describe-route-tables --route-table-id rtb-c1c8faa6

{
  "RouteTables": [
    {
      "Associations": [],
```



```
"RouteTableId": "rtb-c1c8faa6",
"VpcId": "vpc-2f09a348",
"PropagatingVgws": [],
"Tags": [],
"Routes": [
  {
    "GatewayId": "local",
    "DestinationCidrBlock": "10.0.0.0/16",
    "State": "active",
    "Origin": "CreateRouteTable"
  },
  {
    "GatewayId": "igw-1ff7a07b",
    "DestinationCidrBlock": "0.0.0.0/0",
    "State": "active",
    "Origin": "CreateRoute"
  }
]
}
```

- 라우팅 테이블이 현재 서브넷과 연결되지 않았습니다. 서브넷에서 보내는 트래픽이 인터넷 게이트웨이로 라우팅되도록 라우팅 테이블을 VPC의 해당 서브넷과 연결해야 합니다. 먼저 `describe-subnets` 명령을 사용하여 서브넷 ID를 가져옵니다. `--filter` 옵션을 사용하여 새 VPC의 서브넷만 반환하고, `--query` 옵션을 사용하여 서브넷 ID와 그 CIDR 블록만 반환할 수 있습니다.

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-2f09a348" --query
'Subnets[*].{ID:SubnetId,CIDR:CidrBlock}'

[
  {
    "CIDR": "10.0.1.0/24",
    "ID": "subnet-b46032ec"
  },
  {
    "CIDR": "10.0.0.0/24",
    "ID": "subnet-a46032fc"
  }
]
```

- 사용자 지정 라우팅 테이블과 연결할 서브넷을 선택할 수 있습니다(예: `subnet-b46032ec`). 이 서브넷이 퍼블릭 서브넷이 됩니다.

```
aws ec2 associate-route-table --subnet-id subnet-b46032ec --route-table-id rtb-
c1c8faa6
```

- 또는 서브넷에서 시작된 인스턴스가 퍼블릭 IP 주소를 자동으로 받도록 서브넷의 퍼블릭 IP 주소 지정 동작을 수정할 수 있습니다. 그렇지 않은 경우, 시작 후 인터넷에서 연결할 수 있도록 탄력적 IP 주소를 인스턴스와 연결해야 합니다.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-b46032ec --map-public-ip-on-launch
```

### 3단계: 서브넷에서 인스턴스 시작

서브넷이 퍼블릭이고 인터넷을 통해 해당 서브넷의 인스턴스에 액세스할 수 있는지 테스트하려면, 퍼블릭 서브넷에서 인스턴스를 시작하고 여기에 연결합니다. 먼저 인스턴스와 연결할 보안 그룹과 인스턴스에 연결하는 데 사용할 키 페어를 만들어야 합니다. 보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참

조하십시오. 키 페어에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 키 페어](#)를 참조하십시오.

퍼블릭 서브넷에서 인스턴스를 시작하고 연결하려면

1. 키 페어를 만든 다음 `--query` 옵션과 `--output` 텍스트 옵션을 사용하여 확장자가 `.pem`인 파일에 직접 프라이빗 키를 파이프합니다.

```
aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
> MyKeyPair.pem
```

이 예에서는 Amazon Linux 인스턴스를 시작합니다. Linux 또는 Mac OS X 운영 체제에서 SSH 클라이언트를 사용하여 인스턴스에 연결하려면, 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

```
chmod 400 MyKeyPair.pem
```

2. VPC에 보안 그룹을 만든 다음, 어디서나 SSH 액세스를 허용하는 규칙을 추가합니다.

```
aws ec2 create-security-group --group-name SSHAccess --description "Security group for  
SSH access" --vpc-id vpc-2f09a348  
  
{  
  "GroupId": "sg-e1fb8c9a"  
}
```

```
aws ec2 authorize-security-group-ingress --group-id sg-e1fb8c9a --protocol tcp --  
port 22 --cidr 0.0.0.0/0
```

#### Note

0.0.0.0/0을 사용하면 모든 IPv4 주소에서 SSH를 사용하여 인스턴스에 액세스할 수 있습니다. 예제에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 특정 IP 주소 또는 주소 범위에 대해서만 승인하십시오.

3. 생성한 보안 그룹과 키 페어를 사용하여 퍼블릭 서브넷에서 인스턴스를 시작합니다. 출력에 표시된 인스턴스의 인스턴스 ID를 메모해 둡니다.

```
aws ec2 run-instances --image-id ami-a4827dc9 --count 1 --instance-type t2.micro --key-  
name MyKeyPair --security-group-ids sg-e1fb8c9a --subnet-id subnet-b46032ec
```

#### Note

이 예에서 AMI는 미국 동부(버지니아 북부) 리전의 Amazon Linux AMI입니다. 다른 리전에 있는 경우, 해당 리전에 적합한 AMI의 AMI ID가 필요합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Finding a Linux AMI](#) 단원을 참조하십시오.

4. 인스턴스에 연결하려면 인스턴스가 `running` 상태에 있어야 합니다. 인스턴스를 설명하고 인스턴스의 상태를 확인한 다음, 해당 퍼블릭 IP 주소를 메모해 둡니다.

```
aws ec2 describe-instances --instance-id i-0146854b7443af453  
  
{  
  "Reservations": [  
    {  
      ...  
      "Instances": [  
        {
```

```
...
  "State": {
    "Code": 16,
    "Name": "running"
  },
  ...
  "PublicIpAddress": "52.87.168.235",
  ...
}
]
}
```

5. 인스턴스가 실행 상태에 있는 경우, 다음 명령을 사용하여 Linux 또는 Mac OS X 컴퓨터에서 SSH 클라이언트를 통해 인스턴스에 연결할 수 있습니다.

```
ssh -i "MyKeyPair.pem" ec2-user@52.87.168.235
```

Windows 컴퓨터에서 연결하려는 경우, [PuTTY](#)를 사용하여 Windows에서 Linux 인스턴스에 연결 지침을 사용합니다.

## 4단계: 정리

인스턴스에 연결할 수 있는지 확인한 후, 인스턴스가 더 이상 필요하지 않은 경우 종료할 수 있습니다. 이렇게 하려면 `terminate-instances` 명령을 사용합니다. 이 예에서 만든 리소스를 삭제하려면 다음 명령을 나열된 순서대로 사용합니다.

1. 보안 그룹 삭제:

```
aws ec2 delete-security-group --group-id sg-e1fb8c9a
```

2. 서브넷 삭제:

```
aws ec2 delete-subnet --subnet-id subnet-b46032ec
```

```
aws ec2 delete-subnet --subnet-id subnet-a46032fc
```

3. 사용자 지정 라우팅 테이블 삭제:

```
aws ec2 delete-route-table --route-table-id rtb-c1c8faa6
```

4. VPC에서 인터넷 게이트웨이 분리:

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-1ff7a07b --vpc-id vpc-2f09a348
```

5. 인터넷 게이트웨이 삭제:

```
aws ec2 delete-internet-gateway --internet-gateway-id igw-1ff7a07b
```

6. VPC 삭제:

```
aws ec2 delete-vpc --vpc-id vpc-2f09a348
```

## 예: AWS CLI를 사용하여 IPv6 VPC 및 서브넷 생성

다음 예에서는 AWS CLI 명령을 사용하여 IPv6 CIDR 블록이 있는 기본이 아닌 VPC, 퍼블릭 서브넷, 그리고 아웃바운드 인터넷 액세스만 가능한 프라이빗 서브넷을 만듭니다. VPC와 서브넷을 만든 후, 퍼블릭 서브넷에서 인스턴스를 시작하고 여기에 연결할 수 있습니다. 프라이빗 서브넷에서 인스턴스를 시작하고 인터넷에 연결할 수 있는지 확인할 수 있습니다. 시작하려면 먼저 AWS CLI를 설치하고 구성해야 합니다. 자세한 내용은 [AWS 명령줄 인터페이스로 설정](#)을 참조하십시오.

### 항목

- 1단계: VPC와 서브넷 만들기 (p. 54)
- 2단계: 퍼블릭 서브넷 구성 (p. 55)
- 3단계: 외부 전용 프라이빗 서브넷 구성 (p. 57)
- 4단계: 서브넷의 IPv6 주소 지정 동작 변경 (p. 58)
- 5단계: 퍼블릭 서브넷에서 인스턴스 시작 (p. 58)
- 6단계: 프라이빗 서브넷으로 인스턴스를 시작 (p. 59)
- 7단계: 정리 (p. 61)

## 1단계: VPC와 서브넷 만들기

첫 번째 단계는 VPC와 서브넷 두 개를 만드는 것입니다. 이 예에서는 VPC에 대해 IPv4 CIDR 블록 10.0.0.0/16을 사용하지만, 다른 CIDR 블록을 선택할 수 있습니다. 자세한 내용은 [VPC 및 서브넷 크기](#) (p. 63) 단원을 참조하십시오.

AWS CLI를 사용하여 VPC와 서브넷을 만들려면

1. 10.0.0.0/16 CIDR 블록이 있는 VPC를 생성하여 IPv6 CIDR 블록을 VPC와 연결합니다.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --amazon-provided-ipv6-cidr-block
```

반환된 출력에 표시된 VPC ID를 메모해 둡니다.

```
{
  "Vpc": {
    "VpcId": "vpc-2f09a348",
    ...
  }
}
```

2. VPC를 설명하여 VPC에 연결된 IPv6 CIDR 블록을 가져옵니다.

```
aws ec2 describe-vpcs --vpc-id vpc-2f09a348

{
  "Vpcs": [
    {
      ...
      "Ipv6CidrBlockAssociationSet": [
        {
          "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
          "AssociationId": "vpc-cidr-assoc-17a5407e",
          "Ipv6CidrBlockState": {
            "State": "ASSOCIATED"
          }
        }
      ]
    },
    ...
  ]
}
```

```
}
```

3. IPv4 CIDR 블록이 10.0.0.0/24이고 IPv6 CIDR 블록이 2001:db8:1234:1a00::/64(이전 단계에서 반환된 범위에 속함)인 서브넷을 만듭니다.

```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.0.0/24 --ipv6-cidr-block 2001:db8:1234:1a00::/64
```

4. IPv4 CIDR 블록이 10.0.1.0/24이고 IPv6 CIDR 블록이 2001:db8:1234:1a01::/64인 VPC에 두 번째 서브넷을 생성합니다.

```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.1.0/24 --ipv6-cidr-block 2001:db8:1234:1a01::/64
```

## 2단계: 퍼블릭 서브넷 구성

VPC와 서브넷을 만든 후, VPC에 인터넷 게이트웨이를 연결하고, 사용자 지정 라우팅 테이블을 만들고, 서브넷이 인터넷 게이트웨이로 라우팅되도록 구성하여 서브넷 중 하나를 퍼블릭 서브넷으로 만들 수 있습니다. 이 예시에서는 모든 IPv4 트래픽과 IPv6 트래픽을 인터넷 게이트웨이로 라우팅하는 라우팅 테이블이 생성됩니다.

서브넷을 퍼블릭 서브넷으로 만들려면

1. 인터넷 게이트웨이 만들기.

```
aws ec2 create-internet-gateway
```

반환된 출력에 표시된 인터넷 게이트웨이 ID를 메모해 둡니다.

```
{
  "InternetGateway": {
    ...
    "InternetGatewayId": "igw-1ff7a07b",
    ...
  }
}
```

2. 이전 단계에서 메모해 둔 ID를 사용하여 VPC에 인터넷 게이트웨이를 연결합니다.

```
aws ec2 attach-internet-gateway --vpc-id vpc-2f09a348 --internet-gateway-id igw-1ff7a07b
```

3. VPC에 대해 사용자 지정 라우팅 테이블 만듭니다.

```
aws ec2 create-route-table --vpc-id vpc-2f09a348
```

반환된 출력에 표시된 라우팅 테이블 ID를 메모해 둡니다.

```
{
  "RouteTable": {
    ...
    "RouteTableId": "rtb-c1c8faa6",
    ...
  }
}
```

- 라우팅 테이블에 모든 IPv6 트래픽(::/0)이 인터넷 게이트웨이를 가리키는 경로를 만듭니다.

```
aws ec2 create-route --route-table-id rtb-c1c8faa6 --destination-ipv6-cidr-block ::/0  
--gateway-id igw-1ff7a07b
```

#### Note

또한 IPv4 트래픽에 대해 퍼블릭 서브넷을 사용하려면 인터넷 게이트웨이를 가리키는 0.0.0.0/0 트래픽에 대해 또 다른 경로를 추가해야 합니다.

- 경로가 만들어졌고 활성 상태인지 확인하기 위해, 라우팅 테이블을 설명하고 그 결과를 볼 수 있습니다.

```
aws ec2 describe-route-tables --route-table-id rtb-c1c8faa6  
  
{  
  "RouteTables": [  
    {  
      "Associations": [],  
      "RouteTableId": "rtb-c1c8faa6",  
      "VpcId": "vpc-2f09a348",  
      "PropagatingVgws": [],  
      "Tags": [],  
      "Routes": [  
        {  
          "GatewayId": "local",  
          "DestinationCidrBlock": "10.0.0.0/16",  
          "State": "active",  
          "Origin": "CreateRouteTable"  
        },  
        {  
          "GatewayId": "local",  
          "Origin": "CreateRouteTable",  
          "State": "active",  
          "DestinationIpv6CidrBlock": "2001:db8:1234:1a00::/56"  
        },  
        {  
          "GatewayId": "igw-1ff7a07b",  
          "Origin": "CreateRoute",  
          "State": "active",  
          "DestinationIpv6CidrBlock": "::/0"  
        }  
      ]  
    }  
  ]  
}
```

- 라우팅 테이블이 현재 서브넷과 연결되어 있지 않습니다. 서브넷에서 보내는 트래픽이 인터넷 게이트웨이로 라우팅되도록 라우팅 테이블을 VPC의 해당 서브넷과 연결합니다. 먼저 서브넷을 설명하여 ID를 얻습니다. --filter 옵션을 사용하여 새 VPC의 서브넷만 반환하고, --query 옵션을 사용하여 서브넷 ID와 그 IPv4 및 IPv6 CIDR 블록만 반환할 수 있습니다.

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-2f09a348" --query  
'Subnets[*].  
{ID:SubnetId,IPv4CIDR:CidrBlock,IPv6CIDR:Ipv6CidrBlockAssociationSet[*].Ipv6CidrBlock}'  
  
[  
  {  
    "IPv6CIDR": [  
      "2001:db8:1234:1a00::/64"  
    ],  
    "ID": "subnet-b46032ec",  
    "IPv4CIDR": "10.0.0.0/24"  
  },  
]
```

```
{
  "IPv6CIDR": [
    "2001:db8:1234:1a01::/64"
  ],
  "ID": "subnet-a46032fc",
  "IPv4CIDR": "10.0.1.0/24"
}
```

7. 사용자 지정 라우팅 테이블과 연결할 서브넷을 선택할 수 있습니다(예: subnet-b46032ec). 이 서브넷이 퍼블릭 서브넷이 됩니다.

```
aws ec2 associate-route-table --subnet-id subnet-b46032ec --route-table-id rtb-c1c8faa6
```

## 3단계: 외부 전용 프라이빗 서브넷 구성

VPC의 두 번째 서브넷이 IPv6 외부 전용 프라이빗 서브넷이 되도록 구성할 수 있습니다. 이 서브넷에서 시작하는 인스턴스는 외부 전용 인터넷 게이트웨이를 사용하여 IPv6를 통해 인터넷에 액세스할 수 있지만(예: 소프트웨어 업데이트 작업), 인터넷의 호스트는 인스턴스에 접속할 수 없습니다.

서브넷을 외부 전용 프라이빗 서브넷으로 만들려면

1. VPC에 외부 전용 인터넷 게이트웨이를 생성합니다. 반환된 출력에 표시된 게이트웨이 ID를 메모해 둡니다.

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-2f09a348

{
  "EgressOnlyInternetGateway": {
    "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
    "Attachments": [
      {
        "State": "attached",
        "VpcId": "vpc-2f09a348"
      }
    ]
  }
}
```

2. VPC에 대해 사용자 지정 라우팅 테이블 만듭니다. 반환된 출력에 표시된 라우팅 테이블 ID를 메모해 둡니다.

```
aws ec2 create-route-table --vpc-id vpc-2f09a348
```

3. 라우팅 테이블에 모든 IPv6 트래픽(::

```
aws ec2 create-route --route-table-id rtb-abc123ab --destination-ipv6-cidr-block ::/0 --egress-only-internet-gateway-id eigw-015e0e244e24dfe8a
```

4. 라우팅 테이블을 VPC의 두 번째 서브넷에 연결합니다(앞 섹션에서 정의한 서브넷). 이 서브넷은 외부 전용 IPv6 인터넷 액세스가 가능한 프라이빗 서브넷이 됩니다.

```
aws ec2 associate-route-table --subnet-id subnet-a46032fc --route-table-id rtb-abc123ab
```

## 4단계: 서브넷의 IPv6 주소 지정 동작 변경

서브넷에서 시작된 인스턴스가 IPv6 주소를 자동으로 받도록 서브넷의 IP 주소 지정 동작을 수정할 수 있습니다. 서브넷에서 인스턴스를 시작할 경우, 단일 IPv6 주소는 서브넷의 주소 범위로부터 인스턴스의 주 네트워크 인터페이스(eth0)에 할당됩니다.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-b46032ec --assign-ipv6-address-on-creation
```

```
aws ec2 modify-subnet-attribute --subnet-id subnet-a46032fc --assign-ipv6-address-on-creation
```

## 5단계: 퍼블릭 서브넷에서 인스턴스 시작

퍼블릭 서브넷이 퍼블릭인지, 그리고 인터넷을 통해 해당 서브넷의 인스턴스에 액세스할 수 있는지 테스트하려면, 퍼블릭 서브넷에서 인스턴스를 시작하여 여기에 연결합니다. 먼저 인스턴스와 연결할 보안 그룹과 인스턴스에 연결하는 데 사용할 키 페어를 만들어야 합니다. 보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참조하십시오. 키 페어에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 키 페어](#)를 참조하십시오.

퍼블릭 서브넷에서 인스턴스를 시작하고 연결하려면

1. 키 페어를 만든 다음 `--query` 옵션과 `--output` 텍스트 옵션을 사용하여 확장자가 `.pem`인 파일에 직접 프라이빗 키를 파이프합니다.

```
aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem
```

이 예에서는 Amazon Linux 인스턴스를 시작합니다. Linux 또는 OS X 운영 체제에서 SSH 클라이언트를 사용하여 인스턴스에 연결하려면, 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정합니다.

```
chmod 400 MyKeyPair.pem
```

2. VPC에 보안 그룹을 만든 다음, 어떤 IPv6 주소로부터 이루어지는 SSH 액세스도 허용하는 규칙을 추가합니다.

```
aws ec2 create-security-group --group-name SSHAccess --description "Security group for SSH access" --vpc-id vpc-2f09a348  
  
{  
  "GroupId": "sg-e1fb8c9a"  
}
```

```
aws ec2 authorize-security-group-ingress --group-id sg-e1fb8c9a --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "Ipv6Ranges": [{"CidrIpv6": "::0"}]}
```

### Note

`::/0`을 사용하면 모든 IPv6 주소에서 SSH를 사용하여 인스턴스에 액세스할 수 있습니다. 예제에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 특정 IP 주소 또는 주소 범위만 인스턴스에 액세스하도록 승인하십시오.



3. 생성한 보안 그룹과 키 페어를 사용하여 퍼블릭 서브넷에서 인스턴스를 시작합니다. 출력에 표시된 인스턴스의 인스턴스 ID를 메모해 둡니다.

```
aws ec2 run-instances --image-id ami-a4827dc9 --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-e1fb8c9a --subnet-id subnet-b46032ec
```

#### Note

이 예에서 AMI는 미국 동부(버지니아 북부) 리전의 Amazon Linux AMI입니다. 다른 리전에 있는 경우, 해당 리전에 적합한 AMI의 AMI ID가 필요합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Finding a Linux AMI](#) 단원을 참조하십시오.

4. 인스턴스에 연결하려면 인스턴스가 `running` 상태에 있어야 합니다. 인스턴스를 설명하고 인스턴스의 상태를 확인한 다음, 해당 IPv6 주소를 메모해 둡니다.

```
aws ec2 describe-instances --instance-id i-0146854b7443af453

{
  "Reservations": [
    {
      ...
      "Instances": [
        {
          ...
          "State": {
            "Code": 16,
            "Name": "running"
          },
          ...
          "NetworkInterfaces": {
            "Ipv6Addresses": {
              "Ipv6Address": "2001:db8:1234:1a00::123"
            }
          }
          ...
        }
      ]
    }
  ]
}
```

5. 인스턴스가 실행 상태에 있는 경우, 다음 명령을 사용하여 Linux 또는 OS X 컴퓨터에서 SSH 클라이언트를 통해 인스턴스에 연결할 수 있습니다. 로컬 컴퓨터에 IPv6 주소가 구성되어 있어야 합니다.

```
ssh -i "MyKeyPair.pem" ec2-user@2001:db8:1234:1a00::123
```

Windows 컴퓨터에서 연결하려는 경우, PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결 지침을 사용합니다.

## 6단계: 프라이빗 서브넷으로 인스턴스를 시작

외부 전용 프라이빗 서브넷의 인스턴스가 인터넷에 접속할 수 있는지 테스트하려면 프라이빗 서브넷에서 인스턴스를 시작하여 퍼블릭 서브넷의 배스천 인스턴스를 사용하여 그 인스턴스에 연결합니다(앞 섹션에서 시작한 인스턴스를 사용할 수 있음). 먼저 그 인스턴스에 대한 보안 그룹을 생성해야 합니다. 보안 그룹에는 배스천 인스턴스가 SSH를 사용하여 연결하도록 허용하는 규칙, 그리고 `ping6` 명령(ICMPv6 트래픽)을 허용하는 규칙이 있어야 합니다.

1. VPC에 보안 그룹을 생성하고 퍼블릭 서브넷에서 해당 인스턴스의 IPv6 주소로부터 접근하는 인바운드 SSH 액세스를 허용하는 규칙, 그리고 모든 ICMPv6 트래픽을 허용하는 규칙을 다음과 같이 추가합니다.

```
aws ec2 create-security-group --group-name SSHAccessRestricted --description "Security group for SSH access from bastion" --vpc-id vpc-2f09a348

{
  "GroupId": "sg-aabb1122"
}
```

```
aws ec2 authorize-security-group-ingress --group-id sg-aabb1122 --ip-permissions ' [{"IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "Ipv6Ranges": [{"CidrIpv6": "2001:db8:1234:1a00::123/128"}]} ]'
```

```
aws ec2 authorize-security-group-ingress --group-id sg-aabb1122 --ip-permissions ' [{"IpProtocol": "58", "FromPort": -1, "ToPort": -1, "Ipv6Ranges": [{"CidrIpv6": "::0"}]} ]'
```

2. 생성한 보안 그룹과 퍼블릭 서브넷에서 인스턴스를 시작하기 위해 사용했던 키 페어를 사용하여 프라이빗 서브넷에서 인스턴스를 시작합니다.

```
aws ec2 run-instances --image-id ami-a4827dc9 --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-aabb1122 --subnet-id subnet-a46032fc
```

describe-instances 명령을 사용하여 인스턴스가 실행 중인지 확인하고 IPv6 주소를 얻습니다.

3. 로컬 컴퓨터에서 SSH 에이전트 전달을 구성한 다음, 퍼블릭 서브넷의 인스턴스에 접속합니다. Linux의 경우 다음 명령을 사용합니다.

```
ssh-add MyKeyPair.pem
```

```
ssh -A ec2-user@2001:db8:1234:1a00::123
```

OS X의 경우 다음 명령을 사용합니다.

```
ssh-add -K MyKeyPair.pem
```

```
ssh -A ec2-user@2001:db8:1234:1a00::123
```

Windows에서는 [Windows\(PuTTY\)에 대한 SSH 에이전트 전달을 구성하려면 \(p. 177\)](#)의 지침을 사용합니다. 인스턴스의 IPv6 주소를 사용하여 퍼블릭 서브넷의 인스턴스에 접속합니다.

4. 퍼블릭 서브넷의 인스턴스(배스천 인스턴스)에서 IPv6 주소를 사용하여 프라이빗 서브넷의 인스턴스에 접속합니다.

```
ssh ec2-user@2001:db8:1234:1a01::456
```

5. 프라이빗 인스턴스에서 ICMP를 활성화한 웹 사이트에 대해 ping6 명령을 실행하여 인터넷에 연결할 수 있는지 테스트합니다. 예를 들면 다음과 같습니다.

```
ping6 -n ietf.org
```

6. 인터넷의 호스트가 프라이빗 서브넷의 인스턴스에 접속할 수 없는지 테스트하려면 IPv6를 사용할 수 있는 컴퓨터에서 ping6 명령을 사용합니다. 제한 시간 응답을 얻어야 합니다.

```
ping6 2001:db8:1234:1a01::456
```

## 7단계: 정리

퍼블릭 서브넷의 인스턴스에 접속할 수 있고 프라이빗 서브넷의 인스턴스가 인터넷에 접속할 수 있음을 확인한 후에는, 필요 없는 인스턴스를 종료할 수 있습니다. 이렇게 하려면 `terminate-instances` 명령을 사용합니다. 이 예에서 만든 리소스를 삭제하려면 다음 명령을 나열된 순서대로 사용합니다.

### 1. 보안 그룹 삭제:

```
aws ec2 delete-security-group --group-id sg-e1fb8c9a
```

```
aws ec2 delete-security-group --group-id sg-aabb1122
```

### 2. 서브넷 삭제:

```
aws ec2 delete-subnet --subnet-id subnet-b46032ec
```

```
aws ec2 delete-subnet --subnet-id subnet-a46032fc
```

### 3. 사용자 지정 라우팅 테이블 삭제:

```
aws ec2 delete-route-table --route-table-id rtb-c1c8faa6
```

```
aws ec2 delete-route-table --route-table-id rtb-abc123ab
```

### 4. VPC에서 인터넷 게이트웨이 분리:

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-1ff7a07b --vpc-id vpc-2f09a348
```

### 5. 인터넷 게이트웨이 삭제:

```
aws ec2 delete-internet-gateway --internet-gateway-id igw-1ff7a07b
```

### 6. 외부 전용 인터넷 게이트웨이 삭제:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id ewigw-015e0e244e24dfe8a
```

### 7. VPC 삭제:

```
aws ec2 delete-vpc --vpc-id vpc-2f09a348
```

# VPC 및 서브넷

Amazon Virtual Private Cloud(Amazon VPC)를 시작하려면 먼저 VPC와 서브넷을 만듭니다. Amazon VPC의 일반 개요는 [Amazon VPC란 무엇인가? \(p. 1\)](#)를 참조하십시오.

## 항목

- [VPC 및 서브넷 기본 사항 \(p. 62\)](#)
- [VPC 및 서브넷 크기 \(p. 63\)](#)
- [서브넷 라우팅 \(p. 64\)](#)
- [서브넷 보안 \(p. 65\)](#)
- [로컬 네트워크 및 기타 VPC와의 연결 \(p. 65\)](#)
- [VPC 및 서브넷 관련 작업 \(p. 66\)](#)
- [CLI 개요 \(p. 70\)](#)

## VPC 및 서브넷 기본 사항

Virtual Private Cloud(VPC)는 사용자의 AWS 계정 전용 가상 네트워크입니다. VPC는 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 분리되어 있습니다. Amazon EC2 인스턴스와 같은 AWS 리소스를 VPC에서 실행할 수 있습니다.

VPC를 만들 때 VPC의 IPv4 주소의 범위를 CIDR(Classless Inter-Domain Routing) 블록 형태로 지정해야 합니다(예: 10.0.0.0/16). CIDR 표기법에 대한 자세한 내용은 [RFC 4632](#)을 참조하십시오.

다음 다이어그램은 IPv4 CIDR 블록이 있는 새 VPC와 기본 라우팅 테이블을 보여줍니다.

VPC를 만들면 리전의 모든 가용 영역에 적용됩니다. VPC를 만든 후 각 가용 영역에 하나 이상의 서브넷을 추가할 수 있습니다. 서브넷을 만들 때 해당 서브넷에 대한 CIDR 블록을 지정합니다. 이는 VPC CIDR 블록의 서브넷입니다. 각 서브넷은 단일 가용 영역 내에서만 존재해야 하며, 여러 영역으로 확장할 수 없습니다. 각 가용 영역은 다른 가용 영역에서 발생한 장애를 격리시킬 수 있도록 서로 분리된 공간입니다. 별도의 가용 영역에서 인스턴스를 시작함으로써 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다. AWS는 각 서브넷에 고유 ID를 할당합니다.

IPv6 CIDR 블록을 VPC와 서브넷 각각 할당할 수도 있습니다.

다음 그림은 여러 가용 영역의 서브넷으로 구성된 VPC를 보여 줍니다. 1A, 1B, 2A 및 3A는 VPC에 있는 인스턴스들입니다. IPv6 CIDR 블록 하나는 VPC와 연결되어 있고, IPv6 CIDR 블록 하나는 서브넷 1과 연결되어 있습니다. 인터넷 게이트웨이를 통해 인터넷에서 통신할 수 있으며, 가상 프라이빗 네트워크(VPN) 연결을 통해 기업 네트워크와 통신할 수 있습니다.

서브넷 트래픽이 인터넷 게이트웨이로 라우팅되는 경우 해당 서브넷을 퍼블릭 서브넷이라고 합니다. 이 그림에서는 서브넷 1이 퍼블릭 서브넷입니다. 퍼블릭 서브넷의 인스턴스가 IPv4를 통해 인터넷과 통신할 수 있게 하려면 인스턴스에 퍼블릭 IPv4 주소 또는 탄력적 IP 주소(IPv4)가 있어야 합니다. 퍼블릭 IPv4 주소에 대한 자세한 내용은 [퍼블릭 IPv4 주소 \(p. 78\)](#) 단원을 참조하십시오. 퍼블릭 서브넷의 인스턴스가 IPv6를 통해 인터넷과 통신할 수 있으려면 인스턴스에 퍼블릭 IPv6 주소가 있어야 합니다.

인터넷 게이트웨이로 라우팅되지 않는 서브넷을 프라이빗 서브넷이라고 합니다. 이 그림에서는 서브넷 2가 프라이빗 서브넷입니다.

서브넷이 인터넷 게이트웨이에 이르는 경로를 갖고 있지 않지만 그 트래픽이 VPN 연결을 위한 가상 프라이빗 게이트웨이로 라우팅되는 경우, 이 서브넷을 VPN 전용 서브넷이라고 합니다. 이 그림에서는 서브넷 3이 VPN 전용 서브넷입니다. 현재 VPN 연결을 통한 IPv6 트래픽은 지원하지 않습니다.

자세한 내용은 [시나리오 및 예시 \(p. 17\)](#), [인터넷 게이트웨이 \(p. 165\)](#) 또는 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#) 단원을 참조하십시오.

#### Note

서브넷의 유형과 관계없이 서브넷의 내부 IPv4 주소 범위는 항상 프라이빗입니다. 즉 인터넷으로 주소 블록을 알리지 않습니다.

계정에서 만들 수 있는 VPC 및 서브넷 수에는 제한이 있습니다. 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

## VPC 및 서브넷 크기

Amazon VPC는 IPv4 및 IPv6 주소 지정을 지원하고 각각에 대해 다양한 CIDR 블록 크기 제한을 둡니다. 기본적으로 모든 VPC와 서브넷에는 IPv4 CIDR 블록이 있습니다. 이 동작은 변경할 수 없습니다. IPv6 CIDR 블록을 VPC에 연결할지 여부를 선택할 수 있습니다.

### IPv4의 경우, VPC 및 서브넷 크기 조정

VPC에 단일 CIDR 블록을 지정할 수 있습니다. 허용된 블록 크기는 /16 넷마스크 ~ /28 넷마스크입니다. 따라서 VPC는 16 ~ 65,536개의 IP 주소를 포함할 수 있습니다.

VPC를 생성하는 경우, 다음과 같이 [RFC 1918](#) 규격에 따라 프라이빗(비공개적으로 라우팅 가능) IPv4 주소 범위에 속하는 CIDR 블록을 지정하는 것이 좋습니다.

- 10.0.0.0 - 10.255.255.255 (10/8 접두사)
- 172.16.0.0 - 172.31.255.255 (172.16/12 접두사)
- 192.168.0.0 - 192.168.255.255 (192.168/16 접두사)

RFC 1918에 지정된 프라이빗 IPv4 주소 범위에 속하지 않는 공개적으로 라우팅 가능한 CIDR 블록을 사용하여 VPC를 생성할 수 있지만, 이 설명서에서 프라이빗 IP 주소는 VPC의 CIDR 범위 내에 있는 IPv4 주소를 가리킵니다.

VPC를 생성한 후에는 크기를 변경할 수 없습니다. VPC가 너무 작아 요구 사항을 충족하지 못할 경우 더 큰 VPC를 생성한 후 새 VPC로 인스턴스를 마이그레이션해야 합니다. 이렇게 하려면 실행 중인 인스턴스에서 AMI를 생성한 후 더 큰 새 VPC에서 대체 인스턴스를 시작합니다. 그런 다음 이전 인스턴스를 종료한 후 기존의 작은 VPC를 삭제하면 됩니다. 자세한 내용은 [VPC 삭제 \(p. 69\)](#) 단원을 참조하십시오.

서브넷의 CIDR 블록은 VPC(VPC에서 단일 서브넷을 사용할 경우) 또는 하위 집합(VPC에서 여러 서브넷을 사용할 경우)의 CIDR 블록과 동일하게 지정할 수 있습니다. 허용된 블록 크기는 /28 넷마스크 ~ /16 넷마스크입니다. VPC에 두 개 이상의 서브넷을 만들 경우, 서브넷의 CIDR 블록이 겹치지 않아야 합니다.

예를 들어 CIDR 블록이 10.0.0.0/24인 VPC를 만들 경우 256개의 IP 주소를 지원합니다. 이 CIDR 블록을 각각 128개의 IP 주소를 지원하는 2개의 서브넷으로 나눌 수 있습니다. 한 서브넷은 10.0.0.0/25 CIDR 블록(10.0.0.0 ~ 10.0.0.127 사이의 주소)을, 다른 서브넷은 10.0.0.128/25 CIDR 블록(10.0.0.128 ~ 10.0.0.255 사이의 주소)을 사용합니다.

서브넷 CIDR 블록을 계산하는 데 유용한 여러 도구가 있습니다. 예를 보려면 <http://www.subnet-calculator.com/cidr.php>를 참조하십시오. 또한 네트워크 엔지니어링 그룹의 도움을 받아 서브넷에 지정할 CIDR 블록을 정할 수도 있습니다.

각 서브넷 CIDR 블록에서 첫 4개의 IP 주소와 마지막 IP 주소는 사용자가 사용할 수 없으므로 인스턴스에 할당할 수 없습니다. 예를 들어 10.0.0.0/24 CIDR 블록의 서브넷에서는 다음 5개 IP 주소가 예약되어 있습니다.

- 10.0.0.0: 네트워크 주소.
- 10.0.0.1: AWS에서 VPC 라우터용으로 예약한 주소.
- 10.0.0.2: AWS에서 예약한 주소. DNS 서버의 IP 주소는 항상 VPC 네트워크 범위를 기초로 2를 더한 것입니다. 그러나 예에서는 각 서브넷 범위를 기초로 2를 더한 것도 예약합니다. 자세한 내용은 [Amazon DNS 서버 \(p. 191\)](#) 단원을 참조하십시오.
- 10.0.0.3: AWS에서 앞으로 사용하려고 예약한 주소.
- 10.0.0.255: 네트워크 브로드캐스트 주소. VPC에서는 브로드캐스트를 지원하지 않으므로, 이 주소를 예약합니다.

## IPv6의 경우, VPC 및 서브넷 크기 조정

단일 IPv6 CIDR 블록을 계정의 기존 VPC와 연결하거나 새 VPC를 생성하여 연결할 수 있습니다. CIDR 블록은 /56의 고정 접두사 길이를 사용합니다. 주소 범위나 IPv6 CIDR 블록 크기를 선택할 수 없습니다. Amazon은 해당 블록을 자체 IPv6 주소 풀로부터 VPC에 할당합니다.

IPv6 CIDR 블록을 VPC와 연결한 경우, IPv6 CIDR 블록을 VPC의 기존 서브넷 또는 새로 생성한 서브넷과 연결할 수 있습니다. 서브넷의 IPv6 CIDR 블록은 /64의 고정 접두사 길이를 사용합니다.

예를 들어, VPC를 생성하고 이 VPC에 IPv6 CIDR 블록을 연결하도록 지정합니다. Amazon은 VPC에 IPv6 CIDR 블록 2001:db8:1234:1a00::/56을 할당합니다. 서브넷을 생성하고 이 범위에 속하는 IPv6 CIDR 블록을 연결할 수 있습니다(예: 2001:db8:1234:1a00::/64).

서브넷에 연결된 IPv6 CIDR 블록을 분리하고, VPC에 연결된 IPv6 CIDR 블록을 분리할 수 있습니다. VPC에 연결된 IPv6 CIDR 블록을 분리한 후 나중에 다시 VPC에 IPv6 CIDR 블록을 연결하는 경우, 동일한 CIDR을 받을 것으로 기대할 수는 없습니다.

## 서브넷 라우팅

각 서브넷은 서브넷 외부로 나가는 아웃바운드 트래픽에 대해 허용된 경로를 지정하는 라우팅 테이블이 연결되어 있어야 합니다. 생성된 각 서브넷은 자동으로 VPC의 기본 라우팅 테이블에 연결됩니다. 테이블 연결 기본 라우팅 테이블의 내용을 변경할 수 있습니다. 자세한 내용은 [라우팅 테이블 \(p. 154\)](#) 단원을 참조하십시오.

앞의 다이어그램에서 서브넷 1에 연결된 라우팅 테이블은 모든 IPv4 트래픽(0.0.0.0/0)과 IPv6 트래픽 (::/0)을 인터넷 게이트웨이(예: igw-1a2b3c4d)로 라우팅합니다. 1A 인스턴스에는 IPv4 탄력적 IP 주소가 있고 1B 인스턴스에는 IPv6 주소가 있으므로 각각 IPv4 및 IPv6를 통해 인터넷으로 접속할 수 있습니다.

## Note

(IPv4 전용) 인스턴스에 연결된 탄력적 IPv4 주소 또는 퍼블릭 IPv4 주소는 VPC의 인터넷 게이트웨이를 통해 액세스할 수 있습니다. 인스턴스 및 다른 네트워크 간 VPN 연결을 거치는 트래픽은 인터넷 게이트웨이가 아닌 가상 프라이빗 게이트웨이를 통과하며, 따라서 탄력적 IPv4 주소 또는 퍼블릭 IPv4 주소에 액세스하지 않습니다.

2A 인스턴스는 인터넷에는 도달할 수 없지만 VPC의 다른 인스턴스에는 도달할 수 있습니다. 네트워크 주소 변환(NAT) 게이트웨이 또는 인스턴스를 사용하면 VPC의 인스턴스가 IPv4를 통해 인터넷으로 아웃바운드 연결을 시작하고 인터넷으로부터의 원치 않는 인바운드 연결은 차단하도록 할 수 있습니다. 할당 가능한 탄력적 IP 주소의 수가 제한되어 있으므로 고정 퍼블릭 IP 주소가 필요한 인스턴스가 많을 경우 NAT 디바이스를 사용하는 것이 좋습니다. 자세한 내용은 [NAT \(p. 172\)](#) 섹션을 참조하십시오. IPv6를 통해 인터넷에 대한 아웃바운드 전용 연결을 시작하기 위해 외부 전용 인터넷 게이트웨이를 사용할 수 있습니다. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

서브넷 3에 연결된 라우팅 테이블은 모든 IPv4 트래픽(0.0.0.0/0)을 가상 프라이빗 게이트웨이(예: vgw-1a2b3c4d)로 라우팅합니다. 인스턴스 3A는 VPN 연결을 통해 회사 네트워크의 컴퓨터에 접속할 수 있습니다.

## 서브넷 보안

AWS는 VPC의 보안을 강화하기 위해 사용할 수 있는 두 가지 기능, 보안 그룹과 네트워크 ACL을 제공합니다. 보안 그룹은 인스턴스용 인바운드 및 아웃바운드 트래픽을 제어하고, 네트워크 ACL은 서브넷용 인바운드 및 아웃바운드 트래픽을 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 맞출 수 있지만, 원하는 경우 네트워크 ACL을 사용하여 VPC에 보안 계층을 더 추가할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

각 서브넷에는 네트워크 ACL이 연결되어야 합니다. 생성된 각 서브넷에는 자동으로 VPC의 기본 네트워크 ACL이 연결됩니다. 연결된 네트워크 ACL 및 기본 네트워크 ACL의 내용을 변경할 수 있습니다. 자세한 내용은 [네트워크 ACL \(p. 102\)](#) 단원을 참조하십시오.

VPC 또는 서브넷에 흐름 로그를 만들어 VPC 또는 서브넷의 네트워크 인터페이스로 들어오고 나가는 모든 트래픽을 캡처할 수 있습니다. 또한 개별 네트워크 인터페이스에 흐름 로그를 만들 수도 있습니다. 흐름 로그는 CloudWatch Logs에 게시됩니다. 자세한 내용은 [VPC 흐름 로그 \(p. 144\)](#) 단원을 참조하십시오.

## 로컬 네트워크 및 기타 VPC와의 연결

필요할 경우 회사 또는 홈 네트워크에 VPC를 연결할 수 있습니다. 이때 네트워크의 접두사 중 하나가 VPC의 IPv4 주소 접두사와 겹칠 경우, 네트워크의 접두사에 대한 트래픽이 삭제됩니다. 예를 들어 다음과 같은 구성을 가정해 보겠습니다.

- CIDR 블록이 10.0.0.0/16인 VPC
- CIDR 블록이 10.0.1.0/24인 해당 VPC의 서브넷
- 서브넷에서 10.0.1.4 및 10.0.1.5 IP 주소로 실행 중인 인스턴스
- CIDR 블록이 10.0.37.0/24 및 10.1.38.0/24인 온프레미스 호스트 네트워크

이 구성에서 VPC의 인스턴스가 10.0.37.0/24 주소 공간의 호스트와 통신하려고 하면 해당 트래픽이 삭제됩니다. VPC에 지정된 더 큰 접두사(10.0.0.0/16)에 10.0.37.0/24가 포함되기 때문입니다. 10.1.38.0/24 공간의 호스트와는 통신이 가능합니다. 이 블록은 10.0.0.0/16에 포함되지 않기 때문입니다.

또한 사용자의 VPC 또는 다른 AWS 계정의 VPC와 VPC 피어링 연결을 만들 수 있습니다. VPC 피어링 연결을 사용하면 프라이빗 IP 주소를 사용하여 VPC 간 트래픽을 라우팅할 수 있습니다. 그러나 CIDR 블록이 겹치는 VPC 간에는 VPC 피어링 연결을 만들 수 없습니다. 자세한 내용은 [Amazon VPC Peering Guide](#) 항목을 참조하십시오.

따라서 앞으로 증가할 경우를 감안하여 충분히 넓은 CIDR 범위의 VPC를 만드는 것이 좋습니다. 단, 회사 또는 홈 네트워크의 현재나 향후의 서브넷과 현재나 향후의 VPC와 겹치지 않아야 합니다.

현재 IPv6를 통한 VPN 연결은 지원하지 않습니다.

## VPC 및 서브넷 관련 작업

Amazon VPC 콘솔을 사용하여 VPC 및 서브넷을 만들 수 있습니다. 다음은 VPC 및 서브넷을 수동으로 만들기 위한 절차입니다. 또한 게이트웨이와 라우팅 테이블도 수동으로 추가해야 합니다. 또는 Amazon VPC 마법사를 사용하여 VPC는 물론 서브넷, 게이트웨이 및 라우팅 테이블을 한 번에 만들 수 있습니다. 자세한 내용은 [시나리오 및 예시 \(p. 17\)](#) 단원을 참조하십시오.

### 항목

- [VPC 만들기 \(p. 66\)](#)
- [IPv6 CIDR 블록을 VPC와 연결 \(p. 67\)](#)
- [VPC에 서브넷 추가 \(p. 67\)](#)
- [IPv6 CIDR 블록을 서브넷에 연결 \(p. 68\)](#)
- [서브넷에서 인스턴스 시작 \(p. 68\)](#)
- [VPC 또는 서브넷에 연결된 IPv6 CIDR 블록을 분리 \(p. 68\)](#)
- [서브넷 삭제 \(p. 69\)](#)
- [VPC 삭제 \(p. 69\)](#)

### Note

(EC2-Classic) Amazon EC2 콘솔에서 시작 마법사를 사용하여 VPC에서만 사용할 수 있는 인스턴스 유형을 시작하려는데 기존 VPC가 없는 경우, 마법사에서 기본이 아닌 VPC와 서브넷을 생성합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [VPC에서만 사용할 수 있는 인스턴스 유형](#)을 참조하십시오.

## VPC 만들기

Amazon VPC 콘솔을 사용하여 빈 VPC를 만들 수 있습니다.

### VPC를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs와 Create VPC를 차례로 선택합니다.
3. 필요에 따라 다음 VPC 세부 정보를 지정한 후 Create VPC를 선택합니다.
  - Name tag: VPC의 이름을 입력할 수 있는 옵션. Name 키와 지정한 값으로 태그가 생성됩니다.
  - IPv4 CIDR block: VPC에 IPv4 CIDR 블록을 지정합니다. RFC 1918 규격에 따라 프라이빗(비공개적으로 라우팅 가능) IP 주소 범위에 속하는 CIDR 블록을 지정하는 것이 좋습니다(예: 10.0.0.0/16 또는 192.168.0.0/16). 공개적으로 라우팅 가능한 IPv4 주소의 범위를 지정할 수 있지만, 공개적으로 라우팅 가능한 VPC의 CIDR 블록에서 인터넷으로 직접 액세스하는 것은 현재 지원하지 않습니다. 범위가 224.0.0.0 ~ 255.255.255.255(Class D 및 Class E IP 주소 범위)인 VPC로 시작한 Windows 인스턴스는 올바르게 부팅되지 않습니다. IP 주소에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 주제를 참조하십시오.
  - IPv6 CIDR block: Amazon-provided IPv6 CIDR block을 선택하여 IPv6 CIDR 블록을 VPC에 연결할 수 있는 옵션.
  - Tenancy: 테넌시 옵션을 선택합니다. 예를 들어 전용 테넌시는 단일 테넌트 하드웨어에서 인스턴스를 실행할 수 있게 해줍니다. 전용 인스턴스에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [전용 인스턴스](#) 단원을 참조하십시오.



VPC를 생성한 후 서브넷을 추가할 수 있습니다. 자세한 내용은 [VPC에 서브넷 추가 \(p. 67\)](#) 단원을 참조하십시오.

## IPv6 CIDR 블록을 VPC와 연결

IPv6 CIDR 블록은 어떤 기존 VPC에도 연결할 수 있습니다. VPC에는 이와 연결된 기존 IPv6 CIDR 블록이 있어야 합니다.

IPv6 CIDR 블록을 VPC와 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택하고 Actions, Edit CIDRs를 선택합니다.
4. Add IPv6 CIDR을 선택합니다. IPv6 CIDR 블록을 추가한 후 Close를 선택합니다.

## VPC에 서브넷 추가

VPC에 새 서브넷을 추가할 때 서브넷이 속할 가용 영역을 지정할 수 있습니다. 동일한 가용 영역에 여러 서브넷을 둘 수 있습니다. VPC의 범위로부터 서브넷에 대해 IPv4 CIDR 블록을 지정해야 합니다. IPv6 CIDR 블록이 VPC와 연결되어 있는 경우 서브넷에 대해 IPv6 CIDR 블록을 지정할 수도 있습니다.

VPC에 서브넷을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets, Create Subnet을 선택합니다.
3. 필요에 따라 서브넷 세부 정보를 지정한 후 Create Subnet을 선택합니다.
  - Name tag: 서브넷의 이름을 입력할 수 있는 옵션. Name 키와 지정한 값으로 태그가 생성됩니다.
  - VPC: 서브넷을 만들고자 하는 VPC를 선택합니다.
  - Availability Zone: 서브넷이 상주하게 될 가용 영역을 선택하거나 기본 설정된 No Preference를 그대로 두어 AWS가 가용 영역을 자동 선택하도록 합니다.
  - IPv4 CIDR 블록: 서브넷에 IPv4 CIDR 블록을 지정합니다(예: 10.0.1.0/24). 자세한 내용은 [IPv4의 경우, VPC 및 서브넷 크기 조정 \(p. 63\)](#) 단원을 참조하십시오.
  - IPv6 CIDR 블록: (선택 사항) IPv6 CIDR 블록을 VPC에 연결한 경우, Specify a custom IPv6 CIDR을 선택합니다. 서브넷에 16진수 페어 값을 지정하거나 기본값을 그대로 둡니다.
4. (선택 사항) 필요한 경우 위 단계를 반복하여 VPC에 서브넷을 더 만듭니다.

서브넷을 만든 후 다음을 수행할 수 있습니다.

- 라우팅을 구성합니다. 서브넷을 퍼블릭 서브넷이 되게 하려면 먼저 VPC에 인터넷 게이트웨이를 연결해야 합니다. 자세한 내용은 [인터넷 게이트웨이 연결 \(p. 167\)](#) 단원을 참조하십시오. 그런 다음 사용자 지정 라우팅 테이블을 만들고, 인터넷 게이트웨이에 경로를 추가할 수 있습니다. 자세한 내용은 [사용자 지정 라우팅 테이블 생성 \(p. 167\)](#) 단원을 참조하십시오. 다른 라우팅 옵션의 경우, [라우팅 테이블 \(p. 154\)](#)을 참조하십시오.
- 서브넷 설정을 수정하여 해당 서브넷에서 시작하는 모든 인스턴스가 퍼블릭 IPv4 주소나 IPv6 주소, 또는 둘 다를 받도록 지정합니다. 자세한 내용은 [서브넷에 대한 IP 주소 지정 동작 \(p. 79\)](#) 단원을 참조하십시오.
- 필요에 따라 보안 그룹을 생성 또는 수정합니다. 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#) 단원을 참조하십시오.
- 필요에 따라 네트워크 ACL을 생성 또는 수정합니다. 네트워크 ACL에 대한 자세한 내용은 [네트워크 ACL \(p. 102\)](#)을 참조하십시오.

## IPv6 CIDR 블록을 서브넷에 연결

IPv6 CIDR 블록을 VPC의 기존 서브넷에 연결할 수 있습니다. 서브넷에는 이와 연결된 기존 IPv6 CIDR 블록이 있어서는 안 됩니다.

IPv6 CIDR 블록을 서브넷에 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷을 선택하고 Subnet Actions, Edit IPv6 CIDRs를 선택합니다.
4. Add IPv6 CIDR을 선택합니다. 서브넷에 16진수 페어(예: 00)를 지정하고 체크 표시 아이콘을 선택하여 항목을 확정합니다.
5. [Close]를 선택합니다.

## 서브넷에서 인스턴스 시작

서브넷을 만들고 라우팅을 구성한 후, Amazon EC2 콘솔을 사용하여 해당 서브넷에서 인스턴스를 시작할 수 있습니다.

서브넷에서 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 Launch Instance를 선택합니다.
3. 마법사의 지침대로 진행합니다. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.

### Note

인스턴스가 IPv6를 통해 통신하도록 하고 싶은 경우, 지원되는 인스턴스 유형을 선택해야 합니다. M3 및 G2를 제외한 모든 현 세대 인스턴스 유형은 IPv6 주소를 지원합니다.

4. Configure Instance Details 페이지의 Network 목록에서 필요한 VPC를 선택했는지 확인한 다음, 인스턴스를 실행할 서브넷을 선택합니다. 이 페이지의 다른 설정은 기본값으로 두고 Next: Add Storage를 선택합니다.
5. 마법사의 다음 페이지에서 인스턴스의 스토리지를 구성하고 태그를 추가할 수 있습니다. Configure Security Group 페이지에서 기존 보안 그룹 중 하나를 선택하거나 마법사의 안내에 따라 새 보안 그룹을 생성합니다. 작업을 마치면 Review and Launch를 선택합니다.
6. 설정을 검토한 후 Launch를 선택합니다.
7. 소유한 기존 키 페어를 선택하거나 새 키 페어를 생성한 다음, 작업이 끝나면 Launch Instances를 선택합니다.

## VPC 또는 서브넷에 연결된 IPv6 CIDR 블록을 분리

VPC 또는 서브넷에서 IPv6 지원이 더 이상 필요 없지만 IPv4 리소스 생성 및 IPv4 리소스와의 통신을 위해 VPC 또는 서브넷을 계속해서 사용하기를 원하는 경우, 연결된 IPv6 CIDR 블록을 분리할 수 있습니다.

연결된 IPv6 CIDR 블록을 분리하려면 먼저 서브넷의 모든 인스턴스에 할당된 모든 IPv6 주소를 분리해야 합니다. 자세한 내용은 [인스턴스에 할당된 IPv6 주소 해제 \(p. 82\)](#) 단원을 참조하십시오.

서브넷에 연결된 IPv6 CIDR 블록을 분리하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷을 선택하고 Subnet Actions, Edit IPv6 CIDRs를 선택합니다.
4. 십자가 아이콘을 선택하여 서브넷에 대한 IPv6 CIDR 블록을 제거합니다.
5. Close를 선택합니다.

VPC에 연결된 IPv6 CIDR 블록을 분리하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택하고 Actions, Edit CIDRs를 선택합니다.
4. 십자가 아이콘을 선택하여 IPv6 CIDR 블록을 제거합니다.
5. Close를 선택합니다.

#### Note

IPv6 CIDR 블록을 분리해도 IPv6 네트워킹을 위해 구성된 보안 그룹 규칙, 네트워크 ACL 규칙 또는 라우팅 테이블 경로는 자동으로 삭제되지 않습니다. 이 규칙 또는 경로는 수동으로 수정하거나 삭제해야 합니다.

## 서브넷 삭제

서브넷이 더 이상 필요하지 않으면 삭제할 수 있습니다. 먼저 서브넷의 모든 인스턴스를 종료해야 합니다.

서브넷을 삭제하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 서브넷의 모든 인스턴스를 종료합니다. 자세한 내용은 EC2 사용 설명서의 [인스턴스 종료](#) 단원을 참조하십시오.
3. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
4. 탐색 창에서 Subnets을 선택합니다.
5. 삭제할 서브넷을 선택한 다음, Subnet Actions, Delete를 선택합니다.
6. Delete Subnet 대화 상자에서 Yes, Delete를 선택합니다.

## VPC 삭제

VPC는 언제든지 삭제할 수 있습니다. 그러려면 먼저 VPC에서 모든 인스턴스를 종료해야 합니다. Amazon VPC 콘솔을 사용하여 VPC를 삭제할 경우, 서브넷, 보안 그룹, 네트워크 ACL, 인터넷 게이트웨이, VPC 피어링 연결, DHCP 옵션 등의 구성 요소도 모두 삭제됩니다.

VPN 연결이 있는 경우 VPN 연결 또는 VPN에 관련된 다른 구성 요소(예: 고객 게이트웨이, 가상 프라이빗 게이트웨이)를 삭제할 필요는 없습니다. 다른 VPC에서 이 고객 게이트웨이를 사용할 계획이라면 VPN 연결 및 게이트웨이를 유지하는 것이 좋습니다. 삭제하면 새 VPN 연결을 만든 후 네트워크 관리자가 고객 게이트웨이를 다시 구성해야 합니다.

VPC를 삭제하려면 다음을 수행합니다.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. VPC에서 모든 인스턴스를 종료합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [인스턴스 종료](#)를 참조하십시오.
3. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

4. 탐색 창에서 Your VPCs를 선택합니다.
5. 삭제할 VPC를 선택하고 Actions, Delete VPC를 선택합니다.
6. VPN 연결을 삭제하려면 해당 옵션을 선택합니다. 삭제하지 않으려면 선택하지 않은 채로 둡니다. Yes, Delete를 선택합니다.

## CLI 개요

명령줄 인터페이스를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 사용 가능한 API 작업 목록을 비롯한 자세한 정보는 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

### VPC 만들기

- [create-vpc](#)(AWS CLI)
- [New-EC2Vpc](#)(Windows PowerShell용 AWS 도구)

### 서브넷 생성

- [create-subnet](#)(AWS CLI)
- [New-EC2Subnet](#)(Windows PowerShell용 AWS 도구)

### IPv6 CIDR 블록을 VPC에 연결

- [associate-vpc-cidr-block](#)(AWS CLI)
- [Register-EC2VpcCidrBlock](#)(Windows PowerShell용 AWS 도구)

### IPv6 CIDR 블록을 서브넷에 연결

- [associate-subnet-cidr-block](#)(AWS CLI)
- [Register-EC2SubnetCidrBlock](#)(Windows PowerShell용 AWS 도구)

### VPC에 연결된 IPv6 CIDR 블록을 분리

- [disassociate-vpc-cidr-block](#)(AWS CLI)
- [Unregister-EC2VpcCidrBlock](#)(Windows PowerShell용 AWS 도구)

### 서브넷에 연결된 IPv6 CIDR 블록을 분리

- [disassociate-subnet-cidr-block](#)(AWS CLI)
- [Unregister-EC2SubnetCidrBlock](#)(Windows PowerShell용 AWS 도구)

### VPC 설명

- [describe-vpcs](#)(AWS CLI)
- [Get-EC2Vpc](#)(Windows PowerShell용 AWS 도구)

### 서브넷 설명

- [describe-subnets](#)(AWS CLI)
- [Get-EC2Subnet](#)(Windows PowerShell용 AWS 도구)

#### VPC 삭제

- [delete-vpc](#)(AWS CLI)
- [Remove-EC2Vpc](#)(Windows PowerShell용 AWS 도구)

#### 서브넷 삭제

- [delete-subnet](#)(AWS CLI)
- [Remove-EC2Subnet](#)(Windows PowerShell용 AWS 도구)

# 기본 VPC 및 기본 서브넷

2013년 12월 4일 이후에 AWS 계정을 만든 경우에는 EC2-VPC만 지원됩니다. 이 경우 각 AWS 리전에 기본 VPC를 갖게 됩니다. 기본 VPC는 바로 사용할 수 있으므로 별도의 구성 단계 없이 즉시 인스턴스를 기본 VPC에서 시작할 수 있습니다. 기본 VPC는 EC2-VPC 플랫폼이 제공하는 고급 네트워킹 기능의 이점과 EC2-Classic 플랫폼의 편리함을 모두 갖추고 있습니다.

EC2-Classic 및 EC2-VPC 플랫폼에 대한 자세한 내용은 [지원되는 플랫폼](#)을 참조하십시오.

## 항목

- [기본 VPC의 기본 사항](#) (p. 72)
- [지원되는 플랫폼 및 기본 VPC 보유 여부 확인](#) (p. 74)
- [기본 VPC로 EC2 인스턴스 시작](#) (p. 74)
- [기본 서브넷과 기본 VPC 삭제](#) (p. 75)

## 기본 VPC의 기본 사항

이 단원에서는 기본 Virtual Private Cloud(VPC)와 그 기본 서브넷에 대해 설명합니다.

### 가용성

2013년 12월 4일 이후에 AWS 계정을 만든 경우에는 EC2-VPC만 지원됩니다. 이 경우 각 AWS 리전에 기본 VPC가 생성됩니다. 따라서 기본이 아닌 VPC를 만들어 생성하여 인스턴스 시작 시 지정한 경우가 아니면 인스턴스가 기본 VPC에서 시작됩니다.

AWS 계정을 2013년 3월 18일 이전에 만든 경우, 이전에 사용하던 리전에서는 EC2-Classic과 EC2-VPC가 모두 지원되며 이전에 없었던 리전에서는 EC2-VPC만 지원됩니다. 이 경우 AWS 리소스를 생성하지 않은 각 리전에는 기본 VPC가 생성됩니다. 따라서 기본이 아닌 VPC를 생성하여 이전에 없었던 리전에서 인스턴스를 시작할 때 지정한 경우가 아니면 해당 리전에서 인스턴스가 기본 VPC에서 시작됩니다. 하지만 이전에 사용했던 리전에서 인스턴스를 시작하면 인스턴스가 EC2-Classic에서 시작됩니다.

AWS 계정을 2013년 3월 18일부터 2013년 12월 4일 사이에 만든 경우에는 EC2-VPC만 지원될 수 있으며, 이전에 사용했던 일부 리전에서는 EC2-Classic과 EC2-VPC를 모두 지원될 수 있습니다. AWS 계정에 대해 각 리전에서 지원되는 플랫폼을 확인하는 방법은 [지원되는 플랫폼 및 기본 VPC 보유 여부 확인](#) (p. 74) 단원을 참조하십시오. 각 리전이 기본 VPC를 사용하도록 설정된 시점에 대한 자세한 내용은 Amazon VPC에 대한 AWS 포럼에서 [Announcement: Enabling regions for the default VPC feature set](#)를 참조하십시오.

AWS 계정이 EC2-VPC만 지원할 경우, 이 AWS 계정과 연결된 IAM 계정 역시 EC2-VPC만 지원하며 AWS 계정과 동일한 기본 VPC를 사용합니다.

AWS 계정이 EC2-Classic과 EC2-VPC를 모두 지원할 경우 인스턴스를 EC2-Classic에서 시작하는 간편함과 함께 EC2-VPC의 고급 기능을 사용하려면 AWS 계정을 새로 만들거나, 인스턴스를 이전에 사용하지 않았던 리전에서 시작하면 됩니다. 이전에 사용하지 않았던 리전에 기본 VPC를 추가하려면 [기본 VPC FAQ](#)에서 "기존 EC2 계정에 기본 VPC를 보유하고 싶습니다. 가능합니까?" 를 참조하십시오.

## 구성 요소

기본 VPC는 다음과 같이 생성됩니다.

- IPv4 CIDR 블록의 크기가 /16인 VPC를 만듭니다.
- 각 가용 영역에서 기본 서브넷을 생성합니다.
- 인터넷 게이트웨이를 생성하여 기본 VPC와 연결합니다.
- 인터넷으로 향하는 모든 IPv4 트래픽을 인터넷 게이트웨이로 전송하는 규칙을 사용하여 기본 VPC에 대한 기본 라우팅 테이블을 생성합니다.
- 기본 보안 그룹을 만들어 기본 VPC와 연결합니다.
- 네트워크 ACL(액세스 제어 목록)을 생성하여 기본 VPC와 연결합니다.
- AWS 계정에서 설정된 기본 DHCP 옵션을 기본 VPC와 연결합니다.

다음 그림은 기본 VPC에 대해 설정되는 핵심 구성 요소를 보여 줍니다.

기본 서브넷에서 시작한 인스턴스는 퍼블릭 IPv4 주소와 프라이빗 IPv4 주소를 모두 수신합니다. 또한 기본 서브넷의 인스턴스는 퍼블릭 DNS 호스트 이름과 프라이빗 DNS 호스트 이름을 모두 수신합니다. 기본 VPC의 기본이 아닌 서브넷에서 시작한 인스턴스는 퍼블릭 IPv4 주소나 DNS 호스트 이름을 수신하지 않습니다. 서브넷의 퍼블릭 IP 주소 지정 동작은 변경할 수 있습니다. 자세한 내용은 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 80\)](#) 단원을 참조하십시오.

기본 VPC는 다른 VPC를 사용하듯이 사용할 수 있습니다. 즉, 서브넷을 추가하고, 기본 라우팅 테이블을 수정하고, 다른 라우팅 테이블을 추가하며, 추가 보안 그룹을 연결하고, 기본 보안 그룹의 규칙을 업데이트하고, VPN 연결을 추가할 수 있습니다. VPC를 추가로 만들 수도 있습니다.

기본 서브넷도 다른 서브넷을 사용하듯이 사용할 수 있습니다. 즉, 사용자 지정 라우팅 테이블을 추가하고 네트워크 ACL을 설정할 수 있습니다. EC2 인스턴스를 시작할 때 기본 서브넷을 지정할 수도 있습니다.

IPv6 CIDR 블록을 기본 VPC에 연결할 수도 있습니다. 자세한 내용은 [VPC 및 서브넷 관련 작업 \(p. 66\)](#)을 참조하십시오.

## 기본 서브넷

기본 VPC의 CIDR 블록은 항상 /16 넷마스크(172.31.0.0/16)입니다. 이는 최대 65,536개의 프라이빗 IPv4 주소를 제공합니다. 기본 서브넷의 넷마스크는 항상 /20으로, 서브넷당 최대 4,096개의 주소를 제공하며 그중 몇 개는 예약되어 있습니다.

기본 라우팅 테이블은 인터넷으로 목적지가 정해진 서브넷의 트래픽을 인터넷 게이트웨이로 전송하기 때문에 기본적으로 기본 서브넷은 퍼블릭 서브넷입니다. 대상 주소 0.0.0.0/0에서 인터넷 게이트웨이로 라우팅을 제거함으로써 기본 서브넷을 프라이빗 서브넷으로 만들 수 있습니다. 하지만 이렇게 하면 해당 서브넷에서 실행하는 EC2 인스턴스는 인터넷에 액세스할 수 없습니다.

때때로 AWS는 새로운 가용 영역을 리전에 추가할 수 있습니다. 대부분의 경우 기본 VPC에 대한 이 가용 영역에 새로운 기본 서브넷이 자동으로 만들어집니다. 하지만 기본 VPC를 수정했을 경우에는 새로운 기본 서브넷이 추가되지 않습니다. 새로운 가용 영역에 대한 기본 서브넷이 필요할 경우 AWS Support에 문의하여 기본 서브넷을 요청하십시오.

## 지원되는 플랫폼 및 기본 VPC 보유 여부 확인

Amazon VPC에 대해 알지 못해도 EC2 인스턴스를 기본 VPC로 실행하여 Elastic Load Balancing, Amazon Relational Database Service(Amazon RDS), Amazon EMR(Amazon EMR) 등과 같은 서비스를 사용할 수 있습니다. 이러한 서비스는 기본 VPC를 사용하든 EC2-Classic을 사용하든 상관없이 동일하게 사용할 수 있습니다. 하지만 Amazon EC2 콘솔이나 명령줄을 사용하여 AWS 계정이 모든 플랫폼을 지원하는지 그리고 기본 VPC가 있는지 여부를 확인할 수 있습니다.

### 콘솔을 사용하여 플랫폼 지원 확인

Amazon EC2 콘솔은 EC2 인스턴스를 시작할 수 있는 플랫폼 및 기본 VPC가 있는지 여부를 보여 줍니다.

탐색 모음에서 사용할 리전이 선택되어 있는지 확인합니다. Amazon EC2 콘솔 대시보드의 [Account Attributes]에서 [Supported Platforms]를 찾습니다. 두 개의 값(EC2와 VPC)이 있을 경우 두 플랫폼 중 하나에서 인스턴스를 시작할 수 있습니다. 하나의 값(VPC)만 있을 경우 EC2-VPC에서만 인스턴스를 시작할 수 있습니다.

예를 들어 다음은 해당 계정이 EC2-VPC 플랫폼만을 지원하며, 식별자가 vpc-1a2b3c4d인 기본 VPC가 있음을 나타냅니다.

기본 VPC를 삭제하면 [Default VPC] 값이 [None]으로 표시됩니다. 자세한 내용은 [기본 서브넷과 기본 VPC 삭제 \(p. 75\)](#) 단원을 참조하십시오.

### 명령줄을 사용하여 플랫폼 지원 확인

supported-platforms 속성은 EC2 인스턴스를 시작할 수 있는 플랫폼을 나타냅니다. 해당 계정의 이 속성 값을 보려면 다음 명령 중 하나를 사용합니다.

- `describe-account-attributes`(AWS CLI)
- `Get-EC2AccountAttributes`(Windows PowerShell용 AWS 도구)

또한 다음 명령을 사용하여 VPC를 조회하면 출력에 기본 VPC도 표시됩니다.

- `describe-vpcs`(AWS CLI)
- `Get-EC2Vpc`(Windows PowerShell용 AWS 도구)

## 기본 VPC로 EC2 인스턴스 시작

서브넷을 지정하지 않고 EC2 인스턴스를 시작하면 자동으로 기본 VPC의 기본 서브넷에서 시작됩니다. 가용 영역은 기본적으로 선택되고 이 가용 영역의 해당 서브넷으로 인스턴스가 시작됩니다. 또는 콘솔에서 기본 서브넷을 선택하거나, CLI에서 서브넷 또는 가용 영역을 지정하여 인스턴스에 대한 가용 영역을 직접 선택할 수도 있습니다.

### 콘솔을 사용하여 EC2 인스턴스 시작

기본 VPC로 EC2 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. EC2 대시보드에서 Launch Instance를 선택합니다.
3. 마법사의 지침대로 진행합니다. AMI를 선택하고 인스턴스 유형을 선택합니다. Review and Launch를 선택하여 마법사의 나머지 부분에 대한 기본 설정을 수락할 수 있습니다. 그러면 Review Instance Launch 페이지로 바로 이동합니다.



4. 설정을 검토합니다. Instance Details 섹션에서 Subnet의 기본값은 No preference (default subnet in any Availability Zone)입니다. 이는 인스턴스가 자동으로 선택된 가용 영역의 기본 서브넷에서 시작됨을 나타냅니다. 아니면 Edit instance details를 선택하여 특정 가용 영역에 대한 기본 서브넷을 선택할 수도 있습니다.
5. Launch를 선택하여 키 페어를 선택하고 해당 인스턴스를 시작합니다.

## 명령줄을 사용하여 EC2 인스턴스 시작

다음 명령 중 하나를 사용하여 EC2 인스턴스를 시작할 수 있습니다.

- [run-instances](#)(AWS CLI)
- [New-EC2Instance](#)(Windows PowerShell용 AWS 도구)

EC2 인스턴스를 기본 VPC에서 시작하려면 서브넷이나 가용 영역을 지정하지 않고 다음 명령을 사용합니다.

EC2 인스턴스를 기본 VPC의 특정 기본 서브넷에서 시작하려면 해당 서브넷 ID나 가용 영역을 지정합니다.

## 기본 서브넷과 기본 VPC 삭제

기본 서브넷이나 기본 VPC는 다른 서브넷 또는 VPC처럼 삭제할 수 있습니다. 그러나 기본 서브넷이나 기본 VPC를 삭제하면 EC2-Classic으로 인스턴스를 시작할 수 없게 되기 때문에 다른 VPC에서 인스턴스를 시작할 서브넷을 명시적으로 지정해야 합니다. 다른 VPC가 없으면 기본이 아닌 VPC와 기본이 아닌 서브넷을 만들어야 합니다. 자세한 내용은 [VPC 만들기 \(p. 66\)](#) 단원을 참조하십시오.

기본 서브넷을 삭제하면 해당 가용 영역에 기본이 아닌 서브넷을 만들지 않는 한 인스턴스를 기본 VPC의 해당 가용 영역에서 시작할 수 없습니다. 삭제한 기본 서브넷을 복원하려면, 기본이 아닌 서브넷을 만들고 AWS Support에 연락하여 해당 서브넷을 기본 서브넷으로 표시하도록 요청할 수 있습니다. 이때 AWS 계정 ID, 리전, 서브넷 ID를 알려 주어야 합니다. 새로운 기본 서브넷이 정상적으로 동작하는지 확인하려면 서브넷 속성을 수정하여 해당 서브넷에서 시작되는 인스턴스에 퍼블릭 IP 주소를 할당하십시오. 자세한 내용은 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 80\)](#) 단원을 참조하십시오. 가용 영역당 기본 서브넷은 한 개만 가질 수 있습니다. 기본 VPC가 아닌 VPC에는 기본 서브넷을 생성할 수 없습니다.

기본 VPC를 삭제한 후 새 VPC가 필요하다면, AWS Support에 연락하여 해당 리전에 기본 VPC를 새로 생성해 줄 것을 요청할 수 있습니다. 기존 VPC를 기본 VPC로 만들 수는 없습니다.

Amazon VPC 콘솔에서 기본 서브넷이나 기본 VPC를 삭제하려고 하면 대화 상자에 경고가 표시되고, 기본 서브넷이나 기본 VPC를 삭제하려 하고 있음을 알려 줍니다.

# VPC의 IP 주소 지정

IP 주소는 VPC 내의 리소스가 서로 통신하도록, 그리고 인터넷을 통해 다른 리소스와 통신할 수 있게 해줍니다. Amazon EC2와 Amazon VPC는 IPv4 및 IPv6 주소 지정 프로토콜을 지원합니다.

Amazon EC2와 Amazon VPC는 IPv4 주소 지정 프로토콜을 사용하도록 기본 설정되어 있습니다. VPC를 생성할 때 VPC에 IPv4 CIDR 블록(프라이빗 IPv4 주소)을 할당해야 합니다. 프라이빗 IPv4 주소는 인터넷으로 접속할 수 없습니다. 인터넷을 통해 인스턴스에 접속하거나 인스턴스 간의 통신과 퍼블릭 엔드포인트가 있는 다른 AWS 서비스 간의 통신을 가능케 하려면 인스턴스에 전역적으로 고유한 퍼블릭 IPv4 주소를 할당해야 합니다.

IPv6 CIDR 블록을 VPC와 서브넷에 연결하고 그 블록에 속한 IPv6 주소를 VPC의 리소스에 할당할 수도 있습니다. IPv6 주소는 퍼블릭이며 인터넷으로는 접속할 수 없습니다.

## Note

인스턴스가 인터넷과 통신하려면 VPC에 인터넷 게이트웨이드 연결해야 합니다. 자세한 내용은 [인터넷 게이트웨이 \(p. 165\)](#) 단원을 참조하십시오.

VPC는 듀얼 스택 모드로 작동할 수 있으므로, 리소스는 IPv4나 IPv6, 또는 둘 다를 통해 통신할 수 있습니다. IPv4 및 IPv6 주소는 서로 독립적입니다. 따라서 IPv4 및 IPv6에 대해 별도로 VPC의 라우팅 및 보안을 구성해야 합니다.

다음 표에는 Amazon EC2와 Amazon VPC의 IPv4 및 IPv6 간 차이점이 요약되어 있습니다.

## IPv4 및 IPv6의 특징과 제한

IPv4	IPv6
형식은 32비트의 네 자리 숫자로 이루어진 그룹 4개입니다.	형식은 128비트의 네 자리 16진수로 이루어진 그룹 8개입니다.
기본값이며 모든 VPC에 필요, 제거할 수 없음.	옵트인 전용.
VPC CIDR 블록 크기는 /16에서 /28까지 가능합니다.	VPC CIDR 블록의 크기는 /56으로 고정되어 있습니다.

IPv4	IPv6
서브넷 CIDR 블록의 크기는 /16에서 /28까지 가능합니다.	서브넷 CIDR 블록의 크기는 /64로 고정되어 있습니다.
VPC에 대해 프라이빗 IPv4 CIDR 블록을 선택할 수 있습니다.	Amazon은 당사의 IPv6 주소 풀에서 VPC에 대한 IPv6 CIDR 블록을 선택합니다. 자신만의 고유한 범위를 선택할 수 없습니다.
프라이빗 및 퍼블릭 IP 주소는 서로 구분됩니다. 인터넷과 통신하기 위해 퍼블릭 IPv4 주소는 네트워크 주소 변환(NAT)을 통해 기본 프라이빗 IPv4 주소로 매핑됩니다.	퍼블릭 및 프라이빗 IP 주소는 서로 구분되지 않습니다. IPv6 주소는 퍼블릭입니다.
모든 인스턴스 유형에서 지원.	M3 및 G2를 제외한 모든 현 세대 인스턴스 유형에서 지원됨. 자세한 내용은 <a href="#">Amazon EC2 인스턴스 유형</a> 단원을 참조하십시오.
EC2-Classic, 그리고 ClassicLink를 통한 VPC와의 EC2-Classic 연결에서 지원됨.	EC2-Classic에서 지원되지 않고, ClassicLink를 통한 VPC와의 EC2-Classic 연결에 대해 지원되지 않음.
모든 AMI에서 지원.	DHCPv6에 맞게 구성된 AMI에서 자동으로 지원됨. Amazon Linux 2016.09.0 이상 버전과 Windows Server 2008 R2 이상 버전은 DHCPv6에 맞게 구성됩니다. 다른 AMI의 경우, <a href="#">인스턴스를 수동으로 구성 (p. 88)</a> 하여 할당된 IPv6 주소를 모두 인식해야 합니다.
인스턴스는 프라이빗 IPv4 주소에 상응하는 Amazon 제공 프라이빗 DNS 호스트 이름을 받고, 해당되는 경우, 퍼블릭 IPv4 또는 탄력적 IP 주소에 상응하는 퍼블릭 DNS 호스트 이름을 받습니다.	Amazon 제공 DNS 호스트 이름은 지원하지 않습니다.
탄력적 IPv4 주소는 지원하지 않습니다.	탄력적 IPv6 주소는 지원하지 않습니다.
VPC VPN 연결 및 고객 게이트웨이, NAT 디바이스 및 VPC 엔드포인트 지원.	VPC VPN 연결 및 고객 게이트웨이, NAT 디바이스 및 VPC 엔드포인트 지원하지 않음.

가상 프라이빗 게이트웨이를 통해 AWS Direct Connect 연결로 가는 IPv6 트래픽은 지원합니다. 자세한 내용은 [AWS Direct Connect 사용 설명서](#)를 참조하십시오.

#### 항목

- [프라이빗 IPv4 주소 \(p. 77\)](#)
- [퍼블릭 IPv4 주소 \(p. 78\)](#)
- [IPv6 주소 \(p. 78\)](#)
- [서브넷에 대한 IP 주소 지정 동작 \(p. 79\)](#)
- [IP 주소 작업 \(p. 79\)](#)
- [IPv6로 마이그레이션하기 \(p. 83\)](#)

## 프라이빗 IPv4 주소

프라이빗 IPv4 주소(이 단원에서는 프라이빗 IP 주소로도 표시)는 인터넷을 통해 액세스할 수 없고, VPC의 인스턴스 간 통신에 사용할 수 있습니다. VPC에서 인스턴스를 시작할 경우, 서브넷의 IPv4 주소 범위에 속한

주 프라이빗 IP 주소는 인스턴스의 주 네트워크 인터페이스(eth0)에 할당됩니다. 또한 각 인스턴스에는 인스턴스의 프라이빗 IP 주소를 확인하는 프라이빗(내부) DNS 호스트 이름이 할당됩니다. 주 프라이빗 IP 주소를 지정하지 않으면 서브넷 범위에서 사용 가능한 IP 주소가 선택됩니다. 네트워크 인터페이스에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [ENI\(Elastic Network Interfaces\)](#)를 참조하십시오.

보조 프라이빗 IP 주소인 추가 프라이빗 IP 주소를 VPC에서 실행 중인 인스턴스에 할당할 수 있습니다. 주 프라이빗 IP 주소와 달리, 보조 프라이빗 IP 주소는 한 네트워크 인스턴스에서 다른 네트워크 인스턴스로 재할당할 수 있습니다. 인스턴스가 중지 및 재시작될 때 프라이빗 IP 주소는 네트워크 인터페이스와 계속해서 연동되고 인스턴스가 종료되면 연동이 해제됩니다. 주 IP 주소와 보조 IP 주소에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [여러 IP 주소](#)를 참조하십시오.

#### Note

프라이빗 IP 주소란 VPC의 IPv4 CIDR 범위 내 IP 주소를 말합니다. 대부분의 VPC IP 주소 범위는 RFC 1918에서 지정된 프라이빗(비공개적으로 라우팅 가능) IP 주소 범위 내에 들어가지만 VPC에 대해 공개적으로 라우팅이 가능한 CIDR 블록을 사용할 수 있습니다. VPC의 IP 주소 범위에 상관없이 공개적으로 라우팅 가능한 CIDR 블록을 포함해 VPC의 CIDR 블록에서 인터넷으로 직접 액세스하는 것은 지원하지 않습니다. 인터넷 게이트웨이, 가상 프라이빗 게이트웨이, VPN 연결 또는 AWS Direct Connect와 같은 게이트웨이를 통한 인터넷 액세스를 설정해야 합니다.

## 퍼블릭 IPv4 주소

모든 서브넷은 해당 서브넷에서 생성된 네트워크 인터페이스가 퍼블릭 IPv4 주소(이 단원에서는 퍼블릭 IP 주소로도 표시함)를 받을 것인지 여부를 결정하는 속성을 갖습니다. 따라서 이 속성이 활성화된 서브넷에서 인스턴스를 시작할 경우, 퍼블릭 IP 주소는 인스턴스에 대해 생성된 주 네트워크 인터페이스(eth0)에 할당됩니다. 퍼블릭 IP 주소는 NAT(Network Address Translation)를 통해 주 프라이빗 IP 주소로 매핑됩니다.

다음을 수행하여 인스턴스가 퍼블릭 IP 주소를 수신할지 여부를 제어할 수 있습니다.

- 서브넷의 퍼블릭 IP 주소 지정 속성 수정. 자세한 내용은 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 80\)](#) 단원을 참조하십시오.
- 인스턴스를 시작하는 동안 퍼블릭 IP 주소 지정 기능을 활성화하거나 비활성화하면 서브넷의 퍼블릭 IP 주소 지정 속성을 재정의합니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 80\)](#) 단원을 참조하십시오.

퍼블릭 IP 주소는 Amazon의 퍼블릭 IP 주소 풀로부터 할당되며 계정과는 관련이 없습니다. 인스턴스에서 퍼블릭 IP 주소의 연결이 해제되면 이 주소는 풀로 돌아가지만 더 이상 사용할 수 없습니다. 퍼블릭 IP 주소는 수동으로 연결하거나 해제할 수 없습니다. 어떤 경우에는 Amazon에서 귀하의 인스턴스로부터 퍼블릭 IP 주소를 해제하거나 새 인스턴스에 할당합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [퍼블릭 IP 주소](#)를 참조하십시오.

필요에 따라 인스턴스에 할당하거나 인스턴스에서 제거가 가능한, 계정에 할당된 영구 퍼블릭 IP 주소가 필요한 경우, 그 대신에 탄력적 IP 주소를 사용하십시오. 자세한 내용은 [탄력적 IP 주소 \(p. 197\)](#) 단원을 참조하십시오.

VPC에서 DNS 호스트 이름을 지원하는 경우, 퍼블릭 IP 주소 또는 탄력적 IP 주소를 받는 각 인스턴스에도 퍼블릭 DNS 호스트 이름이 할당됩니다. Amazon은 퍼블릭 DNS 호스트 이름을 인스턴스 네트워크 외부에서는 인스턴스의 퍼블릭 IP 주소로 변환하고 인스턴스 네트워크 내부에서는 인스턴스의 프라이빗 IP 주소로 변환합니다. 자세한 내용은 [VPC와 함께 DNS 사용 \(p. 194\)](#) 단원을 참조하십시오.

## IPv6 주소

IPv6 CIDR 블록을 VPC 및 서브넷에 연결할 수도 있습니다. 자세한 내용은 다음 주제를 참조하십시오.

- [IPv6 CIDR 블록을 VPC와 연결 \(p. 67\)](#)
- [IPv6 CIDR 블록을 서브넷에 연결 \(p. 68\)](#)

IPv6 CIDR 블록이 VPC와 서브넷에 연결되어 있고 다음 중 하나가 true이면 VPC의 인스턴스는 IPv6 주소를 받습니다.

- 서브넷은 인스턴스가 시작되는 과정에서 인스턴스의 주 네트워크 인터페이스에 IPv6 주소를 자동 할당하도록 구성되어 있습니다.
- 시작하는 과정에서 인스턴스에 IPv6 주소를 수동으로 할당합니다.
- 시작을 완료한 후에 인스턴스에 IPv6 주소를 할당합니다.
- 동일 서브넷에서 네트워크 인터페이스에 IPv6 주소를 할당하고 시작을 완료한 후에 인스턴스에 네트워크 인터페이스를 연결합니다.

시작하는 과정에서 인스턴스가 IPv6 주소를 받는 경우, 해당 주소는 인스턴스의 주 네트워크 인터페이스 (eth0)와 연결됩니다. 주 네트워크 인터페이스에서 IPv6 주소 연결을 해제할 수 있습니다. 인스턴스에 대해서는 IPv6 DNS 호스트 이름을 지원하지 않습니다.

인스턴스를 중지하고 시작할 때에는 IPv6 주소가 지속되다가 인스턴스를 종료하면 해제됩니다. IPv6 주소는 다른 네트워크 인터페이스—에 할당되는 동안에는 재할당할 수 없으므로, 먼저 할당을 해제해야 합니다.

인스턴스에 연결된 네트워크 인터페이스에 IPv6 주소를 할당함으로써 인스턴스에 추가 IPv6 주소를 할당할 수 있습니다. 네트워크 인터페이스에 할당할 수 있는 IPv6 주소의 개수, 그리고 인스턴스에 연결할 수 있는 네트워크 인터페이스의 개수는 인스턴스 유형에 따라 달라집니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 유형별/네트워크 인터페이스당 프라이빗 IP 주소](#) 단원을 참조하십시오.

IPv6 주소는 전역적으로 고유하므로 인터넷으로 접속할 수 있습니다. 서브넷에 대한 라우팅을 제어하거나 보안 그룹 및 네트워크 ACL 규칙을 사용함으로써 인스턴스가 IPv6 주소를 통해 접속이 가능하도록 할지 여부를 제어할 수 있습니다. 자세한 내용은 [보안 \(p. 94\)](#) 단원을 참조하십시오.

예약된 IPv6 주소 범위에 대한 자세한 내용은 [IANA IPv6 Special-Purpose Address Registry](#) 및 [RFC4291](#)을 참조하십시오.

## 서브넷에 대한 IP 주소 지정 동작

모든 서브넷은 해당 서브넷에서 생성된 네트워크 인터페이스에 퍼블릭 IPv4 주소(해당되는 경우, IPv6 주소)가 할당될 것인지 결정하는 수정 가능한 속성을 갖습니다. 여기에는 해당 서브넷에서 인스턴스를 시작할 때 인스턴스에 대해 생성된 주 네트워크 인터페이스(eth0)가 포함됩니다.

서브넷 속성에 상관없이 특정 인스턴스를 시작하는 중에 해당 인스턴스에 대한 이 설정을 재정의할 수 있습니다. 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 80\)](#) 및 [인스턴스 시작 시 IPv6 주소 배정 \(p. 81\)](#) 섹션을 참조하십시오.

## IP 주소 작업

서브넷의 IP 주소 지정 동작을 수정하고, 시작 중에 퍼블릭 IPv4 주소를 인스턴스에 할당하며, 인스턴스에 대해 IPv6 주소를 할당하거나 IPv6 주소 할당을 해제할 수 있습니다.

항목

- [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 80\)](#)
- [서브넷의 퍼블릭 IPv6 주소 지정 속성 수정 \(p. 80\)](#)

- 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 80)
- 인스턴스 시작 시 IPv6 주소 배정 (p. 81)
- 인스턴스에 IPv6 주소 할당 (p. 82)
- 인스턴스에 할당된 IPv6 주소 해제 (p. 82)
- API 및 명령 개요 (p. 83)

## 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정

기본이 아닌 서브넷은 IPv4 퍼블릭 주소 지정 속성이 `false`로 기본 설정되어 있고, 기본 서브넷은 이 속성이 `true`로 기본 설정되어 있습니다. Amazon EC2 인스턴스 실행 마법사에서 생성되는 기본이 아닌 서브넷은 예외로, 마법사는 이 속성을 `true`로 설정합니다. Amazon VPC 콘솔을 사용하여 이 속성을 수정할 수 있습니다.

서브넷의 퍼블릭 IPv4 주소 지정 동작을 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷을 선택하고 Subnet Actions를 선택한 후, Modify auto-assign IP settings를 선택합니다.
4. Enable auto-assign public IPv4 address 확인란을 선택하면 선택된 서브넷에서 시작된 모든 인스턴스에 대한 퍼블릭 IPv4 주소를 요청합니다. 필요에 따라 확인란을 선택하거나 선택 취소한 후 Save를 선택합니다.

## 서브넷의 퍼블릭 IPv6 주소 지정 속성 수정

모든 서브넷에는 IPv6 주소 지정 속성이 `false`로 기본 설정되어 있습니다. Amazon VPC 콘솔을 사용하여 이 속성을 수정할 수 있습니다. 서브넷에서 IPv6 주소 지정 속성을 사용하는 경우, 해당 서브넷에서 생성된 네트워크 인터페이스는 서브넷의 범위에 속하는 IPv6 주소를 받습니다. 서브넷에서 시작한 인스턴스는 주 네트워크 인터페이스에서 IPv6 주소를 받습니다.

서브넷에는 연결된 IPv6 CIDR 블록이 있어야 합니다.

### Note

서브넷에서 IPv6 주소 지정 기능을 사용하는 경우, 네트워크 인터페이스 또는 인스턴스는 Amazon EC2 API 2016-11-15 이상 버전에서 생성된 경우에만 IPv6 주소를 받습니다. Amazon EC2 콘솔에서는 최신 API 버전을 사용합니다.

서브넷의 IPv6 주소 지정 동작을 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷을 선택하고 Subnet Actions를 선택한 후, Modify auto-assign IP settings를 선택합니다.
4. [Enable auto-assign IPv6 address] 확인란을 선택하면 선택된 서브넷에서 생성된 모든 네트워크 인터페이스에 대한 IPv6 주소를 요청합니다. 필요에 따라 확인란을 선택하거나 선택 취소한 후 Save를 선택합니다.

## 인스턴스 시작 시 퍼블릭 IPv4 주소 배정

시작 시 기본 또는 기본이 아닌 서브넷의 인스턴스에 퍼블릭 IPv4 주소를 할당할지 여부를 제어할 수 있습니다.

### Important

사용자는 인스턴스가 시작된 이후에는 퍼블릭 IPv4 주소를 수동으로 해제할 수 없습니다. 대신, 특정 조건에 자동으로 해제되고 그 이후에 사용자는 해당 주소를 다시 사용할 수 없습니다. 자유롭게 연결하거나 해제할 수 있는 영구 퍼블릭 IP 주소가 필요한 경우 시작 후에 탄력적 IP 주소를 인스턴스와 연결합니다. 자세한 내용은 [탄력적 IP 주소 \(p. 197\)](#) 단원을 참조하십시오.

시작하는 과정에서 인스턴스에 퍼블릭 IPv4 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 Next: Configure Instance Details를 선택합니다.
4. [Configure Instance Details] 페이지의 [Network] 목록에서 VPC를 선택합니다. [Auto-assign Public IP] 목록이 표시됩니다. [Enable] 또는 [Disable]을 선택하여 서브넷의 기본 설정을 재정의합니다.

### Important

네트워크 인터페이스를 한 개 이상 지정하면 퍼블릭 IPv4 주소를 할당할 수 없습니다. 또한 eth0에 대해 기존 네트워크 인터페이스를 지정하면 퍼블릭 IPv4 자동 할당 기능을 사용하여 서브넷 설정을 재정의할 수 없습니다.

5. 마법사의 나머지 단계를 수행하여 인스턴스를 시작합니다.
6. [Instances] 화면에서 인스턴스를 선택합니다. Description 탭의 IPv4 Public IP 필드에서 해당 인스턴스의 퍼블릭 IP 주소를 볼 수 있습니다. 또는 탐색 창에서 Network Interfaces를 선택한 후, 해당 인스턴스의 eth0 네트워크 인터페이스를 선택합니다. IPv4 Public IP 필드에서 퍼블릭 IP 주소를 볼 수 있습니다.

### Note

퍼블릭 IPv4 주소는 콘솔에서 네트워크 인터페이스의 속성으로 표시되지만 NAT를 통해 주 프라이빗 IPv4 주소와 매핑됩니다. 따라서 예를 들어 Windows 인스턴스의 경우 `ipconfig` 또는 Linux 인스턴스의 경우 `ifconfig`를 통해 인스턴스의 네트워크 인터페이스 속성을 확인할 경우 퍼블릭 IP 주소는 표시되지 않습니다. 인스턴스의 퍼블릭 IP 주소를 인스턴스 내에서 결정하려면 인스턴스 메타데이터를 사용할 수 있습니다. 자세한 내용은 [인스턴스 메타데이터 및 사용자 데이터를 참조하십시오](#).

이 기능은 시작 시에만 사용할 수 있습니다. 그러나 시작 도중에 퍼블릭 IPv4 주소를 인스턴스에 할당할지의 여부와는 관계없이 시작 후에는 인스턴스와 탄력적 IP 주소를 연결할 수 있습니다. 자세한 내용은 [탄력적 IP 주소 \(p. 197\)](#) 단원을 참조하십시오.

## 인스턴스 시작 시 IPv6 주소 배정

시작하는 과정에서 인스턴스에 IPv6 주소를 자동 할당할 수 있습니다. 이를 위해서는 [연결된 IPv6 CIDR 블록 \(p. 67\)](#)이 있는 VPC 및 서브넷으로 인스턴스를 시작해야 합니다. IPv6 주소는 서브넷의 범위로부터 할당되고, 주 네트워크 인터페이스(eth0)에 할당됩니다.

시작하는 과정에서 인스턴스에 IPv6 주소를 자동 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.

### Note

IPv6 주소를 지원하는 인스턴스에 유형을 선택합니다.

4. Configure Instance Details 페이지의 Network에서 VPC를, Subnet에서 서브넷을 선택합니다. Auto-assign IPv6 IP에 대해 Enable을 선택합니다.



5. 마법사의 나머지 단계를 수행하여 인스턴스를 시작합니다.

아니면 시작하는 도중 서브넷 범위에 속한 특정 IPv6 주소를 인스턴스에 할당하려면 그 주소를 인스턴스에 대한 주 네트워크 인터페이스에 할당할 수 있습니다.

시작하는 과정에서 인스턴스에 특정 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.
3. AMI와 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.

#### Note

IPv6 주소를 지원하는 인스턴스에 유형을 선택합니다.

4. Configure Instance Details 페이지의 Network에서 VPC를, Subnet에서 서브넷을 선택합니다.
5. Network Interfaces 섹션으로 이동합니다. eth0 네트워크 인터페이스의 경우, IPv6 IPs에서 Add IP를 선택합니다.
6. 서브넷의 주소 범위에 속한 IPv6 주소를 입력합니다.
7. 마법사의 나머지 단계를 수행하여 인스턴스를 시작합니다.

시작하는 도중 인스턴스에 다중 IPv6 주소를 할당하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [다중 IPv6 주소 작업](#) 단원을 참조하십시오.

## 인스턴스에 IPv6 주소 할당

인스턴스가 [연결된 IPv6 CIDR 블록 \(p. 67\)](#)이 있는 VPC와 서브넷에 있는 경우, Amazon EC2 콘솔을 사용하여 IPv6 주소를 서브넷 범위로부터 인스턴스에 할당할 수 있습니다.

IPv6 주소를 인스턴스에 연결하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택하고 인스턴스를 선택합니다.
3. [Actions], [Manage IP Addresses]를 선택합니다.
4. IPv6 Addresses에서 Assign new IP를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 [Auto-assign] 값을 그대로 둡니다.
5. [Save]를 선택합니다.

또는 네트워크 인터페이스에 IPv6 주소 한 개를 할당할 수 있습니다. 자세한 내용은 `&guide-ec2-user;`의 탄력적 네트워크 인터페이스 단원에서 [IPv6 주소 할당](#)을 참조하십시오.

## 인스턴스에 할당된 IPv6 주소 해제

인스턴스에 IPv6 주소가 더 이상 필요하지 않을 경우, Amazon EC2 콘솔을 사용하여 인스턴스에서 해당 주소를 해제할 수 있습니다.

인스턴스에 연결된 IPv6 주소를 해제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택하고 인스턴스를 선택합니다.
3. Actions, Manage IP Addresses를 선택합니다.
4. IPv6 Addresses에서 해당 IPv6 주소에 대해 Unassign을 선택합니다.
5. Save를 선택합니다.



또는 네트워크 인터페이스에 연결된 IPv6 주소를 해제할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스 단원에서 [IPv6 주소 할당 해제](#)를 참조하십시오.

## API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 섹션을 참조하십시오.

### 시작 시 퍼블릭 IPv4 주소 배정

- [run-instances](#) 명령에서 `--associate-public-ip-address` 또는 `--no-associate-public-ip-address` 옵션을 사용합니다. (AWS CLI)
- [New-EC2Instance](#) 명령에서 `-AssociatePublicIp` 파라미터를 사용합니다. (Windows PowerShell용 AWS 도구)

### 시작 시 IPv6 주소 배정

- [run-instances](#) 명령으로 `--ipv6-addresses` 옵션을 사용합니다. (AWS CLI)
- [New-EC2Instance](#) 명령에서 `-Ipv6Addresses` 파라미터를 사용합니다. (Windows PowerShell용 AWS 도구)

### 서브넷의 IP 주소 지정 동작 수정

- [modify-subnet-attribute](#)(AWS CLI)
- [Edit-EC2SubnetAttribute](#)(Windows PowerShell용 AWS 도구)

### 네트워크 인터페이스에 IPv6 주소 할당

- [assign-ipv6-addresses](#)(AWS CLI)
- [Register-EC2Ipv6AddressList](#)(Windows PowerShell용 AWS 도구)

### 네트워크 인터페이스에 할당된 IPv6 주소 해제

- [unassign-ipv6-addresses](#)(AWS CLI)
- [Unregister-EC2Ipv6AddressList](#)(Windows PowerShell용 AWS 도구)

## IPv6로 마이그레이션하기

기존 VPC가 IPv4만을 지원하는 경우, 그리고 서브넷에 IPv4만을 사용하도록 구성된 리소스를 보유한 경우, VPC 및 리소스에 대한 IPv6 지원이 가능하도록 할 수 있습니다. VPC는 듀얼 스택 모드로 작동할 수 있으므로, 리소스는 IPv4나 IPv6, 또는 둘 다를 통해 통신할 수 있습니다. IPv4 및 IPv6 통신 프로토콜은 상호 독립적입니다.

VPC 및 서브넷에 대한 IPv4 지원은 Amazon VPC 및 Amazon EC2의 기본 IP 주소 지정 시스템이므로 비활성화할 수 없습니다.

다음 표는 IPv6를 사용하도록 VPC 및 서브넷을 활성화하는 단계를 간략히 설명합니다.

단계	참고
1단계: <a href="#">IPv6 CIDR 블록을 VPC 및 서브넷에 연결 (p. 85)</a>	Amazon 제공 IPv6 CIDR 블록을 VPC 및 서브넷에 연결합니다.

단계	참고
2단계: 외부 전용 인터넷 게이트웨이 생성 및 구성 (p. 86)	NAT 디바이스는 IPv6 트래픽을 지원하지 않습니다. 따라서 IPv6를 통해 인터넷에 대한 아웃바운드 통신을 가능케 하고 인바운드 통신을 막으려면 프라이빗 서브넷에 외부 전용 인터넷 게이트웨이를 생성합니다. 외부 전용 인터넷 게이트웨이는 IPv6 트래픽만 지원합니다.
3단계: 라우팅 테이블 업데이트 (p. 86)	라우팅 테이블을 업데이트하여 IPv6 트래픽을 라우팅합니다. 예를 들어, 퍼블릭 서브넷에서 모든 IPv6 트래픽을 인터넷 게이트웨이로 라우팅하는 경로를 생성하고, 프라이빗 서브넷에서 모든 인터넷 바인딩된 IPv6 트래픽을 외부 전용 인터넷 게이트웨이로 라우팅하는 경로를 생성합니다.
4단계: 보안 그룹 규칙 업데이트 (p. 86)	보안 그룹 규칙을 업데이트하여 IPv6 주소용 규칙을 포함합니다. 이렇게 하면 IPv6 트래픽이 인스턴스로 그리고 인스턴스로부터 흐르도록 할 수 있습니다. 서브넷으로 가는, 그리고 서브넷에서 나오는 트래픽의 흐름을 제어하기 위해 사용자 지정 네트워크 ACL 규칙을 생성한 경우, IPv6 트래픽에 대한 규칙을 포함시켜야 합니다.
5단계: 인스턴스 유형 변경 (p. 87)	인스턴스 유형이 IPv6를 지원하지 않은 경우, 인스턴스 유형을 변경합니다.
6단계: 인스턴스에 IPv6 주소 할당 (p. 88)	서브넷의 IPv6 주소 범위에서 인스턴스에 IPv6 주소를 할당합니다.
(선택 사항) 인스턴스에서 IPv6 구성하기 (p. 88)	DHCPv6를 사용하도록 구성되지 않은 AMI에서 인스턴스를 시작하는 경우, 이 인스턴스에 지정된 IPv6 주소를 인식하도록 인스턴스를 수동으로 구성해야 합니다.

IPv6를 사용하여 마이그레이션하기 전에 Amazon VPC에 대한 IPv6 주소 지정 기능을 읽어 보시기 바랍니다. IPv4 및 IPv6의 특징과 제한 (p. 76).

## 예: 퍼블릭 및 프라이빗 서브넷이 있는 VPC에서 IPv6 사용

이 예에서는 VPC에 퍼블릭 및 프라이빗 서브넷이 있습니다. VPC의 NAT 게이트웨이를 통해 인터넷과 아웃바운드 통신을 하는 프라이빗 서브넷에 데이터베이스 인스턴스가 있습니다. 인터넷 게이트웨이를 통해 인터넷에 액세스할 수 있는 퍼블릭 서브넷에 퍼블릭 웹 서버가 있습니다. 다음 다이어그램은 VPC의 아키텍처를 보여줍니다.

웹 서버 보안 그룹(`sg-11aa22bb`)에는 다음과 같은 인바운드 규칙이 포함되어 있습니다.

Type	프로토콜	포트 범위	소스	Comment
모든 트래픽	모두	모두	sg-33cc44dd	sg-33cc44dd(데이터베이스 인스턴스)와 연결된 인스턴스에서 나오는 모든 트래픽에 대한 인바운드 액세스를 허용

Type	프로토콜	포트 범위	소스	Comment
HTTP	TCP	80	0.0.0.0/0	HTTP를 통해 인터넷에서 오는 인바운드 트래픽을 허용
HTTPS	TCP	443	0.0.0.0/0	HTTPS를 통해 인터넷에서 오는 인바운드 트래픽을 허용
SSH	TCP	22	203.0.113.123/32	로컬 컴퓨터에서 접근하는 인바운드 SSH 액세스를 허용. 예를 들면, 관리 작업을 수행하기 위해 인스턴스에 연결해야 하는 경우.

데이터베이스 인스턴스 보안 그룹(`sg-33cc44dd`)에는 다음과 같은 인바운드 규칙이 포함되어 있습니다.

Type	프로토콜	포트 범위	소스	Comment
MySQL	TCP	3306	sg-11aa22bb	sg-11aa22bb(웹 서버 인스턴스)와 연결된 인스턴스에서 나오는 MySQL 트래픽에 대한 인바운드 액세스를 허용

두 보안 그룹에는 모든 아웃바운드 IPv4 트래픽을 허용하는 기본 아웃바운드 규칙이 있고, 다른 아웃바운드 규칙은 없습니다.

웹 서버는 `t2.medium` 인스턴스 유형입니다. 데이터베이스 서버는 `m3.large`입니다.

VPC 및 리소스에서 IPv6를 사용하고 VPC 및 리소스를 듀얼 스택 모드로 작동하고 싶은 경우가 있습니다. 바꿔 말하면 인터넷을 통해 VPC 및 리소스의 리소스들 간에 IPv6 및 IPv4 주소 지정을 모두 사용하는 것입니다.

해당 절차를 마치고 나면 VPC는 다음과 같이 구성됩니다.

## 1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결

IPv6 CIDR 블록을 VPC와 연결한 다음, 그 범위의 /64 CIDR 블록을 각 서브넷에 연결할 수 있습니다.

IPv6 CIDR 블록을 VPC와 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택하고 Actions, Edit CIDRs를 선택합니다.
4. Add IPv6 CIDR을 선택합니다. IPv6 CIDR 블록을 추가한 후 Close를 선택합니다.

IPv6 CIDR 블록을 서브넷에 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷을 선택하고 Subnet Actions, Edit IPv6 CIDRs를 선택합니다.
4. Add IPv6 CIDR을 선택합니다. 서브넷에 16진수 페어(예: 00)를 지정하고 체크 표시 아이콘을 선택하여 항목을 확정합니다.
5. Close를 선택합니다. VPC의 다른 서브넷에 대해 이 절차를 반복합니다.

자세한 내용은 [IPv6의 경우, VPC 및 서브넷 크기 조정 \(p. 64\)](#) 단원을 참조하십시오.

## 2단계: 외부 전용 인터넷 게이트웨이 생성 및 구성

VPC에 외부 전용 인터넷 게이트웨이를 생성합니다.

외부 전용 인터넷 게이트웨이를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Egress Only Internet Gateways, Create Egress Only Internet Gateway를 선택합니다.
3. 외부 전용 인터넷 게이트웨이를 생성할 VPC를 선택합니다. Create를 선택합니다.

자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

## 3단계: 라우팅 테이블 업데이트

위의 예에서는 프라이빗 서브넷에 대한 사용자 지정 라우팅 테이블을 업데이트하여 데이터베이스 인스턴스에서 IPv6 트래픽을 위한 외부 전용 인터넷 게이트웨이를 사용할 수 있도록 합니다. 또한 퍼블릭 서브넷에 대한 사용자 지정 라우팅 테이블을 업데이트하여 웹 서버가 IPv6 트래픽을 위한 인터넷 게이트웨이를 사용할 수 있도록 해야 합니다.

라우팅 테이블을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route Tables를 선택하고 프라이빗 서브넷에 연결된 라우팅 테이블을 선택합니다.
3. Routes 탭에서 Edit를 선택하고, Destination에 `::/0`을 지정하고, Target에 대해 외부 전용 인터넷 게이트웨이 ID를 선택한 다음, Save를 선택합니다.
4. 퍼블릭 서브넷과 연결된 라우팅 테이블을 선택합니다.
5. Routes 탭에서 Edit를 선택하고, Destination에 `::/0`을 지정하고, Target에 대해 인터넷 게이트웨이 ID를 선택한 다음, Save를 선택합니다.

자세한 내용은 [라우팅 옵션 \(p. 157\)](#) 단원을 참조하십시오.

## 4단계: 보안 그룹 규칙 업데이트

위의 예에서 웹 서버 보안 그룹(`sg-11aa22bb`)은 특정 포트를 통해 IPv4 주소로부터 이루어지는 인바운드 액세스를 허용합니다. IPv6 주소로부터 이루어지는 동일한 인바운드 액세스, 즉 모든 IPv6 주소로부터 이루어지는 HTTP 및 HTTPS 액세스와 특정 IPv6 주소로부터 이루어지는 SSH 액세스(해당되는 경우)를 허용하는 규칙을 추가해야 합니다. 데이터베이스 보안 그룹에 대한 인바운드 규칙을 변경할 필요는 없습니다. `sg-11aa22bb`의 모든 통신을 허용하는 규칙에는 IPv6 통신이 기본적으로 포함되어 있습니다.

보안 그룹 규칙을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security Groups를 선택하고 웹 서버 보안 그룹을 선택합니다.

3. Inbound Rules 탭에서 Edit를 선택합니다.
4. 다음과 같이 각 규칙에 대해 Add another rule을 선택하고, 다 되었으면 Save를 선택합니다.
  - Type에 대해 HTTP를 선택합니다. Source에 `::/0`을 입력합니다.
  - Type에 대해 HTTPS를 선택합니다. Source에 `::/0`을 입력합니다.
  - Type에 대해 SSH를 선택합니다. Source에 로컬 컴퓨터의 IPv6 주소 또는 로컬 네트워크 주소 범위를 입력합니다.

이 시나리오에서는 IPv6 CIDR 블록을 VPC에 연결하면 모든 IPv6 트래픽을 허용하는 아웃바운드 규칙이 보안 그룹에 자동으로 추가됩니다. 그러나 보안 그룹에 대한 원본 아웃바운드 규칙을 수정한 경우, 이 규칙은 자동으로 추가되지 않으므로 IPv6 트래픽에 대한 동등한 수준의 아웃바운드 규칙을 추가해야 합니다. 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#) 단원을 참조하십시오.

#### 네트워크 ACL 규칙 업데이트

IPv6 CIDR 블록을 VPC에 연결할 때 사용자가 기본 규칙을 수정하지 않았다면 AWS가 자동으로 기본 네트워크 ACL에 규칙을 추가하여 IPv6 트래픽을 허용합니다. 기본 네트워크 ACL을 변경한 경우, 또는 서브넷으로 가는, 그리고 서브넷에서 나오는 트래픽의 흐름을 제어하기 위해 규칙이 있는 사용자 지정 네트워크 ACL을 생성한 경우에는 IPv6 트래픽에 대한 규칙을 수동으로 추가해야 합니다. 프라이빗 및 퍼블릭 서브넷이 있는 VPC의 바람직한 네트워크 ACL 규칙에 대한 자세한 내용은 [시나리오 2를 위한 권장 규칙 \(p. 118\)](#) 단원을 참조하십시오.

## 5단계: 인스턴스 유형 변경

이 예시에서 웹 서버 인스턴스는 `t2.medium` 인스턴스 유형으로서 IPv6를 지원합니다. 데이터베이스 인스턴스는 `m3.large` 인스턴스 유형으로서 IPv6를 지원하지 않습니다. 지원되는 인스턴스 유형에 맞게 인스턴스 크기를 조정해야 합니다(예: `m4.large`). 자세한 내용은 [인스턴스 유형](#) 단원을 참조하십시오.

인스턴스 크기를 조정하려면 호환성 관련 제한 사항에 주의해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 크기 조정을 위한 호환성](#) 단원을 참조하십시오. 이 시나리오에서는 HVM 가상화를 사용하는 AMI에서 데이터베이스 인스턴스를 시작한 경우, 다음 절차를 사용하여 `m4.large` 인스턴스 유형으로 인스턴스 크기를 조정할 수 있습니다.

#### Important

인스턴스 크기를 조정하려면 인스턴스를 중단해야 합니다. 인스턴스를 중단했다가 시작하면 해당 인스턴스에 대한 퍼블릭 IPv4 주소가 변경됩니다(퍼블릭 IPv4 주소가 있는 경우). 인스턴스 스토어 볼륨에 저장된 데이터가 있으면 데이터가 삭제됩니다.

#### 인스턴스 크기를 조정하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택하고 데이터베이스 인스턴스를 선택합니다.
3. Actions, Instance State, Stop을 차례로 선택합니다.
4. 확인 대화 상자가 나타나면 Yes, Stop을 선택합니다.
5. 인스턴스를 선택한 상태에서 Actions, Instance Settings, Change Instance Type를 선택합니다.
6. Instance Type에 대해 `m4.large`, Apply를 선택합니다.
7. 중지된 인스턴스를 다시 시작하려면 인스턴스를 선택하고 Actions, Instance State, Start를 선택합니다. 확인 대화 상자가 나타나면 [Yes, Start]를 선택합니다.

인스턴스가 인스턴스 스토어 지원 AMI인 경우, 앞서 설명한 절차를 통해서도 인스턴스 크기를 조정할 수 없습니다. 그 대신에 인스턴스에서 인스턴스 스토어 지원 AMI를 생성하고 새 인스턴스 유형을 사용하여 AMI에서 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스](#)

스토어 지원 Linux AMI 생성 단원과 Windows 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 스토어 지원 Windows AMI 생성 단원](#)을 참조하십시오.

호환성 관련 제한 사항이 있는 경우, 새 인스턴스 유형으로 마이그레이션하지 못할 수 있습니다. 예를 들면, PV 가상화를 사용하는 AMI에서 인스턴스를 시작하는 경우, PV 가상화 및 IPv6를 둘 다 지원하는 인스턴스 유형은 C3밖에 없습니다. 이 인스턴스 유형은 요구 사항에 적합하지 않을 수 있습니다. 이 경우에는 기본 HVM AMI에 소프트웨어를 다시 설치한 후 새 인스턴스를 시작해야 할 수 있습니다.

새 AMI에서 인스턴스를 실행하는 경우, 시작하는 과정에서 인스턴스에 IPv6 주소를 할당할 수 있습니다.

## 6단계: 인스턴스에 IPv6 주소 할당

인스턴스 유형이 IPv6를 지원한다는 것을 확인했으면 Amazon EC2 콘솔을 사용하여 인스턴스에 IPv6 주소를 할당할 수 있습니다. IPv6 주소는 인스턴스에 대한 주 네트워크 인터페이스(eth0)에 할당됩니다.

인스턴스에 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스를 선택하고 Actions, Networking, Manage IP Addresses를 선택합니다.
4. IPv6 Addresses에서 Assign new IP를 선택합니다. 서브넷 범위에 속한 특정 IPv6 주소를 입력하거나, Amazon이 자동으로 선택하도록 기본 Auto-Assign 값을 그대로 둘 수 있습니다.
5. Yes, Update를 선택합니다.

아니면 교체 인스턴스를 시작하는 경우(예를 들어, 인스턴스 크기를 조정할 수 없어 그 대신에 새 AMI를 생성한 경우), 시작하는 과정에서 IPv6를 할당할 수 있습니다.

시작하는 과정에서 인스턴스에 IPv6 주소를 할당하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. AMI와 IPv6 호환 인스턴스 유형을 선택한 후 Next: Configure Instance Details를 선택합니다.
3. Configure Instance Details 페이지의 Network에서 VPC를, Subnet에서 서브넷을 선택합니다. Auto-assign IPv6 IP에서 Enable을 선택합니다.
4. 마법사의 나머지 단계를 수행하여 인스턴스를 시작합니다.

## (선택 사항) 인스턴스에서 IPv6 구성하기

Amazon Linux 2016.09.0 이상 버전 또는 Windows Server 2008 R2 이상 버전을 사용하여 인스턴스를 시작한 경우, 인스턴스는 IPv6에 맞게 구성되어 있으므로 추가 절차는 필요하지 않습니다.

다른 AMI에서 인스턴스를 시작한 경우, DHCPv6에 맞게 구성되지 않을 수 있습니다. 따라서 인스턴스에 할당하는 어떤 IPv6 주소도 주 네트워크 인터페이스에서 자동 인식되지 않습니다. 네트워크 인터페이스에서 IPv6 주소가 구성되었는지 여부를 확인하려면 Linux에서는 `ifconfig` 명령, Windows에서는 `ipconfig` 명령을 실행합니다.

다음 절차를 수행하여 인스턴스를 구성할 수 있습니다. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#) 또는 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

항목

- [Amazon Linux \(p. 89\)](#)
- [Ubuntu \(p. 89\)](#)

- [RHEL/CentOS \(p. 91\)](#)
- [Windows가 설치된 \(p. 92\)](#)

## Amazon Linux

Amazon Linux에서 DHCPv6을 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. 다음과 같이 인스턴스에 최신 소프트웨어 패키지를 가져옵니다.

```
sudo yum update -y
```

3. 선택한 텍스트 편집기를 사용하여 `/etc/sysconfig/network-scripts/ifcfg-eth0`를 열고 다음과 같은 줄을 검색합니다.

```
IPV6INIT=no
```

이 줄을 다음과 같이 바꿉니다.

```
IPV6INIT=yes
```

다음과 같이 두 줄을 추가하고 변경 내용을 저장합니다.

```
DHCPV6C=yes  
DHCPV6C_OPTIONS=-nw
```

4. `/etc/sysconfig/network`을 열어 다음과 같은 줄을 제거하고 변경 내용을 저장합니다.

```
NETWORKING_IPV6=no  
IPV6INIT=no  
IPV6_ROUTER=no  
IPV6_AUTOCONF=no  
IPV6FORWARDING=no  
IPV6TO4INIT=no  
IPV6_CONTROL_RADVD=no
```

5. `/etc/hosts`를 열어 다음과 같이 내용을 교체하고 변경 내용을 저장합니다.

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
:::1        localhost6 localhost6.localdomain6
```

6. 인스턴스를 재부팅합니다. 인스턴스에 다시 연결하고 `ifconfig` 명령을 사용하여 IPv6 주소가 주 네트워크 인터페이스에서 인식이 되는지 확인합니다.

## Ubuntu

네트워크 인터페이스에 할당된 IPv6 주소를 동적으로 인식하도록 Ubuntu 인스턴스를 구성할 수 있습니다. 인스턴스에 IPv6 주소가 없는 경우 이렇게 구성하면 인스턴스 부팅 시간이 최대 5분까지 연장될 수 있습니다.

이러한 단계는 루트 사용자로 수행해야 합니다.

항목

- [Ubuntu Server 16 \(p. 90\)](#)

- [Ubuntu Server 14 \(p. 90\)](#)
- [DHCPv6 클라이언트 시작 \(p. 91\)](#)

## Ubuntu Server 16

실행 중인 Ubuntu Server 16 인스턴스에서 IPv6를 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. `/etc/network/interfaces.d/50-cloud-init.cfg` 파일 내용을 확인합니다.

```
cat /etc/network/interfaces.d/50-cloud-init.cfg

# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

루프백 네트워크 디바이스(lo)가 구성되어 있는지 확인하고 네트워크 인스턴스의 이름을 기록해 둡니다. 이 예제의 네트워크 인스턴스 이름은 `eth0`입니다. 이름은 인스턴스 유형에 따라 달라질 수 있습니다.

3. `/etc/network/interfaces.d/60-default-with-ipv6.cfg` 파일을 생성하고 다음 줄을 추가합니다. 필요한 경우 `eth0`을 위 단계에서 검색된 네트워크 인터페이스의 이름으로 대체합니다.

```
iface eth0 inet6 dhcp
```

4. 인스턴스를 재부팅하거나, 다음 명령을 실행하여 네트워크 인스턴스를 재시작합니다. 필요한 경우 `eth0`을 네트워크 인터페이스의 이름으로 대체합니다.

```
sudo ifdown eth0 ; sudo ifup eth0
```

5. 인스턴스에 다시 연결하고 `ifconfig` 명령을 사용하여 네트워크 인터페이스에서 IPv6 주소가 구성되었는지 확인합니다.

사용자 데이터를 사용하여 IPv6를 구성하려면

새 Ubuntu 인스턴스를 시작하고 시작되는 동안 다음 사용자 데이터를 지정하여 인스턴스에 할당된 IPv6 주소가 네트워크 인터페이스에 자동으로 구성되는지 확인합니다.

```
#!/bin/bash
echo "iface eth0 inet6 dhcp" >> /etc/network/interfaces.d/60-default-with-ipv6.cfg
dhclient -6
```

이 경우 IPv6 주소를 구성하기 위해 인스턴스에 연결할 필요가 없습니다.

자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Running Commands on Your Linux Instance at Launch](#) 단원을 참조하십시오.

## Ubuntu Server 14

Ubuntu Server 14를 사용하는 경우 듀얼 스택 네트워크 인터페이스를 시작할 때 발생하는 [알려진 문제](#)에 대한 차선책을 포함시켜야 합니다(재시작 시 인스턴스 접속 불가 제한 시간이 연장됨).



이러한 단계는 루트 사용자로 수행해야 합니다.

실행 중인 Ubuntu Server 14 인스턴스에서 IPv6를 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. `/etc/network/interfaces.d/eth0.cfg` 파일을 편집하여 다음 항목을 포함시킵니다.

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
        up dhclient -6 $IFACE
```

3. 인스턴스를 재부팅합니다:

```
sudo reboot
```

4. 인스턴스에 다시 연결하고 `ifconfig` 명령을 사용하여 네트워크 인터페이스에서 IPv6 주소가 구성되었는지 확인합니다.

### DHCPv6 클라이언트 시작

추가 구성을 수행하지 않고 네트워크 인터페이스에 대한 IPv6 주소를 즉시 불러오려면 인스턴스에 대해 DHCPv6 클라이언트를 시작할 수 있습니다. 하지만 IPv6 주소는 재부팅할 경우 네트워크 인터페이스에 대해 지속되지 않습니다.

Ubuntu에서 DHCPv6 클라이언트를 시작하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. DHCPv6 클라이언트 시작:

```
sudo dhclient -6
```

3. `ifconfig` 명령을 사용하여 IPv6 주소가 주 네트워크 인터페이스에서 인식이 되는지 확인합니다.

### RHEL/CentOS

RHEL 7 또는 CentOS 7에서 DHCPv6를 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. 선택한 텍스트 편집기를 사용하여 `/etc/sysconfig/network-scripts/ifcfg-eth0`를 열고 다음과 같은 줄을 검색합니다.

```
IPV6INIT="no"
```

이 줄을 다음과 같이 바꿉니다.

```
IPV6INIT="yes"
```

다음과 같이 두 줄을 추가하고 변경 내용을 저장합니다.

```
DHCPV6C=yes
NM_CONTROLLED=no
```

3. `/etc/sysconfig/network`를 열고 다음 행을 추가하거나 수정한 다음 변경 내용을 저장합니다.

```
NETWORKING_IPV6=yes
```

4. 다음 명령을 실행하여 인스턴스에서 네트워킹을 다시 시작합니다.

```
sudo service network restart
```

ifconfig 명령을 사용하여 IPv6 주소가 주 네트워크 인터페이스에서 인식이 되는지 확인할 수 있습니다.

## RHEL 6 또는 CentOS 6에서 DHCPv6를 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. 위의 RHEL 7/CentOS 7 구성 절차에서 2 ~ 4단계를 따릅니다.
3. 네트워킹을 다시 시작했을 때 IPv6 주소를 가져올 수 없는 오류가 발생할 경우 `/etc/sysconfig/network-scripts/ifup-eth`를 열고 다음 행을 찾습니다(기본적으로 행 327).

```
if /sbin/dhclient "$DHCLIENTARGS"; then
```

`$DHCLIENTARGS`를 묶은 따옴표를 제거하고 변경 사항을 저장합니다. 인스턴스에서 네트워킹을 다시 시작합니다.

```
sudo service network restart
```

## Windows가 설치된

Windows Server 2003 및 Windows Server 2008 SP2에서 IPv6를 구성하려면 다음 절차를 사용합니다.

IPv4가 아니라 IPv6가 기본 설정되도록 하려면 Microsoft 지원 페이지 <https://support.microsoft.com/en-us/help/929852/how-to-disable-ipv6-or-its-components-in-windows>에서 Prefer IPv6 over IPv4 in prefix policies 픽스를 다운로드합니다.

## Windows Server 2003에서 IPv6를 사용 및 구성하려면

1. [describe-instances](#) AWS CLI 명령을 사용하거나 Amazon EC2 콘솔에서 해당 인스턴스에 대한 IPv6 IPs 필드를 확인하여 인스턴스의 IPv6 주소를 얻습니다.
2. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
3. 인스턴스 내부에서 Start, Control Panel, Network Connections, Local Area Connection을 선택합니다.
4. Properties를 선택한 다음, Install을 선택합니다.
5. Protocol, Add를 차례로 선택합니다. Network Protocol 목록에서 Microsoft TCP/IP version 6을 선택한 다음, OK를 선택합니다.
6. 명령 프롬프트를 열고 네트워크 셸을 시작합니다.

```
netsh
```

7. 인터페이스 IPv6 컨텍스트로 전환합니다.

```
interface ipv6
```

8. 다음 명령을 사용하여 로컬 영역 연결에 IPv6 주소를 추가합니다. IPv6 주소의 값을 인스턴스의 IPv6 주소로 교체합니다.

```
add address "Local Area Connection" "ipv6-address"
```

예:

```
add address "Local Area Connection" "2001:db8:1234:1a00:1a01:2b:12:d08b"
```

9. 네트워크 셸을 종료합니다.

```
exit
```

10. `ipconfig` 명령을 사용하여 IPv6 주소가 로컬 영역 연결에 대해 인식이 되는지 확인합니다.

#### Windows Server 2008 SP2에서 IPv6를 사용 및 구성하려면

1. [describe-instances](#) AWS CLI 명령을 사용하거나 Amazon EC2 콘솔에서 해당 인스턴스에 대한 IPv6 IPs 필드를 확인하여 인스턴스의 IPv6 주소를 얻습니다.
2. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 Windows 인스턴스에 연결합니다.
3. Start, Control Panel을 선택합니다.
4. Network and Sharing Center를 연 다음, Network Connections를 엽니다.
5. 네트워크 인터페이스에 해당되는 Local Area Network를 마우스 오른쪽 버튼으로 클릭하고 Properties를 선택합니다.
6. Internet Protocol Version 6 (TCP/IPv6)의 확인란을 선택하고 OK를 선택합니다.
7. 로컬 영역 네트워크의 속성 대화 상자를 다시 엽니다. Internet Protocol Version 6 (TCP/IPv6)를 선택하고 Properties를 선택합니다.
8. Use the following IPv6 address를 선택하고 다음 작업을 수행합니다.
  - IPv6 Address에 1단계에서 얻은 IPv6 주소를 입력합니다.
  - Subnet prefix length에 64를 입력합니다.
9. OK를 선택하여 속성 대화 상자를 닫습니다.
10. 명령 프롬프트를 엽니다. `ipconfig` 명령을 사용하여 IPv6 주소가 로컬 영역 연결에 대해 인식이 되는지 확인합니다.

# 보안

Amazon VPC는 다음과 같이 VPC의 보안을 강화하고 모니터링하는 데 사용할 수 있는 세 가지 기능을 제공합니다.

- 보안 그룹 - 연결된 Amazon EC2 인스턴스에 대한 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다.
- 네트워크 ACL(액세스 제어 목록) - 연결된 서브넷에 대해 방화벽 역할을 하여 서브넷 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다.
- 흐름 로그 - VPC의 네트워크 인터페이스에서 송수신되는 IP 트래픽에 대한 정보를 수집합니다.

VPC에서 인스턴스를 시작할 경우 생성된 보안 그룹을 한 개 이상 연결할 수 있습니다. VPC의 각 인스턴스는 서로 다른 보안 그룹 세트에 속할 수 있습니다. 인스턴스를 시작할 때 보안 그룹을 지정하지 않으면 VPC의 기본 보안 그룹에 자동으로 속하게 됩니다. 보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참조하십시오.

보안 그룹만 사용하여 VPC 인스턴스를 보호할 수 있지만 2차 보안 계층으로 네트워크 ACL을 추가할 수 있습니다. 네트워크 ACL에 대한 자세한 내용은 [네트워크 ACL \(p. 102\)](#)을 참조하십시오.

VPC, 서브넷 또는 개별 네트워크 인터페이스에 대한 흐름 로그를 생성하여 인스턴스에서 송신하고 수신하는 IP 트래픽의 수락과 거부를 모니터링할 수 있습니다. 흐름 로그 데이터는 CloudWatch Logs에 게시되며 너무 거부적이거나 허용적인 보안 그룹과 네트워크 ACL 규칙을 진단하는 데 도움이 됩니다. 자세한 내용은 [VPC 흐름 로그 \(p. 144\)](#)를 참조하십시오.

AWS Identity and Access Management를 사용하면 조직에서 보안 그룹 및 네트워크 ACL과 흐름 로그를 생성하고 관리할 수 있는 권한을 가진 사람을 통제할 수 있습니다. 예를 들어 이러한 권한을 네트워크 관리자에게만 부여하고, 인스턴스만 시작하면 되는 사용자에게는 부여하지 않을 수 있습니다. 자세한 내용은 [Amazon VPC 리소스에 대한 액세스 제어 \(p. 130\)](#) 단원을 참조하십시오.

Amazon 보안 그룹과 네트워크 ACL은 링크-로컬 주소(169.254.0.0/16) 또는 AWS에서 예약한 IPv4 주소(VPC에 대한 Amazon DNS 서버 주소가 포함된 서브넷의 첫 IPv4 주소 4개)를 주고받는 트래픽을 필터링하지 않습니다. 마찬가지로 흐름 로그는 이러한 주소에서 송수신되는 IP 트래픽을 수집하지 않습니다. 이들 주소는 DNS(Domain Name Service), DHCP(Dynamic Host Configuration Protocol), Amazon EC2 인스턴스 메타데이터, KMS(Key Management Server - Windows 인스턴스용 라이선스 관리) 및 서브넷에서 라우팅 등의 서비스를 지원합니다. 인스턴스에서 추가 방화벽 솔루션을 구현하여 링크-로컬 주소와의 네트워크 통신을 차단할 수 있습니다.

## 보안 그룹 및 네트워크 ACL 비교

다음 표는 보안 그룹과 네트워크 ACL의 근본적인 차이를 요약한 것입니다.

보안 그룹	네트워크 ACL
인스턴스 수준에서의 적용(1차 보안 계층)	서브넷 수준에서의 적용(2차 보안 계층)
허용 규칙만 지원	허용 및 거부 규칙 지원
상태 저장: 규칙에 관계없이 반환 트래픽이 자동으로 허용됨	상태 비저장: 반환 트래픽이 규칙에 의해 명시적으로 허용되어야 함
트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가함	트래픽 허용 여부를 결정 시 규칙을 번호순으로 처리함
인스턴스 시작 시 누군가 보안 그룹을 지정하거나, 나중에 보안 그룹을 인스턴스와 연결하는 경우에만 인스턴스에 적용됨	연결된 서브넷에서 모든 인스턴스에 자동 적용됨(백업 보안 계층이므로 보안 그룹을 지정하는 사람에게 의존할 필요 없음)

다음 다이어그램은 보안 그룹과 네트워크 ACL에서 제공하는 보안 계층을 보여 줍니다. 예를 들어, 인터넷 게이트웨이의 트래픽은 라우팅 테이블의 라우팅을 사용하여 적절한 서브넷에 라우팅됩니다. 서브넷과 연결된 네트워크 ACL 규칙은 서브넷에 허용되는 트래픽 유형을 제어합니다. 인스턴스와 연결된 보안 그룹 규칙은 인스턴스에 허용되는 트래픽 유형을 제어합니다.

## VPC의 보안 그룹

보안 그룹은 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다. VPC에서 인스턴스를 시작할 때 최대 5개의 보안 그룹에 인스턴스를 할당할 수 있습니다. 보안 그룹은 서브넷 수준이 아니라 인스턴스 수준에서 작동하므로 VPC에 있는 서브넷의 각 인스턴스를 서로 다른 보안 그룹 세트에 할당할 수 있습니다. 시작할 때 특정 그룹을 지정하지 않으면 인스턴스가 자동으로 VPC의 기본 보안 그룹에 할당됩니다.

각 보안 그룹에 대해 인스턴스에 대한 인바운드 트래픽을 제어하는 규칙과 아웃바운드 트래픽을 제어하는 별도의 규칙 세트를 추가합니다. 이 단원에서는 VPC의 보안 그룹과 그 규칙에 대해 알아야 하는 기본 사항을 설명합니다.

보안 그룹과 비슷한 규칙으로 네트워크 ACL을 설정하여 VPC에 보안 계층을 더 추가할 수 있습니다. 보안 그룹과 네트워크 ACL의 차이에 대한 자세한 정보는 [보안 그룹 및 네트워크 ACL 비교 \(p. 95\)](#) 단원을 참조하십시오.

### 항목

- [보안 그룹 기본 사항 \(p. 95\)](#)
- [VPC의 기본 보안 그룹 \(p. 96\)](#)
- [보안 그룹 규칙 \(p. 97\)](#)
- [EC2-Classic 및 EC2-VPC용 보안 그룹의 차이점 \(p. 98\)](#)
- [보안 그룹 작업 \(p. 99\)](#)
- [API 및 CLI 개요 \(p. 102\)](#)

## 보안 그룹 기본 사항

다음은 VPC의 보안 그룹이 갖는 기본 특징입니다.

- VPC당 생성할 수 있는 보안 그룹의 개수, 각 보안 그룹에 추가할 수 있는 규칙의 개수, 그리고 네트워크 인터페이스에 연결할 수 있는 보안 그룹의 개수에는 제한이 있습니다. 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.
- 허용 규칙을 지정할 수 있지만 거부 규칙은 지정할 수 없습니다.
- 인바운드 트래픽과 아웃바운드 트래픽에 별도의 규칙을 지정할 수 있습니다.
- 보안 그룹을 만드는 경우에는 인바운드 규칙이 없습니다. 따라서 보안 그룹에 인바운드 규칙을 추가하기 전에는 인바운드 트래픽이 허용되지 않습니다.
- 기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용하는 아웃바운드 규칙을 포함합니다. 규칙을 제거할 수 있으며 특정 아웃바운드 트래픽만 허용하는 아웃바운드 규칙을 추가할 수 있습니다. 보안 그룹에 아웃바운드 규칙이 없는 경우 아웃바운드 트래픽이 허용되지 않습니다.
- 보안 그룹은 상태가 저장됩니다. — 사용자가 인스턴스에서 요청을 전송하면 해당 요청의 응답 트래픽은 인바운드 보안 그룹 규칙에 관계없이 인바운드 흐름이 허용됩니다. 아웃바운드 규칙에 상관없이, 허용된 인바운드 트래픽에 대한 반응으로 외부로 나가는 흐름이 수행됩니다.

**Note**

일부 트래픽 유형은 다른 유형까지 별도로 추적됩니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [연결 추적](#)을 참조하십시오.

- 해당 규칙을 추가하지 않으면 보안 그룹과 연결된 인스턴스가 서로 통신할 수 없습니다(예외: 기본 보안 그룹에는 기본적으로 이 규칙이 포함됨).
- 보안 그룹은 네트워크 인터페이스와 연결됩니다. 인스턴스를 시작한 후 인스턴스에 연결된 보안 그룹을 변경할 수 있습니다. 이 경우 주 네트워크 인터페이스(eth0)에 연결된 보안 그룹이 변경됩니다. 다른 네트워크 인터페이스와 연결된 보안 그룹을 변경할 수도 있습니다. 네트워크 인터페이스에 대한 자세한 정보는 [ENI](#) 단원을 참조하십시오.

## VPC의 기본 보안 그룹

VPC는 자동으로 기본 보안 그룹과 함께 제공됩니다. 인스턴스를 시작할 때 다른 보안 그룹을 지정하지 않으면 VPC에서 시작하는 각 EC2 인스턴스가 기본 보안 그룹과 자동으로 연결됩니다.

다음 표에서는 기본 보안 그룹의 기본 규칙을 설명합니다.

인바운드			
소스	프로토콜	포트 범위	설명
보안 그룹 ID(sg-xxxxxxx)	모두	모두	동일한 보안 그룹에 지정된 인스턴스의 인바운드 트래픽 허용.
아웃바운드			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 아웃바운드 IPv4 트래픽을 허용
:::0	모두	모두	모든 아웃바운드 IPv6 트래픽을 허용. 이 규칙은 IPv6 CIDR 블록이 있는 VPC를 생성하거나 IPv6 CIDR 블록을 기존 VPC와 연결하는 경우에 추가되도록 기본 설정되어 있습니다.

기본 보안 그룹에 대한 규칙을 변경할 수 있습니다.

기본 보안 그룹을 삭제할 수 없습니다. 기본 보안 그룹을 삭제하려고 하면 다음 `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user` 오류가 표시됩니다.

Note

보안 그룹의 아웃바운드 규칙을 수정한 경우에는 IPv6 블록을 VPC에 연결할 때 IPv6 트래픽에 대한 아웃바운드 규칙이 자동으로 추가되지 않습니다.

## 보안 그룹 규칙

보안 그룹의 규칙을 추가하거나 제거할 수 있습니다(인바운드 또는 아웃바운드 액세스 권한 부여 또는 취소라고도 함). 규칙은 인바운드 트래픽(수신)이나 아웃바운드 트래픽(송신)에 적용됩니다. 특정 CIDR 범위 또는 VPC나 피어 VPC(VPC 피어링 연결 필요)의 다른 보안 그룹에 대한 액세스 권한을 부여할 수 있습니다.

다음은 VPC의 보안 그룹 규칙 중 기본적인 부분입니다.

- (인바운드 규칙만 해당) 트래픽의 원본과 대상 포트 또는 포트 범위. 원본은 다른 보안 그룹, IPv4/IPv6 CIDR 블록 또는 단일 IPv4/IPv6 주소일 수 있습니다.
- (아웃바운드 규칙만 해당) 트래픽의 대상과 대상 포트 또는 포트 범위. 대상은 다른 보안 그룹, IPv4/IPv6 CIDR 블록 또는 단일 IPv4/IPv6 주소일 수 있습니다.
- 표준 프로토콜 번호를 가진 모든 프로토콜(목록은 [Protocol Numbers](#) 참조) ICMP를 프로토콜로 지정하면 ICMP 유형과 코드 중 일부 또는 전부를 지정할 수 있습니다.

보안 그룹을 규칙의 원본으로 지정할 경우 원본 보안 그룹과 연결된 인스턴스가 보안 그룹의 인스턴스에 액세스할 수 있습니다. 이 보안 그룹에 원본 보안 그룹의 규칙이 추가되지는 않습니다.

IPv4 주소를 한 개만 지정하는 경우, /32 접두사를 사용하십시오. IPv6 주소를 한 개만 지정하는 경우, /128 접두사 길이를 지정하십시오.

방화벽 설정을 위한 일부 시스템에서는 원본 포트를 필터링할 수 있습니다. 보안 그룹을 사용하면 대상 포트만 필터링할 수 있습니다.

규칙을 추가하거나 제거할 때 보안 그룹과 연결된 모든 인스턴스에 자동으로 적용됩니다.

추가하는 규칙의 종류는 인스턴스의 용도에 따라 다를 수 있습니다. 다음 표에서는 웹 서버의 보안 그룹에 대한 규칙의 예를 설명합니다. 웹 서버는 IPv4 및 IPv6 주소로부터 HTTP 및 HTTPS 트래픽을 수신하고, SQL 또는 MySQL 트래픽을 데이터베이스 서버에 전송할 수 있습니다.

인바운드			
소스	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	모든 IPv4 주소에서 이루어지는 인바운드 HTTP 액세스를 허용
:::0	TCP	80	모든 IPv6 주소에서 이루어지는 인바운드 HTTP 액세스를 허용
0.0.0.0/0	TCP	443	모든 IPv4 주소에서 이루어지는 인바운드 HTTPS 액세스 허용
:::0	TCP	443	모든 IPv6 주소에서 이루어지는 인바운드 HTTPS 액세스 허용
네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	네트워크의 IPv4 IP 주소로부터 Linux 인스턴스로 접근하는 인바운드 SSH 액세스 허용(인터넷 게이트웨이 경우)
네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	네트워크의 IPv4 IP 주소로부터 Windows 인스턴스로 접근하는 인바운드 RDP 액세스 허용(인터넷 게이트웨이 경우)

아웃바운드			
목적지	프로토콜	포트 범위	설명
데이터베이스 서버용 보안 그룹의 ID	TCP	1433	지정된 보안 그룹의 인스턴스로 아웃바운드 Microsoft SQL Server 액세스 허용
MySQL 데이터베이스 서버용 보안 그룹의 ID	TCP	3306	지정된 보안 그룹의 인스턴스로 아웃바운드 MySQL 액세스 허용

데이터베이스 서버에는 서로 다른 규칙이 필요할 수 있습니다. 예를 들어 인바운드 HTTP 및 HTTPS 트래픽 대신 인바운드 MySQL이나 Microsoft SQL Server 액세스를 허용하는 규칙을 추가할 수 있습니다. 웹 서버 및 데이터베이스 서버의 보안 그룹 규칙에 대한 예는 [보안 \(p. 37\)](#)을 참조하십시오.

경로 MTU 검색이 올바르게 작동할 수 있도록 보안 그룹 규칙을 생성하는 방법에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [경로 MTU 검색](#)을 참조하십시오.

## 무효 보안 그룹 규칙

VPC에 다른 VPC와의 VPC 피어링 연결이 있는 경우, 보안 그룹 규칙은 피어 VPC의 다른 보안 그룹을 참조할 수 있습니다. 이를 통해 참조된 보안 그룹과 연결된 인스턴스는 참조하는 보안 그룹과 연결된 인스턴스와 통신할 수 있습니다.

피어 VPC의 소유자가 참조된 보안 그룹을 삭제하거나 피어 VPC의 소유자가 VPC 피어링 연결을 삭제하면, 보안 그룹 규칙은 `stale`로 표시됩니다. 다른 보안 그룹 규칙과 같은 방법으로 무효 보안 그룹 규칙을 삭제할 수 있습니다.

자세한 내용은 Amazon VPC Peering Guide의 [Working With Stale Security Groups](#) 단원을 참조하십시오.

## EC2-Classic 및 EC2-VPC용 보안 그룹의 차이점

이미 Amazon EC2를 사용하고 있다면 보안 그룹에 익숙하겠지만, EC2-Classic에서 사용하기 위해 생성한 보안 그룹을 VPC의 인스턴스에 사용할 수는 없습니다. VPC 인스턴스 전용으로 보안 그룹을 생성해야 합니다. VPC용 보안 그룹에서 사용하기 위해 생성한 규칙은 EC2-Classic용 보안 그룹을 참조할 수 없으며 그 반대의 경우도 마찬가지입니다.

다음 표에는 EC2-Classic에 사용하는 보안 그룹과 EC2-VPC에 사용하는 보안 그룹의 차이가 요약되어 있습니다.

EC2-Classic	EC2-VPC
리전마다 최대 500개의 보안 그룹을 생성할 수 있습니다.	VPC당 최대 500개의 보안 그룹을 만들 수 있습니다.
보안 그룹당 최대 100개의 규칙을 추가할 수 있습니다.	보안 그룹당 최대 50개의 규칙을 추가할 수 있습니다.
인바운드 트래픽에만 규칙을 추가할 수 있습니다.	인바운드 및 아웃바운드 모두에 규칙을 지정할 수 있습니다.
인스턴스당 최대 500개의 보안 그룹을 할당할 수 있습니다.	최대 5개의 보안 그룹을 네트워크 인터페이스에 할당할 수 있습니다.
다른 AWS 계정에서 보안 그룹을 참조할 수 있습니다.	VPC 또는 VPC 피어링 연결의 피어 VPC에서만 보안 그룹을 참조할 수 있습니다. 피어 VPC는 다른 계정에 있을 수 있습니다.



EC2-Classic	EC2-VPC
인스턴스를 시작한 후에는 할당된 보안 그룹을 변경할 수 없습니다.	인스턴스를 시작한 후 인스턴스에 할당된 보안 그룹을 변경할 수 있습니다.
보안 그룹에 규칙을 추가할 때 프로토콜을 지정할 필요가 없으며 TCP, UDP 또는 ICMP만 사용할 수 있습니다.	보안 그룹에 규칙을 추가할 때 프로토콜을 지정해야 하며 이 프로토콜은 표준 프로토콜 번호를 가진 프로토콜 또는 모든 프로토콜이 될 수 있습니다 ( <a href="#">Protocol Numbers</a> 참조).
보안 그룹에 규칙을 추가할 때 포트 번호를 지정해야 합니다(TCP 또는 UDP용).	보안 그룹에 규칙을 추가할 때 규칙이 TCP용이거나 UDP용일 경우에만 포트 번호를 지정할 수 있으며 모든 포트 번호를 지정할 수 있습니다.
다른 보안 그룹의 규칙에서 참조된 보안 그룹은 삭제할 수 없습니다.	보안 그룹이 서로 다른 VPC에 있는 경우, 다른 보안 그룹의 규칙에서 참조된 보안 그룹을 삭제할 수 있습니다. 참조된 보안 그룹을 삭제하면, 규칙은 무효로 표시됩니다. <a href="#">describe-stale-security-groups</a> AWS CLI 명령을 사용하여 무효 규칙을 식별할 수 있습니다.
IPv6 CIDR 블록 또는 IPv6 주소를 보안 그룹 규칙의 원본 또는 대상으로 지정할 수 없습니다.	IPv6 CIDR 블록 또는 IPv6 주소를 보안 그룹 규칙의 원본 또는 대상으로 지정할 수 있습니다.

## 보안 그룹 작업

이 단원에서는 Amazon VPC 콘솔을 사용하여 보안 그룹에 대한 작업을 수행하는 방법을 보여줍니다.

### 항목

- [기본 보안 그룹 수정 \(p. 99\)](#)
- [보안 그룹 생성 \(p. 99\)](#)
- [규칙 추가 및 제거 \(p. 100\)](#)
- [인스턴스의 보안 그룹 변경 \(p. 100\)](#)
- [보안 그룹 삭제 \(p. 101\)](#)
- [2009-07-15-default 보안 그룹 삭제 \(p. 101\)](#)

## 기본 보안 그룹 수정

VPC에는 초기 규칙이 인바운드 트래픽을 모두 거부하고, 아웃바운드 트래픽을 모두 허용하며, 그룹의 인스턴스 간 모든 트래픽을 허용하는 기본 보안 그룹이 있습니다. 이 그룹을 삭제할 수 없지만 그룹의 규칙을 변경할 수는 있습니다. 다른 보안 그룹을 수정할 때와 절차는 똑같습니다. 자세한 내용은 [규칙 추가 및 제거 \(p. 100\)](#) 단원을 참조하십시오.

## 보안 그룹 생성

인스턴스의 기본 보안 그룹을 사용할 수 있지만 직접 그룹을 생성하여 시스템에서 인스턴스에 지정되는 다른 역할을 반영할 수 있습니다.

### 보안 그룹을 생성하는 방법

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]을 선택합니다.

4. 보안 그룹 이름을 입력하고(예: `my-security-group`) 설명을 제공합니다. [VPC] 메뉴에서 VPC의 ID를 선택한 다음 [Yes, Create]를 선택합니다.

기본적으로 처음에 새 보안 그룹에는 인스턴스에서 나가는 모든 트래픽을 허용하는 아웃바운드 규칙만 적용됩니다. 인바운드 트래픽을 사용하거나 아웃바운드 트래픽을 제한하려면 규칙을 추가해야 합니다.

## 규칙 추가 및 제거

규칙을 추가하거나 제거할 때 이미 보안 그룹에 할당된 인스턴스가 변경될 수 있습니다. Amazon EC2 API 또는 명령줄 도구를 사용하면 규칙을 수정할 수 없으며 규칙을 추가하고 삭제할 수만 있습니다. Amazon VPC 콘솔을 사용하면 기존 규칙의 항목을 수정할 수 있습니다. 규칙이 제거되고 새 규칙이 추가됩니다.

### Note

VPC 피어링 연결이 있는 경우, 피어 VPC의 보안 그룹을 보안 그룹 규칙의 소스 또는 대상으로 참조할 수 있습니다. 자세한 내용은 Amazon VPC Peering Guide의 [피어링된 VPC 보안 그룹을 참조하도록 보안 그룹 업데이트](#)를 참조하십시오.

### 규칙을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 업데이트할 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 규칙 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
4. Inbound Rules 탭에서 [Edit]를 선택합니다. [Type]에 대해 인바운드 트래픽 규칙에 대한 옵션을 선택한 후 필요한 정보를 입력합니다. 예를 들어, 퍼블릭 웹 서버에 대해 [HTTP] 또는 [HTTPS]를 선택하고 [Source] 값을 `0.0.0.0/0`으로 지정합니다. [Save]를 선택합니다.

### Note

- `0.0.0.0/0`을 사용하면 모든 IPv4 주소가 HTTP 또는 HTTPS를 사용하여 인스턴스에 액세스하도록 할 수 있습니다. 액세스를 제한하려면 특정 IP 주소 또는 주소 범위를 입력합니다.
5. 이 보안 그룹과 연결된 모든 인스턴스 간의 통신을 허용할 수도 있습니다. [Inbound Rules] 탭의 [Type] 목록에서 [All Traffic]를 선택합니다. [Source]에 보안 그룹 ID를 입력합니다. 입력을 시작하면 보안 그룹 목록이 제공됩니다. 목록에서 보안 그룹을 선택한 후 [Save]를 선택합니다.
  6. 필요에 따라 [Outbound Rules] 탭을 사용하여 아웃바운드 트래픽 규칙을 추가할 수도 있습니다.

### 규칙을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 업데이트할 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 규칙 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
4. [Edit]를 선택하고 삭제할 역할을 선택한 다음, [Remove], [Save]를 선택합니다.

## 인스턴스의 보안 그룹 변경

인스턴스가 시작된 후 VPC의 인스턴스가 할당된 보안 그룹을 변경할 수 있습니다. 인스턴스가 실행 중이거나 중단되었을 때 이 보안 그룹을 변경할 수 있습니다.

### Note

이 절차에서는 인스턴스의 기본 네트워크 인터페이스(eth0)와 연결된 보안 그룹이 변경됩니다. 다른 네트워크 인터페이스의 보안 그룹을 변경하려면 [네트워크 인터페이스의 보안 그룹 변경](#)을 참조하십시오.

인스턴스의 보안 그룹을 변경하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 인스턴스에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 [Networking], [Change Security Groups]를 선택합니다.
4. [Change Security Groups] 대화 상자의 목록에서 하나 이상의 보안 그룹을 선택한 후 [Assign Security Groups]를 선택합니다.

## 보안 그룹 삭제

보안 그룹에 할당된 인스턴스가 없는 경우에만(실행 중이거나 중단됨) 보안 그룹을 삭제할 수 있습니다. 보안 그룹을 삭제하기 전에 다른 보안 그룹에 인스턴스를 할당할 수 있습니다([인스턴스의 보안 그룹 변경](#) (p. 100) 단원 참조). 기본 보안 그룹을 삭제할 수 없습니다.

보안 그룹을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 보안 그룹을 선택한 다음 [Security Group Actions], [Delete Security Group]을 선택합니다.
4. [Delete Security Group] 대화 상자에서 [Yes, Delete]를 선택합니다.

## 2009-07-15-default 보안 그룹 삭제

2011-01-01 이전의 API 버전을 사용하여 생성된 VPC에는 2009-07-15-default 보안 그룹이 있습니다. 이 보안 그룹 외에도 모든 VPC와 함께 제공되는 일반 default 보안 그룹이 있습니다. 2009-07-15-default 보안 그룹이 있는 VPC에는 인터넷 게이트웨이를 연결할 수 없으므로 VPC에 인터넷 게이트웨이를 연결하려면 먼저 이 보안 그룹을 삭제해야 합니다.

### Note

이 보안 그룹을 인스턴스에 할당한 경우 이 인스턴스를 다른 보안 그룹에 할당해야 보안 그룹을 삭제할 수 있습니다.

2009-07-15-default 보안 그룹을 삭제하려면

1. 이 보안 그룹이 인스턴스에 할당되지 않았는지 확인합니다.
  - a. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
  - b. 탐색 창에서 Network Interfaces를 선택합니다.
  - c. 목록에서 인스턴스에 대한 네트워크 인터페이스를 선택한 후 [Change Security Groups], [Actions]를 선택합니다.
  - d. [Change Security Groups] 대화 상자의 목록에서 새 보안 그룹을 선택한 후 [Save]를 선택합니다.

### Tip

인스턴스의 보안 그룹을 변경할 때 목록에서 여러 그룹을 선택할 수 있습니다. 선택한 보안 그룹이 인스턴스의 현재 보안 그룹을 대체합니다.

- e. 인스턴스마다 앞의 단계를 반복합니다.
2. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
  3. 탐색 창에서 [Security Groups]를 선택합니다.
  4. 2009-07-15-default 보안 그룹을 선택한 다음 [Security Group Actions], [Delete]을 선택합니다.

5. [Delete Security Group] 대화 상자에서 [Yes, Delete]를 선택합니다.

## API 및 CLI 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 섹션을 참조하십시오.

### 보안 그룹 생성

- [create-security-group](#)(AWS CLI)
- [New-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

### 보안 그룹에 규칙 추가

- [authorize-security-group-ingress](#) 및 [authorize-security-group-egress](#)(AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) 및 [Grant-EC2SecurityGroupEgress](#)(Windows PowerShell용 AWS 도구)

### 하나 이상의 보안 그룹 설명

- [describe-security-groups](#)(AWS CLI)
- [Get-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

### 인스턴스에 대한 보안 그룹 수정

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (Windows PowerShell용 AWS 도구)

### 보안 그룹에서 규칙 제거

- [revoke-security-group-ingress](#) 및 [revoke-security-group-egress](#)(AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) 및 [Revoke-EC2SecurityGroupEgress](#)(Windows PowerShell용 AWS 도구)

### 보안 그룹 삭제

- [delete-security-group](#)(AWS CLI)
- [Remove-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

## 네트워크 ACL

네트워크 ACL(액세스 제어 목록)은 1개 이상의 서브넷 내부와 외부의 트래픽을 제어하기 위한 방화벽 역할을 하는 VPC를 위한 선택적 보안 계층입니다. 보안 그룹과 비슷한 규칙으로 네트워크 ACL을 설정하여 VPC에 보안 계층을 더 추가할 수 있습니다. 보안 그룹과 네트워크 ACL의 차이에 대한 자세한 정보는 [보안 그룹 및 네트워크 ACL 비교 \(p. 95\)](#) 단원을 참조하십시오.

### 항목

- [네트워크 ACL 기본 사항 \(p. 103\)](#)
- [네트워크 ACL 규칙 \(p. 103\)](#)
- [기본 네트워크 ACL \(p. 103\)](#)

- 사용자 지정 네트워크 ACL (p. 105)
- 휘발성 포트 (p. 109)
- 네트워크 ACL 작업 (p. 109)
- 예: 서브넷 내 인스턴스에 대한 액세스 제어 (p. 112)
- API 및 명령 개요 (p. 114)

## 네트워크 ACL 기본 사항

다음은 네트워크 ACL에 대해 알아야 할 기본 사항입니다.

- VPC는 수정 가능한 기본 네트워크 ACL과 함께 자동으로 제공됩니다. 기본적으로 모든 인바운드 및 아웃바운드 IPv4 트래픽을 허용하며, 해당되는 경우 IPv6 트래픽도 허용합니다.
- 사용자 지정 네트워크 ACL을 생성하여 서브넷과 연결할 수 있습니다. 기본적으로 각 사용자 지정 네트워크 ACL은 규칙을 추가하기 전에는 모든 인바운드 및 아웃바운드 트래픽을 거부합니다.
- VPC에 있는 각 서브넷을 네트워크 ACL과 연결해야 합니다. 서브넷을 네트워크 ACL에 명시적으로 연결하지 않을 경우, 서브넷은 기본 네트워크 ACL에 자동적으로 연결됩니다.
- 한 네트워크 ACL을 여러 서브넷과 연결할 수 있습니다. 하지만 한 서브넷은 한 번에 한 네트워크 ACL과만 연결할 수 있습니다. 네트워크 ACL을 서브넷과 연결하면 이전 연결은 제거됩니다.
- 네트워크 ACL은 번호가 매겨진 규칙 목록을 포함하고 있는데, 가장 낮은 번호를 매긴 규칙부터 시작해 이 규칙들을 평가함으로써 네트워크 ACL과 연결된 모든 서브넷의 내부 또는 외부로 트래픽이 허용되는지 확인할 수 있습니다. 규칙에 사용할 수 있는 가장 높은 번호는 32766입니다. 나중에 필요한 곳에 새 규칙을 삽입할 수 있도록, 처음 시작할 때는 규칙 번호를 100의 배수로 지정하여 규칙을 생성하는 것이 좋습니다.
- 네트워크 ACL에는 별개의 인바운드 및 아웃바운드 규칙이 있으며, 각 규칙은 트래픽을 허용하거나 거부할 수 있습니다.
- 네트워크 ACL은 상태 비저장입니다. 즉, 허용되는 인바운드 트래픽에 대한 응답은 아웃바운드 트래픽에 대한 규칙을 따르고, 그 반대의 경우에도 마찬가지입니다.

생성 가능한 네트워크 ACL 개수에 대한 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

## 네트워크 ACL 규칙

기본 네트워크 ACL에 규칙을 추가 또는 제거하거나, VPC에 대한 네트워크 ACL을 추가로 생성할 수 있습니다. 네트워크 ACL에 규칙을 추가하거나 제거할 때 네트워크 ACL이 연결되어 있는 서브넷에 변경 사항이 자동으로 적용됩니다.

다음은 네트워크 ACL 규칙 중 일부입니다.

- 규칙 번호. 번호가 가장 낮은 규칙부터 평가됩니다. 규칙에 일치하는 트래픽이 있으면 이와 모순되는 상위 규칙이 있더라도 적용됩니다.
- 프로토콜. 표준 프로토콜 번호를 가진 어떤 프로토콜이든 지정할 수 있습니다. 자세한 내용은 [프로토콜 번호](#)를 참조하십시오. ICMP를 프로토콜로 지정하면 ICMP 유형과 코드 중 일부 또는 전부를 지정할 수 있습니다.
- [인바운드 규칙만 해당] 트래픽의 원본(CIDR 범위)과 대상(수신 대기) 포트 또는 포트 범위.
- [아웃바운드 규칙만 해당] 트래픽의 대상(CIDR 범위)과 대상 포트 또는 포트 범위.
- 지정된 트래픽에 대한 ALLOW 또는 DENY 선택.

## 기본 네트워크 ACL

기본 네트워크 ACL은 연결된 서브넷을 드나드는 트래픽 흐름을 모두 허용하도록 구성되어 있습니다. 각 네트워크 ACL에는 규칙 번호가 별표로 되어 있는 규칙도 포함되어 있습니다. 이 규칙은 패킷이 번호가 매겨진

다른 어떤 규칙과도 일치하지 않을 경우에는 거부되도록 되어 있습니다. 이 규칙을 수정하거나 제거할 수 없습니다.

다음은 IPv4만을 지원하는 VPC에 대한 기본 네트워크 ACL의 예시입니다.

인바운드					
규칙 #	Type	프로토콜	포트 범위	소스	허용/거부
100	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	ALLOW
*	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	DENY
아웃바운드					
규칙 #	Type	프로토콜	포트 범위	목적지	허용/거부
100	모든 IPv4 트래픽	all	all	0.0.0.0/0	ALLOW
*	모든 IPv4 트래픽	all	all	0.0.0.0/0	DENY

IPv6 CIDR 블록이 있는 VPC를 생성하거나 IPv6 CIDR 블록을 기존 VPC와 연결하는 경우, 모든 IPv6 트래픽이 서브넷으로 그리고 서브넷으로부터 전송되도록 허용하는 규칙이 자동으로 추가됩니다. 또한 규칙 번호가 별도로 되어 있어 패킷이 번호가 매겨진 다른 어떤 규칙과도 일치하지 않을 경우에는 거부되도록 하는 규칙도 추가합니다. 이 규칙은 수정하거나 제거할 수 없습니다. 다음은 IPv4와 IPv6를 지원하는 VPC에 대한 기본 네트워크 ACL의 예시입니다.

**Note**

기본 네트워크 ACL의 인바운드 규칙을 수정한 경우에는, IPv6 블록을 VPC에 연결할 때 인바운드 IPv6 트래픽에 대한 허용 규칙이 자동으로 추가되지는 않습니다. 이와 마찬가지로 아웃바운드 규칙을 수정한 경우에는 아웃바운드 IPv6 트래픽에 대한 허용 규칙이 자동으로 추가되지 않습니다.

인바운드					
규칙 #	Type	프로토콜	포트 범위	소스	허용/거부
100	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	ALLOW
101	모든 IPv6 트래픽	모두	모두	::/0	허용
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY
*	모든 IPv6 트래픽	모두	모두	::/0	DENY
아웃바운드					
규칙 #	Type	프로토콜	포트 범위	목적지	허용/거부
100	모든 트래픽	all	all	0.0.0.0/0	ALLOW
101	모든 IPv6 트래픽	all	all	::/0	허용

*	모든 트래픽	all	all	0.0.0.0/0	DENY
*	모든 IPv6 트래픽	all	all	:::/0	DENY

## 사용자 지정 네트워크 ACL

다음 표는 IPv4만을 지원하는 VPC에 대한 사용자 지정 네트워크 ACL의 예시입니다. 여기에는 HTTP 및 HTTPS 트래픽 수신을 허용하는 규칙이 포함됩니다(인바운드 규칙 100 및 110). 그 인바운드 트래픽에 응답할 수 있도록 해당 아웃바운드 규칙이 있습니다(취발성 포트 49152-65535를 포함하는 아웃바운드 규칙 120). 적절한 취발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 [취발성 포트 \(p. 109\)](#) 단원을 참조하십시오.

이 네트워크 ACL에는 서브넷으로의 SSH 및 RDP 트래픽을 허용하는 인바운드 규칙도 포함됩니다. 아웃바운드 규칙 120을 사용하면 응답을 서브넷에서 송신할 수 있습니다.

이 네트워크 ACL에는 서브넷 외부로의 아웃바운드 HTTP 및 HTTPS 트래픽을 허용하는 아웃바운드 규칙(100 및 110)이 있습니다. 그 아웃바운드 트래픽에 응답할 수 있도록 해당 인바운드 규칙이 있습니다(취발성 포트 49152-65535를 포함하는 인바운드 규칙 140).

### Note

각 네트워크 ACL에는 규칙 번호가 별표로 된 기본 규칙이 포함되어 있습니다. 이 규칙은 패킷이 다른 어떤 규칙과도 일치하지 않을 경우에는 거부되도록 되어 있습니다. 이 규칙을 수정하거나 제거할 수 없습니다.

인바운드						
규칙 #	Type	프로토콜	포트 범위	소스	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	SSH	TCP	22	192.0.2.0/24	ALLOW	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터의 인바운드 SSH 트래픽을 허용(인터넷 게이트웨이를 통해)
130	RDP	TCP	3389	192.0.2.0/24	ALLOW	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터 웹 서버로의 인바운드 RDP 트래픽을 허용(인터넷 게이트웨이를 통해)
140	사용자 지정 TCP	TCP	49152 - 65535	0.0.0.0/0	ALLOW	인터넷으로부터의 인바운드 리턴 IPv4 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 적절한 취발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">취발성 포</a>

						트 (p. 109) 단원을 참조하십시오.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
아웃바운드						
규칙 #	Type	프로토콜	포트 범위	목적지	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTP 트래픽을 허용
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTPS 트래픽을 허용
120	사용자 지정 TCP	TCP	49152 - 65535	0.0.0.0/0	ALLOW	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">휘발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)

패킷이 서브넷에 도착할 때, 이를 서브넷이 연결되어 있는 ACL의 수신 규칙에 대해 평가합니다(규칙 목록의 맨 위에서 시작하여 맨 아래로 이동). 다음은 패킷이 SSL 포트(443)로 전달되는 경우 평가가 이루어지는 방식입니다. 이 패킷은 처음으로 평가되는 규칙(규칙 100)과 일치하지 않습니다. 이 패킷은 패킷을 서브넷으로 수신되도록 허용하는 두 번째 규칙(110)과도 일치하지 않습니다. 패킷이 포트 139(NetBIOS)로 보내진 경우, 첫 두 개의 규칙은 일치하지 않았지만 \* 규칙이 최종적으로 해당 패킷을 거부했을 것입니다.

정당한 방법으로 광범위한 포트를 열 필요가 있는 상황에서는 DENY 규칙을 추가하고 싶을지 모르지만, 거부하려는 범위 내에 특정 포트가 있습니다. 단, 표에서 광범위한 포트 트래픽을 허용하는 규칙보다 앞서 DENY 규칙을 배치해야 합니다.

#### Important

Elastic Load Balancing를 사용하는 경우, 사용자의 백엔드 인스턴스에 대한 서브넷에 소스가 0.0.0.0/0 또는 서브넷의 CIDR인 모든 트래픽에 대해 DENY 규칙을 추가한 네트워크 ACL이 있으면 로드 밸런서가 인스턴스에 대한 상태 확인을 수행할 수 없습니다. 로드 밸런서 및 백엔드 인스턴스에 권장되는 네트워크 ACL 규칙에 대한 자세한 내용은 Classic Load Balancer 가이드에서 [Network ACLs for Load Balancers in a VPC](#) 단원을 참조하십시오.

다음 표는 연결된 IPv6 CIDR 블록이 있는 VPC에 대한 사용자 지정 네트워크 ACL의 동일 예시입니다. 이 네트워크 ACL에는 모든 IPv6 HTTP 및 HTTPS 트래픽에 대한 규칙이 포함됩니다. 이 경우, IPv4 트래픽에 대한 기존 규칙들 사이에 새 규칙이 삽입되었습니다. 그러나 IPv4 규칙에 따라 더 높은 번호 규칙을 추가할 수



도 있습니다. IPv4 및 IPv6 트래픽은 분리되어 있습니다. 따라서 IPv4 트래픽에 대한 규칙 중 어느 것도 IPv6 트래픽에 적용되지 않습니다.

인바운드						
규칙 #	Type	프로토콜	포트 범위	소스	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
105	HTTP	TCP	80	::/0	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
115	HTTPS	TCP	443	::/0	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	SSH	TCP	22	192.0.2.0/24	ALLOW	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터의 인바운드 SSH 트래픽을 허용(인터넷 게이트웨이를 통해)
130	RDP	TCP	3389	192.0.2.0/24	ALLOW	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터 웹 서버로의 인바운드 RDP 트래픽을 허용(인터넷 게이트웨이를 통해)
140	사용자 지정 TCP	TCP	49152 - 65535	0.0.0.0/0	ALLOW	인터넷으로부터의 인바운드 리턴 IPv4 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">휘발성 포트 (p. 109)</a> 단원을 참조하십시오.
145	사용자 지정 TCP	TCP	49152~65535::/0		허용	인터넷으로부터의 인바운드 리턴 IPv6 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">휘발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)

Amazon Virtual Private Cloud 사용 설명서  
사용자 지정 네트워크 ACL

*	모든 트래픽	모두	모두	::/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드						
규칙 #	Type	프로토콜	포트 범위	목적지	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTP 트래픽을 허용
105	HTTP	TCP	80	::/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv6 HTTP 트래픽을 허용
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTPS 트래픽을 허용
115	HTTPS	TCP	443	::/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv6 HTTPS 트래픽을 허용
120	사용자 지정 TCP	TCP	49152 - 65535	0.0.0.0/0	ALLOW	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">휘발성 포트 (p. 109)</a> 단원을 참조하십시오.
125	사용자 지정 TCP	TCP	49152~65535::/0		허용	인터넷에서 클라이언트에 대한 아웃바운드 IPv6 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 <a href="#">휘발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)
*	모든 트래픽	모두	모두	::/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

## 취발성 포트

이전 섹션에서 예로 든 네트워크 ACL에는 49152-65535 범위의 취발성 포트가 사용됩니다. 하지만 사용하거나 통신하는 클라이언트의 유형에 따라 다른 범위의 네트워크 ACL을 사용할 수 있습니다.

요청을 시작하는 클라이언트가 취발성 포트 범위를 선택합니다. 범위는 클라이언트의 운영 체제에 따라 다릅니다. 다수의 Linux 커널(Amazon Linux 커널 포함)이 포트 32768-61000을 사용합니다. Elastic Load Balancing에서 시작하는 요청은 포트 1024-65535를 사용합니다. Windows Server 2003까지의 Windows 운영 체제에서는 포트 1025-5000을 사용합니다. Windows Server 2008 이상 버전은 포트 49152-65535를 사용합니다. NAT 게이트웨이는 포트 1024 - 65535를 사용합니다. 예를 들어, 인터넷을 통해 Windows XP 클라이언트로부터 VPC에 있는 웹 서버로 요청이 수신되는 경우, 네트워크 ACL에는 포트 1025 - 5000으로 트래픽을 전달할 수 있도록 하는 아웃바운드 규칙이 있어야 합니다.

VPC의 인스턴스가 요청을 시작하는 클라이언트인 경우, 사용자의 네트워크 ACL에는 인스턴스의 유형(Amazon Linux, Windows Server 2008 등)에 특정한 취발성 포트로 트래픽을 전달할 수 있도록 하는 인바운드 규칙이 있어야 합니다.

실제로는 VPC에서 퍼블릭 쪽 인스턴스로 향하는 트래픽을 시작할 수도 있는 다양한 유형의 클라이언트를 포괄하기 위해, 취발성 포트 1024-65535를 열 수 있습니다. 하지만 그 범위 내에 있는 악성 포트의 트래픽을 거부하기 위한 규칙을 ACL에 추가할 수도 있습니다. 포에서 광범위한 취발성 포트를 여는 ALLOW 규칙보다 앞서 DENY 규칙을 배치해야 합니다.

## 네트워크 ACL 작업

이 섹션에서는 Amazon VPC 콘솔을 사용하여 네트워크 ACL로 작업하는 방법을 보여줍니다.

### 항목

- [네트워크 ACL 연결 확인 \(p. 109\)](#)
- [네트워크 ACL 생성 \(p. 110\)](#)
- [규칙 추가 및 삭제 \(p. 110\)](#)
- [서브넷을 네트워크 ACL과 연결 \(p. 111\)](#)
- [서브넷에서 네트워크 ACL 연결 끊기 \(p. 111\)](#)
- [서브넷의 네트워크 ACL 변경 \(p. 111\)](#)
- [네트워크 ACL 삭제 \(p. 112\)](#)

## 네트워크 ACL 연결 확인

Amazon VPC 콘솔을 사용하여 특정 서브넷과 연결되어 있는 네트워크 ACL을 확인할 수 있습니다. 네트워크 ACL은 복수의 서브넷과 연결될 수 있으며, 따라서 특정 네트워크 ACL과 연결되어 있는 서브넷을 확인할 수도 있습니다.

서브넷과 연결되어 있는 네트워크 ACL을 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Subnets]를 선택한 후 서브넷을 선택합니다.

서브넷과 연결된 네트워크 ACL은 네트워크 ACL의 규칙과 함께 [Network ACL] 탭에 포함되어 있습니다.

네트워크 ACL과 연결되어 있는 서브넷을 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택합니다. [Associated With] 열에 각 네트워크 ACL에 연결된 서브넷의 수가 표시됩니다.

3. 네트워크 ACL을 선택합니다.
4. 세부 정보 창에서 [Subnet Associations]를 선택하여 네트워크 ACL과 연결된 서브넷을 표시합니다.

## 네트워크 ACL 생성

VPC에 대한 사용자 지정 네트워크 ACL을 생성할 수 있습니다. 기본적으로, 생성된 네트워크 ACL은 사용자가 규칙을 추가할 때까지는 모든 인바운드 및 아웃바운드 트래픽을 차단하며, 사용자가 명시적으로 특정 서브넷과 연결할 때까지는 서브넷과 연결되지 않습니다.

네트워크 ACL을 생성하려면

1. <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택합니다.
3. [Create Network ACL]을 선택합니다.
4. 필요할 경우 [Create Network ACL] 대화 상자에서 네트워크 ACL의 이름을 지정한 다음, [VPC] 목록에서 VPC의 ID를 선택하고 [Yes, Create]를 선택합니다.

## 규칙 추가 및 삭제

ACL에 규칙을 추가하거나 삭제할 때, ACL과 연결된 서브넷은 어느 것이든 변경될 수 있습니다. 변경 사항은 잠시 후에 적용되므로 서브넷에서 인스턴스를 종료했다가 다시 시작할 필요는 없습니다.

Amazon EC2 API 또는 명령줄 도구를 사용하면 규칙을 수정할 수 없으며 규칙을 추가하고 삭제할 수만 있습니다. Amazon VPC 콘솔을 사용하면 기존 규칙의 항목을 수정할 수 있습니다. 규칙이 제거되고 새 규칙이 추가됩니다. ACL에서 규칙의 순서를 변경할 필요가 있는 경우에는 새 규칙 번호와 함께 새 규칙을 추가한 후에 원래 규칙은 삭제해야 합니다.

네트워크 ACL에 규칙을 추가하려면

1. <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택합니다.
3. 세부 정보 창에서 추가해야 할 규칙의 유형에 따라 [Inbound Rules] 또는 [Outbound Rules] 탭을 선택한 후 [Edit]를 선택합니다.
4. [Rule #]에서 규칙 번호를 입력합니다(예: 100). 규칙 번호가 네트워크 ACL에서 이미 사용되고 있는 번호이면 안 됩니다. 규칙은 가장 낮은 번호부터 시작해서 순서대로 처리됩니다.

### Tip

순차 번호(101, 102, 103)를 사용하는 대신 규칙 번호 간에 간격을 두는 것이 좋습니다(예: 100, 200, 300). 그러면 기존 규칙의 번호를 다시 매길 필요 없이 새 규칙을 더 쉽게 추가할 수 있습니다.

5. [Type] 목록에서 규칙을 선택합니다. 예를 들어 HTTP에 대한 규칙을 추가하려면 [HTTP]를 선택합니다. 모든 TCP 트래픽을 허용하는 규칙을 추가하려면 [All TCP]를 선택합니다. 이런 옵션 중 일부에 대해서는 (예: HTTP) 포트가 자동으로 입력됩니다. 나열되지 않은 프로토콜을 사용하려면 [Custom Protocol Rule]을 선택합니다.
6. (선택 사항) 사용자 지정 프로토콜 규칙을 생성할 경우 [Protocol] 목록에서 프로토콜의 번호와 이름을 선택합니다. 자세한 내용은 [프로토콜 번호의 IANA 목록](#)을 참조하십시오.
7. (선택 사항) 선택한 프로토콜에 포트 번호가 필요한 경우 해당 포트 번호를 입력하거나 하이픈으로 구분된 포트 범위(예: 49152-65535)를 입력합니다.
8. (인바운드 또는 아웃바운드 규칙인지에 따라) [Source] 또는 [Destination] 필드에 규칙이 적용되는 CIDR 범위를 입력합니다.
9. [Allow/Deny] 목록에서 지정된 트래픽을 허용하려면 [ALLOW]를, 지정된 트래픽을 거부하려면 [DENY]를 선택합니다.

10. (선택 사항) 또 다른 규칙을 추가하려면 [Add another rule]을 선택하고 필요에 따라 4~9단계를 반복합니다.
11. 마치면 [Save]를 선택합니다.

#### 네트워크 ACL에서 규칙을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택한 후 네트워크 ACL을 선택합니다.
3. 세부 정보 창에서 [Inbound Rules] 또는 [Outbound Rules] 탭을 선택한 후 [Edit]를 선택합니다. 삭제하려는 규칙에서 [Remove]를 선택한 후 [Save]를 선택합니다.

## 서브넷을 네트워크 ACL과 연결

특정 서브넷에 네트워크 ACL의 규칙을 적용하려면 서브넷을 네트워크 ACL과 연결해야 합니다. 한 네트워크 ACL을 여러 서브넷과 연결할 수 있지만, 한 서브넷은 한 네트워크 ACL과 연결할 수 있을 뿐입니다. 기본적으로, 특정 ACL과 연결되지 않은 서브넷은 기본 네트워크 ACL과 연결됩니다.

#### 서브넷을 네트워크 ACL과 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택한 후 네트워크 ACL을 선택합니다.
3. 세부 정보 창의 [Subnet Associations] 탭에서 [Edit]를 선택합니다. 네트워크 ACL과 연결할 서브넷에 대한 [Associate] 확인란을 선택한 후 [Save]를 선택합니다.

## 서브넷에서 네트워크 ACL 연결 끊기

서브넷에서 사용자 지정 네트워크 ACL을 연결 해제할 수 있습니다. 그러면 서브넷이 자동으로 기본 네트워크 ACL과 연결됩니다.

#### 네트워크 ACL에서 서브넷의 연결을 끊으려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택한 후 네트워크 ACL을 선택합니다.
3. 세부 정보 창에서 [Subnet Associations] 탭을 선택합니다.
4. [Edit]를 선택한 다음, 서브넷에 대한 [Associate] 확인란을 선택 취소합니다. [Save]를 선택합니다.

## 서브넷의 네트워크 ACL 변경

서브넷과 연결되어 있는 네트워크 ACL을 변경할 수 있습니다. 예를 들어 서브넷을 생성하면 생성된 서브넷이 처음에는 기본 네트워크 ACL과 연결됩니다. 서브넷을 사용자가 생성한 사용자 지정 네트워크 ACL과 대신 연결할 수도 있을 것입니다.

변경 사항은 잠시 후에 적용되므로, 서브넷의 네트워크 ACL을 변경한 후 서브넷에서 인스턴스를 종료했다가 다시 시작할 필요는 없습니다.

#### 서브넷의 네트워크 ACL 연결을 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Subnets]를 선택한 후 서브넷을 선택합니다.
3. [Network ACL] 탭을 선택한 후 [Edit]를 선택합니다.
4. Change to 목록에서 서브넷과 연결할 네트워크 ACL을 선택한 다음, Save를 선택합니다.

## 네트워크 ACL 삭제

네트워크 ACL과 연결된 서브넷이 없는 경우에만 네트워크 ACL을 삭제할 수 있습니다. 기본 네트워크 ACL은 삭제할 수 없습니다.

네트워크 ACL을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Network ACLs]를 선택합니다.
3. 네트워크 ACL을 선택한 후 [Delete]를 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

## 예: 서브넷 내 인스턴스에 대한 액세스 제어

이 예에서는 서브넷의 인스턴스가 서로 통신할 수 있으며, 신뢰할 수 있는 원격 컴퓨터로부터 액세스될 수 있습니다. 원격 컴퓨터는 로컬 네트워크의 컴퓨터이거나 사용자가 관리 작업을 수행하기 위해 사용자의 인스턴스와 연결하는 데 사용하는 다른 서브넷 또는 VPC의 인스턴스일 수 있습니다. 사용자의 보안 그룹 규칙 및 네트워크 ACL 규칙이 원격 컴퓨터의 IP 주소(172.31.1.2/32)로부터의 액세스를 허용합니다. 인터넷 또는 다른 네트워크로부터의 다른 모든 트래픽은 거부됩니다.

모든 인스턴스는 다음 규칙을 포함하는 동일한 보안 그룹(sg-1a2b3c4d)을 사용합니다.

### 인바운드 규칙

프로토콜 유형	프로토콜	포트 범위	소스	설명
모든 트래픽	모두	모두	sg-1a2b3c4d	동일한 보안 그룹과 연결되어 있는 인스턴스가 서로 통신하도록 허용합니다.
TCP	SSH	22	172.31.1.2/32	원격 컴퓨터로부터의 인바운드 SSH 액세스를 허용합니다. 인스턴스가 Windows 컴퓨터인 경우에는 이 규칙이 포트 3389에 대해 RDP 프로토콜을 사용해야 합니다.

### 아웃바운드 규칙

프로토콜 유형	프로토콜	포트 범위	목적지	설명
모든 트래픽	모두	모두	sg-1a2b3c4d	동일한 보안 그룹과 연결되어 있는 인스턴스가 서로 통신하도록 허용합니다.

이 서브넷은 다음 규칙을 포함하는 네트워크 ACL과 연결되어 있습니다.

### 인바운드 규칙

규칙 #	Type	프로토콜	포트 범위	소스	허용/거부	설명
------	------	------	-------	----	-------	----

100	SSH	TCP	22	172.31.1.2/32	허용	원격 컴퓨터로부터의 인바운드 트래픽을 허용합니다. 인스턴스가 Windows 컴퓨터인 경우에는 이 규칙이 포트 3389에 대해 RDP 프로토콜을 사용해야 합니다.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙과 일치하는 다른 모든 인바운드 트래픽을 거부합니다.

아웃바운드 규칙

규칙 #	Type	프로토콜	포트 범위	목적지	허용/거부	설명
100	사용자 지정 TCP	TCP	1024~65535	172.31.1.2/32	허용	원격 컴퓨터에 대한 아웃바운드 응답을 허용합니다. 네트워크 ACL은 상태 비저장이므로 인바운드 요청에 대한 응답 트래픽을 허용하려면 이 규칙이 필요합니다.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙과 일치하지 않는 다른 모든 아웃바운드 트래픽을 거부합니다.

이 시나리오는 인스턴스에 대한 보안 그룹 또는 보안 그룹 규칙을 변경할 수 있고 네트워크 ACL을 방어의 백업 계층으로 사용할 수 있는 유연성을 제공합니다. 네트워크 ACL 규칙은 서브넷의 모든 인스턴스에 적용되므로, 잘못하여 보안 그룹 규칙을 너무 허용적으로 지정할 경우 네트워크 ACL 규칙이 단일 IP 주소로부터의 액세스만 계속 허용하게 됩니다. 예를 들어 다음 규칙은 모든 IP 주소에 대해 인바운드 SSH 액세스를 허용하므로 이전 규칙보다 허용적입니다.

인바운드 규칙

Type	프로토콜	포트 범위	소스	설명
------	------	-------	----	----

모든 트래픽	모두	모두	sg-1a2b3c4d	동일한 보안 그룹과 연결되어 있는 인스턴스가 서로 통신하도록 허용합니다.
SSH	TCP	22	0.0.0.0/0	모든 IP 주소로부터의 SSH 액세스를 허용합니다.

아웃바운드 규칙

Type	프로토콜	포트 범위	목적지	설명
모든 트래픽	모두	모두	0.0.0.0/0	모든 아웃바운드 트래픽을 허용합니다.

하지만 서브넷 내부의 다른 인스턴스와 로컬 컴퓨터만 이 인스턴스에 액세스할 수 있습니다. 여전히 네트워크 ACL 규칙은 원격 컴퓨터로부터의 트래픽을 제외하고 서브넷에 대한 모든 인바운드 트래픽을 금지합니다.

## API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

### VPC에 대한 네트워크 ACL 만들기

- [create-network-acl](#)(AWS CLI)
- [New-EC2NetworkAcl](#)(Windows PowerShell용 AWS 도구)

### 한 개 이상의 네트워크 ACL에 대해 설명

- [describe-network-acls](#)(AWS CLI)
- [Get-EC2NetworkAcl](#)(Windows PowerShell용 AWS 도구)

### 네트워크 ACL에 규칙 추가

- [create-network-acl-entry](#)(AWS CLI)
- [New-EC2NetworkAclEntry](#)(Windows PowerShell용 AWS 도구)

### 네트워크 ACL에서 규칙 삭제

- [delete-network-acl-entry](#)(AWS CLI)
- [Remove-EC2NetworkAclEntry](#)(Windows PowerShell용 AWS 도구)

### 네트워크 ACL에 있는 기존 규칙 바꾸기

- [replace-network-acl-entry](#)(AWS CLI)
- [Set-EC2NetworkAclEntry](#)(Windows PowerShell용 AWS 도구)

### 네트워크 ACL 연결 바꾸기

- [replace-network-acl-association](#)(AWS CLI)



- [Set-EC2NetworkAclAssociation](#)(Windows PowerShell용 AWS 도구)

#### 네트워크 ACL 삭제

- [delete-network-acl](#)(AWS CLI)
- [Remove-EC2NetworkAcl](#)(Windows PowerShell용 AWS 도구)

## VPC에 권장되는 네트워크 ACL 규칙

VPC 마법사는 Amazon VPC의 일반적인 시나리오를 구현할 수 있도록 도와줍니다. 설명서에서 설명한 대로 이러한 시나리오를 구현하면 기본 네트워크 액세스 제어 목록(ACL)을 사용하게 되며, 이 ACL은 모든 인바운드와 아웃바운드 트래픽을 허용합니다. 추가적인 보안 계층이 필요할 경우 네트워크 ACL을 만들어 규칙을 추가할 수 있습니다. 각 시나리오에 대해 다음과 같은 규칙을 권장합니다.

#### 항목

- [시나리오 1을 위한 권장 규칙 \(p. 115\)](#)
- [시나리오 2를 위한 권장 규칙 \(p. 118\)](#)
- [시나리오 3을 위한 권장 규칙 \(p. 123\)](#)
- [시나리오 4를 위한 권장 규칙 \(p. 128\)](#)

네트워크 ACL 및 그 사용 방법에 대한 자세한 내용은 [네트워크 ACL \(p. 102\)](#)을 참조하십시오.

#### Important

- 여기서는 NAT 게이트웨이에 휘발성 포트 범위(예: 49152-65535) 또는 1024-65535를 사용합니다. 구성에 적합한 범위를 선택해야 합니다. 자세한 내용은 [휘발성 포트 \(p. 109\)](#) 단원을 참조하십시오.
- 서브넷의 호스트 간 최대 전송 단위(MTU)가 다른 경우, 경로 MTU 검색이 올바르게 작동하고 패킷 손실을 방지하도록 [Custom ICMP Rule] 유형과 [Destination Unreachable: fragmentation required, and DF flag set] 포트 범위(유형 3, 코드 4) 등의 인바운드 및 아웃바운드 네트워크 ACL 규칙을 추가해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [EC2 인스턴스에 대한 네트워크 MTU\(최대 전송 단위\)](#)를 참조하십시오.

## 시나리오 1을 위한 권장 규칙

시나리오 1은 인터넷 트래픽을 수신하고 전송할 수 있는 인스턴스를 가진 단일 서브넷입니다. 자세한 내용은 [시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC \(p. 17\)](#) 단원을 참조하십시오.

다음 표에는 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다.

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	0.0.0.0/0	TCP	443	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용

120	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	ALLOW	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	ALLOW	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	49152 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
120	0.0.0.0/0	TCP	49152 - 65535	허용	인터넷에서 클라이언트에 대한 아웃바운드 응답을 허용합니다(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)

## IPv6를 위한 권장 규칙

IPv6 지원을 통해 시나리오 1을 구현하고 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 생성한 경우, 네트워크 ACL에 별도의 규칙을 추가하여 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다.

다음은 네트워크 ACL에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
150	::/0	TCP	80	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
160	::/0	TCP	443	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	ALLOW	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	ALLOW	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
190	::/0	TCP	49152 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
130	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
140	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
150	::/0	TCP	49152 - 65535	허용	인터넷에서 클라이언트에 대한 아웃바운드 응답을 허용합니다(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

## 시나리오 2를 위한 권장 규칙

시나리오 2는 인터넷 트래픽을 수신하고 전송할 수 있는 인스턴스를 가진 퍼블릭 서브넷과, 인터넷으로부터 직접 트래픽을 수신할 수 없는 프라이빗 서브넷입니다. 하지만 퍼블릭 서브넷의 NAT 게이트웨이 또는 NAT 인스턴스를 통해 인터넷으로 트래픽을 시작하고 응답을 받을 수 있습니다. 자세한 내용은 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\)](#) (p. 23) 단원을 참조하십시오.

이 시나리오에서는 퍼블릭 서브넷에 대한 하나의 네트워크 ACL과, 프라이빗 서브넷을 위한 별도의 네트워크 ACL이 있습니다. 다음 표에는 각 ACL에 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다. 이들 규칙은 시나리오의 보안 그룹 규칙을 대부분 모방합니다.

### 퍼블릭 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	0.0.0.0/0	TCP	443	ALLOW	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	홈 네트워크의 공인 IP 주소 범위	TCP	22	ALLOW	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 공인 IP 주소 범위	TCP	3389	ALLOW	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	1024 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트</a> (p. 109) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.

Amazon Virtual Private Cloud 사용 설명서  
시나리오 2를 위한 권장 규칙

120	10.0.1.0/24	TCP	1433	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용.
130	10.0.1.0/24	TCP	3306	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용.
140	0.0.0.0/0	TCP	49152 - 65535	허용	인터넷에서 클라이언트에 대한 아웃바운드 응답을 허용합니다(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">회발성 포트 (p. 109)</a> 단원을 참조하십시오.
150	10.0.1.0/24	TCP	22	허용	프라이빗 서브넷의 인스턴스에 대한 아웃바운드 SSH 액세스 허용(SSH 바스천이 있다면 SSH 바스천으로부터).
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)

프라이빗 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	10.0.0.0/24	TCP	1433	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
110	10.0.0.0/24	TCP	3306	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MySQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
120	10.0.0.0/24	TCP	22	ALLOW	퍼블릭 서브넷의 SSH 바스천으로부터의 인바운드 SSH 트래픽 허용(SSH 바스천이 있는 경우)
130	10.0.0.0/24	TCP	3389	ALLOW	퍼블릭 서브넷의 Microsoft Terminal Services 게이트웨이로부터의 인바운드 RDP 트래픽 허용.
140	0.0.0.0/0	TCP	1024 - 65535	ALLOW	프라이빗 서브넷에서 발생되는 요청에 대해 발생하는 퍼블릭 서브넷의 NAT 디바이스로부터의 인바운드 리턴

					트래픽을 허용합니다(즉, 서브넷에서 시작되는 요청에 대해). 올바른 임시 포트 지정에 대한 정보는 이 주제의 첫 부분에 있는 중요 참고 사항을 보십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 IPv4 인바운드 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
120	10.0.0.0/24	TCP	49152 - 65535	ALLOW	퍼블릭 서브넷에 대한 아웃바운드 응답 허용(예: 퍼블릭 서브넷에서 프라이빗 서브넷의 DB 서버와 통신하는 웹 서버에 대한 응답). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)

## IPv6를 위한 권장 규칙

IPv6 지원을 통해 시나리오 2를 구현하고 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 생성한 경우, 네트워크 ACL에 별도의 규칙을 추가하여 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다.

다음은 네트워크 ACL에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

### 퍼블릭 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
150	::/0	TCP	80	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용

Amazon Virtual Private Cloud 사용 설명서  
시나리오 2를 위한 권장 규칙

160	::/0	TCP	443	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	ALLOW	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해)
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	ALLOW	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해)
190	::/0	TCP	1024 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
<b>아웃바운드</b>					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
160	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
170	::/0	TCP	443	ALLOW	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용
180	2001:db8:1234::/64	TCP	1433	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용.
190	2001:db8:1234::/64	TCP	3306	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용
200	::/0	TCP	49152 - 65535	ALLOW	인터넷에서 클라이언트에 대한 아웃바운드 응답을 허용 (예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.

210	2001:db8:1234::/64	TCP	22	허용	프라이빗 서브넷의 인스턴스에 대한 아웃바운드 SSH 액세스 허용(SSH 바스천이 있다면 SSH 바스천으로부터).
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

프라이빗 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
150	2001:db8:1234::/64	TCP	1433	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
160	2001:db8:1234::/64	TCP	3306	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MySQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
170	2001:db8:1234::/64	TCP	22	ALLOW	퍼블릭 서브넷의 SSH 바스천으로부터의 인바운드 SSH 트래픽 허용(해당되는 경우).
180	2001:db8:1234::/64	TCP	3389	ALLOW	퍼블릭 서브넷의 Microsoft Terminal Services 게이트웨이로부터의 인바운드 RDP 트래픽 허용(해당되는 경우).
190	::/0	TCP	1024 - 65535	ALLOW	프라이빗 서브넷에서 발생되는 요청에 대해 발생하는 외부 전용 인터넷 게이트웨이로부터의 인바운드 리턴 트래픽을 허용.  이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">회발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
130	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.



140	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
150	2001:db8:1234::/64	TCP	49152 - 65535	ALLOW	퍼블릭 서브넷에 대한 아웃바운드 응답 허용(예: 퍼블릭 서브넷에서 프라이빗 서브넷의 DB 서버와 통신하는 웹 서버에 대한 응답). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">회발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

## 시나리오 3을 위한 권장 규칙

시나리오 3은 인터넷 트래픽을 수신하고 전송할 수 있는 인스턴스를 가진 퍼블릭 서브넷과, VPN 연결을 통해 홈 네트워크하고만 통신할 수 있는 인스턴스를 가진 VPN 전용 서브넷입니다. 자세한 내용은 [시나리오 3: 공인 및 사설 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC \(p. 34\)](#) 단원을 참조하십시오.

이 시나리오에서는 퍼블릭 서브넷에 대한 하나의 네트워크 ACL과, VPN 전용 서브넷을 위한 별도의 네트워크 ACL이 있습니다. 다음 표에는 각 ACL에 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다.

### 퍼블릭 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	ALLOW	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTP 트래픽 허용
110	0.0.0.0/0	TCP	443	ALLOW	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTPS 트래픽 허용
120	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	ALLOW	홈 네트워크에서 웹 서버로의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	ALLOW	홈 네트워크에서 웹 서버로의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	49152 - 65535	ALLOW	서브넷에서 발신되는 요청에서 발생하는 인바운드 리턴 IPv4 트래픽 허용(해당되는 경우).

					이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
<b>아웃바운드</b>					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
120	10.0.1.0/24	TCP	1433	ALLOW	VPN 전용 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용.
130	10.0.1.0/24	TCP	3306	ALLOW	VPN 전용 서브넷의 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용.
140	0.0.0.0/0	TCP	49152 - 65535	ALLOW	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	거부	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽을 거부합니다(수정 불가).

### VPN 전용 서브넷의 ACL 설정

<b>인바운드</b>					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	10.0.0.0/24	TCP	1433	ALLOW	퍼블릭 서브넷의 웹 서버가 VPN 전용 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
110	10.0.0.0/24	TCP	3306	ALLOW	퍼블릭 서브넷의 웹 서버가 VPN 전용 서브넷의 MySQL

Amazon Virtual Private Cloud 사용 설명서  
시나리오 3을 위한 권장 규칙

					서버에 읽기 및 쓰기를 할 수 있도록 허용.
120	홈 네트워크의 프라이빗 IPv4 주소 범위	TCP	22	ALLOW	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(가상 프라이빗 게이트웨이를 통해).
130	홈 네트워크의 프라이빗 IPv4 주소 범위	TCP	3389	ALLOW	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(가상 프라이빗 게이트웨이를 통해).
140	홈 네트워크의 프라이빗 IP 주소 범위	TCP	49152 - 65535	ALLOW	홈 네트워크의 클라이언트에서 발신되는 인바운드 리턴 트래픽을 허용(가상 프라이빗 게이트웨이 경유). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	거부	이전 규칙에서 처리하지 않은 모든 인바운드 트래픽을 거부합니다(수정 불가).
<b>아웃바운드</b>					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
100	홈 네트워크의 프라이빗 IP 주소 범위	모두	모두	ALLOW	서브넷에서 홈 네트워크로의 모든 아웃바운드 트래픽 허용(가상 프라이빗 게이트웨이를 통해). 이 규칙에는 규칙 120도 포함되지만, 특정 프로토콜 유형과 포트 번호를 사용하여 이 규칙을 더 제한적으로 만들 수 있습니다. 이 규칙을 더 제한적으로 만드는 경우, 네트워크 ACL에 규칙 120을 포함하여 아웃바운드 응답이 차단되지 않도록 해야 합니다.
110	10.0.0.0/24	TCP	49152 - 65535	ALLOW	퍼블릭 서브넷의 웹 서버에 대해 아웃바운드 응답 허용. 올바른 임시 포트 지정에 대한 정보는 이 주제의 첫 부분에 있는 중요 참고 사항을 보십시오.

120	홈 네트워크의 프라이빗 IP 주소 범위	TCP	49152 - 65535	ALLOW	홈 네트워크의 클라이언트에 대한 아웃바운드 응답 허용 (가상 프라이빗 게이트웨이를 통해). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	거부	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽을 거부합니다(수정 불가).

## IPv6를 위한 권장 규칙

IPv6 지원을 통해 시나리오 3을 구현하고 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 생성한 경우, 네트워크 ACL에 별도의 규칙을 추가하여 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다.

다음은 네트워크 ACL에 대한 IPv6 전용 규칙입니다(위에 나열한 규칙에 추가되는 것).

### 퍼블릭 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
150	::/0	TCP	80	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
160	::/0	TCP	443	ALLOW	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	ALLOW	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해)
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	ALLOW	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해)
190	::/0	TCP	1024 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)

아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
150	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽을 허용합니다.
160	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽을 허용합니다.
170	2001:db8:1234::/64	TCP	1433	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용.
180	2001:db8:1234::/64	TCP	3306	ALLOW	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MySQL 액세스 허용.
190	::/0	TCP	49152 - 65535	ALLOW	인터넷에서 클라이언트에 대한 아웃바운드 응답을 허용 (예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

#### VPN 전용 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
150	2001:db8:1234::/64	TCP	1433	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
160	2001:db8:1234::/64	TCP	3306	ALLOW	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MySQL 서버에 읽기 및 쓰기를 할 수 있도록 허용.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드					

규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
130	2001:db8:1234:1000::/64	TCP	49152 - 65535	ALLOW	퍼블릭 서브넷에 대한 아웃바운드 응답 허용(예: 퍼블릭 서브넷에서 프라이빗 서브넷의 DB 서버와 통신하는 웹 서버에 대한 응답). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

## 시나리오 4를 위한 권장 규칙

시나리오 4는 VPN 연결을 통해 홈 네트워크하고만 통신할 수 있는 인스턴스를 가진 단일 서브넷입니다. 자세한 내용은 [시나리오 4: 사설 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC \(p. 44\)](#)를 참조하십시오.

다음 표에는 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다.

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
100	홈 네트워크의 프라이빗 IP 주소 범위	TCP	22	ALLOW	홈 네트워크에서 서브넷으로의 인바운드 SSH 트래픽 허용.
110	홈 네트워크의 프라이빗 IP 주소 범위	TCP	3389	ALLOW	홈 네트워크에서 서브넷으로의 인바운드 RDP 트래픽 허용.
120	홈 네트워크의 프라이빗 IP 주소 범위	TCP	49152 - 65535	ALLOW	서브넷에서 발신되는 요청으로부터 생기는 인바운드 리턴 트래픽을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	거부	이전 규칙에서 처리하지 않은 모든 인바운드 트래픽을 거부합니다(수정 불가).
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명

100	홈 네트워크의 프라이빗 IP 주소 범위	모두	모두	ALLOW	서브넷에서 홈 네트워크로의 모든 아웃바운드 트래픽 허용. 이 규칙에는 규칙 120도 포함되지만, 특정 프로토콜 유형과 포트 번호를 사용하여 이 규칙을 더 제한적으로 만들 수 있습니다. 이 규칙을 더 제한적으로 만드는 경우, 네트워크 ACL에 규칙 120을 포함하여 아웃바운드 응답이 차단되지 않도록 해야 합니다.
120	홈 네트워크의 프라이빗 IP 주소 범위	TCP	49152 - 65535	ALLOW	홈 네트워크의 클라이언트에 대한 아웃바운드 응답 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 <a href="#">취발성 포트 (p. 109)</a> 단원을 참조하십시오.
*	0.0.0.0/0	all	all	거부	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽을 거부합니다(수정 불가).

## IPv6를 위한 권장 규칙

IPv6 지원을 통해 시나리오 4를 구현하고 연결된 IPv6 CIDR 블록이 있는 VPC 및 서브넷을 생성한 경우, 네트워크 ACL에 별도의 규칙을 추가하여 인바운드 및 아웃바운드 IPv6 트래픽을 제어해야 합니다.

이 시나리오에서 데이터베이스 서버는 IPv6를 통해 VPN 통신으로 접속할 수 없습니다. 따라서 추가 네트워크 ACL 규칙은 필요 없습니다. 다음은 서브넷으로 가는, 그리고 서브넷으로부터 발생하는 IPv6 트래픽을 거부하는 기본 규칙입니다.

### VPN 전용 서브넷의 ACL 규칙

인바운드					
규칙 #	소스 IP	프로토콜	포트	허용/거부	설명
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드					
규칙 #	대상 IP	프로토콜	포트	허용/거부	설명
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

## Amazon VPC 리소스에 대한 액세스 제어

보안 자격 증명은 AWS의 서비스에서 사용자를 식별하고 Amazon VPC 리소스와 같은 AWS 리소스의 무제한 사용을 허가하는 데 사용됩니다. AWS Identity and Access Management(IAM)를 사용하면 보안 자격 증명을 공유하지 않고도 다른 사용자, 서비스 및 애플리케이션에 Amazon VPC 리소스 사용을 허용할 수 있습니다. 특정 Amazon EC2 API 작업을 사용하는 사용자 권한을 부여하여 리소스의 전체 사용 또는 제한적 사용을 허용할지 선택할 수 있습니다. 일부 API 작업은 리소스 수준 권한을 지원하여, 사용자가 생성하거나 수정할 수 있는 특정 리소스를 제어하도록 허용합니다.

### Important

현재, Amazon EC2 API 작업 중 일부는 리소스 수준 권한을 지원하지 않습니다. Amazon EC2 API 작업이 리소스 수준 권한을 지원하지 않는 경우 사용자에게 작업 사용 권한을 부여할 때 정책 명령문의 리소스 요소를 \*로 지정해야 합니다. 이 작업을 수행하는 방법을 보여주는 예제는 [1. VPC 관리 \(p. 130\)](#)에서 설명하는 정책 예제를 참조하십시오. 이후에 추가적인 API 작업을 위한 지원과 추가적인 Amazon EC2 리소스를 위한 ARN을 추가할 예정입니다. 어떤 Amazon EC2 API 작업에 어떤 ARN을 사용할 수 있는지, 그리고 각 ARN에 지원되는 조건 키에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 API 작업을 위해 지원되는 리소스와 조건 단원을](#) 참조하십시오.

Amazon EC2에 대한 정책 예제뿐 아니라 Amazon EC2에 대한 IAM 정책 생성과 EC2 API 작업을 위해 지원되는 리소스에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EC2에 대한 IAM 정책 단원을](#) 참조하십시오.

### 항목

- [AWS CLI 또는 SDK용 예제 정책 \(p. 130\)](#)
- [콘솔용 예제 정책 \(p. 137\)](#)

## AWS CLI 또는 SDK용 예제 정책

다음 예제는 IAM 사용자가 갖는 Amazon VPC 관련 권한을 제어하는 데 사용할 수 있는 정책 명령문을 보여줍니다. 이런 예제는 AWS CLI 또는 AWS SDK를 사용하는 사용자를 위해 고안된 것입니다.

- [1. VPC 관리 \(p. 130\)](#)
- [2. Amazon VPC에 대한 읽기 전용 정책 \(p. 131\)](#)
- [3. Amazon VPC에 대한 사용자 지정 정책 \(p. 132\)](#)
- [4. 특정 서브넷으로 인스턴스 시작 \(p. 132\)](#)
- [5. 특정 VPC로 인스턴스 시작 \(p. 133\)](#)
- [6. VPC에서 보안 그룹 관리 \(p. 133\)](#)
- [7. VPC 피어링 연결 생성 및 관리 \(p. 134\)](#)
- [8. VPC 엔드포인트 생성 및 관리 \(p. 136\)](#)

로 작업하기 위한 정책 예제는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [CLI 또는 SDK용 예제 정책 단원을](#) 참조하십시오.

### Example 1. VPC 관리

다음 정책에서는 사용자에게 VPC를 생성하고 관리하는 권한을 부여합니다. 이 정책을 네트워크 관리자 그룹에 연결할 수 있습니다. Action 요소는 VPC, 서브넷, 인터넷 게이트웨이, 고객 게이트웨이, 가상 프라이빗 게이트웨이, VPN 연결, 라우팅 테이블, 탄력적 IP 주소, 보안 그룹, 네트워크 ACL 및 DHCP 옵션 세트에 연결된 API 작업을 지정합니다. 또한, 이 정책에서는 그룹이 인스턴스를 실행, 중지, 시작 및 종료하도록 허용합니다. 그룹이 Amazon EC2 리소스를 나열하도록 허용하기도 합니다.



이 정책에서는 와일드카드를 사용하여 각각의 객체 유형에 대한 모든 작업을 지정합니다(예: \*SecurityGroup\*). 또는 각 작업을 명시적으로 나열할 수 있습니다. 와일드카드를 사용하는 경우, 이름에 정책의 와일드카드 문자열 중 어느 것이든 포함되어 있는 새 작업을 추가하면 정책에서 이런 새 작업에 대한 그룹 액세스 권한을 자동으로 부여한다는 점에 유의하십시오.

Resource 요소에 와일드카드가 사용되었으므로 사용자가 이러한 API 작업에 모든 리소스를 지정할 수 있습니다. API 작업이 리소스 수준 권한을 지원하지 않는 경우에도 \* 와일드카드가 필요합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:*Vpc*",
      "ec2:*Subnet*",
      "ec2:*Gateway*",
      "ec2:*Vpn*",
      "ec2:*Route*",
      "ec2:*Address*",
      "ec2:*SecurityGroup*",
      "ec2:*NetworkAcl*",
      "ec2:*DhcpOptions*",
      "ec2:RunInstances",
      "ec2:StopInstances",
      "ec2:StartInstances",
      "ec2:TerminateInstances",
      "ec2:Describe*"
    ],
    "Resource": "*"
  }
]
```

### Example 2. Amazon VPC에 대한 읽기 전용 정책

다음 정책에서는 사용자에게 VPC와 그 구성 요소를 나열하는 권한을 부여합니다. 사용자가 VPC와 구성 요소를 생성, 업데이트 또는 삭제할 수는 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeInternetGateways",
      "ec2:DescribeEgressOnlyInternetGateways",
      "ec2:DescribeVpcEndpoints",
      "ec2:DescribeNatGateways",
      "ec2:DescribeCustomerGateways",
      "ec2:DescribeVpnGateways",
      "ec2:DescribeVpnConnections",
      "ec2:DescribeRouteTables",
      "ec2:DescribeAddresses",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeNetworkAcls",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeTags",
      "ec2:DescribeInstances"
    ],
    "Resource": "*"
  }
]
```

### Example 3. Amazon VPC에 대한 사용자 지정 정책

다음 정책에서는 사용자에게 인스턴스를 시작, 중지, 시작 및 종료하고 Amazon EC2와 Amazon VPC에 사용 가능한 리소스를 설명하는 권한을 부여합니다.

정책에 있는 두 번째 설명문은 권한을 명시적으로 거부함으로써 사용자에게 더 광범위한 API 작업에 대한 액세스 권한을 부여할 수도 있는 다른 정책으로부터 보호하는 역할을 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [ "ec2:RunInstances",
               "ec2:StopInstances",
               "ec2:StartInstances",
               "ec2:TerminateInstances",
               "ec2:Describe*" ],
    "Resource": "*"
  },
  {
    "Effect": "Deny",
    "NotAction": [ "ec2:RunInstances",
                  "ec2:StopInstances",
                  "ec2:StartInstances",
                  "ec2:TerminateInstances",
                  "ec2:Describe*" ],
    "Resource": "*"
  }
]
```

### Example 4. 특정 서브넷으로 인스턴스 시작

다음 정책은 사용자에게 인스턴스를 특정 서브넷으로 시작하고 요청에 특정 보안 그룹을 사용하는 권한을 부여합니다. 이 정책에서는 subnet-1a2b3c4d에 대한 ARN과 sg-123abc123에 대한 ARN을 지정하여 이런 권한을 부여합니다. 사용자가 다른 서브넷으로 인스턴스를 시작하거나 다른 보안 그룹을 사용하여 시작하려고 하면 (또 다른 정책 또는 설명문에서 사용자에게 그런 권한을 부여하지 않는 한) 요청이 실패하게 됩니다.

또한, 이 정책에서는 네트워크 인터페이스 리소스를 사용할 권한도 부여합니다. 서브넷으로 시작할 때 기본적으로 RunInstances 요청은 기본 네트워크 인터페이스를 생성하므로, 사용자는 인스턴스를 시작할 때 이 리소스를 생성할 권한이 필요합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:subnet/subnet-1a2b3c4d",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/sg-123abc123"
    ]
  }
]
```

### Example 5. 특정 VPC로 인스턴스 시작

다음 정책에서는 사용자에게 특정 VPC 내에 있는 임의의 서브넷으로 인스턴스를 시작하는 권한을 부여합니다. 이 정책에서는 조건 키(`ec2:Vpc`)를 서브넷 리소스에 적용함으로써 이런 권한을 부여합니다.

또한, 이 정책에서는 사용자에게 "department=dev" 태그가 있는 AMI만 사용하여 인스턴스를 시작하는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:region:account:subnet/*",
    "Condition": {
      "StringEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:region::image/ami-*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/department": "dev"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }
]
```

### Example 6. VPC에서 보안 그룹 관리

다음 정책에서는 사용자에게 특정 VPC 내에 있는 임의의 보안 그룹에 대한 인바운드 및 아웃바운드 규칙을 생성하고 삭제하는 권한을 부여합니다. 이 정책에서는 `Authorize` 및 `Revoke` 작업을 위한 보안 그룹 리소스에 조건 키(`ec2:Vpc`)를 적용하여 이런 권한을 부여합니다.

두 번째 설명문은 사용자에게 모든 보안 그룹을 설명하는 권한을 부여합니다. 이는 사용자가 CLI를 사용하여 보안 그룹 규칙을 수정할 수 있도록 하는 데 필수적입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:RevokeSecurityGroupIngress",
      "ec2:RevokeSecurityGroupEgress"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DescribeSecurityGroups"
  }
]
```

```

    "Resource": "arn:aws:ec2:region:account:security-group/*",
    "Condition": {
      "StringEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d"
      }
    },
    "Effect": "Allow",
    "Action": "ec2:DescribeSecurityGroups",
    "Resource": "*"
  }
}

```

### Example 7. VPC 피어링 연결 생성 및 관리

다음은 VPC 피어링 연결의 생성 및 수정의 관리에 사용할 수 있는 정책 예제입니다.

#### a. VPC 피어링 연결 생성

사용자는 다음 정책을 통해 `Purpose=Peering`으로 태그 지정된 VPC만 사용하여 VPC 피어링 연결 요청을 생성할 수 있습니다. 첫 번째 설명문은 조건 키(`ec2:ResourceTag`)를 VPC 리소스에 적용합니다. `CreateVpcPeeringConnection` 작업을 위한 VPC 리소스는 항상 요청자 VPC입니다.

두 번째 설명문은 사용자에게 VPC 피어링 연결 리소스를 생성하는 권한을 부여하므로, 특정 리소스 ID 대신 \* 와일드카드를 사용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Peering"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account:vpc-peering-connection/*"
    }
  ]
}

```

사용자는 AWS 계정 333333333333에서 다음 정책을 통해 us-east-1 리전에 있는 VPC를 사용하여 VPC 피어링 연결을 생성할 수 있지만, 피어링 조건을 허용할 VPC가 특정 계정(777788889999)의 특정 VPC(`vpc-aaa111bb`)인 경우에만 그렇습니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:333333333333:vpc/*"
    },
    {
      "Effect": "Allow",

```

```

    "Action": "ec2:CreateVpcPeeringConnection",
    "Resource": "arn:aws:ec2:region:333333333333:vpc-peering-connection/*",
    "Condition": {
      "ArnEquals": {
        "ec2:AcceptorVpc": "arn:aws:ec2:region:777788889999:vpc/vpc-aaa111bb"
      }
    }
  }
}

```

#### b. VPC 피어링 연결 허용

사용자는 다음 정책을 통해 AWS 계정 444455556666의 VPC 피어링 연결 요청만 허용할 수 있습니다. 이는 사용자가 알 수 없는 계정의 VPC 피어링 연결 요청을 허용하지 못하게 하는 데 도움이 됩니다. 첫 번째 설명 문에서는 `ec2:RequesterVpc` 조건 키를 사용하여 이를 적용합니다.

또한, 이 정책에서는 사용자에게 VPC에 `Purpose=Peering` 태그가 있을 때만 VPC 피어링 요청을 허용하는 권한을 부여합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:AcceptVpcPeeringConnection",
    "Resource": "arn:aws:ec2:region:account:vpc-peering-connection/*",
    "Condition": {
      "ArnEquals": {
        "ec2:RequesterVpc": "arn:aws:ec2:region:444455556666:vpc/*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:AcceptVpcPeeringConnection",
    "Resource": "arn:aws:ec2:region:account:vpc/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/Purpose": "Peering"
      }
    }
  }
]
}

```

#### c. VPC 피어링 연결 삭제

444455556666 계정의 사용자는 다음 정책을 통해 같은 계정에 있는 지정된 VPC `vpc-1a2b3c4d`를 사용하는 VPC 피어링 연결을 제외한 모든 VPC 피어링 연결을 삭제할 수 있습니다. VPC가 원래 VPC 피어링 연결 요청에서 요청자 VPC 또는 피어 VPC였을 수 있으므로, 이 정책에서는 `ec2:AcceptorVpc` 및 `ec2:RequesterVpc` 조건 키를 모두 지정합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2>DeleteVpcPeeringConnection",
    "Resource": "arn:aws:ec2:region:444455556666:vpc-peering-connection/*",
    "Condition": {
      "ArnNotEquals": {
        "ec2:AcceptorVpc": "arn:aws:ec2:region:444455556666:vpc/vpc-1a2b3c4d",

```

```

    "ec2:RequesterVpc": "arn:aws:ec2:region:444455556666:vpc/vpc-1a2b3c4d"
  }
}
]
}

```

#### d. 특정 계정 내에서의 작업

사용자는 다음 정책을 통해 특정 계정 내에 있는 VPC 피어링 연결 전체를 사용하여 작업을 수행할 수 있습니다. VPC 피어링 연결이 모두 AWS 계정 333333333333 내에 있는 경우, 사용자는 VPC 피어링 연결 보기, 생성, 허용, 거부 및 삭제를 수행할 수 있습니다.

첫 번째 설명문에서는 사용자가 모든 VPC 피어링 연결을 볼 수 있도록 허용합니다. 이 API 작업 (`DescribeVpcPeeringConnections`)은 현재 리소스 수준 권한을 지원하지 않으므로, 이 경우에는 `Resource` 요소에 `*` 와일드카드가 필요합니다.

두 번째 설명문에서는 사용자가 VPC 피어링 연결을 생성하도록 허용하고, 그렇게 하기 위해 333333333333 계정에 있는 모든 VPC에 대한 액세스를 허용합니다.

세 번째 설명문에서는 모든 VPC 피어링 연결 작업을 허용하기 위해 `Action` 요소의 일부로서 `*` 와일드카드를 사용합니다. 조건 키는 333333333333 계정의 일부인 VPC와의 VPC 피어링 연결 시에만 이런 작업을 수행할 수 있도록 합니다. 예를 들어 수락자 또는 요청자 VPC가 다른 계정에 있는 경우에는 사용자가 VPC 피어링 연결을 삭제할 수 없습니다. 사용자는 다른 계정에 있는 VPC와의 VPC 피어링 연결을 생성할 수 없습니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeVpcPeeringConnections",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [ "ec2:CreateVpcPeeringConnection", "ec2:AcceptVpcPeeringConnection" ],
      "Resource": "arn:aws:ec2:*:333333333333:vpc/*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcPeeringConnection",
      "Resource": "arn:aws:ec2:*:333333333333:vpc-peering-connection/*",
      "Condition": {
        "ArnEquals": {
          "ec2:AcceptorVpc": "arn:aws:ec2:*:333333333333:vpc/*",
          "ec2:RequesterVpc": "arn:aws:ec2:*:333333333333:vpc/*"
        }
      }
    }
  ]
}

```

#### Example 8. VPC 엔드포인트 생성 및 관리

다음 정책에서는 사용자에게 VPC 엔드포인트를 생성, 수정, 보기 및 삭제하는 권한을 부여합니다. 어떤 `ec2:*VpcEndpoint*` 작업도 리소스 수준 권한을 지원하지 않으므로, 사용자가 모든 리소스로 작업하도록 허용하려면 `Resource` 요소에 대해 `*` 와일드카드를 사용해야 합니다.

```

{
  "Version": "2012-10-17",

```

```
"Statement": [{  
  "Effect": "Allow",  
  "Action": "ec2:*VpcEndpoint*",  
  "Resource": "*" }  
]
```

## 콘솔용 예제 정책

Amazon VPC 콘솔에서 IAM 정책을 사용하여 특정 리소스를 조회하고 관련 작업을 수행할 권한을 부여할 수 있습니다. 이전 섹션의 예제 정책을 사용할 수 있지만 해당 정책은 AWS CLI 또는 AWS SDK를 통한 요청에 맞게 설계되었습니다. 콘솔에서는 추가적인 API 작업을 통해 해당 기능을 구현하므로 이러한 정책이 예상과 다르게 작동할 수 있습니다.

이 섹션에서는 VPC 콘솔의 특정 부분을 사용하도록 허용하는 정책을 보여줍니다.

- [1. VPC 마법사 사용 \(p. 137\)](#)
- [2. VPC 관리 \(p. 141\)](#)
- [3. 보안 그룹 관리 \(p. 142\)](#)
- [4. VPC 피어링 연결 생성 \(p. 143\)](#)

### Example 1. VPC 마법사 사용

VPC를 바로 사용할 수 있도록, Amazon VPC 콘솔에서 VPC 마법사를 사용하여 VPC를 생성, 설정 및 구성할 수 있습니다. 이 마법사는 사용자의 요구 사항에 따라 다양한 구성 옵션을 제공합니다. VPC 생성을 위해 VPC 마법사를 사용하는 자세한 방법은 [시나리오 및 예시 \(p. 17\)](#) 단원을 참조하십시오.

사용자가 VPC 마법사를 사용할 수 있도록 하려면, 사용자에게 선택한 구성의 일부를 이루는 리소스를 생성하고 수정하는 권한을 부여해야 합니다. 다음 예제 정책에서는 각 마법사 구성 옵션에 필요한 작업을 보여줍니다.

#### Note

어떤 시점에 VPC 마법사가 실패하는 경우 VPC 마법사는 생성한 리소스를 분리하고 삭제하려 합니다. 사용자에게 이런 작업을 사용할 권한을 부여하지 않으면, 해당 리소스가 계정에 그대로 남습니다.

#### 옵션 1: 단일 퍼블릭 서브넷이 있는 VPC

첫 번째 VPC 마법사 구성 옵션은 단일 서브넷이 있는 VPC를 생성하는 옵션입니다. IAM 정책에서는, 사용자가 이 마법사 옵션을 올바르게 사용할 수 있도록 사용자에게 다음 작업을 사용할 권한을 부여해야 합니다.

- `ec2:CreateVpc`, `ec2:CreateSubnet`, `ec2:CreateRouteTable` 및 `ec2:CreateInternetGateway`: VPC, 서브넷, 사용자 지정 라우팅 테이블 및 인터넷 게이트웨이를 생성합니다.
- `ec2:DescribeAvailabilityZones`: 서브넷에 대한 [Availability Zone] 목록 및 CIDR 블록 필드가 있는 마법사의 섹션을 표시합니다. 사용자가 기본 설정을 그대로 두려고 하더라도, 이런 옵션이 표시되어 있지 않으면 VPC를 생성할 수 없습니다.
- `ec2:DescribeVpcEndpointServices`: 마법사의 VPC 종단점 섹션을 표시합니다.
- `ec2:AttachInternetGateway`: 인터넷 게이트웨이를 VPC에 연결합니다.
- `ec2:CreateRoute`: 사용자 지정 라우팅 테이블에 경로를 생성합니다. 이 경로는 트래픽이 인터넷 게이트웨이로 전달되도록 합니다.
- `ec2:AssociateRouteTable`: 사용자 지정 라우팅 테이블을 서브넷에 연결합니다.

- `ec2:ModifyVpcAttribute`: DNS 호스트 이름을 활성화하여 이 VPC로 시작한 각 인스턴스가 DNS 호스트 이름을 수신하도록 VPC의 속성을 수정합니다.

이 정책에서는 어떤 API 작업도 리소스 수준 권한을 지원하지 않으므로, 사용자가 사용할 수 있는 특정 리소스를 제어할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:CreateVpc", "ec2:CreateSubnet", "ec2:DescribeAvailabilityZones",
      "ec2:DescribeVpcEndpointServices",
      "ec2:CreateRouteTable", "ec2:CreateRoute", "ec2:CreateInternetGateway",
      "ec2:AttachInternetGateway", "ec2:AssociateRouteTable", "ec2:ModifyVpcAttribute"
    ],
    "Resource": "*"
  }
]
```

**옵션 2: 퍼블릭 및 프라이빗 서브넷이 있는 VPC**

두 번째 VPC 마법사 구성 옵션은 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 생성하고, NAT 게이트웨이 또는 NAT 인스턴스를 시작하는 옵션을 제공합니다. 다음 정책에는 이전 예제(옵션 1)와 같은 작업 외에도, 사용자가 NAT 게이트웨이 또는 NAT 인스턴스를 실행하고 구성하도록 허용하는 작업도 있습니다.

다음 작업은 NAT 인스턴스를 시작하는지 NAT 게이트웨이를 시작하는지 여부에 상관없이 수행해야 합니다.

- `ec2:DescribeKeyPairs`: 기존 키 페어의 목록을 표시하고 마법사의 NAT 섹션을 로드합니다.

다음 작업은 NAT 게이트웨이를 만드는 경우에 필요하고, NAT 인스턴스를 시작하는 경우에는 필요하지 않습니다.

- `ec2:CreateNatGateway`: NAT 게이트웨이를 만듭니다.
- `ec2:DescribeNatGateways`: NAT 게이트웨이가 사용 가능한 상태가 될 때까지 NAT 게이트웨이의 상태를 확인합니다.
- `ec2:DescribeAddresses`: 계정에서 NAT 게이트웨이와 연결할 수 있는 탄력적 IP 주소를 나열합니다.

다음 작업은 NAT 인스턴스를 시작하는 경우에 필요하고, NAT 게이트웨이를 만드는 경우에는 필요하지 않습니다.

- `ec2:DescribeImages`: NAT 인스턴스로 실행하도록 구성된 AMI를 찾습니다.
- `ec2:RunInstances`: NAT 인스턴스를 시작합니다.
- `ec2:AllocateAddress` 및 `ec2:AssociateAddress`: 계정에 탄력적 IP 주소를 할당한 다음 NAT 인스턴스와 연결합니다.
- `ec2:ModifyInstanceAttribute`: NAT 인스턴스의 원본/대상 확인을 비활성화합니다.
- `ec2:DescribeInstances`: 인스턴스가 실행 중인 상태가 될 때까지 인스턴스의 상태를 확인합니다.
- `ec2:DescribeRouteTables`, `ec2:DescribeVpnGateways` 및 `ec2:DescribeVpcs`: 기본 라우팅 테이블에 추가해야 하는 경로에 대한 정보를 수집합니다.

다음 정책은 사용자가 NAT 인스턴스 또는 NAT 게이트웨이를 만들 수 있도록 허용합니다.

```
{
```



```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateVpc", "ec2:CreateSubnet", "ec2:DescribeAvailabilityZones",
      "ec2:DescribeVpcEndpointServices",
      "ec2:CreateRouteTable", "ec2:CreateRoute", "ec2:CreateInternetGateway",
      "ec2:CreateNatGateway",
      "ec2:AttachInternetGateway", "ec2:AssociateRouteTable", "ec2:ModifyVpcAttribute",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeImages", "ec2:RunInstances", "ec2:AllocateAddress",
      "ec2:AssociateAddress",
      "ec2:DescribeAddresses", "ec2:DescribeInstances", "ec2:ModifyInstanceAttribute",
      "ec2:DescribeRouteTables",
      "ec2:DescribeVpnGateways", "ec2:DescribeVpcs", "ec2:DescribeSubnets",
      "ec2:DescribeNatGateways"
    ],
    "Resource": "*"
  }
]
}

```

ec2:RunInstances 작업에 대한 리소스 수준 권한을 사용하여 사용자의 인스턴스 시작 능력을 제어할 수 있습니다. 예를 들어 사용자가 NAT 사용 AMI에서만 인스턴스를 시작할 수 있도록 이 AMI의 ID를 지정할 수 있습니다. 마법사가 NAT 인스턴스를 시작하기 위해 사용하는 AMI를 찾으려면 전체 권한을 가진 사용자로 Amazon VPC 콘솔에 로그인한 후에 VPC 마법사의 두 번째 옵션을 수행합니다. Amazon EC2 콘솔로 전환하고, [Instances] 페이지를 선택하고, NAT 인스턴스를 선택한 다음, 이 인스턴스를 시작하는 데 사용한 AMI ID를 기록합니다.

사용자는 다음 정책을 통해 ami-1a2b3c4d만 사용하여 인스턴스를 시작할 수 있습니다. 사용자가 다른 AMI를 사용하여 인스턴스를 시작하려고 하면 시작에 실패합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:CreateVpc", "ec2:CreateSubnet", "ec2:DescribeAvailabilityZones",
      "ec2:DescribeVpcEndpointServices",
      "ec2:CreateRouteTable", "ec2:CreateRoute", "ec2:CreateInternetGateway",
      "ec2:AttachInternetGateway", "ec2:AssociateRouteTable", "ec2:ModifyVpcAttribute",
      "ec2:DescribeKeyPairs", "ec2:DescribeImages", "ec2:AllocateAddress",
      "ec2:AssociateAddress",
      "ec2:DescribeInstances", "ec2:ModifyInstanceAttribute", "ec2:DescribeRouteTables",
      "ec2:DescribeVpnGateways", "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:image/ami-1a2b3c4d",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }
]
}

```

### 옵션 3: 퍼블릭 및 프라이빗 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC

세 번째 VPC 마법사 구성 옵션은 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 생성하고, VPC와 자체 네트워크 사이에 VPN 연결을 생성합니다. IAM 정책에서는, 사용자에게 옵션 1과 같은 작업을 사용할 권한을 부여해야 합니다. 이를 통해 사용자는 VPC 한 개와 서브넷 두 개를 생성하고, 퍼블릭 서브넷에 대한 라우팅을 구성할 수 있습니다. VPN 연결을 생성하려면 사용자가 다음과 같은 작업을 사용할 권한도 가져야 합니다.

- `ec2:CreateCustomerGateway`: 고객 게이트웨이를 생성합니다.
- `ec2:CreateVpnGateway` 및 `ec2:AttachVpnGateway`: 가상 프라이빗 게이트웨이를 생성하여 VPC에 연결합니다.
- `ec2:EnableVgwRoutePropagation`: 경로가 라우팅 테이블로 자동으로 전파되도록 경로 전파를 활성화합니다.
- `ec2:CreateVpnConnection`: VPN 연결을 생성합니다.
- `ec2:DescribeVpnConnections`, `ec2:DescribeVpnGateways` 및 `ec2:DescribeCustomerGateways`: 마법사의 두 번째 구성 페이지에 옵션을 표시합니다.
- `ec2:DescribeVpcs` 및 `ec2:DescribeRouteTables`: 기본 라우팅 테이블에 추가해야 하는 경로에 대한 정보를 수집합니다.

이 정책에서는 어떤 API 작업도 리소스 수준 권한을 지원하지 않으므로, 사용자가 사용할 수 있는 특정 리소스를 제어할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:CreateVpc", "ec2:CreateSubnet", "ec2:DescribeAvailabilityZones",
      "ec2:DescribeVpcEndpointServices",
      "ec2:CreateRouteTable", "ec2:CreateRoute", "ec2:CreateInternetGateway",
      "ec2:AttachInternetGateway", "ec2:AssociateRouteTable", "ec2:ModifyVpcAttribute",
      "ec2:CreateCustomerGateway", "ec2:CreateVpnGateway", "ec2:AttachVpnGateway",
      "ec2:EnableVgwRoutePropagation", "ec2:CreateVpnConnection",
      "ec2:DescribeVpnGateways",
      "ec2:DescribeCustomerGateways", "ec2:DescribeVpnConnections",
      "ec2:DescribeRouteTables",
      "ec2:DescribeNetworkAcls", "ec2:DescribeInternetGateways", "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  }
]
```

### 옵션 4: 프라이빗 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC

네 번째 VPC 구성 옵션은 프라이빗 서브넷이 있는 VPC를 생성하고, VPC와 자체 네트워크 사이에 VPN 연결을 생성합니다. 다른 세 옵션과는 달리, 사용자가 VPC에 인터넷 게이트웨이를 연결하거나 생성하는 권한이 필요하지 않고, 라우팅 테이블을 생성하여 서브넷과 연결할 권한이 필요하지 않습니다. VPN 연결을 설정하려면 이전 예제(옵션 3)와 같은 권한이 필요합니다.

이 정책에서는 어떤 API 작업도 리소스 수준 권한을 지원하지 않으므로, 사용자가 사용할 수 있는 특정 리소스를 제어할 수 없습니다.

```
{
  "Version": "2012-10-17",
```

```

"Statement": [{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateVpc", "ec2:CreateSubnet", "ec2:DescribeAvailabilityZones",
    "ec2:DescribeVpcEndpointServices",
    "ec2:ModifyVpcAttribute", "ec2:CreateCustomerGateway", "ec2:CreateVpnGateway",
    "ec2:AttachVpnGateway", "ec2:EnableVgwRoutePropagation", "ec2:CreateVpnConnection",
    "ec2:DescribeVpnGateways", "ec2:DescribeCustomerGateways",
    "ec2:DescribeVpnConnections",
    "ec2:DescribeRouteTables", "ec2:DescribeNetworkAcls", "ec2:DescribeInternetGateways",
    "ec2:DescribeVpcs"
  ],
  "Resource": "*"
}]
}

```

## Example 2. VPC 관리

VPC 콘솔의 [Your VPCs] 페이지에서 VPC를 생성하거나 삭제할 수 있습니다. VPC를 보려면 사용자에게 `ec2:DescribeVpcs` 작업을 사용할 권한이 있어야 합니다. [Create VPC] 대화 상자를 사용하여 VPC를 생성하려면 사용자에게 `ec2:CreateVpc` 작업을 사용할 권한이 있어야 합니다.

### Note

기본적으로, VPC 콘솔은 `Name`의 키와 사용자가 지정하는 값을 포함한 태그를 생성합니다. 사용자가 `ec2:CreateTags` 작업을 사용할 권한이 없는 경우에는 VPC를 생성하려고 할 때 [Create VPC] 대화 상자에 오류가 나타납니다. 하지만 VPC 생성에 성공했을 수도 있습니다.

VPC를 설정할 때는 일반적으로 서브넷 및 인터넷 게이트웨이와 같이 종속적인 객체가 다수 생성됩니다. 이런 종속적 객체의 연결을 끊고 삭제할 때까지는 VPC를 삭제할 수 없습니다. 콘솔을 사용하여 VPC를 삭제하면 이런 작업이 자동으로 수행됩니다(단, 인스턴스 종료는 제외. 인스턴스는 수동으로 종료해야 함).

사용자는 다음 예제를 통해 [Your VPCs] 페이지에서 VPC를 보고 생성하며, VPC 마법사에서 첫 번째 옵션으로 생성된 VPC(단일 퍼블릭 서브넷이 있는 VPC)를 삭제할 수 있습니다. 이 VPC에는 사용자 지정 라우팅 테이블과 연결된 서브넷 한 개와 이 서브넷에 연결된 인터넷 게이트웨이 한 개가 있습니다. 콘솔을 사용하여 VPC와 그 구성 요소를 삭제하려면 콘솔이 이 VPC에 종속적인 다른 리소스가 있는지 확인할 수 있도록 여러 가지 `ec2:Describe*` 작업을 사용할 권한을 사용자에게 부여해야 합니다. 또한, 사용자에게 서브넷에서 라우팅 테이블의 연결을 끊고, VPC에서 인터넷 게이트웨이를 분리하고, 이 두 리소스를 모두 삭제할 수 있는 권한도 부여해야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs", "ec2:DescribeRouteTables", "ec2:DescribeVpnGateways",
      "ec2:DescribeInternetGateways",
      "ec2:DescribeSubnets", "ec2:DescribeDhcpOptions", "ec2:DescribeInstances",
      "ec2:DescribeVpcAttribute",
      "ec2:DescribeNetworkAcls", "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeAddresses",
      "ec2:DescribeVpcPeeringConnections", "ec2:DescribeSecurityGroups",
      "ec2:CreateVpc", "ec2>DeleteVpc", "ec2:DetachInternetGateway",
      "ec2>DeleteInternetGateway",
      "ec2:DisassociateRouteTable", "ec2>DeleteSubnet", "ec2>DeleteRouteTable"
    ],
    "Resource": "*"
  }
]
}

```

어떤 `ec2:Describe*` API 작업에도 리소스 수준 권한을 적용할 수 없지만, 사용자가 삭제할 수 있는 리소스를 제어하기 위해 `ec2:Delete*` 작업 중 일부에 리소스 수준 권한을 적용할 수 있습니다.

예를 들어 사용자는 다음 정책을 통해 `Purpose=Test` 태그가 있는 라우팅 테이블과 인터넷 게이트웨이만 삭제할 수 있습니다. 사용자는 이 태그가 없는 개별 라우팅 테이블 또는 인터넷 게이트웨이를 삭제할 수 없으며, 마찬가지로 사용자는 VPC 콘솔을 사용하여 다른 라우팅 테이블이나 인터넷 게이트웨이와 연결된 VPC를 삭제할 수 없습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs", "ec2:DescribeRouteTables", "ec2:DescribeVpnGateways",
      "ec2:DescribeInternetGateways",
      "ec2:DescribeSubnets", "ec2:DescribeDhcpOptions", "ec2:DescribeInstances",
      "ec2:DescribeVpcAttribute",
      "ec2:DescribeNetworkAcls", "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeAddresses",
      "ec2:DescribeVpcPeeringConnections", "ec2:DescribeSecurityGroups",
      "ec2:CreateVpc", "ec2:DeleteVpc", "ec2:DetachInternetGateway",
      "ec2:DisassociateRouteTable", "ec2:DeleteSubnet"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DeleteInternetGateway",
    "Resource": "arn:aws:ec2:region:account:internet-gateway/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/Purpose": "Test"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DeleteRouteTable",
    "Resource": "arn:aws:ec2:region:account:route-table/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/Purpose": "Test"
      }
    }
  }
  ]
}
```

### Example 3. 보안 그룹 관리

Amazon VPC 콘솔의 [Security Groups] 페이지에서 보안 그룹을 보려면 사용자에게 `ec2:DescribeSecurityGroups` 작업을 사용할 권한이 있어야 합니다. [Create Security Group] 대화 상자를 사용하여 보안 그룹을 생성하려면 사용자에게 `ec2:DescribeVpcs` 및 `ec2:CreateSecurityGroup` 작업을 사용할 권한이 있어야 합니다. 사용자에게 `ec2:DescribeSecurityGroups` 작업을 사용할 권한이 없더라도 사용자가 이 대화 상자를 사용하여 보안 그룹을 생성할 수는 있지만, 해당 그룹이 생성되지 않았음을 나타내는 오류가 발생할 수도 있습니다.

사용자는 [Create Security Group] 대화 상자에서 보안 그룹 이름과 설명을 추가해야 하지만, `ec2:CreateTags` 작업을 사용할 권한이 자신에게 부여되지 않은 경우에는 [Name tag] 필드에 대한 값을 입력할 수 없습니다. 하지만 보안 그룹을 생성하기 위해 이 작업을 수행할 필요는 없습니다.

사용자는 다음 정책을 통해 보안 그룹을 보고 생성하고, `vpc-1a2b3c4d`와 연결된 보안 그룹에 인바운드 및 아웃바운드 규칙을 추가 및 제거할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSecurityGroups", "ec2:DescribeVpcs", "ec2:CreateSecurityGroup"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:RevokeSecurityGroupIngress", "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:*:*:security-group/*",
    "Condition": {
      "ArnEquals": {
        "ec2:Vpc": "arn:aws:ec2:*:*:vpc/vpc-1a2b3c4d"
      }
    }
  }
  ]
}
```

#### Example 4. VPC 피어링 연결 생성

Amazon VPC 콘솔에서 VPC 피어링 연결을 보려면 사용자에게 `ec2:DescribePeeringConnections` 작업을 사용할 권한이 있어야 합니다. [Create VPC Peering Connection] 대화 상자를 사용하면 사용자에게 `ec2:DescribeVpcs` 작업을 사용할 권한이 있어야 합니다. 사용자는 이 권한을 통해 VPC를 보고 선택할 수 있습니다. 이 작업을 사용하지 않으면 대화 상자를 로드할 수 없습니다.

`ec2:DescribeVpcPeeringConnections`를 제외한 모든 `ec2:*PeeringConnection` 작업에 리소스 수준 권한을 적용할 수 있습니다.

사용자는 다음 정책을 통해 VPC 피어링 연결을 보고 [Create VPC Peering Connection] 대화 상자에서 특정 요청자 VPC(`vpc-1a2b3c4d`)만 사용하여 VPC 피어링 연결을 생성할 수 있습니다. 사용자가 다른 요청자 VPC로 VPC 피어링 연결을 생성하려고 하면 요청이 실패합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcPeeringConnections", "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:CreateVpcPeeringConnection",
    "Resource": [
      "arn:aws:ec2:*:*:vpc/vpc-1a2b3c4d",
      "arn:aws:ec2:*:*:vpc-peering-connection/*"
    ]
  }
  ]
}
```

VPC 피어링 연결을 사용한 작업을 위해 IAM 정책을 작성하는 추가 예제를 보려면 [7. VPC 피어링 연결 생성 및 관리 \(p. 134\)](#) 단원을 참조하십시오.

## VPC 흐름 로그

VPC 흐름 로그는 VPC의 네트워크 인터페이스에서 전송되고 수신되는 IP 트래픽에 대한 정보를 수집할 수 있는 기능입니다. 흐름 로그 데이터는 Amazon CloudWatch Logs를 사용하여 저장됩니다. 흐름 로그를 생성하고 난 다음 Amazon CloudWatch Logs의 데이터를 확인하고 가져올 수 있습니다.

흐름 로그는 특정 트래픽이 인스턴스에 액세스하지 않는 문제를 해결하는 등 다양한 작업에 도움을 주므로 매우 제한적인 보안 그룹 규칙을 진단할 수 있게 도와줍니다. 흐름 로그를 인스턴스에 액세스하는 트래픽을 모니터링하기 위한 보안 도구로 사용할 수도 있습니다.

흐름 로그 사용에 대한 추가 비용은 없지만 표준 CloudWatch Logs 비용이 적용됩니다. 자세한 내용은 [Amazon CloudWatch 요금](#)을 참조하십시오.

### 항목

- [흐름 로그 기본 사항](#) (p. 144)
- [흐름 로그 제한](#) (p. 145)
- [흐름 로그 레코드](#) (p. 145)
- [흐름 로그에 대한 IAM 역할](#) (p. 146)
- [흐름 로그 작업](#) (p. 147)
- [문제 해결](#) (p. 149)
- [API 및 CLI 개요](#) (p. 150)
- 예: [흐름 로그 레코드](#) (p. 150)
- 예: [흐름 로그에 대한 CloudWatch 지표 필터 및 경보 만들기](#) (p. 152)

## 흐름 로그 기본 사항

VPC, 서브넷 또는 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 서브넷이나 VPC에 대한 흐름 로그를 생성할 경우, VPC 또는 서브넷의 각 네트워크 인터페이스가 모니터링됩니다. 흐름 로그 데이터는 CloudWatch Logs의 로그 그룹에 게시되고, 각 네트워크 인터페이스는 고유의 로그 스트림을 가집니다. 로그 스트림에는 흐름 로그 레코드가 포함되며, 이것은 해당 네트워크 인터페이스의 트래픽을 설명하는 필드로 구성된 로그 이벤트입니다. 자세한 내용은 [흐름 로그 레코드](#) (p. 145) 단원을 참조하십시오.

흐름 로그를 생성하려면 흐름 로그를 생성하려는 리소스, 캡처할 트래픽 유형(허용된 트래픽, 거부된 트래픽 또는 모든 트래픽), 흐름 로그를 게시할 CloudWatch Logs의 로그 그룹 이름, CloudWatch Logs 로그 그룹에 흐름 로그를 게시할 권한이 있는 IAM 역할의 ARN 등을 지정합니다. 존재하지 않는 로그 그룹의 이름을 지정할 경우 해당 로그 그룹이 만들어집니다. 흐름 로그를 생성한 후, 데이터를 수집하여 CloudWatch Logs에 게시하기까지 몇 분 정도의 시간이 걸릴 수 있습니다. 흐름 로그는 네트워크 인터페이스에 대한 로그 스트림을 실시간으로 캡처하지 않습니다.

여러 흐름 로그를 생성하여 CloudWatch Logs의 같은 로그 그룹에 데이터를 게시할 수 있습니다. 같은 로그 그룹의 하나 이상의 흐름 로그에 동일한 네트워크 인터페이스가 있을 경우 로그 스트림은 하나로 병합됩니다. 한 흐름 로그에서는 거부된 트래픽을 캡처하고, 다른 흐름 로그에서는 허용된 트래픽을 캡처하도록 지정할 경우, 병합된 로그 스트림은 모든 트래픽을 캡처합니다.

서브넷이나 VPC에 대한 흐름 로그를 생성한 후 서브넷에서 더 많은 인스턴스를 시작할 경우, 새로운 네트워크 인터페이스에 대한 네트워크 트래픽이 기록되자마자 해당 네트워크 인터페이스에 대한 새로운 로그 스트림이 생성됩니다.

다른 AWS 서비스(예: Elastic Load Balancing, Amazon RDS, Amazon ElastiCache, Amazon Redshift, Amazon WorkSpaces)에서 생성한 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 하지만 이러한 서비스의 콘솔이나 API를 사용하여 흐름 로그를 생성할 수는 없습니다. Amazon EC2 콘솔이나 Amazon EC2 API를 사용해야 합니다. 마찬가지로 CloudWatch Logs 콘솔이나 API를 사용하여 네트워크 인터페이스에 대한 로그 스트림을 생성할 수 없습니다.

흐름 로그가 더 이상 필요하지 않을 경우 삭제할 수 있습니다. 흐름 로그를 삭제하면 리소스에 대한 흐름 로그 서비스가 비활성화되고, 새로운 흐름 로그 레코드 또는 로그 스트림이 생성되지 않습니다. 네트워크 인터페이스에 대한 기존의 흐름 로그 레코드나 로그 스트림은 삭제되지 않습니다. 기존의 로그 스트림을 삭제하려면 CloudWatch Logs 콘솔을 사용합니다. 흐름 로그를 삭제한 후 데이터 수집이 중단되기까지 몇 분 정도 시간이 걸릴 수 있습니다.

## 흐름 로그 제한

흐름 로그를 사용하려면 다음과 같은 제한 사항을 알아 두어야 합니다.

- EC2-Classic 플랫폼에 있는 네트워크 인터페이스에 대한 흐름 로그는 활성화할 수 없습니다. 여기에는 ClassicLink를 통해 VPC에 연결된 EC2-Classic 인스턴스가 포함됩니다.
- 피어 VPC가 본인의 계정이 아닌 한 본인의 VPC와 피어링된 VPC에 대해 흐름 로그를 활성화할 수 없습니다.
- 흐름 로그에 태그를 지정할 수 없습니다.
- 흐름 로그를 생성한 후에는 구성을 변경할 수 없습니다. 예를 들어 흐름 로그에 다른 IAM 역할을 연결할 수 없습니다. 대신에 흐름 로그를 삭제한 후 필요한 구성으로 새로운 흐름 로그를 생성할 수 있습니다.
- 흐름 로그 API 작업(`ec2:*FlowLogs`)은 리소스 수준 권한을 지원하지 않습니다. 흐름 로그 API 작업의 사용을 제어하는 IAM 정책을 만들려면 명령문에서 해당 리소스에 \* 와일드카드를 사용하여 모든 리소스를 작업에 사용할 수 있는 권한을 사용자에게 부여해야 합니다. 자세한 내용은 [Amazon VPC 리소스에 대한 액세스 제어 \(p. 130\)](#) 단원을 참조하십시오.
- 네트워크 인터페이스에 IPv4 주소가 여러 개 있고 트래픽이 보조 프라이빗 IPv4 주소로 전송되는 경우, 흐름 로그는 대상 IP 주소 필드에 주 프라이빗 IPv4 주소를 표시합니다.

흐름 로그는 모든 IP 트래픽 유형을 캡처하지는 않습니다. 다음 트래픽 유형은 기록되지 않습니다.

- 인스턴스가 Amazon DNS 서버에 연결할 때 생성한 트래픽. 자체 DNS 서버를 사용할 경우 DNS 서버에 대한 모든 트래픽은 기록됩니다.
- Amazon Windows 라이선스 인증을 위해 Windows 인스턴스에서 생성한 트래픽.
- 인스턴스 메타데이터를 위해 169.254.169.254와 주고받는 트래픽.
- DHCP 트래픽.
- 기본 VPC 라우터의 예약된 IP 주소로 보내는 트래픽. 자세한 내용은 [VPC 및 서브넷 크기 \(p. 63\)](#) 단원을 참조하십시오.

## 흐름 로그 레코드

흐름 로그 레코드는 흐름 로그에 네트워크 흐름을 나타냅니다. 각 레코드는 특정 캡처 기간 중 특정 5 튜플의 네트워크 흐름을 캡처합니다. 5 튜플은 인터넷 프로토콜(IP) 흐름의 소스, 대상, 프로토콜을 지정하는 5가지 값의 집합입니다. 캡처 기간은 흐름 로그 서비스에서 로그 레코드를 게시하기 전에 데이터를 집계하는 시간 기간입니다. 캡처 기간은 약 10분 정도 걸리며 최대 15분까지 걸릴 수 있습니다. 흐름 로그 레코드는 공백으로 구분된 문자열로, 다음과 같은 형식입니다.

```
version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes start end action log-status
```

필드	설명
version	VPC 흐름 로그 버전.
account-id	흐름 로그의 AWS 계정 ID.
interface-id	로그 스트림이 적용되는 네트워크 인터페이스 ID.

필드	설명
srcaddr	소스 IPv4 또는 IPv6 주소. 네트워크 인터페이스의 IPv4 주소는 항상 해당 프라이빗 IPv4 주소입니다.
dstaddr	대상 IPv4 또는 IPv6 주소. 네트워크 인터페이스의 IPv4 주소는 항상 해당 프라이빗 IPv4 주소입니다.
srcport	트래픽의 소스 포트
dstport	트래픽의 대상 포트
protocol	트래픽의 IANA 프로토콜 번호. 자세한 내용은 <a href="#">지정된 인터넷 프로토콜 번호</a> 를 참조하십시오.
packets	캡처 기간 중 전송된 패킷 수.
bytes	캡처 기간 중 전송된 바이트 수.
start	캡처 기간의 시작 시간(단위: Unix 초)
end	캡처 기간의 종료 시간(단위: Unix 초)
action	트래픽과 연결된 작업 <ul style="list-style-type: none"> <li>ACCEPT: 보안 그룹 또는 네트워크 ACL에서 허용한 트래픽입니다.</li> <li>REJECT: 보안 그룹 또는 네트워크 ACL에서 허용하지 않은 트래픽입니다.</li> </ul>
log-status	흐름 로그의 로깅 상태: <ul style="list-style-type: none"> <li>OK: 데이터가 정상적으로 CloudWatch Logs에 로깅됩니다.</li> <li>NODATA: 캡처 기간 중 네트워크 인터페이스에서 전송하거나 수신된 네트워크 트래픽이 없었습니다.</li> <li>SKIPDATA: 캡처 기간 중 일부 흐름 로그 레코드를 건너뛰었습니다. 내부 용량 제한 또는 내부 오류가 원인일 수 있습니다.</li> </ul>

필드가 특정 레코드에 해당되지 않을 경우 레코드에 '-' 기호가 표시됩니다.

흐름 로그 레코드에 대한 자세한 내용은 예: [흐름 로그 레코드 \(p. 150\)](#)을 참조하십시오.

흐름 로그 레코드는 CloudWatch Logs에서 수집한 다른 로그 이벤트처럼 사용할 수 있습니다. 로그 데이터 및 지표 필터 모니터링에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [로그 데이터 검색 및 필터링](#)을 참조하십시오. 흐름 로그에 대한 지표 필터 및 경보 설정의 예는 예: [흐름 로그에 대한 CloudWatch 지표 필터 및 경보 만들기 \(p. 152\)](#)를 참조하십시오.

## 흐름 로그에 대한 IAM 역할

흐름 로그와 연결된 IAM 역할에는 CloudWatch Logs의 지정된 로그 그룹에 흐름 로그를 게시할 권한이 있어야 합니다. IAM 역할에 연결된 IAM 정책에는 최소한 다음과 같은 권한이 포함되어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",

```



```
    "logs:DescribeLogStreams"  
  ],  
  "Effect": "Allow",  
  "Resource": "*" }  
}
```

흐름 로그 서비스가 역할을 수임할 수 있도록 역할에 신뢰 관계가 있는지 확인해야 합니다. IAM 콘솔에서 사용자 역할을 선택한 후 [Edit Trust Relationship]을 선택하여 신뢰 관계를 확인합니다.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "vpc-flow-logs.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

또는 아래의 절차를 따라 흐름 로그를 사용할 새로운 역할을 만들 수 있습니다.

## 본인 계정의 CloudWatch Logs 로그 그룹에 대한

흐름 로그에 대한 IAM 역할을 만들려면

1. <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 [Roles]를 선택한 후 [Create New Role]을 선택합니다.
3. 역할 이름(예: Flow-Logs-Role)을 입력한 후 [Next]를 선택합니다.
4. [Select Role Type] 페이지에서 [Amazon EC2] 옆의 [Select]를 선택합니다.
5. [Attach Policy] 페이지에서 [Next Step]을 선택합니다.
6. [Review] 페이지에서 사용자 역할에 대한 ARN을 확인합니다. 사용자의 흐름 로그를 만들 때 이 ARN이 필요합니다. 준비가 됐으면 [Create Role]을 선택합니다.
7. 역할 이름을 선택합니다. [Permissions]에서 [Inline Policies] 섹션을 확장한 후 [click here]를 선택합니다.
8. [Custom Policy]를 선택한 후 [Select]를 선택합니다.
9. 위의 [흐름 로그에 대한 IAM 역할 \(p. 146\)](#) 단원에서 첫 번째 정책을 복사한 후 [Policy Document] 창에 붙여 넣습니다. [Policy Name] 필드에 사용자 정책 이름을 입력한 후 [Apply Policy]를 선택합니다.
10. 위의 [흐름 로그에 대한 IAM 역할 \(p. 146\)](#) 섹션에서 두 번째 정책(신뢰 관계)을 복사한 후 [Edit Trust Relationship]을 선택합니다. 기존 정책 문서를 삭제하고 새 문서를 붙여 넣습니다. 완료했으면 [Update Trust Policy]를 선택합니다.

## 흐름 로그 작업

Amazon EC2, Amazon VPC, CloudWatch 콘솔을 사용하여 흐름 로그에 대한 작업을 수행할 수 있습니다.

항목

- [흐름 로그 생성 \(p. 148\)](#)
- [흐름 로그 확인 \(p. 148\)](#)
- [흐름 로그 삭제 \(p. 149\)](#)

## 흐름 로그 생성

VPC 페이지, Amazon VPC 콘솔의 서브넷 페이지 또는 Amazon EC2 콘솔의 네트워크 인터페이스 페이지에서 흐름 로그를 만들 수 있습니다.

네트워크 인터페이스에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택하고 [Flow Logs] 탭을 선택한 후 [Create Flow Log]를 선택합니다.
4. 대화 상자에 다음 정보를 입력합니다. 완료했으면 [Create Flow Log]를 선택합니다.
  - Filter: 거부된 트래픽, 허용된 트래픽 또는 모든 트래픽 중 흐름 로그가 캡처할 트래픽을 선택합니다.
  - Role: CloudWatch Logs에 로그를 게시할 권한이 있는 IAM 역할의 이름을 지정합니다.
  - Destination Log Group: 흐름 로그를 게시할 CloudWatch Logs의 로그 그룹 이름을 입력합니다. 기존의 로그 그룹을 사용하거나, 새로운 로그 그룹의 이름을 입력할 수 있습니다. 새 이름을 입력하면 로그 그룹이 생성됩니다.

VPC 또는 서브넷에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs] 또는 [Subnets]를 선택합니다.
3. VPC 또는 서브넷을 선택하고 [Flow Logs] 탭을 선택한 후 [Create Flow Log]를 선택합니다.

### Note

여러 VPC에 대한 흐름 로그를 만들려면 VPC를 선택한 후 [Actions] 메뉴에서 [Create Flow Log]를 선택합니다. 여러 서브넷에 대한 흐름 로그를 만들려면 서브넷을 선택한 후 [Subnet Actions] 메뉴에서 [Create Flow Log]를 선택합니다.

4. 대화 상자에 다음 정보를 입력합니다. 완료했으면 [Create Flow Log]를 선택합니다.
  - Filter: 거부된 트래픽, 허용된 트래픽 또는 모든 트래픽 중 흐름 로그가 캡처할 트래픽을 선택합니다.
  - Role: CloudWatch Logs에 로그를 게시할 권한이 있는 IAM 역할의 이름을 지정합니다.
  - Destination Log Group: 흐름 로그를 게시할 CloudWatch Logs의 로그 그룹 이름을 입력합니다. 기존의 로그 그룹을 사용하거나, 새로운 로그 그룹의 이름을 입력할 수 있습니다. 새 이름을 입력하면 로그 그룹이 생성됩니다.

## 흐름 로그 확인

Amazon EC2 및 Amazon VPC 콘솔에서 특정 리소스에 대한 [Flow Logs] 탭을 확인하여 흐름 로그에 대한 정보를 확인할 수 있습니다. 리소스를 선택하면 해당 리소스의 모든 흐름 로그가 나열됩니다. 흐름 로그의 ID, 흐름 로그 구성, 흐름 로그 상태에 대한 정보가 표시됩니다.

네트워크 인터페이스에 대한 흐름 로그 정보를 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 네트워크 인터페이스를 선택한 후 [Flow Logs] 탭을 선택합니다. 흐름 로그에 대한 정보가 탭에 표시됩니다.

VPC 또는 서브넷에 대한 흐름 로그 정보를 보려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 [Your VPCs] 또는 [Subnets]를 선택합니다.
3. VPC 또는 서브넷을 선택한 후 [Flow Logs] 탭을 선택합니다. 흐름 로그에 대한 정보가 탭에 표시됩니다.

CloudWatch Logs 콘솔을 사용하여 흐름 로그 레코드를 볼 수 있습니다. 흐름 로그를 생성한 후 콘솔에서 흐름 로그를 보려면 몇 분 정도 지나야 할 수 있습니다.

흐름 로그의 레코드를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Logs]를 선택합니다.
3. 흐름 로그를 포함하는 로그 그룹의 이름을 선택합니다.
4. 각 네트워크 인터페이스에 대한 로그 스트림 목록이 표시됩니다. 흐름 로그 레코드를 보려는 네트워크 인터페이스의 ID가 있는 로그 스트림의 이름을 선택합니다. 흐름 로그에 대한 자세한 내용은 [흐름 로그 레코드 \(p. 145\)](#)를 참조하십시오.

## 흐름 로그 삭제

Amazon EC2 및 Amazon VPC 콘솔을 사용하여 흐름 로그를 삭제할 수 있습니다.

### Note

흐름 로그를 삭제하면 리소스에 대한 흐름 로그 서비스가 비활성화됩니다. 네트워크 인터페이스에 대한 로그 스트림을 삭제하려면 CloudWatch Logs 콘솔을 사용하십시오.

네트워크 인터페이스에 대한 흐름 로그를 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택한 후 네트워크 인터페이스를 선택합니다.
3. [Flow Logs] 탭을 선택한 후 삭제할 흐름 로그의 삭제 버튼(x 표시)을 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

VPC 또는 서브넷에 대한 흐름 로그를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택하거나 사용자의 [Subnets]를 선택한 후 리소스를 선택합니다.
3. [Flow Logs] 탭을 선택한 후 삭제할 흐름 로그의 삭제 버튼(x 표시)을 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

## 문제 해결

### 불완전한 흐름 로그 레코드

사용자의 흐름 로그 레코드가 불완전하거나 더 이상 게시되지 않는 경우, CloudWatch Logs 로그 그룹에 흐름 로그를 전달하는 데 문제가 있을 수 있습니다. Amazon EC2 콘솔 또는 Amazon VPC 콘솔에서 해당 리소스에 대한 [Flow Logs] 탭을 확인하십시오. 자세한 내용은 [흐름 로그 확인 \(p. 148\)](#) 단원을 참조하십시오. 흐름 로그 테이블의 [Status] 열에는 오류가 표시됩니다. 또는 `describe-flow-logs` 명령을 사용하여 `DeliverLogsErrorMessage` 필드에 반환된 값을 확인하십시오. 다음 중 하나의 오류가 표시될 수 있습니다.

- `Rate limited`: 이 오류는 CloudWatch 로그 조절이 적용된 경우 즉, 네트워크 인터페이스에 대한 흐름 로그 레코드의 수가 특정 시간 범위 내에 게시될 수 있는 최대 레코드의 수보다 많을 경우에 발생할 수 있습니다. 이 오류는 만들 수 있는 CloudWatch Logs 로그 그룹 수의 제한에 도달한 경우에도 발생할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 제한](#)을 참조하십시오.

- `Access error`: 흐름 로그에 대한 IAM 역할에 CloudWatch 로그 그룹에 흐름 로그를 게시할 권한이 없습니다. 자세한 내용은 [흐름 로그에 대한 IAM 역할 \(p. 146\)](#) 단원을 참조하십시오.
- `Unknown error`: 흐름 로그 서비스에서 내부 오류가 발생했습니다.

## 흐름 로그가 활성화되었지만 흐름 로그 레코드 또는 로그 그룹이 없음

흐름 로그를 생성하면 Amazon VPC 또는 Amazon EC2 콘솔에 흐름 로그가 `Active`로 표시됩니다. 하지만 CloudWatch Logs에서 어떠한 로그 스트림도 볼 수 없거나 CloudWatch Logs 로그 그룹이 생성되지 않았습니다. 원인은 다음 중 하나일 수 있습니다.

- 흐름 로그가 아직 생성되는 중입니다. 경우에 따라 흐름 로그를 생성한 후 로그 그룹이 생성되고 데이터가 표시되기까지 수십 분 정도 걸릴 수 있습니다.
- 네트워크 인터페이스에 대해 기록된 트래픽이 아직 없습니다. CloudWatch Logs의 로그 그룹은 트래픽이 기록될 경우에만 생성됩니다.

## API 및 CLI 개요

이 페이지에서 설명한 작업은 명령줄이나 API를 사용하여 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

### 흐름 로그 생성

- `create-flow-logs`(AWS CLI)
- `New-EC2FlowLogs`(Windows PowerShell용 AWS 도구)
- `CreateFlowLogs`(Amazon EC2 Query API)

### 흐름 로그 설명

- `describe-flow-logs`(AWS CLI)
- `Get-EC2FlowLogs`(Windows PowerShell용 AWS 도구)
- `DescribeFlowLogs`(Amazon EC2 Query API)

### 흐름 로그 레코드 확인(로그 이벤트)

- `get-log-events`(AWS CLI)
- `Get-CWLLogEvents`(Windows PowerShell용 AWS 도구)
- `GetLogEvents`(CloudWatch API)

### 흐름 로그 삭제

- `delete-flow-logs`(AWS CLI)
- `Remove-EC2FlowLogs`(Windows PowerShell용 AWS 도구)
- `DeleteFlowLogs`(Amazon EC2 Query API)

## 예: 흐름 로그 레코드

허용 및 거부된 트래픽에 대한 흐름 로그 레코드

다음은 계정 123456789010에서 네트워크 인터페이스 eni-abc123de에 대한 SSH 트래픽(대상 포트 22, TCP 프로토콜)이 허용된 흐름 로그 레코드의 예입니다.

```
2 123456789010 eni-abc123de 172.31.16.139 172.31.16.21 20641 22 6 20 4249 1418530010
1418530070 ACCEPT OK
```

다음은 계정 123456789010에서 네트워크 인터페이스 eni-abc123de에 대한 RDP 트래픽(대상 포트 3389, TCP 프로토콜)이 거부된 흐름 로그 레코드의 예입니다.

```
2 123456789010 eni-abc123de 172.31.9.69 172.31.9.12 49761 3389 6 20 4249 1418530010
1418530070 REJECT OK
```

데이터가 없으며 건너뛴 기록에 대한 흐름 로그 레코드

다음은 캡처 기간 중 아무런 데이터도 기록되지 않은 흐름 로그 레코드의 예입니다.

```
2 123456789010 eni-1a2b3c4d - - - - - 1431280876 1431280934 - NODATA
```

다음은 캡처 기간 중 기록을 건너뛴 흐름 로그 레코드의 예입니다.

```
2 123456789010 eni-4b118871 - - - - - 1431280876 1431280934 - SKIPDATA
```

#### 보안 그룹 및 네트워크 ACL 규칙

너무 제한적이거나 허용적인 보안 그룹 규칙 또는 네트워크 ACL 규칙을 진단하기 위해 흐름 로그를 사용할 경우 이러한 리소스의 상태 저장 여부를 알아야 합니다. 보안 그룹은 상태가 저장됩니다. 이는 보안 그룹의 규칙에서 허용하지 않더라도 허용된 트래픽에 대한 응답도 가능하다는 의미입니다. 반대로 네트워크 ACL은 상태를 저장하지 않으므로 허용된 트래픽에 대한 응답은 네트워크 ACL 규칙을 따릅니다.

예를 들어 홈 컴퓨터(IP 주소: 203.0.113.12)에서 인스턴스(네트워크 인터페이스의 프라이빗 IP 주소: 172.31.16.139)로 ping 명령을 사용합니다. 사용자 보안 그룹의 인바운드 규칙은 ICMP 트래픽을 허용하고 아웃바운드 규칙은 이를 허용하지 않습니다. 하지만 보안 그룹은 상태를 저장하기 때문에 인스턴스로부터의 응답 ping이 허용됩니다. 네트워크 ACL은 인바운드 ICMP 트래픽을 허용하지만 아웃바운드 ICMP 트래픽은 허용하지 않습니다. 왜냐하면 네트워크 ACL은 상태를 저장하지 않아서 응답 ping이 홈 컴퓨터에 도달하지 않기 때문입니다. 흐름 로그에서 이는 3가지 흐름 로그 항목으로 표시됩니다. 보안 그룹이 허용한 요청 ping 과 응답 ping에 대한 2가지 ACCEPT 항목이 있고, 네트워크 ACL이 거부한 응답 ping에 대한 1가지 REJECT 항목이 있습니다.

```
2 123456789010 eni-1235b8ca 203.0.113.12 172.31.16.139 0 0 1 8 672 1432917027 1432917142
ACCEPT OK
```

```
2 123456789010 eni-1235b8ca 172.31.16.139 203.0.113.12 0 0 1 4 336 1432917027 1432917082
ACCEPT OK
```

```
2 123456789010 eni-1235b8ca 172.31.16.139 203.0.113.12 0 0 1 4 336 1432917094 1432917142
REJECT OK
```

#### IPv6 트래픽에 대한 흐름 로그 레코드

다음은 IPv6 주소 2001:db8:1234:a100:8d6e:3477:df66:f105에서 계정 123456789010의 네트워크 인터페이스 eni-f41c42bf에 대한 SSH 트래픽(포트 22)이 허용된 흐름 로그 레코드의 예입니다.

```
2 123456789010 eni-f41c42bf 2001:db8:1234:a100:8d6e:3477:df66:f105
2001:db8:1234:a102:3304:8879:34cf:4071 34892 22 6 54 8855 1477913708 1477913820 ACCEPT OK
```

## 예 :흐름 로그에 대한 CloudWatch 지표 필터 및 경고 만들기

이 예에서는 eni-1a2b3c4d에 대한 흐름 로그를 사용합니다. TCP 포트 22(SSH)를 거쳐 인스턴스에 연결하려는 시도가 한 시간 내에 10번 이상 거부된 경우 이를 알려 주는 알림을 만들 수 있습니다. 우선 경보를 만들려는 트래픽의 패턴과 일치하는 지표 필터를 만들어야 합니다. 그런 다음 지표 필터에 대한 경보를 만듭니다.

거부된 SSH 트래픽에 대한 지표 필터와 필터에 대한 경보를 만들려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Logs]를 선택하고, 흐름 로그에 대한 로그 그룹을 선택한 후 [Create Metric Filter]를 선택합니다.
3. [Filter Pattern] 필드에 다음을 입력합니다.

```
[version, account, eni, source, destination, srcport, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]
```

4. [Select Log Data to Test] 목록에서 네트워크 인터페이스에 대한 로그 스트림을 선택합니다. 원할 경우 [Test Pattern]을 선택하여 필터 패턴과 일치하는 로그 데이터 행을 볼 수 있습니다. 준비가 됐으면 [Assign Metric]을 선택합니다.
5. 지표 네임스페이스, 지표 이름을 제공하고 지표 값을 1로 설정합니다. 완료했으면 [Create Filter]를 선택합니다.
6. 탐색 창에서 [Alarms]를 선택한 후 [Create Alarm]을 선택합니다.
7. [Custom Metrics] 섹션에서, 앞에서 만든 지표 필터에 대한 네임스페이스를 선택합니다.

### Note

새로운 지표가 콘솔에 표시될 때까지 몇 분 정도 걸릴 수 있습니다.

8. 만든 지표 이름을 선택한 후 [Next]를 선택합니다.
9. 경고 이름 및 설명을 입력합니다. [is] 필드에서 [>=]를 선택한 후 [10]를 입력합니다. for 필드에서 연속 기간에 대해 기본값 1을 유지합니다.
10. [Period] 목록에서 [1 Hour]를 선택한 후 [Statistic] 목록에서 [Sum]을 선택합니다. Sum 통계는 지정된 기간 동안 데이터 포인트의 총 수를 캡처합니다.
11. [Actions] 섹션에서 기존 목록에 알림을 보내거나 경보가 트리거 될 때 알림을 받을 이메일 주소를 입력할 수 있습니다. 완료했으면 [Create Alarm]을 선택합니다.

## VPC 네트워킹 구성 요소

다음 구성 요소에 따라 VPC의 네트워킹을 구성할 수 있습니다.

- [네트워크 인터페이스](#) (p. 153)
- [라우팅 테이블](#) (p. 154)
- [인터넷 게이트웨이](#) (p. 165)
- [외부 전용 인터넷 게이트웨이](#) (p. 170)
- [DHCP 옵션 세트](#) (p. 189)
- [DNS](#) (p. 194)
- [탄력적 IP 주소](#) (p. 197)
- [VPC 엔드포인트](#) (p. 200)
- [NAT](#) (p. 172)
- [VPC 피어링](#) (p. 197)
- [\(p. 211\)](#)

## 탄력적 네트워크 인터페이스

탄력적 네트워크 인터페이스(이 문서에서는 네트워크 인터페이스로 표시)는 다음과 같은 속성을 지닌 가상 네트워크 인터페이스입니다.

- 주 프라이빗 IPv4 주소
- 하나 이상의 보조 프라이빗 IPv4 주소
- 프라이빗 IPv4 주소당 한 개의 탄력적 IP 주소
- 인스턴스를 시작할 때 eth0에 대한 네트워크 인터페이스에 자동 할당할 수 있는 퍼블릭 IPv4 주소 한 개
- 하나 이상의 IPv6 주소
- 하나 이상의 보안 그룹
- MAC 주소
- 원본/대상 확인 플래그

- 설명

네트워크 인터페이스를 만들고, 인스턴스에 연결하고, 인스턴스에서 분리한 후 다른 인스턴스에 연결할 수 있습니다. 네트워크 인터페이스를 인스턴스에 연결하거나 인스턴스에서 분리한 후 다른 인스턴스에 다시 연결하면 그 속성은 해당 네트워크 인터페이스를 따릅니다. 네트워크 인터페이스를 인스턴스 간에 이동하면 네트워크 트래픽이 새 인스턴스로 리디렉션됩니다.

VPC의 각 인스턴스마다 기본 네트워크 인터페이스(주 네트워크 인터페이스)가 있어서 VPC의 IPv4 주소 범위에 속하는 프라이빗 IPv4 주소가 이 인터페이스에 할당됩니다. 주 네트워크 인터페이스는 인스턴스에서 분리할 수 없습니다. 추가 네트워크 인터페이스를 만들어 VPC의 모든 인스턴스에 연결할 수 있습니다. 연결 가능한 네트워크 인터페이스의 개수는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 유형별/네트워크 인터페이스당 IP 주소](#) 단원을 참조하십시오.

다음을 수행하려는 경우 여러 네트워크 인터페이스를 하나의 인스턴스에 연결하면 유용합니다.

- 관리 네트워크 생성
- VPC에서 네트워크 및 보안 어플라이언스 사용
- 별도의 서브넷에 워크로드/역할이 있는 이중 홈 인스턴스 생성
- 저예산 고가용성 솔루션 생성

네트워크 인터페이스에 대한 자세한 내용 및 Amazon EC2 콘솔을 사용한 네트워크 인터페이스 작업에 대한 지침은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [탄력적 네트워크 인터페이스](#) 단원을 참조하십시오.

## 라우팅 테이블

라우팅 테이블에는 네트워크 트래픽을 전달할 위치를 결정하는 데 사용되는 라우팅이라는 규칙 집합이 포함되어 있습니다.

VPC의 각 서브넷을 라우팅 테이블에 연결해야 합니다. 테이블에서는 서브넷에 대한 라우팅을 제어합니다. 서브넷을 한 번에 하나의 라우팅 테이블에만 연결할 수 있지만, 여러 서브넷을 동일한 라우팅 테이블에 연결할 수 있습니다.

### 항목

- [라우팅 테이블 기본 사항](#) (p. 154)
- [라우팅 우선순위](#) (p. 156)
- [라우팅 옵션](#) (p. 157)
- [라우팅 테이블 작업](#) (p. 160)
- [API 및 명령 개요](#) (p. 163)

## 라우팅 테이블 기본 사항

다음은 라우팅 테이블에 대해 알아야 할 기본 사항입니다.

- 사용자의 VPC에 암시적인 라우터가 있습니다.
- VPC는 수정할 수 있는 기본 라우팅 테이블과 함께 자동으로 제공됩니다.
- VPC에 사용할 추가 사용자 지정 라우팅 테이블을 만들 수 있습니다.
- 서브넷에 대한 라우팅을 제어하는 라우팅 테이블에 각 서브넷을 연결해야 합니다. 서브넷을 특정 라우팅 테이블에 명시적으로 연결하지 않을 경우 서브넷은 기본 라우팅 테이블에 암시적으로 연결됩니다.
- 기본 라우팅 테이블은 삭제할 수 없지만, 기본 라우팅 테이블을 사용자가 만든 사용자 지정 테이블로 바꿀 수 있습니다(따라서 이 테이블이 각각의 새로운 서브넷이 연결되는 기본 테이블이 됨).



- 테이블에 있는 각각의 라우팅은 대상 CIDR과 대상을 지정합니다(예: 외부 회사 네트워크 172.16.0.0/12로 향하는 트래픽은 가상 프라이빗 게이트웨이에 대해 대상으로 지정됨). Amazon은 트래픽과 일치하는, 가장 구체적인 라우팅을 사용하여 트래픽의 라우팅 방법을 결정합니다.
- IPv4 및 IPv6 CIDR 블록은 별도로 취급됩니다. 예를 들어 대상 CIDR이 0.0.0.0/0인 경로(모두 IPv4 주소)에는 모든 IPv6 주소가 자동으로 포함되지 않습니다. 모든 IPv6 주소에 대해 대상 CIDR이 ::/0인 경로를 생성해야 합니다.
- 모든 라우팅 테이블에는 IPv4를 통한 VPC 내부 통신을 위한 로컬 경로가 포함되어 있습니다. IPv6 CIDR 블록을 VPC와 연결한 경우, 모든 라우팅 테이블에는 IPv6를 통한 VPC 내부 통신을 위한 로컬 경로도 포함되어 있습니다. 이 경로는 수정하거나 삭제할 수 없습니다.
- VPC에 인터넷 게이트웨이, 외부 전용 인터넷 게이트웨이, 가상 프라이빗 게이트웨이, NAT 디바이스, 피어링 연결 또는 VPC 엔드포인트를 추가할 경우, 그러한 게이트웨이나 연결을 사용하는 모든 서브넷에 대해 라우팅 테이블을 업데이트해야 합니다.
- VPC당 생성할 수 있는 라우팅 테이블의 수와 라우팅 테이블당 추가할 수 있는 라우팅의 수에는 제한이 있습니다. 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#)를 참조하십시오.

## 기본 라우팅 테이블

VPC를 만들면 기본 라우팅 테이블이 자동으로 생성됩니다. Amazon VPC 콘솔의 [Route Tables] 페이지의 [Main] 열에서 [Yes]를 찾아 VPC에 대한 기본 라우팅 테이블을 볼 수 있습니다. 기본 라우팅 테이블은 다른 라우팅 테이블과 명시적으로 연결되지 않은 모든 서브넷에 대한 라우팅을 제어합니다. 기본 라우팅 테이블에서 라우팅을 추가 및 제거하고 수정할 수 있습니다.

서브넷이 기본 라우팅 테이블에 명시적으로 연결되어 있지 않은 경우에도 서브넷을 기본 라우팅에 명시적으로 연결할 수 있습니다. 기본 라우팅 테이블을 변경할 경우 이렇게 할 수 있으며, 그러면 추가되는 새로운 서브넷이나 다른 라우팅 테이블에 명시적으로 연결되지 않은 서브넷에 대한 기본 라우팅 테이블이 변경됩니다. 자세한 내용은 [기본 라우팅 테이블 바꾸기 \(p. 163\)](#)를 참조하십시오.

## 사용자 지정 라우팅 테이블

사용자의 VPC에 기본 테이블 이외의 라우팅 테이블이 있을 수 있습니다. VPC를 보호하는 한 가지 방법은 기본 라우팅 테이블을 (로컬 경로만 있는) 원래의 기본 상태로 두고, 사용자가 새로 생성하는 각 서브넷을 자신이 생성한 사용자 지정 라우팅 테이블 중 하나와 명시적으로 연결하는 것입니다. 이렇게 하면 각 서브넷이 아웃바운드 트래픽으로 어떻게 라우팅되는지를 명시적으로 제어할 수 있습니다.

다음 다이어그램에서는 인터넷 게이트웨이와 가상 프라이빗 게이트웨이가 모두 있는 VPC는 물론이고, 퍼블릭 서브넷과 VPN 전용 서브넷에 대한 라우팅도 보여줍니다. 기본 라우팅 테이블은 VPC와 함께 제공되었고, 여기에 VPN 전용 서브넷에 대한 경로도 있습니다. 사용자 지정 라우팅 테이블은 퍼블릭 서브넷과 연결되었습니다. 사용자 지정 라우팅 테이블에는 인터넷 게이트웨이를 통한 라우팅이 지정되어 있습니다(목적지는 0.0.0.0/0, 대상은 인터넷 게이트웨이).

이 VPC에 새 서브넷을 생성하면 서브넷이 트래픽을 가상 프라이빗 게이트웨이로 라우팅하는 기본 라우팅 테이블과 자동으로 연결됩니다. 반대 구성(인터넷 게이트웨이에 대한 경로가 있는 기본 라우팅 테이블과 가상 프라이빗 게이트웨이에 대한 경로가 있는 사용자 지정 라우팅 테이블)을 설정하면 새 서브넷은 인터넷 게이트웨이에 대한 라우팅을 자동으로 갖게 됩니다.

## 라우팅 테이블 연결

VPC 콘솔은 각 라우팅 테이블에 명시적으로 연결된 서브넷의 수를 보여 주며, 기본 라우팅 테이블과 암시적으로 연결된 서브넷에 대한 정보를 제공합니다. 자세한 내용은 [테이블과 명시적으로 연결되어 있는 서브넷 확인 \(p. 161\)](#)을 참조하십시오.

서브넷을 기본 라우팅 테이블과 암시적 또는 명시적으로 연결할 수 있습니다. 기본 라우팅 테이블을 바꿀 경우 서브넷이 일시적으로 기본 라우팅 테이블에 명시적으로 연결될 수도 있겠지만, 일반적으로는 기본 라우팅 테이블에 명시적으로 연결되지 않습니다.

기본 라우팅 테이블을 변경할 수도 있지만, 트래픽 중단을 방지하기 위해 우선은 사용자 지정 라우팅 테이블을 사용하여 경로 변경을 테스트해 볼 수 있습니다. 테스트 결과에 만족하면 기본 라우팅 테이블을 새로운 사용자 지정 테이블로 바꿉니다.

다음 다이어그램에서는 기본 라우팅 테이블(라우팅 테이블 A)과 암시적으로 연결되는 두 개의 서브넷이 있는 VPC와 어떤 서브넷과도 연결되지 않는 사용자 지정 라우팅 테이블(라우팅 테이블 B)을 보여줍니다.

서브넷 2와 라우팅 테이블 B 사이에 명시적 연결을 만들 수 있습니다.

라우팅 테이블 B를 테스트한 후, 이 테이블을 기본 라우팅 테이블로 만들 수 있습니다. 서브넷 2는 여전히 라우팅 테이블 B와 명시적으로 연결되고, 라우팅 테이블 B가 새로운 기본 라우팅 테이블이기 때문에 서브넷 1은 라우팅 테이블 B와 암시적으로 연결됩니다. 라우팅 테이블 A는 더 이상 사용되지 않습니다.

라우팅 테이블 B에서 서브넷 2의 연결을 끊더라도 서브넷 2와 라우팅 테이블 B 사이에는 여전히 암시적 연결이 존재합니다. 라우팅 테이블 A가 더 이상 필요하지 않으면 삭제할 수 있습니다.

## 라우팅 우선순위

Amazon은 라우팅 테이블에서 LPM(Longest Prefix Match)을 통해 트래픽과 일치하는, 가장 구체적인 라우팅을 사용하여 트래픽의 라우팅 방법을 결정합니다.

IPv4 및 IPv6 주소나 CIDR 블록에 대한 경로는 서로 독립적입니다. 따라서 Amazon은 IPv4 트래픽 또는 IPv6 트래픽과 일치하는 가장 구체적인 경로를 사용하여 트래픽 라우팅 방법을 결정합니다.

예를 들어 다음 라우팅 테이블에는 인터넷 게이트웨이를 가리키는 IPv4 인터넷 트래픽(0.0.0.0/0)에 대한 경로와 피어링 연결(pcx-1a2b3c4d)을 가리키는 172.31.0.0/16 IPv4 트래픽에 대한 경로가 있습니다. 서브넷으로부터 수신되는 트래픽 중 172.31.0.0/16 IP 주소 범위가 목적지가 아닌 트래픽은 피어링 연결을 사용합니다. 이 라우팅이 인터넷 게이트웨이에 대한 라우팅보다 구체적이기 때문입니다. VPC(10.0.0.0/16) 내에서 대상으로 전송되는 트래픽은 Local 라우팅이 적용되며 따라서 VPC 내에서 라우팅됩니다. 서브넷으로부터의 다른 모든 트래픽은 인터넷 게이트웨이를 사용합니다.

목적지	Target
10.0.0.0/16	로컬
172.31.0.0/16	pcx-1a2b1a2b
0.0.0.0/0	igw-11aa22bb

가상 프라이빗 게이트웨이를 VPC에 연결하고 라우팅 테이블에서 경로 전파를 활성화한 경우, VPN 연결을 나타내는 경로는 라우팅 테이블에 전파된 경로로 자동으로 나타납니다. 다음이 적용됩니다.

- VPN 연결 또는 AWS Direct Connect 연결에서 전파된 경로가 VPC의 로컬 경로와 중첩되는 경우, 전파된 경로가 더 특정하더라도 로컬 경로가 가장 우선적으로 적용됩니다.
- VPN 연결 또는 AWS Direct Connect 연결에서 전파된 경로에 다른 기존 정적 경로와 동일한 대상 CIDR 블록이 있는 경우(가장 긴 접두사 일치 항목이 적용될 수 없음) 대상이 인터넷 게이트웨이, 가상 프라이빗 게이트웨이, 네트워크 인터페이스, 인스턴스 ID, VPC 피어링 연결, NAT 게이트웨이 또는 VPC 종단점인 정적 경로가 우선적으로 적용됩니다.

VPN 연결에 겹치는 라우팅이 존재하고, 가장 긴 접두사 일치를 적용할 수 없으면 VPC 연결에서 가장 선호하는 경로부터 가장 선호하지 않은 경로의 순으로 다음과 같이 우선순위가 지정됩니다.

- AWS Direct Connect 연결로부터의 BGP 전파 라우팅
- VPN 연결에 수동으로 추가된 고정 라우팅
- VPN 연결로부터의 BGP 전파 라우팅

이 예에서 라우팅 테이블에는 인터넷 게이트웨이의 고정 라우팅(수동으로 추가됨)과 가상 프라이빗 게이트웨이를 전파되는 라우팅이 있습니다. 두 라우팅은 모두 목적지가 172.31.0.0/24입니다. 이 경우 목적지가 172.31.0.0/24인 모든 트래픽은 인터넷 게이트웨이를 라우팅됩니다. 이 라우팅은 고정 라우팅이므로 전파된 라우팅보다 우선합니다.

목적지	Target
10.0.0.0/16	로컬
172.31.0.0/24	vgw-1a2b3c4d(전파됨)
172.31.0.0/24	igw-11aa22bb

이 예시에서 IPv6 CIDR 블록은 VPC와 연결되어 있습니다. 라우팅 테이블에서 VPC(2001:db8:1234:1a00::/56) 내부에서 대상으로 전송되는 IPv6 트래픽은 Local 경로가 적용되며 따라서 VPC 내에서 라우팅됩니다. 라우팅 테이블에는 피어링 연결(pcx-1a2b3c4d)을 가리키는 172.31.0.0/16 IPv4 트래픽에 대한 경로, 인터넷 게이트웨이를 가리키는 모든 IPv4 트래픽(0.0.0.0/0)에 대한 경로, 그리고 외부 전용 인터넷 게이트웨이를 가리키는 모든 IPv6 트래픽(::/0)에 대한 경로가 있습니다. IPv4 및 IPv6 트래픽은 별도로 취급됩니다. 따라서 모든 IPv6 트래픽(VPC 내부의 트래픽 제외)은 외부 전용 인터넷 게이트웨이를 라우팅됩니다.

목적지	Target
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
172.31.0.0/16	pcx-1a2b1a2b
0.0.0.0/0	igw-11aa22bb
::/0	eigw-aabb1122

## 라우팅 옵션

다음 주제에서는 VPC의 특정 게이트웨이 또는 연결을 위한 라우팅을 설명합니다.

### 항목

- 인터넷 게이트웨이를 위한 라우팅 테이블 (p. 157)
- NAT 디바이스를 위한 라우팅 테이블 (p. 158)
- 가상 프라이빗 게이트웨이를 위한 라우팅 테이블 (p. 158)
- VPC 피어링 연결을 위한 라우팅 테이블 (p. 158)
- 에 대한 라우팅 테이블 (p. 159)
- VPC 종단점을 위한 라우팅 테이블 (p. 160)
- 외부 전용 인터넷 게이트웨이에 대한 라우팅 테이블 (p. 160)

## 인터넷 게이트웨이를 위한 라우팅 테이블

인터넷 게이트웨이에 경로를 추가하여 서브넷을 퍼블릭 서브넷으로 만들 수 있습니다. 이렇게 하려면 인터넷 게이트웨이를 만들어 VPC에 연결한 다음, IPv4 트래픽에 대한 목적지가 0.0.0.0/0인 경로나 IPv6 트래픽에 대한 목적지가 ::/0인 경로를 추가하고, 인터넷 게이트웨이 ID(igw-xxxxxxx)의 대상을 추가합니다. 자세한 내용은 [인터넷 게이트웨이 \(p. 165\)](#) 단원을 참조하십시오.

## NAT 디바이스를 위한 라우팅 테이블

프라이빗 서브넷의 인스턴스를 인터넷에 연결하려면 NAT 게이트웨이를 만들거나 퍼블릭 서브넷에서 NAT 인스턴스를 시작한 다음, NAT 디바이스로 IPv4 인터넷 트래픽(0.0.0.0/0)을 라우팅하는 프라이빗 서브넷에 대해 경로를 추가합니다. 자세한 내용은 [NAT 게이트웨이 \(p. 173\)](#) 및 [NAT 인스턴스 \(p. 182\)](#) 섹션을 참조하십시오. NAT 디바이스는 IPv6 트래픽에는 사용할 수 없습니다.

## 가상 프라이빗 게이트웨이를 위한 라우팅 테이블

AWS 하드웨어 VPN 연결을 사용하여 VPC의 인스턴스를 사용자의 네트워크와 통신하도록 할 수 있습니다. 이렇게 하려면 가상 프라이빗 게이트웨이를 만들고 VPC에 연결한 다음, 목적지가 사용자의 네트워크이며 대상이 가상 프라이빗 게이트웨이(vgw-xxxxxxx)인 경로를 추가합니다. 그런 다음 VPN 연결을 만들고 구성할 수 있습니다. 자세한 내용은 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#) 단원을 참조하십시오.

현재 VPN 연결을 통한 IPv6 트래픽은 지원하지 않습니다. 그러나 가상 프라이빗 게이트웨이를 통해 AWS Direct Connect 연결로 라우팅되는 IPv6 트래픽은 지원합니다. 자세한 내용은 [AWS Direct Connect 사용 설명서](#)를 참조하십시오.

## VPC 피어링 연결을 위한 라우팅 테이블

VPC 피어링 연결은 프라이빗 IPv4 주소를 사용하여 두 VPC 간에 트래픽을 라우팅할 수 있게 해주는 두 VPC 사이의 네트워킹 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다.

VPC 피어링 연결에서 VPC 간에 트래픽을 라우팅할 수 있도록 하려면, 피어링 연결에서 다른 VPC의 CIDR 블록 전부 또는 일부에 액세스하기 위해 VPC 피어링 연결을 가리키는 VPC의 라우팅 테이블 중 한 개 이상에 경로를 추가해야 합니다. 마찬가지로, 다른 VPC의 소유자는 트래픽을 다시 사용자의 VPC로 라우팅하기 위해 소유자 자신의 VPC 라우팅 테이블에 경로를 추가해야 합니다.

예를 들어 다음과 같은 정보를 가진 두 VPC 사이에 VPC 피어링 연결(pcx-1a2b1a2b)이 있다고 합시다.

- VPC A: vpc-1111aaaa, CIDR 블록은 10.0.0.0/16
- VPC B: vpc-2222bbbb, CIDR 블록은 172.31.0.0/16

VPC 간에 트래픽을 활성화하고 어느 한 VPC의 전체 IPv4 CIDR 블록에 대한 액세스를 허용하기 위해 VPC A의 라우팅 테이블은 다음과 같이 구성됩니다.

목적지	Target
10.0.0.0/16	로컬
172.31.0.0/16	pcx-1a2b1a2b

VPC B의 라우팅 테이블은 다음과 같이 구성됩니다.

목적지	Target
172.31.0.0/16	로컬
10.0.0.0/16	pcx-1a2b1a2b

VPC와 인스턴스가 IPv6 통신을 할 수 있는 경우, VPC 피어링 연결은 VPC의 인스턴스 간 IPv6 통신도 지원할 수 있습니다. 자세한 내용은 [VPC 및 서브넷 \(p. 62\)](#) 섹션을 참조하십시오. VPC 간 IPv6 트래픽을 라우팅

할 수 있게 하려면, VPC 피어링 연결을 가리키는 라우팅 테이블에 대한 경로를 추가하여 피어 VPC의 IPv6 CIDR 블록 전부 또는 일부에 액세스해야 합니다.

예를 들어 VPC가 위와 같이 동일한 VPC 피어링 연결(`pcx-1a2b1a2b`)을 사용하여 다음과 같은 정보를 갖고 있다고 가정합니다.

- VPC A: IPv6 CIDR 블록은 `2001:db8:1234:1a00::/56`입니다.
- VPC B: IPv6 CIDR 블록은 `2001:db8:5678:2b00::/56`입니다.

VPC 피어링 연결을 통해 IPv6 통신을 할 수 있도록 하려면 VPC A에 대한 라우팅 테이블에 다음 경로를 추가합니다.

목적지	Target
10.0.0.0/16	로컬
172.31.0.0/16	pcx-1a2b1a2b
2001:db8:5678:2b00::/56	pcx-1a2b1a2b

다음 경로를 VPC B에 대한 라우팅 테이블에 추가합니다.

목적지	Target
172.31.0.0/16	로컬
10.0.0.0/16	pcx-1a2b1a2b
2001:db8:1234:1a00::/56	pcx-1a2b1a2b

VPC 피어링 연결에 대한 자세한 내용은 [Amazon VPC Peering Guide](#)를 참조하십시오.

## 에 대한 라우팅 테이블

는 EC2-Classical 인스턴스를 VPC에 연결함으로써 프라이빗 IPv4 주소를 사용하여 EC2-Classical 인스턴스와 VPC의 인스턴스 간의 통신을 허용하는 기능입니다. 에 대한 자세한 내용은 (p. 211) 단원을 참조하십시오.

VPC에서 를 활성화하면 모든 VPC 라우팅 테이블에 목적지가 `10.0.0.0/8`, 대상이 `local`인 경로가 추가됩니다. 따라서 VPC의 인스턴스와 VPC에 링크된 인스턴스 간에 통신이 가능합니다. 가능 VPC에 다른 라우팅 테이블을 추가하면 VPC는 목적지가 `10.0.0.0/8`, 대상이 `local`인 경로를 자동으로 수신합니다. VPC에서 를 비활성화하면 이 라우팅이 모든 VPC의 라우팅 테이블에서 자동으로 삭제됩니다.

VPC의 라우팅 테이블 중 아무 테이블이나 `10.0.0.0/8` CIDR 이내의 주소 범위에 대한 기존 경로가 있는 경우에는 에 대해 VPC를 사용할 수 없습니다. 여기에는 `10.0.0.0/16` 및 `10.1.0.0/16` IP 주소 범위의 VPC에 대한 로컬 경로가 포함되지 않습니다.

VPC에서 를 이미 활성화한 경우 `10.0.0.0/8` IP 주소 범위에 대해 더 구체적인 경로를 라우팅 테이블에 추가할 수 없습니다.

VPC의 인스턴스와 피어 VPC에 연결된 EC2-Classical 인스턴스 간에 통신할 수 있도록 VPC 피어링 연결을 수정하면, 목적지가 `10.0.0.0/8`이고 대상이 `local`인 라우팅 테이블에 고정 경로가 자동으로 추가됩니다. VPC에 연결된 로컬 EC2-Classical 인스턴스와 피어 VPC의 인스턴스 간에 통신할 수 있도록 VPC 피어링 연결을 수정하는 경우, 목적지가 피어 VPC CIDR 블록이고 대상이 VPC 피어링 연결인 기본 라우팅 테이블에 경로를 수동으로 추가해야 합니다. EC2-Classical 인스턴스는 피어 VPC로 라우팅하는 데 기본 라우팅 테이블을 사용합니다. 자세한 내용은 Amazon VPC Peering Guide의 [ClassicLink로 구성](#)을 참조하십시오.

## VPC 종단점을 위한 라우팅 테이블

VPC 엔드포인트를 사용하면 VPC와 다른 AWS 서비스 사이에 프라이빗 연결을 생성할 수 있습니다. 엔드포인트를 만들 경우 VPC에서 엔드포인트가 사용하는 라우팅 테이블을 지정합니다. 경로는 각각의 라우팅 테이블에 자동으로 추가되며 이때 서비스의 접두사 목록 ID(`p1-xxxxxxxx`)를 지정하는 목적지 및 엔드포인트 ID(`vpce-xxxxxxxx`)를 포함한 대상도 함께 추가됩니다. 엔드포인트 경로를 명시적으로 삭제하거나 수정할 수는 없지만, 엔드포인트에서 사용되는 라우팅 테이블을 변경할 수는 있습니다.

엔드포인트에 대한 라우팅과 AWS 서비스로의 경로에 대한 의미를 자세히 알아보려면 [엔드포인트의 라우팅 \(p. 201\)](#) 단원을 참조하십시오.

## 외부 전용 인터넷 게이트웨이에 대한 라우팅 테이블

VPC에 외부 전용 인터넷 게이트웨이를 생성하여 프라이빗 서브넷의 인스턴스가 인터넷에 대한 아웃바운드 통신을 시작하도록 하되 인터넷에서 해당 인스턴스와의 연결을 시작하지 못하도록 할 수 있습니다. 외부 전용 인터넷 게이트웨이는 IPv6 트래픽에만 사용됩니다. 외부 전용 인터넷 게이트웨이에 대한 라우팅을 구성하려면 IPv6 인터넷 트래픽(`::/0`)을 외부 전용 인터넷 게이트웨이로 라우팅하는 프라이빗 서브넷에 대한 경로를 추가해야 합니다. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

## 라우팅 테이블 작업

이 섹션에서는 라우팅 테이블로 작업하는 방법을 설명합니다.

### Note

콘솔에서 마법사를 사용하여 게이트웨이가 있는 VPC를 만들 때, 마법사가 해당 게이트웨이를 사용하도록 라우팅 테이블을 자동으로 업데이트합니다. 명령줄 도구 또는 API를 사용하여 VPC를 설정할 경우에는 라우팅 테이블을 스스로 업데이트해야 합니다.

### 항목

- 서브넷이 연결되어 있는 라우팅 테이블 확인 (p. 160)
- 테이블과 명시적으로 연결되어 있는 서브넷 확인 (p. 161)
- 사용자 지정 라우팅 테이블 생성 (p. 161)
- 라우팅 테이블에 경로 추가 및 라우팅 테이블에서 경로 제거 (p. 161)
- 경로 전파 활성화 및 비활성화 (p. 162)
- 서브넷을 라우팅 테이블과 연결 (p. 162)
- 서브넷의 라우팅 테이블 변경 (p. 162)
- 라우팅 테이블에서 서브넷 연결 끊기 (p. 162)
- 기본 라우팅 테이블 바꾸기 (p. 163)
- 라우팅 테이블 삭제 (p. 163)

## 서브넷이 연결되어 있는 라우팅 테이블 확인

Amazon VPC 콘솔에서 서브넷의 세부 정보를 살펴보면 서브넷이 연결되어 있는 라우팅 테이블을 확인할 수 있습니다.

서브넷이 연결되어 있는 라우팅 테이블을 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets을 선택합니다.
3. 서브넷 세부 정보는 [Summary] 탭에 표시됩니다. 라우팅 테이블 ID와 경로를 보려면 [Route Table] 탭을 선택합니다. 기본 라우팅 테이블인 경우, 콘솔은 연결이 암시적인지, 명시적인지 표시하지 않습니다. 기본 라우팅 테이블에 대한 연결이 명시적인지 확인하려면 [테이블과 명시적으로 연결되어 있는 서브넷 확인 \(p. 161\)](#) 단원을 참조하십시오.

## 테이블과 명시적으로 연결되어 있는 서브넷 확인

어떤 서브넷이 몇 개나 라우팅 테이블과 명시적으로 연결되어 있는지 확인할 수 있습니다.

기본 라우팅 테이블에는 명시적 및 암시적 연결이 있을 수 있습니다. 사용자 지정 라우팅 테이블에는 명시적 연결만 있습니다.

어떤 라우팅 테이블과도 명시적으로 연결되지 않은 서브넷은 기본 라우팅 테이블과 암시적으로 연결됩니다. 서브넷을 기본 라우팅 테이블과 명시적으로 연결할 수 있습니다(그렇게 할 수 있는 이유를 설명하는 예를 보려면 [기본 라우팅 테이블 바꾸기 \(p. 163\)](#) 참조).

어떤 서브넷이 명시적으로 연결되어 있는지 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. [Explicitly Associated With] 열을 확인하여 명시적으로 연결되어 있는 서브넷 수를 결정합니다.
4. 필요한 라우팅 테이블을 선택합니다.
5. 세부 정보 창에서 [Subnet Associations] 탭을 선택합니다. 테이블과 명시적으로 연결되어 있는 서브넷이 탭에 나열됩니다. 어떤 라우팅 테이블과도 연결되지 않아 기본 라우팅 테이블과 암시적으로 연결되어 있는 서브넷도 나열됩니다.

## 사용자 지정 라우팅 테이블 생성

Amazon VPC 콘솔을 사용하여 VPC에 대한 사용자 지정 라우팅 테이블을 만들 수 있습니다.

사용자 지정 라우팅 테이블을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. [Create Route Table]을 선택합니다.
4. [Create Route Table] 대화 상자에서 [Name tag]에 라우팅 테이블 이름을 선택적으로 지정할 수 있습니다. 이름을 지정하면 Name 키와 사용자가 지정하는 값을 가진 태그가 생성됩니다. [VPC]에 대해 VPC를 선택한 후 [Yes, Create]를 선택합니다.

## 라우팅 테이블에 경로 추가 및 라우팅 테이블에서 경로 제거

라우팅 테이블에서 경로를 추가, 삭제 및 수정할 수 있습니다. 사용자가 추가한 경로만 수정할 수 있습니다.

라우팅 테이블에 경로를 추가하거나 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Routes] 탭에서 [Edit]를 선택합니다.
4. 기존 경로를 수정하려면 [Destination]에 대해 목적지 CIDR 블록 또는 단일 IP 주소를 바꾼 다음, [Target]에 대해 대상을 선택합니다. [Add another route], [Save]를 선택합니다.

라우팅 테이블에서 경로를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Routes] 탭에서 [Edit]를 선택한 다음, 삭제할 경로에 대해 [Remove]를 선택합니다.
4. 완료되면 [Save]를 선택합니다.



## 경로 전파 활성화 및 비활성화

경로 전파를 통해 가상 프라이빗 게이트웨이가 라우팅 테이블로 경로를 자동으로 전파할 수 있으므로, 라우팅 테이블에 VPN 경로를 수동으로 입력할 필요가 없습니다. 경로 전파를 활성화하거나 비활성화할 수 있습니다.

VPN 라우팅 옵션에 대한 자세한 내용은 [VPN 라우팅 옵션 \(p. 215\)](#) 단원을 참조하십시오.

경로 전파를 활성화하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Route Propagation] 탭에서 [Edit]를 선택합니다.
4. 가상 프라이빗 게이트웨이 옆에 있는 [Propagate] 확인란을 선택한 후 [Save]를 선택합니다.

경로 전파를 비활성화하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Route Propagation] 탭에서 [Edit]를 선택합니다.
4. [Propagate] 확인란의 선택을 취소하고 [Save]를 선택합니다.

## 서브넷을 라우팅 테이블과 연결

특정 서브넷에 라우팅 테이블의 경로를 적용하려면 라우팅 테이블을 서브넷과 연결해야 합니다. 한 라우팅 테이블을 여러 서브넷과 연결할 수 있지만, 한 서브넷은 한 번에 한 라우팅 테이블과 연결할 수 있을 뿐입니다. 테이블과 명시적으로 연결되지 않은 서브넷은 기본적으로 기본 라우팅 테이블과 암시적으로 연결됩니다.

서브넷에 라우팅 테이블을 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Subnet Associations] 탭에서 [Edit]를 선택합니다.
4. 라우팅 테이블과 연결할 서브넷에 대한 [Associate] 확인란을 선택한 후 [Save]를 선택합니다.

## 서브넷의 라우팅 테이블 변경

서브넷이 연결되는 라우팅 테이블을 변경할 수 있습니다.

서브넷의 라우팅 테이블 연결을 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Subnets]를 선택한 후 서브넷을 선택합니다.
3. [Route Table] 탭에서 [Edit]를 선택합니다.
4. [Change to] 목록에서 서브넷과 연결할 새 라우팅 테이블을 선택한 후 [Save]를 선택합니다.

## 라우팅 테이블에서 서브넷 연결 끊기

서브넷과 라우팅 테이블의 연결을 해제할 수 있습니다. 서브넷에 다른 라우팅 테이블을 명시적으로 연결하지 않는 한 서브넷에는 기본 라우팅 테이블이 암시적으로 연결됩니다.



라우팅 테이블에서 서브넷의 연결을 끊으려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Subnet Associations] 탭에서 [Edit]를 선택합니다.
4. 서브넷에 대한 [Associate] 확인란의 선택을 취소한 후 [Save]를 선택합니다.

## 기본 라우팅 테이블 바꾸기

VPC에서 어떤 라우팅 테이블이 기본 라우팅 테이블인지를 변경할 수 있습니다.

기본 라우팅 테이블을 바꾸려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. 새 기본 라우팅 테이블로 지정할 라우팅 테이블을 선택한 다음, [Set as Main Table]을 선택합니다.
4. 확인 대화 상자가 나타나면 [Yes, Set]를 선택합니다.

다음 절차에서는 서브넷과 기본 라우팅 테이블 간의 명시적 연결을 제거하는 방법을 설명합니다. 결과적으로, 서브넷과 기본 라우팅 테이블 사이에 암시적 연결이 설정됩니다. 이 프로세스는 임의의 라우팅 테이블에서 서브넷의 연결을 끊는 프로세스와 같습니다.

기본 라우팅 테이블과의 명시적 연결을 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 후 라우팅 테이블을 선택합니다.
3. [Subnet Associations] 탭에서 [Edit]를 선택합니다.
4. 서브넷에 대한 [Associate] 확인란의 선택을 취소한 후 [Save]를 선택합니다.

## 라우팅 테이블 삭제

라우팅 테이블과 연결된 서브넷이 없는 경우에만 라우팅 테이블을 삭제할 수 있습니다. 기본 라우팅 테이블은 삭제할 수 없습니다.

라우팅 테이블을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. 라우팅 테이블을 선택한 후 Delete Route Table을 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

## API 및 명령 개요

이 페이지에서 설명한 작업은 명령줄이나 API를 사용하여 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#)를 참조하십시오.

사용자 지정 라우팅 테이블 생성

- [create-route-table](#)(AWS CLI)
- [New-EC2RouteTable](#)(Windows PowerShell용 AWS 도구)

#### 라우팅 테이블에 경로 추가

- [create-route](#)(AWS CLI)
- [New-EC2Route](#)(Windows PowerShell용 AWS 도구)

#### 서브넷을 라우팅 테이블과 연결

- [associate-route-table](#)(AWS CLI)
- [Register-EC2RouteTable](#)(Windows PowerShell용 AWS 도구)

#### 하나 이상의 라우팅 테이블 설명

- [describe-route-tables](#)(AWS CLI)
- [Get-EC2RouteTable](#)(Windows PowerShell용 AWS 도구)

#### 라우팅 테이블에서 경로 삭제

- [delete-route](#)(AWS CLI)
- [Remove-EC2Route](#)(Windows PowerShell용 AWS 도구)

#### 라우팅 테이블에 있는 기존 경로 바꾸기

- [replace-route](#)(AWS CLI)
- [Set-EC2Route](#)(Windows PowerShell용 AWS 도구)

#### 라우팅 테이블에서 서브넷의 연결 끊기

- [disassociate-route-table](#)(AWS CLI)
- [Unregister-EC2RouteTable](#)(Windows PowerShell용 AWS 도구)

#### 서브넷과 연결된 라우팅 테이블 변경

- [replace-route-table-association](#)(AWS CLI)
- [Set-EC2RouteTableAssociation](#)(Windows PowerShell용 AWS 도구)

#### VPN 연결과 관련된 고정 경로 생성

- [create-vpn-connection-route](#)(AWS CLI)
- [New-EC2VpnConnectionRoute](#)(Windows PowerShell용 AWS 도구)

#### VPN 연결과 관련된 고정 경로 삭제

- [delete-vpn-connection-route](#)(AWS CLI)
- [Remove-EC2VpnConnectionRoute](#)(Windows PowerShell용 AWS 도구)

#### 가상 프라이빗 게이트웨이(VGW)를 사용하여 VPC의 라우팅 테이블로 경로 전파

- [enable-vgw-route-propagation](#)(AWS CLI)
- [Enable-EC2VgwRoutePropagation](#)(Windows PowerShell용 AWS 도구)

VGW가 VPC의 라우팅 테이블로 경로를 전파하지 못하도록 비활성화

- [disable-vgw-route-propagation](#)(AWS CLI)
- [Disable-EC2VgwRoutePropagation](#)(Windows PowerShell용 AWS 도구)

라우팅 테이블 삭제

- [delete-route-table](#)(AWS CLI)
- [Remove-EC2RouteTable](#)(Windows PowerShell용 AWS 도구)

## 인터넷 게이트웨이

인터넷 게이트웨이는 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로, VPC의 인스턴스와 인터넷 간에 통신할 수 있게 해줍니다. 따라서 네트워크 트래픽에 가용성 위험이나 대역폭 제약 조건이 발생하지 않습니다.

인터넷 게이트웨이에는 인터넷 라우팅 가능 트래픽에 대한 VPC 라우팅 테이블에 대상을 제공하고, 퍼블릭 IPv4 주소가 배정된 인스턴스에 대해 NAT(네트워크 주소 변환)을 수행하는 두 가지 목적이 있습니다.

인터넷 게이트웨이는 IPv4 및 IPv6 트래픽을 지원합니다.

## 인터넷 액세스 활성화

VPC 서브넷의 인스턴스에 대한 인터넷 액세스를 활성화하려면 다음을 수행해야 합니다.

- VPC에 인터넷 게이트웨이를 연결합니다.
- 서브넷의 라우팅 테이블이 인터넷 게이트웨이를 가리키는지 확인합니다.
- 서브넷의 인스턴스에 전역적으로 고유한 IP 주소(퍼블릭 IPv4 주소, 탄력적 IP 주소 또는 IPv6 주소)가 있는지 확인합니다.
- 네트워크 액세스 제어 및 보안 그룹 규칙에서 적절한 트래픽이 인스턴스로, 그리고 인스턴스에서 흐르도록 허용되는지 확인합니다.

인터넷 게이트웨이를 사용하려면 서브넷의 라우팅 테이블에는 인터넷에 바인딩된 트래픽을 인터넷 게이트웨이로 라우팅하는 경로가 있어야 합니다. 라우팅 테이블에 명시적으로 알려져 있지 않은 모든 대상에 대한 경로의 범위를 지정하거나(IPv4의 경우 `0.0.0.0/0`, IPv6의 경우 `::/0`) 더 좁은 IP 주소 범위로 경로의 범위를 지정할 수 있습니다(예: AWS 외부에 있는 회사 퍼블릭 엔드포인트의 퍼블릭 IPv4 주소 또는 VPC 외부에 있는 다른 Amazon EC2 인스턴스의 탄력적 IP 주소). 서브넷이 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블과 연결되는 경우, 이를 퍼블릭 서브넷이라고 합니다.

IPv4를 위해 인터넷으로 통신할 수 있도록 하려면, 인스턴스의 프라이빗 IPv4 주소와 연결된 퍼블릭 IPv4 주소 또는 탄력적 IP 주소가 인스턴스에 있어야 합니다. 사용자의 인스턴스는 VPC 및 서브넷 내부에서 정의된 프라이빗(내부) IP 주소 공간만 인식합니다. 인터넷 게이트웨이는 사용자의 인스턴스를 대신하여 논리적으로 일대일 NAT를 제공하므로, 트래픽이 VPC 서브넷을 떠나 인터넷으로 이동할 때 수신 주소 필드는 프라이빗 IP 주소가 아니라, 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소로 설정됩니다. 반대로, 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 목적지로 하는 트래픽에는 트래픽이 VPC로 전달되기 전에 인스턴스의 프라이빗 IPv4 주소로 변환되는 대상 주소가 있습니다.

IPv6를 위해 인터넷을 통한 통신을 가능케 하려면, VPC 및 서브넷에 연결된 IPv6 CIDR 블록이 있어야 하고, 서브넷의 범위에 속한 IPv6 주소가 인스턴스에 할당되어야 합니다. IPv6 주소는 전역적으로 고유하므로 퍼블릭으로 기본 설정되어 있습니다.

다음 다이어그램에서 VPC의 서브넷 1은 인터넷 바인딩된 모든 IPv4 트래픽이 인터넷 게이트웨이를 가리키도록 하는 사용자 지정 라우팅 테이블에 연결되어 있습니다. 해당 인스턴스에는 탄력적 IP 주소가 있어 인터넷과의 통신이 가능합니다.

기본 VPC와 기본이 아닌 VPC에 대한 인터넷 액세스

다음 표는 VPC가 IPv4 또는 IPv6를 통해 인터넷 액세스에 필요한 구성 요소와 함께 자동으로 제공으로 제공되는지 여부를 개괄적으로 제시합니다.

	기본 VPC	기본이 아닌 VPC
인터넷 게이트웨이	예	VPC 마법사의 첫 번째 또는 두 번째 옵션을 사용하여 VPC를 생성한 경우. 그렇지 않은 경우, 인터넷 게이트웨이를 수동으로 생성하여 연결해야 합니다.
IPv4 트래픽을 위한 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블(0.0.0.0/0)	예	VPC 마법사의 첫 번째 또는 두 번째 옵션을 사용하여 VPC를 생성한 경우. 그렇지 않은 경우, 라우팅 테이블을 수동으로 생성하여 경로를 추가해야 합니다.
IPv6 트래픽을 위한 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블(::/0)	아니요	VPC 마법사의 첫 번째 또는 두 번째 옵션을 사용하여 VPC를 생성한 경우, 그리고 IPv6 CIDR 블록을 VPC와 연결하기 위해 옵션을 지정한 경우. 그렇지 않은 경우, 라우팅 테이블을 수동으로 생성하여 경로를 추가해야 합니다.
서브넷에서 시작된 인스턴스에 자동 할당된 퍼블릭 IPv4 주소	예(기본 서브넷)	아니요(기본이 아닌 서브넷)
서브넷에서 시작된 인스턴스에 자동 할당된 IPv6 주소	아니요(기본 서브넷)	아니요(기본이 아닌 서브넷)

기본 VPC에 대한 자세한 내용은 [기본 VPC 및 기본 서브넷 \(p. 72\)](#) 단원을 참조하십시오. VPC 마법사를 사용하여 인터넷 게이트웨이로 VPC를 만드는 방법에 대한 자세한 정보는 [시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC \(p. 17\)](#) 또는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\) \(p. 23\)](#) 단원을 참조하십시오.

VPC에서 IP 주소를 지정하고 인스턴스에 퍼블릭 IPv4 또는 IPv6 주소를 할당하는 방식을 제어하는 방법에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

VPC에 새 서브넷을 추가할 때 해당 서브넷의 라우팅 및 보안 설정을 구성해야 합니다.

## 인터넷 게이트웨이로 VPC 생성

다음 섹션에서는 인터넷 액세스를 지원하기 위해 퍼블릭 서브넷을 수동으로 만드는 방법을 설명합니다.

항목

- [서브넷 만들기 \(p. 167\)](#)
- [인터넷 게이트웨이 연결 \(p. 167\)](#)
- [사용자 지정 라우팅 테이블 생성 \(p. 167\)](#)
- [보안 그룹 규칙 업데이트 \(p. 168\)](#)
- [탄력적 IP 주소 추가 \(p. 168\)](#)
- [VPC에서 인터넷 게이트웨이 분리 \(p. 169\)](#)
- [인터넷 게이트웨이 삭제 \(p. 169\)](#)
- [API 및 명령 개요 \(p. 169\)](#)

## 서브넷 만들기

VPC에 서브넷을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Subnets]를 선택한 다음 [Create Subnet]을 선택합니다.
3. Create Subnet 대화 상자에서 해당 VPC 및 가용 영역을 선택하고, 서브넷에 IPv4 CIDR 블록을 지정합니다.
4. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Specify a custom IPv6 CIDR을 선택합니다.
5. [Yes, Create]를 선택합니다.

서브넷에 대한 자세한 정보는 [VPC 및 서브넷 \(p. 62\)](#) 단원을 참조하십시오.

## 인터넷 게이트웨이 연결

인터넷 게이트웨이를 생성하여 VPC에 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Internet Gateways]를 선택한 후 [Create Internet Gateway]를 선택합니다.
3. [Create Internet Gateway] 대화 상자에서 선택적으로 인터넷 게이트웨이의 이름을 지정할 수 있으며, 그런 다음 [Yes, Create]를 선택합니다.
4. 방금 생성한 인터넷 게이트웨이를 선택한 후 [Attach to VPC]를 선택합니다.
5. [Attach to VPC] 대화 상자의 목록에서 VPC를 선택한 후 [Yes, Attach]를 선택합니다.

## 사용자 지정 라우팅 테이블 생성

서브넷을 생성하면 생성된 서브넷이 VPC의 기본 라우팅 테이블과 자동으로 연결됩니다. 기본적으로, 기본 라우팅 테이블에는 인터넷 게이트웨이에 대한 경로가 포함되지 않습니다. 다음 절차에서는 VPC 외부 위치를 목적지로 하는 트래픽을 인터넷 게이트웨이로 보내는 경로를 포함한 사용자 지정 라우팅 테이블을 생성한 다음, 이를 서브넷과 연결합니다.

사용자 지정 라우팅 테이블을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택한 다음 [Create Route Table]을 선택합니다.
3. [Create Route Table] 대화 상자에서 선택적으로 라우팅 테이블의 이름을 지정한 다음, [Yes, Create]를 선택합니다.
4. 방금 생성한 사용자 지정 라우팅 테이블을 선택합니다. 세부 정보 창에는 경로, 연결 및 경로 전파 작업을 위한 탭이 표시됩니다.
5. Routes 탭에서 Edit, Add another route를 선택한 후 필요한 경우 경로를 추가합니다. 완료되면 [Save]를 선택합니다.
  - IPv4 트래픽에 대해 Destination 상자에서 0.0.0.0/0을 지정하고, Target 목록에서 인터넷 게이트웨이 ID를 선택합니다.
  - IPv6 트래픽에 대해 Destination 상자에서 ::/0을 지정하고, Target 목록에서 인터넷 게이트웨이 ID를 선택합니다.
6. [Subnet Associations] 탭에서 [Edit]를 선택하고, 서브넷에 대해 [Associate] 확인란을 선택한 다음, [Save]를 선택합니다.

라우팅 테이블에 대한 자세한 내용은 [라우팅 테이블 \(p. 154\)](#) 단원을 참조하십시오.

## 보안 그룹 규칙 업데이트

VPC는 기본 보안 그룹과 함께 제공됩니다. VPC로 시작하는 각 인스턴스는 기본 보안 그룹과 자동으로 연결됩니다. 기본 보안 그룹에 대한 기본 설정에서는 인터넷으로부터의 inbound 트래픽이 허용되지 않고 인터넷으로 향하는 모든 아웃바운드 트래픽은 허용됩니다. 따라서 인스턴스가 인터넷과 통신할 수 있도록 하려면 퍼블릭 인스턴스가 인터넷에 액세스할 수 있도록 허용하는 새 보안 그룹을 만듭니다.

새 보안 그룹을 생성하여 인스턴스와 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택한 다음, [Create Security Group]을 선택합니다.
3. [Create Security Group] 대화 상자에서 보안 그룹의 이름과 설명을 지정합니다. [VPC] 목록에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.
4. 보안 그룹을 선택합니다. 세부 정보 창에는 inbound 규칙 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
5. Inbound Rules 탭에서 [Edit]를 선택합니다. [Add Rule]을 선택하고 필수 정보를 빠짐없이 제공합니다. 예를 들어 Type 목록에서 HTTP 또는 HTTPS를 선택하고, Source를 IPv4 트래픽의 경우 0.0.0.0/0으로, IPv6 트래픽의 경우 ::/0으로 입력합니다. 완료되면 [Save]를 선택합니다.
6. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
7. 탐색 창에서 [Instances]를 선택합니다.
8. 인스턴스를 선택하고 [Actions]를 선택한 다음, [Networking]과 [Change Security Groups]를 차례로 선택합니다.
9. [Change Security Groups] 대화 상자에서 현재 선택되어 있는 보안 그룹에 대한 확인란을 지우고 새 보안 그룹을 선택합니다. [Assign Security Groups]를 선택합니다.

보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참조하십시오.

## 탄력적 IP 주소 추가

인스턴스를 서브넷으로 시작한 후, IPv4를 통해 인터넷에서 접속할 수 있도록 하려면 인스턴스에 탄력적 IP 주소를 지정해야 합니다.

### Note

시작 중에 퍼블릭 IPv4 주소를 인스턴스에 배정한 경우에는 인터넷에서 인스턴스에 연결할 수 있으므로, 인스턴스에 탄력적 IP 주소를 지정할 필요가 없습니다. 인스턴스에 대한 IP 주소 지정에 관한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 단원을 참조하십시오.

콘솔을 사용하여 인스턴스에 탄력적 IP 주소를 할당하고 지정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. [Allocate]를 선택합니다.

### Note

계정이 EC2-Classic을 지원할 경우 우선 [VPC]를 선택합니다.

5. 목록에서 탄력적 IP 주소와 [Actions], [Associate Address]를 차례로 선택합니다.
6. [Instance] 또는 [Network interface]를 선택한 다음 인스턴스 또는 네트워크 인터페이스 ID를 선택합니다. 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 선택한 다음 [Associate]를 선택합니다.

탄력적 IP 주소에 대한 자세한 내용은 [탄력적 IP 주소 \(p. 197\)](#) 단원을 참조하십시오.

## VPC에서 인터넷 게이트웨이 분리

기본 VPC가 아닌 VPC로 시작하는 인스턴스에 대해 더 이상 인터넷 액세스가 필요하지 않으면, VPC에서 인터넷 게이트웨이를 분리할 수 있습니다. VPC에 탄력적 IP 주소가 연결된 인스턴스가 있는 경우에는 인터넷 게이트웨이를 분리할 수 없습니다.

인터넷 게이트웨이를 분리하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택한 후 탄력적 IP 주소를 선택합니다.
3. [Actions], [Disassociate address]를 선택합니다. Disassociate address를 선택합니다.
4. 탐색 창에서 [Internet Gateways]를 선택합니다.
5. 인터넷 게이트웨이를 선택하고 [Detach from VPC]를 선택합니다.
6. [Detach from VPC] 대화 상자에서 [Yes, Detach]를 선택합니다.

## 인터넷 게이트웨이 삭제

인터넷 게이트웨이가 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 아직 VPC에 연결되어 있는 인터넷 게이트웨이는 삭제할 수 없습니다.

인터넷 게이트웨이를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Internet Gateways]를 선택합니다.
3. 인터넷 게이트웨이를 선택하고 [Delete]를 선택합니다.
4. [Delete Internet Gateway] 대화 상자에서 [Yes, Delete]를 선택합니다.

## API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

인터넷 게이트웨이 만들기

- [create-internet-gateway](#)(AWS CLI)
- [New-EC2InternetGateway](#)(Windows PowerShell용 AWS 도구)

VPC에 인터넷 게이트웨이 연결

- [attach-internet-gateway](#)(AWS CLI)
- [Add-EC2InternetGateway](#)(Windows PowerShell용 AWS 도구)

인터넷 게이트웨이 설명

- [describe-internet-gateways](#)(AWS CLI)
- [Get-EC2InternetGateway](#)(Windows PowerShell용 AWS 도구)

VPC에서 인터넷 게이트웨이 분리

- [detach-internet-gateway](#)(AWS CLI)
- [Dismount-EC2InternetGateway](#)(Windows PowerShell용 AWS 도구)

인터넷 게이트웨이 삭제

- `delete-internet-gateway`(AWS CLI)
- `Remove-EC2InternetGateway`(Windows PowerShell용 AWS 도구)

## 외부 전용 인터넷 게이트웨이

외부 전용 인터넷 게이트웨이는 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로서, VPC의 인스턴스에서 인터넷으로 IPv6를 통한 아웃바운드 통신을 가능케 하되 인터넷에서 해당 인스턴스와의 IPv6 연결을 시작하지 못하게 할 수 있습니다.

### Note

외부 전용 인터넷 게이트웨이는 IPv6 트래픽에만 사용됩니다. IPv4를 통한 아웃바운드 전용 인터넷 통신을 사용하려면 NAT 게이트웨이를 사용하십시오. 자세한 내용은 [NAT 게이트웨이 \(p. 173\)](#) 단원을 참조하십시오.

항목

- [외부 전용 인터넷 게이트웨이 기본 사항 \(p. 170\)](#)
- [외부 전용 인터넷 게이트웨이 작업 \(p. 170\)](#)
- [API 및 CLI 개요 \(p. 172\)](#)

## 외부 전용 인터넷 게이트웨이 기본 사항

퍼블릭 서브넷의 인스턴스에 퍼블릭 IPv4 주소나 IPv6 주소가 있는 경우에는 인터넷 게이트웨이를 통해 인터넷에 접속할 수 있습니다. 이와 마찬가지로 인터넷 상의 리소스는 자체 퍼블릭 IPv4 주소 또는 IPv6 주소를 사용하여 인스턴스에 대한 연결을 시작할 수 있습니다. 해당되는 예로 로컬 컴퓨터를 사용하여 인스턴스에 접속할 때를 들 수 있습니다.

IPv6 주소는 전역적으로 고유하므로 퍼블릭으로 기본 설정되어 있습니다. 인스턴스가 인터넷에 액세스할 수 있게 하되 인터넷 상의 리소스가 해당 인스턴스와의 통신을 시작하지 못하게 하려면 외부 전용 인터넷 게이트웨이를 사용하여 그렇게 할 수 있습니다. 이렇게 하려면 VPC에 외부 전용 인터넷 게이트웨이를 만들어 라우팅 테이블에 모든 IPv6 트래픽(`::/0`)을 가리키는 경로를 추가하거나 IPv6 주소의 특정 범위를 외부 전용 인터넷 게이트웨이에 추가합니다. 라우팅 테이블에 연결된 서브넷의 IPv6 트래픽은 외부 전용 인터넷 게이트웨이로 라우팅됩니다.

외부 전용 인터넷 게이트웨이는 상태 저장 방식으로서, 서브넷의 인스턴스에서 인터넷 또는 기타 AWS 서비스로 트래픽을 전달한 다음, 다시 인스턴스로 응답을 보냅니다.

외부 전용 인터넷 게이트웨이에는 다음과 같은 특성이 있습니다.

- 보안 그룹은 외부 전용 인터넷 게이트웨이와 연결할 수 없습니다. 프라이빗 서브넷의 인스턴스에 대한 보안 그룹을 사용하여 해당 인스턴스에서 주고받는 트래픽을 제어할 수 있습니다.
- 네트워크 ACL을 사용하여 외부 전용 인터넷 게이트웨이가 트래픽을 라우팅하는 서브넷에서 주고받는 트래픽을 제어할 수 있습니다.

다음 다이어그램에서 VPC에는 IPv6 CIDR 블록이, VPC의 서브넷에는 IPv6 CIDR 블록이 있습니다. 사용자 지정 라우팅 테이블은 서브넷 1에 연결되어 있고 VPC의 외부 전용 인터넷 게이트웨이로 가는 모든 인터넷 바인딩된 IPv6 트래픽(`::/0`)을 가리킵니다.

## 외부 전용 인터넷 게이트웨이 작업

다음 단원에서는 프라이빗 서브넷에 대해 외부 전용 인터넷 게이트웨이를 생성하고 서브넷에 대한 라우팅을 구성하는 방법에 대해 설명합니다.



## 외부 전용 인터넷 게이트웨이 생성

Amazon VPC 콘솔을 사용하여 VPC에 대한 외부 전용 인터넷 게이트웨이를 만들 수 있습니다.

외부 전용 인터넷 게이트웨이를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Egress Only Internet Gateways를 선택합니다.
3. Create Egress Only Internet Gateway를 선택합니다.
4. 외부 전용 인터넷 게이트웨이를 생성할 VPC를 선택합니다. Create를 선택합니다.

## 외부 전용 인터넷 게이트웨이 조회

Amazon VPC 콘솔에서 외부 전용 인터넷 게이트웨이에 대한 정보를 볼 수 있습니다.

외부 전용 인터넷 게이트웨이에 대한 정보를 보려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Egress Only Internet Gateways를 선택합니다.
3. 세부 정보 창에서 관련 정보를 조회할 외부 전용 인터넷 게이트웨이를 선택합니다.

## 사용자 지정 라우팅 테이블 생성

VPC 외부 위치를 목적지로 하는 트래픽을 외부 전용 인터넷 게이트웨이로 전송하려면 사용자 지정 라우팅 테이블을 생성하고 트래픽을 게이트웨이로 전송하는 경로를 추가한 다음, 이를 서브넷과 연결해야 합니다.

사용자 지정 라우팅 테이블을 만들고 외부 전용 인터넷 게이트웨이에 경로를 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route Tables를 선택한 다음, Create Route Table을 선택합니다.
3. [Create Route Table] 대화 상자에서 선택적으로 라우팅 테이블의 이름을 지정한 다음, [Yes, Create]를 선택합니다.
4. 방금 생성한 사용자 지정 라우팅 테이블을 선택합니다. 세부 정보 창에는 경로, 연결 및 경로 전파 작업을 위한 탭이 표시됩니다.
5. Routes 탭에서 Edit를 선택하고, Destination 상자에서 `::/0`을 지정하고, Target 목록에서 외부 전용 인터넷 게이트웨이 ID를 선택한 다음, Save를 선택합니다.
6. Subnet Associations 탭에서 Edit를 선택하고 서브넷에 대한 Associate 확인란을 선택합니다. [Save]를 선택합니다.

또는 서브넷과 연결된 기존 라우팅 테이블에 경로를 추가할 수도 있습니다. 기존 라우팅 테이블을 선택하고, 위의 5단계 및 6단계를 수행하여 외부 전용 인터넷 게이트웨이에 대한 경로를 추가합니다.

라우팅 테이블에 대한 자세한 내용은 [라우팅 테이블 \(p. 154\)](#) 단원을 참조하십시오.

## 외부 전용 인터넷 게이트웨이 삭제

외부 전용 인터넷 게이트웨이가 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 삭제된 외부 전용 인터넷 게이트웨이를 가리키는 라우팅 테이블의 모든 경로는 그 경로를 수동으로 삭제하거나 업데이트할 때까지 `blackhole` 상태로 남아 있습니다.

외부 전용 인터넷 게이트웨이를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 Egress Only Internet Gateways를 선택하고 해당되는 외부 전용 인터넷 게이트웨이를 선택합니다.
3. [Delete]를 선택합니다.
4. 확인 대화 상자에서 Delete Egress Only Internet Gateway를 선택합니다.

## API 및 CLI 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

### 외부 전용 인터넷 게이트웨이 생성

- [create-egress-only-internet-gateway](#)(AWS CLI)
- [New-EC2EgressOnlyInternetGateway](#)(Windows PowerShell용 AWS 도구)

### 외부 전용 인터넷 게이트웨이 설명

- [describe-egress-only-internet-gateways](#)(AWS CLI)
- [Get-EC2EgressOnlyInternetGatewayList](#)(Windows PowerShell용 AWS 도구)

### 외부 전용 인터넷 게이트웨이 삭제

- [delete-egress-only-internet-gateway](#)(AWS CLI)
- [Remove-EC2EgressOnlyInternetGateway](#)(Windows PowerShell용 AWS 도구)

## NAT

NAT 디바이스를 사용하여 프라이빗 서브넷의 인스턴스를 인터넷(예: 소프트웨어 업데이트용) 또는 기타 AWS 서비스에 연결하는 한편, 인터넷에서 해당 인스턴스와의 연결을 시작하지 못하도록 할 수 있습니다. NAT 디바이스는 프라이빗 서브넷의 인스턴스에서 인터넷 또는 기타 AWS 서비스로 트래픽을 전달한 다음 인스턴스에 응답을 다시 보냅니다. 트래픽이 인터넷으로 이동하면 소스 IPv4 주소가 NAT 디바이스의 주소로 대체되고, 이와 마찬가지로 응답 트래픽이 해당 인스턴스로 이동하면 NAT 디바이스에서 주소를 해당 인스턴스의 프라이빗 IPv4 주소로 다시 변환합니다.

NAT 디바이스는 IPv6 트래픽을 지원하지 않으므로, 그 대신에 외부 전용 인터넷 게이트웨이를 사용하십시오. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

### Note

이 문서에서는 NAT라는 용어를 일반적인 IT 용례에 따라 사용하지만, 실제로 NAT 디바이스는 주소 변환과 포트 주소 변환(PAT)을 모두 담당합니다.

AWS에서는 두 가지 종류의 NAT 디바이스(NAT 게이트웨이 또는 NAT 인스턴스)를 제공합니다. NAT 게이트웨이가 NAT 인스턴스보다 우수한 가용성 및 대역폭을 제공하므로 NAT 게이트웨이를 사용하는 것이 좋습니다. 또한 NAT 게이트웨이 서비스는 관리 작업이 필요하지 않은 관리형 서비스입니다. NAT 인스턴스는 NAT AMI에서 시작됩니다. 특별한 경우에 NAT 인스턴스를 사용하도록 선택할 수 있습니다.

- [NAT 게이트웨이 \(p. 173\)](#)
- [NAT 인스턴스 \(p. 182\)](#)
- [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 188\)](#)

## NAT 게이트웨이

NAT(네트워크 주소 변환) 게이트웨이를 사용하여 프라이빗 서브넷의 인스턴스를 인터넷 또는 기타 AWS 서비스에 연결하는 한편, 인터넷에서 해당 인스턴스와의 연결을 시작하지 못하도록 할 수 있습니다. NAT에 대한 자세한 내용은 [NAT \(p. 172\)](#)를 참조하십시오.

계정에서 NAT 게이트웨이 생성 및 사용에 대한 요금이 청구됩니다. NAT 게이트웨이 시간당 사용 요금 및 데이터 처리 요금이 적용됩니다. Amazon EC2 데이터 전송 요금도 적용됩니다. 자세한 내용은 [Amazon VPC 요금](#)을 참조하십시오.

NAT 게이트웨이는 IPv6 트래픽을 지원하지 않으므로, 그 대신에 외부 전용 인터넷 게이트웨이를 사용하십시오. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

### 항목

- [NAT 게이트웨이 기본 사항 \(p. 173\)](#)
- [NAT 게이트웨이 사용 \(p. 174\)](#)
- [NAT 게이트웨이 문제 해결 \(p. 177\)](#)
- [NAT 게이트웨이 사용 제어 \(p. 181\)](#)
- [API 및 CLI 개요 \(p. 181\)](#)

## NAT 게이트웨이 기본 사항

NAT 게이트웨이를 만들려면 NAT 게이트웨이가 속할 퍼블릭 서브넷을 지정해야 합니다. 퍼블릭 서브넷과 프라이빗 서브넷에 대한 자세한 내용은 [서브넷 라우팅 \(p. 64\)](#)을 참조하십시오. NAT 게이트웨이를 만들 때 NAT 게이트웨이와 연결할 탄력적 IP 주소도 지정해야 합니다. NAT 게이트웨이를 만든 후에는 인터넷 바운드 트래픽이 NAT 게이트웨이를 가리키도록 하나 이상의 프라이빗 서브넷과 연결된 라우팅 테이블을 업데이트해야 합니다. 그러면 프라이빗 서브넷의 인스턴스가 인터넷과 통신할 수 있습니다.

각 NAT 게이트웨이는 특정 가용 영역에 생성되고 해당 영역에서 중복성을 통해 구현됩니다. 가용 영역에서 만들 수 있는 NAT 게이트웨이 수에는 제한이 있습니다. 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

### Note

여러 가용 영역에 리소스가 있고 하나의 NAT 게이트웨이를 공유하는 경우 NAT 게이트웨이의 가용 영역이 다운되면 다른 가용 영역의 리소스도 인터넷에 액세스할 수 없습니다. 가용 영역과 독립적인 아키텍처를 만들려면 각 가용 영역에 NAT 게이트웨이를 만들고 리소스가 동일한 가용 영역의 NAT 게이트웨이를 사용하도록 라우팅을 구성합니다.

NAT 게이트웨이가 더 이상 필요하지 않으면 삭제할 수 있습니다. NAT 게이트웨이를 삭제하면 탄력적 IP 주소가 연결 해제되지만 계정에서 주소가 해제되지는 않습니다.

NAT 게이트웨이에는 다음과 같은 특성이 있습니다.

- NAT 게이트웨이는 최대 10Gbps 대역폭을 지원합니다. 10Gbps 이상이 필요한 경우 리소스를 여러 서브넷으로 분할하고 각 서브넷에 NAT 게이트웨이를 만들어 워크로드를 분산할 수 있습니다.
- 하나의 탄력적 IP 주소만 NAT 게이트웨이에 연결할 수 있습니다. 연결된 후에는 NAT 게이트웨이에서 탄력적 IP 주소의 연결을 끊을 수 없습니다. NAT 게이트웨이에 다른 탄력적 IP 주소를 사용해야 하는 경우, 필요한 주소가 있는 새로운 NAT 게이트웨이를 만들고 라우팅 테이블을 업데이트한 다음 더 이상 필요하지 않은 경우 기존 NAT 게이트웨이를 삭제해야 합니다.
- NAT 게이트웨이는 TCP, UDP, ICMP 등의 프로토콜을 지원합니다.
- 보안 그룹을 NAT 게이트웨이와 연결할 수 없습니다. 프라이빗 서브넷의 인스턴스에 대한 보안 그룹을 사용하여 해당 인스턴스에서 주고받는 트래픽을 제어할 수 있습니다.
- 네트워크 ACL을 사용하여 NAT 게이트웨이가 위치하고 있는 서브넷에서 주고받는 트래픽을 제어할 수 있습니다. 네트워크 ACL은 NAT 게이트웨이의 트래픽에 적용됩니다. NAT 게이트웨이는 포트 1024 - 65535를 사용합니다. 자세한 내용은 [네트워크 ACL \(p. 102\)](#) 단원을 참조하십시오.

- NAT 게이트웨이가 생성되면 서브넷의 IP 주소 범위에 속하는 프라이빗 IP 주소가 자동으로 할당된 네트워크 인터페이스를 받습니다. Amazon EC2 콘솔에서 NAT 게이트웨이의 네트워크 인터페이스를 볼 수 있습니다. 자세한 내용은 [탄력적 네트워크 인터페이스에 대한 세부 정보 보기](#) 단원을 참조하십시오. 이 네트워크 인터페이스의 속성을 수정할 수 없습니다.
- VPC와 연결된 ClassicLink 연결을 통해 NAT 게이트웨이에 액세스할 수 없습니다.

다음 다이어그램은 NAT 게이트웨이가 있는 VPC의 아키텍처를 보여 줍니다. 기본 라우팅 테이블은 프라이빗 서브넷의 인스턴스에서 NAT 게이트웨이로 인터넷 트래픽을 보냅니다. NAT 게이트웨이는 NAT 게이트웨이의 탄력적 IP 주소를 소스 IP 주소로 사용하여 인터넷 게이트웨이로 트래픽을 보냅니다.

#### 항목

- [NAT 인스턴스에서 마이그레이션](#) (p. 174)
- [VPC 종단점, VPN, AWS Direct Connect 또는 VPC 피어링이 있는 NAT 게이트웨이 사용](#) (p. 174)

## NAT 인스턴스에서 마이그레이션

이미 NAT 인스턴스를 사용하는 경우 이를 NAT 게이트웨이로 대체할 수 있습니다. 이렇게 하려면 NAT 인스턴스와 동일한 서브넷에 NAT 게이트웨이를 만든 다음, NAT 인스턴스를 가리키는 라우팅 테이블의 기존 경로를 NAT 게이트웨이를 가리키는 경로로 대체합니다. NAT 인스턴스에 현재 사용하는 것과 동일한 탄력적 IP 주소를 NAT 게이트웨이에 사용하려는 경우, 먼저 NAT 인스턴스의 탄력적 IP 주소를 연결 해제하고 NAT 게이트웨이를 만들 때 이 주소를 게이트웨이에 연결해야 합니다.

#### Note

NAT 인스턴스에서 NAT 게이트웨이로 라우팅을 변경하거나 NAT 인스턴스에서 탄력적 IP 주소의 연결을 해제하면 현재 연결이 끊어지고 연결을 다시 설정해야 합니다. 중요한 작업(또는 NAT 인스턴스를 통해 작동하는 기타 작업)이 실행 중이지 않은지 확인합니다.

## VPC 종단점, VPN, AWS Direct Connect 또는 VPC 피어링이 있는 NAT 게이트웨이 사용

NAT 게이트웨이는 VPC 종단점, VPN 연결, AWS Direct Connect 또는 VPC 피어링 연결을 통해 트래픽을 보낼 수 없습니다. 프라이빗 서브넷의 인스턴스가 VPC 종단점, VPN 연결 또는 AWS Direct Connect를 통해 리소스에 액세스해야 하는 경우, 프라이빗 서브넷의 라우팅 테이블을 사용하여 이 디바이스로 트래픽을 직접 라우팅합니다.

예를 들어, 프라이빗 서브넷의 라우팅 테이블에 다음 경로가 있는 경우 인터넷 바운드 트래픽(0.0.0.0/0)은 NAT 게이트웨이로 라우팅되고, Amazon S3 트래픽(pl-xxxxxxx, Amazon S3에 대한 특정 IP 주소 범위)은 VPC 종단점으로 라우팅되며, 10.25.0.0/16 트래픽은 VPC 피어링 연결로 라우팅됩니다. pl-xxxxxxx 및 10.25.0.0/16 IP 주소 범위는 0.0.0.0/0보다 구체적입니다. 인스턴스에서 Amazon S3 또는 피어링된 VPC로 트래픽을 보내면 트래픽은 VPC 종단점 또는 VPC 피어링 연결로 전송됩니다. 인스턴스에서 인터넷(Amazon S3 IP 주소 제외)으로 트래픽을 보내면 트래픽은 NAT 게이트웨이로 전송됩니다.

VPC 피어링 연결, VPN 연결 또는 AWS Direct Connect를 통해 NAT 게이트웨이로 트래픽을 라우팅할 수 없습니다. 이 연결의 다른 쪽에 있는 리소스는 NAT 게이트웨이를 사용할 수 없습니다.

## NAT 게이트웨이 사용

Amazon VPC 콘솔을 사용하여 NAT 게이트웨이를 만들고 보고 삭제할 수 있습니다. 또한 Amazon VPC 마법사를 사용하여 퍼블릭 서브넷, 프라이빗 서브넷 및 NAT 게이트웨이가 있는 VPC를 만들 수 있습니다. 자세한 내용은 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\)](#) (p. 23) 단원을 참조하십시오.

#### 항목

- [NAT 게이트웨이 만들기](#) (p. 175)
- [라우팅 테이블 업데이트](#) (p. 175)

- [NAT 게이트웨이 삭제 \(p. 175\)](#)
- [NAT 게이트웨이 테스트 \(p. 176\)](#)

## NAT 게이트웨이 만들기

NAT 게이트웨이를 만들려면 서브넷과 탄력적 IP 주소를 지정해야 합니다. 탄력적 IP 주소가 현재 인스턴스 또는 네트워크 인터페이스에 연결되어 있지 않은지 확인합니다. NAT 인스턴스에서 NAT 게이트웨이로 마이그레이션할 때 NAT 인스턴스의 탄력적 IP 주소를 재사용하려는 경우 먼저 NAT 인스턴스에서 해당 주소를 연결 해제해야 합니다.

NAT 게이트웨이를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [NAT Gateways], [Create NAT Gateway]를 선택합니다.
3. 대화 상자에서 NAT 게이트웨이를 만들 서브넷을 지정하고 NAT 게이트웨이와 연결할 탄력적 IP 주소를 선택합니다. 완료하면 [Create a NAT Gateway]를 선택합니다.
4. NAT 게이트웨이가 콘솔에 표시됩니다. 몇 분 후 상태가 `Available`로 변경되면 사용할 준비가 된 것입니다.

NAT 게이트웨이가 `Failed` 상태로 바뀌면 만드는 중에 오류가 발생한 것입니다. 자세한 내용은 [NAT 게이트웨이가 실패 상태로 바뀔 \(p. 178\)](#) 단원을 참조하십시오.

## 라우팅 테이블 업데이트

NAT 게이트웨이를 만든 후에는 인터넷 트래픽이 NAT 게이트웨이를 가리키도록 프라이빗 서브넷의 라우팅 테이블을 업데이트해야 합니다. Amazon은 LPM(Longest Prefix Match)을 통해 트래픽과 일치하는 고도로 구체적인 라우팅을 사용하여 트래픽의 라우팅 방법을 결정합니다. 자세한 내용은 [라우팅 우선순위 \(p. 156\)](#) 단원을 참조하십시오.

NAT 게이트웨이에 대한 경로를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. 프라이빗 서브넷과 연결된 라우팅 테이블을 선택하고 [Routes], [Edit]를 선택합니다.
4. [Add another route]를 선택합니다. [Destination]에 `0.0.0.0/0`을 입력합니다. [Target]에서 NAT 게이트웨이의 ID를 선택합니다.

### Note

NAT 인스턴스 사용에서 마이그레이션하는 경우 NAT 인스턴스를 가리키는 현재 경로를 NAT 게이트웨이에 대한 경로로 바꿀 수 있습니다.

5. [Save]를 선택합니다.

NAT 게이트웨이가 인터넷에 액세스할 수 있도록 하려면, NAT 게이트웨이가 속한 서브넷과 연결된 라우팅 테이블에 인터넷 트래픽이 인터넷 게이트웨이를 가리키는 경로가 포함되어야 합니다. 자세한 내용은 [사용자 지정 라우팅 테이블 생성 \(p. 167\)](#) 단원을 참조하십시오. NAT 게이트웨이를 삭제하면 NAT 게이트웨이 경로는 경로를 삭제하거나 업데이트할 때까지 `blackhole` 상태로 유지됩니다. 자세한 내용은 [라우팅 테이블에 경로 추가 및 라우팅 테이블에서 경로 제거 \(p. 161\)](#) 단원을 참조하십시오.

## NAT 게이트웨이 삭제

Amazon VPC 콘솔을 사용하여 NAT 게이트웨이를 삭제할 수 있습니다. NAT 게이트웨이를 삭제한 후 잠시 동안(일반적으로 한 시간) Amazon VPC 콘솔에 항목이 표시되며 그 이후 자동으로 제거됩니다. 이 항목을 직접 제거할 수는 없습니다.

## NAT 게이트웨이를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [NAT Gateways]를 선택합니다.
3. NAT 게이트웨이를 선택한 다음 [Delete NAT Gateway]를 선택합니다.
4. 확인 대화 상자에서 [Delete NAT Gateway]를 선택합니다.

## NAT 게이트웨이 테스트

NAT 게이트웨이를 만들고 라우팅 테이블을 업데이트한 후에는 프라이빗 서브넷의 인스턴스에서 인터넷을 ping하여 인터넷에 연결할 수 있는지 테스트할 수 있습니다. 이렇게 하는 방법의 예는 [인터넷 연결 테스트 \(p. 176\)](#) 섹션을 참조하십시오.

인터넷에 연결할 수 있는 경우 다음 테스트를 수행하여 인터넷 트래픽이 NAT 게이트웨이를 통해 라우팅되는지도 확인할 수 있습니다.

- 프라이빗 서브넷의 인스턴스에서 전송되는 트래픽의 경로를 추적할 수 있습니다. 이렇게 하려면 프라이빗 서브넷의 Linux 인스턴스에서 `traceroute` 명령을 실행합니다. 출력에서 홉 중 하나(일반적으로 첫 번째 홉)에 NAT 게이트웨이의 프라이빗 IP 주소가 보여야 합니다.
- 프라이빗 서브넷의 인스턴스에서 소스 IP 주소에 연결할 때 이를 표시하는 타사 웹 사이트 또는 도구를 사용합니다. 소스 IP 주소는 NAT 게이트웨이의 탄력적 IP 주소여야 합니다. Amazon VPC 콘솔의 [NAT Gateways] 페이지에서 정보를 확인하여 NAT 게이트웨이의 탄력적 IP 주소 및 프라이빗 IP 주소를 가져올 수 있습니다.

위의 테스트가 실패하는 경우 [NAT 게이트웨이 문제 해결 \(p. 177\)](#)을 참조하십시오.

## 인터넷 연결 테스트

다음 예제는 프라이빗 서브넷의 인스턴스가 인터넷에 연결할 수 있는지 테스트하는 방법을 보여 줍니다.

1. 퍼블릭 서브넷의 인스턴스를 시작합니다(나중에 이 인스턴스를 접속 서버로 사용). 자세한 내용은 [서브넷에서 퍼블릭 IP 주소 시작 \(p. 68\)](#) 단원을 참조하십시오. 시작 마법사에서 Amazon Linux AMI를 선택하고 인스턴스에 퍼블릭 IP 주소를 할당해야 합니다. 보안 그룹 규칙이 로컬 네트워크의 IP 주소 범위에서 전송되는 인바운드 SSH 트래픽을 허용하고(이 테스트의 경우 `0.0.0.0/0`도 사용할 수 있음), 프라이빗 서브넷의 IP 주소 범위로 전송되는 아웃바운드 SSH 트래픽을 허용하는지 확인합니다.
2. 프라이빗 서브넷에서 인스턴스를 시작합니다. 시작 마법사에서 Amazon Linux AMI를 선택해야 합니다. 인스턴스에 퍼블릭 IP 주소를 할당하지 마십시오. 보안 그룹 규칙이 퍼블릭 서브넷에서 시작한 인스턴스의 프라이빗 IP 주소에서 전송되는 인바운드 SSH 트래픽 및 모든 아웃바운드 ICMP 트래픽을 허용하는지 확인합니다. 퍼블릭 서브넷에서 인스턴스를 시작하는 데 사용한 것과 동일한 키 페어를 선택해야 합니다.
3. 로컬 컴퓨터에서 SSH 에이전트 전달을 구성하고, 퍼블릭 서브넷의 접속 서버에 연결합니다. 자세한 내용은 [Linux 또는 OS X에 대한 SSH 에이전트 전달을 구성하려면 \(p. 176\)](#) 또는 [Windows\(PuTTY\)에 대한 SSH 에이전트 전달을 구성하려면 \(p. 177\)](#)을 참조하십시오.
4. 접속 서버에서 프라이빗 서브넷의 인스턴스에 연결한 다음, 프라이빗 서브넷의 인스턴스에서 인터넷 연결을 테스트합니다. 자세한 내용은 [인터넷 연결을 테스트하려면 \(p. 177\)](#) 단원을 참조하십시오.

## Linux 또는 OS X에 대한 SSH 에이전트 전달을 구성하려면

1. 로컬 시스템에서 인증 에이전트에 프라이빗 키를 추가합니다.

Linux의 경우 다음 명령을 사용합니다.

```
ssh-add -c mykeypair.pem
```

OS X의 경우 다음 명령을 사용합니다.

```
ssh-add -K mykeypair.pem
```

- SSH 에이전트 전달을 활성화하려면 `-A` 옵션을 사용하여 퍼블릭 서브넷의 인스턴스에 연결하고 해당 인스턴스의 퍼블릭 주소를 사용합니다. 예를 들면 다음과 같습니다.

```
ssh -A ec2-user@54.0.0.123
```

Windows(PuTTY)에 대한 SSH 에이전트 전달을 구성하려면

- Pageant가 아직 설치되어 있지 않으면 [PuTTY 다운로드 페이지](#)에서 Pageant를 다운로드하여 설치합니다.
- 프라이빗 키를 .ppk 형식으로 변환합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [PuTTYgen을 사용하여 프라이빗 키 변환](#)을 참조하십시오.
- Pageant를 시작하고 작업 표시줄의 Pageant 아이콘을 마우스 오른쪽 버튼으로 클릭한 다음 [Add Key]를 선택합니다. 생성한 .ppk 파일을 선택하고 필요한 경우 암호를 입력한 다음 [Open]을 선택합니다.
- PuTTY 세션을 시작하고 퍼블릭 IP 주소를 사용하여 퍼블릭 서브넷의 인스턴스에 연결합니다. 자세한 내용은 [PuTTY 세션 시작](#)을 참조하십시오. [Auth] 범주에서 [Allow agent forwarding] 옵션을 선택하고 [Private key file for authentication] 필드를 공백 상태로 둡니다.

인터넷 연결을 테스트하려면

- 퍼블릭 서브넷의 인스턴스에서 프라이빗 IP 주소를 사용하여 프라이빗 서브넷의 인스턴스에 연결합니다. 예를 들면 다음과 같습니다.

```
ssh ec2-user@10.0.1.123
```

- 프라이빗 인스턴스에서 ICMP를 활성화한 웹 사이트에 대해 ping 명령을 실행하여 인터넷에 연결할 수 있는지 테스트합니다. 예를 들면 다음과 같습니다.

```
ping ietf.org
```

```
PING ietf.org (4.31.198.44) 56(84) bytes of data:  
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=47 time=86.0 ms  
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=47 time=75.6 ms  
...
```

키보드에서 [Ctrl+C]를 눌러 ping 명령을 취소합니다. ping 명령이 실패할 경우 [프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 없음 \(p. 179\)](#)을 참조하십시오.

- (선택 사항) 더 이상 필요하지 않으면 인스턴스를 종료합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [인스턴스 종료](#)를 참조하십시오.

## NAT 게이트웨이 문제 해결

다음 주제는 NAT 게이트웨이를 만들거나 사용할 때 발생할 수 있는 일반적인 문제를 해결하는 데 도움이 될 수 있습니다.

항목

- [NAT 게이트웨이가 실패 상태로 바뀜 \(p. 178\)](#)
- [탄력적 IP 주소 또는 NAT 게이트웨이 제한에 도달함 \(p. 179\)](#)
- [가용 영역이 지원되지 않음\(NotAvailableInZone\) \(p. 179\)](#)
- [NAT 게이트웨이를 만들었지만 게이트웨이가 더 이상 표시되지 않음 \(p. 179\)](#)



- NAT 게이트웨이가 Ping 명령에 응답하지 않음 (p. 179)
- 프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 없음 (p. 179)
- 특정 엔드포인트에 대한 TCP 연결이 실패 (p. 180)
- 경로 추적 출력에 NAT 게이트웨이 프라이빗 IP 주소가 표시되지 않음 (p. 181)
- 5분 후 인터넷 연결이 끊어짐 (p. 181)
- IPSec 연결을 설정할 수 없음 (p. 181)
- 대상에 대한 추가 연결을 시작할 수 없음 (p. 181)

## NAT 게이트웨이가 실패 상태로 바뀜

NAT 게이트웨이를 만들 때 게이트웨이가 `Failed` 상태로 바뀌는 경우 게이트웨이를 만들 때 오류가 발생한 것입니다. 오류 메시지를 보려면 Amazon VPC 콘솔로 이동하여 [NAT Gateways]를 선택하고 NAT 게이트웨이를 선택한 다음, 세부 정보 창의 [Status] 필드에서 오류 메시지를 봅니다.

### Note

실패한 NAT 게이트웨이는 잠시 후(일반적으로 약 한 시간 후) 자동으로 삭제됩니다.

다음 표에 Amazon VPC 콘솔에 표시되는 오류의 예상 원인이 나와 있습니다. 표시된 수정 단계를 적용한 후 NAT 게이트웨이를 다시 만들어 볼 수 있습니다.

표시된 오류	이유	수정 단계
서브넷에 이 NAT 게이트웨이를 만들 수 있는 사용 가능한 주소가 부족함	지정한 서브넷에 사용 가능한 프라이빗 IP 주소가 없습니다. NAT 게이트웨이에는 서브넷의 범위에서 할당된 프라이빗 IP 주소가 있는 네트워크 인터페이스가 필요합니다.	Amazon VPC 콘솔의 [Subnets] 페이지로 이동하여 서브넷에 대한 세부 정보 창의 [Available IPs] 필드를 보면 서브넷에서 사용 가능한 IP 주소의 수를 확인할 수 있습니다. 서브넷에서 사용 가능한 IP 주소를 만들려면 사용되지 않은 네트워크 인터페이스를 삭제하거나 필요 없는 인스턴스를 종료할 수 있습니다.
네트워크 vpc-xxxxxxx에 연결된 인터넷 게이트웨이가 없음	인터넷 게이트웨이가 있는 VPC에서 NAT 게이트웨이를 만들어야 합니다.	인터넷 게이트웨이를 생성하여 VPC에 연결합니다. 자세한 내용은 <a href="#">인터넷 게이트웨이 연결 (p. 167)</a> 단원을 참조하십시오.
탄력적 IP 주소 eipalloc-xxxxxxx를 이 NAT 게이트웨이와 연결할 수 없음	지정한 탄력적 IP 주소가 없거나 해당 주소를 찾을 수 없습니다.	탄력적 IP 주소의 할당 ID를 검사하여 올바르게 입력했는지 확인합니다. NAT 게이트웨이를 만드는 리전과 동일한 리전에 있는 탄력적 IP 주소를 지정했는지 확인합니다.
탄력적 IP 주소 eipalloc-xxxxxxx가 이미 연결되어 있음	지정한 탄력적 IP 주소가 다른 리소스와 이미 연결되어 있으므로 NAT 게이트웨이와 연결할 수 없습니다.	Amazon VPC 콘솔의 [Elastic IPs] 페이지로 이동하여 인스턴스 ID 또는 네트워크 인터페이스 ID에 지정한 값을 보면 탄력적 IP 주소와 연결된 리소스를 확인할 수 있습니다. 해당 리소스에 탄력적 IP 주소가 필요 없는 경우 주소를 연결 해제할 수 있습니다. 또는 계정에 새로운 탄력적 IP 주소를 할당합니다. 자세한 내용은 <a href="#">탄력적 IP</a>



표시된 오류	이유	수정 단계
		<a href="#">주소 작업 (p. 198)</a> 단원을 참조하십시오.
이 NAT 게이트웨이에서 내부적으로 생성되고 사용되는 네트워크 인터페이스 eni-xxxxxxx가 잘못된 상태입니다. 다시 시도하십시오.	NAT 게이트웨이에 대한 네트워크 인터페이스를 만들거나 사용하는 중에 문제가 발생했습니다.	이 오류를 수정할 수 없습니다. NAT 게이트웨이를 다시 만들어 보십시오.

## 탄력적 IP 주소 또는 NAT 게이트웨이 제한에 도달함

탄력적 IP 주소 제한에 도달한 경우 다른 리소스에서 탄력적 IP를 연결 해제하거나 [Amazon VPC 제한 양식](#)을 사용하여 제한 증가를 요청할 수 있습니다.

NAT 게이트웨이 제한에 도달한 경우 다음 중 하나를 수행할 수 있습니다.

- [Amazon VPC 제한 양식](#)을 사용하여 제한 증가를 요청합니다. NAT 게이트웨이 제한은 가용 영역별로 적용됩니다.
- NAT 게이트웨이의 상태를 확인합니다. Pending, Available 또는 Deleting의 상태는 제한에 포함됩니다. NAT 게이트웨이를 최근에 삭제한 경우 상태가 Deleting에서 Deleted로 바뀔 때까지 몇 분간 기다린 다음 새 NAT 게이트웨이를 만들어 봅니다.
- 특정 가용 영역에 NAT 게이트웨이가 필요 없는 경우 제한에 도달하지 않은 가용 영역에서 NAT 게이트웨이를 만들어 봅니다.

제한에 대한 자세한 내용은 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

## 가용 영역이 지원되지 않음(NotAvailableInZone)

경우에 따라 제약이 있는 가용 영역(확장이 제약되어 있는 영역)에서 NAT 게이트웨이를 만드는 것일 수 있습니다. 이러한 영역에서는 NAT 게이트웨이를 지원하지 않습니다. 다른 가용 영역에서 NAT 게이트웨이를 만들고 제약이 있는 영역의 프라이빗 서브넷에 사용할 수 있습니다. 리소스와 NAT 게이트웨이가 동일한 가용 영역에 있도록 제약이 없는 가용 영역으로 리소스를 이동할 수도 있습니다.

## NAT 게이트웨이를 만들었지만 게이트웨이가 더 이상 표시되지 않음

NAT 게이트웨이를 만드는 중에 오류가 발생했을 수 있으며, 게이트웨이를 만들지 못했습니다. failed 상태의 NAT 게이트웨이는 VPC 콘솔에 잠시 동안(일반적으로 한 시간) 표시되며 그 이후 자동으로 삭제됩니다. [NAT 게이트웨이가 실패 상태로 바뀐 \(p. 178\)](#)에서 정보를 검토하고 새 NAT 게이트웨이를 만들어 봅니다.

## NAT 게이트웨이가 Ping 명령에 응답하지 않음

인터넷에서(예를 들어 홈 컴퓨터에서) 또는 VPC의 인스턴스에서 NAT 게이트웨이의 탄력적 IP 주소 또는 프라이빗 IP 주소를 ping하려고 시도하는 경우 응답을 얻을 수 없습니다. NAT 게이트웨이는 프라이빗 서브넷의 인스턴스에서 인터넷으로만 트래픽을 전달합니다.

NAT 게이트웨이가 작동하는지 테스트하려면 [NAT 게이트웨이 테스트 \(p. 176\)](#)를 참조하십시오.

## 프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 없음

위의 NAT 게이트웨이 테스트 단계를 수행한 후 ping 명령이 실패하거나 인스턴스가 인터넷에 액세스할 수 없는 경우 다음 정보를 확인합니다.

- NAT 게이트웨이가 Available 상태인지 확인합니다. Amazon VPC 콘솔에서 [NAT Gateways] 페이지로 이동하고 세부 정보 창에서 상태 정보를 봅니다. NAT 게이트웨이가 실패 상태인 경우 게이트웨이가 생성될 때 오류가 발생했을 수 있습니다. 자세한 내용은 [NAT 게이트웨이가 실패 상태로 바뀐 \(p. 178\)](#) 단원을 참조하십시오.

- 라우팅 테이블을 올바르게 구성했는지 확인합니다:
  - NAT 게이트웨이는 인터넷 트래픽을 인터넷 게이트웨이로 라우팅하는 라우팅 테이블이 있는 퍼블릭 서브넷에 있어야 합니다. 자세한 내용은 [사용자 지정 라우팅 테이블 생성 \(p. 167\)](#) 단원을 참조하십시오.
  - 인스턴스는 인터넷 트래픽을 NAT 게이트웨이로 라우팅하는 라우팅 테이블이 있는 프라이빗 서브넷에 있어야 합니다. 자세한 내용은 [라우팅 테이블 업데이트 \(p. 175\)](#) 단원을 참조하십시오.
  - 전체 또는 일부 인터넷 트래픽을 NAT 게이트웨이 대신 다른 디바이스로 라우팅하는 다른 라우팅 테이블 항목이 있는지 확인합니다.
- 프라이빗 인스턴스에 대한 보안 그룹 규칙이 아웃바운드 인터넷 트래픽을 허용하는지 확인합니다. ping 명령이 작동하려면 규칙이 아웃바운드 ICMP 트래픽도 허용해야 합니다.

#### Note

NAT 게이트웨이 자체는 모든 아웃바운드 트래픽과 아웃바운드 요청에 대한 응답으로 받는 트래픽을 허용합니다(따라서 상태 저장).

- 프라이빗 서브넷 및 퍼블릭 서브넷과 연결된 네트워크 ACL에 인바운드 또는 아웃바운드 인터넷 트래픽을 차단하는 규칙이 없는지 확인합니다. ping 명령이 작동하려면 규칙이 인바운드 및 아웃바운드 ICMP 트래픽도 허용해야 합니다.

#### Note

흐름 로그를 활성화하여 네트워크 ACL 또는 보안 그룹 규칙으로 인해 끊어진 연결을 진단할 수 있습니다. 자세한 내용은 [VPC 흐름 로그 \(p. 144\)](#) 단원을 참조하십시오.

- ping 명령을 사용하는 경우 ICMP가 활성화된 웹 사이트를 ping하고 있는지 확인합니다. 그렇지 않으면 수신 패킷이 수신되지 않을 것입니다. 이를 테스트하려면 사용자 자신의 컴퓨터의 명령줄 터미널에서 똑같은 ping 명령을 수행하십시오.
- 인스턴스가 다른 리소스, 예를 들어 프라이빗 서브넷의 다른 인스턴스를 ping할 수 있는지 확인합니다(보안 그룹 규칙이 이 작업을 허용한다고 가정함).
- 연결이 TCP, UDP 또는 ICMP 프로토콜만 사용하는지 확인합니다.

## 특정 엔드포인트에 대한 TCP 연결이 실패

다른 엔드포인트에 대한 TCP 연결은 정상적으로 작동하지만 특정 엔드포인트 또는 호스트에 대한 TCP 연결이 실패하는 경우 연결하려는 엔드포인트가 조각화된 TCP 패킷에 응답하는지 확인합니다. 현재 NAT 게이트웨이는 TCP에 대한 IP 조각화를 지원하지 않습니다. 자세한 내용은 [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 188\)](#) 단원을 참조하십시오.

엔드포인트가 조각화된 TCP 패킷을 전송하는지 확인하려면 퍼블릭 IP 주소를 갖는 퍼블릭 서브넷의 인스턴스를 사용하여 다음을 수행합니다.

- 특정 엔드포인트로부터 조각화를 유발할 정도로 큰 응답을 트리거합니다.
- tcpdump 유틸리티를 사용하여 엔드포인트가 조각화된 패킷을 전송하는지 확인합니다.

#### Important

퍼블릭 서브넷의 인스턴스를 사용하여 이러한 확인을 수행해야 합니다. 원래 연결이 실패한 인스턴스, NAT 게이트웨이 뒤 프라이빗 서브넷의 인스턴스 또는 NAT 인스턴스는 사용할 수 없습니다.

엔드포인트가 조각화된 TCP 패킷을 전송하는 경우 NAT 게이트웨이 대신 NAT 인스턴스를 사용할 수 있습니다.

#### Note

NAT 게이트웨이는 ICMP 프로토콜에 대한 IP 조각화도 지원하지 않습니다. 대량 ICMP 패킷을 전송 또는 수신하는 진단 도구가 패킷 손실을 보고할 것입니다. 예를 들어 NAT 게이트웨이 뒤에서는 `ping -s 10000 example.com` 명령이 작동하지 않습니다.

## 경로 추적 출력에 NAT 게이트웨이 프라이빗 IP 주소가 표시되지 않음

인스턴스가 인터넷에 액세스할 수 있지만, `tracert` 명령을 수행할 때 출력에 NAT 게이트웨이의 프라이빗 IP 주소가 표시되지 않습니다. 이 경우 인스턴스는 인터넷 게이트웨이와 같은 다른 디바이스를 사용하여 인터넷에 액세스하고 있습니다. 인스턴스가 위치하고 있는 서브넷의 라우팅 테이블에서 다음 정보를 확인합니다.

- 인터넷 트래픽을 NAT 게이트웨이로 보내는 경로가 있는지 확인합니다.
- 인터넷 트래픽을 가상 프라이빗 게이트웨이 또는 인터넷 게이트웨이와 같은 다른 디바이스로 보내는 보다 구체적인 경로가 없는지 확인합니다.

## 5분 후 인터넷 연결이 끊어짐

NAT 게이트웨이를 사용하는 연결이 5분 이상 유휴 상태인 경우 연결이 시간 초과됩니다. 연결을 통해 추가 트래픽을 시작하거나 TCP keepalive를 사용하여 연결이 끊어지지 않도록 할 수 있습니다.

## IPSec 연결을 설정할 수 없음

NAT 게이트웨이는 현재 IPSec 프로토콜을 지원하지 않습니다.

## 대상에 대한 추가 연결을 시작할 수 없음

동시 연결 제한에 도달했을 수 있습니다. NAT 게이트웨이는 각 고유 대상에 대해 최대 65,000개의 동시 연결을 지원할 수 있습니다. 프라이빗 서브넷의 인스턴스가 많은 양의 연결을 생성하는 경우 이 제한에 도달할 수 있습니다. 다음 중 하나를 수행할 수 있습니다.

- 가용 영역당 하나의 NAT 게이트웨이를 만들고 해당 영역에 클라이언트를 분산합니다.
- 퍼블릭 서브넷에서 추가 NAT 게이트웨이를 만들고 각각 다른 NAT 게이트웨이에 대한 경로가 있는 여러 프라이빗 서브넷으로 클라이언트를 분할합니다.
- 클라이언트가 대상에 대해 생성할 수 있는 연결 수를 제한합니다.
- 유휴 상태의 연결을 닫아서 용량을 해제합니다.

### Note

대상 IP 주소, 대상 포트 또는 프로토콜(TCP/UDP/ICMP)이 변경되는 경우 65,000개의 연결을 추가로 만들 수 있습니다.

## NAT 게이트웨이 사용 제어

기본적으로 IAM 사용자에게는 NAT 게이트웨이를 사용할 권한이 없습니다. 사용자에게 NAT 게이트웨이를 생성, 설명 및 삭제할 수 있는 권한을 부여하는 IAM 사용자 정책을 만들 수 있습니다. 현재는 `ec2:*NatGateway*` API 작업에 대한 리소스 수준 권한을 지원하지 않습니다. Amazon VPC의 IAM 정책에 대한 자세한 내용은 [Amazon VPC 리소스에 대한 액세스 제어 \(p. 130\)](#)를 참조하십시오.

## API 및 CLI 개요

이 페이지에서 설명한 작업은 명령줄이나 API를 사용하여 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#)를 참조하십시오.

### NAT 게이트웨이 만들기

- [create-nat-gateway](#)(AWS CLI)
- [New-EC2NatGateway](#)(Windows PowerShell용 AWS 도구)

- [CreateNatGateway](#)(Amazon EC2 쿼리 API)

#### NAT 게이트웨이 설명

- [describe-nat-gateways](#)(AWS CLI)
- [Get-EC2NatGateway](#)(Windows PowerShell용 AWS 도구)
- [DescribeNatGateways](#)(Amazon EC2 쿼리 API)

#### NAT 게이트웨이 삭제

- [delete-nat-gateway](#)(AWS CLI)
- [Remove-EC2NatGateway](#)(Windows PowerShell용 AWS 도구)
- [DeleteNatGateway](#)(Amazon EC2 쿼리 API)

## NAT 인스턴스

VPC의 퍼블릭 서브넷에 있는 네트워크 주소 변환(NAT) 인스턴스를 사용하여 프라이빗 서브넷에 있는 인스턴스가 인터넷 또는 다른 AWS 서비스로의 아웃바운드 IPv4 트래픽을 시작하되, 인터넷 상의 누군가가 시작한 인바운드 트래픽은 인스턴스가 수신하지 못하게 막을 수 있습니다.

퍼블릭 서브넷과 프라이빗 서브넷에 대한 자세한 내용은 [서브넷 라우팅 \(p. 64\)](#)을 참조하십시오. NAT에 대한 자세한 내용은 [NAT \(p. 172\)](#)를 참조하십시오.

NAT는 IPv6 트래픽을 지원하지 않으므로, 그 대신에 외부 전용 인터넷 게이트웨이를 사용하십시오. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 170\)](#) 단원을 참조하십시오.

#### Note

더 나은 가용성과 향상된 대역폭을 제공하면서도 관리 작업은 간소화하는 관리형 NAT 서비스인 NAT 게이트웨이를 사용할 수도 있습니다. 일반 사용 사례에서는 NAT 인스턴스보다 NAT 게이트웨이를 사용하는 것이 좋습니다. 자세한 내용은 [NAT 게이트웨이 \(p. 173\)](#) 및 [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 188\)](#) 단원을 참조하십시오.

#### 항목

- [NAT 인스턴스 기본 사항 \(p. 182\)](#)
- [NAT 인스턴스 설정 \(p. 183\)](#)
- [NATSG 보안 그룹 생성 \(p. 184\)](#)
- [원본/대상 확인 비활성화 \(p. 185\)](#)
- [기본 라우팅 테이블 업데이트 \(p. 186\)](#)
- [NAT 인스턴스 구성 테스트 \(p. 186\)](#)

## NAT 인스턴스 기본 사항

다음 그림에서는 NAT 인스턴스 기본 사항을 보여줍니다. 기본 라우팅 테이블은 프라이빗 서브넷의 인스턴스에서 퍼블릭 서브넷의 NAT 인스턴스로 트래픽을 전송합니다. NAT 인스턴스는 VPC용 인터넷 게이트웨이로 트래픽을 전송합니다. 트래픽은 NAT 인스턴스의 탄력적 IP 주소에 기인합니다. NAT 인스턴스는 응답에 대해 높은 포트 번호를 지정합니다. 즉, 응답이 되돌아오면 NAT 인스턴스가 응답에 대한 포트 번호를 기준으로 프라이빗 서브넷에 있는 인스턴스로 이 응답을 보냅니다.

Amazon은 NAT 인스턴스로 작동하도록 구성된 Amazon Linux AMI를 제공합니다. 이런 AMI에는 이름에 `amzn-ami-vpc-nat` 문자열이 포함되므로, Amazon EC2 콘솔에서 이런 AMI를 검색할 수 있습니다. NAT AMI에서 인스턴스를 시작하면 인스턴스에 다음과 같은 구성이 나타납니다.

- `/etc/sysctl.d/10-nat-settings.conf`에서 IPv4 전달을 사용하며 ICMP 리디렉션은 사용하지 않음
- 시작 시 `/usr/sbin/configure-pat.sh`에 위치한 스크립트가 실행되며 iptables IP 매스커레이딩을 구성

NAT 인스턴스 제한은 해당 리전의 인스턴스 유형 제한에 따라 다릅니다. 자세한 내용은 [EC2 FAQ](#)를 참조하십시오. 사용 가능한 NAT AMI 목록은 [Amazon Linux AMI matrix](#) 단원을 참조하십시오.

## NAT 인스턴스 설정

VPC 마법사를 사용하여 NAT 인스턴스가 있는 VPC를 설정할 수 있습니다. 자세한 내용은 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\)](#) (p. 23) 단원을 참조하십시오. 이 마법사는 NAT 인스턴스 시작 및 라우팅 설정을 포함한 여러 구성 단계를 자동으로 수행합니다. 하지만 사용자가 원할 경우 아래 단계에 따라 VPC와 NAT 인스턴스를 수동으로 생성하고 구성할 수 있습니다.

1. 두 개의 서브넷이 있는 VPC를 생성합니다.

### Note

아래 단계는 VPC 마법사를 사용하여 VPC를 생성하는 것이 아니라, VPC를 수동으로 생성하고 구성하기 위한 단계입니다.

- a. VPC를 생성합니다([VPC 만들기](#) (p. 66) 참조).
  - b. 두 개의 서브넷을 생성합니다([서브넷 만들기](#) (p. 167) 참조).
  - c. 인터넷 게이트웨이를 VPC에 연결합니다([인터넷 게이트웨이 연결](#) (p. 167) 참조).
  - d. VPC 외부 위치를 목적지로 하는 트래픽을 인터넷 게이트웨이로 보내는 사용자 지정 라우팅 테이블을 생성한 다음, 이를 한 서브넷과 연결하여 퍼블릭 서브넷으로 만듭니다([사용자 지정 라우팅 테이블 생성](#) (p. 167) 참조).
2. NATSG 보안 그룹을 생성합니다([NATSG 보안 그룹 생성](#) (p. 184) 참조). NAT 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.
  3. NAT 인스턴스로 실행하도록 구성된 AMI에서 퍼블릭 서브넷으로 인스턴스를 시작합니다. Amazon은 NAT 인스턴스로 작동하도록 구성된 Amazon Linux AMI를 제공합니다. 이런 AMI에는 이름에 `amzn-ami-vpc-nat` 문자열이 포함되므로, Amazon EC2 콘솔에서 이런 AMI를 검색할 수 있습니다.
    - a. Amazon EC2 콘솔을 엽니다.
    - b. 대시보드에서 [Launch Instance] 버튼을 선택하고 다음과 같이 마법사가 안내하는 단계를 완료하십시오.
      - i. [Choose an Amazon Machine Image (AMI)] 페이지에서 [Community AMIs] 범주를 선택하고 `amzn-ami-vpc-nat`를 검색합니다. 결과 목록에서 각 AMI의 이름에는 가장 최근의 AMI를 선택할 수 있는 버전이 포함됩니다(예: 2013.09). [Select]를 선택합니다.
      - ii. [Choose an Instance Type] 페이지에서 인스턴스 유형과 [Next: Configure Instance Details]를 차례로 선택합니다.
      - iii. [Configure Instance Details] 페이지의 [Network] 목록에서 사용자가 생성한 VPC를 선택하고 [Subnet] 목록에서 퍼블릭 서브넷을 선택합니다.
      - iv. (선택 사항) [Public IP] 확인란을 선택하여 NAT 인스턴스가 퍼블릭 IP 주소를 수신하도록 요청합니다. 지금 퍼블릭 IP 주소를 지정하지 않는 것으로 선택하는 경우 탄력적 IP 주소를 할당하고 인스턴스가 시작된 후에 이 주소를 인스턴스에 지정할 수 있습니다. 시작 시 퍼블릭 IP 배정에 대한 자세한 내용은 [인스턴스 시작 시 퍼블릭 IPv4 주소 지정](#) (p. 80) 단원을 참조하십시오. [Next: Add Storage]를 선택합니다.
      - v. 인스턴스에 스토리지를 추가하도록 선택할 수 있고, 다음 페이지에서 태그를 추가할 수 있습니다. 모두 마쳤으면 [Next: Configure Security Group]을 선택합니다.
      - vi. [Configure Security Group] 페이지에서 [Select an existing security group] 옵션을 선택하고, 사용자가 생성한 NATSG 보안 그룹을 선택합니다. [Review and Launch]를 선택합니다.
      - vii. 선택한 설정을 검토합니다. 필요한 사항을 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

4. (선택 사항) NAT 인스턴스에 연결하고, 필요한 수정 작업을 수행한 다음, NAT 인스턴스로 작동하도록 구성된 자체 AMI를 생성합니다. 다음에 NAT 인스턴스를 시작할 필요가 있을 때 이 AMI를 사용할 수 있습니다. AMI 생성에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EBS 지원 AMI 생성 단원](#)을 참조하십시오.
5. NAT 인스턴스에 대한 `SrcDestCheck` 속성을 비활성화합니다([원본/대상 확인 비활성화 \(p. 185\)](#) 참조).
6. 시작 도중에(3단계) 퍼블릭 IP 주소를 NAT 인스턴스에 지정하지 않았다면, 탄력적 IP 주소를 NAT 인스턴스와 연결할 필요가 있습니다.
  - a. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
  - b. 탐색 창에서 [Elastic IPs]와 [Allocate New Address]를 차례로 선택합니다.
  - c. [Allocate]를 선택합니다.
  - d. 목록에서 탄력적 IP 주소를 선택한 다음, [Actions], [Associate Address]를 선택합니다.
  - e. 네트워크 인터페이스 리소스를 선택한 다음 NAT 인스턴스에 대한 네트워크 인터페이스를 선택합니다. [Private IP] 목록에서 탄력적 IP 주소와 연결할 주소를 선택한 다음 [Associate]를 선택합니다.
7. NAT 인스턴스로 트래픽을 보내기 위해 기본 라우팅 테이블을 업데이트합니다. 자세한 내용은 [기본 라우팅 테이블 업데이트 \(p. 186\)](#) 단원을 참조하십시오.

## 명령줄을 사용하여 NAT 인스턴스 시작

서브넷으로 NAT 인스턴스를 시작하려면 다음 명령 중 하나를 사용합니다. 다음의 명령줄 인터페이스에 대한 자세한 정보는 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

- `run-instances`(AWS CLI)
- `New-EC2Instance`(Windows PowerShell용 AWS 도구)

NAT 인스턴스로 작동하도록 구성된 AMI의 ID를 가져오려면 이미지를 설명하기 위한 명령을 사용하고 Amazon이 소유하고 있고 이름에 `amzn-ami-vpc-nat` 문자열이 포함된 AMI에 대한 결과만 반환하는 필터를 사용합니다. 다음 예제에서는 AWS CLI를 사용합니다.

```
PROMPT> aws ec2 describe-images --filter Name="owner-alias",Values="amazon" --filter Name="name",Values="amzn-ami-vpc-nat"
```

## NATSG 보안 그룹 생성

다음 표에 설명되어 있는 것처럼 NATSG 보안 그룹을 정의하여 NAT 인스턴스가 프라이빗 서브넷에 있는 인스턴스로부터의 인터넷 바인딩 트래픽뿐 아니라, 네트워크로부터의 SSH 트래픽도 수신할 수 있도록 합니다. 또한, NAT 인스턴스는 인터넷으로 트래픽을 전송할 수 있으며 따라서 프라이빗 서브넷의 인스턴스가 소프트웨어 업데이트를 받을 수 있습니다.

NATSG: 권장 규칙

인바운드			
소스	프로토콜	포트 범위	설명
10.0.1.0/24	TCP	80	프라이빗 서브넷의 서버로부터의 인바운드 HTTP 트래픽 허용
10.0.1.0/24	TCP	443	프라이빗 서브넷의 서버로부터의 인바운드 HTTPS 트래픽 허용
홈 네트워크의 공인 IP 주소 범위	TCP	22	홈 네트워크로부터 NAT 인스턴스에 대한 인바운드 SSH 액세스 허용(인터넷 게이트웨이를 통해)

아웃바운드			
목적지	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	인터넷에 대한 아웃바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	인터넷에 대한 아웃바운드 HTTPS 액세스 허용

#### NATSG 보안 그룹을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택한 다음, [Create Security Group]을 선택합니다.
3. [Create Security Group] 대화 상자에서 보안 그룹의 이름을 NATSG로 지정하고 설명을 입력합니다. [VPC] 목록에서 VPC ID를 선택한 다음 [Yes, Create]를 선택합니다.
4. 방금 만든 NATSG 보안 그룹을 선택합니다. 세부 정보 창에는 인바운드 및 아웃바운드 규칙 작업을 위한 탭과 함께 보안 그룹에 대한 세부 정보가 표시됩니다.
5. 다음과 같이 [Inbound Rules] 탭을 사용하여 인바운드 트래픽에 대한 규칙을 추가합니다.
  - a. [Edit]를 선택합니다.
  - b. [Add another rule]을 선택하고, [Type] 목록에서 [HTTP]를 선택합니다. [Source] 필드에 프라이빗 서브넷의 IP 주소 범위를 지정합니다.
  - c. [Add another rule]을 선택하고, [Type] 목록에서 [HTTPS]를 선택합니다. [Source] 필드에 프라이빗 서브넷의 IP 주소 범위를 지정합니다.
  - d. [Add another rule]을 선택하고, [Type] 목록에서 [SSH]를 선택합니다. [Source] 필드에 네트워크의 퍼블릭 IP 주소 범위를 지정합니다.
  - e. [Save]를 선택합니다.
6. 다음과 같이 [Outbound Rules] 탭을 사용하여 아웃바운드 트래픽에 대한 규칙을 추가합니다.
  - a. [Edit]를 선택합니다.
  - b. [Add another rule]을 선택하고, [Type] 목록에서 [HTTP]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 지정합니다.
  - c. [Add another rule]을 선택하고, [Type] 목록에서 [HTTPS]를 선택합니다. [Destination] 필드에 0.0.0.0/0을 지정합니다.
  - d. [Save]를 선택합니다.

보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참조하십시오.

## 원본/대상 확인 비활성화

각각의 EC2 인스턴스는 기본적으로 원본/대상 확인을 수행합니다. 이는 인스턴스가 보내거나 받는 트래픽의 원본 또는 대상이어야 한다는 의미입니다. 하지만, NAT 인스턴스는 원본 또는 대상이 그 자신이 아닐 때 트래픽을 보내고 받을 수 있어야 합니다. 따라서 NAT 인스턴스에서 원본/대상 확인을 비활성화해야 합니다.

콘솔 또는 명령줄을 사용하여 실행 중이거나 중지된 NAT 인스턴스에 대해 SrcDestCheck 속성을 비활성화할 수 있습니다.

콘솔을 사용하여 원본/대상 확인을 비활성화하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.



3. NAT 인스턴스를 선택하고 [Actions]를 선택하고 [Networking]을 선택한 다음, [Change Source/Dest. Check]를 선택합니다.
4. NAT 인스턴스에 대해 이 속성이 비활성화되어 있는지 확인합니다. 그렇지 않으면 [Yes, Disable]을 선택합니다.

명령줄을 사용하여 원본/대상 확인을 비활성화하려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 섹션을 참조하십시오.

- `modify-instance-attribute` (AWS CLI)
- `Edit-EC2InstanceAttribute` (Windows PowerShell용 AWS 도구)

## 기본 라우팅 테이블 업데이트

다음 절차에 설명되어 있는 것처럼 기본 라우팅 테이블을 업데이트합니다. 기본적으로, 기본 라우팅 테이블을 통해 VPC의 인스턴스가 서로 통신할 수 있습니다. NAT 인스턴스로 다른 모든 서브넷 트래픽을 보내는 경로를 추가할 것입니다.

기본 라우팅 테이블을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택합니다.
3. VPC에 대한 기본 라우팅 테이블을 선택합니다. 세부 정보 창에는 경로, 연결 및 경로 전파 작업을 위한 탭이 표시됩니다.
4. [Routes] 탭에서 [Edit]를 선택하고, [Destination] 상자에서 `0.0.0.0/0`을 지정하고, [Target] 목록에서 NAT 인스턴스의 인스턴스 ID를 선택한 다음, [Save]를 선택합니다.
5. [Subnet Associations] 탭에서 [Edit]를 선택한 다음, 서브넷에 대한 [Associate] 확인란을 선택합니다. [Save]를 선택합니다.

라우팅 테이블에 대한 자세한 내용은 [라우팅 테이블 \(p. 154\)](#) 단원을 참조하십시오.

## NAT 인스턴스 구성 테스트

NAT 인스턴스를 시작하고 위의 구성 단계를 완료한 후, NAT 인스턴스를 접속 서버로 사용하여 프라이빗 서브넷의 인스턴스가 NAT 인스턴스를 통해 인터넷에 액세스할 수 있는지 확인하는 테스트를 수행할 수 있습니다. 이를 위해, 인바운드 및 아웃바운드 ICMP 트래픽과 아웃바운드 SSH 트래픽을 허용하도록 NAT 인스턴스의 보안 그룹 규칙을 업데이트하고, 프라이빗 서브넷으로 인스턴스를 시작하고, 프라이빗 서브넷에 있는 인스턴스에 액세스하도록 SSH 에이전트 전달을 구성하고, 인스턴스에 연결한 다음, 인터넷 연결을 테스트합니다.

NAT 인스턴스의 보안 그룹을 업데이트하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. NAT 인스턴스와 연결된 보안 그룹을 찾고, [Inbound] 탭에서 [Edit]를 선택합니다.
4. [Add Rule]을 선택하고, [Type] 목록에서 [All ICMP]를 선택하고, [Source] 목록에서 [Custom IP]를 선택합니다. 프라이빗 서브넷의 IP 주소 범위를 입력합니다(예: `10.0.1.0/24`). [Save]를 선택합니다.
5. [Outbound] 탭에서 [Edit]를 선택합니다.
6. [Add Rule]을 선택하고, [Type] 목록에서 [SSH]를 선택하고, [Source] 목록에서 [Custom IP]를 선택합니다. 프라이빗 서브넷의 IP 주소 범위를 입력합니다(예: `10.0.1.0/24`). [Save]를 선택합니다.
7. [Add Rule]을 선택하고, [Type] 목록에서 [All ICMP]를 선택하고, [Source] 목록에서 [Custom IP]를 선택합니다. `0.0.0.0/0`을 입력한 다음 [Save]를 선택합니다.



### 프라이빗 서브넷으로 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances를 선택합니다.
3. 프라이빗 서브넷으로 인스턴스를 시작합니다. 자세한 내용은 [서브넷에서 인스턴스 시작 \(p. 68\)](#) 단원을 참조하십시오. 시작 마법사에서 다음 옵션을 구성한 다음, [Launch]를 선택합니다.
  - [Choose an Amazon Machine Image (AMI)] 페이지의 [Quick Start] 범주에서 Amazon Linux AMI를 선택합니다.
  - [Configure Instance Details] 페이지의 [Subnet] 목록에서 프라이빗 서브넷을 선택하고, 인스턴스에 퍼블릭 IP 주소를 배정하지 마십시오.
  - [Configure Security Group] 페이지에서 보안 그룹에 NAT 인스턴스의 프라이빗 IP 주소에서 또는 퍼블릭 서브넷의 IP 주소 범위에서의 SSH 액세스를 허용하는 인바운드 규칙이 포함되어 있는지 확인하고, 아웃바운드 ICMP 트래픽을 허용하는 아웃바운드 규칙이 있는지 확인합니다.
  - [Select an existing key pair or create a new key pair] 대화 상자에서 NAT 인스턴스를 시작하는 데 사용한 것과 동일한 키 페어를 선택합니다.

### Linux 또는 OS X에 대한 SSH 에이전트 전달을 구성하려면

1. 로컬 시스템에서 인증 에이전트에 프라이빗 키를 추가합니다.

Linux의 경우 다음 명령을 사용합니다.

```
PROMPT> ssh-add -c mykeypair.pem
```

OS X의 경우 다음 명령을 사용합니다.

```
PROMPT> ssh-add -K mykeypair.pem
```

2. `-A` 옵션을 사용하여 NAT 인스턴스에 연결해 SSH 에이전트 전달을 활성화합니다. 예를 들면 다음과 같습니다.

```
ssh -A ec2-user@54.0.0.123
```

### Windows(PuTTY)에 대한 SSH 에이전트 전달을 구성하려면

1. Pageant가 아직 설치되어 있지 않으면 [PuTTY 다운로드 페이지](#)에서 Pageant를 다운로드하여 설치합니다.
2. 프라이빗 키를 .ppk 형식으로 변환합니다. 자세한 내용은 [PuTTYgen을 사용하여 프라이빗 키 변환](#) 단원을 참조하십시오.
3. Pageant를 시작하고 작업 표시줄의 Pageant 아이콘을 마우스 오른쪽 버튼으로 클릭한 다음 [Add Key]를 선택합니다. 생성한 .ppk 파일을 선택하고 필요한 경우 암호를 입력한 다음 [Open]을 선택합니다.
4. PuTTY 세션을 시작하여 NAT 인스턴스에 연결합니다. [Auth] 범주에서 [Allow agent forwarding] 옵션을 선택하고 [Private key file for authentication] 필드를 공백 상태로 둡니다.

### 인터넷 연결을 테스트하려면

1. ICMP를 활성화한 웹 사이트에 대해 ping 명령을 사용하여 NAT 인스턴스가 인터넷과 통신할 수 있는지 테스트합니다. 예를 들면 다음과 같습니다.

```
PROMPT> ping ietf.org
```

```
PING ietf.org (4.31.198.44) 56(84) bytes of data.
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=48 time=74.9 ms
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=48 time=75.1 ms
...
```

키보드에서 [Ctrl+C]를 눌러 ping 명령을 취소합니다.

2. NAT 인스턴스에서 프라이빗 IP 주소를 사용하여 프라이빗 서브넷의 인스턴스에 연결합니다. 예를 들면 다음과 같습니다.

```
PROMPT> ssh ec2-user@10.0.1.123
```

3. 프라이빗 인스턴스에서 ping 명령을 실행하여 인터넷에 연결할 수 있는지 테스트합니다.

```
PROMPT> ping ietf.org

PING ietf.org (4.31.198.44) 56(84) bytes of data.
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=47 time=86.0 ms
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=47 time=75.6 ms
...
```

키보드에서 [Ctrl+C]를 눌러 ping 명령을 취소합니다.

ping 명령에 실패하면 다음 정보를 확인합니다.

- NAT 인스턴스의 보안 그룹 규칙이 프라이빗 서브넷에서의 인바운드 ICMP 트래픽을 허용하는지 확인합니다. 허용하지 않으면, NAT 인스턴스가 프라이빗 인스턴스에서 ping 명령을 수신할 수 없습니다.
  - 라우팅 테이블을 올바르게 구성했는지 확인합니다. 자세한 내용은 [기본 라우팅 테이블 업데이트 \(p. 186\)](#) 단원을 참조하십시오.
  - NAT 인스턴스에 대해 원본/대상 확인을 비활성화했는지 확인합니다. 자세한 내용은 [원본/대상 확인 비활성화 \(p. 185\)](#) 단원을 참조하십시오.
  - ICMP가 활성화된 웹 사이트에 대해 ping을 실행 중인지 확인합니다. 그렇지 않으면 회신 패킷이 수신되지 않을 것입니다. 이를 테스트하려면 사용자 자신의 컴퓨터의 명령줄 터미널에서 똑같은 ping 명령을 수행하십시오.
4. (선택 사항) 프라이빗 인스턴스가 더 이상 필요하지 않으면 이를 종료하십시오. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [인스턴스 종료](#)를 참조하십시오.

## NAT 인스턴스 및 NAT 게이트웨이 비교

NAT 인스턴스와 NAT 게이트웨이의 차이점을 세부적으로 요약하면 다음과 같습니다.

속성	NAT 게이트웨이	NAT 인스턴스
가용성	고가용성. 각 가용 영역의 NAT 게이트웨이는 중복적으로 구현됩니다. 각 가용 영역에 하나의 NAT 게이트웨이를 만들어 아키텍처가 영역에 종속되지 않도록 합니다.	스크립트를 사용하여 인스턴스 간의 장애 조치를 관리합니다.
대역폭	최대 10Gbps를 지원합니다.	인스턴스 유형의 대역폭에 따라 다릅니다.
유지 관리	AWS에서 관리하므로 유지 관리 작업을 수행할 필요가 없습니다.	사용자가 관리합니다(예: 인스턴스에 소프트웨어 업데이트 또는 운영 체제 패치 설치).
성능	소프트웨어가 NAT 트래픽 처리에 최적화되어 있습니다.	NAT를 수행하도록 구성된 일반 Amazon Linux AMI입니다.

속성	NAT 게이트웨이	NAT 인스턴스
비용	사용하는 NAT 게이트웨이 수, 사용 기간, NAT 게이트웨이를 통해 보내는 데이터의 양에 따라 요금이 청구됩니다.	사용하는 NAT 인스턴스 수, 사용 기간, 인스턴스 유형과 크기에 따라 요금이 청구됩니다.
유형 및 크기	균일하게 제공되므로, 유형 또는 크기를 결정할 필요가 없습니다.	예상 워크로드에 따라 적합한 인스턴스 유형과 크기를 선택합니다.
퍼블릭 IP 주소	생성할 때 NAT 게이트웨이와 연결할 탄력적 IP 주소를 선택합니다.	탄력적 IP 주소 또는 퍼블릭 IP 주소를 NAT 인스턴스와 함께 사용합니다. 새 탄력적 IP 주소를 인스턴스와 연결하여 언제든지 퍼블릭 IP 주소를 변경할 수 있습니다.
프라이빗 IP 주소	게이트웨이를 만들 때 서브넷의 IP 주소 범위에서 자동으로 선택됩니다.	인스턴스를 시작할 때 서브넷의 IP 주소 범위에서 특정 프라이빗 IP 주소를 할당합니다.
보안 그룹	NAT 게이트웨이와 연결할 수 없습니다. 보안 그룹을 NAT 게이트웨이 뒤의 리소스와 연결하여 인바운드 및 아웃바운드 트래픽을 제어할 수 있습니다.	NAT 인스턴스 뒤의 리소스 및 NAT 인스턴스와 연결하여 인바운드 및 아웃바운드 트래픽을 제어합니다.
네트워크 ACL	네트워크 ACL을 사용하여 NAT 게이트웨이가 위치하고 있는 서브넷에서 보내고 받는 트래픽을 제어합니다.	네트워크 ACL을 사용하여 NAT 인스턴스가 위치하고 있는 서브넷에서 보내고 받는 트래픽을 제어합니다.
흐름 로그	흐름 로그를 사용하여 트래픽을 캡처합니다.	흐름 로그를 사용하여 트래픽을 캡처합니다.
포트 전달	지원하지 않음.	포트 전달을 지원하려면 구성을 수동으로 사용자 지정합니다.
접속 서버	지원하지 않음.	접속 서버로 사용합니다.
트래픽 지표	지원하지 않음.	CloudWatch 지표 보기
제한 시간 초과 동작	연결 제한 시간이 초과하면 NAT 게이트웨이는 연결을 계속하려고 하는 NAT 게이트웨이 뒤의 리소스로 RST 패킷을 반환합니다(FIN 패킷을 보내지 않음).	연결 제한 시간이 초과하면 NAT 인스턴스는 NAT 인스턴스 뒤의 리소스로 FIN 패킷을 전송하여 연결을 닫습니다.
IP 조각화	UDP 프로토콜에서 IP 조각화된 패킷의 전달을 지원합니다.  TCP 및 ICMP 프로토콜에 대해서는 조각화를 지원하지 않습니다. 이러한 프로토콜의 조각화된 패킷은 삭제됩니다.	UDP, TCP 및 ICMP 프로토콜에 대해 IP 조각화된 패킷의 재수집을 지원합니다.

## DHCP 옵션 세트

이 주제에서는 DHCP 옵션 세트와 VPC에 대한 DHCP 옵션을 지정하는 방법을 설명합니다.

### 항목

- [DHCP 옵션 세트 개요 \(p. 190\)](#)
- [Amazon DNS 서버 \(p. 191\)](#)
- [DHCP 옵션 변경 \(p. 191\)](#)

- DHCP 옵션 세트를 사용한 작업 (p. 192)
- API 및 명령 개요 (p. 193)

## DHCP 옵션 세트 개요

DHCP(Dynamic Host Configuration Protocol)는 TCP/IP 네트워크 상의 호스트로 구성 정보를 전달하기 위한 표준을 제공합니다. DHCP 메시지의 `options` 필드에는 구성 파라미터가 포함됩니다. 이런 파라미터 중 일부는 도메인 이름, 도메인 이름 서버 및 `netbios-node-type`입니다.

DHCP 옵션 세트는 AWS 계정에 연결되므로, 모든 가상 프라이빗 클라우드(VPC) 전반에 이런 옵션 세트를 사용할 수 있습니다.

기본 VPC가 아닌 VPC로 시작하는 Amazon EC2 인스턴스는 기본적으로 프라이빗입니다. 시작 중에 특별히 배정하거나 서브넷의 퍼블릭 IPv4 주소 속성을 수정하지 않는 한 Amazon EC2 인스턴스에는 퍼블릭 IPv4 주소가 배정되지 않습니다. 기본적으로 기본 VPC가 아닌 VPC에 있는 모든 인스턴스는 AWS가 배정하는 확인할 수 없는 호스트 이름을 수신합니다(예: ip-10-0-0-202). 인스턴스에 자체 도메인 이름을 배정하고 최대 4개의 자체 DNS 서버를 사용할 수 있습니다. 그렇게 하려면 VPC와 함께 사용할 특별한 DHCP 옵션 세트를 지정해야 합니다.

다음 표에는 DHCP 옵션 세트에 지원되는 모든 옵션이 나와 있습니다. DHCP 옵션 세트에 필요한 옵션만 지정할 수 있습니다. 옵션에 대한 자세한 내용은 [RFC 2132](#)를 참조하십시오.

DHCP 옵션 이름	설명
domain-name-servers	<p>최대 4개의 도메인 이름 서버 또는 AmazonProvidedDNS의 IP 주소입니다. 기본 DHCP 옵션 세트는 AmazonProvidedDNS를 지정합니다. 두 개 이상의 도메인 이름 서버를 지정할 경우 쉼표로 구분하십시오.</p> <p>인스턴스가 <code>domain-name</code>에 지정된 사용자 지정 DNS 호스트 이름을 받으려면, <code>domain-name-servers</code>를 사용자 지정 DNS 서버로 설정해야 합니다.</p>
domain-name	<p><code>us-east-1</code>에서 AmazonProvidedDNS를 사용할 경우 <code>ec2.internal</code>을 지정합니다. 다른 리전에서 AmazonProvidedDNS를 사용할 경우 <code>region.compute.internal</code>을 지정합니다(예: <code>ap-northeast-1.compute.internal</code>). 그렇지 않으면, 도메인 이름을 지정합니다(예: <code>MyCompany.com</code>). 이 값은 정규화되지 않은 DNS 호스트 이름을 완성하는데 사용됩니다.</p> <p><b>Important</b></p> <p>일부 Linux 운영 체제에서는 공백으로 구분된 여러 도메인 이름을 허용합니다. 하지만 다른 Linux 운영 체제와 Windows에서는 이 값을 단일 도메인으로 취급하므로 예기치 않은 동작이 발생합니다. DHCP 옵션 세트가 여러 운영 체제를 포함한 인스턴스가 있는 VPC와 연결되는 경우 한 도메인 이름만 지정합니다.</p>
ntp-servers	<p>최대 4개의 NTP(Network Time Protocol) 서버의 IP 주소입니다. 자세한 내용은 <a href="#">RFC 2132</a>의 섹션 8.3을 참조하십시오.</p>

DHCP 옵션 이름	설명
netbios-name-servers	최대 4개의 NetBIOS 이름 서버의 IP 주소입니다.
netbios-node-type	NetBIOS 노드 유형(1, 2, 4 또는 8)입니다. 2를 지정하는 것이 좋습니다(브로드캐스트와 멀티캐스트는 현재 지원되지 않음). 이런 노드 유형에 대한 자세한 내용은 <a href="#">RFC 2132</a> 를 참조하십시오.

## Amazon DNS 서버

VPC를 생성하면 자동으로 DHCP 옵션 세트가 생성되어 VPC와 연결됩니다. 이 세트에는 `domain-name-servers=AmazonProvidedDNS` 및 `domain-name=domain-name-for-your-region`의 두 옵션이 포함됩니다. AmazonProvidedDNS는 Amazon DNS 서버이고, 이 옵션은 VPC의 인터넷 게이트웨이를 통해 통신할 필요가 있는 인스턴스의 DNS를 활성화합니다. 문자열 AmazonProvidedDNS는 VPC IPv4 네트워크 범위를 기초로 2를 더해 예약된 IP 주소에서 실행되는 DNS 서버에 매핑됩니다. 예를 들어 10.0.0.0/16 네트워크 상의 DNS 서버는 10.0.0.2에 위치합니다.

VPC에서 인스턴스를 시작하는 경우 인스턴스에 프라이빗 DNS 호스트 이름을 제공하며, 인스턴스가 퍼블릭 IPv4 주소를 받는 경우 퍼블릭 DNS 호스트 이름을 제공합니다. DHCP 옵션의 `domain-name-servers`가 AmazonProvidedDNS로 설정된 경우, 퍼블릭 DNS 호스트 이름은 `us-east-1` 리전의 경우 `ec2-public-ipv4-address.compute-1.amazonaws.com`, 다른 리전의 경우 `ec2-public-ipv4-address.region.compute.amazonaws.com` 형태가 됩니다. 프라이빗 호스트 이름은 `us-east-1` 리전의 경우 `ip-private-ipv4-address.ec2.internal`, 다른 리전의 경우 `ip-private-ipv4-address.region.compute.internal` 형태가 됩니다. 이를 사용자 지정 DNS 호스트 이름으로 변경하려면, `domain-name-servers`를 사용자 지정 DNS 서버로 설정해야 합니다.

VPC에 있는 Amazon DNS 서버는 Amazon Route 53의 프라이빗 호스팅 영역에서 지정하는 DNS 도메인 이름을 확인하는 데 사용됩니다. 프라이빗 호스팅 영역에 대한 자세한 내용은 Amazon Route 53 개발자 안내서에서 [프라이빗 호스팅 영역 작업](#) 단원을 참조하십시오.

Amazon EMR과 같은 하둡 프레임워크를 사용하는 서비스에서는 인스턴스가 자신의 FQDN(정규화된 도메인 이름)을 확인해야 합니다. 그런 경우, `domain-name-servers` 옵션이 사용자 지정 값으로 설정되어 있는 경우 DNS 확인이 실패할 수 있습니다. DNS를 올바르게 확인하려면 `region-name.compute.internal` 도메인에 대한 쿼리를 Amazon DNS 서버로 전달하기 위해 DNS 서버 상에 조건부 전달자를 추가하는 방법을 고려하십시오. Amazon EMR 클러스터를 VPC로 시작하는 데 대한 자세한 내용은 Amazon EMR 개발자 안내서에서 [클러스터를 호스팅하기 위한 VPC 설정](#) 단원을 참조하십시오.

### Note

비록 일부 서버에서는 사용할 수 없지만, Amazon DNS 서버 IP 주소 169.254.169.253을 사용할 수 있습니다. 예를 들어 Windows Server 2008에서는 169.254.x.x 네트워크 범위에 있는 DNS 서버의 사용이 허용되지 않습니다.

## DHCP 옵션 변경

DHCP 옵션 세트를 생성한 후에는 이 옵션 세트를 수정할 수 없습니다. VPC에서 다른 DHCP 옵션 세트를 사용하도록 하려면 새 세트를 생성하여 VPC와 연결해야 합니다. DHCP 옵션을 전혀 사용하지 않도록 VPC를 설정할 수도 있습니다.

여러 DHCP 옵션 세트를 둘 수 있지만, 한 번에 한 VPC와 한 DHCP 옵션 세트만 연결할 수 있습니다. VPC를 삭제하면 VPC와 연결된 DHCP 옵션 세트 역시 삭제됩니다.

새 DHCP 옵션 세트를 VPC와 연결한 후, VPC에서 시작하는 기존 인스턴스와 모든 새 인스턴스가 이런 옵션을 사용합니다. 인스턴스를 다시 시작하거나 다시 실행할 필요가 없습니다. 인스턴스가 DHCP 임대를 갱신

하는 빈도에 따라 몇 시간 안에 변경 내용이 자동으로 파악됩니다. 원한다면 인스턴스에서 운영 체제를 사용하여 임대를 명시적으로 갱신할 수 있습니다.

## DHCP 옵션 세트를 사용한 작업

이 섹션에서는 DHCP 옵션 세트로 작업하는 방법을 설명합니다.

### 항목

- DHCP 옵션 세트 생성 (p. 192)
- VPC가 사용하는 DHCP 옵션 세트 변경 (p. 192)
- VPC가 아무런 DHCP 옵션도 사용하지 못하도록 변경 (p. 193)
- DHCP 옵션 세트 삭제 (p. 193)

## DHCP 옵션 세트 생성

원하는 만큼 DHCP 옵션 세트를 추가로 생성할 수 있습니다. 하지만 한 번에 한 DHCP 옵션 세트와 한 VPC만 연결할 수 있습니다. DHCP 옵션 세트를 생성한 후 이를 사용할 수 있도록 VPC를 구성해야 합니다. 자세한 내용은 [VPC가 사용하는 DHCP 옵션 세트 변경 \(p. 192\)](#) 단원을 참조하십시오.

### DHCP 옵션 세트를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [DHCP Options Sets]를 선택한 후 [Create DHCP options set]를 선택합니다.
3. 대화 상자에서 사용하려는 옵션에 대한 값을 입력한 후 [Yes, Create]를 선택합니다.

#### Important

VPC에 인터넷 게이트웨이가 있는 경우 [Domain name servers] 값으로 자체 DNS 서버 또는 Amazon의 DNS 서버(AmazonProvidedDNS)를 지정해야 합니다. 그렇지 않으면, 인터넷과 통신할 필요가 있는 인스턴스가 DNS에 액세스하지 못합니다.

새 DHCP 옵션 세트가 DHCP 옵션 목록에 나타납니다.

4. 새로운 DHCP 옵션 세트의 ID를 기록해 두십시오(dopt-xxxxxxx). 새 옵션 세트를 VPC와 연결할 때 이 ID가 필요하기 때문입니다.

DHCP 옵션 세트를 생성했다라도, 옵션을 적용하려면 옵션 세트를 VPC와 연결해야 합니다. 여러 DHCP 옵션 세트를 생성할 수 있지만, 한 번에 VPC와 한 DHCP 옵션 세트만 연결할 수 있습니다.

## VPC가 사용하는 DHCP 옵션 세트 변경

VPC가 어떤 DHCP 옵션 세트를 사용할지 변경할 수 있습니다. VPC가 DHCP 옵션을 전혀 사용하지 않도록 하는 방법은 [VPC가 아무런 DHCP 옵션도 사용하지 못하도록 변경 \(p. 193\)](#) 단원을 참조하십시오.

### Note

다음 절차에서는 변경하려는 대상 DHCP 옵션 세트를 이미 생성했다고 가정합니다. 생성하지 않았다면, 지금 옵션 세트를 생성하십시오. 자세한 내용은 [DHCP 옵션 세트 생성 \(p. 192\)](#) 단원을 참조하십시오.

### VPC와 연결된 DHCP 옵션 세트를 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.

3. VPC를 선택하고 [Actions] 목록에서 [Edit DHCP Options Set]를 선택합니다.
4. [DHCP Options Set] 목록에서 옵션 세트를 선택한 후 [Save]를 선택합니다.

새 DHCP 옵션 세트를 VPC와 연결한 후, 그 VPC에서 시작하는 기존 인스턴스와 모든 새 인스턴스가 이런 옵션을 사용합니다. 인스턴스를 다시 시작하거나 다시 실행할 필요가 없습니다. 인스턴스가 DHCP 임대를 갱신하는 빈도에 따라 몇 시간 안에 변경 내용이 자동으로 파악됩니다. 원한다면 인스턴스에서 운영 체제를 사용하여 임대를 명시적으로 갱신할 수 있습니다.

## VPC가 아무런 DHCP 옵션도 사용하지 못하도록 변경

DHCP 옵션 세트를 전혀 사용하지 않도록 VPC를 설정할 수 있습니다.

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC를 선택하고 [Actions] 목록에서 [Edit DHCP Options Set]를 선택합니다.
4. [DHCP Options Set] 목록에서 [No DHCP Options Set]를 선택한 후 [Save]를 선택합니다.

인스턴스를 다시 시작하거나 다시 실행할 필요가 없습니다. 인스턴스가 DHCP 임대를 갱신하는 빈도에 따라 몇 시간 안에 변경 내용이 자동으로 파악됩니다. 원한다면 인스턴스에서 운영 체제를 사용하여 임대를 명시적으로 갱신할 수 있습니다.

## DHCP 옵션 세트 삭제

더 이상 DHCP 옵션 세트가 필요하지 않으면 다음 절차에 따라 DHCP 옵션 세트를 삭제합니다. VPC가 옵션 세트를 사용 중이어서는 안 됩니다.

DHCP 옵션 세트를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [DHCP Options Sets]를 선택합니다.
3. 삭제할 DHCP 옵션 세트를 선택한 후 [Delete]를 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

## API 및 명령 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

VPC에 대한 DHCP 옵션 세트 생성

- [create-dhcp-options](#)(AWS CLI)
- [New-EC2DhcpOption](#)(Windows PowerShell용 AWS 도구)

DHCP 옵션 세트를 지정된 VPC와 연결하거나 아무런 DHCP 옵션도 없음

- [associate-dhcp-options](#)(AWS CLI)
- [Register-EC2DhcpOption](#)(Windows PowerShell용 AWS 도구)

하나 이상의 DHCP 옵션 세트 설명

- [describe-dhcp-options](#)(AWS CLI)



- [Get-EC2DhcpOption](#)(Windows PowerShell용 AWS 도구)

#### DHCP 옵션 세트 삭제

- [delete-dhcp-options](#)(AWS CLI)
- [Remove-EC2DhcpOption](#)(Windows PowerShell용 AWS 도구)

## VPC와 함께 DNS 사용

Amazon EC2 인스턴스는 통신을 위한 IP 주소가 필요합니다. 퍼블릭 IPv4 주소를 사용하면 인터넷으로 통신할 수 있는 반면, 프라이빗 IPv4 주소를 사용하면 인스턴스의 네트워크(EC2-Classic 또는 VPC) 내에서 통신할 수 있습니다. 자세한 내용은 [VPC의 IP 주소 지정 \(p. 76\)](#) 섹션을 참조하십시오.

도메인 이름 시스템(DNS)은 인터넷에서 사용되는 이름을 해당 IP 주소로 확인할 때 기준이 됩니다. DNS 호스트 이름은 컴퓨터 이름을 고유하고 절대적으로 지정하는 이름으로서, 호스트 이름과 도메인 이름으로 구성됩니다. DNS 서버는 DNS 호스트 이름을 해당 IP 주소로 확인합니다.

당사에서는 Amazon DNS 서버를 제공합니다. 자체 DNS 서버를 사용하려면 자신의 VPC에 대한 DHCP 옵션 세트를 업데이트하십시오. 자세한 내용은 [DHCP 옵션 세트 \(p. 189\)](#) 단원을 참조하십시오.

#### 항목

- [DNS 호스트 이름 \(p. 194\)](#)
- [VPC에서 DNS 지원 \(p. 194\)](#)
- [EC2 인스턴스의 DNS 호스트 이름 보기 \(p. 195\)](#)
- [VPC에 대한 DNS 지원 업데이트 \(p. 196\)](#)
- [프라이빗 호스팅 영역 사용 \(p. 197\)](#)

## DNS 호스트 이름

기본 VPC에서 인스턴스를 시작하는 경우, 이 인스턴스에 퍼블릭 및 프라이빗 DNS 호스트 이름을 제공하는 데, 이 호스트 이름은 이 인스턴스의 퍼블릭 IPv4 및 프라이빗 IPv4 주소에 상응합니다. 기본이 아닌 VPC로 인스턴스를 시작할 때 인스턴스에 프라이빗 DNS 호스트 이름을 제공하며, VPC 및 인스턴스에 지정한 설정에 따라 퍼블릭 DNS 호스트 이름을 제공할 수 있습니다.

프라이빗(내부) DNS 호스트 이름은 인스턴스의 프라이빗 IPv4 주소로 확인되며, 형태는 us-east-1 리전의 경우 `ip-private-ipv4-address.ec2.internal`, 다른 리전의 경우 `ip-private-ipv4-address.region.compute.internal`(`private.ipv4.address`가 역방향 조회 IP 주소)이 됩니다. 프라이빗 DNS 호스트 이름은 동일 네트워크에서 인스턴스 간의 통신을 위해 사용될 수 있지만 인스턴스가 위치한 네트워크 외부의 DNS 호스트 이름은 확인할 수 없습니다.

퍼블릭(외부) DNS 호스트 이름은 us-east-1 리전의 경우 `ec2-public-ipv4-address.compute-1.amazonaws.com`, 다른 리전의 경우 `ec2-public-ipv4-address.region.amazonaws.com` 형태가 됩니다. Amazon은 퍼블릭 DNS 호스트 이름을 인스턴스 네트워크 외부에서는 인스턴스의 퍼블릭 IPv4 주소로 변환하고, 인스턴스 네트워크 내부에서는 인스턴스의 프라이빗 IPv4 주소로 변환합니다.

IPv6 주소에 대한 DNS 호스트 이름은 제공하지 않습니다.

## VPC에서 DNS 지원

VPC에는 인스턴스가 퍼블릭 DNS 호스트 이름을 수신하는지, Amazon DNS 서버를 통한 DNS 확인이 지원되는지 결정하는 속성이 있습니다. 인스턴스가 인터넷에서 액세스 가능한 퍼블릭 DNS 호스트 이름을 갖도록 하려면 두 속성을 모두 `true`로 설정해야 합니다.



속성	설명
<code>enableDnsHostnames</code>	VPC에서 시작된 인스턴스가 퍼블릭 DNS 호스트 이름을 가져오는지 나타냅니다. 이 속성이 <code>true</code> 이면 VPC의 인스턴스가 DNS 호스트 이름을 가져오고, 그렇지 않으면 가져오지 않습니다. 인스턴스가 DNS 호스트 이름을 가져오도록 하려면 <code>enableDnsSupport</code> 속성도 <code>true</code> 로 설정해야 합니다.
<code>enableDnsSupport</code>	VPC에 대해 DNS 확인이 지원되는지 나타냅니다. 이 속성이 <code>false</code> 이면 퍼블릭 DNS 호스트 이름을 IP 주소로 확인하는 VPC의 Amazon 제공 DNS 서비스가 활성화되지 않습니다. 이 속성이 <code>true</code> 이면, IP 주소 169.254.169.253 또는 VPC IPv4 네트워크 범위를 기초로 2를 더해 예약된 IP 주소의 Amazon 제공 DNS 서버에 대한 쿼리에 성공할 것입니다. 자세한 내용은 <a href="#">Amazon DNS 서버 (p. 191)</a> 단원을 참조하십시오.

기본적으로, DNS 호스트 이름은 기본 VPC와 VPC 콘솔에서 VPC 마법사를 사용하여 생성하는 VPC에만 활성화됩니다.

VPC의 IPv4 주소 범위가 [RFC 1918](#)에 의해 지정된 프라이빗 IPv4 주소 범위를 벗어나는 경우를 비롯한 모든 주소 공간에 대해, Amazon DNS 서버는 프라이빗 DNS 호스트 이름을 프라이빗 IPv4 주소로 확인할 수 있습니다.

#### Important

2016년 10월 이전에 VPC를 생성했다면, VPC의 IPv4 주소 범위가 RFC 1918에 의해 지정된 프라이빗 IPv4 주소 범위를 벗어나는 경우 Amazon DNS 서버가 프라이빗 DNS 호스트 이름을 확인하지 않습니다. Amazon DNS 서버가 이러한 주소에 대해 프라이빗 DNS 호스트 이름을 확인할 수 있도록 하려면 [AWS Support](#)에게 문의하십시오.

이전에는 DNS 호스트 이름과 DNS를 지원하지 않았던 VPC에서 이런 지원을 사용하는 경우, 그 VPC로 이미 시작한 인스턴스는 퍼블릭 IPv4 주소 또는 탄력적 IP 주소가 있는 경우 퍼블릭 DNS 호스트 이름을 가져옵니다.

프라이빗 호스팅 영역에 대한 DNS 지원에 관한 자세한 내용은 [프라이빗 호스팅 영역 사용 \(p. 197\)](#)을 참조하십시오.

## EC2 인스턴스의 DNS 호스트 이름 보기

Amazon EC2 콘솔 또는 명령줄을 사용하여 실행 중인 인스턴스 또는 네트워크 인터페이스의 DNS 호스트 이름을 볼 수 있습니다.

### 인스턴스

콘솔을 사용하여 인스턴스의 DNS 호스트 이름을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 **Instances**를 선택합니다.
3. 목록에서 해당 인스턴스를 선택합니다.
4. 해당되는 경우, 세부 정보 창의 **Public DNS (IPv4)** 및 **Private DNS** 필드에 DNS 호스트 이름이 표시됩니다.

명령줄을 사용하여 인스턴스의 DNS 호스트 이름을 보려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 섹션을 참조하십시오.

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (Windows PowerShell용 AWS 도구)

## 네트워크 인터페이스

콘솔을 사용하여 네트워크 인터페이스의 프라이빗 DNS 호스트 이름을 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Network Interfaces를 선택합니다.
3. 목록에서 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창의 Private DNS(IPv4) 필드에 프라이빗 DNS 호스트 이름이 표시됩니다.

명령줄을 사용하여 네트워크 인터페이스의 DNS 호스트 이름을 보려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 섹션을 참조하십시오.

- [describe-network-interfaces](#)(AWS CLI)
- [Get-EC2NetworkInterface](#)(Windows PowerShell용 AWS 도구)

## VPC에 대한 DNS 지원 업데이트

Amazon VPC 콘솔을 이용해 VPC의 DNS 지원 속성을 확인하고 업데이트할 수 있습니다.

콘솔을 사용하여 VPC에 대한 DNS 지원을 설명하고 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. 목록에서 VPC를 선택합니다.
4. [Summary] 탭의 정보를 검토합니다. 이 예제에서는 두 가지 모두 사용합니다.
5. 설정을 업데이트하려면 Actions를 선택하고, Edit DNS Resolution 또는 Edit DNS Hostnames를 선택합니다. 이때 열리는 대화 상자에서 Yes 또는 No를 선택한 다음, Save를 선택합니다.

명령줄을 사용하여 VPC에 대한 DNS 지원을 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 정보는 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

- [describe-vpc-attribute](#)(AWS CLI)
- [Get-EC2VpcAttribute](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 VPC에 대한 DNS 지원을 업데이트하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음의 명령줄 인터페이스에 대한 자세한 정보는 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

- `modify-vpc-attribute`(AWS CLI)
- `Edit-EC2VpcAttribute`(Windows PowerShell용 AWS 도구)

## 프라이빗 호스팅 영역 사용

프라이빗 IPv4 주소 또는 AWS 제공 프라이빗 DNS 호스트 이름을 사용하는 대신에 `example.com`과 같은 사용자 지정 DNS 도메인 이름을 사용하여 VPC의 리소스에 액세스하려면 Amazon Route 53에서 프라이빗 호스팅 영역을 생성합니다. 프라이빗 호스팅 영역이란 인터넷에 자신의 리소스를 노출하지 않고 하나 이상의 VPC 내에 있는 도메인과 그 하위 도메인의 트래픽을 라우팅하려는 방식에 대한 정보를 담고 있는 컨테이너입니다. Amazon Route 53이 도메인과 하위 도메인에 대한 쿼리에 응답하는 방식을 결정하는 Amazon Route 53 리소스 레코드 세트를 생성할 수 있습니다. 예를 들어 `example.com`에 대한 브라우저 요청이 VPC의 웹 서버로 라우팅되도록 하려는 경우, 프라이빗 호스팅 영역에 A 레코드를 생성하고 그 웹 서버의 IP 주소를 지정할 것입니다. 프라이빗 호스팅 영역의 생성에 대한 자세한 내용은 Amazon Route 53 개발자 안내서에서 [프라이빗 호스팅 영역 작업](#) 단원을 참조하십시오.

사용자 지정 DNS 도메인 이름을 사용하여 리소스에 액세스하려면 VPC 내에 있는 인스턴스에 연결되어 있어야 합니다. 인스턴스에서 `ping` 명령(예: `ping mywebserver.example.com`)을 사용하여 사용자 지정 DNS 이름에서 프라이빗 호스팅 영역에 있는 리소스에 액세스 가능한지 테스트할 수 있습니다. (인스턴스의 보안 그룹 규칙에서 `ping`에 대한 인바운드 ICMP 트래픽 작동을 허용하는지 확인해야 합니다.)

ClassicLink DNS 지원에 대해 VPC가 활성화되어 있는 경우 ClassicLink를 통해 VPC에 연결된 EC2-Classic 인스턴스의 프라이빗 호스팅 영역에 액세스할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Enabling ClassicLink DNS Support](#) 단원을 참조하십시오. 그렇지 않은 경우 프라이빗 호스팅 영역은 VPC 외부에서 전이적 관계를 지원하지 않습니다. 예를 들어 VPN 연결의 반대쪽에서 사용자 지정 프라이빗 DNS 이름을 사용하여 리소스에 액세스할 수 없습니다.

### Important

Amazon Route 53의 프라이빗 호스팅 영역에 정의된 사용자 지정 DNS 도메인 이름을 사용하는 경우, `enableDnsHostnames` 및 `enableDnsSupport` VPC 속성을 `true`로 설정해야 합니다. 자세한 내용은 [VPC에 대한 DNS 지원 업데이트 \(p. 196\)](#) 단원을 참조하십시오.

## VPC 피어링

VPC 피어링 연결은 프라이빗 IPv4 주소 또는 IPv6 주소를 사용하여 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다. 자체 VPC 간의 VPC 피어링 연결, 또는 단일 리전 내에 있는 다른 AWS 계정에서 VPC와의 VPC 피어링 연결을 만들 수 있습니다.

AWS는 VPC의 기존 인프라를 사용하여 VPC 피어링 연결을 생성합니다. 이는 게이트웨어도, VPN 연결도 아니며 물리적 하드웨어 각각에 의존하지 않습니다. 그러므로 통신 또는 대역폭 병목에 대한 단일 지점 장애가 없습니다.

VPC 피어링 연결을 사용하는 방법에 대한 자세한 내용과 VPC 피어링 연결을 사용할 수 있는 여러 시나리오의 예는 [Amazon VPC Peering Guide](#)를 참조하십시오.

## 탄력적 IP 주소

탄력적 IP 주소는 동적 클라우드 컴퓨팅을 위해 고안된 고정 퍼블릭 IPv4 주소입니다. 계정의 모든 VPC에 대한 모든 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결할 수 있습니다. 탄력적 IP 주소로 VPC의 다른 인스턴스에 주소를 신속하게 다시 매핑하여 인스턴스의 오류를 숨길 수 있습니다. 탄력적 IP 주소를

인스턴스에 직접 연결하는 대신에 네트워크 인터페이스와 연결할 경우, 네트워크 인터페이스의 모든 속성들을 한 인스턴스에서 다른 인스턴스로 한 번에 옮길 수 있는 장점이 있습니다.

현재는 IPv6에 대한 탄력적 IP 주소를 지원하지 않습니다.

#### 항목

- 탄력적 IP 주소 기본 사항 (p. 198)
- 탄력적 IP 주소 작업 (p. 198)
- API 및 CLI 개요 (p. 200)

## 탄력적 IP 주소 기본 사항

다음은 탄력적 IP 주소에 대해 알아야 할 기본 사항입니다.

- VPC에서 사용할 탄력적 IP 주소를 먼저 할당한 후 이를 VPC의 인스턴스와 연결합니다(한 번에 하나의 인스턴스에만 할당 가능).
- 탄력적 IP 주소는 네트워크 인터페이스의 속성입니다. 인스턴스에 연결된 네트워크 인터페이스를 업데이트하여 탄력적 IP 주소를 인스턴스와 연결할 수 있습니다.
- 탄력적 IP 주소를 인스턴스의 eth0 네트워크 인터페이스와 연결하면, 현재 퍼블릭 IPv4 주소(있는 경우)는 EC2-VPC 퍼블릭 IP 주소 풀로 연결 해제됩니다. 탄력적 IP 주소의 연결을 해제하면 몇 분 내에 자동으로 eth0 네트워크 인터페이스에 새 퍼블릭 IPv4 주소가 배정됩니다. 인스턴스에 보조 네트워크 인터페이스를 연결한 경우에는 이 동작이 적용되지 않습니다.
- VPC와 EC2-Classic에서 사용하는 탄력적 IP 주소 간에는 차이점이 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [EC2-Classic 및 Amazon EC2-VPC의 탄력적 IP 주소 차이점](#)을 참조하십시오.
- 인스턴스 간에 탄력적 IP 주소를 이동할 수 있습니다. 같은 VPC나 다른 VPC로 인스턴스를 이동할 수 있을 수 있지만 EC2-Classic으로 이동할 수는 없습니다.
- 탄력적 IP 주소는 명시적으로 연결 해제할 때까지 AWS 계정과 연결되어 있습니다.
- 탄력적 IP 주소의 효율적인 사용을 보장하기 위해, 탄력적 IP 주소가 실행 중인 인스턴스와 연결되어 있지 않거나, 중지된 인스턴스 또는 분리된 네트워크 인터페이스와 연결되어 있는 경우 소액의 시간당 요금이 부과됩니다. 인스턴스가 실행 중인 동안에는 이와 연결된 탄력적 IP 주소 하나에 대해서는 요금이 부과되지 않지만 해당 인스턴스와 연결된 추가 탄력적 IP 주소에 대해서는 요금이 부과됩니다. 자세한 내용은 [Amazon EC2 요금](#)을 참조하십시오.
- 탄력적 IP 주소는 5개로 제한되며, 이를 절약하기 위해 NAT 디바이스를 사용할 수 있습니다 ([NAT \(p. 172\)](#) 참조).
- 탄력적 IP 주소는 VPC의 인터넷 게이트웨이를 통해 액세스합니다. VPC와 네트워크 간에 VPN 연결을 설정한 경우, VPN 트래픽은 인터넷 게이트웨이가 아닌 가상 프라이빗 게이트웨이를 통과하기 때문에 탄력적 IP 주소에 액세스할 수 없습니다.
- EC2-Classic 플랫폼에서 사용하기 위해 할당한 탄력적 IP 주소를 VPC 플랫폼으로 이동할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [EC2-Classic에서 EC2-VPC로 탄력적 IP 주소 마이그레이션](#)을 참조하십시오.

## 탄력적 IP 주소 작업

탄력적 IP 주소를 할당한 후 VPC의 인스턴스와 연결할 수 있습니다.

VPC에서 사용할 탄력적 IP 주소를 할당하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.

3. Allocate new address를 선택합니다.
4. [Allocate]를 선택합니다.

**Note**

계정이 EC2-Classic을 지원할 경우 우선 [VPC]를 선택합니다.

탄력적 IP 주소를 보려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 표시되는 목록을 필터링하려면 검색 상자에 배정된 인스턴스의 ID나 탄력적 IP 주소를 몇 자 입력합니다.

VPC에서 실행 중인 인스턴스와 탄력적 IP 주소를 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. VPC에서 사용하기 위해 할당한 탄력적 IP 주소를 선택한 후([Scope] 열에 vpc 값이 표시됨), [Actions]를 선택하고 [Associate Address]를 선택합니다.
4. [Instance] 또는 [Network interface]를 선택한 다음 인스턴스 또는 네트워크 인터페이스 ID를 선택합니다. 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 선택합니다. [Associate]를 선택합니다.

**Note**

네트워크 인터페이스는 탄력적 IP 주소를 포함해 몇몇 속성들을 포함할 수 있습니다. 네트워크 인터페이스를 만들어 내 VPC의 인스턴스에 연결했다가 분리할 수도 있습니다. 탄력적 IP 주소를 인스턴스에 직접 연결하는 대신에 이 주소를 네트워크 인터페이스의 한 속성으로 만들 경우, 네트워크 인터페이스의 모든 속성들을 한 인스턴스에서 다른 인스턴스로 한 번에 옮길 수 있는 장점이 있습니다. 자세한 내용은 [Elastic Network Interfaces\(탄력적 네트워크 인터페이스 \[ENI\]\)](#)를 참조하십시오.

탄력적 IP 주소를 인스턴스와 연결하면 DNS 호스트 이름이 활성화된 경우 DNS 호스트 이름을 받습니다. 자세한 내용은 [VPC와 함께 DNS 사용 \(p. 194\)](#) 단원을 참조하십시오.

탄력적 IP 주소와 연결된 인스턴스를 변경하려면 현재 연결된 인스턴스에서 연결을 해제한 후 VPC의 새 인스턴스와 연결하십시오.

탄력적 IP 주소를 연결 해제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 [Actions]를 선택한 후 [Disassociate Address]를 선택합니다.
4. 메시지가 나타나면 [Disassociate address]를 선택합니다.

더 이상 필요 없는 탄력적 IP 주소는 연결을 해제하는 것이 좋습니다(이 주소는 인스턴스와 연결할 수 없음). VPC에서 사용하기 위해 할당한 탄력적 IP 주소에 대한 요금은 발생하지만 인스턴스와 연결된 주소에 대해서는 요금이 발생하지 않습니다.

탄력적 IP 주소를 해제합니다

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.

3. 탄력적 IP 주소를 선택하고 [Actions]를 선택한 후 [Release Addresses]를 선택합니다.
4. 메시지가 나타나면 [Release]를 선택합니다.

## API 및 CLI 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 4\)](#) 단원을 참조하십시오.

### 탄력적 IP 주소 얻기

- [allocate-address](#) (AWS CLI)
- [New-EC2Address](#)(Windows PowerShell용 AWS 도구)

### 인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소 연결

- [associate-address](#)(AWS CLI)
- [Register-EC2Address](#)(Windows PowerShell용 AWS 도구)

### 하나 이상의 탄력적 IP 주소 설명

- [describe-addresses](#)(AWS CLI)
- [Get-EC2Address](#)(Windows PowerShell용 AWS 도구)

### 탄력적 IP 주소 연결 해제

- [disassociate-address](#)(AWS CLI)
- [Unregister-EC2Address](#)(Windows PowerShell용 AWS 도구)

### 탄력적 IP 주소 릴리스

- [release-address](#)(AWS CLI)
- [Remove-EC2Address](#)(Windows PowerShell용 AWS 도구)

## VPC 엔드포인트

VPC endpoint를 사용하면 NAT 디바이스나 VPN 연결 또는 AWS Direct Connect를 통해 인터넷에 액세스하지 않고도 VPC와 다른 AWS 서비스 간에 프라이빗 연결을 생성할 수 있습니다. 엔드포인트는 가상 디바이스입니다. 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로, 가용성 위험이나 네트워크 트래픽에 대한 대역폭 제약 없이 VPC의 인스턴스와 AWS 서비스 간에 통신할 수 있게 해줍니다.

### Important

현재로서는 엔드포인트와 S3의 연결만 지원됩니다. 엔드포인트는 IPv4 트래픽에 대해서만 지원됩니다.

엔드포인트는 VPC의 인스턴스가 자체 프라이빗 IP 주소를 사용하여 다른 서비스의 리소스와 통신할 수 있게 합니다. 인스턴스에 퍼블릭 IPv4 주소를 지정할 필요가 없고, VPC에서 인터넷 게이트웨이, NAT 디바이스 또는 가상 프라이빗 게이트웨이가 필요 없습니다. 엔드포인트 정책을 사용하여 다른 서비스의 리소스에 대한 액세스를 제어할 수 있습니다. VPC와 AWS 서비스 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

다음 다이어그램에서 서브넷 2의 인스턴스는 VPC 엔드포인트를 통해 Amazon S3에 액세스할 수 있습니다.

엔드포인트 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 리소스 사용량에 대한 표준 요금이 그대로 적용됩니다. 요금에 대한 자세한 내용은 [Amazon EC2 Pricing](#)을 참조하십시오.

#### 항목

- [엔드포인트 기본 사항 \(p. 201\)](#)
- [엔드포인트 사용 제어 \(p. 203\)](#)
- [서비스에 대한 액세스 제어 \(p. 204\)](#)
- [Amazon S3에 대한 엔드포인트 \(p. 204\)](#)
- [엔드포인트 작업 \(p. 208\)](#)
- [API 및 CLI 개요 \(p. 211\)](#)

## 엔드포인트 기본 사항

엔드포인트를 만들려면 연결하려는 VPC 및 서비스를 지정하십시오. 서비스는 접두사 목록 또는 리전의 서비스 이름 및 ID로 식별됩니다. 접두사 목록 ID는 `p1-xxxxxxx` 형식을 사용하고, 접두사 목록 이름은 `com.amazonaws.<region>.<service>` 형식을 사용합니다. 접두사 목록 이름(서비스 이름)을 사용하여 엔드포인트를 만듭니다.

엔드포인트에서 연결하려는 모든 서비스 또는 일부 서비스에 액세스를 허용하는 엔드포인트 정책을 엔드포인트에 추가합니다. 자세한 내용은 [엔드포인트 정책 사용 \(p. 204\)](#) 단원을 참조하십시오. VPC와 다른 서비스 간의 트래픽 라우팅을 제어하려면 VPC에서 엔드포인트에 액세스할 때 사용할 라우팅 테이블을 하나 이상 지정합니다. 이러한 라우팅 테이블을 사용하는 서브넷은 엔드포인트에 액세스할 수 있으며, 해당 서브넷의 인스턴스에서 서비스로 전송되는 트래픽은 엔드포인트를 통해 라우팅됩니다.

엔드포인트를 만든 후 엔드포인트에 추가한 정책을 수정할 수 있고, 엔드포인트에서 사용하는 라우팅 테이블을 추가하거나 제거할 수 있습니다.

하나의 VPC에 여러 엔드포인트를 만들 수 있습니다. 예를 들어 여러 서비스에 대한 엔드포인트를 만들 수 있습니다. 또한 하나의 서비스에 대해 여러 엔드포인트를 만들 수 있고, 서로 다른 라우팅 테이블을 사용하여 동일한 서비스에 대해 서브넷별로 서로 다른 정책을 적용할 수 있습니다.

#### 항목

- [엔드포인트의 라우팅 \(p. 201\)](#)
- [엔드포인트 제한 사항 \(p. 203\)](#)

## 엔드포인트의 라우팅

엔드포인트를 만들거나 수정할 경우 엔드포인트를 통해 서비스에 액세스할 때 사용되는 VPC 라우팅 테이블을 지정해야 합니다. 경로는 각각의 라우팅 테이블에 자동으로 추가되며 이때 서비스의 접두사 목록 ID(`p1-xxxxxxx`)를 지정하는 목적지 및 엔드포인트 ID(`vpce-xxxxxxx`)를 포함한 대상도 함께 추가됩니다. 접두사 목록 ID는 논리적으로 서비스가 사용하는 퍼블릭 IP 주소의 범위를 나타냅니다. 지정된 라우팅 테이블과 연결된 서브넷의 모든 인스턴스는 자동으로 해당 엔드포인트를 사용하여 서비스에 액세스합니다. 지정된 라우팅 테이블과 연결되지 않은 서브넷은 서비스에 액세스하기 위해 해당 엔드포인트를 사용하지 않습니다. 따라서 다른 서브넷의 리소스를 엔드포인트와 분리할 수 있습니다.

Amazon은 LPM(Longest Prefix Match)을 통해 트래픽과 일치하는 고도로 구체적인 라우팅을 사용하여 트래픽의 라우팅 방법을 결정합니다. 인터넷 게이트웨이를 가리키는 모든 인터넷 트래픽(`0.0.0.0/0`)에 대해 라우팅 테이블에 기존 라우팅이 있을 경우, 해당 서비스로 향하는 모든 트래픽에 대해 엔드포인트 라우팅이 우선합니다. 이는 서비스의 IP 주소 범위가 `0.0.0.0/0`보다 구체적이기 때문입니다. 다른 리전에 있는 서비스로 향하는 트래픽을 비롯하여 다른 모든 인터넷 트래픽은 인터넷 게이트웨이로 전송됩니다.



하지만 인터넷 게이트웨이 또는 NAT 디바이스를 가리키는 구체적인 IP 주소 범위의 라우팅이 있을 경우 그러한 라우팅이 우선합니다. 서비스가 사용하는 IP 주소와 동일한 IP 주소 범위를 향하는 기존 라우팅이 있을 경우에는 이 라우팅이 우선합니다.

서비스의 현재 IP 주소 범위를 보려면 [describe-prefix-lists](#) 명령의 [AWS IP 주소 범위](#)를 참조하십시오.

**Note**

서비스의 퍼블릭 IP 주소의 범위는 때때로 변경될 수 있습니다. 서비스의 현재 IP 주소 범위를 기반으로 라우팅을 작성하거나 기타 결정을 내리기 전에 먼저 그 영향을 고려하십시오.

하나의 라우팅 테이블에 서로 다른 서비스에 대한 여러 엔드포인트 라우팅을 작성할 수 있으며, 동일한 서비스에 대한 여러 엔드포인트 라우팅을 서로 다른 라우팅 테이블에 작성할 수 있습니다. 하지만 하나의 라우팅 테이블에 동일한 서비스에 대해 여러 엔드포인트를 작성할 수는 없습니다. 예를 들어 VPC에 Amazon S3에 대한 두 개의 엔드포인트가 있을 경우 두 엔드포인트에 같은 라우팅 테이블을 사용할 수 없습니다.

라우팅 테이블 API 또는 VPC 콘솔의 라우팅 테이블 페이지를 사용하여 라우팅 테이블에서 엔드포인트 라우팅을 명시적으로 추가, 수정 또는 삭제할 수 없습니다. 라우팅 테이블을 엔드포인트와 연결하는 방법으로만 엔드포인트 라우팅을 추가할 수 있습니다. 엔드포인트를 수정하여 라우팅 테이블 연결을 삭제하거나 엔드포인트를 삭제할 경우 엔드포인트 라우팅이 자동으로 삭제됩니다.

엔드포인트와 연결된 라우팅 테이블을 수정하려면 엔드포인트를 수정하면 됩니다. 자세한 내용은 [엔드포인트 수정 \(p. 209\)](#) 단원을 참조하십시오.

예: 라우팅 테이블의 엔드포인트 라우팅

이 시나리오에서는 라우팅 테이블에 모든 인터넷 트래픽(0.0.0.0/0)에 대해 인터넷 게이트웨이를 가리키는 기존의 라우팅이 있습니다. 서브넷에서 또 다른 AWS 서비스를 향하는 모든 트래픽은 인터넷 게이트웨이를 사용합니다.

목적지	Target
10.0.0.0/16	로컬
0.0.0.0/0	igw-1a2b3c4d

지원되는 AWS에 대한 엔드포인트를 만든 후 라우팅 테이블을 엔드포인트와 연결할 수 있습니다. 엔드포인트 라우팅은 목적지 p1-1a2b3c4d(엔드포인트를 만든 서비스를 나타내는 것으로 가정)와 함께 라우팅 테이블에 자동으로 추가됩니다. 이제 같은 리전에서 해당 AWS 서비스를 향하는 서브넷의 모든 트래픽은 인터넷 게이트웨이가 아닌 엔드포인트로 전송됩니다. 다른 서비스로 향하는 트래픽 및 다른 리전에서 AWS 서비스로 향하는 트래픽을 포함하여 다른 모든 인터넷 트래픽은 인터넷 게이트웨이로 전송됩니다.

목적지	Target
10.0.0.0/16	로컬
0.0.0.0/0	igw-1a2b3c4d
p1-1a2b3c4d	vpce-11bb22cc

예: 엔드포인트에 대한 라우팅 테이블 조정

이 시나리오에서는 라우팅 테이블을 구성하여 인터넷 게이트웨이를 통해 서브넷의 인스턴스가 Amazon S3 버킷과 통신할 수 있게 했습니다. 목적지가 54.123.165.0/24이고(현재 Amazon S3 내의 IP 주소 범위로 가정) 대상이 인터넷 게이트웨이인 라우팅을 추가했습니다. 이제 엔드포인트를 만든 후 이 라우팅 테이블을 엔드포인트와 연결합니다. 엔드포인트 라우팅은 자동으로 라우팅 테이블에 추가됩니다. 그런 다음 [describe-](#)



`prefix-lists` 명령을 사용하여 Amazon S3의 IP 주소 범위를 확인합니다. 범위는 54.123.160.0/19이며, 이는 인터넷 게이트웨이를 가리키는 범위보다 구체적이지 않습니다. 따라서 IP 주소 범위 54.123.165.0/24를 향하는 모든 트래픽은 Amazon S3의 퍼블릭 IP 주소 범위를 유지하는 동안은 인터넷 게이트웨이를 계속 사용하고 엔드포인트를 사용하지 않습니다.

목적지	Target
10.0.0.0/16	로컬
54.123.165.0/24	igw-1a2b3c4d
pl-1a2b3c4d	vpce-11bb22cc

같은 리전에서 Amazon S3를 향하는 모든 트래픽이 엔드포인트를 통해 라우팅되게 하려면 라우팅 테이블의 라우팅을 조정해야 합니다. 이렇게 하려면 인터넷 게이트웨이에 대한 라우팅을 삭제하면 됩니다. 그러면 같은 리전에서 Amazon S3를 향하는 모든 트래픽은 엔드포인트를 사용하고, 라우팅 테이블과 연결된 서브넷은 프라이빗 서브넷이 됩니다.

목적지	Target
10.0.0.0/16	로컬
pl-1a2b3c4d	vpce-11bb22cc

## 엔드포인트 제한 사항

엔드포인트를 사용하려면 현재 제한 사항을 알고 있어야 합니다.

- 엔드포인트에 지정된 서비스에 대한 아웃바운드 트래픽을 허용하거나 거부하기 위해 네트워크 ACL의 아웃바운드 규칙에 접두사 목록 ID를 사용할 수 없습니다. 네트워크 ACL 규칙이 트래픽을 제한하는 경우 서비스에 대한 CIDR 블록(IP 주소 범위)을 대신 지정해야 합니다. 하지만 아웃바운드 보안 그룹 규칙에는 접두사 목록 ID를 사용할 수 있습니다. 자세한 내용은 [보안 그룹 \(p. 204\)](#) 단원을 참조하십시오.
- 엔드포인트는 동일 리전에서만 지원됩니다. VPC와 다른 리전의 AWS 서비스 간에 엔드포인트를 만들 수 없습니다.
- 엔드포인트에 태그를 지정할 수 없습니다.
- VPC 간에 또는 서비스 간에 엔드포인트를 전송할 수 없습니다.
- 엔드포인트 연결은 VPC 외부로 확장할 수 없습니다. VPN 연결, VPC 피어링 연결, AWS Direct Connect 연결 또는 VPC의 연결의 반대쪽에 있는 리소스는 엔드포인트를 사용하여 엔드포인트 서비스의 리소스와 통신할 수 없습니다.
- VPC에서 DNS 확인을 활성화해야 합니다. 또는 자체 DNS 서버를 사용 중인 경우, 필요한 서비스(예: Amazon S3)에 대한 DNS 요청이 AWS에서 유지 관리하는 IP 주소로 제대로 확인되어야 합니다. 자세한 내용은 [VPC와 함께 DNS 사용 \(p. 194\)](#) 단원을 참조하십시오.

Amazon S3에 고유한 규칙 및 제한은 [Amazon S3에 대한 엔드포인트 \(p. 204\)](#) 단원을 참조하십시오.

## 엔드포인트 사용 제어

기본적으로 IAM 사용자에게는 엔드포인트 사용 권한이 없습니다. IAM 사용자 정책을 만들어 사용자에게 엔드포인트를 생성, 수정, 설명, 삭제할 수 있는 권한을 부여할 수 있습니다. Amazon은 현재 `ec2:*VpcEndpoint*` API 작업의 일부 또는 `ec2:DescribePrefixLists` 작업에 대한 리소스 수준 권한을 지원하지 않습니다. 특정한 엔드포인트 또는 접두사 목록의 사용 권한을 IAM 정책을 만들어 사용자에게 부여할 수 없습니다. 자세한 내용은 다음 예를 참조하십시오. [8. VPC 엔드포인트 생성 및 관리 \(p. 136\)](#)

## 서비스에 대한 액세스 제어

엔드포인트를 만들 경우, 연결하려는 서비스에 대한 액세스를 제어하는 엔드포인트에 엔드포인트 정책을 연결합니다. 엔드포인트 정책은 JSON 형식으로 작성해야 합니다.

Amazon S3에 대한 엔드포인트를 사용할 경우, Amazon S3 버킷 정책을 사용하여 특정 엔드포인트 또는 특정 VPC의 버킷에 대한 액세스를 제어할 수도 있습니다. 자세한 내용은 [Amazon S3 버킷 정책 사용 \(p. 207\)](#) 단원을 참조하십시오.

항목

- [엔드포인트 정책 사용 \(p. 204\)](#)
- [보안 그룹 \(p. 204\)](#)

## 엔드포인트 정책 사용

VPC 엔드포인트 정책은 엔드포인트를 만들거나 수정 시 엔드포인트에 연결하는 IAM 리소스 정책입니다. 엔드포인트를 만들 때 정책을 추가하지 않으면 서비스에 대한 모든 액세스를 허용하는 기본 정책이 추가됩니다. 엔드포인트 정책은 IAM 사용자 정책 또는 서비스별 정책(예: S3 버킷 정책)을 무시하거나 교체하지 않습니다. 이는 엔드포인트에서 지정된 서비스로의 액세스를 제어하기 위한 별도의 정책입니다.

한 엔드포인트에는 한 개의 정책만 추가할 수 있지만 정책은 언제든지 수정할 수 있습니다. 정책을 수정할 경우, 변경 사항이 적용되기까지 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 [엔드포인트 수정 \(p. 209\)](#) 단원을 참조하십시오. 정책 작성에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책의 개요](#)를 참조하십시오.

엔드포인트 정책은 IAM 정책과 비슷하지만 다음에 유의하십시오.

- 지정된 서비스와 관련된 정책의 일부만 작동합니다. 엔드포인트 정책을 사용하여 VPC의 리소스가 다른 작업을 수행하도록 허용할 수 없습니다. 예를 들어 Amazon S3에 대한 엔드포인트의 엔드포인트 정책에 EC2 작업을 추가할 경우 작동하지 않습니다.
- 정책에는 Principal 요소를 포함시켜야 합니다. 자세한 내용은 IAM 사용 설명서의 [보안 주체](#)를 참조하십시오.

엔드포인트 정책 예는 다음 주제를 참조하십시오.

- [Amazon S3용 엔드포인트 정책 사용 \(p. 206\)](#)

## 보안 그룹

기본적으로 Amazon VPC 보안 그룹은 명시적으로 아웃바운드 액세스를 제한하지 않는 한 모든 아웃바운드 트래픽을 허용합니다. 보안 그룹의 아웃바운드 규칙이 제한된 경우, VPC에서 엔드포인트에 지정된 서비스로의 아웃바운드 트래픽을 허용하는 규칙을 추가해야 합니다. 이렇게 하려면 아웃바운드 규칙에서 서비스의 접두사 목록 ID를 대상으로 사용합니다. 자세한 내용은 [보안 그룹 수정 \(p. 209\)](#) 단원을 참조하십시오.

## Amazon S3에 대한 엔드포인트

VPC에서 Amazon S3 리소스로의 액세스를 이미 설정한 경우, 엔드포인트를 설정한 후에도 Amazon S3 DNS 이름을 사용하여 이러한 리소스에 액세스할 수 있습니다. 하지만 다음에 유의하십시오.

- 엔드포인트에는 Amazon S3 리소스에 액세스하기 위한 엔드포인트 사용을 제어하는 정책이 있습니다. 기본 정책은 VPC가 연결된 계정을 제외하고 AWS 계정에 대한 Amazon S3 리소스를 포함하여 모든 AWS 계정부터 모든 Amazon S3 리소스까지 자격 증명을 사용하는 VPC 내 모든 사용자 또는 서비스의 액세스를 허용합니다. 자세한 내용은 [서비스에 대한 액세스 제어 \(p. 204\)](#) 단원을 참조하십시오.

- Amazon S3가 수신한 해당 서브넷에 있는 인스턴스의 원본 IPv4 주소는 VPC의 퍼블릭 IPv4 주소에서 프라이빗 IPv4 주소로 변경됩니다. 엔드포인트는 네트워크 라우팅을 스위칭하고 열린 TCP 연결을 끊습니다. 전환 시 작업이 중단되고 퍼블릭 IPv4 주소를 사용하는 이전의 모든 연결은 재개되지 않습니다. 따라서 엔드포인트를 만들거나 수정할 때는 중요한 작업을 실행하지 않는 것이 좋으며 또한 연결이 끊어진 후에는 소프트웨어가 자동으로 Amazon S3에 다시 연결할 수 있는지를 테스트하는 것이 좋습니다.
- 버킷 정책 또는 IAM 정책을 사용하여 VPC IPv4 CIDR 범위(프라이빗 IPv4 주소 범위)로부터의 액세스를 허용할 수 없습니다. VPC CIDR 블록이 중첩되거나 동일할 수 있고 이로 인해 예기치 않은 결과가 발생할 수 있습니다. 대신에 버킷 정책을 사용하여 특정 엔드포인트 또는 특정 VPC에 대해 액세스를 제한하고, 라우팅 테이블을 사용하여 엔드포인트를 통해 Amazon S3의 리소스에 액세스할 수 있는 인스턴스를 제어할 수 있습니다.
- VPC 종단점을 통해서 Amazon S3에 요청할 때는 버킷 정책에 `aws:SourceIp` 조건을 사용할 수 없습니다. 버킷 정책의 문에 `aws:SourceIp` 조건이 포함되어 있는 경우, 값이 제공된 IP 주소 또는 범위와 일치하지 않습니다. 자세한 내용은 [Amazon S3 버킷 정책 사용 \(p. 207\)](#) 단원을 참조하십시오.
- 현재 엔드포인트는 교차 리전 요청을 지원하지 않습니다. 버킷과 같은 리전에 엔드포인트를 생성하는 지 확인하십시오. Amazon S3 콘솔 또는 `get-bucket-location` 명령을 사용하여 버킷의 위치를 확인할 수 있습니다. 리전별 Amazon S3 엔드포인트를 사용하여 버킷에 액세스하십시오(예: `mybucket.s3-us-west-2.amazonaws.com`). Amazon S3의 리전별 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 [Amazon Simple Storage Service\(S3\)](#)를 참조하십시오. AWS CLI를 사용하여 Amazon S3에 요청할 경우, 기본 리전을 버킷과 동일한 리전으로 설정하거나, 요청에 `--region` 파라미터를 사용하십시오.

Note

Amazon S3의 미국 표준 리전이 `us-east-1` 리전에 매핑된 것으로 간주하십시오.

- 엔드포인트는 현재 IPv4 트래픽에 대해서만 지원됩니다.

Amazon S3에서 엔드포인트를 사용하기 전에 다음의 일반 제한을 읽으십시오. [엔드포인트 제한 사항 \(p. 203\)](#).

VPC에서 다른 AWS 서비스를 사용할 경우, 이러한 서비스에서 특정 작업에 대해 S3 버킷을 사용할 수 있습니다. 엔드포인트 정책이 Amazon S3에 대해 모든 액세스를 허용하도록 하거나(기본 정책), 이러한 서비스에서 사용하는 특정 버킷에 대해 액세스를 허용하도록 합니다. 또는 이러한 서비스 중 어느 서비스도 사용하지 않는 서브넷에 엔드포인트를 만들어서 서비스가 퍼블릭 IP 주소를 사용하여 S3 버킷에 계속 액세스하도록 허용합니다.

다음 표에는 엔드포인트의 영향을 받을 수 있는 AWS 서비스와 각 서비스에 대한 특정 정보가 나와 있습니다.

AWS 서비스	참고
AWS CloudFormation	VPC에 대기 조건 또는 사용자 지정 리소스에 응답해야 하는 리소스가 있을 경우, 엔드포인트 정책은 최소한 이러한 리소스가 사용하는 특정 버킷에 대해 액세스를 허용해야 합니다. 자세한 내용은 <a href="#">AWS CloudFormation 및 VPC 엔드포인트</a> 를 참조하십시오.
AWS CodeDeploy	엔드포인트 정책이 Amazon S3에 대해 모든 액세스를 허용하거나, AWS CodeDeploy 배포를 위해 만든 S3 버킷에 대해 액세스를 허용해야 합니다.
Elastic Beanstalk	엔드포인트 정책이 최소한 Elastic Beanstalk 애플리케이션에 사용되는 S3 버킷에 대해 액세스를 허용해야 합니다. 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 <a href="#">Amazon S3와 함께 Elastic Beanstalk 사용</a> 을 참조하십시오.

AWS 서비스	참고
AWS OpsWorks	엔드포인트 정책이 최소한 AWS OpsWorks가 사용하는 특정 버킷에 대해 액세스를 허용해야 합니다. 자세한 내용은 AWS OpsWorks User Guide의 <a href="#">VPC에서 스택 실행</a> 을 참조하십시오.
Amazon WorkDocs	Amazon WorkSpaces의 Amazon WorkDocs 클라이언트나 EC2 인스턴스를 사용할 경우, 엔드포인트 정책은 Amazon S3에 대해 모든 액세스를 허용해야 합니다.
Amazon WorkSpaces	Amazon WorkSpaces는 Amazon S3를 직접 사용하지는 않지만, Amazon WorkSpaces 사용자에게 인터넷 액세스를 제공할 경우 다른 회사의 웹 사이트, HTML 이메일 및 인터넷 서비스에서 Amazon S3를 사용할 수 있습니다. 이러한 서비스가 올바르게 작동할 수 있도록 하기 위해 엔드포인트 정책이 Amazon S3에 대해 모든 액세스를 허용하도록 하십시오.

VPC 및 S3 버킷 간 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

## Amazon S3용 엔드포인트 정책 사용

다음은 Amazon S3에 액세스하기 위한 엔드포인트 정책의 예입니다.

### Important

IAM 사용자 정책, 엔드포인트 정책, S3 버킷 정책 및 Amazon S3 ACL 정책(있는 경우)과 같은 모든 유형의 정책은 Amazon S3에 액세스하기 위해 필요한 권한을 부여해야 합니다.

예: 특정 버킷에 대한 액세스 제한

특정 S3 버킷에 대해서만 액세스를 제한하는 정책을 만들 수 있습니다. 이는 VPC에 S3 버킷을 사용하는 다른 AWS 서비스가 있을 경우 유용합니다. 다음은 `my_secure_bucket`에 대해서만 액세스를 제한하는 정책의 예입니다.

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::my_secure_bucket",
        "arn:aws:s3:::my_secure_bucket/*"
      ]
    }
  ]
}
```

예: Amazon Linux AMI 리포지토리 액세스 활성화

Amazon Linux AMI 리포지토리는 각 리전의 Amazon S3 버킷입니다. VPC의 인스턴스가 종단점을 통해 리포지토리에 액세스하도록 하려면 이러한 버킷에 액세스하게 하는 종단점 정책을 만들 수 있습니다.

```
{
  "Statement": [
    {
      "Sid": "AmazonLinuxAMIRepositoryAccess",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::packages.*.amazonaws.com/*",
        "arn:aws:s3:::repo.*.amazonaws.com/*"
      ]
    }
  ]
}
```

## Amazon S3 버킷 정책 사용

버킷 정책을 사용하여 특정 엔드포인트 또는 특정 VPC의 버킷에 대한 액세스를 제어할 수 있습니다.

VPC 종단점을 통해서 Amazon S3에 요청할 때는 버킷 정책에 `aws:SourceIp` 조건을 사용할 수 없습니다. 조건이 지정된 IP 주소 또는 IP 주소 범위와 일치하지 않으므로 Amazon S3 버킷에 요청 시 원하지 않는 결과가 나타날 수 있습니다. 예:

- 버킷 정책에 `Deny` 결과와 하나 또는 제한된 범위의 IP 주소에서의 액세스만 허용하는 `NotIpAddress` 조건이 있습니다. 엔드포인트를 통해 버킷에 요청하는 경우, `NotIpAddress` 조건이 항상 일치되므로 문의 결과가 적용됩니다(정책의 다른 제약이 일치한다고 가정). 버킷에 대한 액세스는 거부됩니다.
- 버킷 정책에 `Deny` 결과와 하나 또는 제한된 범위의 IP 주소에 대해서만 액세스를 거부하는 `IpAddress` 조건이 포함되어 있습니다. 엔드포인트를 통해 버킷에 요청하는 경우, 조건이 일치되지 않으므로 문의 적용되지 않습니다. `IpAddress` 조건 없이 액세스를 허용하는 다른 문이 있다고 가정하면 버킷에 대한 액세스가 허용됩니다.

대신에 버킷 정책을 조정하여 특정 VPC 또는 특정 엔드포인트에 대한 액세스를 제한하십시오.

Amazon S3에 대한 버킷 정책 관련 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 [버킷 정책 및 사용자 정책 사용](#)을 참조하십시오.

예: 특정 엔드포인트에 대한 액세스 제한

다음은 엔드포인트 `vpce-1a2b3c4d`의 특정 버킷인 `my_secure_bucket`에 대해서만 액세스를 허용하는 S3 버킷 정책의 예입니다. 이 정책은 지정된 종단점이 사용되지 않으면 버킷에 대한 모든 액세스를 거부합니다. `aws:sourceVpce` 조건이 종단점을 지정하는 데 사용됩니다. `aws:sourceVpce` 조건은 VPC 엔드포인트 리소스에 대한 ARN을 요구하지 않고 엔드포인트 ID만 요구합니다.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [ "arn:aws:s3:::my_secure_bucket",
                   "arn:aws:s3:::my_secure_bucket/*" ],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-1a2b3c4d"
        }
      }
    }
  ]
}
```

```
}  
  }  
} ]  
}
```

예: 특정 VPC에 대한 액세스 제한

aws:sourceVpc 조건을 사용하여 특정 VPC에 대한 액세스를 제한하는 버킷 정책을 만들 수 있습니다. 이는 같은 VPC에 여러 엔드포인트가 구성되어 있으며, 모든 엔드포인트의 S3 버킷에 대한 액세스를 관리하려는 경우에 유용합니다. 다음은 VPC vpc-111bbb22가 my\_secure\_bucket과 해당 객체에 액세스할 수 있도록 허용하는 정책의 예입니다. 이 정책은 지정된 VPC가 사용되지 않으면 버킷에 대한 모든 액세스를 거부합니다. aws:sourceVpc 조건은 VPC 리소스의 ARN을 요구하지 않고 VPC ID만 요구합니다.

```
{  
  "Version": "2012-10-17",  
  "Id": "Policy1415115909152",  
  "Statement": [  
    {  
      "Sid": "Access-to-specific-VPC-only",  
      "Principal": "*",  
      "Action": "s3:*",  
      "Effect": "Deny",  
      "Resource": [ "arn:aws:s3:::my_secure_bucket",  
                   "arn:aws:s3:::my_secure_bucket/*" ],  
      "Condition": {  
        "StringNotEquals": {  
          "aws:sourceVpc": "vpc-111bbb22"  
        }  
      }  
    }  
  ]  
}
```

## 엔드포인트 작업

Amazon VPC 콘솔을 사용하여 엔드포인트를 만들고 관리할 수 있습니다.

항목

- [엔드포인트 만들기 \(p. 208\)](#)
- [보안 그룹 수정 \(p. 209\)](#)
- [엔드포인트 수정 \(p. 209\)](#)
- [엔드포인트 설명 \(p. 210\)](#)
- [엔드포인트 삭제 \(p. 210\)](#)

## 엔드포인트 만들기

엔드포인트를 만들려면 엔드포인트를 만들려는 VPC 및 연결을 설정할 서비스를 지정해야 합니다. 엔드포인트에 정책을 추가하고, 엔드포인트가 사용할 라우팅 테이블을 지정할 수도 있습니다.

엔드포인트를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택합니다.
3. [Create Endpoint]를 선택합니다.
4. 마법사의 첫 번째 단계에서 다음 정보를 입력한 후 [Next Step]을 선택합니다.

- 엔드포인트를 만들 VPC 및 연결할 서비스를 선택합니다.
  - 정책의 유형을 선택합니다. 기본 옵션인 [Full Access]를 그대로 사용하여 서비스에 대한 모든 액세스를 허용합니다. 또는 [Custom]을 선택한 후 AWS 정책 생성기를 사용하여 사용자 지정 정책을 만들거나, 정책 창에서 직접 정책을 입력할 수도 있습니다.
5. 마법사의 두 번째 단계에서는 엔드포인트가 사용할 라우팅 테이블을 선택합니다. 마법사는 엔드포인트에 대한 서비스를 향하는 트래픽을 가리키는 라우팅을 해당 테이블에 자동으로 추가합니다. 완료했으면 [Create Endpoint]를 선택합니다.

VPC 마법사를 사용하여 새로운 VPC를 만들고 동시에 엔드포인트를 만들 수 있습니다. 엔드포인트가 사용하는 라우팅 테이블을 지정하는 대신, 엔드포인트에 액세스할 수 있게 할 서브넷을 지정합니다. 마법사는 이러한 서브넷에 연결된 라우팅 테이블에 엔드포인트 라우팅을 추가합니다.

VPC 마법사를 사용하여 VPC 및 엔드포인트를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. Amazon VPC 대시보드에서 [Start VPC Wizard]를 선택합니다.
3. 필요에 맞는 VPC 구성을 선택한 후 [Select]를 선택합니다. 구성 유형에 대한 자세한 내용은 [시나리오 및 예시 \(p. 17\)](#)를 참조하십시오.
4. 마법사의 두 번째 페이지에서 필요한 VPC 설정을 입력합니다. [Add Endpoint]를 선택하고 다음 정보를 입력합니다.
  - 연결하고자 하는 서비스를 선택합니다.
  - [Subnet] 목록에서 엔드포인트에 액세스할 수 있게 할 서브넷을 선택합니다. 이러한 서브넷에 연결된 라우팅 테이블에 엔드포인트 라우팅이 포함됩니다.
  - [Policy] 목록에서 정책 유형을 선택합니다. 기본 옵션인 [Full Access]를 그대로 사용하여 서비스에 대한 모든 액세스를 허용합니다. 또는 [Custom]을 선택한 후 AWS 정책 생성기를 사용하여 사용자 지정 정책을 만들거나, 정책 창에서 직접 정책을 입력할 수도 있습니다.
5. 해당되는 경우, 마법사에서 남은 단계를 완료한 다음 [Create VPC]를 클릭합니다.

## 보안 그룹 수정

VPC 보안 그룹이 아웃바운드 트래픽을 제한할 경우, AWS 서비스를 향하는 트래픽을 허용하는 규칙을 추가하여 인스턴스를 유지해야 합니다.

엔드포인트에 대한 아웃바운드 규칙을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. VPC 보안 그룹을 선택하고 [Outbound Rules] 탭을 선택한 후 [Edit]를 선택합니다.
4. [Type] 목록에서 트래픽 유형을 선택한 후 필요한 경우, 포트 범위를 입력합니다. 예를 들어 인스턴스를 사용하여 Amazon S3에서 객체를 검색할 경우 [Type] 목록에서 [HTTPS]를 선택합니다.
5. [Destination] 목록에는 사용 가능한 AWS 서비스의 접두사 목록 ID와 이름이 표시됩니다. 엔드포인트 서비스의 접두사 목록 ID를 선택하거나 입력합니다.
6. [Save]를 선택합니다.

보안 그룹에 대한 자세한 내용은 [VPC의 보안 그룹 \(p. 95\)](#)을 참조하십시오.

## 엔드포인트 수정

정책을 변경하거나 삭제하고, 엔드포인트에서 사용하는 라우팅 테이블을 추가하거나 삭제하여 엔드포인트를 수정할 수 있습니다.



엔드포인트와 연결된 정책을 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택합니다.
3. 엔드포인트를 선택하고 [Actions]를 선택한 후 [Edit Policy]를 선택합니다.
4. 대화 상자에서 [Full Access]를 선택하여 모든 액세스를 허용할 수 있습니다. 또는 [Custom]을 선택한 후 AWS 정책 생성기를 사용하여 사용자 지정 정책을 만들거나, 정책 창에서 직접 정책을 입력할 수도 있습니다. 완료했으면 [Save Policy]를 선택합니다.

#### Note

정책 변경 사항이 적용되려면 몇 분 정도 걸릴 수 있습니다.

엔드포인트가 사용하는 라우팅 테이블을 추가 또는 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택합니다.
3. VPC endpoint를 선택하고 [Actions]를 선택한 후 [Choose Route Tables]를 선택합니다.
4. 대화 상자에서 필요한 라우팅 테이블을 선택하거나 선택 취소한 후 [Save]를 선택합니다.

## 엔드포인트 설명

Amazon VPC 콘솔을 사용하여 엔드포인트를 확인한 다음 각 엔드포인트의 정보를 확인할 수 있습니다.

엔드포인트에 대한 정보를 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택합니다.
3. 엔드포인트를 선택합니다.
4. [Summary] 탭에서는 엔드포인트에 대한 정보를 확인할 수 있습니다. 예를 들어 [Service] 필드에는 서비스에 대한 접두사 목록 이름이 표시됩니다.

[Route Tables] 탭에서는 엔드포인트가 사용하는 라우팅 테이블에 대한 정보를 확인할 수 있습니다. [Policy] 탭에서는 엔드포인트에 연결된 IAM 정책을 확인할 수 있습니다.

#### Note

[Policy] 탭에는 엔드포인트 정책만 표시됩니다. 엔드포인트 작업 권한이 있는 IAM 사용자를 위한 IAM 정책 관련 정보는 표시되지 않습니다. 또한 S3 버킷 정책과 같은 서비스별 정책도 표시되지 않습니다.

## 엔드포인트 삭제

엔드포인트가 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 엔드포인트를 삭제하면 엔드포인트가 사용하는 라우팅 테이블의 엔드포인트 라우팅도 삭제됩니다. 하지만 엔드포인트가 상주하는 VPC와 연결된 보안 그룹에는 아무런 영향이 없습니다.

엔드포인트를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택합니다.
3. 엔드포인트를 선택하고 [Actions]를 선택한 후 [Delete Endpoint]를 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.



## API 및 CLI 개요

이 페이지에서 설명한 작업은 명령줄 도구 또는 Amazon EC2 Query API를 사용하여 수행할 수 있습니다.

### VPC endpoint 만들기

- [create-vpc-endpoint](#)(AWS CLI)
- [New-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [CreateVpcEndpoint](#)(Amazon EC2 Query API)

### 접두사 목록 이름, ID, AWS 서비스에 대한 IP 주소 범위 확인

- [describe-prefix-lists](#)(AWS CLI)
- [Get-EC2PrefixList](#)(Windows PowerShell용 AWS 도구)
- [DescribePrefixLists](#)(Amazon EC2 Query API)

### VPC endpoint 수정

- [modify-vpc-endpoint](#)(AWS CLI)
- [Edit-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [ModifyVpcEndpoint](#)(Amazon EC2 Query API)

### VPC endpoint 설명

- [describe-vpc-endpoints](#)(AWS CLI)
- [Get-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpoints](#)(Amazon EC2 Query API)

### VPC endpoint를 만드는 데 사용할 수 있는 AWS 서비스 목록 확인

- [describe-vpc-endpoint-services](#)(AWS CLI)
- [Get-EC2VpcEndpointService](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpointServices](#)(Amazon EC2 Query API)

### VPC endpoint 삭제

- [delete-vpc-endpoints](#)(AWS CLI)
- [Remove-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [DeleteVpcEndpoints](#)(Amazon EC2 Query API)

를 사용하면 같은 리전 내에 있는 해당 계정의 VPC에 인스턴스를 연결할 수 있습니다. 이를 통해 VPC 보안 그룹을 인스턴스에 연결하여 인스턴스와 VPC의 인스턴스가 프라이빗 IPv4 주소를 사용해 서로 통신하도록 허용할 수 있습니다. 를 사용하면 이러한 플랫폼의 인스턴스 간 통신을 위해 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 사용할 필요가 없습니다. 프라이빗 및 퍼블릭 IPv4 주소에 대한 자세한 내용은 [VPC의 IP 주소 지정](#) (p. 76) 단원을 참조하십시오.

는 플랫폼을 지원하는 계정을 갖는 모든 사용자에게 제공되며 모든 인스턴스에 사용할 수 있습니다.

사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 인스턴스 시간 사용량에 대한 표준 요금이 그대로 적용됩니다.

에 대한 제한한 내용과 그 사용 방법은 Amazon EC2 사용 설명서의 다음 항목을 참조하십시오.

- [기본 사항](#)
- [의 제한 사항](#)
- [작업](#)
- [API 및 CLI 개요](#)

# VPN 연결

VPC 연결을 사용하여 원격 네트워크에 VPC를 연결할 수 있습니다. 다음은 사용자가 사용할 수 있는 몇 가지 연결 옵션입니다.

VPN 연결 옵션	설명
AWS 하드웨어 VPN	VPC와 원격 네트워크 사이에 IPsec, 하드웨어 VPN 연결을 생성할 수 있습니다. AWS 측 VPN 연결에서 가상 프라이빗 게이트웨이는 자동 장애 조치를 위한 2개의 VPN 엔드포인트를 제공합니다. VPN 연결의 원격 측에 있는 물리적 디바이스 또는 소프트웨어 애플리케이션인 고객 게이트웨이를 구성합니다. 자세한 내용은 <a href="#">VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 (p. 214)</a> 및 <a href="#">Amazon VPC 네트워크 관리자 안내서</a> 섹션을 참조하십시오.
AWS Direct Connect	AWS Direct Connect는 원격 네트워크에서 VPC까지 전용 프라이빗 연결을 제공합니다. 이 연결을 AWS 하드웨어 VPN 연결과 결합하여 IPsec 암호화 연결을 생성할 수 있습니다. 자세한 내용은 <a href="#">What is AWS Direct Connect? 섹션</a> (출처: AWS Direct Connect 사용 설명서) 단원을 참조하십시오.
AWS VPN CloudHub	원격 네트워크가 두 개 이상인 경우(예: 여러 지사 사무실), VPC를 통해 여러 개의 AWS 하드웨어 VPN 연결을 생성하여 이들 네트워크 사이의 통신을 활성화할 수 있습니다. 자세한 내용은 <a href="#">VPN CloudHub를 사용하여 사이트 간에 보안 통신 제공 (p. 223)</a> 단원을 참조하십시오.
소프트웨어 VPN	소프트웨어 VPN 어플라이언스를 실행 중인 VPC에서 Amazon EC2 인스턴스를 사용하여 원격 네트워크에 대한 VPN 연결을 생성할 수 있습니다. AWS는 소프트웨어 VPN 어플라이언스를 제공하거나 유지 관리하지 않지만, 파트너와 오픈 소스 커뮤니티에서 제공하는 다양한 제품 중에서 선택할 수 있습니다. <a href="#">AWS Marketplace</a> 에서 소프트웨어 VPN 어플라이언스를 찾으십시오.

이 섹션에서 다룰 주제는 다음과 같습니다.

- [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가 \(p. 214\)](#)
- [VPN CloudHub를 사용하여 사이트 간에 보안 통신 제공 \(p. 223\)](#)

다른 VPC 및 VPN 연결 옵션에 대한 자세한 내용은 [Amazon Virtual Private Cloud\(VPC\) 연결 옵션](#) 백서를 참조하십시오.

## VPC에 하드웨어 가상 프라이빗 게이트웨이 추가

기본적으로 가상 프라이빗 클라우드(VPC)로 시작하는 인스턴스는 자체 네트워크와 통신할 수 없습니다. VPC에 가상 프라이빗 게이트웨이를 연결하고 사용자 지정 라우팅 테이블을 생성하며 보안 그룹 규칙을 업데이트하여 VPC에서 네트워크에 액세스하도록 할 수 있습니다.

이 페이지에서 설명한 대로 이 절차를 수동으로 완료하거나 VPC 생성 마법사를 통해 이 중 여러 단계를 처리할 수 있습니다. VPC 생성 마법사를 사용하여 가상 프라이빗 게이트웨이를 설정하는 방법에 대한 자세한 정보는 [시나리오 3: 공인 및 사설 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC \(p. 34\)](#) 또는 [시나리오 4: 사설 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC \(p. 44\)](#) 단원을 참조하십시오.

VPN 연결이라는 용어는 일반적인 용어지만 Amazon VPC 설명서에서 VPN 연결은 VPC와 자체 네트워크 사이의 연결을 의미합니다. AWS는 인터넷 프로토콜 보안(IPsec) VPN 연결을 지원합니다.

### Important

현재 VPN 연결을 통한 IPv6 트래픽은 지원하지 않습니다.

### 항목

- [VPN의 구성 요소 \(p. 214\)](#)
- [VPN 구성의 예 \(p. 215\)](#)
- [VPN 라우팅 옵션 \(p. 215\)](#)
- [VPN 연결을 위해 필요한 항목 \(p. 215\)](#)
- [VPN 연결을 위해 2개의 VPN 터널 구성 \(p. 216\)](#)
- [중복 VPN 연결을 사용하여 장애 조치 제공 \(p. 216\)](#)
- [VPN 연결 설정 \(p. 217\)](#)
- [인스턴스의 종단 간 연결 테스트 \(p. 219\)](#)
- [손상된 자격 증명 교체 \(p. 220\)](#)
- [VPN 연결의 고정 경로 편집 \(p. 220\)](#)
- [VPN 연결 삭제 \(p. 220\)](#)
- [API 및 CLI 개요 \(p. 221\)](#)

VPC에 VPN 연결을 사용할 경우 요금 부과 방법에 대한 자세한 정보는 [Amazon VPC 제품 페이지](#)를 참조하십시오.

## VPN의 구성 요소

VPN 연결은 다음과 같은 구성 요소로 이루어집니다.

### 가상 프라이빗 게이트웨이

가상 프라이빗 게이트웨이는 VPN 연결의 Amazon 측 VPN 집선기입니다.

VPC 내 다른 구성 요소의 제한과 리전마다 보유할 수 있는 가상 프라이빗 게이트웨이 수에 대한 자세한 정보는 [Amazon VPC 제한 \(p. 224\)](#) 단원을 참조하십시오.

### 고객 게이트웨이

고객 게이트웨이는 VPN 연결을 위해 고객 측에 설치된 물리적 디바이스 또는 소프트웨어 애플리케이션입니다. VPN 연결을 생성하면 VPN 연결의 사용자 쪽에서 트래픽이 생성될 때 VPN 터널이 가동됩니다. 가상 프라이빗 게이트웨이는 개시 장치가 아닙니다. 고객 게이트웨이가 터널을 시작해야 합니다. VPN 연결에 유틸

시간이 발생할 경우(대개 10초, 구성에 따라 다름) 터널이 다운될 수 있습니다. 이를 방지하기 위해 네트워크 모니터링 도구를 사용하여 keepalive ping을 생성할 수 있습니다(예: IP SLA 사용).

고객 게이트웨이에 대한 자세한 정보는 Amazon VPC 네트워크 관리자 안내서의 [고객 게이트웨이](#)를 참조하십시오.

Amazon VPC 항목을 사용하여 테스트한 고객 게이트웨이 목록은 [Amazon Virtual Private Cloud FAQs](#)를 참조하십시오.

## VPN 구성의 예

다음 다이어그램은 단일 및 다중 VPN 연결을 보여 줍니다. VPC에는 가상 프라이빗 게이트웨이가 연결되어 있고, 귀사의 네트워크에는 고객 게이트웨이가 있습니다. 이 고객 게이트웨이는 VPN 연결을 사용하도록 구성해야 합니다. 네트워크로 바인딩되는 VPC의 모든 트래픽이 가상 프라이빗 게이트웨이로 라우팅되도록 라우팅을 설정합니다.

단일 VPC에 대해 여러 VPN 연결을 생성할 때 같은 외부 위치에 대해 중복 연결을 생성하도록 두 번째 고객 게이트웨이를 구성할 수 있습니다. 이 고객 게이트웨이를 사용하여 여러 지리적 위치에 대해 VPN 연결을 생성할 수도 있습니다.

### 단일 VPN 연결

### 여러 VPN 연결

## VPN 라우팅 옵션

VPN 연결을 생성할 때 사용하려는 라우팅 유형을 지정하고 서브넷의 라우팅 테이블을 업데이트해야 합니다. 라우팅 테이블은 네트워크 트래픽이 전달되는 위치를 결정합니다. 따라서 VPC에서 하나 이상의 VPN 연결로 전달되는 트래픽은 가상 프라이빗 게이트웨이로 라우팅해야 합니다.

VPN 디바이스 모델과 메이커에 따라 라우팅 유형을 선택할 수 있습니다. VPN 디바이스에서 BGP(Border Gateway Protocol)를 지원할 경우 VPN 연결을 구성할 때 동적 라우팅을 지정하십시오. 장치에서 BGP를 지원하지 않을 경우 고정 라우팅을 지정하십시오. Amazon VPC 항목을 사용하여 테스트한 고정 및 동적 라우팅 디바이스 목록은 [Amazon Virtual Private Cloud FAQ](#)를 참조하십시오.

BGP 디바이스를 사용할 때는 디바이스에서 BGP를 사용하여 그 경로를 가상 프라이빗 게이트웨이에 알리기 때문에 VPN 연결에 대한 고정 경로를 지정할 필요가 없습니다. BGP를 지원하지 않는 디바이스를 사용하는 경우 고정 라우팅을 선택하고, 가상 프라이빗 게이트웨이에 전달할 경로(IP 접두사)를 네트워크에 대해 입력해야 합니다. BGP 광고를 통하거나 고정 경로 입력을 통해 가상 프라이빗 게이트웨이에 알려진 IP 접두사만 VPC에서 오는 트래픽을 수신할 수 있습니다. 가상 프라이빗 게이트웨이는 수신된 BGP 알림, 정적 라우팅 항목 또는 연결된 VPC CIDR의 외부로 전달되는 다른 모든 트래픽을 라우팅하지 않습니다.

BGP 프로토콜은 첫 번째 터널이 다운될 경우 두 번째 VPN 터널에 대한 장애 조치를 지원할 수 있는 강력한 라이브니스 탐지 검사를 제공하므로 가능하다면 BGP 지원 디바이스를 사용하는 것이 좋습니다. 또한 BGP를 지원하지 않는 디바이스는 상태 확인을 수행하여 필요할 경우 두 번째 터널로 장애 조치를 지원할 수 있습니다.

## VPN 연결을 위해 필요한 항목

Amazon VPC를 VPN 연결에 사용하려면 귀하 또는 네트워크 관리자가 물리적 어플라이언스를 고객 게이트웨이로 지정하고 구성해야 합니다. AWS에서는 VPN 사전 공유 키 및 VPN 연결 설정과 관련된 그 밖의 파라미터를 포함하여 필요한 구성 정보를 제공합니다. 대개 네트워크 관리자가 이 구성을 수행합니다. 고객 게이트웨이 요구 사항 및 구성에 대한 자세한 정보는 [Amazon VPC 네트워크 관리자 안내서](#) 단원을 참조하십시오.

다음 표에는 VPN 연결을 설정하기 위해 필요한 정보가 나와 있습니다.

항목	용도	의견
고객 게이트웨이 유형(예: Cisco ASA, Juniper J-Series, Juniper SSG, Yamaha)	고객 게이트웨이를 구성하는 데 사용할 반환된 정보의 형식을 지정하는 방법을 명시합니다.	테스트한 특정 디바이스에 대한 정보는 <a href="#">What customer gateway devices are known to work with Amazon VPC?</a> (출처: Amazon VPC FAQ)를 참조하십시오.
고객 게이트웨이 외부 인터페이스의 인터넷 라우팅 가능 IP 주소(고정)	고객 게이트웨이를 생성하고 구성하는 데 사용됩니다. YOUR_UPLINK_ADDRESS라고 합니다.	퍼블릭 IP 주소 값은 고정 주소여야 합니다. 고객 게이트웨이가 NAT-T(NAT traversal)를 지원하는 NAT(Network Address Translation) 디바이스 뒤에 상주하는 경우 NAT 디바이스의 퍼블릭 IP 주소를 사용하고 방화벽 규칙을 수정하여 UDP 포트 4500 차단을 해제합니다.
(선택 사항) 동적으로 라우팅되는 VPN 연결을 생성하는 경우 고객 게이트웨이의 BGP(Border Gateway Protocol) ASN(자율 시스템 번호)	고객 게이트웨이를 생성하고 구성하는 데 사용됩니다. YOUR_BGP_ASN이라고 합니다. 콘솔에서 VPC 마법사를 사용하여 VPC를 설정하는 경우 자동으로 65000이 ASN으로 사용됩니다.	네트워크에 할당된 기존 ASN을 사용할 수 있습니다. 기존 ASN이 없는 경우에는 (64512-65534 범위의) 프라이빗 ASN을 사용할 수 있습니다. ASN에 대한 자세한 정보는 <a href="#">Wikipedia 기사</a> 를 참조하십시오.  Amazon VPC에서는 2바이트 ASN 숫자를 지원합니다.
VPN 연결을 통해 VPC에 알릴 내부 네트워크 IP 범위	고정 경로를 지정하는 데 사용됩니다.	

## VPN 연결을 위해 2개의 VPN 터널 구성

VPN 연결을 사용하여 네트워크를 VPC에 연결합니다. VPN 연결마다 2개의 터널이 있으며, 고유의 가상 프라이빗 게이트웨이 퍼블릭 IP 주소가 각 터널에 사용됩니다. 중복성을 위해 두 터널 모두 구성해야 합니다. 유지 관리를 위해 가동을 중지하는 등의 이유로 한 터널을 사용할 수 없을 때 특정 VPN 연결에 사용 가능한 터널로 네트워크 트래픽이 자동 라우팅됩니다.

다음 다이어그램은 VPN 연결의 두 터널을 보여 줍니다.

## 중복 VPN 연결을 사용하여 장애 조치 제공

앞에서 설명한 대로 VPN 연결에는 터널 2개가 있어 VPN 연결 중 하나를 사용할 수 없어도 연결이 보장됩니다. 고객 게이트웨이를 사용할 수 없을 때 연결이 끊어지지 않도록 두 번째 고객 게이트웨이를 사용하여 VPC와 가상 프라이빗 게이트웨이에 대한 두 번째 VPN 연결을 설정할 수 있습니다. 중복 VPN 연결 및 고객 게이트웨이를 사용하면 고객 게이트웨이 중 하나에서 유지 관리를 수행하는 동안에도 두 번째 고객 게이트웨이의 VPN 연결을 통해 트래픽이 계속 전송됩니다. 네트워크에 중복 VPN 연결 및 고객 게이트웨이를 설정하려면 두 번째 VPN 연결을 설정해야 합니다. 두 번째 VPN 연결에 사용되는 고객 게이트웨이 IP 주소는 공개적으로 액세스할 수 있어야 합니다.

다음 다이어그램은 각 VPN 연결의 터널 2개와 고객 게이트웨이 2개를 보여 줍니다.

동적으로 라우팅되는 VPN 연결은 BGP(Border Gateway Protocol)를 사용하여 고객 게이트웨이와 가상 프라이빗 게이트웨이 간에 라우팅 정보를 교환합니다. 고정으로 라우팅되는 VPN 연결에서는 고객 게이트웨이의 귀하 측 네트워크의 고정 경로를 입력해야 합니다. BGP를 통해 알려지고 고정으로 입력된 경로 정보를 사용하여 양 측의 게이트웨이는 사용 가능한 터널을 확인하고 오류가 발생할 경우 트래픽을 다시 라우팅할 수 있

습니다. BGP(사용 가능한 경우)에서 제공한 라우팅 정보를 사용하여 사용 가능한 경로를 선택하도록 네트워크를 구성하는 것이 좋습니다. 네트워크의 아키텍처에 따라 정확한 구성이 결정됩니다.

## VPN 연결 설정

다음 절차에 따라 VPN 연결을 수동으로 설정하십시오. 대신 VPC와 서브넷을 생성하고 VPC 마법사를 사용하여 이 절차의 처음 5단계를 완료할 수 있습니다. 자세한 내용은 [시나리오 3 구현 \(p. 40\)](#) 또는 [시나리오 4 구현 \(p. 47\)](#)을 참조하십시오.

VPN 연결을 설정하려면 다음 단계를 완료해야 합니다.

- 단계 1: [고객 게이트웨이 생성 \(p. 217\)](#)
- 단계 2: [가상 프라이빗 게이트웨이 생성 \(p. 217\)](#)
- 단계 3: [라우팅 테이블에서 경로 전파 활성화 \(p. 217\)](#)
- 단계 4: [보안 그룹을 업데이트하여 인바운드 SSH, RDP 및 ICMP 액세스 활성화 \(p. 218\)](#)
- 단계 5: [VPN 연결 생성 및 고객 게이트웨이 구성 \(p. 218\)](#)
- 단계 6: [서브넷으로 인스턴스 시작 \(p. 219\)](#)

이 절차는 하나 이상의 서브넷이 포함된 VPC가 있고 필요한 네트워크 정보를 알고 있다는 가정 하에 진행됩니다([VPN 연결을 위해 필요한 항목 \(p. 215\)](#) 단원 참조).

## 고객 게이트웨이 생성

고객 게이트웨이를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Customer Gateways]를 선택한 후 [Create Customer Gateway]를 선택합니다.
3. [Create Customer Gateway] 대화 상자에서 다음을 완료한 후 [Yes, Create]를 선택합니다.
  - 필요할 경우 [Name tag] 필드에 고객 게이트웨이 이름을 입력합니다. Name 키와 지정한 값으로 태그가 생성됩니다.
  - [Routing] 목록에서 라우팅 유형을 선택합니다.
  - 동적 라우팅을 선택한 경우 [BGP ASN] 필드에 BGP(Border Gateway Protocol) ASN(자율 시스템 번호)을 입력합니다.
  - 고객 게이트웨이 디바이스의 인터넷 라우팅 가능한 고정 IP 주소를 [IP Address] 필드에 입력합니다. 고객 게이트웨이가 NAT-T를 지원하는 NAT 디바이스 뒤에 상주하는 경우 NAT 디바이스의 퍼블릭 IP 주소를 사용합니다.

## 가상 프라이빗 게이트웨이 생성

가상 프라이빗 게이트웨이를 생성하려면

1. 탐색 창에서 [Virtual Private Gateways]를 선택한 후 [Create Virtual Private Gateway]를 선택합니다.
2. 필요할 경우 가상 프라이빗 게이트웨이의 이름을 입력한 후 [Yes, Create]를 선택합니다.
3. 생성된 가상 프라이빗 게이트웨이를 선택한 후 [Attach to VPC]를 선택합니다.
4. [Attach to VPC] 대화 상자의 목록에서 VPC를 선택한 후 [Yes, Attach]를 선택합니다.

## 라우팅 테이블에서 경로 전파 활성화

VPC의 인스턴스가 고객 게이트웨이에 도달하도록 하려면 VPN 연결에 사용되는 경로를 포함하고 이 경로를 가상 프라이빗 게이트웨이로 연결하도록 라우팅 테이블을 구성해야 합니다. 라우팅 테이블의 경로 전파가 자동으로 해당 경로를 테이블로 전파하도록 할 수 있습니다.

정적 라우팅의 경우, VPN 구성에 지정하는 고정 IP 접두사는 VPN 연결 상태가 UP일 때 라우팅 테이블로 전파됩니다. 이와 마찬가지로 동적 라우팅의 경우, BGP를 통해 알려지고 고객 게이트웨이에서 받은 경로는 VPN 연결 상태가 UP일 때 라우팅 테이블에 전파됩니다.

경로 전파를 활성화하려면

1. 탐색 창에서 [Route Tables]를 선택한 후 서브넷에 연결된 라우팅 테이블을 선택합니다. 기본적으로 VPC의 기본 라우팅 테이블입니다.
2. 세부 정보 창의 [Route Propagation] 탭에서 [Edit]를 선택하고 이전 절차에서 생성된 가상 프라이빗 게이트웨이를 선택한 후 [Save]를 선택합니다.

#### Note

고정 라우팅의 경우 경로 전파를 활성화하지 않으면 VPN 연결에 사용되는 고정 경로를 수동으로 입력해야 합니다. 그러려면 라우팅 테이블을 선택하고 세부 정보 창의 [Routes] 탭에서 [Edit]를 선택합니다. VPN 연결에 사용되는 정적 경로를 [Destination] 필드에 추가하고 [Target] 목록에서 가상 프라이빗 게이트웨이 ID를 선택한 후 [Save]를 선택합니다.

## 보안 그룹을 업데이트하여 인바운드 SSH, RDP 및 ICMP 액세스 활성화

보안 그룹에 규칙을 추가하여 인바운드 SSH, RDP 및 ICMP 액세스를 활성화하려면

1. 탐색 창에서 [Security Groups]를 선택한 후 VPC의 기본 보안 그룹을 선택합니다.
2. 세부 정보 창의 [Inbound] 탭에서 네트워크로부터 인바운드 SSH, RDP 및 ICMP 액세스를 허용하는 규칙을 추가한 후 [Save]를 선택합니다. 인바운드 규칙 추가에 대한 자세한 정보는 [규칙 추가 및 제거 \(p. 100\)](#) 단원을 참조하십시오.

## VPN 연결 생성 및 고객 게이트웨이 구성

VPN 연결을 생성하고 고객 게이트웨이를 구성하려면

1. 탐색 창에서 [VPN Connections]를 선택한 다음 [Create VPN Connection]을 선택합니다.
2. [Create VPN Connection] 대화 상자에서 다음 작업을 수행한 후 [Yes, Create]를 선택합니다.
  - 필요할 경우 [Name tag] 필드에 VPN 연결 이름을 입력합니다. Name 키와 지정한 값으로 태그가 생성됩니다.
  - 앞에서 생성한 가상 프라이빗 게이트웨이를 선택합니다.
  - 앞에서 생성한 고객 게이트웨이를 선택합니다.
  - VPN 라우터에서 BGP(Border Gateway Protocol)를 지원하는지 여부에 따라 라우팅 옵션 중 하나를 선택합니다.
    - VPN 라우터에서 BGP를 지원하면 [Dynamic (requires BGP)]를 선택합니다.
    - VPN 라우터에서 BGP를 지원하지 않으면 [Static]을 선택합니다. [Static IP Prefixes] 필드에서 VPN 연결의 프라이빗 네트워크에 대한 IP 접두사를 각각 쉼표로 구분하여 지정합니다.
3. VPN 연결이 생성되는 데 몇 분 정도 걸릴 수 있습니다. 준비가 되면 연결을 선택한 후 [Download Configuration]을 선택합니다.
4. [Download Configuration] 대화 상자에서 고객 게이트웨이 디바이스나 소프트웨어에 해당하는 공급업체, 플랫폼 및 소프트웨어를 선택한 후 [Yes, Download]를 선택합니다.
5. [Amazon VPC 네트워크 관리자 안내서](#) 문서와 함께 구성 파일을 네트워크 관리자에게 제공합니다. 네트워크 관리자가 고객 게이트웨이를 구성한 후 VPN 연결을 사용할 수 있습니다.



## 서브넷으로 인스턴스 시작

서브넷에서 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 Launch Instance를 선택합니다.
3. [Choose an Amazon Machine Image (AMI)] 페이지에서 AMI를 선택한 후 [Select]를 선택합니다.
4. 인스턴스 유형을 선택하고 [Next: Configure Instance Details]를 선택합니다.
5. [Configure Instance Details] 페이지의 [Network] 목록에서 VPC를 선택하고 [Subnet] 목록에서 서브넷을 선택합니다. [Configure Security Group] 페이지가 나타날 때까지 [Next]를 선택합니다.
6. [Select an existing security group] 옵션을 선택한 후 앞에서 수정한 기본 그룹을 선택합니다. [Review and Launch]를 선택합니다.
7. 선택한 설정을 검토합니다. 필요에 따라 변경한 후 [Launch]를 선택하여 키 페어를 선택하고 인스턴스를 시작합니다.

## 인스턴스의 종단 간 연결 테스트

VPN 연결을 설정하고 인스턴스를 시작한 후 인스턴스를 ping하여 연결을 테스트할 수 있습니다. ping 요청에 응답하는 AMI를 사용해야 하며, 인바운드 ICMP를 활성화하도록 인스턴스의 보안 그룹을 구성해야 합니다. Amazon Linux AMI 중 하나를 사용하는 것이 좋습니다. Windows Server를 실행하는 인스턴스를 사용하는 경우 인스턴스에 로그인하여 Windows 방화벽에서 인바운드 ICMPv4를 활성화해야 인스턴스를 ping할 수 있습니다.

### Important

인스턴스에 대한 트래픽을 필터링하는 VPC의 네트워크 ACL 또는 보안 그룹을 구성하여 인바운드 및 아웃바운드 ICMP 트래픽을 허용해야 합니다.

Amazon VPC 콘솔을 사용하거나 Amazon EC2 API/CLI를 사용하여 VPN 연결의 상태를 모니터링할 수 있습니다. 상태, 마지막으로 상태가 변경된 후의 시간, 오류를 설명하는 텍스트를 포함하여 VPN 연결에 대한 정보를 볼 수 있습니다.

종단 간 연결을 테스트하려면

1. 인스턴스가 실행되면 프라이빗 IP 주소(예: 10.0.0.4)를 가져옵니다. Amazon EC2 콘솔에 주소가 인스턴스 세부 정보의 일부로 표시됩니다.
2. 고객 게이트웨이 뒤에 있는 네트워크의 컴퓨터에서 ping 명령을 인스턴스의 프라이빗 IP 주소와 함께 사용하십시오. 올바른 응답은 다음과 유사합니다.

```
PROMPT> ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),

Approximate round trip times in milliseconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

이제 SSH 또는 RDP를 사용하여 VPC의 인스턴스에 연결할 수 있습니다. Linux 인스턴스 연결 방법에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결](#)을 참조하십시오.

Windows 인스턴스 연결 방법에 대한 자세한 정보는 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

## 손상된 자격 증명 교체

VPN 연결의 터널 자격 증명이 손상된 것 같으면 IKE 사전 공유 키를 변경할 수 있습니다. 그러려면 VPN 연결을 삭제하고 동일한 가상 프라이빗 게이트웨이를 사용하여 새 VPN 연결을 생성한 후 고객 게이트웨이에 새 키를 구성하십시오. 또한 VPN 연결을 다시 생성할 때 터널의 내부 및 외부 주소가 변경될 수 있으므로 이 주소가 일치하는지 확인해야 합니다. 이 절차를 수행하는 동안 VPC의 인스턴스와 통신이 중단되지만 인스턴스는 중단되지 않고 계속 실행됩니다. 네트워크 관리자가 새 구성 정보를 구현한 후 새 자격 증명이 VPN 연결에 사용되며 VPC의 인스턴스에 대한 네트워크 연결이 재개됩니다.

### Important

이 절차에서는 네트워크 관리자 그룹의 도움이 필요합니다.

IKE 사전 공유 키를 변경하려면

1. VPN 연결을 삭제합니다. 자세한 내용은 [VPN 연결 삭제 \(p. 220\)](#) 단원을 참조하십시오. VPC나 가상 프라이빗 게이트웨이는 삭제할 필요가 없습니다.
2. 새 VPN 연결을 생성하고 새 구성 파일을 다운로드합니다. 자세한 내용은 [VPN 연결 생성 및 고객 게이트웨이 구성 \(p. 218\)](#) 단원을 참조하십시오.

## VPN 연결의 고정 경로 편집

고정 라우팅의 경우 VPN 구성의 고정 경로를 추가, 수정 또는 삭제할 수 있습니다.

고정 경로를 추가, 수정 또는 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [VPN Connections]를 선택합니다.
3. [Static Routes] 탭에서 [Edit]를 선택합니다.
4. 기존의 고정 IP 접두사를 수정하거나 [Remove]를 선택하여 이 접두사를 삭제합니다. [Add Another Route]를 선택하여 구성에 새 IP 접두사를 추가합니다. 마지막으로 [Save]를 선택합니다.

### Note

라우팅 테이블에 대해 경로 전파를 활성화하지 않은 경우 라우팅 테이블의 경로를 수동으로 업데이트하여 업데이트된 고정 IP 접두사를 VPN 연결에 반영해야 합니다. 자세한 내용은 [라우팅 테이블에서 경로 전파 활성화 \(p. 217\)](#) 단원을 참조하십시오.

## VPN 연결 삭제

VPN 연결이 더 이상 필요하지 않으면 삭제할 수 있습니다.

### Important

VPN 연결을 삭제한 후 새 VPN 연결을 생성할 경우 새 구성 정보를 다운로드하고 네트워크 관리자 가 고객 게이트웨이를 다시 구성하도록 해야 합니다.

VPN 연결을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창에서 [VPN Connections]를 선택합니다.
3. VPN 연결을 선택하고 [Delete]를 선택합니다.
4. [Delete VPN Connection] 대화 상자에서 [Yes, Delete]를 선택합니다.

고객 게이트웨이를 더 이상 필요로 하지 않는 경우 이를 삭제할 수 있습니다. VPN 연결에 사용 중인 고객 게이트웨이는 삭제할 수 없습니다.

고객 게이트웨이를 삭제하려면

1. 탐색 창에서 [Customer Gateways]를 선택합니다.
2. 삭제할 고객 게이트웨이를 선택하고 [Delete]를 선택합니다.
3. [Delete Customer Gateway] 대화 상자에서 [Yes, Delete]를 선택합니다.

VPC에 가상 프라이빗 게이트웨이가 더 이상 필요하지 않으면 이를 분리할 수 있습니다.

가상 프라이빗 게이트웨이를 분리하려면

1. 탐색 창에서 [Virtual Private Gateways]를 선택합니다.
2. 가상 프라이빗 게이트웨이를 선택하고 [Detach from VPC]를 선택합니다.
3. [Detach from VPC] 대화 상자에서 [Yes, Detach]를 선택합니다.

분리된 가상 프라이빗 게이트웨이가 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 아직 VPC에 연결된 가상 프라이빗 게이트웨이는 삭제할 수 없습니다.

가상 프라이빗 게이트웨이를 삭제하려면

1. 탐색 창에서 [Virtual Private Gateways]를 선택합니다.
2. 삭제할 가상 프라이빗 게이트웨이를 선택하고 [Delete]를 선택합니다.
3. [Delete Virtual Private Gateway] 대화 상자에서 [Yes, Delete]를 선택합니다.

## API 및 CLI 개요

명령줄 또는 API 작업을 사용하여 VPN 연결을 설정하고 관리할 수 있습니다. 사용 가능한 API 작업 목록을 비롯한 자세한 정보는 [Amazon VPC에 액세스](#) (p. 4) 단원을 참조하십시오.

고객 게이트웨이 생성

- [CreateCustomerGateway](#)(Amazon EC2 쿼리 API)
- [create-customer-gateway](#)(AWS CLI)
- [New-EC2CustomerGateway](#)(Windows PowerShell용 AWS 도구)

가상 프라이빗 게이트웨이 생성

- [CreateVpnGateway](#)(Amazon EC2 쿼리 API)
- [create-vpn-gateway](#)(AWS CLI)
- [New-EC2VpnGateway](#)(Windows PowerShell용 AWS 도구)

경로 전파 활성화

- [EnableVgwRoutePropagation](#)(Amazon EC2 쿼리 API)

- [enable-vgw-route-propagation](#)(AWS CLI)
- [Enable-EC2VgwRoutePropagation](#)(Windows PowerShell용 AWS 도구)

#### 보안 그룹 업데이트

- CLI를 사용하는 보안 그룹 작업에 대한 자세한 정보는 [API 및 CLI 개요 \(p. 102\)](#) 단원을 참조하십시오.

#### VPN 연결 생성

- [CreateVpnConnection](#)(Amazon EC2 쿼리 API)
- [create-vpn-connection](#)(AWS CLI)
- [New-EC2VpnConnection](#)(Windows PowerShell용 AWS 도구)

#### 고정 경로 추가

- [CreateVpnConnectionRoute](#)(Amazon EC2 쿼리 API)
- [create-vpn-connection-route](#)(AWS CLI)
- [New-EC2VpnConnectionRoute](#)(Windows PowerShell용 AWS 도구)

#### 고정 경로 삭제

- [DeleteVpnConnectionRoute](#)(Amazon EC2 쿼리 API)
- [delete-vpn-connection-route](#)(AWS CLI)
- [Remove-EC2VpnConnectionRoute](#)(Windows PowerShell용 AWS 도구)

#### VPN 연결 삭제

- [DeleteVpnConnection](#)(Amazon EC2 쿼리 API)
- [delete-vpn-connection](#)(AWS CLI)
- [Remove-EC2VpnConnection](#)(Windows PowerShell용 AWS 도구)

#### 고객 게이트웨이 삭제

- [DeleteCustomerGateway](#)(Amazon EC2 쿼리 API)
- [delete-customer-gateway](#)(AWS CLI)
- [Remove-EC2CustomerGateway](#)(Windows PowerShell용 AWS 도구)

#### 가상 프라이빗 게이트웨이 분리

- [DetachVpnGateway](#)(Amazon EC2 쿼리 API)
- [detach-vpn-gateway](#)(AWS CLI)
- [Dismount-EC2VpnGateway](#)(Windows PowerShell용 AWS 도구)

#### 가상 프라이빗 게이트웨이 삭제

- [DeleteVpnGateway](#)(Amazon EC2 쿼리 API)
- [delete-vpn-gateway](#)(AWS CLI)
- [Remove-EC2VpnGateway](#)(Windows PowerShell용 AWS 도구)

## VPN CloudHub를 사용하여 사이트 간에 보안 통신 제공

다수의 VPN 연결을 사용하는 경우 AWS VPN CloudHub로 사이트 간에 보안 통신을 제공할 수 있습니다. 이를 통해 원격 사이트가 VPC 뿐만 아니라 서로 통신할 수 있습니다. VPN CloudHub는 VPC와 함께 또는 VPC 없이 사용할 수 있는 간단한 허브 앤 스포크 모델에서 작동합니다. 이러한 설계는 여러 지사가 있고 기존 인터넷 연결을 사용하는 고객이 원격 지사 간에 기본 또는 백업 연결을 위해 편리하고도 경제적인 허브 앤 스포크 모델을 구현하고자 할 때 적합합니다.

다음 다이어그램은 VPN CloudHub 아키텍처를 보여 주며, 파란색 점선은 VPN 연결을 통해 라우팅되는 원격 사이트 간의 네트워크 트래픽을 나타냅니다.

AWS VPN CloudHub를 사용하려면 여러 고객 게이트웨이를 포함하는 가상 프라이빗 게이트웨이를 생성해야 합니다. 각 게이트웨이에 동일한 BGP(Border Gateway Protocol) ASN(자율 시스템 번호)을 사용할 수 있으며, 원할 경우 고유한 ASN을 사용할 수도 있습니다. 고객 게이트웨이는 VPN 연결을 통해 적절한 라우팅(BGP 접두사)을 공급합니다. 이처럼 라우팅을 공급하면 각 BGP 피어에서 이를 수신하여 다시 공급함으로써 각 사이트는 다른 사이트와 데이터를 주고받을 수 있습니다. 사이트의 IP 범위가 서로 중복되어서는 안 됩니다. 각 사이트는 스탠다드 VPN 연결을 사용하는 것처럼 VPC에서 데이터를 송수신할 수도 있습니다.

가상 프라이빗 게이트웨이에 대해 AWS Direct Connect 연결을 사용하는 사이트도 AWS VPN CloudHub의 일부가 될 수 있습니다. 예를 들어, 뉴욕 본사에서는 VPC에 대해 AWS Direct Connect 연결을 설정할 수 있고, 지사에서는 VPC에 대해 VPN 연결을 사용할 수 있습니다. 로스앤젤레스와 마이애미의 지사는 데이터를 서로 주고받을 수 있고, 본사와도 주고받을 수 있습니다. 이들 모두는 AWS VPN CloudHub를 사용합니다.

AWS VPN CloudHub를 구성하려면 AWS Management Console을 사용하여 ASN과 게이트웨이의 고유한 퍼블릭 IP 주소를 각각 갖는 여러 개의 고객 게이트웨이를 생성해야 합니다. 그런 다음 각 고객 게이트웨이에서 공통 가상 프라이빗 게이트웨이로 VPN 연결을 만듭니다. 각 VPN 연결은 특정 BGP 라우팅을 공급해야 합니다. 그러기 위해서 VPN 연결에 사용되는 VPN 구성 파일에 네트워크 명령문을 사용합니다. 네트워크 명령문은 사용하는 라우터 유형에 따라 약간 다릅니다.

AWS VPN CloudHub를 사용할 경우 일반 Amazon VPC VPN 연결 요금이 청구됩니다. 각 VPN이 가상 프라이빗 게이트웨이로 연결될 때 시간당 연결 요금이 청구됩니다. AWS VPN CloudHub를 사용하여 한 사이트에서 다른 사이트로 데이터를 전송할 경우 해당 사이트에서 가상 프라이빗 게이트웨이로 데이터를 전송하는 데 따른 비용은 없습니다. 가상 프라이빗 게이트웨이에서 엔드포인트까지 릴레이되는 데이터의 스탠다드 AWS 데이터 전송 요금만 청구됩니다. 예를 들어, 로스앤젤레스에 한 사이트, 뉴욕에 두 번째 사이트가 있으며 두 사이트에 모두 가상 프라이빗 게이트웨이에 대한 VPN 연결이 설정되어 있는 경우, 각 VPN 연결에 대해 시간당 0.05 USD가 청구됩니다(시간당 총 0.10 USD). 또한 로스앤젤레스에서 뉴욕(또는 뉴욕에서 로스앤젤레스)으로 전송하는 모든 데이터에 대해서는 각 VPN 연결을 통과하는 스탠다드 AWS 데이터 전송 요금이 청구됩니다. VPN 연결을 통해 가상 프라이빗 게이트웨이로 전송되는 네트워크 트래픽은 무료이나, VPN 연결을 통해 가상 프라이빗 게이트웨이에서 엔드포인트로 전송되는 네트워크 트래픽의 경우 스탠다드 AWS 데이터 전송 요금이 청구됩니다. 자세한 내용은 [VPN 연결 요금](#)을 참조하십시오.

# Amazon VPC 제한

다음 표에는 AWS 계정 관련 각 리전의 Amazon VPC 리소스에 대한 제한 값이 나열되어 있습니다. 달리 표시되지 않은 한 [Amazon VPC 한도 양식](#)을 사용하여 이 한도 증가를 요청할 수 있습니다. 리소스별로 적용되는 한도를 증가시키려는 경우, AWS는 해당 리전 내 모든 리소스에 대한 한도를 증가시킵니다. 예를 들어 VPC별 보안 그룹에 대한 제한이 해당 리전 내 모든 VPC에 적용됩니다.

## 항목

- [VPC 및 서브넷](#) (p. 224)
- [탄력적 IP 주소 \(IPv4\)](#) (p. 225)
- [흐름 로그](#) (p. 225)
- [게이트웨이](#) (p. 225)
- [네트워크 ACL](#) (p. 226)
- [네트워크 인터페이스](#) (p. 226)
- [라우팅 테이블](#) (p. 226)
- [보안 그룹](#) (p. 227)
- [VPC 피어링 연결](#) (p. 228)
- [VPC 엔드포인트](#) (p. 228)
- [VPN 연결](#) (p. 228)

## VPC 및 서브넷

Resource	기본 한도	의견
리전당 VPC	5	이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오. 리전당 인터넷 게이트웨이의 한도는 이 한도와 직접적으로 연관됩니다. 이 한도를 늘리면 리전당 인터넷 게이트웨이에 대한 한도도 같은 수량 만큼 늘어납니다.

Resource	기본 한도	의견
VPC당 서브넷	200	이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오.

## 탄력적 IP 주소 (IPv4)

Resource	기본 한도	의견
각 리전의 탄력적 IP 주소	5	한 리전에서 할당할 수 있는 VPC 탄력적 IP 주소의 수에 대한 한도입니다. Amazon EC2 탄력적 IP 주소 한도와 별개의 한도입니다. 이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오.

## 흐름 로그

Resource	기본 한도	의견
한 리전에서 단일 네트워크 인터페이스, 단일 서브넷 또는 단일 VPC당 흐름 로그	2	해당 네트워크 인터페이스가 상주하는 서브넷에 대해 흐름 로그를 2개 만들고, VPC에 대해 2개 만들 경우 네트워크 인터페이스당 6개의 흐름 로그가 효과적입니다. 이 한도는 늘릴 수 없습니다.

## 게이트웨이

Resource	기본 한도	의견
리전당 고객 게이트웨이	50	이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오.
리전당 외부 전용 인터넷 게이트웨이	5	이 한도는 리전당 VPC에 대한 한도와 직접적으로 연관됩니다. 이 한도는 개별적으로 늘릴 수 없습니다. 이 한도를 높이는 유일한 방법은 리전당 VPC에 대한 한도를 늘리는 것입니다. 하나의 VPC에는 한 번에 한 개의 외부 전용 인터넷 게이트웨이만 연결할 수 있습니다.
리전당 인터넷 게이트웨이	5	이 한도는 리전당 VPC에 대한 한도와 직접적으로 연관됩니다. 이 한도는 개별적으로 늘릴 수 없습니다. 이 한도를 높이는 유일한 방법은 리전당 VPC에 대한 한도를 늘리는 것입니다. 하나의 VPC에는 한 번에 한 개의 인터넷 게이트웨이만 추가할 수 있습니다.
가용 영역당 NAT 게이트웨이	5	이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오. pending, active 또는 deleting 상태인 NAT 게이트웨이는 이 한도에 포함됩니다.

Resource	기본 한도	의견
리전당 가상 프라이빗 게이트웨이	5	이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오. 하지만, 하나의 VPC에는 한 번에 한 개의 가상 프라이빗 게이트웨이만 연결할 수 있습니다.

## 네트워크 ACL

Resource	기본 한도	의견
VPC당 네트워크 ACL	200	하나의 VPC에 있는 한 개 이상의 서브넷에 한 개의 네트워크 ACL을 연결할 수 있습니다. 이 한도는 네트워크 ACL당 규칙 수와 같지 않습니다.
네트워크 ACL당 규칙	20	<p>단일 네트워크 ACL에 대한 편도 한도이며, 수신 규칙에 대한 한도는 20이고 발신 규칙에 대한 한도는 20입니다. 이 제한에는 IPv4 및 IPv6 규칙 둘 다 포함되며, 기본 거부 규칙도 포함됩니다(규칙 번호는 IPv4의 경우 32767, IPv6의 경우 32768, Amazon VPC 콘솔에서는 별표 *).</p> <p>이 한도는 요청에 따라 최대 40개까지 늘릴 수 있지만, 추가 규칙을 처리하기 위해 워크로드가 증가하기 때문에 네트워크 성능에 영향을 줄 수 있습니다.</p>

## 네트워크 인터페이스

Resource	기본 한도	의견
인스턴스당 네트워크 인터페이스	-	이 한도는 인스턴스 유형에 따라 다릅니다. 자세한 내용은 <a href="#">인스턴스 유형별 ENI당 프라이빗 IP 주소 단원을 참조</a> 하십시오.
리전당 네트워크 인터페이스	350	이 제한은 기본 제한(350) 또는 온디맨드 인스턴스 제한에 5를 곱한 값보다 큼니다. 온디맨드 인스턴스의 기본 한도는 20입니다. 온디맨드 인스턴스 제한이 70 미만인 경우 350의 기본 제한이 적용됩니다. AWS Support에 문의하거나 온디맨드 인스턴스 제한을 늘려 리전당 네트워크 인터페이스 개수를 늘릴 수 있습니다.

## 라우팅 테이블

Resource	기본 한도	의견
VPC당 라우팅 테이블	200	기본 라우팅 테이블 포함. 하나의 VPC에 있는 한 개 이상의 서브넷에 한 개의 라우팅 테



Resource	기본 한도	의견
		이블을 연결할 수 있습니다. 이 한도를 늘려야 하는 경우 <b>요청을 제출</b> 하십시오.
라우팅 테이블에 따른 경로(전파되지 않는 경로)	50	라우팅 테이블당 전파되지 않는 항목 수에 대한 한도입니다. <b>요청을 제출</b> 하여 최대 100개까지 늘릴 수 있지만 네트워크 성능에 영향을 줄 수 있습니다. 이 제한은 IPv4 경로 및 IPv6 경로에 대해 별도로 강제됩니다(각각 50개씩, 최대 100개 보유 가능).
라우팅 테이블에 따라 BGP를 통해 알려지는 경로(전파되는 경로)	100	라우팅 테이블당 최대 100개의 전파된 경로가 있을 수 있습니다. 이 한도는 늘릴 수 없습니다. 100개가 넘는 경로 접두사가 필요할 경우 기본 경로를 공급하십시오.

## 보안 그룹

Resource	기본 한도	의견
VPC당 보안 그룹(각 리전별)	500	이 한도를 늘려야 하는 경우 <b>요청을 제출</b> 할 수 있습니다.
보안 그룹별 인바운드 또는 아웃바운드 규칙	50	보안 그룹별로 50개의 인바운드 규칙과 50개의 아웃바운드 규칙을 지정할 수 있습니다(인바운드 규칙과 아웃바운드 규칙을 합하여 총 100개). 이 한도를 늘리거나 줄여야 하는 경우 AWS Support에 문의할 수 있습니다. 한도 변경은 인바운드 규칙과 아웃바운드 규칙 모두에 적용됩니다. 하지만 보안 그룹당 인바운드 또는 아웃바운드 규칙에 대한 한도와 네트워크 인터페이스당 보안 그룹에 대한 한도의 곱은 250을 초과할 수 없습니다. 예를 들어, 100개로 한도를 늘리려는 경우 네트워크 인터페이스당 보안 그룹의 수는 2로 줄어듭니다.  이 제한은 IPv4 및 IPv6 규칙에 별도로 강제됩니다. 예를 들어 보안 그룹은 IPv4 트래픽의 경우 인바운드 규칙 50개, IPv6 트래픽의 경우 인바운드 규칙 50개를 보유할 수 있습니다. 보안 그룹을 참조하는 규칙은 IPv4와 IPv6에 대해 각각 한 개의 규칙으로 계수됩니다.
네트워크 인터페이스당 보안 그룹	5	이 한도를 늘리거나 줄여야 하는 경우 AWS Support에 문의할 수 있습니다. 최대 한도는 16입니다. 네트워크 인터페이스당 보안 그룹에 대한 한도와 보안 그룹당 규칙에 대한 한도의 곱은 250을 초과할 수 없습니다. 예를 들어 보안 그룹당 규칙 수를 10개로 하려는 경우 네트워크 인터페이스당 보안 그룹의 수는 25개로 줄어듭니다.

## VPC 피어링 연결

Resource	기본 한도	의견
VPC당 활성 VPC 피어링 연결	50	이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오. 최대 한도는 VPC당 125개의 피어링 연결입니다. 라우팅 테이블당 항목 수는 이에 따라 늘어납니다. 하지만 네트워크 성능에 영향을 줄 수 있습니다.
대기중 VPC 피어링 연결 요청	25	해당 계정에서 요청한 대기중 VPC 피어링 연결 요청 수에 대한 한도입니다. 이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오.
수락되지 않은 VPC 피어링 연결 요청에 대한 만료 시간	1주(168시간)	이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오.

## VPC 엔드포인트

Resource	기본 한도	의견
리전당 VPC 엔드포인트	20	이 한도를 늘려야 하는 경우 AWS Support에 문의하십시오. 리전당 엔드포인트 한도에 관계없이 VPC당 엔드포인트의 최대 한도는 255개입니다.

## VPN 연결

Resource	기본 한도	의견
리전당 VPN 연결	50	이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오.
VPC당 VPN 연결(가상 프라이빗 게이트웨이당)	10	이 한도를 늘려야 하는 경우 <a href="#">요청을 제출</a> 하십시오.

# 문서 기록

다음 표에서는 이 Amazon VPC 가이드의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

기능	API 버전	설명	릴리스 날짜
IPv6 지원	2016-11-15	IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 VPC의 리소스에 할당할 수 있습니다. 자세한 내용은 <a href="#">VPC의 IP 주소 지정 (p. 76)</a> 단원을 참조하십시오.	2016년 12월 1일
비 RFC 1918 IP 주소 범위에 대한 DNS 확인 지원		이제 Amazon DNS 서버는 모든 주소 공간에서 프라이빗 DNS 호스트 이름을 프라이빗 IP 주소로 확인할 수 있습니다. 자세한 내용은 <a href="#">VPC와 함께 DNS 사용 (p. 194)</a> 단원을 참조하십시오.	2016년 10월 24일
VPC 피어링에 대한 DNS 확인 지원	2016-04-01	피어 VPC의 인스턴스가 쿼리를 보낼 때 로컬 VPC가 퍼블릭 DNS 호스트 이름을 프라이빗 IP 주소로 확인하도록 설정할 수 있습니다. 자세한 내용은 <a href="#">Amazon VPC Peering Guide에서 VPC 피어링 연결 수정</a> 을 참조하십시오.	2016년 7월 28일
무효 보안 그룹 규칙	2015-10-01	보안 그룹이 피어 VPC의 보안 그룹 규칙에서 참조되는지 여부를 식별할 수 있으며, 무효 보안 그룹 규칙을 식별할 수 있습니다. 자세한 내용은 <a href="#">Amazon VPC Peering Guide의 Working With Stale Security Groups</a> 단원을 참조하십시오.	2016년 12월 5일
VPC 피어링 연결을 통한 ClassicLink 사용	2015-10-01	연결된 로컬 EC2-Classic 인스턴스가 피어 VPC의 인스턴스와 통신하거나 반대로 통신하도록 VPC 피어링 연결을 수정할 수 있습니다. 자세한 내용은 <a href="#">Amazon VPC Peering Guide의 ClassicLink로 구성</a> 을 참조하십시오.	2016년 4월 26일
NAT 게이트웨이	2015-10-01	퍼블릭 서브넷에서 NAT 게이트웨이를 만들고 프라이빗 서브넷에서 인스턴스를 활성화하여 인터넷 또는 다른 AWS 서비스로 아웃바운드 트래픽을 시작할 수 있습니다. 자세한 내용은 <a href="#">NAT 게이트웨이 (p. 173)</a> 단원을 참조하십시오.	2015년 12월 17일
VPC 흐름 로그	2015-04-15	흐름 로그를 생성하여 VPC의 네트워크 인터페이스에서 전송하고 수신하는 IP 트래픽에 대한 정보	2015년 6월 10일

기능	API 버전	설명	릴리스 날짜
		를 캡처할 수 있습니다. 자세한 내용은 <a href="#">VPC 흐름 로그 (p. 144)</a> 단원을 참조하십시오.	
VPC 엔드포인트	2015-03-01	엔드포인트를 사용하면 인터넷, VPN 연결, NAT 인스턴스 또는 AWS Direct Connect를 통해 액세스하지 않고도 VPC와 다른 AWS 서비스 간에 프라이빗 연결을 생성할 수 있습니다. 자세한 내용은 <a href="#">VPC 엔드포인트 (p. 200)</a> 단원을 참조하십시오.	2015년 5월 11일
	2014-10-01	를 사용하면 EC2-Classic 인스턴스를 해당 계정의 VPC에 연결할 수 있습니다. VPC 보안 그룹을 EC2-Classic 인스턴스에 연결할 수 있으므로 EC2-Classic 인스턴스와 VPC의 인스턴스가 프라이빗 IP 주소를 사용하여 서로 통신할 수 있습니다. 자세한 내용은 <a href="#">(p. 211)</a> 단원을 참조하십시오.	2015년 1월 7일
프라이빗 호스팅 영역 사용	2014-09-01	Amazon Route 53의 프라이빗 호스팅 영역에서 정의한 사용자 지정 DNS 도메인 이름을 사용하여 VPC의 리소스에 액세스할 수 있습니다. 자세한 내용은 <a href="#">프라이빗 호스팅 영역 사용 (p. 197)</a> 단원을 참조하십시오.	2014년 11월 5일
서브넷의 퍼블릭 IP 주소 지정 속성 변경	2014-06-15	사용자 서브넷의 퍼블릭 IP 주소 지정 속성을 변경하여 해당 서브넷에서 시작한 인스턴스가 퍼블릭 IP 주소를 받을지 여부를 지정할 수 있습니다. 자세한 내용은 <a href="#">서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 (p. 80)</a> 단원을 참조하십시오.	2014년 6월 21일
VPC 피어링	2014-02-01	두 VPC 간에 VPC 피어링 연결을 생성하여, 각 VPC의 인스턴스가 프라이빗 IP 주소를 사용하여 마치 같은 VPC에 있는 것처럼 서로 통신할 수 있습니다. 자세한 내용은 <a href="#">VPC 피어링 (p. 197)</a> 단원을 참조하십시오.	2014년 3월 24일
새로운 EC2 시작 마법사	2013-10-01	재설계된 EC2 시작 마법사에 대한 정보가 추가되었습니다. 자세한 내용은 <a href="#">3단계: VPC에서 인스턴스 시작 (p. 10)</a> 단원을 참조하십시오.	2013년 10월 10일
퍼블릭 IP 주소 배정	2013-07-15	VPC에서 시작되는 인스턴스에 대한 새로운 IP 주소 지정 기능 관련 정보가 추가되었습니다. 자세한 내용은 <a href="#">인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 80)</a> 단원을 참조하십시오.	2013년 8월 20일
DNS 호스트 이름 활성화 및 DNS 해석 비활성화	2013-02-01	기본적으로 DNS 변환은 활성화됩니다. Amazon VPC 콘솔, Amazon EC2 명령줄 인터페이스 또는 Amazon EC2 API 작업을 사용하여 DNS 변환을 비활성화할 수 있습니다.  기본이 아닌 VPC에 대해서는 기본적으로 DNS 호스트 이름이 비활성화됩니다. Amazon VPC 콘솔, Amazon EC2 명령줄 인터페이스 또는 Amazon EC2 API 작업을 사용하여 DNS 호스트 이름을 활성화할 수 있습니다.  자세한 내용은 <a href="#">VPC와 함께 DNS 사용 (p. 194)</a> 단원을 참조하십시오.	2013년 3월 11일

기능	API 버전	설명	릴리스 날짜
고정 라우팅 구성을 사용하는 VPN 연결	2012-08-15	고정 라우팅 구성을 사용하여 Amazon VPC에 IPsec VPN 연결을 생성할 수 있습니다. 이전에는 VPN 연결 시 BGP(Border Gateway Protocol)를 사용해야 했습니다. Amazon은 이제 두 개의 연결 유형을 모두 지원하며, Cisco ASA와 Microsoft Windows Server 2008 R2를 비롯하여 BGP를 지원하지 않는 디바이스에서도 연결이 가능하다는 기쁜 소식을 전해 드립니다.	2012년 9월 13일
자동 라우팅 전파	2012-08-15	VPN 및 Direct Connect 링크의 라우팅을 해당 VPC 라우팅 테이블에 자동으로 전파하도록 구성할 수 있게 되었습니다. 이 기능은 Amazon VPC에 대한 연결을 생성하고 유지하는 작업을 간편하게 해줍니다.	2012년 9월 13일
AWS VPN CloudHub 및 중복 VPN 연결		VPC가 있든 없든 사이트 간에 안전하게 통신할 수 있습니다. 중복 VPN 연결을 사용하여 VPC에 내결함성이 있는 연결을 제공할 수 있습니다.	2011년 9월 29일
어디서나 사용 가능한 VPC	2011-07-15	AWS 리전 5개, 여러 가용 영역의 VPC, AWS 계정당 여러 VPC, VPC당 여러 VPN 연결, Microsoft Windows Server 2008 R2 및 Microsoft SQL Server 예약 인스턴스를 지원합니다.	2011년 8월 03일
전용 인스턴스	2011-02-28	전용 인스턴스는 단일 고객에게 배정된 하드웨어를 실행하는 VPC 내에서 시작되는 Amazon EC2 인스턴스입니다. 전용 인스턴스에서는 Amazon VPC 및 AWS 탄력적인 프로비저닝, 종량 과금제, 격리된 프라이빗 가상 네트워크의 장점을 모두 활용하는 동시에 하드웨어 수준에서 인스턴스를 격리할 수 있습니다.	2011년 3월 27일

## AWS Glossary

---

### Blank

---

placeholder

This page redirects to the AWS Glossary in the AWS General Reference.