
Amazon Virtual Private Cloud

사용 설명서



Amazon Virtual Private Cloud: 사용 설명서

Copyright © 2019 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	1
기본 VPC와 기본이 아닌 VPC	2
인터넷 액세스	2
회사 또는 홈 네트워크에 액세스	4
AWS PrivateLink를 통한 서비스 액세스	5
처음으로 Amazon VPC 사용하기	6
Amazon VPC에 액세스	7
Amazon VPC 가격	7
Amazon VPC 제한	7
PCI DSS 준수	7
시작하기	8
IPv4 시작하기	8
1단계: VPC 생성	9
2단계: 보안 그룹 만들기	11
3단계: VPC에서 인스턴스 시작	13
4단계: 인스턴스에 엘라스틱 IP 주소 할당	14
5단계: 정리	16
IPv6 시작하기	16
1단계: VPC 생성	17
2단계: 보안 그룹 만들기	19
3단계: 인스턴스 시작	20
시나리오 및 예시	23
시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC	23
개요	24
라우팅	25
보안	26
시나리오 1 구현	28
시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC(NAT)	30
개요	31
라우팅	33
보안	35
시나리오 2 구현	38
NAT 인스턴스를 사용하여 시나리오 2 구현	41
시나리오 3: 퍼블릭 및 프라이빗 서브넷과 액세스를 포함하는 VPC	42
개요	43
라우팅	45
보안	47
시나리오 3 구현	50
시나리오 4: 프라이빗 서브넷만 있고 액세스를 제공하는 VPC	54
개요	55
라우팅	56
보안	57
시나리오 4 구현	57
예: AWS CLI를 사용하여 IPv4 VPC 및 서브넷 생성	60
1단계: VPC와 서브넷 만들기	60
2단계: 서브넷을 퍼블릭으로 만들기	61
3단계: 서브넷에서 인스턴스 시작	63
4단계: 정리	64
예: AWS CLI를 사용하여 IPv6 VPC 및 서브넷 생성	65
1단계: VPC와 서브넷 만들기	65
2단계: 퍼블릭 서브넷 구성	66

3단계: 외부 전용 프라이빗 서브넷 구성	68
4단계: 서브넷의 IPv6 주소 지정 동작 변경	69
5단계: 퍼블릭 서브넷에서 인스턴스 시작	69
6단계: 프라이빗 서브넷으로 인스턴스를 시작	71
7단계: 정리	72
예제: 퍼블릭 서브넷과 프라이빗 서브넷 공유	73
예제: AWS PrivateLink 및 리전 간 VPC 피어링을 사용하여 리전 간에 분산된 서비스	74
예제: 서비스 공급자가 리전에 분산하도록 서비스 구성	75
예제: 서비스 소비자가 리전 간 액세스 구성	75
VPC 및 서브넷	77
VPC 및 서브넷 기본 사항	77
VPC 및 서브넷 크기	80
IPv4의 경우, VPC 및 서브넷 크기 조정	80
VPC에 IPv4 CIDR 블록 추가	81
IPv6의 경우, VPC 및 서브넷 크기 조정	84
서브넷 라우팅	84
서브넷 보안	85
로컬 네트워크 및 기타 VPC와의 연결	85
VPC 및 서브넷 관련 작업	86
VPC 만들기	86
VPC에서 서브넷 만들기	87
VPC에 보조 IPv4 CIDR 블록 연결	88
IPv6 CIDR 블록을 VPC와 연결	88
IPv6 CIDR 블록을 서브넷에 연결	89
서브넷에서 인스턴스 시작	89
서브넷 삭제	90
VPC에서 IPv4 CIDR 블록의 연결을 해제	90
VPC 또는 서브넷에 연결된 IPv6 CIDR 블록을 분리	91
VPC 삭제	91
공유된 VPC에 대한 작업	92
공유 VPC 사전 조건	92
서브넷 공유	93
공유된 서브넷의 공유 해제	93
공유 서브넷의 소유자 식별	94
공유 서브넷 권한	94
소유자 및 참여자에 대한 청구 및 측정	94
공유 서브넷에 대해 지원되지 않는 서비스	94
제한	95
기본 VPC 및 기본 서브넷	96
기본 VPC 구성 요소	96
기본 서브넷	97
가용성 및 지원되는 플랫폼	98
지원되는 플랫폼 및 기본 VPC 보유 여부 확인	98
기본 VPC와 기본 서브넷 보기	99
기본 VPC로 EC2 인스턴스 시작	99
콘솔을 사용하여 EC2 인스턴스 시작	100
명령줄을 사용하여 EC2 인스턴스 시작	100
기본 서브넷과 기본 VPC 삭제	100
기본 VPC 만들기	100
기본 서브넷 생성	101
IP 주소 지정	103
프라이빗 IPv4 주소	104
퍼블릭 IPv4 주소	104
IPv6 주소	105
서브넷에 대한 IP 주소 지정 동작	106
IP 주소 작업	106
서브넷의 퍼블릭 IPv4 주소 지정 속성 수정	106

서브넷의 퍼블릭 IPv6 주소 지정 속성 수정	106
인스턴스 시작 시 퍼블릭 IPv4 주소 배정	107
인스턴스 시작 시 IPv6 주소 배정	108
인스턴스에 IPv6 주소 할당	108
인스턴스에 할당된 IPv6 주소 해제	109
API 및 명령 개요	109
IPv6로 마이그레이션하기	110
예: 퍼블릭 및 프라이빗 서브넷이 있는 VPC에서 IPv6 사용	111
1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결	113
2단계: 라우팅 테이블 업데이트	114
3단계: 보안 그룹 규칙 업데이트	114
4단계: 인스턴스 유형 변경	115
5단계: 인스턴스에 IPv6 주소 할당	116
6단계: (선택 사항) 인스턴스에서 IPv6 구성하기	116
보안	122
보안 그룹 및 네트워크 ACL 비교	122
보안 그룹	123
보안 그룹 기본 사항	124
VPC의 기본 보안 그룹	124
보안 그룹 규칙	125
EC2-Classic 및 EC2-VPC용 보안 그룹의 차이점	127
보안 그룹 작업	127
네트워크 ACL	130
네트워크 ACL 기본 사항	131
네트워크 ACL 규칙	131
기본 네트워크 ACL	132
사용자 지정 네트워크 ACL	133
취발성 포트	137
네트워크 ACL 작업	137
예: 서브넷 내 인스턴스에 대한 액세스 제어	140
API 및 명령 개요	143
VPC에 권장되는 네트워크 ACL 규칙	144
시나리오 1을 위한 권장 규칙	144
시나리오 2를 위한 권장 규칙	147
시나리오 3을 위한 권장 규칙	153
시나리오 4를 위한 권장 규칙	158
액세스 제어	159
AWS CLI 또는 SDK용 예제 정책	160
콘솔용 예제 정책	167
VPC 흐름 로그	173
흐름 로그 기본 사항	174
흐름 로그 레코드	174
흐름 로그 제한	177
CloudWatch Logs에 게시	177
Amazon S3에 게시	180
흐름 로그 작업	184
문제 해결	187
VPC 네트워킹 구성 요소	189
네트워크 인터페이스	189
라우팅 테이블	190
라우팅 테이블 기본 사항	190
라우팅 우선순위	193
라우팅 옵션	194
라우팅 테이블 작업	197
API 및 명령 개요	200
인터넷 게이트웨이	202
인터넷 액세스 활성화	202

인터넷 게이트웨이로 VPC 생성	204
외부 전용 인터넷 게이트웨이	207
외부 전용 인터넷 게이트웨이 기본 사항	208
외부 전용 인터넷 게이트웨이 작업	209
API 및 CLI 개요	210
NAT	210
NAT 게이트웨이	211
NAT 인스턴스	226
NAT 인스턴스 및 NAT 게이트웨이 비교	233
DHCP 옵션 세트	234
DHCP 옵션 세트 개요	235
Amazon DNS 서버	236
DHCP 옵션 변경	236
DHCP 옵션 세트를 사용한 작업	237
API 및 명령 개요	238
DNS	239
DNS 호스트 이름	239
VPC에서 DNS 지원	239
DNS 제한	241
EC2 인스턴스의 DNS 호스트 이름 보기	241
VPC에 대한 DNS 지원 업데이트	242
프라이빗 호스팅 영역 사용	242
VPC 피어링	243
탄력적 IP 주소	243
탄력적 IP 주소 기본 사항	243
탄력적 IP 주소 작업	244
API 및 CLI 개요	246
VPC 엔드포인트	246
인터페이스 엔드포인트	247
게이트웨이 엔드포인트	258
VPC 엔드포인트로 서비스 액세스 제어	271
VPC 엔드포인트 삭제	272
VPC 엔드포인트 서비스	272
개요	273
엔드포인트 서비스 제한	275
VPC 엔드포인트 서비스 구성 생성	275
엔드포인트 서비스에 대한 권한 추가 및 제거	277
Network Load Balancer 변경 및 설정 수락	278
인터페이스 엔드포인트 연결 요청 수락 및 거부	279
엔드포인트 서비스에 대한 알림 생성 및 관리	280
연결 정보에 대한 프록시 프로토콜 사용	282
엔드포인트 서비스 구성 삭제	282
ClassicLink	282
VPN 연결	284
제한	285
VPC 및 서브넷	285
DNS	285
탄력적 IP 주소(IPv4)	285
흐름 로그	285
게이트웨이	286
네트워크 ACL	286
네트워크 인터페이스	287
라우팅 테이블	287
보안 그룹	287
VPC 피어링 연결	288
VPC 엔드포인트	288
연결	288

VPC 공유	289
문서 기록	290

Amazon VPC란 무엇인가?

Amazon Virtual Private Cloud(Amazon VPC)에서는 사용자가 정의한 가상 네트워크로 AWS 리소스를 시작할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다.

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 사용하는 경우, Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EC2는 무엇입니까?](#)를 참조하여 개요를 파악하십시오.

내용

- [VPC 및 서브넷 \(p. 1\)](#)
- [지원되는 플랫폼 \(p. 1\)](#)
- [기본 VPC와 기본이 아닌 VPC \(p. 2\)](#)
- [인터넷 액세스 \(p. 2\)](#)
- [회사 또는 홈 네트워크에 액세스 \(p. 4\)](#)
- [AWS PrivateLink를 통한 서비스 액세스 \(p. 5\)](#)

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. IP 주소 범위와 VPC 범위를 설정하고 서브넷을 추가하고 보안 그룹을 연결한 다음 라우팅 테이블을 구성합니다.

서브넷은 VPC의 IP 주소 범위입니다. 지정된 서브넷으로 AWS 리소스를 시작할 수 있습니다. 인터넷에 연결되어야 하는 리소스에는 퍼블릭 서브넷을 사용하고, 인터넷에 연결되지 않는 리소스에는 프라이빗 서브넷을 사용합니다. 퍼블릭 서브넷과 프라이빗 서브넷에 대한 자세한 정보는 [VPC 및 서브넷 기본 사항 \(p. 77\)](#)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 정보는 [보안 \(p. 122\)](#) 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 정보는 [지원되는 플랫폼 및 기본 VPC 보유 여부 확인 \(p. 98\)](#) 단원을 참조하십시오.

EC2-Classic 대신 VPC에서 인스턴스를 시작하면 다음의 장점이 있습니다.

- 인스턴스의 시작/종지에 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 inbound 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

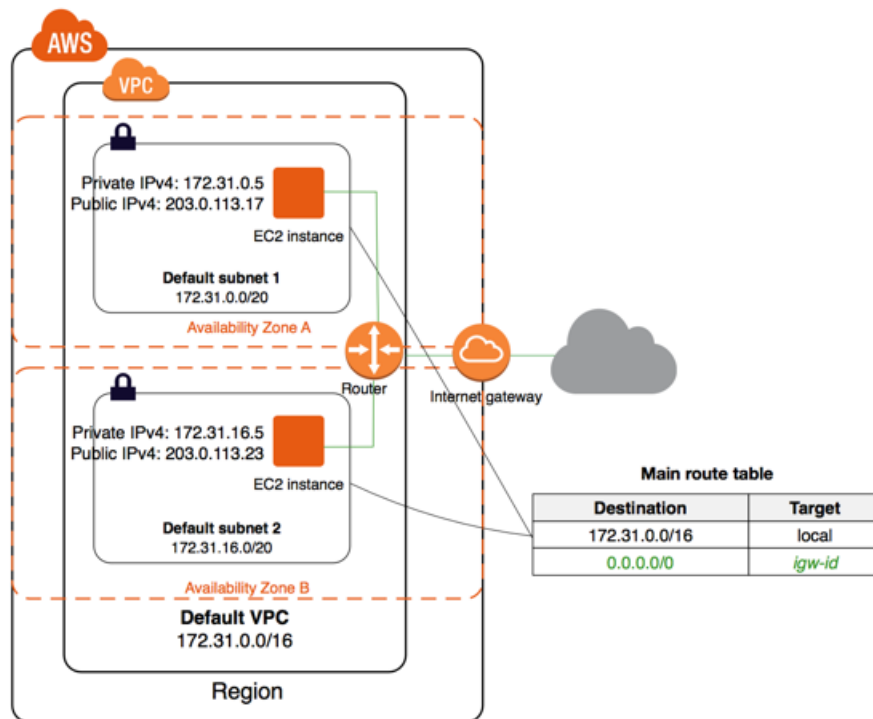
EC2-VPC 플랫폼만 지원하는 계정일 경우 각 가용 영역에 기본 서브넷이 있는 기본 VPC가 제공됩니다. 기본 VPC는 EC2-VPC가 제공하는 고급 기능의 장점을 가지고 있으며 바로 사용할 수 있게 설정되어 있습니다. 기본 VPC가 있고 서브넷을 지정하지 않을 경우 인스턴스를 시작할 때 기본 VPC로 인스턴스가 시작됩니다. Amazon VPC에 대한 지식이 전혀 없어도 기본 VPC로 인스턴스를 시작할 수 있습니다.

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.

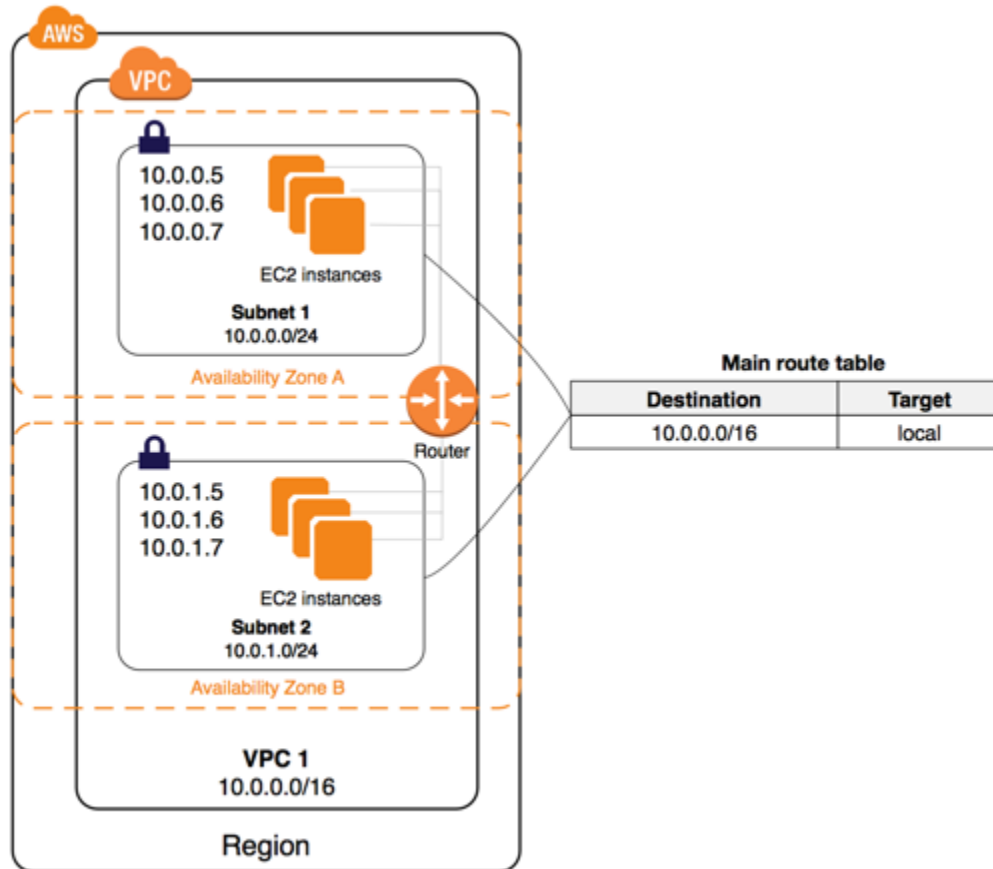
인터넷 액세스

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

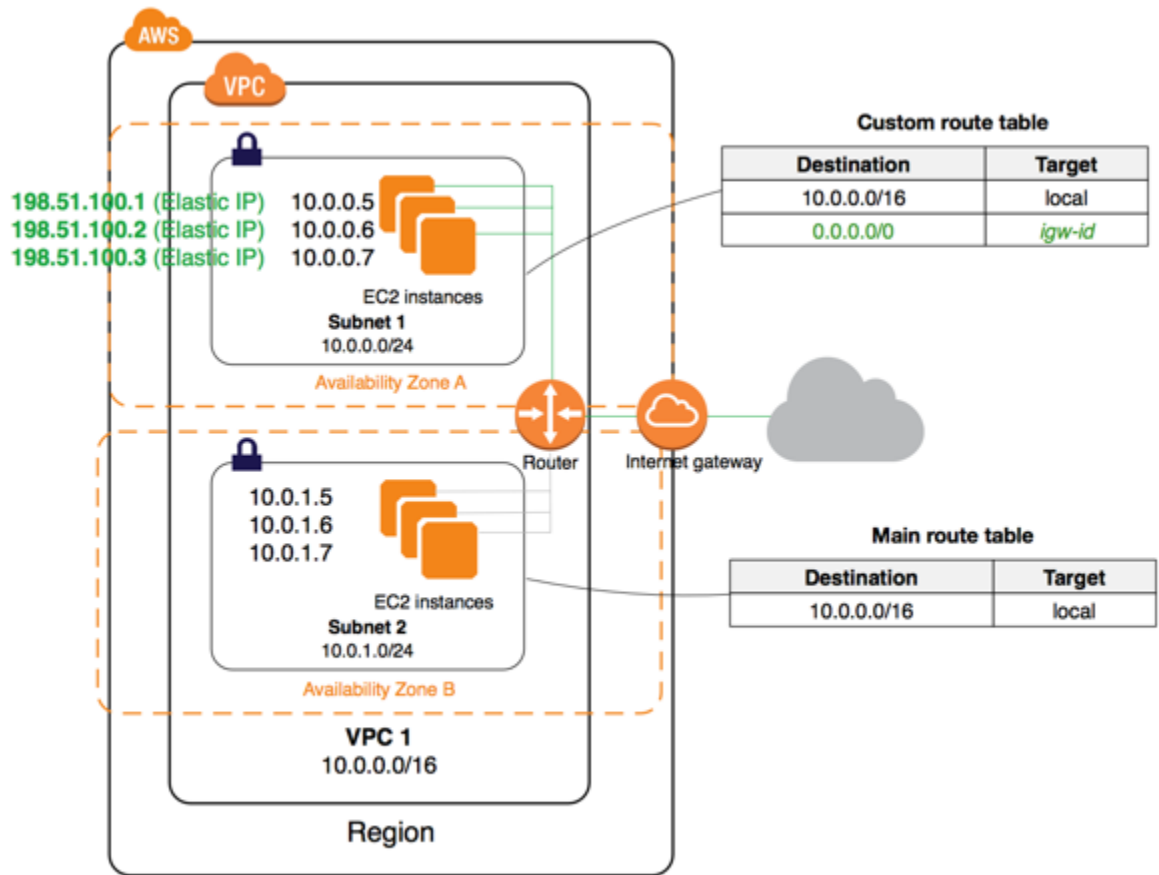
기본 VPC에는 인터넷 게이트웨이가 포함되며, 각각의 기본 서브넷은 퍼블릭 서브넷입니다. 기본 서브넷에서 시작한 각 인스턴스에는 프라이빗 IPv4 주소와 퍼블릭 IPv4 주소가 있습니다. 이러한 인스턴스는 인터넷 게이트웨이를 통해 인터넷과 통신할 수 있습니다. 인터넷 게이트웨이를 통해 인스턴스는 Amazon EC2 네트워크 엣지를 통해 인터넷에 연결할 수 있습니다.



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.

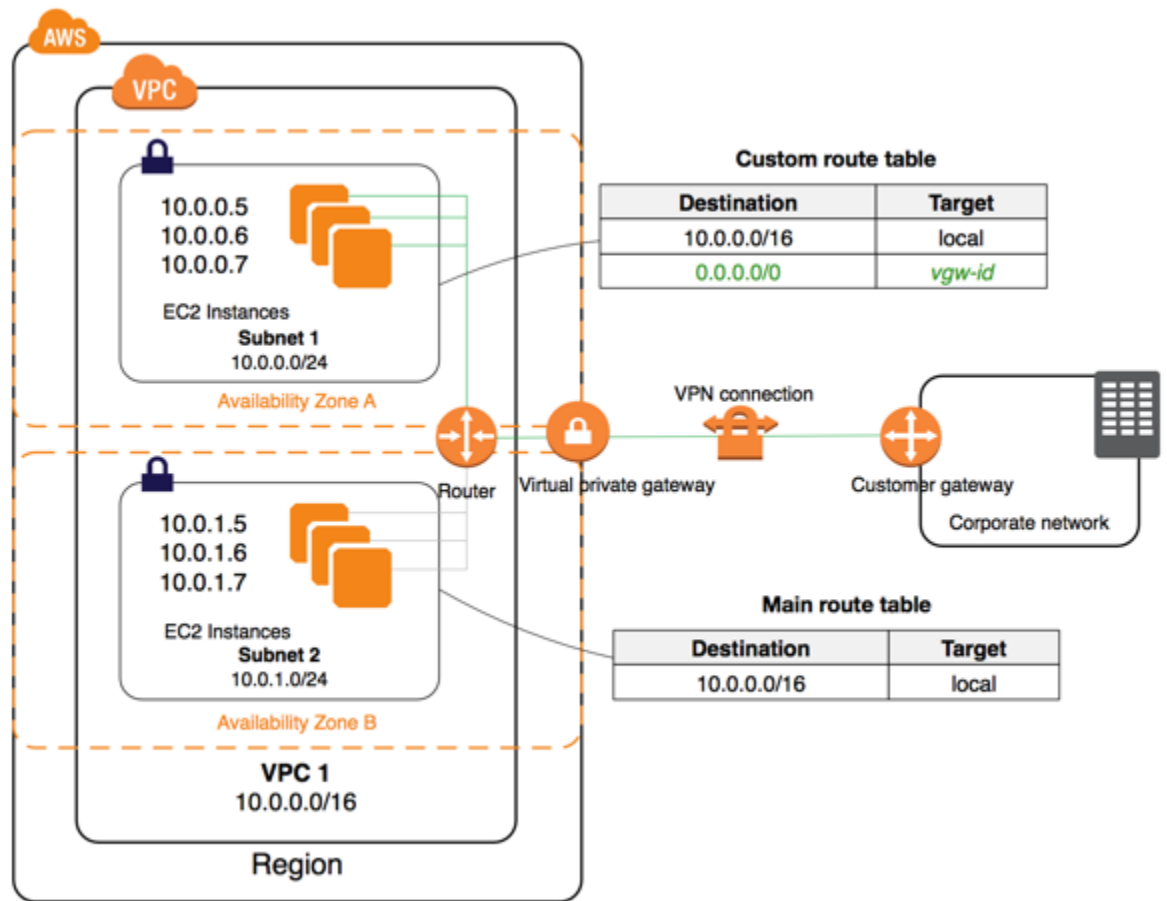
자세한 정보는 [NAT \(p. 210\)](#) 단원을 참조하십시오.

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 정보는 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 단원을 참조하십시오. IPv6 트래픽은 IPv4 트래픽에서 분리되어 있으므로, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다.

회사 또는 홈 네트워크에 액세스

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

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

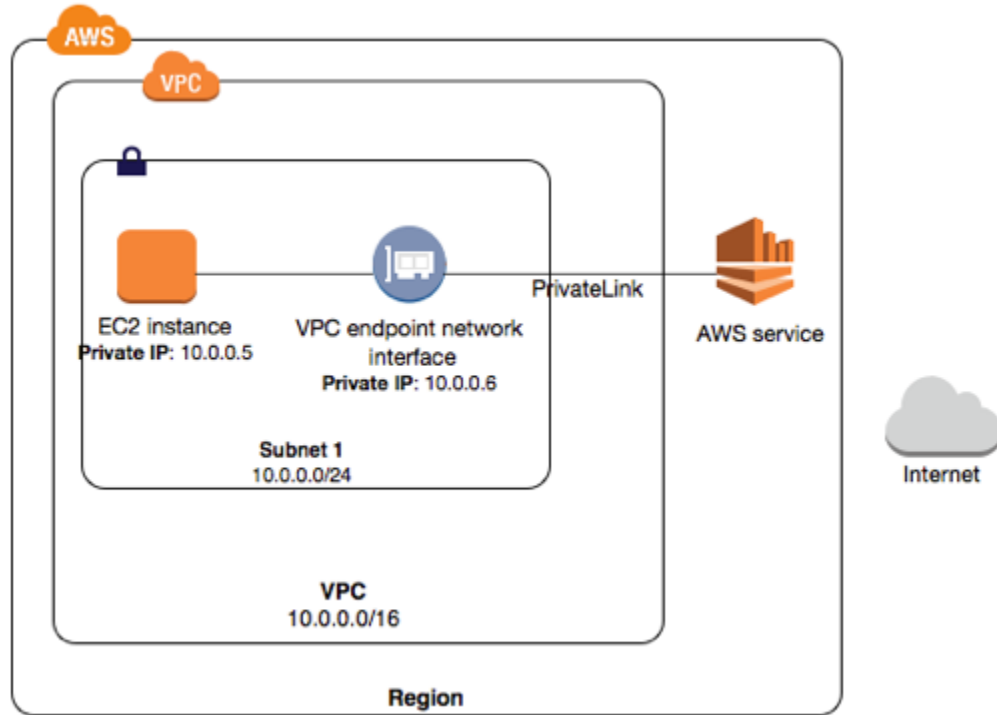


자세한 정보는 사용 설명서의 [란 무엇입니까?](#) 단원을 참조하십시오.

AWS PrivateLink를 통한 서비스 액세스

AWS PrivateLink는 지원되는 AWS 서비스, 기타 AWS 계정에서 호스팅된 서비스(VPC 엔드포인트 서비스) 및 지원 AWS Marketplace 파트너 서비스에 VPC를 비공개로 연결할 수 있도록 하는 가용성과 확장성이 높은 기술입니다. 서비스와 통신하는 데 인터넷 게이트웨이, NAT 디바이스, 퍼블릭 IP 주소, AWS Direct Connect 연결 또는 연결이 필요하지 않습니다. VPC와 서비스 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

AWS PrivateLink를 사용하려면 VPC에 서비스에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 서비스로 전달되는 트래픽에 대한 진입점 역할을 하는 프라이빗 IP 주소를 포함하여 서브넷에 탄력적 네트워크 인터페이스를 생성합니다. 자세한 정보는 [VPC 엔드포인트 \(p. 246\)](#) 단원을 참조하십시오.



자체 AWS PrivateLink 구동 서비스(엔드포인트 서비스)를 생성하고 다른 AWS 고객이 서비스에 액세스할 수 있도록 할 수 있습니다. 자세한 정보는 [VPC 엔드포인트 서비스\(AWS PrivateLink\)](#) (p. 272) 단원을 참조하십시오.

처음으로 Amazon VPC 사용하기

Amazon VPC에 대해 실습해 보려면 [Amazon VPC 시작하기](#) (p. 8)을 완료하십시오. 이 연습은 퍼블릭 서브넷에 기본이 아닌 VPC를 만들고, 귀사의 서브넷에서 인스턴스를 시작하는 절차를 안내합니다.

기본 VPC가 있고 VPC에 대해 추가 구성을 하지 않고 VPC에서 인스턴스를 시작하려면 [기본 VPC로 EC2 인스턴스 시작](#) (p. 99)을 참조하십시오.

Amazon VPC의 기본 시나리오에 대한 자세한 정보는 [시나리오 및 예시](#) (p. 23)을 참조하십시오. VPC와 서브넷을 필요에 맞게 다른 방식으로 구성할 수 있습니다.

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

리소스	설명
Amazon Virtual Private Cloud(VPC) 연결 오류	네트워크 연결 옵션에 대한 개요를 제공합니다.
Amazon VPC forum	Amazon VPC 항목과 관련된 기술적 질문에 대해 토론할 수 있는 커뮤니티 기반 포럼입니다.
AWS 개발자 리소스	설명서, 코드 샘플, 릴리스 정보 및 AWS를 사용하여 혁신적인 애플리케이션을 생성하는 데 도움이 되는 기타 정보를 한 자리에서 찾을 수 있습니다.
AWS 지원 센터	AWS Support 홈 페이지입니다.

리소스	설명
문의처	AWS 결제, 계정 및 이벤트와 관련된 문의를 처리하는 중앙 문의처입니다.

Amazon VPC에 액세스

Amazon VPC는 웹 기반 사용자 인터페이스인 Amazon VPC 콘솔을 제공합니다. AWS 계정에 가입한 고객은 AWS Management 콘솔에 로그인한 뒤 VPC를 선택하여 Amazon VPC 콘솔에 액세스할 수 있습니다.

명령줄 인터페이스를 선호하는 고객의 경우 다음과 같은 옵션이 있습니다.

AWS Command Line Interface (AWS CLI)

다양한 AWS 서비스에서 사용되는 명령어를 제공하며 Windows, macOS, Linux/Unix를 지원합니다. 시작하려면 [AWS Command Line Interface 사용 설명서](#)를 참조하십시오. Amazon VPC의 명령에 대한 자세한 정보는 [ec2](#)를 참조하십시오.

Windows PowerShell용 AWS 도구

PowerShell 환경에서 스크립트 작업을 하는 고객을 위해 다양한 AWS 서비스에 대한 명령을 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. 자세한 정보는 Amazon EC2 API Reference의 [작업](#)을 참조하십시오.

HTTP나 HTTPS 요청을 직접 보내는 대신, 각 언어가 제공하는 고유의 API를 사용하여 애플리케이션을 빌드하기 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 제공합니다. 자세한 정보는 [AWS SDK 및 도구](#) 섹션을 참조하십시오.

Amazon VPC 가격

Amazon VPC 사용에 따르는 추가 요금은 없습니다. 사용하는 인스턴스와 기타 Amazon EC2 기능에 대한 표준 요금만 지불하면 됩니다. 연결 및 NAT 게이트웨이를 사용하면 요금이 부과됩니다. 자세한 정보는 [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 정보는 [Amazon VPC 제한 \(p. 285\)](#) 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.

Amazon VPC 시작하기

다음 사용지침서를 참고하시어 기본값이 아닌 VPC를 빠르게 설정하십시오. VPC의 리소스가 IPv6를 통해 통신하도록 하고 싶다면 연결된 IPv6 CIDR 블록이 있는 VPC를 설정할 수 있습니다. 또는 VPC를 연결된 Pv4 CIDR 블록과 함께 설정하십시오.

기본 VPC가 이미 있으면 새로운 VPC를 만들거나 새롭게 환경을 설정할 필요 없이 기본 VPC에서 인스턴스를 열어 시작하실 수 있습니다. 자세한 내용은 [기본 VPC로 EC2 인스턴스 시작 \(p. 99\)](#) 단원을 참조하십시오.

자습서

- [Amazon VPC용 IPv4 시작하기 \(p. 8\)](#)
- [Amazon VPC용 IPv6 시작하기 \(p. 16\)](#)

Amazon VPC용 IPv4 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 엘라스틱 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 <https://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 \(p. 1\)](#) 단원을 참조하십시오.

작업

- 1단계: VPC 생성 (p. 9)
- 2단계: 보안 그룹 만들기 (p. 11)
- 3단계: VPC에서 인스턴스 시작 (p. 13)

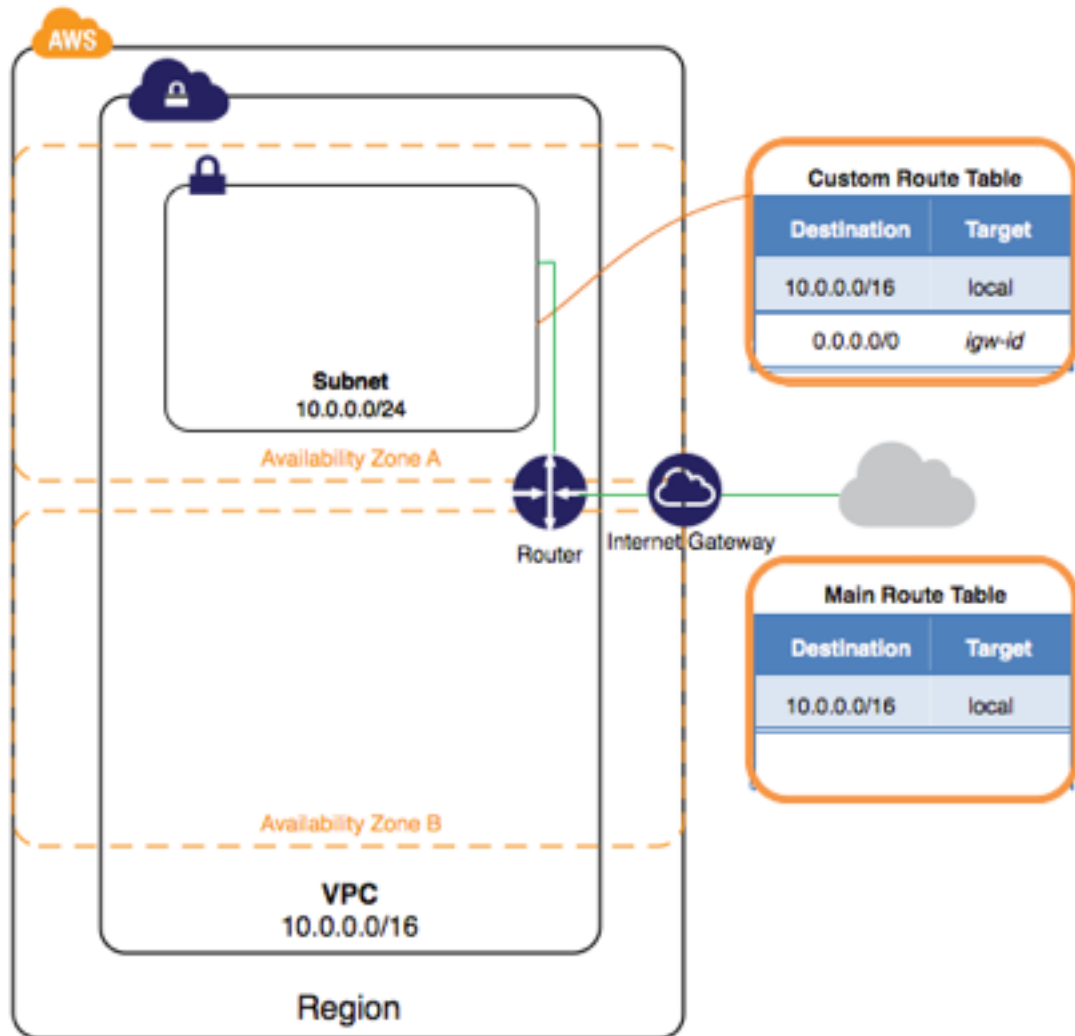
- 4단계: 인스턴스에 엘라스틱 IP 주소 할당 (p. 14)
- 5단계: 정리 (p. 16)

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](#) 단원을 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](#) 단원을 참조하십시오.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.

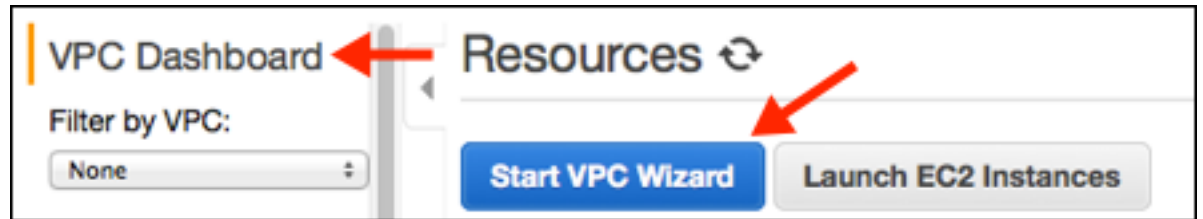


Note

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. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [리전 및 가용 영역](#) 단원을 참조하십시오.
3. 탐색 창에서 VPC 대시보드를 선택합니다. 대시보드에서 Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



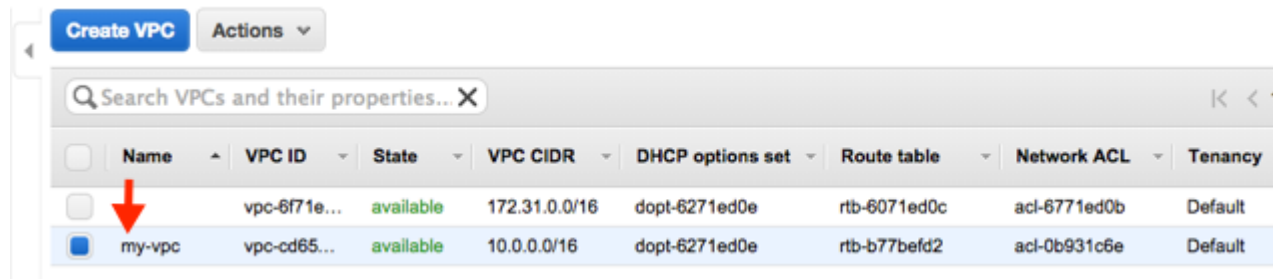
Note

탐색 창에서 VPCs를 선택하지 마십시오. VPC 만들기를 사용하여 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-vpc를 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.
 - 서비스 엔드포인트 단원에서 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](#) 단원을 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 테넌시에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [전용 인스턴스](#) 단원을 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 WebServerSG를 생성하고, VPC에서 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.

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			
대상 주소 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 WebServerSG를 입력하고 설명을 제공합니다. 필요에 따라 [Name tag] 필드를 사용하여 키가 Name인 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] 열에서 이름을 볼 수 있습니다.
7. [Inbound Rules] 탭에서 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
 - a. [Type] 목록에서 HTTP를 선택한 다음, 0.0.0.0/0을 [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 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 0.0.0.0/0을 사용할 수 있습니다.

Important

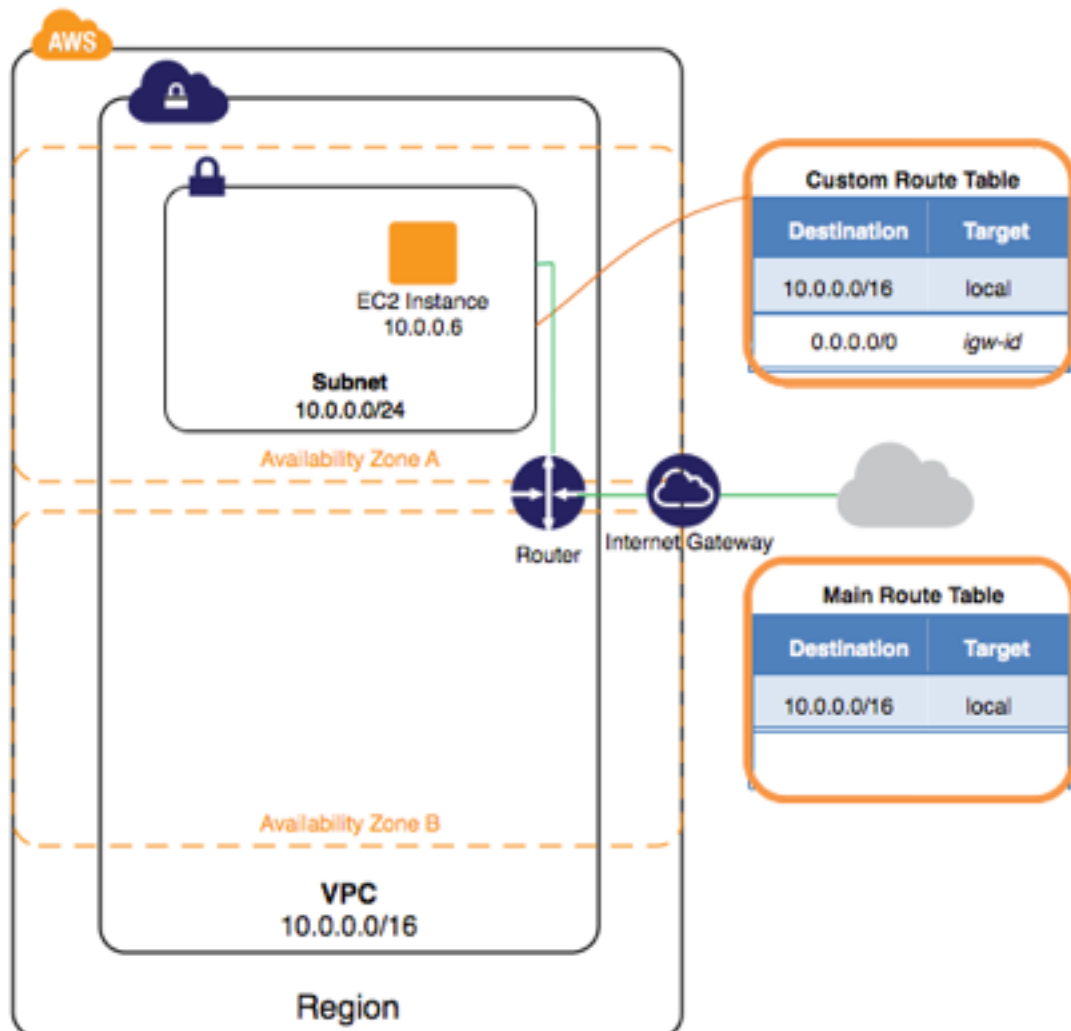
0.0.0.0/0을 사용하는 경우 모든 IP 주소에서 SSH 또는 RDP를 사용하여 인스턴스에 액세스할 수 있습니다. 따라서 연습에서는 잠시 사용해도 되지만 프로덕션 환경에서 사용하는 것은 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하십시오.

- d. Save를 선택합니다.

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.

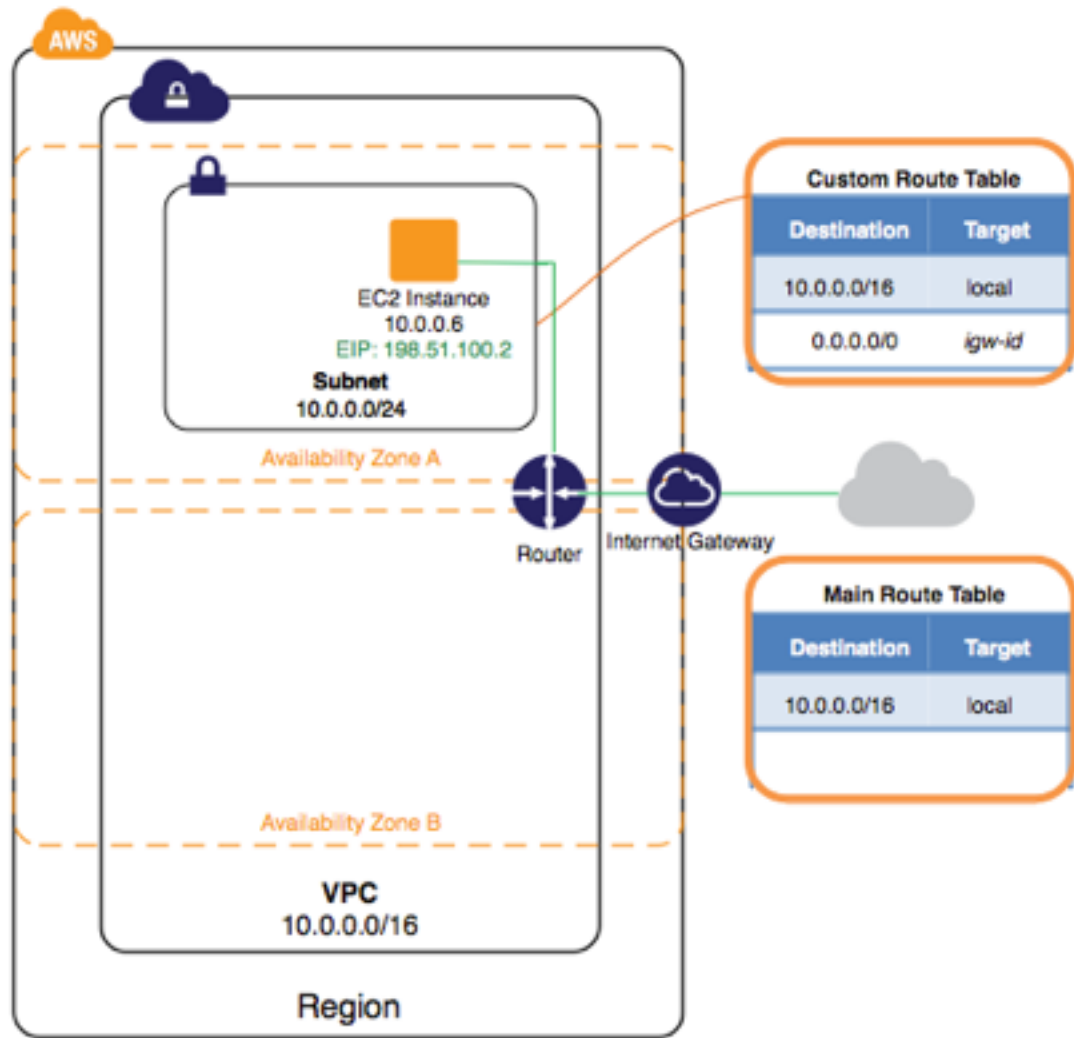
2. 오른쪽 상단의 탐색 모음에서 VPC 및 보안 그룹을 생성했던 리전과 동일한 리전을 선택합니다.
 3. 대시보드에서 [Launch Instance]를 선택합니다.
 4. 마법사의 첫 페이지에서, 사용하려는 AMI를 선택합니다. 이 연습에서는 Amazon Linux AMI 또는 Windows AMI를 선택하는 것이 좋습니다.
 5. [Choose an Instance Type] 페이지에서는 시작할 인스턴스의 하드웨어 구성과 크기를 선택할 수 있습니다. 마법사는 사용자가 선택한 AMI를 기반으로 하여 첫 번째로 사용 가능한 인스턴스 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 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.
 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 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 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 엘라스틱 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.



엘라스틱 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. 새 주소 할당을 선택한 다음 할당을 선택합니다.

Note

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

4. 목록에서 엘라스틱 IP 주소와 [Actions], [Associate Address]를 차례로 선택합니다.
5. 리소스 유형에서 인스턴스를 선택했는지 확인합니다. 인스턴스 목록에서 인스턴스를 선택합니다. 선택했으면 연결을 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 엘라스틱 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 엘라스틱 IP 주소를 릴리스하고 VPC를 삭제하십시오.

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택하고 [Actions]를 선택한 후 [Instance State]와 [Terminate]를 차례로 선택합니다.
4. 대화 상자에서 [Release attached Elastic IPs] 단원을 확장하고 엘라스틱 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 VPCs]를 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 정보는 [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 <https://aws.amazon.com/>으로 이동한 후 무료 계정 생성을 선택합니다.

작업

- 1단계: VPC 생성 (p. 17)
- 2단계: 보안 그룹 만들기 (p. 19)

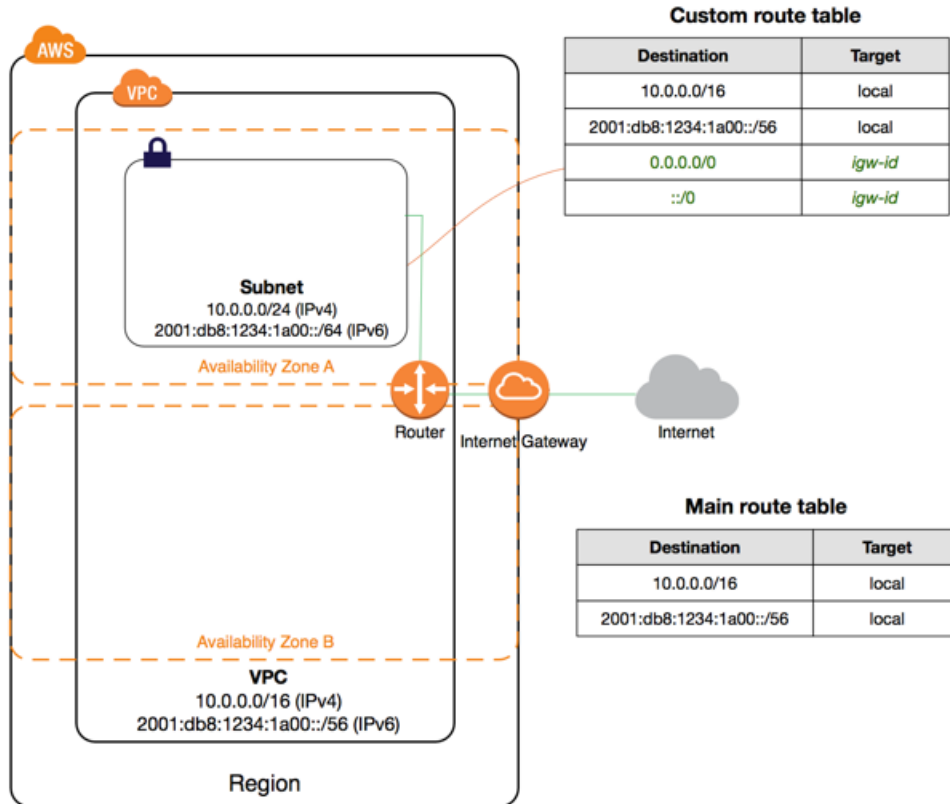
- 3단계: 인스턴스 시작 (p. 20)

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 정보는 [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 정보는 [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 정보는 [라우팅 테이블](#) 단원을 참조하십시오.

다음 다이어그램은 이 단계를 마친 후의 VPC 아키텍처를 보여 줍니다.



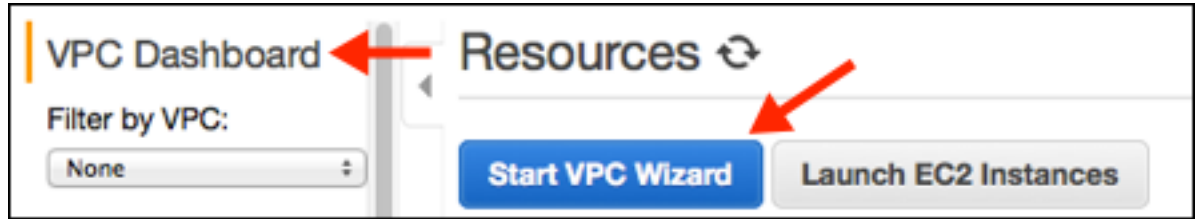
Note

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

Amazon VPC 마법사를 사용하여 VPC를 생성하려면

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

- 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. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [리전 및 가용 영역](#) 단원을 참조하십시오.
- 탐색 창에서 VPC 대시보드를 선택한 다음, Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



Note

탐색 창에서 VPCs를 선택하지 마십시오. VPC 만들기를 사용하여 VPC 마법사에 액세스할 수 없습니다.

- 첫 번째 옵션인 VPC with a Single Public Subnet을 선택한 후 Select를 선택합니다.
- 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.
- IPv4 CIDR block에 대해서는 기본 설정(10.0.0.0/16)을 그대로 두거나 원하는 대로 설정할 수 있습니다. 자세한 정보는 [VPC 및 서브넷 크기](#) 단원을 참조하십시오.

IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다.

- 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 16 진수 페어 값을 그대로 둘 수 있습니다(00).
- 페이지에서 기본 구성의 나머지 부분은 그대로 두고 Create VPC를 선택합니다.
- 상태 창에 진행 중인 작업이 표시됩니다. 작업이 끝나면 [OK]를 선택하여 상태 창을 닫습니다.
- 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 기본적으로 기본 라우팅 테이블을 사용합니다.

VPC에 대한 정보 보기

- <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
- 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.
- 탐색 창에서 서브넷을 선택합니다. VPC를 생성할 때 생성된 서브넷이 콘솔에 표시됩니다. 서브넷을 [Name] 열에서 이름으로 식별하거나, 전 단계에서 얻은 VPC 정보를 참조하여 [VPC] 열에서 살펴볼 수 있습니다.
- 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.
- 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 WebServerSG를 생성하고, VPC에서 인스턴스를 시작할 때 이 보안 그룹을 지정합니다.

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			
대상 주소 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 Groups]와 [Create Security Group]을 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 [Edit]를 선택한 후, 다음과 같이 인바운드 트래픽에 대한 규칙들을 추가합니다.
 - a. Type에서 HTTP를 선택한 다음, `::/0`를 Source 필드에 입력합니다.
 - b. Add another rule을 선택한 후 Type에서 HTTPS를 선택하고 Source 필드에 `::/0`를 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 `::/0`를 사용할 수 있습니다.

Important

`::/0`을 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.

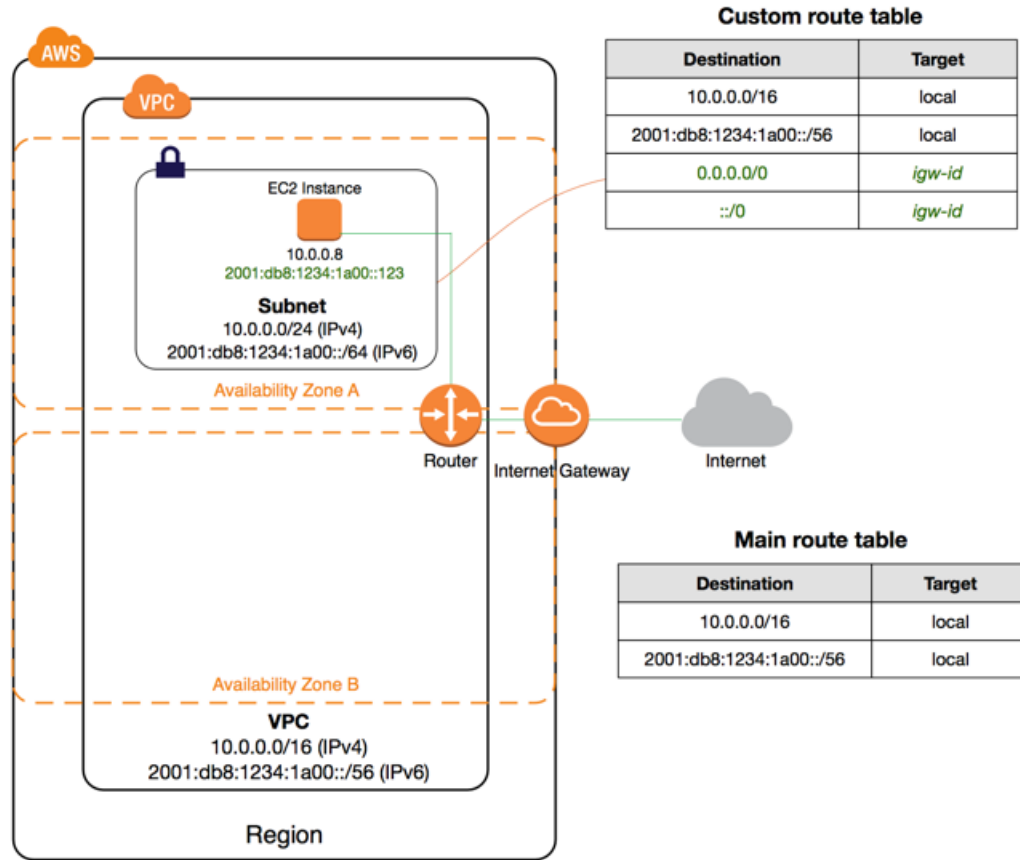
- d. Save를 선택합니다.

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.

VPC에서 EC2 인스턴스를 시작하기 전에 IPv6 IP 주소를 자동 할당하도록 VPC의 서브넷을 구성하십시오. 자세한 정보는 [the section called “서브넷의 퍼블릭 IPv6 주소 지정 속성 수정” \(p. 106\)](#) 단원을 참조하십시오.

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 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 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

홈 네트워크에서 SSH 또는 원격 데스크톱을 사용하여 IPv6 주소를 통해 인스턴스에 접속할 수 있습니다. 로컬 컴퓨터에 IPv6 주소가 있고 IPv6를 사용하도록 컴퓨터를 구성해야 합니다. Linux 인스턴스에 연결하는 방법에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [\[Connecting to Your Linux Instance\]](#)를 참조하십시오. Windows 인스턴스에 연결하는 방법에 대한 자세한 정보는 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. 자세한 정보는 [Amazon VPC 시작하기 \(p. 8\)](#) 단원을 참조하십시오.

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. 23)	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. 30)	퍼블릭 웹 애플리케이션을 실행하기 위해 VPC 마법사를 사용하여 VPC를 생성하는 한편, 두 번째 서브넷에서 공개적으로 액세스가 불가능한 백 엔드 서버는 계속 유지합니다.
시나리오 3: 퍼블릭 및 프라이빗 서브넷과 액세스를 포함하는 VPC (p. 42)	데이터 센터를 클라우드로 확장하기 위해 VPC 마법사를 사용하여 VPC를 생성하고 VPC에서 직접 인터넷에 액세스합니다.
시나리오 4: 프라이빗 서브넷만 있고 액세스를 제공하는 VPC (p. 54)	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. 60)	AWS CLI를 사용하여 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC를 만듭니다.
예: AWS CLI를 사용하여 IPv6 VPC 및 서브넷 생성 (p. 65)	AWS CLI를 사용하여 연결된 IPv6 CIDR 블록이 있는 VPC를 만들고, 연결된 IPv6 CIDR 블록이 있는 퍼블릭 서브넷과 프라이빗 서브넷을 만듭니다.
the section called “예제: 퍼블릭 서브넷과 프라이빗 서브넷 공유” (p. 73)	프라이빗 및 퍼블릭 서브넷을 계정과 공유합니다.
예제: AWS PrivateLink 및 리전 간 VPC 피어링을 사용하여 리전 간에 분산된 서비스 (p. 74)	리전 간 피어링과 AWS PrivateLink를 함께 사용하여 프라이빗 서비스에 대한 액세스를 리전의 소비자로 확장하는 방법을 알아봅니다.

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 시나리오를 IPv6—에 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 \(p. 103\)](#) 단원을 참조하십시오.

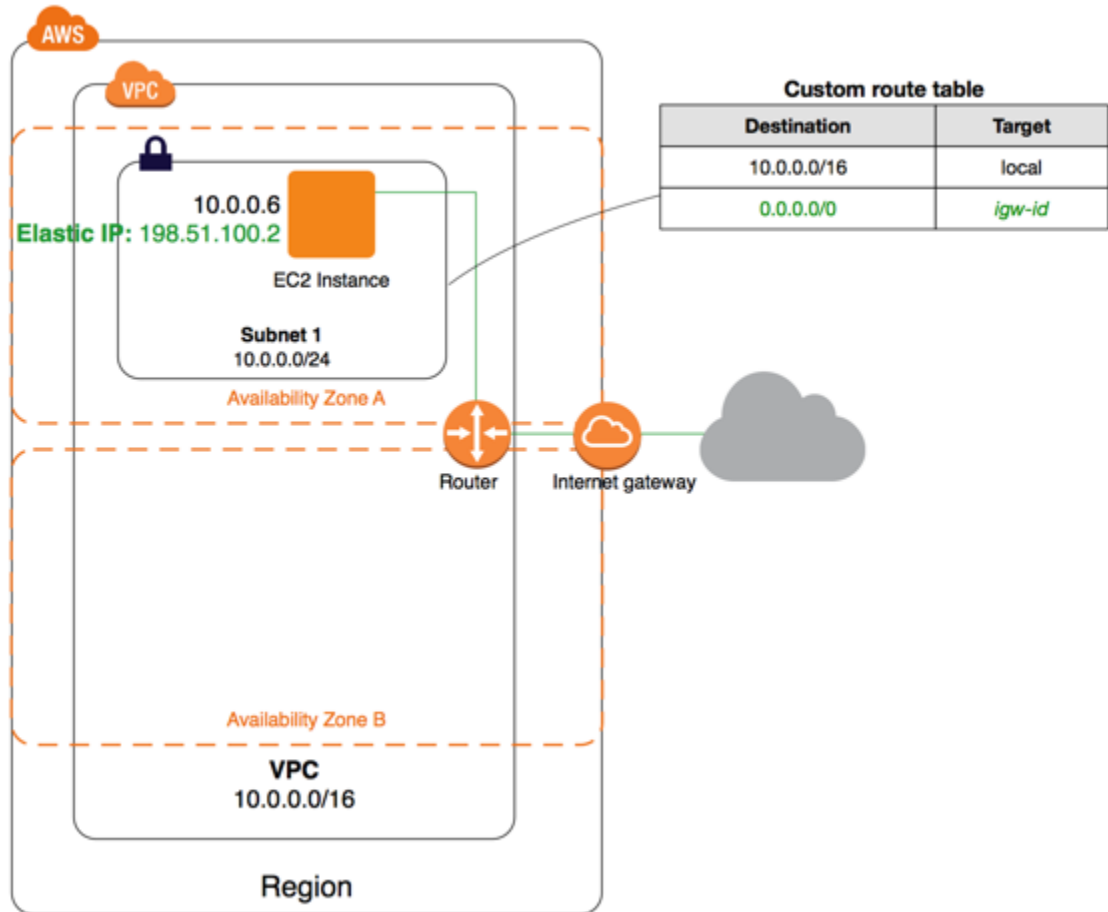
내용

- [개요 \(p. 24\)](#)
- [라우팅 \(p. 25\)](#)
- [보안 \(p. 26\)](#)

- [시나리오 1 구현 \(p. 28\)](#)

개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.



Note

[Amazon VPC 시작하기 \(p. 8\)](#)을 완료했다면 Amazon VPC 콘솔에서 VPC 마법사를 사용하여 이 시나리오를 이미 구현한 것입니다.

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

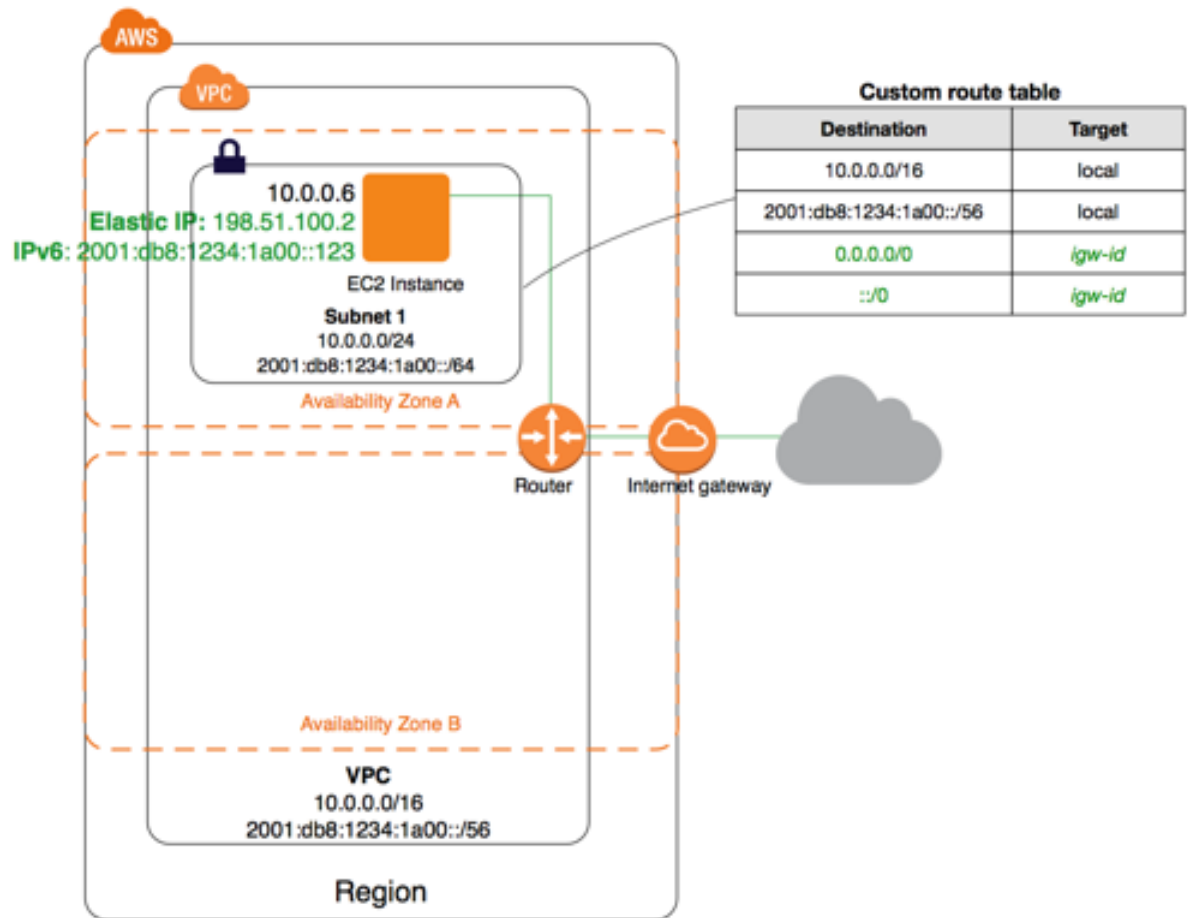
- IPv4 CIDR 블록의 크기가 /16(예: 10.0.0.0/16)인 Virtual Private Cloud(VPC). 이것은 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- IPv4 CIDR 블록의 크기가 /24(예: 10.0.0.0/24)인 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다.
- 인터넷 게이트웨이. VPC를 인터넷 및 다른 AWS 서비스에 연결합니다.
- 인스턴스가 VPC의 다른 인스턴스와 통신할 수 있게 해주는 서브넷 범위(예: 10.0.0.6)의 프라이빗 IPv4 주소와 인터넷에서 인스턴스에 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 내용은 [VPC 및 서브넷 \(p. 77\)](#) 단원을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이 \(p. 202\)](#)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.

대상 주소	대상
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.

대상 주소	대상
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 \(p. 122\)](#) 단원을 참조하십시오.

이 시나리오의 경우, 네트워크 ACL이 아니라 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 1을 위한 권장 규칙 \(p. 144\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 124\)](#)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, WebServerSG라는 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. http://checkip.amazonaws.com 또는 https://checkip.amazonaws.com 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)			
대상 주소	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 IPv4 주소에 대한 아웃바운드 액세스를 모두 허용하는 기본 규칙. 예를 들어 소프트웨어 업데이트를 받기 위해 웹 서버가 아웃바운드 트래픽을 시작하게 하려면 기본 아웃바운드 규칙을 유지합니다. 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)			
대상 주소	프로토콜	포트 범위	설명
::/0	모두	모두	모든 IPv6 주소에 대한 아웃바운드 액세스를 모두 허용하는 기본 규칙. 예를 들어 소프트웨어 업데이트를 받기 위해 웹 서버가 아웃바운드 트래픽을 시작하게 하려면 기본 아웃바운드 규칙을 유지합니다. 아니면 이 규칙을 제거할 수 있습니다.

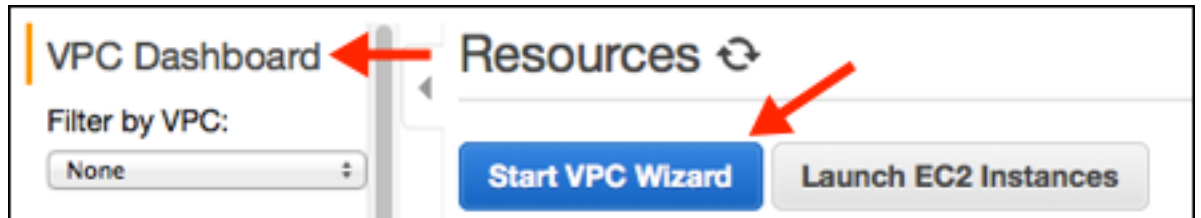
시나리오 1 구현

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 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



3. 첫 번째 옵션인 [VPC with a Single Public Subnet]을 선택한 후 [Select]를 선택합니다.
4. (선택 사항) VPC와 서브넷에 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다. VPC 및 서브넷에 고유한 IPv4 CIDR 블록 범위를 지정하거나 기본값(10.0.0.0/16 및 10.0.0.0/24)을 유지할 수 있습니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. 퍼블릭 서브넷의 IPv6 CIDR에서 사용자 지정 IPv6 CIDR을 지정합니다를 선택하고 서브넷에 16진수 페어 값을 지정하거나 기본값(00)을 유지합니다.
6. 기본 설정의 나머지 부분은 유지하고 VPC 만들기를 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 그룹의 이름과 설명을 입력합니다. 이 주제에서는 WebServerSG라는 이름이 예제로 사용됩니다. [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, 0.0.0.0/0을 [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. Save를 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 정보는 [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 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 107\)](#) 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-Classical을 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 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.

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 인터넷에 바로 아웃바운드 트래픽을 전송할 수 있는 반면, 프라이빗 서브넷의 인스턴스는 그렇게 할 수 없습니다. 반면, 프라이빗 서브넷의 인스턴스는 퍼블릭 서브넷에 있는 NAT(Network Address Translation) 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 \(p. 211\)](#) 단원을 참조하십시오.

이 시나리오를 IPv6—에 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 \(p. 103\)](#) 단원을 참조하십시오.

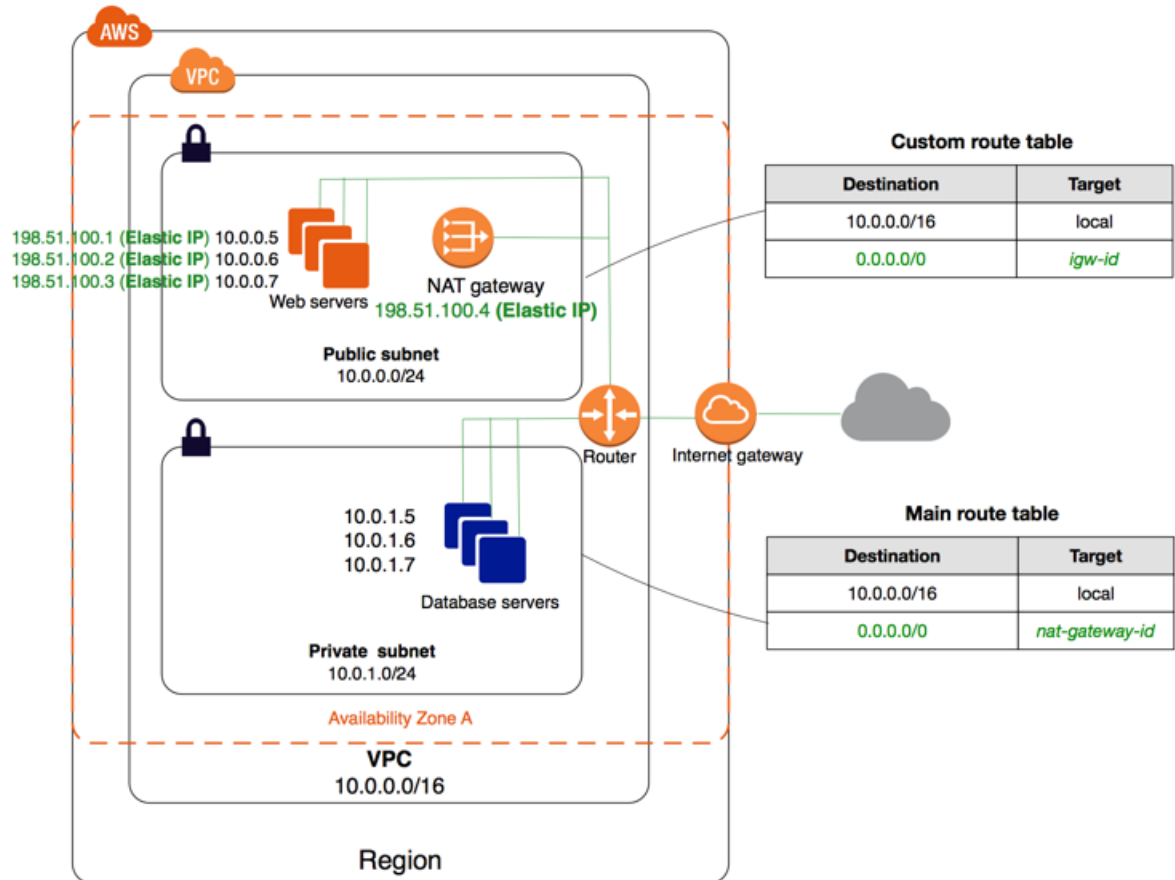
내용

- [개요 \(p. 31\)](#)

- 라우팅 (p. 33)
- 보안 (p. 35)
- 시나리오 2 구현 (p. 38)
- NAT 인스턴스를 사용하여 시나리오 2 구현 (p. 41)

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 주소

가 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).

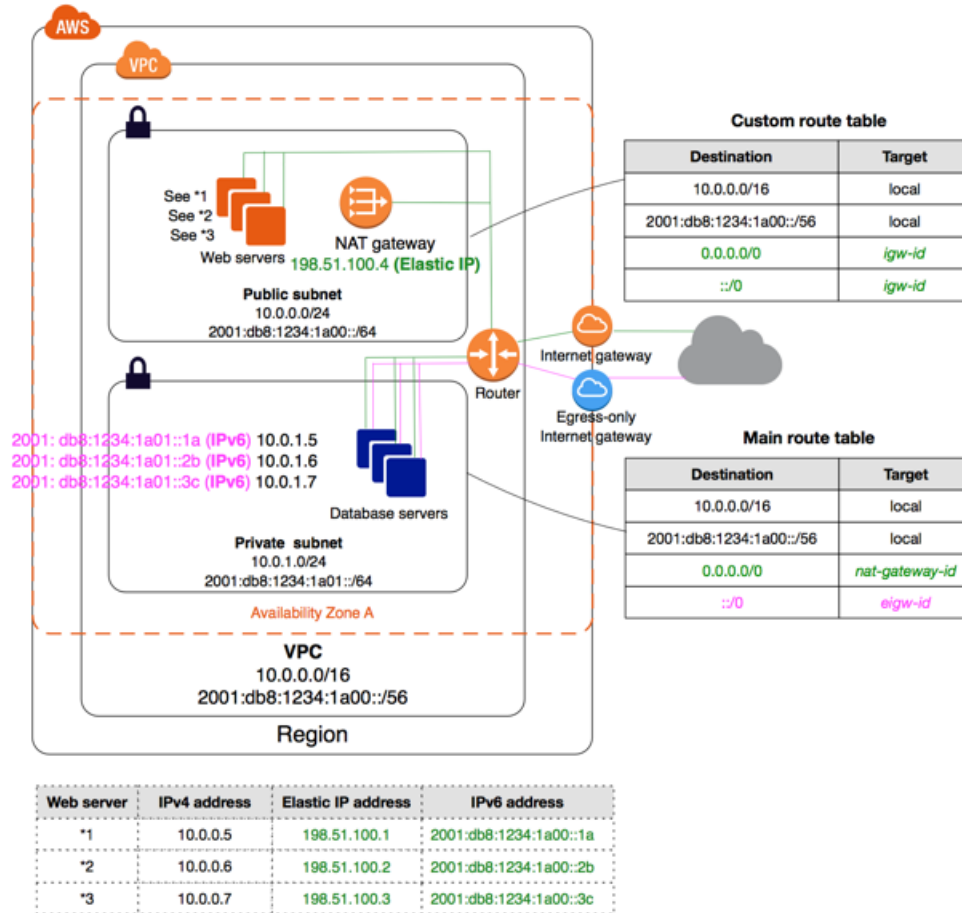
- 자체 탄력적 IPv4 주소를 가진 NAT 게이트웨이. 프라이빗 서브넷의 인스턴스는 IPv4를 통한 NAT 게이트웨이를 통해 인터넷에 요청을 보낼 수 있습니다(예: 소프트웨어 업데이트의 경우).
- 퍼블릭 서브넷과 연결된 사용자 지정 라우팅 테이블. 이 라우팅 테이블에는 서브넷의 인스턴스가 IPv4를 통해 VPC의 다른 인스턴스와 통신할 수 있게 하는 항목과, 서브넷의 인스턴스가 IPv4를 통해 인터넷과 직접 통신할 수 있게 하는 항목이 들어 있습니다.
- 프라이빗 서브넷과 연결된 기본 라우팅 테이블. 라우팅 테이블에는 서브넷의 인스턴스가 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 내용은 [VPC 및 서브넷 \(p. 77\)](#) 단원을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 내용은 [인터넷 게이트웨이 \(p. 202\)](#)를 참조하십시오. NAT 게이트웨이에 대한 자세한 내용은 [NAT 게이트웨이 \(p. 211\)](#)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 \(p. 207\)](#) 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 엘라스틱 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.

대상 주소	대상
10.0.0.0/16	로컬

대상 주소	대상
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 해 줍니다. 두 번째 항목에서는 인터넷 게이트웨이(예: igw-1a2b3d4d)를 통해 기타 서브넷 트래픽을 모두 인터넷으로 라우팅합니다.

대상 주소	대상
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 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.

대상 주소	대상
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.

대상 주소	대상
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 \(p. 122\)](#) 단원을 참조하십시오.

시나리오 2에서는 네트워크 ACL이 아닌 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 2를 위한 권장 규칙 \(p. 147\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 124\)](#)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). http://checkip.amazonaws.com 또는 https://checkip.amazonaws.com 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			
대상 주소	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
아웃바운드			
대상 주소	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	IPv4를 통한 인터넷으로의 아웃바운드 HTTP 액세스를 허용합니다(예: 소프트웨어 업데이트).
0.0.0.0/0	TCP	443	IPv4를 통한 인터넷으로의 아웃바운드 HTTPS 액세스를 허용합니다(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			
대상 주소	프로토콜	포트 범위	설명
보안 그룹의 ID	모두	모두	이 보안 그룹에 지정된 다른 인스턴스에 대해 아웃바운드 트래픽 허용

(선택 사항) 퍼블릭 서브넷에서 배스천 호스트를 시작하여 홈 네트워크에서 프라이빗 서브넷까지에 대한 SSH 또는 RDP 트래픽을 프록시로 사용하는 경우, 배스천 인스턴스 또는 연결된 보안 그룹의 인바운드 SSH 또는 RDP 트래픽을 허용하는 DBServerSG 보안 그룹에 규칙을 추가합니다.

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			
대상 주소	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			
대상 주소	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

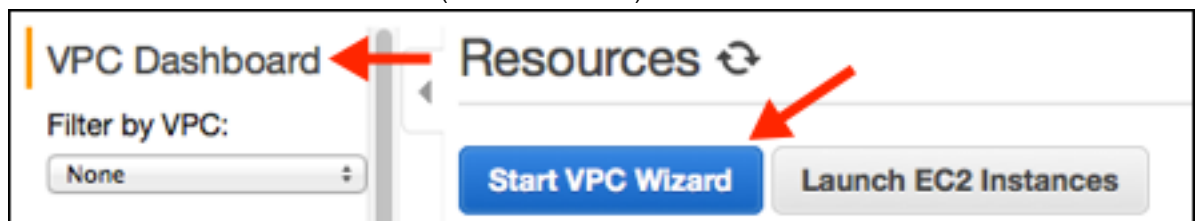
1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. Allocate new address를 선택합니다.
4. [Allocate]를 선택합니다.

Note

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

VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드에서 Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



3. 두 번째 옵션인 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 선택한 후 선택을 선택합니다.
4. (선택 사항) VPC와 서브넷에 이름을 붙이면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다. VPC 및 서브넷에 대해 고유한 IPv4 CIDR 블록 범위를 지정하거나 기본값을 유지할 수 있습니다.
5. (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. 퍼블릭 서브넷의 IPv6 CIDR에서 사용자 지정 IPv6 CIDR을 지정합니다를 선택하고 서브넷에 16진수 페어 값을 지정하거나 기본값을 유지합니다. Private subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 16진수 페어 값을 지정하거나 기본값을 유지합니다.
6. Specify the details of your NAT gateway 단원에서 계정의 탄력적 IP 주소에 할당 ID를 지정합니다.
7. 페이지에서 나머지 기본값은 유지하고 VPC 만들기를 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 그룹의 이름과 설명을 입력합니다. 이 주제에서는 WebServerSG라는 이름이 예제로 사용됩니다. [VPC]에 대해 생성한 VPC의 ID를 선택하고 [Yes, Create]를 선택합니다.
4. [Create Security Group]을 다시 선택합니다.
5. 보안 그룹의 이름과 설명을 입력합니다. 이 주제에서는 DBServerSG라는 이름이 예제로 사용됩니다. [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 선택합니다. 소스에 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. Save를 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. Save를 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 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 107\)](#) 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 정보는 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 \(p. 226\)](#)를 참조하십시오.

위와 동일한 절차를 따를 수 있지만, VPC 마법사의 NAT 단원에서 [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			
대상 주소	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 그룹의 이름과 설명을 지정합니다. 이 주제에서는 NATSG라는 이름이 예제로 사용됩니다. [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. Save를 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 Interfaces]를 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. 35) 참조)을 선택한 후 [Save]를 선택합니다.

시나리오 3: 퍼블릭 및 프라이빗 서브넷과 액세스를 포함하는 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 프런트 엔드가 있는 멀티 티어 애플리케이션을 퍼블릭 서브넷에서 실행하고, IPsec 연결을 통해 네트워크에 연결된 프라이빗 서브넷에 데이터를 보관할 수 있습니다.

이 시나리오를 IPv6—에 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

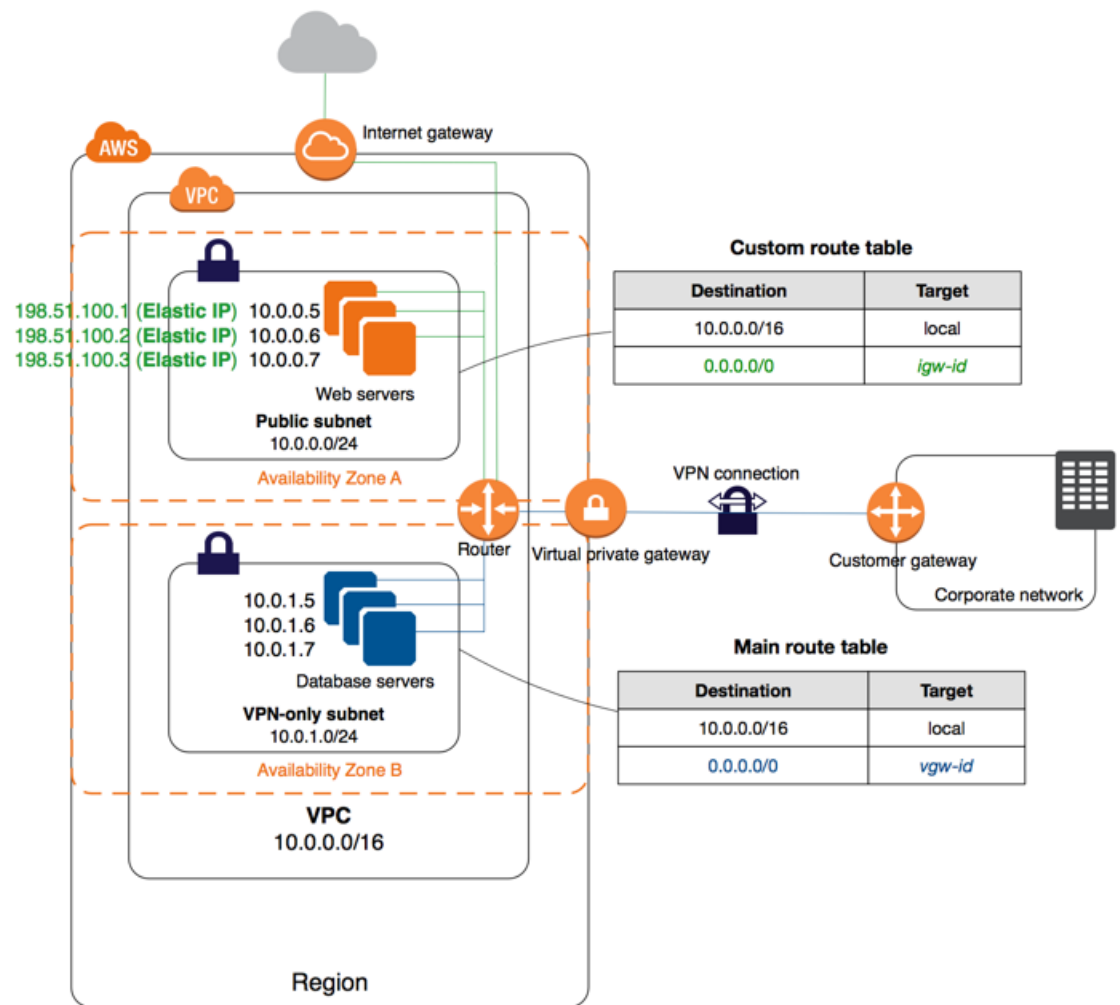
Amazon은 연결을 통한 IPv6 통신을 지원하지 않습니다. 하지만 VPC의 인스턴스들은 IPv6를 통해 서로 통신할 수 있고, 퍼블릭 서브넷의 인스턴스들은 IPv6를 통해 인터넷으로 통신할 수 있습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 정보는 [VPC의 IP 주소 지정 \(p. 103\)](#) 단원을 참조하십시오.

내용

- 개요 (p. 43)
- 라우팅 (p. 45)
- 보안 (p. 47)
- 시나리오 3 구현 (p. 50)

개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.



Important

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

이 시나리오를 위한 구성에는 다음 정보가 포함됩니다.

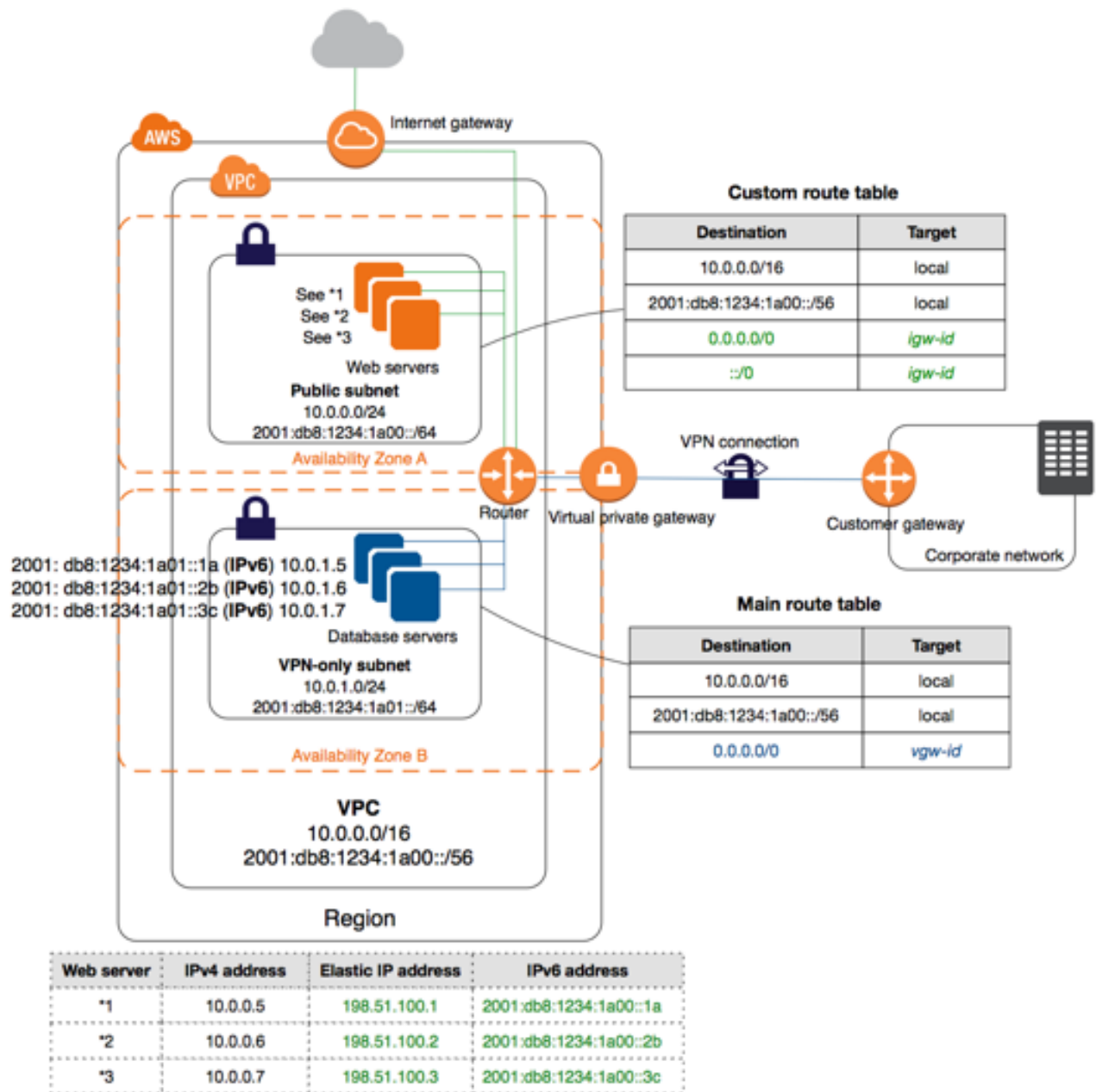
- IPv4 CIDR의 크기가 /16(예: 10.0.0.0/16)인 Virtual Private Cloud(VPC). 이것은 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- IPv4 CIDR의 크기가 /24(예: 10.0.0.0/24)인 퍼블릭 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다. 퍼블릭 서브넷은 인터넷 게이트웨이로 이어지는 경로가 있는 라우팅 테이블과 연결된 서브넷입니다.
- IPv4 CIDR의 크기가 /24(예: 10.0.1.0/24)인 VPN 전용 서브넷. 이것은 256개의 프라이빗 IPv4 주소를 제공합니다.
- 인터넷 게이트웨이. 인터넷 게이트웨이는 VPC를 인터넷 및 다른 AWS 제품에 연결합니다.
- VPC와 네트워크 간의 연결. 연결은 연결의 Amazon 측에 있는 가상 프라이빗 게이트웨이와 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 정보는 [VPC 및 서브넷 \(p. 77\)](#) 및 [VPC의 IP 주소 지정 \(p. 103\)](#)을 참조하십시오. 인터넷 게이트웨이에 대한 자세한 정보는 [인터넷 게이트웨이 \(p. 202\)](#)를 참조하십시오. 연결에 대한 자세한 정보는 사용 설명서의 [란 무엇인가?](#)를 참조하십시오. 고객 게이트웨이 구성에 대한 자세한 정보는 [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 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.

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

대상 주소	대상
10.0.0.0/16	로컬
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.

대상 주소	대상
10.0.0.0/16	로컬
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 정보는 [NAT \(p. 210\)](#)를 참조하십시오. VPC 마법사를 사용하여 NAT 디바이스를 설정하는 것에 대한 자세한 정보는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\) \(p. 30\)](#)를 참조하십시오.

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

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.

대상 주소	대상
10.0.0.0/16	로컬
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.

대상 주소	대상
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.

대상 주소	대상
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 \(p. 122\)](#) 단원을 참조하십시오.

시나리오 3의 경우, 네트워크 ACL이 아닌 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 3을 위한 권장 규칙 \(p. 153\)](#)을 참조하십시오.

VPC는 [기본 보안 그룹 \(p. 124\)](#)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 액세스를 허용합니다.
네트워크의 퍼블릭 IP 주소 범위	TCP	3389	네트워크에서 Windows 인스턴스로 인터넷 게이트웨이를 거쳐 인바운드 RDP 액세스를 허용합니다.
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			
대상 주소	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			
대상 주소	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			
대상 주소	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 연결을 생성합니다.

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

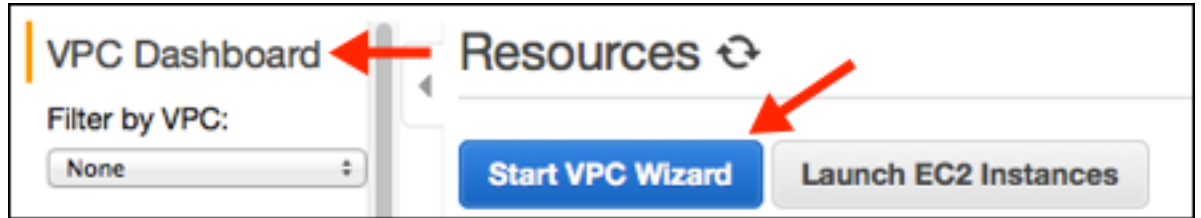
고객 게이트웨이를 준비하려면

1. 고객 게이트웨이로 사용할 디바이스를 결정합니다. 테스트를 통해 검증된 디바이스에 대한 자세한 정보는 [Amazon Virtual Private Cloud FAQ](#)를 참조하십시오. 고객 게이트웨이 요구 사항에 대한 자세한 정보는 [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 연결을 생성하고자 하는 경우, 연결을 통해 가상 프라이빗 게이트웨이로 알려야 하는 내부 IP 범위의 목록(CIDR 표기법)을 얻어야 합니다. 자세한 정보는 사용 설명서의 [라우팅 테이블 및 VPN 라우팅 정책](#)을 참조하십시오.

VPC 마법사를 사용하여 VPC를 생성하려면

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

2. 대시보드에서 Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



3. 세 번째 옵션인 퍼블릭 및 프라이빗 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC를 선택한 후 선택을 선택합니다.
4. 퍼블릭 및 프라이빗 서브넷이 있고 하드웨어 VPN 액세스를 제공하는 VPC 페이지에서 다음을 수행합니다.
- (선택 사항) 필요하다면 VPC 및 서브넷의 IPv4 CIDR 블록 범위를 수정하거나 기본값을 유지합니다.
 - (선택 사항) VPC와 서브넷의 이름을 지정합니다. 이렇게 하면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다.
 - (선택 사항, IPv6 전용) IPv6 CIDR block에 대해 Amazon-provided IPv6 CIDR block을 선택합니다. 퍼블릭 서브넷의 IPv6 CIDR에서 사용자 지정 IPv6 CIDR을 지정합니다를 선택하고 서브넷에 16진수 페어 값을 지정하거나 기본값을 유지합니다. Private subnet's IPv6 CIDR에 대해 Specify a custom IPv6 CIDR을 선택합니다. IPv6 서브넷에 16진수 페어 값을 지정하거나 기본값을 유지합니다.
 - [Next]를 선택합니다.
5. VPN 구성 페이지에서 다음을 수행합니다.
- 고객 게이트웨이 IP에서 VPN 라우터의 퍼블릭 IP 주소를 지정합니다.
 - (선택 사항) 고객 게이트웨이와 연결의 이름을 지정합니다.
 - 라우팅 유형에서 라우팅 옵션 중 하나를 선택합니다. 자세한 정보는 사용 설명서의 [라우팅 옵션](#)을 참조하십시오.
 - VPN 라우터가 BGP(Border Gateway Protocol)를 지원할 경우 [Dynamic (requires BGP)]을 선택합니다.
 - VPN 라우터에서 BGP를 지원하지 않으면 정적을 선택합니다. IP 접두사에서 CIDR 표기법으로 각각의 네트워크 IP 범위를 추가합니다.
 - VPC 만들기를 선택합니다.
6. 마법사가 완료되면 탐색 창에서 연결을 선택합니다. 마법사가 생성한 연결을 선택하고 구성 다운로드를 선택합니다. 대화 상자에서 해당되는 고객 게이트웨이 공급업체, 플랫폼, 소프트웨어 버전을 선택하고 [Yes, Download]를 선택합니다.
7. 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

- <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
- 탐색 창에서 [Security Groups]를 선택합니다.
- [Create Security Group]을 선택합니다.
- 보안 그룹의 이름과 설명을 입력합니다. 이 주제에서는 WebServerSG라는 이름이 예제로 사용됩니다. VPC에서 VPC의 ID를 선택한 다음 예, 생성을 선택합니다.

5. [Create Security Group]을 다시 선택합니다.
6. 보안 그룹의 이름과 설명을 입력합니다. 이 주제에서는 DBServerSG라는 이름이 예제로 사용됩니다. VPC에서 VPC의 ID를 선택한 다음 예, 생성을 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. 유형에서 HTTP를 선택한 다음 소스에 0.0.0.0/0을 입력합니다.
 - b. 다른 규칙 추가를 선택한 다음 유형에서 HTTPS를 선택하고 소스에 0.0.0.0/0을 입력합니다.
 - c. 다른 규칙 추가를 선택한 다음 유형에서 SSH를 선택합니다. 소스에 네트워크의 퍼블릭 IP 주소 범위를 입력합니다.
 - d. 다른 규칙 추가를 선택한 다음 유형에서 RDP를 선택합니다. 소스에 네트워크의 퍼블릭 IP 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. 소스에 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. 유형에서 MS SQL을 선택합니다. 대상 주소에 DBServerSG 보안 그룹의 ID를 입력합니다.
 - c. 다른 규칙 추가를 선택한 다음 유형에서 MySQL을 선택합니다. [Destination]에 DBServerSG 보안 그룹의 ID를 지정합니다.
 - d. 다른 규칙 추가를 선택한 다음 유형에서 HTTPS를 선택합니다. 대상에 0.0.0.0/0을 입력합니다.
 - e. 다른 규칙 추가를 선택한 다음 유형에서 HTTP를 선택합니다. 대상에 0.0.0.0/0을 입력합니다.
 - f. Save를 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. 유형에서 SSH를 선택하고 소스에 네트워크의 IP 주소 범위를 입력합니다.
 - b. 다른 규칙 추가를 선택한 다음 유형에서 RDP를 선택하고 소스에 네트워크의 IP 주소 범위를 입력합니다.
 - c. 다른 규칙 추가를 선택한 다음 유형에서 MS SQL을 선택합니다. 소스에 WebServerSG 보안 그룹의 ID를 입력합니다.
 - d. 다른 규칙 추가를 선택한 다음 유형에서 MYSQL을 선택합니다. 소스에 WebServerSG 보안 그룹의 ID를 입력합니다.
 - e. 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. 유형에서 HTTP를 선택합니다. 대상에 0.0.0.0/0을 입력합니다.

- c. 다른 규칙 추가를 선택한 다음 유형에서 HTTPS를 선택합니다. 대상에 0.0.0.0/0을 입력합니다.
- d. Save를 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 정보는 [Amazon EC2 인스턴스 유형](#) 단원을 참조하십시오.

4. 인스턴스 세부 정보 구성 페이지의 네트워크에서 앞서 만든 VPC를 선택한 후 서브넷을 선택합니다. 예를 들어 웹 서버는 퍼블릭 서버에서 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 할당하려면 퍼블릭 IP 자동 할당에서 활성화를 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 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 107\)](#) 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 정보는 [연결 테스트](#)를 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 엘라스틱 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 주소를 선택한 다음, [Actions], [Associate Address]를 선택합니다.

6. 네트워크 인터페이스 또는 인스턴스를 선택합니다. 해당 프라이빗 IP에서 탄력적 IP 주소와 연결할 주소를 선택한 다음 연결을 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=AmazonProvidedDNS`로 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. DHCP 옵션 세트 생성 대화 상자의 도메인 이름 서버에서 Amazon DNS 서버 (AmazonProvidedDNS)의 주소와 해당 DNS 서버의 주소(예: 192.0.2.1)를 쉼표로 구분하여 지정한 후 예, 생성을 선택합니다.
5. 탐색 창에서 [Your VPCs]를 선택합니다.
6. VPC를 선택한 다음, [Actions], [Edit DHCP Options Set]를 선택합니다.
7. DHCP 옵션 세트에서 새로운 옵션 세트의 ID를 선택한 다음 저장을 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 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.

시나리오 4: 프라이빗 서브넷만 있고 액세스를 제공하는 VPC

이 시나리오의 구성에는 단일 프라이빗 서브넷이 있는 Virtual Private Cloud(VPC)와, IPsec VPN 터널을 통해 귀사의 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.

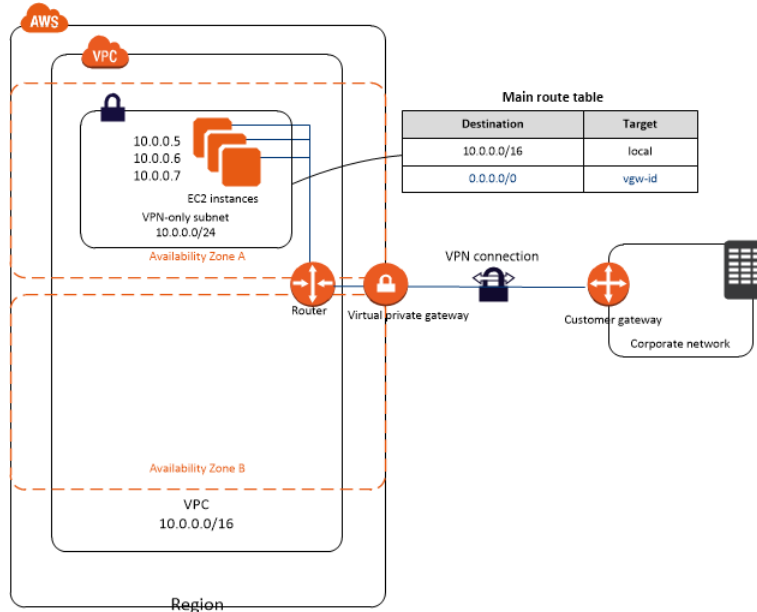
이 시나리오를 IPv6—에 맞게 구성할 수도 있습니다. 즉 VPC 마법사를 이용해 VPC, 그리고 연결된 IPv6 CIDR 블록이 있는 서브넷을 만들 수 있습니다. 서브넷에서 시작한 인스턴스는 IPv6 주소를 받을 수 있습니다. 현재 Amazon은 연결을 통한 IPv6 통신을 지원하지 않습니다. 하지만 VPC의 인스턴스들은 IPv6를 통해 서로 통신할 수 있습니다. IPv4 및 IPv6 주소 지정에 대한 자세한 정보는 [VPC의 IP 주소 지정 \(p. 103\)](#) 단원을 참조하십시오.

내용

- 개요 (p. 55)
- 라우팅 (p. 56)
- 보안 (p. 57)
- 시나리오 4 구현 (p. 57)

개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다.



Important

이 시나리오를 위해 [Amazon VPC 네트워크 관리자 안내서](#)에서는 귀사의 연결에서 네트워크 관리자가 Amazon VPC 고객 게이트웨이를 구성하기 위해 해야 할 사항을 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 간의 연결. 연결은 연결의 Amazon 측에 있는 가상 프라이빗 게이트웨이와 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.
- 기본 라우팅 테이블은 서브넷의 인스턴스가 VPC에서 다른 인스턴스들과 통신할 수 있게 해주는 경로를 포함합니다. 경로 전달이 활성화 되면서 경로는 기본 라우팅 테이블에서 전달된 경로로서 나타나는 네트워크와 서브넷의 인스턴스가 직접적으로 소통할 수 있게 해줍니다.

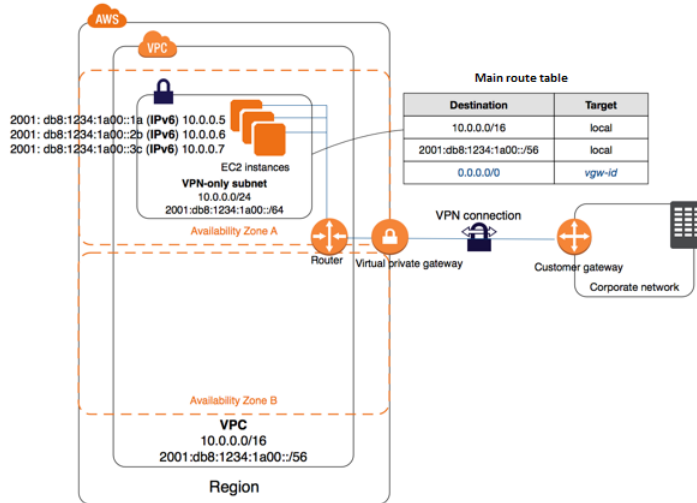
서브넷에 대한 자세한 정보는 [VPC 및 서브넷 \(p. 77\)](#) 및 [VPC의 IP 주소 지정 \(p. 103\)](#)을 참조하십시오. 연결에 대한 자세한 정보는 사용 설명서의 [란 무엇인가?](#)를 참조하십시오. 고객 게이트웨이 구성에 대한 자세한 정보는 [Amazon VPC 네트워크 관리자 안내서](#)를 참조하십시오.

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 인스턴스들이 서로 소통할 수 있도록 IPv6을 사용하게 하는 기본 라우팅 테이블의 라우팅 테이블 입력



라우팅

VPC에는 라우터가 내재되어 있습니다(이 시나리오의 구성 다이어그램 참조). 이 시나리오에서 VPC 마법사는 VPC 외부 주소로 경로가 지정된 모든 트래픽을 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.

대상 주소	대상
10.0.0.0/16	로컬
0.0.0.0/0	vgw-id

연결은 고정 라우팅 연결 또는 동적 라우팅 연결(BGP 사용)로 구성됩니다. 고정 라우팅을 선택하는 경우, 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 다음 인터넷에 도달한 후 AWS에 도달해야 합니다. 현재 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.

대상 주소	대상
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 \(p. 122\)](#) 단원을 참조하십시오.

시나리오 4의 경우, 네트워크 ACL이 아닌 VPC에 대한 기본 보안 그룹을 사용합니다. 네트워크 ACL을 사용하려는 경우 [시나리오 4를 위한 권장 규칙 \(p. 158\)](#)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 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 연결을 생성합니다.

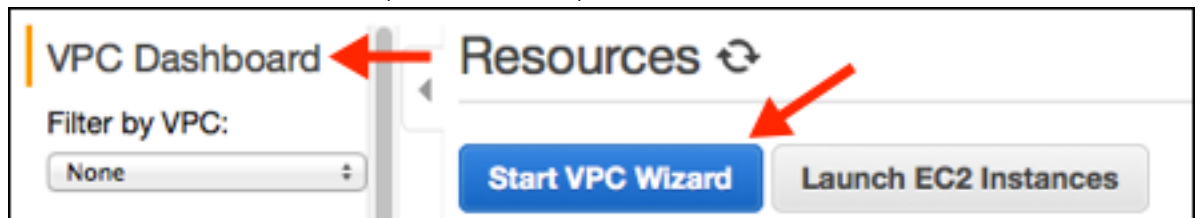
고객 게이트웨이를 준비하려면

1. 고객 게이트웨이로 사용할 디바이스를 결정합니다. 테스트를 통해 검증된 디바이스에 대한 자세한 정보는 [Amazon Virtual Private Cloud FAQ](#)를 참조하십시오. 고객 게이트웨이 요구 사항에 대한 자세한 정보는 [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 연결을 생성하고자 하는 경우, 연결을 통해 가상 프라이빗 게이트웨이로 알려야 하는 내부 IP 범위의 목록(CIDR 표기법)을 얻어야 합니다. 자세한 정보는 사용 설명서의 [라우팅 옵션](#)을 참조하십시오.

VPC 마법사를 사용하여 VPC 및 연결을 생성합니다.

VPC 마법사를 사용하여 VPC를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 Launch VPC Wizard(VPC 마법사 시작)를 선택합니다.



3. 네 번째 옵션인 프라이빗 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC를 선택한 후 선택을 선택합니다.
4. 프라이빗 서브넷만 있고 하드웨어 VPN 액세스를 제공하는 VPC 페이지에서 다음을 수행합니다.
 - a. (선택 사항) 필요하다면 VPC 및 서브넷의 IPv4 CIDR 블록 범위를 수정하거나 기본값을 유지합니다.
 - b. (선택 사항) VPC와 서브넷의 이름을 지정합니다. 이렇게 하면 나중에 콘솔에서 이들을 식별하는 데 도움이 됩니다.
 - c. (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 기본값(00)을 유지합니다.
 - d. [Next]를 선택합니다.
5. VPN 구성 페이지에서 다음을 수행합니다.
 - a. 고객 게이트웨이 IP에서 VPN 라우터의 퍼블릭 IP 주소를 지정합니다.
 - b. (선택 사항) 고객 게이트웨이와 연결의 이름을 지정합니다.
 - c. 라우팅 유형에서 라우팅 옵션 중 하나를 선택합니다. 자세한 정보는 사용 설명서의 [라우팅 옵션](#)을 참조하십시오.
 - VPN 라우터가 BGP(Border Gateway Protocol)를 지원할 경우 [Dynamic (requires BGP)]을 선택합니다.
 - VPN 라우터에서 BGP를 지원하지 않으면 정적을 선택합니다. IP 접두사에서 CIDR 표기법으로 각각의 네트워크 IP 범위를 추가합니다.
 - d. [Create VPC]를 선택합니다.
6. 마법사가 완료되면 탐색 창에서 연결을 선택합니다. 마법사가 생성한 연결을 선택하고 구성 다운로드를 선택합니다. 대화 상자에서 고객 게이트웨이 공급업체, 플랫폼, 소프트웨어 버전을 선택하고 [Yes, Download]를 선택합니다.

7. 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 시나리오의 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. 유형에서 SSH를 선택한 후 소스에 네트워크의 프라이빗 IP 주소 범위(예: 172.0.0.0/8)를 입력합니다.
 - b. 다른 규칙 추가를 선택한 다음 유형에서 RDP를 선택하고 소스에 네트워크의 프라이빗 IP 주소 범위를 입력합니다.
 - c. Save를 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 방법을 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 정보는 [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

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. 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 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.

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.

작업

- 1단계: VPC와 서브넷 만들기 (p. 60)
- 2단계: 서브넷을 퍼블릭으로 만들기 (p. 61)
- 3단계: 서브넷에서 인스턴스 시작 (p. 63)
- 4단계: 정리 (p. 64)

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 \(p. 80\)](#) 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",  
  ...  
}
```

- 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
```

- 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

- 인터넷 게이트웨이 생성.

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

반환된 출력에 표시된 인터넷 게이트웨이 ID를 메모해 둡니다.

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

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

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

- 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 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"  
        }  
      ]  
    }  
  ]  
}
```

6. 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"  
  }  
]
```

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
```

- 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 \(p. 123\)](#) 단원을 참조하십시오. 키 페어에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EC2 키 페어를 참조하십시오.

퍼블릭 서브넷에서 인스턴스를 시작하고 연결하려면

- 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
```

- 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.

- 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 [Linux AMI 찾기](#) 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.

작업

- 1단계: VPC와 서브넷 만들기 (p. 65)
- 2단계: 퍼블릭 서브넷 구성 (p. 66)
- 3단계: 외부 전용 프라이빗 서브넷 구성 (p. 68)
- 4단계: 서브넷의 IPv6 주소 지정 동작 변경 (p. 69)
- 5단계: 퍼블릭 서브넷에서 인스턴스 시작 (p. 69)
- 6단계: 프라이빗 서브넷으로 인스턴스를 시작 (p. 71)
- 7단계: 정리 (p. 72)

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. 80) 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 생성.

```
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 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.

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": "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"
  }
]
```

- 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

- 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(::/0)이 외부 전용 인터넷 게이트웨이를 가리키는 경로를 만듭니다.

```
aws ec2 create-route --route-table-id rtb-abc123ab --destination-ipv6-cidr-block ::/0
--egress-only-internet-gateway-id ewg-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 테스트 하려면, 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 \(p. 123\)](#) 단원을 참조하십시오. 키 페어에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EC2 키 페어를 참조하십시오.

퍼블릭 서브넷에서 인스턴스를 시작하고 연결하려면

1. 키 페어를 만든 다음 --query 옵션과 --output 텍스트 옵션을 사용하여 확장자가 .pem인 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 [Linux AMI 찾기](#) 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 인터넷에 접속할 수 없는 인스턴스를 확인해야 하는 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 \(p. 216\)](#)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
```

```
PING ietf.org(2001:1900:3001:11::2c) 56 data bytes
64 bytes from 2001:1900:3001:11::2c: icmp_seq=1 ttl=46 time=73.9 ms
64 bytes from 2001:1900:3001:11::2c: icmp_seq=2 ttl=46 time=73.8 ms
64 bytes from 2001:1900:3001:11::2c: icmp_seq=3 ttl=46 time=73.9 ms
...
```

6. 인터넷의 호스트가 프라이빗 서브넷의 인스턴스에 접속할 수 있는지 테스트하려면 IPv6를 사용할 수 있는 컴퓨터에서 ping6 명령을 사용합니다. 제한 시간 응답을 얻어야 합니다. 유효한 응답을 얻으면 인터넷에서 인스턴스에 액세스할 수 있습니다.—그러면 프라이빗 서브넷과 연결된 라우팅 테이블을 확인하고 인터넷 게이트웨이에 대한 IPv6 트래픽 경로가 있는지 확인합니다.

```
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 egw-015e0e244e24dfe8a
```

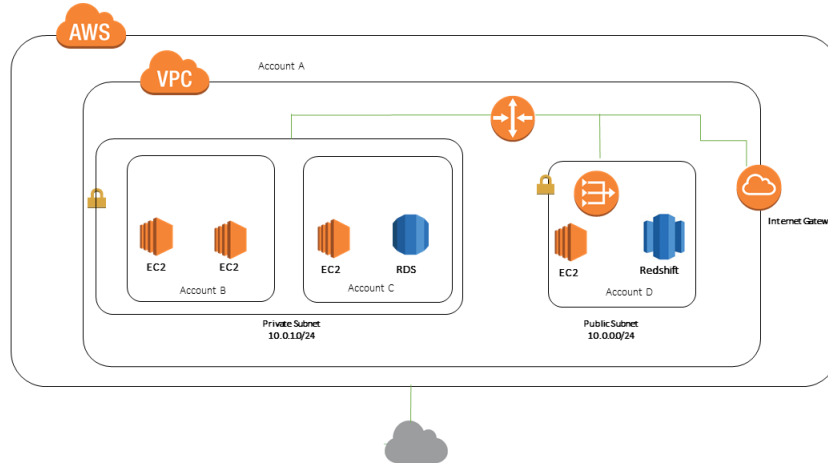
7. VPC 삭제:

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

예제: 퍼블릭 서브넷과 프라이빗 서브넷 공유

서브넷, 라우팅 테이블, 게이트웨이 및 CIDR 범위 및 동일한 AWS 조직에서 서브넷을 사용할 다른 계정을 포함하여, 인프라에 대한 책임을 담당할 계정이 필요한 경우 이 시나리오를 참조해 보십시오. VPC 소유자(계정 A)는 VPC, 서브넷, 라우팅 테이블, 게이트웨이, 네트워크 ACL을 비롯한 라우팅 인프라를 생성합니다. 계정 B와 계정 C는 퍼블릭 애플리케이션을 생성하려고 합니다. 계정 D는 인터넷에 연결할 필요가 없으며 프라이빗 서브넷에 상주해야 하는 프라이빗 애플리케이션을 생성하려고 합니다. 계정 A는 AWS 리소스 액세스 관리자를 사용하여 서브넷에 대한 리소스 공유를 생성하고 서브넷을 공유할 수 있습니다. 계정 A는 프라이빗 및 퍼블릭 서브넷을 계정 B 및 계정 C와 공유하고, 프라이빗 서브넷을 계정 D와 공유합니다. 계정 B, 계정 C, 계정 D는 서브넷에 리소스를 생성할 수 있습니다. 각 계정은 공유된 서브넷만 볼 수 있습니다. 예를 들어 계정 D는 프라이빗 서브넷만 볼 수 있습니다. 각 계정은 인스턴스 및 보안 그룹을 비롯하여 해당 리소스를 제어할 수 있습니다.

계정 A는 퍼블릭 서브넷에 대한 라우팅 테이블 및 프라이빗 서브넷을 비롯하여 IP 인프라를 관리합니다. 공유된 서브넷에 필요한 추가 구성은 없기 때문에 라우팅 테이블은 비공유 서브넷 라우팅 테이블과 동일합니다.



계정 A(계정 ID 111111111111)는 계정 D(444444444444)와 프라이빗 서브넷을 공유합니다. 계정 D는 다음 서브넷을 볼 수 있으며, 소유자 옆에는 서브넷이 공유되었음을 나타내는 두 가지 지표가 표시됩니다.

- 계정 ID는 VPC 소유자(111111111111)이며 계정 D의 ID(444444444444)와 다릅니다.
- 소유자 계정 ID 옆에 "shared"라는 단어가 표시됩니다.

Create subnet Actions		Filter by tags and attributes or search by keyword						
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	Route table	Default subnet	Owner
	subnet-0bb1c79de301436ee	available	vpc-0ee975135d74bdcfe	10.0.2.0/24	251	rtb-0825a8caf09467ea8	No	111111111111 (S)
	subnet-0fe673ef5bd459924	available	vpc-0ee975135d74bdcfe	10.0.1.0/24	251	rtb-0825a8caf09467ea8	No	111111111111 (S)

예제: AWS PrivateLink 및 리전 간 VPC 피어링을 사용하여 리전 간에 분산된 서비스

AWS PrivateLink 서비스 공급자는 Network Load Balancer를 프런트 엔드로 사용하여 해당 VPC에서 서비스를 실행하는 인스턴스를 구성합니다. 서비스 공급자와 동일한 리전에 있는 서비스 소비자는 AWS PrivateLink VPC 엔드포인트를 통해 이러한 서비스에 비공개로 액세스할 수 있습니다. 리전 간 VPC 피어링을 사용하여 서로 다른 리전에서 실행하는 리소스는 프라이빗 통신을 수행할 수 있습니다. AWS PrivateLink 및 리전 간 VPC 피어링은 Amazon의 네트워크를 통해 트래픽을 전송합니다. 리전 간 피어링과 AWS PrivateLink를 함께 사용하여 리전 전체에서 프라이빗 서비스에 대한 액세스를 확장할 수 있습니다.

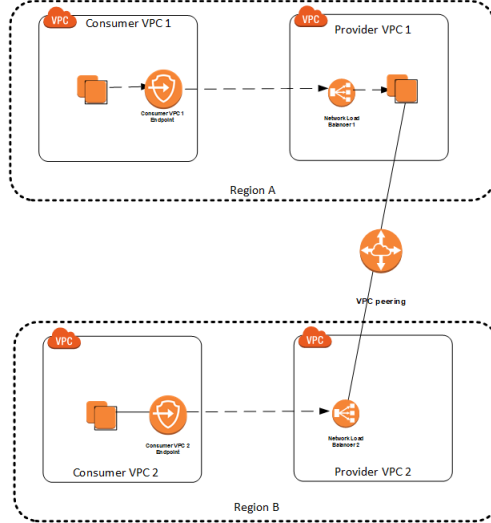
서비스 소비자 또는 서비스 공급자가 구성을 수행할 수 있습니다. 자세한 정보는 다음 예를 참조하십시오.

예제

- 예제: 서비스 공급자가 리전에 분산하도록 서비스 구성 (p. 75)
- 예제: 서비스 소비자가 리전 간 액세스 구성 (p. 75)

예제: 서비스 공급자가 리전에 분산하도록 서비스 구성

서비스가 리전 A(예: us-east-1 리전)에 있는 공급자 VPC 1의 인스턴스에서 실행하는 다음 예제를 참조하십시오. 동일한 리전에 있는 소비자 VPC 1의 리소스는 소비자 VPC 1의 AWS PrivateLink VPC 엔드포인트를 통해 서비스에 직접 액세스할 수 있습니다.



리전 B(예: eu-west-2 리전)의 소비자 VPC 1에 있는 리소스가 서비스에 비공개로 액세스하도록 허용하려면 서비스 공급자가 다음 절차를 완료해야 합니다.

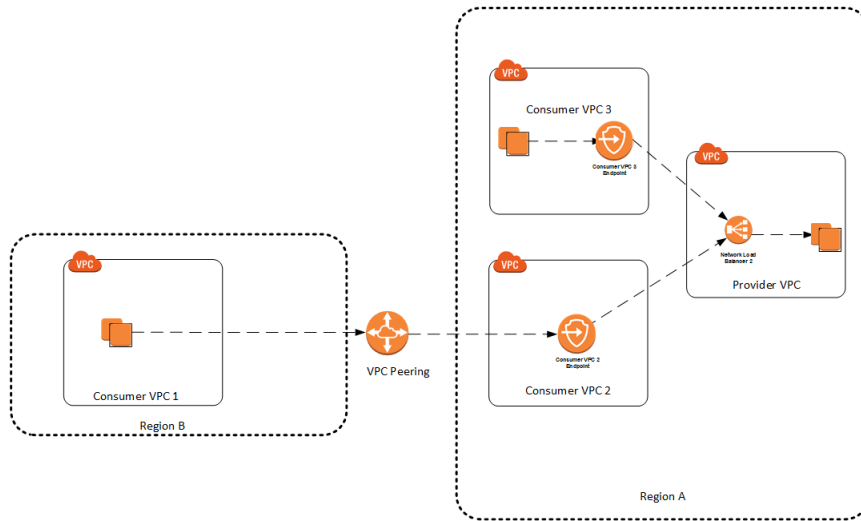
1. 리전 B에 공급자 VPC 2를 생성합니다.
2. 두 개의 VPC 간에 트래픽을 라우팅할 수 있도록 공급자 VPC 1과 공급자 VPC 2 간의 리전 간 피어링을 구성합니다.
3. 공급자 VPC 2에 Network Load Balancer를 생성합니다.
4. Network Load Balancer 2에서 VPC 1에 있는 서비스 인스턴스의 IP 주소를 가리키는 대상 그룹을 구성합니다.
5. Network Load Balancer 2로부터의 트래픽을 허용하도록 공급자 VPC 1의 서비스 인스턴스와 연결된 보안 그룹을 조정합니다.
6. 공급자 VPC 2에 VPC 엔드포인트 서비스 구성을 생성하여 Network Load Balancer 2에 연결합니다.

구성 완료 후 소비자는 소비자 VPC 2에 VPC 엔드포인트를 생성할 수 있으며 서비스에 비공개로 액세스할 수 있습니다.

소비자 계정에는 VPC 엔드포인트 데이터 처리 요금 및 VPC 엔드포인트 시간별 요금이 부과됩니다. 공급자 계정에는 리전 간 피어링 데이터 전송 요금, Network Load Balancer 요금 및 서비스 인스턴스 요금이 부과됩니다.

예제: 서비스 소비자가 리전 간 액세스 구성

서비스가 리전 A(예: us-east-1 리전)에 있는 공급자 VPC의 인스턴스에서 실행하는 다음 예제를 참조하십시오. 소비자 VPC 3의 리소스는 소비자 VPC 3의 AWS PrivateLink VPC 엔드포인트 서비스를 통해 서비스에 직접 액세스할 수 있습니다.



리전 B(예: eu-west-1 리전)의 소비자 VPC 1에 있는 리소스가 서비스에 비공개로 액세스하도록 허용하려면 서비스 소비자가 다음 절차를 완료해야 합니다.

1. 리전 A에 소비자 VPC 2를 생성합니다.
2. 소비자 VPC 2의 한 개 이상 서브넷에 분산되어 있는 VPC 엔드포인트를 생성합니다.
3. 소비자 VPC 1의 인스턴스로부터의 트래픽을 허용하도록 소비자 VPC 2의 VPC 엔드포인트 서비스와 연결된 보안 그룹을 조정합니다. 소비자 VPC 2에서 VPC 엔드포인트로의 트래픽을 허용하도록 소비자 VPC 1의 인스턴스와 연결된 보안 그룹을 조정합니다.
4. 두 개의 VPC 간에 트래픽이 라우팅되도록 소비자 VPC 1과 소비자 VPC 2 간의 리전 간 피어링을 구성합니다.

구성 완료 후 리전 B의 소비자 VPC 1은 서비스에 비공개로 액세스할 수 있습니다.

소비자 계정에는 리전 간 피어링 데이터 전송 요금, VPC 엔드포인트 데이터 처리 요금 및 VPC 엔드포인트 시간별 요금이 부과됩니다. 공급자에는 Network Load Balancer 요금과 서비스 인스턴스 요금이 부과됩니다.

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.

내용

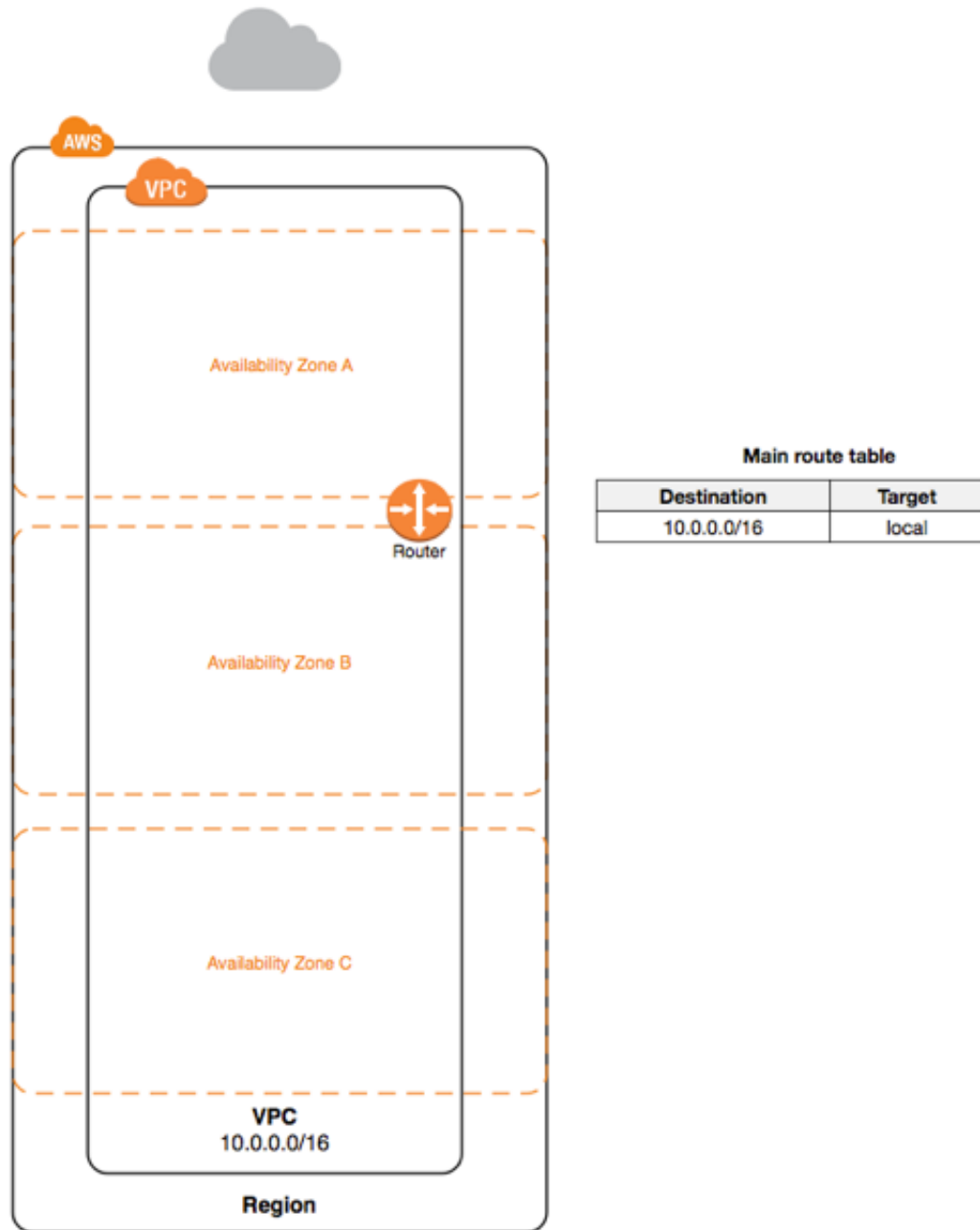
- [VPC 및 서브넷 기본 사항 \(p. 77\)](#)
- [VPC 및 서브넷 크기 \(p. 80\)](#)
- [서브넷 라우팅 \(p. 84\)](#)
- [서브넷 보안 \(p. 85\)](#)
- [로컬 네트워크 및 기타 VPC와의 연결 \(p. 85\)](#)
- [VPC 및 서브넷 관련 작업 \(p. 86\)](#)
- [공유된 VPC에 대한 작업 \(p. 92\)](#)

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). 이것은 VPC의 기본 CIDR 블록입니다. CIDR 표기법에 대한 자세한 정보는 [RFC 4632](#)을 참조하십시오.

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

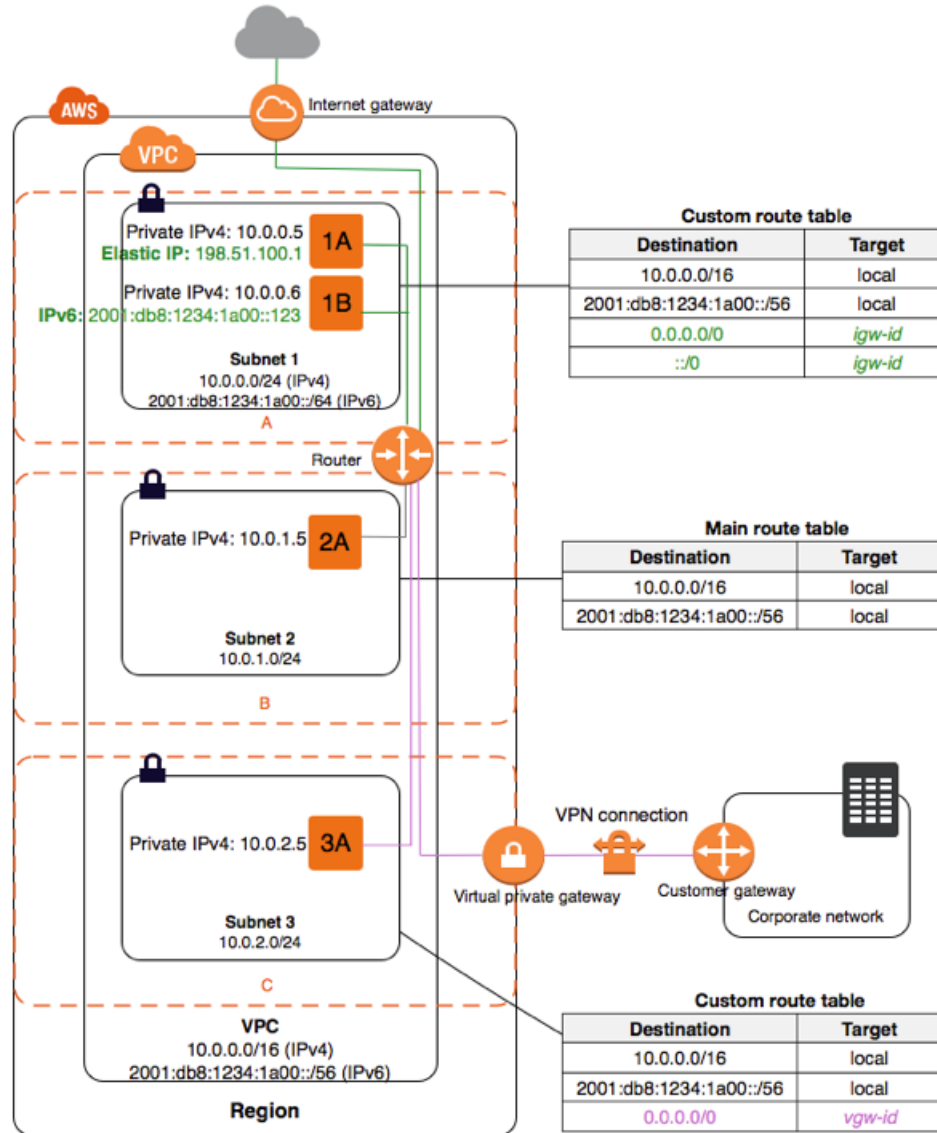


VPC는 리전의 모든 가용 영역에 적용됩니다. VPC를 만든 후 각 가용 영역에 하나 이상의 서브넷을 추가할 수 있습니다. 서브넷을 만들 때 해당 서브넷에 대한 CIDR 블록을 지정합니다. 이는 VPC CIDR 블록의 서브넷입니다. 각 서브넷은 단일 가용 영역 내에서만 존재해야 하며, 여러 영역으로 확장할 수 없습니다. 각 가용 영역은 다른 가용 영역에서 발생한 장애를 격리시킬 수 있도록 서로 분리된 공간입니다. 별도의 가용 영역에서 인스턴스를 시작함으로써 단일 위치에서 장애가 발생할 경우 애플리케이션을 보호할 수 있습니다. AWS는 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 정보는 [퍼블릭 IPv4 주소 \(p. 104\)](#) 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 연결을 위한 가상 프라이빗 게이트웨이로 라우팅되는 경우, 이 서브넷을 VPN 전용 서브넷이라고 합니다. 이 그림에서는 서브넷 3이 VPN 전용 서브넷입니다. 현재 연결을 통한 IPv6 트래픽은 지원하지 않습니다.

자세한 정보는 사용 설명서의 [시나리오 및 예시 \(p. 23\)](#), [인터넷 게이트웨이 \(p. 202\)](#), [란 무엇인가?](#)를 참조하십시오.

Note

서브넷의 유형과 관계없이 서브넷의 내부 IPv4 주소 범위는 항상 프라이빗—입니다. 즉 인터넷으로 주소 블록을 알리지 않습니다.

계정에서 만들 수 있는 VPC 및 서브넷 수에는 제한이 있습니다. 자세한 정보는 [Amazon VPC 제한 \(p. 285\)](#) 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. 필요할 경우 IPv6 CIDR 블록을 VPC에 연결할 수 있습니다.

IP 주소에 대한 자세한 정보는 [VPC의 IP 주소 지정 \(p. 103\)](#) 단원을 참조하십시오.

내용

- [IPv4의 경우, VPC 및 서브넷 크기 조정 \(p. 80\)](#)
- [VPC에 IPv4 CIDR 블록 추가 \(p. 81\)](#)
- [IPv6의 경우, VPC 및 서브넷 크기 조정 \(p. 84\)](#)

IPv4의 경우, VPC 및 서브넷 크기 조정

VPC를 만들 때 VPC의 IPv4 CIDR 블록을 지정해야 합니다. 허용된 블록 크기는 /16 넷마스크 (IP 주소 65,536개) ~ /28 넷마스크 (IP 주소 16개)입니다. VPC 생성을 마쳤으면 보조 CIDR 블록을 VPC에 연결할 수 있습니다. 자세한 정보는 [VPC에 IPv4 CIDR 블록 추가 \(p. 81\)](#) 단원을 참조하십시오.

VPC를 생성하는 경우, 다음과 같이 /16 RFC 1918: [규격에 따라 프라이빗\(비공개적으로 라우팅 가능\) IPv4 주소 범위에 속하는 CIDR 블록](#) (또는 이하)을 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.

Note

다른 AWS 서비스에 사용할 VPC를 만들려면, 해당 서비스 설명서를 참조하여 IP 주소 범위나 네트워크 구성 요소에 대한 특정 요구 사항이 있는지 확인하십시오.

서브넷의 CIDR 블록은 VPC(VPC에서 단일 서브넷을 사용할 경우)에 대한 CIDR 블록 또는 VPC에 대한 CIDR 블록의 하위 집합(여러 서브넷을 사용할 경우)과 동일할 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.

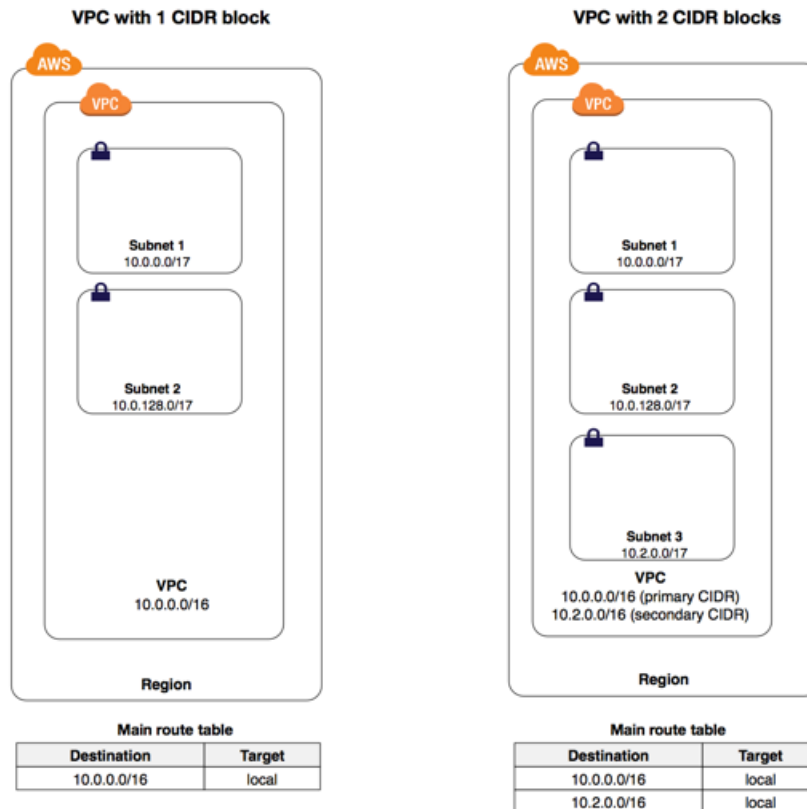
각 서브넷 CIDR 블록에서 첫 4개의 IP 주소와 마지막 IP 주소는 사용자가 사용할 수 없으므로 인스턴스에 할당할 수 없습니다. 예를 들어 10.0.0.0/24 CIDR 블록의 서브넷에서는 다음 5개 IP 주소가 예약되어 있습니다.

- 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 AWS는 각 서브넷 범위의 기준 + 2도 예약합니다. CIDR 블록이 여러 개인 VPC의 경우, DNS 서버의 IP 주소가 기본 CIDR에 위치합니다. 자세한 정보는 [Amazon DNS 서버 \(p. 236\)](#) 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.

VPC에 IPv4 CIDR 블록 추가

보조 IPv4 CIDR 블록을 VPC와 연결할 수 있습니다. CIDR 블록을 VPC에 연결하면 VPC 라우팅 테이블에 경로가 자동으로 추가되면서 VPC 내에서 라우팅이 가능하게 됩니다(대상 주소는 CIDR 블록이고 대상은 local).

아래 예에서 왼쪽의 VPC는 단일 CIDR 블록(10.0.0.0/16)과 서브넷 두 개를 가지고 있습니다. 오른쪽 VPC는 두 번째 CIDR 블록(10.2.0.0/16)을 추가하고 두 번째 CIDR 범위에서 새로운 서브넷을 생성하고 나면 동일한 VPC 아키텍처를 보여줍니다.



VPC에 CIDR 블록을 추가할 경우 다음 규칙이 적용됩니다.

- 허용된 블록 크기는 /28 넷마스크 ~ /16 넷마스크입니다.
- CIDR 블록은 VPC에 연결된 기존 CIDR 블록과 겹치지 않습니다.
- 사용 가능한 IPv4 주소 범위에 제한이 있습니다. 자세한 정보는 [IPv4 CIDR 블록 연결 제한 \(p. 82\)](#) 단원을 참조하십시오.
- 기존 CIDR 블록의 크기를 늘리거나 줄일 수 없습니다.
- VPC에 연결할 수 있는 CIDR 블록의 수와 라우팅 테이블에 추가할 수 있는 경로의 수에는 제한이 있습니다. 최대 수를 초과하는 경우에는 CIDR 블록을 연결할 수 없습니다. 자세한 정보는 [Amazon VPC 제한 \(p. 285\)](#) 단원을 참조하십시오.
- CIDR 블록은 모든 VPC 라우팅 테이블에서 경로의 CIDR 범위보다 작아야 합니다. 예를 들어 경로에서 가상 프라이빗 게이트웨이에 대한 대상 주소가 10.0.0.0/24인 경우, 이와 범위가 같거나 큰 CIDR 블록을 연결할 수 없습니다. 그러나 대상 주소의 범위가 10.0.0.0/25 이하인 CIDR 블록은 연결이 가능합니다.
- ClassicLink에서 VPC를 활성화하면 10.0.0.0/16 및 10.1.0.0/16 범위의 CIDR 블록은 연결할 수 있지만, 10.0.0.0/8 범위의 기타 CIDR 블록은 연결할 수 없습니다.
- VPC 피어링 연결에 포함된 VPC에 IPv4 CIDR 블록을 추가할 때 다음 규칙이 적용됩니다.
 - VPC 피어링 연결이 active인 경우, 피어 VPC의 CIDR 블록과 겹치지 않으면 VPC에 CIDR 블록을 추가할 수 있습니다.
 - VPC 피어링 연결이 pending-acceptance인 경우, 수락자 VPC의 CIDR 블록과 겹치는지 여부에 관계 없이 요청자 VPC의 소유자가 VPC에 CIDR 블록을 추가할 수 없습니다. 수락자 VPC의 소유자가 피어링 연결을 수락하거나, 요청자 VPC의 소유자가 VPC 피어링 연결 요청을 삭제하고 CIDR 블록을 추가한 다음 VPC 피어링 연결을 새로 요청해야 합니다.
 - VPC 피어링 연결이 pending-acceptance인 경우, 수락자 VPC의 소유자는 VPC에 CIDR 블록을 추가할 수 있습니다. 보조 CIDR 블록이 요청자 VPC의 CIDR 블록과 겹치는 경우에는 VPC 피어링 연결 요청이 실패하고 요청을 수락할 수 없게 됩니다.
- AWS Direct Connect를 사용하여 직접 연결 게이트웨이를 통해 여러 VPC에 연결하는 경우 직접 연결 게이트웨이에 연결된 VPC에는 중첩되는 CIDR 블록이 있으면 안 됩니다. 직접 연결 게이트웨이와 연결된 VPC 중 하나에 CIDR 블록을 추가한 경우 새로운 CIDR 블록이 다른 연결된 VPC에 있는 기존 CIDR 블록과 중첩되어서는 안 됩니다. 자세한 정보는 AWS Direct Connect 사용 설명서의 [Direct Connect 게이트웨이](#)를 참조하십시오.
- CIDR 블록을 추가하거나 제거하는 경우, associating | associated | disassociating | disassociated | failing | failed와 같은 다양한 상태를 통과할 수 있습니다. 사용자가 CIDR 블록을 사용할 수 있는 상태가 되면 associated 상태가 됩니다.

아래 표에는 VPC의 기본 CIDR 블록이 상주하는 IPv4 주소 범위에 따라 수락 및 제한되는 CIDR 블록 연결에 대한 개요를 제공합니다.

IPv4 CIDR 블록 연결 제한

기본 VPC CIDR 블록이 상주하는 IP 주소 범위	제한되는 CIDR 블록 연결	허용되는 CIDR 블록 연결
10.0.0.0/8	<p>다른 RFC 1918* 범위(172.16.0.0/12 및 192.168.0.0/16)의 CIDR 블록입니다.</p> <p>기본 CIDR이 10.0.0.0/15 범위에 해당되면 10.0.0.0/16 범위의 CIDR 블록을 추가할 수 없습니다.</p> <p>198.19.0.0/16 범위의 CIDR 블록입니다.</p>	<p>제한되지 않는 10.0.0.0/8 r범위의 기타 모든 CIDR입니다.</p> <p>공개적으로 라우팅이 가능한 모든 IPv4 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.</p>

기본 VPC CIDR 블록이 상주하는 IP 주소 범위	제한되는 CIDR 블록 연결	허용되는 CIDR 블록 연결
172.16.0.0/12	<p>다른 RFC 1918* 범위(10.0.0.0/8 및 192.168.0.0/16)의 CIDR 블록입니다.</p> <p>172.31.0.0/16 범위의 CIDR 블록입니다.</p> <p>198.19.0.0/16 범위의 CIDR 블록입니다.</p>	<p>제한되지 않는 172.16.0.0/12 r범위의 기타 모든 CIDR입니다.</p> <p>공개적으로 라우팅이 가능한 모든 IPv4 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.</p>
192.168.0.0/16	<p>다른 RFC 1918* 범위 (172.16.0.0/12 및 10.0.0.0/8)의 CIDR 블록입니다.</p> <p>198.19.0.0/16 범위의 CIDR 블록입니다.</p>	<p>192.168.0.0/16 범위의 기타 모든 CIDR입니다.</p> <p>공개적으로 라우팅이 가능한 모든 IPv4 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.</p>
198.19.0.0/16	RFC 1918* 범위의 CIDR 블록입니다.	공개적으로 라우팅이 가능한 모든 IPv4 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.
공개적으로 라우팅이 가능한 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.	<p>RFC 1918* 범위의 CIDR 블록입니다.</p> <p>198.19.0.0/16 범위의 CIDR 블록입니다.</p>	공개적으로 라우팅이 가능한 모든 다른 IPv4 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록.

*RFC 1918 범위는 [RFC 1918](#)에 지정된 프라이빗 IPv4 주소 범위입니다.

VPC에 연결한 CIDR 블록은 연결 해제가 가능하지만, 원래 VPC(기본 CIDR 블록)를 생성한 CIDR 블록은 연결을 해제할 수 없습니다. Amazon VPC 콘솔에서 VPC의 기본 CIDR을 보려면 Your VPCs를 선택하고 VPC를 선택한 다음, CIDR 블록 아래 첫 번째 항목을 기록해 둡니다. 아니면 [describe-vpcs](#) 명령을 사용할 수 있습니다.

```
aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d
```

출력 반환 시 기본 CIDR이 최상위 CidrBlock 요소(아래 출력 예제의 끝에서 두 번째 요소)에 반환됩니다.

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-1a2b3c4d",
      "InstanceTenancy": "default",
      "Tags": [
        {
          "Value": "MyVPC",
          "Key": "Name"
        }
      ],
      "CidrBlockAssociations": [
        {
          "AssociationId": "vpc-cidr-assoc-3781aa5e",
          "CidrBlock": "10.0.0.0/16",
          "CidrBlockState": {
```

```

        "State": "associated"
      }
    },
    {
      "AssociationId": "vpc-cidr-assoc-0280ab6b",
      "CidrBlock": "10.2.0.0/16",
      "CidrBlockState": {
        "State": "associated"
      }
    }
  ],
  "State": "available",
  "DhcpOptionsId": "dopt-e0fe0e88",
  "CidrBlock": "10.0.0.0/16",
  "IsDefault": false
}
]
}

```

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.

각 서브넷 CIDR 블록에서 첫 4개의 IPv6 주소와 마지막 IPv6 주소는 사용자가 사용할 수 없으므로 인스턴스에 할당할 수 없습니다. 예를 들어 2001:db8:1234:1a00/64 CIDR 블록의 서브넷에서는 다음 5개 IP 주소가 예약되어 있습니다.

- 2001:db8:1234:1a00::
- 2001:db8:1234:1a00::1
- 2001:db8:1234:1a00::2
- 2001:db8:1234:1a00::3
- 2001:db8:1234:1a00:ffff:ffff:ffff:ffff

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 및 기본 라우팅 테이블의 내용을 변경할 수 있습니다. 자세한 정보는 [라우팅 테이블 \(p. 190\)](#) 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

(IPv4만 해당) 인스턴스에 연결된 탄력적 IPv4 주소 또는 퍼블릭 IPv4 주소는 VPC의 인터넷 게이트웨이를 통해 액세스할 수 있습니다. 인스턴스 및 다른 네트워크 간 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 정보는 [NAT \(p. 210\)](#) 단원을 참조하십시오. IPv6를 통해 인터넷에 대한 아웃바운드 전용 연결을 시작하기 위해 외부 전용 인터넷 게이트웨이를 사용할 수 있습니다. 자세한 정보는 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 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 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 \(p. 122\)](#) 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 정보는 [네트워크 ACL \(p. 130\)](#) 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 정보는 [VPC 흐름 로그 \(p. 173\)](#) 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 겹

치는 VPC 간에는 VPC 피어링 연결을 만들 수 없습니다. 자세한 정보는 [Amazon VPC Peering Guide](#) 단원을 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 연결은 지원하지 않습니다.

VPC 및 서브넷 관련 작업

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 정보는 [시나리오 및 예시 \(p. 23\)](#) 단원을 참조하십시오.

작업

- [VPC 만들기 \(p. 86\)](#)
- [VPC에서 서브넷 만들기 \(p. 87\)](#)
- [VPC에 보조 IPv4 CIDR 블록 연결 \(p. 88\)](#)
- [IPv6 CIDR 블록을 VPC와 연결 \(p. 88\)](#)
- [IPv6 CIDR 블록을 서브넷에 연결 \(p. 89\)](#)
- [서브넷에서 인스턴스 시작 \(p. 89\)](#)
- [서브넷 삭제 \(p. 90\)](#)
- [VPC에서 IPv4 CIDR 블록의 연결을 해제 \(p. 90\)](#)
- [VPC 또는 서브넷에 연결된 IPv6 CIDR 블록을 분리 \(p. 91\)](#)
- [VPC 삭제 \(p. 91\)](#)

VPC 만들기

Amazon VPC 콘솔을 사용하여 빈 VPC를 만들 수 있습니다.

콘솔을 사용하여 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 VPC 만들기를 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).
- Note**
- 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.
- IPv6 CIDR block: Amazon-provided IPv6 CIDR block을 선택하여 IPv6 CIDR 블록을 VPC에 연결할 수 있는 옵션.
 - 테넌시: 테넌시 옵션을 선택합니다. 전용 테넌시는 단일 테넌트 하드웨어에서 인스턴스를 실행할 수 있게 해줍니다. 자세한 정보는 [Linux 인스턴스용 Amazon EC2 사용 설명서](#)의 전용 인스턴스를 참조하십시오.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄을 사용하여 VPC를 만들려면

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

명령줄을 사용하여 VPC를 설명하려면

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

IP 주소에 대한 자세한 정보는 [VPC의 IP 주소 지정 \(p. 103\)](#) 주제를 참조하십시오.

VPC를 생성한 후 서브넷을 생성할 수 있습니다. 자세한 정보는 [VPC에서 서브넷 만들기 \(p. 87\)](#) 단원을 참조하십시오.

VPC에서 서브넷 만들기

VPC에 새로운 서브넷을 추가하려면 VPC 범위에서 서브넷의 IPv4 CIDR 블록을 설정해야 합니다. 서브넷을 상주시키려는 가용 영역을 지정할 수 있습니다. 동일한 가용 영역에 여러 서브넷을 둘 수 있습니다.

IPv6 CIDR 블록이 VPC와 연결되어 있는 경우 서브넷에 대해 IPv6 CIDR 블록을 지정할 수도 있습니다.

콘솔을 사용하여 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 정보는 [IPv4의 경우, VPC 및 서브넷 크기 조정 \(p. 80\)](#) 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.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄을 사용하여 서브넷을 추가하려면

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

명령줄을 사용하여 서브넷을 설명하려면

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

서브넷을 만든 후 다음을 수행할 수 있습니다.

- 라우팅을 구성합니다. 서브넷을 퍼블릭 서브넷이 되게 하려면 먼저 VPC에 인터넷 게이트웨이를 연결해야 합니다. 자세한 정보는 [인터넷 게이트웨이 생성 및 연결 \(p. 204\)](#) 단원을 참조하십시오. 그런 다음 사용자 지정 라우팅 테이블을 만들고, 인터넷 게이트웨이에 경로를 추가할 수 있습니다. 자세한 정보는 [사용자 지정 라우팅 테이블 생성 \(p. 205\)](#) 단원을 참조하십시오. 다른 라우팅 옵션의 경우, [라우팅 테이블 \(p. 190\)](#)을 참조하십시오.
- 서브넷 설정을 수정하여 해당 서브넷에서 시작하는 모든 인스턴스가 퍼블릭 IPv4 주소나 IPv6 주소, 또는 둘 다를 받도록 지정합니다. 자세한 정보는 [서브넷에 대한 IP 주소 지정 동작 \(p. 106\)](#) 단원을 참조하십시오.
- 필요에 따라 보안 그룹을 생성 또는 수정합니다. 자세한 정보는 [VPC의 보안 그룹 \(p. 123\)](#) 단원을 참조하십시오.
- 필요에 따라 네트워크 ACL을 생성 또는 수정합니다. 자세한 정보는 [네트워크 ACL \(p. 130\)](#) 단원을 참조하십시오.
- 다른 계정과 서브넷을 공유합니다. 자세한 정보는 [??? \(p. 93\)](#) 단원을 참조하십시오.

VPC에 보조 IPv4 CIDR 블록 연결

VPC에 또 다른 IPv4 CIDR 블록을 추가할 수 있습니다. 해당하는 [제한 사항 \(p. 81\)](#)을 숙지해야 합니다.

CIDR 블록을 연결하고 나면 상태가 `associating`이 됩니다. CIDR 블록이 사용할 준비가 되면 `associated` 상태가 됩니다.

콘솔을 사용하여 VPC에 CIDR 블록을 추가하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다.
3. VPC를 선택하고 [Actions]와 [Edit CIDRs]를 차례로 선택합니다.
4. [Add IPv4 CIDR]을 선택하고 추가할 CIDR 블록(예: 10.2.0.0/16)을 입력합니다. 체크 표시 아이콘을 선택합니다.
5. 닫기를 선택합니다.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄을 사용하여 CIDR 블록을 추가하려면

- `associate-vpc-cidr-block`(AWS CLI)
- `Register-EC2VpcCidrBlock`(Windows PowerShell용 AWS 도구)

필요한 IPv4 CIDR 블록을 추가하고 난 후에는 서브넷을 생성할 수 있습니다. 자세한 정보는 [VPC에서 서브넷 만들기 \(p. 87\)](#) 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.

콘솔을 사용하여 VPC에 IPv6 CIDR 블록을 연결하려면

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

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 VPC에 IPv6 CIDR 블록을 연결하려면

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

IPv6 CIDR 블록을 서브넷에 연결

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

콘솔을 사용하여 서브넷에 IPv6 CIDR 블록을 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 서브넷을 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. 닫기를 선택합니다.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 서브넷에 IPv6 CIDR 블록을 연결하려면

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

서브넷에서 인스턴스 시작

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

콘솔을 사용하여 서브넷으로 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 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. 현재 세대의 모든 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 Storage]를 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 [Launch]를 선택합니다.

7. 소유한 기존 키 페어를 선택하거나 새 키 페어를 생성한 다음, 작업이 끝나면 Launch Instances를 선택합니다.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 서브넷으로 인스턴스를 시작하려면

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

서브넷 삭제

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

콘솔을 사용하여 서브넷을 삭제하려면

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

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 서브넷을 삭제하려면

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

VPC에서 IPv4 CIDR 블록의 연결을 해제

VPC에 하나 이상의 IPv4 CIDR 블록이 연결되어 있는 경우, VPC에서 IPv4 CIDR 블록의 연결을 해제할 수 있습니다. 기본 IPv4 CIDR 블록의 연결을 해제할 수 없습니다. 전체 CIDR 블록의 연결을 해제하는 것만 가능하며, CIDR 블록의 서브넷이나 병합된 CIDR 블록 범위의 연결을 해제할 수 없습니다. 먼저 CIDR 블록에서 모든 서브넷을 삭제해야 합니다.

콘솔을 사용하여 VPC에서 CIDR 블록을 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다.
3. VPC를 선택하고 [Actions]와 [Edit CIDRs]를 차례로 선택합니다.
4. [VPC IPv4 CIDRs]에서 제거하려는 CIDR 블록에 대해 삭제 단추(x 아이콘)를 선택합니다.
5. 닫기를 선택합니다.

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 VPC에서 IPv4 CIDR 블록을 제거하려면

- [disassociate-vpc-cidr-block](#)(AWS CLI)

- [Unregister-EC2VpcCidrBlock](#)(Windows PowerShell용 AWS 도구)

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 정보는 [인스턴스에 할당된 IPv6 주소 해제 \(p. 109\)](#) 단원을 참조하십시오.

콘솔을 사용하여 서브넷에 IPv6 CIDR 블록의 연결을 해제하려면

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

콘솔을 사용하여 VPC에서 IPv6 CIDR 블록의 연결을 해제하려면

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

아니면 명령줄 도구를 사용할 수 있습니다.

명령줄 도구를 사용하여 서브넷에 IPv6 CIDR 블록의 연결을 해제하려면

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

명령줄 도구를 사용하여 VPC에서 IPv6 CIDR 블록의 연결을 해제하려면

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

VPC 삭제

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

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 연결 및 게이트웨이를 유지하는 것이 좋습니다. 삭제하면 새 연결을 만든 후 네트워크 관리자가 고객 게이트웨이를 다시 구성해야 합니다.

콘솔을 사용하여 VPC를 삭제하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. VPC에서 모든 인스턴스를 종료합니다. 자세한 정보는 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 VPCs]를 선택합니다.
5. 삭제할 VPC를 선택하고 Actions, Delete VPC를 선택합니다.
6. 연결을 삭제하려면 해당 옵션을 선택합니다. 삭제하지 않으려면 선택하지 않은 채로 둡니다. 예, 삭제를 선택합니다.

아니면 명령줄 도구를 사용할 수 있습니다. 명령줄을 사용하여 VPC를 삭제할 경우 먼저 모든 인스턴스를 종료하고 모든 서브넷, 사용자 지정 보안 그룹 및 사용자 지정 라우팅 테이블을 삭제한 다음, VPC에서 모든 인터넷 게이트웨이를 분리합니다.

명령줄을 사용하여 VPC를 삭제하려면

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

공유된 VPC에 대한 작업

VPC 공유를 사용하면 여러 AWS 계정에서 Amazon EC2 인스턴스, Amazon Relational Database Service(RDS) 데이터베이스, Amazon Redshift 클러스터, AWS Lambda 함수 등과 같은 애플리케이션 리소스를 공유되는 중앙 관리형 Amazon Virtual Private Cloud(VPC)에 생성할 수 있습니다. 이 모델에서 VPC(소유자)를 소유하는 계정은 AWS Organizations의 동일한 조직에 속한 다른 계정(참여자)과 한 개 또는 여러 개의 서브넷을 공유합니다. 서브넷을 공유한 후 참여자는 공유된 서브넷의 해당 애플리케이션 리소스를 보고, 생성하고, 수정하고, 삭제할 수 있습니다. 참여자는 다른 참여자 또는 VPC 소유자에 속한 리소스를 보거나 수정하거나 삭제할 수 없습니다.

내용

- [공유 VPC 사전 조건](#) (p. 92)
- [서브넷 공유](#) (p. 93)
- [공유된 서브넷의 공유 해제](#) (p. 93)
- [공유 서브넷의 소유자 식별](#) (p. 94)
- [공유 서브넷 권한](#) (p. 94)
- [소유자 및 참여자에 대한 청구 및 측정](#) (p. 94)
- [공유 서브넷에 대해 지원되지 않는 서비스](#) (p. 94)
- [제한](#) (p. 95)

공유 VPC 사전 조건

해당 조직의 마스터 계정에서 리소스 공유를 활성화해야 합니다. 리소스 공유 활성화에 대한 자세한 정보는 [AWS Organizations를 사용하여 공유 활성화](#)를 참조하십시오.

서브넷 공유

기본이 아닌 서브넷을 조직의 다른 계정과 공유할 수 있습니다. 서브넷을 공유하려면 먼저 공유할 서브넷 및 서브넷을 공유하려는 AWS 계정, 조직 단위 또는 전체 조직과 리소스 공유를 생성해야 합니다. 리소스 공유 생성에 대한 자세한 정보는 [의 리소스 공유 생성](#)을 참조하십시오.

콘솔을 사용하여 서브넷을 공유하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 서브넷을 선택합니다.
3. 서브넷을 선택하고 작업, 서브넷 공유를 선택합니다.
4. 리소스 공유를 선택하고 서브넷 공유를 선택합니다.

AWS CLI를 사용하여 서브넷을 공유하려면

`create-resource-share` 명령과 `associate-resource-share` 명령을 사용합니다.

가용 영역에 서브넷 매핑

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 AWS는 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a` 가용 영역과 위치가 동일하지 않을 수 있습니다.

VPC 공유를 위해 계정에 대해 가용 영역을 조정하려면 가용 영역에 대한 고유하고 일관된 식별자인 AZ ID를 사용해야 합니다. 예를 들어, `use1-az1`은 `us-east-1` 리전의 가용 영역 중 하나입니다. 가용 영역 ID를 사용하여 다른 계정의 리소스를 기준으로 한 계정의 리소스 위치를 확인할 수 있습니다. 자세한 정보는 [에서 리소스의 AZ ID](#)를 참조하십시오.

공유된 서브넷의 공유 해제

소유자는 참여자와 공유한 서브넷을 언제든지 공유 해제할 수 있습니다. 소유자가 공유한 서브넷을 해제하면 다음 규칙이 적용됩니다.

- 기존 참여자 리소스는 공유 해제된 서브넷에서 계속 실행됩니다.
- 참여자는 공유 해제된 서브넷에 더 이상 새로운 리소스를 생성할 수 없습니다.
- 참여자는 서브넷에 있는 리소스를 수정, 기술하고 삭제할 수 있습니다.
- VPC 흐름 로그 구독에는 참여자 ENI에 속한 모든 데이터가 제외됩니다. 참여자는 ENI 흐름 로그를 생성할지 또는 삭제할지를 결정할 수 있습니다.
- 참여자가 공유 해제된 서브넷의 리소스를 여전히 가지고 있을 경우 소유자는 공유 서브넷 또는 공유 서브넷 VPC를 삭제할 수 없습니다. 참여자가 공유 해제된 서브넷의 모든 리소스를 삭제한 후에만 소유자는 서브넷 또는 공유 서브넷 VPC를 삭제할 수 있습니다.

콘솔을 사용하여 서브넷을 공유 해제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 서브넷을 선택합니다.
3. 서브넷을 선택하고 작업, 서브넷 공유를 선택합니다.
4. 작업, 공유 중지를 선택합니다.

AWS CLI를 사용하여 서브넷을 공유 해제하려면

`disassociate-resource-share` 명령을 사용합니다.

공유 서브넷의 소유자 식별

참여자자는 공유된 서브넷을 Amazon VPC 콘솔이나 명령줄 도구를 사용하여 볼 수 있습니다.

서브넷 소유자를 식별하려면(콘솔)

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 서브넷을 선택합니다. 소유자 열에 서브넷 소유자가 표시됩니다.

AWS CLI를 사용하여 서브넷 소유자를 식별하려면

`describe-subnets` 명령과 `describe-vpcs` 명령을 사용합니다. 출력에 소유자의 ID가 포함됩니다.

공유 서브넷 권한

소유자 권한

VPC 소유자는 서브넷, 경로 테이블, 네트워크 ACL, 피어링 연결, VPC 엔드포인트, PrivateLink 엔드포인트, 인터넷 게이트웨이, NAT 게이트웨이, 가상 프라이빗 게이트웨이 및 전송 게이트웨이 연결을 포함하여, VPC 수준의 모든 리소스를 생성, 관리 및 삭제할 책임이 있습니다.

VPC 소유자는 참여자가 생성한 보안 그룹을 비롯하여, 참여자 리소스를 수정하거나 삭제할 수 없습니다. VPC 소유자는 문제 해결 및 감사를 위해 모든 네트워크 인터페이스 및 참여자 리소스에 연결된 보안 그룹에 대한 세부 정보를 볼 수 있습니다. VPC 소유자는 트래픽 모니터링 또는 문제 해결을 위해 VPC, 서브넷 또는 ENI 수준에서 흐름 로그 구독을 생성할 수 있습니다.

참여자 권한

공유 VPC의 참여자는 Amazon EC2 인스턴스, Amazon RDS 데이터베이스, 로드 밸런서를 비롯하여 해당 리소스에 대한 생성, 관리 및 삭제를 책임집니다. 참여자는 다른 참여자 계정에 속한 리소스를 보거나 수정할 수 없습니다. 참여자는 라우팅 테이블 및 공유된 서브넷에 연결된 네트워크 ACL의 세부 정보를 볼 수 있습니다. 하지만 라우팅 테이블, 네트워크 ACL, 서브넷 등을 포함하여 VPC 수준 리소스를 수정할 수 없습니다. 참여자는 보안 그룹 ID를 사용하여 다른 참여자 또는 소유자에게 속한 보안 그룹을 참조할 수 있습니다. 참여자는 자신이 소유한 인터페이스에 대해서만 흐름 로그 구독을 생성할 수 있습니다.

소유자 및 참여자에 대한 청구 및 측정

공유 VPC에서 각 참여자는 Amazon EC2 인스턴스, Amazon Relational Database Service 데이터베이스, Amazon Redshift 클러스터, AWS Lambda 함수를 비롯하여 해당 애플리케이션 리소스에 대한 요금을 지불합니다. 또한 참여자는 가용 영역 간 데이터 전송, VPC 피어링 연결을 통한 데이터 전송, AWS Direct Connect 게이트웨이를 통한 데이터 전송과 연결된 데이터 전송 요금을 지불합니다. VPC 소유자는 NAT 게이트웨이, 가상 프라이빗 게이트웨이, 전송 게이트웨이, PrivateLink 및 VPC 엔드포인트에서의 데이터 처리 및 데이터 전송 요금을 시간당 요금(해당하는 경우)으로 지불합니다. 동일한 가용 영역(AZ-ID를 사용하여 고유하게 식별됨) 내에서의 데이터 전송은 통신 리소스의 계정 소유권과 상관없이 무료입니다.

공유 서브넷에 대해 지원되지 않는 서비스

참여자자는 공유 서브넷을 다음 서비스와 공유할 수 없습니다.

- Amazon Aurora Serverless
- AWS CloudHSM
- AWS Glue
- Amazon EMR

- Network Load Balancer

제한

다음 제한은 VPC 공유를 통한 작업에 적용됩니다.

- 소유자는 AWS Organizations의 동일한 조직에 있는 다른 조직 단위 또는 계정하고만 서브넷을 공유할 수 있습니다.
- 소유자는 기본 VPC에 있는 서브넷을 공유할 수 없습니다.
- 참여자는 다른 참여자 또는 소유자가 소유한 보안 그룹을 사용하여 리소스를 시작할 수 없습니다.
- 참여자는 VPC의 기본 보안 그룹을 사용하여 리소스를 시작할 수 없습니다. 그러한 그룹은 소유자에게 속해 있기 때문입니다.
- 서비스 제한은 개별 계정별로 적용됩니다. 서비스 제한에 대한 자세한 정보는 Amazon Web Services 일반 참조에서 [AWS 서비스 제한](#)을 참조하십시오.

기본 VPC 및 기본 서브넷

2013년 12월 4일 이후에 AWS 계정을 만든 경우에는 EC2-VPC만 지원됩니다. 이 경우 각 AWS 리전에 기본 VPC가 있습니다. 기본 VPC는 바로 사용할 수 있는 상태이므로 VPC를 직접 생성하고 구성할 필요가 없습니다. 기본 VPC로 Amazon EC2 인스턴스를 즉시 시작할 수 있습니다. 기본 VPC에서 Elastic Load Balancing, Amazon RDS, Amazon EMR 등의 서비스를 사용할 수도 있습니다.

기본 VPC는 준비 과정 없이 빠르게 시작하여 블로그나 간단한 웹 사이트 같은 퍼블릭 인스턴스를 시작하는 데 적합합니다. 기본 VPC의 구성 요소를 필요에 따라 수정할 수 있습니다. 기본 VPC를 사용하지 않고, 원하는 CIDR 블록 범위 및 서브넷 크기 등과 같은 특정 요구 사항에 적합한 VPC를 만들려면 [예제 시나리오 \(p. 23\)](#)를 참조하십시오.

내용

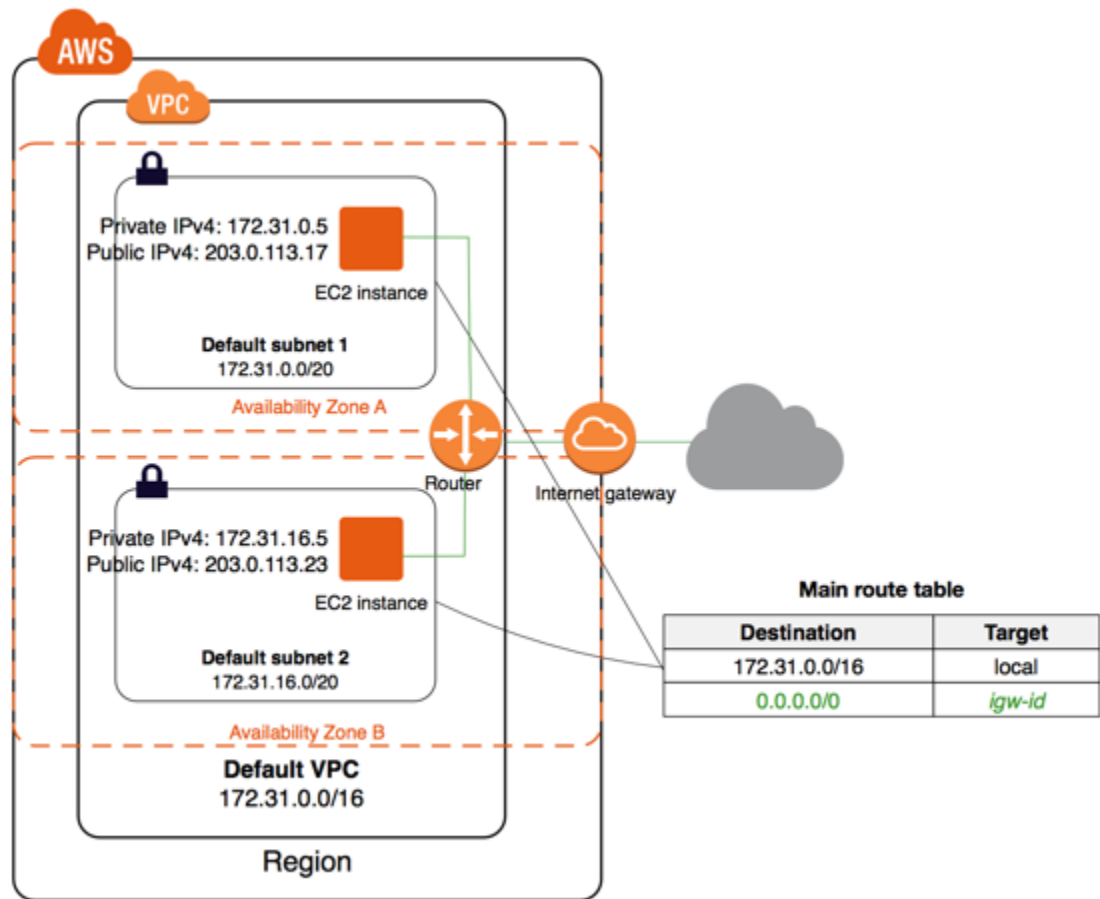
- [기본 VPC 구성 요소 \(p. 96\)](#)
- [가용성 및 지원되는 플랫폼 \(p. 98\)](#)
- [기본 VPC와 기본 서브넷 보기 \(p. 99\)](#)
- [기본 VPC로 EC2 인스턴스 시작 \(p. 99\)](#)
- [기본 서브넷과 기본 VPC 삭제 \(p. 100\)](#)
- [기본 VPC 만들기 \(p. 100\)](#)
- [기본 서브넷 생성 \(p. 101\)](#)

기본 VPC 구성 요소

기본 VPC는 다음과 같이 생성됩니다.

- IPv4 CIDR 블록의 크기가 /16인 VPC를 만듭니다 (172.31.0.0/16). 이는 최대 65,536개의 프라이빗 IPv4 주소를 제공합니다.
- 각 가용 영역에 크기 /20의 기본 서브넷을 생성합니다. 이렇게 하면 서브넷당 최대 4,096개의 주소가 제공되며, 그중 몇 개는 내부용으로 예약되어 있습니다.
- [인터넷 게이트웨이 \(p. 202\)](#)를 만들어 기본 VPC에 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(엑세스 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 VPC는 다른 일반 VPC와 동일한 방식으로 사용할 수 있습니다.

- 기본 서브넷이 아닌 서브넷을 추가합니다.
- 기본 라우팅 테이블을 수정합니다.
- 라우팅 테이블을 추가합니다.
- 추가 보안 그룹을 연결합니다.
- 기본 보안 그룹의 규칙을 업데이트합니다.
- 연결을 추가합니다.
- 더 많은 IPv4 CIDR 블록을 추가합니다.

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

IPv6 CIDR 블록을 기본 VPC에 연결할 수도 있습니다. 자세한 정보는 [VPC 및 서브넷 관련 작업 \(p. 86\)](#) 단원을 참조하십시오.

기본 서브넷

기본 라우팅 테이블은 인터넷으로 대상 주소가 정해진 서브넷의 트래픽을 인터넷 게이트웨이로 전송하기 때문에 기본적으로 기본 서브넷은 퍼블릭 서브넷입니다. 대상 주소 0.0.0.0/0에서 인터넷 게이트웨이로 라우

팅을 제거함으로써 기본 서브넷을 프라이빗 서브넷으로 만들 수 있습니다. 하지만 이렇게 하면 해당 서브넷에서 실행하는 EC2 인스턴스는 인터넷에 액세스할 수 없습니다.

기본 서브넷에서 시작한 인스턴스는 퍼블릭 IPv4 주소와 프라이빗 IPv4 주소, 퍼블릭 DNS 호스트 이름과 프라이빗 DNS 호스트 이름을 둘 다 받습니다. 기본 VPC의 기본이 아닌 서브넷에서 시작하는 인스턴스는 퍼블릭 IPv4 주소나 DNS 호스트 이름을 수신하지 않습니다. 서브넷의 퍼블릭 IP 주소 지정 동작은 변경할 수 있습니다. 자세한 정보는 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 106\)](#) 단원을 참조하십시오.

때때로 AWS는 새로운 가용 영역을 리전에 추가할 수 있습니다. 대부분의 경우 며칠 내로 기본 VPC에 대한 이 가용 영역에 새로운 기본 서브넷이 자동으로 생성됩니다. 하지만 기본 VPC를 수정했을 경우에는 새로운 기본 서브넷이 추가되지 않습니다. 원한다면, 새로운 가용 영역에 대한 기본 서브넷을 직접 생성할 수 있습니다. 자세한 정보는 [기본 서브넷 생성 \(p. 101\)](#) 단원을 참조하십시오.

가용성 및 지원되는 플랫폼

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. 사용자가 기본이 아닌 VPC를 만들어 새로운 리전에서 인스턴스를 시작할 때 지정하지 않는 한, 인스턴스는 해당 리전의 기본 VPC로 시작됩니다. 하지만 이전에 사용했던 리전에서 인스턴스를 시작하면 인스턴스가 EC2-Classic에서 시작됩니다.

AWS 계정을 2013년 3월 18일-2013년 12월 4일 사이에 만든 경우 EC2-VPC만 지원됩니다. 또는 사용했던 일부 리전에서 EC2-Classic과 EC2-VPC를 둘 다 지원할 수 있습니다. AWS 계정에 대해 각 리전에서 지원되는 플랫폼을 확인하는 방법은 [지원되는 플랫폼 및 기본 VPC 보유 여부 확인 \(p. 98\)](#) 단원을 참조하십시오. 각 리전이 기본 VPC를 사용하도록 설정된 시점에 대한 자세한 정보는 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.

귀하의 AWS 계정이 EC2-Classic과 EC2-VPC를 둘 다 지원할 경우, AWS 계정을 새로 만들거나, 이전에 사용하지 않았던 리전에서 인스턴스를 시작할 수 있습니다. EC2-VPC를 활용하면서 EC2-Classic에서 인스턴스를 시작하는 간편함을 누리기 위해 이렇게 할 수 있습니다. 기본 VPC가 없으며 EC2-Classic을 지원하는 리전에 기본 VPC를 추가하기를 계속해서 선호하는 경우, [기본 VPC FAQ](#)에서 "기존 EC2 계정에 기본 VPC를 추가하고 싶습니다. 가능합니까?"를 참조하십시오.

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

지원되는 플랫폼 및 기본 VPC 보유 여부 확인

Amazon EC2 콘솔이나 명령줄을 사용하여 AWS 계정이 모든 플랫폼을 지원하는지 또는 기본 VPC가 있는지를 확인할 수 있습니다.

Amazon EC2 콘솔을 사용하여 플랫폼 지원을 확인하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 상단 오른쪽의 리전 선택기를 사용하여 리전을 선택합니다.
3. 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.

Supported Platforms

VPC

Default VPC

vpc-1a2b3c4d

기본 VPC를 삭제하면 [Default VPC] 값이 [None]으로 표시됩니다. 자세한 정보는 [기본 서브넷과 기본 VPC 삭제 \(p. 100\)](#) 단원을 참조하십시오.

명령줄을 사용하여 플랫폼 지원을 확인하려면

- [describe-account-attributes](#)(AWS CLI)
- [Get-EC2AccountAttributes](#)(Windows PowerShell용 AWS 도구)

출력에서 supported-platforms 속성은 EC2 인스턴스를 시작할 수 있는 플랫폼을 나타냅니다.

기본 VPC와 기본 서브넷 보기

Amazon VPC 콘솔이나 명령줄을 사용하여 기본 VPC와 서브넷을 볼 수 있습니다.

Amazon VPC 콘솔을 사용하여 기본 VPC와 서브넷을 보려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다.
3. [Default VPC] 열에서 [Yes] 값을 확인합니다. 기본 VPC의 ID를 메모합니다.
4. 탐색 창에서 서브넷을 선택합니다.
5. 검색줄에 기본 VPC의 ID를 입력합니다. 검색 결과에 표시되는 서브넷이 기본 VPC의 서브넷입니다.
6. 어느 서브넷이 기본 서브넷인지 확인하려면 [Default Subnet] 열에서 [Yes] 값을 확인합니다.

명령줄을 사용하여 기본 VPC를 나타내려면

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

isDefault 필터를 포함하여 명령을 사용하고 필터 값을 true로 설정합니다.

명령줄을 사용하여 기본 서브넷을 나타내려면

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

vpc-id 필터를 포함하여 명령을 사용하고 필터 값을 기본 VPC의 ID로 설정합니다. 기본 서브넷은 출력에서 DefaultForAz 필드가 true로 설정되어 있습니다.

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 기

본 서브넷을 선택하거나, AWS 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 단원에서 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-Classical로 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 정보는 [VPC 만들기 \(p. 86\)](#) 단원을 참조하십시오.

기본 VPC를 삭제한 경우 새로 만들 수 있습니다. 자세한 정보는 [기본 VPC 만들기 \(p. 100\)](#) 단원을 참조하십시오.

기본 서브넷을 삭제한 경우 새로 만들 수 있습니다. 자세한 정보는 [기본 서브넷 생성 \(p. 101\)](#) 단원을 참조하십시오. 대신 기본 VPC에 기본이 아닌 서브넷을 만들고 AWS Support에 연락하여 해당 서브넷을 기본 서브넷으로 설정하도록 요청하십시오. 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 정보는 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 106\)](#) 단원을 참조하십시오. 가용 영역당 기본 서브넷은 한 개만 가질 수 있습니다. 기본 VPC가 아닌 VPC에는 기본 서브넷을 생성할 수 없습니다.

기본 VPC 만들기

기본 VPC를 삭제한 경우 새로 만들 수 있습니다. 기본 VPC를 삭제하면 복구할 수 없으며, 기본 VPC가 아닌 기존 VPC를 기본 VPC로 설정할 수도 없습니다. 귀하의 계정에서 EC2-Classical을 지원할 경우 이 절차를 사용하여 EC2-Classical을 지원하는 리전에 기본 VPC를 만들 수 없습니다.

기본 VPC를 만들면 각 가용 영역의 기본 서브넷을 비롯하여 기본 VPC의 표준 [구성 요소 \(p. 96\)](#) 와 함께 생성됩니다. 구성 요소를 직접 지정할 수 없습니다. 새로운 기본 VPC의 서브넷 CIDR 블록은 기존 기본 VPC와 동일한 가용 영역에 매핑되지 않을 수 있습니다. 예를 들어 기존 기본 VPC에서 CIDR 블록 172.31.0.0/20을 포함하는 서브넷이 us-east-2a에 생성된 경우, 새로운 기본 VPC에서는 us-east-2b에 생성될 수 있습니다.

리전에 기본 VPC가 이미 있으면 다른 기본 VPC를 만들 수 없습니다.

Amazon VPC 콘솔을 사용하여 기본 VPC를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 선택합니다.
3. [Actions], [Create Default VPC]를 선택합니다.
4. Create를 선택합니다. 확인 화면을 닫습니다.

명령줄을 사용하여 기본 VPC를 만들려면

- [create-default-vpc](#) AWS CLI 명령을 사용할 수 있습니다. 이 명령에는 입력 파라미터가 없습니다.

```
aws ec2 create-default-vpc
```

```
{
  "Vpc": {
    "VpcId": "vpc-3f139646",
    "InstanceTenancy": "default",
    "Tags": [],
    "Ipv6CidrBlockAssociationSet": [],
    "State": "pending",
    "DhcpOptionsId": "dopt-61079b07",
    "CidrBlock": "172.31.0.0/16",
    "IsDefault": true
  }
}
```

또는 [New-EC2DefaultVpc](#) Windows PowerShell용 도구 명령이나 [CreateDefaultVpc](#) Amazon EC2 API 작업을 사용할 수 있습니다.

기본 서브넷 생성

기본 서브넷이 없는 가용 영역에서 기본 서브넷을 생성할 수 있습니다. 예를 들어 기본 서브넷을 삭제한 경우 이를 생성하고자 할 수 있습니다. 또는 AWS가 새 가용 영역을 추가했지만 기본 VPC에서 해당 영역에 대한 기본 서브넷을 자동적으로 생성하지 않은 경우가 있습니다.

기본 서브넷을 생성할 때, 기본 VPC의 다음 사용 가능한 연속 공간에 IPv4 CIDR 블록의 크기가 /20인 서브넷이 생성됩니다. 다음 규칙이 적용됩니다.

- CIDR 블록을 직접 지정할 수 없습니다.
- 삭제한 이전 기본 서브넷을 복원할 수 없습니다.
- 가용 영역당 기본 서브넷은 한 개만 가질 수 있습니다.
- 기본 VPC가 아닌 VPC에는 기본 서브넷을 생성할 수 없습니다.

CIDR 블록 크기 /20을 생성할 충분한 주소 공간이 기본 VPC에 없는 경우 요청은 실패합니다. 더 많은 주소 공간이 필요한 경우 [VPC에 IPv4 CIDR 블록을 추가 \(p. 81\)](#)할 수 있습니다.

기본 VPC에 IPv6 CIDR 블록을 연결한 경우 새로운 기본 서브넷이 자동적으로 IPv6 CIDR 블록을 수신하지 않습니다. 대신 기본 서브넷을 생성한 다음 이에 IPv6 CIDR 블록을 연결할 수 있습니다. 자세한 정보는 [IPv6 CIDR 블록을 서브넷에 연결 \(p. 89\)](#) 단원을 참조하십시오.

현재는 AWS CLI, AWS SDK 또는 Amazon EC2 API만을 사용하여 기본 서브넷을 생성할 수 있습니다.

명령줄을 사용하여 기본 서브넷을 생성하려면

- [create-default-subnet](#) AWS CLI 명령을 사용하고 서브넷을 생성할 가용 영역을 지정합니다.

```
aws ec2 create-default-subnet --availability-zone us-east-2a
```

```
{
  "Subnet": {
    "AvailabilityZone": "us-east-2a",
    "Tags": [],
    "AvailableIpAddressCount": 4091,
    "DefaultForAz": true,
    "Ipv6CidrBlockAssociationSet": [],
    "VpcId": "vpc-1a2b3c4d",
    "State": "available",
    "MapPublicIpOnLaunch": true,
    "SubnetId": "subnet-1122aabb",
    "CidrBlock": "172.31.32.0/20",
    "AssignIpv6AddressOnCreation": false
  }
}
```

또는 [CreateDefaultSubnet](#) Amazon EC2 API 작업을 사용할 수 있습니다.

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 게이트웨이도 연결해야 합니다. 자세한 정보는 [인터넷 게이트웨이 \(p. 202\)](#) 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 최대 3자리 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.
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.	모든 현재 세대 인스턴스 유형과 C3, R3 및 I2 이전 세대 인스턴스 유형에서 지원됩니다. 자세한 정보는 인스턴스 유형 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.

IPv4	IPv6
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, 인스턴스를 수동으로 구성 (p. 116) 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.
연결 및 고객 게이트웨이, NAT 디바이스 및 VPC 엔드포인트 지원.	연결 및 고객 게이트웨이, NAT 디바이스 및 VPC 엔드포인트 지원 안 함.

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

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 정보는 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 상관 없이 공개적으로 라우팅 가능한 CIDR 블록을 포함해 VPC의 CIDR 블록에서 인터넷으로 직접 액세스하는 것은 지원하지 않습니다. 인터넷 게이트웨이, 가상 프라이빗 게이트웨이, 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.

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

- 서브넷의 퍼블릭 IP 주소 지정 속성 수정. 자세한 정보는 [서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 \(p. 106\)](#) 단원을 참조하십시오.
- 인스턴스를 시작하는 동안 퍼블릭 IP 주소 지정 기능을 활성화하거나 비활성화하면 서브넷의 퍼블릭 IP 주소 지정 속성을 재정의합니다. 자세한 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 107\)](#) 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 정보는 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 정보는 [탄력적 IP 주소 \(p. 243\)](#) 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 정보는 [VPC와 함께 DNS 사용 \(p. 239\)](#) 단원을 참조하십시오.

IPv6 주소

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

- [IPv6 CIDR 블록을 VPC와 연결 \(p. 88\)](#)
- [IPv6 CIDR 블록을 서브넷에 연결 \(p. 89\)](#)

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 인터페이스에 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 정보는 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 정보는 [보안 \(p. 122\)](#) 단원을 참조하십시오.

예약된 IPv6 주소 범위에 대한 자세한 정보는 [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 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정 \(p. 107\)](#) 및 [인스턴스 시작 시 IPv6 주소 배정 \(p. 108\)](#) 단원을 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.

작업

- 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 ([p. 106](#))
- 서브넷의 퍼블릭 IPv6 주소 지정 속성 수정 ([p. 106](#))
- 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 ([p. 107](#))
- 인스턴스 시작 시 IPv6 주소 배정 ([p. 108](#))
- 인스턴스에 IPv6 주소 할당 ([p. 108](#))
- 인스턴스에 할당된 IPv6 주소 해제 ([p. 109](#))
- API 및 명령 개요 ([p. 109](#))

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 서브넷을 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 API의 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 서브넷을 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 정보는 [탄력적 IP 주소 \(p. 243\)](#) 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 ifconfig를 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 정보는 [인스턴스 메타데이터 및 사용자 데이터](#)를 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 정보는 [탄력적 IP 주소 \(p. 243\)](#) 단원을 참조하십시오.

인스턴스 시작 시 IPv6 주소 배정

시작하는 과정에서 인스턴스에 IPv6 주소를 자동 할당할 수 있습니다. 이를 위해서는 [연결된 IPv6 CIDR 블록 \(p. 88\)](#)이 있는 VPC 및 서브넷으로 인스턴스를 시작해야 합니다. IPv6 주소는 서브넷의 범위로부터 할당 되고, 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 단원으로 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [다중 IPv6 주소 작업](#) 단원을 참조하십시오.

인스턴스에 IPv6 주소 할당

인스턴스가 [연결된 IPv6 CIDR 블록 \(p. 88\)](#)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 [Instances]를 선택하고 인스턴스를 선택합니다.

3. 작업을 선택하고, 네트워킹, IP 주소 관리를 선택합니다.
4. [IPv6 Addresses]에서 [Assign new IP]를 선택합니다. 서브넷 범위에 속한 IPv6 주소를 지정하거나, Amazon이 IPv6 주소를 자동으로 선택하도록 자동 할당 값을 그대로 둡니다.
5. Save를 선택합니다.

또는 네트워크 인터페이스에 IPv6 주소 한 개를 할당할 수 있습니다. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스 주제에서 [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 [Instances]를 선택하고 인스턴스를 선택합니다.
3. 작업을 선택하고, 네트워킹, IP 주소 관리를 선택합니다.
4. IPv6 주소에서 해당 IPv6 주소에 대해 할당 해제를 선택합니다.
5. Save를 선택합니다.

또는 네트워크 인터페이스에 연결된 IPv6 주소를 해제할 수 있습니다. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스 주제에서 [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 정보는 [Amazon VPC에 액세스 \(p. 7\)](#) 단원을 참조하십시오.

시작 시 퍼블릭 IPv4 주소 배정

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

시작 시 IPv6 주소 배정

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

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 모드—로 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.

Note

이 설명에서는 퍼블릭 및 프라이빗 서브넷을 포함하는 VPC를 보유하고 있다고 가정합니다. IPv6와 함께 사용할 새 VPC 설정에 대한 자세한 내용은 [Amazon VPC용 IPv6 시작하기 \(p. 16\)](#) 단원을 참조하십시오.

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

단계	참고
1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결 (p. 113)	Amazon 제공 IPv6 CIDR 블록을 VPC 및 서브넷에 연결합니다.
2단계: 라우팅 테이블 업데이트 (p. 114)	라우팅 테이블을 업데이트하여 IPv6 트래픽을 라우팅합니다. 퍼블릭 서브넷의 경우, 서브넷의 모든 IPv6 트래픽을 인터넷 게이트웨이로 라우팅하는 경로를 생성합니다. 프라이빗 서브넷의 경우, 서브넷의 인터넷 바인딩된 모든 IPv6 트래픽을 외부 전용 인터넷 게이트웨이로 라우팅하는 경로를 생성합니다.
3단계: 보안 그룹 규칙 업데이트 (p. 114)	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.
4단계: 인스턴스 유형 변경 (p. 115)	인스턴스 유형이 IPv6를 지원하지 않은 경우, 인스턴스 유형을 변경합니다.
5단계: 인스턴스에 IPv6 주소 할당 (p. 116)	서브넷의 IPv6 주소 범위에서 인스턴스에 IPv6 주소를 할당합니다.
6단계: (선택 사항) 인스턴스에서 IPv6 구성하기 (p. 116)	DHCPv6를 사용하도록 구성되지 않은 AMI에서 인스턴스를 시작하는 경우, 이 인스턴스에 지정된 IPv6 주소를 인식하도록 인스턴스를 수동으로 구성해야 합니다.

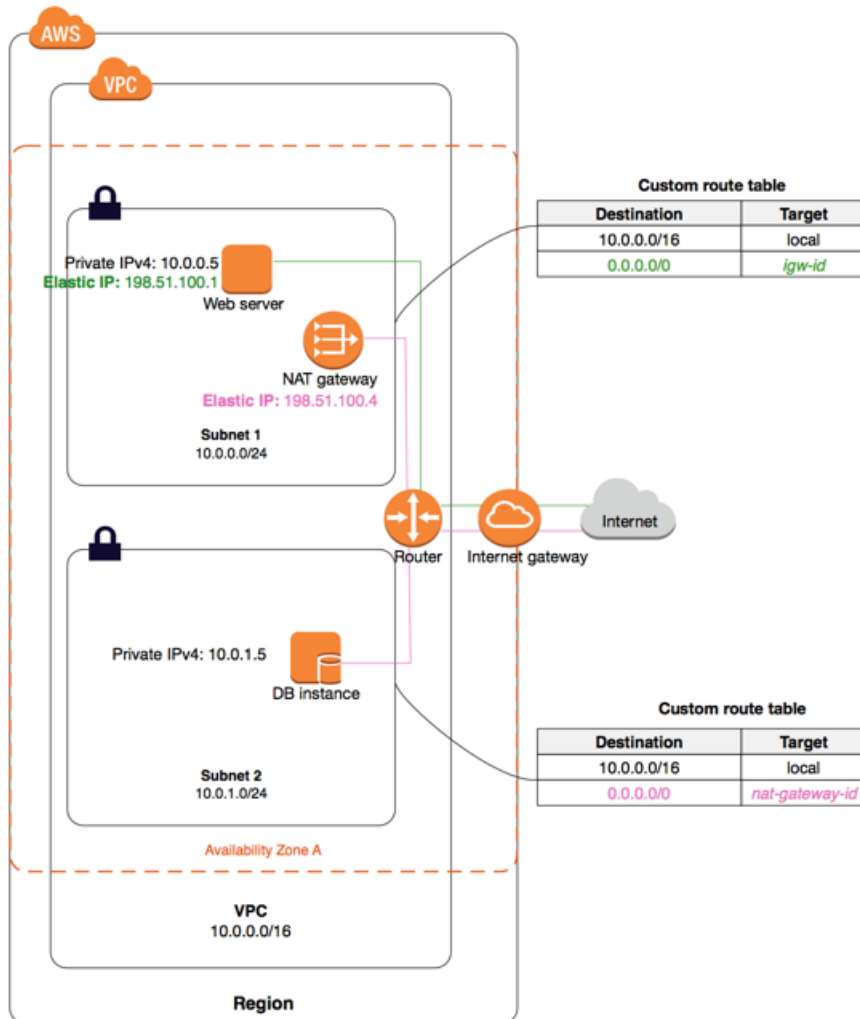
IPv6를 사용하여 마이그레이션하기 전에 Amazon VPC에 대한 IPv6 주소 지정 기능을 읽어 보시기 바랍니다. [IPv4 및 IPv6의 특징과 제한 \(p. 103\)](#).

콘텐츠

- 예: 퍼블릭 및 프라이빗 서브넷이 있는 VPC에서 IPv6 사용 (p. 111)
- 1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결 (p. 113)
- 2단계: 라우팅 테이블 업데이트 (p. 114)
- 3단계: 보안 그룹 규칙 업데이트 (p. 114)
- 4단계: 인스턴스 유형 변경 (p. 115)
- 5단계: 인스턴스에 IPv6 주소 할당 (p. 116)
- 6단계: (선택 사항) 인스턴스에서 IPv6 구성하기 (p. 116)

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.

유형	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
HTTP	TCP	80	0.0.0.0/0	HTTP를 통해 인터넷에서 오는 인바운드 트래픽을 허용합니다.
HTTPS	TCP	443	0.0.0.0/0	HTTPS를 통해 인터넷에서 오는 인바운드 트래픽을 허용합니다.
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.

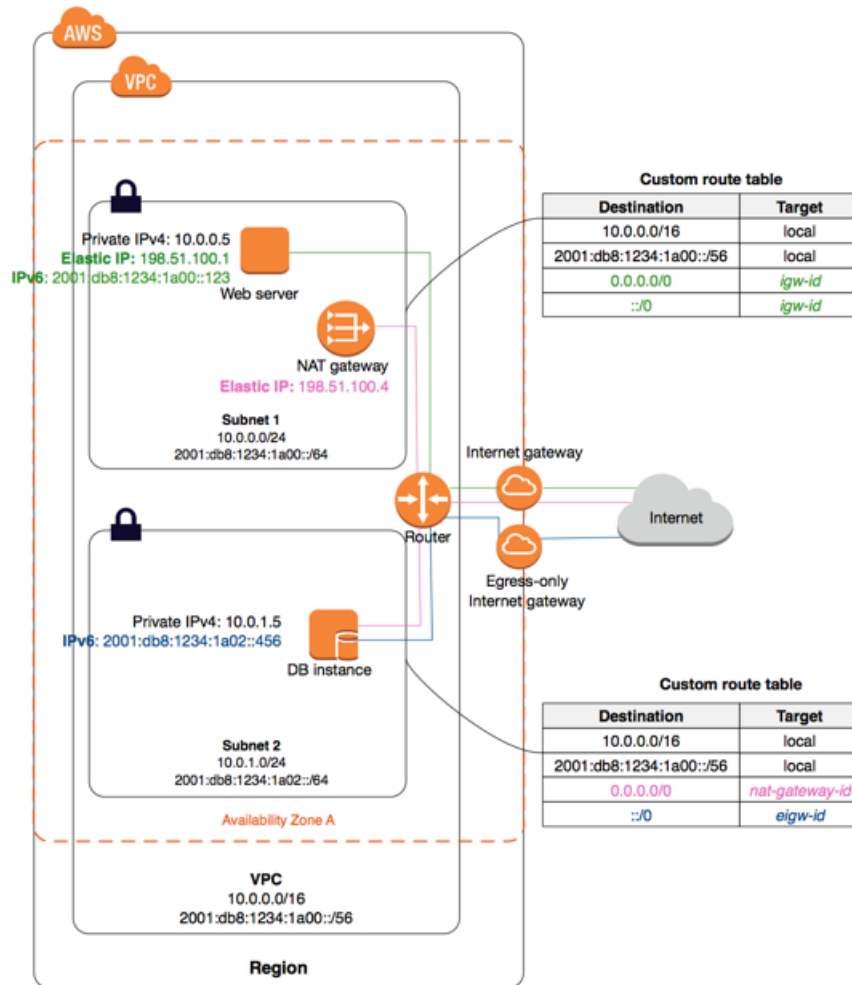
유형	프로토콜	포트 범위	소스	Comment
MySQL	TCP	3306	sg-11aa22bb	sg-11aa22bb(웹 서버 인스턴스)와 연결된 인스턴스에서 나오는 MySQL 트래픽에 대한 인바운드 액세스를 허용

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.large입니다.

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 VPCs]를 선택합니다.
3. VPC를 선택하고 Actions, Edit CIDRs를 선택합니다.
4. Add IPv6 CIDR을 선택합니다. IPv6 CIDR 블록을 추가한 후 닫기를 선택합니다.

IPv6 CIDR 블록을 서브넷에 연결하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Subnets]를 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. 닫기를 선택합니다. VPC의 다른 서브넷에 대해 이 절차를 반복합니다.

자세한 내용은 [IPv6의 경우, VPC 및 서브넷 크기 조정 \(p. 84\)](#) 단원을 참조하십시오.

2단계: 라우팅 테이블 업데이트

퍼블릭 서브넷의 경우, 라우팅 테이블을 업데이트하여 인스턴스(웹 서버 등)가 IPv6 트래픽을 위한 인터넷 게이트웨이를 사용할 수 있도록 해야 합니다.

프라이빗 서브넷의 경우, 라우팅 테이블을 업데이트하여 인스턴스(데이터베이스 인스턴스 등)가 IPv6 트래픽을 위한 외부 전용 인터넷 게이트웨이를 사용할 수 있도록 해야 합니다.

퍼블릭 서브넷의 라우팅 테이블을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Route Tables]를 선택하고 퍼블릭 서브넷에 연결된 라우팅 테이블을 선택합니다.
3. [Routes] 탭에서 [Edit]를 선택합니다.
4. [Add another route]를 선택합니다. 대상 주소에 `::/0`을 지정하고 대상에 인터넷 게이트웨이 ID를 선택한 후 저장을 선택합니다.

프라이빗 서브넷의 라우팅 테이블을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 프라이빗 서브넷에 NAT 디바이스를 사용 중인 경우, IPv6 트래픽을 지원하지 않습니다. 대신, 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. 자세한 내용은 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 단원을 참조하십시오.
3. 탐색 창에서 Route Tables를 선택하고 프라이빗 서브넷에 연결된 라우팅 테이블을 선택합니다.
4. [Routes] 탭에서 [Edit]를 선택합니다.
5. [Add another route]를 선택합니다. [Destination]에 `::/0`을 지정합니다. 대상에 외부 전용 인터넷 게이트웨이 ID를 선택한 후 저장을 선택합니다.

자세한 내용은 [라우팅 옵션 \(p. 194\)](#) 단원을 참조하십시오.

3단계: 보안 그룹 규칙 업데이트

인스턴스가 IPv6를 통해 트래픽을 보내고 받을 수 있도록 하려면 보안 그룹 규칙을 업데이트하여 IPv6 주소용 규칙을 포함해야 합니다.

예를 들어 위의 예에서, 웹 서버 보안 그룹(`sg-11aa22bb`)을 업데이트하여 IPv6 주소로부터 인바운드 HTTP, HTTPS, SSH 액세스를 허용하는 규칙을 추가할 수 있습니다. 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 rule]을 선택하고, 다 되었으면 [Save]를 선택합니다. 예를 들어, [Type]에 대해 IPv6를 통한 모든 HTTP 트래픽을 허용하는 규칙을 추가하려면 [HTTP]를 선택하고 [Source]에 `::/0`을 입력합니다.

기본적으로, 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 \(p. 123\)](#) 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. 자세한 내용은 [VPC에 권장되는 네트워크 ACL 규칙 \(p. 144\)](#) 단원을 참조하십시오.

4단계: 인스턴스 유형 변경

현재 세대의 모든 인스턴스 유형은 IPv6를 지원합니다. 자세한 내용은 [인스턴스 유형](#) 단원을 참조하십시오.

인스턴스 유형이 IPv6를 지원하지 않는 경우, 지원되는 인스턴스 유형에 맞게 인스턴스 크기를 조정해야 합니다. 위의 예에서 데이터베이스 인스턴스는 m3.large 인스턴스 유형으로서 IPv6를 지원하지 않습니다. 지원되는 인스턴스 유형에 맞게 인스턴스 크기를 조정해야 합니다(예: m4.large).

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], [Stop]을 차례로 선택합니다.
4. 확인 대화 상자가 나타나면 Yes, Stop을 선택합니다.
5. 인스턴스를 선택한 상태에서 Actions, Instance Settings, Change Instance Type를 선택합니다.
6. Instance Type에 새 인스턴스 유형을 선택한 다음 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](#) 단원을 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.

5단계: 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 인스턴스를 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, Update]를 선택합니다.

아니면 새 인스턴스를 시작하는 경우(예를 들어, 인스턴스 크기를 조정할 수 없어 그 대신에 새 AMI를 생성한 경우), 시작하는 과정에서 IPv6주소를 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.

6단계: (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](#) 및 Windows 인스턴스용 Amazon EC2 사용 설명서의 [Windows 인스턴스에 연결](#)을 참조하십시오.

운영 체제

- [Amazon Linux \(p. 116\)](#)
- [Ubuntu \(p. 117\)](#)
- [RHEL/CentOS \(p. 119\)](#)
- [Windows가 설치된 \(p. 120\)](#)

Amazon Linux

Amazon Linux에서 DHCPv6을 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.

- 다음과 같이 인스턴스에 최신 소프트웨어 패키지를 가져옵니다.

```
sudo yum update -y
```

- 선택한 텍스트 편집기를 사용하여 /etc/sysconfig/network-scripts/ifcfg-eth0를 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
```

- /etc/sysconfig/network을 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
```

- /etc/hosts를 열어 다음과 같이 내용을 교체하고 변경 내용을 저장합니다.

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

- 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.

Ubuntu Server 16

실행 중인 Ubuntu Server 16 인스턴스에서 IPv6를 구성하려면

- 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
- /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 eth0입니다. 이름은 인스턴스 유형에 따라 달라질 수 있습니다.

3. /etc/network/interfaces.d/60-default-with-ipv6.cfg 파일을 생성하고 다음 줄을 추가합니다. 필요한 경우 eth0을 위 단계에서 검색된 네트워크 인터페이스의 이름으로 대체합니다.

```
iface eth0 inet6 dhcp
```

4. 인스턴스를 재부팅하거나, 다음 명령을 실행하여 네트워크 인스턴스를 재시작합니다. 필요한 경우 eth0을 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 [시작 시 Linux 인스턴스에서 명령 실행 단원](#)을 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 인식되는지 확인합니다.

RHEL/CentOS

RHEL 7.4 및 CentOS 7 이상은 `cloud-init`를 사용해 네트워크 인터페이스를 구성하고 `/etc/sysconfig/network-scripts/ifcfg-eth0` 파일을 생성합니다. 사용자 지정 `cloud-init` 구성 파일을 만들어 DHCPv6를 활성화할 수 있습니다. 그러면 매번 재부팅 후 DHCPv6를 활성화하는 설정으로 `ifcfg-eth0` 파일이 생성됩니다.

Note

알려진 문제로 인해, `cloud-init-0.7.9`의 최신 버전과 함께 RHEL/CentOS 7.4를 사용 중인 경우, 이 단계의 결과로 재부팅 후 인스턴스에 대한 연결이 끊길 수 있습니다. 차선책으로서 `/etc/sysconfig/network-scripts/ifcfg-eth0` 파일을 수동으로 편집할 수 있습니다.

RHEL 7.4 또는 CentOS 7에서 DHCPv6를 구성하려면

1. 인스턴스의 퍼블릭 IPv4 주소를 사용하여 인스턴스에 연결합니다.
2. 원하는 텍스트 편집기를 사용하여 사용자 지정 파일을 만듭니다. 예:

```
/etc/cloud/cloud.cfg.d/99-custom-networking.cfg
```

3. 파일에 다음과 같은 줄을 추가하고 변경 내용을 저장합니다.

```
network:
  version: 1
  config:
    - type: physical
      name: eth0
      subnets:
        - type: dhcp
        - type: dhcp6
```

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

RHEL 버전 7.3 이하인 경우, 다음 절차를 이용해 `/etc/sysconfig/network-scripts/ifcfg-eth0` 파일을 직접 수정할 수 있습니다.

RHEL 7.3 이전 버전에서 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 Virtual Private Cloud는 Virtual Private Cloud(VPC)의 보안을 강화하고 모니터링하기 위해 사용할 수 있는 여러 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 \(p. 123\)](#) 단원을 참조하십시오.

보안 그룹만 사용하여 VPC 인스턴스를 보호할 수 있지만 2차 보안 계층으로 네트워크 ACL을 추가할 수 있습니다. 자세한 내용은 [네트워크 ACL \(p. 130\)](#) 단원을 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 \(p. 173\)](#) 단원을 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 \(p. 159\)](#) 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 4개를 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.

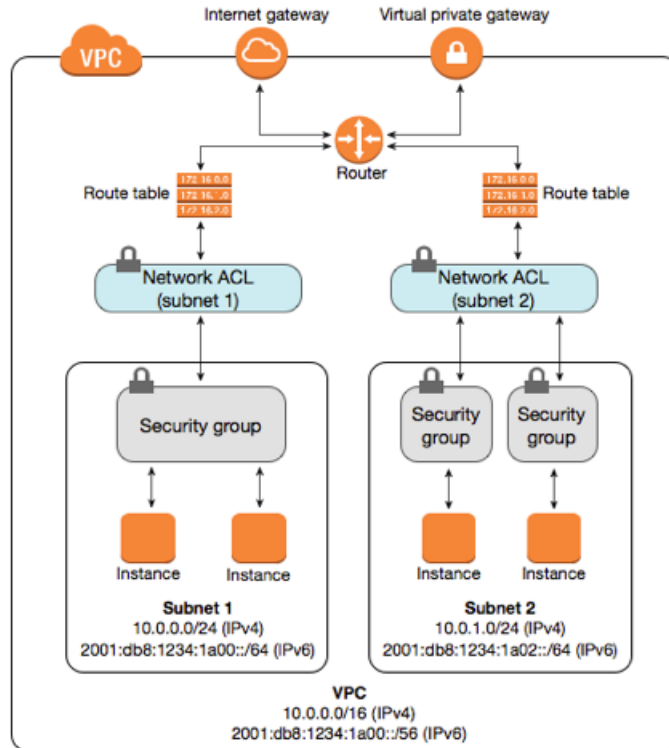
보안 그룹 및 네트워크 ACL 비교

다음 표는 보안 그룹과 네트워크 ACL의 근본적인 차이를 요약한 것입니다.

보안 그룹	네트워크 ACL
인스턴스 레벨에서 운영됩니다.	서브넷 레벨에서 운영됩니다.
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

보안 그룹	네트워크 ACL
인스턴스 시작 시 누군가 보안 그룹을 지정하거나, 나중에 보안 그룹을 인스턴스와 연결하는 경우에만 인스턴스에 적용됨	연결된 서브넷에서 모든 인스턴스에 자동 적용됨(보안 그룹을 지정하는 사용자에게 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 \(p. 122\)](#) 단원을 참조하십시오.

내용

- [보안 그룹 기본 사항](#) (p. 124)
- [VPC의 기본 보안 그룹](#) (p. 124)
- [보안 그룹 규칙](#) (p. 125)
- [EC2-Classic 및 EC2-VPC용 보안 그룹의 차이점](#) (p. 127)
- [보안 그룹 작업](#) (p. 127)

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 정보는 [Amazon VPC 제한](#) (p. 285) 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 또 다른 호스트에서 시작하여 인스턴스로 가는 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 인스턴스에서 시작하는 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 정보는 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.
- 보안 그룹을 생성할 때 이름과 설명을 제공해야 합니다. 다음 규칙이 적용됩니다.
 - 이름과 설명은 최대 255자일 수 있습니다.
 - 이름과 설명은 다음과 같은 문자로 제한됩니다. a-z, A-Z, 0-9, 공백 및 . _ : / () # , @ [] + = & ; { } ! \$ *
 - 보안 그룹 이름은 sg-로 시작할 수 없습니다.
 - 보안 그룹 이름은 VPC 내에서 고유해야 합니다.

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.

Inbound

Source	Protocol	Port Range	Comments
보안 그룹 ID(sg-xxxxxxx)	모두	모두	동일한 보안 그룹에 지정된 인스턴스의 인바운드 트래픽 허용.
Outbound			
Destination	Protocol	Port Range	Comments
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, 단일 IPv4/IPv6 주소 또는 접두사 목록 ID일 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.
- 나중에 쉽게 식별할 수 있도록 보안 그룹 규칙에 대한 설명 옵션이 제공됩니다. 설명 길이는 최대 255자입니다. 허용되는 문자는 a-z, A-Z, 0-9, 공백 및 _-:/()#,@[]+=;{}!\$*.

CIDR 블록을 규칙의 소스로 지정하면 지정된 프로토콜과 포트의 경우, 지정된 주소에서 트래픽이 허용됩니다. 보안 그룹을 규칙의 원본으로 지정하면 지정된 프로토콜과 포트의 경우, 원본 보안 그룹과 연결된 인스턴스의 탄력적 네트워크 인터페이스(ENI)에서 트래픽이 허용됩니다. 보안 그룹을 소스로 추가해도 원본 보안 그룹의 규칙이 추가되지는 않습니다.

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

Inbound			
Source	Protocol	Port Range	Comments
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 경유)
네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	네트워크의 IPv4 IP 주소로부터 Windows 인스턴스로 접근하는 인바운드 RDP 액세스 허용(인터넷 게이트웨이 경유)
Outbound			
Destination	Protocol	Port Range	Comments
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 \(p. 47\)](#)을 참조하십시오.

특정한 종류의 액세스에 대한 보안 그룹 규칙의 예는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [보안 그룹 규칙 참조](#)를 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 stale로 표시됩니다. 다른 보안 그룹 규칙과 같은 방법으로 무효 보안 그룹 규칙을 삭제할 수 있습니다.

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

EC2-Classic 및 EC2-VPC용 보안 그룹의 차이점

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. EC2-Classic에서의 보안 그룹 사용과 VPC에서의 보안 그룹 사용 간의 차이점에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [EC2-Classic과 VPC 간의 차이](#)를 참조하십시오.

보안 그룹 작업

다음 작업은 Amazon VPC 콘솔을 사용한 보안 그룹 작업 방법을 보여 줍니다.

작업

- [기본 보안 그룹 수정](#) (p. 127)
- [보안 그룹 생성](#) (p. 127)
- [규칙 추가, 제거 및 업데이트](#) (p. 128)
- [인스턴스의 보안 그룹 변경](#) (p. 129)
- [보안 그룹 삭제](#) (p. 129)
- [2009-07-15-default 보안 그룹 삭제](#) (p. 130)

기본 보안 그룹 수정

VPC에는 [기본 보안 그룹](#) (p. 124)이 포함되어 있습니다. 이 그룹을 삭제할 수 없지만 그룹의 규칙을 변경할 수는 있습니다. 다른 보안 그룹을 수정할 때와 절차는 똑같습니다. 자세한 정보는 [규칙 추가, 제거 및 업데이트](#) (p. 128) 단원을 참조하십시오.

보안 그룹 생성

인스턴스의 기본 보안 그룹을 사용할 수 있지만 직접 그룹을 생성하여 시스템에서 인스턴스에 지정되는 다른 역할을 반영할 수 있습니다.

콘솔을 사용하여 보안 그룹을 생성하려면

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.

명령줄을 사용하여 보안 그룹을 생성하려면

- [create-security-group](#)(AWS CLI)
- [New-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 하나 이상의 보안 그룹을 설명합니다.

- [describe-security-groups](#)(AWS CLI)
- [Get-EC2SecurityGroup](#)(Windows PowerShell용 AWS 도구)

기본적으로 처음에 새 보안 그룹에는 인스턴스에서 나가는 모든 트래픽을 허용하는 아웃바운드 규칙만 적용됩니다. 인바운드 트래픽을 사용하거나 아웃바운드 트래픽을 제한하려면 규칙을 추가해야 합니다.

규칙 추가, 제거 및 업데이트

규칙을 추가하거나 제거할 때 이미 보안 그룹에 할당된 인스턴스가 변경될 수 있습니다.

VPC 피어링 연결이 있는 경우, 피어 VPC의 보안 그룹을 보안 그룹 규칙의 소스 또는 대상으로 참조할 수 있습니다. 자세한 정보는 Amazon VPC Peering Guide의 [피어링된 VPC 보안 그룹을 참조하도록 보안 그룹 업데이트](#)를 참조하십시오.

콘솔을 사용하여 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 0.0.0.0/0으로 지정합니다.

Note

0.0.0.0/0을 사용하면 모든 IPv4 주소가 HTTP 또는 HTTPS를 사용하여 인스턴스에 액세스하도록 할 수 있습니다. 액세스를 제한하려면 특정 IP 주소 또는 주소 범위를 입력합니다.

5. 필요에 따라 이 규칙에 대한 설명을 입력한 다음, [Save]를 선택합니다.
6. 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.
7. 필요에 따라 [Outbound Rules] 탭을 사용하여 아웃바운드 트래픽 규칙을 추가할 수도 있습니다.

콘솔을 사용하여 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.

콘솔을 사용하여 기존 보안 그룹의 프로토콜, 포트 범위 또는 소스/대상 주소를 수정하면 콘솔은 기존 규칙을 삭제하고 새 규칙을 추가합니다.

콘솔을 사용하여 규칙을 업데이트하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 업데이트할 보안 그룹을 선택하고 인바운드 트래픽에 대한 규칙을 업데이트하려면 인바운드 규칙을, 아웃바운드 트래픽에 대한 규칙을 업데이트하려면 아웃바운드 규칙을 선택합니다.
4. [Edit]를 선택합니다. 필요에 따라 규칙 항목을 수정한 다음, [Save]를 선택합니다.

Amazon EC2 API 또는 명령줄 도구를 사용하여 기존 규칙의 프로토콜, 포트 범위 또는 소스/대상 주소를 업데이트하기 위해 규칙을 수정할 수 없습니다. 대신에 기존 규칙을 삭제하고 새 규칙을 추가해야 합니다. 규칙

설명을 업데이트하기 위해 [update-security-group-rule-descriptions-ingress](#) 및 [update-security-group-rule-descriptions-egress](#) 명령을 사용할 수 있습니다.

명령줄을 사용하여 보안 그룹에 규칙을 추가하려면 다음을 수행합니다.

- [authorize-security-group-ingress](#) 및 [authorize-security-group-egress](#)(AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) 및 [Grant-EC2SecurityGroupEgress](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 보안 그룹에서 규칙을 삭제하려면

- [revoke-security-group-ingress](#) 및 [revoke-security-group-egress](#)(AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) 및 [Revoke-EC2SecurityGroupEgress](#)(Windows PowerShell용 AWS 도구)

명령줄을 사용하여 보안 그룹 규칙에서 설명을 업데이트하려면

- [update-security-group-rule-descriptions-ingress](#) 및 [update-security-group-rule-descriptions-egress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleIngressDescription](#) 및 [Update-EC2SecurityGroupRuleEgressDescription](#) (Windows PowerShell용 AWS 도구)

인스턴스의 보안 그룹 변경

VPC에서 인스턴스를 시작한 후 해당 인스턴스와 연결된 보안 그룹을 변경할 수 있습니다. 인스턴스가 running 또는 stopped 상태에 있는 경우 해당 인스턴스의 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.

콘솔을 사용하여 인스턴스의 보안 그룹을 변경하려면

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

명령줄을 사용하여 인스턴스의 보안 그룹을 변경하려면

- [modify-instance-attribute](#)(AWS CLI)
- [Edit-EC2InstanceAttribute](#)(Windows PowerShell용 AWS 도구)

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. 129) 단원 참조). 기본 보안 그룹을 삭제할 수 없습니다.

콘솔을 사용하면 한 번에 두 개 이상의 보안 그룹을 삭제할 수 있습니다. 명령줄 또는 API를 사용하면 한 번에 한 개의 보안 그룹만 삭제할 수 있습니다.

콘솔을 사용하여 보안 그룹을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 하나 이상의 보안 그룹을 선택한 다음, [Security Group Actions], [Delete Security Group]을 선택합니다.
4. [Delete Security Group] 대화 상자에서 [Yes, Delete]를 선택합니다.

명령줄을 사용하여 보안 그룹을 삭제하려면

- `delete-security-group`(AWS CLI)
- `Remove-EC2SecurityGroup`(Windows PowerShell용 AWS 도구)

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 Interfaces]를 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.

Note

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 보안 그룹 작업, 보안 그룹 삭제를 선택합니다.
 5. [Delete Security Group] 대화 상자에서 [Yes, Delete]를 선택합니다.

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 내용은 [보안 그룹 및 네트워크 ACL 비교](#) (p. 122) 단원을 참조하십시오.

콘텐츠

- [네트워크 ACL 기본 사항 \(p. 131\)](#)
- [네트워크 ACL 규칙 \(p. 131\)](#)
- [기본 네트워크 ACL \(p. 132\)](#)
- [사용자 지정 네트워크 ACL \(p. 133\)](#)
- [휘발성 포트 \(p. 137\)](#)
- [네트워크 ACL 작업 \(p. 137\)](#)
- [예: 서브넷 내 인스턴스에 대한 액세스 제어 \(p. 140\)](#)
- [API 및 명령 개요 \(p. 143\)](#)

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 중분 방식으로(예: 10 또는 100 단위씩 중분) 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.

자세한 내용은 [Amazon VPC 제한 \(p. 285\)](#) 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 #	유형	프로토콜	포트 범위	소스	허용/거부
100	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	허용
*	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	DENY
아웃바운드					
규칙 #	유형	프로토콜	포트 범위	대상 주소	허용/거부
100	모든 IPv4 트래픽	all	all	0.0.0.0/0	허용
*	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 별표로 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 #	유형	프로토콜	포트 범위	소스	허용/거부
100	모든 IPv4 트래픽	모두	모두	0.0.0.0/0	허용
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 #	유형	프로토콜	포트 범위	대상 주소	허용/거부
100	모든 트래픽	all	all	0.0.0.0/0	허용
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 32768-65535를 포함하는 아웃바운드 규칙 120). 적절한 취발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 [취발성 포트 \(p. 137\)](#) 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 32768-65535를 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 #	유형	프로토콜	포트 범위	소스	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	HTTPS	TCP	443	0.0.0.0/0	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	SSH	TCP	22	192.0.2.0/24	허용	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터의 인바운드 SSH 트래픽을 허용(인터넷 게이트웨이를 통해)
130	RDP	TCP	3389	192.0.2.0/24	허용	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터 웹 서버로의 인바운드 RDP 트래픽을 허용(인터넷 게이트웨이를 통해)

140	사용자 지정 TCP	TCP	32768-65535	0.0.0.0/0	허용	인터넷으로부터의 인바운드 리턴 IPv4 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 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 #	유형	프로토콜	포트 범위	대상 주소	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTP 트래픽을 허용
110	HTTPS	TCP	443	0.0.0.0/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTPS 트래픽을 허용
120	사용자 지정 TCP	TCP	32768-65535	0.0.0.0/0	허용	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 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 어떠한 규칙과도 일치하지 않으며, * 규칙이 최종적으로 해당 패킷을 거부합니다.

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 사용 설명서에서 [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 수도 있습니다. IPv4 및 IPv6 트래픽은 분리되어 있습니다. 따라서 IPv4 트래픽에 대한 규칙 중 어느 것도 IPv6 트래픽에 적용되지 않습니다.

인바운드						
규칙 #	유형	프로토콜	포트 범위	소스	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	허용	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	허용	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	허용	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터의 인바운드 SSH 트래픽을 허용(인터넷 게이트웨이를 통해)
130	RDP	TCP	3389	192.0.2.0/24	허용	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터 웹 서버로의 인바운드 RDP 트래픽을 허용(인터넷 게이트웨이를 통해)
140	사용자 지정 TCP	TCP	32768-65535	0.0.0.0/0	허용	인터넷으로부터의 인바운드 리턴 IPv4 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 단원을 참조하십시오.

145	사용자 지정 TCP	TCP	32768-65535	::/0	허용	인터넷으로부터의 인바운드 리턴 IPv6 트래픽을 허용(즉, 서브넷에서 시작되는 요청에 대해). 이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 단원을 참조하십시오.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
*	모든 트래픽	모두	모두	::/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
아웃바운드						
규칙 #	유형	프로토콜	포트 범위	대상 주소	허용/거부	설명
100	HTTP	TCP	80	0.0.0.0/0	허용	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	허용	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	32768-65535	0.0.0.0/0	허용	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답을 허용(예: 서브넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공). 이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 단원을 참조하십시오.

125	사용자 지정 TCP	TCP	32768-65535::/0		허용	<p>인터넷에서 클라이언트에 대한 아웃바운드 IPv6 응답을 허용(예: 서버넷에 있는 웹 서버를 방문하는 사람들에게 웹 페이지 제공).</p> <p>이 범위는 예시일 뿐입니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 (p. 137) 단원을 참조하십시오.</p>
*	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)

휘발성 포트

이전 단원에서 예로 든 네트워크 ACL에는 32768-65535 범위의 휘발성 포트가 사용됩니다. 하지만 사용하거나 통신하는 클라이언트의 유형에 따라 다른 범위의 네트워크 ACL을 사용할 수 있습니다.

요청을 시작하는 클라이언트가 휘발성 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 휘발성 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, 휘발성 포트 1024-65535를 열 수 있습니다. 하지만 그 범위 내에 있는 악성 포트의 트래픽을 거부하기 위한 규칙을 ACL에 추가할 수도 있습니다. 표에서 광범위한 휘발성 포트를 여는 ALLOW 규칙보다 앞서 DENY 규칙을 배치해야 합니다.

네트워크 ACL 작업

다음 작업은 Amazon VPC 콘솔을 사용한 네트워크 ACL 작업 방법을 보여 줍니다.

작업

- [네트워크 ACL 연결 확인 \(p. 138\)](#)
- [네트워크 ACL 생성 \(p. 138\)](#)
- [규칙 추가 및 삭제 \(p. 138\)](#)
- [서브넷을 네트워크 ACL과 연결 \(p. 139\)](#)

- 서브넷에서 네트워크 ACL 연결 끊기 (p. 139)
- 서브넷의 네트워크 ACL 변경 (p. 140)
- 네트워크 ACL 삭제 (p. 140)

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

서브넷—에서 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. Save를 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

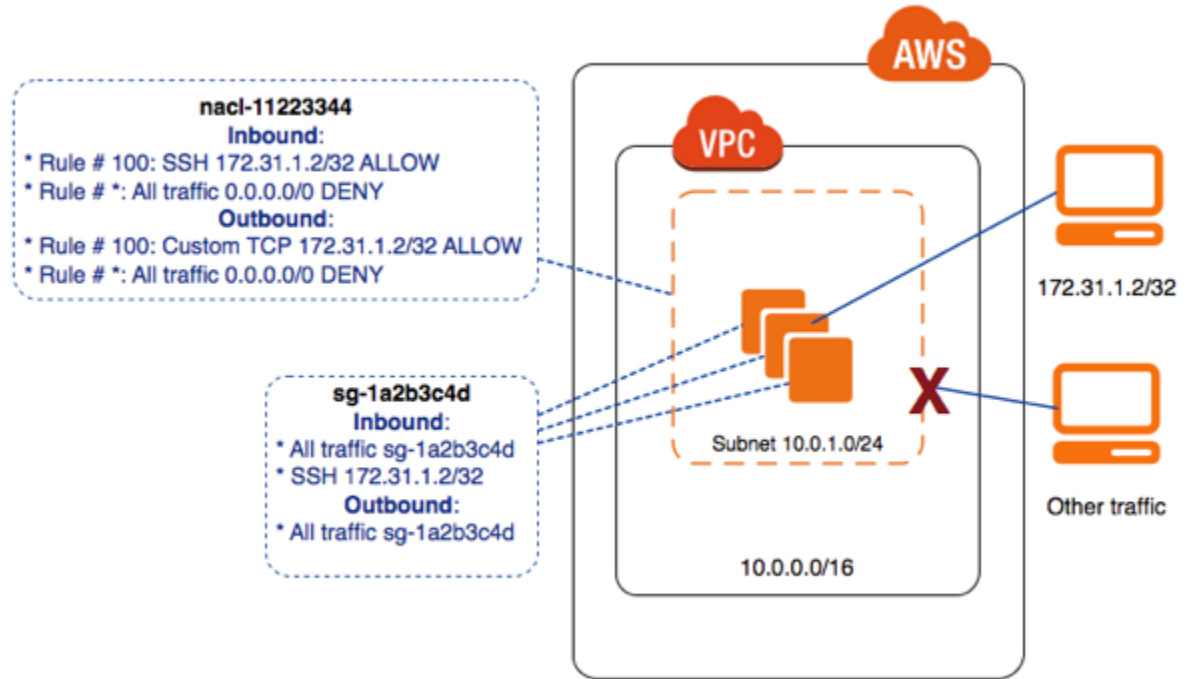
네트워크 ACL과 연결된 서브넷이 없는 경우에만 네트워크 ACL을 삭제할 수 있습니다. 기본 네트워크 ACL은 삭제할 수 없습니다.

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.
SSH	TCP	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	대상 주소	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 #	유형	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 일치하지 않는 다른 모든 인바운드 트래픽을 거부합니다.

아웃바운드 규칙

규칙 #	유형	프로토콜	포트 범위	대상 주소	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

유형	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

유형	프로토콜	포트 범위	대상 주소	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 \(p. 7\)](#) 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. 자세한 정보는 [네트워크 ACL \(p. 130\)](#) 단원을 참조하십시오.

각 시나리오에 대해 다음과 같은 규칙을 권장합니다.

권장 사항

- [시나리오 1을 위한 권장 규칙 \(p. 144\)](#)
- [시나리오 2를 위한 권장 규칙 \(p. 147\)](#)
- [시나리오 3을 위한 권장 규칙 \(p. 153\)](#)
- [시나리오 4를 위한 권장 규칙 \(p. 158\)](#)

고려 사항

- 여기서는 NAT 게이트웨이에 휘발성 포트 범위(예: 32768-65535) 또는 1024-65535를 사용합니다. 구성에 적합한 범위를 선택해야 합니다. 자세한 정보는 [휘발성 포트 \(p. 137\)](#) 단원을 참조하십시오.
- 서브넷의 호스트 간 최대 전송 단위(MTU)가 다른 경우, 경로 MTU 검색이 올바르게 작동하고 패킷 손실을 방지하도록 인바운드와 아웃바운드 모두에 다음 규칙을 추가해야 합니다. 즉, 유형에 사용자 지정 ICMP 규칙, 포트 범위에 대상에 연결할 수 없음, 조각화 필요, DF 플래그 설정(유형 3, 코드 4)을 선택합니다. traceroute를 사용할 경우에는 다음 규칙도 추가합니다. 즉, 유형에 사용자 지정 ICMP 규칙, 포트 범위에 시간 초과, TTL 전송 만료(유형 11, 코드 0)를 선택합니다. 자세한 정보는 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 정보는 [시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC \(p. 23\)](#) 단원을 참조하십시오.

다음 표에는 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다.

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용

110	0.0.0.0/0	TCP	443	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	허용	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	허용	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	32768-65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
120	0.0.0.0/0	TCP	32768-65535	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 취발성 포트 (p. 137) 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).

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
150	::/0	TCP	80	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
160	::/0	TCP	443	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	허용	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	허용	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
190	::/0	TCP	32768-65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
130	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.
140	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
150	::/0	TCP	32768-65535	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 취발성 포트 (p. 137) 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 정보는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\)](#) (p. 30) 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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
110	0.0.0.0/0	TCP	443	허용	어떤 IPv4 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
120	홈 네트워크의 공인 IP 주소 범위	TCP	22	허용	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 공인 IP 주소 범위	TCP	3389	허용	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	1024~65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)

Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
120	10.0.1.0/24	TCP	1433	허용	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
140	0.0.0.0/0	TCP	32768-65535	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 회발성 포트 (p. 137) 단원을 참조하십시오.
150	10.0.1.0/24	TCP	22	허용	프라이빗 서브넷의 인스턴스에 대한 아웃바운드 SSH 액세스 허용(SSH 배스천이 있다면 SSH 배스천으로부터).
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽 거부(수정 불가)

프라이빗 서브넷의 ACL 규칙

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	10.0.0.0/24	TCP	1433	허용	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/

					Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
120	10.0.0.0/24	TCP	22	허용	퍼블릭 서브넷의 SSH 바스천으로부터의 인바운드 SSH 트래픽 허용(SSH 바스천이 있는 경우)
130	10.0.0.0/24	TCP	3389	허용	퍼블릭 서브넷의 Microsoft Terminal Services 게이트웨이로부터의 인바운드 RDP 트래픽 허용.
140	0.0.0.0/0	TCP	1024~65535	허용	프라이빗 서브넷에서 발신되는 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)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.
110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
120	10.0.0.0/24	TCP	32768-65535	허용	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 취발성 포트 (p. 137) 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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
150	::/0	TCP	80	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
160	::/0	TCP	443	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	허용	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해)
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	허용	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해)
190	::/0	TCP	1024~65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
160	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.

170	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용
180	2001:db8:1234::/64	TCP	1433	허용	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
200	::/0	TCP	32768-65535	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 취발성 포트 (p. 137) 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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
150	2001:db8:1234::/64	TCP	1433	허용	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.

170	2001:db8:1234::/64	TCP	22	허용	퍼블릭 서브넷의 SSH 바운드로부터의 인바운드 SSH 트래픽 허용(해당되는 경우).
180	2001:db8:1234::/64	TCP	3389	허용	퍼블릭 서브넷의 Microsoft Terminal Services 게이트웨이로부터의 인바운드 RDP 트래픽 허용(해당되는 경우).
190	::/0	TCP	1024~65535	허용	프라이빗 서브넷에서 발신되는 요청에 대해 발생하는 외부 전용 인터넷 게이트웨이로부터의 인바운드 리턴 트래픽을 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
130	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.
140	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
150	2001:db8:1234::/64	TCP	32768-65535	허용	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 취발성 포트 (p. 137) 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, 연결을 통해 홈 네트워크하고만 통신할 수 있는 인스턴스를 가진 VPN 전용 서브넷입니다. 자세한 정보는 [시나리오 3: 퍼블릭 및 프라이빗 서브넷과 액세스를 포함하는 VPC \(p. 42\)](#) 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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTP 트래픽 허용
110	0.0.0.0/0	TCP	443	허용	어떤 IPv4 주소에서든 웹 서버로의 인바운드 HTTPS 트래픽 허용
120	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	22	허용	홈 네트워크에서 웹 서버로의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해).
130	홈 네트워크의 퍼블릭 IPv4 주소 범위	TCP	3389	허용	홈 네트워크에서 웹 서버로의 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해).
140	0.0.0.0/0	TCP	32768-65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 회발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv4 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	0.0.0.0/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.

110	0.0.0.0/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
120	10.0.1.0/24	TCP	1433	허용	VPN 전용 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
140	0.0.0.0/0	TCP	32768-65535	허용	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 취발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽 거부(수정 불가).

VPN 전용 서브넷의 ACL 설정

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	10.0.0.0/24	TCP	1433	허용	퍼블릭 서브넷의 웹 서버가 VPN 전용 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
120	홈 네트워크의 프라이빗 IPv4 주소 범위	TCP	22	허용	홈 네트워크로부터의 인바운드 SSH 트래픽 허용(가상 프라이빗 게이트웨이를 통해).

130	홈 네트워크의 프라이빗 IPv4 주소 범위	TCP	3389	허용	홈 네트워크로부터의 인바운드 RDP 트래픽 허용(가상 프라이빗 게이트웨이를 통해).
140	홈 네트워크의 프라이빗 IP 주소 범위	TCP	32768-65535	허용	<p>홈 네트워크의 클라이언트로부터의 인바운드 반환 트래픽 허용(가상 프라이빗 게이트웨이를 통해)</p> <p>이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.</p>
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 처리하지 않은 모든 인바운드 트래픽 거부(수정 불가).
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	홈 네트워크의 프라이빗 IP 주소 범위	모두	모두	허용	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	32768-65535	허용	<p>퍼블릭 서브넷의 웹 서버에 대해 아웃바운드 응답 허용.</p> <p>이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.</p>
120	홈 네트워크의 프라이빗 IP 주소 범위	TCP	32768-65535	허용	<p>홈 네트워크의 클라이언트에 대한 아웃바운드 응답 허용(가상 프라이빗 게이트웨이를 통해).</p> <p>이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.</p>

*	0.0.0.0/0	all	all	DENY	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽 거부(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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
150	::/0	TCP	80	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTP 트래픽도 모두 허용
160	::/0	TCP	443	허용	어떤 IPv6 주소에서 이루어지는 인바운드 HTTPS 트래픽도 모두 허용
170	홈 네트워크의 IPv6 주소 범위	TCP	22	허용	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해)
180	홈 네트워크의 IPv6 주소 범위	TCP	3389	허용	홈 네트워크에서 발생하는 IPv6를 통한 인바운드 RDP 트래픽 허용(인터넷 게이트웨이를 통해)
190	::/0	TCP	1024~65535	허용	서브넷에서 발신되는 요청에 응답하는 인터넷 호스트로부터의 인바운드 리턴 트래픽 허용. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
150	::/0	TCP	80	허용	서브넷에서 인터넷으로의 아웃바운드 HTTP 트래픽 허용.

160	::/0	TCP	443	허용	서브넷에서 인터넷으로의 아웃바운드 HTTPS 트래픽 허용.
170	2001:db8:1234::/64	TCP	1433	허용	프라이빗 서브넷의 데이터베이스 서버에 대해 아웃바운드 MS SQL 액세스 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
190	::/0	TCP	32768-65535	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 취발성 포트 (p. 137) 단원을 참조하십시오.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽 거부(수정 불가)

VPN 전용 서브넷의 ACL 규칙

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
150	2001:db8:1234::/64	TCP	1433	허용	퍼블릭 서브넷의 웹 서버가 프라이빗 서브넷의 MS SQL 서버에 읽기 및 쓰기를 할 수 있도록 허용. 이 포트 번호는 예시일 뿐입니다. 다른 예로는 MySQL/Aurora 액세스용 3306, PostgreSQL 액세스용 5432, Amazon Redshift 액세스용 5439, Oracle 액세스용 1521 등이 있습니다.
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments

130	2001:db8:1234::/64	TCP	32768-65535	허용	퍼블릭 서버넷에 대한 아웃바운드 응답 허용(예: 퍼블릭 서버넷에서 프라이빗 서버넷의 DB 서버와 통신하는 웹 서버에 대한 응답). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 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 연결을 통해 홈 네트워크하고만 통신할 수 있는 인스턴스를 가진 단일 서버넷입니다. 자세한 정보는 [시나리오 4: 프라이빗 서버넷만 있고 액세스를 제공하는 VPC \(p. 54\)](#) 단원을 참조하십시오.

다음 표에는 권장되는 규칙이 나와 있습니다. 이 규칙은 명시적으로 요구되는 트래픽을 제외한 모든 트래픽을 차단합니다.

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
100	홈 네트워크의 프라이빗 IP 주소 범위	TCP	22	허용	홈 네트워크에서 서버넷으로의 인바운드 SSH 트래픽 허용.
110	홈 네트워크의 프라이빗 IP 주소 범위	TCP	3389	허용	홈 네트워크에서 서버넷으로의 인바운드 RDP 트래픽 허용.
120	홈 네트워크의 프라이빗 IP 주소 범위	TCP	32768-65535	허용	서버넷에서 발신되는 요청에서 발생하는 인바운드 리턴 트래픽 허용(해당되는 경우). 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 취발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 처리하지 않은 모든 인바운드 트래픽 거부(수정 불가).
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
100	홈 네트워크의 프라이빗 IP 주소 범위	모두	모두	허용	서버넷에서 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	32768-65535	허용	홈 네트워크의 클라이언트에 대한 아웃바운드 응답을 허용합니다. 이 범위는 예시일 뿐입니다. 구성에 맞는 올바른 임시 포트 선택에 대한 자세한 정보는 회발성 포트 (p. 137) 단원을 참조하십시오.
*	0.0.0.0/0	all	all	DENY	이전 규칙에서 처리하지 않은 모든 아웃바운드 트래픽 거부(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

Inbound					
Rule #	Source IP	Protocol	Port	Allow/Deny	Comments
*	::/0	all	all	DENY	이전 규칙에서 아직 처리하지 않은 모든 인바운드 IPv6 트래픽 거부(수정 불가)
Outbound					
Rule #	Dest IP	Protocol	Port	Allow/Deny	Comments
*	::/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 증명

을 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 \(p. 160\)](#)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 정보는 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [Amazon EC2에 대한 IAM 정책](#) 단원을 참조하십시오.

내용

- [AWS CLI 또는 SDK용 예제 정책 \(p. 160\)](#)
- [콘솔용 예제 정책 \(p. 167\)](#)

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.

예제

- [1. VPC 관리 \(p. 160\)](#)
- [2. Amazon VPC에 대한 읽기 전용 정책 \(p. 161\)](#)
- [3. Amazon VPC에 대한 사용자 지정 정책 \(p. 161\)](#)
- [4. 특정 서브넷으로 인스턴스 시작 \(p. 162\)](#)
- [5. 특정 VPC로 인스턴스 시작 \(p. 162\)](#)
- [6. VPC에서 보안 그룹 관리 \(p. 163\)](#)
- [7. VPC 피어링 연결 생성 및 관리 \(p. 164\)](#)
- [8. VPC 엔드포인트 생성 및 관리 \(p. 166\)](#)

ClassicLink로 작업하기 위한 정책 예제는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [CLI 또는 SDK용 예제 정책](#) 단원을 참조하십시오.

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, 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": "*"
  }
]
```

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": "*"
  }
]
```

3. Amazon VPC에 대한 사용자 지정 정책

다음 정책에서는 사용자에게 인스턴스를 시작, 중지, 시작 및 종료하고 Amazon EC2와 Amazon VPC에 사용할 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": "*"
  }
]
```

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"
    ]
  }
]
```

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*"
    ]
  }
]
}
```

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"
    ],
    "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": "*"
  }
]
```

```
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:DescribeSecurityGroups",
    "Resource": "*"
  }
]
```

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=Peering으로 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:AccepterVpc` 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:AccepterVpc": "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 작업할 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/*"
      }
    }
  }
]
}
```

8. VPC 엔드포인트 생성 및 관리

다음 정책에서는 사용자에게 VPC 엔드포인트, VPC 엔드포인트 서비스 및 VPC 엔드포인트 연결 알림을 생성, 수정, 보기 및 삭제하는 권한을 부여합니다. 사용자는 또한 VPC 엔드포인트 연결 요청을 수락 및 거부할 수 있습니다. 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

IAM 콘솔에서 Amazon VPC 정책을 사용하여 특정 리소스를 조회하고 관련 작업을 수행할 권한을 부여할 수 있습니다. 이전 단원의 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 단원에서는 VPC 콘솔의 특정 부분을 사용하도록 허용하는 정책을 보여줍니다.

예제

- 1. VPC 마법사 사용 (p. 167)
- 2. VPC 관리 (p. 171)
- 3. 보안 그룹 관리 (p. 172)
- 4. VPC 피어링 연결 생성 (p. 173)

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 \(p. 23\)](#) 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 단원을 표시합니다. 사용자가 기본 설정을 그대로 두려고 하더라도, 이런 옵션이 표시되어 있지 않으면 VPC를 생성할 수 없습니다.
- `ec2:DescribeVpcEndpointServices`: 마법사의 VPC 엔드포인트 단원을 표시합니다.
- `ec2:AttachInternetGateway`: 인터넷 게이트웨이를 VPC에 연결합니다.
- `ec2:CreateRoute`: 사용자 지정 라우팅 테이블에 경로를 생성합니다. 이 경로는 트래픽이 인터넷 게이트웨이를 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 단원을 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 서브넷과 액세스를 포함하는 VPC

세 번째 VPC 마법사 구성 옵션은 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 생성하고, VPC와 자체 네트워크 사이에 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. 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`: 연결을 생성하려면 다음과 같이 합니다.
- `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 액세스를 제공하는 VPC

네 번째 VPC 구성 옵션은 프라이빗 서브넷이 있는 VPC를 생성하고, VPC와 자체 네트워크 사이에 연결을 생성합니다. 다른 세 옵션과는 달리, 사용자가 VPC에 인터넷 게이트웨이를 연결하거나 생성하는 권한이 필요하지 않고, 라우팅 테이블을 생성하여 서브넷과 연결할 권한이 필요하지 않습니다. 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": "*"
  }
}

```

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 Name의 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"
        }
      }
    }
  ]
}

```

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"
        }
      }
    }
  ]
}

```

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 사용하려면 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 \(p. 164\)](#) 단원을 참조하십시오.

VPC 흐름 로그

VPC 흐름 로그는 VPC의 네트워크 인터페이스에서 전송되고 수신되는 IP 트래픽에 대한 정보를 수집할 수 있는 기능입니다. 플로우 로그 데이터를 Amazon CloudWatch Logs 및 Amazon S3로 게시할 수 있습니다. 플로우 로그를 생성한 다음 선택된 대상의 데이터를 가져와 확인할 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.

플로우 로그를 사용하면, CloudWatch Logs 또는 Amazon S3 중 어디로 보내든, CloudWatch Logs 요금이 적용됩니다. 자세한 정보는 [Amazon CloudWatch 요금](#)을(를) 참조하십시오.

내용

- [흐름 로그 기본 사항](#) (p. 174)
- [흐름 로그 레코드](#) (p. 174)
- [흐름 로그 제한](#) (p. 177)
- [CloudWatch Logs에 플로우 로그 게시](#) (p. 177)
- [Amazon S3에 플로우 로그 게시](#) (p. 180)
- [흐름 로그 작업](#) (p. 184)
- [문제 해결](#) (p. 187)

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.

모니터링된 네트워크 인터페이스를 위한 플로우 로그 데이터는 트래픽 흐름을 설명하는 필드로 구성된 로그 이벤트인 플로우 로그 레코드로서 기록됩니다. 자세한 정보는 [흐름 로그 레코드](#) (p. 174) 단원을 참조하십시오.

플로우 로그를 생성하려면 플로우 로그 생성을 필요로 하는 리소스, 캡처할 트래픽 유형(허용된 트래픽, 거부된 트래픽 또는 모든 트래픽) 및 플로우 로그 데이터를 게시할 대상을 지정합니다. 플로우 로그를 생성한 후에는, 데이터를 수집하여 선택된 대상에 게시하는 데 몇 분의 시간이 소요될 수 있습니다. 흐름 로그는 네트워크 인터페이스에 대한 로그 스트림을 실시간으로 캡처하지 않습니다. 자세한 정보는 [흐름 로그 생성](#) (p. 185) 단원을 참조하십시오.

서브넷이나 VPC에 대한 플로우 로그를 생성한 후 서브넷에서 더 많은 인스턴스를 시작할 경우, 새로운 네트워크 인터페이스에 대한 네트워크 트래픽이 기록되자마자 해당 네트워크 인터페이스에 대한 새로운 (CloudWatch Logs에 대한) 로그 스트림 또는 (Amazon S3에 대한) 로그 파일 객체가 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 플로우 로그를 생성할 수는 없습니다. Amazon EC2 콘솔이나 Amazon EC2 API를 사용해야 합니다. 마찬가지로 CloudWatch Logs 또는 Amazon S3 콘솔이나 API를 사용해서는 네트워크 인터페이스에 대한 플로우 로그를 생성할 수 없습니다.

흐름 로그가 더 이상 필요하지 않을 경우 삭제할 수 있습니다. 플로우 로그를 삭제하면 리소스에 대한 플로우 로그 서비스가 비활성화되고, 새로운 플로우 로그 레코드가 생성되거나 CloudWatch Logs 또는 Amazon S3에 게시됩니다. 네트워크 인터페이스에 대한 기존의 플로우 로그 레코드나 (CloudWatch Logs에 대한) 로그 스트림 또는 (Amazon S3에 대한) 로그 파일 객체는 삭제되지 않습니다. 기존의 로그 스트림을 삭제하려면 CloudWatch Logs 콘솔을 사용합니다. 기존의 로그 파일 객체를 삭제하려면 Amazon S3 콘솔을 사용합니다. 흐름 로그를 삭제한 후 데이터 수집이 중단되기까지 몇 분 정도 시간이 걸릴 수 있습니다. 자세한 정보는 [흐름 로그 삭제](#) (p. 186) 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.

플로우 로그 레코드 구분

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>
```

다음 표에서는 플로우 로그 레코드의 필드를 설명합니다.

필드	설명
version	VPC 흐름 로그 버전.
account-id	흐름 로그의 AWS 계정 ID.
interface-id	트래픽이 기록되는 네트워크 인터페이스 ID.
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 정보는 지정된 인터넷 프로토콜 번호 단원을 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"> ACCEPT: 보안 그룹 또는 네트워크 ACL에서 허용한 트래픽입니다. REJECT: 보안 그룹 또는 네트워크 ACL에서 허용하지 않은 트래픽입니다.
log-status	흐름 로그의 로깅 상태: <ul style="list-style-type: none"> OK: 데이터가 선택된 대상에 정상적으로 로깅됩니다. NODATA: 캡처 기간 중 네트워크 인터페이스에서 전송하거나 수신된 네트워크 트래픽이 없었습니다. SKIPDATA: 캡처 기간 중 일부 흐름 로그 레코드를 건너뛰었습니다. 내부 용량 제한 또는 내부 오류가 원인일 수 있습니다.

Note

필드가 특정 레코드에 해당되지 않을 경우 레코드에 '-' 기호가 표시됩니다.

플로우 로그 레코드의 예

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. 이는 플로우 로그에서 다음과 같은 2가지 플로우 로그 레코드로 표시됩니다.

- 네트워크 ACL과 보안 그룹이 모두 허용했으며 따라서 인스턴스에 접속하도록 허용된 요청 ping에 대한 ACCEPT 레코드
- 네트워크 ACL이 거부한 응답 ping에 대한 REJECT 레코드

```
2 123456789010 eni-1235b8ca 203.0.113.12 172.31.16.139 0 0 1 4 336 1432917027 1432917142
ACCEPT OK
```

```
2 123456789010 eni-1235b8ca 172.31.16.139 203.0.113.12 0 0 1 4 336 1432917094 1432917142
REJECT OK
```

네트워크 ACL이 아웃바운드 ICMP 트래픽을 허용한 경우, 흐름 로그에 두 가지 ACCEPT 레코드(하나는 요청 ping에 대한 레코드, 다른 하나는 응답 ping에 대한 레코드)가 표시됩니다. 보안 그룹이 인바운드 ICMP 트래픽을 거부한 경우, 흐름 로그에는 하나의 REJECT 레코드만 표시됩니다. 해당 트래픽이 인스턴스에 접속하도록 허용되지 않았기 때문입니다.

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
```

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 플로우 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. 플로우 로그 API 작업의 사용을 제어하는 IAM 정책을 만들려면 명령문에서 해당 리소스에 * 와일드카드를 사용하여 모든 리소스를 작업에 사용할 수 있는 권한을 사용자에게 부여해야 합니다. 자세한 정보는 [Amazon VPC 리소스에 대한 액세스 제어 \(p. 159\)](#) 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.
- 트래픽이 ENI로 전송될 경우 대상이 ENI IP 주소가 아니면 흐름 로그에 대상 IP 주소 필드의 기본 프라이빗 IPv4 주소가 표시됩니다.
- ENI로부터 트래픽이 전송될 경우 소스가 ENI IP 주소가 아니면 흐름 로그에 소스 IP 주소 필드의 기본 프라이빗 IPv4 주소가 표시됩니다.

흐름 로그는 모든 IP 트래픽을 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.
- Amazon Time Sync Service를 위해 169.254.169.123과 주고받는 트래픽.
- DHCP 트래픽.
- 기본 VPC 라우터의 예약된 IP 주소로 보내는 트래픽. 자세한 정보는 [VPC 및 서브넷 크기 \(p. 80\)](#) 단원을 참조하십시오.
- 엔드포인트 네트워크 인터페이스와 Network Load Balancer 네트워크 인터페이스 간의 트래픽. 자세한 정보는 [VPC 엔드포인트 서비스\(AWS PrivateLink\) \(p. 272\)](#) 단원을 참조하십시오.

CloudWatch Logs에 플로우 로그 게시

플로우 로그는 플로우 로그 데이터를 Amazon CloudWatch에 직접 게시할 수 있습니다.

CloudWatch Logs에 게시하는 경우, 플로우 로그 데이터는 로그 그룹에 게시되고, 각 네트워크 인터페이스는 로그 그룹에 고유의 로그 스트림을 가집니다. 로그 스트림에는 플로우 로그 레코드가 포함됩니다. 여러 개의 플로우 로그를 생성하여, 그 데이터를 같은 로그 그룹에 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 지정한 경우, 병합된 로그 스트림은 모든 트래픽을 캡처합니다. 자세한 정보는 [흐름 로그 레코드 \(p. 174\)](#) 단원을 참조하십시오.

내용

- [CloudWatch Logs에 플로우 로그를 게시하기 위한 IAM 역할 \(p. 178\)](#)
- [CloudWatch Logs에 게시하는 플로우 로그 생성 \(p. 179\)](#)
- [CloudWatch Logs에서 플로우 로그 레코드 처리 \(p. 180\)](#)

CloudWatch Logs에 플로우 로그를 게시하기 위한 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": "*"
    }
  ]
}
```

또한 귀하의 역할에 플로우 로그 서비스의 역할 수임을 허용하는 신뢰 관계가 포함되어 있는지 확인해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

또한 사용자에게 해당 플로우 로그와 연결된 IAM 역할에 대한 iam:PassRole 작업의 사용 권한이 있어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole"],
      "Resource": "arn:aws:iam::account-id:role/flow-log-role-name"
    }
  ]
}
```

이미 존재하는 역할을 업데이트하거나 플로우 로그를 사용할 수 있도록 새로운 역할을 만들기 위해서 다음 절차를 따라하실 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], [Create role]을 선택합니다.
3. 이 역할을 사용할 서비스로서 EC2를 선택합니다. 사용 사례에서 EC2를 선택합니다. 다음: 권한을 선택합니다.
4. 권한 정책 연결 페이지에서 다음: 검토를 선택합니다.
5. 역할 이름을 입력하고(예: Flow-Logs-Role) 선택 사항으로써 설명을 제공합니다. [Create role]을 선택합니다.
6. 역할 이름을 선택합니다. 권한에서 Add inline policy(인라인 정책 추가), JSON을 선택합니다.
7. CloudWatch Logs에 플로우 로그를 게시하기 위한 IAM 역할 (p. 178) 에서 첫 번째 정책을 복사한 후 창에 붙여 넣습니다. [Review policy]를 선택합니다.
8. 정책 이름을 입력하고 Create policy(정책 생성)를 선택합니다.
9. 역할 이름을 선택합니다. [Trust Relationships] 으로 들어가려면 [Edit Trust Relationship]을 선택합니다. 기존 정책 문서에서 `ec2.amazonaws.com` 부터 `vpc-flow-logs.amazonaws.com`까지 서비스를 바꾸어주십시오. [Update Trust Policy(신뢰 정책 업데이트)]를 선택합니다.
10. [Summary] 페이지에서 사용자 역할에 대한 ARN을 확인합니다. 사용자의 흐름 로그를 만들 때 이 ARN이 필요합니다.

CloudWatch Logs에 게시하는 플로우 로그 생성

VPC, 서브넷 또는 네트워크 인터페이스에 대한 플로우 로그를 생성할 수 있습니다.

네트워크 인터페이스에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택합니다.
3. 하나 이상의 네트워크 인터페이스를 선택하고 작업, 플로우 로그 생성을 선택합니다.
4. 필터에서 로깅할 IP 트래픽 데이터의 유형을 지정합니다. 모두를 선택하여 수락된 트래픽 및 거부된 트래픽을 로깅하거나, 거부됨을 선택하여 거부된 트래픽만을 기록하거나, 수락 완료를 선택해 수락된 트래픽만을 기록합니다.
5. 대상 주소에서 CloudWatch Logs로 전송을 선택합니다.
6. 대상 로그 그룹에 플로우 로그를 게시할 CloudWatch Logs의 로그 그룹 이름을 입력합니다. 존재하지 않는 로그 그룹의 이름을 지정할 경우 해당 로그 그룹이 만들어집니다.
7. IAM 역할에서 CloudWatch Logs에 로그를 게시할 권한이 있는 역할의 이름을 지정합니다.
8. Create를 선택합니다.

VPC 또는 서브넷에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs 또는 서브넷을 선택합니다.
3. 하나 이상의 VPC 또는 서브넷을 선택한 다음 작업, 플로우 로그 생성을 선택합니다.
4. 필터에서 로깅할 IP 트래픽 데이터의 유형을 지정합니다. 모두를 선택하여 수락된 트래픽 및 거부된 트래픽을 로깅하거나, 거부됨을 선택하여 거부된 트래픽만을 기록하거나, 수락 완료를 선택해 수락된 트래픽만을 기록합니다.
5. 대상 주소에서 CloudWatch Logs로 전송을 선택합니다.
6. 대상 로그 그룹에 플로우 로그를 게시할 CloudWatch Logs의 로그 그룹 이름을 입력합니다. 존재하지 않는 로그 그룹의 이름을 지정할 경우 해당 로그 그룹이 만들어집니다.

7. IAM 역할에서 CloudWatch Logs에 로그를 게시할 권한이 있는 IAM 역할의 이름을 지정합니다.
8. Create를 선택합니다.

CloudWatch Logs에서 플로우 로그 레코드 처리

흐름 로그 레코드는 CloudWatch Logs에서 수집한 다른 로그 이벤트처럼 사용할 수 있습니다. 로그 데이터 및 지표 필터 모니터링에 대한 자세한 정보는 Amazon CloudWatch 사용 설명서의 [로그 데이터 검색 및 필터링](#)을 참조하십시오.

예 :흐름 로그에 대한 CloudWatch 지표 필터 및 경보 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 알람을 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. 필터 패턴에 다음을 입력합니다.

```
[version, account, eni, source, destination, srcport, destport="22", protocol="6", packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]
```

4. 테스트할 로그 데이터 선택에서 네트워크 인터페이스에 대한 로그 스트림을 선택합니다. (선택 사항) 필터 패턴과 일치하는 로그 데이터 행을 보려면 패턴 테스트를 선택합니다. 준비가 됐으면 [Assign Metric]을 선택합니다.
5. 지표 네임스페이스와 이름을 제공하고 지표 값이 1로 설정되어 있는지 확인합니다. 완료했으면 [Create Filter]를 선택합니다.
6. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
7. [Custom Metrics] 단원에서, 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 다음을 선택합니다.
9. 경보 이름 및 설명을 입력합니다. is 필드에서 >=를 선택한 후 10을 입력합니다. 기간 필드에서 연속 기간에 대해 기본값 1을 유지합니다.
10. 기간에서 1시간을 선택합니다. 통계에서 합계를 선택합니다. Sum 통계는 지정된 기간 동안 데이터 포인트의 총 수를 캡처합니다.
11. 작업 단원에서, 기존 목록에 알람을 보내도록 선택할 수 있습니다. 또는 새 목록을 만들고 경보가 트리거될 때 알람을 받을 이메일 주소를 입력할 수 있습니다. 완료했으면 [Create Alarm]을 선택합니다.

Amazon S3에 플로우 로그 게시

플로우 로그는 플로우 로그 데이터를 Amazon S3에 게시할 수 있습니다.

Amazon S3에 게시하는 경우, 플로우 로그 데이터가 지정해 놓은 기존 Amazon S3 버킷에 게시됩니다. 모니터링 모든 네트워크 인터페이스에 대한 플로우 로그 레코드는 버킷에 저장된 일련의 로그 파일 객체에 게시됩니다. 플로우 로그가 VPC에 대한 데이터를 캡처하면, 플로우 로그가 모든 네트워크 인터페이스에 대한

플로우 로그 레코드를 선택된 VPC에 게시합니다. 자세한 정보는 [흐름 로그 레코드 \(p. 174\)](#) 단원을 참조하십시오.

플로우 로그를 사용할 수 있도록 Amazon S3 버킷을 생성하려면 Amazon Simple Storage Service 시작 안내서의 [Create a Bucket\(버킷 생성\)](#)을 참조하십시오.

내용

- [플로우 로그 파일 \(p. 181\)](#)
- [Amazon S3에 플로우 로그를 게시하기 위한 IAM 역할 \(p. 181\)](#)
- [Amazon S3 버킷의 플로우 로그에 대한 권한 \(p. 182\)](#)
- [SSE-KMS 버킷에 사용할 경우 필요한 CMK 키 정책 \(p. 183\)](#)
- [Amazon S3 로그 파일 권한 \(p. 183\)](#)
- [Amazon S3에 게시하는 플로우 로그 생성 \(p. 183\)](#)
- [Amazon S3에서 플로우 로그 레코드 처리 \(p. 184\)](#)

플로우 로그 파일

플로우 로그는 플로우 로그 레코드를 수집하여 로그 파일로 통합한 다음 해당 로그 파일을 5분 간격으로 Amazon S3 버킷에 게시합니다. 각 로그 파일에는 이전 5분 동안 기록된 IP 트래픽에 대한 플로우 로그 레코드가 포함됩니다.

로그 파일의 최대 크기는 75MB입니다. 로그 파일이 5분 내에 파일 크기 한도에 도달하는 경우, 플로우 로그는 플로우 로그 레코드의 로그 파일로의 추가를 중단하고 Amazon S3 버킷에 게시한 다음 새로운 로그 파일을 생성합니다.

로그 파일은 플로우 로그의 ID, 리전 및 생성된 날짜에 따라 결정된 폴더 구조를 사용하여 지정된 Amazon S3 버킷에 저장됩니다. 버킷 폴더 구조는 다음 형식을 사용합니다.

```
bucket_ARN/optional_folder/AWSLogs/aws_account_id/  
vpcflowlogs/region/year/month/day/log_file_name.log.gz
```

마찬가지로 로그 파일의 파일 이름은 플로우 로그의 ID, 리전 및 생성된 날짜에 따라 결정됩니다. 파일 이름은 다음의 형식을 사용합니다.

```
aws_account_id_vpcflowlogs_region_flow_log_id_timestamp_hash.log.gz
```

Note

타임스탬프는 YYYYMMDDTHHmmZ 형식을 사용합니다.

예를 들어, 아래에서는 16:20 UTC의 June 20, 2018에 us-east-1 리전의 리소스에 대하여 AWS 계정 123456789012에서 생성한 플로우 로그에 대한 로그 파일의 폴더 구조 및 파일 이름을 보여 주며, 이에는 16:15:01~16:20:00에 대한 플로우 로그 레코드가 포함됩니다.

```
arn:aws:s3:::my-flow-log-bucket/AWSLogs/123456789012/  
vpcflowlogs/us-east-1/2018/06/20/123456789012_vpcflowlogs_us-  
east-1_fl-1234abcd_20180620T1620Z_fe123456.log.gz
```

Amazon S3에 플로우 로그를 게시하기 위한 IAM 역할

IAM 사용자와 같은 IAM 보안 주체에는 Amazon S3 버킷에 플로우 로그를 게시할 충분한 권한이 있어야 합니다. IAM 정책에 다음 권한이 포함되어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs>DeleteLogDelivery"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon S3 버킷의 플로우 로그에 대한 권한

기본적으로 Amazon S3 버킷과 그에 포함된 객체는 비공개입니다. 버킷 소유자만이 해당 버킷과 그 안에 저장된 객체에 액세스할 수 있습니다. 그러나 버킷 소유자는 액세스 정책을 작성하여 다른 리소스 및 사용자에게 액세스를 허용할 수 있습니다.

플로우 로그를 생성하는 사용자가 버킷을 소유한 경우, 다음 정책을 해당 버킷에 자동으로 연결하여 로그를 버킷에 게시할 플로우 로그 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*",
      "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
    },
    {
      "Sid": "AWSLogDeliveryAclCheck",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::bucket_name"
    }
  ]
}
```

플로우 로그를 생성하는 사용자가 버킷을 소유하지 않거나 해당 버킷에 대한 GetBucketPolicy와 PutBucketPolicy가 없는 경우, 플로우 로그 생성이 실패합니다. 이 경우 버킷 소유자가 위의 정책을 수동으로 버킷에 추가하고 플로우 로그 생성자의 AWS 계정 ID를 지정해야 합니다. 자세한 정보는 Amazon Simple Storage Service 콘솔 사용 설명서의 [S3 버킷 정책을 추가하려면 어떻게 해야 하나요?](#)를 참조하십시오. 버킷이 여러 계정으로부터 플로우 로그를 수신하는 경우, Resource 요소 입력 내용을 각 계정의 AWSLogDeliveryWrite 정책 설명에 추가합니다. 예를 들어 다음 버킷 정책이 AWS 계정 123123123123과 456456456456이 플로우 로그를 log-bucket이란 이름의 버킷의 flow-logs란 이름의 폴더에 게시하도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {"Service": "delivery.logs.amazonaws.com"},
```

```
    "Action": "s3:PutObject",
    "Resource": [
      "arn:aws:s3::log-bucket/flow-logs/AWSLogs/123123123123/*",
      "arn:aws:s3::log-bucket/flow-logs/AWSLogs/456456456456/*"
    ],
    "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
  },
  {
    "Sid": "AWSLogDeliveryAclCheck",
    "Effect": "Allow",
    "Principal": {"Service": "delivery.logs.amazonaws.com"},
    "Action": "s3:GetBucketAcl",
    "Resource": "arn:aws:s3::log-bucket"
  }
]
```

Note

AWSLogDeliveryAclCheck 및 AWSLogDeliveryWrite 권한을 각 AWS 계정 ARN보다는 로그 전송 서비스 보안 주체에게 부여하는 것이 좋습니다.

SSE-KMS 버킷에 사용할 경우 필요한 CMK 키 정책

고객 관리형 고객 마스터 키(CMK)와 함께 AWS KMS 관리형 키(SSE-KMS)를 사용하여 Amazon S3 버킷에 대하여 서버 측 암호화를 활성화한 경우, 플로우 로그가 로그 파일을 버킷에 쓸 수 있도록 CMK의 키 정책에 다음을 추가해야 합니다.

```
{
  "Sid": "Allow VPC Flow Logs to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "delivery.logs.amazonaws.com"
    ]
  },
  "Action": "kms:GenerateDataKey*",
  "Resource": "*"
}
```

Amazon S3 로그 파일 권한

필요한 버킷 정책 외에도 Amazon S3는 액세스 제어 목록(ACL)을 사용해 플로우 로그에 의해 생성된 로그 파일에 대한 액세스를 관리합니다. 기본적으로 버킷 소유자는 각 로그 파일에 대한 FULL_CONTROL 권한을 보유합니다. 로그 전송 소유자가 버킷 소유자와 다른 경우에는 권한이 없습니다. 로그 전송 계정에는 READ 및 WRITE 권한이 부여됩니다. 자세한 정보는 Amazon Simple Storage Service 개발자 가이드의 [ACL\(액세스 제어 목록\) 개요](#) 단원을 참조하십시오.

Amazon S3에 게시하는 플로우 로그 생성

Amazon S3 버킷을 생성한 후 VPC, 서브넷 또는 네트워크 인터페이스에 대한 플로우 로그를 생성할 수 있습니다.

네트워크 인터페이스에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택합니다.
3. 하나 이상의 네트워크 인터페이스를 선택하고 작업, 플로우 로그 생성을 선택합니다.

4. 필터에서 로깅할 IP 트래픽 데이터의 유형을 지정합니다. 모두를 선택하여 수락된 트래픽 및 거부된 트래픽을 로깅하거나, 거부됨을 선택하여 거부된 트래픽만을 기록하거나, 수락 완료를 선택해 수락된 트래픽만을 기록합니다.
5. 대상 주소에서 Amazon S3 버킷으로 전송을 선택합니다.
6. S3 버킷 ARN에서 기존 Amazon S3 버킷의 Amazon 리소스 이름(ARN)을 지정합니다. 버킷 ARN에 하위 폴더를 포함할 수 있습니다. 버킷에 AWSLogs를 하위 폴더 이름으로 사용할 수 없습니다. 이것은 예약된 용어입니다.

예를 들어 my-bucket이란 이름의 버킷에 my-logs이란 이름의 하위 폴더를 지정하려면 다음 ARN을 사용하십시오.

```
arn:aws:s3:::my-bucket/my-logs/
```

버킷을 소유한 경우, 자동으로 리소스 정책을 생성하여 버킷에 연결합니다. 자세한 정보는 [Amazon S3 버킷의 폴로우 로그에 대한 권한 \(p. 182\)](#) 단원을 참조하십시오.

7. Create를 선택합니다.

VPC 또는 서브넷에 대한 흐름 로그를 만들려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs 또는 서브넷을 선택합니다.
3. 하나 이상의 VPC 또는 서브넷을 선택한 다음 작업, 폴로우 로그 생성을 선택합니다.
4. 필터에서 로깅할 IP 트래픽 데이터의 유형을 지정합니다. 모두를 선택하여 수락된 트래픽 및 거부된 트래픽을 로깅하거나, 거부됨을 선택하여 거부된 트래픽만을 기록하거나, 수락 완료를 선택해 수락된 트래픽만을 기록합니다.
5. 대상 주소에서 Amazon S3 버킷으로 전송을 선택합니다.
6. S3 버킷 ARN에서 기존 Amazon S3 버킷의 Amazon 리소스 이름(ARN)을 지정합니다. 버킷 ARN에 하위 폴더를 포함할 수 있습니다. 버킷에 AWSLogs를 하위 폴더 이름으로 사용할 수 없습니다. 이것은 예약된 용어입니다.

예를 들어 my-bucket이란 이름의 버킷에 my-logs이란 이름의 하위 폴더를 지정하려면 다음 ARN을 사용하십시오.

```
arn:aws:s3:::my-bucket/my-logs/
```

버킷을 소유한 경우, 자동으로 리소스 정책을 생성하여 버킷에 연결합니다. 자세한 정보는 [Amazon S3 버킷의 폴로우 로그에 대한 권한 \(p. 182\)](#) 단원을 참조하십시오.

7. Create를 선택합니다.

Amazon S3에서 폴로우 로그 레코드 처리

로그 파일은 압축된 상태입니다. Amazon S3 콘솔을 사용해 로그 파일을 열면 압축이 해제되고 폴로우 로그 레코드가 표시됩니다. 파일을 다운로드하는 경우, 압축을 해제해야 폴로우 로그 레코드를 볼 수 있습니다.

또한 을 사용해 로그 파일의 폴로우 로그 레코드를 쿼리할 수도 있습니다. 은 표준 SQL을 사용해 Amazon S3에 저장된 데이터를 간편하게 분석할 수 있는 대화형 쿼리 서비스입니다. 자세한 정보는 [사용 설명서의 Amazon VPC 폴로우 로그 쿼리 방법](#)을 참조하십시오.

흐름 로그 작업

Amazon EC2, Amazon VPC, CloudWatch 및 Amazon S3 콘솔을 사용하여 폴로우 로그에 대한 작업을 수행할 수 있습니다.

내용

- [흐름 로그 사용 제어](#) (p. 185)
- [흐름 로그 생성](#) (p. 185)
- [흐름 로그 확인](#) (p. 185)
- [플로우 로그 레코드 보기](#) (p. 186)
- [흐름 로그 삭제](#) (p. 186)
- [API 및 CLI 개요](#) (p. 187)

흐름 로그 사용 제어

기본적으로 IAM 사용자에게는 플로우 로그 사용 권한이 없습니다. IAM 사용자 정책을 만들어 사용자에게 플로우 로그를 생성, 설명, 삭제할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 Amazon EC2 API Reference의 [IAM 사용자에게 Amazon EC2 리소스에 대한 필요 권한 부여](#) 단원을 참조하십시오.

다음은 사용자에게 플로우 로그를 생성, 설명 및 삭제할 수 있는 전체 권한을 부여하는 정책의 예입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteFlowLogs",
        "ec2:CreateFlowLogs",
        "ec2:DescribeFlowLogs"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatch Logs 또는 Amazon S3 중 어느 쪽에 게시하는지에 따라 일부 추가적인 IAM 역할 및 권한 구성 이 필요할 수 있습니다. 자세한 정보는 [CloudWatch Logs에 플로우 로그 게시](#) (p. 177) 및 [Amazon S3에 플로우 로그 게시](#) (p. 180) 단원을 참조하십시오.

흐름 로그 생성

VPC, 서브넷 또는 네트워크 인터페이스에 대한 플로우 로그를 생성할 수 있습니다. 플로우 로그가 데이터를 CloudWatch Logs 및 Amazon S3에 게시할 수 있습니다.

자세한 정보는 [CloudWatch Logs에 게시하는 플로우 로그 생성](#) (p. 179) 및 [Amazon S3에 게시하는 플로우 로그 생성](#) (p. 183) 단원을 참조하십시오.

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 정보 등이 표시됩니다.

네트워크 인터페이스에 대한 플로우 로그 정보를 보려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Network Interfaces]를 선택합니다.
3. 네트워크 인터페이스를 선택한 후 플로우 로그를 선택합니다. 흐름 로그에 대한 정보가 탭에 표시됩니다. Destination type(대상 유형) 열은 플로우 로그를 게시할 대상을 표시합니다.

VPC 또는 서브넷에 대한 플로우 로그 정보를 보려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs 또는 서브넷을 선택합니다.
3. VPC 또는 서브넷을 선택한 후 플로우 로그를 선택합니다. 흐름 로그에 대한 정보가 탭에 표시됩니다. Destination type(대상 유형) 열은 플로우 로그를 게시할 대상을 표시합니다.

플로우 로그 레코드 보기

선택된 대상 유형에 따라 CloudWatch Logs 콘솔 또는 Amazon S3 콘솔을 사용하여 플로우 로그 레코드를 볼 수 있습니다. 흐름 로그를 생성한 후 콘솔에서 흐름 로그를 보려면 몇 분 정도 지나야 할 수 있습니다.

CloudWatch Logs에 게시된 플로우 로그 레코드를 보려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그를 선택하고 플로우 로그가 포함된 로그 그룹을 선택합니다. 각 네트워크 인터페이스에 대한 로그 스트림 목록이 표시됩니다.
3. 흐름 로그 레코드를 보려는 네트워크 인터페이스의 ID가 있는 로그 스트림을 선택합니다. 자세한 정보는 [흐름 로그 레코드](#) (p. 174) 단원을 참조하십시오.

Amazon S3에 게시된 플로우 로그 레코드를 보려면

1. <https://console.aws.amazon.com/s3/>에서 Amazon S3 콘솔을 엽니다.
2. 버킷 이름에서 플로우 로그를 게시할 버킷을 선택합니다.
3. 이름에서 로그 파일 옆의 확인란을 선택합니다. 객체 개요 패널에서 다운로드를 선택합니다.

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. 플로우 로그를 삭제해도, CloudWatch Logs에서 로그 스트림이 삭제되지 않으며 Amazon S3에서 로그 파일이 삭제되지 않습니다. 각 서비스의 콘솔을 사용해 기존 플로우 로그 데이터를 삭제해야 합니다. 또한 Amazon S3에 게시되는 플로우 로그를 삭제해도, 버킷 정책과 로그 파일 액세스 제어 목록(ACL)은 삭제되지 않습니다.

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. 플로우 로그를 선택한 후 삭제할 플로우 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 VPCs 또는 서브넷을 선택한 후 리소스를 선택합니다.
3. 플로우 로그를 선택한 후 삭제할 플로우 로그의 삭제 버튼(x 표시)을 선택합니다.
4. 확인 대화 상자에서 [Yes, Delete]를 선택합니다.

API 및 CLI 개요

이 페이지에서 설명한 작업은 명령줄이나 API를 사용하여 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 정보는 [Amazon VPC에 액세스 \(p. 7\)](#) 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)

문제 해결

다음은 플로우 로그로 작업할 때 발생할 수 있는 문제입니다.

문제

- 불완전한 흐름 로그 레코드 (p. 187)
- 흐름 로그가 활성화되었지만 흐름 로그 레코드 또는 로그 그룹이 없음 (p. 188)
- 오류: `LogDestinationNotFoundException` (p. 188)
- Amazon S3 버킷 정책 제한 초과 (p. 188)

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 플로우 로그 탭을 확인합니다. 자세한 정보는 [흐름 로그 확인 \(p. 185\)](#) 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 플로우 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 정보는 Amazon CloudWatch 사용 설명서의 [CloudWatch 제한](#)을 참조하십시오.

- **Access error:** 흐름 로그에 대한 IAM 역할에 CloudWatch 로그 그룹에 흐름 로그를 게시할 권한이 없습니다. 자세한 정보는 [CloudWatch Logs에 플로우 로그를 게시하기 위한 IAM 역할 \(p. 178\)](#) 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 Amazon S3 버킷에서 로그 파일을 볼 수 없습니다. 원인은 다음 중 하나일 수 있습니다.

- 흐름 로그가 아직 생성되는 중입니다. 경우에 따라 플로우 로그를 생성한 후 로그 그룹이 생성되고 데이터가 표시되기까지 10분 이상 걸릴 수 있습니다.
- 네트워크 인터페이스에 대해 기록된 트래픽이 아직 없습니다. CloudWatch Logs의 로그 그룹은 트래픽이 기록될 경우에만 생성됩니다.

오류: LogDestinationNotFoundException

데이터를 Amazon S3 버킷에 게시하는 플로우 로그를 생성할 때 이 오류 코드가 뜰 수 있습니다. 이 오류 코드는 지정한 S3 버킷을 찾을 수 없음을 표시합니다.

이 오류를 해결하려면 기존 S3 버킷에 ARN을 지정했는지, 그리고 그 ARN의 형식이 올바른지 확인해야 합니다.

Amazon S3 버킷 정책 제한 초과

Amazon S3 버킷 정책은 크기가 20KB로 제한됩니다.

Amazon S3 버킷에 게시하는 플로우 로그를 생성할 때마다, 폴더 경로를 포함하는 지정된 버킷 ARN을 버킷 정책의 **Resource** 요소에 자동으로 추가합니다.

동일한 버킷에 게시하는 여러 개의 플로우 로그를 생성하면 버킷 정책 제한을 초과할 수 있습니다. 버킷 정책 제한을 초과할 경우 **LogDestinationPermissionIssueException** 오류가 발생합니다.

버킷 정책 제한을 초과하여 이 오류 메시지를 받은 경우 다음 중 하나를 수행하십시오.

- 더 이상 필요 없는 플로우 로그 항목을 제거하여 버킷의 정책을 정리합니다.
- 개별 플로우 로그 항목을 다음으로 대체하여 전체 버킷에 권한을 부여합니다.

```
arn:aws:s3:::bucket_name/*
```

전체 버킷에 권한을 부여할 경우, 새 플로우 로그 구독이 버킷 정책에 새 권한을 추가합니다.

VPC 네트워킹 구성 요소

다음 구성 요소에 따라 VPC의 네트워킹을 구성할 수 있습니다.

네트워킹 구성 요소

- [네트워크 인터페이스](#) (p. 189)
- [라우팅 테이블](#) (p. 190)
- [인터넷 게이트웨이](#) (p. 202)
- [외부 전용 인터넷 게이트웨이](#) (p. 207)
- [DHCP 옵션 세트](#) (p. 234)
- [DNS](#) (p. 239)
- [탄력적 IP 주소](#) (p. 243)
- [VPC 엔드포인트](#) (p. 246)
- [NAT](#) (p. 210)
- [VPC 피어링](#) (p. 243)
- [ClassicLink](#) (p. 282)

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 범위 내에 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.

다음은 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
- 저예산고가용성 솔루션 생성

네트워크 인스턴스에 대한 자세한 내용 및 Amazon EC2 콘솔을 사용한 네트워크 인스턴스 작업에 대한 지침은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [탄력적 네트워크 인터페이스](#)를 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.

내용

- [라우팅 테이블 기본 사항 \(p. 190\)](#)
- [라우팅 우선순위 \(p. 193\)](#)
- [라우팅 옵션 \(p. 194\)](#)
- [라우팅 테이블 작업 \(p. 197\)](#)
- [API 및 명령 개요 \(p. 200\)](#)

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. VPC에 하나 이상의 IPv4 CIDR블록이 연결되어 있는 경우, 라우팅 테이블에 각 IPv4 CIDR 블록의 로컬 경로가 포함됩니다. IPv6 CIDR 블록을 VPC와 연결한 경우, 라우팅 테이블에 IPv6 CIDR 블록의 로컬 경로가 포함됩니다. 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 정보는 [Amazon VPC 제한 \(p. 285\)](#) 단원을 참조하십시오.

기본 라우팅 테이블

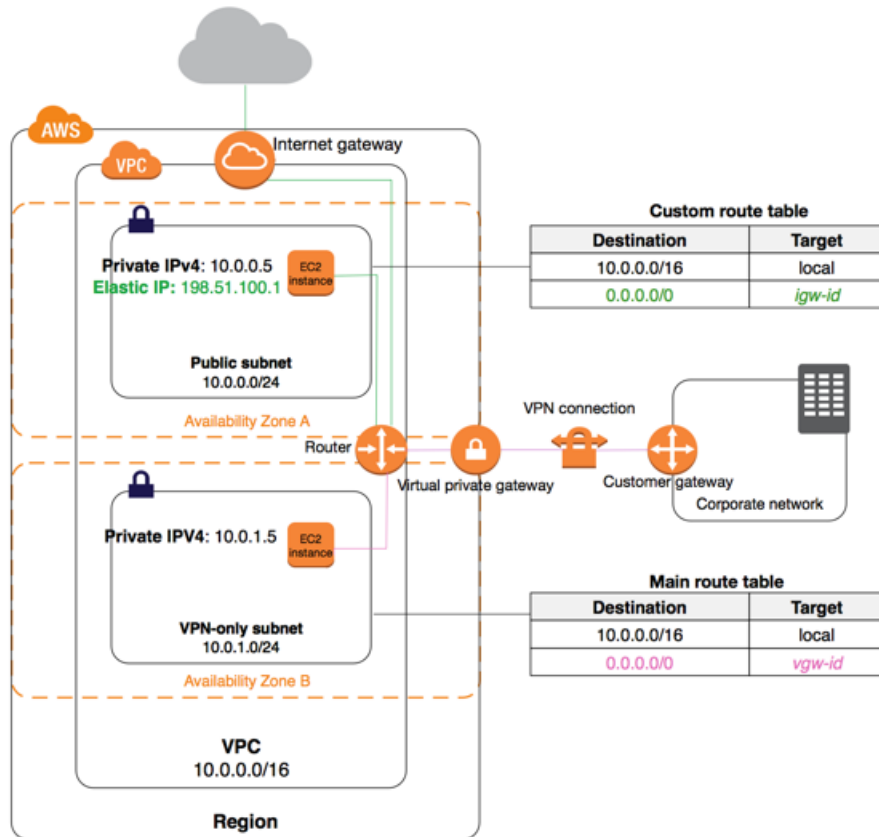
VPC를 만들면 기본 라우팅 테이블이 자동으로 생성됩니다. Amazon VPC 콘솔의 [Route Tables] 페이지의 [Main] 열에서 [Yes]를 찾아 VPC에 대한 기본 라우팅 테이블을 볼 수 있습니다. 기본 라우팅 테이블은 다른 라우팅 테이블과 명시적으로 연결되지 않은 모든 서브넷에 대한 라우팅을 제어합니다. 기본 라우팅 테이블에서 라우팅을 추가 및 제거하고 수정할 수 있습니다.

서브넷이 기본 라우팅 테이블에 명시적으로 연결되어 있지 않을 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 정보는 [기본 라우팅 테이블 바꾸기 \(p. 200\)](#) 단원을 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(대상 주소는 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 가

상 프라이빗 게이트웨이에 대한 경로가 있는 사용자 지정 라우팅 테이블)을 설정하면 새 서브넷은 인터넷 게이트웨이에 대한 라우팅을 자동으로 갖게 됩니다.

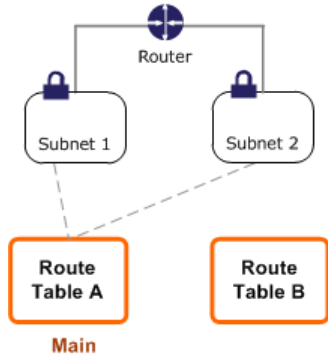
라우팅 테이블 연결

VPC 콘솔은 각 라우팅 테이블에 명시적으로 연결된 서브넷의 수를 보여 주며, 기본 라우팅 테이블과 암시적으로 연결된 서브넷에 대한 정보를 제공합니다. 자세한 정보는 [테이블과 명시적으로 연결되어 있는 서브넷 확인](#) (p. 197) 단원을 참조하십시오.

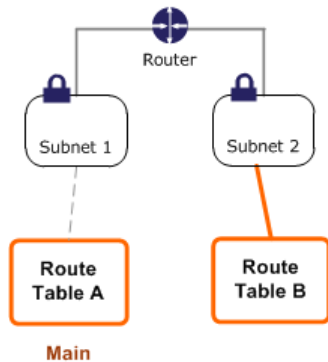
서브넷을 기본 라우팅 테이블과 암시적 또는 명시적으로 연결할 수 있습니다. 기본 라우팅 테이블을 바꿀 경우 서브넷이 일시적으로 기본 라우팅 테이블에 명시적으로 연결될 수도 있겠지만, 일반적으로는 기본 라우팅 테이블에 명시적으로 연결되지 않습니다.

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.

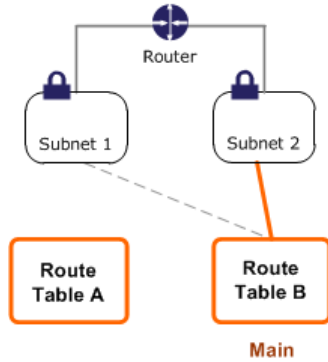
다음 다이어그램에서는 기본 라우팅 테이블(라우팅 테이블 A)과 암시적으로 연결되는 두 개의 서브넷이 있는 VPC와 어떤 서브넷과도 연결되지 않는 사용자 지정 라우팅 테이블(라우팅 테이블 B)을 보여줍니다.



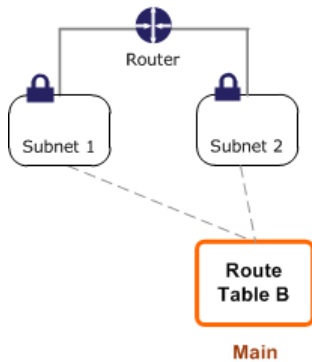
서브넷 2와 라우팅 테이블 B 사이에 명시적 연결을 만들 수 있습니다.



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 대상 주소가 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.

대상 주소	대상
10.0.0.0/16	로컬
172.31.0.0/16	pcx-1a2b3c4d
0.0.0.0/0	igw-11aa22bb

가상 프라이빗 게이트웨이를 VPC에 연결하고 라우팅 테이블에서 경로 전파를 활성화한 경우, 연결을 나타내는 경로는 라우팅 테이블에 전파된 경로로 자동으로 나타납니다. 자세한 정보는 [라우팅 테이블 및 VPN 라우팅 우선순위](#) 단원을 참조하십시오.

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 게이트웨이로 라우팅됩니다.

대상 주소	대상
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
172.31.0.0/16	pcx-1a2b3c4d
0.0.0.0/0	igw-11aa22bb
::/0	eigw-aabb1122

라우팅 옵션

다음 주제에서는 VPC의 특정 게이트웨이 또는 연결을 위한 라우팅을 설명합니다.

옵션

- [인터넷 게이트웨이를 위한 라우팅 테이블 \(p. 194\)](#)
- [NAT 디바이스를 위한 라우팅 테이블 \(p. 194\)](#)
- [가상 프라이빗 게이트웨이를 위한 라우팅 테이블 \(p. 194\)](#)
- [VPC 피어링 연결을 위한 라우팅 테이블 \(p. 195\)](#)
- [ClassicLink에 대한 라우팅 테이블 \(p. 196\)](#)
- [VPC 엔드포인트를 위한 라우팅 테이블 \(p. 196\)](#)
- [외부 전용 인터넷 게이트웨이에 대한 라우팅 테이블 \(p. 197\)](#)

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 대상 주소가 0.0.0.0/0인 경로나 IPv6 트래픽에 대한 대상 주소가 ::/0인 경로를 추가하고, 인터넷 게이트웨이 ID(igw-xxxxxxx)의 대상을 추가합니다. 자세한 정보는 [인터넷 게이트웨이 \(p. 202\)](#) 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 정보는 [NAT 게이트웨이 \(p. 211\)](#) 및 [NAT 인스턴스 \(p. 226\)](#) 단원을 참조하십시오. NAT 디바이스는 IPv6 트래픽에는 사용할 수 없습니다.

가상 프라이빗 게이트웨이를 위한 라우팅 테이블

연결을 사용하여 VPC의 인스턴스를 사용자의 네트워크와 통신하도록 할 수 있습니다. 이렇게 하려면 가상 프라이빗 게이트웨이를 만들고 VPC에 연결한 다음, 대상 주소가 사용자의 네트워크이며 대상이 가상 프라

이빗 게이트웨이(vgw-xxxxxxx)인 경로를 추가합니다. 그런 다음 연결을 만들고 구성할 수 있습니다. 자세한 정보는 사용 설명서의 [란 무엇입니까?](#) 단원을 참조하십시오.

현재 연결을 통한 IPv6 트래픽은 지원하지 않습니다. 그러나 가상 프라이빗 게이트웨이를 통해 AWS Direct Connect 연결로 라우팅되는 IPv6 트래픽은 지원합니다. 자세한 정보는 [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.

대상 주소	대상
10.0.0.0/16	로컬
172.31.0.0/16	pcx-1a2b1a2b

VPC B의 라우팅 테이블은 다음과 같이 구성됩니다.

대상 주소	대상
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 정보는 [VPC 및 서브넷 \(p. 77\)](#) 단원을 참조하십시오. VPC 간 IPv6 트래픽을 라우팅할 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::/56입니다.
- VPC B: IPv6 CIDR 블록은 2001:db8:5678:2b00::/56입니다.

VPC 피어링 연결을 통해 IPv6 통신을 할 수 있도록 하려면 VPC A에 대한 라우팅 테이블에 다음 경로를 추가합니다.

대상 주소	대상
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.

대상 주소	대상
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 정보는 [Amazon VPC Peering Guide](#)를 참조하십시오.

ClassicLink에 대한 라우팅 테이블

ClassicLink는 EC2-Classic 인스턴스를 VPC에 연결함으로써 프라이빗 IPv4 주소를 사용하여 EC2-Classic 인스턴스와 VPC의 인스턴스 간의 통신을 허용하는 기능입니다. For more information about ClassicLink, see [ClassicLink \(p. 282\)](#).

VPC에서 ClassicLink를 활성화하면 모든 VPC 라우팅 테이블에 대상 주소가 10.0.0.0/8, 대상이 local인 경로가 추가됩니다. 따라서 VPC의 인스턴스와 VPC에 링크된 EC2-Classic 인스턴스 간에 통신이 가능합니다. ClassicLink 가능 VPC에 다른 라우팅 테이블을 추가하면 VPC는 대상 주소가 10.0.0.0/8, 대상이 local인 경로를 자동으로 수신합니다. VPC에서 ClassicLink를 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 ClassicLink에 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 ClassicLink를 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-Classic 인스턴스 간에 통신할 수 있도록 VPC 피어링 연결을 수정하면, 대상 주소가 10.0.0.0/8이고 대상이 local인 라우팅 테이블에 고정 경로가 자동으로 추가됩니다. VPC에 연결된 로컬 EC2-Classic 인스턴스와 피어 VPC의 인스턴스 간에 통신할 수 있도록 VPC 피어링 연결을 수정하는 경우, 대상 주소가 피어 VPC CIDR 블록이고 대상이 VPC 피어링 연결인 기본 라우팅 테이블에 경로를 수동으로 추가해야 합니다. EC2-Classic 인스턴스는 피어 VPC로 라우팅하는 데 기본 라우팅 테이블을 사용합니다. 자세한 정보는 Amazon VPC Peering Guide의 [ClassicLink로 구성](#)을 참조하십시오.

VPC 엔드포인트를 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 대상 주소 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 [게이트웨이 엔드포인트의 라우팅 \(p. 260\)](#) 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 정보는 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 단원을 참조하십시오.

라우팅 테이블 작업

이 작업에서는 라우팅 테이블로 작업하는 방법을 보여줍니다.

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.

작업

- [서브넷이 연결되어 있는 라우팅 테이블 확인 \(p. 197\)](#)
- [테이블과 명시적으로 연결되어 있는 서브넷 확인 \(p. 197\)](#)
- [사용자 지정 라우팅 테이블 생성 \(p. 198\)](#)
- [라우팅 테이블에 경로 추가 및 라우팅 테이블에서 경로 제거 \(p. 198\)](#)
- [경로 전파 활성화 및 비활성화 \(p. 198\)](#)
- [서브넷을 라우팅 테이블과 연결 \(p. 199\)](#)
- [서브넷의 라우팅 테이블 변경 \(p. 199\)](#)
- [라우팅 테이블에서 서브넷 연결 끊기 \(p. 199\)](#)
- [기본 라우팅 테이블 바꾸기 \(p. 200\)](#)
- [라우팅 테이블 삭제 \(p. 200\)](#)

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 서브넷을 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 \(p. 197\)](#) 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. 200) 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 대상 주소 CIDR 블록 또는 단일 IP 주소를 바꾼 다음, [대상]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 정보는 사용 설명서의 [라우팅 옵션](#)을 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 정보는 [Amazon VPC에 액세스 \(p. 7\)](#)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)

연결과 관련된 고정 경로 생성

- [create-vpn-connection-route](#)(AWS CLI)
- [New-EC2VpnConnectionRoute](#)(Windows PowerShell용 AWS 도구)

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 할당된 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.

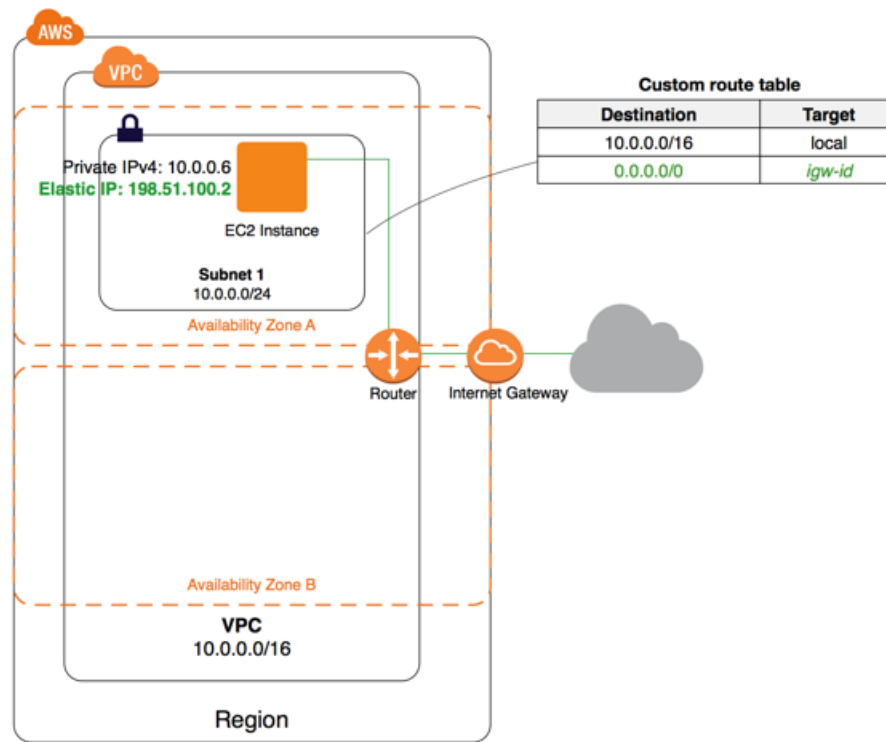
- VPC에 인터넷 게이트웨이를 연결합니다.
- 서브넷의 라우팅 테이블이 인터넷 게이트웨이를 가리키는 지 확인합니다.
- 서브넷의 인스턴스에 전역적으로 고유한 IP 주소(퍼블릭 IPv4 주소, 탄력적 IP 주소 또는 IPv6 주소)가 있는지 확인합니다.
- 네트워크 액세스 제어 및 보안 그룹 규칙에서 적절한 트래픽이 인스턴스로, 그리고 인스턴스에서 흐르도록 허용되는 지 확인합니다.

인터넷 게이트웨이를 사용하려면 서브넷의 라우팅 테이블에 인터넷에 바인딩된 트래픽을 인터넷 게이트웨이로 전송하는 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.

IPv4 인터넷 통신이 가능하도록 만들려면, 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 회신 주소 필드는 프라이빗 IP 주소가 아니라, 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소(EIP)로 설정됩니다. 반대로, 인스턴스의 퍼블릭 IPv4 주소 또는 탄력적 IP 주소를 대상 주소로 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 테이블에서 VPC가 IPv4 또는 IPv6를 통한 인터넷 액세스에 필요한 구성 요소와 함께 자동으로 제공되는지를 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)

	기본 VPC	기본이 아닌 VPC
서브넷에서 시작된 인스턴스에 자동 할당된 IPv6 주소	아니요(기본 서브넷)	아니요(기본이 아닌 서브넷)

기본 VPC에 대한 자세한 내용은 [기본 VPC 및 기본 서브넷 \(p. 96\)](#) 단원을 참조하십시오. VPC 마법사를 사용하여 인터넷 게이트웨이로 VPC를 만드는 방법에 대한 자세한 내용은 [시나리오 1: 단일 퍼블릭 서브넷을 가진 VPC \(p. 23\)](#) 또는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\) \(p. 30\)](#) 를 참조하십시오.

VPC에서 IP 주소를 지정하고 인스턴스에 퍼블릭 IPv4 또는 IPv6 주소를 할당하는 방식을 제어하는 방법에 대한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 103\)](#) 단원을 참조하십시오.

VPC에 새 서브넷을 추가할 때 해당 서브넷의 라우팅 및 보안 설정을 구성해야 합니다.

인터넷 게이트웨이로 VPC 생성

아래에서는 인터넷 액세스를 지원하기 위해 퍼블릭 서브넷을 수동으로 만드는 방법을 설명합니다.

작업

- [서브넷 만들기 \(p. 204\)](#)
- [인터넷 게이트웨이 생성 및 연결 \(p. 204\)](#)
- [사용자 지정 라우팅 테이블 생성 \(p. 205\)](#)
- [보안 그룹 규칙 업데이트 \(p. 205\)](#)
- [탄력적 IP 주소 추가 \(p. 206\)](#)
- [VPC에서 인터넷 게이트웨이 분리 \(p. 206\)](#)
- [인터넷 게이트웨이 삭제 \(p. 206\)](#)
- [API 및 명령 개요 \(p. 207\)](#)

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 내용은 [VPC 및 서브넷 \(p. 77\)](#) 단원을 참조하십시오.

인터넷 게이트웨이 생성 및 연결

인터넷 게이트웨이를 생성하여 VPC에 연결

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 인터넷 게이트웨이를 선택한 후 인터넷 게이트웨이 생성을 선택합니다.
3. 선택적으로 인터넷 게이트웨이의 이름을 지정한 후 생성을 선택합니다.

4. 방금 생성한 인터넷 게이트웨이를 선택한 후 Actions, Attach to VPC(작업, VPC에 연결)을 선택합니다.
5. 목록에서 VPC를 선택한 다음 연결을 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 대상 주소로 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 대상 주소 상자에서 0.0.0.0/0을 지정하고, 대상 목록에서 인터넷 게이트웨이 ID를 선택합니다.
 - IPv6 트래픽에 대해 대상 주소 상자에서 ::/0을 지정하고, 대상 목록에서 인터넷 게이트웨이 ID를 선택합니다.
6. [Subnet Associations] 탭에서 [Edit]를 선택하고, 서브넷에 대해 [Associate] 확인란을 선택한 다음, [Save]를 선택합니다.

자세한 내용은 [라우팅 테이블 \(p. 190\)](#) 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 인바운드 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 인바운드 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 인스턴스를 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.

자세한 내용은 [VPC의 보안 그룹 \(p. 123\)](#) 단원을 참조하십시오.

탄력적 IP 주소 추가

인스턴스를 서브넷으로 시작한 후, IPv4를 통해 인터넷에서 접속할 수 있도록 하려면 인스턴스에 탄력적 IP 주소를 할당해야 합니다.

Note

시작 중에 퍼블릭 IPv4 주소를 인스턴스에 할당한 경우에는 인터넷에서 인스턴스에 연결할 수 있으므로, 인스턴스에 탄력적 IP 주소를 할당할 필요가 없습니다. 인스턴스에 대한 IP 주소 지정에 관한 자세한 내용은 [VPC의 IP 주소 지정 \(p. 103\)](#) 단원을 참조하십시오.

콘솔을 사용하여 인스턴스에 엘라스틱 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.

자세한 내용은 [탄력적 IP 주소 \(p. 243\)](#) 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 퍼블릭 IP 주소 또는 탄력적 IP 주소가 연결된 리소스가 있는 경우에는 인터넷 게이트웨이를 분리할 수 없습니다.

인터넷 게이트웨이 분리

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 Actions, Detach from VPC(작업, VPC에서 분리)를 선택합니다.
6. VPC에서 분리 대화 상자에서 분리를 선택합니다.

인터넷 게이트웨이 삭제

더 이상 필요하지 않게 된 인터넷 게이트웨이를 삭제할 수 있습니다. 아직 VPC에 연결되어 있는 인터넷 게이트웨이는 삭제할 수 없습니다.

인터넷 게이트웨이 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Internet Gateways]를 선택합니다.
3. 인터넷 게이트웨이를 선택한 다음 작업, 인터넷 게이트웨이 삭제를 선택합니다.
4. 인터넷 게이트웨이 삭제 대화 상자에서 삭제를 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 \(p. 7\)](#) 단원을 참조하십시오.

인터넷 게이트웨이 생성

- [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 \(p. 211\)](#) 단원을 참조하십시오.

콘텐츠

- 외부 전용 인터넷 게이트웨이 기본 사항 (p. 208)
- 외부 전용 인터넷 게이트웨이 작업 (p. 209)
- API 및 CLI 개요 (p. 210)

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.

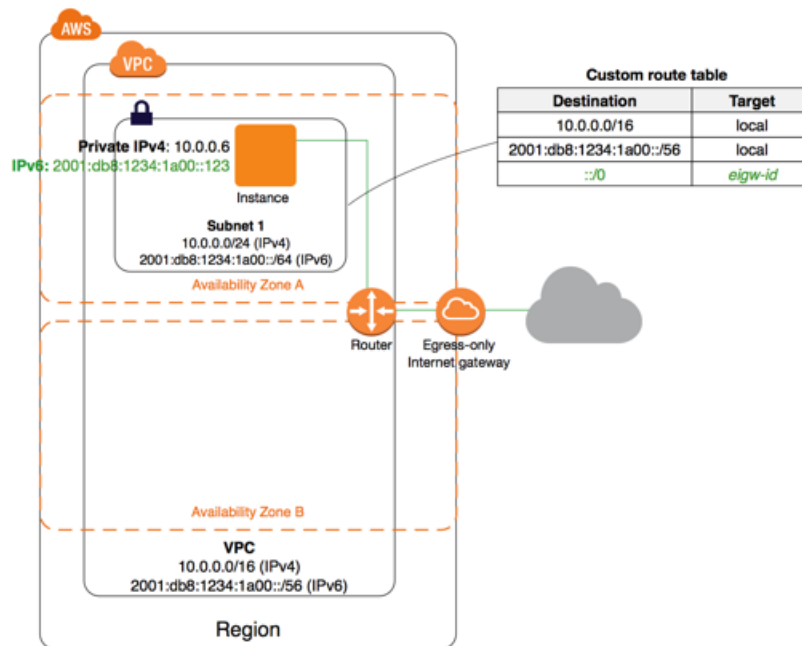
IPv6 주소는 전역적으로 고유하므로 퍼블릭으로 기본 설정되어 있습니다. 인스턴스가 인터넷에 액세스할 수 있게 하되 인터넷 상의 리소스가 해당 인스턴스와의 통신을 시작하지 못하게 하려면 외부 전용 인터넷 게이트웨이를 사용하여 그렇게 할 수 있습니다. 이렇게 하려면 VPC에 외부 전용 인터넷 게이트웨이를 만들어 라우팅 테이블에 모든 IPv6 트래픽(::/0)을 가리키는 경로를 추가하거나 IPv6 주소의 특정 범위를 외부 전용 인터넷 게이트웨이에 추가합니다. 라우팅 테이블에 연결된 서브넷의 IPv6 트래픽은 외부 전용 인터넷 게이트웨이로 라우팅됩니다.

외부 전용 인터넷 게이트웨이는 상태 저장 방식으로, 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 대상 주소로 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, 대상 목록에서 외부 전용 인터넷 게이트웨이 ID를 선택한 다음, Save를 선택합니다.
6. Subnet Associations 탭에서 Edit를 선택하고 서브넷에 대한 Associate 확인란을 선택합니다. Save를 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 \(p. 190\)](#) 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. 삭제를 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 \(p. 7\)](#) 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 \(p. 207\)](#) 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 인스턴스를 제공합니다. NAT 게이트웨이가 NAT 인스턴스보다 우수한 가용성 및 대역폭을 제공하므로 NAT 게이트웨이를 사용하는 것이 좋습니다.

습니다. 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 \(p. 211\)](#)
- [NAT 인스턴스 \(p. 226\)](#)
- [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 233\)](#)

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 정보는 [NAT \(p. 210\)](#)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 정보는 [Amazon VPC 요금](#)을 참조하십시오.

NAT 게이트웨이는 IPv6 트래픽을— 지원하지 않으므로, 그 대신에 외부 전용 인터넷 게이트웨이를 사용하십시오. 자세한 정보는 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 단원을 참조하십시오.

내용

- [NAT 게이트웨이 기본 사항 \(p. 211\)](#)
- [NAT 게이트웨이 사용 \(p. 213\)](#)
- [NAT 게이트웨이 문제 해결 \(p. 216\)](#)
- [NAT 게이트웨이 사용 제어 \(p. 220\)](#)
- [NAT 게이트웨이 태그 지정 \(p. 220\)](#)
- [API 및 CLI 개요 \(p. 221\)](#)
- [Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링 \(p. 221\)](#)

NAT 게이트웨이 기본 사항

NAT 게이트웨이를 만들려면 NAT 게이트웨이가 속할 퍼블릭 서브넷을 지정해야 합니다. 퍼블릭 서브넷과 프라이빗 서브넷에 대한 자세한 정보는 [서브넷 라우팅 \(p. 84\)](#)을 참조하십시오. NAT 게이트웨이를 만들 때 NAT 게이트웨이와 연결할 [탄력적 IP 주소 \(p. 243\)](#)도 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.

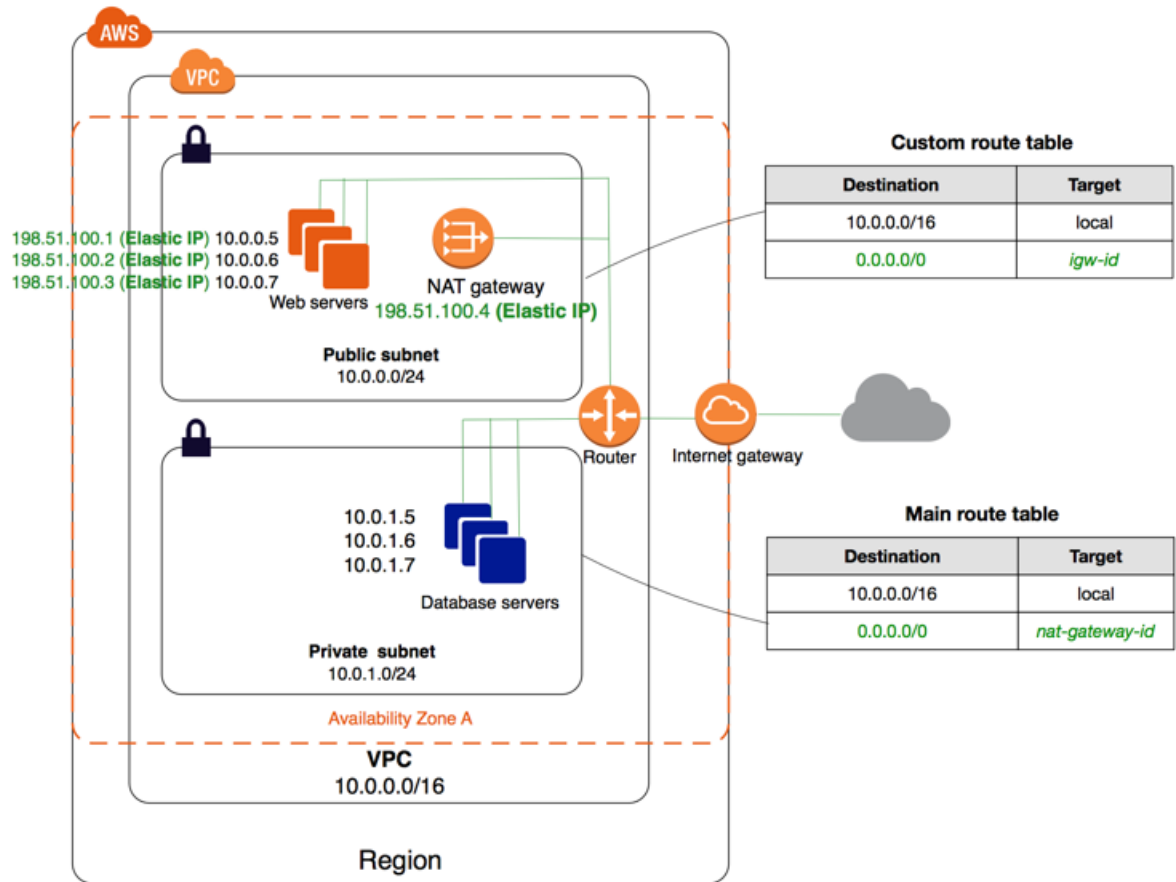
각 NAT 게이트웨이는 특정 가용 영역에 생성되고 해당 영역에서 중복성을 통해 구현됩니다. 가용 영역에서 만들 수 있는 NAT 게이트웨이 수에는 제한이 있습니다. 자세한 정보는 [Amazon VPC 제한 \(p. 285\)](#) 단원을 참조하십시오.

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.

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.



NAT 게이트웨이 규칙 및 제한

NAT 게이트웨이에는 다음과 같은 특성과 제한이 있습니다.

- NAT 게이트웨이는 5Gbps의 대역폭을 지원하며 최대 45Gbps까지 자동 확장합니다. 그 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 사용하려면 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 정보는 [네트워크 ACL \(p. 130\)](#) 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 정보는 [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.
- NAT 게이트웨이는 각 고유 대상에 대해 최대 55,000개의 동시 연결을 지원할 수 있습니다. 단일 대상에 대해 초당 약 900개의 연결(분당 약 55,000개의 연결)을 만들 경우에도 이 제한이 적용됩니다. 대상 IP 주소,

대상 포트 또는 프로토콜(TCP/UDP/ICMP)이 변경되는 경우 55,000개의 연결을 추가로 만들 수 있습니다. 55,000개보다 더 많은 연결에서는 포트 할당 오류로 인해 연결 오류가 발생할 가능성이 증가합니다. NAT 게이트웨이에 대한 `ErrorPortAllocation` CloudWatch 측정치를 통해 이러한 오류를 모니터링할 수 있습니다. 자세한 정보는 [Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링 \(p. 221\)](#) 단원을 참조하십시오.

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 인스턴스에서 현재 사용하는 것과 동일한 탄력적 IP 주소를 NAT 게이트웨이에 사용하려면 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 엔드포인트, , AWS Direct Connect 또는 VPC 피어링을 통해 NAT 게이트웨이 사용

NAT 게이트웨이는 VPC 엔드포인트, 연결, AWS Direct Connect 또는 VPC 피어링 연결을 통해 트래픽을 보낼 수 없습니다. 프라이빗 서브넷의 인스턴스가 VPC 엔드포인트, 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 엔드포인트로 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 엔드포인트 또는 VPC 피어링 연결로 전송됩니다. 인스턴스에서 인터넷(Amazon S3 IP 주소 제외)으로 트래픽을 보내면 트래픽은 NAT 게이트웨이로 전송됩니다.

VPC 피어링 연결, 연결 또는 AWS Direct Connect를 통해 NAT 게이트웨이로 트래픽을 라우팅할 수 없습니다. 이 연결의 다른 쪽에 있는 리소스는 NAT 게이트웨이를 사용할 수 없습니다.

Amazon S3 또는 DynamoDB로 트래픽 전송 시 모범 사례

Amazon S3 및 DynamoDB에 액세스할 때 NAT 게이트웨이에 대한 데이터 처리 요금이 청구되지 않도록, 게이트웨이 엔드포인트를 설정하고 NAT 게이트웨이 대신 게이트웨이 엔드포인트를 통해 트래픽을 라우팅합니다. 게이트웨이 엔드포인트 사용에 대한 요금은 없습니다. 자세한 정보는 [게이트웨이 VPC 엔드포인트 \(p. 258\)](#) 단원을 참조하십시오.

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 정보는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\) \(p. 30\)](#) 단원을 참조하십시오.

작업

- [NAT 게이트웨이 만들기 \(p. 214\)](#)
- [라우팅 테이블 업데이트 \(p. 214\)](#)
- [NAT 게이트웨이 삭제 \(p. 214\)](#)

- [NAT 게이트웨이 테스트 \(p. 215\)](#)

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. NAT 게이트웨이를 만들 서브넷을 지정하고 NAT 게이트웨이와 연결할 탄력적 IP 주소의 할당 ID를 선택합니다. 완료하면 [Create a NAT Gateway]를 선택합니다.
4. NAT 게이트웨이가 콘솔에 표시됩니다. 몇 분 후 상태가 Available로 변경되면 사용할 준비가 된 것입니다.

NAT 게이트웨이가 Failed 상태로 바뀌면 만드는 중에 오류가 발생한 것입니다. 자세한 정보는 [NAT 게이트웨이가 실패 상태로 바뀔 \(p. 217\)](#) 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 정보는 [라우팅 우선순위 \(p. 193\)](#) 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 0.0.0.0/0을 입력합니다. [대상]에서 NAT 게이트웨이의 ID를 선택합니다.

Note

NAT 인스턴스 사용에서 마이그레이션하는 경우 NAT 인스턴스를 가리키는 현재 경로를 NAT 게이트웨이에 대한 경로로 바꿀 수 있습니다.

5. Save를 선택합니다.

NAT 게이트웨이가 인터넷에 액세스할 수 있도록 하려면, NAT 게이트웨이가 속한 서브넷과 연결된 라우팅 테이블에 인터넷 트래픽이 인터넷 게이트웨이를 가리키는 경로가 포함되어야 합니다. 자세한 정보는 [사용자 지정 라우팅 테이블 생성 \(p. 205\)](#) 단원을 참조하십시오. NAT 게이트웨이를 삭제하면 NAT 게이트웨이 경로는 경로를 삭제하거나 업데이트할 때까지 blackhole 상태로 유지됩니다. 자세한 정보는 [라우팅 테이블에 경로 추가 및 라우팅 테이블에서 경로 제거 \(p. 198\)](#) 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 [Actions], [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 \(p. 215\)](#) 단원을 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 홑 중 하나(일반적으로 첫 번째 홑)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 \(p. 216\)](#)을 참조하십시오.

인터넷 연결 테스트

다음 예제는 프라이빗 서브넷의 인스턴스가 인터넷에 연결할 수 있는지 테스트하는 방법을 보여 줍니다.

1. 퍼블릭 서브넷의 인스턴스를 시작합니다(이 인스턴스를 접속 호스트로 사용). 자세한 정보는 [서브넷에서 인스턴스 시작 \(p. 89\)](#) 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 0.0.0.0/0도 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 호스트에 연결합니다. 자세한 정보는 [Linux 또는 macOS에 대한 SSH 에이전트 전달을 구성하려면 \(p. 215\)](#) 또는 [Windows\(PuTTY\)에 대한 SSH 에이전트 전달을 구성하려면 \(p. 216\)](#) 단원을 참조하십시오.
4. 접속 호스트에서 프라이빗 서브넷의 인스턴스에 연결한 다음, 프라이빗 서브넷의 인스턴스에서 인터넷 연결을 테스트합니다. 자세한 정보는 [인터넷 연결을 테스트하려면 \(p. 216\)](#) 단원을 참조하십시오.

Linux 또는 macOS에 대한 SSH 에이전트 전달을 구성하려면

1. 로컬 시스템에서 인증 에이전트에 프라이빗 키를 추가합니다.

Linux의 경우 다음 명령을 사용합니다.

```
ssh-add -c mykeypair.pem
```

macOS의 경우 다음 명령을 사용합니다.

```
ssh-add -K mykeypair.pem
```

2. 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

1. Pageant가 아직 설치되어 있지 않으면 [PuTTY 다운로드 페이지](#)에서 Pageant를 다운로드하여 설치합니다.
2. 프라이빗 키를 .ppk 형식으로 변환합니다. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [PuTTYgen을 사용하여 프라이빗 키 변환](#)을 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 시작하고 퍼블릭 IP 주소를 사용하여 퍼블릭 서브넷의 인스턴스에 연결합니다. 자세한 정보는 [PuTTY 세션 시작](#)을 참조하십시오. [Auth] 범주에서 [Allow agent forwarding] 옵션을 선택하고 [Private key file for authentication] 상자를 공백 상태로 둡니다.

인터넷 연결을 테스트하려면

1. 퍼블릭 서브넷의 인스턴스에서 프라이빗 IP 주소를 사용하여 프라이빗 서브넷의 인스턴스에 연결합니다. 예를 들면 다음과 같습니다.

```
ssh ec2-user@10.0.1.123
```

2. 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 \(p. 218\)](#)을 참조하십시오.

3. (선택 사항) 더 이상 필요하지 않으면 인스턴스를 종료합니다. 자세한 정보는 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.

문제

- NAT 게이트웨이가 실패 상태로 바뀜 (p. 217)
- 탄력적 IP 주소 또는 NAT 게이트웨이 제한에 도달함 (p. 218)
- 가용 영역이 지원되지 않음(NotAvailableInZone) (p. 218)

- NAT 게이트웨이를 만들었지만 게이트웨이가 더 이상 표시되지 않음 (p. 218)
- NAT 게이트웨이가 Ping 명령에 응답하지 않음 (p. 218)
- 프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 없음 (p. 218)
- 특정 엔드포인트에 대한 TCP 연결이 실패 (p. 219)
- 경로 추적 출력에 NAT 게이트웨이 프라이빗 IP 주소가 표시되지 않음 (p. 220)
- 350초 후 인터넷 연결이 끊어짐 (p. 220)
- IPSec 연결을 설정할 수 없음 (p. 220)
- 대상에 대한 추가 연결을 시작할 수 없음 (p. 220)

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 게이트웨이를 선택하고 NAT 게이트웨이를 선택한 다음, 세부 정보 창의 상태 상자에서 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 서브넷 페이지로 이동하여 서브넷에 대한 세부 정보 창의 사용 가능한 IP 상자를 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 정보는 인터넷 게이트웨이 생성 및 연결 (p. 204) 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 연

표시된 오류	이유	수정 단계
		결 해제할 수 있습니다. 또는 계정에 새로운 탄력적 IP 주소를 할당합니다. 자세한 정보는 탄력적 IP 주소 작업 (p. 244) 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 IP 주소를 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.

자세한 정보는 [Amazon VPC 제한 \(p. 285\)](#) 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 Amazon VPC 콘솔에 잠시 동안(일반적으로 한 시간) 표시되며 그 이후 자동으로 삭제됩니다. [NAT 게이트웨이가 실패 상태로 바뀜 \(p. 217\)](#)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 \(p. 215\)](#)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 정보는 [NAT 게이트웨이가 실패 상태로 바뀔 \(p. 217\)](#) 단원을 참조하십시오.
- 라우팅 테이블을 올바르게 구성했는지 확인합니다:
 - NAT 게이트웨이는 인터넷 트래픽을 인터넷 게이트웨이로 라우팅하는 라우팅 테이블이 있는 퍼블릭 서브넷에 있어야 합니다. 자세한 정보는 [사용자 지정 라우팅 테이블 생성 \(p. 205\)](#) 단원을 참조하십시오.
 - 인스턴스는 인터넷 트래픽을 NAT 게이트웨이로 라우팅하는 라우팅 테이블이 있는 프라이빗 서브넷에 있어야 합니다. 자세한 정보는 [라우팅 테이블 업데이트 \(p. 214\)](#) 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 정보는 [VPC 흐름 로그 \(p. 173\)](#) 단원을 참조하십시오.

- ping 명령을 사용하는 경우 ICMP가 활성화된 웹 사이트를 ping하고 있는지 확인합니다. 그렇지 않으면 회신 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 정보는 [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 233\)](#) 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, `traceroute` 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.

350초 후 인터넷 연결이 끊어짐

NAT 게이트웨이를 사용하는 연결이 350초 이상 유휴 상태인 경우 연결이 시간 초과됩니다. 연결이 끊어지지 않게 하려면 연결을 통해 추가 트래픽을 시작하거나 인스턴스에서 TCP keepalive를 350초 미만의 값으로 활성화하면 됩니다.

IPSec 연결을 설정할 수 없음

현재 NAT 게이트웨이는 IPsec 프로토콜을 지원하지 않지만, NAT-Traversal(NAT-T)을 사용하여 NAT 게이트웨이에 대해 지원되는 프로토콜인 UDP의 IPsec 트래픽을 캡슐화할 수 있습니다. NAT-T 및 IPsec 구성을 테스트하여 IPsec 트래픽이 삭제되지 않는지 확인하십시오.

대상에 대한 추가 연결을 시작할 수 없음

동시 연결 제한에 도달했을 수 있습니다. 자세한 정보는 [NAT 게이트웨이 규칙 및 제한 \(p. 212\)](#) 단원을 참조하십시오. 프라이빗 서브넷의 인스턴스가 많은 수의 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.

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. IAM의 Amazon VPC 정책에 대한 자세한 정보는 [Amazon VPC 리소스에 대한 액세스 제어 \(p. 159\)](#) 단원을 참조하십시오.

NAT 게이트웨이 태그 지정

NAT 게이트웨이에 태그를 지정하면 조직의 요구에 따라 이를 식별 또는 분류할 수 있습니다. 태그 사용에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 리소스에 태그 지정](#)을 참조하십시오.

NAT 게이트웨이에서 비용 할당 태그가 지원되므로 태그를 사용하여 AWS 청구서를 구성하고 자체 비용 구조를 반영할 수 있습니다. 자세한 정보는 AWS Billing and Cost Management 사용 설명서의 [비용 할당 태그](#)

사용을 참조하십시오. 태그를 사용한 비용 할당 보고서 설정에 대한 자세한 정보는 AWS 계정 결제 정보의 [월간 비용 할당 보고서](#) 섹션을 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 정보는 [Amazon VPC에 액세스 \(p. 7\)](#)를 참조하십시오.

NAT 게이트웨이 만들기

- [create-nat-gateway](#)(AWS CLI)
- [New-EC2NatGateway](#)(Windows PowerShell용 AWS 도구)
- [CreateNatGateway](#)(Amazon EC2 쿼리 API)

NAT 게이트웨이 태그 지정

- [create-tags](#)(AWS CLI)
- [New-EC2Tag](#)(Windows PowerShell용 AWS 도구)
- [CreateTags](#)(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)

Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링

CloudWatch를 이용하여 NAT 게이트웨이를 모니터링함으로써 NAT 게이트웨이에 대한 정보를 수집하고 읽기 가능한 실시간에 가까운 지표를 만들 수 있습니다. 이 정보를 사용하여 NAT 게이트웨이를 모니터링하고 문제를 해결할 수 있습니다. NAT 게이트웨이 지표 데이터는 1분마다 제공되며, 통계는 15개월 동안 기록됩니다.

Amazon CloudWatch에 대한 자세한 정보는 [Amazon CloudWatch 사용 설명서](#) 단원을 참조하십시오. 요금에 대한 자세한 정보는 [Amazon CloudWatch 요금](#)을 참조하십시오.

NAT 게이트웨이 지표 및 차원

NAT 게이트웨이에 사용할 수 있는 측정치는 아래와 같습니다.

측정치	설명
ActiveConnectionCount	NAT 게이트웨이를 통한 동시 활성 TCP 연결의 총 수입니다. 0의 값은 NAT 게이트웨이를 통한 활성 연결이 없음을 나타냅니다.

측정치	설명
	<p>단위: 개수</p> <p>통계: 가장 유용한 통계는 Max입니다.</p>
BytesInFromDestination	<p>NAT 게이트웨이가 대상으로부터 수신한 바이트 수입니다.</p> <p>BytesOutToSource 값이 BytesInFromDestination 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수 있습니다.</p> <p>단위: 바이트</p> <p>통계: 가장 유용한 통계는 Sum입니다.</p>
BytesInFromSource	<p>NAT 게이트웨이가 VPC 내 클라이언트로부터 수신한 바이트 수입니다.</p> <p>BytesOutToDestination 값이 BytesInFromSource 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수 있습니다.</p> <p>단위: 바이트</p> <p>통계: 가장 유용한 통계는 Sum입니다.</p>
BytesOutToDestination	<p>NAT 게이트웨이를 통해 대상으로 전송된 바이트 수입니다.</p> <p>0보다 큰 값은 NAT 게이트웨이 뒤에 있는 클라이언트에서 인터넷으로 가는 트래픽이 있음을 나타냅니다. BytesOutToDestination 값이 BytesInFromSource 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수 있습니다.</p> <p>단위: 바이트</p> <p>통계: 가장 유용한 통계는 Sum입니다.</p>
BytesOutToSource	<p>NAT 게이트웨이를 통해 VPC 내 클라이언트로 전송된 바이트 수입니다.</p> <p>0보다 큰 값은 인터넷에서 NAT 게이트웨이 뒤에 있는 클라이언트로 오는 트래픽이 있음을 나타냅니다. BytesOutToSource 값이 BytesInFromDestination 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수 있습니다.</p> <p>단위: 바이트</p> <p>통계: 가장 유용한 통계는 Sum입니다.</p>

측정치	설명
ConnectionAttemptCount	<p>NAT 게이트웨이를 통해 이루어진 연결 시도 횟수.</p> <p>ConnectionEstablishedCount 값이 ConnectionAttemptCount 값보다 작은 경우, NAT 게이트웨이 뒤의 클라이언트가 응답이 없는 새 연결을 시도했음을 나타냅니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
ConnectionEstablishedCount	<p>NAT 게이트웨이를 통해 설정된 연결 수.</p> <p>ConnectionEstablishedCount 값이 ConnectionAttemptCount 값보다 작은 경우, NAT 게이트웨이 뒤의 클라이언트가 응답이 없는 새 연결을 시도했음을 나타냅니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
ErrorPortAllocation	<p>NAT 게이트웨이가 소스 포트 할당에 실패한 횟수.</p> <p>0보다 큰 값은 너무 많은 동시 연결이 NAT 게이트웨이를 통해 열려 있음을 나타냅니다.</p> <p>단위: 개수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
IdleTimeoutCount	<p>활성 상태가 유휴 상태로 전환된 연결 수입니다. 활성 연결은 적절하게 종료되지 않고 직전 350초 동안 활동이 없는 경우 유휴 상태로 전환됩니다.</p> <p>0보다 큰 값은 유휴 상태로 이동된 연결이 있었음을 나타냅니다. IdleTimeoutCount 값이 증가하는 경우, NAT 게이트웨이 뒤의 클라이언트가 부실 연결을 재사용하고 있음을 나타낼 수 있습니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
PacketsDropCount	<p>NAT 게이트웨이가 삭제한 패킷 수입니다.</p> <p>0보다 큰 값은 NAT 게이트웨이에 일시적 문제가 있음을 나타낼 수 있습니다. 이 값이 높다면 AWS 서비스 상태 대시보드를 참조하십시오.</p> <p>단위: 개수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>

측정치	설명
<code>PacketsInFromDestination</code>	<p>NAT 게이트웨이가 대상으로부터 수신한 패킷 수입니다.</p> <p><code>PacketsOutToSource</code> 값이 <code>PacketsInFromDestination</code> 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수 있습니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
<code>PacketsInFromSource</code>	<p>NAT 게이트웨이가 VPC 내 클라이언트로부터 수신한 패킷 수입니다.</p> <p><code>PacketsOutToDestination</code> 값이 <code>PacketsInFromSource</code> 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수 있습니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
<code>PacketsOutToDestination</code>	<p>NAT 게이트웨이를 통해 대상으로 전송된 패킷 수입니다.</p> <p>0보다 큰 값은 NAT 게이트웨이 뒤에 있는 클라이언트에서 인터넷으로 가는 트래픽이 있음을 나타냅니다. <code>PacketsOutToDestination</code> 값이 <code>PacketsInFromSource</code> 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수 있습니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>
<code>PacketsOutToSource</code>	<p>NAT 게이트웨이를 통해 VPC 내 클라이언트로 전송된 패킷 수입니다.</p> <p>0보다 큰 값은 인터넷에서 NAT 게이트웨이 뒤에 있는 클라이언트로 오는 트래픽이 있음을 나타냅니다. <code>PacketsOutToSource</code> 값이 <code>PacketsInFromDestination</code> 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수 있습니다.</p> <p>단위: 수</p> <p>통계: 가장 유용한 통계는 sum입니다.</p>

지표 데이터를 필터링하려면 다음 차원을 사용하십시오.

차원	설명
NatGatewayId	NAT 게이트웨이 ID를 기준으로 측정치 데이터를 필터링합니다.

NAT 게이트웨이 CloudWatch 지표 보기

NAT 게이트웨이 지표는 1분 가격으로 CloudWatch로 전송됩니다. NAT 게이트웨이에 대해 다음과 같이 측정치를 볼 수 있습니다.

CloudWatch 콘솔을 사용한 메트릭 확인

측정치는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다.

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [All metrics]에서 [NAT gateway] 지표 네임스페이스를 선택합니다.
4. 지표를 보려면 지표 차원을 선택합니다.

AWS CLI(를) 사용하여 지표를 보려면

명령 프롬프트에서 다음 명령을 사용하여 NAT 게이트웨이 서비스에서 사용 가능한 모든 지표 목록을 확인합니다.

```
aws cloudwatch list-metrics --namespace "AWS/NATGateway"
```

NAT 게이트웨이 모니터링을 위한 CloudWatch 경고 만들기

경보가 상태를 변경하면 Amazon SNS 메시지를 보내는 CloudWatch 경보를 생성할 수 있습니다. 경보는 지정한 기간 동안 단일 지표를 감시합니다. 경보는 기간 수에 대한 주어진 임계값과 지표 값을 비교하여 Amazon SNS 주제에 알리를 보냅니다.

예를 들어 NAT 게이트웨이로 들어오거나 나가는 트래픽의 양을 모니터링하는 경보를 만들 수 있습니다. 아래 경보는 NAT 게이트웨이를 통해 VPC의 클라이언트에서 인터넷으로 가는 아웃바운드 트래픽의 양을 모니터링합니다. 그리고 15분 동안 바이트 수가 임계값인 5,000,000에 도달하면 알리를 보냅니다.

NAT 게이트웨이를 통한 아웃바운드 트래픽에 대한 경보를 만들려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. [NAT gateway]를 선택합니다.
4. NAT 게이트웨이와 [BytesOutToDestination] 지표를 선택하고 [Next]를 선택합니다.
5. 다음과 같이 경보를 구성하고 구성을 완료하면 [Create Alarm]을 선택합니다.
 - [Alarm Threshold]에서 경보의 이름 및 설명을 입력합니다. [Whenever]에서 [\geq]를 선택하고 [5000000]을 입력합니다. 연속 기간에 대해 [1]을 입력합니다.
 - [Actions]에서 기존 알림 목록을 선택하거나 [New list]를 선택하여 새로운 목록을 생성합니다.
 - [Alarm Preview]에서 15분을 선택하고 [Sum] 통계를 지정합니다.

ErrorPortAllocation 지표를 모니터링하는 경보를 만들고 이 값이 3회 연속 5분간 0보다 클 경우에 알리를 보낼 수 있습니다.

경보를 만들어 포트 할당 오류를 모니터링하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. [NAT Gateway]를 선택합니다.
4. NAT 게이트웨이와 [ErrorPortAllocation] 지표를 선택하고 [Next]를 선택합니다.
5. 다음과 같이 경보를 구성하고 구성을 완료하면 [Create Alarm]을 선택합니다.
 - [Alarm Threshold]에서 경보의 이름 및 설명을 입력합니다. [Whenever]에서 [>]를 선택하고 0을 입력합니다. 연속 기간에 대해 [3]을 입력합니다.
 - [Actions]에서 기존 알림 목록을 선택하거나 [New list]를 선택하여 새로운 목록을 생성합니다.
 - [Alarm Preview]에서 5분을 선택하고 [Maximum] 통계를 지정합니다.

경보를 만드는 더 많은 예를 보려면 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch경보 생성](#)을 참조하십시오.

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 정보는 [서브넷 라우팅 \(p. 84\)](#)을 참조하십시오. NAT에 대한 자세한 정보는 [NAT \(p. 210\)](#)를 참조하십시오.

NAT는 IPv6 트래픽을— 지원하지 않으므로, 그 대신에 외부 전용 인터넷 게이트웨이를 사용하십시오. 자세한 정보는 [외부 전용 인터넷 게이트웨이 \(p. 207\)](#) 단원을 참조하십시오.

Note

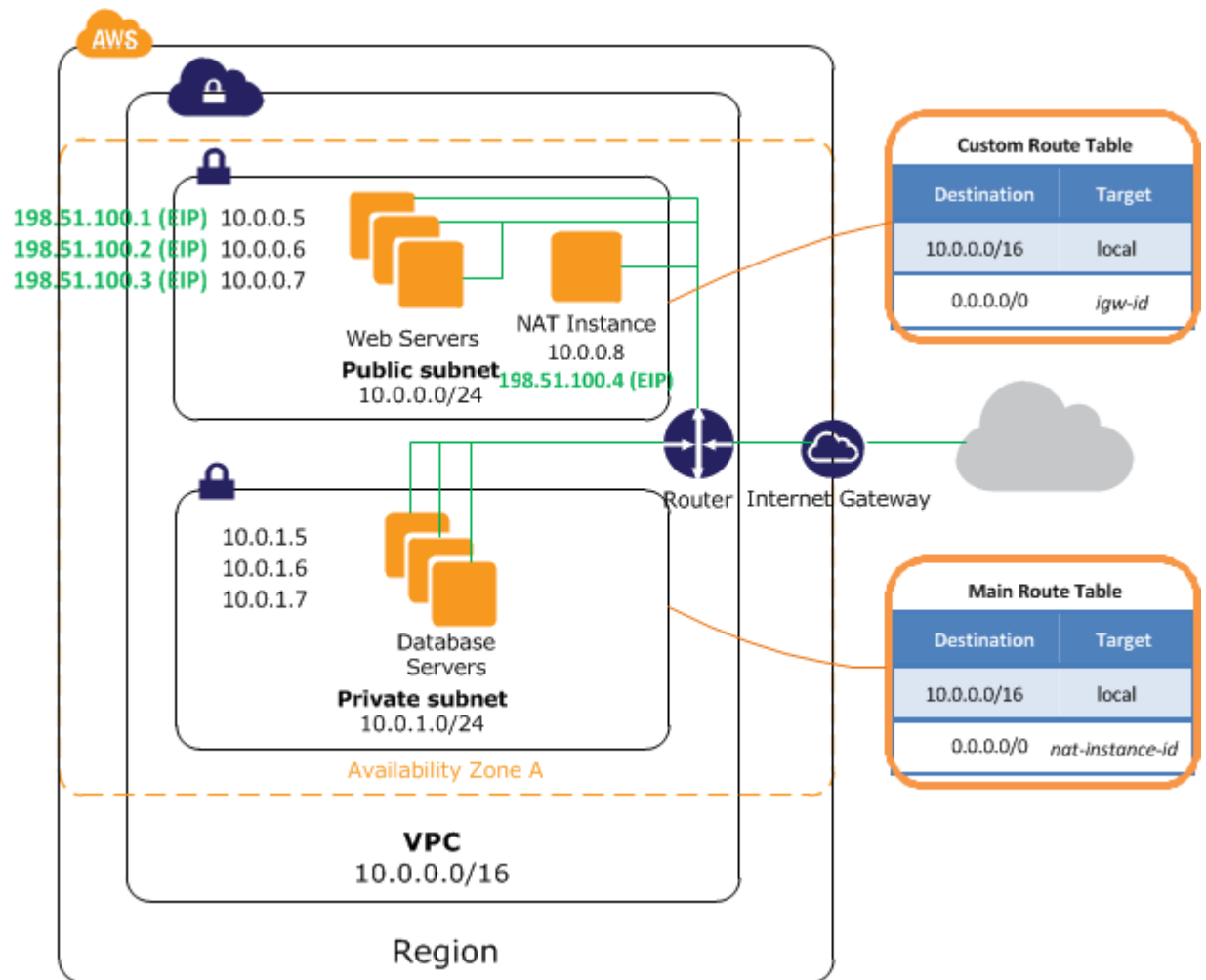
더 나은 가용성과 향상된 대역폭을 제공하면서도 관리 작업은 간소화하는 관리형 NAT 서비스인 NAT 게이트웨이를 사용할 수도 있습니다. 일반 사용 사례에서는 NAT 인스턴스보다 NAT 게이트웨이를 사용하는 것이 좋습니다. 자세한 정보는 [NAT 게이트웨이 \(p. 211\)](#) 및 [NAT 인스턴스 및 NAT 게이트웨이 비교 \(p. 233\)](#) 단원을 참조하십시오.

내용

- [NAT 인스턴스 기본 사항 \(p. 226\)](#)
- [NAT 인스턴스 설정 \(p. 228\)](#)
- [NATSG 보안 그룹 생성 \(p. 229\)](#)
- [원본/대상 확인 비활성화 \(p. 230\)](#)
- [기본 라우팅 테이블 업데이트 \(p. 231\)](#)
- [NAT 인스턴스 구성 테스트 \(p. 231\)](#)

NAT 인스턴스 기본 사항

다음 그림에서는 NAT 인스턴스 기본 사항을 보여줍니다. 기본 라우팅 테이블은 프라이빗 서브넷과 연결되며 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 매스캐레이딩을 구성.

Note

구성 업데이트를 활용하려면 NAT AMI의 최신 버전을 사용하는 것이 좋습니다.
VPC에서 보조 IPv4 CIDR 블록을 추가 및 제거하는 경우에는 AMI 버전 `amzn-ami-vpc-nat-hvm-2017.03.1.20170623-x86_64-ebs` 이상을 사용해야 합니다.

NAT 인스턴스 제한은 해당 리전의 인스턴스 유형 제한에 따라 다릅니다. 자세한 정보는 [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 정보는 [시나리오 2: 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC\(NAT\)](#) (p. 30) 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. 86) 참조).
 - b. 두 개의 서브넷을 생성합니다([서브넷 만들기](#) (p. 204) 참조).
 - c. 인터넷 게이트웨이를 VPC에 연결합니다([인터넷 게이트웨이 생성 및 연결](#) (p. 204) 참조).
 - d. VPC 외부 위치를 대상 주소로 하는 트래픽을 인터넷 게이트웨이로 보내는 사용자 지정 라우팅 테이블을 생성한 다음, 이를 한 서브넷과 연결하여 퍼블릭 서브넷으로 만듭니다([사용자 지정 라우팅 테이블 생성](#) (p. 205) 참조).
2. NATSG 보안 그룹을 생성합니다([NATSG 보안 그룹 생성](#) (p. 229) 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 배정할 수 있습니다. 시작 시 퍼블릭 IP 배정에 대한 자세한 정보는 [인스턴스 시작 시 퍼블릭 IPv4 주소 배정](#) (p. 107) 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. 검토 및 시작을 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. 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EBS 지원 AMI 생성](#)을 참조하십시오.
 5. NAT 인스턴스에 대한 `SrcDestCheck` 속성을 비활성화합니다([원본/대상 확인 비활성화](#) (p. 230) 참조).

6. 시작 도중에(3단계) 퍼블릭 IP 주소를 NAT 인스턴스에 지정하지 않았다면, 엘라스틱 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 정보는 [기본 라우팅 테이블 업데이트 \(p. 231\)](#) 단원을 참조하십시오.

명령줄을 사용하여 NAT 인스턴스 시작

서브넷으로 NAT 인스턴스를 시작하려면 다음 명령 중 하나를 사용합니다. 자세한 정보는 [Amazon VPC에 액세스 \(p. 7\)](#) 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.

```
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

Inbound			
Source	Protocol	Port Range	Comments
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)
Outbound			
Destination	Protocol	Port Range	Comments
0.0.0.0/0	TCP	80	IPv4를 통해 인터넷에 접근하는 아웃바운드 HTTP

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. Save를 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. Save를 선택합니다.

자세한 정보는 [VPC의 보안 그룹 \(p. 123\)](#) 단원을 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 인스턴스를 선택합니다.
3. NAT 인스턴스를 선택하고 작업, 네트워킹, 소스/대상 확인 Check]를 선택합니다.
4. NAT 인스턴스에 대해 이 속성이 비활성화되어 있는지 확인합니다. 그렇지 않으면 [Yes, Disable]을 선택합니다.
5. NAT 인스턴스에 보조 네트워크 인터페이스가 있다면 설명 탭의 네트워크 인터페이스에서 선택한 다음 인터페이스 ID를 선택해 네트워크 인터페이스 페이지로 이동합니다. 작업, 소스/대상 확인 을 차례로 선택하고 설정을 비활성화한 다음 저장을 선택합니다.

명령줄을 사용하여 원본/대상 확인을 비활성화하려면

다음 명령 중 하나를 사용할 수 있습니다. 자세한 정보는 [Amazon VPC에 액세스 \(p. 7\)](#) 단원을 참조하십시오.

- `modify-instance-attribute`(AWS CLI)
- `Edit-EC2InstanceAttribute`(Windows PowerShell용 AWS 도구)

기본 라우팅 테이블 업데이트

VPC의 프라이빗 서브넷은 사용자 지정 라우팅 테이블과 연결되지 않으므로 기본 라우팅 테이블을 사용합니다. 기본적으로, 기본 라우팅 테이블을 통해 VPC의 인스턴스가 서로 통신할 수 있습니다. NAT 인스턴스로 다른 모든 서브넷 트래픽을 보내는 경로를 추가해야 합니다.

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([Main] 열이 [Yes]로 표시되어 있음). 세부 정보 창에는 경로, 연결 및 경로 전파 작업을 위한 탭이 표시됩니다.
4. [Routes] 탭에서 [Edit]를 선택하고, [Destination] 상자에서 0.0.0.0/0을 지정하고, [대상] 목록에서 NAT 인스턴스의 인스턴스 ID를 선택한 다음, [Save]를 선택합니다.
5. [Subnet Associations] 탭에서 [Edit]를 선택한 다음, 서브넷에 대한 [Associate] 확인란을 선택합니다. Save를 선택합니다.

자세한 정보는 [라우팅 테이블 \(p. 190\)](#) 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. 규칙 추가를 선택하고, 유형 목록에서 모든 ICMP - IPv4를, 소스 목록에서 사용자 지정을 선택합니다. 프라이빗 서브넷의 IP 주소 범위를 입력합니다(예: 10.0.1.0/24). Save를 선택합니다.
5. [Outbound] 탭에서 [Edit]를 선택합니다.
6. 규칙 추가를 선택하고, 유형 목록에서 SSH를, 대상 목록에서 사용자 지정을 선택합니다. 프라이빗 서브넷의 IP 주소 범위를 입력합니다(예: 10.0.1.0/24). Save를 선택합니다.
7. 규칙 추가를 선택하고, 유형 목록에서 모든 ICMP - IPv4를, 대상 목록에서 사용자 지정을 선택합니다. 0.0.0.0/0을 입력한 다음 [Save]를 선택합니다.

프라이빗 서브넷으로 인스턴스를 시작하려면

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

2. 탐색 창에서 인스턴스를 선택합니다.
3. 프라이빗 서브넷으로 인스턴스를 시작합니다. 자세한 정보는 [서브넷에서 인스턴스 시작 \(p. 89\)](#) 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 지정하지 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.

```
ssh-add -c mykeypair.pem
```

OS X의 경우 다음 명령을 사용합니다.

```
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 정보는 [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.

```
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.

```
ssh ec2-user@10.0.1.123
```

3. 프라이빗 인스턴스에서 ping 명령을 실행하여 인터넷에 연결할 수 있는지 테스트합니다.

```
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 정보는 [기본 라우팅 테이블 업데이트 \(p. 231\)](#) 단원을 참조하십시오.
 - NAT 인스턴스에 대해 원본/대상 확인을 비활성화했는지 확인합니다. 자세한 정보는 [원본/대상 확인 비활성화 \(p. 230\)](#) 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 정보는 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 45Gbps까지 확장할 수 있습니다.	인스턴스 유형의 대역폭에 따라 다릅니다.
유지 관리	AWS에서 관리합니다. 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	NAT 게이트웨이의 CloudWatch 지표 (p. 221) 를 확인합니다.	인스턴스의 CloudWatch 지표를 확인합니다.
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

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.

가상 사설 클라우드(VPC)의 DHCP 옵션 세트를 구성할 수 있습니다.

콘텐츠

- [DHCP 옵션 세트 개요 \(p. 235\)](#)

- [Amazon DNS 서버 \(p. 236\)](#)
- [DHCP 옵션 변경 \(p. 236\)](#)
- [DHCP 옵션 세트를 사용한 작업 \(p. 237\)](#)
- [API 및 명령 개요 \(p. 238\)](#)

DHCP 옵션 세트 개요

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. 최대 4개의 도메인 이름 서버를 지정할 수 있지만 일부 운영 체제에서는 더 낮은 제한이 적용될 수 있습니다.</p> <p>인스턴스가 domain-name에 지정된 사용자 지정 DNS 호스트 이름을 받으려면, domain-name-servers를 사용자 지정 DNS 서버로 설정해야 합니다.</p>
domain-name	<p>us-east-1에서 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>example.com</code>). 이 값은 정규화되지 않은 DNS 호스트 이름을 완성하는 데 사용됩니다. VPC의 DNS 호스트 이름과 DNS 지원에 대한 자세한 내용은 VPC와 함께 DNS 사용 (p. 239)을 참조하십시오.</p> <p>Important</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>

DHCP 옵션 이름	설명
ntp-servers	최대 4개의 NTP(Network Time Protocol) 서버의 IP 주소입니다. 자세한 내용은 RFC 2132 의 단원 8.3을 참조하십시오.
netbios-name-servers	최대 4개의 NetBIOS 이름 서버의 IP 주소입니다.
netbios-node-type	NetBIOS 노드 유형(1, 2, 4 또는 8)입니다. 2(지점 간 또는 P-노드)를 지정하는 것이 좋습니다. 브로드캐스트 및 멀티캐스트는 현재 지원되지 않습니다. 이러한 노드 유형에 대한 자세한 내용은 RFC 2132 의 단원 8.7 및 RFC1001 의 단원 10을 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. IPv4 CIDR 블록이 여러 개인 VPC의 경우, DNS 서버의 IP 주소가 기본 CIDR 블록에 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 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. 자세한 내용은 Amazon EMR 관리 안내서의 [클러스터를 호스팅하기 위한 VPC 설정](#)을 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 갱신하는 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 단원에서는 DHCP 옵션 세트로 작업하는 방법을 설명합니다.

작업

- [DHCP 옵션 세트 생성 \(p. 237\)](#)
- [VPC가 사용하는 DHCP 옵션 세트 변경 \(p. 237\)](#)
- [VPC가 아무런 DHCP 옵션도 사용하지 못하도록 변경 \(p. 238\)](#)
- [DHCP 옵션 세트 삭제 \(p. 238\)](#)

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 \(p. 237\)](#) 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 \(p. 238\)](#) 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 \(p. 237\)](#) 단원을 참조하십시오.

VPC와 연결된 DHCP 옵션 세트를 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Your VPCs]를 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 VPCs]를 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 \(p. 7\)](#) 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

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.

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 \(p. 103\)](#) 단원을 참조하십시오.

당사에서는 Amazon DNS 서버를 제공합니다. 자체 DNS 서버를 사용하려면 자신의 VPC에 대한 새로운 DHCP 옵션 세트를 생성하십시오. 자세한 내용은 [DHCP 옵션 세트 \(p. 234\)](#) 단원을 참조하십시오.

콘텐츠

- [DNS 호스트 이름 \(p. 239\)](#)
- [VPC에서 DNS 지원 \(p. 239\)](#)
- [DNS 제한 \(p. 241\)](#)
- [EC2 인스턴스의 DNS 호스트 이름 보기 \(p. 241\)](#)
- [VPC에 대한 DNS 지원 업데이트 \(p. 242\)](#)
- [프라이빗 호스팅 영역 사용 \(p. 242\)](#)

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 퍼블릭 IPv4 주소가 있을 경우에 따라 지정한 [DNS 속성 \(p. 239\)](#)에 따라 퍼블릭 DNS 호스트 이름을 제공할 수 있습니다.

Amazon 제공의 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**.compute.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.

속성	설명
<code>enableDnsHostnames</code>	VPC에서 시작된 인스턴스가 퍼블릭 DNS 호스트 이름을 가져오는지 나타냅니다. 이 속성이 <code>true</code> 이고 VPC에서 인스턴스가 퍼블릭 DNS 호스트 이름을 얻은 경우, <code>enableDnsSupport</code> 속성이 또한 <code>true</code> 로 설정된 경우에만 해당됩니다.
<code>enableDnsSupport</code>	VPC에 대해 DNS 확인이 지원되는지 나타냅니다. 이 속성이 <code>false</code> 이면 퍼블릭 DNS 호스트 이름을 IP 주소로 확인하는 VPC의 Amazon 제공의 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 Amazon DNS 서버 (p. 236) 단원을 참조하십시오.

모든 속성이 `true`로 설정되면, 다음 이 발생합니다.

- 인스턴스가 퍼블릭 DNS 호스트 이름을 수신합니다.
- Amazon 제공의 DNS 서버는 Amazon 제공의 프라이빗 DNS 호스트 이름을 확인할 수 있습니다.

각 속성 또는 모든 속성이 `false`로 설정되면, 다음 이 발생합니다.

- 인스턴스가 Amazon EC2 콘솔에서 볼 수 있거나 명령줄 도구 또는 AWS SDK로 설명할 수 있는 퍼블릭 DNS 호스트 이름을 수신하지 않습니다.
- Amazon 제공의 DNS 서버는 Amazon 제공의 프라이빗 DNS 호스트 이름을 확인할 수 없습니다.
- [DHCP 옵션 세트 \(p. 234\)](#)에서 사용자 지정 도메인 이름을 지정하는 경우에는 인스턴스가 사용자 지정 프라이빗 DNS 호스트 이름을 수신합니다. Amazon 제공 DNS 서버를 사용하지 않는 경우에는 사용자 지정 도메인 이름 서버가 적절하게 호스트 이름을 확인해야 합니다.

기본적으로, 모든 속성은 기본 VPC 또는 VPC 마법사에서 생성된 VPC의 `true`로 설정됩니다. 기본적으로, 오직 `enableDnsSupport` 속성만 VPC에서 `true`로 설정되며 이는 VPC 콘솔 또는 AWS CLI, API, 또는 AWS SDK를 사용하는 Your VPCs 페이지에서 생성된 것입니다.

Important

Amazon Route 53의 프라이빗 호스팅 영역에 정의된 사용자 지정 DNS 도메인 이름을 사용하는 경우, `enableDnsHostnames` 및 `enableDnsSupport` 속성을 `true`로 설정해야 합니다.

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.

DNS 제한

각 Amazon EC2 인스턴스는 Amazon에서 제공하는 DNS 서버로 전송할 수 있는 패킷 수를 네트워크 인터페이스당 초당 최대 1024 패킷으로 제한합니다. 이 한도는 늘릴 수 없습니다. Amazon에서 제공하는 DNS 서버가 지원하는 초당 DNS 쿼리 수는 쿼리 유형, 응답 크기 및 사용 중인 프로토콜에 따라 다릅니다. 확장 가능한 DNS 아키텍처에 대한 자세한 내용과 권장 사항은 [Hybrid Cloud DNS Solutions for Amazon VPC](#) 백서를 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 인스턴스를 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. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 7\)](#) 항목을 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 Interfaces]를 선택합니다.
3. 목록에서 네트워크 인터페이스를 선택합니다.
4. 세부 정보 창의 프라이빗 DNS(IPv4) 필드에 프라이빗 DNS 호스트 이름이 표시됩니다.

명령줄을 사용하여 네트워크 인터페이스의 DNS 호스트 이름을 보려면

다음 명령 중 하나를 사용할 수 있습니다. 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 7\)](#) 단원을 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 VPCs]를 선택합니다.
3. 목록에서 VPC를 선택합니다.
4. [Summary] 탭의 정보를 검토합니다. 이 예제에서는 두 가지 모두 사용합니다.

DNS resolution: yes

DNS hostnames: yes

5. 설정을 업데이트하려면 Actions를 선택하고, Edit DNS Resolution 또는 Edit DNS Hostnames를 선택합니다. 이때 열리는 대화 상자에서 [Yes] 또는 [No]를 선택한 후 [Save]를 선택합니다.

명령줄을 사용하여 VPC에 대한 DNS 지원을 설명하려면

다음 명령 중 하나를 사용할 수 있습니다. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 7\)](#) 항목을 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. 다음 명령줄 인터페이스에 대한 자세한 내용은 [Amazon VPC에 액세스 \(p. 7\)](#) 항목을 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.com과 같은 사용자 지정 DNS 도메인 이름을 사용하여 VPC의 리소스에 액세스하려면 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. Route 53이 도메인과 하위 도메인에 대한 쿼리에 응답하는 방식을 결정하는 Route 53 리소스 레코드 세트를 생성할 수 있습니다. 예를 들어 example.com에 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 ping에 대한 인바운드 ICMP 트래픽 작동을 허용하는지 확인해야 합니다.)

ClassicLink DNS 지원에 대해 VPC가 활성화되어 있는 경우 ClassicLink를 사용하여 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` 속성을 `true`로 설정해야 합니다.

VPC 피어링

VPC 피어링 연결은 비공개적으로 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워크 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다. 자체 VPC 간, 다른 AWS 계정에서 VPC를 사용하여 또는 다른 AWS 리전에서 VPC를 사용하여 VPC 피어링 연결을 만들 수 있습니다.

AWS는 VPC의 기존 인프라를 사용하여 VPC 피어링 연결을 생성합니다. 이는 게이트웨이도 아니고 연결도 아니며 각각의 물리적 하드웨어를 사용하지 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.

내용

- [탄력적 IP 주소 기본 사항 \(p. 243\)](#)
- [탄력적 IP 주소 작업 \(p. 244\)](#)
- [API 및 CLI 개요 \(p. 246\)](#)

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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [EC2-Classic 및 Amazon EC2-VPC의 엘라스틱 IP 주소 차이점](#)을 참조하십시오.
- 인스턴스 간에 엘라스틱 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 정보는 [Amazon EC2 요금](#)을 참조하십시오.
- 탄력적 IP 주소는 5개로 제한되며, 이를 절약하기 위해 NAT 디바이스를 사용할 수 있습니다 ([NAT \(p. 210\)](#) 참조).
- 탄력적 IP 주소는 VPC의 인터넷 게이트웨이를 통해 액세스합니다. VPC와 네트워크 간에 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 정보는 Amazon EC2 사용 설명서의 [EC2-Classic에서 EC2-VPC로 엘라스틱 IP 주소 마이그레이션](#)을 참조하십시오.
- VPC에 사용하도록 할당된 탄력적 IP 주소를 태그할 수 있지만, 비용 할당 태그가 지원되지 않습니다. 탄력적 IP 주소를 복구하는 경우, 태그가 복구되지 않습니다.

탄력적 IP 주소 작업

탄력적 IP 주소를 할당한 후 VPC의 인스턴스와 연결할 수 있습니다.

VPC에서 사용할 엘라스틱 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 엘라스틱 IP 주소를 몇 자 입력합니다.

VPC에서 실행 중인 인스턴스와 엘라스틱 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 정보는 [Elastic Network Interfaces\(탄력적 네트워크 인터페이스 \[ENI\]\)](#)를 참조하십시오.

탄력적 IP 주소를 인스턴스와 연결하면 DNS 호스트 이름이 활성화된 경우 DNS 호스트 이름을 받습니다. 자세한 정보는 [VPC와 함께 DNS 사용 \(p. 239\)](#) 단원을 참조하십시오.

탄력적 IP 주소에 태그를 적용하면 조직의 요구에 따라 이를 식별 또는 분류할 수 있습니다.

탄력적 IP 주소를 태그하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Elastic IPs]를 선택합니다.
3. 탄력적 IP 주소를 선택하고 [Tags]를 선택합니다.
4. [Add/Edit Tags]를 선택하고, 필요에 따라 태그 키 및 값을 입력하고, [Save]를 선택합니다.

탄력적 IP 주소와 연결된 인스턴스를 변경하려면 현재 연결된 인스턴스에서 연결을 해제한 후 VPC의 새 인스턴스와 연결하십시오.

엘라스틱 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 엘라스틱 IP 주소에 대한 요금은 발생하지만 인스턴스와 연결된 주소에 대해서는 요금이 발생하지 않습니다.

엘라스틱 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.

탄력적 IP 주소를 해제한 경우 복구할 수 있습니다. 탄력적 IP 주소가 다른 AWS 계정에 할당되었거나, 탄력적 IP 주소 한도를 초과하는 경우에는 탄력적 IP 주소를 복구할 수 없습니다.

현재 Amazon EC2 API 또는 명령줄 도구만을 사용하여 탄력적 IP 주소를 복구할 수 있습니다.

AWS CLI를 사용하여 탄력적 IP 주소를 복구하려면

- `allocate-address` 명령을 사용하고, `--address` 파라미터를 사용하여 IP를 지정합니다.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

API 및 CLI 개요

명령줄 또는 API를 사용하여 이 페이지에서 설명하는 작업을 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 목록에 대한 자세한 정보는 [Amazon VPC에 액세스 \(p. 7\)](#) 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 태그

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (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 엔드포인트를 통해 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 필요로 하지 않고 PrivateLink 구동 지원 AWS 서비스 및 VPC 엔드포인트 서비스에 비공개로 연결할 수 있습니다. VPC의 인스턴스는 서비스의 리소스와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다. VPC와 기타 서비스 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

엔드포인트는 가상 디바이스입니다. 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로, 가용성 위험이나 네트워크 트래픽에 대한 대역폭 제약 없이 VPC의 인스턴스와 서비스 간에 통신할 수 있도록 합니다.

VPC 엔드포인트에는 두 가지 종류가 있습니다. 인터페이스 엔드포인트 및 게이트웨이 엔드포인트 제공된 서비스가 요구하는 VPC 엔드포인트를 만드십시오.

인터페이스 엔드포인트(AWS PrivateLink 구동)

[인터페이스 엔드포인트 \(p. 247\)](#) 는 지원 서비스로 전달되는 트래픽에 대한 진입점 역할을 하는 프라이빗 IP 주소를 포함하는 탄력적 네트워크 인터페이스입니다. 다음 서비스가 지원됩니다.

- [Amazon API Gateway](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch Events](#)
- [Amazon CloudWatch Logs](#)
- [AWS CodeBuild](#)
- [Amazon EC2 API](#)
- [Elastic Load Balancing API](#)
- [AWS Key Management Service](#)
- [Amazon Kinesis Data Streams](#)
- [Amazon SageMaker 실행 시간](#)
- [AWS Secrets Manager](#)
- [AWS Service Catalog](#)
- [Amazon SNS](#)
- [AWS 시스템 관리자](#)
- 기타 AWS 계정이 호스팅한 [엔드포인트 서비스 \(p. 272\)](#)
- 지원 AWS Marketplace 파트너 서비스

게이트웨이 엔드포인트

[게이트웨이 엔드포인트 \(p. 258\)](#)는 라우팅 테이블의 지정된 경로에 대한 대상인 게이트웨이로, 지원 AWS 서비스로 전달되는 트래픽에 사용됩니다. 다음 AWS 서비스가 지원됩니다.

- Amazon S3
- DynamoDB

VPC 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. 현재는 `ec2:*VpcEndpoint*` API 작업 또는 `ec2:DescribePrefixLists` 작업에 대한 리소스 수준 권한을 지원하지 않습니다. 사용자에게 특정 엔드포인트 또는 접두사 목록을 사용할 수 있는 권한을 부여하는 IAM 정책을 생성할 수 없습니다. 다음은 그 한 예입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:*VpcEndpoint*",
    "Resource": "*"
  }]
}
```

인터페이스 VPC 엔드포인트 (AWS PrivateLink)

인터페이스 VPC 엔드포인트(인터페이스 엔드포인트)를 통해 AWS PrivateLink 구동 서비스에 연결할 수 있습니다. 이러한 서비스에는 일부 AWS 서비스, 다른 AWS 고객 및 파트너가 자체 VPC로 호스팅한 서비스(엔드포인트 서비스라고 함) 및 지원되는 AWS Marketplace 파트너 서비스가 포함됩니다. 서비스의 소유자는 서비스 공급자이고, 인터페이스 엔드포인트를 생성하는 주체는 서비스 소비자입니다.

다음 서비스가 지원됩니다.

- [Amazon API Gateway](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch Events](#)
- [Amazon CloudWatch Logs](#)
- [AWS CodeBuild](#)
- [Amazon EC2 API](#)
- [Elastic Load Balancing API](#)
- [AWS Key Management Service](#)
- [Amazon Kinesis Data Streams](#)
- [Amazon SageMaker 실행 시간](#)
- [AWS Secrets Manager](#)
- [AWS Service Catalog](#)
- [Amazon SNS](#)
- [AWS 시스템 관리자](#)
- 기타 AWS 계정이 호스팅한 [엔드포인트 서비스 \(p. 272\)](#)
- [지원 AWS Marketplace 파트너 서비스](#)

다음은 인터페이스 엔드포인트 설정의 일반적인 단계입니다.

1. 인터페이스 엔드포인트를 생성할 VPC를 선택하고, 연결하는 AWS 서비스, 엔드포인트 서비스 또는 AWS Marketplace 서비스의 이름을 입력합니다.
2. 인터페이스 엔드포인트를 사용할 VPC의 서브넷을 선택합니다. 서브넷에서 엔드포인트 네트워크 인터페이스가 생성됩니다. 각기 다른 가용 영역(서비스에 의해 지원)에서 하나 이상의 서브넷을 지정하여 인터페이스 엔드포인트가 가용 영역 장애 발생 시 복원을 지원합니다. 이러한 경우 사용자가 지정한 각 서브넷에서 엔드포인트 네트워크 인터페이스가 생성됩니다.

Note

엔드포인트 네트워크 인터페이스는 요청자 관리형 네트워크 인터페이스입니다. 계정에서 볼 수는 있지만 직접 관리할 수는 없습니다. 자세한 정보는 [탄력적 네트워크 인터페이스](#)를 참조하십시오.

3. 보안 그룹을 지정하여 엔드포인트 네트워크 인터페이스와 연결합니다. 보안 그룹 규칙은 VPC의 리소스로부터 엔드포인트 네트워크 인터페이스로의 트래픽을 제어합니다. 보안 그룹을 지정하지 않은 경우 VPC에 대한 기본 보안 그룹이 연결됩니다.
4. (AWS 서비스 및 AWS Marketplace 파트너 서비스에서만 옵션임) 엔드포인트에 대한 [프라이빗 DNS \(p. 249\)](#)를 활성화하면 기본 DNS 호스트 이름을 사용하여 서비스에 요청을 생성할 수 있습니다.
5. 인터페이스 엔드포인트를 생성한 이후 서비스 공급자가 이를 수락하면 사용할 수 있습니다. 서비스 공급자는 서비스가 요청을 자동 또는 수동으로 수락하도록 구성해야 합니다. AWS 서비스 및 AWS Marketplace 서비스는 일반적으로 모든 엔드포인트 요청을 자동적으로 수락합니다. 엔드포인트의 수명 주기에 대한 자세한 정보는 [인터페이스 엔드포인트 수명 주기 \(p. 251\)](#) 단원을 참조하십시오.

서비스가 엔드포인트를 통한 VPC의 리소스 요청을 시작할 수 없습니다. 엔드포인트는 VPC의 리소스로부터 시작한 트래픽에 대한 응답만을 반환합니다.

내용

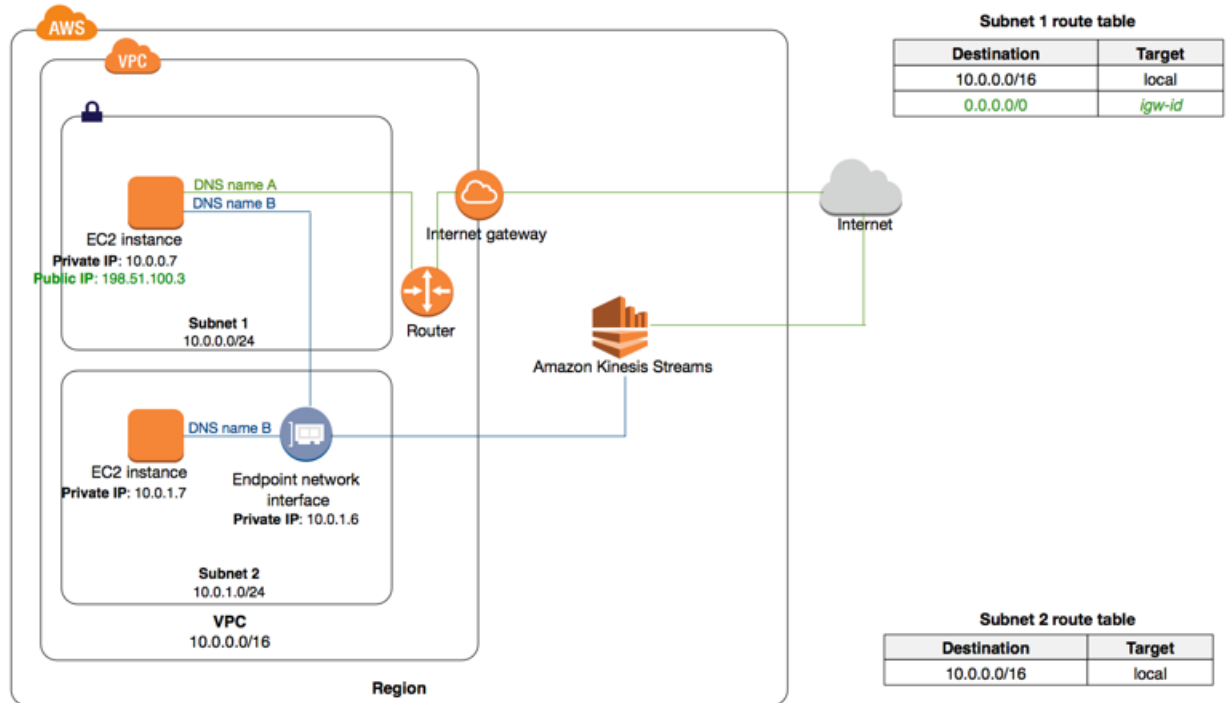
- [프라이빗 DNS \(p. 249\)](#)
- [인터페이스 엔드포인트 속성 및 제한 \(p. 250\)](#)
- [인터페이스 엔드포인트 수명 주기 \(p. 251\)](#)
- [인터페이스 엔드포인트 요금 \(p. 252\)](#)

- 인터페이스 엔드포인트 생성 (p. 252)
- 인터페이스 엔드포인트 보기 (p. 255)
- 인터페이스 엔드포인트에 대한 알림 생성 및 관리 (p. 256)
- 인터페이스 엔드포인트를 통해 서비스 액세스 (p. 257)
- 인터페이스 엔드포인트 수정 (p. 258)

프라이빗 DNS

인터페이스 엔드포인트를 생성할 때 서비스와의 통신에 사용할 수 있는 엔드포인트별 DNS 호스트 이름이 생성됩니다. AWS 서비스 및 AWS Marketplace 파트너 서비스의 경우 엔드포인트에 대한 프라이빗 DNS를 선택적으로 활성화할 수 있습니다. 이 옵션은 VPC와 프라이빗 호스팅 영역을 연결합니다. 호스팅 영역에는 VPC에 있는 엔드포인트 네트워크 인터페이스의 프라이빗 IP 주소를 확인하는 서비스(예: `ec2.us-east-1.amazonaws.com`)에 대한 기본 DNS 이름의 레코드 세트가 포함됩니다. 이를 통해 엔드포인트별 DNS 호스트 이름 대신 기본 DNS 호스트 이름을 사용하여 서비스에 요청을 생성할 수 있습니다. 예를 들어 기존 애플리케이션이 AWS 서비스에 대한 요청을 생성하는 경우 구성을 변경할 필요 없이 인터페이스 엔드포인트를 통해 계속해서 요청을 생성할 수 있습니다.

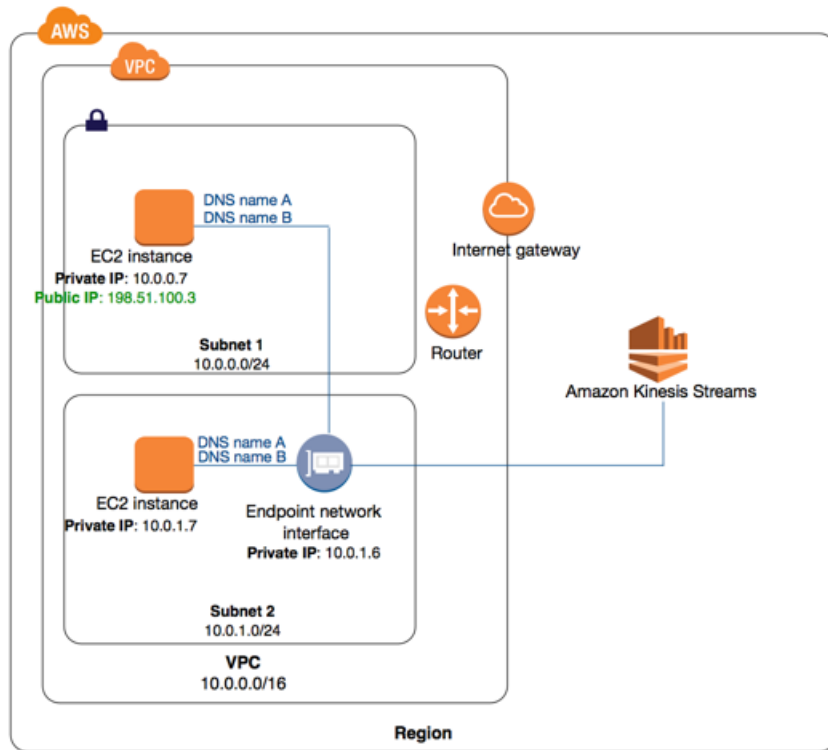
다음 다이어그램에서, 서브넷 2에 Amazon Kinesis Data Streams에 대한 인터페이스 엔드포인트와 엔드포인트 네트워크 인터페이스를 생성했습니다. 인터페이스 엔드포인트에 대한 프라이빗 DNS가 활성화되지 않았습니다. 서브넷의 인스턴스가 엔드포인트별 DNS 호스트 이름(DNS 이름 B)을 사용하여 인터페이스 엔드포인트를 통해 Amazon Kinesis Data Streams와 통신할 수 있습니다. 서브넷 1의 인스턴스는 서비스에 대한 기본 DNS 이름(DNS 이름 A)을 사용하여 AWS 리전에서 퍼블릭 IP 주소 공간을 통해 Amazon Kinesis Data Streams와 통신할 수 있습니다.



DNS name A: `kinesis.us-east-1.amazonaws.com` (default DNS hostname)

DNS name B: `vpce-123-ab.kinesis.us-east-1.vpc.amazonaws.com` (endpoint-specific hostname)

다음 다이어그램에서 엔드포인트에 대한 프라이빗 DNS가 활성화되었습니다. 서브넷의 인스턴스가 엔드포인트별 DNS 호스트 이름(DNS 이름 B) 또는 서비스에 대한 기본 DNS 이름(DNS 이름 A)을 사용하여 인터페이스 엔드포인트를 통해 Amazon Kinesis Data Streams와 통신할 수 있습니다.



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

Destination	Target
10.0.0.0/16	local

DNS name A: kinesis.us-east-1.amazonaws.com (default DNS hostname)

DNS name B: vpce-123-ab.kinesis.us-east-1.vpce.amazonaws.com (endpoint-specific hostname)

Important

프라이빗 DNS를 사용하려면 다음 VPC 속성을 true로 설정해야 합니다. `enableDnsHostnames` 및 `enableDnsSupport`. 자세한 정보는 [VPC에 대한 DNS 지원 업데이트 \(p. 242\)](#) 단원을 참조하십시오. IAM 사용자는 호스팅 영역으로 작업할 권한이 있어야 합니다. 자세한 정보는 [에 대한 인증 및 액세스 제어Route 53](#)를 참조하십시오.

인터페이스 엔드포인트 속성 및 제한

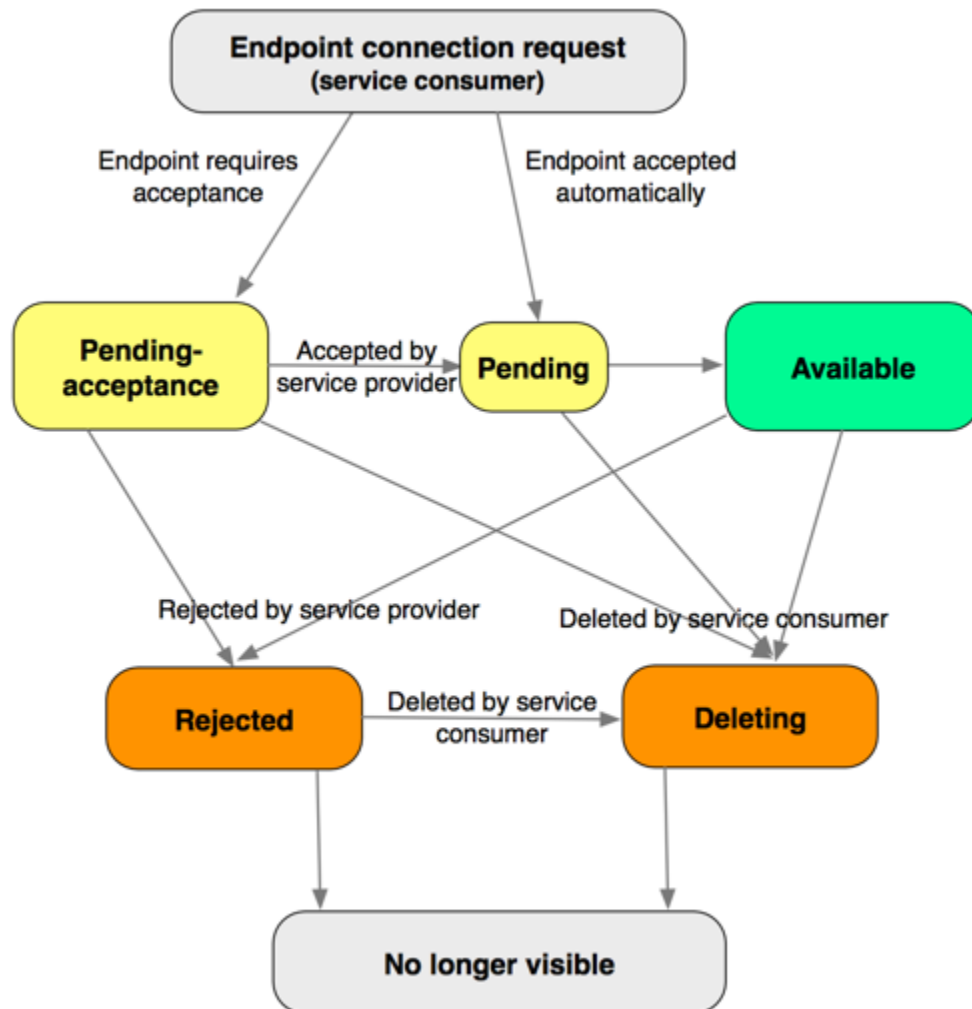
인터페이스 엔드포인트를 사용하려면 속성 및 현재 제한 사항을 알고 있어야 합니다.

- 인터페이스 엔드포인트는 AWS VPN 연결 또는 AWS Direct Connect 연결을 통해 수락될 수 있습니다. 인터페이스 엔드포인트는 [Nitro 인스턴스](#)로부터 리전 내 VPC 피어링 연결을 통해 액세스할 수 있습니다. 인터페이스 엔드포인트는 모든 유형의 인스턴스로부터 리전 내 VPC 피어링 연결을 통해 액세스할 수 있습니다.
- 각 인터페이스 엔드포인트에서 가용 영역당 1개의 서브넷만 선택할 수 있습니다.
- 인터페이스 엔드포인트는 엔드포인트 정책을 지원하지 않습니다. 인터페이스 엔드포인트를 통한 서비스에 대한 모든 액세스가 허용됩니다.
- 인터페이스 엔드포인트를 통해 모든 가용 영역에서 서비스를 사용할 수 없을 수 있습니다. 지원되는 가용 영역을 확인하려면 [describe-vpc-endpoint-services](#) 명령을 사용하거나 Amazon VPC 콘솔을 사용하십시오. 자세한 정보는 [인터페이스 엔드포인트 생성 \(p. 252\)](#) 단원을 참조하십시오.
- 사용자 계정의 가용 영역이 다른 계정의 가용 영역과 동일한 위치로 매핑되지 않을 수 있습니다. 예를 들어 가용 영역인 `us-east-1a`가 다른 계정의 `us-east-1a`와 위치가 다를 수 있습니다. 자세한 정보는 [리전 및 가용 영역 개념](#)을 참조하십시오. 인터페이스 엔드포인트를 생성할 때 계정의 가용 영역에 생성됩니다.
- 각 인터페이스 엔드포인트는 가용 영역당 기본적으로 최대 10Gbps의 대역폭을 지원할 수 있습니다. 추가 용량은 사용량에 따라 자동적으로 추가될 수 있습니다.

- 서브넷에 대한 네트워크 ACL이 트래픽을 제한하는 경우 엔드포인트 네트워크 인터페이스를 통해 트래픽을 보내지 못할 수 있습니다. 서브넷의 CIDR 블록에서 주고 받는 트래픽을 허용하는 적절한 규칙을 추가해야 합니다.
- 인터페이스 엔드포인트는 TCP 트래픽만을 지원합니다.
- 엔드포인트는 동일 리전에서만 지원됩니다. VPC와 다른 리전의 서비스 간에 엔드포인트를 만들 수 없습니다.
- 엔드포인트에 태그를 지정할 수 없습니다.
- 엔드포인트는 IPv4 트래픽만 지원합니다.
- VPC 간에 또는 서비스 간에 엔드포인트를 전송할 수 없습니다.
- VPC당 만들 수 있는 엔드포인트 수는 제한이 있습니다. 자세한 내용은 [VPC 엔드포인트 \(p. 288\)](#) 단원을 참조하십시오.

인터페이스 엔드포인트 수명 주기

인터페이스 엔드포인트는 생성 시(엔드포인트 연결 요청)부터 다양한 단계를 거칩니다. 각 단계에서 서비스 소비자和服务 공급자가 수행할 수 있는 작업이 있을 수 있습니다.



다음 규칙이 적용됩니다.

- 서비스 공급자는 서비스가 인터페이스 엔드포인트 요청을 자동 또는 수동으로 수락하도록 구성할 수 있습니다. AWS 서비스 및 AWS Marketplace 서비스는 일반적으로 모든 엔드포인트 요청을 자동적으로 수락합니다.
- 서비스 공급자는 서비스에 대한 인터페이스 엔드포인트를 삭제할 수 없습니다. 인터페이스 엔드포인트 연결을 요청한 서비스 소비자만이 인터페이스 엔드포인트를 삭제할 수 있습니다.
- 서비스 공급자는 인터페이스 엔드포인트를 수락(수동 또는 자동으로)한 이후에도 이를 거부할 수 있으며, available 상태가 됩니다.

인터페이스 엔드포인트 요금

서비스에 대한 인터페이스 엔드포인트 생성 및 사용에 대한 요금이 청구됩니다. 시간당 사용 요금 및 데이터 처리 요금이 적용됩니다. 자세한 정보는 [Amazon VPC 요금](#)을 참조하십시오.

인터페이스 엔드포인트 생성

인터페이스 엔드포인트를 생성하려면 인터페이스 엔드포인트를 생성하려는 VPC 및 연결을 설정할 서비스를 지정해야 합니다.

AWS 서비스 및 AWS Marketplace 파트너 서비스에 연결하는 경우 인터페이스 엔드포인트에 대한 [프라이빗 DNS \(p. 249\)](#)를 활성화할 수 있습니다. 이를 통해 인터페이스 엔드포인트를 생성할 때 생성된 엔드포인트 별 DNS 호스트 이름 대신 기본 DNS 이름을 사용하여 서비스에 요청을 생성할 수 있습니다.

AWS 서비스에 대한 특정 정보는 [VPC 엔드포인트 \(p. 246\)](#) 단원을 참조하십시오.

콘솔을 사용하여 AWS 서비스에 대한 인터페이스 엔드포인트를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]와 [Create Endpoint]를 선택합니다.
3. [Service category]에서 [AWS services]를 선택해야 합니다.
4. [Service Name]에서 연결할 서비스를 선택합니다. [Type]에서 [Interface]를 나타내는 지 확인합니다.
5. 다음 정보를 입력한 다음 [Create endpoint]를 선택합니다.

- [VPC]에서 엔드포인트를 생성할 VPC를 선택합니다.
- [Subnets]에서 엔드포인트 네트워크 인터페이스를 생성할 서브넷(가용 영역)을 선택합니다.

Note

일부 AWS 서비스에서 지원되지 않는 가용 영역도 있습니다.

- [Enable Private DNS Name]에서 인터페이스 엔드포인트에 대한 프라이빗 DNS를 활성화하는 확인란을 선택할 수 있습니다.

Note

프라이빗 DNS 옵션 사용하려면 다음 VPC 속성을 true로 설정해야 합니다.

`enableDnsHostnames` 및 `enableDnsSupport`. 자세한 정보는 [VPC에 대한 DNS 지원 업데이트 \(p. 242\)](#) 단원을 참조하십시오.

- [Security group]에서 보안 그룹을 선택하여 엔드포인트 네트워크 인터페이스와 연결합니다.

엔드포인트 서비스에 대한 인터페이스 엔드포인트를 생성하려면 연결할 서비스의 이름을 보유해야 합니다. 서비스 공급자가 이름을 제공할 수 있습니다.

엔드포인트 서비스에 대한 인터페이스 엔드포인트를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]와 [Create Endpoint]를 선택합니다.

3. [Service category]에서 [Find service by name]을 선택합니다.
4. [Service Account Name]에서 서비스의 이름(예: com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc)을 입력한 다음 [Verify]를 선택합니다.
5. 다음 정보를 입력한 다음 [Create endpoint]를 선택합니다.

- [VPC]에서 엔드포인트를 생성할 VPC를 선택합니다.
- [Subnets]에서 엔드포인트 네트워크 인터페이스를 생성할 서브넷(가용 영역)을 선택합니다.

Note

모든 가용 영역이 서비스를 지원할 수 있는 것은 아닙니다.

- [Security group]에서 보안 그룹을 선택하여 엔드포인트 네트워크 인터페이스와 연결합니다.

AWS Marketplace 파트너 서비스에 대한 인터페이스 엔드포인트를 생성하려면

1. AWS Marketplace의 [PrivateLink](#) 페이지로 이동한 다음 SaaS(Software as a Service) 공급자로부터 서비스를 구독합니다. 인터페이스 엔드포인트를 지원하는 서비스는 엔드포인트를 통한 연결 옵션을 포함합니다.
2. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
3. 탐색 창에서 [Endpoints]와 [Create Endpoint]를 선택합니다.
4. [Service category]에서 [Your AWS Marketplace services]를 선택합니다.
5. 구독한 AWS Marketplace 서비스를 선택합니다.
6. 다음 정보를 입력한 다음 [Create endpoint]를 선택합니다.

- [VPC]에서 엔드포인트를 생성할 VPC를 선택합니다.
- [Subnets]에서 엔드포인트 네트워크 인터페이스를 생성할 서브넷(가용 영역)을 선택합니다.

Note

모든 가용 영역이 서비스를 지원할 수 있는 것은 아닙니다.

- [Security group]에서 보안 그룹을 선택하여 엔드포인트 네트워크 인터페이스와 연결합니다.

AWS CLI를 사용하여 인터페이스 엔드포인트를 생성하려면

1. `describe-vpc-endpoint-services` 명령을 사용하여 사용 가능한 목록을 가져옵니다. 반환된 출력에 표시된 연결할 서비스의 이름을 메모해 둡니다. `ServiceType` 필드는 인터페이스 또는 게이트웨이 엔드포인트를 통해 서비스에 연결할지 여부를 나타냅니다. `ServiceName` 필드는 서비스의 이름을 제공합니다.

```
aws ec2 describe-vpc-endpoint-services
```

```
{
  "ServiceDetails": [
    ...
    {
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "PrivateDnsName": "elasticloadbalancing.us-east-1.amazonaws.com",
      "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
      "VpcEndpointPolicySupported": false,
      "Owner": "amazon",
      "AvailabilityZones": [
        "us-east-1a",

```

```

        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ],
    "AcceptanceRequired": false,
    "BaseEndpointDnsNames": [
        "elasticloadbalancing.us-east-1.vpce.amazonaws.com"
    ]
},
...
}

```

2. 인터페이스 엔드포인트를 생성하려면 `create-vpc-endpoint` 명령을 사용하고 VPC ID, VPC 엔드포인트의 유형(인터페이스), 서비스 이름, 엔드포인트를 사용할 서브넷 및 엔드포인트 네트워크 인터페이스와 연결할 보안 그룹을 지정합니다.

다음 예제는 Elastic Load Balancing 서비스에 대한 인터페이스 엔드포인트를 생성합니다.

```

aws ec2 create-vpc-endpoint --vpc-id vpc-ec43eb89 --vpc-endpoint-type Interface --
service-name com.amazonaws.us-east-1.elasticloadbalancing --subnet-id subnet-abababab
--security-group-id sg-1a2b3c4d

```

```

{
  "VpcEndpoint": {
    "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*\", \n\n\n    }\n  ]\n}\",\n    \"VpcId\": \"vpc-ec43eb89\",
    \"NetworkInterfaceIds\": [
      \"eni-bf8aa46b\"
    ],
    \"SubnetIds\": [
      \"subnet-abababab\"
    ],
    \"PrivateDnsEnabled\": true,
    \"State\": \"pending\",
    \"ServiceName\": \"com.amazonaws.us-east-1.elasticloadbalancing\",
    \"RouteTableIds\": [],
    \"Groups\": [
      {
        \"GroupName\": \"default\",
        \"GroupId\": \"sg-1a2b3c4d\"
      }
    ],
    \"VpcEndpointId\": \"vpce-088d25a4bbf4a7abc\",
    \"VpcEndpointType\": \"Interface\",
    \"CreationTimestamp\": \"2017-09-05T20:14:41.240Z\",
    \"DnsEntries\": [
      {
        \"HostedZoneId\": \"Z7HUB22UULQXV\",
        \"DnsName\": \"vpce-088d25a4bbf4a7abc-ks83awe7.elasticloadbalancing.us-
east-1.vpce.amazonaws.com\"
      },
      {
        \"HostedZoneId\": \"Z7HUB22UULQXV\",
        \"DnsName\": \"vpce-088d25a4bbf4a7abc-ks83awe7-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com\"
      },
      {
        \"HostedZoneId\": \"Z1K56Z6FNPJRR\",
        \"DnsName\": \"elasticloadbalancing.us-east-1.amazonaws.com\"
      }
    ]
  }
}

```

```
}  
  ]  
}  
}
```

또는 다음 예제는 다른 AWS 계정의 엔드포인트 서비스(서비스 공급자가 제공한 엔드포인트 서비스의 이름)에 대한 인터페이스 엔드포인트를 생성합니다.

```
aws ec2 create-vpc-endpoint --vpc-id vpc-ec43eb89 --vpc-endpoint-type Interface  
--service-name com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc --subnet-  
id subnet-abababab --security-group-id sg-1a2b3c4d
```

반환된 출력에 표시된 DnsName 필드를 메모해 둡니다. 이 DNS 이름을 사용하여 AWS 서비스에 액세스할 수 있습니다.

Windows PowerShell용 AWS 도구 또는 API를 사용하여 사용 가능한 서비스를 설명하려면

- [Get-EC2VpcEndpointService](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpointServices](#)(Amazon EC2 Query API)

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 만들려면

- [New-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [CreateVpcEndpoint](#)(Amazon EC2 Query API)

인터페이스 엔드포인트 보기

인터페이스 엔드포인트를 생성한 후 그에 관한 정보를 볼 수 있습니다.

콘솔을 사용하여 인터페이스 엔드포인트에 대한 정보를 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.
3. 인터페이스 엔드포인트에 대한 정보를 보려면 [Details]를 선택합니다. [DNS Names] 필드는 서비스 액세스에 사용할 DNS 이름을 표시합니다.
4. 인터페이스 엔드포인트가 생성된 서브넷과 각 서브넷의 엔드포인트 네트워크 인터페이스 ID를 보려면 [Subnets]를 선택합니다.
5. 엔드포인트 네트워크 인터페이스와 연결된 보안 그룹을 보려면 [Security Groups]를 선택합니다.

AWS CLI를 사용하여 인터페이스 엔드포인트를 설명하려면

- [describe-vpc-endpoints](#) 명령을 사용하여 엔드포인트를 설명할 수 있습니다.

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-088d25a4bbf4a7abc
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 설명하려면

- [Get-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpoints](#)(Amazon EC2 Query API)

인터페이스 엔드포인트에 대한 알림 생성 및 관리

인터페이스 엔드포인트에서 발생하는 특정 이벤트에 대한 알림을 받도록 알림을 생성할 수 있습니다. 예를 들어 서비스 공급자가 인터페이스 엔드포인트를 수락할 때 이메일을 받을 수 있습니다. 알림을 생성하려면 [Amazon SNS 주제](#)를 알림에 연결해야 합니다. SNS 주제를 구독하여 엔드포인트 이벤트가 발생할 때 이메일 알림을 받을 수 있습니다.

알림에 대해 사용할 Amazon SNS 주제는 Amazon의 VPC 엔드포인트 서비스가 사용자를 대신해 알림을 게시하도록 허용하는 주제 정책을 보유해야 합니다. 주제 정책에 다음 문이 포함되어야 합니다. 자세한 정보는 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 주제에 대한 액세스 관리](#) 단원을 참조하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account:topic-name"
    }
  ]
}
```

인터페이스 엔드포인트에 대한 알림을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.
3. [Actions], [Create notification]을 선택합니다.
4. SNS 주제가 알림과 연결할 ARN을 선택합니다.
5. [Events]에서 알림을 수신할 대상이 되는 엔드포인트 이벤트를 선택합니다.
6. [Create Notification]을 선택합니다.

알림을 생성한 후 알림에 연결된 SNS 주제를 변경하거나 알림에 대한 다른 엔드포인트 이벤트를 지정할 수 있습니다.

엔드포인트 서비스에 대한 알림을 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.
3. [Actions], [Modify Notification]을 선택합니다.
4. SNS 주제에 대한 ARN을 지정하고 필요한 경우 엔드포인트 이벤트를 변경합니다.
5. [Modify Notification]을 선택합니다.

알림이 더 이상 필요하지 않으면 삭제할 수 있습니다.

알림을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.

3. [Actions], [Delete notification]을 선택합니다.
4. 예, 삭제를 선택합니다.

AWS CLI를 사용하여 알림을 생성 및 관리하려면

1. 인터페이스 엔드포인트에 대한 알림을 생성하려면 `create-vpc-endpoint-connection-notification` 명령을 사용하고 SNS 주제의 ARN, 알림을 받을 이벤트, 엔드포인트 서비스 ID를 지정합니다. 예를 들면 다음과 같습니다.

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-arn arn:aws:sns:us-east-2:123456789012:EndpointNotification --connection-events Accept Reject --vpc-endpoint-id vpce-123abc3420c1931d7
```

2. 알림을 보려면 `describe-vpc-endpoint-connection-notifications` 명령을 사용합니다.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

3. 알림에 대한 SNS 주제 및 엔드포인트 이벤트를 변경하려면 `modify-vpc-endpoint-connection-notification` 명령을 사용합니다. 예를 들면 다음과 같습니다.

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-id vpce-nfn-008776de7e03f5abc --connection-events Accept --connection-notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

4. 알림을 삭제하려면 `delete-vpc-endpoint-connection-notifications` 명령을 사용합니다.

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

인터페이스 엔드포인트를 통해 서비스 액세스

인터페이스 엔드포인트를 생성한 후 엔드포인트 URL을 통해 지원 서비스에 대한 요청을 제출할 수 있습니다. 다음을 사용할 수 있습니다.

- 인터페이스 엔드포인트에 대해 생성한 엔드포인트별 리전 DNS 호스트 이름입니다. 호스트 이름에는 고유한 엔드포인트 식별자, 서비스 식별자, 리전 및 이름에 `vpce.amazonaws.com`을 포함합니다(예: `vpce-0fe5b17a0707d6abc-29p5708s.ec2.us-east-1.vpce.amazonaws.com`).
- 엔드포인트를 사용할 수 있는 각 가용 영역에 대해 생성한 엔드포인트별 영역 DNS 호스트 이름입니다. 호스트 이름에는 이름의 가용 영역을 포함합니다(예: `vpce-0fe5b17a0707d6abc-29p5708s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com`). 아키텍처가 가용 영역을 격리하는 경우(예를 들어 결함 제한이나 리전 데이터 전송 비용 절감 등을 위해) 이 옵션을 사용할 수 있습니다.

Note

영역 단위 DNS 호스트 이름에 대한 요청이 서비스 공급자 계정의 해당 가용 영역 위치로 전달되는데, 사용자 계정과 가용 영역의 이름이 다를 수도 있습니다. 자세한 정보는 [리전 및 가용 영역 개념](#)을 참조하십시오.

- 엔드포인트(프라이빗 호스팅 영역)에 대한 프라이빗 DNS를 활성화한 경우 리전에 대한 AWS 서비스의 기본 DNS 호스트 이름입니다(예: `ec2.us-east-1.amazonaws.com`).
- VPC에 있는 엔드포인트 네트워크 인터페이스의 프라이빗 IP 주소입니다.

예를 들어 Elastic Load Balancing에 대한 인터페이스 엔드포인트를 보유하고 프라이빗 DNS 옵션을 활성화하지 않은 서브넷에서 인스턴스로부터 다음 AWS CLI 명령을 사용하여 로드 밸런서를 설명합니다. 이 명령은 엔드포인트별 리전 DNS 호스트 이름을 사용하여 인터페이스 엔드포인트를 통해 요청을 생성합니다.


```
aws elbv2 describe-load-balancers --endpoint-url https://vpce-0f89a33420c193abc-bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com/
```

프라이빗 DNS 옵션을 활성화한 경우 요청의 엔드포인트 URL을 지정할 옵션이 없습니다. AWS CLI는 리전의 서비스에 대한 기본 엔드포인트를 사용합니다(elasticloadbalancing.us-east-1.amazonaws.com).

인터페이스 엔드포인트 수정

인터페이스 엔드포인트가 있는 서브넷을 변경하고 엔드포인트 네트워크 인터페이스와 연결된 보안 그룹을 변경하여 인터페이스 엔드포인트를 수정할 수 있습니다. 인터페이스 엔드포인트에 대한 서브넷을 제거한 경우 서브넷의 해당 엔드포인트 네트워크 인터페이스가 삭제됩니다.

인터페이스 엔드포인트에 대한 서브넷을 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.
3. [Actions], [Manage Subnets]를 선택합니다.
4. 필요에 따라 서브넷을 선택하거나 선택 취소한 후 [Modify Subnets]를 선택합니다.

인터페이스 엔드포인트와 연결된 보안 그룹을 추가 또는 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 인터페이스 엔드포인트를 선택합니다.
3. [Actions], [Manage security groups]를 선택합니다.
4. 필요에 따라 보안 그룹을 선택하거나 선택 취소한 후 [Save]를 선택합니다.

AWS CLI를 사용하여 VPC 엔드포인트를 수정하려면

1. `describe-vpc-endpoints` 명령을 사용하여 인터페이스 엔드포인트의 ID를 가져옵니다.

```
aws ec2 describe-vpc-endpoints
```

2. 다음 예제는 `modify-vpc-endpoint` 명령을 사용하여 서브넷 subnet-aabb1122를 인터페이스 엔드포인트에 추가합니다.

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6abc --add-subnet-id subnet-aabb1122
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 수정하려면

- [Edit-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [ModifyVpcEndpoint](#)(Amazon EC2 Query API)

게이트웨이 VPC 엔드포인트

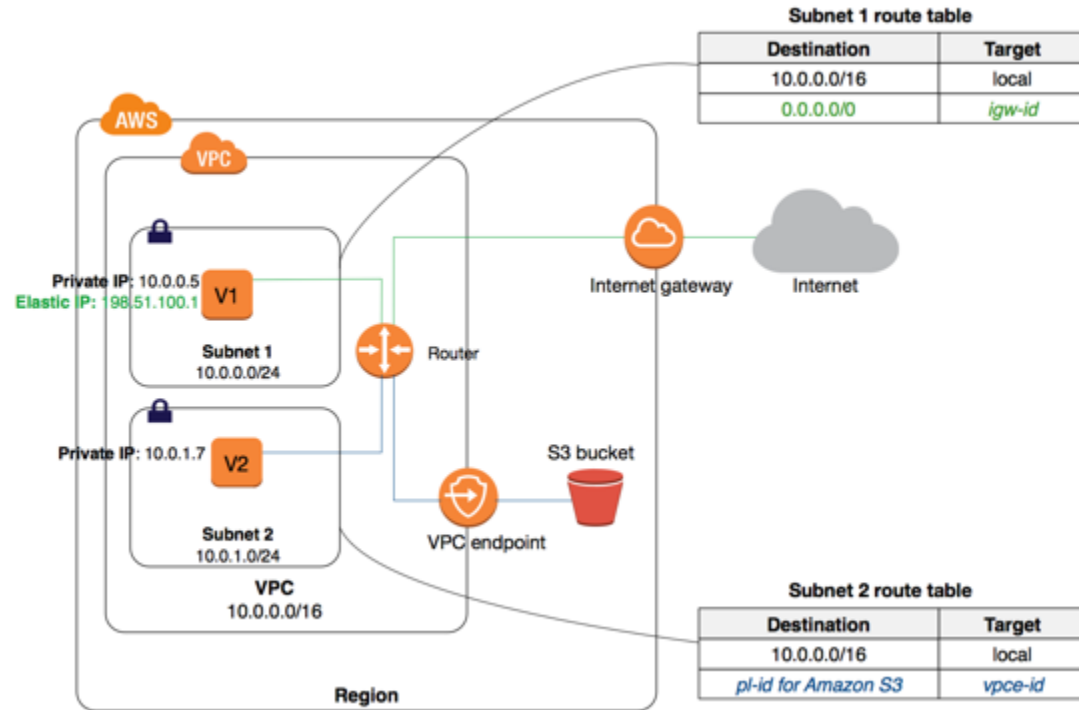
게이트웨이 엔드포인트를 생성 및 설정하려면 다음의 일반 단계를 따릅니다.

1. 엔드포인트를 만들 VPC와 여기에 연결하려는 서비스를 지정합니다. 서비스는 접두사 목록—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.

2. 연결하려는 모든 서비스 또는 일부 서비스에 액세스를 허용하는 엔드포인트 정책을 엔드포인트에 추가합니다. 자세한 정보는 [VPC 엔드포인트 정책 사용 \(p. 271\)](#) 단원을 참조하십시오.
3. VPC와 다른 서비스 간의 트래픽 라우팅을 제어하려면 하나 이상의 라우팅 테이블을 지정합니다. 이러한 라우팅 테이블을 사용하는 서브넷은 엔드포인트에 액세스할 수 있으며, 해당 서브넷의 인스턴스에서 서비스로 전송되는 트래픽은 엔드포인트를 통해 라우팅됩니다.

다음 다이어그램에서 서브넷 2의 인스턴스는 게이트웨이 엔드포인트를 통해 Amazon S3에 액세스할 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.

엔드포인트를 만든 후 엔드포인트에 추가한 엔드포인트 정책을 수정할 수 있고, 엔드포인트에서 사용하는 라우팅 테이블을 추가하거나 제거할 수 있습니다.

게이트웨이 엔드포인트 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 리소스 사용량에 대한 표준 요금이 그대로 적용됩니다. 요금에 대한 자세한 정보는 [Amazon EC2 요금](#)을 참조하십시오.

내용

- [게이트웨이 엔드포인트의 라우팅 \(p. 260\)](#)
- [게이트웨이 엔드포인트 제한 \(p. 261\)](#)
- [Amazon S3에 대한 엔드포인트 \(p. 262\)](#)
- [Amazon DynamoDB에 대한 엔드포인트 \(p. 266\)](#)
- [게이트웨이 엔드포인트 생성 \(p. 268\)](#)
- [보안 그룹 수정 \(p. 270\)](#)
- [게이트웨이 엔드포인트 수정 \(p. 270\)](#)

게이트웨이 엔드포인트의 라우팅

엔드포인트를 만들거나 수정할 경우 엔드포인트를 통해 서비스에 액세스할 때 사용되는 VPC 라우팅 테이블을 지정합니다. 경로는 각각의 라우팅 테이블에 자동으로 추가되며 이때 서비스의 접두사 목록 ID(`p1-xxxxxxxx`)를 지정하는 대상 주소 및 엔드포인트 ID(`vpce-xxxxxxxx`)를 포함한 대상도 함께 추가됩니다. 예를 들면 다음과 같습니다.

대상 주소	대상
10.0.0.0/16	로컬
p1-1a2b3c4d	vpce-11bb22cc

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 해당 엔드포인트를 사용하지 않습니다. 따라서 다른 서브넷의 리소스를 엔드포인트와 분리할 수 있습니다.

서비스의 현재 퍼블릭 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.

다음 규칙이 적용됩니다.

- 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 Amazon VPC 콘솔의 라우팅 테이블 페이지를 사용하여 라우팅 테이블에서 엔드포인트 라우팅을 명시적으로 추가, 수정 또는 삭제할 수 없습니다. 라우팅 테이블을 엔드포인트와 연결하는 방법뿐만 아니라 엔드포인트 라우팅을 추가할 수 있습니다. 엔드포인트와 연결된 라우팅 테이블을 변경하려면 [엔드포인트를 수정](#) (p. 270)하면 됩니다.
- 엔드포인트를 수정하여 엔드포인트에서 라우팅 테이블 연결을 제거하거나 엔드포인트를 삭제할 경우 엔드포인트 라우팅이 자동으로 삭제됩니다.

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.

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.

대상 주소	대상
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 대상 주소 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.

대상 주소	대상
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. 대상 주소가 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.

대상 주소	대상
10.0.0.0/16	로컬
54.123.165.0/24	igw-1a2b3c4d
p1-1a2b3c4d	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.

대상 주소	대상
10.0.0.0/16	로컬
p1-1a2b3c4d	vpce-11bb22cc

게이트웨이 엔드포인트 제한

게이트웨이 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 정보는 [보안 그룹 \(p. 272\)](#) 단원을 참조하십시오.
- 엔드포인트는 동일 리전에서만 지원됩니다. VPC와 다른 리전의 서비스 간에 엔드포인트를 만들 수 없습니다.
- 엔드포인트에 태그를 지정할 수 없습니다.
- 엔드포인트는 IPv4 트래픽만 지원합니다.
- VPC 간에 또는 서비스 간에 엔드포인트를 전송할 수 없습니다.
- VPC당 만들 수 있는 엔드포인트 수는 제한이 있습니다. 자세한 내용은 [VPC 엔드포인트 \(p. 288\)](#) 단원을 참조하십시오.
- 엔드포인트 연결은 VPC 외부로 확장할 수 없습니다. VPN 연결, VPC 피어링 연결, AWS Direct Connect 연결 또는 VPC의 ClassicLink 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 정보는 Amazon Web Services 일반 참조의 [VPC와 함께 DNS 사용 \(p. 239\)](#) 및 [AWS IP 주소 범위를 참조하십시오](#).

Amazon S3와 관련된 규칙 및 제한에 대한 자세한 정보는 [Amazon S3에 대한 엔드포인트 \(p. 262\)](#) 단원을 참조하십시오.

DynamoDB와 관련된 규칙 및 제한에 대한 자세한 정보는 [Amazon DynamoDB에 대한 엔드포인트 \(p. 266\)](#) 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 [VPC 엔드포인트로 서비스 액세스 제어 \(p. 271\)](#) 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.
- IAM 정책 또는 버킷 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. 따라서 VPC 엔드포인트를 통해서 Amazon S3에 요청할 때는 IAM 정책에 `aws:SourceIp` 조건을 사용할 수 없습니다. 이는 사용자 및 역할의 IAM 정책과 모든 버킷 정책에 적용됩니다. 문에 `aws:SourceIp` 조건이 포함되어 있는 경우, 값이 제공된 IP 주소 또는 범위와 일치하지 않습니다. 대신에 다음 작업을 할 수 있습니다.
 - 라우팅 테이블을 사용하여 엔드포인트를 통해 Amazon S3의 리소스에 액세스할 수 있는 인스턴스를 제어할 수 있습니다.
 - 버킷 정책의 경우 특정 엔드포인트 또는 특정 VPC에 대한 액세스를 제한할 수 있습니다. 자세한 내용은 [Amazon S3 버킷 정책 사용 \(p. 265\)](#) 단원을 참조하십시오.
- 현재 엔드포인트는 교차 리전 요청—을 지원하지 않습니다. 버킷과 같은 리전에 엔드포인트를 생성하는지 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 정보는 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. [게이트웨이 엔드포인트 제한 \(p. 261\)](#)

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
Amazon AppStream 2.0	엔드포인트 정책이 사용자 콘텐츠를 저장할 수 있도록 AppStream 2.0이 사용하는 특정 버킷에 대한 액세스를 허용해야 합니다. 자세한 내용은 Amazon AppStream 2.0 개발자 안내서의 Home 폴더 및 VPC 엔드포인트 단원을 참조하십시오.
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 AWS CloudFormation 및 VPC 엔드포인트 를 참조하십시오.
CodeDeploy	엔드포인트 정책이 Amazon S3에 대해 모든 액세스를 허용하거나, 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 Amazon S3와 함께 Elastic Beanstalk 사용 을 참조하십시오.
AWS OpsWorks	엔드포인트 정책이 최소한 AWS OpsWorks가 사용하는 특정 버킷에 대해 액세스를 허용해야 합니다. 자세한 내용은 AWS OpsWorks User Guide의 VPC에서 스택 실행 을 참조하십시오.
AWS 시스템 관리자	AWS 리전의 패치 기준선 작업을 위해, 엔드포인트 정책이 패치 관리자가 사용하는 Amazon S3 버킷에 대한 액세스를 허용해야 합니다. 이 버킷에는 패치 기준선 서비스가 가져와 인스턴스에서 실행하는 코드가 포함되어 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 시스템 관리자에 대한 VPC 엔드포인트 설정 단원을 참조하십시오.

AWS 서비스	참고
	이 작업의 SSM 에이전트에 필요한 S3 버킷 권한 목록은 AWS 시스템 관리자 사용 설명서의 SSM 에이전트를 위한 최소 S3 버킷 권한 단원을 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 S3에 직접 의존하지 않습니다. 하지만 Amazon WorkSpaces 사용자에게 인터넷 액세스를 제공한 경우 다른 회사의 웹사이트, HTML 이메일 및 인터넷 서비스는 Amazon S3에 의존할 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. 자세한 내용은 [VPC 엔드포인트 정책 사용 \(p. 271\)](#) 단원을 참조하십시오.

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.

Example 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_bucket에 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/*"
      ]
    }
  ]
}
```

Example 예: Amazon Linux AMI 리포지토리 액세스 활성화

Amazon Linux AMI 리포지토리는 각 리전의 Amazon S3 버킷입니다. VPC의 인스턴스가 엔드포인트를 통해 리포지토리에 액세스하도록 하려면 이러한 버킷에 액세스하게 하는 엔드포인트 정책을 만들 수 있습니다.

다음 정책은 사용자가 Amazon Linux 리포지토리에 읽기 전용 액세스를 할 수 있도록 허용합니다.

```
{
  "Statement": [
    {
      "Sid": "AmazonLinuxAMIRepositoryAccess",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::packages.*.amazonaws.com/*",
        "arn:aws:s3:::repo.*.amazonaws.com/*"
      ]
    }
  ]
}
```

다음 정책은 사용자가 Amazon Linux 2 리포지토리에 읽기 전용 액세스를 할 수 있도록 허용합니다.

```
{
  "Statement": [
    {
      "Sid": "AmazonLinux2AMIRepositoryAccess",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
      ]
    }
  ]
}
```

Amazon S3 버킷 정책 사용

버킷 정책을 사용하여 특정 엔드포인트 또는 특정 VPC의 버킷에 대한 액세스를 제어할 수 있습니다.

VPC 엔드포인트를 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.

Example 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 엔드포인트가 사용되지 않으면 버킷에 대한 모든 액세스를 거부합니다. `aws:sourceVpce` 조건이 엔드포인트를 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"
        }
      }
    }
  ]
}
```

Example 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-111bbb22`가 `my_secure_bucket`과 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"
        }
      }
    }
  ]
}
```

Amazon DynamoDB에 대한 엔드포인트

VPC에서 DynamoDB 테이블로의 액세스를 이미 설정한 경우, 엔드포인트를 설정한 후에도 보통 때처럼 테이블에 계속 액세스할 수 있습니다. 하지만 다음에 유의하십시오.

- 엔드포인트에는 DynamoDB 리소스에 액세스하기 위한 엔드포인트 사용을 제어하는 정책이 있습니다. 기본 정책은 모든 AWS 계정부터 모든 DynamoDB 리소스까지 자격 증명을 사용하는 VPC 내 모든 사용자 또는 서비스의 액세스를 허용합니다. 자세한 내용은 [VPC 엔드포인트로 서비스 액세스 제어 \(p. 271\)](#) 단원을 참조하십시오.
- DynamoDB는 리소스 기반 정책(예: 테이블에 대해)을 지원하지 않습니다. DynamoDB에 대한 액세스는 개별 IAM 사용자 및 역할에 대한 엔드포인트 정책 및 IAM 정책을 통해 제어됩니다.
- VPC 엔드포인트를 통해 Amazon DynamoDB 스트림에 액세스할 수 없습니다.
- 현재 엔드포인트는 교차 리전 요청을 지원하지 않습니다.—DynamoDB 테이블과 같은 리전에 엔드포인트를 생성하는지 확인하십시오.
- AWS CloudTrail를 사용하여 DynamoDB 작업을 기록할 경우 로그 파일에는 VPC에 있는 EC2 인스턴스의 프라이빗 IP 주소와 엔드포인트를 통해 수행된 모든 작업에 대한 엔드포인트 ID가 포함되어 있습니다.
- 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 DynamoDB에 다시 연결할 수 있는지를 테스트하는 것이 좋습니다.

DynamoDB에서 엔드포인트를 사용하기 전에 다음의 일반 제한을 읽으십시오. [게이트웨이 엔드포인트 제한 \(p. 261\)](#)

DynamoDB용 엔드포인트 정책 사용

다음은 DynamoDB에 액세스하기 위한 엔드포인트 정책의 예입니다.

Important

IAM 사용자 정책, 엔드포인트 정책과 같은 모든 유형의 정책은 DynamoDB에 액세스하기 위해 필요한 권한을 부여해야 합니다.

Example 예: 읽기 전용 액세스

VPC 엔드포인트를 통한 DynamoDB 테이블 나열 및 설명으로만 작업을 제한하는 정책을 만들 수 있습니다.

```
{
  "Statement": [
    {
      "Sid": "ReadOnly",
      "Principal": "*",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTables"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Example 예: 특정 테이블에 대한 액세스 제한

특정 DynamoDB 테이블에 대한 액세스를 제한하는 정책을 만들 수 있습니다. 이 예의 엔드포인트 정책은 stockTable에 대해서만 액세스를 허용합니다.

```
{
  "Statement": [
    {

```

```
{
  "Sid": "AccessToSpecificTable",
  "Principal": "*",
  "Action": [
    "dynamodb:Batch*",
    "dynamodb:Delete*",
    "dynamodb:DescribeTable",
    "dynamodb:GetItem",
    "dynamodb:PutItem",
    "dynamodb:Update*"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/StockTable"
}
```

IAM 정책을 사용하여 DynamoDB에 대한 액세스 제어

IAM 사용자, 그룹 또는 역할이 특정 VPC 엔드포인트에서만 DynamoDB 테이블에 액세스하도록 제한하는 IAM 정책을 만들 수 있습니다. 이렇게 하려면 IAM 정책에서 테이블 리소스에 대한 `aws:sourceVpc` 조건 키를 사용할 수 있습니다.

DynamoDB에 대한 액세스를 관리하는 방법에 대한 자세한 내용은 Amazon DynamoDB 개발자 안내서의 [Amazon DynamoDB에서의 인증 및 액세스 제어](#) 단원을 참조하십시오.

Example 예: 특정 엔드포인트에서의 액세스 제한

이 예에서는 엔드포인트 `vpce-11aa22bb`를 통해 액세스하는 경우를 제외하고 사용자의 DynamoDB 테이블 사용 권한이 거부됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessFromSpecificEndpoint",
      "Action": "dynamodb:*",
      "Effect": "Deny",
      "Resource": "arn:aws:dynamodb:region:account-id:table/*",
      "Condition": { "StringNotEquals" : { "aws:sourceVpc": "vpce-11aa22bb" } }
    }
  ]
}
```

게이트웨이 엔드포인트 생성

엔드포인트를 만들려면 엔드포인트를 만들려는 VPC 및 연결을 설정할 서비스를 지정해야 합니다.

콘솔을 사용하여 게이트웨이 엔드포인트를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]와 [Create Endpoint]를 선택합니다.
3. [Service Name]에서 연결할 서비스를 선택합니다. DynamoDB 또는 Amazon S3에 대한 게이트웨이 엔드포인트를 생성하려면 Type(유형) 열이 게이트웨이를 나타내야 합니다.
4. 다음 정보를 입력한 다음 [Create endpoint]를 선택합니다.
 - [VPC]에서 엔드포인트를 생성할 VPC를 선택합니다.
 - [Configure route tables]에서 엔드포인트에서 사용할 라우팅 테이블을 선택합니다. 엔드포인트, 선택한 라우팅 테이블에 대한 서비스를 향하는 트래픽을 가리키는 경로가 자동으로 추가됩니다.

- [Policy]에서 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.

엔드포인트를 생성한 후 그에 관한 정보를 볼 수 있습니다.

콘솔을 사용하여 게이트웨이 엔드포인트에 대한 정보를 확인하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 엔드포인트를 선택합니다.
3. 엔드포인트에 대한 정보를 보려면 [Summary]를 선택합니다. [Service] 상자에서 서비스에 대한 접두사 목록 이름을 가져올 수 있습니다.
4. 엔드포인트가 사용하는 라우팅 테이블에 대한 정보를 보려면 [Route Tables]를 선택합니다.
5. 엔드포인트에 연결된 IAM 정책을 보려면 정책을 선택합니다.

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.

AWS CLI를 사용하여 엔드포인트를 생성하고 보려면

1. [describe-vpc-endpoint-services](#) 명령을 사용하여 사용 가능한 목록을 가져옵니다. 반환된 출력에 표시된 연결하고자 하는 서비스의 이름을 메모해 둡니다. `serviceType` 필드는 인터페이스 엔드포인트 또는 게이트웨이 엔드포인트를 통해 서비스에 연결할지 여부를 나타냅니다.

```
aws ec2 describe-vpc-endpoint-services
```

```
{
  "serviceDetailSet": [
    {
      "serviceType": [
        {
          "serviceType": "Gateway"
        }
      ]
    }
  ]
}
```

2. 게이트웨이 엔드포인트(예: Amazon S3에 대한 엔드포인트)를 생성하려면 [create-vpc-endpoint](#) 명령을 사용하고 VPC ID, 서비스 이름, 엔드포인트를 사용할 라우팅 테이블을 지정합니다. 선택적으로 `--policy-document` 파라미터를 사용하여 서비스 액세스를 제어할 사용자 지정 정책을 지정할 수 있습니다. 파라미터를 사용하지 않는 경우 서비스에 대한 전체 액세스를 허용하는 기본 정책이 연결됩니다.

```
aws ec2 create-vpc-endpoint --vpc-id vpc-1a2b3c4d --service-name com.amazonaws.us-east-1.s3 --route-table-ids rtb-11aa22bb
```

3. [describe-vpc-endpoints](#) 명령을 사용하여 엔드포인트를 설명합니다.

```
aws ec2 describe-vpc-endpoints
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 사용 가능한 서비스를 설명하려면

- [Get-EC2VpcEndpointService](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpointServices](#)(Amazon EC2 Query API)

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 만들려면

- [New-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [CreateVpcEndpoint](#)(Amazon EC2 Query API)

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 설명하려면

- [Get-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [DescribeVpcEndpoints](#)(Amazon EC2 Query API)

보안 그룹 수정

인스턴스에 연결된 VPC 보안 그룹이 아웃바운드 트래픽을 제한할 경우, AWS 서비스를 향하는 트래픽을 허용하는 규칙을 추가하여 인스턴스를 유지해야 합니다.

게이트웨이 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. AWS 서비스의 접두사 목록 ID를 선택하거나 입력합니다.
6. Save를 선택합니다.

보안 그룹에 대한 자세한 정보는 [VPC의 보안 그룹 \(p. 123\)](#) 단원을 참조하십시오.

명령줄 또는 API를 사용하여 접두사 목록 이름, ID 및 AWS 서비스에 대한 IP 주소 범위를 가져오려면

- [describe-prefix-lists](#)(AWS CLI)
- [Get-EC2PrefixList](#)(Windows PowerShell용 AWS 도구)
- [DescribePrefixLists](#)(Amazon EC2 Query API)

게이트웨이 엔드포인트 수정

정책을 변경하거나 삭제하고, 엔드포인트에서 사용하는 라우팅 테이블을 추가하거나 삭제하여 게이트웨이 엔드포인트를 수정할 수 있습니다.

게이트웨이 엔드포인트와 연결된 정책을 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 엔드포인트를 선택합니다.
3. [Actions], [Edit policy]를 선택합니다.
4. [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. 완료되면 [Save]를 선택합니다.

Note

정책 변경 사항이 적용되려면 몇 분 정도 걸릴 수 있습니다.

게이트웨이 엔드포인트가 사용하는 라우팅 테이블을 추가 또는 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 엔드포인트를 선택합니다.
3. [Actions], [Manage route tables]를 선택합니다.
4. 필요한 라우팅 테이블을 선택하거나 선택 취소한 후 라우팅 테이블 수정을 선택합니다.

AWS CLI를 사용하여 게이트웨이 엔드포인트를 수정하려면

1. `describe-vpc-endpoints` 명령을 사용하여 게이트웨이 엔드포인트의 ID를 가져옵니다.

```
aws ec2 describe-vpc-endpoints
```

2. 다음 예제는 `modify-vpc-endpoint` 명령을 사용하여 라우팅 테이블 `rtb-aaa222bb`를 게이트웨이 엔드포인트에 연결하고 정책 문서를 재설정합니다.

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-ids rtb-aaa222bb --reset-policy
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 VPC 엔드포인트를 수정하려면

- [Edit-EC2VpcEndpoint](#)(Windows PowerShell용 AWS 도구)
- [ModifyVpcEndpoint](#)(Amazon EC2 Query API)

VPC 엔드포인트로 서비스 액세스 제어

엔드포인트를 만들 경우, 연결하려는 서비스에 대한 액세스를 제어하는 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 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 정보는 [Amazon S3 버킷 정책 사용 \(p. 265\)](#) 단원을 참조하십시오.

내용

- [VPC 엔드포인트 정책 사용 \(p. 271\)](#)
- [보안 그룹 \(p. 272\)](#)

VPC 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. 정책 작성에 대한 자세한 정보는 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. 게이트웨이 엔드포인트에 `"AWS": "AWS-account-ID"` 또는 `"AWS": "arn:aws:iam::AWS-account-ID:root"` 형식으로 보안 주체를 지정하는 경우, AWS 계정 루트 사용자에게 대해서만 액세스 권한이 부여되며 계정의 모든 IAM 사용자 및 역할에 대해서 액세스 권한이 부여되는 것은 아닙니다.
- 엔드포인트 정책의 크기는 20,480자를 초과할 수 없습니다(공백 포함).

엔드포인트 정책 예는 다음 주제를 참조하십시오.

- [Amazon S3용 엔드포인트 정책 사용 \(p. 264\)](#)
- [DynamoDB용 엔드포인트 정책 사용 \(p. 267\)](#)
- AWS CodeBuild에서 VPC 엔드포인트를 사용합니다. 자세한 정보는 AWS CodeBuild 사용 설명서의 [VPC 엔드포인트 사용](#)을 참조하십시오.

보안 그룹

기본적으로 Amazon VPC 보안 그룹은 명시적으로 아웃바운드 액세스를 제한하지 않는 한 모든 아웃바운드 트래픽을 허용합니다.

인터페이스 엔드포인트를 생성할 때 서비스와의 통신에 사용할 수 있는 엔드포인트별 VPC 호스트 이름이 생성됩니다. 보안 그룹을 지정하지 않을 경우 VPC에 대한 기본 보안 그룹이 엔드포인트 네트워크 인터페이스에 자동적으로 연결됩니다. 보안 그룹에 대한 규칙이 엔드포인트 네트워크 인터페이스 및 서비스와 통신하는 VPC의 리소스 사이의 통신을 허용하도록 해야 합니다.

게이트웨이 엔드포인트에서 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 정보는 [보안 그룹 수정 \(p. 270\)](#) 단원을 참조하십시오.

VPC 엔드포인트 삭제

엔드포인트가 더 이상 필요하지 않으면 이를 삭제할 수 있습니다. 게이트웨이 엔드포인트를 삭제하면 엔드포인트가 사용하는 라우팅 테이블의 엔드포인트 라우팅도 삭제됩니다. 하지만 엔드포인트가 상주하는 VPC와 연결된 보안 그룹에는 아무런 영향이 없습니다. 인터페이스 엔드포인트를 삭제하면 엔드포인트 네트워크 인터페이스가 삭제됩니다.

엔드포인트를 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoints]를 선택한 후 엔드포인트를 선택합니다.
3. [Actions], [Delete Endpoint]를 차례로 선택합니다.
4. 확인 화면에서 [Yes, Delete]를 선택합니다.

VPC 엔드포인트를 삭제하려면

- `delete-vpc-endpoints`(AWS CLI)
- `Remove-EC2VpcEndpoint`(Windows PowerShell용 AWS 도구)
- `DeleteVpcEndpoints`(Amazon EC2 Query API)

VPC 엔드포인트 서비스(AWS PrivateLink)

VPC에서 자체 애플리케이션을 생성하고 AWS PrivateLink 구동 서비스(엔드포인트 서비스라고도 함)로서 구성할 수 있습니다. 기타 AWS 보안 주체는 [인터페이스 VPC 엔드포인트 \(p. 247\)](#)를 사용하여 VPC에서 엔

드포인트 서비스에 연결을 생성할 수 있습니다. 서비스 공급자인 경우 서비스에 연결을 생성하는 AWS 보안 주체는 서비스 소비자가 됩니다.

콘텐츠

- [개요 \(p. 273\)](#)
- [엔드포인트 서비스 제한 \(p. 275\)](#)
- [VPC 엔드포인트 서비스 구성 생성 \(p. 275\)](#)
- [엔드포인트 서비스에 대한 권한 추가 및 제거 \(p. 277\)](#)
- [Network Load Balancer 변경 및 설정 수락 \(p. 278\)](#)
- [인터페이스 엔드포인트 연결 요청 수락 및 거부 \(p. 279\)](#)
- [엔드포인트 서비스에 대한 알림 생성 및 관리 \(p. 280\)](#)
- [연결 정보에 대한 프록시 프로토콜 사용 \(p. 282\)](#)
- [엔드포인트 서비스 구성 삭제 \(p. 282\)](#)

개요

다음은 엔드포인트 서비스 생성의 일반적인 단계입니다.

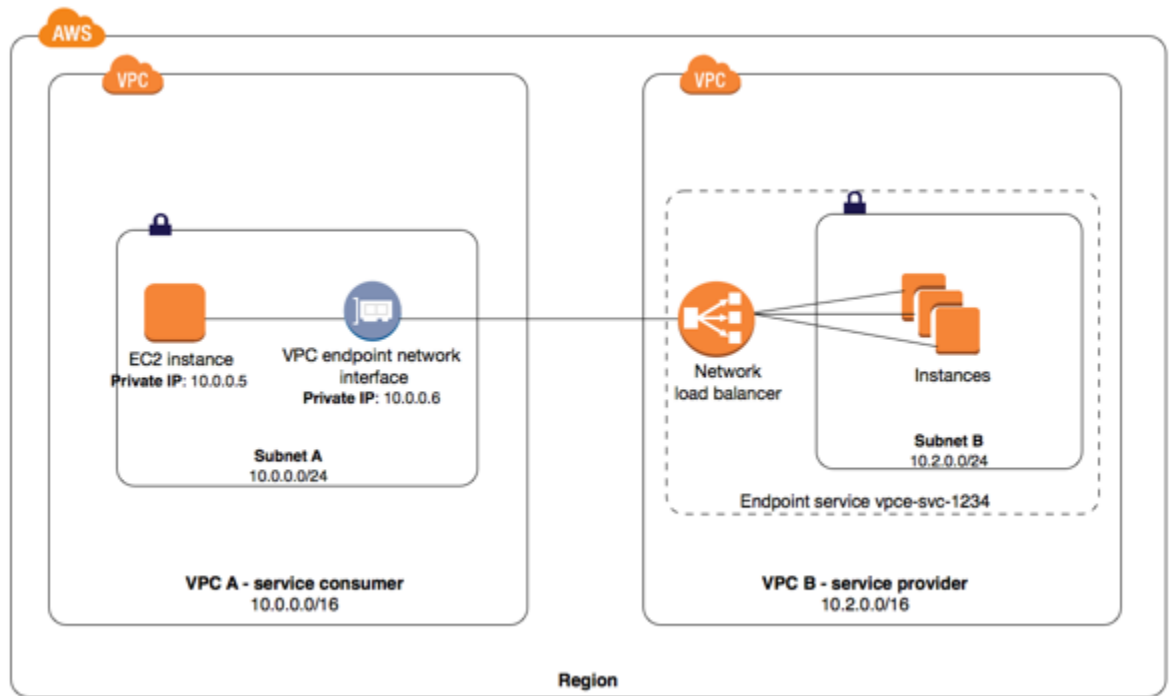
1. VPC에서 애플리케이션에 대한 Network Load Balancer를 생성하고 서비스를 사용할 수 있어야 하는 각 서브넷(가용 영역)에 대해 이를 구성합니다. 로드 밸런서는 서비스 소비자로부터 요청을 받아 서비스로 전달합니다. 자세한 내용은 Network Load Balancer 사용 설명서의 [네트워크 로드 밸런서 시작하기](#) 단원을 참조하십시오. 리전 내 모든 가용 영역에 서비스를 구성하는 것이 좋습니다.
2. VPC 엔드포인트 서비스 구성을 생성하고 Network Load Balancer를 지정합니다.

다음은 서비스 소비자를 서비스에 연결해주는 일반적인 단계입니다.

1. 특정 서비스 소비자(AWS 계정, IAM 사용자 및 IAM 역할)에게 엔드포인트 서비스에 연결을 생성할 수 있는 권한을 부여합니다.
2. 권한을 부여 받은 서비스 소비자는 선택적으로 사용자가 사용자 서비스를 구성한 각 가용 영역 내에서 서비스에 대한 인터페이스 엔드포인트를 생성합니다.
3. 연결을 활성화하려면 인터페이스 엔드포인트 연결 요청을 수락합니다. 기본적으로 연결 요청은 수동으로 수락되어야 합니다. 그러나 연결 요청을 자동으로 수락하도록 엔드포인트 서비스에 대한 수락 설정을 구성할 수도 있습니다.

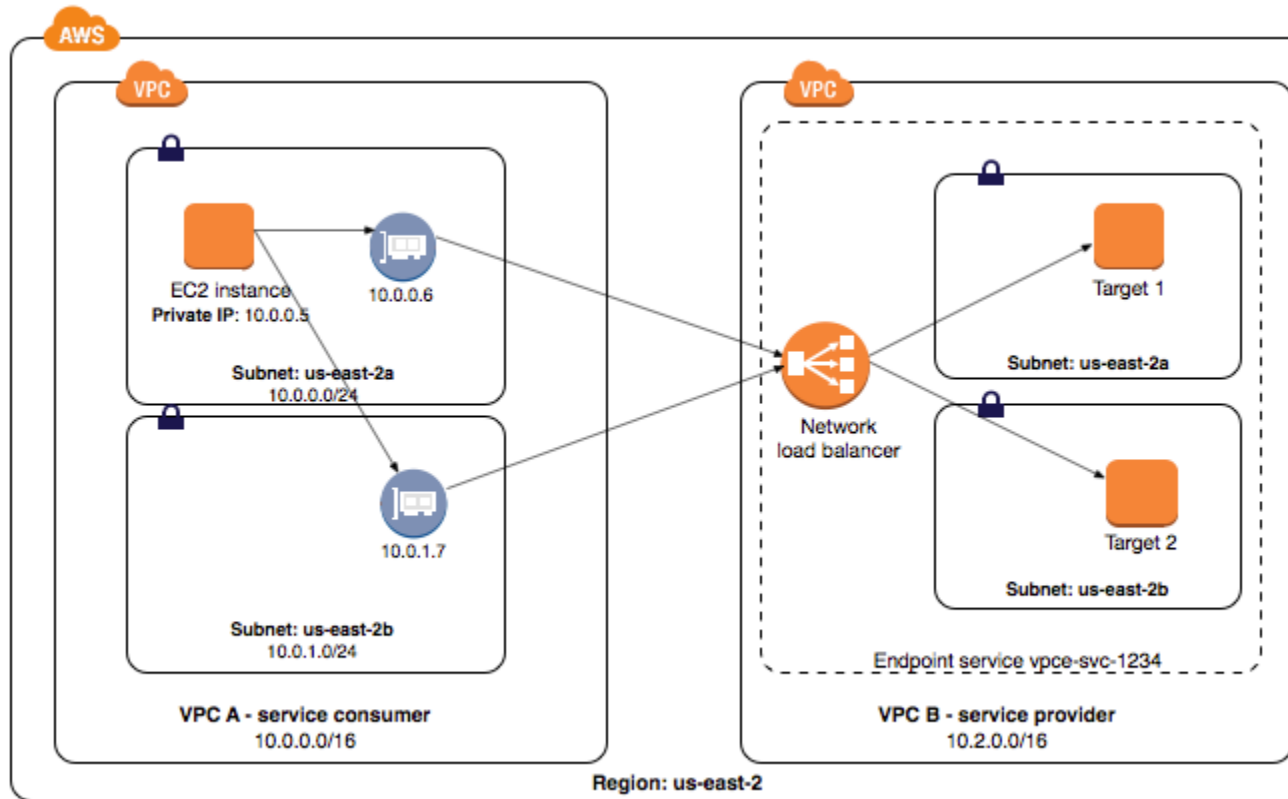
권한 및 수락 설정을 조합하면 서비스에 액세스할 수 있는 서비스 소비자(AWS 보안 주체)를 제어하는 데 도움이 될 수 있습니다. 예를 들어 신뢰할 수 있고 자동으로 모든 연결 요청을 수락하는 선택된 주체에 권한을 부여하거나, 더 넓은 범위의 주체 그룹에 권한을 부여하고 신뢰할 수 있는 특정 연결 요청을 수동으로 수락할 수 있습니다.

다음 다이어그램에서 VPC B의 계정 소유자가 서비스 공급자이고, 서브넷 B의 인스턴스에서 실행 중인 서비스가 있습니다. VPC B의 소유자는 서브넷 B의 인스턴스를 대상으로 가리키는 연결된 Network Load Balancer를 포함한 서비스 엔드포인트(vpc-svc-1234)를 보유하고 있습니다. VPC A의 서브넷 A에 있는 인스턴스는 인터페이스 엔드포인트를 사용하여 서브넷 B의 서비스에 액세스합니다.



낮은 지연 시간과 내결함성을 위해 AWS 리전의 모든 가용 영역에 대상을 포함한 Network Load Balancer를 사용하는 것을 권장합니다. 서비스 액세스에 [영역 단위 DNS 호스트 이름 \(p. 257\)](#)을 사용하는 서비스 소비자에 고가용성을 보장하는 데 도움을 받기 위해 교차 영역 로드 밸런싱을 활성화시킬 수 있습니다. 교차 영역 로드 밸런싱은 로드 밸런서를 활성화해 모든 활성 가용 영역의 등록된 대상으로 트래픽을 분산시킵니다. 자세한 내용은 Network Load Balancer 사용 설명서의 [교차 영역 로드 밸런싱](#) 단원을 참조하십시오. 교차 영역 로드 밸런싱이 활성화된 경우 리전별 데이터 전송 요금이 계정에 적용될 수 있습니다.

다음 다이어그램에서 VPC B의 소유자는 서비스 공급자이며, 두 개 가용 영역에 대상이 포함된 Network Load Balancer를 구성했습니다. 서비스 소비자(VPC A)는 자신의 VPC에 동일한 두 개 가용 영역의 인터페이스 엔드포인트를 생성했습니다. VPC A 인스턴스에서 서비스를 요청하는데 두 인터페이스 엔드포인트 중 하나를 사용할 수 있습니다.



엔드포인트 서비스 제한

엔드포인트 서비스를 사용하려면 현재 규칙과 제한 사항을 알고 있어야 합니다.

- 엔드포인트 서비스에 태그를 지정할 수 없습니다.
- 엔드포인트 서비스는 TCP를 통한 IPv4 트래픽만 지원합니다.
- 서비스 소비자는 엔드포인트별 DNS 호스트 이름을 사용하여 엔드포인트 서비스에 액세스해야 합니다. 프라이빗 DNS는 지원되지 않습니다. 자세한 내용은 [인터페이스 엔드포인트를 통해 서비스 액세스 \(p. 257\)](#) 단원을 참조하십시오.
- 엔드포인트 서비스는 생성한 AWS 리전에서만 사용 가능합니다.
- 엔드포인트 서비스가 여러 개의 Network Load Balancer에 연결되어 있는 경우, 특정 가용 영역에 대해 인터페이스 엔드포인트는 로드 밸런서 한 개와만의 연결을 설정합니다.
- 사용자 계정의 가용 영역이 다른 계정의 가용 영역과 동일한 위치로 매핑되지 않을 수 있습니다. 예를 들어 가용 영역인 us-east-1a가 다른 계정의 us-east-1a와 위치가 다를 수 있습니다. 자세한 내용은 [리전 및 가용 영역 개념](#)을 참조하십시오. 엔드포인트를 구성하는 경우 사용자 계정으로 매핑된 가용 영역으로 구성됩니다.

VPC 엔드포인트 서비스 구성 생성

Amazon VPC 콘솔 또는 명령줄을 사용하여 엔드포인트 서비스 구성을 생성할 수 있습니다. 시작하기 전에 서비스에 대한 VPC에 하나 이상의 Network Load Balancer가 있는지 확인합니다. 자세한 내용은 Network Load Balancer 사용 설명서의 [네트워크 로드 밸런서 시작하기](#) 단원을 참조하십시오.

구성에서, 사용자가 서비스에 대한 인터페이스 엔드포인트 연결 요청을 수동으로 수락해야 하도록 선택적으로 지정할 수 있습니다. [알림을 생성 \(p. 280\)](#)하여 연결 요청이 있을 때 알림을 받을 수 있습니다. 연결을 수락하지 않으면 서비스 소비자가 서비스에 액세스할 수 없습니다.

Note

수락 설정에 관계없이 서비스 소비자는 서비스에 대해 연결을 생성할 수 있는 [권한 \(p. 277\)](#)이 있어야 합니다.

콘솔을 사용하여 엔드포인트 서비스를 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 [Create Endpoint Service]를 선택합니다.
3. Associate Network Load Balancers(네트워크 로드 밸런서 연결)에서 엔드포인트 서비스에 연결할 Network Load Balancer를 선택합니다.
4. [Require acceptance for endpoint]에서 서비스에 대한 연결 요청을 수동으로 수락한다는 확인란을 선택합니다. 이 옵션을 선택하지 않은 경우 엔드포인트 연결은 자동적으로 수락됩니다.
5. [Create service]를 선택합니다.

엔드포인트 서비스 구성을 생성한 후에는 서비스 소비자가 서비스에 대한 인터페이스 엔드포인트를 생성하도록 해주는 권한을 추가해야 합니다.

AWS CLI를 사용하여 엔드포인트 서비스를 생성하려면

- [create-vpc-endpoint-service-configuration](#) 명령을 사용하고 Network Load Balancer에 대한 하나 이상의 ARN을 지정합니다. 선택적으로 서비스 연결 수락이 필요한지 여부를 지정할 수 있습니다.

```
aws ec2 create-vpc-endpoint-service-configuration --network-load-balancer-arns
arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532 --acceptance-required
```

```
{
  "ServiceConfiguration": {
    "ServiceType": [
      {
        "ServiceType": "Interface"
      }
    ],
    "NetworkLoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532"
    ],
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",
    "ServiceState": "Available",
    "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
    "AcceptanceRequired": true,
    "AvailabilityZones": [
      "us-east-1d"
    ],
    "BaseEndpointDnsNames": [
      "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"
    ]
  }
}
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 엔드포인트 서비스를 생성하려면

- [New-EC2VpcEndpointServiceConfiguration](#)(Windows PowerShell용 AWS 도구)

- [CreateVpcEndpointServiceConfiguration](#)(Amazon EC2 쿼리 API)

엔드포인트 서비스에 대한 권한 추가 및 제거

엔드포인트 서비스 구성을 생성한 후에는 어떤 서비스 소비자가 서비스에 연결할 인터페이스 엔드포인트를 생성할 수 있는지를 제어할 수 있습니다. 서비스 소비자는 [IAM 보안 주체](#)—IAM 사용자, IAM 역할 및 AWS 계정입니다. 보안 주체에 대한 권한을 추가하거나 제거하려면, ARN(Amazon 리소스 이름)이 필요합니다.

- AWS 계정(및 계정에 있는 모든 보안 주체)의 경우, ARN이 `arn:aws:iam::aws-account-id:root` 양식으로 되어 있습니다.
- 특정 IAM 사용자의 경우, ARN이 `arn:aws:iam::aws-account-id:user/user-name` 양식으로 되어 있습니다.
- 특정 IAM 역할의 경우, ARN이 `arn:aws:iam::aws-account-id:role/role-name` 양식으로 되어 있습니다.

콘솔을 사용하여 권한을 추가 또는 제거하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Actions], [Add principals to whitelist]를 선택합니다.
4. 권한을 추가할 주체에 대한 ARN을 지정합니다. 주체를 더 추가하려면 [Add principal]을 선택합니다. 주체를 제거하려면 항목 옆에 있는 x 아이콘을 선택합니다.

Note

*를 지정하면 모든 주체에 권한을 추가합니다. 이렇게 하면 모든 AWS 계정에 있는 모든 보안 주체가 엔드포인트 서비스에 대해 인터페이스 엔드포인트를 생성할 수 있습니다.

5. [Add to Whitelisted principals]를 선택합니다.
6. 주체를 제거하려면 목록에서 주체를 선택한 다음 [Delete]를 선택합니다.

AWS CLI를 사용하여 권한을 추가 및 제거하려면

1. 엔드포인트 서비스에 대한 권한을 추가하려면 [modify-vpc-endpoint-service-permissions](#) 명령을 사용하고 `--add-allowed-principals` 파라미터를 사용하여 주체에 대한 하나 이상의 ARN을 추가합니다.

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
--add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

2. 엔드포인트 서비스에 대해 추가한 권한을 보려면 [describe-vpc-endpoint-service-permissions](#) 명령을 사용합니다.

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
```

```
{
  "AllowedPrincipals": [
    {
      "PrincipalType": "Account",
      "Principal": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

3. 엔드포인트 서비스에 대한 권한을 제거하려면 `modify-vpc-endpoint-service-permissions` 명령을 사용하고 `--remove-allowed-principals` 파라미터를 사용하여 주체에 대한 하나 이상의 ARN을 제거합니다.

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
--remove-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 엔드포인트 서비스 권한을 수정하려면

- `Edit-EC2EndpointServicePermission`(Windows PowerShell용 AWS 도구)
- `ModifyVpcEndpointServicePermissions`(Amazon EC2 쿼리 API)

Network Load Balancer 변경 및 설정 수락

엔드포인트 서비스에 연결된 Network Load Balancer를 변경하고 엔드포인트 서비스 연결 요청 시 수락이 필요한지 여부를 변경함으로써 엔드포인트 서비스 구성을 수정할 수 있습니다.

엔드포인트 서비스에 연결된 인터페이스 엔드포인트가 있는 경우 로드 밸런서를 연결 해제할 수 없습니다.

콘솔을 사용하여 엔드포인트 서비스에 대한 네트워크 로드 밸런서를 변경하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Actions], [Associate/Disassociate Network Load Balancers]를 선택합니다.
4. 필요에 따라 로드 밸런서를 선택하거나 선택 취소한 후 [Save]를 선택합니다.

콘솔을 사용하여 수락 설정을 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Actions], [Modify endpoint acceptance setting]를 선택합니다.
4. [Require acceptance for endpoint]를 선택 또는 선택 취소한 다음 [Modify]를 선택합니다.

AWS CLI를 사용하여 로드 밸런서 및 수락 설정을 수정하려면

1. 엔드포인트 서비스에 대한 로드 밸런서를 변경하려면 `modify-vpc-endpoint-service-configuration` 명령을 사용하고 `--add-network-load-balancer-arn` 또는 `--remove-network-load-balancer-arn` 파라미터를 사용합니다. 예를 들면 다음과 같습니다.

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --remove-network-load-balancer-arn
arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532
```

2. 수락이 필요한지 여부를 변경하려면 `modify-vpc-endpoint-service-configuration` 명령을 사용하고 `--acceptance-required` 또는 `--no-acceptance-required`를 지정합니다. 예를 들면 다음과 같습니다.

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 엔드포인트 서비스 구성을 수정하려면

- [Edit-EC2VpcEndpointServiceConfiguration](#) (Windows PowerShell용 AWS 도구)
- [ModifyVpcEndpointServiceConfiguration](#)(Amazon EC2 쿼리 API)

인터페이스 엔드포인트 연결 요청 수락 및 거부

엔드포인트 서비스를 생성한 후에는 권한이 추가된 서비스 소비자가 서비스에 연결할 인터페이스 엔드포인트를 생성할 수 있습니다. 인터페이스 엔드포인트에 대한 자세한 내용은 [인터페이스 VPC 엔드포인트 \(AWS PrivateLink\)](#) (p. 247) 단원을 참조하십시오.

연결 요청에 대한 수락이 필요하다고 지정한 경우 엔드포인트 서비스에 대한 인터페이스 엔드포인트 연결 요청을 수동으로 수락 또는 거부해야 합니다. 인터페이스 엔드포인트가 수락되면 `available` 상태가 됩니다.

`available` 상태가 된 이후 인터페이스 엔드포인트 연결을 거부할 수 있습니다.

콘솔을 사용하여 연결 요청을 수락 또는 거부하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Endpoint Connections] 탭에는 현재 수락 대기 중인 엔드포인트 연결이 나열됩니다. 엔드포인트를 선택하고 [Actions]를 선택한 다음 [Accept endpoint connection request]를 선택하여 연결을 수락하거나 [Reject endpoint connection request]를 선택하여 연결을 거부합니다.

AWS CLI를 사용하여 연결 요청을 수락 또는 거부하려면

1. 수락 대기 중인 엔드포인트 연결을 보려면 [describe-vpc-endpoint-connections](#) 명령을 사용하고 `pendingAcceptance` 상태로 필터링합니다.

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-state,Values=pendingAcceptance
```

```
{
  "VpcEndpointConnections": [
    {
      "VpcEndpointId": "vpce-0c1308d7312217abc",
      "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
      "CreationTimestamp": "2017-11-30T10:00:24.350Z",
      "VpcEndpointState": "pendingAcceptance",
      "VpcEndpointOwner": "123456789012"
    }
  ]
}
```

2. 엔드포인트 연결 요청을 수락하려면 [accept-vpc-endpoint-connections](#) 명령을 사용하고 엔드포인트 ID와 엔드포인트 서비스 ID를 지정합니다.

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --vpc-endpoint-ids vpce-0c1308d7312217abc
```

3. 엔드포인트 연결 요청을 거부하려면 [reject-vpc-endpoint-connections](#) 명령을 사용하고 엔드포인트 ID와 엔드포인트 서비스 ID를 지정합니다.

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --vpc-endpoint-ids vpce-0c1308d7312217abc
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 엔드포인트 연결을 수락하거나 거절하려면

- [Confirm-EC2EndpointConnection](#) 및 [Deny-EC2EndpointConnection](#)(Windows PowerShell용 AWS 도구)
- [AcceptVpcEndpointConnections](#) 및 [RejectVpcEndpointConnections](#)(Amazon EC2 쿼리 API)

엔드포인트 서비스에 대한 알림 생성 및 관리

엔드포인트 서비스에 연결된 엔드포인트에서 발생하는 특정 이벤트에 대한 알림을 받도록 알림을 생성할 수 있습니다. 예를 들어 엔드포인트 서비스에 대한 엔드포인트 요청이 수락 또는 거부될 때 이메일을 받을 수 있습니다. 알림을 생성하려면 Amazon SNS 주제를 알림에 연결해야 합니다. SNS 주제를 구독하여 엔드포인트 이벤트가 발생할 때 이메일 알림을 받을 수 있습니다. 자세한 내용은 [Amazon Simple Notification Service 개발자 안내서](#)를 참조하십시오.

알림에 대해 사용할 Amazon SNS 주제는 Amazon VPC 엔드포인트 서비스가 사용자를 대신해 알림을 게시하도록 허용하는 주제 정책을 보유해야 합니다. 주제 정책에 다음 문이 포함되어야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS 주제에 대한 액세스 관리](#) 단원을 참조하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account:topic-name"
    }
  ]
}
```

엔드포인트 서비스에 대한 알림을 생성하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Notifications], [Create Notification]을 선택합니다.
4. SNS 주제가 알림과 연결할 ARN을 선택합니다.
5. [Events]에서 알림을 수신할 대상이 되는 엔드포인트 이벤트를 선택합니다.
6. [Create Notification]을 선택합니다.

알림을 생성한 후 알림에 연결된 SNS 주제를 변경하거나 알림에 대한 다른 엔드포인트 이벤트를 지정할 수 있습니다.

엔드포인트 서비스에 대한 알림을 수정하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Notifications], [Actions], [Modify Notification]을 선택합니다.
4. SNS 주제에 대한 ARN을 지정하고 필요한 경우 엔드포인트 이벤트를 선택 또는 선택 해제합니다.
5. [Modify Notification]을 선택합니다.

알림이 더 이상 필요하지 않으면 삭제할 수 있습니다.

알림을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 엔드포인트 서비스를 선택합니다.
3. [Notifications], [Actions], [Delete Notification]을 선택합니다.
4. 예, 삭제를 선택합니다.

AWS CLI를 사용하여 알림을 생성 및 관리하려면

1. 엔드포인트 서비스에 대한 알림을 생성하려면 [create-vpc-endpoint-connection-notification](#) 명령을 사용하고 SNS 주제의 ARN, 알림을 받을 이벤트, 엔드포인트 서비스 ID를 지정합니다. 예를 들면 다음과 같습니다.

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-events Connect Accept Delete Reject --service-id vpce-svc-1237881c0d25a3abc
```

```
{
  "ConnectionNotification": {
    "ConnectionNotificationState": "Enabled",
    "ConnectionNotificationType": "Topic",
    "ServiceId": "vpce-svc-1237881c0d25a3abc",
    "ConnectionEvents": [
      "Reject",
      "Accept",
      "Delete",
      "Connect"
    ],
    "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
    "ConnectionNotificationArn": "arn:aws:sns:us-east-2:123456789012:VpceNotification"
  }
}
```

2. 알림을 보려면 [describe-vpc-endpoint-connection-notifications](#) 명령을 사용합니다.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

3. 알림에 대한 SNS 주제 및 엔드포인트 이벤트를 변경하려면 [modify-vpc-endpoint-connection-notification](#) 명령을 사용합니다. 예를 들면 다음과 같습니다.

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

4. 알림을 삭제하려면 [delete-vpc-endpoint-connection-notifications](#) 명령을 사용합니다.

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 알림을 생성 및 관리하려면

- [New-EC2VpcEndpointConnectionNotification](#), [Get-EC2EndpointConnectionNotification](#), [Edit-EC2VpcEndpointConnectionNotification](#) 및 [Remove-EC2EndpointConnectionNotification](#)(Windows PowerShell용 AWS 도구)

- [CreateVpcEndpointConnectionNotification](#), [DescribeVpcEndpointConnectionNotifications](#), [ModifyVpcEndpointConnectionNotification](#) 및 [DeleteVpcEndpointConnectionNotifications](#)(Amazon EC2 쿼리 API)

연결 정보에 대한 프록시 프로토콜 사용

Network Load Balancer는 애플리케이션(서비스)에 대한 원본 IP 주소를 제공합니다. 서비스 소비자가 인터페이스 엔드포인트를 통해 서비스로 트래픽을 전송할 때 애플리케이션에 제공된 원본 IP 주소는 Network Load Balancer 노드의 프라이빗 IP 주소이며, 서비스 소비자의 IP 주소가 아닙니다.

서비스 소비자의 IP 주소와 해당 인터페이스 엔드포인트 ID가 필요한 경우 로드 밸런서의 프록시 프로토콜을 활성화하고 프록시 프로토콜 헤더에서 클라이언트 IP 주소를 가져옵니다. 자세한 내용은 Network Load Balancer 사용 설명서의 [프록시 프로토콜](#) 항목을 참조하십시오.

엔드포인트 서비스 구성 삭제

엔드포인트 서비스 구성을 삭제할 수 있습니다. 구성을 삭제해도 VPC 또는 연결된 로드 밸런서에 호스팅된 애플리케이션이 삭제되지 않습니다.

엔드포인트 서비스 구성을 삭제하기 전에 서비스에 연결된 `available` 또는 `pending-acceptance` 상태의 모든 VPC 엔드포인트를 거부해야 합니다. 자세한 내용은 [인터페이스 엔드포인트 연결 요청 수락 및 거부 \(p. 279\)](#) 단원을 참조하십시오.

콘솔을 사용하여 엔드포인트 서비스 구성을 삭제하려면

1. <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Endpoint Services]를 선택한 후 서비스를 선택합니다.
3. [Actions], [Delete]를 선택합니다.
4. 예, 삭제를 선택합니다.

AWS CLI를 사용하여 엔드포인트 서비스 구성을 삭제하려면

- [delete-vpc-endpoint-service-configurations](#) 명령을 사용하고 서비스의 ID를 지정합니다.

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-  
svc-03d5ebb7d9579a2b3
```

Windows PowerShell용 AWS 도구 또는 API를 사용하여 엔드포인트 서비스 구성을 삭제하려면

- [Remove-EC2EndpointServiceConfiguration](#) (Windows PowerShell용 AWS 도구)
- [DeleteVpcEndpointServiceConfigurations](#)(Amazon EC2 쿼리 API)

ClassicLink

ClassicLink를 사용하면 같은 리전 내에 있는 해당 계정의 VPC에 EC2-Classic 인스턴스를 연결할 수 있습니다. 이렇게 하면 VPC 보안 그룹과 EC2-Classic 인스턴스를 연결할 수 있고, 이를 통해 EC2-Classic 인스턴스와 프라이빗 IPv4 주소를 사용하는 VPC 내 인스턴스 간에 통신할 수 있습니다. ClassicLink를 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 \(p. 103\)](#) 단원을 참조하십시오.

ClassicLink는 EC2-Classic 플랫폼을 지원하는 계정을 갖는 모든 사용자에게 제공되며 모든 EC2-Classic 인스턴스에 사용할 수 있습니다.

ClassicLink 사용에 따르는 추가 요금은 없습니다. 데이터 전송 및 인스턴스 시간 사용량에 대한 표준 요금이 그대로 적용됩니다.

ClassicLink에 대한 제사한 내용과 그 사용 방법은 Amazon EC2 사용 설명서의 다음 항목을 참조하십시오.

- [ClassicLink 기본 사항](#)
- [ClassicLink의 제한 사항](#)
- [ClassicLink 작업](#)
- [ClassicLinkAPI 및 CLI 개요](#)

VPN 연결

다음 VPN 연결 옵션을 사용하여 Amazon VPC를 원격 네트워크 및 사용자에게 연결할 수 있습니다.

VPN 연결 옵션	설명
	VPC와 원격 네트워크 사이에 IPsec VPN 연결을 생성할 수 있습니다. AWS 측 연결에서 가상 프라이빗 게이트웨이는 자동 장애 조치를 위한 2개의 VPN 엔드포인트(터널)를 제공합니다. 연결의 원격 측에서 고객 게이트웨이를 구성합니다. 자세한 정보는 사용 설명서 및 Amazon VPC 네트워크 관리자 안내서 를 참조하십시오.
AWS 클라이언트 VPN	AWS 클라이언트 VPN은 온프레미스 네트워크의 AWS 리소스를 안전하게 액세스할 수 있게 해주는 관리형 클라이언트 기반 VPN 서비스입니다. AWS 클라이언트 VPN을 사용하여 사용자가 연결할 수 있는 엔드포인트를 구성하여 보안 TLS VPN 세션을 설정할 수 있습니다. 이렇게 하면 클라이언트가 OpenVPN 기반 VPN 클라이언트를 사용하여 어느 위치에서든 온프레미스 또는 AWS의 리소스를 액세스할 수 있습니다. 자세한 정보는 AWS 클라이언트 VPN 사용 설명서 를 참조하십시오.
AWS VPN CloudHub	원격 네트워크가 두 개 이상인 경우(예: 여러 지사 사무실), 가상 프라이빗 게이트웨이를 통해 여러 개의 연결을 생성하여 이들 네트워크 사이의 통신을 활성화할 수 있습니다. 자세한 정보는 사용 설명서의 VPN CloudHub를 사용하여 사이트 간에 보안 통신 제공 을 참조하십시오.
타사 소프트웨어 VPN 어플라이언스	타사 소프트웨어 VPN 어플라이언스를 실행 중인 VPC에서 Amazon EC2 인스턴스를 사용하여 원격 네트워크에 대한 VPN 연결을 생성할 수 있습니다. AWS는 타사 소프트웨어 VPN 어플라이언스를 제공하거나 유지 관리하지 않지만, 파트너와 오픈 소스 커뮤니티에서 제공하는 다양한 제품 중에서 선택할 수 있습니다. AWS Marketplace 에서 타사 소프트웨어 VPN 어플라이언스를 찾으십시오.

또한 AWS Direct Connect를 사용하여 원격 네트워크에서 VPC까지 전용 프라이빗 연결을 생성할 수도 있습니다. 이 연결을 과 결합하여 IPsec 암호화 연결을 생성할 수 있습니다. 자세한 정보는 [AWS Direct Connect](#)의 AWS Direct Connect 사용 설명서란 무엇입니까?를 참조하십시오.

Amazon VPC 제한

다음 표에는 AWS 계정 관련 각 리전의 Amazon VPC 리소스에 대한 제한 값이 나열되어 있습니다. 달리 표시되지 않는 한 [Amazon VPC 제한 양식](#)을 사용하여 이 제한의 증가를 요청할 수 있습니다. 이러한 제한 일부의 경우, Amazon EC2 콘솔의 제한 페이지를 사용하여 현재 제한을 볼 수 있습니다.

리소스별로 적용되는 제한 증가를 요청하는 경우, AWS는 리전에 있는 모든 리소스의 제한을 증가시킵니다. 예를 들어 VPC별 보안 그룹에 대한 제한은 해당 리전의 모든 VPC에 적용됩니다.

VPC 및 서브넷

리소스	기본 한도	설명
리전당 VPC	5	리전당 인터넷 게이트웨이의 한도는 이 한도와 직접적으로 연관됩니다. 이 한도를 늘리면 리전당 인터넷 게이트웨이에 대한 한도도 같은 수량 만큼 늘어납니다.
VPC당 서브넷	200	-
VPC당 IPv4 CIDR 블록	5	이 한도는 기본 CIDR 블록과 4개의 보조 CIDR 블록으로 이루어져 있습니다.
VPC당 IPv6 CIDR 블록	1	이 한도는 늘릴 수 없습니다.

DNS

자세한 정보는 [DNS 제한 \(p. 241\)](#) 단원을 참조하십시오.

탄력적 IP 주소(IPv4)

리소스	기본 한도	설명
각 리전의 탄력적 IP 주소	5	EC2-VPC에서 사용할 수 있는 탄력적 IP 주소 수에 대한 한도입니다. EC2-Classic에서 사용할 수 있는 탄력적 IP 주소에 대한 자세한 정보는 Amazon Web Services 일반 참조의 Amazon EC2 제한 을 참조하십시오.

흐름 로그

리소스	기본 한도	설명
한 리전에서 단일 네트워크 인터페이스, 단일 서브넷 또는 단일 VPC당 플로우 로그	2	이 한도는 늘릴 수 없습니다. 해당 네트워크 인터페이스가 상주하는 서브넷에 대해 흐름

리소스	기본 한도	설명
		로그를 2개 만들고, VPC에 대해 2개 만들 경우 네트워크 인터페이스당 6개의 흐름 로그가 효과적입니다.

게이트웨이

리소스	기본 한도	설명
리전당 고객 게이트웨이	50	-
리전당 외부 전용 인터넷 게이트웨이	5	이 한도는 리전당 VPC 한도와 직접적인 상관 관계가 있습니다. 이 한도를 늘리려면 리전당 VPC 한도를 늘리십시오. 하나의 VPC에는 한 번에 한 개의 외부 전용 인터넷 게이트웨이만 연결할 수 있습니다.
리전당 인터넷 게이트웨이	5	이 한도는 리전당 VPC 한도와 직접적인 상관 관계가 있습니다. 이 한도를 늘리려면 리전당 VPC 한도를 늘리십시오. 하나의 VPC에는 한 번에 한 개의 인터넷 게이트웨이만 추가할 수 있습니다.
가용 영역당 NAT 게이트웨이	5	pending, active 또는 deleting 상태인 NAT 게이트웨이는 이 한도에 포함됩니다.
리전당 가상 프라이빗 게이트웨이	5	하나의 VPC에는 한 번에 한 개의 가상 프라이빗 게이트웨이만 연결할 수 있습니다.

네트워크 ACL

리소스	기본 한도	설명
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	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 *).
		이 한도는 최대 40개까지 늘릴 수 있지만, 추가 규칙을 처리하기 위해 워크로드가 증가하기 때문에 네트워크 성능에 영향을 줄 수 있습니다.

네트워크 인터페이스

리소스	기본 한도	설명
인스턴스당 네트워크 인터페이스	-	이 한도는 인스턴스 유형에 따라 다릅니다. 자세한 정보는 인스턴스 유형별 ENI당 프라이빗 IP 주소 단원을 참조 하십시오.
리전당 네트워크 인터페이스	350	이 한도는 기본 한도(350) 또는 온디맨드 인스턴스 한도에 5를 곱한 값보다 큼니다. 온디맨드 인스턴스의 기본 한도는 20입니다. 온디맨드 인스턴스 한도가 70 미만인 경우 기본 한도 350이 적용됩니다. 이 한도를 늘리려면 요청을 제출하거나 온디맨드 인스턴스 한도를 늘리십시오.

라우팅 테이블

리소스	기본 한도	설명
VPC당 라우팅 테이블	200	이 한도에는 기본 라우팅 테이블이 포함됩니다.
라우팅 테이블에 따른 경로(전파되지 않는 경로)	50	이 한도는 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	이 한도는 늘릴 수 없습니다. 100개가 넘는 경로 접두사가 필요할 경우 기본 경로를 공급하십시오.

보안 그룹

리소스	기본 한도	설명
리전당 VPC 보안 그룹	2500	최대 한도는 10,000입니다. 리전에 보안 그룹에 5,000개 이상 있다면 성능 향상을 위해 호출에 페이지 번호를 붙여 보안 그룹을 설명하는 것이 좋습니다.
보안 그룹별 인바운드 또는 아웃바운드 규칙	60	보안 그룹별로 60개의 인바운드 규칙과 60개의 아웃바운드 규칙을 지정할 수 있습니다(총 120개 규칙 지정). 이 제한은 IPv4 및 IPv6 규칙에 별도로 강제됩니다. 예를 들어 보안 그룹은 IPv4 트래픽의 경우 인바운드 규칙 60개, IPv6 트래픽의 경우 인바운드 규칙 60개를 보유할 수 있습니다. 보안 그룹이나 접두사 목록 ID를 참조하는 규칙은 IPv4와 IPv6에 대해 각각 한 개의 규칙으로 계수됩니다.

리소스	기본 한도	설명
		한도 변경은 인바운드 규칙과 아웃바운드 규칙에 모두 적용됩니다. 이 한도와 네트워크 인터페이스당 보안 그룹의 한도를 곱한 값이 300을 초과하면 안 됩니다. 예를 들어, 이 한도를 100개로 증가시키면 네트워크 인터페이스당 보안 그룹의 수의 한도는 3개로 감소합니다.
네트워크 인터페이스당 보안 그룹	5	최대 한도는 16입니다. 네트워크 인터페이스당 보안 그룹에 대한 한도와 보안 그룹당 규칙에 대한 한도의 곱은 300을 초과할 수 없습니다. 예를 들어, 이 한도를 10개로 증가시키면 보안 그룹당 규칙 수의 한도는 30개로 감소합니다.

VPC 피어링 연결

리소스	기본 한도	설명
VPC당 활성 VPC 피어링 연결	50	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.
수락되지 않은 VPC 피어링 연결 요청에 대한 만료 시간	1주(168시간)	-

VPC 엔드포인트

리소스	기본 한도	설명
리전당 게이트웨이 VPC 엔드포인트	20	VPC당 게이트웨이 엔드포인트는 255개를 초과할 수 없습니다.
VPC당 인터페이스 VPC 엔드포인트	20	리전당 인터페이스 엔드포인트의 최대 한도는 이 한도에 해당 리전의 VPC 수를 곱한 값입니다.

연결

리소스	기본 한도	설명
리전당 연결	50	-

리소스	기본 한도	설명
VPC당 연결(가상 프라이빗 게이트웨이당)	10	-

VPC 공유

모든 표준 VPC 제한은 공유된 VPC에 적용됩니다.

리소스	기본 한도	설명
VPC를 공유할 수 있는 고유한 계정의 수	30	-
계정과 공유할 수 있는 서브넷의 수	100	-

문서 기록

다음 표에서는 Amazon VPC 사용 설명서, Amazon VPC Peering Guide 및 Amazon VPC 네트워크 관리자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

기능	API 버전	설명	릴리스 날짜
	2016-11-15	AWS 관리형 VPN(현재) 내용이 사용 설명서 로 이동됨.	2018년 12월 18일
VPC 공유	2016-11-15	동일한 VPC에 있는 서브넷을 동일한 AWS 조직의 여러 계정과 공유할 수 있습니다.	2018년 11월 27일
리전 간 피어링	2016-11-15	서로 다른 지역에 있는 VPC 간에는 VPC 피어링 연결을 생성할 수 있습니다. 자세한 정보는 Amazon VPC Peering Guide 단원을 참조하십시오.	2017년 11월 29일
VPC 엔드포인트 서비스	2016-11-15	VPC에서 자체 PrivateLink 서비스를 생성하고 다른 AWS 계정 및 사용자가 인터페이스 VPC 엔드포인트를 통해 서비스에 연결할 수 있도록 할 수 있습니다. 자세한 정보는 VPC 엔드포인트 서비스(AWS PrivateLink) (p. 272) 단원을 참조하십시오.	2017년 11월 28일
기본 서브넷 생성	2016-11-15	기본 서브넷이 없는 가용 영역에서 기본 서브넷을 생성할 수 있습니다. 자세한 정보는 기본 서브넷 생성 (p. 101) 단원을 참조하십시오.	2017년 11월 9일
AWS 서비스에 대한 인터페이스 VPC 엔드포인트	2016-11-15	인터페이스 엔드포인트를 생성하여 일부 AWS 서비스를 비공개로 연결할 수 있습니다. 인터페이스 엔드포인트는 서비스로 전달되는 트래픽에 대한 진입점 역할을 하는 프라이빗 IP 주소를 포함한 네트워크 인터페이스(ENI)입니다. 자세한 정보는 VPC 엔드포인트 (p. 246) 단원을 참조하십시오.	2017년 11월 8일
사용자 지정 ASN	2016-11-15	가상 프라이빗 게이트웨이를 생성할 때 Amazon 측 게이트웨이의 프라이빗 자율 시스템 번호(ASN)를 지정할 수 있습니다. 자세한 정보는 사용 설명서 에서 가상 프라이빗 게이트웨이 를 참조하십시오.	2017년 10월 10일
VPN 터널 옵션	2016-11-15	VPN 터널의 내부 CIDR 블록과 사용자 지정 사전 공유 키를 지정할 수 있습니다. 자세한 정보는 Amazon VPC 네트워크 관리자 안내서의 연결에 대한 VPN 터널 구성, 연결 설정 개요 를 참조하십시오.	2017년 10월 3일
VPN 범주	2016-11-15	VPN 연결의 범주를 볼 수 있습니다. 자세한 정보는 범주 를 참조하십시오.	2017년 10월 3일
NAT 게이트웨이에 태그 지정 지원	2016-11-15	NAT 게이트웨이에 태그를 지정할 수 있습니다. 자세한 정보는 NAT 게이트웨이 태그 지정 (p. 220) 단원을 참조하십시오.	2017년 9월 7일
NAT 게이트웨이의 Amazon CloudWatch 지표	2016-11-15	NAT 게이트웨이에 대한 CloudWatch 지표를 볼 수 있습니다. 자세한 정보는 Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링 (p. 221) 단원을 참조하십시오.	2017년 9월 7일

기능	API 버전	설명	릴리스 날짜
보안 그룹 규칙 설명	2016-11-15	보안 그룹 규칙에 설명을 추가할 수 있습니다. 자세한 정보는 보안 그룹 규칙 (p. 125) 단원을 참조하십시오.	2017년 8월 31일
VPC의 보조 IPv4 CIDR 블록	2016-11-15	VPC에 여러 개의 IPv4 CIDR 블록을 추가할 수 있습니다. 자세한 정보는 VPC에 IPv4 CIDR 블록 추가 (p. 81) 단원을 참조하십시오.	2017년 8월 29일
DynamoDB에 대한 VPC 엔드포인트	2016-11-15	VPC에서 VPC 엔드포인트를 사용하여 Amazon DynamoDB를 액세스할 수 있습니다. 자세한 정보는 Amazon DynamoDB에 대한 엔드포인트 (p. 266) 단원을 참조하십시오.	2017년 8월 16일
탄력적 IP 주소 복구	2016-11-15	탄력적 IP 주소를 해제한 경우 복구할 수 있습니다. 자세한 정보는 탄력적 IP 주소 작업 (p. 244) 단원을 참조하십시오.	2017년 8월 11일
기본 VPC 만들기	2016-11-15	기존 기본 VPC를 삭제한 경우 새로운 기본 VPC를 만들 수 있습니다. 자세한 정보는 기본 VPC 만들기 (p. 100) 단원을 참조하십시오.	2017년 7월 27일
VPN 지표	2016-11-15	VPN 연결에 대한 CloudWatch 지표를 볼 수 있습니다. 자세한 정보는 연결 모니터링 을 참조하십시오.	2017년 5월 15일
IPv6 지원	2016-11-15	IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 VPC의 리소스에 할당할 수 있습니다. 자세한 정보는 VPC의 IP 주소 지정 (p. 103) 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 정보는 VPC와 함께 DNS 사용 (p. 239) 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 정보는 Amazon VPC Peering Guide에서 VPC 피어링 연결 수정 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 정보는 Amazon VPC Peering Guide의 Working With Stale Security Groups 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 정보는 Amazon VPC Peering Guide의 ClassicLink로 구성 을 참조하십시오.	2016년 4월 26일

기능	API 버전	설명	릴리스 날짜
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 정보는 NAT 게이트웨이 (p. 211) 단원을 참조하십시오.	2015년 12월 17일
VPN 기능 향상	2015-04-15	이제 VPN 연결의 1단계 및 2단계에서 AES 256비트 암호화 기능, SHA-256 해시 기능, NAT-T 및 추가적인 Diffie-Hellman 그룹이 지원됩니다. 또한 동일한 고객 게이트웨이 디바이스를 사용하는 각 VPN 연결에 대해 동일한 고객 게이트웨이 IP 주소를 사용할 수 있습니다.	2015년 10월 28일
VPC 흐름 로그	2015-04-15	흐름 로그를 생성하여 VPC의 네트워크 인터페이스에서 전송하고 수신하는 IP 트래픽에 대한 정보를 캡처할 수 있습니다. 자세한 정보는 VPC 흐름 로그 (p. 173) 단원을 참조하십시오.	2015년 6월 10일
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 정보는 VPC 엔드포인트 (p. 246) 단원을 참조하십시오.	2015년 5월 11일
ClassicLink	2014-10-01	ClassicLink를 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 정보는 ClassicLink (p. 282) 단원을 참조하십시오.	2015년 1월 7일
프라이빗 호스팅 영역 사용	2014-09-01	Route 53의 프라이빗 호스팅 영역에서 정의한 사용자 지정 DNS 도메인 이름을 사용하여 VPC의 리소스에 액세스할 수 있습니다. 자세한 정보는 프라이빗 호스팅 영역 사용 (p. 242) 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 정보는 서브넷의 퍼블릭 IPv4 주소 지정 속성 수정 (p. 106) 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 정보는 VPC 피어링 (p. 243) 단원을 참조하십시오.	2014년 3월 24일
새로운 EC2 시작 마법사	2013-10-01	재설계된 EC2 시작 마법사에 대한 정보가 추가되었습니다. 자세한 정보는 3단계: VPC에서 인스턴스 시작 (p. 13) 단원을 참조하십시오.	2013년 10월 10일

기능	API 버전	설명	릴리스 날짜
퍼블릭 IP 주소 배정	2013-07-15	VPC에서 시작되는 인스턴스에 대한 새로운 IP 주소 지정 기능 관련 정보가 추가되었습니다. 자세한 정보는 인스턴스 시작 시 퍼블릭 IPv4 주소 배정 (p. 107) 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 정보는 VPC와 함께 DNS 사용 (p. 239) 단원을 참조하십시오.	2013년 3월 11일
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 가능합니다.	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 있는 모든 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