VPC 피어링

Amazon Virtual Private Cloud

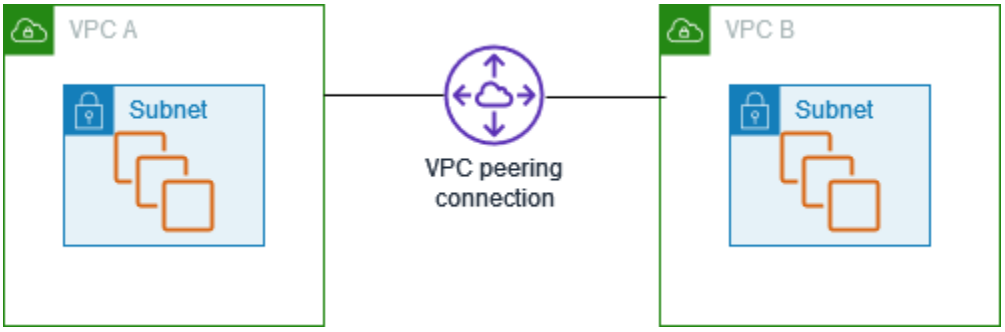
Amazon Virtual Private Cloud: VPC 피어링

Table of Contents

VPC 피어링이란?

Virtual Private Cloud(VPC)는 사용자의 AWS 계정 전용 가상 네트워크입니다. VPC는 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 분리되어 있습니다. AWS 리소스(예: Amazon EC2 인스턴스)를 VPC에서 시작할 수 있습니다.

VPC 피어링 연결은 프라이빗 IPv4 주소 또는 IPv6 주소를 사용하여 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다. 사용자의 자체 VPC 또는 다른 AWS 계정의 VPC와 VPC 피어링 연결을 만들 수 있습니다. VPC는 상이한 리전에 있을 수 있습니다(리전 간 VPC 피어링 연결이라고도 함).



AWS는 VPC의 기존 인프라를 사용하여 VPC 피어링 연결을 생성합니다. 이는 게이트웨이도, VPN 연결도 아니며 물리적 하드웨어 각각에 의존하지 않습니다. 그러므로 통신 또는 대역폭 병목에 대한 단일 지점 장애가 없습니다.

VPC 피어링 연결은 원활한 데이터 전송에 도움이 됩니다. 예를 들어, AWS 계정이 두 개 이상인 경우 이들 계정을 대상으로 VPC를 피어링하여 파일 공유 네트워크를 만들 수 있습니다. VPC 피어링 연결을 사용하여 다른 VPC가 사용자의 VPC 중 하나에 있는 리소스에 액세스하도록 허용할 수도 있습니다.

서로 다른 AWS 리전에 위치한 VPC 사이에 피어링 관계를 설정하는 경우 상이한 AWS 리전의 VPC에 있는 리소스(예: EC2 인스턴스와 Lambda 함수)에서 게이트웨이, VPN 연결 또는 네트워크 어플라이언스를 사용하지 않고 프라이빗 IP 주소를 사용하여 서로 통신할 수 있습니다. 트래픽은 프라이빗 IP 주소 공간 안에서 유지됩니다. 모든 리전 간 트래픽은 암호화되며 단일 장애 지점 또는 대역폭 제한이 없습니다. 트래픽은 항상 글로벌 AWS 백본에서만 유지되고 절대로 퍼블릭 인터넷을 통과하지 않으므로 일반적인 취약점 공격과 DDoS 공격 같은 위협이 감소합니다. 리전 간 VPC 피어링은 리전 간에 리소스를 공유하거나 지리적 중복성을 위해 데이터를 복제할 수 있는 간단하고 비용 효율적인 방법을 제공합니다.

VPC 피어링 연결 요금

VPC 피어링 연결 생성에는 요금이 부과되지 않습니다. 가용 영역 내에 있는 VPC 피어링 연결을 통한 데이터 전송은 서로 다른 계정 간이라도 모두 무료입니다. 가용 영역 및 리전 간 VPC 피어링 연결을 통한 데이터 전송에는 요금이 부과됩니다. 자세한 내용은 Amazon EC2 요금을 참조하세요.

VPC 피어링 연결의 작동 방식

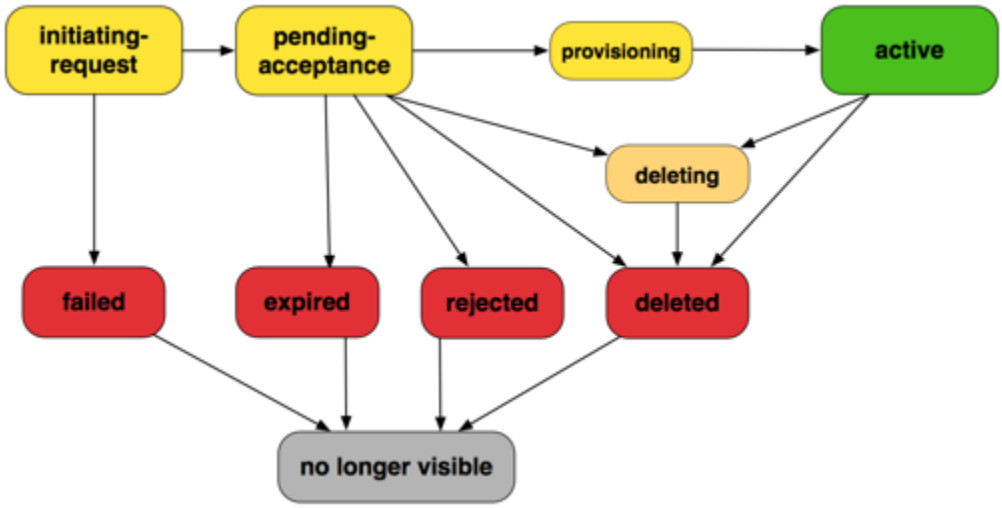
다음 단계에서는 VPC 피터링 프로세스에 대해 설명합니다.

1. 요청자 VPC의 소유자가 수락자 VPC의 소유자에게 VPC 피어링 연결을 생성하도록 요청을 보냅니다. 수락자 VPC는 사용자 또는 다른 AWS 계정에서 소유할 수 있으며, 요청자 VPC의 CIDR 블록과 중첩되는 CIDR 블록은 사용할 수 없습니다.
2. 수락자 VPC의 소유자가 VPC 피어링 연결 요청을 수락하여 VPC 피어링 연결을 활성화합니다.
3. 프라이빗 IP 주소를 사용하여 VPC 간의 트래픽 흐름을 활성화하려면 VPC 피어링 연결 내 각 VPC의 소유자가 다른 VPC(피어 VPC)의 IP 주소 범위를 가리키는 경로를 하나 이상의 VPC 라우팅 테이블에 수동으로 추가해야 합니다.
4. 필요한 경우 피어 VPC에서 주고 받는 트래픽이 제한되지 않도록 EC2 인스턴스와 연결되어 있는 보안 그룹을 업데이트합니다. 두 VPC가 동일한 리전에 있는 경우 보안 그룹의 인바운드 또는 아웃바운드 규칙에 대한 소스나 대상으로 피어 VPC의 보안 그룹을 참조할 수 있습니다.
5. 기본 VPC 피어링 연결 옵션에서, 어느 한 쪽에 위치하는 EC2 인스턴스가 퍼블릭 DNS 호스트 이름을 사용하여 상대방을 참조하는 경우 호스트 이름은 EC2 인스턴스의 퍼블릭 IP 주소로 확인됩니다. 이 동작을 변경하려면 VPC 연결에 대해 DNS 호스트 이름 확인을 활성화합니다. DNS 호스트 이름 확인을 활성화한 후, VPC 피어링 연결의 어느 한 쪽에 위치하는 EC2 인스턴스가 퍼블릭 DNS 호스트 이름을 사용하여 상대방을 참조하는 경우 호스트 이름은 EC2 인스턴스의 프라이빗 IP 주소로 확인됩니다.

자세한 내용은 VPC 피어링 연결 단원을 참조하세요.

VPC 피어링 연결 수명 주기

VPC 피어링 연결은 요청이 시작될 때부터 시작하는 다양한 단계를 거칩니다. 각 단계에는 취사선택할 수 있는 몇 가지 작업이 있으며, 수명 주기가 끝날 때 VPC 피어링 연결은 Amazon VPC 콘솔과 API 또는 명령줄 출력에 일정 시간 동안 표시됩니다.



- Initiating-request: VPC 피어링 연결 요청이 시작되었습니다. 이 단계에서는 피어링 연결이 실패하거나 `pending-acceptance`로 이동할 수 있습니다.
- Failed: VPC 피어링 연결 요청이 실패했습니다. 이 상태에서는 수락, 거부 또는 삭제할 수 없습니다. 실패한 VPC 피어링 연결은 2시간 동안 요청자에게 보이는 상태로 남습니다.
- Pending-acceptance: VPC 피어링 연결 요청이 수락자 VPC 소유자의 수락을 기다리고 있습니다. 이 상태에서 요청자 VPC의 소유자가 요청을 삭제할 수 있고 수락자 VPC의 소유자는 요청을 수락하거나 거부할 수 있습니다. 요청에 대해 아무런 조치도 취하지 않으면 요청이 7일 후에 만료됩니다.
- Expired: VPC 피어링 연결 요청이 만료되었으며, 어느 한 VPC 소유자가 만료된 요청에 대해 어떤 조치도 취할 수 없습니다. 만료된 VPC 피어링 연결은 2일 동안 양쪽 VPC 소유자에게 모두 보이는 상태로 남습니다.
- Rejected: 수락자 VPC의 소유자가 `pending-acceptance` VPC 피어링 연결 요청을 거부했습니다. 이 상태에서는 요청을 수락할 수 없습니다. 거부된 VPC 피어링 연결은 2일 동안 요청자 VPC의 소유자에게 표시되고, 수락자 VPC의 소유자에게는 2시간 동안 표시됩니다. 동일한 AWS 계정 내에서 요청이 생성되었다면, 거부된 요청은 2시간 동안 보이는 상태로 남습니다.
- Provisioning: VPC 피어링 연결 요청이 수락되었으며, 곧 `active` 상태가 됩니다.
- Active: VPC 피어링 연결이 활성화되고, VPC 사이에서 트래픽이 전달될 수 있습니다(보안 그룹 및 라우팅 테이블에서 트래픽 전달을 허용). 이 상태에서는 VPC 소유자 중 한 명의 VPC 피어링 연결을 삭제할 수 있지만 거부할 수는 없습니다.

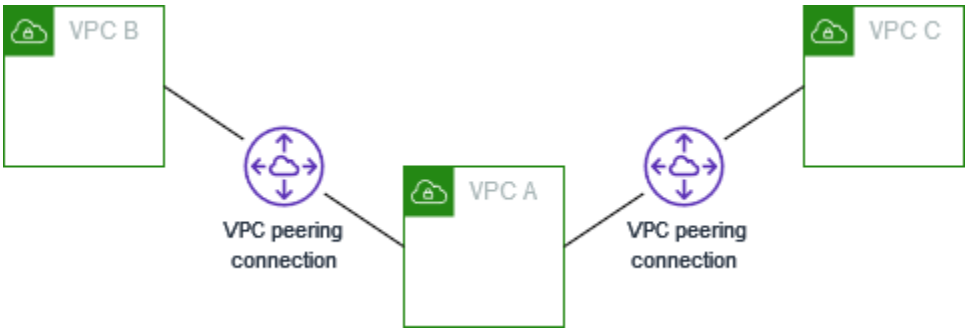
> **Note**
>
> VPC가 상주하는 리전에서 발생하는 이벤트로 인해 트래픽 전달을 막는 경우 VPC 피어링 연결 상태는 그대로 `Active`를 유지합니다.

- Deleting(삭제 중): 삭제가 진행 중인 리전 간 VPC 피어링 연결에 적용됩니다. 한 쪽 VPC 소유자가 `active` 상태인 VPC 피어링 연결 삭제 요청을 제출했거나 요청자 VPC의 소유자가 `pending-acceptance` 상태인 VPC 피어링 연결 요청을 삭제하는 요청을 제출했습니다.
- Deleted: 어느 한 VPC 소유자가 `active` VPC 피어링 연결을 삭제했거나, 요청자 VPC의 소유자가 `pending-acceptance` VPC 피어링 연결 요청을 삭제했습니다. 이 상태에서는 VPC 피어링 연결을 허용하거나 거부할 수 없습니다. VPC 피어링 연결은 2시간 동안 해당 연결을 삭제한 쪽에 보이는 상태로 남고, 다른 쪽에는 2일 동안 보이는 상태로 남습니다. 동일한 AWS 계정 내에서 VPC 피어링 연결이 생성되었다면, 삭제된 요청은 2시간 동안 보이는 상태로 남습니다.

여러 VPC 피어링 연결

VPC 피어링 연결은 두 VPC 간에 일대일 관계로 수행됩니다. 소유하고 있는 각 VPC에 대한 VPC 피어링 연결을 여러 개 생성할 수 있지만, 전이성 피어링 관계는 지원되지 않으므로, VPC가 직접 피어링되지 않은 VPC와의 피어링 관계는 없습니다.

다음 다이어그램은 한 VPC가 다른 두 VPC에 피어링되는 예를 보여줍니다. 두 개의 VPC 피어링 연결이 있는데, VPC A는 VPC B 및 VPC C와 모두 피어링되고 VPC B와 VPC C는 피어링되지 않습니다. VPC B와 VPC C 사이의 피어링을 위해 VPC A를 중계점으로 사용할 수 없습니다. VPC B와 VPC C 간의 트래픽 라우팅을 활성화하려면 둘 사이에 고유한 VPC 피어링 연결을 생성해야 합니다.



VPC 피어링 제한

VPC 피어링 연결에 대한 다음과 같은 제한 사항을 고려하세요. 경우에 따라 VPC 피어링 연결 대신에 Transit Gateway Attachment를 사용할 수 있습니다. 자세한 내용은 Amazon VPC Transit Gateway의 전송 게이트웨이 시나리오 예를 참조하세요.

연결

- VPC당 활성 및 보류 중인 VPC 피어링 연결 수에는 할당량이 있습니다. 자세한 내용은 할당량 단원을 참조하세요.

- VPC 2개 사이에 2개 이상의 VPC 피어링 연결을 동시에 보유할 수 없습니다.
- VPC 피어링 연결에 대해 생성하는 모든 태그는 태그를 생성하는 계정 또는 리전에서만 적용됩니다.
- 피어 VPC에서는 Amazon DNS 서버에 연결하거나 쿼리할 수 없습니다.
- VPC 피어링 연결에서 VPC의 IPv4 CIDR 블록이 RFC 1918에 의해 지정된 프라이빗 IPv4 주소 범위를 벗어나는 경우, 해당 VPC의 프라이빗 DNS 호스트 이름을 프라이빗 IP 주소로 확인할 수 없습니다. 프라이빗 DNS 호스트 이름을 프라이빗 IP 주소로 확인하려면 VPC 피어링 연결에 대한 DNS 확인 지원을 활성화할 수 있습니다. 자세한 내용은 VPC 피어링 연결에 대한 DNS 확인 활성화 단원을 참조하세요.
- IPv6를 통해 통신하도록 VPC 피어링 연결의 한쪽에 있는 리소스를 활성화할 수 있습니다. IPv6 CIDR 블록을 각 VPC와 연결하고, IPv6 통신을 위해 VPC의 인스턴스를 활성화하고, 피어 VPC를 위한 IPv6 트래픽을 VPC 피어링 연결로 라우팅해야 합니다.
- VPC 피어링 연결에서의 유니캐스트 역경로 전송은 지원되지 않습니다. 자세한 내용은 응답 트래픽을 위한 라우팅 단원을 참조하세요.

겹치는 CIDR 블록

- 일치하거나 중첩되는 IPv4 또는 IPv6 CIDR 블록이 있는 VPC 간에는 VCP 피어링 연결을 생성할 수 없습니다.
- 여러 개의 IPv4 CIDR 블록이 있는 경우, 겹치지 않는 CIDR 블록 또는 IPv6 CIDR 블록만 사용할 의도이더라도 CIDR 블록이 겹치면 VPC 피어링 연결을 생성할 수 없습니다.

전이적 피어링

- VPC 피어링은 전이적 피어링 관계를 지원하지 않습니다. 예를 들어 VPC A와 VPC B 및 VPC A와 VPC C 사이에 VPC 피어링 연결이 있으면 VPC A를 통해 VPC B에서 VPC C로 트래픽을 라우팅할 수 없습니다. VPC B와 VPC C 사이에 트래픽을 라우팅하려면 둘 사이에 VPC 피어링 연결을 생성해야 합니다. 자세한 내용은 함께 피어링된 세 개의 VPC 단원을 참조하세요.

게이트웨이 또는 프라이빗 연결을 통한 엣지 간 라우팅

- VPC A에 인터넷 게이트웨이가 있으면 VPC B의 리소스에서는 VPC A의 인터넷 게이트웨이를 사용하여 인터넷에 액세스할 수 없습니다.
- VPC A의 서브넷 대한 인터넷 액세스 권한을 제공하는 NAT 디바이스가 VPC A에 있으면 VPC B의 리소스에서는 VPC A의 NAT 디바이스를 사용하여 인터넷에 액세스할 수 없습니다.

- 회사 네트워크에 대한 VPN 연결이 VPC A에 있으면 VPC B의 리소스에서는 VPN 연결을 사용하여 회사 네트워크와 통신할 수 없습니다.
- 회사 네트워크에 대한 AWS Direct Connect 연결이 VPC A에 있으면 VPC B의 리소스에서는 AWS Direct Connect연결을 사용하여 회사 네트워크와 통신할 수 없습니다.
- Amazon S3 대한 연결을 VPC A의 프라이빗 서브넷에 제공하는 게이트웨이 엔드포인트가 VPC A에 있으면 VPC B의 리소스에서는 게이트웨이 엔드포인트를 사용하여 Amazon S3에 액세스할 수 없습니다.

리전 간 VPC 피어링 연결

- 점보 프레임의 경우 동일한 리전 내의 VPC 피어링 연결 간 최대 전송 단위(MTU)는 9,001바이트입니다. 리전 간 VPC 피어링 연결의 MTU는 8,500바이트입니다. 점보 프레임에 대한 자세한 내용은 Amazon EC2 사용 설명서의 점보 프레임(9001 MTU)을 참조하세요.
- VPC에 대한 IPv4 CIDR이 RFC 1918에서 지정한 프라이빗 IPv4 주소 범위에 포함되더라도, VPC 피어링 연결에 대해 DNS 확인 지원을 활성화하여 피어링된 VPC의 프라이빗 DNS 호스트 이름을 프라이빗 IP 주소로 확인해야 합니다.

공유 VPC 및 서브넷

- VPC 소유자만 피어링 연결을 통한 작업(설명, 생성, 수락, 거부, 수정 또는 삭제)을 수행할 수 있습니다. 참가자는 피어링 연결을 통한 작업을 수행할 수 없습니다. 자세한 내용은 Amazon VPC 사용 설명서의 다른 계정과 VPC 공유하기를 참조하세요.

VPC 피어링 연결

VPC 피어링을 사용하면 동일하거나 상이한 AWS 리전에 있는 두 VPC를 연결할 수 있습니다. 이렇게 하면 한 VPC의 인스턴스가 다른 VPC의 인스턴스와 모두 동일한 네트워크에 속한 것처럼 통신할 수 있습니다.

VPC 피어링은 프라이빗 IPv4 주소 또는 IPv6 주소를 사용하여 두 VPC 간에 직접 네트워크 라우팅을 생성합니다. 연결된 VPC 간에 전송되는 트래픽은 인터넷, VPN 연결 또는 AWS Direct Connect 연결을 통과하지 않습니다. 따라서 VPC 피어링은 VPC 경계 간에 데이터베이스 또는 웹 서버와 같은 리소스를 공유하는 안전한 방법입니다.

VPC 피어링 연결을 설정하려면 사용자가 한 VPC에서 피어링 연결 요청을 생성하고, 다른 VPC의 소유자가 그 요청을 수락해야 합니다. 연결이 설정되고 나면 라우팅 테이블을 업데이트하여 VPC 간에 트래픽을 라우팅할 수 있습니다. 이렇게 하면 한 VPC의 인스턴스가 다른 VPC의 리소스에 액세스할 수 있습니다.

VPC 피어링은 AWS에서 조직의 경계를 넘어 다중 VPC 아키텍처를 구축하고 리소스를 공유하는 데 중요한 도구입니다. VPN이나 기타 네트워킹 서비스를 복잡하게 구성하지 않고도 간단하고 지연 시간이 짧은 방법으로 VPC를 연결할 수 있습니다.

다음 절차를 사용하여 VPC 피어링 연결을 생성하고 연동할 수 있습니다.

업무

- VPC 피어링 연결 생성
- VPC 피어링 연결 수락 또는 거부
- VPC 피어링 연결을 위한 라우팅 테이블 업데이트
- 피어 보안 그룹을 참조하도록 보안 그룹 업데이트
- VPC 피어링 연결에 대한 DNS 확인 활성화
- VPC 피어링 연결 삭제
- VPC 피어링 연결 문제 해결

VPC 피어링 연결 생성

VPC 피어링 연결을 생성하려면, 우선 다른 VPC와 피어링하기 위한 요청을 생성합니다. 요청을 활성화하려면 수락자 VPC의 소유자가 요청을 수락해야 합니다. 다음과 같은 피어링 연결이 지원됩니다.

- 동일한 계정 및 리전의 VPC 간
- 동일한 계정의 다른 리전에 있는 VPC 간
- 다른 계정의 동일한 리전에 있는 VPC 간
- 다른 계정의 다른 리전에 있는 VPC 간

리전 간 VPC 피어링 연결의 경우 요청자 VPC의 리전에서 요청을 수행해야 하며 수락자 VPC의 리전에서 요청이 수락되어야 합니다. 자세한 내용은 the section called “수락 또는 거부” 섹션을 참조하세요.

업무

- 사전 조건
- 콘솔을 사용하여 피어링 연결 생성
- 명령줄을 사용하여 피어링 연결 생성

사전 조건

- VPC 피어링 연결에 대한 제한 사항을 검토합니다.
- VPC에 겹치는 IPv4 CIDR 블록이 없는지 확인합니다. 겹치는 경우 VPC 피어링 연결의 상태가 즉시 `failed`로 바뀝니다. 이 제한 사항은 VPC에 고유한 IPv6 CIDR 블록이 있는 경우에도 적용됩니다.

콘솔을 사용하여 피어링 연결 생성

다음 절차에 따라 VPC 피어링 연결을 생성합니다.

콘솔을 사용하여 피어링 연결을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Peering connections를 선택합니다.
3. Create peering connection(피어링 연결 생성)을 선택합니다.
4. (선택 사항) 이름에 VPC 피어링 연결의 이름을 지정합니다. 그러면 Name 키와 지정한 값으로 태그가 생성됩니다.
5. VPC ID(요청자)에서 현재 계정의 VPC를 선택합니다.
6. 피어링할 다른 VPC 선택에서 다음을 수행합니다.
    a. 다른 계정의 VPC와 피어링하려면 계정에서 다른 계정을 선택하고 계정 ID를 입력합니다. 그렇지 않으면 내 계정이 선택된 상태로 둡니다.

b. 다른 리전의 VPC와 피어링하려면 리전에서 다른 리전을 선택하고 리전을 선택합니다. 그렇지 않으면 이 리전이 선택된 상태로 둡니다.

c. VPC ID(수락자)에서 지정된 계정 및 리전의 VPC를 선택합니다.

7. (선택 사항) 태그를 추가하려면 새 태그 추가(Add new tag)를 선택하고 태그 키와 태그 값을 입력합니다.
8. Create peering connection(피어링 연결 생성)을 선택합니다.
9. 수락자 계정의 소유자가 피어링 연결을 수락해야 합니다. 자세한 내용은 the section called “수락 또는 거부” 섹션을 참조하세요.
10. 두 VPC 간의 연결이 활성화되도록 두 VPC의 라우팅 테이블을 업데이트합니다. 자세한 내용은 the section called “라우팅 테이블 업데이트” 섹션을 참조하세요.

명령줄을 사용하여 피어링 연결 생성

다음과 같은 명령을 사용하여 VPC 피어링 연결을 생성할 수 있습니다.

- create-vpc-peering-connection(AWS CLI)
- New-EC2VpcPeeringConnection(AWS Tools for Windows PowerShell)

VPC 피어링 연결 수락 또는 거부

`pending-acceptance` 상태의 VPC 피어링 연결은 사용 설정할 수락자 VPC의 소유자가 수락해야 합니다. `Deleted` 피어링 연결 상태에 대한 자세한 내용은 VPC 피어링 연결 수명 주기를 참조하세요. 다른 AWS 계정으로 보낸 VPC 피어링 연결 요청은 수락할 수 없습니다. 같은 AWS 계정의 VPC 간에 VPC 피어링 연결을 생성하려는 경우, 직접 요청을 생성하고 수락할 수 있습니다.

`pending-acceptance` 상태로 받은 VPC 피어링 연결 요청을 거부할 수 있습니다. 알고 있고 신뢰하는 AWS 계정으로부터의 VPC 피어링 연결만 허용해야 합니다. 불필요한 요청은 거부할 수 있습니다. `Rejected` 피어링 연결 상태에 대한 자세한 내용은 VPC 피어링 연결 수명 주기를 참조하세요.

> **Important**
>
> 알 수 없는 AWS 계정에서의 VPC 피어링 연결은 수락하지 마세요. 악의적 사용자가 VPC에 대한 무단 네트워크 액세스를 위해 VPC 피어링 연결 요청을 보냈을 수도 있기 때문입니다. 이런 수법은 피어 피싱으로 알려져 있습니다. 요청자가 AWS 계정 또는 VPC에 대한 정보에 액세스할 위험 없이 원치 않는 VPC 피어링 연결 요청을 안전하게 거부할 수 있습니다. 자세한 내용은

VPC 피어링 연결 수락 또는 거부 단원을 참조하세요. 요청을 무시하고 그냥 만료되도록 할 수도 있습니다. 기본적으로, 요청은 7일 후에 만료됩니다.

콘솔을 사용하여 피어링 연결을 수락 또는 거부하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 리전 선택기를 사용하여 수락자 VPC의 리전을 선택합니다.
3. 탐색 창에서 Peering Connections(피어링 연결)를 선택합니다.
4. 피어링 연결을 거부하려면 VPC 피어링 연결을 선택한 다음 작업, 요청 거부를 선택합니다. 확인 메시지가 나타나면 요청 거부를 선택합니다.
5. 피어링 연결을 수락하려면 보류 중인 VPC 피어링 연결(상태: `pending-acceptance`)을 선택한 다음 작업, 요청 수락을 선택합니다. 피어링 연결 수명 주기 상태에 대한 자세한 내용은 VPC 피어링 연결 수명 주기를 참조하세요.

   보류 중인 VPC 피어링 연결이 없는 경우 수락자 VPC의 리전을 선택했는지 확인합니다.
6. 확인 메시지가 나타나면 요청 수락을 선택합니다.
7. 피어링 연결을 통해 트래픽을 보내고 받을 수 있도록 VPC 라우팅 테이블에 경로를 추가하려면 지금 내 라우팅 테이블 수정을 선택합니다. 자세한 내용은 VPC 피어링 연결을 위한 라우팅 테이블 업데이트 섹션을 참조하세요.

명령줄을 사용하여 피어링 연결을 수락하려면

- accept-vpc-peering-connection(AWS CLI)
- Approve-EC2VpcPeeringConnection(AWS Tools for Windows PowerShell)

명령줄을 사용하여 피어링 연결을 거부하려면

- reject-vpc-peering-connection(AWS CLI)
- Deny-EC2VpcPeeringConnection(AWS Tools for Windows PowerShell)

VPC 피어링 연결을 위한 라우팅 테이블 업데이트

피어링된 VPC의 인스턴스 간에 프라이빗 IPv4 트래픽을 활성화하려면 두 인스턴스의 서브넷과 연결된 라우팅 테이블에 경로를 추가해야 합니다. 이 경로 대상은 피어 VPC의 CIDR 블록(또는 CIDR 블록

의 일부)이고 대상은 VPC 피어링 연결의 ID입니다. 자세한 내용은 Amazon VPC 사용 설명서의 라우팅 테이블 구성을 참조하세요.

다음은 피어링된 두 VPC(VPC A 및 VPC B)의 인스턴스 간 통신을 가능하게 하는 라우팅 테이블의 예입니다. 각 테이블에는 로컬 경로와 피어 VPC에 대한 트래픽을 VPC 피어링 연결로 보내는 경로가 있습니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-*11112222*
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-*11112222*

이와 마찬가지로 VPC 피어링 연결의 VPC에 연결된 IPv6 CIDR 블록이 있는 경우, IPv6를 통해 피어 VPC와 통신할 수 있게 해주는 경로를 추가할 수 있습니다.

VPC 피어링 연결을 위해 지원되는 라우팅 테이블 구성에 대한 자세한 내용은 일반 VPC 피어링 연결 구성을 참조하세요.

고려 사항

- 겹치거나 일치하는 IPv4 CIDR 블록을 가진 여러 VPC와 피어링된 VPC가 있는 경우, 라우팅 테이블이 자신의 VPC에서 잘못된 VPC로 응답 트래픽을 보내지 못하도록 구성되어 있는지 확인하세요. AWS에서는 현재 패킷의 원본 IP를 확인하고 응답 패킷을 다시 원본으로 라우팅하는 VPC 피어링 연결에서 유니캐스트 역경로 전달을 지원하지 않습니다. 자세한 내용은 응답 트래픽을 위한 라우팅 단원을 참조하세요.
- 계정에는 라우팅 테이블당 추가할 수 있는 항목 수에 대한 할당량이 있습니다. VPC의 VPC 피어링 연결 수가 하나의 라우팅 테이블의 라우팅 테이블 항목 할당량을 초과하는 경우, 사용자 지정 라우팅 테이블과 각각 연결된 여러 서브넷을 사용하는 것도 좋습니다.
- `pending-acceptance` 상태에 있는 VPC 피어링 연결을 위한 경로를 추가할 수 있습니다. 하지만, 이 경로는 `blackhole`의 상태를 가지고 VPC 피어링 연결이 `active` 상태가 될 때까지 아무런 효과도 미치지 않습니다.

VPC 피어링 연결을 위한 IPv4 경로를 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 인스턴스가 상주하는 서브넷과 연결된 라우팅 테이블 옆의 확인란을 선택합니다.

   서브넷을 특정 라우팅 테이블에 명시적으로 연결하지 않을 경우 VPC의 기본 라우팅 테이블이 서브넷과 암시적으로 연결됩니다.
4. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
5. 라우팅 추가를 선택합니다.
6. Destination에 VPC 피어링 연결에서 네트워크 트래픽을 전달해야 할 IPv4 주소 범위를 입력합니다. 피어 VPC의 전체 IPv4 CIDR 블록, 특정 범위 또는 개별 IPv4 주소(예: 통신할 인스턴스의 IP 주소)를 지정할 수 있습니다. 예를 들어 피어 VPC의 CIDR 블록이 `10.0.0.0/16`인 경우 `10.0.0.0/24` 부분이나 특정 IP 주소 `10.0.0.7/32`를 지정할 수 있습니다.
7. 대상에서 VPC 피어링 연결을 선택합니다.
8. 변경 사항 저장을 선택합니다.

또한 피어 VPC의 소유자는 이러한 단계를 완료하여 VPC 피어링 연결을 통해 VPC로 트래픽을 다시 보내도록 라우팅을 추가해야 합니다.

IPv6 주소를 사용하는 여러 AWS 리전에 리소스가 있는 경우 리전 간 피어링 연결을 생성할 수 있습니다. 그런 다음 리소스 간의 통신을 위해 IPv6 라우팅을 추가할 수 있습니다.

VPC 피어링 연결을 위한 IPv6 라우팅을 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 인스턴스가 상주하는 서브넷과 연결된 라우팅 테이블 옆의 확인란을 선택합니다.

   > **Note**
   >
   > 그 서브넷과 연결된 라우팅 테이블이 없는 경우, 서브넷이 기본적으로 이 라우팅 테이블을 사용하므로 VPC에 대한 기본 라우팅 테이블을 선택합니다.
4. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
5. 라우팅 추가를 선택합니다.

6. Destination에 피어 VPC의 IPv6 주소 범위를 입력합니다. 피어 VPC의 전체 IPv6 CIDR 블록, 특정 범위 또는 개별 IPv6 주소를 지정합니다. 예를 들어 피어 VPC의 CIDR 블록이 `2001:db8:1234:1a00::/56`인 경우 `2001:db8:1234:1a00::/64` 부분이나 특정 IP 주소 `2001:db8:1234:1a00::123/128`를 지정할 수 있습니다.
7. 대상에서 VPC 피어링 연결을 선택합니다.
8. 변경 사항 저장을 선택합니다.

자세한 내용은 Amazon VPC 사용 설명서의 라우팅 테이블을 참조하세요.

명령줄을 사용하여 경로를 추가하거나 바꾸려면

- create-route 및 replace-route(AWS CLI)
- New-EC2Route 및 Set-EC2Route(AWS Tools for Windows PowerShell)

피어 보안 그룹을 참조하도록 보안 그룹 업데이트

피어링된 VPC의 보안 그룹을 참조하도록 VPC 보안 그룹의 인바운드 또는 아웃바운드 규칙을 업데이트할 수 있습니다. 그렇게 하면 피어링된 VPC의 참조 보안 그룹과 연결된 인스턴스 간에 트래픽을 주고받을 수 있습니다.

> **Note**
>
> 피어 VPC의 보안 그룹은 콘솔에 표시되지 않으므로 선택할 수 없습니다.

요구 사항

- 피어 VPC의 보안 그룹을 참조하려면 VPC 피어링 연결이 `active` 상태여야 합니다.
- 피어 VPC는 사용자 계정의 VPC이거나 다른 AWS 계정의 VPC일 수 있습니다. 다른 AWS 리전에 있지만 리전이 동일한 보안 그룹을 참조하려면 계정 번호를 보안 그룹의 ID와 함께 포함합니다. 예를 들어 `123456789012/sg-1a2b3c4d`입니다.
- 다른 리전에 있는 피어 VPC의 보안 그룹을 참조할 수 없습니다. 그 대신 피어 VPC의 CIDR 블록을 사용하세요.
- 미들박스 어플라이언스를 통해 서로 다른 서브넷에 있는 두 인스턴스 간의 트래픽을 전달하도록 경로를 구성하는 경우 두 인스턴스에 대한 보안 그룹이 인스턴스 간에 트래픽이 흐르도록 허용해야 합니다. 각 인스턴스의 보안 그룹은 다른 인스턴스의 프라이빗 IP 주소 또는 다른 인스턴스가 포함된

서브넷의 CIDR 범위를 소스로 참조해야 합니다. 다른 인스턴스의 보안 그룹을 소스로 참조하면 인스턴스 간에 트래픽이 흐를 수 없습니다.

콘솔을 사용하여 보안 그룹 규칙을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹을 선택하고 다음 중 하나를 수행합니다.
   - 인바운드 규칙을 수정하려면 작업, 인바운드 규칙 편집을 선택합니다.
   - 아웃바운드 규칙을 수정하려면 작업, 아웃바운드 규칙 편집을 선택합니다.
4. 규칙을 추가하려면 규칙 추가를 선택한 다음 유형, 프로토콜 및 포트 범위를 지정합니다. 소스(인바운드 규칙) 또는 대상(아웃바운드 규칙)의 경우 다음 중 하나를 수행합니다.
   - 계정과 리전이 동일한 피어 VPC의 경우 보안 그룹의 ID를 입력합니다.
   - 계정은 다르지만 리전은 동일한 피어 VPC의 경우 계정 ID와 보안 그룹 ID를 슬래시로 구분하여 입력합니다(예: `123456789012/sg-1a2b3c4d`).
   - 리전이 다른 피어 VPC의 경우 피어 VPC의 CIDR 블록을 입력합니다.
5. 기존 규칙을 편집하려면 해당 값(예: 소스 또는 설명)을 변경합니다.
6. 규칙을 삭제하려면 규칙 옆의 삭제를 선택합니다.
7. 규칙 저장을 선택합니다.

명령줄을 사용하여 인바운드 규칙을 업데이트하려면

- authorize-security-group-ingress 및 revoke-security-group-ingress(AWS CLI)
- Grant-EC2SecurityGroupIngress 및 Revoke-EC2SecurityGroupIngress(AWS Tools for Windows PowerShell)

예를 들어, 피어 VPC의 `sg-bbbb2222`에서 HTTP를 통해 인바운드 액세스를 허용하도록 보안 그룹 `sg-aaaa1111`을 업데이트하려면 다음과 같은 명령을 사용합니다. 피어 VPC의 리전은 동일하지만 계정은 다른 경우 `--group-owner` *`aws-account-id`*를 추가합니다.

```
aws ec2 authorize-security-group-ingress --group-id sg-aaaa1111 --protocol tcp --
port 80 --source-group sg-bbbb2222
```

명령줄을 사용하여 아웃바운드 규칙을 업데이트하려면

- authorize-security-group-egress 및 revoke-security-group-egress(AWS CLI)
- Grant-EC2SecurityGroupEgress 및 Revoke-EC2SecurityGroupEgress(AWS Tools for Windows PowerShell)

보안 그룹 규칙을 업데이트한 후 describe-security-groups 명령을 사용하여 보안 그룹 규칙에서 참조된 보안 그룹을 볼 수 있습니다.

참조된 보안 그룹 식별

보안 그룹이 피어 VPC의 보안 그룹 규칙에서 참조되고 있는지 여부를 확인하려면 계정의 하나 이상의 보안 그룹에 대해 다음 명령 중 하나를 사용할 수 있습니다.

- describe-security-group-references(AWS CLI)
- Get-EC2SecurityGroupReference(AWS Tools for Windows PowerShell)

다음 예의 응답은 보안 그룹 sg-bbbb2222가 VPC 의 보안 그룹 vpc-aaaaaaaa에서 참조되고 있음을 나타냅니다.

```
aws ec2 describe-security-group-references --group-id sg-bbbb2222
```

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa",
      "GroupId": "sg-bbbb2222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

VPC 피어링 연결을 삭제하거나 피어 VPC의 소유자가 참조된 보안 그룹을 삭제하는 경우 보안 그룹 규칙은 무효가 됩니다.

무효 보안 그룹 규칙으로 보기 및 삭제

무효 보안 그룹 규칙이란 동일한 VPC 또는 피어 VPC에서 삭제된 보안 그룹을 참조하거나, VPC 피어링 연결이 삭제된 피어 VPC의 보안 그룹을 참조하는 규칙입니다. 보안 그룹 규칙이 무효로 되면, 해당

규칙은 보안 그룹에서 자동으로 제거되지 않습니다. 따라서 규칙을 수동으로 제거해야 합니다. VPC 피어링 연결이 삭제되어 보안 그룹 규칙이 유효하지 않은 경우 동일한 VPC로 새 VPC 피어링 연결을 생성하면 규칙이 더 이상 부실한 것으로 표시되지 않습니다.

Amazon VPC 콘솔을 사용하여 VPC에 대한 무효 보안 그룹 규칙을 보고 삭제할 수 있습니다.

무효 보안 그룹 규칙을 보고 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 작업(Actions), 오래된 규칙 관리(Manage stale rules)를 선택합니다.
4. VPC에서 오래된 규칙이 있는 VPC를 선택합니다.
5. 편집(Edit)을 선택합니다.
6. 삭제할 규칙 옆에 있는 삭제(Delete) 버튼을 선택합니다. 변경 사항 미리 보기(Preview changes), 규칙 저장(Save rules)을 선택합니다.

명령줄을 사용하여 부실한 보안 그룹 규칙을 설명하려면

- describe-stale-security-groups(AWS CLI)
- Get-EC2StaleSecurityGroup(AWS Tools for Windows PowerShell)

다음 예에서는 VPC A (`vpc-aaaaaaaa`)와 VPC B가 피어링되었고, VPC 피어링 연결이 삭제되었습니다. VPC A의 보안 그룹 `sg-aaaa1111`은 VPC B의 `sg-bbbb2222`를 참조합니다. VPC에 대해 `describe-stale-security-groups` 명령을 실행하면, 응답은 보안 그룹 `sg-aaaa1111`에 `sg-bbbb2222`를 참조하는 무효 SSH 규칙이 있음을 나타냅니다.

```
aws ec2 describe-stale-security-groups --vpc-id vpc-aaaaaaaa
```

```
{
    "StaleSecurityGroupSet": [
        {
            "VpcId": "vpc-aaaaaaaa",
            "StaleIpPermissionsEgress": [],
            "GroupName": "Access1",
            "StaleIpPermissions": [
                {
                    "ToPort": 22,
```

```
                    "FromPort": 22,
                    "UserIdGroupPairs": [
                        {
                            "VpcId": "vpc-bbbbbbbb",
                            "PeeringStatus": "deleted",
                            "UserId": "123456789101",
                            "GroupName": "Prod1",
                            "VpcPeeringConnectionId": "pcx-b04deed9",
                            "GroupId": "sg-bbbb2222"
                        }
                    ],
                    "IpProtocol": "tcp"
                }
            ],
            "GroupId": "sg-aaaa1111",
            "Description": "Reference remote SG"
        }
    ]
}
```

무효 보안 그룹 규칙을 식별한 후에는 revoke-security-group-ingress 또는 revoke-security-group-egress 명령을 사용하여 해당 규칙을 삭제할 수 있습니다.

VPC 피어링 연결에 대한 DNS 확인 활성화

VPC 피어링 연결의 DNS 설정은 VPC 피어링 연결을 통과하는 요청의 퍼블릭 DNS 호스트 이름을 확인하는 방법을 결정합니다. VPC 피어링 연결의 한 쪽에 있는 EC2 인스턴스가 인스턴스의 퍼블릭 IPv4 DNS 호스트 이름을 사용하여 다른 쪽에 있는 EC2 인스턴스에 요청을 보내는 경우, DNS 호스트 이름은 다음과 같이 확인됩니다.

DNS 확인 비활성화됨(기본값)

퍼블릭 IPv4 DNS 호스트 이름으로 인스턴스의 퍼블릭 IPv4 주소를 확인합니다.

DNS 확인 활성화됨

퍼블릭 IPv4 DNS 호스트 이름으로 인스턴스의 프라이빗 IPv4 주소를 확인합니다.

요구 사항

- DNS 호스트 이름과 DNS 확인에 대해 두 VPC를 모두 활성화해야 합니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 VPC에 대한 DNS 속성을 참조하세요.

- 피어링 연결은 `active` 상태여야 합니다. 피어링 연결을 생성할 때 DNS 확인을 활성화할 수 없습니다.
- 요청자 VPC의 소유자는 요청자 VPC 피어링 옵션을 수정해야 하며, 수락자 VPC의 소유자는 수락자 VPC 피어링 옵션을 수정해야 합니다. VPC가 동일한 계정과 리전에 있는 경우 요청자 및 수락자 VPC의 DNS 확인을 동시에 활성화할 수 있습니다.

콘솔을 사용하여 피어링 연결에 대한 DNS 확인을 활성화하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Peering connections를 선택합니다.
3. VPC 피어링 연결을 선택합니다.
4. 작업, DNS 설정 편집을 차례로 선택합니다.
5. 요청자 VPC의 요청에 대한 DNS 확인을 활성화하려면 요청자 DNS 확인, 수락자 VPC가 요청자 VPC의 DNS를 확인하도록 허용을 차례로 선택합니다.
6. 수락자 VPC의 요청에 대한 DNS 확인을 활성화하려면 수락자 DNS 확인, 요청자 VPC가 수락자 VPC의 DNS를 확인하도록 허용을 차례로 선택합니다.
7. 변경 사항 저장을 선택합니다.

명령줄을 사용하여 DNS 확인을 활성화하려면

- modify-vpc-peering-connection-options(AWS CLI)
- Edit-EC2VpcPeeringConnectionOption(AWS Tools for Windows PowerShell)

명령줄을 사용하여 VPC 피어링 연결 옵션을 설명하려면

- describe-vpc-peering-connections(AWS CLI)
- Get-EC2VpcPeeringConnection(AWS Tools for Windows PowerShell)

VPC 피어링 연결 삭제

피어링 연결에서 VPC의 어느 한 소유자가 언제든 VPC 피어링 연결을 삭제할 수 있습니다. 자신이 요청하여 아직도 `pending-acceptance` 상태에 있는 VPC 피어링 연결도 삭제할 수 있습니다.

VPC 피어링 연결이 `rejected` 상태이면 VPC 피어링 연결을 삭제할 수 없습니다. 연결은 자동으로 삭제됩니다.

Amazon VPC 콘솔에서 활성 VPC 피어링 연결에 속한 VPC를 삭제하면 해당 VPC 피어링 연결도 삭제됩니다. 또 다른 계정에서 VPC와의 VPC 피어링 연결을 요청했는데 상대방이 그 요청을 수락하기 전에 자신의 VPC를 삭제하면 VPC 피어링 연결도 삭제됩니다. 또 다른 계정의 VPC에서 보낸 `pending-acceptance` 요청이 있는 VPC는 삭제할 수 없습니다. 우선 VPC 피어링 연결 요청을 거부해야 합니다.

피어링 연결을 삭제하면 상태가 `Deleting`으로 설정된 다음 `Deleted`로 설정됩니다. 연결을 삭제한 후에는 수락, 거부 또는 편집할 수 없습니다. 피어링 연결이 표시되는 기간에 대한 자세한 내용은 VPC 피어링 연결 수명 주기를 참조하세요.

VPC 피어링 연결을 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Peering connections를 선택합니다.
3. VPC 피어링 연결을 선택합니다.
4. Actions(작업), Delete peering connection(피어링 연결 삭제)을 선택합니다.
5. 확인 메시지가 나타나면 **`delete`**을 입력한 다음 삭제를 선택합니다.

명령줄을 사용하여 VPC 피어링 연결을 삭제하려면

- delete-vpc-peering-connection(AWS CLI)
- Remove-EC2VpcPeeringConnection(AWS Tools for Windows PowerShell)

VPC 피어링 연결 문제 해결

피어 VPC의 리소스에서 VPC의 리소스에 연결하는 데 문제가 있는 경우 다음 작업을 수행합니다.

- 각 VPC의 각 리소스에 대해 서브넷의 라우팅 테이블에 피어 VPC로 향하는 트래픽을 VPC 피어링 연결로 보내는 경로가 포함되어 있는지 확인합니다. 이렇게 하면 네트워크 트래픽이 두 VPC 간에 올바르게 흐를 수 있습니다. 자세한 내용은 라우팅 테이블 업데이트 단원을 참조하세요.
- 관련된 모든 EC2 인스턴스의 경우, 해당 인스턴스의 보안 그룹이 피어 VPC의 인바운드 및 아웃바운드 트래픽을 허용하는지 확인합니다. 보안 그룹 규칙은 EC2 인스턴스에 액세스할 수 있는 트래픽을 제어합니다. 자세한 내용은 피어 보안 그룹 참조 단원을 참조하세요.
- 리소스가 포함된 서브넷의 네트워크 ACL이 피어 VPC에서 필요한 트래픽을 허용하는지 확인합니다. 네트워크 ACL은 서브넷 수준에서 트래픽을 필터링하는 추가 보안 계층입니다.

문제가 계속되면 Reachability Analyzer를 활용하면 됩니다. Reachability Analyzer는 라우팅 테이블, 보안 그룹, 네트워크 ACL 등 두 VPC 간의 연결 문제를 일으키는 특정 구성 요소를 식별하는 데 도움을 제공합니다. 자세한 내용은 Reachability Analyzer 사용 설명서를 참조하세요.

VPC 네트워킹 구성을 철저히 확인하는 것은 발생할 수 있는 모든 VPC 피어링 연결 문제를 해결하는 데 있어 핵심입니다.

일반 VPC 피어링 연결 구성

이 섹션에서는 구현할 수 있는 두 가지 일반적인 유형의 VPC 피어링 구성에 대해 설명합니다.

- 전체 VPC로 향하는 경로가 포함된 VPC 피어링 구성: 이 구성에서는 각 VPC의 라우팅 테이블에 경로를 생성하여 피어 VPC로 향하는 모든 트래픽을 VPC 피어링 연결로 보냅니다. 이렇게 하면 한 VPC의 모든 리소스가 피어 VPC의 모든 리소스와 통신할 수 있으므로 관리가 간소화됩니다. 그러나 이는 또한 VPC 간의 모든 트래픽이 피어링 연결을 통해 흐르기 때문에 트래픽이 많으면 병목 현상이 발생할 수 있음을 의미합니다.
- 특정 경로를 사용한 VPC 피어링 구성: 또는 각 VPC의 라우팅 테이블에 피어 VPC의 특정 서브넷이나 리소스로만 트래픽을 보내는 세분화된 경로를 만들 수 있습니다. 이렇게 하면 피어링 연결을 통한 트래픽 흐름을 필요한 정도로만 제한할 수 있어 더 효율적일 수 있습니다. 하지만 통신이 필요한 피어 VPC에 새 리소스를 추가할 때마다 라우팅 테이블을 업데이트해야 하므로 유지 관리가 더 필요합니다.

최선의 접근 방식은 VPC 아키텍처의 규모와 복잡성, VPC 간에 예상되는 트래픽의 양, 보안 및 리소스 액세스와 관련된 조직의 요구 사항 등의 요인에 따라 달라집니다. 많은 기업이 일반적인 트래픽 패턴에는 광범위한 경로를 사용하고 더 민감하거나 대역폭 집약적인 사용 사례에는 특정 경로를 사용하는 하이브리드 접근 방식을 취합니다.

구성

- 전체 VPC에 대한 경로가 있는 VPC 피어링 구성
- 특정 경로로 VPC 피어링 구성

전체 VPC에 대한 경로가 있는 VPC 피어링 구성

라우팅 테이블이 피어 VPC의 전체 CIDR 블록에 액세스하도록 VPC 피어링 연결을 구성할 수 있습니다. 특정 VPC 피어링 연결 구성이 필요할 수 있는 시나리오에 대한 자세한 정보는 VPC 피어링 연결 네트워킹 시나리오을(를) 참조하세요. VPC 피어링 연결을 생성하고 사용하는 방법에 대한 자세한 내용은 VPC 피어링 연결 단원을 참조하세요.

라우팅 테이블 업데이트에 대한 자세한 내용은 VPC 피어링 연결을 위한 라우팅 테이블 업데이트 단원을 참조하세요.

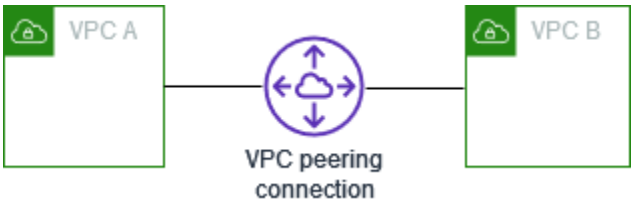
구성

- 함께 피어링된 두 개의 VPC

- VPC 두 개와 피어링된 하나의 VPC
- 함께 피어링된 세 개의 VPC
- 서로 피어링된 여러 VPC

함께 피어링된 두 개의 VPC

이 구성에서는 VPC A와 VPC B 간에 피어링 연결이 있습니다(pcx-11112222). VPC가 동일한 AWS 계정에 있으며 해당 CIDR 블록이 겹치지 않습니다.



서로의 리소스가 필요한 2개의 VPC가 있을 때 이 구성을 사용할 수 있습니다. 예를 들어 회계 레코드에 VPC A를 설정하고, 재무 레코드에 VPC B를 설정하면 이러한 각 VPC에서 다른 VPC의 리소스에 제한 없이 액세스하게 할 수 있게 됩니다.

단일 VPC CIDR

피어 VPC의 CIDR 블록에 대한 트래픽을 VPC 피어링 연결로 보내는 경로로 각 VPC의 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-11112222
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-11112222

여러 IPv4 VPC CIDR

VPC A와 VPC B에 IPv4 CIDR 블록이 여러 개 연결되어 있는 경우 피어 VPC의 일부 또는 모든 IPv4 CIDR 블록에 대한 경로로 각 VPC의 라우팅 테이블을 업데이트할 수 있습니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR 1*	로컬
	VPC A CIDR 2	로컬
	VPC B CIDR 1	pcx-11112222
	VPC B CIDR 2	pcx-11112222
VPC B	*VPC B CIDR 1*	로컬
	VPC B CIDR 2	로컬
	VPC A CIDR 1	pcx-11112222
	VPC A CIDR 2	pcx-11112222

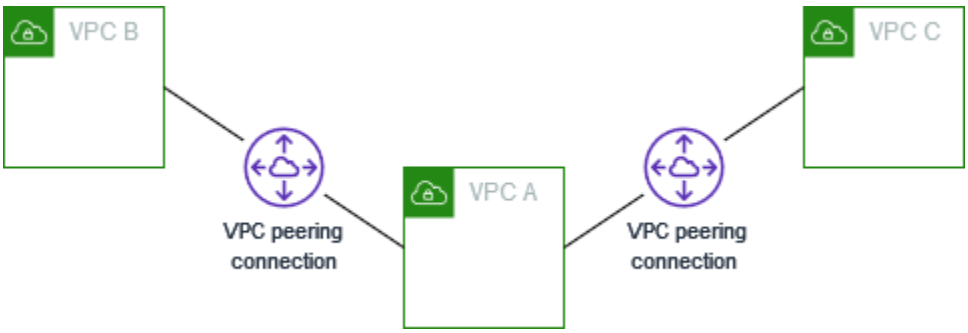
IPv4 및 IPv6 VPC CIDR

연결된 IPv6 CIDR 블록이 VPC A와 VPC B에 있는 경우 피어 VPC의 IPv4 블록과 IPv6 CIDR 블록 둘 다에 대한 경로로 각 VPC의 라우팅 테이블을 업데이트할 수 있습니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A IPv4 CIDR*	로컬
	VPC A IPv6 CIDR	로컬
	VPC B IPv4 CIDR	pcx-11112222
	VPC B IPv6 CIDR	pcx-11112222
VPC B	*VPC B IPv4 CIDR*	로컬
	VPC B IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-11112222
	VPC A IPv6 CIDR	pcx-11112222

VPC 두 개와 피어링된 하나의 VPC

이 구성에는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-12121212) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-23232323)이 있습니다. 세 VPC 모두 동일한 AWS 계정에 있으며 해당 CIDR 블록이 겹치지 않습니다.



VPC는 전이적 피어링 관계가 지원되지 않으므로 VPC B와 VPC C는 허브 VPC를 통해 서로 직접 트래픽을 전송할 수 없습니다. 함께 피어링된 세 개의 VPC에 표시된 대로 VPC B와 VPC C 사이의 VPC 피어링 연결을 생성할 수 있습니다. 지원되지 않는 피어링 시나리오에 대한 자세한 내용은 the section called “VPC 피어링 제한”을 참조하세요.

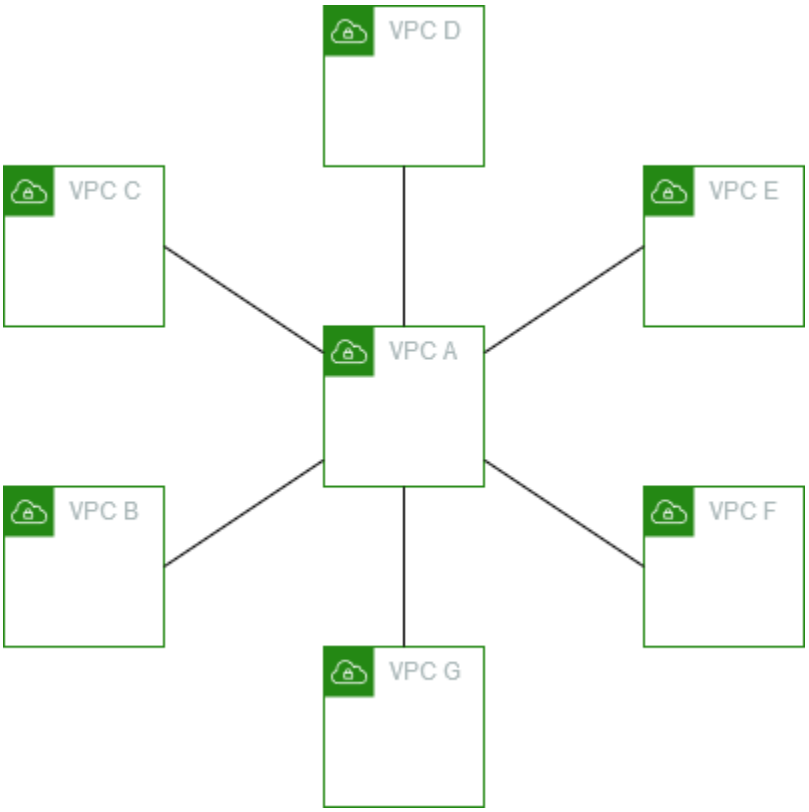
다른 VPC에서 액세스해야 하는 중앙 VPC(예: 서비스 리포지토리)에 리소스가 있을 때 이 구성을 사용할 수 있습니다. 다른 VPC에서는 서로의 리소스를 액세스할 필요가 없으며 중앙 VPC의 리소스만 액세스하면 됩니다.

VPC당 하나의 CIDR 블록을 사용하여 이 구성을 구현하려면 다음과 같이 각 VPC의 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-12121212
	VPC C CIDR	pcx-23232323
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-12121212
VPC C	*VPC C CIDR*	로컬

라우팅 테이블	대상 주소	대상
	VPC A CIDR	pcx-23232323

이 구성을 추가 VPC로 확장할 수 있습니다. 예를 들어 VPC A는 IPv4 및 IPv6 CIDR를 모두 사용하여 VPC B ~ VPC G에 피어링되지만, 다른 VPC는 서로 피어링되지 않습니다. 이 다이어그램에서 선은 VPC 피어링 연결을 나타냅니다.



다음과 같이 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A IPv4 CIDR*	로컬
	VPC A IPv6 CIDR	로컬
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb

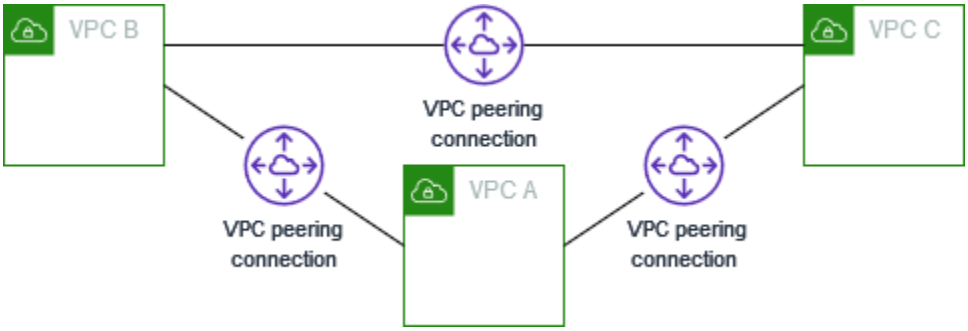
라우팅 테이블	대상 주소	대상
	VPC C IPv4 CIDR	pcx-aaaacccc
	VPC C IPv6 CIDR	pcx-aaaacccc
	VPC D IPv4 CIDR	pcx-aaaadddd
	VPC D IPv6 CIDR	pcx-aaaadddd
	VPC E IPv4 CIDR	pcx-aaaaeeee
	VPC E IPv6 CIDR	pcx-aaaaeeee
	VPC F IPv4 CIDR	pcx-aaaaffff
	VPC F IPv6 CIDR	pcx-aaaaffff
	VPC G IPv4 CIDR	pcx-aaaagggg
	VPC G IPv6 CIDR	pcx-aaaagggg
VPC B	*VPC B IPv4 CIDR*	로컬
	VPC B IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
VPC C	*VPC C IPv4 CIDR*	로컬
	VPC C IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC A IPv6 CIDR	pcx-aaaacccc
VPC D	*VPC D IPv4 CIDR*	로컬
	VPC D IPv6 CIDR	로컬

라우팅 테이블	대상 주소	대상
	VPC A IPv4 CIDR	pcx-aaaadddd
	VPC A IPv6 CIDR	pcx-aaaadddd
VPC E	*VPC E IPv4 CIDR*	로컬
	VPC E IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaaeeee
	VPC A IPv6 CIDR	pcx-aaaaeeee
VPC F	*VPC F IPv4 CIDR*	로컬
	VPC F IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaaffff
	VPC A IPv6 CIDR	pcx-aaaaffff
VPC G	*VPC G IPv4 CIDR*	로컬
	VPC G IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaagggg
	VPC A IPv6 CIDR	pcx-aaaagggg

함께 피어링된 세 개의 VPC

이 구성에서는 동일한 AWS 계정에 겹치지 않는 CIDR 블록이 있는 3개의 VPC가 있습니다. VPC는 다음과 같이 풀 메시로 피어링됩니다.

- VPC A는 VPC 피어링 연결 `pcx-aaaabbbb`를 통해 VPC B와 피어링됩니다.
- VPC A는 VPC 피어링 연결 `pcx-aaaacccc`를 통해 VPC C와 피어링됩니다.
- VPC B는 VPC 피어링 연결 `pcx-bbbbcccc`를 통해 VPC C와 피어링됩니다.



리소스를 서로 제한 없이 공유해야 하는 VPC가 있을 때 이 구성을 사용할 수 있습니다. 예를 들어 파일 공유 시스템으로 사용할 수 있습니다.

이 구성을 구현하려면 다음과 같이 각 VPC의 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-bbbbcccc
VPC C	*VPC C CIDR*	로컬
	VPC A CIDR	pcx-aaaacccc
	VPC B CIDR	pcx-bbbbcccc

VPC A와 VPC B에 IPv4 및 IPv6 CIDR 블록이 모두 있지만, VPC C에 IPv6 CIDR 블록이 없는 경우 다음과 같이 라우팅 테이블을 업데이트합니다. VPC A와 VPC B의 리소스에서는 VPC 피어링 연결을 통해 IPv6를 사용하여 통신할 수 있습니다. 그러나 VPC C에서는 IPv6를 사용하여 VPC A 또는 VPC B와 통신할 수 없습니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A IPv4 CIDR*	로컬
	VPC A IPv6 CIDR	로컬
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-aaaacccc
VPC B	*VPC B IPv4 CIDR*	로컬
	VPC B IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-bbbbcccc
VPC C	*VPC C IPv4 CIDR*	로컬
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC B IPv4 CIDR	pcx-bbbbcccc

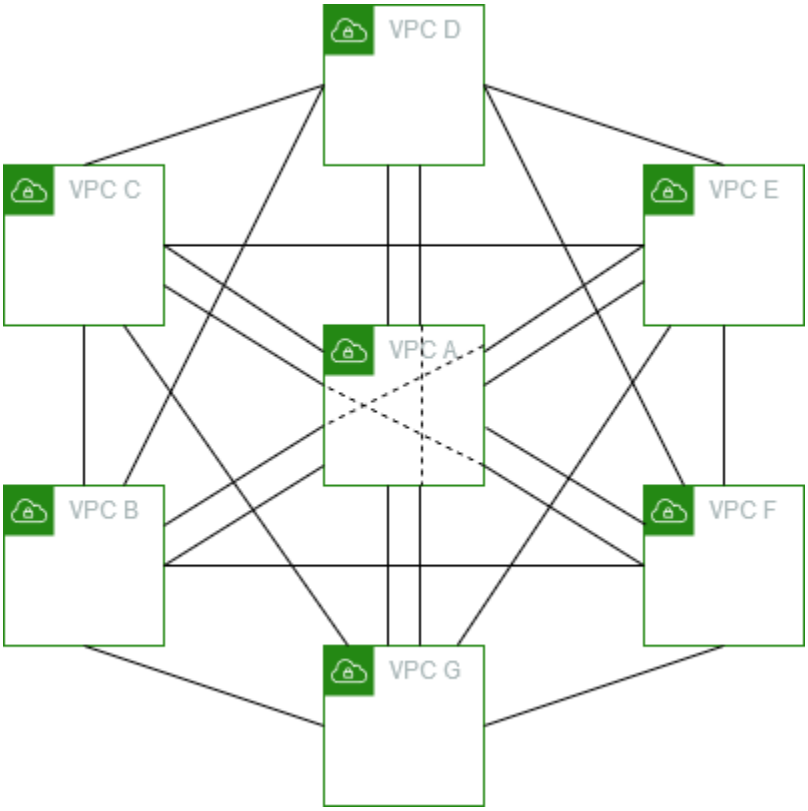
서로 피어링된 여러 VPC

이 구성에는 풀 메시 구성에서 피어링된 7개의 VPC가 있습니다. VPC가 동일한 AWS 계정에 있으며 해당 CIDR 블록이 겹치지 않습니다.

VPC	VPC	VPC 피어링 연결
A	B	pcx-aaaabbbb
A	C	pcx-aaaacccc
A	D	pcx-aaaadddd

VPC	VPC	VPC 피어링 연결
A	E	pcx-aaaaeeee
A	F	pcx-aaaaffff
A	G	pcx-aaaagggg
B	C	pcx-bbbbcccc
B	D	pcx-bbbbdddd
B	E	pcx-bbbbeeee
B	F	pcx-bbbbffff
B	G	pcx-bbbbgggg
C	D	pcx-ccccdddd
C	E	pcx-cccceeee
C	F	pcx-ccccffff
C	G	pcx-ccccgggg
D	E	pcx-ddddeeee
D	F	pcx-ddddffff
D	G	pcx-ddddgggg
E	F	pcx-eeeeffff
E	G	pcx-eeeegggg
F	G	pcx-ffffgggg

서로의 리소스를 제한 없이 액세스할 수 있어야 하는 여러 VPC가 있을 때 이 구성을 사용할 수 있습니다. 예를 들어 파일 공유 네트워크로 사용할 수 있습니다. 이 다이어그램에서 선은 VPC 피어링 연결을 나타냅니다.



이 구성을 구현하려면 다음과 같이 각 VPC의 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
	VPC D CIDR	pcx-aaaadddd
	VPC E CIDR	pcx-aaaaeeee
	VPC F CIDR	pcx-aaaaffff
	VPC G CIDR	pcx-aaaagggg
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-aaaabbbb

라우팅 테이블	대상 주소	대상
	VPC C CIDR	pcx-bbbbcccc
	VPC D CIDR	pcx-bbbbdddd
	VPC E CIDR	pcx-bbbbeeee
	VPC F CIDR	pcx-bbbbffff
	VPC G CIDR	pcx-bbbbgggg
VPC C	*VPC C CIDR*	로컬
	VPC A CIDR	pcx-aaaacccc
	VPC B CIDR	pcx-bbbbcccc
	VPC D CIDR	pcx-ccccdddd
	VPC E CIDR	pcx-cccceeee
	VPC F CIDR	pcx-ccccffff
	VPC G CIDR	pcx-ccccgggg
VPC D	*VPC D CIDR*	로컬
	VPC A CIDR	pcx-aaaadddd
	VPC B CIDR	pcx-bbbbdddd
	VPC C CIDR	pcx-ccccdddd
	VPC E CIDR	pcx-ddddeeee
	VPC F CIDR	pcx-ddddffff
	VPC G CIDR	pcx-ddddgggg
VPC E	*VPC E CIDR*	로컬

라우팅 테이블	대상 주소	대상
	VPC A CIDR	pcx-aaaaeeee
	VPC B CIDR	pcx-bbbbeeee
	VPC C CIDR	pcx-cccceeee
	VPC D CIDR	pcx-ddddeeee
	VPC F CIDR	pcx-eeeeffff
	VPC G CIDR	pcx-eeeegggg
VPC F	*VPC F CIDR*	로컬
	VPC A CIDR	pcx-aaaaffff
	VPC B CIDR	pcx-bbbbffff
	VPC C CIDR	pcx-ccccffff
	VPC D CIDR	pcx-ddddffff
	VPC E CIDR	pcx-eeeeffff
	VPC G CIDR	pcx-ffffgggg
VPC G	*VPC G CIDR*	로컬
	VPC A CIDR	pcx-aaaagggg
	VPC B CIDR	pcx-bbbbgggg
	VPC C CIDR	pcx-ccccgggg
	VPC D CIDR	pcx-ddddgggg
	VPC E CIDR	pcx-eeeegggg
	VPC F CIDR	pcx-ffffgggg

모든 VPC에 IPv6 CIDR 블록이 연결되어 있는 경우 다음과 같이 라우팅 테이블을 업데이트합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A IPv4 CIDR*	로컬
	VPC A IPv6 CIDR	로컬
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-aaaacccc
	VPC C IPv6 CIDR	pcx-aaaacccc
	VPC D IPv4 CIDR	pcx-aaaadddd
	VPC D IPv6 CIDR	pcx-aaaadddd
	VPC E IPv4 CIDR	pcx-aaaaeeee
	VPC E IPv6 CIDR	pcx-aaaaeeee
	VPC F IPv4 CIDR	pcx-aaaaffff
	VPC F IPv6 CIDR	pcx-aaaaffff
	VPC G IPv4 CIDR	pcx-aaaagggg
	VPC G IPv6 CIDR	pcx-aaaagggg
VPC B	*VPC B IPv4 CIDR*	로컬
	VPC B IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-bbbbcccc

라우팅 테이블	대상 주소	대상
	VPC C IPv6 CIDR	pcx-bbbbcccc
	VPC D IPv4 CIDR	pcx-bbbbdddd
	VPC D IPv6 CIDR	pcx-bbbbdddd
	VPC E IPv4 CIDR	pcx-bbbbeeee
	VPC E IPv6 CIDR	pcx-bbbbeeee
	VPC F IPv4 CIDR	pcx-bbbbffff
	VPC F IPv6 CIDR	pcx-bbbbffff
	VPC G IPv4 CIDR	pcx-bbbbgggg
	VPC G IPv6 CIDR	pcx-bbbbgggg
VPC C	*VPC C IPv4 CIDR*	로컬
	VPC C IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC A IPv6 CIDR	pcx-aaaacccc
	VPC B IPv4 CIDR	pcx-bbbbcccc
	VPC B IPv6 CIDR	pcx-bbbbcccc
	VPC D IPv4 CIDR	pcx-ccccdddd
	VPC D IPv6 CIDR	pcx-ccccdddd
	VPC E IPv4 CIDR	pcx-cccceeee
	VPC E IPv6 CIDR	pcx-cccceeee
	VPC F IPv4 CIDR	pcx-ccccffff

라우팅 테이블	대상 주소	대상
	VPC F IPv6 CIDR	pcx-ccccffff
	VPC G IPv4 CIDR	pcx-ccccgggg
	VPC G IPv6 CIDR	pcx-ccccgggg
VPC D	*VPC D IPv4 CIDR*	로컬
	VPC D IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaadddd
	VPC A IPv6 CIDR	pcx-aaaadddd
	VPC B IPv4 CIDR	pcx-bbbbdddd
	VPC B IPv6 CIDR	pcx-bbbbdddd
	VPC C IPv4 CIDR	pcx-ccccdddd
	VPC C IPv6 CIDR	pcx-ccccdddd
	VPC E IPv4 CIDR	pcx-ddddeeee
	VPC E IPv6 CIDR	pcx-ddddeeee
	VPC F IPv4 CIDR	pcx-ddddffff
	VPC F IPv6 CIDR	pcx-ddddffff
	VPC G IPv4 CIDR	pcx-ddddgggg
	VPC G IPv6 CIDR	pcx-ddddgggg
VPC E	*VPC E IPv4 CIDR*	로컬
	VPC E IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaaeeee

라우팅 테이블	대상 주소	대상
	VPC A IPv6 CIDR	pcx-aaaaeeee
	VPC B IPv4 CIDR	pcx-bbbbeeee
	VPC B IPv6 CIDR	pcx-bbbbeeee
	VPC C IPv4 CIDR	pcx-cccceeee
	VPC C IPv6 CIDR	pcx-cccceeee
	VPC D IPv4 CIDR	pcx-ddddeeee
	VPC D IPv6 CIDR	pcx-ddddeeee
	VPC F IPv4 CIDR	pcx-eeeeffff
	VPC F IPv6 CIDR	pcx-eeeeffff
	VPC G IPv4 CIDR	pcx-eeeegggg
	VPC G IPv6 CIDR	pcx-eeeegggg
VPC F	*VPC F IPv4 CIDR*	로컬
	VPC F IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaaffff
	VPC A IPv6 CIDR	pcx-aaaaffff
	VPC B IPv4 CIDR	pcx-bbbbffff
	VPC B IPv6 CIDR	pcx-bbbbffff
	VPC C IPv4 CIDR	pcx-ccccffff
	VPC C IPv6 CIDR	pcx-ccccffff
	VPC D IPv4 CIDR	pcx-ddddffff

라우팅 테이블	대상 주소	대상
	VPC D IPv6 CIDR	pcx-ddddffff
	VPC E IPv4 CIDR	pcx-eeeeffff
	VPC E IPv6 CIDR	pcx-eeeeffff
	VPC G IPv4 CIDR	pcx-ffffgggg
	VPC G IPv6 CIDR	pcx-ffffgggg
VPC G	*VPC G IPv4 CIDR*	로컬
	VPC G IPv6 CIDR	로컬
	VPC A IPv4 CIDR	pcx-aaaagggg
	VPC A IPv6 CIDR	pcx-aaaagggg
	VPC B IPv4 CIDR	pcx-bbbbgggg
	VPC B IPv6 CIDR	pcx-bbbbgggg
	VPC C IPv4 CIDR	pcx-ccccgggg
	VPC C IPv6 CIDR	pcx-ccccgggg
	VPC D IPv4 CIDR	pcx-ddddgggg
	VPC D IPv6 CIDR	pcx-ddddgggg
	VPC E IPv4 CIDR	pcx-eeeegggg
	VPC E IPv6 CIDR	pcx-eeeegggg
	VPC F IPv4 CIDR	pcx-ffffgggg
	VPC F IPv6 CIDR	pcx-ffffgggg

특정 경로로 VPC 피어링 구성

서브넷 CIDR 블록, 특정 CIDR 블록(VPC에 여러 CIDR 블록이 있는 경우) 또는 피어 VPC의 특정 리소스에 대한 액세스를 제한하도록 VPC 피어링 연결에 대한 라우팅 테이블을 구성할 수 있습니다. 이 예시에서는 겹치는 CIDR 블록이 있는 최소 2개의 VPC로 중앙 VPC가 피어링됩니다.

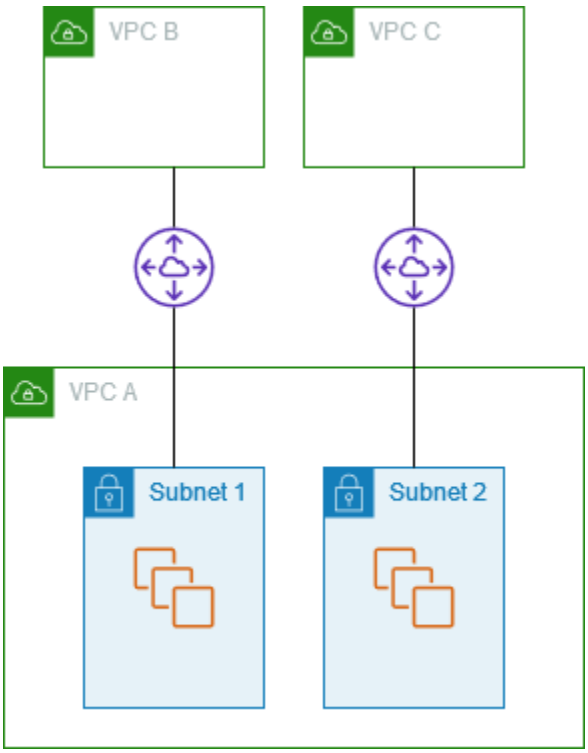
특정 VPC 피어링 연결 구성이 필요한 시나리오의 예시는 VPC 피어링 연결 네트워킹 시나리오 단원을 참조하세요. VPC 피어링 연결을 사용하는 방법에 대한 자세한 내용은 VPC 피어링 연결 섹션을 참조하세요. 라우팅 테이블 업데이트에 대한 자세한 내용은 VPC 피어링 연결을 위한 라우팅 테이블 업데이트 단원을 참조하세요.

구성

- 1개의 VPC에서 특정 서브넷에 액세스하는 VPC 2개
- 1개의 VPC에서 특정 CIDR 블록에 액세스하는 VPC 2개
- 2개의 VPC에서 특정 서브넷에 액세스하는 VPC 1개
- VPC 2개의 특정 인스턴스에 액세스하는 VPC 1개의 인스턴스
- 가장 긴 접두사 일치를 사용하여 VPC 2개에 액세스하는 VPC 1개
- 여러 VPC 구성

1개의 VPC에서 특정 서브넷에 액세스하는 VPC 2개

이 구성에는 2개의 서브넷이 있는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-aaaabbbb) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-aaaacccc)이 있습니다. 각 VPC는 VPC A의 서브넷 중 하나에 있는 리소스에만 액세스해야 합니다.



서브넷 1의 라우팅 테이블은 VPC 피어링 연결 pcx-aaaabbbb를 사용하여 VPC B의 전체 CIDR 블록에 액세스합니다. VPC B의 라우팅 테이블은 pcx-aaaabbbb를 사용하여 VPC A에 있는 서브넷 1의 CIDR 블록에 액세스합니다. 서브넷 2의 라우팅 테이블은 VPC 피어링 연결 pcx-aaaacccc를 사용하여 VPC C의 전체 CIDR 블록에 액세스합니다. VPC C 테이블의 라우팅 테이블은 pcx-aaaacccc를 사용하여 VPC A에 있는 서브넷 2의 CIDR 블록에 액세스합니다.

라우팅 테이블	대상 주소	대상
서브넷 1(VPC A)	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-aaaabbbb
서브넷 2(VPC A)	*VPC A CIDR*	로컬
	VPC C CIDR	pcx-aaaacccc
VPC B	*VPC B CIDR*	로컬
	### 1 CIDR	pcx-aaaabbbb
VPC C	*VPC C CIDR*	로컬

라우팅 테이블	대상 주소	대상
	### 2 CIDR	pcx-aaaacccc

이 구성을 여러 CIDR 블록으로 확장할 수 있습니다. VPC A와 VPC B에 IPv4 및 IPv6 CIDR 블록이 모두 있으며, 서브넷 1에는 연결된 IPv6 CIDR 블록이 있다고 가정하겠습니다. VPC 피어링 연결을 사용하여 IPv6를 통해 VPC A의 서브넷 1과 통신하도록 VPC B를 활성화할 수 있습니다. 이렇게 하려면 VPC B의 IPv6 CIDR 블록 대상이 있는 VPC A의 라우팅 테이블에 대한 경로와 VPC A에 있는 서브넷 1의 IPv6 CIDR 대상이 있는 VPC B의 라우팅 테이블에 대한 경로를 추가합니다.

라우팅 테이블	대상 주소	대상	Notes
VPA A의 서브넷 1	*VPC A IPv4 CIDR*	로컬	
	VPC A IPv6 CIDR	로컬	VPC 내부의 IPv6 통신에 자동으로 추가되는 로컬 경로.
	VPC B IPv4 CIDR	pcx-aaaabbbb	
	VPC B IPv6 CIDR	pcx-aaaabbbb	VPC B의 IPv6 CIDR 블록에 대한 경로.
VPA A의 서브넷 2	*VPC A IPv4 CIDR*	로컬	
	VPC A IPv6 CIDR	로컬	VPC 내부의 IPv6 통신에 자동으로 추가되는 로컬 경로.
	VPC C IPv4 CIDR	pcx-aaaacccc	
VPC B	*VPC B IPv4 CIDR*	로컬	
	VPC B IPv6 CIDR	로컬	VPC 내부의 IPv6 통신에 자동으로 추가되는 로컬 경로.
	### 1 IPv4 CIDR	pcx-aaaabbbb	

라우팅 테이블	대상 주소	대상	Notes
VPC C	*### 1 IPv6 CIDR*	pcx-aaaabbbb	VPC A의 IPv6 CIDR 블록에 대한 경로.
	VPC C IPv4 CIDR	로컬	
	### 2 IPv4 CIDR	pcx-aaaacccc	

1개의 VPC에서 특정 CIDR 블록에 액세스하는 VPC 2개

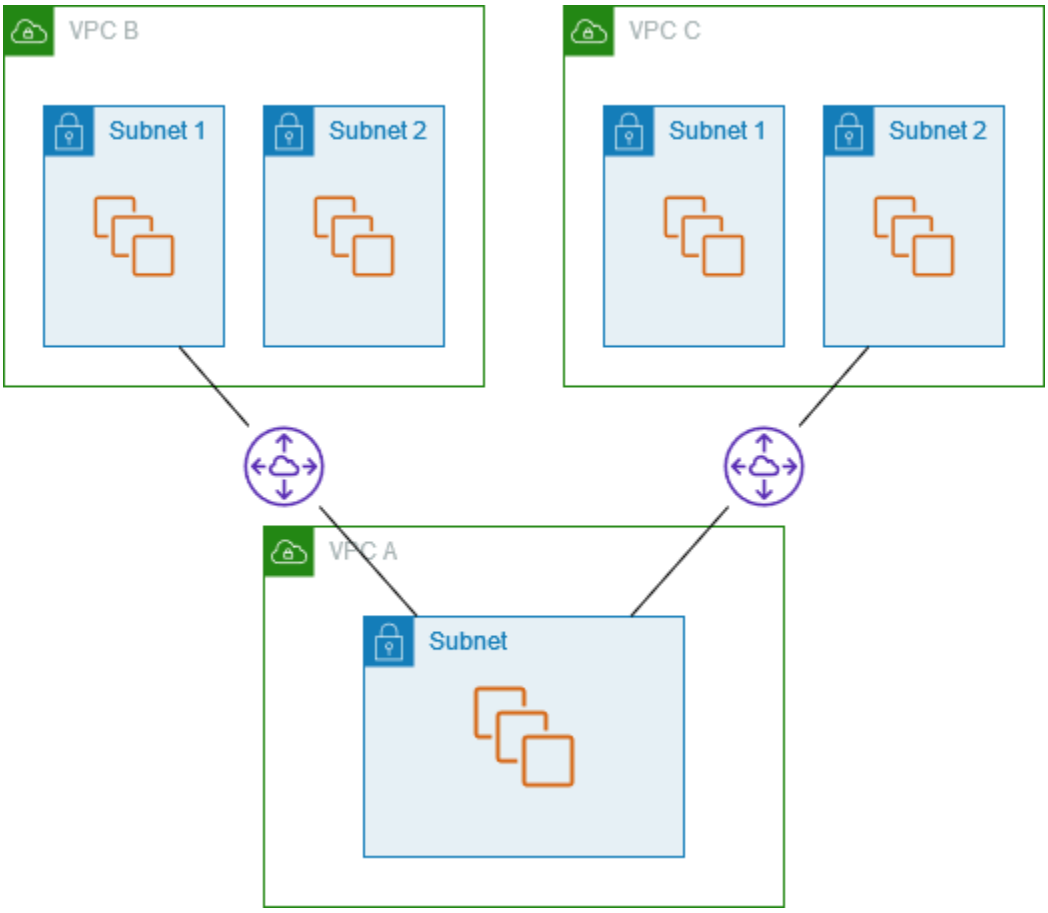
이 구성에는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-aaaabbbb) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-aaaacccc)이 있습니다. VPC A에는 각 피어링 연결에 대해 하나의 CIDR 블록이 있습니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR 1*	로컬
	VPC A CIDR 2	로컬
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR 1	pcx-aaaabbbb
VPC C	*VPC C CIDR*	로컬
	VPC A CIDR 2	pcx-aaaacccc

2개의 VPC에서 특정 서브넷에 액세스하는 VPC 1개

이 구성에는 1개의 서브넷이 있는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-aaaabbbb) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-aaaacccc)이 있습니다. VPC B와 VPC C에

는 각각 2개의 서브넷이 있습니다. VPC A와 VPC B 간의 피어링 연결은 VPC B의 서브넷 중 하나만 사용합니다. VPC A와 VPC C 간의 피어링 연결은 VPC C의 서브넷 중 하나만 사용합니다.



다른 VPC에서 액세스해야 하는 단일 리소스 집합(예: Active Directory 서비스)이 있는 중앙 VPC가 있을 때 이 구성을 사용합니다. 중앙 VPC에는 피어링할 VPC에 대한 모든 액세스 권한이 필요하지 않습니다.

VPC A의 라우팅 테이블은 피어링 연결을 사용하여 피어링된 VPC의 특정 서브넷에만 액세스합니다. 서브넷 1의 라우팅 테이블은 VPC A와의 피어링 연결을 사용하여 VPC A의 서브넷에 액세스합니다. 서브넷 2의 라우팅 테이블은 VPC A와의 피어링 연결을 사용하여 VPC A의 서브넷에 액세스합니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	### 1 CIDR	pcx-aaaabbbb
	### 2 CIDR	pcx-aaaacccc

라우팅 테이블	대상 주소	대상
서브넷 1(VPC B)	*VPC B CIDR*	로컬
	VPC A CIDR# ###	pcx-aaaabbbb
서브넷 2(VPC C)	*VPC C CIDR*	로컬
	VPC A CIDR# ###	pcx-aaaacccc

응답 트래픽을 위한 라우팅

겹치거나 일치하는 CIDR 블록을 가진 여러 VPC와 피어링된 VPC가 있는 경우, 라우팅 테이블이 자신의 VPC에서 잘못된 VPC로 응답 트래픽을 전송하지 못하도록 구성되어 있는지 확인하세요. AWS에서는 패킷의 소스 IP를 확인하고 응답 패킷을 다시 소스로 라우팅하는 VPC 피어링 연결에서 유니캐스트 역경로 전달을 지원하지 않습니다.

예를 들어, VPC A는 VPC B 및 VPC C와 피어링됩니다. VPC B 및 VPC C에는 일치하는 CIDR 블록이 있으며, 해당 서브넷에는 일치하는 CIDR 블록이 있습니다. VPC B의 서브넷 2에 대한 라우팅 테이블은 VPC A 서브넷에 액세스하는 VPC 피어링 연결 pcx-aaaabbbb를 가리킵니다. VPC A 라우팅 테이블은 VPC CIDR로 향하는 트래픽을 피어링 연결 pcx-aaaaccccc에 보내도록 구성됩니다.

라우팅 테이블	대상 주소	대상
서브넷 2(VPC B)	*VPC B CIDR*	로컬
	VPC A CIDR# ###	pcx-aaaabbbb
VPC A	*VPC A CIDR*	로컬
	VPC C CIDR	pcx-aaaacccc

VPC B에 있는 서브넷 2의 인스턴스에서 VPC 피어링 연결 pcx-aaaabbbb를 사용하여 VPC A에 있는 Active Directory 서버에 트래픽을 보낸다고 가정하겠습니다. VPC A에서는 Active Directory 서버에 응답 트래픽을 보냅니다. 그러나 VPC A 라우팅 테이블은 VPC CIDR 범위 내의 모든 트래픽을 VPC 피어링 연결 pcx-aaaacccc에 보내도록 구성됩니다. VPC C의 서브넷 2의 경우 VPC B의 서브넷 2에 있는 인스턴스와 동일한 IP 주소가 있으면 VPC A의 응답 트래픽을 받습니다. VPC B의 서브넷 2에 있는 인스턴스에서는 VPC A에 대한 요청의 응답을 받지 않습니다.

이를 방지하기 위해 VPC B에 있는 서브넷 2의 CIDR로 VPC A 라우팅 테이블에 대한 특정 경로를 대상 겸 pcx-aaaabbbb의 대상으로 추가할 수 있습니다. 새 경로가 더 구체적이므로 서브넷 2 CIDR로 향하는 트래픽이 VPC 피어링 연결 pcx-aaaabbbb로 라우팅됩니다.

또는, 다음 예에서는 VPC A 라우팅 테이블에는 각 VPC 피어링 연결의 각 서브넷에 대한 경로가 있습니다. VPC A에서는 VPC B의 서브넷 2 및 VPC C의 서브넷 1과 통신할 수 있습니다. 이 시나리오는 VPC B 및 VPC C와 동일한 IP 주소 범위 내에 속하는 다른 서브넷으로 다른 VPC 피어링 연결을 추가해야 하는 경우에 유용합니다. 해당 특정 서브넷에 대한 다른 경로를 단하게 추가할 수 있습니다.

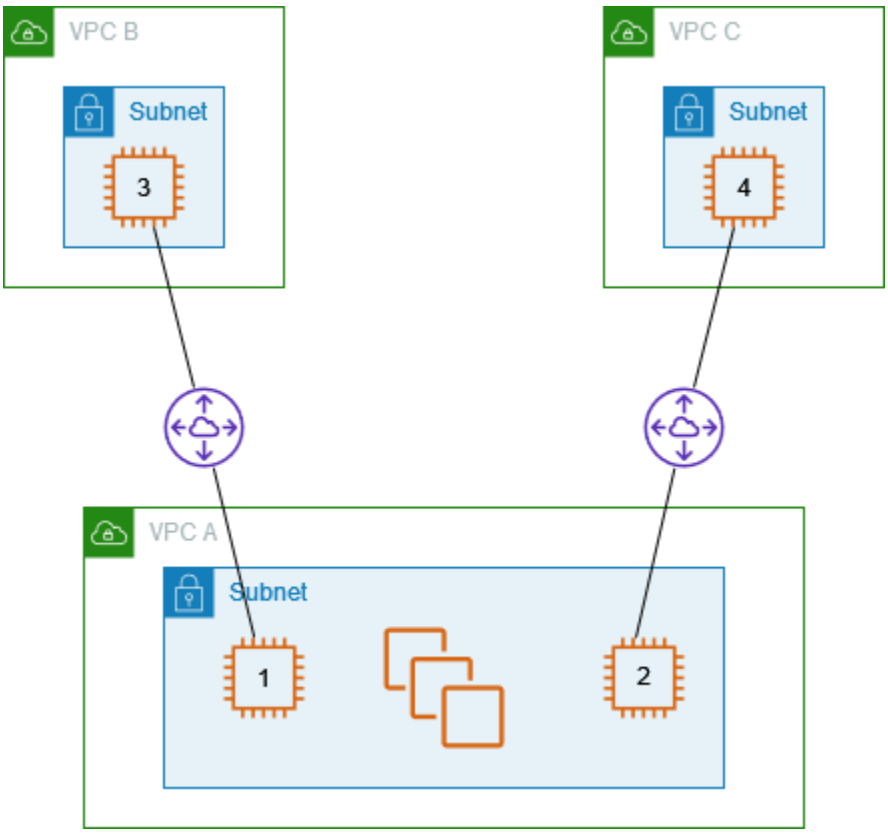
대상 주소	대상
VPC A CIDR	로컬
### 2 CIDR	pcx-aaaabbbb
### 1 CIDR	pcx-aaaacccc

그 대신, 사용 사례에 따라 다음과 같이 트래픽이 올바른 서버로 다시 라우팅되도록 VPC B에서 특정 IP 주소에 대한 경로를 생성할 수 있습니다(라우팅 테이블은 가장 긴 접두사 일치 항목을 사용하여 경로의 우선 순위를 지정함).

대상 주소	대상
VPC A CIDR	로컬
### 2# ## IP ##	pcx-aaaabbbb
VPC B CIDR	pcx-aaaacccc

VPC 2개의 특정 인스턴스에 액세스하는 VPC 1개의 인스턴스

이 구성에는 1개의 서브넷이 있는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-aaaabbbb) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-aaaacccc)이 있습니다. VPC A에는 각 피어링 연결에 대해 하나의 인스턴스가 있는 서브넷이 있습니다. 이 구성을 사용하여 특정 인스턴스에 대한 피어링 트래픽을 제한할 수 있습니다.

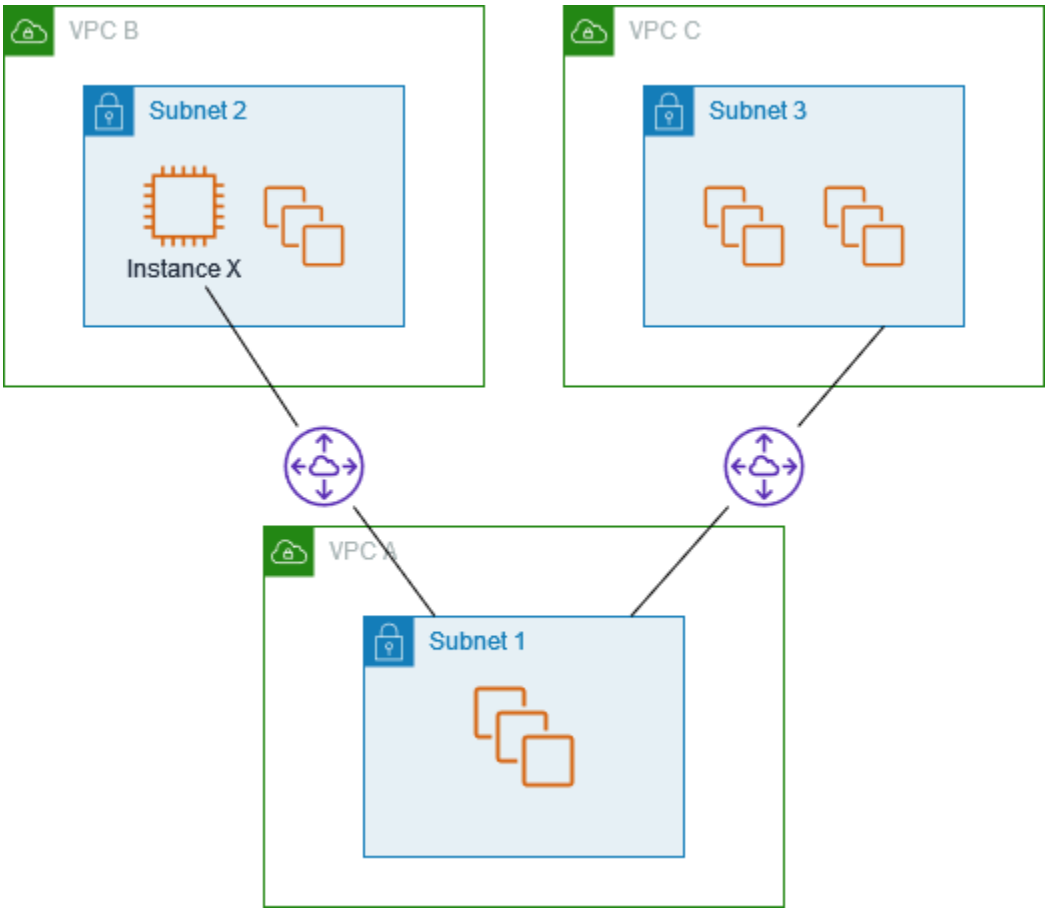


각 VPC 라우팅 테이블은 피어 VPC의 단일 IP 주소(더불어 특정 인스턴스)에 액세스하기 위해 관련 VPC 피어링 연결을 가리킵니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	#### 3 IP ##	pcx-aaaabbbb
	#### 4 IP ##	pcx-aaaacccc
VPC B	*VPC B CIDR*	로컬
	#### 1 IP ##	pcx-aaaabbbb
VPC C	*VPC C CIDR*	로컬
	#### 2 IP ##	pcx-aaaacccc

가장 긴 접두사 일치를 사용하여 VPC 2개에 액세스하는 VPC 1개

이 구성에는 1개의 서브넷이 있는 중앙 VPC(VPC A), VPC A와 VPC B 사이의 피어링 연결(pcx-aaaabbbb) 및 VPC A와 VPC C 사이의 피어링 연결(pcx-aaaacccc)이 있습니다. VPC B와 VPC C에 일치하는 CIDR 블록이 있습니다. VPC 피어링 연결 pcx-aaaabbbb를 사용하여 VPC A와 VPC B의 특정 인스턴스 사이에 트래픽을 라우팅합니다. VPC B와 VPC C에서 공유하는 CIDR IP 주소 범위로 향하는 다른 모든 트래픽은 pcx-aaaacccc를 통해 VPC C로 라우팅됩니다.



VPC 라우팅 테이블은 가장 긴 접두사 일치 항목을 사용하여 의도하는 VPC 피어링 연결에서 가장 구체적인 경로를 선택합니다. 다른 모든 트래픽은 그 다음 일치하는 경로를 통해 라우팅되며 여기서는 VPC 피어링 연결 pcx-aaaacccc입니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR ##*	로컬
	#### X IP ##	pcx-aaaabbbb

라우팅 테이블	대상 주소	대상
	VPC C CIDR ##	pcx-aaaacccc
VPC B	*VPC B CIDR ##*	로컬
	VPC A CIDR ##	pcx-aaaabbbb
VPC C	*VPC C CIDR ##*	로컬
	VPC A CIDR ##	pcx-aaaacccc

Important

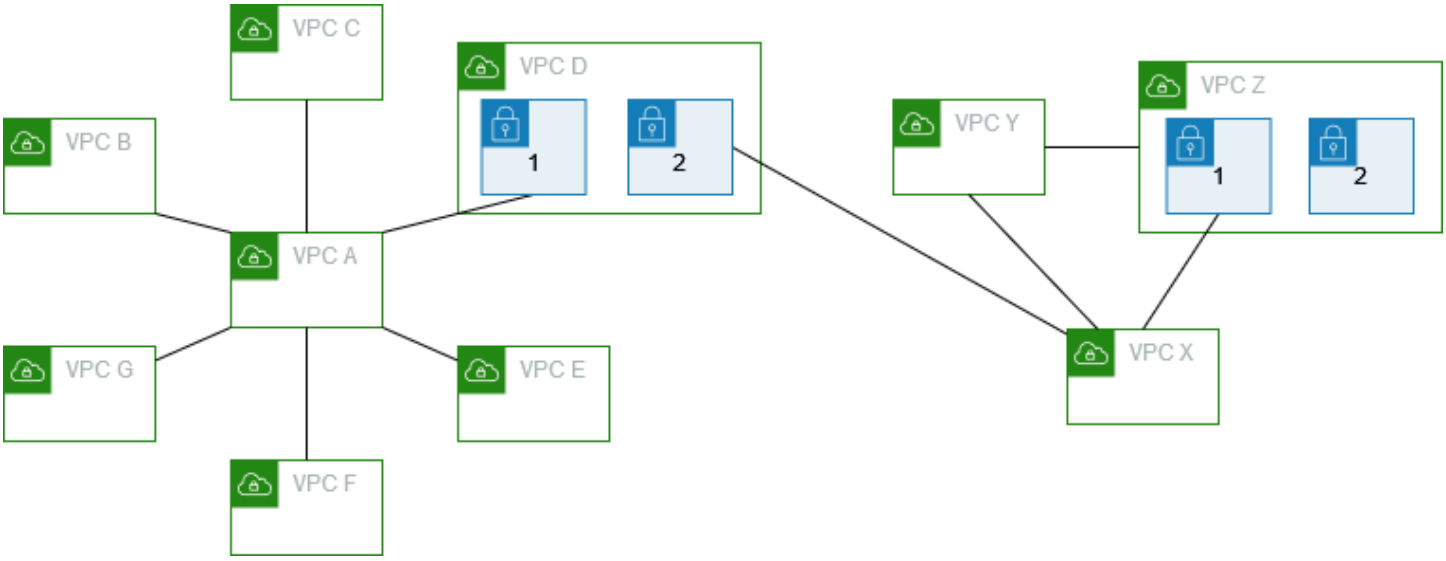
VPC B의 인스턴스 X가 아닌 인스턴스에서 VPC A에 트래픽을 보내는 경우 응답 트래픽이 VPC B 대신에 VPC C로 라우팅될 수 있습니다. 자세한 내용은 응답 트래픽을 위한 라우팅 섹션을 참조하세요.

여러 VPC 구성

이 구성에서는 중앙 VPC(VPC A)가 스포크 구성에서 여러 VPC와 피어링됩니다. 또한 풀 메시 구성에서 피어링된 VPC 3개(VPC X, Y, Z)도 있습니다.

VPC D에도 VPC X(`pcx-ddddxxxx`)와 VPC 피어링 연결이 있습니다. VPC A와 VPC X에 겹치는 CIDR 블록이 있습니다. 즉, VPC A와 VPC D 사이의 피어링 트래픽이 VPC D의 특정 서브넷(서브넷 1)으로 제한됩니다. 따라서 VPC D에서는 VPC A 또는 VPC X에서 요청을 받으면 올바른 VPC로 응답 트래픽을 보낼 수 있습니다. AWS에서는 패킷의 소스 IP를 확인하고 회신 패킷을 소스로 다시 라우팅하는 VPC 피어링 연결의 유니캐스트 역경로 전달이 지원되지 않습니다. 자세한 내용은 응답 트래픽을 위한 라우팅 단원을 참조하세요.

마찬가지로, VPC D와 VPC Z에 겹치는 CIDR 블록이 있습니다. VPC D와 VPC X 사이의 피어링 트래픽이 VPC D의 서브넷 2로 제한되며 VPC X와 VPC Z 사이의 피어링 트래픽이 VPC Z의 서브넷 1로 제한됩니다. 따라서 VPC X에서 VPC D 또는 VPC Z의 피어링 트래픽을 받으면 올바른 VPC로 응답 트래픽을 다시 보낼 수 있습니다.



VPC B, C, E, F 및 G의 라우팅 테이블에서는 VPC A의 전체 CIDR 블록에 액세스하는 관련 피어링 연결을 가리키고, VPC A 라우팅 테이블에서는 전체 CIDR 블록에 액세스하는 VPC B, C, E, F 및 G의 관련 피어링 연결을 가리킵니다. 피어링 연결 `pcx-aaaadddd`의 경우, VPC A 라우팅 테이블에서는 VPC D의 서브넷 1로만 트래픽을 라우팅하고 VPC D의 서브넷 1 라우팅 테이블에서는 VPC A의 전체 CIDR 블록을 가리킵니다.

VPC Y 라우팅 테이블에서는 VPC 와 VPC Z의 전체 CIDR 블록에 액세스하는 과련 피어링 연결을 가리키고, VPC Z 라우팅 테이블에서는 VPC Y의 전체 CIDR 블록에 액세스하는 관련 피어링 연결을 가리킵니다. VPC Z의 서브넷 1 라우팅 테이블에서는 VPC Y의 전체 CIDR 블록에 액세스하는 관련 피어링 연결을 가리킵니다. VPC X 라우팅 테이블에서는 VPC D의 서브넷 2와 VPC Z의 서브넷 1에 액세스하는 관련 피어링 연결을 가리킵니다.

라우팅 테이블	대상 주소	대상
VPC A	*VPC A CIDR*	로컬
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
	VPC D# ### 1 CIDR	pcx-aaaadddd
	VPC E CIDR	pcx-aaaaeeee
	VPC F CIDR	pcx-aaaaffff

라우팅 테이블	대상 주소	대상
	VPC G CIDR	pcx-aaaagggg
VPC B	*VPC B CIDR*	로컬
	VPC A CIDR	pcx-aaaabbbb
VPC C	*VPC C CIDR*	로컬
	VPC A CIDR	pcx-aaaacccc
VPA D의 서브넷 1	*VPC D CIDR*	로컬
	VPC A CIDR	pcx-aaaadddd
VPA D의 서브넷 2	*VPC D CIDR*	로컬
	VPC X CIDR	pcx-ddddxxxx
VPC E	*VPC E CIDR*	로컬
	VPC A CIDR	pcx-aaaaeeee
VPC F	*VPC F CIDR*	로컬
	VPC A CIDR	pcx-aaaaffff
VPC G	*VPC G CIDR*	로컬
	VPC A CIDR	pcx-aaaagggg
VPC X	*VPC X CIDR*	로컬
	VPC D# ### 2 CIDR	pcx-ddddxxxx
	VPC Y CIDR	pcx-xxxxyyyy
	VPC Z# ### 1 CIDR	pcx-xxxxzzzz
VPC Y	*VPC Y CIDR*	로컬

라우팅 테이블	대상 주소	대상
	VPC X CIDR	pcx-xxxxyyyy
	VPC Z CIDR	pcx-yyyyzzzz
VPC Z	*VPC Z CIDR*	로컬
	VPC Y CIDR	pcx-yyyyzzzz
	VPC X CIDR	pcx-xxxxzzzz

VPC 피어링 연결 네트워킹 시나리오

자신의 VPC 사이 또는 자신이 소유하는 VPC와 다른 AWS 계정의 VPC 사이에 VPC 피어링 연결을 설정해야 할 여러 가지 이유가 있습니다. 다음 시나리오는 어떤 구성이 자신의 네트워킹 요구 사항에 가장 적합한지 결정하는 데 도움이 될 수 있습니다.

시나리오

- 리소스에 대한 모든 권한을 제공하기 위한 두 개 이상의 VPC 피어링
- 중앙 집중식 리소스에 액세스하기 위해 한 VPC에 피어링

리소스에 대한 모든 권한을 제공하기 위한 두 개 이상의 VPC 피어링

이 시나리오에서는 모든 VPC 간에 리소스를 완전히 공유할 수 있도록 피어링할 VPC가 두 개 이상 있습니다. 다음은 몇 가지 예시입니다.

- 회사에 재무부서용 VPC가 하나 있고, 회계부서용 VPC도 따로 하나가 있습니다. 재무부서로서는 회계부서에 있는 모든 리소스에 액세스할 필요가 있고, 회계부서 역시 재무부서의 모든 리소스에 액세스할 필요가 있습니다.
- 회사에 여러 IT 부서가 있는데, 각 부서마다 자체 VPC가 있습니다. 일부 VPC는 동일한 AWS 계정 내에 있고, 다른 VPC는 상이한 AWS 계정에 있습니다. IT 부서들이 서로의 리소스에 대해 모든 권한을 가질 수 있도록 모든 VPC를 함께 피어링하려고 합니다.

이 시나리오에 적합한 VPC 피어링 연결 구성 및 라우팅 테이블 설정 방법에 관한 자세한 내용은 다음 문서를 참조하세요.

- 함께 피어링된 두 개의 VPC
- 함께 피어링된 세 개의 VPC
- 서로 피어링된 여러 VPC

Amazon VPC 콘솔에서 VPC 피어링 연결을 생성하고 사용하는 방법에 대한 자세한 내용은 VPC 피어링 연결을 참조하세요.

중앙 집중식 리소스에 액세스하기 위해 한 VPC에 피어링

이 시나리오에서는 다른 VPC와 공유하려는 리소스가 포함되어 있는 중앙 VPC가 하나 있습니다. 중앙 VPC는 피어 VPC에 대한 모든 권한 또는 일부 권한이 필요할 수 있으며, 마찬가지로 피어 VPC 역시 중앙 VPC에 대해 모든 권한 또는 일부 권한이 필요할 수 있습니다. 다음은 몇 가지 예시입니다.

- 회사의 IT 부서에 파일 공유를 위한 VPC가 있습니다. 다른 VPC를 중앙 VPC에 피어링하고 싶지만, 다른 VPC가 서로 트래픽을 보내도록 하고 싶지는 않습니다.
- 회사에 고객과 공유하려는 VPC가 있습니다. 각각의 고객이 VPC와의 VPC 피어링 연결을 생성할 수 있지만, 고객은 회사 VPC에 피어링되어 있는 다른 VPC로 트래픽을 라우팅할 수 없고 다른 고객의 라우팅을 인식하지도 못합니다.
- Active Directory 서비스에서 사용하는 중앙 VPC가 있습니다. 피어 VPC의 특정 인스턴스가 Active Directory 서버로 요청을 보내고 중앙 VPC에 대한 모든 권한이 필요합니다. 중앙 VPC는 피어 VPC에 대해 모든 권한을 요구하지 않으며, 단지 특정 인스턴스로 응답 트래픽을 라우팅만 하면 됩니다.

Amazon VPC 콘솔에서 VPC 피어링 연결을 생성하고 사용하는 방법에 대한 자세한 내용은 VPC 피어링 연결을 참조하세요.

VPC 피어링에 대한 자격 증명 및 액세스 관리

기본적으로, 사용자는 VPC 피어링 연결을 생성하거나 수정할 수 없습니다. VPC 피어링 리소스에 대한 액세스 권한을 부여하려면 IAM 정책을 IAM 아이덴티티(예: 역할)에 연결하세요.

예시

- 예시: VPC 피어링 연결 생성
- 예시: VPC 피어링 연결 수락
- 예시: VPC 피어링 연결 삭제
- 예시: 특정 계정 내 작업
- 예시: 콘솔을 사용하여 VPC 피어링 연결 관리

Amazon VPC 작업 목록과 각 작업에 대해 지원되는 리소스 및 조건 키는 Service Authorization Reference(서비스 승인 참조)의 Actions, resources, and condition keys for Amazon EC2(Amazon EC2에 사용되는 작업, 리소스 및 조건 키)를 참조하세요.

예시: VPC 피어링 연결 생성

다음과 같은 정책을 통해 `Purpose=Peering`으로 태그가 지정된 VPC를 사용하여 VPC 피어링 연결 요청을 생성하는 권한이 사용자에게 부여됩니다. 첫 번째 설명문은 조건 키(`ec2:ResourceTag`)를 VPC 리소스에 적용합니다. `CreateVpcPeeringConnection` 작업을 위한 VPC 리소스는 항상 요청자 VPC입니다.

두 번째 명령문을 통해 사용자에게 VPC 피어링 연결 리소스를 생성하는 권한이 부여됩니다. 따라서 특정 리소스 ID 대신에 와일드카드(*)를 사용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id:vpc/*",
```

```
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Peering"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id:vpc-peering-connection/*"
    }
  ]
}
```

다음과 같은 정책을 통해 지정된 리전의 VPC를 사용하여 VPC 피어링 연결을 생성하는 권한이 지정된 AWS 계정의 사용자에게 부여되지만, 피어링 연결을 수락하는 VPC가 특정 계정의 특정 VPC인 경우에만 해당합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id-1:vpc/*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id-1:vpc-peering-connection/*",
      "Condition": {
        "ArnEquals": {
          "ec2:AccepterVpc": "arn:aws:ec2:region:account-id-2:vpc/vpc-id"
        }
      }
    }
  ]
}
```

예시: VPC 피어링 연결 수락

다음과 같은 정책을 통해 특정 AWS 계정의 VPC 피어링 연결 요청을 수락하는 권한이 사용자에게 부여됩니다. 이는 사용자가 알 수 없는 계정의 VPC 피어링 연결 요청을 허용하지 못하게 하는 데 도움이 됩니다. 설명문에서는 `ec2:RequesterVpc` 조건 키를 사용하여 이를 적용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id-1:vpc-peering-connection/*",
      "Condition": {
        "ArnEquals": {
          "ec2:RequesterVpc": "arn:aws:ec2:region:account-id-2:vpc/*"
        }
      }
    }
  ]
}
```

다음과 같은 정책을 통해 VPC에 `Purpose=Peering` 태그가 있으면 VPC 피어링 요청을 수락하는 권한이 사용자에게 부여됩니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement":[
    {
      "Effect": "Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Peering"
```

```
        }
      }
    }
  ]
}
```

예시: VPC 피어링 연결 삭제

다음과 같은 정책을 통해 동일한 계정에 있는 지정된 VPC를 사용하는 경우를 제외한 VPC 피어링 연결을 삭제하는 권한이 지정된 권한의 사용자에게 부여됩니다. VPC가 원래 VPC 피어링 연결 요청에서 요청자 VPC 또는 피어 VPC였을 수 있으므로, 이 정책에서는 `ec2:AccepterVpc` 및 `ec2:RequesterVpc` 조건 키를 모두 지정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action": "ec2:DeleteVpcPeeringConnection",
      "Resource": "arn:aws:ec2:region:account-id:vpc-peering-connection/*",
      "Condition": {
        "ArnNotEquals": {
          "ec2:AccepterVpc": "arn:aws:ec2:region:account-id:vpc/vpc-id",
          "ec2:RequesterVpc": "arn:aws:ec2:region:account-id:vpc/vpc-id"
        }
      }
    }
  ]
}
```

예시: 특정 계정 내 작업

다음과 같은 정책을 통해 특정 계정 내 VPC 피어링 연결을 연동하는 권한이 사용자에게 부여됩니다. VPC 피어링 연결이 모두 동일한 AWS 계정 내에 있는 경우, 사용자는 VPC 피어링 연결 보기, 생성, 허용, 거부 및 삭제를 수행할 수 있습니다.

첫 번째 명령문을 통해 모든 VPC 피어링 연결을 보는 권한이 사용자에게 부여됩니다. 이 API 작업(DescribeVpcPeeringConnections)은 현재 리소스 수준 권한을 지원하지 않으므로, 이 경우에는 Resource 요소에 * 와일드카드가 필요합니다.

두 번째 명령문을 통해 VPC 피어링 연결을 생성하고 이를 위해 지정된 계정의 모든 VPC에 액세스하는 권한이 사용자에게 부여됩니다.

세 번째 명령문에서는 Action 요소의 일부로 와일드카드(*)를 사용하여 모든 VPC 피어링 연결 작업에 대한 권한을 부여합니다. 조건 키는 계정의 일부인 VPC와의 VPC 피어링 연결 시에만 이런 작업을 수행할 수 있도록 합니다. 예를 들어 수락자 또는 요청자 VPC가 상이한 계정에 있는 경우에는 사용자가 VPC 피어링 연결을 삭제할 수 없습니다. 사용자는 다른 계정에 있는 VPC와의 VPC 피어링 연결을 생성할 수 없습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeVpcPeeringConnections",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action":
 ["ec2:CreateVpcPeeringConnection","ec2:AcceptVpcPeeringConnection"],
      "Resource": "arn:aws:ec2:*:account-id:vpc/*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcPeeringConnection",
      "Resource": "arn:aws:ec2:*:account-id:vpc-peering-connection/*",
      "Condition": {
        "ArnEquals": {
          "ec2:AccepterVpc": "arn:aws:ec2:*:account-id:vpc/*",
          "ec2:RequesterVpc": "arn:aws:ec2:*:account-id:vpc/*"
        }
      }
    }
  ]
```

```
}
```

예시: 콘솔을 사용하여 VPC 피어링 연결 관리

Amazon VPC 콘솔에서 VPC 피어링 연결을 보려면 사용자에게 `ec2:DescribeVpcPeeringConnections` 작업을 사용할 권한이 있어야 합니다. 피어링 연결 생성 페이지를 사용하려면 사용자에게 `ec2:DescribeVpcs` 작업을 사용할 권한이 있어야 합니다. VPC를 보고 선택하는 권한이 이를 통해 부여됩니다. `ec2:DescribeVpcPeeringConnections`를 제외한 모든 `ec2:*PeeringConnection` 작업에 리소스 수준 권한을 적용할 수 있습니다.

다음과 같은 정책을 통해 VPC 피어링 연결을 보고 Create VPC Peering Connection(VPC 피어링 연결 생성) 대화 상자를 사용하여 특정 요청자 VPC만 사용하는 VPC 피어링 연결을 생성하는 권한을 사용자에게 부여합니다. 사용자가 다른 요청자 VPC로 VPC 피어링 연결을 생성하려고 하면 요청이 실패합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action": [
        "ec2:DescribeVpcPeeringConnections", "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect":"Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": [
        "arn:aws:ec2:*:*:vpc/vpc-id",
        "arn:aws:ec2:*:*:vpc-peering-connection/*"
      ]
    }
  ]
}
```

계정의 VPC 피어링 연결 할당량

VPC 피어링으로 두 VPC를 연결할 수 있습니다. 이렇게 하면 한 VPC의 리소스가 동일한 네트워크에 속한 것처럼 다른 VPC의 리소스와 통신할 수 있습니다. VPC 피어링은 VPC가 동일한 AWS 리전에 있든 상이한 리전에 있든 관계없이 VPC를 연결하는 데 유용한 기능입니다. 이 섹션에서는 VPC 피어링 연결을 사용할 때 알아야 할 할당량에 대해 설명합니다.

다음 표에는 AWS 계정에 대한 VPC 피어링 연결의 할당량(이전에는 제한이라고 불렀음)이 나열되어 있습니다. 달리 명시되지 않는 한 이러한 할당량의 증가를 요청할 수 있습니다.

현재 VPC 피어링 연결 요구 사항이 기본 할당량을 초과하는 경우 서비스 제한 증가 요청을 제출하는 것이 좋습니다. 사용 사례를 검토하고 귀하와 협력하여 할당량을 적절히 조정하여 VPC 환경이 증가하는 비즈니스 요구 사항을 지원할 것입니다.

명칭	기본값	조정 가능
VPC당 활성 VPC 피어링 연결	50	예 (최대 125개)
대기중 VPC 피어링 연결 요청	25	예
수락되지 않은 VPC 피어링 연결 요청에 대한 만료 시간	1주(168시간)	아니요

VPC 피어링 연결 사용의 규칙에 대한 자세한 내용은 VPC 피어링 제한을 참조하세요. Amazon VPC 할당량에 대한 추가 정보는 Amazon VPC 사용 설명서의 Amazon VPC 할당량을 참조하세요.

Amazon VPC 피어링 설명서의 문서 기록

다음 표에서는 Amazon VPC 피어링 설명서에 대한 설명서 릴리스를 설명합니다.

변경 사항	설명	날짜
생성 시 태그	VPC 피어링 연결 및 라우팅 테이블을 생성할 때 태그를 추가할 수 있습니다.	2020년 7월 20일
리전 간 피어링	아시아 태평양(홍콩)의 리전 간 VPC 피어링 연결에는 DNS 호스트 이름 확인이 지원됩니다.	2019년 8월 26일
리전 간 피어링	서로 다른 AWS 리전에 있는 VPC 간에는 VPC 피어링 연결을 생성할 수 있습니다.	2017년 11월 29일
VPC 피어링에 대한 DNS 확인 지원	피어 VPC의 인스턴스가 쿼리를 보낼 때 로컬 VPC가 퍼블릭 DNS 호스트 이름을 프라이빗 IP 주소로 확인하도록 설정할 수 있습니다.	2016년 7월 28일
무효 보안 그룹 규칙	보안 그룹이 피어 VPC의 보안 그룹 규칙에서 참조되는지 여부를 식별할 수 있으며, 무효 보안 그룹 규칙을 식별할 수 있습니다.	2016년 5월 12일
VPC 피어링 연결을 통한 ClassicLink 사용	연결된 로컬 EC2-Classic 인스턴스가 피어 VPC의 인스턴스와 통신하거나 반대로 통신하도록 VPC 피어링 연결을 수정할 수 있습니다.	2016년 4월 26일
VPC 피어링	두 VPC 간에 VPC 피어링 연결을 생성하여, 각 VPC의 인스턴	2014년 3월 24일

스가 프라이빗 IP 주소를 사용하여 서로 통신하도록 할 수 있습니다.