aws

사용 설명서

Amazon Virtual Private Cloud



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Virtual Private Cloud: 사용 설명서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon VPC란 무엇인가? 1
특성 1
Amazon VPC 시작하기 2
Amazon VPC 작업 3
Amazon VPC 요금 3
Amazon VPC 작동 방식 6
VPC 및 서브넷 7
기본 VPC와 기본이 아닌 VPC 7
라우팅 테이블 8
인터넷 액세스 8
회사 또는 홈 네트워크에 액세스 9
VPC 및 네트워크 연결 9
AWS 프라이빗 글로벌 네트워크 10
VPC 계획 11
AWS 계정 가입 11
AWS 계정에 가입 11
권한 확인 11
IP 주소 범위 설명 11
가용 영역 선택 12
인터넷 연결 계획 12
VPC 생성 13
애플리케이션 배포 13
IP 주소 지정 14
프라이빗 IPv4 주소 15
퍼블릭 IPv4 주소 15
IPv6 주소 17
퍼블릭 IPv6 주소 17
프라이빗 IPv6 주소 18
고유 IP 주소 사용 19
Amazon VPC IP 주소 관리자 사용 20
VPC CIDR 블록 20
IPv4 VPC CIDR 블록 20
VPC에 IPv4 CIDR 블록 관리 22
IPv4 CIDR 블록 연결 제한 24

IPv6 VPC CIDR 블록 ... 26
서브넷 CIDR 블록 ... 27
IPv4에 대한 서브넷 크기 조정 ... 27
IPv6에 대한 서브넷 크기 조정 ... 28
IPv4 및 IPv6 비교 ... 29
관리형 접두사 목록 ... 30
접두사 목록 개념 및 규칙 ... 31
접두사 목록에 대한 자격 증명 및 액세스 관리 ... 32
고객 관리형 접두사 목록 ... 33
AWS 관리형 접두사 목록 ... 42
접두사 목록으로 AWS 인프라 관리 최적화 ... 45
AWS IP 주소 범위 ... 47
다운로드 ... 48
송신 제어 ... 48
지리적 위치 피드 ... 48
주소 범위 찾기 ... 49
구문 ... 55
알림 구독 ... 60
VPC에 대한 IPv6 지원 ... 62
VPC에 대한 IPv6 지원 추가 ... 63
이중 스택 VPC 예시 ... 67
AWS에서 IPv6 지원 ... 69
IPv6를 지원하는 서비스 ... 69
추가 IPv6 지원 ... 82
자세히 알아보기 ... 83
Virtual Private Cloud ... 84
VPC 기초 ... 85
VPC IP 주소 범위 ... 85
VPC 다이어그램 ... 85
VPC 리소스 ... 86
VPC 구성 옵션 ... 87
기본 VPC ... 88
기본 VPC 구성 요소 ... 89
기본 서브넷 ... 91
기본 VPC와 기본 서브넷 작업 ... 91
VPC 생성 ... 95

VPC 및 기타 VPC 리소스 생성 96
VPC만 생성 97
AWS CLI를 사용하여 VPC 생성 99
VPC의 리소스 시각화 103
CIDR 블록 추가 또는 제거 105
DHCP 옵션 세트 107
DHCP란 무엇인가요? 108
DHCP 옵션 세트 개념 108
DHCP 옵션 세트로 작업 112
DNS 속성 116
Amazon DNS 이해 116
EC2 인스턴스의 DNS 호스트 이름 보기 121
VPC에 대한 DNS 속성 보기 및 업데이트 122
네트워크 주소 사용량 123
NAU 할당량 초과 124
NAU를 계산하는 방법 124
NAU 예시 125
VPC 서브넷 공유 127
공유 서브넷 사전 조건 128
공유 서브넷 작업 128
소유자와 참여자에 대한 청구 및 측정 131
소유자 및 참가자에 대한 책임 및 권한 131
AWS 리소스 및 공유 VPC 서브넷 134
다른 영역으로 VPC 확장 135
AWS Local Zones의 서브넷 136
AWS Wavelength의 서브넷 141
AWS Outposts의 서브넷 144
VPC 삭제 145
콘솔을 사용하여 삭제 146
CLI를 사용하여 삭제 147
콘솔 작업에서 IaC 생성 148
서브넷 150
서브넷 기본 사항 150
서브넷 IP 주소 범위 150
서브넷 유형 151
서브넷 다이어그램 151

서브넷 라우팅 ... 152
서브넷 설정 ... 152
서브넷 보안 ... 153
서브넷 생성 ... 153
서브넷에서 IPv6 CIDR 블록 추가 또는 제거 ... 155
서브넷의 IP 주소 지정 속성 수정 ... 156
서브넷 CIDR 예약 ... 157
콘솔을 사용한 서브넷 CIDR 예약 작업 ... 158
AWS CLI를 사용한 서브넷 CIDR 예약 작업 ... 158
라우팅 테이블 ... 159
라우팅 테이블 개념 ... 160
서브넷 라우팅 테이블 ... 162
게이트웨이 라우팅 테이블 ... 169
라우팅 우선순위 ... 172
라우팅 옵션 예 ... 174
라우팅 테이블 및 라우팅 생성 ... 189
서브넷 라우팅 테이블 관리 ... 191
기본 라우팅 테이블 교체 ... 195
게이트웨이와 라우팅 테이블 연결 ... 196
로컬 경로의 대상 교체 또는 복원 ... 197
고급 라우팅 ... 198
연결 문제 해결 ... 244
미들박스 라우팅 마법사 ... 244
미들박스 라우팅 마법사 사전 조건 ... 245
보안 어플라이언스로 VPC 트래픽 리디렉션 ... 245
미들박스 라우팅 마법사 고려 사항 ... 247
미들박스 시나리오 ... 248
서브넷 삭제 ... 257
VPC 연결 ... 259
인터넷 게이트웨이 ... 260
인터넷 게이트웨이 기본 사항 ... 261
인터넷 게이트웨이 생성 ... 263
인터넷 게이트웨이 삭제 ... 265
외부 전용 인터넷 게이트웨이 ... 267
외부 전용 인터넷 게이트웨이 기본 사항 ... 267
서브넷에 외부 전용 인터넷 액세스 추가 ... 268

NAT 디바이스 ... 271
NAT 게이트웨이 ... 272
NAT 인스턴스 ... 321
NAT 디바이스 비교 ... 333
탄력적 IP 주소 ... 335
탄력적 IP 주소 개념 및 규칙 ... 336
탄력적 IP 주소 사용 시작 ... 338
AWS Transit Gateway ... 346
AWS Virtual Private Network ... 347
VPC 피어링 연결 ... 349
모니터링 ... 350
VPC 흐름 로그 ... 351
흐름 로그 기본 사항 ... 352
흐름 로그 레코드 ... 355
흐름 로그 레코드의 예시 ... 369
흐름 로그 제한 ... 380
가격 책정 ... 384
흐름 로그 작업 ... 384
CloudWatch Logs에 게시 ... 387
Amazon S3에 게시 ... 395
Amazon Data Firehose에 게시 ... 404
Athena를 사용하여 쿼리 ... 411
VPC 흐름 로그 피어링의 ID 및 액세스 관리 ... 415
문제 해결 ... 423
CloudWatch 지표 ... 427
NAU 지표 및 차원 ... 427
NAU 모니터링 활성화 또는 비활성화 ... 430
NAU CloudWatch 경보 예시 ... 431
결제 및 사용량 보고서 ... 432
IP 주소 관리 ... 432
VPC 엔드포인트 ... 433
전송 게이트웨이 ... 434
네트워크 분석 ... 434
트래픽 미러링 ... 435
VPC Lattice ... 435
교차 계정/리전 리소스 ... 436

VPC 네트워크 설명 436
지리적 위치 437
서브넷 438
네트워크 연결 440
보안 제어 444
트래픽 관리 446
관련 리소스 449
보안 450
데이터 보호 451
인터네트워크 트래픽 개인 정보 451
전송 중 VPC 암호화 적용 452
암호화 제어 모드 452
트래픽 흐름의 암호화 상태 모니터링 453
VPC 암호화 제어 제외 항목 454
워크플로 구현 455
VPC 암호화 제어 상태 455
AWS 서비스 지원 및 호환성 456
가격 책정 3
AWS CLI 명령 참조 461
추가 리소스 461
자격 증명 및 액세스 관리 462
대상 462
ID로 인증 463
정책을 사용하여 액세스 관리 464
Amazon VPC가 IAM과 작동하는 방식 465
정책 예시 469
문제 해결 482
AWS 관리형 정책 483
서비스 연결 역할 사용 486
인프라 보안 491
네트워크 격리 492
네트워크 트래픽 제어 492
보안 그룹 및 네트워크 ACL 비교 493
보안 그룹 494
보안 그룹 기본 사항 496
보안 그룹 예시 497

보안 그룹 규칙 ... 498
기본 보안 그룹 ... 503
보안 그룹 생성 ... 505
보안 그룹 규칙 구성 ... 506
보안 그룹 삭제 ... 508
보안 그룹을 여러 VPC와 연결 ... 509
AWS Organizations와 보안 그룹 공유 ... 512
네트워크 ACL ... 518
네트워크 ACL 기본 사항 ... 519
네트워크 ACL 규칙 ... 520
기본 네트워크 ACL ... 521
사용자 지정 네트워크 ACL ... 523
경로 MTU 검색 ... 528
네트워크 ACL 생성 ... 528
네트워크 ACL 연결 관리 ... 532
네트워크 ACL 삭제 ... 534
예: 서브넷의 인스턴스에 대한 액세스 제어 ... 535
복원력 ... 539
규정 준수 확인 ... 540
VPC 및 서브넷에 대한 퍼블릭 액세스 차단 ... 540
VPC BPA 기본 사항 ... 541
VPC BPA의 영향 평가 및 VPC BPA 모니터링 ... 547
고급 예제 ... 551
모범 사례 ... 605
다른 서비스와 함께 사용 ... 606
AWS PrivateLink ... 607
AWS Network Firewall ... 608
Route 53 Resolver DNS Firewall ... 609
Reachability Analyzer ... 610
예시 ... 612
테스트 환경 ... 613
개요 ... 613
1. VPC 생성 ... 615
2. 애플리케이션 배포 ... 616
3. 구성 테스트 ... 617
4. 정리 ... 617

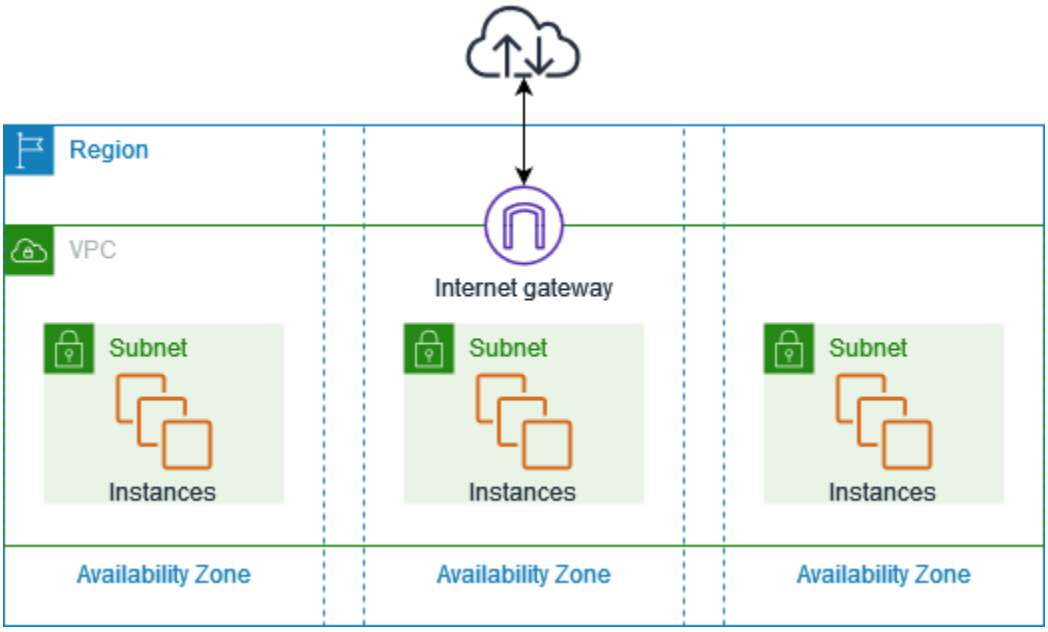
웹 및 데이터베이스 서버 617
개요 617
1. VPC 생성 621
2. 애플리케이션 배포 622
3. 구성 테스트 623
4. 정리 623
프라이빗 서버 623
개요 624
1. VPC 생성 626
2. 애플리케이션 배포 627
3. 구성 테스트 627
4. 정리 628
자습서 629
AWS CLI를 사용하여 Amazon VPC 시작하기 629
사전 조건 630
VPC 생성 630
서브넷 생성 631
인터넷 연결 구성 633
NAT 게이트웨이 생성 634
서브넷 설정 구성 635
보안 그룹 생성 635
VPC 구성 확인 636
EC2 인스턴스 배포 637
문제 해결 212
리소스 정리 640
프로덕션으로 이동 642
다음 단계 643
AWS CLI를 사용하여 프라이빗 서브넷과 NAT 게이트웨이가 포함된 VPC 생성 643
사전 조건 630
VPC 및 서브넷 생성 645
인터넷 연결 생성 및 구성 647
NAT 게이트웨이 생성 648
Amazon S3용 VPC 엔드포인트 생성 650
보안 그룹 구성 650
EC2 인스턴스용 시작 템플릿 생성 652
로드 밸런서 및 대상 그룹 생성 653

Auto Scaling 그룹 생성 654
구성 테스트 655
리소스 정리 640
다음 단계 643
할당량 658
VPC 및 서브넷 658
DNS 659
탄력적 IP 주소 659
게이트웨이 659
고객 관리형 접두사 목록 660
네트워크 ACL 661
네트워크 인터페이스 661
라우팅 테이블 662
라우팅 서버 663
보안 그룹 664
VPC 서브넷 공유 665
네트워크 주소 사용량 666
Amazon EC2 API 스로틀링 666
추가 할당량 리소스 666
문서 기록 668

Amazon VPC란 무엇인가?

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

다음 다이어그램에서는 VPC 예를 보여 줍니다. VPC에는 리전의 각 가용성 영역에 하나의 서브넷이 있고, 각 서브넷에 EC2 인스턴스가 있고, VPC의 리소스와 인터넷 간의 통신을 허용하는 인터넷 게이트웨이가 있습니다.



자세한 내용은 Amazon Virtual Private Cloud(Amazon VPC)를 참조하십시오.

특성

다음 기능은 애플리케이션에 필요한 연결을 제공하도록 VPC 구성하는 데 도움이 됩니다.

Virtual Private Cloud(VPC)

VPC는 자체 데이터 센터에서 운영하는 기존 네트워크와 아주 유사한 가상 네트워크입니다. VPC를 생성한 후 서브넷을 추가할 수 있습니다.

서브넷

서브넷은 VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 상주해야 합니다. 서브넷을 추가한 후에는 VPC에 AWS 리소스 배포할 수 있습니다.

IP 주소 지정

VPC와 서브넷에 IP 주소를 IPv4와 IPv6 모두 할당할 수 있습니다. 또한 퍼블릭 IPv4 주소 및 IPv6 GUA 주소를 AWS로 가져오고 VPC의 리소스(예: EC2 인스턴스, NAT 게이트웨이, Network Load Balancer)에 할당할 수 있습니다.

라우팅

라우팅 테이블을 사용하여 서브넷 또는 게이트웨이의 네트워크 트래픽이 전달되는 위치를 결정합니다.

게이트웨이 및 엔드포인트

게이트웨이는 VPC를 다른 네트워크에 연결합니다. 예를 들면, 인터넷 게이트웨이를 사용하여 VPC를 인터넷에 연결합니다. VPC 엔드포인트를 사용하여 인터넷 게이트웨이 또는 NAT 장치를 사용하지 않고 AWS 서비스에 비공개로 연결합니다.

피어링 연결

VPC 피어링 연결을 사용하여 두 VPC의 리소스 간 트래픽을 라우팅합니다.

트래픽 미러링

네트워크 인터페이스에서 네트워크 트래픽을 복사하고 심층 패킷 검사를 위해 보안 및 모니터링 어플라이언스로 전송합니다.

전송 게이트웨이

중앙 허브 역할을 하는 전송 게이트웨이를 사용하여 VPC, VPN 연결 및 Direct Connect 연결 간에 트래픽을 라우팅합니다.

VPC 흐름 로그

흐름 로그는 VPC의 네트워크 인터페이스로 들어오고 나가는 IP 트래픽에 대한 정보를 캡처합니다.

VPN 연결

AWS Virtual Private Network(Site-to-Site VPN)을 사용하여 온프레미스 네트워크에 VPC를 연결합니다.

Amazon VPC 시작하기

AWS 계정의 각 에는 기본 VPCAWS 리전가 있습니다. 기본 VPC는 EC2 인스턴스 시작 및 연결을 즉시 시작할 수 있도록 구성되어 있습니다. 자세한 정보는 VPC 계획를 참조하십시오.

필요한 서브넷, IP 주소, 게이트웨이 및 라우팅으로 추가 VPC를 생성하도록 선택할 수 있습니다. 자세한 내용은 the section called “VPC 생성” 섹션을 참조하세요.

Amazon VPC 작업

다음 인터페이스 중 하나를 사용하여 VPC를 생성하고 관리할 수 있습니다.

- AWS Management Console — VPC에 액세스할 때 사용할 수 있는 웹 인터페이스를 제공합니다.
- AWS Command Line Interface(AWS CLI) - Amazon VPC를 포함한 다양한 AWS 서비스에서 사용되는 명령을 제공하며 Windows, macOS 및 Linux에서 지원됩니다. 자세한 내용은 AWS Command Line Interface를 참조하세요.
- AWS SDK — 언어별 API를 제공하고, 서명 계산, 요청 재시도 처리 및 오류 처리와 같은 많은 연결 세부 정보를 관리합니다. 자세한 정보는 AWS SDK를 참조하세요.
- 쿼리 API — HTTPS 요청을 사용하여 호출하는 하위 수준의 API 작업을 제공합니다. 쿼리 API 사용이 Amazon VPC에 액세스하는 가장 직접적인 방법이지만, 애플리케이션에서 요청에 서명할 해시 생성 및 오류 처리와 같은 하위 수준의 세부 정보를 처리해야 합니다. 자세한 내용은 Amazon EC2 API Reference(Amazon EC2 API 참조)의 Amazon VPC actions(Amazon VPC 작업)를 참조하세요.

Amazon VPC 요금

VPC 사용에 따르는 추가 요금은 없습니다. 단 NAT 게이트웨이, IP 주소 관리자, 트래픽 미러링, Reachability Analyzer, Network Access Analyzer와 같은 일부 VPC 구성 요소에 대해 요금이 부과됩니다. 자세한 내용은 Amazon VPC 요금을 참조하세요.

Virtual Private Cloud(VPC)에서 시작하는 거의 모든 리소스는 연결을 위한 IP 주소와 함께 제공됩니다. VPC에 있는 대부분의 리소스는 프라이빗 IPv4 주소를 사용합니다. 하지만 IPv4를 통해 인터넷에 직접 액세스해야 하는 리소스는 퍼블릭 IPv4 주소를 사용합니다.

Amazon VPC에서는 VPC를 미리 설정하지 않고도 Elastic Load Balancing, Amazon RDS, Amazon EMR과 같은 관리형 서비스를 시작할 수 있습니다. 계정에 기본 VPC가 있는 경우 해당 VPC가 사용됩니다. 관리형 서비스가 계정에 프로비저닝한 모든 퍼블릭 IPv4 주소에 요금이 부과됩니다. 이러한 요금은 AWS Cost and Usage Report의 Amazon VPC 서비스와 연결됩니다.

퍼블릭 IPv4 주소 요금

퍼블릭 IPv4 주소는 인터넷에서 라우팅할 수 있는 IPv4 주소입니다. 퍼블릭 IPv4 주소는 인터넷에서 IPv4를 통해 리소스에 직접 연결하는 데 필요합니다.

기존 또는 신규 AWS 프리 티어 고객은 750시간 동안 EC2 서비스에서 퍼블릭 IPv4 주소를 무료로 사용할 수 있습니다. EC2 서비스를 AWS 프리 티어로 사용하지 않는 경우에는 퍼블릭 IPv4 주소에 요금이 부과됩니다. 구체적인 요금 정보는 Amazon VPC 요금의 퍼블릭 IPv4 주소 탭을 참조하세요.

프라이빗 IPv4 주소(RFC 1918)에는 요금이 부과되지 않습니다. 공유 VPC의 퍼블릭 IPv4 주소에 요금이 부과되는 방식에 대한 자세한 내용은 소유자 및 참여자에 대한 청구 및 측정을 참조하세요.

퍼블릭 IPv4 주소에는 다음 유형이 있습니다.

- 탄력적 IP 주소(EIP): EC2 인스턴스, 탄력적 네트워크 인터페이스 또는 AWS 리소스와 연결할 수 있는 Amazon에서 제공하는 정적 퍼블릭 IPv4 주소입니다.
- EC2 퍼블릭 IPv4 주소: Amazon에서 EC2 인스턴스에 할당한 퍼블릭 IPv4 주소입니다(EC2 인스턴스가 기본 서브넷에서 시작되거나 인스턴스가 퍼블릭 IPv4 주소를 자동으로 할당하도록 구성된 서브넷에서 시작되는 경우).
- BYOIPv4 주소: 고유 IP 주소 가져오기(BYOIP)를 사용하여 AWS로 가져온 IPv4 주소 범위의 퍼블릭 IPv4 주소입니다.
- 서비스 관리형 IPv4 주소: 퍼블릭 IPv4 주소는 AWS 리소스에 자동으로 프로비저닝되고 AWS 서비스에 의해 관리됩니다. 예를 들어 Amazon ECS, Amazon RDS 또는 Amazon WorkSpaces의 퍼블릭 IPv4 주소입니다.

다음 목록은 퍼블릭 IPv4 주소를 사용할 수 있는 가장 일반적인 AWS 서비스를 보여줍니다.

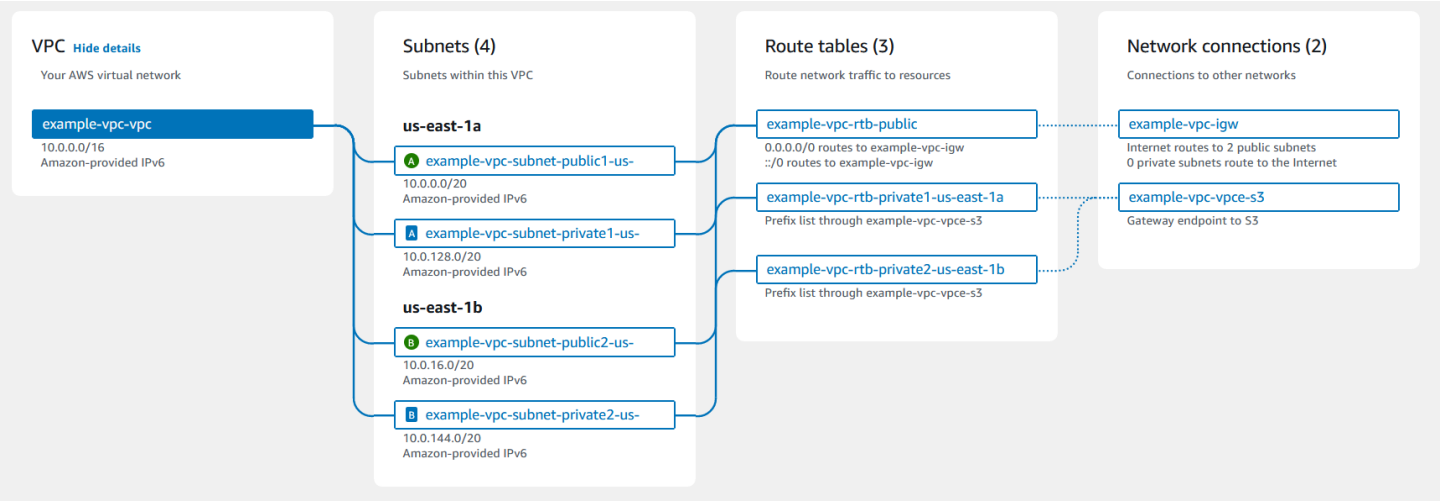
- Amazon WorkSpaces Applications
- AWS Client VPN
- AWS Database Migration Service
- Amazon EC2
- Amazon Elastic Container Service
- Amazon EKS
- Amazon EMR
- Amazon GameLift Servers
- AWS Global Accelerator
- AWS Mainframe Modernization
- Amazon Managed Streaming for Apache Kafka
- Amazon MQ
- Amazon RDS

- Amazon Redshift
- AWS Site-to-Site VPN
- Amazon VPC NAT 게이트웨이
- Amazon WorkSpaces
- Elastic Load Balancing

Amazon VPC 작동 방식

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

다음은 AWS Management Console를 사용하여 VPC를 생성할 때 표시되는 미리 보기 창에서 VPC와 그 리소스를 시각적으로 표현한 것입니다. 기존 VPC의 경우 리소스 맵 탭에서 이 시각화 기능을 사용할 수 있습니다. 이 예제는 VPC 및 기타 네트워킹 리소스 생성을 선택할 때 VPC 생성 페이지에서 처음 선택되는 리소스를 보여 줍니다. 이 VPC는 IPv4 CIDR 및 Amazon에서 제공하는 IPv6 CIDR, 2개의 가용 영역에 있는 서브넷, 3개의 라우팅 테이블, 인터넷 게이트웨이 및 게이트웨이 엔드포인트로 구성됩니다. 인터넷 게이트웨이를 선택했기 때문에 해당 라우팅 테이블이 트래픽을 인터넷 게이트웨이로 전송하므로 퍼블릭 서브넷의 트래픽이 인터넷으로 라우팅되는 것이 시각화에 표시됩니다.



개념

- VPC 및 서브넷
- 기본 VPC와 기본이 아닌 VPC
- 라우팅 테이블
- 인터넷 액세스
- 회사 또는 홈 네트워크에 액세스
- VPC 및 네트워크 연결
- AWS 프라이빗 글로벌 네트워크

VPC 및 서브넷

Virtual Private Cloud(VPC)는 사용자의 AWS 계정 전용 가상 네트워크입니다. VPC는 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 분리되어 있습니다. VPC의 IP 주소 범위를 지정하고 서브넷과 게이트웨이를 추가하고 보안 그룹을 연결합니다.

서브넷은 VPC의 IP 주소 범위입니다. Amazon EC2 인스턴스와 같은 AWS 리소스를 서브넷으로 실행할 수 있습니다. 서브넷을 인터넷, 다른 VPC 및 자체 데이터 센터에 연결하고 라우팅 테이블을 사용하여 서브넷으로/서브넷에서 트래픽을 라우팅할 수 있습니다.

자세히 알아보기

- IP 주소 지정
- Virtual Private Cloud
- 서브넷

기본 VPC와 기본이 아닌 VPC

계정이 2013년 12월 4일 이후에 생성된 경우 각 리전에 기본 VPC가 함께 제공됩니다. 기본 VPC가 구성되어 사용할 준비가 되었습니다. 예를 들어, 리전의 각 가용 영역에 기본 서브넷, 연결된 인터넷 게이트웨이, 모든 트래픽을 인터넷 게이트웨이로 보내는 기본 라우팅 테이블의 경로, 퍼블릭 IP 주소가 있는 인스턴스에 퍼블릭 DNS 호스트 이름을 자동으로 할당하고 Amazon 제공 DNS 서버를 통해 DNS 확인을 활성화하는 DNS 설정이 있습니다(VPC의 DNS 속성 섹션 참조). 따라서 기본 서브넷에서 시작된 EC2 인스턴스는 자동으로 인터넷에 액세스할 수 있습니다. 리전에 기본 VPC가 있고 해당 리전에서 EC2 인스턴스를 시작할 때 서브넷을 지정하지 않으면 기본 서브넷 중 하나가 선택되고 해당 서브넷에서 인스턴스가 시작됩니다.

자체 VPC를 생성하고 필요에 따라 구성할 수도 있습니다. 이를 기본이 아닌 VPC라고 합니다. 기본이 아닌 VPC에 만든 서브넷과 기본 VPC에 만든 추가 서브넷은 기본이 아닌 서브넷이라고 합니다.

자세히 알아보기

- the section called “기본 VPC”
- the section called “VPC 생성”

라우팅 테이블

라우팅 테이블에는 VPC의 네트워크 트래픽을 전달할 위치를 결정하는 데 사용되는 라우팅이라는 규칙 집합이 포함되어 있습니다. 서브넷을 특정 라우팅 테이블과 명시적으로 연결할 수 있습니다. 그렇지 않으면 서브넷이 기본 라우팅 테이블과 암시적으로 연결됩니다.

라우팅 테이블의 각 라우팅은 트래픽을 전달할 IP 주소 범위(대상 주소)와 트래픽을 전송할 게이트웨이, 네트워크 인터페이스 또는 연결(대상)을 지정합니다.

자세히 알아보기

- 라우팅 테이블 구성

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 하지만 인터넷으로부터의 원치 않는 인바운드 연결은 차단하려면 네트워크 주소 변환(NAT) 디바이스를 사용하면 됩니다. NAT는 여러 개의 프라이빗 IPv4 주소를 하나의 퍼블릭 IPv4 주소에 매핑합니다. 탄력적 IP 주소로 NAT 디바이스를 구성하고 인터넷 게이트웨이를 통해 인터넷에 연결할 수 있습니다. 프라이빗 서브넷의 인스턴스를 NAT 디바이스를 통해 인터넷에 연결할 수 있으며, 이렇게 하면 인스턴스의 트래픽이 인터넷 게이트웨이로 라우팅되고, 모든 응답은 인스턴스로 라우팅됩니다.

IPv6 CIDR 블록을 VPC와 연결하고 인스턴스에 IPv6 주소를 할당하면 인스턴스가 IPv6으로 인터넷 게이트웨이를 통해 인터넷에 연결할 수 있습니다. 또는 인스턴스는 외부 전용 인터넷 게이트웨이를 사용

하여 IPv6를 통해 인터넷에 대한 아웃바운드 연결을 시작할 수 있습니다. IPv6 트래픽은 IPv4 트래픽에서 분리되어 있으므로, 라우팅 테이블에는 IPv6 트래픽에 대한 별도의 경로가 포함되어야 합니다.

자세히 알아보기

- 인터넷 게이트웨이를 사용하여 VPC에 대한 인터넷 액세스 활성화
- 송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽 활성화
- NAT 디바이스를 사용하여 인터넷 또는 다른 네트워크에 연결

회사 또는 홈 네트워크에 액세스

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

Site-to-Site VPN 연결은 AWS 측의 가상 프라이빗 게이트웨이 또는 전송 게이트웨이와 데이터 센터의 고객 게이트웨이 디바이스 간 두 개의 VPN 터널로 구성됩니다. 고객 게이트웨이는 Site-to-Site VPN 연결에서 고객 측이 구성하는 물리적 디바이스 또는 소프트웨어 애플리케이션입니다.

자세히 알아보기

- AWS Site-to-Site VPN 사용자 안내서
- Amazon VPC Transit Gateway

VPC 및 네트워크 연결

두 VPC 간에 VPC 피어링 연결을 생성하여 비공개적으로 두 VPC 간에 트래픽을 라우팅할 수 있습니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신할 수 있습니다.

또한 전송 게이트웨이를 생성하고 사용해 VPC와 온프레미스 네트워크를 상호 연결할 수 있습니다. 전송 게이트웨이는 VPC, VPN 연결, Direct Connect 게이트웨이, 전송 게이트웨이 피어링 연결 등 연결 간에 이동하는 트래픽에 대해 리전 가상 라우터 역할을 합니다.

자세히 알아보기

- Amazon VPC Peering Guide
- Amazon VPC Transit Gateway

AWS 프라이빗 글로벌 네트워크

AWS는 고객의 네트워킹 요구 사항을 지원하는 안전한 클라우드 컴퓨팅 환경을 제공하기 위해 고성능, 낮은 대기 시간의 프라이빗 글로벌 네트워크를 운영합니다.AWS 리전은 여러 인터넷 서비스 제공업체(ISP)와 연결되는 것은 물론 프라이빗 글로벌 네트워크 백본과도 연결되어 고객으로부터 전송되는 교차 리전 트래픽을 향상된 네트워크 성능으로 처리합니다.

AWS 네트워크에서 출발하여 AWS 네트워크를 목적지로 하는 패킷은 AWS 중국 리전 및 AWS European Sovereign Cloud 리전으로 송수신되는 트래픽을 제외하고 AWS 글로벌 네트워크 내에 유지됩니다. 대상이 프라이빗 IP 주소이든 퍼블릭 IP 주소이든 마찬가지입니다. 예를 들어 두 VPC의 EC2 인스턴스가 퍼블릭 IP 주소를 사용하여 통신하는 경우 트래픽은 프라이빗 글로벌 네트워크에 유지됩니다. 목적지는 동일한 가용 영역, 동일한 리전 내 다른 가용 영역 또는 다른 리전에 위치할 수 있습니다.

네트워크 패킷 손실은 네트워크 흐름 충돌, 낮은 수준(계층 2) 오류 및 기타 네트워크 오류를 비롯한 여러 요인으로 인해 발생할 수 있습니다. 패킷 손실이 최소화되도록 네트워크가 엔지니어링되고 운영됩니다. 또한 AWS 리전을 연결하는 글로벌 백본에서 PLR(패킷 손실률)을 측정합니다. 백본 네트워크를 운영하여 0.0001% 미만의 시간당 PLR 중 p99를 목표로 합니다.

VPC 계획

VPC 생성 및 연결을 준비하려면 다음 작업를 완료합니다. 작업을 마치면 AWS에서 애플리케이션을 배포할 준비가 됩니다.

작업

- AWS 계정 가입
- 권한 확인
- IP 주소 범위 설명
- 가용 영역 선택
- 인터넷 연결 계획
- VPC 생성
- 애플리케이션 배포

AWS 계정 가입

AWS 계정에 가입

AWS를 시작하려면 AWS 계정이 필요합니다. AWS 계정 생성에 대한 자세한 내용은 AWS Account Management 참조 안내서의 AWS 계정 시작하기를 참조하세요.

권한 확인

Amazon VPC 사용하려면 먼저 필요한 권한이 있어야 합니다. 자세한 내용은 Amazon VPC용 자격 증명 및 액세스 관리 및 Amazon VPC 정책 예시를 참조하세요.

IP 주소 범위 설명

VPC의 리소스는 IP 주소를 사용하여 인터넷을 통해 서로 통신하고 다른 리소스와 통신합니다. VPC와 서브넷을 생성할 때 IP 주소 범위를 선택할 수 있습니다. 서브넷에 EC2 인스턴스와 같은 리소스를 배포하면 서브넷의 IP 주소 범위에서 IP 주소를 받습니다. 자세한 내용은 섹션을 참조하세요IP 주소 지정

VPC의 크기를 선택할 때는 AWS 계정 및 VPC 간에 필요한 IP 주소 수를 고려하세요. VPC의 IP 주소 범위가 자체 네트워크의 IP 주소 범위와 겹치지 않도록 하세요. 여러 VPC 간에 연결이 필요한 경우 겹치는 IP 주소가 없는지 확인해야 합니다.

IP 주소 관리자(IPAM)를 사용하면 애플리케이션의 IP 주소를 보다 쉽게 계획, 추적, 모니터링할 수 있습니다. 자세한 내용은 IP 주소 관리자 설명서를 참조하세요.

가용 영역 선택

AWS 지역은 가용 영역이라고 하는 데이터 센터를 클러스터링하는 물리적 위치입니다. 각 가용 영역은 중복 전원, 네트워킹 및 연결과 함께 독립된 전원, 냉각 및 물리적 보안이 있습니다. 리전의 가용 영역은 유의미한 거리만큼 물리적으로 분리되어 있으며 지연 시간이 짧은 높은 대역폭 네트워킹을 통해 상호 연결됩니다. 여러 가용 영역에서 실행되도록 애플리케이션을 설계하여 내결함성을 더욱 높일 수 있습니다.

프로덕션 환경

프로덕션 환경의 경우 두 개 이상의 가용 영역을 선택하고 각 활성 가용 영역에서 AWS 리소스를 균등하게 배포하는 것이 좋습니다.

개발 또는 테스트 환경

개발 또는 테스트 환경의 경우 하나의 가용 영역에만 리소스를 배포하여 비용을 절감할 수 있습니다.

인터넷 연결 계획

연결 요구 사항에 따라 각 VPC를 서브넷으로 나눌 계획을 세웁니다. 예제:

- 인터넷에 있는 클라이언트로부터 트래픽을 받는 웹 서버가 있는 경우 각 가용 영역에서 해당 서버에 대한 서브넷을 생성합니다.
- 또한 VPC에 있는 다른 서버로부터 트래픽을 받는 서버가 있는 경우 각 가용 영역에서 해당 서버에 대한 별개의 서브넷을 생성합니다.
- 네트워크에 대한 VPN 연결을 통해서만 트래픽을 받는 서버가 있는 경우 각 가용 영역에서 해당 서버에 대한 별개의 서브넷을 생성합니다.

애플리케이션이 인터넷에서 트래픽을 수신하려면 VPC에 인터넷 게이트웨이가 있어야 합니다. VPC에 인터넷 게이트웨이를 연결한다고 해서 인터넷에서 인스턴스에 자동으로 액세스할 수 있는 것은 아닙니다. 인터넷 게이트웨이를 연결하는 것 외에도 서브넷 라우팅 테이블을 인터넷 게이트웨이에 대한 경로로 업데이트해야 합니다. 또한 인스턴스에 퍼블릭 IP 주소와 애플리케이션에 필요한 특정 포트 및 프로토콜을 통한 인터넷의 트래픽을 허용하는 연결된 보안 그룹이 있는지 확인해야 합니다.

또는 인터넷이 연결된 로드 밸런서를 사용하여 인스턴스를 등록합니다. 로드 밸런서는 클라이언트에서 트래픽을 수신하고 해당 트래픽을 하나 이상의 가용 영역에 등록된 인스턴스로 분산합니다. 자세한

내용은 Elastic Load Balancing을 참조하세요. 인터넷에서 요청하지 않은 인바운드 연결을 허용하지 않고 프라이빗 서브넷의 인스턴스가 인터넷에 액세스(예: 업데이트 다운로드)할 수 있도록 하려면 각 활성 가용 영역에 퍼블릭 NAT 게이트웨이를 추가하고 라우팅 테이블을 업데이트하여 인터넷 트래픽을 NAT 게이트웨이로 보냅니다. 자세한 내용은 섹션을 참조하세요the section called “프라이빗 서브넷에서 인터넷 액세스”

VPC 생성

필요한 VPC와 서브넷 수, VPC와 서브넷에 할당할 CIDR 블록, VPC를 인터넷에 연결하는 방법을 결정했으면 이제 VPC를 생성할 준비가 되었습니다. AWS Management Console를 사용하여 VPC를 생성하고 구성에 퍼블릭 서브넷을 포함하면 서브넷에 대한 라우팅 테이블이 생성되고 인터넷에 직접 액세스하는 데 필요한 경로가 추가됩니다. 자세한 내용은 섹션을 참조하세요the section called “VPC 생성”

애플리케이션 배포

VPC를 생성한 후에는 애플리케이션을 배포할 수 있습니다.

프로덕션 환경

프로덕션 환경의 경우 다음 서비스 중 하나를 사용하여 여러 가용 영역에 서버를 배포하고, 애플리케이션에 필요한 최소 서버 수를 유지하도록 조정을 구성하고, 로드 밸런서에 서버를 등록하여 트래픽을 서버 간에 균등하게 분산할 수 있습니다.

- Amazon EC2 Auto Scaling
- EC2 플릿
- Amazon Elastic Container Service(Amazon ECS)

개발 또는 테스트 환경

개발 또는 테스트 환경의 경우 단일 EC2 인스턴스를 시작하도록 선택할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 시작하기를 참조하세요.

VPC 및 서브넷의 IP 주소 지정

IP 주소가 있으면 VPC 내의 리소스가 서로 통신하고, 인터넷을 통해 다른 리소스와 통신할 수 있습니다.

CIDR(Classless Inter-Domain Routing) 표기법은 IP 주소 및 네트워크 마스크를 나타내는 방법입니다. 이러한 주소의 형식은 다음과 같습니다.

- 개별 IPv4 주소는 32비트로, 최대 3개의 십진수(0~255)로 구성된 그룹 4개를 포함합니다. 예를 들어 10.0.1.0입니다.
- IPv4 CIDR 블록에는 IPv4 주소가 있고, 주소 뒤에 슬래시와 0~32 사이의 숫자가 추가됩니다. 예를 들어 10.0.0.0/16은 10.0.0.0에서 10.0.255.255에 이르는 IPv4 주소 65,536개를 나타냅니다.
- 개별 IPv6 주소는 128비트로, 4개의 16진수로 구성된 세그먼트 8개를 포함합니다. 예: 2001:0db8:85a3:0000:0000:8a2e:0370:7334. 세그먼트 앞에 0이 포함될 필요는 없습니다. 주소에서 연속해서 0으로만 이루어진 세그먼트를 이중 콜론(::)으로 바꿀 수도 있습니다. 따라서 예시 주소는 2001:db8:85a3::8a2e:370:7334로 축약할 수 있습니다.
- IPv6 CIDR 블록에는 0으로만 이루어진 세그먼트로 끝나는 IPv6 주소가 있고, 이 세그먼트는 이중 콜론으로 대체되고, 그 뒤에 슬래시와 0~128 사이의 숫자가 추가됩니다. 예를 들어 2001:db8:1234:1a00::/56은 2001:db8:1234:1a00:0000:0000:0000:0000에서 2001:db8:1234:1aff:ffff:ffff:ffff:ffff에 이르는 2^72개의 IPv6 주소를 나타냅니다.

자세한 내용은 CIDR이란 무엇입니까?를 참조하세요.

내용

- 프라이빗 IPv4 주소
- 퍼블릭 IPv4 주소
- IPv6 주소
- 고유 IP 주소 사용
- Amazon VPC IP 주소 관리자 사용
- VPC CIDR 블록
- 서브넷 CIDR 블록
- IPv4 및 IPv6 비교
- 관리형 접두사 목록으로 네트워크 CIDR 블록 통합 및 관리
- AWS IP 주소 범위

- VPC에 대한 IPv6 지원
- IPv6를 지원하는 AWS 서비스

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(예: eth0)에 할당됩니다. 또한 각 인스턴스에는 인스턴스의 프라이빗 IP 주소를 확인하는 프라이빗(내부) DNS 호스트 이름이 할당됩니다. 호스트 이름은 리소스 기반 또는 IP 기반의 두 가지 유형이 될 수 있습니다. 자세한 내용은 EC2 인스턴스 이름 지정을 참조하세요. 주 프라이빗 IP 주소를 지정하지 않으면 서브넷 범위에서 사용 가능한 IP 주소가 선택됩니다. 네트워크 인터페이스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스를 참조하세요.

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

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

AWS는 서브넷의 IPv4 주소 범위를 인터넷에 알리지 않습니다.

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 인스턴스를 위해 생성된 주 네트워크 인터페이스에 할당됩니다. 퍼블릭 IP 주소는 네트워크 주소 변환(NAT)을 통해 주 프라이빗 IP 주소로 매핑됩니다.

> **Note**
>
> AWS에서는 탄력적 IP 주소 및 실행 중인 인스턴스에 연결된 퍼블릭 IPv4 주소를 포함하여 모든 퍼블릭 IPv4 주소에 요금을 부과합니다. 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.

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

- 서브넷의 퍼블릭 IP 주소 지정 속성 수정. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정 단원을 참조하세요.
- 인스턴스를 시작하는 동안 퍼블릭 IP 주소 지정 기능을 활성화하거나 비활성화하면 서브넷의 퍼블릭 IP 주소 지정 속성을 재정의합니다.
- 네트워크 인터페이스와 연결된 IP 주소를 관리하여 시작 후 인스턴스에서 퍼블릭 IP 주소 할당을 취소할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 IP 주소 관리를 참조하세요.

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. 다음과 같은 특정 경우에 인스턴스에서 퍼블릭 IP 주소를 해제하거나 새 인스턴스에 할당합니다. 자세한 내용은 Amazon EC2 사용 설명서의 퍼블릭 IP 주소를 참조하세요.

필요에 따라 인스턴스에 할당하거나 인스턴스에서 제거가 가능한, 계정에 할당된 영구 퍼블릭 IP 주소가 필요한 경우, 그 대신에 탄력적 IP 주소를 사용하십시오. 자세한 내용은 탄력적 IP 주소를 VPC의 리소스와 연결 단원을 참조하세요.

VPC에서 DNS 호스트 이름을 지원하는 경우, 퍼블릭 IP 주소 또는 탄력적 IP 주소를 받는 각 인스턴스에도 퍼블릭 DNS 호스트 이름이 할당됩니다. Amazon은 퍼블릭 DNS 호스트 이름을 인스턴스 네트워크 외부에서는 인스턴스의 퍼블릭 IP 주소로 변환하고 인스턴스 네트워크 내부에서는 인스턴스의 프라이빗 IP 주소로 변환합니다. 자세한 내용은 VPC의 DNS 속성 섹션을 참조하세요.

Amazon VPC IP Address Manager(IPAM)를 사용하는 경우, AWS에서 퍼블릭 연속적 IPv4 주소 블록을 가져와서 AWS 리소스에 순차적 탄력적 IP 주소를 할당하는 데 사용할 수 있습니다. 연속적 IPv4 주소 블록을 사용하면 보안 액세스 제어 목록에 대한 관리 오버헤드를 크게 줄이고 AWS의 규모를 조정하는 기업의 IP 주소 할당 및 추적을 단순화할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀에서 순차적 탄력적 IP 주소 할당을 참조하세요.

IPv6 주소

인터넷이 계속 성장함에 따라 IP 주소에 대한 필요성도 증가하고 있습니다. IP 주소의 가장 일반적인 형식은 IPv4입니다. IP 주소의 새로운 형식은 IPv4보다 더 큰 주소 공간을 제공하는 IPv6입니다. IPv6는 IPv4 주소 고갈 문제를 해결하고 더 많은 장치를 인터넷에 연결할 수 있게 해줍니다. 전환은 점진적으로 이루어지지만 IPv6 채택이 증가함에 따라 네트워크를 간소화하고 IPv6 고급 기능을 활용하여 연결성, 성능 및 보안을 개선할 수 있습니다.

Amazon EC2, Amazon S3, Amazon CloudFront 등 다양한 AWS 서비스에서 듀얼 스택(IPv4 및 IPv6) 또는 IPv6 전용 모드를 지원하므로, 리소스에 IPv6 주소를 할당하고 IPv6 프로토콜을 통해 액세스할 수 있습니다. 이를 통해 IPv6를 도입하는 고객은 네트워크 구성 및 관리를 간소화할 수 있습니다.

다른 서비스에서는 제한적 또는 부분적으로 듀얼 스택 및 IPv6 전용 지원을 제공합니다.

IPv6을 지원하는 서비스에 대한 자세한 내용은 IPv6를 지원하는 AWS 서비스 단원을 참조하세요.

일부 IPv6 주소는 Internet Engineering Task Force에서 예약하고 있다는 점에 유의합니다. 예약된 IPv6 주소 범위에 대한 자세한 정보는 IANA IPv6 Special-Purpose Address Registry 및 RFC4291을 참조하세요.

> **Note**
>
> 퍼블릭 및 프라이빗 IPv6 주소는 모두 AWS에서 사용할 수 있습니다. AWS는 퍼블릭 IP 주소를 AWS에서 인터넷에서 알려진 것으로 정의하지만 프라이빗 IP 주소는 AWS에서 인터넷에 알려질 수 없습니다.

내용

- 퍼블릭 IPv6 주소
- 프라이빗 IPv6 주소

퍼블릭 IPv6 주소

Amazon에서 제공하는 IPv6 주소는 항상 인터넷에 알려져 있습니다. 전역적으로 고유하므로 인터넷으로 접속할 수 있습니다. 서브넷에 대한 라우팅을 제어하거나 보안 그룹 및 네트워크 ACL을 사용함으로써 EC2 인스턴스가 IPv6 주소를 사용하여 접속이 가능한지 여부를 제어할 수 있습니다.

다음은 워크로드에 퍼블릭 IPv6 주소를 사용하기 위해 준비할 수 있는 몇 가지 방법입니다.

- Amazon VPC IP 주소 관리자로 IPAM을 생성하고 Amazon 소유의 퍼블릭 IPv6 주소 범위를 IPAM 주소 풀에 프로비저닝합니다. 자세한 정보는 Amazon VPC IPAM 사용 설명서의 VPC IPv6 풀을 참조하세요.
- IPAM이 있고 퍼블릭 IPv6 주소 범위를 소유하고 있는 경우, 퍼블릭 IPv6 주소 범위의 일부 또는 전부를 IPAM으로 가져오고 퍼블릭 IPv6 주소 범위를 IPAM 주소 풀에 프로비저닝합니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 튜토리얼: IPAM에 IP 주소 가져오기를 참조하세요.
- IPAM은 없지만 퍼블릭 IPv6 주소 범위를 소유하고 있는 경우 퍼블릭 IPv6 주소 범위 중 일부 또는 전체를 AWS로 가져옵니다. 자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 Amazon EC2에 고유 IP 주소 가져오기(BYOIP)를 참조하세요.

퍼블릭 IPv6 주소를 사용할 준비가 되면 인스턴스에 퍼블릭 IPv6 주소를 할당하고(Amazon EC2 사용 설명서의 IPv6 주소 참조), VPC에 퍼블릭 IPv6 CIDR 블록을 할당하고(VPC에서 CIDR 블록 추가 또는 제거 참조), 서브넷에 IPv6 CIDR 블록을 연결할 수 있습니다(서브넷의 IP 주소 지정 속성 수정 참조).

프라이빗 IPv6 주소

프라이빗 IPv6 주소는 AWS에서 인터넷에 알리지 않고 알릴 수 없는 IPv6 주소입니다.

프라이빗 네트워크가 IPv6를 지원하도록 하고 이 주소에서 인터넷으로 트래픽을 라우팅할 의도가 없는 경우 프라이빗 IPv6 주소를 사용할 수 있습니다. 프라이빗 IPv6 주소가 있는 리소스에서 인터넷에 연결하려는 경우 가능하지만, 이렇게 하려면 퍼블릭 IPv6 주소를 가진 다른 서브넷의 리소스를 통해 트래픽을 라우팅해야 합니다.

프라이빗 IPv6 주소에는 두 가지 유형이 있습니다.

- IPv6 ULA 범위: RFC4193 내에 정의된 IPv6 주소. 이러한 주소 범위는 항상 "fc" 또는 "fd"로 시작하므로 쉽게 식별할 수 있습니다. 유효한 IPv6 ULA 공간은 Amazon 예약 범위 fd00::/16과 겹치지 않는 fd00::/8 미만의 모든 공간입니다.
- IPv6 GUA 범위: RFC3587 내에 정의된 IPv6 주소. IPv6 GUA 범위를 프라이빗 IPv6 주소로 사용하는 옵션은 기본적으로 비활성화되며 사용하려면 먼저 활성화해야 합니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 프라이빗 IPv6 GUA CIDR 프로비저닝 활성화를 참조하세요.

다음 사항에 유의하세요.

- 프라이빗 IPv6 주소는 Amazon VPC IP Address Manager(IPAM)를 통해서만 사용할 수 있습니다. IPAM에서는 IPv6 ULA 및 GUA 주소가 있는 리소스를 검색하고 풀에서 겹치는 IPv6 ULA 및 GUA 주소 공간을 모니터링합니다.

- 프라이빗 IPv6 GUA 범위를 사용하는 경우, 귀하가 소유한 IPv6 GUA 범위를 사용하도록 요구합니다.
- 프라이빗 IPv6 주소는 AWS에 의해 인터넷에 알리지 않으며 알릴 수도 없습니다. AWS는 VPC에 인터넷 게이트웨이 또는 이그레스 전용 인터넷 게이트웨이가 있더라도 프라이빗 IPv6 범위에서 퍼블릭 인터넷으로 직접 송신하는 것을 허용하지 않습니다. 프라이빗 IPv6 주소는 인터넷 게이트웨이 엣지에서 자동으로 삭제되므로 공개적으로 라우팅되지 않습니다.
- AWS는 처음 4개의 서브넷 프라이빗 IPv6 주소와 마지막 1개의 주소를 보유합니다.
- 프라이빗 IPv6 ULA의 유효한 범위는 fd80::/9로 시작하는 /9에서 /60까지입니다.
- VPC에 프라이빗 IPv6 GUA 범위를 할당한 경우 동일한 VPC의 프라이빗 IPv6 GUA 공간과 겹치는 퍼블릭 IPv6 GUA 공간을 사용할 수 없습니다.
- 프라이빗 IPv6 ULA 및 GUA 주소 범위를 사용하는 리소스 간 통신이 지원됩니다(예: Direct Connect, VPC 피어링, 트랜짓 게이트웨이 또는 VPN 연결).
- IPv6 전용 및 이중 스택 VPC 서브넷, 탄력적 로드 밸런서 및 AWS Global Accelerator 엔드포인트와 함께 프라이빗 IPv6 주소를 사용할 수 있습니다.
- 프라이빗 IPv6 주소에는 요금이 부과되지 않습니다.

다음은 워크로드에 프라이빗 IPv6 주소를 사용하기 위해 준비할 수 있는 몇 가지 방법입니다.

- Amazon VPC IP 주소 관리자로 IPAM을 생성하고 프라이빗 IPv6 ULA 범위를 IPAM 주소 풀에 프로비저닝합니다. 자세한 정보는 Amazon VPC IPAM 사용 설명서의 VPC IPv6 풀을 참조하세요.
- Amazon VPC IP 주소 관리자로 IPAM을 생성하고 프라이빗 IPv6 GUA 범위를 IPAM 주소 풀에 프로비저닝합니다. IPv6 GUA 범위를 프라이빗 IPv6 주소로 사용하는 옵션은 기본적으로 비활성화되며 사용하려면 먼저 IPAM에서 활성화해야 합니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 프라이빗 IPv6 GUA CIDR 프로비저닝 활성화를 참조하세요.

프라이빗 IPv6 주소를 사용할 준비가 되면 IPAM 풀의 프라이빗 IPv6 CIDR 블록을 VPC에 할당하고(VPC에서 CIDR 블록 추가 또는 제거 참조) IPv6 CIDR 블록을 서브넷에 연결할 수 있습니다(서브넷의 IP 주소 지정 속성 수정 참조).

고유 IP 주소 사용

AWS 계정으로 고유 퍼블릭 IPv4 주소 범위 또는 IPv6 주소 범위의 일부 또는 전체를 가져올 수 있습니다. 주소 범위를 계속해서 소유할 수 있지만 AWS에서는 기본적으로 인터넷에 이러한 주소 범위를 알립니다. 주소 범위를 AWS(으)로 가져오고 나면 이러한 주소가 계정에 주소 풀로 나타납니다. IPv4 주

소 풀에서 탄력적 IP 주소를 생성할 수 있으며, IPv6 주소 풀의 IPv6 CIDR 블록을 VPC와 연결할 수 있습니다.

자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 고유 IP 주소 가져오기(BYOIP)를 참조하세요.

Amazon VPC IP 주소 관리자 사용

Amazon VPC IP Address Manager(IPAM)는 AWS 워크로드의 IP 주소를 보다 쉽게 계획, 추적 및 모니터링할 수 있게 해주는 VPC 기능입니다. IPAM을 통해 특정 비즈니스 규칙을 사용하여 VPC에 IP 주소 CIDR을 할당할 수 있습니다.

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM이란 무엇인가?를 참조하세요.

VPC CIDR 블록

가상 프라이빗 클라우드(VPC)의 IP 주소는 CIDR(Classless Inter-Domain Routing) 표기법을 사용하여 표시됩니다. VPC에는 연결된 IPv4 CIDR 블록이 있어야 합니다. 선택에 따라 추가 IPv4 CIDR 블록과 하나 이상의 IPv6 CIDR 블록을 연결할 수 있습니다. 자세한 내용은 VPC 및 서브넷의 IP 주소 지정 섹션을 참조하세요.

내용

- IPv4 VPC CIDR 블록
- VPC에 IPv4 CIDR 블록 관리
- IPv4 CIDR 블록 연결 제한
- IPv6 VPC CIDR 블록

IPv4 VPC CIDR 블록

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 추가 IPv4 CIDR 블록을 VPC에 연결할 수 있습니다. 자세한 내용은 VPC에서 CIDR 블록 추가 또는 제거 섹션을 참조하세요.

VPC를 생성하는 경우, 다음과 같이 RFC 1918 규격에 따라 프라이빗 IPv4 주소 범위에 속하는 CIDR 블록을 지정하는 것이 좋습니다.

RFC 1918 범위	CIDR 블록의 예
10.0.0.0 - 10.255.255.255 (10/8 접두사)	10.0.0.0/16
172.16.0.0 - 172.31.255.255 (172.16/12 접두사)	172.31.0.0/16
192.168.0.0 - 192.168.255.255 (192.168/16 접두사)	192.168.0.0/20

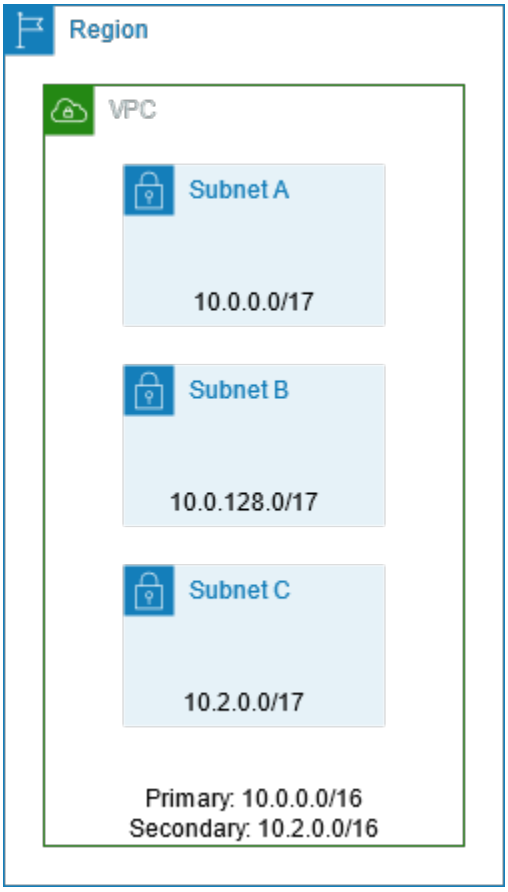
고려 사항

- 다음 CIDR 블록은 VPC에 지정할 수 없습니다.
  - 0.0.0.0/8
  - 127.0.0.0/8(내부 호스트 루프백 주소 범위)
  - 169.254.0.0/16(링크-로컬 주소 범위)
  - 224.0.0.0/4(멀티캐스트 주소 범위)
- AWS 서비스와 함께 사용할 VPC를 생성하려면 해당 서비스 설명서를 참조하여 해당 구성에 대한 특정 요구 사항이 있는지 확인하십시오.
- 일부 AWS 서비스는 `172.17.0.0/16` CIDR 범위를 사용합니다. 네트워크에서 IP 주소 범위를 이미 사용 중인 경우 서비스에서 IP 주소 충돌이 발생할 수 있습니다. 예를 들어, AWS Cloud9 및 Amazon SageMaker AI는 `172.17.0.0/16`을 사용합니다. 충돌을 방지하려면 VPC를 생성할 때 이 범위를 사용하지 마세요. 자세한 내용은 AWS Cloud9 사용 설명서의 Docker에서 VPC의 IP 주소를 사용하므로 EC2 환경에 연결할 수 없음을 참조하세요.
- RFC 1918에 지정된 프라이빗 IPv4 주소 범위에 속하지 않는 공개적으로 라우팅 가능한 CIDR 블록을 사용하여 VPC를 생성할 수 있습니다. 하지만 이 설명서에서는 프라이빗 IP 주소는 VPC의 CIDR 범위 내에 있는 프라이빗 IPv4 주소를 말합니다.
- 명령줄 도구 또는 Amazon EC2 API를 사용하여 VPC를 생성하면 CIDR 블록이 표준 형식으로 자동 수정됩니다. 예를 들어 CIDR 블록에 100.68.0.18/18을 지정하면 100.68.0.0/18의 CIDR 블록이 생성됩니다.

VPC에 IPv4 CIDR 블록 관리

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`).

다음 예에서 VPC에는 기본 및 보조 CIDR 블록이 모두 있습니다. 서브넷 A와 서브넷 B의 CIDR 블록은 기본 VPC CIDR 블록에서 가져옵니다. 서브넷 C의 CIDR 블록은 보조 VPC CIDR 블록에서 가져옵니다.



다음 라우팅 테이블은 VPC의 로컬 경로를 보여줍니다.

대상 주소	대상
10.0.0.0/16	로컬
10.2.0.0/16	로컬

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 내용은 IPv4 CIDR 블록 연결 제한 단원을 참조하세요.
- 기존 CIDR 블록의 크기를 늘리거나 줄일 수 없습니다.
- VPC에 연결할 수 있는 CIDR 블록의 수와 라우팅 테이블에 추가할 수 있는 경로의 수에는 할당량이 있습니다. 할당량을 초과하는 경우에는 CIDR 블록을 연결할 수 없습니다. 자세한 내용은 Amazon VPC 할당량 단원을 참조하세요.
- CIDR 블록은 모든 VPC 라우팅 테이블에서 경로의 대상 CIDR 범위보다 작아야 합니다. 예를 들어, 기본 CIDR 블록이 `10.2.0.0/16`인 VPC에서 가상 프라이빗 게이트웨이에 대한 대상이 `10.0.0.0/24`인 라우팅 테이블에 기존 라우팅이 있습니다. `10.0.0.0/16` 범위의 보조 CIDR 블록을 연결하려고 합니다. 기존 경로 때문에 `10.0.0.0/24` 이상의 CIDR 블록을 연결할 수 없습니다. 그러나 `10.0.0.0/25` 이하의 보조 CIDR 블록은 연결할 수 있습니다.
- VPC 피어링 연결에 포함된 VPC에 IPv4 CIDR 블록을 추가할 때 다음 규칙이 적용됩니다.
  - VPC 피어링 연결이 `active`인 경우, 피어 VPC의 CIDR 블록과 겹치지 않으면 VPC에 CIDR 블록을 추가할 수 있습니다.
  - VPC 피어링 연결이 `pending-acceptance`인 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-acceptance`인 경우, 수락자 VPC이 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.
- Direct Connect를 사용하여 Direct Connect 게이트웨이를 통해 여러 VPC에 연결하는 경우 Direct Connect 게이트웨이에 연결된 VPC에는 중첩되는 CIDR 블록이 있으면 안 됩니다. Direct Connect 게이트웨이와 연결된 VPC 중 하나에 CIDR 블록을 추가한 경우 새로운 CIDR 블록이 다른 연결된 VPC에 있는 기존 CIDR 블록과 중첩되어서는 안 됩니다. 자세한 내용은 Direct Connect 사용 설명서의 Direct Connect 게이트웨이를 참조하세요.
- CIDR 블록을 추가하거나 제거하는 경우, `associating` | `associated` | `disassociating` | `disassociated` | `failing` | `failed`와 같은 다양한 상태를 통과할 수 있습니다. 사용자가 CIDR 블록을 사용할 수 있는 상태가 되면 `associated` 상태가 됩니다.

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 사용자 VPC(Your VPCs)를 선택하고 자신의 VPC의 확인란을 선택한 다음 CIDRs 탭을 선택하십시오. AWS CLI을 사용하여 기본 CIDR을 보려면 다음과 같이 describe-vpcs 명령을 사용하십시오. 기본 CIDR은 최상위 `CidrBlock element`로 반환됩니다.

```
aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d --query Vpcs[*].CidrBlock --output text
```

출력의 예시는 다음과 같습니다.

```
10.0.0.0/16
```

IPv4 CIDR 블록 연결 제한

다음 표에서는 기존 VPC CIDR 블록에 대한 허용 및 제한된 VPC CIDR 블록 연결의 개요를 제공합니다. 제한 이유는 일부 AWS 서비스가 AWS 서비스 측에서 충돌하지 않는 CIDR 블록이 필요한 교차 VPC 및 교차 계정 기능을 사용하기 때문입니다.

기존 IPv4 주소 범위	제한된 연결	허용된 연결
10.0.0.0/8	다른 RFC 1918* 범위(172.16.0.0/12 및 192.168.0.0/16)의 CIDR 블록입니다. VPC에 연결된 CIDR 블록 중 하나가 10.0.0.0/15 범위(10.0.0.0~10.1.255.255)에 해당되면 10.0.0.0/16 범위(10.0.0.0~10.0.255.255)의 CIDR 블록을 추가할 수 없습니다. 198.19.0.0/16 범위의 CIDR 블록입니다.	제한되지 않는 10.0.0.0/8 범위의 기타 모든 /16 넷마스크~/28 넷마스크 CIDR 블록입니다. 100.64.0.0/10 범위의 공개적으로 라우팅할 수 있는 모든 /16 넷마스크~/28 넷마스크 IPv4 CIDR 블록(비 RFC 1918) 또는 /16 넷마스크~/28 넷마스크 CIDR 블록입니다.
169.254.0.0/16	'link local' 블록의 CIDR 블록은 RFC 5735에 설명된 대로 예약되어 있으며 VPC에 할당할 수 없습니다.	

기존 IPv4 주소 범위	제한된 연결	허용된 연결
172.16.0.0/12	다른 RFC 1918* 범위(10.0.0.0/8 및 192.168.0.0/16)의 CIDR 블록입니다. 172.31.0.0/16 범위의 CIDR 블록입니다. 198.19.0.0/16 범위의 CIDR 블록입니다.	제한되지 않는 172.16.0.0/12 범위의 기타 모든 /16 넷마스크~/28 넷마스크 CIDR 블록입니다. 100.64.0.0/10 범위의 공개적으로 라우팅할 수 있는 모든 /16 넷마스크~/28 넷마스크 IPv4 CIDR 블록(비 RFC 1918) 또는 /16 넷마스크~/28 넷마스크 CIDR 블록입니다.
192.168.0.0/16	다른 RFC 1918* 범위(10.0.0.0/8 및 172.16.0.0/12)의 CIDR 블록입니다. 198.19.0.0/16 범위의 CIDR 블록입니다.	192.168.0.0/16 범위의 기타 모든 /16 넷마스크~/28 넷마스크 CIDR 블록입니다. 공개적으로 라우팅할 수 있는 모든 /16 넷마스크~/28 넷마스크 IPv4 CIDR 블록(비 RFC 1918) 또는 100.64.0.0/10 범위의 /16 넷마스크~/28 넷마스크 CIDR 블록입니다.
198.19.0.0/16	RFC 1918* 범위의 CIDR 블록입니다.	공개적으로 라우팅할 수 있는 모든 /16 넷마스크~/28 넷마스크 IPv4 CIDR 블록(비 RFC 1918) 또는 100.64.0.0/10 범위의 /16 넷마스크~/28 넷마스크 CIDR 블록입니다.

기존 IPv4 주소 범위	제한된 연결	허용된 연결
공개적으로 라우팅이 가능한 CIDR 블록(비-RFC 1918) 또는 100.64.0.0/10 범위의 CIDR 블록	RFC 1918* 범위의 CIDR 블록입니다. 198.19.0.0/16 범위의 CIDR 블록입니다.	공개적으로 라우팅할 수 있는 기타 모든 /16 넷마스크~/28 넷마스크 IPv4 CIDR 블록(비 RFC 1918) 또는 100.64.0.0/10 범위의 /16 넷마스크~/28 넷마스크 CIDR 블록입니다. RFC 1918 범위 중 하나에 CIDR을 연결할 수도 있지만, 이렇게 하려면 VPC를 생성할 때 먼저 해당 CIDR을 추가한 다음 비 RFC 1918 CIDR을 추가해야 합니다.

* RFC 1918 범위는 RFC 1918에 지정된 프라이빗 IPv4 주소 범위입니다.

IPv6 VPC CIDR 블록

새 VPC를 생성할 때 단일 IPv6 CIDR 블록을 연결하거나 /44에서 /60까지 /4씩 증가하면서 최대 5개의 IPv6 CIDR 블록을 연결할 수 있습니다. Amazon의 IPv6 주소 풀에서 IPv6 CIDR 블록을 요청할 수 있습니다. 자세한 내용은 VPC에서 CIDR 블록 추가 또는 제거 섹션을 참조하세요.

IPv6 CIDR 블록을 VPC와 연결한 경우, IPv6 CIDR 블록을 VPC의 기존 서브넷 또는 새로 생성한 서브넷과 연결할 수 있습니다. 자세한 내용은 the section called “IPv6에 대한 서브넷 크기 조정” 섹션을 참조하세요.

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

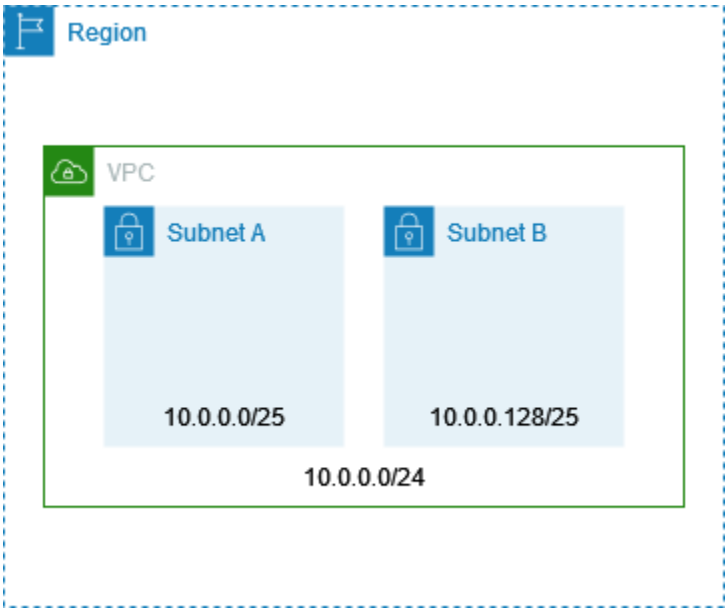
VPC에서 IPv6 CIDR 블록을 연결 해제할 수 있습니다. VPC에 연결된 IPv6 CIDR 블록을 분리한 후 나중에 다시 VPC에 IPv6 CIDR 블록을 연결하는 경우, 동일한 CIDR을 받을 것으로 기대할 수는 없습니다.

서브넷 CIDR 블록

서브넷의 IP 주소는 CIDR(Classless Inter-Domain Routing) 표기법을 사용하여 표시됩니다. 서브넷의 CIDR 블록은 VPC에 대한 CIDR 블록(VPC의 단일 서브넷 생성용) 또는 VPC에 대한 CIDR 블록의 하위 세트(VPC에 여러 서브넷 생성용)와 동일할 수 있습니다.

VPC에 두 개 이상의 서브넷을 만들 경우, 서브넷의 CIDR 블록이 겹치지 않아야 합니다.

예를 들어 CIDR 블록이 `10.0.0.0/24`인 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.



인터넷에서 IPv4 및 IPv6 서브넷 CIDR 블록을 계산하고 생성하는 데 도움이 되는 도구가 있습니다. '서브넷 계산기' 또는 'CIDR 계산기'와 같은 용어를 검색하여 필요에 맞는 도구를 찾을 수 있습니다. 또한 네트워크 엔지니어링 그룹의 도움을 받아 서브넷에 지정할 IPv4 및 IPv6 CIDR 블록을 정할 수도 있습니다.

IPv4에 대한 서브넷 크기 조정

서브넷에 허용되는 IPv4 CIDR 블록 크기는 /28 넷마스크~/16 넷마스크입니다. 각 서브넷 CIDR 블록에서 첫 4개의 IP 주소와 마지막 IP 주소는 사용자가 사용할 수 없으므로 EC2 인스턴스 등의 리소스에 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 기본 VPC 네트워크 범위에 2를 더한 주소입니다. CIDR 블록이 여러 개인 VPC의 경우, DNS 서버의 IP 주소가 기본 CIDR에 위치합니다. 또한 각 서브넷 범위의 기본에 2를 더한 주소를 VPC의 모든 CIDR 블록에 대해 예약합니다. 자세한 내용은 Amazon DNS 서버 섹션을 참조하세요.
- 10.0.0.3: AWS에서 앞으로 사용하려고 예약한 주소.
- 10.0.0.255: 네트워크 브로드캐스트 주소. VPC에서는 브로드캐스트를 지원하지 않으므로, 이 주소를 예약합니다.

명령줄 도구 또는 Amazon EC2 API를 사용하여 서브넷을 추가하면 CIDR 블록이 표준 형식으로 자동 수정됩니다. 예를 들어 CIDR 블록에 100.68.0.18/18을 지정하면 100.68.0.0/18의 CIDR 블록이 생성됩니다.

BYOIP를 사용하여 AWS로 IPv4 주소 범위를 가져오면 첫 번째 주소(네트워크 주소)와 마지막 주소(브로드캐스트 주소)를 포함하여 범위 내의 IP 주소를 모두 사용할 수 있습니다.

IPv6에 대한 서브넷 크기 조정

IPv6 CIDR 블록을 VPC와 연결한 경우, IPv6 CIDR 블록을 VPC의 기존 서브넷 또는 새로 생성한 서브넷과 연결할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/64(/4씩 증가)입니다.

인터넷에서 IPv6 서브넷 CIDR 블록을 계산하고 생성하는 데 도움이 되는 도구가 있습니다. 'IPv6 서브넷 계산기' 또는 'IPv6 CIDR 계산기'와 같은 용어를 검색하여 필요에 맞는 도구를 찾을 수 있습니다. 또한 네트워크 엔지니어링 그룹의 도움을 받아 서브넷에 지정할 IPv6 CIDR 블록을 정할 수도 있습니다.

각 서브넷 CIDR 블록에서 첫 4개의 IPv6 주소와 마지막 IPv6 주소는 사용자가 사용할 수 없으므로 EC2 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`: AWS에서 VPC 라우터용으로 예약한 주소.
- `2001:db8:1234:1a00::2`
- `2001:db8:1234:1a00::3`
- `2001:db8:1234:1a00:ffff:ffff:ffff:ffff`

위 예시에서 VPC 라우터용으로 AWS에서 예약한 IP 주소 외에도 다음 IPv6 주소가 기본 VPC 라우터용으로 예약되어 있습니다.

- EUI-64를 사용하여 생성된 FE80::/10 범위의 링크-로컬 IPv6 주소입니다. 링크-로컬 주소에 대한 자세한 내용은 링크-로컬 주소를 참조하세요.
- 링크-로컬 IPv6 주소 `FE80:ec2::1`입니다.

IPv6를 통해 VPC 라우터와 통신해야 하는 경우 필요에 가장 적합한 주소와 통신하도록 애플리케이션을 구성할 수 있습니다.

IPv4 및 IPv6 비교

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

듀얼 스택 구성(IPv4 및 IPv6)과 IPv6 전용 구성을 지원하는 AWS 서비스 목록은 IPv6를 지원하는 서비스 섹션을 참조하세요

기능	IPv4	IPv6
VPC 크기	/16부터 /28까지 최대 5개의 CIDR. 이 할당량은 조정할 수 있습니다.	/44부터 /60까지(/4씩 증가) 최대 5개 CIDR. 이 할당량은 조정할 수 있습니다.
서브넷 크기	/16~/28	/44에서 /64까지(/4씩 증가).
주소 선택	VPC용 IPv4 CIDR 블록을 선택하거나 Amazon VPC IP 주소 관리자(IPAM)에서 CIDR 블록을 할당할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM이란 무엇인가?를 참조하세요.	자체 IPv6 CIDR 블록을 VPC용으로 AWS로 가져오거나, Amazon 제공 IPv6 CIDR 블록을 선택하거나, Amazon VPC IP 주소 관리자(IPAM)에서 CIDR 블록을 할당할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM이란 무엇인가?를 참조하세요.
인터넷 액세스	인터넷 게이트웨이가 필요합니다.	인터넷 게이트웨이가 필요합니다. 송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 전용 통신을 지원합니다.
탄력적 IP 주소	지원 EC2 인스턴스에 영구 정적 퍼블릭 IPv4 주소를 제공합니다.	지원하지 않음. EIP는 인스턴스 재시작 시 인스턴스의 퍼블릭 IPv4 주소를

기능	IPv4	IPv6
		정적으로 유지합니다. IPv6 주소는 기본적으로 정적입니다.
NAT 게이트웨이	지원 프라이빗 서브넷의 인스턴스는 퍼블릭 NAT 게이트웨이를 사용하여 인터넷에 연결하거나 프라이빗 NAT 게이트웨이를 사용하여 다른 VPC의 리소스에 연결할 수 있습니다.	지원 NAT 게이트웨이를 NAT64와 함께 사용하면 IPv6 전용 서브넷의 인스턴스가 VPC 내, VPC 간, 온프레미스 네트워크 또는 인터넷을 통해 IPv4 전용 리소스와 통신할 수 있습니다.
DNS 이름	인스턴스는 Amazon에서 제공한 IPBN 또는 RBN 기반 DNS 이름을 받습니다. DNS 이름은 인스턴스에 대해 선택한 DNS 레코드로 확인됩니다.	인스턴스는 Amazon에서 제공한 IPBN 또는 RBN 기반 DNS 이름을 받습니다. DNS 이름은 인스턴스에 대해 선택한 DNS 레코드로 확인됩니다.

관리형 접두사 목록으로 네트워크 CIDR 블록 통합 및 관리

관리형 접두사 목록은 하나 이상의 CIDR 블록 세트입니다. 접두사 목록을 사용하면 보안 그룹과 라우팅 테이블을 보다 쉽게 구성하고 유지 관리할 수 있습니다. 자주 사용하는 IP 주소에서 접두사 목록을 만들고, 이를 개별적으로 참조하지 않고 보안 그룹 규칙 및 경로의 집합으로 참조할 수 있습니다. 예를 들어, 서로 다른 CIDR 블록은 있지만 포트와 프로토콜은 동일한 보안 그룹 규칙을 접두사 목록을 사용하는 단일 규칙으로 통합할 수 있습니다. 네트워크를 확장하고 다른 CIDR 블록의 트래픽을 허용해야 하는 경우, 관련 접두사 목록을 업데이트할 수 있으며 그러면 접두사 목록을 사용하는 모든 보안 그룹이 업데이트됩니다. Resource Access Manager(RAM)를 사용하여 다른 AWS 계정과 함께 관리형 접두사 목록을 사용할 수도 있습니다.

접두사 목록에는 두 가지 유형이 있습니다.

- 고객 관리형 접두사 목록 — 사용자가 정의하고 관리하는 IP 주소 범위 세트입니다. 접두사 목록을 다른 AWS 계정과 공유하여 해당 계정이 자체 리소스의 접두사 목록을 참조하도록 할 수 있습니다.
- AWS 관리형 접두사 목록 - AWS 서비스의 IP 주소 범위 세트입니다. AWS 관리형 접두사 목록은 생성, 수정, 공유 또는 삭제할 수 없습니다.

목차

- 접두사 목록 개념 및 규칙
- 접두사 목록에 대한 자격 증명 및 액세스 관리

- 고객 관리형 접두사 목록
- AWS 관리형 접두사 목록
- 접두사 목록으로 AWS 인프라 관리 최적화

접두사 목록 개념 및 규칙

접두사 목록은 항목으로 구성됩니다. 각 항목은 CIDR 블록과 CIDR 블록에 대한 설명(선택 사항)으로 구성됩니다.

고객 관리형 접두사 목록

고객 관리형 접두사 목록에는 다음 규칙이 적용됩니다.

- 접두사 목록은 단일 IP 주소 지정 유형(IPv4 또는 IPv6)만 지원합니다. IPv4 및 IPv6 CIDR 블록을 하나의 접두사 목록에 결합할 수 없습니다.
- 접두사 목록은 해당 목록을 생성한 리전에만 적용됩니다.
- 접두사 목록을 생성할 때 접두사 목록에서 지원할 수 있는 최대 항목 수를 지정해야 합니다.
- 리소스의 접두사 목록을 참조할 때 접두사 목록의 최대 항목 수는 리소스의 항목 수에 대한 할당량에 따라 계산됩니다. 예를 들어, 최대 항목이 20개인 접두사 목록을 만들고 보안 그룹 규칙에서 해당 접두사 목록을 참조하는 경우, 20개의 보안 그룹 규칙이 있는 것으로 계산됩니다.
- 라우팅 테이블의 접두사 목록을 참조할 때 라우팅 우선 순위 규칙이 적용됩니다. 자세한 내용은 접두사 목록의 라우팅 우선 순위 단원을 참조하세요.
- 접두사 목록을 수정할 수 있습니다. 항목을 추가하거나 제거하면 접두사 목록의 새 버전이 생성됩니다. 접두사를 참조하는 리소스는 항상 현재(최신) 버전을 사용합니다. 이전 버전의 접두사 목록에 있는 항목을 복원할 수 있으며 이것도 새 버전을 생성합니다.
- 접두사 목록과 관련된 할당량이 있습니다. 자세한 내용은 고객 관리형 접두사 목록 섹션을 참조하세요.
- 고객 관리형 접두사 목록은 모든 상용 AWS 리전(GovCloud(미국) 및 중국 리전 포함)에서 사용할 수 있습니다.

AWS 관리형 접두사 목록

AWS 관리형 접두사 목록에는 다음 규칙이 적용됩니다.

- AWS 관리형 접두사 목록은 생성, 수정, 공유 또는 삭제할 수 없습니다.

- 서로 다른 AWS 관리형 접두사 목록은 사용할 때 다른 가중치를 갖습니다. 자세한 내용은 AWS 관리형 접두사 목록 가중치 섹션을 참조하세요.
- AWS 관리형 접두사 목록의 버전 번호는 볼 수 없습니다.

접두사 목록에 대한 자격 증명 및 액세스 관리

기본적으로 사용자는 접두사 목록을 생성, 보기, 수정 또는 삭제할 수 있는 권한이 없습니다. IAM 정책을 생성하여 이를 사용자가 접두사 목록 작업을 할 수 있는 역할에 연결할 수 있습니다.

IAM 정책에서 사용할 수 있는 Amazon VPC 작업 목록과 리소스 및 조건 키를 보려면 서비스 승인 참조의 Amazon EC2에서 사용되는 작업, 리소스 및 조건 키를 참조하세요.

다음 예시 정책은 사용자가 접두사 목록 pl-123456abcde123456만 보고 작업할 수 있도록 허용합니다. 사용자는 접두사 목록을 생성하거나 삭제할 수 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetManagedPrefixListAssociations",
                "ec2:GetManagedPrefixListEntries",
                "ec2:ModifyManagedPrefixList",
                "ec2:RestoreManagedPrefixListVersion"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:prefix-
list/pl-123456abcde123456"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeManagedPrefixLists",
            "Resource": "*"
        }
    ]
}
```

Amazon VPC에서 IAM을 이용하는 자세한 방법은 Amazon VPC용 자격 증명 및 액세스 관리을 참조하세요.

고객 관리형 접두사 목록

고객 관리형 접두사 목록을 사용하면 AWS 내에서 접두사라는 IP 주소 범위 세트를 직접 정의하고 유지할 수 있습니다. 이러한 IP 주소를 다양한 리소스에 하드코딩하는 대신 중앙 집중식 접두사 목록을 생성하여 필요할 때마다 참조할 수 있습니다. 이렇게 하면 IP 주소 관리가 간소화될 뿐만 아니라 AWS 환경 전반에서 일관성과 재사용성이 향상됩니다.

고객 관리형 접두사 목록의 뛰어난 기능 중 하나는 이를 다른 AWS 계정과 공유할 수 있다는 것입니다. 접두사 목록에 대한 액세스 권한을 부여하면 다른 팀이나 조직이 자체 리소스에서 사용자가 정의한 IP 주소 범위를 사용할 수 있습니다. 이러한 협업 접근 방식은 IP 주소 관리를 공유하고 동기화하는 보다 응집력 있고 효율적인 클라우드 환경을 조성합니다.

다음 섹션에서는 IP 주소 범위 생성, 관리 및 공유에 대한 단계별 지침을 포함하여 고객 관리형 접두사 목록 작업의 실제 측면을 자세히 살펴보겠습니다.

> **Note**
>
> Amazon VPC IPAM을 사용하여 접두사 목록 관리를 자동화하고, 정의한 규칙에 따라 CIDR을 자동으로 동기화할 수 있습니다. 이렇게 하면 인프라가 변경될 때 수동 업데이트가 필요하지 않습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM으로 접두사 목록 자동화를 참조하세요.

작업

- 고객 관리형 접두사 목록 작업

고객 관리형 접두사 목록 작업

이 섹션에서는 고객 관리형 접두사 목록으로 작업하는 방법을 설명합니다.

내용

- 접두사 목록 생성
- 접두사 목록 보기
- 접두사 목록 항목 보기
- 접두사 목록에 대한 연결(참조) 보기

- 접두사 목록 수정
- 접두사 목록 크기 조정
- 이전 버전의 접두사 목록 복원
- 접두사 목록 삭제
- 고객 관리형 접두사 목록 공유

접두사 목록 생성

접두사 목록을 생성할 때 접두사 목록에서 지원할 수 있는 최대 항목 수를 지정해야 합니다.

제한 사항

규칙 수에 접두사 목록의 최대 항목을 더한 값이 계정의 보안 그룹당 규칙의 할당량을 초과하는 경우, 보안 그룹 규칙에 접두사 목록을 추가할 수 없습니다.

콘솔을 사용하여 접두사 목록을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록 생성을 선택합니다.
4. 접두사 목록 이름에 접두사 목록 이름을 입력합니다.
5. 최대 항목에 접두사 목록의 최대 항목 수를 입력합니다.
6. 주소 패밀리에 접두사 목록에서 IPv4 항목을 지원하는지 IPv6 항목을 지원하는지 여부를 선택합니다.
7. 접두사 목록 항목에서 새 항목 추가를 선택하고 CIDR 블록과 항목에 대한 설명을 입력합니다. 각 항목에 대해 이 단계를 반복합니다.
8. (선택 사항) 나중에 식별할 수 있도록 태그에서 접두사 목록에 대한 태그를 추가합니다.
9. 접두사 목록 생성을 선택합니다.

AWS CLI를 사용하여 접두사 목록을 생성하려면

create-managed-prefix-list 명령을 사용합니다.

접두사 목록 보기

접두사 목록, 공유된 접두사 목록 및 AWS 관리형 접두사 목록을 볼 수 있습니다.

콘솔을 사용하여 접두사 목록을 보려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 소유자 ID 열에 접두사 목록 소유자의 AWS 계정 ID가 표시됩니다. AWS 관리형 접두사 목록의 경우 소유자 ID는 AWS입니다.

AWS CLI를 사용하여 접두사 목록을 보려면

describe-managed-prefix-lists 명령을 사용합니다.

접두사 목록 항목 보기

접두사 목록, 공유된 접두사 목록 및 AWS 관리형 접두사 목록의 항목을 볼 수 있습니다.

콘솔을 사용하여 접두사 목록의 항목을 보려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록의 확인란을 선택합니다.
4. 아래쪽 창에서 항목을 선택하여 접두사 목록의 항목을 봅니다.

AWS CLI를 사용하여 접두사 목록의 항목을 보려면

get-managed-prefix-list-entries 명령을 사용합니다.

접두사 목록에 대한 연결(참조) 보기

접두사 목록과 연결된 리소스의 ID 및 소유자를 볼 수 있습니다. 연결된 리소스는 해당 항목 또는 규칙에서 접두사 목록을 참조하는 리소스입니다.

제한 사항

AWS 관리형 접두사 목록에 대한 관련 리소스는 볼 수 없습니다.

콘솔을 사용하여 접두사 목록 연결을 보려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록의 확인란을 선택합니다.

4. 아래쪽 창에서 연결을 선택하여 접두사 목록을 참조하는 리소스를 봅니다.

AWS CLI를 사용하여 접두사 목록 연결을 보려면

get-managed-prefix-list-associations 명령을 사용합니다.

접두사 목록 수정

접두사 목록의 이름을 수정하고 항목을 추가 또는 제거할 수 있습니다. 최대 항목 수를 수정하려면 접두사 목록 크기 조정 섹션을 참조하세요.

접두사 목록의 항목을 업데이트하면 접두사 목록의 새 버전이 생성됩니다. 접두사 목록의 이름이나 최대 항목 수를 업데이트해도 접두사 목록의 새 버전이 생성되지는 않습니다.

고려 사항

- AWS 관리형 접두사 목록은 수정할 수 없습니다.
- 접두사 목록에서 최대 항목 수를 늘리면 접두사 목록을 참조하는 리소스의 항목 할당량에 증가된 최대 크기가 적용됩니다. 이러한 리소스 중 증가된 최대 크기를 지원할 수 없는 경우 수정 작업이 실패하고 이전 최대 크기가 복원됩니다.

콘솔을 사용하여 접두사 목록을 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록의 확인란을 선택하고 작업(Actions), 접두사 목록 수정(Modify prefix list)을 선택합니다.
4. 접두사 목록 이름에 접두사 목록의 새 이름을 입력합니다.
5. 관리형 접두사 목록이 IPAM 접두사 목록 해석기 대상으로 구성된 경우 IPAM 접두사 목록 해석기 동기화 옵션이 표시됩니다.

   IPAM 접두사 목록 해석기와의 동기화를 활성화할지 비활성화할지 선택합니다. 활성화하면 접두사 목록 CIDR은 연결된 해석기의 CIDR 선택 규칙에 따라 자동으로 업데이트됩니다. 비활성화하면 접두사 목록 CIDR이 자동으로 업데이트되지 않습니다. 이 기능에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM으로 접두사 목록 업데이트 자동화를 참조하세요.
6. 접두사 목록 항목에서 제거를 선택하여 기존 항목을 제거합니다. 새 항목을 추가하려면 새 항목 추가를 선택하고 CIDR 블록과 항목에 대한 설명을 입력합니다.
7. 접두사 목록 저장을 선택합니다.

AWS CLI를 사용하여 접두사 목록을 수정하려면

modify-managed-prefix-list 명령을 사용합니다.

접두사 목록 크기 조정

접두사 목록의 크기를 조정하고 접두사 목록의 최대 항목 수를 최대 1000개까지 수정할 수 있습니다. 고객 관리형 접두사 키에 대한 자세한 내용은 고객 관리형 접두사 목록 섹션을 참조하세요.

콘솔을 사용하여 접두사 목록 크기 조정

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록의 확인란을 선택하고 작업(Actions), 접두사 목록 크기 조정(Resize prefix list)을 선택합니다.
4. 새 최대 항목(New max entries)에 값을 입력합니다.
5. 크기 조정(Resize)을 선택합니다.

AWS CLI를 사용하여 접두사 목록 크기 조정

modify-managed-prefix-list 명령을 사용합니다.

이전 버전의 접두사 목록 복원

이전 버전의 접두사 목록에서 항목을 복원할 수 있습니다. 그러면 새로운 버전의 접두사 목록이 생성됩니다.

접두사 목록의 크기를 줄인 경우 접두사 목록이 이전 버전의 항목을 포함할 수 있을 만큼 충분히 커야 합니다.

콘솔을 사용하여 이전 버전의 접두사 목록을 복원하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록의 확인란을 선택하고 작업(Actions), 접두사 목록 복원(Restore prefix list)을 선택합니다.
4. 접두사 목록 버전 선택(Select prefix list version)에서 이전 버전을 선택합니다. 선택한 버전에 대한 항목은 접두사 목록 항목(Prefix list entries)에 표시됩니다.
5. 접두사 목록 복원(Restore prefix list)을 선택합니다.

AWS CLI를 사용하여 이전 버전의 접두사 목록을 복원하려면

restore-managed-prefix-list-version 명령을 사용합니다.

접두사 목록 삭제

접두사 목록을 삭제하려면 먼저 라우팅 테이블과 같은 리소스에서 접두사 목록에 대한 참조를 제거해야 합니다. AWS RAM을 사용하여 접두사 목록을 공유한 경우 먼저 소비자가 소유한 리소스에서 참조를 제거해야 합니다.

제한 사항

AWS 관리형 접두사 목록은 삭제할 수 없습니다.

콘솔을 사용하여 접두사 목록을 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록을 선택하고 작업, 접두사 목록 삭제를 선택합니다.
4. 확인 대화 상자에 `delete`를 입력한 다음 삭제를 선택합니다.

AWS CLI를 사용하여 접두사 목록을 삭제하려면

delete-managed-prefix-list 명령을 사용합니다.

고객 관리형 접두사 목록 공유

AWS Resource Access Manager(AWS RAM)를 사용하여 고객 관리형 접두사 목록의 소유자는 접두사 목록을 다음과 공유할 수 있습니다.

- AWS 조직 내부 또는 외부의 특정 AWS Organizations 계정
- AWS Organizations에서 조직 내부의 조직 단위
- AWS Organizations의 전체 조직

접두사 목록을 공유하는 소비자는 접두사 목록과 해당 항목을 볼 수 있으며 AWS 리소스에서 접두사 목록을 참조할 수 있습니다.

AWS RAM에 대한 추가 정보는 AWS RAM 사용 설명서를 참조하세요. 자세한 내용은 AWS RAM 참조 안내서의 Service Quotas를 참조하세요.

> **Important**
>
> 접두사 목록 공유에 대한 추가 비용은 없습니다.

내용

- 공유 접두사 목록 사용 권한
- 공유 접두사 목록 작업

공유 접두사 목록 사용 권한

소유자에 대한 권한

소유자는 공유 접두사 목록 및 해당 항목을 관리할 책임이 있습니다. 소유자는 접두사 목록을 참조하는 AWS 리소스의 ID를 볼 수 있습니다. 그러나 소비자가 소유한 AWS 리소스의 접두사 목록에 대한 참조를 추가하거나 제거할 수는 없습니다.

접두사 목록이 소비자가 소유한 리소스에서 참조되는 경우 소유자는 접두사 목록을 삭제할 수 없습니다.

소비자에 대한 권한

소비자는 공유 접두사 목록의 항목을 볼 수 있으며 AWS 리소스에서 공유 접두사 목록을 참조할 수 있습니다. 그러나 소비자는 공유 접두사 목록을 수정, 복원 또는 삭제할 수 없습니다.

공유 접두사 목록 작업

AWS 접두사 목록은 다양한 AWS 서비스에서 사용하는 IP 주소 범위를 관리하고 참조하는 편리한 방법을 제공합니다. AWS 관리형 접두사 목록 외에도 고객 관리형 접두사 목록을 직접 생성하고 다른 AWS 계정과 공유할 수도 있습니다.

접두사 목록 공유는 복잡한 네트워킹 요구 사항이 있거나 여러 AWS 워크로드에서 IP 주소 사용을 조정해야 하는 조직에 특히 유용할 수 있습니다. 접두사 목록을 공유하여 일관된 IP 주소 관리를 보장하고 공동 작업자를 위한 네트워킹 구성을 단순화할 수 있습니다.

이 섹션에서는 접두사 목록을 공유하는 방법 및 계정과 공유된 접두사 목록을 식별하고 사용하는 방법을 설명합니다.

내용

- 접두사 목록 공유
- 공유 접두사 목록 공유 해제
- 공유 접두사 목록 식별
- 공유 접두사 목록에 대한 참조 식별

접두사 목록 공유

접두사 목록을 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유가 없는 경우 먼저 AWS RAM 콘솔을 사용하여 리소스 공유를 생성해야 합니다.

AWS Organizations의 조직에 속해 있고 조직 내의 공유가 활성화되어 있으면 조직의 소비자에게 공유 접두사 목록에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않은 경우 리소스 공유에 가입하라는 초대를 받은 소비자가 초대를 수락하면 공유 접두사 목록에 대한 액세스 권한이 부여됩니다.

리소스 공유를 생성하고 AWS RAM 콘솔 또는 AWS CLI를 사용하여 소유한 접두사 목록을 공유할 수 있습니다.

Important

- 접두사 목록을 공유하려면 계정에서 해당 접두사 목록을 소유해야 합니다. 다른 사용자가 자신과 공유한 접두사 목록은 공유할 수 없습니다. AWS 관리형 접두사 목록은 공유할 수 없습니다.
- AWS Organizations의 조직 또는 조직 단위와 접두사 목록을 공유하려면 AWS Organizations와의 공유를 활성화해야 합니다. 자세한 내용은 AWS RAM 사용 설명서에서 AWS Organizations를 사용하여 공유 사용을 참조하세요.

AWS RAM 콘솔을 사용하여 리소스 공유를 생성하고 접두사 목록을 공유하려면

AWS RAM 사용 설명서의 리소스 공유 생성 단계를 따릅니다. 리소스 유형 선택에서 접두사 목록을 선택한 다음 접두사 목록에 대한 확인란을 선택합니다.

AWS RAM 콘솔을 사용하여 기존 리소스 공유에 접두사 목록을 추가하려면

자신이 소유한 관리형 접두사를 기존 리소스 공유에 추가하려면 AWS RAM 사용 설명서의 리소스 공유 업데이트 단계를 따릅니다. 리소스 유형 선택에서 접두사 목록을 선택한 다음 접두사 목록에 대한 확인란을 선택합니다.

AWS CLI를 사용하여 자신이 소유한 접두사 목록을 공유하려면

다음 명령을 사용하여 리소스 공유를 생성하고 업데이트합니다.

- create-resource-share
- associate-resource-share
- update-resource-share

공유 접두사 목록 공유 해제

접두사 목록의 공유를 해제하면 소비자가 더 이상 해당 계정에서 접두사 목록 또는 해당 항목을 볼 수 없으며 리소스에서 접두사 목록을 참조할 수 없습니다. 접두사 목록이 이미 소비자의 리소스에서 참조된 경우 이러한 참조는 정상적으로 계속 작동하며 이러한 참조를 계속 볼 수 있습니다. 접두사 목록을 새 버전으로 업데이트하는 경우 참조는 최신 버전을 사용합니다.

자신이 소유한 공유 접두사 목록을 공유 해제하려면 AWS RAM을 사용해 리소스 공유에서 제거해야 합니다.

AWS RAM 콘솔을 사용하여 자신이 소유한 공유 접두사 목록을 공유 해제하려면

AWS RAM 사용 설명서에서 리소스 공유 업데이트를 참조하세요.

AWS CLI를 사용하여 자신이 소유한 공유 접두사 목록을 공유 해제하려면

disassociate-resource-share 명령을 사용합니다.

공유 접두사 목록 식별

소유자와 소비자는 Amazon VPC 콘솔 및 AWS CLI를 사용하여 공유 접두사 목록을 식별할 수 있습니다.

Amazon VPC 콘솔을 사용하여 공유 접두사 목록을 식별하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 이 페이지에는 자신이 소유한 접두사 목록과 다른 사용자가 자신과 공유한 접두사 목록이 표시됩니다. 소유자 ID 열에 접두사 목록 소유자의 AWS 계정 ID가 표시됩니다.
4. 접두사 목록에 대한 리소스 공유 정보를 보려면 접두사 목록을 선택하고 아래쪽 창에서 공유를 선택합니다.

AWS CLI를 사용하여 공유 접두사 목록을 식별하려면

describe-managed-prefix-lists 명령을 사용합니다. 이 명령은 자신이 소유한 접두사 목록과 다른 사용자가 자신과 공유한 접두사 목록을 반환합니다. `OwnerId`에는 접두사 목록 소유자의 AWS 계정 ID가 표시됩니다.

공유 접두사 목록에 대한 참조 식별

소유자는 공유 접두사 목록을 참조하는 소비자 소유 리소스를 식별할 수 있습니다.

Amazon VPC 콘솔을 사용하여 공유 접두사 목록에 대한 참조를 식별하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 접두사 목록을 선택하고 아래쪽 창에서 연결을 선택합니다.
4. 접두사 목록을 참조하는 리소스의 ID가 리소스 ID 열에 나열됩니다. 리소스 소유자가 리소스 소유자 열에 나열됩니다.

AWS CLI를 사용하여 공유 접두사 목록에 대한 참조를 식별하려면

get-managed-prefix-list-associations 명령을 사용합니다.

AWS 관리형 접두사 목록

AWS 관리형 접두사 목록은 AWS 서비스에 대한 IP 주소 범위 세트입니다. 이러한 접두사 목록은 Amazon Web Services에서 유지 관리하며 다양한 AWS 제품에서 사용되는 IP 주소를 참조하는 방법을 제공합니다. 이는 VPC 내에서 보안 그룹 또는 기타 네트워크 수준 컨트롤을 구성할 때 특히 유용할 수 있습니다.

접두사 목록은 S3, DynamoDB 등 다양한 AWS 서비스를 포함합니다. 관리형 접두사 목록을 사용하면 네트워크 구성을 최신 상태로 유지하고 사용자가 의존하는 AWS 서비스에서 사용하는 IP 주소를 올바르게 파악할 수 있습니다. 이를 통해 네트워킹 작업을 단순화하고 IP 주소 목록을 수동으로 유지 관리하는 데 따르는 관리 부담을 줄일 수 있습니다.

관리형 접두사 목록 사용은 실질적 이점을 제공할 뿐만 아니라 AWS 보안 모범 사례와도 부합합니다. AWS에서 제공하는 신뢰할 수 있는 IP 주소 정보를 사용하면 잘못된 구성이나 예기치 않은 연결 문제의 위험을 최소화할 수 있습니다. 이는 엄격한 규정 준수 요구 사항이 있는 미션 크리티컬 애플리케이션 또는 워크로드에 특히 중요할 수 있습니다.

내용

- 사용 가능한 AWS 관리형 접두사 목록
- AWS 관리형 접두사 목록 가중치
- AWS 관리형 접두사 목록 사용

사용 가능한 AWS 관리형 접두사 목록

다음과 같은 서비스에서 AWS 관리형 접두사 목록이 제공됩니다.

AWS 서비스	접두사 목록 이름	가중치
Amazon CloudFront +	com.amazonaws.global.cloudfront.origin-facing (IPv4) com.amazonaws.global.ipv6.cloudfront.origin-facing(IPv6)	55
Amazon DynamoDB	com.amazonaws.*region*.dynamodb	1
Amazon EC2 Instance Connect	com.amazonaws.*region*.ec2-instance-connect	2
	com.amazonaws.*region*.ipv6.ec2-instance-connect	2
AWS Ground Station	com.amazonaws.global.groundstation	5
Amazon Route 53	com.amazonaws.*region*.ipv6.route53-healthchecks	25
	com.amazonaws.*region*.route53-healthchecks	25
Amazon S3	com.amazonaws.*region*.s3	1
Amazon S3 Express One Zone	com.amazonaws.*region*.s3express	6
AWS Secrets Manager	com.amazonaws.*region*.secretsmanager-managed-external-secrets	20
Amazon VPC Lattice	com.amazonaws.*region*.vpc-lattice	10

AWS 서비스	접두사 목록 이름	가중치
	com.amazonaws.*region*.ipv6.vpc-lattice	10

콘솔을 사용하여 AWS 관리형 접두사 목록을 보는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Managed Prefix Lists를 선택합니다.
3. 검색 필드에서 Owner ID: AWS 필터를 추가합니다.

AWS CLI를 사용하여 AWS 관리형 접두사 목록을 보는 방법

다음과 같이 describe-managed-prefix-lists 명령을 사용합니다.

```
aws ec2 describe-managed-prefix-lists --filters Name=owner-id,Values=AWS
```

AWS 관리형 접두사 목록 가중치

AWS 관리형 접두사 목록의 가중치는 리소스에서 차지하는 항목 수를 나타냅니다.

예를 들어 Amazon CloudFront 관리형 접두사 목록의 가중치는 55입니다. Amazon VPC 할당량에 미치는 영향은 다음과 같습니다.

- 보안 그룹 – 기본 할당량이 규칙 60개이므로 보안 그룹에 추가 규칙의 여지가 5개만 남습니다. 이 할당량에 대해 할당량 증가 요청을 할 수 있습니다.
- 라우팅 테이블 – 기본 할당량이 경로 50개이므로 라우팅 테이블에 접두사 목록을 추가하기 전에 할당량 증가를 요청해야 합니다.

AWS 관리형 접두사 목록 사용

AWS 관리형 접두사 목록은 AWS에서 생성하고 유지관리할 수 있고 AWS 계정이 있는 모든 사용자가 사용할 수 있습니다. AWS 관리형 접두사 목록은 생성, 수정, 공유 또는 삭제할 수 없습니다.

고객 관리형 접두사 목록과 마찬가지로 AWS 관리형 접두사 목록을 보안 그룹 및 라우팅 테이블과 같은 AWS 리소스와 함께 사용할 수 있습니다. 자세한 내용은 접두사 목록으로 AWS 인프라 관리 최적화 섹션을 참조하세요.

접두사 목록으로 AWS 인프라 관리 최적화

다음 AWS 리소스에서 접두사 목록을 참조할 수 있습니다.

리소스

- VPC 보안 그룹
- 서브넷 라우팅 테이블
- 전송 게이트웨이 라우팅 테이블
- AWS Network Firewall 규칙 그룹
- Amazon Managed Grafana 네트워크 액세스 제어
- AWS Outposts 랙 로컬 게이트웨이

VPC 보안 그룹

접두사 목록을 인바운드 규칙의 소스로 지정하거나 아웃바운드 규칙의 대상으로 지정할 수 있습니다. 자세한 내용은 보안 그룹 섹션을 참조하세요.

> **Important**
>
> 접두사 목록을 사용하도록 기존 규칙을 수정할 수는 없습니다. 접두사 목록을 사용하려면 새 규칙을 성성해야 합니다.

콘솔을 사용하여 보안 그룹 규칙에서 접두사 목록을 참조하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security Groups를 선택합니다.
3. 업데이트할 보안 그룹을 선택합니다.
4. 작업, 인바운드 규칙 편집 또는 작업, 아웃바운드 규칙 편집을 선택합니다.
5. [Add another rule]을 선택합니다. 유형에서 트래픽 유형을 선택합니다. 소스(인바운드 규칙) 또는 대상(아웃바운드 규칙)에서 사용자 지정을 선택합니다. 다음 필드인 접두사 목록에서 접두사 목록의 ID를 선택합니다.
6. 규칙 저장을 선택합니다.

AWS CLI를 사용하여 보안 그룹 규칙에서 접두사 목록을 참조하려면

authorize-security-group-ingress 및 authorize-security-group-egress 명령을 사용합니다. `--ip-permissions` 파라미터의 경우 `PrefixListIds`를 사용하여 접두사 목록의 ID를 지정합니다.

서브넷 라우팅 테이블

라우팅 테이블 항목의 대상으로 접두사 목록을 지정할 수 있습니다. 게이트웨이 라우팅 테이블에서는 접두사 목록을 참조할 수 없습니다. 라우팅 테이블에 대한 자세한 내용은 라우팅 테이블 구성 단원을 참조하세요.

콘솔을 사용하여 라우팅 테이블에서 접두사 목록을 참조하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route Tables를 선택한 후 라우팅 테이블을 선택합니다.
3. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
4. 라우팅을 추가하려면 라우팅 추가를 선택합니다.
5. 대상에 접두사 목록의 ID를 입력합니다.
6. 대상에서 대상을 선택합니다.
7. [Save changes]를 선택합니다.

AWS CLI를 사용하여 라우팅 테이블에서 접두사 목록을 참조하려면

create-route(AWS CLI) 명령을 사용합니다. `--destination-prefix-list-id` 파라미터를 사용하여 접두사 목록의 ID를 지정합니다.

전송 게이트웨이 라우팅 테이블

라우팅의 대상으로 접두사 목록을 지정할 수 있습니다. 자세한 내용은 Amazon VPC Transit Gateways의 접두사 목록 참조를 참조하세요.

AWS Network Firewall 규칙 그룹

AWS Network Firewall 그룹은 네트워크 트래픽을 검사하고 처리하기 위한 재사용 가능한 기준 세트입니다. AWS Network Firewall에서 Suricata와 호환되는 상태 저장 규칙 그룹을 생성하는 경우 규칙 그룹에서 접두사 목록을 참조할 수 있습니다. 자세한 내용은 AWS Network Firewall개발자 안내서의 Amazon VPC 접두사 목록 참조 및 상태 저장 규칙 그룹 생성을 참조하세요.

Amazon Managed Grafana 네트워크 액세스 제어

하나 이상의 접두사 목록을 Amazon Managed Grafana 작업 영역에 대한 요청에 대한 인바운드 규칙으로 지정할 수 있습니다. 접두사 목록을 참조하는 방법을 포함하여 Grafana 작업 영역 네트워크 액세스 제어에 대한 자세한 내용은 Amazon Managed Grafana 사용 설명서의 네트워크 액세스 관리를 참조하세요.

AWS Outposts 랙 로컬 게이트웨이

각 AWS Outposts 랙에서는 Outpost 리소스를 온프레미스 네트워크와 연결할 수 있는 로컬 게이트웨이가 제공됩니다. 접두사 목록에서 자주 사용하는 CIDR를 그룹화하고 이 목록을 로컬 게이트웨이 라우팅 테이블의 라우팅 대상으로 참조할 수 있습니다. 자세한 내용은 랙용 AWS Outposts 사용 설명서의 로컬 게이트웨이 라우팅 테이블 경로 관리를 참조하세요.

AWS IP 주소 범위

AWS은(는) 현재 IP 주소 범위를 JSON 형식으로 게시합니다. 이 정보를 사용하여 AWS에서 트래픽을 식별할 수 있습니다. 또한 이 정보를 사용하여 일부 AWS 서비스로 전송하거나 그로부터 수신되는 트래픽을 허용 또는 거부할 수도 있습니다.

고려 사항

- AWS는 고객이 송신 필터링을 수행하는 데 일반적으로 사용하는 서비스의 IP 주소 범위를 게시합니다. 모든 서비스의 IP 주소 범위를 게시하지는 않습니다.
- 서비스에서 IP 주소 범위를 사용하여 다른 서비스와 통신하거나 고객 네트워크와 통신합니다.
- 고유 IP 주소 가져오기(BYOIP)를 통해 AWS(으)로 가져온 IP 주소 범위는 `.json` 파일에 포함되지 않습니다. 자세한 내용은 Amazon EC2 사용 설명서의 AWS를 통해 주소 범위 광고를 참조하세요.

일부 서비스는 AWS 관리형 접두사 목록을 사용하여 주소 범위를 게시합니다. 자세한 내용은 the section called “사용 가능한 AWS 관리형 접두사 목록” 단원을 참조하세요.

내용

- JSON 파일 다운로드
- 송신 제어
- 지리적 위치 피드
- AWS 서비스의 IP 주소 범위를 찾습니다.

- AWS IP 주소 범위 JSON 구문
- AWS IP 주소 범위 알림

JSON 파일 다운로드

현재 주소 범위를 보려면 ip-ranges.json을 다운로드합니다. 기록을 유지하려면 연속 버전의 JSON 파일을 본인 시스템에 저장합니다. 파일을 마지막으로 저장한 이후에 변경 사항이 있는지 확인하려면 현재 파일의 게시 시간을 마지막으로 저장한 파일의 게시 시간과 비교합니다.

다음은 JSON 파일을 현재 디렉터리에 저장하는 curl 명령 예시입니다.

```
curl -O https://ip-ranges.amazonaws.com/ip-ranges.json
```

이 파일에 프로그래밍 방식으로 액세스할 경우 애플리케이션에서 서버에 제공된 TLS 인증서를 확인한 이후에 파일을 다운로드하는지 직접 확인해야 합니다.

JSON 파일 업데이트 알림을 받으려면 AWS IP 주소 범위 알림 섹션을 참조하세요.

송신 제어

하나의 AWS 서비스를 이용하여 생성한 리소스가 다른 AWS 서비스에만 액세스하는 것을 허용하기 위해서는 ip-ranges.json 파일에 있는 IP 주소 범위 정보를 사용하여 송신 필터링 작업을 수행할 수 있습니다. 보안 그룹 규칙에서 AMAZON 목록의 CIDR 블록으로의 아웃바운드 트래픽이 허용되는지 확인하세요. 보안 그룹에 대한 할당량이 있습니다. 각 리전의 IP 주소 범위 수에 따라 리전당 여러 보안 그룹이 필요할 수 있습니다.

> **Note**
>
> 일부 AWS 서비스는 EC2를 기반으로 구축되었으며 EC2 IP 주소 공간을 사용합니다. EC2 IP 주소 공간으로의 트래픽을 차단할 경우 이러한 EC2 이외의 서비스에 대한 트래픽도 차단됩니다.

지리적 위치 피드

`ip-ranges.json`의 IP 주소 범위는 AWS 리전 기준입니다. 그러나 로컬 영역의 물리적 위치는 상위 리전과 동일하지 않습니다. geo-ip-feed.csv에 게시된 로컬 영역에 대한 지리적 위치 데이터입니다. 데이터는 RFC 8805를 따릅니다.

AWS 서비스의 IP 주소 범위를 찾습니다.

AWS에서 제공하는 AWS IP 주소 범위 JSON 파일은 다양한 AWS 서비스의 IP 주소를 확인하고, 해당 정보를 활용하여 네트워크 보안 및 액세스 제어를 강화하는 데 유용한 리소스입니다. 이 JSON 파일에 포함된 세부 데이터를 구문 분석하면 특정 AWS 서비스 및 리전과 연결된 IP 주소 범위를 정확하게 식별할 수 있습니다.

예를 들어 이 IP 주소 범위를 사용하여 강력한 네트워크 보안 정책을 구성하고, 세분화된 방화벽 규칙을 설정하여 특정 AWS 리소스에 대한 액세스를 허용하거나 거부할 수 있습니다. 이 정보는 다양한 AWS Network Firewall 네트워크 방화벽 작업에도 유용할 수 있습니다. 이러한 수준의 제어는 애플리케이션과 데이터를 보호하는 데 매우 중요하며, 승인된 트래픽만 필요한 AWS 서비스에 도달할 수 있도록 보장합니다. 또한 이러한 IP 인텔리전스를 확보하면 애플리케이션이 올바른 AWS 엔드포인트와 통신하도록 적절하게 구성되어 전반적인 안정성과 성능을 개선하는 데 도움이 됩니다.

방화벽 규칙뿐 아니라 `ip-ranges.json` 파일을 활용하여 네트워크 인프라에서 정교한 송신 필터링을 구성할 수도 있습니다. 다양한 AWS 서비스의 목적지 IP 주소 범위를 파악하면 라우팅 정책을 설정하거나 고급 네트워크 보안 솔루션을 활용하여 지정된 목적지에 따라 아웃바운드 트래픽을 선택적으로 허용 또는 차단할 수 있습니다. 이러한 송신 제어는 데이터 유출과 무단 액세스의 위험을 완화하는 데 필수적입니다.

`ip-ranges.json` 파일은 정기적으로 업데이트되므로 가장 정확하고 최신 정보를 확보하려면 최신 로컬 복사본을 유지하는 것이 중요합니다. 이 파일의 내용을 지속적으로 활용하면 AWS 기반 애플리케이션의 네트워크 액세스 및 보안을 효율적으로 관리하고 전반적인 클라우드 보안 태세를 강화할 수 있습니다.

다음 예시는 AWS IP 주소 범위를 원하는 내용으로 필터링하는 데 도움이 될 수 있습니다. Linux에서는 jq 도구를 다운로드하여 사용하여 JSON 파일의 로컬 사본을 파싱할 수 있습니다. AWS Tools for Windows PowerShell에는 이 JSON 파일을 파싱하는 데 사용할 수 있는 Get-AWSPublicIpAddressRange라는 cmdlet이 포함되어 있습니다. 자세한 내용은 다음 블로그를 참조하세요. AWS의 퍼블릭 IP 주소 범위 쿼리

JSON 파일을 가져오려면 the section called “다운로드” 섹션을 참조하세요. JSON 파일의 구문에 대한 자세한 내용은 the section called “구문” 단원을 참조하세요.

예제

- 파일 생성 날짜 가져오기
- 특정 리전의 IP 주소 가져오기
- 모든 IPv4 주소 가져오기

- 특정 서비스에 대한 모든 IPv4 주소를 가져옵니다.
- 특정 리전의 특정 서비스에 대한 모든 IPv4 주소 가져오기
- 모든 IPv6 주소 가져오기
- 특정 서비스에 대한 모든 IPv6 주소를 가져옵니다.
- 특정 경계 그룹의 모든 IP 주소 가져오기

파일 생성 날짜 가져오기

다음 예시에서는 `ip-ranges.json`의 생성 날짜를 가져옵니다.

jq

```
$ jq .createDate < ip-ranges.json

"2024-08-01-17-22-15"
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -OutputPublicationDate

Thursday, August 1, 2024 9:22:35 PM
```

특정 리전의 IP 주소 가져오기

다음은 지정된 리전의 IP 주소에 대해 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq '.prefixes[] | select(.region=="us-east-1")' < ip-ranges.json

{
  "ip_prefix": "23.20.0.0/14",
  "region": "us-east-1",
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.16.0.0/15",
  "region": "us-east-1",
```

```
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.19.0.0/16",
  "region": "us-east-1",
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -Region us-east-1

IpPrefix        Region      NetworkBorderGroup         Service
--------        ------       -------                   -------
23.20.0.0/14    us-east-1    us-east-1                 AMAZON
50.16.0.0/15    us-east-1    us-east-1                 AMAZON
50.19.0.0/16    us-east-1    us-east-1                 AMAZON
...
```

모든 IPv4 주소 가져오기

다음은 IPv4 주소의 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.prefixes | .[].ip_prefix' < ip-ranges.json

23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.IpAddressFormat -eq "Ipv4"} | select
 IpPrefix

IpPrefix
--------
```

```
23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

특정 서비스에 대한 모든 IPv4 주소를 가져옵니다.

다음은 지정된 서비스의 IPv4 주소에 대해 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.prefixes[] | select(.service=="GLOBALACCELERATOR") | .ip_prefix' < ip-
ranges.json

13.248.117.0/24
15.197.34.0/23
15.197.36.0/22
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -ServiceKey GLOBALACCELERATOR | where
 {$_.IpAddressFormat -eq "Ipv4"} | select IpPrefix

IpPrefix
--------
13.248.117.0/24
15.197.34.0/23
15.197.36.0/22
...
```

특정 리전의 특정 서비스에 대한 모든 IPv4 주소 가져오기

다음은 지정된 리전에서 지정된 서비스의 IPv4 주소에 대해 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.prefixes[] | select(.region=="us-east-1") |
 select(.service=="GLOBALACCELERATOR") | .ip_prefix' < ip-ranges.json
```

```
13.248.124.0/24
99.82.166.0/24
99.82.171.0/24
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -Region us-east-1 -ServiceKey GLOBALACCELERATOR
 | where {$_.IpAddressFormat -eq "Ipv4"} | select IpPrefix

IpPrefix
--------
13.248.117.0/24
99.82.166.0/24
99.82.171.0/24
...
```

모든 IPv6 주소 가져오기

다음은 IPv6 주소의 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.ipv6_prefixes | .[].ipv6_prefix' < ip-ranges.json

2a05:d07c:2000::/40
2a05:d000:8000::/40
2406:dafe:2000::/40
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.IpAddressFormat -eq "Ipv6"} | select
 IpPrefix

IpPrefix
--------
2a05:d07c:2000::/40
2a05:d000:8000::/40
2406:dafe:2000::/40
...
```

특정 서비스에 대한 모든 IPv6 주소를 가져옵니다.

다음은 지정된 서비스의 IPv6 주소에 대해 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.ipv6_prefixes[] | select(.service=="GLOBALACCELERATOR") | .ipv6_prefix' <
 ip-ranges.json

2600:1f01:4874::/47
2600:1f01:4802::/47
2600:1f01:4860::/47
2600:9000:a800::/40
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -ServiceKey GLOBALACCELERATOR | where
 {$_.IpAddressFormat -eq "Ipv6"} | select IpPrefix

IpPrefix
--------
2600:1f01:4874::/47
2600:1f01:4802::/47
2600:1f01:4860::/47
2600:9000:a800::/40
...
```

특정 경계 그룹의 모든 IP 주소 가져오기

다음은 지정된 경계 그룹의 모든 IP 주소에 대해 JSON 파일을 필터링하는 예시입니다.

jq

```
$ jq -r '.prefixes[] | select(.network_border_group=="us-west-2-lax-1")
 | .ip_prefix' < ip-ranges.json
70.224.192.0/18
52.95.230.0/24
15.253.0.0/16
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.NetworkBorderGroup -eq "us-west-2-
lax-1"} | select IpPrefix

IpPrefix
--------
70.224.192.0/18
52.95.230.0/24
15.253.0.0/16
...
```

AWS IP 주소 범위 JSON 구문

AWS은(는) 현재 IP 주소 범위를 JSON 형식으로 게시합니다. JSON 파일을 가져오려면 the section called “다운로드” 섹션을 참조하세요. JSON 파일의 구문은 다음과 같습니다.

```
{
  "syncToken": "0123456789",
  "createDate": "yyyy-mm-dd-hh-mm-ss",
  "prefixes": [
    {
      "ip_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ],
  "ipv6_prefixes": [
    {
      "ipv6_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ]
}
```

syncToken

게시 시간(Unix epoch 시간 형식)

유형: 문자열

예시: `"syncToken": "1416435608"`

createDate

발행 날짜 및 시간(UTC YY-MM-DD-hh-mm-ss 형식)

유형: 문자열

예시: `"createDate": "2014-11-19-23-29-02"`

prefixes

IPv4 주소 범위에 대한 IP 접두사

유형: 배열

ipv6_prefixes

IPv6 주소 범위에 대한 IP 접두사

유형: 배열

ip_prefix

퍼블릭 IPv4 주소 범위(CIDR 표기법)입니다. AWS에서는 구체적인 범위로 접두사를 알릴 수 있습니다. 예를 들어, 파일의 96.127.0.0/17 접두사를 96.127.0.0/21, 96.127.8.0/21, 96.127.32.0/19 및 96.127.64.0/18로 알릴 수 있습니다.

유형: 문자열

예시: `"ip_prefix": "198.51.100.2/24"`

ipv6_prefix

퍼블릭 IPv6 주소 범위(CIDR 표기법)입니다. AWS에서는 구체적인 범위로 접두사를 알릴 수 있습니다.

유형: 문자열

예시: `"ipv6_prefix": "2001:db8:1234::/64"`

network_border_group

AWS가 IP 주소 또는 `GLOBAL`을 알리는 가용 영역 또는 로컬 영역의 고유한 집합인 네트워크 경계 그룹의 이름입니다. `GLOBAL` 서비스의 트래픽이 AWS가 IP 주소를 알리는 여러 또는 모든 가용 영역이나 로컬 영역으로 유입되거나 이들 영역에서 시작될 수 있습니다.

유형: 문자열

예시: `"network_border_group": "us-west-2-lax-1"`

region

AWS 리전 또는 GLOBAL입니다. GLOBAL 서비스의 트래픽이 여러 또는 모든 AWS 리전으로 유입되거나 이들 리전에서 시작될 수 있습니다.

유형: 문자열

유효한 값: af-south-1 | ap-east-1 | ap-east-2 | ap-northeast-1 | ap-northeast-2 | ap-northeast-3 | ap-south-1 | ap-south-2 | ap-southeast-1 | ap-southeast-2 | ap-southeast-3 | ap-southeast-4 | ap-southeast-5 | ap-southeast-6 | ap-southeast-7 | ca-central-1 | ca-west-1 | cn-north-1 | cn-northwest-1 | eu-central-1 | eu-central-2 | eu-north-1 | eu-south-1 | eu-south-2 | eu-west-1 | eu-west-2 | eu-west-3 | il-central-1 | mx-central-1 | me-central-1 | me-south-1 | sa-east-1 | us-east-1 | us-east-2 | us-gov-east-1 | us-gov-west-1 | us-west-1 | us-west-2 | GLOBAL

예시: `"region": "us-east-1"`

서비스

IP 주소 범위의 일부입니다. API_GATEWAY에 대해 나열된 주소는 송신 전용입니다. 모든 IP 주소 범위를 가져오도록 AMAZON을 지정합니다. 이렇게 하면 모든 서브셋이 AMAZON 서브셋에도 있습니다. 하지만 일부 IP 주소 범위는 AMAZON 서브셋에만 있습니다. 즉, 다른 서브셋에서 사용할 수 없습니다.

유형: 문자열

유효한 값: AMAZON | AMAZON_APPFLOW | AMAZON_CONNECT | API_GATEWAY | AURORA_DSQL | CHIME_MEETINGS | CHIME_VOICECONNECTOR | CLOUD9 | CLOUDFRONT | CLOUDFRONT_ORIGIN_FACING | CODEBUILD | DYNAMODB | EBS | EC2 | EC2_INSTANCE_CONNECT | GLOBALACCELERATOR | IVS_LOW_LATENCY | IVS_REALTIME | KINESIS_VIDEO_STREAMS | MEDIA_PACKAGE_V2 | ROUTE53 | ROUTE53_HEALTHCHECKS | ROUTE53_HEALTHCHECKS_PUBLISHING | ROUTE53_RESOLVER | S3 | WORKSPACES_GATEWAYS

예시: `"service": "AMAZON"`

범위 중복

모든 서비스 코드에서 반환되는 IP 주소 범위는 AMAZON 서비스 코드에서도 반환됩니다. 예를 들어 S3 서비스 코드에서 반환되는 모든 IP 주소 범위는 AMAZON 서비스 코드에서도 반환됩니다.

서비스 A가 서비스 B의 리소스를 사용하는 경우 서비스 A와 서비스 B 모두에 대해 서비스 코드에서 반환되는 IP 주소 범위가 있습니다. 그러나 이러한 IP 주소 범위는 서비스 A에서만 사용되며 서비스 B에서는 사용할 수 없습니다. 예를 들어 Amazon S3는 Amazon EC2의 리소스를 사용하므로 S3 및 EC2 서비스 코드 모두에서 반환되는 IP 주소 범위가 있습니다. 그러나 이러한 IP 주소 범위는 Amazon S3에서만 사용됩니다. 따라서 S3 서비스 코드는 Amazon S3에서만 사용하는 모든 IP 주소 범위를 반환합니다. Amazon EC2에서만 사용되는 IP 주소 범위를 식별하려면 EC2 서비스 코드에서 반환되지만 S3 서비스 코드에서는 반환되지 않는 IP 주소 범위를 찾으세요.

자세히 알아보기

이 섹션에서는 다양한 서비스 코드에 대한 추가 정보 링크를 제공합니다.

- AMAZON_APPFLOW – IP 주소 범위
- AMAZON_CONNECT – 네트워크 설정
- CHIME_MEETINGS – 미디어 및 신호 전송 구성
- CLOUDFRONT – CloudFront 엣지 서버의 위치 및 IP 주소 범위
- DYNAMODB – IP 주소 범위
- EC2 – 퍼블릭 IPV4 주소
- EC2_INSTANCE_CONNECT – EC2 인스턴스 연결 사전 조건
- GLOBALACCELERATOR – Global Accelerator 엣지 서버의 위치 및 IP 주소 범위
- ROUTE53 – Amazon Route 53 서버의 IP 주소 범위
- ROUTE53_HEALTHCHECKS – Amazon Route 53 서버의 IP 주소 범위
- ROUTE53_HEALTHCHECKS_PUBLISHING – Amazon Route 53 서버의 IP 주소 범위
- WORKSPACES_GATEWAYS – PCoIP 게이트웨이 서버

릴리스 노트

다음 표에서는 ip-ranges.json 구문에 대한 업데이트를 설명합니다. 또한 각 리전 출시와 함께 새로운 리전 코드도 추가됩니다.

설명	릴리스 날짜
IVS_LOW_LATENCY 서비스 코드를 추가했습니다.	2025년 7월 29일
AURORA_DSQL 서비스 코드를 추가했습니다.	2025년 5월 21일
IVS_REALTIME 서비스 코드를 추가했습니다.	2024년 6월 11일
MEDIA_PACKAGE_V2 서비스 코드를 추가했습니다.	2023년 5월 9일
CLOUDFRONT_ORIGIN_FACING 서비스 코드를 추가했습니다.	2021년 10월 12일
ROUTE53_RESOLVER 서비스 코드를 추가했습니다.	2021년 6월 24일
EBS 서비스 코드를 추가했습니다.	2021년 5월 12일
KINESIS_VIDEO_STREAMS 서비스 코드를 추가했습니다.	2020년 11월 19일
CHIME_MEETINGS 및 CHIME_VOICECONNECTOR 서비스 코드를 추가했습니다.	2020년 6월 19일
AMAZON_APPFLOW 서비스 코드를 추가했습니다.	2020년 6월 9일
네트워크 경계 그룹에 대한 지원을 추가합니다.	2020년 4월 7일
WORKSPACES_GATEWAYS 서비스 코드를 추가했습니다.	2020년 3월 30일
ROUTE53_HEALTHCHECK_PUBLISHING 서비스 코드를 추가했습니다.	2020년 1월 30일
API_GATEWAY 서비스 코드를 추가했습니다.	2019년 9월 26일

설명	릴리스 날짜
EC2_INSTANCE_CONNECT 서비스 코드를 추가했습니다.	2019년 26월 6일
DYNAMODB 서비스 코드를 추가했습니다.	2019년 4월 25일
GLOBALACCELERATOR 서비스 코드를 추가했습니다.	2018년 12월 20일
AMAZON_CONNECT 서비스 코드를 추가했습니다.	2018년 6월 20일
CLOUD9 서비스 코드를 추가했습니다.	2018년 6월 20일
CODEBUILD 서비스 코드를 추가했습니다.	2018년 4월 19일
S3 서비스 코드를 추가했습니다.	2017년 2월 28일
IPv6 주소 범위에 대한 지원을 추가했습니다.	2016년 8월 22일
초기 릴리스	2014년 11월 19일

AWS IP 주소 범위 알림

AWS은(는) 현재 IP 주소 범위를 JSON 형식으로 게시합니다. AWS IP 주소 범위가 변경될 때마다 Amazon SNS 주제 구독자에게 AmazonIpSpaceChanged라는 알림을 보냅니다. JSON 파일의 구문에 대한 자세한 내용은 the section called “구문” 단원을 참조하세요.

이 알림의 페이로드에는 다음 형식의 정보가 포함되어 있습니다.

```
{
  "create-time":"yyyy-mm-ddThh:mm:ss+00:00",
  "synctoken":"0123456789",
  "md5":"6a45316e8bc9463c9e926d5d37836d33",
  "url":"https://ip-ranges.amazonaws.com/ip-ranges.json"
}
```

create-time

생성 날짜 및 시간

알림이 제공되는 순서는 정해져 있지 않습니다. 따라서 타임스탬프를 통해 올바른 순서를 확인하는 것이 좋습니다.

synctoken

게시 시간(Unix epoch 시간 형식)

md5

`ip-ranges.json` 파일의 암호화 해시 값입니다. 이 값을 사용하여 다운로드한 파일이 손상되었는지 여부를 확인할 수 있습니다.

url

`ip-ranges.json` 파일의 위치 자세한 내용은 the section called “다운로드” 섹션을 참조하세요.

다음과 같이 구독하여 알림을 받을 수 있습니다.

AWS IP 주소 범위 알림을 구독하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. 구독하는 SNS 알림이 이 리전에 생성되었기 때문에 이 리전을 선택해야 합니다.
3. 탐색 창에서 Subscriptions를 선택합니다.
4. 구독 생성을 선택합니다.
5. 구독 생성 대화 상자에서 다음 작업을 수행합니다.
   a. [Topic ARN]의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다.

      ```
      arn:aws:sns:us-east-1:806199016981:AmazonIpSpaceChanged
      ```

   b. [Protocol]의 경우, 사용할 프로토콜을 선택합니다(예: `Email`).
   c. Endpoint의 경우, 알림을 받을 엔드포인트를 입력합니다(예: 이메일 주소).
   d. 구독 생성을 선택합니다.
6. 지정한 엔드포인트에서 연락이 오고 구독을 확인하라는 메시지가 표시됩니다. 예를 들어 이메일 주소를 지정한 경우, 제목이 `AWS Notification - Subscription Confirmation`인 이메일 메시지를 받게 됩니다. 지시에 따라 구독을 확인합니다.

알림은 엔드포인트의 가용성을 따릅니다. 따라서 JSON 파일을 주기적으로 확인하여 최신 범위를 가져왔는지 확인할 수 있습니다. Amazon SNS 안정성에 대한 자세한 내용은 https://aws.amazon.com/sns/faqs/#Reliability를 참조하세요.

이런 알림을 더 이상 받지 않기를 원하는 경우, 다음 절차를 수행해서 구독을 해제하세요.

AWS IP 주소 범위 알림을 구독 해제하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 구독을 선택합니다.
3. 구독의 확인란을 선택합니다.
4. [Actions], [Delete subscriptions]를 차례로 선택합니다.
5. 확인 메시지가 나타나면 삭제를 선택합니다.

Amazon SNS에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 가이드를 참조하세요.

VPC에 대한 IPv6 지원

IPv4만을 지원하는 기존 VPC와 서브넷에서 IPv4만을 사용하도록 구성된 리소스가 있으면 VPC 및 리소스에 대한 IPv6 지원을 추가할 수 있습니다. VPC는 듀얼 스택 모드로 작동할 수 있으므로, 리소스는 IPv4나 IPv6 또는 둘 다를 통해 통신할 수 있습니다. IPv4 및 IPv6 통신 프로토콜은 상호 독립적입니다.

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

고려 사항

- IPv4 전용 서브넷에서 IPv6 전용 서브넷으로의 마이그레이션 경로가 없습니다.
- 이 예시에서는 퍼블릭 및 프라이빗 서브넷을 포함하는 VPC를 보유하고 있다고 가정합니다. IPv6에서 사용할 새 VPC 생성에 대한 자세한 내용은 the section called “VPC 생성” 섹션을 참조하세요.
- IPv6를 사용하여 시작하기 전에 Amazon VPC에 대한 IPv6 주소 지정 기능을 읽어 보시길 바랍니다 (IPv4 및 IPv6 비교).

내용

- VPC에 대한 IPv6 지원 추가

- 이중 스택 VPC 구성 예시

VPC에 대한 IPv6 지원 추가

다음 표는 VPC에 대해 IPv6를 활성화하는 프로세스의 개요를 간략히 설명합니다

내용

- 1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결
- 2단계: 라우팅 테이블 업데이트
- 3단계: 보안 그룹 규칙 업데이트
- 4단계: 인스턴스에 IPv6 주소 할당

단계	참고
1단계: IPv6 CIDR 블록을 VPC 및 서브넷에 연결	Amazon 제공 또는 BYOIP IPv6 CIDR 블록을 VPC 및 서브넷과 연결합니다.
2단계: 라우팅 테이블 업데이트	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	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: 인스턴스에 IPv6 주소 할당	서브넷의 IPv6 주소 범위에서 인스턴스에 IPv6 주소를 할당합니다.

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. 해당 VPC를 선택합니다.
4. 작업, CIDR 편집을 선택한 후 새 IPv6 CIDR 추가를 선택합니다.
5. 다음 옵션 중 하나를 선택한 다음 CIDR 선택을 선택합니다.
   - Amazon에서 제공한 IPv6 CIDR 블록 - Amazon의 IPv6 주소 풀에서 IPv6 CIDR 블록을 사용합니다. 네트워크 경계 그룹에서 AWS가 IP 주소를 알리는 그룹을 선택합니다.
   - IPAM에 할당된 IPv6 CIDR 블록 - IPAM 풀의 IPv6 CIDR 블록을 사용합니다. IPAM 풀과 IPv6 CIDR 블록을 선택합니다.
   - 내가 소유한 IPv6 CIDR - IPv6 주소 풀에서 IPv6 CIDR 블록을 사용합니다(BYOIP). IPv6 주소 풀과 IPv6 CIDR 블록을 선택합니다.
6. 닫기를 선택하세요.

IPv6 CIDR 블록을 서브넷에 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. 작업, IPv6 CIDR 편집을 선택한 후 IPv6 CIDR 추가를 선택합니다.
5. 필요한 경우 CIDR 블록을 편집합니다(예: 00 교체).
6. 저장을 선택합니다.
7. 이 절차를 VPC의 다른 서브넷에 반복합니다.

자세한 내용은 IPv6 VPC CIDR 블록 섹션을 참조하세요.

2단계: 라우팅 테이블 업데이트

IPv6 CIDR 블록을 VPC에 연결하면 로컬 경로가 각 라우팅 테이블에 자동으로 추가되어 VPC 내부에서 IPv6 트래픽이 허용됩니다.

인스턴스(예: 웹 서버)가 IPv6 트래픽용 인터넷 게이트웨이를 사용할 수 있도록 퍼블릭 서브넷의 라우팅 테이블을 업데이트해야 합니다. 또한 NAT 게이트웨이에서 IPv6이 지원되지 않기 때문에 인스턴스(예: 데이터베이스 인스턴스)가 IPv6 트래픽용 송신 전용 인터넷 게이트웨이를 사용할 수 있도록 프라이빗 서브넷의 라우팅 테이블을 업데이트해야 합니다.

퍼블릭 서브넷의 라우팅 테이블을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다. 퍼블릭 서브넷을 선택합니다. 라우팅 테이블 탭에서 라우팅 테이블 ID를 선택하여 라우팅 테이블에 대한 세부 정보 페이지를 엽니다.
3. 라우팅 테이블을 선택합니다. 라우팅 탭에서 라우팅 편집을 선택합니다.
4. 라우팅 추가를 선택합니다. 대상에 대해 `::/0`을 선택합니다. 대상에 대한 인터넷 게이트웨이의 ID를 선택합니다.
5. 변경 사항 저장을 선택합니다.

프라이빗 서브넷의 라우팅 테이블을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 송신 전용 인터넷 게이트웨이를 선택합니다. 송신 전용 인터넷 게이트웨이 생성을 선택합니다. VPC에서 VPC를 선택한 후 송신 전용 인터넷 게이트웨이 만들기를 선택합니다.

   자세한 내용은 송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽 활성화 섹션을 참조하세요.
3. 탐색 창에서 Subnets를 선택합니다. 프라이빗 서브넷을 선택합니다. 라우팅 테이블 탭에서 라우팅 테이블 ID를 선택하여 라우팅 테이블에 대한 세부 정보 페이지를 엽니다.
4. 라우팅 테이블을 선택합니다. 라우팅 탭에서 라우팅 편집을 선택합니다.
5. 라우팅 추가를 선택합니다. 대상에 대해 `::/0`을 선택합니다. 대상에 대한 송신 전용 인터넷 게이트웨이의 ID를 선택합니다.
6. 변경 사항 저장을 선택합니다.

> **Note**
>
> 라우팅 테이블은 인터넷 게이트웨이와 외부 전용 인터넷 게이트웨이 양쪽을 동시에 가리키는 동일한 대상(::/0)을 가질 수 없습니다. 외부 전용 인터넷 게이트웨이를 구성할 때 "다음 홉을 인터넷 게이트웨이로 사용하는 기존 ipv6 경로가 있습니다"라는 오류 메시지가 표시되는 경우 외

부 전용 인터넷 게이트웨이에 경로를 추가하기 전에 먼저 인터넷 게이트웨이에 대한 기존 IPv6 경로를 제거해야 합니다.

자세한 내용은 라우팅 옵션 예 섹션을 참조하세요.

3단계: 보안 그룹 규칙 업데이트

인스턴스가 IPv6을 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-11aa22bb11aa22bb1)을 업데이트하여 IPv6 주소로부터 인바운드 HTTP, HTTPS, SSH 액세스를 허용하는 규칙을 추가할 수 있습니다. 데이터베이스 보안 그룹에 대한 인바운드 규칙을 변경할 필요는 없습니다. sg-11aa22bb11aa22bb1의 모든 통신을 허용하는 규칙에는 IPv6 통신이 포함되어 있습니다.

인바운드 보안 그룹 규칙을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택하고 웹 서버 보안 그룹을 선택합니다.
3. 인바운드 규칙 탭에서 인바운드 규칙 편집을 선택합니다.
4. IPv4 트래픽이 허용되는 각 규칙에 대해 규칙 추가를 선택하고 해당 IPv6 트래픽이 허용되도록 규칙을 구성합니다. 예를 들어 IPv6를 통한 모든 HTTP 트래픽을 허용하는 규칙을 추가하려면 유형에서 HTTP를 선택하고 소스에서 ::/0을 선택합니다.
5. 규칙 추가가 완료되면 규칙 저장을 선택합니다.

아웃바운드 보안 그룹 규칙 업데이트

IPv6 CIDR 블록을 VPC와 연결할 경우 모든 IPv6 트래픽을 허용하는 VPC의 보안 그룹에 아웃바운드 규칙이 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.

네트워크 ACL 규칙 업데이트

IPv6 CIDR 블록을 VPC와 연결하면 규칙이 기본 네트워크 ACL에 자동으로 추가되어 IPv6 트래픽이 허용됩니다. 그러나 기본 네트워크 ACL을 수정했거나 사용자 지정 네트워크 ACL을 생성한 경우 IPv6 트래픽에 대한 규칙을 수동으로 추가해야 합니다. 자세한 내용은 규칙 추가 및 삭제를 참조하세요.

4단계: 인스턴스에 IPv6 주소 할당

현재 세대의 모든 인스턴스 유형은 IPv6를 지원합니다. 인스턴스 유형이 IPv6를 지원하지 않는 경우 IPv6 주소를 할당하기 전에 지원되는 인스턴스 유형에 맞게 인스턴스 크기를 조정해야 합니다. 사용하는 프로세스는 선택한 새 인스턴스 유형과 현재 인스턴스 유형과의 호환 가능 여부에 의해 결정됩니다. 자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 인스턴스 유형 변경을 참조하세요. IPv6를 지원하기 위해 새 AMI에서 인스턴스를 실행해야 하는 경우 시작 중에 인스턴스에 IPv6 주소를 할당할 수 있습니다.

인스턴스 유형이 IPv6를 지원한다는 것을 확인했으면 Amazon EC2 콘솔을 사용하여 인스턴스에 IPv6 주소를 할당할 수 있습니다. IPv6 주소는 인스턴스의 주 네트워크 인터페이스(예: eth0)에 할당됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스에 IPv6 주소 할당을 참조하세요.

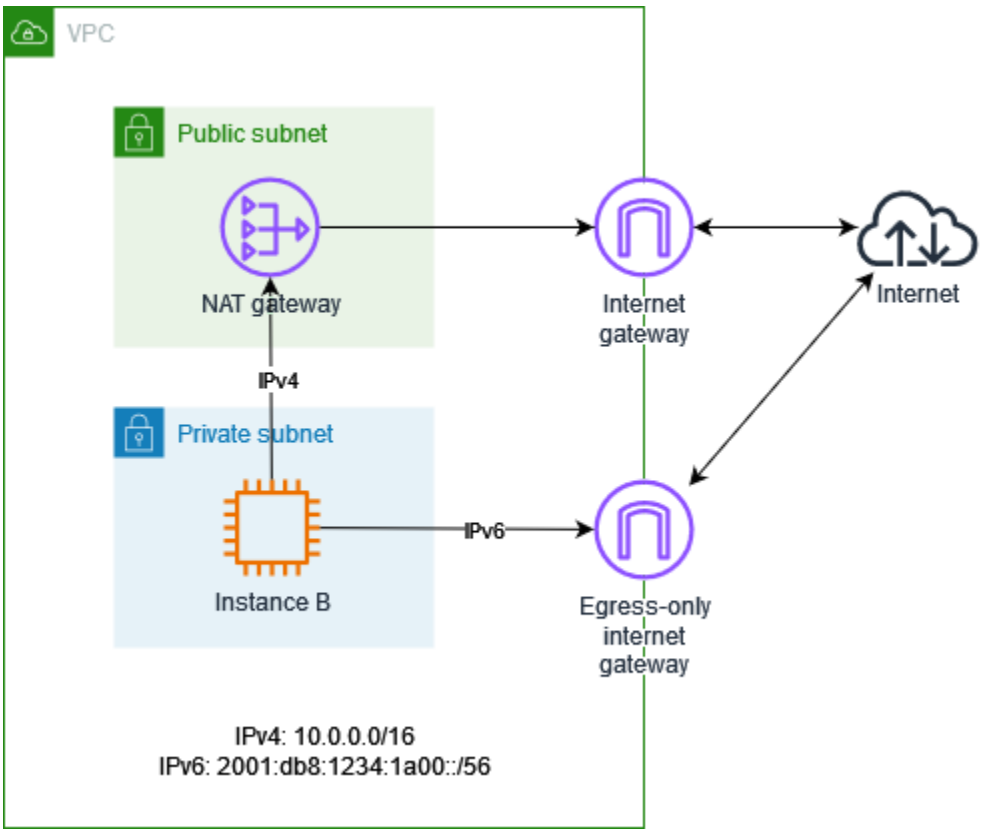
IPv6 주소를 사용하여 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 SSH 클라이언트를 사용하여 Linux 인스턴스에 연결을 참조하세요.

운영 체제의 현재 버전에 맞는 AMI를 사용하여 인스턴스를 실행한 경우에는 인스턴스가 IPv6에 대해 구성됩니다. 인스턴스에서 IPv6 주소를 ping할 수 없는 경우에는 운영 체제에 대한 설명서를 참조하여 IPv6를 구성하세요.

이중 스택 VPC 구성 예시

이중 스택 구성을 사용하면 인터넷을 통해 VPC의 리소스와 리소스 간의 통신에 IPv4 주소와 IPv6 주소를 모두 사용할 수 있습니다.

다음 다이어그램은 VPC의 아키텍처를 보여줍니다. VPC에는 퍼블릭 서브넷과 프라이빗 서브넷이 있습니다. VPC 및 서브넷에는 IPv4 CIDR 블록과 IPv6 CIDR 블록이 둘 다 있어야 합니다. 프라이빗 서브넷에 IPv4 주소와 IPv6 주소가 모두 있는 EC2 인스턴스가 있습니다. 이 인스턴스는 NAT 게이트웨이를 사용하여 아웃바운드 IPv4 트래픽을 인터넷으로 전송하고 외부 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽을 인터넷으로 전송할 수 있습니다.



퍼블릭 서브넷의 라우팅 테이블

다음은 퍼블릭 서브넷용 라우팅 테이블입니다. 처음 두 항목은 로컬 경로입니다. 세 번째 항목은 모든 IPv4 트래픽을 인터넷 게이트웨이로 보냅니다. 네 번째 항목은 퍼블릭 서브넷의 IPv6 주소를 사용하여 EC2 인스턴스를 시작하려는 경우에만 필요합니다.

대상 주소	대상
VPC IPv4 CIDR	로컬
VPC IPv6 CIDR	로컬
0.0.0.0/0	*internet-gateway-id*
::/0	*internet-gateway-id*

프라이빗 서브넷의 라우팅 테이블

다음은 프라이빗 서브넷용 라우팅 테이블입니다. 처음 두 항목은 로컬 경로입니다. 세 번째 항목은 모든 IPv4 트래픽을 NAT 게이트웨이로 보냅니다. 마지막 항목은 모든 IPv6 트래픽을 외부 전용 인터넷 게이트웨이로 보냅니다.

대상 주소	대상
VPC IPv4 CIDR	로컬
VPC IPv6 CIDR	로컬
0.0.0.0/0	*nat-gateway-id*
::/0	*egress-only-gateway-id*

IPv6를 지원하는 AWS 서비스

컴퓨터와 스마트 장치는 IP 주소를 사용하여 인터넷 및 기타 네트워크를 통해 서로 통신합니다. 인터넷이 계속 성장함에 따라 IP 주소에 대한 필요성도 증가하고 있습니다. IP 주소의 가장 일반적인 형식은 IPv4입니다. IP 주소의 새로운 형식은 IPv4보다 더 큰 주소 공간을 제공하는 IPv6입니다.

IPv6에 대한 AWS 서비스 지원에는 듀얼 스택 구성(IPv4 및 IPv6) 또는 IPv6 전용 구성에 대한 지원이 포함됩니다. 예를 들어 가상 프라이빗 클라우드(VPC)는 AWS 리소스를 시작할 수 있는 AWS 클라우드의 논리적으로 격리된 영역입니다. VPC 내에서 IPv4 전용, 듀얼 스택 또는 IPv6 전용 서브넷을 생성할 수 있습니다.

AWS 서비스는 퍼블릭 엔드포인트를 통한 액세스를 지원합니다. 일부 AWS 서비스는 AWS PrivateLink에서 제공하는 프라이빗 엔드포인트를 사용하는 액세스도 지원합니다. AWS 서비스는 퍼블릭 엔드포인트를 통해 IPv6를 지원하지 않더라도 프라이빗 엔드포인트를 통해 IPv6를 지원할 수 있습니다. IPv6를 지원하는 엔드포인트는 AAAA 레코드를 사용하여 DNS 쿼리에 응답할 수 있습니다.

IPv6를 지원하는 서비스

다음 표에는 듀얼 스택 지원, IPv6 전용 지원 및 IPv6를 지원하는 엔드포인트를 제공하는 AWS 서비스가 나와 있습니다. IPv6에 대한 추가 지원이 출시되면 이 표를 업데이트할 예정입니다. 서비스에서 IPv6이 지원되는 방법에 대한 구체적인 사항을 확인하려면 서비스에 대한 설명서를 참조하세요.

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS Amplify	예	아니요	예	
Amazon API Gateway	예	아니요	예	예
AWS App Mesh	예	예	예	아니요
AWS App Runner	예	아니요	예	예
AWS AppConfig	예	아니요	예	예
Application Auto Scaling	아니요	아니요	예	예
AWS Application Discovery Service	예	아니요	예	예
Application Recovery Controller(ARC)	예	아니요	예	
Amazon WorkSpaces Applications	예	아니요	아니요	아니요
AWS AppSync[2]	부분적	아니요	부분적	예
Amazon Athena	예	아니요	예	예
AWS Audit Manager	아니요	아니요	예	예
Amazon Aurora	예	아니요	예	아니요
Amazon Aurora DSQL	아니요	아니요	예	예
AWS Auto Scaling	아니요	아니요	예	예
AWS B2B Data Interchange	예	아니요	예	예
AWS Backup	예	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS Batch	예	아니요	예	예
Amazon Bedrock	아니요	아니요	예	예
AWS 결제 및 비용 관리 데이터 내보내기	예	아니요	예	예
AWS 결제 및 비용 관리 요금 계산기	예	아니요	예	예
AWS Billing Conductor	예	아니요	예	예
AWS Budgets	예	아니요	예	
Amazon Braket	예	예	예	예
AWS Certificate Manager	예	아니요	예	아니요
Amazon Chime SDK	예	아니요	예	
Amazon Comprehend	예	예	예	예
AWS Clean Rooms	예	예	예	예
AWS Clean Rooms ML	예	예	예	예
AWS Cloud9	예	아니요	예	
AWS Cloud Control API	예	아니요	예	예
CloudFormation	아니요	아니요	예	예
Amazon CloudFront	예	예	예	
AWS CloudHSM	예	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS CloudTrail	예	아니요	예	예
Amazon CloudWatch	예	예	예	예
Amazon CloudWatch Application Insights	아니요	아니요	예	예
Amazon CloudWatch Internet Monitor	아니요	아니요	예	예
Amazon CloudWatch Logs	예	예	예	예
Amazon CloudWatch Observability Access Manager	예	예	예	
Amazon CloudWatch Synthetics	예	아니요	예	예
AWS Cloud Map	예	예	예	예
AWS Cloud WAN	예	아니요	예	아니요
AWS CodeArtifact	예	아니요	예	예
고객 Customer Profiles 연결	예	아니요	예	예
AWS CodeBuild	아니요	아니요	예	예
AWS CodeCommit	아니요	아니요	예	예
AWS CodeDeploy	아니요	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS Compute Optimizer	아니요	아니요	예	예
Amazon Comprehend Medical	아니요	아니요	예	예
Amazon CodeGuru Profiler	예	아니요	예	예
Amazon Cognito	예	아니요	예	
AWS Config	아니요	아니요	예	예
AWS Control Tower	아니요	아니요	예	예
AWS Cost Explorer	예	아니요	예	예
AWS Cost Optimization Hub	예	아니요	예	예
AWS Data Exchange	아니요	아니요	예	예
Amazon Data Firehose	아니요	아니요	예	예
Amazon Data Lifecycle Manager	예	아니요	예	예
AWS Database Migration Service	예	아니요	아니요	예
AWS DataSync	예	예	예	예
Amazon DataZone	아니요	아니요	예	예
AWS Deadline Cloud	예	아니요	예	예
Amazon Detective	예	예	예	

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
Direct Connect	예	예	아니요	예
Directory Service	아니요	아니요	예	예
Amazon EBS 다이렉트 API	예	아니요	예	예
Amazon EC2	예	예	예	아니요
EC2 Image Builder	예	예	예	예
Amazon ECR	예	아니요	예	아니요
Amazon ECS	예	예	예	예
Amazon EFS	예	예	예	예
Amazon EKS	부분적	부분적	예	예
Amazon EMR	아니요	아니요	예	예
AWS Elastic Beanstalk	예	아니요	예	예
AWS Elastic Disaster Recovery	아니요	아니요	예	예
Elastic Load Balancing	부분적	부분적	예	아니요
Amazon ElastiCache	예	예	아니요	예
AWS Elemental MediaConvert	아니요	아니요	예	예
AWS Elemental MediaConnect	예	예	예	부분적

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS 최종 사용자 메시징 소셜	예	아니요	예	아니요
AWS Entity Resolution	예	아니요	예	예
Amazon EventBridge	아니요	아니요	예	예
AWS Fargate	예	아니요	예	예
Amazon FSx	아니요	아니요	예	예
Amazon GameLift Streams	예	아니요	예	예
AWS Global Accelerator	예	아니요	예	
AWS Glue	예	아니요	아니요	예
AWS Glue DataBrew	아니요	아니요	예	예
Amazon Managed Grafana [3]	예	아니요	예	예
AWS Ground Station [4]	예	아니요	예	예
Amazon GuardDuty	아니요	아니요	예	예
AWS HealthImaging	아니요	아니요	예	예
AWS HealthLake	아니요	아니요	예	예
AWS HealthOmics	아니요	아니요	예	예
AWS Identity and Access Managemen t(IAM)	예	예	예	아니요

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS IAM Access Analyzer	예	아니요	예	예
AWS IAM Identity Center	예	아니요	예	
AWS IAM Roles Anywhere	아니요	아니요	예	예
Amazon Inspector –	예	예	예	예
Amazon Interactive Video Service (IVS) [5]	예	아니요	예	예
AWS IoT Core	예	아니요	예	예
AWS IoT Device Defender	예	아니요	예	아니요
AWS IoT Device Management	예	아니요	예	아니요
AWS IoT FleetWise	예	아니요	예	예
AWS IoT Greengrass	예	아니요	예	아니요
AWS IoT SiteWise	예	아니요	예	예
AWS IoT TwinMaker	예	아니요	예	예
AWS IoT 무선	예	아니요	예	예
Amazon Kendra	아니요	아니요	예	아니요
AWS Key Management Service	예	부분적	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
Amazon Keyspaces	예	예	예	예
Amazon Keyspaces CDC streams	예	예	예	예
Amazon Kinesis Data Streams	예	아니요	예	예
AWS Lake Formation	아니요	아니요	아니요	예
AWS Lambda	예	아니요	예	예
AWS Launch Wizard	아니요	아니요	예	예
AWS License Manager	아니요	아니요	예	예
Amazon Lightsail	예	예	예	예
Amazon Location Service	아니요	아니요	예	예
Amazon MQ	아니요	아니요	예	예
Amazon MWAA	아니요	아니요	예	예
Amazon Macie	예	아니요	예	예
AWS Mainframe Modernization	예	아니요	예	예
Amazon Managed Grafana	아니요	아니요	예	예
– Amazon Managed Service for Prometheus	예	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS Migration Hub Orchestrator	아니요	아니요	예	예
AWS Network Firewall	예	예	아니요	예
AWS Network Manager	예	아니요	예	아니요
Amazon OpenSearch Service	예	아니요	예	
AWS Organizations	예	아니요	예	예
AWS Outposts	아니요	아니요	예	예
Amazon Personalize	예	아니요	예	예
Amazon Pinpoint	예	아니요	예	예
Amazon Polly	예	아니요	예	예
AWS 가격표	아니요	아니요	예	아니요
AWS Private Certificate Authority	예	아니요	예	예
액티브 디렉터리 AWS Private CA 커넥터	예	아니요	예	예
AWS Private CA SCEP 용 커넥터	예	아니요	예	예
AWS PrivateLink	예	예	예	
Amazon Q Business	아니요	아니요	예	아니요
Amazon RDS	예	아니요	예	아니요

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
Amazon RDS Data API	아니요	아니요	예	예
Amazon RDS Performance Insights	아니요	아니요	예	예
Amazon Redshift	예	아니요	예	
Amazon Rekognition	아니요	아니요	예	예
휴지통	예	아니요	예	예
AWS re:Post 프라이빗	예	아니요	예	예
AWS Resource Access Manager	예	아니요	예	예
AWS 리소스 탐색기	예	아니요	예	아니요
AWS Resource Groups	예	예	예	예
AWS Resource Groups Tagging API	예	예	예	예
Amazon Route 53	예	예	예	예
Amazon S3	예	아니요	예	아니요
Outposts에서의 Amazon S3	아니요	아니요	예	아니요
Amazon SageMaker	아니요	아니요	예	예
AWS Secrets Manager	예	아니요	예	예
AWS Security Hub	아니요	아니요	예	예
Amazon Security Lake	예	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
AWS Security Token Service	예	아니요	예	예
AWS Service Catalog	아니요	아니요	예	예
AWS Shield	예	예	아니요	예
Amazon Simple Email Service	예	아니요	예	예
Amazon Simple Notification Service	예	아니요	예	예
Amazon Simple Queue Service	예	아니요	예	예
Amazon Simple Workflow Service	예	아니요	예	예
AWS Site-to-Site VPN	예	아니요	예	아니요
AWS Snow Family	아니요	아니요	예	
AWS Step Functions	예	아니요	예	예
AWS Storage Gateway	예	예	예	예
AWS Systems Manager	아니요	아니요	예	예
AWS Systems Manager Incident Manager	아니요	아니요	예	예
SAP용 AWS Systems Manager	아니요	아니요	예	예
Amazon Textract	아니요	아니요	예	예

서비스 이름	듀얼 스택 지원	IPv6 전용 지원	IPv6를 지원하는 퍼블릭 엔드포인트	IPv6를 지원하는 프라이빗 엔드포인트 [1]
Amazon Timestream	아니요	아니요	예	예
Amazon Transcribe	예	예	예	예
AWS Transfer Family [6]	예	아니요	예	예
AWS Transit Gateway	예	아니요	예	아니요
Amazon Translate	예	예	예	예
AWS Trusted Advisor	아니요	아니요	예	예
AWS 사용자 알림	아니요	아니요	예	예
Amazon Verified Permissions	예	아니요	예	예
AWS의 VMware Cloud	아니요	아니요	예	예
Amazon VPC	예	예	예	아니요
Amazon VPC Lattice	아니요	아니요	예	예
AWS WAF	예	예	아니요	
AWS WAFV2	아니요	아니요	예	예
AWS Well-Architected Tool	아니요	아니요	예	예
Amazon WorkMail	아니요	아니요	예	예
Amazon WorkSpaces	예	아니요	예	예
AWS X-Ray	예	아니요	예	예

[1] 빈 셀은 해당 서비스가 AWS PrivateLink와 통합되지 않음을 나타냅니다.

[2] 이 항목은 AWS AppSync SDK API를 통해 AWS AppSync GraphIQL 및 Event API 구성 작업에 대한 IPv6 지원을 나타냅니다. IPv6는 고객 관리형 AWS AppSync GraphQL 및 Event API에 대한 클라이언트 연결에는 지원되지 않습니다.

[3] 이 항목은 워크스페이스 및 워크스페이스 권한 업데이트와 같은 Grafana 워크스페이스 관리 작업에 대한 IPv6 지원을 나타냅니다. 대시보드 생성 및 편집, 데이터 소스 쿼리와 같은 일반적인 Grafana 워크스페이스 작업에 대한 IPv6 지원은 없습니다.

[4] 이 항목은 AWS Ground Station API을 직접적으로 호출하는 것과 같은 AWS Ground Station 컨트롤 플레인 작업에 대한 IPv6 지원을 나타냅니다. IPv6는 AWS Ground Station 데이터 플레인에서 지원되지 않으므로 IPv4를 통해 데이터를 전송하는 리소스(예: Amazon EC2 인스턴스)에 액세스할 수 있어야 합니다.

[5] 이 항목은 IVS 엔드포인트를 호출하는 등과 같은 Amazon IVS 컨트롤 플레인 작업에 대한 IPv6 지원을 나타냅니다.

[6] AWS Transfer Family의 IPv6 지원에 대한 자세한 내용은 IPv6 제한 사항을 참조하세요.

추가 IPv6 지원

컴퓨팅

- Amazon EC2는 Nitro System 기반 인스턴스를 IPv6 전용 서브넷에서 시작할 수 있도록 지원합니다.
- Amazon EC2는 인스턴스 메타데이터 서비스(IMDS) 및 Amazon Time Sync Service에 대한 IPv6 엔드포인트를 제공합니다.

게임 개발

- Amazon GameLift Streams는 Microsoft Windows Server 2022 기본 런타임에서 IPv6를 통한 스트리밍을 지원합니다.

네트워킹 및 콘텐츠 전송

- Amazon VPC는 IPv6 전용 서브넷 생성을 지원합니다.
- Amazon VPC는 서브넷에서 DNS64를 지원하고 NAT 게이트웨이에서 NAT64를 지원하여 IPv6 AWS 리소스가 IPv4 리소스와 통신할 수 있도록 돕습니다.

보안, 자격 증명 및 규정 준수

- Amazon Detective는 네트워크 관련 조사 결과 및 엔터티 프로필에서 IPv6 주소를 지원합니다.
- AWS Identity and Access Management(IAM)는 IAM ID 기반 정책에서 IPv6 주소를 지원합니다.
- Amazon Macie는 개인 식별 정보(PII)에서 IPv6 주소를 지원합니다.
- Amazon Security Lake는 로그 소스 및 구독자의 모든 작업에서 IPv6 주소를 지원합니다.

관리 및 거버넌스

- AWS CloudTrail 레코드에는 소스 IPv6 정보가 포함됩니다.
- AWS CLI v2는 IPv6 전용 클라이언트에 대해 IPv6 연결을 통한 다운로드를 지원합니다.

자세히 알아보기

- AWS에서의 IPv6
- 듀얼 스택 및 IPv6 전용 Amazon VPC 참조 아키텍처(PDF)

가상 프라이빗 클라우드 구성

Amazon Virtual Private Cloud(VPC)는 기본 구성 요소로서, 이를 통해 AWS 클라우드 내에서 논리적으로 격리된 가상 네트워크를 프로비저닝할 수 있습니다. 자체 VPC를 생성하면 네트워킹 환경을 완전히 제어할 수 있습니다. 예를 들어, IP 주소 범위, 서브넷, 라우팅 테이블 및 연결 옵션을 정의할 수 있습니다.

AWS 계정에는 각 AWS 리전의 기본 VPC가 포함되어 있습니다. 이 기본 VPC는 리소스를 빠르게 실행할 수 있는 편리한 옵션으로 설정이 미리 구성되어 있습니다. 그러나 기본 VPC가 장기적인 네트워킹 요구 사항에 항상 부합하는 것은 아닙니다. 이 경우 추가 VPC를 생성하는 것이 유리할 수 있습니다.

추가 VPC를 생성하면 새 AWS 계정마다 프로비저닝되는 기본 VPC를 사용하는 것보다 나은 몇 가지 이점이 있습니다. 자체 관리형 VPC를 사용하면 다중 계층 애플리케이션 구현, 온프레미스 리소스에 연결, 부서 또는 사업부별 워크로드 분리 등 특정 요구사항에 맞게 네트워크 토폴로지를 정확하게 설계할 수 있습니다.

또한 VPC를 여러 개 생성하면 서로 다른 애플리케이션 또는 사업부 간에 보안과 격리를 강화할 수 있습니다. 각 VPC는 별도의 가상 네트워크 역할을 하므로 각 환경에 맞는 고유한 보안 정책, 액세스 제어 및 라우팅 구성을 적용할 수 있습니다.

궁극적으로 기본 VPC를 사용할지 아니면 하나 이상의 사용자 지정 VPC를 생성할지는 특정 애플리케이션 요구 사항, 보안 요구 사항 및 장기적인 확장성 목표에 따라 결정해야 합니다. 시간을 투자하여 VPC 인프라를 신중하게 설계하면 강력하고 안전하며 적응력이 뛰어난 클라우드 네트워킹 기반을 구축할 수 있습니다.

내용

- VPC 기초
- VPC 구성 옵션
- 기본 VPC
- VPC 생성
- VPC의 리소스 시각화
- VPC에서 CIDR 블록 추가 또는 제거
- Amazon VPC의 DHCP 옵션 세트
- VPC의 DNS 속성
- VPC의 네트워크 주소 사용량

- 다른 계정과 VPC 서브넷 공유
- 로컬 영역, Wavelength 영역 또는 Outpost로 VPC 확장
- VPC 삭제
- 콘솔 투 코드를 사용하여 VPC 콘솔 작업에서 코드형 인프라 생성

VPC 기초

VPC는 리전의 모든 가용 영역에 적용됩니다. VPC를 생성한 후 각 가용 영역에 하나 이상의 서브넷을 추가할 수 있습니다. 자세한 내용은 서브넷 섹션을 참조하세요.

내용

- VPC IP 주소 범위
- VPC 다이어그램
- VPC 리소스

VPC IP 주소 범위

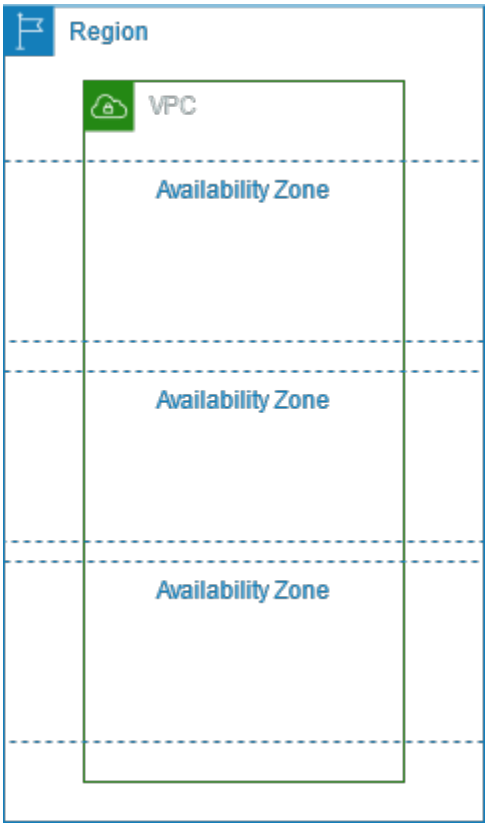
VPC를 생성할 때 다음과 같이 IP 주소를 지정합니다.

- IPv4 전용 - VPC에 IPv4 CIDR 블록은 있지만 IPv6 CIDR 블록은 없습니다.
- 듀얼 스택 - VPC에 IPv4 CIDR 블록 및 IPv6 CIDR 블록이 둘 다 있습니다.

자세한 내용은 VPC 및 서브넷의 IP 주소 지정 섹션을 참조하세요.

VPC 다이어그램

다음 다이어그램은 추가 VPC 리소스가 없는 VPC를 보여 줍니다. VPC 구성에 대한 예시는 예시 섹션을 참조하세요.



VPC 리소스

각 VPC는 다음 리소스와 함께 제공됩니다.

- 기본 DHCP 옵션 세트
- 기본 네트워크 ACL
- 기본 보안 그룹
- 기본 라우팅 테이블

VPC에 대해 다음 리소스를 생성할 수 있습니다.

- 네트워크 ACL
- 사용자 지정 라우팅 테이블
- 보안 그룹
- 인터넷 게이트웨이
- NAT 게이트웨이

VPC 구성 옵션

VPC를 생성할 때 다음 구성 옵션을 지정할 수 있습니다.

가용 영역

AWS 리전에 중복 전원, 네트워킹 및 연결이 있는 하나 이상의 개별 데이터 센터입니다. 여러 AZ를 사용하여 단일 데이터 센터에서 가능한 것보다 더 높은 가용성, 내결함성 및 확장성을 갖춘 프로덕션 애플리케이션 및 데이터베이스를 운영할 수 있습니다. 여러 AZ의 서브넷에서 실행되는 애플리케이션을 분할하면 정전, 낙뢰, 토네이도, 지진과 같은 문제로부터 더 잘 격리되고 보호됩니다.

CIDR 블록

VPC와 서브넷의 IP 주소 범위를 지정해야 합니다. 자세한 내용은 VPC 및 서브넷의 IP 주소 지정 섹션을 참조하세요.

DNS 옵션

서브넷에서 시작된 EC2 인스턴스의 퍼블릭 IPv4 DNS 호스트 이름이 필요한 경우 두 DNS 옵션을 모두 활성화해야 합니다. 자세한 내용은 VPC의 DNS 속성 섹션을 참조하세요.

- DNS 호스트 이름 활성화: VPC에서 시작되는 EC2 인스턴스는 퍼블릭 IPv4 주소에 해당하는 퍼블릭 DNS 호스트 이름을 받습니다.
- DNS 확인 활성화: 프라이빗 DNS 호스트 이름에 대한 DNS 확인은 Route◦53 Resolver라고 하는 Amazon DNS 서버에서 VPC에 제공합니다.

인터넷 게이트웨이

인터넷에 VPC를 연결합니다. 퍼블릭 서브넷의 인스턴스는 서브넷 라우팅 테이블에 인터넷으로 향하는 트래픽을 인터넷 게이트웨이로 전송하는 경로가 포함되어 있으므로 인터넷에 액세스할 수 있습니다. 인터넷에서 서버에 직접 연결할 필요가 없는 경우 퍼블릭 서브넷에 배포하면 안 됩니다. 자세한 내용은 인터넷 게이트웨이를 참조하세요.

이름

VPC 및 기타 VPC 리소스에 지정한 이름은 이름 태그를 생성하는는 데 사용됩니다. 콘솔에서 이름 태그 자동 생성 기능을 사용하면 태그 값은 *name* - *resource* 형식을 갖습니다.

NAT 게이트웨이

프라이빗 서브넷의 인스턴스가 아웃바운드 트래픽을 전송할 수 있도록 하지만 인터넷의 리소스가 인스턴스에 연결하는 것을 방지합니다. 프로덕션 환경에서는 각 활성 AZ에 NAT 게이트웨이를 배포하는 것이 좋습니다. 자세한 내용은 NAT 게이트웨이 섹션을 참조하세요.

라우팅 테이블

서브넷 또는 게이트웨이의 네트워크 트래픽이 전송되는 위치를 결정하는 라우팅이라는 규칙 집합을 포함합니다. 자세한 내용은 라우팅 테이블을 참조하세요.

서브넷

VPC의 IP 주소 범위입니다. EC2 인스턴스와 같은 AWS 리소스를 서브넷에서 실행할 수 있습니다. 각 서브넷은 단일 가용 영역 내에서만 존재합니다. 두 개 이상의 가용 영역에서 인스턴스를 시작하면 단일 가용 영역의 장애로부터 애플리케이션을 보호할 수 있습니다.

퍼블릭 서브넷에는 인터넷 게이트웨이로 직접 연결되는 경로가 있습니다. 퍼블릭 서브넷의 리소스는 퍼블릭 인터넷에 액세스할 수 있습니다. 프라이빗 서브넷에는 인터넷 게이트웨이로 직접 연결되는 경로가 없습니다. 프라이빗 서브넷의 리소스에는 퍼블릭 인터넷에 액세스하기 위해 NAT 장치와 같은 다른 구성 요소가 필요합니다.

자세한 내용은 서브넷을 참조하세요.

Tenancy

이 옵션은 VPC로 시작하는 EC2 인스턴스가 다른 AWS 계정과 공유되는 하드웨어에서 실행되는지 아니면 사용자 전용 하드웨어에서 실행되는지를 정의합니다. VPC의 테넌시를 `Default`로 선택하면 이 VPC로 시작된 EC2 인스턴스는 인스턴스를 시작할 때 지정된 테넌시 속성을 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 정의된 파라미터를 사용하여 인스턴스 시작을 참조하세요. VPC의 테넌시를 `Dedicated`로 선택하면 인스턴스는 항상 전용 하드웨어에서 전용 인스턴스로 실행됩니다. AWS Outposts를 사용하는 경우 Outpost에 프라이빗 연결이 필요합니다. `Default` 테넌시를 사용해야 합니다.

기본 VPC

Amazon VPC를 사용하기 시작하는 경우 각 AWS 리전에 기본 VPC가 있습니다. 기본 VPC는 각 가용 영역의 퍼블릭 서브넷, 인터넷 게이트웨이 및 DNS 확인 활성화 설정과 함께 제공됩니다. 따라서 기본 VPC로 Amazon EC2 인스턴스를 즉시 시작할 수 있습니다. 기본 VPC에서 Elastic Load Balancing, Amazon RDS, Amazon EMR 같은 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 VPC에 서브넷을 추가할 수 있습니다. 자세한 내용은 the section called “서브넷 생성” 섹션을 참조하세요.

내용

- 기본 VPC 구성 요소
- 기본 서브넷
- 기본 VPC와 기본 서브넷 작업

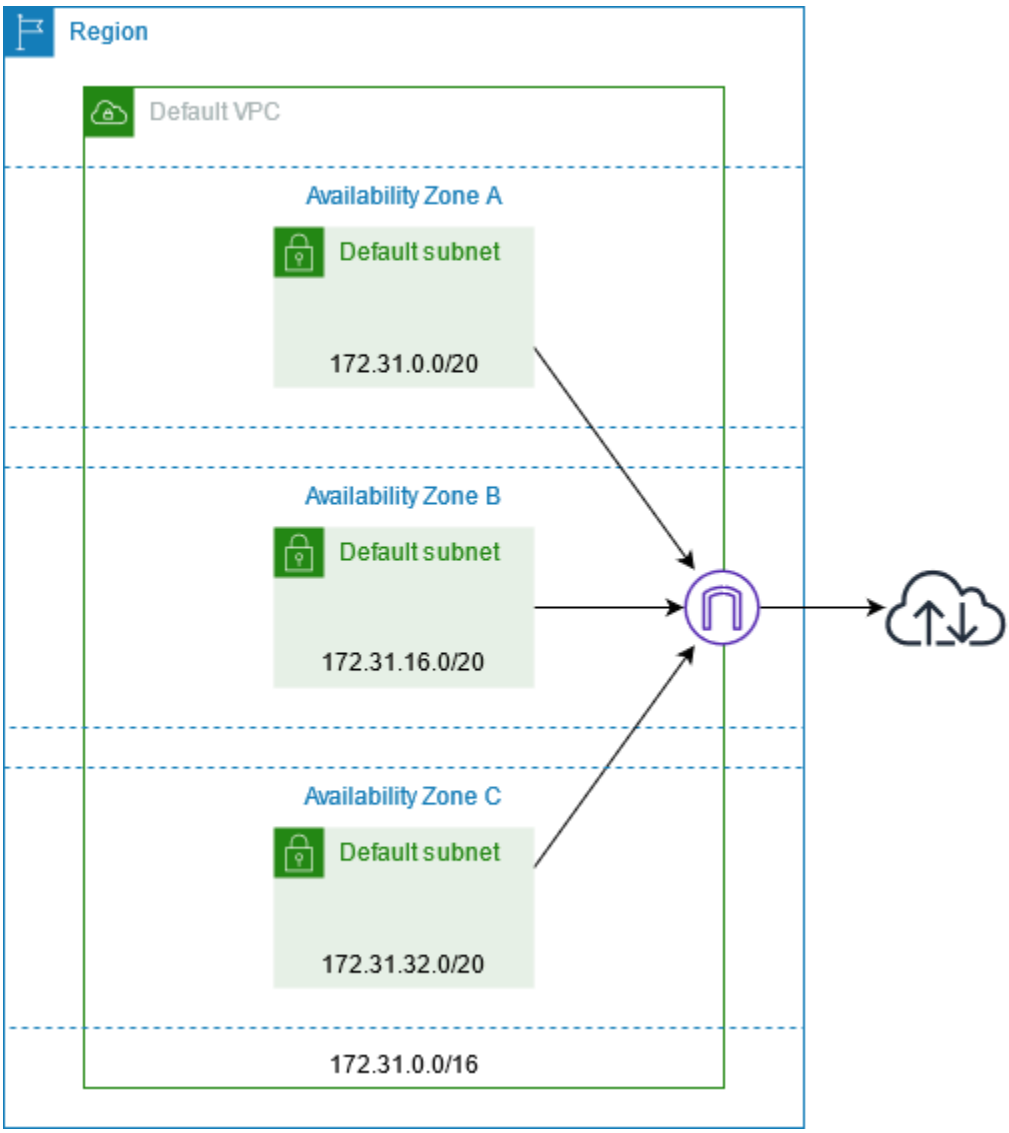
기본 VPC 구성 요소

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 게이트웨이를 만들어 기본 VPC에 연결합니다.
- 기본 라우팅 테이블에 모든 트래픽(0.0.0.0/0)이 인터넷 게이트웨이로 전달되는 경로를 추가합니다.
- 기본 보안 그룹을 만들어 기본 VPC와 연결합니다.
- 네트워크 ACL(액세스 제어 목록)을 생성하여 기본 VPC와 연결합니다.
- AWS 계정에서 설정된 기본 DHCP 옵션을 기본 VPC와 연결합니다.

> **Note**
>
> - 위의 리소스는 사용자 대신 Amazon에서 생성합니다. 사용자가 이러한 작업을 수행하는 것이 아니므로 여기에는 IAM 정책이 적용되지 않습니다. 예를 들어 CreateInternetGateway를 호출하는 기능을 거부하는 IAM 정책이 있고 CreateDefaultVpc를 호출하면 기본 VPC의 인터넷 게이트웨이가 여전히 생성됩니다. Amazon이 인터넷 게이트웨이를 생성하지 못하도록 하려면 CreateDefaultVpc 및 CreateInternetGateway를 거부해야 합니다.
> - 계정의 인터넷 게이트웨이와 주고받는 모든 트래픽을 차단하려면 VPC 및 서브넷에 대한 퍼블릭 액세스 차단 섹션을 참조하세요.

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



다음 표는 기본 VPC에 대한 기본 라우팅 테이블의 경로를 보여줍니다.

대상 주소	대상
172.31.0.0/16	로컬
0.0.0.0/0	*internet_gateway_id*

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.
- AWS Site-to-Site VPN 연결을 추가합니다.
- 더 많은 IPv4 CIDR 블록을 추가합니다.
- Direct Connect 게이트웨이를 사용하여 원격 리전의 VPC에 액세스합니다. Direct Connect 게이트웨이 옵션에 대한 자세한 내용은 AWS Direct Connect 사용 설명서의 Direct Connect 게이트웨이를 참조하세요.

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.

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 게이트웨이로의 라우팅을 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 내용은 서브넷의 IP 주소 지정 속성 수정 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 내용은 기본 서브넷 만들기 섹션을 참조하세요.

기본 VPC와 기본 서브넷 작업

이 섹션에서는 기본 VPC와 기본 서브넷으로 작업하는 방법을 설명합니다.

내용

- 기본 VPC와 기본 서브넷 보기
- 기본 VPC 생성
- 기본 서브넷 만들기
- 기본 서브넷과 기본 VPC 삭제

기본 VPC와 기본 서브넷 보기

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

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(AWS Tools for Windows PowerShell)를 사용합니다.

`isDefault` 필터를 포함하여 명령을 사용하고 필터 값을 `true`로 설정합니다.

명령줄을 사용하여 기본 서브넷을 나타내려면

- describe-subnets(AWS CLI)를 사용합니다.
- Get-EC2Subnet(AWS Tools for Windows PowerShell)를 사용합니다.

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

기본 VPC 생성

기본 VPC를 삭제한 경우 새로 만들 수 있습니다. 기본 VPC를 삭제하면 복구할 수 없으며, 기본 VPC가 아닌 기존 VPC를 기본 VPC로 설정할 수도 없습니다.

기본 VPC를 만들면 각 가용 영역의 기본 서브넷을 비롯하여 기본 VPC의 표준 구성 요소 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 `172.31.0.0/20`을 포함하는 서브넷이 `us-east-2a`에 생성된 경우, 새로운 기본 VPC에서는 `us-east-2b`에 생성될 수 있습니다.

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

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
```

다음은 예제 출력입니다.

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

또는 Windows PowerShell용 New-EC2DefaultVpc 도구나 CreateDefaultVpc Amazon EC2 API 작업을 사용해도 됩니다.

기본 서브넷 만들기

> **Note**
>
> AWS Management Console을 사용하여 기본 서브넷을 생성할 수 없습니다.

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 `/20`인 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 `/20`을 생성할 충분한 주소 공간이 기본 VPC에 없는 경우 요청은 실패합니다. 더 많은 주소 공간이 필요한 경우 VPC에 IPv4 CIDR 블록을 추가할 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 내용은 서브넷에서 IPv6 CIDR 블록 추가 또는 제거 섹션을 참조하세요.

AWS CLI를 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
```

다음은 예제 출력입니다.

```
{
```

```
    "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
    }
}
```

AWS CLI 설정에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 참조하세요.

또는 Windows PowerShell용 New-EC2DefaultSubnet 도구 명령이나CreateDefaultSubnet Amazon EC2 API 작업을 사용할 수도 있습니다.

기본 서브넷과 기본 VPC 삭제

기본 서브넷이나 기본 VPC는 다른 서브넷 또는 VPC처럼 삭제할 수 있습니다. 그러나 기본 서브넷이나 기본 VPC를 삭제하면 인스턴스를 시작할 때 VPC 중 하나에 서브넷을 명시적으로 지정해야 합니다. 다른 VPC가 없으면 하나 이상의 가용 영역에 서브넷이 있는 VPC를 생성해야 합니다. 자세한 내용은 VPC 생성 섹션을 참조하세요.

기본 VPC를 삭제한 경우 새로 만들 수 있습니다. 자세한 내용은 기본 VPC 생성 단원을 참조하십시오.

기본 서브넷을 삭제한 경우 새로 만들 수 있습니다. 자세한 내용은 기본 서브넷 만들기 섹션을 참조하세요. 새로운 기본 서브넷이 정상적으로 동작하는지 확인하려면 서브넷 속성을 수정하여 해당 서브넷에서 시작되는 인스턴스에 퍼블릭 IP 주소를 할당하십시오. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정 단원을 참조하십시오. 가용 영역당 기본 서브넷은 한 개만 가질 수 있습니다. 기본 VPC가 아닌 VPC에는 기본 서브넷을 생성할 수 없습니다.

VPC 생성

다음 절차에 따라 Virtual Private Cloud(VPC)를 생성합니다. VPC에서 AWS 리소스를 생성하려면 먼저 VPC에 서브넷, 라우팅 테이블, 게이트웨이와 같은 추가 리소스가 있어야 합니다.

내용

- VPC 및 기타 VPC 리소스 생성
- VPC만 생성
- AWS CLI를 사용하여 VPC 생성

VPC 수정에 대한 자세한 내용은 the section called “CIDR 블록 추가 또는 제거” 섹션을 참조하세요.

VPC 및 기타 VPC 리소스 생성

다음 절차에 따라 VPC 및 애플리케이션을 실행하는 데 필요한 추가 VPC 리소스(예: 서브넷, 라우팅 테이블, 인터넷 게이트웨이, NAT 게이트웨이)를 생성합니다. VPC 구성에 대한 예시는 예시 섹션을 참조하세요.

콘솔을 사용하여 VPC, 서브넷 및 기타 VPC 리소스를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드에서 VPC 생성을 선택합니다.
3. Resources to create(생성할 리소스)에서 VPC and more(VPC 등)를 선택합니다.
4. 이름 태그 자동 생성을 선택한 상태로 유지하여 VPC 리소스에 이름 태그를 생성하거나 선택을 취소하여 VPC 리소스에 고유한 이름 태그를 제공합니다.
5. IPv4 CIDR 블록에 VPC의 IPv4 주소 범위를 입력합니다. VPC는 IPv4 주소 범위를 가져야 합니다.
6. (선택 사항) IPv6 트래픽을 지원하려면 IPv6 CIDR 블록, Amazon에서 제공한 IPv6 CIDR 블록을 선택합니다.
7. 테넌시 옵션을 선택합니다. 이 옵션은 VPC로 시작하는 EC2 인스턴스가 다른 AWS 계정과 공유되는 하드웨어에서 실행되는지 아니면 사용자 전용 하드웨어에서 실행되는지를 정의합니다. VPC의 테넌시를 `Default`로 선택하면 이 VPC로 시작된 EC2 인스턴스에서는 인스턴스를 시작할 때 지정된 테넌시 속성을 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 정의된 파라미터를 사용하여 인스턴스 시작을 참조하세요. VPC의 테넌시를 `Dedicated`로 선택하면 인스턴스는 항상 전용 하드웨어에서 전용 인스턴스로 실행됩니다. AWS Outposts를 사용하는 경우 Outpost에 프라이빗 연결이 필요합니다. `Default` 테넌시를 사용해야 합니다.
8. 가용 영역(AZ) 수에서 프로덕션 환경의 경우 2개 이상의 가용 영역에 서브넷을 프로비저닝하는 것이 좋습니다. 서브넷의 AZ를 선택하려면 AZ 사용자 지정을 확장합니다. 그렇지 않으면 AWS가 선택하도록 합니다.
9. 서브넷을 구성하려면 퍼블릭 서브넷 수 및 프라이빗 서브넷 수의 값을 선택합니다. 서브넷의 IP 주소 범위를 선택하려면 서브넷 CIDR 블록 사용자 지정을 확장합니다. 그렇지 않으면 AWS가 선택하도록 합니다.

10. (선택 사항) 프라이빗 서브넷의 리소스가 IPv4를 통해 퍼블릭 인터넷에 액세스해야 하는 경우 NAT 게이트웨이에서 NAT 게이트웨이를 생성할 AZ 수를 선택합니다. 프로덕션 환경에서는 퍼블릭 인터넷에 액세스해야 하는 리소스가 있는 각 AZ에 NAT 게이트웨이를 배포하는 것이 좋습니다. NAT 게이트웨이와 관련된 비용이 있습니다. 자세한 내용은 NAT 게이트웨이 요금 섹션을 참조하세요.
11. (선택 사항) 프라이빗 서브넷의 리소스가 IPv6을 통해 퍼블릭 인터넷에 액세스해야 하는 경우 송신 전용 인터넷 게이트웨이에서 예를 선택합니다.
12. (선택 사항) VPC에서 직접 Amazon S3에 액세스해야 하는 경우 VPC 엔드포인트, S3 게이트웨이를 선택합니다. 이는 Amazon S3 게이트웨이 VPC 엔드포인트를 생성합니다. 자세한 내용은 AWS PrivateLink 안내서의 게이트웨이 엔드포인트를 참조하세요.
13. (선택 사항) DNS 옵션의 경우 도메인 이름 확인을 위한 두 가지 옵션이 기본적으로 모두 활성화됩니다. 기본값이 요구 사항을 충족하지 않는 경우 해당 옵션을 비활성화할 수 있습니다.
14. (선택 사항) VPC에 태그를 추가하려면 추가 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
15. 미리 보기 창에서 구성한 VPC 리소스 간의 관계를 시각화할 수 있습니다. 실선은 리소스 간의 관계를 나타냅니다. 점선은 NAT 게이트웨이, 인터넷 게이트웨이 및 게이트웨이 엔드포인트에 대한 네트워크 트래픽을 나타냅니다. VPC를 생성한 후에는 리소스 맵 탭을 사용하여 언제든지 VPC의 리소스를 이 형식으로 시각화할 수 있습니다. 자세한 내용은 VPC의 리소스 시각화 섹션을 참조하세요.
16. VPC 구성을 마치면 VPC 생성을 선택합니다.

VPC만 생성

다음 절차에 따라 Amazon VPC 콘솔을 사용하여 추가 VPC 리소스 없이 VPC를 생성합니다.

콘솔을 사용하여 추가 VPC 리소스 없이 VPC를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드에서 VPC 생성을 선택합니다.
3. 생성할 리소스에서 VPC 전용을 선택합니다.
4. (선택 사항) 이름 태그에 VPC의 이름을 입력합니다. 이렇게 하면 Name 키와 지정한 값으로 태그가 생성됩니다.
5. IPv4 CIDR 블록의 경우 다음 중 하나를 수행합니다.
   - IPv4 CIDR 수동 입력 을 선택하고 VPC에 대한 IPv4 주소 범위를 입력합니다.
   - IPAM에서 할당된 IPv4 CIDR 블록을 선택하고, Amazon VPC IP 주소 관리자(IPAM) IPv4 주소 풀과 넷마스크를 선택합니다. CIDR 블록의 크기는 IPAM 풀의 할당 규칙에 의해 제한됩니

다. IPAM는 AWS 워크로드의 IP 주소를 보다 쉽게 계획, 추적 및 모니터링할 수 있게 해주는 VPC 기능입니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서를 참조하세요.

IPAM을 사용하여 IP 주소를 관리하는 경우 이 옵션을 선택하는 것이 좋습니다. 그렇지 않으면 VPC에 지정한 CIDR 블록이 IPAM CIDR 할당과 겹칠 수 있습니다.

6. (선택 사항) 듀얼 스택 VPC를 생성하려면 VPC에 IPv6 주소 범위를 지정합니다. IPv6 CIDR 블록의 경우 다음 중 하나를 수행합니다.

   - Amazon VPC IP 주소 관리자를 사용 중이고 IPAM 풀에서 IPv6 CIDR을 프로비저닝하려는 경우 IPAM 할당 IPv6 CIDR 블록을 선택합니다. IPAM 할당 IPv6 CIDR 블록을 사용하여 VPC에 IPv6 CIDR을 프로비저닝하면 VPC 생성에서 연속 IPv6 CIDR의 이점을 누릴 수 있습니다. 연속 할당 CIDR은 순차적으로 할당되는 CIDR입니다. 이를 통해 보안 및 네트워킹 규칙을 간소화할 수 있습니다. IPv6 CIDRs은 액세스 제어 목록, 라우팅 테이블, 보안 그룹 및 방화벽과 같은 네트워킹 및 보안 구성 요소 전반에 걸쳐 단일 항목으로 집계할 수 있습니다.

     CIDR 블록에서 VPC에 IP 주소 범위를 프로비저닝하는 데는 두 가지 옵션이 있습니다.

     - 넷마스크 길이: CIDR의 넷마스크 길이를 선택하려면 이 옵션을 선택합니다. 다음 중 하나를 수행합니다.
       - IPAM 풀에 대해 선택된 기본 넷마스크 길이가 있는 경우 IPAM 넷마스크 길이 기본값을 선택하여 IPAM 관리자가 IPAM 풀에 설정한 기본 넷마스크 길이를 사용할 수 있습니다. 선택적 기본 넷마스크 길이 할당 규칙에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 리전 IPv6 풀 생성을 참조하세요.
       - IPAM 풀에 대해 선택된 기본 넷마스크 길이가 없는 경우 IPAM 풀 CIDR의 넷마스크 길이보다 더 구체적인 넷마스크 길이를 선택합니다. 예를 들어 IPAM 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 넷마스크 길이는/44~/60(/4씩 증가)입니다.
     - CIDR 선택: IPv6 주소를 수동으로 입력하려면 이 옵션을 선택합니다. IPAM 풀 CIDR 넷마스크 길이보다 더 구체적인 넷마스크 길이만 선택할 수 있습니다. 예를 들어 IPAM 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/60(/4씩 증가)입니다.
   - Amazon의 IPv6 주소 풀에서 IPv6 CIDR 블록을 요청하려면 Amazon 제공 IPv6 CIDR 블록을 선택합니다. 네트워크 경계 그룹(Network Border Group)에서 AWS가 IP 주소를 알리는 그룹을 선택합니다. Amazon은 /56의 고정 IPv6 CIDR 블록 크기를 제공합니다.
   - 이미 AWS에 가져온 IPv6 CIDR을 프로비저닝하려면 내가 소유한 IPv6 CIDR 을 선택합니다. 고유 IP 주소 범위를 AWS로 가져오는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의

고유 IP 주소 가져오기(BYOIP)를 참조하세요. CIDR 블록에 대한 다음과 같은 옵션을 사용하여 VPC의 IP 주소 범위를 프로비저닝할 수 있습니다.

- 기본 설정 없음: /56의 넷마스크 길이를 사용하려면 이 옵션을 선택합니다.
- CIDR 선택: IPv6 주소를 수동으로 입력하고 BYOIP CIDR 크기보다 더 구체적인 넷마스크 길이를 선택하려면 이 옵션을 선택합니다. 예를 들어 BYOIP 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/60(/4씩 증가)입니다.

7. (선택 사항) 테넌시 옵션을 선택합니다. 이 옵션은 VPC로 시작하는 EC2 인스턴스가 다른 AWS 계정과 공유되는 하드웨어에서 실행되는지 아니면 사용자 전용 하드웨어에서 실행되는지를 정의합니다. VPC의 테넌시를 `Default`로 선택하면 이 VPC로 시작된 EC2 인스턴스는 인스턴스를 시작할 때 지정된 테넌시 속성을 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 정의된 파라미터를 사용하여 인스턴스 시작을 참조하세요. VPC의 테넌시를 `Dedicated`로 선택하면 인스턴스는 항상 전용 하드웨어에서 전용 인스턴스로 실행됩니다. AWS Outposts를 사용하는 경우 Outpost에 프라이빗 연결이 필요합니다. `Default` 테넌시를 사용해야 합니다.
8. (선택 사항) VPC에 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
9. VPC 생성을 선택합니다.
10. VPC를 생성한 후 서브넷을 추가할 수 있습니다. 자세한 내용은 서브넷 생성 섹션을 참조하세요.

AWS CLI를 사용하여 VPC 생성

다음 절차에는 VPC를 만들기 위한 예제 AWS CLI 명령과 애플리케이션을 실행하는 데 필요한 추가 VPC 리소스가 포함되어 있습니다. 이 절차의 모든 명령을 실행하면 VPC, 퍼블릭 서브넷, 프라이빗 서브넷, 각 서브넷의 라우팅 테이블, 인터넷 게이트웨이, 송신 전용 인터넷 게이트웨이 및 퍼블릭 NAT 게이트웨이가 생성됩니다. 이러한 리소스가 모두 필요하지 않은 경우 필요한 예제 명령만 사용할 수 있습니다.

사전 조건

시작하기 전에 AWS CLI을(를) 설치하고 구성합니다. AWS CLI를 구성하면 AWS 보안 인증 정보를 입력하라는 메시지가 표시됩니다. 이 절차의 예제에서는 기본 리전도 구성했다고 가정합니다. 그렇지 않을 경우 각 명령에 `--region` 옵션을 적용합니다. 자세한 내용은 AWS CLI 설치 또는 업데이트 및 AWS CLI 구성을 참조하세요.

태그 지정

create-tags 명령을 사용하여 리소스를 생성한 후 리소스에 태그를 추가할 수 있습니다. 또는 다음과 같이 리소스 생성 명령에 `--tag-specification` 옵션을 추가할 수 있습니다.

```
--tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=my-project}]
```

AWS CLI를 사용하여 VPC 및 VPC 리소스 생성

1. 다음 create-vpc 명령을 사용하여 지정된 IPv4 CIDR 블록으로 VPC를 생성합니다.

   ```
   aws ec2 create-vpc --cidr-block 10.0.0.0/24 --query Vpc.VpcId --output text
   ```

   또는 듀얼 스택 VPC를 생성하려면 다음 예제와 같이 Amazon에서 제공하는 IPv6 CIDR 블록을 추가하는--amazon-provided-ipv6-cidr-block 옵션을 추가합니다.

   ```
   aws ec2 create-vpc --cidr-block 10.0.0.0/24 --amazon-provided-ipv6-cidr-block --
   query Vpc.VpcId --output text
   ```

   해당 명령은 새 VPC의 ID를 반환합니다. 다음은 예입니다.

   ```
   vpc-1a2b3c4d5e6f1a2b3
   ```

2. [듀얼 스택 VPC] 다음 describe-vpcs 명령을 사용하여 VPC에 연결된 IPv6 CIDR 블록을 가져옵니다.

   ```
   aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
    Vpcs[].Ipv6CidrBlockAssociationSet[].Ipv6CidrBlock --output text
   ```

   출력의 예시는 다음과 같습니다.

   ```
   2600:1f13:cfe:3600::/56
   ```

3. 사용 사례에 따라 하나 이상의 서브넷을 생성합니다. 프로덕션 환경에서는 2개 이상의 가용 영역에서 리소스를 시작하는 것이 좋습니다. 다음 명령 중 하나를 사용하여 각 서브넷을 생성합니다.

   - IPv4 전용 서브넷 - 특정 IPv4 CIDR 블록으로 서브넷을 생성하려면 다음 create-subnet 명령을 사용합니다.

     ```
     aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --cidr-block 10.0.1.0/20
      --availability-zone us-east-2a --query Subnet.SubnetId --output text
     ```

   - 듀얼 스택 서브넷 - 듀얼 스택 VPC를 생성한 경우 다음 명령과 같이 --ipv6-cidr-block 옵션을 사용하여 듀얼 스택 서브넷을 생성할 수 있습니다.

```
aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --cidr-block 10.0.1.0/20
 --ipv6-cidr-block 2600:1f13:cfe:3600::/64 --availability-zone us-east-2a --
query Subnet.SubnetId --output text
```

- IPv6 전용 서브넷 - 듀얼 스택 VPC를 생성한 경우 다음 명령과 같이 `--ipv6-native` 옵션을 사용하여 IPv6 전용 서브넷을 생성할 수 있습니다.

```
aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --ipv6-native --ipv6-
cidr-block 2600:1f13:cfe:3600::/64 --availability-zone us-east-2a --query
 Subnet.SubnetId --output text
```

해당 명령은 새 서브넷의 ID를 반환합니다. 다음은 예입니다.

```
subnet-1a2b3c4d5e6f1a2b3
```

4. 웹 서버용 또는 NAT 게이트웨이용 퍼블릭 서브넷이 필요한 경우 다음을 수행합니다.

   a. 다음 create-internet-gateway 명령을 사용하여 인터넷 게이트웨이를 생성합니다. 이 명령은 새 인터넷 게이트웨이의 ID를 반환합니다.

```
aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --
output text
```

   b. 다음 attach-internet-gateway 명령을 사용하여 VPC에 인터넷 게이트웨이를 연결합니다. 이전 단계에서 반환한 인터넷 게이트웨이 ID를 사용합니다.

```
aws ec2 attach-internet-gateway --vpc-id vpc-1a2b3c4d5e6f1a2b3 --internet-
gateway-id igw-id
```

   c. 다음 create-route-table 명령을 사용하여 퍼블릭 서브넷에 대한 사용자 지정 라우팅 테이블을 생성합니다. 이 명령은 새 라우팅 테이블의 ID를 반환합니다.

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

   d. 다음 create-route 명령을 사용하여 모든 IPv4 트래픽을 인터넷 게이트웨이로 보내는 라우팅 테이블의 경로를 생성합니다. 퍼블릭 서브넷의 라우팅 테이블 ID를 사용합니다.

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

e. 다음 associate-route-table 명령을 사용하여 라우팅 테이블을 퍼블릭 서브넷과 연결합니다. 퍼블릭 서브넷의 라우팅 테이블 ID와 퍼블릭 서브넷의 ID를 사용합니다.

```
aws ec2 associate-route-table --route-table-id rtb-id-public --subnet-
id subnet-id-public-subnet
```

5. [IPv6] 프라이빗 서브넷의 인스턴스가 IPv6를 통해 인터넷에 액세스할 수 있도록 송신 전용 인터넷 게이트웨이를 추가할 수 있지만(예: 소프트웨어 업데이트 받기) 인터넷의 호스트는 인스턴스에 액세스할 수 없습니다.

a. 다음 create-egress-only-internet-gateway 명령을 사용하여 송신 전용 인터넷 게이트웨이를 생성합니다. 이 명령은 새 인터넷 게이트웨이의 ID를 반환합니다.

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-1a2b3c4d5e6f1a2b3 --
query EgressOnlyInternetGateway.EgressOnlyInternetGatewayId --output text
```

b. 다음 create-route-table 명령을 사용하여 프라이빗 서브넷에 대한 사용자 지정 라우팅 테이블을 생성합니다. 이 명령은 새 라우팅 테이블의 ID를 반환합니다.

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

c. 다음 create-route 명령을 사용하여 모든 IPv6 트래픽을 송신 전용 인터넷 게이트웨이로 보내는 프라이빗 서브넷에 대한 라우팅 테이블의 경로를 생성합니다. 이전 단계에서 반환한 라우팅 테이블의 ID를 사용합니다.

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

d. 다음 associate-route-table 명령을 사용하여 라우팅 테이블을 프라이빗 서브넷과 연결합니다.

```
aws ec2 associate-route-table --route-table-id rtb-id-private --subnet-
id subnet-id-private-subnet
```

6. 프라이빗 서브넷의 리소스에 대한 NAT 게이트웨이가 필요한 경우 다음을 수행합니다.

a. 다음 allocate-address 명령을 사용하여 NAT 게이트웨이의 탄력적 IP 주소를 생성합니다.

```
aws ec2 allocate-address --domain vpc --query AllocationId --output text
```

b. 다음create-nat-gateway 명령을 사용하여 퍼블릭 서브넷의 NAT 게이트웨이를 생성합니다. 이전 단계에서 반환되는 할당 ID를 사용합니다.

```
aws ec2 create-nat-gateway --subnet-id subnet-id-public-subnet --allocation-
id eipalloc-id
```

c. (선택 사항) 5단계에서 프라이빗 서브넷에 대한 라우팅 테이블을 이미 생성한 경우 이 단계를 건너뜁니다. 그렇지 않으면 다음 create-route-table 명령을 사용하여 프라이빗 서브넷에 대한 라우팅 테이블을 생성합니다. 이 명령은 새 라우팅 테이블의 ID를 반환합니다.

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

d. 다음 create-route 명령을 사용하여 모든 IPv4 트래픽을 NAT 게이트웨이로 보내는 프라이빗 서브넷에 대한 라우팅 테이블의 경로를 생성합니다. 이 단계 또는 5단계에서 생성한 프라이빗 서브넷의 라우팅 테이블 ID를 사용합니다.

```
aws ec2 create-route --route-table-id rtb-id-private --destination-cidr-
block 0.0.0.0/0 --gateway-id nat-id
```

e. (선택 사항) 5단계에서 프라이빗 서브넷과 라우팅 테이블을 이미 연결한 경우 이 단계를 건너뜁니다. 그렇지 않으면 다음 associate-route-table 명령을 사용하여 라우팅 테이블을 프라이빗 서브넷과 연결합니다. 이 단계 또는 5단계에서 생성한 프라이빗 서브넷의 라우팅 테이블 ID를 사용합니다.

```
aws ec2 associate-route-table --route-table-id rtb-id-private --subnet-
id subnet-id-private-subnet
```

VPC의 리소스 시각화

이 섹션에서는 리소스 맵 탭을 사용하여 VPC 내 리소스의 시각적 표현을 볼 수 있는 방법을 설명합니다. 리소스 맵에는 다음 리소스가 표시됩니다.

- VPC
- 서브넷

  - 가용 영역은 문자로 표시됩니다.
  - 퍼블릭 서브넷은 녹색입니다.
  - 프라이빗 서브넷은 파란색입니다.

- 라우팅 테이블
- 인터넷 게이트웨이
- 외부 전용 인터넷 게이트웨이
- NAT 게이트웨이
- 게이트웨이 엔드포인트(Amazon S3 및 Amazon DynamoDB)

리소스 맵은 VPC 내 리소스 간의 관계와 서브넷에서 NAT 게이트웨이, 인터넷 게이트웨이 및 게이트웨이 엔드포인트로 트래픽이 흐르는 방식을 보여줍니다.

리소스 맵을 사용하여 VPC의 아키텍처를 이해하고 VPC에 있는 서브넷 수, 어떤 서브넷이 어떤 라우팅 테이블과 연결되어 있는지, 어떤 라우팅 테이블에 NAT 게이트웨이, 인터넷 게이트웨이 및 게이트웨이 엔드포인트에 대한 경로가 있는지 볼 수 있습니다.

또한 리소스 맵을 사용하여 NAT 게이트웨이에서 연결 해제된 프라이빗 서브넷 또는 인터넷 게이트웨이로 직접 연결되는 경로가 있는 프라이빗 서브넷과 같이 바람직하지 않거나 잘못된 구성을 찾아낼 수 있습니다. 리소스 맵 내에서 라우팅 테이블과 같은 리소스를 선택하고 해당 리소스의 구성을 편집할 수 있습니다.

VPC의 리소스를 시각화하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs를 선택합니다.
3. VPC를 선택합니다.
4. 리소스 맵 탭을 선택하여 리소스 시각화를 표시합니다.
5. 세부 정보 표시를 선택하여 기본적으로 표시되는 리소스 ID 및 영역 외의 세부 정보를 봅니다.
   - VPC: VPC에 할당된 IPv4 및 IPv6 CIDR 범위입니다.
   - 서브넷: 각 서브넷에 할당된 IPv4 및 IPv6 CIDR 범위입니다.
   - 라우팅 테이블: 라우팅 테이블의 서브넷 연결 및 경로 수입니다.
   - 네트워크 연결: 각 연결 유형과 관련된 세부 정보입니다.

- VPC에 퍼블릭 서브넷이 있는 경우 인터넷 게이트웨이를 사용하는 트래픽의 소스 및 대상 서브넷과 경로 수가 포함된 인터넷 게이트웨이 리소스가 있습니다.
- 외부 전용 인터넷 게이트웨이가 있는 경우 외부 전용 인터넷 게이트웨이를 사용하는 트래픽의 소스 및 대상 서브넷과 경로 수가 포함된 외부 전용 인터넷 게이트웨이 리소스가 있습니다.
- NAT 게이트웨이가 있는 경우 NAT 게이트웨이의 탄력적 IP 주소와 네트워크 인터페이스 수가 포함된 NAT 게이트웨이 리소스가 있습니다.
- 게이트웨이 엔드포인트가 있는 경우 엔드포인트를 사용하여 연결할 수 있는 AWS 서비스(Amazon S3 또는 Amazon DynamoDB)의 이름이 포함된 게이트웨이 엔드포인트 리소스가 있습니다.

6. 리소스 위로 마우스를 가져가면 리소스 간의 관계를 볼 수 있습니다. 실선은 리소스 간의 관계를 나타냅니다. 점선은 네트워크 연결에 대한 네트워크 트래픽을 나타냅니다.

VPC에서 CIDR 블록 추가 또는 제거

이 섹션에서는 VPC에서 IPv4 및 IPv6 CIDR 블록을 추가하거나 제거하는 방법을 설명합니다.

Important

- VPC는 기본적으로 최대 5개의 IPv4 블록과 5개의 IPv6 CIDR 블록을 가질 수 있지만 이 제한을 조정할 수 있습니다. 자세한 내용은 Amazon VPC 할당량 섹션을 참조하세요. VPC의 CIDR 블록 제한 사항에 대한 자세한 내용은 VPC CIDR 블록 섹션을 참조하세요.
- VPC에 하나 이상의 IPv4 CIDR 블록이 연결되어 있는 경우 VPC에서 IPv4 CIDR 블록을 제거할 수 있습니다. 기본 IPv4 CIDR 블록을 제거할 수 없습니다. 전체 CIDR 블록을 제거해야 합니다. CIDR 블록의 하위 집합이나 병합된 CIDR 블록 범위를 제거할 수 없습니다. 먼저 CIDR 블록에서 모든 서브넷을 삭제해야 합니다.
- VPC에서 IPv6 지원이 더 이상 필요 없지만 IPv4 리소스 생성 및 IPv4 리소스와의 통신을 위해 VPC를 계속 사용하려는 경우 IPv6 CIDR 블록을 제거할 수 있습니다.
- IPv6 CIDR 블록을 제거하려면 먼저 서브넷의 모든 인스턴스에 할당된 모든 IPv6 주소를 할당 해제해야 합니다.
- IPv6 CIDR 블록을 제거해도 IPv6 네트워킹을 위해 구성한 보안 그룹 규칙, 네트워크 ACL 규칙 또는 라우팅 테이블 경로는 자동으로 삭제되지 않습니다. 이 규칙 또는 경로는 수동으로 수정하거나 삭제해야 합니다.

콘솔을 사용하여 VPC에서 CIDR 블록을 추가하거나 제거하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 사용자 VPC(Your VPCs)를 선택합니다.
3. VPC를 선택한 다음 작업과 CIDR 편집을 차례로 선택합니다.
4. CIDR을 제거하려면 해당 CIDR 옆의 제거를 선택합니다.
5. CIDR을 추가하려면 새 IPv4 CIDR 추가 또는 새 IPv6 CIDR 추가를 선택합니다.
6. IPv4 CIDR 블록에 대한 CIDR을 추가하려면 다음 중 하나를 수행하세요.
   - IPv4 CIDR 수동 입력을 선택하고 IPv4 CIDR 블록을 입력합니다.
   - IPAM에 할당된 IPv4 CIDR을 선택하고 IPv4 IPAM 풀에서 CIDR을 선택합니다.
   - 저장을 선택합니다.
7. IPv6 CIDR 블록에 대한 CIDR을 추가하려면 다음 중 하나를 수행하세요.
   - Amazon VPC IP 주소 관리자를 사용 중이고 IPAM 풀에서 IPv6 CIDR을 프로비저닝하려는 경우 IPAM 할당 IPv6 CIDR 블록을 선택합니다. CIDR 블록에서 VPC에 IP 주소 범위를 프로비저닝하는 데는 두 가지 옵션이 있습니다.
     - 넷마스크 길이: CIDR의 넷마스크 길이를 선택하려면 이 옵션을 선택합니다. 다음 중 하나를 수행합니다.
       - IPAM 풀에 대해 선택된 기본 넷마스크 길이가 있는 경우 IPAM 넷마스크 길이 기본값을 선택하여 IPAM 관리자가 IPAM 풀에 설정한 기본 넷마스크 길이를 사용할 수 있습니다. 선택적 기본 넷마스크 길이 할당 규칙에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 리전 IPv6 풀 생성을 참조하세요.
       - IPAM 풀에 대해 선택된 기본 넷마스크 길이가 없는 경우 IPAM 풀 CIDR의 넷마스크 길이보다 더 구체적인 넷마스크 길이를 선택합니다. 예를 들어 IPAM 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 넷마스크 길이는/44~/60(/4씩 증가)입니다.
     - CIDR 선택: IPv6 주소를 수동으로 입력하려면 이 옵션을 선택합니다. IPAM 풀 CIDR 넷마스크 길이보다 더 구체적인 넷마스크 길이만 선택할 수 있습니다. 예를 들어 IPAM 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/60(/4씩 증가)입니다.
   - Amazon의 IPv6 주소 풀에서 IPv6 CIDR 블록을 요청하려면 Amazon 제공 IPv6 CIDR 블록을 선택합니다. 네트워크 경계 그룹(Network Border Group)에서 AWS가 IP 주소를 알리는 그룹을 선택합니다. Amazon은 /56의 고정 IPv6 CIDR 블록 크기를 제공합니다.

- 이미 AWS에 가져온 IPv6 CIDR을 프로비저닝하려면 내가 소유한 IPv6 CIDR 을 선택합니다. 자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 Amazon EC2에 고유 IP 주소 가져오기(BYOIP)를 참조하세요. CIDR 블록에서 VPC에 IP 주소 범위를 프로비저닝하는 데는 두 가지 옵션이 있습니다.
  - 기본 설정 없음: /56의 넷마스크 길이를 사용하려면 이 옵션을 선택합니다.
  - CIDR 선택: IPv6 주소를 수동으로 입력하고 BYOIP CIDR 크기보다 더 구체적인 넷마스크 길이를 선택하려면 이 옵션을 선택합니다. 예를 들어 BYOIP 풀 CIDR이 /50인 경우 VPC의 넷마스크 길이를 /52에서 /60 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/60(/4씩 증가)입니다.
  - 완료되면 CIDR 선택을 선택합니다.

8. 닫기를 선택하세요.
9. VPC에 CIDR 블록을 추가한 경우 새 CIDR 블록을 사용하는 서브넷을 생성할 수 있습니다. 자세한 내용은 서브넷 생성 섹션을 참조하세요.

AWS CLI를 사용하여 VPC에서 CIDR 블록을 연결하거나 연결 해제하려면 다음을 수행하세요.

associate-vpc-cidr-block 및 disassociate-vpc-cidr-block명령을 사용합니다.

Amazon VPC의 DHCP 옵션 세트

VPC의 네트워크 디바이스는 동적 호스트 구성 프로토콜(DHCP)을 사용합니다. DHCP 옵션 세트를 사용하여 가상 네트워크에서 네트워크 구성의 다음 측면을 제어할 수 있습니다.

- VPC의 디바이스에서 사용하는 DNS 서버, 도메인 이름 또는 NTP(Network Time Protocol) 서버입니다.
- VPC에서 DNS 확인이 활성화되었는지 여부.

내용

- DHCP란 무엇인가요?
- DHCP 옵션 세트 개념
- DHCP 옵션 세트로 작업

DHCP란 무엇인가요?

TCP/IP 네트워크의 모든 디바이스에는 네트워크를 통해 통신하기 위해 IP 주소가 필요합니다. 이전에는 네트워크의 각 디바이스에 IP 주소가 수동으로 할당되었습니다. 오늘날 IP 주소는 동적 호스트 구성 프로토콜(DHCP)을 사용하여 DHCP 서버에서 동적으로 할당됩니다.

EC2 인스턴스에서 실행되는 애플리케이션은 필요에 따라 Amazon DHCP 서버와 통신하여 IP 주소 임대 또는 기타 네트워크 구성 정보(예: Amazon DNS 서버의 IP 주소 또는 VPC의 라우터 IP 주소)를 검색할 수 있습니다.

DHCP 옵션 세트를 사용하여 Amazon DHCP 서버에서 제공하는 네트워크 구성을 지정할 수 있습니다.

애플리케이션이 Amazon IPv6 DHCP 서버에 직접 요청해야 하는 VPC 구성이 있는 경우 다음 사항에 유의하세요.

- 이중 스택 서브넷의 EC2 인스턴스는 IPv6 DHCP 서버에서만 해당 IPv6 주소를 검색할 수 있습니다. DNS 서버 이름 또는 도메인 이름과 같은 IPv6 DHCP 서버에서 추가 네트워크 구성을 검색할 수 없습니다.
- IPv6 전용 서브넷의 EC2 인스턴스는 IPv6 DHCP 서버에서 IPv6 주소를 검색하고 DNS 서버 이름, 도메인 이름 등의 추가 네트워킹 구성 정보를 검색할 수 있습니다.
- IPv6 전용 서브넷에 있는 EC2 인스턴스의 경우 DHCP 옵션 세트에 'AmazonProvidedDNS'가 명시적으로 언급된 경우 IPv4 DHCP 서버에서는 이름 서버로 169.254.169.253가 반환됩니다. 옵션 세트에 'AmazonProvidedDNS'가 누락된 경우 옵션 세트에 다른 IPv4 이름 서버가 언급되어 있는지 여부와 관계없이 IPv4 DHCP 서버에서는 주소가 반환되지 않습니다.

또한 Amazon DHCP 서버는 접두사 위임을 사용하여 VPC의 네트워크 인터페이스에 전체 IPv4 또는 IPv6 접두사를 제공할 수도 있습니다(Amazon EC2 사용 설명서의 Amazon EC2 네트워크 인터페이스에 접두사 할당 참조). IPv4 접두사 위임은 DHCP 응답에서 제공되지 않습니다. 인터페이스에 할당된 IPv4 접두사는 IMDS를 사용하여 검색할 수 있습니다(Amazon EC2 사용 설명서의 인스턴스 메타데이터 범주 참조).

DHCP 옵션 세트 개념

DHCP 옵션 세트는 EC2 인스턴스와 같은 VPC의 리소스가 가상 네트워크를 통해 통신하는 데 사용하는 네트워크 설정 그룹입니다.

각 리전에 기본 DHCP 옵션 세트가 있습니다. 각 VPC는 해당 리전의 기본 DHCP 옵션 세트를 사용합니다. 단, 사용자 지정 DHCP 옵션 세트를 만들어 VPC에 연결하거나 DHCP 옵션 세트 없이 VPC를 구성한 경우는 예외입니다.

구성된 DHCP 옵션 세트가 VPC에 없는 경우:

- Nitro 시스템에 구축된 EC2 인스턴스의 경우 AWS에서는 `169.254.169.253`이 기본 도메인 이름 서버로 구성됩니다.
- Xen에 구축된 EC2 인스턴스의 경우 도메인 이름 서버가 구성되지 않으며, DNS 서버에 액세스하는 권한이 VPC의 인스턴스에 없기 때문에 해당 인스턴스에서 인터넷에 액세스할 수 없습니다.

여러 VPC와 DHCP 옵션 세트를 연결할 수 있지만 각 VPC에는 하나의 DHCP 옵션 세트만 연결되어 있어야 합니다.

VPC를 삭제하면 VPC와 연결된 DHCP 옵션 세트가 VPC에서 분리됩니다.

내용

- 기본 DHCP 옵션 세트
- 사용자 정의 DHCP 옵션 세트

기본 DHCP 옵션 세트

기본 DHCP 옵션 세트에는 다음과 같은 설정이 포함되어 있습니다.

- 도메인 이름 서버: 네트워크 인터페이스가 도메인 이름 확인에 사용하는 DNS 서버입니다. 기본 DHCP 옵션 세트의 경우 이 항목은 언제나 `AmazonProvidedDNS`입니다. 자세한 내용은 Amazon DNS 서버 섹션을 참조하세요.
- 도메인 이름: 도메인 이름 시스템(DNS)을 사용하여 호스트 이름을 확인하는 경우 클라이언트가 사용해야 하는 도메인 이름입니다. EC2 인스턴스에 사용되는 도메인 이름에 대한 자세한 내용은 Amazon EC2 인스턴스 호스트 이름을 참조하세요.
- IPv6 선호 임대 시간: IPv6가 할당되어 실행 중인 인스턴스의 DHCPv6 임대 갱신 빈도입니다. 기본 임대 시간은 140초입니다. 임대 갱신은 일반적으로 임대 기간의 절반이 경과했을 때 발생합니다.

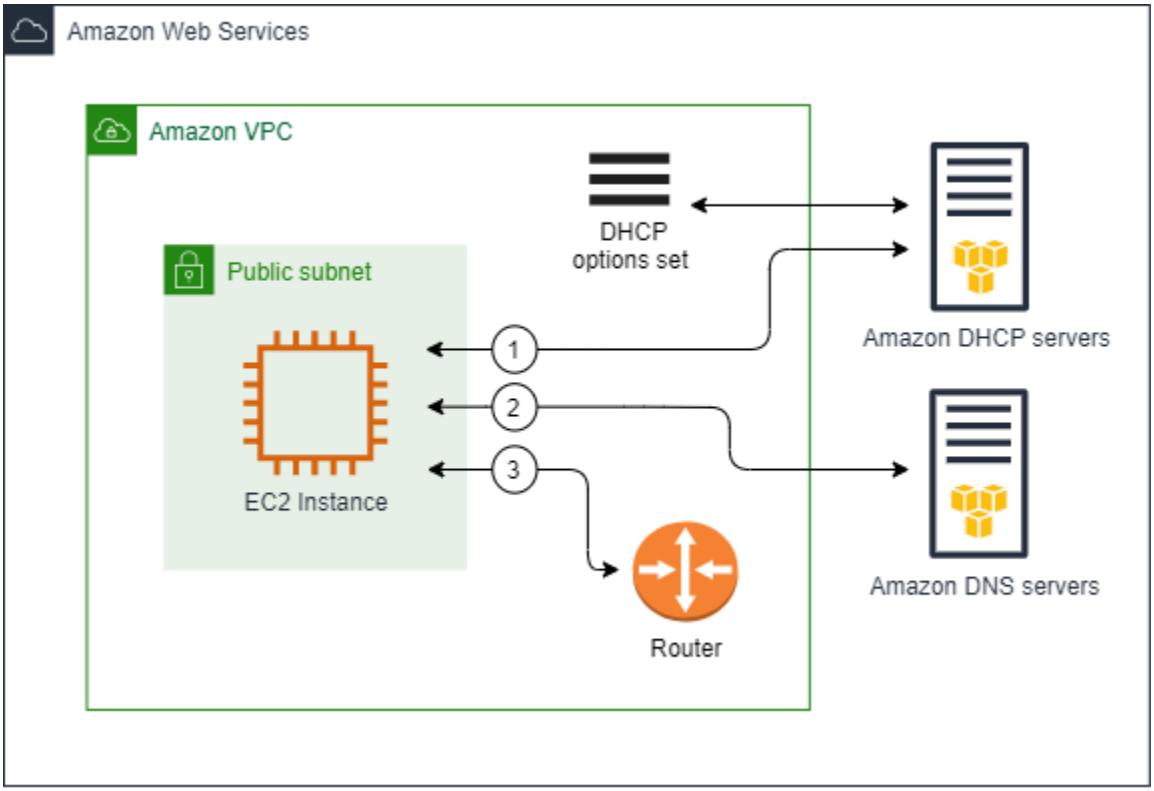
기본 DHCP 옵션 세트를 사용하는 경우 다음의 설정은 사용되지 않습니다. 하지만 EC2 인스턴스에 대한 기본값이 있습니다.

- NTP 서버: 기본적으로 EC2 인스턴스는 시간을 검색하기 위해 Amazon Time Sync Service를 사용합니다.
- NetBIOS 이름 서버: Windows에서 실행되는 EC2 인스턴스의 경우 NetBIOS 컴퓨터 이름은 네트워크에서 인스턴스를 식별하기 위해 인스턴스에 할당된 친숙한 이름입니다. NetBIOS 이름 서버는

NetBIOS를 이름 지정 서비스로 사용하는 네트워크의 NetBIOS 컴퓨터 이름과 네트워크 주소 간의 매핑 목록을 유지 관리합니다.

- NetBIOS 노드 유형: Windows에서 실행되는 EC2 인스턴스의 경우 인스턴스가 NetBIOS 이름을 IP 주소로 확인하는 데 사용하는 방법입니다.

기본 옵션 세트를 사용하는 경우 Amazon DHCP 서버는 기본 옵션 세트의 네트워크 설정을 사용합니다. VPC에서 인스턴스를 실행하면 다이어그램에 표시된 대로 (1) DHCP 서버와 상호 작용하고, (2) Amazon DNS 서버와 상호 작용하며, (3) VPC용 라우터를 통해 네트워크의 다른 장치에 연결됩니다. 인스턴스는 언제든지 Amazon DHCP 서버와 상호 작용하여 IP 주소 임대 및 추가 네트워크 설정을 가져올 수 있습니다.



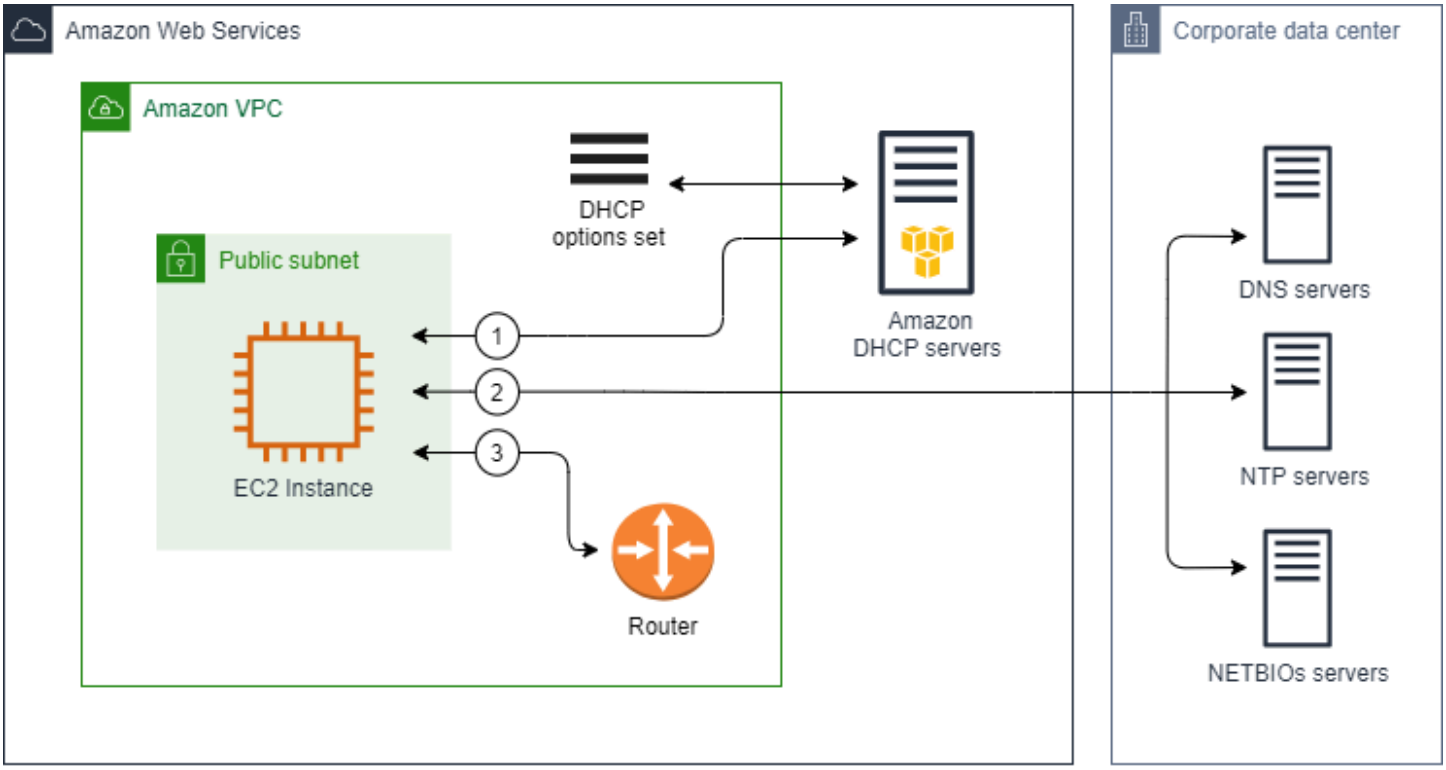
사용자 정의 DHCP 옵션 세트

설정이 다음과 같은 사용자 지정 DHCP 옵션 세트를 생성한 후 VPC와 연결할 수 있습니다.

- 도메인 이름 서버: 네트워크 인터페이스가 도메인 이름 확인에 사용하는 DNS 서버입니다.
- 도메인 이름: 도메인 이름 시스템(DNS)을 사용하여 호스트 이름을 확인하는 경우 클라이언트가 사용하는 도메인 이름입니다.
- NTP 서버: 인스턴스에 시간을 제공하는 NTP 서버입니다.

- NetBIOS 이름 서버: Windows에서 실행되는 EC2 인스턴스의 경우 NetBIOS 컴퓨터 이름은 네트워크에서 인스턴스를 식별하기 위해 인스턴스에 할당된 친숙한 이름입니다. NetBIOS 이름 서버는 NetBIOS를 이름 지정 서비스로 사용하는 네트워크의 NetBIOS 컴퓨터 이름과 네트워크 주소 간의 매핑 목록을 유지 관리합니다.
- NetBIOS 노드 유형: Windows에서 실행되는 EC2 인스턴스의 경우 인스턴스가 NetBIOS 이름을 IP 주소로 확인하는 데 사용하는 방법입니다.
- IPv6 선호 임대 시간(선택 사항): IPv6가 할당되어 실행 중인 인스턴스의 DHCPv6 임대 갱신 빈도의 값입니다(초, 분, 시간 또는 년). 허용되는 값은 140~4,294,967,295초(약 138년)입니다. 값을 입력하지 않는 경우 기본 임대 시간은 140초입니다. EC2 인스턴스에 장기 주소 지정을 사용하면 임대 시간을 늘리고 빈번한 임대 갱신 요청을 방지할 수 있습니다. 임대 갱신은 일반적으로 임대 기간의 절반이 경과했을 때 발생합니다.

사용자 지정 옵션 세트를 사용하면 다이어그램처럼 VPC에서 실행된 인스턴스가 (1) 사용자 지정 DHCP 옵션 세트의 네트워크 설정 사용, (2) 사용자 정의 DHCP 옵션 세트에 지정된 DNS, NTP 및 NetBIOS 서버와 상호 작용, (3) VPC용 라우터를 통해 네트워크의 다른 장치에 연결 작업을 수행합니다.



관련 작업

- DHCP 옵션 세트 생성

- VPC와 연결된 옵션 세트 변경

DHCP 옵션 세트로 작업

다음 절차를 따라 DHCP 옵션 세트를 확인하고 작업하세요. DHCP 옵션 세트의 작동 방식에 대한 자세한 내용은 the section called “DHCP 옵션 세트 개념” 섹션을 참조하세요.

Tasks

- DHCP 옵션 세트 생성
- VPC와 연결된 옵션 세트 변경
- DHCP 옵션 세트 삭제

DHCP 옵션 세트 생성

사용자 지정 DHCP 옵션 세트를 사용하면 자체 DNS 서버, 도메인 이름 등으로 VPC를 사용자 지정할 수 있습니다. 원하는 만큼 DHCP 옵션 세트를 추가로 생성할 수 있습니다. 하지만 한 번에 한 DHCP 옵션 세트와 한 VPC만 연결할 수 있습니다.

> **Note**
>
> DHCP 옵션 세트를 생성한 후에는 이 옵션 세트를 수정할 수 없습니다. VPC의 DHCP 옵션을 업데이트하려면 새 DHCP 옵션 세트를 생성한 다음 이를 VPC와 연결해야 합니다.

콘솔을 사용하여 DHCP 옵션 세트를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 DHCP option sets(DHCP 옵션 세트)를 선택합니다.
3. DHCP 옵션 세트 생성(Create DHCP options set)을 선택합니다.
4. 태그 설정(Tag settings)의 경우 선택적으로 DHCP 옵션 세트의 이름을 입력합니다. 값을 입력하면 DHCP 옵션 세트의 이름 태그가 자동으로 생성됩니다.
5. DHCP 옵션의 경우 필요한 구성 설정을 제공합니다.
   - 도메인 이름(Domain name)(선택 사항): DNS를 사용하여 호스트 이름을 확인하는 경우 클라이언트가 사용해야 하는 도메인 이름을 입력합니다. AmazonProvidedDNS를 사용하지 않는 경우에는 사용자 지정 도메인 이름 서버가 적절하게 호스트 이름을 확인해야 합니다. Amazon Route

53 프라이빗 호스팅 영역을 사용하는 경우 AmazonProvidedDNS를 사용할 수 있습니다. 자세한 내용은 VPC의 DNS 속성 섹션을 참조하세요.

> **Note**
>
> 완전히 제어하는 도메인 이름만 사용하세요.

일부 Linux 운영 체제에서는 공백으로 구분된 여러 도메인 이름을 허용합니다. 하지만 Windows와 기타 Linux 운영 체제에서는 이 값을 단일 도메인으로 취급하므로 예기치 않은 동작이 발생합니다. DHCP 옵션 세트가 값을 단일 도메인으로 취급하는 운영 체제가 실행되는 인스턴스가 있는 VPC와 연결되어 있는 경우 도메인 이름을 하나만 지정합니다.

- 도메인 이름 서버(Domain name servers)(선택 사항): 호스트의 이름에서 호스트의 IP 주소를 확인하는 데 사용할 DNS 서버를 입력합니다.

  `AmazonProvidedDNS` 또는 사용자 지정 도메인 이름 서버 중 하나를 입력할 수 있습니다. 둘 다 사용하면 예상치 못한 동작이 발생할 수 있습니다. 최대 4개의 IPv4 도메인 이름 서버(또는 최대 3개의 IPv4 도메인 이름 서버 및 **`AmazonProvidedDNS`**)와 4개의 IPv6 도메인 이름 서버의 IP 주소를 쉼표로 구분하여 입력할 수 있습니다. 최대 8개의 도메인 이름 서버를 지정할 수 있지만 일부 운영 체제에서는 더 낮은 제한이 적용될 수 있습니다. AmazonProvidedDNS 및 Amazon DNS 서버에 대한 자세한 내용은 Amazon DNS 서버 섹션을 참조하세요.

  > **Important**
  >
  > VPC에 인터넷 게이트웨이가 있는 경우 도메인 이름 서버 값에 자체 DNS 서버 또는 Amazon DNS 서버(AmazonProvidedDNS)를 지정해야 합니다. 그렇지 않으면 VPC의 인스턴스가 DNS에 액세스할 수 없고 이로 인해 인터넷 액세스가 비활성화됩니다.

- NTP 서버(NTP servers)(선택 사항): 최대 8개의 NTP(Network Time Protocol) 서버의 IP 주소(4개의 IPv4 주소와 4개의 IPv6 주소)를 입력합니다.

  NTP 서버는 네트워크에 시간을 제공합니다. IPv4 주소 `169.254.169.123` 또는 IPv6 주소 `fd00:ec2::123`에 Amazon Time Sync Service를 지정할 수 있습니다. 인스턴스는 기본적으로 Amazon Time Sync Service와 통신합니다. IPv6 주소는 Nitro 시스템에 구축된 EC2 인스턴스에만 액세스할 수 있습니다.

  NTP 서버 옵션에 대한 자세한 내용은 RFC 2132를 참조하세요. Amazon Time Sync Service에 대한 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스의 시간 설정을 참조하세요.

- NetBIOS 이름 서버(NetBIOS name servers)(선택 사항): 최대 4개의 NetBIOS 이름 서버의 IP 주소를 입력합니다.

  Windows OS에서 실행되는 EC2 인스턴스의 경우 NetBIOS 컴퓨터 이름은 네트워크에서 인스턴스를 식별하기 위해 인스턴스에 할당된 친숙한 이름입니다. NetBIOS 이름 서버는 NetBIOS를 이름 지정 서비스로 사용하는 네트워크의 NetBIOS 컴퓨터 이름과 네트워크 주소 간의 매핑 목록을 유지 관리합니다.

- NetBIOS 노드 유형(NetBIOS node type)(선택 사항): **1**, **2**, **4** 또는 **8**을 입력합니다. **2**(지점 간 또는 P-노드)를 지정하는 것이 좋습니다. 브로드캐스트 및 멀티캐스트는 현재 지원되지 않습니다. 이러한 노드 유형에 대한 자세한 내용은 RFC 2132의 단원 8.7 및 RFC1001의 단원 10을 참조하십시오.

  Windows OS에서 실행되는 EC2 인스턴스의 경우 인스턴스가 NetBIOS 이름을 IP 주소로 확인하는 데 사용하는 방법입니다. 기본 옵션 세트에는 NetBIOS 노드 유형에 대한 값이 없습니다.

- IPv6 선호 임대 시간(선택 사항): IPv6가 할당되어 실행 중인 인스턴스의 DHCPv6 임대 갱신 빈도의 값입니다(초, 분, 시간 또는 년). 허용되는 값은 140~2,147,483,647초(약 68년)입니다. 값을 입력하지 않는 경우 기본 임대 시간은 140초입니다. EC2 인스턴스에 장기 주소 지정을 사용하면 임대 시간을 늘리고 빈번한 임대 갱신 요청을 방지할 수 있습니다. 임대 갱신은 일반적으로 임대 기간의 절반이 경과했을 때 발생합니다.

6. 태그(Tags)를 추가합니다.
7. DHCP 옵션 세트 생성(Create DHCP options set)을 선택합니다. 새 DHCP 옵션 세트의 이름이나 ID를 기록해 둡니다.
8. 새 옵션 세트를 사용하도록 VPC를 구성하려면 VPC와 연결된 옵션 세트 변경 섹션을 참조하세요.

명령줄을 사용하여 VPC에 대한 DHCP 옵션 세트를 생성하려면

- create-dhcp-options(AWS CLI)
- New-EC2DhcpOption(AWS Tools for Windows PowerShell)

VPC와 연결된 옵션 세트 변경

DHCP 옵션 세트를 만들고 나면 해당 옵션 세트를 하나 이상의 VPC와 연결할 수 있습니다. 한 번에 하나의 DHCP 옵션 세트만 VPC와 연결할 수 있습니다. DHCP 옵션 세트를 VPC에 연결하지 않을 경우 VPC의 도메인 이름 확인이 비활성화됩니다.

새 DHCP 옵션 세트를 VPC와 연결하면 해당 VPC에서 시작하는 모든 새 인스턴스와 기존 인스턴스가 새 옵션을 사용합니다. 인스턴스를 다시 시작하거나 다시 실행할 필요가 없습니다. 인스턴스가 DHCP 임대를 갱신하는 빈도에 따라 몇 시간 안에 변경 내용이 자동으로 파악됩니다. 선호하는 경우 인스턴스에서 운영 체제를 사용하여 임대를 명시적으로 갱신할 수 있습니다.

콘솔을 사용하여 VPC와 연결된 DHCP 옵션 세트를 변경하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC의 확인란을 선택한 다음에 Actions(작업), Edit VPC settings(VPC 설정 편집)를 선택합니다.
4. DHCP 옵션 세트(DHCP options set)에서 새 DHCP 옵션 세트를 선택합니다. 아니면 DHCP 옵션 세트 없음을 선택하여 VPC의 도메인 이름 확인을 비활성화합니다.
5. 저장을 선택합니다.

명령줄을 사용하여 VPC와 연결된 DHCP 옵션 세트를 변경하려면

- associate-dhcp-options(AWS CLI)
- Register-EC2DhcpOption(AWS Tools for Windows PowerShell)

DHCP 옵션 세트 삭제

더 이상 DHCP 옵션 세트가 필요하지 않으면 다음 절차에 따라 DHCP 옵션 세트를 삭제합니다. DHCP 옵션 세트가 사용 중인 경우 해당 옵션 세트는 삭제할 수 없습니다. 삭제하려는 DHCP 옵션 세트와 연결된 각 VPC의 경우, 다른 DHCP 옵션 세트를 VPC에 연결하거나 VPC가 DHCP 옵션 세트를 사용하지 않도록 구성해야 합니다. 자세한 내용은 the section called “VPC와 연결된 옵션 세트 변경” 섹션을 참조하세요.

콘솔을 사용하여 DHCP 옵션 세트를 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 DHCP option sets(DHCP 옵션 세트)를 선택합니다.
3. DHCP 옵션 세트에 대해 라디오 버튼을 선택한 다음 작업, DHCP 옵션 세트 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **delete**를 입력한 다음 DHCP 옵션 세트 삭제를 선택합니다.

명령줄을 사용하여 DHCP 옵션 세트를 삭제하려면

- delete-dhcp-options(AWS CLI)
- Remove-EC2DhcpOption(AWS Tools for Windows PowerShell)

VPC의 DNS 속성

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 내에서 통신할 수 있습니다. 자세한 내용은 VPC 및 서브넷의 IP 주소 지정 단원을 참조하십시오.

Amazon은 VPC용 DNS 서버(Amazon Route 53 Resolver)를 제공합니다. 자체 DNS 서버를 대신 사용하려면 자신의 VPC에 대한 새로운 DHCP 옵션 세트를 생성하세요. 자세한 내용은 Amazon VPC의 DHCP 옵션 세트 섹션을 참조하세요.

내용

- Amazon DNS 이해
- EC2 인스턴스의 DNS 호스트 이름 보기
- VPC에 대한 DNS 속성 보기 및 업데이트

Amazon DNS 이해

AWS 아키텍트 또는 관리자로서 접하게 될 기본 네트워킹 구성 요소 중 하나는 Route 53 Resolver라고도 하는 Amazon DNS 서버입니다. 이 DNS 확인자 서비스는 기본적으로 AWS 리전 내의 각 가용 영역에 통합되어 Virtual Private Cloud(VPC) 내에서 도메인 이름 확인을 위한 안정적이고 확장 가능한 솔루션을 제공합니다. 이 섹션에서는 Amazon DNS 서버의 IP 주소, 해당 서버가 확인할 수 있는 프라이빗 DNS 호스트 이름, 사용을 관리하는 규칙에 대해 알아봅니다.

내용

- Amazon DNS 서버
- 규칙 및 고려 사항
- EC2 인스턴스의 DNS 호스트 이름

- VPC의 DNS 속성
- DNS 할당량
- 프라이빗 호스팅 영역

Amazon DNS 서버

Route 53 Resolver('Amazon DNS 서버' 또는 'AmazonProvidedDNS'라고도 함)는 AWS 리전의 각 가용 영역에 내장된 DNS Resolver 서비스입니다. Route 53 Resolver는 `169.254.169.253`(IPv4), `fd00:ec2::253`(IPv6), VPC에 프로비저닝된 기본 프라이빗 IPV4 CIDR 범위에 2를 더한 범위에 있습니다. 예를 들어 IPv4 CIDR이 `10.0.0.0/16`이고 IPv6 CIDR이 `2001:db8::/32`인 VPC가 있는 경우 `169.254.169.253`(IPv4), `fd00:ec2::253`(IPv6) 또는`10.0.0.2`(IPv4)에서 Route 53 Resolver에 연결할 수 있습니다. VPC 내의 리소스에서는 링크 로컬 주소가 DNS 쿼리에 사용됩니다. 이러한 쿼리는 Route 53 Resolver로 비공개로 전송되며 네트워크에는 표시되지 않습니다. IPv6 전용 서브넷에서는 'AmazonProvidedDNS'가 DHCP 옵션 세트의 이름 서버인 한 IPv4 링크-로컬 주소(169.254.169.253)에 계속 연결할 수 있습니다.

VPC에 인스턴스를 시작하는 경우 인스턴스에 프라이빗 DNS 호스트 이름을 제공합니다. 인스턴스가 퍼블릭 IPv4 주소로 구성되어 있고 VPC DNS 속성이 사용 설정된 경우 퍼블릭 DNS 호스트 이름도 제공합니다.

프라이빗 DNS 호스트 이름의 형식은 EC2 인스턴스를 시작할 때 구성하는 방법에 따라 다릅니다. 프라이빗 DNS 호스트이름에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스 호스트 이름 유형을 참조하세요.

VPC에 있는 Amazon DNS 서버는 Route 53의 프라이빗 호스팅 영역에서 지정하는 DNS 도메인 이름을 확인하는 데 사용합니다. 프라이빗 호스팅 영역에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 프라이빗 호스팅 영역 작업을 참조하세요.

규칙 및 고려 사항

Amazon DNS 서버를 사용할 때 다음 규칙 및 고려 사항이 적용됩니다.

- 네트워크 ACL 또는 보안 그룹을 사용하여 Amazon DNS 서버의 양방향 트래픽을 필터링할 수는 없습니다.
- Amazon EMR 같은 Hadoop 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 고려하세요. 자세한 내용은 Amazon EMR 관리 가이드의 클러스터를 호스팅하도록 VPC를 설정을 참조하세요.

- Amazon Route 53 Resolver는 재귀 DNS 쿼리만 지원합니다.

EC2 인스턴스의 DNS 호스트 이름

인스턴스를 시작하면 인스턴스가 프라이빗 IPv4 주소와 프라이빗 IPv4 주소에 해당하는 프라이빗 DNS 호스트 이름을 항상 수신합니다. 인스턴스에 퍼블릭 IPv4 주소가 있는 경우 VPC의 DNS 속성은 퍼블릭 IPv4 주소에 해당하는 퍼블릭 DNS 호스트 이름을 수신할 여부를 결정합니다. 자세한 내용은 VPC의 DNS 속성 섹션을 참조하세요.

Amazon에서 제공한 DNS 서버를 활성화하면 DNS 호스트 이름이 다음과 같이 확인됩니다.

프라이빗 IPv4 DNS 이름

인스턴스의 프라이빗 IPv4 DNS 호스트 이름은 인스턴스의 프라이빗 IPv4 주소로 확인합니다. 연결의 프라이빗 IPv4 DNS 호스트 이름을 사용하여 같은 VPC 또는 연결된 VPC의 인스턴스 간에 통신할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서에서 프라이빗 IPv4 주소를 참조하세요.

퍼블릭 IPv4 DNS 이름

인스턴스의 퍼블릭 IPv4 DNS 호스트 이름은 인스턴스의 퍼블릭 IPv4 주소로 확인되거나(인스턴스의 네트워크 외부에서) 인스턴스의 프라이빗 IPv4 주소(인스턴스의 네트워크 내부에서)로 확인됩니다. 자세한 내용은 Amazon EC2 사용 설명서에서 퍼블릭 IPv4 주소를 참조하세요.

VPC 피어링 연결을 통해 퍼블릭 IPv4 DNS 이름을 프라이빗 IPv4 주소로 확인하려면 피어링 연결에 대해 DNS 확인을 활성화해야 합니다. 자세한 정보는 VPC 피어링 연결에 대해 DNS 확인 사용 설정을 참조하세요.

프라이빗 리소스 DNS 이름

이 인스턴스에 대해 선택한 A 및 AAAA DNS 레코드로 확인할 수 있는 RBN 기반 DNS 이름입니다. 이 DNS 호스트 이름은 이중 스택 및 IPv6 전용 서브넷의 인스턴스에 대한 인스턴스 세부 정보에서 볼 수 있습니다. RBN에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 EC2 인스턴스 호스트 이름 유형을 참조하세요.

VPC의 DNS 속성

다음 VPC 속성에 따라 VPC에 제공되는 DNS 지원이 결정됩니다. 두 속성을 모두 사용하면, VPC로 시작된 인스턴스는 생성 시 퍼블릭 IPv4 주소 또는 탄력적 IP 주소가 할당된 경우 퍼블릭 DNS 호스트 이

름을 받습니다. 이전에는 모두 사용하지 않았던 두 VPC 속성을 모두 사용하도록 설정하면, 그 VPC로 이미 시작된 인스턴스는 퍼블릭 IPv4 주소 또는 탄력적 IP 주소가 있는 경우 퍼블릭 DNS 호스트 이름을 받습니다.

VPC에 대해 이러한 속성이 활성화되어 있는지 확인하려면 VPC에 대한 DNS 속성 보기 및 업데이트을 참조하십시오.

속성	설명
`enableDnsHostnames`	VPC가 퍼블릭 IP 주소가 있는 인스턴스에 퍼블릭 DNS 호스트 이름을 할당하도록 지원할 여부를 결정합니다. VPC가 기본 VPC가 아닌 경우 이 속성의 기본값은 `false`입니다. 아래 이 속성에 대한 규칙 및 고려 사항을 참고하세요.
`enableDnsSupport`	VPC가 Amazon에서 제공하는 DNS 서버를 통해 DNS 확인을 지원하는지 여부를 결정합니다. 이 속성이 `true`인 경우, Amazon 제공 DNS 서버에 대한 쿼리가 성공합니다. 자세한 내용은 Amazon DNS 서버 섹션을 참조하세요. 이 속성의 기본값은 `true`입니다. 아래 이 속성에 대한 규칙 및 고려 사항을 참고하십시오.

규칙 및 고려 사항

- 두 속성이 모두 `true`로 설정되는 경우 다음이 발생합니다.
  - 퍼블릭 IP 주소를 갖는 인스턴스가 해당하는 퍼블릭 DNS 호스트 이름을 받습니다.
  - Route 53 Resolver 서버는 Amazon에서 제공하는 프라이빗 DNS 호스트 이름을 확인할 수 있습니다.
- 속성 중 하나 이상이 `false`로 설정된 경우 다음이 발생합니다.
  - 퍼블릭 IP 주소를 갖는 인스턴스가 해당하는 퍼블릭 DNS 호스트 이름을 받지 않습니다.
  - Route 53 Resolver는 Amazon에서 제공하는 프라이빗 DNS 호스트 이름을 확인할 수 없습니다.
  - 사용자 지정 도메인 이름에 DHCP 옵션 세트가 있을 경우 인스턴스가 프라이빗 DNS 호스트 이름을 받습니다. Route 53 Resolver 서버를 사용하지 않는 경우에는 사용자 지정 도메인 이름 서버가 적절한 방법으로 호스트 이름을 확인해야 합니다.

- Amazon Route 53의 프라이빗 호스팅 영역에서 정의된 사용자 지정 DNS 도메인 이름을 사용하거나 프라이빗 DNS를 인터페이스 VPC 엔드포인트(AWS PrivateLink)와 함께 사용하는 경우, `enableDnsHostnames` 및 `enableDnsSupport` 속성을 `true`로 설정해야 합니다.
- Route 53 Resolver는 프라이빗 DNS 호스트 이름을 프라이빗 IPv4 주소로 확인할 수 있습니다. VPC의 IPv4 주소 범위가 RFC 1918에서 지정하는 프라이빗 IPv4 주소 범위를 벗어나는 경우도 마찬가지입니다. 그러나 2016년 10월 이전에 VPC를 생성했다면 VPC의 IPv4 주소 범위가 이 범위를 벗어나는 경우 Route 53 Resolver는 프라이빗 DNS 호스트 이름을 확인하지 않습니다. 이에 대한 지원을 활성화하려면 지원에 문의하십시오.

DNS 할당량

링크-로컬 주소를 사용하는 서비스에는 초당 1024 패킷(PPS) 제한이 있습니다. 이 제한에는 Route 53 Resolver DNS 쿼리, 인스턴스 메타데이터 서비스(IMDS) 요청, Amazon Time Service Network Time Protocol(NTP) 요청, Windows 라이선스 서비스(Microsoft Windows 기반 인스턴스용) 요청의 집계가 포함됩니다. 이 할당량은 늘릴 수 없습니다.

Route 53 Resolver가 지원하는 초당 DNS 쿼리 수는 쿼리 유형, 응답 크기 및 사용 중인 프로토콜에 따라 다릅니다. 확장 가능한 DNS 아키텍처에 대한 자세한 내용과 권장 사항은 Active Directory 포함 AWS 하이브리드 DNS 기술 가이드를 확인하세요.

할당량에 도달하면 Route 53 Resolver가 트래픽을 거부합니다. 할당량에 도달하는 원인으로는 DNS 조절 문제나, Route 53 Resolver 네트워크 인터페이스를 사용하는 인스턴스 메타데이터 쿼리 등이 있습니다. VPC DNS 조절 문제를 해결하는 방법에 대한 자세한 내용은 Amazon에서 제공한 DNS 서버에 대한 내 DNS 쿼리가 VPC DNS 조절로 인해 실패하는지 확인하려면 어떻게 해야 하나요?를 참조하세요. 인스턴스 메타데이터 검색에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 인스턴스 메타데이터 검색을 참조하세요.

프라이빗 호스팅 영역

프라이빗 IPv4 주소 또는 AWS 제공 프라이빗 DNS 호스트 이름을 사용하는 대신 `example.com`과 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 53에서 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 안내서의 프라이빗 호스팅 영역 작업을 참조하세요.

사용자 지정 DNS 도메인 이름을 사용하여 리소스에 액세스하려면 VPC 내에 있는 인스턴스에 연결되어 있어야 합니다. 인스턴스에서 `ping` 명령(예: `ping mywebserver.example.com`)을 사용하여 사용자 지정 DNS 이름에서 프라이빗 호스팅 영역에 있는 리소스에 액세스 가능한지 테스트할 수 있습니다. (인스턴스의 보안 그룹 규칙에서 `ping`에 대한 인바운드 ICMP 트래픽 작동을 허용하는지 확인해야 합니다.)

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 둘 다 `true`(으)로 설정해야 합니다.

EC2 인스턴스의 DNS 호스트 이름 보기

Amazon EC2 콘솔 또는 명령줄을 사용하면 실행 중인 인스턴스 또는 네트워크 인터페이스의 DNS 호스트 이름을 볼 수 있습니다. 리소스에 연결하려면 이러한 호스트 이름을 아는 것이 중요합니다.

퍼블릭 DNS(IPv4) 및 프라이빗 DNS 필드는 인스턴스와 연결된 VPC에 대해 DNS 옵션을 활성화한 경우 사용할 수 있습니다. 자세한 내용은 the section called “VPC의 DNS 속성” 섹션을 참조하세요.

Instance

콘솔을 사용하여 인스턴스의 DNS 호스트 이름을 보려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스(Instances)를 선택합니다.
3. 목록에서 해당 인스턴스를 선택합니다.
4. 해당되는 경우, 세부 정보 창의 퍼블릭 DNS(Public DNS)(IPv4) 및 프라이빗 DNS(Private DNS) 필드에 DNS 호스트 이름이 표시됩니다.

명령줄을 사용하여 인스턴스의 DNS 호스트 이름을 보려면

- describe-instances (AWS CLI)

- Get-EC2InstanceAWS Tools for Windows PowerShell

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 네트워크 인터페이스(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

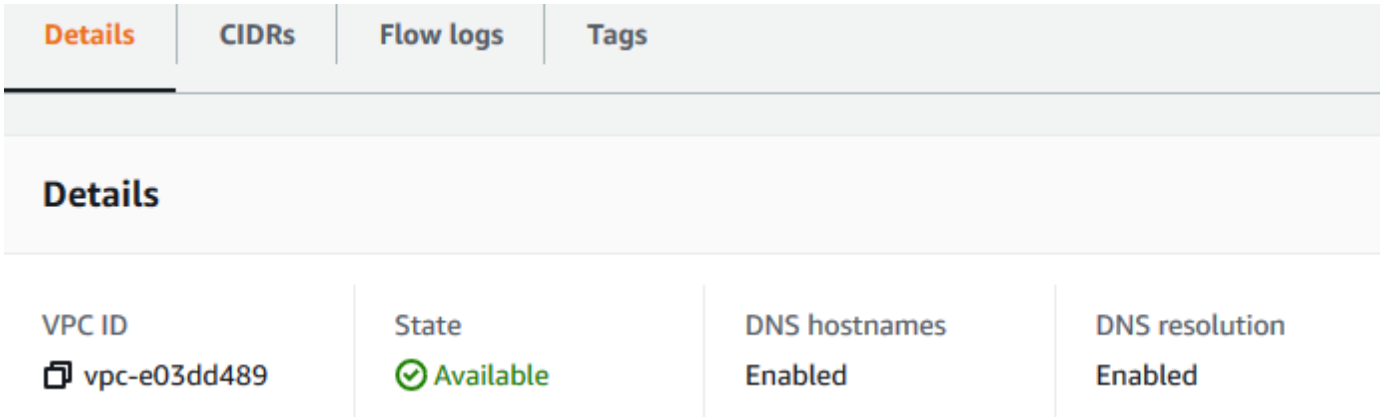
- describe-network-interfaces(AWS CLI)
- Get-EC2NetworkInterface(AWS Tools for Windows PowerShell)

VPC에 대한 DNS 속성 보기 및 업데이트

Amazon VPC 콘솔을 이용해 VPC의 DNS 지원 속성을 확인하고 업데이트할 수 있습니다. 이러한 설정은 인스턴스가 퍼블릭 DNS 호스트 이름을 받는지 여부와 Amazon DNS 서버가 프라이빗 DNS 이름을 확인할 수 있는지 여부를 제어합니다. 이러한 속성을 올바르게 구성하는 것은 VPC 내에서 원활한 통신을 보장하는 데 중요합니다.

콘솔을 사용하여 VPC에 대한 DNS 지원을 설명하고 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC에 대한 확인란을 선택합니다.
4. 세부 정보(Details)의 정보를 검토합니다. 이 예에서는 DNS 호스트 이름(DNS hostnames) 및 DNS 확인(DNS resolution)이 모두 활성화되었습니다.



5. 이러한 설정을 업데이트하려면 Actions(작업)와 Edit VPC settings(VPC 설정 편집)를 차례로 선택합니다. 적절한 DNS 속성에 대한 Enable(활성화)을 선택하거나 선택 취소하고 Save changes(변경 사항 저장)를 선택합니다.

명령줄을 사용하여 VPC에 대한 DNS 지원을 설명하려면

- describe-vpc-attribute(AWS CLI)
- Get-EC2VpcAttribute(AWS Tools for Windows PowerShell)

명령줄을 사용하여 VPC에 대한 DNS 지원을 업데이트하려면

- modify-vpc-attribute(AWS CLI)
- Edit-EC2VpcAttribute(AWS Tools for Windows PowerShell)

VPC의 네트워크 주소 사용량

NAU(네트워크 주소 사용량)는 VPC 크기를 계획하고 모니터링하는 데 도움이 되도록 가상 네트워크의 리소스에 적용되는 지표입니다. 각 NAU 단위가 VPC 크기를 나타내는 합계에 포함됩니다.

다음과 같은 VPC 할당량을 통해 VPC 크기가 제한되므로 VPC의 NAU를 구성하는 총 단위 수를 파악하는 것이 중요합니다.

- 네트워크 주소 사용량 – 단일 VPC에서 보유할 수 있는 최대 NAU 단위 수 합계입니다. 각 VPC에서는 기본적으로 최대 64,000개의 NAU 단위 수 합계를 보유할 수 있습니다. 256,000개까지 할당량 증가를 요청할 수 있습니다.
- 피어링된 네트워크 주소 사용량 – VPC 및 피어링된 모든 VPC의 최대 NAU 단위 수 합계입니다. VPC가 동일한 리전의 다른 VPC와 피어링된 경우 결합된 VPC에서는 기본적으로 128,000개까지

NAU 단위 수 합계를 보유할 수 있습니다. 512,000까지 할당량 증가를 요청할 수 있습니다. 이 할당량은 동일한 리전 내 VPC 간 피어링에 적용되며, 여기에는 서로 다른 AWS 계정의 VPC 간 피어링도 포함됩니다. 상이한 리전에서 피어링되는 VPC는 이 할당량에 포함되지 않습니다.

다음과 같은 방법으로 NAU를 사용할 수 있습니다.

- 가상 네트워크를 생성하기 전에 여러 VPC에 워크로드를 분산해야 하는지 결정하는 데 도움이 되도록 NAU 단위 수 합계를 계산합니다.
- VPC를 생성한 후에는 NAU 할당량을 초과하지 않도록 Amazon CloudWatch를 사용하여 VPC의 NAU 사용량을 모니터링하세요. NAU 할당량을 모니터링하려면 Amazon CloudWatch 경보를 생성하는 것이 좋습니다. 자세한 내용은 the section called “CloudWatch 지표” 섹션을 참조하세요.

NAU 할당량 초과

VPC의 NAU 할당량을 초과하면 클라이언트 측 예외와 함께 다음 API 직접 호출이 실패합니다.

- `RunInstances`
- `AttachNetworkInterface`
- `AssignPrivateIpAddresses`
- `AssignIpv6Addresses`
- `AcceptVpcPeeringConnection`

구체적인 예외 사항은 초과한 할당량에 따라 달라집니다.

- `NetworkAddressUsageLimitExceeded` - VPC가 NAU 할당량을 초과합니다.
- `NetworkAddressUsagePeeredLimitExceeded` - VPC가 피어링된 VPC NAU 할당량을 초과합니다.

이러한 오류는 영향을 받는 VPC에서 인스턴스 시작, 네트워크 인터페이스 연결, 새 주소 할당, VPC 피어링 연결 수락, 워크로드 확장 또는 관리 기능에 영향을 줍니다. 중단을 방지하려면 Amazon CloudWatch로 NAU 사용량을 모니터링하고 할당량에 도달하기 전에 증량을 요청하세요.

NAU를 계산하는 방법

NAU를 계산하는 방법을 이해하면 VPC 확장을 계획하는 데 도움이 될 수 있습니다.

다음 표에 VPC의 NAU 개수를 구성하는 리소스와 각 리소스에서 사용하는 NAU 단위 수 합계 수가 설명되어 있습니다. 일부 AWS 리소스는 단일 NAU 단위 수 합계로 표시되고 일부 리소스는 여러 NAU 단위 수 합계로 표시됩니다. 이 표를 사용하여 NAU를 계산하는 방식을 알아볼 수 있습니다.

리소스	NAU 단위 수 합계
VPC의 EC2 인스턴스에 대한 네트워크 인터페이스에 할당된 각 프라이빗 또는 퍼블릭 IPv4 및 각 IPv6 주소	1
EC2 인스턴스에 연결된 추가 네트워크 인터페이스	1
네트워크 인터페이스에 할당된 접두사	1
AZ당 Network Load Balancer	6
AZ당 Gateway Load Balancer	6
AZ당 VPC 엔드포엔드포인트	6
Transit Gateway Attachment	6
Lambda 함수	6
NAT 게이트웨이	6
EFS 탑재 대상	6
EFA 인터페이스(ENA 디바이스가 있는 EFA) 또는 EFA 전용 인터페이스	1
Amazon EKS 포드	1

NAU 예시

다음 예시에서는 NAU를 계산하는 방법을 보여줍니다.

예시 1 - VPC 피어링을 사용하여 연결된 VPC 2개

결합된 NAU 할당량에 동일한 리전의 피어링된 VPC가 포함됩니다.

- VPC 1
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개

  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- VPC 2
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개
  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- 총 피어링 NAU 개수: 단위 42,400개
- 기본 피어링 NAU 할당량: 단위 128,000개

예시 2 - Transit Gateway를 사용하여 연결된 VPC 2개

Transit Gateway를 사용하여 연결된 VPC는 피어링된 VPC와는 달리, 결합된 NAU 할당량에 포함되지 않습니다.

- VPC 1
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개
  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- VPC 2
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개
  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- VPC당 총 NAU 개수: 단위 21,200개
- VPC당 기본 NAU 할당량: 단위 64,000개

예시 3 - 교차 리전 VPC 피어링을 사용하여 연결된 VPC 2개

서로 다른 리전에 걸쳐 피어링된 VPC는 통합 NAU 할당량에 포함되지 않습니다. AWS는 각 VPC를 VPC당 NAU 할당량과 비교하여 평가하며, 교차 리전 피어의 리소스는 VPC의 피어링된 NAU에 반영되

지 않습니다. 이 예제에서는 각 VPC의 유일한 피어링이 표시된 교차 리전 피어링이라고 가정합니다(추가 리전 내 피어 없음).

- VPC 1(리전 A)
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개
  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- VPC 2(리전 B)
  - 별도의 가용 영역에 있는 서브넷 2개의 Network Load Balancer 50개 - NAU 단위 수 합계 600개
  - 하나의 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) 및 다른 서브넷에 5,000개의 인스턴스(각각 IPv4 주소 및 IPv6 주소 포함) - 20,000개
  - Lambda 함수 100개 - NAU 단위 수 합계 600개
- VPC당 총 NAU 개수: 단위 21,200개
- VPC당 기본 NAU 할당량: 단위 64,000개
- VPC당 피어링된 NAU 수: 21,200단위(VPC 자체 리소스만 포함되며, 교차 리전 피어는 0으로 계산됨)
- VPC당 기본 피어링된 NAU 할당량: 128,000단위(교차 리전 피어는 이 할당량에 포함되지 않음)

다른 계정과 VPC 서브넷 공유

VPC 서브넷 공유를 이용하면 여러 AWS 계정에서 Amazon EC2 인스턴스, Amazon Relational Database Service(RDS) 데이터베이스, Amazon Redshift 클러스터, AWS Lambda 함수 등의 애플리케이션 리소스를 중앙 관리형 공유 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.

높은 상호 연결성이 요구되고 동일한 신뢰 경계 내에 있는 애플리케이션의 경우, VPC 서브넷을 공유하여 VPC 내부의 암시적 라우팅 기능을 활용할 수 있습니다. 이렇게 하면 생성 및 관리하는 VPC 수가 줄어들고 청구 및 액세스 제어에 별도의 계정을 사용할 수 있습니다. AWS PrivateLink, 전송 게이트웨이, VPC 피어링과 같은 연결 특성으로 공유 Amazon VPC 서브넷을 상호 연결하여 네트워크 토폴로지를 간소화할 수 있습니다. VPC 서브넷 공유의 이점에 대한 자세한 내용은 VPC 공유: 여러 계정 및 VPC 관리에 대한 새로운 접근 방식을 참조하세요.

VPC 서브넷 공유와 관련된 할당량이 있습니다. 자세한 내용은 VPC 서브넷 공유 섹션을 참조하세요.

내용

- 공유 서브넷 사전 조건
- 공유 서브넷 작업
- 소유자와 참여자에 대한 청구 및 측정
- 소유자 및 참가자에 대한 책임 및 권한
- AWS 리소스 및 공유 VPC 서브넷

공유 서브넷 사전 조건

이 섹션에는 공유 서브넷 작업을 위한 사전 조건이 포함되어 있습니다.

- VPC 소유자 및 참여자의 계정은 AWS Organizations에서 관리해야 합니다.
- AWS RAM 콘솔의 조직 관리 계정에서 리소스 공유를 활성화해야 합니다. 자세한 내용은 AWS RAM 사용 설명서의 AWS Organizations 내 리소스 공유 활성화를 참조하세요.
- 리소스 공유를 생성해야 합니다. 리소스 공유를 생성할 때 공유할 서브넷을 지정하거나, 나중에 다음 섹션의 절차를 사용하여 리소스 공유에 서브넷을 추가할 수 있습니다. 자세한 내용은 AWS RAM 사용 설명서의 리소스 공유 생성을 참조하세요.

공유 서브넷 작업

이 섹션에서는 AWS 콘솔과 AWS CLI에서 공유 서브넷으로 작업하는 방법을 설명합니다.

내용

- 서브넷 공유
- 공유 서브넷 공유 해제
- 공유 서브넷의 소유자 식별

서브넷 공유

기본이 아닌 서브넷을 조직의 다른 계정과 다음과 같이 공유할 수 있습니다. 또한 AWS Organizations 간에서 보안 그룹을 공유할 수 있습니다. 자세한 내용은 AWS Organizations와 보안 그룹 공유 섹션을 참조하세요.

콘솔을 사용하여 서브넷을 공유하려면

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

AWS CLI를 사용하여 서브넷을 공유하려면

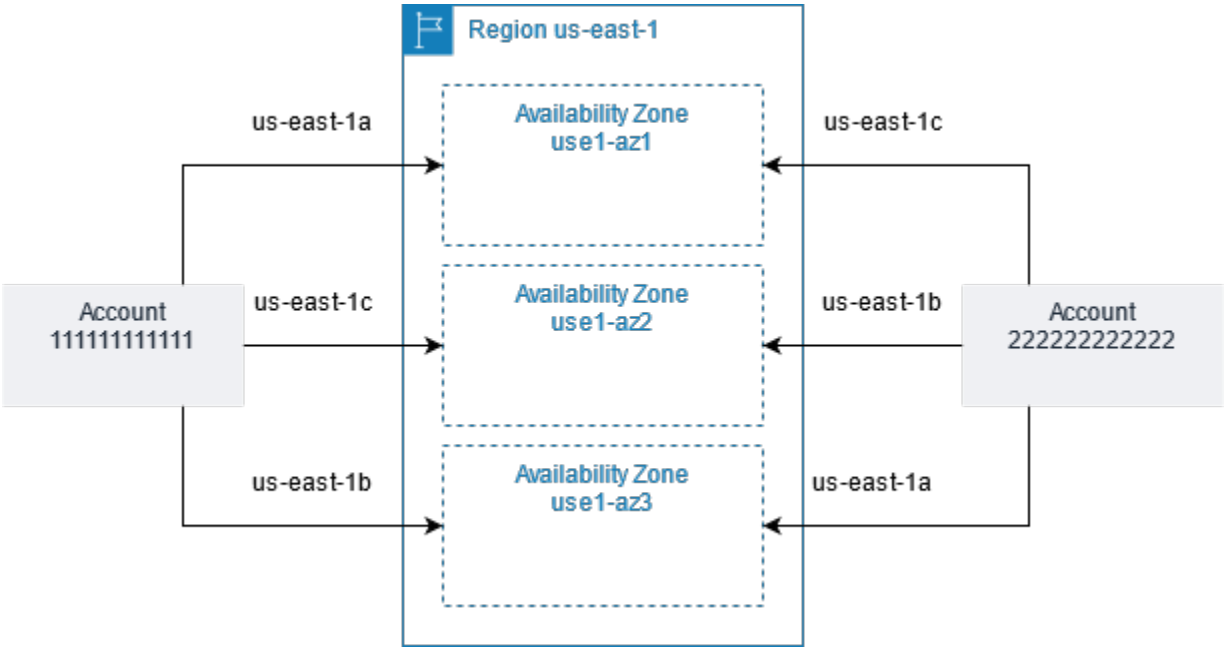
create-resource-share 명령과 associate-resource-share 명령을 사용합니다.

여러 가용 영역에서 서브넷 매핑

리전의 가용 영역에 걸쳐 리소스가 배포될 수 있도록 각 계정의 이름에 가용 영역을 독립적으로 매핑합니다. 예를 들어 AWS 계정의 `us-east-1a` 가용 영역은 다른 AWS 계정에 대한 `us-east-1a`로 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 하나에 대한 AZ ID입니다. AZ ID를 사용하여 다른 계정과 관련된 하나의 계정의 리소스 위치를 판단합니다. Amazon VPC 콘솔에서 각 서브넷에 대한 AZ ID를 확인할 수 있습니다.

다음 다이어그램은 가용 영역 코드를 AZ ID에 매핑하는 서로 다른 두 개의 계정을 보여 줍니다.



공유 서브넷 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. 자동/관리형 워크플로(예: Auto Scaling 또는 노드 교체)가 있는 AWS 관리형 서비스 (예: Elastic Load Balancing의 경우 일부 리소스의 공유 서브넷에 지속적으로 액세스해야 할 수 있습니다.
- 참여자는 공유 해제된 서브넷에 더 이상 새로운 리소스를 생성할 수 없습니다.
- 참여자는 서브넷에 있는 리소스를 수정, 기술하고 삭제할 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 Subnets를 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

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

콘솔을 사용하여 서브넷 소유자를 식별하려면

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

AWS CLI를 사용하여 서브넷 소유자를 식별하려면

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

소유자와 참여자에 대한 청구 및 측정

이 섹션에는 공유 서브넷을 소유한 사용자와 공유 서브넷으로 작업하는 사용자를 위한 청구 및 측정 세부 정보가 들어 있습니다.

- 공유 VPC에서 각 참여자는 Amazon EC2 인스턴스, Amazon Relational Database Service 데이터베이스, Amazon Redshift 클러스터 및 AWS Lambda 함수를 비롯한 애플리케이션 리소스에 대한 비용을 지불합니다. 또한 참가자는 가용 영역 간 데이터 전송은 물론 VPC 피어링 연결을 통한 데이터 전송, 인터넷 게이트웨이 간 데이터 전송 및 AWS Direct Connect 게이트웨이 간 데이터 전송과 연결된 데이터 전송 요금을 지불합니다.
- VPC 소유자는 NAT 게이트웨이, 가상 프라이빗 게이트웨이, 전송 게이트웨이, AWS PrivateLink 및 VPC 엔드포인트에서의 데이터 처리 및 데이터 전송 요금을 시간당 요금(해당하는 경우)으로 지불합니다. 아울러 공유 VPC에서 사용되는 퍼블릭 IPv4 주소는 VPC 소유자에게 요금이 청구됩니다. 퍼블릭 IPv4 주소 요금에 대한 자세한 내용은 Amazon VPC 요금 페이지의 퍼블릭 IPv4 주소 탭을 참조하세요.
- 동일한 가용 영역(AZ-ID를 사용하여 고유하게 식별됨) 내에서의 데이터 전송은 통신 리소스의 계정 소유권과 상관없이 무료입니다.

소유자 및 참가자에 대한 책임 및 권한

이 섹션에는 공유 서브넷을 소유한 사용자(소유자)와 공유 서브넷을 사용하는 사용자(참가자)의 책임과 권한에 대한 세부 정보가 들어 있습니다.

소유자 리소스

소유자는 자신이 소유한 VPC 리소스에 대한 책임이 있습니다. VPC 소유자는 공유 VPC와 연결된 리소스를 생성, 관리 및 삭제할 책임이 있습니다. 이러한 책임에는 서브넷, 라우팅 테이블, 네트워크 ACL, 피어링 연결, 게이트웨이 엔드포인트, 인터페이스 엔드포인트, Route 53 Resolver 엔드포인트, 인터넷 게이트웨이, NAT 게이트웨이, 가상 프라이빗 게이트웨이 및 전송 게이트웨이 연결이 포함됩니다.

참여자 리소스

참여자는 자신이 소유한 VPC 리소스에 대한 책임이 있습니다. 참여자는 공유된 VPC에서 제한된 VPC 리소스 세트를 생성할 수 있습니다. 예를 들어 참여자는 네트워크 인터페이스 및 보안 그룹을 생성하고 자신이 소유한 네트워크 인터페이스에 대한 흐름 로그를 활성화할 수 있습니다. 참여자가 생성하는 VPC 리소스는 소유자 계정이 아닌 참여자 계정의 VPC 할당량에 포함됩니다. 자세한 내용은 VPC 서브넷 공유 섹션을 참조하세요.

VPC 리소스

공유 VPC 서브넷으로 작업할 때 VPC 리소스에는 다음과 같은 책임 및 권한이 적용됩니다.

흐름 로그

- 참가자는 공유 VPC 서브넷에서 자신이 소유하는 네트워크 인터페이스에 대한 흐름 로그를 생성하고, 삭제하고, 설명할 수 있습니다.
- 참가자는 공유 VPC 서브넷에서 자신이 소유하지 않는 네트워크 인터페이스에 대한 흐름 로그를 생성하고, 삭제하고, 설명할 수 없습니다.
- 참가자는 공유 VPC 서브넷에 대한 흐름 로그를 생성하고, 삭제하고, 설명할 수 없습니다.
- VPC 소유자는 공유 VPC 서브넷에서 자신이 소유하지 않는 네트워크 인터페이스에 대한 흐름 로그를 생성하고, 삭제하고, 설명할 수 있습니다.
- VPC 소유자는 공유 VPC 서브넷에 대한 흐름 로그를 생성하고, 삭제하고, 설명할 수 있습니다.
- VPC 소유자는 참가자가 생성한 흐름 로그를 설명하거나 삭제할 수 없습니다.

인터넷 게이트웨이 및 외부 전용 인터넷 게이트웨이

- 참가자는 공유 VPC 서브넷에서 인터넷 게이트웨이 및 외부 전용 인터넷 게이트웨이를 생성, 연결 또는 삭제할 수 없습니다. 참가자는 공유 VPC 서브넷의 인터넷 게이트웨이를 설명할 수 있습니다. 참가자는 공유 VPC 서브넷의 송신 전용 인터넷 게이트웨이를 설명할 수 없습니다.

NAT 게이트웨이

- 참가자는 공유 VPC 서브넷에서 NAT 게이트웨이를 생성, 삭제 또는 설명할 수 없습니다.

네트워크 액세스 제어 목록(NACL)

- 참가자는 공유 VPC 서브넷에서 NACL을 생성, 삭제 또는 교체할 수 없습니다. 참가자는 공유 VPC 서브넷에서 VPC 소유자가 생성한 NACL을 설명할 수 있습니다.

네트워크 인터페이스

- 참가자는 공유 VPC 서브넷에서 네트워크 인터페이스를 만들 수 있습니다. 참가자는 공유 VPC 서브넷에서 VPC 소유자가 생성한 네트워크 인터페이스에 대해 다른 방식(예: 네트워크 인터페이스 연결, 연결 해제 또는 수정)으로 작업할 수 없습니다. 참가자는 공유 VPC에서 자신이 생성한 네트워크 인

터페이스를 수정 또는 삭제할 수 있습니다. 예를 들어 참가자는 자신이 생성한 네트워크 인터페이스에 IP 주소를 연결하거나 연결 해제할 수 있습니다.

- VPC 소유자는 공유 VPC 서브넷의 참가자가 소유한 네트워크 인터페이스를 설명할 수 있습니다. VPC 소유자는 참가자가 소유한 네트워크 인터페이스를 다른 방식(예: 공유 VPC 서브넷의 참가자가 소유한 네트워크 인터페이스의 연결, 연결 해제 또는 수정)으로 작업할 수 없습니다.

라우팅 테이블

- 참가자는 공유 VPC 서브넷에서 라우팅 테이블에 대한 작업(예: 라우팅 테이블 생성, 삭제 또는 연결)을 수행할 수 없습니다. 참가자는 공유 VPC 서브넷의 라우팅 테이블을 설명할 수 있습니다.

보안 그룹

- 참가자는 공유 VPC 서브넷에서 자신이 소유하는 보안 그룹을 작업할 수 있습니다(수신 및 송신 규칙 생성, 삭제, 설명 또는 수정). VPC 소유자가 참가자와 보안 그룹을 공유하는 경우 참가자는 VPC 소유자가 생성한 보안 그룹으로 작업할 수 있습니다.
- 참가자는 자신이 소유한 보안 그룹에 다른 참가자나 VPC 소유자에게 속하는 보안 그룹을 참조하는 규칙을 만들 수 있습니다(예: account-number/security-group-id)
- 참가자는 VPC의 기본 보안 그룹을 사용하여 인스턴스를 시작할 수 없습니다. 이는 소유자에게 속해 있기 때문입니다.
- 참가자는 보안 그룹이 공유된 경우가 아니라면 VPC 소유자나 다른 참가자나 소유한 기본이 아닌 보안 그룹을 사용하여 인스턴스를 시작할 수 없습니다.
- 참가자는 공유 VPC 서브넷에서 참가자가 생성한 보안 그룹을 설명할 수 있습니다. VPC 소유자는 참가자가 생성한 보안 그룹을 다른 방식으로 작업할 수 없습니다. 예를 들어, VPC 소유자는 참가자가 생성한 보안 그룹을 사용하여 인스턴스를 시작할 수 없습니다.

서브넷

- 참가자는 공유 서브넷 또는 관련 속성을 수정할 수 없습니다. VPC 소유자만 수정할 수 있습니다. 참가자는 공유 VPC 서브넷에서 서브넷을 설명할 수 있습니다.
- VPC 소유자는 AWS Organizations에서 동일한 조직에 속한 다른 계정 또는 다른 조직 단위와만 서브넷을 공유할 수 있습니다. VPC 소유자는 기본 VPC에 있는 서브넷을 공유할 수 없습니다.

전송 게이트웨이

- 서브넷 소유자만 공유 VPC 서브넷에 전송 게이트웨이를 연결할 수 있습니다. 참여자는 선택할 수 없습니다.

VPC

- 참가자는 VPC 또는 그 해당 속성을 수정할 수 없습니다. VPC 소유자만 수정할 수 있습니다. 참가자는 VPC, 해당 속성 및 DHCP 옵션 세트를 설명할 수 있습니다.
- VPC 태그와 공유 VPC 내 리소스에 대한 태그는 참여자와 공유되지 않습니다.
- 참가자는 자신의 보안 그룹을 공유 VPC와 연결할 수 있습니다. 이렇게 하면 참가자가 공유 VPC에 소유한 탄력적 네트워크 인터페이스에서 보안 그룹을 사용할 수 있습니다.

AWS 리소스 및 공유 VPC 서브넷

다음 AWS 서비스는 공유 VPC 서브넷의 리소스를 지원합니다. 자세한 내용은 해당 서비스 설명서의 링크를 참조하세요.

- Amazon Aurora
- AWS Database Migration Service
- (Amazon EC2)
- Amazon ECS:
- Amazon ElastiCache(Redis OSS)
- Amazon EFS
- Amazon Elastic Kubernetes Service
- Elastic Load Balancing
  - Application Load Balancers.
  - Gateway Load Balancers
  - Network Load Balancers
- Amazon EMR
- AWS Glue
- AWS Lambda
- Apache MQ(Rabbit MQ 아님)를 실행하는 Amazon MQ

- Amazon MSK
- AWS Network Manager
  - AWS 클라우드 WAN
  - Network Access Analyzer
  - Reachability Analyzer
- Amazon OpenSearch Service
- AWS PrivateLink†
- Amazon Relational Database Service(RDS)
- Amazon Redshift
- Amazon Route 53
- Amazon SageMaker Unified Studio
- AWS Transit Gateway
- AWS Verified Access
- Amazon VPC
  - 피어링
  - 트래픽 미러링
- Amazon VPC Lattice

† 공유 VPC에서 VPC 엔드포인트를 사용하여 PrivateLink를 지원하는 모든 AWS 서비스에 연결할 수 있습니다. PrivateLink를 지원하는 서비스 목록은 AWS PrivateLink 가이드에서 AWS PrivateLink와 통합되는 AWS 서비스를 참조하세요.

이 목록에는 공유 VPC 서브넷에서의 리소스 시작을 지원하는 모든 서비스를 포함하려고 합니다. 최선의 노력에도 불구하고 공유 VPC 서브넷에서의 리소스 시작을 지원하지만 목록에 없는 서비스가 있을 수 있습니다. 질문이 있는 경우 설명서 피드백을 제출하시기 바랍니다.

로컬 영역, Wavelength 영역 또는 Outpost로 VPC 확장

서브넷과 같은 VPC 리소스를 전 세계 여러 위치에서 호스팅할 수 있습니다. 이 위치는 리전, 가용 영역, Local Zones 및 Wavelength Zone으로 구성됩니다. 각 리전은 개별 지리 영역입니다.

- 가용 영역은 각 리전 내에 있는 여러 격리된 위치입니다.

- Local Zones에서는 최종 사용자에게 가까운 여러 위치에 컴퓨팅, 스토리지 등의 리소스를 배치할 수 있습니다.
- AWS Outposts는 네이티브 AWS 서비스, 인프라 및 운영 모델을 사실상 모든 데이터 센터, 코로케이션 공간 또는 온프레미스 시설로 옮길 수 있습니다.
- Wavelength Zone을 사용하면 개발자는 5G 디바이스 및 최종 사용자에게 매우 짧은 지연 시간을 제공하는 애플리케이션을 빌드할 수 있습니다. Wavelength는 표준 AWS 컴퓨팅 및 스토리지 서비스를 통신 사업자의 5G 네트워크 엣지에 배포합니다.

AWS은 최신 기술을 탑재한 고가용성 데이터 센터를 운영하고 있습니다. 드물기는 하지만 동일한 위치에 있는 인스턴스의 가용성에 영향을 미치는 장애가 발생할 수도 있습니다. 장애의 영향을 받는 위치 한 곳에서 모든 인스턴스를 호스팅하면 인스턴스를 전혀 사용하지 못하게 될 수 있습니다.

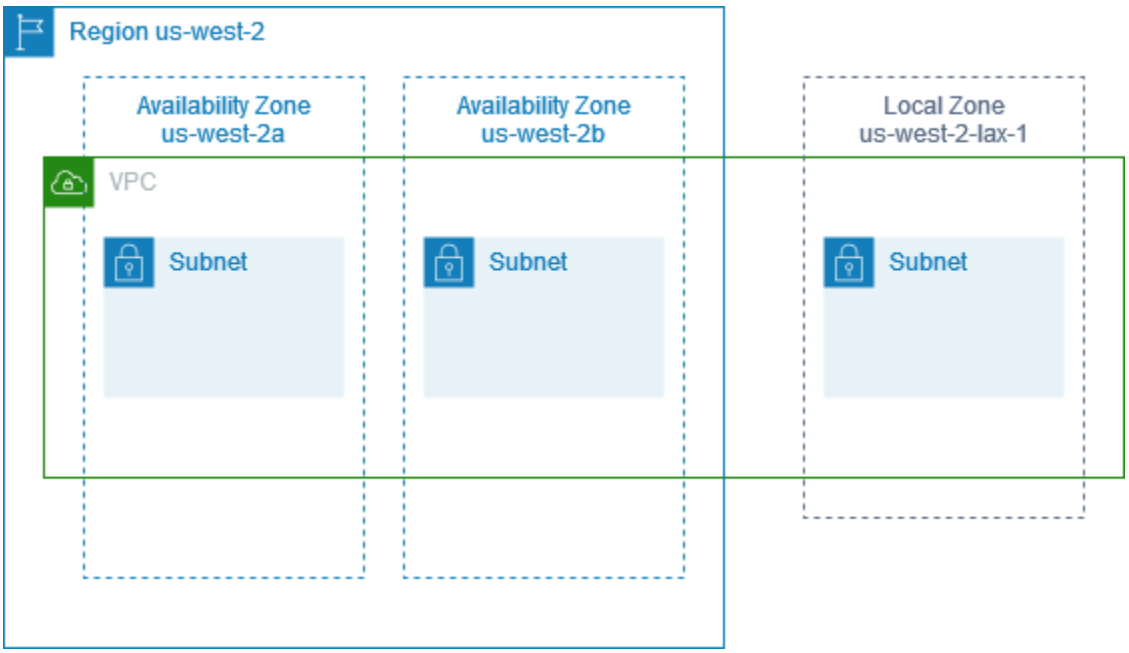
AWS Local Zones의 서브넷

AWS Local Zones를 사용하면 리소스를 최종 사용자에게 가까이 배치하고 친숙한 API 및 도구 세트를 사용하여 AWS 리전의 모든 서비스에 원활하게 연결할 수 있습니다. 로컬 영역에 서브넷을 생성하면 VPC가 해당 로컬 영역으로 확장됩니다.

로컬 영역을 사용하려면 다음과 같은 프로세스를 사용합니다.

- 로컬 영역에 옵트인합니다.
- 로컬 영역에서 서브넷을 만듭니다.
- 애플리케이션이 최종 사용자에게 더 가까이 접근하도록 로컬 영역 서브넷에서 리소스를 시작합니다.

다음 다이어그램은 여러 가용 영역과 하나의 로컬 영역을 아우르는 미국 서부(오레곤)(us-west-2) 리전의 VPC를 보여줍니다.



VPC 를생성할 때 Amazon에서 제공하는 퍼블릭 IP 주소를 VPC 할당하도록 선택할 수 있습니다. 주소를 그룹으로 제한하는 주소에 대한 네트워크 경계 그룹을 설정하도록 설정할 수도 있습니다. 네트워크 경계 그룹을 설정하면 IP 주소가 네트워크 경계 그룹 간에 이동할 수 없습니다. 로컬 영역 네트워크 트래픽은 로컬 영역의 상위 리전을 통과하지 않고 인터넷이나 접속 지점(POP)으로 직접 이동하므로 대기 시간이 짧은 컴퓨팅에 액세스할 수 있습니다. 로컬 영역 및 해당 상위 리전의 모든 목록을 확인하려면 AWS 로컬 영역 사용 설명서의 가용 로컬 영역 섹션을 참조하세요.

로컬 영역에는 다음 규칙이 적용됩니다.

- 로컬 영역 서브넷은 라우팅 테이블, 보안 그룹, 네트워크 ACL 등의 가용 영역 서브넷과 동일한 라우팅 규칙을 따릅니다.
- 아웃바운드 인터넷 트래픽은 로컬 영역을 떠납니다.
- 로컬 영역에서 사용할 퍼블릭 IP 주소를 프로비저닝해야 합니다. 주소를 할당할 때 IP 주소가 공고되는 위치를 지정할 수 있습니다. 이를 네트워크 경계 그룹이라고 하며 이 파라미터를 설정하여 주소를 이 위치로 제한할 수 있습니다. IP 주소를 프로비저닝한 후에는 로컬 영역과 상위 리전(예: `us-west-2-lax-1a`에서 `us-west-2`로) 간에 IP 주소를 이동할 수 없습니다.
- 로컬 영역에서 IPv6를 지원하는 경우, Amazon에서 제공한 IPv6 IP 주소를 요청하여 새 VPC 또는 기존 VPC의 네트워크 경계 그룹과 연결할 수 있습니다. IPv6이 지원되는 로컬 영역의 목록을 확인하려면 AWS 로컬 영역 사용 설명서의 고려 사항을 참조하세요.
- 로컬 영역 서브넷에서는 VPC 엔드포인트를 만들 수 없습니다.

로컬 영역 작업에 대한 자세한 내용은 AWS 로컬 영역 사용 설명서를 참조하세요.

인터넷 게이트웨이에 대한 고려 사항

Local Zones에서 상위 리전의 인터넷 게이트웨이를 사용할 때 다음 사항을 고려하세요.

- 탄력적 IP 주소 또는 Amazon 자동 할당 퍼블릭 IP 주소가 있는 Local Zones에서 인터넷 게이트웨이를 사용할 수 있습니다. 연결하는 탄력적 IP 주소에는 로컬 영역의 네트워크 경계 그룹이 포함되어야 합니다. 자세한 내용은 the section called “탄력적 IP 주소” 단원을 참조하세요.

  리전에 설정된 탄력적 IP 주소를 연결할 수 없습니다.
- Local Zones에서 사용되는 탄력적 IP 주소는 리전의 탄력적 IP 주소와 동일한 할당량을 갖습니다. 자세한 내용은 the section called “탄력적 IP 주소” 단원을 참조하세요.
- 로컬 영역 리소스와 연결된 라우팅 테이블에서 인터넷 게이트웨이를 사용할 수 있습니다. 자세한 내용은 the section called “인터넷 게이트웨이로 라우팅” 단원을 참조하세요.

Direct Connect 게이트웨이를 사용하여 Local Zones에 액세스

온프레미스 데이터 센터에서 로컬 영역에 있는 리소스에 액세스하는 시나리오를 생각해보세요. 로컬 영역과 연결된 VPC에 가상 프라이빗 게이트웨이를 사용하여 로컬 영역을 Direct Connect 게이트웨이에 연결합니다. Direct Connect 게이트웨이는 리전의 Direct Connect 위치에 연결됩니다. 온프레미스 데이터 센터에 해당 Direct Connect 위치에 대한 Direct Connect 연결이 있습니다.

> **Note**
>
> Direct Connect를 사용하여 로컬 영역의 서브넷으로 전송되는 트래픽은 로컬 영역의 상위 리전을 통과하지 않습니다. 그 대신에 트래픽에서는 로컬 영역까지 최단 경로를 이용합니다. 그러면 대기 시간이 감소하고 애플리케이션의 응답성 향상에 도움이 됩니다.

이 구성에 대해 다음 리소스를 구성합니다.

- 로컬 영역 서브넷과 연결된 VPC의 가상 프라이빗 게이트웨이. Amazon VPC 콘솔의 서브넷 세부 정보 페이지에서 서브넷의 VPC를 보거나 describe-subnets 명령을 사용합니다.

  가상 프라이빗 게이트웨이를 만드는 자세한 방법은 AWS Site-to-Site VPN 사용 설명서의 대상 게이트웨이 생성을 참조하세요.
- Direct Connect 연결. 최고의 대기 시간 성능을 위해 AWS는 서브넷을 확장할 로컬 영역에 가장 가까운 Direct Connect 위치를 사용할 것을 권장합니다.

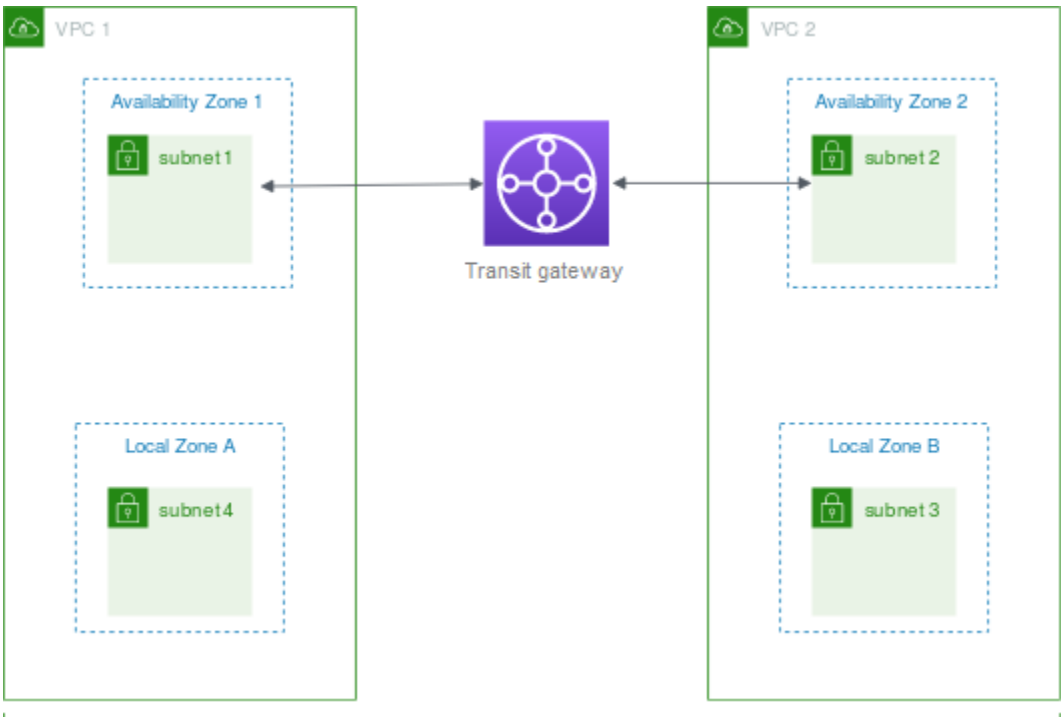
연결 순서 지정 방법에 대한 자세한 내용은 Direct Connect 사용 설명서의 교차 연결을 참조하세요.

- Direct Connect 게이트웨이 Direct Connect 게이트웨이를 만드는 자세한 방법은 Direct Connect 사용 설명서의 Direct Connect 게이트웨이 생성을 참조하세요.
- VPC를 Direct Connect 게이트웨이에 연결하기 위한 가상 프라이빗 게이트웨이 연결입니다. 가상 프라이빗 게이트웨이 연결을 생성하는 자세한 방법은 Direct Connect 사용 설명서의 가상 프라이빗 게이트웨이 연결 및 연결 해제를 참조하세요.
- Direct Connect 위치에서 온프레미스 데이터 센터로의 연결에 대한 프라이빗 가상 인터페이스입니다. Direct Connect 게이트웨이를 만드는 자세한 방법은 Direct Connect 사용 설명서의 Direct Connect 게이트웨이에 대한 프라이빗 가상 인터페이스 생성을 참조하세요.

Transit Gateway에 로컬 영역 서브넷 연결

로컬 영역의 서브넷에 대해서는 Transit Gateway Attachment를 생성할 수 없습니다. 다음 다이어그램에서는 로컬 영역의 서브넷이 상위 가용 영역을 통해 전송 게이트웨이에 연결되도록 네트워크를 구성하는 방법을 보여 줍니다. Local Zones에 서브넷을 만들고 상위 가용 영역에서 서브넷을 생성합니다. 상위 가용 영역의 서브넷을 전송 게이트웨이에 연결한 다음 다른 VPC CIDR로 향하는 트래픽을 Transit Gateway Attachment에 대한 네트워크 인터페이스로 라우팅하는 각 VPC의 라우팅 테이블에 경로를 만듭니다.

> **Note**
>
> Transit Gateway에서 시작하는 로컬 영역의 서브넷으로 향하는 트래픽은 먼저 상위 리전을 통과합니다.



이 시나리오에서는 다음 리소스를 생성합니다.

- 각 상위 가용 영역에 있는 서브넷입니다. 자세한 내용은 the section called “서브넷 생성” 섹션을 참조하세요.
- Transit Gateway. 자세한 내용은 Amazon VPC Transit Gateway의 전송 게이트웨이 생성을 참조하세요.
- 상위 가용 영역을 사용하는 각 VPC에 대한 Transit Gateway Attachment입니다. 자세한 내용은 Amazon VPC Transit Gateway의 VPC에 Transit Gateway Attachment 생성을 참조하세요.
- Transit Gateway Attachment와 연결된 전송 게이트웨이 라우팅 테이블입니다. 자세한 내용은 Amazon VPC Transit Gateways의 Transit Gateway 라우팅 테이블을 참조하세요.
- 각 VPC에 대해 다른 VPC CIDR이 대상으로 있고 Transit Gateway Attachment에 대한 네트워크 인터페이스 ID가 대상으로 있는 로컬 영역 서브넷 라우팅 테이블의 항목입니다. Transit Gateway Attachment의 네트워크 인터페이스를 찾으려면 네트워크 인터페이스 설명에서 Transit Gateway Attachment의 ID를 검색합니다. 자세한 내용은 the section called “전송 게이트웨이에 대한 라우팅” 섹션을 참조하세요.

다음은 VPC 1의 라우팅 테이블의 예입니다.

대상 주소	대상

대상 주소	대상
VPC 1 CIDR	*##*
VPC 2 CIDR	*vpc1-attachment-network-interface-id*

다음은 VPC 2의 라우팅 테이블의 예입니다.

대상 주소	대상
VPC 2 CIDR	*##*
VPC 1 CIDR	*vpc2-attachment-network-interface-id*

다음은 Transit Gateway 라우팅 테이블의 예입니다. 각 VPC의 CIDR 블록이 Transit Gateway 라우팅 테이블에 전파됩니다.

CIDR	연결	경로 유형
VPC 1 CIDR	*VPC 1# ##*	전파
VPC 2 CIDR	*VPC 2# ##*	전파

AWS Wavelength의 서브넷

AWS Wavelength를 사용하면 개발자는 모바일 디바이스 및 최종 사용자에게 매우 짧은 지연 시간을 제공하는 애플리케이션을 빌드할 수 있습니다. Wavelength는 표준 AWS 컴퓨팅 및 스토리지 서비스를 통신 사업자의 5G 네트워크 엣지에 배포합니다. 개발자는 Virtual Private Cloud(VPC)를 하나 이상의 Wavelength Zone으로 확장한 다음, Amazon EC2 인스턴스와 같은 AWS 리소스를 사용하여 매우 짧은 지연 시간으로 리전의 AWS 서비스에 연결해야 하는 애플리케이션을 실행할 수 있습니다.

Wavelength Zone을 사용하려면 먼저 Zone에 옵트인해야 합니다. 그런 다음 Wavelength Zone에 서브넷을 생성합니다. Wavelength Zone에서는 Amazon EC2 인스턴스, Amazon EBS 볼륨, Amazon VPC 서브넷 및 캐리어 게이트웨이를 생성할 수 있습니다. EC2, EBS 및 VPC와 함께 오케스트레이션 또는 연동되는 Amazon EC2 Auto Scaling, Amazon EKS 클러스터, Amazon ECS 클러스터, Amazon EC2 Systems Manager, Amazon CloudWatch, AWS CloudTrail, CloudFormation 등의 서비스를 사용할 수도 있습니다. Wavelength의 서비스는 Amazon DynamoDB 및 Amazon RDS를 비롯한 서비스에 쉽게 액세스할 수 있도록 안정적인 고대역폭 연결을 통해 AWS 리전에 연결되는 VPC의 일부입니다.

Wavelength Zone에는 다음 규칙이 적용됩니다.

- VPC에 서브넷을 생성하여 Wavelength Zone에 연결하면 VPC가 Wavelength Zone으로 확장됩니다.
- 기본적으로 Wavelength Zone에 걸쳐 있는 VPC에서 생성하는 모든 서브넷은 로컬 경로를 포함하여 기본 VPC 라우팅 테이블을 상속합니다.
- Wavelength Zone의 서브넷에서 EC2 인스턴스를 시작할 때 통신 사업자 IP 주소를 할당합니다. 통신 사업자 게이트웨이는 인터페이스에서 인터넷 또는 모바일 디바이스로의 트래픽에 주소를 사용합니다. 통신 사업자 게이트웨이는 NAT를 사용하여 주소를 변환한 다음 트래픽을 대상으로 보냅니다. 전기 통신 사업자 네트워크의 트래픽은 통신 사업자 게이트웨이를 통해 라우팅됩니다.
- VPC 라우팅 테이블 또는 Wavelength Zone의 서브넷 라우팅 테이블의 대상을 통신 사업자 게이트웨이로 설정할 수 있습니다. 이 경우 특정 위치의 통신 사업자 네트워크에서 들어오는 인바운드 트래픽과 통신 사업자 네트워크 및 인터넷으로의 아웃바운드 트래픽이 허용됩니다. Wavelength Zone의 라우팅 옵션에 대한 자세한 내용은 AWS Wavelength 개발자 안내서의 라우팅을 참조하세요.
- Wavelength Zone의 서브넷은 IPv4 주소, DHCP 옵션 세트 및 네트워크 ACL을 포함하여 가용 영역의 서브넷과 동일한 네트워킹 구성 요소를 가지고 있습니다.
- Wavelength 영역의 서브넷에 대해서는 Transit Gateway Attachment를 생성할 수 없습니다. 대신 상위 가용 영역의 서브넷을 통해 연결을 생성한 다음 Transit Gateway를 통해 트래픽을 원하는 대상으로 라우팅합니다. 다음 섹션의 예를 참조하세요.

다중 Wavelength Zone 고려 사항

동일한 VPC의 서로 다른 Wavelength Zone에 있는 EC2 인스턴스는 서로 통신할 수 없습니다. Wavelength Zone과 Wavelength Zone 간 통신이 필요한 경우 AWS에서는 각 Wavelength Zone마다 하나씩 여러 VPC를 사용하는 것이 좋습니다. 전송 게이트웨이를 사용하여 VPC를 연결할 수 있습니다. 이 구성을 사용하면 Wavelength Zone의 인스턴스 간에 통신할 수 있습니다.

Wavelength Zone과 Wavelength Zone 간 트래픽은 AWS 리전을 통해 라우팅됩니다. 자세한 내용은 AWS Transit Gateway를 참조하세요.

다음 다이어그램은 서로 다른 두 Wavelength Zone의 인스턴스가 통신할 수 있도록 네트워크를 구성하는 방법을 보여줍니다. 두 개의 Wavelength Zone(Wavelength Zone A 및 Wavelength Zone B)이 있습니다. 통신을 활성화하려면 다음 리소스를 생성해야 합니다.

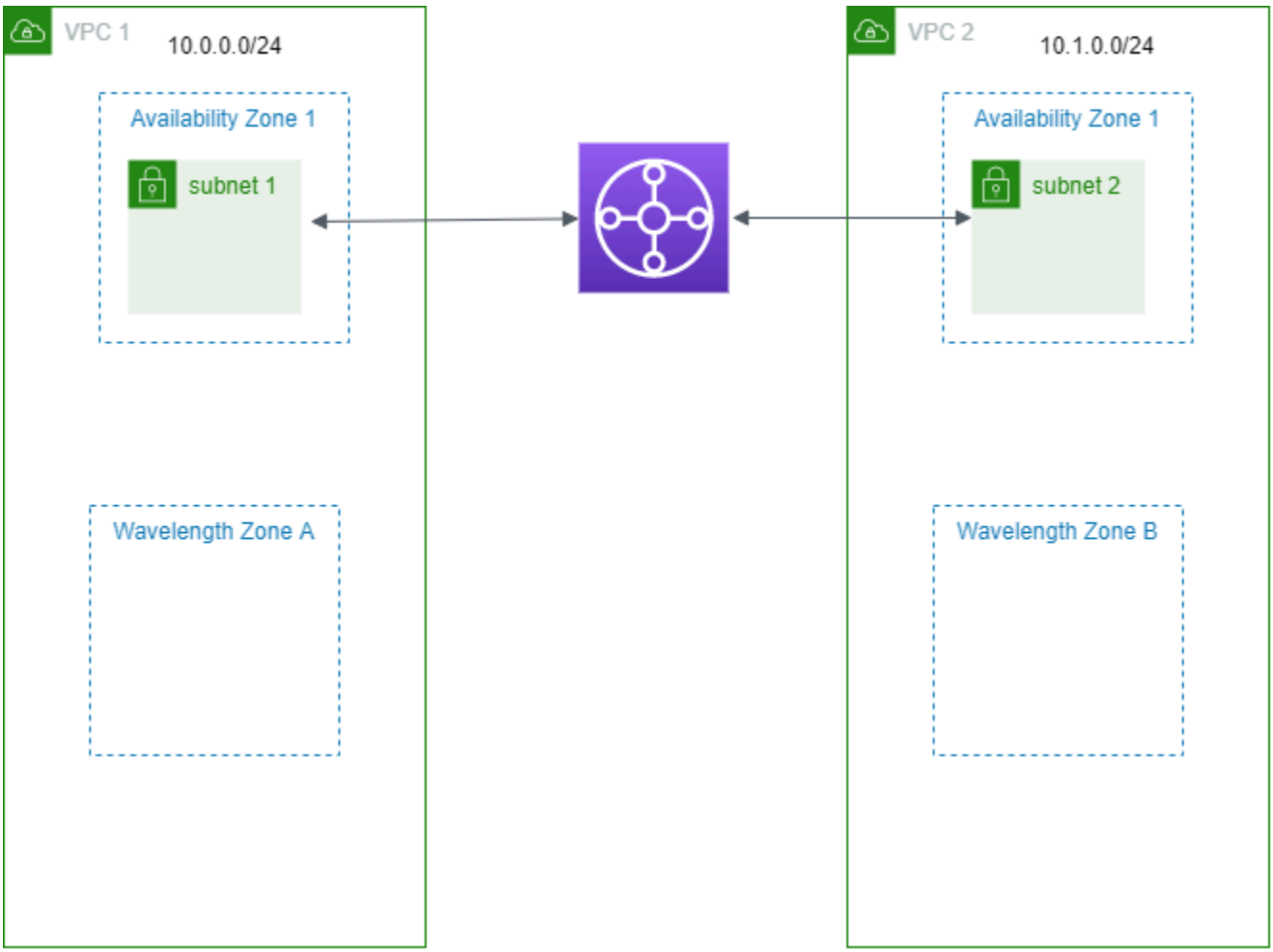
- 각 Wavelength Zone에 대해 Wavelength Zone의 상위 가용 영역인 가용 영역의 서브넷. 이 예에서는 서브넷 1과 서브넷 2를 생성합니다. 서브넷 생성에 대한 자세한 내용은 the section called “서브넷 생성” 단원을 참조하세요. describe-availability-zones 명령을 사용하여 상위 영역을 찾습니다.
- Transit Gateway. 전송 게이트웨이는 VPC를 연결합니다. Transit Gateway를 생성하는 자세한 방법은 Amazon VPC Transit Gateways 가이드의 Transit Gateway 생성을 참조하세요.
- 각 VPC에 대해 Wavelength 영역의 상위 가용 영역에 있는 Transit Gateway에 대한 VPC 연결입니다. 자세한 내용은 Amazon VPC Transit Gateway 가이드의 VPC에 전송 게이트웨이 연결을 참조하세요.
- Transit Gateway 라우팅 테이블의 각 VPC에 대한 항목. Transit Gateway 경로 생성에 대한 자세한 내용은 Amazon VPC Transit Gateways 안내서에서 Transit Gateway 라우팅 테이블을 참조하세요.
- 각 VPC에서, 대상으로 다른 VPC CIDR을 목적지로 사용하며 전송 게이트웨이 ID를 대상으로 사용하는 VPC 라우팅 테이블의 항목. 자세한 내용은 the section called “전송 게이트웨이에 대한 라우팅” 단원을 참조하세요.

  이 예에서 VPC 1의 라우팅 테이블에는 다음 항목이 있습니다.

대상 주소	대상
10.1.0.0/24	tgw-22222222222222222

VPC 2의 라우팅 테이블에는 다음 항목이 있습니다.

대상 주소	대상
10.0.0.0/24	tgw-22222222222222222



AWS Outposts의 서브넷

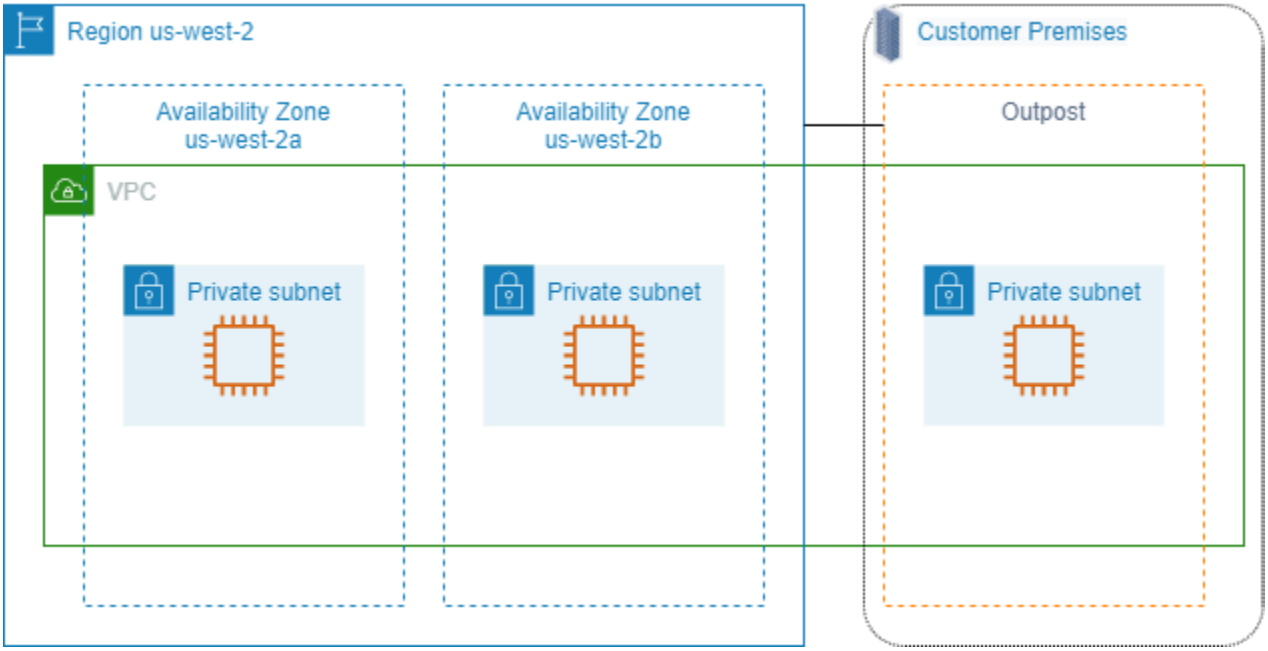
AWS Outposts는 온프레미스 및 클라우드에서 애플리케이션을 구축하고 실행할 수 있는 것과 동일한 AWS 하드웨어 인프라, 서비스, API 및 툴을 제공합니다. AWS Outposts는 온프레미스 애플리케이션 또는 시스템에 대한 짧은 대기 시간 액세스가 필요한 워크로드와 로컬에서 데이터를 저장 및 처리해야 하는 워크로드에 적합합니다. 에 대한 자세한 내용은 AWS Outposts 섹션을 참조하세요..AWS Outposts

VPC는 AWS 리전의 모든 가용 영역에 적용됩니다. Outpost를 상위 리전에 연결한 후 해당 VPC의 Outpost를 위한 서브넷을 생성하여 해당 리전의 VPC를 Outpost로 확장할 수 있습니다.

다음 규칙은 AWS Outposts에 적용됩니다.

- 서브넷은 하나의 Outposts 위치에 있어야 합니다.
- Outpost를 위한 서브넷을 생성하려면 서브넷을 생성할 때 Outpost의 Amazon 리소스 이름(ARN)을 지정합니다.

- Outposts 랙 - 로컬 게이트웨이가 VPC와 온프레미스 네트워크 간의 네트워크 연결을 처리합니다. 자세한 내용은 Outposts 랙용 AWS Outposts 사용 설명서의 로컬 게이트웨이를 참조하세요.
- Outposts 랙 - 로컬 게이트웨이가 VPC와 온프레미스 네트워크 간의 네트워크 연결을 처리합니다. 자세한 내용은 Outposts 서버 AWS Outposts 사용 설명서의 로컬 네트워크 인터페이스를 참조하세요.
- 기본적으로 Outposts의 서브넷을 포함하여 VPC에서 생성하는 모든 서브넷은 암시적으로 VPC의 기본 라우팅 테이블에 연결됩니다. 또는 사용자 지정 라우팅 테이블을 VPC의 서브넷과 명시적으로 연결하고 로컬 게이트웨이를 온프레미스 네트워크로 라우팅할 모든 트래픽에 대한 다음 홉 대상으로 사용할 수 있습니다.



VPC 삭제

VPC 사용을 마치면 이를 삭제할 수 있습니다.

요구 사항

VPC를 삭제하려면 먼저 VPC에서 요청자 관리형 네트워크 인터페이스를 생성한 모든 리소스를 종료하거나 삭제해야 합니다. 예를 들어 EC2 인스턴스를 종료하고 로드 밸런서, NAT 게이트웨이, 전송 게이트웨이 VPC 연결 및 인터페이스 VPC 엔드포인트를 삭제해야 합니다.

> **Note**
>
> 삭제 중인 VPC에 대한 흐름 로그를 생성한 경우 삭제된 VPC에 대한 흐름 로그는 결국 자동으로 제거된다는 점을 참고하세요.

내용

- 콘솔을 사용하여 VPC 삭제
- 명령줄을 사용하여 VPC 삭제

콘솔을 사용하여 VPC 삭제

Amazon VPC 콘솔을 사용하여 VPC를 삭제하면 다음과 같은 VPC 구성 요소도 삭제됩니다.

- DHCP 옵션
- 외부 전용 인터넷 게이트웨이
- 게이트웨이 엔드포인트
- 인터넷 게이트웨이
- 네트워크 ACL
- 라우팅 테이블
- 보안 그룹
- 서브넷

콘솔을 사용하여 VPC를 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. VPC에서 모든 인스턴스를 종료합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 종료를 참조하세요.
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. VPC를 삭제하기 전에 삭제하거나 종료해야 하는 리소스가 있는 경우 해당 리소스가 표시됩니다. 해당 리소스를 삭제하거나 종료한 다음 다시 시도하세요. 그렇지 않으면 VPC와 함께 삭제할 리소스가 표시됩니다. 목록을 검토한 후 다음 단계로 이동합니다.

7. (선택 사항) Site-to-Site VPN 연결이 있는 경우 삭제할 옵션을 선택할 수 있습니다. 다른 VPC에서 이 고객 게이트웨이를 사용할 계획이라면 Site-to-Site VPN 연결 및 게이트웨이를 유지하는 것이 좋습니다. 그렇지 않으면 새 Site-to-Site VPN 연결을 생성한 후 고객 게이트웨이 디바이스를 다시 구성해야 합니다.
8. 확인 메시지가 나타나면 **delete**을 입력한 다음 삭제를 선택합니다.

명령줄을 사용하여 VPC 삭제

명령줄을 사용하여 VPC를 삭제하려면 먼저 VPC에서 요청자 관리형 네트워크 인터페이스를 생성한 모든 리소스를 종료하거나 삭제해야 합니다. 또한 서브넷, 보안 그룹, 네트워크 ACL, 라우팅 테이블, 인터넷 게이트웨이 및 송신 전용 인터넷 게이트웨이와 같은 생성한 모든 VPC 리소스를 삭제하거나 분리해야 합니다. 기본 보안 그룹, 기본 라우팅 테이블 또는 기본 네트워크 ACL은 삭제할 필요가 없습니다.

다음 절차는 일반적인 VPC 리소스를 삭제한 다음 VPC를 삭제하는 데 사용하는 명령을 보여줍니다. 이러한 명령을 이 순서대로 사용해야 합니다. VPC 리소스를 추가로 생성한 경우 VPC를 삭제하기 전에 해당하는 삭제 명령도 사용해야 합니다.

AWS CLI를 사용하여 VPC 삭제

1. delete-security-group 명령을 사용하여 보안 그룹을 삭제합니다.

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

2. delete-network-acl 명령을 사용하여 각 네트워크 ACL을 삭제합니다.

```
aws ec2 delete-network-acl --network-acl-id acl-id
```

3. delete-subnet 명령을 사용하여 각 서브넷을 삭제합니다.

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

4. delete-route-table 명령을 사용하여 각 사용자 지정 라우팅 테이블을 삭제합니다.

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

5. detach-internet-gateway 명령을 사용하여 VPC에서 인터넷 게이트웨이를 분리합니다.

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-id --vpc-id vpc-id
```

6. delete-internet-gateway 명령을 사용하여 인터넷 게이트웨이를 삭제합니다.

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

7. [듀얼 스택 VPC] delete-egress-only-internet-gateway 명령을 사용하여 송신 전용 인터넷 게이트웨이를 삭제합니다.

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id eigw-id
```

8. delete-vpc 명령을 사용하여 VPC를 삭제합니다.

```
aws ec2 delete-vpc --vpc-id vpc-id
```

콘솔 투 코드를 사용하여 VPC 콘솔 작업에서 코드형 인프라 생성

콘솔은 리소스를 생성하고 프로토타입을 테스트하기 위한 안내 경로를 제공합니다. 대규모로 동일한 리소스를 생성하려면 자동화 코드가 필요합니다. 콘솔 투 코드는 자동화 코드를 시작하는 데 도움이 되는 Amazon Q Developer의 기능입니다. 콘솔 투 코드는 기본값과 호환 가능한 파라미터를 포함하여 콘솔 작업을 기록합니다. 그런 다음 생성형 AI를 사용하여 원하는 작업에 대해 선호하는 코드형 인프라(IaC) 형식의 코드를 제안합니다. 콘솔 워크플로는 사용자가 지정한 파라미터 값이 모두 유효한지 확인하기 때문에 콘솔 투 코드를 사용하여 생성하는 코드는 호환되는 파라미터 값을 가집니다. 코드를 시작점으로 사용하여 특정 사용 사례에서 프로덕션에 바로 사용할 수 있도록 사용자 지정할 수 있습니다.

예를 들어 콘솔 투 코드에서는 자신의 작업을 기록하면서 VPC 콘솔을 사용하여 서브넷, 보안 그룹, NACL, 사용자 지정 라우팅 테이블, 인터넷 게이트웨이를 생성하고 CloudFormation JSON 형식으로 코드를 생성할 수 있습니다. 그런 다음 해당 코드를 복사하고 CloudFormation 템플릿에서 사용할 수 있도록 사용자 지정할 수 있습니다.

콘솔 투 코드는 현재 다음과 같은 언어 및 형식으로 코드형 인프라(IAC)를 생성할 수 있습니다.

- CDK Java
- CDK Python
- CDK TypeScript
- CloudFormation JSON
- CloudFormation YAML

콘솔 투 코드를 사용하는 방법에 대한 자세한 내용 및 지침은 Amazon Q Developer 사용 설명서의 Amazon Q Developer 콘솔 투 코드를 사용하여 AWS 서비스 자동화를 참조하세요.

VPC의 서브넷

서브넷은 VPC의 IP 주소 범위입니다. 특정 서브넷에서 EC2 인스턴스와 같은 AWS 리소스를 생성할 수 있습니다.

내용

- 서브넷 기본 사항
- 서브넷 보안
- 서브넷 생성
- 서브넷에서 IPv6 CIDR 블록 추가 또는 제거
- 서브넷의 IP 주소 지정 속성 수정
- 서브넷 CIDR 예약
- 라우팅 테이블 구성
- 미들박스 라우팅 마법사
- 서브넷 삭제

서브넷 기본 사항

각 서브넷은 단일 가용 영역 내에서만 존재해야 하며, 여러 영역으로 스케일 아웃할 수 없습니다. 별도의 가용 영역에서 AWS 리소스를 시작하면 단일 가용 영역의 장애로부터 애플리케이션을 보호할 수 있습니다.

내용

- 서브넷 IP 주소 범위
- 서브넷 유형
- 서브넷 다이어그램
- 서브넷 라우팅
- 서브넷 설정

서브넷 IP 주소 범위

서브넷을 만들 때 VPC 구성에 따라 다음과 같이 IP 주소를 지정합니다.

- IPv4 전용 - 서브넷에 IPv4 CIDR 블록은 있지만 IPv6 CIDR 블록은 없습니다. IPv4 전용 서브넷의 리소스는 IPv4를 통해 통신해야 합니다.
- 듀얼 스택 - 서브넷에 IPv4 CIDR 블록 및 IPv6 CIDR 블록이 둘 다 있습니다. VPC에는 IPv4 CIDR 블록 및 IPv6 CIDR 블록이 둘 다 있어야 합니다. 듀얼 스택 서브넷의 리소스는 IPv4 및 IPv6를 통해 통신할 수 있습니다.
- IPv6 전용 - 서브넷에 IPv6 CIDR 블록은 있지만 IPv4 CIDR 블록은 없습니다. VPC에 IPv6 CIDR 블록이 있어야 합니다. IPv6 전용 서브넷의 리소스는 IPv6를 통해 통신해야 합니다.

> **Note**
>
> IPv6 전용 서브넷에 있는 리소스에는 CIDR 차단 169.254.0.0/16로부터 IPv4 링크-로컬 주소가 할당됩니다. 이러한 주소는 VPC에서만 사용할 수 있는 서비스와 통신하는 데 사용됩니다. 예제는 Amazon EC2 사용 설명서의 링크-로컬 주소를 참조하세요.

자세한 내용은 VPC 및 서브넷의 IP 주소 지정 섹션을 참조하세요.

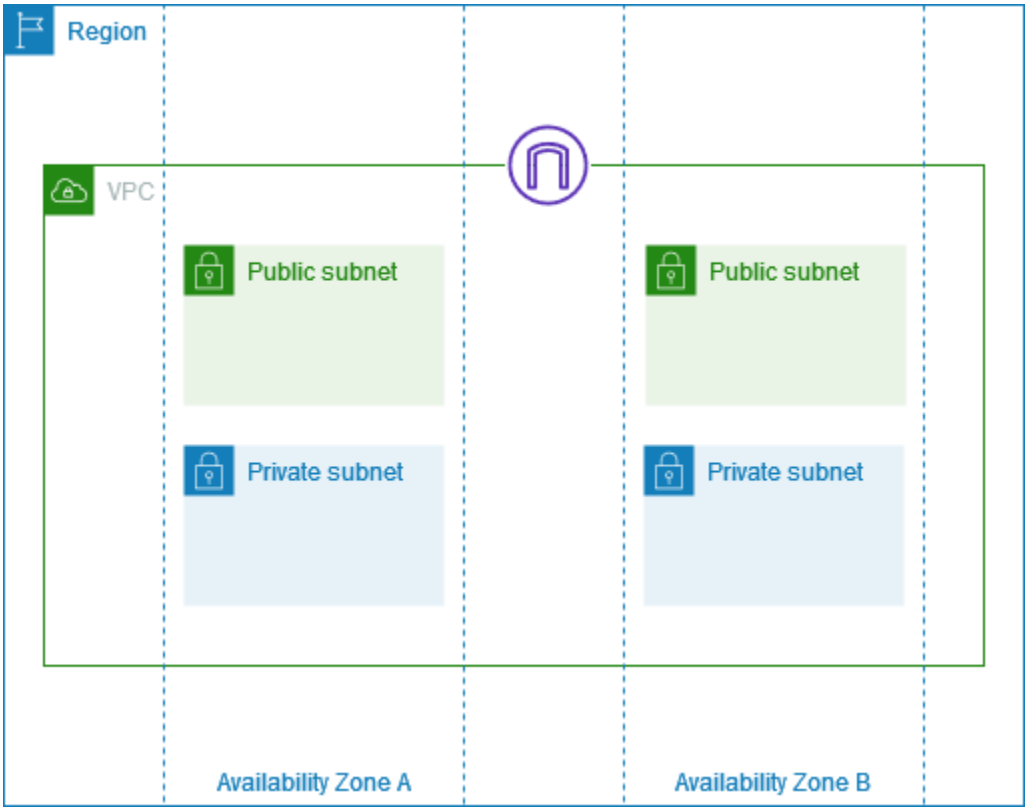
서브넷 유형

서브넷 유형은 서브넷에 대한 라우팅을 구성하는 방법에 따라 결정됩니다. 예제:

- 퍼블릭 서브넷 - 서브넷에 인터넷 게이트웨이로 직접 연결되는 경로가 있습니다. 퍼블릭 서브넷의 리소스는 퍼블릭 인터넷에 액세스할 수 있습니다.
- 프라이빗 서브넷 - 서브넷에 인터넷 게이트웨이로 직접 연결되는 경로가 없습니다. 프라이빗 서브넷의 리소스에는 퍼블릭 인터넷에 액세스하기 위해 NAT 디바이스가 필요합니다.
- VPN 전용 서브넷 - 서브넷에 가상 프라이빗 게이트웨이를 통해 Site-to-Site VPN 연결으로 연결되는 경로가 있습니다. 서브넷에는 인터넷 게이트웨이에 대한 경로가 없습니다.
- 격리된 서브넷 - 서브넷에 VPC 외부 대상에 대한 경로가 없습니다. 격리된 서브넷의 리소스는 동일한 VPC의 다른 리소스와만 서로 액세스할 수 있습니다.
- EVS 서브넷 - 이 유형의 서브넷은 Amazon EVS를 사용하여 생성됩니다. 자세한 내용은 Amazon EVS 사용 설명서의 VLAN 서브넷을 참조하세요.

서브넷 다이어그램

다음 다이어그램은 두 개의 가용 영역 인터넷 게이트웨이에 있는 서브넷을 포함한 VPC를 보여 줍니다. 각 가용 영역에는 퍼블릭 서브넷과 프라이빗 서브넷이 있습니다.



로컬 영역 및 Wavelength 영역의 서브넷을 보여주는 다이어그램은 AWS 로컬 영역의 작동 방식 및 AWS Wavelength 작동 방식을 참조하세요.

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 내용은 라우팅 테이블 구성 섹션을 참조하세요.

서브넷 설정

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(예: eth0)가 포함됩니다. 서브넷 속성에 상관없이 특정 인스턴스를 시작하는 중에 해당 인스턴스에 대한 이 설정을 재정의할 수 있습니다.

서브넷을 생성한 후 서브넷에 대한 다음 설정을 수정할 수 있습니다.

- IP 설정 자동 할당: 이 서브넷의 새 네트워크 인터페이스에 대한 퍼블릭 IPv4 또는 IPv6 주소를 자동으로 요청하도록 자동 할당 IP 설정을 구성할 수 있습니다.

- 리소스 기반 이름(RBN) 설정: 이 서브넷에서 EC2 인스턴스에 대한 호스트 이름 유형을 지정하고 DNS A 및 AAAA 레코드 쿼리가 처리되는 방법을 구성할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스 호스트 이름 유형을 참조하세요.

서브넷 보안

AWS 리소스를 보호하려면 프라이빗 서브넷을 사용하는 것이 좋습니다. Bastion Host 또는 NAT 디바이스를 사용하여 프라이빗 서브넷에 있는 EC2 인스턴스와 같은 리소스에 대한 인터넷 액세스를 제공합니다.

AWS은(는) VPC에서 리소스에 대한 보안을 강화하기 위해 사용할 수 있는 기능을 제공합니다. 보안 그룹은 EC2 인스턴스와 같은 관련 리소스에 대한 인바운드 및 아웃바운드 트래픽을 허용합니다. 네트워크 ACL은 사용하여 서브넷 수준에서 인바운드 및 아웃바운드 트래픽을 허용하거나 제어합니다. 대부분의 경우 보안 그룹은 사용자의 요구 사항을 충족할 수 있습니다. 단, 추가 보안 계층을 원하는 경우 네트워크 ACL을 사용할 수 있습니다. 자세한 내용은 the section called “보안 그룹 및 네트워크 ACL 비교” 섹션을 참조하세요.

각 서브넷에는 네트워크 ACL이 연결되어야 합니다. 생성하는 모든 서브넷은 VPC의 기본 네트워크 ACL과 자동으로 연결됩니다. 기본 네트워크 ACL은 인바운드와 아웃바운드 트래픽을 모두 허용합니다. 기본 네트워크 ACL을 업데이트하거나, 사용자 지정 네트워크 ACL을 생성하여 서브넷에 연결할 수 있습니다. 자세한 내용은 네트워크 액세스 제어 목록으로 서브넷 트래픽 제어 섹션을 참조하세요.

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. 자세한 내용은 VPC 흐름 로그를 사용하여 IP 트래픽 로깅 섹션을 참조하세요.

서브넷 생성

다음 절차에 따라 Virtual Private Cloud(VPC)에 대한 서브넷을 생성합니다. 필요한 연결에 따라 게이트웨이와 라우팅 테이블을 추가해야 할 수도 있습니다.

고려 사항

- VPC의 범위로부터 서브넷에 대해 IPv4 CIDR 블록을 지정해야 합니다. IPv6 CIDR 블록이 VPC와 연결되어 있는 경우 서브넷에 대해 IPv6 CIDR 블록을 지정할 수도 있습니다. 자세한 내용은 VPC 및 서브넷의 IP 주소 지정 섹션을 참조하세요.

- IPv6 전용 서브넷을 생성하는 경우 다음 사항에 유의합니다. IPv6 전용 서브넷에서 시작된 EC2 인스턴스는 IPv6 주소를 수신하지만 IPv4 주소는 수신하지 않습니다. IPv6 전용 서브넷으로 시작하는 모든 인스턴스는 Nitro 시스템에 구축된 인스턴스여야 합니다.
- 로컬 영역 또는 Wavelength 영역에 서브넷을 생성하려면 영역을 활성화해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서에서 리전 및 가용 영역을 참조하세요.

VPC에 서브넷을 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷 생성(Create subnet)을 선택합니다.
4. VPC ID에서 서브넷의 VPC를 선택합니다.
5. (선택 사항) 서브넷 이름(Subnet name)에 서브넷의 이름을 입력합니다. 이렇게 하면 Name 키와 지정한 값으로 태그가 생성됩니다.
6. 가용 영역(Availability Zone)에서 서브넷의 영역을 선택하거나 AWS에서 자동으로 선택하도록 기본값인 기본 설정 없음(No Preference)을 그대로 둘 수 있습니다.
7. IPv4 CIDR 블록의 경우 수동 입력을 선택하여 서브넷의 IPv4 CIDR 블록을 입력하거나(예: `10.0.1.0/24`) IPv4 CIDR 없음을 선택합니다. Amazon VPC IP 주소 관리자(IPAM)를 사용하여 AWS 워크로드의 IP 주소를 계획, 추적 및 모니터링하는 경우 서브넷을 생성할 때 IPAM(IPAM 할당)에서 CIDR 블록을 할당할 수 있는 옵션이 제공됩니다. 서브넷 IP 할당을 위한 VPC IP 주소 공간 계획에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 자습서: 서브넷 IP 할당을 위한 VPC IP 주소 공간 계획을 참조하세요.
8. IPv6 CIDR 블록의 경우 수동 입력을 선택하여 서브넷을 생성할 VPC의 IPv6 CIDR을 선택합니다. 이 옵션은 VPC에 연결된 IPv6 CIDR 블록이 있는 경우에만 사용할 수 있습니다. Amazon VPC IP 주소 관리자(IPAM)를 사용하여 AWS 워크로드의 IP 주소를 계획, 추적 및 모니터링하는 경우 서브넷을 생성할 때 IPAM(IPAM 할당)에서 CIDR 블록을 할당할 수 있는 옵션이 제공됩니다. 서브넷 IP 할당을 위한 VPC IP 주소 공간 계획에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 자습서: 서브넷 IP 할당을 위한 VPC IP 주소 공간 계획을 참조하세요.
9. IPv6 VPC CIDR 블록을 선택합니다.
10. IPv6 서브넷 CIDR 블록의 경우 VPC CIDR과 같거나 더 구체적인 서브넷의 CIDR을 선택합니다. 예를 들어 VPC 풀 CIDR이 /50인 경우 서브넷의 넷마스크 길이를 /50에서 /64 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/64(/4씩 증가)입니다.
11. 서브넷 생성(Create subnet)을 선택합니다.

AWS CLI를 사용하여 VPC에 서브넷을 추가하려면

create-subnet 명령을 사용합니다.

다음 단계

서브넷을 생성한 후 다음과 같이 구성할 수 있습니다.

- 라우팅 구성. 그런 다음 인터넷 게이트웨이와 같이 VPC와 연결된 게이트웨이로 트래픽을 전송하는 사용자 정의 라우팅 테이블과 라우팅을 생성할 수 있습니다. 자세한 내용은 라우팅 테이블 구성 섹션을 참조하세요.
- IP 주소 지정 동작 수정. 서브넷에서 시작하는 인스턴스가 퍼블릭 IPv4 주소, IPv6 주소 또는 둘 다 받도록 지정할 수 있습니다. 자세한 내용은 서브넷의 IP 주소 지정 속성 수정 섹션을 참조하세요.
- RBN(리소스 기반 이름) 설정을 수정합니다. 자세한 내용은 Amazon EC2 인스턴스 호스트 이름 유형을 참조하십시오.
- 네트워크 ACL을 생성하거나 수정합니다. 자세한 내용은 네트워크 액세스 제어 목록으로 서브넷 트래픽 제어 섹션을 참조하세요.
- 다른 계정과 서브넷을 공유합니다. 자세한 내용은 ??? 섹션을 참조하세요.

서브넷에서 IPv6 CIDR 블록 추가 또는 제거

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

서브넷에서 IPv6 지원이 더 이상 필요 없지만 IPv4 리소스 생성 및 IPv4 리소스와의 통신을 위해 서브넷을 계속 사용하려는 경우 IPv6 CIDR 블록을 제거할 수 있습니다.

IPv6 CIDR 블록을 제거하려면 먼저 서브넷의 모든 인스턴스에 할당된 모든 IPv6 주소를 할당 해제해야 합니다.

서브넷에서 IPv6 CIDR 블록을 추가하거나 제거하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택하고 작업(Actions), IPv6 CIDR 편집(Edit IPv6 CIDRs)을 선택합니다.
4. CIDR을 추가하려면 Add IPv6 CIDR 추가와 VPC CIDR 블록을 차례로 선택하고 서브넷 CIDR 블록을 입력한 다음 VPC CIDR의 넷마스크 길이와 같거나 더 구체적인 넷마스크 길이를 선택합니

다. 예를 들어 VPC 풀 CIDR이 /50인 경우 서브넷의 넷마스크 길이를 /50에서 /64 사이로 선택할 수 있습니다. 가능한 IPv6 넷마스크 길이는/44~/64(/4씩 증가)입니다.

5. CIDR을 제거하려면 IPv6 CIDR 블록을 찾고 제거를 선택합니다.
6. 저장을 선택합니다.

AWS CLI를 사용하여 IPv6 CIDR 블록을 서브넷에 연결하려면

associate-subnet-cidr-block 명령을 사용합니다.

AWS CLI를 사용하여 서브넷에서 IPv6 CIDR 블록을 연결 해제하려면

disassociate-subnet-cidr-block 명령을 사용합니다.

서브넷의 IP 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. 단 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.

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

서브넷에는 연결된 IPv6 CIDR 블록이 있어야 합니다.

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

서브넷의 IP 주소 지정 동작을 수정하려면 다음을 수행하세요.

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

2. 탐색 창에서 서브넷(Subnets)을 선택합니다.
3. 서브넷을 선택하고 작업(Actions), 서브넷 설정 편집(Edit subnet settings)을 선택합니다.
4. 퍼블릭 IPv4 주소 자동 할당 활성화(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.
5. 퍼블릭 IPv4 주소 자동 할당 활성화(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.

AWS CLI를 사용하여 서브넷 속성을 수정하려면

modify-subnet-attribute 명령을 사용합니다.

서브넷 CIDR 예약

서브넷 CIDR 예약은 AWS가 사용자의 네트워크 인터페이스에 할당하지 않도록 따로 설정한 IPv4 또는 IPv6 주소 범위입니다. 이렇게 하면 네트워크 인터페이스에 사용할 IPv4 또는 IPv6 CIDR 블록('접두사'라고도 함)을 예약할 수 있습니다.

서브넷 CIDR 예약을 생성할 때 예약된 ID 주소를 사용하는 방법을 지정합니다. 다음과 같은 옵션을 사용할 수 있습니다.

- 접두사 - 단일 네트워크 인터페이스에 접두사를 할당하도록 허용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 네트워크 인터페이스에 접두사 할당을 참조하세요.
- 명시적 - 단일 네트워크 인터페이스에 개별 IP 주소를 수동으로 할당하도록 허용합니다.

다음 규칙이 서브넷 CIDR 예약에 적용됩니다.

- 서브넷 CIDR 예약을 생성할 때 IP 주소 범위에는 이미 사용 중인 주소가 포함될 수 있습니다. 서브넷 예약을 생성해도 이미 사용 중인 IP 주소는 할당 해제되지 않습니다.
- 서브넷당 여러 CIDR 범위를 예약할 수 있습니다. 동일한 VPC 내에 여러 CIDR 범위를 예약하면 CIDR 범위가 겹쳐지지 않습니다.
- 접두사 위임을 위해 서브넷에서 2개 이상의 범위를 예약하고 접두사 위임을 자동 할당하도록 구성한 경우 네트워크 인터페이스에 할당할 IP 주소를 임의로 선택합니다.
- 서브넷 예약을 삭제하면 사용하지 않은 IP 주소를 AWS에서 네트워크 인터페이스에 할당할 수 있습니다. 서브넷 예약을 삭제해도 사용 중인 IP 주소는 할당 해제되지 않습니다.

- 예약 유형은 서브넷에 사용 가능한 IP 주소 수에 영향을 줍니다. 접두사 예약을 생성하면 개수가 즉시 감소합니다. 명시적 접두사 예약을 생성하면 IP 주소가 할당될 때 수가 감소합니다.

Classless Inter-Domain Routing(CIDR) 표기법에 대한 자세한 내용은 IP 주소 지정 섹션을 참조하세요.

내용

- 콘솔을 사용한 서브넷 CIDR 예약 작업
- AWS CLI를 사용한 서브넷 CIDR 예약 작업

콘솔을 사용한 서브넷 CIDR 예약 작업

다음과 같이 서브넷 CIDR 예약을 만들고 관리할 수 있습니다.

서브넷 CIDR 예약 편집

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. 기존 서브넷 CIDR 예약에 대한 정보를 얻으려면 CIDR 예약 탭을 선택합니다.
5. 서브넷 CIDR 예약을 추가하거나 제거하려면 작업, CIDR 예약 편집을 선택한 후 다음을 수행합니다.
   - IPv4 CIDR 예약을 추가하려면 IPv4, IPv4 CIDR 예약 추가를 선택합니다. 예약 유형을 선택하고 CIDR 범위를 입력한 다음 추가를 선택합니다.
   - IPv6 CIDR 예약을 추가하려면IPv6, IPv6 CIDR 예약 추가를 선택합니다. 예약 유형을 선택하고 CIDR 범위를 입력한 다음 추가를 선택합니다.
   - CIDR 예약을 제거하려면 서브넷 CIDR 예약에 대해 제거를 선택합니다.

AWS CLI를 사용한 서브넷 CIDR 예약 작업

AWS CLI를 사용하여 서브넷 CIDR 예약을 만들고 관리할 수 있습니다.

Tasks

- 서브넷 CIDR 예약 만들기
- 서브넷 CIDR 예약 보기
- 서브넷 CIDR 예약 삭제

서브넷 CIDR 예약 만들기

create-subnet-cidr-reservation을 사용하여 서브넷 CIDR 예약을 생성합니다.

```
aws ec2 create-subnet-cidr-reservation --subnet-id subnet-03c51e2eEXAMPLE --
reservation-type prefix --cidr 2600:1f13:925:d240:3a1b::/80
```

출력의 예시는 다음과 같습니다.

```
{
    "SubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2ef5EXAMPLE",
        "Cidr": "2600:1f13:925:d240:3a1b::/80",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

서브넷 CIDR 예약 보기

get-subnet-cidr-reservations을 사용하여 서브넷 CIDR 예약의 세부 정보를 볼 수 있습니다.

```
aws ec2 get-subnet-cidr-reservations --subnet-id subnet-05eef9fb78EXAMPLE
```

서브넷 CIDR 예약 삭제

delete-subnet-cidr-reservation을 사용하여 서브넷 CIDR 예약을 삭제합니다.

```
aws ec2 delete-subnet-cidr-reservation --subnet-cidr-reservation-
id scr-044f977c4eEXAMPLE
```

라우팅 테이블 구성

라우팅 테이블은 Virtual Private Cloud(VPC)의 트래픽 컨트롤러 역할을 수행합니다. 각 라우팅 테이블에는 라우팅이라는 규칙 세트가 포함되어 있어 서브넷 또는 게이트웨이의 네트워크 트래픽이 전송되는 위치를 결정합니다. VPC를 생성할 때 해당 VPC의 기본 라우팅 테이블도 생성됩니다. VPC에 대해 라우팅 테이블을 추가로 생성하여 VPC의 네트워크 경로를 보다 세분화하여 제어할 수 있습니다.

라우팅 테이블을 사용해 다른 VPC 또는 온프레미스 네트워크와 같이 VPC가 통신할 수 있는 네트워크를 지정할 수 있습니다. 각 라우팅은 목적지(CIDR 블록 또는 접두사 목록)과 대상(예: 인터넷 게이트웨이, NAT 게이트웨이, VPC 피어링 연결 또는 VPN 연결)을 지정합니다. 트래픽은 목적지 IP 주소에 기반해 대상으로 라우팅됩니다. 라우팅 테이블을 사용하면 퍼블릭 서브넷, 프라이빗 서브넷, VPN 전용 서브넷, 격리된 서브넷 등을 포함하는 복잡한 네트워킹 아키텍처를 생성할 수 있습니다.

내용

- 라우팅 테이블 개념
- 서브넷 라우팅 테이블
- 게이트웨이 라우팅 테이블
- 라우팅 우선 순위 작동 방식
- 라우팅 옵션 예
- VPC에 대해 라우팅 테이블 생성
- 서브넷 라우팅 테이블 관리
- 기본 라우팅 테이블 교체
- 게이트웨이 라우팅 테이블을 사용하여 VPC로 들어오는 트래픽 제어
- 로컬 경로의 대상 교체 또는 복원
- VPC의 고급 라우팅
- VPC의 연결성 문제 해결

라우팅 테이블 개념

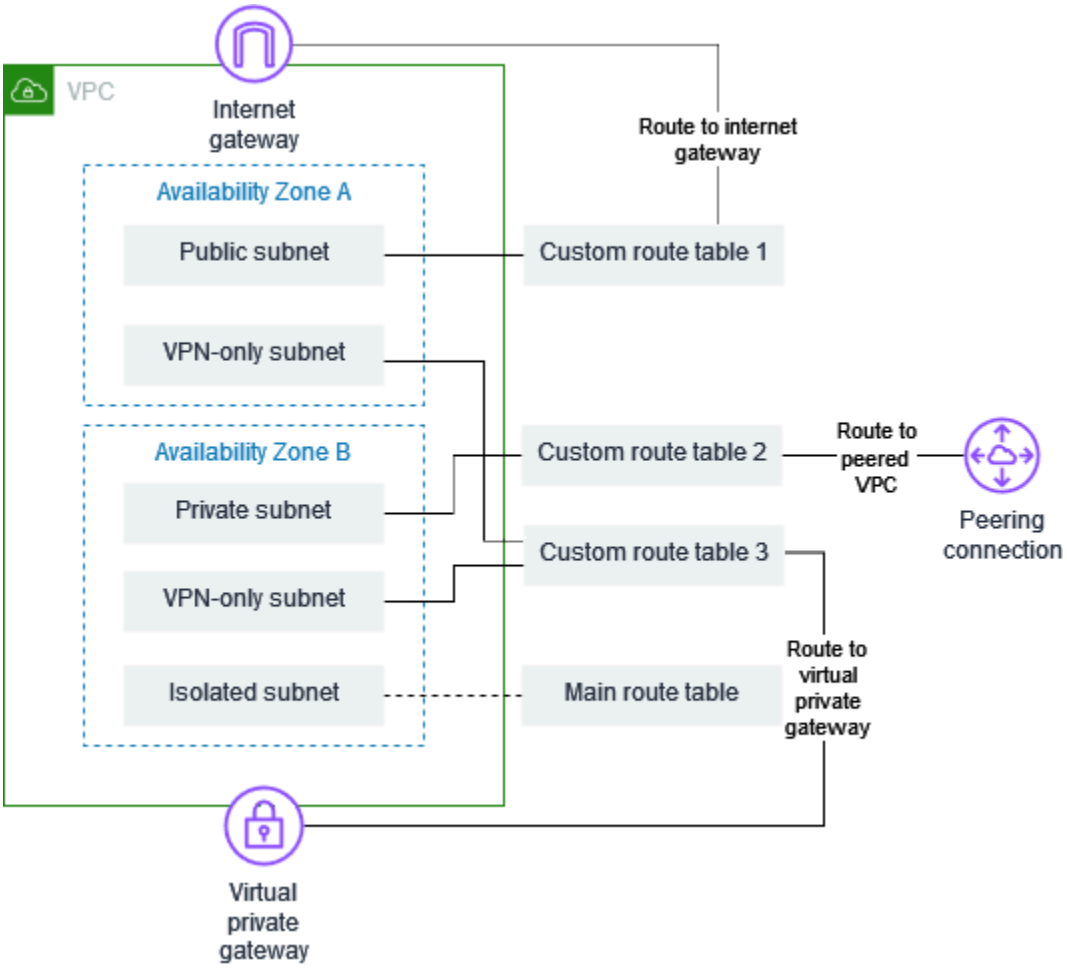
다음은 라우팅 테이블의 핵심 개념입니다.

- 기본 라우팅 테이블 — VPC와 함께 자동으로 제공되는 라우팅 테이블입니다. 다른 라우팅 테이블과 명시적으로 연결되지 않은 모든 서브넷의 라우팅을 제어합니다.
- 사용자 지정 라우팅 테이블(Custom route table) - VPC에 대해 생성하는 라우팅 테이블입니다.
- 대상(Destination) - 트래픽을 이동할 대상 IP 주소(대상 CIDR)의 범위입니다. 예를 들어, CIDR `172.16.0.0/12`가 있는 외부 회사 네트워크입니다.
- 대상(Target) - 대상 트래픽을 전송할 때 사용할 게이트웨이, 네트워크 인터페이스 또는 연결입니다(예: 인터넷 게이트웨이).
- 로컬 라우팅 — VPC 내 통신을 위한 기본 라우팅입니다. VPC에 IPv4 및 IPV6 주소가 모두 있는 경우 IPv4에 대한 로컬 라우팅과 IPv6에 대한 로컬 라우팅이 있습니다.

- 라우팅 테이블 연결(Route table association) - 라우팅 테이블과 서브넷, 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이 간의 연결입니다.
- 서브넷 라우팅 테이블(Subnet route table) - 서브넷과 연결된 라우팅 테이블입니다.
- 전파 - 가상 프라이빗 게이트웨이를 VPC에 연결하고 라우팅 전파를 활성화한 경우 VPN 연결을 위한 경로를 서브넷 라우팅 테이블에 자동으로 추가합니다. 따라서 VPN 경로를 수동으로 추가하거나 제거할 필요가 없습니다. 자세한 내용은 Site-to-Site VPN 사용 설명서의 Site-to-Site VPN 라우팅 옵션을 참조하세요.
- 게이트웨이 라우팅 테이블(Gateway route table) - 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결된 라우팅 테이블입니다.
- 엣지 연결(Edge association) - 인바운드 VPC 트래픽을 어플라이언스로 라우팅하는 데 사용하는 라우팅 테이블입니다. 라우팅 테이블을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결하고 어플라이언스의 네트워크 인터페이스를 VPC 트래픽의 대상으로 지정합니다.
- Transit 게이트웨이 라우팅 테이블(Transit gateway route table) - Transit 게이트웨이와 연결된 라우팅 테이블입니다. 자세한 내용은 Amazon VPC Transit Gateways의 Transit Gateway 라우팅 테이블을 참조하세요.
- 로컬 게이트웨이 라우팅 테이블(Local gateway route table) - Outposts 로컬 게이트웨이와 연결된 라우팅 테이블입니다. 자세한 내용은 AWS Outposts 사용 설명서의 로컬 게이트웨이를 참조하십시오.

라우팅 테이블이 있는 VPC 예

다음 다이어그램은 서브넷 5개, 기본 라우팅 테이블 1개, 사용자 지정 라우팅 테이블 3개가 포함된 VPC를 보여줍니다. 라우팅 테이블 4개에 모두 로컬 라우팅이 있습니다. 사용자 지정 라우팅 테이블 1에는 인터넷 게이트웨이에 대한 라우팅이 있으며 가용 영역 A의 퍼블릭 서브넷과 연결됩니다. 사용자 지정 라우팅 테이블 2에는 피어링된 VPC에 대한 라우팅이 있으며 가용 영역 B의 프라이빗 서브넷과 연결됩니다. 사용자 지정 라우팅 테이블 3에는 가상 프라이빗 게이트웨이에 대한 라우팅이 있으며 두 가용 영역의 VPN 전용 서브넷과 연결됩니다.



서브넷 라우팅 테이블

VPC에는 암시적 라우터가 있으며 라우팅 테이블을 사용하여 네트워크 트래픽이 전달되는 위치를 제어합니다. VPC의 각 서브넷을 라우팅 테이블에 연결해야 합니다. 테이블에서는 서브넷에 대한 라우팅을 제어합니다(서브넷 라우팅 테이블). 서브넷을 특정 라우팅 테이블과 명시적으로 연결할 수 있습니다. 그렇지 않으면 서브넷이 기본 라우팅 테이블과 암시적으로 연결됩니다. 서브넷은 한 번에 하나의 라우팅 테이블에만 연결할 수 있지만 여러 서브넷을 동일한 서브넷 라우팅 테이블에 연결할 수 있습니다.

내용

- Routes
- 기본 라우팅 테이블
- 사용자 지정 라우팅 테이블
- 서브넷 라우팅 테이블 연결

Routes

테이블의 각 라우팅은 목적지 및 대상을 지정합니다. 예를 들어 서브넷이 인터넷 게이트웨이를 통해 인터넷에 액세스할 수 있도록 하려면 서브넷 라우팅 테이블에 다음 라우팅을 추가합니다. 라우팅의 대상은 모든 IPv4 주소를 나타내는 `0.0.0.0/0`입니다. 대상은 VPC에 연결된 인터넷 게이트웨이입니다.

대상 주소	대상
0.0.0.0/0	*igw-id*

IPv4 및 IPv6 CIDR 블록은 별도로 취급됩니다. 예를 들어 대상 CIDR이 `0.0.0.0/0`인 라우팅에는 모든 IPv6 주소가 자동으로 포함되지 않습니다. 모든 IPv6 주소에 대해 대상 CIDR이 `::/0`인 라우팅을 생성해야 합니다.

AWS 리소스 전체에서 동일한 CIDR 블록 세트를 자주 참조하는 경우, 고객 관리형 접두사 목록을 생성하여 함께 그룹화할 수 있습니다. 그런 다음 라우팅 테이블 항목의 대상으로 접두사 목록을 지정할 수 있습니다.

모든 라우팅 테이블에는 VPC 내부 통신을 위한 로컬 라우팅이 포함되어 있습니다. 이 라우팅은 기본적으로 모든 라우팅 테이블에 추가됩니다. 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. 필요에 따라 각 로컬 라우팅의 대상을 교체하거나 복원할 수 있습니다.

규칙 및 고려 사항

- 로컬 경로보다 더 구체적인 경로를 라우팅 테이블에 추가할 수 있습니다. 대상은 VPC에 있는 서브넷의 전체 IPv4 또는 IPv6 CIDR 블록과 일치해야 합니다. 대상은 NAT 게이트웨이, 네트워크 인터페이스 또는 게이트웨이 Load Balancer 엔드포인트여야 합니다.
- 라우팅 테이블에 라우팅이 여러 개 있는 경우 트래픽과 일치하는 가장 구체적인 라우팅(가장 긴 접두사 일치)을 사용하여 트래픽의 라우팅 방법을 결정합니다.
- 정확히 일치하거나 169.254.168.0/22 범위의 하위 세트인 IPv4 주소에는 경로를 추가할 수 없습니다. 이 범위는 링크 로컬 주소 공간 내에 있으며 AWS 서비스에서 사용하도록 예약되어 있습니다. 예를 들어 Amazon EC2는 인스턴스 메타데이터 서비스(IMDS) 및 Amazon DNS 서버와 같이 EC2 인스턴스에서만 액세스할 수 있는 서비스에 대해 이 범위의 주소를 사용합니다. 169.254.168.0/22보다 크지만 겹치는 CIDR 블록을 사용할 수 있지만 169.254.168.0/22의 주소로 향하는 패킷은 전달되지 않습니다.

- 정확히 일치하거나 fd00:ec2::/32 범위의 하위 세트인 IPv6 주소에는 경로를 추가할 수 없습니다. 이 범위는 고유 로컬 주소(ULA) 공간 내에 있으며 AWS 서비스에서 사용하도록 예약되어 있습니다. 예를 들어 Amazon EC2는 인스턴스 메타데이터 서비스(IMDS) 및 Amazon DNS 서버와 같이 EC2 인스턴스에서만 액세스할 수 있는 서비스에 대해 이 범위의 주소를 사용합니다. fd00:ec2::/32보다 크지만 겹치는 CIDR 블록을 사용할 수 있지만 fd00:ec2::/32의 주소로 향하는 패킷은 전달되지 않습니다.
- VPC에 대한 라우팅 경로에 미들박스 어플라이언스를 추가할 수 있습니다. 자세한 내용은 the section called “미들박스 어플라이언스에 대한 라우팅” 섹션을 참조하세요.

예제

다음 예제에서 VPC에는 IPv4 CIDR 블록 및 IPv6 CIDR 블록이 모두 있다고 가정합니다. IPv4 및 IPv6 트래픽은 다음 라우팅 테이블에 표시된 것처럼 별도로 처리됩니다.

대상 주소	대상
10.0.0.0/16	Local
2001:db8:1234:1a00::/56	Local
172.31.0.0/16	pcx-11223344556677889
0.0.0.0/0	igw-12345678901234567
::/0	eigw-aabbccddee1122334

- VPC(10.0.0.0/16) 내에서 라우팅되는 IPv4 트래픽은 Local 경로로 처리됩니다.
- VPC(2001:db8:1234:1a00::/56) 내에서 라우팅되는 IPv6 트래픽은 Local 경로로 처리됩니다.
- 172.31.0.0/16의 경로는 피어링 연결로 트래픽을 전송합니다.
- 모든 IPv4 트래픽(0.0.0.0/0)의 경로는 인터넷 게이트웨이로 트래픽을 전송합니다. 따라서 VPC 내 그리고 피어링 연결을 통한 트래픽을 제외한 모든 IPv4 트래픽은 인터넷 게이트웨이로 라우팅됩니다.
- 모든 IPv6 트래픽(::/0)의 경로는 외부 전용 인터넷 게이트웨이로 트래픽을 전송합니다. 따라서 VPC 내 트래픽을 제외한 모든 IPv6 트래픽은 외부 전용 인터넷 게이트웨이로 라우팅됩니다.

기본 라우팅 테이블

VPC를 만들면 기본 라우팅 테이블이 자동으로 생성됩니다. 서브넷이 라우팅 테이블과 명시적으로 연결되지 않은 경우 서브넷은 기본 라우팅 테이블이 기본적으로 사용됩니다. Amazon VPC 콘솔에서 라우팅 테이블(Route tables) 페이지의 기본(Main) 열에서 예(Yes)를 찾으면 VPC에 대한 기본 라우팅 테이블을 볼 수 있습니다.

기본적으로 기본이 아닌 VPC를 만들면 기본 라우팅 테이블에는 로컬 라우팅만 포함됩니다. 만약NAT 게이트웨이를 VPC 생성 및 선택하면 Amazon VPC가 게이트웨이의 기본 라우팅 테이블에 경로를 자동으로 추가합니다.

기본 라우팅 테이블에는 다음 규칙이 적용됩니다.

- 기본 라우팅 테이블에서 라우팅을 추가 및 제거하고 수정할 수 있습니다.
- 기본 라우팅 테이블은 삭제할 수 없습니다.
- 게이트웨이 라우팅 테이블은 기본 라우팅 테이블로 설정할 수 없습니다.
- 사용자 지정 라우팅 테이블을 서브넷에 연결하여 기본 라우팅 테이블을 대체할 수 있습니다.
- 서브넷이 기본 라우팅 테이블에 명시적으로 연결되어 있지 않을 경우에도 서브넷을 기본 라우팅에 명시적으로 연결할 수 있습니다.

  기본 라우팅 테이블을 변경하면 이를 수행할 수 있습니다. 기본 라우팅 테이블을 변경하면 추가되는 새로운 서브넷 또는 다른 라우팅 테이블에 명시적으로 연결되지 않은 서브넷의 기본값도 변경됩니다. 자세한 내용은 기본 라우팅 테이블 교체 단원을 참조하세요.

사용자 지정 라우팅 테이블

기본적으로 라우팅 테이블에는 VPC 내부 통신을 위한 로컬 라우팅이 포함되어 있습니다. 만약 퍼블릭 서브넷을 VPC 생성 및 선택하면 Amazon VPC는 사용자 지정 라우팅 테이블을 만들고 인터넷 게이트웨이를 가리키는 경로를 추가합니다. VPC를 보호하는 한 가지 방법은 기본 라우팅 테이블을 원래 기본 상태로 두는 것입니다. 그런 다음 생성한 각 새 서브넷을 생성한 사용자 지정 라우팅 테이블 중 하나에 명시적으로 연결합니다. 이렇게 하면 각 서브넷이 트래픽으로 어떻게 라우팅되는지를 명시적으로 제어할 수 있습니다.

기본 라우팅 테이블에서 라우팅을 추가 및 제거하고 수정할 수 있습니다. 사용자 지정 라우팅 테이블에 연결이 없는 경우에만 삭제할 수 있습니다.

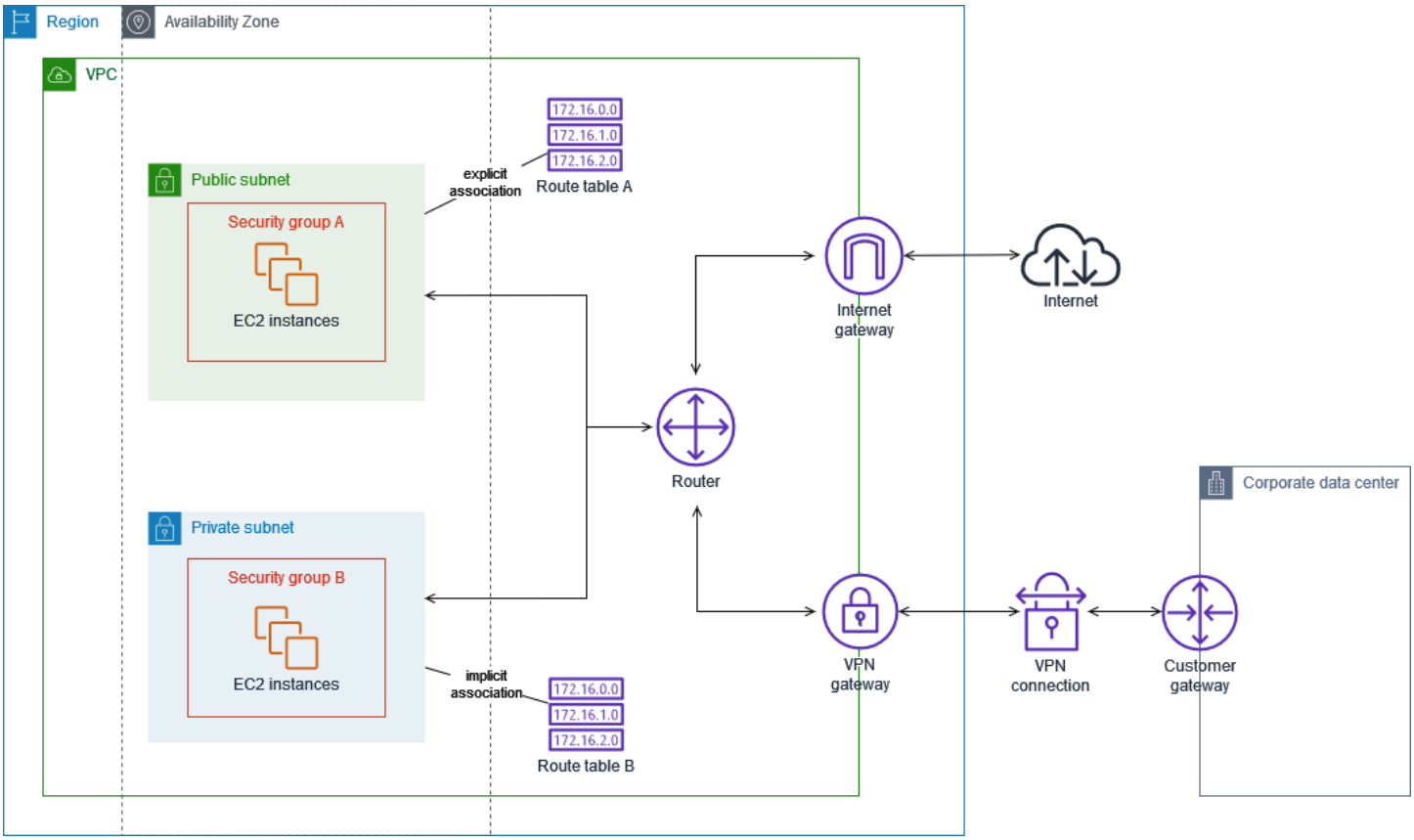
서브넷 라우팅 테이블 연결

VPC에 있는 각 서브넷을 라우팅 테이블과 연결해야 합니다. 서브넷은 사용자 지정 라우팅 테이블과 명시적으로 연결되거나 기본 라우팅 테이블과 암시적 또는 명시적으로 연결될 수 있습니다. 서브넷 및 라우팅 테이블 연결 보기에 대한 자세한 내용은 명시적 연결 확인를 참조하세요.

Outposts와 연결된 VPC에 있는 서브넷은 로컬 게이트웨이의 추가 대상 유형을 가질 수 있습니다. 이 점이 Outposts가 아닌 서브넷과의 유일한 라우팅 차이점입니다.

예제 1: 암시적/명시적 서브넷 연결

다음 다이어그램에서는 인터넷 게이트웨이, 가상 프라이빗 게이트웨이, 퍼블릭 서브넷 및 VPN 전용 서브넷이 있는 VPC의 라우팅을 보여줍니다.



라우팅 테이블 A는 퍼블릭 서브넷과 명시적으로 연결되는 사용자 지정 라우팅 테이블입니다. 이 서브넷은 모든 트래픽을 인터넷 게이트웨이로 보내는 경로가 있기 때문에 퍼블릭 서브넷이 됩니다.

대상 주소	대상
VPC CIDR	로컬

대상 주소	대상
0.0.0.0/0	*igw-id*

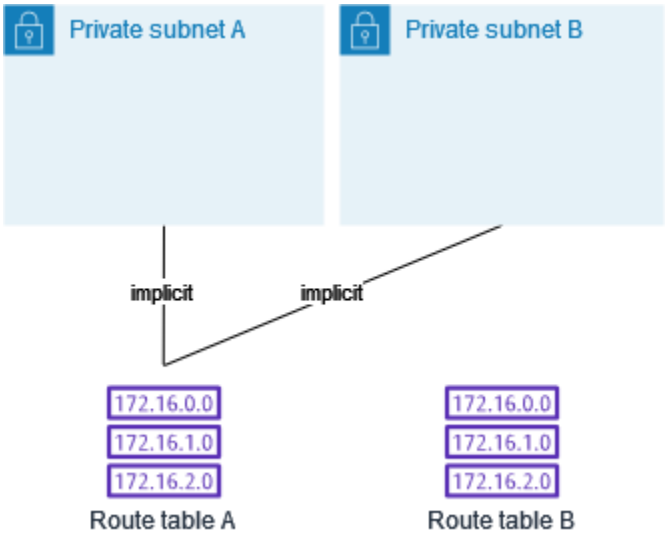
라우팅 테이블 B는 기본 라우팅 테이블입니다. 프라이빗 서브넷과는 묵시적으로 연결됩니다. 이 서브넷은 모든 트래픽을 가상 프라이빗 게이트웨이로 보내는 경로는 있지만 인터넷 게이트웨이로 보내는 경로는 없기 때문에 VPN 전용 서브넷이 됩니다. 이 VPC에 다른 서브넷을 만들고 사용자 지정 라우팅 테이블에 연결하지 않으면, 서브넷은 기본 라우팅 테이블인 이 라우팅 테이블과도 묵시적으로 연결됩니다.

대상 주소	대상
VPC CIDR	로컬
0.0.0.0/0	*vgw-id*

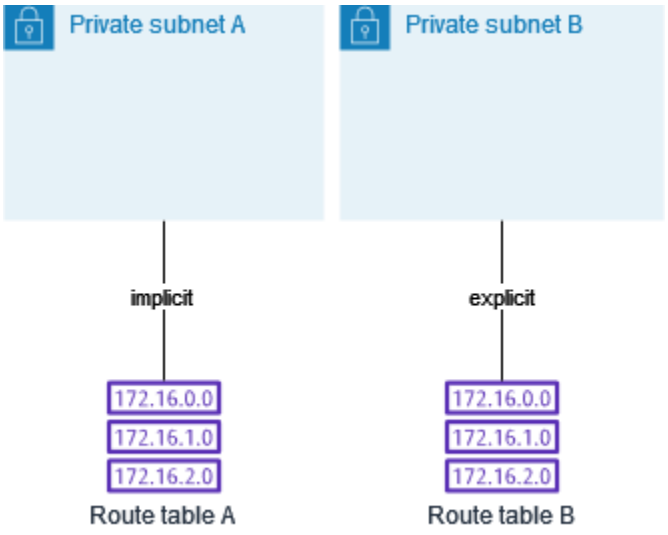
예제 2: 기본 라우팅 테이블 바꾸기

기본 라우팅 테이블을 변경할 수 있습니다. 트래픽 중단을 방지하려면 먼저 사용자 지정 라우팅 테이블을 사용하여 라우팅 변경을 테스트하는 것이 좋습니다. 테스트 결과에 만족하면 기본 라우팅 테이블을 새로운 사용자 지정 테이블로 바꿉니다.

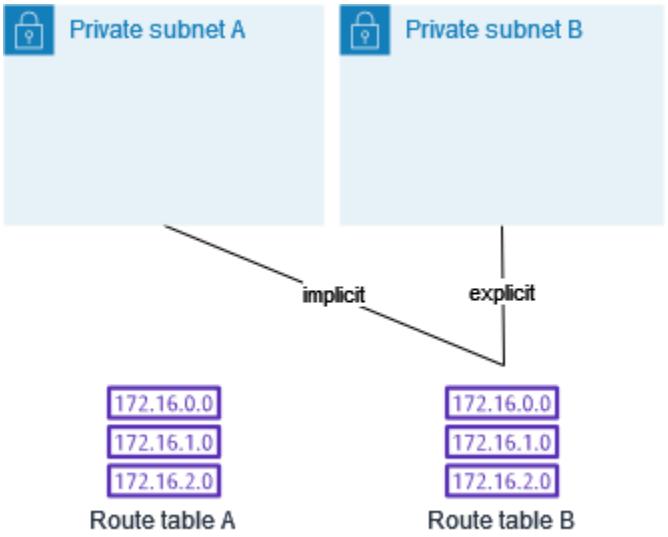
다음 구성도는 서브넷 2개와 라우팅 테이블 2개를 보여 줍니다. 서브넷 A는 기본 라우팅 테이블인 라우팅 테이블 A와 묵시적으로 연결되어 있습니다. 서브넷 B는 라우팅 테이블 A와 묵시적으로 연결되어 있습니다. 사용자 지정 라우팅 테이블인 라우팅 테이블 B는 어느 서브넷과도 연결되어 있지 않습니다.



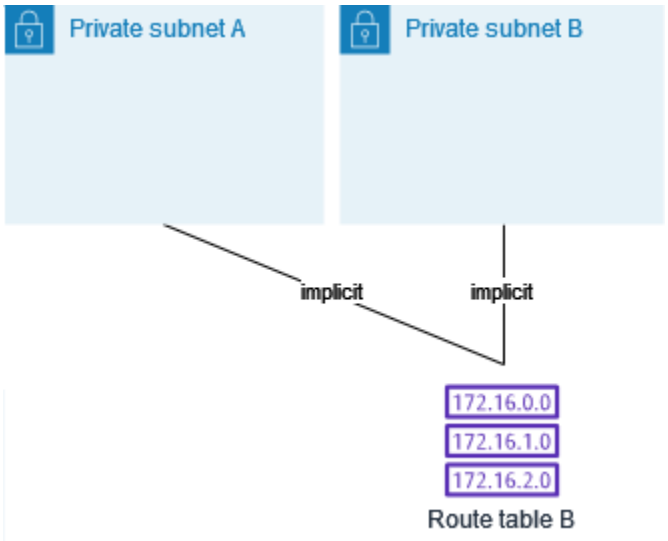
기본 라우팅 테이블을 교체하려면 먼저 서브넷 B와 라우팅 테이블 B 간에 명시적 연결을 만듭니다. 라우팅 테이블 B를 테스트합니다.



라우팅 테이블 B를 테스트한 후, 이 테이블을 기본 라우팅 테이블로 만듭니다. 서브넷 B는 여전히 라우팅 테이블 B와 명시적으로 연결되어 있습니다. 단, 라우팅 테이블 B가 이제 새로운 기본 라우팅 테이블이기 때문에 서브넷 A는 라우팅 테이블 B와 묵시적으로 연결됩니다. 라우팅 테이블 A는 더 이상 어느 서브넷과도 연결되지 않습니다.



(선택) 서브넷 B를 라우팅 테이블 B와의 연결에서 해제해도 서브넷 B와 라우팅 테이블 B의 묵시적 연결은 유지됩니다. 라우팅 테이블 A가 더 이상 필요하지 않은 경우 삭제할 수 있습니다.



게이트웨이 라우팅 테이블

라우팅 테이블을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결할 수 있습니다. 라우팅 테이블이 게이트웨이와 연결된 경우 이를 게이트웨이 라우팅 테이블이라고 합니다. VPC로 들어오는 트래픽의 라우팅 경로를 세밀하게 제어할 수 있는 게이트웨이 라우팅 테이블을 생성할 수 있습니다. 예를 들어 인터넷 게이트웨이를 통해 VPC로 들어오는 트래픽을 VPC의 미들박스 어플라이언스(예: 보안 어플라이언스)로 리디렉션하여 가로챌 수 있습니다.

내용

- 게이트웨이 라우팅 테이블 라우팅
- 규칙 및 고려 사항

게이트웨이 라우팅 테이블 라우팅

인터넷 게이트웨이와 연결된 게이트웨이 라우팅 테이블은 다음 대상을 가진 경로를 지원합니다.

- 기본 로컬 경로
- Gateway Load Balancer 엔드포인트
- 미들박스 어플라이언스에 대한 네트워크 인터페이스

가상 프라이빗 게이트웨이와 연결된 게이트웨이 라우팅 테이블은 다음 대상을 가진 경로를 지원합니다.

- 기본 로컬 경로
- Gateway Load Balancer 엔드포인트
- 미들박스 어플라이언스에 대한 네트워크 인터페이스

대상이 Gateway Load Balancer 엔드포인트 또는 네트워크 인터페이스인 경우 다음 대상이 허용됩니다.

- VPC의 전체 IPv4 또는 IPv6 CIDR 블록. 이 경우 기본 로컬 라우팅의 대상을 대체합니다.
- VPC에 있는 서브넷의 전체 IPv4 또는 IPv6 CIDR 블록입니다. 이는 기본 로컬 라우팅보다 더 구체적인 라우팅입니다.

게이트웨이 라우팅 테이블에 있는 로컬 라우팅의 대상을 VPC의 네트워크 인터페이스로 변경하면 나중에 기본 `local` 대상으로 복원할 수 있습니다. 자세한 내용은 로컬 경로의 대상 교체 또는 복원 섹션을 참조하세요.

예제

다음 게이트웨이 라우팅 테이블에서 `172.31.0.0/20` CIDR 블록이 있는 서브넷으로 향하는 트래픽은 특정 네트워크 인터페이스로 라우팅됩니다. VPC의 다른 모든 서브넷으로 향하는 트래픽은 로컬 라우팅을 사용합니다.

대상 주소	대상
172.31.0.0/16	로컬
172.31.0.0/20	*eni-id*

예제

다음 게이트웨이 라우팅 테이블에서 로컬 라우팅의 대상이 네트워크 인터페이스 ID로 대체됩니다. VPC 내의 모든 서브넷으로 향하는 트래픽은 네트워크 인터페이스로 라우팅됩니다.

대상 주소	대상
172.31.0.0/16	*eni-id*

규칙 및 고려 사항

다음 중 하나라도 해당되는 경우 라우팅 테이블을 게이트웨이와 연결할 수 없습니다.

- 라우팅 테이블에 네트워크 인터페이스, Gateway Load Balancer 엔드포인트 또는 기본 로컬 라우팅이 아닌 대상이 있는 기존 라우팅이 포함된 경우
- 라우팅 테이블에 VPC의 범위를 벗어나는 CIDR 블록에 대한 기존 라우팅이 포함된 경우
- 라우팅 테이블에 대해 라우팅 전파가 활성화된 경우

또한 다음 규칙 및 고려 사항이 적용됩니다.

- 개별 VPC CIDR 블록보다 큰 범위를 포함하여 VPC의 범위를 벗어나는 CIDR 블록에는 라우팅을 추가할 수 없습니다.
- `local`, Gateway Load Balancer 엔드포인트 또는 네트워크 인터페이스를 대상으로 지정할 수만 있습니다. 개별 호스트 IP 주소 등 다른 유형의 대상은 지정할 수 없습니다. 자세한 내용은 the section called “라우팅 옵션 예” 섹션을 참조하세요.
- 접두사 목록을 대상으로 지정할 수 없습니다.
- 게이트웨이 라우팅 테이블을 사용하여 VPC 외부의 트래픽(예: 연결된 전송 게이트웨이를 통한 트래픽)을 제어하거나 가로챌 수 없습니다. VPC로 들어오는 트래픽을 가로채서 동일한 VPC에 있는 다른 대상으로만 리디렉션할 수 있습니다.

- 트래픽이 미들박스 어플라이언스에 도달하도록 하려면 실행 중인 인스턴스에 대상 네트워크 인터페이스가 연결되어야 합니다. 인터넷 게이트웨이를 통해 흐르는 트래픽의 경우, 대상 네트워크 인터페이스에는 퍼블릭 IP 주소도 있어야 합니다.
- 미들박스 어플라이언스를 구성할 때 어플라이언스 고려 사항을 기록해 둡니다.
- 미들박스 어플라이언스를 통해 트래픽을 라우팅하는 경우 대상 서브넷의 반환 트래픽은 동일한 어플라이언스를 통해 라우팅되어야 합니다. 비대칭 라우팅은 지원되지 않습니다.
- 라우팅 테이블 규칙은 서브넷을 떠나는 모든 트래픽에 적용됩니다. 서브넷을 떠나는 트래픽은 해당 서브넷의 게이트웨이 라우터의 MAC 주소를 대상으로 하는 트래픽으로 정의됩니다. 서브넷에 있는 다른 네트워크 인터페이스의 MAC 주소를 대상으로 하는 트래픽은 네트워크(계층 3) 대신 데이터 링크(계층 2) 라우팅을 사용하므로 규칙이 이 트래픽에 적용되지 않습니다.
- 모든 로컬 영역이 가상 프라이빗 게이트웨이와의 엣지 연결을 지원하는 것은 아닙니다. 사용 가능한 영역에 대한 자세한 내용은 AWS 로컬 영역 사용 설명서의 고려 사항을 참조하세요.

라우팅 우선 순위 작동 방식

일반적으로 트래픽은 트래픽과 일치하는 가장 구체적인 경로를 사용하여 전달됩니다. 이를 가장 긴 접두사 일치라고 합니다. 라우팅 테이블에 겹치거나 일치하는 경로가 있는 경우 추가적인 규칙이 적용됩니다.

다음 목록은 더 자세한 정보 및 예와 함께 아래 섹션으로 연결되는 링크가 있는 경로 우선순위 요약을 보여줍니다.

1. 가장 긴 접두사(예: 10.10.2.15/32는 10.10.2.0/24보다 우선순위가 높음)
2. 정적 경로(예: VPC 피어링 및 인터넷 게이트웨이 연결)
3. 접두사 목록 경로
4. 전파된 경로
   a. Direct Connect BGP 경로(동적 경로)
   b. VPN 정적 경로
   c. VPN BGP 경로(동적 경로)(예: 가상 프라이빗 게이트웨이)

가장 긴 접두사 일치

IPv4 및 IPv6 주소 또는 CIDR 블록에 대한 라우팅은 서로 독립적입니다. IPv4 트래픽 또는 IPv6 트래픽과 일치하는 가장 구체적인 라우팅을 사용하여 트래픽 라우팅 방법을 결정합니다.

다음 예의 서브넷 라우팅 테이블에는 인터넷 게이트웨이를 가리키는 IPv4 인터넷 트래픽(0.0.0.0/0)에 대한 경로와 피어링 연결(172.31.0.0/16)을 가리키는 pcx-11223344556677889 IPv4 트래픽에 대한 경로가 있습니다. 172.31.0.0/16 IP 주소 범위로 향하는 서브넷의 모든 트래픽은 피어링 연결을 사용합니다. 이 라우팅은 인터넷 게이트웨이에 대한 라우팅보다 더 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-11223344556677889
0.0.0.0/0	igw-12345678901234567

정적 및 동적으로 전파된 경로의 경로 우선순위

가상 프라이빗 게이트웨이를 VPC에 연결하고 서브넷 라우팅 테이블에서 라우팅 전파를 활성화한 경우, Site-to-Site VPN 연결을 나타내는 라우팅은 라우팅 테이블에 전파된 라우팅으로 자동으로 나타납니다.

전파된 경로의 대상이 정적 경로의 대상과 동일한 경우 정적 경로의 우선순위가 높습니다. 다음과 같은 리소스에서 정적 경로를 사용합니다.

- 인터넷 게이트웨이
- NAT 게이트웨이
- 네트워크 인터페이스
- 인스턴스 ID
- 게이트웨이 VPC 엔드포인트
- 전송 게이트웨이
- VPC 피어링 연결
- Gateway Load Balancer 엔드포인트

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 라우팅 테이블 및 VPN 경로 우선 순위를 참조하세요.

예를 들어 다음 라우팅 테이블에는 인터넷 게이트웨이에 대한 정적 경로와 가상 프라이빗 게이트웨이에 대한 전파 경로가 있습니다. 두 라우팅은 모두 대상 주소가 `172.31.0.0/24`입니다. 인터넷 게이트웨이에 대한 정적 경로가 우선 적용되므로 대상이 `172.31.0.0/24`인 모든 트래픽이 인터넷 게이트웨이로 라우팅됩니다.

대상 주소	대상	전파
10.0.0.0/16	로컬	아니요
172.31.0.0/24	vgw-11223344556677889	예
172.31.0.0/24	igw-12345678901234567	아니요

접두사 목록의 라우팅 우선 순위

라우팅 테이블이 접두사 목록을 참조하는 경우 다음 규칙이 적용됩니다.

- 라우팅 테이블에 접두사 목록을 참조하는 라우팅과 일치하는 전파된 라우팅이 포함된 경우 접두사 목록을 참조하는 라우팅이 먼저 적용됩니다. 겹치는 경로의 경우 전파된 경로, 정적 경로 또는 접두사 목록을 참조하는 경로인지와 관계없이 보다 구체적인 경로가 항상 우선합니다.
- 라우팅 테이블이 서로 다른 대상에 겹치는 CIDR 블록이 있는 여러 접두사 목록을 참조하는 경우 우선 적용할 라우팅을 임의로 선택합니다. 그 이후에는 동일한 라우팅이 항상 우선합니다.

라우팅 옵션 예

다음 주제에서는 VPC의 특정 게이트웨이 또는 연결을 위한 라우팅을 설명합니다.

내용

- 인터넷 게이트웨이로 라우팅
- NAT 디바이스로 라우팅
- 가상 프라이빗 게이트웨이로 라우팅
- AWS Outposts 로컬 게이트웨이로 라우팅
- VPC 피어링 연결로 라우팅
- 게이트웨이 VPC 엔드포인트로 라우팅
- 외부 전용 인터넷 게이트웨이로 라우팅

- 전송 게이트웨이에 대한 라우팅
- 미들박스 어플라이언스에 대한 라우팅
- 접두사 목록을 사용한 라우팅
- Gateway Load Balancer 엔드포인트로의 라우팅

인터넷 게이트웨이로 라우팅

서브넷 라우팅 테이블의 라우팅을 인터넷 게이트웨이에 추가하여 서브넷을 퍼블릭 서브넷으로 만들 수 있습니다. 이렇게 하려면 인터넷 게이트웨이를 생성하여 VPC에 연결한 다음, IPv4 트래픽에 대한 대상 주소가 `0.0.0.0/0`인 라우팅이나 IPv6 트래픽에 대한 대상 주소가 `::/0`인 라우팅을 추가하고, 인터넷 게이트웨이 ID(`igw-xxxxxxxxxxxxxxxxx`)의 대상을 추가합니다.

대상 주소	대상
0.0.0.0/0	*igw-id*
::/0	*igw-id*

자세한 내용은 인터넷 게이트웨이를 사용하여 VPC에 대한 인터넷 액세스 활성화 단원을 참조하세요.

NAT 디바이스로 라우팅

프라이빗 서브넷의 인스턴스가 인터넷에 연결되도록 하려면 NAT 게이트웨이를 생성하거나 퍼블릭 서브넷에서 NAT 인스턴스를 시작할 수 있습니다. 그런 다음 IPv4 인터넷 트래픽(`0.0.0.0/0`)을 NAT 디바이스로 라우팅하는 프라이빗 서브넷의 라우팅 테이블에 대한 라우팅을 추가합니다.

대상 주소	대상
0.0.0.0/0	*nat-gateway-id*

또한 NAT 게이트웨이 사용에 대한 불필요한 데이터 처리 비용을 피하거나 특정 트래픽을 비공개로 라우팅하기 위해 다른 대상에 대한 보다 구체적인 라우팅을 생성할 수 있습니다. 다음 예에서 Amazon S3 트래픽(pl-xxxxxxxx, 특정 리전의 Amazon S3에 대한 IP 주소 범위가 포함된 접두사 목록)은 게이트웨이 VPC 엔드포인트로 라우팅되고 10.25.0.0/16 트래픽은 VPC 피어링 연결로 라우팅됩니다. 이러한 IP 주소 범위는 0.0.0.0/0보다 더 구체적입니다. 인스턴스가 Amazon S3 또는 피어 VPC로 트래픽을 보

내면 트래픽이 게이트웨이 VPC 엔드포인트 또는 VPC 피어링 연결로 전송됩니다. 다른 모든 트래픽은 NAT 게이트웨이로 전송됩니다.

대상 주소	대상
0.0.0.0/0	*nat-gateway-id*
pl-*xxxxxxxx*	*vpce-id*
10.25.0.0/16	*pcx-id*

자세한 내용은 NAT 디바이스 섹션을 참조하세요.

가상 프라이빗 게이트웨이로 라우팅

AWS Site-to-Site VPN 연결을 사용하여 VPC의 인스턴스를 사용자의 네트워크와 통신하도록 할 수 있습니다. 이렇게 하려면 가상 프라이빗 게이트웨이를 생성하여 VPC에 연결합니다. 그런 다음 네트워크 대상 및 가상 프라이빗 게이트웨이(vgw-xxxxxxxxxxxxxxxxx)의 대상이 있는 서브넷 라우팅 테이블에 라우팅을 추가합니다.

대상 주소	대상
10.0.0.0/16	*vgw-id*

이제 Site-to-Site VPN 연결을 만들고 구성할 수 있습니다. 자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 AWS Site-to-Site VPN이란 무엇입니까? 및 라우팅 테이블 및 VPN 경로 우선 순위를 참조하세요.

가상 프라이빗 게이트웨이의 Site-to-Site VPN 연결은 IPv6 트래픽을 지원하지 않습니다. 그러나 가상 프라이빗 게이트웨이를 통해 Direct Connect 연결로 라우팅되는 IPv6 트래픽은 지원합니다. 자세한 내용은 Direct Connect 사용 설명서를 참조하세요.

AWS Outposts 로컬 게이트웨이로 라우팅

이 섹션에서는 AWS Outposts 로컬 게이트웨이로 라우팅을 위한 라우팅 테이블 구성에 대해 설명합니다.

내용

- Outpost 서브넷과 온프레미스 네트워크 간 트래픽 활성화
- Outpost 전체에서 동일한 VPC에 있는 서브넷 간 트래픽 활성화

Outpost 서브넷과 온프레미스 네트워크 간 트래픽 활성화

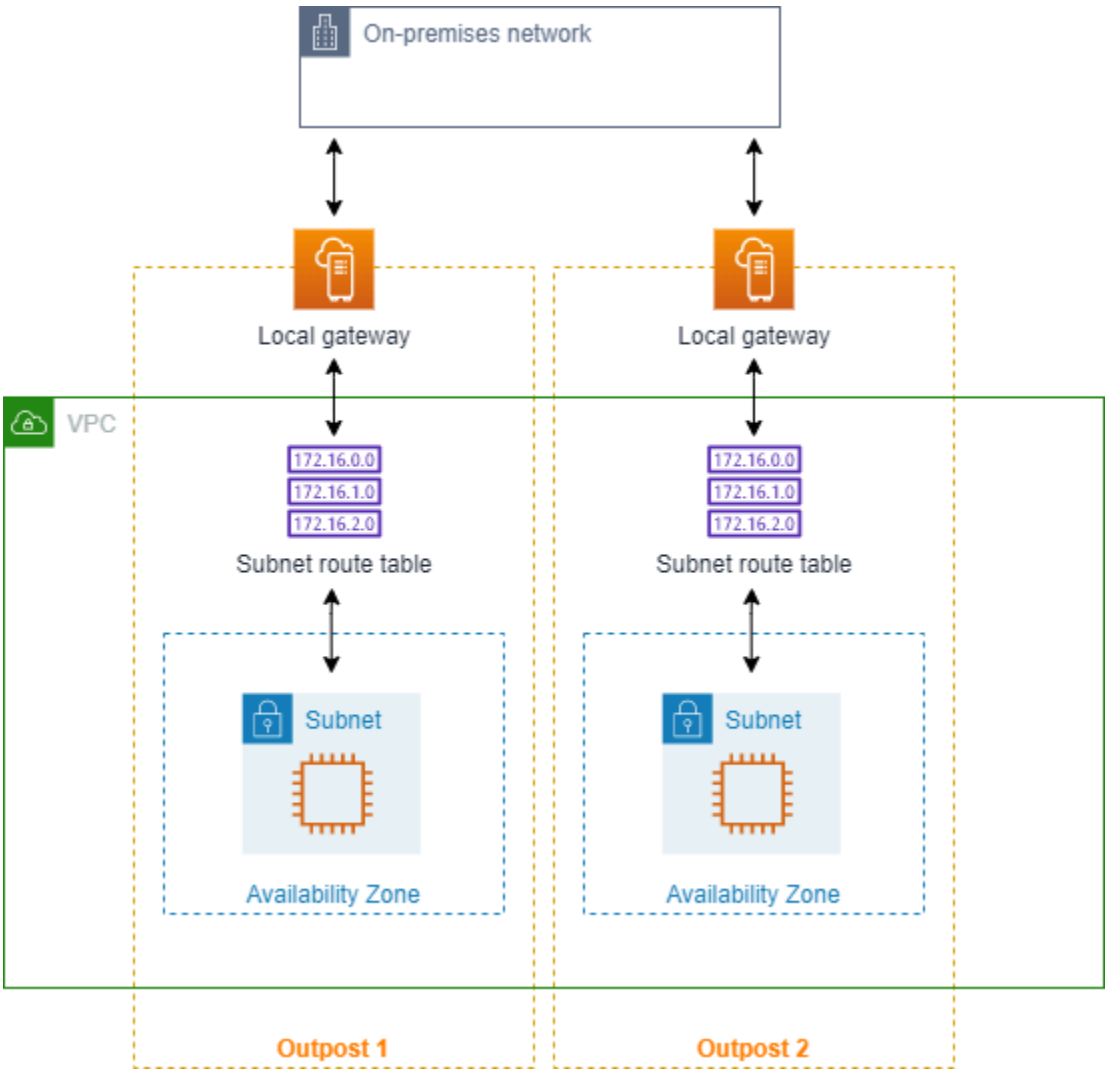
AWS Outposts와 연결된 VPC에 있는 서브넷은 로컬 게이트웨이의 추가 대상 유형을 가질 수 있습니다. 대상 주소가 192.168.10.0/24인 로컬 게이트웨이 트래픽을 고객 네트워크로 라우팅하려는 경우를 생각해 보세요. 이렇게 하려면 대상 네트워크와 로컬 게이트웨이(`lgw-xxxx`)의 대상을 사용하여 다음 라우팅을 추가합니다.

대상 주소	대상
192.168.10.0/24	*lgw-id*

Outpost 전체에서 동일한 VPC에 있는 서브넷 간 트래픽 활성화

Outpost 로컬 게이트웨이와 온프레미스 네트워크를 사용하여 여러 Outpost에서 동일한 VPC에 있는 서브넷 간 통신을 설정할 수 있습니다.

이 기능을 사용하면 서로 다른 가용 영역에 고정된 Outpost 랙 간에 연결을 설정하여 Outpost 랙에서 실행되는 온프레미스 애플리케이션을 위한 다중 가용 영역 아키텍처와 유사한 아키텍처를 구축할 수 있습니다.



이 기능을 활성화하려면 Outpost 랙 서브넷 라우팅 테이블에 해당 라우팅 테이블의 로컬 라우팅보다 더 구체적이고 대상 유형이 로컬 게이트웨이인 경로를 추가합니다. 경로의 대상은 다른 Outpost의 VPC에 있는 서브넷의 전체 IPv4 블록과 일치해야 합니다. 통신해야 하는 모든 Outpost 서브넷에 대해 이 구성을 반복합니다.

Important

- 이 기능을 사용하려면 직접 VPC 라우팅을 사용해야 합니다. 고객 소유의 IP 주소는 사용할 수 없습니다.
- 서브넷이 서로 액세스할 수 있도록 Outpost 로컬 게이트웨이가 연결된 온프레미스 네트워크에 필요한 라우팅이 있어야 합니다.
- 서브넷의 리소스에 보안 그룹을 사용하려면 IP 주소 범위를 Outpost 서브넷의 소스 또는 대상으로 포함하는 규칙을 사용해야 합니다. 보안 그룹 ID는 사용할 수 없습니다.

- 여러 Outpost 간의 VPC 내 통신 지원을 활성화하기 위해 기존 Outpost 랙의 업데이트가 필요할 수 있습니다. 이 기능이 작동하지 않으면 AWS Support에 문의하세요.

Example예제

CIDR이 10.0.0.0/16인 VPC, CIDR이 10.0.1.0/24인 Outpost 1 서브넷 및 CIDR이 10.0.2.0/24인 Outpost 2 서브넷 환경에서 Outpost 1 서브넷의 라우팅 테이블 항목은 다음과 같습니다.

대상 주소	대상
10.0.0.0/16	로컬
10.0.2.0/24	*lgw-1-id*

Outpost 2 서브넷의 라우팅 테이블 항목은 다음과 같습니다.

대상 주소	대상
10.0.0.0/16	로컬
10.0.1.0/24	*lgw-2-id*

VPC 피어링 연결로 라우팅

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 간 트래픽 라우팅을 활성화하려면 VPC 피어링 연결을 가리키는 하나 이상의 서브넷 라우팅 테이블에 라우팅을 추가해야 합니다. 이렇게 하면 피어링 연결에서 다른 VPC의 CIDR 블록 전체 또는 일부에 액세스할 수 있습니다. 마찬가지로, 다른 VPC의 소유자는 트래픽을 다시 사용자의 VPC로 라우팅하기 위해 소유자 자신의 서브넷 라우팅 테이블에 라우팅을 추가해야 합니다.

예를 들어 다음과 같은 정보를 가진 두 VPC 사이에 VPC 피어링 연결(pcx-11223344556677889)이 있다고 합시다.

- VPC A: CIDR 블록은 10.0.0.0/16
- VPC B: 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-11223344556677889

VPC B의 라우팅 테이블은 다음과 같이 구성됩니다.

대상 주소	대상
172.31.0.0/16	로컬
10.0.0.0/16	pcx-11223344556677889

VPC와 인스턴스가 IPv6 통신을 할 수 있는 경우, VPC 피어링 연결은 VPC의 인스턴스 간 IPv6 통신도 지원할 수 있습니다.

VPC 간 IPv6 트래픽 라우팅을 활성화하려면, 상대(피어) VPC의 IPv6 CIDR 블록 전체 또는 일부로 연결되는 VPC 피어링 연결을 라우팅 테이블의 전송 대상(경로)으로 추가하세요.

예를 들어 VPC가 위와 같이 동일한 VPC 피어링 연결(pcx-11223344556677889)을 사용하여 다음과 같은 정보를 갖고 있다고 가정합시다.

- 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 서브넷 라우팅 테이블에 다음 라우팅을 추가합니다.

대상 주소	대상
10.0.0.0/16	로컬
172.31.0.0/16	pcx-11223344556677889
2001:db8:5678:2b00::/56	pcx-11223344556677889

다음 라우팅을 VPC B에 대한 라우팅 테이블에 추가합니다.

대상 주소	대상
172.31.0.0/16	로컬
10.0.0.0/16	pcx-11223344556677889
2001:db8:1234:1a00::/56	pcx-11223344556677889

VPC 피어링 연결에 대한 자세한 내용은 Amazon VPC 사용 설명서를 참조하세요.

게이트웨이 VPC 엔드포인트로 라우팅

게이트웨이 VPC 엔드포인트를 사용하면 VPC와 다른 AWS 서비스 사이에 프라이빗 연결을 생성할 수 있습니다. 게이트웨이 엔드포인트를 생성할 때 VPC에서 게이트웨이 엔드포인트에서 사용하는 서브넷 라우팅 테이블을 지정합니다. 경로는 각각의 라우팅 테이블에 자동으로 추가되며 이때 서비스의 접두사 목록 ID(pl-*xxxxxxxx*)를 지정하는 목적지 및 엔드포인트 ID(vpce-*xxxxxxxxxxxxxxxxx*)를 포함한 대상도 함께 추가됩니다. 엔드포인트 경로를 명시적으로 삭제하거나 수정할 수는 없지만, 엔드포인트에서 사용되는 라우팅 테이블을 변경할 수는 있습니다.

엔드포인트에 대한 라우팅과 AWS 서비스에 대한 경로에 대한 자세한 내용은 게이트웨이 엔드포인트에 대한 라우팅을 참조하세요.

외부 전용 인터넷 게이트웨이로 라우팅

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 대

한 라우팅을 구성하려면 IPv6 인터넷 트래픽(`::/0`)을 외부 전용 인터넷 게이트웨이로 라우팅하는 프라이빗 서브넷 라우팅 테이블에 대한 라우팅을 추가해야 합니다.

대상 주소	대상
::/0	*eigw-id*

자세한 내용은 송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽 활성화 단원을 참조하세요.

전송 게이트웨이에 대한 라우팅

전송 게이트웨이에 VPC를 연결할 때 전송 게이트웨이를 통해 라우팅할 트래픽에 대한 라우팅을 서브넷 라우팅 테이블에 추가해야 합니다.

전송 게이트웨이에 연결된 VPC 3개가 있는 다음 시나리오를 고려하세요. 이 시나리오에서는 모든 연결이 전송 게이트웨이 라우팅 테이블과 연결되어 전송 게이트웨이 라우팅 테이블에 전파됩니다. 따라서 모든 연결은 패킷을 서로 라우팅할 수 있으며 전송 게이트웨이는 단순한 계층 3 IP 허브 역할을 합니다.

예를 들어, 다음과 같은 정보를 가진 두 VPC가 있다고 가정합니다.

- VPC A: 10.1.0.0/16, 연결 ID tgw-attach-11111111111111111
- VPC B: 10.2.0.0/16, 연결 ID tgw-attach-22222222222222222

VPC 간에 트래픽을 활성화하고 전송 게이트웨이에 대한 액세스를 허용하려면 VPC A의 라우팅 테이블을 다음과 같이 구성해야 합니다.

대상 주소	대상
10.1.0.0/16	로컬
10.0.0.0/8	*tgw-id*

다음은 VPC 연결에 대한 전송 게이트웨이 라우팅 테이블 항목의 예입니다.

대상 주소	대상
10.1.0.0/16	tgw-attach-11111111111111111
10.2.0.0/16	tgw-attach-22222222222222222

전송 게이트웨이 라우팅 테이블에 대한 자세한 내용은 Amazon VPC Transit Gateways의 라우팅을 참조하세요.

미들박스 어플라이언스에 대한 라우팅

VPC에 대한 라우팅 경로에 미들박스 어플라이언스를 추가할 수 있습니다. 다음은 몇 가지 가능한 사용 사례입니다.

- 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이를 통해 VPC로 들어오는 트래픽을 VPC의 미들박스 어플라이언스로 전송하여 가로챕니다. 미들박스 라우팅 마법사를 사용하여 AWS가 게이트웨이, 미들박스 및 대상 서브넷에 적합한 라우팅 테이블을 자동으로 구성할 수 있습니다. 자세한 내용은 the section called “미들박스 라우팅 마법사” 섹션을 참조하세요.
- 두 서브넷 간의 트래픽을 미들박스 어플라이언스로 보냅니다. 이렇게 하려면 다른 서브넷의 서브넷 CIDR과 일치하는 서브넷 라우팅 테이블에 대한 경로를 생성하고 Gateway Load Balancer 엔드포인트, NAT 게이트웨이, Network Firewall 엔드포인트 또는 어플라이언스의 네트워크 인터페이스를 대상으로 지정합니다. 또는 서브넷에서 다른 서브넷으로 모든 트래픽을 리디렉션하려면 로컬 경로의 대상을 Gateway Load Balancer 엔드포인트, NAT 게이트웨이 또는 네트워크 인터페이스로 바꿉니다.

필요에 맞게 어플라이언스를 구성할 수 있습니다. 예를 들어 모든 트래픽을 차단하는 보안 어플라이언스 또는 WAN 가속 어플라이언스를 구성할 수 있습니다. 어플라이언스는 VPC의 서브넷에 Amazon EC2 인스턴스로 배포되며 서브넷의 탄력적 네트워크 인터페이스(네트워크 인터페이스)로 표시됩니다.

대상 서브넷 라우팅 테이블에 대해 라우팅 전파를 활성화한 경우 라우팅 우선 순위를 알고 있어야 합니다. 가장 구체적인 라우팅에 대해 우선 순위를 두며 라우팅이 일치하면 전파된 라우팅보다 정적 라우팅에 우순 순위를 둡니다.

라우팅을 검토하여 트래픽이 올바르게 라우팅되고 라우팅 전파를 활성화 또는 비활성화할 경우 의도하지 않은 결과가 없는지 확인합니다(예: 점보 프레임을 지원하는 Direct Connect 연결에 대해 라우팅 전파가 필요함).

인바운드 VPC 트래픽을 어플라이언스로 라우팅하려면 라우팅 테이블을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결하고 어플라이언스의 네트워크 인터페이스를 VPC 트래픽의 대상으로 지정합니다. 자세한 내용은 게이트웨이 라우팅 테이블 섹션을 참조하세요. 서브넷에서 다른 서브넷의 미들박스 어플라이언스로 아웃바운드 트래픽을 라우팅할 수도 있습니다.

미들박스 라우팅 예제에 대해서는 미들박스 시나리오을(를) 참조하세요.

내용

- 어플라이언스 고려 사항
- 게이트웨이와 어플라이언스 간 트래픽 라우팅
- 어플라이언스로 서브넷 간 트래픽 라우팅

어플라이언스 고려 사항

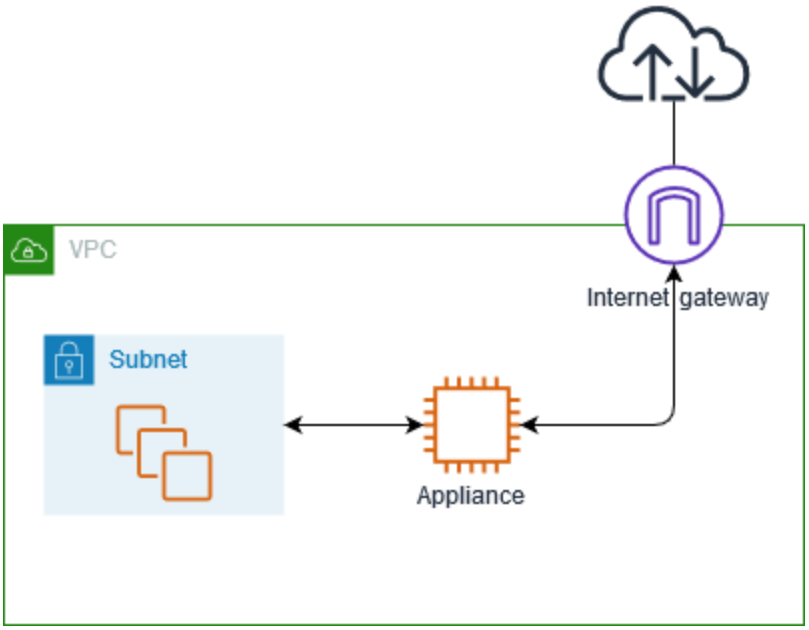
AWS Marketplace에서 서드 파티 어플라이언스를 선택하거나 직접 어플라이언스를 구성할 수 있습니다. 어플라이언스를 생성하거나 구성할 때 다음 사항에 유의하세요.

- 어플라이언스는 원본 또는 대상 트래픽과 별도의 서브넷에 구성되어야 합니다.
- 어플라이언스에서 원본/대상 확인을 비활성화해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 소스 또는 대상 변경 확인을 참조하세요.
- 어플라이언스를 통해 동일한 서브넷에 있는 호스트 간에 트래픽을 라우팅할 수 없습니다.
- 어플라이언스는 NAT(네트워크 주소 변환)를 수행할 필요가 없습니다.
- 로컬 경로보다 더 구체적인 경로를 라우팅 테이블에 추가할 수 있습니다. 보다 구체적인 경로를 사용하여 VPC 내의 서브넷 간의 트래픽(동서 트래픽)을 미들박스 어플라이언스로 리디렉션할 수 있습니다. 경로의 대상은 VPC에 있는 서브넷의 전체 IPv4 또는 IPv6 CIDR 블록과 일치해야 합니다.
- IPv6 트래픽을 가로채려면 VPC, 서브넷 및 어플라이언스가 IPv6를 지원하는지 확인합니다.
- **Important**

  미들박스 어플라이언스를 통해 전송 게이트웨이, NAT 게이트웨이 또는 Network Load Balancer와 같은 AWS 관리형 서비스에서 해당 서비스가 연결된 서브넷으로 트래픽을 라우팅하지 않습니다. 이 구성은 예측할 수 없는 라우팅 동작을 유발하며 지원되지 않습니다.

게이트웨이와 어플라이언스 간 트래픽 라우팅

인바운드 VPC 트래픽을 어플라이언스로 라우팅하려면 라우팅 테이블을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결하고 어플라이언스의 네트워크 인터페이스를 VPC 트래픽의 대상으로 지정합니다. 다음 예제에서 VPC에는 인터넷 게이트웨이, 어플라이언스 및 인스턴스가 있는 서브넷이 있습니다. 인터넷으로부터의 트래픽은 어플라이언스를 통해 라우팅됩니다.



이 라우팅 테이블을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이와 연결합니다. 첫 번째 항목은 로컬 경로입니다. 두 번째 항목은 서브넷을 대상으로 하는 IPv4 트래픽을 어플라이언스에 대한 네트워크 인터페이스로 전송합니다. 이는 로컬 경로보다 더 구체적인 경로입니다.

대상 주소	대상
VPC CIDR	로컬
### CIDR	*###### #### ##### ID*

또는 로컬 경로의 대상을 어플라이언스의 네트워크 인터페이스로 대체할 수 있습니다. 이렇게 하면 향후 VPC에 추가하는 서브넷으로 향하는 트래픽을 포함하여 모든 트래픽이 해당 어플라이언스로 자동 라우팅됩니다.

대상 주소	대상
VPC CIDR	*###### #### ##### ID*

서브넷에서 다른 서브넷의 어플라이언스로 트래픽을 라우팅하려면 트래픽을 어플라이언스의 네트워크 인터페이스로 라우팅하는 라우팅을 서브넷 라우팅 테이블에 추가합니다. 대상은 로컬 라우팅의 대상보다 덜 구체적이어야 합니다. 예를 들어 인터넷으로 향하는 트래픽의 경우 대상에 대해 `0.0.0.0/0`(모든 IPv4 주소)을 지정합니다.

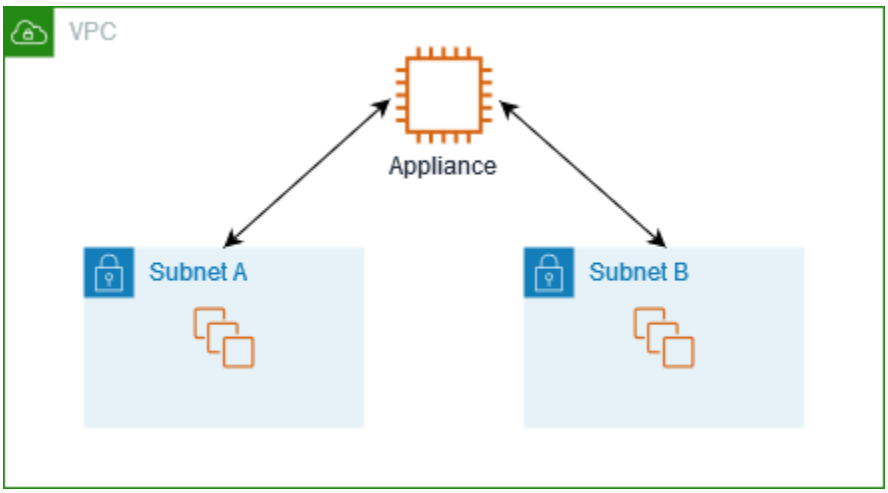
대상 주소	대상
VPC CIDR	로컬
0.0.0.0/0	*###### #### ##### ID*

그런 다음 어플라이언스의 서브넷과 연결된 라우팅 테이블에서 트래픽을 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이로 다시 전송하는 라우팅을 추가합니다.

대상 주소	대상
VPC CIDR	로컬
0.0.0.0/0	*igw-id*

어플라이언스로 서브넷 간 트래픽 라우팅

특정 서브넷을 대상으로 하는 트래픽을 어플라이언스의 네트워크 인터페이스로 라우팅할 수 있습니다. 다음 예제에서 VPC는 서브넷 두 개와 어플라이언스로 구성됩니다. 서브넷 간에 트래픽은 어플라이언스를 통해 라우팅됩니다.



보안 그룹

미들박스 어플라이언스를 통해 서로 다른 서브넷에 있는 인스턴스 간에 트래픽을 라우팅하는 경우 두 인스턴스에 대한 보안 그룹에서 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.

라우팅

다음은 서브넷 A에 대한 라우팅 테이블의 예제입니다. 첫 번째 항목을 사용하면 VPC 인스턴스가 서로 통신할 수 있습니다. 두 번째 항목은 서브넷 A에서 서브넷 B로의 모든 트래픽을 장치의 네트워크 인터페이스로 라우팅합니다.

대상 주소	대상
VPC CIDR	로컬
### B CIDR	*###### #### ##### ID*

다음은 서브넷 B에 대한 라우팅 테이블의 예제입니다. 첫 번째 항목을 사용하면 VPC 인스턴스가 서로 통신할 수 있습니다. 두 번째 항목은 서브넷 B에서 서브넷 A로의 모든 트래픽을 장치의 네트워크 인터페이스로 라우팅합니다.

대상 주소	대상
VPC CIDR	로컬
### A CIDR	*###### #### ##### ID*

또는 로컬 경로의 대상을 어플라이언스의 네트워크 인터페이스로 대체할 수 있습니다. 이렇게 하면 향후 VPC에 추가하는 서브넷으로 향하는 트래픽을 포함하여 모든 트래픽이 해당 어플라이언스로 자동 라우팅됩니다.

대상 주소	대상
VPC CIDR	*###### #### ##### ID*

접두사 목록을 사용한 라우팅

AWS 리소스 전체에서 동일한 CIDR 블록 세트를 자주 참조하는 경우, 고객 관리형 접두사 목록을 생성하여 함께 그룹화할 수 있습니다. 그런 다음 라우팅 테이블 항목의 대상으로 접두사 목록을 지정할 수 있습니다. 나중에 라우팅 테이블을 업데이트할 필요 없이 접두사 목록에 대한 항목을 추가하거나 제거할 수 있습니다.

예를 들어, 여러 VPC 연결이 있는 전송 게이트웨이가 있습니다. VPC는 다음과 같은 CIDR 블록이 있는 두 개의 특정 VPC 연결과 통신할 수 있어야 합니다.

- 10.0.0.0/16
- 10.2.0.0/16

두 항목이 모두 포함된 접두사 목록을 만듭니다. 서브넷 라우팅 테이블에서 라우팅을 생성하고 접두사 목록을 대상 주소로 지정한 다음 전송 게이트웨이를 대상으로 지정합니다.

대상 주소	대상
172.31.0.0/16	로컬
pl-123abc123abc123ab	*tgw-id*

접두사 목록의 최대 항목 수는 라우팅 테이블에 있는 항목 수와 같습니다.

Gateway Load Balancer 엔드포인트로의 라우팅

Gateway Load Balancer를 사용하면 방화벽 같은 가상 어플라이언스 플릿에 트래픽을 분산할 수 있습니다. Gateway Load Balancer를 생성하고 Gateway Load Balancer 엔드포인트 서비스를 구성한 다음 VPC에서 Gateway Load Balancer 엔드포인트를 생성하여 서비스에 연결할 수 있습니다.

트래픽을 Gateway Load Balancer로 라우팅하려면(예: 보안 검사) Gateway Load Balancer 엔드포인트를 라우팅 테이블에서 대상으로 지정합니다.

Gateway Load Balancer 뒤에 있는 보안 어플라이언스의 예제는 the section called “보안 어플라이언스를 사용하여 트래픽 검사”를 참조하세요.

라우팅 테이블에 Gateway Load Balancer 엔드포인트를 지정하려면 VPC 엔드포인트의 ID를 사용합니다. 예를 들어 10.0.1.0/24에 대한 트래픽을 Gateway Load Balancer 엔드포인트로 라우팅하려면 다음 경로를 추가합니다.

대상 주소	대상
10.0.1.0/24	*vpc-endpoint-id*

Gateway Load Balancer 엔드포인트를 대상으로 사용하는 경우 접두사 목록을 대상으로 지정할 수 없습니다. VPC 엔드포인트를 대상으로 하는 접두사 목록 라우팅을 생성하거나 교체하려고 하면 "VPC 엔드포인트를 대상으로 하는 접두사 목록 라우팅을 생성하거나 교체할 수 없습니다."라는 오류가 표시됩니다.

자세한 내용은 Gateway Load Balancers를 참조하세요.

VPC에 대해 라우팅 테이블 생성

다음 작업을 완료하여 VPC에 대한 사용자 지정 라우팅 테이블을 생성하고 구성합니다. 기본적으로 새로운 라우팅 테이블에는 VPC 내부 통신을 허용하는 로컬 라우팅이 포함되어 있습니다. 네트워크 트래픽을 목적지 IP 주소 범위에 따라 특정 대상으로 보내는 라우팅을 추가할 수 있습니다.

특정 서브넷에 라우팅 테이블의 경로를 적용하려면 라우팅 테이블을 서브넷과 연결해야 합니다. 라우팅 테이블은 여러 서브넷과 연결될 수 있습니다. 그러나 서브넷은 한 번에 하나의 라우팅 테이블에만 연결할 수 있습니다. 테이블과 명시적으로 연결되지 않은 서브넷은 기본적으로 기본 라우팅 테이블과 암시적으로 연결됩니다.

서브넷과 라우팅 테이블의 연결을 해제할 수 있습니다. 서브넷에 다른 라우팅 테이블을 명시적으로 연결하지 않는 한 서브넷에는 기본 라우팅 테이블이 암시적으로 연결됩니다.

> **Note**
>
> VPC당 생성할 수 있는 라우팅 테이블 수에는 할당량이 있습니다. 라우팅 테이블당 추가할 수 있는 라우팅 수에도 할당량이 있습니다. 자세한 내용은 Amazon VPC 할당량 섹션을 참조하세요.

작업

- 라우팅 테이블 생성
- 라우팅 테이블에 라우팅 추가
- 서브넷과 라우팅 테이블 연결

라우팅 테이블 생성

콘솔을 사용하여 라우팅 테이블을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 라우팅 테이블 생성을 선택합니다.
4. (선택 사항) 이름(Name)에 라우팅 테이블의 이름을 입력합니다.
5. VPC에서 VPC를 선택합니다.
6. (선택 사항) 태그를 추가하려면 새 태그 추가(Add new tag)를 선택하고 태그 키와 태그 값을 입력합니다.
7. 라우팅 테이블 생성을 선택합니다.

AWS CLI를 사용하여 라우팅 테이블 생성

create-route-table 명령을 사용합니다.

라우팅 테이블에 라우팅 추가

콘솔을 사용하여 라우팅 테이블에 라우팅을 추가하려면

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

2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
4. 라우팅 추가(Add route)를 선택합니다.
5. 목적지 란에 다음 중 하나를 입력합니다.
   - IP 주소 범위 - 예: 192.168.0.0/16
   - 단일 IP 주소 - 예: 192.168.10.1/32
   - 접두사 목록 ID - 예: pl-0abcdef1234567890
6. 대상에서 리소스 유형(예: 네트워크 인터페이스)을 선택한 다음 리소스 ID(예: eni-11223344556677889)를 입력합니다.
7. 변경 사항 저장을 선택합니다.

AWS CLI를 사용하여 라우팅 테이블에 라우팅 추가

create-route 명령을 사용합니다.

서브넷과 라우팅 테이블 연결

콘솔을 사용하여 라우팅 테이블을 서브넷과 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. [서브넷 연결(Subnet associations)] 탭에서 [서브넷 연결 편집(Edit subnet associations)]을 선택합니다.
4. 라우팅 테이블과 연결할 서브넷에 대한 확인란을 선택합니다.
5. [연결 저장(Save associations)]을 선택합니다.

AWS CLI를 사용하여 서브넷과 라우팅 테이블 연결 또는 연결 해제

- associate-route-table
- disassociate-route-table

서브넷 라우팅 테이블 관리

다음 절차에 따라 라우팅 테이블을 사용하여 VPC 라우팅을 관리합니다.

작업

- 서브넷의 라우팅 테이블 확인
- 명시적 연결 확인
- 라우팅 추가, 수정 및 제거
- 경로 전파 활성화 또는 비활성화
- 서브넷의 라우팅 테이블 변경

서브넷의 라우팅 테이블 확인

Amazon VPC 콘솔에서 서브넷의 세부 정보를 살펴보면 서브넷이 연결되어 있는 라우팅 테이블을 확인할 수 있습니다.

콘솔을 사용해 서브넷의 라우팅 테이블을 확인하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. 라우팅 테이블 ID와 경로에 대한 정보를 보려면 라우팅 테이블(Route Table) 탭을 선택합니다. 기본 라우팅 테이블에 대한 연결인지, 해당 연결이 명시적인지 확인하려면 명시적 연결 확인 단원을 참조하십시오.

명시적 연결 확인

라우팅 테이블과 명시적으로 연결된 서브넷 또는 게이트웨이와 그 수를 확인할 수 있습니다.

기본 라우팅 테이블에는 명시적 및 암시적 서브넷 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. 이 작업을 해야 하는 대표적인 이유는 기본 라우팅 테이블 교체을 참조하세요.

콘솔을 사용하여 명시적으로 연결된 서브넷을 확인하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.

3. 명시적 서브넷 연결(Explicit subnet association) 열을 선택하여 명시적으로 연결된 서브넷을 확인하고 기본(Main) 열을 선택하여 기본 라우팅 테이블인지도 확인합니다.
4. 라우팅 테이블을 선택하고 서브넷 연결(Subnet associations) 탭을 선택합니다.
5. 명시적으로 서브넷 연결(Explicit subnet associations) 아래에 있는 서브넷은 라우팅 테이블과 명시적으로 연결되어 있습니다. 명시적 연결이 없는 서브넷(Subnets without explicit associations) 아래에 있는 서브넷은 라우팅 테이블로 같은 VPC에 속해 있지만 어떤 라우팅 테이블과도 연결되지 않아 VPC 기본 라우팅 테이블과 암시적으로 연결되어 있습니다.

콘솔을 사용하여 명시적으로 연결된 게이트웨이를 확인하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 라우팅 테이블을 선택하고 엣지 연결(Edge associations) 탭을 선택합니다.

AWS CLI를 사용하여 하나 이상의 라우팅 테이블을 설명하고 해당 연결 보기

describe-route-tables 명령을 사용합니다.

라우팅 추가, 수정 및 제거

라우팅 테이블에서 라우팅을 추가, 삭제 및 수정할 수 있습니다.

Site-to-Site VPN 연결에 대한 정적 라우팅 작업에 대한 자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 Site-to-Site VPN 연결에 대한 고정 경로 편집을 참조하세요.

고려 사항

- 사용자가 추가한 경로만 수정할 수 있습니다.
- 라우팅을 수정하거나 삭제하면 이러한 라우팅을 사용하는 기존 연결이 영향을 받습니다. 다른 라우팅을 사용하는 연결은 영향을 받지 않습니다.
- 라우팅 테이블당 추가할 수 있는 라우팅 수에는 할당량이 있습니다. 자세한 내용은 Amazon VPC 할당량 섹션을 참조하세요.

콘솔을 사용하여 라우팅 테이블의 경로를 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.

3. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
4. 라우팅을 추가하려면 라우팅 추가(Add route)를 선택합니다. 대상에 IP 주소 범위, 단일 IP 주소 또는 접두사 목록의 ID를 입력합니다. 대상에서 리소스 유형을 선택하고, 리소스의 ID를 입력합니다.
5. 라우팅을 수정하려면 새 대상 CIDR 블록 또는 접두사 목록 ID를 입력하고 대상을 선택합니다.
6. 라우팅을 삭제하려면 제거(Remove)를 선택합니다.
7. 변경 사항 저장을 선택합니다.

AWS CLI를 사용하여 라우팅 테이블의 라우팅 업데이트

명령줄 도구 또는 API를 사용하여 라우팅을 추가하면 대상 CIDR 블록이 표준 형식으로 자동 수정됩니다. 예를 들어 CIDR 블록에 대해 `100.68.0.18/18`을 지정하면 대상 CIDR 블록이 `100.68.0.0/18`인 라우팅이 생성됩니다.

- create-route
- replace-route
- delete-route

경로 전파 활성화 또는 비활성화

라우팅 전파를 사용하면 가상 프라이빗 게이트웨이가 라우팅 테이블에 경로를 자동으로 전파할 수 있습니다. 따라서 VPN 경로를 수동으로 추가하거나 제거할 필요가 없습니다.

이 프로세스를 완료하려면 가상 프라이빗 게이트웨이가 있어야 합니다.

자세한 내용은 Site-to-Site VPN 사용 설명서의 Site-to-Site VPN 라우팅 옵션을 참조하세요.

콘솔을 사용하여 라우팅 전파 활성화 또는 비활성화

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 작업, Edit route propagation(라우팅 속성 편집)을 선택합니다.
4. 가상 프라이빗 게이트웨이 옆에 있는 활성화 확인란을 선택하거나 선택 취소합니다.
5. 저장을 선택합니다.

AWS CLI를 사용하여 라우팅 전파 활성화 또는 비활성화

- enable-vgw-route-propagation
- disable-vgw-route-propagation

서브넷의 라우팅 테이블 변경

서브넷에 대한 라우팅 테이블 연결을 변경할 수 있습니다.

라우팅 테이블을 변경하면 새 라우팅 테이블에 동일한 대상의 동일한 트래픽에 대한 라우팅이 포함되어 있지 않은 경우 서브넷의 기존 연결이 삭제됩니다.

콘솔을 사용하여 서브넷 라우팅 테이블 연결을 변경하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택한 후 서브넷을 선택합니다.
3. 라우팅 테이블(Route table) 탭에서 라우팅 테이블 연결 편집(Edit route table association)을 선택합니다.
4. 라우팅 테이블 ID(Route table ID)에서 새 라우팅 테이블을 선택합니다.
5. 저장을 선택합니다.

AWS CLI를 사용하여 서브넷과 연결된 라우팅 테이블 변경

replace-route-table-association 명령을 사용합니다.

기본 라우팅 테이블 교체

이 섹션에서는 VPC의 기본 경로 테이블을 변경하는 방법을 설명합니다.

콘솔을 사용하여 기본 라우팅 테이블을 바꾸려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)을 선택한 후 새로운 기본 라우팅 테이블을 선택합니다.
3. 작업(Actions), 기본 라우팅 테이블 설정(Set main route table)을 선택합니다.
4. 확인 메시지가 나타나면 **set**을 입력한 다음 확인(OK)을 선택합니다.

AWS CLI를 사용하여 기본 라우팅 테이블 교체

- replace-route-table-association 명령을 사용합니다.

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 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 서브넷 연결(Subnet associations) 탭에서 서브넷 연결 편집(Edit subnet associations)을 선택합니다.
4. 서브넷에 대한 확인란을 선택 취소합니다.
5. [연결 저장(Save associations)]을 선택합니다.

게이트웨이 라우팅 테이블을 사용하여 VPC로 들어오는 트래픽 제어

게이트웨이 라우팅 테이블을 사용하여 VPC로 들어오는 트래픽을 제어하려면 인터넷 게이트웨이나 가상 프라이빗 게이트웨이를 라우팅 테이블과 연결하거나 연결 해제합니다. 자세한 내용은 게이트웨이 라우팅 테이블 섹션을 참조하세요.

콘솔을 사용하여 게이트웨이를 라우팅 테이블과 연결하거나 연결 해제하려면 다음을 수행하세요.

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 엣지 연결(Edge associations) 탭에서 엣지 연결 편집(Edit edge associations)을 선택합니다.
4. 게이트웨이에 대한 확인란을 선택하거나 선택 취소하세요.
5. 변경 사항 저장을 선택합니다.

AWS CLI를 사용하여 게이트웨이를 라우팅 테이블과 연결하려면

associate-route-table 명령을 사용합니다. 다음 예에서는 지정한 라우팅 테이블을 지정한 인터넷 게이트웨이와 연결합니다.

```
aws ec2 associate-route-table
    --route-table-id rtb-01234567890123456 \
    --gateway-id igw-11aa22bb33cc44dd1
```

AWS CLI를 사용하여 게이트웨이와 라우팅 테이블의 연결을 해제하려면

disassociate-route-table 명령을 사용합니다. 라우팅 테이블과 게이트웨이 사이의 연결 ID를 지정합니다.

```
aws ec2 disassociate-route-table \
    --association-id rtbassoc-0abcdef1234567890
```

로컬 경로의 대상 교체 또는 복원

기본 로컬 경로의 대상을 변경할 수 있습니다. 로컬 라우팅의 대상을 교체하면 나중에 기본 `local` 대상으로 복원할 수 있습니다. VPC에 여러 CIDR 블록이 있는 경우 라우팅 테이블에 CIDR 블록당 하나씩 여러 개의 로컬 라우팅이 있습니다. 필요에 따라 각 로컬 라우팅의 대상을 교체하거나 복원할 수 있습니다.

콘솔을 사용하여 로컬 라우팅을 교체하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 라우팅(Routes) 탭에서 라우팅 편집(Edit routes)을 선택합니다.
4. 로컬 라우팅에서 대상(Target)을 지운 다음 새로운 대상을 선택합니다.
5. 변경 사항 저장을 선택합니다.

콘솔을 사용하여 로컬 라우팅의 대상을 복원하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 라우팅 테이블(Route Tables)를 선택한 후 라우팅 테이블을 선택합니다.
3. 작업(Actions), Edit routes(라우팅 편집)를 선택합니다.
4. 라우팅에서 대상(Target)을 지운 다음 로컬(local)을 선택합니다.
5. 변경 사항 저장을 선택합니다.

AWS CLI를 사용하여 로컬 라우팅의 대상을 대체하려면

replace-route 명령을 사용합니다. 다음 예제에서는 로컬 라우팅의 대상을 지정된 라우팅 테이블로 바꿉니다.

```
aws ec2 replace-route \
    --route-table-id rtb-01234567890123456 \
    --destination-cidr-block 10.0.0.0/16 \
    --network-interface-id eni-11223344556677889
```

AWS CLI를 사용하여 로컬 라우팅의 대상을 복원하려면

다음 예제에서는 지정된 라우팅 테이블의 로컬 대상을 복원합니다.

```
aws ec2 replace-route \
    --route-table-id rtb-01234567890123456 \
    --destination-cidr-block 10.0.0.0/16 \
    --local-target
```

VPC의 고급 라우팅

VPC에 대한 고급 라우팅 시나리오를 구성합니다. 이 섹션에서는 트래픽 흐름을 관리하기 위한 정적 및 동적 라우팅 접근 방식을 모두 다룹니다.

- 정적 수신 라우팅: BYOIP(Bring Your Own IP) 주소 풀로 향하는 인바운드 인터넷 트래픽을 VPC 내의 특정 네트워크 인터페이스로 전달하도록 정적 경로를 구성합니다.
- VPC Route Server를 사용한 동적 라우팅: BGP 기반 동적 라우팅을 통해 VPC 및 인터넷 게이트웨이 라우팅 테이블을 자동으로 업데이트하여 워크로드에 내결함성 및 자동 장애 조치를 제공합니다.

내용

- 인터넷 트래픽을 단일 네트워크 인터페이스로 라우팅
- VPC Route Server를 사용한 VPC의 동적 라우팅

인터넷 트래픽을 단일 네트워크 인터페이스로 라우팅

대규모 퍼블릭 IP 주소 풀로 향하는 인바운드 인터넷 트래픽을 VPC의 단일 탄력적 네트워크 인터페이스(ENI)로 라우팅할 수 있습니다.

이전에는 인터넷 게이트웨이가 VPC의 네트워크 인터페이스와 직접 연결된 퍼블릭 IP 주소로 향하는 트래픽만 수락했습니다. 인스턴스 유형에는 네트워크 인터페이스와 연결할 수 있는 IP 주소 수에 제한

이 있으므로 이러한 제한보다 큰 IP 풀의 트래픽을 처리해야 하는 통신 및 사물 인터넷(IoT)과 같은 산업에 문제가 발생합니다.

이 라우팅은 인바운드 인터넷 연결에서 복잡한 주소 변환을 제거합니다. 자체 퍼블릭 IP 풀(BYOIP)을 가져오고 전체 풀의 트래픽을 수락하고 단일 네트워크 인터페이스로 라우팅하도록 VPC 인터넷 게이트웨이를 구성할 수 있습니다. 이 기능은 다음과 같은 경우에 특히 유용합니다.

- 통신: 주소 변환 오버헤드 없이 대규모 구독자 IP 풀 관리
- IoT 애플리케이션: 수천 개의 디바이스 IP 주소에서 트래픽 통합
- 모든 시나리오: ENI 연결 제한을 초과하는 트래픽 라우팅 요구

장애 조치 시나리오 중에 동적 라우팅 업데이트를 위해 이 라우팅을 VPC Route Server와 통합할 수 있습니다.

주요 이점

이러한 라우팅 접근 방식에는 다음과 같은 이점이 있습니다.

- 주소 번역 불필요 - 직접 라우팅으로 NAT 복잡성 제거
- ENI 제한 우회 - 인스턴스 연결 제한보다 큰 IP 풀 처리
- 산업 최적화 - 통신 및 IoT 요구 사항에 맞게 특별히 구축됨
- 동적 장애 조치 - 자동 업데이트를 위해 Route Server와 통합

가용성

이 기능은 모든 AWS 상용 리전, AWS 중국 리전 및 AWS GovCloud 리전에서 사용할 수 있습니다.

내용

- 시작하기 전 준비 사항
- 이 기능의 작동 방식
- 1단계: VPC 생성
- 2단계: 인터넷 게이트웨이 생성 및 연결
- 3단계: 대상 인스턴스에 대한 서브넷 생성
- 4단계: 서브넷에 대한 라우팅 테이블 생성
- 5단계: 대상 인스턴스에 대한 보안 그룹 생성
- 6단계: 대상 EC2 인스턴스 시작

- 7단계: 인터넷 게이트웨이 라우팅 테이블 생성
- 8단계: 라우팅 테이블을 인터넷 게이트웨이와 연결
- 9단계: BYOIP 풀을 인터넷 게이트웨이와 연결
- 10단계: 인스턴스를 대상으로 정적 경로 추가
- 11단계: 대상 인스턴스 구성
- 12단계: 트래픽 처리를 위한 인스턴스 구성
- 13단계: 연결 테스트
- 문제 해결
- 고급 옵션: 동적 라우팅을 위한 라우팅 서버 통합
- 정리

시작하기 전 준비 사항

이 자습서를 시작하기 전에 다음 사항을 확인합니다.

1. BYOIP 풀: 이미 자체 IP 주소 범위를 AWS로 가져왔을 것입니다. Amazon EC2에서 고유 IP 주소 가져오기(BYOIP)의 단계를 완료합니다.
2. BYOIP 풀 확인: 다음을 실행하여 풀이 준비되었는지 확인합니다.

   ```
   aws ec2 describe-public-ipv4-pools --region us-east-1
   ```

   출력에서 풀을 찾고 `PoolAddressRanges`에 `Available` 주소가 표시되는지 확인합니다.
3. 권한 적합성: AWS 계정에 VPC 리소스, EC2 인스턴스를 생성하고 BYOIP 풀을 관리할 수 있는 권한이 있는지 확인합니다.

이 기능의 작동 방식

이 섹션에서는 인터넷 게이트웨이 수신 라우팅의 기술적 개념과 인터넷에서 대상 인스턴스에 대한 트래픽 플로우 방식을 설명합니다.

인터넷 게이트웨이 수신 라우팅을 사용하는 이유

이전에는 ENI 연결 제한으로 인해 많은 수의 IP 주소에 대한 트래픽을 통합하기 위해 주소 변환을 수행해야 했습니다. 이 개선 사항을 통해 BYOIP 풀을 대상 인스턴스로 직접 라우팅할 수 있게 되어 이와 같은 복잡성이 제거됩니다.

라우팅 작동 방식

이 기능은 BYOIP 프로세스에 따라 AWS에 가져오는 퍼블릭 IP CIDR에서만 작동합니다. BYOIP 프로세스는 해당 계정이 퍼블릭 IP CIDR을 소유하도록 보장합니다. BYOIP 퍼블릭 CIDR이 있으면 다음을 수행합니다.

1. 이 퍼블릭 IP 주소 풀을 인터넷 게이트웨이 라우팅 테이블과 연결합니다. 인터넷 게이트웨이는 이미 VPC와 연결되어 있어야 합니다. 이 연결을 통해 VPC는 IP CIDR로 향하는 트래픽을 수락할 수 있습니다. 인터넷 게이트웨이에 서브넷과 공유되지 않는 전용 라우팅 테이블이 있는지 확인합니다.
2. 이제 BYOIP 풀을 인터넷 게이트웨이 라우팅 테이블과 연결했으므로 대상이 IP CIDR 또는 인터넷 게이트웨이 라우팅 테이블의 하위 집합과 동일한 경로를 입력할 수 있습니다. 이 경로의 대상은 트래픽을 라우팅하려는 ENI입니다.
3. BYOIP CIDR로 향하는 트래픽이 AWS에 들어오면 AWS는 인터넷 게이트웨이 라우팅 테이블을 살펴보고 그에 따라 트래픽을 관련 VPC로 라우팅합니다.
4. VPC 내에서 인터넷 게이트웨이는 트래픽을 대상 ENI로 라우팅합니다.
5. 대상(워크로드와 연결된 탄력적 네트워크 인터페이스)은 트래픽을 처리합니다.

모범 사례:

- 라우팅 테이블을 별도로 유지: 인터넷 게이트웨이 라우팅 테이블은 인터넷 게이트웨이 전용이어야 합니다. 이 라우팅 테이블을 VPC 서브넷과 연결하지 마세요. 서브넷 라우팅에는 별도의 라우팅 테이블을 사용합니다.
- BYOIP IP를 직접 할당하지 않음: BYOIP 풀의 퍼블릭 IP 주소를 EC2 인스턴스 또는 네트워크 인터페이스에 직접 연결하지 마세요. 인터넷 게이트웨이 수신 라우팅 기능은 직접 IP 연결 없이 트래픽을 인스턴스로 라우팅합니다.

> **Important**
>
> VPC 퍼블릭 액세스 차단(BPA)을 사용하는 경우 BPA가 활성화되면 서브넷 수준 BPA 제외를 설정했더라도 수신 라우팅을 사용하여 서브넷으로의 트래픽을 차단합니다. 수신 라우팅에는 서브넷 수준 제외가 작동하지 않습니다. BPA가 활성화된 상태에서 수신 라우팅 트래픽을 허용하려면 다음을 수행합니다.
>
> - BPA를 완전히 비활성화 또는
> - VPC 수준 제외 사용

1단계: VPC 생성

이 단계를 완료하여 대상 인스턴스와 인터넷 게이트웨이를 호스팅할 VPC를 생성합니다.

> **Note**
>
> VPC 할당량 한도에 도달하지 않았는지 확인합니다. 자세한 내용은 Amazon VPC 할당량 섹션을 참조하세요.

AWS 콘솔

1. Amazon VPC 콘솔을 엽니다.
2. VPC 대시보드에서 VPC 생성을 선택합니다.
3. 생성할 리소스에서 VPC 전용을 선택합니다.
4. 이름 태그에 VPC의 이름을 입력합니다(예: **IGW-Ingress-VPC**).
5. IPv4 CIDR 블록에 CIDR 블록을 입력합니다(예: **10.0.0.0/16**).
6. VPC 생성을 선택합니다.

AWS CLI

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=IGW-Ingress-VPC}]' --region us-east-1
```

2단계: 인터넷 게이트웨이 생성 및 연결

이 단계를 완료하여 인터넷 게이트웨이를 생성하고 VPC에 연결해 인터넷 연결을 활성화합니다.

AWS 콘솔

1. Amazon VPC 콘솔을 엽니다.
2. VPC 콘솔에서 인터넷 게이트웨이를 선택합니다.
3. 인터넷 게이트웨이 생성을 선택합니다.
4. 이름 태그에 인터넷 게이트웨이의 이름을 입력합니다(예: **IGW-Ingress-Gateway**).
5. 인터넷 게이트웨이 생성을 선택합니다.
6. 인터넷 게이트웨이를 선택한 후 작업, VPC에 연결을 선택합니다.

7. VPC를 선택한 후 인터넷 게이트웨이 연결을 선택합니다.

AWS CLI

```
aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-
gateway,Tags=[{Key=Name,Value=IGW-Ingress-Gateway}]' --region us-east-1

aws ec2 attach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0 --region us-east-1
```

참고: 리소스 ID를 이전 단계의 실제 ID로 바꿉니다.

3단계: 대상 인스턴스에 대한 서브넷 생성

이 단계를 완료하여 대상 인스턴스가 배포될 서브넷을 생성합니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 서브넷(Subnets)을 선택합니다.
2. 서브넷 생성(Create subnet)을 선택합니다.
3. VPC ID에서 VPC를 선택합니다.
4. 서브넷 이름에 이름을 입력합니다(예: **`Target-Subnet`**).
5. 가용 영역(Availability Zone)에서 서브넷의 영역을 선택하거나 AWS에서 자동으로 선택하도록 기본값인 기본 설정 없음(No Preference)을 그대로 둘 수 있습니다.
6. IPv4 CIDR 블록에서 수동 입력을 선택하고 CIDR 블록을 입력합니다(예: **`10.0.1.0/24`**).
7. 서브넷 생성(Create subnet)을 선택합니다.

AWS CLI

```
aws ec2 create-subnet \
    --vpc-id vpc-0123456789abcdef0 \
    --cidr-block 10.0.1.0/24 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Target-Subnet}]' \
    --region us-east-1
```

4단계: 서브넷에 대한 라우팅 테이블 생성

이 단계를 완료하여 서브넷에 대한 라우팅 테이블을 생성하고 서브넷과 연결합니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 라우팅 테이블을 선택합니다.
2. 라우팅 테이블 생성을 선택합니다.
3. 이름에 라우팅 테이블의 이름을 입력합니다(예: **Target-Subnet-Route-Table**).
4. VPC에서 VPC를 선택합니다.
5. 라우팅 테이블 생성을 선택합니다.
6. 라우팅 테이블을 선택한 후 작업, 서브넷 연결 편집을 선택합니다.
7. 서브넷을 선택한 후 연결 저장을 선택합니다.

AWS CLI

```
aws ec2 create-route-table \
    --vpc-id vpc-0123456789abcdef0 \
    --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Target-Subnet-
Route-Table}]' \
    --region us-east-1

aws ec2 associate-route-table \
    --route-table-id rtb-0987654321fedcba0 \
    --subnet-id subnet-0123456789abcdef0 \
    --region us-east-1
```

5단계: 대상 인스턴스에 대한 보안 그룹 생성

이 단계를 완료하여 대상 인스턴스에 대한 네트워크 액세스를 제어하는 보안 그룹을 생성합니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 보안 그룹을 선택합니다.
2. 보안 그룹 생성을 선택합니다.
3. 보안 그룹 이름에 이름을 입력합니다(예: **IGW-Target-SG**).
4. 설명에 **Security group for IGW ingress routing target instance**를 입력합니다.
5. VPC에서 해당 VPC를 선택합니다.
6. 인바운드 규칙을 추가하려면 인바운드 규칙을 선택합니다. 각 규칙에 대해 규칙 추가를 선택하고 다음을 지정합니다.

- 유형: 모든 ICMP - IPv4, 소스: 0.0.0.0/0(ping 테스트용).
- 유형: SSH, 포트: 22, 소스: 0.0.0.0/0(EC2 Instance Connect용).

> **Note**
>
> 이 보안 그룹은이 자습서의 모든 인터넷 트래픽에 SSH 포트를 엽니다. 이 자습서는 교육용이므로 프로덕션 환경에 맞게 구성해서는 안 됩니다. 프로덕션 환경에서는 SSH 액세스를 특정 IP 범위로 제한합니다.

- 보안 그룹 생성을 선택합니다.

AWS CLI

```
aws ec2 create-security-group \
    --group-name IGW-Target-SG \
    --description "Security group for IGW ingress routing target instance" \
    --vpc-id vpc-0123456789abcdef0 \
    --region us-east-1

aws ec2 authorize-security-group-ingress \
    --group-id sg-0123456789abcdef0 \
    --protocol icmp \
    --port -1 \
    --cidr 0.0.0.0/0 \
    --region us-east-1

aws ec2 authorize-security-group-ingress \
    --group-id sg-0123456789abcdef0 \
    --protocol tcp \
    --port 22 \
    --cidr 0.0.0.0/0 \
    --region us-east-1
```

6단계: 대상 EC2 인스턴스 시작

이 단계를 완료하여 BYOIP 풀에서 트래픽을 수신할 EC2 인스턴스를 시작합니다.

AWS 콘솔

1. Amazon EC2 콘솔을 엽니다.
2. 인스턴스 시작을 선택합니다.
3. 이름에 인스턴스의 이름을 입력합니다(예: **`IGW-Target-Instance`**).
4. 애플리케이션 및 OS 이미지(Amazon Machine Image)에서 Amazon Linux 2023 AMI를 선택합니다.
5. 인스턴스 유형에서 t2.micro(프리 티어에서 사용 가능)를 선택합니다.
6. 키 페어(로그인)에서 기존 키 페어를 선택하거나 새 키 페어를 생성합니다.
7. 네트워크 설정에서 편집을 선택하고 다음을 구성합니다.
   - VPC: VPC 선택
   - 서브넷: 서브넷 선택
   - 퍼블릭 IP 자동 할당: 활성화
   - 방화벽(보안 그룹): 기존 보안 그룹을 선택한 후 해당 보안 그룹 선택
8. 인스턴스 시작을 선택합니다.
9. 중요: 시작 후 인스턴스 세부 정보로 이동하여 네트워크 인터페이스 ID("eni-"로 시작)를 기록해 둡니다. 10단계에서 이 ID가 필요합니다.

AWS CLI

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name your-key-pair \
    --security-group-ids sg-0123456789abcdef0 \
    --subnet-id subnet-0123456789abcdef0 \
    --associate-public-ip-address \
    --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=IGW-Target-
Instance}]' \
    --region us-east-1
```

콘솔에서 ENI ID 찾기:

1. EC2 콘솔에서 인스턴스를 선택합니다.
2. 네트워킹 탭으로 이동합니다.
3. 네트워크 인터페이스 ID(예: `eni-0abcdef1234567890`)를 기록해 둡니다.

를 사용하여 ENI ID 찾기:AWS CLI

```
aws ec2 describe-instances --instance-ids i-0123456789abcdef0 --query
 'Reservations[0].Instances[0].NetworkInterfaces[0].NetworkInterfaceId' --output text
 --region us-east-1
```

7단계: 인터넷 게이트웨이 라우팅 테이블 생성

이 단계를 완료하여 수신 라우팅을 처리할 인터넷 게이트웨이 전용 라우팅 테이블을 생성합니다.

AWS 콘솔

1. VPC 콘솔에서 라우팅 테이블을 선택합니다.
2. 라우팅 테이블 생성을 선택합니다.
3. 이름에 라우팅 테이블의 이름을 입력합니다(예: **IGW-Ingress-Route-Table**).
4. VPC에서 VPC를 선택합니다.
5. 라우팅 테이블 생성을 선택합니다.
6. 라우팅 테이블을 선택한 후 엣지 연결 탭을 선택합니다.
7. 엣지 연결 편집을 선택합니다.
8. 인터넷 게이트웨이를 선택한 후 변경 사항 저장을 선택합니다.

AWS CLI

```
aws ec2 create-route-table \
    --vpc-id vpc-0123456789abcdef0 \
    --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=IGW-Ingress-
Route-Table}]' \
    --region us-east-1
```

8단계: 라우팅 테이블을 인터넷 게이트웨이와 연결

이 단계를 완료하여 라우팅 테이블을 인터넷 게이트웨이와 연결해 수신 라우팅 기능을 활성화합니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 라우팅 테이블을 선택한 후 생성한 라우팅 테이블을 선택합니다.
2. 엣지 연결(Edge associations) 탭에서 엣지 연결 편집(Edit edge associations)을 선택합니다.
3. 해당 인터넷 게이트웨이의 확인란을 선택합니다.

4. 변경 사항 저장을 선택합니다.

AWS CLI

```
aws ec2 associate-route-table \
    --route-table-id rtb-0123456789abcdef0 \
    --gateway-id igw-0123456789abcdef0 \
    --region us-east-1
```

9단계: BYOIP 풀을 인터넷 게이트웨이와 연결

이 단계를 완료하여 BYOIP 풀을 인터넷 게이트웨이 라우팅 테이블과 연결하면 VPC가 IP 범위에 대한 트래픽을 수락할 수 있습니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 라우팅 테이블을 선택한 후 생성한 인터넷 게이트웨이 라우팅 테이블을 선택합니다.
2. IPv4 풀 연결 탭을 선택하세요.
3. 연결 편집 버튼을 선택하세요.
4. BYOIP 풀을 선택합니다(예: `pool-12345678901234567`).
5. 연결 저장 버튼을 선택하세요.

AWS CLI

```
aws ec2 associate-route-table \
    --route-table-id rtb-0123456789abcdef0 \
    --public-ipv4-pool pool-12345678901234567 \
    --region us-east-1
```

참고: `rtb-0123456789abcdef0`을 인터넷 게이트웨이 라우팅 테이블 ID로 바꾸고 `pool-12345678901234567`을 BYOIP 풀 ID로 바꿉니다.

10단계: 인스턴스를 대상으로 정적 경로 추가

이 단계를 완료하여 BYOIP 범위에서 대상 인스턴스의 네트워크 인터페이스로 트래픽을 전달하는 경로를 추가합니다.

AWS 콘솔

1. VPC 콘솔 탐색 창에서 라우팅 테이블을 선택한 후 생성한 인터넷 게이트웨이 라우팅 테이블을 선택합니다.
2. 작업(Actions), 라우팅 편집(Edit routes)을 선택합니다.
3. 라우팅 추가(Add route)를 선택합니다.
4. 대상에 BYOIP CIDR 또는 하위 집합을 입력합니다(예: **203.0.113.0/24**). 이 접두사는 /23에서 /28 사이여야 합니다.
5. 대상에서 네트워크 인터페이스를 선택하고 인스턴스의 ENI ID를 입력합니다(예: eni-0abcdef1234567890).
6. 변경 사항 저장을 선택합니다.

AWS CLI

```
aws ec2 create-route \
    --route-table-id rtb-0123456789abcdef0 \
    --destination-cidr-block 203.0.113.0/24 \
    --network-interface-id eni-0abcdef1234567890 \
    --region us-east-1
```

11단계: 대상 인스턴스 구성

이 단계를 완료하여 BYOIP 주소로 향하는 트래픽을 올바르게 처리하도록 대상 인스턴스를 구성합니다.

중요: 연결을 테스트하기 전에 이 인스턴스 구성 단계를 완료합니다(12단계). 수신 라우팅이 제대로 작동하려면 BYOIP 주소에 응답하도록 인스턴스를 구성해야 합니다.

AWS 콘솔

1. EC2 Instance Connect를 사용하여 대상 인스턴스에 연결하려면 다음을 수행합니다.
   - EC2 콘솔에서 인스턴스를 선택합니다.
   - 작업 > 연결을 선택합니다.
   - EC2 Instance Connect 탭을 선택합니다.
   - 연결을 선택합니다.
2. 인스턴스 인터페이스에 특정 BYOIP IP 주소 추가:

   먼저 네트워크 인터페이스 이름을 찾습니다.

```
ip link show
```

그런 다음 IP 주소를 추가합니다(203.0.113.10을 BYOIP 범위의 IP로 대체).

```
sudo ip addr add 203.0.113.10/32 dev eth0
```

참고: 203.0.113.10을 테스트하려는 BYOIP 범위의 IP 주소로 바꿉니다. 인터페이스 이름은 인스턴스 유형에 따라 eth0, ens5이거나 이와 유사할 수 있습니다.

3. EC2 콘솔에서 소스/대상 확인을 비활성화하려면 다음을 수행합니다.

- 인스턴스를 선택합니다.
- 네트워킹 탭으로 이동하여 네트워크 인터페이스를 선택하세요.
- 작업, 소스/대상 확인 변경, 비활성화를 선택합니다.

AWS CLI

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-0abcdef1234567890 \
    --no-source-dest-check \
    --region us-east-1
```

12단계: 트래픽 처리를 위한 인스턴스 구성

이 단계를 완료하여 인스턴스에 BYOIP 주소를 추가하고 소스/대상 확인을 비활성화해 적절한 트래픽 처리를 활성화합니다.

AWS 콘솔

1. EC2 Instance Connect를 사용하여 대상 인스턴스에 연결하려면 다음을 수행합니다.

- EC2 콘솔에서 인스턴스를 선택합니다.
- 작업 > 연결을 선택합니다.
- EC2 Instance Connect 탭을 선택합니다.
- 연결을 선택합니다.

2. 인스턴스 인터페이스에 특정 BYOIP IP 주소 추가:

먼저 네트워크 인터페이스 이름을 찾습니다.

```
ip link show
```

그런 다음 IP 주소를 추가합니다(ens5를 실제 인터페이스 이름으로 대체).

```
sudo ip addr add 203.0.113.10/32 dev ens5
```

참고: 203.0.113.10을 테스트하려는 BYOIP 범위의 IP 주소로 바꿉니다. 인터페이스 이름은 인스턴스 유형에 따라 eth0, ens5이거나 이와 유사할 수 있습니다.

3. EC2 콘솔에서 소스/대상 확인을 비활성화하려면 다음을 수행합니다.

- 인스턴스를 선택합니다.
- 네트워킹 탭으로 이동하여 네트워크 인터페이스를 선택하세요.
- 작업, 소스/대상 확인 변경, 비활성화를 선택합니다.

AWS CLI

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-0abcdef1234567890 \
    --no-source-dest-check \
    --region us-east-1
```

13단계: 연결 테스트

이 단계를 완료하여 인터넷 트래픽이 BYOIP 주소를 통해 대상 인스턴스로 올바르게 라우팅되었는지 확인합니다.

1. 대상 인스턴스에서 tcpdump를 사용하여 수신 트래픽을 모니터링합니다.

```
sudo tcpdump -i any icmp
```

2. 다른 터미널 또는 컴퓨터에서 BYOIP IP 주소에 대한 연결을 테스트합니다.

```
ping 203.0.113.10
```

3. 예상 결과:

- Ping이 성공하고 BYOIP IP 주소의 응답을 표시해야 합니다.
- tcpdump는 다음과 같이 BYOIP 주소에 대한 수신 패킷을 표시해야 합니다.

```
12:34:56.789012 IP 203.0.113.100 > 203.0.113.10: ICMP echo request, id 1234, seq
 1, length 64
12:34:56.789123 IP 203.0.113.10 > 203.0.113.100: ICMP echo reply, id 1234, seq 1,
 length 64
```

- 트래픽은 외부 IP 주소에서 오는 것처럼 보여야 합니다. 이는 인터넷 게이트웨이 수신 라우팅이 인스턴스에 인터넷 트래픽을 전달하고 있음을 증명합니다.

문제 해결

이 섹션을 사용하여 인터넷 게이트웨이 수신 라우팅을 설정할 때 발생할 수 있는 일반적인 문제를 해결합니다.

트래픽이 인스턴스에 도달하지 않음

- 라우팅 테이블의 ENI ID가 대상으로 올바른지 확인합니다.
- BYOIP 풀이 인터넷 게이트웨이 라우팅 테이블과 연결되어 있는지 확인합니다.
- 인스턴스에서 소스/대상 확인이 비활성화되어 있는지 확인합니다.
- 보안 그룹이 테스트 중인 트래픽 유형을 허용하는지 확인합니다.

경로 생성 실패

- BYOIP 풀이 라우팅 테이블과 올바르게 연결되어 있는지 확인합니다.
- 대상 CIDR이 BYOIP 범위 내에 있는지 확인합니다.
- 대상 ENI가 존재하고 실행 중인 인스턴스에 연결되어 있는지 확인합니다.
- BYOIP 접두사가 /23에서 /28 사이인지 확인합니다(이 범위 밖의 접두사는 지원되지 않음).

Ping/연결 실패

- IP 주소가 인스턴스 인터페이스에 추가되었는지 확인합니다.
- 보안 그룹이 ICMP(ping용) 또는 관련 포트를 허용하는지 확인합니다.
- 인스턴스가 실행 중 상태인지 확인합니다.
- 여러 외부 위치에서 테스트합니다.

고급 옵션: 동적 라우팅을 위한 라우팅 서버 통합

자동 장애 조치가 필요한 환경의 경우 이 기능은 VPC Route Server와 통합되어 다음을 수행합니다.

- 인스턴스 장애 발생 시 경로를 동적으로 업데이트합니다.

- 라우팅 관리를 위한 수동 개입을 제거합니다.
- 중요한 워크로드에 대한 엔터프라이즈급 가용성을 제공합니다.

이는 고가용성이 필수적인 통신 및 IoT 사용 사례에 특히 중요합니다.

> **Note**
>
> 여러 BGP 피어와 함께 Route Server를 사용하는 경우 최대 32개의 BGP 피어가 라우팅 서버를 사용하여 동일한 라우팅 테이블에 동일한 접두사를 알릴 수 있습니다.

동적 라우팅, 자동 장애 조치 및 여러 인스턴스에 걸친 로드 분산이 필요한 환경의 경우 AWS Route Server와 통합하는 것이 좋습니다. Route Server는 정적 경로 대신 BGP 기반 동적 라우팅을 활성화하여 다음을 제공합니다.

- BGP를 통한 인스턴스의 동적 라우팅 광고.
- 여러 대상 인스턴스 간의 자동 장애 조치.
- 여러 엔드포인트에 대한 로드 분산.
- BGP 프로토콜을 통한 중앙 집중식 라우팅 관리.

이는 고가용성 및 동적 라우팅 기능이 필요한 엔터프라이즈 배포에 중요한 사용 사례입니다. 자세한 Route Server 설정 지침은 AWS Route Server 설명서를 참조하세요.

정리

이 자습서에서 생성한 리소스에 계속해서 비용이 부과되지 않도록 하려면 해당 리소스를 삭제해야 합니다.

1단계: EC2 인스턴스 종료

이 단계를 완료하여 EC2 인스턴스를 종료하고 컴퓨팅 리소스에 대한 요금 발생을 중지합니다.

AWS 콘솔

1. Amazon EC2 콘솔을 엽니다.
2. EC2 콘솔 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스를 선택한 후 인스턴스 상태, 인스턴스 종료를 차례로 선택합니다.
4. 종료를 선택하여 확인합니다.

AWS CLI

```
aws ec2 terminate-instances --instance-ids i-0123456789abcdef0 --region us-east-1
```

2단계: VPC에서 인터넷 게이트웨이 분리

이 단계를 완료하여 VPC에서 인터넷 게이트웨이를 분리하고 삭제합니다.

AWS 콘솔

1. Amazon VPC 콘솔을 엽니다.
2. VPC 콘솔 탐색 창에서 인터넷 게이트웨이를 선택합니다.
3. 인터넷 게이트웨이를 선택한 후 작업, VPC에서 분리를 선택합니다.
4. 인터넷 게이트웨이 분리를 선택합니다.
5. 분리한 후 작업, 인터넷 게이트웨이 삭제를 선택합니다.
6. 인터넷 게이트웨이 삭제를 선택합니다.

AWS CLI

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0 --region us-east-1

aws ec2 delete-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --region
 us-east-1
```

3단계: VPC 삭제

이 단계를 완료하여 VPC 및 연결된 모든 리소스를 삭제하고 정리 프로세스를 완료합니다.

AWS 콘솔

1. VPC 콘솔에서 VPC를 선택합니다.
2. VPC를 선택한 후 작업, VPC 삭제를 선택합니다.
3. **delete**를 입력하여 확인한 후 삭제를 선택합니다.

AWS CLI

```
aws ec2 delete-vpc --vpc-id vpc-0123456789abcdef0 --region us-east-1
```

> **Note**
>
> VPC를 삭제하면 연결된 서브넷, 라우팅 테이블 및 보안 그룹도 삭제됩니다.

> **Note**
>
> BYOIP 풀은 나중에 사용할 수 있으며 이 정리 프로세스 과정에서 삭제되지 않습니다.

VPC Route Server를 사용한 VPC의 동적 라우팅

Amazon VPC Route Server는 VPC 내에 배포된 워크로드와 인터넷 게이트웨이 간의 트래픽 라우팅을 간소화합니다. 이 기능을 사용하면 VPC Route Server가 선호하는 IPv4 또는 IPv6 경로로 VPC 및 인터넷 게이트웨이 라우팅 테이블을 동적으로 업데이트하여 해당 워크로드에 대한 라우팅 내결함성을 달성합니다. 이를 활용하면 VPC 내에서 트래픽을 자동으로 다시 라우팅할 수 있어 VPC 라우팅의 관리 용이성과 타사 워크로드와의 상호 운용성이 개선됩니다.

라우팅 서버는 다음 라우팅 테이블 유형을 지원합니다.

- 서브넷과 연결되지 않은 VPC 라우팅 테이블
- 서브넷 라우팅 테이블
- 인터넷 게이트웨이 라우팅 테이블

라우팅 서버는 가상 프라이빗 게이트웨이와 연결된 라우팅 테이블을 지원하지 않습니다. 전송 게이트웨이 라우팅 테이블에 라우팅을 전파하려면 Transit Gateway Connect를 사용합니다.

할당량

Amazon VPC Route Server와 관련된 할당량은 라우팅 서버 할당량을 참조하세요.

가격 책정

Amazon VPC Route Server와 관련된 비용에 대한 자세한 내용은 Amazon VPC 요금 페이지의 VPC Route Server 탭을 참조하세요.

내용

- 용어

- Amazon VPC Route Server 작동 방식
- 라우팅 서버 피어 로깅
- 시작하기 자습서

용어

이 가이드에서 사용되는 용어:

- FIB: FIB(Forwarding Information Base)는 사용 가능한 모든 라우팅 정보와 정책을 평가한 후 RIB에서 라우팅 서버가 최적의 경로라고 판단한 경로의 전달 테이블 역할을 합니다. 라우팅 테이블에 설치된 FIB 경로입니다. RIB에 변경 사항이 있을 때마다 FIB가 다시 계산됩니다.
- RIB: RIB(Routing Information Base)는 BGP 피어에서 학습한 경로와 같이 라우터 또는 라우팅 시스템에서 수집한 모든 라우팅 정보와 네트워크 토폴로지 데이터를 저장하는 데이터베이스 역할을 합니다. RIB는 새 라우팅 정보가 수신되거나 기존 경로가 변경되면 지속적으로 업데이트됩니다. 이렇게 하면 라우팅 서버가 항상 네트워크 토폴로지의 최신 상태를 확인할 수 있고 최적의 라우팅 결정을 내릴 수 있습니다.
- 라우팅 서버: 라우팅 서버 구성 요소는 VPC 및 인터넷 게이트웨이 라우팅 테이블을 FIB(Forwarding Information Base)의 IPv4 또는 IPv6 경로로 업데이트합니다. 라우팅 서버는 단일 FIB 및 RIB(Routing Information Base)를 나타냅니다.
- 라우팅 서버 연결: 라우팅 서버 연결은 라우팅 서버와 VPC 사이에 설정된 연결입니다.
- Route Server 엔드포인트: Route Server 엔드포인트는 라우팅 서버와 BGP 피어 간의 BGP(Border Gateway Protocol) 연결을 지원하는 서브넷 내부의 AWS 관리형 구성 요소입니다.
- 라우팅 서버 피어: 라우팅 서버 피어는 라우팅 서버 엔드포인트와에 AWS에 배포된 디바이스(예: EC2 인스턴스에서 실행되는 방화벽 어플라이언스 또는 기타 네트워크 보안 기능) 사이의 세션입니다. 디바이스는 다음 요구 사항을 충족해야 합니다.
  - VPC에 탄력적 네트워크 인터페이스 보유
  - BGP(Border Gateway Protocol) 지원
  - BGP 세션 시작 가능
- 라우팅 서버 전파: 활성화되면 지정한 라우팅 테이블의 FIB에 경로가 설치됩니다. 라우팅 서버는 IPv4 및 IPv6 경로 전파를 지원합니다.

Amazon VPC Route Server 작동 방식

이 섹션에서는 Amazon VPC Route Server의 작동 방식을 설명하고, 서브넷에서 실행되는 워크로드의 라우팅 내결함성을 달성하는 방법을 이해하는 데 유용합니다.

내용

- 개요
- 다이어그램

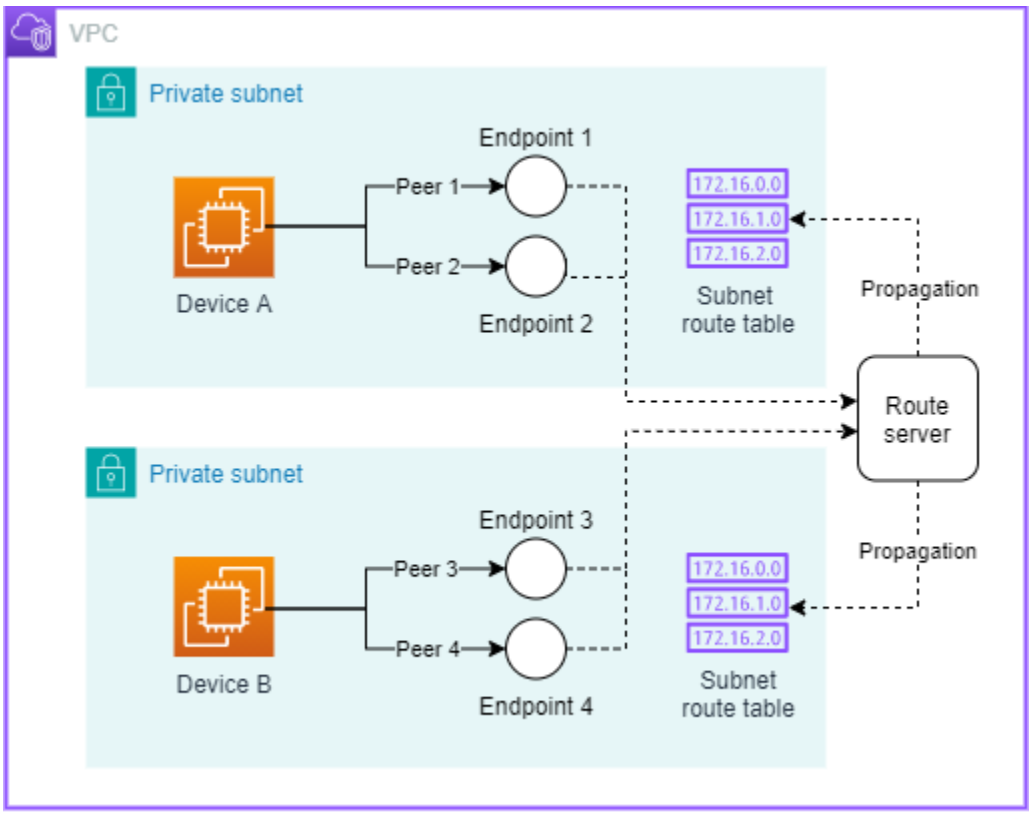
개요

Amazon VPC Route Server 작동 방식:

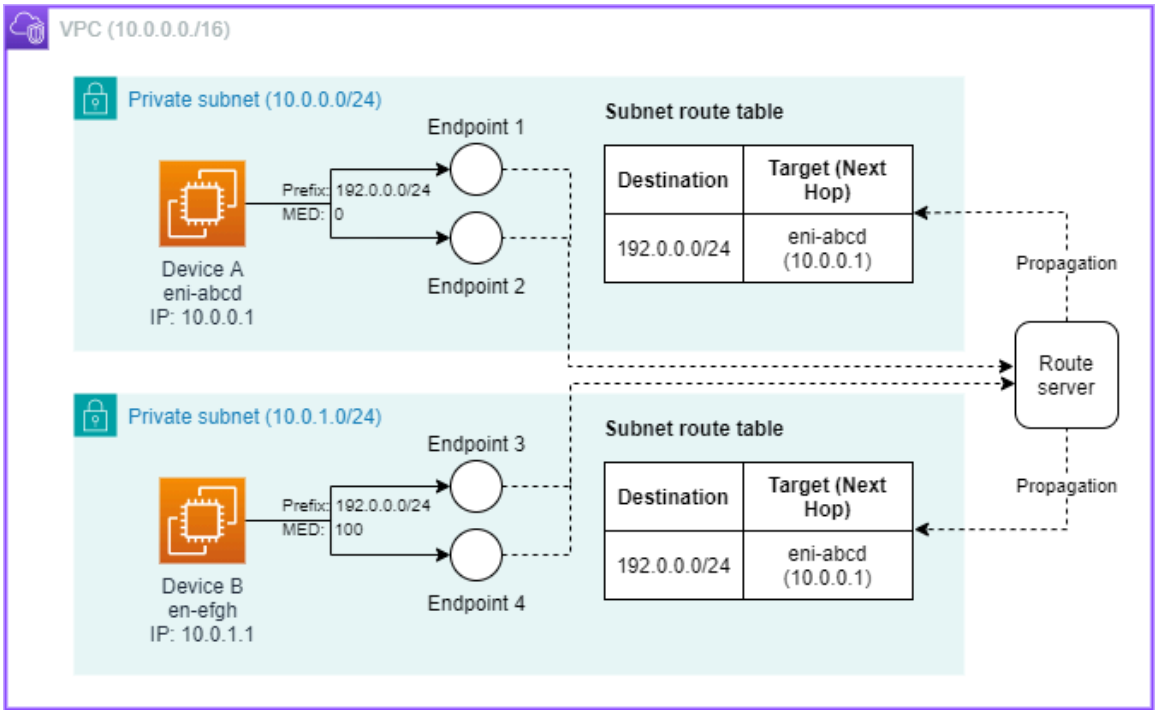
1. 네트워크 디바이스(예: VPC의 EC2 인스턴스에서 실행되는 방화벽)를 구성하여 Amazon VPC Route Server를 사용합니다.
2. 네트워크 디바이스에서 장애가 발생합니다.
3. 라우팅 서버 엔드포인트는 라우팅 서버 피어에 구성된 BFD(Bidirectional Forwarding Detection)를 통해 장애를 감지합니다.
4. 라우팅 서버 엔드포인트는 라우팅 서버를 업데이트하여 장애가 발생한 디바이스가 다음 홉인 RIB(Routing Information Base)의 경로를 철회합니다.
5. 라우팅 서버는 RIB에서 FIB(Forwarding Information Base)를 계산하여 사용할 수 있는 최상의 경로를 선택합니다.
6. 라우팅 서버는 FIB의 경로로 구성된 라우팅 테이블을 업데이트합니다.
7. 모든 새 트래픽은 대기 디바이스로 전달됩니다.

다이어그램

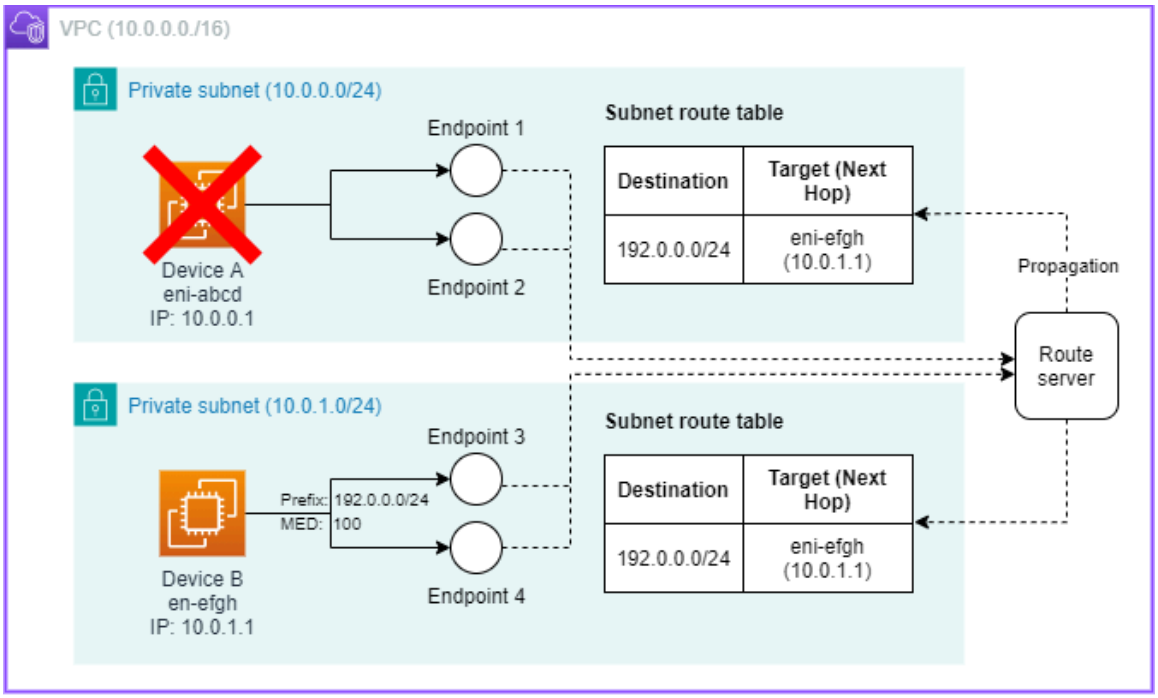
다음은 두 서브넷의 디바이스에 대해 라우팅 서버 엔드포인트가 구성된 VPC 라우팅 서버의 다이어그램 예제입니다.



위의 예제를 기준으로 아래의 예제는 보다 세부적인 설계를 보여줍니다. 여기에서 디바이스 A와 디바이스 B 모두 BGP를 통해 대상 IP가 192.0.0.0/24 범위(192.0.0.0부터 192.0.0.255까지)에 있는 트래픽을 수락할 수 있음을 알립니다. MED(Multi-Exit Discriminator) 속성 0은 라우팅 서버에 디바이스 B보다 디바이스 A를 우선해야 함을 알립니다. 라우팅 서버는 디바이스 A로부터 경로 및 MED 속성을 수신하고 디바이스 A의 네트워크 인터페이스를 "다음 홉"으로 사용하여 서브넷 라우팅 테이블에 해당 경로를 설치합니다. 그에 따라 대상 IP가 192.0.0.0/24 범위에 있는 서브넷 내의 모든 트래픽은 디바이스 A로 보냅니다. 이후 디바이스 A는 트래픽을 처리하고 계속 송신합니다. 192.0.0.0/24에 바인딩된 서브넷(10.0.0.0/24 또는 10.0.1.0/24) 내의 트래픽은 다음 홉으로 디바이스 A eni-abcd(10.0.0.1)로 라우팅됩니다.



아래의 마지막 예제는 라우팅 서버가 장애 조치를 처리하는 방법을 보여줍니다. MED 속성이 높을수록 라우팅 서버에 디바이스 B가 디바이스 A보다 우선한다고 알리지만 디바이스 A eni-abcd(10.0.0.1)가 중단되면 라우팅 서버는 서브넷 라우팅 테이블을 업데이트하고, 192.0.0.0/24로 향하는 트래픽은 다음 홉인 디바이스 B eni-efgh(10.0.1.1)로 라우팅됩니다.



라우팅 서버 피어 로깅

다음의 상황에서 VPC Route Server 피어 로깅을 사용합니다.

- BGP 및 BFD 세션 상태 모니터링
- 연결 문제 해결
- 세션 변경 사항 이력 검토
- 네트워크 상태 추적

가격 책정

- CloudWatch: CloudWatch Logs에 라우팅 서버 피어를 게시할 때는 Vended 로그에 대한 데이터 수집 및 아카이브 요금이 부과됩니다.
- S3: Amazon S3에 라우팅 서버 피어 로그를 게시할 때는 Vended 로그에 대한 데이터 수집 및 아카이브 요금이 부과됩니다.
- Data Firehose: 표준 수집 및 전송 요금이 적용됩니다.

Vended 로그는 볼륨 티어별 요금으로 사용할 수 있고 CloudWatch Logs, Amazon S3 또는 Amazon Data Firehose로 전송되는 특정 AWS 서비스의 로그입니다. 자세히 알아보려면 Amazon CloudWatch 요금을 열고, 로그를 선택하고, 벤딩 로그를 찾으세요.

로그 형식 예

```
{
    "resource_arn": "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0",
    "event_timestamp": 1746643505367,
    "type": "RouteStatus",
    "status": "ADVERTISED",
    "message": {
        "prefix": "10.24.34.0/32",
        "asPath": "65000",
        "med": 100,
        "nextHopIp": "10.24.34.1"
    }
}

{
```

```
    "resource_arn": "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0",
    "event_timestamp": 1746643490000,
    "type": "BGPStatus",
    "status": "UP",
    "message": null
}
```

위치:

- `resource_arn`는 라우팅 서버 피어의 ARN입니다.
- `event_timestamp`는 이벤트의 타임스탬프입니다.
- 생산되는 로그 이벤트의 `type`입니다(`RouteStatus`, `BGPStatus`, `BFDStatus`).
- `status` 필드는 상태 업데이트입니다.
  - `RouteStatus` 유형 메시지의 경우
    - `ADVERTISED`(피어가 라우팅을 알림)
    - `UPDATED`(피어가 기존 라우팅을 업데이트함)
    - `WITHDRAWN`(피어에 의해 라우팅이 철회됨)
  - `BFDStatus` 및 `BGPStatus`에 대한 업데이트의 경우
    - `UP`, `DOWN`.
- `message` 필드는 현재 RouteStatus 메시지 유형의 라우팅 속성에만 사용되지만 모든 유형의 관련 정보로 채워질 수 있습니다.

AWS Management Console

라우팅 서버 피어 로그 생성:

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 가상 프라이빗 클라우드에서 라우팅 서버를 선택합니다.
3. 라우팅 서버 페이지에서 라우팅 서버 피어을 선택합니다.
4. 로그 전송 탭을 선택합니다.
5. 로그 전송 추가를 선택합니다.
6. 목적지를 선택하고 설정을 구성합니다.
   - Amazon CloudWatch Logs

- 로그 유형: 전송할 로그 유형입니다. 지원되는 로그 유형은 EVENT_LOGS가 유일합니다.
- 목적지 로그 그룹: 로그가 전송될 CloudWatch 로그 그룹입니다. 기존의 로그 그룹을 선택하거나 새 로그 그룹을 생성할 수 있습니다(예: /aws/vpc/route-server-peers).
- 필드 선택: 로그에 포함할 데이터 필드입니다.
- 출력 형식: 로그의 형식을 다음 중에서 지정합니다.
  - JSON: 컴퓨터 처리용으로 구조화된 형식
  - 텍스트: 일반 텍스트 형식
- 필드 구분 기호: 텍스트 형식을 사용할 때 필드 구분에 사용하는 문자입니다(예: 쉼표, 탭, 공백).

- Amazon S3
  - 교차 계정 - 다른 AWS 계정으로 로그 전송
    - 로그 유형: 전송할 로그 유형입니다. 지원되는 로그 유형은 EVENT_LOGS가 유일합니다.
    - 전송 목적지 ARN: 로그가 전송될 다른 AWS 계정에 있는 S3 버킷의 Amazon 리소스 이름입니다.
    - 필드 선택: 로그에 포함할 데이터 필드입니다.
    - 접미사: 로그 파일 이름 끝에 추가되는 글자입니다(예: .log, .txt).
    - Hive 호환: 이 기능을 켜면 Hive 기반 도구와 함께 작동하는 폴더 구조로 로그를 구성하여 Amazon Athena와 같은 서비스를 더 쉽게 검색할 수 있습니다.
    - 필드 구분 기호: 텍스트 형식을 사용할 때 필드 구분에 사용하는 문자입니다.
  - 현재 계정
    - 로그 유형: 전송할 로그 유형입니다. 지원되는 로그 유형은 EVENT_LOGS가 유일합니다.
    - 목적지 S3 버킷: 로그가 전송되는 계정의 S3 버킷입니다. 하위 폴더 경로를 지정할 수 있습니다.
    - 필드 선택: 로그에 포함할 데이터 필드입니다.
    - 접미사: 로그 파일 이름 끝에 추가되는 글자입니다(예: .log, .txt).
    - Hive 호환: 이 기능을 켜면 Hive 기반 도구와 함께 작동하는 폴더 구조로 로그를 구성하여 검색이 더 쉬워집니다.
    - 필드 구분 기호: 텍스트 형식을 사용할 때 필드를 구분하는 문자입니다.
- Amazon Data Firehose
  - 교차 계정

- 로그 유형: 전송할 로그 유형입니다. 지원되는 로그 유형은 EVENT_LOGS가 유일합니다.
- 전송 목적지 ARN: 다른 AWS 계정에 있는 Firehose 전송 스트림의 Amazon 리소스 이름입니다.
- 필드 선택: 로그에 포함할 데이터 필드입니다.
- 필드 구분 기호: 텍스트 형식을 사용할 때 필드를 구분하는 문자입니다.

- 현재 계정
  - 로그 유형: 전송할 로그 유형입니다. 지원되는 로그 유형은 EVENT_LOGS가 유일합니다.
  - 전송 목적지 스트림: 로그가 전송되는 계정의 Firehose 전송 스트림입니다. 스트림은 'Direct Put' 소스 유형을 사용해야 합니다.
  - 필드 선택: 로그에 포함되는 데이터 필드입니다.
  - 출력 형식: 로그의 형식을 다음 중에서 지정합니다.
    - JSON: 컴퓨터 처리용으로 구조화된 형식
    - 텍스트: 일반 텍스트 형식
  - 필드 구분 기호: 텍스트 형식을 사용할 때 필드를 구분하는 문자입니다.

Command line

이 섹션의 명령은 AWS CLI 명령 참조로 연결됩니다. 이 설명서에서는 명령을 실행할 때 사용할 수 있는 옵션에 대한 자세한 설명을 제공합니다.

라우팅 서버 피어 로그 생성:

1. put-delivery-source 명령을 사용합니다.
   - 요청 예

```
aws logs put-delivery-source --name "source-rsp-1234567890abcdef0" --
resource-arn "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0" --log-type "EVENT_LOGS"
```

   - 응답의 예

```
{
    "deliverySource": {
       "name": "source-rsp-1234567890abcdef0",
```

```
        "arn": "arn:aws:logs:us-east-1:111122223333:delivery-source:source-
rsp-1234567890abcdef0",
        "resourceArns": [
            "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0"
        ],
        "service": "ec2",
        "logType": "EVENT_LOGS"
    }
}
```

2. put-delivery-destination 명령을 사용합니다.

   - 다음 AWS CLI 예제에서는 라우팅 서버 로그를 생성합니다. 로그가 지정된 로그 그룹에 전송됩니다.
   - 요청 예

```
aws logs put-delivery-destination --name "destination-rsp-abcdef01234567890"
 --destination-resource-arn "arn:aws:logs:us-east-1:111122223333:log-group:/
aws/vendedlogs/ec2/route-server-peer/EVENT_LOGS/rsp-abcdef01234567890"
```

   - 응답의 예

```
{
     "deliveryDestination": {
        "name": "destination-rsp-abcdef01234567890",
        "arn": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:destination-rsp-abcdef01234567890",
        "deliveryDestinationType": "CWL",
        "deliveryDestinationConfiguration": {
            "destinationResourceArn": "arn:aws:logs:us-
east-1:111122223333:log-group:/aws/vendedlogs/ec2/route-server-peer/
EVENT_LOGS/rsp-abcdef01234567890"
        }
    }
}
```

3. create-delivery 명령을 사용합니다.

   - 요청 예

```
aws logs create-delivery --delivery-source-name "source-rsp-1234567890abcdef0"
 --delivery-destination-arn "arn:aws:logs:us-east-1:111122223333:delivery-
destination:destination-rsp-abcdef01234567890"
```

- 응답의 예

```
{
     "delivery": {
        "id": "1234567890abcdef0",
        "arn": "arn:aws:logs:us-
east-1:111122223333:delivery:1234567890abcdef0",
        "deliverySourceName": "source-rsp-1234567890abcdef0",
        "deliveryDestinationArn": "arn:aws:logs:us-
east-1:111122223333:delivery-destination:destination-rsp-abcdef01234567890",
        "deliveryDestinationType": "CWL",
        "recordFields": [
            "resource_arn",
            "event_timestamp",
            "type",
            "status",
            "message"
        ]
    }
}
```

시작하기 자습서

이 자습서에서는 VPC에서 동적 라우팅을 활성화하기 위한 VPC Route Server 설정 및 구성 프로세스를 안내합니다. 필요한 모든 구성 요소를 생성 및 구성하고, BGP 피어링을 설정하고, 제대로 작동하는지 확인하는 방법을 알아봅니다. 이 자습서에서는 최초 IAM 설정부터 테스트와 정리에 이르기까지 모든 사항을 다룹니다.

이 자습서를 시작하기 전에 확인해야 할 사항:

- AWS 계정에 대한 관리 액세스
- 동적 라우팅을 활성화하려는 서브넷이 2개 이상인 VPC
- BGP를 지원하고 라우팅 서버 피어 디바이스 역할을 할 수 있는 네트워크 디바이스(예: EC2 인스턴스에서 실행되는 방화벽)
- BGP 개념 및 AWS 네트워킹 관련 기본 지식

AWS 관리 콘솔 또는 AWS CLI를 사용하여 단계를 완료할 수 있습니다. 각 단계에서 두 가지 방법 모두가 제공됩니다.

예상 완료 시간: 15~30분

단계

- 1단계: 필요한 IAM 역할 권한 구성
- 2단계: 라우팅 서버 생성
- 3단계: 라우팅 서버와 VPC 연결
- 4단계: 라우팅 서버 엔드포인트 생성
- 5단계: 라우팅 서버 전파 활성화
- 6단계: 라우팅 서버 피어 생성
- 7단계: 디바이스에서 BGP 세션 시작
- 8단계: 정리

1단계: 필요한 IAM 역할 권한 구성

VPC Route Server를 사용하려면 사용 중인 IAM 사용자 또는 역할에 필요한 IAM 권한이 있어야 합니다. 아래에 각 API에 필요한 권한이 나와 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CreateRouteServer",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServer",
            "Effect": "Allow",
            "Action": [
                "sns:DeleteTopic"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateRouteServerEndpoint",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServerEndpoint",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateRouteServerPeer",
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServerPeer",
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

2단계: 라우팅 서버 생성

이 섹션의 단계에 따라 라우팅 서버를 생성합니다.

라우팅 서버 구성 요소는 VPC 및 인터넷 게이트웨이 라우팅 테이블을 FIB(Forwarding Information Base)의 IPv4 또는 IPv6 경로로 업데이트합니다. 라우팅 서버는 단일 FIB 및 RIB(Routing Information Base)를 나타냅니다.

AWS Management Console

라우팅 서버 생성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 가상 프라이빗 클라우드에서 라우팅 서버를 선택합니다.
3. 라우팅 서버 페이지에서 라우팅 서버 생성을 선택합니다.
4. 라우팅 서버 생성 페이지에서 다음 설정을 구성합니다.
   - 이름에 라우팅 서버의 이름(예: “my-route-server-01”)을 입력합니다. 이름은 255자 이하여야 합니다.
   - Amazon 측 ASN에 BGP ASN 값을 입력합니다. 값은 1~4294967295 범위에 있어야 합니다. 64512~65534(16비트 ASN) 또는 4200000000~4294967294(32비트 ASN) 범위에서 프라이빗 ASN을 사용하는 것이 좋습니다.
   - 경로 지속에 대해 활성화 또는 비활성화를 선택합니다. 이 옵션은 모든 BGP 세션이 종료된 후 경로가 유지될지 여부를 결정합니다.
     - 활성화된 경우: 모든 BGP 세션이 종료되어도 경로가 라우팅 서버의 라우팅 데이터베이스에 보존됩니다.
     - 비활성화된 경우: 모든 BGP 세션이 종료되면 라우팅 데이터베이스에서 경로가 제거됩니다.
   - 경로 지속을 활성화한 경우 지속 시간에 1~5분 사이의 값을 입력합니다. 이 지속 시간으로 BGP가 다시 설정된 후 경로의 지속을 취소하기까지 라우팅 서버가 대기하는 시간이 지정됩니다. 예를 들어, 1분으로 설정하면 디바이스가 BGP를 다시 설정하고 1분이 지난 후 경로를 다시 학습하고 알린 다음 라우팅 서버가 정상 기능을 재개합니다. 일반적으로 1분이면 충분

하지만 BGP 네트워크에서 모든 경로를 완전히 다시 설정하고 다시 학습하는 데 더 많은 시간이 필요한 경우 최대 5분을 설정할 수 있습니다.

- (선택 사항) BGP 상태 변경에 대해 SNS 알림을 활성화하려면 SNS 알림 활성화 스위치를 전환하세요. SNS 알림을 활성화하면 라우팅 서버 피어의 BGP 또는 BFD 세션 상태 변경과 라우팅 서버 엔드포인트에 대한 유지 관리 알림이 AWS에서 프로비저닝한 SNS 주제로 유지됩니다. 이러한 알림에 대한 자세한 내용은 아래에 있는 SNS 알림 세부 정보 표를 참조하세요.

5. (선택 사항) 라우팅 서버에 태그를 추가하려면 태그 - 선택 사항 섹션까지 아래로 스크롤하고 새 태그 추가를 선택합니다. 각 태그의 키와 값(선택 사항)을 입력합니다. 최대 50개의 태그를 추가할 수 있습니다.
6. 설정을 검토하고 라우팅 서버 생성을 선택합니다.
7. 라우팅 서버가 생성될 때까지 대기합니다. 완료되면 라우팅 서버 페이지로 리디렉션되고, 새 라우팅 서버가 사용 가능 상태로 나열된 것을 확인할 수 있습니다.

Command line

다음 절차에 따라 VPC에서 동적 라우팅을 관리할 새 라우팅 서버를 생성합니다.

`--amazon-side-asn`의 경우 BGP ASN 값을 입력합니다. 값은 1~4294967295 범위에 있어야 합니다. 64512~65534(16비트 ASN) 또는 4200000000~4294967294(32비트 ASN) 범위에서 프라이빗 ASN을 사용하는 것이 좋습니다.

1. 명령:

```
aws ec2 create-route-server --amazon-side-asn 65000
```

응답:

```
{
    "RouteServer": {
        "RouteServerId": "rs-1",
        "AmazonSideAsn": 65000,
        "State": "pending"
    }
}
```

2. 라우팅 서버를 사용할 수 있을 때까지 대기합니다.

명령:

```
aws ec2 describe-route-servers
```

응답:

```
{
    "RouteServer": {
        "RouteServerId": "rs-1",
        "AmazonSideAsn": 65000,
        "State": "available"
    }
}
```

SNS 알림 세부 정보

다음 표는 Amazon VPC Route Server에서 Amazon SNS를 사용하여 보내는 메시지에 관한 세부 정보가 나와 있습니다.

표준 필드		메시지 속성 (메타데이터)			
메시지	전송되는 시점	timestamp	eventCode	routeServerEndpointId	affectedRouteServerPeerIds
라우팅 서버 엔드포인트 [ENDPOINT ID]은(는) 현재 유지 관리 중입니다. BFD 및 BGP 세션에 영향이 있을 수 있습니다.	라우팅 서버 엔드포인트 유지 관리	형식: 2025-02-17T15:55:00Z	ROUTE_SERVER_ENDPOINT_MAINTENANCE	영향을 받는 엔드포인트 ID	영향을 받는 피어 ID 목록
메시지	전송되는 시점	timestamp	eventCode	routeServerPeerId	newBgpStatus

표준 필드		메시지 속성 (메타데이터)			
BGP for Route Server Peer [PEER ID] is now [UP/DOWN].	라우팅 서버 피어 BGP 상태 변경	형식: 2025-02-17T15:55:00Z	ROUTE_SERVER_PEER_BGP_STATUS_CHANGE	영향을 받는 피어 ID	UP 또는 DOWN
메시지	전송되는 시점	timestamp	eventCode	routeServerPeerId	newBfdStatus
BFD for Route Server Peer [PEER ID] is now [UP/DOWN].	라우팅 서버 피어 BFD 상태 변경	형식: 2025-02-17T15:55:00Z	ROUTE_SERVER_PEER_BFD_STATUS_CHANGE	영향을 받는 피어 ID	UP 또는 DOWN

3단계: 라우팅 서버와 VPC 연결

이 섹션의 단계를 완료하여 라우팅 서버와 VPC를 연결합니다.

라우팅 서버 연결은 라우팅 서버와 VPC 사이에 설정된 연결입니다. 이는 라우팅 서버가 VPC의 어플라이언스와 함께 작동하는 데 있어 기본 구성 단계입니다.

라우팅 서버 연결을 생성하는 경우:

- 라우팅 서버를 특정 VPC에 연결합니다.
- 이 기능을 통해 Route Server가 VPC 서브넷 내의 라우팅 테이블과 상호 작용할 수 있습니다.
- 라우팅 서버가 연결된 VPC 내에서 경로를 수신하고 전파하도록 허용합니다.
- 라우팅 서버가 작동할 수 있는 범위를 설정합니다.

라우팅 서버 연결의 주요 측면:

- 각 라우팅 서버는 1개의 VPC와 연결할 수 있습니다. 각 VPC는 기본적으로 최대 5개의 개별 라우팅 서버 연결을 보유할 수 있습니다. 할당량에 대한 자세한 내용은 라우팅 서버 할당량을 참조하세요.
- 라우팅 서버가 경로를 관리하려면 먼저 연결을 생성해야 합니다.

- 연결을 모니터링하여 상태(예: 연결 중 및 연결됨)를 추적할 수 있습니다.
- 라우팅 서버가 더 이상 해당 VPC에서 작동하지 않도록 하려면 연결을 제거(연결 해제)할 수 있습니다.

AWS Management Console

라우팅 서버와 VPC 연결

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 가상 프라이빗 클라우드에서 라우팅 서버를 선택합니다.
3. VPC와 연결할 라우팅 서버를 선택합니다.
4. 연결 탭에서 라우팅 서버 연결을 선택합니다.
5. 라우팅 서버 연결 대화 상자에서:
   - 라우팅 서버 ID 필드에는 선택한 라우팅 서버가 자동으로 채워집니다.
   - VPC ID의 경우 드롭다운 목록에서 연결할 VPC를 선택합니다.
6. 라우팅 서버 연결을 선택합니다.
7. 연결이 완료될 때까지 대기합니다. 완료되면 연결 탭에 상태가 연결됨으로 표시됩니다.

Command line

다음 절차를 사용하여 라우팅 서버와 VPC를 연결합니다.

1. 명령:

```
aws ec2 associate-route-server --route-server-id rs-1 --vpc-id vpc-1
```

응답:

```
{
    "RouteServerAssociation": {
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "State": "associating"
    }
}
```

2. 연결이 완료될 때까지 대기합니다.

명령:

```
aws ec2 get-route-server-associations --route-server-id rs-1
```

응답:

```
{
    "RouteServerAssociation": {
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "State": "associated"
    }
}
```

4단계: 라우팅 서버 엔드포인트 생성

이 섹션의 단계를 완료하여 라우팅 서버 엔드포인트를 생성합니다. 중복성을 위해 서브넷당 2개의 엔드포인트를 생성합니다.

Route Server 엔드포인트는 라우팅 서버와 BGP 피어 간의 BGP(Border Gateway Protocol) 연결을 지원하는 서브넷 내부의 AWS 관리형 구성 요소입니다.

라우팅 서버 엔드포인트는 네트워크 디바이스가 라우팅 서버와의 BGP 세션을 설정하는 "접점" 역할을 합니다. 실제로 BGP 연결을 처리하는 구성 요소인 반면 라우팅 서버 자체는 라우팅 결정 및 라우팅 전파를 관리합니다.

> **Note**
>
> 라우팅 서버 엔드포인트에는 시간당 0.75 USD의 요금이 부과됩니다.

AWS Management Console

라우팅 서버 엔드포인트 생성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 가상 프라이빗 클라우드에서 라우팅 서버를 선택합니다.
3. 엔드포인트를 생성할 라우팅 서버를 선택합니다.

4. 하단 창에서 라우팅 서버 엔드포인트 탭을 선택합니다.
5. 라우팅 서버 엔드포인트 생성을 선택합니다.
6. 라우팅 서버 엔드포인트 생성 페이지에서 다음 설정을 구성합니다.
   - 이름에 엔드포인트를 설명하는 이름을 입력합니다.
   - 라우팅 서버에서 올바른 라우팅 서버를 선택했는지 확인합니다.
   - 서브넷에서 엔드포인트를 생성할 서브넷을 선택합니다.
7. (선택 사항) 라우팅 서버 엔드포인트에 태그를 추가하려면 태그 - 선택 사항 섹션까지 아래로 스크롤하고 새 태그 추가를 선택합니다. 각 태그의 키와 값(선택 사항)을 입력합니다.
8. 설정을 검토하고 라우팅 서버 엔드포인트 생성을 선택합니다.
9. 엔드포인트가 생성될 때까지 대기합니다. 완료되면 성공 메시지가 표시됩니다.
10. 5~9단계를 반복하여 동일한 서브넷에 이름이 다른 두 번째 엔드포인트를 생성합니다.
11. 라우팅 서버 엔드포인트가 필요한 각 서브넷에 대해 5~10단계를 반복합니다.
12. 엔드포인트를 생성한 후 라우팅 서버의 라우팅 서버 엔드포인트 탭으로 돌아갑니다.
13. 각 서브넷에 대해 2개의 엔드포인트가 나열되어 있는지 확인합니다.
14. 각 엔드포인트의 상태가 사용 가능인지 확인합니다.

Command line

다음 절차에 따라 라우팅 서버 엔드포인트를 생성합니다.

1. 명령:

```
aws ec2 create-route-server-endpoint --route-server-id rs-1 --subnet-id subnet-1
```

응답:

```
{
    "RouteServerEndpoint": {
        "RouteServerId": "rs-1",
        "RouteServerEndpointId": "rse-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "pending"
    }
}
```

2. 생성 후 엔드포인트를 완전히 사용할 수 있을 때까지 몇 분 정도 대기해야 할 수 있습니다.

   명령:

   ```
   aws ec2 describe-route-server-endpoints
   ```

   응답:

   ```
   {
       "RouteServerEndpoint": {
           "RouteServerId": "rs-1",
           "RouteServerEndpointId": "rse-1",
           "VpcId": "vpc-1",
           "SubnetId": "subnet-1",
           "EniId": "eni-123",
           "EniAddress": "10.1.2.3",
           "State": "available"
       }
   }
   ```

   단계를 반복하여 동일한 서브넷에 이름이 다른 두 번째 엔드포인트를 생성하고 라우팅 서버 엔드포인트가 필요한 각 서브넷에 대해 엔드포인트를 생성합니다.

5단계: 라우팅 서버 전파 활성화

이 단계를 완료하여 라우팅 서버 전파를 활성화합니다.

활성화되면 지정한 라우팅 테이블의 FIB에 경로가 설치됩니다. 라우팅 서버는 IPv4 및 IPv6 경로 전파를 지원합니다.

라우팅 서버 전파는 라우팅 테이블 업데이트를 자동화하는 메커니즘입니다. 라우팅 테이블을 수동으로 업데이트하는 대신 라우팅 서버는 FIB의 경로를 사용하여 구성된 라우팅 테이블에 적절한 경로를 자동으로 전파합니다.

라우팅 서버 전파의 주요 측면:

- 구성
  - 라우팅 서버를 특정 라우팅 테이블에 연결
  - 동적 경로 업데이트를 수신할 라우팅 테이블 결정

  - 라우팅 테이블별로 활성화 또는 비활성화 가능
- 기능
  - BGP 피어에서 학습한 경로로 라우팅 테이블 자동 업데이트
  - BGP 속성을 기반으로 사용 가능한 최상의 경로 전파
  - 지정된 라우팅 테이블에서 경로 일관성 유지
  - 네트워크 조건이 변경될 때 동적으로 경로 업데이트
- 상태
  - 활성화 가능(경로가 전파됨)
  - 비활성화 가능(경로가 전파되지 않음)

AWS Management Console

라우팅 서버 전파 활성화

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 전파를 활성화할 라우팅 서버를 선택합니다.
3. 라우팅 서버 세부 정보 패널에서 전파 탭을 선택합니다.
4. 전파 활성화를 선택합니다.
5. 전파 활성화 대화 상자에서:
   - 라우팅 서버 ID는 미리 채워집니다.
   - 라우팅 테이블의 드롭다운 메뉴에서 새로 전파된 경로의 대상 라우팅 테이블을 선택합니다.
6. 전파 활성화를 선택하여 확인합니다.
7. 전파 목록에서 전파 상태가 사용 가능으로 변경될 때까지 대기합니다.
8. 선택한 라우팅 테이블이 전파 목록에 사용 가능 상태로 표시되는지 확인합니다.

Command line

다음 절차에 따라 라우팅 서버 전파를 활성화합니다.

1. 명령:

```
aws ec2 enable-route-server-propagation --route-table-id rtb-1 --route-server-id
 rs-1
```

응답:

```
{
    "RouteServerRoutePropagation": {
        "RouteServerId": "rs-1",
        "RouteTableId": "rtb-1",
        "State": "pending"
    }
}
```

2. 전파 상태가 사용 가능으로 변경될 때까지 대기합니다.

명령:

```
aws ec2 get-route-server-propagations --route-server-id rs-1
```

응답:

```
{
    "RouteServerRoutePropagation": {
        "RouteServerId": "rs-1",
        "RouteTableId": "rtb-1",
        "State": "available"
    }
}
```

6단계: 라우팅 서버 피어 생성

라우팅 서버 피어는 라우팅 서버 엔드포인트와에 AWS에 배포된 디바이스(예: EC2 인스턴스에서 실행되는 방화벽 어플라이언스 또는 기타 네트워크 보안 기능) 사이의 세션입니다. 디바이스는 다음 요구 사항을 충족해야 합니다.

- VPC에 탄력적 네트워크 인터페이스 보유
- BGP(Border Gateway Protocol) 지원
- BGP 세션 시작 가능

> **Note**
>
> 중복성을 위해 라우팅 서버 엔드포인트당 1개의 라우팅 서버 피어를 생성하는 것이 좋습니다.

AWS Management Console

라우팅 서버 피어 생성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 경로에서 VPC > 라우팅 서버 피어 > 라우팅 서버 피어 생성을 선택합니다.
3. 세부 정보에서 다음을 구성합니다.
   - 이름: 라우팅 서버 피어의 이름을 입력합니다(최대 255자). 예: my-route-server-peer-01
   - 라우팅 서버 엔드포인트 ID: 드롭다운에서 라우팅 서버 엔드포인트를 선택합니다. 라우팅 서버 엔드포인트 생성을 선택하여 새 엔드포인트를 생성할 수도 있습니다.
   - 피어 주소: 피어의 IPv4 주소를 입력합니다. 유효한 IP 주소여야 합니다. 피어 주소는 라우팅 서버 엔드포인트에서 연결할 수 있어야 합니다.
   - 피어 ASN: BGP 피어의 ASN(자율 시스템 번호)을 입력합니다. 값은 1~4294967295 범위에 있어야 합니다. ASN은 일반적으로 프라이빗 범위(16비트의 경우 64512~65534 또는 32비트의 경우 4200000000~4294967294)를 사용해야 합니다.
   - 피어 활성 감지:
     - BGP 연결 유지(기본값): 표준 BGP 연결 유지 메커니즘
     - BFD: 더 빠른 장애 조치를 위한 Bidirectional Forwarding Detection
   - (선택 사항) 태그에서 새 태그 추가를 선택하여 키-값 페어 태그를 추가합니다. 태그는 AWS 리소스를 식별하고 추적하는 데 도움이 됩니다.
4. 설정을 검토하고 라우팅 서버 피어 생성을 선택합니다.

Command line

다음 절차에 따라 라우팅 서버 피어를 생성합니다.

1. 명령:

```
aws ec2 create-route-server-peer --route-server-endpoint-id rse-1 --peer-address
 10.0.2.3 --bgp-options PeerAsn=65001,PeerLivenessDetection=bfd
```

응답:

응답에서 상태 값은 pending|available|deleting|deleted일 수 있습니다.

```
{
    "RouteServerPeer": {
        "RouteServerPeerId": "rsp-1",
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "pending",
        "EndpointEniId": "eni-2,
        "EndpointEniAddress": "10.0.2.4",
        "PeerEniId": "eni-1",
        "PeerAddress": "10.0.2.3",
        "BgpOptions": {
            "PeerAsn": 65001,
   "PeerLivenessDetection": "bfd"
        },
        "BgpStatus": {
            "Status": "Up"
        }
    }
}
```

2. 전파 상태가 사용 가능으로 변경될 때까지 대기합니다.

명령:

```
aws ec2 describe-route-server-peers
```

응답:

```
{
    "RouteServerPeer": {
        "RouteServerPeerId": "rsp-1",
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "available",
        "EndpointEniId": "eni-2,
        "EndpointEniAddress": "10.0.2.4",
```

```
        "PeerEniId": "eni-1",
        "PeerAddress": "10.0.2.3",
        "BgpOptions": {
            "PeerAsn": 65001,
   "PeerLivenessDetection": "bfd"
        },
        "BgpStatus": {
            "Status": "down"
        }
    }
}
```

7단계: 디바이스에서 BGP 세션 시작

라우팅 서버 피어의 상태가 사용 가능인 경우 라우팅 서버 엔드포인트로 BGP 세션을 시작하도록 워크로드를 구성합니다.

서브넷의 디바이스에서 BGP 세션을 시작하는 것은 이 가이드의 범위를 벗어납니다. 라우팅 서버 엔드포인트는 BGP 세션을 시작하지 않습니다.

라우팅 테이블에 라우팅 서버에서 전파한 최상의 경로가 포함되어 있는지 확인하여 VPC Route Server 기능이 작동하는지 확인할 수 있습니다.

8단계: 정리

자습서의 빌드 부분이 완료되었습니다. 이 섹션의 단계를 완료하여 생성한 VPC Route Server 구성 요소를 제거합니다.

7.1: 디바이스에서 BGP 알림 철회

서브넷의 디바이스에서 BGP 알림을 철회하는 것은 이 가이드의 범위를 벗어납니다. 필요에 따라 BGP 구성은 타사 공급업체에 문의하세요.

7.2: 라우팅 서버 전파 비활성화

다음 절차에 따라 라우팅 서버 전파를 비활성화합니다.

AWS Management Console

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

2. 전파를 비활성화할 라우팅 서버를 선택합니다.
3. 작업 > 라우팅 서버 수정을 선택합니다.
4. 라우팅 서버 세부 정보 패널에서 전파 탭을 선택합니다.
5. 비활성화하려는 전파를 선택한 다음 전파 비활성화를 선택합니다.
6. 대화 상자에서 라우팅 서버 전파 비활성화를 선택합니다.

Command line

1. 전파 비활성화:

```
aws ec2 disable-route-server-route-propagation --route-table-id rtb-1 --route-server-id rs-1
```

2. 전파가 삭제되었는지 확인:

```
aws ec2 get-route-server-route-propagations --route-server-id rs-1 [--route-table-id rtb-1]
```

7.3: 라우팅 서버 피어 삭제

다음 절차에 따라 라우팅 서버 피어를 삭제합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 경로에서 라우팅 서버 > 라우팅 서버 피어를 선택합니다.
3. 라우팅 서버 피어를 선택합니다.
4. 작업 > 라우팅 서버 피어 삭제를 선택합니다.

Command line

1. 피어 삭제:

```
aws ec2 delete-route-server-peer --route-server-peer-id rsp-1
```

2. 삭제 확인:

```
aws ec2 describe-route-server-peers [--route-server-peer-ids rsp-1] [--filters
 Key=RouteServerId|RouteServerEndpointId|VpcId]
```

7.4: 라우팅 서버 엔드포인트 삭제

다음 절차에 따라 라우팅 서버 엔드포인트를 삭제합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 엔드포인트를 삭제할 라우팅 서버를 선택합니다.
3. 라우팅 서버 엔드포인트를 선택합니다.
4. 엔드포인트를 선택하고 작업 > 라우팅 서버 엔드포인트 삭제를 선택합니다.
5. 삭제를 입력하고 삭제를 선택합니다.

Command line

1. 엔드포인트 설명:

```
aws ec2 describe-route-server-endpoints
```

2. 라우팅 서버 엔드포인트 삭제:

```
aws ec2 delete-route-server-endpoint --route-server-endpoint-id rse-1
```

3. 엔드포인트가 삭제되었는지 확인:

```
aws ec2 describe-route-server-endpoints [--route-server-endpoint-ids rsp-1] [--
filters Key=RouteServerId|VpcId|SubnetId]
```

7.5: VPC에서 라우팅 서버 연결 해제

다음 절차에 따라 VPC에서 라우팅 서버 연결을 해제합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 연결을 해제할 라우팅 서버를 선택합니다.
3. 연결을 선택합니다.
4. 라우팅 서버 연결 해제를 선택합니다.
5. 변경할 사항을 확인하고 라우팅 서버 연결 해제를 선택합니다.

Command line

1. VPC에서 라우팅 서버 연결 해제:

```
aws ec2 disassociate-route-server --route-server-id rs-1 --vpc-id vpc-1
```

2. 연결 해제 확인:

```
aws ec2 get-route-server-associations --route-server-id rs-1
```

7.6 라우팅 서버 삭제

다음 절차에 따라 라우팅 서버를 삭제합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 삭제할 라우팅 서버를 선택합니다.
3. 작업 > 라우팅 서버 삭제를 선택합니다.
4. 삭제를 입력하고 삭제를 선택합니다.

Command line

1. 라우팅 서버 삭제:

```
aws ec2 delete-route-server --route-server-id rs-1
```

2. 삭제 확인:

```
aws ec2 describe-route-servers [--route-server-ids rs-1] [--filters Key=VpcId]
```

Amazon VPC Route Server 자습서가 완료되었습니다.

VPC의 연결성 문제 해결

Reachability Analyzer는 정적 구성 분석 도구입니다. Reachability Analyzer를 사용하여 VPC의 두 리소스 간 네트워크 연결성을 분석하고 디버깅할 수 있습니다. Reachability Analyzer에서는 연결할 수 있는 경우 이러한 리소스 간 가상 경로에 대한 홉별 세부 정보가 생성되고, 그렇지 않다면 차단 구성 요소가 식별됩니다. 예를 들면 누락되거나 잘못 구성된 라우팅 테이블 경로가 식별될 수 있습니다.

자세한 내용은 Reachability Analyzer 사용 설명서를 참조하십시오.

미들박스 라우팅 마법사

트래픽을 보안 어플라이언스로 리디렉션하는 것과 같이 VPC로 들어오고 나가는 트래픽 라우팅 경로에 세분화된 제어를 구성하려는 경우 VPC 콘솔에서 미들박스 라우팅 마법사를 사용할 수 있습니다. 미들박스 라우팅 마법사는 필요에 따라 트래픽을 리디렉션하는 데 필요한 라우팅 테이블과 라우팅(홉)을 자동으로 생성하도록 도와줍니다.

미들박스 라우팅 마법사는 다음과 같은 시나리오에 대한 라우팅을 구성하는 데 도움을 줄 수 있습니다.

- 미들박스 어플라이언스(예: 보안 어플라이언스로 구성된 Amazon EC2 인스턴스)로 트래픽을 라우팅합니다.
- Gateway Load Balancer 엔드포인트로 라우팅합니다. 자세한 내용은 로드 밸런서 게이트웨이 사용 설명서를 참조하십시오.

자세한 내용은 the section called “미들박스 시나리오” 섹션을 참조하세요.

내용

- 미들박스 라우팅 마법사 사전 조건
- 보안 어플라이언스로 VPC 트래픽 리디렉션
- 미들박스 라우팅 마법사 고려 사항
- 미들박스 시나리오

미들박스 라우팅 마법사 사전 조건

the section called “미들박스 라우팅 마법사 고려 사항” 섹션을 검토합니다. 그런 다음 미들박스 라우팅 마법사를 사용하기 전에 다음 정보가 있는지 확인합니다.

- VPC.
- 인터넷 게이트웨이, 가상 프라이빗 게이트웨이 또는 네트워크 인터페이스와 같은 VPC에서 트래픽이 발생하거나 트래픽이 시작되는 리소스입니다.
- 미들박스 네트워크 인터페이스 또는 Gateway Load Balancer 엔드포인트
- 트래픽에 대한 대상 서브넷입니다.

보안 어플라이언스로 VPC 트래픽 리디렉션

미들박스 라우팅 마법사는 Amazon VPC 콘솔에서 사용할 수 있습니다.

내용

- 1. 미들박스 라우팅 마법사를 사용하여 경로 생성
- 2. 미들박스 경로 수정
- 3. 미들박스 라우팅 마법사 구성 삭제

1. 미들박스 라우팅 마법사를 사용하여 경로 생성

미들박스 라우팅 마법사를 사용하여 경로를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC 선택한 다음작업(Actions), 미들박스 경로 관리(Manage middlebox routes)를 선택합니다.
4. 경로 생성(Create route)을 선택합니다.
5. 세부 정보 지정(Specify routes) 페이지에서 다음 작업을 수행합니다.
   - 소스(Source)에서 트래픽의 소스를 선택합니다. 가상 프라이빗 게이트웨이를 선택한 경우 대상 IPv4 CIDR(Destination IPv4 CIDR)에 가상 프라이빗 게이트웨이에서 VPC로 들어오는 온프레미스 트래픽에 대한 CIDR을 입력합니다.
   - 미들박스(Middlebox)에서 미들박스 어플라이언스와 연결된 네트워크 인터페이스 ID를 선택하거나 Gateway Load Balancer 엔드포인트를 사용하는 경우 VPC 엔드포인트 ID를 선택합니다.

- 대상 서브넷(Destination subnet)에서 대상 서브넷을 선택합니다.

6. (선택 사항) 다른 대상 서브넷을 추가하려면 서브넷 추가(Add additional subnet)를 선택하고 다음 중 하나를 수행합니다.

   - 미들박스(Middlebox)에서 미들박스 어플라이언스와 연결된 네트워크 인터페이스 ID를 선택하거나 Gateway Load Balancer 엔드포인트를 사용하는 경우 VPC 엔드포인트 ID를 선택합니다.

     여러 서브넷에 대해 동일한 미들박스 어플라이언스를 사용해야 합니다.
   - 대상 서브넷(Destination subnet)에서 대상 서브넷을 선택합니다.

7. (선택 사항) 다른 소스를 추가하려면 소스 추가(Add source)를 선택한 다음 이전 단계를 반복합니다.
8. 다음을 선택합니다.
9. 검토 및 생성(Review and create) 페이지에서 경로를 확인한 다음 경로 생성(Create routes)을 선택합니다.

2. 미들박스 경로 수정

게이트웨이, 미들박스 또는 대상 서브넷을 변경하여 경로 구성을 편집할 수 있습니다.

수정 사항이 있으면 미들박스 라우팅 마법사는 다음의 작업을 자동으로 수행합니다.

- 게이트웨이, 미들박스 및 대상 서브넷에 대한 새 라우팅 테이블을 생성합니다.
- 필요한 경로를 새 라우팅 테이블에 추가합니다.
- 미들박스 라우팅 마법사가 리소스와 연결된 현재 라우팅 테이블을 연결 해제합니다.
- 미들박스 라우팅 마법사가 리소스를 사용해 생성한 새 라우팅 테이블을 연결합니다.

미들박스 라우팅 마법사를 사용하여 미들박스 경로를 수정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC 선택한 다음작업(Actions), 미들박스 경로 관리(Manage middlebox routes)를 선택합니다.
4. 라우팅 편집(Edit routes)을 선택합니다.
5. 게이트웨이를 변경하려면 소스(Source)에서 트래픽이 VPC로 들어가는 게이트웨이를 선택합니다. 가상 프라이빗 게이트웨이를 선택한 경우 대상 IPv4 CIDR(Destination IPv4 CIDR)에 대상 서브넷 CIDR을 입력합니다.

6. 다른 대상 서브넷을 추가하려면 서브넷 추가(Add additional subnet)를 선택하고 다음 중 하나를 수행합니다.

   - 미들박스(Middlebox)에서 미들박스 어플라이언스와 연결된 네트워크 인터페이스 ID를 선택하거나 Gateway Load Balancer 엔드포인트를 사용하는 경우 VPC 엔드포인트 ID를 선택합니다.

     여러 서브넷에 대해 동일한 미들박스 어플라이언스를 사용해야 합니다.
   - 대상 서브넷(Destination subnet)에서 대상 서브넷을 선택합니다.
7. 다음을 선택합니다.
8. 검토 및 업데이트(Review and update) 페이지에서 미들박스 라우팅 마법사에 의해 생성될 라우팅 테이블 및 경로 목록이 표시됩니다. 경로를 확인한 다음 확인 대화 상자에서 경로 업데이트(Update routes)를 선택합니다.

3. 미들박스 라우팅 마법사 구성 삭제

미들박스 라우팅 마법사 구성을 더 이상 사용하지 않기로 결정한 경우 라우팅 테이블을 수동으로 삭제해야 합니다.

미들박스 라우팅 마법사 구성을 삭제하려면

1. 미들박스 라우팅 마법사 라우팅 테이블을 봅니다.

   작업을 수행하고 나면 미들박스 라우팅 마법사가 생성한 라우팅 테이블이 별도의 라우팅 테이블 페이지에 표시됩니다.
2. 표시된 각 라우팅 테이블을 삭제합니다.

미들박스 라우팅 마법사 고려 사항

미들박스 라우팅 마법사를 사용할 때는 다음 사항을 고려합니다.

- 트래픽을 검사하려면 소스로 인터넷 게이트웨이 또는 가상 프라이빗 게이트웨이를 사용할 수 있습니다.
- 동일한 VPC 내의 여러 미들박스 구성에서 동일한 미들박스를 사용하는 경우 미들박스가 두 서브넷에 대해 동일한 홉 위치에 있는지 확인합니다.
- 어플라이언스는 원본 또는 대상 서브넷과는 별도의 서브넷에 구성되어야 합니다.
- 어플라이언스에서 원본/대상 확인을 비활성화해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 소스 또는 대상 변경 확인을 참조하세요.

- 미들박스 라우팅 마법사가 생성하는 라우팅 테이블과 경로는 할당량에 포함됩니다. 자세한 내용은 the section called “라우팅 테이블” 섹션을 참조하세요.
- 리소스(예: 네트워크 인터페이스)를 삭제하면 리소스와 라우팅 테이블과의 연결이 제거됩니다. 리소스가 대상이면 경로 대상이 블랙홀로 설정됩니다. 라우팅 테이블은 삭제되지 않습니다.
- 미들박스 서브넷 및 대상 서브넷은 기본이 아닌 라우팅 테이블과 연결되어야 합니다.

> **Note**
>
> 미들박스 라우팅 마법사를 사용하여 생성한 라우팅 테이블을 수정하거나 삭제할 때에는 미들박스 라우팅 마법사를 사용하는 것이 좋습니다.

- 미들박스 라우팅을 사용하여 보안 어플라이언스를 통해 라우팅하는 경우 검사 후 소스와 최종 대상 간에 보안 그룹 참조는 지원되지 않습니다.

미들박스 시나리오

Amazon Virtual Private Cloud(VPC)는 가상 네트워크 내에서 트래픽의 라우팅을 사용자 지정하고 제어할 수 있는 다양한 네트워킹 기능을 제공합니다. 이러한 기능 중 하나인 미들박스 라우팅 마법사를 사용하여 VPC로 들어오고 나가는 트래픽의 라우팅 경로를 세밀하게 제어할 수 있습니다.

검사, 모니터링 또는 최적화 목적으로 트래픽을 보안 어플라이언스, 로드 밸런서 또는 기타 네트워크 디바이스로 리디렉션해야 하는 경우 미들박스 라우팅 마법사를 사용하여 프로세스를 간소화할 수 있습니다. 이 마법사는 필요한 라우팅 테이블과 경로(홉)를 자동으로 생성하여 필요에 따라 지정된 트래픽을 리디렉션하므로 복잡한 라우팅 구성을 설정하기 위한 수동 작업이 필요하지 않습니다.

미들박스 라우팅 마법사는 여러 시나리오를 지원합니다. 예를 들어, 이를 사용하여 특정 서브넷으로 향하는 트래픽을 검사하거나, 전체 VPC에 대한 미들박스 트래픽 라우팅 및 검사를 구성하거나, 특정 서브넷 간의 트래픽을 선택적으로 검사할 수 있습니다. 이렇게 세분화된 트래픽 라우팅 제어를 통해 고급 보안 정책을 이행하거나, 중앙 집중식 네트워크 모니터링을 활성화하거나, 클라우드 기반 애플리케이션의 성능을 최적화할 수 있습니다.

다음 예에서는 미들박스 라우팅 마법사의 시나리오를 설명합니다.

내용

- 서브넷을 대상으로 하는 트래픽 검사
- VPC에서 미들박스 트래픽 라우팅 및 검사 구성
- 서브넷 간 트래픽 검사

서브넷을 대상으로 하는 트래픽 검사

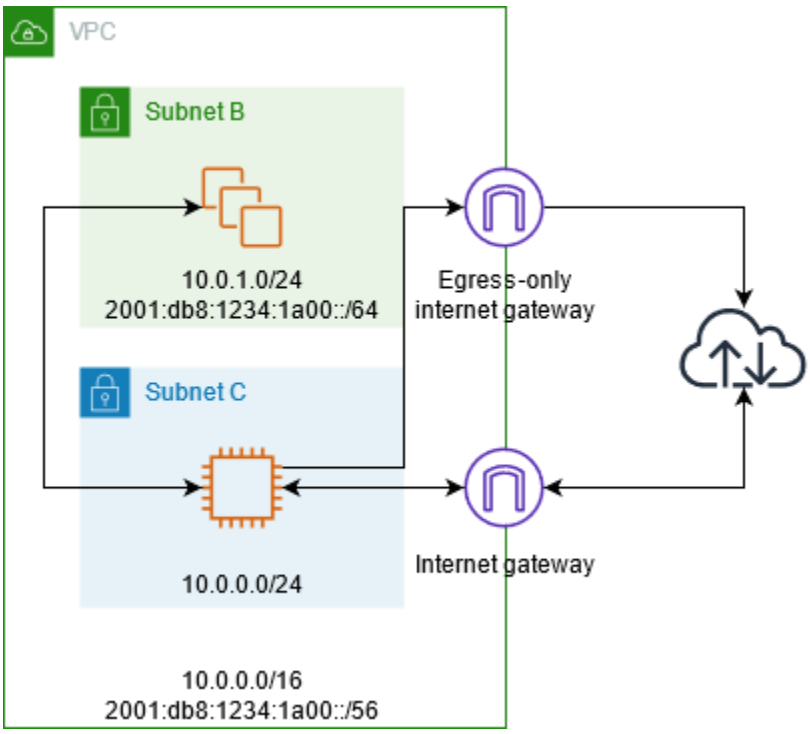
인터넷 게이트웨이를 통해 VPC 로 들어오는 트래픽이 있고, EC2 인스턴스에 설치된 방화벽 어플라이언스를 사용하여 서브넷(서브넷 B로 가정)을 대상으로 하는 모든 트래픽을 검사하려는 시나리오를 생각해봅시다. 방화벽 어플라이언스는 VPC의 서브넷 B와는 별도인 서브넷(서브넷 C로 가정)의 EC2 인스턴스에 설치 및 구성해야 합니다. 그런 다음 미들박스 라우팅 마법사를 사용하여 서브넷 B와 인터넷 게이트웨이 간의 트래픽 경로를 구성할 수 있습니다.

미들박스 라우팅 마법사는 다음의 작업을 자동으로 수행합니다.

- 다음의 라우팅 테이블을 생성합니다.
  - 인터넷 게이트웨이의 라우팅 테이블
  - 대상 서브넷의 라우팅 테이블
  - 미들박스 서브넷의 라우팅 테이블
- 다음 단원에 설명된 내용에 따라 필요한 라우팅을 새 라우팅 테이블에 추가합니다.
- 인터넷 게이트웨이, 서브넷 B 및 서브넷 C와 연결된 현재 라우팅 테이블의 연결을 해제합니다.
- 라우팅 테이블 A를 인터넷 게이트웨이(미들박스 라우팅 마법사의 소스(Source))와 연결하고 라우팅 테이블 C를 서브넷 C(미들박스 라우팅 마법사의 미들박스(Middlebox))와 연결하고 라우팅 테이블 B를 서브넷 B(미들박스 라우팅 마법사의 대상(Destination))과 연결합니다.
- 미들박스 라우팅 마법사에 의해 생성되었음을 나타내는 태그와 생성 날짜를 나타내는 태그를 생성합니다.

미들박스 라우팅 마법사는 기존 라우팅 테이블을 수정하지 않습니다. 새 라우팅 테이블을 생성한 다음 게이트웨이 및 서브넷 리소스와 연결합니다. 리소스가 이미 기존 라우팅 테이블과 명시적으로 연결되어 있는 경우 기존 라우팅 테이블의 연결이 우선적으로 해제된 다음 새 라우팅 테이블이 리소스와 연결됩니다. 기존 라우팅 테이블은 삭제되지 않습니다.

미들박스 라우팅 마법사를 사용하지 않는 경우 서브넷과 인터넷 게이트웨이에 라우팅 테이블을 수동으로 구성한 다음 할당해야 합니다.



인터넷 게이트웨이 라우팅 테이블

인터넷 게이트웨이의 라우팅 테이블에 다음의 경로를 추가합니다.

대상 주소	대상	용도
10.0.0.0/16	로컬	IPv4에 대한 로컬 경로
10.0.1.0/24	*appliance-eni*	서브넷 B를 대상으로 하는 IPv4 트래픽을 미들박스로 라우팅
2001:db8:1234:1a00::/56	로컬	IPv6에 대한 로컬 경로
2001:db8:1234:1a00::/64	*appliance-eni*	서브넷 B를 대상으로 하는 IPv6 트래픽을 미들박스로 라우팅

인터넷 게이트웨이와 VPC 간에는 엣지 연결이 있습니다.

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.

- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

대상 서브넷 라우팅 테이블

대상 서브넷(예제 다이어그램의 서브넷 B)에 대한 라우팅 테이블에 다음 경로를 추가합니다.

대상 주소	대상	용도
10.0.0.0/16	로컬	IPv4에 대한 로컬 경로
0.0.0.0/0	*appliance-eni*	인터넷을 대상으로 하는 IPv4 트래픽을 미들박스로 라우팅
2001:db8:1234:1a00::/56	로컬	IPv6에 대한 로컬 경로
::/0	*appliance-eni*	인터넷을 대상으로 하는 IPv6 트래픽을 미들박스로 라우팅

미들박스 서브넷과의 서브넷 연결이 있습니다.

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

미들박스 서브넷 라우팅 테이블

대상 서브넷(예제 다이어그램의 서브넷 C)에 대한 라우팅 테이블에 다음 경로를 추가합니다.

대상 주소	대상	용도
10.0.0.0/16	로컬	IPv4에 대한 로컬 경로
0.0.0.0/0	*igw-id*	IPv4 트래픽을 인터넷 게이트웨이로 라우팅

대상 주소	대상	용도
2001:db8:1234:1a00::/56	로컬	IPv6에 대한 로컬 경로
::/0	*eigw-id*	송신 전용 인터넷 게이트웨이로 IPv6 트래픽 라우팅

대상 서브넷과의 서브넷 연결이 있습니다.

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

VPC에서 미들박스 트래픽 라우팅 및 검사 구성

Gateway Load Balancer 뒤에 구성된 보안 어플라이언스 플릿을 사용하여 인터넷 게이트웨이에서 VPC로 들어오고 서브넷을 대상으로 하는 트래픽을 검사해야 하는 시나리오를 고려합니다. 서비스 소비자 VPC의 소유자는 VPC의 서브넷에 Gateway Load Balancer 엔드포인트를 생성합니다(엔드포인트 네트워크 인터페이스로 표시됨). 인터넷 게이트웨이를 통해 VPC로 들어오는 모든 트래픽은 먼저 검사할 수 있도록 Gateway Load Balancer 엔드포인트로 라우팅된 후 애플리케이션 서브넷으로 라우팅됩니다. 마찬가지로 애플리케이션 서브넷에서 나가는 모든 트래픽은 검사할 수 있도록 먼저 Gateway Load Balancer 엔드포인트로 라우팅된 후 인터넷으로 라우팅됩니다.

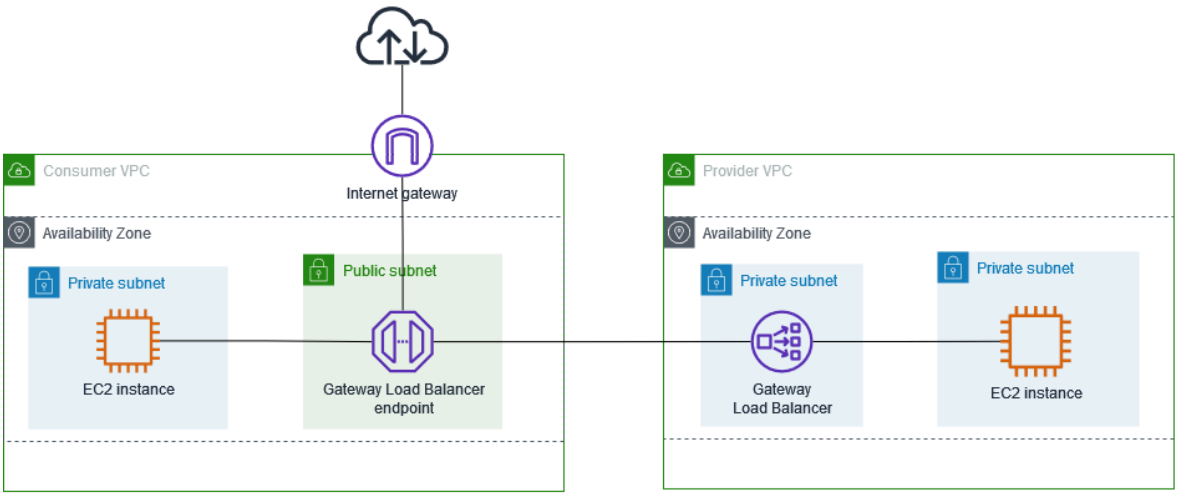
미들박스 라우팅 마법사는 다음의 작업을 자동으로 수행합니다.

- 라우팅 테이블을 생성합니다.
- 필요한 경로를 새 라우팅 테이블에 추가합니다.
- 서브넷과 연결된 현재 라우팅 테이블의 연결을 해제합니다.
- 미들박스 라우팅 마법사가 생성하는 라우팅 테이블을 서브넷과 연결합니다.
- 미들박스 라우팅 마법사에 의해 생성되었음을 나타내는 태그와 생성 날짜를 나타내는 태그를 생성합니다.

미들박스 라우팅 마법사는 기존 라우팅 테이블을 수정하지 않습니다. 새 라우팅 테이블을 생성한 다음 게이트웨이 및 서브넷 리소스와 연결합니다. 리소스가 이미 기존 라우팅 테이블과 명시적으로 연결되

어 있는 경우 기존 라우팅 테이블의 연결이 우선적으로 해제된 다음 새 라우팅 테이블이 리소스와 연결됩니다. 기존 라우팅 테이블은 삭제되지 않습니다.

미들박스 라우팅 마법사를 사용하지 않는 경우 서브넷과 인터넷 게이트웨이에 라우팅 테이블을 수동으로 구성한 다음 할당해야 합니다.



인터넷 게이트웨이 라우팅 테이블

인터넷 게이트웨이의 라우팅 테이블에는 다음의 경로가 포함됩니다.

대상 주소	대상	용도
### VPC CIDR	로컬	로컬 경로
###### ### CIDR	*endpoint-id*	애플리케이션 서브넷을 대상으로 하는 트래픽을 Gateway Load Balancer 엔드포인트로 라우팅

게이트웨이와의 엣지 연결이 있습니다.

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

애플리케이션 서브넷 라우팅 테이블

애플리케이션 서브넷의 라우팅 테이블에는 다음과 같은 경로가 있습니다.

대상 주소	대상	용도
### VPC CIDR	로컬	로컬 경로
0.0.0.0/0	*endpoint-id*	애플리케이션 서버에서 트래픽이 인터넷으로 라우팅되기 전에 Gateway Load Balancer 엔드포인트로 라우팅

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

공급자 서브넷 라우팅 테이블

공급자 서브넷의 라우팅 테이블에는 다음과 같은 경로가 있습니다.

대상 주소	대상	용도
### VPC CIDR	로컬	로컬 경로 인터넷에서 시작된 트래픽이 애플리케이션 서버로 라우팅되는지 확인
0.0.0.0/0	*igw-id*	모든 트래픽을 인터넷 게이트웨이로 라우팅

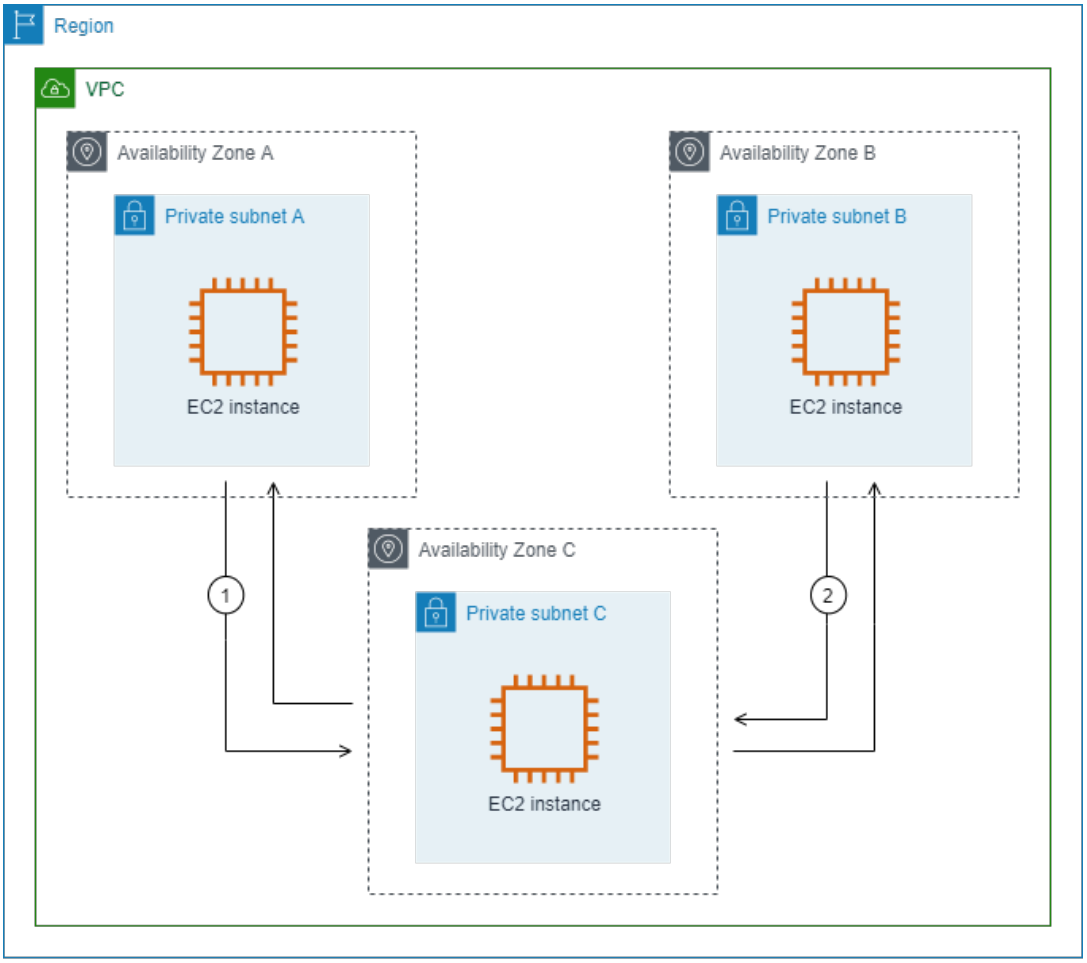
미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

서브넷 간 트래픽 검사

VPC에 여러 개의 서브넷이 있고 방화벽 어플라이언스를 통해 해당 서브넷 간의 트래픽을 검사하려는 시나리오를 생각해봅시다. VPC에 별도의 서브넷에 있는 EC2 인스턴스에 방화벽 어플라이언스를 구성하고 설치합니다.

다음 다이어그램은 서브넷 C의 EC2 인스턴스에 설치된 방화벽 어플라이언스를 보여줍니다. 해당 어플라이언스는 서브넷 A에서 서브넷 B로(1 참조), 서브넷 B에서 서브넷 A(2 참조)로 이동하는 트래픽을 모두 검사합니다.



VPC 및 미들박스 서브넷에 대한 기본 라우팅 테이블을 사용합니다. 서브넷 A와 B에는 각각 사용자 지정 라우팅 테이블이 있습니다.

미들박스 라우팅 마법사는 다음의 작업을 자동으로 수행합니다.

- 라우팅 테이블을 생성합니다.
- 필요한 경로를 새 라우팅 테이블에 추가합니다.
- 서브넷과 연결된 현재 라우팅 테이블의 연결을 해제합니다.
- 미들박스 라우팅 마법사가 생성하는 라우팅 테이블을 서브넷과 연결합니다.
- 미들박스 라우팅 마법사에 의해 생성되었음을 나타내는 태그와 생성 날짜를 나타내는 태그를 생성합니다.

미들박스 라우팅 마법사는 기존 라우팅 테이블을 수정하지 않습니다. 새 라우팅 테이블을 생성한 다음 게이트웨이 및 서브넷 리소스와 연결합니다. 리소스가 이미 기존 라우팅 테이블과 명시적으로 연결되어 있는 경우 기존 라우팅 테이블의 연결이 우선적으로 해제된 다음 새 라우팅 테이블이 리소스와 연결됩니다. 기존 라우팅 테이블은 삭제되지 않습니다.

미들박스 라우팅 마법사를 사용하지 않는 경우 서브넷과 인터넷 게이트웨이에 라우팅 테이블을 수동으로 구성한 다음 할당해야 합니다.

서브넷 A에 대한 사용자 지정 라우팅 테이블

서브넷 A의 라우팅 테이블에는 다음과 같은 경로가 있습니다.

대상 주소	대상	용도
VPC CIDR	로컬	로컬 경로
### B CIDR	*appliance-eni*	서브넷 B를 대상으로 하는 트래픽을 미들박스로 라우팅

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

서브넷 B에 대한 사용자 지정 라우팅 테이블

서브넷 B의 라우팅 테이블에는 다음과 같은 경로가 있습니다.

대상 주소	대상	용도
VPC CIDR	로컬	로컬 경로
### A CIDR	*appliance-eni*	서브넷 A를 대상으로 하는 트래픽을 미들박스로 라우팅

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.

- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

기본 라우팅 테이블

서브넷 C는 기본 라우팅 테이블을 사용합니다. 기본 라우팅 테이블에는 다음과 같은 경로가 있습니다.

대상 주소	대상	용도
VPC CIDR	로컬	로컬 경로

미들박스 라우팅 마법사를 사용하면 다음의 태그가 라우팅 테이블과 연결됩니다.

- 키는 “Origin”이고 값은 “미들박스 마법사”입니다.
- 키는 “date_created”이고 값은 생성 시간입니다(예: “2021-02-18T22:25:49.137Z”)

서브넷 삭제

서브넷이 더 이상 필요하지 않으면 삭제할 수 있습니다. 네트워크 인터페이스가 포함된 서브넷은 삭제할 수 없습니다. 예를 들어 서브넷의 모든 인스턴스를 종료해야 서브넷을 삭제할 수 있습니다.

서브넷을 삭제하면 해당 서브넷과 연결된 CIDR 블록이 VPC의 사용 가능한 IP 주소 풀로 반환됩니다. 즉, 서브넷의 CIDR 범위 내의 IP 주소를 동일한 VPC 내의 다른 서브넷이나 리소스에 재할당할 수 있습니다.

서브넷을 삭제해도 그 안에 있는 리소스가 자동으로 삭제되지는 않는다는 점에 유의해야 합니다. 서브넷 삭제를 진행하려면 먼저 EC2 인스턴스를 종료하고, 네트워크 인터페이스를 삭제하고, 서브넷과 연결된 다른 리소스를 모두 제거해야 합니다.

콘솔을 사용하여 서브넷을 삭제하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 서브넷의 모든 인스턴스를 종료합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 종료를 참조하세요.
3. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
4. 탐색 창에서 Subnets를 선택합니다.
5. 서브넷을 선택한 다음 작업(Actions), 서브넷 삭제(Delete subnet)를 선택합니다.

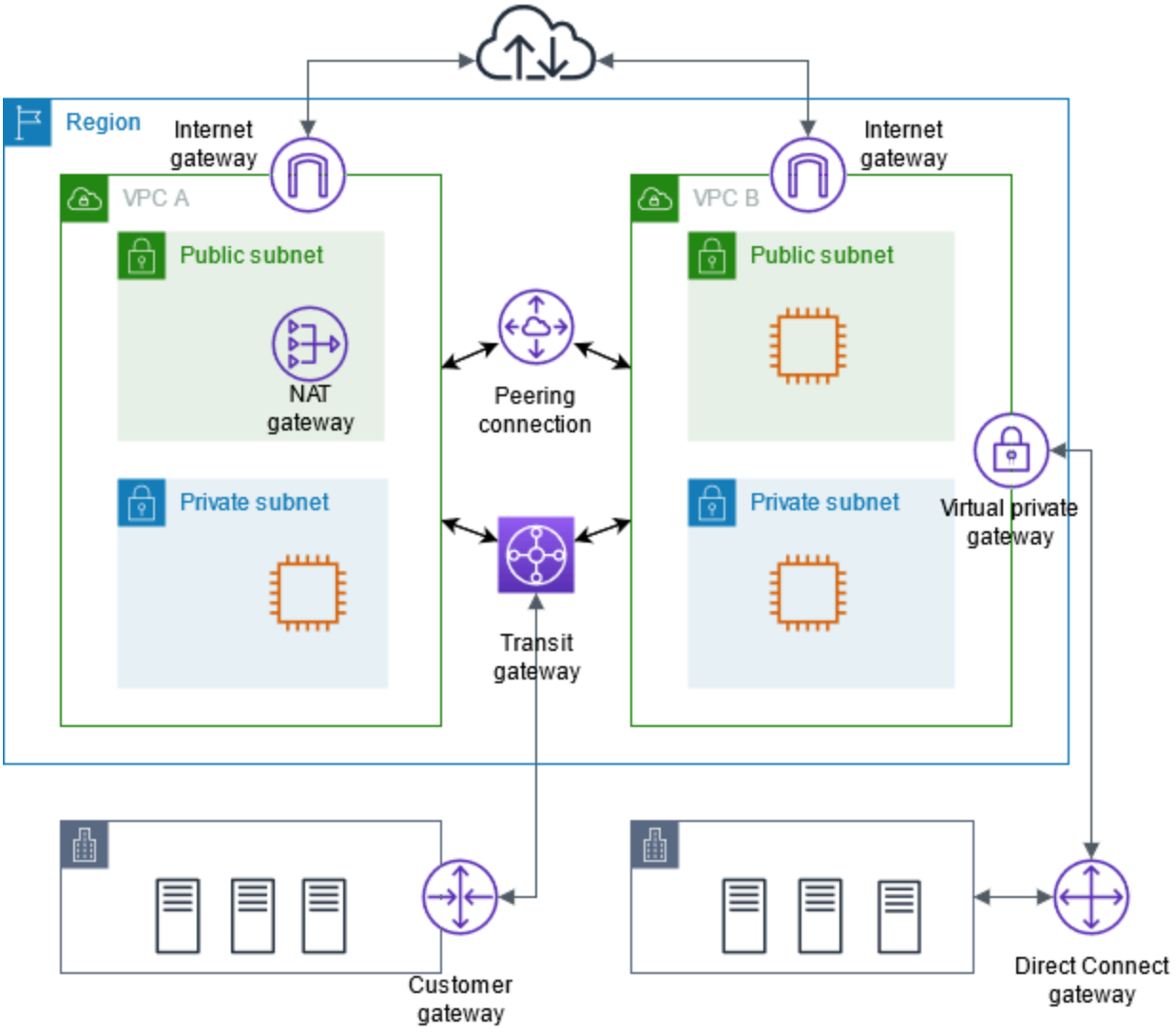
6. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제(Delete)를 선택합니다.

AWS CLI를 사용하여 서브넷을 삭제하려면

delete-subnet 명령을 사용합니다.

다른 네트워크에 VPC 연결

Virtual Private Cloud(VPC)를 다른 VPC, 인터넷, 온프레미스 네트워크 등의 다른 네트워크에 연결할 수 있습니다.



Virtual Private Cloud(VPC)를 다른 VPC, 인터넷, 온프레미스 네트워크 등의 다른 네트워크에 연결할 수 있습니다.

다이어그램에서는 이러한 연결 옵션 중 일부를 보여줍니다. VPC A는 인터넷 게이트웨이를 통해 인터넷에 연결되며, 프라이빗 서브넷의 EC2 인스턴스는 퍼블릭 서브넷의 NAT 게이트웨이를 사용하여 인터넷에 연결할 수 있습니다. VPC B도 인터넷에 연결되지만 직접 인터넷 게이트웨이를 통해 연결되므로 퍼블릭 서브넷의 EC2 인스턴스가 인터넷에 액세스할 수 있습니다.

또한 VPC A와 VPC B는 VPC 피어링 연결과 전송 게이트웨이를 통해 서로 연결됩니다. 전송 게이트웨이에는 데이터 센터에 대한 VPN 연결이 있고, VPC B에는 동일한 데이터 센터에 대한 Direct Connect 연결이 있습니다. 이러한 상호 연결성을 통해 조직은 클라우드 리소스를 온프레미스 인프라와 통합하여 하이브리드 클라우드 환경을 구축할 수 있습니다.

VPC를 다른 네트워크에 연결하는 것은 AWS 내에서 클라우드 인프라를 구축하는 데 있어 중요한 측면입니다. 이를 통해 조직은 네트워킹 구성을 유연하게 제어할 수 있으므로 비즈니스 요건과 보안 요구 사항에 맞는 VPC 아키텍처를 설계할 수 있습니다. 이러한 연결 옵션은 구성 요소가 클라우드 내에 있든 온프레미스에 있든 상관없이, 분산 IT 환경의 다양한 구성 요소 간에 데이터가 효율적으로 흐르도록 지원합니다.

AWS는 인터넷 게이트웨이, NAT 게이트웨이, VPC 피어링, 전송 게이트웨이 및 를 포함하여 이러한 VPC 연결을 활성화하는 다양한 도구와 기능을 제공합니다.Direct Connect 조직은 이러한 기능을 활용하여 기존 IT 인프라와 원활하게 연동되는 안전하고 통합된 클라우드 환경을 생성할 수 있습니다.

Virtual Private Cloud(VPC)를 다른 네트워크에 연결할 수 있습니다. 다른 VPC, 인터넷 또는 온프레미스 네트워크를 예로 들 수 있습니다.

자세한 내용은 Amazon Virtual Private Cloud 연결 옵션을 참조하세요.

내용

- 인터넷 게이트웨이를 사용하여 VPC에 대한 인터넷 액세스 활성화
- 송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽 활성화
- NAT 디바이스를 사용하여 인터넷 또는 다른 네트워크에 연결
- 탄력적 IP 주소를 VPC의 리소스와 연결
- 전송 게이트웨이를 사용하여 다른 VPC 및 네트워크에 VPC 연결
- AWS Virtual Private Network를 사용하여 VPC를 원격 네트워크에 연결
- VPC 피어링을 사용하여 VPC 연결

인터넷 게이트웨이를 사용하여 VPC에 대한 인터넷 액세스 활성화

인터넷 게이트웨이는 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로, VPC와 인터넷 간에 통신할 수 있게 해줍니다. IPv4 트래픽 및 IPv6 트래픽을 지원합니다. 네트워크 트래픽에 가용성 위험이나 대역폭 제약이 발생하지 않습니다.

리소스에 퍼블릭 IPv4 주소 또는 IPv6 주소가 있는 경우 인터넷 게이트웨이를 사용하면 퍼블릭 서브넷(예: EC2 인스턴스)의 리소스가 인터넷에 연결할 수 있습니다. 마찬가지로 인터넷의 리소스는 퍼블릭

IPv4 주소 또는 IPv6 주소를 사용하여 서브넷의 리소스에 대한 연결을 시작할 수 있습니다. 예를 들어 인터넷 게이트웨이를 사용하면 로컬 컴퓨터로 AWS의 EC2 인스턴스에 연결할 수 있습니다.

인터넷 게이트웨이는 VPC 라우팅 테이블에서 인터넷 라우팅 가능 트래픽에 대한 대상을 제공합니다. IPv4 통신의 경우 인터넷 게이트웨이는 Network Address Translation(NAT)도 수행합니다. 자세한 내용은 IP 주소 및 NAT를 참조하세요.

가격 책정

인터넷 게이트웨이에는 요금이 부과되지 않지만 인터넷 게이트웨이를 사용하는 EC2 인스턴스에는 데이터 전송 요금이 부과됩니다. 자세한 내용은 Amazon EC2 온디맨드 요금을 참조하세요.

내용

- 인터넷 게이트웨이 기본 사항
- 서브넷에 인터넷 액세스 추가
- 인터넷 게이트웨이 삭제

인터넷 게이트웨이 기본 사항

인터넷 게이트웨이를 사용하려면 이를 VPC에 연결하고 라우팅을 구성해야 합니다.

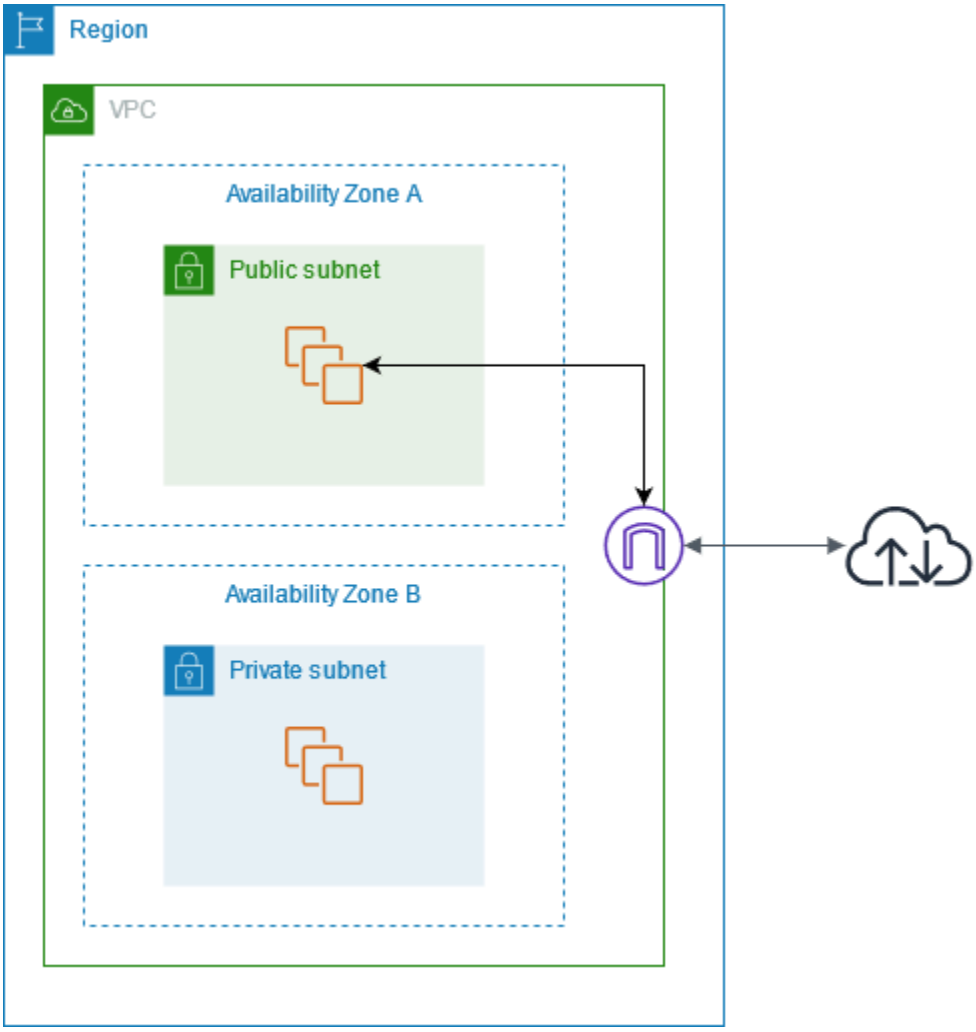
라우팅 구성

서브넷이 인터넷 게이트웨이로 향하는 라우팅이 있는 라우팅 테이블과 연결되는 경우, 이를 퍼블릭 서브넷이라고 합니다. 서브넷이 인터넷 게이트웨이로 향하는 라우팅이 없는 라우팅 테이블과 연결되는 경우 이를 프라이빗 서브넷이라고 합니다.

퍼블릭 서브넷의 라우팅 테이블에서 라우팅 테이블에 명시적으로 알려지지 않은 모든 대상에 대한 인터넷 게이트웨이의 라우팅을 지정할 수 있습니다(IPv4의 경우 0.0.0.0/0 또는 IPv6의 경우 ::/0). 또는 라우팅 범위를 더 좁은 IP 주소 범위(예: AWS 외부에 있는 회사 퍼블릭 엔드포인트의 퍼블릭 IPv4 주소, 또는 VPC 외부에 있는 다른 Amazon EC2 인스턴스의 탄력적 IP 주소)로 제한할 수 있습니다.

인터넷 게이트웨이 다이어그램

다음 다이어그램에서 가용 영역 A의 서브넷은 퍼블릭 서브넷입니다. 라우팅 테이블에 모든 인터넷 바운드 IPv4 트래픽을 인터넷 게이트웨이로 전송하는 경로가 있기 때문입니다. 퍼블릭 서브넷의 인스턴스에는 퍼블릭 IP 주소 또는 탄력적 IP 주소가 있어서 인터넷 게이트웨이를 통해 인터넷과의 통신을 지원할 수 있어야 합니다. 비교해 보면 가용 영역 B의 서브넷은 라우팅 테이블에 인터넷 게이트웨이에 대한 경로가 없기 때문에 프라이빗 서브넷에 해당합니다. 인터넷 게이트웨이로 향하는 경로가 없기 때문에 퍼블릭 IP 주소가 있더라도 프라이빗 서브넷의 인스턴스가 인터넷과 통신할 수 없습니다.



IP 주소 및 NAT

IPv4 인터넷 통신이 가능하게 하려면, 인스턴스에 퍼블릭 IPv4 주소가 있어야 합니다. 인스턴스에 퍼블릭 IPv4 주소를 자동 할당하도록 인스턴스에 탄력적 IP 주소를 할당하도록 VPC를 구성할 수 있습니다. 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.

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.

구성 요소	기본 VPC	기본이 아닌 VPC
인터넷 게이트웨이	예	아니요
IPv4 트래픽을 위한 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블(0.0.0.0/0)	예	아니요
IPv6 트래픽을 위한 인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블(::/0)	아니요	아니요
서브넷에서 시작된 인스턴스에 자동 할당된 퍼블릭 IPv4 주소	예(기본 서브넷)	아니요(기본이 아닌 서브넷)
서브넷에서 시작된 인스턴스에 자동 할당된 IPv6 주소	아니요(기본 서브넷)	아니요(기본이 아닌 서브넷)

서브넷에 인터넷 액세스 추가

다음은 인터넷 게이트웨이를 사용하여 VPC의 서브넷에서 인터넷 액세스를 지원하는 방법을 설명합니다. 인터넷 게이트웨이를 생성하여 VPC에 연결하고 서브넷에 대한 라우팅을 구성해야 합니다.

서브넷에 대한 인터넷 액세스를 구성한 후에는 서브넷의 리소스가 인터넷에 액세스할 수 있는지 확인해야 합니다. 예를 들어 EC2 인스턴스에는 퍼블릭 IPv4 또는 IPv6 주소가 있어야 하며 인스턴스의 보안 그룹은 인터넷과 주고받는 특정 트래픽을 허용해야 합니다.

인스턴스에 퍼블릭 IP 주소를 할당하지 않고 인터넷에 액세스할 수 있도록 하려면 대신 NAT 디바이스를 사용하면 됩니다. 자세한 내용은 NAT 디바이스 섹션을 참조하세요.

인터넷 액세스를 제거하려면 VPC에서 인터넷 게이트웨이를 분리한 후 삭제할 수 있습니다. 자세한 내용은 the section called “인터넷 게이트웨이 삭제” 섹션을 참조하세요.

작업

- 1단계: 인터넷 게이트웨이 생성
- 2단계: 인터넷 게이트웨이를 VPC에 연결

- 3단계: 서브넷 라우팅 테이블에 라우팅 추가

1단계: 인터넷 게이트웨이 생성

인터넷 게이트웨이를 생성하려면 다음 절차를 따르세요.

콘솔을 사용하여 인터넷 게이트웨이 생성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 인터넷 게이트웨이(Internet gateways)를 선택합니다.
3. 인터넷 게이트웨이 생성을 선택합니다.
4. (선택 사항) 인터넷 게이트웨이에 이름을 입력합니다.
5. (선택 사항) 태그를 추가하려면 Add new tag(새 태그 추가)를 선택하고 태그 키와 태그 값을 입력합니다.
6. 인터넷 게이트웨이 생성을 선택합니다.
7. (선택 사항) 지금 인터넷 게이트웨이를 VPC에 연결하려면 화면 상단의 배너에서 VPC에 연결을 선택하고 사용 가능한 VPC를 선택한 다음 인터넷 게이트웨이 연결을 선택합니다. 그렇지 않으면 나중에 인터넷 게이트웨이를 VPC에 연결할 수 있습니다.

명령줄을 사용하여 인터넷 게이트웨이 생성

- create-internet-gateway(AWS CLI)
- New-EC2InternetGateway(AWS Tools for Windows PowerShell)

2단계: 인터넷 게이트웨이를 VPC에 연결

인터넷 게이트웨이를 사용하려면 이를 VPC에 연결해야 합니다.

콘솔을 사용하여 인터넷 게이트웨이를 VPC에 연결

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 인터넷 게이트웨이(Internet gateways)를 선택합니다.
3. 해당 인터넷 게이트웨이의 확인란을 선택합니다.
4. 연결하려면 작업, VPC에 연결을 선택하고 사용 가능한 VPC를 선택한 다음 인터넷 게이트웨이 연결을 선택합니다.

5. 분리하려면 작업, VPC에서 분리, 인터넷 게이트웨이 분리를 차례로 선택합니다. 확인 메시지가 나타나면 인터넷 게이트웨이 분리를 선택합니다.

명령줄을 사용하여 인터넷 게이트웨이를 VPC에 연결

- attach-internet-gateway(AWS CLI)
- Add-EC2InternetGateway(AWS Tools for Windows PowerShell)

3단계: 서브넷 라우팅 테이블에 라우팅 추가

서브넷의 라우팅 테이블에는 인터넷 게이트웨이로 인터넷 트래픽을 보내는 경로가 있어야 합니다.

콘솔을 사용하여 서브넷 라우팅 테이블을 구성

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 서브넷에 대한 라우팅 테이블을 선택합니다. 기본적으로 서브넷은 VPC의 기본 라우팅 테이블을 사용합니다. 또는 라우팅 테이블을 생성한 다음 서브넷을 새 라우팅 테이블과 연결할 수 있습니다.
4. 경로(Routes) 탭에서 경로 편집(Edit routes) 및 경로 추가(Add route)를 차례로 선택합니다.
5. 대상 주소에 0.0.0.0/0을 입력하고 대상에 대한 인터넷 게이트웨이를 선택합니다.
6. 변경 사항 저장을 선택합니다.

명령줄을 사용하여 서브넷 라우팅 테이블을 구성

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

인터넷 게이트웨이 삭제

더 이상 인터넷 액세스가 필요 없으면 VPC에서 인터넷 게이트웨이를 분리한 후 삭제할 수 있습니다. 아직 VPC에 연결되어 있는 인터넷 게이트웨이는 삭제할 수 없습니다. VPC에 퍼블릭 IP 주소 또는 탄력적 IP 주소가 연결된 리소스가 있는 경우에는 인터넷 게이트웨이를 분리할 수 없습니다.

콘솔을 사용하여 VPC에서 인터넷 게이트웨이를 분리

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

2. 탐색 창에서 인터넷 게이트웨이(Internet gateways)를 선택합니다.
3. 해당 인터넷 게이트웨이의 확인란을 선택합니다.
4. 연결하려면 작업, VPC에 연결을 선택하고 사용 가능한 VPC를 선택한 다음 인터넷 게이트웨이 연결을 선택합니다.
5. 분리하려면 작업, VPC에서 분리, 인터넷 게이트웨이 분리를 차례로 선택합니다. 확인 메시지가 나타나면 인터넷 게이트웨이 분리를 선택합니다.

명령줄을 사용하여 연결을 포함한 인터넷 게이트웨이를 설명

- describe-internet-gateways(AWS CLI)
- Get-EC2InternetGateway(AWS Tools for Windows PowerShell)

명령줄을 사용하여 VPC에서 인터넷 게이트웨이를 분리

- detach-internet-gateway(AWS CLI)
- Dismount-EC2InternetGateway(AWS Tools for Windows PowerShell)

콘솔을 사용하여 인터넷 게이트웨이 삭제

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 인터넷 게이트웨이(Internet gateways)를 선택합니다.
3. 해당 인터넷 게이트웨이의 확인란을 선택합니다.
4. 작업, 인터넷 게이트웨이 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **delete**를 입력한 다음 인터넷 게이트웨이 삭제를 선택합니다.

명령줄을 사용하여 인터넷 게이트웨이 삭제

- delete-internet-gateway(AWS CLI)
- Remove-EC2InternetGateway(AWS Tools for Windows PowerShell)

송신 전용 인터넷 게이트웨이를 사용하여 아웃바운드 IPv6 트래픽 활성화

외부 전용 인터넷 게이트웨이는 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로서, VPC의 인스턴스에서 인터넷으로 IPv6을 통한 아웃바운드 통신을 가능케 하되 인터넷에서 해당 인스턴스와의 IPv6 연결을 시작하지 못하게 할 수 있습니다.

외부 전용 인터넷 게이트웨이는 IPv6 트래픽에만 사용됩니다. IPv4를 통한 아웃바운드 전용 인터넷 통신을 사용하려면 NAT 게이트웨이를 사용하십시오. 자세한 내용은 NAT 게이트웨이 섹션을 참조하세요.

가격 책정

외부 전용 인터넷 게이트웨이에는 요금이 부과되지 않지만 인터넷 게이트웨이를 사용하는 EC2 인스턴스에는 데이터 전송 요금이 부과됩니다. 자세한 내용은 Amazon EC2 온디맨드 요금을 참조하세요.

내용

- 외부 전용 인터넷 게이트웨이 기본 사항
- 서브넷에 외부 전용 인터넷 액세스 추가

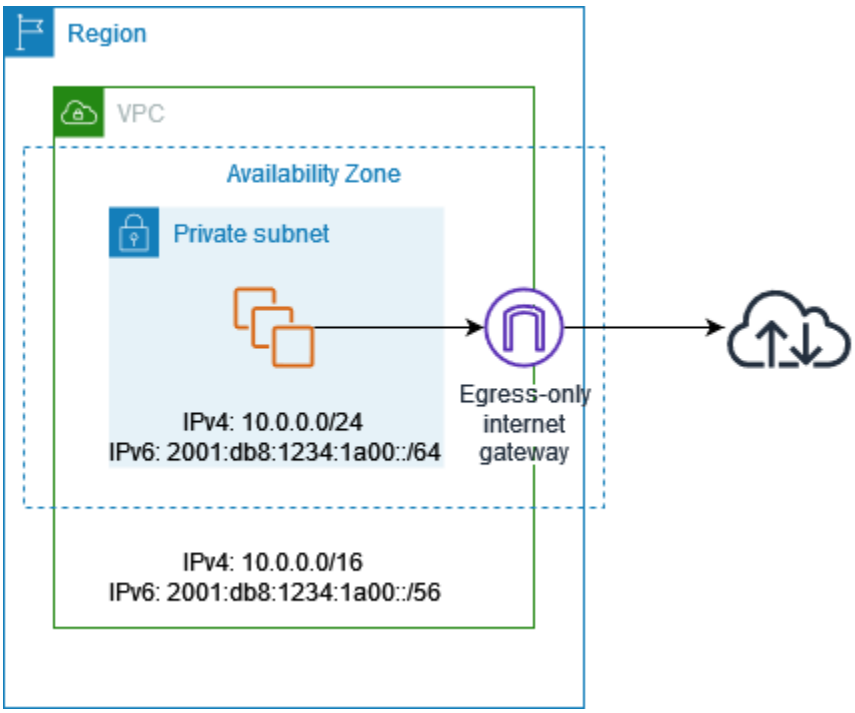
외부 전용 인터넷 게이트웨이 기본 사항

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 사용하면 됩니다. 이렇게 하려면 VPC에 외부 전용 인터넷 게이트웨이를 만들어 라우팅 테이블에 모든 IPv6 트래픽(`::/0`)을 가리키는 라우팅을 추가하거나 IPv6 주소의 특정 범위를 외부 전용 인터넷 게이트웨이에 추가합니다. 라우팅 테이블에 연결된 서브넷의 IPv6 트래픽은 외부 전용 인터넷 게이트웨이로 라우팅됩니다.

외부 전용 인터넷 게이트웨이는 상태 저장 방식으로서, 서브넷의 인스턴스에서 인터넷 또는 기타 AWS 서비스로 트래픽을 전달한 다음, 다시 인스턴스로 응답을 보냅니다.

보안 그룹을 외부 전용 인터넷 게이트웨이와 연결하여 외부 전용 인터넷 게이트웨이에 도달하거나 출발하는 트래픽을 제어할 수 없습니다. 네트워크 ACL을 사용하여 외부 전용 인터넷 게이트웨이가 트래픽을 라우팅하는 서브넷에서 주고받는 트래픽을 제어할 수 있습니다.

다음 다이어그램에서 VPC에는 IPv4 및 IPv6 CIDR 블록이 모두 있고 서브넷에는 IPv4 및 IPv6 CIDR 블록이 모두 있습니다. VPC에는 송신 전용 인터넷 게이트웨이가 있습니다.



다음은 서브넷과 연결된 라우팅 테이블의 예입니다. 모든 인터넷 바인딩 IPv6 트래픽(::/0)을 외부 전용 인터넷 게이트웨이로 전송하는 경로가 있습니다.

대상 주소	대상
10.0.0.0/16	로컬
2001:db8:1234:1a00:/64	로컬
::/0	*eigw-id*

서브넷에 외부 전용 인터넷 액세스 추가

다음 작업에서는 프라이빗 서브넷에 대해 외부 전용(아웃바운드) 인터넷 게이트웨이를 생성하고 서브넷에 대한 라우팅을 구성하는 방법에 대해 설명합니다.

작업

- 1. 외부 전용 인터넷 게이트웨이 생성
- 2. 사용자 지정 라우팅 테이블 생성
- 3. 외부 전용 인터넷 게이트웨이 삭제
- 명령줄 개요

1. 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. (선택) 태그를 추가하거나 제거할 수 있습니다.

   [태그 추가] 새 태그 추가를 선택하고 다음을 수행합니다.

   - 키에서 키 이름을 입력합니다.
   - 값에 키 값을 입력합니다.

   [태그 제거] 태그의 키 및 값 오른쪽에 있는 제거를 선택하세요.
5. 외부 전용 인터넷 게이트웨이를 생성할 VPC를 선택합니다.
6. 생성(Create)을 선택합니다.

2. 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 라우팅을 추가한 다음, 이를 서브넷과 연결해야 합니다.

사용자 지정 라우팅 테이블을 만들고 외부 전용 인터넷 게이트웨이에 라우팅을 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [라우팅 테이블(Route Tables)]을 선택한 다음, [라우팅 테이블 생성(Create Route Table)]을 선택합니다.
3. [라우팅 테이블 생성(Create route table)] 대화 상자에서 선택적으로 라우팅 테이블의 이름을 지정한 다음 VPC를 선택하고 [라우팅 테이블 생성(Create route table)]을 선택합니다.
4. 방금 생성한 사용자 지정 라우팅 테이블을 선택합니다. 세부 정보 창에는 경로, 연결 및 경로 전파 작업을 위한 탭이 표시됩니다.

5. [경로(Routes)] 탭에서 [경로 편집(Edit routes)]을 선택하고, [대상 주소(Destination)] 상자에서 `::/0`을 지정하고, [대상(Target)] 목록에서 송신 전용 인터넷 게이트웨이 ID를 선택한 다음, [변경 사항 저장(Save changes)]을 선택합니다.
6. [서브넷 연결(Subnet associations)] 탭에서 [서브넷 연결 편집(Edit subnet associations)]을 선택하고 서브넷에 대한 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 라우팅을 추가합니다.

라우팅 테이블에 대한 자세한 내용은 라우팅 테이블 구성 단원을 참조하십시오.

3. 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 라우팅은 그 라우팅을 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 외부 전용 인터넷 게이트웨이를 선택하고 해당하는 외부 전용 인터넷 게이트웨이를 선택합니다.
3. 삭제를 선택합니다.
4. 확인 대화 상자에서 Delete Egress Only Internet Gateway를 선택합니다.

명령줄 개요

이 페이지에서 설명한 작업은 명령줄을 사용하여 수행할 수 있습니다.

외부 전용 인터넷 게이트웨이 생성

- create-egress-only-internet-gateway (AWS CLI)
- New-EC2EgressOnlyInternetGateway (AWS Tools for Windows PowerShell)

외부 전용 인터넷 게이트웨이 설명

- describe-egress-only-internet-gateways (AWS CLI)

- Get-EC2EgressOnlyInternetGatewayList (AWS Tools for Windows PowerShell)

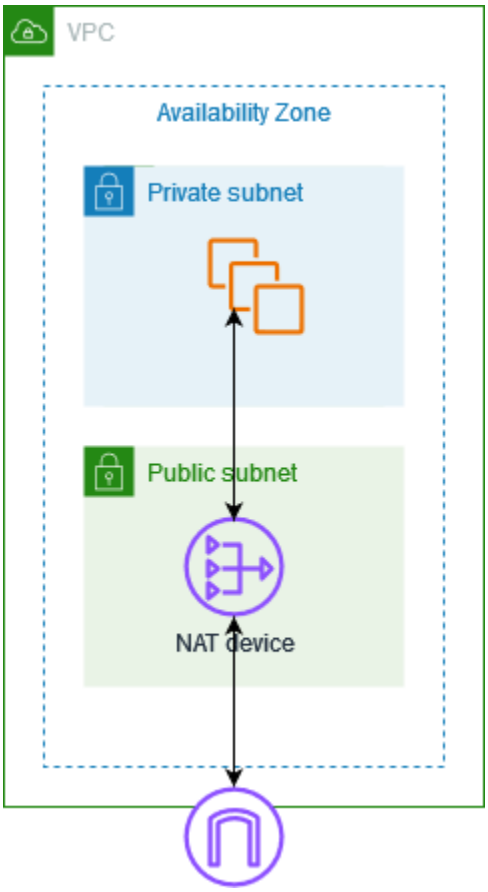
외부 전용 인터넷 게이트웨이 삭제

- delete-egress-only-internet-gateway (AWS CLI)
- Remove-EC2EgressOnlyInternetGateway (AWS Tools for Windows PowerShell)

NAT 디바이스를 사용하여 인터넷 또는 다른 네트워크에 연결

NAT 디바이스를 사용하여 프라이빗 서브넷의 리소스가 인터넷, 다른 VPC 또는 온프레미스 네트워크에 연결되도록 허용할 수 있습니다. 이러한 인스턴스는 VPC 외부의 서비스와 통신할 수 있지만 원치 않는 연결 요청은 받을 수는 없습니다.

예를 들면 다음 다이어그램에서는 인터넷 게이트웨이를 통해 프라이빗 서브넷의 EC2 인스턴스를 인터넷에 연결할 수 있는 퍼블릭 서브넷의 NAT 디바이스를 보여줍니다. NAT 디바이스는 인스턴스의 소스 IPv4 주소를 NAT 디바이스의 주소로 바꿉니다. 인스턴스에 응답 트래픽을 전송할 때 NAT 디바이스는 주소를 원래 소스 IPv4 주소로 다시 변환합니다.



> **Important**
>
> - 이 문서에서는 NAT라는 용어를 일반적인 IT 용례에 따라 사용하지만, 실제로 NAT 디바이스는 주소 변환과 포트 주소 변환(PAT)을 모두 담당합니다.
> - AWS에서 제공하는 NAT 게이트웨이라는 관리형 NAT 디바이스를 사용하거나, EC2 인스턴스에서 NAT 디바이스(여기서는 NAT 인스턴스라고 함)를 생성할 수 있습니다. NAT 게이트웨이는 더 나은 가용성과 대역폭을 제공하고 관리에 소요되는 작업이 줄어들기 때문에 권장합니다.

내용

- NAT 게이트웨이
- NAT 인스턴스
- NAT 게이트웨이 및 NAT 인스턴스 비교

NAT 게이트웨이

NAT 게이트웨이는 네트워크 주소 변환(NAT) 서비스입니다. NAT 게이트웨이를 사용하면 프라이빗 서브넷의 인스턴스가 VPC 외부의 서비스에 연결할 수 있지만 외부 서비스에서 이러한 인스턴스와의 연결을 시작할 수는 없도록 할 수 있습니다.

NAT 게이트웨이를 만들 때 다음 연결 유형 중 하나를 지정합니다.

- 퍼블릭 - (기본값) 프라이빗 서브넷의 인스턴스는 퍼블릭 NAT 게이트웨이를 통해 인터넷에 연결할 수 있지만 인스턴스가 인터넷에서 원치 않는 인바운드 연결을 수신할 수 없습니다. 퍼블릭 서브넷에서 퍼블릭 NAT 게이트웨이를 생성하고 생성 시 탄력적 IP 주소를 NAT 게이트웨이와 연결해야 합니다. 트래픽을 NAT 게이트웨이에서 VPC용 인터넷 게이트웨이로 라우팅합니다. 또는 퍼블릭 NAT 게이트웨이를 사용하여 다른 VPC 또는 온프레미스 네트워크에 연결할 수 있습니다. 이 경우 NAT 게이트웨이에서 Transit Gateway 또는 가상 프라이빗 게이트웨이를 통해 트래픽을 라우팅합니다.
- 프라이빗 – 프라이빗 서브넷의 인스턴스는 프라이빗 NAT 게이트웨이를 통해 다른 VPC 또는 온프레미스 네트워크에 연결할 수 있지만 인스턴스가 다른 VPC나 온프레미스 네트워크에서 원치 않는 인바운드 연결을 수신할 수 없습니다. 트래픽을 NAT 게이트웨이에서 Transit Gateway 또는 가상 프라이빗 게이트웨이를 통해 트래픽을 라우팅할 수 있습니다. 탄력적 IP 주소를 프라이빗 NAT 게이트웨이에 연결할 수 없습니다. 프라이빗 NAT 게이트웨이를 사용하여 VPC에 인터넷 게이트웨이를 연결할 수 있지만 프라이빗 NAT 게이트웨이에서 인터넷 게이트웨이로 트래픽을 라우팅하는 경우 인터넷 게이트웨이가 트래픽을 삭제합니다.

NAT 게이트웨이는 IPv4 또는 IPv6 트래픽에 사용됩니다(DNS64 및 NAT64 사용). IPv6를 통한 아웃바운드 전용 인터넷 통신을 활성화하는 다른 옵션은 외부 전용 인터넷 게이트웨이를 대신 사용하는 것입니다.

프라이빗 및 퍼블릭 NAT 게이트웨이는 모두 인스턴스의 소스 프라이빗 IPv4 주소를 NAT 게이트웨이의 프라이빗 IPv4 주소에 매핑합니다. 퍼블릭 NAT 게이트웨이의 경우, 인터넷 게이트웨이가 퍼블릭 NAT 게이트웨이의 프라이빗 IPv4 주소를 해당 NAT 게이트웨이에 연결된 탄력적 IP 주소에 매핑합니다.

인스턴스에 응답 트래픽을 전송할 때 인스턴스는 퍼블릭 또는 프라이빗 NAT 게이트웨어 여부에 상관없이 NAT 게이트웨이는 주소를 원래 소스 IP 주소로 다시 변환합니다.

고려 사항

- 연결은 항상 NAT 게이트웨이가 포함된 VPC 내에서 시작되어야 합니다.
- 퍼블릭 또는 프라이빗 NAT 게이트웨이를 사용하여 트래픽을 전송 게이트웨이와 가상 프라이빗 게이트웨이로 라우팅할 수 있습니다.
- 프라이빗 NAT 게이트웨이를 사용하여 전송 게이트웨이 또는 가상 프라이빗 게이트웨이에 연결하는 경우 대상에 대한 트래픽은 프라이빗 NAT 게이트웨이의 프라이빗 IP 주소에서 전송됩니다.
- 퍼블릭 NAT 게이트웨이를 사용하여 전송 게이트웨이 또는 가상 프라이빗 게이트웨이에 연결하는 경우 대상에 대한 트래픽은 퍼블릭 NAT 게이트웨이의 프라이빗 IP 주소에서 전송됩니다. 퍼블릭 NAT 게이트웨이는 동일한 VPC의 인터넷 게이트웨이와 함께 사용할 때만 탄력적 IP 주소를 소스 IP 주소로 사용합니다.

내용

- NAT 게이트웨이 기본 사항
- NAT 게이트웨이 작업
- 자동 다중 AZ 확장을 위한 리전 NAT 게이트웨이
- NAT Gateway 사용 사례
- DNS64 및 NAT64
- NAT 게이트웨이의 트래픽 검사
- Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링
- NAT 게이트웨이 문제 해결
- NAT 게이트웨이 요금

NAT 게이트웨이 기본 사항

각 NAT 게이트웨이는 특정 가용 영역에 생성되고 해당 영역에서 중복성을 통해 구현됩니다. 각 가용 영역에서 만들 수 있는 NAT 게이트웨이 개수에는 할당량이 있습니다. 자세한 내용은 게이트웨이 섹션을 참조하세요.

여러 가용 영역의 리소스가 하나의 NAT 게이트웨이를 공유할 때 해당 NAT 게이트웨이가 위치한 가용 영역 장애가 발생하면, 다른 가용 영역의 리소스도 인터넷에 액세스할 수 없게 됩니다. 복원력 향상을 위해 각 가용 영역에 NAT 게이트웨이를 생성하고 리소스가 동일한 가용 영역의 NAT 게이트웨이를 사용하도록 라우팅을 구성합니다.

NAT 게이트웨이에 적용되는 특성 및 규칙은 다음과 같습니다.

- NAT 게이트웨이는 TCP, UDP, ICMP 등의 프로토콜을 지원합니다.
- NAT 게이트웨이는 IPv4 또는 IPv6 트래픽에 대해 지원됩니다. IPv6 트래픽의 경우 NAT 게이트웨이가 NAT64를 수행합니다. 이를 DNS64(Route 53 Resolver에서 사용 가능)와 함께 사용하면 Amazon VPC의 서브넷에 있는 IPv6 워크로드가 IPv4 리소스와 통신할 수 있습니다. 이러한 IPv4 서비스는 동일한 VPC(별도의 서브넷에 있음) 또는 다른 VPC, 온프레미스 환경 또는 인터넷에 존재할 수 있습니다.
- NAT 게이트웨이는 5Gbps의 대역폭을 지원하며 최대 100Gbps까지 자동 확장합니다. 더 많은 대역폭이 필요한 경우 리소스를 여러 서브넷으로 분할하고 각 서브넷에 NAT 게이트웨이를 만들 수 있습니다.
- NAT 게이트웨이는 초당 1,000,000개의 패킷을 처리할 수 있으며 초당 최대 10,000,000개의 패킷까지 자동으로 확장됩니다. 이 제한을 초과하면 NAT 게이트웨이가 패킷을 삭제합니다. 패킷 손실을 방지하려면 리소스를 여러 서브넷으로 분할하고 각 서브넷에 대해 별도의 NAT 게이트웨이를 생성합니다.
- 각 IPv4 주소는 각 고유 대상에 대해 55,000개까지 동시 연결을 지원할 수 있습니다. 고유 대상은 대상 IP 주소, 대상 포트 및 프로토콜(TCP/UDP/ICMP)의 고유한 조합으로 식별됩니다. 최대 8개의 IPv4 주소를 NAT 게이트웨이에 연결하여 이 제한을 늘릴 수 있습니다(기본 IPv4 주소 1개 및 보조 IPv4 주소 7개). 기본적으로 2개의 탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하는 것으로 제한됩니다. 할당량 조정을 요청하여 이 제한을 늘릴 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.
- NAT 게이트웨이를 생성할 때 NAT 게이트웨이에 할당된 기본 프라이빗 IPv4 주소를 선택할 수 있습니다. 그렇지 않으면 서브넷의 IPv4 주소 범위에서 사용자를 대신해 하나를 선택합니다. 기본 프라이빗 IPv4 주소는 변경하거나 제거할 수 없습니다. 필요에 따라 보조 프라이빗 IPv4 주소를 추가할 수 있습니다.

- 보안 그룹을 NAT 게이트웨이와 연결할 수 없습니다. 보안 그룹을 인스턴스에 연결하여 인바운드 및 아웃바운드 트래픽을 제어할 수 있습니다.
- NAT 게이트웨이에 대한 요청자 관리형 네트워크 인터페이스를 생성합니다. Amazon EC2 콘솔을 사용하여 이 네트워크 인터페이스를 볼 수 있습니다. 설명에서 NAT 게이트웨이의 ID를 검색합니다. 네트워크 인터페이스에 태그를 추가할 수 있지만 이 네트워크 인터페이스의 다른 속성은 수정할 수 없습니다.
- 네트워크 ACL을 사용하여 NAT 게이트웨이에 대해 서브넷에서 주고받는 트래픽을 제어할 수 있습니다. NAT 게이트웨이는 포트 1024-65535를 사용합니다. 자세한 내용은 네트워크 ACL 섹션을 참조하세요.
- VPC 피어링 연결을 통해 NAT 게이트웨이로 트래픽을 라우팅할 수 없습니다. 그러나 VPC 피어링을 통해 NAT 게이트웨이에서 피어링된 VPC의 대상으로 가는 트래픽은 '발신자에게 반환' 동작을 지원합니다. 반환 트래픽은 대상 VPC에 반환 경로가 구성되지 않은 경우에도 자동으로 원래 NAT 게이트웨이로 다시 라우팅됩니다. 이 동작은 NAT 게이트웨이에만 해당되며 표준 EC2 인스턴스에는 적용되지 않습니다. 이를 방지하려면 NACL을 사용하여 반환 트래픽을 차단합니다.

  지원되지 않음:

  ```
  Client # Peering # NAT # Internet
  ```

  지원됨:

  ```
  Client # NAT # Peering # Destination
  ```

- 가상 프라이빗 게이트웨이를 사용하여 Site-to-Site VPN 또는 Direct Connect에서 NAT 게이트웨이로 트래픽을 라우팅할 수 없습니다. 가상 프라이빗 게이트웨이 대신 전송 게이트웨이를 사용하면 Site-to-Site VPN 또는 Direct Connect에서 NAT 게이트웨이로 트래픽을 라우팅할 수 있습니다.
- NAT 게이트웨이에서 지원되는 트래픽의 최대 전송 단위(MTU)는 8,500이지만, 다음 사항에 유의해야 합니다.
  - 네트워크 연결의 MTU는 연결을 통해 전달할 수 있는 허용되는 최대 패킷의 크기(바이트)입니다. 연결의 MTU가 클수록 하나의 패킷으로 전달할 수 있는 데이터의 양이 늘어납니다.
  - NAT 게이트웨이에 도착하는 8,500바이트보다 큰 패킷은 삭제됩니다(또는 조각화가 가능한 경우 조각화됨).
  - 퍼블릭 NAT 게이트웨이를 사용하여 인터넷을 통해 리소스와 통신할 때 잠재적 패킷 손실을 방지하려면 EC2 인스턴스의 MTU 설정이 1,500바이트를 초과하지 않아야 합니다. 인스턴스의 MTU 확인 및 설정에 대한 자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스의 네트워크 MTU를 참조하세요.

- NAT 게이트웨이는 FRAG_NEEDED ICMPv4 패킷 및 Packet Too Big(PTB) ICMPv6 패킷을 통한 경로 MTU 검색(PMTUD)을 지원합니다.
- NAT 게이트웨이에서는 모든 패킷에 MSS(최대 세그먼트 크기) 클램핑이 강제로 적용됩니다. 자세한 내용은 RFC879를 참조하세요.

NAT 게이트웨이 작업

Amazon VPC 콘솔을 사용하여 NAT 게이트웨이를 생성하고 관리할 수 있습니다.

작업

- NAT 게이트웨이 사용 제어
- NAT 게이트웨이 만들기
- 보조 IP 주소 연결 편집
- NAT 게이트웨이 태그 지정
- NAT 게이트웨이 삭제
- 명령줄 개요

NAT 게이트웨이 사용 제어

기본적으로 사용자에게는 NAT 게이트웨이를 사용할 권한이 없습니다. 사용자에게 NAT 게이트웨이를 생성, 설명, 삭제할 수 있는 권한을 부여하는 정책이 연결된 IAM 역할을 만들 수 있습니다. 자세한 내용은 Amazon VPC용 자격 증명 및 액세스 관리 섹션을 참조하세요.

NAT 게이트웨이 만들기

다음 절차에 따라 NAT 게이트웨이를 생성하세요.

관련 할당량

- 계정에 할당된 탄력적 IP 주소 수를 모두 사용한 경우 퍼블릭 NAT 게이트웨이를 생성할 수 없습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.
- 프라이빗 NAT 게이트웨이에 최대 8개의 프라이빗 IPv4 주소를 할당할 수 있습니다. 이 제한은 조정할 수 없습니다.
- 기본적으로 2개의 탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하는 것으로 제한됩니다. 할당량 조정을 요청하여 이 제한을 늘릴 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.

NAT 게이트웨이를 만들려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 NAT 게이트웨이를 선택합니다.
3. NAT 게이트웨이 생성을 선택합니다.
4. (선택 사항) NAT 게이트웨이의 이름을 지정합니다. 이렇게 하면 키가 **Name**이고 값이 사용자가 지정한 이름인 태그가 생성됩니다.
5. NAT 게이트웨이를 생성할 서브넷을 선택합니다.
6. 연결 유형에서 기본 선택 퍼블릭을 그대로 사용하여 퍼블릭 NAT 게이트웨이를 생성하거나 프라이빗을 선택하여 프라이빗 NAT 게이트웨이를 생성합니다. 퍼블릭과 프라이빗 NAT 게이트웨이 간의 차이점에 대한 자세한 정보는 NAT 게이트웨이를 참조하세요.
7. 퍼블릭을 선택한 경우 다음을 수행합니다. 그렇지 않으면 8단계로 건너뜁니다.

   1. 탄력적 IP 할당 ID를 선택하여 탄력적 IP 주소를 NAT 게이트웨이에 할당하거나 탄력적 IP 할당을 선택하여 퍼블릭 NAT 게이트웨이에 사용할 탄력적 IP를 자동으로 할당합니다. 기본적으로 2개의 탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하는 것으로 제한됩니다. 할당량 조정을 요청하여 이 제한을 늘릴 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.

      > **Important**
      >
      > 퍼블릭 NAT 게이트웨이에 탄력적 IP 주소를 할당하는 경우, EIP의 네트워크 경계 그룹은 반드시 퍼블릭 NAT 게이트웨이를 시작하는 대상 가용 영역의 네트워크 경계 그룹과 일치해야 합니다. 일치하지 않을 경우 NAT 게이트웨이가 시작되지 않습니다. 서브넷의 세부 정보를 확인하면 해당 서브넷 AZ의 네트워크 경계 그룹을 확인할 수 있습니다. 마찬가지로 EIP 주소의 세부 정보를 확인하면 EIP의 네트워크 경계 그룹을 확인할 수 있습니다. 자세한 내용은 1. 탄력적 IP 주소 할당 섹션을 참조하세요.

   2. (선택 사항) 추가 설정을 선택하고 프라이빗 IP 주소 - 선택 사항에 NAT 게이트웨이의 프라이빗 IPv4 주소를 입력합니다. 주소를 입력하지 않으면 AWS는 NAT 게이트웨이가 있는 서브넷에서 무작위로 NAT 게이트웨이에 프라이빗 IPv4 주소를 자동으로 할당합니다.
   3. 11단계로 건너뜁니다.
8. 프라이빗을 선택한 경우 추가 설정, 프라이빗 IPv4 주소 할당 방법에서 다음 중 하나를 선택합니다.

- 자동 할당: AWS가 NAT 게이트웨이의 기본 프라이빗 IPv4 주소를 선택합니다. 자동 할당된 프라이빗 IPv4 주소 수에서 NAT 게이트웨이의 보조 프라이빗 IPv4 주소 수를 필요한 경우 지정할 수 있습니다. AWS가 NAT 게이트웨이의 서브넷에서 임의로 해당 IP 주소를 선택합니다.
- 사용자 지정: 기본 프라이빗 IPv4 주소에서 NAT 게이트웨이의 기본 프라이빗 IPv4 주소를 선택합니다. 보조 프라이빗 IPv4 주소에서 NAT 게이트웨이에 최대 7개의 보조 프라이빗 IPv4 주소를 필요한 경우 지정할 수 있습니다.

9. 8단계에서 사용자 지정을 선택한 경우 이 단계를 건너뛰세요. 자동 할당을 선택한 경우 자동 할당된 프라이빗 IP 주소 수에서 AWS가 이 프라이빗 NAT 게이트웨이에 할당할 보조 IPv4 주소 수를 선택합니다. IPv4 주소를 최대 7개까지 선택할 수 있습니다.

   > **Note**
   >
   > 보조 IPv4 주소는 선택 사항이며 NAT 게이트웨이를 사용하는 워크로드가 단일 대상(동일한 대상 IP, 대상 포트 및 프로토콜)에 대한 동시 연결 55,000개를 초과하는 경우 지정 또는 할당해야 합니다. 보조 IPv4 주소는 사용 가능한 포트 수를 늘리므로 워크로드가 NAT 게이트웨이를 사용하여 설정할 수 있는 동시 연결 수에 대한 제한이 늘어납니다.

10. 9단계에서 자동 할당을 선택한 경우 이 단계를 건너뛰세요. 사용자 지정을 선택한 경우 다음을 수행합니다.

    1. 기본 프라이빗 IPv4 주소에 프라이빗 IPv4 주소를 입력합니다.
    2. 보조 프라이빗 IPv4 주소에 최대 7개의 보조 프라이빗 IPv4 주소를 입력합니다.

11. (선택 사항) NAT 게이트웨이에 태그를 추가하려면 새로운 태그 추가(Add new tag)를 선택하고 키 이름과 값을 입력합니다. 최대 50개의 태그를 추가할 수 있습니다.
12. NAT 게이트웨이 생성을 선택합니다.
13. NAT 게이트웨이의 초기 상태는 `Pending`입니다. 상태가 `Available`(으)로 변경되면 NAT 게이트웨이가 사용 준비 상태가 됩니다. 필요에 따라 라우팅 테이블을 업데이트해야 합니다. 예시는 the section called “사용 사례” 섹션을 참조하세요.

NAT 게이트웨이의 상태가 `Failed` 상태로 바뀌면 생성 중에 오류가 발생한 것입니다. 자세한 내용은 NAT 게이트웨이 생성 실패 섹션을 참조하세요.

보조 IP 주소 연결 편집

각 IPv4 주소는 각 고유 대상에 대해 55,000개까지 동시 연결을 지원할 수 있습니다. 고유 대상은 대상 IP 주소, 대상 포트 및 프로토콜(TCP/UDP/ICMP)의 고유한 조합으로 식별됩니다. 최대 8개의 IPv4 주

소를 NAT 게이트웨이에 연결하여 이 제한을 늘릴 수 있습니다(기본 IPv4 주소 1개 및 보조 IPv4 주소 7개). 기본적으로 2개의 탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하는 것으로 제한됩니다. 할당량 조정을 요청하여 이 제한을 늘릴 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.

NAT 게이트웨이 CloudWatch 지표, ErrorPortLocation, PacketsDropCount를 사용하여 NAT 게이트웨이에서 포트 할당 오류가 발생하거나 패킷이 손실되는지 확인할 수 있습니다. 이 문제를 해결하려면 NAT 게이트웨이에 보조 IPv4 주소를 추가하세요.

고려 사항

- 프라이빗 NAT 게이트웨이를 생성할 때 또는 이 섹션의 절차를 사용하여 NAT 게이트웨이를 생성한 후에 보조 프라이빗 IPv4 주소를 추가할 수 있습니다. 이 섹션의 절차를 사용하여 NAT 게이트웨이를 생성한 후에만 퍼블릭 NAT 게이트웨이에 탄력적 IP 주소를 추가할 수 있습니다.
- NAT 게이트웨이는 IPv4 주소를 최대 8개(기본 IPv4 주소 1개 및 보조 IPv4 주소 7개) 연결할 수 있습니다. 프라이빗 NAT 게이트웨이에 최대 8개의 프라이빗 IPv4 주소를 할당할 수 있습니다. 기본적으로 2개의 탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하는 것으로 제한됩니다. 할당량 조정을 요청하여 이 제한을 늘릴 수 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.

보조 IPv4 주소 연결을 편집하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 NAT gateways를 선택합니다.
3. 보조 IPv4 주소 연결을 편집하려는 NAT 게이트웨이를 선택합니다.
4. 작업을 선택한 다음 보조 IP 주소 연결 편집을 선택합니다.
5. 프라이빗 NAT 게이트웨이의 보조 IPv4 주소 연결을 편집하는 경우 작업에서 새 IPv4 주소 할당 또는 기존 IPv4 주소 할당 해제를 선택합니다. 퍼블릭 NAT 게이트웨이의 보조 IPv4 주소 연결을 편집하는 경우 작업에서 새 IPv4 주소 연결 또는 기존 IPv4 주소 연결 해제를 선택합니다.
6. 다음 중 하나를 수행하세요.

   - 새 IPv4 주소를 할당하거나 연결하도록 선택한 경우 다음을 수행합니다.

     1. 이 단계는 필수입니다. 프라이빗 IPv4 주소를 선택해야 합니다. 프라이빗 IPv4 주소 할당 방법을 선택합니다.

        - 자동 할당: AWS가 기본 프라이빗 IPv4 주소를 자동으로 선택합니다. AWS가 NAT 게이트웨이에 할당할 보조 프라이빗 IPv4 주소를 최대 7개까지 할당할지 선택합니다. AWS는 NAT 게이트웨이가 있는 서브넷에서 무작위로 주소를 자동으로 선택하고 할당합니다.

- 사용자 지정: 기본 프라이빗 IPv4 주소와 최대 7개의 보조 프라이빗 IPv4 주소를 선택하여 NAT 게이트웨이에 할당합니다.

2. 탄력적 IP 할당 ID에서 보조 IPv4 주소로 추가할 탄력적 IP 주소를 선택합니다. 이 단계는 필수입니다. 프라이빗 IPv4 주소와 함께 탄력적 IP 주소를 선택해야 합니다. 프라이빗 IP 주소 할당 방법으로 사용자 지정을 선택한 경우 추가하는 각 탄력적 IP 주소의 프라이빗 IPv4 주소도 입력해야 합니다.

> **Important**
>
> 퍼블릭 NAT 게이트웨이에 보조 EIP를 할당하는 경우 EIP의 네트워크 경계 그룹은 반드시 퍼블릭 NAT 게이트웨이가 있는 가용 영역의 네트워크 경계 그룹과 일치해야 합니다. 동일하지 않을 경우 EIP가 할당되지 않습니다. 서브넷의 세부 정보를 확인하면 해당 서브넷 AZ의 네트워크 경계 그룹을 확인할 수 있습니다. 마찬가지로 EIP 주소의 세부 정보를 확인하면 EIP의 네트워크 경계 그룹을 확인할 수 있습니다. 자세한 내용은 1. 탄력적 IP 주소 할당 섹션을 참조하세요.

NAT 게이트웨이는 IP 주소를 8개까지 연결할 수 있습니다. 퍼블릭 NAT 게이트웨이인 경우 리전당 탄력적 IP 주소에 대한 기본 할당량 제한이 있습니다. 자세한 내용은 탄력적 IP 주소 섹션을 참조하세요.

- 새 IPv4 주소를 할당 해제하거나 연결 해제하도록 선택한 경우 다음을 완료합니다.

  1. 할당 해제할 기존 보조 IP 주소에서 할당 해제하려는 보조 IP 주소를 선택합니다.
  2. (선택 사항) 연결 드레인 기간에 연결이 아직 진행 중인 경우 IP 주소를 강제로 해제하기 전에 대기할 최대 시간(초)을 입력합니다. 값을 입력하지 않으면 기본값은 350초입니다.

7. 변경 사항 저장을 선택합니다.

NAT 게이트웨이의 상태가 `Failed` 상태로 바뀌면 생성 중에 오류가 발생한 것입니다. 자세한 내용은 NAT 게이트웨이 생성 실패 단원을 참조하세요.

NAT 게이트웨이 태그 지정

NAT 게이트웨이에 태그를 지정하면 조직의 요구에 따라 이를 식별 또는 분류할 수 있습니다. 태그 사용에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 리소스에 태깅을 참조하세요.

NAT 게이트웨이는 비용 할당 태그를 지원합니다. 따라서 태그를 사용하여 AWS 청구서를 구성하고 고유한 원가 구조를 반영할 수도 있습니다. 자세한 내용은 AWS Billing 사용 설명서에서 비용 할당 태

그 사용을 참조하세요. 태그를 사용한 비용 할당 보고서 설정에 대한 자세한 내용은 AWS 계정 결제 정보의 월간 비용 할당 보고서를 참조하세요.

NAT 게이트웨이에 태그를 지정하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 NAT 게이트웨이를 선택합니다.
3. 태그를 지정하려는 NAT 게이트웨이를 선택하고 작업을 선택합니다. 태그 관리를 선택합니다.
4. 새 태그 추가를 선택하고 태그의 키와 값을 정의합니다. 최대 50개의 태그를 추가할 수 있습니다.
5. 저장을 선택합니다.

NAT 게이트웨이 삭제

NAT 게이트웨이가 더 이상 필요하지 않으면 삭제할 수 있습니다. NAT 게이트웨이를 삭제하면 해당 항목은 한 시간 동안 Amazon VPC 콘솔에 표시된 후 자동으로 제거됩니다. 이 항목을 직접 제거할 수는 없습니다.

NAT 게이트웨이를 삭제하면 탄력적 IP 주소가 연결 해제되지만 계정에서 주소가 해제되지는 않습니다. NAT 게이트웨이를 삭제하면 NAT 게이트웨이 경로는 경로를 삭제하거나 업데이트할 때까지 `blackhole` 상태로 유지됩니다.

NAT 게이트웨이를 삭제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 NAT 게이트웨이를 선택합니다.
3. NAT 게이트웨이에 대한 라디오 버튼을 선택한 후, 작업, NAT 게이트웨이 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제를 선택합니다.
5. 퍼블릭 NAT 게이트웨이와 연결된 탄력적 IP 주소가 더 이상 필요 없는 경우 해당 주소를 릴리스하는 것이 좋습니다. 자세한 내용은 5. 탄력적 IP 주소 릴리스 섹션을 참조하세요.

명령줄 개요

이 페이지에서 설명한 작업은 명령줄을 사용하여 수행할 수 있습니다.

프라이빗 NAT 게이트웨이에 프라이빗 IPv4 주소 할당

- assign-private-nat-gateway-address(AWS CLI)

- Register-EC2PrivateNatGatewayAddress(AWS Tools for Windows PowerShell)

탄력적 IP 주소 및 프라이빗 IPv4 주소를 퍼블릭 NAT 게이트웨이와 연결

- associate-nat-gateway-address(AWS CLI)
- Register-EC2NatGatewayAddress(AWS Tools for Windows PowerShell)

NAT 게이트웨이 만들기

- create-nat-gateway(AWS CLI)
- New-EC2NatGateway(AWS Tools for Windows PowerShell)

NAT 게이트웨이 삭제

- delete-nat-gateway(AWS CLI)
- Remove-EC2NatGateway(AWS Tools for Windows PowerShell)

NAT 게이트웨이 설명

- describe-nat-gateways(AWS CLI)
- Get-EC2NatGateway(AWS Tools for Windows PowerShell)

퍼블릭 NAT 게이트웨이에서 보조 탄력적 IP 주소 연결 해제

- disassociate-nat-gateway-address(AWS CLI)
- Unregister-EC2NatGatewayAddress(AWS Tools for Windows PowerShell)

NAT 게이트웨이 태그 지정

- create-tags(AWS CLI)
- New-EC2Tag(AWS Tools for Windows PowerShell)

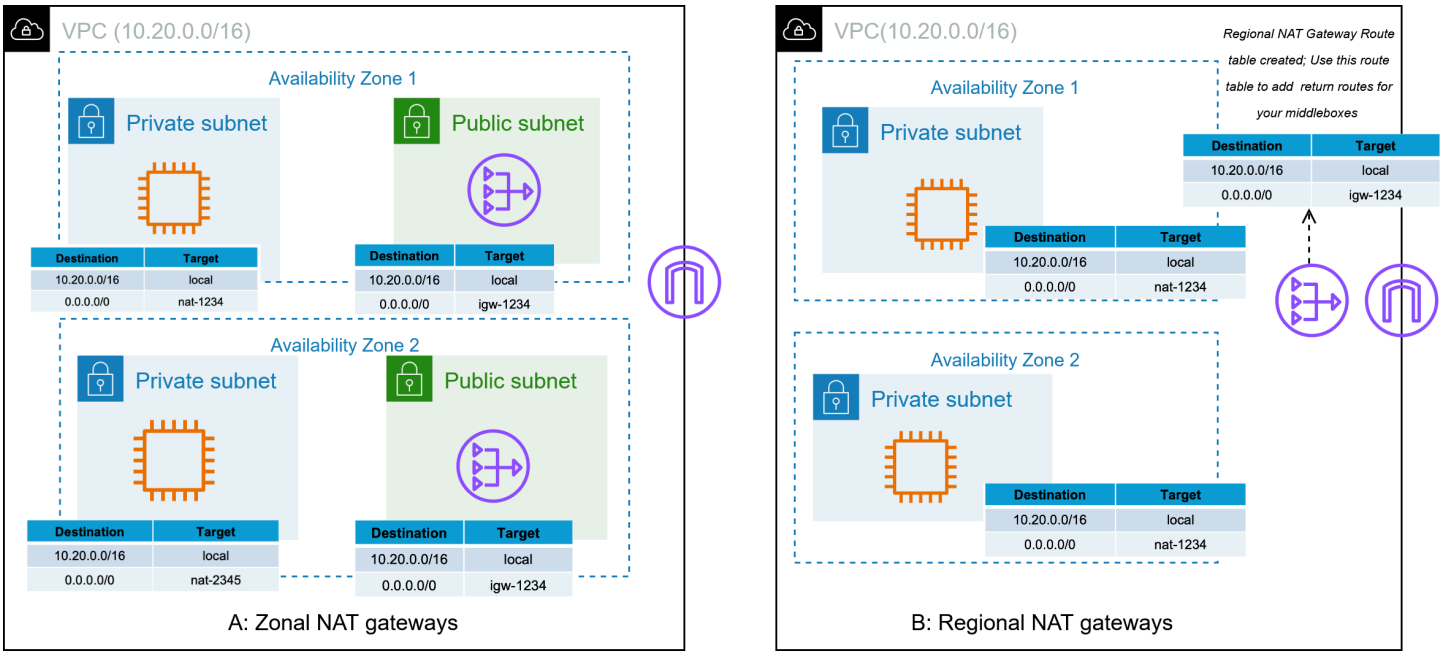
프라이빗 NAT 게이트웨이에서 보조 IPv4 주소 할당 해제

- unassign-private-nat-gateway-address(AWS CLI)

- Unregister-EC2PrivateNatGatewayAddress(AWS Tools for Windows PowerShell)

자동 다중 AZ 확장을 위한 리전 NAT 게이트웨이

네트워크 아키텍처를 단순화하고, 보안 태세를 개선하고, 기본적으로 고가용성을 구성하려는 경우 리전 NAT 게이트웨이를 사용하세요. 리전 NAT 게이트웨이는 워크로드 존재 여부에 따라 가용 영역 전체에 자동으로 확장됩니다. 단일 가용 영역에서 작동하는 표준 NAT 게이트웨이(영역 NAT 게이트웨이라고 함)와 달리, 리전 NAT 게이트웨이는 워크로드를 따라 자동 고가용성을 제공합니다.



왼쪽의 다이어그램 A는 영역 NAT 게이트웨이를 사용한 현재 설정을 나타냅니다. 먼저 가용 영역별로 영역 NAT 게이트웨이를 생성하고 퍼블릭 서브넷에 NAT를 호스팅합니다. 그런 다음 가용 영역별로 프라이빗 서브넷에서 해당 가용 영역의 NAT까지 별도의 경로를 구성합니다. 고가용성을 위해 워크로드가 새 가용 영역으로 확장될 때마다 이 단계를 반복합니다. 또한 가용 영역별로 NAT 서브넷의 라우팅 테이블에 인터넷 게이트웨이의 경로를 추가해야 합니다.

반면 리전 NAT 게이트웨이를 사용하면 퍼블릭 서브넷을 생성하여 호스팅할 필요가 없습니다. 또한 워크로드가 새 가용 영역으로 확장될 때마다 NAT 게이트웨이를 생성 및 삭제하고 라우팅 테이블을 편집할 필요도 없습니다. 대신 리전 모드를 사용하여 NAT 게이트웨이를 생성하고 VPC를 선택하기만 하면 워크로드의 존재 여부에 따라 모든 AZ에서 자동으로 확장 및 축소되어 고가용성을 제공합니다. 다이어그램 B와 같이 모든 AZ의 프라이빗 서브넷에 있는 리소스에서 이 단일 리전 NAT 게이트웨이 ID로 트래픽을 라우팅하거나, AZ의 서브넷에 있는 동일한 라우팅 테이블을 사용하여 네트워크 주소 변환을 수행할 수 있습니다. 리전 NAT 게이트웨이를 생성하면 AWS는 사전 구성된 인터넷 게이트웨이로의 경로

와 함께 제공되는 라우팅 테이블을 자동으로 생성합니다. 이 라우팅 테이블을 사용하여 미들박스로의 반환 경로를 추가할 수 있습니다.

이점

리전 NAT 게이트웨이는 다음과 같은 이점을 제공합니다.

- 설정 간소화 - 네트워크 인터페이스가 있는 모든 가용 영역에서 단일 NAT ID를 사용하므로 서로 다른 가용 영역에 있는 서브넷에 동일한 라우팅 항목을 사용할 수 있습니다.
- 보안 향상 - 퍼블릭 서브넷이 필요하지 않습니다. 리전 NAT 게이트웨이는 자체 라우팅 테이블이 있는 독립형 리소스이며 리전 NAT 게이트웨이를 호스팅하기 위해 VPC에 퍼블릭 서브넷이 필요하지 않으므로, 퍼블릭 연결이 있는 서브넷에서 프라이빗 리소스를 잘못 구성할 가능성이 줄어듭니다.
- 자동 고가용성 - 워크로드 공간에 따라 자동으로 확장 및 축소되여 영역 친화도를 유지하므로 기본적으로 고가용성을 제공합니다.
- 더 높은 포트 및 IP 제한 - 리전 NAT 게이트웨이는 가용 영역당 최대 32개의 IP 주소를 지원합니다(영역 NAT 게이트웨이의 경우 8개). 각 IP 주소는 인기 있는 대상(대상 IP, 대상 포트 및 프로토콜의 고유한 조합으로 식별됨)에 대한 동시 연결 제한을 55,000만큼 늘립니다.

리전 NAT 게이트웨이를 사용해야 하는 경우

프라이빗 연결이 필요한 경우를 제외한 모든 사용 사례에 리전 NAT 게이트웨이를 사용하는 것이 좋습니다. 리전 NAT 게이트웨이는 프라이빗 연결을 제공하지 않으므로 프라이빗 NAT 사용 사례의 경우 영역 가용성 모드에서 NAT 게이트웨이를 사용하는 것이 좋습니다.

리전 NAT 게이트웨이 작동 방식

새 가용 영역에서 리소스를 시작하면 리전 NAT 게이트웨이가 해당 가용 영역에 네트워크 인터페이스(ENI)가 존재하는지 감지하고 자동으로 해당 영역으로 확장됩니다. 마찬가지로 활성 워크로드가 없는 가용 영역에서는 NAT 게이트웨이가 축소됩니다.

리소스가 새 가용 영역에서 인스턴스화된 후 리전 NAT 게이트웨이가 해당 영역으로 확장되는 데 최대 60분이 걸릴 수 있습니다. 이 확장이 완료될 때까지 이 리소스의 관련 트래픽은 기존 가용 영역 중 하나의 리전 NAT 게이트웨이를 통해 영역 간에 처리됩니다.

리전 NAT 게이트웨이는 다음 두 가지 모드를 지원합니다.

- 자동 모드 - 이 모드에서는 AWS가 IP 주소 및 가용 영역 확장을 자동으로 관리합니다(권장). 이 모드에서 자체 IP 주소를 사용하려고 하고 Amazon VPC IPAM을 사용하는 경우 Amazon VPC IPAM 사용 설명서의 IPAM 정책을 사용하여 퍼블릭 IPv4 할당 전략 정의를 참조하세요.

- 수동 모드 - 이 모드에서는 수동으로 IP 주소를 관리하고 각 가용 영역에 대한 네트워크 주소 변환을 제어합니다. 수동 모드에서는 사용자가 가용 영역 전체에서 NAT 게이트웨이를 확장 및 축소해야 합니다.

> **Important**
>
> 리전 NAT 게이트웨이는 리전 NAT 게이트웨이 라우팅 테이블에서 AWS 전송 게이트웨이를 유효한 경로로 지원합니다. 리전 NAT 게이트웨이는 프라이빗 NAT를 지원하지 않습니다. 프라이빗 NAT가 필요한 경우 영역 NAT 게이트웨이를 대신 사용하세요. 리전 NAT 게이트웨이는 제약이 있는 가용 영역에서 지원되지 않습니다.

가격 책정

요금에 대한 자세한 내용은 Amazon VPC 요금을 참조하세요.

리전 NAT 게이트웨이 생성

콘솔 사용

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 NAT 게이트웨이를 선택합니다.
3. NAT 게이트웨이 생성을 선택합니다.
4. 가용 모드에서 리전을 선택합니다. 리전 가용성을 선택할 때는 서브넷을 지정할 필요가 없습니다.
5. VPC를 선택합니다.
6. 나머지 구성을 완료하고 NAT 게이트웨이 생성을 선택합니다.

AWS CLI 사용

리전 NAT 게이트웨이 생성

```
aws ec2 create-nat-gateway --vpc-id vpc-12345678 --availability-mode regional
```

NAT 게이트웨이 세부 정보 보기

```
aws ec2 describe-nat-gateways --nat-gateway-ids nat-12345678
```

IP 주소 추가(수동 모드)

```
aws ec2 associate-nat-gateway-address --nat-gateway-id nat-12345678 --availability-zone
 us-east-1b --allocation-ids eipalloc-12345678
```

IP 주소 제거

```
aws ec2 disassociate-nat-gateway-address --nat-gateway-id nat-12345678 --association-
ids eipassoc-12345678
```

리전 NAT 게이트웨이 삭제

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-12345678
```

영역에서 리전 NAT 게이트웨이로 변환

> **Important**
>
> 이렇게 하면 기존 연결이 재설정됩니다. 유지 관리 기간에 이 단계를 완료하는 것이 좋습니다.

다음 두 가지 방법 중 하나를 사용하여 기존 영역 NAT 게이트웨이를 리전 NAT 게이트웨이로 변환할 수 있습니다.

새 IP 주소를 사용하는 리전 NAT 게이트웨이를 사용하려는 경우

1. 새 리전 NAT 게이트웨이 생성
2. 리전 NAT 게이트웨이를 가리키도록 라우팅 테이블 업데이트
3. 이전 영역 NAT 게이트웨이 삭제

이 방법은 새 IP 주소를 사용하고 경로가 업데이트될 때 기존 연결을 재설정합니다.

기존 IP 주소를 리전 NAT 게이트웨이에서 재사용하려는 경우

1. 기존 영역 NAT 게이트웨이를 삭제하여 IP 주소 해제
2. 해제된 IP 주소를 사용하여 리전 NAT 게이트웨이 생성
3. 리전 NAT 게이트웨이를 가리키도록 라우팅 테이블 업데이트

이 방법은 IP 주소를 유지하지만 전환 중에 트래픽이 중단되므로 유지 관리 기간이 필요합니다.

NAT Gateway 사용 사례

다음은 퍼블릭 및 프라이빗 NAT 게이트웨이의 사용 사례입니다.

시나리오

- 프라이빗 서브넷에서 인터넷 액세스
- 허용 목록에 있는 IP 주소를 사용하여 네트워크에 액세스
- 중첩되는 네트워크 간에 통신 사용

프라이빗 서브넷에서 인터넷 액세스

퍼블릭 NAT 게이트웨이를 사용하여 프라이빗 서브넷의 인스턴스가 아웃바운드 트래픽을 인터넷으로 전송할 수 있도록 하는 동시에 인터넷이 인스턴스에 대한 연결을 설정하는 것을 방지할 수 있습니다.

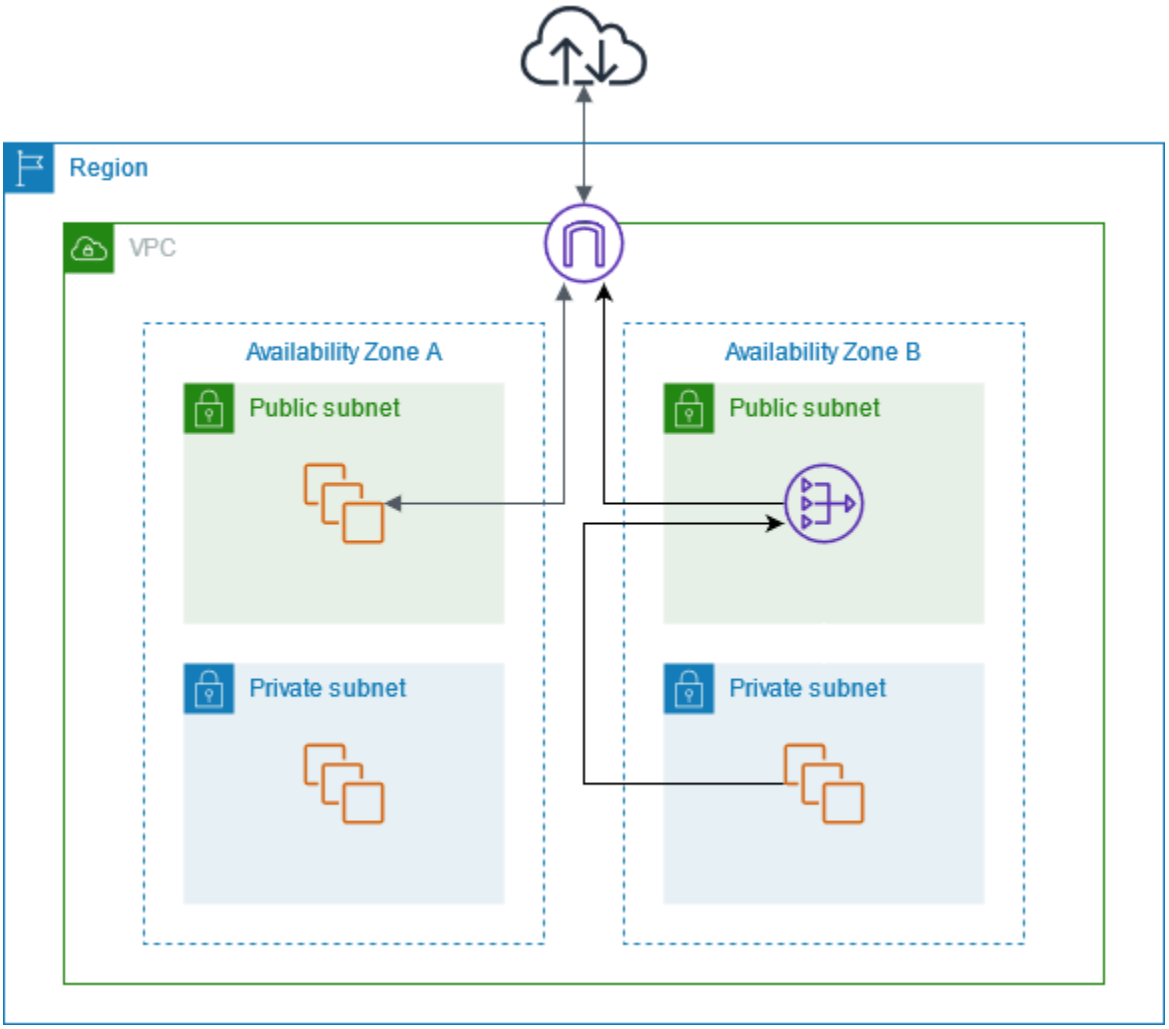
내용

- 개요
- 라우팅
- 퍼블릭 NAT 게이트웨이 테스트

개요

다음 다이어그램에서 이 사용 사례를 보여줍니다. 각 가용 영역에 2개의 서브넷이 있는 2개의 가용 영역이 있습니다. 각 서브넷에 대한 라우팅 테이블에 따라 트래픽이 라우팅되는 방법이 결정됩니다. 가용 영역 A에서 퍼블릭 서브넷의 인스턴스는 인터넷 게이트웨이에 대한 경로를 통해 인터넷에 연결할 수 있지만 프라이빗 서브넷의 인스턴스는 인터넷에 대한 경로가 없습니다. 가용 영역 B에서 퍼블릭 서브넷은 NAT 게이트웨이를 포함하고 프라이빗 서브넷의 인스턴스는 퍼블릭 서브넷의 NAT 게이트웨이 경로를 통해 인터넷에 연결할 수 있습니다.

프라이빗 및 퍼블릭 NAT 게이트웨이는 모두 인스턴스의 소스 프라이빗 IPv4 주소를 프라이빗 NAT 게이트웨이의 프라이빗 IPv4 주소에 매핑하지만 퍼블릭 NAT 게이트웨이의 경우 인터넷 게이트웨이는 퍼블릭 NAT 게이트웨이의 프라이빗 IPv4 주소를 NAT 게이트웨이와 연결된 엘라스틱 IP 주소에 매핑합니다. 인스턴스에 응답 트래픽을 전송할 때 인스턴스는 퍼블릭 또는 프라이빗 NAT 게이트웨어 여부에 상관없이 NAT 게이트웨이는 주소를 원래 소스 IP 주소로 다시 변환합니다.



가용 영역 A의 프라이빗 서브넷에 있는 인스턴스도 인터넷에 연결해야 하는 경우 이 서브넷에서 가용 영역 B의 NAT 게이트웨이로 가는 경로를 생성할 수 있습니다. 또는 인터넷 액세스가 필요한 리소스가 포함된 각 가용 영역에 NAT 게이트웨이를 생성하여 복원력을 향상시킬 수 있습니다. 예시 다이어그램은 the section called “프라이빗 서버” 섹션을 참조하세요.

라우팅

다음은 가용 영역 A의 퍼블릭 서브넷과 연결된 라우팅 테이블입니다. 첫 번째 항목은 로컬 경로입니다. 서브넷의 인스턴스가 프라이빗 IP 주소를 사용하여 VPC의 다른 인스턴스와 통신할 수 있도록 합니다. 두 번째 항목은 다른 모든 서브넷 트래픽을 인터넷 게이트웨이로 전송하여 서브넷의 인스턴스가 인터넷에 액세스할 수 있도록 합니다.

대상 주소	대상
VPC CIDR	로컬

대상 주소	대상
0.0.0.0/0	*internet-gateway-id*

다음은 가용 영역 A의 프라이빗 서브넷과 연결된 라우팅 테이블입니다. 첫 번째 항목은 로컬 경로이며 이 경로는 서브넷의 인스턴스가 프라이빗 IP 주소를 사용하여 VPC의 다른 인스턴스와 통신할 수 있도록 합니다. 이 서브넷의 인스턴스에는 인터넷에 대한 액세스 권한이 없습니다.

대상 주소	대상
VPC CIDR	로컬

다음은 가용 영역 B의 퍼블릭 서브넷과 연결된 라우팅 테이블입니다. 첫 번째 항목은 로컬 경로이며, 이 경로는 서브넷의 인스턴스가 프라이빗 IP 주소를 사용하여 VPC의 다른 인스턴스와 통신할 수 있도록 합니다. 두 번째 항목은 다른 모든 서브넷 트래픽을 인터넷 게이트웨이로 전송하여 서브넷의 NAT 게이트웨이가 인터넷에 액세스할 수 있도록 합니다.

대상 주소	대상
VPC CIDR	로컬
0.0.0.0/0	*internet-gateway-id*

다음은 가용 영역 B의 프라이빗 서브넷과 연결된 라우팅 테이블입니다. 첫 번째 항목은 로컬 경로입니다. 서브넷의 인스턴스가 프라이빗 IP 주소를 사용하여 VPC의 다른 인스턴스와 통신할 수 있도록 합니다. 두 번째 항목에서는 기타 서브넷 트래픽을 모두 NAT 게이트웨이로 전송합니다.

대상 주소	대상
VPC CIDR	로컬
0.0.0.0/0	*nat-gateway-id*

자세한 내용은 the section called “서브넷 라우팅 테이블 관리” 섹션을 참조하세요.

퍼블릭 NAT 게이트웨이 테스트

NAT 게이트웨이를 만들고 라우팅 테이블을 업데이트한 후에는 프라이빗 서브넷의 인스턴스에서 인터넷의 원격 주소를 ping하여 인터넷에 연결할 수 있는지 테스트할 수 있습니다. 이렇게 하는 방법의 예는 인터넷 연결 테스트 섹션을 참조하세요.

인터넷에 연결할 수 있는 경우 인터넷 트래픽이 NAT 게이트웨이를 통해 라우팅되는지도 확인할 수 있습니다.

- 프라이빗 서브넷의 인스턴스에서 전송되는 트래픽의 경로를 추적합니다. 이렇게 하려면 프라이빗 서브넷의 Linux 인스턴스에서 `traceroute` 명령을 실행합니다. 출력에서 홉 중 하나(일반적으로 첫 번째 홉)에 NAT 게이트웨이의 프라이빗 IP 주소가 보여야 합니다.
- 프라이빗 서브넷의 인스턴스에서 소스 IP 주소에 연결할 때 이를 표시하는 타사 웹 사이트 또는 도구를 사용합니다. 소스 IP 주소는 NAT 게이트웨이의 탄력적 IP 주소여야 합니다.

이러한 테스트가 실패하는 경우 NAT 게이트웨이 문제 해결 섹션을 참조하세요.

인터넷 연결 테스트

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

1. 퍼블릭 서브넷에서 인스턴스를 시작합니다(이 인스턴스를 Bastion Host로 사용). 시작 마법사에서 Amazon Linux AMI를 선택하고 인스턴스에 퍼블릭 IP 주소를 할당해야 합니다.

   보안 그룹 규칙에서 로컬 네트워크 IP 주소 범위의 인바운드 SSH 트래픽과 프라이빗 서브넷 IP 주소 범위로 향하는 아웃바운드 SSH 트래픽을 허용하는지 확인하세요. 이 테스트에서는 인바운드 및 아웃바운드 SSH 트래픽 모두에 `0.0.0.0/0`을 사용할 수도 있습니다.
2. 프라이빗 서브넷에서 인스턴스를 시작합니다. 시작 마법사에서 Amazon Linux AMI를 선택해야 합니다. 인스턴스에 퍼블릭 IP 주소를 할당하지 마세요. 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 Bastion Host에 연결합니다. 자세한 내용은 Linux 또는 macOS에 대한 SSH 에이전트 전달을 구성하려면 또는 Windows에 대한 SSH 에이전트 전달 구성 단원을 참조하세요.
4. Bastion Host에서 프라이빗 서브넷의 인스턴스에 연결한 다음, 프라이빗 서브넷의 인스턴스에서 인터넷 연결을 테스트합니다. 자세한 내용은 인터넷 연결을 테스트하려면 단원을 참조하세요.

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. 다음 예시와 같이 `-A` 옵션으로 퍼블릭 서브넷의 인스턴스에 연결하여 SSH 에이전트 전달을 활성화하고 해당 인스턴스의 퍼블릭 주소를 사용합니다.

   ```
   ssh -A ec2-user@54.0.0.123
   ```

Windows에 대한 SSH 에이전트 전달 구성

Windows에서 사용 가능한 OpenSSH 클라이언트를 사용하거나 선호하는 SSH 클라이언트(예: PuTTY)를 설치할 수 있습니다.

OpenSSH

Getting started with OpenSSH for Windows의 설명에 따라 Windows용 OpenSSH를 설치합니다. 그런 다음 인증 에이전트에 키를 추가합니다. 자세한 내용은 Key-based authentication in OpenSSH for Windows를 참조하세요.

PuTTY

1. Pageant가 아직 설치되어 있지 않으면 PuTTY 다운로드 페이지에서 Pageant를 다운로드하여 설치합니다.
2. 프라이빗 키를 .ppk 형식으로 변환합니다. 자세한 내용은 Amazon EC2 사용 설명서의 PuTTYgen을 사용하여 프라이빗 키 변환을 참조하세요.
3. Pageant를 실행하고 작업 표시줄의 Pageant 아이콘(숨겨져 있을 수 있음)을 마우스 오른쪽 버튼으로 클릭한 후 Add Key를 선택하세요. 생성한 .ppk 파일을 선택하고 필요에 따라 암호를 입력한 다음 열기(Open)를 선택합니다.

4. PuTTY 세션을 시작하고 퍼블릭 IP 주소를 사용하여 퍼블릭 서브넷의 인스턴스에 연결합니다. 자세한 내용은 PuTTY를 사용하여 Linux 인스턴스에 연결을 참조하세요. [Auth] 범주에서 [Allow agent forwarding] 옵션을 선택하고 [Private key file for authentication] 상자를 공백 상태로 둡니다.

인터넷 연결을 테스트하려면

1. 다음 예시와 같이 퍼블릭 서브넷의 인스턴스에서 프라이빗 IP 주소를 사용하여 프라이빗 서브넷의 인스턴스에 연결합니다.

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

2. 프라이빗 인스턴스에서 ICMP가 활성화된 웹 사이트에 대해 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 명령이 실패할 경우 인스턴스에서 인터넷에 액세스할 수 없음을 참조하세요.

3. (선택 사항) 더 이상 필요하지 않으면 인스턴스를 종료합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 종료를 참조하세요.

허용 목록에 있는 IP 주소를 사용하여 네트워크에 액세스

프라이빗 NAT 게이트웨이를 통해 허용 목록에 있는 주소 풀을 사용하여 VPC에서 온프레미스 네트워크로의 통신을 사용할 수 있습니다.

각 인스턴스에 허용 목록 범위의 개별 IP 주소를 할당하는 대신, 허용 목록 범위의 IP 주소가 구성된 프라이빗 NAT 게이트웨이를 통해 서브넷 트래픽을 라우팅할 수 있습니다.

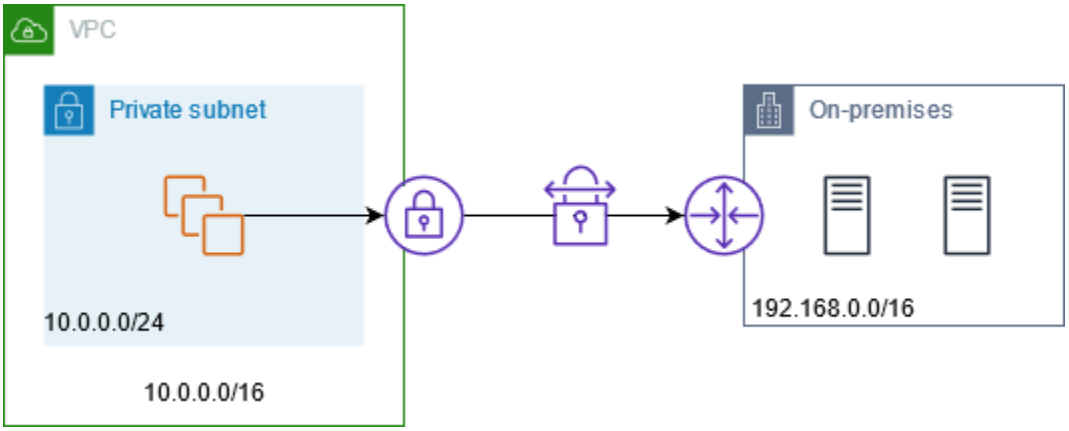
내용

- 개요
- 리소스

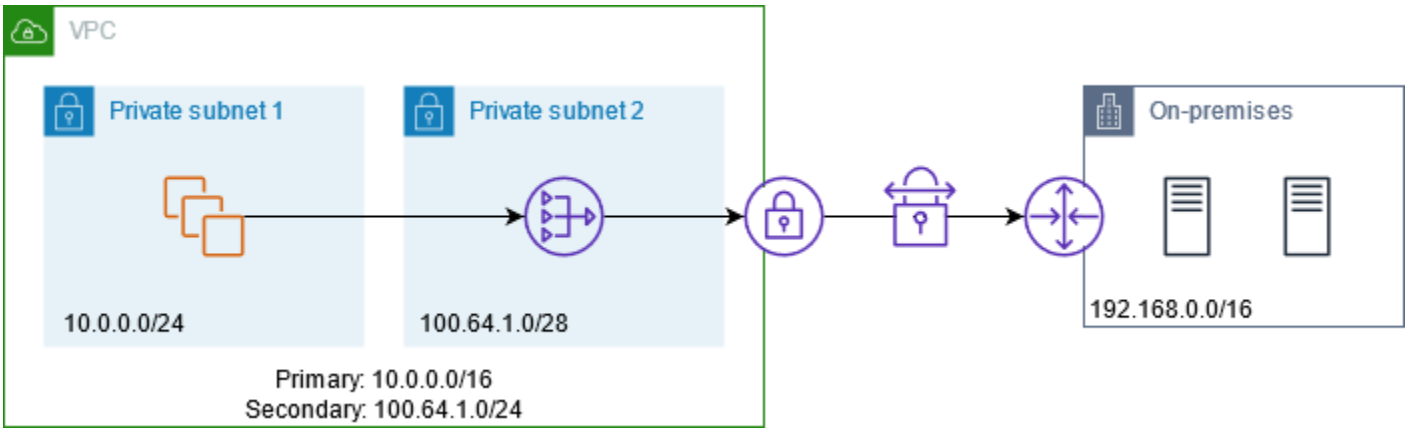
- 라우팅

개요

다음 다이어그램은 인스턴스가 Site-to-Site VPN을 통해 온프레미스 리소스에 액세스할 수 있는 방법을 보여줍니다. 인스턴스의 트래픽은 VPN 연결을 통해 가상 프라이빗 게이트웨이로 라우팅되고 고객 게이트웨이로 라우팅된 후 온프레미스 네트워크의 대상으로 라우팅됩니다. 그러나 대상에서 100.64.1.0/28과 같은 특정 IP 주소 범위의 트래픽만 허용된다고 가정해 보겠습니다. 이 경우에는 이러한 인스턴스의 트래픽이 온프레미스 네트워크에 도달하지 못하게 됩니다.



다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다. VPC에는 원본 IP 주소 범위와 함께 허용된 IP 주소 범위가 있습니다. VPC에는 프라이빗 NAT 게이트웨이가 있는 허용된 IP 주소 범위의 서브넷이 있습니다. 온프레미스 네트워크로 향하는 인스턴스의 트래픽은 VPN 연결로 라우팅되기 전에 NAT 게이트웨이로 전송됩니다. 온프레미스 네트워크는 허용된 IP 주소 범위에서 제공되는 NAT 게이트웨이의 소스 IP 주소가 있는 인스턴스에서 트래픽을 수신합니다.



리소스

다음과 같이 리소스를 생성하거나 업데이트합니다.

- 허용된 IP 주소 범위를 VPC와 연결합니다.
- 허용된 IP 주소 범위의 VPC에서 서브넷을 생성합니다.
- 새 서브넷에서 프라이빗 NAT 게이트웨이를 생성합니다.
- 서브넷의 라우팅 테이블을 인스턴스로 업데이트하여 온프레미스 네트워크로 향하는 트래픽을 NAT 게이트웨이로 전송합니다. 온프레미스 네트워크로 향하는 트래픽을 가상 프라이빗 게이트웨이로 전송하는 프라이빗 NAT 게이트웨이가 있는 서브넷의 라우팅 테이블에 경로를 추가합니다.

라우팅

다음은 첫 번째 서브넷과 연결된 라우팅 테이블입니다. 각 VPC CIDR에 대한 로컬 경로가 있습니다. 로컬 경로를 사용하면 서브넷의 리소스가 프라이빗 IP 주소를 사용하여 VPC의 다른 리소스와 통신할 수 있습니다. 세 번째 항목은 온프레미스 네트워크로 향하는 트래픽을 프라이빗 NAT 게이트웨이로 전송합니다.

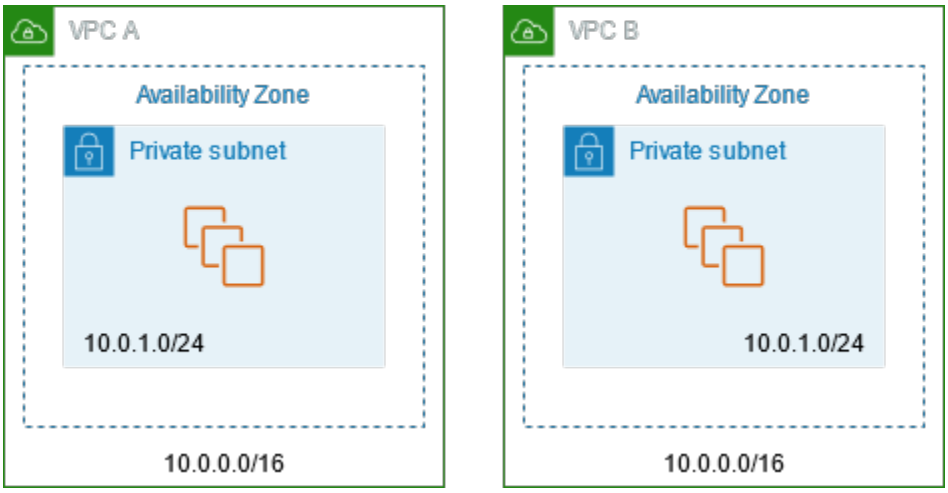
대상 주소	대상
10.0.0.0/16	로컬
100.64.1.0/24	로컬
192.168.0.0/16	*nat-gateway-id*

다음은 두 번째 서브넷과 연결된 라우팅 테이블입니다. 각 VPC CIDR에 대한 로컬 경로가 있습니다. 로컬 경로를 사용하면 서브넷의 리소스가 프라이빗 IP 주소를 사용하여 VPC의 다른 리소스와 통신할 수 있습니다. 세 번째 항목은 온프레미스 네트워크로 향하는 트래픽을 가상 프라이빗 게이트웨이로 전송합니다.

대상 주소	대상
10.0.0.0/16	로컬
100.64.1.0/24	로컬
192.168.0.0/16	*vgw-id*

중첩되는 네트워크 간에 통신 사용

프라이빗 NAT 게이트웨이를 사용하여 네트워크 간에 CIDR 범위가 중첩되더라도 통신할 수 있습니다. 예를 들어, VPC A의 인스턴스가 VPC B의 인스턴스에서 제공되는 서비스에 액세스해야 한다고 가정해 보십시오.



내용

- 개요
- 리소스
- 라우팅

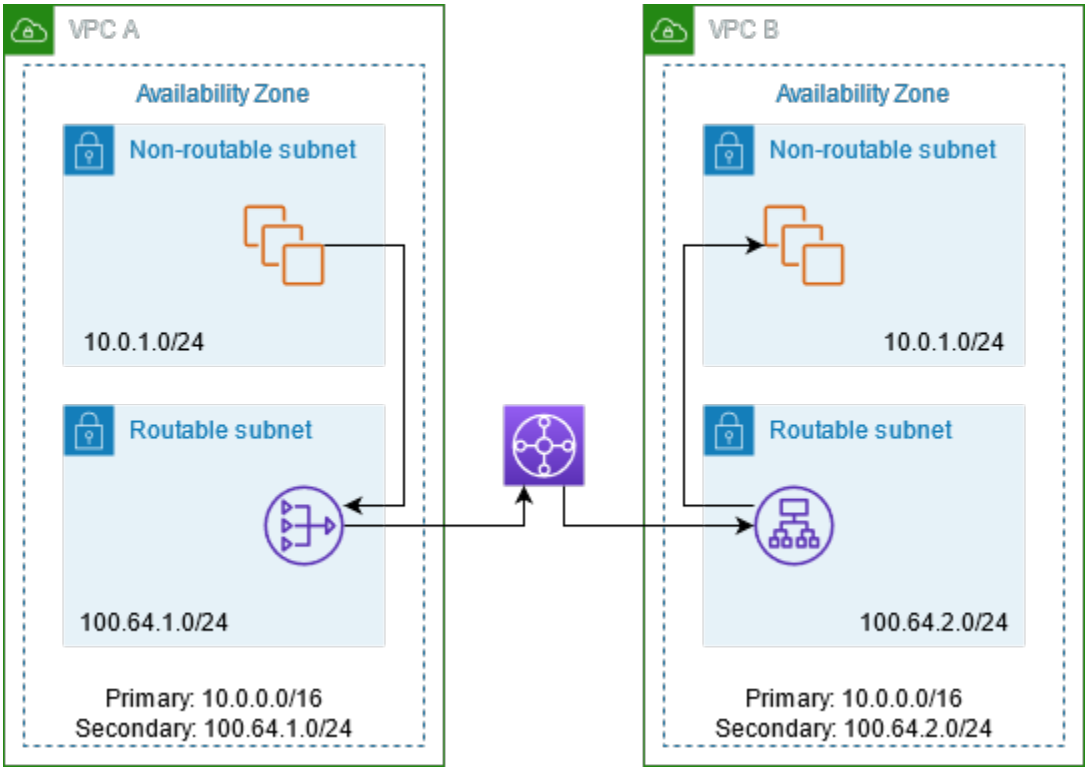
개요

다음 다이어그램은 이 시나리오를 위한 구성의 주요 구성 요소를 보여줍니다. 먼저 IP 관리 팀이 중첩될 수 있는 주소 범위(라우팅 불가능한 주소 범위) 및 중첩될 수 없는 주소 범위(라우팅 가능한 주소 범위)를 결정합니다. IP 관리 팀은 라우팅 가능한 주소 범위 풀에서 요청이 있을 경우 프로젝트에 주소 범위를 할당합니다.

각 VPC에는 라우팅이 불가능한 원래 IP 주소 범위와 IP 관리 팀에 의해 할당된 라우팅 가능한 IP 주소 범위가 있습니다. VPC A에는 프라이빗 NAT 게이트웨이가 포함된 라우팅 가능한 범위의 서브넷이 있습니다. 프라이빗 NAT 게이트웨이는 해당 서브넷에서 해당 IP 주소를 가져옵니다. VPC B에는 Application Load Balancer가 포함된 라우팅 가능한 범위의 서브넷이 있습니다. Application Load Balancer는 해당 서브넷에서 해당 IP 주소를 가져옵니다.

VPC A의 라우팅 불가 서브넷에 위치한 인스턴스에서 VPC B의 라우팅 불가 서브넷에 위치한 인스턴스로 향하는 트래픽은 프라이빗 NAT 게이트웨이를 통해 전송됩니다. 그런 다음 Transit Gateway로 라우팅됩니다.

전송 게이트웨이는 트래픽을 Application Load Balancer로 전송합니다. Application Load Balancer는 VPC B의 라우팅 불가능한 서브넷에 있는 대상 인스턴스 중 하나로 해당 트래픽을 라우팅합니다. 전송 게이트웨이에서 Application Load Balancer로의 이 트래픽에는 프라이빗 NAT 게이트웨이의 소스 IP 주소가 있습니다. 따라서 로드 밸런서의 응답 트래픽은 프라이빗 NAT 게이트웨이의 주소를 대상으로 사용합니다. 응답 트래픽은 전송 게이트웨이로 보내진 후 프라이빗 NAT 게이트웨이로 라우팅됩니다. NAT 게이트웨이는 대상을 VPC A의 라우팅 불가능한 서브넷에 있는 인스턴스로 변환합니다.



리소스

다음과 같이 리소스를 생성하거나 업데이트합니다.

- 할당된 라우팅 가능한 IP 주소 범위를 각각의 해당 VPC에 연결합니다.
- 라우팅 가능한 IP 주소 범위의 VPC A에서 서브넷을 생성하고 이 새 서브넷에서 프라이빗 NAT 게이트웨이를 생성합니다.
- 라우팅 가능한 IP 주소 범위의 VPC B에서 서브넷을 생성하고 이 새 서브넷에서 Application Load Balancer를 생성합니다. 로드 밸런서의 대상 그룹에 라우팅 불가능한 서브넷의 인스턴스를 등록합니다.
- VPC를 연결할 전송 게이트웨이를 생성합니다. 경로 전파를 사용 중지했는지 확인합니다. 각 VPC를 전송 게이트웨이에 연결하는 경우에는 VPC의 라우팅 가능한 주소 범위를 사용하세요.
- VPC A의 라우팅 불가능한 서브넷의 라우팅 테이블을 업데이트하여 VPC B의 라우팅 가능한 주소 범위로 향하는 모든 트래픽을 프라이빗 NAT 게이트웨이로 전송합니다. VPC A의 라우팅 가능한 서

브넷의 라우팅 테이블을 업데이트하여 VPC B의 라우팅 가능한 주소 범위로 향하는 모든 트래픽을 전송 게이트웨이로 전송합니다.

- VPC B의 라우팅 가능한 서브넷의 라우팅 테이블을 업데이트하여 VPC A의 라우팅 가능한 주소 범위로 향하는 모든 트래픽을 전송 게이트웨이로 전송합니다.

라우팅

다음은 VPC A의 라우팅 불가능한 서브넷에 대한 라우팅 테이블입니다.

대상 주소	대상
10.0.0.0/16	로컬
100.64.1.0/24	로컬
100.64.2.0/24	*nat-gateway-id*

다음은 VPC A의 라우팅 가능한 서브넷에 대한 라우팅 테이블입니다.

대상 주소	대상
10.0.0.0/16	로컬
100.64.1.0/24	로컬
100.64.2.0/24	*transit-gateway-id*

다음은 VPC B의 라우팅 불가능한 서브넷에 대한 라우팅 테이블입니다.

대상 주소	대상
10.0.0.0/16	로컬
100.64.2.0/24	로컬

다음은 VPC B의 라우팅 가능한 서브넷에 대한 라우팅 테이블입니다.

대상 주소	대상
10.0.0.0/16	로컬
100.64.2.0/24	로컬
100.64.1.0/24	*transit-gateway-id*

다음은 전송 게이트웨이 라우팅 테이블입니다.

CIDR	연결	경로 유형
100.64.1.0/24	*VPC A ##*	정적
100.64.2.0/24	*VPC B ##*	정적

DNS64 및 NAT64

NAT 게이트웨이는 IPv6에서 IPv4로의 네트워크 주소 변환을 지원합니다(일반적으로 NAT64라고 함). NAT64는 IPv6 AWS 리소스가 동일한 VPC 또는 다른 VPC, 온프레미스 네트워크 또는 인터넷을 통해 IPv4 리소스와 통신하도록 지원합니다. Amazon Route 53 Resolver 기반 DNS64를 포함하여 NAT64를 사용하거나 고유 DNS64 서버를 사용할 수 있습니다.

내용

- DNS64란 무엇입니까?
- NAT64란 무엇입니까?
- DNS64 및 NAT64 구성

DNS64란 무엇입니까?

VPC에서 실행되는 IPv6 전용 워크로드는 IPv6 네트워크 패킷만 보내고 받을 수 있습니다. DNS64 없이는 IPv4 전용 서비스에 대한 DNS 쿼리는 응답으로 IPv4 대상 주소를 출력하며 IPv6 전용 서비스와 통신할 수 없습니다. 이러한 통신 격차를 해소하기 위해 서브넷에 대해 DNS64를 활성화하면 서브넷 내의 모든 해당 AWS 리소스에 적용됩니다. DNS64에서는 Amazon Route 53 Resolver가 쿼리한 서비스에 대한 DNS 레코드를 조회하고 다음 중 하나를 수행합니다.

- 레코드에 IPv6 주소가 포함되어 있으면 원래 레코드를 반환하고 IPv6을 통한 변환 없이 연결이 설정됩니다.
- DNS 레코드의 대상과 연결된 IPv6 주소가 없는 경우 Route 53 Resolver는 레코드의 IPv4 주소에 대하여 RFC6052(`64:ff9b::/96`)에 정의된 잘 알려진 `/96` 접두사를 가장하여 주소를 합성합니다.

  IPv6 전용 서비스는 네트워크 패킷을 합성된 IPv6 주소로 보냅니다. 그런 다음 이 트래픽을 NAT 게이트웨이로 라우팅하면, NAT 게이트웨이가 필요한 주소 변환을 수행하여 서브넷 내 IPv6 서비스가 서브넷 외부의 IPv4 서비스와 통신할 수 있게 됩니다.

AWS CLI로 modify-subnet-attribute를 사용하거나 서브넷을 선택하고 작업(Actions) > 서브넷 설정 편집(Edit subnet settings)을 선택하여 VPC 콘솔에서 서브넷의 DNS64를 사용 또는 사용 중지할 수 있습니다.

NAT64란 무엇입니까?

NAT64를 사용하면 Amazon VPC의 IPv6 전용 서비스가 동일한 VPC(다른 서브넷에 있음) 또는 연결된 VPC, 온프레미스 네트워크 또는 인터넷을 통해 IPv4 전용 서비스와 통신할 수 있습니다.

NAT64는 기존 NAT 게이트웨이 또는 새로 만든 NAT 게이트웨이에서 자동으로 사용할 수 있습니다. 이 기능을 활성화하거나 비활성화할 수 없습니다. NAT 게이트웨이가 있는 서브넷은 이중 스택 서브넷이 아니어도 NAT64가 작동합니다.

DNS64를 사용하도록 설정한 후 IPv6 전용 서비스가 NAT 게이트웨이를 통해 합성된 IPv6 주소로 네트워크 패킷을 전송하면 다음과 같은 일이 발생합니다.

- `64:ff9b::/96` 접두사에서 NAT 게이트웨이는 원래 대상이 IPv4임을 인식하고 다음을 대체하여 IPv6 패킷을 IPv4로 변환합니다.
  - 인터넷 게이트웨이에 의해 탄력적 IP 주소로 변환되는 자체 프라이빗 IP가 있는 소스 IPv6
  - `64:ff9b::/96` 접두사를 잘라서 대상 IPv6에서 IPv4로 연결합니다.
- NAT 게이트웨이는 인터넷 게이트웨이, 가상 프라이빗 게이트웨이 또는 전송 게이트웨이를 통해 변환된 IPv4 패킷을 대상으로 전송하고 연결을 시작합니다.
- IPv4 전용 호스트는 IPv4 응답 패킷을 다시 반환합니다. 연결이 설정된 후 NAT 게이트웨이는 외부 호스트에서 응답 IPv4 패킷을 수락합니다.
- 응답 IPv4 패킷은 NAT 게이트웨이로 향합니다. 이 게이트웨이는 IP(대상 IP)를 호스트의 IPv6 주소로 바꾸고 소스 IPv4 주소에 `64:ff9b::/96`로 다시 가장하여 패킷을 수신합니다. 그런 다음 패킷은 로컬 경로를 따라 호스트로 흐릅니다.

이러한 방식으로 NAT 게이트웨이를 사용하면 서브넷의 IPv6 전용 워크로드가 서브넷 외부의 IPv4 전용 서비스와 통신할 수 있습니다.

DNS64 및 NAT64 구성

이 단원의 단계에 따라 IPv4 전용 서비스와의 통신을 사용하도록 DNS64 및 NAT64를 구성합니다.

내용

- AWS CLI를 통해 인터넷에서 IPv4 전용 서비스와의 통신을 활성화합니다.
- 온프레미스 환경에서 IPv4 전용 서비스와의 통신 사용

AWS CLI를 통해 인터넷에서 IPv4 전용 서비스와의 통신을 활성화합니다.

서브넷 외부의 IPv4 전용 서비스와 통신해야 하는 IPv6 전용 워크로드를 포함한 서브넷이 있는 경우, 이 예에서는 이러한 IPv6 전용 서비스가 인터넷에서 IPv4 전용 서비스와 통신하도록 설정하는 방법을 보여 줍니다.

먼저 퍼블릭 서브넷(IPv6 전용 워크로드가 포함된 서브넷과는 별개)에서 NAT 게이트웨이를 구성해야 합니다. 예를 들어 NAT 게이트웨이가 포함된 서브넷에는 인터넷 게이트웨이를 가리키는 `0.0.0.0/0` 경로가 있어야 합니다.

이러한 IPv6 전용 서비스가 인터넷에서 IPv4 전용 서비스에 연결할 수 있도록 하려면 다음 단계를 완료하세요.

1. IPv6 전용 워크로드가 포함된 서브넷의 라우팅 테이블에 다음 세 가지 경로를 추가합니다.

   - NAT 게이트웨이를 가리키는 IPv4 경로(있는 경우).
   - `64:ff9b::/96`NAT 게이트웨이를 가리키는 경로. 이렇게 하면 IPv4 전용 서비스로 향하는 IPv6 전용 워크로드의 트래픽이 NAT 게이트웨이를 통해 라우팅될 수 있습니다.
   - 송신 전용 인터넷 게이트웨이(또는 인터넷 게이트웨이)를 가리키는 IPv6 `::/0` 경로.

   참고로 `::/0`이 인터넷 게이트웨이를 가리키면 외부 IPv6 호스트(VPC 외부)가 IPv6을 통한 연결을 시작할 수 있습니다.

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-cidr-block
0.0.0.0/0 --nat-gateway-id nat-05dba92075d71c408
```

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-ipv6-cidr-block
64:ff9b::/96 --nat-gateway-id nat-05dba92075d71c408
```

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-ipv6-cidr-block
::/0 --egress-only-internet-gateway-id eigw-c0a643a9
```

2. IPv6 전용 워크로드가 포함된 서브넷에서 DNS64 기능을 활성화합니다.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --enable-dns64
```

이제 프라이빗 서브넷의 리소스가 인터넷의 IPv4 및 IPv6 서비스와 상태 저장 연결을 설정할 수 있습니다. `64:ff9b::/96` 트래픽에 대한 송신 및 수신 트래픽을 허용하도록 보안 그룹 및 NACL을 적절히 구성합니다.

온프레미스 환경에서 IPv4 전용 서비스와의 통신 사용

Amazon Route 53 Resolver를 사용하면 VPC를 온프레미스 네트워크로 또는 그 반대로 DNS 쿼리를 전달할 수 있습니다. 이를 위해 다음 정책을 사용할 수 있습니다.

- VPC에 Route 53 Resolver 아웃바운드 엔드포인트를 생성하고 Route 53 Resolver가 쿼리를 전달할 IPv4 주소를 할당합니다. 온프레미스 DNS 해석기의 경우 이 주소는 DNS 쿼리가 시작되는 IP 주소이므로 IPv4 주소여야 합니다.
- Route 53 Resolver가 온프레미스 해석기에 전달할 DNS 쿼리의 도메인 이름을 지정하는 규칙을 하나 이상 만듭니다. 또한 온프레미스 해석기의 IPv4 주소를 지정합니다.
- Route 53 Resolver 아웃바운드 엔드포인트를 설정한 후, IPv6 전용 워크로드가 포함된 서브넷에서 DNS64를 활성화해야 합니다. 또한 온프레미스 네트워크로 향하는 모든 데이터 트래픽을 NAT 게이트웨이로 라우팅해야 합니다.

온프레미스 네트워크의 IPv4 전용 대상에서 DNS64 작동 방식:

1. VPC의 Route 53 Resolver 아웃바운드 엔드포인트에 IPv4 주소를 할당합니다.
2. IPv6 서비스의 DNS 쿼리는 IPv6을 통한 Route 53 Resolver로 이동합니다. Route 53 Resolver는 전달 규칙과 쿼리를 일치시키고 온프레미스 해석기의 IPv4 주소를 가져옵니다.

3. Route 53 Resolver는 쿼리 패킷을 IPv6에서 IPv4로 변환하여 아웃바운드 엔드포인트로 전달합니다. 엔드포인트의 각 IP 주소는 DNS 해석기의 온프레미스 IPv4 주소로 요청을 전달하는 하나의 ENI를 나타냅니다.
4. 온프레미스 해석기는 IPv4를 사용하여 아웃바운드 엔드포인트를 통해 Route 53 Resolver로 응답 패킷을 다시 보냅니다.
5. 쿼리가 DNS64 지원 서브넷에서 수행되었다고 가정하면 Route 53 Resolver는 다음 두 가지 작업을 수행합니다.

    a. 응답 패킷의 내용을 확인합니다. 레코드에 IPv6 주소가 포함되어 있으면 해당 콘텐츠를 그대로 유지하지만, IPv4 레코드만 포함된 경우에는 IPv4 주소에 대하여 `64:ff9b::/96`을 가장하여 IPv6 레코드도 합성합니다.

    b. 콘텐츠를 다시 패키징하여 IPv6을 통해 VPC 서비스로 전송합니다.

NAT 게이트웨이의 트래픽 검사

NAT 게이트웨이에 Network Firewall Proxy를 연결하여 NAT 게이트웨이의 트래픽을 검사하고 필터링할 수 있습니다. 이 보안 컨트롤을 사용하면 신뢰할 수 있는 경계 외부로의 데이터 유출을 방지하고 원치 않는 인바운드 응답을 차단할 수 있습니다.

작동 방식

Network Firewall Proxy를 생성할 때는 Proxy를 연결할 기존 NAT 게이트웨이를 선택해야 합니다. Proxy가 생성되면 다음과 같은 작업을 수행합니다.

- Proxy는 정규화된 도메인 이름과 함께 제공되며 애플리케이션이 http 및 https 연결 요청을 Proxy로 보내도록 설정해야 합니다. 프록시는 먼저 고객이 입력한 규칙에 따라 연결 요청의 도메인 이름을 필터링합니다. 고객이 허용하는 경우 프록시는 DNS 쿼리를 수행하여 도메인의 IP 주소를 가져옵니다. 그런 다음 최종 대상과의 TCP 연결을 설정합니다. TLS 복호화가 활성화되었는지 여부에 따라 프록시는 IP 주소 및 헤더 속성에 대해 TLS 연결을 필터링하고 정책에서 IP 및 헤더 속성(헤더 작업 및 URL 경로 포함)을 허용하는 경우에만 대상과 TLS 연결을 설정합니다.
- 어플라이언스는 트래픽을 검사하고 필터링합니다.
- 허용된 트래픽은 대상(인터넷, 온프레미스 환경 또는 다른 VPC)으로 계속됩니다.

어플라이언스 연결

어플라이언스는 AWS Network Firewall을 통해 NAT 게이트웨이에 연결됩니다. 어플라이언스를 생성 및 연결하는 단계는 Network Firewall Proxy 개발자 안내서를 참조하세요.

연결된 어플라이언스 보기

NAT 게이트웨이에 연결된 어플라이언스를 보려면 describe-nat-gateways 명령을 사용하세요.

```
aws ec2 describe-nat-gateways --nat-gateway-ids nat-1234567890abcdef0
```

응답에는 다음을 표시하는 `AttachedAppliances` 필드가 포함됩니다.

- Type - 어플라이언스 유형(예: `network-firewall-proxy`)
- ApplianceArn - 연결된 어플라이언스의 ARN
- AttachmentState - 현재 연결 상태(`attached`, `detaching`, `detached`, `attach_failed`, `detach_failed`)
- ModificationState - 현재 수정 상태(`modifying`, `completed`, `failed`)
- VpcEndpointId - 검사 및 필터링을 위해 애플리케이션 VPC에서 프록시로 트래픽을 라우팅하는 데 사용되는 VPC 엔드포인트 ID
- FailureCode - 어플라이언스 연결 또는 수정 작업이 실패한 경우의 실패 코드
- FailureMessage - 어플라이언스 연결 또는 수정 작업이 실패한 경우 실패를 설명하는 설명 메시지

Amazon CloudWatch를 사용하여 NAT 게이트웨이 모니터링

CloudWatch를 이용하여 NAT 게이트웨이를 모니터링하면 NAT 게이트웨이에 대한 정보를 수집하고, 거의 실시간에 가까운 읽기 가능한 지표를 만들 수 있습니다. 이 정보를 사용하여 NAT 게이트웨이를 모니터링하고 문제를 해결할 수 있습니다. 이러한 지표를 통해 NAT 게이트웨이의 상태와 성능을 파악하여 NAT 게이트웨이의 작업을 면밀히 모니터링하고 문제를 신속하게 해결할 수 있습니다.

CloudWatch에서 수집한 NAT 게이트웨이 지표에는 처리된 바이트, 패킷 수, 연결 수, 오류율과 같은 데이터 포인트가 포함됩니다. 이를 통해 NAT 게이트웨이를 통과하는 트래픽을 철저히 파악하고 이상 항목이나 병목 현상을 파악할 수 있습니다. CloudWatch는 1분 간격으로 이 지표 데이터를 제공하여 NAT 게이트웨이 동작에 대한 세부적인 최신 보기를 제공합니다.

또한 CloudWatch는 이 NAT 게이트웨이 지표 데이터를 15개월 동안 장기간 유지하므로 시간 경과에 따른 추세와 패턴을 분석할 수 있습니다. 이 기록 데이터를 사용하여 용량을 계획하고, 성능을 최적화하고, NAT 게이트웨이 사용의 장기적인 변화를 파악할 수 있습니다.

이러한 강력한 모니터링 기능을 활용하려면 필요에 맞게 맞춤형 CloudWatch 대시보드 및 경보를 생성하면 됩니다. 예를 들어 NAT 게이트웨이의 아웃바운드 데이터 전송이 특정 임곗값을 초과할 때마다 알리도록 설정하여 잠재적인 대역폭 제약을 사전에 해결할 수 있습니다.

요금에 대한 자세한 정보는 Amazon CloudWatch 요금을 참조하세요.

내용

- NAT 게이트웨이 지표 및 차원
- NAT 게이트웨이 CloudWatch 지표 보기
- NAT 게이트웨이를 모니터링하기 위한 CloudWatch 경보 생성

NAT 게이트웨이 지표 및 차원

NAT 게이트웨이에 사용할 수 있는 측정치는 아래와 같습니다. 설명 열에는 각 지표에 대한 설명과 단위 및 통계가 포함됩니다.

지표	설명
`ActiveConnectionCount`	NAT 게이트웨이를 통한 동시 활성 TCP 연결의 총 수입니다. 0의 값은 NAT 게이트웨이를 통한 활성 연결이 없음을 나타냅니다. 단위: 개수 통계: 가장 유용한 통계는 `Max`입니다.
`BytesInFromDestination`	NAT 게이트웨이가 대상으로부터 수신한 바이트 수입니다. `BytesOutToSource` 값이 `BytesInFromDestination` 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수도 있습니다. 단위: 바이트 통계: 가장 유용한 통계는 `Sum`입니다.
`BytesInFromSource`	NAT 게이트웨이가 VPC 내 클라이언트로부터 수신한 바이트 수입니다.

지표	설명
	BytesOutToDestination 값이 BytesInFromSource 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수도 있습니다. 단위: 바이트 통계: 가장 유용한 통계는 Sum입니다.
BytesOutToDestination	NAT 게이트웨이를 통해 대상으로 전송된 바이트 수입니다. 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 수도 있습니다. 단위: 바이트 통계: 가장 유용한 통계는 Sum입니다.
BytesOutToSource	NAT 게이트웨이를 통해 VPC 내 클라이언트로 전송된 바이트 수입니다. 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 수도 있습니다. 단위: 바이트 통계: 가장 유용한 통계는 Sum입니다.

지표	설명
ConnectionAttemptCount	NAT 게이트웨이를 통해 이루어진 연결 시도 횟수. 여기에는 최초 SYN만 포함됩니다. 경우에 따라 ConnectionAttemptCount 는 SYN 재전송으로 인해 ConnectionEstablishedCount 보다 낮을 수 있습니다. ConnectionEstablishedCount 값이 ConnectionAttemptCount 값보다 작은 경우, NAT 게이트웨이 뒤의 클라이언트가 응답이 없는 새 연결을 시도했음을 나타냅니다. 단위: 수 통계: 가장 유용한 통계는 Sum입니다.
ConnectionEstablishedCount	NAT 게이트웨이를 통해 설정된 연결 수. 여기에는 SYN 및 SYN 재전송이 포함됩니다. ConnectionEstablishedCount 값이 ConnectionAttemptCount 값보다 작은 경우, NAT 게이트웨이 뒤의 클라이언트가 응답이 없는 새 연결을 시도했음을 나타냅니다. 단위: 수 통계: 가장 유용한 통계는 Sum입니다.
ErrorPortAllocation	NAT 게이트웨이가 소스 포트 할당에 실패한 횟수. 0보다 큰 값은 너무 많은 동시 연결이 NAT 게이트웨이를 통해 열려 있음을 나타냅니다. 단위: 개수 통계: 가장 유용한 통계는 Sum입니다.

지표	설명
IdleTimeoutCount	활성 상태가 유휴 상태로 전환된 연결 수입니다. 활성 연결은 적절하게 종료되지 않고 직전 350초 동안 활동이 없는 경우 유휴 상태로 전환됩니다. 0보다 큰 값은 유휴 상태로 이동된 연결이 있었음을 나타냅니다. IdleTimeoutCount 값이 증가하는 경우, NAT 게이트웨이 뒤의 클라이언트가 부실 연결을 재사용하고 있음을 나타낼 수도 있습니다. 단위: 개 통계: 가장 유용한 통계는 Sum입니다.
PacketsDropCount	NAT 게이트웨이가 삭제한 패킷 수입니다. 삭제된 패킷 수를 전체 패킷 트래픽의 백분율로 계산하려면 PacketsDropCount/(PacketsInFromSource+PacketsInFromDestination)*100 공식을 사용합니다. 이 값이 NAT 게이트웨이의 총 트래픽 중 0.01퍼센트를 초과한다면 Amazon VPC 서비스에 문제가 있는 것일 수 있습니다. NAT 게이트웨이에서 패킷이 삭제되는 원인이 될 수 있는 서비스 관련 문제는 AWS 서비스 상태 대시보드를 사용하여 식별할 수 있습니다. 단위: 개 통계: 가장 유용한 통계는 Sum입니다.

지표	설명
PacketsInFromDestination	NAT 게이트웨이가 대상으로부터 수신한 패킷 수입니다. PacketsOutToSource 값이 PacketsInFromDestination 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수도 있습니다. 단위: 개 통계: 가장 유용한 통계는 Sum입니다.
PacketsInFromSource	NAT 게이트웨이가 VPC 내 클라이언트로부터 수신한 패킷 수입니다. PacketsOutToDestination 값이 PacketsInFromSource 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수도 있습니다. 단위: 개 통계: 가장 유용한 통계는 입니다Sum
PacketsOutToDestination	NAT 게이트웨이를 통해 대상으로 전송된 패킷 수입니다. 0보다 큰 값은 NAT 게이트웨이 뒤에 있는 클라이언트에서 인터넷으로 가는 트래픽이 있음을 나타냅니다. PacketsOutToDestination 값이 PacketsInFromSource 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있을 수도 있습니다. 단위: 개 통계: 가장 유용한 통계는 입니다Sum

지표	설명
PacketsOutToSource	NAT 게이트웨이를 통해 VPC 내 클라이언트로 전송된 패킷 수입니다. 0보다 큰 값은 인터넷에서 NAT 게이트웨이 뒤에 있는 클라이언트로 오는 트래픽이 있음을 나타냅니다. PacketsOutToSource 값이 PacketsInFromDestination 값보다 작은 경우, NAT 게이트웨이 처리 중에 데이터 손실이 있거나 NAT 게이트웨이가 적극적으로 차단하는 트래픽이 있을 수도 있습니다. 단위: 개 통계: 가장 유용한 통계는 입니다Sum
PeakBytesPerSecond	이 지표는 특정 분당 가장 높은 초당 10초 바이트 평균값을 보고합니다. 단위: 개 통계: 가장 유용한 통계는 입니다Maximum
PeakPacketsPerSecond	이 지표는 60초 동안 10초마다 평균 패킷 속도(초당 처리된 패킷 수)를 계산한 다음 최대 6개 속도(가장 높은 평균 패킷 속도)를 보고합니다. 단위: 개 통계: 가장 유용한 통계는 입니다Maximum

지표 데이터를 필터링하려면 다음 차원을 사용하세요.

차원	설명
NatGatewayId	NAT 게이트웨이 ID를 기준으로 측정치 데이터를 필터링합니다. 영역 NAT 게이트웨이는 이 차원만 사용합니다. 리전 NAT 게이트웨이는 이 차

차원	설명
	원과 AvailabilityZone 을 함께 사용합니다.
AvailabilityZone	가용 영역을 기준으로 지표 데이터를 필터링합니다. 리전 NAT 게이트웨이는 이 차원과 NatGatewayId 을 함께 사용합니다. 영역 NAT 게이트웨이는 이 차원을 사용하지 않습니다.

NAT 게이트웨이 CloudWatch 지표 보기

NAT 게이트웨이 지표는 1분 가격으로 CloudWatch로 전송됩니다. 지표는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 가능한 차원 조합에 따라 그룹화됩니다. NAT 게이트웨이에 대해 다음과 같이 측정치를 볼 수 있습니다.

CloudWatch 콘솔을 사용하여 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표, 모든 지표를 선택합니다.
3. NATGateway 지표 네임스페이스를 선택합니다.
4. 지표 차원을 선택합니다.

를 사용하여 지표를 보려면AWS CLI

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

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

NAT 게이트웨이를 모니터링하기 위한 CloudWatch 경보 생성

경보로 인해 상태가 변경되면 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 알림을 보냅니다.

예를 들어 NAT 게이트웨이로 들어오거나 나가는 트래픽의 양을 모니터링하는 경보를 만들 수 있습니다. 아래 경보는 NAT 게이트웨이를 통해 VPC의 클라이언트에서 인터넷으로 가는 아웃바운드 트래픽

의 양을 모니터링합니다. 그리고 15분 동안 바이트 수가 임계값인 5,000,000에 도달하면 알림을 보냅니다.

NAT 게이트웨이를 통한 아웃바운드 트래픽에 대한 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보(Alarms) 모든 경보(All Alarms)를 선택합니다.
3. 경보 생성(Create alarm)을 선택하세요.
4. 지표 선택(Select metric)을 선택하세요.
5. NATGateway 지표 네임스페이스를 선택한 다음 지표 차원을 선택합니다. 지표를 가져오면 NAT 게이트웨이에 대한 BytesOutToDestination 지표 옆 확인란을 선택한 다음 지표 선택을 선택합니다.
6. 경보를 다음과 같이 구성한 다음 다음(Next)을 선택합니다.
   - 통계(Statistic)에서 합계(Sum)를 선택합니다.
   - 기간에서 15분을 선택합니다.
   - 항상에서 초과/같음을 선택하고 임계값으로 `5000000`을 입력합니다.
7. 알림에서 기존 SNS 주제를 선택하거나 새 주제 생성을 선택하여 새로 생성합니다. Next(다음)를 선택합니다.
8. 경보의 이름과 설명을 입력하고 다음을 선택합니다.
9. 경보 구성을 마쳤으면 경보 생성을 선택합니다.

다른 예와 같이, 포트 할당 오류를 모니터링하는 경보를 만들고 이 값이 3회 연속 5분간 0보다 클 경우에 알림을 보낼 수 있습니다.

경보를 만들어 포트 할당 오류를 모니터링하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보(Alarms) 모든 경보(All Alarms)를 선택합니다.
3. 경보 생성(Create alarm)을 선택하세요.
4. 지표 선택(Select metric)을 선택하세요.
5. NATGateway 지표 네임스페이스를 선택한 다음 지표 차원을 선택합니다. 지표를 가져오면 NAT 게이트웨이에 대한 ErrorPortAllocation 지표 옆 확인란을 선택한 다음 지표 선택을 선택합니다.
6. 경보를 다음과 같이 구성한 다음 다음(Next)을 선택합니다.

- 통계에서 최대를 선택합니다.
- 기간에서 5분을 선택합니다.
- 항상에서 초과를 선택하고 임계값으로 0을 입력합니다.
- 추가 구성에서 경보를 생성할 데이터 포인트에 대해 3을 입력합니다.

7. 알림에서 기존 SNS 주제를 선택하거나 새 주제 생성을 선택하여 새로 생성합니다. Next(다음)를 선택합니다.
8. 경보의 이름과 설명을 입력하고 다음을 선택합니다.
9. 경보 구성이 완료되면 경보 생성을 선택합니다.

자세한 정보는 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 사용을 참조하세요.

NAT 게이트웨이 문제 해결

다음 주제는 NAT 게이트웨이를 만들거나 사용할 때 발생할 수 있는 일반적인 문제를 해결하는 데 도움이 됩니다.

문제

- NAT 게이트웨이 생성 실패
- NAT 게이트웨이 할당량
- 탄력적 IP 주소 할당량
- 가용 영역이 지원되지 않음
- NAT 게이트웨이가 더 이상 표시되지 않음
- NAT 게이트웨이가 Ping 명령에 응답하지 않음
- 인스턴스에서 인터넷에 액세스할 수 없음
- 대상에 대한 TCP 연결 실패
- 경로 추적 출력에 NAT 게이트웨이 프라이빗 IP 주소가 표시되지 않음
- 350초 후 인터넷 연결이 끊어짐
- IPSec 연결을 설정할 수 없음
- 추가 연결을 시작할 수 없음

NAT 게이트웨이 생성 실패

문제

NAT 게이트웨이를 생성하면 상태가 `Failed`가 됩니다.

> **Note**
>
> 실패한 NAT 게이트웨이는 일반적으로 약 1시간 내에 자동으로 삭제됩니다.

원인

NAT 게이트웨이를 생성할 때 오류가 발생했습니다. 반환된 상태 메시지를 통해 오류 원인을 확인할 수 있습니다.

Solution

오류 메시지를 보려면 Amazon VPC 콘솔열 연 후 NAT 게이트웨이를 선택하세요. NAT 게이트웨이에 대한 라디오 버튼을 선택한 후, 세부 정보 탭에서 상태 메시지를 찾습니다.

다음 표에는 Amazon VPC 콘솔에 표시되는 오류의 예상 원인이 나와 있습니다. 표시된 수정 단계를 적용한 후 NAT 게이트웨이를 다시 만들어 볼 수 있습니다.

표시된 오류	원인	솔루션
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 서브넷에서 사용 가능한 IP 주소 개수를 확인합니다. 서브넷의 세부 정보 창에서 사용 가능한 IP를 볼 수 있습니다. 서브넷에서 사용 가능한 IP 주소를 만들려면 사용되지 않은 네트워크 인터페이스를 삭제하거나 필요 없는 인스턴스를 종료할 수 있습니다.
네트워크 vpc-xxxxxxxx에 연결된 인터넷 게이트웨이가 없음	인터넷 게이트웨이가 있는 VPC에서 NAT 게이트웨이를 만들어야 합니다.	인터넷 게이트웨이를 생성하여 VPC에 연결합니다. 자세한 내용은 서브넷에 인터넷 액세스 추가 섹션을 참조하세요.

표시된 오류	원인	솔루션
탄력적 IP 주소 eipalloc-xxxxxxxx가 이미 연결되어 있음	지정한 탄력적 IP 주소가 다른 리소스와 이미 연결되어 있으므로 NAT 게이트웨이와 연결할 수 없습니다.	탄력적 IP 주소와 연결된 리소스를 확인합니다. Amazon VPC 콘솔에서 탄력적 IP 페이지로 이동하여 인스턴스 ID 또는 네트워크 인터페이스 ID에 지정된 값을 확인합니다. 해당 리소스에 탄력적 IP 주소가 필요 없는 경우 주소를 연결 해제할 수 있습니다. 또는 계정에 새로운 탄력적 IP 주소를 할당합니다. 자세한 내용은 탄력적 IP 주소 사용 시작 섹션을 참조하세요.

NAT 게이트웨이 할당량

NAT 게이트웨이를 생성하려고 하면 다음 오류가 발생합니다.

```
Performing this operation would exceed the limit of 5 NAT gateways
```

원인

해당 가용 영역에 대한 NAT 게이트웨이 수가 할당량에 도달했습니다.

Solution

계정에 대한 이 NAT 게이트웨이 할당량에 도달한 경우, 다음 중 하나를 수행할 수 있습니다.

- Service Quotas 콘솔을 사용하여 가용 영역 할당량당 NAT 게이트웨이 증가를 요청합니다.
- NAT 게이트웨이의 상태를 확인합니다. `Pending`, `Available` 또는 `Deleting`의 상태는 할당량에 포함됩니다. NAT 게이트웨이를 최근에 삭제한 경우 상태가 `Deleting`에서 `Deleted`로 바뀔 때까지 몇 분간 기다립니다. 그런 다음 새 NAT 게이트웨이를 생성해 보세요.
- 특정 가용 영역에 NAT 게이트웨이가 필요 없는 경우 할당량에 도달하지 않은 가용 영역에서 NAT 게이트웨이를 만들어 봅니다.

자세한 내용은 Amazon VPC 할당량 단원을 참조하세요.

탄력적 IP 주소 할당량

문제

퍼블릭 NAT 게이트웨이에 탄력적 IP 주소를 할당하려고 하면 다음 오류가 발생합니다.

```
The maximum number of addresses has been reached.
```

원인

해당 리전의 계정에 대한 탄력적 IP 주소 수가 할당량에 도달했습니다.

Solution

탄력적 IP 주소 할당량에 도달한 경우 다른 리소스에서 탄력적 IP 주소의 연결을 해제할 수 있습니다. 또는 Service Quotas 콘솔을 사용하여 탄력적 IP 할당량 증가를 요청할 수 있습니다.

가용 영역이 지원되지 않음

문제

NAT 게이트웨이를 생성하려고 하면 `NotAvailableInZone` 오류가 발생합니다.

원인

제약이 있는 가용 영역(확장이 제약되어 있는 영역)에서 NAT 게이트웨이를 생성하려 했을지도 모릅니다.

Solution

이러한 가용 영역에서는 NAT 게이트웨이를 지원하지 않습니다. 다른 가용 영역에서 NAT 게이트웨이를 만들고 제약이 있는 영역의 프라이빗 서브넷에 사용할 수 있습니다. 리소스와 NAT 게이트웨이가 동일한 영역에 있도록 제약이 없는 가용 영역으로 리소스를 이동할 수도 있습니다.

NAT 게이트웨이가 더 이상 표시되지 않음

문제

NAT 게이트웨이를 만들었지만 Amazon VPC 콘솔에 표시되지 않습니다.

원인

NAT 게이트웨이를 만드는 동안 오류가 발생하여 생성에 실패했을 수 있습니다. `Failed` 상태의 NAT 게이트웨이는 Amazon VPC 콘솔에 1시간 가량 표시됩니다. 한 시간 후에는 자동으로 삭제됩니다.

Solution

NAT 게이트웨이 생성 실패에서 정보를 검토하고 새 NAT 게이트웨이를 만들어 봅니다.

NAT 게이트웨이가 Ping 명령에 응답하지 않음

문제

인터넷에서(예를 들어 홈 컴퓨터에서) 또는 VPC의 인스턴스에서 NAT 게이트웨이의 탄력적 IP 주소 또는 프라이빗 IP 주소를 ping하려고 시도하는 경우 응답을 얻을 수 없습니다.

원인

NAT 게이트웨이는 프라이빗 서브넷의 인스턴스에서 인터넷으로만 트래픽을 전달합니다.

Solution

NAT 게이트웨이가 작동하는지 테스트하려면 퍼블릭 NAT 게이트웨이 테스트를 참조하세요.

인스턴스에서 인터넷에 액세스할 수 없음

문제

퍼블릭 NAT 게이트웨이를 생성하고 테스트 단계를 수행했지만 `ping` 명령이 실패하거나 프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 없습니다.

원인

이 문제의 원인은 다음 중 하나일 수 있습니다.

- NAT 게이트웨이가 트래픽을 제공할 준비가 되지 않았습니다.
- 라우팅 테이블이 올바르게 구성되지 않았습니다.
- 보안 그룹 또는 네트워크 ACL이 인바운드 또는 아웃바운드 트래픽을 차단하고 있습니다.
- 지원되지 않는 프로토콜을 사용하고 있습니다.

Solution

다음 정보를 확인하세요.

- NAT 게이트웨이가 `Available` 상태인지 확인합니다. Amazon VPC 콘솔에서 NAT 게이트웨이 페이지로 이동하고 세부 정보 창에서 상태 정보를 봅니다. NAT 게이트웨이가 실패 상태인 경우 게이트웨이가 생성될 때 오류가 발생했을 수 있습니다. 자세한 내용은 NAT 게이트웨이 생성 실패 단원을 참조하세요.
- 라우팅 테이블을 올바로 구성했는지 확인합니다:
  - NAT 게이트웨이는 인터넷 트래픽을 인터넷 게이트웨이로 라우팅하는 라우팅 테이블이 있는 퍼블릭 서브넷에 있어야 합니다.
  - 인스턴스는 인터넷 트래픽을 NAT 게이트웨이로 라우팅하는 라우팅 테이블이 있는 프라이빗 서브넷에 있어야 합니다.
  - 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.

  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.

  흐름 로그를 활성화하여 네트워크 ACL 또는 보안 그룹 규칙으로 인해 끊어진 연결을 진단할 수 있습니다. 자세한 내용은 VPC 흐름 로그를 사용하여 IP 트래픽 로깅 단원을 참조하세요.
- `ping` 명령을 사용하는 경우 ICMP가 활성화된 호스트를 ping하고 있는지 확인합니다. ICMP가 활성화되지 않은 경우 회신 패킷을 받지 못합니다. 이를 테스트하려면 사용자 자신의 컴퓨터의 명령줄 터미널에서 똑같은 `ping` 명령을 수행하세요.
- 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.

대상에 대한 TCP 연결 실패

문제

프라이빗 서브넷의 인스턴스에서 NAT 게이트웨이를 통해 특정 대상에 연결할 때 일부 TCP 연결은 성공하지만 일부는 실패하거나 시간이 초과됩니다.

원인

이 문제의 원인은 다음 중 하나일 수 있습니다.

- 대상 엔드포인트가 조각난 TCP 패킷으로 응답하고 있습니다. NAT 게이트웨이는 TCP 또는 ICMP에 대한 IP 조각화를 지원하지 않습니다. 자세한 내용은 NAT 게이트웨이 및 NAT 인스턴스 비교 단원을 참조하세요.
- `tcp_tw_recycle` 옵션이 원격 서버에서 활성화되었으며, 이 옵션은 NAT 디바이스 뒤에 여러 연결이 있는 경우 문제를 일으키는 것으로 알려져 있습니다.

Solutions

다음을 수행하여 연결하려는 엔드포인트가 조각난 TCP 패킷으로 응답하는지 확인하세요.

1. 퍼블릭 IP 주소가 있는 퍼블릭 서브넷의 인스턴스를 사용하여 특정 엔드포인트로부터 조각화를 유발할 정도로 큰 응답을 트리거합니다.
2. `tcpdump` 유틸리티를 사용하여 엔드포인트가 조각화된 패킷을 전송하는지 확인합니다.

   > **Important**
   >
   > 이러한 확인을 수행하려면 퍼블릭 서브넷의 인스턴스를 사용해야 합니다. 원래 연결이 실패한 인스턴스, NAT 게이트웨이 뒤 프라이빗 서브넷의 인스턴스 또는 NAT 인스턴스는 사용할 수 없습니다.

   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.
3. 엔드포인트가 조각화된 TCP 패킷을 전송하는 경우 NAT 게이트웨이 대신 NAT 인스턴스를 사용할 수 있습니다.

원격 서버에 액세스할 수 있는 경우 다음을 수행하여 `tcp_tw_recycle` 옵션이 사용 가능한지 확인할 수 있습니다.

1. 서버에서 다음 명령을 실행합니다.

```
cat /proc/sys/net/ipv4/tcp_tw_recycle
```

1이 출력될 경우 tcp_tw_recycle 옵션이 활성화된 것입니다.

2. tcp_tw_recycle이 활성화된 경우 비활성화하는 것이 좋습니다. 연결을 재사용해야 하는 경우 tcp_tw_reuse 옵션을 사용하는 것이 더 안전합니다.

원격 서버에 액세스할 수 없는 경우 프라이빗 서브넷의 인스턴스에서 tcp_timestamps 옵션을 일시적으로 비활성화하여 테스트할 수 있습니다. 그런 다음 원격 서버에 다시 연결합니다. 연결에 성공하면 원격 서버에서 tcp_tw_recycle이 활성화된 것이 이전 오류의 원인일 수 있습니다. 가능하면 원격 서버 소유자에게 이 옵션이 활성화되어 있는지 확인하고 비활성화하도록 요청하세요.

경로 추적 출력에 NAT 게이트웨이 프라이빗 IP 주소가 표시되지 않음

문제

인스턴스가 인터넷에 액세스할 수 있지만, traceroute 명령을 수행할 때 출력에 NAT 게이트웨이의 프라이빗 IP 주소가 표시되지 않습니다.

원인

인스턴스가 인터넷 게이트웨이 등의 다른 게이트웨이를 사용하여 인터넷에 액세스하고 있습니다.

Solution

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

문제

인스턴스에서 인터넷에 액세스할 수 있지만 350초 후에 연결이 끊어집니다.

원인

NAT 게이트웨이를 사용하는 연결이 350초 이상 유휴 상태인 경우 연결이 시간 초과됩니다.

연결 제한 시간이 초과하면 NAT 게이트웨이는 연결을 계속하려고 하는 NAT 게이트웨이 뒤의 리소스로 RST 패킷을 반환합니다(FIN 패킷을 보내지 않음).

Solution

연결이 끊어지지 않도록 하려면 연결을 통해 더 많은 트래픽을 시작합니다. 또는 인스턴스에서 350초 미만의 값으로 TCP keepalive를 활성화할 수 있습니다.

IPSec 연결을 설정할 수 없음

문제

대상에 대한 IPsec 연결을 설정할 수 없습니다.

원인

NAT 게이트웨이는 현재 IPSec 프로토콜을 지원하지 않습니다.

Solution

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 확인하세요.

추가 연결을 시작할 수 없음

문제

대상에 대해 NAT 게이트웨이를 통한 기존 연결이 있지만 추가 연결을 설정할 수 없습니다.

원인

단일 NAT 게이트웨이에 대한 동시 연결 제한에 도달했을 수 있습니다. 자세한 내용은 NAT 게이트웨이 기본 사항 단원을 참조하세요. 프라이빗 서브넷의 인스턴스가 많은 수의 연결을 생성하는 경우 이 제한에 도달할 수 있습니다.

Solution

다음 중 하나를 수행하세요.

- 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.
- CloudWatch의 `IdleTimeoutCount` 지표를 사용하여 유휴 접속의 증가를 모니터링합니다. 유휴 상태의 연결을 닫아서 용량을 확보합니다.
- 다양한 IP 주소를 이용하여 NAT 게이트웨이를 생성하거나 보조 IP 주소를 기존의 NAT 게이트웨이에 추가합니다. 각각의 신규 IPv4 주소에서는 최대 55,000개의 동시 연결이 지원됩니다. 자세한 내용은 NAT 게이트웨이 만들기 또는 보조 IP 주소 연결 편집 섹션을 참조하세요.

NAT 게이트웨이 요금

NAT 게이트웨이를 프로비저닝하면 NAT 게이트웨이를 사용할 수 있는 시간당 요금 및 처리하는 데이터 기가바이트당 요금이 부과됩니다 자세한 내용은 Amazon VPC 요금을 참조하세요.

다음 전략은 NAT 게이트웨이에 대한 데이터 전송 요금을 줄이는 데 도움이 될 수 있습니다.

- AWS 리소스가 가용 영역에서 상당한 양의 트래픽을 전송하거나 수신하는 경우 리소스가 NAT 게이트웨이와 동일한 가용 영역에 있는지 확인합니다. 또는 리소스가 있는 각 가용 영역에 NAT 게이트웨이를 생성합니다.
- NAT 게이트웨이를 통과하는 대부분의 트래픽이 인터페이스 엔드포인트 또는 게이트웨이 엔드포인트를 지원하는 AWS 서비스를 사용하는 경우 이러한 서비스에 대한 인터페이스 엔드포인트 또는 게이트웨이 엔드포인트를 만드는 것이 좋습니다. 잠재적인 비용 절감에 관한 자세한 내용은 AWS PrivateLink 요금을 참조하세요.

NAT 인스턴스

NAT 인스턴스는 Network Address Translation(NAT)을 제공합니다. NAT 인스턴스를 사용하면 프라이빗 서브넷의 리소스가 인터넷이나 온프레미스 네트워크와 같은 Virtual Private Cloud(VPC) 외부의 대상과 통신할 수 있습니다. 프라이빗 서브넷의 리소스는 인터넷으로 향하는 아웃바운드 IPv4 트래픽을 시작할 수 있지만 인터넷에서 시작된 인바운드 트래픽을 수신할 수는 없습니다.

> **Important**
>
> NAT AMI는 2020년 12월 31일에 표준 지원이 종료되고 2023년 12월 31일에 유지 관리 지원이 종료된 Amazon Linux AMI, 2018.03의 마지막 버전을 기반으로 구축되었습니다. 자세한 내용은 Amazon Linux AMI 지원 종료 블로그 게시물을 참조하세요.
> 기존 NAT AMI 사용하는 경우 AWS는 NAT 게이트웨이로의 마이그레이션을 권장합니다. NAT 게이트웨이는 더 나은 가용성과 향상된 대역폭을 제공하면서 관리 작업은 간소화합니다. 자세한 내용은 NAT 게이트웨이 및 NAT 인스턴스 비교 섹션을 참조하세요.

NAT 게이트웨이보다 NAT 인스턴스가 사용 사례와 더 잘 일치하는 경우 the section called “3. NAT AMI 생성”에 설명된 대로 Amazon Linux 현재 버전에서 자체 NAT AMI를 생성할 수 있습니다.

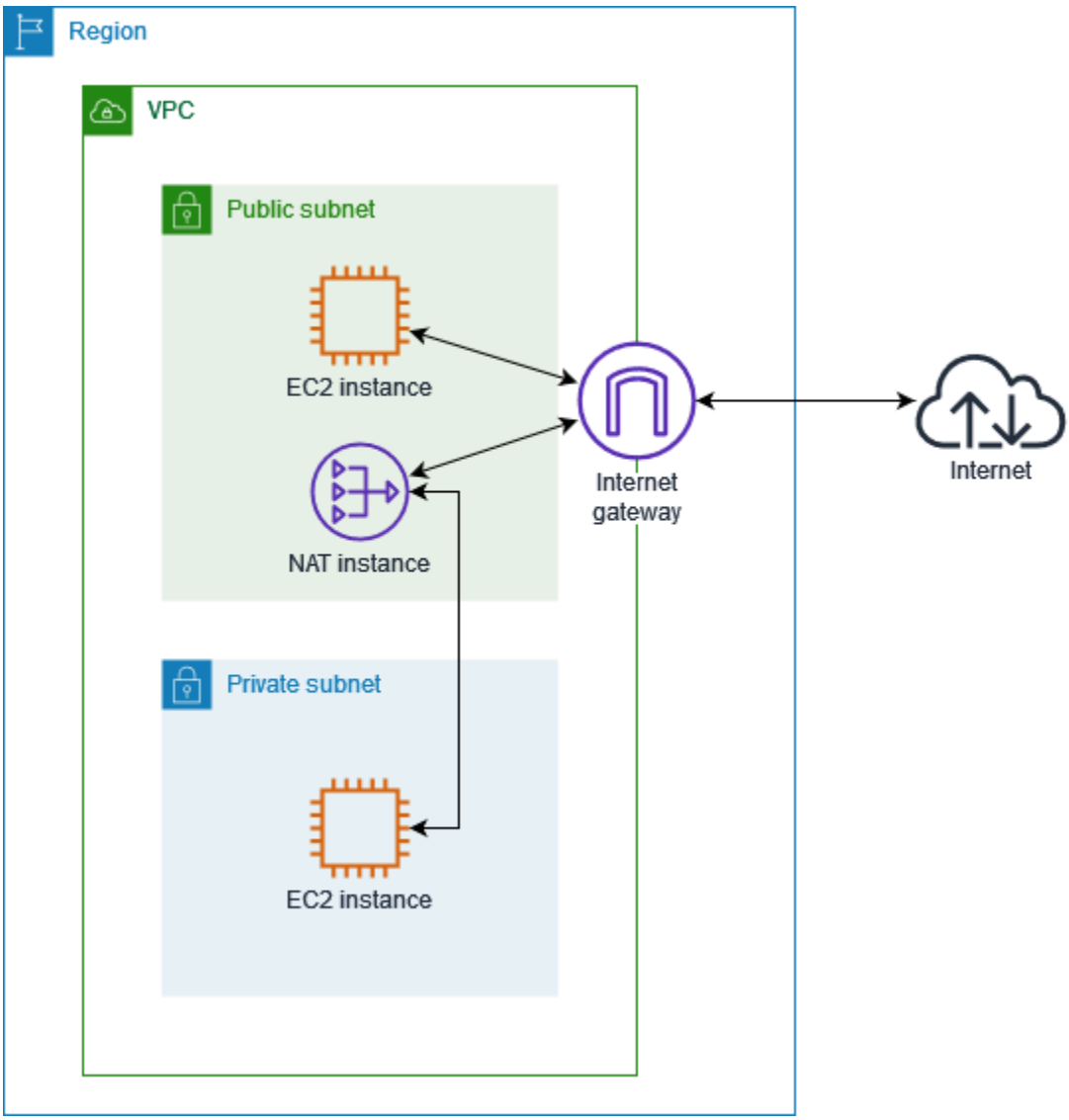
내용

- NAT 인스턴스 기본 사항
- 프라이빗 리소스가 VPC 외부에서 통신할 수 있도록 지원

NAT 인스턴스 기본 사항

다음 그림에서는 NAT 인스턴스 기본 사항을 보여줍니다. 프라이빗 서브넷과 연결된 라우팅 테이블은 프라이빗 서브넷의 인스턴스에서 퍼블릭 서브넷의 NAT 인스턴스로 인터넷 트래픽을 전송합니다. 그러면 NAT 인스턴스는 인터넷 게이트웨이로 트래픽을 전송합니다. 트래픽은 NAT 인스턴스의 퍼블릭 IP 주소로 귀속됩니다. 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.

NAT 인스턴스는 인터넷에 액세스할 수 있어야 하므로, 퍼블릭 서브넷(인터넷 게이트웨이로 가는 경로가 있는 라우팅 테이블이 있는 서브넷)에 있어야 하며, NAT 인스턴스에는 퍼블릭 IP 주소 또는 탄력적 IP 주소가 있어야 합니다.



NAT 인스턴스를 시작하려면 NAT AMI를 생성하고 NAT 인스턴스용 보안 그룹을 생성한 다음 VPC로 NAT 인스턴스를 시작합니다.

NAT 인스턴스 할당량은 리전의 인스턴스 할당량에 따라 다릅니다. 자세한 내용은 AWS 일반 참조의 Amazon EC2 서비스 할당량을 참조하세요.

프라이빗 리소스가 VPC 외부에서 통신할 수 있도록 지원

이 섹션에서는 프라이빗 서브넷의 리소스가 가상 프라이빗 외부에서 통신할 수 있도록 NAT 인스턴스를 생성하고 사용하는 방법을 설명합니다.

작업

- 1. NAT 인스턴스의 VPC 생성

- 2. NAT 인스턴스에 대한 보안 그룹 생성
- 3. NAT AMI 생성
- 4. NAT 인스턴스 시작
- 5. 원본/대상 확인 비활성화
- 6. 라우팅 테이블 업데이트
- 7. NAT 인스턴스 테스트

1. NAT 인스턴스의 VPC 생성

다음 절차를 따라 퍼블릭 서브넷 및 프라이빗 서브넷이 있는 VPC를 생성합니다.

VPC를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. VPC 생성을 선택합니다.
3. Resources to create(생성할 리소스)에서 VPC and more(VPC 등)를 선택합니다.
4. Name tag auto-generation(이름 태그 자동 생성)에 VPC의 이름을 입력합니다.
5. 서브넷을 구성하려면 다음을 수행합니다.
    a. Number of Availability Zones(가용 영역 수)에서 필요에 따라 1 또는 2를 선택합니다.
    b. Number of public subnets(퍼블릭 서브넷 수)에서 가용 영역당 하나의 퍼블릭 서브넷이 있는지 확인합니다.
    c. Number of private subnets(프라이빗 서브넷 수)에서 가용 영역당 하나의 프라이빗 서브넷이 있는지 확인합니다.
6. VPC 생성을 선택합니다.

2. NAT 인스턴스에 대한 보안 그룹 생성

다음 표에 설명된 규칙을 사용하여 보안 그룹을 생성합니다. 이 규칙을 사용하면 NAT 인스턴스가 프라이빗 서브넷에 있는 인스턴스로부터 오는 인터넷 트래픽뿐 아니라, 네트워크에서 오는 SSH 트래픽도 수신할 수 있습니다. 또한 NAT 인스턴스는 인터넷으로 트래픽을 전송할 수 있으며 따라서 프라이빗 서브넷의 인스턴스가 소프트웨어 업데이트를 받을 수 있습니다.

다음은 인바운드 권장 규칙입니다.

소스	프로토콜	포트 범위	설명
#### ### CIDR	TCP	80	프라이빗 서브넷의 서버로부터의 인바운드 HTTP 트래픽 허용
#### ### CIDR	TCP	443	프라이빗 서브넷의 서버로부터의 인바운드 HTTPS 트래픽 허용
##### ### IP ## ##	TCP	22	네트워크로부터 NAT 인스턴스에 대한 인바운드 SSH 액세스 허용 (인터넷 게이트웨이를 통해)

다음은 권장 아웃바운드 규칙입니다.

대상	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	인터넷에 대한 아웃바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	인터넷에 대한 아웃바운드 HTTPS 액세스 허용

보안 그룹을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹 생성을 선택합니다.
4. 보안 그룹의 이름과 설명을 입력합니다.
5. VPC인 경우 NAT 인스턴스에 대한 VPC ID를 선택합니다.
6. 다음과 같이 인바운드 규칙(Inbound Rules) 하에 인바운드 트래픽에 대한 규칙을 추가합니다.

    a. 규칙 추가를 선택합니다. 유형으로 HTTP를 선택하고 소스(Source)에 대한 프라이빗 서브넷의 IP 주소 범위를 입력합니다.

    b. 규칙 추가를 선택합니다. 유형으로 HTTPS를 선택하고 소스(Source)에 대한 프라이빗 서브넷의 IP 주소 범위를 입력합니다.

c. 규칙 추가를 선택합니다. 유형으로 SSH를 선택하고 소스(Source)에 대한 네트워크의 IP 주소 범위를 입력합니다.

7. 다음과 같이 아웃바운드 규칙(Outbound Rules) 하에 아웃바운드 트래픽에 대한 규칙을 추가합니다.

   a. 규칙 추가를 선택합니다. 유형으로 HTTP를 선택하고 대상에 0.0.0.0/0을 입력합니다.

   b. 규칙 추가를 선택합니다. 유형으로 HTTPS를 선택하고 대상에 0.0.0.0/0을 입력합니다.

8. 보안 그룹 생성을 선택합니다.

자세한 내용은 보안 그룹 섹션을 참조하세요.

3. NAT AMI 생성

EC2 인스턴스에서 NAT를 실행하기 위해 NAT AMI가 구성됩니다. NAT AMI를 생성한 다음 NAT AMI를 사용하여 NAT 인스턴스를 시작해야 합니다.

NAT AMI에 Amazon Linux 이외의 운영 체제를 사용하려는 경우 해당 운영 체제의 설명서를 참조하여 NAT 구성 방법을 확인하십시오. 인스턴스를 재부팅한 후에도 이 설정이 유지되도록 설정을 저장해야 합니다.

Amazon Linux용 NAT AMI 생성

1. AL2023 또는 Amazon Linux 2를 실행하는 EC2 인스턴스를 시작합니다. NAT 인스턴스용으로 생성한 보안 그룹을 지정해야 합니다.
2. 인스턴스에 연결하고 인스턴스에서 다음 명령을 실행하여 iptables를 활성화합니다.

```
sudo yum install iptables-services -y
sudo systemctl enable iptables
sudo systemctl start iptables
```

3. 인스턴스에서 다음 작업을 수행하여 IP 전달을 활성화하고 IP 전달이 재부팅 후에도 계속 유지되도록 합니다.

   a. nano 또는 vim 등의 텍스트 편집기를 사용하여 구성 파일(`/etc/sysctl.d/custom-ip-forwarding.conf`)을 생성합니다.

   b. 구성 파일에 다음 줄을 추가합니다.

```
net.ipv4.ip_forward=1
```

c. 구성 파일을 저장하고 텍스트 편집기를 종료합니다.

d. 다음 명령을 실행하여 구성 파일을 적용합니다.

```
sudo sysctl -p /etc/sysctl.d/custom-ip-forwarding.conf
```

4. 인스턴스에서 다음 명령을 실행하고, 기본 네트워크 인터페이스의 이름을 기록해 둡니다. 다음 단계에서 이 정보가 필요합니다.

```
netstat -i
```

다음 예제 출력에서 `docker0`은 도커에 의해 생성된 네트워크 인터페이스이고, `eth0`은 기본 네트워크 인터페이스이고, `lo`는 루프백 인터페이스입니다.

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0   1500        0      0      0 0             0      0      0      0 BMU
eth0      9001  7276052      0      0 0       5364991      0      0      0 BMRU
lo       65536   538857      0      0 0        538857      0      0      0 LRU
```

다음 예제 출력에서 기본 네트워크 인터페이스는 `enX0`입니다.

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enX0      9001     1076      0      0 0          1247      0      0      0 BMRU
lo       65536       24      0      0 0            24      0      0      0 LRU
```

다음 예제 출력에서 기본 네트워크 인터페이스는 `ens5`입니다.

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens5      9001    14036      0      0 0          2116      0      0      0 BMRU
lo       65536       12      0      0 0            12      0      0      0 LRU
```

5. 인스턴스에서 다음 명령을 실행하여 NAT를 구성합니다. 기본 네트워크 인터페이스가 `eth0`이 아닌 경우 *`eth0`*을 이전 단계에서 기록한 기본 네트워크 인터페이스로 바꿉니다.

```
sudo /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo /sbin/iptables -F FORWARD
sudo service iptables save
```

6. EC2 인스턴스에서 NAT AMI를 생성합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스에서 Linux AMI 생성을 참조하세요.

4. NAT 인스턴스 시작

다음 절차에 따라 생성한 VPC, 보안 그룹, NAT AMI를 사용하여 NAT 인스턴스를 시작합니다.

NAT 인스턴스를 시작하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 대시보드에서 인스턴스 시작을 선택합니다.
3. 이름(Name)에 NAT 인스턴스의 이름을 입력합니다.
4. 애플리케이션 및 OS 이미지에서 NAT AMI를 선택합니다(더 많은 AMI 찾아보기, 내 AMI 선택).
5. 인스턴스 유형에서 NAT 인스턴스에 필요한 컴퓨팅, 메모리, 스토리지 리소스를 제공하는 인스턴스 유형을 선택합니다.
6. 키 페어에서 기존 키 페어를 선택하거나 새 키 페어 생성을 선택합니다.
7. Network settings(네트워크 설정)에서 다음을 수행합니다.
   a. 편집을 선택합니다.
   b. [VPC]에 대해 생성된 VPC를 선택합니다.
   c. 서브넷에서 생성한 퍼블릭 서브넷을 선택합니다.
   d. Auto-assign Public IP(퍼블릭 IP 자동 할당)에서 Enable(활성화)을 선택합니다. 또는 NAT 인스턴스를 시작한 후 탄력적 IP 주소를 할당하고 이를 NAT 인스턴스에 할당합니다.
   e. 방화벽에서 기존 보안 그룹 선택을 선택한 다음 생성한 보안 그룹을 선택합니다.
8. 인스턴스 시작을 선택합니다. 인스턴스 ID를 선택하여 인스턴스 세부 정보 페이지를 엽니다. 인스턴스 상태가 실행 중으로 변경되고 상태 확인이 성공할 때까지 기다립니다.
9. NAT 인스턴스의 소스/대상 확인을 비활성화합니다(5. 원본/대상 확인 비활성화 참조).
10. NAT 인스턴스로 트래픽을 보내기 위한 라우팅 테이블을 업데이트합니다(6. 라우팅 테이블 업데이트 참조).

5. 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.

소스/대상 확인을 비활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. NAT 인스턴스를 선택합니다.
4. 작업, 네트워킹, 소스/대상 확인 변경을 선택합니다.
5. 소스/대상 확인에서 중지를 선택합니다.
6. Save를 선택합니다.
7. NAT 인스턴스에 보조 네트워크 인터페이스가 있는 경우 [네트워크 인터페이스(Network interfaces)]에서 [네트워킹(Networking)] 탭을 선택합니다. 인터페이스 ID를 선택하여 네트워크 인터페이스 페이지로 이동합니다. [작업(Actions)], [소스/대상 변경. 확인(Change source/dest)]을 선택하고 [활성화(Enable)]을 지우고 [저장(Save)]을 선택합니다.

6. 라우팅 테이블 업데이트

프라이빗 서브넷의 라우팅 테이블에는 NAT 인스턴스로 인터넷 트래픽을 보내는 경로가 있어야 합니다.

라우팅 테이블을 업데이트하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Route tables을 선택합니다.
3. 프라이빗 서브넷에 대한 라우팅 테이블을 선택합니다.
4. 경로(Routes) 탭에서 경로 편집(Edit routes) 및 경로 추가(Add route)를 차례로 선택합니다.
5. 대상(Destination)에 0.0.0.0/0을, 타겟(Target)에 NAT 인스턴스의 인스턴스 ID를 입력합니다.
6. 변경 사항 저장을 선택합니다.

자세한 내용은 라우팅 테이블 구성 섹션을 참조하세요.

7. NAT 인스턴스 테스트

NAT 인스턴스를 시작하고 위의 구성 단계를 완료한 후, NAT 인스턴스를 bastion 서버로 사용하여 프라이빗 서브넷의 인스턴스가 NAT 인스턴스를 통해 인터넷에 액세스할 수 있는지 여부를 테스트할 수 있습니다.

작업

- 1단계: NAT 인스턴스 보안 그룹 업데이트
- 2단계: 프라이빗 서브넷에서 테스트 인스턴스 시작
- 3단계: ICMP 지원 웹사이트 ping
- 4단계: 정리

1단계: NAT 인스턴스 보안 그룹 업데이트

프라이빗 서브넷의 인스턴스가 ping 트래픽을 NAT 인스턴스로 전송하도록 허용하려면 인바운드 및 아웃바운드 ICMP 트래픽을 허용하는 규칙을 추가합니다. NAT 인스턴스가 Bastion 서버 역할을 하도록 허용하려면 프라이빗 서브넷으로의 아웃바운드 SSH 트래픽을 허용하는 규칙을 추가합니다.

NAT 인스턴스의 보안 그룹 업데이트

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 보안 그룹을 선택합니다.
3. NAT 인스턴스와 연결된 보안 그룹의 확인란을 선택합니다.
4. [인바운드 규칙(Inbound rules)] 탭에서 [인바운드 규칙 편집(Edit inbound rules)]을 선택합니다.
5. [Add another rule]을 선택합니다. [유형(Type)]에서 [모든 ICMP - IPv4(All ICMP - IPv4)]를 선택합니다. 소스(Source)에서 사용자 정의(Custom)를 선택하고 프라이빗 서브넷의 IP 주소 범위를 입력합니다. 규칙 저장을 선택합니다.
6. 아웃바운드 규칙(Outbound rules) 탭에서 아웃바운드 규칙 편집(Edit outbound rules)을 선택합니다.
7. [Add another rule]을 선택합니다. [유형(Type)]에서 SSH를 선택합니다. Destination(대상)에서 사용자 정의(Custom)를 선택하고 프라이빗 서브넷의 IP 주소 범위를 입력합니다.
8. [Add another rule]을 선택합니다. [유형(Type)]에서 [모든 ICMP - IPv4(All ICMP - IPv4)]를 선택합니다. 대상(Destination)에 대해 어디서나 - IPv4(Anywhere - IPv4)를 선택합니다 규칙 저장을 선택합니다.

2단계: 프라이빗 서브넷에서 테스트 인스턴스 시작

프라이빗 서브넷으로 인스턴스를 시작합니다. NAT 인스턴스에서 SSH 액세스를 허용해야 하며 NAT 인스턴스에 사용한 것과 동일한 키 쌍을 사용해야 합니다.

프라이빗 서브넷에서 테스트 인스턴스 시작

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

2. 대시보드에서 인스턴스 시작을 선택합니다.
3. 프라이빗 서브넷을 선택합니다.
4. 이 인스턴스에는 퍼블릭 IP 주소를 할당하지 마세요.
5. 이 인스턴스의 보안 그룹은 NAT 인스턴스 또는 퍼블릭 서브넷의 IP 주소 범위에서 오는 인바운드 SSH 액세스와 아웃바운드 ICMP 트래픽를 허용해야 합니다.
6. NAT 인스턴스에 사용한 것과 동일한 키 페어를 선택합니다.

3단계: ICMP 지원 웹사이트 ping

프라이빗 서브넷의 테스트 인스턴스가 NAT 인스턴스를 사용하여 인터넷과 통신할 수 있는지 확인하려면 ping 명령을 실행합니다.

프라이빗 인스턴스에서 인터넷 연결을 테스트하려면

1. 로컬 컴퓨터에서 SSH 에이전트 전달을 구성하여 NAT 인스턴스를 Bastion 서버로 사용할 수 있습니다.

   Linux and macOS

   ```
   ssh-add key.pem
   ```

   Windows

   아직 설치되지 않은 경우 Pageant를 다운로드하여 설치합니다.

   PuTTYgen을 사용하여 프라이빗 키를 변환합니다.

   Pageant를 실행하고 작업 표시줄의 Pageant 아이콘(숨겨져 있을 수 있음)을 마우스 오른쪽 버튼으로 클릭한 후 Add Key를 선택하세요. 생성한 .ppk 파일을 선택하고 필요한 경우 암호를 입력한 다음 열기를 선택합니다.
2. 로컬 컴퓨터에서 NAT 인스턴스에 연결합니다.

   Linux and macOS

   ```
   ssh -A ec2-user@nat-instance-public-ip-address
   ```

Windows

PuTTY를 사용하여 NAT 인스턴스에 연결합니다. 인증의 경우 에이전트 전달 허용을 선택하고 인증을 위한 프라이빗 키 파일을 비워 두어야 합니다.

3. NAT 인스턴스에서 ping 명령을 실행하여 ICMP에 대해 활성화된 웹 사이트를 지정합니다.

```
[ec2-user@ip-10-0-4-184]$ ping ietf.org
```

NAT 인스턴스에 인터넷 액세스 권한이 있는지 확인하려면 다음과 같은 출력을 수신했는지 확인한 다음 Ctrl+C를 눌러 ping 명령을 취소합니다. 아니면 NAT 인스턴스가 퍼블릭 서브넷에 있는지 확인합니다(라우팅 테이블에 인터넷 게이트웨이에 대한 경로가 있음).

```
PING ietf.org (104.16.45.99) 56(84) bytes of data.
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=1 ttl=33 time=7.88 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=2 ttl=33 time=8.09 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=3 ttl=33 time=7.97 ms
...
```

4. NAT 인스턴스에서 프라이빗 IP 주소를 사용하여 프라이빗 서브넷의 인스턴스에 연결합니다.

```
[ec2-user@ip-10-0-4-184]$ ssh ec2-user@private-server-private-ip-address
```

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

```
[ec2-user@ip-10-0-135-25]$ ping ietf.org
```

프라이빗 인스턴스가 NAT 인스턴스를 통해 인터넷에 액세스할 수 있는지 확인하려면 다음과 같은 출력을 수신했는지 확인한 다음 Ctrl+C를 눌러 ping 명령을 취소합니다.

```
PING ietf.org (104.16.45.99) 56(84) bytes of data.
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=1 ttl=33 time=8.76 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=2 ttl=33 time=8.26 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=3 ttl=33 time=8.27 ms
...
```

문제 해결

프라이빗 서브넷의 서버에서 ping 명령이 실패하는 경우 다음 단계에 따라 문제를 해결하십시오.

- ICMP가 활성화된 웹 사이트를 ping했는지 확인합니다. 수행되지 않았으면 서버가 응답 패킷을 수신할 수 없습니다. 이를 테스트하려면 사용자 컴퓨터의 명령줄 터미널에서 똑같은 ping 명령을 수행하세요.
- NAT 인스턴스의 보안 그룹이 프라이빗 서브넷에서 오는 인바운드 ICMP 트래픽을 허용하는지 확인합니다. 허용하지 않으면, NAT 인스턴스가 프라이빗 인스턴스로부터 ping 명령을 수신할 수 없습니다.
- NAT 인스턴스에 대해 소스/대상 확인을 비활성화했는지 확인합니다. 자세한 내용은 5. 원본/대상 확인 비활성화 섹션을 참조하세요.
- 라우팅 테이블을 올바로 구성했는지 확인합니다. 자세한 내용은 6. 라우팅 테이블 업데이트 섹션을 참조하세요.

4단계: 정리

프라이빗 서브넷에 테스트 서버가 더 이상 필요하지 않은 경우 더 이상 요금이 청구되지 않도록 인스턴스를 종료합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 종료를 참조하세요.

NAT 인스턴스가 더 이상 필요하지 않은 경우 요금이 더 이상 청구되지 않도록 중지하거나 종료할 수 있습니다. NAT AMI를 생성한 경우 필요할 때마다 새 NAT 인스턴스를 생성할 수 있습니다.

NAT 게이트웨이 및 NAT 인스턴스 비교

NAT 인스턴스와 NAT 게이트웨이의 차이점을 세부적으로 요약하면 다음과 같습니다. NAT 게이트웨이는 더 나은 가용성과 대역폭을 제공하고 관리에 소요되는 작업이 줄어들기 때문에 권장합니다.

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 100Gbps까지 확장합니다.	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).

속성	NAT 게이트웨이	NAT 인스턴스
성능	소프트웨어가 NAT 트래픽 처리에 최적화되어 있습니다.	NAT를 수행하도록 구성된 일반 AMI입니다.
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	단일 제공 항목이므로, 유형이나 크기를 별도로 결정하지 않아도 됩니다.	예상 워크로드에 따라 적합한 인스턴스 유형과 크기를 선택합니다.
퍼블릭 IP 주소	생성할 때 퍼블릭 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	보안 그룹을 NAT 게이트웨이와 연결할 수 없습니다. 보안 그룹을 NAT 게이트웨이 기반 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.
Bastion 서버	지원하지 않음.	Bastion 서버로 사용합니다.

속성	NAT 게이트웨이	NAT 인스턴스
트래픽 지표	NAT 게이트웨이에 대한 CloudWatch 지표를 확인합니다.	인스턴스에 대한 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.

NAT 인스턴스에서 NAT 게이트웨이로 마이그레이션

이미 NAT 인스턴스를 사용하는 경우 이를 NAT 게이트웨이로 대체하는 것이 좋습니다. NAT 인스턴스와 동일한 서브넷에 NAT 게이트웨이를 만든 다음, NAT 인스턴스를 가리키는 라우팅 테이블의 기존 경로를 NAT 게이트웨이를 가리키는 경로로 대체할 수 있습니다. 현재 NAT 인스턴스에 사용하는 것과 동일한 탄력적 IP 주소를 NAT 게이트웨이에 사용하려는 경우에도 먼저 NAT 인스턴스의 탄력적 IP 주소를 연결 해제하고 NAT 게이트웨이를 만들 때 이 주소를 게이트웨이에 연결해야 합니다.

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.

탄력적 IP 주소를 VPC의 리소스와 연결

탄력적 IP 주소는 클라우드 컴퓨팅의 동적 특성을 위해 고안된 고정 퍼블릭 IPv4 주소입니다. 이 기능을 사용하면 탄력적 IP 주소를 AWS 계정의 Virtual Private Cloud(VPC) 내의 모든 인스턴스 또는 네트워크 인터페이스와 연결할 수 있습니다. 탄력적 IP 주소를 사용하면 클라우드 기반 인프라의 관리와 복원력을 단순화하여 다양한 이점을 누릴 수 있습니다.

탄력적 IP 주소의 주요 장점 중 하나는 인스턴스 오류를 마스킹하는 기능입니다. 인스턴스에 예상치 못한 중단이 발생하거나 교체가 필요한 경우 연결된 탄력적 IP 주소를 VPC 내의 다른 인스턴스에 다시

매핑할 수 있습니다. 이 장애 조치 프로세스를 통해 애플리케이션과 서비스가 일관되고 안정적인 퍼블릭 엔드포인트를 유지하여 가동 중지 시간을 최소화하고 우수한 사용자 경험을 제공할 수 있습니다.

또한 탄력적 IP 주소를 사용하여 네트워크 리소스를 유연하게 관리할 수 있습니다. 필요에 따라 이러한 주소를 프로그래밍 방식으로 연결하고 연결 해제하여 변화하는 비즈니스 요구 사항에 따라 트래픽을 다른 인스턴스로 보낼 수 있습니다. 이렇게 동적으로 퍼블릭 IP 주소를 할당하면 고정 IP 할당의 제약 없이 변화하는 수요에 적응하고 인프라를 확장하고 혁신적인 아키텍처를 구현할 수 있습니다.

탄력적 IP 주소는 인스턴스 장애 조치에 사용하는 것 외에도 클라우드 기반 리소스의 안정적인 식별자 역할도 할 수 있습니다. 이는 AWS 호스팅 애플리케이션과 통신하도록 DNS 레코드, 방화벽 규칙 등의 외부 서비스를 구성할 때 유용할 수 있습니다. 영구 퍼블릭 IP 주소를 연결하면 네트워킹 구성을 미래에도 사용할 수 있으며 기본 인스턴스를 교체하거나 확장할 때 외부 참조를 업데이트할 필요가 없습니다.

내용

- 탄력적 IP 주소 개념 및 규칙
- 탄력적 IP 주소 사용 시작

탄력적 IP 주소 개념 및 규칙

탄력적 IP 주소를 사용하려면 먼저 계정에서 사용할 수 있도록 할당합니다. 그런 다음, VPC의 인스턴스 또는 네트워크 인터페이스와 연결할 수 있습니다. 탄력적 IP 주소는 명시적으로 릴리스될 때까지 AWS 계정에 할당되어 있습니다.

탄력적 IP 주소는 네트워크 인터페이스의 속성입니다. 인스턴스에 연결된 네트워크 인터페이스를 업데이트하여 탄력적 IP 주소를 인스턴스와 연결할 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. 자세한 내용은 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스를 참조하세요.

다음 규칙이 적용됩니다.

- 탄력적 IP 주소는 한 번에 단일 인스턴스 또는 네트워크 인터페이스와 연결할 수 있습니다.
- 인스턴스 또는 네트워크 인터페이스 간에 탄력적 IP 주소를 이동할 수 있습니다.
- 탄력적 IP 주소를 인스턴스의 주 네트워크 인터페이스와 연결하면, 현재 퍼블릭 IPv4 주소(있는 경우)는 퍼블릭 IP 주소 풀로 연결 해제됩니다. 탄력적 IP 주소의 연결을 해제하면 몇 분 내에 자동으로 주 네트워크 인터페이스에 새 퍼블릭 IPv4 주소가 배정됩니다. 인스턴스에 보조 네트워크 인터페이스를 연결한 경우에는 이 동작이 적용되지 않습니다.

- 탄력적 IP 주소는 5개로 제한됩니다. 주소를 절약하기 위해 NAT 디바이스를 사용할 수 있습니다. 자세한 내용은 NAT 디바이스를 사용하여 인터넷 또는 다른 네트워크에 연결 단원을 참조하세요.
- IPv6에 대한 탄력적 IP 주소는 지원되지 않습니다.
- VPC에 사용하도록 할당된 탄력적 IP 주소에 태그를 지정할 수 있지만, 비용 할당 태그가 지원되지 않습니다. 탄력적 IP 주소를 복구하는 경우, 태그가 복구되지 않습니다.
- 보안 그룹 및 네트워크 ACL이 원본 IP 주소의 트래픽을 허용하면 인터넷에서 탄력적 IP 주소에 액세스할 수 있습니다. VPC 내에서 인터넷으로 다시 돌아가는 응답 트래픽에는 인터넷 게이트웨이가 필요합니다. 자세한 내용은 보안 그룹 및 네트워크 ACL(을)를 참조하세요.
- 탄력적 IP 주소에 대해 다음 옵션 중 하나를 사용할 수 있습니다.
  - Amazon이 탄력적 IP 주소를 제공하도록 합니다. 이 옵션을 선택하면 탄력적 IP 주소를 네트워크 경계 그룹과 연결할 수 있습니다. 이는 CIDR 블록을 공고하는 위치입니다. 네트워크 경계 그룹을 설정하면 CIDR 블록이 이 그룹으로 제한됩니다.
  - 고유 IP 주소를 사용합니다. 고유 IP 주소 가져오기에 대한 자세한 내용은 Amazon EC2 사용 설명서의 고유 IP 주소 가져오기(BYOIP)를 참조하세요.
- 퍼블릭 IPv4 주소에서는 비용 할당 태그가 지원됩니다. 탄력적 IP 주소에 태그를 적용하면 AWS Cost Explorer에서 해당 태그를 사용하여 퍼블릭 IPv4 주소 비용을 추적할 수 있습니다.

  태그를 비용 할당 태그로 사용하려면 먼저 태그를 활성화해야 합니다. 자세한 내용은 AWS Billing 사용 설명서에서 사용자 정의 비용 할당 태그 활성화를 참조하세요. 참고로, 사용자 정의 태그를 생성하여 리소스에 적용한 후 태그 키가 활성화를 위해 비용 할당 태그 페이지에 나타나는 데 최대 24시간이 걸릴 수 있습니다.

  비용 할당 태그가 활성화되면...

  - 탄력적 네트워크 인터페이스와 연결된 모든 퍼블릭 IPv4 주소(EC2 인스턴스에 할당된 퍼블릭 IPv4 주소 및 탄력적 IP 주소 포함)의 경우, 사용량 유형 > PublicIPv4InUseAddress(시간)를 선택하여 Cost Explorer에서 퍼블릭 IPv4 주소와 연결된 비용을 볼 수 있습니다.
  - 태그가 지정된 탄력적 IP 주소가 ENI와 연결되지 않았거나 중지된 리소스(예: 중지된 EC2 인스턴스)와 연결된 경우에는 유휴 IPv4 주소로 간주합니다. 사용량 유형 > PublicIPv4IdleAddress(시간)을 선택하여 Cost Explorer에서 유휴 IPv4 주소와 연결된 비용을 볼 수 있습니다.

  Cost Explorer에 대한 자세한 내용은 AWS Billing 사용 설명서의 AWS Cost Explorer를 사용한 비용 분석을 참조하세요.

탄력적 IP 주소는 리전별입니다. Global Accelerator를 사용하여 글로벌 IP 주소를 프로비저닝하는 것에 대한 자세한 내용은 AWS Global Accelerator 개발자 안내서의 리전별 고정 IP 주소 대신 글로벌 고정 IP 주소 사용을 참조하세요.

탄력적 IP 주소 요금에 대한 자세한 내용은 Amazon VPC 요금의 퍼블릭 IPv4 주소를 참조하세요.

탄력적 IP 주소 사용 시작

다음 섹션에서는 탄력적 IP 주소 사용을 시작하는 방법을 설명합니다.

작업

- 1. 탄력적 IP 주소 할당
- 2. 탄력적 IP 주소 연결
- 3. 탄력적 IP 주소 연결 해제
- 4. 탄력적 IP 주소 전송
- 5. 탄력적 IP 주소 릴리스
- 6. 탄력적 IP 주소 복구
- 명령줄 개요

1. 탄력적 IP 주소 할당

탄력적 IP를 사용하기 전에 VPC에서 사용할 탄력적 IP를 할당해야 합니다.

탄력적 IP 주소 할당

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Elastic IPs를 선택합니다.
3. 탄력적 IP 주소 할당을 선택합니다.
4. (선택 사항) 탄력적 IP 주소(EIP)를 할당할 때는 EIP를 할당할 네트워크 경계 그룹을 선택합니다. 네트워크 경계 그룹은 AWS가 퍼블릭 IP 주소를 알리는 가용 영역, Local Zone 또는 Wavelength Zone의 집합입니다. Local Zone 및 Wavelength Zone은 AWS 네트워크와 해당 영역의 리소스에 액세스하는 고객 간의 지연 시간 또는 물리적 거리를 최소화하기 위해 리전의 AZ와 다른 네트워크 경계 그룹을 가질 수 있습니다.

> **Important**
>
> EIP와 연결될 AWS 리소스와 동일한 네트워크 경계 그룹에 EIP를 할당해야 합니다. 한 네트워크 경계 그룹의 EIP는 해당 네트워크 경계 그룹의 영역에서만 알릴 수 있으며 다른 네트워크 경계 그룹이 나타내는 다른 영역에서는 알릴 수 없습니다.

Local Zone 또는 Wavelength Zone을 활성화한 경우(자세한 내용은 Local Zone 활성화 또는 Wavelength Zone 활성화 참조) AZ, Local Zone 또는 Wavelength Zone에 대한 네트워크 경계 그룹을 선택할 수 있습니다. EIP와 연결된 AWS 리소스는 동일한 네트워크 경계 그룹에 있어야 하므로 네트워크 경계 그룹을 신중하게 선택하세요. EC2 콘솔을 사용하여 가용 영역, Local Zone 또는 Wavelength Zone이 속한 네트워크 경계 그룹을 볼 수 있습니다(Local Zones 참조). 일반적으로 한 리전의 모든 가용 영역은 동일한 네트워크 경계 그룹에 속하지만 Local Zone 또는 Wavelength Zone은 별도의 자체 네트워크 경계 그룹에 속합니다.

Local Zone 또는 Wavelength Zone이 활성화되지 않은 경우 EIP를 할당하면 해당 리전의 모든 AZ를 나타내는 네트워크 경계 그룹(예:`us-west-2`)이 미리 정의되어 있으며 변경할 수 없습니다. 즉, 이 네트워크 경계 그룹에 할당한 EIP는 현재 속한 리전의 모든 AZ에 광고됩니다.

5. 퍼블릭 IPv4 주소 풀(Public IPv4 address pool)에서 다음 중 하나를 선택합니다.
   - Amazon의 IP 주소 풀 — IPv4 주소를 Amazon의 IP 주소 풀에서 할당하려는 경우.
   - [내 퍼블릭 IPv4 주소 풀(My pool of public IPv4 addresses)] - AWS 계정으로 가져온 IP 주소 풀에서 IPv4 주소를 할당하려는 경우. IP 주소 풀이 없는 경우에는 이 옵션을 사용할 수 없습니다.
   - Customer owned pool of IPv4 addresses(고객 소유 IPv4 주소 풀)—Outpost에서 사용하기 위해 온프레미스 네트워크에서 만든 풀에서 IPv4 주소를 할당하려는 경우. Outpost가 없는 경우 이 옵션만 사용할 수 있습니다.
6. (선택) 태그를 추가하거나 제거할 수 있습니다.

   [태그 추가] 새 태그 추가를 선택하고 다음을 수행합니다.
   - 키에서 키 이름을 입력합니다.
   - 값에 키 값을 입력합니다.

   [태그 제거] 태그의 키 및 값 오른쪽에 있는 제거를 선택하세요.
7. 할당을 선택합니다.

2. 탄력적 IP 주소 연결

VPC에서 실행 중인 인스턴스 또는 네트워크 인터페이스에 탄력적 IP를 연결할 수 있습니다.

탄력적 IP 주소를 인스턴스와 연결하면 DNS 호스트 이름이 활성화된 경우 인스턴스가 퍼블릭 DNS 호스트 이름을 받습니다. 자세한 내용은 VPC의 DNS 속성 단원을 참조하세요.

인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소를 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Elastic IPs를 선택합니다.
3. VPC에서 사용하기 위해 할당한 탄력적 IP 주소를 선택한 후(범위(Scope) 열에 vpc 값이 표시됨), 작업(Actions), 주소 연결(Associate Address)을 선택합니다.
4. [Instance] 또는 [Network interface]를 선택한 다음 인스턴스 또는 네트워크 인터페이스 ID를 선택합니다. 탄력적 IP 주소를 연결할 프라이빗 IP 주소를 선택합니다. 연결을 선택합니다.

3. 탄력적 IP 주소 연결 해제

탄력적 IP 주소가 연결된 리소스를 변경하려면 먼저 현재 연결된 리소스에서 해당 주소를 연결 해제해야 합니다.

엘라스틱 IP 주소를 연결 해제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Elastic IPs를 선택합니다.
3. 탄력적 IP 주소를 선택한 다음 작업(Actions), 탄력적 IP 주소 연결 해제(Disassociate Elastic IP address)를 선택합니다.
4. 메시지가 나타나면 주소 연결 해제(Disassociate address)를 선택합니다.

4. 탄력적 IP 주소 전송

이 섹션에서는 하나의 AWS 계정에서 다른 계정으로 탄력적 IP 주소를 전송하는 방법을 설명합니다. 탄력적 IP 주소 전송은 다음 상황에서 유용할 수 있습니다.

- 조직 구조 조정 - 탄력적 IP 주소 전송을 사용하여 워크로드를 하나의 AWS 계정에서 다른 계정으로 빠르게 이동합니다. 새 탄력적 IP 주소가 보안 그룹 및 NACL의 허용 목록에 추가될 때까지 기다리지 않아도 됩니다.

- 중앙 집중식 보안 관리 - 중앙 집중식 AWS 보안 계정을 사용하여 보안 규정 준수를 위해 검증된 탄력적 IP 주소를 추적하고 전송합니다.
- 재해 복구 - 탄력적 IP 주소 전송을 사용하여 긴급 상황 동안 공용 인터넷 워크로드의 IP를 신속하게 다시 매핑합니다.

탄력적 IP 주소 전송에는 요금이 부과되지 않습니다.

작업

- 탄력적 IP 주소 전송 활성화
- 탄력적 IP 주소 전송 비활성화
- 전송된 탄력적 IP 주소 수락

탄력적 IP 주소 전송 활성화

이 섹션에서는 전송된 탄력적 IP 주소를 수락하는 방법을 설명합니다. 전송을 위한 탄력적 IP 주소 활성화와 관련해 다음과 같은 제한 사항에 유의하세요.

- 모든 AWS 계정(소스 계정)의 탄력적 IP 주소를 동일한 AWS 리전의 다른 AWS 계정(전송 계정)으로 전송할 수 있습니다.
- 탄력적 IP 주소를 전송할 때 AWS 계정 간에 2단계 핸드셰이크가 발생합니다. 소스 계정에서 전송이 시작되면 전송 계정은 7일 내에 탄력적 IP 주소 전송을 수락해야 합니다. 이 7일 동안 소스 계정은 대기 중인 전송을 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers AWS CLI 명령 사용). 7일이 지나면 전송이 만료되고 탄력적 IP 주소의 소유권이 소스 계정으로 반환됩니다.
- 수락된 전송은 전송이 수락된 후 14일 동안 소스 계정에서 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers AWS CLI 명령 사용).
- AWS에서는 보류 중인 탄력적 IP 주소 전송 요청에 대해 전송 계정에 알리지 않습니다. 소스 계정의 소유자는 반드시 수락해야 하는 탄력적 IP 주소 전송 요청이 있음을 전송 계정 소유자에게 알려야 합니다.
- 전송되는 탄력적 IP 주소와 연결된 모든 태그는 전송이 완료된 후에 재설정됩니다.
- AWS 계정에 가져온 퍼블릭 IPv4 주소 풀(일반적으로 고유 Bring Your Own IP(BYOIP) 주소 풀이라고 함)에서 할당된 탄력적 IP 주소는 전송할 수 없습니다.
- 역방향 DNS 레코드가 연결된 탄력적 IP 주소의 전송을 시도하는 경우 전송 프로세스를 시작할 수는 있습니다. 하지만 연결된 DNS 레코드가 제거될 때까지 수신 계정은 전송을 수락할 수 없습니다.
- AWS Outposts를 활성화하고 구성한 경우 고객 소유의 IP 주소 풀(COIP)에서 탄력적 IP 주소를 할당했을 수 있습니다. CoIP에서 할당된 탄력적 IP 주소는 전송할 수 없습니다. 하지만 AWS RAM을 사

용하여 다른 계정과 CoIP를 공유할 수 있습니다. 자세한 내용은 AWS Outposts 사용 설명서에서 고객 소유 IP 주소를 참조하세요.

- Amazon VPC IPAM을 사용하여 AWS Organizations의 조직 내 계정으로 탄력적 IP 주소 전송을 추적할 수 있습니다. 자세한 내용은 IP 주소 기록 보기를 참조하세요. 그러나 탄력적 IP 주소가 조직 외부의 AWS 계정으로 전송되는 경우 탄력적 IP 주소에 대한 IPAM 감사 기록은 손실됩니다.

소스 계정으로 이 단계를 수행해야 합니다.

탄력적 IP 주소 전송 활성화

1. 소스 AWS 계정을 사용 중인지 확인하세요.
2. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
3. 탐색 창에서 Elastic IPs를 선택합니다.
4. 전송을 활성화할 탄력적 IP 주소를 하나 이상 선택하고 Actions(작업), Enable transfer(전송 활성화)를 선택합니다.
5. 탄력적 IP 주소를 여러 개 전송하는 경우 전송 유형 옵션이 표시됩니다. 다음 옵션 중 하나를 선택하세요.
   - 탄력적 IP 주소를 단일 AWS 계정으로 전송하려는 경우 Single account(단일 계정)를 선택합니다.
   - 탄력적 IP 주소를 여러 AWS 계정으로 전송하려는 경우 Multiple accounts(다중 계정)를 선택합니다.
6. Transfer account ID(전송 계정 ID)에 탄력적 IP 주소를 전송하려는 AWS 계정의 ID를 입력합니다.
7. 텍스트 상자에 **enable**을 입력하여 전송을 확인합니다.
8. 제출을 선택합니다.
9. 전송을 수락하려면 전송된 탄력적 IP 주소 수락 섹션을 참조하세요. 전송을 비활성화하려면 탄력적 IP 주소 전송 비활성화 섹션을 참조하세요.

탄력적 IP 주소 전송 비활성화

이 섹션에서는 탄력적 IP 전송을 활성화한 후 이를 비활성화하는 방법을 설명합니다.

전송을 활성화한 소스 계정으로 다음 단계를 수행해야 합니다.

탄력적 IP 주소 전송 비활성화

1. 소스 AWS 계정을 사용 중인지 확인하세요.
2. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
3. 탐색 창에서 Elastic IPs를 선택합니다.
4. 탄력적 IP의 리소스 목록에서 Transfer status(전송 상태) 열을 표시하는 속성이 활성화되어 있는지 확인합니다.
5. Transfer status(전송 상태)가 Pending(보류 중)인 탄력적 IP 주소를 하나 이상 선택하고 Actions(작업), Disable transfer(전송 비활성화)를 선택합니다.
6. 텍스트 상자에 **`disable`**을 입력하여 확인합니다.
7. 제출을 선택합니다.

전송된 탄력적 IP 주소 수락

이 섹션에서는 전송된 탄력적 IP 주소를 수락하는 방법을 설명합니다.

탄력적 IP 주소를 전송할 때 AWS 계정 간에 2단계 핸드셰이크가 발생합니다. 소스 계정에서 전송이 시작되면 전송 계정은 7일 내에 탄력적 IP 주소 전송을 수락해야 합니다. 이 7일 동안 소스 계정은 대기 중인 전송을 볼 수 있습니다(예: AWS 콘솔에서 보거나 describe-address-transfers AWS CLI 명령 사용). 7일이 지나면 전송이 만료되고 탄력적 IP 주소의 소유권이 소스 계정으로 반환됩니다.

전송을 수락할 때 발생할 수 있는 다음 예외 사항과 이에 대한 해결 방법을 참고하세요.

- AddressLimitExceeded: 전송 계정이 탄력적 IP 주소 할당량을 초과한 경우, 소스 계정에서 탄력적 IP 주소 전송을 활성화할 수 있지만 전송 계정이 전송을 수락하려고 하면 이 예외가 발생합니다. 기본적으로 모든 AWS 계정은 리전당 탄력적 IP 주소 5개로 제한됩니다. 제한 상향에 대한 지침은 Amazon EC2 사용 설명서의 탄력적 IP 주소 제한을 참조하세요.
- InvalidTransfer.AddressCustomPtrSet: 전송하려는 탄력적 IP 주소에 역방향 DNS 조회가 구성되어 있습니다. 원본 계정에서 탄력적 IP 주소의 전송을 활성화할 수 있습니다. 하지만 이 예외는 수신 계정이 전송을 수락하려고 할 때 발생합니다. 이 문제를 해결하기 위해서는 소스 계정에서 탄력적 IP 주소의 DNS 레코드를 제거해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 역방향 DNS 레코드 제거를 참조하세요.
- InvalidTransfer.AddressAssociated: 탄력적 IP 주소가 ENI 또는 EC2 인스턴스와 연결된 경우, 소스 계정에서 탄력적 IP 주소 전송을 활성화할 수 있지만 전송 계정이 전송을 수락하려고 하면 이 예외가 발생합니다. 이 문제를 해결하기 위해서는 탄력적 IP 주소의 연결을 해제해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 탄력적 IP 주소 연결 해제를 참조하세요.

기타 예외 사항은 지원에 문의하세요.

전송 계정으로 이 단계를 수행해야 합니다.

탄력적 IP 주소 전송 수락

1. 전송 계정을 사용 중인지 확인하세요.
2. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
3. 탐색 창에서 Elastic IPs를 선택합니다.
4. Actions(작업), Accept transfer(전송 수락)를 선택합니다.
5. 전송을 수락하면 전송 중인 탄력적 IP 주소와 연결된 태그가 탄력적 IP 주소로 전송되지 않습니다. 수락하려는 탄력적 IP 주소의 Name(이름) 태그를 정의하려는 경우 Create a tag with a key of 'Name' and a value that you specify('Name' 키와 지정한 값으로 태그 생성)를 선택합니다.
6. 전송할 탄력적 IP 주소를 입력합니다.
7. 전송된 탄력적 IP 주소를 여러 개 수락 중인 경우 Add address(주소 추가)를 선택하여 추가 탄력적 IP 주소를 입력합니다.
8. 제출을 선택합니다.

5. 탄력적 IP 주소 릴리스

탄력적 IP 주소가 더 이상 필요하지 않으면 해당 주소를 릴리스하는 것이 좋습니다. 인스턴스와 연결되어 있지 않더라도 VPC와 함께 사용하도록 할당된 모든 탄력적 IP 주소에 대해서는 요금이 부과됩니다. 탄력적 IP 주소를 인스턴스 또는 네트워크 인터페이스와 연결해서는 안 됩니다.

엘라스틱 IP 주소를 해제합니다

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Elastic IPs를 선택합니다.
3. 탄력적 IP 주소를 선택한 다음 작업(Actions), 탄력적 IP 주소 릴리스(Release Elastic IP addresses)를 선택합니다.
4. 메시지가 나타나면 [Release]를 선택합니다.

6. 탄력적 IP 주소 복구

탄력적 IP 주소를 해제했지만 생각이 바뀐 경우 복구할 수 있습니다. 탄력적 IP 주소가 다른 AWS 계정에 할당되었거나, 복구로 인해 탄력적 IP 주소 할당량을 초과하는 경우에는 탄력적 IP 주소를 복구할 수 없습니다.

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

AWS CLI를 사용하여 탄력적 IP 주소를 복구하려면

allocate-address 명령을 사용한 다음 `--address` 파라미터를 사용하여 IP 주소를 지정합니다.

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

명령줄 개요

명령줄 또는 API를 사용하여 이 섹션에서 설명하는 태스크를 수행할 수 있습니다. 명령줄 인터페이스 및 사용 가능한 API 작업 목록에 대한 자세한 내용은 Amazon VPC 작업 단원을 참조하세요.

탄력적 IP 주소 전송 수락

- accept-address-transfer(AWS CLI)
- Approve-EC2AddressTransfer(AWS Tools for Windows PowerShell)

탄력적 IP 주소 할당

- allocate-address (AWS CLI)
- New-EC2Address(AWS Tools for Windows PowerShell)

인스턴스 또는 네트워크 인터페이스에 탄력적 IP 주소 연결

- associate-address(AWS CLI)
- Register-EC2Address(AWS Tools for Windows PowerShell)

탄력적 IP 주소 전송 설명

- describe-address-transfers(AWS CLI)
- Get-EC2AddressTransfer(AWS Tools for Windows PowerShell)

탄력적 IP 주소 전송 비활성화

- disable-address-transfer(AWS CLI)
- Disable-EC2AddressTransfer(AWS Tools for Windows PowerShell)

탄력적 IP 주소 연결 해제

- disassociate-address(AWS CLI)
- Unregister-EC2Address(AWS Tools for Windows PowerShell)

탄력적 IP 주소 전송 활성화

- enable-address-transfer(AWS CLI)
- Enable-EC2AddressTransfer(AWS Tools for Windows PowerShell)

탄력적 IP 주소 릴리스

- release-address(AWS CLI)
- Remove-EC2Address(AWS Tools for Windows PowerShell)

탄력적 IP 주소 태그

- create-tags(AWS CLI)
- New-EC2Tag(AWS Tools for Windows PowerShell)

탄력적 IP 주소 보기

- describe-addresses(AWS CLI)
- Get-EC2Address(AWS Tools for Windows PowerShell)

전송 게이트웨이를 사용하여 다른 VPC 및 네트워크에 VPC 연결

VPC, VPN 연결 및 Direct Connect 연결 간에 트래픽을 라우팅하는 중앙 허브 역할을 하는 전송 게이트웨이를 사용하여 Virtual Private Cloud(VPC)와 온프레미스 네트워크를 연결할 수 있습니다.

전송 게이트웨이 사용의 주요 이점 중 하나는 VPC와 온프레미스 네트워크 간의 연결 관리를 중앙 집중화하고 단순화할 수 있다는 것입니다. 여러 VPN 연결이나 Direct Connect 링크를 구성하는 대신 Transit Gateway를 단일 통합 지점으로 사용하면 네트워크 아키텍처의 전반적인 복잡성과 운영 오버헤드를 줄이는 데 도움이 됩니다.

전송 게이트웨이 사용 요금은 게이트웨이를 통해 전송되는 데이터 양을 기준으로 책정됩니다. 전송 게이트웨이 내외로 전송되는 데이터에는 GB당 요금이 적용되고 전송 게이트웨이 리소스 자체에는 별도의 시간당 요금이 적용됩니다. 구체적인 요금은 AWS 리전에 따라 다르고 변경될 수 있으므로 최신 정보를 보려면 최신 AWS Transit Gateway 요금 페이지를 참조해야 합니다. 전송 게이트웨이의 요금 모델을 이해하면 이 AWS 네트워킹 서비스와 관련된 지속적인 비용을 더 잘 계획하고 예산을 책정할 수 있습니다. 이에 운영 효율성과 연결 이점이 결합되어 확장 가능하고 비용 효율적인 하이브리드 클라우드 솔루션을 구축하고자 하는 조직에 전송 게이트웨이는 탁월한 선택이 됩니다.

다음 표에서는 전송 게이트웨이의 몇 가지 일반적인 사용 사례를 설명합니다. 각 사용 사례에 대한 자세한 내용은 AWS Transit Gateway 사용 설명서의 전송 게이트웨이 시나리오 예제를 참조하세요.

예제	사용량
중앙 집중식 라우터	Transit Gateway를 모든 VPC, AWS Direct Connect 및 AWS Site-to-Site VPN 연결을 연결하는 중앙 집중식 라우터로 구성합니다.
격리된 VPC	Transit Gateway를 여러 격리된 라우터로 구성합니다. 이는 여러 개의 Transit Gateway를 사용하는 것과 유사하지만 라우팅 및 연결이 변경될 수 있는 경우 더 많은 유연성을 제공합니다.
공유 서비스를 사용하는 격리된 VPC	Transit Gateway를 공유 서비스를 사용하는 여러 격리된 라우터로 구성합니다. 이는 여러 개의 Transit Gateway를 사용하는 것과 유사하지만 라우팅 및 연결이 변경될 수 있는 경우 더 많은 유연성을 제공합니다.

자세한 내용은 AWS Transit Gateway를 참조하세요.

AWS Virtual Private Network를 사용하여 VPC를 원격 네트워크에 연결

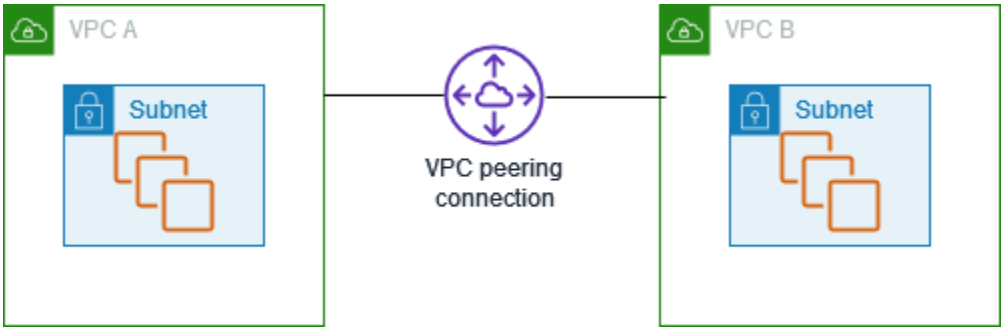
다음 VPN 연결 옵션을 사용하여 VPC를 원격 네트워크 및 사용자에 연결할 수 있습니다.

VPN 연결 옵션	설명
AWS Site-to-Site VPN	VPC와 원격 네트워크 사이에 IPsec VPN 연결을 생성할 수 있습니다. AWS 측 Site-to-Site VPN 연결에서 가상 프라이빗 게이트웨이 또는 Transit Gateway는 자동 장애 조치를 위한 2개의 VPN 엔드포인트(터널)를 제공합니다. Site-to-Site VPN 원격 연결 측에서 고객 게이트웨이 디바이스를 구성합니다. 자세한 내용은 AWS Site-to-Site VPN 사용 설명서를 참조하세요.
AWS Client VPN	AWS Client VPN은 AWS 리소스와 온프레미스 네트워크에 안전하게 액세스할 수 있게 해주는 관리형 클라이언트 기반 VPN 서비스입니다. AWS Client 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 리소스에 액세스할 수 있습니다. 자세한 내용은 AWS Client VPN 관리자 안내서를 참조하세요.
AWS VPN CloudHub	원격 네트워크가 두 개 이상인 경우(예: 여러 지사 사무실), 가상 프라이빗 게이트웨이를 통해 AWS Site-to-Site VPN 연결을 여러 개 만들고 이들 네트워크 사이의 통신을 활성화할 수 있습니다. 자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 VPN CloudHub를 사용하여 사이트 간 보안 통신 제공을 참조하세요.
타사 소프트웨어 VPN 어플라이언스	서드 파티 소프트웨어 VPN 어플라이언스를 실행 중인 VPC에서 Amazon EC2 인스턴스를 사용하여 원격 네트워크에 대한 VPN 연결을 생성할 수 있습니다. AWS는 서드 파티 소프트웨어 VPN 어플라이언스를 제공하거나 유지 관리하지 않지만, 파트너와 오픈 소스 커뮤니티에서 제공하는 다양한 제품 중에서 선택할 수 있습니다. 에서 서드 파티 소프트웨어 VPN 어플라이언스를 찾으세요..AWS Marketplace

또한 Direct Connect를 사용하여 원격 네트워크에서 VPC까지 전용 프라이빗 연결을 생성할 수도 있습니다. 이 연결을 AWS Site-to-Site VPN과 결합하여 IPsec 암호화 연결을 생성할 수 있습니다. 자세한 정보는 Direct Connect 사용 설명서의 Direct Connect(이)란 무엇입니까? 섹션을 참조하세요.

VPC 피어링을 사용하여 VPC 연결

VPC 피어링 연결은 AWS 인프라 내의 두 Virtual Private Cloud(VPC) 간에 안전하고 직접적인 통신을 가능하게 하는 네트워킹 기능입니다. 이 프라이빗 연결을 통해 피어링된 VPC의 리소스는 마치 동일한 네트워크에 속한 것처럼 서로 상호 작용할 수 있으므로 퍼블릭 인터넷을 통과할 필요가 없습니다.



VPC 피어링 연결 생성 시 별도의 게이트웨이, AWS Site-to-Site VPN 또는 추가 물리 하드웨어 없이 기존 VPC 인프라를 활용하여 연결을 구성합니다. 이 설계를 통해 단일 장애 지점이나 대역폭 병목 현상이 발생하지 않습니다.

VPC 피어링 연결의 주요 장점 중 하나는 다양한 AWS 계정 또는 다양한 AWS 리전에 걸쳐 VPC를 연결할 수 있다는 것입니다. 이러한 유연성을 통해 조직은 클라우드 리소스가 동일한 계정 내에 있든 여러 계정과 지리적 위치에 분산되어 있든 관계없이 원활하게 클라우드 리소스를 통합할 수 있습니다. 또한 연결의 프라이빗 특성으로 인해 피어링된 VPC 간의 모든 데이터 트래픽은 퍼블릿 인터넷을 통과하지 않고 AWS 네트워크 내에 유지됩니다.

VPC 피어링 연결의 사용 사례는 광범위합니다. 조직에서는 이 기능을 사용하여 애플리케이션의 다양한 계층(예: 웹 서버 및 데이터베이스 서버) 간 보안 통신을 구현하고, 여러 팀이나 사업부 간 리소스 공유를 지원하며, 온프레미스 네트워크를 AWS VPC에 연결하여 하이브리드 클라우드 아키텍처를 구현할 수 있습니다.

VPC 피어링 연결은 비공개적으로 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 피어링된 VPC의 리소스는 동일한 네트워크 내에 있는 것처럼 서로 통신할 수 있습니다. 자체 VPC 간, 다른 AWS 계정에서 VPC를 사용하여 또는 다른 AWS 리전에서 VPC를 사용하여 VPC 피어링 연결을 생성할 수 있습니다. 피어링된 VPC 간의 트래픽은 공용 인터넷을 통과하지 않습니다.

자세한 내용은 Amazon VPC 피어링 가이드를 참조하세요.

VPC 모니터링

다음 도구를 사용하여 가상 프라이빗 클라우드(VPC)에서 트래픽 또는 네트워크 액세스를 모니터링할 수 있습니다.

VPC 흐름 로그

VPC 흐름 로그를 사용하여 VPC의 네트워크 인터페이스에서 송수신되는 트래픽에 대한 세부 정보를 캡처할 수 있습니다.

Amazon CloudWatch Internet Monitor

인터넷 문제가 AWS에서 호스팅되는 애플리케이션과 최종 사용자 간 성능 및 가용성에 미치는 영향에 대한 가시성에 Internet Monitor를 사용할 수 있습니다. 다른 서비스를 사용하도록 전환하거나 다른 AWS 리전를 통해 워크로드 트래픽을 다시 라우팅하여 애플리케이션의 예상 지연 시간을 개선하는 방법을 거의 실시간으로 탐색할 수도 있습니다. 자세한 내용은 Amazon CloudWatch Internet Monitor API 사용을 참조하세요.

Amazon VPC IP Address Manager(IPAM)

IPAM을 사용하여 워크로드의 IP 주소를 계획, 추적 및 모니터링할 수 있습니다. 자세한 내용은 IP 주소 관리자를 참조하세요.

트래픽 미러링

이 기능을 사용하여 Amazon EC2 인스턴스의 네트워크 인터페이스에서 네트워크 트래픽을 복사하고 심층 패킷 검사를 위해 대역 외 보안 및 모니터링 어플라이언스로 전송할 수 있습니다. 네트워크 및 보안 이상을 감지하고, 운영 인사이트를 얻고, 규정 준수 및 보안 제어를 구현하고, 문제를 해결할 수 있습니다. 자세한 내용은 트래픽 미러링을 참조하세요.

Reachability Analyzer

이 도구를 사용하여 VPC에 있는 두 리소스 간의 네트워크 연결성을 분석하고 디버깅할 수 있습니다. 소스 및 대상 리소스를 지정한 후 Reachability Analyzer는 연결할 수 있는 경우 두 리소스 간의 가상 경로에 대한 홉별 세부 정보를 생성하고 연결할 수 없는 경우 차단 구성 요소를 식별합니다. 자세한 내용은 Reachability Analyzer를 참조하세요.

Network Access Analyzer

Network Access Analyzer를 사용하여 리소스에 대한 네트워크 액세스를 파악할 수 있습니다. 이를 통해 네트워크 보안 태세의 개선 사항을 식별하고 네트워크가 특정 규정 준수 요구 사항을 충족함을 입증할 수 있습니다. 자세한 내용은 Network Access Analyzer를 참조하세요.

CloudTrail 로그

AWS CloudTrail은 다음과 같은 Amazon VPC에 대한 API 호출을 로깅합니다.

- API 호출 내용(예: VPC 리소스 생성 또는 수정과 같은 작업)
- 호출의 소스 IP 주소
- 호출한 사람
- 호출한 시간

`CreateVpc`, `DeleteVpc` 및 `CreateDefaultVpc` 작업에 대해 별도의 로그가 생성됩니다. 이러한 로그에는 VPC와 연결되고 생성된 기본 리소스(예: 기본 인터넷 게이트웨이 또는 기본 보안 그룹)도 포함됩니다.

자세한 내용은 Amazon EC2 사용 설명서의 AWS CloudTrail을 사용하여 Amazon EC2 API 직접 호출 로깅을 참조하세요.

VPC 흐름 로그를 사용하여 IP 트래픽 로깅

VPC 흐름 로그는 VPC의 네트워크 인터페이스에서 전송되고 수신되는 IP 트래픽에 대한 정보를 수집할 수 있는 기능입니다. 흐름 로그 데이터가 게시될 수 있는 위치는 Amazon CloudWatch Logs, Amazon S3 또는 Amazon Data Firehose입니다. 네트워크 트래픽 로그를 CloudWatch Logs 또는 S3와 같은 대상으로 전송할 수 있도록 구성된 전송 경로와 권한을 구독이라고 합니다. 흐름 로그를 생성하면 구성한 로그 그룹, 버킷 또는 전송 스트림의 흐름 로그 레코드를 검색하고 볼 수 있습니다.

흐름 로그는 다음과 같은 여러 작업에 도움이 될 수 있습니다.

- 지나치게 제한적인 보안 그룹 규칙 진단
- 인스턴스에 도달하는 트래픽 모니터링
- 네트워크 인터페이스를 오가는 트래픽 방향 결정

흐름 로그 데이터는 네트워크 트래픽 경로 외부에서 수집되므로 네트워크 처리량이나 지연 시간에 영향을 주지 않습니다. 네트워크 성능에 영향을 주지 않고 흐름 로그를 생성하거나 삭제할 수 있습니다.

> **Note**
>
> 이 섹션에서는 VPC의 흐름 로그에 대해서만 설명합니다. 버전 6에 도입된 전송 게이트웨이의 흐름 로그에 대한 자세한 내용은 Amazon VPC Transit Gateway 사용 설명서의 Logging network traffic using Transit Gateway Flow Logs를 참조하세요.

내용

- 흐름 로그 기본 사항
- 흐름 로그 레코드
- 흐름 로그 레코드의 예시
- 흐름 로그 제한
- 가격 책정
- 흐름 로그 작업
- CloudWatch Logs에 흐름 로그 게시
- Amazon S3에 흐름 로그 게시
- Amazon Data Firehose에 흐름 로그 게시
- Amazon Athena를 사용하여 흐름 로그 쿼리
- VPC 흐름 로그 피어링의 ID 및 액세스 관리
- VPC 흐름 로그 문제 해결

흐름 로그 기본 사항

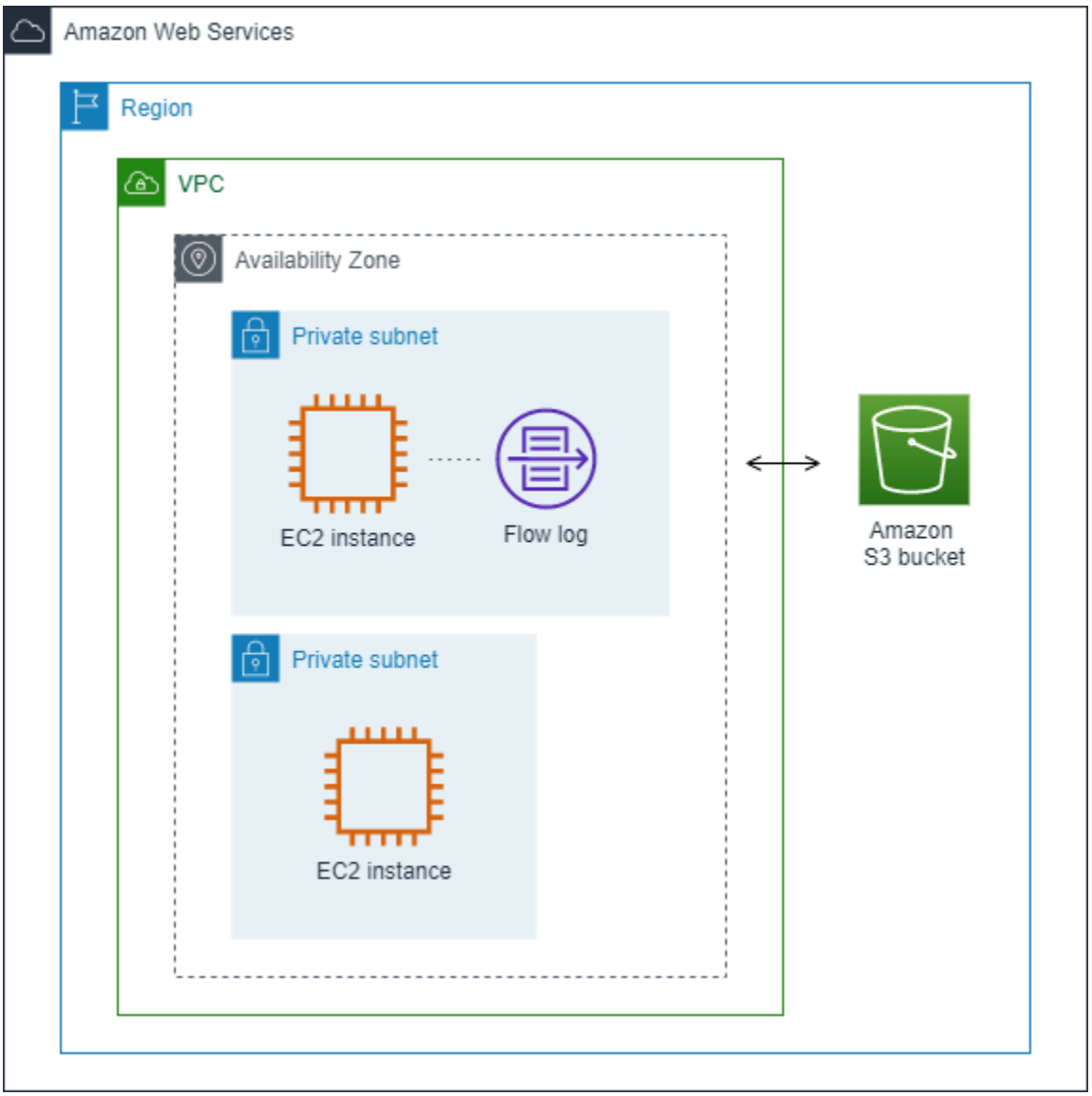
VPC, 서브넷 또는 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 서브넷이나 VPC에 대한 흐름 로그를 생성할 경우, VPC 또는 서브넷의 각 네트워크 인터페이스가 모니터링됩니다.

모니터링된 네트워크 인터페이스를 위한 흐름 로그 데이터는 트래픽 흐름을 설명하는 필드로 구성된 로그 이벤트인 흐름 로그 레코드로서 기록됩니다. 자세한 내용은 흐름 로그 레코드 단원을 참조하세요.

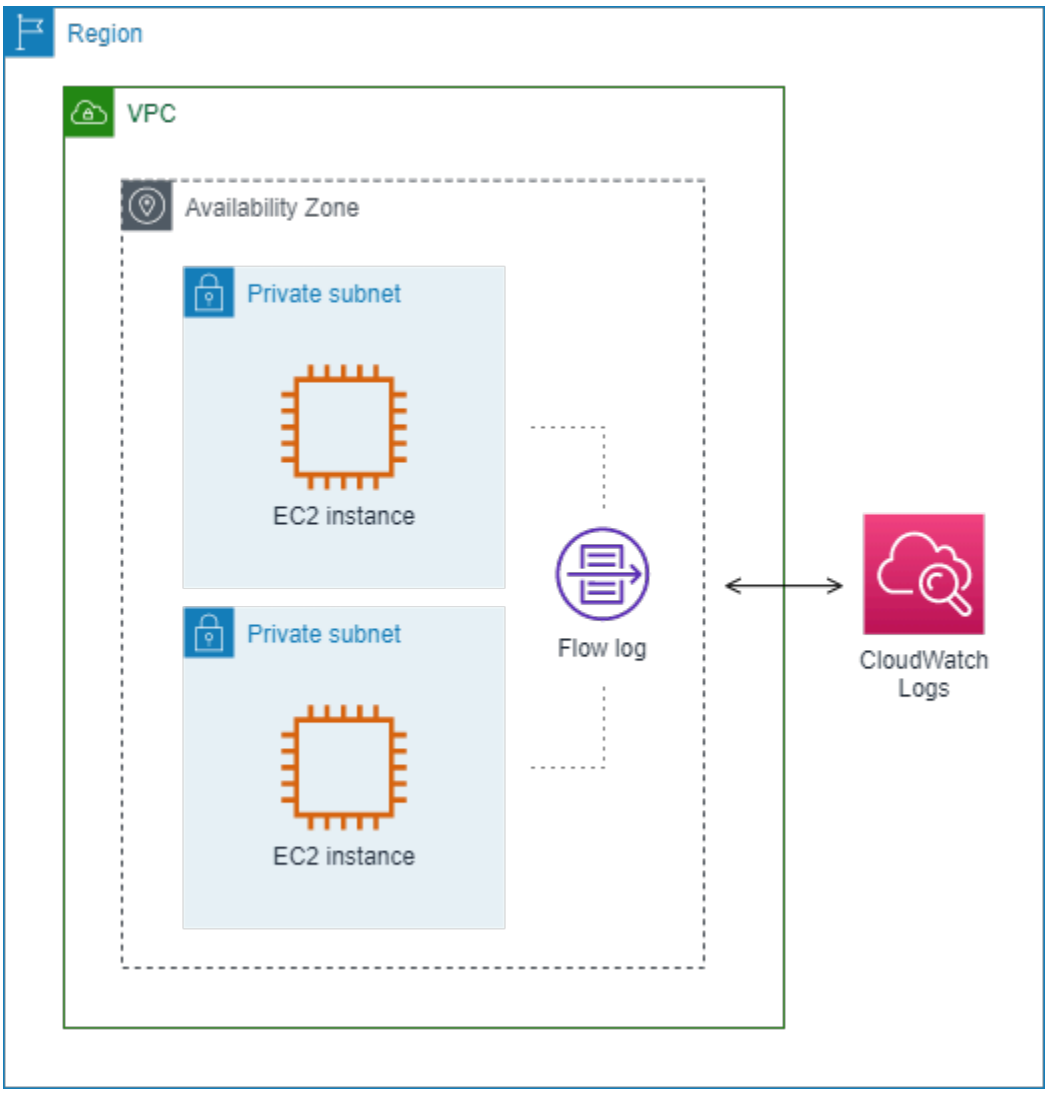
흐름 로그를 생성하려면 다음을 지정합니다.

- 흐름 로그를 생성할 리소스
- 캡처할 트래픽 유형(허용된 트래픽, 거부된 트래픽 또는 모든 트래픽)
- 흐름 로그 데이터를 게시할 대상

다음 예에서는 프라이빗 서브넷의 EC2 인스턴스 중 하나의 네트워크 인터페이스에 대해 허용된 트래픽을 캡처하고 흐름 로그 레코드를 Amazon S3 버킷에 게시하는 흐름 로그를 생성합니다.



다음 예에서 흐름 로그는 서브넷의 모든 트래픽을 캡처하고 흐름 로그 레코드를 Amazon CloudWatch Logs에 게시합니다. 흐름 로그는 서브넷의 모든 네트워크 인터페이스에 대한 트래픽을 캡처합니다.



흐름 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 내용은 2. 흐름 로그 생성 섹션을 참조하세요.

서브넷 또는 VPC에 대한 Flow Logs를 생성한 후 서브넷에서 인스턴스를 시작하면 새 네트워크 인터페이스에 대한 로그 스트림(CloudWatch Logs용) 또는 로그 파일 객체(Amazon S3용)가 생성됩니다. 로그 스트림 또는 파일 객체는 인터페이스에서 네트워크 트래픽이 발생하는 즉시 생성됩니다.

다음과 같은 다른 AWS 서비스에서 생성한 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다.

- Elastic Load Balancing
- Amazon RDS
- Amazon ElastiCache

- Amazon Redshift
- Amazon WorkSpaces
- NAT 게이트웨이
- 전송 게이트웨이

네트워크 인터페이스 유형에 관계없이 Amazon EC2 콘솔 또는 Amazon EC2 API를 사용하여 네트워크 인터페이스에 대한 흐름 로그를 작성해야 합니다.

흐름 로그에 태그를 적용할 수 있습니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 태그는 흐름 로그를 용도나 소유자별로 구성하는 데 도움이 될 수 있습니다.

흐름 로그가 더 이상 필요하지 않을 경우 삭제할 수 있습니다. 흐름 로그를 삭제하면 리소스에 대한 흐름 로그 서비스가 비활성화되어 생성되거나 게시되는 새 흐름 로그 레코드가 없습니다. 흐름 로그를 삭제해도 기존 흐름 로그 데이터는 삭제되지 않습니다. 흐름 로그를 삭제하면 작업을 마무리했을 때 대상에서 직접 흐름 로그 데이터를 삭제할 수 있습니다. 자세한 내용은 4. 흐름 로그 삭제 단원을 참조하세요.

흐름 로그 레코드

흐름 로그 레코드는 VPC에 네트워크 흐름을 나타냅니다. 기본적으로 각 레코드는 캡처 기간이라고도 하는 집계 간격 내에 발생하는 네트워크 인터넷 프로토콜(IP) 트래픽 흐름(네트워크 인터페이스별로 5튜플을 특징으로 함)을 캡처합니다.

각 레코드는 필드가 공백으로 구분되어 있는 문자열입니다. 레코드에는 소스, 대상, 프로토콜 등 IP 흐름의 다양한 구성 요소에 대한 값이 포함됩니다.

흐름 로그를 생성할 때 흐름 로그 레코드의 기본 형식을 사용하거나 사용자 지정 형식을 지정할 수 있습니다.

목차

- 집계 간격
- 기본 형식
- 사용자 지정 형식
- 사용 가능한 필드

집계 간격

집계 간격은 특정 흐름이 캡처되어 흐름 로그 레코드로 집계되는 기간입니다. 기본적으로 최대 집계 간격은 10분입니다. 흐름 로그를 만들 때 선택적으로 최대 집계 간격을 1분으로 지정할 수 있습니다. 최대 집계 간격이 1분인 흐름 로그는 최대 집계 간격이 10분인 흐름 로그보다 더 많은 양의 흐름 로그 레코드를 생성합니다.

네트워크 인터페이스가 NITRO 기반 인스턴스에 연결된 경우 집계 간격은 지정된 최대 집계 간격에 관계없이 항상 1분 이하입니다.

집계 간격 내에서 데이터를 캡처한 후에는 데이터를 처리하고 CloudWatch Logs 또는 Amazon S3에 게시하느라 추가 시간이 걸립니다. 흐름 로그 서비스는 일반적으로 로그를 약 5분 만에 CloudWatch Logs로 전송하고 약 10분 만에 Amazon S3로 전송합니다. 그러나 로그 전달에 최선의 노력을 기울이고 로그가 일반적인 전달 시간을 초과하여 지연될 수 있습니다.

기본 형식

기본 형식의 흐름 로그 레코드에는 사용 가능한 필드 테이블에 표시되는 순서대로 버전 2 필드가 포함됩니다. 기본 형식을 사용자 정의하거나 변경할 수 없습니다. 추가 필드 또는 다른 필드 하위 세트를 캡처하려면 사용자 지정 형식을 지정합니다.

사용자 지정 형식

사용자 지정 형식을 사용하면 흐름 로그 레코드에 포함되는 필드와 그 순서를 지정할 수 있습니다. 이를 통해 요구 사항에 맞는 흐름 로그를 만들고 관련이 없는 필드를 생략할 수 있습니다. 사용자 지정 형식을 사용하면 게시된 흐름 로그에서 특정 정보를 추출하기 위해 별도의 프로세스가 필요하지 않습니다. 사용 가능한 흐름 로그 필드를 얼마든지 지정할 수 있지만 하나 이상을 지정해야 합니다.

사용 가능한 필드

다음 표는 흐름 로그 레코드에 사용 가능한 모든 필드를 설명합니다. 버전(Version) 열은 해당 필드를 도입한 VPC 흐름 로그의 버전을 나타냅니다. 기본 형식에는 모든 버전 2 필드가 테이블에 표시되는 순서와 동일하게 포함됩니다.

Amazon S3 흐름 로그 데이터를 게시할 때 필드의 데이터 유형은 흐름 로그 형식에 따라 다릅니다. 형식이 일반 텍스트인 경우 모든 필드는 STRING 유형입니다. 형식이 Parquet 인 경우 필드 데이터 유형에 대한 표를 참조하세요.

필드를 적용할 수 없거나 특정 레코드에 대해 계산할 수 없는 경우 레코드는 해당 항목에 대해 '-' 기호를 표시합니다. 패킷 헤더에서 직접 제공되지 않는 메타데이터 필드는 최선의 작업 수준 근사값이며 해당 값이 누락되거나 정확하지 않을 수 있습니다.

필드	설명	버전
version	VPC 흐름 로그 버전. 기본 형식을 사용하는 경우, 버전은 2입니다. 사용자 지정 형식을 사용하는 경우, 버전은 지정된 필드 중에서 가장 높은 버전입니다. 예를 들어 버전 2의 필드만 지정한다면 버전은 2가 됩니다. 버전 2, 3 및 4의 필드를 혼합하여 지정한다면 버전은 4가 됩니다. Parquet 데이터 유형: INT_32	2
account-id	트래픽이 기록되는 소스 네트워크 인터페이스 소유자의 AWS 계정 ID입니다. 네트워크 인터페이스가 AWS 서비스에 의해 생성된 경우, 예를 들어 VPC 엔드포인트 또는 Network Load Balancer 생성 시 이 필드의 레코드에 unknown이 표시될 수도 있습니다. Parquet 데이터 유형: 문자열	2
interface-id	트래픽이 기록되는 네트워크 인터페이스 ID. 리전 NAT 게이트웨이와 연결된 흐름의 경우 '-' 기호를 반환합니다. Parquet 데이터 유형: 문자열	2
srcaddr	수신 트래픽의 경우 트래픽 소스의 IP 주소를 뜻합니다. 송신 트래픽의 경우 트래픽을 전송하는 네트워크 인터페이스의 프라이빗 IPv4 주소 또는 IPv6 주소를 뜻합니다. 리전 NAT 게이트웨이에서 나가는 트래픽의 경우 pkt-srcaddr에서와 동일한 패킷 수준 소스 IP 주소를 뜻합니다. 또한 pkt-srcaddr 단원도 참조하세요. Parquet 데이터 유형: 문자열	2
dstaddr	나가는 트래픽의 대상 주소 또는 네트워크 인터페이스의 들어오는 트래픽의 네트워크 인터페이스의 IPv4 또는 IPv6 주소. 네트워크 인터페이스의 IPv4 주소는 항상 해당 프라이빗 IPv4 주소입니다. 리전 NAT 게이트웨이로 들어오는 트래픽의 경우 pkt-dstaddr에서와 동일한 패킷 수준 대상 IP 주소를 뜻합니다. 또한 pkt-dstaddr 단원도 참조하세요. Parquet 데이터 유형: 문자열	2

필드	설명	버전
srcport	트래픽의 소스 포트 Parquet 데이터 유형: INT_32	2
dstport	트래픽의 대상 포트 Parquet 데이터 유형: INT_32	2
protocol	트래픽의 IANA 프로토콜 번호. 자세한 정보는 Assigned Internet Protocol Numbers를 참조하세요. Parquet 데이터 유형: INT_32	2
packets	흐름 중 전송된 패킷 수. Parquet 데이터 유형: INT_64	2
bytes	흐름 중 전송된 바이트 수. Parquet 데이터 유형: INT_64	2
start	흐름의 첫 번째 패킷이 집계 간격 내에서 수신된 시간(단위: Unix 초)입니다. 이 시간은 패킷이 네트워크 인터페이스에 수신되기 최대 60초 전이거나, 패킷이 전송된 후 최대 60초 후일 수 있습니다 Parquet 데이터 유형: INT_64	2
end	집계 간격 내에서 흐름의 마지막 패킷을 수신한 시간(단위: Unix 초)입니다. 이 시간은 패킷이 네트워크 인터페이스에서 전송되거나 수신된 후 최대 60초가 될 수 있습니다. Parquet 데이터 유형: INT_64	2

필드	설명	버전
action	트래픽과 연결된 작업 • ACCEPT — 트래픽이 수락되었습니다. • REJECT — 트래픽이 거부되었습니다. 예를 들어 보안 그룹이나 네트워크 ACL에서 트래픽을 허용하지 않았거나 연결이 닫힌 후 패킷이 도착했습니다. Parquet 데이터 유형: 문자열	2
log-status	흐름 로그의 로깅 상태: • OK — 선택된 대상에 정상적으로 로깅됩니다. • NODATA — 집계 간격 중 네트워크 인터페이스에서 전송하거나 수신된 네트워크 트래픽이 없었습니다. • SKIPDATA — 집계 간격 중 일부 흐름 로그 레코드를 건너뛰었습니다. 내부 용량 제한 또는 내부 오류가 원인일 수 있습니다. 집계 간격 중 일부 흐름 로그 레코드를 건너뛸 수 있습니다(사용 가능한 필드의 log-status 참조). 내부 AWS 용량 제한 또는 내부 오류가 원인일 수 있습니다. AWS Cost Explorer를 사용하여 VPC 흐름 로그 요금을 확인하고 흐름 로그 집계 간격 중에 일부 흐름 로그를 건너뛴 경우 AWS Cost Explorer에 보고된 흐름 로그 수가 Amazon VPC에서 게시한 흐름 로그 수보다 많습니다. Parquet 데이터 유형: 문자열	2
vpc-id	트래픽이 기록되는 네트워크 인터페이스를 포함하는 VPC의 ID. Parquet 데이터 유형: 문자열	3
subnet-id	트래픽이 기록되는 네트워크 인터페이스를 포함하는 서브넷의 ID. 리전 NAT 게이트웨이와 연결된 흐름의 경우 '-' 기호를 반환합니다. Parquet 데이터 유형: 문자열	3

필드	설명	버전
instance-id	인스턴스를 소유한 경우 트래픽이 기록되는 네트워크 인터페이스와 연결된 인스턴스의 ID입니다. NAT 게이트웨이의 네트워크 인터페이스 같은 요청자 관리 네트워크 인터페이스에는 '-' 기호를 반환합니다. Parquet 데이터 유형: 문자열	3
tcp-flags	다음 TCP 플래그의 비트 마스크 값: • FIN — 1 • SYN — 2 • RST — 4 • SYN-ACK — 18 지원되는 플래그가 기록되지 않는 경우 TCP 플래그 값은 0입니다. 예를 들어 tcp-flags는 ACK 또는 PSH 플래그 로깅을 지원하지 않으므로 이러한 지원되지 않는 플래그가 있는 트래픽에 대한 레코드의 결과는 tcp-flags 값 0이 됩니다. 하지만 지원되지 않는 플래그에 지원되는 플래그가 동반된 경우 지원되는 플래그의 값을 보고합니다. 예를 들어 ACK가 SYN-ACK의 일부인 경우 18을 보고합니다. 그리고 SYN+ECE와 같은 레코드가 있는 경우 SYN은 지원되는 플래그이고 ECE는 지원되지 않으므로 TCP 플래그 값은 2입니다. 어떤 이유로든 플래그 조합이 유효하지 않아 값을 계산할 수 없는 경우 값은 '-'입니다. 플래그가 전송되지 않는 경우 TCP 플래그 값은 0입니다. TCP 플래그는 집계 간격 동안 OR일 수 있습니다. 짧은 연결의 경우 SYN-ACK 및 FIN에 대해 19, SYN 및 FIN에 대해 3과 같이 흐름 로그 레코드의 동일한 행에 플래그가 설정될 수 있습니다. 문제 해결 예는 TCP 플래그 시퀀스을(를) 참조하세요. TCP 플래그에 대한 일반 정보(예: FIN, SYN 및 ACK와 같은 플래그의 의미)는 Wikipedia에서 TCP 세그먼트 구조를 참조하세요. Parquet 데이터 유형: INT_32	3

필드	설명	버전
type	트래픽 유형입니다. 가능한 값: IPv4 | IPv6 | EFA. 자세한 내용은 Elastic Fabric Adapter(EFA) 섹션을 참조하세요. Parquet 데이터 유형: 문자열	3
pkt-srcaddr	트래픽의 패킷 수준(원본) 소스 IP 주소입니다. 이 필드를 srcaddr 필드와 함께 사용하면 트래픽이 흐르는 중간 계층의 IP 주소와 트래픽의 원래 소스 IP 주소를 구별 할 수 있습니다. 예를 들어 트래픽이 NAT 게이트웨이용 네트워크 인터페이스를 통해 흐를 수 있습니다. 또 다른 경우는 Amazon EKS 파드의 IP 주소가 파드가 실행 중인 인스턴스 노드의 네트워크 인터페이스 IP 주소와 다른 경우입니다(VPC 내 통신용). Parquet 데이터 유형: 문자열	3
pkt-dstaddr	트래픽의 패킷 수준(원본) 대상 IP 주소입니다. 이 필드를 dstaddr 필드와 함께 사용하면 트래픽이 흐르는 중간 계층의 IP 주소와 트래픽의 최종 대상 IP 주소를 구별 할 수 있습니다. 예를 들어 트래픽이 NAT 게이트웨이용 네트워크 인터페이스를 통해 흐를 수 있습니다. 또 다른 경우는 Amazon EKS 파드의 IP 주소가 파드가 실행 중인 인스턴스 노드의 네트워크 인터페이스 IP 주소와 다른 경우입니다(VPC 내 통신용). Parquet 데이터 유형: 문자열	3
region	트래픽이 기록되는 네트워크 인터페이스가 포함된 리전입니다. Parquet 데이터 유형: 문자열	4
az-id	트래픽이 기록되는 네트워크 인터페이스가 포함된 가용 영역의 ID입니다. 하위 위치에서 트래픽이 발생한 경우 레코드는 이 필드에 대해 '-' 기호를 표시합니다. Parquet 데이터 유형: 문자열	4

필드	설명	버전
sublocation-type	sublocation-id 필드에 반환되는 하위 위치 유형입니다. 가능한 값: wavelength | outpost | localzone 트래픽이 하위 위치에서 발생하지 않는 경우 레코드는 이 필드에 대해 '-' 기호를 표시합니다. Parquet 데이터 유형: 문자열	4
sublocation-id	트래픽이 기록되는 네트워크 인터페이스가 포함된 하위 위치의 ID입니다. 트래픽이 하위 위치에서 발생하지 않는 경우 레코드는 이 필드에 대해 '-' 기호를 표시합니다. Parquet 데이터 유형: 문자열	4
pkt-src-aws-service	소스 IP 주소가 AWS 서비스용인 경우 pkt-srcaddr 필드에 대한 IP 주소 범위의 하위 집합 이름입니다. 소스 IP 주소가 겹친 범위에 속하는 경우 pkt-src-aws-service는 AWS 서비스 코드 중 하나만 표시합니다. 가능한 값: AMAZON | AMAZON_APPFLOW | AMAZON_CONNECT | API_GATEWAY | AURORA_DSQL | CHIME_MEETINGS | CHIME_VOICECONNECTOR | CLOUD9 | CLOUDFRONT | CLOUDFRONT_ORIGIN_FACING | CODEBUILD | DYNAMODB | EBS | EC2 | EC2_INSTANCE_CONNECT | GLOBALACCELERATOR | IVS_LOW_LATENCY | IVS_REALTIME | KINESIS_VIDEO_STREAMS | MEDIA_PACKAGE_V2 | ROUTE53 | ROUTE53_HEALTHCHECKS | ROUTE53_HEALTHCHECKS_PUBLISHING | ROUTE53_RESOLVER | S3 | WORKSPACES_GATEWAYS . Parquet 데이터 유형: 문자열	5
pkt-dst-aws-service	대상 IP 주소가 AWS 서비스용인 경우 pkt-dstaddr 필드에 대한 IP 주소 범위의 하위 집합 이름입니다. 가능한 값 목록은 pkt-src-aws-service 필드를 참조하세요. Parquet 데이터 유형: 문자열	5

필드	설명	버전
flow-direction	트래픽이 캡처되는 인터페이스에 대한 흐름 방향입니다. 가능한 값: ingress | egress Parquet 데이터 유형: 문자열	5
traffic-path	송신 트래픽이 대상으로 이동하는 경로입니다. 트래픽이 송신 트래픽인지 여부를 확인하려면 flow-direction 필드를 확인하십시오. 가능한 값은 다음과 같습니다. 적용되는 값이 없는 경우 필드는 -로 설정됩니다. • 1 — AWS 관리형 네트워크 인터페이스 또는 Outpost 로컬 게이트웨이를 포함하여 동일한 VPC의 다른 리소스를 통합니다. • 2 — 인터넷 게이트웨이 또는 게이트웨이 VPC 엔드포인트를 통해 • 3 — 가상 프라이빗 게이트웨이를 통해 • 4 — 리전 내 VPC 피어링 연결을 통해 • 5 — 리전 간 VPC 피어링 연결을 통해 • 6 - 로컬 영역 또는 Wavelength 영역을 통해 • 7 - 게이트웨이 VPC 엔드포인트를 통해 • 8 - 인터넷 게이트웨이를 통해 Parquet 데이터 유형: INT_32	5
ecs-cluster-arn	AWS트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 ECS 클러스터의 리소스 이름(ARN)입니다. 구독에 이 필드를 포함하려면 ecs:ListClusters를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-cluster-name	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 ECS 클러스터의 이름입니다. 구독에 이 필드를 포함하려면 ecs:ListClusters를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7

필드	설명	버전
ecs-container-instance-arn	트래픽이 EC2 인스턴스에서 실행 중인 ECS 태스크에서 발생하는 경우 ECS 컨테이너 인스턴스의 ARN입니다. 용량 공급자가 AWS Fargate인 경우 이 필드는 '-'가 됩니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListContainerInstances를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-container-instance-id	트래픽이 EC2 인스턴스에서 실행 중인 ECS 태스크에서 발생하는 경우 ECS 컨테이너 인스턴스의 ID입니다. 용량 공급자가 AWS Fargate인 경우 이 필드는 '-'가 됩니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListContainerInstances를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-container-id	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 컨테이너의 Docker 런타임 ID입니다. ECS 태스크에 컨테이너가 하나 이상 있는 경우 이 ID는 첫 번째 컨테이너의 Docker 런타임 ID가 됩니다. 구독에 이 필드를 포함하려면 ecs:ListClusters를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-second-container-id	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 컨테이너의 Docker 런타임 ID입니다. ECS 태스크에 컨테이너가 두 개 이상 있는 경우 이 ID는 두 번째 컨테이너의 Docker 런타임 ID가 됩니다. 구독에 이 필드를 포함하려면 ecs:ListClusters를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-service-name	트래픽이 실행 중인 ECS 태스크에서 발생하고 ECS 태스크가 ECS 서비스에 의해 시작되는 경우 ECS 서비스의 이름입니다. ECS 태스크가 ECS 서비스에 의해 시작되지 않는 경우 이 필드는 '-'가 됩니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListServices를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7

필드	설명	버전
ecs-task-definition-arn	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 ECS 태스크 정의의 ARN입니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListTaskDefinitions를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-task-arn	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 ECS 태스크의 ARN입니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListTasks를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
ecs-task-id	트래픽이 실행 중인 ECS 태스크에서 발생하는 경우 ECS 태스크의 ID입니다. 구독에 이 필드를 포함하려면 ecs:ListClusters와 ecs:ListTasks를 직접적으로 호출할 수 있는 권한이 필요합니다. Parquet 데이터 유형: 문자열	7
reject-reason	트래픽이 거부된 이유입니다. 가능한 값은 BPA, EC입니다. 다른 거부 이유는 '-'를 반환합니다. • BPA - VPC 퍼블릭 액세스 차단(BPA)에 대한 자세한 내용은 VPC 및 서브넷에 대한 퍼블릭 액세스 차단 섹션을 참조하세요. • EC - 암호화 제어로 인해 VPC 엔드포인트에서 흐름이 거부되었습니다. VPC 암호화 제어에 대한 자세한 내용은 전송 중 VPC 암호화 적용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	8
resource-id	트래픽이 기록되는 네트워크 인터페이스를 포함하는 리전 NAT 게이트웨이의 ID입니다. 리전 NAT 게이트웨이와 연결되지 않은 트래픽 흐름의 경우 '-' 기호를 반환합니다. 리전 NAT 게이트웨이에 대한 자세한 내용은 자동 다중 AZ 확장을 위한 리전 NAT 게이트웨이 섹션을 참조하세요. Parquet 데이터 유형: 문자열	9

필드	설명	버전

encryption-status	흐름의 암호화 상태입니다. VPC 암호화 제어에 대한 자세한 내용은 전송 중 VPC 암호화 적용 섹션을 참조하세요. 가능한 값은 다음과 같습니다.
	• 0 - 암호화되지 않음.
	• 1 - Nitro로 암호화됨. Nitro System 하드웨어로 암호화됩니다.
	• 2 - 애플리케이션으로 암호화됨. 다음의 사항만 애플리케이션 암호화로 간주됩니다.
	  • AWS 서비스에 대한 인터페이스 엔드포인트의 TCP 포트 443에서의 흐름*
	  • 게이트웨이 엔드포인트의 TCP 포트 443에서의 흐름*
	  • VPC 엔드포인트를 통해 암호화된 Amazon Redshift 클러스터로 이동하는 트래픽**
	• 3 - Nitro 암호화 및 애플리케이션 암호화됨.
	
	VPC 암호화 제어가 활성화되지 않았거나 FlowLog가 상태를 가져올 수 없는 경우 값은 '-'입니다.
	
	Note
	* 인터페이스 및 게이트웨이 엔드포인트의 경우 AWS는 암호화 상태를 확인하기 위해 패킷 데이터를 살펴보지 않고, 대신 사용된 포트를 기반으로 암호화 상태를 추정합니다.
	** 지정된 AWS 관리형 엔드포인트의 경우 AWS는 서비스 구성의 TLS 요구 사항에 따라 암호화 상태를 확인합니다.
	
	Parquet 데이터 유형: INT_32
		10

필드	설명	버전
instance-tag	TagFieldSpecifications에 포함된 첫 번째 instance 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 ec2:DescribeTags와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11
instance-tag-2	TagFieldSpecifications에 포함된 두 번째 instance 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 ec2:DescribeTags와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11
interface-tag	TagFieldSpecifications에 포함된 첫 번째 network-interface 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 ec2:DescribeTags와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11

필드	설명	버전
interface-tag-2	TagFieldSpecifications에 포함된 두 번째 network-interface 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 ec2:DescribeTags와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11
asg-tag	TagFieldSpecifications에 포함된 첫 번째 auto-scaling-group 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 autoscaling:DescribeTag와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. Auto Scaling 그룹 태그 값을 실시간으로 올바르게 추적하고 업데이트하려면 계정에 활성화된 CloudTrail 추적이 하나 이상 있어야 합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11
asg-tag-2	TagFieldSpecifications에 포함된 두 번째 auto-scaling-group 태그의 태그 값입니다. 태그 값은 모든 특수 문자를 UTF-8 퍼센트 인코딩으로 변환하여 표시됩니다. 구독에 이 필드를 포함하려면 autoscaling:DescribeTag와 iam:CreateServiceLinkedRole을 직접적으로 호출할 수 있는 권한이 필요합니다. Auto Scaling 그룹 태그 값을 실시간으로 올바르게 추적하고 업데이트하려면 계정에 활성화된 CloudTrail 추적이 하나 이상 있어야 합니다. 흐름 로그 Amazon EC2 태그 서비스 연결 역할에 대한 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요. Parquet 데이터 유형: 문자열	11

필드	설명	버전
interface-type	흐름을 캡처하는 로컬 네트워크 인터페이스의 유형입니다. 지원되는 값은 `nat_gateway` | `network_load_balancer` | `regional_nat_gateway` | `transit_gateway` | `vpc_endpoint` 입니다. 네트워크 인터페이스에 연결된 서비스가 없거나 해당 서비스가 지원되지 않는 경우 값은 '-'입니다. Parquet 데이터 유형: 문자열	11
next-hop-interface-id	다음 홉 네트워크 인터페이스의 ID입니다. 송신 트래픽의 경우 트래픽을 수신하는 인터페이스입니다. 수신 트래픽의 경우 트래픽을 송신하는 인터페이스입니다. Parquet 데이터 유형: 문자열	11
next-hop-subnet-id	다음 홉 네트워크 인터페이스가 포함된 서브넷의 ID입니다. Parquet 데이터 유형: 문자열	11
next-hop-az-id	다음 홉 네트워크 인터페이스가 포함된 가용 영역의 ID입니다. Parquet 데이터 유형: 문자열	11
next-hop-vpc-id	다음 홉 네트워크 인터페이스가 포함된 VPC의 ID입니다. Parquet 데이터 유형: 문자열	11
next-hop-interface-type	다음 홉 네트워크 인터페이스의 유형입니다. 가능한 값 목록은 interface-type 필드를 참조하세요. Parquet 데이터 유형: 문자열	11

흐름 로그 레코드의 예시

다음은 특정 트래픽 흐름을 캡처하는 흐름 로그 레코드의 예시입니다.

흐름 로그 레코드 형식에 대한 자세한 내용은 흐름 로그 레코드를 참조하세요. 흐름 로그를 생성하는 방법에 대한 자세한 내용은 흐름 로그 작업 단원을 참조하세요.

목차

- 허용 및 거부된 트래픽
- 데이터가 없고 건너뛴 레코드
- 보안 그룹 및 네트워크 ACL 규칙
- IPv6 트래픽
- TCP 플래그 시퀀스
- 영역 NAT 게이트웨이를 통한 트래픽
- 리전 NAT 게이트웨이를 통한 트래픽
- 전송 게이트웨이를 통한 트래픽
- 서비스 이름, 트래픽 경로 및 흐름 방향

허용 및 거부된 트래픽

다음은 기본 흐름 로그 레코드의 예시입니다.

이 예시에서는 프라이빗 IP 주소가 172.31.16.21이고 계정 123456789010의 ID가 eni-1235b8ca123456789인 네트워크 인터페이스로 IP 주소 172.31.16.139의 SSH 트래픽(대상 포트 22, TCP 프로토콜)이 허용되었습니다.

```
2 123456789010 eni-1235b8ca123456789 172.31.16.139 172.31.16.21 20641 22 6 20 4249
 1418530010 1418530070 ACCEPT OK
```

이 예시에서는 계정 123456789010에서 네트워크 인터페이스 eni-1235b8ca123456789에 대한 RDP 트래픽(대상 포트 3389, TCP 프로토콜)이 거부되었습니다.

```
2 123456789010 eni-1235b8ca123456789 172.31.9.69 172.31.9.12 49761 3389 6 20 4249
 1418530010 1418530070 REJECT OK
```

데이터가 없고 건너뛴 레코드

다음은 기본 흐름 로그 레코드의 예입니다.

이 예에서는 집계 간격 동안 데이터가 기록되지 않았습니다.

```
2 123456789010 eni-1235b8ca123456789 - - - - - - - 1431280876 1431280934 - NODATA
```

VPC 흐름 로그는 내부 용량 초과로 집계 간격 동안 흐름 로그 데이터를 캡처할 수 없는 경우 레코드를 건너뜁니다. 건너뛴 단일 레코드는 집계 간격 중 네트워크 인터페이스에 대해 캡처되지 않은 여러 흐름을 나타낼 수 있습니다.

```
2 123456789010 eni-11111111aaaaaaaaa - - - - - - - 1431280876 1431280934 - SKIPDATA
```

> **Note**
>
> 집계 간격 중 일부 흐름 로그 레코드를 건너뛸 수 있습니다(사용 가능한 필드의 log-status 참조). 내부 AWS 용량 제한 또는 내부 오류가 원인일 수 있습니다. AWS Cost Explorer를 사용하여 VPC 흐름 로그 요금을 확인하고 흐름 로그 집계 간격 중에 일부 흐름 로그를 건너뛴 경우 AWS Cost Explorer에 보고된 흐름 로그 수가 Amazon VPC에서 게시한 흐름 로그 수보다 많습니다.

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. 보안 그룹의 규칙에서 허용하지 않더라도 허용된 트래픽에 응답할 수 있다는 뜻입니다. 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. 보안 그룹의 인바운드 규칙은 ICMP 트래픽을 허용하지만 아웃바운드 규칙은 ICMP 트래픽을 허용하지 않습니다. 보안 그룹은 상태 저장이므로 인스턴스의 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 기본 흐름 로그에서 다음과 같은 2가지 흐름 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-1235b8ca123456789 203.0.113.12 172.31.16.139 0 0 1 4 336 1432917027
 1432917142 ACCEPT OK
```

```
2 123456789010 eni-1235b8ca123456789 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 트래픽

다음은 기본 흐름 로그 레코드의 예입니다. 이 예시에서는 계정 123456789010에서, IPv6 주소 2001:db8:1234:a100:8d6e:3477:df66:f105로부터 네트워크 인터페이스 eni-1235b8ca123456789로의 SSH 트래픽(포트 22)이 허용되었습니다.

```
2 123456789010 eni-1235b8ca123456789 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
```

TCP 플래그 시퀀스

이 섹션에는 아래와 같은 순서로 다음 필드를 캡처하는 사용자 지정 흐름 로그의 예시가 포함되어 있습니다.

```
version vpc-id subnet-id instance-id interface-id account-id type srcaddr dstaddr
 srcport dstport pkt-srcaddr pkt-dstaddr protocol bytes packets start end action tcp-
flags log-status
```

이 섹션에서 예시의 tcp-flags 필드는 흐름 로그의 마지막 두 번째 값으로 표시됩니다. TCP 플래그는 트래픽의 방향(예: 연결을 시작한 서버)을 식별하는 데 도움이 됩니다.

> **Note**
>
> tcp-flags 옵션에 대한 자세한 내용 및 각 TCP 플래그에 대한 설명은 사용 가능한 필드의 내용을 참조하세요.

다음 레코드(오후 7:47:55 오후에 시작하고 오후 7:48:53에 끝남)에서는 클라이언트가 포트 5001에서 실행 중인 서버에 대한 두 개의 연결을 시작했습니다. 클라이언트의 다른 소스 포트(43416 및 43418)

에서 서버가 두 개의 SYN 플래그(2)를 수신했습니다. 각 SYN에 대해 SYN-ACK가 서버에서 해당 포트의 클라이언트(18)로 전송되었습니다.

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 52.213.180.42 10.0.0.62 43416 5001
 52.213.180.42 10.0.0.62 6 568 8 1566848875 1566848933 ACCEPT 2 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43416 10.0.0.62
 52.213.180.42 6 376 7 1566848875 1566848933 ACCEPT 18 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 52.213.180.42 10.0.0.62 43418 5001
 52.213.180.42 10.0.0.62 6 100701 70 1566848875 1566848933 ACCEPT 2 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43418 10.0.0.62
 52.213.180.42 6 632 12 1566848875 1566848933 ACCEPT 18 OK
```

두 번째 집계 간격에서 이전 흐름 중에 설정된 연결 중 하나가 닫힙니다. 서버는 포트 43418 연결을 위해 FIN 플래그(1)를 클라이언트로 보냈습니다. 클라이언트가 43418 포트에서 서버에 FIN으로 응답했습니다.

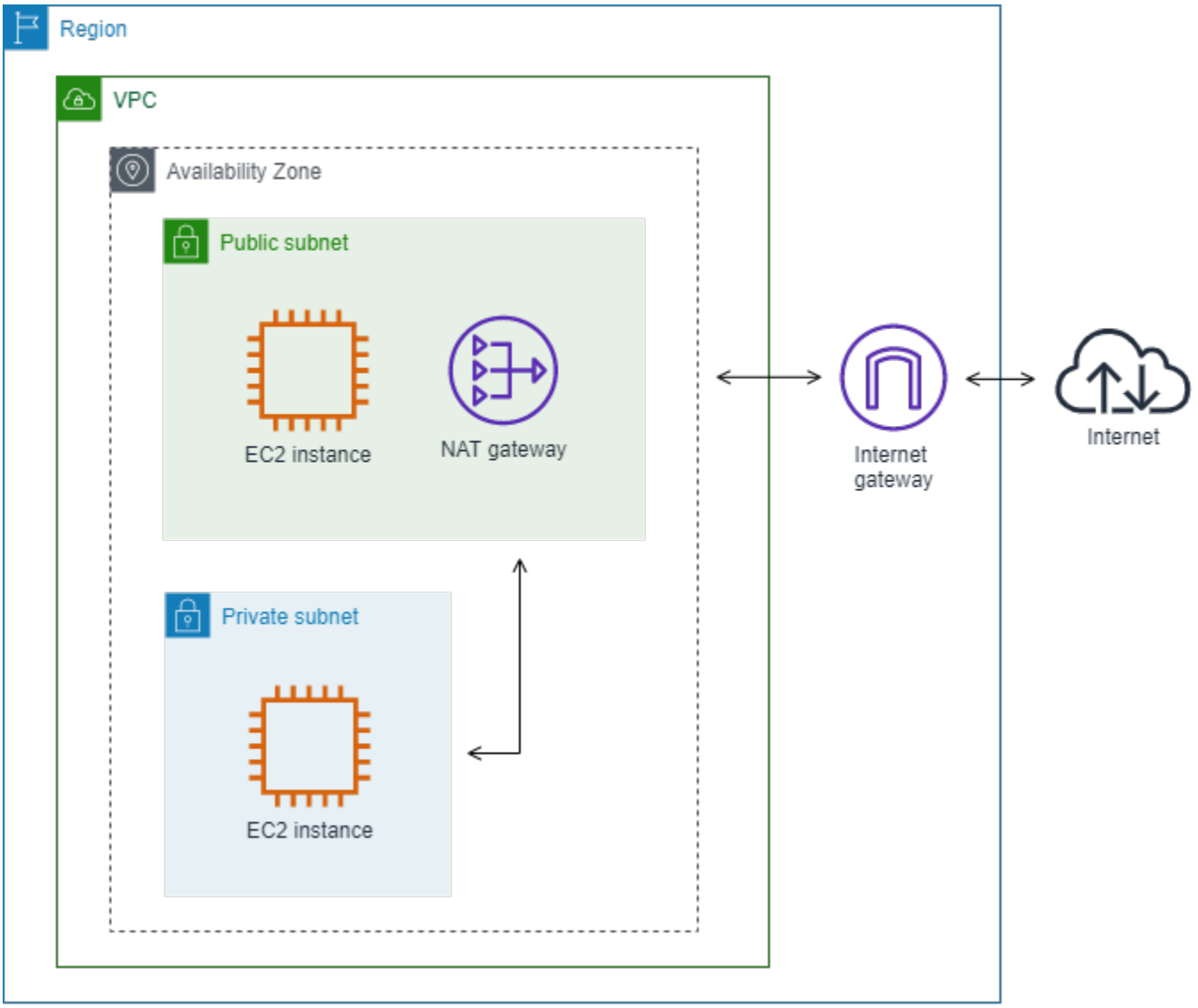
```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43418 10.0.0.62
 52.213.180.42 6 63388 1219 1566848933 1566849113 ACCEPT 1 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 52.213.180.42 10.0.0.62 43418 5001
 52.213.180.42 10.0.0.62 6 23294588 15774 1566848933 1566849113 ACCEPT 1 OK
```

단일 집계 간격 내에서 열리고 닫히는 짧은 연결의 경우 동일한 방향의 트래픽 흐름에 대해 Flow Logs 레코드의 동일한 줄에 플래그가 설정될 수 있습니다. 다음 예시에서는 동일한 집계 간격 내에서 연결이 설정되고 완료됩니다. 첫 번째 줄에서 TCP 플래그 값은 3입니다. 이는 클라이언트에서 서버로 전송된 SYN 및 FIN 메시지가 있음을 나타냅니다. 두 번째 줄에서 TCP 플래그 값은 19입니다. 이는 서버에서 클라이언트로 전송된 SYN-ACK 및 FIN 메시지가 있음을 나타냅니다.

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 52.213.180.42 10.0.0.62 43638 5001
 52.213.180.42 10.0.0.62 6 1260 17 1566933133 1566933193 ACCEPT 3 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43638  10.0.0.62
 52.213.180.42 6 967 14 1566933133 1566933193 ACCEPT 19 OK
```

영역 NAT 게이트웨이를 통한 트래픽

이 예시에서 프라이빗 서브넷의 인스턴스는 퍼블릭 서브넷에 있는 영역 NAT 게이트웨이를 통해 인터넷에 액세스합니다.



영역 NAT 게이트웨이 네트워크 인터페이스에 대한 다음 사용자 지정 흐름 로그는 다음 필드를 다음 순서로 캡처합니다.

```
instance-id interface-id srcaddr dstaddr pkt-srcaddr pkt-dstaddr
```

흐름 로그는 영역 NAT 게이트웨이 네트워크 인터페이스를 통해 인스턴스 IP 주소(10.0.1.5)에서 인터넷의 호스트(203.0.113.5)로의 트래픽 흐름을 보여줍니다. 영역 NAT 게이트웨이 네트워크 인터페이스는 요청자 관리형 네트워크 인터페이스이므로, 흐름 로그 레코드는 instance-id 필드에 '-' 기호를 표시합니다. 다음 줄은 소스 인스턴스에서 영역 NAT 게이트웨이 네트워크 인터페이스로의 트래픽을 보여

줍니다. dstaddr 및 pkt-dstaddr 필드의 값은 다릅니다. dstaddr 필드에는 영역 NAT 게이트웨이 네트워크 인터페이스의 프라이빗 IP 주소가 표시되고, pkt-dstaddr 필드에는 인터넷에 있는 호스트의 최종 대상 IP 주소가 표시됩니다.

```
- eni-1235b8ca123456789 10.0.1.5 10.0.0.220 10.0.1.5 203.0.113.5
```

다음 두 줄은 영역 NAT 게이트웨이 네트워크 인터페이스에서 인터넷의 대상 호스트로의 트래픽과 호스트에서 NAT 게이트웨이 네트워크 인터페이스로의 응답 트래픽을 보여줍니다.

```
- eni-1235b8ca123456789 10.0.0.220 203.0.113.5 10.0.0.220 203.0.113.5
- eni-1235b8ca123456789 203.0.113.5 10.0.0.220 203.0.113.5 10.0.0.220
```

다음 줄은 영역 NAT 게이트웨이 네트워크 인터페이스에서 소스 인스턴스로의 응답 트래픽을 보여줍니다. srcaddr 및 pkt-srcaddr 필드의 값은 다릅니다. srcaddr 필드에는 영역 NAT 게이트웨이 네트워크 인터페이스의 프라이빗 IP 주소가 표시되고, pkt-srcaddr 필드에는 인터넷에 있는 호스트의 IP 주소가 표시됩니다.

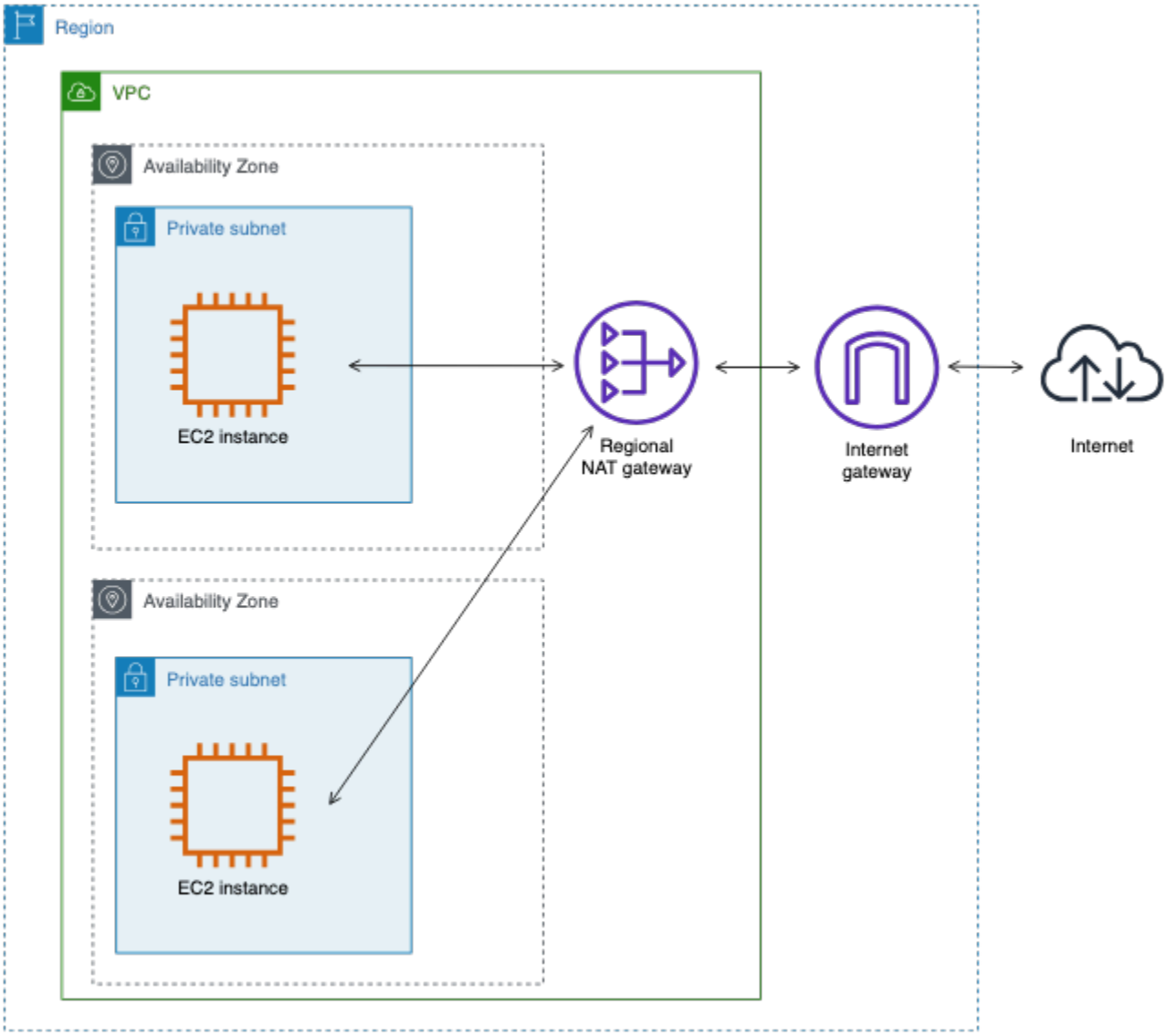
```
- eni-1235b8ca123456789 10.0.0.220 10.0.1.5 203.0.113.5 10.0.1.5
```

위와 동일한 필드 세트를 사용하여 다른 사용자 정의 흐름 로그를 작성합니다. 프라이빗 서브넷에서 인스턴스의 네트워크 인터페이스에 대한 흐름 로그를 생성합니다. 이 경우 instance-id 필드는 네트워크 인터페이스와 연결된 인스턴스의 ID를 반환합니다. dstaddr 필드 및 pkt-dstaddr 필드와 srcaddr 필드 및 pkt-srcaddr 필드 간에는 차이가 없습니다. 영역 NAT 게이트웨이의 네트워크 인터페이스와 달리, 이 네트워크 인터페이스는 트래픽의 중간 네트워크 인터페이스가 아닙니다.

```
i-01234567890123456 eni-1111aaaa2222bbbb3 10.0.1.5 203.0.113.5 10.0.1.5 203.0.113.5
 #Traffic from the source instance to host on the internet
i-01234567890123456 eni-1111aaaa2222bbbb3 203.0.113.5 10.0.1.5 203.0.113.5 10.0.1.5
 #Response traffic from host on the internet to the source instance
```

리전 NAT 게이트웨이를 통한 트래픽

리전 NAT 게이트웨이는 서로 다른 가용 영역에 있는 여러 서브넷에 연결할 수 있습니다. 이 예시에서 서로 다른 두 가용 영역의 프라이빗 서브넷에 있는 두 인스턴스는 동일한 리전 NAT 게이트웨이를 통해 인터넷에 액세스합니다. 다음 흐름 로그는 리전 NAT 게이트웨이를 통해 인스턴스 중 하나에서 인터넷으로의 트래픽을 보여줍니다.



리전 NAT 게이트웨이에 대한 다음 사용자 지정 흐름 로그는 다음 필드를 다음 순서로 캡처합니다.

```
resource-id instance-id interface-id subnet-id srcaddr dstaddr pkt-srcaddr pkt-dstaddr
```

Flow Logs는 인스턴스 IP 주소(10.0.1.5)에서 리전 NAT 게이트웨이를 거쳐 인터넷 호스트(203.0.113.5)로 향하는 트래픽 흐름을 보여줍니다. instance-id, interface-id, subnet-id은 리전 NAT 게이트웨이에 적용되지 않습니다. 따라서 흐름 로그 레코드는 해당 필드에 '-' 기호를 표시합니다. 대신 resource-id 필드에는 리전 NAT 게이트웨이의 ID가 표시됩니다. dstaddr 및 pkt-dstaddr 필드에는 인터넷에 있는 호스트의 최종 대상 IP 주소가 표시됩니다.

```
nat-1234567890abcdef - - - 10.0.1.5 203.0.113.5 10.0.1.5 203.0.113.5
```

다음 두 행은 리전 NAT 게이트웨이(퍼블릭 IP 주소: 107.22.182.139)에서 인터넷의 대상 호스트로 향하는 트래픽을 나타냅니다. 또한 호스트에서 리전 NAT 게이트웨이로 향하는 응답 트래픽도 표시됩니다.

```
nat-1234567890abcdef - - - 107.22.182.139 203.0.113.5 107.22.182.139 203.0.113.5
nat-1234567890abcdef - - - 203.0.113.5 107.22.182.139 203.0.113.5 107.22.182.139
```

다음 줄은 리전 NAT 게이트웨이에서 소스 인스턴스로의 응답 트래픽을 보여줍니다. srcaddr 및 pkt-srcaddr 필드에는 인터넷에 있는 호스트의 IP 주소가 표시됩니다.

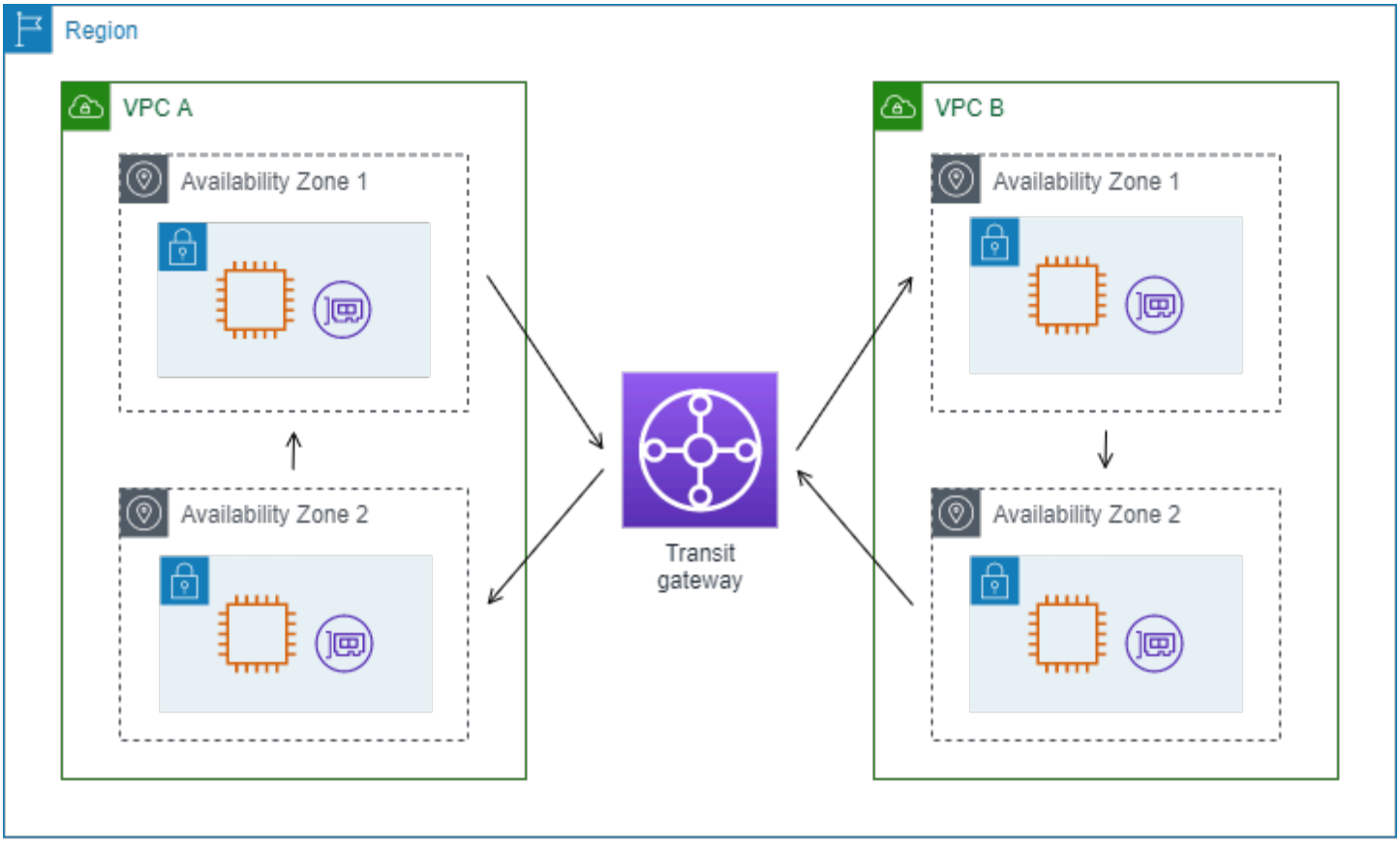
```
nat-1234567890abcdef - - - 203.0.113.5 10.0.1.5 203.0.113.5 10.0.1.5
```

위와 동일한 필드 세트를 사용하여 다른 사용자 정의 흐름 로그를 작성합니다. 프라이빗 서브넷에서 인스턴스의 네트워크 인터페이스에 대한 흐름 로그를 생성합니다. 이 경우 instance-id 필드는 네트워크 인터페이스와 연결된 인스턴스의 ID를 반환하며, resource-id는 '-'입니다. dstaddr 및 pkt-dstaddr 필드와 srcaddr 및 pkt-srcaddr 필드 간에는 차이가 없습니다.

```
- i-01234567890123456 eni-1111aaaa2222bbbb3 subnet-aaaaaaaa012345678 10.0.1.5
 203.0.113.5 10.0.1.5 203.0.113.5 #Traffic from the source instance to host on the
 internet
- i-01234567890123456 eni-1111aaaa2222bbbb3 subnet-aaaaaaaa012345678 203.0.113.5
 10.0.1.5 203.0.113.5 10.0.1.5 #Response traffic from host on the internet to the
 source instance
```

전송 게이트웨이를 통한 트래픽

이 예에서 VPC A의 클라이언트는 전송 게이트웨이를 통해 VPC B의 웹 서버에 연결합니다. 클라이언트와 서버가 서로 다른 가용 영역에 있습니다. 트래픽은 탄력적 네트워크 인터페이스 ID 하나(이 예에서 ID가 eni-11111111111111111임)를 사용하여 VPC B의 서버에 도착하고 다른 ID(예: eni-22222222222222222)를 사용하여 VPC B를 떠납니다.



다음 형식으로 VPC B에 대한 사용자 지정 흐름 로그를 생성합니다.

```
version interface-id account-id vpc-id subnet-id instance-id srcaddr dstaddr srcport
 dstport protocol tcp-flags type pkt-srcaddr pkt-dstaddr action log-status
```

흐름 로그 레코드의 다음 줄은 웹 서버의 네트워크 인터페이스에서의 트래픽 흐름을 보여줍니다. 첫 번째 줄은 클라이언트의 요청 트래픽이고 마지막 줄은 웹 서버의 응답 트래픽입니다.

```
3 eni-33333333333333333 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb
 i-01234567890123456 10.20.33.164 10.40.2.236 39812 80 6 3 IPv4 10.20.33.164
 10.40.2.236 ACCEPT OK
...
3 eni-33333333333333333 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb
 i-01234567890123456 10.40.2.236 10.20.33.164 80 39812 6 19 IPv4 10.40.2.236
 10.20.33.164 ACCEPT OK
```

다음 줄은 subnet-11111111aaaaaaaaa 서브넷의 전송 게이트웨이에 대한 요청자 관리 네트워크 인터페이스인 eni-1111111111111111에서의 요청 트래픽입니다. 흐름 로그 레코드는 instance-id 필드에

'-' 기호를 표시합니다. srcaddr 필드에는 전송 게이트웨이 네트워크 인터페이스의 프라이빗 IP 주소가 표시되고, pkt-srcaddr 필드에는 VPC A의 클라이언트의 원본 IP 주소가 표시됩니다.

```
3 eni-1111111111111111 123456789010 vpc-abcdefab012345678 subnet-11111111aaaaaaaaa -
 10.40.1.175 10.40.2.236 39812 80 6 3 IPv4 10.20.33.164 10.40.2.236 ACCEPT OK
```

다음 줄은 subnet-22222222bbbbbbbbb 서브넷의 전송 게이트웨이에 대한 요청자 관리 네트워크 인터페이스인 eni-22222222222222222에서의 응답 트래픽입니다. dstaddr 필드에는 전송 게이트웨이 네트워크 인터페이스의 프라이빗 IP 주소가 표시되고, pkt-dstaddr 필드에는 VPC A의 클라이언트의 IP 주소가 표시됩니다.

```
3 eni-22222222222222222 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb -
 10.40.2.236 10.40.2.31 80 39812 6 19 IPv4 10.40.2.236 10.20.33.164 ACCEPT OK
```

서비스 이름, 트래픽 경로 및 흐름 방향

다음은 사용자 지정 흐름 로그 레코드의 필드 예시입니다.

```
version srcaddr dstaddr srcport dstport protocol start end type packets bytes account-
id vpc-id subnet-id instance-id interface-id region az-id sublocation-type sublocation-
id action tcp-flags pkt-srcaddr pkt-dstaddr pkt-src-aws-service pkt-dst-aws-service
 traffic-path flow-direction log-status
```

다음 예시에서는 레코드에 버전 5 필드가 포함되어 있으므로 버전이 5입니다. EC2 인스턴스가 Amazon S3 서비스를 호출합니다. 흐름 로그는 인스턴스의 네트워크 인터페이스에서 캡처됩니다. 첫 번째 레코드의 흐름 방향은 ingress이고 두 번째 레코드의 흐름 방향은 egress입니다. egress 레코드의 traffic-path는 8로, 트래픽이 인터넷 게이트웨이를 통해 전송된다는 것을 나타냅니다. traffic-path 트래픽에 대해서는 ingress 필드가 지원되지 않습니다. pkt-srcaddr 또는 pkt-dstaddr이 퍼블릭 IP 주소인 경우 서비스 이름이 표시됩니다.

```
5 52.95.128.179 10.0.0.71 80 34210 6 1616729292 1616729349 IPv4 14 15044
 123456789012 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-0c50d5961bcb2d47b
 eni-1235b8ca123456789 ap-southeast-2 apse2-az3 - - ACCEPT 19 52.95.128.179 10.0.0.71
 S3 - - ingress OK
5 10.0.0.71 52.95.128.179 34210 80 6 1616729292 1616729349 IPv4 7 471 123456789012 vpc-
abcdefab012345678 subnet-aaaaaaaa012345678 i-0c50d5961bcb2d47b eni-1235b8ca123456789
 ap-southeast-2 apse2-az3 - - ACCEPT 3 10.0.0.71 52.95.128.179 - S3 8 egress OK
```

흐름 로그 제한

흐름 로그를 사용하려면 다음과 같은 제한 사항을 알아 두어야 합니다.

- 흐름 로그를 생성한 후에는 선택한 네트워크 인터페이스, 서브넷 또는 VPC에 대한 활성 트래픽이 있을 때까지 흐름 로그 데이터가 표시되지 않습니다.
- 피어 VPC가 본인의 계정이 아닌 한, 본인의 VPC와 피어링된 VPC에 대해 흐름 로그를 활성화할 수 없습니다.
- 흐름 로그를 생성한 후에는 구성 또는 흐름 로그 레코드 형식을 변경할 수 없습니다. 예를 들어 다른 IAM 역할을 흐름 로그와 연결하거나 흐름 로그 레코드에서 필드를 추가 또는 제거할 수 없습니다. 대신에 흐름 로그를 삭제한 후 필요한 구성으로 새로운 흐름 로그를 생성할 수 있습니다.
- 네트워크 인터페이스에 IPv4 주소가 여러 개 있고 트래픽이 보조 프라이빗 IPv4 주소로 전송되는 경우, 흐름 로그는 `dstaddr` 필드에 주 프라이빗 IPv4 주소를 표시합니다. 원래 대상 IP 주소를 캡처하려면 `pkt-dstaddr` 필드로 흐름 로그를 작성하세요.
- 트래픽이 네트워크 인터페이스로 전송된 경우 대상이 네트워크 인터페이스의 IP 주소가 아니면 흐름 로그에 `dstaddr` 필드의 기본 프라이빗 IPv4 주소가 표시됩니다. 원래 대상 IP 주소를 캡처하려면 `pkt-dstaddr` 필드로 흐름 로그를 작성하세요.
- 네트워크 인터페이스에서 트래픽이 전송되고 소스 IP가 해당 네트워크 인터페이스의 IP 주소 중 하나가 아닌 경우, 동작은 흐름 방향에 따라 달라집니다. 아웃바운드 흐름의 경우 Flow Logs는 `srcaddr` 필드에 기본 프라이빗 IPv4 주소를 표시합니다. 원래 소스 IP 주소를 캡처하려면 `pkt-srcaddr` 필드로 흐름 로그를 작성하세요.

  로그 레코드가 네트워크 인터페이스로의 수신 흐름 관련인 경우 `srcaddr` 필드에 네트워크 인터페이스의 기본 프라이빗 IP가 표시되지 않습니다.
- 네트워크 인터페이스가 NITRO 기반 인스턴스에 연결된 경우 집계 간격은 지정된 최대 집계 간격에 관계없이 항상 1분 이하입니다.
- `pkt-srcaddr` 및 `pkt-dstaddr` 필드의 경우 중간 계층에 클라이언트 IP 주소 보존이 활성화된 경우 이 필드에는 중간 계층의 IP 주소 대신 보존된 클라이언트 IP가 표시될 수 있습니다.
- `traffic-path` 필드의 경우, 값은 동일한 VPC의 리소스를 통과하는 흐름과 Outpost 로컬 게이트웨이를 통과하는 흐름에 대해 동일합니다.
- 집계 간격 중 일부 흐름 로그 레코드를 건너뛸 수 있습니다(사용 가능한 필드의 log-status 참조). 내부 AWS 용량 제한 또는 내부 오류가 원인일 수 있습니다. AWS Cost Explorer를 사용하여 VPC 흐름 로그 요금을 확인하고 흐름 로그 집계 간격 중에 일부 흐름 로그를 건너뛴 경우 AWS Cost Explorer에 보고된 흐름 로그 수가 Amazon VPC에서 게시한 흐름 로그 수보다 많습니다.
- VPC 퍼블릭 액세스 차단(BPA)을 사용하는 경우:

  - VPC BPA의 흐름 로그에는 건너뛴 레코드가 포함되지 않습니다.
  - 흐름 로그에 bytes 필드를 포함하더라도 VPC BPA의 흐름 로그에는 bytes가 포함되지 않습니다.
- VPC 흐름 로그는 계정별로 리소스당 최대 250개의 구독을 지원합니다. 이 한도에 도달한 리소스에 대해 추가 구독을 생성하려면 먼저 기존 구독을 삭제해야 합니다.

흐름 로그는 모든 IP 트래픽을 캡처하지는 않습니다. 다음 트래픽 유형은 기록되지 않습니다.

- 인스턴스가 Amazon DNS 서버에 연결할 때 생성한 트래픽. 고유 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 169.254.169.123와 주고받는 트래픽.
- DHCP 트래픽.
- 트래픽 미러링 소스 트래픽. 트래픽 미러링 대상 트래픽만 볼 수 있습니다.
- 기본 VPC 라우터의 예약된 IP 주소로 보내는 트래픽.
- 엔드포인트 네트워크 인터페이스와 Network Load Balancer 네트워크 인터페이스 간의 트래픽.
- 주소 확인 프로토콜(ARP) 트래픽.
- 생성 후 몇 분 내에 삭제되는 수명이 짧은 리전 NAT 게이트웨이의 트래픽.

버전 7에서 사용할 수 있는 ECS 필드에만 적용되는 제한 사항:

- 흐름 로그 구독의 소유자가 기본 ECS 태스크를 소유하지 않은 경우 ECS 필드는 계산되지 않습니다.

  예를 들어 서브넷(SubnetA)을 다른 계정(AccountB)과 공유한 후, SubnetA에 대한 Flow Logs 구독을 생성한다고 가정해 보겠습니다. AccountB이 공유 서브넷에서 ECS 태스크를 시작하면 구독에 해당 태스크의 트래픽 로그가 수신됩니다. 그러나 이러한 로그의 ECS 필드는 보안상의 이유로 계산되지 않습니다.
- VPC/서브넷 리소스 수준에서 ECS 필드를 사용하여 흐름 로그 구독을 생성하는 경우 비ECS 네트워크 인터페이스에 대해 생성된 모든 트래픽도 구독에 전달됩니다. 비ECS IP 트래픽의 경우 ECS 필드 값은 '-'가 됩니다. 예를 들어 서브넷(subnet-000000)이 있고 ECS 필드(fl-00000000)로 이 서브넷에 대한 흐름 로그 구독을 생성합니다. subnet-000000에서 인터넷에 연결되어 있고 IP 트래픽을 활발하게 생성하는 EC2 인스턴스(i-0000000)를 시작합니다. 또한 동일한 서브넷에서 실행 중인 ECS 태스크(ECS-Task-1)를 시작합니다. i-0000000과 ECS-Task-1 모두 IP 트래픽을

생성하므로 흐름 로그 구독 `fl-00000000`은 두 엔터티 모두에 대한 트래픽 로그를 제공합니다. 그러나 `ECS-Task-1`만이 logFormat에 포함된 ECS 필드에 대한 실제 ECS 메타데이터를 갖습니다. `i-0000000` 관련 트래픽의 경우 이러한 필드의 값은 '-'가 됩니다.

- `ecs-container-id`와 `ecs-second-container-id`는 VPC 흐름 로그 서비스가 ECS 이벤트 스트림에서 수신할 때 순서가 지정됩니다. ECS 콘솔 또는 DescribeTask API 직접 호출에서 볼 수 있는 순서와 동일하다는 보장은 없습니다. 태스크가 계속 실행되는 동안 컨테이너가 STOPPED 상태가 되면 로그에 계속 표시될 수 있습니다.
- ECS 메타데이터와 IP 트래픽 로그는 서로 다른 두 소스에서 가져온 것입니다. 업스트림 종속성에서 필요한 모든 정보를 얻는 즉시 ECS 트래픽 계산이 시작됩니다.

  새 태스크를 시작한 후, 기본 네트워크 인터페이스에 대한 IP 트래픽이 수신되고 해당 ECS 태스크가 실행 중임을 나타내는 메타데이터 포함 ECS 이벤트가 수신되는 두 가지 조건이 모두 충족되면 ECS 필드 계산이 시작됩니다.

  태스크를 중지한 후, 기본 네트워크 인터페이스에 대한 IP 트래픽이 더 이상 수신되지 않거나(또는 1일 이상 지연된 IP 트래픽이 수신됨), 해당 ECS 태스크가 더 이상 실행되지 않음을 나타내는 메타데이터 포함 ECS 이벤트가 수신되는 두 가지 조건이 모두 충족되면 ECS 필드 계산이 중지됩니다.
- `awsvpc` 네트워크 모드에서 시작된 ECS 태스크만 지원됩니다.

`encryption-status` 필드에만 적용되는 제한 사항:

- 일부 네트워크 어플라이언스의 암호화 상태 보고 제한으로 인해 일부 흐름에서 암호화 상태가 '-'(사용할 수 없음)일 수 있습니다. 사용자는 분석에서 이러한 흐름을 무시할 수 있습니다.
- 모니터링 모드에서 암호화된 상태로 표시된다고 해서 적용 모드에서 흐름이 허용된다는 것은 아닙니다. 그 반대의 경우도 마찬가지입니다.
  - 흐름이 모니터링 모드에서 암호화된 경우 적용 모드에서 규정을 준수하지 않을 수 있습니다.
    - 흐름에 AWS 서비스에서 생성한 ENI가 포함된 경우 서비스가 암호화 제어를 지원해야 합니다.
    - 흐름이 VPC 피어링을 통과하는 경우 피어링된 VPC는 암호화 제어를 강제하지 않을 수 있습니다.
  - 흐름이 모니터링 모드에서 암호화되지 않은 경우 흐름과 관련된 서비스가 제외 항목으로 추가되어도 여전히 적용 모드에서 규정을 준수할 수 있습니다.

버전 11에서 사용할 수 있는 흐름 로그 Amazon EC2 태그 필드 관련 제한 사항:

- 흐름 로그 구독의 소유자가 리소스의 태그를 소유하지 않은 경우 태그 필드는 계산되지 않습니다.

예를 들어 서브넷(SubnetA)을 다른 계정(AccountB)과 공유한 후, 네트워크 인터페이스에 대한 태그 필드를 지정하여 SubnetA의 Flow Logs 구독을 생성한다고 가정해 보겠습니다. AccountB이 사용자가 구성한 키로 태그가 지정된 네트워크 인터페이스를 시작하는 경우, 구독에 AccountB가 시작한 네트워크 인터페이스의 트래픽 로그가 수신되지만 구독에서 구성한 태그 필드는 보안상의 이유로 계산되지 않습니다. 원하는 경우 AccountB이 시작한 네트워크 인터페이스에 태그를 지정하여 태그를 표시할 수 있습니다.

- VPC/서브넷 리소스 수준에서 태그 필드를 사용하여 흐름 로그 구독을 생성하는 경우 태그가 지정되지 않은 네트워크 인터페이스에 대해 생성된 모든 트래픽도 구독에 전달됩니다. 태그가 지정되지 않은 리소스의 경우 태그 필드 값은 '-'입니다.
- '-'는 값이 없거나 누락되었음을 나타내는 예약된 문자이므로 Auto Scaling 그룹의 이름에 이 문자가 포함된 경우 해당 그룹은 아무 태그도 표시할 수 없습니다.
- 태그 필드가 포함된 흐름 로그 구독을 생성하면 고객을 대신하여 몇 가지 리소스가 생성됩니다. 태그 값에 대한 업데이트를 사용하기 위해 EventBridge 관리형 규칙이 생성되어 흐름 로그 서비스로 태그 변경 사항을 전송합니다. 태그 필드가 포함된 구독이 모두 삭제되면 이러한 EventBridge 관리형 규칙이 자동으로 정리됩니다. 이러한 EventBridge 관리형 규칙을 수동으로 삭제하지 마세요. 그러면 태그 값 업데이트가 크게 지연됩니다. 이 생성/정리를 제어하고 태그 값에 대한 액세스를 제공하기 위해 계정에 서비스 연결 역할이 생성됩니다. 자세한 내용은 VPC 흐름 로그에 서비스 링크 역할 사용 섹션을 참조하세요.
- Amazon CloudWatch Logs 구문 분석 및 Athena 구문 분석 제약 조건으로 인해 태그 값의 모든 특수 문자는 UTF-8 퍼센트 인코딩을 사용하여 인코딩됩니다. 이는 Athena에서 `url_decode`를 사용하여 기본적으로 디코딩하거나, URL 또는 URI 디코더를 통해 디코딩할 수 있습니다.
- 새 구독이 생성된 후 첫 1시간 동안은 태그 값이 누락되거나 부정확할 수 있습니다. 이 첫 1시간이 지나면 태그 값은 1분 단위로 태그가 지정된 리소스의 값을 정확하게 반영합니다.
- 동일한 리소스에 대해 1초 내에 태그가 여러 번 변경되는 경우 업데이트가 손실되어 최대 1시간 동안 태그 값이 최신 상태가 아닐 수 있습니다.
- Auto Scaling 그룹 태그 필드의 경우 계정에 활성화된 CloudTrail 추적이 하나 이상 있어야 합니다. 활성화된 추적이 없으면 Auto Scaling 그룹 태그 값이 오래되었거나 부정확할 수 있습니다.

`next-hop-` 필드에만 적용되는 제한 사항:

- `next-hop-az-id`를 제외하고 흐름 로그 구독 소유자가 다음 홉 네트워크 인터페이스를 소유하지 않은 경우 다음 홉 필드는 계산되지 않습니다.
- 다음 홉에 네트워크 인터페이스(예: 인터넷 게이트웨이로 전송되는 트래픽)가 없는 경우 다음 홉 필드를 사용할 수 없습니다.

- 교차 리전 트래픽에는 다음 홉 필드를 사용할 수 없습니다.
- 일부 네트워크 서비스(예: Transit Gateway 및 Network Load Balancer)의 수신 트래픽에는 다음 홉 필드를 사용할 수 없습니다.
- 트래픽이 미들박스(예: Transit Gateway 또는 Network Load Balancer)를 통과하는 경우, 다음 홉 네트워크 인터페이스는 트래픽의 최종 목적지가 아니라 미들박스(예: transit gateway attachment)와 연결된 네트워크 인터페이스입니다.

가격 책정

흐름 로그를 게시하면 벤딩 로그에 대한 데이터 모으기 및 보관 요금이 적용됩니다. 벤딩 로그를 게시할 때 요금에 대해 자세히 알아보려면 Amazon CloudWatch Pricing(Amazon CloudWatch 요금)을 열고 Logs(로그)를 선택하고 Vended Logs(벤딩 로그)를 찾으세요.

흐름 로그 게시의 요금을 추적하려는 경우 대상 리소스에 비용 할당 태그를 적용할 수 있습니다. 이후에 AWS 비용 할당 보고서에 이러한 태그로 집계된 사용량 및 비용이 포함됩니다. 비즈니스 범주를 나타내는 태그(예: 비용 센터, 애플리케이션 이름 또는 소유자)를 적용하여 비용을 정리할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- AWS Billing 사용 설명서의 비용 할당 태그 사용
- Amazon CloudWatch Logs 사용 설명서의 Amazon CloudWatch Logs의 로그 그룹에 태그 지정
- Amazon Simple Storage Service 사용 설명서의 비용 할당 S3 버킷 태그 사용
- Amazon Data Firehose 개발자 안내서의 전송 스트림 태그 지정

흐름 로그 작업

Amazon EC2 및 Amazon VPC의 콘솔을 사용하여 흐름 로그를 연동할 수 있습니다.

작업

- 1. IAM으로 흐름 로그 사용 제어
- 2. 흐름 로그 생성
- 3. 흐름 로그 태그 지정
- 4. 흐름 로그 삭제
- 명령줄 개요

1. IAM으로 흐름 로그 사용 제어

기본적으로 사용자에게는 흐름 로그 사용 권한이 없습니다. 사용자에게 흐름 로그를 생성, 설명, 삭제하는 권한을 부여하는 정책이 연결된 IAM 역할을 만들 수 있습니다.

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

JSON

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

자세한 내용은 the section called “Amazon VPC가 IAM과 작동하는 방식” 단원을 참조하세요.

2. 흐름 로그 생성

VPC, 서브넷 또는 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 흐름 로그를 생성할 때 흐름 로그의 대상을 지정해야 합니다. 자세한 내용은 다음 자료를 참조하세요.

- the section called “CloudWatch Logs에 게시하는 흐름 로그 생성”
- the section called “Amazon S3에 게시하는 흐름 로그 생성”
- the section called “Amazon Data Firehose에 게시하는 흐름 로그 생성”

3. 흐름 로그 태그 지정

언제든지 흐름 로그의 태그를 추가하거나 제거할 수 있습니다.

흐름 로그의 태그를 관리하는 방법

1. 다음 중 하나를 수행하세요.
   - https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택합니다. 네트워크 인터페이스의 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택합니다. VPC에 대한 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets를 선택합니다. 서브넷의 확인란을 선택합니다.
2. Flow Logs(흐름 로그)를 선택합니다.
3. 작업, 태그 관리를 선택합니다.
4. 태그를 추가하려면 Add new tag(새 태그 추가)를 선택하고 키와 값을 입력합니다. 태그를 제거하려면 제거를 선택합니다.
5. 태그 추가 또는 제거를 마쳤으면 Save(저장)를 선택합니다.

4. 흐름 로그 삭제

언제든지 흐름 로그를 삭제할 수 있습니다. 흐름 로그를 삭제하면 데이터 수집 중단까지 몇 분 정도 걸릴 수 있습니다.

흐름 로그를 삭제해도 대상의 로그 데이터가 삭제되거나 대상 리소스가 수정되지 않습니다. 대상 서비스의 콘솔을 사용하여 기존 흐름 로그 데이터를 대상에서 직접 삭제하고 대상 리소스를 정리해야 합니다.

흐름 로그를 삭제하는 방법

1. 다음 중 하나를 수행하세요.
   - https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택합니다. 네트워크 인터페이스의 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택합니다. VPC에 대한 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets를 선택합니다. 서브넷의 확인란을 선택합니다.
2. Flow Logs(흐름 로그)를 선택합니다.
3. Actions(작업), Delete flow logs(흐름 로그 삭제)를 선택합니다.

4. 확인 메시지가 나타나면 **delete**를 입력한 다음 삭제(Delete)를 선택합니다.

명령줄 개요

이 페이지에서 설명한 작업은 명령줄을 사용하여 수행할 수 있습니다.

흐름 로그 생성

- create-flow-logs(AWS CLI)
- New-EC2FlowLog(AWS Tools for Windows PowerShell)

흐름 로그 설명

- describe-flow-logs(AWS CLI)
- Get-EC2FlowLog(AWS Tools for Windows PowerShell)

흐름 로그 태그 지정

- create-tags 및 delete-tags(AWS CLI)
- New-EC2Tag 및Remove-EC2Tag(AWS Tools for Windows PowerShell)

흐름 로그 삭제

- delete-flow-logs(AWS CLI)
- Remove-EC2FlowLog(AWS Tools for Windows PowerShell)

CloudWatch Logs에 흐름 로그 게시

흐름 로그는 흐름 로그 데이터를 Amazon CloudWatch에 직접 게시할 수 있습니다. Amazon CloudWatch는 포괄적인 모니터링 및 관찰성 서비스로, 다양한 AWS 리소스뿐만 아니라 자체 애플리케이션과 서비스에서 지표, 로그 및 이벤트 데이터를 수집하고 추적합니다. CloudWatch는 리소스 사용률, 애플리케이션 성능 및 운영 상태에 대한 가시성을 제공하여 시스템 전반의 성능 변화와 잠재적 문제를 감지하고 대응할 수 있도록 지원합니다. CloudWatch를 사용하면 경보를 설정하고, 로그와 지표를 시각화하고, 자동으로 대응하여 클라우드 리소스를 수집하고 최적화할 수 있습니다. 이는 클라우드 기반 인프라와 애플리케이션의 신뢰성, 가용성 및 성능을 보장하는 데 필수적인 도구입니다.

CloudWatch Logs에 게시하는 경우 흐름 로그 데이터는 로그 그룹에 게시되고, 각 네트워크 인터페이스는 로그 그룹에 고유의 로그 스트림을 가집니다. 로그 스트림에는 흐름 로그 레코드가 포함됩니다. 여러 개의 흐름 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.

CloudWatch Logs에서 timestamp 필드는 흐름 로그 레코드에서 캡처된 시작 시간에 해당합니다. ingestionTime 필드는 CloudWatch Logs에서 흐름 로그 레코드가 수신된 날짜와 시간을 나타냅니다. 이 타임스탬프는 흐름 로그 레코드에 캡처된 종료 시간보다 이후입니다.

CloudWatch Logs에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs로 전송된 로그를 참조하세요.

가격 책정

CloudWatch Logs에 흐름 로그를 게시할 때는 Vended 로그에 대한 데이터 수집 및 아카이브 요금이 부과됩니다. 자세히 알아보려면 Amazon CloudWatch Pricing(Amazon CloudWatch 요금)을 열고, Logs(로그)를 선택하고, Vended Logs(벤딩 로그)를 찾으세요.

내용

- CloudWatch Logs에 흐름 로그를 게시하는 IAM 역할
- CloudWatch Logs에 게시하는 흐름 로그 생성
- CloudWatch Logs로 흐름 로그 레코드 보기
- 흐름 로그 레코드 검색
- CloudWatch Logs에서 흐름 로그 레코드 처리

CloudWatch Logs에 흐름 로그를 게시하는 IAM 역할

흐름 로그와 연결된 IAM 역할에는 CloudWatch Logs의 지정된 로그 그룹에 흐름 로그를 게시할 권한이 있어야 합니다. IAM 역할은 AWS 계정에 속해야 합니다.

IAM 역할에 연결된 IAM 정책에는 최소한 다음과 같은 권한이 포함되어야 합니다.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": "*"
    }
  ]
}
```

흐름 로그 서비스에서 역할을 수임할 수 있는 다음과 같은 신뢰 정책이 역할에 있는지 확인하세요.

JSON

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

혼동된 대리자 문제로부터 자신을 보호하기 위하여 `aws:SourceAccount` 및 `aws:SourceArn` 조건 키를 사용할 것을 권장합니다. 예를 들어 이전 신뢰 정책에 다음 조건 블록을 추가할 수 있습니다. 소스 계정은 흐름 로그의 소유자이고 소스 ARN은 흐름 로그 ARN입니다. 흐름 로그 ID를 모르는 경우 ARN의 해당 부분을 와일드카드(*) 로 바꾼 다음 흐름 로그를 만든 후 정책을 업데이트할 수 있습니다.

```
"Condition": {
    "StringEquals": {
```

```
            "aws:SourceAccount": "account_id"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:ec2:region:account_id:vpc-flow-log/flow-log-id"
        }
}
```

흐름 로그에 대한 IAM 역할 생성

위에 설명된 대로 기존 역할을 업데이트할 수 있습니다. 또는 다음과 같은 절차에 따라 흐름 로그에서 사용할 새 역할을 생성할 수 있습니다. 이 역할은 흐름 로그를 생성할 때 지정합니다.

흐름 로그에 대한 IAM 역할 생성

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택합니다.
3. [정책 생성]을 선택합니다.
4. 정책 생성 페이지에서 다음을 수행합니다.
   a. JSON을 선택합니다.
   b. 이 창의 내용을 이 섹션의 시작 부분에 있는 권한 정책으로 대체합니다.
   c. 다음을 선택합니다.
   d. 정책의 이름과 설명(선택 사항) 및 태그를 입력한 다음에 정책 생성을 선택합니다.
5. 탐색 창에서 Roles를 선택합니다.
6. 역할 생성(Create role)을 선택합니다.
7. 신뢰할 수 있는 엔터티 유형(Trusted entity type)에서 사용자 지정 정책(Custom trust policy)을 선택합니다. Custom trust policy(사용자 지정 정책)에서 `"Principal": {},`을(를) 다음으로 대체하고 Next(다음)를 선택합니다.

   ```
   "Principal": {
      "Service": "vpc-flow-logs.amazonaws.com"
   },
   ```

8. Add permissions(권한 추가) 페이지에서 이 절차의 앞부분에서 생성한 정책의 확인란을 선택한 후 Next(다음)를 선택합니다.
9. 역할 이름을 입력하고 선택적으로 설명을 제공합니다.
10. 역할 생성(Create role)을 선택합니다.

CloudWatch Logs에 게시하는 흐름 로그 생성

VPC, 서브넷 또는 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 특정 IAM 역할을 사용하는 사용자로 이러한 단계를 수행하는 경우 `iam:PassRole` 작업을 사용할 수 있는 권한이 있는지 확인하세요.

사전 조건

요청을 수행하는 데 사용 중인 IAM 보안 주체에 `iam:PassRole` 작업을 직접적으로 호출할 수 있는 권한이 있는지 확인하세요.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/flow-log-role-name"
        }
    ]
}
```

콘솔을 사용하여 흐름 로그를 생성하는 방법

1. 다음 중 하나를 수행하세요.
   - https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택합니다. 네트워크 인터페이스의 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택합니다. VPC에 대한 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets를 선택합니다. 서브넷의 확인란을 선택합니다.
2. 작업, 흐름 로그 생성을 선택합니다.

3. 필터(Filter)에 기록할 트래픽 유형을 지정합니다. 모두(All)를 선택하여 수락된 트래픽 및 거부된 트래픽을 기록하거나, 거부(Reject)를 선택하여 거부된 트래픽만 기록하거나, 수락(Accept)을 선택해 수락된 트래픽만 기록합니다.
4. Maximum aggregation interval(최대 집계 간격)에서 흐름이 캡처되어 흐름 로그 레코드로 집계되는 최대 기간을 선택합니다.
5. Destination(대상)에서 Send to CloudWatch Logs(CloudWatch Logs로 전송)를 선택합니다.
6. 대상 로그 그룹에서 기존 로그 그룹의 이름을 선택하거나 새 흐름 로그의 이름을 입력합니다. 이름을 입력하면 로깅할 트래픽이 있을 때 로그 그룹이 생성됩니다.
7. 서비스 액세스에서 CloudWatch Logs에 로그를 게시할 권한이 있는 기존 IAM 서비스 역할을 선택하거나 새 서비스 역할을 생성하도록 선택합니다.
8. 로그 레코드 형식에서 흐름 로그 레코드의 형식을 선택합니다.
   - 기본 형식을 사용하려면 AWS 기본 형식을 선택하세요.
   - 사용자 지정 형식을 사용하려면 사용자 지정 형식을 선택하고 로그 형식에서 필드를 선택합니다.
9. 추가 메타데이터에서 Amazon ECS의 메타데이터를 로그 형식으로 포함할지 선택합니다.
10. (선택 사항) 새 태그 추가를 선택하여 흐름 로그에 태그를 적용합니다.
11. 흐름 로그 생성을 선택합니다.

명령줄을 사용하여 흐름 로그를 만들려면 다음을 수행합니다.

다음 명령 중 하나를 사용합니다.

- create-flow-logs(AWS CLI)
- New-EC2FlowLog(AWS Tools for Windows PowerShell)

다음 AWS CLI 예시에서는 지정된 서브넷에 대해 수락된 모든 트래픽을 캡처하는 흐름 로그를 생성합니다. 흐름 로그가 지정된 로그 그룹에 전송됩니다. `--deliver-logs-permission-arn` 파라미터에서는 CloudWatch Logs에 게시하는 데 필요한 IAM 역할을 지정합니다.

```
aws ec2 create-flow-logs --resource-type Subnet --resource-ids subnet-1a2b3c4d --
traffic-type ACCEPT --log-group-name my-flow-logs --deliver-logs-permission-arn
 arn:aws:iam::123456789101:role/publishFlowLogs
```

CloudWatch Logs로 흐름 로그 레코드 보기

CloudWatch Logs 콘솔을 사용하여 흐름 로그 레코드를 볼 수 있습니다. 흐름 로그를 생성하면 콘솔에 표시되는 데 몇 분 정도 걸릴 수도 있습니다.

콘솔을 사용하여 CloudWatch Logs에 게시된 흐름 로그 레코드를 보는 방법

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Logs, Log groups를 선택합니다.
3. 흐름 로그가 있는 로그 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 흐름 로그 레코드가 있는 로그 스트림의 이름을 선택합니다. 자세한 내용은 흐름 로그 레코드 단원을 참조하세요.

명령줄을 사용하여 CloudWatch Logs에 게시된 흐름 로그 레코드를 보는 방법

- get-log-events(AWS CLI)
- Get-CWLLogEvent(AWS Tools for Windows PowerShell)

흐름 로그 레코드 검색

CloudWatch Logs 콘솔을 사용하여 CloudWatch Logs에 게시된 흐름 로그 레코드를 검색할 수 있습니다. 지표 필터를 사용하여 흐름 로그 레코드를 필터링할 수 있습니다. 흐름 로그 레코드는 공백으로 구분됩니다.

CloudWatch Logs 콘솔을 사용하여 흐름 로그 레코드를 검색하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Logs, Log groups를 선택합니다.
3. 흐름 로그가 있는 로그 그룹을 선택한 다음에 검색 중인 네트워크 인터페이스를 아는 경우 로그 스트림을 선택합니다. 또는 Search log group(로그 그룹 검색)을 선택합니다. 로그 그룹에 네트워크 인터페이스가 많거나 선택한 시간 범위에 따라 시간이 걸릴 수 있습니다.
4. 이벤트 필터링에서 아래 문자열을 입력합니다. 여기서는 흐름 로그 레코드가 기본 형식을 사용한다고 가정합니다.

```
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport, protocol,
 packets, bytes, start, end, action, logstatus]
```

5. 필드의 값을 지정하여 필요에 따라 필터를 수정합니다. 다음 예시에서는 특정 원본 IP 주소를 기준으로 필터링합니다.

```
[version, accountid, interfaceid, srcaddr = 10.0.0.1, dstaddr, srcport, dstport,
 protocol, packets, bytes, start, end, action, logstatus]
[version, accountid, interfaceid, srcaddr = 10.0.2.*, dstaddr, srcport, dstport,
 protocol, packets, bytes, start, end, action, logstatus]
```

다음 예시에서는 대상 포트, 바이트 수 및 트래픽이 거부되었는지 여부를 기준으로 필터링합니다.

```
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport = 80 ||
 dstport = 8080, protocol, packets, bytes, start, end, action, logstatus]
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport = 80 ||
 dstport = 8080, protocol, packets, bytes >= 400, start, end, action = REJECT,
 logstatus]
```

CloudWatch Logs에서 흐름 로그 레코드 처리

CloudWatch Logs에서 수집한 다른 로그 이벤트처럼 흐름 로그 레코드를 처리할 수 있습니다. 로그 데이터 및 지표 필터 모니터링에 대한 자세한 내용을 알아보려면 Amazon CloudWatch Logs 사용 설명서의 필터를 사용하여 로그 이벤트에서 지표 생성을 참조하세요.

예: 흐름 로그에 대한 CloudWatch 지표 필터 및 경보 생성

이 예에서는 `eni-1a2b3c4d`에 대한 흐름 로그를 사용합니다. TCP 포트 22(SSH)를 거쳐 인스턴스에 연결하려는 시도가 한 시간 내에 10번 이상 거부된 경우 이를 알려 주는 알림을 만들 수 있습니다. 우선 경보를 만들려는 트래픽의 패턴과 일치하는 지표 필터를 만들어야 합니다. 그런 다음 지표 필터에 대한 경보를 만듭니다.

거부된 SSH 트래픽에 대한 지표 필터와 필터에 대한 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Logs, Log groups를 선택합니다.
3. 로그 그룹에 대한 확인란을 선택한 다음 작업(Actions),지표 필터 생성(Create metric filter)을 선택합니다.
4. 필터 패턴(Filter Pattern)에서는 다음과 같은 문자열을 입력합니다.

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

5. 테스트할 로그 데이터 선택에서 네트워크 인터페이스에 대한 로그 스트림을 선택합니다. (선택 사항) 필터 패턴과 일치하는 로그 데이터 행을 보려면 패턴 테스트를 선택합니다.
6. 준비가 되면 다음을 선택합니다.
7. 필터 이름, 지표 네임스페이스 및 지표 이름을 입력합니다. 지표 값을 1로 설정합니다. 완료되면 다음을 선택하고 지표 필터 생성을 선택합니다.
8. 탐색 창에서 Alarms, All alarms를 선택합니다.
9. 경보 생성을 선택하세요.
10. 생성한 지표 이름을 선택한 후 지표 선택을 선택합니다.
11. 경보를 다음과 같이 구성한 후 다음(Next)을 선택합니다.
    - 통계에서 합계를 선택합니다. 이것으로 지정된 기간 동안 데이터 포인트의 총 수를 캡처할 수 있습니다.
    - 기간에서 1시간을 선택합니다.
    - TimeSinceLastActive가 다음과 같은 때마다에는 크거나 같음을 선택하고 10을 임계값으로 입력합니다.
    - 추가 구성에서 경보에 대한 데이터 포인트를 기본값 1로 남겨둡니다.
12. 다음을 선택합니다.
13. 알림에서 기존 SNS 주제를 선택하거나 새 주제 생성을 선택하여 새로 생성합니다. Next(다음)를 선택합니다.
14. 경보의 이름과 설명을 입력하고 다음을 선택합니다.
15. 경보 미리 보기를 완료했으면 경보 생성을 선택합니다.

Amazon S3에 흐름 로그 게시

흐름 로그는 흐름 로그 데이터를 Amazon S3에 게시할 수 있습니다. Amazon Simple Storage Service(S3)는 확장성과 내구성이 뛰어난 객체 스토리지 서비스입니다. 이는 웹을 통해 어디서든 원하는 양의 데이터를 저장하고 검색하도록 설계되었습니다. S3는 데이터 버전 관리, 암호화 및 액세스 제어를 위한 내장 기능으로 업계 최고의 내구성과 가용성을 제공합니다.

Amazon S3에 게시하는 경우 흐름 로그 데이터가 지정해 놓은 기존 Amazon S3 버킷에 게시됩니다. 모니터링된 모든 네트워크 인터페이스에 대한 흐름 로그 레코드는 버킷에 저장된 일련의 로그 파일 객체

에 게시됩니다. 흐름 로그가 VPC에 대한 데이터를 캡처하면, 흐름 로그가 모든 네트워크 인터페이스에 대한 흐름 로그 레코드를 선택된 VPC에 게시합니다.

흐름 로그와 함께 사용할 Amazon S3 버킷을 만드는 방법은 Amazon S3 사용 설명서의 버킷 생성을 참조하세요.

VPC 흐름 로그 수집, 흐름 로그 처리 및 흐름 로그 시각화를 간소화하는 방법에 대한 자세한 내용은 AWS Solutions Library의 OpenSearch를 통한 중앙 집중식 로깅을 참조하세요.

CloudWatch Logs에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 Amazon S3로 전송된 로그를 참조하세요.

가격 책정

Amazon S3에 흐름 로그를 게시할 때는 Vended 로그에 대한 데이터 수집 및 아카이브 요금이 부과됩니다. 자세히 알아보려면 Amazon CloudWatch Pricing(Amazon CloudWatch 요금)을 열고, Logs(로그)를 선택하고, Vended Logs(벤딩 로그)를 찾으세요.

내용

- 흐름 로그 파일
- Amazon S3 버킷의 흐름 로그에 대한 권한
- SSE-KMS를 사용할 경우 필요한 키 정책
- Amazon S3 로그 파일 권한
- Amazon S3에 게시하는 흐름 로그 생성
- Amazon S3로 흐름 로그 레코드 보기

흐름 로그 파일

VPC Flow Logs는 VPC에서 들어오고 나가는 IP 트래픽에 대한 데이터를 로그 레코드로 수집하고 해당 레코드를 로그 파일로 집계한 다음 5분 간격으로 Amazon S3 버킷에 로그 파일을 게시합니다. 여러 파일이 게시될 수 있으며 각 로그 파일에는 이전 5분 동안 기록된 IP 트래픽에 대한 흐름 로그 레코드의 일부 또는 전체가 포함될 수 있습니다.

Amazon S3에서 흐름 로그 파일의 마지막 수정(Last modified) 필드는 파일이 Amazon S3 버킷에 업로드된 날짜와 시간을 나타냅니다. 파일 이름의 타임스탬프보다 이후이며 파일을 Amazon S3 버킷에 업로드하는 데 걸리는 시간에 따라 다릅니다.

로그 파일 형식

로그 파일에 대해 다음 형식 중 하나를 지정할 수 있습니다. 각 파일은 단일 Gzip 파일로 압축됩니다.

- 텍스트(Text) – 일반 텍스트. 이것은 기본 형식입니다.
- Parquet – Apache Parquet은 열 기반 데이터 형식입니다. Parquet 형식의 데이터에 대한 쿼리는 일반 텍스트 데이터에 대한 쿼리에 비해 10배에서 100배 빠릅니다. Gzip 압축을 사용하는 Parquet 형식 데이터는 Gzip 압축을 사용하는 일반 텍스트보다 스토리지 공간을 20% 적게 사용합니다.

> **Note**
>
> Gzip 압축을 사용하는 Parquet 형식 데이터가 집계 기간별로 100KB 미만이라면 데이터를 Parquet 형식으로 저장할 경우 Parquet 파일 메모리 요구 사항으로 인해 Gzip 압축을 사용하는 일반 텍스트보다 더 많은 공간을 차지할 수 있습니다.

로그 파일 옵션

필요한 경우 다음과 같은 옵션을 지정할 수 있습니다.

- Hive 호환 S3 접두사 – 분할을 Hive 호환 도구로 가져오는 대신 Hive 호환 접두사를 활성화합니다. 쿼리 실행 전에 MSCK REPAIR TABLE 명령을 사용합니다.
- 시간당 분할 – 대량의 로그가 있고 일반적으로 특정 시간까지 쿼리를 타겟팅하는 경우 로그를 시간별로 분할하여 더 결과를 빠르게 얻고 쿼리 비용을 절감할 수 있습니다.

로그 파일 S3 버킷 구조

로그 파일은 흐름 로그의 ID, 리전, 생성된 날짜 및 대상 옵션에 따라 폴더 구조를 사용하여 지정된 Amazon S3 버킷에 저장됩니다.

기본적으로 파일은 다음 위치로 전달됩니다.

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/
```

Hive 호환 S3 접두사를 사용하도록 설정하면 파일이 다음 위치로 전달됩니다.

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/aws-service=vpcflowlogs/
aws-region=region/year=year/month=month/day=day/
```

시간별 분할을 사용하도록 설정하면 파일이 다음 위치로 전달됩니다.

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/hour/
```

Hive 호환 분할을 사용하도록 설정하고 시간당 흐름 로그를 분할하면 파일이 다음 위치로 전달됩니다.

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/aws-service=vpcflowlogs/
aws-region=region/year=year/month=month/day=day/hour=hour/
```

로그 파일 이름

로그 파일의 파일 이름은 흐름 로그 ID, 리전 및 생성 날짜 및 시간을 기반으로 합니다. 파일 이름은 다음 형식을 사용합니다.

```
aws_account_id_vpcflowlogs_region_flow_log_id_YYYYMMDDTHHmmZ_hash.log.gz
```

예를 들어, 다음은 June 20, 2018, 16:20 UTC에 us-east-1 리전의 리소스에 대해 AWS 계정 123456789012에서 생성한 흐름 로그에 대한 로그 파일의 예를 보여 줍니다. 종료 시간이 16:20:00에서 16:24:59 사이인 흐름 로그 레코드가 파일에 포함됩니다.

```
123456789012_vpcflowlogs_us-east-1_fl-1234abcd_20180620T1620Z_fe123456.log.gz
```

Amazon S3 버킷의 흐름 로그에 대한 권한

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

흐름 로그를 생성하는 사용자가 버킷을 소유하고 해당 버킷에 대한 `PutBucketPolicy` 및 `GetBucketPolicy` 권한을 소유한 경우, 다음 정책을 해당 버킷에 자동으로 연결합니다. 이 정책은 버킷에 연결된 모든 기존 정책을 덮어씁니다.

그렇지 않으면 버킷 소유자가 이 정책을 버킷에 추가하고 흐름 로그 작성자의 AWS 계정 ID 지정 또는 흐름 로그 생성이 실패합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 버킷 정책 사용을 참조하세요.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
                }
            }
        }
    ]
}
```

*my-s3-arn*에 지정한 ARN은 Hive 호환 S3 접두사를 사용하는지 여부에 따라 다릅니다.

- 기본 접두사

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*
```

- Hive 호환 S3 접두사

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/aws-account-id=account_id/*
```

개별 AWS 계정 ARN 대신 로그 전송 서비스 보안 주체에 이 권한들을 부여하는 것이 좋습니다. 또한 혼동된 대리자 문제로부터 보호하려면 `aws:SourceAccount` 및 `aws:SourceArn` 조건 키를 사용하는 것 좋습니다. 소스 계정은 흐름 로그의 소유자이고 원본 ARN은 로그 서비스의 와일드카드(*) ARN입니다.

로그 전송 서비스는 `HeadBucket` Amazon S3 API 작업을 호출하여 S3 버킷의 존재 및 위치를 확인합니다. 로그 전송 서비스에 이 작업을 호출할 수 있는 권한을 부여하지 않아도 됩니다. S3 버킷의 존재 여부를 확인할 수 없는 경우에도 서비스는 VPC Flow Logs를 계속 전송합니다. 하지만 CloudTrail 로그에서 `HeadBucket`에 대한 호출에 `AccessDenied` 오류가 발생합니다.

SSE-KMS를 사용할 경우 필요한 키 정책

해당 S3 버킷에서 Amazon S3-관리형 키(SSE-S3)를 사용한 서버 측 암호화 또는 KMS 키(SSE-KMS)를 사용한 서버 측 암호화를 활성화하여 Amazon S3 버킷의 데이터를 보호할 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서의 서버 측 암호화를 사용하여 데이터 보호를 참조하세요.

SSE-S3를 선택하면 추가 구성이 필요하지 않습니다. Amazon S3는 암호화 키를 처리합니다.

SSE-KMS를 선택하면 고객 관리형 키 ARN을 사용해야 합니다. 키 ID를 사용하는 경우 흐름 로그를 생성할 때 LogDestination 전송 불가 오류가 발생할 수 있습니다. 또한 로그 전달 계정이 S3 버킷에 쓸 수 있도록 고객 관리형 키에 대한 키 정책을 업데이트해야 합니다. SSE-KMS와 함께 사용하는 데 필요한 키 정책에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 Amazon S3 버킷 서버 측 암호화를 참조하세요.

Amazon S3 로그 파일 권한

필요한 버킷 정책 외에도, Amazon S3는 ACL(액세스 제어 목록)을 사용하여 흐름 로그에서 생성한 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 내용은 Amazon S3 사용 설명서의 액세스 제어 목록(ACL) 개요를 참조하세요.

Amazon S3에 게시하는 흐름 로그 생성

Amazon S3 버킷을 생성하고 구성한 후에는 네트워크 인터페이스, 서브넷 및 VPC에 대한 흐름 로그를 생성할 수 있습니다.

사전 조건:

흐름 로그를 생성하는 IAM 보안 주체는 대상 Amazon S3 버킷에 흐름 로그를 게시하는 데 필요한 다음 권한이 있는 IAM 역할을 사용해야 합니다.

JSON

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

콘솔을 사용하여 흐름 로그를 생성하는 방법

1. 다음 중 하나를 수행하세요.
   - https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택합니다. 네트워크 인터페이스의 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택합니다. VPC에 대한 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets를 선택합니다. 서브넷의 확인란을 선택합니다.
2. 작업, 흐름 로그 생성을 선택합니다.
3. 필터(Filter)에서 로깅할 IP 트래픽 데이터의 유형을 지정합니다.
   - 수락 - 수락한 트래픽만 로그합니다.

- 거부 - 거부한 트래픽만 로그합니다.
- 모두 – 허용 및 거부된 트래픽을 로그합니다.

4. 최대 집계 간격(Maximum aggregation interval)에서 흐름이 캡처되어 흐름 로그 레코드로 집계되는 최대 기간을 선택합니다.
5. 대상에서 Amazon S3 버킷으로 전송(Send to an Amazon S3 bucket)을 선택합니다.
6. S3 버킷 ARN의 경우 기존 Amazon S3 버킷의 Amazon 리소스 이름(ARN)을 지정합니다. 필요한 경우 하위 폴더를 포함할 수 있습니다. 예를 들어 `my-bucket`이란 이름의 버킷에 `my-logs`란 이름의 하위 폴더를 지정하려면 다음 ARN을 사용하세요.

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

   버킷에 `AWSLogs`를 하위 폴더 이름으로 사용할 수 없습니다. 이것은 예약된 용어입니다.

   버킷을 소유한 경우, 자동으로 리소스 정책을 생성하여 버킷에 연결합니다. 자세한 내용은 Amazon S3 버킷의 흐름 로그에 대한 권한 단원을 참조하세요.
7. 로그 레코드 형식에서 흐름 로그 레코드의 형식을 지정합니다.

   - 기본 흐름 로그 레코드 형식을 사용하려면 AWS 기본 형식을 선택하세요.
   - 사용자 지정 형식을 만들려면 사용자 지정 형식을 선택하세요. 로그 형식에 대해 흐름 로그 레코드에 포함할 필드를 선택하세요.
8. 추가 메타데이터에서 Amazon ECS의 메타데이터를 로그 형식으로 포함할지 선택합니다.
9. 로그 파일 형식의 경우 로그 파일의 형식을 지정합니다.

   - Text – 일반 텍스트. 이것은 기본 형식입니다.
   - Parquet – Apache Parquet은 열 기반 데이터 형식입니다. Parquet 형식의 데이터에 대한 쿼리는 일반 텍스트 데이터에 대한 쿼리에 비해 10배에서 100배 빠릅니다. Gzip 압축을 사용하는 Parquet 형식 데이터는 Gzip 압축을 사용하는 일반 텍스트보다 스토리지 공간을 20% 적게 사용합니다.
10. (선택 사항) Hive 호환 S3 접두사를 사용하려면 Hive 호환 S3 접두사,활성화를 선택합니다.
11. (선택 사항) 흐름 로그를 시간당 분할하려면1시간마다 (60분)을 선택합니다.
12. (선택 사항) 흐름 로그에 태그를 추가하려면새 태그 추가를 선택하여 태그 키와 값을 지정하십시오.
13. 흐름 로그 생성을 선택합니다.

명령줄 도구를 사용하여 Amazon S3에 게시하는 흐름 로그 생성

다음 명령 중 하나를 사용합니다.

- create-flow-logs(AWS CLI)
- New-EC2FlowLog(AWS Tools for Windows PowerShell)

다음 AWS CLI 예시에서는 지정된 VPC의 모든 트래픽을 캡처하고 지정된 Amazon S3 버킷에 흐름 로그를 전송하는 흐름 로그를 생성합니다. `--log-format` 파라미터는 흐름 로그 레코드의 사용자 지정 형식을 지정합니다.

```
aws ec2 create-flow-logs --resource-type VPC --resource-ids vpc-00112233344556677 --
traffic-type ALL --log-destination-type s3 --log-destination arn:aws:s3:::flow-log-
bucket/custom-flow-logs/ --log-format '${version} ${vpc-id} ${subnet-id} ${instance-
id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-
srcaddr} ${pkt-dstaddr}'
```

Amazon S3로 흐름 로그 레코드 보기

Amazon S3 콘솔을 사용하여 흐름 로그 레코드를 볼 수 있습니다. 흐름 로그를 생성하면 콘솔에 표시되는 데 몇 분 정도 걸릴 수도 있습니다.

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

Amazon S3에 게시된 흐름 로그 레코드를 보려면

1. https://console.aws.amazon.com/s3/에서 S3 콘솔을 엽니다.
2. 버킷 이름을 선택하여 세부 정보 페이지를 엽니다.
3. 로그 파일이 있는 폴더로 이동합니다. 예: *prefix*/AWSLogs/*account_id*/vpcflowlogs/*region*/*year*/*month*/*day*/.
4. 파일 이름 옆에 있는 확인란을 선택한 다음에 Download(다운로드)를 선택합니다.

또한 Amazon Athena를 사용해 로그 파일의 흐름 로그 레코드를 쿼리할 수도 있습니다. Amazon Athena는 표준 SQL을 사용해 Amazon S3에 저장된 데이터를 더 쉽게 분석할 수 있는 대화식 쿼리 서비스입니다. 자세한 내용은 Amazon Athena 사용 설명서의 Amazon VPC 흐름 로그 쿼리 방법을 참조하세요.

Amazon Data Firehose에 흐름 로그 게시

흐름 로그에서는 흐름 로그 데이터를 Amazon Data Firehose에 직접 게시할 수 있습니다. Amazon Data Firehose는 실시간 데이터 스트림을 수집하고 변환하여 다양한 AWS 데이터 스토어 및 분석 서비스로 전송하는 완전관리형 서비스로, 사용자 대신 데이터 수집을 처리합니다.

VPC 흐름 로그의 경우 Firehose가 유용할 수 있습니다. VPC 흐름 로그는 VPC의 네트워크 인터페이스에서 송수신되는 IP 트래픽에 대한 정보를 수집합니다. 이 데이터는 보안 모니터링, 성능 분석, 규제 준수에 매우 중요할 수 있습니다. 그러나 이러한 지속적인 로그 데이터 흐름의 저장과 처리를 관리하는 것은 복잡하고 리소스 집약적인 작업일 수 있습니다.

Firehose를 VPC 흐름 로그와 통합하면 Amazon S3 또는 Amazon Redshift 등의 선호하는 대상으로 이 데이터를 전송할 수 있습니다. Firehose는 VPC 흐름 로그의 수집, 변환, 전송을 처리하도록 확장되므로 운영 부담이 줄어듭니다. 이를 통해 기본 인프라에 대해 걱정할 필요 없이 로그를 분석하고 인사이트를 도출하는 데 집중할 수 있습니다.

또한 Firehose는 데이터 변환, 압축, 암호화와 같은 기능을 제공하여 VPC 흐름 로그 처리 파이프라인의 효율성과 보안을 강화할 수 있습니다. VPC 흐름 로그에 Firehose를 사용하면 데이터 관리를 간소화하고 네트워크 트래픽 데이터에서 인사이트를 얻을 수 있습니다.

Amazon Data Firehose에 게시할 때 흐름 로그 데이터는 일반 텍스트 형식으로 Amazon Data Firehose 전송 스트림에 게시됩니다.

가격 책정

표준 모으기 및 전송 요금이 적용됩니다. 자세히 알아보려면 Amazon CloudWatch Pricing(Amazon CloudWatch 요금)을 열고, Logs(로그)를 선택하고, Vended Logs(벤딩 로그)를 찾으세요.

내용

- 교차 계정 전송에 대한 IAM 역할
- Amazon Data Firehose에 게시하는 흐름 로그 생성

교차 계정 전송에 대한 IAM 역할

Amazon Data Firehose에 게시할 때 모니터링할 리소스와 동일한 계정(소스 계정) 또는 상이한 계정(대상 계정)에 있는 전송 스트림을 선택할 수 있습니다. Amazon Data Firehose에 대한 흐름 로그의 교차 계정 전송을 활성화하려면 소스 계정에서 IAM 역할을 생성하고 대상 계정에서 IAM 역할을 생성해야 합니다.

역할

- 소스 계정 역할
- 대상 계정 역할

소스 계정 역할

소스 계정에서 다음과 같은 권한을 부여하는 역할을 생성합니다. 이 예시에서는 역할 이름이 mySourceRole이지만, 이 역할에 대해 다른 이름을 선택할 수 있습니다. 마지막 명령문에서는 대상 계정의 역할에 이 역할 수임을 허용합니다. 조건문에서는 지정된 리소스를 모니터링할 때만 이 역할이 로그 전송 서비스에만 전달되도록 합니다. 정책을 생성할 때 모니터링 중인 VPC, 네트워크 인터페이스 또는 서브넷을 조건 키(iam:AssociatedResourceARN)로 지정합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/mySourceRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "delivery.logs.amazonaws.com"
                },
                "StringLike": {
                    "iam:AssociatedResourceARN": [
                        "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-00112233344556677"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:GetLogDelivery"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::111122223333:role/
AWSLogDeliveryFirehoseCrossAccountRole"
        }
    ]
}
```

로그 전송 서비스에서 역할을 수임할 수 있는 다음과 같은 신뢰 정책이 이 역할에 있는지 확인하세요.

JSON

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

소스 계정에서 다음과 같은 절차를 사용하여 역할을 생성합니다.

소스 계정 역할을 생성하는 방법

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택합니다.
3. [정책 생성]을 선택합니다.
4. 정책 생성 페이지에서 다음을 수행합니다.
    a. JSON을 선택합니다.
    b. 이 창의 내용을 이 섹션의 시작 부분에 있는 권한 정책으로 대체합니다.

c. 다음을 선택합니다.

d. 정책의 이름과 설명(선택 사항) 및 태그를 입력한 다음에 정책 생성을 선택합니다.

5. 탐색 창에서 Roles를 선택합니다.
6. 역할 생성(Create role)을 선택합니다.
7. 신뢰할 수 있는 엔터티 유형(Trusted entity type)에서 사용자 지정 정책(Custom trust policy)을 선택합니다. Custom trust policy(사용자 지정 신뢰 정책)에서 로그 전송 서비스를 지정하는 다음으로 `"Principal": {},`을 대체합니다. 다음을 선택합니다.

```
"Principal": {
   "Service": "delivery.logs.amazonaws.com"
},
```

8. 권한 추가 페이지에서 이 절차의 앞부분에서 생성한 정책의 확인란을 선택한 후 다음을 선택합니다.
9. 역할 이름을 입력하고 선택적으로 설명을 제공합니다.
10. 역할 생성(Create role)을 선택합니다.

대상 계정 역할

대상 계정에서 AWSLogDeliveryFirehoseCrossAccountRole로 시작하는 이름으로 역할을 생성합니다. 이 역할에서는 다음과 같은 권한을 부여해야 합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iam:CreateServiceLinkedRole",
          "firehose:TagDeliveryStream"
      ],
      "Resource": "*"
    }
  ]
}
```

이 역할을 수임할 수 있도록 소스 계정에서 생성한 역할이 허용되는 다음과 같은 신뢰 정책이 이 역할에 있는지 확인하세요.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/mySourceRole"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

대상 계정에서 다음과 같은 절차를 사용하여 역할을 생성합니다.

대상 계정 역할을 생성하는 방법

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택합니다.
3. [정책 생성]을 선택합니다.
4. 정책 생성 페이지에서 다음을 수행합니다.
    a. JSON을 선택합니다.
    b. 이 창의 내용을 이 섹션의 시작 부분에 있는 권한 정책으로 대체합니다.
    c. 다음을 선택합니다.
    d. AWSLogDeliveryFirehoseCrossAccountRole로 시작하는 정책 이름을 입력한 다음에 정책 생성을 선택합니다.
5. 탐색 창에서 Roles를 선택합니다.
6. 역할 생성(Create role)을 선택합니다.
7. 신뢰할 수 있는 엔터티 유형(Trusted entity type)에서 사용자 지정 정책(Custom trust policy)을 선택합니다. Custom trust policy(사용자 지정 신뢰 정책)에서 소스 계정 역할을 지정하는 다음으로 `"Principal": {},`을 대체합니다. 다음을 선택합니다.

```
"Principal": {
   "AWS": "arn:aws:iam::source-account:role/mySourceRole"
},
```

8. 권한 추가 페이지에서 이 절차의 앞부분에서 생성한 정책의 확인란을 선택한 후 다음을 선택합니다.
9. 역할 이름을 입력하고 선택적으로 설명을 제공합니다.
10. 역할 생성(Create role)을 선택합니다.

Amazon Data Firehose에 게시하는 흐름 로그 생성

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

사전 조건

- 대상 Amazon Data Firehose 전송 스트림을 생성합니다. Direct Put을 원본으로 사용합니다. 자세한 내용은 Amazon Data Firehose 전송 스트림 생성을 참조하세요.
- 흐름 로그를 생성하는 계정은 Amazon Data Firehose에 흐름 로그를 게시할 수 있는 다음 권한을 부여하는 IAM 역할을 사용해야 합니다.

  JSON

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

- 흐름 로그를 다른 계정에 게시하는 경우 the section called “교차 계정 전송에 대한 IAM 역할”에 설명된 대로 필수 IAM 역할을 생성합니다.

Amazon Data Firehose에 게시하는 흐름 로그 생성 방법

1. 다음 중 하나를 수행하세요.
   - https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택합니다. 네트워크 인터페이스의 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택합니다. VPC에 대한 확인란을 선택합니다.
   - https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets를 선택합니다. 서브넷의 확인란을 선택합니다.
2. 작업, 흐름 로그 생성을 선택합니다.
3. 필터(Filter)에 기록할 트래픽 유형을 지정합니다.
   - 수락 – 수락한 트래픽만 로그합니다.
   - 거부 – 거부한 트래픽만 로그합니다.
   - 모두 – 수락 및 거부한 트래픽을 로그합니다.
4. 최대 집계 간격(Maximum aggregation interval)에서 흐름이 캡처되어 흐름 로그 레코드로 집계되는 최대 기간을 선택합니다.
5. 대상(Destination)에서는 다음과 같은 옵션 중 하나를 선택합니다.
   - 동일한 계정의 Amazon Data Firehose로 보내기 - 전송 스트림과 모니터링할 리소스가 동일한 계정에 있습니다.
   - 다른 계정의 Amazon Data Firehose로 보내기 - 전송 스트림과 모니터링할 리소스가 상이한 계정에 있습니다.
6. Amazon Data Firehose 스트림 이름에는 생성한 전송 스트림을 선택합니다.
7. [크로스 계정 전송만 해당] 서비스 액세스에서 로그 게시 권한이 있는 교차 계정 전송용 기존 IAM 서비스 역할을 선택하세요. 또는 권한 설정을 선택하여 IAM 콘솔을 열고 서비스 역할을 생성하세요..
8. 로그 레코드 형식에서 흐름 로그 레코드의 형식을 지정합니다.
   - 기본 흐름 로그 레코드 형식을 사용하려면 AWS 기본 형식을 선택하세요.
   - 사용자 지정 형식을 만들려면 사용자 지정 형식을 선택하세요. 로그 형식에 대해 흐름 로그 레코드에 포함할 필드를 선택하세요.
9. 추가 메타데이터에서 Amazon ECS의 메타데이터를 로그 형식으로 포함할지 선택합니다.
10. (선택 사항) 태그 추가를 선택하여 흐름 로그에 태그를 적용합니다.

11. 흐름 로그 생성을 선택합니다.

명령줄을 사용하여 Amazon Data Firehose에 게시하는 흐름 로그 생성

다음 명령 중 하나를 사용합니다.

- create-flow-logs(AWS CLI)
- New-EC2FlowLog(AWS Tools for Windows PowerShell)

다음 AWS CLI 예시에서는 지정된 VPC의 모든 트래픽을 캡처하고 동일한 계정의 지정된 Amazon Data Firehose 전송 스트림에 흐름 로그를 전송하는 흐름 로그를 생성합니다.

```
aws ec2 create-flow-logs --traffic-type ALL \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --log-destination-type kinesis-data-firehose \
  --log-destination arn:aws:firehose:us-
east-1:123456789012:deliverystream/flowlogs_stream
```

다음 AWS CLI 예시에서는 지정된 VPC의 모든 트래픽을 캡처하고 상이한 계정의 지정된 Amazon Data Firehose 전송 스트림에 흐름 로그를 전송하는 흐름 로그를 생성합니다.

```
aws ec2 create-flow-logs --traffic-type ALL \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --log-destination-type kinesis-data-firehose \
  --log-destination arn:aws:firehose:us-
east-1:123456789012:deliverystream/flowlogs_stream \
  --deliver-logs-permission-arn arn:aws:iam::source-account:role/mySourceRole \
  --deliver-cross-account-role arn:aws:iam::destination-account:role/
AWSLogDeliveryFirehoseCrossAccountRole
```

전송 스트림에 대해 구성한 대상에서 흐름 로그 데이터를 가져올 수 있습니다.

Amazon Athena를 사용하여 흐름 로그 쿼리

Amazon Athena는 표준 SQL을 사용하여 흐름 로그와 같은 Amazon S3의 데이터를 분석할 수 있는 대화형 쿼리 서비스입니다. Athena를 VPC 흐름 로그와 함께 사용하여 VPC를 통해 흐르는 트래픽에 대해 유용한 인사이트를 빠르게 얻을 수 있습니다. 예를 들어 가상 프라이빗 클라우드(VPC)의 어떤 리소스가 상위 대화자인지 식별하거나 TCP 연결이 가장 많이 거부된 IP 주소를 식별할 수 있습니다.

옵션

- VPC Flow Logs와 Amazon Athena의 통합을 간소화하고 자동화할 수 있습니다. VPC를 통해 흐르는 트래픽에 대한 인사이트를 얻기 위해 필요한 AWS 리소스와 사전 정의된 쿼리를 생성하는 CloudFormation 템플릿을 생성합니다.
- Athena를 사용하여 고유한 쿼리를 생성할 수 있습니다. 자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 쿼리 실행을 참조하세요.

가격 책정

쿼리 실행에 대해 표준 Amazon Athena 요금이 발생합니다 . 반복 일정에 따라 새 파티션을 로드하는 Lambda 함수에 대해 표준 AWS Lambda 요금이 발생합니다(파티션 로드 빈도를 지정하고 시작 날짜와 종료 날짜는 지정하지 않은 경우).

미리 정의된 쿼리를 사용하려면

- 콘솔을 사용하여 CloudFormation 템플릿 생성
- AWS CLI를 사용하여 CloudFormation 템플릿 생성
- 사전 정의된 쿼리 실행

콘솔을 사용하여 CloudFormation 템플릿 생성

첫 번째 흐름 로그가 S3 버킷으로 전송된 후 CloudFormation 템플릿을 생성하고 해당 템플릿을 사용하여 스택을 생성함으로써 Athena와 통합할 수 있습니다.

요구 사항

- 선택한 리전은 AWS Lambda 및 Amazon Athena를 지원해야 합니다.
- 선택한 리전에 Amazon S3 버킷이 있어야 합니다.
- 흐름 로그의 로그 레코드 형식에는 실행하려는 미리 정의된 특정 쿼리에서 사용하는 필드가 포함되어야 합니다.

콘솔을 사용하여 템플릿을 생성하려면

1. 다음 중 하나를 수행하세요.
    - Amazon VPC 콘솔을 엽니다. 탐색 창에서 Your VPCs를 선택한 후 해당 VPC를 선택합니다.
    - Amazon VPC 콘솔을 엽니다. 탐색 창에서 Subnets을 선택한 후 해당 서브넷을 선택합니다.

- Amazon EC2 콘솔을 엽니다. 탐색 창에서 Network Interfaces를 선택한 후 해당 네트워크 인터페이스를 선택합니다.

2. [흐름 로그(Flow logs)] 탭에서 Amazon S3에 게시하는 흐름 로그를 선택한 후 [작업(Actions)], [Athena 통합 생성(Generate Athena integration)]을 차례로 선택합니다.
3. 파티션 로드 빈도를 지정합니다. 없음(None)을 선택하는 경우 과거 날짜를 사용하여 파티션 시작 및 종료 날짜를 지정해야 합니다. [매일(Daily)], [매주(Weekly)] 또는 [매월(Monthly)]을 선택하는 경우 파티션 시작 및 종료 날짜는 선택 사항입니다. 시작 날짜와 종료 날짜를 지정하지 않으면 CloudFormation 템플릿은 반복 일정에 따라 새 파티션을 로드하는 Lambda 함수를 생성합니다.
4. 생성된 템플릿을 저장할 S3 버킷을 선택하고, 쿼리 결과를 저장할 S3 버킷을 선택하거나 생성합니다.
5. [Athena 통합 생성(Generate Athena integration)]을 선택합니다.
6. (선택 사항) 성공 메시지에서 링크를 선택하여 CloudFormation 템플릿용으로 지정한 버킷으로 이동하고, 템플릿을 사용자 지정합니다.
7. 성공 메시지에서 [CloudFormation 스택 생성(Create CloudFormation stac)]을 선택하여 CloudFormation 콘솔에서 [스택 생성(Create Stack)] 마법사를 엽니다. 생성된 CloudFormation 템플릿의 URL은 템플릿(Template) 섹션에 명시되어 있습니다. 마법사를 완료하여 템플릿에 지정된 리소스를 생성합니다.

CloudFormation 템플릿으로 생성된 리소스

- Athena 데이터베이스. 데이터베이스 이름은 vpcflowlogsathenadatabase<flow-logs-subscription-id> 입니다.
- Athena 작업 그룹. 작업 그룹 이름은 <flow-log-subscription-id><partition-load-frequency><start-date><end-date>workgroup입니다.
- 흐름 로그 레코드에 해당하는 분할된 Athena 테이블. 테이블 이름은 <flow-log-subscription-id><partition-load-frequency><start-date><end-date>입니다.
- Athena 명명된 쿼리 세트. 자세한 내용은 사전 정의된 쿼리 단원을 참조하세요.
- 지정된 일정(매일, 매주 또는 매월)에 따라 테이블에 새 파티션을 로드하는 Lambda 함수입니다.
- Lambda 함수를 실행할 권한을 부여하는 IAM 역할입니다.

AWS CLI를 사용하여 CloudFormation 템플릿 생성

첫 번째 흐름 로그가 S3 버킷으로 전송된 후 CloudFormation 템플릿을 생성하고 해당 템플릿을 사용하여 Athena와 통합할 수 있습니다.

다음 get-flow-logs-integration-template 명령을 사용하여 CloudFormation 템플릿을 생성합니다.

```
aws ec2 get-flow-logs-integration-template --cli-input-json file://config.json
```

다음은 config.json 파일의 예입니다.

```
{
    "FlowLogId": "fl-12345678901234567",
    "ConfigDeliveryS3DestinationArn": "arn:aws:s3:::my-flow-logs-athena-integration/
templates/",
    "IntegrateServices": {
        "AthenaIntegrations": [
            {
                "IntegrationResultS3DestinationArn": "arn:aws:s3:::my-flow-logs-
analysis/athena-query-results/",
                "PartitionLoadFrequency": "monthly",
                "PartitionStartDate": "2021-01-01T00:00:00",
                "PartitionEndDate": "2021-12-31T00:00:00"
            }
        ]
    }
}
```

다음 create-stack 명령을 사용하여 생성된 CloudFormation 템플릿으로 스택을 생성합니다.

```
aws cloudformation create-stack --stack-name my-vpc-flow-logs --template-body file://
my-cloudformation-template.json
```

사전 정의된 쿼리 실행

생성된 CloudFormation 템플릿은 AWS 네트워크의 트래픽에 대한 의미 있는 인사이트를 빠르게 얻기 위해 실행할 수 있는 사전 정의된 쿼리 세트를 제공합니다. 스택을 생성하고 모든 리소스가 올바르게 생성되었는지 확인한 후 사전 정의된 쿼리 중 하나를 실행할 수 있습니다.

콘솔을 사용하여 사전 정의된 쿼리를 실행하려면

1. Athena 콘솔을 엽니다.
2. 탐색 창에서 쿼리 편집기(Query editor)를 선택합니다. 작업 그룹(Workgroups)에서 CloudFormation 템플릿으로 생성한 작업 그룹을 선택합니다.
3. 저장된 쿼리(Saved queries)를 선택하고 필요에 따라 파라미터를 수정한 후 쿼리를 실행합니다. 사용 가능한 사전 정의된 쿼리 목록은 사전 정의된 쿼리를 참조하세요.

4. 쿼리 결과(Query results)에서 쿼리 결과를 확인합니다.

사전 정의된 쿼리

다음은 Athena 명명된 쿼리의 전체 목록입니다. 템플릿을 생성할 때 제공되는 미리 정의된 쿼리는 흐름 로그에 대한 로그 레코드 형식의 일부인 필드에 따라 달라집니다. 따라서 템플릿에 이러한 미리 정의된 쿼리가 모두 포함되어 있지 않을 수 있습니다.

- VpcFlowLogsAcceptedTraffic - 보안 그룹 및 네트워크 ACL에 따라 허용된 TCP 연결입니다.
- VpcFlowLogsAdminPortTraffic – 관리 포트에서 요청을 처리하는 애플리케이션에서 기록된 트래픽이 가장 많은 상위 10개의 IP 주소입니다.
- VpcFlowLogsIPv4Traffic - 기록된 IPv4 트래픽의 총 바이트 수입니다.
- VpcFlowLogsIPv6Traffic - 기록된 IPv6 트래픽의 총 바이트 수입니다.
- VpcFlowLogsRejectedTCPTraffic - 보안 그룹 또는 네트워크 ACL에 따라 거부된 TCP 연결입니다.
- VpcFlowLogsRejectedTraffic - 보안 그룹 또는 네트워크 ACL에 따라 거부된 트래픽입니다.
- VpcFlowLogsSshRdpTraffic - SSH 및 RDP 트래픽입니다.
- VpcFlowLogsTopTalkers - 기록된 트래픽이 많은 IP 주소 50개입니다.
- VpcFlowLogsTopTalkersPacketLevel - 기록된 트래픽이 가장 많은 50개의 패킷 수준 IP 주소입니다.
- VpcFlowLogsTopTalkingInstances - 기록된 트래픽이 가장 많은 50개 인스턴스의 ID입니다.
- VpcFlowLogsTopTalkingSubnets - 기록된 트래픽이 가장 많은 50개 서브넷의 ID입니다.
- VpcFlowLogsTopTCPTraffic - 특정 소스 IP 주소에 대해 기록된 모든 TCP 트래픽입니다.
- VpcFlowLogsTopTCPTraffic - 기록된 바이트 수가 가장 많은 50쌍의 소스 및 대상 IP 주소입니다.
- VpcFlowLogsTotalBytesTransferredPacketLevel - 기록된 바이트 수가 가장 많은 50쌍의 패킷 수준 소스 및 대상 IP 주소입니다.
- VpcFlowLogsTrafficFrmSrcAddr - 특정 소스 IP 주소에 대해 기록된 트래픽입니다.
- VpcFlowLogsTrafficToDstAddr - 특정 대상 IP 주소에 대해 기록된 트래픽입니다.

VPC 흐름 로그 피어링의 ID 및 액세스 관리

AWS는 보안 보안 인증을 사용하여 사용자를 식별하고 AWS리소스에 대한 액세스 권한을 부여합니다. AWS Identity and Access Management(IAM)의 기능을 사용하면 보안 보안 인증을 공유하지 않고도 다른 사용자, 서비스 및 애플리케이션이 AWS리소스를 완전히 또는 제한된 방식으로 사용할 수 있습니다.

이 섹션에서는 VPC 흐름 로그를 위해 특별히 생성된 AWS 서비스 연결 역할 및 VPC 흐름 로그 서비스 연결 역할에 연결된 관리형 정책에 대해 설명합니다. AWS IAM 역할 및 정책에 대한 자세한 내용은 IAM 사용 설명서의 역할 용어 및 개념을 참조하세요.

VPC의 자격 증명 및 액세스 관리에 대한 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC의 자격 증명 및 액세스 관리를 참조하세요.

내용

- VPC 흐름 로그에 서비스 링크 역할 사용
- VPC 흐름 로그에 대한 AWS 관리형 정책

VPC 흐름 로그에 서비스 링크 역할 사용

VPC 흐름 로그는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 VPC 흐름 로그에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 VPC 흐름 로그에 의해 사전 정의되며 서비스에서 다른 AWS 서비스를 직접적으로 자동 호출하는 데 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 VPC Flow Logs를 더 쉽게 설정할 수 있습니다. VPC 흐름 로그에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의하지 않은 한 VPC 흐름 로그에서만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 삭제할 수 없기 때문에 VPC 흐름 로그 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 AWS IAM으로 작업하는 서비스를 참조하고 서비스 연결 역할 열이 예로 표시된 서비스를 찾으세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

VPC 흐름 로그에 대한 서비스 연결 역할 권한

VPC 흐름 로그는 AWSServiceRoleForVPCFlowLogs라는 서비스 연결 역할을 사용합니다. 이 서비스 연결 역할은 VPC 흐름 로그가 EventBridge 관리형 규칙을 생성 및 관리하고 사용자 대신 DescribeTag API를 직접적으로 호출하여 태그 필드가 포함된 흐름 로그 구독에서 리소스와 연결된 EC2 태그 값에 대한 업데이트를 자동으로 추적할 수 있도록 합니다.

AWSServiceRoleForVPCFlowLogs 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `vpc-flow-logs.amazonaws.com`

이름이 AWSVPCFlowLogsServiceRolePolicy인 연결 권한 정책은 VPC 흐름 로그가 지정된 리소스에 대해 다음 작업을 수행하도록 허용합니다.

- 작업: 태그 값 검증을 위한 EC2 Autoscaling 그룹에 대한 `autoscaling:DescribeTags`

  작업: 태그 값을 검증하기 위해 EC2 인스턴스 및 ElasticNetworkInterfaces에서 수행하는 `tag:GetResources`

  작업: 태그 변경 이벤트와 관련된 detail-type에 대해 소스 `aws.tag` 및 `aws.autoscaling`에서 가져온 새 관리형 규칙에 대한 `events:PutRule`

  작업: `VPCFlowLogsEC2TagsManagedRule` 및/또는 `VPCFlowLogsASGTagsManagedRule`이라는 이름으로 VPC 흐름 로그에서 생성한 관리형 규칙에 대한 `events:DeleteRule`

  작업: `VPCFlowLogsEC2TagsManagedRule` 및/또는 `VPCFlowLogsASGTagsManagedRule`이라는 이름으로 VPC 흐름 로그에서 생성한 관리형 규칙에 대한 `events:DescribeRule`

  작업: `VPCFlowLogsEC2TagsManagedRule` 및/또는 `VPCFlowLogsASGTagsManagedRule`이라는 이름으로 VPC 흐름 로그에서 생성한 관리형 규칙에 대한 `events:PutTargets`

  작업: `VPCFlowLogsEC2TagsManagedRule` 및/또는 `VPCFlowLogsASGTagsManagedRule`이라는 이름으로 VPC 흐름 로그에서 생성한 관리형 규칙에 대한 `events:RemoveTargets`

사용자, 그룹 또는 역할이 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 사용 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

VPC 흐름 로그에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS CLI 또는 AWS API에서 로그 형식의 태그 필드 및 연결된 TagFieldSpecifications 파라미터와 함께 CreateFlowLogs를 사용하면 VPC 흐름 로그에서 서비스 연결 역할을 자동으로 생성합니다.

> **Important**
>
> 이러한 서비스 연결 역할은 해당 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 자세한 내용은 내 AWS 계정에 표시되는 새 역할을 참조하세요.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. 로그 형식의 태그 필드 및 연결된 TagFieldSpecifications 파라미터와 함께 CreateFlowLogs를 사용하면 VPC 흐름 로그에서 서비스 연결 역할을 자동으로 생성합니다.

IAM 콘솔에서 AWSServiceRoleForVPCFlowLogs 사용 사례를 사용하여 서비스 연결 역할을 생성할 수도 있습니다. AWS CLI 또는 AWS API에서 `vpc-flow-logs.amazonaws.com` 서비스 이름의 서비스 연결 역할을 생성합니다. 자세한 내용은 IAM 사용자 설명서의 서비스 연결 역할 생성을 참조하세요. 이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

VPC 흐름 로그에 대한 서비스 연결 역할 편집

VPC 흐름 로그는 AWSServiceRoleForVPCFlowLogs 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하십시오.

VPC 흐름 로그에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

> **Note**
>
> 리소스를 삭제하려 할 때 VPC 흐름 로그 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForVPCFlowLogs에서 사용하는 VPC 흐름 로그 리소스를 삭제하는 방법

1. 로그 형식의 태그 필드를 사용하는 모든 VPC 흐름 로그 구독을 삭제합니다.
2. VPC 흐름 로그가 계정에 대한 모든 태그 구독이 삭제되었음을 처리하고 로그 보강을 지원하기 위해 생성된 EventBridge 관리형 규칙을 자동으로 정리할 때까지 1시간 이상 기다리세요.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면

IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 AWSServiceRoleForVPCFlowLogs 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스에 연결 역할 삭제를 참조하십시오.

VPC 흐름 로그 서비스 연결 역할이 지원되는 리전

VPC 흐름 로그는 서비스가 제공되는 모든 리전에서 서비스 연결 역할 사용을 지원하지 않습니다. 다음 리전에서 AWSServiceRoleForVPCFlowLogs 역할을 사용할 수 있습니다.

리전 이름	리전 자격 증명	VPC 흐름 로그에서 지원
미국 동부(버지니아 북부)	us-east-1	예
미국 동부(오하이오)	us-east-2	예
미국 서부(캘리포니아 북부)	us-west-1	예
미국 서부(오리건)	us-west-2	예
아프리카(케이프타운)	af-south-1	예
아시아 태평양(홍콩)	ap-east-1	예
아시아 태평양(자카르타)	ap-southeast-3	예
아시아 태평양(뭄바이)	ap-south-1	예
아시아 태평양(오사카)	ap-northeast-3	예
아시아 태평양(서울)	ap-northeast-2	예
아시아 태평양(싱가포르)	ap-southeast-1	예
아시아 태평양(시드니)	ap-southeast-2	예
아시아 태평양(도쿄)	ap-northeast-1	예
캐나다(중부)	ca-central-1	예
유럽(프랑크푸르트)	eu-central-1	예
유럽(아일랜드)	eu-west-1	예
유럽(런던)	eu-west-2	예
유럽(밀라노)	eu-south-1	예

리전 이름	리전 자격 증명	VPC 흐름 로그에서 지원
유럽(파리)	eu-west-3	예
유럽(스톡홀름)	eu-north-1	예
중동(바레인)	me-south-1	예
중동(UAE)	me-central-1	예
남아메리카(상파울루)	sa-east-1	예
AWS GovCloud (US-East)	us-gov-east-1	예
AWS GovCloud(미국 서부)	us-gov-west-1	예

VPC 흐름 로그에 대한 AWS 관리형 정책

VPC 흐름 로그를 사용하고 태그 필드와 연결된 TagFieldSpecifications 파라미터로 구독을 생성하는 경우 AWSVPCFlowLogsServiceRolePolicy 관리형 정책이 IAM 계정에서 자동으로 생성되고 AWSServiceRoleForVPCFlowLogs 서비스 연결 역할에 연결됩니다.

이 관리형 정책을 사용하면 VPC 흐름 로그가 다음을 수행할 수 있습니다.

- EventBridge 관리형 규칙을 생성하고 관리하여 VPC 흐름 로그 서비스로 태그 업데이트 이벤트를 전송합니다.
- 고객 대신 API를 직접적으로 호출하여 로그 보강을 위한 태그 값의 최신성을 검증합니다.

다음은 생성된 관리형 정책의 세부 정보를 보여주는 예입니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleOnSpecificSourcesAndDetailTypes",
            "Effect": "Allow",
            "Action": "events:PutRule",
```

```
            "Resource": [
                "arn:aws:events:*:*:rule/VPCFlowLogsEC2TagsManagedRule",
                "arn:aws:events:*:*:rule/VPCFlowLogsASGTagsManagedRule"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [
                        "aws.tag",
                        "aws.autoscaling"
                    ],
                    "events:detail-type": [
                        "AWS API Call via CloudTrail",
                        "Tag Change on Resource"
                    ]
                },
                "Null": {
                    "events:source": "false",
                    "events:detail-type": "false"
                },
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "AllowOtherOperationsOnRulesManagedByVPCFlowLogs",
            "Effect": "Allow",
            "Action": [
                "events:DeleteRule",
                "events:DescribeRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/VPCFlowLogsEC2TagsManagedRule",
                "arn:aws:events:*:*:rule/VPCFlowLogsASGTagsManagedRule"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
```

```
            "Sid": "AllowDescribeTagsOnAllEC2Resources",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources",
                "autoscaling:DescribeTags"
            ],
            "Resource": "*"
        }
    ]
}
```

앞서 제시한 예제의 첫 번째 문은 VPC 흐름 로그가 AWS 계정 내에 소스 `aws.tag` 및 `aws.autoscaling`에 대해 태그 변경 이벤트와 관련된 detail-type에 대한 EventBridge 관리형 규칙을 생성할 수 있도록 합니다.

앞서 제시한 예제의 두 번째 문은 VPC 흐름 로그가 `VPCFlowLogsEC2TagsManagedRule` 및/또는 `VPCFlowLogsASGTagsManagedRule`이라는 리소스에 대해 AWS 계정에 생성된 관리형 규칙의 수명 주기를 제어할 수 있도록 합니다.

앞서 제시한 예제의 세 번째 문은 VPC 흐름 로그가 고객 대신 태그 API를 직접적으로 호출하여 로그 보강을 위한 태그 값의 최신성을 검증할 수 있도록 합니다.

AWS 관리형 정책: AWSVPCFlowLogsServiceRolePolicy

`AWSVPCFlowLogsServiceRolePolicy` 정책을 IAM ID에 연결할 수 있습니다. 이 정책은 VPC 흐름 로그가 EventBridge 관리형 규칙을 생성 및 관리하고 사용자 대신 DescribeTag API를 직접적으로 호출하여 태그 필드가 포함된 흐름 로그 구독에서 리소스와 연결된 EC2 태그 값에 대한 업데이트를 자동으로 추적할 수 있는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSVPCFlowLogsServiceRolePolicy를 참조하세요.

AWS 관리형 정책으로 업데이트

이 서비스가 해당 변경 사항을 추적하기 시작한 이후 VPC 흐름 로그용 AWS 관리형 정책 업데이트에 대한 세부 정보를 확인합니다.

변경	설명	날짜
the section called “AWSVPCFlowLogsServiceRolePolicy ” - 새 정책	새로운 AWSVPCFlowLogsServiceRolePolicy 정책은 VPC 흐름 로그가 EventBridge 관리형 규칙을 생성 및 관리하고 사용자 대신 DescribeTag API를 직접적으로 호출하여 태그 필드가 포함된 흐름 로그 구독에서 리소스와 연결된 EC2 태그 값에 대한 업데이트를 자동으로 추적할 수 있도록 합니다.	2026년 3월 31일
VPC 흐름 로그에서 변경 사항 추적 시작	VPC 흐름 로그에서 AWS 관리형 정책에 대한 변경 내용 추적을 시작했습니다.	2026년 3월 31일

VPC 흐름 로그 문제 해결

다음은 흐름 로그로 작업할 때 발생할 수 있는 문제입니다.

문제

- 불완전한 흐름 로그 레코드
- 흐름 로그가 활성화되었지만 흐름 로그 레코드 또는 로그 그룹이 없음
- 'LogDestinationNotFoundException' 또는 'LogDestination에 대한 액세스가 거부됨' 오류
- Amazon S3 버킷 정책 제한 초과
- LogDestination 전송 불가
- 흐름 로그 데이터 크기가 결제 데이터와 일치하지 않음

불완전한 흐름 로그 레코드

문제

흐름 로그 레코드가 불완전하거나 더 이상 게시되지 않습니다.

원인

흐름 로그를 CloudWatch Logs 로그 그룹으로 전달하는 데 문제가 있거나 SkipData 항목이 제공될 수 있습니다.

Solution

흐름 로그 탭에서 VPC, 서브넷 또는 네트워크 인터페이스를 확인합니다. 공유받은 VPC 또는 서브넷의 Flow Logs 정보는 조회할 수 없습니다. 하지만 공유받은 VPC 또는 서브넷에 직접 생성한 네트워크 인터페이스의 Flow Logs 정보는 조회할 수 있습니다. 오류가 있는 경우 상태 열에 오류가 표시됩니다. 또는 describe-flow-logs 명령을 사용하여 `DeliverLogsErrorMessage` 필드에 반환된 값을 확인하세요.

다음은 상태에 대해 가능한 오류 값입니다.

- `Rate limited`: 이 오류는 특정 시간 내에 게시할 수 있는 네트워크 인터페이스의 Flow Logs 레코드 수가 최대 제한을 초과할 때 발생할 수 있습니다. 이 오류는 만들 수 있는 CloudWatch Logs 로그 그룹 수 할당량에 도달한 경우에 발생하기도 합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch Service Quotas를 참조하세요.
- `Access error`: 이 오류는 다음과 같은 이유로 발생할 수 있습니다.
  - 흐름 로그용 IAM 역할에 CloudWatch 로그 그룹에 흐름 로그를 게시할 권한이 없습니다.
  - IAM 역할이 흐름 로그 서비스와 신뢰 관계를 갖지 않습니다.
  - 신뢰 관계는 흐름 로그 서비스를 주체로 지정하지 않습니다.

  자세한 내용은 CloudWatch Logs에 흐름 로그를 게시하는 IAM 역할 단원을 참조하세요.
- `Unknown error`: 흐름 로그 서비스에서 내부 오류가 발생했습니다.

흐름 로그가 활성화되었지만 흐름 로그 레코드 또는 로그 그룹이 없음

문제

흐름 로그를 생성했고 Amazon VPC 또는 Amazon EC2 콘솔에서 흐름 로그를 `Active`로 표시합니다. 하지만 CloudWatch Logs에서 어떠한 로그 스트림도 볼 수 없거나 Amazon S3 버킷에서 로그 파일을 볼 수 없습니다.

가능한 원인

- 흐름 로그가 아직 생성되는 중입니다. 경우에 따라 Flow Logs를 생성한 후 로그 그룹이 생성되고 데이터가 표시될 때까지 10분 이상 소요될 수 있습니다.

- 네트워크 인터페이스에 대해 기록된 트래픽이 아직 없습니다. CloudWatch Logs의 로그 그룹은 트래픽이 기록될 때만 생성됩니다.

Solution

로그 그룹이 생성되거나 트래픽이 기록될 때까지 몇 분 정도 기다리세요.

'LogDestinationNotFoundException' 또는 'LogDestination에 대한 액세스가 거부됨' 오류

문제

흐름 로그를 생성할 때 `Access Denied for LogDestination` 또는 `LogDestinationNotFoundException` 오류가 발생합니다.

가능한 원인

- Amazon S3 버킷으로 데이터를 게시하는 Flow Logs를 생성할 때 이 오류가 발생하면 지정된 S3 버킷을 찾을 수 없음을 의미합니다. 또한 버킷 정책에서 해당 버킷으로의 로그 전송을 허용하지 않는 경우일 수도 있습니다.
- Amazon CloudWatch Logs에 데이터를 게시하는 흐름 로그를 생성하는 경우 이 오류는 IAM 역할이 로그를 로그 그룹에 전달할 수 없음을 나타냅니다.

Solution

- Amazon S3에 게시하는 경우 기존 S3 버킷에 ARN을 지정했는지, 그리고 그 ARN의 형식이 올바른지 확인합니다. S3 버킷을 소유하지 않은 경우 버킷 정책이 필수 권한을 보유하고 있고 ARN에서 계정 ID와 버킷 이름을 올바르게 사용하는지 확인합니다.
- CloudWatch Logs Logs에 게시하는 경우 IAM 역할에 필수 권한이 있는지 확인합니다.

Amazon S3 버킷 정책 제한 초과

문제

흐름 로그를 생성할 때 `LogDestinationPermissionIssueException` 오류가 발생합니다.

가능한 원인

Amazon S3 버킷 정책은 크기가 20KB로 제한됩니다.

Amazon S3 버킷에 게시하는 흐름 로그가 생성될 때마다 폴더 경로를 포함하는 지정된 버킷 ARN을 버킷 정책의 Resource 요소에 자동으로 추가합니다.

동일한 버킷에 게시하는 여러 개의 흐름 로그를 생성하면 버킷 정책 제한을 초과할 수 있습니다.

Solution

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

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

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

LogDestination 전송 불가

문제

흐름 로그를 생성할 때 LogDestination <bucket name> is undeliverable 오류가 발생합니다.

가능한 원인

대상 Amazon S3 버킷은 AWS KMS(SSE-KMS)로 서버 측 암호화를 사용하여 암호화되며 버킷의 기본 암호화는 KMS 키 ID입니다.

Solution

값은 KMS 키 ARN이어야 합니다. 기본 S3 암호화 유형을 KMS 키 ID에서 KMS 키 ARN으로 변경합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 기본 암호화 사용 설정을 참조하세요.

흐름 로그 데이터 크기가 결제 데이터와 일치하지 않음

문제

흐름 로그의 총 데이터 크기가 결제 데이터에 보고된 크기와 일치하지 않습니다.

가능한 원인

흐름 로그에 SKIPDATA 항목이 있을 수 있습니다. SKIPDATA 항목에 관한 설명은 데이터가 없고 건너뛴 레코드 섹션을 참조하세요.

Solution

로그 상태 필드의 다양한 항목에 대한 로그를 쿼리하여 로그 항목에 SKIPDATA 항목이 있는지 확인합니다.

SKIPDATA 확인을 위한 샘플 쿼리:

CW Insights:

```
fields @timestamp, @message, @logStream, @log
| filter interfaceId = 'eni-123'
| stats count(*) by interfaceId, logStatus
| sort by interfaceId, logStatus
```

Athena:

```
SELECT log_status, interface_id, count(1)
FROM vpc_flow_logs
WHERE interface_id IN ('eni-1', 'eni-2', 'eni-3')
GROUP BY log_status, interface_id
```

VPC의 CloudWatch 지표

Amazon VPC에서는 VPC에 대한 데이터를 Amazon CloudWatch에 게시합니다. 지표라고 알려진 정렬된 시계열 데이터 집합으로 VPC에 대한 통계를 검색할 수 있습니다. 지표를 모니터링할 변수, 데이터는 시간에 따른 해당 변수의 값으로 생각하세요. 자세한 설명은 Amazon CloudWatch 사용자 가이드를 참조하세요.

내용

- NAU 지표 및 차원
- NAU 모니터링 활성화 또는 비활성화
- NAU CloudWatch 경보 예시

NAU 지표 및 차원

네트워크 주소 사용량(NAU)은 VPC 크기에 대한 계획을 세우고 이를 모니터링하는 데 도움이 되도록 가상 네트워크의 리소스에 적용되는 지표입니다. NAU를 모니터링하는 비용은 없습니다. VPC에 대

한 NAU 또는 피어링된 NAU 할당량이 소진되는 경우 새 EC2 인스턴스를 시작하거나 새 리소스(예: Network Load Balancer, VPC 엔드포인트, Lambda 함수, Transit Gateway Attachment 또는 NAT 게이트웨이)를 프로비저닝할 수 없기 때문에 NAU 모니터링이 도움이 됩니다.

VPC에 대한 네트워크 주소 사용량 모니터링을 활성화한 경우 Amazon VPC에서는 NAU와 관련된 지표를 Amazon CloudWatch로 보냅니다. VPC의 크기는 VPC에 포함된 NAU(네트워크 주소 사용량) 단위 수로 측정합니다.

이러한 지표를 사용하여 VPC 성장 속도를 파악하거나, VPC에서 언제 크기 제한에 도달할지 예측하거나, 크기 임계값을 초과할 때 경보를 생성할 수 있습니다.

AWS/EC2네임스페이스에는 NAU 모터터링에 대한 다음과 같은 지표가 포함됩니다.

지표	설명
NetworkAddressUsage	VPC당 NAU 개수. 보고 기준 • 24시간마다. Dimensions • 이름: Per-VPC Metrics, 값: VPC ID.
NetworkAddressUsagePeered	VPC 및 피어링되는 모든 VPC에 대한 NAU 개수. 보고 기준 • 24시간마다. Dimensions • 이름: Per-VPC Metrics, 값: VPC ID.

AWS/Usage네임스페이스에는 NAU 모터터링에 대한 다음과 같은 지표가 포함됩니다.

지표	설명
ResourceCount	VPC당 NAU 개수. 보고 기준 • 24시간마다. Dimensions • 이름: Service, 값: EC2 • 이름: Type, 값: Resource • 이름: Resource, 값: VPC ID. • 이름: Class, 값: NetworkAddressUsage
ResourceCount	VPC 및 피어링되는 모든 VPC에 대한 NAU 개수. 보고 기준 • 24시간마다. Dimensions • 이름: Service, 값: EC2 • 이름: Type, 값: Resource • 이름: Resource, 값: VPC ID. • 이름: Class, 값: NetworkAddressUsagePeered
ResourceCount	VPC 전체의 NAU 사용량 결합 보기. 보고 기준 • 24시간마다.

지표	설명
	Dimensions • 이름: Service, 값: EC2 • 이름: Type, 값: Resource • 이름: Resource, 값: VPC • 이름: Class, 값: NetworkAddressUsage
ResourceCount	피어링된 VPC의 NAU 사용량 결합 보기. 보고 기준 • 24시간마다. Dimensions • 이름: Service, 값: EC2 • 이름: Type, 값: Resource • 이름: Resource, 값: VPC • 이름: Class, 값: NetworkAddressUsagePeered

NAU 모니터링 활성화 또는 비활성화

CloudWatch에서 NAU 지표를 보려면 먼저 모니터링할 각 VPC의 모니터링을 활성화해야 합니다.

NAU 모니터링을 활성화하거나 비활성화하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Your VPCs를 선택합니다.
3. VPC의 확인란을 선택합니다.
4. Actions(작업), Edit VPC settings(VPC 설정 편집)를 선택합니다.
5. 다음 중 하나를 수행합니다.

- 모니터링을 활성화하려면 Network mapping units metrics settings(네트워크 매핑 단위 지표 설정), Enable network address usage metrics(네트워크 주소 사용량 지표 활성화)를 선택합니다.
- 모니터링을 비활성화하려면 Network mapping units metrics settings(네트워크 매핑 단위 지표 설정), Enable network address usage metrics(네트워크 주소 사용량 지표 활성화)를 선택 취소합니다.

명령줄을 사용하여 모니터링을 활성화하거나 비활성화하는 방법

- modify-vpc-attribute(AWS CLI)
- Edit-EC2VpcAttribute(AWS Tools for Windows PowerShell)

NAU CloudWatch 경보 예시

다음과 같은 AWS CLI 명령과 예제 `.json`을 사용하여 임계값이 50,000 NAU인 VPC의 NAU 사용률을 추적하는 Amazon CloudWatch 경보 및 SNS 알림을 생성할 수 있습니다. 이 샘플에서는 먼저 Amazon SNS 주제를 생성해야 합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 Amazon SNS 시작하기를 참조하세요.

```
aws cloudwatch put-metric-alarm --cli-input-json file://nau-alarm.json
```

다음은 `nau-alarm.json`의 예제입니다.

```
{
    "Namespace": "AWS/EC2",
    "MetricName": "NetworkAddressUsage",
    "Dimensions": [{
        "Name": "Per-VPC Metrics",
        "Value": "vpc-0123456798"
    }],
    "AlarmActions": ["arn:aws:sns:us-west-1:123456789012:my_sns_topic"],
    "ComparisonOperator": "GreaterThanThreshold",
    "Period": 86400,
    "EvaluationPeriods": 1,
    "Threshold": 50000,
    "AlarmDescription": "Tracks NAU utilization of the VPC with 50k NAUs as the
 threshold",
    "AlarmName": "VPC NAU Utilization",
    "Statistic": "Maximum"
```

```
}
```

결제 및 사용량 보고서의 Amazon VPC 코드 이해

Amazon VPC를 사용하는 경우 AWS 결제 및 사용량 보고서에 관련 코드가 포함됩니다. 이러한 코드를 검토하면 Amazon VPC의 비용 및 사용량 패턴을 이해하는 데 도움이 됩니다. 비용 추적 및 관리는 비용 최적화에 있어 필수적인 요소입니다.

다음 표에서는 결제 및 사용량 보고서에 표시되는 Amazon VPC 코드를 설명합니다. 결제 및 사용량 보고서에 사용되는 리전 코드 목록은 AWS 리전 결제 코드를 참조하세요.

결제 코드:

- IP 주소 관리
- VPC 엔드포인트
- 전송 게이트웨이
- 네트워크 분석
- 트래픽 미러링
- VPC Lattice
- 교차 계정/리전 리소스

관련 리소스

- Amazon VPC 요금
- AWS PrivateLink 요금
- AWS Transit Gateway 요금
- Amazon VPC Lattice 요금

IP 주소 관리

코드	설명	단위	세부 수준
region-PublicIP	리소스에서 퍼블릭 IPv4 주소를 사용하는 시간.	시간	초당

코드	설명	단위	세부 수준
v4:InUseAddress			
region-PublicIPv4:IdleAddress	리소스에서 퍼블릭 IPv4 주소를 사용하지 않는 시간.	시간	초당
region-PublicIPv4:ContiguousBlock	Amazon에서 제공하는 연속 IPv4 블록에서의 퍼블릭 IPv4 주소 사용 시간.	시간	시간당
region-IPAddressManager-IP-Hours	IPAM 고급 티어에서 IP 주소를 관리하는 시간.	시간	시간당

VPC 엔드포인트

코드	설명	단위	세부 수준
region-VpcEndpoint-Hours	인터페이스 VPC 엔드포인트가 프로비저닝되는 시간.	시간	시간당
region-VpcEndpoint-Bytes	인터페이스 VPC 엔드포인트에서 처리하는 데이터.	GB	시간당
region-VpcEndpoint-GWLBE-Hours	Gateway Load Balancer 엔드포인트가 프로비저닝되는 시간.	시간	시간당

코드	설명	단위	세부 수준
region-VpcEndpoint-GWLBE-Bytes	Gateway Load Balancer 엔드포인트에서 처리하는 데이터.	GB	시간당

전송 게이트웨이

코드	설명	단위	세부 수준
region-TransitGateway-Hours	전송 게이트웨이 연결의 사용.	시간	시간당
region-TransitGateway-Bytes	전송 게이트웨이에서 처리하는 데이터.	GB	시간당
region-TGW-Multicast-Consumer-Bytes	멀티캐스트 수신기 인스턴스에서 처리하는 데이터.	GB	시간당

네트워크 분석

코드	설명	단위	세부 수준
region-Analysis-Runs	Reachability Analyzer에서 분석한 네트워크 경로 수.	개수	분석당

코드	설명	단위	세부 수준
region-NetworkInterface-Assessment	Network Access Analyzer에서 분석한 네트워크 인터페이스 수.	개수	평가당

트래픽 미러링

코드	설명	단위	세부 수준
region-ENI-Mirror	트래픽 미러링을 위해 네트워크 인터페이스가 구성된 시간.	시간	시간당

VPC Lattice

코드	설명	단위	세부 수준
region-VPCLattice-Service-Hourly	VPC Lattice 서비스의 실행 시간.	시간	시간당
region-VPCLattice-DataProcessing-Bytes	VPC Lattice 서비스에서 처리하는 데이터.	GB	시간당
region-VPCLattice-RequestCount-Free	무료 HTTP 요청 및 TCP 연결.	개수	시간당

코드	설명	단위	세부 수준
region-VpcLattice-Service-Network-Resource-Hours	VPC Lattice 서비스 네트워크의 실행 시간.	시간	시간당

교차 계정/리전 리소스

코드	설명	단위	세부 수준
region-VpcResource-Provider-Bytes	계정 또는 리전 간에 공급자 리소스에서 전송된 데이터.	GB	시간당
region-VpcResource-Consumer-Bytes	계정 또는 리전 간에 소비자 리소스가 전송한 데이터.	GB	시간당

VPC 네트워크 아키텍처 설명

Amazon VPC를 사용하면 가상 프라이빗 클라우드(VPC)로 알려져 있는 AWS 클라우드에서 논리적으로 독립된 가상 네트워크를 정의할 수 있습니다. 별도의 VPC를 생성하여 워크로드별 또는 조직체별로 인프라를 격리합니다. IP 주소 범위를 선택하고, 라우팅을 구성하고, VPC를 서로, 인터넷 또는 자체 회사 네트워크에 연결하기 위한 네트워크 게이트웨이를 추가하여 VPC를 구성할 수 있습니다. VPC에서 EC2 인스턴스 또는 RDS 인스턴스와 같은 AWS 리소스를 시작합니다.

다음 표에서는 VPC 네트워크의 주요 특성을 설명합니다. 네트워크 관리자는 이 지침을 사용하여 VPC 네트워크의 아키텍처 및 구성을 설명할 수 있습니다. 이 정보를 활용하면 온프레미스에서 또는 다른 클라우드 공급자를 사용하여 기능적으로 동등한 네트워크를 구성할 수 있습니다.

기능	설명
지리적 위치	Amazon VPC는 전 세계 모든 AWS 리전에서 호스팅됩니다. 고객에게 가장 가까운 AWS 리소스를 배치하는 VPC 네트워크의 리전을 선택할 수 있습니다.
서브넷	VPC에 대해 정의하는 서브넷은 네트워크 경계를 정의하고 AWS 리소스의 IP 주소를 결정합니다. 여러 가용 영역에 서브넷을 추가하여 리소스의 가용성을 높일 수 있습니다.
네트워크 연결	VPC 네트워크와 다른 VPC 또는 서브넷, 인터넷 또는 온프레미스 네트워크와 같은 다른 네트워크 간에 연결을 제공하기 위해 VPC 또는 서브넷에 연결하는 게이트웨이입니다.
보안 제어	VPC에 대해 생성한 보안 그룹은 컴퓨팅 리소스, 데이터베이스 리소스 및 로드 밸런서 등 관련 리소스를 오가는 트래픽을 제어합니다. 각 서브넷에는 서브넷을 통과하는 트래픽을 제어하는 네트워크 ACL이 있습니다.
트래픽 관리	라우팅 규칙은 서브넷, VPC 및 외부 위치 간의 트래픽 흐름을 제어합니다. Elastic Load Balancing에서 제공하는 로드 밸런서는 EC2 인스턴스, 컨테이너 및 Lambda 함수와 같은 여러 대상에 수신 트래픽을 분산합니다.

지리적 위치

Amazon VPC는 전 세계 모든 AWS 리전에 제공됩니다. 각 리전은 개별 지리 영역입니다. 다수의 사용자와 가까운 리전에서 리소스에 대한 VPC를 생성할 때 네트워크 지연 시간을 줄일 수 있습니다.

Amazon EC2 Global View를 사용하면 그래픽 사용자 인터페이스를 통해 모든 리전의 VPC를 나열할 수 있습니다(동등한 프로그래밍 방식 인터페이스는 없음). Amazon VPC 콘솔, AWS API 및 AWS 명령줄 인터페이스를 사용하면 각 리전의 VPC 및 VPC 리소스를 개별적으로 나열해야 합니다.

중요한 이유

VPC의 위치를 결정한 후 필요에 따라 동일한 위치 또는 다른 위치에서 기능적으로 동등한 네트워크를 구성할지 여부를 결정할 수 있습니다.

모든 리전의 VPC 요약 보기

1. https://console.aws.amazon.com/ec2globalview/home에서 Amazon EC2 Global View 콘솔을 엽니다.
2. 리전 탐색기 탭의 요약에서 VPC 수와 리전 수를 포함하는 VPC 리소스 수를 확인합니다. 여기에는 AWS가 사용자를 대신하여 생성하는 기본 VPC와 사용자가 생성하는 비기본 VPC가 모두 포함됩니다. 밑줄이 그어진 텍스트를 선택하여 여러 리전에 걸친 VPC 수의 분산 현황을 확인하세요. 리전에 VPC가 하나만 있는 경우 해당 리전의 기본 VPC일 가능성이 높습니다.
3. 전역 검색 탭에서 클라이언트 필터 리소스 유형 = Vpc를 선택합니다. 리전 또는 태그를 지정하여 결과를 추가로 필터링할 수 있습니다.

AWS CLI를 사용하여 리전의 VPC 가져오기

다음 describe-vpcs 명령을 사용합니다. VPC가 있는 각 리전에서 이 명령을 실행해야 합니다. `--query` 파라미터는 VPC ID만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-vpcs \
    --region us-east-2 \
    --query "Vpcs[*].VpcId"
```

각 리전에는 기본 VPC가 함께 제공됩니다. 기본 VPC를 사용하지 않는 경우 다음 필터를 추가하여 결과에서 제외시킬 수 있습니다.

```
--filters Name=is-default,Values=false
```

서브넷

서브넷은 VPC의 논리적 네트워크 경계입니다. 서브넷을 생성할 때 IP 주소 블록을 할당합니다. 서브넷에서 시작되는 리소스에는 서브넷의 IP 주소 블록에서 가져온 IP 주소가 할당됩니다. IP 주소를 사용하면 리소스가 로컬 네트워크 또는 인터넷을 통해 서로 통신할 수 있습니다.

Amazon VPC 콘솔의 리소스 맵을 통해 VPC의 서브넷을 시각적으로 확인할 수 있습니다.

중요한 이유

서브넷은 네트워크 관리자가 보안 경계를 구현하고 애플리케이션 계층 간의 트래픽을 제어할 수 있게 해 줍니다. 서브넷의 IP 주소를 기록하면 기능적으로 동등한 네트워크의 리소스가 VPC 네트워크에서와 마찬가지로 클라이언트 또는 애플리케이션과 통신할 수 있습니다.

리소스 맵을 사용하여 VPC의 서브넷 보기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs를 선택합니다.
3. VPC에 대한 확인란을 선택합니다.
4. 리소스 맵 탭을 선택합니다.
5. VPC 창에서 세부 정보 표시를 선택합니다. 서브넷 창에는 VPC의 모든 서브넷이 나열되고 해당 IP 주소 범위가 표시됩니다. 서브넷 위로 마우스 커서를 두면 연결된 라우팅 테이블과 네트워크 연결이 강조 표시됩니다. 자세한 내용은 링크를 선택하여 서브넷 세부 정보 페이지를 열어보세요.

AWS CLI를 사용하여 VPC의 서브넷 설명

다음 describe-subnets 명령을 사용합니다. `--filters` 파라미터는 지정된 VPC의 서브넷을 설명하는 검색 범위를 지정합니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-subnets \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query Subnets[*].
[SubnetId,AvailabilityZoneId,CidrBlock,Ipv6CidrBlockAssociationSet[0].Ipv6CidrBlock] \
    --output table
```

다음은 예제 출력입니다. 서브넷 ID, AZ ID, IPv4 주소 범위 및 첫 번째 IPv6 주소 범위(있는 경우) 열이 있습니다.

```
-------------------------------------------------------------------------------------
|                                  DescribeSubnets                                  |
+---------------------------+-----------+----------------+---------------------------+
|  subnet-0d2d1b81e0bc9c6d4 |  usw2-az1 |  10.0.144.0/20 |  2600:1f14:1e6e:a003::/64  |
|  subnet-0e01d500780bb7468 |  usw2-az1 |  10.0.16.0/20  |  2600:1f14:1e6e:a001::/64  |
|  subnet-0eb17d85f5dfd33b1 |  usw2-az2 |  10.0.128.0/20 |  2600:1f14:1e6e:a002::/64  |
|  subnet-0e990c67809773b19 |  usw2-az2 |  10.0.0.0/20   |  2600:1f14:1e6e:a000::/64  |
```

```
+--------------------------+----------+---------------+--------------------------+
```

네트워크 연결

Amazon VPC에서 제공하는 연결 옵션을 사용하면 여러 계정, 리전 및 원격 네트워크의 VPC에 걸쳐 네트워크를 생성할 수 있습니다.

Amazon VPC 콘솔에서 리소스 맵을 사용하여 VPC가 인터넷 게이트웨이, 외부 전용 인터넷 게이트웨이, NAT 게이트웨이 또는 게이트웨이 VPC 엔드포인트를 사용하는지 여부를 확인할 수 있습니다. 리소스 맵에는 전송 게이트웨이, 피어링 연결, 가상 프라이빗 게이트웨이 또는 사용 중인 기타 유형의 VPC 엔드포인트가 표시되지 않습니다. 콘솔, API 또는 명령줄 인터페이스를 사용하여 한 번에 하나씩 유형을 설명하여 VPC에 대한 게이트웨이 및 피어링 연결의 전체 목록을 가져올 수 있습니다.

중요한 이유

VPC 네트워크에서 제공하는 연결을 이해한 후에는 기능적으로 동등한 네트워크의 리소스가 동일한 로컬 및 원격 리소스와 통신할 수 있습니다.

리소스 맵을 사용하여 VPC의 네트워크 연결 보기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs를 선택합니다.
3. VPC에 대한 확인란을 선택합니다.
4. 리소스 맵 탭을 선택합니다.
5. VPC 창에서 세부 정보 표시를 선택합니다. 네트워크 연결 창에는 모든 인터넷 게이트웨이, 외부 전용 인터넷 게이트웨이, NAT 게이트웨이 및 게이트웨이 VPC 엔드포인트가 나열됩니다. 리소스 유형이 불분명한 경우 네트워크 연결의 링크 아이콘 위에 마우스 커서를 두고 결과 URL을 검사합니다. 이 URL은 콘솔의 리소스에 대한 링크이며, 리소스 유형 및 리소스 ID(예: internetGatewayId=igw-0123456780abcdef)를 포함합니다.

AWS CLI를 사용하여 VPC에 대한 네트워크 연결 가져오기

1. describe-internet-gateways 명령을 사용하여 지정된 리전의 인터넷 게이트웨이를 가져옵니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-internet-gateways \
    --region us-east-2 \
    --query InternetGateways[*].[Attachments[0].VpcId,InternetGatewayId] \
```

```
    --output table
```

다음은 예제 출력입니다. 열에는 VPC ID와 인터넷 게이트웨이 ID가 표시됩니다.

```
-----------------------------------------------------
|             DescribeInternetGateways              |
+-------------------------+-------------------------+
|  None                   |  igw-04c61dba10EXAMPLE  |
|  vpc-0bf4c2739bEXAMPLE  |  igw-09737a4029EXAMPLE  |
|  vpc-060415a18fEXAMPLE  |  igw-0c562bd22aEXAMPLE  |
|  vpc-0ea9d41094EXAMPLE  |  igw-0e06f7033dEXAMPLE  |
|  vpc-03b86de356EXAMPLE  |  igw-0a9ff72d05EXAMPLE  |
+-------------------------+-------------------------+
```

2. describe-egress-only-internet-gateways 명령을 사용하여 지정된 리전의 외부 전용 인터넷 게이트웨이를 가져옵니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-egress-only-internet-gateways \
    --region us-east-2 \
    --query EgressOnlyInternetGateways[*].
[Attachments[0].VpcId,EgressOnlyInternetGatewayId] \
    --output table
```

다음은 예제 출력입니다. 열에는 VPC ID와 외부 전용 인터넷 게이트웨이 ID가 표시됩니다.

```
-----------------------------------------------------
|        DescribeEgressOnlyInternetGateways         |
+-------------------------+-------------------------+
|  vpc-060415a18fEXAMPLE  |  eigw-0b8ca558acEXAMPLE  |
+-------------------------+-------------------------+
```

3. describe-nat-gateways 명령을 사용하여 지정된 리전의 NAT 게이트웨이를 가져옵니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-nat-gateways \
    --region us-east-2 \
    --query NatGateways[*].[VpcId,NatGatewayId,SubnetId] \
    --output table
```

다음은 예제 출력입니다. 열에는 VPC ID, NAT 게이트웨이 ID 및 서브넷 ID가 표시됩니다.

```
-------------------------------------------------------------------------------
|                            DescribeNatGateways                              |
+------------------------+------------------------+---------------------------+
|  vpc-060415a18fEXAMPLE |  nat-026316334aEXAMPLE  |  subnet-0eb17d85f5EXAMPLE  |
|  vpc-060415a18fEXAMPLE |  nat-0f08bc5f52EXAMPLE  |  subnet-0d2d1b81e0EXAMPLE  |
+------------------------+------------------------+---------------------------+
```

4. describe-transit-gateway-vpc-attachments 명령을 사용하여 지정된 리전의 전송 게이트웨이 VPC 연결을 가져옵니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-transit-gateway-vpc-attachments \
    --region us-east-2 \
    --query TransitGatewayVpcAttachments[*].
[VpcId,TransitGatewayId,length(SubnetIds[])] \
    --output table
```

다음은 예제 출력입니다. 열에는 VPC ID, 전송 게이트웨이 ID 및 서브넷 수가 표시됩니다.

```
-----------------------------------------------------------
|         DescribeTransitGatewayVpcAttachments          |
+------------------------+------------------------+----+
|  vpc-0bf4c2739bEXAMPLE |  tgw-055dc4e47bEXAMPLE  |  4 |
|  vpc-0ea9d41094EXAMPLE |  tgw-055dc4e47bEXAMPLE  |  2 |
+------------------------+------------------------+----+
```

5. describe-vpc-peering-connections 명령을 사용하여 지정된 리전의 VPC 피어링 연결을 가져옵니다. `--query` 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-vpc-peering-connections \
    --region us-east-2 \
    --query VpcPeeringConnections[*].[AccepterVpcInfo.VpcId,RequesterVpcInfo.VpcId]
 \
    --output table
```

다음은 예제 출력입니다. 열에는 수락자 VPC ID, 수락자 VPC 소유자, 요청자 VPC ID 및 요청자 VPC 소유자가 표시됩니다.

```
-------------------------------------------------------------------------------
```

```
|                          DescribeVpcPeeringConnections
 |
+------------------------+--------------+------------------------+---------------
+
|  vpc-0ea9d41094EXAMPLE |  123456789012 |  vpc-03b86de356EXAMPLE |  123456789012
 |
+------------------------+--------------+------------------------+---------------
+
```

6. describe-vpn-gateways 명령을 사용하여 지정된 리전의 가상 프라이빗 게이트웨이를 가져옵니다. --query 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-vpn-gateways \
    --region us-east-2 \
    --query VpnGateways[*].[VpcAttachments[0].VpcId,VpnGatewayId] \
    --output table
```

다음은 예제 출력입니다. 열에는 VPC ID와 가상 프라이빗 게이트웨이 ID가 표시됩니다.

```
-----------------------------------------------------
|                DescribeVpnGateways                |
+------------------------+--------------------------+
|  vpc-0bf4c2739bEXAMPLE |  vgw-0cb3226c4aEXAMPLE   |
+------------------------+--------------------------+
```

7. describe-vpc-endpoints 명령을 사용하여 지정된 리전의 VPC 엔드포인트를 가져옵니다. --query 파라미터는 지정된 필드만 출력에 포함합니다. 필요에 따라 추가 필드를 포함할 수 있습니다.

```
aws ec2 describe-vpc-endpoints \
    --region us-east-2 \
    --query 'VpcEndpoints[*].[VpcId,VpcEndpointType,ServiceName||
ServiceNetworkArn||ResourceConfigurationArn]' \
    --output table
```

다음은 예제 출력입니다. 첫 번째 열에는 VPC ID가 표시되고, 두 번째 열에는 VPC 엔드포인트 유형이 표시됩니다. 세 번째 열은 엔드포인트 유형에 따라 다르고, 서비스 이름, 리소스 구성 ARN 또는 서비스 네트워크 ARN이 표시됩니다.

```
-------------------------------------------------------------------------------------
```

```
|                                                                    DescribeVpcEndpoints
                                                                  |
+------------------------+-----------------
+-------------------------------------------------------------------------------------------
+
|  vpc-060415a18fcc9afde |  Interface      |  com.amazonaws.vpce.us-west-2.vpce-
svc-007832a03d60fc387                                  |
|  vpc-060415a18fcc9afde |  Interface      |  com.amazonaws.vpce.us-west-2.vpce-
svc-007832a03d60fc387                                  |
|  vpc-0bf4c2739bc05a694 |  Gateway        |  com.amazonaws.us-west-2.s3
                                                       |
|  vpc-0ea9d410947d27b7d |  Interface      |  com.amazonaws.us-west-2.logs
                                                       |
|  vpc-0bf4c2739bc05a694 |  Resource       |  arn:aws:vpc-lattice:us-
east-2:123456789012:resourceconfiguration/rcfg-07129f3acded87625  |
|  vpc-0bf4c2739bc05a694 |  ServiceNetwork |  arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetwork/sn-0808d1748faee0c1e           |
|  vpc-0bf4c2739bc05a694 |  ServiceNetwork |  arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetwork/sn-0808d1748faee0c1e           |
+------------------------+-----------------
+-------------------------------------------------------------------------------------------
+
```

보안 제어

Amazon VPC에서 제공하는 보안 제어에 따라 VPC 및 VPC에 배포된 리소스에 대한 네트워크 액세스가 결정됩니다.

중요한 이유

서브넷 및 리소스에 도달할 수 있는 인바운드 트래픽과 서브넷 및 리소스에서 나갈 수 있는 출력 트래픽을 확인한 후 기능적으로 동등한 네트워크에 필요한 방화벽 규칙을 계획할 수 있습니다.

보안 제어

- 보안 그룹
- 네트워크 ACL

보안 그룹

보안 그룹은 리소스 수준에서 특정 인바운드 및 아웃바운드 트래픽을 허용합니다. 보안 그룹은 VPC의 리소스에 대한 액세스를 제어하는 기본 메커니즘입니다.

VPC의 보안 그룹 가져오기

describe-security-groups 명령을 사용하면 지정된 VPC의 보안 그룹이 표시됩니다.

```
aws ec2 describe-security-groups \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query SecurityGroups[*].GroupId
```

보안 그룹의 인바운드 규칙 가져오기

describe-security-group-rules 명령을 사용하면 `IsEgress`가 `false`인 지정된 보안 그룹의 규칙이 표시됩니다.

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-0abcdef1234567890 \
    --query 'SecurityGroupRules[?IsEgress==`false`]'
```

보안 그룹의 아웃바운드 규칙 가져오기

describe-security-group-rules 명령을 사용하면 `IsEgress`가 `true`인 지정된 보안 그룹의 규칙이 표시됩니다.

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-0abcdef1234567890 \
    --query 'SecurityGroupRules[?IsEgress==`true`]'
```

네트워크 ACL

네트워크 액세스 제어 목록(ACL)은 서브넷 수준에서 특정 인바운드 및 아웃바운드 트래픽을 허용하거나 거부합니다. 올바른 보안 그룹 없이 리소스가 배포되는 경우 네트워크 ACL을 심층 방어로 사용할 수 있습니다.

서브넷의 네트워크 ACL 가져오기

describe-network-acls 명령을 사용하면 지정된 VPC와 서브넷 연결의 네트워크 ACL이 표시됩니다.

```
aws ec2 describe-network-acls \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query "NetworkAcls[*].{ID:NetworkAclId,Subnets:Associations[].SubnetId}"
```

네트워크 ACL의 인바운드 규칙 가져오기

describe-network-acls 명령을 사용하면 Egress가 false인 지정된 네트워크 ACL의 규칙이 표시됩니다.

```
aws ec2 describe-network-acls \
    --network-acl-ids acl-0abcdef1234567890 \
    --query 'NetworkAcls[*].Entries[?Egress==`false`]'
```

네트워크 ACL의 아웃바운드 규칙 가져오기

describe-network-acls 명령을 사용하면 Egress가 true인 지정된 네트워크 ACL의 규칙이 표시됩니다.

```
aws ec2 describe-network-acls \
    --network-acl-ids acl-0abcdef1234567890 \
    --query 'NetworkAcls[*].Entries[?Egress==`true`]'
```

트래픽 관리

효과적인 트래픽 관리는 라우팅 테이블에서 제공하는 네트워크 수준 라우팅 결정과 로드 밸런싱에서 제공하는 애플리케이션 수준 배포 전략을 결합합니다.

중요한 이유

네트워크 관리자는 보안 경계 및 성능 요구 사항을 유지하면서 트래픽 흐름을 최적화하기 위해 서브넷, 라우팅, DNS 확인 및 로드 밸런싱을 설계해야 합니다. VPC 네트워크에서 이러한 구성 요소의 구성을 기록하면 기능적으로 동등한 네트워크의 리소스가 VPC 네트워크에서와 동일한 클라이언트 또는 디바이스와 통신할 수 있습니다.

트래픽 관리

- 라우팅 테이블
- DHCP 옵션 세트
- 로드 밸런서

라우팅 테이블

라우팅 테이블은 서브넷, VPC, 온프레미스 네트워크 및 인터넷 등의 네트워크 경계를 오가며 네트워크 트래픽이 흐르는 방식을 결정합니다.

Amazon VPC 콘솔의 리소스 맵을 통해 VPC의 라우팅 테이블을 시각적으로 확인할 수 있습니다.

리소스 맵을 사용하여 VPC의 라우팅 테이블 보기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 VPCs를 선택합니다.
3. VPC에 대한 확인란을 선택합니다.
4. 리소스 맵 탭을 선택합니다.
5. 라우팅 테이블 창에는 VPC의 모든 라우팅 테이블이 나열됩니다. 라우팅 테이블 위로 마우스 커서를 두면 연결된 서브넷과 네트워크 연결이 강조 표시됩니다. 자세한 내용은 링크를 선택하여 라우팅 테이블 세부 정보 페이지를 열어보세요.

라우팅 테이블 설명

describe-route-tables 명령을 사용하여 지정된 VPC 및 해당 서브넷 연결의 라우팅 테이블을 설명합니다.

```
aws ec2 describe-route-tables \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query "RouteTables[*].{ID:RouteTableId,Subnets:Associations[].SubnetId}"
```

라우팅 테이블의 경로 가져오기

describe-route-tables 명령을 사용하여 지정된 라우팅 테이블의 라우팅을 설명합니다.

```
aws ec2 describe-route-tables \
    --route-table-ids rtb-02ec01715bEXAMPLE \
    --query RouteTables[*].Routes
```

DHCP 옵션 세트

VPC에는 다양한 네트워크 설정을 구성하는 데 사용할 수 있는 DHCP 옵션 세트가 있습니다. 예를 들어, EC2 인스턴스가 기존 DNS 인프라를 사용하여 내부 호스트 이름을 확인할 수 있도록 사용자 지정

DNS 서버를 구성할 수 있습니다. 자세한 내용은 the section called “DHCP 옵션 세트 개념” 섹션을 참조하세요.

사용하는 VPC의 DHCP 옵션 설명

describe-dhcp-options 명령을 사용하여 지정된 DHCP 옵션을 설명합니다. 이 예제에서는 describe-vpcs 명령을 사용하여 지정된 VPC에 대한 DHCP 옵션의 ID 또한 가져옵니다.

```
aws ec2 describe-dhcp-options \
    --dhcp-options-id "$(aws ec2 describe-vpcs \
        --vpc-id vpc-1234567890abcdef0 \
        --query Vpcs[].DhcpOptionsId --output text)"
```

다음은 기본 DHCP 옵션을 사용하는 VPC 출력의 예제입니다.

```
{
    "DhcpOptions": [
        {
            "OwnerId": "415546850671",
            "Tags": [],
            "DhcpOptionsId": "dopt-1234567890abcdef0",
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-west-2.compute.internal"
                        }
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
                        }
                    ]
                }
            ]
        }
    ]
}
```

로드 밸런서

로드 밸런싱은 클라이언트에서 들어오는 트래픽을 여러 대상에 분산합니다. 로드 밸런서는 정상적인 대상만 사용되도록 대상의 상태를 모니터링하고 트래픽 분산에서 비정상적인 대상을 자동으로 제거합니다. 이를 통해 애플리케이션의 가용성 및 성능이 향상되고, 리소스 사용률이 최적화됩니다. 자세한 내용은 Elastic Load Balancing User Guide를 참조하세요.

로드 밸런서를 설명하려면

describe-load-balancers 명령을 사용하면 지정된 VPC의 로드 밸런서가 표시됩니다.

```
aws elbv2 describe-load-balancers \
    --query 'LoadBalancers[?VpcId==`vpc-1234567890abcdef0`].LoadBalancerArn'
```

관련 리소스

다음은 VPC 네트워크에서 사용할 수 있는 선택적 서비스 또는 기능입니다.

- Direct Connect
- AWS Network Firewall
- IPAM
- 트래픽 미러링
- VPC 흐름 로그

Amazon Virtual Private Cloud에 대한 보안 책임 관리

AWS은 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객으로서 여러분은 가장 높은 보안 요구 사항을 충족하기 위해 설계된 데이터 센터 및 네트워크 아키텍처의 혜택을 받게 됩니다.

보안은 AWS와 여러분의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안: AWS는 AWS클라우드에서 AWS서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS는 안전하게 사용할 수 있는 서비스 또한 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon Virtual Private Cloud에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하세요.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon VPC를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon VPC를 구성하는 방법을 보여줍니다. 또한 Amazon VPC 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배우게 됩니다.

내용

- Amazon Virtual Private Cloud에서 데이터 보호 보장
- 전송 중 VPC 암호화 적용
- Amazon VPC용 자격 증명 및 액세스 관리
- Amazon VPC의 인프라 보안
- 보안 그룹을 사용하여 AWS 리소스에 대한 트래픽 제어
- 네트워크 액세스 제어 목록으로 서브넷 트래픽 제어
- Amazon Virtual Private Cloud에서의 복원성
- Amazon Virtual Private Cloud에 대한 규정 준수 확인
- VPC 및 서브넷에 대한 퍼블릭 액세스 차단
- VPC에 대한 보안 모범 사례

Amazon Virtual Private Cloud에서 데이터 보호 보장

AWS 공동 책임 모델 은 Amazon Virtual Private Cloud의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ을 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 일반 데이터 보호 규정(GDPR) 센터를 참조하세요.

데이터를 보호하려면 AWS 계정자격 증명을 보호하고 AWS IAM Identity Center또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail으로 API 및 사용자 활동 로깅을 설정하세요. AWS 활동 캡처에 CloudTrail 추적을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스내의 모든 기본 보안 컨트롤과 함께 사용하세요.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon VPC 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

Amazon VPC에서 인터네트워크 트래픽 프라이버시 보장

Amazon Virtual Private Cloud는 가상 프라이빗 클라우드(VPC)의 보안을 강화하고 모니터링하는 데 사용할 수 있는 여러 기능을 제공합니다.

- 보안 그룹: 보안 그룹은 리소스 수준(예: EC2 인스턴스)에서 특정 인바운드 및 아웃바운드 트래픽을 허용합니다. 인스턴스를 시작할 때 하나 이상의 보안 그룹과 연결할 수 있습니다. VPC의 각 인스턴스는 서로 다른 보안 그룹 세트에 속할 수 있습니다. 인스턴스를 시작할 때 보안 그룹을 지정하지 않을 경우 해당 VPC에 대해 인스턴스는 기본 보안 그룹과 자동으로 연결됩니다. 자세한 내용은 보안 그룹 단원을 참조하세요.
- 네트워크 액세스 제어 목록(ACL): 네트워크 ACL은 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용하거나 거부합니다. 자세한 내용은 네트워크 액세스 제어 목록으로 서브넷 트래픽 제어 단원을 참조하세요.
- 흐름 로그: 흐름 로그는 VPC의 네트워크 인터페이스에서 양방향으로 이동하는 IP 트래픽에 대한 정보를 캡처합니다. VPC, 서브넷 또는 개별 네트워크 인터페이스에 대한 흐름 로그를 생성할 수 있습니다. 흐름 로그 데이터는 CloudWatch Logs 또는 Amazon S3에 게시되며 과도하게 제한하거나 과도하게 허용하는 보안 그룹과 네트워크 ACL 규칙을 진단하는 데 도움이 됩니다. 자세한 내용은 VPC 흐름 로그를 사용하여 IP 트래픽 로깅 단원을 참조하세요.
- 트래픽 미러링: Amazon EC2 인스턴스의 탄력적 네트워크 인터페이스에서 네트워크 트래픽을 복사할 수 있습니다. 그런 다음 트래픽을 대역 외 보안 및 모니터링 어플라이언스에 보낼 수 있습니다. 자세한 내용은 트래픽 미러링 안내서를 참조하세요.

전송 중 VPC 암호화 적용

VPC 암호화 제어는 트래픽 흐름의 암호화 상태를 모니터링하기 위한 신뢰할 수 있는 중앙 집중식 제어를 제공하고, 일반 텍스트 통신을 허용하는 리소스를 식별하고, 최종적으로 리전의 VPC 내부 및 전체에서 전송 중 암호화를 적용하는 메커니즘을 제공하는 보안 및 규정 준수 기능입니다.

VPC 암호화 제어는 애플리케이션 계층 암호화와 AWS Nitro System 하드웨어의 내장 전송 중 암호화 기능을 모두 사용하여 암호화 적용을 보장합니다. 또한 이 기능은 기본 하드웨어 계층 암호화를 최신 Nitro 인스턴스를 넘어 Fargate, Application Load Balancer, Transit Gateways 등의 다른 AWS 서비스로 확장합니다.

이 기능은 모든 트래픽의 암호화 상태를 파악하고 제어하려는 사용자를 위해 설계되었습니다. 특히 HIPAA, FedRamp, PCI DSS와 같은 규정 준수 표준을 충족하기 위해 데이터 암호화가 중요한 산업에서 유용합니다. 보안 관리자와 클라우드 아키텍트는 이를 사용하여 AWS 환경 전체에서 전송 중 암호화 정책을 중앙에서 적용할 수 있습니다.

이 기능은 모니터링 모드와 적용 모드의 두 가지 모드에서 사용할 수 있습니다.

암호화 제어 모드

모니터링 모드

모니터링 모드에서 암호화 제어는 VPC 내부 및 VPC 전체 AWS 리소스 간 트래픽 흐름의 암호화 상태에 대한 가시성을 제공합니다. 또한 전송 중 암호화를 적용하지 않는 VPC 리소스를 식별하는 데도 도움이 됩니다. 트래픽이 암호화되었는지 여부를 알려주는 보강 필드(encryption-status)를 내보내도록 VPC 흐름 로그를 구성할 수 있습니다. 콘솔 또는 GetVpcResourcesBlockingEncryptionEnforcement 명령을 사용하여 전송 중 암호화를 적용하지 않는 리소스를 식별할 수도 있습니다.

> **Note**
>
> 기존 VPC는 먼저 모니터링 모드에서만 활성화할 수 있습니다. 이를 통해 일반 텍스트 트래픽을 허용하거나 허용할 수 있는 리소스를 파악할 수 있습니다. 이러한 리소스가 암호화를 적용하기 시작(또는 해당 리소스에 대한 제외 항목을 생성)한 후에만 VPC에서 적용 모드를 켤 수 있습니다.

적용 모드

적용 모드에서는 VPC 암호화 제어가 VPC 경계 내에서 암호화되지 않은 트래픽을 허용하는 기능 또는 서비스를 사용하지 못하도록 합니다. 기존 VPC에서는 적용 모드에서 암호화 제어를 직접 활성화할 수 없습니다. 먼저 모니터링 모드에서 암호화 제어를 켜고, 규정을 준수하지 않는 리소스를 식별 및 수정하여 전송 중 암호화를 적용한 다음 적용 모드를 켜야 합니다. 그러나 새 VPC의 경우 생성 중에 적용 모드에서 암호화 제어를 켤 수 있습니다.

적용 모드가 활성화되면 기본 내장 암호화 또는 인터넷 게이트웨이 등을 지원하지 않는 이전 EC2 인스턴스와 같이 암호화되지 않은 VPC 리소스를 생성하거나 연결할 수 없습니다. 암호화가 적용된 VPC에서 규정을 준수하지 않는 리소스를 실행하려면 해당 리소스에 대한 제외 항목을 생성해야 합니다.

트래픽 흐름의 암호화 상태 모니터링

VPC 흐름 로그의 encryption-status 필드를 사용하여 VPC 내 트래픽 흐름의 암호화 상태를 감사할 수 있습니다. 다음과 같은 값을 가질 수 있습니다.

- 0 = 암호화되지 않음
- 1 = Nitro로 암호화됨(VPC 암호화 제어에서 관리)
- 2 = 애플리케이션으로 암호화됨
  - AWS 서비스에 대한 인터페이스 엔드포인트의 TCP 포트 443에서의 흐름 *
  - 게이트웨이 엔드포인트의 TCP 포트 443에서의 흐름 *

  - VPC 엔드포인트를 통한 암호화된 Redshift 클러스터로의 트래픽 흐름 **
- 3 = Nitro와 애플리케이션 모두로 암호화됨
- (-) = 암호화 상태 알 수 없음 또는 VPC 암호화 제어 꺼짐

참고:

* 인터페이스 및 게이트웨이 엔드포인트의 경우 AWS는 패킷 데이터를 검사하여 암호화 상태를 확인하지 않으며, 대신 사용된 포트를 기반으로 암호화 상태를 판단합니다.

** 지정된 AWS 관리형 엔드포인트의 경우 AWS는 서비스 구성의 TLS 요구 사항에 따라 암호화 상태를 확인합니다.

VPC 흐름 로그 제한 사항

- VPC 암호화 제어에 대한 흐름 로그를 활성화하려면 encryption-status 필드가 포함된 새 흐름 로그를 수동으로 생성해야 합니다. encryption-status 필드는 기존 흐름 로그에 자동으로 추가되지 않습니다.
- 흐름 로그에 더 자세한 내용을 포함하려면 흐름 로그에 ${traffic-path} 및 ${flow-direction} 필드를 추가하는 것이 좋습니다.

  예제:

```
aws ec2 create-flow-logs \
--resource-type VPC \
--resource-ids vpc-12345678901234567 \
--traffic-type ALL \
--log-group-name my-flow-logs \
--deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
--log-format '${encryption-status} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
 ${protocol} ${traffic-path} ${flow-direction} ${reject-reason}'
```

VPC 암호화 제어 제외 항목

VPC 암호화 제어 적용 모드에서는 VPC의 모든 리소스에 암호화를 적용해야 합니다. 이렇게 하면 리전의 AWS 내에서 암호화가 보장됩니다. 그러나 사용자가 종단 간 암호화를 구성하고 유지 관리하는 AWS의 네트워크 외부에서 연결을 허용하는 인터넷 게이트웨이, NAT 게이트웨이 또는 가상 프라이빗 게이트웨이와 같은 리소스가 있을 수 있습니다. 암호화가 적용된 VPC에서 이러한 리소스를 실행하기 위해 리소스 제외 항목을 생성할 수 있습니다. 제외 항목은 고객이 암호화 유지 관리(일반적으로 애플리케이션 계층)를 담당하는 리소스에 대해 감사 가능한 예외를 생성합니다.

VPC 암호화 제어에서 지원되는 제외 항목은 단 8개입니다. VPC에 이러한 리소스가 있고 적용 모드로 전환하려는 경우, 모니터링 모드에서 적용 모드로 전환할 때 이러한 제외 항목을 추가해야 합니다. 다른 리소스는 제외할 수 없습니다. 이러한 리소스에 대한 제외 항목을 생성하여 VPC를 적용 모드로 마이그레이션할 수 있습니다. 이러한 리소스 간 트래픽 흐름의 암호화에 대한 책임은 사용자에게 있습니다.

- 인터넷 게이트웨이
- NAT 게이트웨이
- 외부 전용 인터넷 게이트웨이
- 암호화가 적용되지 않은 VPC에 대한 VPC 피어링 연결(자세한 시나리오는 VPC 피어링 지원 섹션 참조)
- 가상 프라이빗 게이트웨이
- VPC 내 Lambda 함수
- VPC Lattice
- Elastic File System

워크플로 구현

1. 모니터링 활성화 - 모니터링 모드에서 VPC 암호화 제어를 생성합니다.
2. 트래픽 분석 - 흐름 로그를 검토하여 트래픽 흐름의 암호화 상태를 모니터링합니다.
3. 리소스 분석 - 콘솔 또는 `GetVpcResourcesBlockingEncryptionEnforcement` 명령을 사용하여 전송 중 암호화를 적용하지 않는 리소스를 식별합니다.
4. 준비 [선택 사항] - 적용 모드를 켜려는 경우 리소스 마이그레이션 및 필수 제외 항목을 계획합니다.
5. 적용 [선택 사항] - 필수 제외 항목이 구성된 적용 모드로 전환합니다.
6. 감사 - 흐름 로그를 통해 지속적인 규정 준수 모니터링을 수행합니다.

자세한 설정 지침은 블로그 VPC 암호화 제어 기능 소개: 리전 내 VPC 간 및 VPC 내 전송 중 암호화 적용을 참조하세요.

VPC 암호화 제어 상태

VPC 암호화 제어는 다음 상태 중 하나일 수 있습니다.

creating

VPC 암호화 제어가 VPC에서 생성되고 있습니다.

modify-in-progress

VPC 암호화 제어가 VPC에서 수정되고 있습니다.

삭제 중

VPC 암호화 제어가 VPC에서 삭제되고 있습니다.

사용 가능

VPC 암호화 제어가 VPC에 모니터링 모드 또는 적용 모드를 성공적으로 구현했습니다.

AWS 서비스 지원 및 호환성

암호화 규정을 준수하려면 리소스가 하드웨어 계층 또는 애플리케이션 계층에서 전송 중 암호화를 항상 적용해야 합니다. 대부분의 리소스의 경우 별도의 조치가 필요하지 않습니다.

자동 규정 준수를 제공하는 서비스

PrivateLink에서 지원하는 대부분의 AWS 서비스(교차 리전 PrivateLinks 포함)는 애플리케이션 계층에서 암호화된 트래픽을 허용합니다. 이러한 리소스를 변경할 필요는 없습니다. AWS는 애플리케이션 계층 암호화되지 않은 트래픽을 자동으로 차단합니다. Redshift 클러스터(프로비저닝 및 서버리스 모두 - 기본 리소스를 수동으로 마이그레이션해야 함) 등 몇 가지 예외가 있습니다.

자동으로 마이그레이션되는 리소스

Network Load Balancer, Application Load Balancer, Fargate 클러스터, EKS 컨트롤 플레인은 모니터링 모드를 켜면 암호화를 기본적으로 지원하는 하드웨어로 자동으로 마이그레이션됩니다. 이러한 리소스를 수정할 필요는 없습니다. AWS는 마이그레이션을 자동으로 처리합니다.

수동 마이그레이션이 필요한 리소스

특정 VPC 리소스 및 서비스에서는 기본 인스턴스 유형을 선택해야 합니다. 모든 최신 EC2 인스턴스는 전송 중 암호화를 지원합니다. 서비스에서 이미 최신 EC2 인스턴스를 사용하는 경우 변경할 필요가 없습니다. 콘솔 또는 GetVpcResourcesBlockingEncryptionEnforcement 명령을 사용하여 이러한 서비스에서 이전 인스턴스를 사용하고 있는지 확인할 수 있습니다. 이러한 리소스를 확인한 경우 Nitro System 하드웨어의 기본 암호화를 지원하는 최신 EC2 인스턴스로 업그레이드해야 합니다. 이러한 서비스에는 EC2 인스턴스, Auto Scaling 그룹, RDS(모든 데이터베이스 및 DocumentDB), ElastiCache

프로비저닝형, Amazon Redshift 프로비저닝형 클러스터, EKS, ECS-EC2, OpenSearch 프로비저닝형 및 EMR이 포함됩니다.

호환되는 리소스:

다음 리소스는 VPC 암호화 제어와 호환됩니다.

- Nitro 기반 EC2 인스턴스
- Network Load Balancer(제한 있음)
- Application Load Balancer
- AWS Fargate 클러스터
- Amazon Elastic Kubernetes Service(EKS)
- Amazon EC2 Auto Scaling 그룹
- Amazon Relational Database Service(RDS - 모든 데이터베이스)
- Amazon ElastiCache 노드 기반 클러스터
- Amazon Redshift 프로비저닝된 클러스터 및 서버리스 클러스터
- Amazon Elastic Container Service(ECS) - EC2 컨테이너 인스턴스
- Amazon OpenSearch Service
- Amazon Elastic MapReduce(EMR)
- Amazon Managed Streaming for Apache Kafka(Amazon MSK)
- VPC 암호화 제어는 PrivateLink를 통해 액세스하는 모든 AWS 서비스에 대해 애플리케이션 계층 수준에서 암호화를 적용합니다. 애플리케이션 계층에서 암호화되지 않은 모든 트래픽은 적용 모드에서 암호화 제어를 사용하여 VPC 내에서 호스팅되는 PrivateLink 엔드포인트에 의해 차단됩니다.

서비스별 제한 사항

Network Load Balancer 제한 사항

TLS 구성: 포함된 VPC에서 암호화 제어를 적용하는 경우, TLS 리스너를 사용하여 암호화 및 복호화 작업을 로드 밸런서로 오프로드할 수 없습니다. 그러나 TLS 암호화 및 복호화를 수행하도록 대상을 구성할 수는 있습니다.

Redshift 프로비저닝 및 서버리스

고객은 기존 클러스터/엔드포인트가 있는 VPC에서 적용 모드로 이동할 수 없습니다. Redshift에서 VPC 암호화 제어를 사용하려면 스냅샷에서 클러스터 또는 네임스페이스를 복원해야 합니다. 프로비저닝된 클러스터의 경우 기존 Redshift 클러스터의 스냅샷을 생성한 다음 클러스터 스냅샷에서 복원 작

업을 사용하여 스냅샷에서 복원하세요. 서버리스의 경우 기존 네임스페이스의 스냅샷을 생성한 다음 서버리스 작업 그룹에서 스냅샷에서 복원 작업을 사용하여 스냅샷에서 복원하세요. 스냅샷 및 복원 프로세스를 수행하지 않으면 기존 클러스터 또는 네임스페이스에서 VPC 암호화 제어를 활성화할 수 없습니다. 스냅샷 생성에 대해서는 Amazon Redshift documentation을 참조하세요.

Amazon MSK(Managed streaming for Apache Kafka)

이 기능은 자체 VPC 내 4.1 버전의 새 클러스터에서 지원됩니다. 다음 단계는 MSK에서 VPC 암호화를 사용하는 데 도움이 됩니다.

- 고객이 다른 MSK 클러스터가 없는 VPC에서 VPC 암호화를 활성화합니다.
- 고객이 Kafka 버전이 4.1이고 instancetype이 M7g인 클러스터를 생성합니다.

Gateway Load Balancer 및 AWS Network Firewall

적용 모드의 VPC에서는 Gateway Load Balancer와 AWS Network Firewall이 지원되지 않습니다. 이러한 리소스가 VPC에 있는 경우 VPC를 모니터 모드에서 실행해야 합니다.

리전 및 영역 제한 사항

- 로컬 영역 서브넷: 적용 모드에서 지원되지 않으므로 VPC에서 삭제해야 합니다.

VPC 피어링 지원

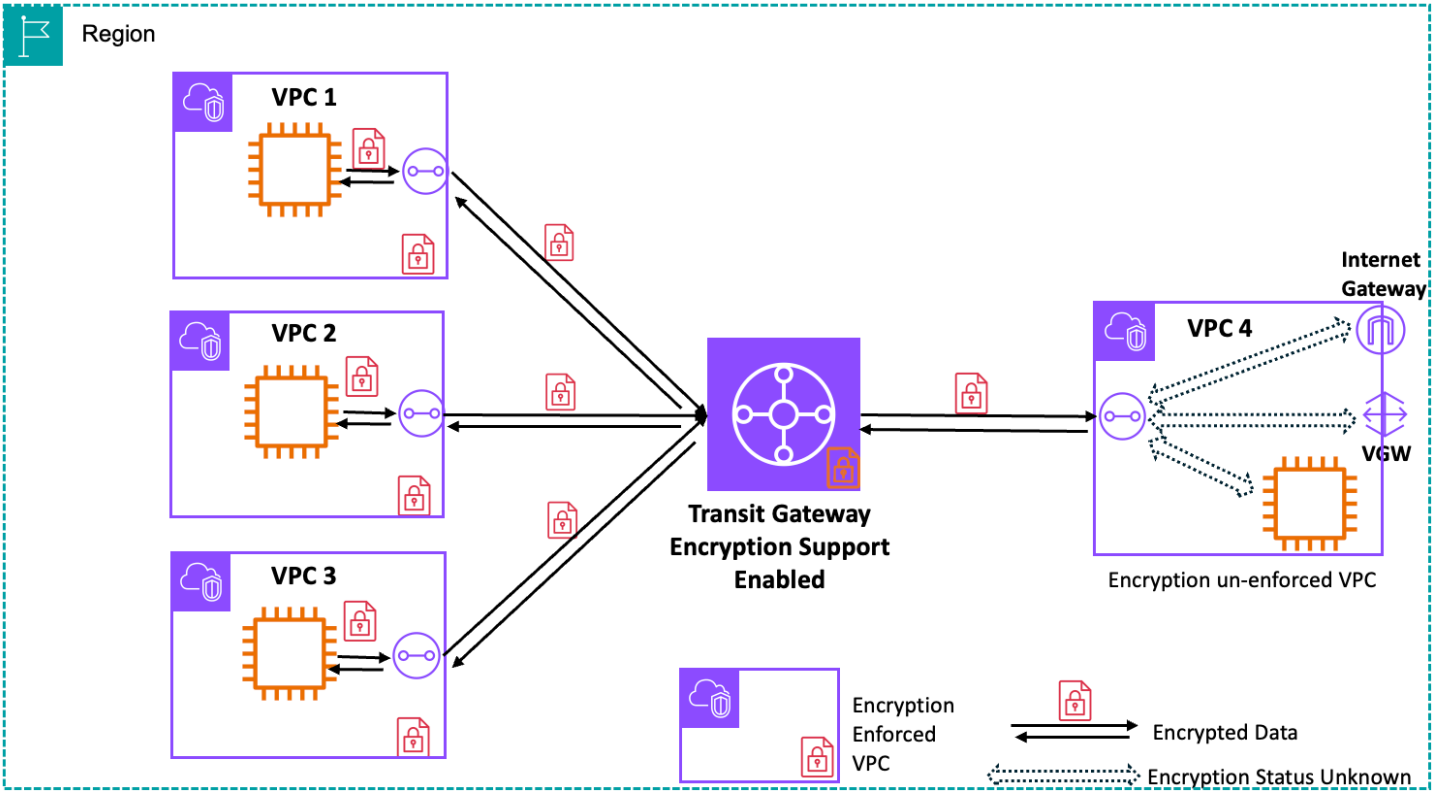
두 VPC 간의 VPC 피어링을 통한 전송 중 암호화를 보장하려면 두 VPC가 동일한 리전에 있어야 하며 제외 사항 없이 적용 모드에서 암호화 제어가 켜져 있어야 합니다. 암호화가 적용된 VPC를 다른 리전에 상주하거나 적용 모드(제외 항목 없음)에서 암호화 제어가 활성화되지 않은 다른 VPC와 피어링하려면 피어링 제외 항목을 생성해야 합니다.

두 VPC가 적용 모드이고 서로 피어링하는 경우 모드를 적용에서 모니터링으로 변경할 수 없습니다. VPC 암호화 제어 모드를 모니터링으로 수정하기 전에 먼저 피어링 제외 항목을 생성해야 합니다.

Transit Gateway 암호화 지원

암호화 제어가 켜져 있는 VPC 간의 트래픽을 암호화하려면 Transit Gateway에서 암호화 지원을 명시적으로 활성화해야 합니다. 기존 Transit Gateway에서 암호화를 활성화해도 기존 트래픽 흐름은 중단되지 않으며 VPC 연결은 암호화된 경로로 원활하게 자동으로 마이그레이션됩니다. Transit Gateway를 통한 강제 적용 모드(예외 없음)의 2개 VPC 간 트래픽은 100% 암호화된 경로를 수반합니다. 또한 Transit Gateway의 암호화를 사용하면 서로 다른 암호화 제어 모드에 있는 두 VPC도 연결할 수 있습니다. 암호화가 적용되지 않은 VPC에 연결된 VPC에서 암호화 제어를 적용하려는 경우 이를 사용해야 합

니다. 이러한 시나리오에서는 암호화가 적용된 VPC 내의 모든 트래픽(VPC 간 트래픽을 포함)이 암호화됩니다. VPC 간 트래픽은 암호화가 적용된 VPC의 리소스와 Transit Gateway 간에 암호화됩니다. 그 외에도 암호화는 미적용 VPC에서 트래픽이 전송되는 리소스에 따라 달라지며 암호화가 보장되지 않습니다(VPC가 적용 모드가 아니기 때문). 모든 VPC는 동일한 리전에 있어야 합니다(여기에서 세부 정보 참조).



- 이 다이어그램에서 VPC 1, VPC 2, VPC3은 암호화 제어가 적용 모드로 설정되어 있으며, 모니터링 모드에서 실행되는 암호화 제어가 있는 VPC 4에 연결되어 있습니다.
- VPC1, VPC2, VPC3 간의 모든 트래픽은 암호화됩니다.
- 자세히 설명하면, VPC 1의 리소스와 VPC 4의 리소스 간의 모든 트래픽은 Nitro System 하드웨어에서 제공하는 암호화를 사용하여 Transit Gateway까지 암호화됩니다. 그 외에도 암호화 상태는 VPC 4의 리소스에 따라 달라지며 암호화가 보장되지 않습니다.

Transit Gateway 암호화 지원에 대한 자세한 내용은 Transit Gateway 설명서를 참조하세요.

계정 수준에서 VPC Encryption Controls 활성화

계정 수준 VPC Encryption Controls를 사용하면 계정의 모든 VPC에 적용되는 Encryption Controls 모드와 리소스 제외 항목을 설정할 수 있습니다. 이를 통해 각 VPC를 개별적으로 구성할 필요 없이 암호화 정책을 중앙 집중식으로 관리할 수 있습니다.

계정 수준 Encryption Controls는 VPC 수준 제어와 동일한 8가지 리소스 제외 항목(인터넷 게이트웨이, NAT 게이트웨이, 외부 전용 인터넷 게이트웨이, VPC 피어링, 가상 프라이빗 게이트웨이, Lambda, VPC Lattice, Elastic File System)을 지원합니다.

계정 수준 Encryption Controls 모드

계정 수준에서 다음과 같은 모드를 설정할 수 있습니다.

비관리형

VPC Encryption Controls는 현재 모드로 유지되며 VPC 수준에서 관리할 수 있습니다.

시도 모니터

계정의 모든 VPC를 모니터 모드로 전환하려고 시도합니다. 생성을 차단하지 않고 암호화되지 않은 리소스를 식별합니다.

시도 강제

계정의 모든 VPC를 강제 모드로 전환하려고 시도합니다. 암호화되지 않은 리소스의 생성을 차단합니다.

계정 수준 Encryption Controls 상태

계정 수준 VPC Encryption Controls는 다음 상태 중 하나일 수 있습니다.

default-state

계정 수준 VPC Encryption Controls가 활성화되지 않았습니다.

transitions-in-progress

계정의 VPC가 지정된 모드로 전환되고 있습니다.

transitions-partially-successful

하나 이상의 VPC가 예상 모드로 전환되지 않았습니다.

transitions-successful

모든 VPC가 예상 모드로 성공적으로 전환되었습니다.

transitions-failed

모든 VPC가 예상 모드로 전환되지 못했습니다.

조직 수준에서 VPC Encryption Controls 활성화

AWS Organizations를 사용하여 조직의 계정을 관리하는 경우 AWS Organizations 선언적 정책을 사용하여 조직의 계정에 VPC Encryption Controls를 적용할 수 있습니다. VPC Encryption Controls 선언적 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 지원되는 선언적 정책을 참조하세요.

가격 책정

요금에 관한 자세한 내용은 Amazon VPC 요금을 참조하세요.

AWS CLI 명령 참조

설정 및 구성

- aws ec2 create-vpc-encryption-control
- aws ec2 modify-vpc-encryption-control
- aws ec2 modify-account-vpc-encryption-control
- aws ec2 describe-account-vpc-encryption-control
- aws ec2 tgw modify-transit-gateway

모니터링 및 문제 해결

- aws ec2 describe-vpc-encryption-controls
- aws ec2 get-vpc-resources-blocking-encryption-enforcement
- aws ec2 create-flow-logs
- aws ec2 describe-flow-logs
- aws logs query

정리

- aws ec2 delete-vpc-encryption-control

추가 리소스

자세한 설정 지침은 블로그 VPC 암호화 제어 기능 소개: 리전 내 VPC 간 및 VPC 내 전송 중 암호화 적용을 참조하세요.

API에 대한 자세한 내용은 EC2 API 참조 안내서를 참조하세요.

Amazon VPC용 자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 어떤 사용자가 Amazon VPC 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지를 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

내용

- 대상
- ID로 인증
- 정책을 사용하여 액세스 관리
- Amazon VPC가 IAM과 작동하는 방식
- Amazon VPC 정책 예시
- Amazon VPC 자격 증명 및 액세스 문제 해결
- AWSAmazon Virtual Private Cloud에 대한 관리형 정책
- VPC에 서비스 링크 역할 사용

대상

AWS Identity and Access Management(IAM)를 사용하는 방법은 Amazon VPC에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 – Amazon VPC 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 다른 Amazon VPC 기능을 사용하여 작업을 수행한다면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Amazon VPC의 기능에 액세스할 수 없다면 Amazon VPC 자격 증명 및 액세스 문제 해결를 참조하세요.

서비스 관리자 – 회사에서 Amazon VPC 리소스를 책임지고 있다면 사용하는 서비스에 대한 완전한 액세스 권한이 있을 것입니다. 서비스 관리자는 직원이 액세스해야 하는 Amazon VPC 기능과 리소스를 결정합니다. IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 Amazon VPC에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 Amazon VPC가 IAM과 작동하는 방식을 참조하세요.

IAM 관리자 - IAM 관리자라면 Amazon VPC에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. 정책 예시를 보려면 Amazon VPC 정책 예시를 참조하세요.

ID로 인증

인증은 ID 자격 증명을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자이나 IAM 사용자로, 또는 IAM 역할을 수임하여 인증(에 로그인)받아야 합니다.

AWS IAM Identity Center(IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해 AWS는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용 AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

AWS 계정을 생성하는 경우에는 모든 AWS 서비스 서비스와 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 루트 사용자라는 단일 로그인 ID로 시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자가 필요한 작업 목록은 IAM 사용자 설명서의 루트 사용자 자격 증명이 필요한 작업을 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서에서 임시 자격 증명을 사용하여 AWS에 액세스하려면 인간 사용자가 ID 제공업체와의 페더레이션을 사용하도록 요구를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 AWS CLI 또는 AWS API 작업을 직접적으로 호출하여 역할을 수임할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 생성하고 AWS ID 또는 리소스에 연결하여 AWS에서 내 액세스를 제어합니다. 정책은 자격 증명이나 리소스와 연결될 때 해당 권한을 정의합니다. AWS는 보안 주체가 요청을 보낼 때 이러한 정책을 평가합니다. 대부분의 정책은 AWS에 JSON 문서로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 위탁자(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3, AWS WAF 및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 유형

AWS는 이러한 정책 타입이 부여하는 최대 권한을 설정할 수 있는 추가 정책 타입을 지지합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은 AWS Organizations사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

Amazon VPC가 IAM과 작동하는 방식

IAM을 사용하여 Amazon VPC에 대한 액세스를 관리하기 전에 Amazon VPC에서 사용할 수 있는 IAM 기능을 이해해야 합니다. Amazon VPC 및 기타 AWS 서비스에서 IAM을 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하세요.

목차

- 작업
- 리소스
- 조건 키
- Amazon VPC 리소스 기반 정책
- 태그 기반 권한 부여
- IAM 역할

IAM 자격 증명 기반 정책을 사용하면 허용 또는 거부된 작업을 지정할 수 있습니다. 일부 작업의 경우 작업이 허용 또는 거부되는 리소스 및 조건을 지정할 수 있습니다. Amazon VPC는 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용자 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

작업

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

Amazon VPC는 API 네임스페이스를 Amazon EC2와 공유합니다. Amazon VPC의 정책 작업은 작업 앞에 `ec2:` 접두사를 사용합니다. 예를 들어 `CreateVpc` API 작업을 사용하여 사용자에게 VPC를 생성할 수 있는 권한을 부여하려면 `ec2:CreateVpc` 작업에 대한 액세스 권한을 부여합니다. 정책 설명에는 `Action` 또는 `NotAction` 요소가 반드시 추가되어야 합니다.

단일 명령문에서 여러 작업을 지정하려면 다음 예시와 같이 쉼표로 구분합니다.

```
"Action": [
      "ec2:action1",
      "ec2:action2"
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `Describe`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "ec2:Describe*"
```

Amazon VPC 작업 목록을 보려면 서비스 승인 참조의 Amazon EC2에서 정의한 작업을 참조하세요.

리소스

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

VPC 리소스에는 다음 예시와 같이 ARN이 있습니다.

```
arn:${Partition}:ec2:${Region}:${Account}:vpc/${VpcId}
```

예를 들어, 명령문에서 `vpc-1234567890abcdef0` VPC를 지정하려면 다음 예시에 표시된 ARN을 사용합니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-1234567890abcdef0"
```

특정 리전에서 특정 계정에 속하는 모든 VPC를 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*"
```

리소스 생성 작업과 같은 일부 Amazon VPC 작업은 특정 리소스에서 수행할 수 없습니다. 이러한 경우 와일드카드(*)를 사용해야 합니다.

```
"Resource": "*"
```

다양한 Amazon EC2 API 작업에는 여러 리소스가 관여합니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
      "resource1",
      "resource2"
]
```

Amazon VPC 리소스 유형 및 해당 ARN 목록을 보려면 서비스 승인 참조의 Amazon EC2에서 정의한 리소스 유형을 참조하세요.

조건 키

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수

있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용자 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

모든 Amazon EC2 작업은 `aws:RequestedRegion` 및 `ec2:Region` 조건 키를 지원합니다. 자세한 내용은 예: 특정 리전에 대한 액세스 제한을 참조하세요.

Amazon VPC는 자체 조건 키 세트를 정의하며 일부 전역 조건 키 사용도 지원합니다. Amazon VPC 조건 키 목록을 보려면 서비스 승인 참조의 Amazon EC2에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon EC2에서 정의한 작업을 참조하세요.

Amazon VPC 리소스 기반 정책

리소스 기반 정책은 지정된 보안 주체가 Amazon VPC 리소스에 대해 수행할 수 있는 작업 및 관련 조건을 지정하는 JSON 정책 문서입니다.

교차 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 교차 계정 위탁자를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 위탁자와 리소스가 다른 AWS 계정에 있는 경우 위탁자 엔터티가 리소스에 액세스할 권한도 부여해야 합니다. 엔터티에 자격 증명 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 위탁자에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

태그 기반 권한 부여

태그를 Amazon VPC 리소스에 연결하거나 요청을 통해 태그를 전달할 수 있습니다. 태그를 기반으로 액세스를 제어하려면 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다. 자세한 내용은 Amazon EC2 사용 설명서의 생성 시 리소스 태깅에 대한 권한 부여를 참조하세요.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예시는 특정 VPC로 인스턴스 시작에서 확인할 수 있습니다.

IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 내 엔터티입니다.

임시 자격 증명 사용

임시 자격 증명을 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 맡거나, 교차 계정 역할을 맡을 수 있습니다. AssumeRole 또는 GetFederationToken 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 가져옵니다.

Amazon VPC는 임시 자격 증명 사용을 지원합니다.

서비스 연결 역할

서비스 연결 역할을 사용하면 AWS 서비스에서 다른 서비스의 리소스에 액세스하여 사용자 대신 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 나타나고 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

전송 게이트웨이는 서비스 연결 역할을 지원합니다.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 서비스 역할을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역할은 IAM 계정에 나타나고, 해당 계정이 소유합니다. 즉, IAM 관리자가 이 역할에 대한 권한을 변경할 수 있습니다. 그러나 권한을 변경하면 서비스의 기능이 손상될 수 있습니다.

Amazon VPC는 흐름 로그에 대한 서비스 역할을 지원합니다. 흐름 로그를 만들 때 흐름 로그 서비스에서 CloudWatch Logs에 액세스할 수 있는 역할을 선택해야 합니다. 자세한 내용은 the section called “CloudWatch Logs에 흐름 로그를 게시하는 IAM 역할” 단원을 참조하세요.

Amazon VPC 정책 예시

기본적으로 IAM 역할은 VPC 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS CLI 또는 AWS API를 사용해 태스크를 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 역할에 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 역할에 이러한 정책을 연결해야 합니다.

이러한 예시 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

내용

- 정책 모범 사례
- Amazon VPC 콘솔 사용
- 퍼블릭 서브넷이 포함된 VPC 만들기
- VPC 리소스 수정 및 삭제
- 보안 그룹 관리
- 보안 그룹 규칙 관리
- 특정 서브넷으로 인스턴스 시작

- 특정 VPC로 인스턴스 시작
- VPC 및 서브넷에 대한 퍼블릭 액세스 차단
- 추가 Amazon VPC 정책 예시

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Amazon VPC 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS관리형 정책을 사용합니다. AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. CloudFormation와 같이, 특정 AWS 서비스를 통해 사용되는 경우에만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 내용은 IAM 사용자 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 – AWS 계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우, 추가 보안을 위해 MFA를 설정합니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

Amazon VPC 콘솔 사용

Amazon VPC 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정에서 Amazon VPC 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(IAM 역할)에 대해 의도대로 작동하지 않습니다.

다음 정책은 VPC 콘솔에 리소스를 나열할 수 있는 권한을 역할에 부여하지만 리소스를 생성, 업데이트 또는 삭제할 수는 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeClassicLinkInstances",
                "ec2:DescribeClientVpnEndpoints",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeFlowLogs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeMovingAddresses",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeNetworkInterfacePermissions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeStaleSecurityGroups",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeTags",
                "ec2:DescribeTrafficMirrorFilters",
                "ec2:DescribeTrafficMirrorSessions",
                "ec2:DescribeTrafficMirrorTargets",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeTransitGatewayVpcAttachments",
                "ec2:DescribeTransitGatewayRouteTables",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeVpcClassicLinkDnsSupport",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcEndpointConnectionNotifications",
                "ec2:DescribeVpcEndpointConnections",
                "ec2:DescribeVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpointServicePermissions",
                "ec2:DescribeVpcEndpointServices",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ec2:GetManagedPrefixListAssociations",
                "ec2:GetManagedPrefixListEntries"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS CLI 또는 AWS API만 호출하는 역할에는 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 역할이 수행해야 하는 API 작업과 일치하는 작업에만 액세스를 허용합니다.

퍼블릭 서브넷이 포함된 VPC 만들기

다음 예시에서는 역할이 VPC, 서브넷, 라우팅 테이블 및 인터넷 게이트웨이를 만들 수 있도록 설정합니다. 또한 역할은 인터넷 게이트웨이를 VPC에 연결하고 라우팅 테이블에 라우팅을 생성할 수 있습니다. `ec2:ModifyVpcAttribute` 작업을 통해 역할은 VPC로 시작된 각 인스턴스가 DNS 호스트 이름을 수신할 수 있도록 VPC에 대한 DNS 호스트 이름을 활성화합니다.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpc",
        "ec2:CreateSubnet",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateRouteTable",
        "ec2:CreateRoute",
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",
        "ec2:AssociateRouteTable",
        "ec2:ModifyVpcAttribute"
      ],
      "Resource": "*"
    }
   ]
}
```

또한 위의 정책은 역할이 Amazon VPC 콘솔에서 VPC를 생성할 수 있도록 허용합니다.

VPC 리소스 수정 및 삭제

역할이 수정하거나 삭제할 수 있는 VPC 리소스를 제어할 수 있습니다. 예를 들어 역할은 다음 정책을 통해 `Purpose=Test` 태그가 있는 라우팅 테이블에서 작업하고 삭제할 수 있습니다. 또한 이 정책은 역할이 `Purpose=Test` 태그가 있는 인터넷 게이트웨이를 삭제할 수만 있도록 지정합니다. 역할은 이 태그가 없는 라우팅 테이블 또는 인터넷 게이트웨이를 사용할 수 없습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteInternetGateway",
            "Resource": "arn:aws:ec2:*:*:internet-gateway/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Test"
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRouteTable",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

보안 그룹 관리

다음 정책을 사용하여 역할은 보안 그룹을 관리할 수 있습니다. 첫 번째 문을 통해 역할은 Stack=test 태그가 있는 모든 보안 그룹을 삭제할 수 있고 Stack=test 태그가 있는 모든 보안 그룹에 대한 인바운드 및 아웃바운드 규칙을 관리할 수 있습니다. 두 번째 문은 역할이 Stack=Test 태그로 만든 보안 그룹에 태그를 지정하도록 요구합니다. 세 번째 문을 통해 역할은 보안 그룹을 만들 때 태그를 생성할 수 있습니다. 네 번째 문을 통해 역할은 모든 보안 그룹 및 보안 그룹 규칙을 볼 수 있습니다. 다섯 번째 문을 통해 역할은 VPC에서 보안 그룹을 생성할 수 있습니다.

> **Note**
>
> AWS CloudFormation 서비스에서 이 정책을 사용하여 필수 태그가 있는 보안 그룹을 생성할 수 없습니다. 태그가 필요한 ec2:CreateSecurityGroup 작업에서 조건을 제거하면 정책이 작동합니다.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:DeleteSecurityGroup",
                "ec2:ModifySecurityGroupRules",
                "ec2:UpdateSecurityGroupRuleDescriptionsIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Stack": "test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSecurityGroup",
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Stack": "test"
                },
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "Stack"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateSecurityGroup"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeVpcs",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSecurityGroup",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

역할이 인스턴스와 연결된 보안 그룹을 변경할 수 있도록 하려면 정책에 `ec2:ModifyInstanceAttribute` 작업을 추가합니다.

역할이 네트워크 인터페이스의 보안 그룹을 변경할 수 있도록 하려면 정책에 `ec2:ModifyNetworkInterfaceAttribute` 작업을 추가합니다.

보안 그룹 규칙 관리

다음 정책은 역할이 모든 보안 그룹 및 보안 그룹 규칙을 조회하고 특정 VPC의 보안 그룹에 대한 인바운드 및 아웃바운드 규칙을 추가 및 제거하며 지정된 VPC에 대한 규칙 설명을 수정할 권한을 부여합니다. 첫 번째 문은 `ec2:Vpc` 조건 키를 사용하여 특정 VPC 대한 권한 범위를 지정할 수 있습니다.

두 번째 문은 역할에 모든 보안 그룹, 보안 그룹 규칙 및 태그를 설명하는 권한을 부여합니다. 이를 통해 역할은 보안 그룹 규칙을 수정하기 위해 볼 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
                "ec2:ModifySecurityGroupRules"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:security-group/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifySecurityGroupRules"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:security-group-rule/
*"
        }
    ]
}
```

특정 서브넷으로 인스턴스 시작

다음 정책은 역할에 인스턴스를 특정 서브넷으로 시작하고 요청에 특정 보안 그룹을 사용하는 권한을 부여합니다. 이 정책에서는 서브넷에 대한 ARN과 보안 그룹에 대한 ARN을 지정하여 이런 권한을 부여합니다. 역할이 다른 서브넷으로 인스턴스를 시작하거나 다른 보안 그룹을 사용하여 시작하려고 하면 (또 다른 정책 또는 설명문에서 역할에 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, 역할은 인스턴스를 시작할 때 이 리소스를 생성할 권한이 필요합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:us-east-1::image/ami-*",
                "arn:aws:ec2:us-east-1:123456789012:instance/*",
                "arn:aws:ec2:us-
east-1:123456789012:subnet/subnet-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:123456789012:network-interface/*",
                "arn:aws:ec2:us-east-1:123456789012:volume/*",
                "arn:aws:ec2:us-east-1:123456789012:key-pair/*",
                "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-0abcdef1234567890"
            ]
        }
    ]
}
```

특정 VPC로 인스턴스 시작

다음 정책에서는 역할에 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 역할에 "department=dev" 태그가 있는 AMI만 사용하여 인스턴스를 시작하는 권한을 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:subnet/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:us-east-1:123456789012:instance/*",
                "arn:aws:ec2:us-east-1:123456789012:volume/*",
                "arn:aws:ec2:us-east-1:123456789012:network-interface/*",
                "arn:aws:ec2:us-east-1:123456789012:key-pair/*",
                "arn:aws:ec2:us-east-1:123456789012:security-group/*"
            ]
        }
    ]
}
```

VPC 및 서브넷에 대한 퍼블릭 액세스 차단

다음 정책 예제에서는 VPC 및 서브넷의 리소스에 대한 퍼블릭 액세스를 차단할 수 있도록 VPC 퍼블릭 액세스 차단(BPA) 기능으로 작업할 수 있는 권한을 역할에 부여합니다.

예제 1 - VPC BPA 계정 전체 설정 및 VPC BPA 제외 항목에 대한 읽기 전용 액세스를 허용합니다.

JSON

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

예제 2 - VPC BPA 계정 전체 설정 및 VPC BPA 제외 항목에 대한 전체 읽기 및 쓰기 액세스를 허용합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VPCBPAFullAccess",
      "Action": [
        "ec2:DescribeVpcBlockPublicAccessOptions",
        "ec2:DescribeVpcBlockPublicAccessExclusions",
        "ec2:ModifyVpcBlockPublicAccessOptions",
        "ec2:CreateVpcBlockPublicAccessExclusion",
        "ec2:ModifyVpcBlockPublicAccessExclusion",
        "ec2:DeleteVpcBlockPublicAccessExclusion"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

예제 3 - VPC BPA 설정 수정 및 제외 항목 생성을 제외한 모든 EC2 API에 대한 액세스를 허용합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2FullAccess",
            "Action": [
                "ec2:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "VPCBPAPartialAccess",
            "Action": [
                "ec2:ModifyVpcBlockPublicAccessOptions",
                "ec2:CreateVpcBlockPublicAccessExclusion"
            ],
            "Effect": "Deny",
            "Resource": "*"
        }
    ]
}
```

추가 Amazon VPC 정책 예시

다음 문서에서 Amazon VPC와 관련된 추가 IAM 정책 예시를 확인할 수 있습니다.

- 관리형 접두사 목록
- 트래픽 미러링
- 전송 게이트웨이
- VPC 엔드포인트 및 VPC 엔드포인트 서비스(AWS PrivateLink)
- VPC 피어링

Amazon VPC 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Amazon VPC 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

문제

- Amazon VPC에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 AWS 계정 외부의 사용자가 내 Amazon VPC 리소스에 액세스할 수 있도록 허용하려고 합니다.

Amazon VPC에서 작업을 수행할 권한이 없음

AWS Management Console에서 작업을 수행할 권한이 없다는 메시지가 나타나는 경우 관리자에게 문의하여 도움을 받아야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 오류 예는 `mateojackson` IAM 사용자가 콘솔을 사용하여 서브넷에 대한 세부 정보를 보려고 하지만 `ec2:DescribeSubnets` 권한이 없는 IAM 역할에 속하는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:DescribeSubnets on resource: subnet-id
```

이 경우 Mateo는 관리자에게 정책을 업데이트하여 자신에게 서브넷에 대한 액세스를 허용하도록 요청합니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon VPC에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새로운 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예시 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 Amazon VPC에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 AWS 계정 외부의 사용자가 내 Amazon VPC 리소스에 액세스할 수 있도록 허용하려고 합니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스하는 데 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- Amazon VPC에서 이러한 기능을 지원하는지 여부를 알아보려면 Amazon VPC가 IAM과 작동하는 방식을 참조하세요.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하세요.
- 리소스에 대한 액세스 권한을 서드 파티 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 서드 파티가 소유한 AWS 계정에 대한 액세스 제공을 참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 교차 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

AWSAmazon Virtual Private Cloud에 대한 관리형 정책

AWS 관리형 정책은 AWS에 의해 생성되고 관리되는 독립 실행형 정책입니다. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있기 때문에 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에서 정의한 권한은 변경할 수 없습니다. 만약 AWS가 AWS 관리형 정책에 정의된 권한을 업데이트할 경우 정책이 연결되어 있는 모든 위탁자 ID(사용자, 그룹 및 역할)에도 업데이트가

적용됩니다. 새 AWS 서비스을(를) 시작하거나 새 API 작업을 기존 서비스에 이용하는 경우, AWS가 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AmazonVPCFullAccess

`AmazonVPCFullAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다. 이 정책은 Amazon VPC 대한 전체 액세스를 허용하는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 AmazonVPCFullAccess를 참조하세요.

AWS 관리형 정책: AmazonVPCReadOnlyAccess

`AmazonVPCReadOnlyAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다. 이 정책은 Amazon VPC 대한 읽기 전용 액세스를 허용하는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 AmazonVPCReadOnlyAccess를 참조하세요.

AWS 관리형 정책: AmazonVPCCrossAccountNetworkInterfaceOperations

`AmazonVPCCrossAccountNetworkInterfaceOperations` 정책을 IAM ID에 연결할 수 있습니다. 이 정책은 ID가 네트워크 인터페이스를 만들어 교차 계정 리소스에 연결할 수 있는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 AmazonVPCCrossAccountNetworkInterfaceOperations를 참조하세요.

AWS 관리형 정책: AWSServiceRoleForNATGateway

`AWSServiceRoleForNATGateway` 정책을 IAM ID에 연결할 수 있습니다. 이 정책은 자격 증명이 사용자를 대신하여 리전 NAT 게이트웨이를 자동으로 규모 조정할 수 있는 권한을 부여합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSServiceRoleForNATGateway를 참조하세요.

AWS 관리형 정책에 대한 Amazon VPC 업데이트

이 서비스가 2021년 3월에 이러한 변경 사항을 추적하기 시작한 이후 Amazon VPC용 AWS 관리형 정책 업데이트에 대한 세부 정보를 확인합니다.

변경 사항	설명	날짜
the section called “AmazonVPCFullAccess” - 기존 정책에 대한 업데이트	IPAM이 관리형 접두사 목록을 수정하고 읽을 수 있도록 AWSIPAMServiceRolePolicy 관리형 정책(ec2:ModifyManagedPrefixList, ec2:DescribeManagedPrefixLists 및 ec2:GetManagedPrefixListEntries)에 작업을 추가했습니다.	2025년 10월 31일
the section called “AWSServiceRoleForNATGateway ” - 새 정책	새로운 AWSServiceRoleForNATGateway 정책은 자격 증명이 리전 NAT 게이트웨이를 자동으로 규모 조정할 수 있도록 허용합니다.	2025년 11월 19일
the section called “AmazonVPCFullAccess” - 기존 정책 업데이트	VPC와 보안 그룹 연결을 연결, 연결 해제 및 볼 수 있는 AssociateSecurityGroupVpc, DescribeSecurityGroupVpcAssociations 및 DisassociateSecurityGroupVpc 작업이 추가되었습니다.	2024년 12월 9일
the section called “AmazonVPCReadOnlyAccess” -기존 정책 업데이트	VPC와 보안 그룹 연결을 볼 수 있는 DescribeSecurityGroupVpcAssociations 작업이 추가되었습니다.	2024년 12월 9일
the section called “AmazonVPCFullAccess” -기존 정책 업데이트	VPC에서 사용할 수 있는 보안 그룹을 가져올 수 있는 GetSecurityGroupsForVpc 작업이 추가되었습니다.	2024년 2월 8일
the section called “AmazonVPCReadOnlyAccess” -기존 정책 업데이트	VPC에서 사용할 수 있는 보안 그룹을 가져올 수 있는	2024년 2월 8일

변경 사항	설명	날짜
	GetSecurityGroupsForVpc 작업이 추가되었습니다.	
the section called “AmazonVPCCrossAccountNetworkInterfaceOperations” -기존 정책 업데이트	AssignIpv6Addresses 및 UnassignIpv6Addresses 작업이 추가되었으며, 이를 통해 네트워크 인터페이스와 연결된 IPv6 주소를 관리할 수 있습니다.	2023년 9월 25일
the section called “AmazonVPCReadOnlyAccess” -기존 정책 업데이트	보안 그룹 규칙을 볼 수 있는 DescribeSecurityGroupRules 작업이 추가되었습니다.	2021년 8월 2일
the section called “AmazonVPCFullAccess” - 기존 정책 업데이트	보안 그룹 규칙을 보고 수정할 수 있는 DescribeSecurityGroupRules 및 ModifySecurityGroupRules 작업이 추가되었습니다.	2021년 8월 2일
the section called “AmazonVPCFullAccess” - 기존 정책 업데이트	통신업체 게이트웨이, IPv6 풀, 로컬 게이트웨이 및 로컬 게이트웨이 라우팅 테이블에 대한 작업이 추가되었습니다.	2021년 6월 23일
the section called “AmazonVPCReadOnlyAccess” -기존 정책 업데이트	통신업체 게이트웨이, IPv6 풀, 로컬 게이트웨이 및 로컬 게이트웨이 라우팅 테이블에 대한 작업이 추가되었습니다.	2021년 6월 23일

VPC에 서비스 링크 역할 사용

Amazon VPC는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 VPC에 직접 연결되는 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 VPC에서 사전

정의되며 서비스에서 사용자를 대신하여 다른 AWS 서비스를 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 VPC를 더 쉽게 설정할 수 있습니다. VPC에서 서비스 연결 역할의 권한을 정의하므로 별도로 정의되지 않는 한 VPC만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 삭제할 수 없기 때문에 VPC 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 AWS IAM으로 작업하는 서비스를 참조하고 서비스 연결 역할 열이 예로 표시된 서비스를 찾으세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

VPC에 대한 서비스 연결 역할 권한

VPC는 AWSServiceRoleForNATGateway라는 서비스 연결 역할을 사용합니다. 이 서비스 연결 역할을 사용하면 Amazon VPC가 사용자를 대신하여 탄력적 IP 주소를 할당하여 리전 NAT 게이트웨이를 자동으로 규모 조정하고, 요청 시 기존 탄력적 IP를 리전 NAT 게이트웨이에 연결 및 연결 해제하고, 네트워크 인터페이스를 설명하여 새 가용 영역으로 자동으로 확장할 기존 인프라를 식별할 수 있습니다.

AWSServiceRoleForNATGateway 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `ec2-nat-gateway.amazonaws.com`

AWSNATGatewayServiceRolePolicy라는 역할 권한 정책은 VPC가 지정된 리소스에 대해 다음 작업을 수행할 수 있도록 허용합니다.

- 작업: `AllocateAddress`는 서비스 관리형 EIP에 대해 사용자를 대신하여 EIP를 할당합니다. 서비스 관리형 EIP는 서비스 관리형 태그 및 ReleaseAddress를 사용하여 후속 태그 지정을 자동으로 처리합니다.
- 작업: `AssociateAddress`는 기존 탄력적 IP 주소에 대해 요청에 따라 해당 주소를 리전 NA 게이트웨이에 수동으로 연결합니다.
- 작업: `DisassociateAddress`는 기존 탄력적 IP 주소에 대해 요청에 따라 해당 주소를 리전 NAT 게이트웨이에서 제거합니다.
- 작업: `DescribeAddresses`는 연결된 고객 제공 EIP에서 퍼블릭 IP 주소 정보를 가져옵니다.

- 작업: `DescribeNetworkInterface`는 기존 네트워크 인터페이스에 대해 인프라가 상주하는 가용 영역을 자동으로 식별하여 새 영역으로 자동으로 스케일 아웃합니다.

사용자, 그룹 또는 역할이 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 사용 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

VPC에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS CLI 또는 AWS API에서 '리전' 가용 모드로 NAT 게이트웨이를 생성하면 VPC가 서비스 연결 역할을 생성합니다.

> **Important**
>
> 이러한 서비스 연결 역할은 해당 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 또한 VPC 서비스가 서비스 연결 역할을 지원하기 시작한 2017년 1월 1일 이전에 이 서비스를 사용 중이었다면 VPC가 사용자 계정에 AWSServiceRoleForNATGateway 역할을 이미 생성했습니다. 자세한 내용은 내 AWS 계정에 표시되는 새 역할을 참조하세요.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. '리전' 가용 모드로 NAT 게이트웨이를 생성하면 VPC가 서비스 연결 역할을 다시 생성합니다.

IAM 콘솔에서 AWSServiceRoleForNATGateway 사용 사례를 사용하여 서비스 연결 역할을 생성할 수도 있습니다. AWS CLI 또는 AWS API에서 `ec2-nat-gateway.amazonaws.com` 서비스 이름의 서비스 연결 역할을 생성합니다. 자세한 내용은 IAM 사용자 설명서의 서비스 연결 역할 생성을 참조하세요. 이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

VPC에 대한 서비스 연결 역할 편집

VPC는 AWSServiceRoleForNATGateway 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

VPC에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

> **Note**
>
> 리소스를 삭제하려 할 때 VPC 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForNATGateway에서 사용하는 VPC 리소스를 삭제하는 방법

- 배포된 모든 리전에서 모든 리전 NAT 게이트웨이를 삭제합니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요

IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 AWSServiceRoleForNATGateway 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

VPC 서비스 연결 역할이 지원되는 리전

VPC는 서비스가 제공되는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하세요.

VPC는 서비스가 제공되는 모든 리전에서 서비스 연결 역할 사용을 지원하지 않습니다. 다음 리전에서 AWSServiceRoleForNATGateway 역할을 사용할 수 있습니다.

리전 이름	리전 자격 증명	VPC에서 지원
미국 동부(버지니아 북부)	us-east-1	예
미국 동부(오하이오)	us-east-2	예
미국 서부(캘리포니아 북부)	us-west-1	예
미국 서부(오리건)	us-west-2	예
아프리카(케이프타운)	af-south-1	예

리전 이름	리전 자격 증명	VPC에서 지원
아시아 태평양(홍콩)	ap-east-1	예
아시아 태평양(타이베이)	ap-east-2	예
아시아 태평양(자카르타)	ap-southeast-3	예
아시아 태평양(뭄바이)	ap-south-1	예
아시아 태평양(하이데라바드)	ap-south-2	예
아시아 태평양(오사카)	ap-northeast-3	예
아시아 태평양(서울)	ap-northeast-2	예
아시아 태평양(싱가포르)	ap-southeast-1	예
아시아 태평양(시드니)	ap-southeast-2	예
아시아 태평양(도쿄)	ap-northeast-1	예
아시아 태평양(멜버른)	ap-southeast-4	예
아시아 태평양(말레이시아)	ap-southeast-5	예
아시아 태평양(뉴질랜드)	ap-southeast-6	예
아시아 태평양(태국)	ap-southeast-7	예
캐나다(중부)	ca-central-1	예
캐나다 서부(캘거리)	ca-west-1	예
유럽(프랑크푸르트)	eu-central-1	예
유럽(취리히)	eu-central-2	예
유럽(아일랜드)	eu-west-1	예
유럽(런던)	eu-west-2	예

리전 이름	리전 자격 증명	VPC에서 지원
유럽(밀라노)	eu-south-1	예
유럽(스페인)	eu-south-2	예
유럽(파리)	eu-west-3	예
유럽(스톡홀름)	eu-north-1	예
이스라엘(텔아비브)	il-central-1	예
중동(바레인)	me-south-1	예
중동(UAE)	me-central-1	예
중동(사우디아라비아)	me-west-1	예
멕시코(중부)	mx-central-1	예
남아메리카(상파울루)	sa-east-1	예
AWS GovCloud (US-East)	us-gov-east-1	아니요
AWS GovCloud(미국 서부)	us-gov-west-1	아니요

Amazon VPC의 인프라 보안

관리형 서비스인 Amazon Virtual Private Cloud는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon VPC에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

네트워크 격리

가상 프라이빗 클라우드(VPC)는 AWS 클라우드에서 논리적으로 격리된 고유한 영역의 가상 네트워크입니다. 별도의 VPC를 사용하여 워크로드별 또는 조직체별로 인프라를 격리합니다.

서브넷은 VPC의 IP 주소 범위입니다. 인스턴스를 시작할 때 VPC의 서브넷에서 인스턴스를 시작합니다. 서브넷을 사용하여 단일 VPC 내의 애플리케이션 티어(예: 웹, 애플리케이션 및 데이터베이스)를 격리합니다. 인터넷에서 직접 액세스하면 안 되는 경우 프라이빗 서브넷을 인스턴스에 사용합니다.

AWS PrivateLink를 사용하여 VPC의 리소스에서 프라이빗 IP 주소를 사용하여 AWS 서비스에 연결하고 서비스를 VPC 직접 호스팅된 것처럼 이용하도록 할 수 있습니다. 따라서 AWS 서비스에 액세스하기 위해 인터넷 게이트웨이나 NAT 장치를 사용할 필요가 없습니다.

네트워크 트래픽 제어

EC2 인스턴스와 같은 VPC의 네트워크 트래픽을 제어하기 위해 다음 옵션을 고려해 보세요.

- 보안 그룹을 VPC에 대한 네트워크 액세스를 제어하는 기본 메커니즘으로 활용합니다. 필요한 경우 네트워크 ACL을 사용하여 상태 비저장의 거친 네트워크 제어를 제공합니다. 보안 그룹은 상태 저장 패킷 필터링을 수행하고 다른 보안 그룹을 참조하는 규칙을 만들 수 있기 때문에 네트워크 ACL보다 다재다능합니다. 네트워크 ACL은 보조 제어 장치(예: 트래픽의 특정 하위 집합 거부) 또는 상위 수준의 서브넷 가드레일로 효과를 발휘할 수 있습니다. 또한 네트워크 ACL은 전체 서브넷에 적용되므로 인스턴스가 올바른 보안 그룹 없이 시작될 경우 이를 심층 방어 기능으로 사용할 수 있습니다.
- 인터넷에서 직접 액세스하면 안 되는 경우 프라이빗 서브넷을 인스턴스에 사용합니다. 프라이빗 서브넷에 있는 인스턴스에서 인터넷에 엑세스하려면 배스천 호스트 또는 NAT 게이트웨이를 사용합니다.
- 연결 요구 사항을 지원하기 위해 최소 네트워크 경로로 서브넷 라우팅 테이블을 구성합니다.
- 보안 그룹 또는 네트워크 인터페이스를 추가로 사용하여 일반 애플리케이션 트래픽과 별도로 Amazon EC2 인스턴스 관리 트래픽을 제어하고 감사하는 방법을 고려해 보세요. 따라서 변경 제어를 위한 특별한 IAM 정책을 고객이 구현할 수 있으므로 보안 그룹 규칙 또는 자동화된 규칙 확인 스크립트의 변경 사항을 감사하기가 쉬워집니다. 또한 다중 네트워크 인터페이스를 사용하면 호스트 기반 라우팅 정책을 생성하거나 서브넷에 할당된 네트워크 인터페이스에 따라 서로 다른 VPC 서브넷 라우팅 규칙을 적용하는 등 네트워크 트래픽 제어를 위한 추가 옵션을 이용할 수 있습니다.
- AWS Virtual Private Network 또는 Direct Connect를 사용하여 원격 네트워크에서 VPC로 프라이빗 연결을 설정합니다. 자세한 내용은 네트워크-Amazon VPC 연결 옵션을 참조하세요.
- VPC 흐름 로그를 사용하여 인스턴스에 도달하는 트래픽을 모니터링합니다.

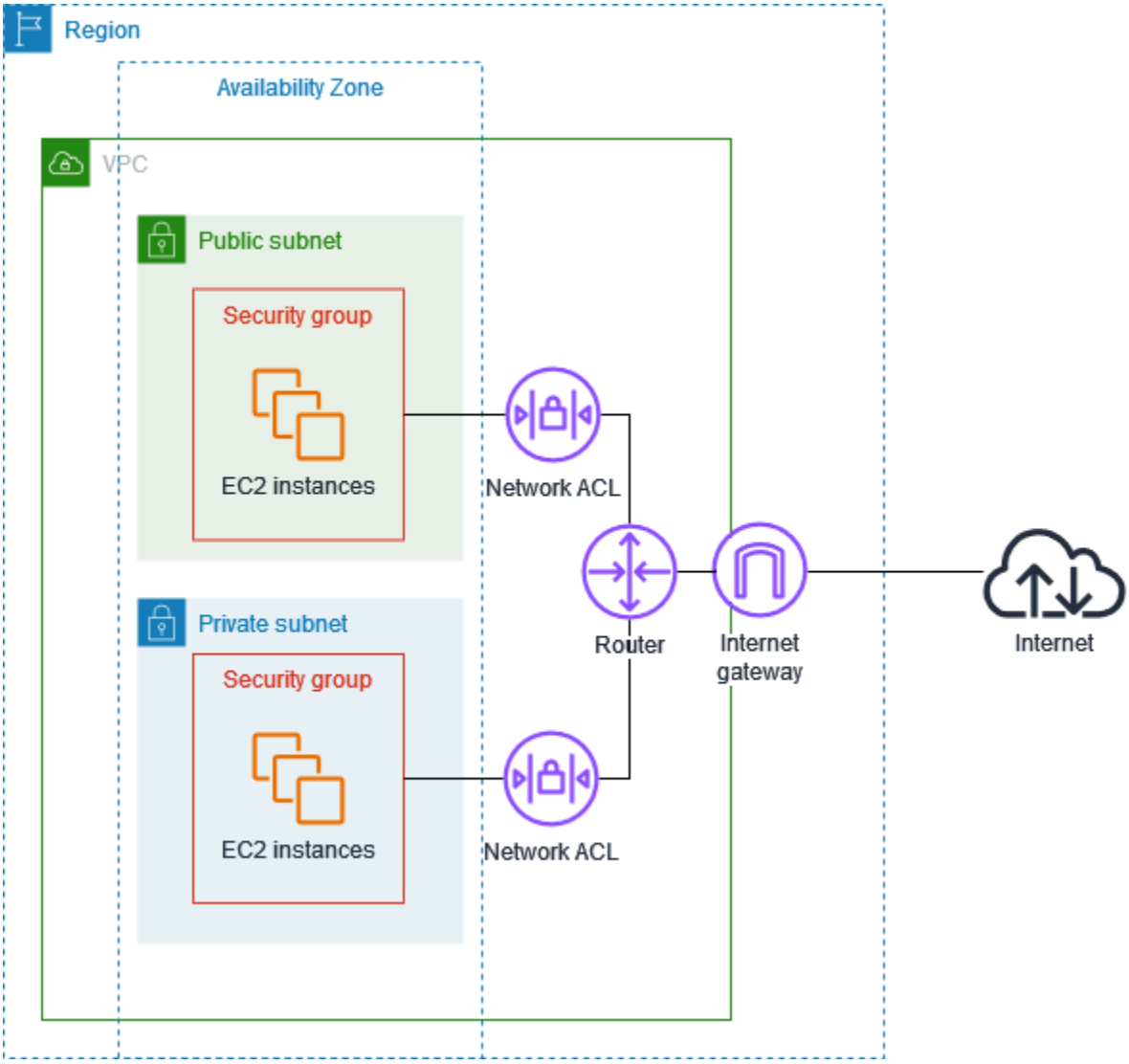
- AWS Security Hub CSPM를 사용하여 인스턴스에서 의도하지 않게 네트워크에 액세스할 수 있는지 확인합니다.
- AWS Network Firewall을 사용하여 VPC의 서브넷을 일반적인 네트워크 위협으로부터 보호합니다.

보안 그룹 및 네트워크 ACL 비교

다음 표는 보안 그룹과 네트워크 ACL의 근본적인 차이를 요약한 것입니다.

기능	보안 그룹	네트워크 ACL
작업 수준	인스턴스 수준	서브넷 수준
범위	보안 그룹과 연결된 모든 인스턴스에 적용됨	연결된 서브넷의 모든 인스턴스에 적용됨
규칙 타입	규칙만 허용	규칙 허용 및 거부
규칙 평가	트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가	트래픽과 일치하는 항목이 발견될 때까지 규칙을 오름차순으로 평가
트래픽 반환	자동 허용(상태 저장)	허용 명시 필요(상태 비저장)

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.



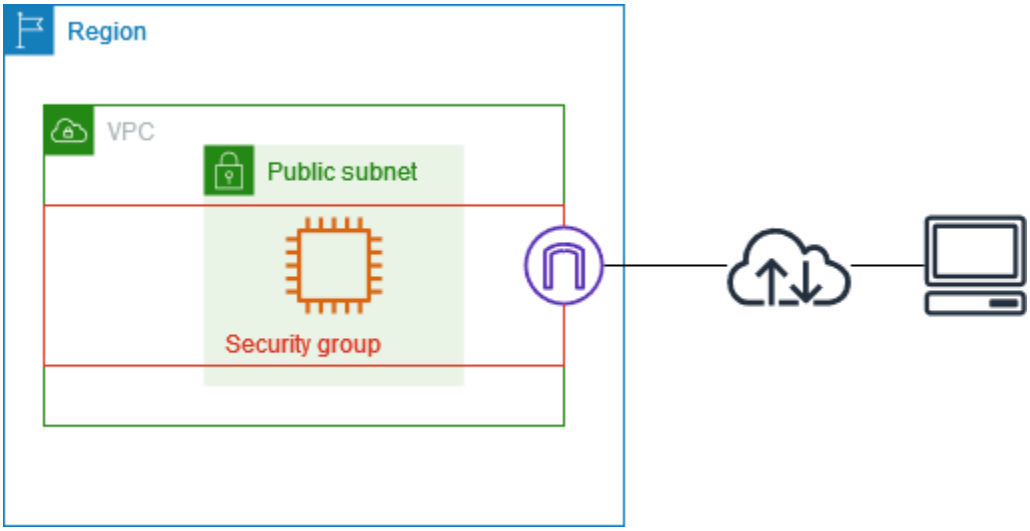
보안 그룹만 사용하여 인스턴스를 보호할 수 있습니다. 그러나 네트워크 ACL을 추가 방어 계층으로 추가할 수 있습니다. 자세한 내용은 예: 서브넷의 인스턴스에 대한 액세스 제어 단원을 참조하세요.

보안 그룹을 사용하여 AWS 리소스에 대한 트래픽 제어

보안 그룹은 연결된 리소스에 도달하고 나갈 수 있는 트래픽을 제어합니다. 예를 들어 보안 그룹을 EC2 인스턴스와 연결하면 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어합니다.

VPC를 생성할 경우 VPC는 기본 보안 그룹과 함께 제공됩니다. 각각 고유한 인바운드 및 아웃바운드 규칙이 있는 추가 보안 그룹을 생성할 수 있습니다. 각 인바운드 규칙에 대해 소스, 포트 범위 및 프로토콜을 지정할 수 있습니다. 각 아웃바운드 규칙에 대해 대상, 포트 범위 및 프로토콜을 지정할 수 있습니다.

다음 다이어그램은 서브넷, 인터넷 게이트웨이 및 보안 그룹이 있는 VPC를 보여줍니다. 이 서브넷에는 EC2 인스턴스가 포함되어 있습니다. 보안 그룹은 인스턴스에 할당됩니다. 보안 그룹은 가상 방화벽의 기능을 수행합니다. 인스턴스에 연결되는 트래픽은 보안 그룹 규칙에서 허용되는 트래픽이 유일합니다. 예를 들어, 보안 그룹에 해당 네트워크에서 인스턴스로 이동하는 ICMP 트래픽을 허용하는 규칙이 포함되어 있는 경우 컴퓨터에서 인스턴스를 ping할 수 있습니다. 보안 그룹에 SSH 트래픽을 허용하는 규칙이 포함되어 있지 않은 경우에는 SSH를 사용하여 인스턴스에 연결할 수 없습니다.



내용

- 보안 그룹 기본 사항
- 보안 그룹 예시
- 보안 그룹 규칙
- VPC에 대한 기본 보안 그룹
- VPC의 보안 그룹을 생성
- 보안 그룹 규칙 구성
- 보안 그룹 삭제
- 보안 그룹을 여러 VPC와 연결
- AWS Organizations와 보안 그룹 공유

가격 책정

보안 그룹을 사용해도 추가 요금이 부과되지 않습니다.

보안 그룹 기본 사항

- 보안 그룹 VPC 연결 기능을 사용하여 보안 그룹을 동일한 리전의 다른 VPC에 연결하는 경우, 보안 그룹을 동일한 VPC에서 생성된 리소스 또는 다른 VPC의 리소스에 할당할 수 있습니다. 단일 리소스에 여러 개의 보안 그룹을 할당할 수도 있습니다.
- 보안 그룹을 생성할 때 이름과 설명을 제공해야 합니다. 다음 규칙이 적용됩니다.
  - 보안 그룹 이름은 VPC 내에서 고유해야 합니다.
  - 보안 그룹 이름은 대/소문자를 구분하지 않습니다.
  - 이름과 설명은 최대 255자일 수 있습니다.
  - 이름과 설명은 다음과 같은 문자로 제한됩니다. a-z, A-Z, 0-9, 공백 및 ._-:/()#,@[]+=&;{}!$*
  - 이름에 후행 공백이 포함되어 있으면 이름 끝의 공백을 자릅니다. 예를 들어 이름에 “테스트 보안 그룹 ”을 입력하면 “테스트 보안 그룹”으로 저장됩니다.
  - 보안 그룹 이름은 `sg-`로 시작할 수 없습니다.
- 보안 그룹은 상태가 저장됩니다. 예를 들어 사용자가 인스턴스에서 요청을 전송하면 해당 요청의 응답 트래픽은 인바운드 보안 그룹 규칙에 관계없이 인스턴스에 도달할 수 있습니다. 허용된 인바운드 트래픽에 대한 응답은 아웃바운드 규칙에 관계없이 인스턴스를 떠날 수 있습니다.
- 보안 그룹은 다음에서 송수신되는 트래픽을 필터링하지 않습니다.
  - Amazon Domain Name Services(DNS)
  - Amazon Dynamic Host Configuration Protocol(DHCP)
  - Amazon EC2 인스턴스 메타데이터
  - Amazon ECS 태스크 메타데이터 엔드포인트
  - Windows 인스턴스에 대한 라이선스 활성화
  - Amazon Time Sync Service
  - 기본 VPC 라우터에서 사용하는 예약된 IP 주소
- VPC당 생성할 수 있는 보안 그룹의 개수, 각 보안 그룹에 추가할 수 있는 규칙의 개수, 그리고 네트워크 인터페이스에 연결할 수 있는 보안 그룹의 개수에는 할당량이 있습니다.

  자세한 내용은 Amazon VPC 할당량 단원을 참조하세요.

모범 사례

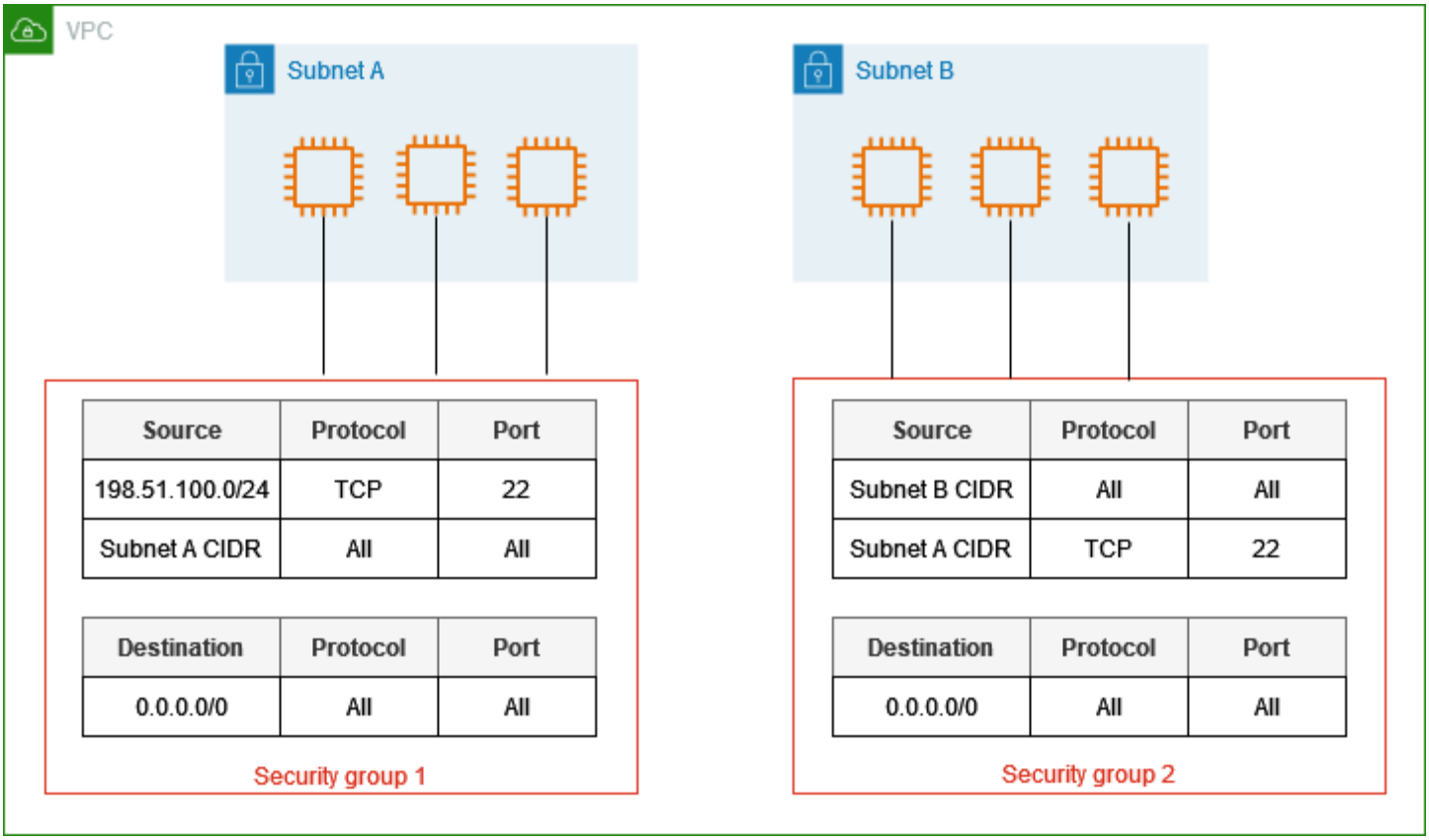
- 특정 IAM 보안 주체에만 보안 그룹을 생성 및 수정하는 권한을 부여합니다.

- 오류 위험을 줄이려면 필요한 최소 보안 그룹 수를 생성합니다. 각 보안 그룹을 사용하여 함수 및 보안 요구 사항이 비슷한 리소스에 대한 액세스 권한을 관리합니다.
- EC2 인스턴스에 액세스할 수 있도록 포트 22(SSH) 또는 3389(RDP)에 대한 인바운드 규칙을 추가하는 경우 특정 IP 주소 범위만 권한을 부여합니다. 0.0.0.0/0(IPv4)과 ::/(IPv6)를 지정하면 누구든지 지정된 프로토콜을 사용하여 모든 IP 주소에서 인스턴스에 액세스할 수 있게 됩니다.
- 큰 포트 범위를 열지 마세요. 각 포트를 통한 액세스 권한이 해당 포트를 필요로 하는 원본 또는 대상으로 제한되도록 하세요.
- 보안 그룹과 비슷한 규칙으로 네트워크 ACL을 생성하여 VPC에 추가 보안 계층을 추가해 보세요. 보안 그룹과 네트워크 ACL의 차이에 대한 자세한 정보는 보안 그룹 및 네트워크 ACL 비교 단원을 참조하세요.

보안 그룹 예시

다음 다이어그램에서는 보안 그룹과 서브넷이 각각 2개인 VPC를 보여줍니다. 서브넷 A의 인스턴스는 연결 요구 사항이 동일한 보안 그룹 1과 연결됩니다. 서브넷 B의 인스턴스는 연결 요구 사항이 동일한 보안 그룹 2와 연결됩니다. 보안 그룹 규칙에서는 다음의 트래픽이 허용됩니다.

- 보안 그룹 1의 첫 번째 인바운드 규칙은 지정된 주소 범위(예: 자체 네트워크의 범위)에서 서브넷 A의 인스턴스로 이동하는 SSH 트래픽을 허용합니다.
- 보안 그룹 1의 두 번째 인바운드 규칙에서는 서브넷 A의 인스턴스가 프로토콜과 포트를 사용하여 상호 간에 통신하는 것이 허용됩니다.
- 보안 그룹 2의 첫 번째 인바운드 규칙에서는 서브넷 B의 인스턴스가 프로토콜과 포트를 사용하여 상호 간에 통신하는 것이 허용됩니다.
- 보안 그룹 2의 두 번째 인바운드 규칙에서는 서브넷 A의 인스턴스가 SSH를 사용하여 서브넷 B의 인스턴스와 통신하는 것이 허용됩니다.
- 두 보안 그룹 모두 모든 트래픽을 허용하는 기본 아웃바운드 규칙을 사용합니다.



보안 그룹 규칙

보안 그룹의 규칙은 보안 그룹과 연결된 리소스에 도달하도록 허용된 인바운드 트래픽을 제어합니다. 인스턴스에서 나갈 수 있는 아웃바운드 트래픽을 제어합니다.

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 소스 또는 대상에 대한 액세스 권한을 부여할 수 있습니다.

내용

- 보안 그룹 규칙 기본 사항
- 보안 그룹 규칙의 구성 요소
- 보안 그룹 참조
- 보안 그룹 크기
- 무효 보안 그룹 규칙

보안 그룹 규칙 기본 사항

다음은 보안 그룹 규칙의 특징입니다.

- 허용 규칙을 지정할 수 있지만 거부 규칙은 지정할 수 없습니다.
- 보안 그룹을 처음 만들 때 인바운드 규칙이 없습니다. 따라서 보안 그룹에 인바운드 규칙을 추가하기 전에는 어떤 인바운드 트래픽도 허용되지 않습니다.
- 보안 그룹을 처음 생성하면 리소스의 모든 아웃바운드 트래픽을 허용하는 아웃바운드 규칙이 있습니다. 규칙을 제거할 수 있으며 특정 아웃바운드 트래픽만 허용하는 아웃바운드 규칙을 추가할 수 있습니다. 보안 그룹에 아웃바운드 규칙이 없는 경우 어떤 아웃바운드 트래픽도 허용되지 않습니다.
- 여러 보안 그룹을 리소스와 연결하면 각 보안 그룹의 규칙이 집계되어 액세스 허용 여부를 결정하는 데 사용되는 단일 규칙 집합을 형성합니다.
- 규칙을 추가, 업데이트 또는 제거할 때 변경 사항은 보안 그룹과 연결된 모든 리소스에 자동으로 적용됩니다. 지침은 보안 그룹 규칙 구성 단원을 참조하세요.
- 일부 규칙 변경 사항이 미치는 효과는 트래픽의 추적 방법에 따라 다를 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 연결 추적을 참조하세요.
- 보안 그룹 규칙을 생성하면 AWS에서는 규칙에 고유한 ID가 할당됩니다. API 또는 CLI를 사용하여 규칙을 수정하거나 삭제할 때 규칙의 ID를 사용할 수 있습니다.

제한 사항

보안 그룹은 'VPC+2 IP 주소'(Amazon Route 53 개발자 안내서의 Amazon Route 53 Resolver 참조) 또는 AmazonProvidedDNS라고 하는 Route 53 Resolver와 주고받는 DNS 요청을 차단할 수 없습니다. Route 53 Resolver를 통해 DNS 요청을 필터링하려면 Route 53 Resolver DNS 방화벽을 사용하세요.

보안 그룹 규칙의 구성 요소

다음은 인바운드 및 아웃바운드 보안 그룹 규칙의 구성 요소입니다.

- 프로토콜: 허용할 프로토콜. 가장 일반적인 프로토콜은 6(TCP), 17(UDP) 및 1(ICMP)입니다.
- 포트 범위: TCP, UDP 또는 사용자 지정 프로토콜의 경우 허용할 포트의 범위. 단일 포트 번호(예: 22) 또는 포트 번호의 범위(예: `7000-8000`)를 지정할 수 있습니다.
- ICMP 유형 및 코드: ICMP의 경우, ICMP 유형과 코드. 예를 들어 ICMP 에코 요청에 대해 유형 8을 사용하고 ICMPv6 에코 요청에 대해 유형 128을 입력합니다. 자세한 내용은 Amazon EC2 사용 설명서의 ping/ICMP 규칙을 참조하세요.
- 소스 또는 대상: 허용할 트래픽에 대한 소스(인바운드 규칙) 또는 대상(아웃바운드 규칙)입니다. 다음 중 하나를 지정하세요.

- 단일 IPv4 주소. /32 접두사 길이를 사용해야 합니다. 예: 203.0.113.1/32.
- 단일 IPv6 주소. /128 접두사 길이를 사용해야 합니다. 예: 2001:db8:1234:1a00::123/128.
- CIDR 블록 표기법으로 표시된 IPv4 주소의 범위. 예를 들어 203.0.113.0/24입니다.
- CIDR 블록 표기법으로 표시된 IPv6 주소의 범위. 예를 들어 2001:db8:1234:1a00::/64입니다.
- 접두사 목록의 ID. 예를 들어 pl-1234abc1234abc123입니다. 자세한 내용은 관리형 접두사 목록 섹션을 참조하세요.
- 보안 그룹의 ID. 예를 들어 sg-1234567890abcdef0입니다. 자세한 내용은 the section called "보안 그룹 참조" 단원을 참조하세요.
- (선택 사항) 설명: 나중에 쉽게 식별할 수 있도록 규칙에 대한 설명을 입력할 수 있습니다. 설명 길이는 최대 255자입니다. 허용되는 문자는 a-z, A-Z, 0-9, 공백 및 ._-:/()#,@[]+=;{}!$*입니다.

예제는 Amazon EC2 사용 설명서의 다양한 사용 사례의 보안 그룹 규칙을 참조하세요.

보안 그룹 참조

보안 그룹을 규칙의 소스 또는 대상으로 지정할 경우 규칙은 보안 그룹과 연결된 모든 인스턴스에 영향을 줍니다. 인스턴스는 인스턴스의 프라이빗 IP 주소를 사용하여 지정된 프로토콜 및 포트를 통해 지정된 방향으로 통신할 수 있습니다.

예를 들어 다음은 보안 그룹 sg-0abcdef1234567890을(를) 참조하는 보안 그룹에 대한 인바운드 규칙을 나타냅니다. 이 규칙에서는 sg-0abcdef1234567890과(와) 연결된 인스턴스에서 발생한 인바운드 SSH 트래픽이 허용됩니다.

소스	프로토콜	포트 범위
sg-0abcdef1234567890	TCP	22

보안 그룹 규칙에서 보안 그룹을 참조할 때 다음 사항에 유의하세요.

- 다음 중 하나에 해당하는 경우 다른 보안 그룹의 인바운드 규칙에서 보안 그룹을 참조할 수 있습니다.
  - 보안 그룹이 동일한 VPC에 연결되어 있습니다.
  - 보안 그룹이 연결된 VPC 간에 피어링 연결이 있습니다.
  - 보안 그룹이 연결된 VPC 간에 전송 게이트웨이가 있습니다.

- 다음 중 하나에 해당하는 경우 아웃바운드 규칙에서 보안 그룹을 참조할 수 있습니다.
  - 보안 그룹이 동일한 VPC에 연결되어 있습니다.
  - 보안 그룹이 연결된 VPC 간에 피어링 연결이 있습니다.
- 참조된 보안 그룹의 규칙은 해당 그룹을 참조하는 보안 그룹에 추가되지 않습니다.
- 인바운드 규칙의 경우 보안 그룹과 연결된 EC2 인스턴스는 참조된 보안 그룹과 연결된 EC2 인스턴스의 네트워크 인터페이스에 할당된 프라이빗 IP 주소로부터 인바운드 트래픽을 수신할 수 있습니다.
- 아웃바운드 규칙의 경우 보안 그룹과 연결된 EC2 인스턴스는 참조된 보안 그룹과 연결된 EC2 인스턴스의 네트워크 인터페이스에 할당된 프라이빗 IP 주소로 아웃바운드 트래픽을 보낼 수 있습니다.
- `AuthorizeSecurityGroupIngress`, `AuthorizeSecurityGroupEgress`, `RevokeSecurityGroupIngress` 및 `RevokeSecurityGroupEgress` 작업에서는 참조된 보안 그룹에 대해 권한을 부여하지 않습니다. 보안 그룹이 존재하는지 여부만 확인합니다. 이 결과는 다음과 같습니다.
  - 이러한 작업에 대해 IAM 정책에서 참조된 보안 그룹을 지정하는 것은 아무런 효과가 없습니다.
  - 참조된 보안 그룹을 다른 계정이 소유한 경우 소유자 계정은 이러한 작업에 대한 CloudTrail 이벤트를 수신하지 않습니다.

제한 사항

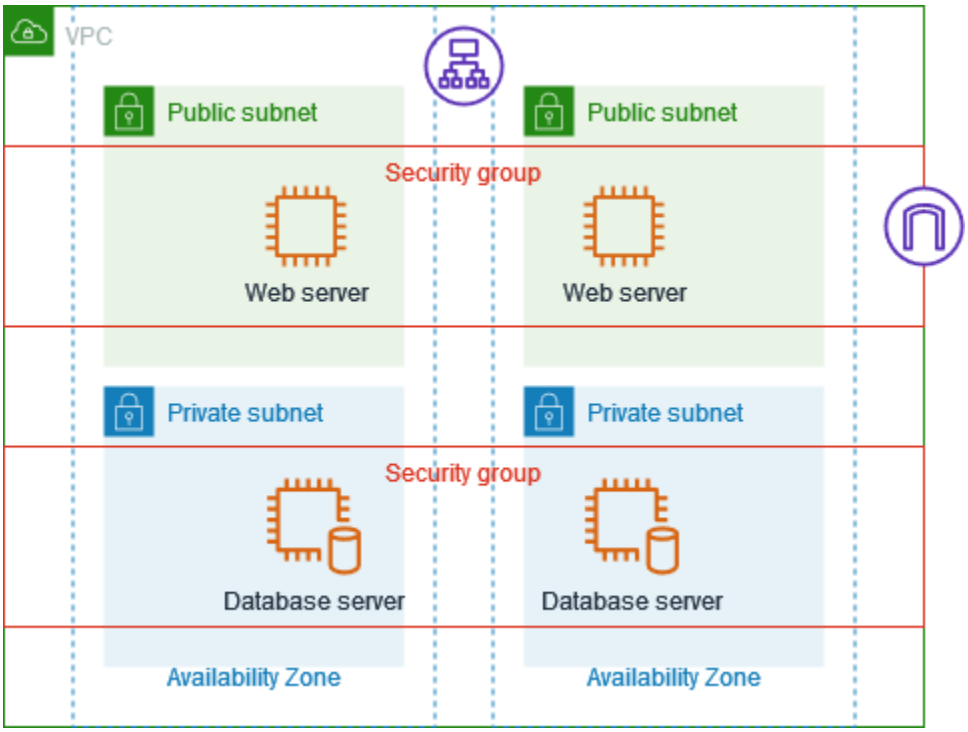
미들박스 어플라이언스를 통해 서로 다른 서브넷에 있는 두 인스턴스 간의 트래픽을 전달하도록 경로를 구성하는 경우 두 인스턴스에 대한 보안 그룹이 인스턴스 간에 트래픽이 흐르도록 허용해야 합니다. 각 인스턴스의 보안 그룹은 다른 인스턴스의 프라이빗 IP 주소 또는 다른 인스턴스가 포함된 서브넷의 CIDR 범위를 소스로 참조해야 합니다. 다른 인스턴스의 보안 그룹을 소스로 참조하면 인스턴스 간에 트래픽이 흐를 수 없습니다.

예제

다음 다이어그램은 두 개의 가용 영역에 있는 서브넷, 인터넷 게이트웨이 및 Application Load Balancer가 있는 VPC를 보여 줍니다. 각 가용 영역에는 웹 서버용 퍼블릭 서브넷과 데이터베이스 서버용 프라이빗 서브넷이 있습니다. 로드 밸런서, 웹 서버 및 데이터베이스 서버에는 별도의 보안 그룹이 있습니다. 다음 보안 그룹 규칙을 생성하여 트래픽을 허용하세요.

- 인터넷의 HTTP 및 HTTPS 트래픽을 허용하도록 로드 밸런서 보안 그룹에 규칙을 추가합니다. 소스는 0.0.0.0/0.입니다.
- 로드 밸런서의 HTTP 및 HTTPS 트래픽만 허용하도록 웹 서버의 보안 그룹에 규칙을 추가합니다. 소스는 로드 밸런서에 대한 보안 그룹입니다.

- 웹 서버의 데이터베이스 요청을 허용하도록 데이터베이스 서버의 보안 그룹에 규칙을 추가합니다. 소스는 웹 서버에 대한 보안 그룹입니다.



보안 그룹 크기

소스 또는 대상의 유형에 따라 각 규칙이 보안 그룹당 가질 수 있는 최대 규칙 수에 포함되는 방식이 결정됩니다.

- CIDR 블록을 참조하는 규칙은 하나의 규칙으로 계산됩니다.
- 다른 보안 그룹을 참조하는 규칙은 참조된 보안 그룹의 크기와 상관없이 하나의 규칙으로 계산됩니다.
- 고객이 관리하는 접두사 목록을 참조하는 규칙은 접두사 목록의 최대 크기로 계산됩니다. 예를 들어 접두사 목록의 최대 크기가 20인 경우 이 접두사 목록을 참조하는 규칙은 20개의 규칙으로 계산됩니다.
- AWS 관리형 접두사 목록을 참조하는 규칙은 접두사 목록의 가중치로 계산됩니다. 예를 들어 접두사 목록의 가중치가 10인 경우 이 접두사 목록을 참조하는 규칙은 10개의 규칙으로 계산됩니다. 자세한 내용은 the section called “사용 가능한 AWS 관리형 접두사 목록” 단원을 참조하세요.

무효 보안 그룹 규칙

VPC에 다른 VPC와의 VPC 피어링 연결이 있는 경우 또는 다른 계정에 의해 공유된 VPC를 사용하는 경우 VPC의 보안 그룹 규칙은 해당 피어 VPC 또는 공유 VPC의 보안 그룹을 참조할 수 있습니다. 이를 통해 참조된 보안 그룹과 연결된 리소스가 참조하는 보안 그룹과 연결된 리소스와 서로 통신할 수 있습니다. 자세한 내용은 Amazon VPC 피어링 가이드의 피어 VPC 보안 그룹을 참조하도록 보안 그룹 업데이트를 참조하세요.

피어 VPC 또는 공유 VPC의 보안 그룹을 참조하는 보안 그룹 규칙이 있는 경우, 참조된 보안 그룹이 삭제되거나 VPC 피어링 연결이 삭제되면 해당 규칙은 만료 상태로 표시됩니다. 다른 보안 그룹 규칙과 같은 방법으로 무효 보안 그룹 규칙을 삭제할 수 있습니다.

VPC에 대한 기본 보안 그룹

기본 VPC와 사용자가 생성한 VPC는 기본 보안 그룹과 함께 제공됩니다. 기본 보안 그룹의 이름은 “default“입니다.

기본 보안 그룹을 사용하는 대신 특정 리소스 또는 리소스 그룹에 대한 보안 그룹을 만드는 것이 좋습니다. 단, 일부 리소스는 생성 시점에 보안 그룹을 연결하지 않으면 기본 보안 그룹이 연결됩니다. 예를 들어 EC2 인스턴스를 시작할 때 보안 그룹을 지정하지 않은 경우 인스턴스는 그 VPC의 기본 보안 그룹과 연결됩니다.

기본 보안 그룹 기본 사항

- 기본 보안 그룹에 대한 규칙을 변경할 수 있습니다.
- 기본 보안 그룹을 삭제할 수 없습니다. 기본 보안 그룹을 삭제하려고 하면 `Client.CannotDelete`라는 오류 코드가 반환됩니다.

기본 규칙

다음 표에서는 기본 보안 그룹의 기본 인바운드 규칙을 설명합니다.

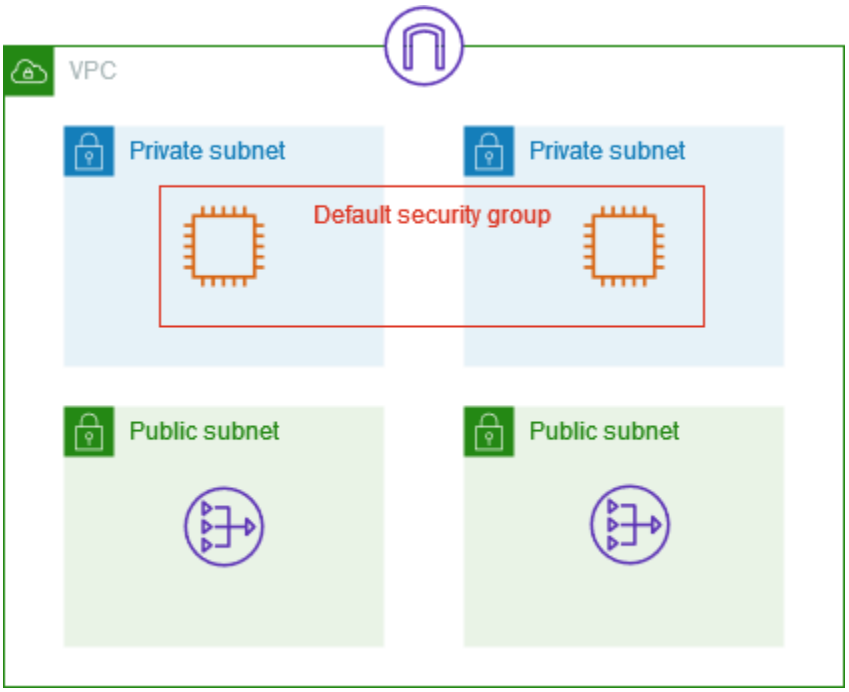
소스	프로토콜	포트 범위	설명
sg-1234567890abcdef0	모두	모두	이 보안 그룹에 할당된 모든 리소스로부터의 인바운드 트래픽을 허용합니다. 소스는 보안 그룹의 ID입니다.

다음 표에서는 기본 보안 그룹의 기본 아웃바운드 규칙을 설명합니다.

대상	프로토콜	포트 범위	설명
0.0.0.0/0	모두	모두	모든 아웃바운드 IPv4 트래픽을 허용합니다.
::/0	모두	모두	모든 아웃바운드 IPv6 트래픽을 허용합니다. 이 규칙은 VPC에 연결된 IPv6 CIDR 블록이 있는 경우에만 추가됩니다.

예제

다음 다이어그램은 기본 보안 그룹, 인터넷 게이트웨이 및 NAT 게이트웨이가 있는 VPC를 보여 줍니다. 기본 보안에는 기본 규칙만 포함되며 VPC에서 실행되는 두 개의 EC2 인스턴스와 연결됩니다. 이 시나리오에서 각 인스턴스는 모든 포트 및 프로토콜에서 다른 인스턴스로부터 인바운드 트래픽을 수신할 수 있습니다. 기본 규칙에서는 인스턴스가 인터넷 게이트웨이 또는 NAT 게이트웨이로부터 트래픽을 수신하는 것이 허용되지 않습니다. 인스턴스에서 반드시 추가 트래픽을 수신해야 하는 경우 필수 규칙이 있는 보안 그룹을 만들고 새 보안 그룹을 기본 보안 그룹 대신 인스턴스에 연결하는 것을 권장합니다.



VPC의 보안 그룹을 생성

가상 프라이빗 클라우드(VPC)에는 기본 보안 그룹이 제공됩니다. 추가 보안 그룹을 생성할 수 있습니다. 보안 그룹은 해당 보안 그룹이 생성된 VPC의 리소스에서만 사용할 수 있습니다.

기본적으로 처음에 새 보안 그룹에는 리소스에서 나가는 모든 트래픽을 허용하는 아웃바운드 규칙만 적용됩니다. 인바운드 트래픽을 사용하거나 아웃바운드 트래픽을 제한하려면 규칙을 추가해야 합니다. 보안 그룹을 생성할 때나 나중에 규칙을 추가할 수 있습니다. 자세한 내용은 보안 그룹 규칙 단원을 참조하세요.

필수 권한

시작하기 전에 필수 권한을 받았는지 확인하세요. 자세한 내용은 다음 자료를 참조하세요.

- 보안 그룹 관리
- 보안 그룹 규칙 관리

콘솔을 사용하여 보안 그룹을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹 생성을 선택합니다.
4. 보안 그룹의 이름과 설명을 입력합니다. 보안 그룹을 생성한 후에는 보안 그룹에 대한 이름과 설명을 변경할 수 없습니다.
5. VPC에서는 보안 그룹을 연결할 리소스를 생성할 VPC를 선택합니다.
6. (선택 사항) 인바운드 규칙을 추가하려면 인바운드 규칙을 선택합니다. 각 규칙에 대해 규칙 추가를 선택하고 프로토콜, 포트 및 소스를 지정합니다. 자세한 내용은 보안 그룹 규칙 구성 단원을 참조하세요.
7. (선택 사항) 아웃바운드 규칙을 추가하려면 아웃바운드 규칙을 선택합니다. 각 규칙에 대해 규칙 추가를 선택하고 프로토콜, 포트 및 대상을 지정합니다.
8. (선택 사항) 태그를 추가하려면 Add new tag(새 태그 추가)를 선택하고 태그 키와 태그 값을 입력합니다.
9. 보안 그룹 생성을 선택합니다.

AWS CLI를 사용하여 보안 그룹을 생성하려면

create-security-group 명령을 사용합니다.

또는 기존 보안 그룹을 복사하여 새 보안 그룹을 생성할 수도 있습니다. 보안 그룹을 복사하면 원래 보안 그룹과 동일한 인바운드 및 아웃바운드 규칙을 자동으로 추가하고 원래 보안 그룹과 동일한 VPC를 사용합니다. 새 보안 그룹의 이름과 설명을 입력할 수 있습니다. 필요에 따라 다른 VPC를 선택하고 필요에 따라 인바운드 및 아웃바운드 규칙을 수정할 수 있습니다. 그러나 보안 그룹을 한 리전에서 다른 리전으로 복사할 수 없습니다.

기존 보안 그룹을 기준으로 보안 그룹을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹을 선택합니다.
4. 작업, 새 보안 그룹에 복사를 선택합니다.
5. 보안 그룹의 이름과 설명을 입력합니다.
6. (선택 사항) 필요한 경우 다른 VPC를 선택합니다.
7. (선택 사항) 필요에 따라 보안 그룹 규칙을 추가, 제거 또는 편집합니다.
8. 보안 그룹 생성을 선택합니다.

보안 그룹 규칙 구성

보안 그룹을 생성한 후 해당 보안 그룹 규칙을 추가, 업데이트 및 삭제할 수 있습니다. 규칙을 추가, 업데이트 또는 삭제하면 변경 내용이 보안 그룹과 연결된 모든 리소스에 자동으로 적용됩니다.

필수 권한

시작하기 전에 필요한 권한이 있는지 확인하세요. 자세한 내용은 보안 그룹 규칙 관리 섹션을 참조하세요.

포트 및 프로토콜

- 콘솔에서 사전 정의된 유형을 선택하면 프로토콜 및 포트 범위가 지정됩니다. 포트 범위를 입력하려면 사용자 지정 TCP 또는 사용자 지정 UDP 중 하나를 선택해야 합니다.
- AWS CLI에서 `--protocol` 및 `--port` 옵션을 사용하여 단일 포트가 있는 단일 규칙을 추가할 수 있습니다. 여러 규칙 또는 포트 범위가 있는 규칙을 추가하려면 `--ip-permissions` 옵션을 대신 사용합니다.

소스 및 대상

- 콘솔에서 다음을 인바운드 규칙의 소스 또는 아웃바운드 규칙의 대상으로 지정할 수 있습니다.
  - 사용자 지정 - IPv4 CIDR 블록, IPv6 CIDR 블록, 보안 그룹 또는 접두사 목록.
  - Anywhere-IPv4 – 0.0.0.0/0 IPv4 CIDR 블록.
  - Anywhere-IPv6 – ::/0 IPv6 CIDR 블록.
  - 내 IP: 로컬 컴퓨터의 퍼블릭 IPv4 주소.
- AWS CLI에서 `--cidr` 옵션을 사용하여 IPv4 CIDR 블록을 지정하거나 `--source-group` 옵션을 사용하여 보안 그룹을 지정할 수 있습니다. 접두사 목록 또는 IPv6 CIDR 블록을 지정하려면 `--ip-permissions` 옵션을 사용합니다.

> **Warning**
>
> Anywhere-IPv4를 선택하면 모든 IPv4 주소의 트래픽이 허용됩니다. Anywhere-IPv6을 선택하면 모든 IPv6 주소의 트래픽이 허용됩니다. 리소스에 액세스해야 하는 특정 IP 주소 범위에만 권한을 부여하는 것이 가장 좋습니다.

콘솔을 사용하여 보안 그룹 규칙을 구성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹을 선택합니다.
4. 인바운드 규칙을 편집하려면 작업 또는 인바운드 규칙 탭에서 인바운드 규칙 편집을 선택합니다.

   a. 규칙을 추가하려면 규칙 추가를 선택한 다음 규칙의 유형, 프로토콜, 포트 및 소스를 입력합니다.

      유형이 TCP 또는 UDP인 경우 허용할 포트 범위를 입력해야 합니다. 사용자 지정 ICMP의 경우 프로토콜(Protocol)에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위(Port range)에서 코드 이름을 선택해야 합니다. 다른 유형에 대해 프로토콜과 포트 범위가 구성됩니다.

   b. 규칙을 업데이트하려면 필요에 따라 프로토콜, 설명 및 소스를 변경합니다. 하지만 소스 유형을 변경할 수는 없습니다. 예를 들어 소스가 IPv4 CIDR 블록인 경우 IPv6 CIDR 블록, 접두사 목록 또는 보안 그룹을 지정할 수 없습니다.

   c. 규칙을 삭제하려면 해당 삭제 버튼을 선택합니다.

5. 아웃바운드 규칙을 편집하려면 작업 또는 아웃바운드 규칙 탭에서 아웃바운드 규칙 편집을 선택합니다.

   a. 규칙을 추가하려면 규칙 추가를 선택한 다음 규칙의 유형, 프로토콜, 포트 및 대상을 입력합니다. 또한 설명을 입력할 수 있습니다(선택 사항).

      유형이 TCP 또는 UDP인 경우 허용할 포트 범위를 입력해야 합니다. 사용자 지정 ICMP의 경우 프로토콜(Protocol)에서 ICMP 유형 이름을 선택하고, 해당되는 경우 포트 범위(Port range)에서 코드 이름을 선택해야 합니다. 다른 유형에 대해 프로토콜과 포트 범위가 구성됩니다.

   b. 규칙을 업데이트하려면 필요에 따라 프로토콜, 설명 및 소스를 변경합니다. 하지만 소스 유형을 변경할 수는 없습니다. 예를 들어 소스가 IPv4 CIDR 블록인 경우 IPv6 CIDR 블록, 접두사 목록 또는 보안 그룹을 지정할 수 없습니다.

   c. 규칙을 삭제하려면 해당 삭제 버튼을 선택합니다.

6. 규칙 저장을 선택합니다.

AWS CLI를 사용하여 보안 그룹 규칙 구성

- 추가 - authorize-security-group-ingress 및 authorize-security-group-egress 명령을 사용합니다.
- 제거 - revoke-security-group-ingress 및 revoke-security-group-egress 명령을 사용합니다.
- 수정 - modify-security-group-rules, update-security-group-rule-descriptions-ingress 및 update-security-group-rule-descriptions-egress 명령을 사용합니다.

보안 그룹 삭제

생성한 보안 그룹 사용을 완료하면 이를 삭제할 수 있습니다.

요구 사항

- 보안 그룹은 어떤 리소스와도 연결할 수 없습니다.
- 보안 그룹은 다른 보안 그룹의 규칙에서 참조할 수 없습니다.
- 보안 그룹은 VPC의 기본 보안 그룹이 될 수 없습니다.

콘솔을 사용하여 보안 그룹을 삭제하려면

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

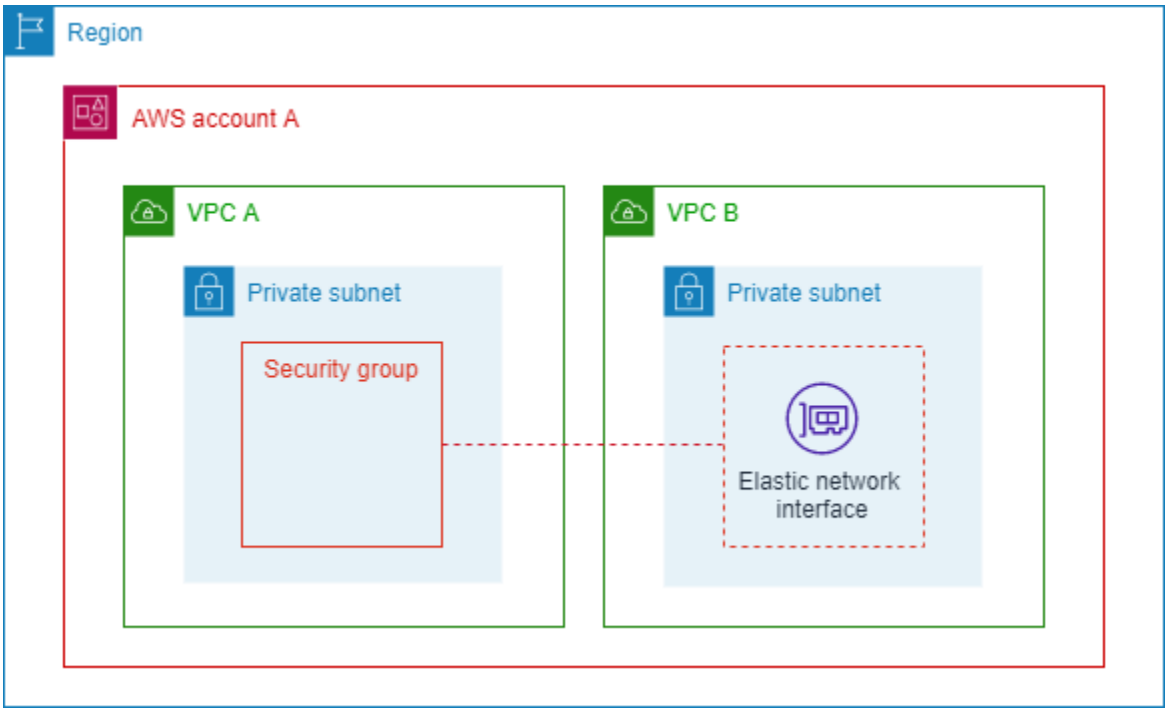
2. 탐색 창에서 Security groups를 선택합니다.
3. 보안 그룹을 선택한 다음 작업, 보안 그룹 삭제를 선택합니다.
4. 보안 그룹을 둘 이상 선택하면 확인 메시지가 표시됩니다. 일부 보안 그룹을 삭제할 수 없는 경우 각 보안 그룹의 상태가 표시되어 삭제 여부를 나타냅니다. 삭제를 확인하려면 삭제를 누릅니다.
5. 삭제를 선택합니다.

AWS CLI를 사용하여 보안 그룹을 삭제하려면

delete-security-group 명령을 사용합니다.

보안 그룹을 여러 VPC와 연결

네트워크 보안 요구 사항을 공유하는 여러 VPC에서 워크로드를 실행하는 경우 보안 그룹 VPC 연결 기능을 사용하여 보안 그룹을 동일한 리전의 여러 VPC와 연결할 수 있습니다. 이를 통해 계정의 여러 VPC에 대한 보안 그룹을 한 곳에서 관리하고 유지할 수 있습니다.



위의 다이어그램은 두 개의 VPC가 있는 AWS 계정 A를 보여줍니다. 각 VPC에는 프라이빗 서브넷에서 실행되는 워크로드가 있습니다. 이 경우 VPC A 및 B의 서브넷에 있는 워크로드들은 동일한 네트워크 트래픽 요구 사항을 공유하므로 계정 A는 보안 그룹 VPC 연결 기능을 사용하여 VPC A의 보안 그룹을 VPC B와 연결할 수 있습니다. 연결된 보안 그룹에 대한 모든 업데이트는 VPC B 서브넷의 워크로드에 대한 트래픽에 자동으로 적용됩니다.

보안 그룹 VPC 연결 기능의 요구 사항

- 보안 그룹을 VPC와 연결하려면 VPC를 소유하거나 VPC 서브넷 중 하나를 공유해야 합니다.
- VPC와 보안 그룹이 동일한 AWS 리전에 있어야 합니다.
- 기본 보안 그룹을 다른 VPC와 연결하거나 보안 그룹을 기본 VPC와 연결할 수 없습니다.
- 보안 그룹 소유자와 VPC 소유자 모두 보안 그룹 VPC 연결을 볼 수 있습니다.

이 기능을 지원하는 서비스

- Amazon API Gateway(REST API만 해당)
- AWS Auto Scaling
- CloudFormation
- Amazon EC2
- Amazon EFS
- Amazon EKS
- Amazon FSx
- AWS PrivateLink
- Amazon Route 53
- Elastic Load Balancing
  - Application Load Balancer
  - Network Load Balancer

보안 그룹을 다른 VPC에 연결

이 섹션에서는 AWS Management Console 및 AWS CLI를 사용하여 보안 그룹을 VPC와 연결하는 방법을 설명합니다.

AWS Management Console

보안 그룹을 다른 VPC와 연결하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택하여 세부 정보를 표시합니다.

4. VPC 연결 탭을 선택합니다.
5. VPC 연결을 선택합니다.
6. VPC ID에서 보안 그룹과 연결할 VPC를 선택합니다.
7. VPC 연결을 선택합니다.

Command line

보안 그룹을 다른 VPC와 연결하려면

1. associate-security-group-vpc를 사용하여 VPC 연결을 생성합니다.
2. describe-security-group-vpc-associations를 사용하여 VPC 연결의 상태를 확인하고 상태가 `associated`가 될 때까지 기다립니다.

이제 VPC가 보안 그룹과 연결됩니다.

VPC에 보안 그룹을 연결한 후에는 예를 들어 VPC 내에서 인스턴스를 시작하고 이 새 보안 그룹을 선택하거나 기존 보안 그룹 규칙에서 이 보안 그룹을 참조할 수 있습니다.

다른 VPC에서 보안 그룹 연결 해제

이 섹션에서는 AWS Management Console 및 AWS CLI를 사용하여 VPC에서 보안 그룹을 연결 해제하는 방법을 설명합니다. 보안 그룹을 삭제하는 것이 목표인 경우 이 작업을 수행할 수 있습니다. 연결된 보안 그룹은 삭제할 수 없습니다. 연결된 VPC에서 보안 그룹을 사용하는 네트워크 인터페이스가 없는 경우에만 해당 보안 그룹을 연결 해제할 수 있습니다.

AWS Management Console

VPC에서 보안 그룹을 연결 해제하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택하여 세부 정보를 표시합니다.
4. VPC 연결 탭을 선택합니다.
5. VPC 연결 해제를 선택합니다.
6. VPC ID에서 보안 그룹과의 연결을 해제할 VPC를 선택합니다.
7. VPC 연결 해제를 선택합니다.

8. VPC 연결 탭에서 연결 해제 상태를 확인하고 상태가 `disassociated`가 될 때까지 기다립니다.

Command line

VPC에서 보안 그룹을 연결 해제하려면

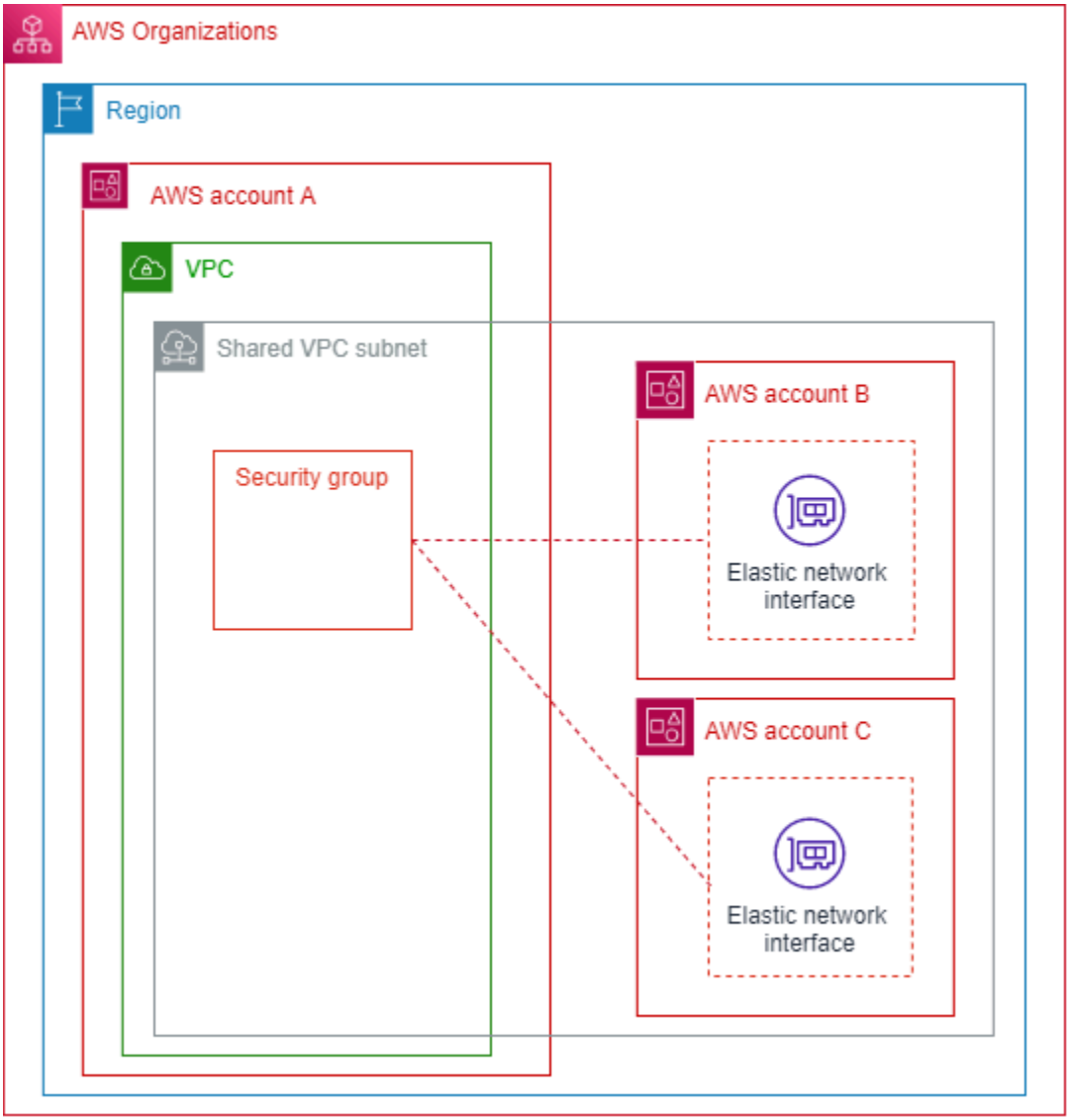
1. disassociate-security-group-vpc를 사용하여 VPC 연결을 연결 해제합니다.
2. describe-security-group-vpc-associations를 사용하여 VPC 연결 해제의 상태를 확인하고 상태가 `disassociated`가 될 때까지 기다립니다.

이제 VPC가 보안 그룹과 연결 해제됩니다.

AWS Organizations와 보안 그룹 공유

공유 보안 그룹 기능을 사용하면 보안 그룹을 동일한 AWS 리전 내의 다른 AWS Organizations 계정과 공유하고 해당 계정에서 보안 그룹을 사용할 수 있도록 만들 수 있습니다.

다음 다이어그램은 공유 보안 그룹 기능을 사용하여 AWS Organizations의 계정 전체에서 보안 그룹 관리를 간소화하는 방법을 보여줍니다.



이 다이어그램은 동일한 조직의 일부인 세 개의 계정을 보여줍니다. 계정 A는 계정 B 및 C와 VPC 서브넷을 공유합니다. 계정 A는 공유 보안 그룹 기능을 사용하여 계정 B 및 C와 보안 그룹을 공유합니다. 계속해서 계정 B 및 C는 공유 서브넷에서 인스턴스를 시작할 때 해당 보안 그룹을 사용합니다. 이렇게 하면 계정 A가 보안 그룹을 관리할 수 있으며 보안 그룹에 대한 모든 업데이트는 계정 B 및 C가 공유 VPC 서브넷에서 실행한 리소스에 적용됩니다.

공유 보안 그룹 기능의 요구 사항

- 이 기능은 AWS Organizations에서 동일한 조직에 있는 계정에서만 사용할 수 있습니다. AWS Organizations에서 리소스 공유를 활성화해야 합니다.
- 보안 그룹을 공유하는 계정은 VPC와 보안 그룹을 모두 소유해야 합니다.
- 기본 보안 그룹은 공유할 수 없습니다.

- 기본 VPC에 있는 보안 그룹은 공유할 수 없습니다.
- 참가자 계정은 공유 VPC에서 보안 그룹을 생성할 수 있지만 해당 보안 그룹은 공유할 수는 없습니다.
- IAM 보안 주체가 AWS RAM과 보안 그룹을 공유하려면 최소 권한 세트가 필요합니다. `AmazonEC2FullAccess` 및 `AWSResourceAccessManagerFullAccess` 관리형 IAM 정책을 사용하여 IAM 보안 주체가 공유 보안 기능을 공유하고 사용하는 데 필요한 권한을 갖추게 해야 합니다. 사용자 지정 IAM 정책을 사용하는 경우 `c2:PutResourcePolicy` 및 `ec2:DeleteResourcePolicy` 작업이 필요합니다. 이는 권한 전용 IAM 작업입니다. IAM 보안 주체에게 이러한 권한이 부여되어 있지 않은 경우, AWS RAM을 사용하여 보안 그룹을 공유하려고 하면 오류가 발생합니다.

이 기능을 지원하는 서비스

- Amazon API Gateway
- Amazon EC2
- Amazon ECS
- Amazon EFS
- Amazon EKS
- Amazon EMR
- Amazon FSx
- Amazon ElastiCache
- AWS Elastic Beanstalk
- AWS Glue
- Amazon MQ
- Amazon SageMaker AI
- Elastic Load Balancing
  - Application Load Balancer
  - Network Load Balancer

이 기능이 기존 할당량에 미치는 영향

보안 그룹 할당량이 적용됩니다. 그러나 '네트워크 인터페이스당 보안 그룹' 할당량의 경우 참가자가 탄력적 네트워크 인터페이스(ENI)에서 소유 그룹과 공유 그룹을 모두 사용하는 경우 소유자 및 참자자의 최소 할당량이 적용됩니다.

할당량이 이 기능의 영향을 받는 방식을 보여주는 예:

- 소유자 계정 할당량: 인터페이스당 보안 그룹 4개
- 참가자 계정 할당량: 인터페이스당 보안 그룹 5개.
- 소유자는 그룹 SG-O1, SG-O2, SG-O3, SG-O4, SG-O5를 참가자와 공유합니다. 참가자는 이미 VPC에 SG-P1, SG-P2, SG-P3, SG-P4, SG-P5의 자체 그룹이 있습니다.
- 참가자가 ENI를 생성하고 소유 그룹만 사용하는 경우 할당량에 해당하므로 5개 보안 그룹(SG-P1, SG-P2, SG-P3, SG-P4, SG-P5)을 모두 연결할 수 있습니다.
- 참가자가 ENI를 생성하고 ENI에서 공유 그룹을 사용하는 경우 최대 4개의 그룹만 연결할 수 있습니다. 이 경우 이러한 ENI의 할당량은 소유자 및 참가자의 최소 할당량입니다. 가능한 유효한 구성은 다음과 같습니다.
  - SG-O1, SG-P1, SG-P2, SG-P3
  - SG-O1, SG-O2, SG-O3, SG-O4

보안 그룹 공유

이 섹션에서는 AWS Management Console 및 AWS CLI를 사용하여 조직의 다른 계정과 보안 그룹을 공유하는 방법을 설명합니다.

AWS Management Console

보안 그룹을 공유하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택하여 세부 정보를 표시합니다.
4. 공유 탭을 선택합니다.
5. 보안 그룹 공유를 선택합니다.
6. 리소스 공유 생성을 선택합니다. 그러면 AWS RAM 콘솔이 열리며, 여기에서 보안 그룹에 대한 리소스 공유를 생성할 수 있습니다.
7. 리소스 공유의 이름을 입력합니다.
8. 리소스 - 선택 사항에서 보안 그룹을 선택합니다.

9. 보안 그룹을 선택합니다. 보안 그룹은 기본 보안 그룹일 수 없으며 기본 VPC와 연결할 수 없습니다.
10. 다음을 선택합니다.
11. 보안 주체가 수행하도록 허용할 작업을 검토하고 다음을 선택합니다.
12. 보안 주체 - 선택 사항 아래에서 조직 내에서만 공유 허용을 선택합니다.
13. 보안 주체에서 다음 보안 주체 유형 중 하나를 선택하고 적절한 숫자를 입력합니다.

    - AWS 계정: 조직에 있는 계정의 계정 번호입니다.
    - 조직: AWS Organizations ID입니다.
    - 조직 단위(OU): 조직에 있는 OU의 ID입니다.
    - IAM 역할: IAM 역할의 ARN입니다. 역할을 생성한 계정은 이 리소스 공유를 생성하는 계정과 동일한 조직의 멤버여야 합니다.
    - IAM 사용자: IAM 사용자의 ARN입니다. 사용자를 생성한 계정은 이 리소스 공유를 생성하는 계정과 동일한 조직의 멤버여야 합니다.
    - 서비스 보안 주체: 보안 그룹을 서비스 보안 주체와 공유할 수 없습니다.
14. 추가를 선택합니다.
15. 다음을 선택합니다.
16. 리소스 공유 생성을 선택합니다.
17. 공유 리소스에서 `Associated` 상태가 표시될 때까지 기다립니다. 보안 그룹 연결에 실패하는 경우 위에 나열된 제한 사항 중 하나 때문일 수 있습니다. 보안 그룹의 세부 정보를 표시하고 세부 정보 페이지의 공유 탭에서 보안 그룹을 공유할 수 없는 이유와 관련된 메시지를 확인합니다.
18. VPC 콘솔 보안 그룹 목록으로 돌아갑니다.
19. 공유한 보안 그룹을 선택합니다.
20. 공유 탭을 선택합니다. AWS RAM 리소스가 여기에 표시되어야 합니다. 그렇지 않은 경우 리소스 공유 생성이 실패했을 수 있으므로 다시 생성해야 할 수 있습니다.

Command line

보안 그룹을 공유하려면

1. 먼저와 AWS RAM과 공유하려는 보안 그룹에 대한 리소스 공유를 생성해야 합니다. AWS CLI를 사용하여 AWS RAM과의 리소스 공유를 생성하는 방법에 대한 단계는 AWS RAM 사용 설명서의 AWS RAM에서 리소스 공유 생성을 참조하세요.

2. 생성된 리소스 공유 연결을 보려면 get-resource-share-associations를 사용합니다.

이제 보안 그룹이 공유됩니다. 동일한 VPC 내의 공유 서브넷에서 EC2 인스턴스를 시작할 때 보안 그룹을 선택할 수 있습니다.

보안 그룹 공유 중지

이 섹션에서는 AWS Management Console 및 AWS CLI를 사용하여 조직의 다른 계정과 공유한 보안 그룹의 공유를 중지하는 방법을 설명합니다.

AWS Management Console

보안 그룹 공유를 중지하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 보안 그룹을 선택합니다.
3. 보안 그룹을 선택하여 세부 정보를 표시합니다.
4. 공유 탭을 선택합니다.
5. 보안 그룹 리소스 공유를 선택하고 공유 중지를 선택합니다.
6. 예, 공유 중지를 선택합니다.

Command line

보안 그룹 공유를 중지하려면

delete-resource-share를 사용하여 리소스 공유를 삭제합니다.

보안 그룹이 더 이상 공유되지 않습니다. 소유자가 보안 그룹 공유를 중지하면 다음 규칙이 적용됩니다.

- 기존 참가자의 탄력적 네트워크 인터페이스(ENI)는 공유되지 않은 보안 그룹에 대한 모든 보안 그룹 규칙 업데이트를 계속 가져옵니다. 공유를 해제하면 참가자가 공유되지 않은 그룹과의 새 연결을 생성할 수 없습니다.
- 참가자는 더 이상 공유되지 않은 보안 그룹을 자신이 소유한 ENI와 연결할 수 없습니다.
- 참가자는 공유되지 않은 보안 그룹과 여전히 연결되어 있는 ENI를 설명하고 삭제할 수 있습니다.

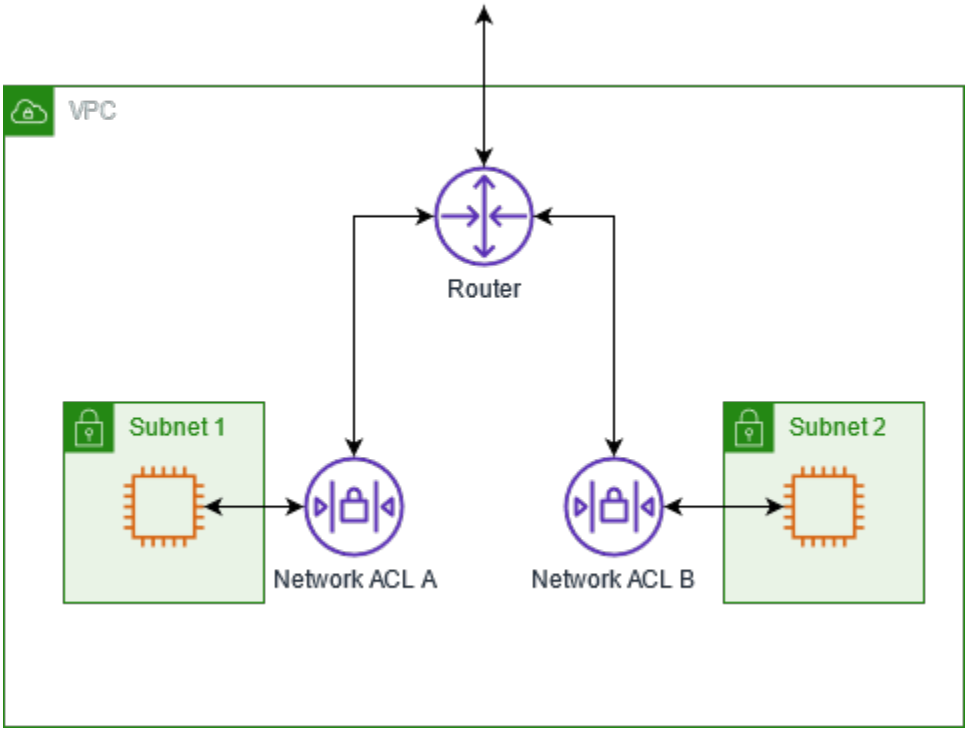
- 참가자에게 공유되지 않은 보안 그룹과 연결된 ENI가 여전히 있는 경우 소유자는 공유되지 않은 보안 그룹을 삭제할 수 없습니다. 소유자는 참가자가 모든 ENI에서 보안 그룹의 연결을 해제(제거)한 후에만 보안 그룹을 삭제할 수 있습니다.
- 참가자는 공유되지 않은 보안 그룹과 연결된 ENI를 사용하여 새 EC2 인스턴스를 시작할 수 없습니다.

네트워크 액세스 제어 목록으로 서브넷 트래픽 제어

네트워크 액세스 제어 목록(ACL)은 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용하거나 거부합니다. VPC의 기본 네트워크 ACL을 사용하거나, 보안 그룹 규칙과 유사한 규칙으로 VPC용 사용자 지정 네트워크 ACL을 생성할 수 있습니다. 사용자 지정 네트워크 ACL은 VPC에 추가 보안 계층을 제공합니다.

네트워크 ACL을 사용해도 추가 요금이 부과되지 않습니다.

다음 다이어그램에서는 서브넷이 2개인 VPC를 보여줍니다. 각 서브넷에 네트워크 ACL이 있습니다. 트래픽이 VPC로 들어오면(예: 피어링된 VPC, VPN 연결 또는 인터넷의 트래픽) 라우터에서 트래픽을 대상으로 보냅니다. 네트워크 ACL A에서는 서브넷 1로 향하는 트래픽 중 서브넷 1로 들어가도록 허용되는 트래픽과 서브넷 1 외부 위치로 향하는 트래픽 중 서브넷 1에서 나가도록 허용되는 트래픽을 결정합니다. 마찬가지로, 네트워크 ACL B에서는 서브넷 2에 들어오고 나갈 수 있는 트래픽을 결정합니다.



보안 그룹과 네트워크 ACL의 차이점에 대한 자세한 내용은 보안 그룹 및 네트워크 ACL 비교를 참조하세요.

내용

- 네트워크 ACL 기본 사항
- 네트워크 ACL 규칙
- VPC의 기본 네트워크 ACL
- VPC의 사용자 지정 네트워크 ACL
- 경로 MTU 검색 및 네트워크 ACL
- VPC에 대한 네트워크 ACL 만들기
- VPC에 대한 네트워크 ACL 연결 관리
- VPC의 네트워크 ACL 삭제
- 예: 서브넷의 인스턴스에 대한 액세스 제어

네트워크 ACL 기본 사항

다음은 시작하기 전에 네트워크 ACL에 대해 알아야 할 기본 사항입니다.

네트워크 ACL 연결

- VPC에 있는 각 서브넷을 네트워크 ACL과 연결해야 합니다. 서브넷을 네트워크 ACL에 명시적으로 연결하지 않을 경우, 서브넷은 기본 네트워크 ACL에 자동적으로 연결됩니다.
- 사용자 지정 네트워크 ACL을 생성하고 서브넷과 연결하여 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용하거나 거부할 수 있습니다.
- 네트워크 ACL을 여러 서브넷과 연결할 수 있습니다. 그러나 서브넷은 한 번에 하나의 네트워크 ACL에만 연결할 수 있습니다. 네트워크 ACL을 서브넷과 연결하면 이전 연결은 제거됩니다.

네트워크 ACL 규칙

- 네트워크 ACL에는 인바운드 규칙과 아웃바운드 규칙이 있습니다. 네트워크 ACL당 보유할 수 있는 규칙 수에는 할당량(또는 제한)이 있습니다. 각 규칙에서는 트래픽을 허용하거나 거부할 수 있습니다. 각 규칙에는 1부터 32766까지 번호가 있습니다. 규칙은 트래픽 허용 또는 거부가 결정될 때 가장 낮은 번호의 규칙부터 순서대로 평가됩니다. 트래픽이 규칙과 일치하면 규칙이 적용되며 추가 규칙은 평가되지 않습니다. 필요한 경우 나중에 새 규칙을 삽입할 수 있도록 증분 방식으로(예: 10 또는 100 단위씩 증분) 규칙을 생성하여 시작하는 것이 좋습니다.

- 네트워크 ACL 규칙은 트래픽이 서브넷 내에서 라우팅될 때가 아니라 서브넷에 들어오고 나갈 때 평가됩니다.
- NACL은 상태 비저장 목록이므로 이전에 전송했거나 수신한 트래픽에 대한 정보가 저장되지 않습니다. 예를 들어, 서브넷에 대한 특정 인바운드 트래픽을 허용하는 NACL 규칙을 생성하는 경우 해당 트래픽에 대한 응답이 자동으로 허용되지 않습니다. 이는 보안 그룹의 작동 방식과 대조적입니다. 보안 그룹은 상태 저장 그룹이므로 이전에 전송했거나 수신한 트래픽에 대한 정보가 저장됩니다. 예를 들어, 보안 그룹이 EC2 인스턴스에 대한 인바운드 트래픽을 허용하는 경우 아웃바운드 보안 그룹 규칙에 관계없이 응답이 자동으로 허용됩니다.

제한 사항

- 네트워크 ACL에 대한 할당량(제한이라고도 함)이 있습니다. 자세한 내용은 네트워크 ACL 섹션을 참조하세요.
- 네트워크 ACL은 Route 53 Resolver(VPC+2 IP 주소 또는 AmazonProvidedDNS라고도 함)에서 송수신되는 DNS 요청을 차단할 수 없습니다. Route 53 Resolver를 통해 DNS 요청을 필터링하려면 Route 53 Resolver DNS 방화벽을 활성화할 수 있습니다.
- 네트워크 ACL은 인스턴스 메타데이터 서비스(IMDS)에 대한 트래픽을 차단할 수 없습니다. IMDS에 대한 액세스를 관리하려면 Amazon EC2 사용 설명서의 인스턴스 메타데이터 옵션 구성을 참조하세요.
- 네트워크 ACL은 다음에서 송수신되는 트래픽을 필터링하지 않습니다.
  - Amazon Domain Name Services(DNS)
  - Amazon Dynamic Host Configuration Protocol(DHCP)
  - Amazon EC2 인스턴스 메타데이터
  - Amazon ECS 태스크 메타데이터 엔드포인트
  - Windows 인스턴스에 대한 라이선스 활성화
  - Amazon Time Sync Service
  - 기본 VPC 라우터에서 사용하는 예약된 IP 주소

네트워크 ACL 규칙

기본 네트워크 ACL에 규칙을 추가 또는 제거하거나, VPC에 대한 네트워크 ACL을 추가로 생성할 수 있습니다. 네트워크 ACL에서 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.
- 유형. 트래픽 유형(예: SSH)입니다. 모든 트래픽 또는 사용자 지정 범위를 지정할 수도 있습니다.
- 프로토콜 . 표준 프로토콜 번호를 가진 어떤 프로토콜이든 지정할 수 있습니다. 자세한 내용은 프로토콜 번호를 참조하세요. ICMP를 프로토콜로 지정하면 ICMP 유형과 코드 중 일부 또는 전부를 지정할 수 있습니다.
- 포트 범위. 트래픽에 대한 수신 포트 또는 포트 범위입니다. 예를 들어, HTTP 트래픽의 경우 80입니다.
- 소스: . [인바운드 규칙만 해당] 트래픽의 소스(CIDR 범위)입니다.
- 대상 [아웃바운드 규칙만 해당] 트래픽의 대상(CIDR 범위)입니다.
- 허용/거부. 지정된 트래픽을 허용 또는 거부 할지 여부입니다.

예시 규칙은 the section called “예: 서브넷의 인스턴스에 대한 액세스 제어” 섹션을 참조하세요.

고려 사항

- 네트워크 ACL당 규칙 수에 대한 할당량(제한이라고도 함)이 있습니다. 자세한 내용은 Amazon VPC 할당량 섹션을 참조하세요.
- ACL에서 규칙을 추가하거나 삭제할 때 ACL과 연관된 서브넷이 변경될 수 있습니다. 변경 사항은 잠시 후 적용됩니다.
- 명령줄 도구 또는 Amazon EC2 API를 사용하여 규칙을 추가하면 CIDR 범위가 표준 형식으로 자동 수정됩니다. 예를 들어 CIDR 범위에 `100.68.0.18/18`을 지정하면 `100.68.0.0/18` CIDR 범위를 가진 규칙이 작성됩니다.
- 다양한 포트를 열어야 하지만, 거부하려는 범위에 속하는 특정 포트가 있는 경우 거부 규칙을 추가할 수 있습니다. 거부 규칙에는 더 넓은 범위의 포트 트래픽을 허용하는 규칙보다 적은 수를 지정해야 합니다.
- 네트워크 ACL에서 규칙을 동시에 추가하고 삭제하는 경우 주의해야 합니다. 인바운드 또는 아웃바운드 규칙을 삭제한 다음 허용된 항목보다 많은 새 항목을 추가하면(Amazon VPC 할당량 참조) 삭제하도록 선택한 항목이 제거되고 새 항목이 추가되지 않습니다. 이로 인해 예기치 않은 연결 문제가 발생하고 VPC에 대한 액세스가 차단될 수 있습니다.

VPC의 기본 네트워크 ACL

가상 프라이빗 클라우드(VPC)에는 기본 네트워크 ACL이 자동으로 제공됩니다. 기본 네트워크 ACL은 연결된 서브넷 안팎으로 전송되는 트래픽 흐름을 모두 허용하도록 구성되어 있습니다. 각 네트워크

ACL에는 규칙 번호가 별표(*)로 되어 있는 규칙도 포함되어 있습니다. 이러한 규칙은 패킷이 번호가 매겨진 다른 어떤 규칙과도 일치하지 않을 경우에는 거부되도록 되어 있습니다.

규칙을 추가하거나 번호가 매겨진 기본 규칙을 제거하여 기본 네트워크 ACL을 수정할 수 있습니다. 규칙 번호가 별표인 규칙은 삭제할 수 없습니다.

기본 인바운드 규칙

다음 표에서는 기본 네트워크 ACL의 기본 인바운드 규칙을 보여줍니다. IPv6에 대한 규칙은 연결된 IPv6 CIDR 블록으로 VPC를 생성하거나 IPv6 CIDR 블록을 VPC와 연결하는 경우에만 추가됩니다. 하지만 기본 네트워크 ACL의 인바운드 규칙을 수정한 경우 IPv6 블록을 VPC에 연결할 때 모든 인바운드 IPv6 트래픽을 허용하는 규칙이 추가되지는 않습니다.

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

기본 아웃바운드 규칙

다음 표에서는 기본 네트워크 ACL의 기본 아웃바운드 규칙을 보여줍니다. IPv6에 대한 규칙은 연결된 IPv6 CIDR 블록으로 VPC를 생성하거나 IPv6 CIDR 블록을 VPC와 연결하는 경우에만 추가됩니다. 하지만 기본 네트워크 ACL의 아웃바운드 규칙을 수정한 경우 IPv6 블록을 VPC에 연결할 때 모든 아웃바운드 IPv6 트래픽을 허용하는 규칙이 추가되지는 않습니다.

규칙 #	유형	프로토콜	포트 범위	대상 주소	허용/거부
100	모든 트래픽	모두	모두	0.0.0.0/0	허용
101	모든 IPv6 트래픽	모두	모두	::/0	허용

규칙 #	유형	프로토콜	포트 범위	대상 주소	허용/거부
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY
*	모든 IPv6 트래픽	모두	모두	::/0	DENY

VPC의 사용자 지정 네트워크 ACL

사용자 지정 네트워크 ACL을 생성하고 서브넷과 연결하여 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용하거나 거부할 수 있습니다. 자세한 내용은 the section called “네트워크 ACL 생성” 섹션을 참조하세요.

각 네트워크 ACL에는 규칙 번호가 별표(*)로 된 기본 인바운드 규칙과 기본 아웃바운드 규칙이 포함되어 있습니다. 이러한 규칙은 패킷이 다른 어떤 규칙과도 일치하지 않을 경우에는 거부되도록 되어 있습니다.

규칙을 추가하거나 제거하여 네트워크 ACL을 수정할 수 있습니다. 규칙 번호가 별표인 규칙은 삭제할 수 없습니다.

추가하는 모든 규칙에 대해 응답 트래픽을 허용하는 인바운드 또는 아웃바운드 규칙이 있어야 합니다. 적절한 휘발성 포트 범위를 선택하는 방법에 대한 자세한 내용은 휘발성 포트 단원을 참조하세요.

인바운드 규칙의 예

다음 표에는 네트워크 ACL의 인바운드 규칙 예가 나와 있습니다. IPv6에 대한 규칙은 VPC에 연결된 IPv6 CIDR 블록이 있는 경우에만 추가됩니다. IPv4 및 IPv6 트래픽은 각각 개별적으로 평가됩니다. 따라서 IPv4 트래픽에 대한 규칙은 IPv6 트래픽에 적용되지 않습니다. 해당하는 IPv4 규칙 옆에 IPv6 규칙을 추가하거나, 마지막 IPv4 규칙 뒤에 IPv6 규칙을 추가할 수 있습니다.

패킷이 서브넷에 도달할 때, 이를 서브넷이 연결되어 있는 네트워크 ACL의 인바운드 규칙을 기준으로 (번호가 가장 빠른 규칙부터 시작) 평가합니다. 예를 들어 HTTPS 포트(443)로 전송되는 IPv4 트래픽이 있다고 가정해 보겠습니다. 그런데 패킷이 규칙 100 또는 105와 일치하지 않고 서브넷으로의 트래픽을 허용하는 규칙 110과 일치합니다. 패킷이 포트 139(NetBIOS)로 전송되는 경우 번호가 매겨진 어떠한 규칙과도 일치하지 않으므로 IPv4 트래픽에 대한 * 규칙이 최종적으로 해당 패킷을 거부합니다.

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.(* */24*	허용	홈 네트워크의 퍼블릭 IPv4 주소 범위로부터의 인바운드 SSH 트래픽 허용(인터넷 게이트웨이를 통해)
140	사용자 지정 TCP	TCP	*32768-65* *35*	0.0.0.0/0	허용	인터넷으로부터의 인바운드 리턴 IPv4 트래픽 허용(서브넷에서 시작되는 요청에 대해).
145	사용자 지정 TCP	TCP	*32768-65* *35*	::/0	허용	인터넷으로부터의 인바운드 리턴 IPv6 트래픽 허용(서브넷에서 시작되는 요청에 대해).

규칙 #	유형	프로토콜	포트 범위	소스	허용/거부	설명
*	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 규칙의 예

다음 표에는 사용자 지정 네트워크 ACL의 아웃바운드 규칙 예가 나와 있습니다. IPv6에 대한 규칙은 VPC에 연결된 IPv6 CIDR 블록이 있는 경우에만 추가됩니다. IPv4 및 IPv6 트래픽은 각각 개별적으로 평가됩니다. 따라서 IPv4 트래픽에 대한 규칙은 IPv6 트래픽에 적용되지 않습니다. 해당하는 IPv4 규칙 옆에 IPv6 규칙을 추가하거나, 마지막 IPv4 규칙 뒤에 IPv6 규칙을 추가할 수 있습니다.

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 트래픽 허용
105	HTTP	TCP	80	::/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv6 HTTP 트래픽 허용
110	HTTPS	TCP	443	0.0.0.0/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv4 HTTPS 트래픽 허용
115	HTTPS	TCP	443	::/0	허용	서브넷에서 인터넷으로의 아웃바운드 IPv6 HTTPS 트래픽 허용

규칙 #	유형	프로토콜	포트 범위	대상 주소	허용/거부	설명
120	사용자 지정 TCP	TCP	*1024-655 5*	*192.0.2.(/24*	허용	홈 네트워크로에서 전송되는 SSH 트래픽에 대한 아웃바운드 응답을 허용합니다.
140	사용자 지정 TCP	TCP	*32768-65 35*	0.0.0.0/0	허용	인터넷에서 클라이언트에 대한 아웃바운드 IPv4 응답 허용(예: 웹 페이지 제공).
145	사용자 지정 TCP	TCP	*32768-65 35*	::/0	허용	인터넷에서 클라이언트에 대한 아웃바운드 IPv6 응답 허용(예: 웹 페이지 제공).
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv4 트래픽을 거부합니다.
*	모든 트래픽	모두	모두	::/0	DENY	이전 규칙에서 아직 처리하지 않은 모든 아웃바운드 IPv6 트래픽을 거부합니다.

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 시작된 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.
- AWS Lambda 함수는 포트 1024-65535를 사용합니다.

예를 들어, 인터넷을 통해 Windows 10 클라이언트로부터 VPC에 있는 웹 서버로 요청이 수신되는 경우, 네트워크 ACL에는 포트 49152-65535로 트래픽을 전달할 수 있도록 하는 아웃바운드 규칙이 있어야 합니다.

VPC의 인스턴스가 요청을 시작하는 클라이언트인 경우, 사용자의 네트워크 ACL에는 인스턴스의 운영 체제별로 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. 광범위한 임시 포트를 여는 허용 규칙보다 거부 규칙을 먼저 테이블에 배치해야 합니다.

사용자 지정 네트워크 ACL 및 기타 AWS 서비스

사용자 지정 네트워크 ACL을 생성하는 경우 다른 AWS 서비스를 사용하여 생성하는 리소스에 미칠 수 있는 영향에 주의해야 합니다.

Elastic Load Balancing을 사용하는 경우 사용자의 백엔드 인스턴스에 대한 서브넷에 소스가 `0.0.0.0/0` 또는 서브넷의 CIDR인 모든 트래픽에 대해 거부 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 다음을 참조하세요.

- Application Load Balancer를 위한 네트워크 ACL
- Network Load Balancer를 위한 네트워크 ACL
- Classic Load Balancer를 위한 네트워크 ACL

연결 문제 해결

Reachability Analyzer는 정적 구성 분석 도구입니다. Reachability Analyzer를 사용하여 VPC의 두 리소스 간 네트워크 연결성을 분석하고 디버깅할 수 있습니다. Reachability Analyzer에서는 연결할 수 있는 경우 이러한 리소스 간 가상 경로에 대한 홉별 세부 정보가 생성되고, 그렇지 않다면 차단 구성 요소가 식별됩니다. 예를 들면 누락되거나 잘못 구성된 네트워크 ACL 규칙이 식별될 수 있습니다.

자세한 내용은 Reachability Analyzer 사용 설명서를 참조하세요.

경로 MTU 검색 및 네트워크 ACL

경로 MTU 검색을 사용하여 두 디바이스 간의 경로 MTU를 확인할 수 있습니다. 경로 MTU는 발신 호스트와 수신 호스트 간의 경로에서 지원되는 최대 패킷 사이즈입니다.

IPv4의 경우 호스트가 수신 호스트 또는 경로상의 디바이스 MTU보다 큰 패킷을 전송하면, 수신 호스트 또는 디바이스가 해당 패킷을 폐기합니다. 그런 다음 다음과 같은 ICMP 메시지를 반환합니다. `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (Type 3, Code 4)

이렇게 하면 전송 호스트에 페이로드를 여러 개의 작은 패킷으로 분할한 다음 다시 전송하도록 지시합니다.

IPv6 프로토콜은 네트워크의 조각화를 지원하지 않습니다. 호스트가 수신 호스트 또는 경로상의 디바이스 MTU보다 큰 패킷을 전송하면, 수신 호스트 또는 디바이스가 해당 패킷을 폐기합니다. 그런 다음 다음과 같은 ICMP 메시지를 반환합니다. `ICMPv6 Packet Too Big (PTB)` (Type 2)

이렇게 하면 전송 호스트에 페이로드를 여러 개의 작은 패킷으로 분할한 다음 다시 전송하도록 지시합니다.

서브넷에 있는 호스트 간의 최대 MTU(전송 단위)가 다르거나 인스턴스가 인터넷을 통해 피어와 통신하는 경우 인바운드 및 아웃바운드 네트워크 ACL 규칙을 추가해야 합니다. 이렇게 하면 경로 MTU 검색이 올바르게 작동하고 패킷 손실을 방지할 수 있습니다. 유형에 대해 사용자 지정 ICMP 규칙을 선택하고 포트 범위(유형 3, 코드 4)에 대해 대상에 연결할 수 없음, 조각화 필요, DF 플래그 설정을 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 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스에 대한 네트워크 MTU(최대 전송 단위)를 참조하세요.

VPC에 대한 네트워크 ACL 만들기

다음 태스크는 네트워크 ACL을 생성하고, 네트워크 ACL에 규칙을 추가한 다음, 네트워크 ACL을 서브넷과 연결하는 방법을 보여줍니다.

작업

- 1단계: 네트워크 ACL 생성
- 2단계: 규칙 추가
- 3단계: 서브넷을 네트워크 ACL과 연결
- (선택 사항) Firewall Manager를 사용하여 네트워크 ACL 관리

1단계: 네트워크 ACL 생성

VPC에 대한 사용자 지정 네트워크 ACL을 생성할 수 있습니다. 사용자 지정 네트워크 ACL의 초기 규칙은 모든 인바운드 및 아웃바운드 트래픽을 차단합니다. 새 사용자 지정 네트워크 ACL은 기본적으로 서브넷과 연결되지 않으며, 따라서 서브넷과 명시적으로 연결해야 합니다.

콘솔을 사용하여 네트워크 ACL을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Network ACLs를 선택합니다.
3. 네트워크 ACL 생성을 선택합니다.
4. (선택 사항) 이름에 네트워크 ACL의 이름을 입력합니다.
5. VPC에서 VPC를 선택합니다.
6. (선택 사항) 태그에서 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. 네트워크 ACL 생성을 선택합니다.

명령줄을 사용하여 네트워크 ACL을 생성하려면

- create-network-acl(AWS CLI)
- New-EC2NetworkAcl(AWS Tools for Windows PowerShell)

2단계: 규칙 추가

인바운드 또는 아웃바운드 트래픽을 허용하거나 거부하는 규칙을 추가할 수 있습니다.

규칙은 가장 낮은 번호의 규칙부터 시작해서 순서대로 처리됩니다. 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.

Amazon EC2 API 또는 명령줄 도구를 사용하는 경우에는 규칙을 수정할 수 없습니다. 규칙을 추가 및 삭제할 수만 있습니다. Amazon VPC 콘솔을 사용하는 경우에는 기존 규칙의 항목을 수정할 수 있습니다. 콘솔은 기존 규칙을 제거하고 새 규칙을 추가합니다. ACL에서 규칙의 순서를 변경할 필요가 있는 경우에는 새 규칙 번호와 함께 새 규칙을 추가한 후에 원래 규칙은 삭제해야 합니다.

콘솔을 사용하여 네트워크 ACL에 규칙을 추가하려면

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

2. 탐색 창에서 Network ACLs를 선택합니다.
3. 네트워크 ACL을 선택합니다.
4. 인바운드 규칙을 추가하려면 다음을 수행합니다.

   a. 인바운드 규칙 탭을 선택합니다.
   b. 인바운드 규칙 편집, 새 규칙 추가를 차례로 선택합니다.
   c. 아직 사용되지 않은 규칙 번호, 유형, 프로토콜, 포트 범위, 소스, 그리고 트래픽을 허용할지 아니면 거부할지 여부를 입력합니다. 일부 유형의 경우 프로토콜과 포트를 입력합니다. 포트 범위를 입력하라는 메시지가 표시되면 포트 번호 또는 포트 범위(예: 49152~65535)를 입력합니다.

      목록에 없는 프로토콜을 사용하려면 유형에서 사용자 지정 프로토콜을 선택한 다음 프로토콜을 선택합니다. 자세한 내용은 IANA 프로토콜 번호를 참조하세요.
   d. 변경 사항 저장을 선택합니다.
5. 아웃바운드 규칙을 추가하려면 다음을 수행합니다.

   a. Outbound rules(아웃바운드 규칙) 탭을 선택합니다.
   b. 아웃바운드 규칙 편집, 새 규칙 추가를 차례로 선택합니다.
   c. 아직 사용되지 않은 규칙 번호, 유형, 프로토콜, 포트 범위, 소스, 그리고 트래픽을 허용할지 아니면 거부할지 여부를 입력합니다. 일부 유형의 경우 프로토콜과 포트를 입력합니다. 포트 범위를 입력하라는 메시지가 표시되면 포트 번호 또는 포트 범위(예: 49152~65535)를 입력합니다.

      목록에 없는 프로토콜을 사용하려면 유형에서 사용자 지정 프로토콜을 선택한 다음 프로토콜을 선택합니다. 자세한 내용은 IANA 프로토콜 번호를 참조하세요.
   d. 변경 사항 저장을 선택합니다.

명령줄을 사용하여 네트워크 ACL에 새 규칙을 추가하려면

- create-network-acl-entry(AWS CLI)
- New-EC2NetworkAclEntry(AWS Tools for Windows PowerShell)

명령줄을 사용하여 네트워크 ACL의 규칙을 바꾸려면

- replace-network-acl-entry(AWS CLI)
- Set-EC2NetworkAclEntry(AWS Tools for Windows PowerShell)

명령줄을 사용하여 네트워크 ACL에서 규칙을 삭제하려면

- delete-network-acl-entry(AWS CLI)
- Remove-EC2NetworkAclEntry(AWS Tools for Windows PowerShell)

3단계: 서브넷을 네트워크 ACL과 연결

특정 서브넷에 네트워크 ACL의 규칙을 적용하려면 서브넷을 네트워크 ACL과 연결해야 합니다. 네트워크 ACL을 여러 서브넷과 연결할 수 있습니다. 그러나 서브넷은 하나의 네트워크 ACL에만 연결될 수 있습니다. 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 ACLs를 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.

(선택 사항) Firewall Manager를 사용하여 네트워크 ACL 관리

AWS Firewall Manager는 여러 계정과 리소스 간에 네트워크 ACL 관리 및 유지 관리 작업을 간소화합니다. Firewall Manager를 사용하여 조직의 계정 및 서브넷을 모니터링하고 정의한 네트워크 ACL 구성을 자동으로 적용할 수 있습니다. Firewall Manager는 조직 전체를 보호해야 하거나 중앙 관리자 계정으로 자동 보호할 새 서브넷을 자주 추가하는 경우에 특히 유용합니다.

Firewall Manager 네트워크 ACL 정책을 사용하면 단일 관리자 계정을 사용하여 조직 전체에서 사용하는 네트워크 ACL에 정의하려는 최소 규칙 세트를 구성, 모니터링 및 관리할 수 있습니다. 조직의 어떤 계정과 서브넷이 Firewall Manager 정책 범위 내에 속하는지 지정합니다. Firewall Manager는 범위 내 서브넷에 대한 네트워크 ACL의 규정 준수 상태를 보고하며, 규정에 부합하지 않는 네트워크 ACL을 자동으로 수정하도록 Firewall Manager를 구성할 수 있습니다.

자세한 내용은 AWS Firewall Manager 개발자 가이드에서 다음 리소스를 참조하세요.

- AWS Firewall Manager 사전 조건
- AWS Firewall Manager 네트워크 ACL 정책 설정
- Firewall Manager로 네트워크 ACL 정책 사용

VPC에 대한 네트워크 ACL 연결 관리

각 서브넷은 하나의 네트워크 ACL에 연결됩니다. 서브넷을 처음 생성하면 생성된 서브넷이 VPC의 기본 네트워크 ACL과 연결됩니다. 사용자 지정 네트워크 ACL을 생성한 후 하나 이상의 서브넷에 연결하여 이전 네트워크 ACL 연결을 대체할 수 있습니다.

작업

- 네트워크 ACL 연결 설명
- 네트워크 ACL과 연결된 서브넷 변경
- 서브넷과 연결된 네트워크 ACL 변경

네트워크 ACL 연결 설명

서브넷과 연결된 네트워크 ACL을 설명할 수 있으며, 네트워크 ACL과 연결된 서브넷을 설명할 수도 있습니다.

콘솔을 사용하여 서브넷과 연결된 네트워크 ACL을 설명하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. 네트워크 ACL 탭을 선택합니다.

AWS CLI를 사용하여 서브넷과 연결된 네트워크 ACL을 설명하려면

지정된 서브넷과 연결된 네트워크 ACL을 나열하려면 다음 describe-network-acls 명령을 사용합니다.

```
aws ec2 describe-network-acls --filters Name=association.subnet-
id,Values=subnet-0d2d1b81e0bc9c6d4 --query NetworkAcls[*].NetworkAclId
```

다음은 예제 출력입니다.

```
[
    "acl-03701d1f82d8c3fd6"
]
```

콘솔을 사용하여 네트워크 ACL과 연결된 서브넷을 설명하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Network ACLs를 선택합니다.
3. 네트워크 ACL을 선택합니다.
4. 서브넷 연결 탭을 선택합니다.

AWS CLI를 사용하여 네트워크 ACL과 연결된 서브넷을 설명하려면

지정된 네트워크 ACL과 연결된 서브넷을 나열하려면 다음 describe-network-acls 명령을 사용합니다.

```
aws ec2 describe-network-acls --network-acl-ids acl-060415a18fcc9afde --query
 NetworkAcls[*].Associations[].SubnetId
```

다음은 예제 출력입니다.

```
[
    "subnet-0d2d1b81e0bc9c6d4",
    "subnet-0e990c67809773b19",
    "subnet-0eb17d85f5dfd33b1",
    "subnet-0e01d500780bb7468"
]
```

네트워크 ACL과 연결된 서브넷 변경

서브넷에서 사용자 지정 네트워크 ACL을 연결 해제할 수 있습니다. 서브넷과 사용자 지정 네트워크 ACL의 연결이 끊어지면 VPC의 기본 네트워크 ACL과 자동으로 연결됩니다. 이 변경 사항은 잠시 후 적용됩니다.

네트워크 ACL과 연결된 서브넷을 변경하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Network ACLs를 선택합니다.
3. 네트워크 ACL을 선택합니다.
4. 작업, 서브넷 연결 편집을 차례로 선택합니다.
5. 선택한 서브넷에서 해당 서브넷을 제거합니다.
6. 변경 사항 저장을 선택합니다.

서브넷과 연결된 네트워크 ACL 변경

서브넷과 연결되어 있는 네트워크 ACL을 변경할 수 있습니다. 예를 들어 서브넷을 생성하면 생성된 서브넷이 처음에는 VPC의 기본 네트워크 ACL과 연결됩니다. 사용자 지정 네트워크 ACL을 생성하는 경우 네트워크 ACL을 하나 이상의 서브넷과 연결하여 네트워크 ACL 규칙을 적용합니다.

서브넷의 네트워크 ACL을 변경하면 잠시 후에 변경 사항이 적용됩니다.

서브넷과 연결된 네트워크 ACL을 변경하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Subnets를 선택합니다.
3. 서브넷을 선택합니다.
4. 작업, 네트워크 ACL 연결 편집을 차례로 선택합니다.
5. 네트워크 ACL ID에서 서브넷과 연결할 네트워크 ACL을 선택하고, 선택한 네트워크 ACL의 인바운드 및 아웃바운드 규칙을 검토합니다.
6. 저장을 선택합니다.

명령줄을 사용하여 네트워크 ACL 연결을 바꾸려면

- replace-network-acl-association(AWS CLI)
- Set-EC2NetworkAclAssociation(AWS Tools for Windows PowerShell)

VPC의 네트워크 ACL 삭제

사용을 마친 네트워크 ACL을 삭제할 수 있습니다. 서브넷이 연결되어 있는 네트워크 ACL은 삭제할 수 없습니다. 기본 네트워크 ACL은 삭제할 수 없습니다.

콘솔을 사용하여 네트워크 ACL에서 서브넷 연결을 제거하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Network ACLs를 선택합니다. 다음과 연결 열에 각 네트워크 ACL에 연결된 서브넷의 수가 표시됩니다. 이 열은 연결된 서브넷이 없는 경우 -입니다.
3. 네트워크 ACL을 선택합니다.
4. 작업, 서브넷 연결 편집을 차례로 선택합니다.
5. 서브넷 연결을 제거합니다.

6. 변경 사항 저장을 선택합니다.

명령줄을 사용하여 연결을 포함한 네트워크 ACL을 설명하려면

- describe-network-acls(AWS CLI)
- Get-EC2NetworkAcl(AWS Tools for Windows PowerShell)

명령줄을 사용하여 네트워크 ACL 연결을 바꾸려면

- replace-network-acl-association(AWS CLI)
- Set-EC2NetworkAclAssociation(AWS Tools for Windows PowerShell)

콘솔을 사용하여 네트워크 ACL을 삭제하려면

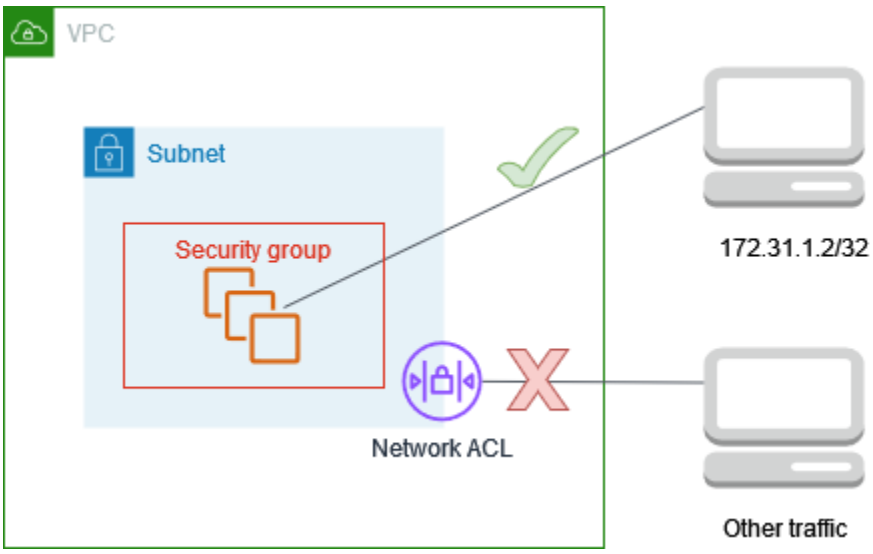
1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 Network ACLs를 선택합니다.
3. 네트워크 ACL을 선택합니다.
4. 작업, 네트워크 ACL 삭제를 차례로 선택합니다.
5. 확인 메시지가 나타나면 **delete**을 입력한 다음 삭제를 선택합니다.

명령줄을 사용하여 네트워크 ACL을 삭제하려면

- delete-network-acl(AWS CLI)
- Remove-EC2NetworkAcl(AWS Tools for Windows PowerShell)

예: 서브넷의 인스턴스에 대한 액세스 제어

이 예시에서는 서브넷 내 인스턴스가 서로 통신할 수 있으며, 신뢰할 수 있는 원격 컴퓨터에서 관리 작업을 수행하기 위해 해당 인스턴스에 액세스할 수 있습니다. 원격 컴퓨터는 다이어그램에 나와 있는 것처럼 로컬 네트워크의 컴퓨터이거나 다른 서브넷 또는 VPC의 인스턴스일 수 있습니다. 서브넷에 대한 네트워크 ACL 규칙과 인스턴스에 대한 보안 그룹 규칙은 원격 컴퓨터의 IP 주소로부터의 액세스를 허용합니다. 인터넷 또는 다른 네트워크로부터의 다른 모든 트래픽은 거부됩니다.



네트워크 ACL을 사용하면, 네트워크 ACL을 방어의 백업 계층으로 사용하면서 인스턴스에 대한 보안 그룹 또는 보안 그룹 규칙을 변경할 수 있는 유연성을 얻을 수 있습니다.

예를 들어 어디서나 인바운드 SSH 액세스를 허용하도록 보안 그룹을 실수로 업데이트하더라도, 네트워크 ACL은 원격 컴퓨터의 IP 주소 범위로만 액세스를 제한합니다. 네트워크 ACL은 그 외 모든 IP 주소로부터의 인바운드 SSH 트래픽을 거부합니다.

네트워크 ACL 규칙

다음은 서브넷과 연결된 네트워크 ACL의 인바운드 규칙 예입니다. 이들 규칙은 서브넷의 모든 인스턴스에 적용됩니다.

규칙 #	유형	프로토콜	포트 범위	소스	허용/거부	설명
100	SSH	TCP	22	*172.31.1.2/32*	허용	원격 컴퓨터로부터의 인바운드 트래픽을 허용합니다.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	다른 모든 인바운드 트래픽을

규칙 #	유형	프로토콜	포트 범위	소스	허용/거부	설명
						거부합니다.

다음은 서브넷과 연결된 네트워크 ACL의 아웃바운드 규칙 예입니다. 네트워크 ACL은 상태가 저장되지 않습니다. 따라서 인바운드 트래픽에 대한 응답을 허용하는 규칙을 포함해야 합니다.

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.
*	모든 트래픽	모두	모두	0.0.0.0/0	DENY	다른 모든 아웃바운드 트래픽을 거부합니다.

보안 그룹 규칙

다음은 인스턴스에 연결된 보안 그룹에 대한 인바운드 규칙 예입니다. 이들 규칙은 보안 그룹과 연결된 모든 인스턴스에 적용됩니다. 인스턴스와 연결된 키 페어의 프라이빗 키가 있는 사용자는 원격 컴퓨터에서 SSH를 사용하여 이 인스턴스에 연결할 수 있습니다.

프로토콜 유형	프로토콜	포트 범위	소스	설명
모든 트래픽	모두	모두	*sg-1234567890abcdef0*	이 보안 그룹과 연결된 인스턴스 간의 통신을 허용합니다.
SSH	TCP	22	*172.31.1.2/32*	원격 컴퓨터로부터의 인바운드

프로토콜 유형	프로토콜	포트 범위	소스	설명
				SSH 액세스를 허용합니다.

다음은 인스턴스에 연결된 보안 그룹에 대한 아웃바운드 규칙 예입니다. 보안 그룹은 상태가 저장됩니다. 따라서 인바운드 트래픽에 대한 응답을 허용하는 규칙은 필요하지 않습니다.

프로토콜 유형	프로토콜	포트 범위	대상 주소	설명
모든 트래픽	모두	모두	*sg-1234567890abcdef0*	이 보안 그룹과 연결된 인스턴스 간의 통신을 허용합니다.

네트워크 ACL과 보안 그룹 간의 차이점

다음 표는 보안 그룹과 네트워크 ACL의 근본적인 차이를 요약한 것입니다.

기능	네트워크 ACL	보안 그룹
작업 수준	서브넷 수준	인스턴스 수준
범위	연결된 서브넷의 모든 인스턴스에 적용됨	보안 그룹과 연결된 모든 인스턴스에 적용됨
규칙 타입	규칙 허용 및 거부	규칙만 허용
규칙 평가	트래픽과 일치하는 항목이 발견될 때까지 규칙을 오름차순으로 평가합니다.	트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가
트래픽 반환	허용 명시 필요(상태 비저장)	자동 허용(상태 저장)

Amazon Virtual Private Cloud에서의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 짧은 대기 시간, 높은 처리량, 높은 중복성을 갖춘 네트워크를 사용하여 연결된 물리적으로 분리되고 격리된 다수의 가용 영역을 제공합니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 리전은 기본 구성 요소로, 각각 물리적으로 분리되고 격리된 여러 가용 영역이 있는 별개의 지리적 위치를 나타냅니다. 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워킹 패브릭을 통해 연결되므로 가용 영역 간에 원활한 통신과 데이터 전송이 가능합니다.

가용 영역 아키텍처는 기존의 단일 또는 다중 데이터 센터 설정보다 훨씬 더 강력하고 내결함성을 갖도록 설계되었기 때문에 주요 차별화 요소입니다. 리전 내의 여러 가용 영역에 리소스를 분산하여 서비스 중단 없이 애플리케이션과 데이터베이스가 영역 간에 자동으로 장애 조치되도록 설계할 수 있습니다. 이러한 수준의 중복성과 고가용성은 미션 크리티컬 워크로드의 핵심 요구 사항이며, 이를 통해 조직은 복원력이 뛰어난 클라우드 네이티브 솔루션을 구축할 수 있습니다.

또한 AWS 인프라의 규모와 글로벌 지원 범위를 통해 고객은 애플리케이션을 최종 사용자에게 더 가깝게 배포하여 지연 시간을 줄이고 전반적인 사용자 경험을 개선할 수 있습니다. 전 세계 여러 리전을 사용할 수 있기 때문에 고객은 특정 규제 및 비즈니스 요구 사항에 따라 필요한 지리적 경계 내에서 데이터를 저장하고 처리할 수 있으므로 효과적인 데이터 주권 실현과 규정 준수가 가능합니다.

조직은 AWS 글로벌 인프라를 활용하여 변화하는 요구 사항과 발전하는 비즈니스 니즈에 유연하게 적응할 수 있는 고가용성, 내결함성 및 확장성을 갖춘 클라우드 환경을 설계할 수 있습니다. 이러한 견고한 기반은 최신 클라우드 기반 애플리케이션 및 서비스를 성공적으로 구현하기 위한 핵심 지원 요소입니다.

AWS 리전 및 가용 영역에 대한 자세한 정보는 AWS 글로벌 인프라를 참조하세요.

워크로드의 복원력 요구 사항을 충족하도록 VPC를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 복원력 패턴 및 장단점 이해(AWS 아키텍처 블로그)
- 네트워크 토폴로지 계획(AWS Well-Architected 프레임워크)
- Amazon Virtual Private Cloud 연결 옵션(AWS 백서)

Amazon Virtual Private Cloud에 대한 규정 준수 확인

새로운 AWS 가입 환경은 규제를 따르는 워크로드용으로 설계되지 않았습니다. 새로운 AWS 가입 환경을 사용하고 있지만 규제를 따르는 워크로드에 AWS를 사용하려는 경우 AWS(고급)에 가입하거나 AWS 환경에 대해 고급 기능을 활성화할 수 있습니다.

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 결정됩니다. AWS 서비스 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서 참조하세요.

VPC 및 서브넷에 대한 퍼블릭 액세스 차단

VPC 퍼블릭 액세스 차단(BPA)은 전체 AWS 계정의 VPC 리소스에 대한 퍼블릭 인터넷 액세스를 절대적으로 차단하는 중앙 집중식 보안 기능입니다. 이 기능은 보안 요구 사항의 준수를 보장하는 동시에, 특정 예외 처리 및 감사 기능을 위한 유연성을 제공합니다.

VPC BPA 기능에는 다음과 같은 모드가 있습니다.

- 양방향: 이 리전의 인터넷 게이트웨이 및 송신 전용 인터넷 게이트웨이(제외된 VPC 및 서브넷 제외)를 오가는 모든 트래픽이 차단됩니다.
- 수신 전용: 이 리전의 VPC에 대한 모든 인터넷 트래픽(제외된 VPC 또는 서브넷 제외)이 차단됩니다. NAT 게이트웨이 및 송신 전용 인터넷 게이트웨이를 오가는 트래픽만 허용되는데, 이러한 게이트웨이는 아웃바운드 연결만 설정되도록 허용하기 때문입니다.

차단하지 않으려는 트래픽에 대해서는 이 기능의 "제외 항목"을 생성할 수도 있습니다. 제외 항목은 계정의 VPC BPA 모드에서는 제외하고 양방향 또는 송신 전용 액세스를 허용하는 단일 VPC 또는 서브넷에 적용할 수 있는 모드입니다.

제외 항목에는 다음 모드 중 하나를 사용할 수 있습니다.

- 양방향: 제외된 VPC 및 서브넷을 오가는 모든 인터넷 트래픽이 허용됩니다.

- 송신 전용: 제외된 VPC 및 서브넷의 아웃바운드 인터넷 트래픽만 허용됩니다. 제외된 VPC 및 서브넷으로의 인바운드 인터넷 트래픽은 차단됩니다. 이 모드는 VPC BPA가 양방향으로 설정된 경우에만 적용됩니다.

내용

- VPC BPA 기본 사항
- VPC BPA의 영향 평가 및 VPC BPA 모니터링
- 고급 예제

VPC BPA 기본 사항

이 섹션에서는 VPC BPA를 지원하는 서비스와 이를 사용하는 방법을 포함하여 VPC BPA에 대한 중요한 세부 정보를 다룹니다.

내용

- 리전별 가용성
- AWS 서비스 영향 및 지원
- VPC BPA 제한 사항
- IAM 정책을 사용하여 VPC BPA에 대한 액세스 제어
- 계정에 대해 VPC BPA 양방향 모드 활성화
- VPC BPA 모드를 수신 전용으로 변경
- 제외 항목 생성 및 삭제
- 조직 수준에서 VPC BPA 활성화

리전별 가용성

VPC BPA는 GovCloud 및 중국 리전을 포함한 모든 상용 AWS 리전에서 사용할 수 있습니다.

이 설명서에서는 VPC BPA와 함께 Network Access Analyzer 및 Reachability Analyzer를 사용하는 방법에 대한 정보도 확인할 수 있습니다. Network Access Analyzer 및 Reachability Analyzer는 일부 상용 리전에서 사용할 수 없습니다. Network Access Analyzer 및 Reachability Analyzer의 리전별 가용성에 대한 자세한 내용은 Network Access Analyzer 설명서의 제한 사항 및 Reachability Analyzer 설명서의 고려 사항을 참조하세요.

AWS 서비스 영향 및 지원

다음 리소스 및 서비스는 VPC BPA를 지원하며 이러한 서비스 및 리소스로의 트래픽은 VPC BPA의 영향을 받습니다.

- 인터넷 게이트웨이: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다.
- 송신 전용 인터넷 게이트웨이: 모든 아웃바운드 트래픽이 차단됩니다. 송신 전용 인터넷 게이트웨이는 인바운드 트래픽을 허용하지 않습니다.
- Gateway Load Balancer(GWLB): GWLB 엔드포인트가 포함된 서브넷이 제외된 경우 외에는 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다.
- NAT 게이트웨이: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다. NAT 게이트웨이에는 인터넷 연결을 위한 인터넷 게이트웨이가 필요합니다.
- 인터넷 연결 Network Load Balancer: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다. 인터넷 연결 Network Load Balancer에는 인터넷 연결을 위한 인터넷 게이트웨이가 필요합니다.
- 인터넷 연결 Application Load Balancer: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다. 인터넷 연결 Application Load Balancer에는 인터넷 연결을 위한 인터넷 게이트웨이가 필요합니다.
- Amazon CloudFront VPC 오리진: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다.
- Direct Connect: 퍼블릭 가상 인터페이스(퍼블릭 IPv4 또는 글로벌 유니캐스트 IPv6 주소)를 사용하는 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다. 이 트래픽은 연결에 인터넷 게이트웨이(또는 송신 전용 인터넷 게이트웨이)를 사용합니다.
- AWS Global Accelerator: 인터넷에서 대상에 액세스할 수 있는지 여부에 관계없이 VPC에 대한 인바운드 트래픽이 차단됩니다.
- AWS Network Firewall: 방화벽 엔드포인트가 포함된 서브넷을 제외하지 않는 한 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다.
- AWS Wavelength 통신 사업자 게이트웨이: 모든 인바운드 및 아웃바운드 트래픽이 차단됩니다.

다음 서비스 및 리소스에 대한 트래픽과 같은 프라이빗 연결과 관련된 트래픽은 VPC BPA에 의해 차단되거나 영향을 받지 않습니다.

- AWS Client VPN
- AWS CloudWAN
- AWS Outposts 로컬 게이트웨이
- AWS Site-to-Site VPN

- Transit Gateway
- AWS Verified Access

> **Important**
>
> - VPC 인그레스 라우팅(VPC Ingress Routing)을 사용하여 서브넷 내 EC2 인스턴스에서 실행되는 어플라이언스(예: 서드파티 보안 또는 모니터링 도구)를 통해 인바운드 및 아웃바운드 트래픽을 라우팅하는 경우, VPC 퍼블릭 액세스 차단 대상에서 워크로드 서브넷을 제외해야 합니다. 워크로드 서브넷은 어플라이언스를 통해 트래픽이 라우팅되는 인스턴스가 포함된 서브넷입니다. 어플라이언스가 포함된 서브넷은 제외하지 않아도 됩니다.
> - VPC의 리소스로부터 VPC에서 실행되는 다른 서비스(예: Route 53 Resolver 또는)로 비공개로 전송되는 트래픽은 VPC의 인터넷 게이트웨이를 통과하지 않으므로 BPA가 켜져 있더라도 허용됩니다. 이러한 서비스는 사용자를 대신하여 VPC 외부의 리소스에 요청을 보낼 수 있으며(예: DNS 쿼리 처리), 다른 보안 제어로 이를 완화하지 않으면 VPC 내 리소스 활동에 대한 정보가 노출될 수 있습니다.
> - 인터넷 경계 로드 밸런서가 있고 서브넷 중 하나에 대해서만 VPC BPA 제외를 생성하는 경우에도 로드 밸런서는 제외된 서브넷에서 퍼블릭 트래픽을 수신하고 제외되지 않은 서브넷의 대상에 비공개적으로 라우팅할 수 있습니다. VPC BPA가 대상에 대한 퍼블릭 액세스를 완전히 차단하도록 하려면 어떤 로드 밸런서 서브넷도 제외되지 않도록 해야 합니다.

VPC BPA 제한 사항

VPC BPA 수신 전용 모드는 NAT 게이트웨이 및 송신 전용 인터넷 게이트웨이가 허용되지 않는 로컬 영역(LZ)에서는 지원되지 않습니다.

IAM 정책을 사용하여 VPC BPA에 대한 액세스 제어

VPC BPA 기능에 대한 액세스를 허용/거부하는 IAM 정책의 예는 VPC 및 서브넷에 대한 퍼블릭 액세스 차단 섹션을 참조하세요.

계정에 대해 VPC BPA 양방향 모드 활성화

VPC BPA 양방향 모드는 이 리전의 인터넷 게이트웨이 및 송신 전용 인터넷 게이트웨이(제외된 VPC 및 서브넷 제외)를 오가는 모든 트래픽을 차단합니다. 제외 항목에 관한 자세한 내용은 제외 항목 생성 및 삭제 섹션을 참조하세요.

> **Important**
>
> 프로덕션 계정에서 VPC BPA를 활성화하기 전에 인터넷 액세스가 필요한 워크로드를 면밀히 검토할 것을 강력히 권장합니다.

> **Note**
>
> - 계정의 VPC 및 서브넷에서 VPC BPA를 활성화하려면 VPC 및 서브넷을 소유해야 합니다.
> - 현재 다른 계정과 VPC 서브넷을 공유하는 경우 서브넷 소유자가 적용한 VPC BPA 모드가 참가자 트래픽에도 적용되지만 참가자는 공유 서브넷에 영향을 미치는 VPC BPA 설정을 제어할 수 없습니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 설정 편집을 선택합니다.
4. 퍼블릭 액세스 차단 켜기 및 양방향을 선택한 다음 변경 사항 저장을 선택합니다.
5. 상태가 켜기로 변경될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

이제 VPC BPA 양방향 모드가 켜져 있습니다.

AWS CLI

1. VPC BPA 켜기:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-bidirectional
```

VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

2. VPC BPA 상태 보기:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

VPC BPA 모드를 수신 전용으로 변경

VPC BPA 수신 전용 모드는 이 리전의 VPC에 대한 모든 인터넷 트래픽(제외된 VPC 또는 서브넷 제외)을 차단합니다. NAT 게이트웨이 및 송신 전용 인터넷 게이트웨이를 오가는 트래픽만 허용되는데, 이러한 게이트웨이는 아웃바운드 연결만 설정되도록 허용하기 때문입니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 설정 편집을 선택합니다.
4. 방향을 수신 전용으로 변경합니다.
5. 변경 사항을 저장하고 상태가 업데이트될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

1. VPC BPA 차단 방향 수정:

   ```
   aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-ingress
   ```

   VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.
2. VPC BPA 상태 보기:

   ```
   aws ec2 --region us-east-2 describe-vpc-block-public-access-options
   ```

제외 항목 생성 및 삭제

VPC BPA 제외 모드는 계정의 VPC BPA 모드 적용을 면제하여 양방향 또는 아웃바운드 전용(Egress-only) 액세스를 허용하는 모드로, 단일 VPC 또는 서브넷에 적용할 수 있습니다. 계정에서 VPC BPA가 활성화되지 않은 경우에도 VPC 및 서브넷에 대한 VPC BPA 제외 항목을 생성하여 VPC BPA가 켜져 있을 때 제외 항목에 대한 트래픽 중단이 없도록 할 수 있습니다. VPC에 대한 제외는 VPC의 모든 서브넷에 자동으로 적용됩니다.

최대 50개의 제외 항목을 생성할 수 있습니다. 한도 증가 요청에 대한 자세한 내용은 Amazon VPC 할당량의 계정별 VPC BPA 제외 항목을 참조하세요.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 차단 탭의 제외 항목 아래에서 다음 중 하나를 수행합니다.
   - 제외를 삭제하려면 제외를 선택한 다음 작업 > 제외 삭제를 선택합니다.
   - 제외를 생성하려면 제외 생성을 선택하고 다음 단계를 계속합니다.
4. 차단 방향을 선택합니다.
   - 양방향: 제외된 VPC 및 서브넷을 오가는 모든 인터넷 트래픽을 허용합니다.
   - 송신 전용: 제외된 VPC 및 서브넷의 아웃바운드 인터넷 트래픽을 허용합니다. 제외된 VPC 및 서브넷으로의 인바운드 인터넷 트래픽을 차단합니다. 이 설정은 VPC BPA가 양방향으로 설정된 경우에 적용됩니다.
5. VPC 또는 서브넷을 선택합니다.
6. 제외 항목 생성을 선택합니다.
7. 제외 상태가 활성으로 변경될 때까지 기다립니다. 변경 사항을 보려면 제외 항목 테이블을 새로 고쳐야 할 수 있습니다.

제외 항목이 생성되었습니다.

AWS CLI

1. 제외 항목 허용 방향 수정:

```
aws ec2 --region us-east-2 create-vpc-block-public-access-exclusion --subnet-
id subnet-id --internet-gateway-exclusion-mode allow-bidirectional
```

2. 제외 상태가 업데이트되는 데 시간이 걸릴 수 있습니다. 제외 항목의 상태를 보려면:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusions --
exclusion-ids exclusion-id
```

조직 수준에서 VPC BPA 활성화

AWS Organizations를 사용하여 조직의 계정을 관리하는 경우 AWS Organizations 선언적 정책을 사용하여 조직의 계정에 VPC BPA를 적용할 수 있습니다. VPC BPA 선언적 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 지원되는 선언적 정책을 참조하세요.

> **Note**
>
> - VPC BPA 선언적 정책을 사용하여 제외 허용 여부를 구성할 수 있지만 정책으로 제외를 생성할 수는 없습니다. 제외를 생성하려면 VPC를 소유한 계정에서 해당 제외를 생성해야 합니다. VPC BPA 제외에 대한 자세한 내용은 제외 항목 생성 및 삭제 섹션을 참조하세요.
> - VPC BPA 선언적 정책이 활성화된 경우 퍼블릭 액세스 차단 설정에서 선언적 정책으로 관리됨이 표시되며 계정 수준에서는 VPC BPA 설정을 수정할 수 없습니다.

VPC BPA의 영향 평가 및 VPC BPA 모니터링

이 섹션에는 VPC BPA를 켜기 전에 VPC BPA의 영향을 평가하고 VPC BPA를 켠 후에 트래픽이 차단되는지 모니터링하는 방법에 대한 정보가 포함되어 있습니다.

내용

- Network Access Analyzer를 사용하여 VPC BPA의 영향 평가
- 흐름 로그를 사용하여 VPC BPA 영향 모니터링
- CloudTrail을 사용하여 제외 항목 삭제 추적
- Reachability Analyzer를 사용하여 연결이 차단되었는지 확인

Network Access Analyzer를 사용하여 VPC BPA의 영향 평가

이 섹션에서는 VPC BPA를 활성화하여 액세스를 차단하기 전에 Network Access Analyzer를 사용하여 인터넷 게이트웨이를 사용하는 계정의 리소스를 확인하는 방법을 살펴봅니다. 이 분석을 사용하여 계정에서 VPC BPA를 켜고 트래픽을 차단할 때의 영향을 파악할 수 있습니다.

> **Note**
>
> - Network Access Analyzer는 IPv6을 지원하지 않으므로 송신 전용 인터넷 게이트웨이 아웃바운드 IPv6 트래픽에 대한 VPC BPA의 잠재적 영향 확인에 사용할 수 없습니다.

- Network Access Analyzer를 사용하여 수행하는 분석에는 요금이 부과됩니다. 자세한 내용을 알아보려면 Network Access Analyzer Guide(Network Access Analyzer 설명서)의 가격을 참조하세요.
- Network Access Analyzer의 리전별 가용성에 대한 자세한 내용은 Network Access Analyzer 설명서의 제한 사항을 참조하세요.

AWS Management Console

1. https://console.aws.amazon.com/networkinsights/에서 AWS Network Insights 콘솔을 엽니다.
2. Network Access Analyzer를 선택합니다.
3. 네트워크 액세스 범위 생성을 선택합니다.
4. VPC 퍼블릭 액세스 차단의 영향 평가를 선택하고 다음을 선택합니다.
5. 템플릿은 이미 계정의 인터넷 게이트웨이를 오가는 트래픽을 분석하도록 구성되어 있습니다. 소스 및 대상에서 이를 확인할 수 있습니다.
6. 다음을 선택합니다.
7. 네트워크 액세스 범위 생성을 선택합니다.
8. 방금 생성한 범위를 선택하고 분석을 선택합니다.
9. 분석이 완료될 때까지 기다립니다.
10. 분석 결과를 확인합니다. 분석 결과 아래의 각 행에는 패킷이 네트워크에서 계정의 인터넷 게이트웨이를 들어오고 나갈 때 거칠 수 있는 네트워크 경로가 표시됩니다. 이 경우 VPC BPA를 켜고 이러한 분석 결과에 나타나는 VPC 및/또는 서브넷 중 VPC BPA 제외 항목으로 구성된 것이 없는 경우 해당 VPC 및 서브넷으로의 트래픽이 제한됩니다.
11. 각 분석 결과를 확인하여 VPC BPA가 VPC의 리소스에 미치는 영향을 파악합니다.

영향 분석이 완료되었습니다.

AWS CLI

1. 네트워크 액세스 범위 생성:

```
aws ec2 create-network-insights-access-scope --region us-east-2 --match-paths
 "Source={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 "Destination={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
```

2. 범위 분석 시작:

```
aws ec2 start-network-insights-access-scope-analysis  --region us-east-2 --
network-insights-access-scope-id nis-id
```

3. 분석 결과 가져오기:

```
aws ec2 get-network-insights-access-scope-analysis-findings  --region us-east-2
 --network-insights-access-scope-analysis-id nisa-0aa383a1938f94cd1 --max-items
 1
```

결과는 계정의 모든 VPC에서 인터넷 게이트웨이를 들어오고 나가는 트래픽을 보여줍니다. 결과는 "분석 결과"로 구성됩니다. "FindingId": "AnalysisFinding-1"은 이것이 분석의 첫 번째 결과임을 나타냅니다. 여러 분석 결과가 있는 경우 각 분석 결과는 VPC BPA를 켜면 영향을 받는 트래픽 흐름을 나타냅니다. 첫 번째 분석 결과는 인터넷 게이트웨이("SequenceNumber": 1)에서 시작된 트래픽이 NACL("SequenceNumber": 2)을 거쳐 보안 그룹("SequenceNumber": 3)으로 전달되고 인스턴스("SequenceNumber": 4)에서 종료된 것을 보여줍니다.

4. 분석 결과를 확인하여 VPC BPA가 VPC의 리소스에 미치는 영향을 파악합니다.

영향 분석이 완료되었습니다.

흐름 로그를 사용하여 VPC BPA 영향 모니터링

VPC 흐름 로그는 VPC의 탄력적 네트워크 인터페이스에서 전송되고 수신되는 IP 트래픽에 대한 정보를 수집할 수 있는 기능입니다. 이 기능을 사용하여 VPC BPA가 인스턴스 네트워크 인터페이스에 도달하지 못하도록 차단한 트래픽을 모니터링할 수 있습니다.

흐름 로그 작업의 단계에 따라 VPC에 대한 흐름 로그를 생성합니다.

흐름 로그를 생성할 때 `reject-reason` 필드가 포함된 사용자 지정 형식을 사용해야 합니다.

흐름 로그를 볼 때 VPC BPA로 인해 ENI에 대한 트래픽이 거부된 경우 흐름 로그 항목에 `reject-reason`이 BPA로 표시됩니다.

VPC 흐름 로그의 표준 제한 사항 외에도 VPC BPA와 관련된 다음 제한 사항에 유의하세요.

- VPC BPA의 흐름 로그에는 건너뛴 레코드가 포함되지 않습니다.
- 흐름 로그에 `bytes` 필드를 포함하더라도 VPC BPA의 흐름 로그에는 `bytes`가 포함되지 않습니다.

CloudTrail을 사용하여 제외 항목 삭제 추적

이 섹션에서는 AWS CloudTrail을 사용하여 VPC BPA 제외 항목 삭제를 모니터링하고 추적하는 방법을 설명합니다.

AWS Management Console

https://console.aws.amazon.com/cloudtrailv2/의 AWS CloudTrail 콘솔에서 리소스 유형 > `AWS::EC2::VPCBlockPublicAccessExclusion`을 조회하면 CloudTrail 이벤트 기록의 삭제된 제외 항목을 볼 수 있습니다.

AWS CLI

`lookup-events` 명령을 사용하여 제외 항목 삭제와 관련된 이벤트를 볼 수 있습니다.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceType,AttributeValue=AWS::EC2::VPCBlockPublicAccessExclusion
```

Reachability Analyzer를 사용하여 연결이 차단되었는지 확인

VPC Reachability Analyzer를 사용하면 VPC BPA 설정을 포함한 지정된 네트워크 구성에서 특정 네트워크 경로에 도달할 수 있는지 여부를 평가할 수 있습니다.

Reachability Analyzer의 리전별 가용성에 대한 자세한 내용은 Reachability Analyzer 설명서의 고려 사항을 참조하세요.

AWS Management Console

1. https://console.aws.amazon.com/networkinsights/home#ReachabilityAnalyzer에서 AWS Network Insights 콘솔을 엽니다.
2. 경로 생성 및 분석을 선택하세요.
3. 소스 유형에서 인터넷 게이트웨이를 선택하고 소스 드롭다운에서 트래픽을 차단할 인터넷 게이트웨이를 선택합니다.
4. 대상 유형에서 인스턴스를 선택하고 대상 드롭다운에서 트래픽을 차단할 인스턴스를 선택합니다.
5. 경로 생성 및 분석을 선택하세요.
6. 분석이 완료될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다.
7. 완료되면 연결성 상태가 연결할 수 없음으로 나타나야 하고 경로 세부 정보에 이 연결성 문제의 원인이 `VPC_BLOCK_PUBLIC_ACCESS_ENABLED` 로 표시됩니다.

AWS CLI

1. 소스의 트래픽을 차단하려는 인터넷 게이트웨이의 ID와 대상의 트래픽을 차단하려는 인스턴스의 ID를 사용하여 네트워크 경로를 생성합니다.

```
aws ec2 --region us-east-2 create-network-insights-path --source igw-id --
destination instance-id --protocol TCP
```

2. 네트워크 경로에 대한 분석 시작:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-
path-id nip-id
```

3. 분석 결과 검색:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-
insights-analysis-ids nia-id
```

4. 연결성이 부족한 경우 `VPC_BLOCK_PUBLIC_ACCESS_ENABLED`가 `ExplanationCode`인지 확인합니다.

고급 예제

이 섹션에는 VPC 퍼블릭 액세스 차단 기능이 다양한 시나리오에서 작동하는 방식을 이해하는 데 도움이 되는 고급 예제가 포함되어 있습니다. 각 시나리오는 이전 시나리오를 기반으로 구축되므로 순서대로 단계를 완료하는 것이 중요합니다.

Important

프로덕션 계정에서 이 예제를 실행하지 마세요. 프로덕션 계정에서 VPC BPA를 활성화하기 전에 인터넷 액세스가 필요한 워크로드를 면밀히 검토할 것을 강력히 권장합니다.

Note

VPC BPA 기능을 완전히 이해하려면 계정에 특정 리소스가 필요합니다. 이 섹션에서는 이 기능의 작동 방식을 완전히 이해하는 데 필요한 리소스를 프로비저닝하는 데 사용할 수 있는 CloudFormation 템플릿을 제공합니다. CloudFormation 템플릿을 사용하여 프로비저닝하는 리소스와 Network Access Analyzer 및 Reachability Analyzer로 수행하는 분석에는 관련된 비용

이 있습니다. 이 섹션의 템플릿을 사용하는 경우 이 예제를 완료한 후 정리 단계를 완료해야 합니다.

내용

- CloudFormation 템플릿 배포(선택)
- Network Access Analyzer를 사용하여 VPC BPA의 영향 확인
- 시나리오 1 - VPC BPA가 켜져 있지 않은 인스턴스에 연결
- 시나리오 2 - 양방향 모드에서 VPC BPA 켜기
- 시나리오 3 - VPC BPA 모드를 수신 전용으로 변경
- 시나리오 4 - 제외 생성
- 시나리오 5 - 제외 모드 수정
- 시나리오 6 - VPC BPA 모드 수정
- 정리

CloudFormation 템플릿 배포(선택)

이 기능의 작동 방식을 확인하려면 VPC, 서브넷, 인스턴스 및 기타 리소스가 필요합니다. 이 실습을 쉽게 완료할 수 있도록 아래 시나리오에 필요한 리소스를 신속하게 배포할 수 있는 CloudFormation 템플릿을 제공합니다. 이 단계는 선택 사항이며이 섹션의 시나리오에서 다이어그램을 볼 수 있습니다.

> **Note**
>
> - NAT 게이트웨이 및 퍼블릭 IPv4 주소의 비용과 같이 CloudFormation 템플릿을 사용하여 이 섹션에서 생성하는 리소스에는 관련된 비용이 있습니다. 과도한 비용이 발생하지 않도록 정리 단계를 완료하여 이 예제를 위해 생성된 모든 리소스를 제거해야 합니다.
> - 이 CloudFormation 템플릿은 VPC BPA에 필요한 기본 리소스를 생성하지만 VPC BPA 기능 자체는 활성화하지 않습니다. 여기에 배포된 리소스는 VPC BPA 기능을 별도로 활성화하도록 선택한 후 이를 이해하고 테스트하는 데 도움이 됩니다.

이 템플릿은 계정에 다음 리소스를 생성합니다.

- 외부 전용 인터넷 게이트웨이
- 인터넷 게이트웨이

- NAT 게이트웨이
- 퍼블릭 서브넷 2개
- 프라이빗 서브넷 1개
- 퍼블릭 및 프라이빗 IPv4 주소가 있는 EC2 인스턴스 2개
- IPv6 주소 및 프라이빗 IPv4 주소가 있는 EC2 인스턴스 1개
- 프라이빗 IPv4 주소만 있는 EC2 인스턴스 1개
- SSH 및 ICMP 인바운드 트래픽이 허용되고 모든 아웃바운드 트래픽이 허용되는 보안 그룹
- VPC 흐름 로그
- 서브넷 B의 EC2 Instance Connect 엔드포인트 1개

아래 템플릿을 복사하여 .yaml 파일로 저장합니다.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Creates a VPC with public and private subnets, NAT gateway, and EC2
 instances for VPC BPA.

Parameters:
  InstanceAMI:
    Description: ID of the Amazone Machine Image (AMI) to use with the instances
 launched by this template
    Type: AWS::EC2::Image::Id
  InstanceType:
    Description: EC2 Instance type to use with the instances launched by this template
    Type: String
    Default: t2.micro

Resources:

  # VPC
  VPCBPA:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
      InstanceTenancy: default
      Tags:
        - Key: Name
          Value: VPC BPA
```

```
# VPC IPv6 CIDR
VPCBPAIpv6CidrBlock:
  Type: AWS::EC2::VPCCidrBlock
  Properties:
    VpcId: !Ref VPCBPA
    AmazonProvidedIpv6CidrBlock: true

# EC2 Key Pair
VPCBPAKeyPair:
  Type: AWS::EC2::KeyPair
  Properties:
    KeyName: vpc-bpa-key

# Internet Gateway
VPCBPAInternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: VPC BPA Internet Gateway

VPCBPAInternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId: !Ref VPCBPA
    InternetGatewayId: !Ref VPCBPAInternetGateway

# Egress-Only Internet Gateway
VPCBPAEgressOnlyInternetGateway:
  Type: AWS::EC2::EgressOnlyInternetGateway
  Properties:
    VpcId: !Ref VPCBPA

# Subnets
VPCBPAPublicSubnetA:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPCBPA
    CidrBlock: 10.0.1.0/24
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: VPC BPA Public Subnet A
```

```
  VPCBPAPublicSubnetB:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPCBPA
      CidrBlock: 10.0.2.0/24
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: VPC BPA Public Subnet B

  VPCBPAPrivateSubnetC:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPCBPA
      CidrBlock: 10.0.3.0/24
      MapPublicIpOnLaunch: false
      Ipv6CidrBlock: !Select [0, !GetAtt VPCBPA.Ipv6CidrBlocks]
      AssignIpv6AddressOnCreation: true
      Tags:
        - Key: Name
          Value: VPC BPA Private Subnet C

  # NAT Gateway
  VPCBPANATGateway:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt VPCBPANATGatewayEIP.AllocationId
      SubnetId: !Ref VPCBPAPublicSubnetB
      Tags:
        - Key: Name
          Value: VPC BPA NAT Gateway

  VPCBPANATGatewayEIP:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc
      Tags:
        - Key: Name
          Value: VPC BPA NAT Gateway EIP

  # Route Tables
  VPCBPAPublicRouteTable:
    Type: AWS::EC2::RouteTable
```

```
    Properties:
      VpcId: !Ref VPCBPA
      Tags:
        - Key: Name
          Value: VPC BPA Public Route Table

  VPCBPAPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: VPCBPAInternetGatewayAttachment
    Properties:
      RouteTableId: !Ref VPCBPAPublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref VPCBPAInternetGateway

  VPCBPAPublicSubnetARouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      SubnetId: !Ref VPCBPAPublicSubnetA
      RouteTableId: !Ref VPCBPAPublicRouteTable

  VPCBPAPublicSubnetBRouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      SubnetId: !Ref VPCBPAPublicSubnetB
      RouteTableId: !Ref VPCBPAPublicRouteTable

  VPCBPAPrivateRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPCBPA
      Tags:
        - Key: Name
          Value: VPC BPA Private Route Table

  VPCBPAPrivateRoute:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref VPCBPAPrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref VPCBPANATGateway

  VPCBPAPrivateSubnetCRoute:
    Type: AWS::EC2::Route
    Properties:
```

```
      RouteTableId: !Ref VPCBPAPrivateRouteTable
      DestinationIpv6CidrBlock: ::/0
      EgressOnlyInternetGatewayId: !Ref VPCBPAEgressOnlyInternetGateway

  VPCBPAPrivateSubnetCRouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      SubnetId: !Ref VPCBPAPrivateSubnetC
      RouteTableId: !Ref VPCBPAPrivateRouteTable

  # EC2 Instances Security Group
  VPCBPAInstancesSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: VPC BPA Instances Security Group
      GroupDescription: Allow SSH and ICMP access
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: icmp
          FromPort: -1
          ToPort: -1
          CidrIp: 0.0.0.0/0
      VpcId: !Ref VPCBPA
      Tags:
        - Key: Name
          Value: VPC BPA Instances Security Group

  # EC2 Instances
  VPCBPAInstanceA:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: t2.micro
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPublicSubnetA
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance A
```

```
  VPCBPAInstanceB:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPublicSubnetB
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance B

  VPCBPAInstanceC:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance C

  VPCBPAInstanceD:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      NetworkInterfaces:
        - DeviceIndex: '0'
          GroupSet:
            - !Ref VPCBPAInstancesSecurityGroup
          SubnetId: !Ref VPCBPAPrivateSubnetC
          Ipv6AddressCount: 1
      Tags:
        - Key: Name
          Value: VPC BPA Instance D

  # Flow Logs IAM Role
  VPCBPAFlowLogRole:
```

```
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: vpc-flow-logs.amazonaws.com
            Action: 'sts:AssumeRole'
      Tags:
        - Key: Name
          Value: VPC BPA Flow Logs Role

  VPCBPAFlowLogPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: VPC-BPA-FlowLogsPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - 'logs:CreateLogGroup'
              - 'logs:CreateLogStream'
              - 'logs:PutLogEvents'
              - 'logs:DescribeLogGroups'
              - 'logs:DescribeLogStreams'
            Resource: '*'
      Roles:
        - !Ref VPCBPAFlowLogRole

  # Flow Logs
  VPCBPAFlowLog:
    Type: AWS::EC2::FlowLog
    Properties:
      ResourceId: !Ref VPCBPA
      ResourceType: VPC
      TrafficType: ALL
      LogDestinationType: cloud-watch-logs
      LogGroupName: /aws/vpc-flow-logs/VPC-BPA
      DeliverLogsPermissionArn: !GetAtt VPCBPAFlowLogRole.Arn
      LogFormat: '${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr}
 ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-
status} ${vpc-id} ${subnet-id} ${instance-id} ${tcp-flags} ${type} ${pkt-srcaddr}
```

```
 ${pkt-dstaddr} ${region} ${az-id} ${sublocation-type} ${sublocation-id} ${pkt-src-aws-
service} ${pkt-dst-aws-service} ${flow-direction} ${traffic-path} ${reject-reason}'
      Tags:
        - Key: Name
          Value: VPC BPA Flow Logs

  # EC2 Instance Connect Endpoint
  VPCBPAEC2InstanceConnectEndpoint:
    Type: AWS::EC2::InstanceConnectEndpoint
    Properties:
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      SubnetId: !Ref VPCBPAPublicSubnetB

Outputs:
  VPCBPAVPCId:
    Description: A reference to the created VPC
    Value: !Ref VPCBPA
    Export:
      Name: vpc-id

  VPCBPAPublicSubnetAId:
    Description: The ID of the public subnet A
    Value: !Ref VPCBPAPublicSubnetA

  VPCBPAPublicSubnetAName:
    Description: The name of the public subnet A
    Value: VPC BPA Public Subnet A

  VPCBPAPublicSubnetBId:
    Description: The ID of the public subnet B
    Value: !Ref VPCBPAPublicSubnetB

  VPCBPAPublicSubnetBName:
    Description: The name of the public subnet B
    Value: VPC BPA Public Subnet B

  VPCBPAPrivateSubnetCId:
    Description: The ID of the private subnet C
    Value: !Ref VPCBPAPrivateSubnetC

  VPCBPAPrivateSubnetCName:
    Description: The name of the private subnet C
    Value: VPC BPA Private Subnet C
```

```
VPCBPAInstanceAId:
  Description: The ID of instance A
  Value: !Ref VPCBPAInstanceA

VPCBPAInstanceBId:
  Description: The ID of instance B
  Value: !Ref VPCBPAInstanceB

VPCBPAInstanceCId:
  Description: The ID of instance C
  Value: !Ref VPCBPAInstanceC

VPCBPAInstanceDId:
  Description: The ID of instance D
  Value: !Ref VPCBPAInstanceD
```

AWS Management Console

1. CloudFormation에서 https://console.aws.amazon.com/cloudformation/ 콘솔을 엽니다.
2. 스택 생성을 선택하고 .yaml 템플릿 파일을 업로드합니다.
3. 단계를 진행하여 템플릿을 시작합니다. 이미지 ID와 인스턴스 유형(예: t2.micro)을 입력해야 합니다. 또한 CloudFormation에서 흐름 로그를 생성하고 CloudWatch에 로깅할 수 있는 권한이 있는 IAM 역할을 생성하도록 허용해야 합니다.
4. 스택이 시작되면 이벤트 탭을 표시하여 진행 상황을 확인하고 계속하기 전에 스택이 완료되었는지 확인합니다.

AWS CLI

1. 다음 명령을 실행하여 CloudFormation 스택을 생성합니다.

```
aws cloudformation create-stack --stack-name VPC-BPA-stack --template-body
 file://sampletemplate.yaml --capabilities CAPABILITY_IAM --region us-east-2
```

출력:

```
{
    "StackId": "arn:aws:cloudformation:us-east-2:470889052923:stack/VPC-BPA-
stack/8a7a2cc0-8001-11ef-b196-06386a84b72f"
```

```
}
```

2. 진행 상황을 확인하고 계속하기 전에 스택이 완료되었는지 확인합니다.

```
aws cloudformation describe-stack-events --stack-name VPC-BPA-stack --region us-east-2
```

Network Access Analyzer를 사용하여 VPC BPA의 영향 확인

이 섹션에서는 Network Access Analyzer를 사용하여 인터넷 게이트웨이를 사용하는 계정의 리소스를 확인합니다. 이 분석을 사용하여 계정에서 VPC BPA를 켜고 트래픽을 차단할 때의 영향을 파악할 수 있습니다.

Network Access Analyzer의 리전별 가용성에 대한 자세한 내용은 Network Access Analyzer 설명서의 제한 사항을 참조하세요.

AWS Management Console

1. https://console.aws.amazon.com/networkinsights/에서 AWS Network Insights 콘솔을 엽니다.
2. Network Access Analyzer를 선택합니다.
3. 네트워크 액세스 범위 생성을 선택합니다.
4. VPC 퍼블릭 액세스 차단의 영향 평가를 선택하고 다음을 선택합니다.
5. 템플릿은 이미 계정의 인터넷 게이트웨이를 오가는 트래픽을 분석하도록 구성되어 있습니다. 소스 및 대상에서 이를 확인할 수 있습니다.
6. 다음을 선택합니다.
7. 네트워크 액세스 범위 생성을 선택합니다.
8. 방금 생성한 범위를 선택하고 분석을 선택합니다.
9. 분석이 완료될 때까지 기다립니다.
10. 분석 결과를 확인합니다. 분석 결과 아래의 각 행에는 패킷이 네트워크에서 계정의 인터넷 게이트웨이를 들어오고 나갈 때 거칠 수 있는 네트워크 경로가 표시됩니다. 이 경우 VPC BPA를 켜고 이러한 분석 결과에 나타나는 VPC 및/또는 서브넷 중 VPC BPA 제외 항목으로 구성된 것이 없는 경우 해당 VPC 및 서브넷으로의 트래픽이 제한됩니다.
11. 각 분석 결과를 확인하여 VPC BPA가 VPC의 리소스에 미치는 영향을 파악합니다.

영향 분석이 완료되었습니다.

AWS CLI

1. 네트워크 액세스 범위 생성:

```
aws ec2 create-network-insights-access-scope --match-paths
 "Source={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 "Destination={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 --region us-east-2
```

출력:

```
{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-2:470889052923:network-insights-access-scope/nis-04cad3c4b3a1d5e3e",
    "CreatedDate": "2024-09-30T15:55:53.171000+00:00",
    "UpdatedDate": "2024-09-30T15:55:53.171000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      },
      {
        "Destination": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
```

```
}
```

2. 범위 분석 시작:

```
aws ec2 start-network-insights-access-scope-analysis --network-insights-access-
scope-id nis-04cad3c4b3a1d5e3e --region us-east-2
```

출력:

```
{
  "NetworkInsightsAccessScopeAnalysis": {
    "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-2:470889052923:network-insights-access-scope-analysis/
nisa-0aa383a1938f94cd",
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "Status": "running",
    "StartDate": "2024-09-30T15:56:59.109000+00:00",
    "AnalyzedEniCount": 0
  }
}
```

3. 분석 결과 가져오기:

```
aws ec2 get-network-insights-access-scope-analysis-findings --network-insights-
access-scope-analysis-id nisa-0aa383a1938f94cd1 --region us-east-2 --max-items 1
```

출력:

```
{
  "AnalysisFindings": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
      "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
      "FindingId": "AnalysisFinding-1",
      "FindingComponents": [
        {
          "SequenceNumber": 1,
          "Component": {
            "Id": "igw-04a5344b4e30486f1",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:internet-gateway/
igw-04a5344b4e30486f1",
```

```
            "Name": "VPC BPA Internet Gateway"
          },
          "OutboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ]
          },
          "InboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ],
            "DestinationPortRanges": [
              {
                "From": 22,
                "To": 22
              }
            ],
            "Protocol": "6",
            "SourceAddresses": [
              "0.0.0.0/5",
              "100.0.0.0/10",
              "96.0.0.0/6"
            ],
            "SourcePortRanges": [
              {
                "From": 0,
                "To": 65535
              }
            ]
          },
          "Vpc": {
            "Id": "vpc-0762547ec48b6888d",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:vpc/
vpc-0762547ec48b6888d",
            "Name": "VPC BPA"
          }
        },
        {
          "SequenceNumber": 2,
          "AclRule": {
            "Cidr": "0.0.0.0/0",
            "Egress": false,
            "Protocol": "all",
            "RuleAction": "allow",
```

```
            "RuleNumber": 100
          },
          "Component": {
            "Id": "acl-06194fc3a4a03040b",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:network-acl/
acl-06194fc3a4a03040b"
          }
        },
        {
          "SequenceNumber": 3,
          "Component": {
            "Id": "sg-093dde06415d03924",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:security-group/
sg-093dde06415d03924",
            "Name": "VPC BPA Instances Security Group"
          },
          "SecurityGroupRule": {
            "Cidr": "0.0.0.0/0",
            "Direction": "ingress",
            "PortRange": {
              "From": 22,
              "To": 22
            },
            "Protocol": "tcp"
          }
        },
        {
          "SequenceNumber": 4,
          "AttachedTo": {
            "Id": "i-058db34f9a0997895",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:instance/
i-058db34f9a0997895",
            "Name": "VPC BPA Instance A"
          },
          "Component": {
            "Id": "eni-0fa23f2766f03b286",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:network-interface/
eni-0fa23f2766f03b286"
          },
          "InboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ],
            "DestinationPortRanges": [
```

```
              {
                "From": 22,
                "To": 22
              }
            ],
            "Protocol": "6",
            "SourceAddresses": [
              "0.0.0.0/5",
              "100.0.0.0/10",
              "96.0.0.0/6"
            ],
            "SourcePortRanges": [
              {
                "From": 0,
                "To": 65535
              }
            ]
          },
          "Subnet": {
            "Id": "subnet-035d235a762eeed04",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:subnet/
subnet-035d235a762eeed04",
            "Name": "VPC BPA Public Subnet A"
          },
          "Vpc": {
            "Id": "vpc-0762547ec48b6888d",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:vpc/
vpc-0762547ec48b6888d",
            "Name": "VPC BPA"
          }
        }
      ]
    }
  ],
  "AnalysisStatus": "succeeded",
  "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
  "NextToken":
 "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}
```


결과는 계정의 모든 VPC에서 인터넷 게이트웨이를 들어오고 나가는 트래픽을 보여줍니다. 결과는 "분석 결과"로 구성됩니다. "FindingId": "AnalysisFinding-1"은 이것이 분석의 첫 번째 결과임을 나타냅니다. 여러 분석 결과가 있는 경우 각 분석 결과는 VPC BPA를 켜면 영향을 받는 트

래픽 흐름을 나타냅니다. 첫 번째 분석 결과는 인터넷 게이트웨이("SequenceNumber": 1)에서 시작된 트래픽이 NACL("SequenceNumber": 2)을 거쳐 보안 그룹("SequenceNumber": 3)으로 전달되고 인스턴스("SequenceNumber": 4)에서 종료된 것을 보여줍니다.

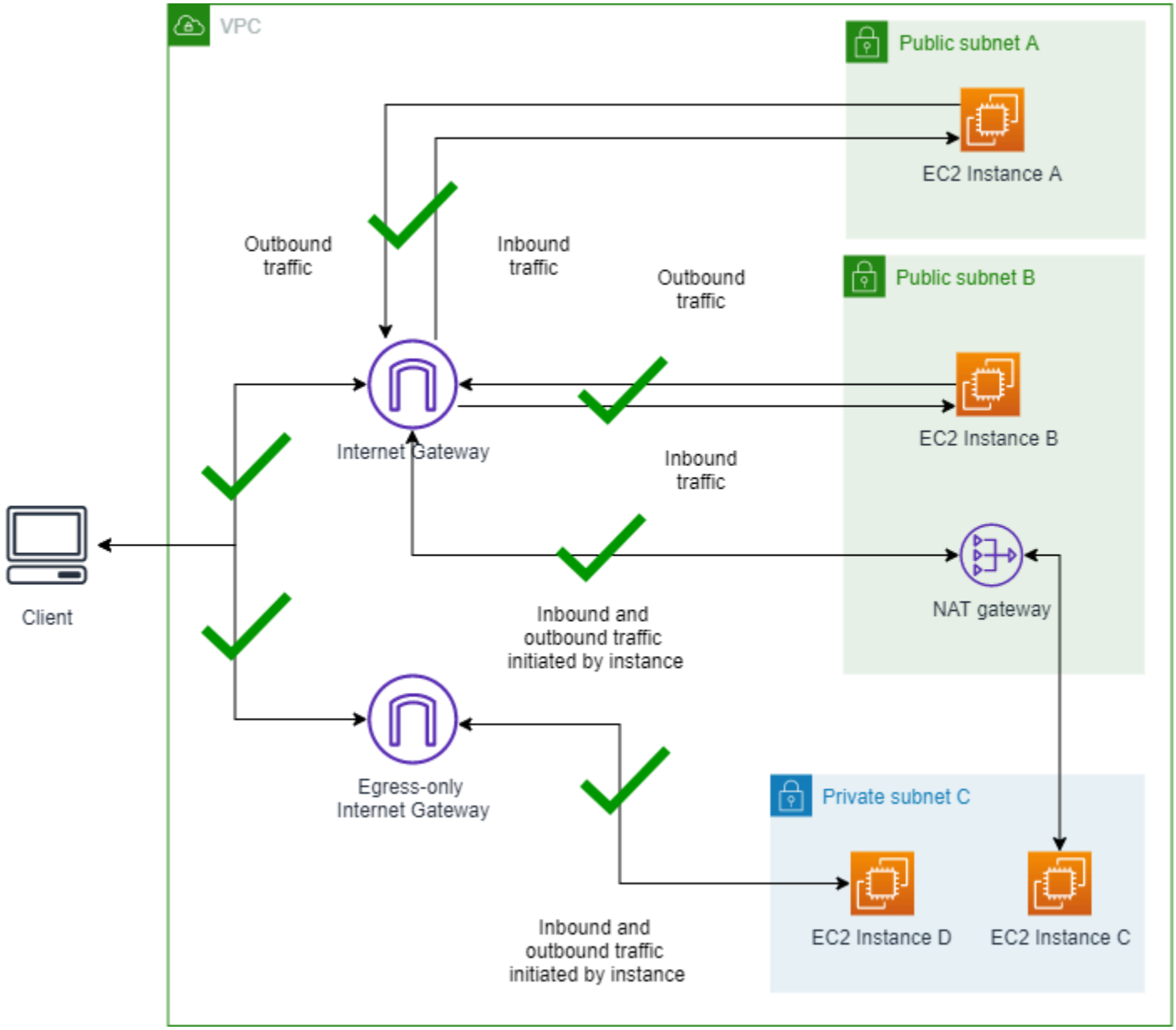
4. 분석 결과를 확인하여 VPC BPA가 VPC의 리소스에 미치는 영향을 파악합니다.

영향 분석이 완료되었습니다.

시나리오 1 - VPC BPA가 켜져 있지 않은 인스턴스에 연결

이 단원에서는 인터넷 게이트웨이를 통해 인터넷에서 퍼블릭 서브넷 A 및 B의 EC2 인스턴스에 연결할 수 있으므로 인바운드 트래픽과 아웃바운드 트래픽을 모두 허용합니다. 프라이빗 서브넷의 인스턴스 C와 D는 NAT 게이트웨이 또는 송신 전용 인터넷 게이트웨이를 통해 아웃바운드 트래픽을 시작할 수 있지만 인터넷에서 직접 연결할 수는 없습니다. 이 설정은 일부 리소스에 대한 인터넷 액세스를 제공하는 동시에 다른 리소스를 보호합니다. 이 설정의 목적은 기준을 설정하고 VPC BPA를 활성화하기 전에 모든 인스턴스에 연결할 수 있는지 확인하기 위해 모든 인스턴스에 연결하고 퍼블릭 IP 주소를 ping합니다.

VPC BPA가 켜져 있지 않은 VPC의 다이어그램:



1.1 인스턴스에 연결

이 섹션의 단계에 따라 VPC BPA가 꺼져 있는 인스턴스에 연결하여 문제 없이 연결할 수 있는지 확인합니다. 이 예제의 CloudFormation으로 생성된 모든 인스턴스에는 "VPC BPA Instance A"와 같은 이름이 있습니다.

AWS Management Console

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 A 세부 정보를 엽니다.

3. EC2 Instance Connect > EC2 Instance Connect 엔드포인트를 사용하여 연결 옵션을 통해 인스턴스 A에 연결합니다.
4. 연결을 선택합니다. 인스턴스에 성공적으로 연결되면 www.amazon.com을 ping하여 인터넷으로 아웃바운드 요청을 보낼 수 있는지 확인합니다.
5. 인스턴스 A에 연결할 때와 동일한 방법을 사용하여 인스턴스 B, C, D에 연결하고 ping www.amazon.com을 통해 인터넷으로 아웃바운드 요청을 보낼 수 있는지 테스트합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

출력:

```
Pinging 18.225.8.244 with 32 bytes of data:

Reply from 18.225.8.244: bytes=32 time=51ms TTL=110
Reply from 18.225.8.244: bytes=32 time=61ms TTL=110
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available. Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
```

```
Last login: Fri Sep 27 18:27:57 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING www-amazon-com.customer.fastly.net (18.65.233.187) 56(84) bytes of data.
64 bytes from 18.65.233.187 (18.65.233.187): icmp_seq=15 ttl=58 time=2.06 ms
64 bytes from 18.65.233.187 (18.65.233.187): icmp_seq=16 ttl=58 time=2.26 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 B를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
Reply from 3.18.106.198: bytes=32 time=83ms TTL=110
Reply from 3.18.106.198: bytes=32 time=54ms TTL=110
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id  i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.
Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Fri Sep 27 18:12:27 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

```
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=249 time=1.55 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=249 time=1.67 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.
Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Thu Sep 19 20:31:26 2024 from 10.0.2.86
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.75 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.97 ms
64 bytes from server-3-160-24-26.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=3 ttl=248 time=1.08 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

출력:

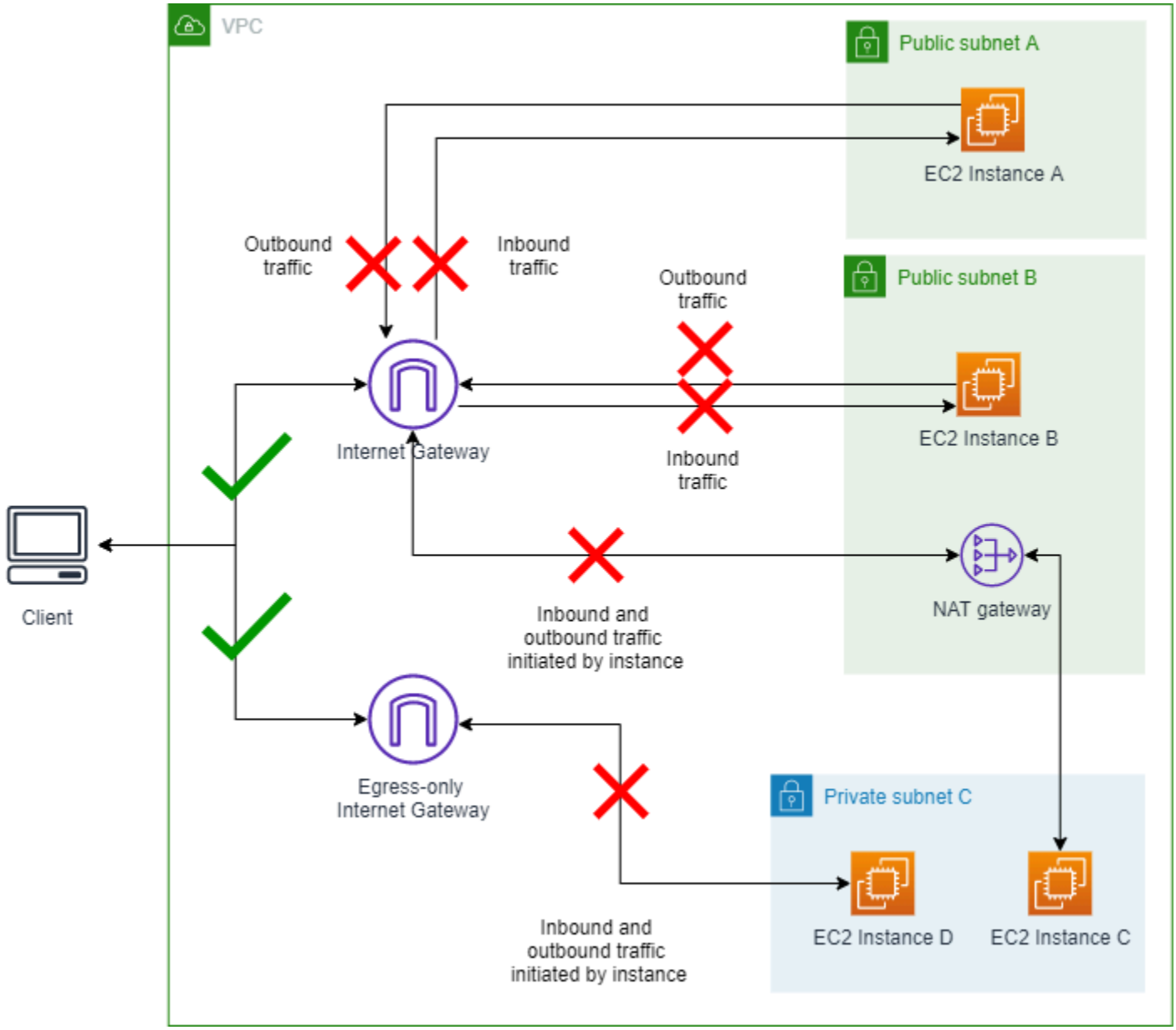
```
The authenticity of host '10.0.3.59 can't be established.
ECDSA key fingerprint is SHA256:c4naBCqbC61/cExDyccEproNU+1HHSpMSzl2J6cOtIZA8g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.3.59' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ _/
_/m/'
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121)) 56 data bytes
64 bytes from 2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121): icmp_seq=1 ttl=58 time=1.19 ms
64 bytes from 2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121): icmp_seq=2 ttl=58 time=1.38 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

시나리오 2 - 양방향 모드에서 VPC BPA 켜기

이 섹션에서는 VPC BPA를 켜고 계정의 인터넷 게이트웨이에서 들어오고 나가는 트래픽을 차단합니다.

BPA 양방향 모드가 켜져 있는 VPC의 다이어그램:



2.1 VPC BPA 양방향 모드 활성화

이 섹션을 완료하여 VPC BPA를 활성화합니다. VPC BPA 양방향 모드는 이 리전의 인터넷 게이트웨이 및 송신 전용 인터넷 게이트웨이(제외된 VPC 및 서브넷 제외)를 오가는 모든 트래픽을 차단합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 설정 편집을 선택합니다.

4. 퍼블릭 액세스 차단 켜기 및 양방향을 선택한 다음 변경 사항 저장을 선택합니다.
5. 상태가 켜기로 변경될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

이제 VPC BPA가 켜져 있습니다.

AWS CLI

1. modify-vpc-block-public-access-options 명령을 사용하여 VPC BPA 켜기:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-
gateway-block-mode block-bidirectional
```

VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

2. VPC BPA 상태 보기:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

2.2 인스턴스에 연결

이 섹션을 완료하여 인스턴스에 연결합니다.

AWS Management Console

1. 시나리오 1에서와 같이 인스턴스 A 및 인스턴스 B의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 차단되는지 확인합니다.
2. 시나리오 1에서와 같이EC2 Instance Connect > EC2 Instance Connect 엔드포인트를 사용하여 연결 옵션을 통해 인스턴스 A에 연결합니다. 엔드포인트 옵션을 사용해야 합니다.
3. 연결을 선택합니다. 인스턴스에 연결되면 ping www.amazon.com을 수행합니다. 모든 아웃바운드 트래픽이 차단되는지 확인합니다.
4. 인스턴스 A에 연결할 때와 동일한 방법을 사용하여 인스턴스 B, C, D에 연결하고 인터넷으로 아웃바운드 요청을 보낼 수 있는지 테스트합니다. 모든 아웃바운드 트래픽이 차단되는지 확인합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

출력:

```
Pinging 18.225.8.244 with 32 bytes of data:

Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
The authenticity of host '10.0.1.85' can't be established.
ECDSA key fingerprint is SHA256:3zo/gSss+HAZ+7eTyWlOB/Ke04IM+hadjsoLJeRTWBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.85' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 14:16:53 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 B를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id  i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

출력:

```
The authenticity of host '10.0.2.98' can't be established.
ECDSA key fingerprint is SHA256:0IjXKKyVlDthcCfI0IPIJMUiItAOLYKRNLGTYURnFXo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.98' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Fri Sep 27 14:18:16 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Tue Sep 24 15:17:56 2024 from 10.0.2.86
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ _/
_/m/'
Last login: Fri Sep 27 16:42:01 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:8200:7:49a5:5fd4:b121
 (2600:9000:25f3:8200:7:49a5:5fd4:b121)) 56 data bytes
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

2.3 선택 사항: Reachability Analyzer를 사용하여 연결이 차단되었는지 확인

VPC Reachability Analyzer를 사용하면 VPC BPA 설정을 포함한 지정된 네트워크 구성에서 특정 네트워크 경로에 도달할 수 있는지 여부를 파악할 수 있습니다. 이 예제에서는 이전에 시도한 것과 동일한 네트워크 경로를 분석하여 VPC BPA가 연결이 실패하는 이유인지 확인합니다.

AWS Management Console

1. https://console.aws.amazon.com/networkinsights/home#ReachabilityAnalyzer에서 Network Insights 콘솔로 이동합니다.
2. 경로 생성 및 분석을 선택하세요.
3. 소스 유형에서 인터넷 게이트웨이를 선택하고 소스 드롭다운에서 인터넷 게이트웨이에 태그가 지정된 VPC BPA 인터넷 게이트웨이를 선택합니다.
4. 대상 유형에서 인스턴스를 선택하고 대상 드롭다운에서 VPC BPA 인스턴스 A로 태그가 지정된 인스턴스를 선택합니다.
5. 경로 생성 및 분석을 선택하세요.
6. 분석이 완료될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다.
7. 완료되면 연결성 상태가 연결할 수 없음으로 나타나야 하고 경로 세부 정보에 `VPC_BLOCK_PUBLIC_ACCESS_ENABLED`가 원인으로 표시되어야 합니다.

AWS CLI

1. 인터넷 게이트웨이에 태그가 지정된 VPC BPA 인터넷 게이트웨이의 ID와 인스턴스에 태그가 지정된 VPC BPA 인스턴스 A의 ID를 사용하여 네트워크 경로 생성:

```
aws ec2 --region us-east-2 create-network-insights-path --source igw-id --
destination instance-id --protocol TCP
```

2. 네트워크 경로에 대한 분석 시작:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-
path-id nip-id
```

3. 분석 결과 검색:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-
insights-analysis-ids nia-id
```

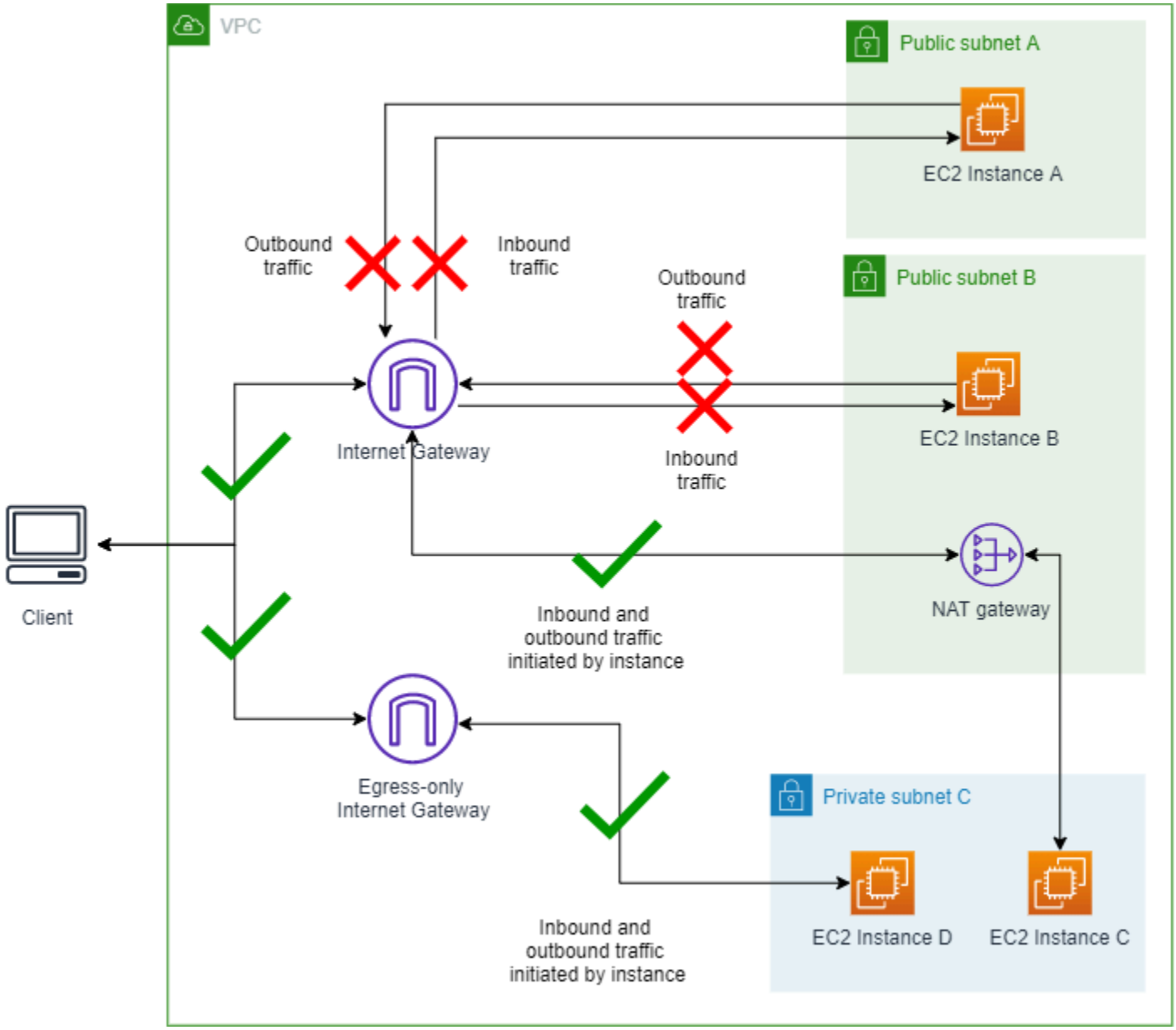
4. 연결성이 부족한 경우 `VPC_BLOCK_PUBLIC_ACCESS_ENABLED`가 `ExplanationCode`인지 확인합니다.

흐름 로그를 사용하여 VPC BPA 영향 모니터링도 가능합니다.

시나리오 3 - VPC BPA 모드를 수신 전용으로 변경

이 섹션에서는 VPC BPA 트래픽 방향을 변경하고 NAT 게이트웨이 또는 송신 전용 인터넷 게이트웨이를 사용하는 트래픽만 허용합니다. BPA는 인터넷 게이트웨이를 통한 인바운드 트래픽을 차단하므로 퍼블릭 서브넷의 EC2 인스턴스 A와 B는 인터넷을 통해 연결할 수 없습니다. 프라이빗 서브넷의 인스턴스 C와 D는 NAT 게이트웨이와 송신 전용 인터넷 게이트웨이를 통해 아웃바운드 트래픽을 시작할 수 있으므로 인터넷에 계속 연결할 수 있습니다.

BPA 수신 전용 모드가 켜져 있는 VPC의 다이어그램:



3.1 모드를 수신 전용으로 변경

이 섹션을 완료하여 모드를 변경합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 차단 탭에서 퍼블릭 액세스 설정 편집을 선택합니다.
4. VPC 콘솔에서 퍼블릭 액세스 설정을 수정하고 방향을 수신 전용으로 변경합니다.

5. 변경 사항을 저장하고 상태가 업데이트될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

1. VPC BPA 모드 수정:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-ingress
```

VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

2. VPC BPA 상태 보기:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

3.2 인스턴스에 연결

이 섹션을 완료하여 인스턴스에 연결합니다.

AWS Management Console

1. 시나리오 1에서와 같이 인스턴스 A 및 인스턴스 B의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 차단되는지 확인합니다.
2. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 A 및 B에 연결하고 www.amazon.com을 ping합니다. 인스턴스 A 또는 B에서는 인터넷의 퍼블릭 사이트를 ping할 수 없으며 트래픽이 차단되는지 확인합니다.
3. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 C 및 D에 연결하고 www.amazon.com을 ping합니다. 인스턴스 C 또는 D에서는 인터넷의 퍼블릭 사이트를 ping할 수 있으며 트래픽이 허용되는지 확인합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

출력:

```
Pinging 18.225.8.244 with 32 bytes of data:

Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
The authenticity of host '10.0.1.85' can't be established.
ECDSA key fingerprint is SHA256:3zo/gSss+HAZ+7eTyWlOB/Ke04IM+hadjsoLJeRTWBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.85' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 14:16:53 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 B를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
```

```
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

출력:

```
The authenticity of host '10.0.2.98 ' can't be established.
ECDSA key fingerprint is SHA256:0IjXKKyVlDthcCfI0IPIJMUiItAOLYKRNLGTYURnFXo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.98' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ /
/m/'
Last login: Fri Sep 27 14:18:16 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
```

```
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

Last login: Tue Sep 24 15:28:09 2024 from 10.0.2.86

[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com

PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.

64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.84 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.40 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 16:48:38 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121)) 56 data bytes
```

```
64 bytes from 2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121): icmp_seq=14 ttl=58 time=1.47 ms
64 bytes from 2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121): icmp_seq=16 ttl=58 time=1.59 ms
```

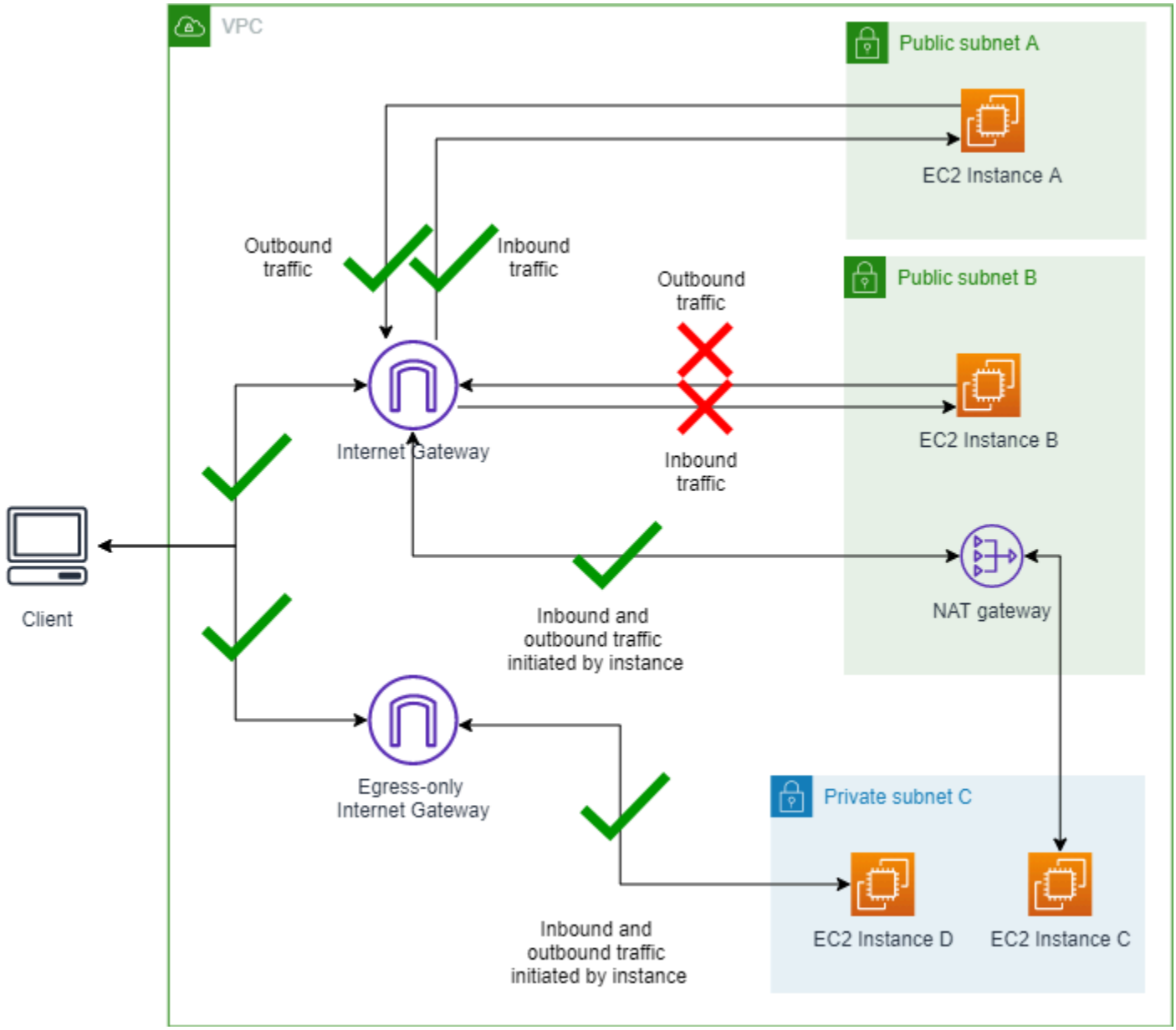
ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

시나리오 4 - 제외 생성

이 섹션에서는 제외 항목을 생성합니다. 그러면 VPC BPA는 제외 항목 없이 서브넷의 트래픽만 차단합니다. VPC BPA 제외 모드는 계정의 VPC BPA 모드 적용을 면제하여 양방향 또는 아웃바운드 전용(Egress-only) 액세스를 허용하는 모드로, 단일 VPC 또는 서브넷에 적용할 수 있습니다. 계정에서 VPC BPA가 활성화되지 않은 경우에도 VPC 및 서브넷에 대한 VPC BPA 제외 항목을 생성하여 VPC BPA가 켜져 있을 때 제외 항목에 대한 트래픽 중단이 없도록 할 수 있습니다.

이 예제에서는 VPC BPA가 제외 항목의 트래픽에 어떤 영향을 미치는지 보여주기 위해 서브넷 A에 대한 제외 항목을 생성합니다.

BPA 수신 전용 모드가 켜져 있는 VPC와 양방향 모드가 켜져 있는 서브넷 A의 다이어그램:



4.1 서브넷 A에 대한 제외 항목 생성

이 섹션을 완료하여 제외 항목을 생성합니다. VPC BPA 제외 모드는 계정의 VPC BPA 모드 적용을 면제하여 양방향 또는 아웃바운드 전용(Egress-only) 액세스를 허용하는 모드로, 단일 VPC 또는 서브넷에 적용할 수 있습니다. 계정에서 VPC BPA가 활성화되지 않은 경우에도 VPC 및 서브넷에 대한 VPC BPA 제외 항목을 생성하여 VPC BPA가 켜져 있을 때 제외 항목에 대한 트래픽 중단이 없도록 할 수 있습니다.

AWS Management Console

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

2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 차단 탭의 제외 항목에서 제외 항목 생성을 선택합니다.
4. VPC BPA 퍼블릭 서브넷 A를 선택하고 양방향 허용 방향이 선택되었는지 확인한 다음 제외 항목 생성을 선택합니다.
5. 제외 상태가 활성으로 변경될 때까지 기다립니다. 변경 사항을 보려면 제외 항목 테이블을 새로 고쳐야 할 수 있습니다.

제외 항목이 생성되었습니다.

AWS CLI

1. 제외 항목 허용 방향 수정:

```
aws ec2 --region us-east-2 create-vpc-block-public-access-exclusion --subnet-
id subnet-id --internet-gateway-exclusion-mode allow-bidirectional
```

2. 제외 상태가 업데이트되는 데 시간이 걸릴 수 있습니다. 제외 항목의 상태를 보려면:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusions --
exclusion-ids exclusion-id
```

4.2 인스턴스에 연결

이 섹션을 완료하여 인스턴스에 연결합니다.

AWS Management Console

1. 인스턴스 A의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 허용되는지 확인합니다.
2. 인스턴스 B의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 차단되는지 확인합니다.
3. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 A에 연결하고 www.amazon.com을 ping합니다. 인스턴스 A에서는 인터넷의 퍼블릭 사이트를 ping할 수 있으며 트래픽이 허용되는지 확인합니다.
4. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 B에 연결하고 www.amazon.com을 ping합니다. 인스턴스 B에서는 인터넷의 퍼블릭 사이트를 ping할 수 없으며 트래픽이 차단되는지 확인합니다.

5. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 C 및 D에 연결하고 www.amazon.com을 ping합니다. 인스턴스 C 또는 D에서는 인터넷의 퍼블릭 사이트를 ping할 수 있으며 트래픽이 허용되는지 확인합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

출력:

```
Pinging 18.225.8.244 with 32 bytes of data:

Reply from 18.225.8.244: bytes=32 time=51ms TTL=110
Reply from 18.225.8.244: bytes=32 time=61ms TTL=110
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 17:58:12 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=249 time=1.03 ms
```

```
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=249 time=1.72 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 B를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ /
/m/'
Last login: Fri Sep 27 18:12:03 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

Output

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ /
/m/'

Last login: Tue Sep 24 15:28:09 2024 from 10.0.2.86

[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com

PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.

64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.84 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.40 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

Output

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
```

```
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:00:52 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING
 www.amazon.com(g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologie
 (2600:141f:4000:59a::3bd4)) 56 data bytes
64 bytes from
 g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologies.com
 (2600:141f:4000:59a::3bd4): icmp_seq=1 ttl=48 time=15.9 ms
64 bytes from
 g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologies.com
 (2600:141f:4000:59a::3bd4): icmp_seq=2 ttl=48 time=15.8 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

4.3 선택 사항: Reachability Analyzer와의 연결 확인

이제 시나리오 2의 Reachability Analyzer에서 생성된 것과 동일한 네트워크 경로를 사용하여 새 분석을 실행할 수 있습니다. 퍼블릭 서브넷 A에 대한 제외 항목이 생성되었으므로 경로에 연결할 수 있는지 확인합니다.

Reachability Analyzer의 리전별 가용성에 대한 자세한 내용은 Reachability Analyzer 설명서의 고려 사항을 참조하세요.

AWS Management Console

1. Network Insights 콘솔에서 이전에 생성한 네트워크 경로를 선택한 후 분석 재실행을 선택하세요.
2. 분석이 완료될 때까지 기다립니다. 몇 분 정도 걸릴 수 있습니다.
3. 이제 경로가 연결 가능인지 확인합니다.

AWS CLI

1. 이전에 생성한 네트워크 경로 ID를 사용하여 새 분석 시작:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-
path-id nip-id
```

2. 분석 결과 검색:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-
insights-analysis-ids nia-id
```

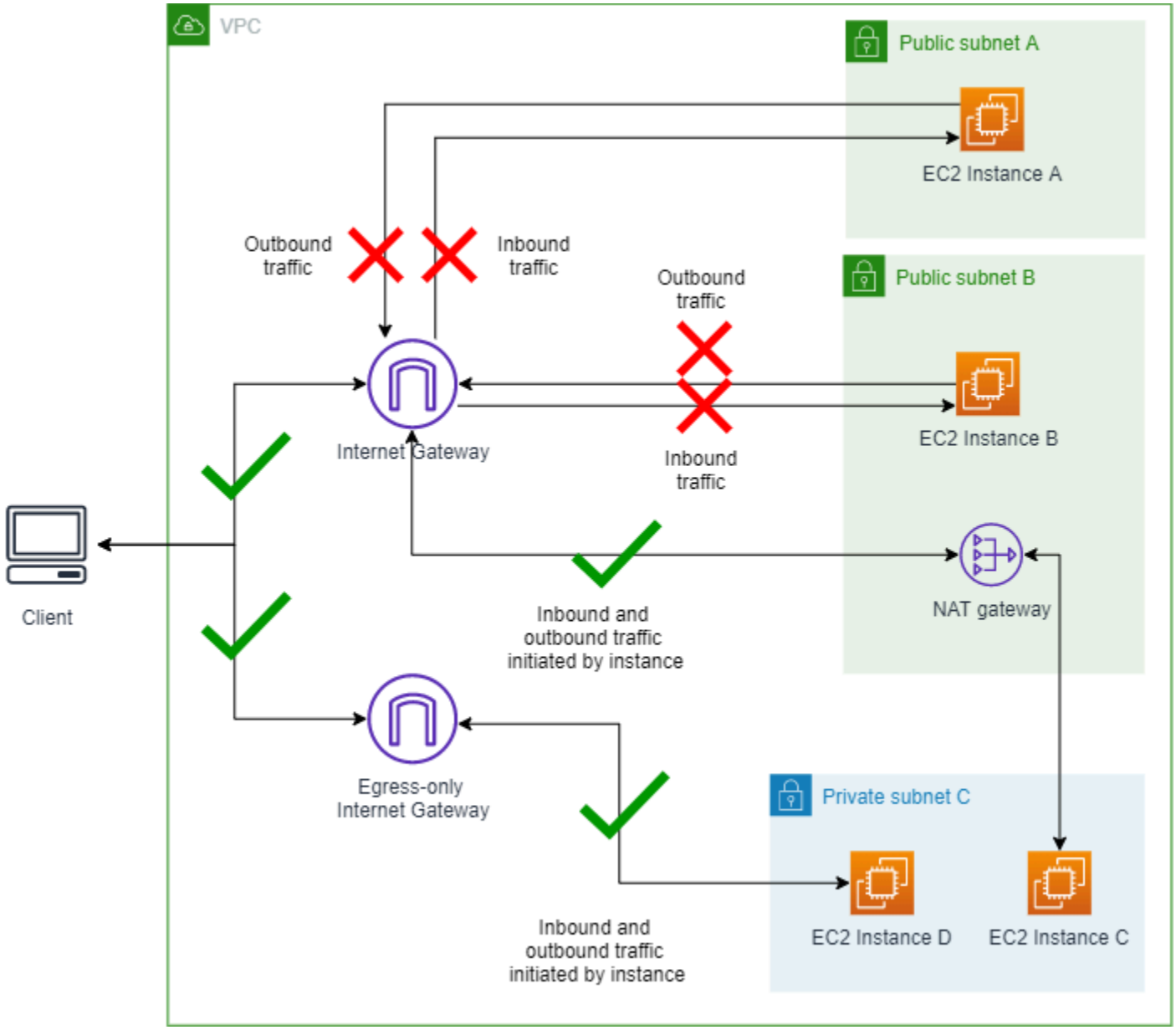
3. VPC_BLOCK_PUBLIC_ACCESS_ENABLED 설명 코드가 더 이상 존재하지 않는지 확인합니다.

시나리오 5 - 제외 모드 수정

이 섹션에서는 제외 항목의 허용 트래픽 방향을 변경하여 VPC BPA에 어떤 영향을 미치는지 확인합니다.

> **Note**
>
> 이 시나리오에서는 제외 모드를 송신 전용으로 변경합니다. 이렇게 하면 서브넷 A의 아웃바운드 전용 제외가 아웃바운드 트래픽을 허용하지 않습니다. 이는 아웃바운드 트래픽 허용을 예상하는 사용자의 직관에 어긋날 수 있습니다. 그러나 계정 수준 BPA가 인바운드 전용인 경우 아웃바운드 전용 제외는 무시되므로, 서브넷 A의 인터넷 게이트웨이 라우팅이 VPC BPA에 의해 제한되어 아웃바운드 트래픽이 차단됩니다. 서브넷 A에서 아웃바운드 트래픽을 활성화하려면 VPC BPA를 양방향 모드로 전환해야 합니다.

BPA 수신 전용 모드가 켜져 있는 VPC와 송신 전용 모드가 켜져 있는 서브넷 A 제외 항목의 다이어그램:



5.1 제외 항목 허용 방향을 송신 전용으로 변경

이 섹션을 완료하여 제외 항목 허용 방향을 변경합니다.

AWS Management Console

1. 시나리오 4에서 생성한 제외 항목을 편집하고 허용 방향을 송신 전용으로 변경합니다.
2. 변경 사항 저장을 선택합니다.
3. 제외 상태가 활성으로 변경될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다. 변경 사항을 보려면 제외 항목 테이블을 새로 고쳐야 할 수 있습니다.

AWS CLI

1. 제외 항목 허용 방향 수정:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-exclusion --exclusion-
id exclusion-id --internet-gateway-exclusion-mode allow-egress
```

   VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

2. 제외 상태가 업데이트되는 데 시간이 걸릴 수 있습니다. 제외 항목의 상태를 보려면:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusion
```

5.2 인스턴스에 연결

이 섹션을 완료하여 인스턴스에 연결합니다.

AWS Management Console

1. 인스턴스 A 및 B의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 차단되는지 확인합니다.
2. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 A 및 B에 연결하고 www.amazon.com을 ping합니다. 인스턴스 A 또는 B에서는 인터넷의 퍼블릭 사이트를 ping할 수 없으며 트래픽이 차단되는지 확인합니다.
3. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 C 및 D에 연결하고 www.amazon.com을 ping합니다. 인스턴스 C 또는 D에서는 인터넷의 퍼블릭 사이트를 ping할 수 있으며 트래픽이 허용되는지 확인합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

   출력:

```
Pinging 18.225.8.244 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 B를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:00:31 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121)) 56 data bytes
```

```
64 bytes from 2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121): icmp_seq=1 ttl=58 time=1.51 ms
64 bytes from 2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121): icmp_seq=2 ttl=58 time=1.49 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

출력:

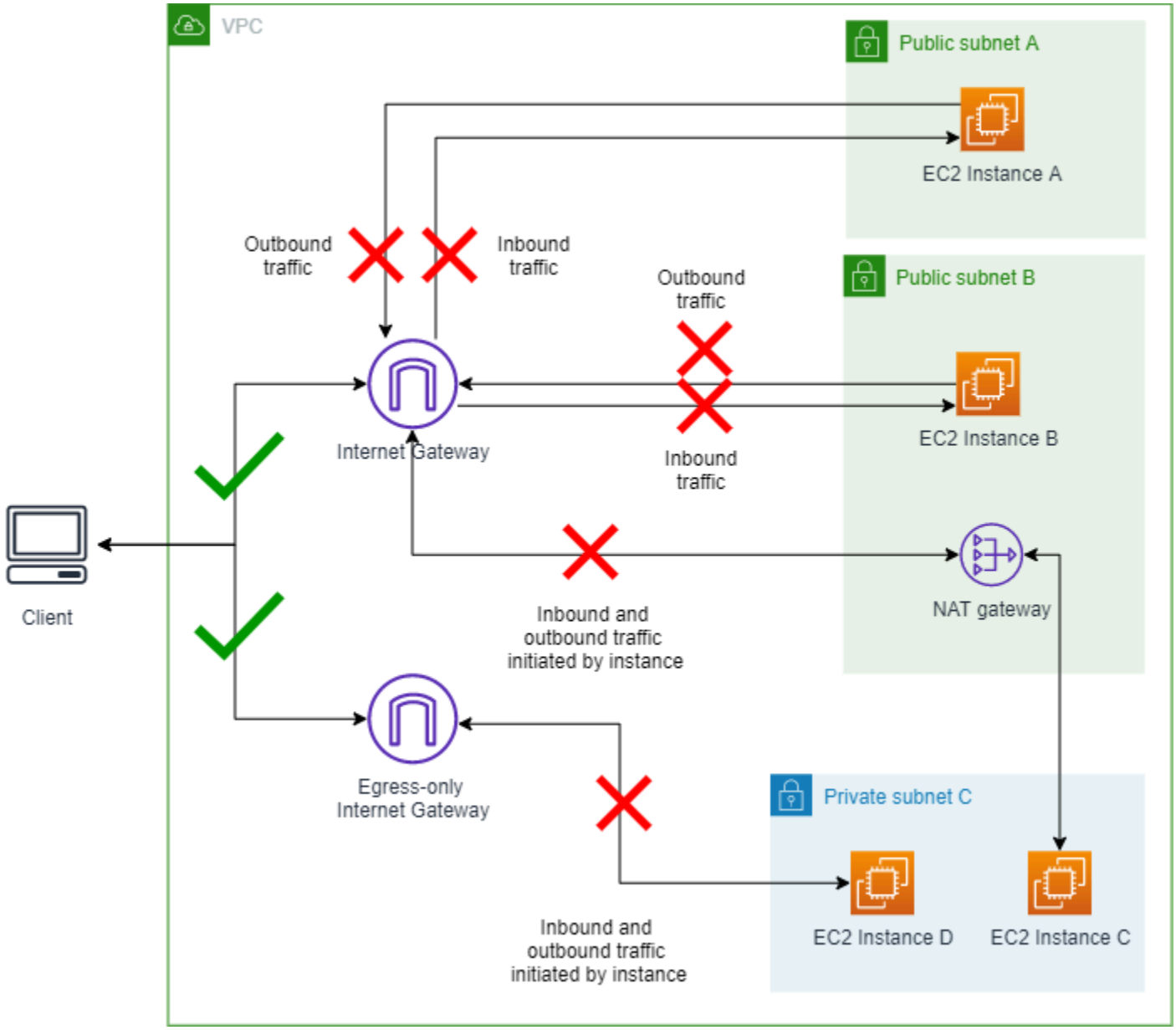
```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:13:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2606:2cc0::374 (2606:2cc0::374)) 56 data bytes
64 bytes from 2606:2cc0::374 (2606:2cc0::374): icmp_seq=1 ttl=58 time=1.21 ms
64 bytes from 2606:2cc0::374 (2606:2cc0::374): icmp_seq=2 ttl=58 time=1.51 ms
```

ping이 성공하며 트래픽이 차단되지 않는지 확인합니다.

시나리오 6 - VPC BPA 모드 수정

이 섹션에서는 VPC BPA 차단 방향을 변경하여 트래픽에 미치는 영향을 알아봅니다. 이 시나리오에서 양방향 모드로 활성화된 VPC BPA는 시나리오 1과 마찬가지로 모든 트래픽을 차단합니다. 제외 항목에 NAT 게이트웨이 또는 송신 전용 인터넷 게이트웨이에 대한 액세스 권한이 없는 경우 트래픽이 차단됩니다.

BPA 양방향 모드가 켜져 있는 VPC와 송신 전용 모드가 켜져 있는 서브넷 A의 다이어그램:



6.1 VPC BPA를 양방향 모드로 변경

이 섹션을 완료하여 VPC BPA 모드를 변경합니다.

AWS Management Console

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 설정을 선택합니다.
3. 퍼블릭 액세스 설정 편집을 선택합니다.
4. 차단 방향을 양방향으로 변경한 다음 변경 사항 저장을 선택합니다.

5. 상태가 켜기로 변경될 때까지 기다립니다. VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

1. VPC BPA 차단 방향 수정:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-
gateway-block-mode block-bidirectional
```

   VPC BPA 설정이 적용되고 상태가 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

2. VPC BPA 상태 보기:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

6.2 인스턴스에 연결

이 섹션을 완료하여 인스턴스에 연결합니다.

AWS Management Console

1. 인스턴스 A 및 B의 퍼블릭 IPv4 주소를 ping합니다. 트래픽이 차단되는지 확인합니다.
2. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 A 및 B에 연결하고 www.amazon.com을 ping합니다. 인스턴스 A 또는 B에서는 인터넷의 퍼블릭 사이트를 ping할 수 없으며 트래픽이 차단되는지 확인합니다.
3. 시나리오 1에서와 같이 EC2 Instance Connect를 사용하여 인스턴스 C 및 D에 연결하고 www.amazon.com을 ping합니다. 인스턴스 C 또는 D에서는 인터넷의 퍼블릭 사이트를 ping할 수 없으며 트래픽이 차단되는지 확인합니다.

AWS CLI

1. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 18.225.8.244
```

   출력:

```
Pinging 18.225.8.244 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

2. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:17:44 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

3. 퍼블릭 IPv4 주소를 사용하여 인스턴스 A를 ping하여 인바운드 트래픽 확인:

```
ping 3.18.106.198
```

출력:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

4. 프라이빗 IPv4 주소를 사용하여 연결하고 아웃바운드 트래픽 확인:

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

5. 인스턴스 C에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:19:45 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
```

```
PING www.amazon.com(2600:9000:25f3:6200:7:49a5:5fd4:b121
 (2600:9000:25f3:6200:7:49a5:5fd4:b121)) 56 data bytes
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

6. 인스턴스 D에 연결합니다. ping할 퍼블릭 IP 주소가 없으므로 EC2 Instance Connect를 사용하여 연결한 다음 인스턴스에서 퍼블릭 IP를 ping하여 아웃바운드 트래픽을 확인합니다.

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

출력:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:20:58 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:b400:7:49a5:5fd4:b121
 (2600:9000:25f3:b400:7:49a5:5fd4:b121)) 56 data bytes
```

ping이 실패하고 트래픽이 차단되는지 확인합니다.

정리

이 섹션에서는 이 고급 예제에서 생성한 모든 리소스를 삭제합니다. 계정에서 생성된 리소스에 대한 과도한 추가 요금을 방지하려면 리소스를 정리하는 것이 중요합니다.

CloudFormation 리소스 삭제

이 섹션을 완료하여 CloudFormation 템플릿으로 생성한 리소스를 삭제합니다.

AWS Management Console

1. CloudFormation에서 https://console.aws.amazon.com/cloudformation/ 콘솔을 엽니다.
2. VPC BPA 스택을 선택합니다.
3. 삭제를 선택합니다.
4. 스택 삭제를 시작한 후 이벤트 탭을 표시하여 진행 상황을 확인하고 스택이 삭제되었는지 확인합니다. 스택을 강제로 삭제해야 스택이 완전히 삭제될 수 있습니다.

AWS CLI

1. CloudFormation 스택을 삭제합니다. 스택을 강제로 삭제해야 스택이 완전히 삭제될 수 있습니다.

```
aws cloudformation delete-stack --stack-name VPC-BPA-stack --region us-east-2
```

2. 진행 상황을 보고 스택이 삭제되었는지 확인합니다.

```
aws cloudformation describe-stack-events --stack-name VPC-BPA-stack --region us-east-2
```

CloudTrail을 사용하여 제외 항목 삭제 추적

이 섹션의 단계에 따라 AWS CloudTrail을 사용하여 제외 항목 삭제를 추적합니다. 제외 항목을 삭제하면 CloudTrail 항목이 나타납니다.

AWS Management Console

https://console.aws.amazon.com/cloudtrailv2/의 AWS CloudTrail 콘솔에서 리소스 유형 > AWS::EC2::VPCBlockPublicAccessExclusion을 조회하면 CloudTrail 이벤트 기록에서 삭제된 제외 항목을 볼 수 있습니다.

AWS CLI

lookup-events 명령을 사용하여 제외 항목 삭제와 관련된 이벤트를 볼 수 있습니다.

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceType,AttributeValue=AWS::EC2::VPCBlockPublicAccessExclusion
```

고급 예제가 완료되었습니다.

VPC에 대한 보안 모범 사례

다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용하세요.

- VPC에 서브넷을 추가하여 애플리케이션을 호스팅하는 경우 여러 가용 영역에 서브넷을 생성합니다. 가용 영역은 AWS 리전에 중복 전원, 네트워킹 및 연결이 있는 하나 이상의 개별 데이터 센터입니다. 여러 가용 영역을 사용하면 프로덕션 애플리케이션의 가용성, 내결함성 및 확장성이 향상됩니다.
- 보안 그룹을 사용하여 서브넷의 EC2 인스턴스에 대한 트래픽을 제어합니다. 자세한 내용은 보안 그룹 단원을 참조하세요.
- 네트워크 ACL을 사용하여 서브넷 수준에서 인바운드 및 아웃바운드 트래픽을 제어합니다. 자세한 내용은 네트워크 액세스 제어 목록으로 서브넷 트래픽 제어 단원을 참조하세요.
- (AWS Identity and Access Management)(IAM) 아이덴티티 페더레이션, 사용자, 역할을 사용하여 VPC의 AWS 리소스에 대한 액세스를 관리합니다. 자세한 내용은 Amazon VPC용 자격 증명 및 액세스 관리 단원을 참조하세요.
- VPC 흐름 로그를 사용하여 VPC, 서브넷 또는 네스워크 인터페이스에서 양쪽에서 이동하는 IP 트래픽을 모니터링합니다. 자세한 내용은 VPC 흐름 로그 단원을 참조하세요.
- Network Access Analyzer를 사용하여 VPC에서 리소스에 대한 의도하지 않은 네트워크 액세스를 식별합니다. 자세한 내용을 알아보려면 Network Access Analyzer Guide(Network Access Analyzer 설명서)를 참조하세요.
- AWS Network Firewall를 사용하여 인바운드 및 아웃바운드 트래픽을 필터링하여 VPC를 모니터링하고 보호합니다. 자세한 정보는 AWS Network Firewall 안내서를 참조하세요.
- Amazon GuardDuty로 AWS 환경 내 계정과 컨테이너, 워크로드, 데이터에 대한 잠재적인 위협을 탐지할 수 있습니다. 기본 위협 탐지에는 Amazon EC2 인스턴스와 관련된 VPC 흐름 로그 모니터링이 포함됩니다. 자세한 내용은 Amazon GuardDuty 사용 설명서의 VPC 흐름 로그를 참조하세요.

VPC 보안 관련된 대한 자주 하는 질문에 대한 답변은 Amazon VPC FAQ의 보안 및 필터링을 참조하세요.

다른 AWS 서비스와 함께 Amazon VPC 사용

Amazon Virtual Private Cloud(VPC)는 클라우드 인프라를 위한 안전하고 사용자 지정 가능한 네트워킹 환경을 제공하는 기본 AWS 서비스입니다. 사용자 소유의 VPC를 생성하고 관리하는 기능 외에도, VPC와 다른 AWS 서비스 간의 연동 기능을 활용하여 요구 사항에 맞는 포괄적인 솔루션을 구축할 수 있습니다.

AWS PrivateLink를 사용하여 VPC를 다양한 AWS 서비스에 연결할 수 있습니다. 이를 통해 VPC와 지원되는 AWS 서비스 또는 온프레미스 애플리케이션 간의 프라이빗 연결이 가능하며, 네트워크 트래픽을 AWS 네트워크 내에 유지하고 퍼블릭 인터넷에 노출을 방지할 수 있습니다. 이는 엄격한 보안 경계 및 규정 준수 요구 사항을 유지하는 데 특히 도움이 됩니다.

VPC의 보안을 더욱 강화하기 위해 AWS Network Firewall을 사용할 수 있습니다. 이 관리형 방화벽 서비스를 사용하면 네트워크 수준의 보안 정책을 정의하고 적용하여 VPC 내의 north-south 및 east-west 트래픽을 모두 필터링할 수 있습니다. Network Firewall을 VPC와 페어링하여 방어 전략을 강화하고 무단 액세스 또는 악의적인 활동으로부터 클라우드 리소스를 보호할 수 있습니다.

또한 Route 53 Resolver DNS 방화벽을 사용하여 VPC 내의 DNS 트래픽을 필터링할 수 있습니다. 이 기능을 사용하면 VPC 리소스가 확인할 수 있는 도메인을 제어하는 사용자 지정 DNS 필터링 규칙을 생성하여 보안 및 규정 준수를 한 층 더 강화할 수 있습니다.

VPC 내부 리소스 또는 VPC에 연결된 리소스 간에 도달 가능성 문제가 발생하는 경우 Reachability Analyzer를 사용할 수 있습니다. Reachability Analyzer는 가상 연결 테스트를 수행하여 자세한 홉별 경로 정보를 제공하고 차단 구성 요소를 식별합니다. 이 문제 해결 도구는 네트워크 연결 문제를 신속하게 식별하고 해결하는 데 도움이 됩니다.

이러한 보완적인 AWS 서비스를 VPC와 통합하면 고유한 비즈니스 및 아키텍처 요구 사항을 충족하는 강력하고 안전하며 복원력이 뛰어난 클라우드 솔루션을 구축할 수 있습니다.

내용

- AWS PrivateLink를 사용하여 서비스에 VPC 연결
- AWS Network Firewall을 사용하여 네트워크 트래픽 필터링
- Route 53 Resolver DNS 방화벽을 사용하여 DNS 트래픽 필터링
- Reachability Analyzer를 사용하여 연결 문제 해결

AWS PrivateLink를 사용하여 서비스에 VPC 연결

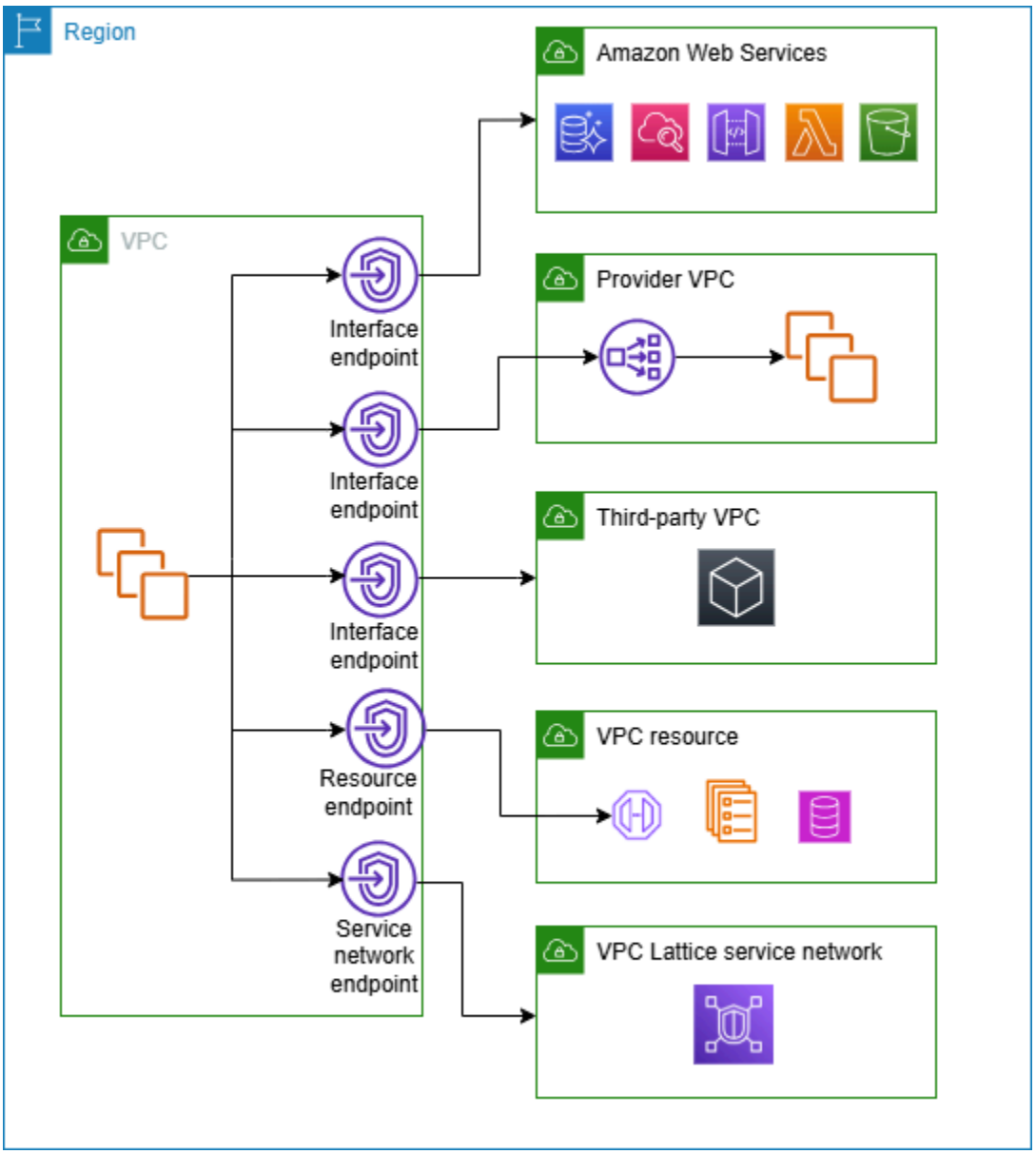
AWS PrivateLink는 가상 프라이빗 클라우드(VPC)와 지원되는 AWS 서비스, 다른 AWS 계정에서 호스팅하는 서비스 및 지원되는 AWS Marketplace 서비스, 지원되는 리소스 간에 프라이빗 연결을 설정합니다. 서비스 또는 리소스와 통신하는 데 인터넷 게이트웨이, NAT 디바이스, Direct Connect 연결 또는 AWS Site-to-Site VPN 연결을 사용하지 않아도 됩니다.

AWS PrivateLink를 사용하려면 서비스 또는 리소스에 액세스해야 하는 서브넷에서 VPC 엔드포인트를 생성합니다. 그러면 지정된 서브넷에 서비스 또는 리소스로 전달되는 트래픽에 대한 진입점 역할을 하는 탄력적 네트워크 인터페이스가 생성됩니다.

또한 AWS PrivateLink로 구동되는 자체 VPC 엔드포인트 서비스를 생성하고 다른 AWS 고객이 해당 서비스에 액세스할 수 있도록 합니다. PrivateLink를 사용하면 프라이빗 API 엔드포인트를 생성하여 조직이 자체 서비스를 다른 AWS 고객에게 안전하게 노출할 수 있습니다. 이를 통해 기업은 내부 역량으로 수익을 창출하고, 협업 생태계를 조성하고, 서비스 액세스 및 소비 방식을 지속적으로 제어할 수 있습니다.

AWS PrivateLink 사용의 주요 이점 중 하나는 인터넷 게이트웨이, NAT 디바이스 또는 VPN 연결과 같은 기존 네트워킹 구성 없이도 안전한 프라이빗 연결을 설정할 수 있다는 것입니다. 이를 통해 네트워크 아키텍처를 단순화하고, 공격 표면을 줄이고, 데이터 트래픽을 AWS 네트워크 내로 제한하여 전반적인 보안을 강화할 수 있습니다.

다음 다이어그램에서는 AWS PrivateLink의 일반적인 사용 사례를 보여줍니다. VPC에는 프라이빗 서브넷에 여러 EC2 인스턴스가 있으며 5개의 VPC 엔드포인트를 통해 리소스에 액세스할 수 있습니다. 인터페이스 VPC 엔드포인트 3개, 리소스 VPC 엔드포인트 1개, 서비스 네트워크 VPC 엔드포인트 1개가 있습니다.



자세한 내용은 AWS PrivateLink 단원을 참조하세요.

AWS Network Firewall을 사용하여 네트워크 트래픽 필터링

AWS Network Firewall을 사용하면 VPC 경계에서 네트워크 트래픽을 필터링할 수 있습니다. Network Firewall은 상태를 저장하는 관리형 네트워크 방화벽이자, 침입 탐지 및 방지 서비스입니다. 자세한 내용은 AWS Network Firewall 개발자 안내서를 참조하십시오.

다음 AWS 리소스를 사용하여 Network Firewall을 구현합니다.

Network Firewall 리소스	설명
방화벽	방화벽은 방화벽 정책의 네트워크 트래픽 필터링 동작을 보호하려는 VPC에 연결합니다. 방화벽 구성에는 방화벽 엔드포인트가 배치되는 가용 영역 및 서브넷에 대한 사양이 포함됩니다. 또한 AWS 방화벽 리소스의 방화벽 로깅 구성 및 태깅과 같은 상위 수준 설정을 정의합니다. 자세한 내용은 AWS Network Firewall의 방화벽을 참조하세요.
방화벽 정책	방화벽 정책은 방화벽에 대한 모니터링 및 보호 동작을 정의합니다. 동작의 세부 정보는 정책에 추가하는 규칙 그룹과 일부 정책 기본 설정에 정의됩니다. 방화벽 정책을 사용하려면 하나 이상의 방화벽과 연결합니다. 자세한 내용은 AWS Network Firewall의 방화벽 정책을 참조하세요.
규칙 그룹	규칙 그룹은 네트워크 트래픽을 검사하고 처리하기 위한 재사용 가능한 기준 세트입니다. 정책 구성의 일부로 방화벽 정책에 하나 이상의 규칙 그룹을 추가합니다. 상태 비저장 규칙 그룹을 정의하여 각 네트워크 패킷을 격리 상태에서 검사할 수 있습니다. 상태 비저장 규칙 그룹은 Amazon VPC 네트워크 액세스 제어 목록(ACL)과 동작 및 사용법이 유사합니다. 상태 저장 규칙 그룹을 정의하여 트래픽 흐름의 컨텍스트에서 패킷을 검사할 수도 있습니다. 상태 저장 규칙 그룹은 Amazon VPC 보안 그룹과 동작 및 사용법이 유사합니다. 자세한 내용은 AWS Network Firewall의 규칙 그룹을 참조하세요.

AWS Firewall Manager를 사용하여 AWS Organizations의 계정 및 애플리케이션에서 Network Firewall 리소스를 중앙에서 구성하고 관리할 수도 있습니다. Firewall Manager에서 단일 계정을 사용하여 여러 계정의 방화벽을 관리할 수 있습니다. 자세한 내용은 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 안내서의 AWS Firewall Manager 섹션을 참조하세요.

Route 53 Resolver DNS 방화벽을 사용하여 DNS 트래픽 필터링

DNS 방화벽을 사용하여 VPC와 연결하는 규칙 그룹에 도메인 이름 필터링 규칙을 정의합니다. 허용하거나 차단할 도메인 이름 목록을 지정하고 차단하는 DNS 쿼리에 대한 응답을 사용자 지정할 수 있습니다. 자세한 내용은 Route 53 Resolver DNS Firewall 설명서를 참조하세요.

다음 AWS 리소스를 사용하여 DNS 방화벽을 구현합니다.

DNS 방화벽 리소스	설명
DNS 방화벽 규칙 그룹	DNS 방화벽 규칙 그룹은 DNS 쿼리를 필터링하기 위한 DNS 방화벽 규칙의 명명된 재사용 가능한 컬렉션입니다. 규칙 그룹을 필터링 규칙으로 채운 후, 규칙 그룹을 Amazon VPC의 하나 이상의 VPC와 연결합니다. 규칙 그룹을 VPC와 연결하면 VPC에 대해 DNS 방화벽 필터링을 활성화합니다. 그런 다음 Resolver가 규칙 그룹이 연결된 VPC에 대한 DNS 쿼리를 수신하면 Resolver는 필터링을 위해 쿼리를 DNS 방화벽으로 전달합니다. 규칙 그룹 내의 각 규칙은 하나의 도메인 목록을 지정하고 도메인이 목록의 도메인 사양과 일치하는 DNS 쿼리에 대해 수행할 작업을 지정합니다. 일치하는 쿼리에 대해 허용, 차단 또는 경고를 할 수 있습니다. 차단된 쿼리에 대한 사용자 지정 응답을 정의할 수도 있습니다. 자세한 내용은 Route 53 Resolver DNS Firewall의 규칙 그룹 및 규칙을 참조하세요.
도메인 목록	도메인 목록은 규칙 그룹 내부의 DNS 방화벽 규칙에서 사용하는 재사용 가능한 도메인 사양 집합입니다. 자세한 내용은 Route 53 Resolver DNS Firewall의 도메인 목록을 참조하세요.

AWS Firewall Manager를 사용하여 AWS Organizations의 계정 및 조직에서 DNS 방화벽 리소스를 중앙에서 구성하고 관리할 수도 있습니다. Firewall Manager에서 단일 계정을 사용하여 여러 계정의 방화벽을 관리할 수 있습니다. 자세한 내용은 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 안내서의 AWS Firewall Manager 섹션을 참조하세요.

Reachability Analyzer를 사용하여 연결 문제 해결

Reachability Analyzer는 정적 구성 분석 도구입니다. Reachability Analyzer를 사용하여 VPC의 두 리소스 간 네트워크 연결성을 분석하고 디버깅할 수 있습니다. Reachability Analyzer에서는 연결할 수 있는 경우 이러한 리소스 간 가상 경로에 대한 홉별 세부 정보가 생성되고, 그렇지 않다면 차단 구성 요소가 식별됩니다.

Reachability Analyzer를 사용하여 다음과 같은 리소스 간 연결성을 분석할 수 있습니다.

- 인스턴스
- 인터넷 게이트웨이
- 네트워크 인터페이스
- 전송 게이트웨이
- Transit Gateway Attachment
- VPC 엔드포인트 서비스
- VPC 엔드포인트
- VPC 피어링 연결
- VPN 게이트웨이

자세한 내용은 Reachability Analyzer 사용 설명서를 참조하십시오.

VPC 예시

Amazon Virtual Private Cloud(VPC)는 AWS 에코시스템 내의 기본 구성 요소로서, 이를 통해 특정 요구 사항에 맞게 조정된 격리된 가상 네트워크를 프로비저닝할 수 있습니다. 자체 VPC를 생성하고 관리하면 네트워킹 환경을 완전히 제어할 수 있습니다. 예를 들어, IP 주소 범위, 서브넷, 라우팅 테이블 및 연결 옵션을 정의할 수 있습니다.

이 섹션에는 각각 다른 요구 사항을 해결하도록 설계된 세 가지 Virtual Private Cloud(VPC) 구성 예시가 있습니다.

- 테스트 환경용 VPC: 이 구성에서는 개발 또는 테스트 환경으로 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다.
- 웹 및 데이터베이스 서버용 VPC: 이 구성에서는 프로덕션 환경에서 복원력이 뛰어난 아키텍처에 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다.
- 프라이빗 서브넷 및 NAT에 서버가 있는 VPC: 이 고급 구성에서는 모든 EC2 인스턴스가 프라이빗 서브넷 내에서 프로비저닝되며, NAT 게이트웨이를 통해 안전한 아웃바운드 인터넷 액세스가 가능합니다. 이는 리소스에 대한 직접 인터넷 연결을 제한하면서도 필요한 아웃바운드 통신을 활성화해야 하는 예시입니다.

이러한 VPC 구성에 대한 예시를 제공하여 클라우드 네트워킹 환경을 설계할 때 사용할 수 있는 유연성과 사용자 지정 옵션을 설명하고자 합니다. 애플리케이션의 아키텍처, 보안 요구 사항 및 전반적인 비즈니스 목표에 따라 특정 VPC 설정을 선택해야 합니다. VPC 인프라를 신중하게 계획하면 클라우드 기반 워크로드의 성장과 발전을 지원하는 강력하고 확장 가능하며 안전한 가상 네트워크를 구축하는 데 도움이 될 수 있습니다.

예시

- 예시: 테스트 환경을 위한 VPC
- 예시: 웹 및 데이터베이스 서버용 VPC
- 예시: 프라이빗 서브넷과 NAT에 서버가 있는 VPC

관련 예시

- VPC를 서로 연결하려면 Amazon VPC 피어링 설명서의 VPC 피어링 구성을 참조하세요.
- VPC를 자체 네트워크에 연결하려면 AWS Site-to-Site VPN 사용 설명서의 Site-to-Site VPN 시나리오를 참조하세요.

- VPC를 서로 연결하거나 자체 네트워크에 연결하려면 Amazon VPC Transit Gateway의 전송 게이트웨이 시나리오 예를 참조하세요.

추가 리소스

- 복원력 패턴 및 장단점 이해(AWS 아키텍처 블로그)
- 네트워크 토폴로지 계획(AWS Well-Architected 프레임워크)
- Amazon Virtual Private Cloud 연결 옵션(AWS 백서)

예시: 테스트 환경을 위한 VPC

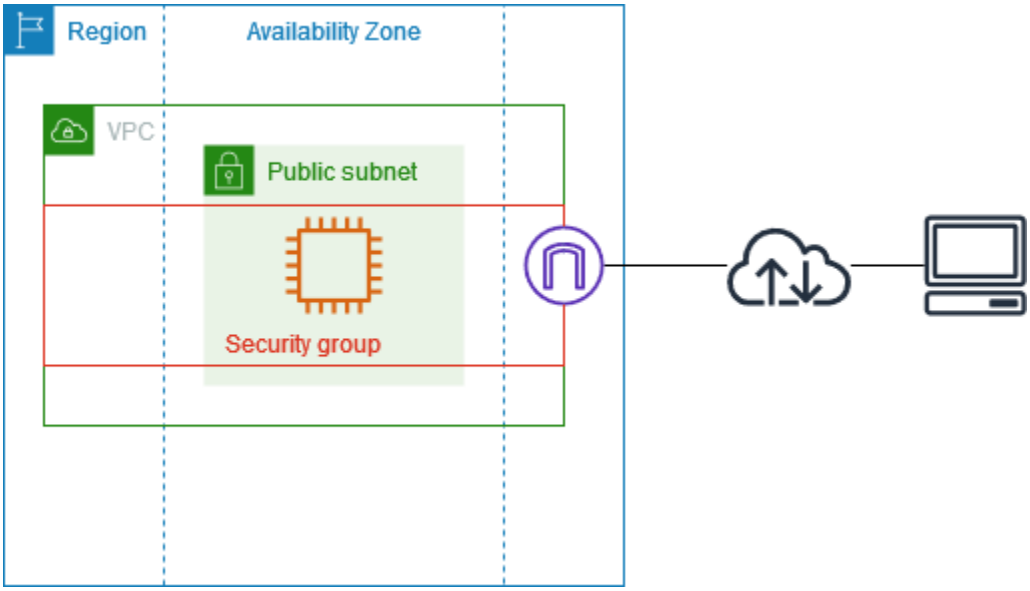
이 예시에서는 개발 또는 테스트 환경으로 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다. 이 VPC는 프로덕션에서 사용하기 위한 것이 아니므로 여러 가용 영역에 서버를 배포할 필요가 없습니다. 비용과 복잡성을 낮게 유지하기 위해 단일 가용 영역에 서버를 배포할 수 있습니다.

내용

- 개요
- 1. VPC 생성
- 2. 애플리케이션 배포
- 3. 구성 테스트
- 4. 정리

개요

다음 다이어그램에서는 이 예시에 포함된 리소스의 개요를 제공합니다. VPC는 단일 가용 영역과 인터넷 게이트웨이에 퍼블릭 서브넷이 있습니다. 서버는 퍼블릭 서브넷에서 실행되는 EC2 인스턴스입니다. 인스턴스의 보안 그룹은 사용자 컴퓨터의 SSH 트래픽과 개발 또는 테스트 활동에 특별히 필요한 기타 트래픽을 허용합니다.



라우팅

Amazon VPC 콘솔을 사용하여 이 VPC를 생성하면 로컬 경로와 인터넷 게이트웨이에 대한 경로가 있는 퍼블릭 서브넷에 대한 라우팅 테이블이 생성됩니다. 다음은 IPv4와 IPv6 모두에 대한 경로가 있는 라우팅 테이블 예시입니다. 이중 스택 서브넷 대신 IPv4 전용 서브넷을 생성하는 경우 라우팅 테이블에는 IPv4 경로만 있습니다.

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

이 예시 구성에서는 애플리케이션에 필요한 트래픽을 허용하는 인스턴스에 대한 보안 그룹을 생성해야 합니다. 예를 들어 컴퓨터의 SSH 트래픽이나 네트워크의 HTTP 트래픽을 허용하는 규칙을 추가해야 할 수 있습니다.

다음은 IPv4와 IPv6 모두에 대한 규칙이 포함된 보안 그룹의 인바운드 규칙의 예시입니다. 이중 스택 서브넷 대신 IPv4 전용 서브넷을 생성하는 경우 IPv4에 대한 규칙만 필요합니다.

소스	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	모든 IPv4 주소에서 이루어지는 인바운드 HTTP 액세스 허용
::/0	TCP	80	모든 IPv6 주소에서 이루어지는 인바운드 HTTP 액세스 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
##### ### IPv4 ## ##	TCP	22	(선택 사항) 네트워크에서 IPv4 IP 주소의 인바운드 SSH 액세스 허용
##### IPv6 ## ##	TCP	22	(선택 사항) 네트워크에서 IPv6 IP 주소의 인바운드 SSH 액세스 허용
##### ### IPv4 ## ##	TCP	3389	(선택 사항) 네트워크에서 IPv4 IP 주소의 인바운드 RDP 액세스 허용
##### IPv6 ## ##	TCP	3389	(선택 사항) 네트워크에서 IPv6 IP 주소의 인바운드 RDP 액세스 허용

1. VPC 생성

다음 절차에 따라 1개의 가용 영역에 퍼블릭 서브넷이 있는 VPC를 생성하세요. 이 구성은 개발 또는 테스트 환경에 적합합니다.

VPC를 생성하려면

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

2. 대시보드에서 VPC 생성을 선택합니다.
3. 생성할 리소스에서 VPC 등을 선택합니다.
4. VPC 구성

    a. Name tag auto-generation(이름 태그 자동 생성)에 VPC의 이름을 입력합니다.

    b. IPv4 CIDR 블록에 애플리케이션 또는 네트워크에 필요한 CIDR 블록을 입력하거나 기본 사항을 유지할 수 있습니다. 자세한 내용은 the section called “VPC CIDR 블록” 섹션을 참조하세요.

    c. (선택 사항) 애플리케이션이 IPv6 주소를 사용하여 통신하는 경우 IPv6 CIDR 블록, 즉 Amazon에서 제공한 IPv6 CIDR 블록을 선택합니다.

5. 서브넷 구성

    a. 가용 영역 수에서 1을 선택합니다. 기본 가용 영역을 유지하거나, AZ 사용자 지정을 확장하고 가용 영역을 선택할 수 있습니다.

    b. 퍼블릭 서브넷 수는 1을 선택합니다.

    c. 프라이빗 서브넷 수는 0을 선택합니다.

    d. 퍼블릭 서브넷에 대한 기본 CIDR 블록을 유지하거나 서브넷 CIDR 블록 사용자 지정을 확장하고 CIDR 블록을 입력할 수 있습니다. 자세한 내용은 the section called “서브넷 CIDR 블록” 섹션을 참조하세요.

6. NAT 게이트웨이에서 기본값인 없음을 유지합니다.
7. VPC 엔드포인트는 없음을 선택합니다. S3용 게이트웨이 VPC 엔드포인트는 프라이빗 서브넷에서 Amazon S3에 액세스하는 데만 사용됩니다.
8. DNS 옵션에서 두 옵션을 모두 선택된 상태로 둡니다. 결과적으로 인스턴스는 퍼블릭 IP 주소에 해당하는 퍼블릭 DNS 호스트 이름을 수신하게 됩니다.
9. VPC 생성을 선택합니다.

2. 애플리케이션 배포

EC2 인스턴스를 배포하는 다양한 방법이 있습니다. 예제:

- Amazon EC2 인스턴스 시작 마법사
- Amazon EC2 Auto Scaling
- AWS CloudFormation
- Amazon Elastic Container Service(Amazon ECS)

EC2 인스턴스를 배포한 후 인스턴스에 연결하고, 애플리케이션에 필요한 소프트웨어를 설치한 다음, 나중에 사용할 이미지를 생성할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 AMI 생성을 참조하세요. 또는 EC2 Image Builder를 사용하여 Amazon Machine Image(AMI)를 생성하고 관리할 수 있습니다.

3. 구성 테스트

애플리케이션 배포를 완료한 후 테스트할 수 있습니다. EC2 인스턴스에 연결하거나 애플리케이션이 예상한 트래픽을 전송하거나 수신할 수 없는 경우 Reachability Analyzer를 사용하여 문제를 해결할 수 있습니다. 예를 들어, Reachability Analyzer는 라우팅 테이블 또는 보안 그룹의 구성 문제를 식별할 수 있습니다. 자세한 내용은 Reachability Analyzer 사용 설명서를 참조하세요.

4. 정리

이 구성을 마치면 이를 삭제할 수 있습니다. VPC를 삭제하려면 먼저 인스턴스를 종료해야 합니다. 자세한 내용은 the section called “VPC 삭제” 섹션을 참조하세요.

예시: 웹 및 데이터베이스 서버용 VPC

이 예시에서는 프로덕션 환경에서 2계층 아키텍처에 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다. 복원력 향상을 위해 2개의 가용 영역에 서버를 배포합니다.

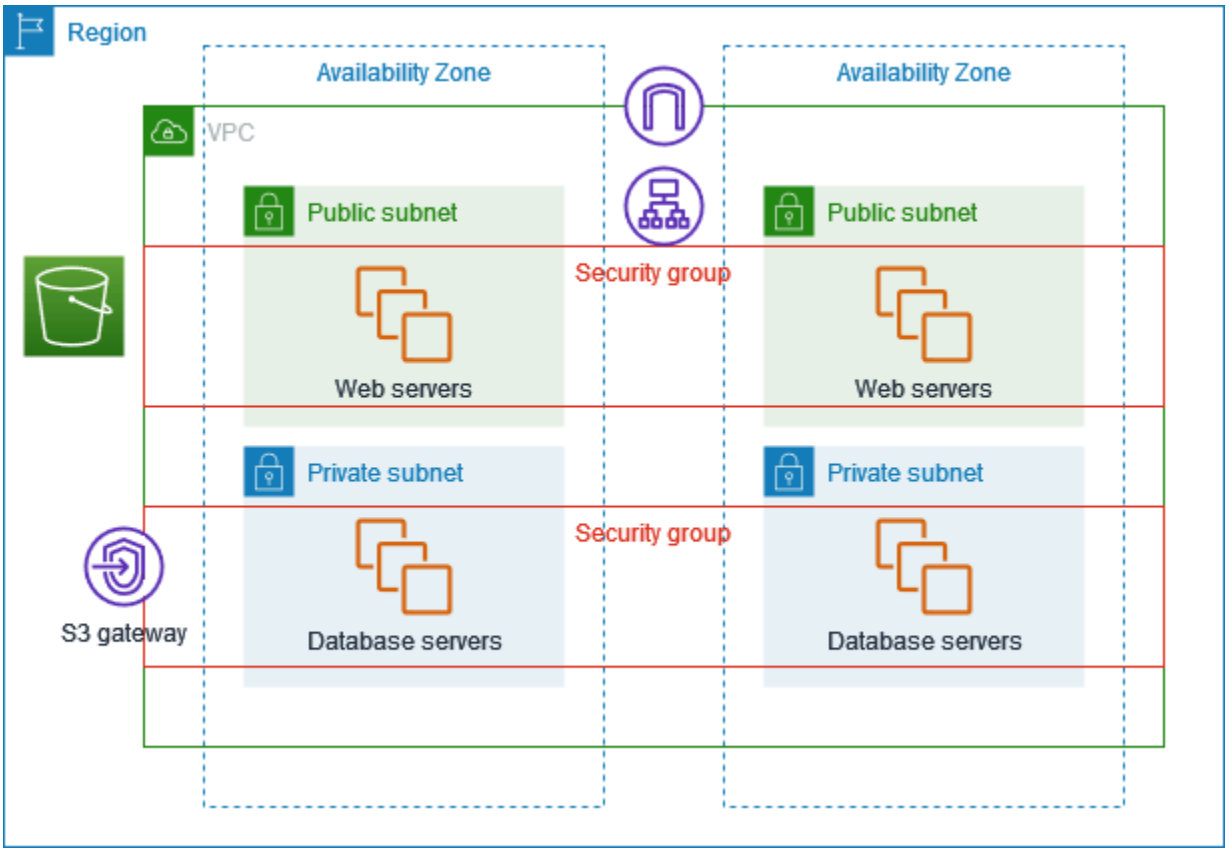
내용

- 개요
- 1. VPC 생성
- 2. 애플리케이션 배포
- 3. 구성 테스트
- 4. 정리

개요

다음 다이어그램에서는 이 예시에 포함된 리소스의 개요를 제공합니다. VPC에는 2개의 가용 영역에 퍼블릭 서브넷과 프라이빗 서브넷이 있습니다. 웹 서버는 퍼블릭 서브넷에서 실행되며 로드 밸런서를 통해 클라이언트로부터 트래픽을 수신합니다. 웹 서버의 보안 그룹은 로드 밸런서의 트래픽을 허용합니다. 데이터베이스 서버는 프라이빗 서브넷에서 실행되며 웹 서버로부터 트래픽을 수신합니다. 데이

터베이스 서버의 보안 그룹은 웹 서버의 트래픽을 허용합니다. 데이터베이스 서버는 게이트웨이 VPC 엔드포인트를 사용하여 Amazon S3에 연결할 수 있습니다.



라우팅

Amazon VPC 콘솔을 사용하여 이 VPC를 생성하면 로컬 경로와 인터넷 게이트웨이에 대한 경로가 있는 퍼블릭 서브넷의 라우팅 테이블과 로컬 경로 및 게이트웨이 VPC 엔드포인트에 대한 경로가 있는 각 프라이빗 서브넷의 경로 테이블이 생성됩니다.

다음은 IPv4와 IPv6 모두에 대한 경로가 있는 퍼블릭 서브넷의 라우팅 테이블 예시입니다. 이중 스택 서브넷 대신 IPv4 전용 서브넷을 생성하는 경우 라우팅 테이블에는 IPv4 경로만 있습니다.

대상 주소	대상
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	*igw-id*

대상 주소	대상
::/0	*igw-id*

다음은 IPv4와 IPv6 모두에 대한 로컬 경로가 있는 프라이빗 서브넷의 라우팅 테이블 예시입니다. IPv4 전용 서브넷을 생성한 경우 라우팅 테이블에는 IPv4 경로만 있습니다. 마지막 경로는 Amazon S3로 향하는 트래픽을 게이트웨이 VPC 엔드포인트로 전송합니다.

대상 주소	대상
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
s3-prefix-list-id	*s3-gateway-id*

보안

이 예시 구성에서는 로드 밸런서용 보안 그룹, 웹 서버용 보안 그룹 및 데이터베이스 서버용 보안 그룹을 생성합니다.

로드 밸런서

Application Load Balancer 또는 Network Load Balancer의 보안 그룹에서는 로드 밸런서 리스너 포트의 클라이언트에서 제공되는 인바운드 트래픽이 허용되어야 합니다. 인터넷상의 위치와 관계 없이 모든 트래픽을 수락하려면 소스를 0.0.0.0/0으로 지정하세요. 로드 밸런서 보안 그룹에서는 로드 밸런서에서 출발하여 인스턴스 리스너 포트 및 상태 확인 포트의 대상 인스턴스로 향하는 아웃바운드 트래픽도 허용되어야 합니다.

웹 서버

다음 보안 그룹 규칙을 사용하면 웹 서버가 로드 밸런서로부터 HTTP 및 HTTPS 트래픽을 수신할 수 있습니다. 웹 서버가 네트워크에서 SSH 또는 RDP 트래픽을 수신하도록 허용할 수도 있습니다. 웹 서버는 데이터베이스 서버에 SQL 또는 MySQL 트래픽을 전송할 수 있습니다.

소스	프로토콜	포트 범위	설명
## ### ## ### ID	TCP	80	로드 밸런서에서 접근하는 인바운드 HTTP 액세스를 허용
## ### ## ### ID	TCP	443	로드 밸런서에서 접근하는 인바운드 HTTPS 액세스를 허용
##### ### IPv4 ## ##	TCP	22	(선택 사항) 네트워크에서 IPv4 IP 주소의 인바운드 SSH 액세스 허용
##### IPv6 ## ##	TCP	22	(선택 사항) 네트워크에서 IPv6 IP 주소의 인바운드 SSH 액세스 허용
##### ### IPv4 ## ##	TCP	3389	(선택 사항) 네트워크에서 IPv4 IP 주소의 인바운드 RDP 액세스 허용
##### IPv6 ## ##	TCP	3389	(선택 사항) 네트워크에서 IPv6 IP 주소의 인바운드 RDP 액세스 허용

Destination	프로토콜	포트 범위	설명
Microsoft SQL Server# #### ##### ## ## ID	TCP	1433	데이터베이스 서버에 대한 아웃바운드 Microsoft SQL Server 액세스 허용
MySQL# #### ##### ## ## ID	TCP	3306	데이터베이스 서버에 대한 아웃바운드 MySQL 액세스 허용

데이터베이스 서버

다음 보안 그룹 규칙은 데이터베이스 서버가 웹 서버로부터 읽기 및 쓰기 요청을 수신하도록 허용합니다.

소스	프로토콜	포트 범위	설명
# ## ## ### ID	TCP	1433	웹 서버의 인바운드 Microsoft SQL Server 액세스 허용
# ## ## ### ID	TCP	3306	웹 서버의 인바운드 MySQL Server 액세스 허용

Destination	프로토콜	포트 범위	설명
0.0.0.0/0	TCP	80	IPv4를 통해 인터넷에 대한 아웃바운드 HTTP 액세스 허용
0.0.0.0/0	TCP	443	IPv4를 통해 인터넷에 대한 아웃바운드 HTTPS 액세스 허용

Amazon RDS DB 인스턴스의 보안 그룹에 대한 자세한 내용은 Amazon RDS 사용 설명서의 보안 그룹을 통한 액세스 제어를 참조하세요.

1. VPC 생성

다음 절차에 따라 2개의 가용 영역에 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC를 생성하세요.

VPC를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 VPC 생성을 선택합니다.
3. 생성할 리소스에서 VPC 등을 선택합니다.
4. VPC 구성:
   a. 이름 태그 자동 생성을 선택한 상태로 유지하여 VPC 리소스에 이름 태그를 생성하거나 선택을 취소하여 VPC 리소스에 고유한 이름 태그를 제공합니다.
   b. IPv4 CIDR 블록에 애플리케이션 또는 네트워크에 필요한 CIDR 블록을 입력하거나 기본 사항을 유지할 수 있습니다. 자세한 내용은 the section called “VPC CIDR 블록” 섹션을 참조하세요.

c. (선택 사항) 애플리케이션이 IPv6 주소를 사용하여 통신하는 경우 IPv6 CIDR 블록, 즉 Amazon에서 제공한 IPv6 CIDR 블록을 선택합니다.

d. 테넌시 옵션을 선택합니다. 이 옵션은 VPC로 시작하는 EC2 인스턴스가 다른 AWS 계정과 공유되는 하드웨어에서 실행되는지 아니면 사용자 전용 하드웨어에서 실행되는지를 정의합니다. VPC의 테넌시를 `Default`로 선택하면 이 VPC로 시작된 EC2 인스턴스에서는 인스턴스를 시작할 때 지정된 테넌시 속성을 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 정의된 파라미터를 사용하여 인스턴스 시작을 참조하세요. VPC의 테넌시를 `Dedicated`로 선택하면 인스턴스는 항상 전용 하드웨어에서 전용 인스턴스로 실행됩니다.

5. 서브넷 구성:

a. 복원 향상을 위해 2개의 가용 영역에서 인스턴스를 시작할 수 있도록 가용 영역 수에서 2를 선택합니다.

b. 퍼블릭 서브넷 수는 2를 선택합니다.

c. 프라이빗 서브넷 수는 2를 선택합니다.

d. 서브넷에 대한 기본 CIDR 블록을 유지하거나 서브넷 CIDR 블록 사용자 지정을 확장하고 CIDR 블록을 입력할 수 있습니다. 자세한 내용은 the section called “서브넷 CIDR 블록” 섹션을 참조하세요.

6. NAT 게이트웨이에서 기본값인 없음을 유지합니다.

7. VPC 엔드포인트에서 기본값인 S3 게이트웨이를 유지합니다. S3 버킷에 액세스하지 않으면 효과가 없지만 이 VPC 엔드포인트를 활성화하는 데 비용이 들지 않습니다.

8. DNS 옵션에서 두 옵션을 모두 선택된 상태로 둡니다. 결과적으로 웹 서버는 퍼블릭 IP 주소에 해당하는 퍼블릭 DNS 호스트 이름을 수신하게 됩니다.

9. VPC 생성을 선택합니다.

2. 애플리케이션 배포

개발 또는 테스트 환경에서 웹 서버와 데이터베이스 서버 테스트를 완료하고 프로덕션 환경에서 애플리케이션을 배포하는 데 사용할 스크립트 또는 이미지를 생성하는 것이 가장 좋습니다.

웹 서버에 EC2 인스턴스를 사용할 수 있습니다. EC2 인스턴스를 배포하는 다양한 방법이 있습니다. 예제:

- Amazon EC2 인스턴스 시작 마법사
- AWS CloudFormation
- Amazon Elastic Container Service(Amazon ECS)

가용성 향상을 위해 Amazon EC2 Auto Scaling을 사용하여 여러 가용 영역에 서버를 배포하고 애플리케이션에 필요한 최소 서버 용량을 유지할 수 있습니다.

Elastic Load Balancing을 사용하여 서버 전체에 트래픽을 균일하게 배포할 수 있습니다. Auto Scaling에 로드 밸런서를 연결할 수 있습니다.

데이터베이스 서버용 EC2 인스턴스를 사용하거나 목적별 데이터베이스 유형 중 하나를 사용할 수 있습니다. 자세한 내용은 AWS의 데이터베이스: 선택 방법을 참조하세요.

3. 구성 테스트

애플리케이션 배포를 완료한 후 테스트할 수 있습니다. 애플리케이션이 예상한 트래픽을 전송하거나 수신할 수 없는 경우 Reachability Analyzer를 사용하여 문제를 해결할 수 있습니다. 예를 들어, Reachability Analyzer는 라우팅 테이블 또는 보안 그룹의 구성 문제를 식별할 수 있습니다. 자세한 내용은 Reachability Analyzer 사용 설명서를 참조하세요.

4. 정리

이 구성을 마치면 이를 삭제할 수 있습니다. VPC를 삭제하려면 먼저 인스턴스를 종료하고 로드 밸런서를 삭제해야 합니다. 자세한 내용은 the section called “VPC 삭제” 섹션을 참조하세요.

예시: 프라이빗 서브넷과 NAT에 서버가 있는 VPC

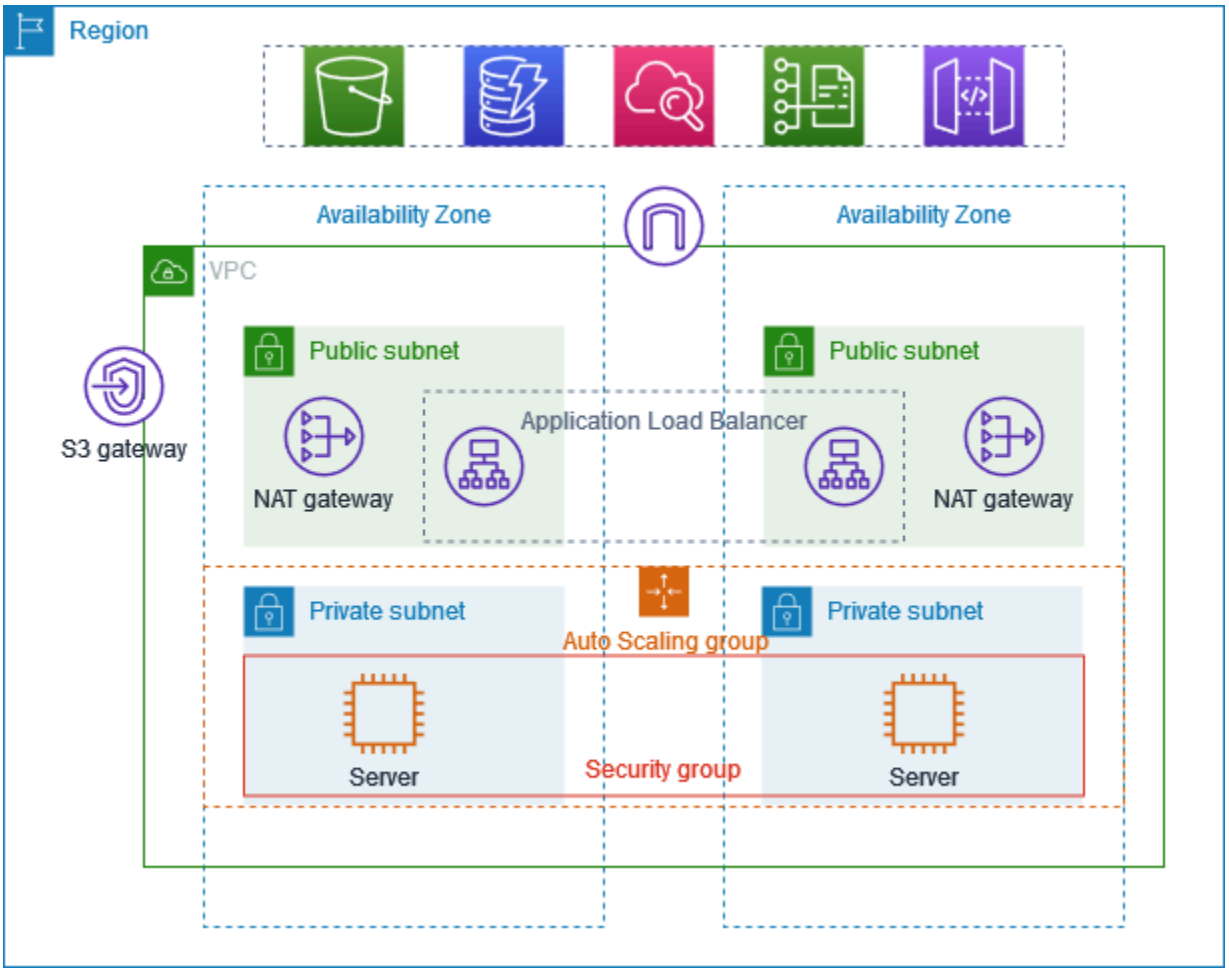
이 예시에서는 프로덕션 환경에서 서버에 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다. 복원력 향상을 위해 Auto Scaling과 Application Load Balancer를 사용하여 2개의 가용 영역에 서버를 배포합니다. 보안 강화를 위해 프라이빗 서브넷에 서버를 배포합니다. 서버는 로드 밸런서를 통해 요청을 수신합니다. 서버는 NAT 게이트웨이를 사용하여 인터넷에 연결할 수 있습니다. 복원력 향상을 위해 두 가용 영역 모두에 NAT 게이트웨이를 배포합니다.

내용

- 개요
- 1. VPC 생성
- 2. 애플리케이션 배포
- 3. 구성 테스트
- 4. 정리

개요

다음 다이어그램에서는 이 예시에 포함된 리소스의 개요를 제공합니다. VPC에는 2개의 가용 영역에 퍼블릭 서브넷과 프라이빗 서브넷이 있습니다. 각 퍼블릭 서브넷에는 NAT 게이트웨이와 로드 밸런서 노드가 있습니다. 서버는 프라이빗 서브넷에서 실행되고, Auto Scaling을 사용하여 시작 및 종료되고, 로드 밸런서에서 트래픽을 수신합니다. 서버는 NAT 게이트웨이를 사용하여 인터넷에 연결할 수 있습니다. 서버는 게이트웨이 VPC 엔드포인트를 사용하여 Amazon S3에 연결할 수 있습니다.



라우팅

Amazon VPC 콘솔을 사용하여 이 VPC를 생성하면 로컬 경로와 인터넷 게이트웨이에 대한 경로가 있는 퍼블릭 서브넷에 대한 라우팅 테이블이 생성됩니다. 또한 로컬 경로와 NAT 게이트웨이, 송신 전용 인터넷 게이트웨이 및 게이트웨이 VPC 엔드포인트에 대한 경로가 있는 프라이빗 서브넷의 라우팅 테이블이 생성됩니다.

다음은 IPv4와 IPv6 모두에 대한 경로가 있는 퍼블릭 서브넷의 라우팅 테이블 예시입니다. 이중 스택 서브넷 대신 IPv4 전용 서브넷을 생성하는 경우 라우팅 테이블에는 IPv4 경로만 포함됩니다.

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*

다음은 IPv4와 IPv6 모두에 대한 경로가 있는 프라이빗 서브넷 중 하나의 라우팅 테이블 예시입니다. IPv4 전용 서브넷을 생성한 경우 라우팅 테이블에는 IPv4 경로만 포함됩니다. 마지막 경로는 Amazon S3로 향하는 트래픽을 게이트웨이 VPC 엔드포인트로 전송합니다.

대상 주소	대상
10.0.0.0/16	로컬
2001:db8:1234:1a00::/56	로컬
0.0.0.0/0	*nat-gateway-id*
::/0	*eigw-id*
s3-prefix-list-id	*s3-gateway-id*

보안

다음은 서버와 연결하는 보안 그룹에 대해 생성할 수 있는 규칙의 예시입니다. 보안 그룹은 리스너 포트 및 프로토콜을 통한 로드 밸런서의 트래픽을 허용해야 합니다. 또한 상태 확인 트래픽을 허용해야 합니다.

소스	프로토콜	포트 범위	설명
## ### ## ### ID	*### ####*	*### ##*	리스너 포트에서 로드 밸런서의 인바운드 트래픽 허용

소스	프로토콜	포트 범위	설명
## ### ## ### ID	*## ## ####*	*## ## ##*	로드 밸런서의 인바운드 상태 확인 트래픽 허용

1. VPC 생성

다음 절차에 따라 2개의 가용 영역에 퍼블릭 서브넷과 프라이빗 서브넷이 있고 각 가용 영역에 NAT 게이트웨이가 있는 VPC를 생성하세요.

VPC를 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 대시보드에서 VPC 생성을 선택합니다.
3. 생성할 리소스에서 VPC 등을 선택합니다.
4. VPC 구성
   a. Name tag auto-generation(이름 태그 자동 생성)에 VPC의 이름을 입력합니다.
   b. IPv4 CIDR 블록에 애플리케이션 또는 네트워크에 필요한 CIDR 블록을 입력하거나 기본 사항을 유지할 수 있습니다.
   c. 애플리케이션이 IPv6 주소를 사용하여 통신하는 경우 IPv6 CIDR 블록, 즉 Amazon에서 제공한 IPv6 CIDR 블록을 선택합니다.
5. 서브넷 구성
   a. 복원 향상을 위해 여러 가용 영역에서 인스턴스를 시작할 수 있도록 가용 영역 수에서 2를 선택합니다.
   b. 퍼블릭 서브넷 수는 2를 선택합니다.
   c. 프라이빗 서브넷 수는 2를 선택합니다.
   d. 퍼블릭 서브넷에 대한 기본 CIDR 블록을 유지하거나 서브넷 CIDR 블록 사용자 지정을 확장하고 CIDR 블록을 입력할 수 있습니다. 자세한 내용은 the section called “서브넷 CIDR 블록” 섹션을 참조하세요.
6. NAT 게이트웨이에서 복원력 향상을 위해 AZ당 1개를 선택합니다.
7. 애플리케이션이 IPv6 주소를 사용하여 통신하는 경우 송신 전용 인터넷 게이트웨이에서 예를 선택합니다.

8. VPC 엔드포인트에서 인스턴스가 S3 버킷에 액세스해야 하는 경우 S3 게이트웨이를 기본값으로 유지합니다. 그렇지 않으면 프라이빗 서브넷의 인스턴스가 Amazon S3에 액세스할 수 없습니다. 이 옵션에는 비용이 들지 않으므로 나중에 S3 버킷을 사용할 경우 기본값을 유지할 수 있습니다. 없음을 선택하면 나중에 언제든지 게이트웨이 VPC 엔드포인트를 추가할 수 있습니다.
9. DNS 옵션에서 DNS 호스트 이름 활성화를 선택 취소합니다.
10. VPC 생성을 선택합니다.

2. 애플리케이션 배포

개발 또는 테스트 환경에서 서버 테스트를 완료하고 프로덕션 환경에서 애플리케이션을 배포하는 데 사용할 스크립트 또는 이미지를 생성하는 것이 가장 좋습니다.

Amazon EC2 Auto Scaling을 사용하여 여러 가용 영역에 서버를 배포하고 애플리케이션에 필요한 최소 서버 용량을 유지할 수 있습니다.

Auto Scaling을 사용하여 인스턴스 시작

1. 시작 템플릿을 생성하여 Amazon EC2 Auto Scaling으로 EC2 인스턴스를 시작하는 데 필요한 구성 정보를 지정합니다. 단계별 지침은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling을 위한 시작 템플릿 생성을 참조하세요.
2. EC2 인스턴스 모음인 Auto Scaling을 최소, 최대 및 원하는 크기로 생성합니다. 단계별 지침은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 사용하여 Auto Scaling 생성을 참조하세요.
3. Auto Scaling의 인스턴스 간에 트래픽을 고르게 배포하는 로드 밸런서를 생성하고 Auto Scaling에 로드 밸런서를 연결합니다. 자세한 내용은 Elastic Load Balancing 사용 설명서https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/와 Amazon EC2 Auto Scaling 사용 설명서의 Elastic Load Balancing 사용을 참조하세요.

3. 구성 테스트

애플리케이션 배포를 완료한 후 테스트할 수 있습니다. 애플리케이션이 예상한 트래픽을 전송하거나 수신할 수 없는 경우 Reachability Analyzer를 사용하여 문제를 해결할 수 있습니다. 예를 들어, Reachability Analyzer는 라우팅 테이블 또는 보안 그룹의 구성 문제를 식별할 수 있습니다. 자세한 내용은 Reachability Analyzer 사용 설명서를 참조하세요.

4. 정리

이 구성을 마치면 이를 삭제할 수 있습니다. VPC를 삭제하려면 먼저 Auto Scaling을 삭제하고, 인스턴스를 종료하고, NAT 게이트웨이를 삭제하고, 로드 밸런서를 삭제해야 합니다. 자세한 내용은 the section called “VPC 삭제” 섹션을 참조하세요.

VPC 자습서

Amazon 가상 프라이빗 클라우드(VPC)는 AWS 네트워크 인프라의 기반입니다. AWS Management Console은 사용자 친화적인 인터페이스를 제공하지만 AWS 명령줄 인터페이스(CLI)는 VPC 리소스를 생성 및 관리할 수 있도록 뛰어난 유연성과 자동화 기능을 제공합니다.

이 가이드에서 제공되는 두 가지 필수 VPC 배포 시나리오:

- 퍼블릭 서브넷이 포함된 기본 VPC 설정(간단한 웹 애플리케이션에 적합)
- NAT 게이트웨이를 사용하는 프라이빗 서브넷과 퍼블릭 서브넷이 모두 포함된 고급 VPC 구성(다중 계층 애플리케이션에 적합)

이 튜토리얼을 따라 다음 작업을 수행하는 방법을 알아봅니다.

- 서브넷 구성이 서로 다른 VPC 생성
- 인터넷 및 NAT 게이트웨이 설정
- 라우팅 테이블 및 보안 그룹 구성
- AWS CLI 명령을 사용하여 네트워크 인프라 관리
- AWS 네트워킹 모범 사례 구현

명령줄을 사용하여 AWS 네트워크 인프라 구축을 시작하겠습니다.

자습서

- AWS CLI를 사용하여 Amazon VPC 시작하기
- AWS CLI를 사용하여 프라이빗 서브넷과 NAT 게이트웨이가 포함된 VPC 생성

AWS CLI를 사용하여 Amazon VPC 시작하기

이 튜토리얼에서는 AWS Command Line Interface(AWS Command Line Interface(AWS CLI)AWS AWS CLI)를 사용하여 가상 프라이빗 클라우드(VPC)를 생성하는 방법을 안내합니다. 퍼블릭 및 프라이빗 서브넷이 포함된 VPC를 설정하고, 인터넷 연결을 구성하고, EC2 인스턴스를 배포하여 일반적인 웹 애플리케이션 아키텍처를 시연하는 방법을 알아봅니다.

사전 조건

이 튜토리얼을 시작하기 전에 다음 사항을 확인해야 합니다.

1. AWS CLI. 설치해야 하는 경우 AWS CLI 설치 안내서를 따르세요.
2. 적절한 자격 증명으로 AWS CLI 구성. 자격 증명을 아직 설정하지 않은 경우 `aws configure`를 실행합니다.
3. 네트워킹 개념에 대한 기본적인 이해.
4. AWS 계정에서 VPC 리소스를 생성하고 관리하는 Amazon VPC용 자격 증명 및 액세스 관리.

비용 고려 사항

이 튜토리얼에서는 계정에 비용이 발생할 수 있는 AWS 리소스를 생성합니다. 기본 비용은 NAT 게이트웨이(시간당 0.045 USD + 데이터 처리 비용) 및 EC2 인스턴스(t2.micro, 시간당 약 0.0116 USD)에서 발생합니다. 이 튜토리얼을 1시간 안에 완료하고 모든 리소스를 정리하면 총 비용은 약 0.07 USD가 됩니다. 개발 환경에서 비용을 최적화하려면 NAT 게이트웨이 대신 NAT 인스턴스를 사용하는 것이 좋으며, 비용을 크게 줄일 수 있습니다.

계속하기 전에 AWS CLI가 올바르게 구성되었는지 확인해 보겠습니다.

```
aws configure list
```

AWS 액세스 키, 비밀 키 및 기본 리전이 표시되어야 합니다. 또한 VPC 리소스를 생성하는 데 필요한 권한이 있는지 확인합니다.

```
aws sts get-caller-identity
```

이 명령을 실행하면 AWS 계정 ID, 사용자 ID 및 ARN이 표시되어 자격 증명이 유효한지 확인합니다.

VPC 생성

가상 프라이빗 클라우드(VPC)는 AWS 계정 전용 가상 네트워크입니다. 이 섹션에서는 최대 65,536개의 IP 주소를 제공하는 CIDR 블록이 10.0.0.0/16인 VPC를 생성합니다.

VPC 생성

다음 명령을 실행하면 새 VPC를 생성되고 이름 태그가 할당됩니다.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=MyVPC}]'
```

출력에서 VPC ID를 확인합니다. 후속 명령에 필요합니다. 이 튜토리얼에서는 "vpc-0123456789abcdef0"을 VPC ID 예시로 사용합니다. 모든 명령에서 이를 실제 VPC ID로 바꾸세요.

DNS 지원 및 호스트 이름 활성화

기본적으로 DNS 확인 및 DNS 호스트 이름은 새 VPC에서 비활성화됩니다. 이러한 기능을 활성화하면 VPC의 인스턴스가 도메인 이름을 확인할 수 있습니다.

```
aws ec2 modify-vpc-attribute --vpc-id vpc-0123456789abcdef0 --enable-dns-support
aws ec2 modify-vpc-attribute --vpc-id vpc-0123456789abcdef0 --enable-dns-hostnames
```

이러한 명령은 성공 시 출력을 생성하지 않습니다. 이제 VPC에 DNS 지원 및 호스트 이름 확인이 활성화됩니다.

서브넷 생성

서브넷은 격리된 리소스 그룹을 배치할 수 있는 VPC IP 주소 범위의 세그먼트입니다. 이 섹션에서는 고가용성을 위해 2개의 가용 영역에 퍼블릭 및 프라이빗 서브넷을 생성합니다.

사용 가능한 가용 영역 가져오기

먼저 해당 리전에서 사용 가능한 가용 영역을 검색합니다.

```
aws ec2 describe-availability-zones
```

이 튜토리얼에서는 처음 2개의 가용 영역을 사용합니다. 출력에서 이름을 기록합니다(예: "us-east-1a" 및 "us-east-1b").

퍼블릭 서브넷 생성

퍼블릭 서브넷은 웹 서버와 같이 인터넷에서 액세스해야 하는 리소스에 사용됩니다.

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.0.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-AZ1}]'
```

출력에서 서브넷 ID를 기록합니다. 이 튜토리얼에서는 "subnet-0123456789abcdef0"을 첫 번째 퍼블릭 서브넷의 예시로 사용합니다.

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.1.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-AZ2}]'
```

출력에서 서브넷 ID를 기록합니다. 이 튜토리얼에서는 "subnet-0123456789abcdef1"을 두 번째 퍼블릭 서브넷의 예시로 사용합니다.

프라이빗 서브넷 만들기

프라이빗 서브넷은 데이터베이스와 같이 인터넷에서 직접 액세스할 수 없어야 하는 리소스에 사용됩니다.

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.2.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-AZ1}]'
```

출력에서 서브넷 ID를 기록합니다. 이 튜토리얼에서는 "subnet-0123456789abcdef2"를 첫 번째 프라이빗 서브넷의 예시로 사용합니다.

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.3.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-AZ2}]'
```

출력에서 서브넷 ID를 기록합니다. 이 튜토리얼에서는 "subnet-0123456789abcdef3"을 두 번째 프라이빗 서브넷의 예시로 사용합니다.

이제 2개의 가용 영역에 분산된 2개의 퍼블릭 서브넷과 2개의 프라이빗 서브넷, 총 4개의 서브넷이 있습니다.

팁: CIDR 블록을 계획할 때 기존 네트워크와 겹치지 않도록 하세요. 프로덕션 환경의 경우 향후 증가에 대비해 충분한 IP 주소를 할당하는 동시에 보안 및 관리를 위해 서브넷 크기를 적절하게 유지하세요.

인터넷 연결 구성

VPC의 리소스가 인터넷과 통신하도록 허용하려면 인터넷 게이트웨이를 생성하고 연결해야 합니다. 이 섹션에서는 VPC의 인터넷 연결을 설정합니다.

인터넷 게이트웨이 생성

인터넷 게이트웨이를 통해 VPC와 인터넷 간의 통신이 가능합니다.

```
aws ec2 create-internet-gateway \
  --tag-specifications 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=MyIGW}]'
```

출력에서 인터넷 게이트웨이 ID를 기록합니다. 이 튜토리얼에서는 "igw-0123456789abcdef0"을 예시로 사용하겠습니다.

인터넷 게이트웨이를 VPC에 연결

인터넷 게이트웨이를 생성한 후, VPC에 연결합니다.

```
aws ec2 attach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0
```

라우팅 테이블 생성 및 구성

라우팅 테이블에는 네트워크 트래픽이 전달되는 위치를 결정하는 규칙(라우팅)이 있습니다. 먼저 퍼블릭 서브넷의 라우팅 테이블을 생성합니다.

```
aws ec2 create-route-table \
  --vpc-id vpc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Public-RT}]'
```

출력에서 라우팅 테이블 ID를 기록합니다. 이 튜토리얼에서는 "rtb-0123456789abcdef0"을 퍼블릭 라우팅 테이블의 예시로 사용합니다.

퍼블릭 라우팅 테이블에서 인터넷 게이트웨이 경로를 추가합니다.

```
aws ec2 create-route --route-table-id rtb-0123456789abcdef0 --destination-cidr-block
 0.0.0.0/0 --gateway-id igw-0123456789abcdef0
```

퍼블릭 서브넷과 퍼블릭 라우팅 테이블을 연결합니다.

```
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef0 --subnet-id
 subnet-0123456789abcdef0
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef0 --subnet-id
 subnet-0123456789abcdef1
```

이제 프라이빗 서브넷용 라우팅 테이블을 생성합니다.

```
aws ec2 create-route-table \
  --vpc-id vpc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Private-RT}]'
```

출력에서 라우팅 테이블 ID를 기록합니다. 이 튜토리얼에서는 "rtb-0123456789abcdef1"을 프라이빗 라우팅 테이블의 예시로 사용합니다.

프라이빗 서브넷과 프라이빗 라우팅 테이블을 연결합니다.

```
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef1 --subnet-id
 subnet-0123456789abcdef2
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef1 --subnet-id
 subnet-0123456789abcdef3
```

NAT 게이트웨이 생성

NAT 게이트웨이를 사용하면 프라이빗 서브넷의 인스턴스가 인터넷으로의 아웃바운드 트래픽을 시작하는 동시에 인터넷에서의 인바운드 트래픽을 방지할 수 있습니다. 업데이트를 다운로드하거나 외부 서비스에 액세스해야 하는 인스턴스에 있어 필수적입니다.

탄력적 IP 할당

먼저 NAT 게이트웨이의 탄력적 IP 주소를 할당합니다.

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

출력에서 할당 ID를 기록합니다. 이 튜토리얼에서는 "eipalloc-0123456789abcdef0"을 예시로 사용하겠습니다.

NAT 게이트웨이 생성

할당된 탄력적 IP를 사용하여 퍼블릭 서브넷 중 하나에 NAT 게이트웨이를 생성합니다.

```
aws ec2 create-nat-gateway \
```

```
  --subnet-id subnet-0123456789abcdef0 \
  --allocation-id eipalloc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=MyNATGateway}]'
```

출력에서 NAT 게이트웨이 ID를 기록합니다. 이 튜토리얼에서는 "nat-0123456789abcdef0"을 예시로 사용하겠습니다.

NAT 게이트웨이를 사용할 수 있을 때까지 기다린 다음 계속합니다.

```
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-0123456789abcdef0
```

NAT 게이트웨이에 경로 추가

프라이빗 서브넷의 인스턴스가 인터넷에 액세스할 수 있도록 프라이빗 라우팅 테이블의 NAT 게이트웨이에 라우팅을 추가합니다.

```
aws ec2 create-route --route-table-id rtb-0123456789abcdef1 --destination-cidr-block
 0.0.0.0/0 --nat-gateway-id nat-0123456789abcdef0
```

참고: 프로덕션 환경의 경우 단일 장애 지점을 제거하기 위해 프라이빗 서브넷이 있는 각 가용 영역에 NAT 게이트웨이를 생성하는 것이 좋습니다.

서브넷 설정 구성

시작된 인스턴스에 퍼블릭 IP 주소를 자동으로 할당하도록 퍼블릭 서브넷을 구성합니다.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-0123456789abcdef0 --map-public-ip-
on-launch
aws ec2 modify-subnet-attribute --subnet-id subnet-0123456789abcdef1 --map-public-ip-
on-launch
```

이렇게 하면 퍼블릭 서브넷에서 시작된 인스턴스가 기본적으로 퍼블릭 IP 주소를 수신하므로 인터넷에서 액세스할 수 있습니다.

보안 그룹 생성

보안 그룹은 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다. 이 섹션에서는 웹 서버 및 데이터베이스 서버용 보안 그룹을 생성합니다.

웹 서버용 보안 그룹 생성

```
aws ec2 create-security-group \
  --group-name WebServerSG \
  --description "Security group for web servers" \
  --vpc-id vpc-0123456789abcdef0
```

출력에서 보안 그룹 ID를 기록합니다. 이 튜토리얼에서는 "sg-0123456789abcdef0"을 웹 서버 보안 그룹의 예시로 사용합니다.

웹 서버로의 HTTP 및 HTTPS 트래픽을 허용합니다.

```
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef0 --protocol tcp
 --port 80 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef0 --protocol tcp
 --port 443 --cidr 0.0.0.0/0
```

참고: 프로덕션 환경의 경우 0.0.0.0/0(모든 IP 주소)에서의 트래픽을 허용하는 대신 인바운드 트래픽을 특정 IP 범위로 제한합니다.

데이터베이스 서버용 보안 그룹 생성

```
aws ec2 create-security-group \
  --group-name DBServerSG \
  --description "Security group for database servers" \
  --vpc-id vpc-0123456789abcdef0
```

출력에서 보안 그룹 ID를 기록합니다. 이 튜토리얼에서는 "sg-0123456789abcdef1"을 데이터베이스 서버 보안 그룹의 예시로 사용합니다.

웹 서버에서의 MySQL/Aurora 트래픽만 허용합니다.

```
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef1 --protocol tcp
 --port 3306 --source-group sg-0123456789abcdef0
```

이 구성을 통해 최소 권한 원칙에 따라 웹 서버 보안 그룹의 인스턴스만 포트 3306의 데이터베이스 서버에 연결할 수 있습니다.

VPC 구성 확인

필요한 구성 요소를 모두 생성한 후 VPC 구성을 확인하여 모든 구성 요소가 올바르게 설정되었는지 확인합니다.

VPC 확인

```
aws ec2 describe-vpcs --vpc-id vpc-0123456789abcdef0
```

서브넷 확인

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

라우팅 테이블 확인

```
aws ec2 describe-route-tables --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

인터넷 게이트웨이 확인

```
aws ec2 describe-internet-gateways --filters "Name=attachment.vpc-
id,Values=vpc-0123456789abcdef0"
```

NAT 게이트웨이 확인

```
aws ec2 describe-nat-gateways --filter "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

보안 그룹 확인

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

이러한 명령을 실행하면 VPC의 각 구성 요소에 대한 자세한 정보가 제공되므로 모든 것이 올바르게 구성되었는지 확인할 수 있습니다.

EC2 인스턴스 배포

이제 VPC 인프라를 생성했으므로 EC2 인스턴스를 배포하여 아키텍처의 작동 방식을 시연할 수 있습니다. 퍼블릭 서브넷에서 웹 서버를 시작하고 프라이빗 서브넷에서 데이터베이스 서버를 시작합니다.

SSH 액세스를 위한 키 페어 생성

먼저 인스턴스에 안전하게 연결하도록 키 페어를 생성합니다.

```
aws ec2 create-key-pair --key-name vpc-tutorial-key --query 'KeyMaterial' --output text
 > vpc-tutorial-key.pem
chmod 400 vpc-tutorial-key.pem
```

이 명령을 실행하면 새 키 페어가 생성되고 권한이 제한된 파일에 프라이빗 키가 저장됩니다.

최신 Amazon Linux 2 AMI 찾기

인스턴스에 사용할 최신 Amazon Linux 2 AMI를 찾습니다.

```
aws ec2 describe-images --owners amazon \
  --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2" "Name=state,Values=available"
 \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" --output text
```

출력에서 AMI ID를 기록합니다. 이 튜토리얼에서는 "ami-0123456789abcdef0"을 예시로 사용하겠습니다.

퍼블릭 서브넷에서 웹 서버 시작

이제 퍼블릭 서브넷에서 웹 서버 역할을 할 EC2 인스턴스를 시작합니다.

```
aws ec2 run-instances \
  --image-id ami-0123456789abcdef0 \
  --count 1 \
  --instance-type t2.micro \
  --key-name vpc-tutorial-key \
  --security-group-ids sg-0123456789abcdef0 \
  --subnet-id subnet-0123456789abcdef0 \
  --associate-public-ip-address \
  --user-data '#!/bin/bash
               yum update -y
               yum install -y httpd
               systemctl start httpd
               systemctl enable httpd
               echo "<h1>Hello from $(hostname -f)</h1>" > /var/www/html/index.html' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]'
```

출력에서 인스턴스 ID를 기록합니다. 이 튜토리얼에서는 "i-0123456789abcdef0"을 웹 서버 인스턴스의 예시로 사용합니다.

프라이빗 서브넷에서 데이터베이스 서버 시작

다음으로 프라이빗 서브넷에서 데이터베이스 서버 역할을 할 EC2 인스턴스를 시작합니다.

```
aws ec2 run-instances \
```

```
  --image-id ami-0123456789abcdef0 \
  --count 1 \
  --instance-type t2.micro \
  --key-name vpc-tutorial-key \
  --security-group-ids sg-0123456789abcdef1 \
  --subnet-id subnet-0123456789abcdef2 \
  --user-data '#!/bin/bash
               yum update -y
               yum install -y mariadb-server
               systemctl start mariadb
               systemctl enable mariadb' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=DBServer}]'
```

출력에서 인스턴스 ID를 기록합니다. 이 튜토리얼에서는 "i-0123456789abcdef1"을 데이터베이스 서버 인스턴스의 예시로 사용합니다.

웹 서버 액세스

웹 서버 인스턴스가 실행 중이면 퍼블릭 IP 주소를 사용하여 액세스할 수 있습니다.

```
aws ec2 describe-instances \
  --instance-ids i-0123456789abcdef0 \
  --query 'Reservations[0].Instances[0].PublicIpAddress' \
  --output text
```

이 명령을 실행하면 웹 서버의 퍼블릭 IP 주소가 출력됩니다. 이 튜토리얼에서는 "203.0.113.10"을 예시로 사용하겠습니다.

이제 웹 브라우저에서 http://203.0.113.10 URL을 열 수 있습니다.

SSH를 사용하여 인스턴스에 연결

웹 서버에 연결:

```
ssh -i vpc-tutorial-key.pem ec2-user@203.0.113.10
```

데이터베이스 서버에 연결하려면 먼저 웹 서버에 SSH를 연결하고, 데이터베이스 서버에 SSH를 연결해야 합니다.

```
# Get the private IP of the database server
aws ec2 describe-instances \
  --instance-ids i-0123456789abcdef1 \
```

```
  --query 'Reservations[0].Instances[0].PrivateIpAddress' \
  --output text
```

이 명령을 실행하면 데이터베이스 서버의 프라이빗 IP 주소가 출력됩니다. 이 튜토리얼에서는 "10.0.2.10"을 예시로 사용하겠습니다.

```
# First SSH to web server, then to database server
ssh -i vpc-tutorial-key.pem -A ec2-user@203.0.113.10
ssh ec2-user@10.0.2.10
```

다음은 생성한 네트워크 아키텍처입니다. 웹 서버는 공개적으로 액세스할 수 있는 반면 데이터베이스 서버는 VPC 내에서만 액세스할 수 있습니다.

문제 해결

다음은 VPC 생성 시 발생할 수 있는 몇 가지 일반적인 문제와 이를 해결하는 방법입니다.

CIDR 블록 겹침

CIDR 블록 겹침 관련 오류가 발생하는 경우 VPC 및 서브넷의 CIDR 블록이 계정의 기존 VPC 또는 서브넷과 겹치지 않아야 합니다.

권한 오류

권한 오류가 발생하면 IAM 사용자 또는 역할에 VPC 리소스를 생성 및 관리하는 데 필요한 권한이 있는지 확인합니다. `AmazonVPCFullAccess` 정책을 연결하거나 필요한 권한이 포함된 사용자 지정 정책을 생성해야 할 수 있습니다.

리소스 제한

AWS 계정에는 생성할 수 있는 VPC, 서브넷 및 기타 리소스의 수에 대한 기본 제한이 있습니다. 이러한 제한에 도달하면 AWS 지원 센터를 통해 증가를 요청할 수 있습니다.

정리 중 종속성 오류

리소스를 정리할 때 잘못된 순서로 리소스를 삭제하려고 하면 종속성 오류가 발생할 수 있습니다. 항상 최하위로 종속된 리소스부터 시작하여 생성 순서의 역순으로 리소스를 삭제하세요.

리소스 정리

VPC 사용이 끝나면 요금이 발생하지 않도록 리소스를 정리할 수 있습니다. 종속성을 올바르게 처리하려면 생성 순서의 역순으로 리소스를 삭제하세요.

EC2 인스턴스 종료

```
aws ec2 terminate-instances --instance-ids i-0123456789abcdef0 i-0123456789abcdef1
aws ec2 wait instance-terminated --instance-ids i-0123456789abcdef0 i-0123456789abcdef1
```

키 페어 삭제

```
aws ec2 delete-key-pair --key-name vpc-tutorial-key
rm vpc-tutorial-key.pem
```

NAT 게이트웨이 삭제

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-0123456789abcdef0
aws ec2 wait nat-gateway-deleted --nat-gateway-ids nat-0123456789abcdef0
```

탄력적 IP 릴리스

```
aws ec2 release-address --allocation-id eipalloc-0123456789abcdef0
```

보안 그룹 삭제

```
aws ec2 delete-security-group --group-id sg-0123456789abcdef1
aws ec2 delete-security-group --group-id sg-0123456789abcdef0
```

라우팅 테이블 삭제

먼저 라우팅 테이블 연결 ID를 찾습니다.

```
aws ec2 describe-route-tables --route-table-id rtb-0123456789abcdef0
aws ec2 describe-route-tables --route-table-id rtb-0123456789abcdef1
```

다음으로 서브넷에서 라우팅 테이블의 연결을 해제합니다(연결 ID를 출력의 연결 ID로 바꾸기).

```
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef0
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef1
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef2
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef3
```

그런 다음 라우팅 테이블을 삭제합니다.

```
aws ec2 delete-route-table --route-table-id rtb-0123456789abcdef1
aws ec2 delete-route-table --route-table-id rtb-0123456789abcdef0
```

인터넷 게이트웨이를 분리 및 삭제

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0
aws ec2 delete-internet-gateway --internet-gateway-id igw-0123456789abcdef0
```

서브넷 삭제

```
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef0
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef1
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef2
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef3
```

VPC 삭제

```
aws ec2 delete-vpc --vpc-id vpc-0123456789abcdef0
```

프로덕션으로 이동

이 튜토리얼은 AWS CLI를 사용하여 VPC를 생성하는 방법을 알아보는 데 도움이 되도록 설계되었습니다. 프로덕션 환경의 경우 다음 보안 및 아키텍처 모범 사례를 고려하세요.

1. 보안 그룹 규칙: 0.0.0.0/0에서의 트래픽을 허용하는 대신 인바운드 트래픽을 특정 IP 범위로 제한합니다.
2. 고가용성: 프라이빗 서브넷이 있는 각 가용 영역에 NAT 게이트웨이를 배포하여 단일 장애 지점을 없앱니다.
3. 네트워크 ACL: 네트워크 ACL을 보안 그룹 이외의 추가 보안 계층으로 구현합니다.
4. VPC 흐름 로그: VPC 흐름 로그를 활성화하여 네트워크 트래픽 패턴을 모니터링 및 분석합니다.
5. 리소스 태그 지정: 효과적인 리소스 관리를 위해 포괄적인 태그 지정 전략을 구현합니다.

프로덕션에서 바로 사용할 수 있는 아키텍처 구축에 대한 자세한 내용은 AWS Well-Architected Framework 및 AWS 보안 모범 사례를 참조하세요.

다음 단계

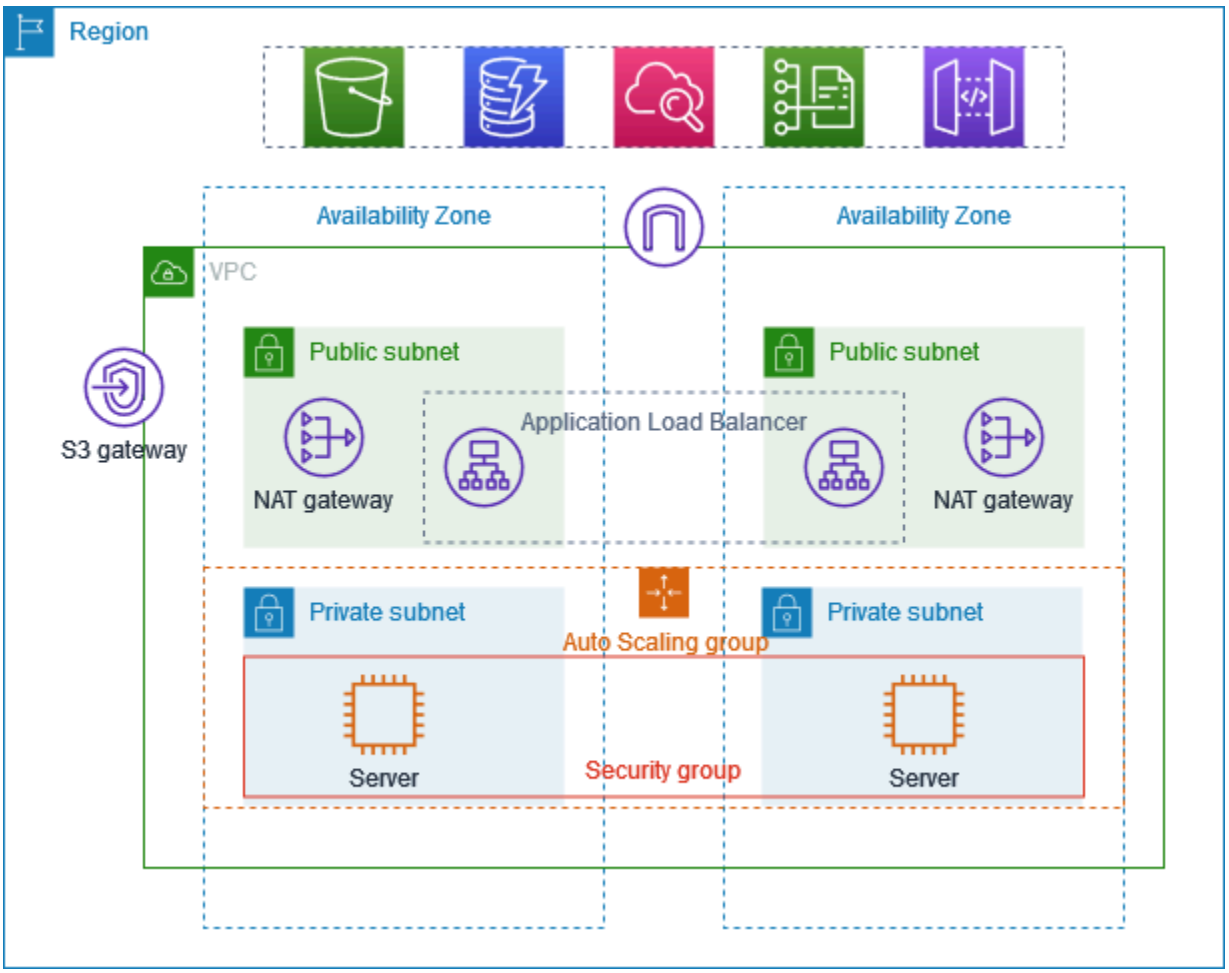
이제 퍼블릭 및 프라이빗 서브넷이 포함된 VPC를 생성했으므로 다음 작업이 가능합니다.

1. 퍼블릭 또는 프라이빗 서브넷에서EC2 인스턴스를 시작합니다.
2. 로드 밸런서를 배포하여 여러 인스턴스에 걸쳐 트래픽을 분산합니다.
3. 고가용성 및 확장성을 위해 Auto Scaling 그룹을 설정합니다.
4. 프라이빗 서브넷에서 RDS 데이터베이스를 구성합니다.
5. VPC 피어링을 구현하여 다른 VPC와 연결합니다.
6. VPC를 온프레미스 네트워크에 연결하도록 VPN 연결을 설정합니다.

AWS CLI를 사용하여 프라이빗 서브넷과 NAT 게이트웨이가 포함된 VPC 생성

이 튜토리얼에서는 AWS CLI를 사용하여 프로덕션 환경의 서버에 사용할 수 있는 VPC를 생성하는 방법을 보여줍니다. 복원력 향상을 위해 Auto Scaling 그룹과 Application Load Balancer를 사용하여 2개의 가용 영역에 서버를 배포합니다. 보안 강화를 위해 프라이빗 서브넷에 서버를 배포합니다. 서버는 로드 밸런서를 통해 요청을 수신하고 NAT 게이트웨이를 사용하여 인터넷에 연결할 수 있습니다. 복원력 향상을 위해 각 가용 영역에 NAT 게이트웨이를 배포합니다.

다음 다이어그램에서는 이 튜토리얼에 포함된 리소스의 개요를 제공합니다. VPC에는 2개의 가용 영역에 퍼블릭 서브넷과 프라이빗 서브넷이 있습니다. 각 퍼블릭 서브넷에는 NAT 게이트웨이와 로드 밸런서 노드가 있습니다. 서버는 프라이빗 서브넷에서 실행되고, Auto Scaling을 사용하여 시작 및 종료되고, 로드 밸런서에서 트래픽을 수신합니다. 서버는 NAT 게이트웨이를 사용하여 인터넷에 연결할 수 있습니다. 서버는 게이트웨이 VPC 엔드포인트를 사용하여 Amazon S3에 연결할 수 있습니다.



사전 조건

이 튜토리얼을 시작하려면 다음이 필요합니다.

- 설치 및 구성된 AWS CLI(VPC 리소스, EC2 인스턴스, 로드 밸런서 및 Auto Scaling 그룹을 생성할 수 있는 권한 포함). AWS CLI 설치에 대한 내용은 AWS CLI의 최신 버전 설치 또는 업데이트를 참고하세요.
- 서브넷, 라우팅 테이블 및 인터넷 게이트웨이를 포함한 VPC 개념에 대한 기본 지식.
- `jq` 명령줄 JSON 프로세서 설치. AWS CLI 명령의 출력을 구문 분석하는 데 사용됩니다. jq 설치에 대한 자세한 내용은 Download jq를 참조하세요.
- 다음을 포함하여 생성할 리소스에 대한 충분한 서비스 할당량:
- 2개 이상의 사용 가능한 탄력적 IP 주소
- 2개 이상의 NAT 게이트웨이
- 1개 이상의 VPC

- 4개 이상의 서브넷
- 1개 이상의 Application Load Balancer

예상 비용: 이 튜토리얼에서 생성된 리소스에 대해 AWS 계정에 요금이 발생합니다. NAT 게이트웨이: 시간당 ~0.045 USD(데이터 처리 요금 별도) 탄력적 IP 주소: 실행 중인 인스턴스와 연결된 경우 무료, 연결되지 않은 경우 시간당 ~ 0.005 USD EC2 인스턴스: 인스턴스 유형에 따라 다름(이 튜토리얼에서는 t3.micro 사용) Application Load Balancer: 시간당 ~0.0225 USD(데이터 처리 요금 별도)

VPC 및 서브넷 생성

먼저 최대 65,536개의 프라이빗 IP 주소를 제공하는 CIDR 블록이 10.0.0.0/16인 VPC를 생성합니다.

```
# Create a VPC with CIDR block 10.0.0.0/16
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=ProductionVPC}]'
```

이 명령은 다음과 비슷한 출력을 반환합니다.

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-abcd1234",
        "State": "pending",
        "VpcId": "vpc-abcd1234",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-abcd1234",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Name",
                "Value": "ProductionVPC"
```

```
            }
        ]
    }
}
```

출력의 VPC ID를 기록하세요(예: vpc-abcd1234). 후속 명령에서 이 ID를 사용합니다.

다음으로 리전에서 2개의 가용 영역을 식별하여 복원력이 있는 아키텍처를 생성합니다.

```
# Get available Availability Zones
aws ec2 describe-availability-zones --query 'AvailabilityZones[0:2].ZoneName' --output
 text
```

이 명령은 다음과 비슷한 출력을 반환합니다.

```
us-east-1a us-east-1b
```

이제 4개의 서브넷을 생성합니다. 로드 밸런서 및 NAT 게이트웨이용 퍼블릭 서브넷 2개, 애플리케이션 서버용 프라이빗 서브넷 2개입니다. vpc-abcd1234를 실제 VPC ID로 바꾸고, us-east-1a 및 us-east-1b를 실제 가용 영역으로 바꿉니다.

```
# Create public subnet in first AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.0.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet1}]'

# Create private subnet in first AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.1.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet1}]'

# Create public subnet in second AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.2.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet2}]'
```

```
# Create private subnet in second AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.3.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet2}]'
```

각 명령으로 서브넷 ID가 포함된 출력이 반환됩니다. 후속 명령에 사용할 ID를 기록합니다.

- 퍼블릭 서브넷 1: subnet-abcd1234
- 프라이빗 서브넷 1: subnet-abcd5678
- 퍼블릭 서브넷 2: subnet-efgh1234
- 프라이빗 서브넷 2: subnet-efgh5678

인터넷 연결 생성 및 구성

이 섹션에서는 VPC와 인터넷 간의 통신을 허용하는 인터넷 게이트웨이를 생성하고, 이를 VPC에 연결합니다.

```
# Create an Internet Gateway
aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-
gateway,Tags=[{Key=Name,Value=ProductionIGW}]'
```

명령을 실행하면 인터넷 게이트웨이 ID가 포함된 출력이 반환됩니다. 이 ID를 기록합니다(예: igw-abcd1234).

인터넷 게이트웨이를 VPC에 연결합니다. igw-abcd1234를 실제 인터넷 게이트웨이 ID로 바꾸고, vpc-abcd1234를 실제 VPC ID로 바꿉니다.

```
# Attach the Internet Gateway to the VPC
aws ec2 attach-internet-gateway --internet-gateway-id igw-abcd1234 --vpc-id vpc-
abcd1234
```

다음으로 퍼블릭 및 프라이빗 서브넷용 라우팅 테이블을 생성합니다. vpc-abcd1234를 실제 VPC ID로 바꿉니다.

```
# Create a route table for public subnets
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PublicRouteTable}]'
```

```
# Create route table for private subnet in first AZ
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable1}]'

# Create route table for private subnet in second AZ
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable2}]'
```

각 명령으로 라우팅 테이블 ID가 포함된 출력이 반환됩니다. 다음 ID를 기록합니다.

- 퍼블릭 라우팅 테이블: `rtb-abcd1234`
- 프라이빗 라우팅 테이블 1: `rtb-efgh1234`
- 프라이빗 라우팅 테이블 2: `rtb-ijkl1234`

퍼블릭 라우팅 테이블에서 인터넷 게이트웨이에 경로를 추가하여 인터넷 액세스를 활성화합니다. `rtb-abcd1234`를 실제 퍼블릭 라우팅 테이블 ID로 바꾸고, `igw-abcd1234`를 실제 인터넷 게이트웨이 ID로 바꿉니다.

```
# Add a route to the Internet Gateway
aws ec2 create-route --route-table-id rtb-abcd1234 --destination-cidr-block 0.0.0.0/0
 --gateway-id igw-abcd1234
```

서브넷을 해당되는 라우팅 테이블과 연결합니다. 라우팅 테이블 ID와 서브넷 ID를 실제 ID로 바꿉니다.

```
# Associate public subnets with the public route table
aws ec2 associate-route-table --route-table-id rtb-abcd1234 --subnet-id subnet-abcd1234
aws ec2 associate-route-table --route-table-id rtb-abcd1234 --subnet-id subnet-efgh1234

# Associate private subnets with their respective route tables
aws ec2 associate-route-table --route-table-id rtb-efgh1234 --subnet-id subnet-abcd5678
aws ec2 associate-route-table --route-table-id rtb-ijkl1234 --subnet-id subnet-efgh5678
```

NAT 게이트웨이 생성

NAT 게이트웨이를 사용하여 프라이빗 서브넷의 인스턴스를 인터넷 또는 다른 AWS 서비스에 연결할 수 있도록 하되, 인터넷에서 해당 인스턴스와의 연결을 시작하지 못하도록 할 수 있습니다. 먼저 NAT 게이트웨이의 탄력적 IP 주소를 할당합니다.

```
# Allocate Elastic IP for NAT Gateway in first AZ
```

```
aws ec2 allocate-address --domain vpc --tag-specifications 'ResourceType=elastic-
ip,Tags=[{Key=Name,Value=NAT1-EIP}]'

# Allocate Elastic IP for NAT Gateway in second AZ
aws ec2 allocate-address --domain vpc --tag-specifications 'ResourceType=elastic-
ip,Tags=[{Key=Name,Value=NAT2-EIP}]'
```

각 명령으로 할당 ID가 포함된 출력이 반환됩니다. 다음 ID를 기록합니다.

- EIP 1 할당 ID: `eipalloc-abcd1234`
- EIP 2 할당 ID: `eipalloc-efgh1234`

각 퍼블릭 서브넷에서 NAT 게이트웨이를 생성합니다. 서브넷 ID와 할당 ID를 실제 ID로 바꿉니다.

```
# Create NAT Gateway in public subnet of first AZ
aws ec2 create-nat-gateway \
  --subnet-id subnet-abcd1234 \
  --allocation-id eipalloc-abcd1234 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway1}]'

# Create NAT Gateway in public subnet of second AZ
aws ec2 create-nat-gateway \
  --subnet-id subnet-efgh1234 \
  --allocation-id eipalloc-efgh1234 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway2}]'
```

각 명령으로 NAT 게이트웨이 ID가 포함된 출력이 반환됩니다. 다음 ID를 기록합니다.

- NAT 게이트웨이 1: `nat-abcd1234`
- NAT 게이트웨이 2: `nat-efgh1234`

NAT 게이트웨이를 프로비저닝하는 데 몇 분 정도 걸립니다. 사용 가능할 때까지 기다린 다음 계속할 수 있습니다. NAT 게이트웨이 ID를 실제 ID로 바꿉니다.

```
# Wait for NAT Gateways to be available
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-abcd1234
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-efgh1234
```

프라이빗 라우팅 테이블의 NAT 게이트웨이에 경로를 추가하여 프라이빗 서브넷의 인스턴스에 대한 인터넷 액세스를 활성화합니다. 라우팅 테이블 ID와 NAT 게이트웨이 ID를 실제 ID로 바꿉니다.

```
# Add route to NAT Gateway 1 in private route table 1
aws ec2 create-route \
  --route-table-id rtb-efgh1234 \
  --destination-cidr-block 0.0.0.0/0 \
  --nat-gateway-id nat-abcd1234

# Add route to NAT Gateway 2 in private route table 2
aws ec2 create-route \
  --route-table-id rtb-ijkl1234 \
  --destination-cidr-block 0.0.0.0/0 \
  --nat-gateway-id nat-efgh1234
```

Amazon S3용 VPC 엔드포인트 생성

Amazon S3용 VPC 엔드포인트를 사용하면 프라이빗 서브넷의 인스턴스가 NAT 게이트웨이를 거치지 않고 S3에 액세스할 수 있어 데이터 전송 비용이 절감되고 네트워크 성능이 향상됩니다. vpc-abcd1234를 실제 VPC ID로 바꾸고, 라우팅 테이블 ID를 실제 ID로 바꿉니다.

```
# Get the prefix list ID for S3 in your region
S3_PREFIX_LIST_ID=$(aws ec2 describe-prefix-lists --filters "Name=prefix-
list-name,Values=com.amazonaws.$(aws configure get region).s3" --query
 'PrefixLists[0].PrefixListId' --output text)

# Create the VPC endpoint for S3
aws ec2 create-vpc-endpoint \
  --vpc-id vpc-abcd1234 \
  --service-name com.amazonaws.$(aws configure get region).s3 \
  --route-table-ids rtb-efgh1234 rtb-ijkl1234 \
  --tag-specifications 'ResourceType=vpc-endpoint,Tags=[{Key=Name,Value=S3-Endpoint}]'
```

명령을 실행하면 VPC 엔드포인트 ID가 포함된 출력이 반환됩니다. 이 ID를 기록합니다(예: vpce-abcd1234).

보안 그룹 구성

보안 그룹은 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 역할을 합니다. 어디서나 인바운드 HTTP 트래픽을 허용하는 로드 밸런서용 보안 그룹을 생성합니다. vpc-abcd1234를 실제 VPC ID로 바꿉니다.

```
# Create security group for the load balancer
aws ec2 create-security-group \
```

```
  --group-name LoadBalancerSG \
  --description "Security group for the load balancer" \
  --vpc-id vpc-abcd1234 \
  --tag-specifications 'ResourceType=security-
group,Tags=[{Key=Name,Value=LoadBalancerSG}]'
```

명령을 실행하면 보안 그룹 ID가 포함된 출력이 반환됩니다. 이 ID를 기록합니다(예: sg-abcd1234).

로드 밸런서에 대한 인바운드 HTTP 트래픽을 허용합니다. sg-abcd1234를 실제 로드 밸런서 보안 그룹 ID로 바꿉니다.

```
# Allow inbound HTTP traffic from anywhere
aws ec2 authorize-security-group-ingress \
  --group-id sg-abcd1234 \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/0
```

로드 밸런서에서의 인바운드 트래픽만 허용하는 애플리케이션 서버용 보안 그룹을 생성합니다. vpc-abcd1234를 실제 VPC ID로 바꿉니다.

```
# Create security group for the application servers
aws ec2 create-security-group \
  --group-name AppServerSG \
  --description "Security group for the application servers" \
  --vpc-id vpc-abcd1234 \
  --tag-specifications 'ResourceType=security-
group,Tags=[{Key=Name,Value=AppServerSG}]'
```

명령을 실행하면 보안 그룹 ID가 포함된 출력이 반환됩니다. 이 ID를 기록합니다(예: sg-efgh1234).

로드 밸런서 보안 그룹에서 애플리케이션 서버로의 인바운드 HTTP 트래픽을 허용합니다. sg-efgh1234를 실제 애플리케이션 서버 보안 그룹 ID로 바꾸고, sg-abcd1234를 실제 로드 밸런서 보안 그룹 ID로 바꿉니다.

```
# Allow inbound HTTP traffic from the load balancer security group
aws ec2 authorize-security-group-ingress \
  --group-id sg-efgh1234 \
  --protocol tcp \
  --port 80 \
  --source-group sg-abcd1234
```

EC2 인스턴스용 시작 템플릿 생성

시작 템플릿에는 AMI ID, 인스턴스 유형 및 보안 그룹 등 인스턴스를 시작하기 위한 구성 정보가 포함되어 있습니다. 먼저 인스턴스가 시작될 때 실행될 사용자 데이터 스크립트를 생성합니다.

```
cat > user-data.sh << 'EOF'
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello from $(hostname -f) in $(curl -s http://169.254.169.254/latest/meta-
data/placement/availability-zone)</h1>" > /var/www/html/index.html
EOF
```

사용자 데이터 스크립트를 base64로 인코딩합니다.

```
USER_DATA=$(base64 -w 0 user-data.sh)
```

최신 Amazon Linux 2 AMI ID를 가져옵니다.

```
# Get the latest Amazon Linux 2 AMI ID
aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-
*-x86_64-gp2" "Name=state,Values=available" --query 'sort_by(Images, &CreationDate)
[-1].ImageId' --output text
```

AMI ID, 인스턴스 유형, 보안 그룹 및 사용자 데이터를 사용하여 시작 템플릿을 생성합니다. `sg-efgh1234`를 실제 애플리케이션 서버 보안 그룹 ID로 바꾸고, `$AMI_ID` 및 `$USER_DATA`를 이전 명령에서 얻은 값으로 바꿉니다.

```
# Create a launch template
aws ec2 create-launch-template \
  --launch-template-name AppServerTemplate \
  --version-description "Initial version" \
  --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=Name,Value=AppServerTemplate}]' \
  --launch-template-data '{
    "NetworkInterfaces": [{
      "DeviceIndex": 0,
      "Groups": ["sg-efgh1234"],
      "DeleteOnTermination": true
```

```
    }],
    "ImageId": "ami-abcd1234",
    "InstanceType": "t3.micro",
    "UserData":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQp5dW0gaW5zdGFsbCAteSBodHRwZApzeXN0ZW1jdGwgc3RhcnQgaHR0cGQKc
+SGVsbG8gZnJvbSAkKGhvc3RuYW1lIC1mKSBpbiAkKGN1cmwgLXMgaHR0cDovLzE2OS4yNTQuMTY5LjI1NC9sYXRlc3QvbW
    "TagSpecifications": [{
      "ResourceType": "instance",
      "Tags": [{
        "Key": "Name",
        "Value": "AppServer"
      }]
    }]
  }'
```

로드 밸런서 및 대상 그룹 생성

대상 그룹은 지정한 프로토콜과 포트를 사용하여 EC2 인스턴스 같은 등록된 대상으로 요청을 라우팅합니다. 애플리케이션 서버의 대상 그룹을 생성합니다. vpc-abcd1234를 실제 VPC ID로 바꿉니다.

```
# Create a target group
aws elbv2 create-target-group \
  --name AppTargetGroup \
  --protocol HTTP \
  --port 80 \
  --vpc-id vpc-abcd1234 \
  --target-type instance \
  --health-check-protocol HTTP \
  --health-check-path / \
  --health-check-port traffic-port
```

명령을 실행하면 대상 그룹 ARN이 포함된 출력이 반환됩니다. 이 ARN을 기록합니다(예: arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/AppTargetGroup/abcd1234).

퍼블릭 서브넷에서 Application Load Balancer를 생성합니다. 서브넷 ID와 보안 그룹 ID를 실제 ID로 바꿉니다.

```
# Create a load balancer
aws elbv2 create-load-balancer \
  --name AppLoadBalancer \
  --subnets subnet-abcd1234 subnet-efgh1234 \
```

```
  --security-groups sg-abcd1234 \
  --tags Key=Name,Value=AppLoadBalancer
```

명령을 실행하면 로드 밸런서 ARN이 포함된 출력이 반환됩니다. 이 ARN을 기록합니다(예: `arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234`).

로드 밸런서가 활성화될 때까지 기다린 다음 계속합니다. `arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234`를 실제 로드 밸런서 ARN으로 바꿉니다.

```
# Wait for load balancer to be active
aws elbv2 wait load-balancer-available \
  --load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234
```

HTTP 트래픽을 대상 그룹으로 전달하는 로드 밸런서의 리스너를 생성합니다. 로드 밸런서 ARN 및 대상 그룹 ARN을 실제 ARN으로 바꿉니다.

```
# Create a listener
aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
app/AppLoadBalancer/abcd1234 \
  --protocol HTTP \
  --port 80 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
east-1:123456789012:targetgroup/AppTargetGroup/abcd1234
```

Auto Scaling 그룹 생성

Auto Scaling 그룹에는 자동 크기 조정 및 관리를 위해 논리적 그룹으로 취급되는 EC2 인스턴스 모음이 포함되어 있습니다. 시작 템플릿을 사용하고 프라이빗 서브넷에 인스턴스를 배치하는 Auto Scaling 그룹을 생성합니다. 서브넷 ID와 대상 그룹 ARN을 실제 ID와 ARN으로 바꿉니다.

```
# Create an Auto Scaling group
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name AppAutoScalingGroup \
  --launch-template LaunchTemplateName=AppServerTemplate,Version='$Latest' \
  --min-size 2 \
  --max-size 4 \
  --desired-capacity 2 \
```

```
  --vpc-zone-identifier "subnet-abcd5678,subnet-efgh5678" \
  --target-group-arns arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/
AppTargetGroup/abcd1234 \
  --health-check-type ELB \
  --health-check-grace-period 300 \
  --tags Key=Name,Value=AppServer,PropagateAtLaunch=true
```

구성 테스트

Auto Scaling 그룹이 인스턴스를 시작하고 상태 확인을 통과한 후 로드 밸런서를 테스트할 수 있습니다. 로드 밸런서의 DNS 이름을 가져옵니다. 로드 밸런서 ARN을 실제 ARN으로 바꿉니다.

```
# Get the DNS name of the load balancer
aws elbv2 describe-load-balancers \
  --load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234 \
  --query "LoadBalancers[0].DNSName" \
  --output text
```

curl을 사용하여 로드 밸런서 이름으로 애플리케이션을 테스트합니다.

```
curl http://LoadBalancerName
```

페이지를 여러 번 새로 고치는 경우 서로 다른 가용 영역에 있는 서로 다른 인스턴스로부터 다른 응답이 표시될 것입니다.

리소스 정리

이 튜토리얼을 완료하면 요금이 발생하지 않도록 모든 리소스를 삭제해야 합니다. 모든 ID를 실제 리소스 ID로 바꿉니다.

```
# Delete the Auto Scaling group
aws autoscaling delete-auto-scaling-group \
  --auto-scaling-group-name AppAutoScalingGroup \
  --force-delete

# Wait for the Auto Scaling group to be deleted
sleep 60

# Delete the load balancer
aws elbv2 delete-load-balancer \
```

```
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
app/AppLoadBalancer/abcd1234

# Wait for the load balancer to be deleted
sleep 30

# Delete the target group
aws elbv2 delete-target-group \
  --target-group-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/
AppTargetGroup/abcd1234

# Delete the launch template
aws ec2 delete-launch-template \
  --launch-template-name AppServerTemplate

# Delete the NAT Gateways
aws ec2 delete-nat-gateway --nat-gateway-id nat-abcd1234
aws ec2 delete-nat-gateway --nat-gateway-id nat-efgh1234

# Wait for the NAT Gateways to be deleted
sleep 90

# Release the Elastic IPs
aws ec2 release-address --allocation-id eipalloc-abcd1234
aws ec2 release-address --allocation-id eipalloc-efgh1234

# Delete the VPC endpoint
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-abcd1234

# Wait for security group dependencies to clear
sleep 30

# Delete the security groups
aws ec2 delete-security-group --group-id sg-efgh1234
aws ec2 delete-security-group --group-id sg-abcd1234

# Detach the Internet Gateway
aws ec2 detach-internet-gateway --internet-gateway-id igw-abcd1234 --vpc-id vpc-
abcd1234

# Delete the Internet Gateway
aws ec2 delete-internet-gateway --internet-gateway-id igw-abcd1234

# Delete the route tables
```

```
aws ec2 delete-route-table --route-table-id rtb-efgh1234
aws ec2 delete-route-table --route-table-id rtb-ijkl1234
aws ec2 delete-route-table --route-table-id rtb-abcd1234

# Delete the subnets
aws ec2 delete-subnet --subnet-id subnet-abcd1234
aws ec2 delete-subnet --subnet-id subnet-efgh1234
aws ec2 delete-subnet --subnet-id subnet-abcd5678
aws ec2 delete-subnet --subnet-id subnet-efgh5678

# Delete the VPC
aws ec2 delete-vpc --vpc-id vpc-abcd1234
```

다음 단계

프라이빗 서브넷 및 NAT 게이트웨이가 포함된 VPC를 생성했으므로 이제 관련 주제를 살펴볼 수 있습니다.

- VPC에 대한 보안 모범 사례
- VPC 흐름 로그를 사용하여 IP 트래픽 로깅
- Auto Scaling group scaling policies
- Load balancer target group health checks

Amazon VPC 할당량

다음 표에는 AWS 계정의 Amazon VPC 리소스 할당량(이전 명칭은 '제한')이 나열되어 있습니다. 달리 표시되지 않는 한 이러한 할당량은 리전당 할당량입니다.

리소스별로 적용되는 할당량 증가를 요청하는 경우 리전에 있는 모든 리소스의 할당량이 증가합니다.

VPC 및 서브넷

이름	기본값	조정 가능	설명
리전당 VPC	5	예*	이 할당량을 늘리면 리전당 인터넷 게이트웨이에 대한 할당량도 같은 수량만큼 늘어납니다. 리전당 수백 개의 VPC를 사용할 수 있도록 이 제한을 늘릴 수 있습니다.
VPC당 서브넷	200	예*	
VPC당 IPv4 CIDR 블록	5	예* (최대 50개)	이 기본 CIDR 블록 및 모든 보조 CIDR 블록은 이 할당량에 포함됩니다.
VPC당 IPv6 CIDR 블록	5	예* (최대 50개)	단일 VPC에 할당할 수 있는 CIDR의 수입니다.
개별 리전의 계정별 VPC 퍼블릭 액세스 차단 제외 항목	50	예. 할당량 증가를 요청하려면 AWS사용자 지원 사용 설명서의 서비스 할당량 증가 요청 섹션을 참조하세요.	계정에서 생성할 수 있는 VPC BPA 제외 항목의 수입니다.

DNS

각 EC2 인스턴스는 Route 53 Resolver로(구체적으로 10.0.0.2 및 169.254.169.253과 같은 .2 주소) 네트워크 인터페이스별 초당 1024개의 패킷을 보낼 수 있습니다. 이 할당량은 늘릴 수 없습니다. Route 53 Resolver가 지원하는 초당 DNS 쿼리 수는 쿼리 유형, 응답 크기 및 사용 중인 프로토콜에 따라 다릅니다. 확장 가능한 DNS 아키텍처에 대한 자세한 내용과 권장 사항은 Active Directory 포함 AWS 하이브리드 DNS 기술 가이드를 확인하세요.

탄력적 IP 주소

이름	기본값	조정 가능	설명
각 리전의 탄력적 IP 주소	5	예*	이 할당량은 개별 AWS 계정 VPC 및 공유 VPC에 적용됩니다.
퍼블릭 NAT 게이트웨이당 탄력적 IP 주소	2	예*	8개까지 할당량 증가를 요청할 수 있습니다.

게이트웨이

이름	기본값	조정 가능	설명
리전당 외부 전용 인터넷 게이트웨이	5	예*	이 할당량을 늘리려면 리전당 VPC 할당량을 늘리세요. 하나의 VPC에는 한 번에 한 개의 외부 전용 인터넷 게이트웨이만 연결할 수 있습니다.
리전당 인터넷 게이트웨이	5	예*	이 할당량을 늘리려면 리전당 VPC 할당량을 늘리세요. 한 번에 하나의 인터넷 게이트웨이만 하나의 VPC에 연결할 수 있습니다.

이름	기본값	조정 가능	설명
가용 영역당 NAT 게이트웨이	5	예*	NAT 게이트웨이는 `pending`, `active`, `deleting` 상태의 할당량에 포함됩니다.
NAT 게이트웨이당 프라이빗 IP 주소 할당량	8	예*	
VPC당 통신 사업자 게이트웨이	1	아니요	

고객 관리형 접두사 목록

고객 관리형 접두사 목록의 기본 할당량은 조정 가능하지만 Service Quotas 콘솔을 사용하여 증가를 요청할 수는 없습니다. Support 사례를 개설해야 합니다. AWS사용자 지원 사용 설명서의 서비스 할당량 증가 요청 섹션을 참조하세요.

이름	기본값	조정 가능	설명
리전별 접두사 목록	100	예	
접두사 목록당 버전	1,000	예	접두사 목록에 1,000개의 저장된 버전이 있고 새 버전을 추가하는 경우 새 버전이 추가되도록 가장 오래된 버전이 제거됩니다.
접두사 목록당 최대 항목 수	1,000	예	고객 관리형 접두사 목록의 크기를 최대 1000개까지 조정할 수 있습니다. 자세한 내용은 섹션을 참조하세요접두사 목록 크기 조정 리소스의 접두사 목록을 참조할 때 접두사 목록의 최대 항목 수는 리소스의 항목 수에 대한 할당량에 따라 계산됩니다. 예를 들어, 최대 항목이 20개인 접두사 목록을 만들고 보안 그룹 규칙에서 해당 접두사 목록을 참조하는 경우, 20개의 보안 그룹 규칙이 있는 것으로 계산됩니다.

이름	기본값	조정 가능	설명
리소스 유형별 접두사 목록 참조	10,000개	예	이 할당량은 접두사 목록을 참조할 수 있는 리소스 유형별로 적용됩니다. 예를 들어, 모든 보안 그룹에서 접두사 목록에 대한 참조 10,000개와 모든 서브넷 라우팅 테이블에서 접두사 목록에 대한 참조 10,000개를 가질 수 있습니다. 접두사 목록을 다른 AWS 계정과 공유하는 경우, 접두사 목록에 대한 다른 계정의 참조가 사용자 할당량에 포함됩니다.

네트워크 ACL

이름	기본값	조정 가능	설명
VPC당 네트워크 ACL	200	예*	하나의 VPC에 있는 한 개 이상의 서브넷에 한 개의 네트워크 ACL을 연결할 수 있습니다.
네트워크 ACL당 규칙	20	예	이 할당량은 최대 인바운드 규칙 수와 최대 아웃바운드 규칙 수를 결정합니다. 이 할당량은 최대 40개의 인바운드 규칙과 40개의 아웃바운드 규칙(총 80개의 규칙)까지 늘릴 수 있지만 네트워크 성능에 영향을 미칠 수 있습니다.

네트워크 인터페이스

이름	기본값	조정 가능	설명
인스턴스당 네트워크 인터페이스	인스턴스 유형에 따	아니요	자세한 내용은 인스턴스 유형당 네트워크 인터페이스를 참조하세요.

이름	기본값	조정 가능	설명
	라 다릅니다.		
리전당 네트워크 인터페이스	5,000	예*	이 할당량은 개별 AWS 계정 VPC 및 공유 VPC에 적용됩니다. 이 제한은 가용 영역별로 적용됩니다. 예를 들어, 네트워크 인터페이스가 3개의 AZ에 있는 경우 각 AZ의 제한은 5,000이고 리전의 제한은 15,000입니다.

라우팅 테이블

이름	기본값	조정 가능	설명
VPC당 라우팅 테이블	200	예*	기본 라우팅 테이블은 이 할당량에 포함됩니다. 라우팅 테이블에 대한 할당량 증가를 요청하는 경우 서브넷에 대한 할당량 증가를 요청할 수도 있습니다. 라우팅 테이블들을 여러 개의 서브넷과 공유할 수 있지만, 한 서브넷을 한 번에 단 하나의 라우팅 테이블과 연결할 수 있습니다.
라우팅 테이블에 따른 경로 (전파되지 않는 경로)	500	예*	이 할당량은 최대 1,000개까지 늘릴 수 있지만 네트워크 성능에 영향을 줄 수 있습니다. 이 할당량은 IPv4 및 IPv6 경로에 개별적으로 적용됩니다. 경로가 125개 이상 있다면 성능 향상을 위해 호출에 페이지 번호를 붙여 라우팅 테이블을 설명하는 것이 좋습니다.
라우팅 테이블에 따른 전파되는 경로	100	아니요	추가 접두사가 필요할 경우 기본 경로를 알리세요.

라우팅 서버

이름	기본값	조정 가능	설명
VPC당 라우팅 서버	5	예. 할당량 증가를 요청하려면 AWS 사용자 지원 사용 설명서의 서비스 할당량 증가 요청 섹션을 참조하세요.	
라우팅 서버당 라우팅 서버 엔드포인트	10	예. 할당량 증가를 요청하려면 AWS 사용자 지원 사용 설명서의 서비스 할당량 증가 요청 섹션을 참조하세요.	
네트워크 인터페이스당 피어링 세션	20	예. 할당량 증가를 요청하려면 AWS 사용자 지원 사용 설명서의	

이름	기본값	조정 가능	설명
		서비스 할당량 증가 요청 섹션을 참조하세요.	
라우팅 서버 및 서브넷당 라우팅 서버 엔드포인트	2	아니요	중복성을 위해 동일한 라우팅 서버의 동일한 서브넷에 2개의 엔드포인트만 있을 수 있습니다.
라우팅 서버 피어당 경로	100	아니요	라우팅 서버 피어를 통해 동적으로 알릴 수 있는 경로 수
라우팅 서버당 경로	100	아니요	라우팅 서버의 FIB(Forwarding Information Base)에 설치할 수 있는 경로 수입니다.

보안 그룹

이름	기본값	조정 가능	설명
리전당 VPC 보안 그룹	2,500	예*	이 할당량은 개별 AWS 계정 VPC 및 공유 VPC에 적용됩니다. 리전 하나에 할당량을 보안 그룹 5,000개 이상으로 늘리는 경우, 성능 향상을 위해 호출에 페이지 매김을 하여 보안 그룹을 설명하는 것이 좋습니다.
보안 그룹별 인바운드 또는 아웃바운드 규칙	60	예*	이 할당량은 인바운드 규칙과 아웃바운드 규칙에 개별적으로 적용됩니다. 규칙 60개의 기본 할당량을 가진 계정의 경우 보안 그룹은 인바운드 규칙 60개와 아웃바운드 규칙 60개를 가질 수 있습니다. 또한 이 할당량은 IPv4 및 IPv6 규칙에 개

이름	기본값	조정 가능	설명
			별적으로 적용됩니다. 규칙 60개의 기본 할당량을 가진 계정의 경우 보안 그룹은 IPv4 트래픽에 대한 인바운드 규칙 60개와 IPv6 트래픽에 대한 인바운드 규칙 60개를 가질 수 있습니다. 자세한 내용은 섹션을 참조하세요the section called “보안 그룹 크기” 할당량 변경은 인바운드 규칙과 아웃바운드 규칙에 모두 적용됩니다. 이 할당량과 네트워크 인터페이스당 보안 그룹의 할당량을 곱한 값이 1,000을 초과하면 안 됩니다.
네트워크 인터페이스당 보안 그룹	5	예* (최대 16개)	이 할당량과 보안 그룹당 규칙 할당량을 곱한 값이 1,000을 초과하면 안 됩니다. 네트워크 인터페이스당 보안 그룹 할당량을 기본값인 5 미만으로 줄이려면 AWS 사용자 지원 사용 설명서의 서비스 할당량 증가 요청을 참조하세요. 증가를 요청할 때와 동일한 프로세스로 감소를 요청합니다.

VPC 서브넷 공유

모든 표준 VPC 할당량은 공유된 VPC 서브넷에 적용됩니다.

이름	기본값	조정 가능	설명
VPC당 참가자 계정	100	예	VPC의 서브넷을 공유할 수 있는 참가자 계정의 최대 수입니다. 이는 VPC당 할당량이며, VPC에서 공유되는 모든 서브넷에 적용됩니다.

이름	기본값	조정 가능	설명
			VPC 소유자는 참가자 리소스에 연결된 네트워크 인터페이스 및 보안 그룹을 볼 수 있습니다.
계정과 공유할 수 있는 서브넷	100	예	AWS 계정과 공유할 수 있는 최대 서브넷 수입니다.

네트워크 주소 사용량

NAU(네트워크 주소 사용량)는 IP 주소, 네트워크 인터페이스 및 관리형 접두사 목록의 CIDR로 구성됩니다. NAU는 VPC 크기를 계획하고 모니터링하는 데 도움이 되도록 VPC의 리소스에 적용되는 지표입니다. 자세한 내용은 섹션을 참조하세요네트워크 주소 사용량

NAU 개수를 구성하는 리소스에는 자체적인 개별 서비스 할당량이 있습니다. VPC에 사용 가능한 NAU 용량이 있더라도 리소스에서 해당 서비스 할당량을 초과하면 리소스를 VPC에서 시작할 수 없습니다.

이름	기본값	조정 가능	설명
네트워크 주소 사용량	64,000	예(최대 256,000개)	VPC당 최대 NAU 단위 수입니다.
피어링된 네트워크 주소 사용량	128,000	예(최대 512,000개)	VPC 및 해당 VPC와 리전 내 피어링된 모든 VPC의 최대 NAU 단위 수입니다. 다른 리전 간에 피어링된 VPC는 이 수에 포함되지 않습니다.

Amazon EC2 API 스로틀링

Amazon EC2 스로틀링에 대한 자세한 내용은 Amazon EC2 개발자 안내서의 스로틀링 요청을 참조하세요.

추가 할당량 리소스

자세한 내용은 다음을 참조하세요.

- AWS Client VPN 관리자 가이드의 AWS Client VPN 할당량
- AWS Direct Connect 사용 설명서의 Direct Connect 할당량
- Amazon VPC 피어링 설명서의 피어링 할당량
- AWS PrivateLink 설명서의 PrivateLink 할당량
- AWS Site-to-Site VPN 사용 설명서의 Site-to-Site VPN 할당량
- Amazon VPC Traffic Mirroring 설명서의 Traffic Mirroring 할당량
- Amazon VPC Transit Gateway 설명서의 Transit Gateway 할당량

문서 기록

다음 표에서는 Amazon VPC 사용 설명서의 각 릴리스에서 변경된 중요 사항에 관해 설명합니다.

변경 사항	설명	날짜
VPC 암호화 제어	이제 VPC 암호화 제어를 사용하여 VPC 내의 네트워크 트래픽에 전송 중 암호화를 적용할 수 있습니다. 이 기능은 중앙 집중식 암호화 정책 적용 및 모니터링 기능을 제공합니다.	2025년 11월 21일
자동 다중 AZ 확장을 위한 리전 NAT 게이트웨이	이제 워크로드 공간에 따라 가용 영역 간에 자동으로 확장되는 리전 NAT 게이트웨이를 사용할 수 있습니다. 리전 NAT 게이트웨이는 수동 개입 없이 설정 간소화, 보안 향상 및 자동 고가용성을 제공합니다.	2025년 11월 19일
인바운드 VPC 트래픽을 퍼블릭 IP 주소로 라우팅	이제 인터넷의 인바운드 트래픽을 VPC 내의 특정 퍼블릭 IP 주소로 전달하도록 고급 라우팅 규칙을 구성하여 수신 시나리오에 대한 트래픽 흐름 및 라우팅 결정을 더욱 세밀하게 제어할 수 있습니다.	2025년 8월 13일
Amazon VPC Route Server를 사용한 VPC의 동적 라우팅	Amazon VPC Route Server는 VPC 내에 배포된 워크로드와 인터넷 게이트웨이 간의 트래픽 라우팅을 간소화합니다. 이 기능을 사용하면 VPC Route Server가 선호하는 IPv4 또는 IPv6 경로로 VPC 및 게이트웨이 라우팅 테이블을 동적으로	2025년 3월 31일

	업데이트하여 해당 워크로드에 대한 라우팅 내결함성을 달성합니다. 이를 활용하면 VPC 내에서 트래픽을 자동으로 다시 라우팅할 수 있어 VPC 라우팅의 관리 용이성과 타사 워크로드와의 상호 운용성이 개선됩니다.	
AWS 관리형 정책 업데이트	Amazon VPC의 AmazonVPCFullAccess 및 AmazonVPCReadOnlyAccess 관리형 정책이 업데이트되었습니다.	2024년 12월 9일
VPC BPA에 대한 선언적 정책 지원	AWS Organizations를 사용하여 조직의 계정을 관리하는 경우 선언적 정책을 사용하여 조직의 계정에 VPC BPA를 적용할 수 있습니다.	2024년 12월 1일
VPC 퍼블릭 액세스 차단(BPA)	VPC 퍼블릭 액세스 차단(BPA)을 사용하면 인터넷 게이트웨이 및 외부 전용 인터넷 게이트웨이를 통해 리전에서 소유한 VPC 및 서브넷의 리소스와 인터넷 간의 송신 또는 수신을 차단할 수 있습니다.	2024년 11월 19일
공유 보안 그룹	이 기능을 사용하면 보안 그룹을 다른 AWS Organizations 계정과 공유할 수 있습니다.	2024년 10월 30일
보안 그룹 VPC 연결	이 기능을 사용하면 보안 그룹을 동일한 리전의 여러 VPC와 연결할 수 있습니다.	2024년 10월 30일

NAT 게이트웨이 MTU 지원	NAT 게이트웨이에서 지원되는 트래픽의 최대 전송 단위(MTU)는 8,500입니다.	2024년 9월 10일
프라이빗 IPv6 주소 지정	프라이빗 IPv6 주소 지정에 대한 정보가 추가되었습니다. 프라이빗 IPv6 주소는 Amazon VPC IP 주소 관리자에서만 사용할 수 있습니다.	2024년 8월 8일
IPv6 선호 임대 시간	이제는 IPv6가 할당되어 실행 중인 인스턴스의 DHCPv6 임대 갱신 빈도를 선택할 수 있습니다.	2024년 2월 20일
가이드 구조 검토 및 개선	특정 시나리오에 대한 정보 찾기와 관련된 고객 경험을 개선하기 위해 가이드의 구조를 검토하고 개선했습니다.	2024년 2월 20일
AWS 관리형 정책 업데이트	Amazon VPC의 AmazonVPCFullAccess 및 AmazonVPCReadOnlyAccess 관리형 정책이 업데이트되었습니다.	2024년 2월 8일
AWS 관리형 정책 업데이트	Amazon VPC의 AmazonVPCCrossAccountNetworkInterfaceOperations 관리형 정책이 업데이트되었습니다.	2023년 9월 25일

EC2-Classic 더 이상 사용되지 않음	EC2-Classic을 사용하여 EC2 인스턴스는 다른 고객과 공유되는 단일 플랫 네트워크에서 실행되었습니다. Amazon VPC가 EC2-Classic을 대체합니다. Amazon VPC 사용을 통해 AWS 계정에 속하도록 논리적으로 독립된 가상 프라이빗 클라우드(VPC)에서 인스턴스가 실행됩니다.	2023년 7월 31일
NAT 게이트웨이에 보조 IPv4 주소 추가	퍼블릭 및 프라이빗 NAT 게이트웨이에 보조 프라이빗 IPv4 주소를 추가할 수 있습니다. 보조 IPv4 주소는 사용 가능한 포트 수를 늘리므로 워크로드가 NAT 게이트웨이를 사용하여 설정할 수 있는 동시 연결 수에 대한 제한이 늘어납니다.	2023년 1월 31일
IAM 모범 실무 따르기	IAM 모범 실무에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례를 참조하세요.	2023년 1월 4일
NAT 게이트웨이의 프라이빗 IP 주소 선택	NAT 게이트웨이를 생성할 때 이제 NAT 게이트웨이에 할당된 프라이빗 IP 주소를 선택할 수 있습니다. 이전에는 서브넷의 IP 주소 범위로부터 프라이빗 IP 주소가 자동으로 할당되었습니다.	2022년 11월 17일
IPv6 기본 게이트웨이 라우터 구성	이제 기본 VPC 라우터에서 사용할 수 있도록 세 개의 IPv6 주소가 예약되어 있습니다.	2022년 11월 11일

탄력적 IP 주소 전송	이제 하나의 AWS 계정에서 다른 계정으로 탄력적 IP 주소를 전송할 수 있습니다.	2022년 10월 31일
네트워크 주소 사용량 지표	VPC의 크기를 계획하고 모니터링하는 데 도움이 되도록 VPC의 네트워크 주소 사용량 지표를 활성화할 수 있습니다.	2022년 10월 4일
Amazon Data Firehose에 흐름 로그 게시	Amazon Data Firehose 전송 스트림을 흐름 로그 데이터의 대상으로 지정할 수 있습니다.	2022년 9월 8일
NAT 게이트웨이 대역폭	NAT 게이트웨이는 이제 최대 100Gbps의 대역폭(기존 45Gbps에서 증가)을 지원하며, 초당 최대 1,000만 개의 패킷(기존 400만 개에서 증가)을 처리할 수 있습니다.	2022년 6월 15일
다중 IPv6 CIDR 블록	IPv6 CIDR 블록은 최대 5개까지 VPC에 연결할 수 있습니다.	2022년 5월 12일
재구성	Amazon Virtual Private Cloud 사용 설명서의 일반적인 재구성입니다.	2022년 1월 2일
NAT 게이트웨이 IPv6에서 IPv4로 연결	NAT 게이트웨이는 IPv6에서 IPv4로의 네트워크 주소 변환을 지원합니다(일반적으로 NAT64라고 함).	2021년 11월 24일
VPC의 IPv6 전용 서브넷	IPv6 전용 EC2 인스턴스를 시작할 수 있는 IPv6 전용 서브넷을 생성할 수 있습니다.	2021년 11월 23일

Amazon S3로 VPC 흐름 로그 전송 옵션	Apache Parquet 로그 파일 형식, 시간별 분할 및 Hive 호환 S3 접두사를 지정할 수 있습니다.	2021년 10월 13일
Amazon EC2 Global View	Amazon EC2 Global View를 사용하면 단일 콘솔의 여러 AWS 리전에 걸쳐 VPC, 서브넷, 인스턴스, 보안 그룹 및 볼륨을 볼 수 있습니다.	2021년 9월 1일
더 구체적인 경로	로컬 경로보다 더 구체적인 경로를 라우팅 테이블에 추가할 수 있습니다. 보다 구체적인 경로를 사용하여 VPC 내의 서브넷 간의 트래픽(동서 트래픽)을 미들박스 어플라이언스로 리디렉션할 수 있습니다. 경로의 대상은 VPC에 있는 서브넷의 전체 IPv4 또는 IPv6 CIDR 블록과 일치하도록 설정할 수 있습니다.	2021년 8월 30일
보안 그룹 규칙에 대한 리소스 ID 및 태깅 지원	리소스 ID별로 보안 그룹 규칙을 참조할 수 있습니다. 보안 그룹 규칙에 태그를 추가할 수도 있습니다.	2021년 7월 7일
프라이빗 NAT 게이트웨이	프라이빗 NAT 게이트웨이를 VPC 간 또는 VPC와 온프레미스 네트워크 간의 아웃바운드 전용 프라이빗 통신에 사용할 수 있습니다.	2021년 6월 10일

생성 시 태그	VPC, DHCP 옵션, 인터넷 게이트웨이, 외부 전용 게이트웨이, 네트워크 ACL 및 보안 그룹을 생성할 때 태그를 추가할 수 있습니다.	2020년 6월 30일
관리형 접두사 목록	접두사 목록에서 CIDR 블록 세트를 생성 및 관리할 수 있습니다.	2020년 6월 29일
흐름 로그 개선	새 흐름 로그 필드를 사용할 수 있으며 CloudWatch Logs에 게시되는 흐름 로그의 사용자 지정 형식을 지정할 수 있습니다.	2020년 5월 4일
흐름 로그에 대한 태그 지정 지원	흐름 로그에 태그를 추가할 수 있습니다.	2020년 3월 16일
NAT 게이트웨이 생성 시 태그	NAT 게이트웨이를 만들 때 태그를 추가할 수 있습니다.	2020년 3월 9일
흐름 로그의 최대 집계 간격	흐름을 캡처하고 흐름 로그 레코드로 집계하는 최대 기간을 지정할 수 있습니다.	2020년 2월 4일
네트워크 경계 그룹 구성	Amazon VPC 콘솔에서 VPC에 대한 네트워크 경계 그룹을 구성할 수 있습니다.	2020년 1월 22일
게이트웨이 라우팅 테이블	라우팅 테이블을 게이트웨이와 연결하고 인바운드 VPC 트래픽을 VPC의 특정 네트워크 인터페이스로 라우팅할 수 있습니다.	2019년 12월 3일

흐름 로그 개선	흐름 로그의 사용자 지정 형식을 지정하고 흐름 로그 레코드에 반환할 필드를 선택할 수 있습니다.	2019년 9월 11일
VPC 공유	동일한 VPC에 있는 서브넷을 동일한 AWS 조직의 여러 계정과 공유할 수 있습니다.	2018년 11월 27일
기본 서브넷 생성	기본 서브넷이 없는 가용 영역에서 기본 서브넷을 생성할 수 있습니다.	2017년 11월 9일
NAT 게이트웨이에 태그 지정 지원	NAT 게이트웨이에 태그를 지정할 수 있습니다.	2017년 9월 7일
NAT 게이트웨이에 대한 Amazon CloudWatch 지표	NAT 게이트웨이에 대한 CloudWatch 지표를 볼 수 있습니다.	2017년 9월 7일
보안 그룹 규칙 설명	보안 그룹 규칙에 설명을 추가할 수 있습니다.	2017년 8월 31일
VPC의 보조 IPv4 CIDR 블록	VPC에 여러 개의 IPv4 CIDR 블록을 추가할 수 있습니다.	2017년 8월 29일
탄력적 IP 주소 복구	탄력적 IP 주소를 해제한 경우 복구할 수 있습니다.	2017년 8월 11일
기본 VPC 만들기	기존 기본 VPC를 삭제한 경우 새로운 기본 VPC를 만들 수 있습니다.	2017년 7월 27일
IPv6 지원	IPv6 CIDR 블록을 VPC에 연결하고 IPv6 주소를 VPC의 리소스에 할당할 수 있습니다.	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.	2016년 10월 24일
NAT 게이트웨이	퍼블릭 서브넷에서 NAT 게이트웨이를 만들고 프라이빗 서브넷의 인스턴스를 활성화하여 인터넷 또는 다른 AWS 서비스로 가는 아웃바운드 트래픽을 시작할 수 있습니다.	2015년 12월 17일
VPC 흐름 로그	흐름 로그를 생성하여 VPC의 네트워크 인터페이스에서 전송하고 수신하는 IP 트래픽에 대한 정보를 캡처할 수 있습니다.	2015년 6월 10일
ClassicLink	ClassicLink를 사용하여 EC2-Classic 인스턴스를 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.	2015년 1월 7일
프라이빗 호스팅 영역 사용	Route 53의 프라이빗 호스팅 영역에서 정의한 사용자 지정 DNS 도메인 이름을 사용하여 VPC의 리소스에 액세스할 수 있습니다.	2014년 11월 5일

서브넷의 퍼블릭 IP 주소 지정 속성 변경	사용자 서브넷의 퍼블릭 IP 주소 지정 속성을 변경하여 해당 서브넷에서 시작한 인스턴스가 퍼블릭 IP 주소를 받을지 여부를 지정할 수 있습니다.	2014년 6월 21일
퍼블릭 IP 주소 할당	시작하는 과정에서 인스턴스에 퍼블릭 IP 주소를 할당할 수 있습니다.	2013년 8월 20일
DNS 호스트 이름 활성화 및 DNS 해석 비활성화	VPC 기본값을 수정하고 DNS 확인을 비활성화하고 DNS 호스트 이름을 활성화할 수 있습니다.	2013년 3월 11일
어디서나 사용 가능한 VPC	5개 AWS 리전의 VPC, 여러 가용 영역의 VPC, AWS 계정당 여러 VPC 및 VPC당 여러 VPN 연결에 대한 지원이 추가되었습니다.	2011년 8월 3일
전용 인스턴스	전용 인스턴스는 단일 고객에게 배정된 하드웨어를 실행하는 VPC 내에서 시작되는 Amazon EC2 인스턴스입니다.	2011년 3월 27일