티맵 네트워크 재조립하기

많은 VPC 효율적으로 관리하기

오동근, DevOps Engineer

2023-11-15

TMAP MOBILITY

목차

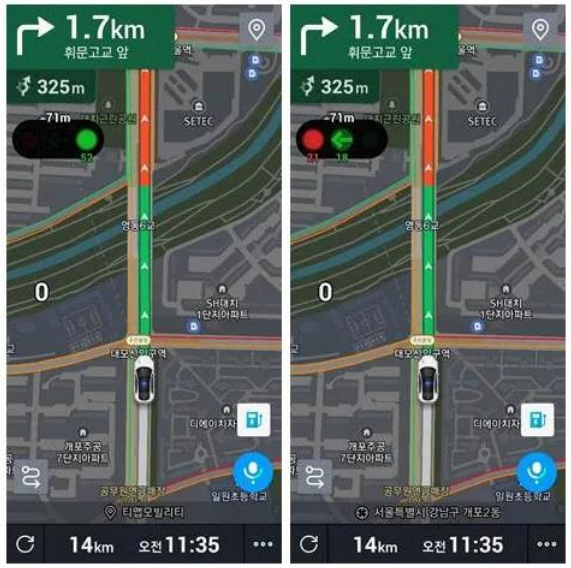
티맵모빌리티 소개

1. 티맵모빌리티
2. AWS Migration

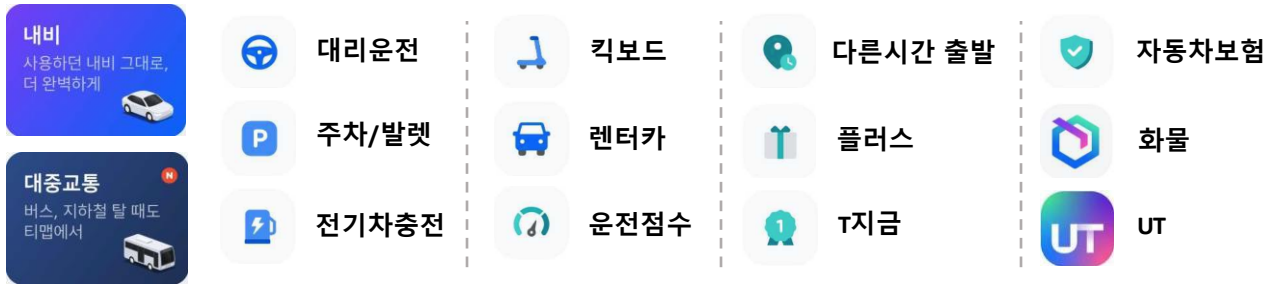
티맵모빌리티 소개

2002년부터 20년넘게 사랑받는 국민 내비게이션 서비스

과거 티맵 화면



최근 티맵 화면

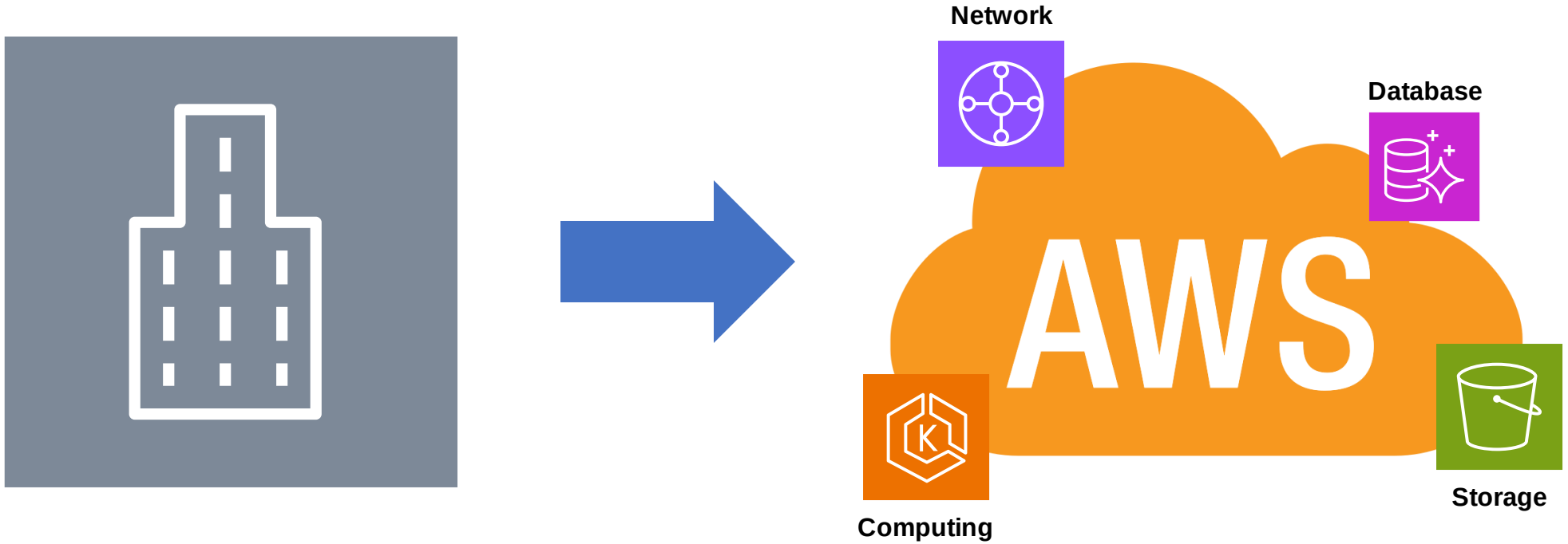


내비게이션을 포함한 다양한 서비스

2,000만 가입자	10,000 TPS (최대치)
1,400만 MAU	3,000대 서버
600만 DAU	300억 로그 (월)
월 실행횟수 약 50회 월 실행시간 약 6시간	약 150여종의 서버

AWS Migration

IDC기반의 인프라를 AWS로 이전하는 프로젝트 진행 중

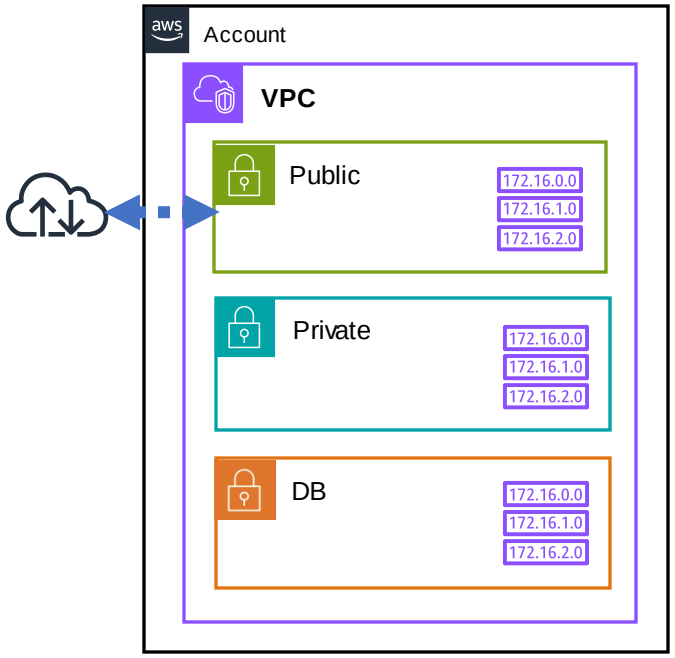


네트워크 구성

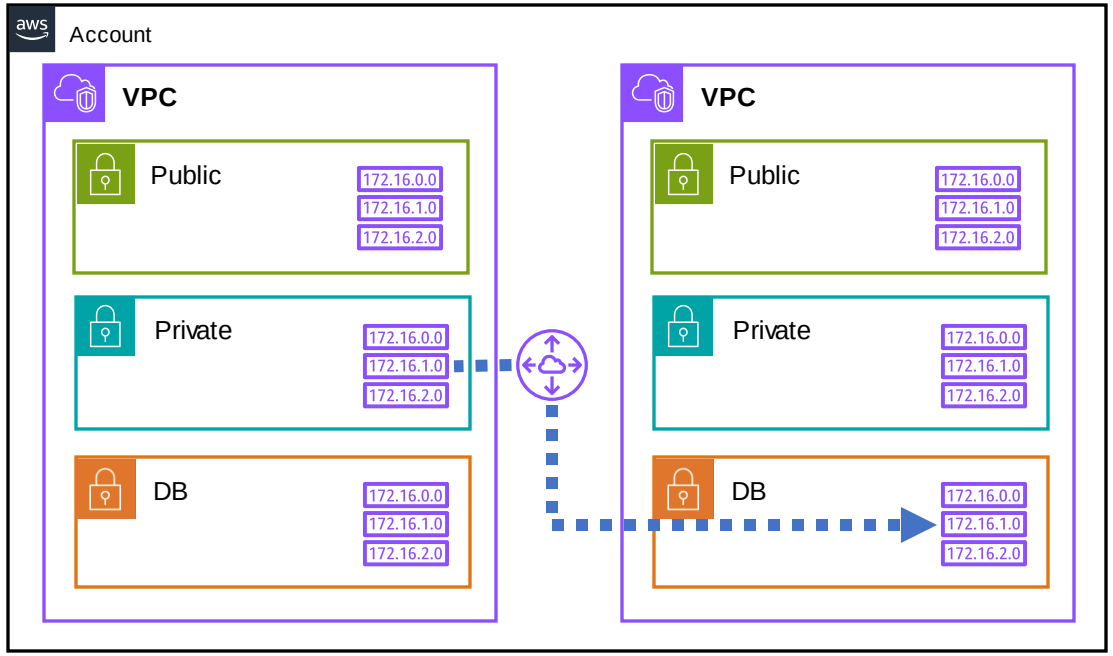
1. 네트워크 성장과정
2. 티맵 네트워크 구성
3. 문제점
4. 개선 요구사항

네트워크 성장 과정: 단순한 시작

하나의 VPC로 시작해 점점 VPC가 증가



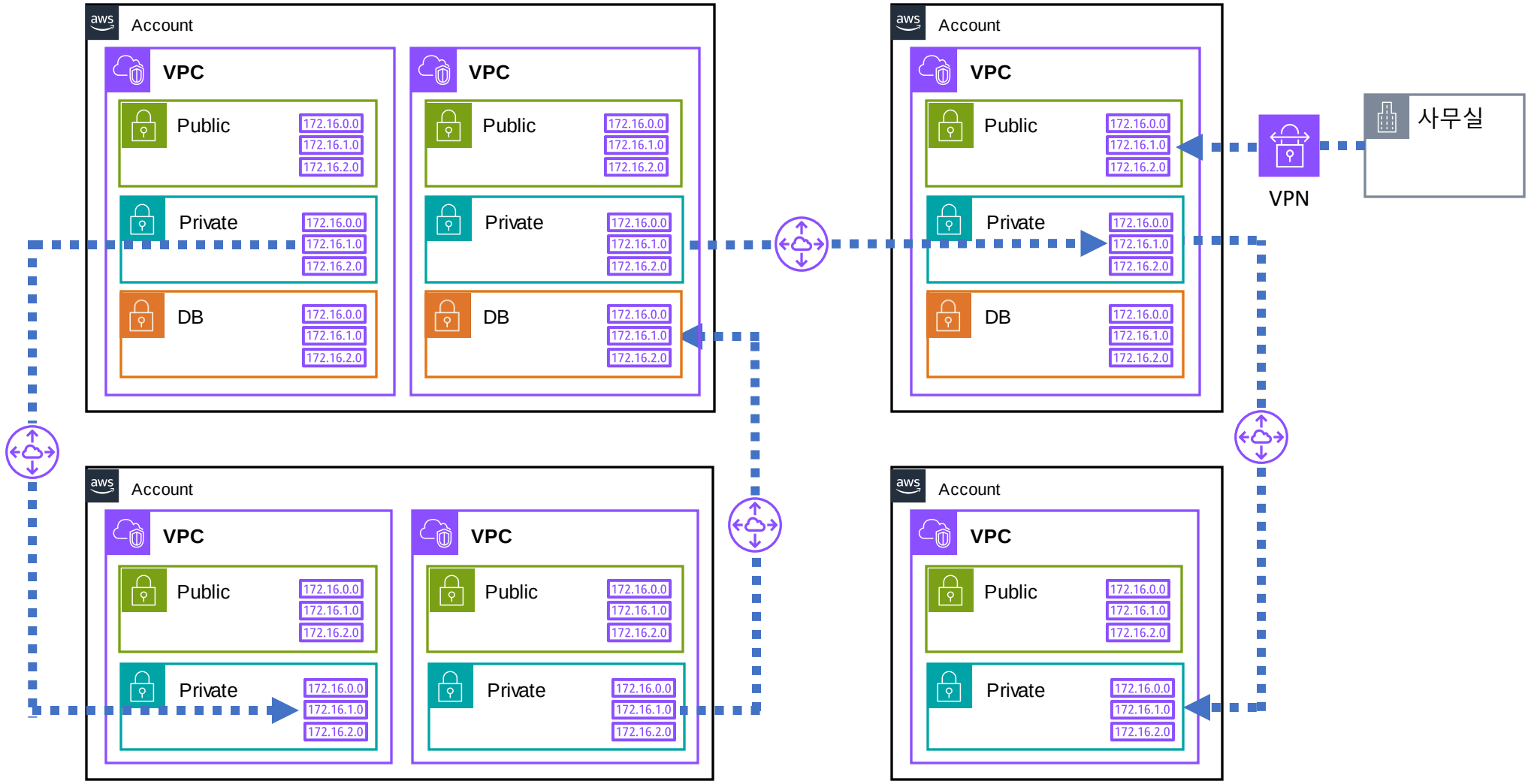
단일 VPC 구성



복수 VPC 구성과 VPC Peering을 통한 연결

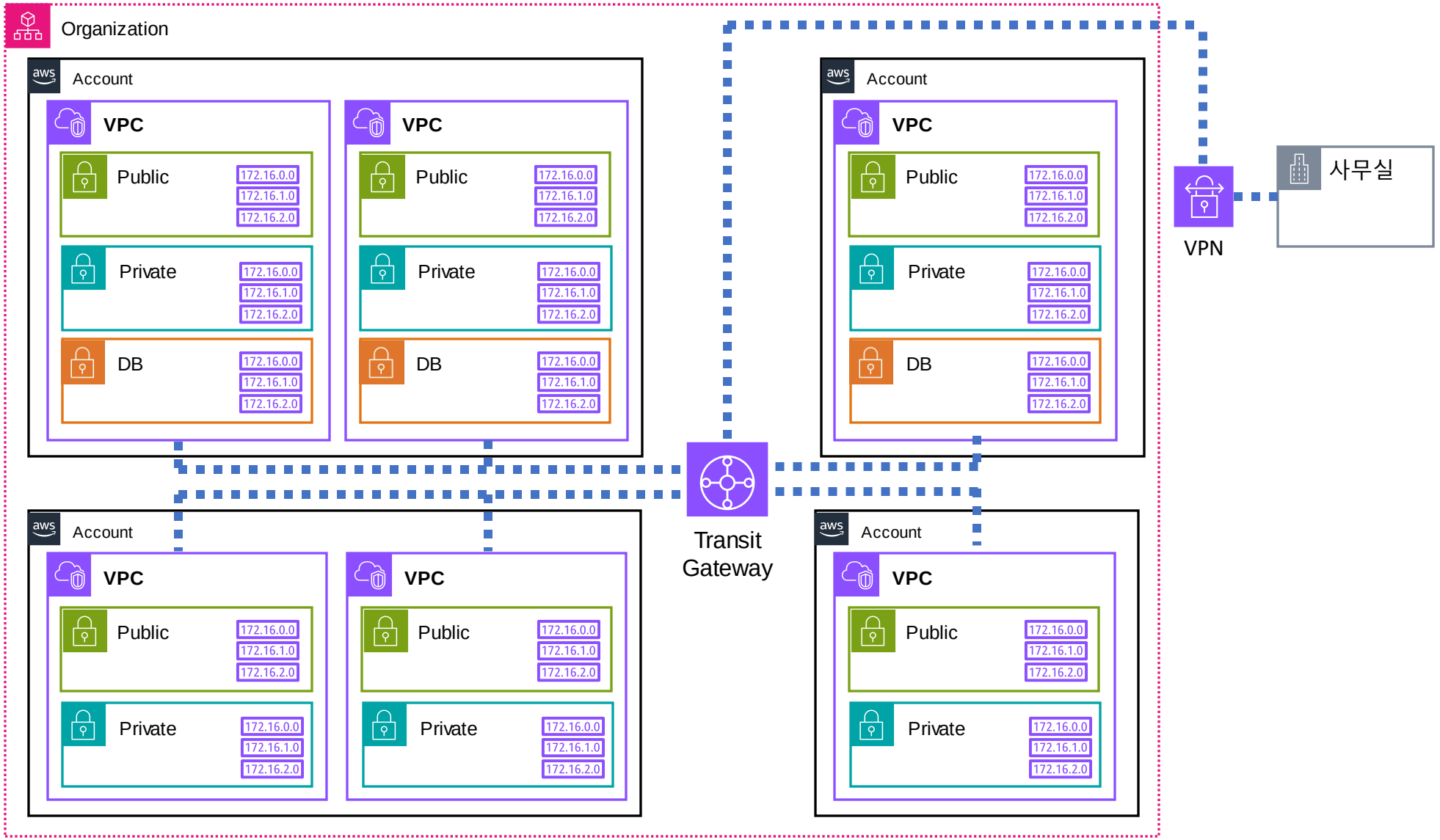
네트워크 성장 과정: 증가하는 VPC와 연결 구간

AWS 계정과 VPC가 증가하고 외부연결이 추가되면서 연결 구간이 복잡해짐



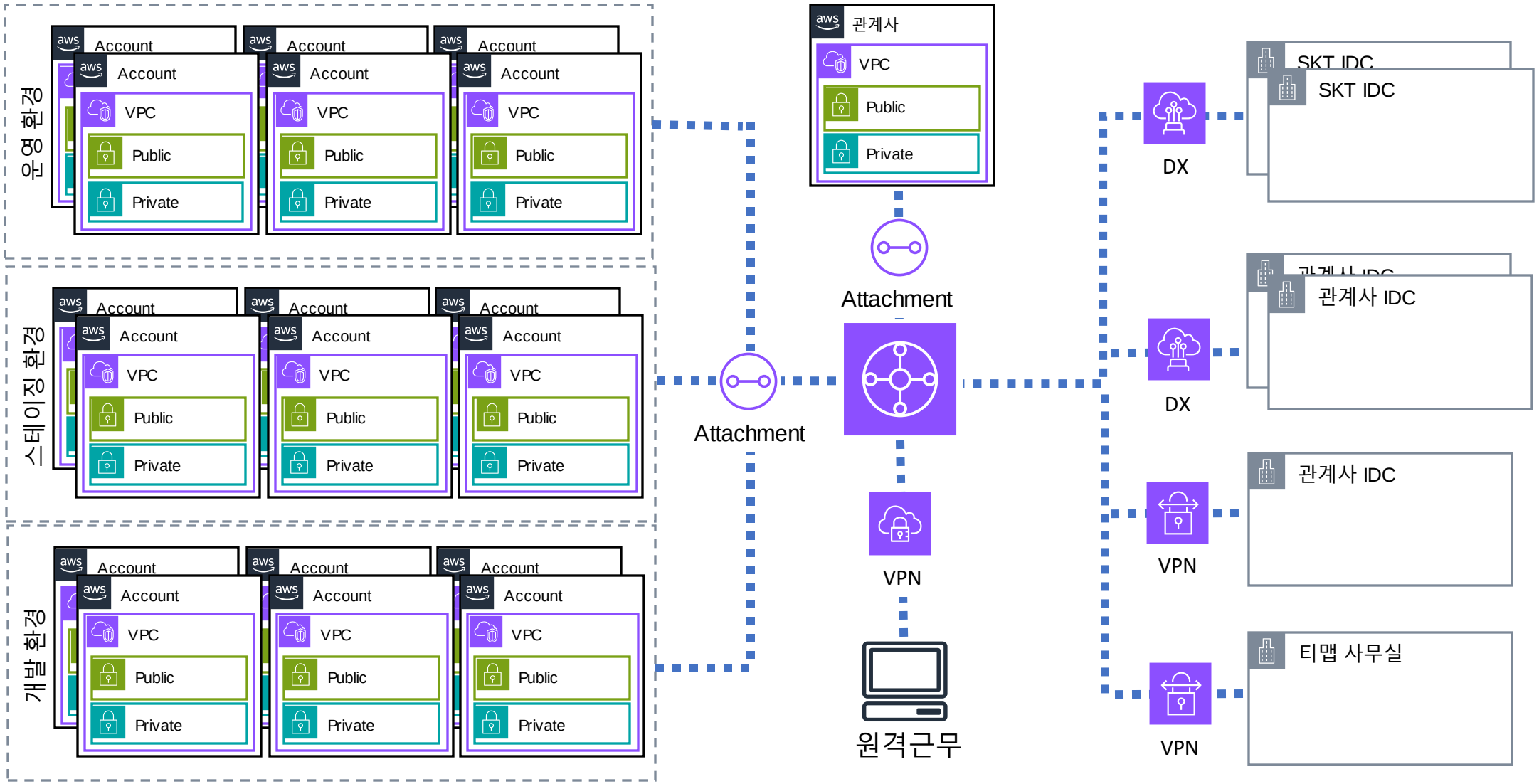
네트워크 성장 과정: 통합 관리 방안

Organization과 TransitGateway를 통한 통합 관리 방안



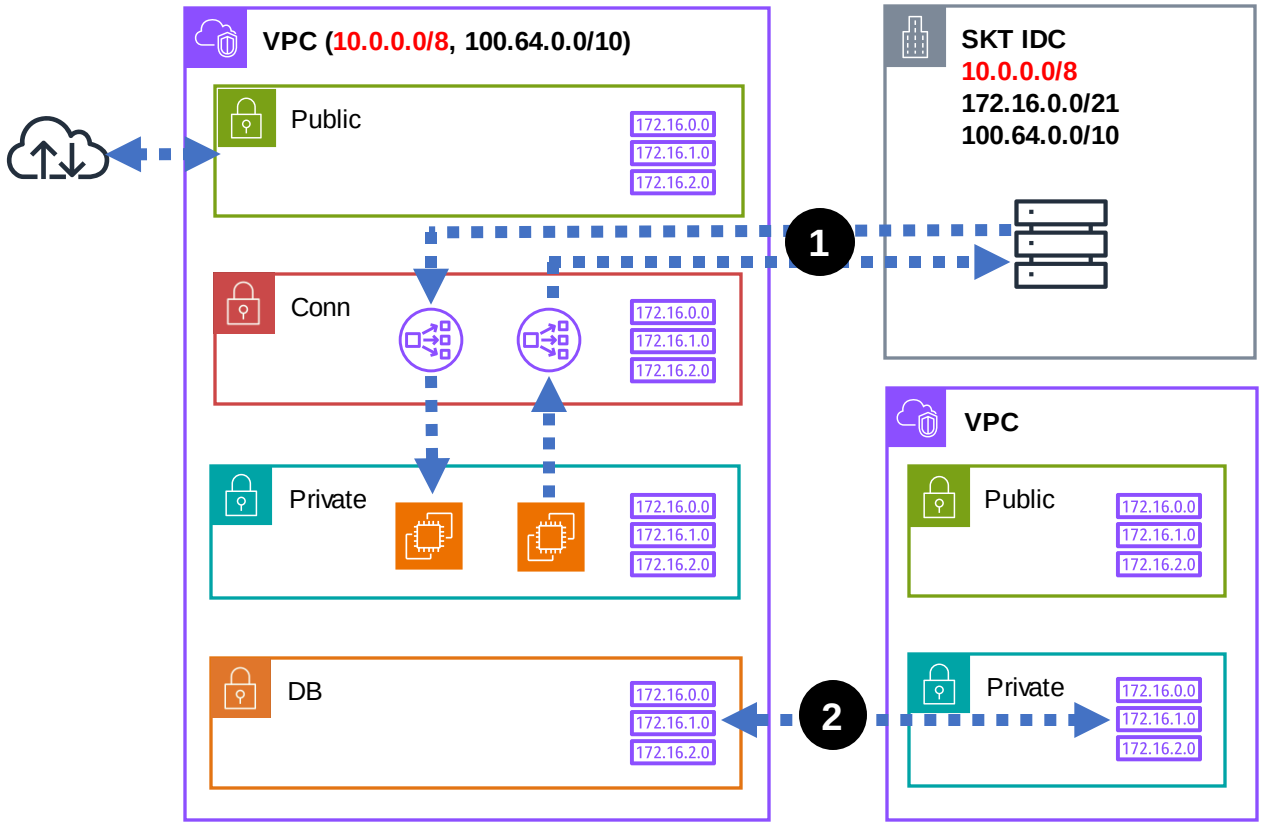
티맵 네트워크 구성: 전체적인 구조

TransitGateway를 중심으로 다수의 AWS 계정과 VPC, SKT/관계사의 IDC, 사무실 등 연결



티맵 네트워크 구성 : 연결 구간

내부 네트워크 구간별 연결 방안



1 SKT IDC 연결

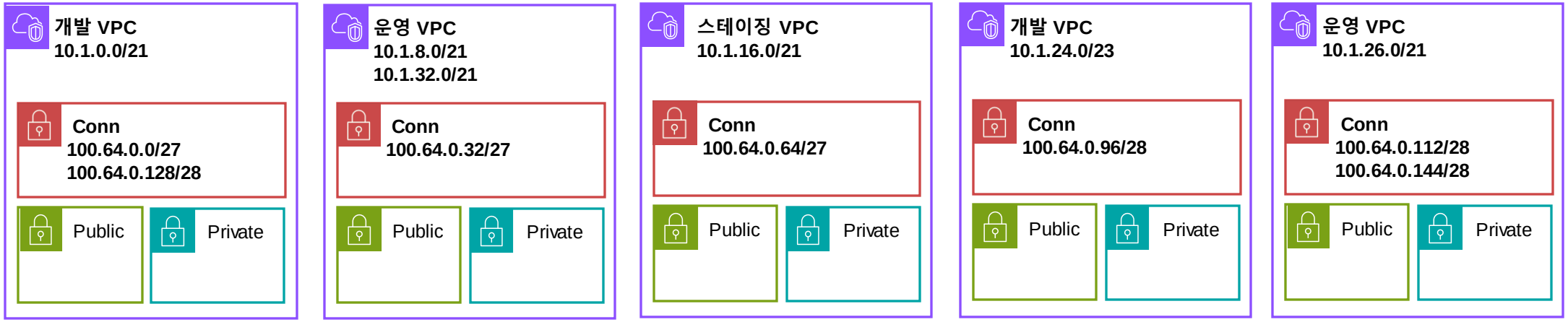
- SKT IDC와 겹치는 아이피 대역간 통신 필요
- 협의된 100.64.0.0/10 대역에 NLB를 구성
- 한정된 대역을 /27, /28로 나누어 사용

2 VPC 연결

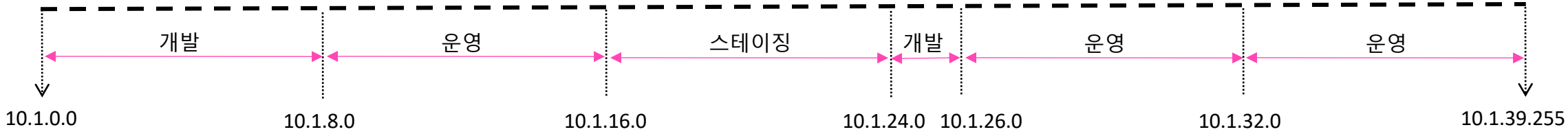
- 개발<>운영간 트래픽 분리 필요
- VPC 라우팅 테이블로 트래픽 제어
- Subnet 단위로 라우팅 규칙이 상이함

문제점: 아이피 대역 관리

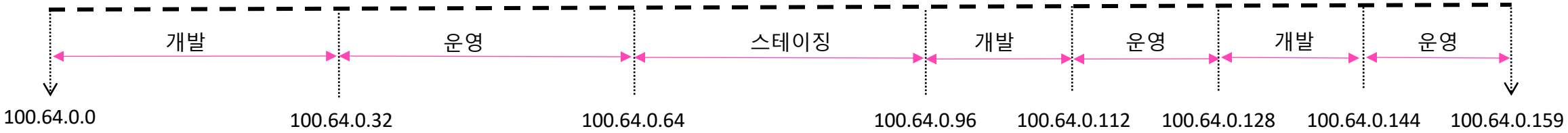
아이피 대역만으로 환경 예측이 어려움



10LAN 대역의 아이피 대역



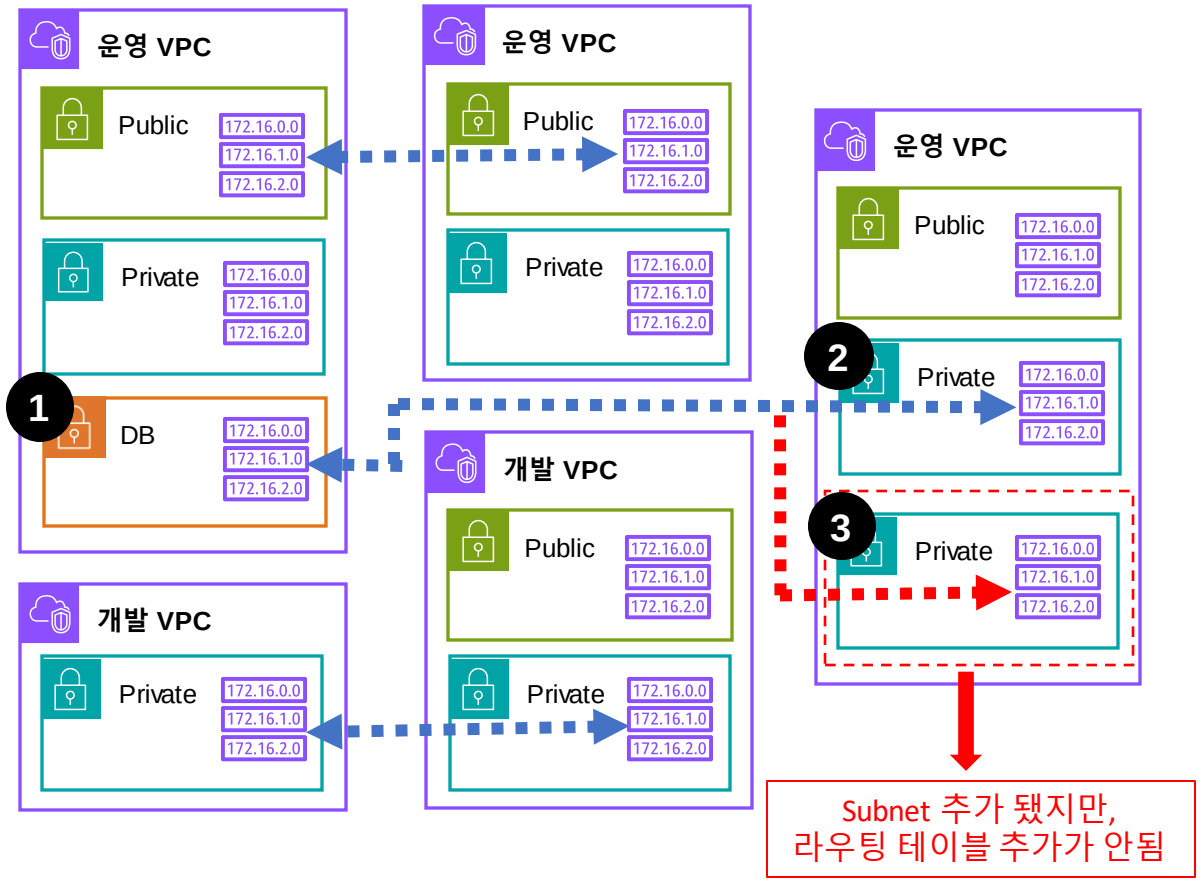
100LAN 대역의 아이피 대역



문제점: 복잡한 VPC 라우팅 테이블

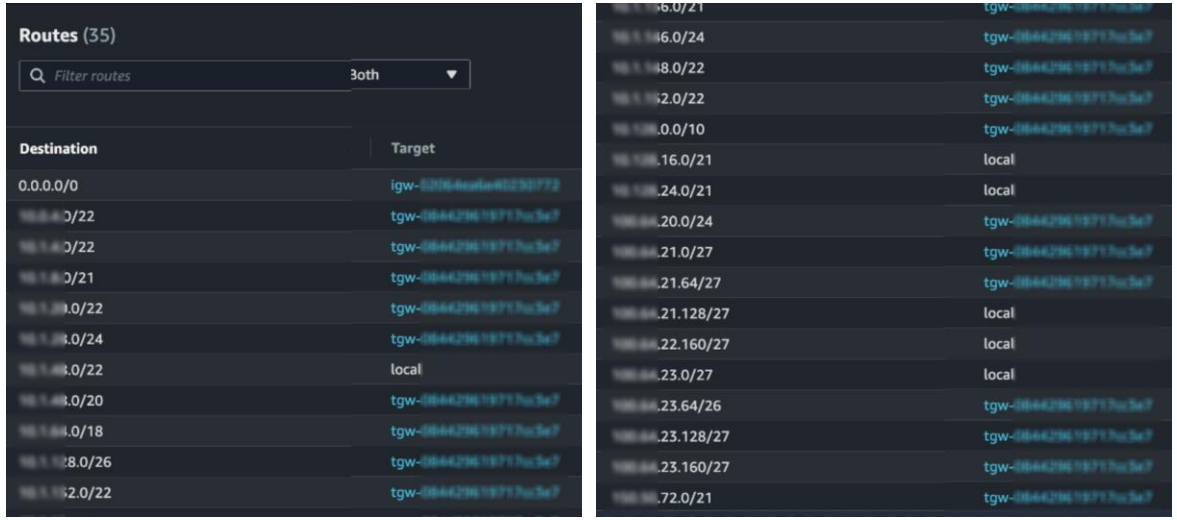
개발<>운영 환경 트래픽 분리를 위해 VPC 라우팅 테이블 기반으로 트래픽 제어

개발간, 운영간 트래픽 제어



- VPC 증가로 라우팅 테이블 복잡도가 올라감
- 다수 AWS 계정으로 인한 여러 계정을 돌아다니며 설정 필요
- 신규 Subnet 생성 시, 라우팅 테이블 반영이 안되는 경우가 생김
- 라우팅 테이블의 아이피 대역만으로 환경 확인이 어려움

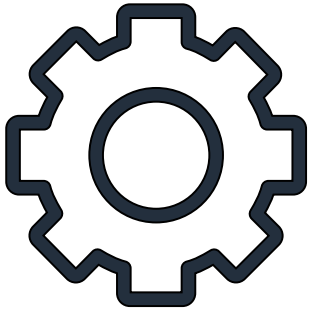
이런 라우팅 테이블이 수 백개 존재



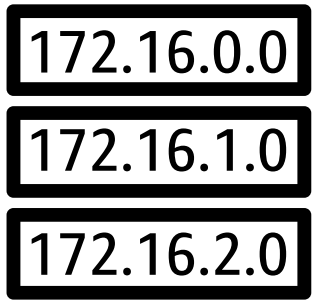
개선 요구사항

기존 구조를 개선하기위해 다음과 같은 요구사항을 도출함

Infrastructure as Code

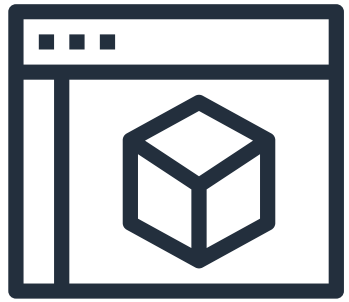


IP 대역 가시성 확보



라우팅 테이블 단순화

외부 인입 트래픽 제어



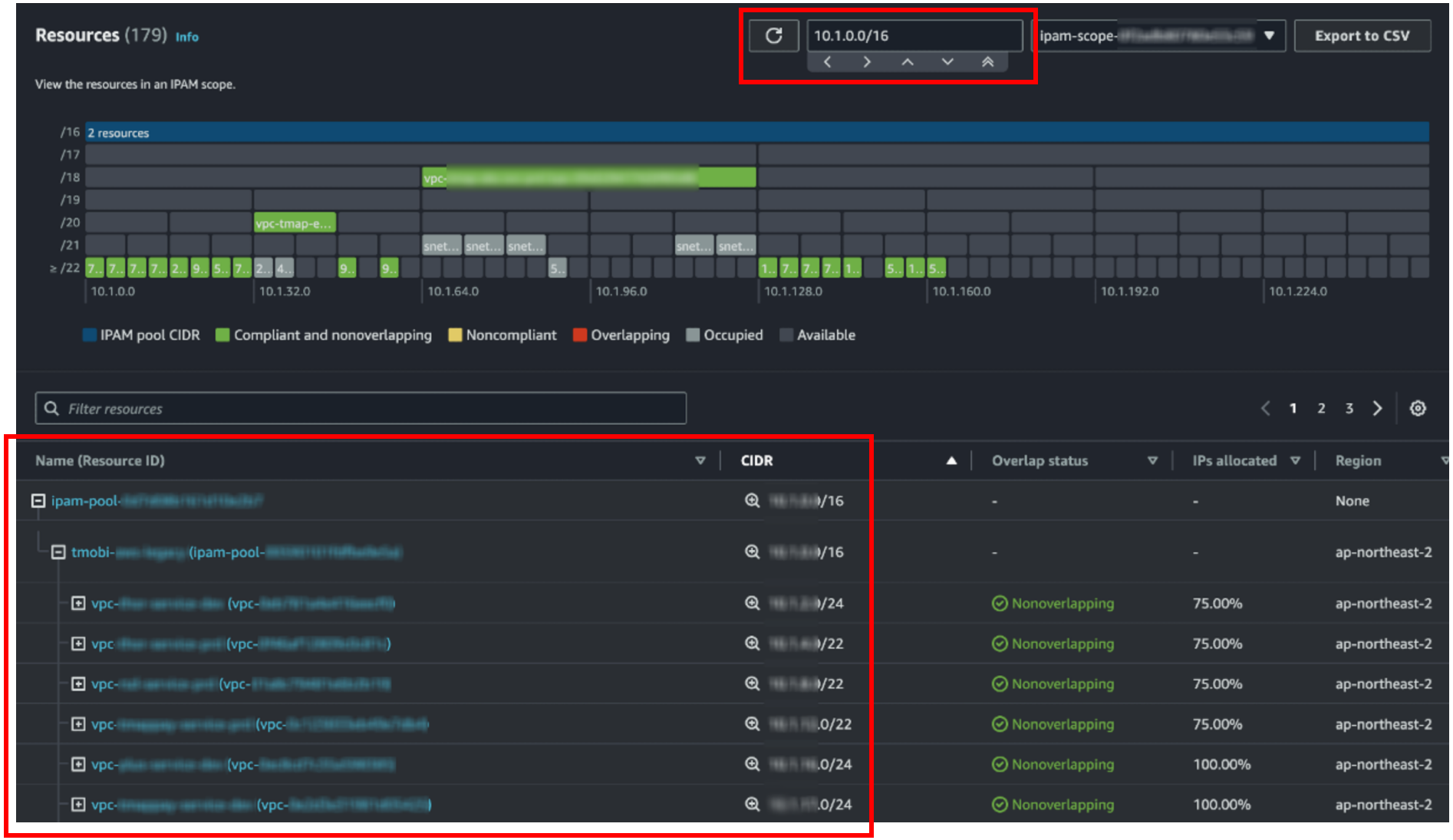
중앙 집중화 관리

작업 상세

1. 사전 작업
2. 반영 작업

사전 작업: IP 대역 가시성 확보

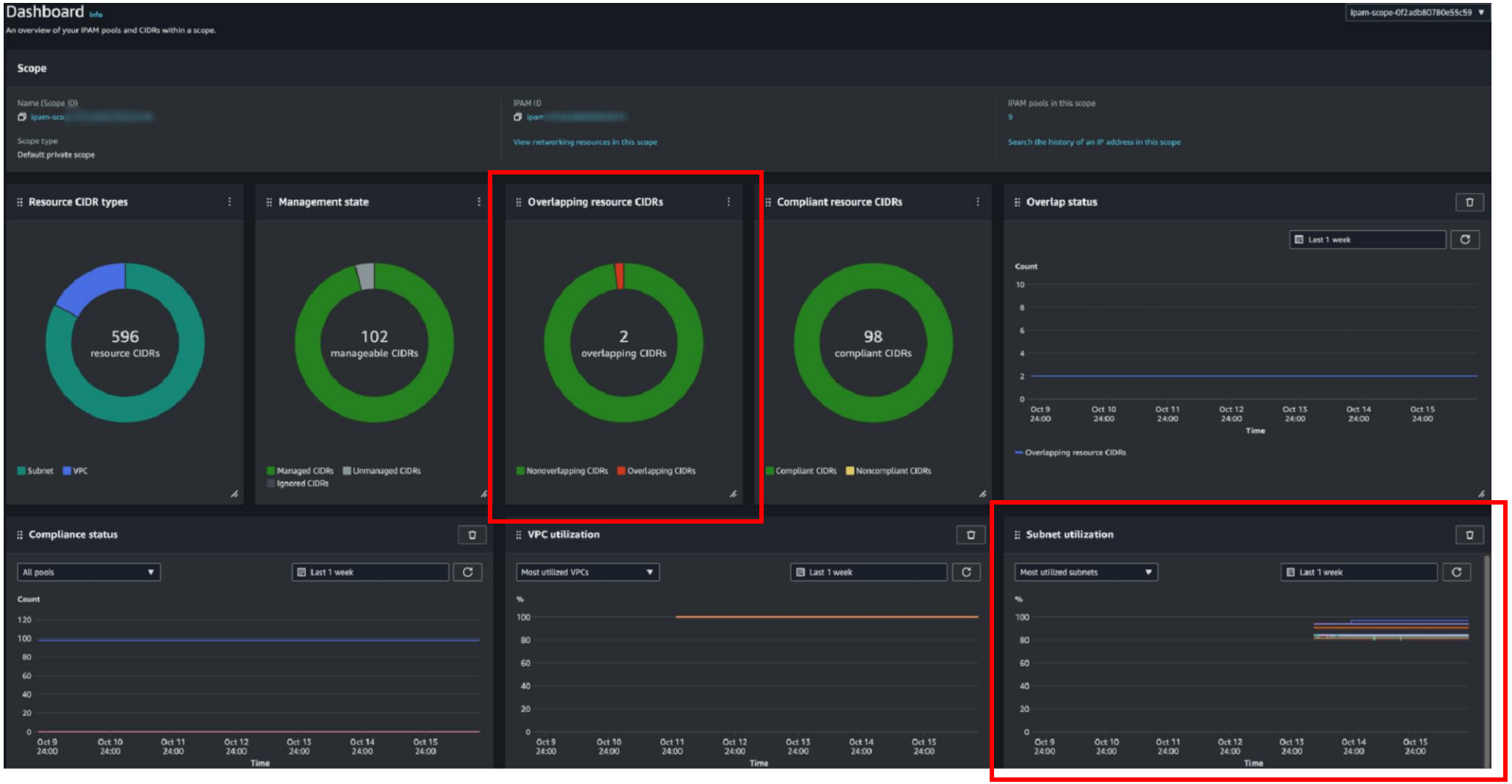
IPAM 기능을 이용해 IDC, AWS 네트워크 대역을 한 군데서 확인하고 가시성 확보



참고: https://docs.aws.amazon.com/ko_kr/vpc/latest/ipam/what-it-is-ipam.html

사전 작업: IP 대역 가시성 확보

관리되지 않는 미사용 VPC 및 실제 IP 사용률 확인



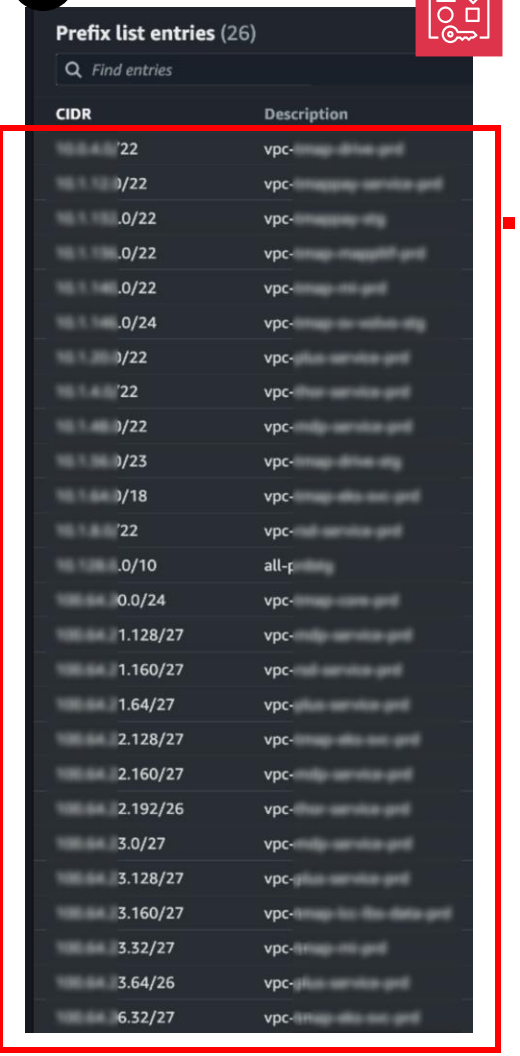
참고: https://docs.aws.amazon.com/ko_kr/vpc/latest/ipam/what-it-is-ipam.html

사전 작업: VPC 라우팅 테이블 정리

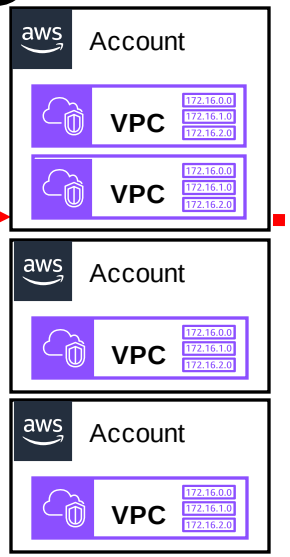
환경별로 Managed IP Prefix List 기반으로 아이피 대역 통합 관리

1 변경 전 VPC 라우팅 테이블

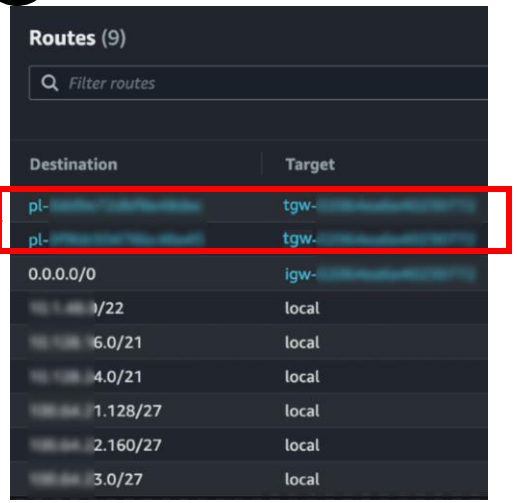
2 환경별로 대역 설정



3 다수 계정 전파



4 변경 후 VPC 라우팅 테이블

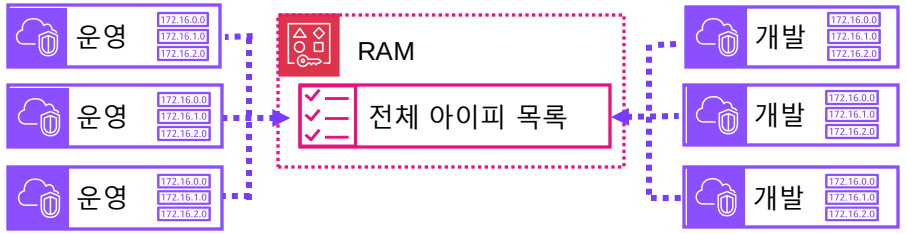


- IPAM Resource Discovery 기반으로 IP 대역 취합
- 개발/운영 등 환경별로 Managed IP Prefix List 생성
- RAM을 이용해 다수 계정의 VPC에 한 번에 변경 전파

참고: https://docs.aws.amazon.com/vpc/latest/userguide/working-with-aws-managed-prefix-lists.html

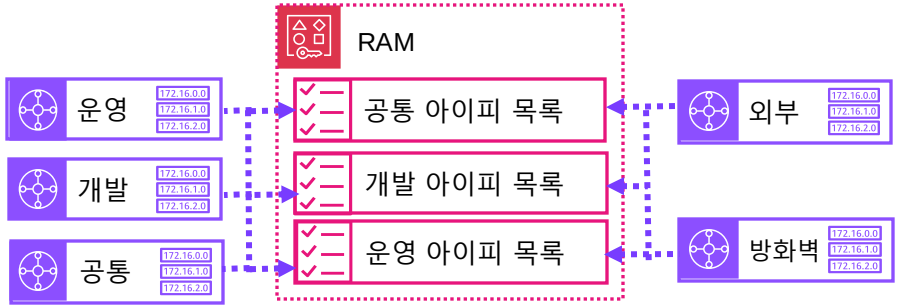
사전 작업: Managed IP Prefix List

Managed IP Prefix List의 티맵 사용 사례 소개



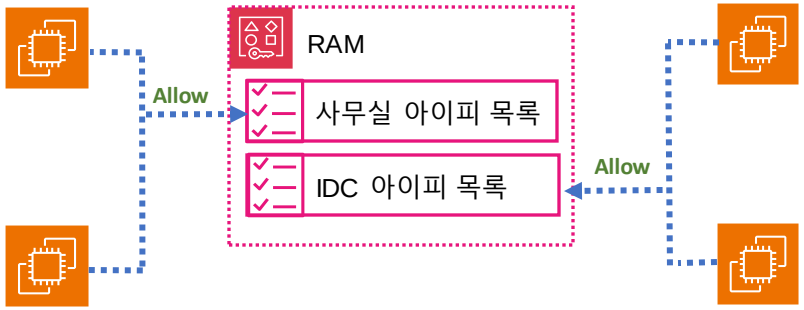
1 전체 아이피 목록을 TGW로 라우팅

VPC 라우팅 테이블 규칙



2 개발<>운영간 트래픽 제어

TransitGateway 라우팅 테이블 규칙

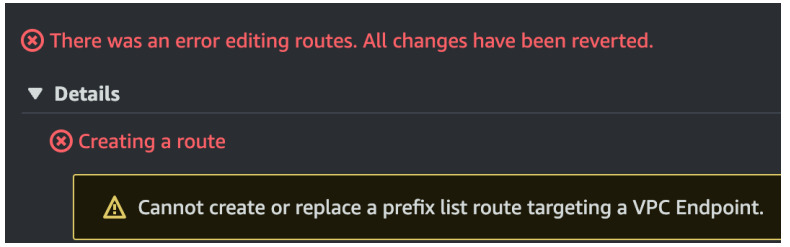


3 공인 아이피로 접근 허용 필요시

SecurityGroup 규칙

TIP: 주의 할 점

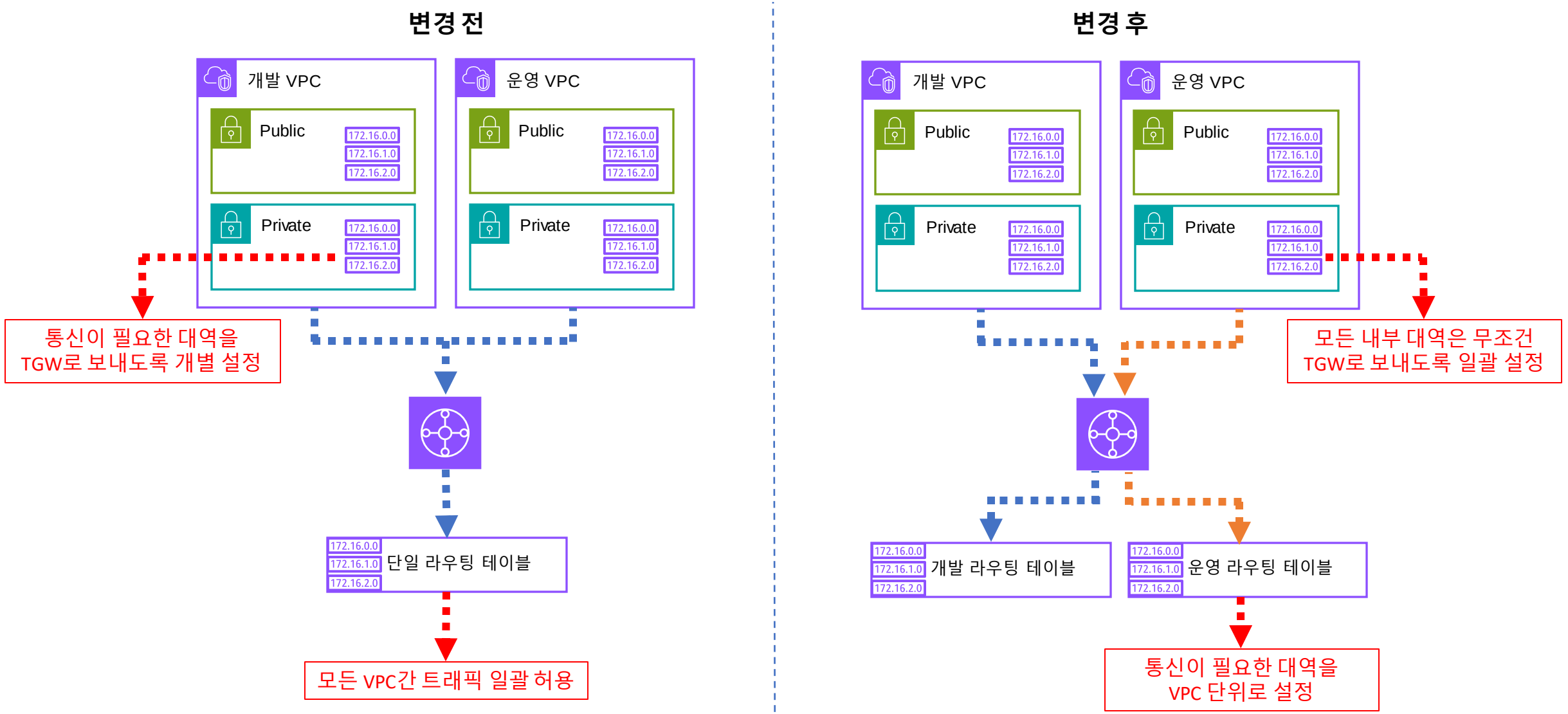
- 사소한 변경이 전체 계정의 장애로 이어질 가능성이 있음
- 라우팅 테이블에 사용 시, Prefix List 우선 순위 확인 필요
- GWLB VPC Endpoint를 Destination으로 설정 불가



참고: https://docs.aws.amazon.com/vpc/latest/userguide/working-with-aws-managed-prefix-lists.html

사전 작업: TGW 라우팅 테이블 분리

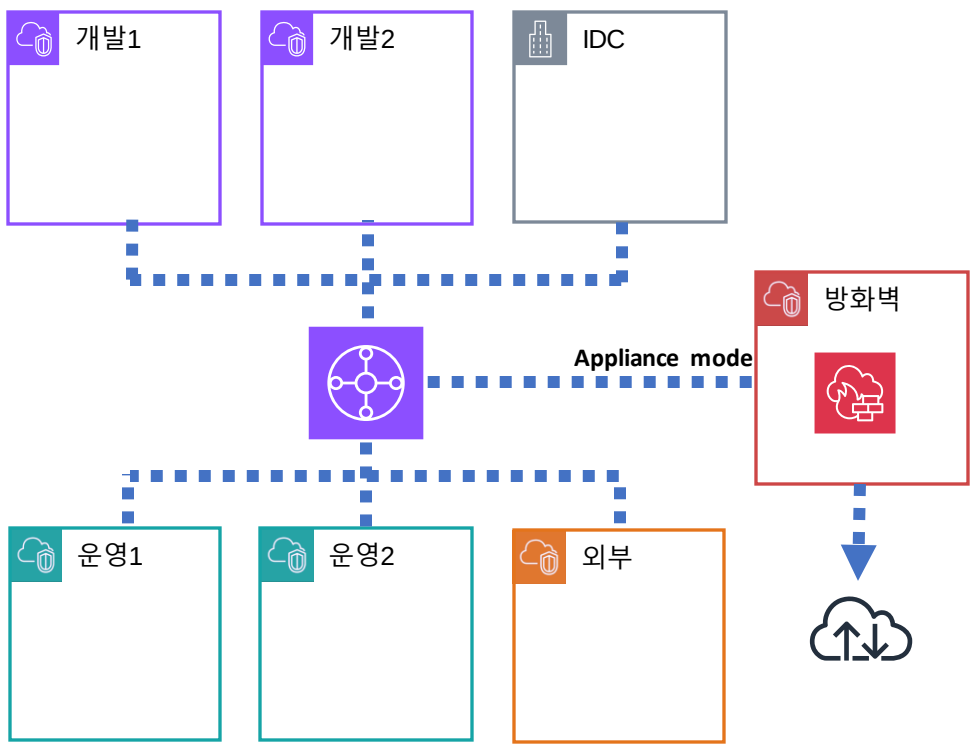
단일 TransitGateway 라우팅 테이블을 환경별로 나누어 구성



사전 작업: TGW 라우팅 테이블 분리

TransitGateway 라우팅 테이블과 AWS Network Firewall을 이용해 트래픽 제어

네트워크 구조도



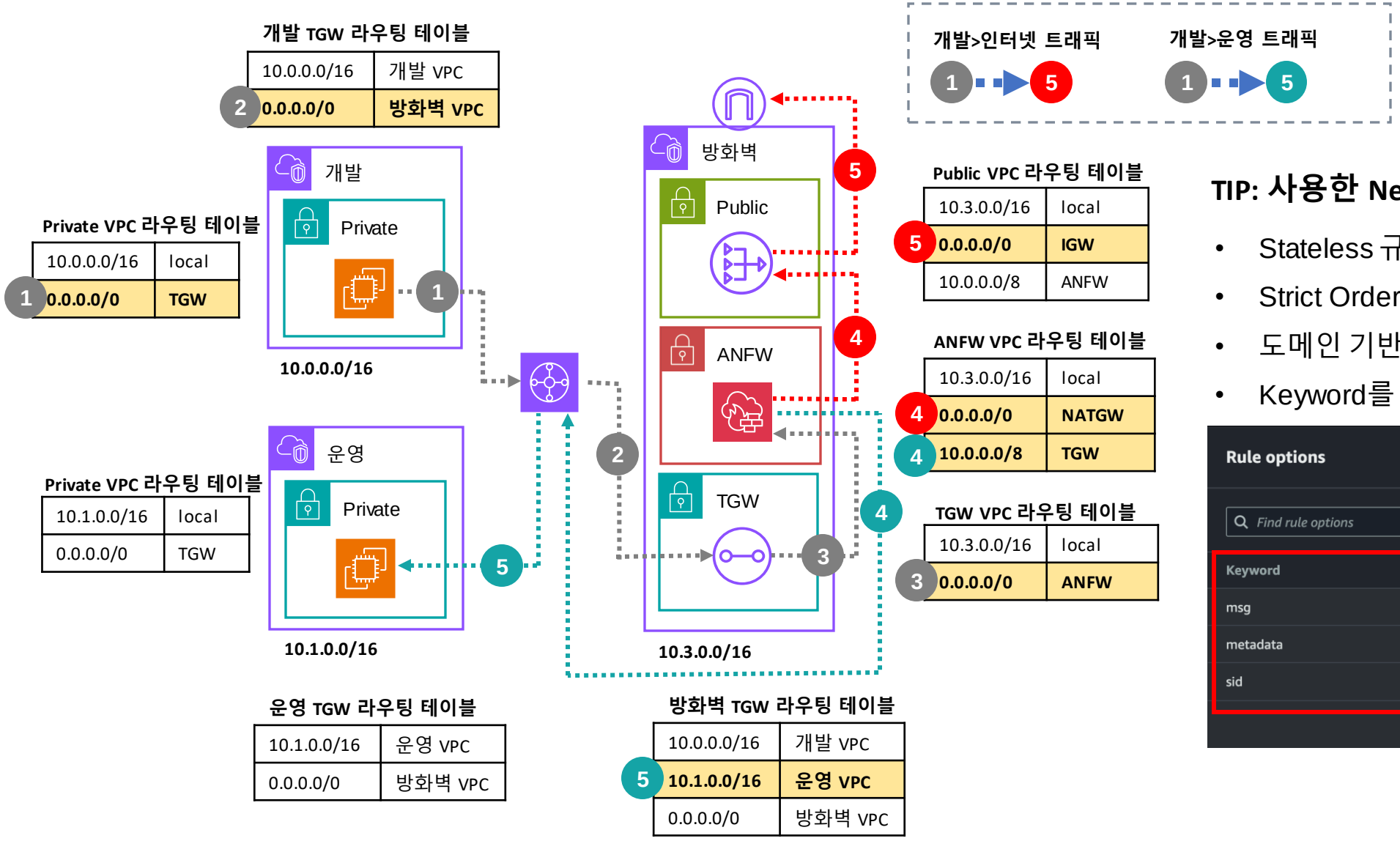
- 같은 환경 VPC간의 요청은 바로 통신
- 다른 환경 VPC간의 요청은 방화벽 통하도록 구성
- 외부 환경에서 들어오는 요청은 방화벽 통하도록 구성
- 인터넷망으로 나가는 요청은 방화벽 통하도록 구성

트래픽 흐름 예시

참고: https://aws.amazon.com/ko/blogs/networking-and-content-delivery/deployment-models-for-aws-network-firewall/

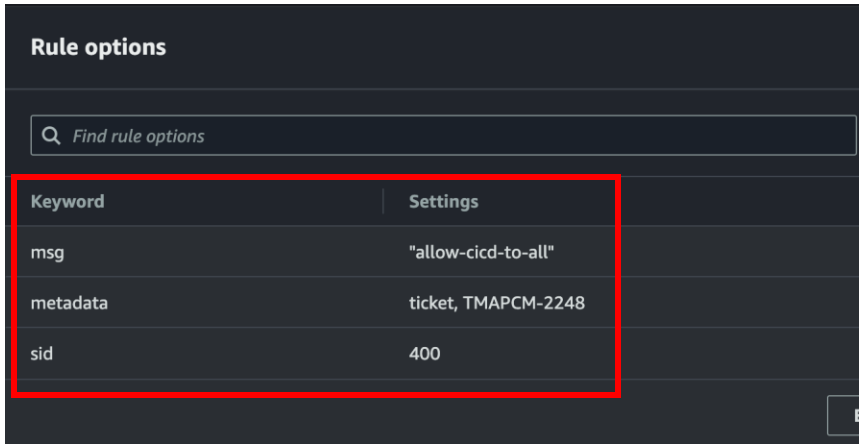
사전 작업: 방화벽 VPC 구성

TransitGateway 라우팅 테이블과 VPC 라우팅 테이블 구성 및 트래픽 흐름 예시



TIP: 사용한 Network Firewall 옵션

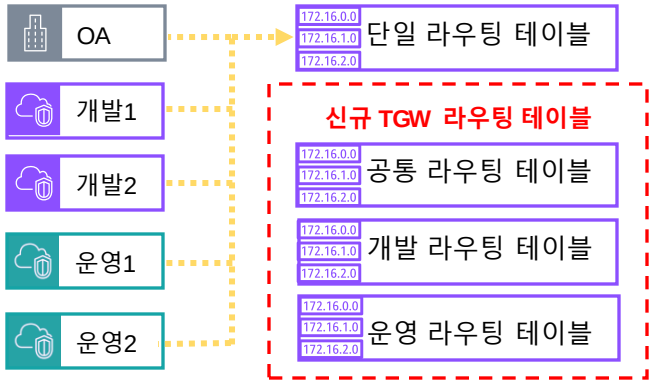
- Stateless 규칙은 Forward
- Strict Order 사용
- 도메인 기반 규칙은 Suricata rule 사용
- Keyword를 작업 메모 용도로 활용



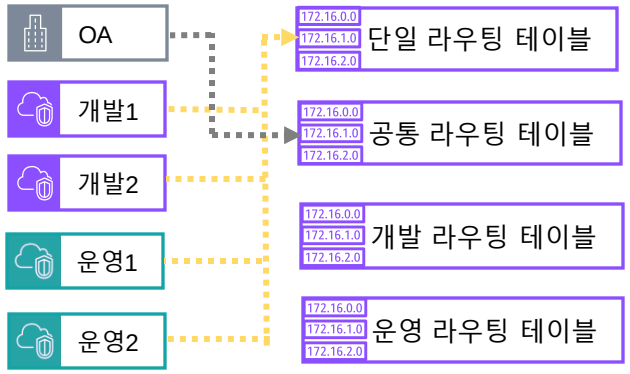
반영 작업: TGW 라우팅 테이블 변경

실제 서비스 트래픽 흐름이 변경되는 작업

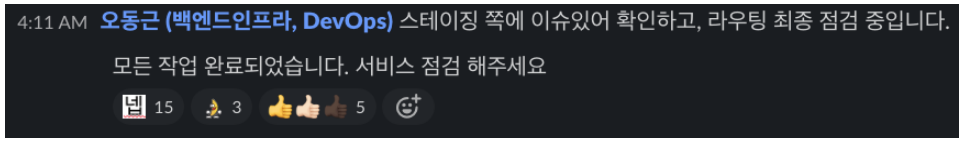
1 사전 준비 완료



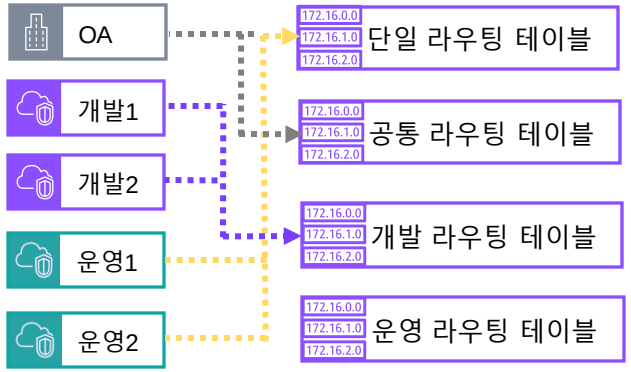
2 오피스 TGW 라우팅 테이블 변경



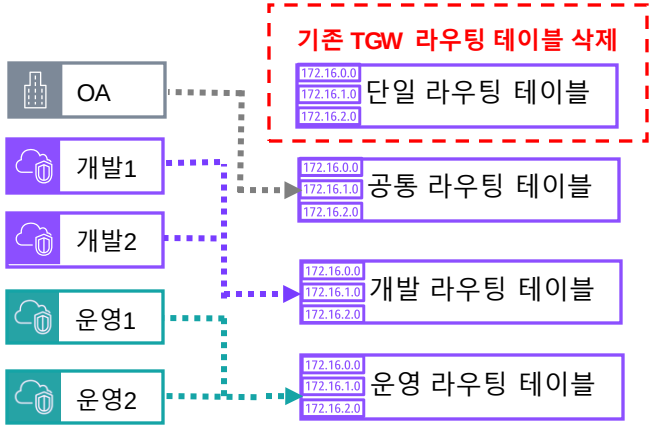
- TGW Attachment의 라우팅 테이블 변경 작업
- 변경 시, 약 40초 정도 트래픽 중단 발생
- 서비스 트래픽 최빈 시간대 확인
- 중요도 낮은 OA, 개발 환경부터 서서히 트래픽 이전
- 운영 환경까지 최종 작업 완료



3 개발 TGW 라우팅 테이블 이전



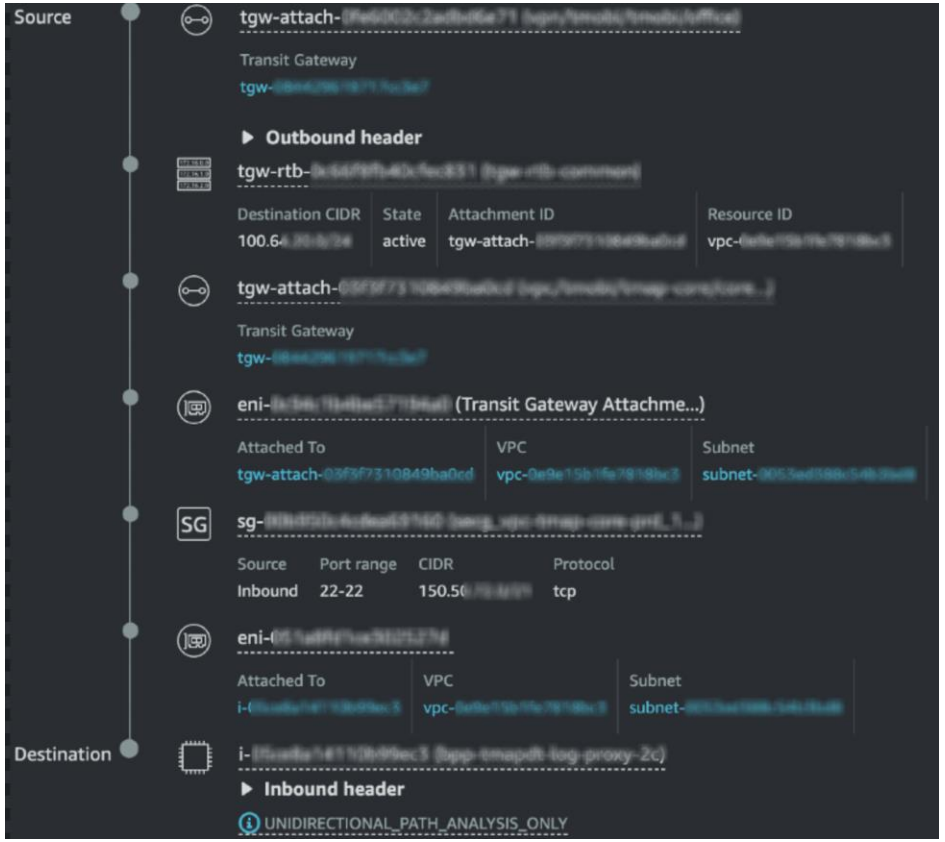
4 운영 TGW 라우팅 테이블



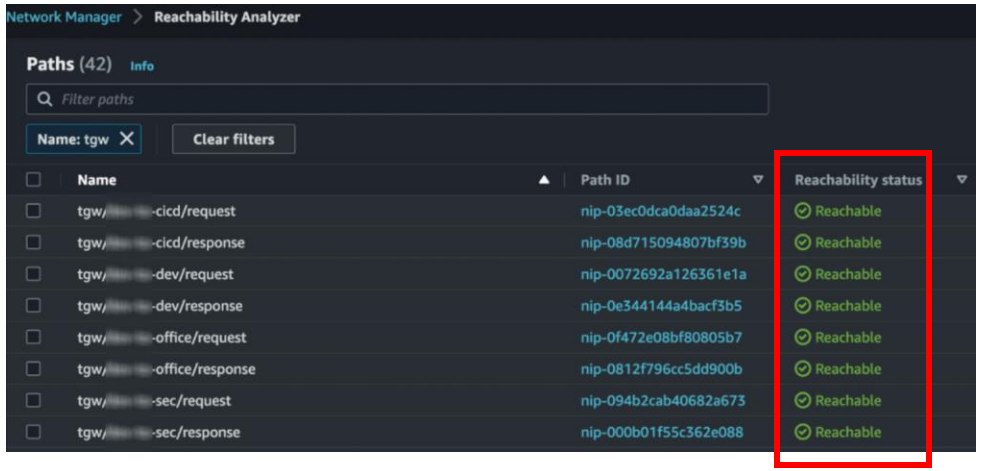
반영 작업: 정상 여부 점검

미리 구성해놓은 Reachability Analyzer로 TGW 라우팅 정상 여부 점검

네트워크간 라우팅 경로 검사



실제 화면



TIP: 고려 할 점

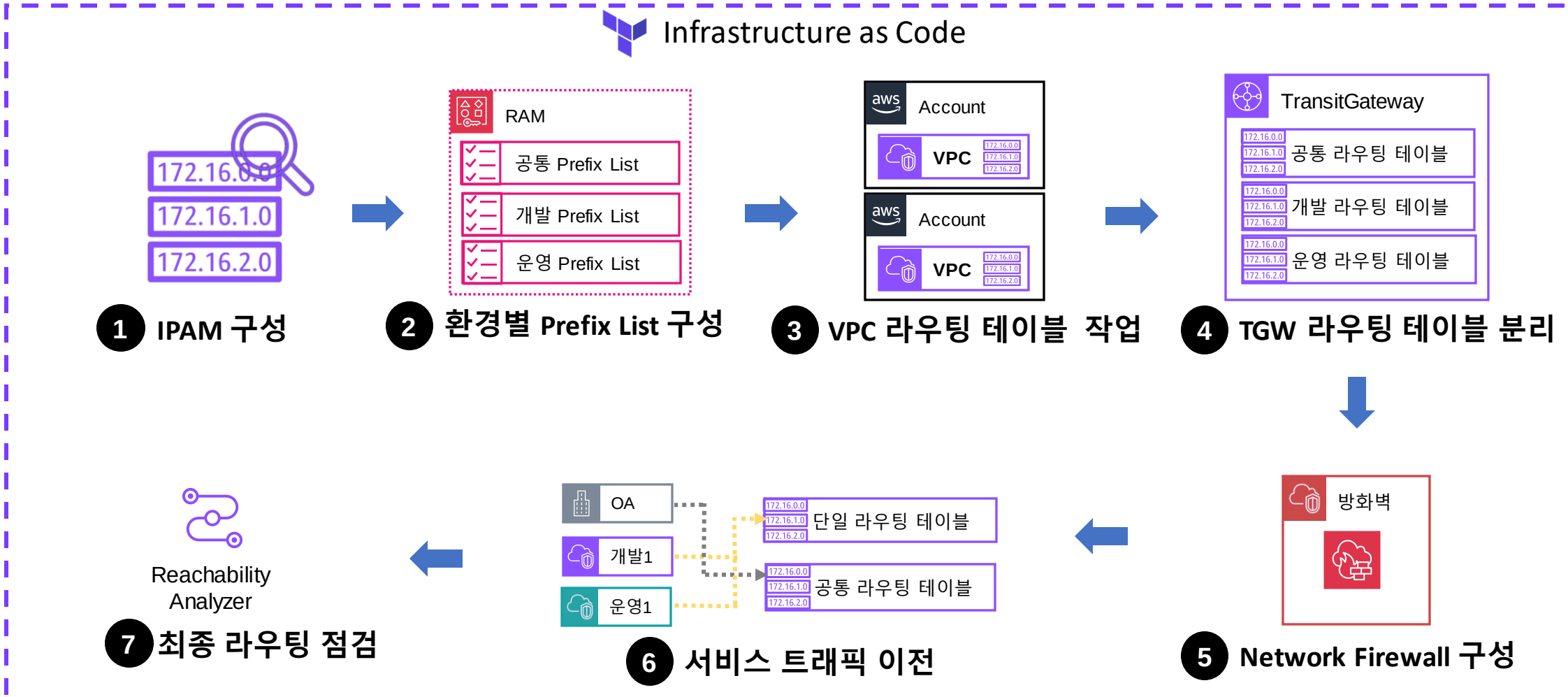
- TGW 고려시 단방향으로 동작하므로 양방향 확인 필요
- 요청 후 상태 확인까지 약 4분정도 소요됨
- GUI 편의성이 떨어져, CLI 및 스크립트 사용 권고
- Organization 기반 다중 계정 환경에서는 권한 위임 설정 필요

참고: https://docs.aws.amazon.com/vpc/latest/reachability/what-is-reachability-analyzer.html

마무리

마무리: 작업 요약

모든 작업은 Terraform을 통해 GitOps 파이프라인 구성



E.O.D

TMAP MOBILITY