26년 상반기 현대오토에버 공개채용 포트폴리오

IT Architect 직무

지원자 조유성

조유성 Cloud Enginner

신입 클라우드 엔지니어 지망생 조유성입니다. 지속적인 학습과 실습을 통해 변화하는 클라우드 환경에 빠르게 적응하는 엔지니어가 되겠습니다.

Profile

010-9913-6659

cys990617@gmail.com

github.com/whdbtjd

velog.io/@whdbtjd/posts

Skills

- AWS
- Terraform
- Docker
- Kubernetes
- Python

PROJECT

대규모 트래픽 대응 인프라 구축 프로젝트
- 전체 아키텍처 설계
- DR 환경 구성
- 비용 관리

서버리스 뉴스 애플리케이션 개발
- 서버리스 기반 아키텍처 설계
- Frontend/Backend 풀스택 개발
- 뉴스 데이터 수집·처리 파이프라인 설계

EDUCATION

성공회대학교 소프트웨어 공학과 / 학사 / 2020 - 2026.02
- 전체 학점 3.93 / 4.5
- 전공 학점 3.98 / 4.5
- 입학 전체 수석

CERTIFICATIONS

- AWS Certified Security – Specialty / 2026.03
- AWS Certified Solutions Architect – Associate / 2025.05
- AWS Certified SysOps Administrator – Associate / 2025.06
- AWS Certified Cloud Practitioner – Foundational / 2025.03
- HashiCorp Certified: Terraform Associate / 2026.01

TRAINING

[CJ 올리브네트웍스] CloudWave 6기 / 2025.06 ~ 2025.09
- 네트워크 · 리눅스
- Docker·Kubernetes 컨테이너 기초
- Terraform·Ansible 기반 인프라 자동화
- AWS 인프라 구축 프로젝트 수행

ACTIVITIES

GDG on SKHU / 2024.09 ~ 2025.03
- 대학생 연합 개발 동아리에서 Backend 파트 멤버로 활동

팝콘 / 2024.03 ~ 2024.12
- 영화 관람 소모임에서 활동

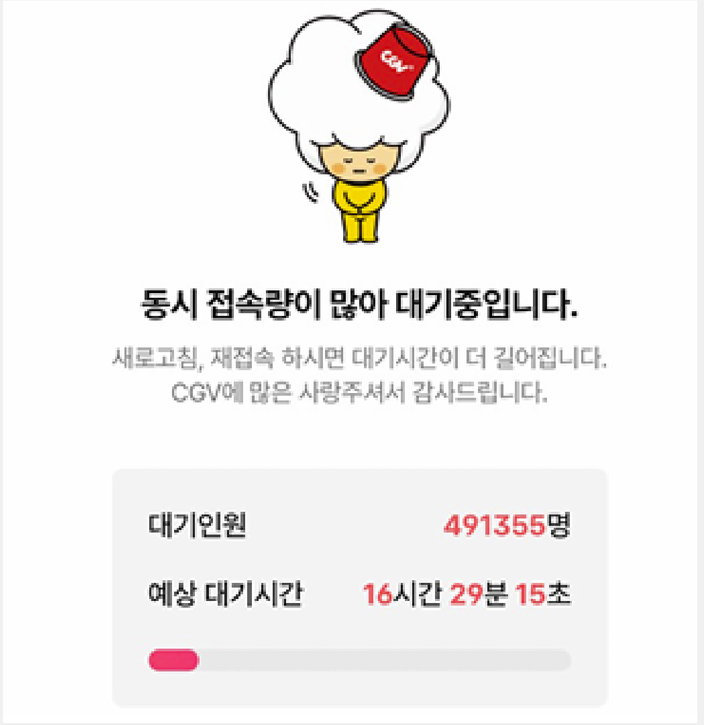
01 프로젝트 개요

HYUNDAI AutoEver

문제 상황

할인 정책 시행 후 트래픽 급증 !!

실제 서비스 장애 당시 화면



[프로젝트 정보]

- **프로젝트명**: 대규모 트래픽 대응 예매 서비스
- **구성**: 팀 프로젝트 (6명)
- **기간**: 2025.08 ~ 2025.09 (3주)
- **목적**: 안정적인 클라우드 인프라 구축
- **기여도**: 60%

[기여한 부분]

- 전체 아키텍처 설계
- 재해복구 환경 구축
- 아키텍처 구성도 업데이트
- 비용 관리

[사용 기술]

- **AWS**: EC2, NLB, RDS/Aurora 등
- **IaC**: Terraform
- **diagram**: draw.io

02 설계 원칙

HYUNDAI AutoEver

AWS Well-Architected

[성능 최적화]

- 다양한 대안을 비교 후 서비스에 적합한 리소스 선택
- 트래픽 급증 상황을 고려한 안정적 요청 처리 구조

[비용 최적화 및 운영 효율 설계]

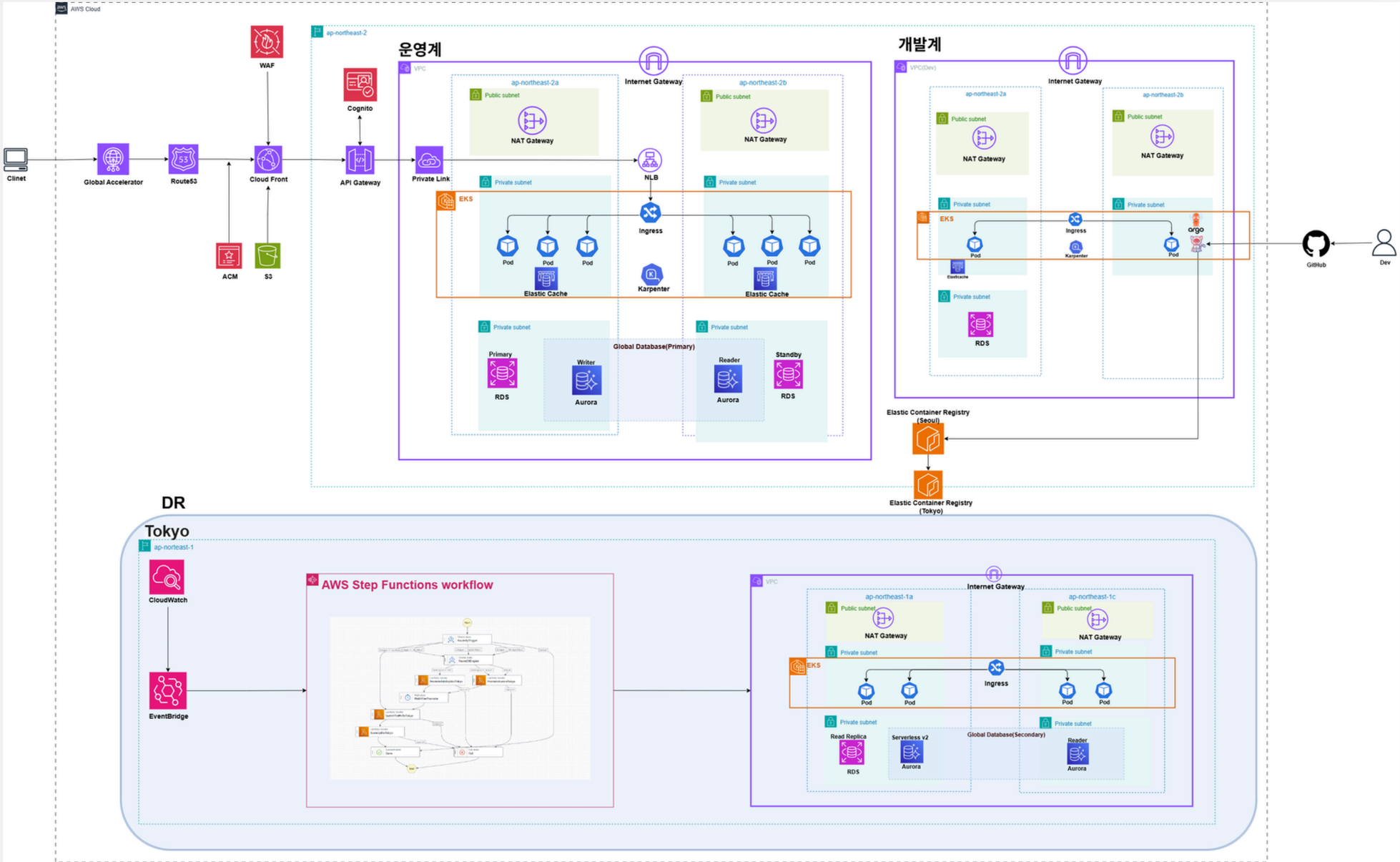
- 비용 대비 운영 편의성을 고려한 리소스 선택
- 자동화 기반으로 관리 효율 및 대응 속도 개선

[장애 대응 기반 구조 설계]

- 예상 가능한 장애 상황을 대응하기 위한 구조 설계
- 서비스 연속성을 위한 기본적인 안정성 요소 반영

03 전체 아키텍처

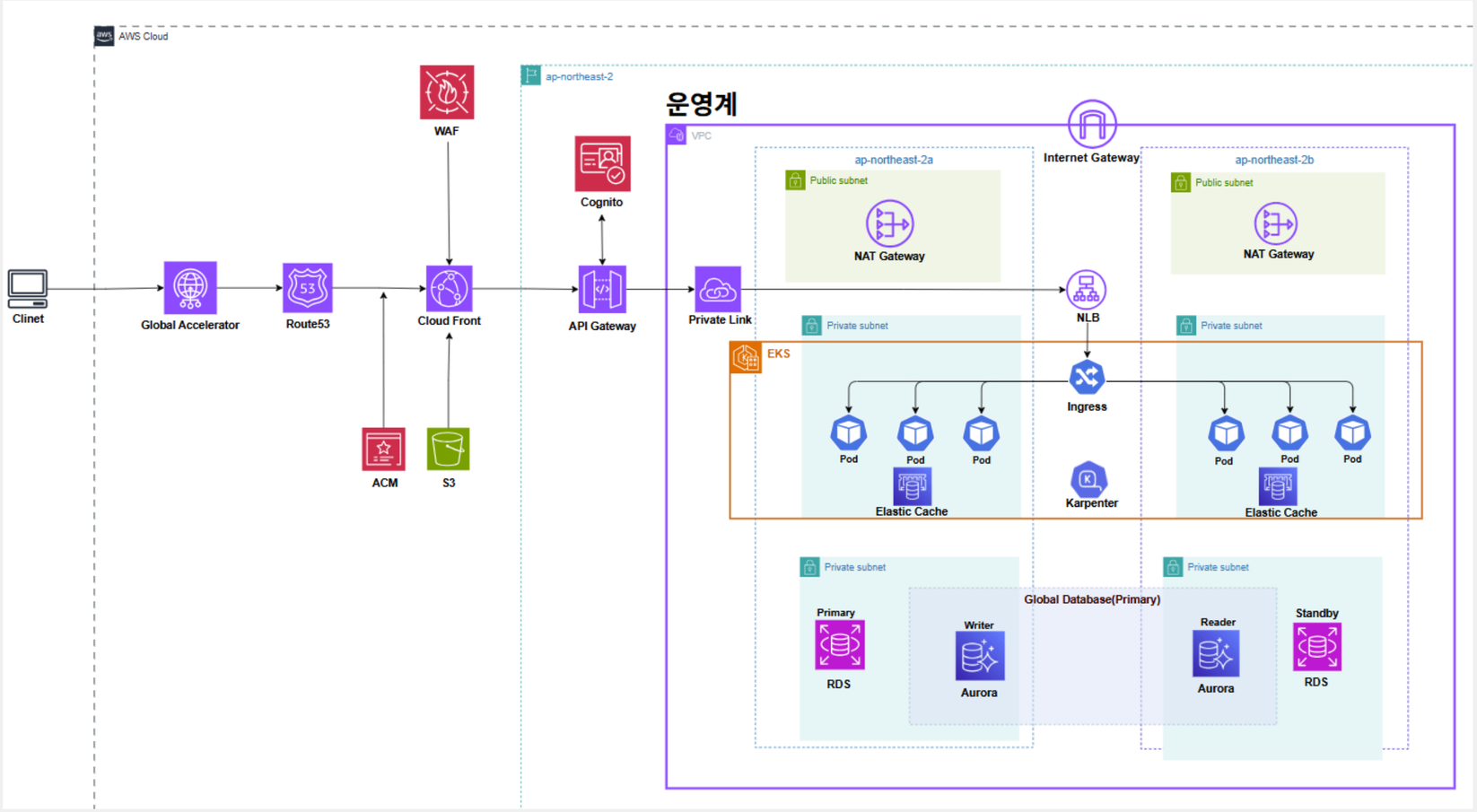
HYUNDAI AutoEver



[아키텍처 특징]

- 운영계 / 개발계 분리
- 멀티 리전 DR 구성
- 자동화 기반 장애 대응

04 세부 아키텍처 - 운영계



[아키텍처 특징]

- Global Accelerator 기반 저지연 트래픽 처리
- 정적/동적 트래픽 분리를 통한 효율적인 요청 처리 구조
- 트래픽 특성을 고려한 서비스별 DB 분리 구조
- WAF 기반 애플리케이션 공격 탐지·알림 구축

05 주요 리소스 - DB

HYUNDAI AutoEver

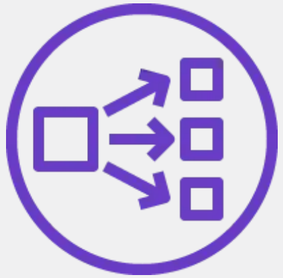
Aurora (티켓팅 svc)

- 리전 간 빠른 전환이 가능한 Aurora Global Database 를 적용
- 예매 오픈 시 트래픽 집중 특성을 고려해 높은 TPS 처리가 가능한 구조 선택
- 장애 발생 시 자동화된 Failover를 고려한 데이터베이스 구성

RDS (예매 정보 svc)

- 안정적인 트래픽 특성을 고려 해 Aurora 대신 RDS 선택
- 리전 내 장애 대비 Multi-AZ 구성 적용
- 보조 리전에는 Read Replica를 활용해 비용을 절감

06 주요 리소스 - ELB

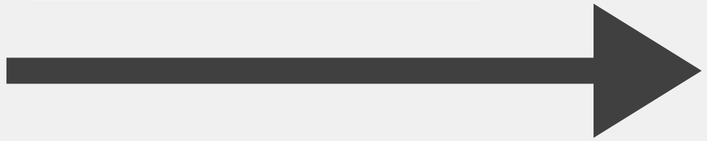


NLB

OR

ALB

API Gateway-EKS 연동 구조에 적합한 로드밸런서 선정 필요



EKS

[NLB 사용 시]

- VPC Link 추가 구성이 필요
- 높은 처리량과 낮은 지연 시간 제공
- 비용과 구성 측면에서는 ALB 대비↑

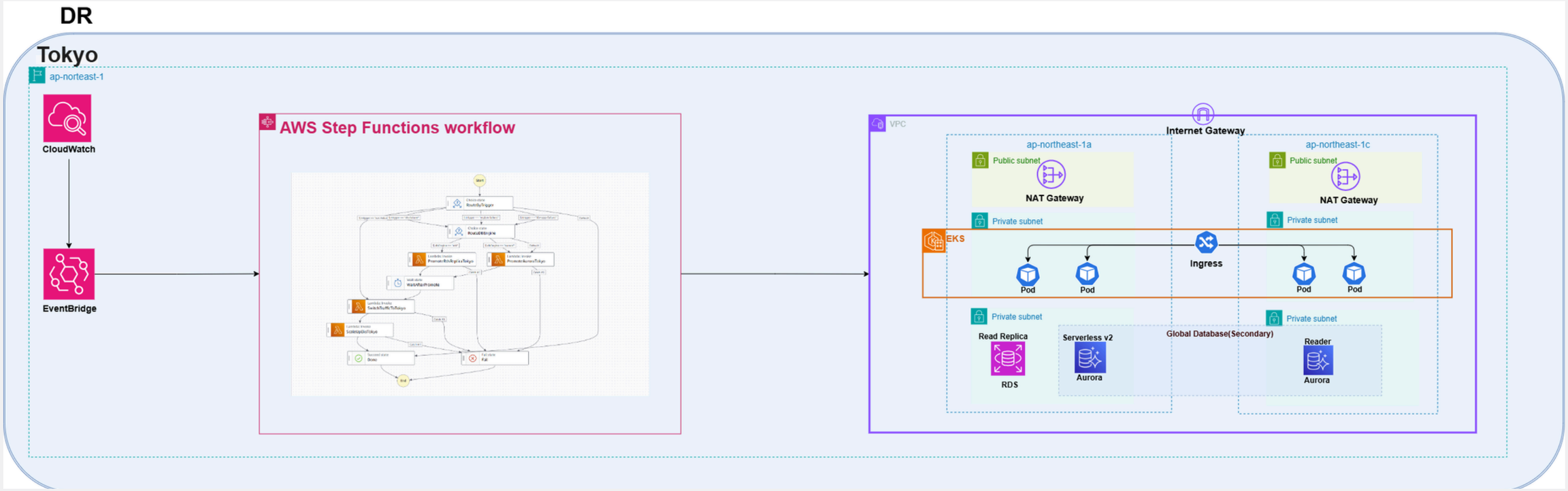
[ALB 사용 시]

- 유연한 트래픽 제어가 가능
- 고성능 단순 처리에는 비효율적
- 추가 리소스 없이 API Gateway와 직접 연동 가능

[NLB 선택 이유]

- L7 처리는 API Gateway, NLB는 프록시 역할로 분리
- ALB 사용 시 API Gateway와 일부 기능이 중복되어 불필요한 오버헤드 발생 가능

07 DR 아키텍처



[설계 목표]

- RTO 10분 이내를 목표로 DR 구조 설계
- 운영 자동화를 통해 개입 최소화
- 리전 장애 시에도 서비스 연속성 유지

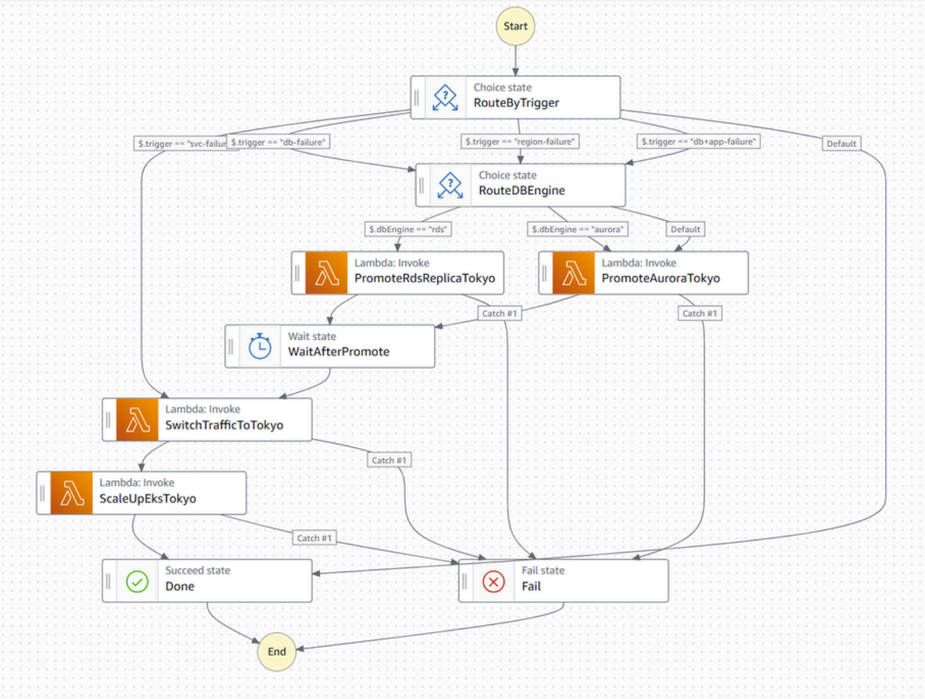
[아키텍처 특징]

- 서비스 및 DB 등 레벨별 장애를 고려한 Failover 구조 구축
- StepFuction기반 장애 조치 자동화
- GA 기반 즉각적인 트래픽 전환 구조

08 StepFunction기반 Failover 구성

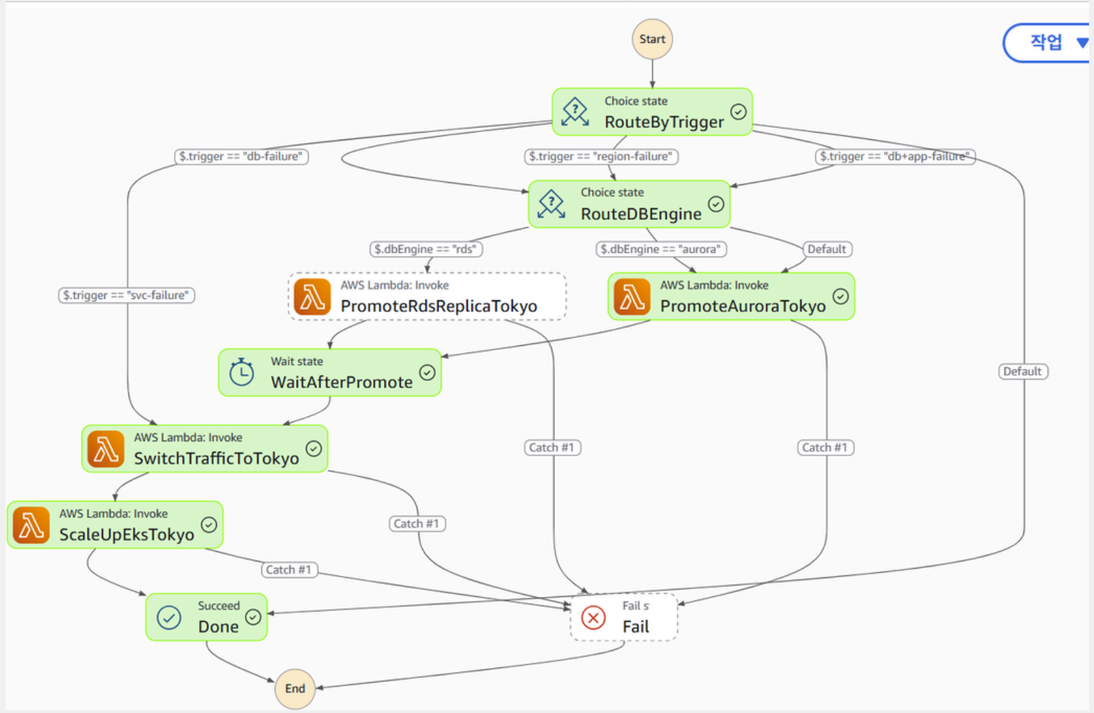
HYUNDAI AutoEver

StepFunction 상태함수



Aurora 장애 발생 시

Failover 실행 후



[워크플로우 개요]

- 장애 유형에 따라 분기되는 단일 DR Failover 워크플로우
- Step Functions를 중심으로 장애 대응 흐름을 오케스트레이션

[처리 흐름 요약]

- 장애 유형(DB / EKS / Region)에 따라 대응 로직 분기
- DB 엔진(RDS / Aurora)에 따른 승격 처리

09 회고

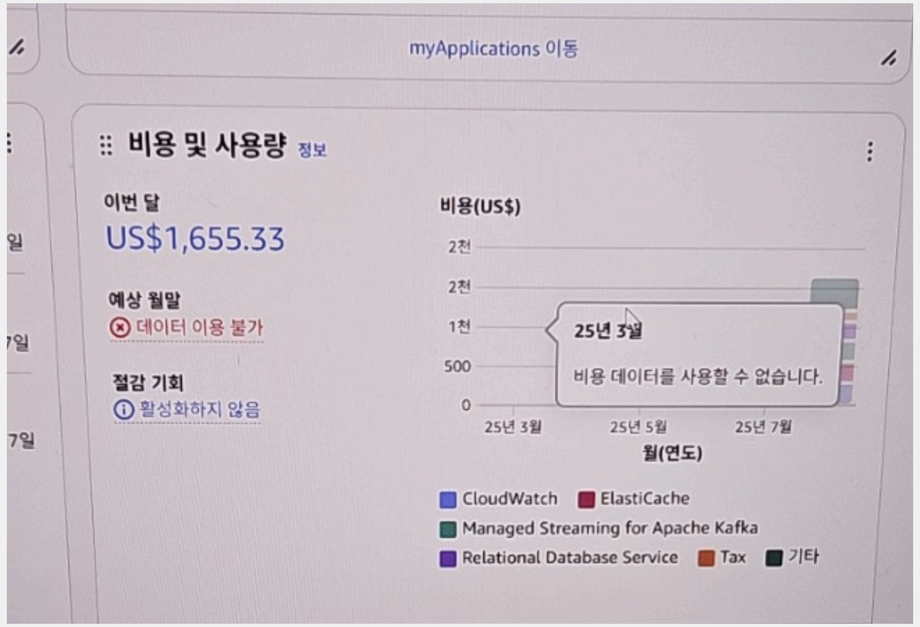
[결과 요약]

- Aurora / RDS 기준 RTO 5분·10분 이내 달성
- PoC 구성 후 DR 시나리오 테스트(10회+)를 통해 구조 검증 및 안정성 확보
- 초기 아키텍처 대비 약 33% 비용 절감 달성

[아쉬웠던 점]

- 아키텍처 설계 시 개념적 타당성은 확보했으나, 정량적 지표 기반 검증 부족
- 리소스 관리 미흡으로 비용 관리에 실패
- DB 장애만 발생해도 트래픽 전환 등 불필요한 오버헤드 발생

실제 프로젝트에 사용된 비용



[다시 한다면?]

- 정량 지표 기반 성능 검증 적용
- 리소스 사용량 기반 비용 최적화 설계
- 서비스·DB 단위 선택적 Failover 구조 적용
- 보안·백업 전략을 확장