

# CGV 예매 서비스 고도화

CLOUD SERVICE ENGINEER  
우정인

# 자기소개



우정인

소프트웨어학 전공  
네트워크 엔지니어 인턴  
CloudWave 3기  
최우수상 및 최우수수료

Terraform 기반의 DR 설계

# 목차

01 기획 과정

02 주요 아키텍처

03 CICD

04 모니터링/보안

05 비용 분석

06 마무리

# 목차

## 01 기획 과정

02 주요 아키텍처

03 CICD

04 모니터링/보안

05 비용 분석

06 마무리

올리브 시네마는 전 세계적으로 영화관을 운영하는 글로벌 체인으로, 고객들에게 최고의 영화 경험을  
날이면 급격히 증가하는 트래픽으로 인해 웹사이트와 앱에서 응답 속도가 느려지거나 서버 과부하  
못해 개인화된 서비스 제공에 어려움을 겪고 있으며, 지역별로 최적화되지 않은 콘텐츠 배포로 인해  
이러한 문제를 해결하고자 올리브 시네마는 클라우드 기반의 혁신적인 예매 플랫폼을 구축하려고  
효율성을 극대화하는 것입니다.

최근, 인기 영화 개봉일이나 대규모 이벤트가  
있는 날이면 급격히 증가하는 트래픽

인기 영화가 개봉하는 날처럼 트래픽이 급격히 증가할 경우를 대비해서 웹 애플리케이션과 백엔드 시스템을 설계하고, 이를 자동으로 확장할 수 있는 구조를 도입합니다. 이를 통해 예매, 결제, 좌석  
관리와 같은 각 기능을 유연하게 설계하고, 트래픽이 늘어날 때 필요한 리소스를 추가할 수 있습니다. 이러한 설계는 단순히 서버 과부하를 방지하는 데 그치지 않고, 고객이 어떤 상황에서도 빠르고  
안정적인 예매 서비스를 이용할 수 있도록 보장합니다.

비동기 방식

자동으로 확장할 수 있는 구조

작석 이 글로벌 영화 예매 플랫폼 혁신을 위한  
있습니다. 이러한 비동기 방식은 시스템 전체의 응답 속도를 높여 고객들이 원하는 영화를 빠르게 예약할 수 있도록 도와줍니다. 또한, 글로벌 사용자를 위해 정적 콘텐츠(영화 포스터, 예고편 등)를 여러 지역에 분산된 서버에서 제공하는 CDN을 도입하여 콘텐츠 전달 시간을 크게 줄일 수 있습니다.

글로벌

전 세계적으로 분산된 네트워크

운영팀은 실시간 예매 데이터를 활용해 서비스 상태를 모니터링하고, 리소스 할당을 최적화할 수 있습니다. 이를 분석 가능한 형태로 저장하여 인기 영화,  
시간대별 예매 패턴, 지역별 트래픽 데이터를 시각화합니다. 이를 통해 운영팀은 실시간으로 데이터를 확인하고, 각 영화에 적합한 프로모션을 적용하거나 특정 시간대에 대한 추가 리소스 배포를  
결정할 수 있습니다.

클라우드 아키텍처 설계

강화된 모니터링 시스템

이상 상황이 발생하면

자동 알림이 운영팀에 전송

강화된 모니터링 시스템은 운영팀에 이상 상황을 실시간으로 알립니다. 리소스를 실시간으로 추적하고, 애플리케이션 성능, 서버 상태, 네트워크 지연 시간 등을 모니터링합니다. 이상 상황이 발생하면 자동  
알림이 운영팀에 전송되어 즉각적인 대응이 가능하도록 설계됩니다.

예를 들어, 블록버스터 영화가 개봉하는 날에는 미리 설정된 확장 규칙이 발동되어 필요한 리소스가 자동으로 배포됩니다. 이를 통해 빠르게  
제공되고, 고객에게 푸시 알림으로 맞춤형 프로모션을 보냅니다. 고객이 영화와 좌석을 선택하면 예약 정보가 실시간으로 업데이트되고, 결제 완료 후에는 확인 메시지가 전송됩니다.

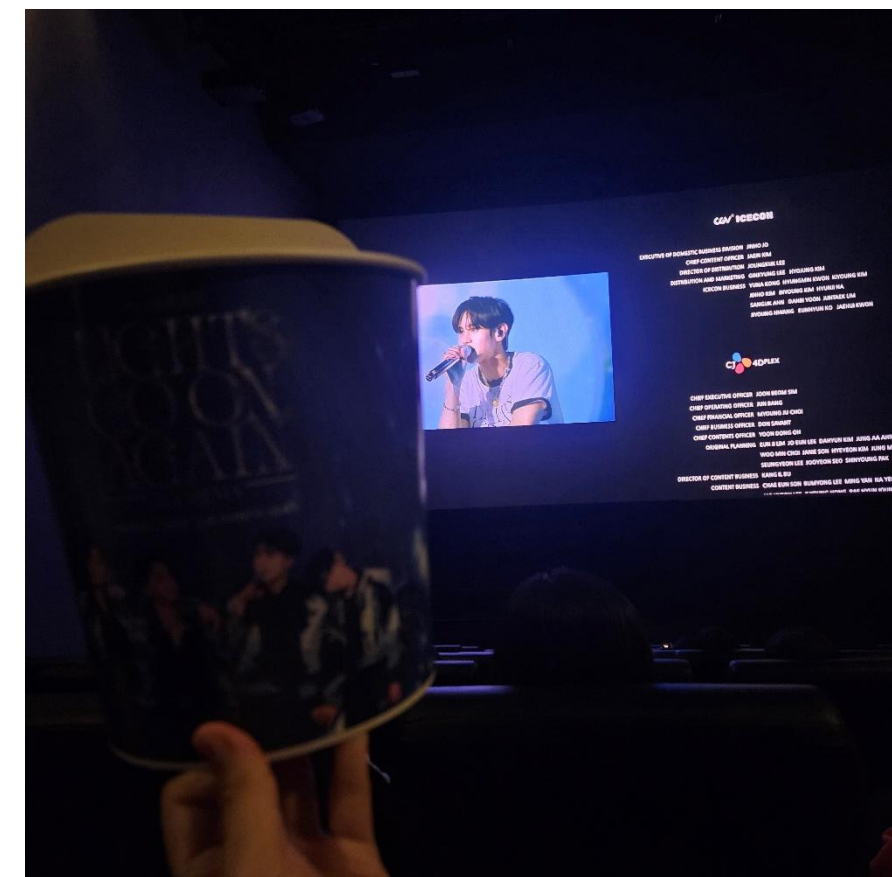
이벤트가 종료된 후에는 추가로 배포된 리소스가 자동으로 축소되어 비용을 절감합니다. 저장된 데이터를 분석해 다음 이벤트를 위한 인사이트를 도출함으로써 지속적으로 서비스를 개선할 수  
있습니다.

새로운 플랫폼은 SI, 컨테이너 기반 설계, 서버리스 아키텍처, 글로벌 콘텐츠 배포, 실시간 데이터 분석 등 고도화된 기술을 통해 제공되어 올리브 시네마가 글로벌 영화 예매 시장에서 디지털 리더로  
자리 잡을 수 있도록 지원합니다. 고객은 더 개인화된 추천과 빠른 서비스를 통해 최상의 예매 경험을 누릴 수 있으며, 올리브 시네마는 강화된 운영 효율성으로 비용을 절감하면서도 글로벌 확장을  
이어갈 수 있습니다.

# CGV의 ICECON 사업

# 얼터콘텐츠 상영을 통한 팬덤형 소비자 증가

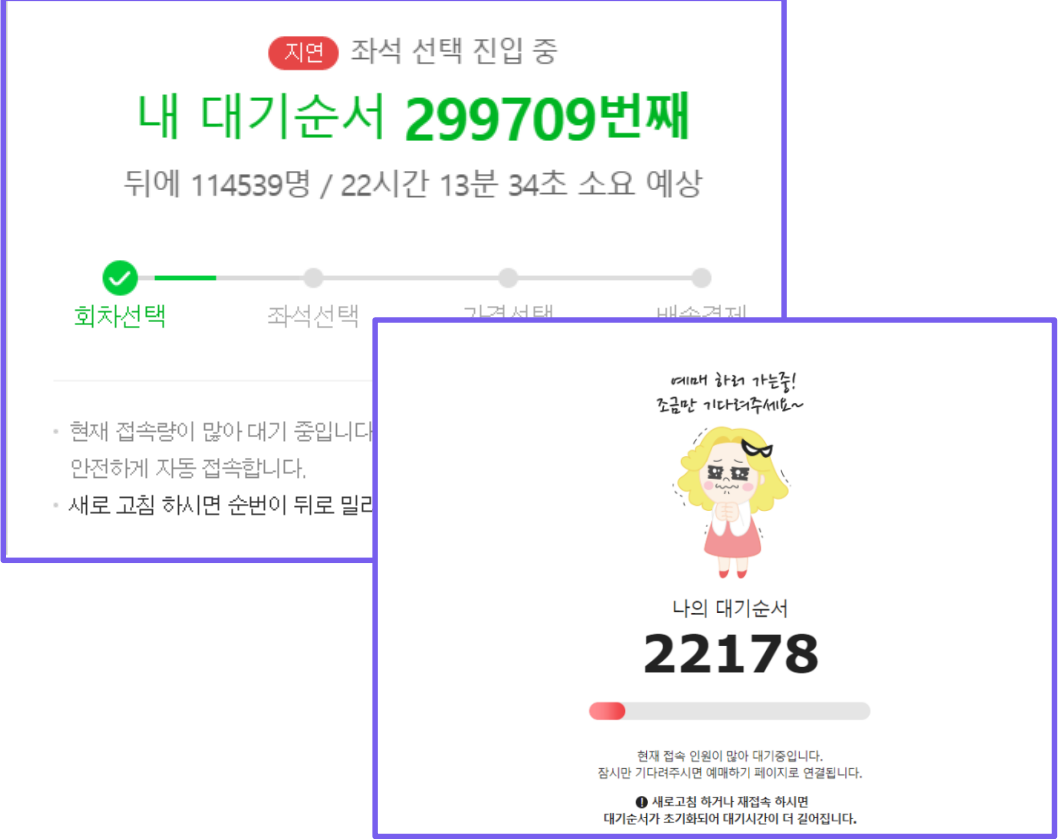
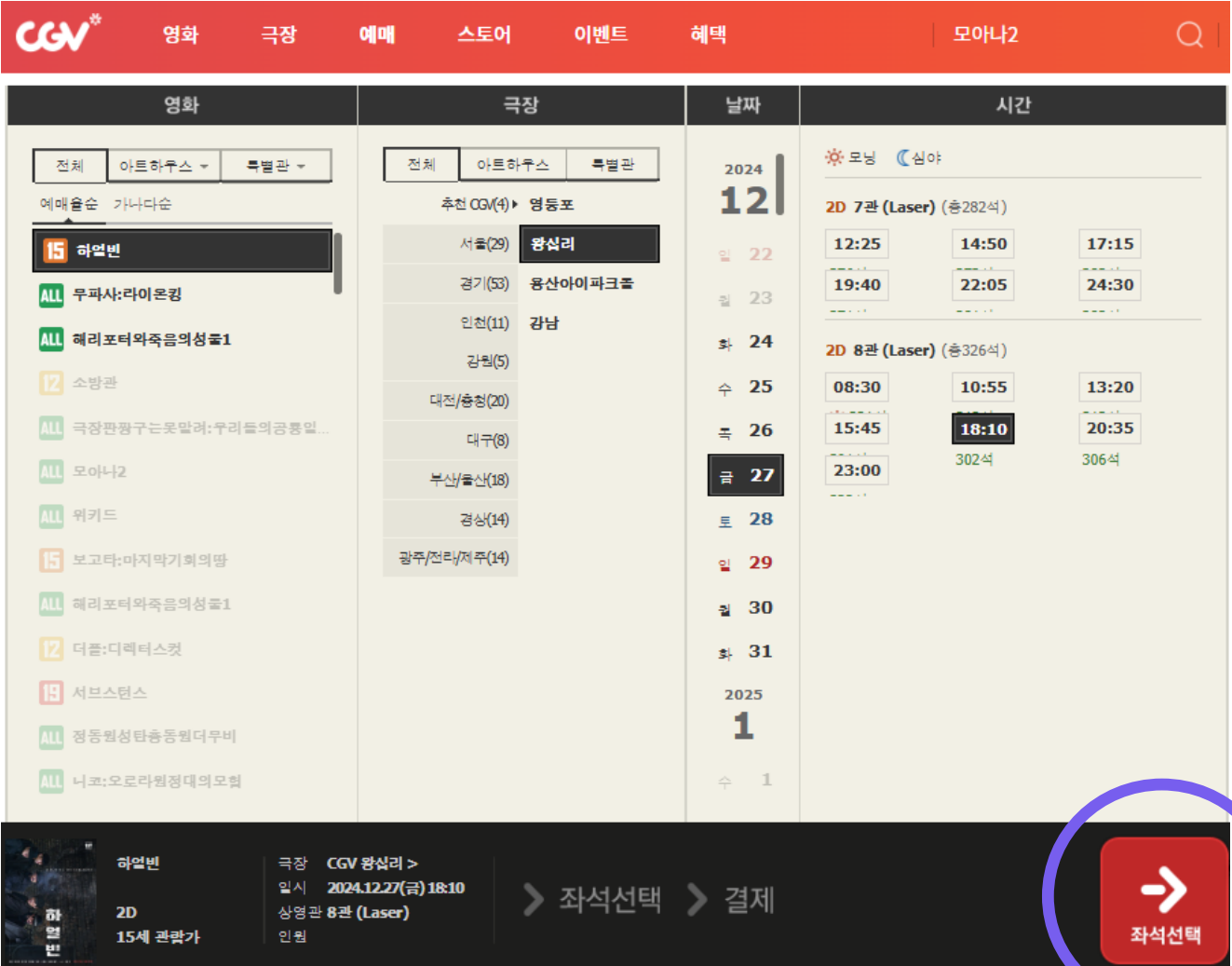
# 예매 오픈 시 **트래픽 집중**



24.10 개봉 하이라이트 콘서트 영화

# 시나리오 기획 과정

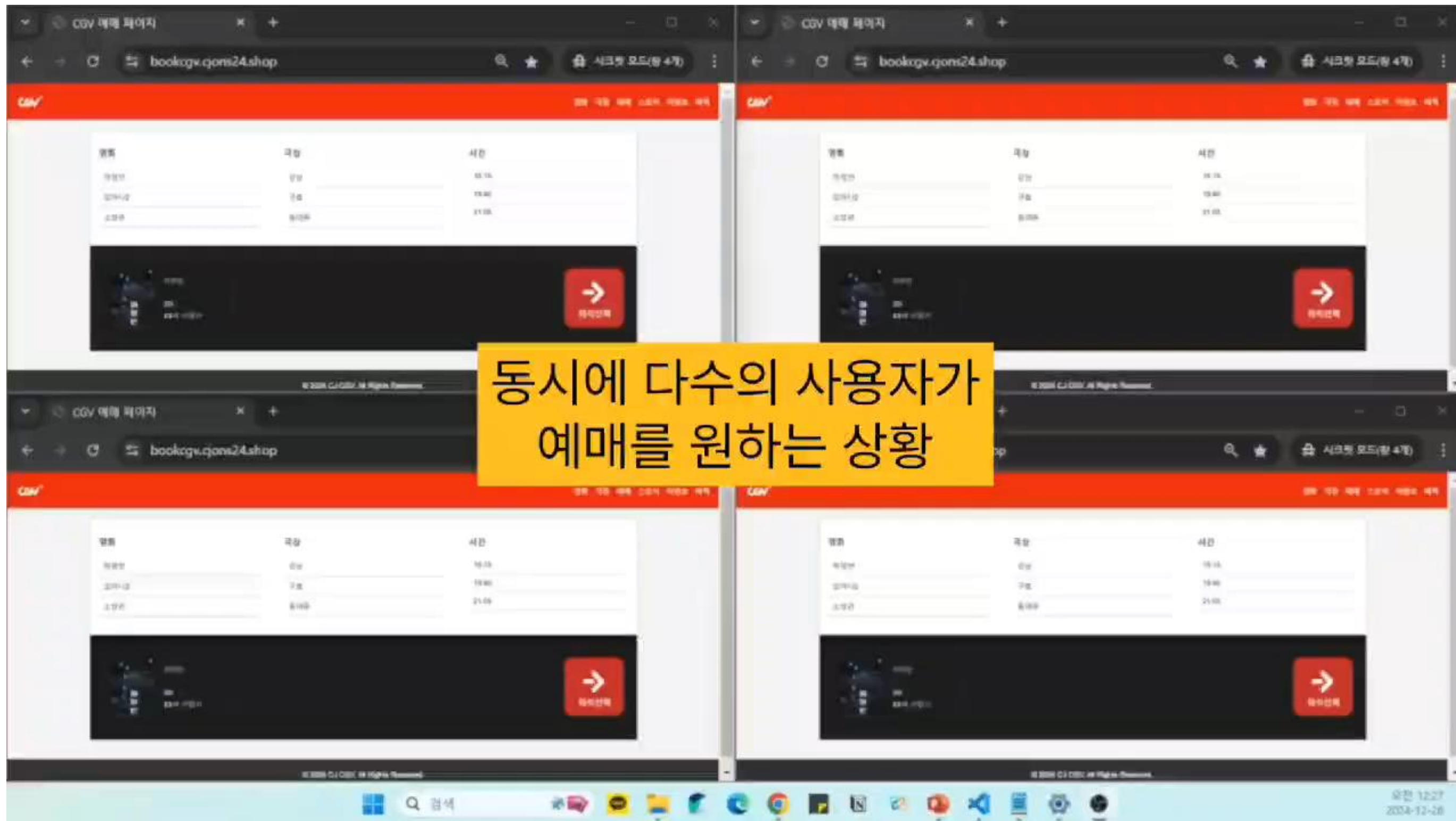
## 01 기획과정



대기열 제공

# 시연

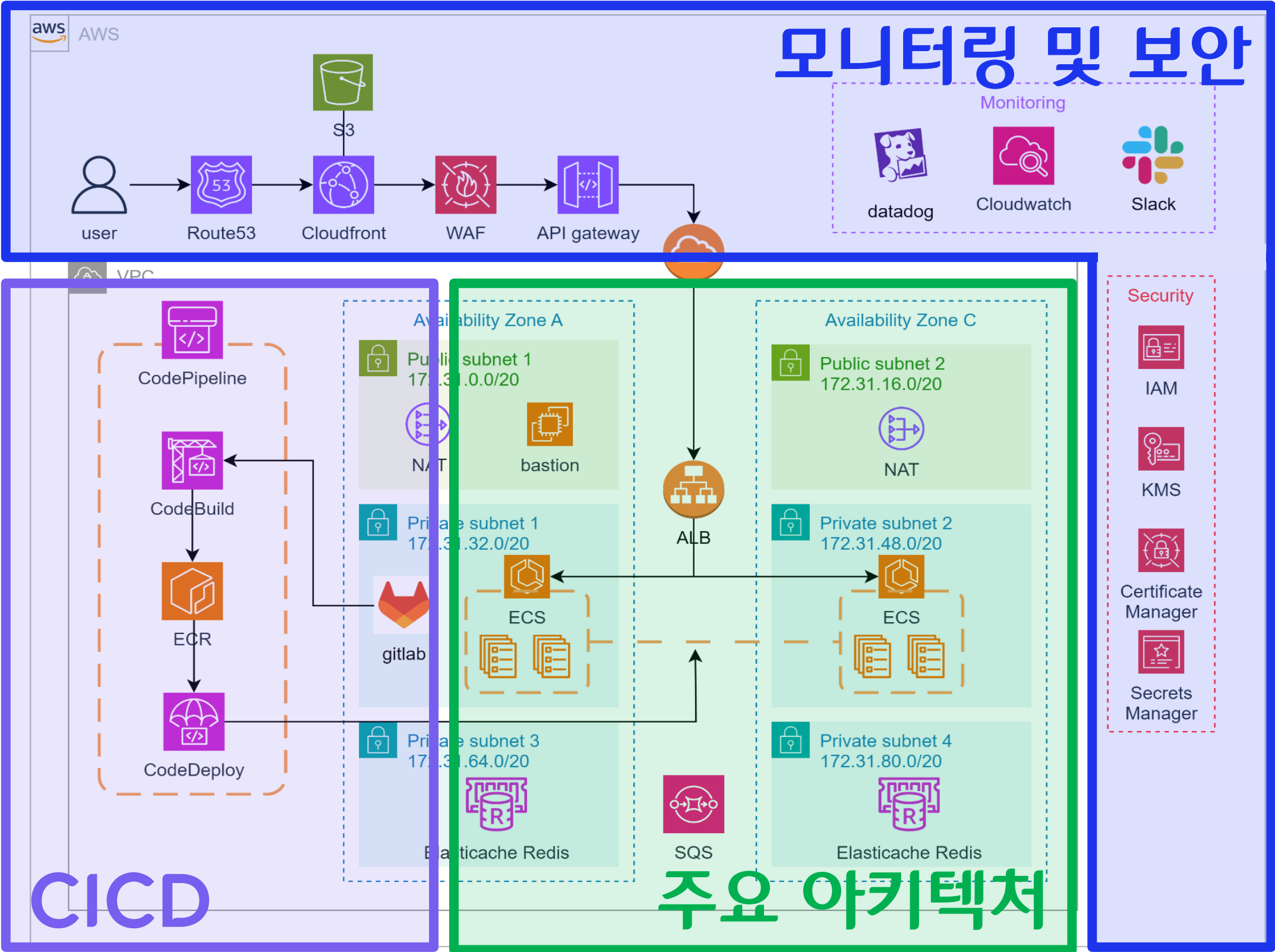
## 01 기획과정





# 전체 아키텍처 구성도

01 기획과정



# 목차

01 기획 과정

**02 주요 아키텍처**

03 CICD

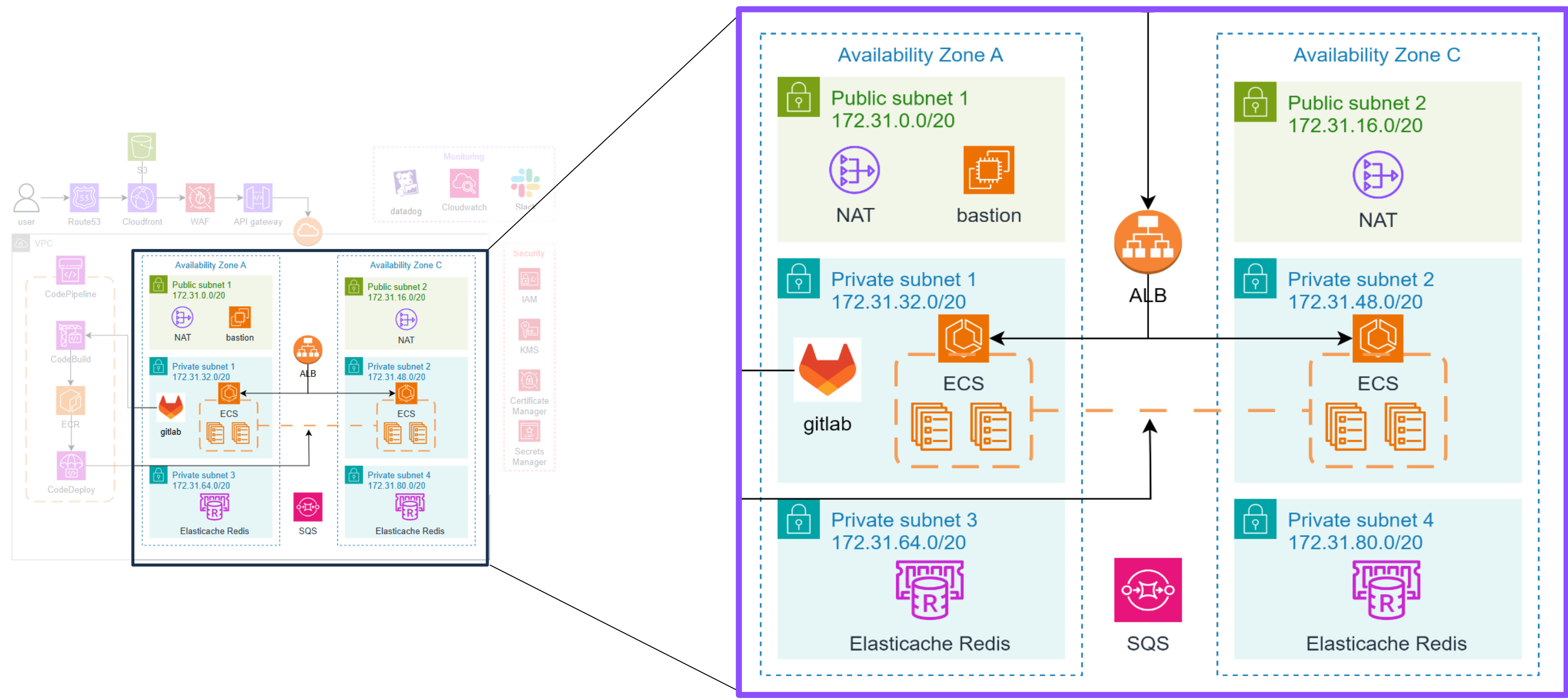
04 모니터링/보안

05 비용 분석

06 마무리

# 주요 아키텍처 구성도

## 02 주요 아키텍처



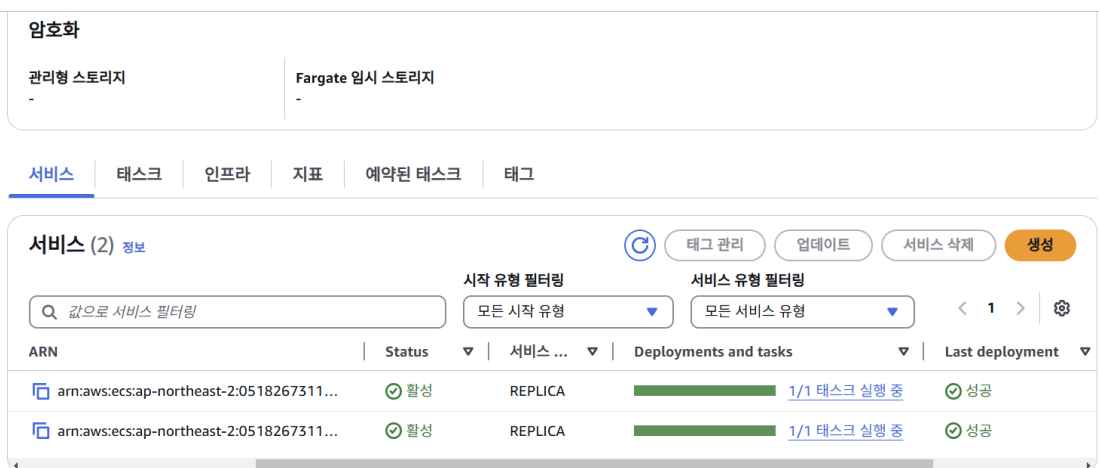
# ECS VS EKS

02 주요 아키텍처

ECS



- 초기 구성이 용이
- 직관적인 UI 제공
- 클러스터 운영 안정성 보장



VS

EKS



- Kubernetes 오픈소스 생태계
- 높은 러닝커브
- Control Plane 비용 발생

월별 비용 ▼	설명 ▼	리전
73.00 USD	-	아시아 태평양(서울)
구성 요약		
EKS 클러스터 수 (1), 하이브리드 노드 수 (1 월간)		

## Redis



- 빠른 처리 속도를 기반으로 대기번호 부여
- 대기 순번 및 대기열 이탈자 관리

## SQS

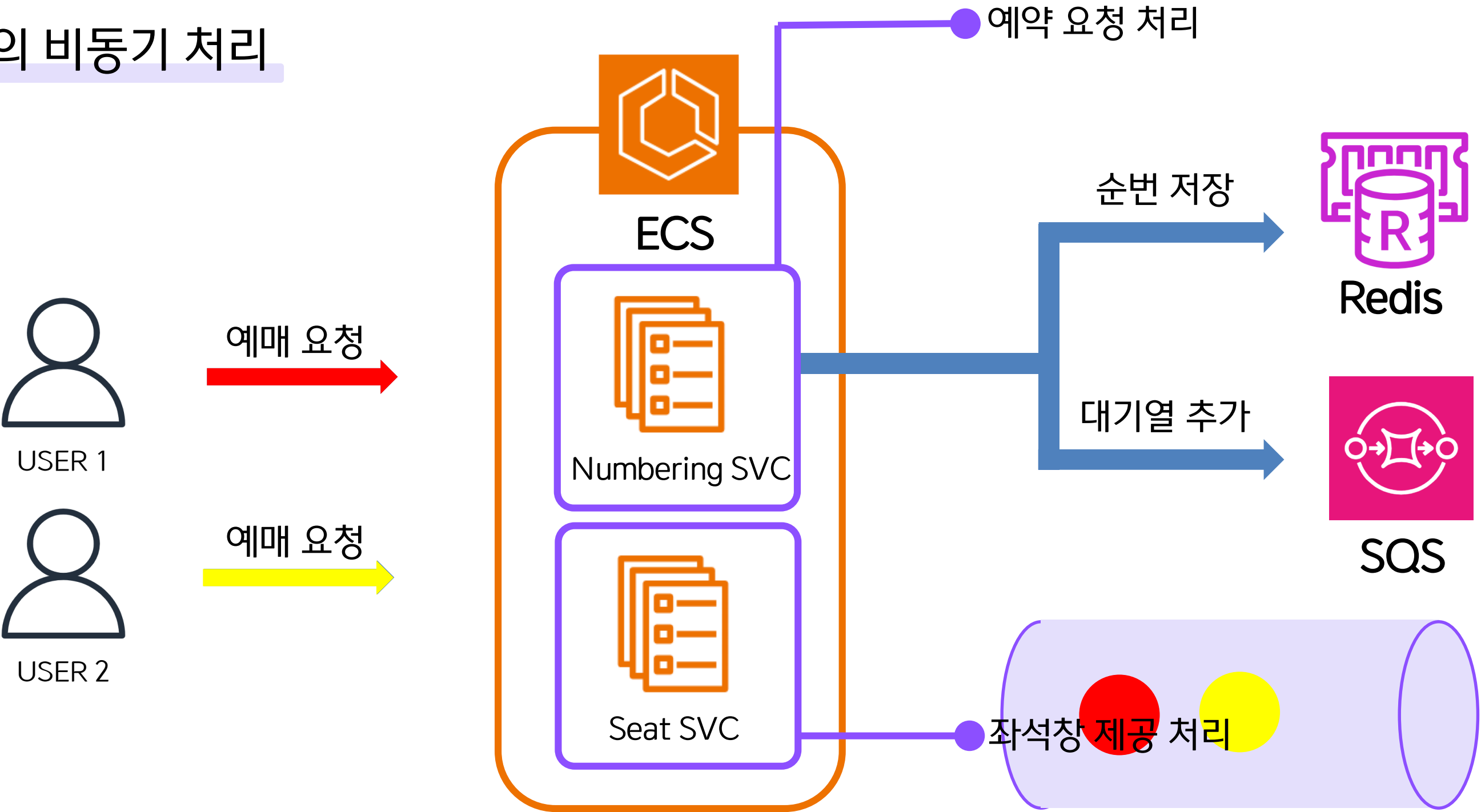


- 대기열 수와 메시지 크기 제한 없이 확장 가능
- 자동 실패 관리

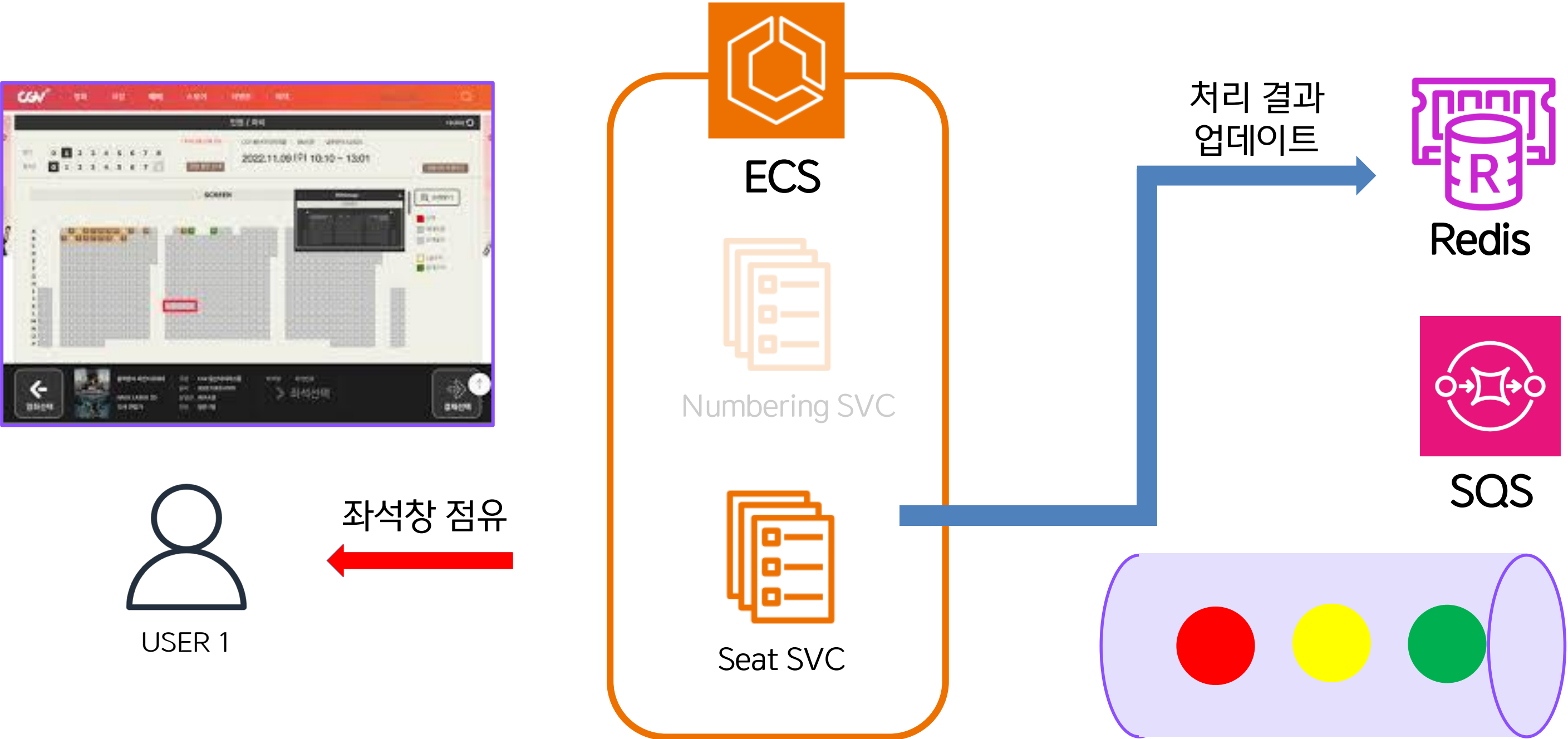
# 대기열 구현 로직

02 주요 아키텍처

## MSA 기반의 비동기 처리



# 대기열 구현 로직



# 목차

01 기획 과정

02 주요 아키텍처

**03 CICD**

04 모니터링/보안

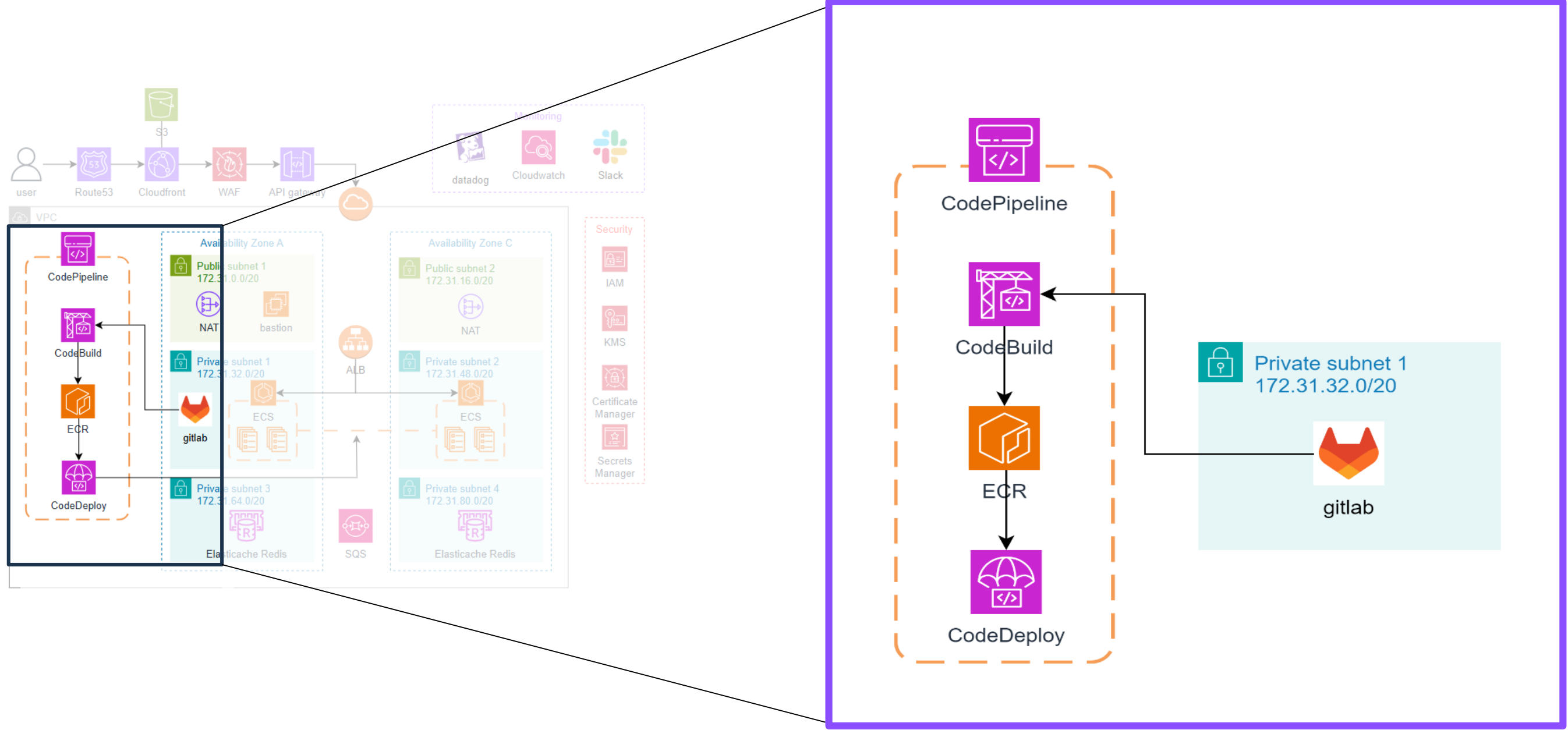
05 비용 분석

06 마무리



# CICD 구성도

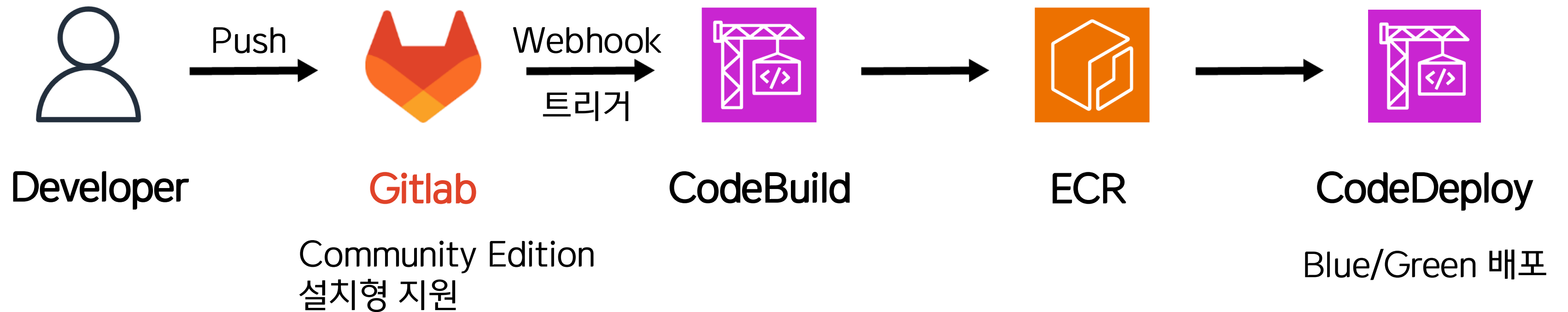
## 03 CICD



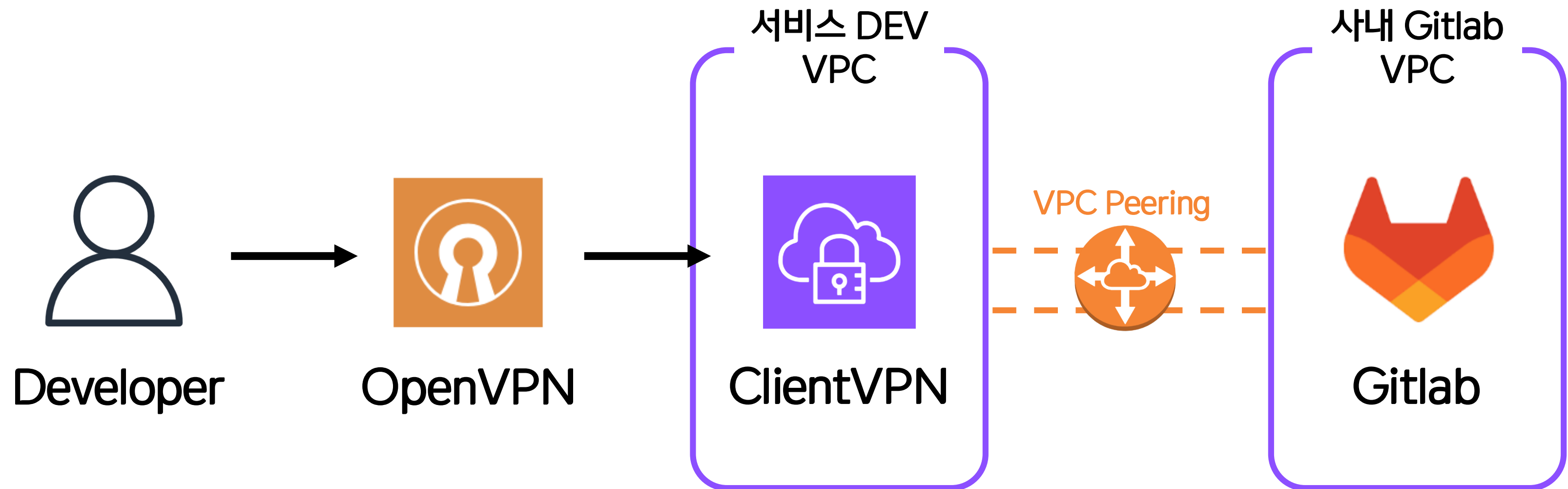


## CodePipeline

AWS와의 유연한 연계를 위한 선택



## ClientVPN과 VPC Peering을 통한 사내 Gitlab 연결



# 목차

01 기획 과정

02 주요 아키텍처

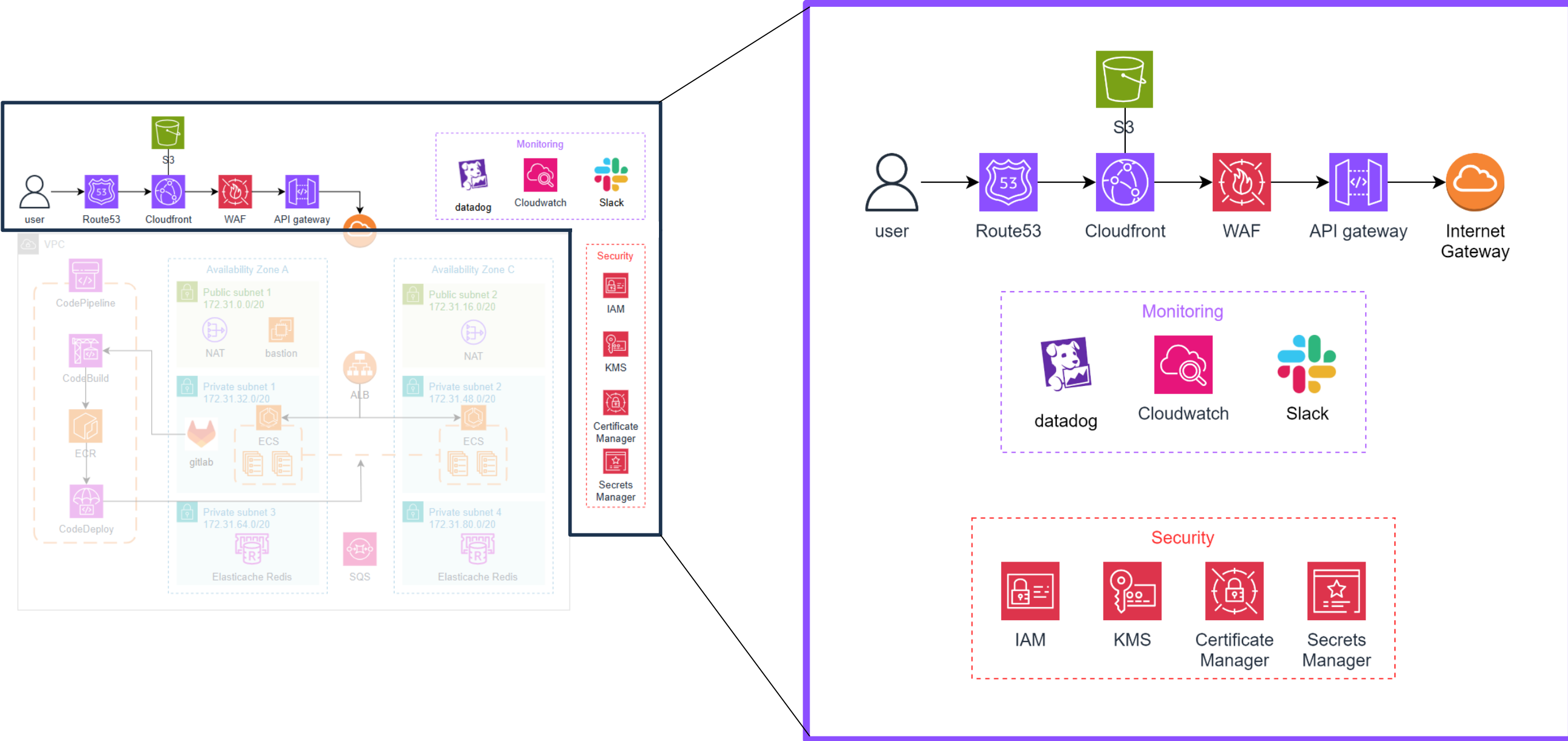
03 CICD

**04 모니터링/보안**

05 비용 분석

06 마무리

# 모니터링 및 보안 아키텍처



## WHY DATADOG



Datadog

X



Cloudwatch

VS



Prometheus

X



Grafana

- 메트릭부터 APM까지 **올인원 서비스**
- Integration을 통한 **빠른 초기 구성**
- Cloudwatch와의 연동을 통해  
이슈상황의 로그를 간편하게 확인

- 초기 구축 비용이 낮음
- 운영 시의 공수와 러닝커브가 높음

### DATADOG 대시보드 구성

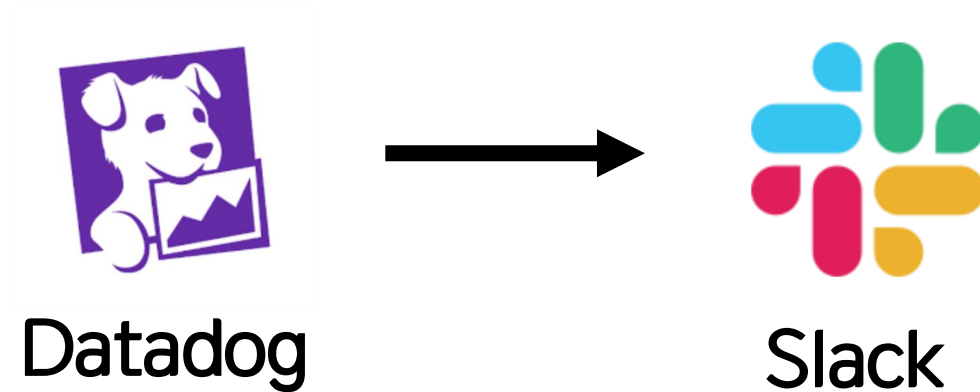
컨테이너별  
메모리 사용률



컨테이너별  
CPU 사용률

ALB HTTP 상태  
코드별 응답 수

### DATADOG 슬랙 알람 연동



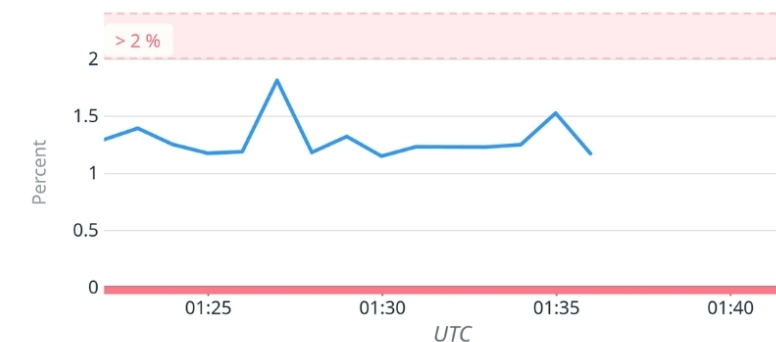
aws.ecs.service.cpuutilization

80% 도달 시 슬랙 알람 트리거

Triggered: [TEST] [INFRA] CPU 사용률 > 2.0  
on  
clustername:wji-dev-ecs,servicename:wji-b  
ookcgv-numbering  
@slack-cgv-ecs-alert

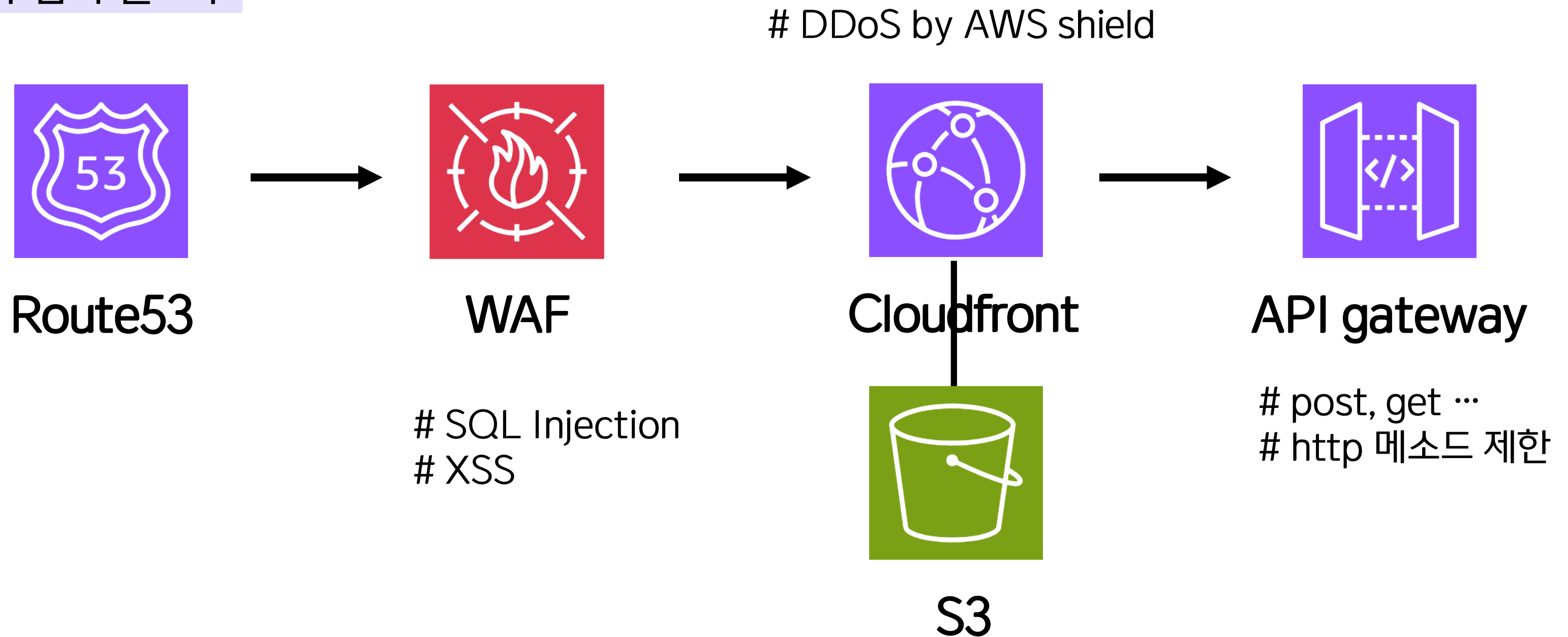
\  
[알람발생]\  
대상서버: CPU 사용량 > 2.0\  
발생시간(KST): 2024-12-24 10:42:02+09:00\

Tags	Notified
clustername:wji-dev-	@slack-cgv-ecs-alert
ecs,	
servicename:wji-boo	
kcgv-numbering	





## 사용자 접속 플로우



## AWS 서비스를 활용한 보안 구성



KMS

ECS – Redis 간  
데이터 전송 암호화



ACM

클라이언트 – ALB 간  
HTTPS 암호화 통신



IAM

사용자 그룹별 최소 권한  
정책으로 서비스 접근 제어



Secrets  
Manager

AWS, DB 계정 정보 등의  
민감 정보 관리 및 자동 갱신

# 목차

01 기획 과정

02 주요 아키텍처

03 CICD

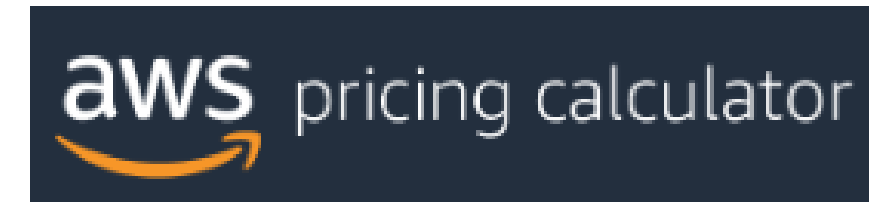
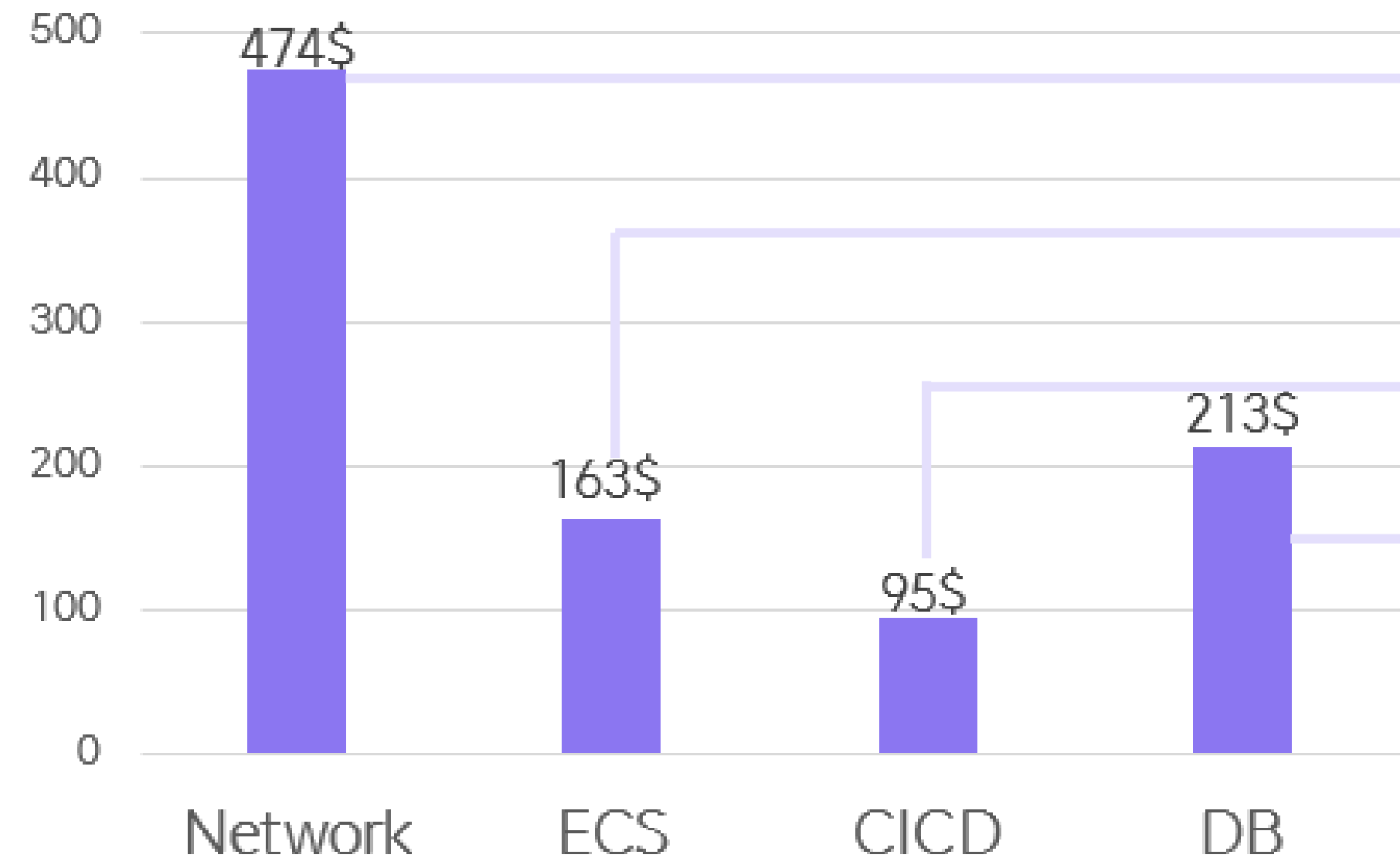
04 모니터링/보안

**05 비용 분석**

06 마무리

## AWS pricing calculator 활용 비용 산정

월별 / \$



VPC(NAT 2개), API Gateway, Route53

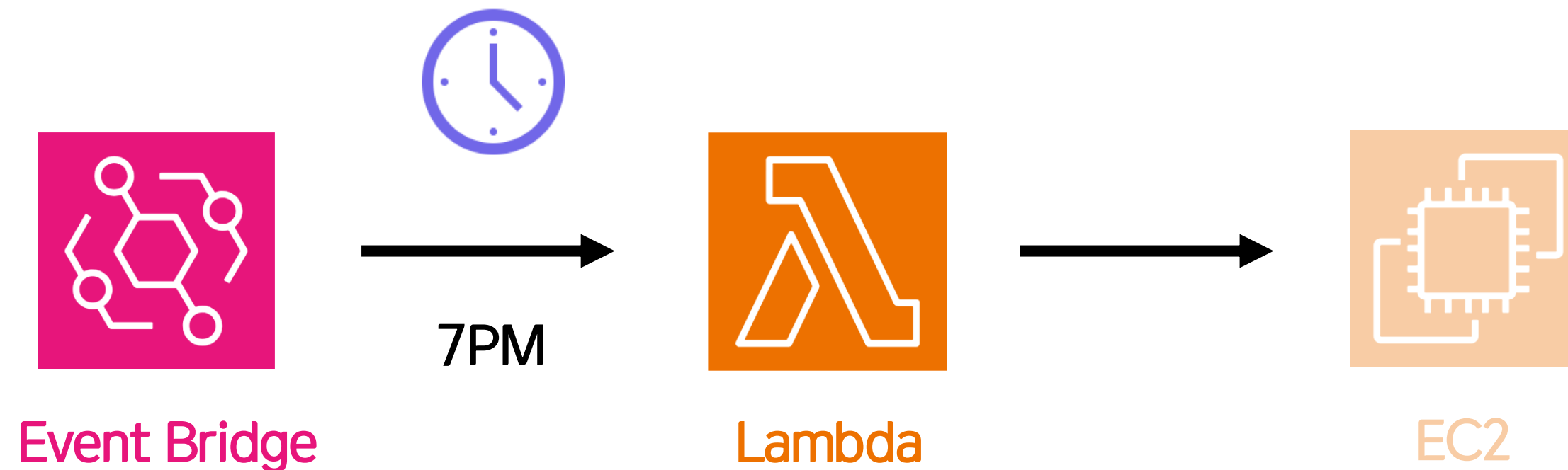
2개 태스크 2vCPU \* 4GB 운영 기준

Gitlab t3.large 50GB, Pipeline

Redis, SQS

총 945\$

## EventBridge와 Lambda를 활용한 EC2 자동 On-Off



매달 약 60\$ 절감 효과

# 목차

01 기획 과정

02 DEV/QA

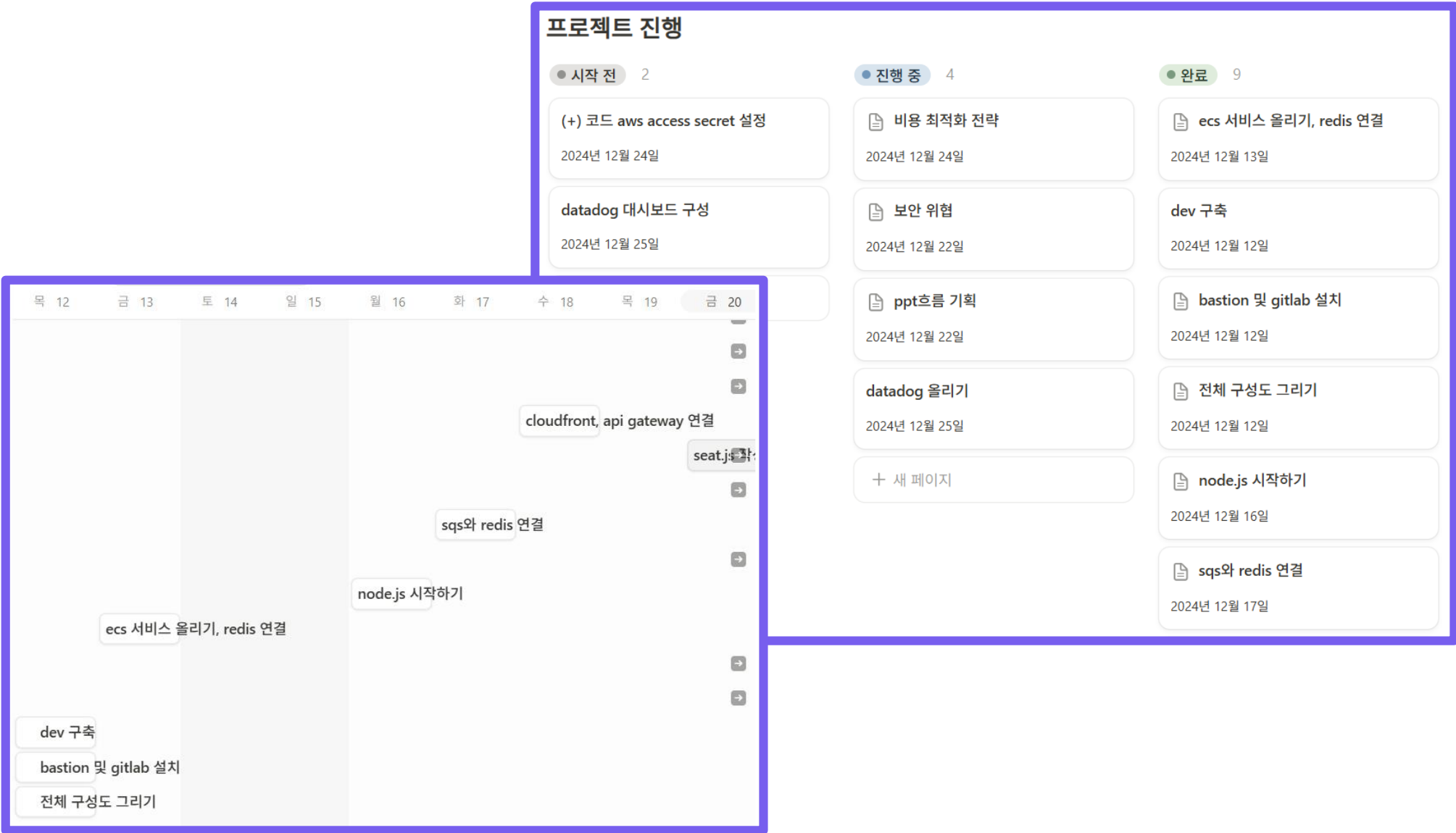
03 PROD

04 DR

05 비용 분석

**06 마무리**

## WBS, 칸반보드를 활용한 일정 관리



## Datadog 연동

- 디버그 레벨의 로그 확인
- 환경변수 수정
- ECS 서비스 재배포
- API 키 유효성 검사

Datadog named a Leader in the 2024 Gartner® Magic Quadrant™ for Digital Experience Monitoring

PRODUCT CUSTOMERS PRICING SOLUTIONS

BLOG LOGIN GET STARTED FREE

### Get Started with Datadog

No credit card required

Try it free for 14 days and monitor as many servers as you like.

\*Required Fields

Region\*

Please choose carefully. You can't migrate data between regions.

Where do you want your data housed?

United States (US-Central)

Business Email\*

Full Name\*

Company\*

Password\*

Use at least 8 characters containing at least 1 number and 1 lowercase letter.

Phone

\*Required fields. By signing up, you agree to the [Master Subscription Agreement](#), [Privacy Policy](#) and [Cookie Policy](#).



## Datadog 연동

- 사이드카 방식의 Datadog agent 설치 시
- 태스크 정의 -> 추가 컨테이너

### ▼ 환경 변수 - 선택 사항

환경 변수 | 정보

개별적으로 추가

키 값 페어를 추가하여 환경 변수를 지정합니다.

키

DD\_SITE

값 유형

값



값

us5.datadoghq.com

제거

US\$ 계정에 해당하는 값본값

## Datadog 연동

제품 고객 가격 솔루션

소개 블로그 로그인 GET STARTED FREE

Datadog 문서

Search documentation...

통합 가이드

개방형텔레메트리

개발자

Administrator's Guide

API

Datadog Mobile App

CoScreen

Cloudcraft

앱 내

Dashboards

2. 왼쪽 메뉴에서 **작업 정의**를 클릭한 뒤 **작업 정의** 탭을 클릭하여 기존 Fargate 작업 정의를 선택합니다.

3. 새 작업 정의의 경우 다음을 수행합니다.

- a. 시작 유형으로 **Fargate**를 선택한 후 **다음 단계** 버튼을 클릭합니다.
- b. **my-app-and-datadog** 와 같은 **작업 정의 이름**을 입력합니다.
- c. 작업 실행 IAM 역할을 선택합니다. 아래의 **IAM 생성 또는 수정 정책** 섹션에서 권한 요구 사항을 참조하세요.
- d. 필요에 따라 **작업 메모리** 및 **작업 CPU**를 선택하세요.

4. 추가 **컨테이너** 버튼을 클릭하여 Datadog 에이전트 컨테이너를 추가하기 시작합니다.

- a. **컨테이너 이름**에 대해 **datadog-agent** 를 입력합니다.
- b. **이미지**의 경우 **public.ecr.aws/datadog/agent:latest** 를 입력합니다.
- c. **환경 변수**에 키 **DD\_API\_KEY** 를 추가하고 Datadog API 키를 값으로 입력합니다.
- d. 키 **ECS\_FARGATE** 및 **true** 값을 사용하여 환경 변수를 추가합니다. **추가**를 클릭하여 컨테이너를 추가합니다.
- e. 키 **DD\_SITE** 및 **us5.datadoghq.com** 값을 사용하여 환경 변수를 추가합니다. 설정하지 않으면 기본값인 **datadoghq.com** 입니다.
- f. (윈도우즈(Windows) 전용) **C:\** 를 작업 디렉터리로 선택합니다.

5. 다른 애플리케이션 컨테이너를 작업 정의에 추가합니다. 통합 메트릭 수집에 대한 자세한 정보는 **통합 ECS Fargate** 설정을 참조하세요.

6. **생성**을 클릭하여 작업 정의를 생성합니다.

LANGUAGE

한국어

DATADOG SITE ?

US5

US1

US3

US5

EU

AP1

US1-FED

메트릭 수집

크롤러-기반 메트릭

로그 수집

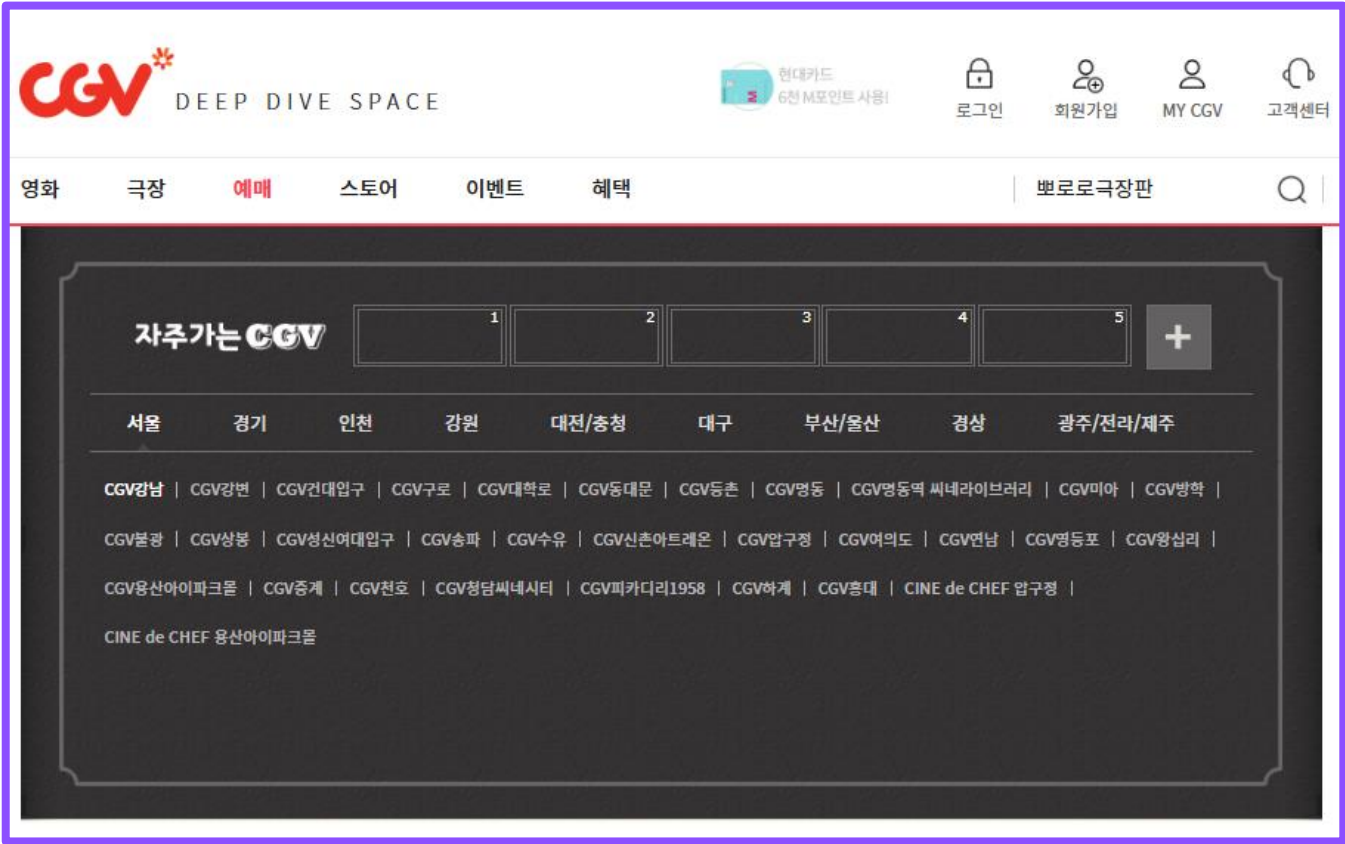
트레이스 수집

프로세스 수집

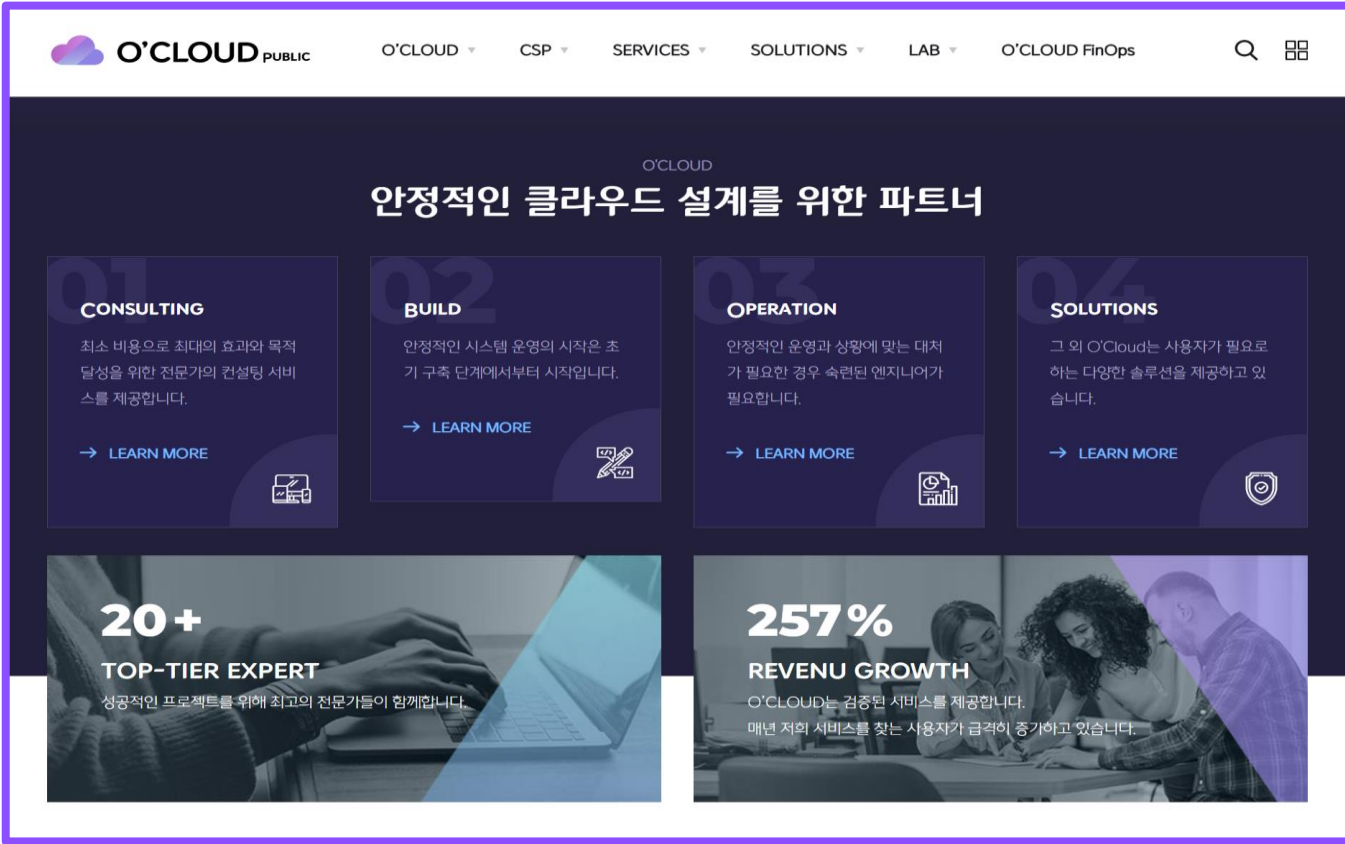
즉시 사용 가능한 태그

Feedback

## CGV 인프라 개선 경험



## O'CLOUD 사업



# Q&A