



MSA 서비스 개발 및 운영환경 구축





INDEX

목차

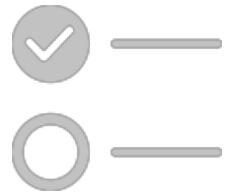
- 1.** 개요 : 개발 동기, 벤치 마킹, 소개
- 2.** 관리 : 협업 방식, 일정
- 3.** 결과 : 기술 스택, 아키텍처
- 4.** 시연
- 5.** Q&A



INDEX

프로젝

트
개요



shortpingoo.shop

1. 개발 동기

- 필요성 : 리서치, 설문 조사

2. 벤치 마킹

- TikTok, Instagram, Musinsa

3. 서비스 소개

- 서비스 특징, 차별점



motivation

개발 동기

- 팀원의 생활 속 불편한 점에서 •
시작
온라인 의류 쇼핑을 할 경우
1. 리뷰 사진만으로는 부족한 색감, 재질 정보
 2. 유튜브에서 직접 검색하여 해당 상품의 정보를 찾아봐야 하는
번거로움



필요성 리서치 결과

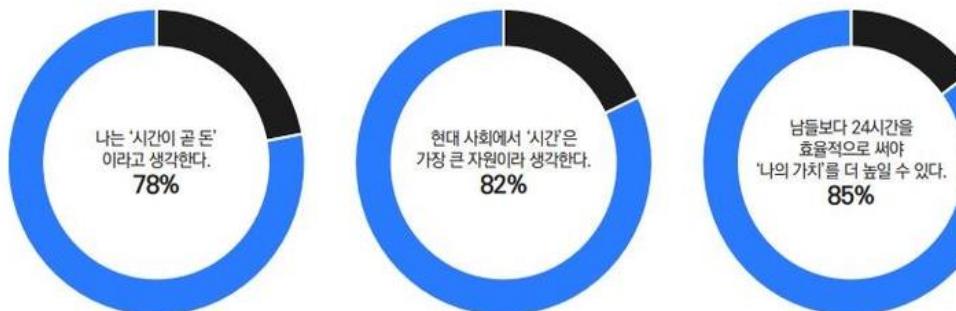
시간이 곧 돈이 된 현대 사회

효율적인 시간 활용에 대한 중요성 증대

불가리, 놀거리, 먹을거리가 다양해지며 한정된 시간을 얼마나 가치 있게 사용하는가에 대한 소비자들의 관심이 어느 때보다 높습니다.

관련 설문조사에 따르면 10명 중 8명이 시간은 곧 돈이자 중요한 자원이며, 나의 가치를 높이기 위해 24시간을 효율적으로 사용해야 한다고 응답했습니다.

이에 따라 실패의 경험은 줄이고 시간 절약엔 도움이 되는 제품·서비스, 장면보단 짧은 영상 위주의 속품 등을 소비하는 트렌드가 나타나고 있습니다.



엠브레이인 트렌드모니터, 시간 절약 서비스 & 자원 부족 체감도 및 효율적 활용의 필요성 관련 조사, 2023

AI 쇼핑부터 콘텐츠 배속·요약까지...'초단축 소비' 뜬다

(서울=연합뉴스) 이정현 기자 = 최소한의 시간으로 최대한의 쇼핑 정보를 획득하기, 최소한의 시간으로 최대한의 게임 성취감 획득하기, 최소한의 ...

연합뉴스 / 7월 29일

출처 : 연합뉴스

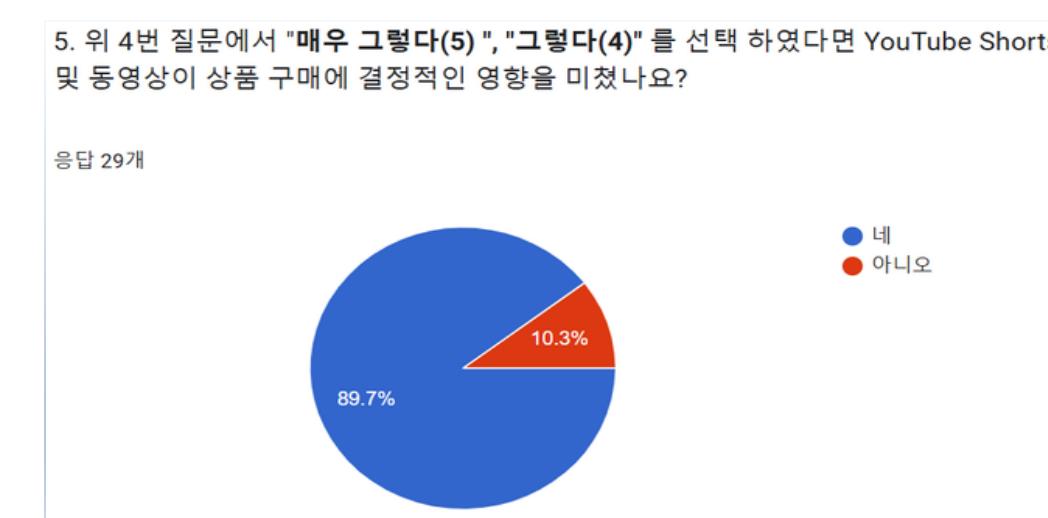
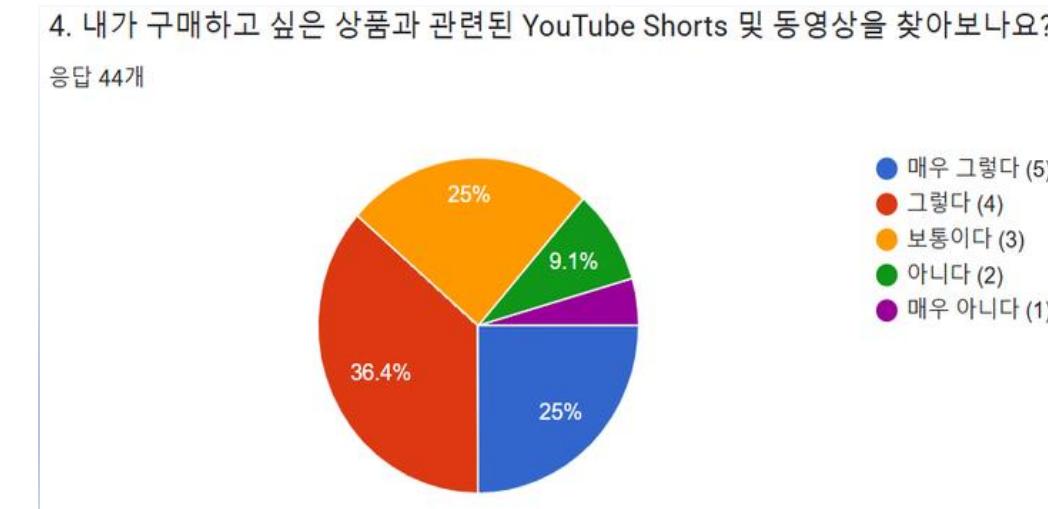
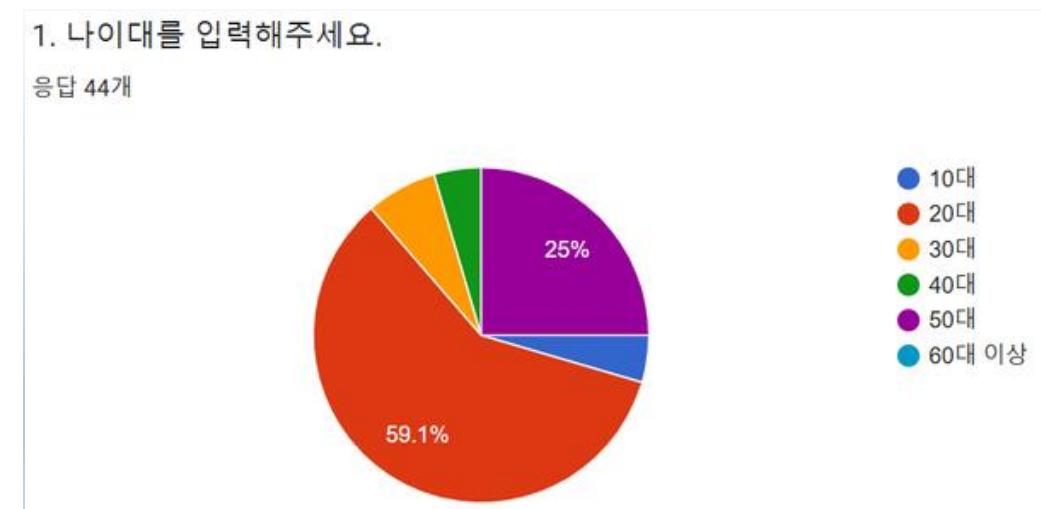
최소한의 시간으로 최대 효율을 추구하는
'숏폼'의 영향이
AI 쇼핑/숏츠 콘텐츠로 확산 중



필요성 설문 조사

● 결과 ●

많은 사용자가
의류 구매 시 영상에 의존





benchmarking

벤치 마킹



- **TikTok Shop**
- 솟폼 광고를 활용한 플랫폼
- **Instagram Reels + Shop**
- 릴스와 쇼핑 기능을 연동하여 솟폼 활용 플랫폼 구축
- **Musinsa**
- AI 서비스를 통한 개인화 추천 서비스



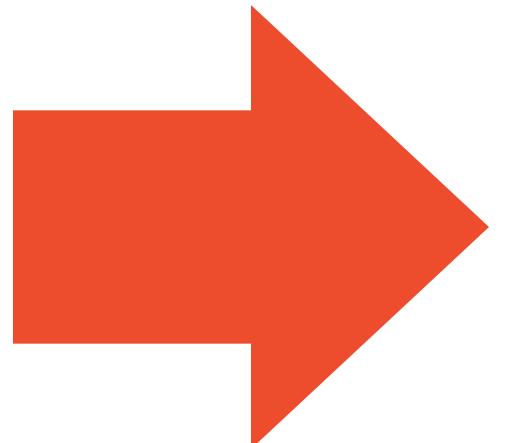
벤치 마킹 결과

차별점

광고 - 쇼핑 연계

상품 필터 및 추천

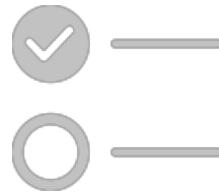
추천 시스템 및 개인화된
광고
리뷰 및 커뮤니티
기능



- 쇼츠 광고를 더한 이커머스
- AI 기반 서비스 개인화
- 커스텀 장바구니를 통한 고객
유치
구매자 편의와 판매 촉진을
- 동시에 지원하는 AI 도구

INDEX

프로젝트 관리



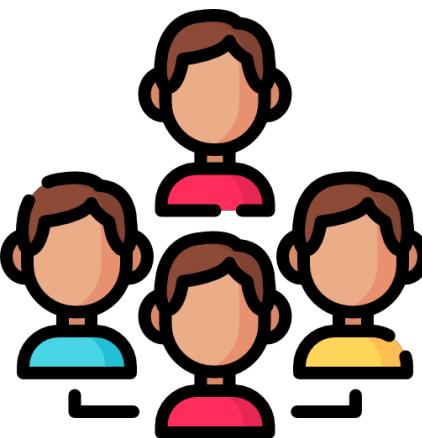
1. 협업 방식

- 스프린트, 스크럼 회의, 회고



2. 일정

- 마일 스톤



3. 팀원 소개

- 역할



Collaboration

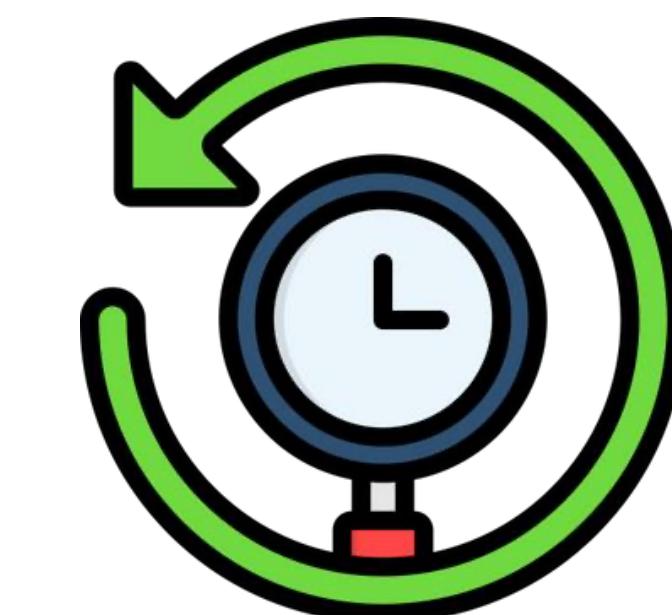
협업 방식

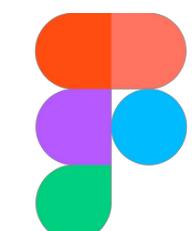


 Jira : 스프린트



 Notion : 스크럼 회의



 Figma : 회고



Sprint Board

스프린트 보드



프로젝트 / rookies_final3
4주차 개발 2주차

스탠드업 시작 스팰트 완료 Team Insights ...

그룹화 기준 에피 인사이트 설정 보기

RF-26 / RF-33

+ 추가 앱

원료 완료 작업 설명 편집

세부 사항

담당자 우예리 나에게 할당

레이블 없음

상위 항목 RF-26 설계

Team 없음

수정 버전 없음

기한 없음

Sprint 4주차 개발 2주차 +4

개발 브랜치 만들기 커밋 만들기

릴리스 보고자 우예리

Automation Rule executions

Q 검색 H B M P A +2 E 스팰트 1 필터 지우기

할 일 3 / 3 진행 중 5 / 9 완료 21 / 43

RF-26 설계 (1개 이슈) 해야 할 일

RF-29 프론트 개발 (9개 이슈) 해야 할 일

브랜드 페이지(Brand)
RF-126

회원 탈퇴(Mypage)
RF-48

1:1 문의(Mypage)
RF-47

정보 수정(Mypage)
RF-46

로딩페이지
RF-209

상품 상세 페이지(Shop-detail)
RF-65

로그아웃(Mypage 기능)
RF-59

위시리스트(Mypage)
RF-60

+

이슈 만들기

RF-67 백 개발 (6개 이슈) 해야 할 일

API gateway Routing 처리
RF-215

Gateway Filter
RF-217

[user]/api/user/exit
RF-175

[user]/api/user/{userId}
RF-173

Eureka Server

[user]/api/user/update/{userId}
RF-174



Scrum meeting

스크럼 회의



스크럼 회의

9시 10분부터 시작하는 회의

이전 수행 사항, 오류 또는 어려운 부분, 오늘 진행 계획 등을 공유

스크럼 회의 종료 시 모든 팀원이 본인만큼 모 진행 사항, 진행 계획을 인지할 수 있게 공유 필수

▶ 개발 일정

▶ 10월

▼ 11월

▼ 11/01

▼ 이전 수행 사항

프론트 : 마이페이지 닉네임 null 값으로 넘어오는거랑 회원 탈퇴 Id로 회원 정보가 삭제 되어야 하는데 토큰으로 되어있어 수정해 회원 탈퇴 및 닉네임 표시 마무리, Mypage 주문 조회 및 정보 수정 폐이지

DevOps(환경설정) :

- terraform 로드밸런서 생성 시 cname Route 53으로 자동화시키는 코드를 terraform에 추가
- 도메인 provider 에서 domain 구매 후 acm ssl 인증서 발급 완료
- router 53 name server 들을 domain provider name server 에 mapping.
- helm chart 수정 후 aws에서 지원하는 loadbalancer를 k8s 내부 loadbalancer에 할당 완료 후 external-ip 확인

백지연 : user 수정 후 최종 마무리, brand 리뷰 등록, 조회 구현

우예리: api게이트웨이 인가 코드 타타타타 치기

AI/pipeline:

- + :: - kafka P, C 구현 순서 config 및 Listener annotation 구체화 → kafka props , Factory(props) → ContainerFactory(branch specific case Factory by Listener Annotations) → 분기된 이름의 containerFactory 차등 적용 + filter set 구현후 동일 이식이 가능하지 테스트 함
- DeferredResult, completeFuture callback 이용한 non webflux - http non blocking (Http Queueing) 구현 테스트함
 - 다른 로컬환경에서 쇼츠 긍/부정 API 테스트 완료

▼ 오류, 어려운 부분

프론트 : Mypage 정보 수정에서 성별 선택하는 부분 애니메이션 어려웠어요..(민수해결)

우예리: 게이트웨이 의존성 충돌 어려워요 ㅎ.....

백지연 : 유레카 서버 의존성 충돌 때문에 다시 user 들어갑니다...(해결 완)

▼ 오늘 진행 계획



Retrospect

회고

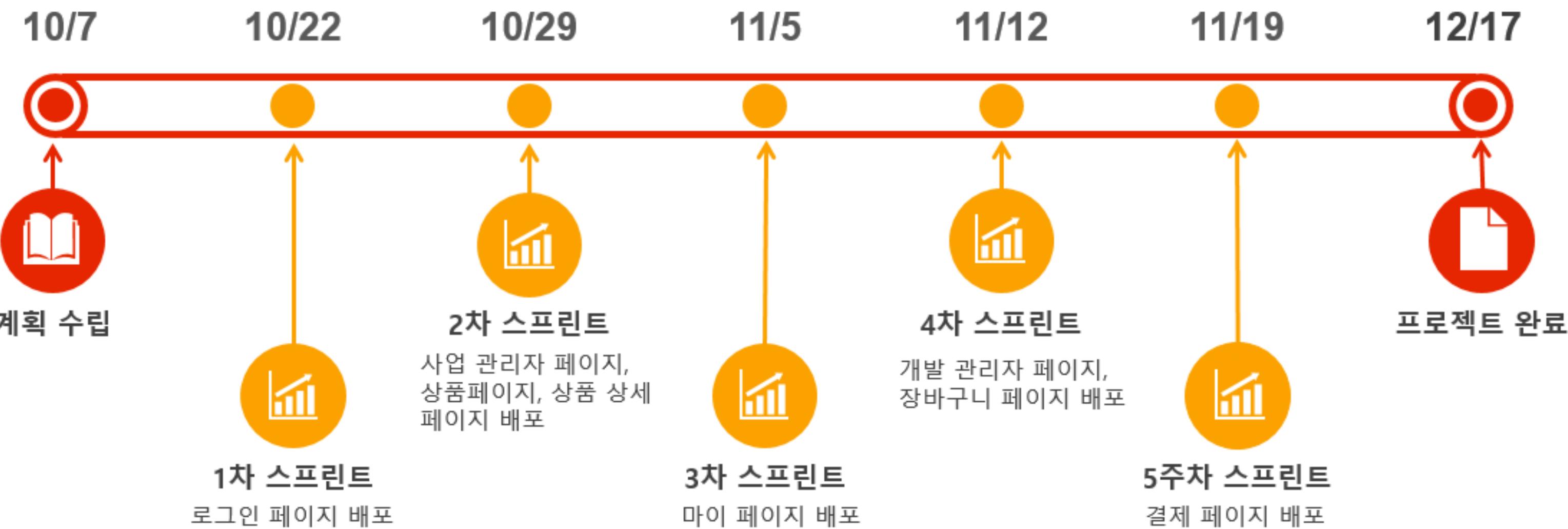
- Liked: 당신이 만족스러웠던 점(파랑)
- Learned: 당신이 깨달았거나 배웠던 점(초록)
- Lacked: 당신이 놓쳤거나 부족했던 점(빨강)

2024.11.1:4주차 개발 2주 회고

Liked	Learned	Lacked
<p>개인</p> <ul style="list-style-type: none"> • 난이도가 다양한 애니메이션 추가 + 커밋 규칙 너무 잘 지킴 <p>기능 개발 후의 과정 (sqlite로 db생성, flask로 라우터 올려서 실행, 도커 파일 생성)을 진행하여 사용하는 이유와 방법을 더 명확히 이해하게 됨</p> <p>개인적으로 회고문이나 사용자 인터페이스, 다양한 관심분야에 대한 글쓰기를 더 노력하고 경험을 쌓아가면서 단계에서 단계에서 좋아집니다.</p>	<p>개인</p> <ul style="list-style-type: none"> • 어려운 애니메이션 추가 + 개인 디디 다른 부분 개발 들어가서 신나게 <p>각종 aws 서비스들을 사용하거나 역할을 맡아서 다양한 경험을 통해 학습하는 등 aws를 다루는 것 자체가 자연스러워져 만족스러웠다.</p> <p>전반적으로 기술적 지식에 대한 관심을 가지고 있어가지고 있고 있음</p> <p>팀</p> <ul style="list-style-type: none"> • 칭찬을 아끼지 않음 + 상황 공유가 빠름 <p>모르는 부분과 구조적인 부분을 짚어서 남기면서 좋은 대화로 일어나고 있다.</p> <p>개인</p> <ul style="list-style-type: none"> • 개인 디디 다른 부분 개발 들어가서 신나게 <p>개인으로 개인 디디 다른 부분 개발 들어가서 신나게</p> <p>팀</p> <ul style="list-style-type: none"> • 서로 도움요청시 문제 해결에 적극적으로 기여 <p>제작된 결과물을 잘 살피고 평가하는 것과 함께 서로에게 원하는 바를 말하거나 원하는 것을 해주는 것이 한데입니다.</p> <p>팀</p> <ul style="list-style-type: none"> • 프론트엔드 앱 텍시 구성도 (Layer 형식, Slice 형식, Segment 형식) <p>조금씩 업데이트되고 소통을 통해 업데이트되는 것과 같은 구조를 이해하게 됐습니다.</p> <p>팀</p> <ul style="list-style-type: none"> • 프론트엔드 앱 텍시 구성도 (Layer 형식, Slice 형식, Segment 형식) <p>제작된 결과물을 잘 살피고 평가하는 것과 함께 서로에게 원하는 바를 말하거나 원하는 것을 해주는 것이 한데입니다.</p> <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>API Gateway 같은 경우도 이런 계기로 향후 서비스를 확장하는 계획과 세부적인 구조를 따라 구현을 전향적으로 실현이 가능합니다.</p> <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>전반적으로 기술적 지식에 대한 관심을 가지고 있고 있음</p>	<p>개인</p> <ul style="list-style-type: none"> • 커밋 컨센서스 및지 암기 <p>- 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다.</p> <p>개인</p> <ul style="list-style-type: none"> • 숙취이슈, 컨디션 조절 실패 <p>날씨에 따라 개인에게 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다.</p> <p>개인</p> <ul style="list-style-type: none"> • 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다. <p>팀</p> <ul style="list-style-type: none"> • 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다. <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>전반적인 구조 틀을 풀어서 다른 사람에게 설명하는 능력이 있는 것 같았습니다. 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다.</p> <p>팀</p> <ul style="list-style-type: none"> • 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다. <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>전반적으로 기술적 지식에 대한 관심을 가지고 있고 있음</p> <p>개인</p> <ul style="list-style-type: none"> • 개인 커밋에 따라 영향을 미친다는 걸 알게 되었고 개인 커밋을 보 할 수 있도록 조치를 취하여 한다는 것을 알게 되었습니다. <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>전반적으로 기술적 지식에 대한 관심을 가지고 있고 있음</p> <p>팀</p> <ul style="list-style-type: none"> • 디비 엔진 선택 (MySQL vs PostgreSQL) <p>전반적으로 기술적 지식에 대한 관심을 가지고 있고 있음</p>



Milestone 마일스톤





Role

역할

● Back-End

우예리
백지연
이선희

MSA 기반 BackEnd 개발

● Front-End

강민수
장나영

사용자 맞춤형
UI 개발

● Scrum Master : 우예리

● DevOps

황서정
김민재

AWS 기반 CI/CD 파이프라인 구축
Terraform으로 인프라 관리 및 배포
Helm과 ArgoCD를 활용한 Kubernetes
애플리케이션 배포 자동화

● AI

정우석

사용자 맞춤형 추천
시스템,
상품별 솟폼 추천
시스템 개발

● Platform Pipeline

홍민혁

전체 Data pipeline 구현,
AI BackEnd 개발 및 배포,
추천 시스템 개발

INDEX

프로젝

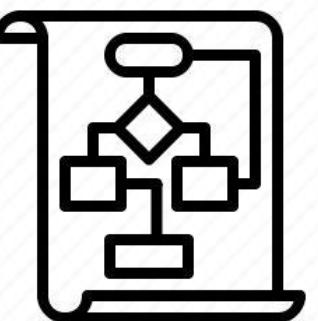
트

결과



1. 기술 스택

- 각 파트 별 기술 스택 소개



2. 아키텍처

- 각 파트 별 아키텍처 소개



3. 시연

- 프로젝트 시연 및 마무리



Event storming

이벤트

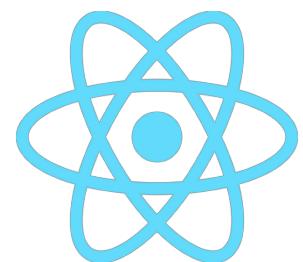




Technology stack

기술 스택

- Front-End



sweetalert2



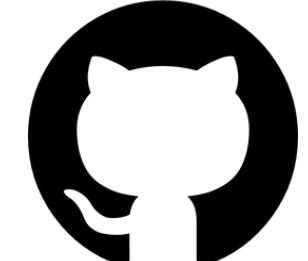
- Back-End



MariaDB



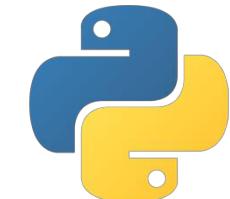
- DevOps



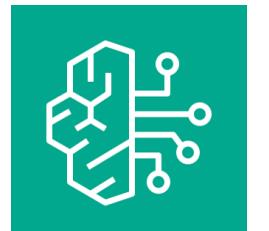
Terraform



- AI



PyTorch



Hugging Face





Sprint Management

FrontEnd

스프린트 관리

사업자 승인 요청	민수	2024/10/22 → 2024/10/22	● 완료	RF-98	github.com/SK-.../RF-98
상단 메뉴(헤더)	나영	2024/10/22 → 2024/10/22	● 완료	RF-66	github.com/SK-.../RF-66
물품 목록 확인(기업)	민수	2024/10/22 → 2024/10/23	● 완료	RF-52	github.com/SK-.../RF-52
주문 목록 확인(기업)	민수	2024/10/22 → 2024/10/23	● 완료	RF-56	github.com/SK-.../RF-56
물품 등록 (기업)	나영	2024/10/22 → 2024/10/23	● 완료	RF-127	github.com/SK-.../RF-127
물품 등록 로딩페이지 (기 업)	민수	2024/11/15 → 2024/11/15	● 완료	RF-249	github.com/SK-.../RF-249
메인페이지(Home)	민수	2024/10/22 → 2024/10/30	● 완료	RF-42	github.com/SK-.../RF-42
상품페이지(Shop)	나영	2024/10/29 → 2024/10/30	● 완료	RF-57	github.com/SK-.../RF-57
상품 상세 페이지(Shop- detail)	민수	2024/10/29 → 2024/10/30	● 완료	RF-65	github.com/SK-.../RF-65
브랜드 페이지(Brands)	나영	2024/10/29 → 2024/10/30	● 완료	RF-126	github.com/SK-.../RF-126
주문조회(Mypage)	나영	2024/10/30 → 2024/11/06	● 완료	RF-60	github.com/SK-.../RF-60
위시리스트(Mypage)	민수	2024/10/30 → 2024/11/06	● 완료	RF-50	github.com/SK-.../RF-50
1:1 문의(Mypage)	민수	2024/10/30 → 2024/11/06	● 완료	RF-47	github.com/SK-.../RF-47
정보수정(Mypage)	민수	2024/10/30 → 2024/11/06	● 완료	RF-46	github.com/SK-.../RF-46
로그아웃(Mypage)	나영	2024/10/30 → 2024/11/06	● 완료	RF-59	버튼 기능
회원탈퇴(Mypage)	민수	2024/10/30 → 2024/11/06	● 완료	RF-48	버튼 기능
사용자 목록(관리자)	민수	2024/11/05 → 2024/11/08	● 완료	RF-61	github.com/SK-.../RF-61
사업자 승인/거절(관리 자)	나영	2024/11/05 → 2024/11/08	● 완료	RF-62	github.com/SK-.../RF-62
물품 목록 확인(관리자)	민수	2024/11/05 → 2024/11/08	● 완료	RF-63	github.com/SK-.../RF-63
장바구니	민수	2024/11/05 → 2024/11/15	● 완료	RF-49	github.com/SK-.../RF-49



Figma UI 설계

The image displays a grid of Figma wireframes for a shopping website, each labeled with its corresponding file name:

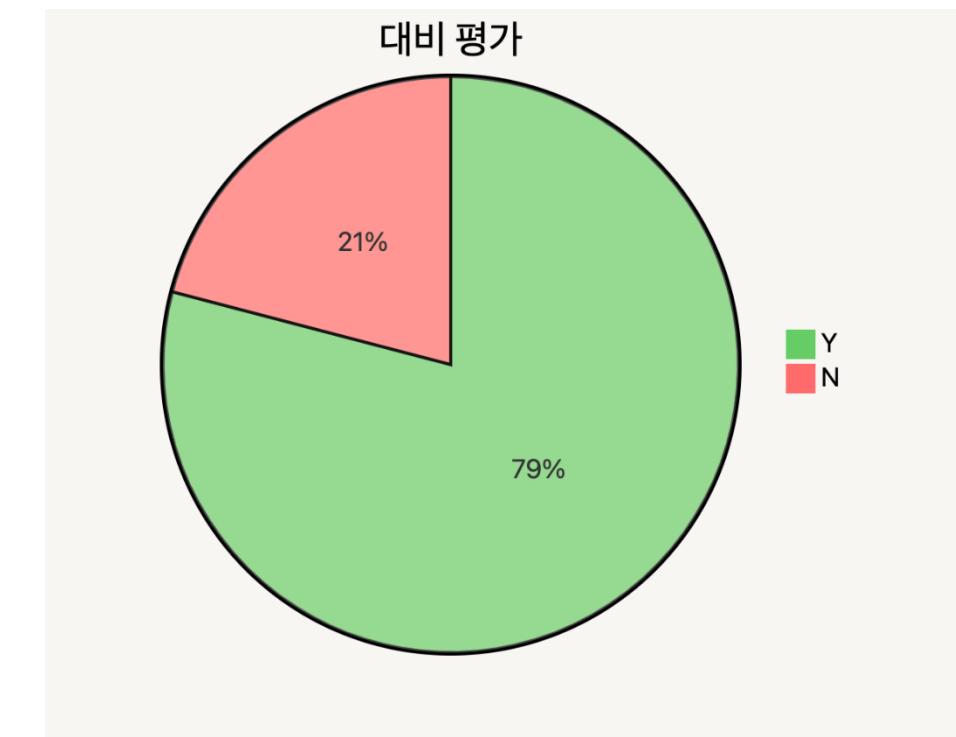
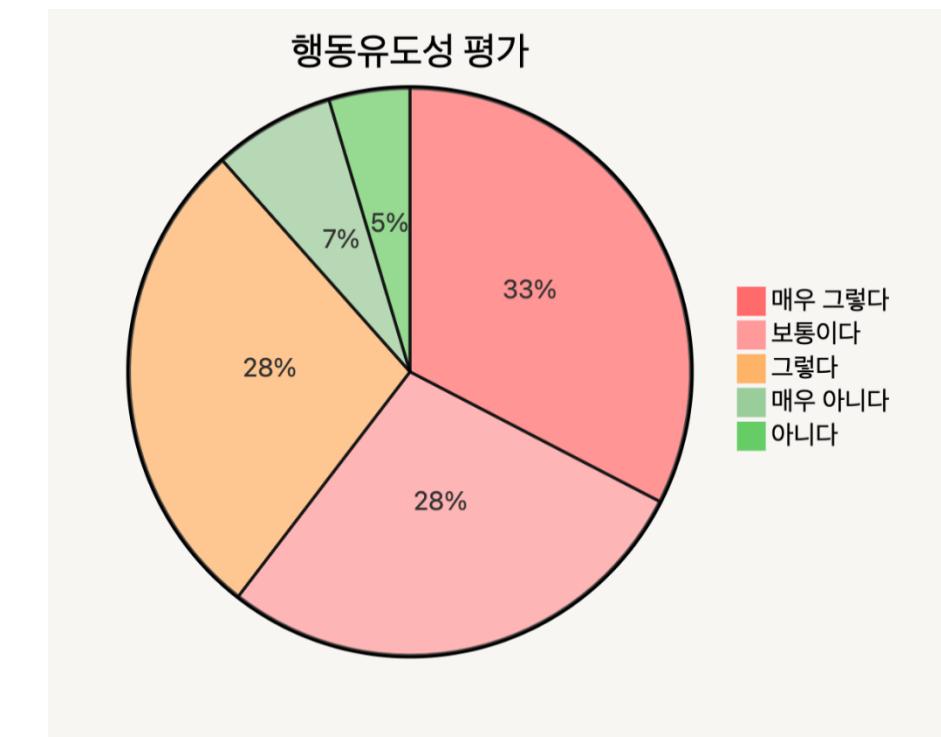
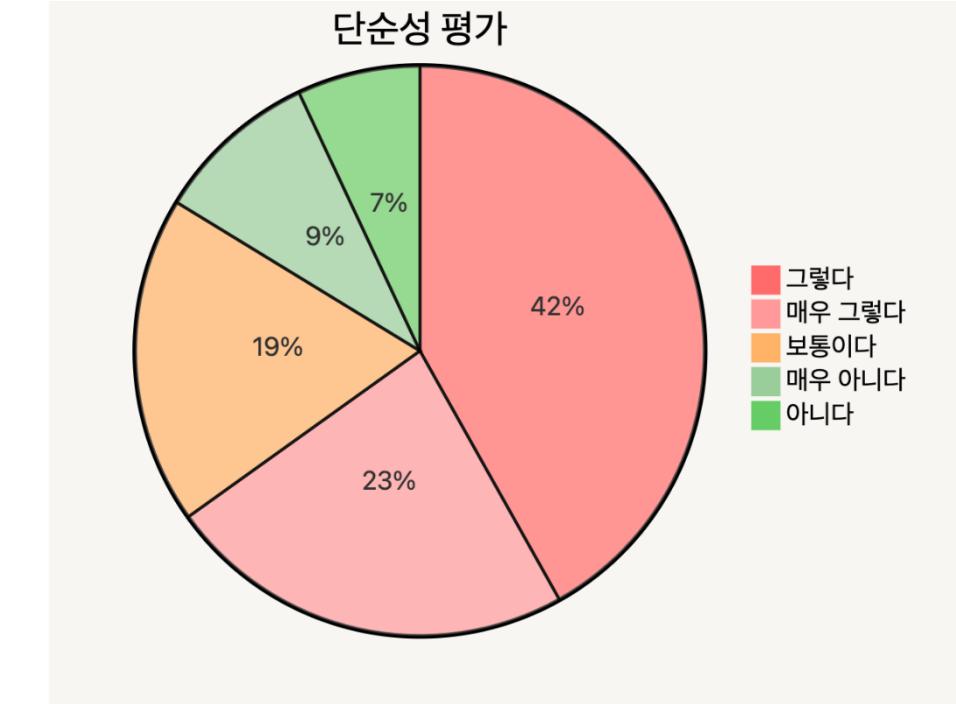
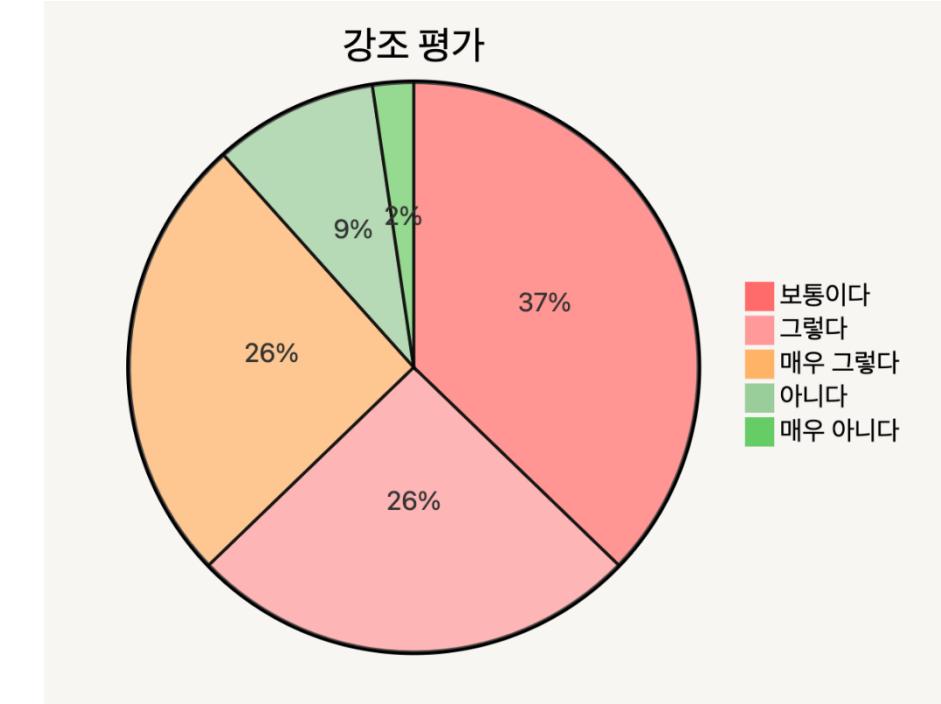
- Home.js**: Home page featuring a large banner with "쇼핑몰에 대한 설명" and "사람 이미지", followed by a "Short-Form" section with three items labeled "AI 기반 솔품".
- Shop.js**: Product listing page titled "Category" showing a 2x3 grid of products labeled "상품".
- 상품 상세 페이지 - 광고(숏츠)**: Product detail page for a short-sleeved shirt, showing a large image, price (10W 200,000), and descriptions for "광고", "상품리뷰", and "상세정보".
- AboutUs.js**: About Us page featuring a grid of team members categorized as "Leader & Back-end", "Front-end", "Back-end", and "DevOps".
- Brand.js**: Brand page listing brands like NIKE, ADDIAS, POLO, GAPT, and MAISON MARGIELA, with a product grid below.
- Mypage_Cart.js**: My Page/Cart page showing a shopping cart with two items: a white shirt and grey pants, along with a "내가 만든 장바구니 이동" button.



UI 개선 설문 조사

● 결과 ●

UI 개선 사항

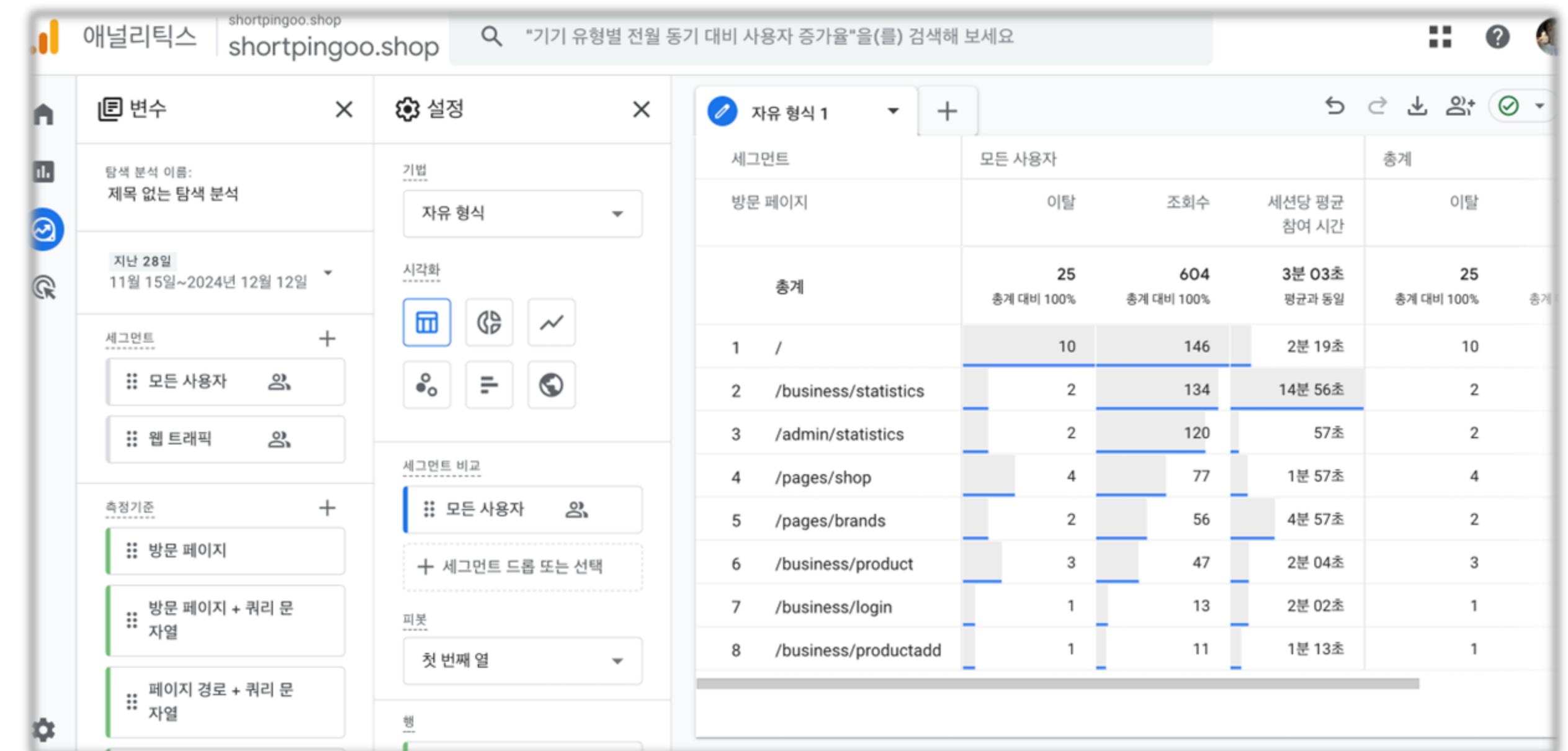




통계페이지

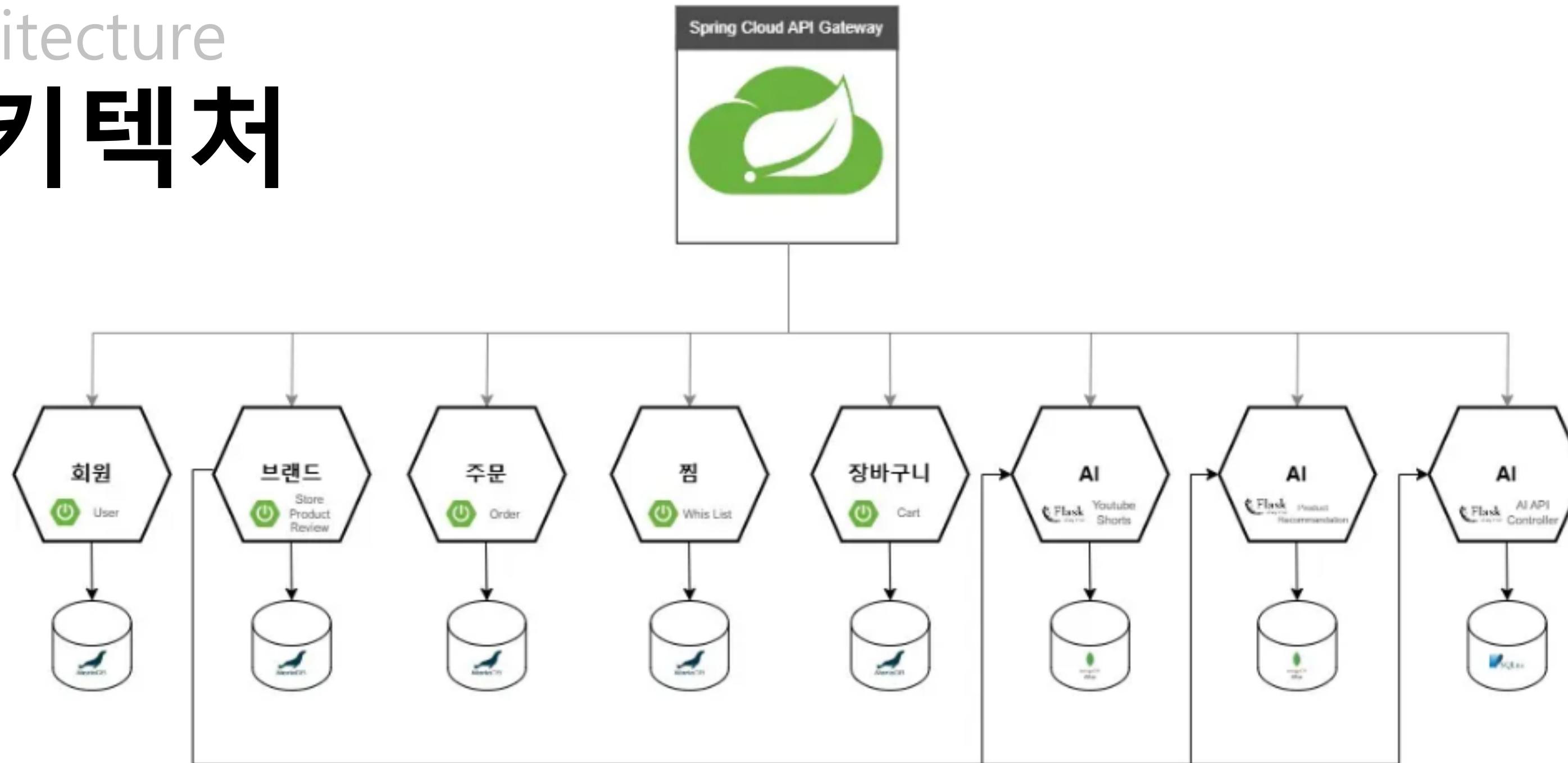
Google

Analytics





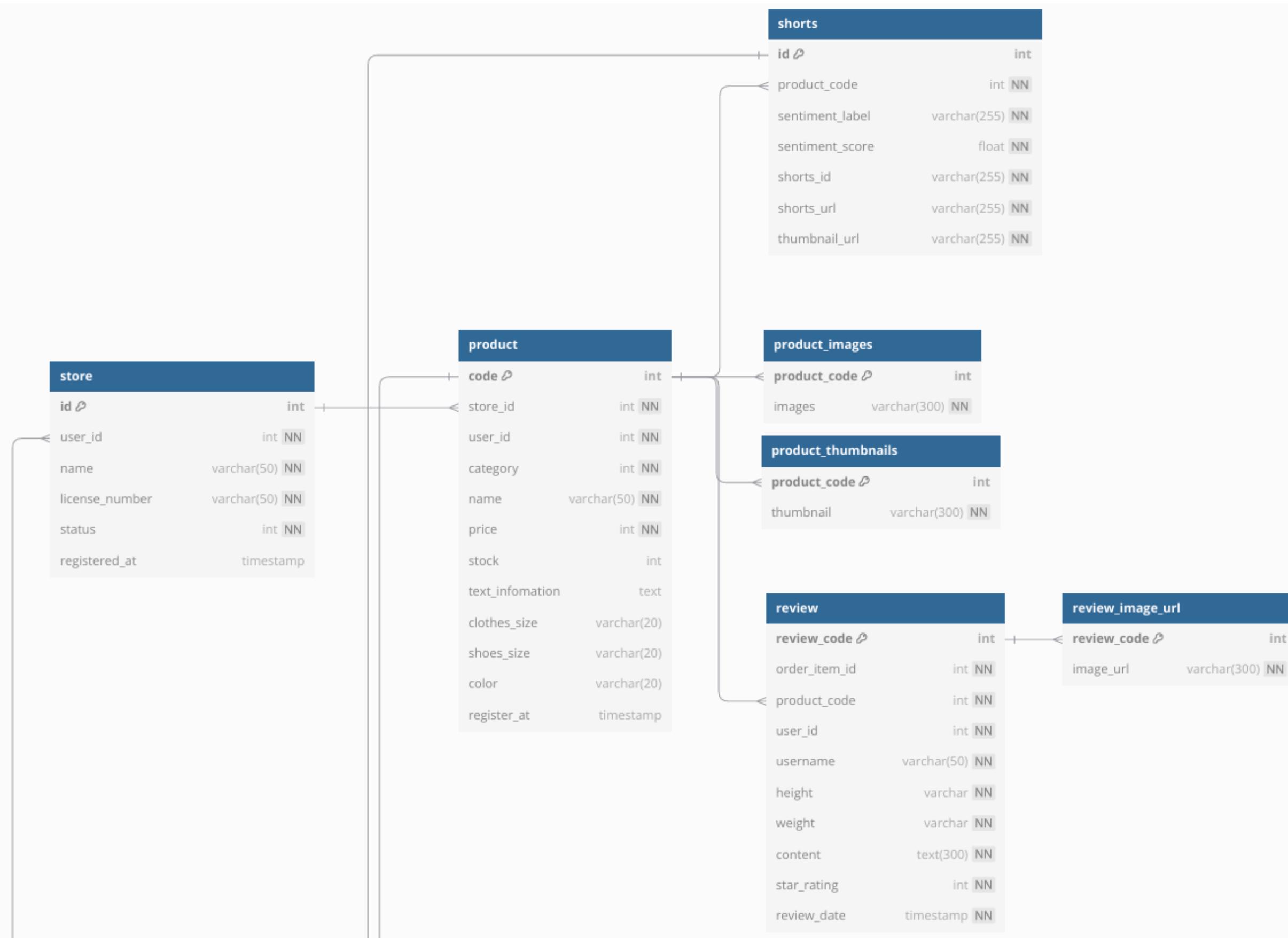
Architecture 아키텍처





가게, AI, 상품

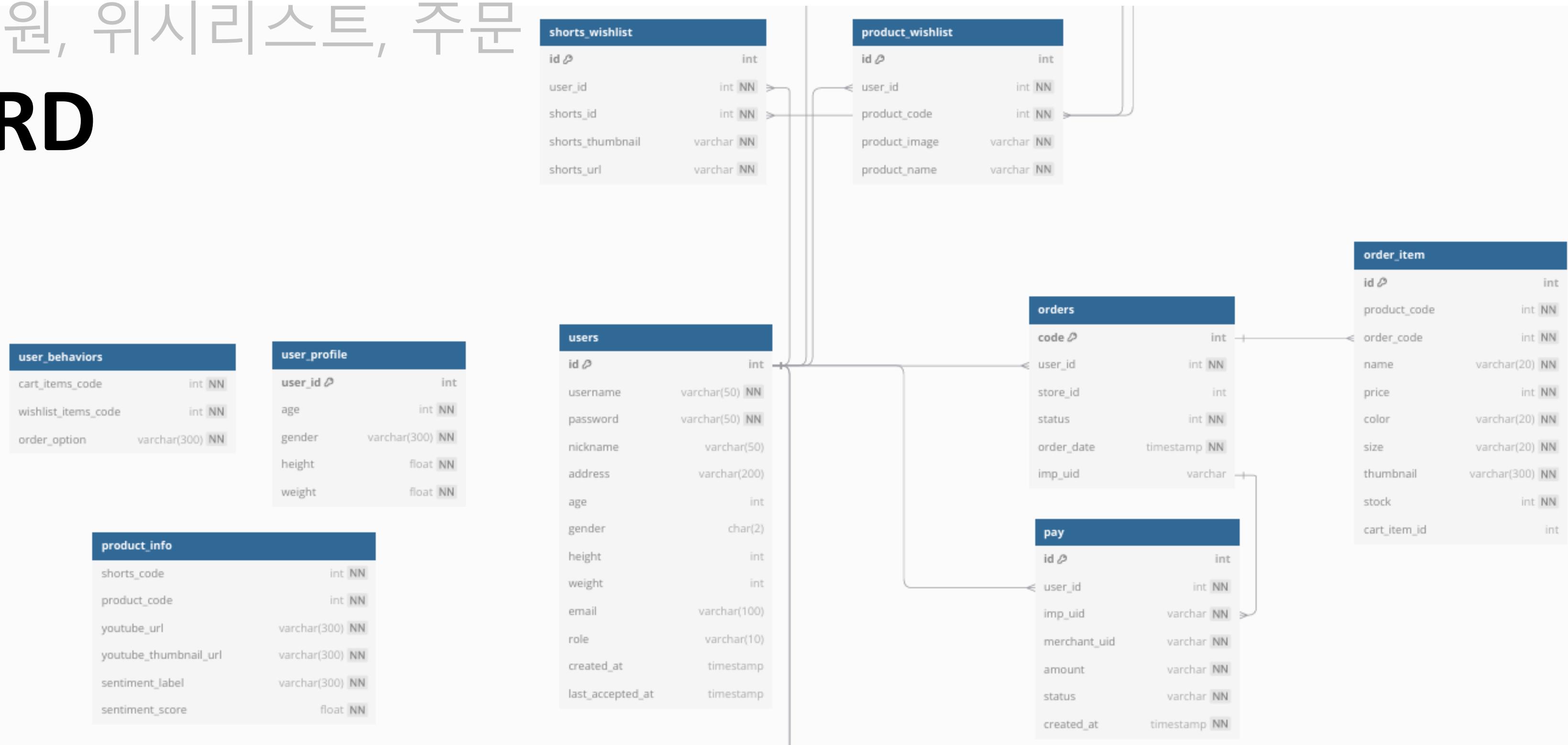
ERD





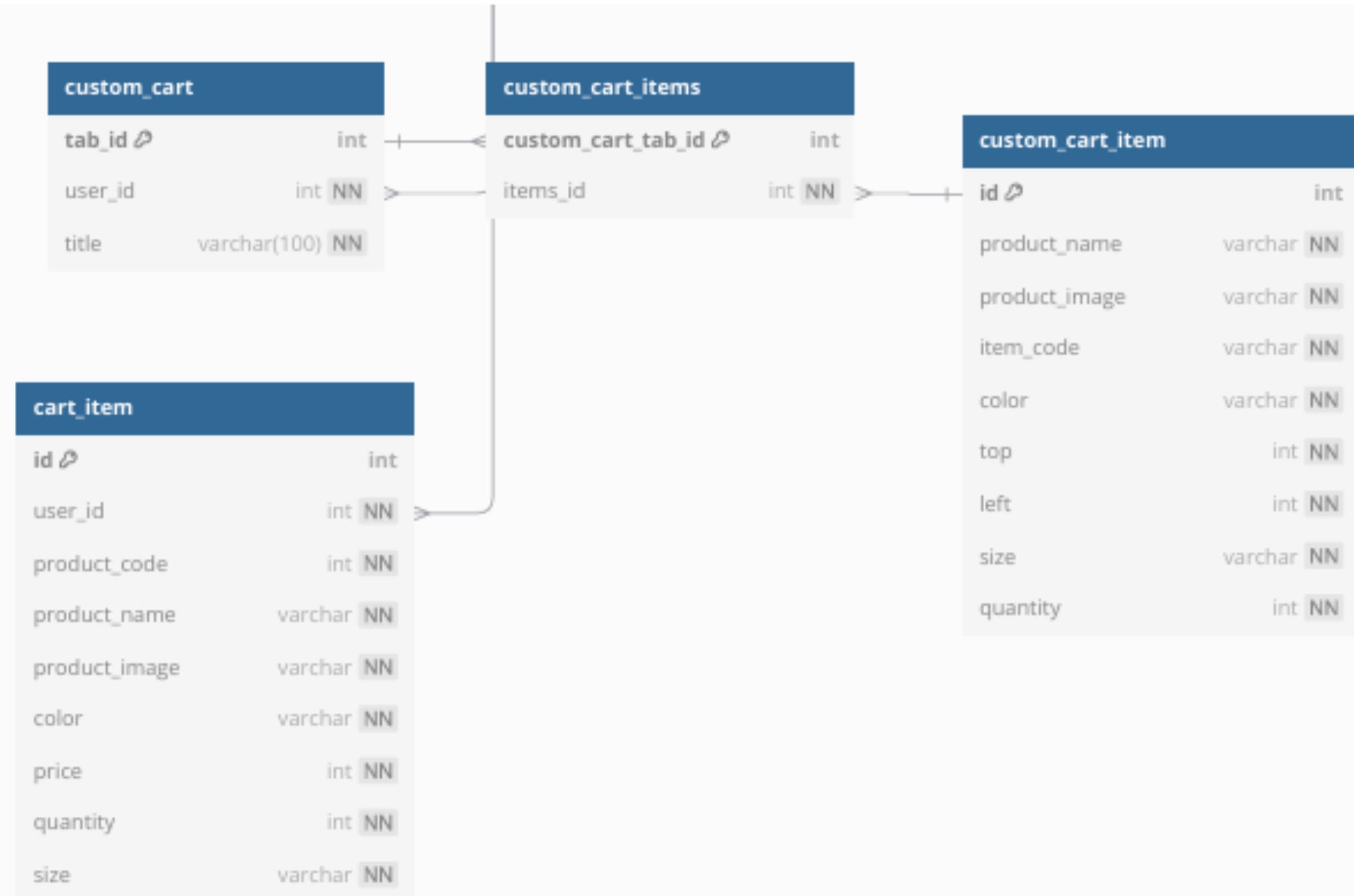
회원, 위시리스트, 주문

ERD





장바구니 ERD





Application Programming Interface

API 명세

Request

```
{  
    "status": 0  
}
```

Path Variable

이름	타입	필수	규칙	설명
userId	int	O	role: master	개발관리자 식별키
storeId	int	O	5****	가게 식별키

Parameters

이름	타입	필수	규칙	설명

Response

```
{  
    "id": 50018,  
    "userId": 202,  
    "name": "보테가",  
    "licenseNumber": 123,  
    "status": 1,  
    "registeredAt": null  
}
```

Header

키	값	설명
Authorization	JWT	Bearer 미포함

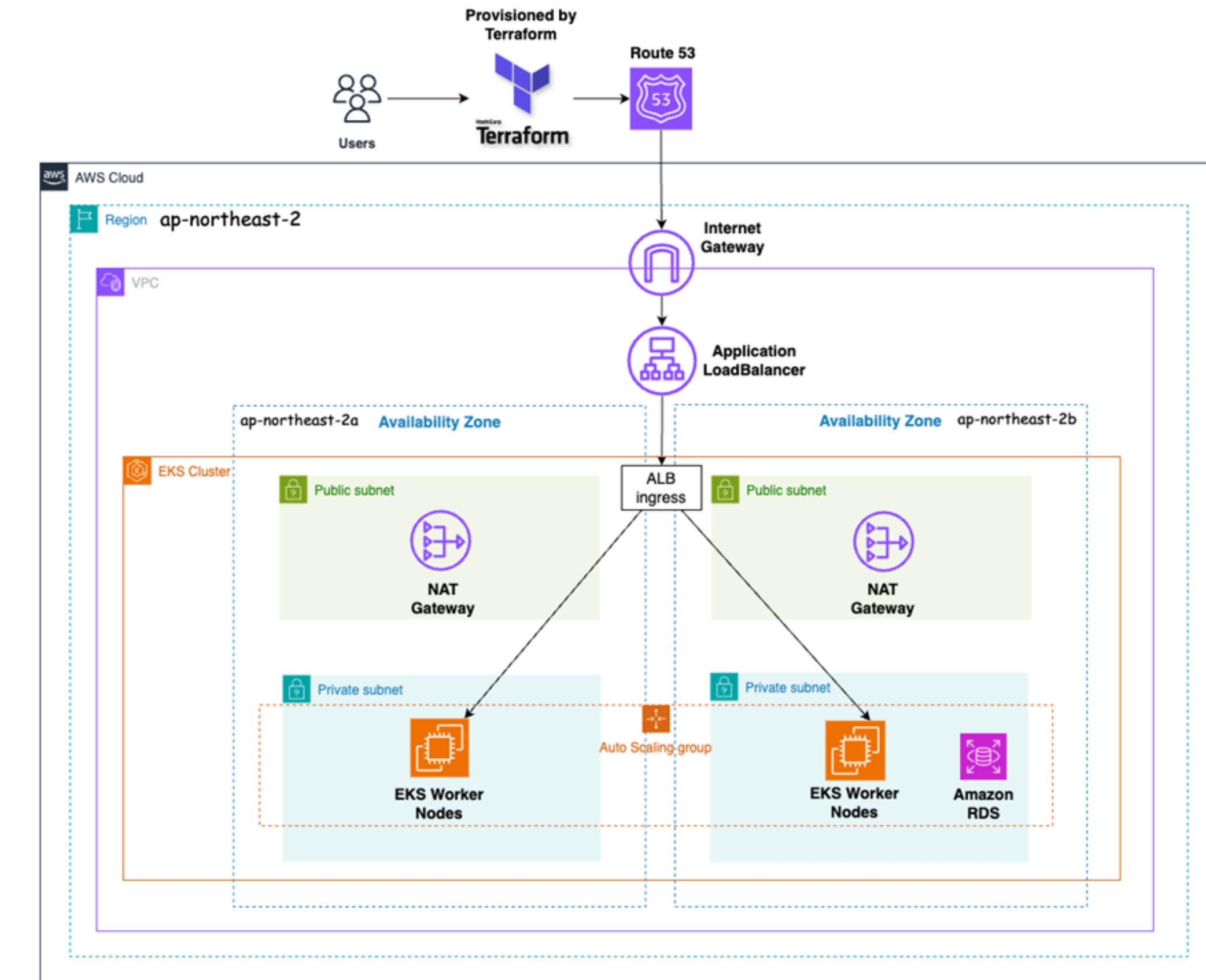
API 명세서 (Base URL : /api)

↑ 도메인 ↓

도메인	Aa 기능	HTTP 메서드	API Path
user	회원가입	POST	/open-api/user/register
user	로그인	POST	/open-api/user/login
user	회원 정보 조회	GET	/api/user
user	회원 정보 수정	PATCH	/api/user/update
user	회원 탈퇴	DELETE	/api/user/exit/{id}
user	회원 정보 전체 조회	GET	/api/user/master
user	회원 강퇴	DELETE	/api/user/master/exit/{targetId}
user	회원 ID 조회	GET	/api/user/{id}
cart	장바구니 일반 담기	POST	/api/cart/{userId}/product
cart	장바구니 페이지	GET	/api/cart/{userId}
cart	장바구니 상품 일반 수정	PATCH	/api/cart/{userId}/product
cart	장바구니 상품 일반 제거	DELETE	/api/cart/{userId}/product/{productId}
cart	장바구니 커스텀 담기	POST	/api/customCart/{userId}/product
cart	장바구니 상품 커스텀 수정	PATCH	/api/customCart/{userId}/product/{productId}
cart	장바구니 상품 커스텀 제거	DELETE	/api/customCart/{userId}/product/{productId}
order	주문하기	POST	/api/order/{userId}

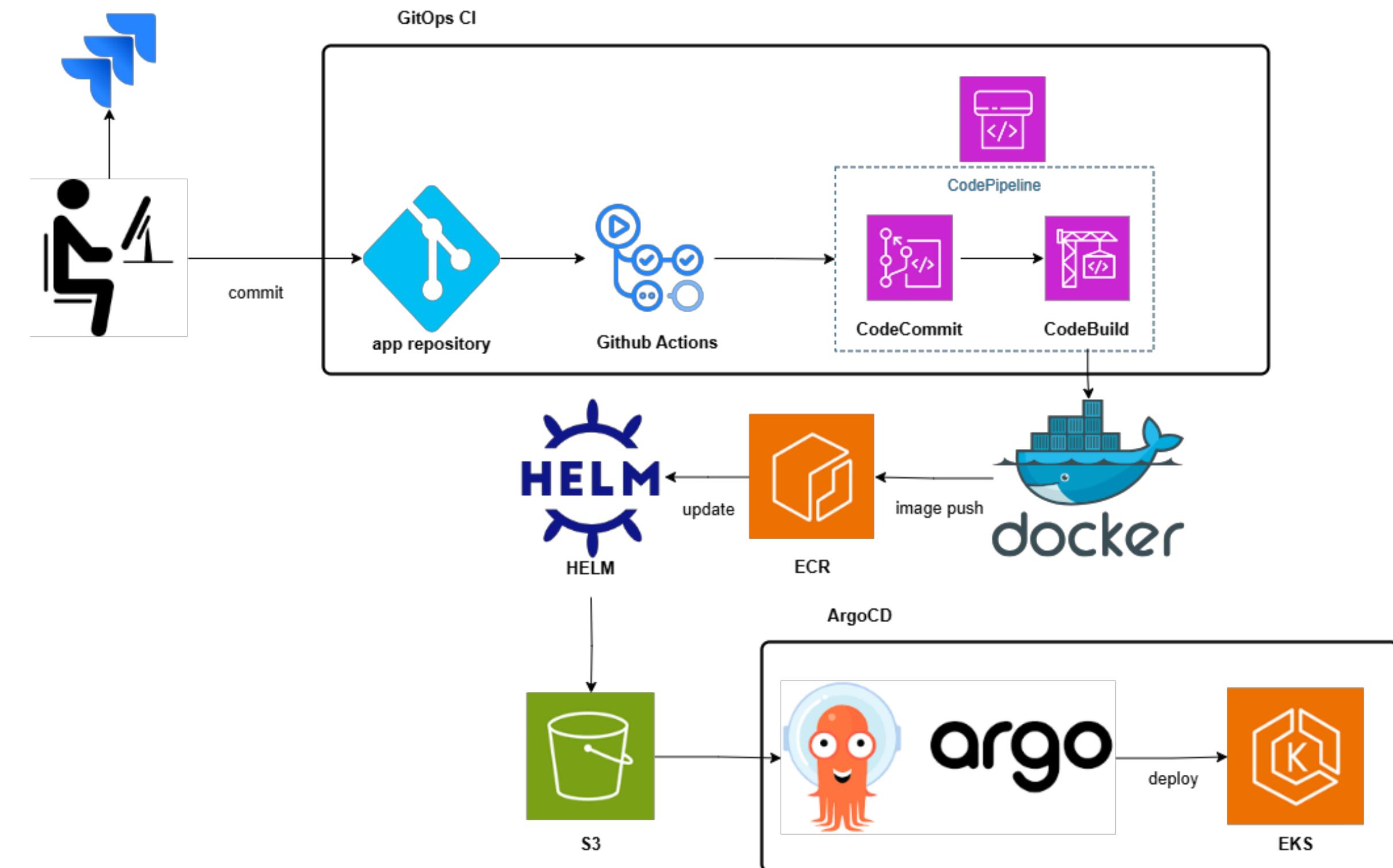


Infra 아키텍 처





DevOps 아키텍처





Infra & DevOps

아키텍처 영상

The screenshot shows a code editor with a dark theme displaying Terraform configuration files. The left sidebar shows a file tree for a directory named 'EKS-GETTING-STARTED'. The main pane displays the contents of the 'vpc.tf' file.

```
resource "aws_vpc" "rookies-final" {
    tags = tomap({
        "Name" = "rookies-final-eks-public",
        "kubernetes.io/cluster/${var.cluster_name}" = "shared",
    })
}

resource "aws_subnet" "public_subnet" {
    count = 2
    availability_zone = data.aws_availability_zones.available.names[count.index]
    cidr_block = "10.0.${count.index}.0/24"
    map_public_ip_on_launch = true
    vpc_id = aws_vpc.rookies-final.id
}

tags = tomap({
    "Name" = "rookies-final-eks-public",
    "kubernetes.io/cluster/${var.cluster_name}" = "shared",
})
```

The configuration defines an AWS VPC with two public subnets. Each subnet has a specific CIDR range and is associated with the VPC's ID. Tags are applied to both the VPC and the subnets, indicating they belong to the 'rookies-final-eks-public' cluster.



AI & Data Pipeline

사용자 활동 데이터 기반 상품 추천 시스템

개요

서비스 개요 및 주요 특징

사용자의 쇼핑 활동(구매, 위시 등록, 리뷰 등)으로 발생하는 상품에 대한 추이나 선호도를 기반, **userID**를 통해, 사용자 활동 데이터 기반 **상품 추천 정보(상품ID)**를 제공

1. LLM을 통해 학습 가능한 상품 데이터 특성 추론 및 증강
: 전 처리 + 실제 데이터 기반 feature 확장

2. 변환된 상품 정보부터, 최종 분류 학습단까지의 파이프라인 내 데이터의 본질이 동일함
: 효과적 이식 및 확장이 용이함

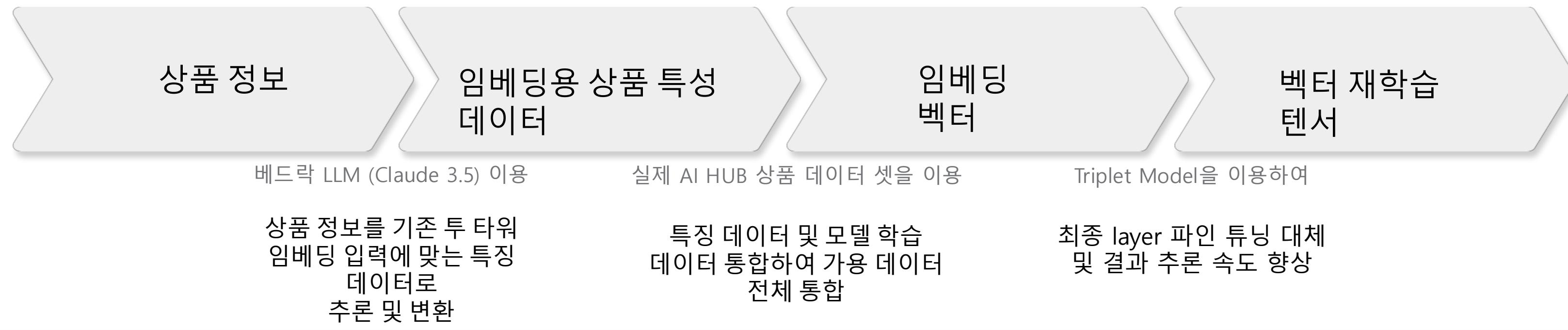
비즈니스 조건을 고려한 설계 요인

1. 기존 상품-유저 임베딩 모델을 효과적으로 이용하기 위해 베드락(LLM)으로 input을 기존 모델에 전처리 추론 및 최적화하여 제공함
2. 파이프라인 데이터 응용 전략: AI hub에 존재하는 옷 분류 데이터 셋을 기반으로 전체 파이프라인의 데이터 속성을 통일하여 데이터 확장 및 합성을 효율적으로 진행.
3. 유동적인 데이터 환경에서, 지속적인 feature set에 대한 확장이나 수렴이 가능함
: 프로젝트 기간, 인력, 실제 서비스에서 파생되는 데이터의 여러 비즈니스 제약조건을 최소화 하기 위한 AI 시스템 설계



AI & Data Pipeline

학습 구조 및 전략



실제 데이터 구성

서비스 제공 기본 상품 정보 :
상품 이름,
상품 이미지,
상품 카테고리,
가격
색상

→
특징 데이터 추론

임베딩용 상품 특징 추론 데이터

feature :
category : [01_outer_01jacket, 01_outer_02coat ..]
gender : [male, female, both] ,
...
specification : [outer, trench coat]
reinforced_feature : [feature 값에서 존재하지 않는 특성을 생성]
Embedding Concat : [64 차원 벡터]

유저 임베딩 벡터 : [...]
상품 임베딩 벡터 : [...]
→
임베딩 레이어 (512 차원)
↓
재정렬된 상품 임베딩 벡터 Set (Triplet loss 최소화)



AI & Data Pipeline

파이프라인 내부 데이터 통합

AI Hub 상품 데이터
feature

실제 상품 정보

```
"clothes": {  
    "category": [  
        "01outer_01coat",  
        "01outer_02jacket",  
        "01outer_03jumper",  
        "01outer_04cardigan",  
        :  
    ],  
    "season": ["spring&fall", "summer", "winter"],  
    "fiber_composition": ["Cotton", "Hemp", "cellulose fiber Others", "Silk",  
    "elasticity": ["none at all", "none", "contain", "contain little", "conta  
    "transparency": ["none at all", "none", "contain", "contain little", "con  
    "isfleece": ["fleece_contain", "fleece_none"],  
    "color": ["Black", "White", "Gray", "Red", "Orange", "Pink", "Yellow", "B  
    "gender": ["male", "female", "both"],  
}
```

상품 특징 추론 데이터**feature**

```
▼ clothes : Object  
  ▼ category : Array (1)  
    0: "05shoes"  
  ▶ season : Array (2)  
  ▶ fiber_composition : Array (2)  
  ▶ elasticity : Array (1)  
  ▶ transparency : Array (1)  
  ▶ isfleece : Array (1)  
  ▼ color : Array (3)  
    0: "White"  
    1: "Black"  
    2: "Mixed"  
  ▶ gender : Array (1)  
  ▼ category_specification : Array (1)  
    0: "shoes"
```

Reinforced feature

```
▼ reinforced_feature_value : Object  
  ▶ category : Array (1)  
  ▶ fiber_composition : Array (2)  
  ▶ color : Array (3)  
  ▶ category_specification : Array (1)  
  ▼ specification.metadata : Array (7)  
    0: "casual"  
    1: "sneakers"  
    2: "lace-up"  
    3: "flat sole"  
    4: "iconic design"  
    5: "versatile"  
    6: "comfortable"
```

상품 특징 추론 데이터

임베딩 벡터

임베딩 모델 INPUT에 맞게
변환

임베딩 벡터 위치 관계 재정렬(학습)

AI HUB 상품 데이터 셋

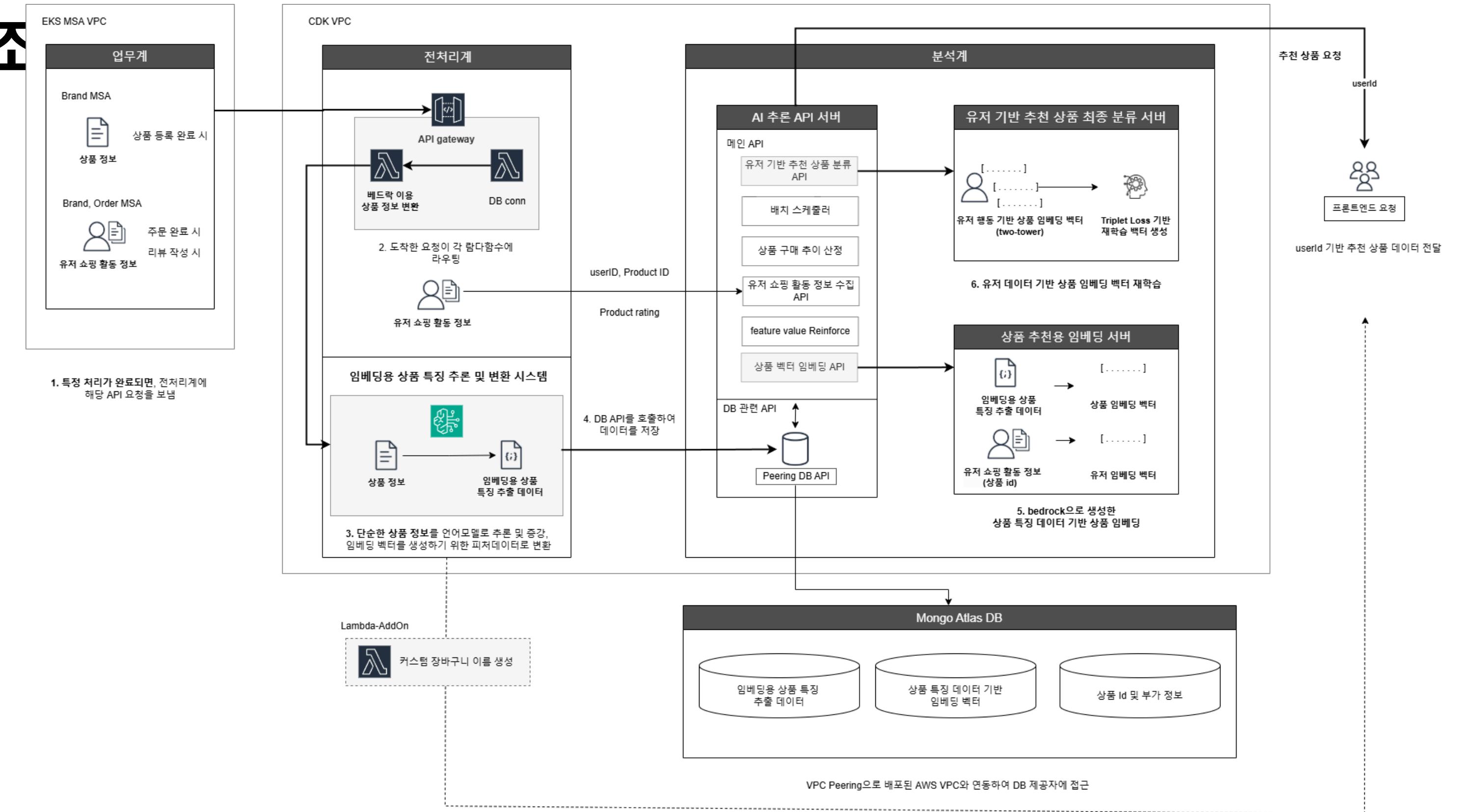
사전 모델
학습

A: 파이프라인 내 모든 데이터의 본질이
동일함



AI & Data Pipeline

전체 구조





AI & Data Pipeline

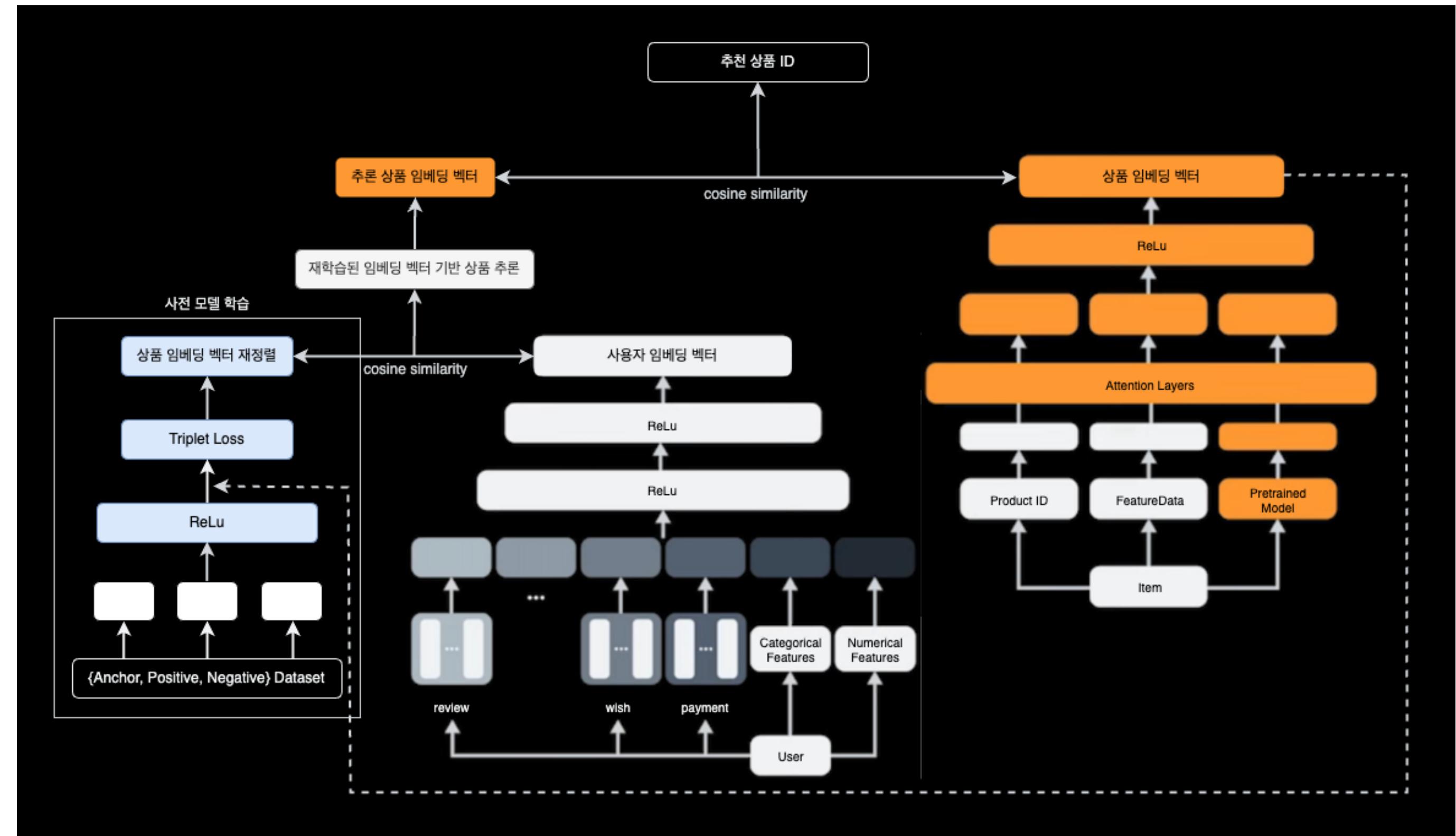
임베딩 벡터 재정렬 : Triplet 학습

학습 데이터셋으로 생성한 각각의 사전 상품 임베딩 벡터(Anchor) 간의 Positive, Negative 관계를 기준으로 Triplet 손실 함수 최소값을 구하는 과정을 통해 상품 벡터 간 거리를 재정렬.





AI & Data Pipeline

AI 모델
구조

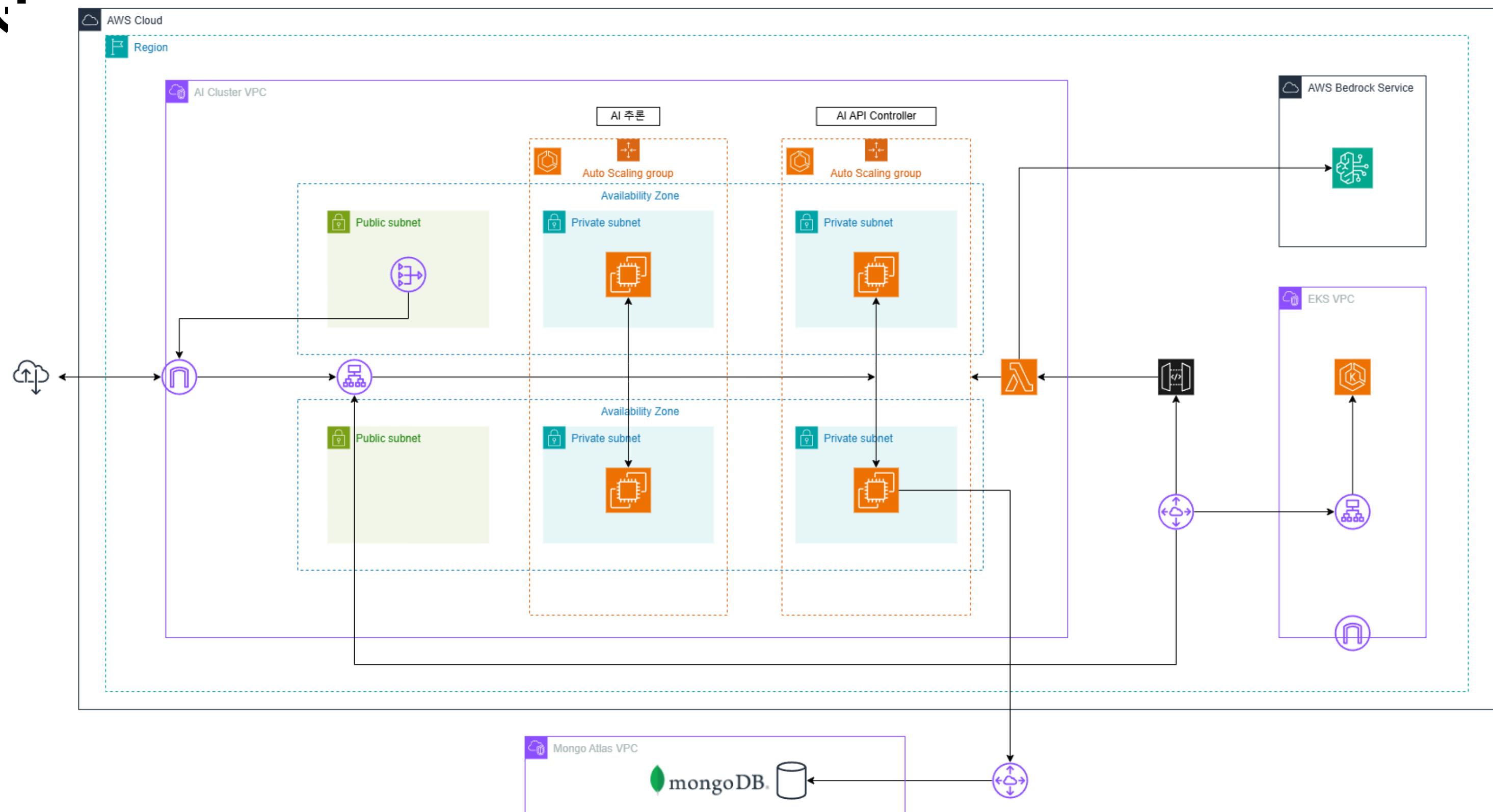


AI & Data Pipeline 아키텍처

CDK .ts 기반

업무계 - 추론계
분리

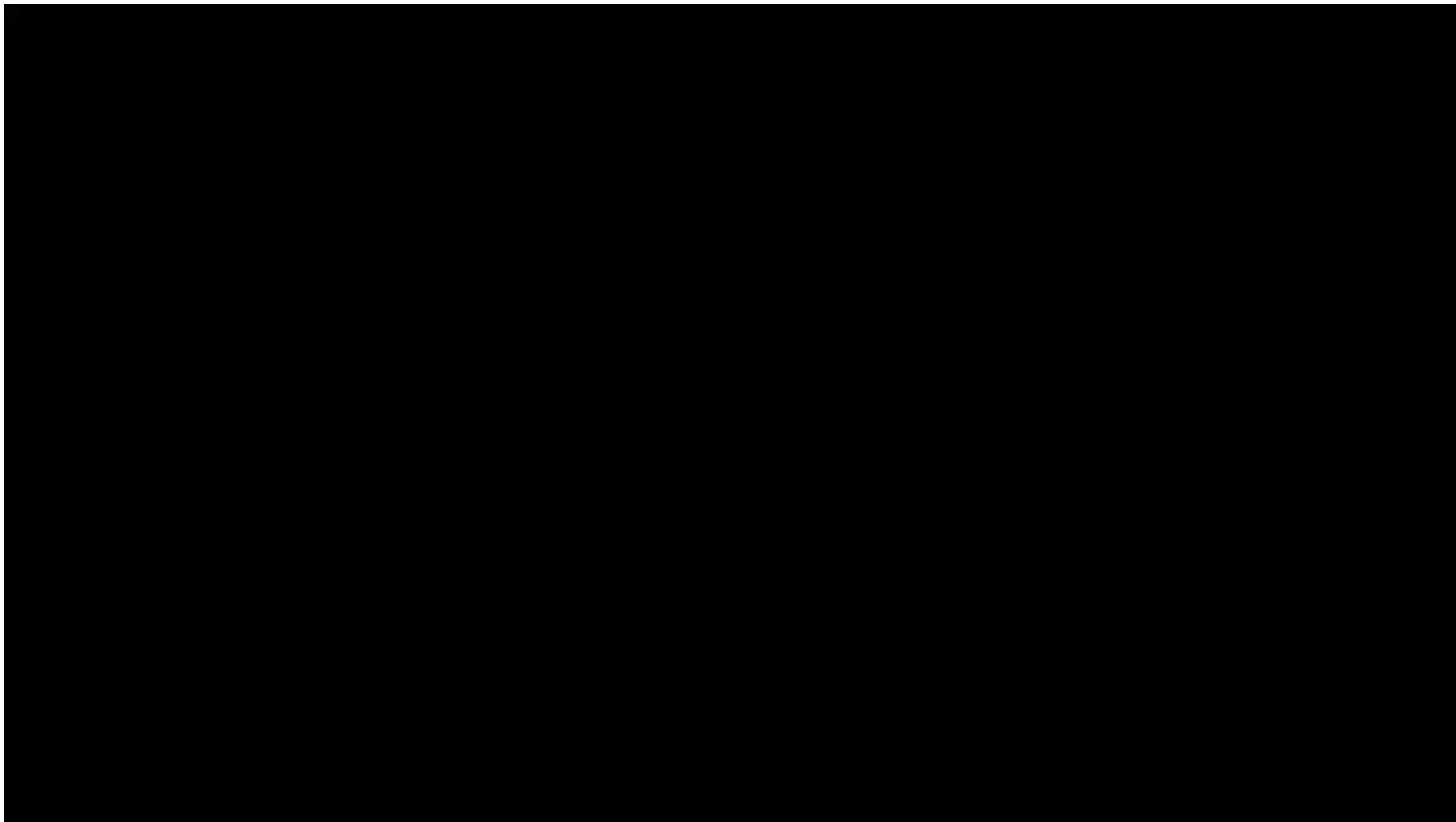
MSA EKS – AI 추론 VPC 간 Peering 기반 통신 구조





AI & Data Pipeline

아키텍처 영상





Demonstration

시연





Demonstration

시연 영상

The screenshot shows the homepage of the shortpingoo.shop website. At the top, there is a navigation bar with links for Home, Shop, Brands, and About Us. A user account icon and a button for "회원가입하고 나만의 맞춤 추천 받기!" (Join membership and get personalized recommendations) are also present. The main content area features a large, stylized title "Crafting Elegance One Story at a Time" in a serif font. Below the title, there is a paragraph in Korean: "우아함과 이야기의 결합을 통해 브랜드의 세련된 감성을 전달합니다. 저희는 단순히 제품을 제공하는 것이 아니라, 고객의 삶에 우아함과 의미를 더하는 특별한 경험을 만들어갑니다. 각 제품은 하나의 이야기처럼 정성껏 제작되었으며, 고객이 이를 통해 자신만의 이야기를 표현하고 새로운 순간을 만들어갈 수 있도록 돋는 것이 저희의 철학입니다." To the right of the text, there is a photograph of a woman wearing a white blouse and orange pants, standing in front of a circular backdrop. At the bottom left, there is a small URL: <https://shortpingoo.shop>.