프로젝트 최종 발표

인터넷 쇼핑몰 리뷰를 활용한 NLP감성분석 파이프라인

말하는 감자Team

Contents

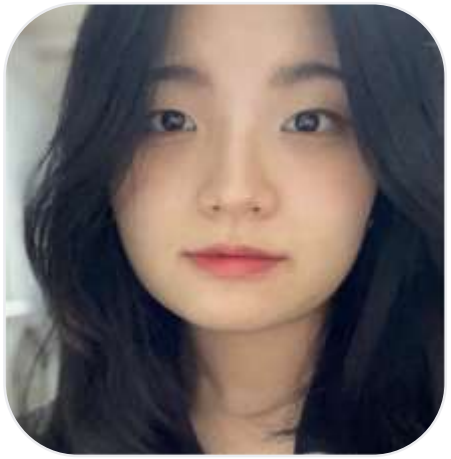
프로젝트 개요

1.1 프로젝트 역할 분담

1.2 AS-IS, TO-BE

1.3 프로젝트 일정

말하는감자 Team



도효주

- 프로젝트 팀장
- 프로젝트 전체 일정 관리
- 문서 작업
- 발표

선우지훈

- 데이터 정제 및 저장
- 대시보드 구현
- 인프라 구축
  - OpenSearch
  - Logstash
  - Kibana
  - Kafka

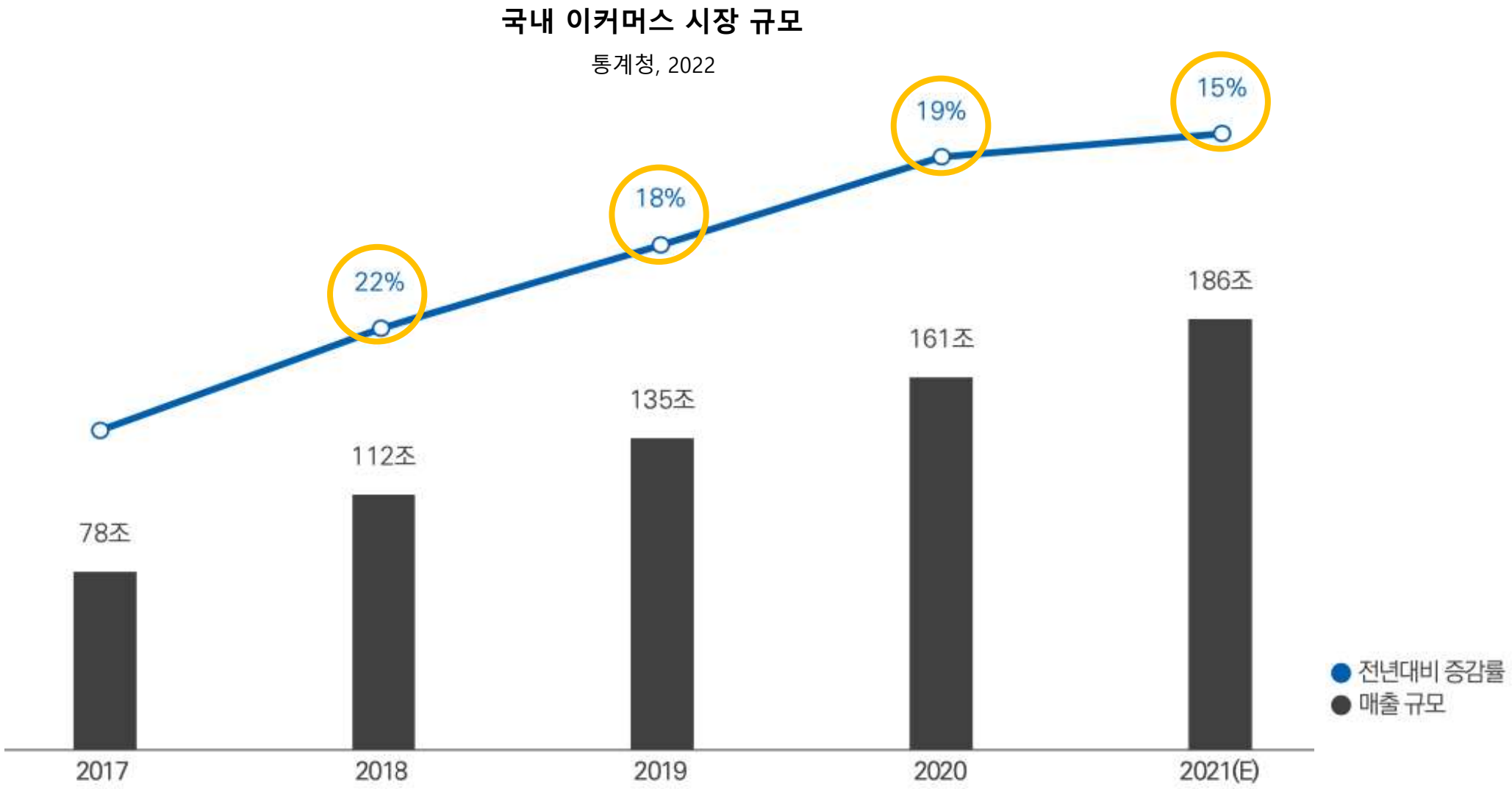
오승우

- 프로젝트 기획
- 대시보드 구현
- 데이터 분석 프로그래밍

전중석

- 데이터 크롤링
- NLP 처리
- 데이터 분석 프로그래밍
- 인프라 구축
- 대시보드 구현

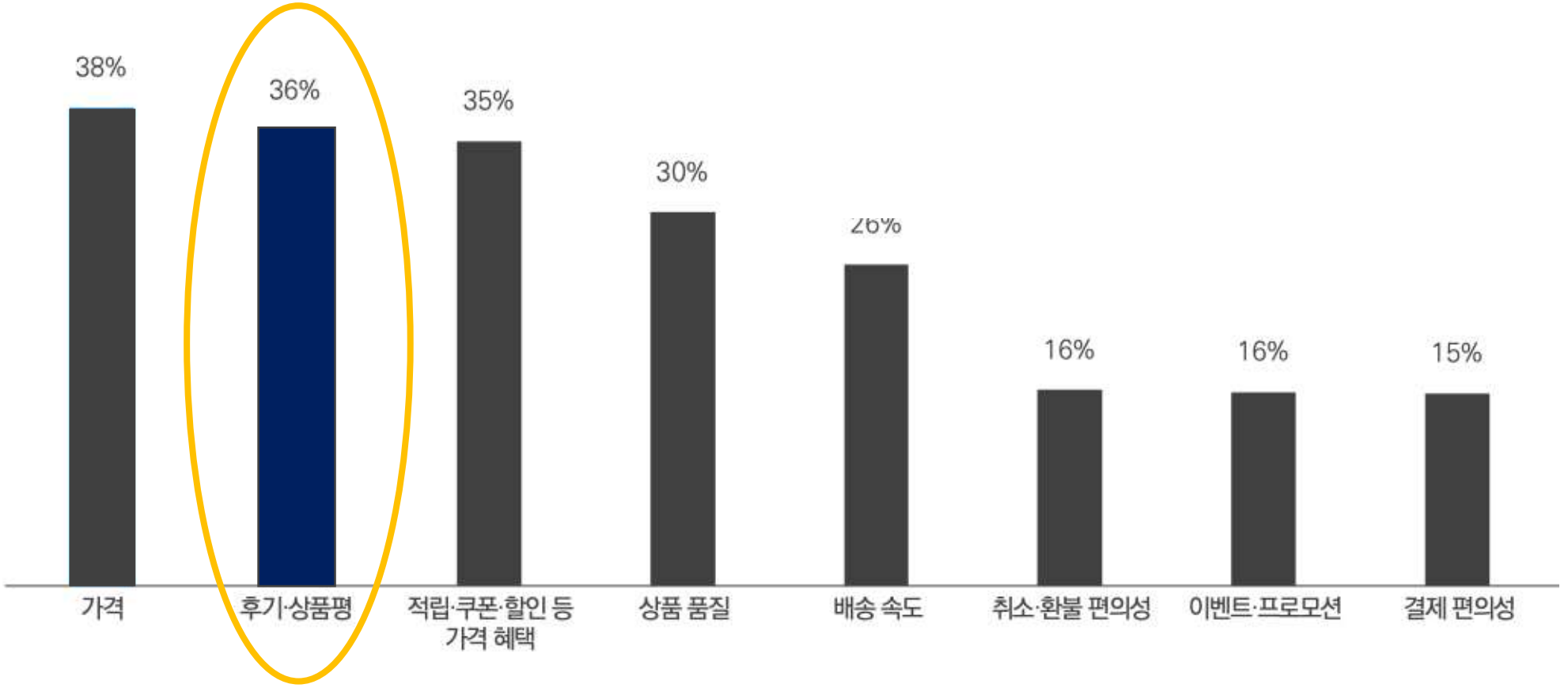
AS-IS, TO-BE


국내 이커머스 시장 규모
통계청, 2022
22%
18%
19%
15%
78조
112조
135조
161조
186조
2017
2018
2019
2020
2021(E)
전년대비 증감률
매출 규모

AS-IS, TO-BE

Q. 온라인 쇼핑 구매 고려요인

-복수 응답- 통계청, 2022



사용자 총 평점

★★★★★

4.8/5

전체 리뷰수

287,893

평점 비율

스토어PICK 판매자가 직접 선정한 베스트 리뷰입니다.

★★★★★ 5

aste***** · 22.03.31. · 상품명: 09.나토밴드 / 색상: 세도우브라운⚡...

직접 구입해 보니 가격 대비 퀄리티가 너무 좋고 줄 바꾸는
재미가 쏠쏠해서 6개 살걸 그랬나 싶어요 게다가 평일 1시
이전에 주문 시 당일도착 서비스로 쿽 배송을 받을 수 있는 점도
좋았어요 다양한 워치를 연출해 보고 싶어 스타라이트 컬러의...

★★★★★ 5

lsh2**** · 22.03.12. · 상품명: 08.모던가죽스트랩⚡BEST⚡ / 색상: ...

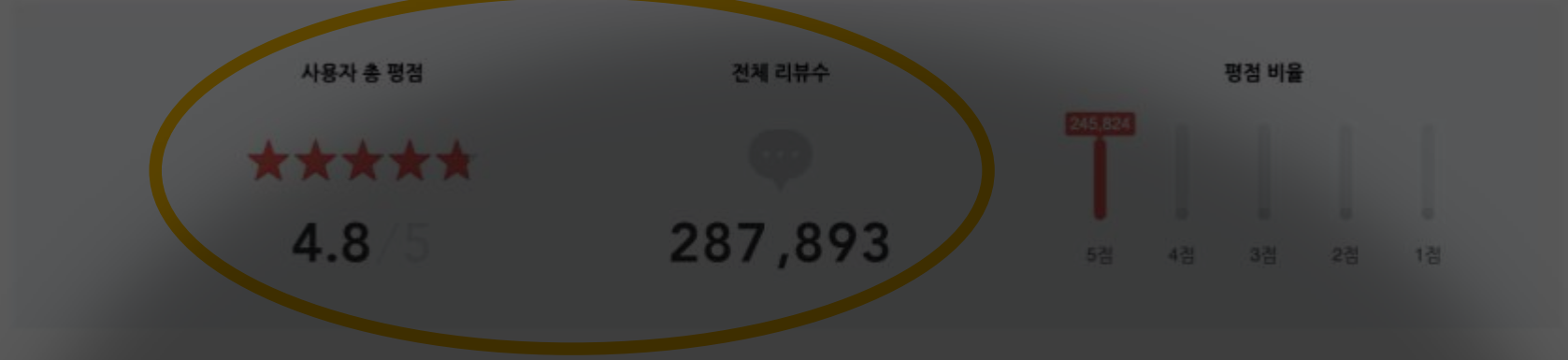
애플워치 se 40mm 실버 애플워치 스트랩을 검색해 봤는데
여러 쇼핑몰이 있었지만 우먼스9999가 후기도 많고 평도 좋았고
이벤트도 하고 있어서 구매했습니다! 구매해 보니까 왜 후기가
많은 지 알 것 같아요 저 또 살 예정입니다ㅋㅌㅋ 자세한 후기...

1/15

포토&동영상 173,009건


리뷰 288,039건

✓랭킹순 | 최신순 | 평점 높은순 | 평점 낮은순



스토어PICK 판매자가 직접 선정한 베스트 리뷰입니다.

5
aste***** 22.03.31. 상품명: 09.나토밴드 / 색상: 세도우브라운⚡...

직접 구입해 보니 가격 대비 퀄리티가 너무 좋고 줄 바꾸는
재미가 쏠쏠해서 6개 살걸 그랬나 싶어요 게다가 평일 1시

5
lsh2***** 22.03.12. 상품명: 08.오션가죽스트랩⚡BEST⚡ / 색상: ...

애플워치 se 40mm 실버 애플워치 스트랩을 검색해 봤는데
가 후기도 많고 평도 좋았고

관리하기엔 너무 방대한 양의 리뷰

1/15

포토&동영상 173,009건

리뷰 288,039건

✓랭킹순 | 최신순 | 평점 높은순 | 평점 낮은순

AS-IS, TO-BE

★★★★☆ 4

zizi****** · 22.08.05. | 신고

[1301반팔] 사이즈: L / 색상: 37.세이프티그린

너무 얇아요ㅠㅜㅠ
구매할 수록 얇아지는 느낌 ㅠㅠ
처음 구매했을때랑도
너무 다른 얇음이네요...

0

별점은 5점이지만

부정적인 내용의 리뷰들

★★★★★ 5

pass******* · 22.07.26. | 신고

[1301반팔] 사이즈: XL / 색상: 9.하버블루

18수라고 보기 힘듭니다. 너무 얇아요. 특히 목부분이 너무 쉽게 늘어납니다. 예전에 다른 곳에서 구입한 제품은 3년이 지난 지금도 두툼하도 목 부분도 늘어나지 않았는데 두번 빨아 입으니 바로 티셔츠가 엉망이 되네요 ㅠㅠ 매우 아쉽습니다.

0

★★★★★ 5

skdu******* · 22.07.24. | 신고

색상: 화이트 / 사이즈: S

예쁘긴한데 티가 너무 길어요 .. 무슨 치마수준이라 바지 안입어도 될 길이에요 ;; 수선해서 입어야할거같아여 .. 재질도 그닥 좋은 재질은 아니에요

0

분석된 데이터를 바탕으로

쇼핑몰 운영의 편리성,효율성 증대

감자 모자

리뷰 갯수

38,871
analyzed_data - Count

기존 리뷰

88.778%
긍정 - Count

10.494%
부정 - Count

평균 평점

4.592
Average Star

분석된 리뷰

82.613%
긍정 - Count

16.659%
부정 - Count

긍정 리뷰

Word: Descending - Count

부정 리뷰

Word: Descending - Count

전체 리뷰

Word: Descending - Count

자사 키워드

탁사 키워드

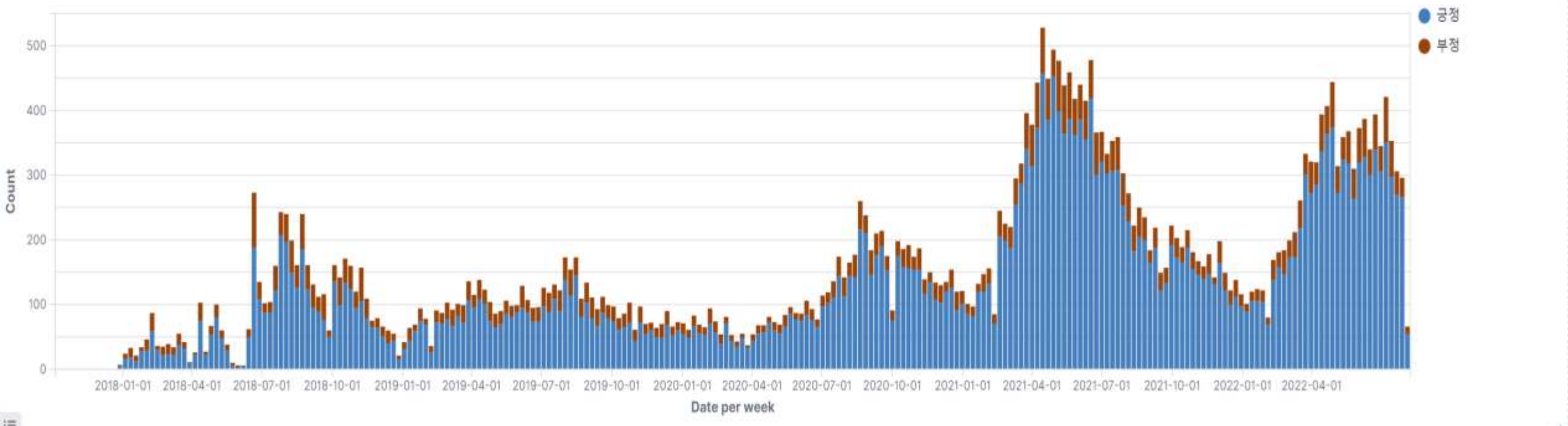
자사 키워드

가격	긍정	4,208
가격	부정	527
배송	긍정	3,788
배송	부정	216
사이즈	긍정	2,997
사이즈	부정	903
만족	긍정	3,021
만족	부정	75
구매	긍정	2,734
구매	부정	242

타사 키워드

가격	긍정	1,186
가격	부정	102
배송	긍정	1,023
배송	부정	32
만족	긍정	769
만족	부정	13
사이즈	긍정	613
사이즈	부정	130
구매	긍정	636
구매	부정	47

일자별 추이



프로젝트 일정

PHASE		DETAILS	7월																									8월									
			1주차					2주차					3주차					4주차					5주차					6주차					7주차				
	PROJECT WEEK: 7 weeks		29	30	1			4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29	1	2	3	4	5	8	9	10		
1	프로젝트 계획 및 분석	이슈 및 적용 사례 조사																																	PROJECT END		
		이슈 및 적용 사례 분석																																			
		데이터 분석 소프트웨어 조사																																			
2	프로젝트 설계	프로젝트 기획서 작성																																			
		소프트웨어 아키텍처 설계																																			
3	프로젝트 개발	ML 프로세스 모델링																																			
		데이터 저장소 프로세스																																			
		데이터 정제 프로세스																																			
		알림 프로세스																																			
		데이터 Queueing 프로세스																																			
		시각화																																			
		추가 구현																																			
4	프로젝트 테스트	분석 모델 유효성 검증																																			
		부하 테스트																																			
5	프로젝트 종료	보고서 작성																																			
		발표준비																																			

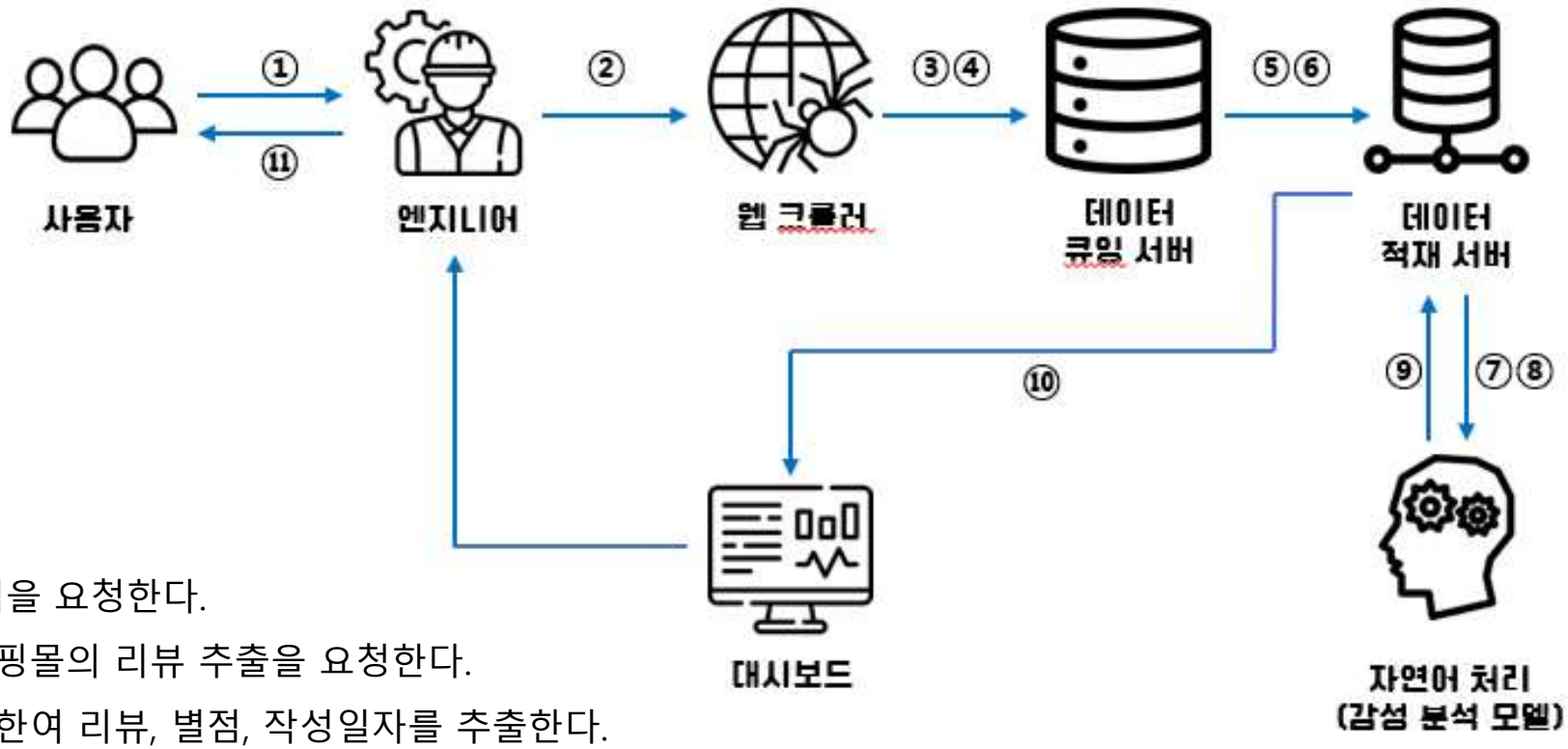
분석 및 설계 구현

프로젝트 환경구성(활용 도구)

Software	Reason for use
Python	사용자 쇼핑몰 리뷰 추출 및 자연어 처리에 사용되는 프로그래밍 언어
AWS CLI	Amazon 서비스 통합 관리 (AWS EKS)
CUDA	GPU의 가상 명령어셋을 사용할 수 있도록 만들어주는 소프트웨어 레이어
tensorflow	딥러닝 라이브러리 중 하나이며 Python을 활용하여 연산처리 작성
Amazon EKS	AWS 상에서 컨테이너화 된 애플리케이션 관리 자동화
Anaconda	과학 연구 및 머신러닝 분야에 적합한 Python 및 R 언어의 패키지/의존성 관리 및 배포를 편리하게 해주는 패키지 관리자
Kafka	쇼핑몰 추출 리뷰 유실 방지
Opensearch	쇼핑몰 리뷰 데이터, 감성 분석 결과 데이터 적재

Software	Reason for use
Terraform	클라우드 프로바이더에 IaC 배포 자동화
cuDNN	심층 신경망을 위한 GPU 가속 프리미티브 라이브러리
ChromeDriver	쇼핑몰 댓글 추출 시 렌더링 되는 웹 드라이버
Jupyter Notebook	탐색적 데이터 분석, 데이터 정리 및 변환, 데이터 시각화, 통계적 모델링, 머신 러닝, 딥러닝 등의 각종 데이터 사이언스 문서 생성 애플리케이션
Docker	크롤링 파드 이미지 생성
mecab	일본어와 한국어의 유사점으로 한글 분석에도 동작하는 것을 확인하고 개발한 한국어 형태소 분석기
Logstash	- Kafka에 저장된 데이터 수집 (Consumer) - 쇼핑몰 리뷰 데이터 정제 및 인덱싱
Kibana	리뷰 감성 분석 결과 시각화

서비스 플로우 -1



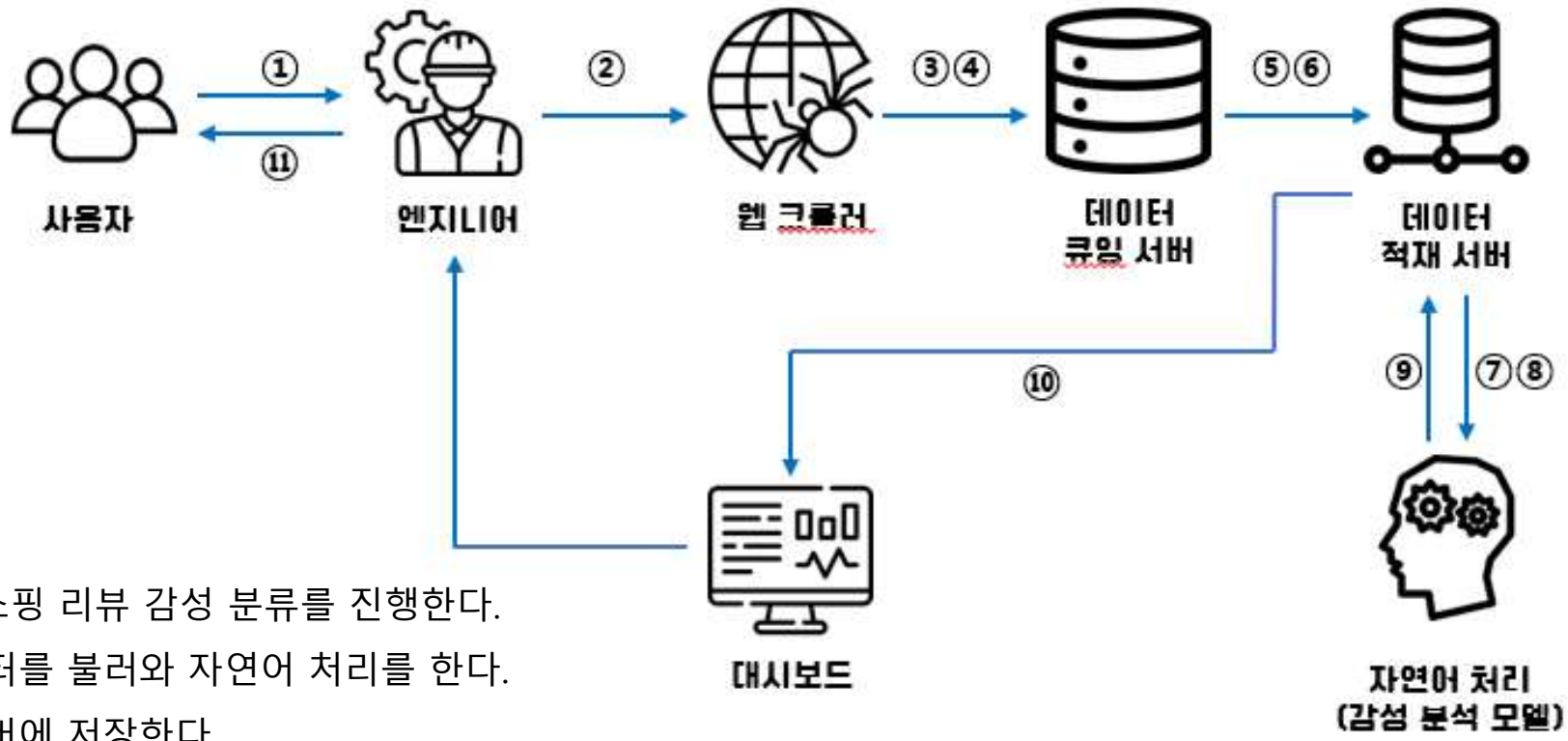
쇼핑몰 리뷰 분석 요청하기

1. 사용자가 온라인 쇼핑몰 리뷰 분석을 요청한다.
2. 엔지니어는 크롤러에게 사용자 쇼핑몰의 리뷰 추출을 요청한다.
3. 크롤러는 사용자의 쇼핑몰에 접근한여 리뷰, 별점, 작성일자를 추출한다.
4. 크롤러는 추출한 데이터를 카프카 브로커의 토픽에 저장한다.

쇼핑몰 리뷰 데이터 정제 및 저장하기

5. 토픽에 저장된 데이터를 불러와 특수문자를 제거하고 JSON 포맷으로 변환한다.
6. 정제된 데이터를 인덱싱하여 데이터 적재 서버(OpenSearch)에 저장한다

서비스 플로우 -2



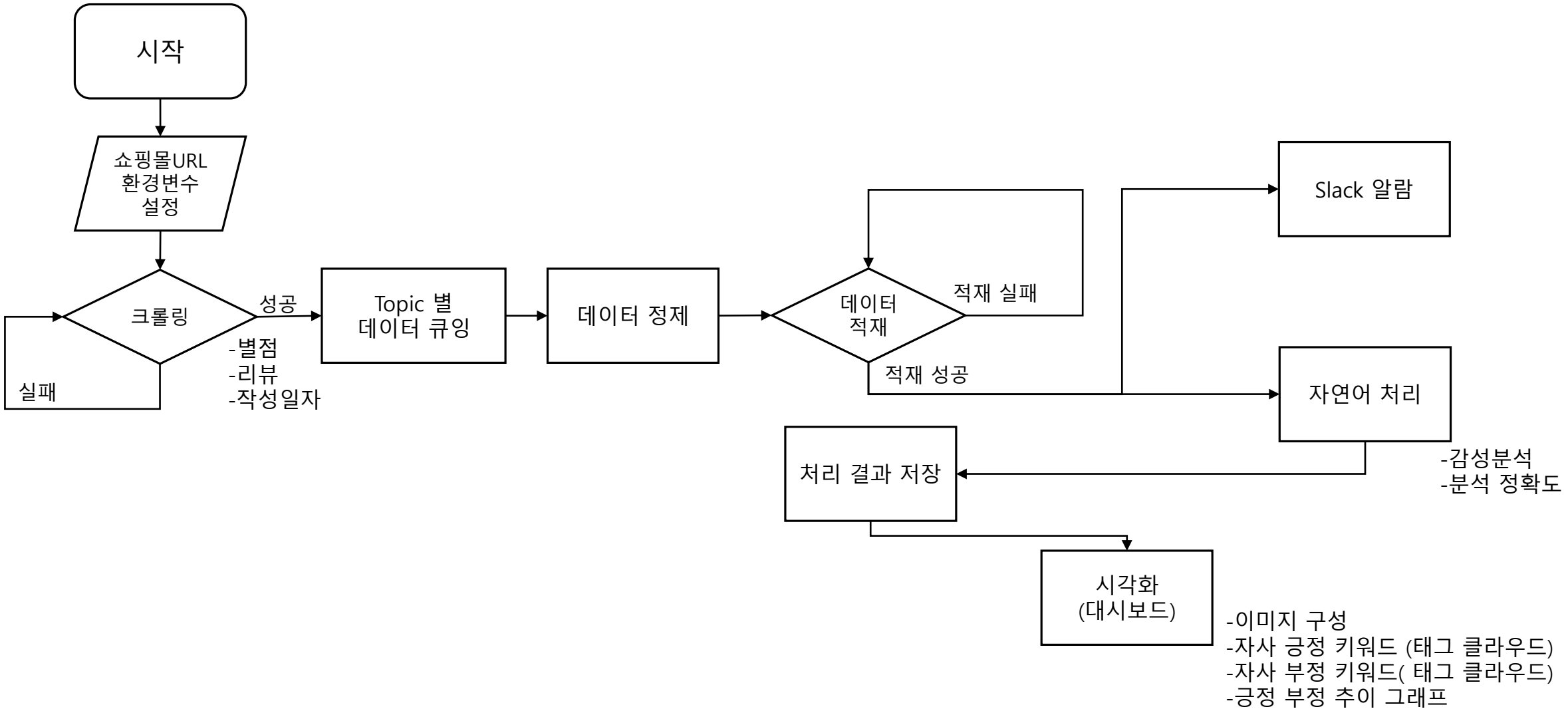
쇼핑몰 리뷰 NLP 처리

7. GRU를 사용하여 자사·타사·유사 쇼핑 리뷰 감성 분류를 진행한다.
8. 데이터 적재 서버에 저장 된 데이터를 불러와 자연어 처리를 한다.
9. 처리 결과를 다시 데이터 적재 서버에 저장한다.

분석 결과 시각화

10. 저장 된 데이터의 인덱스를 기반으로 대시보드 형태로 시각화 한다.
11. 사용자에게 해당 대시보드의 주소를 제공한다.

기능 플로우


시작
쇼핑몰URL
환경변수
설정
크롤링
성공
실패
-별점
-리뷰
-작성일자
Topic 별
데이터 큐잉
데이터 정제
데이터
적재
적재 실패
적재 성공
Slack 알람
자연어 처리
-감성분석
-분석 정확도
처리 결과 저장
시각화
(대시보드)
-이미지 구성
-자사 긍정 키워드 (태그 클라우드)
-자사 부정 키워드(태그 클라우드)
-긍정 부정 추이 그래프

데이터 플로우

★★★★★ 5
pr**** · 22.04.18. | 신고
선택1: (스트랩)핵빅사이즈 / 선택2: 베이지

머리 크신분들은 여기 무조건 추천합니다.

★★★★★ 5
pr**** · 22.04.18. | 신고
선택1: (스트랩)핵빅사이즈 / 선택2: 블랙

머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여


```
{
  "_index" : "smartstore.drstyle.review-220804",
  "_type" : "_doc",
  "_id" : "G167ZoIBNDnq2cJu7hzC",
  "_score" : 6.6732597,
  "_source" : {
    "timestamp" : "2022-08-04T11:42:48.000Z",
    "star" : 5,
    "topic" : "smartstore.drstyle.review",
    "comment" : "머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여",
    "date" : "22.04.18."
  }
},
{
  "_index" : "smartstore.drstyle.review-220804",
  "_type" : "_doc",
  "_id" : "kl68ZoIBNDnq2cJu5Bws",
  "_score" : 6.6732597,
  "_source" : {
    "timestamp" : "2022-08-04T11:43:11.000Z",
    "star" : 5,
    "topic" : "smartstore.drstyle.review",
    "comment" : "머리 크신분들은 여기 무조건 추천합니다.",
    "date" : "22.04.18."
  }
```


Crawler

Logstash


```
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "머리 크신분들은 여기 무조건 추천합니다.", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "너무 좋아요 담에 또 이용하께요", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "흐물거리지 않고 빳빳 하니 정말 좋아요 정사이즈입니다", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "흐물거리지 않고 빳빳 하니 정말 좋아요 정사이즈입니다", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "머리 크면 제약이 많은데 여기 제품은 너무 알잘딱이네여", "date": "22.04.18."}
{"topic": "smartstore.drstyle.review", "star": "5", "comment": "너무 좋아요 담에 또 이용하께요", "date": "22.04.18."}
```

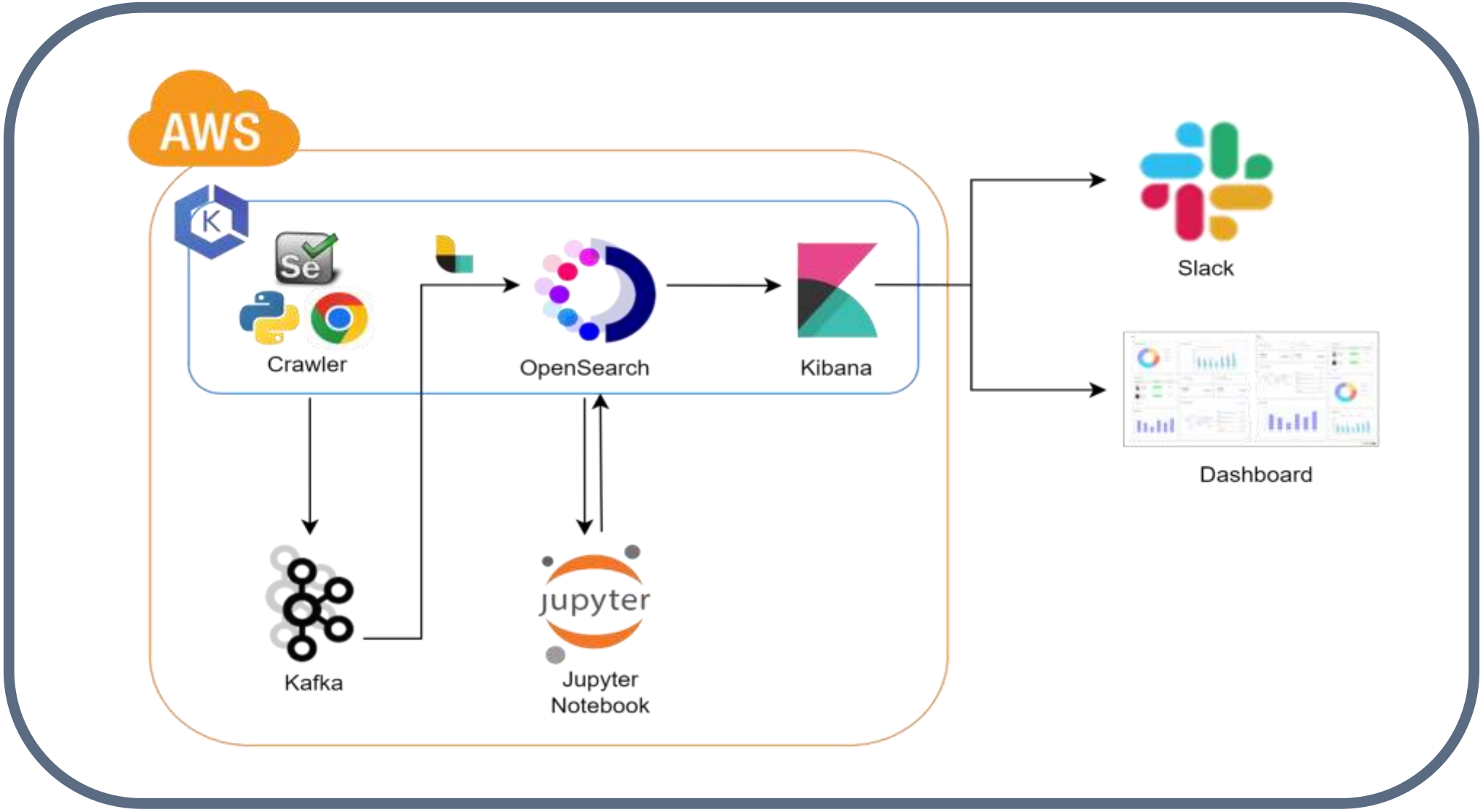
요구사항 정의서

구분	ID	요구사항명	기능	세부사항
기능 요구사항	SFR-001	리뷰 크롤링을 통한 쇼핑몰 데이터 추출	데이터 추출	소비자의 쇼핑몰 리뷰에서 댓글, 별점, 작성일자를 추출
				추출하고자 하는 URL에 따라 추출 진행
				추출이 완료된 후 성공/실패 여부와 추출에 걸린 시간을 kafka에 전송
	SFR-002	리뷰 데이터 적재 알림 전송	알림 전송	Kafka에 크롤링 한 데이터가 적재 완료될 경우 Slack을 통해 알림 전송
	SFR-003	온라인 쇼핑몰 리뷰을 분석하기 위한 NLP	상위 키워드 추출	빈도 수에 따라 상위 키워드 추출하여 시각화
			감성 분석	20만개의 리뷰와 별점 데이터를 크롤링한 후 test데이터와 train 데이터로 나눔
				train 데이터로 리뷰에 대한 긍정/부정 분석 모델링
			분석 정확도 계산	test 데이터로 모델 검증
			단어 간 유사도 계산	키워드를 지정 후 키워드와 유사한 단어를 추출
			월별 추이	분석하고자 하는 쇼핑몰 리뷰의 긍정/부정 추이를 간트 차트로 시각화
	SFR-004	시각화를 통한 대시보드 생성	대시보드 생성	분석 결과를 바탕으로 데이터 프레임 생성
				데이터 프레임에 맞게 대시보드에서 시각화
데이터 요구사항	DR-001	크롤링 데이터 유실 방지를 위한 큐잉 서비스	데이터 유실 방지	크롤링 한 데이터를 받아 오기 위해 상품 별 토픽 생성
				토픽에 쌓인 데이터를 차례로 전달
	DR-002	온라인 쇼핑몰 리뷰 데이터 정제 및 형식 변환	수신 데이터 특수 문자 제거, 인덱싱 및 형식 변환	원하는 상품에 해당하는 토픽의 데이터 수신
				수신 받은 데이터의 특수 문자 제거 및 인덱싱
				정제 된 데이터를 JSON 형태로 변환
	DR-003	온라인 쇼핑몰 리뷰 데이터 NLP를 위한 데이터 적재	크롤링 후 정제된 데이터 저장	자연어 처리를 하기 위한 데이터를 받아와 저장
	DR-004	온라인 쇼핑몰 리뷰 데이터를 NLP한 결과 데이터 적재	감성 분석 결과 데이터 저장	분석 결과를 데이터 프레임에 맞추어 저장
성능 요구사항	PER-001	질의 · 응답 시간	질의응답 시간 및 오류메시지 응답시간	시스템의 모든 기능은 화면에 출력할 때 브라우저의 영향을 받지 않고 보여주지 않아야 함
				구축 시스템의 사용자 서비스 페이지는 평균 5초 이내에 처리되어야 함
				사용자 요청 작업 관련 평균 시간 초과 응답 시 성능향상 방안을 강구하여야 함

요구사항 정의서

구분	ID	요구사항명	기능	세부사항
시스템 장비구성 요구사항	ECR-001	시스템 장비구성 요구사항	시스템 장비구성 요구사항	Crawler: t3.medium * 2 EKS 내 2개의 노드에 지정
				ElasticSearch-master: t3.large 1~3 ElasticSearch-data: t3.large 1~3 ElasticSearch-client: t3.large 1~3 Kibana: t3.medium 1~3 Logstash: t3.large 1~3 EKS 내 1~3개의 노드로 가용성 확보
				Jupyter Notebook: g4dn.xlarge 1 Kafka Cluster: t3.medium 3 EC2 내 1 혹은 3개의 노드로 구성
테스트 요구사항	TER-001	테스트 요구사항	단위 테스트	프로그램 개발 일정 및 테스트 일정에 따라 개발된 프로그램에 대해 단위 테스트 실시 방안을 수립
			통합 테스트	통합 테스트는 최소 1회 이상 실시해야 하며, 테스트 일정에 따라 구체적인 실시 방안을 수립하여 수행 후 결과 보고
			시험 운영	계획된 일정에 따라 시험운영 방안을 수립하여 제시
품질 요구사항	QR-001	품질 보증 활동	기능 구현의 정확성 향상 방안 수립 및 준수	개발 시스템은 제공되기로 한 기능 요구사항을 모두 제공해야하며, 초기 협의한 요구사항에서 변경이 필요한 경우 주관기관 담당자와 협의하여 요구사항을 변경 협의 및 과업 변경 가능
프로젝트 관리 요구사항	PMR-001	프로젝트 일정관리	프로젝트 일정관리에 관한 사항	일정계획 제시 - 개발의 완성도를 높이기 위해서 프로젝트 착수에서 종료까지 체계적으로 프로젝트를 관리해야 함
				단계별 일정관리 - 각 업무단위별 단계별 일정 계획을 수립하여 제시하고, 일정 지연이 발생하는 경우 별도 계획수립 등 만회 계획을 작성
	PMR-002	투입인력 및 역할 분담	투입인력에 관한 사항	투입인력 및 역할 분담 - 모든 투입인력은 프로젝트 인력으로 구성하여야 하며, 업무분담(역할)을 관리 및 작성
	PMR-003	형상 관리	형상 관리 방안 수립 및 준수	형상 관리 어플리케이션 개발부터 소멸까지의 소스 코드를 포함한 각종 산출물은 중앙에서 통합적으로 변경관리가 가능하도록 관리

프로세스 아키텍처


AWS
Crawler
OpenSearch
Kibana
Slack
Dashboard
Kafka
jupyter
Jupyter
Notebook

인프라 구현

인프라 구현 – EKS

프로젝트의 소프트웨어 중 파이썬 크롤러, Logstash, Opensearch, Kibana는 EKS를 통해 배포

```
├── fixture.tc1.tfvars
├── main.tf
├── modules
│   ├──addonalb-ingress
│   ├──app-mesh
│   ├──cluster-autoscaler
│   ├──container-insights
│   ├──ecr
│   ├──iam-role-for-service-account
│   ├──karpenter
│   ├──lb-contoller
│   ├──metrics-server
│   ├──node-termination-handler
│   ├──prometheus
├── outputs.tf
└── variables.tf
```

fixture.tc1.tfvars

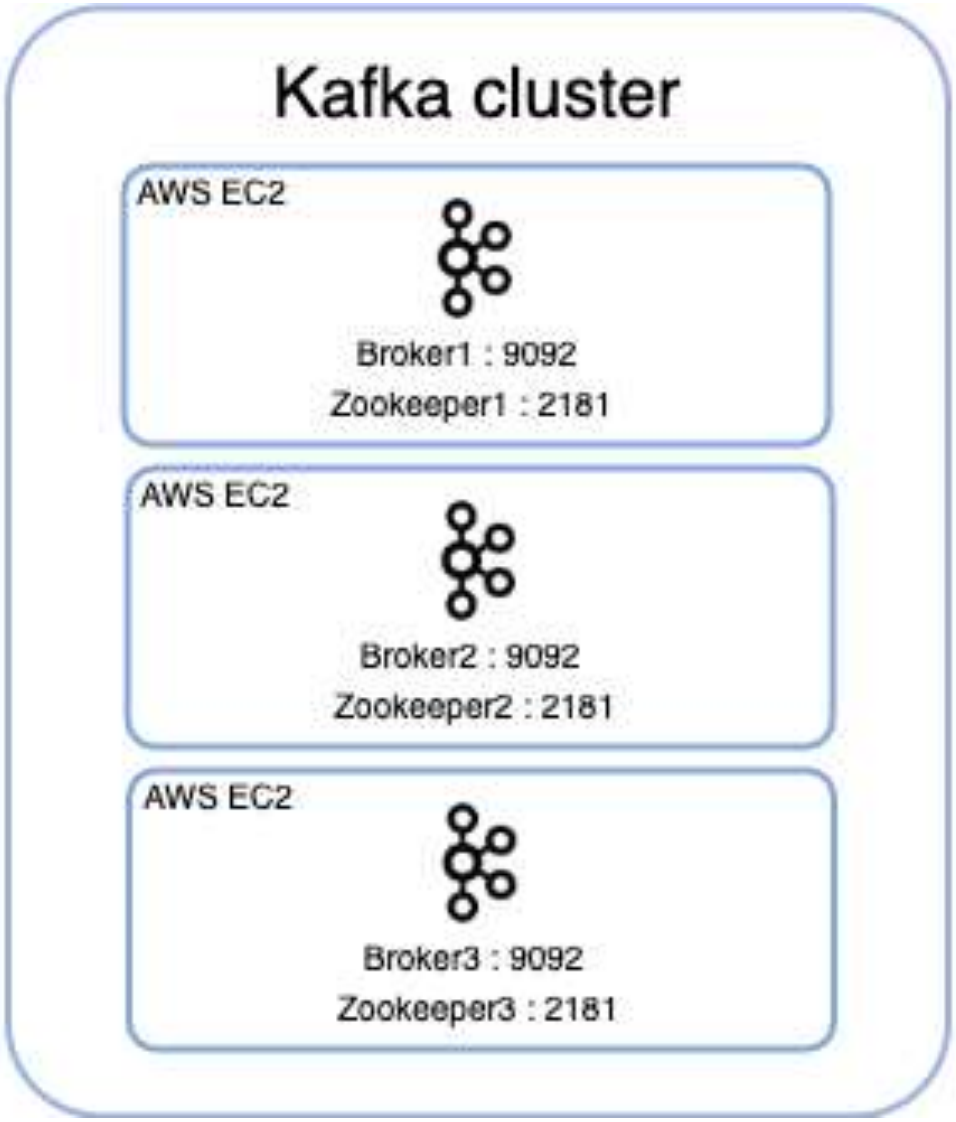
```
aws_region      = "ap-northeast-2"
azs             = ["ap-northeast-2a", "ap-northeast-2b", "ap-northeast-2c"]
cidr       = "10.1.0.0/16"
enable_igw = true
enable_ngw = true
single_ngw = true
name            = "eks-autoscaling-tc1"
tags = {
  env  = "dev"
  test = "tc1"
}
kubernetes_version  = "1.21"
enable_ssm          = true
managed_node_groups = [
  {
    name          = "crawler"
    min_size      = 1
    max_size      = 6
    desired_size  = 1
    instance_type = "t3.medium"
  },
  {
    name          = "ElasticSearch-master"
    min_size      = 1
    max_size      = 3
    desired_size  = 1
    instance_type = "t3.large"
  },
```

인프라 구현 - Python Crawler

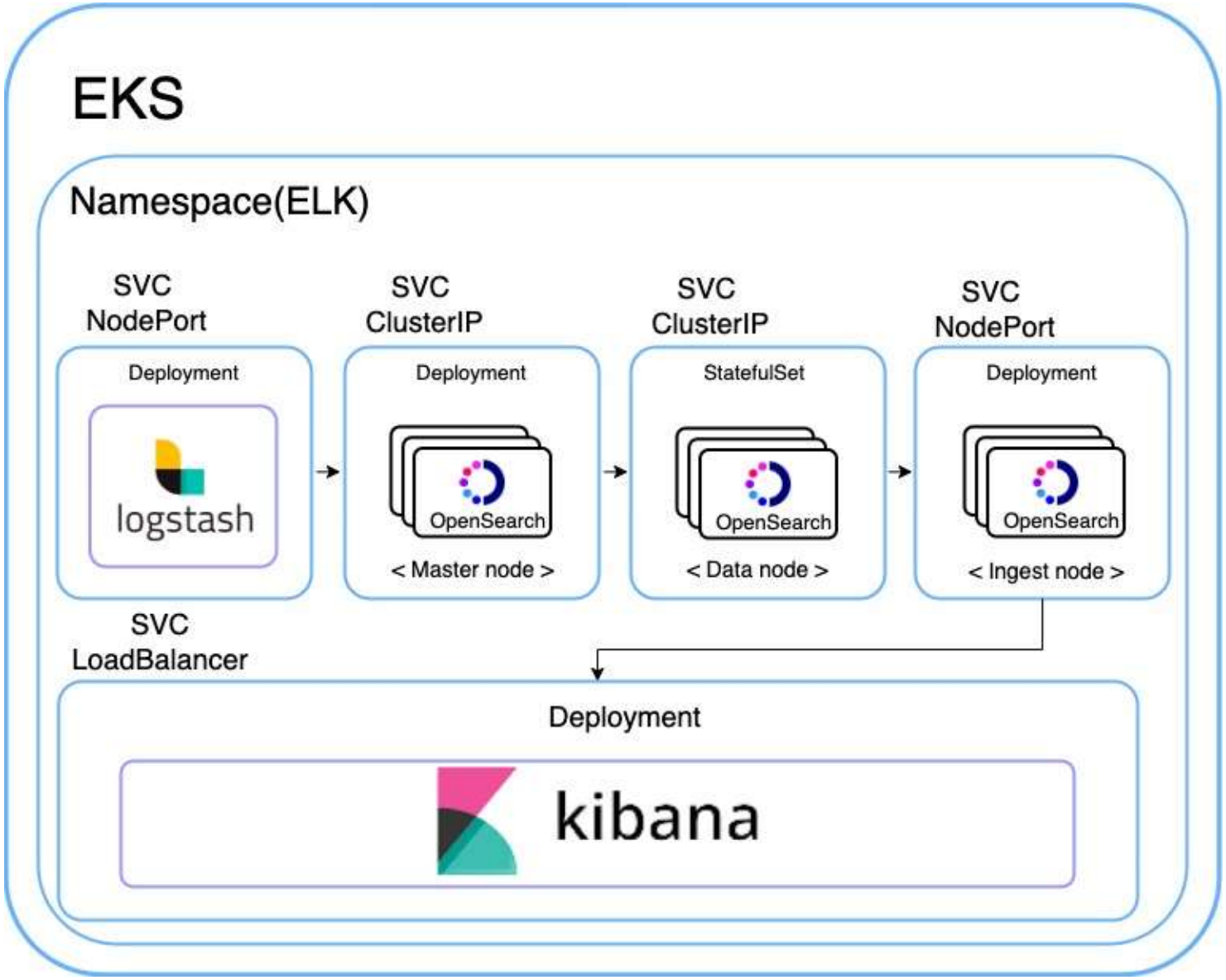
Crawler파드 생성을 위한 YAML파일

```yaml
apiVersion: batch/v1
kind: CronJob
metadata:
  namespace: crawler
  name: crawler-1
  labels:
    app: crawler
spec:
  schedule: "00 00 * * *"
  jobTemplate:
    spec:
      template:
        metadata:
          labels:
            app: crawler
          annotations:
            "cluster-autoscaler.kubernetes.io/safe-to-evict": "false"
        spec:
          nodeSelector:
            Name: Crawler
          containers:
          - name: crawler-1
            image: ddung1203/kafkacrawler:10
            resources:
              requests:
                memory: "3000Mi"
              limits:
                memory: "3000Mi"
                cpu: "1500m"
            env:
            - name: url
              value: 'https://smartstore.naver.com/goodnara/products/37118c9'
            - name: topic
              value: smartstore.goodnara.review
            - name: server
              value: "3.38.10.106:9092,3.34.18.190:9092,13.209.146.71:9092"
            command: ["/bin/sh", "-c"]
            args: ["cd /Datapipeline_Project/crawler; ./kafka_producer.py"]
          restartPolicy: Never
```

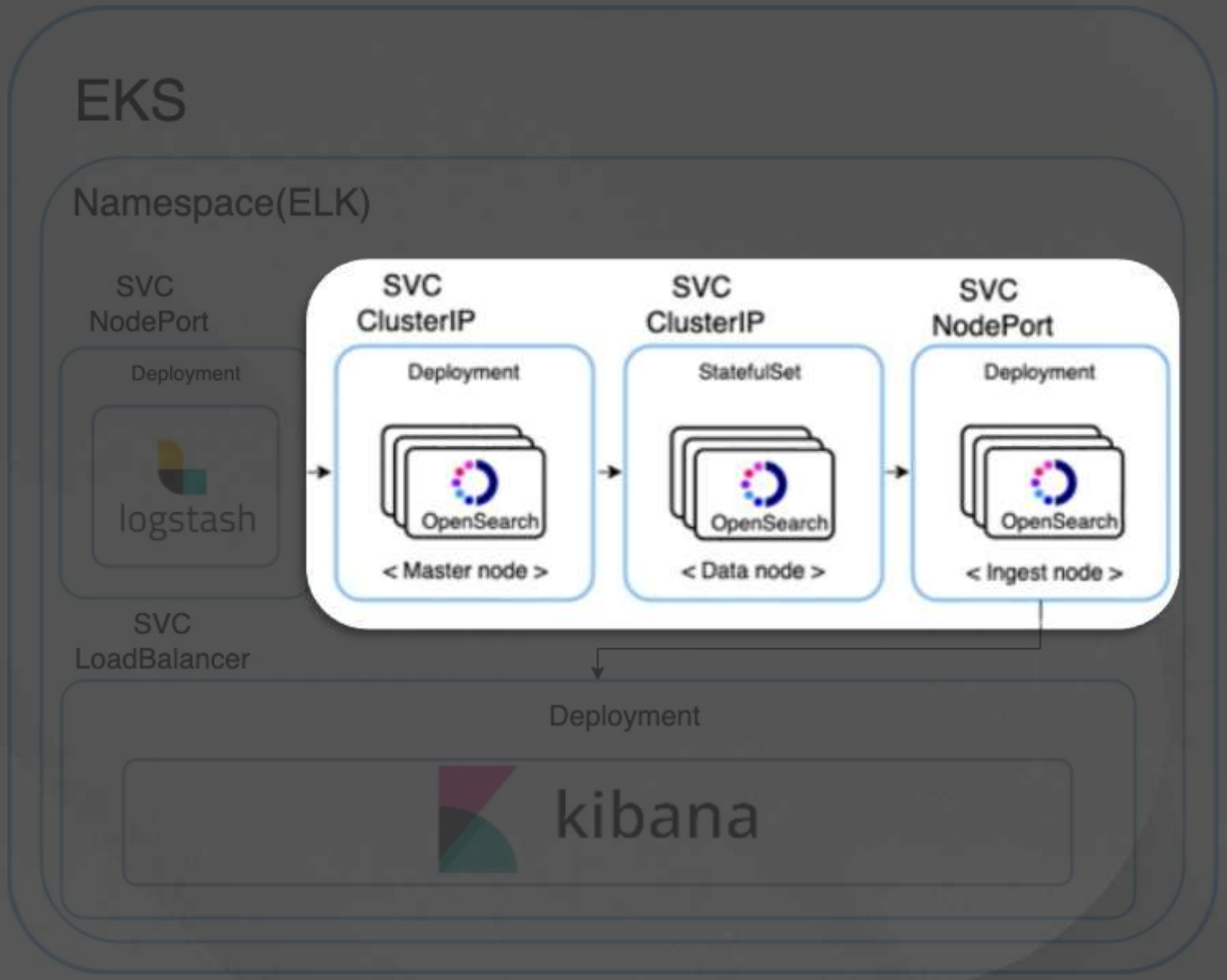
인프라 구현 - Kafka cluster



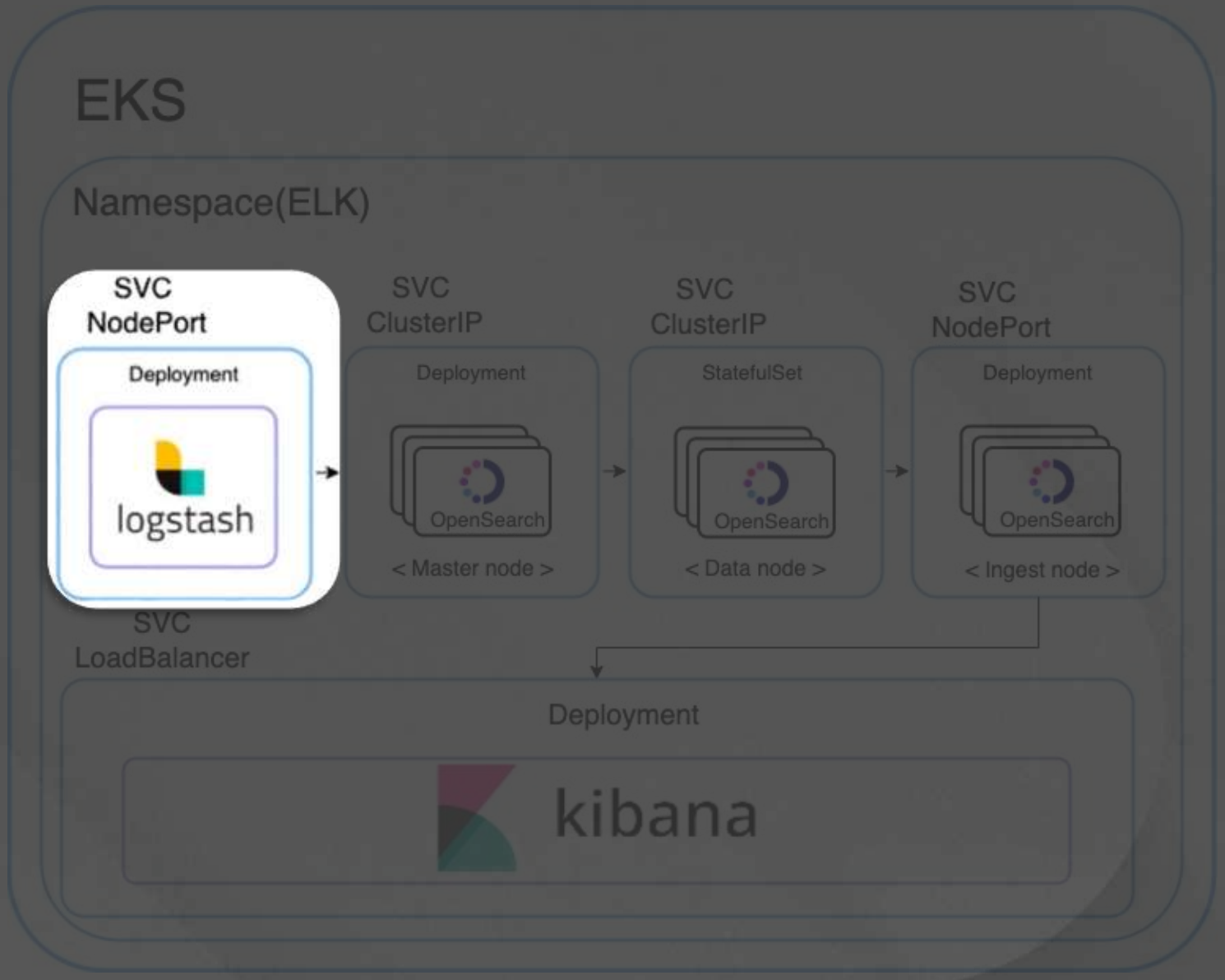
인프라 구현 - ELK


EKS
Namespace(ELK)
SVC
NodePort
SVC
ClusterIP
SVC
ClusterIP
SVC
NodePort
Deployment
logstash
Deployment
OpenSearch
< Master node >
StatefulSet
OpenSearch
< Data node >
Deployment
OpenSearch
< Ingest node >
SVC
LoadBalancer
Deployment
kibana

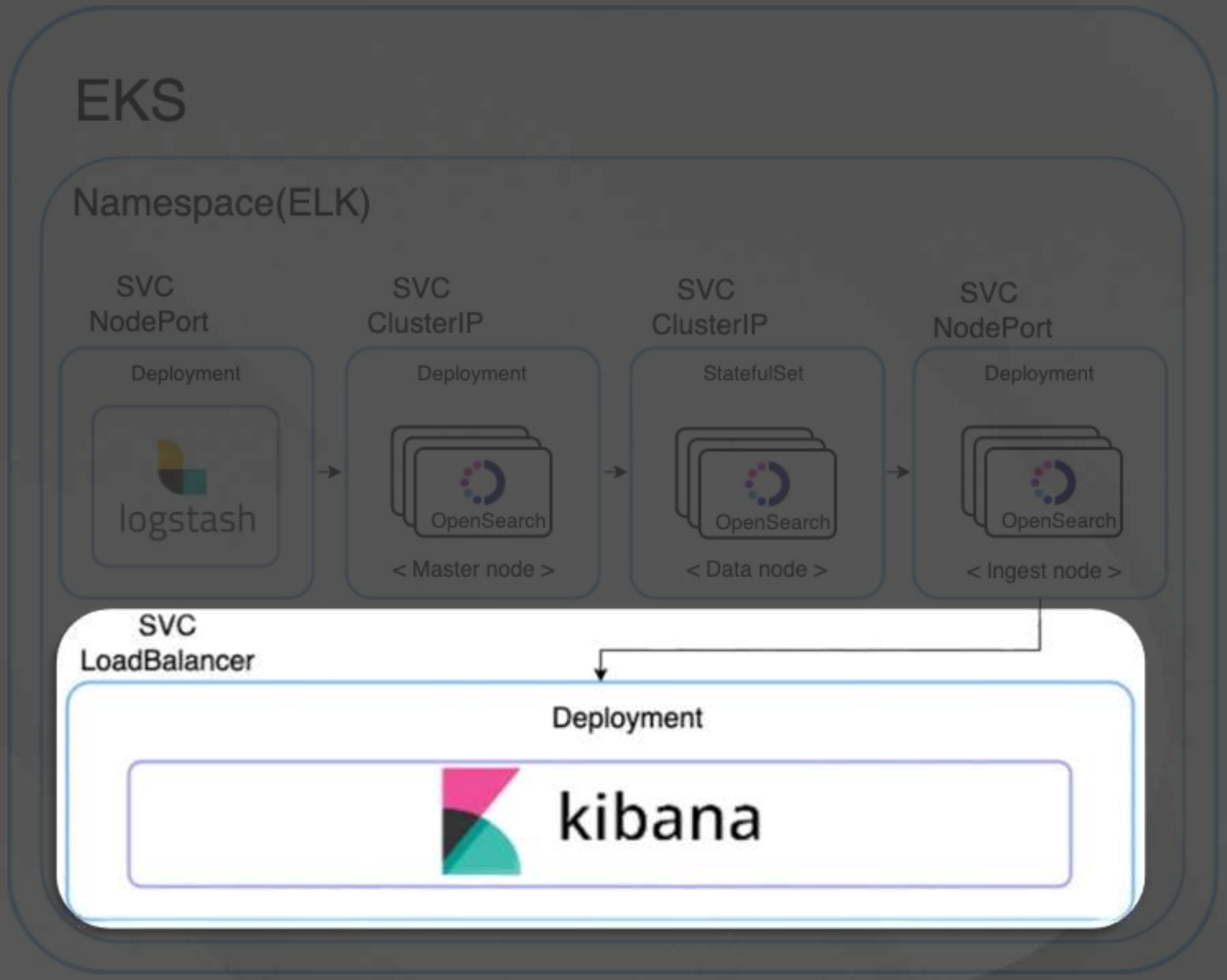
인프라 구현 - ELK


EKS
Namespace(ELK)
SVC
NodePort
Deployment
logstash
SVC
LoadBalancer
SVC
ClusterIP
Deployment
OpenSearch
< Master node >
SVC
ClusterIP
StatefulSet
OpenSearch
< Data node >
SVC
NodePort
Deployment
OpenSearch
< Ingest node >
Deployment
kibana

인프라 구현 - ELK


EKS
Namespace(ELK)
SVC
NodePort
Deployment
logstash
SVC
LoadBalancer
SVC
ClusterIP
Deployment
OpenSearch
< Master node >
SVC
ClusterIP
StatefulSet
OpenSearch
< Data node >
SVC
NodePort
Deployment
OpenSearch
< Ingest node >
Deployment
kibana

인프라 구현 - ELK


EKS
Namespace(ELK)
SVC
NodePort
SVC
ClusterIP
SVC
ClusterIP
SVC
NodePort
Deployment
Deployment
StatefulSet
Deployment
logstash
OpenSearch
OpenSearch
OpenSearch
< Master node >
< Data node >
< Ingest node >
SVC
LoadBalancer
Deployment
kibana

기능 구현

기능 구현 – 데이터 크롤링

```python
# kafka producer
producer = KafkaProducer(acks=1, compression_type='gzip',bootstrap_servers=[server],
                         value_serializer=lambda x: json.dumps(x, ensure_ascii=False).encode('utf-8'))
...

for cmt in range(1,21):
    #comment
    time.sleep(1)
    list_comnt = driver.find_element(By.XPATH, '경로'.format(cmt))
    if list_comnt.text == '한달사용기':
        list_comnt = driver.find_element(By.XPATH, '경로[2]'.format(cmt))
    if list_comnt.text == '재구매':
        list_comnt = driver.find_element(By.XPATH, '경로[3]'.format(cmt))
    elif list_comnt.text == '재구매':
        list_comnt = driver.find_element(By.XPATH, '경로[2]'.format(cmt))
    comment.append(list_comnt.text)

    #star
    list_star = driver.find_element(By.XPATH, '경로'.format(cmt))
    star.append(list_star.text)

    #date
    list_date = driver.find_element(By.XPATH, '경로'.format(cmt))
    date.append(list_date.text)
...

tmp={'star':star.pop(), 'comment':comment.pop(), 'date':date.pop()}
producer.send(topic, value=tmp)

producer.flush()
```

기능 구현 – 데이터 정제

로그 데이터의 파이프 라인 설정 파일

```
logstash.conf|
>> input {
      kafka {
        bootstrap_servers => ""    # 브로커 주소 10.0.0.92
          topics => ["smartstore.goodnara.review",
                     "smartstore.drstyle.review",
                     "smartstore.thecheaper.review",
                     "smartstore.180store.review",
                     "smartstore.cloony.review",
                     "smartstore.theshopsw.review"]
          consumer_threads => 3
          # 컨슈머 쓰레드(파티션 갯수와 동일)
          isolation_level => "read_committed"
          # 트랜잭션 격리수준(폴링 메세지는 커밋된 트랜잭션 메세지만 반환한다.)
          value_deserializer_class => "org.apache.kafka.common.serialization.StringDeserializer"
          # 레코드값 역직렬화에 사용되는 JAVA Class(String)
          auto_offset_reset => "earliest"
          # 처음 브로커에 진입했을때, 데이터를 가지고오는 시작점을 지정한다.
          #earliest는 초기에 consumer group이 설정되어 있지 않은 경우에만 적용되므로,
          #추후 컨슈머를 재시작 하더라도 데이터 중복 이슈를 해결할 수 있다.
          group_id => "smartstore" # 컨슈머 그룹이름을 지정한다.
          }
      }
```

기능 구현 – 데이터 정제

filter –timestamp 설정

```
filter {
      # ------------------------ UTC(default) Timestamp -> KST Timestamp로 변환하기 ------------------
        mutate {
          add_field => {
            "timestamp" => ""   # timestamp 필드 생성(새로 생성된 필드의 기본 데이터 타입은 String이다.)
          }
        }
        # ruby 코드로 "@timestamp" 필드의 UTC 기준 현재 시간에 9시간을 더한 값을 timestamp 필드에 저장한다.
        ruby {
          code => "event.set('timestamp', event.get('@timestamp').time.localtime('+09:00').strftime('%Y-%m-%d %H:%M:%S'))"
        }
        # timestamp 필드의 데이터는 String이므로 날짜 형식으로 지정함(ISO8601)
        # ISO8601 = 2019-01-26T17:00:00.000Z
        date {
          match => ["timestamp", "ISO8601", "YYYY-MM-dd HH:mm:ss"]
          target => "timestamp"   # date 필터가 적용될 필드 지정
        }
        # timestamp를 파싱하여 yy mm dd만 추출해 yymmdd 필드로 저장한다.
        grok {
          match => {
            "timestamp" => "\d\d%{INT:yy}-%{MONTHNUM:mm}-%{MONTHDAY:dd}%{GREEDYDATA}"
          }
          # yymmdd를 메타필드로 저장한다.
          add_field => {
            "[@metadata][yymmdd]" => "%{yy}%{mm}%{dd}"
          }
        }
```

기능 구현 - 데이터 정제

filter – message 필드 정제

```
# ------------------------- message 필드 정제 ----------------------
        # 정규표현식으로 특수문자 1차 제거한다.
        mutate {
          gsub => ["message", "[\"/{}]", ""]
        }
        # comma를 기준으로 메세지 필드를 나눈 후, colon을 기준으로 Key, value 형식으로 필드를 생성한다.
        kv {
          field_split => ","
          value_split => ":"
        }
        # 사용하지 않을 필드들을 제거하고 필드의 이름을 재설정한다.
        mutate {
          remove_field => [ "port","@version","host","message","@timestamp", "yy", "mm", "dd" ]
          rename => {" comment" => "comment"}
          rename => {" date" => "date"}
          rename => { " star" => "star" }
        }
        # star 필드를 String에서 Integer 타입으로 변환한다.
        mutate {
          convert => {
            "star" => "integer"
          }
        }
    }
```

기능 구현 – 데이터 정제

output

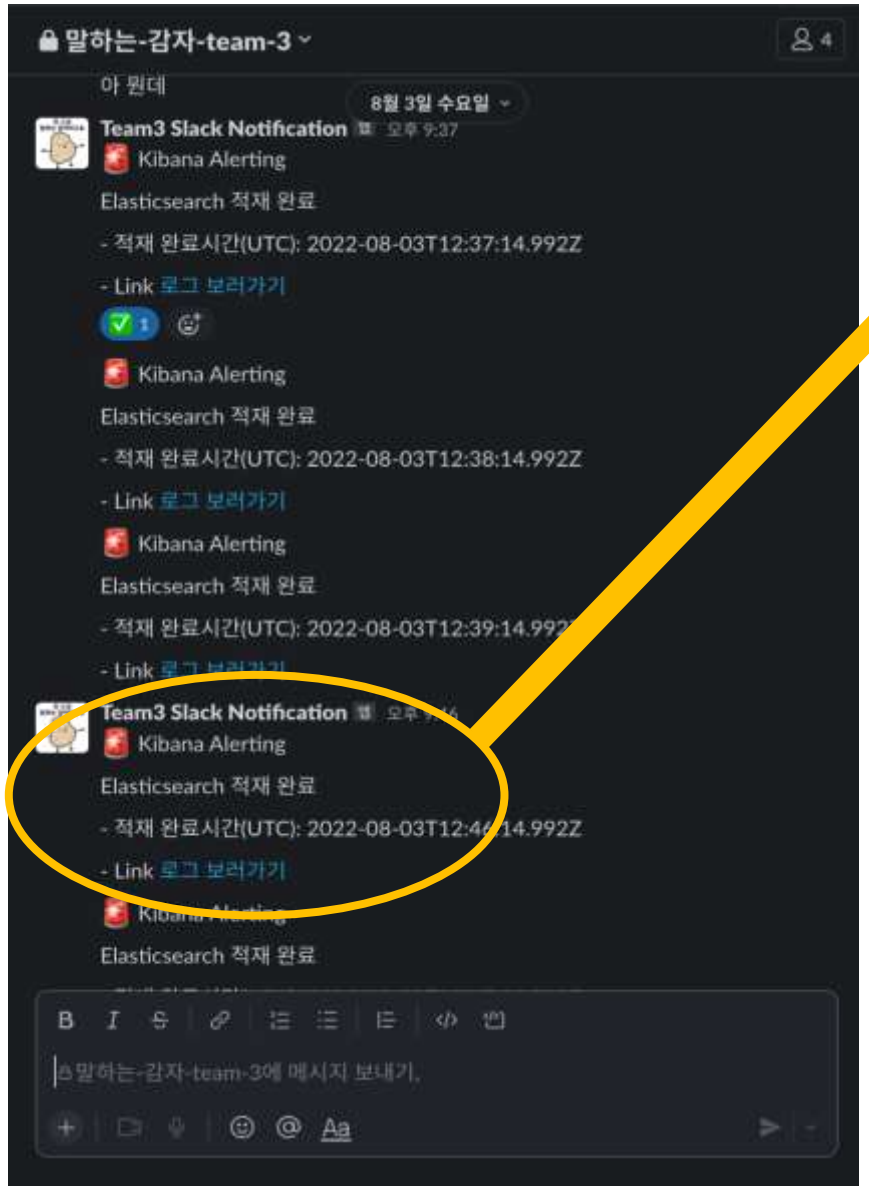
```
output {
      stdout { codec => rubydebug }
      # ---------- 쇼핑몰 토픽에 따라 인덱스를 나누어 저장한다. ----------
      # 스마트 스토어 모자 Topic
      if [topic] =~ "smartstore.goodnara.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.goodnara.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
      else if [topic] =~ "smartstore.drstyle.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.drstyle.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
      else if [topic] =~ "smartstore.thecheaper.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.thecheaper.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
      # 스마트 스토어 티셔츠 Topic
      else if [topic] =~ "smartstore.180store.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.180store.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
      else if [topic] =~ "smartstore.cloony.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.cloony.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
      else if [topic] =~ "smartstore.theshopsw.review" {
        elasticsearch {
          hosts => "http://elasticsearch-client-http.elk.svc.cluster.local:9200"
          index => "smartstore.theshopsw.review-%{[@metadata][yymmdd]}"
          codec => "json"
          timeout => 120
        }
      } # if end
    } # output end
```

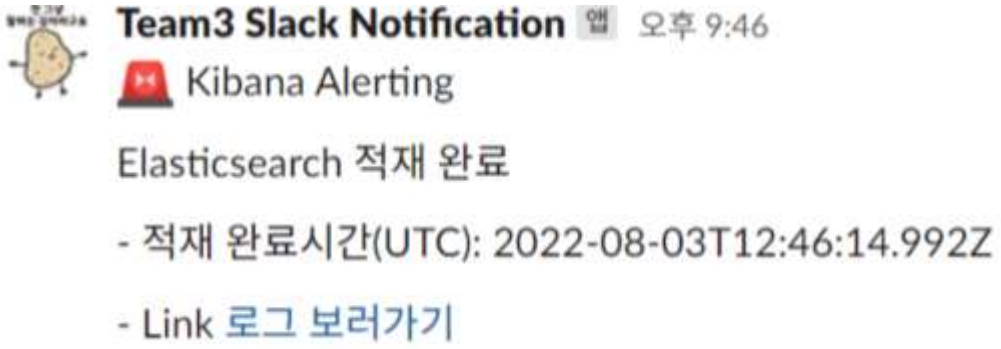
정제 된 데이터 예시

쇼핑몰 리뷰 데이터가 성공적으로 적재된 것을 알 수 있다.

```
{
    "timestamp" => 2022-08-03T09:51:08.000Z,
         "star" => 4,
        "topic" => "smartstore.180store.review",
      "comment" => "후드티안에 입을려고샀는대 길이가 후드티아래로\\n딱 적당하게 내려오네요 좋아요",
         "date" => "21.01.31."
}
{
    "timestamp" => 2022-08-03T09:51:08.000Z,
         "star" => 4,
        "topic" => "smartstore.180store.review",
      "comment" => "후드티안에 입을려고샀는대 길이가 후드티아래로\\n딱 적당하게 내려오네요 좋아요",
         "date" => "21.01.31."
}
{
    "timestamp" => 2022-08-03T09:51:08.000Z,
         "star" => 5,
        "topic" => "smartstore.180store.review",
      "comment" => "빠른 배송 잘 받았습니다.",
         "date" => "21.02.07."
}
{
    "timestamp" => 2022-08-03T09:51:08.000Z,
         "star" => 5,
        "topic" => "smartstore.180store.review",
      "comment" => "한치수 크게사면 오바핏으로 알맞게 떨아지네요",
         "date" => "21.02.04."
}
```

기능 구현 - Slack Alert





기능 구현 - Slack Alert

Define extraction query

```
{
    "size": 0,
    "query": {
        "bool": {
            "filter": [
                {
                    "range": {
                        "timestamp": {
                            "from": "{{period_end}}||+9h-2m",
                            "to": "{{period_end}}||+9h",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                },
                {
                    "match": {
                        " status": {
                            "query": "Success",
                            "operator": "OR",
                            "prefix_length": 0,
                            "max_expansions": 50,
                            "fuzzy_transpositions": true,
                            "lenient": false,
                            "zero_terms_query": "NONE",
                            "auto_generate_synonyms_phrase_quer
                            "boost": 1
                        }
```

Extraction query response

```
{
    "_shards": {
        "total": 6,
        "failed": 0,
        "successful": 6,
        "skipped": 0
    },
    "hits": {
        "hits": [],
        "total": {
            "value": 0,
            "relation": "eq"
        },
        "max_score": null
    },
    "took": 9,
    "timed_out": false
}
```

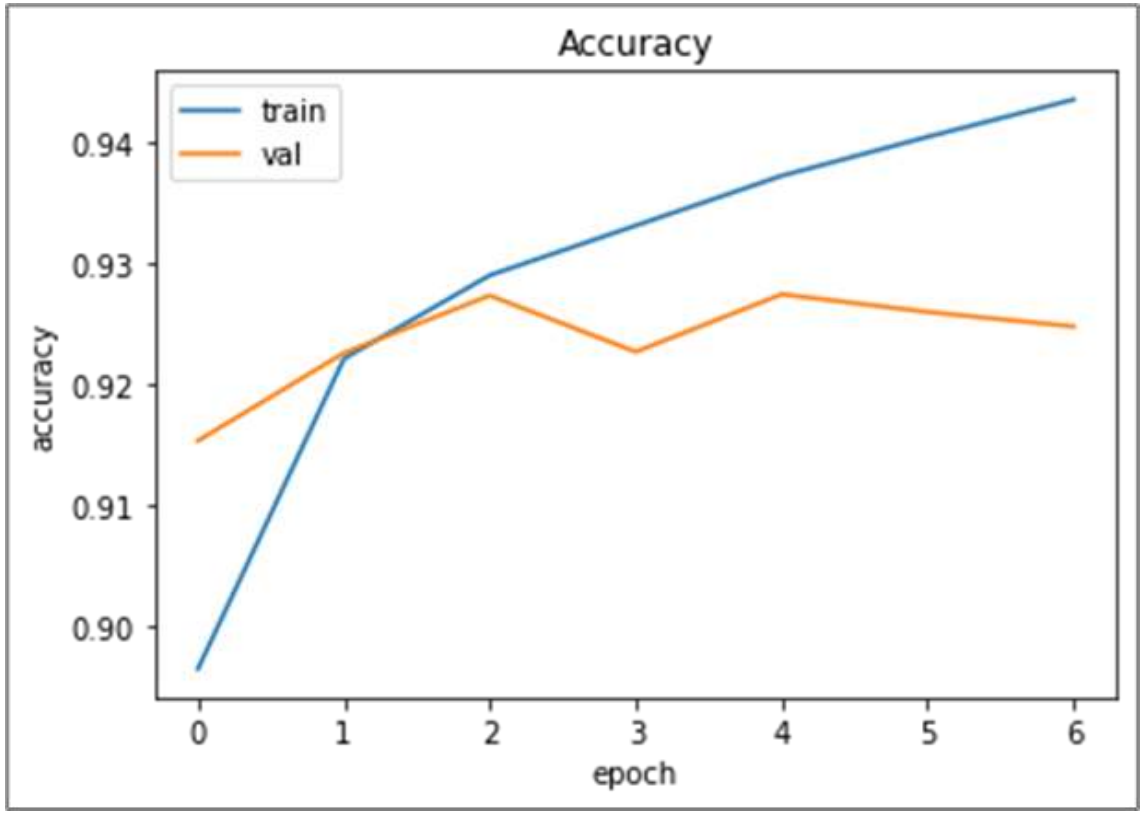
기능 구현 – 감성분석 (NLP)

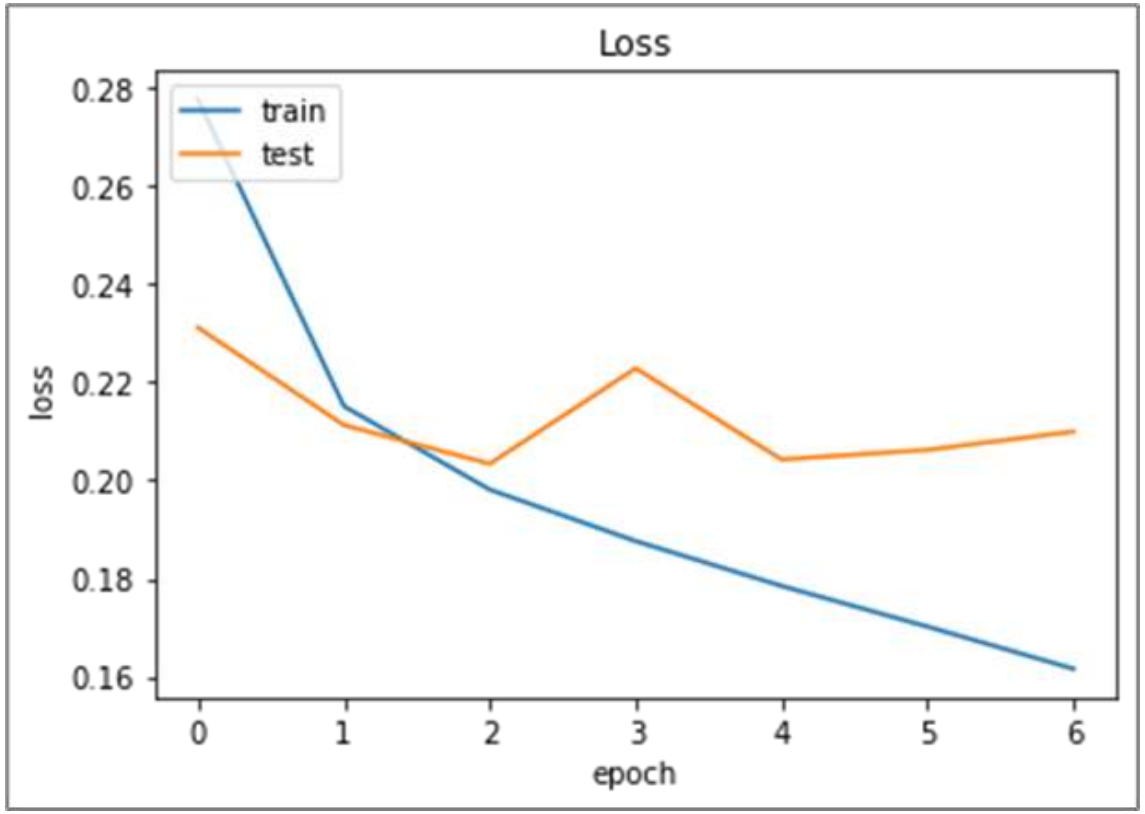
네이버 쇼핑 리뷰 감성 분류

```
from tensorflow.keras.layers import Embedding, Dense, GRU
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import load_model
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

    embedding_dim = 100
    hidden_units = 128

    model = Sequential()
    model.add(Embedding(vocab_size, embedding_dim))
    model.add(GRU(hidden_units))
    model.add(Dense(1, activation='sigmoid'))

    es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=4)
    mc = ModelCheckpoint('best_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)

    model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
    history = model.fit(X_train, y_train, epochs=15, callbacks=[es, mc], batch_size=64, validation_split=0.2)
```

기능 구현 – 감성분석 (NLP)

네이버 쇼핑 리뷰 감성 분류





기능 구현 – 감성분석 (NLP)

```python
# 불러온 데이터 전처리
# 중복 제거
data.drop_duplicates(subset = ['comment'], inplace=True, ignore_index=True)
# 정규 표현식 수행
data['comment'] = data['comment'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]","")
# 공백은 Null 값으로 변경
data['comment'].replace('', np.nan, inplace=True)
# Null 값 제거
data = data.dropna(how='any')

stopwords = ['도', '는', '다', '의', '가', '이', '은', '한', '에', '하', '고', '을',
             '를', '인', '듯', '과', '와', '네', '들', '듯', '지', '임', '게']

data=data.reset_index(drop=False)
data=data.loc[:, ['comment', 'date', 'star', 'topic']].dropna()
…
for i in range(len(data)):
    # 한국어 맞춤법 검사
    sent = data['comment'][i]
    spelled_sent = spell_checker.check(sent)
    hanspell_sent = spelled_sent.checked
    data['comment'][i] = hanspell_sent
…
for i in range(len(data)):
    # 감성분석
    sp = sentiment_predict(data['comment'][i])
    if sp == 1:
        data['new_label'][i] = '긍정'
    elif sp == 0:
        data['new_label'][i] = '부정'
    # 댓글 토큰화
    data['tokenized_comment'][i] = mecab.nouns(data['comment'][i])
```

기능 구현 - 분석 데이터 인덱스

```
## 분석결과 인덱스 mapping 설정
PUT analyzed_data
{
  "settings": {
    "number_of_shards": 5,
    "number_of_replicas": 3
  },
  "mappings": {
    "properties": {
      "Name" : {
        "type": "keyword"
      },
      "Star" : {
        "type": "long"
      },
      "Date" : {
        "type" : "date"
      },
      "Word" : {
        "type": "keyword"
      },
      "Sentiment" : {
        "type" : "keyword"
      },
      "Adjustment-sentiment" : {
        "type" : "keyword"
      }
    }
  }
}
```

감자 모자

리뷰 갯수

38,871
analyzed_data - Count

기존 리뷰

88.778%
긍정 - Count

10.494%
부정 - Count

평균 평점

4.592
Average Star

분석된 리뷰

82.613%
긍정 - Count

16.659%
부정 - Count

긍정 리뷰

Word: Descending - Count

부정 리뷰

Word: Descending - Count

전체 리뷰

Word: Descending - Count

자사 키워드

탁사 키워드

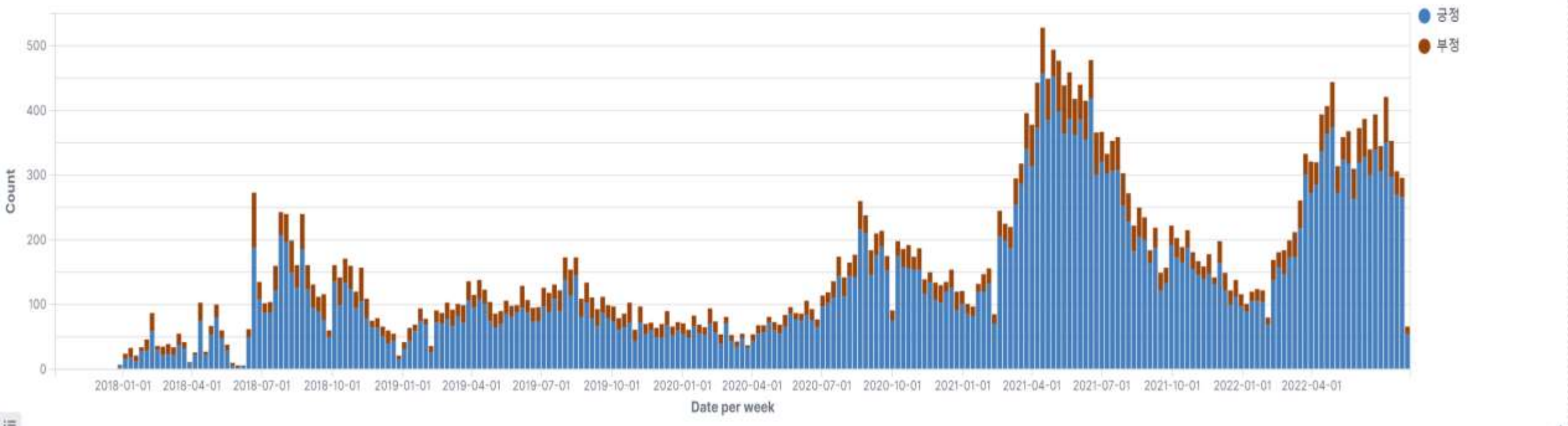
자사 키워드

가격	긍정	4,208
가격	부정	527
배송	긍정	3,788
배송	부정	216
사이즈	긍정	2,997
사이즈	부정	903
만족	긍정	3,021
만족	부정	75
구매	긍정	2,734
구매	부정	242

타사 키워드

가격	긍정	1,186
가격	부정	102
배송	긍정	1,023
배송	부정	32
만족	긍정	769
만족	부정	13
사이즈	긍정	613
사이즈	부정	130
구매	긍정	636
구매	부정	47

일자별 추이



프로젝트 결론

3.1 결과물 활용 방안

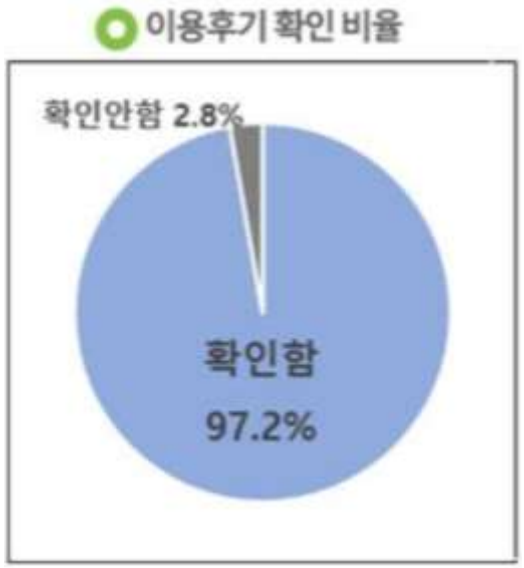
3.2 문제점 발생 및 해결

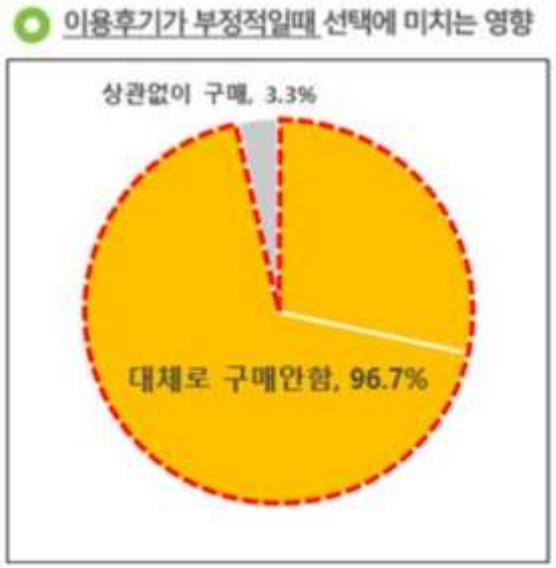
3.3 향후 개선점

3.4 Q&A

결과물 활용 방안

[표=한국소비자연맹]





조사결과 소비자의 대부분에 해당하는 97.2%가 구매 전 이용후기를 확인

특히 부정적 이용후기가 소비자의 구매결정에 더 많은 영향을 미치는 것으로 조사됨

문제점 발생 및 해결

- 크롤링 코드 작동 오류

- OOMKilled 오류로 인한 크롤러 파드 강제 종료

- 크롤링 진행 중 파이썬 크롤러 파드 재부팅

- Replication Factor 에러

- Zookeeper 클러스터 실행 오류

- Logstash Timestamp UTC 설정

향후 개선점

- **EKS에서 Kafka 배포**
- **Jupyter Notebook 애플리케이션화**
- **데이터 전송 단의 로그 분석 대시보드 생성**
- **효율적인 협업 툴 사용**

Thank you