

DANIM

TRAVEL DIARY APP

팀장 송지율

팀원 기남석. 김지나.
이영차. 이호준. 정우진



목차

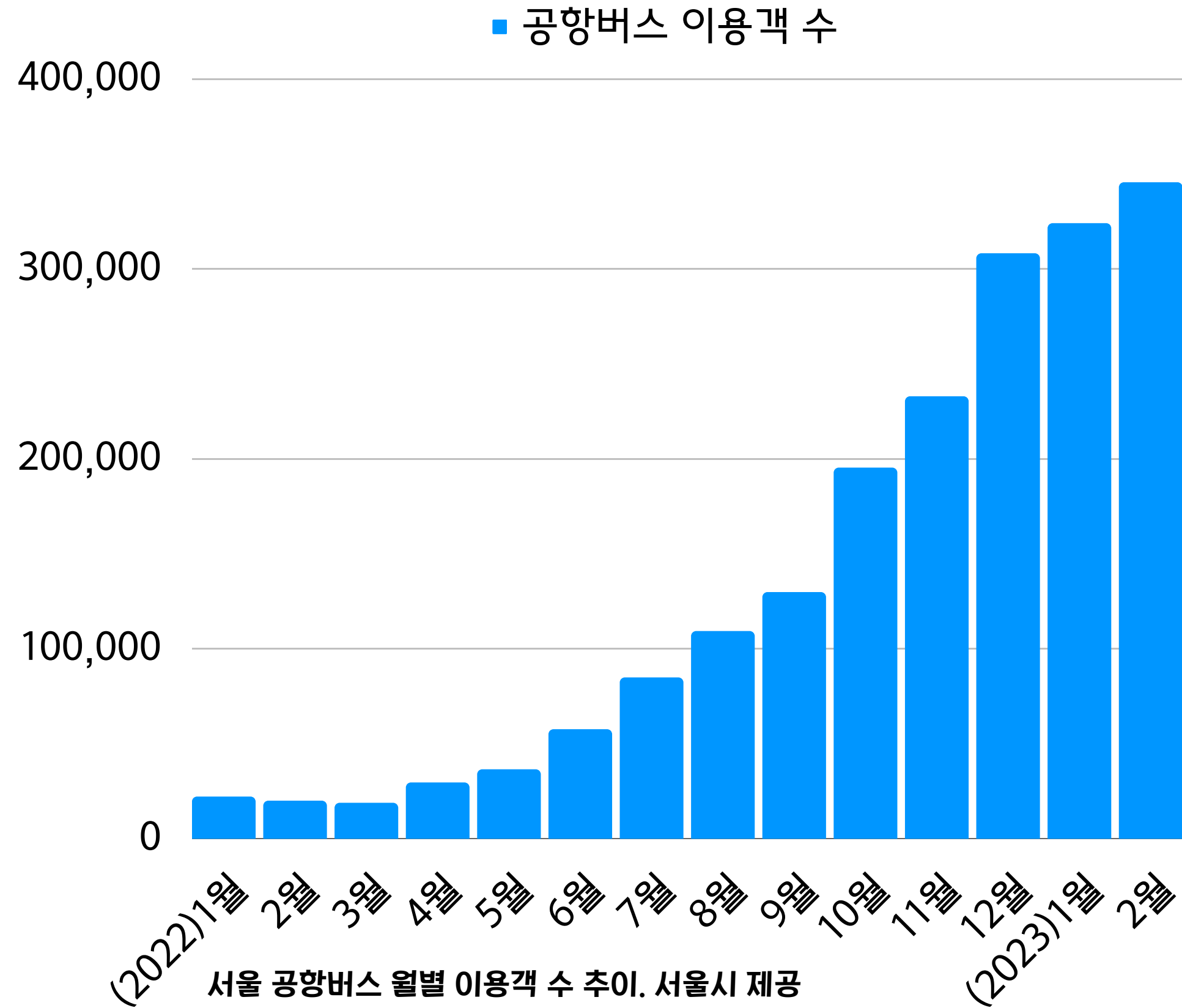
- ✈ 기획 배경
- ✈ 서비스 소개
- ✈ 시연
- ✈ 성능.기술 스택 및 아키텍처
- ✈ 마무리

기획 배경 – 문제인식

해외여행 급증

서울 공항버스 이용객 100만명

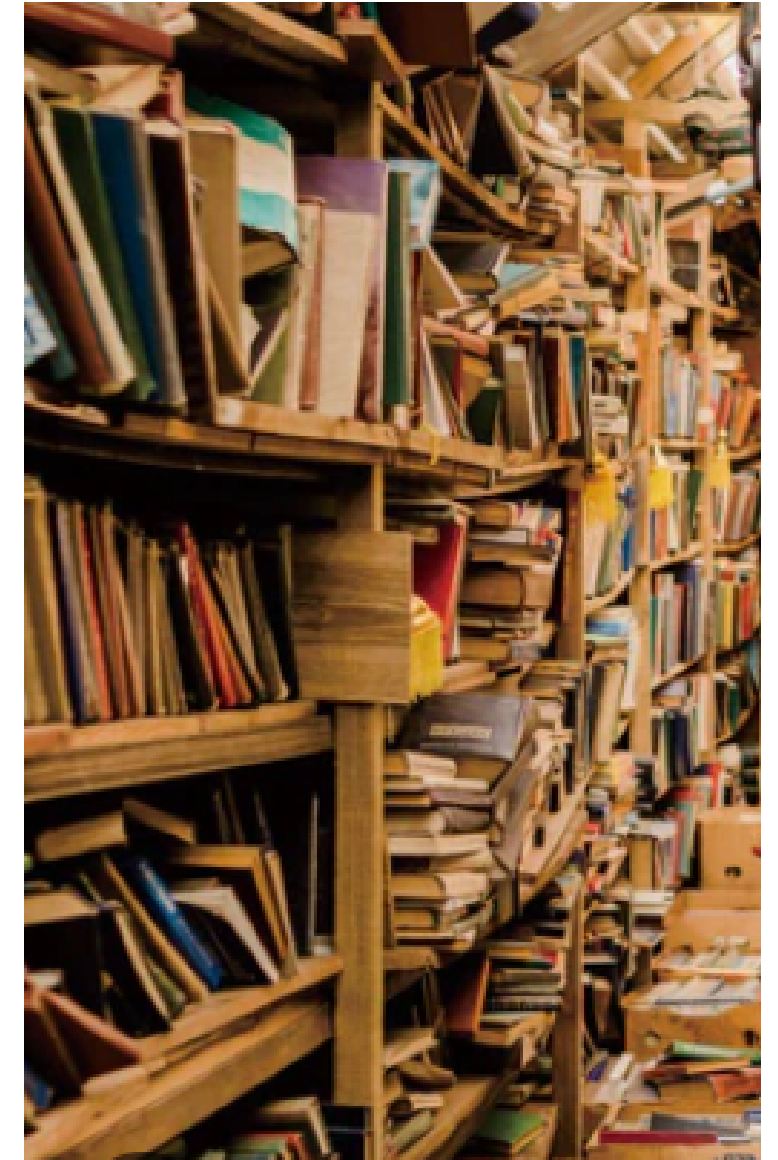
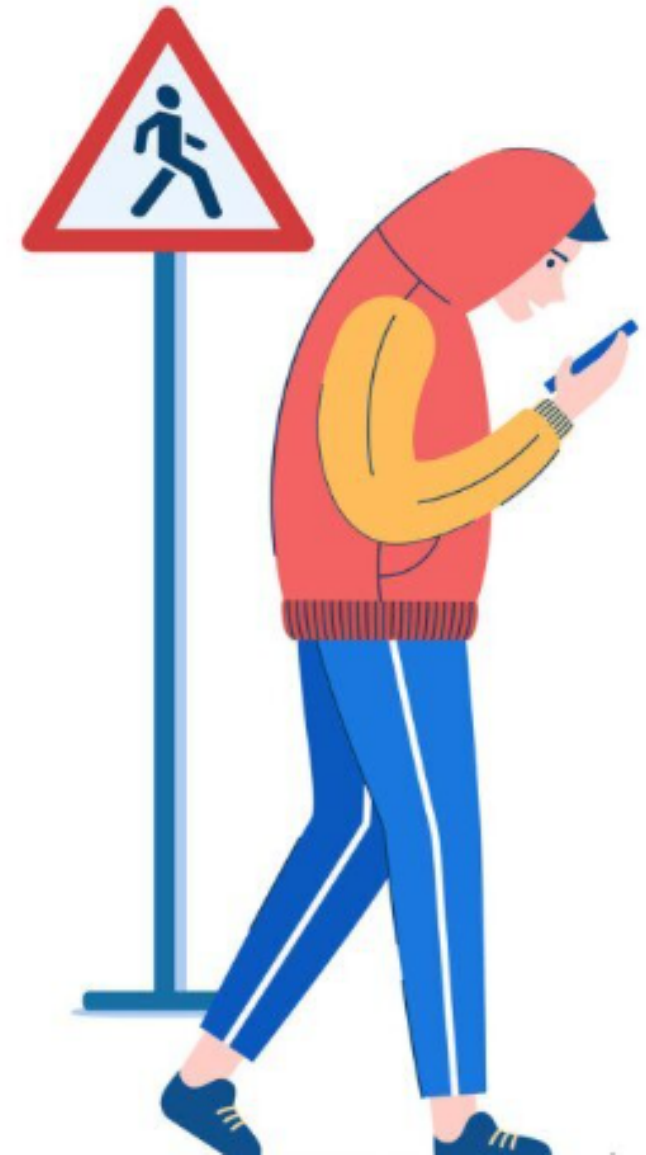
▶ 여행(국내/해외) 수요 증가



기획 배경 – 문제인식

여행 기록 – 불편한 점

- 사진만으로 현장 기록 어려움
- 텍스트 입력의 어려움
- 여행 타임라인 파악 어려움



기획 배경 – 원인 분석

 사진만으로 현장 기록 어려움

→ 앨범 속 사진만으로는 당시의 감정, 상황 등을 기억하기에 어려움

 텍스트 입력의 어려움

→ 여행지에서 사진과 텍스트를 매번 직접 입력하기에 어려움

 여행 타임라인 파악 어려움

→ 사진들이 여행 단위로 구분되어 있지 않아 한눈에 알아보기 어려움

기획 배경 – 해결 방안

사진만으로 현장 기록 어려움

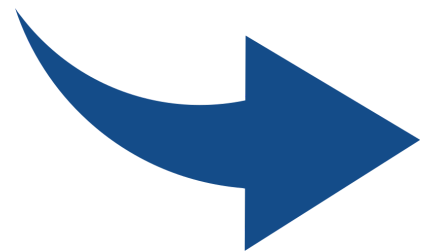
사진과 음성으로 생생한 현장을 기록

텍스트 입력의 어려움

Speech To Text를 이용한 입력 자동화

여행 타임라인 파악 어려움

설정된 여행 단위로 여행 기록을 관리



인공지능(음성) 도메인을 활용한 서비스 기획



서비스 소개

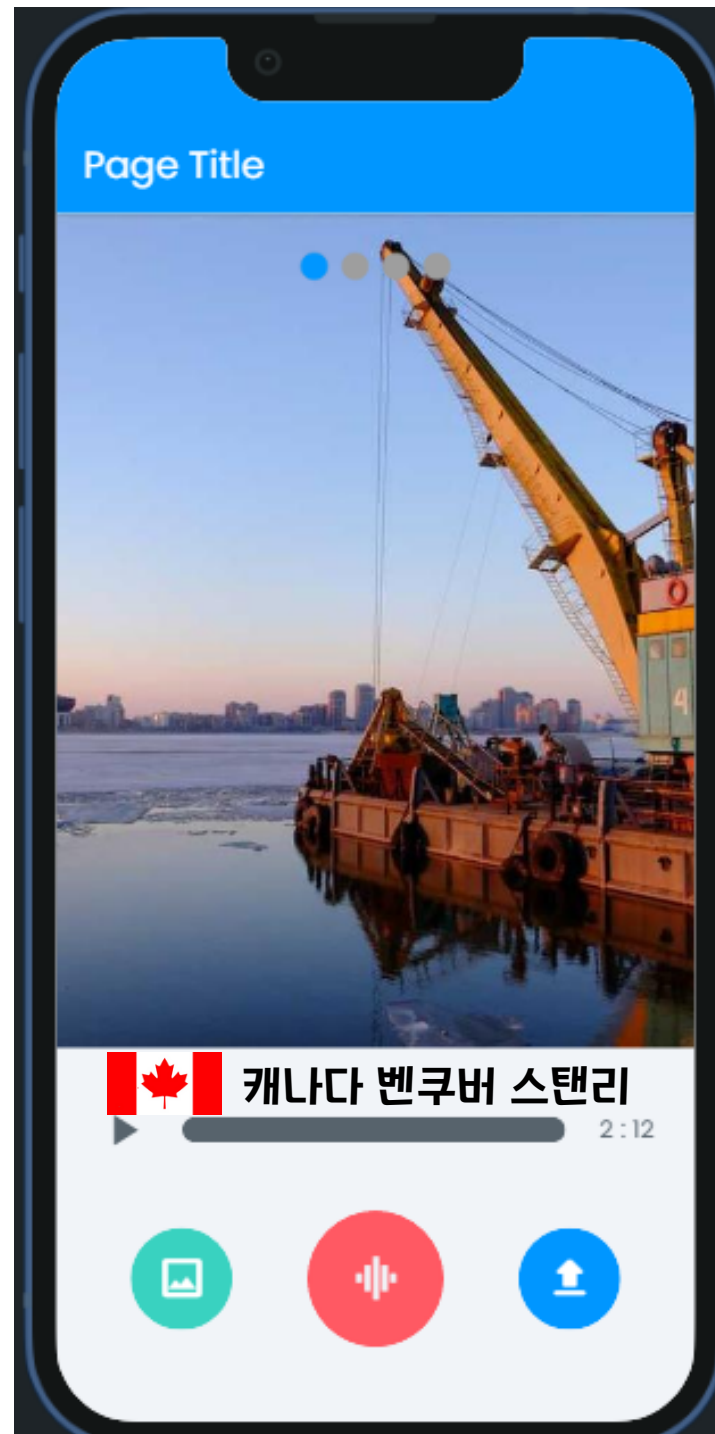


서비스 소개

- ✈ 핵심 기능
- ✈ 차별점



핵심 기능 여행 기록 하기



여러장의 사진



GPS 기반 위치 정보



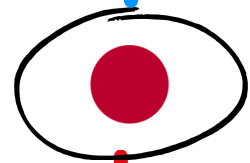
감동적인 순간의 생생함을 기록

핵심 기능 여행 루트 확인



인천 국제 공항

2023.06.29



나리타 국제 공항

2023.06.29



교토시 히가시야마구 가와바타초

2023.07.03



나라별 포스트 자동 구별



여행 시간 순서 기록

핵심 기능 여행 기록 상세



교토시 히가시야마구 가와바타초

2023.07.03



생생한 기록을 확인

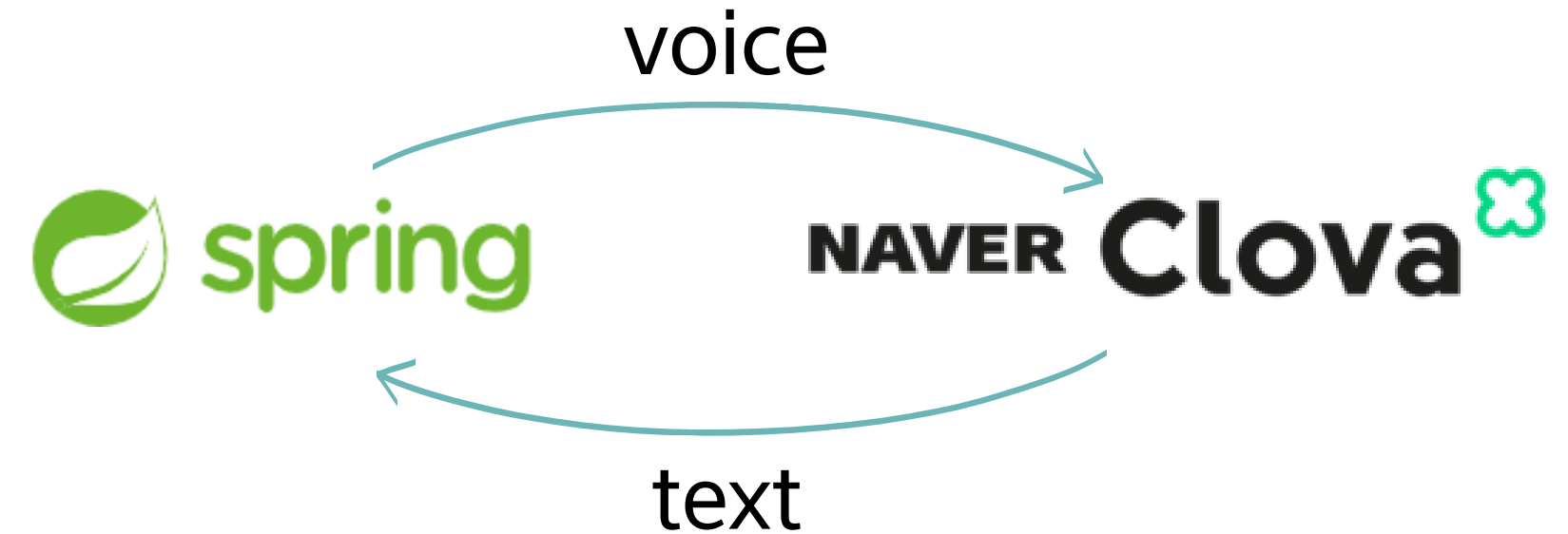
여러 뜻으로 많은 것을 얻은
여행이었다. 게스트하우스는
여행의 또 다른 재미!



음성 데이터 텍스트로 변환

핵심 기능 Speech to Text

CLOVA Speech API



1. 네이버 자체 엔진 (NEST) 사용

NEST(Neural End-to-end Speech Transcriber)

- 네이버에 축적된 풍부한 데이터 스스로 학습 및 성능 자동 개선하는 머신 러닝 적용
- **노이즈 필터링**, **금칙어 필터링**, 키워드 부스팅 등을 통한 인식 결과 개선
- 경쟁사 대비 5% 이상 높은 한국어 인식 정확도와 3배 이상 빠른 인식 속도

차별점

BadWordFiltering



민감 키워드 추출
[start end,"word"]



Serialize
Object,File

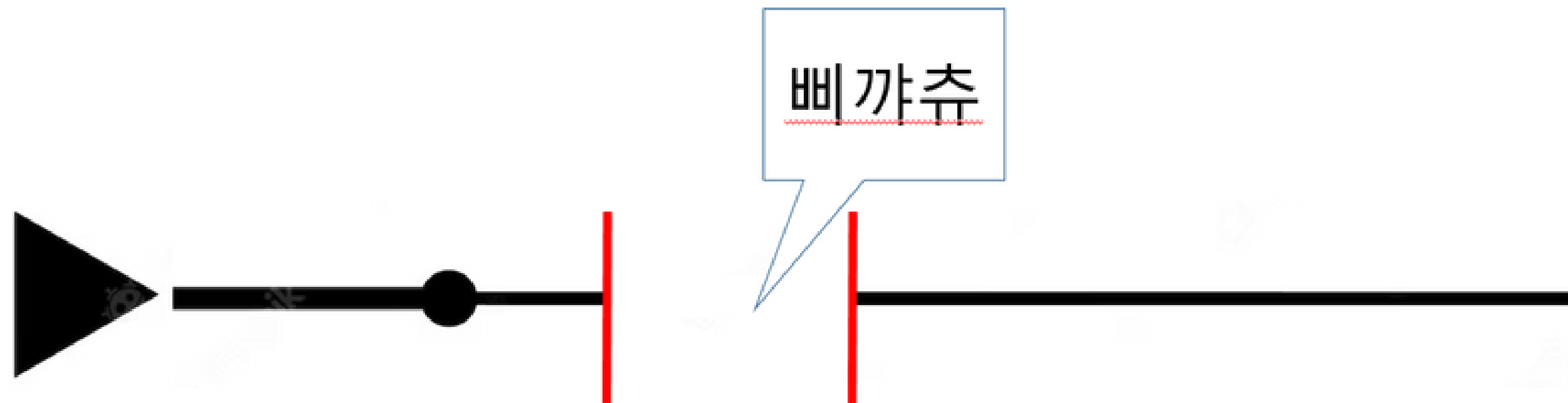


Flask

```
{  
  "words": [  
    [160,440,"크게"],  
    [520,1080,"파스."]  
  ],  
}
```

차별점

BadWordFiltering



시연



부하 테스트 nGrinder

시나리오 : 유저 사용 빈도가 높으며, 많은 DB 리소스를 조합해 결과를 보여주는
메소드를 대상으로 선정

대상 메소드 : 타임라인 조회 (페이징 처리) 성능 테스트

테스트 조건

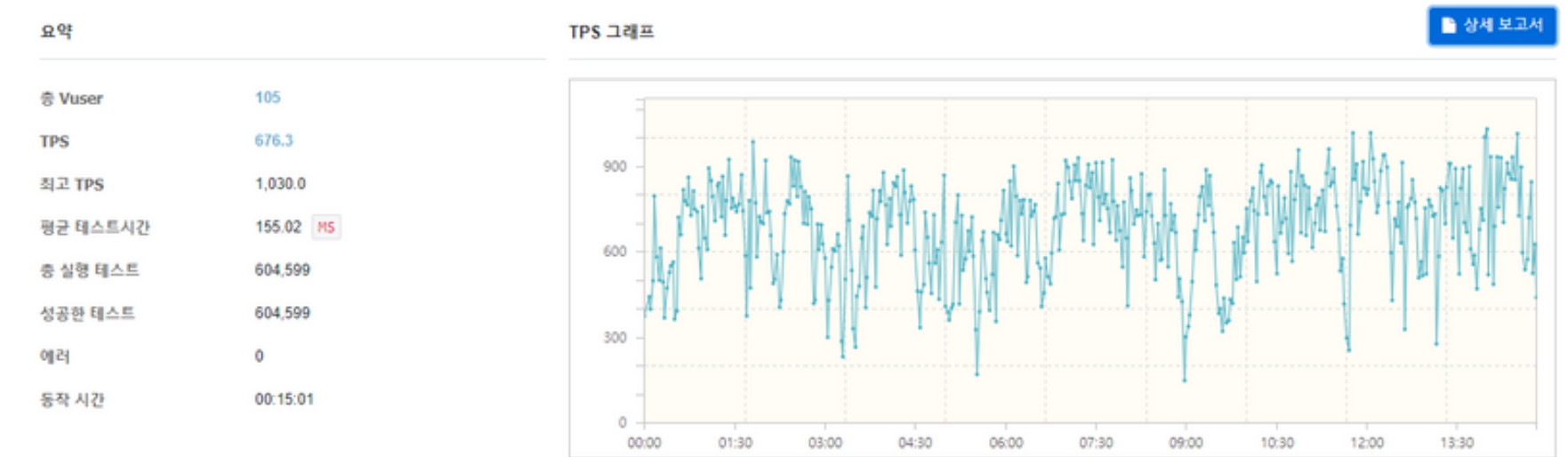
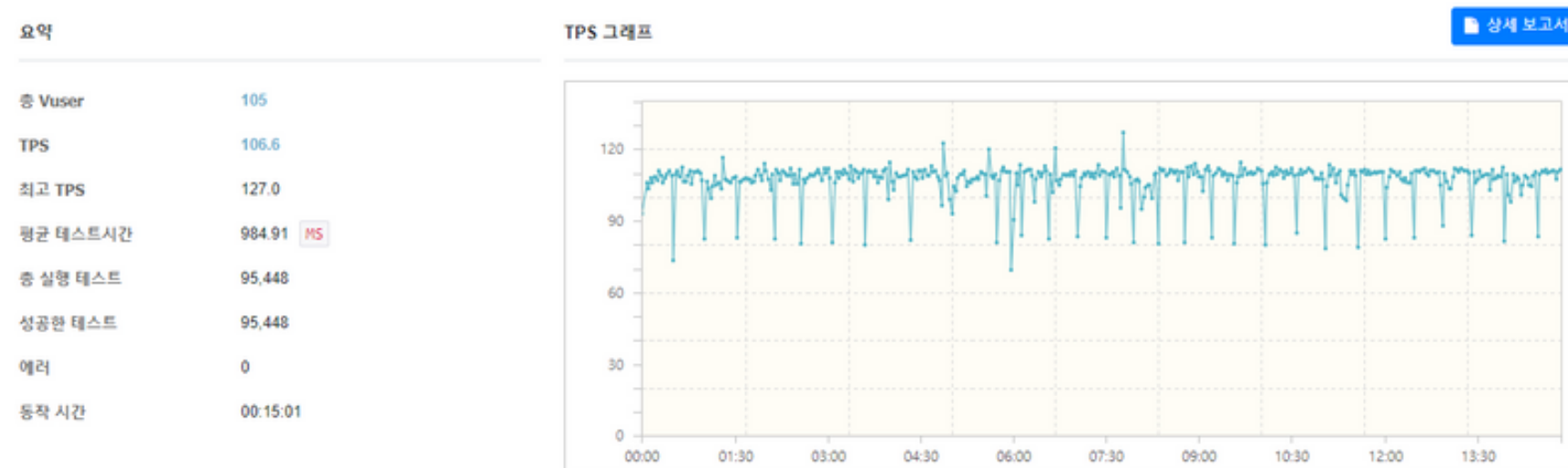
- 1일 예상 사용자(DAU) : 10만명
 - 1명당 1일 평균 접속 수 : 15회
 - 1일 총 접속 횟수 : 150만회
 - 1일 평균 RPS(Request Per Second) : 약 17회
- VUSER : 105, 테스트 시간 : 15min

부하 테스트 nGrinder

테스트 환경 : test 서버 컨테이너 & time_line 테이블에 100,000개 데이터

JPA 페이징 처리 (Pageable 사용)

Redis 캐싱  redis



```
127.0.0.1:6379> keys *
```

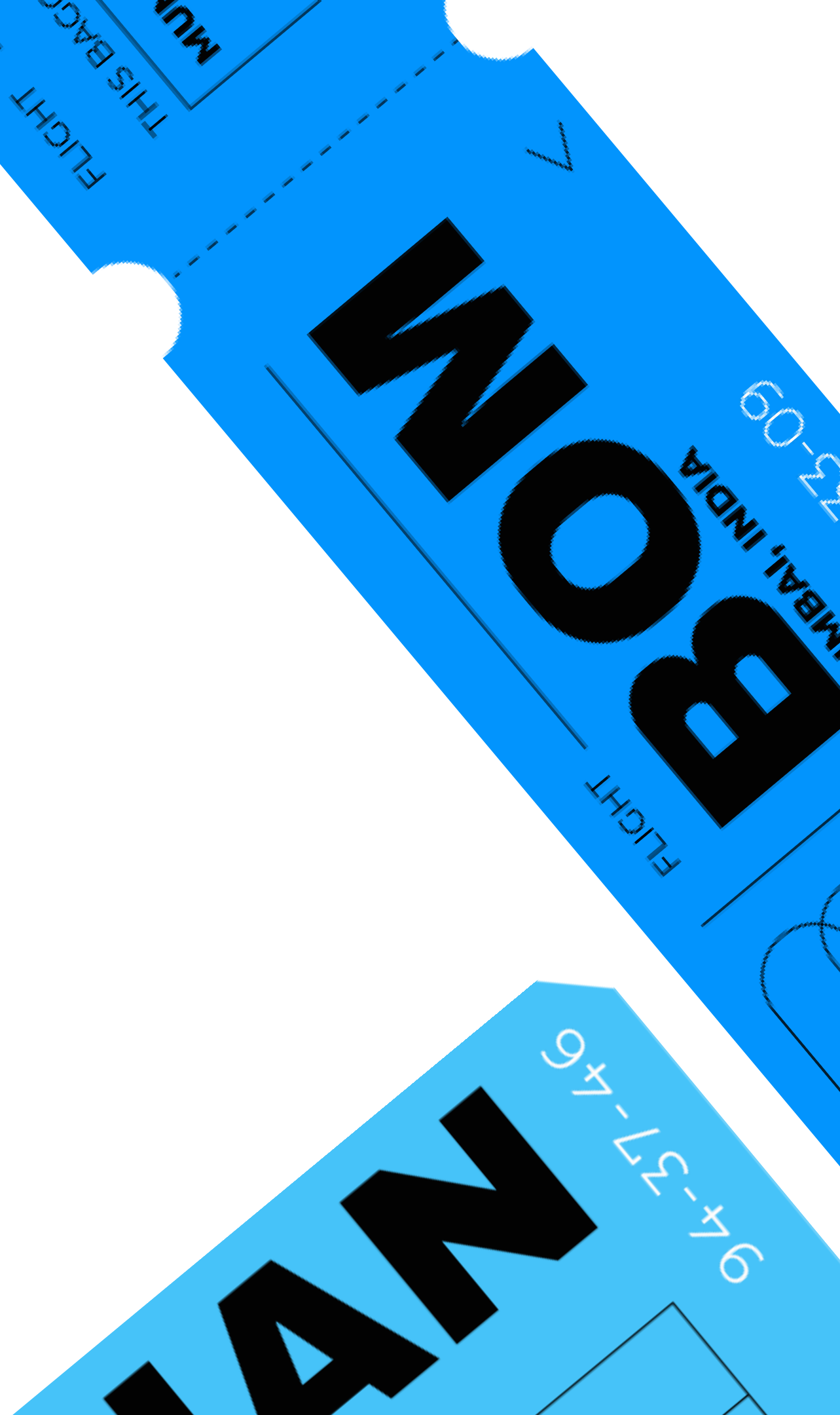
- 1) "page"
- 2) "page:200"
- 3) "page:3000"
- 4) "page:5000"
- 5) "page:800"

<redis key value>

검색한 페이지를 키 값으로 redis에 저장하여 TPS를 약 106.6에서 676.3으로 최적화
UX적으로 최선의 성능을 제공

* TPS : 초당 몇개의 테스트를 처리할 수 있는지 (Test Per Second)

기술 스택 및 아키텍처



기술 스택

BackEnd



Spring security & OAuth2

FrontEnd



Provider

DB

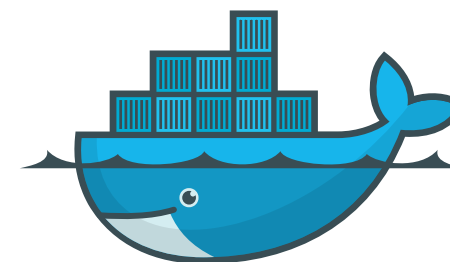


redis

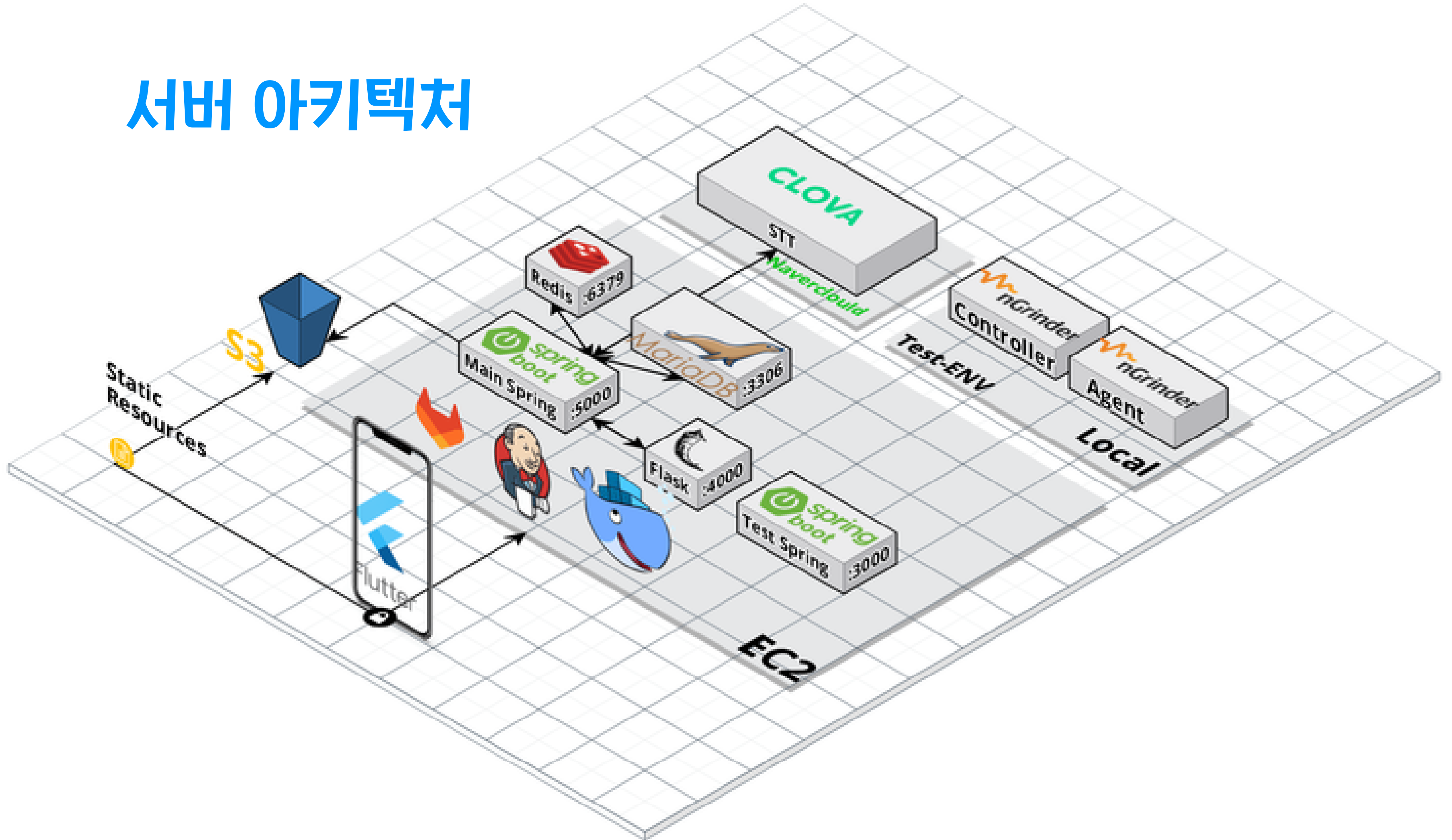
Infra



Jenkins



서버 아키텍처





Q & A

감사합니다.

Q & A

사진만으로 현장 기록 어렵다는데[동영상으로 하면안되나요.

⇒ 동영상으로 기록할 때 만족하지 못하면 다시촬영해야하는데에 비하면 사진은 빠르고 간단하다는 장점이 있다. 하지만 소리가 기록이 안된다는것에서 저희 팀은 사진에 집중하기로 했습니다.

인공지능 음성 트랙인데 모델을 사용하지 않은 이유가 있나요?

⇒ 뭐

레디스 에 관련된 정보가 업데이트 될시 어떻게 반영을 했나요?

⇒ 페이징 넘버를 키 값으로 레디스에 저장했고, 해당 데이터는 **5**분 저장 및 유저 정보 변경, 포스트 정보 수정 시 레디스를 비우도록 하여 최신 정보가 업데이트 되도록 했습니다.

Q & A

BAD WORD FILTERING 데이터는 얼마만큼 넣었고 , 어떻게 실행 실행 했나요>
⇒ **BAD WORD**가 들어 있는 데이터를 정의 하여서 , 필터링을 하도록 하였습니다.

부하테스트 **NGRIDER**를 사용한 이유?

서버 성능 테스트용 툴은 **ngrinder, loadrunner, jmeter, Gatling, vegeta** 등 많음
유료 및 비 주류를 제외하면, **nGrinder**와 **JMeter** 중 선택

nGrinder : 화면이 심플 및 예약을 통해 주기적인 테스트 가능.

IDE에서 **Groovy** 스크립트 자동 생성. 쉽게 테스트 시나리오 작성 가능함,

테스트 서버 환경을 어떻게 구성했는지?

동일한 **EC2** 인스턴스에 특정 브랜치 **WEBHOOK**을 사용한 **TEST**용 컨테이너를 사용했고,
MARIADB에 **TEST**용 스키마를 생성해서 같은 환경에서 테스트를 진행했습니다.