Big Data Processing & Trend Analysis

Big Smile

김가윤 이규은 정석준 조인식 추연호

01
프로젝트 개요

- 배경
- 주제 및 컨셉

02
프로젝트 목표

- 팀 목표

03
프로젝트 설계

- UI / UX 설계
- 아키텍쳐 설계
- DB 스키마 설계
- 기능 정의서

04
마일스톤

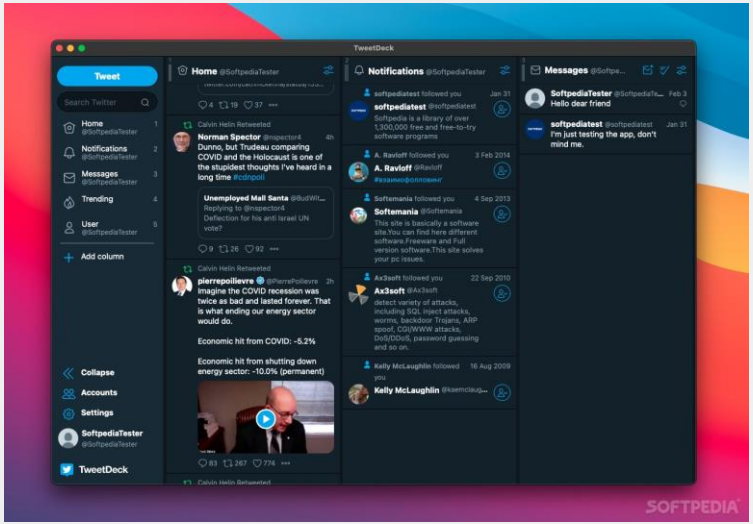
05
팀원 소개

- 개인 목표
- 기술 스택 및 R&R
- 개발 기능 및 역할

01 프로젝트 개요 - 배경

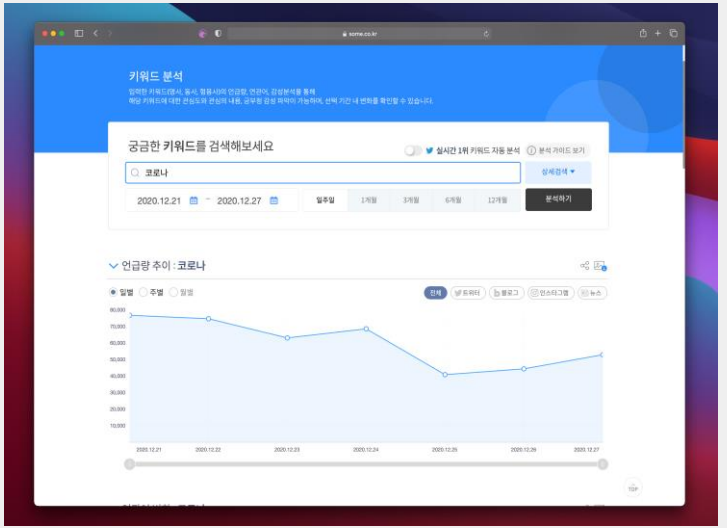
TweetDeck

키워드 별 실시간 피드 목록 제공



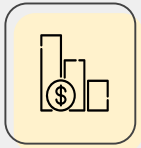
Sometrend

키워드 별 통계 데이터 제공



TweeTrend

> 코로나 연관어 기반 트렌드 분석 대시보드 서비스



주요 기능

- 트위터에서 언급되는 코로나 관련 키워드를 포함한 실시간 피드를 제공한다.
- 키워드를 분석하고 통계를 사용자에게 대시보드 형태로 제공한다.

확장 가능성

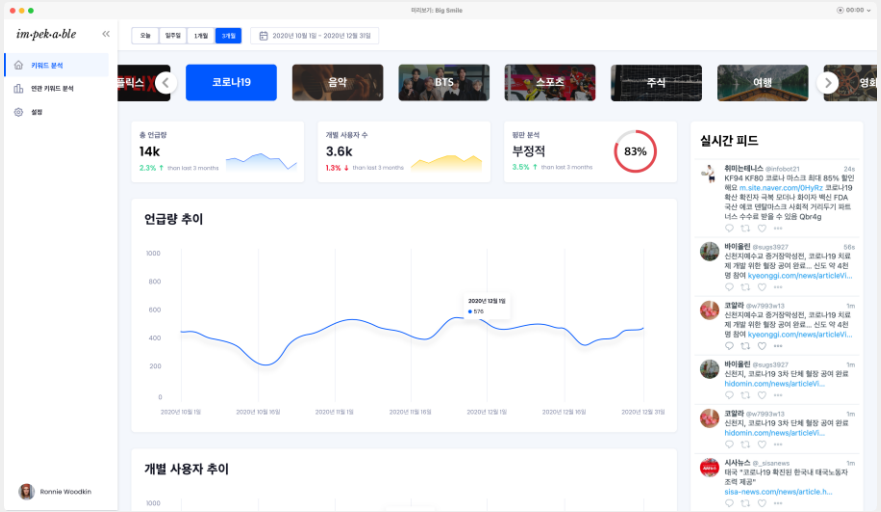
- 특정 주제에 국한되지 않고, 유저가 검색한 키워드에 대한 트렌드 분석 페이지를 제공한다.
- 트위터 외에 다른 SNS의 데이터를 추가적으로 분석한다.

02 **프로젝트 목표** – 팀 목표

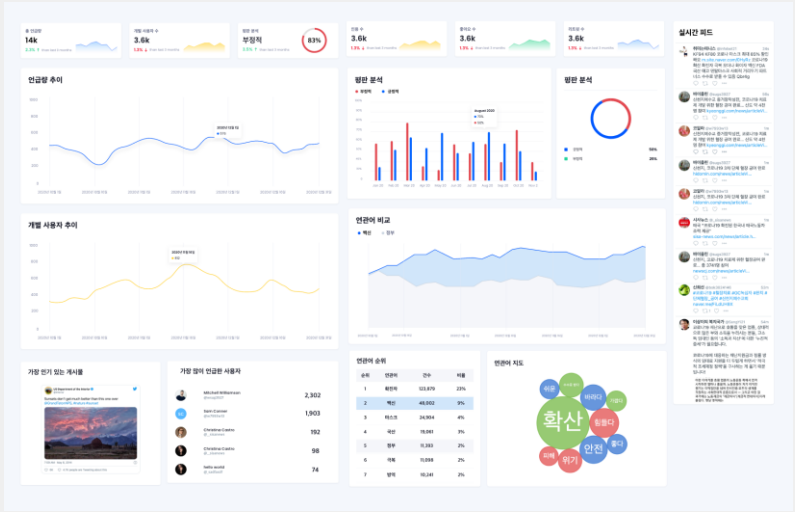
“ TweetDeck과 SomeTrend를 벤치마킹하여 각자 관심있는 기술을 습득하고, 기술을 통합하여 **TweeTrend**라는 하나의 서비스를 완성한다.

“ **TweeTrend** 서비스를 개발하면서 사용한 기술에 대해 서로에게 설명할 수 있을 정도로 관심 분야 기술에 대한 이해를 높이는 것.

03 **프로젝트 설계** – UI/UX 설계

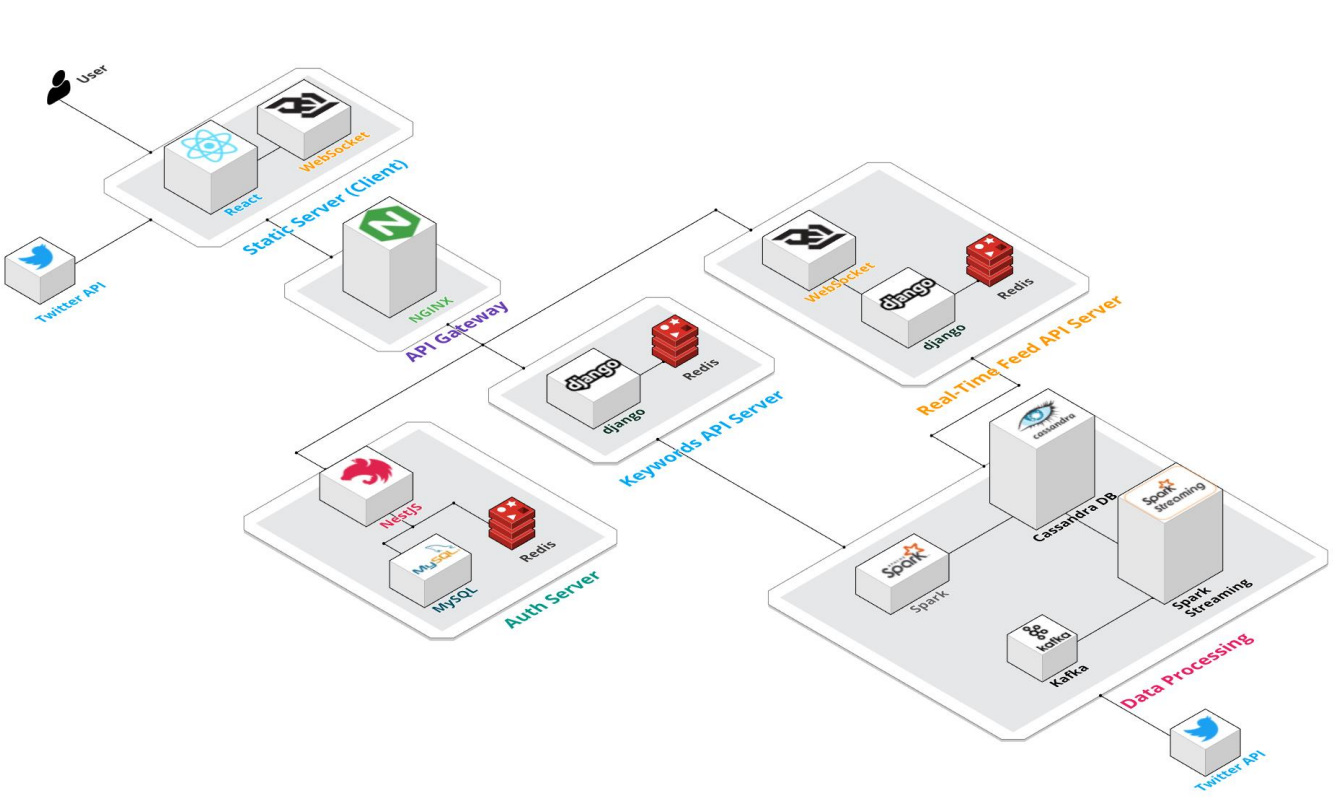


메인 페이지

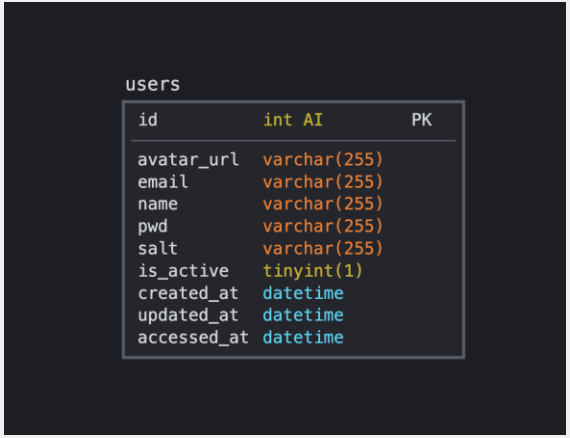


데이터 컴포넌트 모음

03 **프로젝트 설계** – 아키텍쳐 설계



03 **프로젝트 설계** – DB 스키마 설계



Auth-Server Database (MySQL)

- 유저의 회원가입 정보 및 프로필 정보를 가진다.

03 **프로젝트 설계** – DB 스키마 설계

Data Processing server Database (Cassandra DB)

03 **프로젝트 설계** – 기능 정의서

Category	Feature	Detail
회원가입 & 로그인	일반 회원가입	
회원가입 & 로그인	이메일 인증	
회원가입 & 로그인	소셜 로그인	선택 기능으로 기간에 여유가 있을 때 구현. (트위터 or 구글 or 카카오)
유저 관심 키워드 관리	입력 폼 개발	회원 가입이 완료되면 폼을 제공해 유저에게 관심 주제를 입력 받는다. Ex) 코로나, BTS, 넷플릭스
유저 관심 키워드 관리	관심 키워드 추가/삭제	유저는 관심 있는 주제를 추가하거나 삭제할 수 있다.
유저 관심 키워드 관리	관심 키워드 별 페이지 이동	메인 화면에는 하나의 주제에 대한 통계 데이터가 보여지고 상단 탭으로 유저가 관심 있는 주제별로 이동할 수 있다.
통계 데이터	기간 별 통계 데이터	메인 화면에는 키워드에 대한 여러 통계 데이터를 보여준다. 유저는 일간, 주간, 월간, 3개월간 통계를 선택할 수 있다.
통계 데이터	언급량 추이	해시태그의 언급량 추이 그래프를 보여준다.
통계 데이터	개별 사용자 추이	해시태그를 사용한 개별 사용자 추이 그래프를 보여준다.
통계 데이터	컨텐츠 분석 – 인기 게시물	기간 내에 가장 인기있는 게시물 목록을 보여준다.
통계 데이터	컨텐츠 분석 – 사용자	기간 내에 해당 해시태그를 가장 많이 사용한 사용자 목록을 보여준다.
통계 데이터	연관어 분석 – word cloud	관련 키워드를 word cloud 형태로 보여준다.
통계 데이터	연관어 분석 – 순위	연관어 순위를 보여준다.
통계 데이터	연관어 분석 – 빈도수	연관어 빈도 수를 비교하는 그래프를 보여준다.

04 마일스톤

M1	DevCamp 기간 내에 반드시 완료해야하는 목표	- Adobe XD와 Figma를 활용한 디자인 프로토타입 제작 - 5개의 코로나 관련 키워드에 대한 트렌드 분석 제공 (코로나, 백신, 방역, 확진자, 여행) - 특정 토픽에 대한 트윗을 실시간으로 제공하는 실시간 피드 개발 - 기본적인 트렌드 분석 대시보드 구현 => 총합, 평균 등 단순 통계치 ex) 총 언급량, 언급량 추이, 연관어 분석 - 데이터 처리 Kafka - Spark - Cassandra 연결하여 데이터 통신 파이프라인 구축 - Spark Streaming과 Spark Consumer, Spark Cassandra Connector 구현 - 키워드 수에 따라 Scale-Out 가능한 구조를 설계하여 Kafka broker(topic 및 partition) 구현 - 유지 보수가 원활하도록 분산 DB 구축 및 분석에 필요한 추가적인 데이터 모델링 - 트위터 소셜 로그인을 사용한 유저 인증 (토큰 기반 인증으로 MSA로 동작) - AWS 내에서 개발 환경 구축 및 전체 서비스 통합 - 기간별 데이터 제공 기능 구현
M2	DevCamp 기간 안에 어떻게든 최선을 다한다면 가능할 수 있을 것 같은 목표	- 코로나 관련 키워드 개수 늘리기 (재택근무, 변이, 사망자, 언택트, 비대면, 사회적 거리두기, 지원금 등) - 추가적으로 분석 기법이 들어간 Trend 컴포넌트들을 완성 (워드 클라우드, 버블 차트 등) - 트위터와 상호작용할 수 있는 기능 구현 (좋아요, 리트윗, 댓글 등) - 데이터 유실이 없도록(또는 속도와의 Trade-Off) Kafka 파라미터 튜닝 및 키워드 확장 적용 - DB 퍼포먼스 향상을 위한 튜닝 - 웹 서버와 웹 인터페이스 연결에 있어 API 게이트웨이를 통해 짜임새 있는 MSA 구조 구축 - 트위터에서 제공하는 데이터 최대한 활용하여 통계 컴포넌트 늘리기 (사용자 위치, 작성 디바이스 등) - 웹페이지 모바일 반응형 지원 - 랜딩 페이지 구축 (홈 화면, 서비스 소개)
M3	DevCamp 기간 내에는 불가능하지만 궁극적으로 만들고자 하는 목표	- 유저가 검색한 검색어 데이터 실시간 반영으로 확장하기 - 유저가 검색한 내용을 twitter API 의 파라미터로 입력해 실시간으로 데이터를 받아옴(유동적) - twitter API의 Sample Stream을 이용해 많은 양의 전체 tweets 수를 샘플링하여 감당할 수 있는 양의 데이터에서도 동향을 파악할 수 있도록 함 - 해당 검색어와 코로나와의 상관성을 분석하여 인사이트 제공 - 분석 결과 개인 유저 별 저장 - 트위터 API 뿐만 아니라 다른 데이터 소스까지 확장(인스타그램 등)

04 마일스톤

M1	DevCamp 기간 내에 반드시 완료해야하는 목표	한 키워드에 대해 데이터 처리부터 프론트엔드까지 완성된 하나의 흐름 만들기
M2	DevCamp 기간 안에 어떻게든 최선을 다한다면 가능할 수 있을 것 같은 목표	부가적인 기능을 추가하고, 서버의 성능과 안정성을 높이는 것
M3	DevCamp 기간 내에는 불가능하지만 궁극적으로 만들고자 하는 목표	유저의 검색어 기반으로 확장하고, 다른 SNS 데이터까지도 활용하는 것

05 팀원 소개

T E A M

김가윤	이규은	조인식	정석준	추연호
데이터 전처리 및 DB 전달 (Spark)	가공 데이터 분산 저장 및 백엔드 전달 (Cassandra DB)	데이터 수집 및 분산 처리 (Kafka)	백엔드 개발 및 MSA 구성 (Django)	프론트엔드 개발 및 인증 서버 개발 (React & TypeScript)

05 **팀원 소개** – 개인 목표

김가윤

데이터 처리 - Spark & Spark Streaming

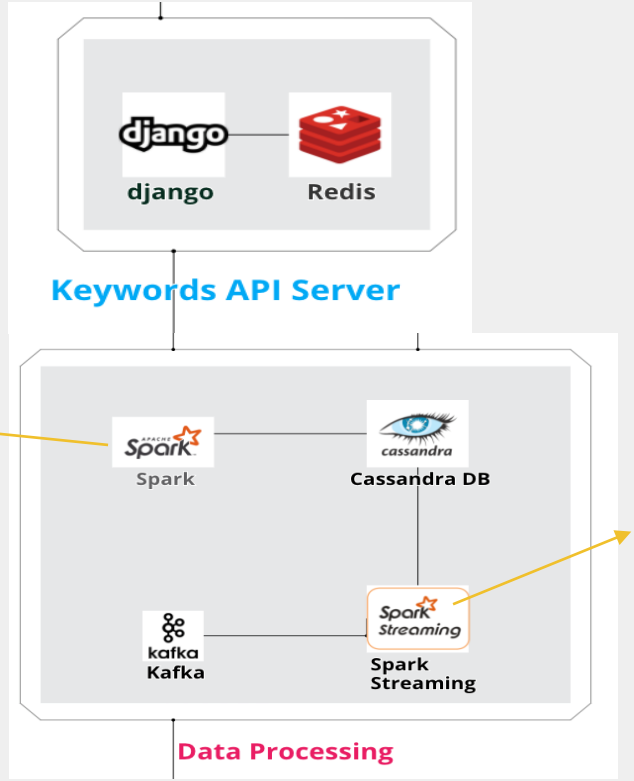
“ Kafka에서 받아온 실시간 데이터를 Cassandra DB에
안정적으로 전달하고 요구사항에 맞춰 알맞게 가공하려고 합니다. ”

05 **팀원 소개** – 기술 스택 및 R&R

김가윤

데이터 처리 - Spark & Spark Streaming

1. Cassandra DB 데이터 가공
2. Django 데이터 전달 기능 구현
3. 메모리 저장 방식의 빠른 성능



1. 데이터 전처리를 맡아 Kafka와 Cassandra DB 중간 연결 담당
2. 실시간 데이터처리 가능

05 **팀원 소개** – 개발 기능 및 역할

김가윤 데이터 처리 - Spark & Spark Streaming

진행 상황

1. Kafka 브로커에 저장된 메시지를 가져오기 위한 **Spark Streaming 컨슈머** 기능 구현
2. Cassandra DB에 데이터를 저장하기 위한 **Spark Cassandra Connector** 기능 구현

향후 계획

1. Kafka에서 수집한 데이터를 가공하여 **Cassandra DB에 저장**
2. 핵심 키워드 분석을 위한 **데이터 가공&분석** 이후 Django와 통신하기
3. AWS 서버 구축 및 **데이터처리 기술 통합**

05 **팀원 소개** – 개인 목표

이규은

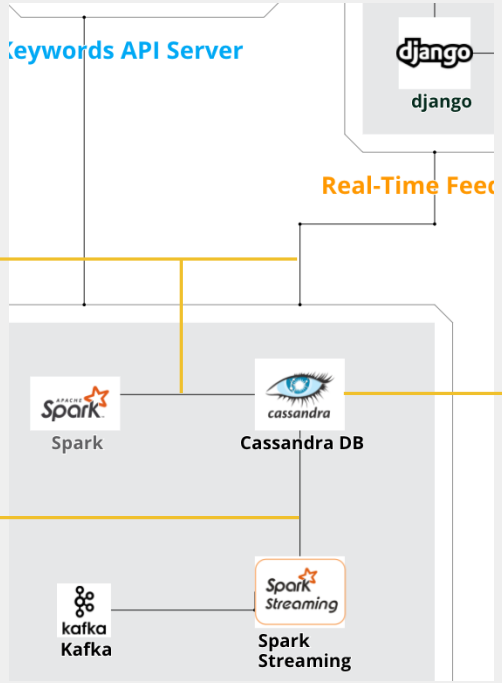
데이터 처리 - Cassandra DB

“ 빠른 데이터 CRUD를 가능하게 하여 실시간성을 확보하고,
장애 발생 시에도 안정적으로 데이터를 공급할 수 있도록 한다. ”

05 **팀원 소개** – 기술 스택 및 R&R

이규은

데이터 처리 - Cassandra DB



데이터 전달

가공 데이터 분산 저장

1. 데이터 모델링
2. 복제된 데이터 유효성 검사
3. 퍼포먼스 향상을 위한 DB 최적화

05 **팀원 소개** – 개발 기능 및 역할

이규은 데이터 처리 - Cassandra DB

진행 상황

1. 실시간 피드에 필요한 테이블을 Twitter API와 흡사한 형태로 모델링
2. AWS Cassandra DB 구축
3. 실시간 피드 서버와 DB 연결
4. 가공 데이터를 테이블 형태에 맞게 재정렬

향후 계획

1. Spark & Spark streaming과 연결
2. 트렌드 분석 테이블 모델링
3. 복제된 데이터 유효성 검사 및 관리
4. 분산 배치 전략, 일관성 레벨, 데이터 처리 정책을 조정하며 퍼포먼스 향상

05 **팀원 소개** – 개인 목표

조인식

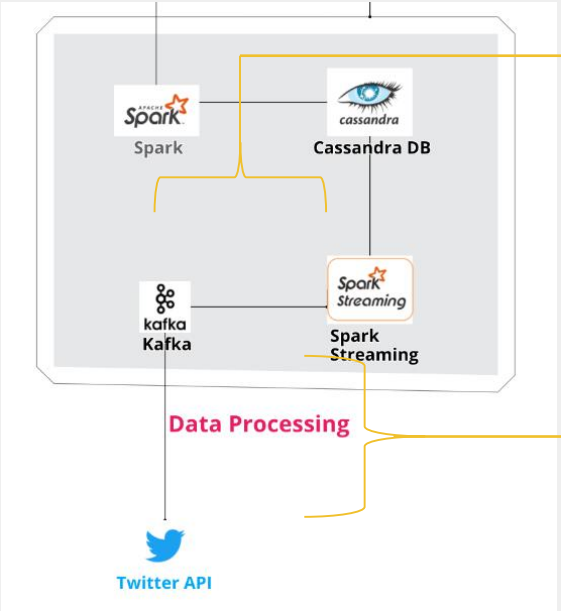
데이터처리 - Kafka

“ 실시간으로 생산되는 많은 양의 Twitter 데이터를
빠르고 유실없이 전달하고자 합니다. ”

05 **팀원 소개** – 기술 스택 및 R&R

조인식

데이터처리 - Kafka



1. raw data 를 가공하기 위해 spark streaming 과의 연동
2. data consumer 로서 성능 최적화를 위한 파라미터 설정

1. Twitter API 로부터 게시되는 글을 실시간으로 수집
2. data producer 로서 성능 최적화를 위한 파라미터 설정

05 **팀원 소개** – 개발 기능 및 역할

조인식 데이터처리 - Kafka

진행 상황

1. Twitter API를 통해 ' 코로나 ' 와 관련된 **실시간 데이터 수집** 기능 구현
2. 분산 처리를 위한 Zookeeper / Kafka **클러스터 환경 구축**
3. Kafka 브로커를 통해 메시지를 전달하기위해 Spark Streaming 와 연동

향후 계획

1. Zookeeper 클러스터 확장 (서버 1대 -> 3대)
2. 분산 처리의 **성능 테스트 및 최적화**를 위한 파라미터 설정
3. 키워드 추가에 따라 데이터 수집 및 처리 시스템 조정 (**topic 및 partition 관리**)

05 **팀원 소개** – 개인 목표

정석준

백엔드 - 기능 API 서버 & MSA 구축

실시간 서비스에 있어, 각 기능을 철저하게 독립적으로 분리하여
안정적인 서비스 환경을 제공할 수 있는 서버를 구축하고자 합니다.

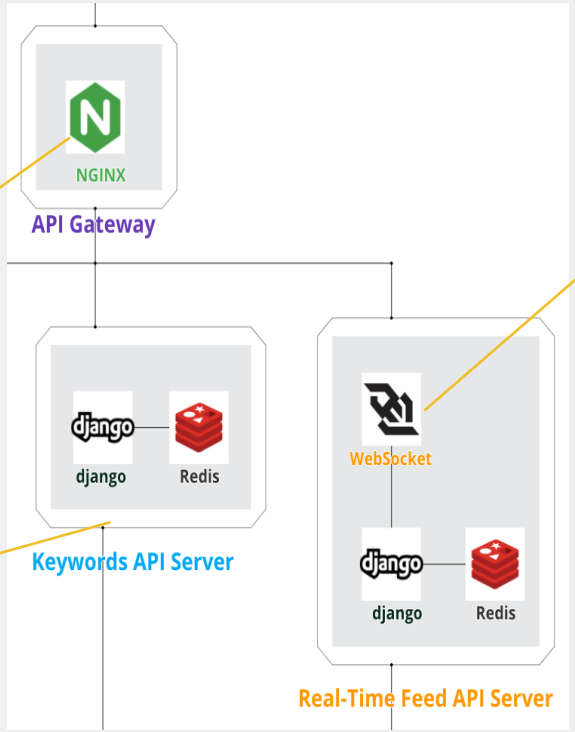
05 **팀원 소개** – 기술 스택 및 R&R

정석준

백엔드 - 기능 API 서버 & MSA 구축

1. API 게이트웨이를 통한 MSA 구조 구축
2. 웹서버와 웹 인터페이스를 통한 연결

1. 키워드 조회에 대한 데이터 제공
2. 실시간 데이터 캐싱



1. 웹소켓을 통한 실시간 트윗 전송

05 **팀원 소개** – 개발 기능 및 역할

정석준 백엔드 - 기능 API 서버 & MSA 구축

진행 상황

1. Cassandra DB 연동 및 조회 구현
2. **웹소켓**을 통한 실시간 데이터 전송 구현
3. **REST API**를 통한 트윗 조회 기능 구현
4. 피드 API 서버 AWS 구축 및 업로드

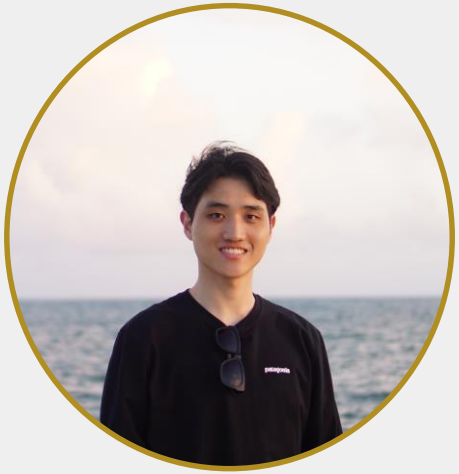
향후 계획

1. **권한**에 맞는 데이터 제공하는 기능 구현
2. **캐싱**을 통한 실시간 서비스의 성능 보장
3. **API 게이트웨이** 구축
4. **소켓 연결**을 통한 통신 효율 높이기

05 **팀원 소개** – 개인 목표

추연호

프론트엔드 개발 및 인증 서버 개발

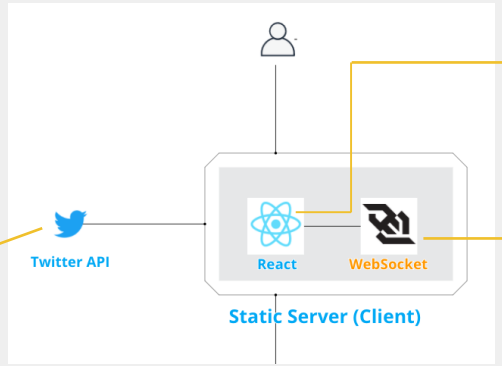


> 웹 앱 결과물 뿐만 아니라, 디자인시스템을 함께 만들어
> 팀 내 소통 가능하고, 쉽게 확장할 수 있도록 개발하기

05 **팀원 소개** – 기술 스택 및 R&R

추연호

프론트엔드 개발 및 인증 서버 개발



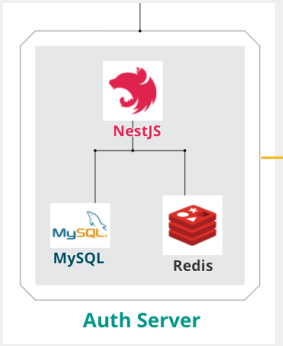
트렌드 대시보드 형태의 웹 애플리케이션 구현

차트 라이브러리를 활용한 **통계 데이터 시각화**

서버와 웹소켓 통신하여 **실시간 피드** 구현

트위터 API를 사용하여 좋아요, 리트윗 등

트위터와 상호작용



토큰 기반 인증 서버 구현

트위터 소셜 로그인 사용

05 **팀원 소개** – 개발 기능 및 역할

추연호

프론트엔드 개발 및 인증 서버 개발

진행 상황

1. 기초 UI 프로토타입 제작 & 글로벌 테마 설정
2. Storybook 활용 컴포넌트 개발 **환경 구축 및 배포**
3. 차트 라이브러리 적용, 통계 컴포넌트 구현
4. Twitter Stream API 활용 **웹소켓 Mock 서버 구축**
5. Mock 서버와 클라이언트 **실시간 피드 소켓 통신** 구현

향후 계획

실시간 피드
- 백엔드와 연결

트렌드
- 막대, 도넛, 꺾은 선 등 차트 구현
- 연관어 순위 테이블 구현
- 워드 크라우드 구현
- 트렌드 API와 연결 및 데이터 연동
- **기간별 필터** 구현

인증
- JWT 토큰 기반 인증 서버 구현
- 트위터 **소셜 로그인** 구현

확장
- 트위터 API로 트위터와 상호작용 구현
- 통계 데이터 저장 기능 구현
- 랜딩 페이지

THANK YOU