

안녕하세요, 임성한 입니다.

✉ Email : seonghanim@naver.com

📁 Github : <https://github.com/seonghanim>

📝 Blog : <https://jungjin.oopy.io/>

☎ 01073325532

💻 Full Stack Engineer

꾸준히 배움의 자세를 가지는 개발자

새로운 기술을 탐구하고 연구하고 도입하는 개발자가 되고 싶습니다.

👤 경력

2023년 2월 ~ 재직중

클로버 스튜디오 기업 부설 연구소 근무

▼ Java Spring 을 활용한 드론 관제 플랫폼 서비스 구축(국가 사업 과제 중 MSA 구조 중 하나의 서비스)

- 하드웨어(드론 스테이션) 제어를 위한 하드웨어와 웹 서버 통신을 위한 프로토콜 정의 및 웹 서버 측 통신 구축(Netty 활용)

▼ 드론 비행 중 이상 상황을 감지하기 위한 웹 서비스 구축(국가 사업 과제 중 MSA 구조 중 하나의 서비스)

- 모바일과 통신을 위한 socketIO 활용
- 드론 데이터 전송을 위한 프로토콜 구축 및 통신 코드 작성(Netty UDP)활용

▼ 드론 FC 보드,전파 측정 장비,관제 플랫폼 간의 proxy 서버 구축

- 전파 측정 장비와 관제 서버간의 통신을 위한 프록시 서버 구축
- 전파 측정 장비와 프록시 서버간 통신을 위한 프로토콜 설계 및 구조 설계
- 전파 측정 장비에서의 명령을 마브링크로 치환하여 관제에 비행 중 명령 전송
- 커스텀 마브링크를 설계 및 개발하여 드론과 전파 측정 장비간의 통신 구축
 - 전파 측정 데이터를 관제 서버로 송신
 - 드론 GPS 데이터를 전파 측정 장비로 송신

▼ 도커 허브를 활용한 DevOps 업무 진행

- 도커 파일 작성, 허브에 push 및 실서버에서 pull 하여 배포 작업 진행
- 도커 마운트 활용하여 다수의 서비스들의 로그 관리

2022년 9월 ~ 2022.12.30

소프트젠 인턴 근무

- NextJS를 사용한 프론트 엔드 구축
- multer 라이브러리를 활용한 로컬 파일 업로드, 다운로드 게시판 작업
- NextJS api 를 활용한 백엔드 API 설계 및 작성
- JWT를 이용한 사용자 인증
- HTML, JS를 활용한 장애인 교육용 웹사이트 제작

2019년 10월 ~ 2021년 2월

삼성 전자 영국 법인 IT팀 근무

- 사내 Infra 운영지원
- SAP 운영지원
- IT 기술 지원



프로젝트

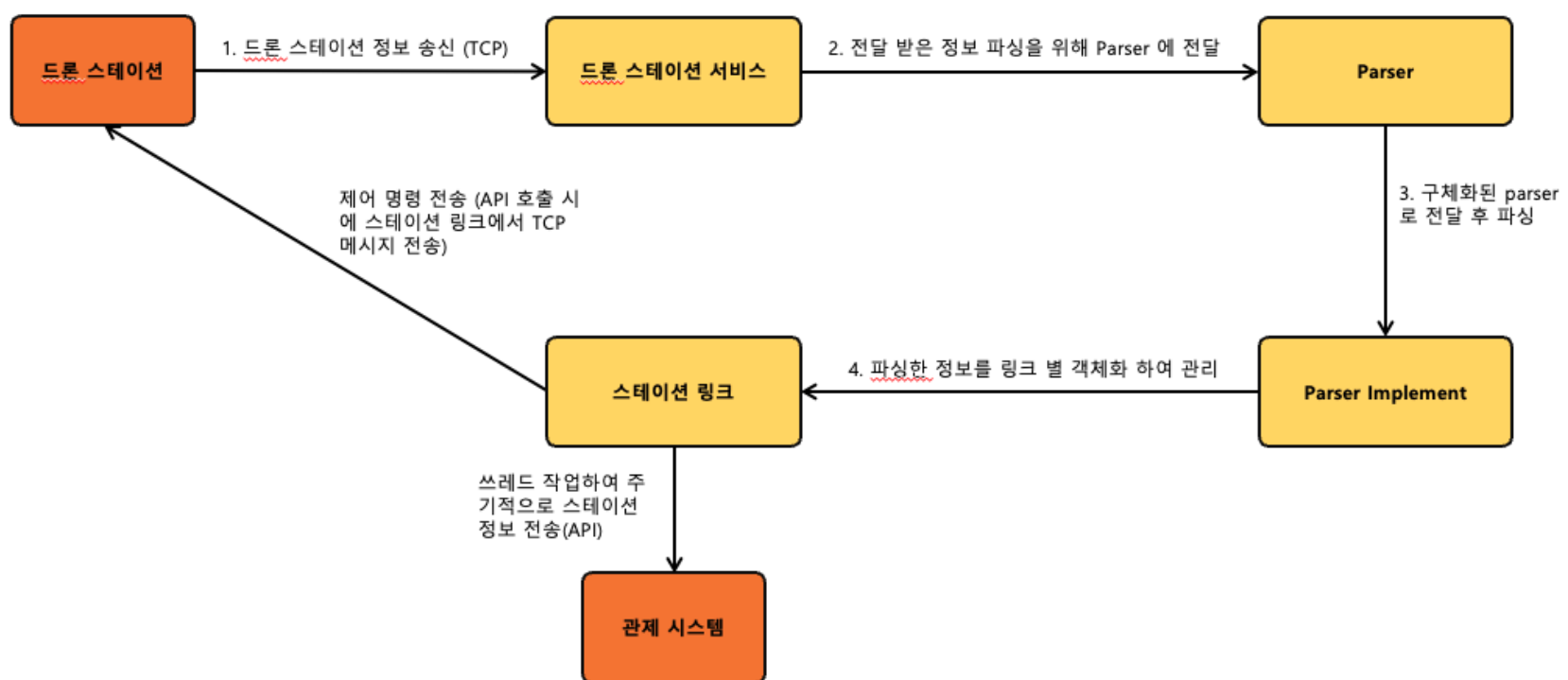
1. 드론 스테이션 서비스(MSA 프로젝트 중 하나의 서비스)

프로젝트 소개

드론 통합 관제를 위한 플랫폼의 MSA 구조의 서비스 중 하나로써, 드론 스테이션 제어 및 관제와의 통신을 위한 서비스(Spring boot) 실제 사업에 적용되어 운용 중이며 다중 연결 처리, 다중 연결 후 제어 명령 처리 구축

- 스테이션 제어 기능(스테이션 이착륙 준비 및 완료 행정, LED, 드론 배터리 탈부착 행정 등 제어)
 - 커스텀한 프로토콜을 사용하여 드론스테이션과 웹 서비스간의 통신 구축(Netty) 사용
 - 차후에 프로토콜이 다른 드론스테이션 연결 및 제어를 위한 드론스테이션 메시지 파서 추상화 구축
 - 다중 연결 및 제어를 위한 연결 링크 관리 및 객체화 구축
- 스테이션 정보 관제 전달 기능
 - timer 쓰레드를 활용하여 드론 스테이션 정보를 관제에 보내줌으로써 관제 시스템의 드론스테이션 정보 표출
- 스테이션 알람 처리 기능
 - 프로토콜에 16비트 hexa 데이터를 사용 및 파싱하여 드론스테이션의 알람 정보 파싱 및 관제 전송 기능 구축

서비스 동작 흐름 도식화



2. 비행 중 조건 검사 서비스(MSA 프로젝트 중 하나의 서비스)

프로젝트 소개

국가 과제 사업의 MSA 구조 중 하나의 서비스로서, 드론 임무(서비스) 비행 시작 시점(Takeoff) 시점 부터 작동하며, 드론 비행 중 필요한 조건들을 주기적으로 업데이트, 검사 하여 드론이 서비스 조건 내에서 비행할 수 있도록 하는 비행 조건 검사를 위한 서비스(Spring boot)

- 드론 비행을 위한 조건 상태 업데이트 기능(spring 내의 `@Scheduled` 어노테이션 사용하여 쓰레딩 처리)
 - 날씨 정보 업데이트
 - KP지수 업데이트
 - 드론 상태 업데이트(배터리, 기체 상태)

- d. 드론 스테이션 상태 업데이트
- e. 드론 비행 요청한 사용자 정보 업데이트(웹 소켓 으로 앱 사용자 위치 정보 업데이트)

2. 업데이트한 조건 상태 검사 기능(@Scheduled 쓰레딩 처리)

- a. 사용자가 공역내에 있는지 검사(사용자가 앱으로 구조 요청 시에 드론 비행 시작)
- b. 위 업데이트 된 조건들이 정해진 기준에 충족되지 않을 시에 드론 RTL(Return to Lunch) 처리

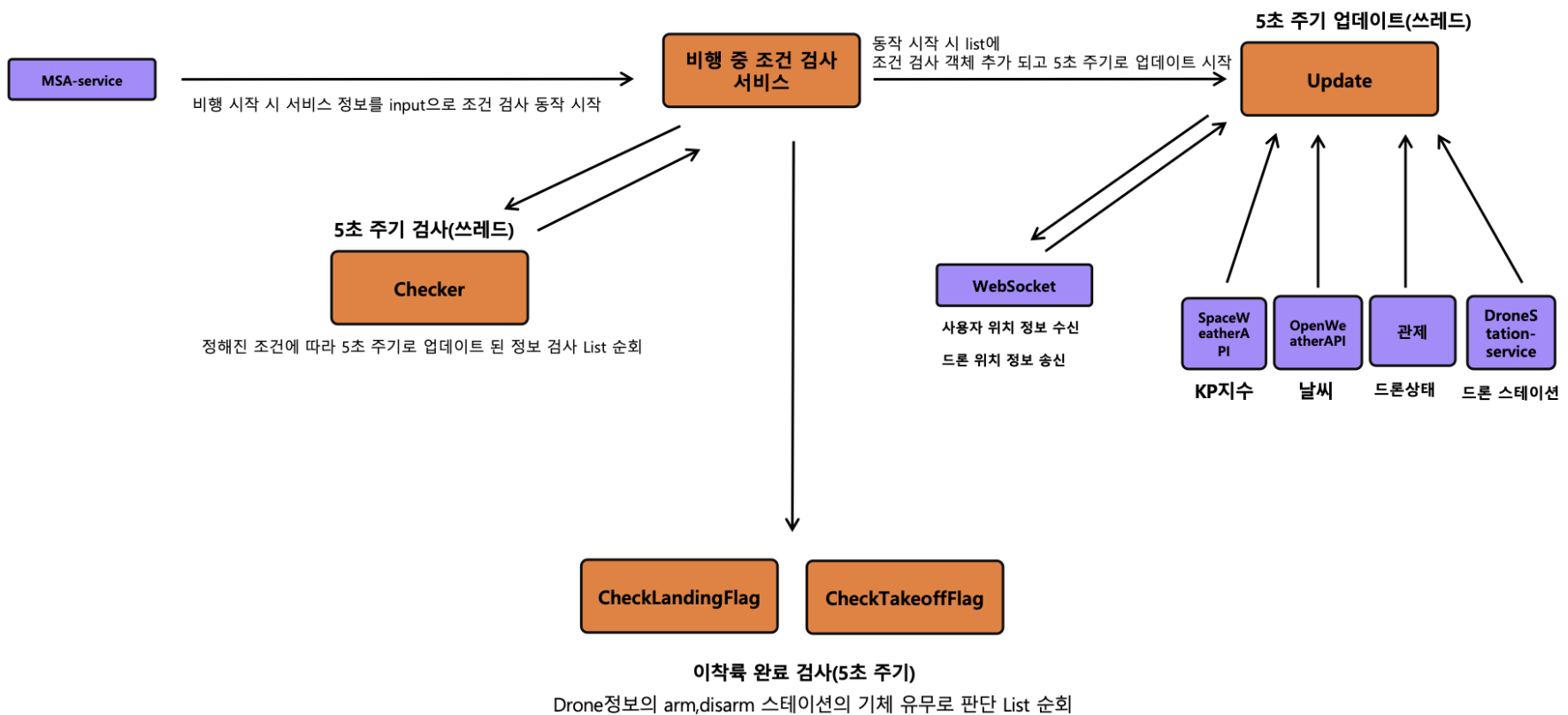
3. 업데이트한 조건들을 파싱하여 스테이션 제어

- a. 이착륙 완료 판단을 위한 드론의 arm, disarm 상태 / 드론스테이션 센서의 기체 유무 정보 파싱
- b. arming , 스테이션 기체 무 상태 일 때, 드론 이륙 완료 상태로 판단 후 30초후에 스테이션 닫기 행정
- c. disarming , 스테이션 기체 유 상태 일 때, 드론 착륙 완료 상태로 판단 후 30초후에 스테이션 열기 행정

4. 웹 소켓(socketIO)을 활용하여 어플 사용자의 위치 정보를 수신 및 관제 서버로 송신(관제 서버에 앱 사용자 위치 표출)

- a. 모바일 서버의 socketIO 서버를 구독하여 이벤트(앱 사용자 위치 변경)발생 시에 사용자 위치 정보를 수신할 수 있도록 구축
- b. 구축된 socketIO는 별도의 쓰레드로 동작, application-context에 저장된 객체를 업데이트 함으로써 쓰레드와 별도로 해당 (사용자 정보를 담고있는) 객체에 접근 가능

서비스 동작 흐름 도식화

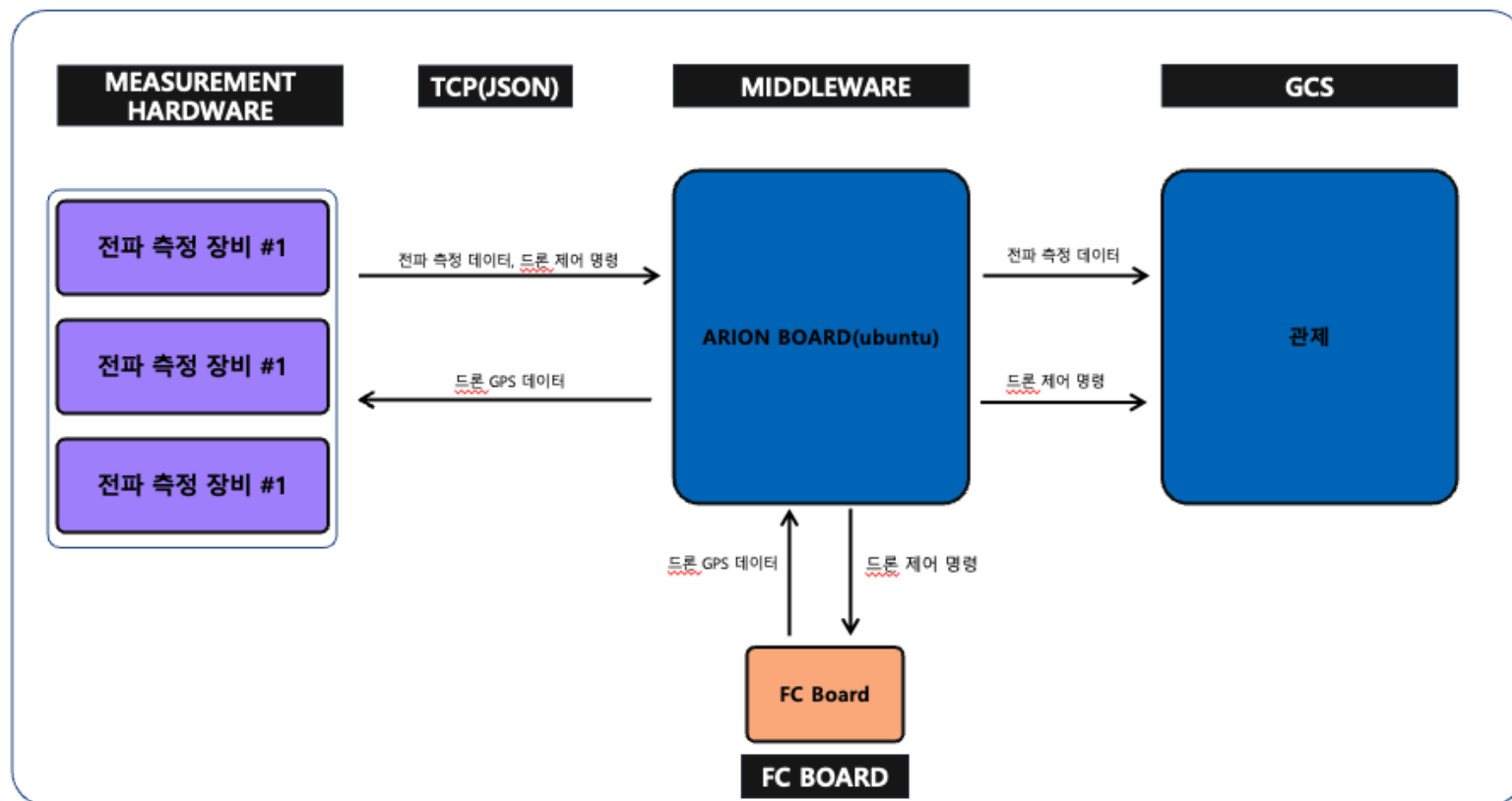


3. 전파 장비와 드론, 관제 통신을 위한 프록시 서버

프로젝트 소개

1. 전파 측정 장비를 드론에 장착 하여 비행, 비행 시작 시에 드론은 GPS 값을 전파 측정 장비는 전파데이터를 상호 송수신 함으로써 상공의 전파 데이터를 측정(파이썬)
2. 전파 측정 장비 다중 연결(1대의 드론에 통신사 별 전파 측정 장비 장착)을 위한 포트 별 연결 객체 링크 관리 구축
3. 전파 측정 장비와, 드론, 관제 통신을 위한 프록시 서버 구축(python, mavproxy, pymavlink)
 - a. 전파 장비와의 데이터(비행 명령, GPS 정보, 전파 측정 데이터)송수신을 처리하기 위한 프로토콜 정의 및 데이터 전송 흐름, 명령 흐름 설계

- b. 전파 측정 장비와 프록시 서버간의 ADB를 활용한 통신(serial to tcp) 구축
- c. 프록시 서버에서 파이썬을 활용하여 전파 데이터를 수신(adb통신) 및 관제로 송신(tcp,mavlink 통신) 구축
- d. FC 보드의 gps 값을 획득하여, 전파 측정 장비로 송신 구축



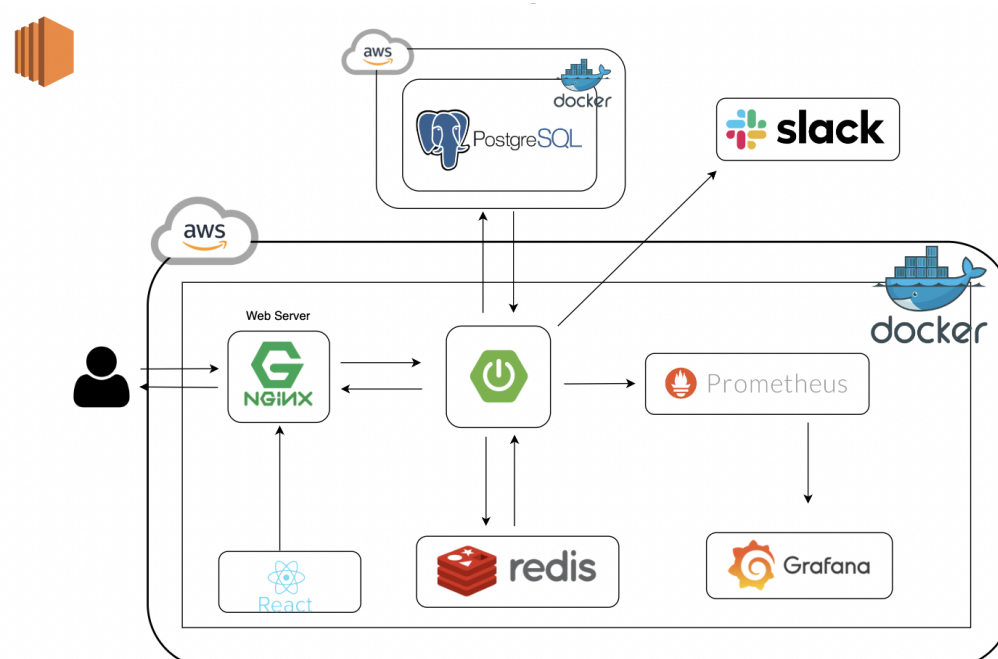
4. In For Plan

프로젝트 소개

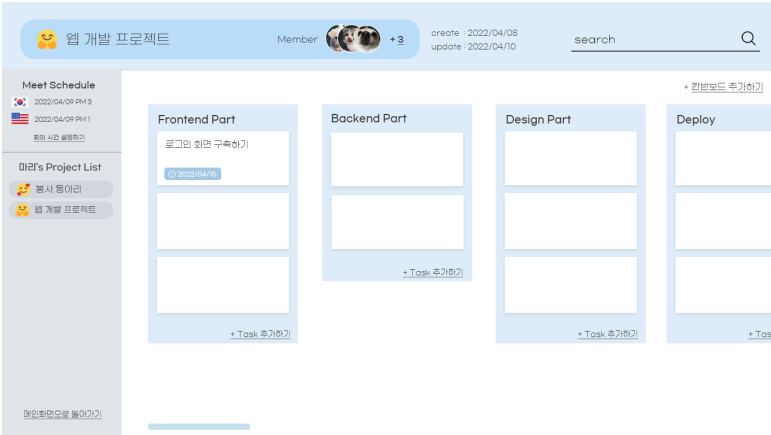
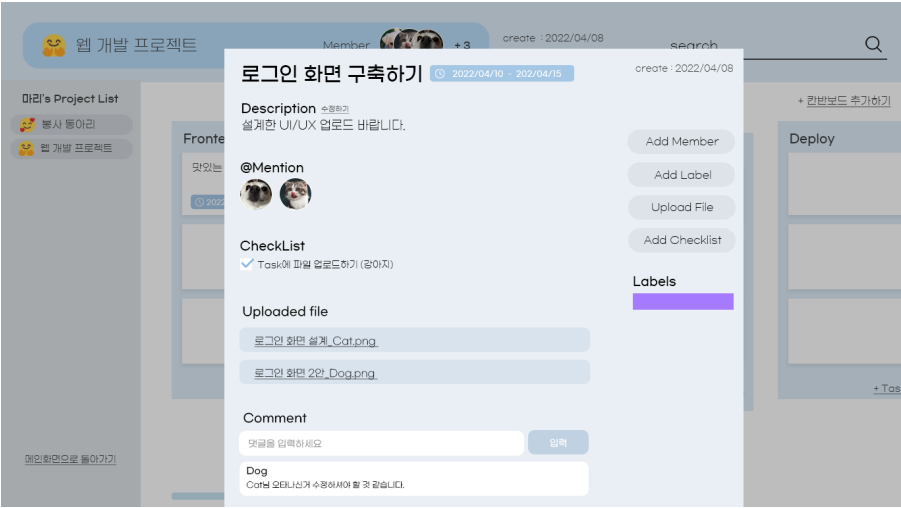
해외 거주자들과의 시간 약속을 편리하게 해주는 시간 동기화 기능이 추가된 일정 관리 웹 사이트

1. 구글 통합 로그인 기능
2. Docker를 사용하여 개발 환경 통일 및 compose를 활용하여 AWS에 배포
3. github action을 사용하여 CI/CD 활용하여 배포 자동화
4. 데이터 베이스 설계 시 다대다 관계의 단점을 발견하고, 이를 두 Entity 사이에 중간 테이블을 만듦으로써 해결
5. Prometheus, Grafana를 활용한 모니터링 시각화 구현

아키텍처 다이어그램



화면



깃허브링크

<https://github.com/team-g-techeer>

team-g-techeer

team-g-techeer has 4 repositories available. Follow their code on GitHub.

<https://github.com/team-g-techeer>

5. Oembed Test

프로젝트 소개

- Youtube, Twitter, vimeo 등의 provider 가 제공하는 Oembed 서비스를 활용하여 클라이언트가 원하는 웹 사이트를 입력하면 backed 서버를 거쳐, Oembed 정보를 보여주는 웹사이트
- Oembed 에는 title, author_name, author_url, type, height, width, version, provider_name, provider_url, thumbnail 등의 종류가 있습니다.
- backend 로는 Springboot 를 사용하였고, frontend 로는 mustache를 사용하였습니다.

화면

- 검색 시 화면

Oembed 검색

ex) <https://www.youtube.com/watch?v=qsshPV4unDU&t=1s>

검색

title	대한민국 구호대 118명 튀르키예 도착...활동 개시 / SBS
author_name	SBS 뉴스
author_url	https://www.youtube.com/@sbsnews8
type	video
height	113
width	200
version	1.0
provider_name	YouTube
provider_url	https://www.youtube.com/
thumbnail_height	360
thumbnail_width	480
thumbnail_url	https://i.ytimg.com/vi/MfiULf0Omw8/hqdefault.jpg



html

```
<iframe width="200" height="113" src="https://www.youtube.com/embed/MfiULf0Omw8?feature=oembed" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen title="대한민국 구호대 118명 튀르키예 도착...활동 개시 / SBS"></iframe>
```



Github 링크

<https://github.com/seonghanIm/OembedTestWebsite>

배운점

- 서버 템플릿 엔진을 처음 사용해 보는 계기가 되었습니다. 기존의 스프링 부트 + 클라이언트 템플릿 엔진에서의 Controller 단에서 객체를 반환하는 방식에 익숙해 저서 처음 코드 작성 시 객체를 반환하도록 구현하였었습니다.하지만 서버 템플릿 엔진은 문자열을 반환하는 방식이었기 때문에 다시 코드를 작성해보면서 이런 방식도 있다는것을 깨닫는 계기가 되었습니다.
- AOP의 개념을 이해하게되었습니다. Controller 단에서의 Exception 처리를 위해 @ControllerAdvice를 사용하며 이론으로만 알고 있던 횡단 관심사의 분리 개념을 알게되었습니다.

6. Log Frequency Analyzer

프로젝트 소개

- 로그 빈도수를 체크하기 위한 pojo 기반 프로그램
- [200][<http://apis.daum.net/search/knowledge?apikey=23jf&q=daum>][IE][2012-06-10 08:00:00] 이와 같은 형식의 로그들의 빈도수를 분석.
- pojo 기반으로 작성되었으며, 최다 호출 API key, 상위 3개의 api ServiceID 및 요청 수, 웹 브라우저 별 사용 비율을 확인.

화면

- Input 파일

```
[200][http://apis.daum.net/search/knowledge?apikey=23jf&q=daum][IE][2012-06-10 08:00:00]
[200][http://apis.daum.net/search/knowledge?apikey=dcj8&q=daum][IE][2012-06-10 08:00:01]
[200][http://apis.daum.net/search/blog?apikey=wejf&q=daum][IE][2012-06-10 08:00:02]
[200][http://apis.daum.net/search/knowledge?apikey=e3ea&q=daum][IE][2012-06-10 08:00:03]
[200][http://apis.daum.net/search/vclip?apikey=2jdc&q=daum][IE][2012-06-10 08:00:04]
[200][http://apis.daum.net/search/vclip?apikey=wejf&q=daum][IE][2012-06-10 08:00:05]
[200][http://apis.daum.net/search/image?apikey=fwji&q=daum][IE][2012-06-10 08:00:06]
[200][http://apis.daum.net/search/image?apikey=anw1&q=daum][IE][2012-06-10 08:00:07]
[200][http://apis.daum.net/search/image?apikey=2jdc&q=daum][IE][2012-06-10 08:00:08]
```

- Output 파일

```
최다 호출 API KEY
e3ea

상위 3개의 API Service ID와 각각의 요청수
knowledge: 809
news: 803
blog: 799

웹 브라우저별 사용 비율
IE: 81%
Firefox: 6%
Opera: 2%
Chrome: 2%
Safari: 1%
```

Github Link

<https://github.com/seonghanIm/LogFrequencyAnalyzer>

배운점

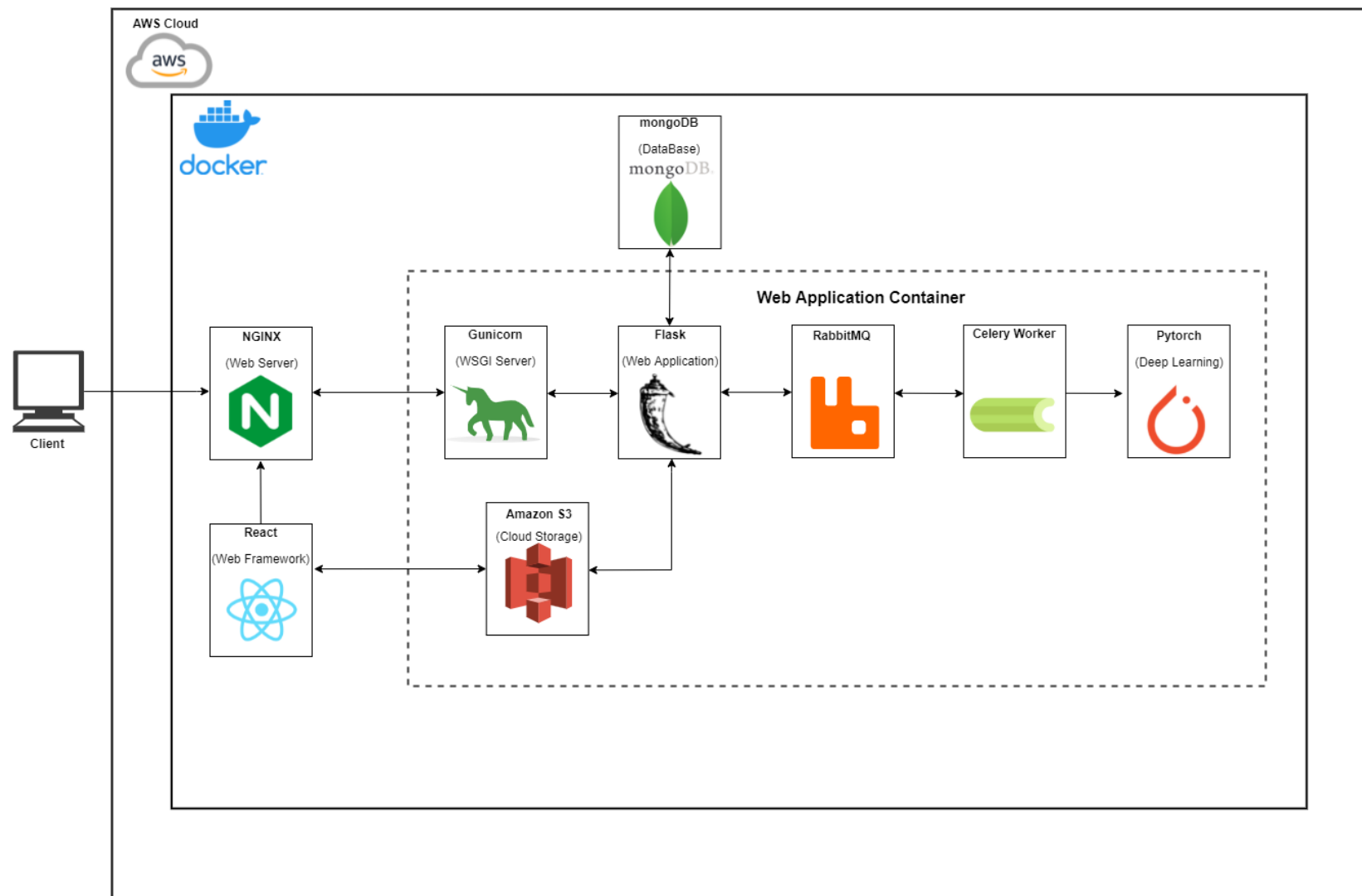
- 자바8 의 기능인 Stream 및 Interface의 static 메소드 구현 등 을 사용하려고 노력하였습니다. 보다 가독성 높고 효율적인 코드를 작성할 수 있었습니다.
- 빌더 패턴, 팩토리 메소드 패턴등을 활용해 보았습니다. 디자인 패턴을 의도적으로 사용하기 위해서는 코드가 더 복잡해지고 가독성이 떨어지는 경향이 있는것 같습니다. 코드를 작성 후 더 효율적인 방법이 없나 고민하는 과정에서 프로그램에 맞는 디자인 패턴을 사용하는 것이 더 효율적인 것 같다고 느꼈습니다.

7. Pictoon

프로젝트 소개

- AI를 활용하여 두 창작물(이미지)을 결합하여 새로운 창작물을 만들 수 있는 웹 사이트
 1. Style_transfer 라는 오픈소스를 활용하여 이미지의 결합, 인물 + 배경 이미지 조합으로 다양한 분위기의 인물사진을 얻을 수 있음
 2. Docker를 활용하여 팀원들 간의 개발 환경 통일 및 compose를 통한 배포 간편화
 3. S3를 사용하여 이미지 저장 및 사용
 4. 비동기 방식으로 API를 처리하기 위해 RabbitMQ 와 Celery worker, Redis 사용

아키텍처 다이어그램



배운점

1. 동기 방식으로 동작하는 플라스크가 시간이 오래걸리는 이미지 변환 작업을 기다림 → Nginx 타임 아웃 문제 발생 → 비동기 처리를 위한 celery 를 사용 메시지 브로커로 RabbitMQ 사용.

깃허브 링크

https://github.com/Pictoon-siliconvalley-Bteam/Pictoon-project_Main

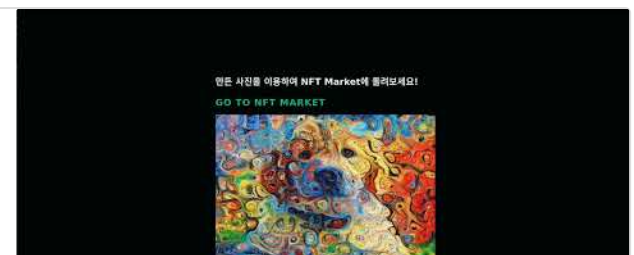
https://github.com/Pictoon-siliconvalley-Bteam/Pictoon-project_Main

2021.12~2022.1

동작 화면 Youtube 링크

Pictoon 동작 영상

<https://youtu.be/7SNFsBJ7SG0>



Education

한국공학대학교 컴퓨터 공학과 졸업

2015.03 ~ 2023.02

