**# 1**

안녕하십니까? AIker 팀원 윤가영입니다.

지금부터 AI를 이용한 도커 컨테이너 자동분류기에 대한 개발계획서 발표를 시작하겠습니다.

**# 2**

목차는 총 5개로 이루어져 있습니다.

**# 3**

첫번째 AIker란?

**# 4**

현재 MSA의 특징상 전체 서비스가 커지면, 복잡도가 기하급수적으로 늘어날 수 있다는 큰 단점을 가지고 있습니다.

특히 성능, 테스트/트랜잭션, 데이터관리 이렇게 크게 3가지 측면으로 살펴보겠습니다.

첫번째 성능은 서비스 간 호출 시 API를 사용하므로 통신 비용, Latency가 사용하는 만큼 늘어나게 됩니다.

두번째 테스트/트랜잭션은 서비스가 분리되어 있어 테스트와 트랜잭션의 복잡도가 증가하여 많은 자원을 필요로 합니다.

세번째 데이터 관리는 데이터가 여러 서비스에 걸쳐 분산되어 한 번에 조회하기 어렵고 데이터의 정합성 또한 관리하기 어렵습니다.

**# 5**

따라서 저희 팀은

**# 6**

최대단점인 복잡도에 초점을 잡아

컨테이너의 유지보수를 용이하게 하고 효율성을 확보하는 서비스를 개발하고자 합니다.

방금 말씀드렸다시피

서비스의 규모가 커질수록 배포가 많아질 수밖에 없고,

결국 컨테이너의 복잡도가 높아진다는 단점이 있으므로

이를 DNN기반의 자동분류기를 통해 컨테이너를 효율적으로 관리하고자 합니다.

**# 7**

따라서 AIker는 AI와 DOCKER의 합성어로서

DNN기반, 복잡한 컨테이너의 자동 분류 및 배포를 해주는 웹 서비스를 개발하는 것이 저희 AIker의 목표입니다.

**# 8**

두번째로 분류기에 관해 설명하겠습니다.

총 두 가지의 분류기를 준비해봤는데요

**# 9**

첫번째는 DNN단독사용입니다.

저희는 Container data가 JSON형식으로 2차원임을 확인했기 때문에

전처리 후 바로 학습하고자 했습니다.

**# 10**

두번째는 DNN에 AutoEncoder를 결합한 모델입니다.

Auto Encoder는 비지도 학습의 일종으로서, 특징추출의 기능으로 쓰입니다.

따라서 데이터를 간단한 전처리 후에 특징 추출한 input의 값을 다시 DNN으로 학습시킨다면 훨씬 더 깔끔한 분류기 성능이 나올 것이라 생각했습니다.

**# 11**

결국 이 분류기를 통한 학습에 관한 가장 중요한 부분은 전처리와 모델링 설계라 생각합니다.

**# 12**

세번째 로직입니다.

**# 13**

웹서버의 설계는 POST 형식으로 컨테이너의 추가 조회 삭제로 설계했으며 NodeJS로 개발하고자 합니다.

**# 14**

서버는 컨테이너 추가요청 시에만 작동하고, AI 기반 분류기를 통해 컨테이너 분류작업을 할 예정입니다.

**# 15**

분류기는 Python으로 AI 기반 분류작업을 수행할 예정이며, 분류된 결과는 Web server로 전송됩니다.

**# 16**

분류된 컨테이너의 정보를 데이터베이스에 저장하는 구조로 설계했고, Mysql을 사용합니다.

따라서 전체적인 로직은 다음과 같습니다.

**# 17**

웹서버로부터 컨테이너 create 요청이 발생합니다.

그럼 도커 컨테이너를 생성하여 웹서버로 컨테이너의 정보를 반환합니다.

반환된 메타데이터를 분류기에 보내면, 분류기는 전처리 후 분류작업을 수행하여 결과를 반환합니다.

웹서버는 반환된 결과를 DB에 저장합니다.

**# 18 , # 19**

컨테이너 조회와 삭제 같은 경우는, 바로 데이터베이스에서 불러오도록 하여 기존에 설계했던 서버의 부담을 덜어주었습니다.

**# 20**

네번째 UI입니다.

**# 21**

현재 프로토타입으로 다음과 같은 형식으로 만들 예정이나, 웹페이지 디자인은 바뀔 수 있다는 점을 말씀드립니다.

**# 22**

이런 식으로 컨테이너가 어떻게 분류되는지 인덱스를 부여하여 리스트를 구현할 예정이며, 컨테이너가 어떤 식으로 구성되어 있는지 상세하게 설명할 수 있는 웹페이지를 구축할 것입니다.

**# 23**

다음으로 개발 일정입니다.

**# 24**

현재 전체적인 개발일정은 다음과 같으나, 세부적으로 바뀔 수 있습니다.

**# 25**

감사합니다.

지금까지 AIker였습니다.