|  |  |  |  |
| --- | --- | --- | --- |
| **종합설계 프로젝트 회의록** | | | |
| 팀명 | 1팀 | | |
| 회의주제 | Kubernetes를 활용한 Hyper Parameter Tuning과 ML model 배포를 위한 MLOps플랫폼 개발 | | |
| 일시 | ○ 일 자: 2022. 9. 26. (월)    ○ 시간: 15:00 ~ 16:30 | | |
| 장소 | Google Meet | | |
| 참석자 | 박보경, 박종민, 신시온, 이예림, 이동욱 책임연구원 | 불참자 | 김석희 |
| 회의내용 | 1. 쿠버네티스, 컨테이너, 도커, MLOps에 대한 설명  * 컨테이너는 프로세스이다   OS에서 돌아가기 때문에 프로세스이다.  도커는 특수한 형태의 프로세스이다.   * 쿠버네티스는 컨테이너 오케스트레이터이다.   쿠버네티스는 서버 또는 호스트 머신이라고 하는 것을 여러대의 서버, 여러대의 머신, 여러대의 호스트로 클러스터로 묶어서 실행각각 서버 10개를 클러스터로 묶어서 띄운다.   * 쿠버네티스를 쓰는 이유   쿠버네티스가 없으면 쿠버네티스가 할 수 있는 작업 할수있긴 하지만, 사람이 하나하나 관리하기 힘들다  ->이를 쿠버네티스가 관리하면 편하다   * 컨테이너를 쓰는 이유   컨테이너는 하나의 독립된 호스트  내가 개발했던 환경이랑 테스팅 환경 또는 다른 개발자의 환경이 다를 수 있는데, 컨테이너 기반 프로젝트는 이러한 접근성 문제를 해결할 수 있다.  ->컨테이너를 쓰면 훨씬 편하다(다른 개발자들과 협업을 할 때, 혼자 개발할때도 편함-동일한 소스코드에 컨테이너 환경만 바꾸면 훨씬 수월해진다)   * 컨테이너를 직접 빌드하는 방법을 익히고 직접 해봐라   도커로 연습하는걸 권장   * 쿠버네티스 외의 컨테이너 오케스트레이터 * 도커 컴포즈: 컨테이너 오케스트레이션을 해주지만 여러 호스트를 해주지는 않음. 하나의 서버에 띄워야 할 컨테이너가 여러 개일때 사용 * 도커 스윔: 쿠버네티스와 겹친다 * 도커 스윔을 쓰지 않고 쿠버네티스만 쓰는 이유   사람들이 쿠버네티스를 많이 쓰기 시작해서  구글이 내부에서 10년동안 썼던 오케스트레이터  노하우가 쌓이고 경험들이 잘 녹아 있어서 좋음  (mlops, 우버의 미켈란젤로를 검색해보면 알 수 있음)   * mlops가 필요한 이유   단계를 자동화하면 편리해서 필요  mlops 개념은 2015년 구글에서 논문을 발표  예전에는 사람들이 Ml코드 짜는데에 집중했지만, 실제 코드짜는건 전체 시간 비중의 5%도 차지하지 않음  서비스를 하기 위한 과정에 시간을 대부분 쓰더라 (ML을 붙이고, 유지보수를 하는게 중요하다)   * 취업을 하면 AI와 관련된 과정을 하게 될거임   Al, ML에 대한 기본적인 지식은 알고 있어야 함   * 개발자가 사용하는 ML서비스는 API형태로 호출하게 됨   어딘가에선 AI 서비스에 대한 API를 제공해야함  AI 서비스 REST API를 접근하게 오픈해야 하는데 그것을 MLOps 시스템이라 함   1. 프로젝트 설명   ml endpoint를 서빙할 수 있는 서비스를 만드는 것이다  1) 하이퍼 파라미터 튜닝 시스템  기본 모델(코드가 될 수도, 모델 자체가 될 수도)이 하나 들어올텐데, 그 모델이 하이퍼 파라미터 세팅에 따라 세팅이 바뀌게 됨.  2) 서빙  Input: ML 모델(파일)을 입력, 파일에 대한 정보를 읽음  Output: REST API endpoint열어서 해당하는 모델을 출력   1. 이 프로젝트는 단순히 한번 해보겠다가 아니라 현업에서 기본적으로 하는 것이라서 열심히 해야 함   (쿠버네티스에 가장 많이 집중해라)   1. 프로젝트 관련  * 협업 툴   위키, 이슈 등록과 해결과 같은 협업 툴들을 모두 잘 사용해서 프로젝트를 수행해봐라  해야 함   * 쿠버네티스가 러닝 커브가 높을 수 있으므로, 프로젝트 한달 간 쿠버네티스 관련해서 공부를 빠르게 하는게 좋다 * 논문 제출하면 좋음  1. 웹 페이지 전체적인 흐름 2. 시스템에 접속   입력을 받음(학습할 수 있는 코드 또는 모델)  입력 받는 란에 blank로 ‘하이퍼파라미터 ~를 ~로 해줘’를 입력받는 웹페이지 필요   1. 하이퍼 파라미터 튜닝 시작 버튼   버튼을 누르면 내부에서 쿠버네티스가 매우 많이 만들어서돌리고 나온 모델들에 대한 결과값을 화면에 띄워줌(가장 좋은 10개 or 1개)   1. 결과화면   ‘요청하신 모델을 rest api ~~주소로 보냈습니다’ 출력   1. 쿠버네티스 관련 자격증(3가지)   쿠버네티스 관련 자격증을 따면 좋음  회사도 쿠버네티스 자격증을 가진 엔지니어들이 많으면 인증을 받을 수 있어서 회사도 좋아함   * Cka * Ckad * Cks   순서: cka->ckad->cks   1. 전체적 목표: 프로젝트 전반적인 이해, 기술 익히기 2. 진행 방식   각자 해결->팀원간 해결->팀장이 모아서 질문(이메일 or 미팅 요청)  기본적으로 한달에 한 번 미팅 예정   1. 멘토님께 질문 및 답변 2. 만들 서비스 한가지를 정하고 그 서비스를 쿠버네티스를 통해 RestAPI로 사이트 개발자가 그. 사람이 만든 코드를 hyperParameter값으로 제공하는 것인가?  * 5번을 참고  1. 튜닝과 서빙 기능을 합쳐서 사용자가 모델과 hyperparameter세팅값을 튜닝해서 restAPI형태로 제공하는 것인가?  * 5번을 참고  1. 결과를 얻은 사람은 그걸로 서비스를 만드는 것인가?  * 사용자에 따라 다름   개발자라면, 그걸로 서비스를 만들고, 일반 사용자의 경우는 아님   1. ‘pod마다 hyper parameter조합을 한가지씩 돌리게 된다’라고 이해를 했는데, 세부적으로 어떻게 돌리는지 모르겠다  * 학습을 어떻게 시키냐? 함수콜과 똑같음 * 파드를 어떻게 구현? 여러 방법들이 있음. 리눅스 환경변수 형태로 전달. 쿠버네티스 리소스를 활용해 전달하는 방식(쿠버네티스 공부가 진행되어야 할 수 있음)  1. 쿠버네티스 엔진과 백엔드는 어떻게 진행?  * 백엔드와 쿠버네티스 엔진을 합쳐도 됨   나뉘게 된다면 엔진쪽은 플라스크, 벡엔드는 자바스프링  구조는 정해져있는게 아니라 개발의 수월함과 단순성을 위해서는 파이썬으로 통일하는게 좋을 수 있다.  파이썬 장고, 플라스크 등 써서 사용하는게 개발에는 수월할 것임 | | |
| 개인별  역할 분담 | 1. 공통  * 작성했던 회의록 pdf파일도 함께 업로드 * AWS, 쿠버네티스 공부  1. 김석희, 박보경  * 프론트 구현 공부  1. 신시온, 이예림  * 백엔드 구현 공부  1. 박종민  * ML 모델 하이퍼 파라미터 튜닝 공부 | | |
| 진행할  업무 | 1. 노트북 대여 2. AWS 강의를 각자 모두 듣기(~9/29(목) 조별 회의 시간 전까지)   목요일까지 각자 AWS에서 배포, 과정, 쿠버네티스에 대해 정확하게 숙지, 공부해오기(정리를 해서 어떻게 하자고 말할 수 있을 정도)  전체적인 프로세스 슈도코드 느낌으로 다 알고 와라  [강의 링크]  <https://www.youtube.com/watch?v=kb6s0Tmp2CA&t=4562s>  <https://www.youtube.com/watch?v=XTyuK_gaN8Q>   1. 클러스터링, CI/CD, 젠킨스, virtual box, virtual machine 학습 2. 회의록 pdf 파일 업로드 | | |
| 회의 사진 |  | | |
| 다음 회의 때 논의해야할 사항 | 1. 쿠버네티스 환경설정 및 실습환경 결정(실습실, 집 등) 2. 세부 기능 정하기   (배포기능, pod 개수, pod또는 node의 cpu 및 메모리 할당, 로그인 기능 유무, argument 처리 등)   1. 웹 디자인(구체적인 디자인 완성하기) | | |