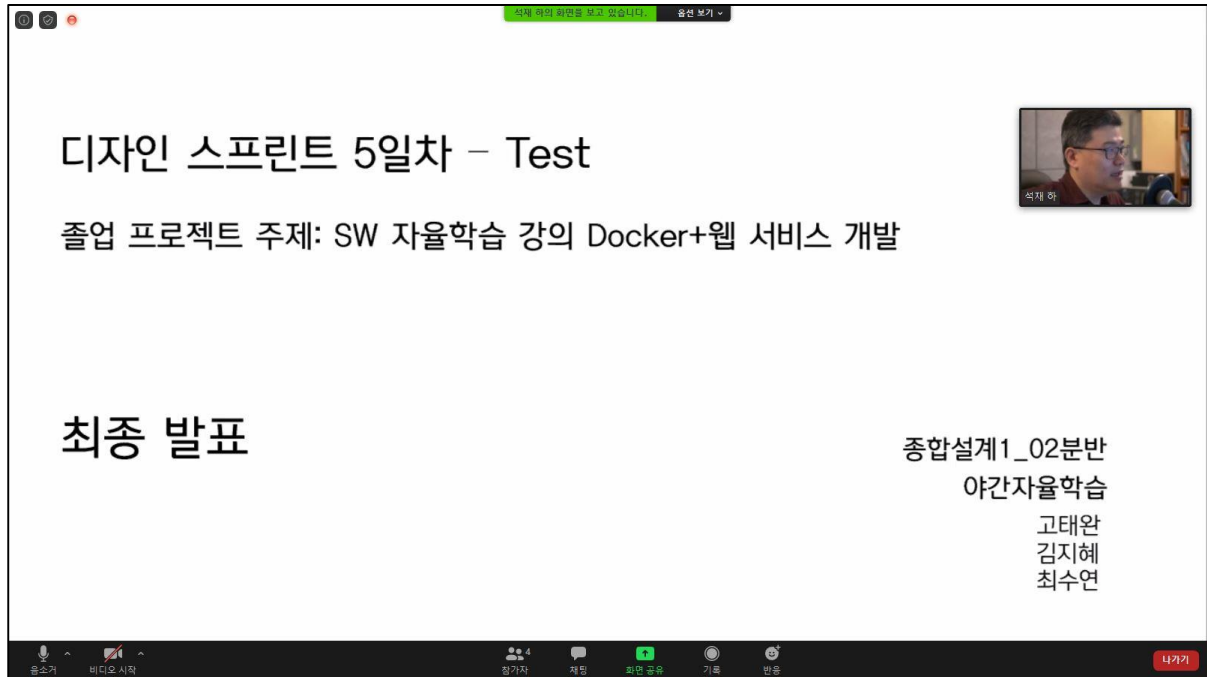


종합설계1_02분반 멘토링보고서

고태완, 김지혜, 최수연

화상회의

5월 22일 17시 하석재 멘토님과 온라인 화상 회의 진행



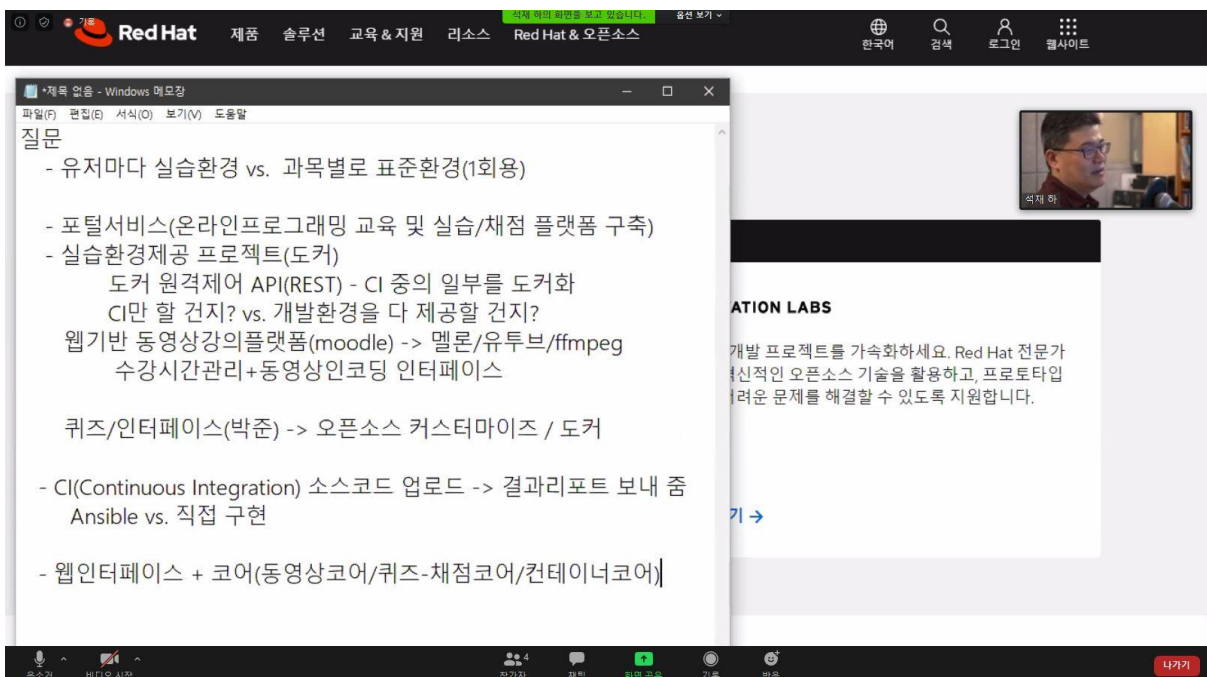
디자인 스프린트 5일차 - Test

졸업 프로젝트 주제: SW 자율학습 강의 Docker+웹 서비스 개발

최종 발표

종합설계1_02분반
야간자율학습
고태완
김지혜
최수연

야간자율학습 팀의 발표자료와 문서를 공유하고 이에 대한 피드백을 받음.



Red Hat

제품 솔루션 교육 & 지원 리소스 Red Hat & 오픈소스

질문

- 유저마다 실습환경 vs. 과목별로 표준환경(1회용)
- 포털서비스(온라인프로그래밍 교육 및 실습/채점 플랫폼 구축)
- 실습환경제공 프로젝트(도커)
 - 도커 원격제어 API(REST) - CI 중의 일부를 도커화
 - CI만 할 건지? vs. 개발환경을 다 제공할 건지?
- 웹기반 동영상강의플랫폼(moodle) -> 멜론/유튜브/ffmpeg
- 수강시간관리+동영상인코딩 인터페이스
- 퀴즈/인터페이스(박준) -> 오픈소스 커스터마이즈 / 도커
- CI(Continuous Integration) 소스코드 업로드 -> 결과리포트 보내 줌
- Ansible vs. 직접 구현
- 웹인터페이스 + 코어(동영상코어/퀴즈-채점코어/컨테이너코어)

프로젝트에 대한 대표님의 피드백

회의록

- 유저마다 실습환경 vs. 과목별로 표준환경(1회용)
- 포털서비스(온라인프로그래밍 교육 및 실습/채점 플랫폼 구축)
- 실습환경제공 프로젝트(도커)

도커 원격제어 REST API - CI 중의 일부를 도커화

CI만 할 건지? vs. 개발환경을 다 제공할 건지?

웹기반 동영상강의플랫폼(moodle) -> 멜론/유튜브/ffmpeg

수강시간관리+동영상인코딩 인터페이스

퀴즈/인터페이스(박준) -> 오픈소스 커스터마이즈 / 도커

- CI(Continuous Integration) 소스코드 업로드 -> 결과리포트 보내 줌

Ansible vs. Docker REST API

- 웹인터페이스 + 코어(동영상코어/퀴즈-채점코어/컨테이너코어)

인터페이스만 정의한 상태로 보인다

코어를 어떻게 구축할 건지(오픈소스기반 vs. 자체구축)

객체지향기술을 사용한다면 UML(필수)

cf. Agile을 하면 TDD(Test case를 작성)

쿠버네티스 - 도커로 여러 대를 사용한 서비스를 운영(Orchestration)할 때 사용하는 기술

1. VMWare Workstation(standalone)

2. VMWare ESXi + vCenter(여러 대 관리-> orchestration) -> vSphere

Docker + Kubernetes(orchestration)

vSphere(기존) + Kubernetes(도입) - 같이 어떻게 사용할 수 있나?

orchestration(DB이중화/WAS)

멘토링을 통해 도출된 결과

- 현재 메인 시스템이 3개로 규모가 너무 큼 (웹기반 동영상강의플랫폼, 퀴즈-채점, 도커를 이용한 실습 환경)
- 도커를 이용한 실습 환경에 초점을 맞추기로 결정(선택과 집중)
- 이미 시장에서 잘 사용되고 있는 이론 강의에 대한 서비스 및 퀴즈는 최종 프로젝트에서 삭제
- 실습 환경을 이미지로 저장하여 수강생들 개개인에게 하나의 컨테이너로 제공
 - => OS에 대한 접근이 필요한 실습에서 수강생은 root 권한으로 자유롭게 실습 진행 가능, 단 1회용 컨테이너, 실습이 끝난 후에 삭제됨
 - => Shell과 Code Editor를 웹페이지에서 제공.
 - 단 Shell을 사용하지 않아도 되는 경우(OS에 접근하지 않아도 되는 경우) Shell을 제공하지 않음
 - Code Editor를 사용할 경우에 채점을 하지 않고 컴파일 된 결과만 보여줌.
 - => 기본적으로 사용자가 업로드한 실습 강의에 대해서 저장된 실습의 난이도를 이용해 사용자에게 수준별 학습을 제공할 수 있도록 Curation한다.
 - => 특정 강사가 특정 사용자들에 대해서만 실습을 가능하게 할 것을 원하는 경우, 이 실습은 private 상태가 되며 특정 강사가 저장한 환경으로 사용 권한을 가진 수강생들에 한해서 해당 환경으로 바로 실습을 진행 할 수 있도록 한다.
- Docker와 Kubernetes를 이용한 Orchestration 방식의 유연한 서버 운영
- CI(Continuous Integration) 기반의 소스 코드 에디터 사용. 소스코드를 올리면 빌드 후 에러 포함한 결과 리포트를 사용자에게 보여줌.

Youtube Link : <https://youtu.be/ZGloJ7QCm-s>

Github Link : <https://github.com/Night-Self-Study/the-design-sprint>