

천방지축 + 우당탕탕 K8s 서비스 구현기(**with AWS, GitOps**):

쿠버네티스를 이용한
프로그래밍 과제 제출 환경 통합 플랫폼 개발

18기 엔지니어링 금나연

시작하기 전에...

Table of Contents

01

배경 및 의의

02

아키텍처

03

구현

04

한계점

01

배경 및 의의

주제 선정 배경 및 목표

이 프로젝트는

1학년 1학기 전공필수 수업 때부터 구상되었습니다...

(부제: 내가 빵점이라니?!)

주제 선정 배경 및 목표

조교님 제 점수가 잘못된 것 같아요..



주제 선정 배경 및 목표

조교님 제 점수가 잘못된 것 같아요..



저런... 연구실로 와보시겠어요?

주제 선정 배경 및 목표

조교님 제 점수가 잘못된 것 같아요..



저런... 연구실로 와보시겠어요?

...

주제 선정 배경 및 목표

조교님 제 점수가 잘못된 것 같아요..



저런... 연구실로 와보시겠어요?

...

아... 진짜 안 되네요..? 죄송합니다...



주제 선정 배경 및 목표

조교님 제 점수가 잘못된 것 같아요..



내 노트북에서 잘 실행되던 파일이 왜 조교님 환경에서 에러가

나는 걸까?

아... 진짜 안 되네요..? 죄송합니다...



주제 선정 배경 및 목표

문제 정의

- 개발 환경의 불일치로 인해 교수의 환경에서 학생의 의도와 다른 결과가 보여질 수 있음
- 교수가 모든 코드를 열어보고 실행시키고, 결과를 확인하기 번거로움

기존 방법의 한계

- 모든 학생에게 서버 또는 가상 환경을 배포하기에는 많은 비용과 관리를 위한 노력이 필요
- 하나의 거대한 리눅스 서버를 이용하여 제출자 수에 맞는 계정을 생성하기에는 권한, 보안, 설치 프로그램 등 제출자들 사이에서 충돌 문제 혹은 용량과 관련된 문제가 발생할 가능성이 높음

목표

- 학생 별로 독립적인 컨테이너(환경)를 할당하고, 교수는 이러한 리소스들을 손쉽게 관리하고 열람할 수 있는 플랫폼을 개발하고자 함

활용 분야

- B2B
 - 코딩 교육을 진행하는 플랫폼에서 사용 가능

주제 선정 배경 및 목표

문제 정의

- 개발 환경의 불일치로 인해 교수의 환경에서 학생의 의도와 다른 결과가 보여질 수 있음
- 교수가 모든 코드를 열어보고 실행시키고, 결과를 확인하기 번거로움

기존 방법의 한계

- 모든 학생에게 서버 또는 가상 환경을 배포하기에는 많은 비용과 관리를 위한 필요
- 하나의 거대한 리눅스 서버를 이용하여 제출자 수에 맞는 계정을 생성하기에는 권한, 보안, 설치 프로그램 등 제출자들 사이에서 충돌 문제 혹은 용량과 관련된 문제가 발생할 가능성이 높음

목표

- 학생 별로 독립적인 컨테이너(환경)를 할당하고, 교수는 이러한 리소스들을 손쉽게 관리하고 열람할 수 있는 플랫폼을 개발하고자 함

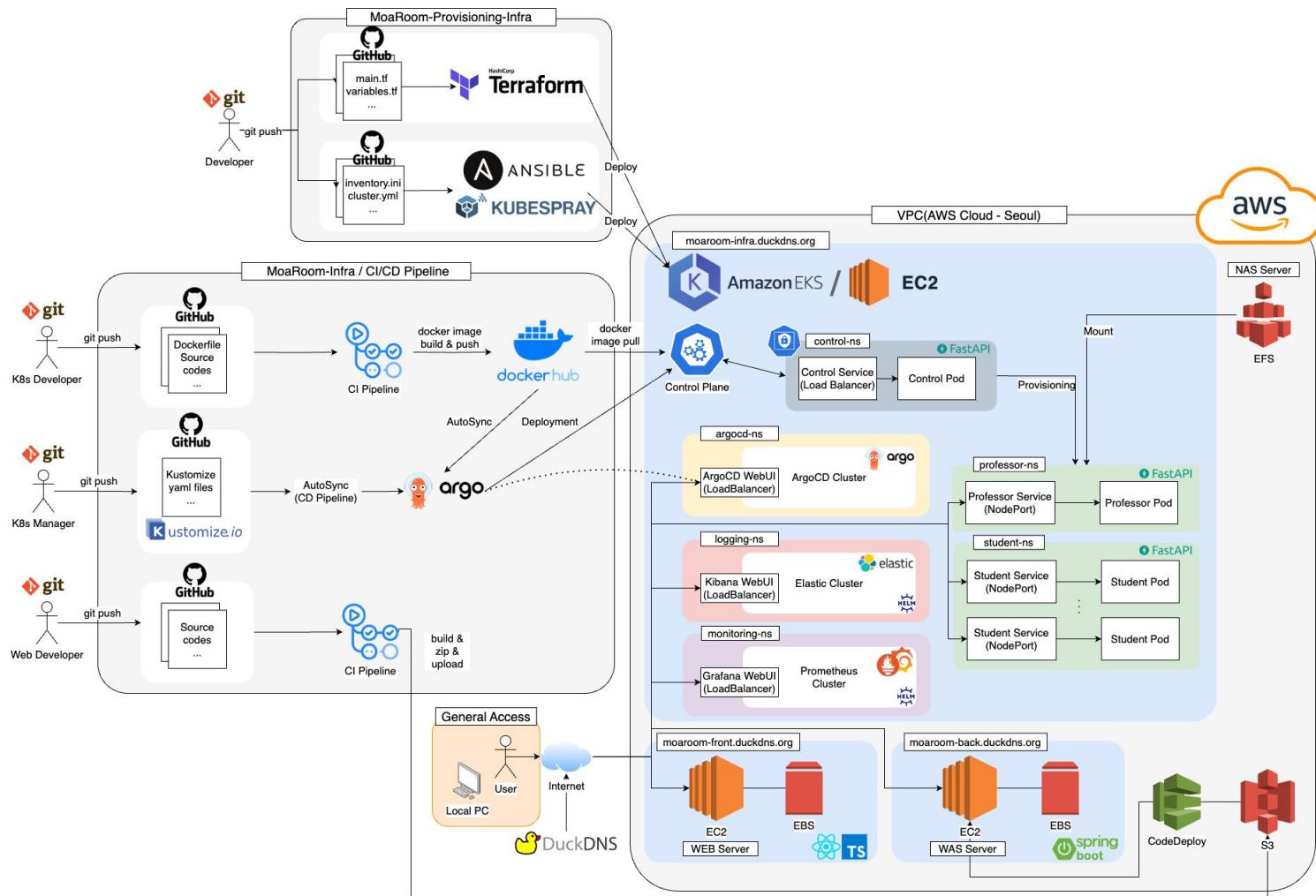
#컨테이너
#오픈소스

정 교육을 진행하는 플랫폼에서 사용 가능



02

아키텍처



쿠버네티스(Kubernetes) 클러스터 아키텍처

Docker란? 🚢

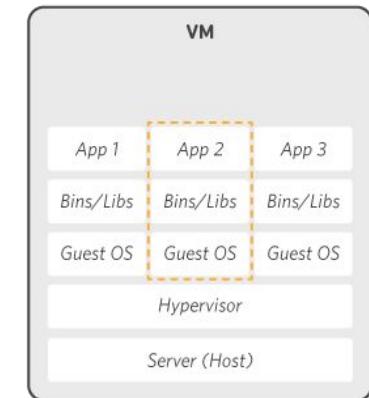
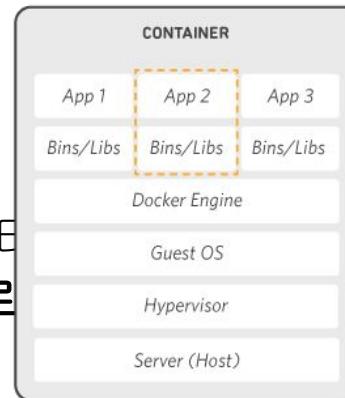
- 서버의 하드웨어를 가상화하여 호스트 OS에 여러 Guest OS를 설치하는 가상머신 대신, 운영 체제를 가상화하여 같은 호스트 OS의 커널을 공유하며 격리된 컴퓨팅 자원을 제공하는 가상화 기술인 컨테이너를 위한 운영 체제. (aws)



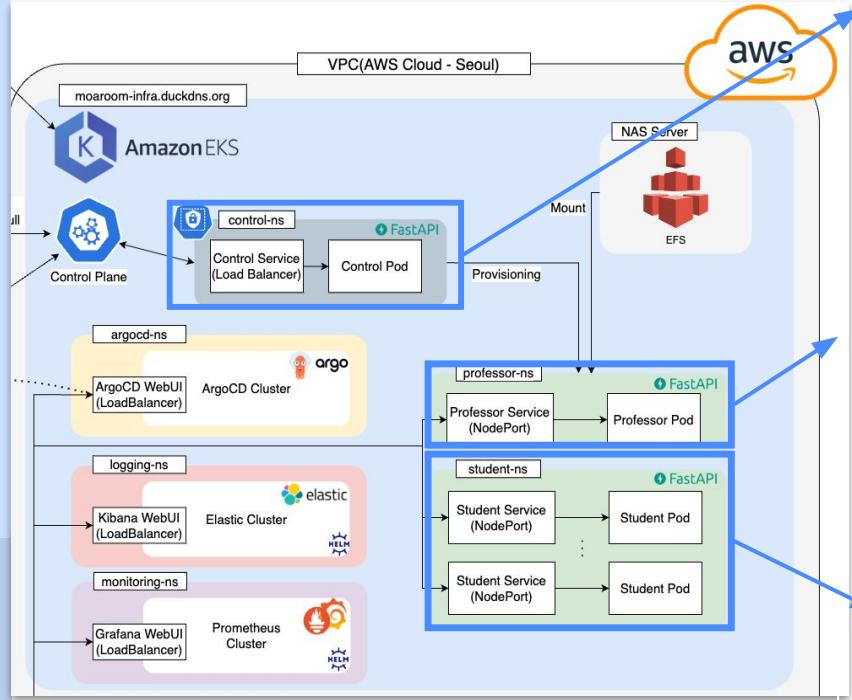
Amazon EKS

Kubernetes란?

- 컨테이너 오케스트레이션 툴
- 여러 개의 컨테이너들의 생명 주기, 다운 태그, 네트워크, 확장, 장애조치 등을 통합적으로 관리할 수 있도록 함
- **EKS**: AWS에서 Kubernetes 클러스터를 완전 관리형으로 운영할 수 있도록 하는 서비스



쿠버네티스(Kubernetes) 클러스터 아키텍처



Control namespace

- 교수 및 학생 네임스페이스에 해당 uid에 대한 컨테이너를 생성
- Kubernetes Control Plane과 소통
- 백엔드 서버와 통신하기 위한 api port 열림

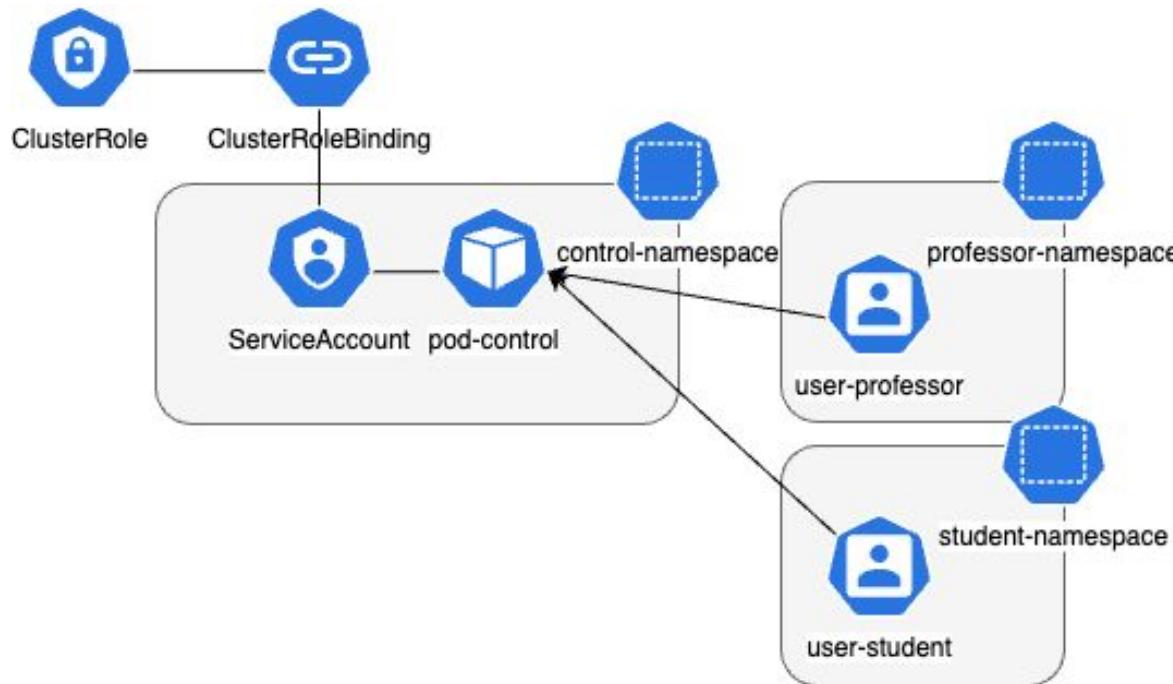
Professor namespace

- 교수 회원 가입 시 교수 당 하나의 컨테이너를 생성
- 백엔드 서버와 통신하기 위한 api port 열림

Student namespace

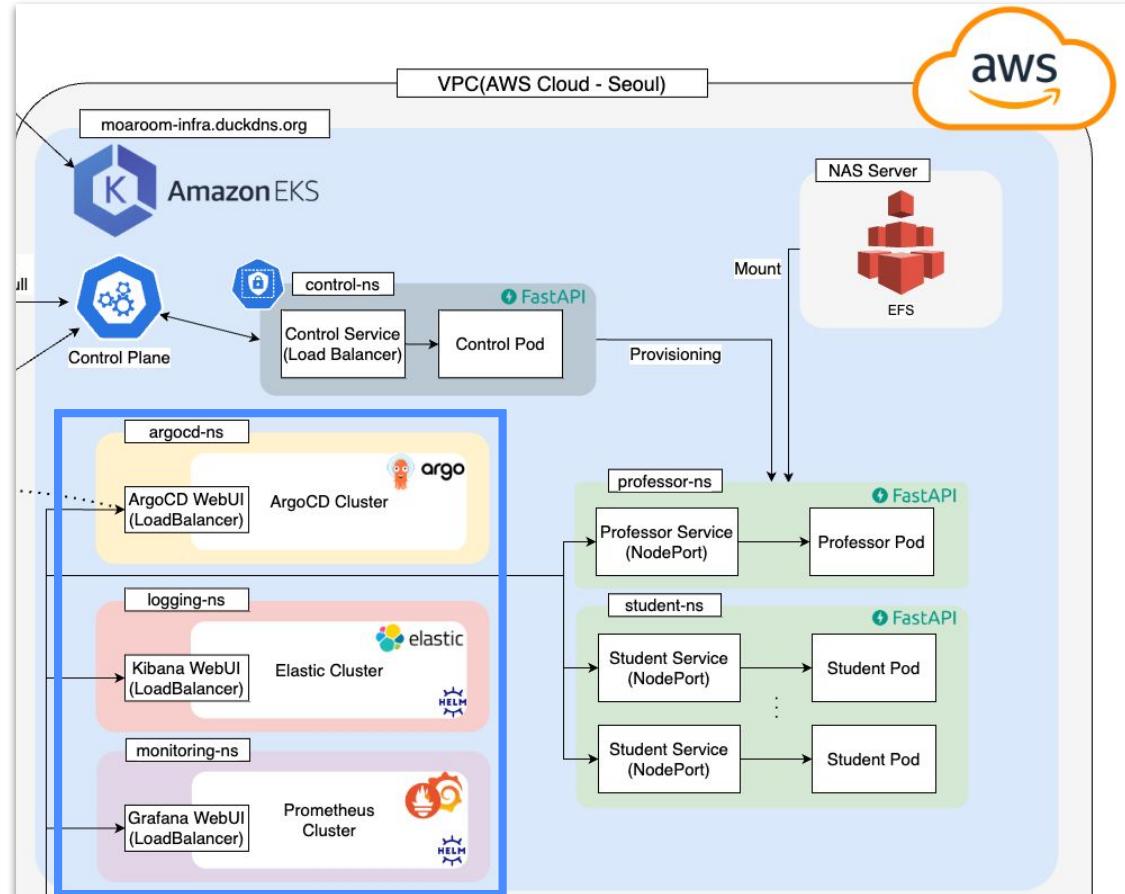
- 학생이 수강 신청 시 강의 하나 당 컨테이너를 생성
- 학생들이 프론트엔드로 보던 컨테이너에 접속

쿠버네티스(Kubernetes) 클러스터 아키텍처 - RBAC

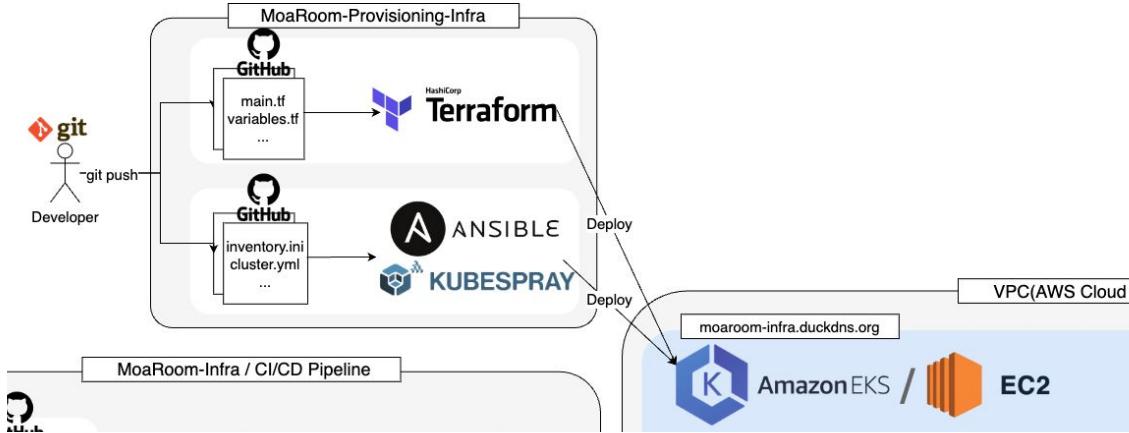


쿠버네티스(Kubernetes) 클러스터 아키텍처

- 기본적인 컨테이너 뿐만 아니라 Logging, Monitoring을 위한 클러스터 구축

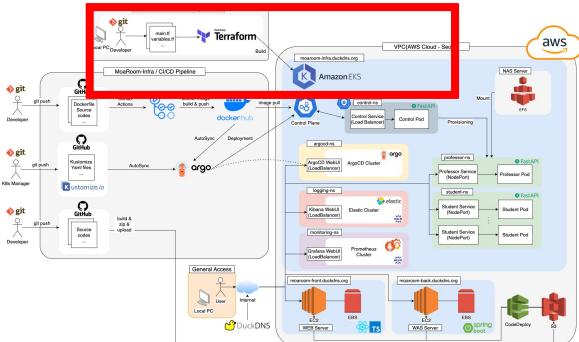


인프라 프로비저닝 파이프라인



- IaC(Infrastructure as a Code) 툴인 Terraform과 Kubespray를 사용해

코드 수정 만으로 인프라를 재현 가능하게 구현



인프라 프로비저닝 파이프라인

- Kubeadm vs Kubespray vs Terraform vs Rancher

Kubeadm

- K8s 클러스터를 컴포넌트별로 커스텀, 설치
- 클러스터 기본 구성을 위한 도구
- 인프라를 따로 관리해주지는 않음(인프라 종속성이 없다)
- 노드마다 동일한 설정 명령을 수행해야함(자동화 X)



Kubespray

- 배포를 위해 ***Ansible**을 사용함
- 원격 설치 및 클러스터 구성 가능
- 멀티 클라우드, 온프레미스 등 다양한 환경에서 배포



인프라 프로비저닝 파이프라인

- Kubeadm vs Kubespray vs Terraform vs Rancher

Rancher - SUSE

- Web UI(!!)로 K8s 배포 및 모니터링 가능(마우스 개발의 시대가...)
 - Yaml 파일 필요 없음
- 멀티 테넌트 운영 관리 가능



Terraform - HashiCorp

- 코드형 인프라 접근 방식으로 인프라 설정 및 관리
- 인프라 설정(Setup)에 적합



*Ansible

- 인프라 프로비저닝 및 배포 같은 작업 자동화에 중점
- 인프라 구성(Configuration)에 적합



인프라 프로비저닝 파이프라인

- Terraform vs Ansible

Terraform - HashiCorp

- 코드형 인프라 접근 방식으로 인프라 설정 및 관리
- 인프라 설정(Setup)에 적합
- 인프라 배포 후 상태(state)를 추적하고 변경 사항을 적용하기 전에 plan 할 수 있음
- 인프라 종속성 그래프도 쉽게 구축 가능(올바른 순서로 배포되도록)
- 인적 오류에 매우 민감함

*Ansible

- 인프라 프로비저닝 및 배포 같은 작업 자동화에 중점
- 인프라 구성(Configuration)에 적합
- 배포된 인프라에서 애플리케이션 수준의 작업을 수행하는 데 강점이 있음
- 순서에 민감한 복잡한 애플리케이션은 문제가 될 수 있음
- Terraform의 경우 일부 리소스에 문제가 생기면 그 상태를 추적하여 삭제하는데 Ansible은 다시 재배포함

<https://b-yarbrough.medium.com/understanding-terraform-and-ansible-438af22c0771>

인프라 프로비저닝 파이프라인

그 외에도 kops, kind, crossplane, k3s 등등이 있지만...

다음과 같은 기준을 두었습니다.

1. 오픈소스인가?
 - 테라폼 눈감아...
2. 참고할 자료가 많은가?(== 사용자 수가 많은가?)
3. 기한 내에 개발할 수 있는가?
4. 개발하면서 배울 것이 많을까?
5. 경량 클러스터가 아닌가?

-> Terraform, Kubespray 선택!

인프라 프로비저닝 파이프라인 = Terraform, Kubespray를 사용하여 느낀 단점

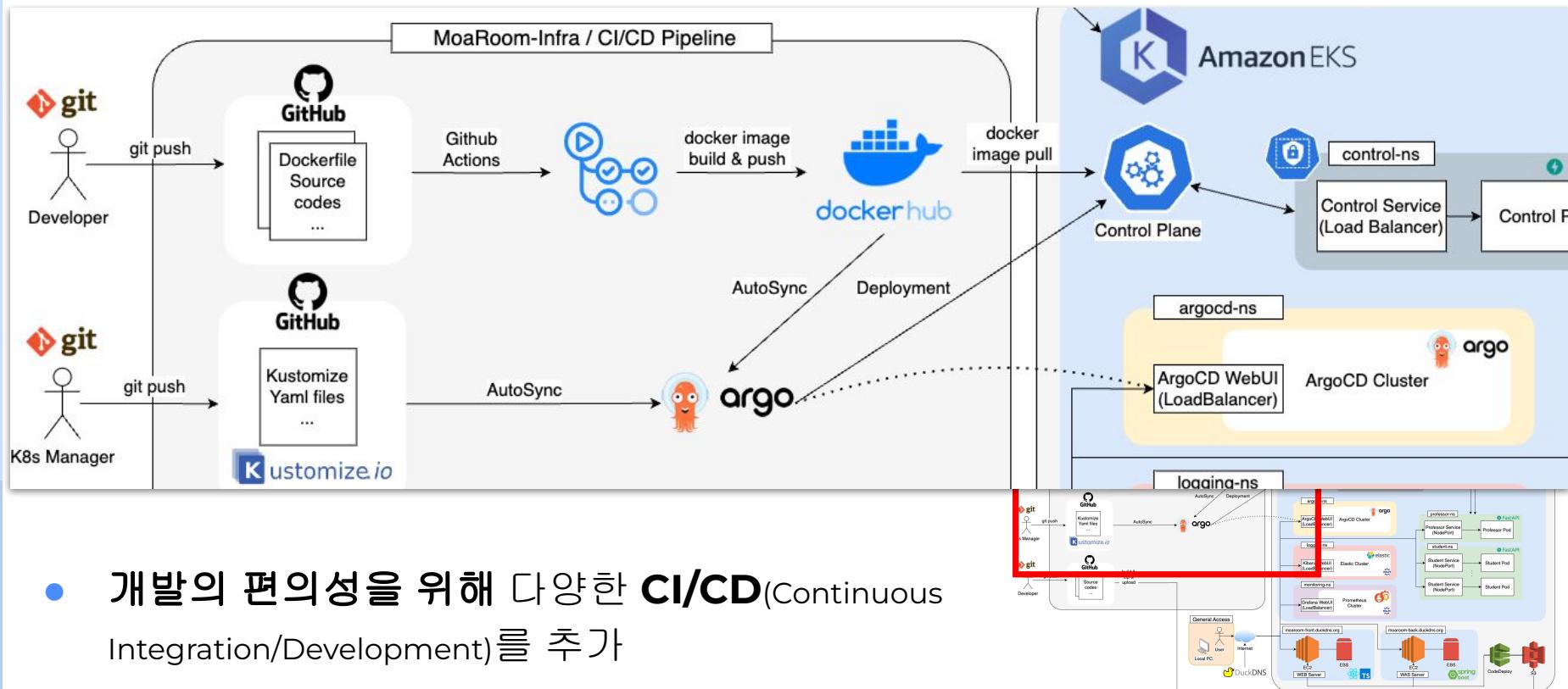
Terraform

- full-managed에만 이점이 있는 것 같다
- 누가 Ansible이랑 통합한 툴 만들어줬으면 좋겠다
- 혼자 개발해서 다행이다

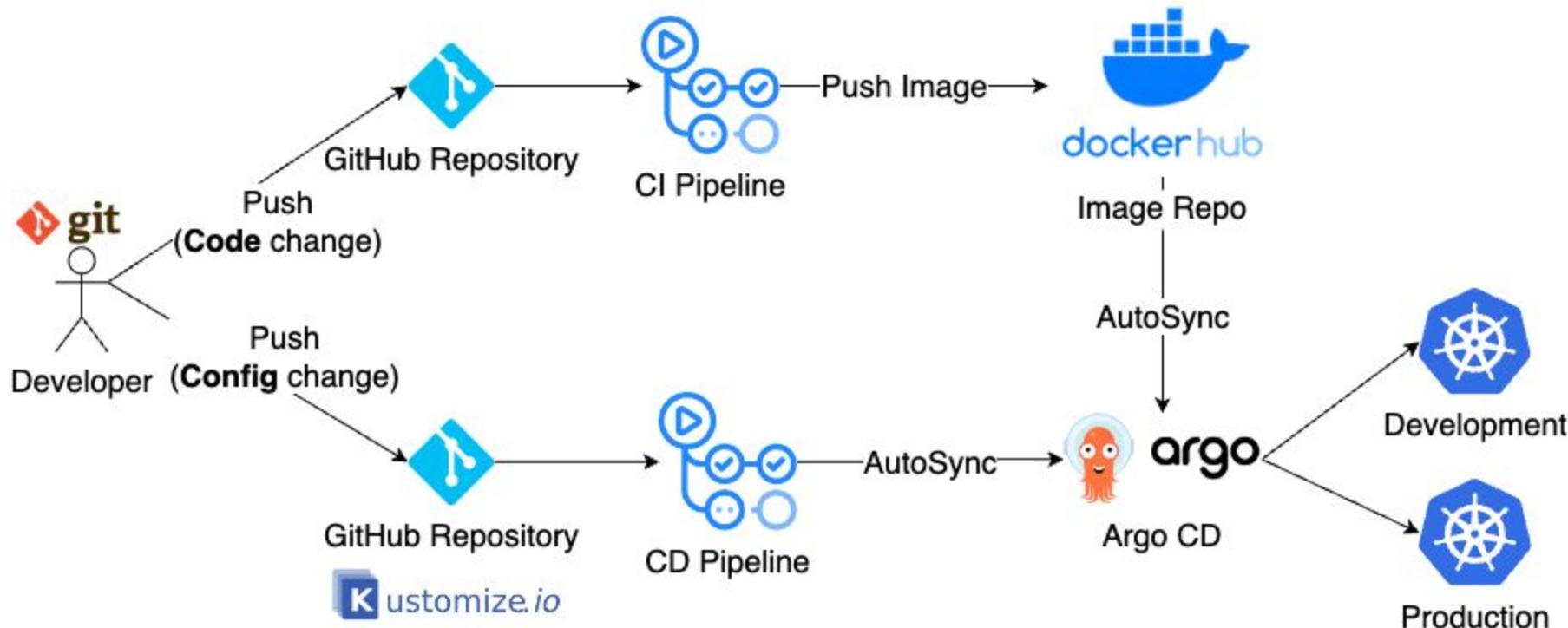
Kubespray

- 시간이 너무 오래 걸림...(리소스 프로비저닝도 하는 테라폼보다 오래 걸림)
- Node들 사이에 ssh 연결을 열어줘야함(컨트롤 플레인 사이, 컨트롤-데이터 플레인 사이)
- 로컬에서 kubectl 쓰려면 직접 원격으로 가능하게 포트 뚫고 IP 연결해야함

개발 및 CI/CD 파이프라인



개발 및 CI/CD 파이프라인



인프라 프로비저닝 파이프라인

- **Manifest** 관리의 양대 산맥

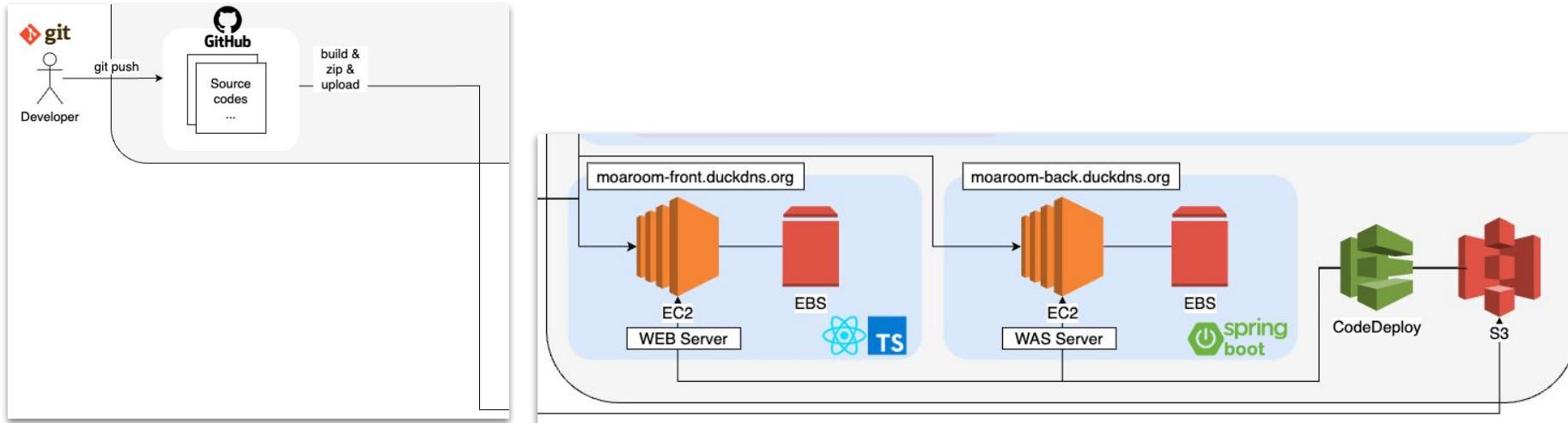
Helm

- 템플릿 방식을 이용함
- 변경해야 할 항목이 있으면 템플릿에서 항목 추가, 대체

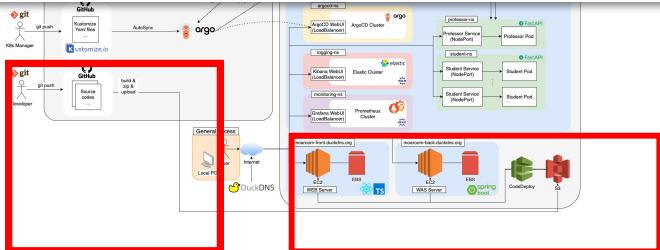
Kustomize

- polymorphic(다형성)한 상속 이용
 - 패치 접근 방식 사용(기존 manifest를 선언적으로 수정)
- staging 배포에 관한 내용을 서술한 파일이 generic 파일에 상속되어 커스텀한 설정을 통해 수정사항과 패치 추가(K8s+Customize = Kustomize...ㅋㅋ)

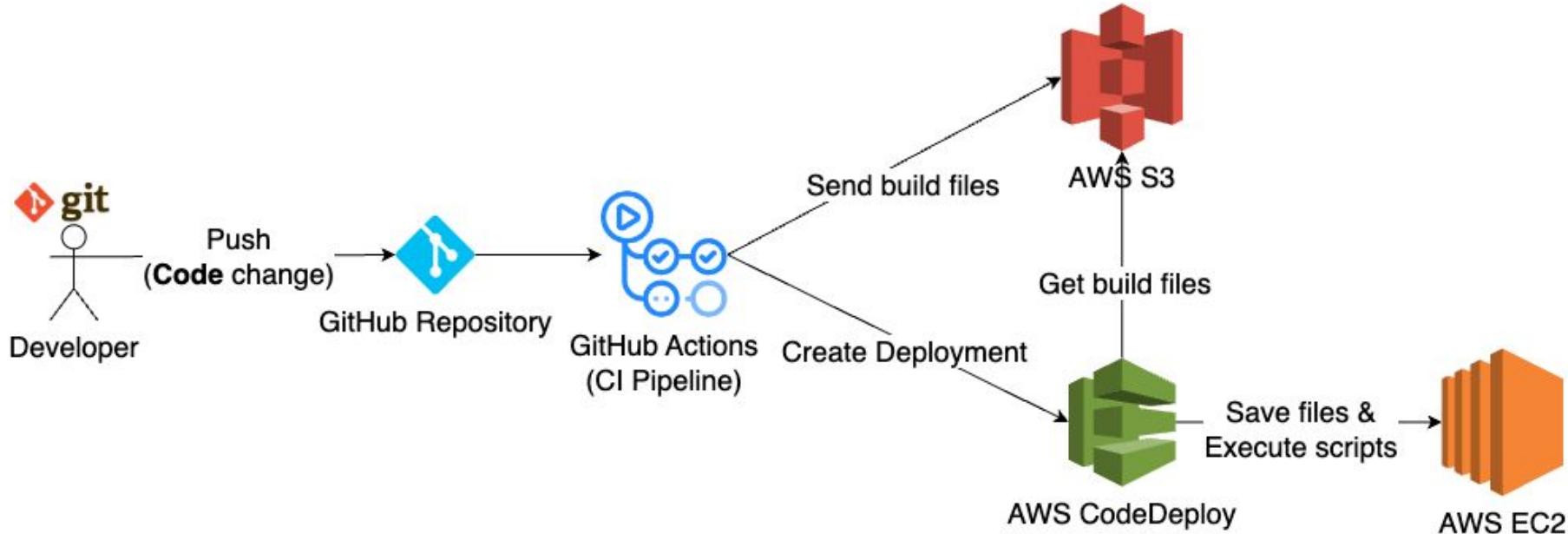
개발 및 CI/CD 파이프라인



- 개발의 편의성을 위해 다양한 CI/CD(Continuous Integration/Development)를 추가



개발 및 CI/CD 파이프라인



03

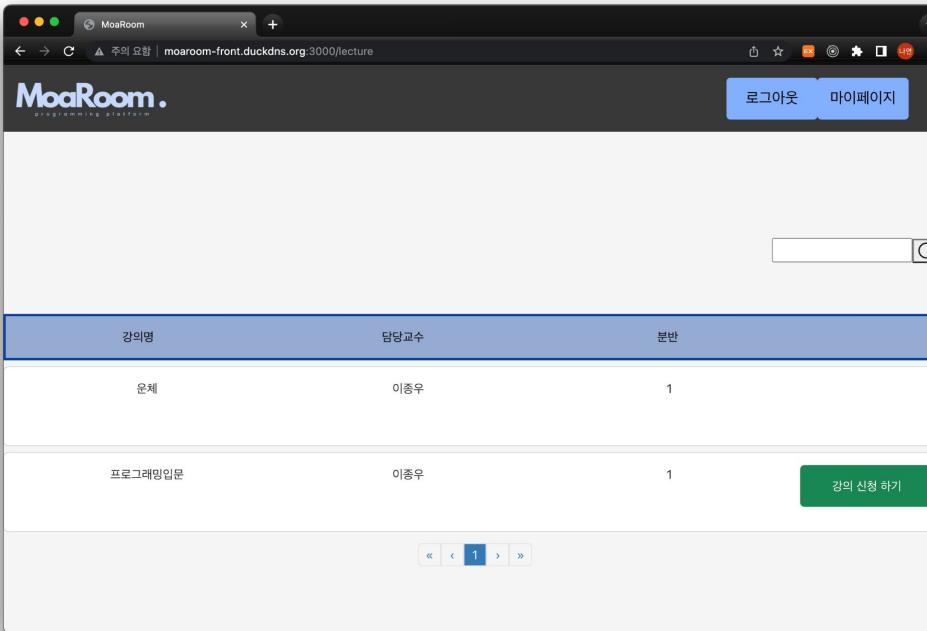
서비스

학생 시나리오

학생 시나리오

강의 수강

- 메인 페이지에서 강의 목록을 조회하여 수강 신청

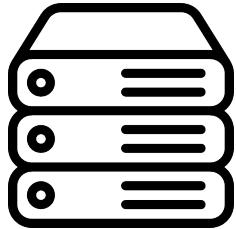


학생 시나리오

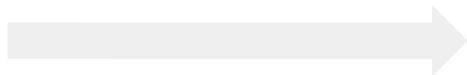
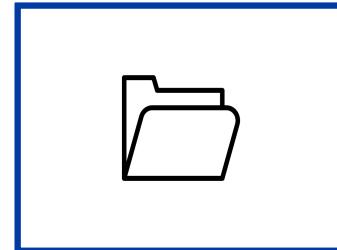
강의 수강

- 수강 신청 시 해당 강의의 과제를 수행할 컨테이너 생성
 - 컨테이너는 학생 당, 강의 당 하나씩 생성됨
 - ex) 학생10이 강의 1, 2를 수강할 시 강의1 용, 강의2 용 컨테이너 총 두 개가 생성된다.

Control Server



Student container

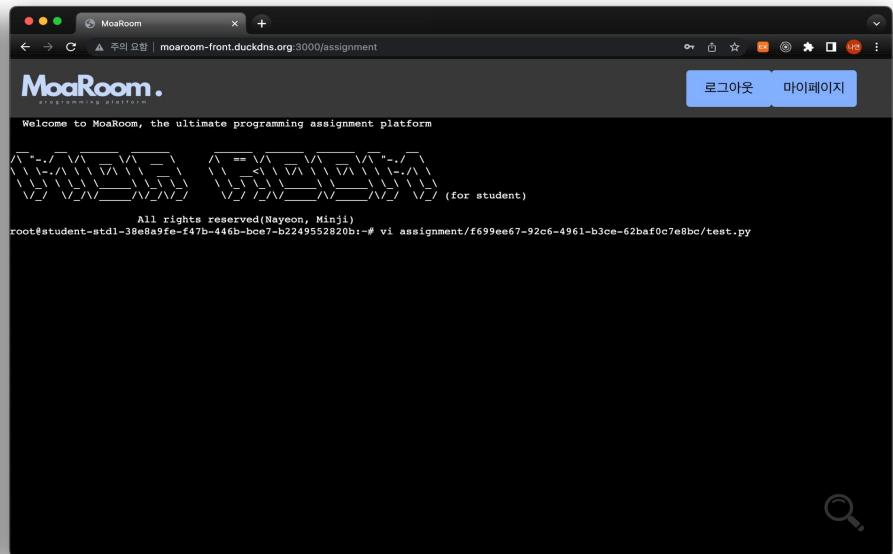
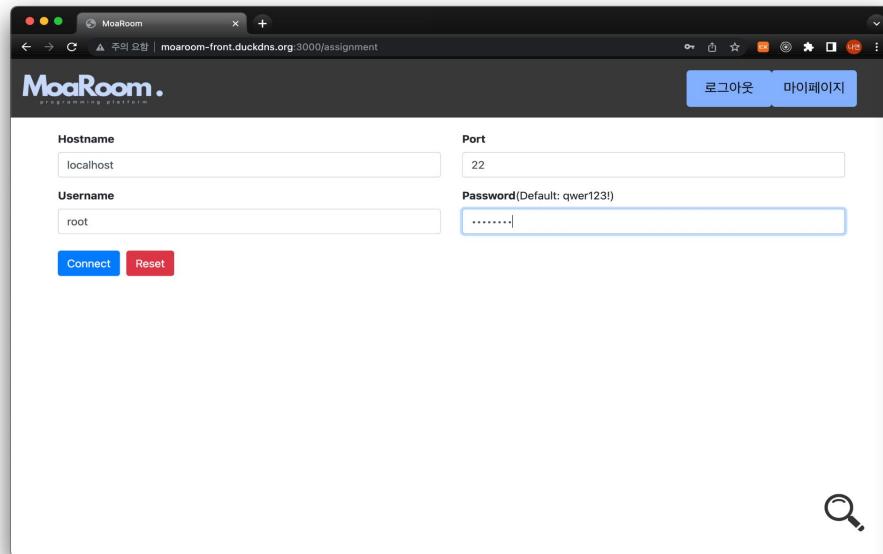


Control 서버에서
컨테이너 생성
trigger

학생 시나리오

과제 제출

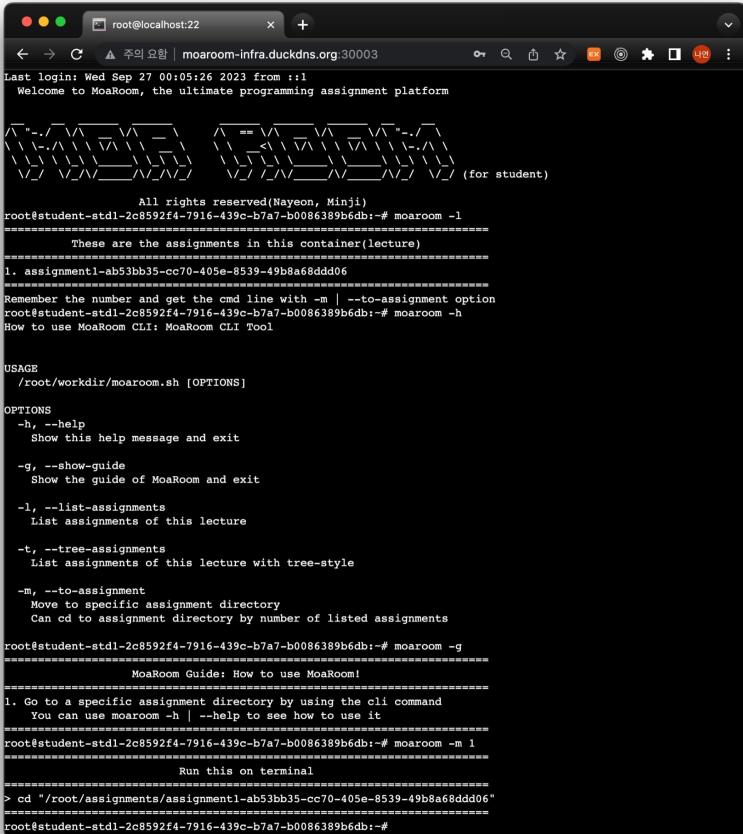
- 과제 페이지에서 전용 컨테이너에 접속하여 과제 수행
 - ssh로 터미널에 접속할 수 있는 webui를 통해 간편하게 접근한다.



학생 시나리오

과제 제출

- ‘moaroom [command]’를 이용해서 과제 수행 관련 command 가능
 - moaroom -h (--help)
 - moaroom -g (--show-guide)
 - moaroom -l (--list-assignments)
 - moaroom -t (--tree-assignments)
 - moaroom -m (--to-assignment)



The screenshot shows a terminal window titled 'root@localhost:22' with the URL 'moaroom-infra.duckdns.org:30003'. The window displays the Moaroom CLI interface, which includes a welcome message, assignment details, and usage information for various commands like -h, -g, -l, -t, and -m.

```
Last login: Wed Sep 27 00:05:26 2023 from ::1
Welcome to MoaRoom, the ultimate programming assignment platform

All rights reserved(Nayeon, Minji)
root@student-std1-2c8592f4-7916-439c-b7a7-b0086389b6db:# moaroom -l
=====
These are the assignments in this container(lecture)
=====
1. assignment1-ab53bb35-cc70-405e-8539-49b8a68dd0d6
=====
Remember the number and get the cmd line with -m | --to-assignment option
root@student-std1-2c8592f4-7916-439c-b7a7-b0086389b6db:# moaroom -h
How to use MoaRoom CLI: MoaRoom CLI Tool

USAGE
/root/workdir/moaroom.sh [OPTIONS]

OPTIONS
-h, --help
Show this help message and exit

-g, --show-guide
Show the guide of MoaRoom and exit

-l, --list-assignments
List assignments of this lecture

-t, --tree-assignments
List assignments of this lecture with tree-style

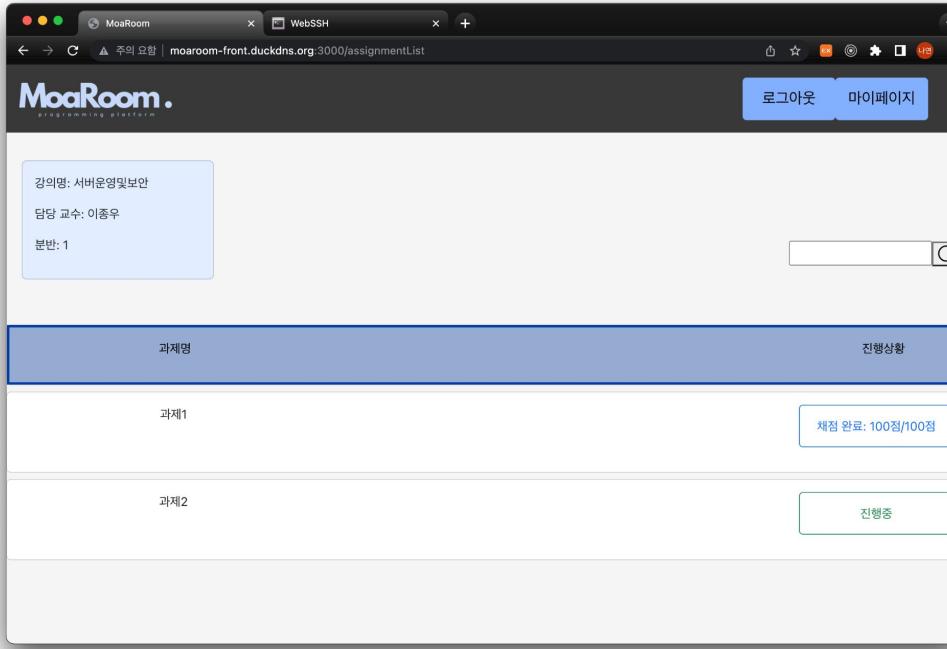
-m, --to-assignment
Move to specific assignment directory
Can cd to assignment directory by number of listed assignments

root@student-std1-2c8592f4-7916-439c-b7a7-b0086389b6db:# moaroom -g
=====
MoaRoom Guide: How to use MoaRoom
=====
1. Go to a specific assignment directory by using the cli command
You can use moaroom -h | --help to see how to use it
=====
root@student-std1-2c8592f4-7916-439c-b7a7-b0086389b6db:# moaroom -m 1
=====
Run this on terminal
=====
> cd "/root/assignments/assignment1-ab53bb35-cc70-405e-8539-49b8a68dd0d6"
=====
root@student-std1-2c8592f4-7916-439c-b7a7-b0086389b6db:#
```

학생 시나리오

점수 확인

- 마감 기한이 지나고 채점 시 점수 확인

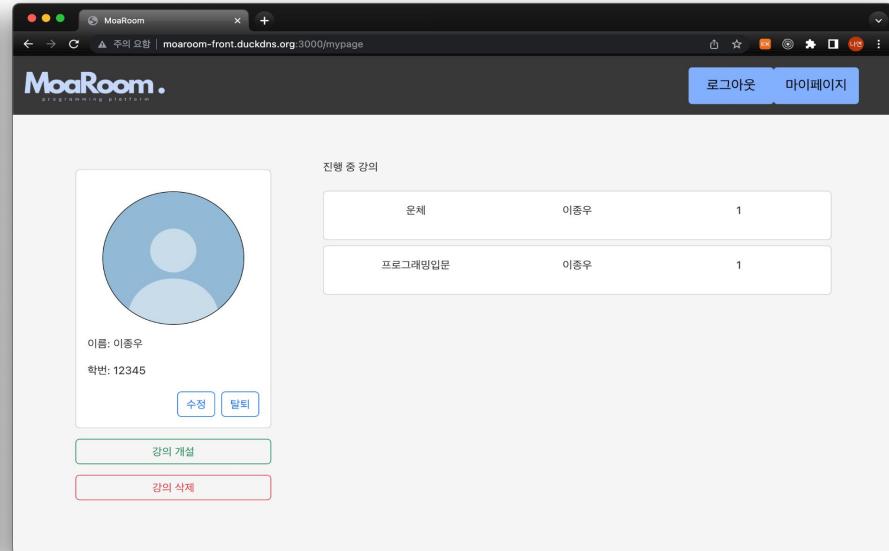
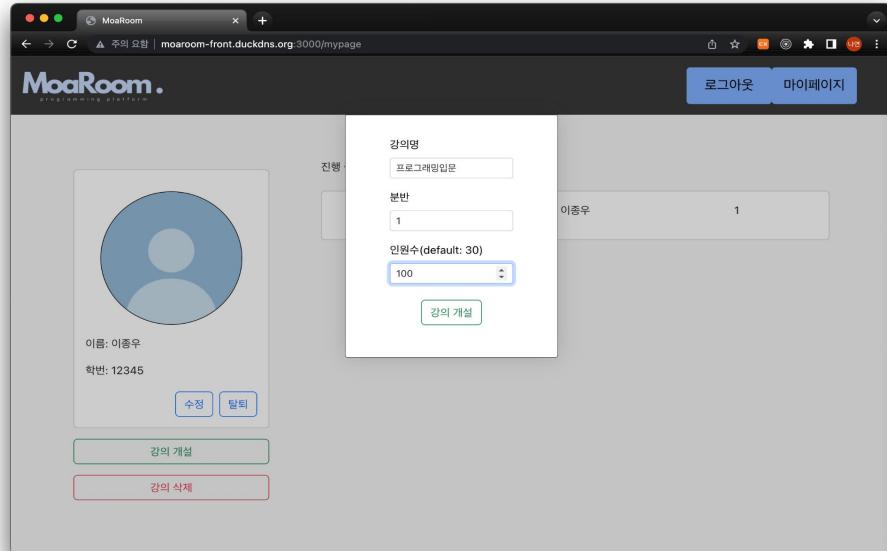


교수 시나리오

교수 시나리오

강의 - 생성

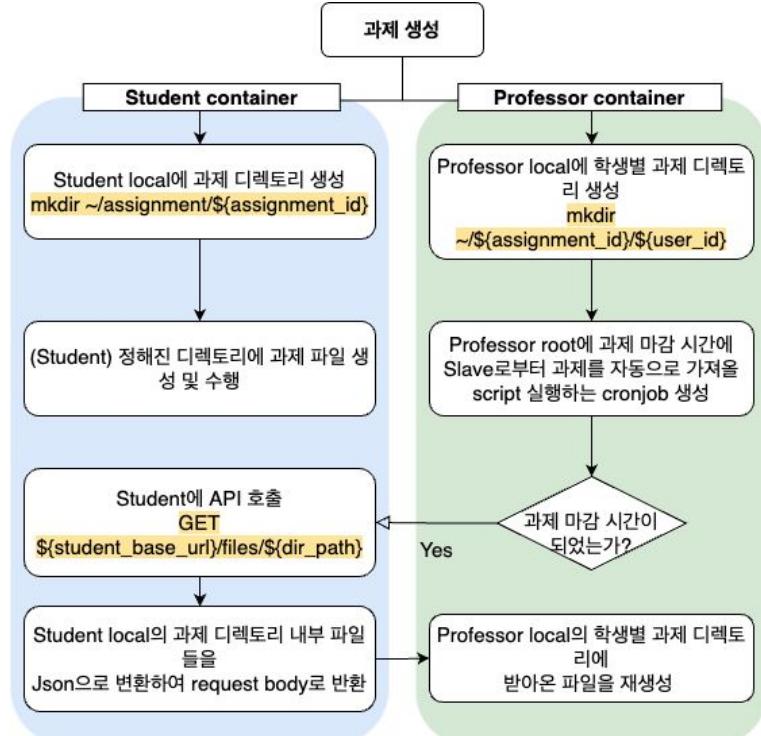
- 강의명, 분반, 인원수를 입력하여 강의를 생성



교수 시나리오

과제 - 생성 및 반환

- 과제 생성 시, 과제를 수강하는 학생들의 컨테이너에 과제를 수행할 폴더,
교수 컨테이너에는 과제를 저장할 폴더 생성
- 마감 시간에 맞춰 학생 컨테이너로부터 과제 및 실행 결과를 Retrieve하는 cronjob을 교수 컨테이너에 read-only로 생성함



교수 시나리오

과제 - 생성

- 과제명, 오픈/마감일, 설명, 답안, 실행시간을 통해 과제(py, c, cpp)를 생성

The screenshot shows the MoaRoom web application interface for creating a new assignment. The URL in the address bar is `moaroom-front.duckdns.org:3000/newassignment`. The page has a dark header with the MoaRoom logo and navigation links for '로그아웃' and 'マイページ'. The main form area contains fields for assignment details:

- 과제명: 과제1
- 오픈 예정일: 2023/09/15 00:49
- 마감일: 2023/09/23 00:48
- 설명: 프로그래밍입문 과제1입니다.
- 답안: hello
- 실행시간: 3

At the bottom right of the form is a blue '생성하기' (Create) button.

The screenshot shows the MoaRoom web application interface displaying a list of assignments. The URL in the address bar is `moaroom-front.duckdns.org:3000/assignmentList`. The page has a dark header with the MoaRoom logo and navigation links for '로그아웃' and 'マイページ'. The main content area shows a table with one row of data:

과제명	진행상황
과제1	진행중

On the right side of the table, there are buttons for '과제 추가' (Add Assignment) and '과제 삭제' (Delete Assignment). Above the table, there is a light blue box containing assignment details:

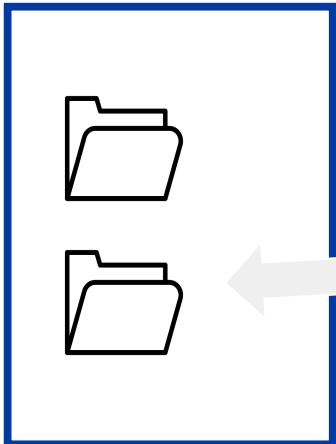
- 강의명: 프로그래밍입문
- 담당 교수: 이종우
- 분반: 1

At the bottom right of the page is a search bar with a magnifying glass icon.

교수 시나리오

과제 - 생성

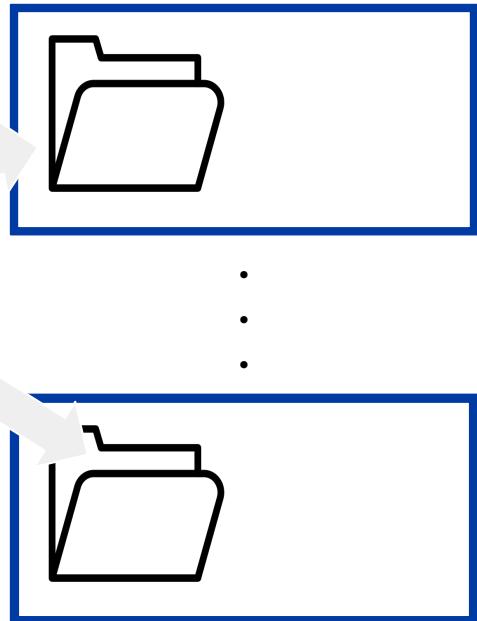
Professor container



~/강의id/학생id/과제id/

과제 생성 시
- 과제를 수행하는
학생들의 컨테이너에
과제를 수행할 폴더,
교수 컨테이너에는 과제를
저장할 폴더 생성

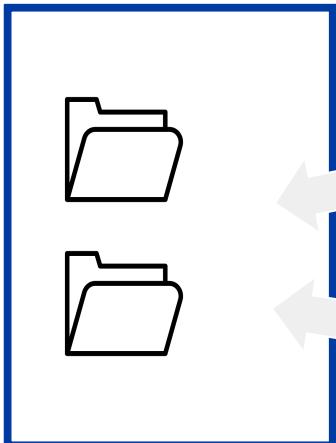
Student containers



교수 시나리오

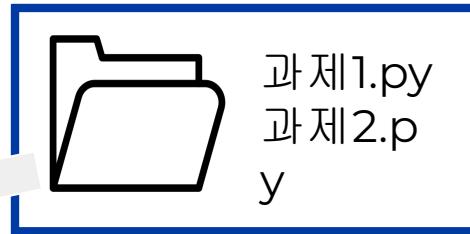
과제 - 생성 및 반환

Professor container



마감시간 시
과제 파일과 실행결과
파일을 전송하는 api
trigger
과제1.py values.txt

Student containers



과제1.py
과제2.p
y

⋮
⋮
⋮



Proj1.py
Proj2.py

⋮

교수 시나리오

과제 - 생성 및 반환

Professor container



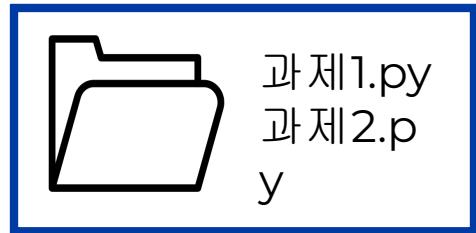
.../student1/assignment1/과제1.py
.../student1/assignment1/values.txt



...
.../student2/assignment1/Proj1.py
.../student2/assignment1/values.txt

마감시간 시
trigger

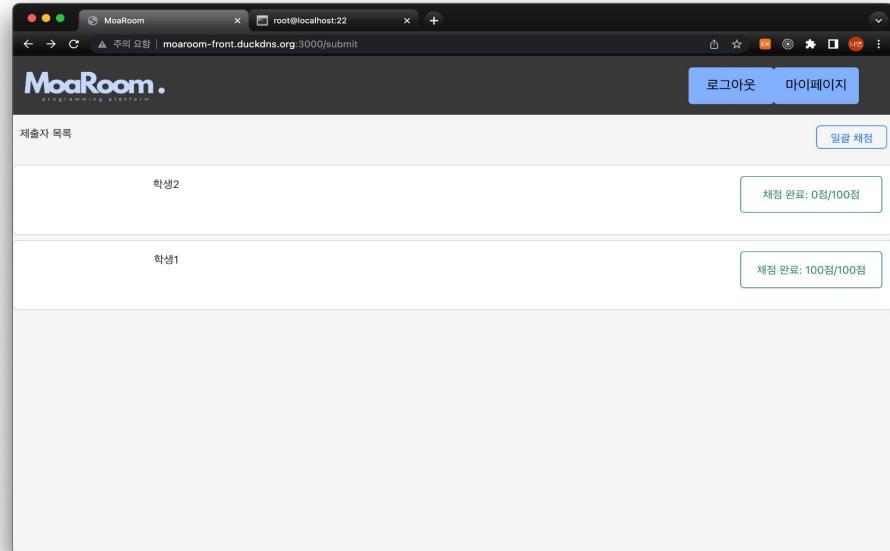
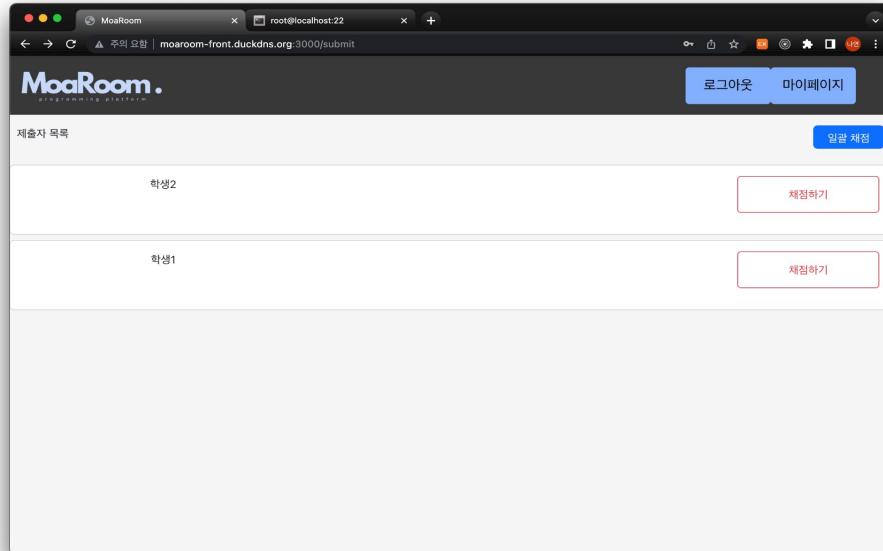
Student containers



교수 시나리오

과제 - 자동 채점

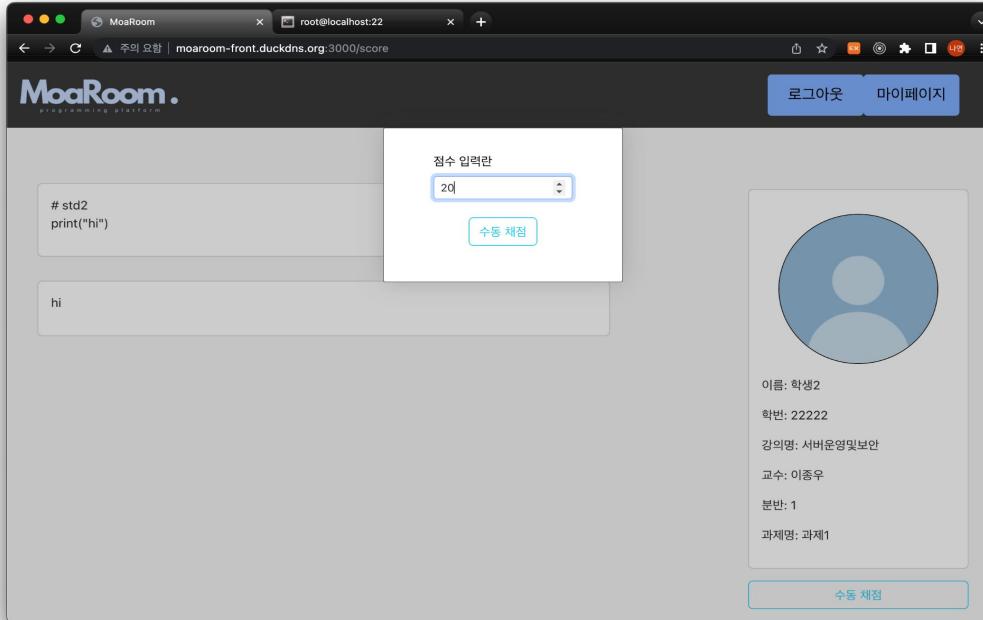
- 마감시간 이후 [일괄 채점] 또는 [수동 채점]을 통해 과제를 채점



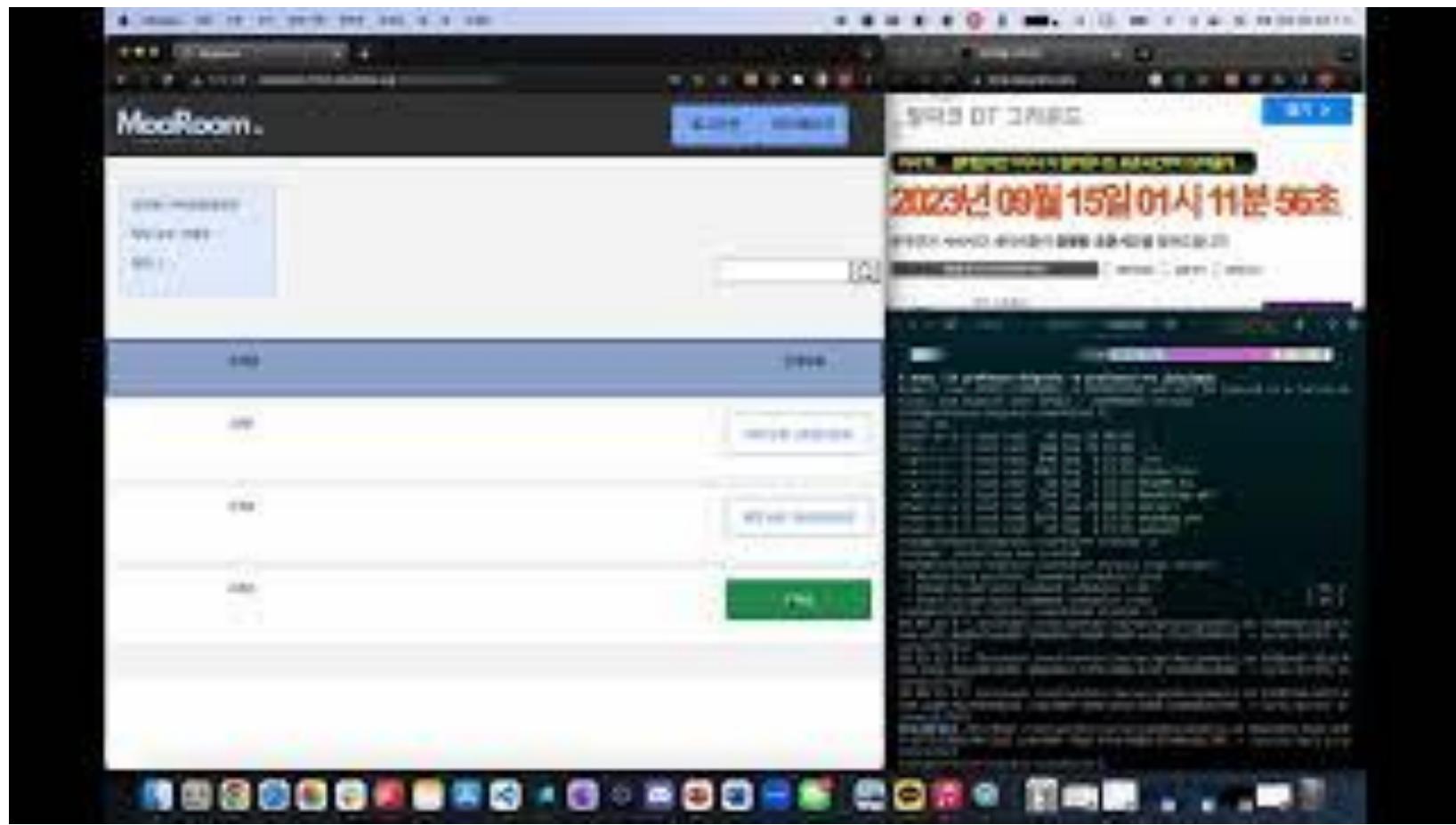
교수 시나리오

과제 - 수동 채점

- 마감시간 이후 [일괄 채점] 또는 [수동 채점]을 통해 과제를 채점



시연영상



<https://youtu.be/OGXiHSBJDq0>



04

시사점과 한계점

시사점

1. 컨테이너를 이용하여 효율적으로 리소스 사용

- 가상 머신보다 가볍지만 독립성을 보장하는 컨테이너를 쉽고 빠르게 배포/확장하고 관리를 자동화해주는 오픈소스 Kubernetes를 이용하여 제출자 별로 상호 독립적인 환경 조성

2. 타 플랫폼 대비 확장된 기능

- 스크립트 단위로 실행, 채점만 가능한 타 플랫폼에 비해 컨테이너 자체를 할당하여 과제 수행의 범위를 운영체제의 기능 사용까지 확장

3. 오픈소스로 구현

- 개발에 사용된 모든 소스는 오픈소스인 것만을 선택하였으며, 개발 결과물을 GitHub에 공개하여 누구든 재현 가능하고 변형하여 사용할 수 있게 함

한계점

기능 관련

- 컨테이너를 동적으로 할당할 수 있었다면 리소스를 더 효율적으로 관리할 수 있었을 것 같음
 - 사실 거의 완성했는데 테스트를 못함...
- 단계별 과제 수행 확인 및 채점 기능이 있었으면 더 유용했을 것 같음
 - 발전 가능성 유

UI 관련

- 사용자 친화적인 화면 인터페이스
 - 백엔드/인프라 개발에 집중하고 화면 디자인 및 컴포넌트 배치에 대한 지식이 많지 않아 사용자 입장에서 더 좋은 인터페이스를 제공하지 못한 것에 대한 아쉬움이 있음

QnA



예상 질문

1. 유저별 접속 URL 어떻게 구분했는지

- 원래 Ingress로 restAPI 기반으로 하고 싶었는데 (/nkeum, /minji, ...) 계속 설정을 업데이트 해줘야하는 문제가 발생해서 Port 기반으로 대신했음
- 사용 가능한 Port list를 npy로 control server에 저장해서 가능한 port 순서대로 사용함(DB 사용 안 했음)
- LB type으로 서비스 설정했는데 각 LB의 URL을 받아오는 방법을 못찾아서 NodePort로 해서 Node의 ENI에 public IP 달았음

예상 질문

2. 학생 <-> 교수 파드 사이 보안 문제

- 해결 못함..
- 현재 사용하는 API URL도 K8s 내부 DNS 기반으로 service discovery를 하고 있어서 Core DNS를 지우거나 막을 수도 없고..
- Namespace 별로 보안을 거는 방법을 못 찾았음

예상 질문

3. 파드 동적 할당?

- EC2 리소스 별로 띄울 수 있는 최대 Pod 개수 때문에 고안하게 됨
- 최대 사용자 200명을 목표로 했는데 -> 동시 접속자 수 30명으로 변경
- 거의 다 했음 진짜로...
- 과제 수행하는 디렉토리를 EFS에 [유저 UUID - 강의 UUID]로 저장해서 유저가 접속할 때 파드 띄우고 마운트하는 방식으로..
- 하지만...

별첨



깃허브 주소

Main



<https://github.com/MoaRoom>

Frontend



<https://github.com/MoaRoom/MoaRoom-Front>

Backend



<https://github.com/MoaRoom/MoaRoom-Back>

Infra



<https://github.com/MoaRoom/MoaRoom-Infra>

Infra Provisioning



<https://github.com/MoaRoom/Moaroom-Provisioning-Infra>

Thanks

