
안녕하세요! 저는 구름 속 질서를 설계하는 윤순상입니다!

겸손한 학습 의지력 / 조화를 이끄는 협업

About Me



저는 대학에서 MS Azure와 Amazon AWS를 통해 클라우드 컴퓨팅을 시작했고, 졸업 후 카카오 클라우드 스쿨 엔지니어 4기에서 모범생으로 수료했습니다. 이곳에서 클라우드 엔지니어와 DevOps 직무에 필요한 클라우드 인프라에 대해 입문했습니다.

이후 티맥스티베로에서 진행하는 TABA 7기에 참여하여 2개월 동안 알고 리뷰 프로젝트에 참여했습니다. 팀에서 저는 서비스 인프라 구현 및 배포를 주로 담당하였고, 프로젝트 목표를 달성하기 위한 협업 능력을 발휘한 경험이 있습니다.

Tech

Strong

Python / AWS / Docker / Linux(Rocky 9, Ubuntu)

Knowledgeable

MySQL / RESTfulAPI / Kubernetes / Docker-Swarm / Grafana /
Prometheus / Git

ETC

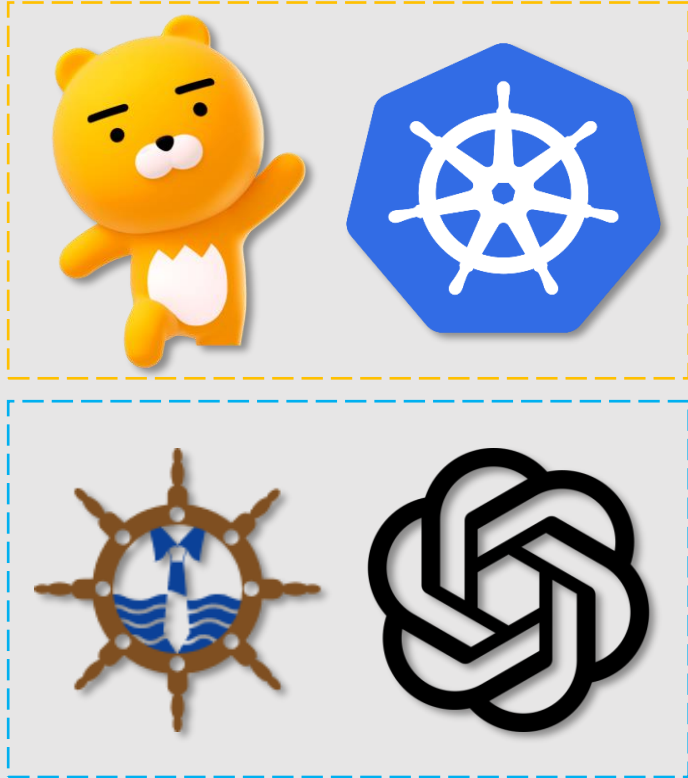
Fastapi / Django / ELK Stack / Ansible / Terraform / MongoDB /
PostgreSQL / TiberioDB / Jenkins / Argocd / KVM

Strengths 01



저의 강점 큰 강점은 겸손히 이어가는 학습 의지력입니다. SA 직무에서 가장 중요한 점은 배우는 자세라고 생각합니다. 저는 교육 및 프로젝트 프로그램 활동을 이어가면서 부족한 역량을 채우고 있고, 매듭으로 언제나 관련 자격 증명 시험을 통해 저의 위치를 확인하고 있습니다. 이러한 강점이 제가 맡은 업무를 책임감 있게 수행할 수 있을 것이라고 확신하면서 강점을 키우고 있습니다.

Strengths 02



저의 또 다른 강점은 조화를 이끄는 협업 능력입니다. 팀장으로 참여한 쿠버네티스 프로젝트와 팀원으로 참여한 알고 리뷰 프로젝트에서 동료들과 함께 토의하면서 여러 상황을 경험했습니다. 저는 동료들과 함께 의견을 통합시키기 위해 노력했고, 함께 시행착오를 통해 성장하며 프로젝트 목표를 달성할 수 있었습니다.

Project 1

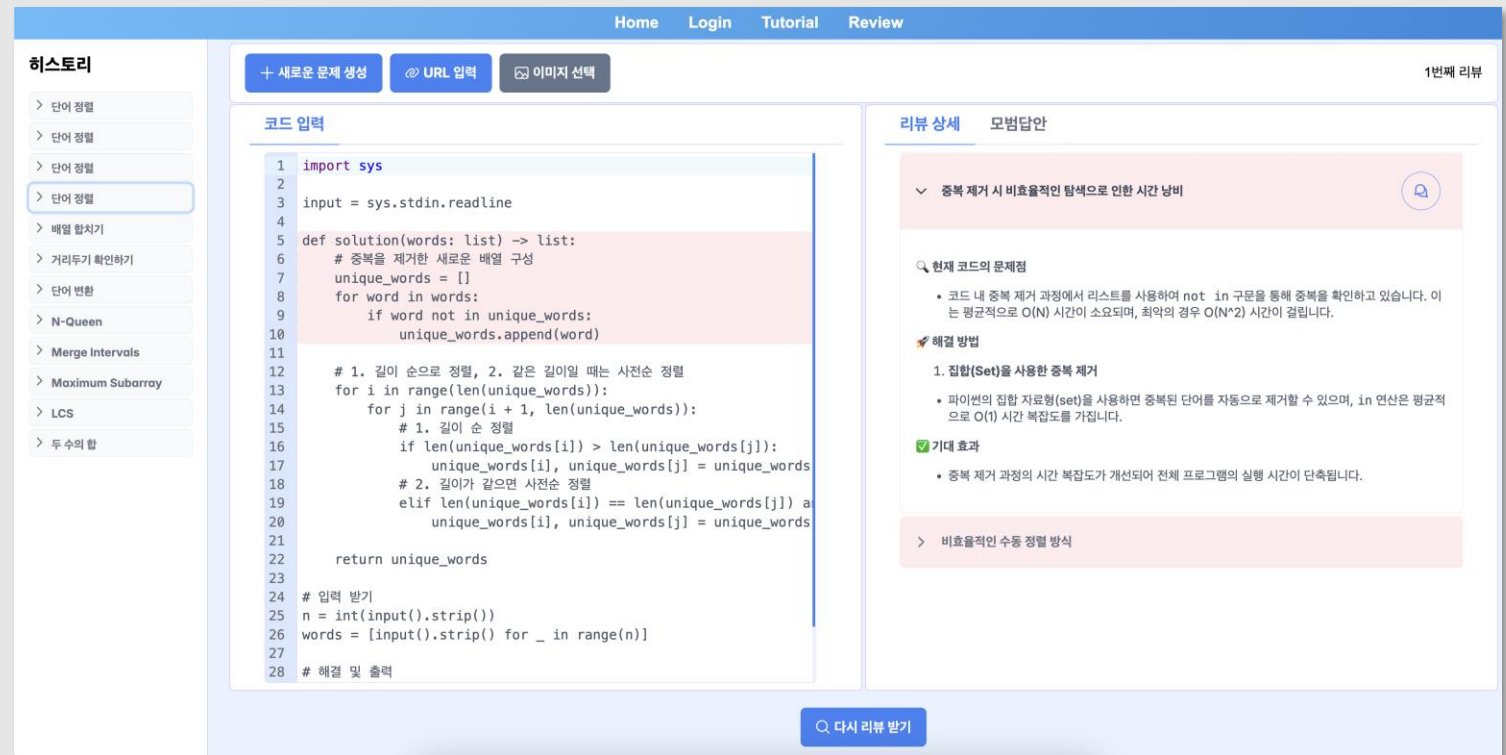
알고리뷰

개요

- 목적 : 알고리즘 문제 풀이에 대한 **코드 리뷰** 서비스를 제공
- 기간 : 2024.12.30 ~ 2025.2.28
- 인원 : Front-end 2명, Back-end 2명, AI 1명
- 담당 : 기획, Back-end, AWS 인프라
- 기여도 : 30%

기술스택

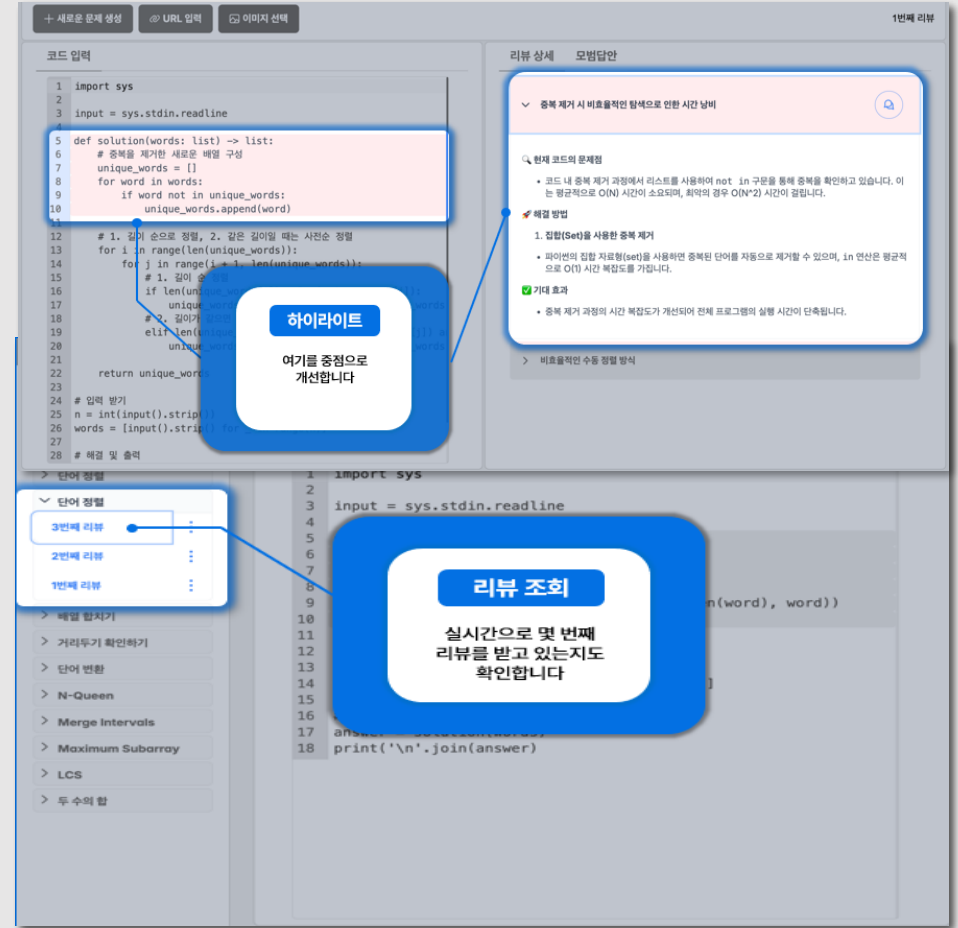
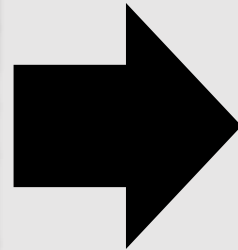
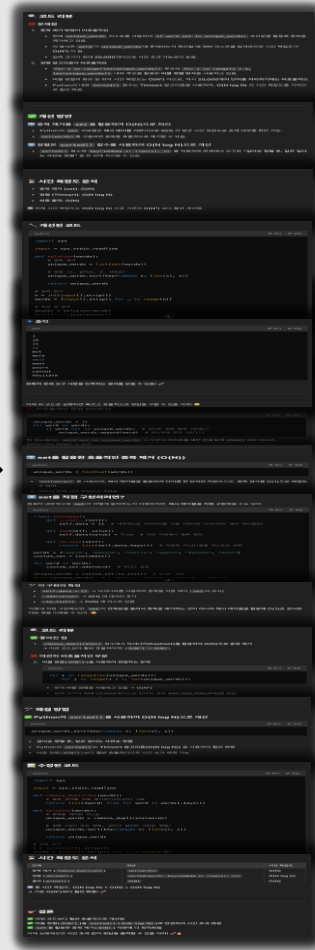
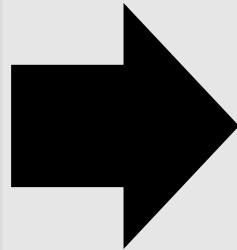
- FE - React, Vite, Figma
- **BE** - **Django**, Postgresql
- **Infra** - **AWS**(Private Subnet, EC2)
- **DevOps** - **Github**, **ECR**, **Docker-Swarm**



Project 1 - 소개



알고리즘 문제 풀이



UX / UI 관점에서 해결해보자!

생성형 AI 코드리뷰 우수!
그러나 관리 불편함이 존재

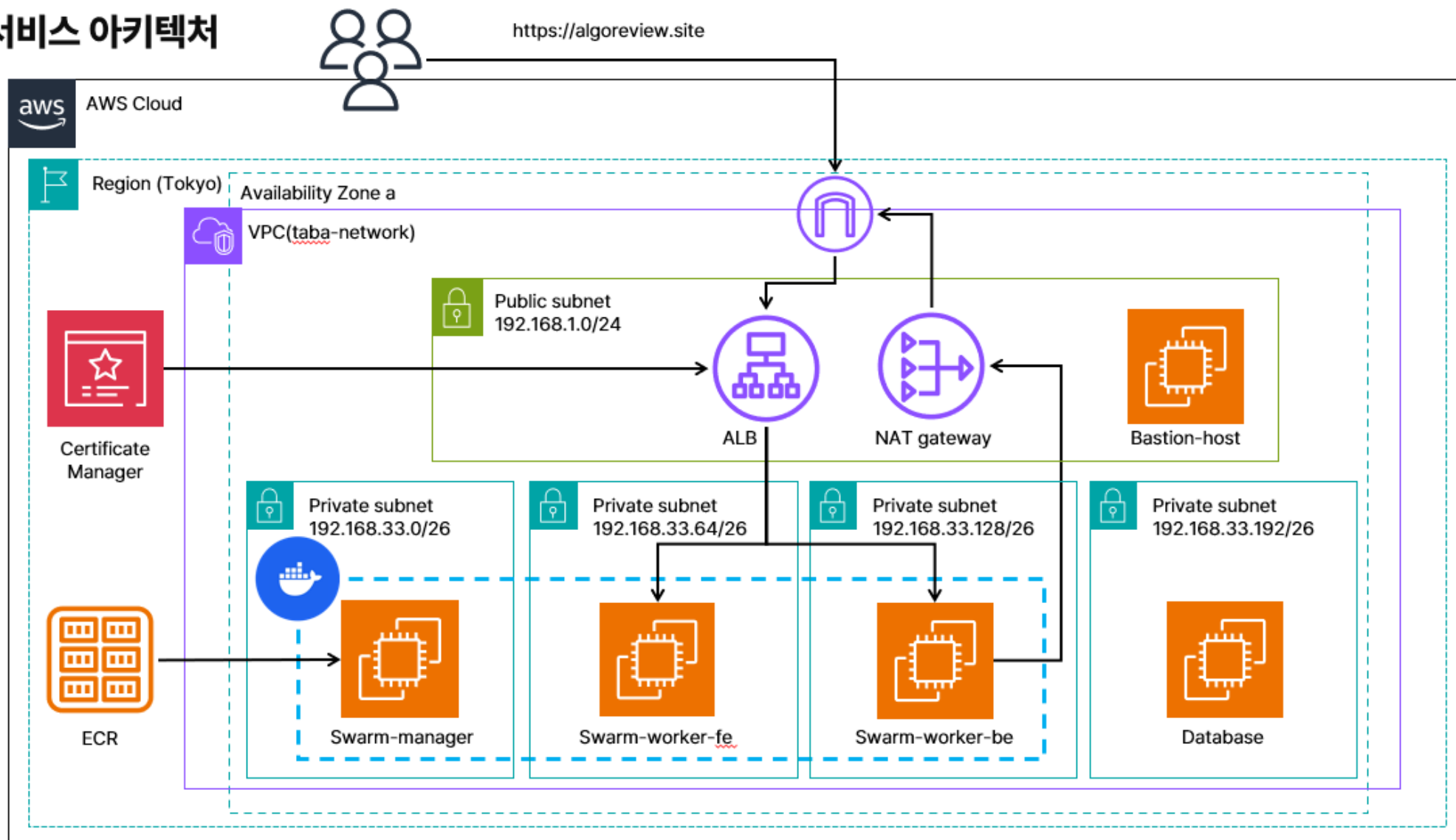
나의 역할 - Backend

general API 디버깅 리소스	
GET	/api 환영 메시지를 반환합니다
review 코드 리뷰 리소스	
POST	/api/v1/review 새로운 코드 리뷰를 생성하고 즉시 결과를 반환합니다
solution 모범 답안 리소스	
GET	/api/v1/solution/{problem_id} 특정 문제의 모범 답안 조회
POST	/api/v1/solution/{problem_id} 모범 답안 생성 요청
chatbot 챗봇 API	
POST	/api/v1/chatbot 특정 리뷰 피드백에 대한 질문 요청 (핑퐁 대화 지원)
history 사용자 리뷰 히스토리 리소스	
PUT	/api/v1/history/{history_id} 특정 문제 이름 수정
DELETE	/api/v1/history/{history_id} 특정 문제 삭제

problem 사용자 리뷰 문제 리소스	
PUT	/api/v1/problem/{problem_id} 특정 문제 이름 수정
DELETE	/api/v1/problem/{problem_id} 특정 문제 삭제
histories 사용자 리뷰 문제 + 히스토리 리소스	
GET	/api/v1/user-histories/{user_id} 사용자 히스토리 목록 조회
GET	/api/v1/histories/{history_id} 특정 히스토리 상세 조회
GET	/api/v1/histories/{problem_id}/first-review 최초 리뷰에 쓰인
user	
POST	/api/v1/user/login 사용자 로그인 인증

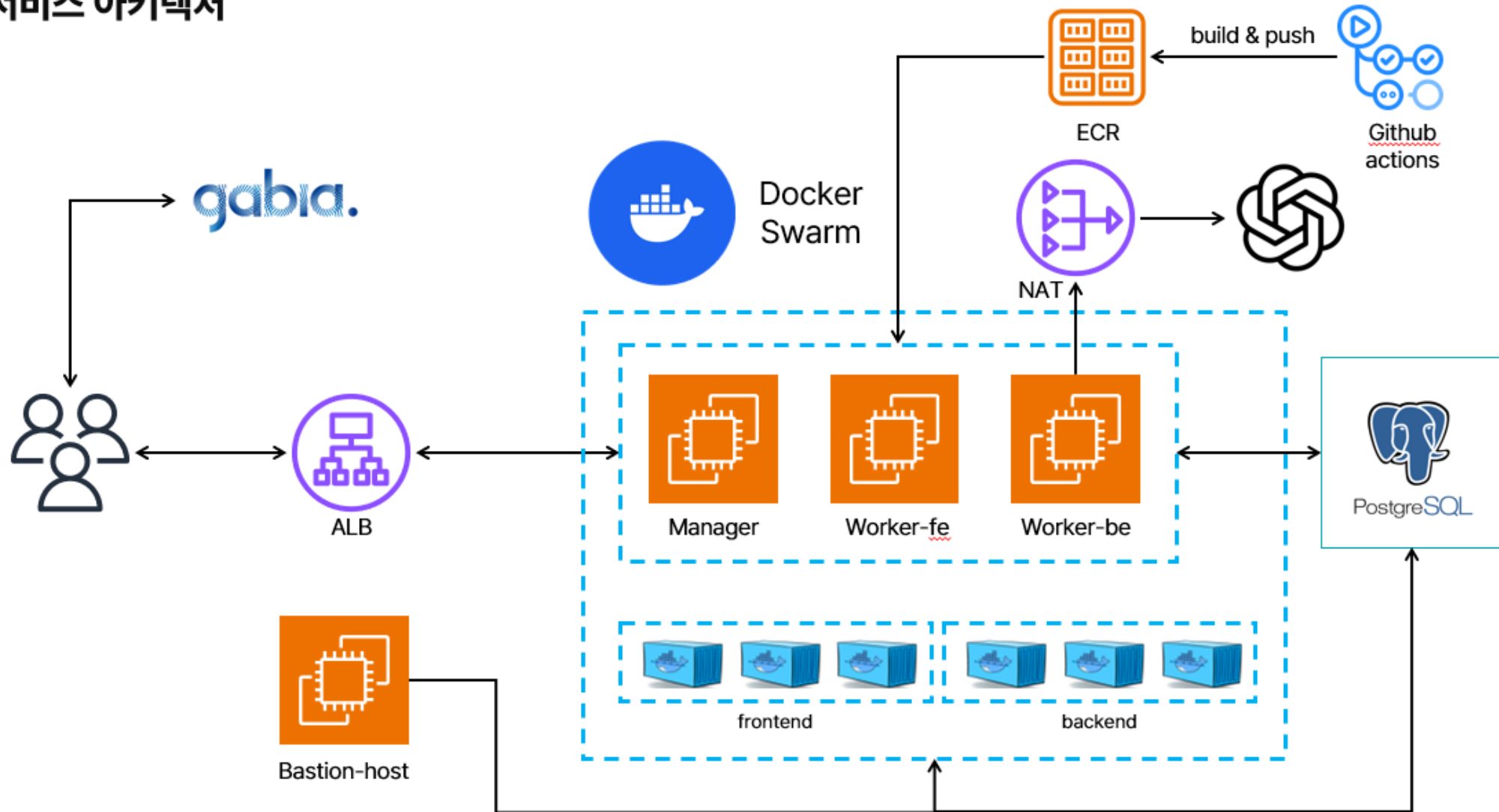
나의 역할 - AWS 인프라 & 배포

서비스 아키텍처



나의 역할 - AWS 인프라 & 배포

서비스 아키텍처



트러블 슈팅 - 문제

Docker-swarm ECR 저장소 Pull 이슈

```
ubuntu@ip-192-168-33-46:~$ docker service ps algoreview_backend
ID            NAME            CURRENT STATE    ERROR                                     PORTS
vscwzmtymv9y  algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b  ip-192-168
-33-135      Ready             Rejected 2 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b"
i2dkcmefvffu  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b  ip-192-168
-33-135      Shutdown          Rejected 7 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b"
pu4q6ms3tar7  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b  ip-192-168
-33-135      Shutdown          Rejected 12 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b"
zaqay2y7oglr  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b  ip-192-168
-33-135      Shutdown          Rejected 17 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b"
t1z2kw27x3hi  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b
Running       Pending 3 minutes ago    "no suitable node (scheduling ..."
```

```
ubuntu@ip-192-168-33-46:~$ docker service ps algoreview_frontend
ID            NAME            CURRENT STATE    ERROR                                     PORTS
pqayhnoofuk1  algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
8-33-87      Ready             Rejected 2 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest"
ultx6c93mcxu  \_ algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
8-33-87      Shutdown          Rejected 7 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest"
wfm1lw7v54z5  \_ algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
8-33-87      Shutdown          Rejected 12 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest"
```

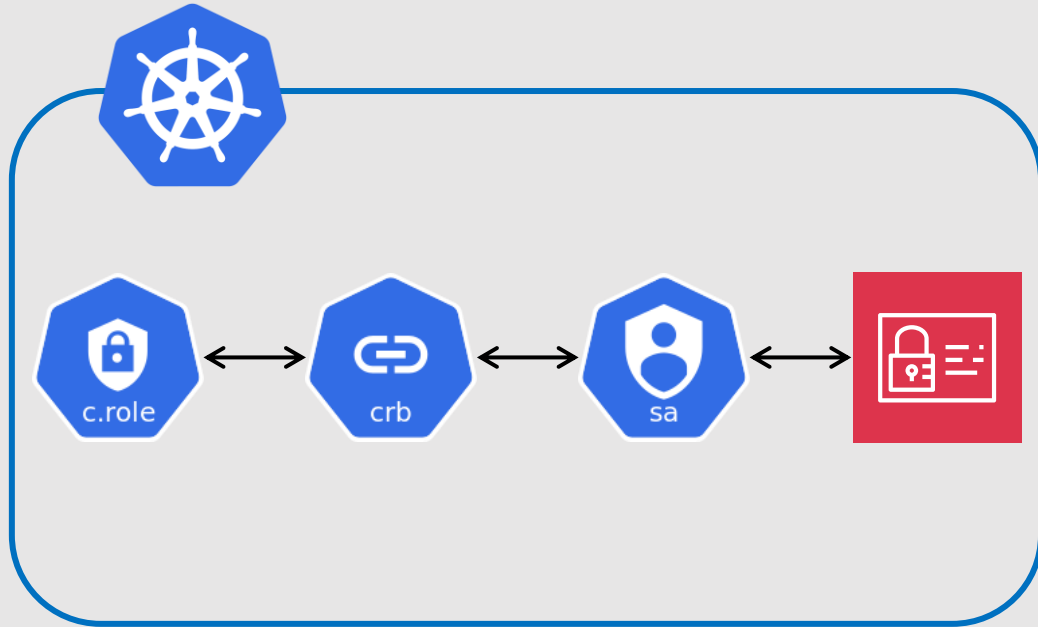
No such image 에러

```
ubuntu@ip-192-168-33-46:~$ docker service ps algoreview_backend
ID            NAME            CURRENT STATE    ERROR                                     PORTS
xosgges33wuh  algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest  ip-192-168
-33-135      Ready             Assigned less than a second ago
ivl75mtsbyk  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest  ip-192-168
-33-135      Ready             Rejected 3 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest"
8gyb2uw2s3r1  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest  ip-192-168
-33-135      Shutdown          Rejected 4 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest"
ep4ib0kzstlt  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest  ip-192-168
-33-135      Shutdown          Rejected 8 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest"
xp060x000ggu  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest  ip-192-168
-33-135      Shutdown          Rejected 9 seconds ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:latest"
t1z2kw27x3hi  \_ algoreview_backend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8c8a29b
Running       Pending 4 minutes ago    "no suitable node (scheduling ..."
```

```
ubuntu@ip-192-168-33-46:~$ docker service ps algoreview_frontend
ID            NAME            CURRENT STATE    ERROR                                     PORTS
ke5xw1zoscvo  algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
Ready         Pending less than a second ago
j57ex8xod1j3  \_ algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
8-33-87      Ready             Rejected 1 second ago    "No such image: 165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest"
w9dtr51s1p8x  \_ algoreview_frontend.1  165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/fe:latest  ip-192-168
```

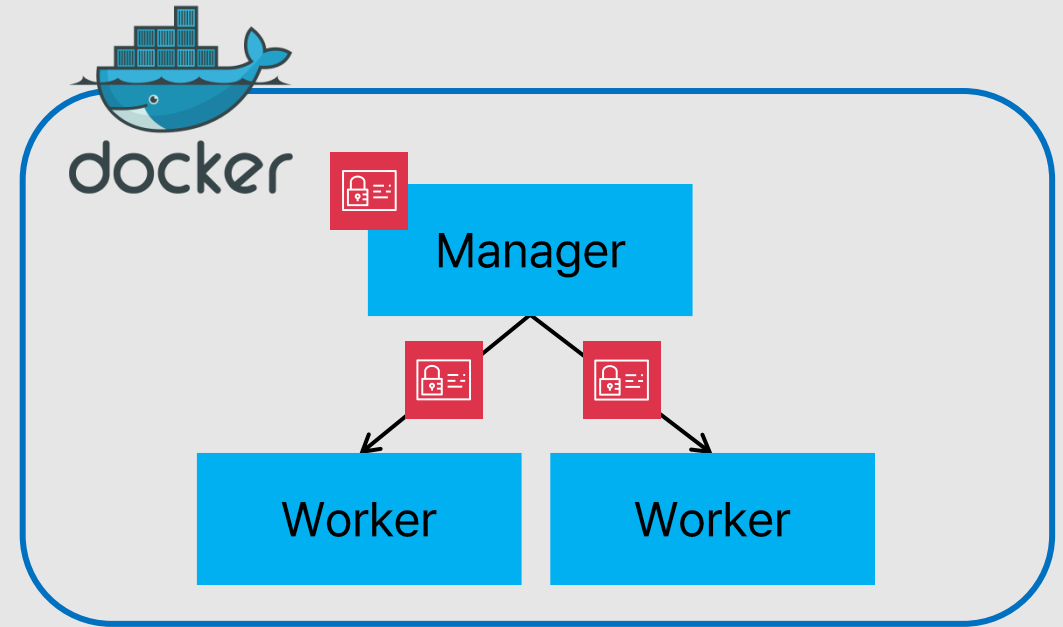
- 적절한 권한을 보유한 클러스터가 ECR 저장소의 이미지를 가져오지 못해 배포에 실패한 상황입니다.
- 처음에는 수동으로 ECR 저장소에서 최신 이미지를 가져와서 해결했습니다.
- 과거 k8s 경험은 이런 이슈를 겪지 못했습니다.
- **Docker-swarm과 k8s 환경은 어떤 차이가 있는 것일까요?**

트러블 슈팅 - 원인 분석



k8s(EKS) 인증 방식

- **ClusterRole, ClusterRoleBinding, ServiceAccount**를 사용하여 ECR 저장소 인증을 자동화할 수 있습니다.
- ServiceAccount는 AWS IAM 역할과 연결되어 자동으로 인증을 수행하기 때문에 ECR 저장소에서 이미지를 가져올 수 있습니다.



Docker Swarm 인증 방식

- Docker Swarm도 기본적으로 배포 과정에서 인증을 전달하지 않습니다.
- 따라서, k8s에서 구성한 SA처럼 Docker Swarm도 인증을 수행할 수 있도록 프로세스를 구성해야 합니다.
- **Docker Swarm**는 이를 **명령어 옵션으로 해결**할 수 있습니다. 그러나 Manager의 ECR 저장소 로그인 인증은 만료 기한이 있습니다. 따라서 자격 증명 갱신 자동화가 필요합니다.

트러블 슈팅 - 해결 방법

```
ubuntu@ip-192-168-33-46:~$ docker service update algoreview_backend --force --with-registry-auth
algoreview_backend
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Service algoreview_backend converged
ubuntu@ip-192-168-33-46:~$ docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE
lphr18xn5ubm c8a29b	algoreview_backend	replicated	2/1	165304734991.dkr.ecr.ap-northeast-1.amazonaws.com/algoreview/be:8
	PORTS			
	*:8000->8000/tcp			

- **--with-registry-auth** 옵션을 추가하여 배포 및 버전 업데이트 수행합니다.
- Manager의 ECR 저장소 로그인 자격 증명을 갱신하는 로직을 Cronjob으로 구현합니다.
- AWS ECR 공식 문서에 따르면 ECR 저장소 로그인 자격 증명의 유효 기간은 12시간입니다.
- 따라서 Cronjob으로 12시간마다 ecr-login 명령어를 수행하도록 Cronjob을 생성했습니다.

Project 2

샌드박스 테스트 플랫폼

개요

- 목적 : 헬름으로 빠르고 반복적인 샌드박스 테스트 환경을 지원하는 플랫폼 개발
- 기간 : 2024.04 ~ 2024.06
- 인원 : Front-end 2명, Back-end 1명, Infra 1명
- 담당 : PM, Back-end, DevOps, 일부 Infra
- 기여도 : 30%


기술스택

- FE – React, Vite, Figma
- BE – Fastapi, MongoDB, S3, Jenkins, Helm
- Infra – AWS(Private Subnet, EKS, EC2)
- DevOps – Prometheus, Grafana, CI/CD(Git, ECR, EKS)


SB 생성하기

프로젝트 이름

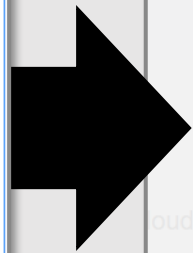
Helm 파일

 Click to upload or drag and drop
2048game-1.0.0.tgz
4.0 KB

Value 파일

 Click to upload or drag and drop
values.yaml
2.2 KB

생성하기

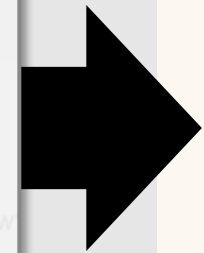


cloudcrew1

도메인 주소: a07aab35001e24270b26a96377d2d6f6-ce63 

- 이름: cloudcrew1
- 위치: cloudcrew1
- 상태: deployed
- 리비전: 1
- 차트: 2048game-1.0.0
- 버전: 1.15.0-rolling
- 마지막 수정 날짜: Wed Jun 5 10:38:15 2024

프로젝트 수정하기 프로젝트 삭제하기



2048

Join the numbers and get to the 2048 tile!

SCORE 0 BEST 0

New Game

2			
2			

Project 2 - 소개

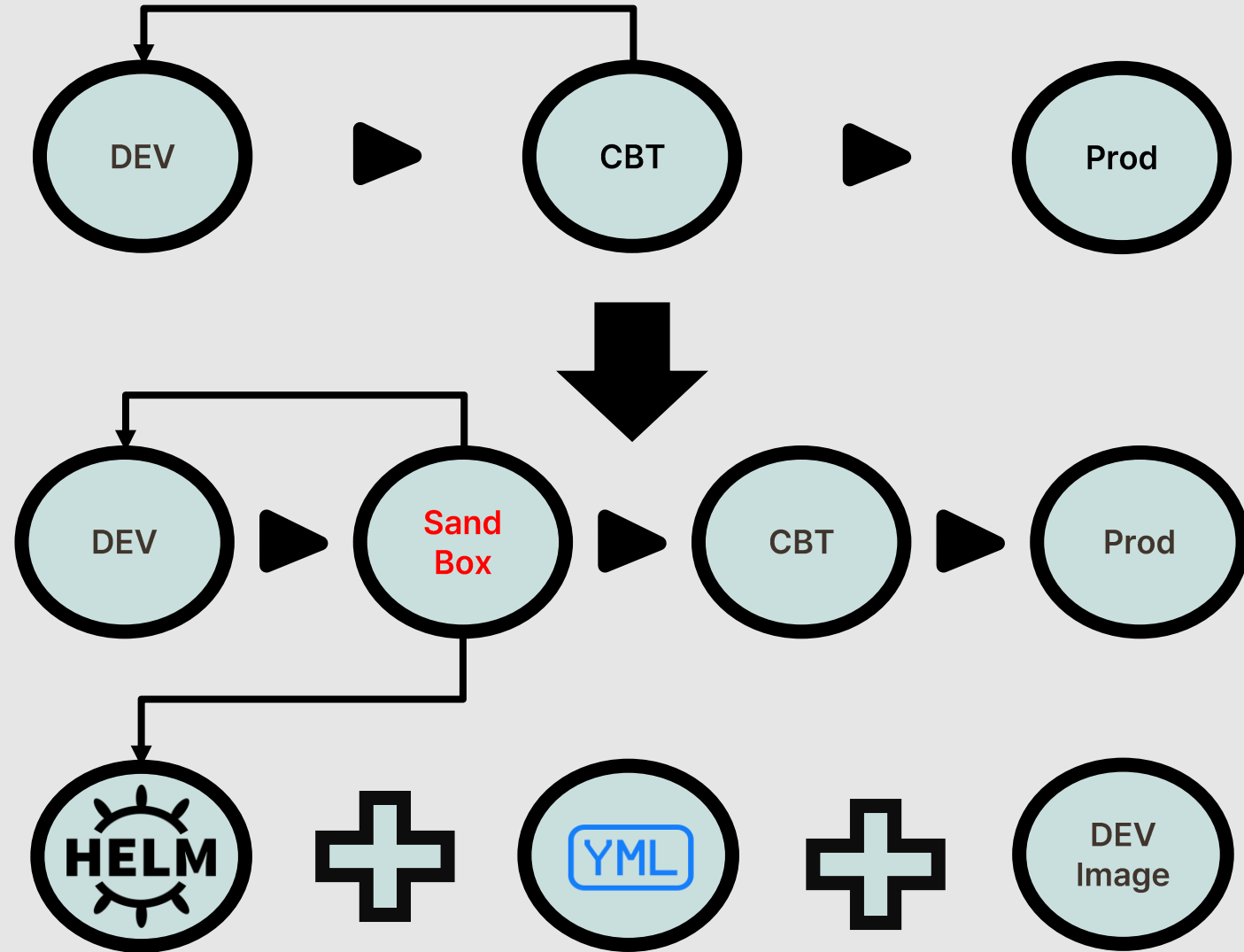
기존 개발 프로세스의 문제점

- 운영 환경 ≠ 개발 환경 → 예상치 못한 버그 발생
- CBT 지연 = 비용 증가 & 개발 속도 저하

샌드박스 테스트 플랫폼을 도입해 해결

- Helm + YAML로 가상화된 샌드박스 즉시 생성
- 개발자가 CBT 전 미리 테스트하여 버그 최소화
- 개발 버전 관리 -> 결과가 좋은 버전을 선정

샌드박스로 비용 절감 + 개발 속도 향상 = 안정적인 배포 가능!



나의 역할 - Backend

- 프로젝트 기능 플로우 및 데이터 설계
- Swagger를 통해 API 기능 명세 후 RESTful API 개발

default

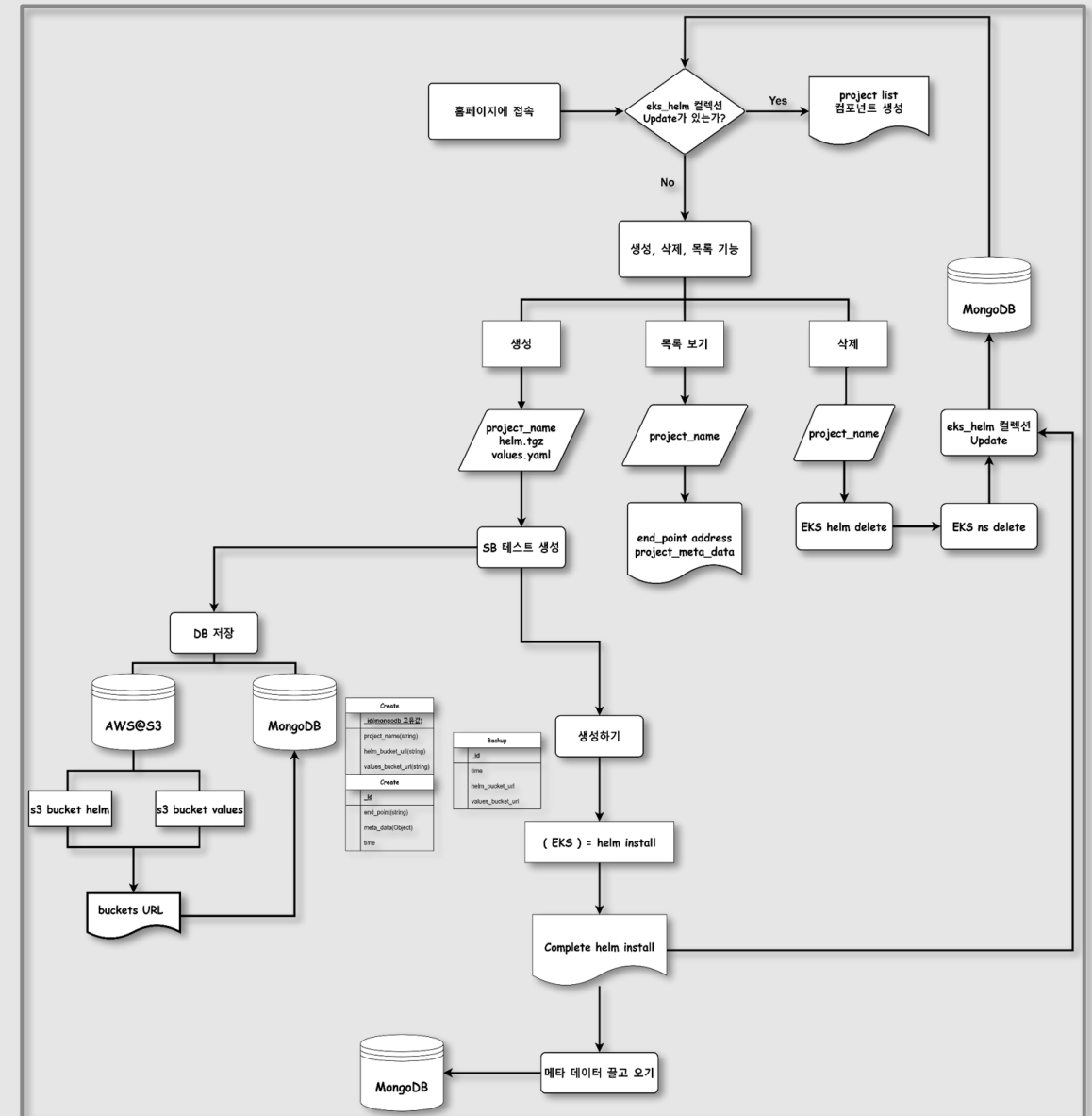
GET /api/v1/projects 프로젝트 목록 전체 조회

POST /api/v1/projects 새로운 샌드박스 테스트 프로젝트 생성하기

GET /api/v1/projects/{project_id} 단일 프로젝트 상세 내용 조회

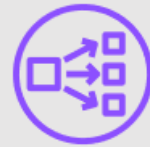
DELETE /api/v1/projects/{project_id} 단일 프로젝트 삭제

PUT /api/v1/projects/{project_id} 프로젝트 수정



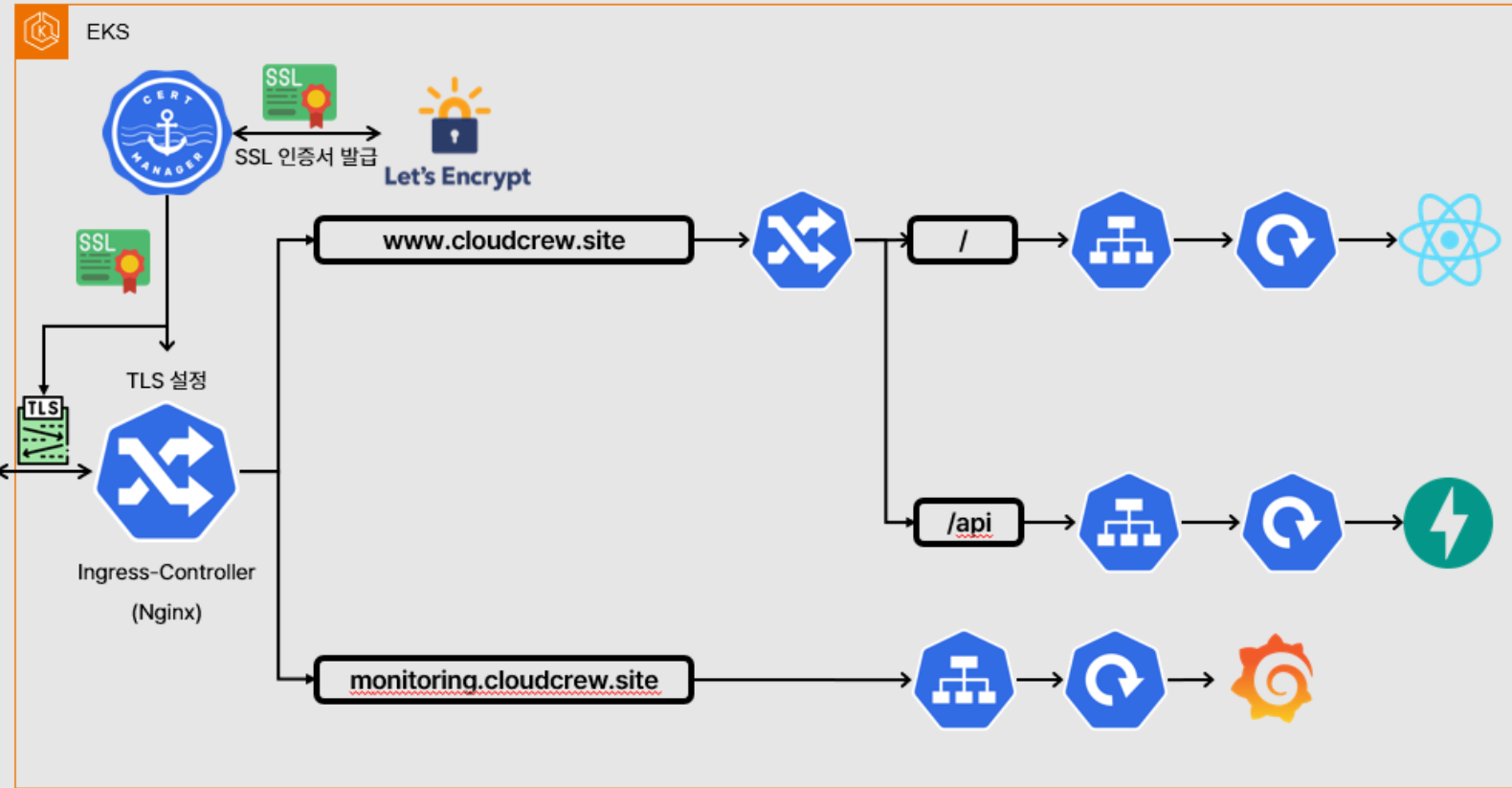
나의 역할 - HTTPS 배포

- Cert Manager 배포
- Let's Encrypt를 통해 SSL 인증서 발급
- 발급된 인증서를 Ingress Controller에 적용하여 TLS(HTTPS) 설정 구성
- HTTP 트래픽을 자동으로 HTTPS(301 리디렉션)로 강제 전환 설정



```
annotations:
  kubernetes.io/ingress.class: "nginx" # 사용할 Ingress Controller
  nginx.ingress.kubernetes.io/auth-type: basic
  nginx.ingress.kubernetes.io/auth-secret: basic-auth
  nginx.ingress.kubernetes.io/auth-realm: 'Authentication Required - cloudcrew'
  cert-manager.io/cluster-issuer: letsencrypt-prod
  nginx.ingress.kubernetes.io/ssl-redirect: "true"
  nginx.ingress.kubernetes.io/force-ssl-redirect: "true"

spec:
  tls:
  - hosts:
    - cloudcrew.site
    secretName: cloudcrew-tls
  - hosts:
    - monitoring.cloudcrew.site
    secretName: monitoring-cloudcrew-tls
```



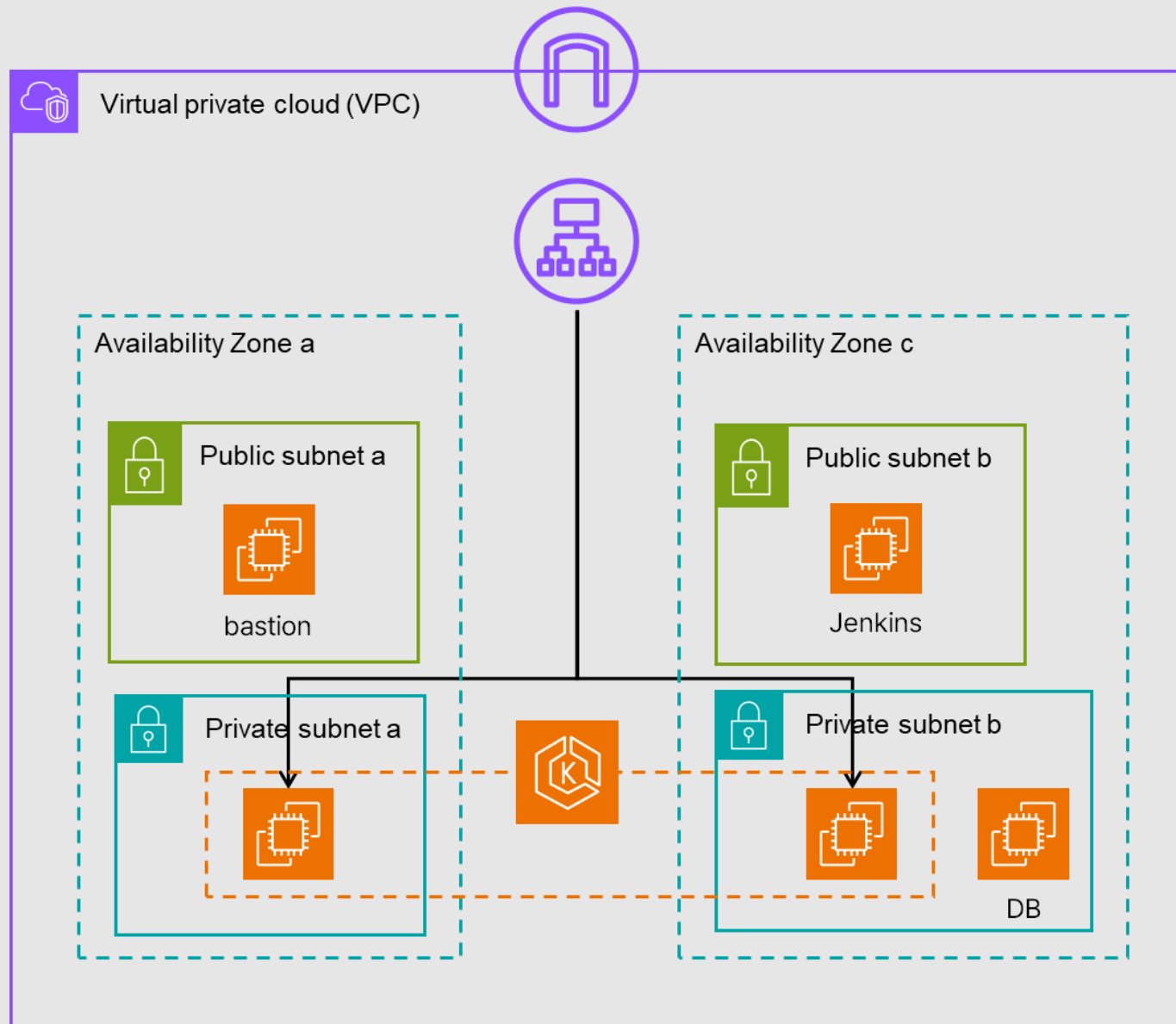
트러블 슈팅 - 문제 및 원인

문제 : 단일 EKS에 서비스 구성

- PoC와 기능 테스트를 위한 EKS 비용 건적은 오직 EKS를 하나만 사용하는 것입니다.
- 따라서 **단일 EKS에 서비스를 구성**해야 했습니다.
- 웹 서비스와 서비스 요청의 결과로 Helm 프로젝트를 배포해야 하기 때문에 격리된 환경인 **컨테이너에서 호스트 영역의 권한을 가지는 방향을 고려**했습니다.

원인 : Pod의 과도한 권한

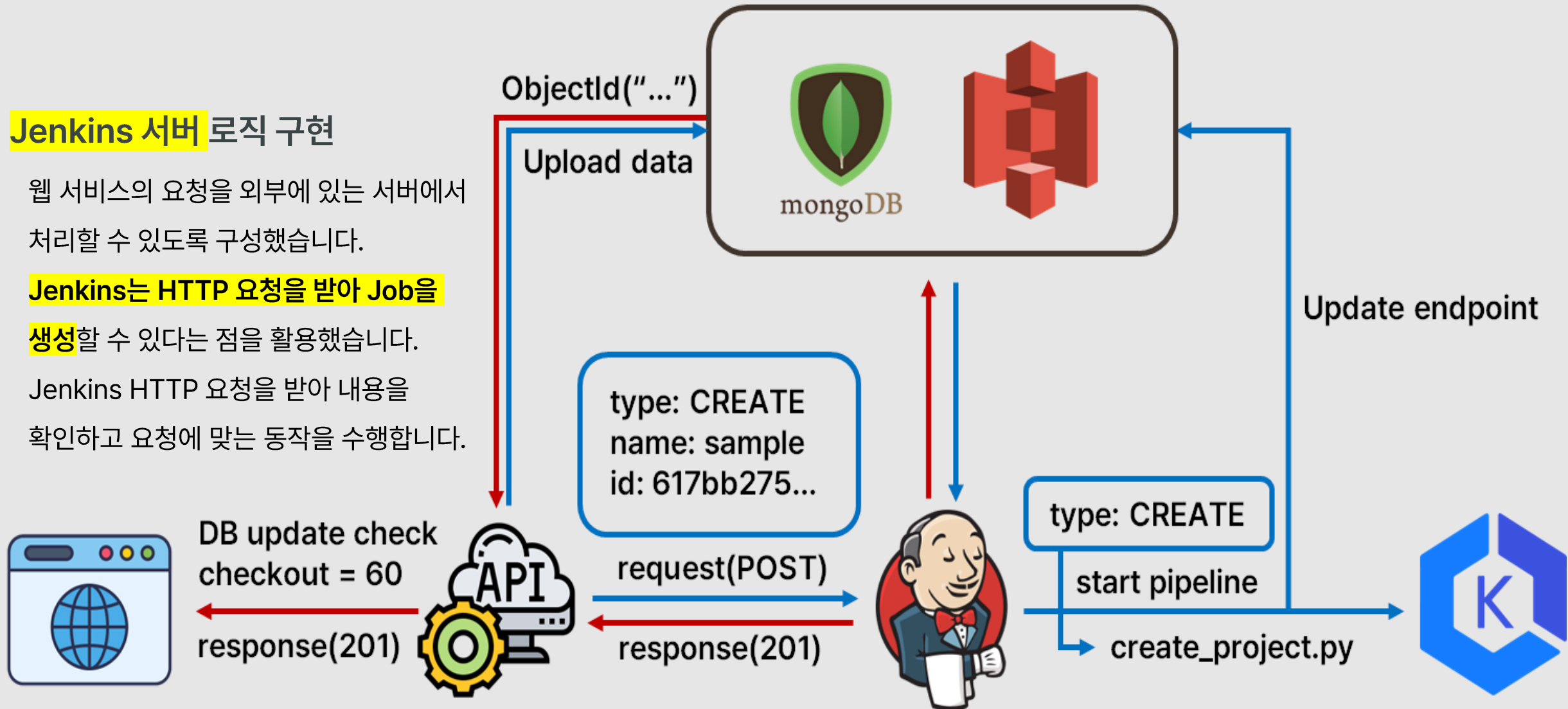
- Pod에 높은 수준의 권한을 부여하거나, Pod 이미지에 AWS EKS에 대한 자격 증명 파일을 포함하는 방법으로 구현할 수 있었습니다.
- 하지만, 이것은 Pod에 과도한 권한을 부여하고 **보안적으로 좋지 않은 선택**이었습니다.



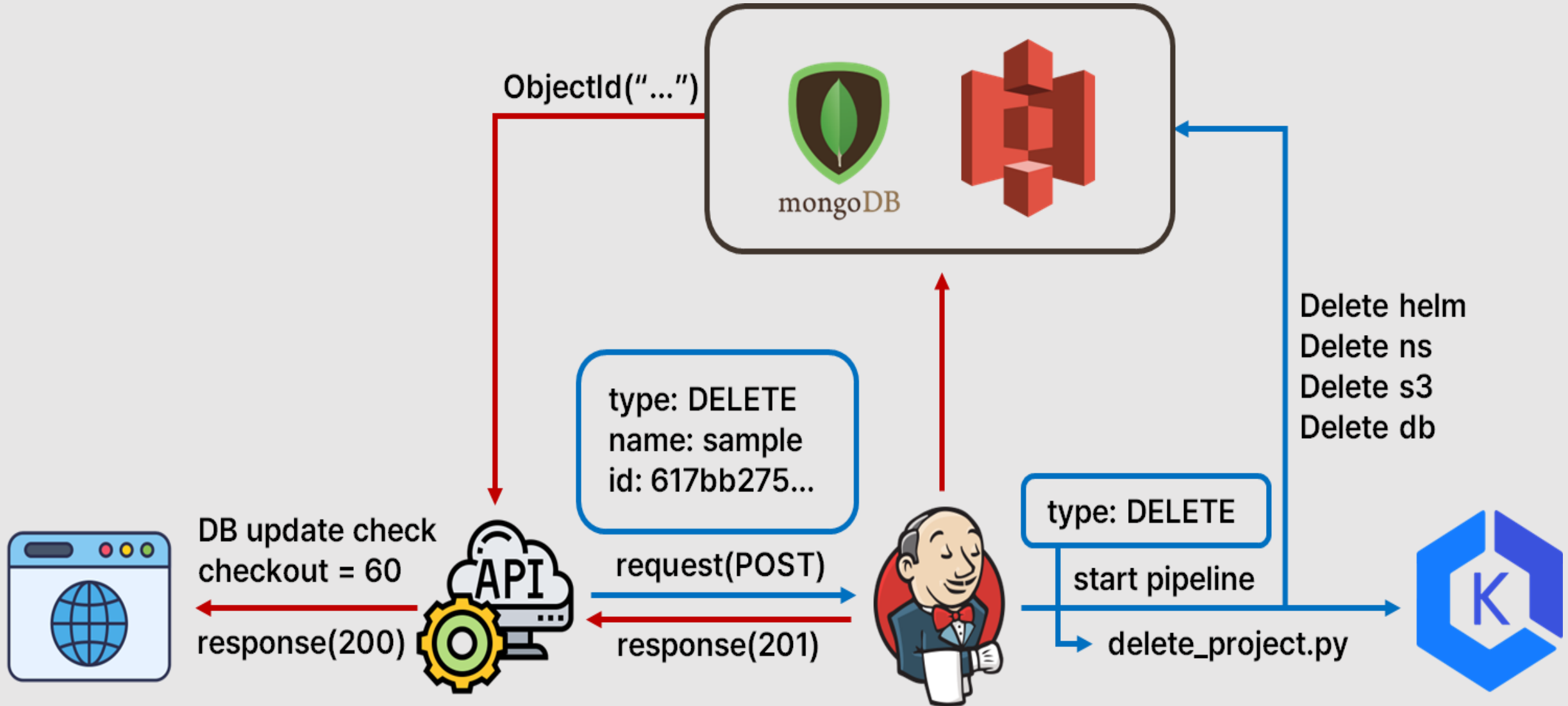
트러블 슈팅 - 해결 방법

Jenkins 서버 로직 구현

- 웹 서비스의 요청을 외부에 있는 서버에서 처리할 수 있도록 구성했습니다.
- **Jenkins는 HTTP 요청을 받아 Job을 생성**할 수 있다는 점을 활용했습니다.
- Jenkins HTTP 요청을 받아 내용을 확인하고 요청에 맞는 동작을 수행합니다.



트러블 슈팅 - 해결 방법



트러블 슈팅 - 해결 방법

