



# Goorm Bank

## 문제 있는 은행장

자격증 기출 문제 공유 서비스

팀명 : 구름은행 3지점

팀장 : 박정환

팀원 : 권순국, 설희관, 신흥석, 이종은, 정수철, 정우철

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## 목 차

<b>1. 프로젝트 개요</b>	<b>3</b>
1.1 프로젝트 목표	3
1.2 프로젝트 일정	3
1.3 기대효과 (AS-IS - TO-BE)	3
<b>2. 프로젝트 팀 구성 및 역할</b>	<b>4</b>
2.1 팀원 및 역할	4
2.2 프로젝트 환경	5
2.3 프로젝트 관리 방안	5
<b>3. 프로젝트 진행 프로세스 및 결과</b>	<b>6</b>
3.1 Frontend	6
3.2 Backend	17
3.3 CI/CD	21
3.4 Logging & Monitoring	28
3.5 AWS	31
3.6 AutoScaling	31

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

**4. 자체 평가 및 보완 37**

**5. 참고자료 40**

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## 1. 프로젝트 개요

### 1.1 프로젝트 목표

- 시험기간에 한시적인 트래픽 수요로 인한 서비스 불안정을 개선하고 비수기에는 리소스를 최소화하여 비용절감하는 것을 서브 목표로 하고, EKS를 통해 클러스터를 구축 후 오토스케일링, 로드밸런싱을 통한 부하 분산을 경험하는 것을 메인 목표로 합니다.

### 1.2 프로젝트 일정

	1주차		2주차		3주차		4주차		5주차	
주제 선정										
서비스 기획 및 개발										
EKS 클러스터 구성										
CI/CD 구축										
로그 & 모니터링 구축										
보고서 작성										

[표 1-1] 프로젝트 일정

### 1.3 기대효과 (AS-IS - TO-BE)

#### 1.3.1 현황 (AS-IS)

- 시험전에는 트래픽이 급상승, 시험후에는 트래픽이 급하강
- 익명 게시판 위주의 서비스
- 유동적 리소스 활용이 어려움
- 트래픽 증가시 대응이 어려움

#### 1.3.2 개선 및 기대효과 (TO-BE)

- 클라우드 및 K8S를 활용하여 리소스 관리
- 오토스케일링으로 실시간 리소스 대응
- SNS 회원가입 & 로그인으로 쉬운 접근성 제공

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

- 오토스케일링, 로드밸런싱을 통한 부하분산 효과 기대
- 성수기, 비수기 기간에 따른 리소스 조절로 비용 절감 목표

## 2. 프로젝트 팀 구성 및 역할

### 2.1 팀원 및 역할

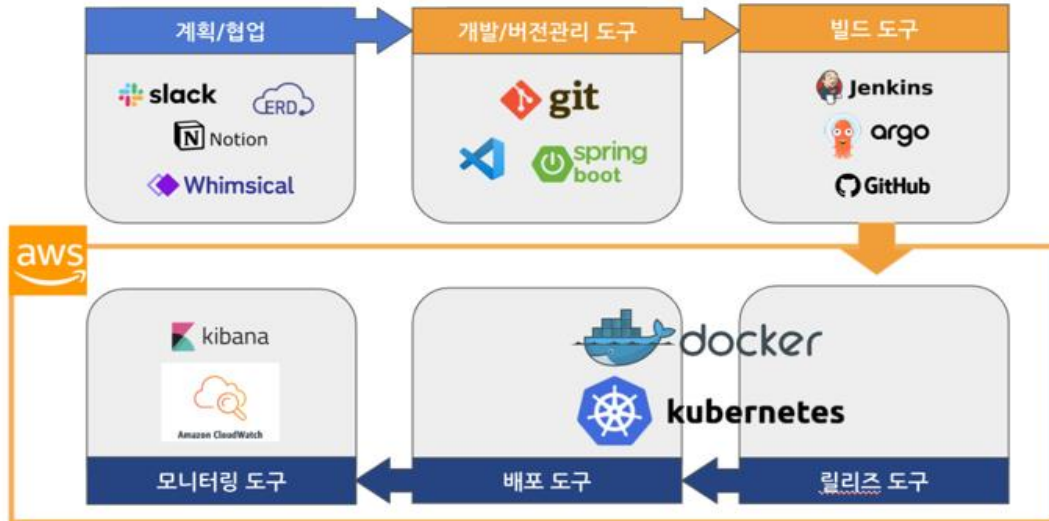
이름	역할
권순국	- Amazon OpenSearch Service - EKS 클러스터 구성
설희관	- EKS 클러스터 구성 - Frontend 서비스 개발 - 오토스케일링
신흥석	- 오토스케일링 - EKS 클러스터 구성
이종은	- Amazon CloudWatch 구성 - 로그 스토리지 구축 관리
정우철	- 젠킨스, ArgoCD 등 CI/CD - EKS 클러스터 구성
정수철	- 젠킨스, ArgoCD 등 CI/CD - EKS 클러스터 구성
박정환	- Backend 서비스 개발 - RDS를 이용한 DB 관리 - EKS 클러스터 구성

[표 2-1] 팀원 및 역할

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## 2.2 프로젝트 환경

아래 [그림 2-2]의 기술 스택들을 이용하여 프로젝트를 진행하였다.



[이미지 2-1] 프로젝트 환경

## 2.3 프로젝트 관리 방안



[이미지 2-2] 프로젝트 관리 방안

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

### 3. 프로젝트 진행 프로세스 및 결과

#### 3.1 Frontend

##### 3.1.1 기술 스택

프론트엔드(Front-End) 개발을 위해 사용한 기술 스택은 리액트(React)이다. 리액트를 사용하게 된 이유는 다음과 같다. 우선 사용자마다 다른 화면을 보여주게 되는 동적 웹 페이지 SPA(Single Page Application)을 개발하기에 가장 적합하기 때문이다. Virtual DOM을 사용하여 동적으로 변화하는 부분의 컴포넌트만 찾아내어 바꾸게 됨으로 효율적인 렌더링이 가능하게 된다. 또한 컴포넌트 단위의 개발이 가능하고 컴포넌트를 재사용할 수 있는 재사용성도 높아지기 때문에 효율적인 개발이 가능하다. 이러한 이유들로 인해 리액트를 프론트엔드 기술 스택으로 결정하게 되었다.

##### 3.1.2 Home 화면



[이미지 3-1] Home Page 1

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

Typelt

Pricing

Demos

Flavors

Build an Animation

Documentation

Q

Purchase a License

The most versatile  
typing utility on the  
*planet.*

```
new TypeIt("#hero", {
  speed: 50,
  startDelay: 900,
})
.type("the mot verst", { delay: 100 })
.move(-8, { delay: 100 })
.type("s", { delay: 400 })
.move(5, { delay: 100 })
.type("START", { delay: 100 })
```

[이미지 3-2] Typeit JS

홈 화면의 상단에는 유저들이 서비스를 이용하며 길을 잃지 않도록 네비게이션바 (Navigation Bar) 컴포넌트를 배치해두었다. 네비게이션 바 아래로는 우리의 서비스 명을 포함한 "Goorm Bank3 Let's Solve Problem!" 라는 문구와 서비스 로고(Logo)를 배치하였다. 해당 문구는 가시적으로 효과를 주기 위해 오픈 소스인 'Typeit.js'를 이용하여 타이핑 되는 형태로 출력되도록 하였다.



#### Service Introduction

"Goorm Bank 서비스는 다양한 국가 자격증 시험, 일반 시험에 대한 기술문제를 제공하고 있습니다. 문제를 제공할뿐만 아니라 각 유저들의 문제해결 내역을 대시보드(DashBoard)로 제공합니다."

Goorm Bank

#### Free download



© COPYRIGHT © 2022 Goorm Bank ALL RIGHT RESERVED  
Office in USA THE CORPORATION 139 CENTRE ST STE 304 NEW YORK, NY 10013-4554  
Contact us: goormbank@gmail.com

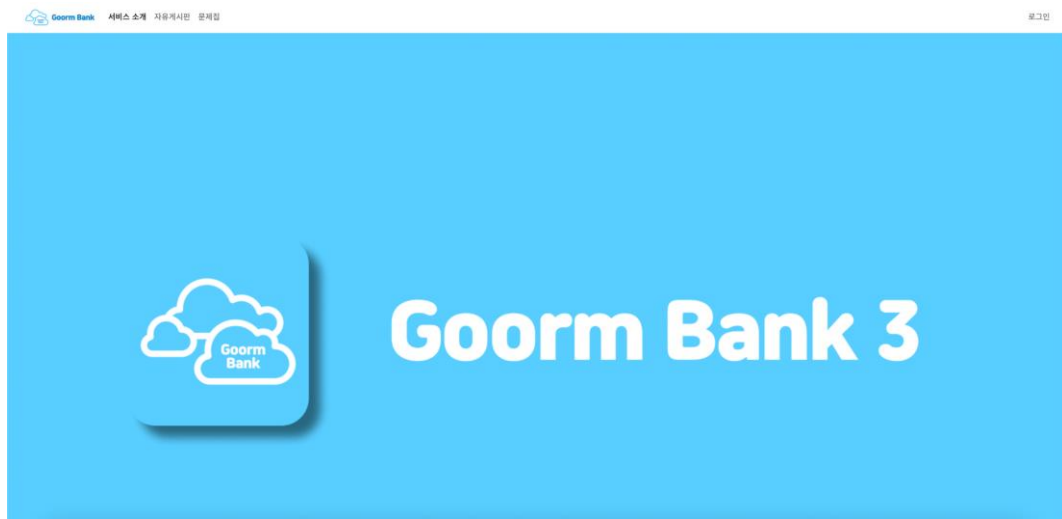


	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

[이미지 3-3] Home Page 2

서비스 명을 포함한 문구와 로고 아래에는 서비스에 대한 간단한 소개와 추후에 앱 서비스를 개발했을 때 앱 다운로드 링크로 리다이렉션 할 수 있도록 Google Play 와 App Store 버튼을 두었다. 그 아래로는 웹 애플리케이션의 최하단을 담당하는 푸터(Footer)를 두어 가상으로 회사 위치, 회사 이메일 등의 정보를 작성해두었다.

### 3.1.3 서비스 소개



[이미지 3-4] 서비스 소개 1

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-5] 서비스 소개 2

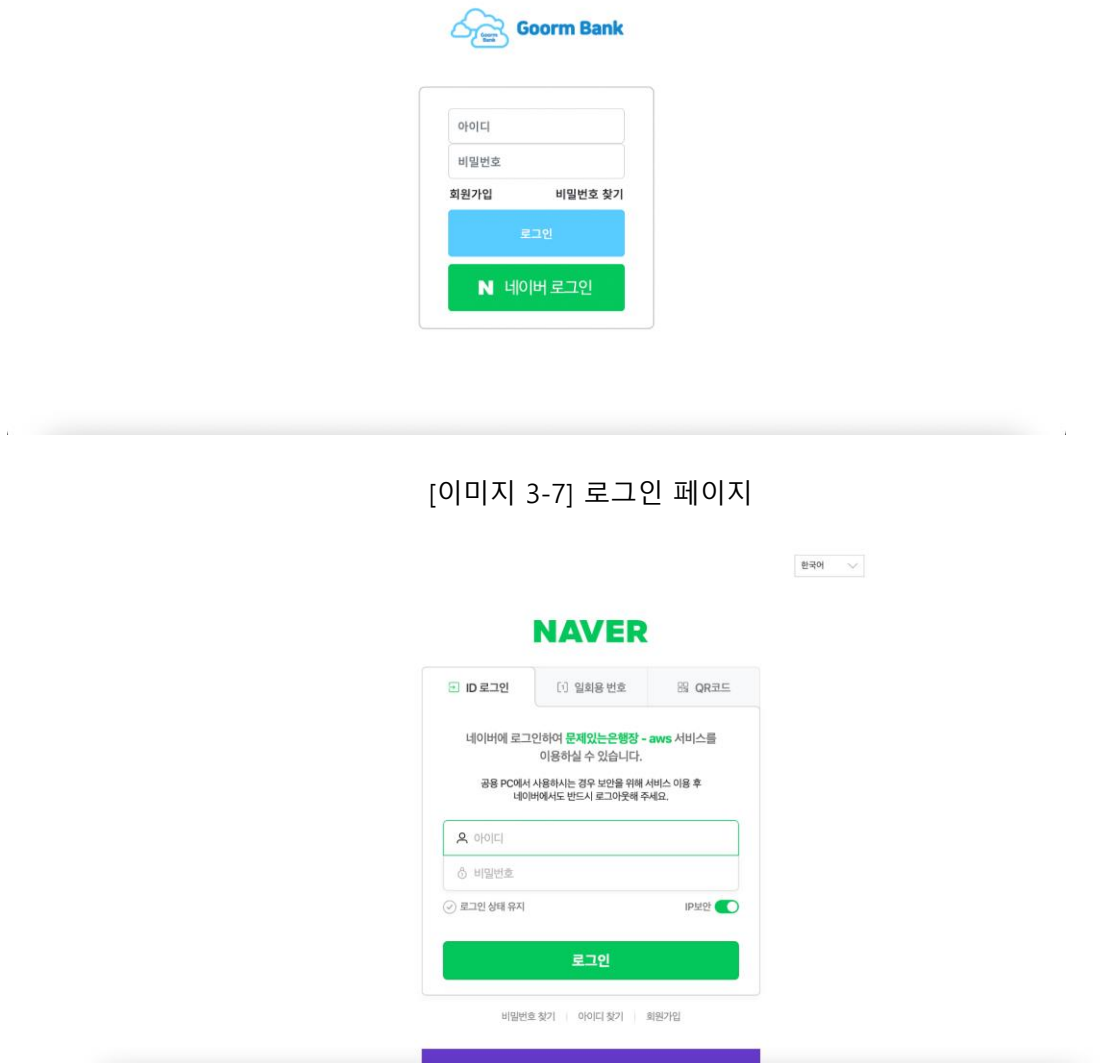


[이미지 3-6] 서비스 소개 3

서비스 소개 페이지(\* /service)로 접속하면 [이미지 3-5] 과 [이미지 3-6]에서 보는 바와 같이 우리 서비스의 Brand Identity와 서비스를 하는 목적에 대한 정보를 제공하고 있다.

### 3.1.3 로그인(Login)

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

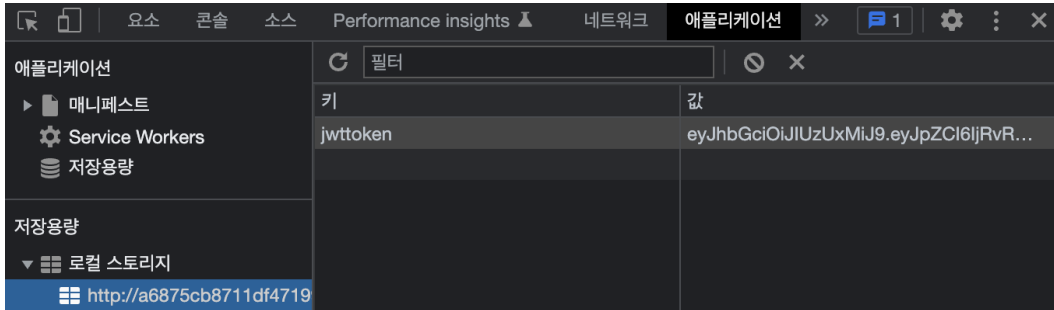


[이미지 3-7] 로그인 페이지

[이미지 3-8] 네이버 로그인 페이지

서비스 이용시 필요한 로그인 화면을 위의 [이미지 3-7]와 같이 구성하였다. 서비스 자체에서 제공하는 유저 서비스는 추후에 구현할 예정이기 때문에 현재 회원가입, 비밀번호 찾기는 아직 동작하지 않는다. 서비스 테스트를 위해 자체 유저 서비스를 대신해 소셜 로그인 기능을 우선적으로 구현해보았다.

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-9] 로컬 스토리지에 저장된 JWT 토큰

네이버 소셜 로그인 동작 방식은 다음과 같다. 클라이언트 측에서 네이버 로그인 요청을 할 경우 백엔드(Back-End) 측으로 JWT 토큰을 요청한다. 요청에 대한 응답으로 받은 JWT 토큰을 로컬 스토리지에 저장하여 로그인 여부를 확인하고 유저 정보를 가져온다.



[이미지 3-10] 비로그인 시 댓글 작성 화면

<PrivateRoute> 컴포넌트 코드를 작성하여 유저 인증을 하지 않았을 경우 댓글 작성 기능을 이용할 수 없고 프로필 또는 글쓰기 버튼을 클릭할 경우 로그인 페이지로 리다이렉션(Redirection) 되도록 구현하였다.

### 3.1.4 자유게시판

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	


서비스 소개 자유게시판 문제집


설화관님 로그아웃

### 자유게시판

No.	제목	작성일	작성자
5	7급[0]	2022-04-14	각각로로로
6	덕분에 감사합니다.[0]	2022-03-04	엑션 드라이버
7	감사합니다[0]	2022-02-14	핵밀
8	좋은 사이트 감사합니다[0]	2021-08-28	감사합니다
9	이거 어디서 관리 하는거임??[1]	2021-08-04	토목
10	간호조무사 기술문제 열람할수 있게해줄수 있나요?[0]	2021-06-25	짬요
11	추가 가능할까요?[0]	2021-06-22	oo
12	완전 절실히 필요한데.....[1]	2021-05-14	아름이
13	없어졌네요?[2]	2021-05-13	갑자기 없어졌어요
14	채팅하고 합력인지 과박인지 그것도 알려줬으면합니다[1]	2021-04-04	123431

1 2 3

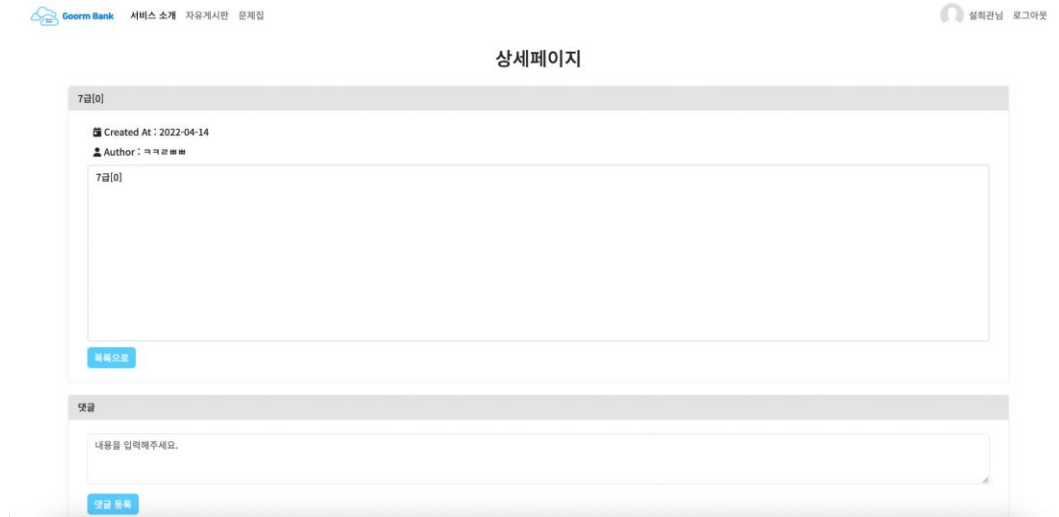
[이미지 3-11] 자유게시판 글 목록

No.	제목	작성일	작성자
26	test34	2022-10-26	test4
25	test3	2022-10-26	test3
24	test	2022-10-26	test2
22	test	2022-10-26	sss

[이미지 3-12] 자유게시판 검색 기능

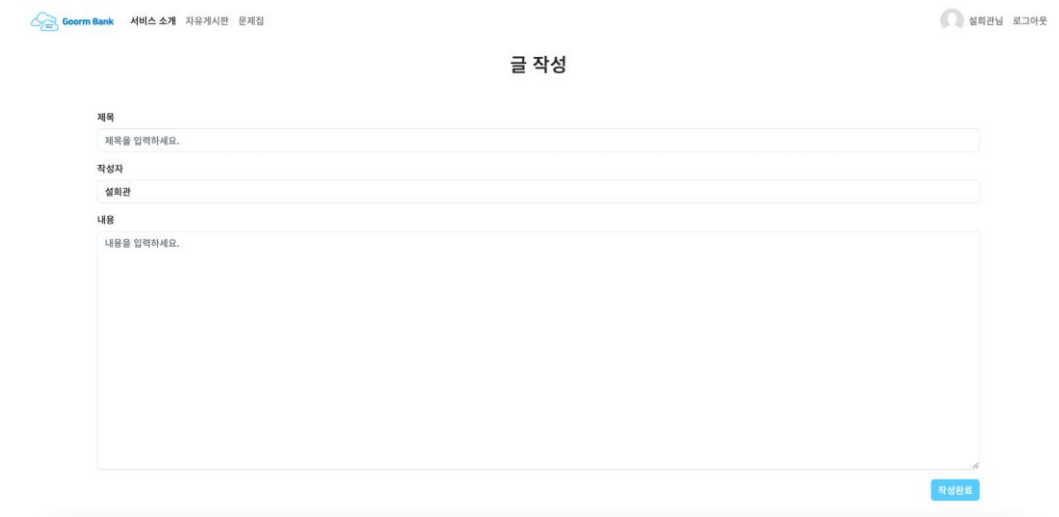
자유게시판의 글 목록은 비로그인 시에도 확인이 가능하다. 우측 상단에는 제목을 기반으로 게시글을 검색할 수 있는 검색 기능을 구현하였으며 하단에는 게시물 10개 단위로 페이지 이동을 할 수 있도록 페이지네이션(Pagination) 기능을 구현하였다.

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-13] 자유게시판 글 상세 페이지

자유게시판의 글 상세 페이지 또한 비로그인 시에도 확인할 수 있도록 하였다. 글 상세페이지에는 게시글의 작성일, 작성자, 작성 내용, 댓글을 볼 수 있다.



[이미지 3-14]. 자유게시판 글 작성 페이지

로그인한 유저는 글 작성 페이지에 접속하여 게시글을 작성할 수 있다. 게시글 작성 완료 시 자

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

유게시판 글 목록을 보여주는 페이지로 리다이렉션 하도록 하였다.

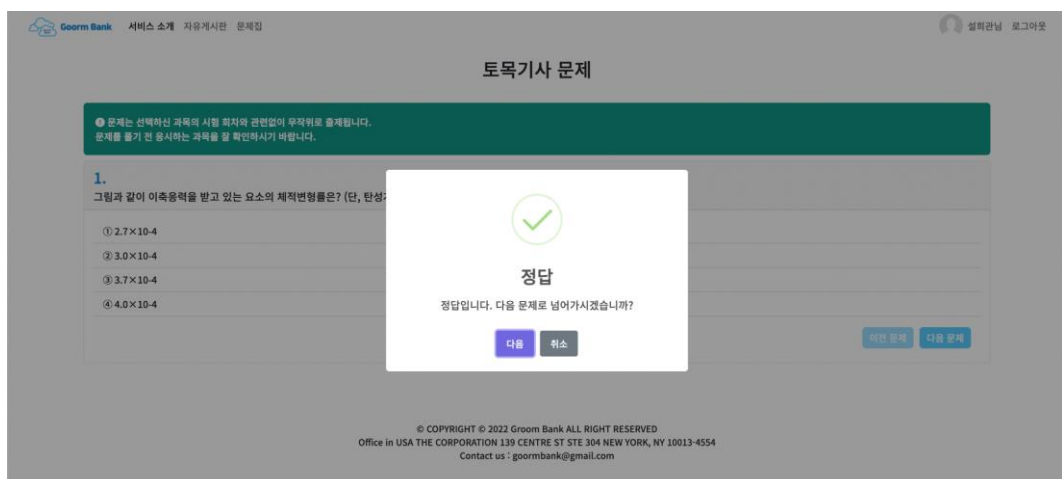
### 3.1.5 문제집



[이미지 3-15]. 문제집 페이지

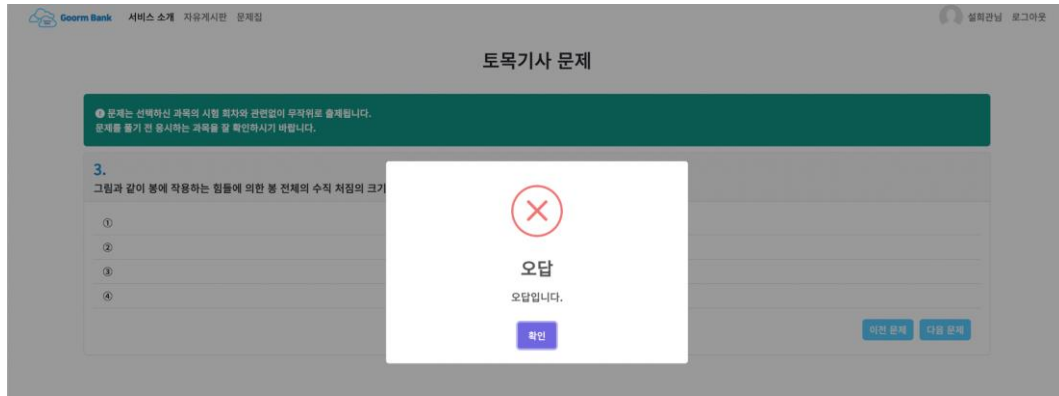
[이미지 3-16] 시험일정 별 토목기사 문제집 목록

문제집 페이지로 접속하면 다양한 타입의 문제집을 선택할 수 있도록 하였으며 문제집 클릭 시 시험 일정별로 문제집 목록을 확인할 수 있다. 또한 접속 날짜를 기준으로 기한이 지났을 경우 '시험 완료' 문구를 보여주며 기한이 지나지 않았을 경우 '시험 전' 이라는 문구를 보여주도록 하였다.



	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

[이미지 3-17] 문제집에서 정답을 선택했을 경우



[이미지 3-18] 문제집에서 오답을 선택했을 경우

앞서 봤던 문제집 목록 페이지에서 특정 시험일정의 문제집을 클릭하면 실제 문제 순서와 상관 없이 무작위인 순서로 문제를 보여준다. 정답 선택 시 [이미지 3-1-18]과 같이 다음 문제로 리다이렉션 할 수 있는 '다음' 버튼과 함께 알림(Alert) 창을 띄운다. 오답 선택 시 [이미지 3-1-19]와 같이 문제를 다시 풀 수 있도록 '확인' 버튼과 함께 알림 창을 띄운다.

### 3.1.6 프로필 페이지



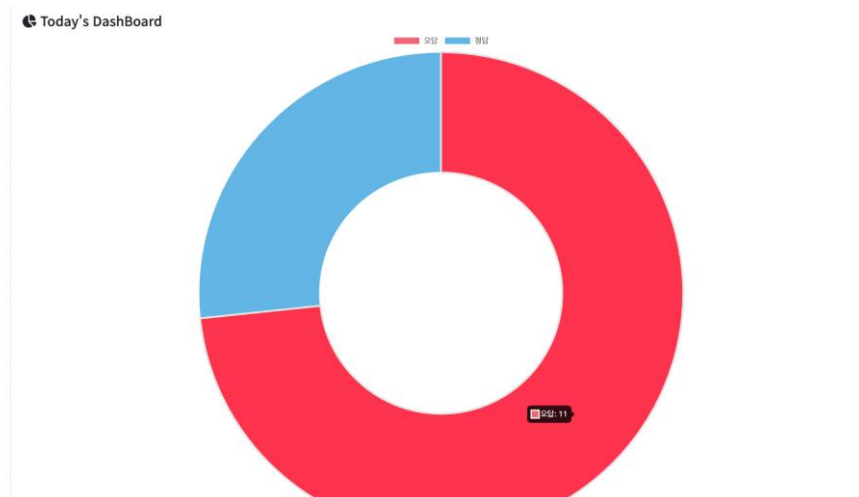
[이미지 3-19] 프로필 페이지 1

로그인 시 우측 상단에 로그인한 유저의 프로필 사진과 이름을 출력하도록 하였다. 프로필 버튼



	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

을 클릭할 경우 로그인한 유저의 프로필 사진, 이름, 별명, 이메일을 보여주도록 하였다. 프로필 사진, 이름, 별명, 이메일 정보는 로그인 시 받아온 JWT 토큰을 이용하여 백엔드(Back-End) API를 통해 가져올 수 있다.



[이미지 3-20] 프로필 페이지 2

프로필 페이지 하단에는 오늘 풀었던 문제에 대한 정답률과 오답률을 파이차트(Pie Chart)를 이용하여 보여주도록 하였다. 차트는 오픈 소스인 'Chart.js'를 이용하였다. 현재는 로컬 스토리지에 로그인한 유저에 대한 정답/오답률을 저장하고 있기 때문에 로그아웃 시 데이터가 사라진다. 추후에 DB와 연동하여 매일 정답/오답률 데이터가 갱신되도록 할 예정이다.

### 3.1.7 404 Not Found 페이지

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-21] 404 Not Found 페이지

404 Not Found 페이지를 제작하여 지정된 경로가 아닌 비정상적인 경로로 접속했을 경우 404 Not Found 페이지로 이동하도록 하였다.

## 3.2 Backend

### 3.2.1 JWT 이용한 네이버 간편가입 & 로그인 기능 개발

#### 3.2.1.1 회원 가입 및 회원 전용 기능 제공에 활용

#### 3.2.1.2 JWT(JSON Web Tokens)를 Front에 발급, 회원정보 조회 가능한 API 제공

### 3.2.2 데이터 모델링 & API 개발

#### 3.2.2.1 Spring Boot + JPA + Maven + JWT + Swagger 활용하여 개발 진행

### 3.2.3 기사 시험 기준 회차별 공개 문제에 대해 데이터 생성

#### 3.2.3.1 시험일 회차 총 2700건 생성

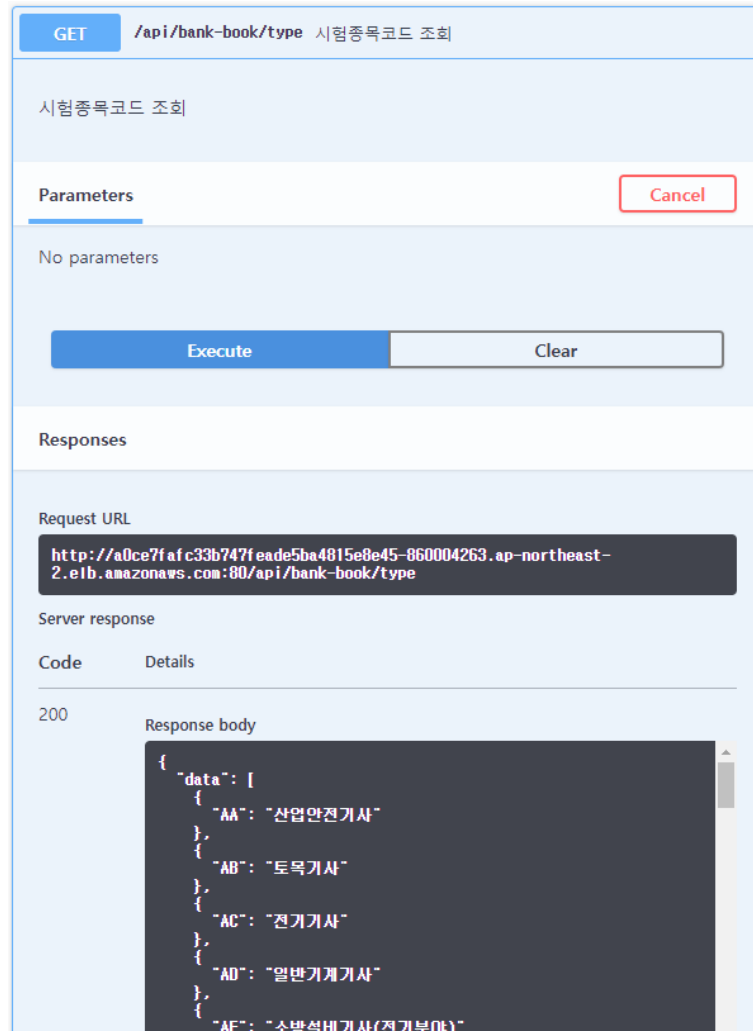
#### 3.2.3.2 시험 문제 총 27만건 생성

#### 3.2.3.3 시험 문제에 대한 저작권 문제 [\[참고자료 -10\] 참고](#)

### 3.2.4 Swagger-ui로 API 규격서 자동 생성 및 제공

#### 3.2.4.1 실행 예시

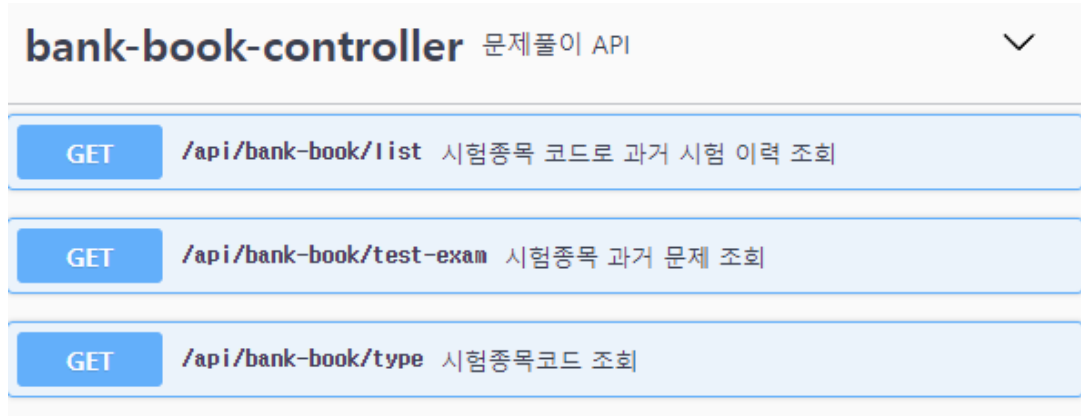
	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-22] Swagger 활용 API 규격서 제공

#### 3.2.4.2 문제풀이 API

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-23] 시험 종목에 대한 코드 관리, 시험문제 데이터 조회

#### 3.2.4.3 게시물 API



[이미지 3-24] Frontend용 게시판 API 개발

#### 3.2.4.4 회원 API

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

user-controller 회원 API

GET /api/user/info 회원정보 조회

토큰으로 회원정보 조회

Parameters
Cancel

Name	Description
<b>token</b> * required string (query)	Token <div>eyJhbGciOiJIUzUxMiJ9.eyJpZCI6IkdBdUxiRm'</div>

Execute Clear

Responses

Server response

Code	Details
200	Response body <pre>{   "data": [     {       "id": "BAuLbFeBdRenI-1qeA6raImzAJeC7NWC2nhc53tA1Q1"     },     {       "nickname": "     },     {       "profile_image": "https://ssl.pstatic.net/static/pwe/address/img_profile.png"     },     {       "email": "     },     {       "name": "     }   ] }</pre>

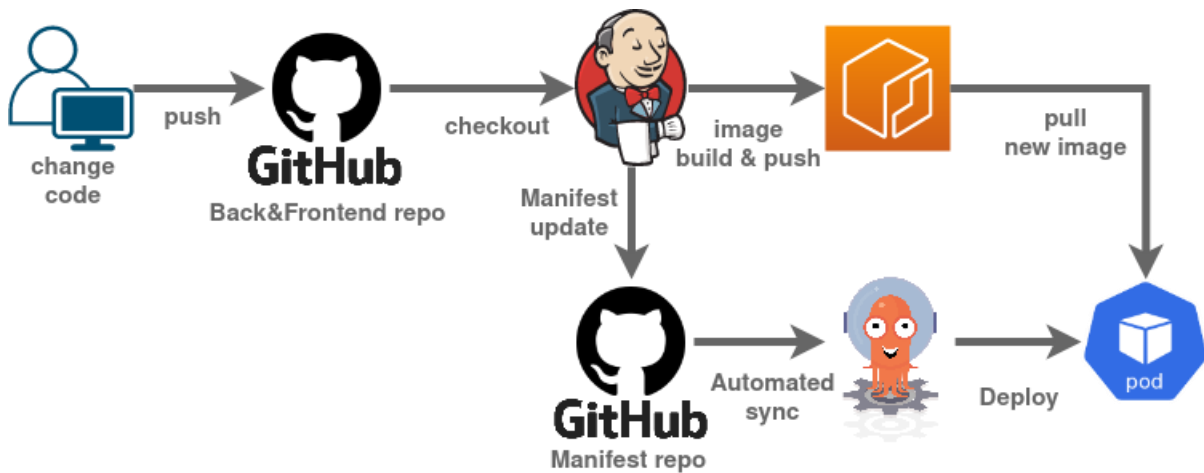
[이미지 3-25] JWT 활용 회원 조회 API

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

### 3.3 CI/CD

#### 3.3.1 구성

- CI/CD 구성은 아래 [그림 3-26] 과 같다.  
(제일 안정화 된 버전으로 구성함, Jenkins: 2.361.2버전, Argo CD: 2.4.15버전)
- code와 manifest를 관리하는 저장소를 구분하여 관리하도록 구성



[이미지 3-26] CI/CD configuration

#### 3.3.2 Jenkins

##### 3.3.2.1 구성

- Frontend 빌드 & 도커 이미지 생성(Dockerfile & Jenkinsfile)

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## Pipeline goormbank-frontend

### Stage View

		Declarative: Checkout SCM	Declarative: Tool Install	Webpack Build	Dockerizing	Manifest Update
Average stage times: (Average full run time: ~3min 12s)		890ms	118ms	2min 21s	45s	4s
#64	Nov 01 01:16 No Changes	824ms	122ms	12min 38s	47s	5s
#63	Oct 31 01:16 1 commit	1s	122ms	33s	48s	4s
#62	Oct 30 01:16 2 commits	1s	104ms	38s	44s	4s

[이미지 3-27] 프론트엔드 CI 과정

- Backend 빌드 & 도커 이미지 생성(Dockerfile & Jenkinsfile)

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## Pipeline goormbank-backend

### Stage View

		Declarative: Checkout SCM	Maven Build	Docker build	ECR push	K8S Manifest Update
Average stage times: (Average <u>full</u> run time: ~39s)		811ms	22s	4s	2s	2s
#70	Nov 01 06:14 No Changes	775ms	25s	4s	4s	5s
#69	Nov 01 01:14 No Changes	751ms	25s	4s	374ms failed	54ms failed
#68	Oct 31 02:12 No Changes	779ms	25s	4s	3s	4s

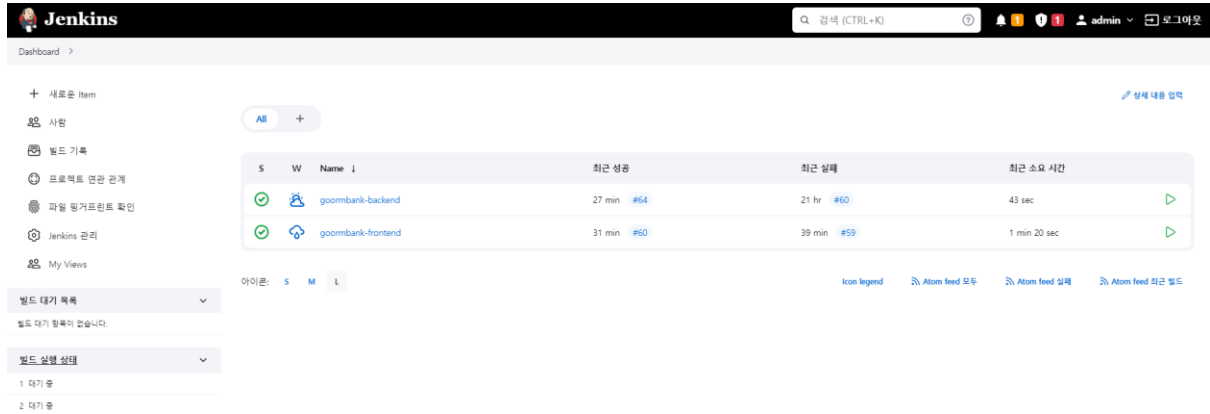
[이미지 3-28] 백엔드 CI 과정

3.3.2.2 git push 이벤트 발생시 image 생성, 일정 주기로 빌드 자동화

- H(10-30) 01 \* \* \* (UTC 기준 매일 새벽 1시 10~30분 사이  
(한국시간 오전 10시 10~30분) 주기로 실행
- 프론트엔드 : 1시 16분 30초에 빌드
- 백엔드 : 1시 14분 07초에 빌드



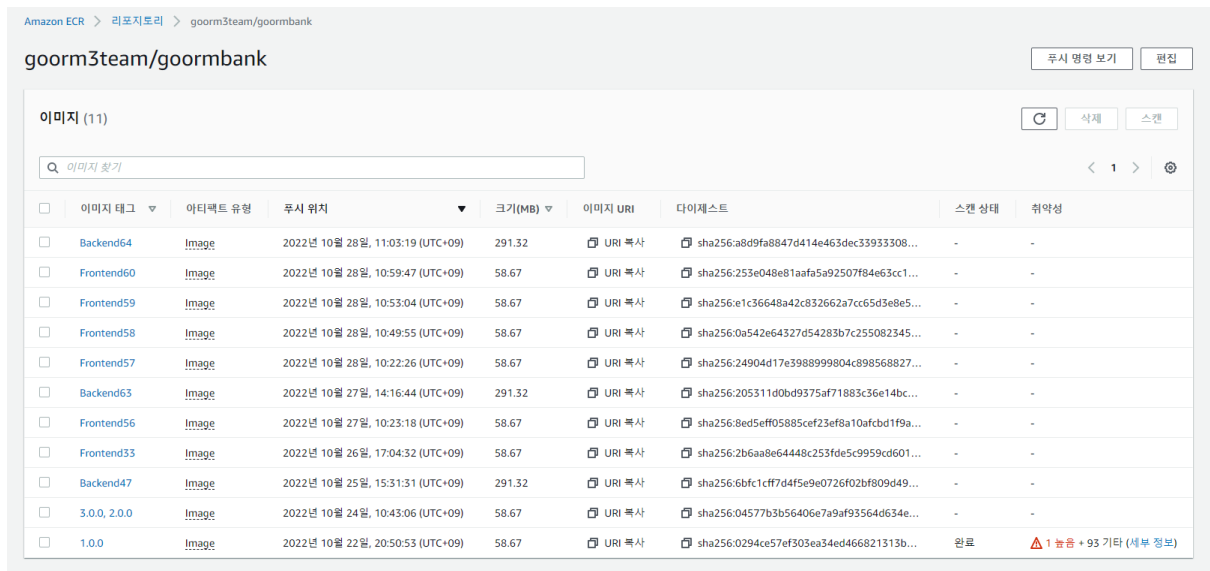
	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-29] Jenkins Dashboard

### 3.3.2.3 Amazon Elastic Container Registry (Amazon ECR)

- Jenkins에서 빌드를 하면, goorm3team/goormbank 레포지토리에  
백엔드, 프론트엔드 이미지가 업로드 됨.



[이미지 3-30] Jenkins에서 빌드한 Amazon ECR 레포지토리 이미지

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

- ECR 레포지토리에 이미지가 업로드 된 지 7일이 경과하면 만료되도록 수명 주기 규칙 설정

## 수명 주기 규칙 생성

### 수명 주기 규칙 구성

이 규칙을 수명 주기 정책으로 적용하면 즉시 평가하게 됩니다. 먼저 이 규칙으로 모의 실험을 하는 것이 좋습니다.

#### 규칙 우선 순위

고유한 규칙 우선 순위를 정하십시오. 정책의 규칙에서 값이 순차적일 필요는 없습니다.

1

#### 규칙 설명

수명 주기 방식에 대해 설명하십시오.

이전 이미지 제거

#### 이미지 상태

이미지에 태그가 있는지 여부를 나타냅니다.

- ☐ 태그가 지정됨
- ☒ 태그가 지정되지 않음
- ☐ 모두 선택

#### 매치 범위

이미지에 적용시킬 카운트 타입을 지정하십시오. "countType"을 "imageCountMoreThan"으로 설정한 경우, "countNumber"도 지정하여 리포지토리에 존재하는 이미지 수의 한도를 정하는 규칙을 만들어야 합니다. "countType"을 "sinceImagePushed"로 설정한 경우, "countUnit"과 "countNumber"도 지정하여 리포지토리에 존재하는 이미지의 시간 한도를 정해야 합니다.

이미지가 푸시된 후

7

일

#### 규칙 작업

지원되는 유일한 값이 만기됐습니다. 이미지가 저장소에서 삭제될 것입니다. 이 작업은 실행 취소할 수 없습니다.

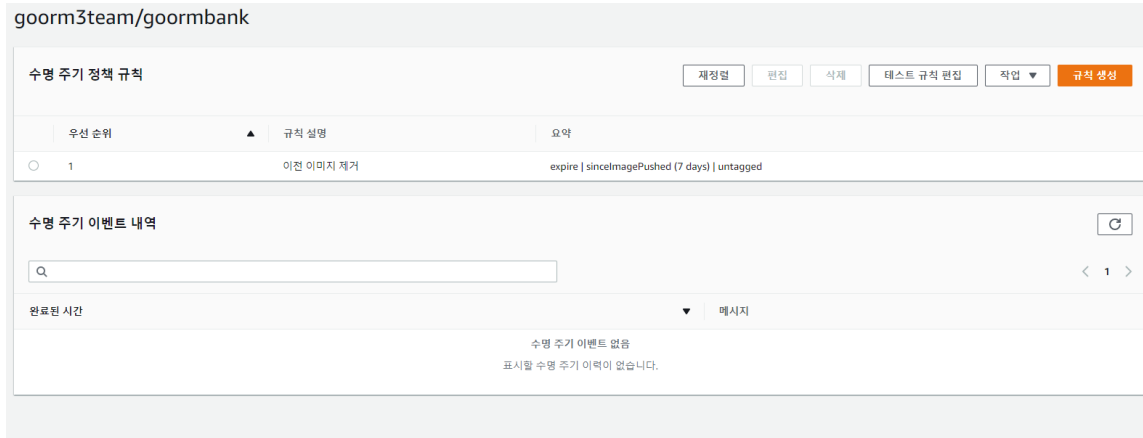
expire

취소

저장

[이미지 3-31] Amazon ECR 레포지토리 이미지 수명 주기 규칙 설정

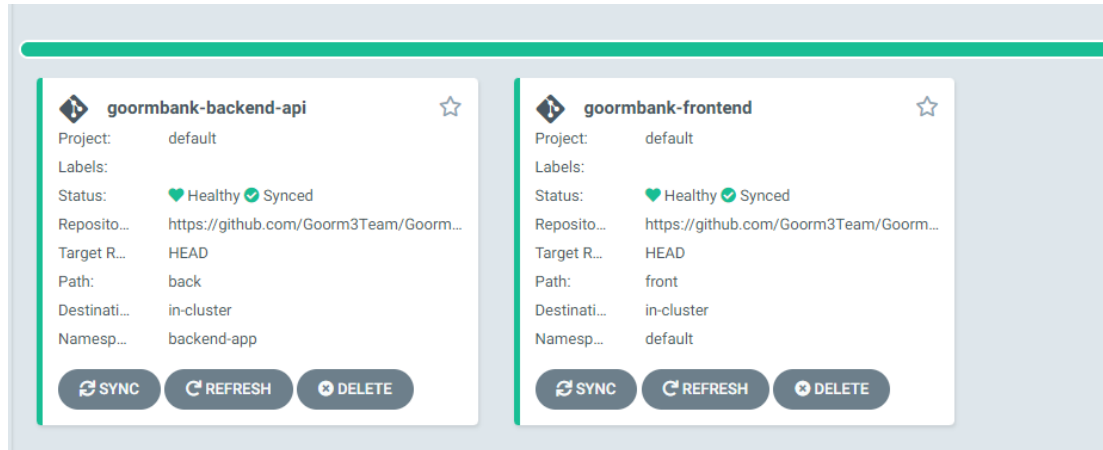
	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-32] Amazon ECR 레포지토리 이미지 수명 주기 규칙 설정

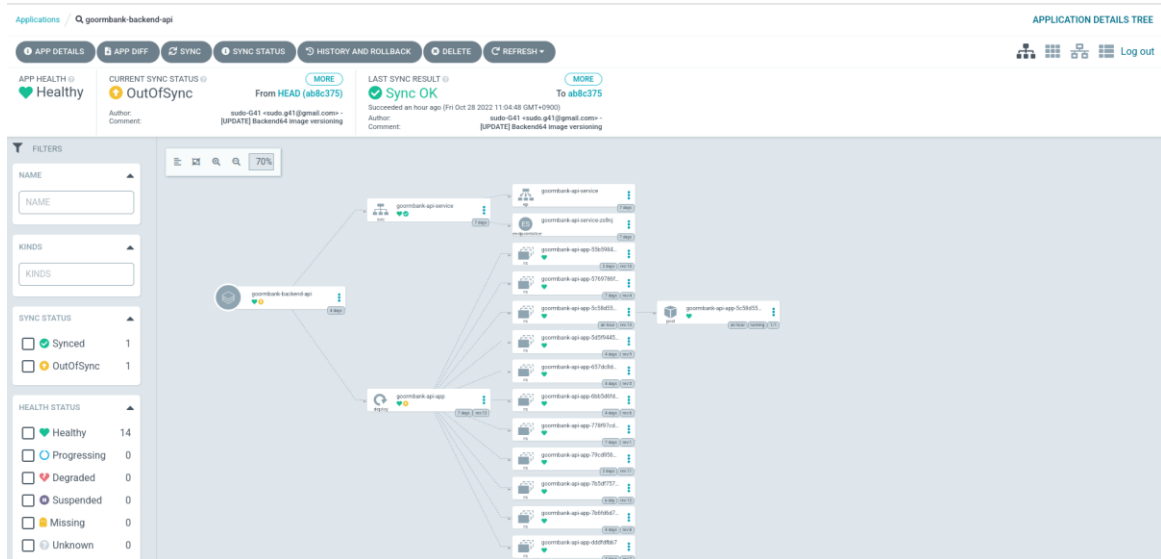
### 3.3.3 Argo CD

#### 3.3.3.1 Git Manifest Repo에 있는 백엔드, 프론트엔드 각 Manifest 파일을 통해서 자동화 배포 관리



[이미지 3-33] Argo CD Application

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-34] Argo CD Application 배포 현황 (백엔드)

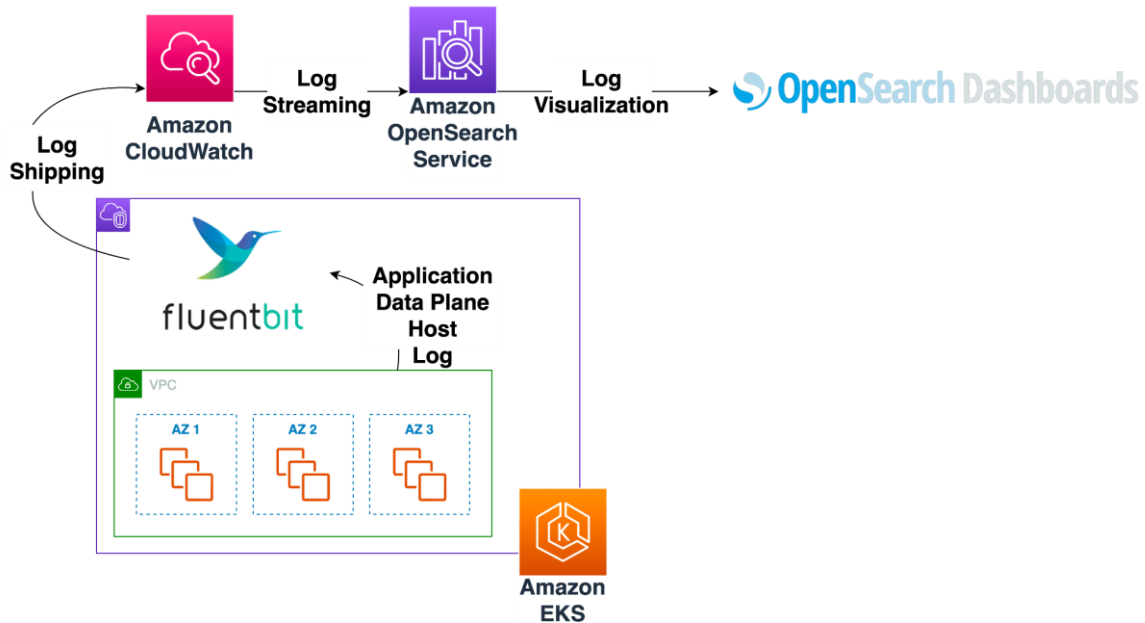


[이미지 3-35] Argo CD Application 배포 현황 (프론트엔드)

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

### 3.4 Logging & Monitoring

#### 3.4.1 Configuration



[이미지 3-36] Logging & Monitoring Configuration

Logging과 Monitoring을 위한 구성은 [그림 3-36]과 같다. Amazon EKS 내부에서 동작 중인 클러스터 로그를 fluent bit으로 수집한다. 수집된 로그는 Amazon CloudWatch에 정의된 각 로그 그룹에 전달되고 로그 그룹은 OpenSearch에 생성된 도메인에 대한 구독필터를 생성하여 로그를 해당 도메인으로 스트림한다. Amazon OpenSearch Service에서는 스트림된 로그를 검색, 분석, 시각화를 지원하며 OpenSearch Dashboards에서 인덱싱된 로그를 인덱스 패턴에 따라 시각화 할 수 있다.

#### 3.4.2 CloudWatch

Amazon CloudWatch는 AWS 리소스와 Amazon EKS 클러스터 전반적인 상태, 마스터 노드, 데이터 노드 성능 지표를 모니터링하는 기능과 로그 그룹을 활용하여 EKS 클러스터에서 발생하는 로그를 수집 및 수집된 로그를 Amazon OpenSearch Service로 스트림하는 기능을 제공한다. EKS 클러스터의 application, dataplane, host에 정

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

보에 대한 로그 그룹을 생성하여 각 항목에 대해 로그를 분리하여 수집 하고 Opensearch service로 로그를 전송 하는 람다에 대한 로그 그룹이 별도로 생성되어 있다.



[이미지 3-37] Amazon CloudWatch - Amazon EKS Cluster Performance Monitoring

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

로그 그룹	보존	구독 필터
/aws/containerinsights/eks-goormbank-cluster/application	3개월	slowquery
/aws/containerinsights/eks-goormbank-cluster/dataplane	3개월	slowquery
/aws/containerinsights/eks-goormbank-cluster/host	3개월	slowquery
/aws/lambda/LogsToElasticsearch_slowquery	만기 없음	

[표 3-1] 로그 그룹

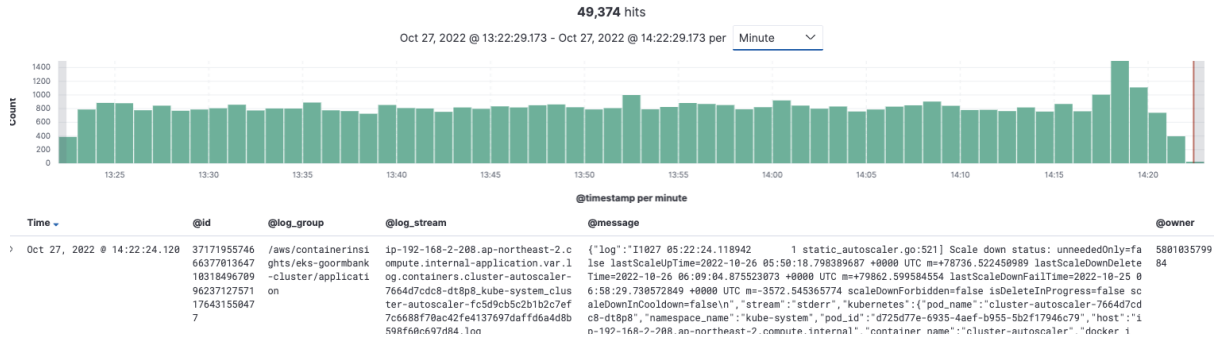
### 3.4.3 OpenSearch Service

Amazon OpenSearch Service를 사용해서 CloudWatch로부터 수신된 클러스터 정보와 로그를 검색, 시각화, 분석하였다. 수집된 로그를 날짜별로 인덱싱하여 OpenSearch Dashboards에서 로그 시각화 하였고, 로그 표시 기간을 설정한 후 밀리초, 초, 분, 시간 등의 간격으로 로그를 확인 할 수 있다.

인덱스	문서 수	크기	쿼리 합계	매핑 유형
cwl-2022.10.25	864316	260.58 MiB	0	properties
cwl-2022.10.26	8867	5.90 MiB	0	properties
cwl-2022.10.27	1183958	427.26 MiB	0	properties
cwl-2022.10.28	8597	5.43 MiB	0	properties

[표 3-2] Amazon OpenSearch Service - 인덱싱된 로그 항목

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-38] OpenSearch Dashboards - 로그 시각화

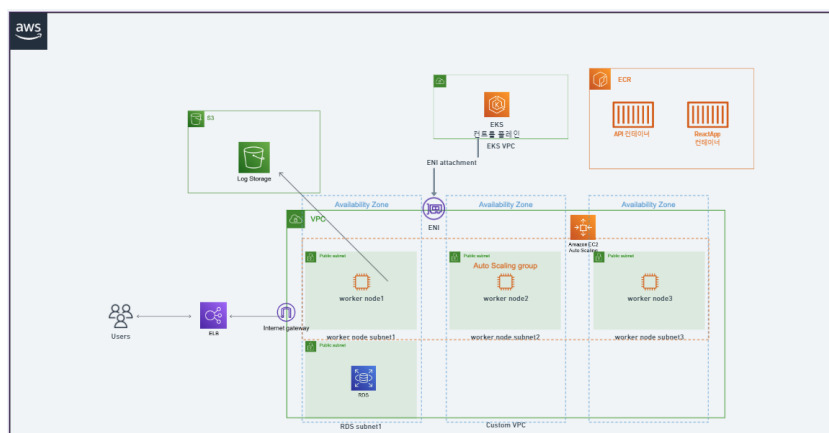
## 3.5 AWS

### 3.5.1 EKS & 클러스터 구성

#### 3.5.1.1 AWS EKS [\[참고자료 1\]](#)

- AWS에서 제공하는 관리형 Kubernetes 서비스
- 컨테이너 예약, 애플리케이션 가용성 관리에 용이
- 클러스터 데이터 저장 및 컨트롤 플레인의 가용성과 확장성 관리에 용이
- 클러스터 제어 영역에 보안 패치가 자동으로 적용 되어 안정성 보장

#### 3.5.1.2 클러스터 구성도



[이미지 3-39] 클러스터 구성도



	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## 3.6 AutoScaling

### 3.6.1 HPA

HPA는 Horizontal Pod AutoScaler의 약자이다. HPA를 이용하기 위해서는 쿠버네티스에서 공식적으로 지원하는 Add-On인 Metric Server를 설치해야 한다. Metric Server는 워커 노드의 kubelet과 K8S API Server 사이에서 컨테이너의 메트릭 정보를 수집하여 CPU와 Memory의 사용량을 확인한다. HPA는 애플리케이션 메트릭을 메트릭 서버를 통해 모니터링하여 Deployment, ReplicaSet, StatefulSet의 Pod 수를 자동으로 조절하는 기능을 한다.

NAME	READY	STATUS	RESTARTS	AGE	
'''					
pod/ <b>metrics-server</b> -847dcc659d-wjtm	1/1	Running	0	5d19h	
'''					
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kube-dns	ClusterIP	10.100.0.10	<none>	53/UDP,53/TCP	14d
service/metrics-server	ClusterIP	10.100.88.84	<none>	443/TCP	7d22h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/cluster-autoscaler	1/1	1	1	6d17h
deployment.apps/coredns	2/2	2	2	14d
deployment.apps/ <b>metrics-server</b>	1/1	1	1	7d22h

[텍스트 3-1] Metric Server 설치

우선 쿠버네티스에서 Add-On으로 제공해주는 Metric Server를 설치하였다.

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

```

'''
spec:
  containers:
    - image: 580103579984.dkr.ecr.ap-northeast-
2.amazonaws.com/goorm3team/goormbank:Frontend63
      imagePullPolicy: Always
      name: front-end-app
      ports:
        - containerPort: 80
          protocol: TCP
      resources:
        limits:
          cpu: "1"
          memory: 1Gi
        requests:
          cpu: 500m
          memory: 512Mi
'''

```

[텍스트 3-2] 프론트엔드 디플로이먼트 리소스 제한 걸기

HPA 서비스를 이용하기 위해서는 컨트롤러(Controller)를 이용하여 파드(Pod)를 배포할 때 파드의 자원량에 대한 제한을 두고 있어야 정상적으로 동작한다. 그렇기 때문에 프론트엔드 파드를 생성하는 'front-end-deployment' 설정에 CPU를 1000m Memory를 1Gi 제한을 두었다.

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS
AGE					
front-end-deployment	Deployment/front-end-deployment	0%/50%	1	10	1
7d21h					

[텍스트 3-3] HPA 오토스케일러 설정

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

다음으로 HPA를 설정해주었다. Pod가 CPU 50% 이상일 때 최소 개수는 1개, 최대 개수는 10개로 Scale Out 하도록 구성하였다.

NAME	READY	STATUS	RESTARTS	AGE
<a href="#">pod/front-end-deployment-69ff7584cf-fkrgt</a>	1/1	Running	0	24h
<a href="#">pod/front-end-deployment-69ff7584cf-yfsxz</a>	1/1	Running	0	24h
<a href="#">pod/front-end-deployment-69ff7584cf-ugdsa</a>	1/1	Running	0	24h

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/front-end-deployment	3/3	3	3	40h

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/front-end-deployment-69ff7584cf	3	3	3	24h

[텍스트 3-4] HPA 테스트 1

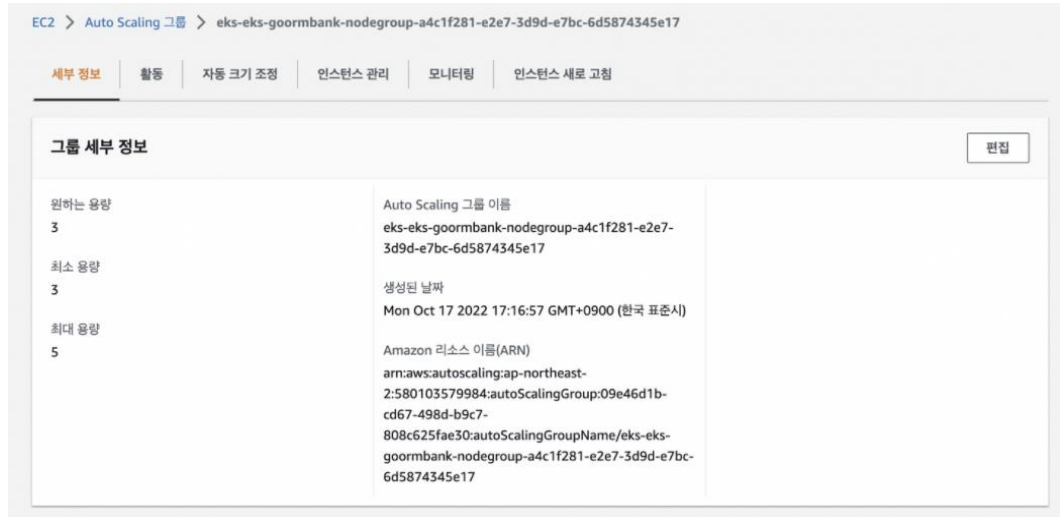
NAME	REFERENCE	TARGETS	MINPODS	MAXPODS
REPLICAS AGE				
front-end-deployment 1	Deployment/front-end-deployment	22%/7%	1	10
7d22h				

[텍스트 3-4] HPA 테스트 2

부하 테스트를 위해 HPA 설정 중 CPU 제한을 7%로 변경한 후 명령어를 통해 고의적으로 부하를 주었더니 [이미지 3-6-4]와 같이 'front-end'라는 이름을 가진 파드의 수가 늘어난 것을 확인할 수 있었다. 부하 주는 것을 멈추었더니 다시 파드의 수가 감소하는 것 또한 확인할 수 있었다.

### 3.6.2 CA

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-40] 오토스케일링 그룹 확인

이번에는 파드의 수를 늘리는 HPA가 아닌 노드의 수를 늘리는 CA를 구성해보았다. Cluster AutoScaling은 파드 생성 시 생성될 공간이 없어 Pending 되어있는 파드가 존재할 때 워커 노드의 수를 자동으로 늘려주는 기능을 한다. EKS Cluster를 통해 클러스터를 구축하였다면 클러스터를 생성할 때 자동으로 오토스케일링 그룹인 ASG는 [이미지 3-6-5]와 같이 생성되었을 것이다.

```
spec:
  containers:
  - command:
    - ./cluster-autoscaler
    - --v=4
    - --stderrthreshold=info
    - --cloud-provider=aws
    - --skip-nodes-with-local-storage=false
    - --expander=least-waste
    - --node-group-auto-discovery=asg:tag=k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/<YOUR
      CLUSTER NAME>

spec:
  containers:
  - command:
    - ./cluster-autoscaler
    - --v=4
    - --stderrthreshold=info
    - --cloud-provider=aws
    - --skip-nodes-with-local-storage=false
    - --expander=least-waste
    - --node-group-auto-discovery=asg:tag=k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/eks-goormbank-cluster
```

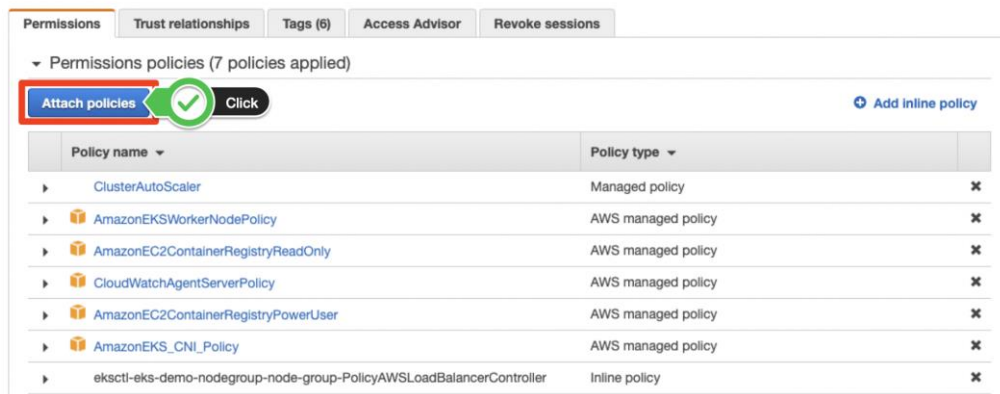
[이미지 3-41] Cluster AutoScaler YAML 파일 다운 후 수정

AWS에서 제공해주는 Cluster AutoScaler 파일을 다운 받은 후 위와 같이 '< YOUR CLUSTER NAME >' 부분을 우리 클러스터 이름인 'eks-goormbank-cluster'로 변경해주었다.

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-42] 오토스케일링을 위한 정책

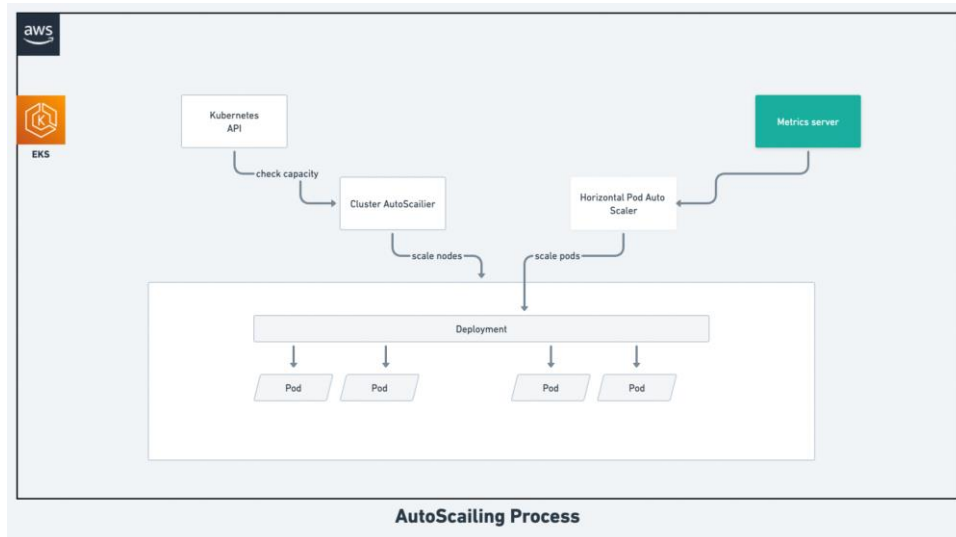


[이미지 3-43] ASG에 생성한 정책 적용

오토스케일러가 이용할 수 있도록 AWS 정책을 따로 생성해주었다. 생성한 정책을 오토스케일링 노드 그룹인 ASG에 적용해주었다.

### 3.6.3 오토스케일링 구성도

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	



[이미지 3-44] 오토스케일링 최종 구성도

클러스터에 적용한 오토스케일링은 HPA와 CA이며 구성도는 [이미지 3-6-9]와 같다. 추후에는 쿠버네티스에서 제공하는 Metric Server가 아닌 CloudWatch를 이용하여 메트릭 정보를 수집한 후 오토스케일링 할 수 있도록 변경할 예정이다.

## 4. 자체 평가 및 보완

### 4.1 보완

- 공개된 시험 문제에 대한 저작권 [\[참고자료 10\]](#)
  - 한국저작권보호원에서 2022년 8월에 나온 안내에 따르면 공공저작물은 문제가 없다.
  - 명확히 저작권자가 존재하며 무단전재 및 재배포가 금지된 경우는 허용하지 않고 (예를 들어 토익 등)
  - 공공저작물인지 애매해서 잘 알 수 없는 문제 및 개인의 창작 문제의 경우 타사이트의 선례를 따라 '저작권자의 요청시 삭제한다'로 표기
- 뉴스레터처럼 시험 정보에 대해 메일 발송 서비스
- 크롤링, 배치 시스템을 통한 정보 수집 자동화 필요

### 4.2 자체 평가

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

- EKS 도입 과정
  - EC2, VPC, IAM, EBS 및 기타 AWS 서비스와의 기본적인 통합이 가능하며 도입을 검토, 새로 사용하는 AWS 서비스라서 각자 사용법에 대한 적응 기간이 필요하여 책이나 문서등을 참고하며 진행, 여러 시행착오와 어려움을 겪었지만 이슈를 공유하며 같이 해결하였다.

- CI/CD
  - Jenkinsfile 작성 시 ECR로 Push하는 부분을 어떻게 정의해주는지 어려움이 있었고,
  - Jenkins에서 빌드 되어 ECR에 올라간 이미지 번호에 맞게, Git Manifest repository에서 자동으로 Manifest 파일을 업데이트 해주는 부분이 인증 관련 문제로 어려움이 있었다.

이와 같은 크고 작은 어려운 문제에 계속 부딪혔지만, 성공적으로 CI/CD 파이프라인을 구축할 수 있었고, 많이 배울 수 있었기 때문에 좋았다.

- FrontEnd
 

프론트엔드 애플리케이션을 빌드하고 이미지화 하는 과정에서 고려해야 할 사항이 많다는 것을 깨달았다. 특히 이미지를 빌드하는 로컬의 환경이 arm64 코어를 사용할 경우 amd64 코어를 사용하는 노드에서 해당 이미지를 사용하고자 할 때 문제가 발생한다는 사실을 알게 되었다.

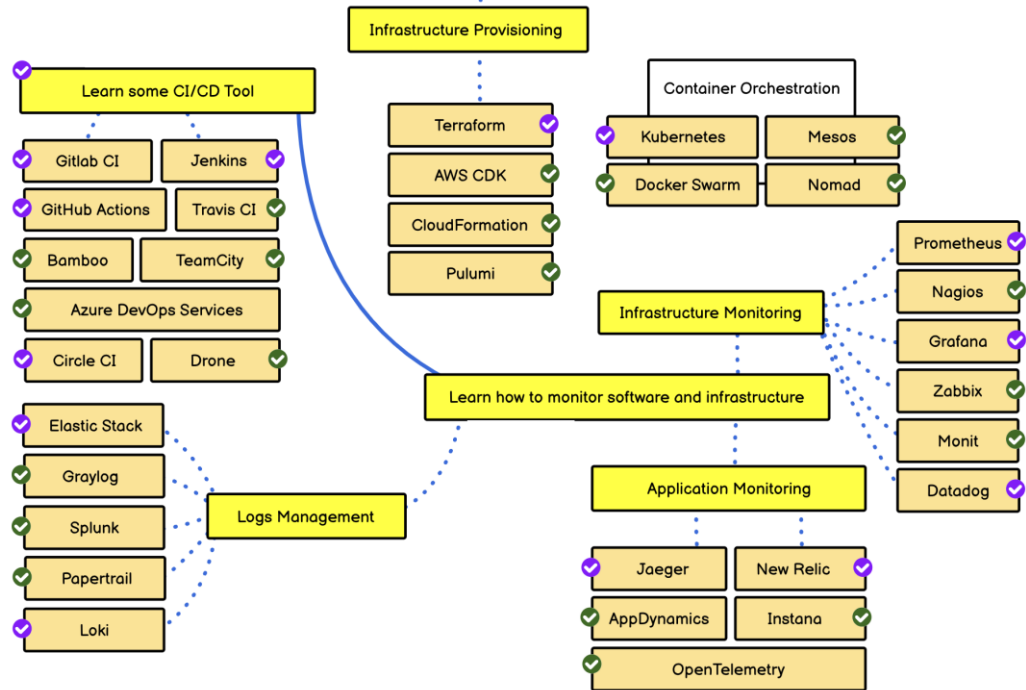
- Logging & Monitoring
  - Amazon과 Elasticsearch B.V.간의 라이선스 분쟁은 Oracle의 JDK 라이선스 분쟁을 떠올리게 하는 면이 있어 ELK(Elasticsearch, Logstash, Kibana) 대신 AWS에서 제공하는 OpenSearch Service 시리즈를 사용해야 했다. 오픈 소스를 더 많이 사용함에 따라 이러한 분쟁이 더욱 늘어날 것으로 보여 개발자조차도 법적인 문제를 먼저 고려해야하며 조금 더 신경써야할 것 같다.

- Cloudwatch에서 OpenSearch로 로그를 스트림할 때 VPC를 사용할 때 고려해야 할 점이 있다는 것을 알게 되었다. VPC에서 플로우 로그를 사용해야 CloudWatch에서 VPC에서의 로그를 수집할 수 있었고 이를 적용하여 OpenSearch에서 로그 스트림이 되지 않는 문제를 해결할 수 있었다.

- 로드맵 [\[참고자료 13\]](#)에 따라 EKS에 한두가지의 툴을 설치해보는 중에도 새로운 툴이 나오고 기존 툴은 업데이트 되는 등 끊임없이 따라가는 것이 다소 힘든 일이었다. kubernetes.dev,

	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

karpenter, kube-ops-view 등 새로운 툴을 마음껏 학습할 수 있었던 좋은 기회였다.





	Open			Over
	Category	첨부파일 버전	문서 최종 수정일	
	Manual + Utility	1.0	2022.11.01	

## 5. 참고자료

- [1] AWS EKS - [관리형 Kubernetes 서비스 | Amazon EKS](#)
- [2] AWS OpenSearch Service - [OpenSearch란 무엇인가요? 오픈 소스 검색 엔진 - AWS](#)
- [3] AWS RDS - <https://aws.amazon.com/ko/rds/>
- [4] Jenkins - <https://www.jenkins.io/doc/tutorials/>
- [5] ArgoCD - <https://argo-cd.readthedocs.io/en/stable/>
- [6] Kubernetes(k8s) - <https://kubernetes.io/ko/>
- [7] CloudWatch Agent - [클러스터 지표를 수집하도록 CloudWatch 에이전트 설정](#)
- [8] Streaming CloudWatch Logs data to Amazon OpenSearch Service  
- [Streaming CloudWatch Logs data to Amazon OpenSearch Service](#)
- [9] Enable VPC Flow Logs  
- [Stream Cloudwatch Logs to Amazon OpenSearch Service](#)
- [10] 한국저작권보호원 저작권 참고 PDF  
- <https://www.kcopa.or.kr/download.do?uuid=d754d525-f3e3-48d4-99fb-9f45603500f6.pdf>
- [11] AWS AutoScaling - [Cluster Autoscaler 적용하기](#)
- [12] 베스핀 글로벌 HPA - [\[AWS\] EKS Pod AutoScaler - HPA : BESPIN GLOBAL Support Portal](#)
- [13] 로드맵 - [Learn to become a DevOps Engineer or SRE](#)