

포트폴리오

안상운

기술블로그

<https://velog.io/@luckyprice1103/posts>

클릭



웹 포트폴리오

portfolio.cloudwoon.com

클릭



기술 스택 & 자격증

기술 스택



AWS



GCP



Azure

클라우드



Terraform



Ansible

IaC



MySQL

ORACLE

Oracle



AWS RDS

데이터베이스



Linux



VMware



VirtualBox

가상화 & OS

기술 스택



Prometheus



Grafana



AWS
CloudWatch

시스템 운영
& 모니터링



Docker



Kubernetes



Jenkins



ArgoCD



GitLab



GitHub Actions

CI/CD & 컨테이너



Python



Java

프로그래밍 언어

Feature & Stack

자격증 & 어학 능력



**AWS Solutions
Architect - Associate**



정보처리기사



SQLD



OIPc IH

주요 프로젝트

On The Top

TILing

MUGU



On The Top 카카오테크 부트캠프(4개월, 6명)

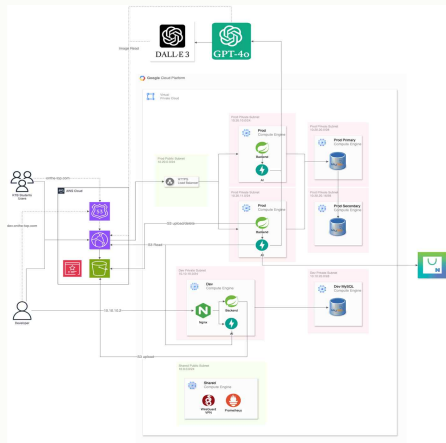
On The Top은 사용자의 현재 책상 사진을 기반으로 추가하면 좋을 데스크테리어 아이템을 추천해 주고, 해당 사진을 기반으로 추천된 아이템이 적용된 모습을 AI를 통해 생성하여 제공하는 서비스입니다.

커뮤니티를 즐길 수 있고, 포인트로 물건들을 구매할 수 있습니다.

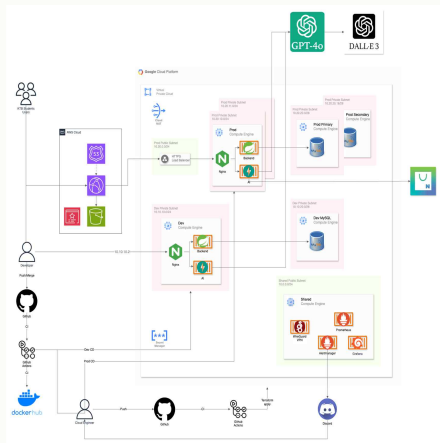


Design

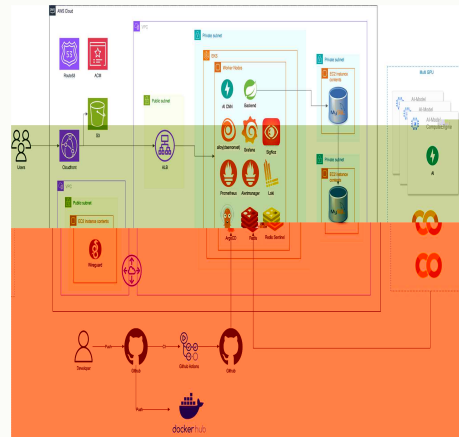
프로젝트 아키텍처 개선 과정



MVP 버전 개발 아키텍처



도커 컨테이너화 배포 아키텍처

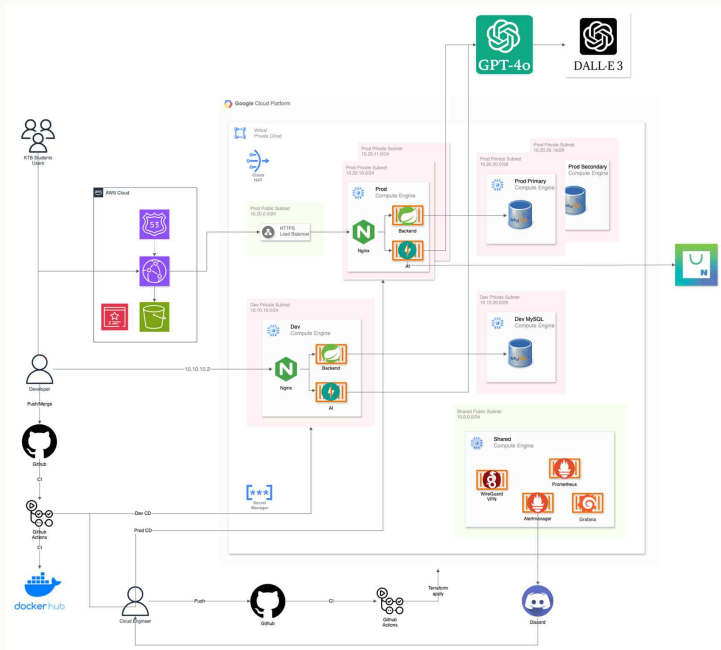


Kubernetes 기반 배포 자동화 아키텍처

Design

클라우드 아키텍처: 도커 도입

CI/CD (애플리케이션 배포): 개발자가 Github에 코드를 푸시하면, Github Actions가 이를 감지하여 Docker 이미지를 빌드하고 Docker Hub에 푸시합니다. 이후 개발(Dev) 환경에 자동으로 배포됩니다.



도커 단계 Dev 환경 CI/CD 파이프라인

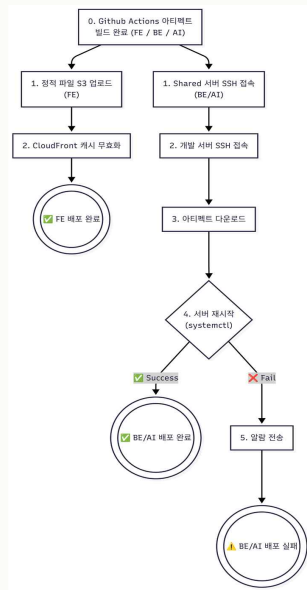
개발(Dev) 환경 CI

BE : dev 브랜치 push시 github actions를 통해 자동으로 도커를 빌드합니다.

개발(Dev) 환경 CD

BE/AI 배포: Shared 서버(Bastion Host)를 통해 개발 서버에 SSH로 접속한 뒤, dev 브랜치 push시 빌드되는 최신 도커를 다운받아 가동합니다.

FE 배포: 빌드가 완료되면, 정적 파일을 AWS S3에 업로드하고 CloudFront 캐시를 무효화하여 배포를 완료합니다.



도커 단계 Prod 환경 CI/CD 파이프라인

운영(Prod) 환경 CI

BE : push시 version tag를 입력하면
github actions를 통해 자동으로 prod
버전 도커를 릴리즈합니다.

운영(Prod) 환경 CD

BE 배포: 특정 버전과 필요 값들을
입력하면 해당 버전의 도커를 다운받아
가동합니다.
이때 스크립트를 통해 blue-green배포를
구현하였습니다.

FE 배포: 버전을 입력하면 정적 파일을
AWS S3에 업로드하고 CloudFront
캐시를 무효화하여 배포를 완료합니다.

Use workflow from

Branch: main ▼

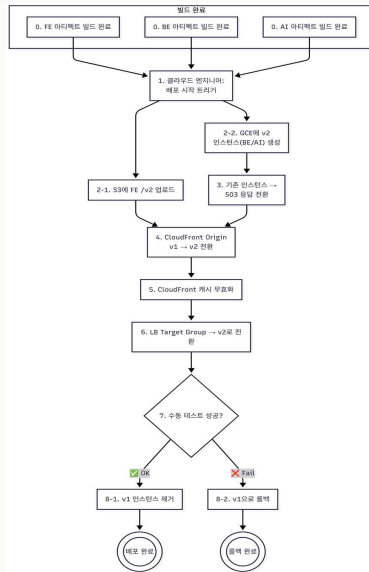
배포할 BE docker 버전 (예: 1.2.3) *

배포할 FE 버전 (예: 1.2.3) *

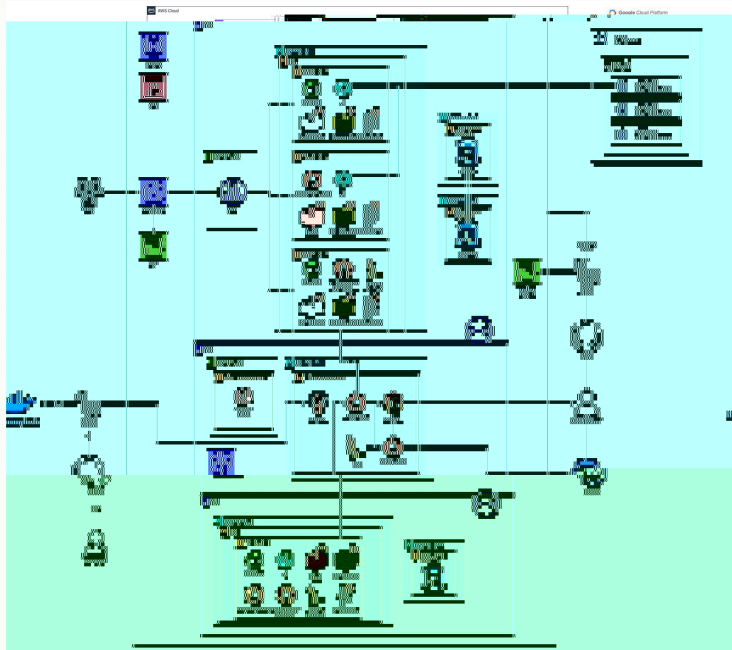
FE 슬롯 선택 (예: blue, green) *

green

Run workflow



클라우드 아키텍처: EKS 도입(AWS로 서버 마이그레이션)



AWS와 GCP를 함께 사용하는 멀티 클라우드 아키텍처에서 AWS로 모든 서버를 마이그레이션 했습니다.

Argocd yaml파일을 통해 버전과 replicas를 컨트롤 합니다.

네임스페이스를 통해 DEV환경과 PROD환경을 논리적으로 분리했습니다.

AWS Secrets Manager, ESO, ClusterSecretStore, Kubernetes Secret을 통해 secret값을 안전하게 pod에 주입했습니다.

Design

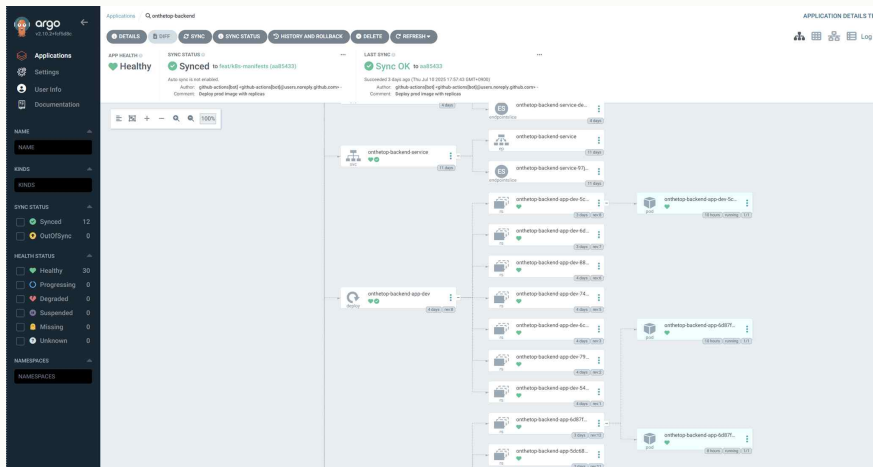
EKS CI/CD

CI

push시 version tag를 입력하면 github actions를 통해 자동으로 prod 버전 도커를 릴리즈합니다.

CD

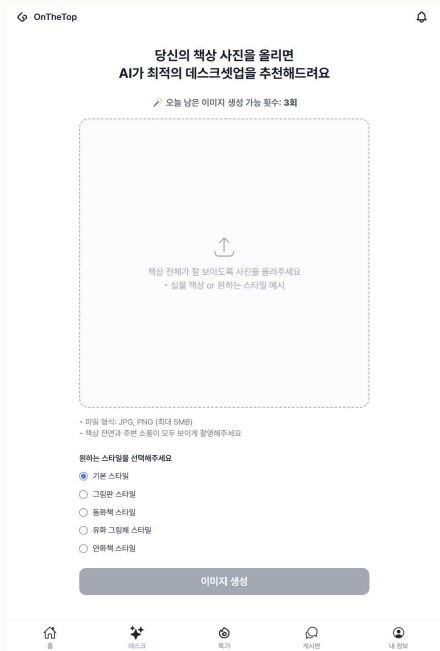
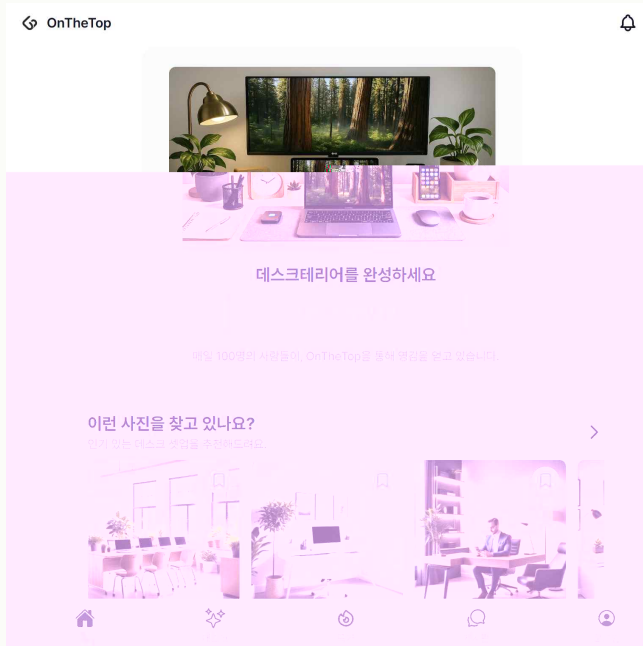
argocd와 깃허브에 있는 yaml파일들을 통해 네임스페이스로 분리한 dev와 prod환경의 eks를 쉽게 컨트롤 합니다.



<https://github.com/100-hours-a-week/16-Hot6-cloud/tree/feat/k8s-manifests/k8s-manifests>

구현 화면

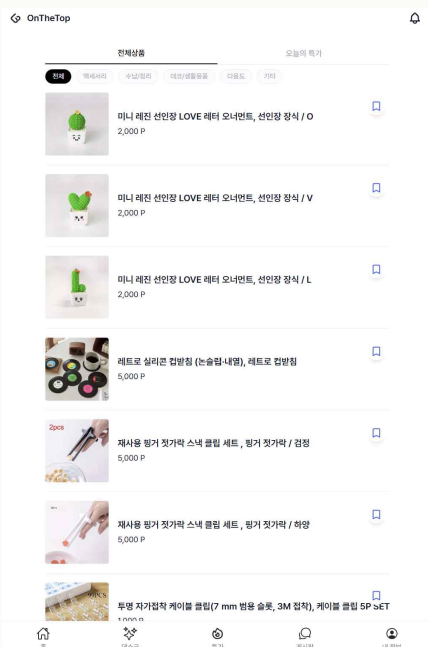
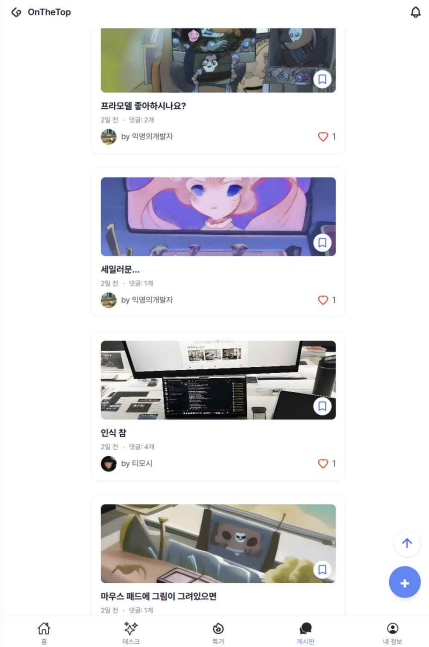
www.onthe-top.com



Implementation

구현 화면

www.onthe-top.com

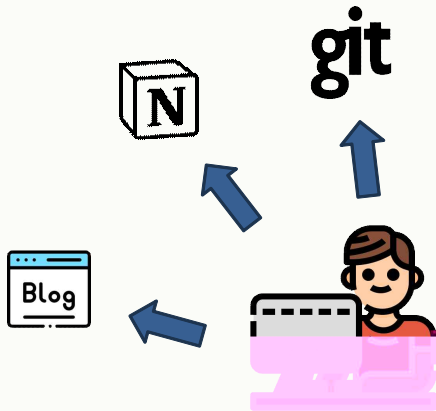




TILing 카카오테크 부트캠프(2개월, 6명)

하나의 플랫폼에서 TIL을 작성 및 관리할 수 있는 서비스

- 작성한 TIL을 Markdown 형식으로 변환하여
연동된 플랫폼에 자동으로 업로드할 수 있는 기능을 제공
- 효율적인 기록을 위해 TIL 템플릿을 제공
- LLM의 도움으로 Markdown이 익숙하지 않은 사용자도 쉽게 작성 가능
- LLM을 통해 각 플랫폼 별 Markdown 형식으로 쉽게 변환, 업로드
- 클라우드 기반으로 서비스의 배포 및 확장성을 지원



Design

클라우드 아키텍처

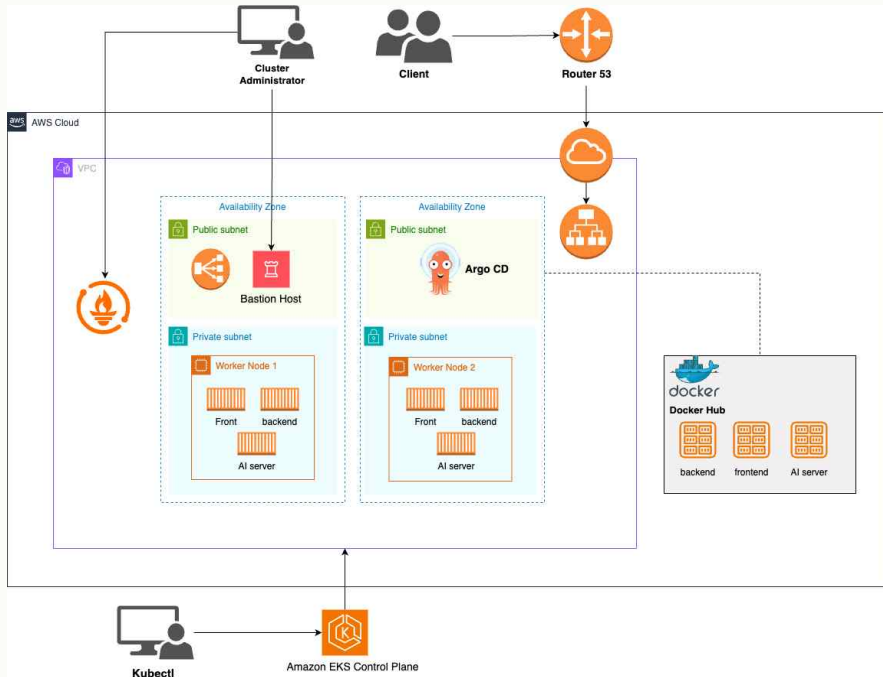
구조 설명

GitHub를 통해 병행 개발 진행

Jenkins가 Git Hub main branch 모니터링.

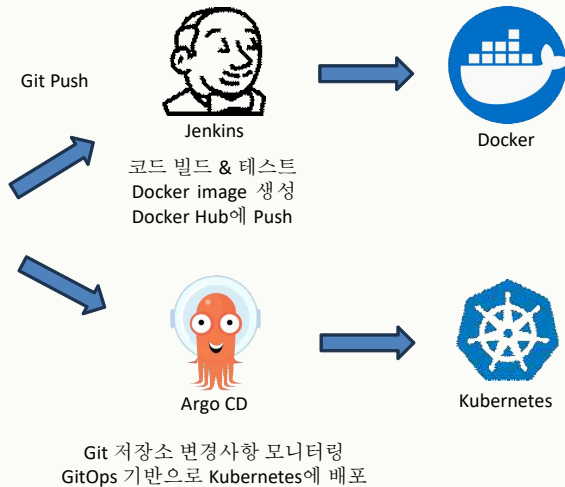
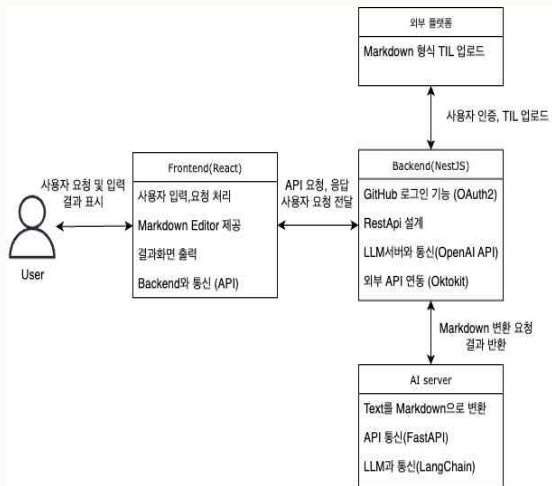
Main branch 변경 시 Jenkins를 통해
자동으로 Docker 이미지 생성,
Docker Hub에 Push

Argo CD를 활용해 Kubernetes의
배포 자동화

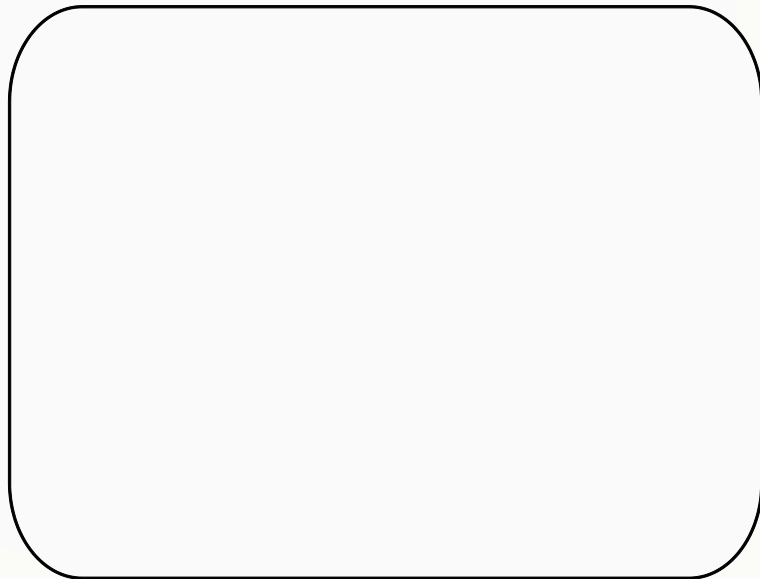
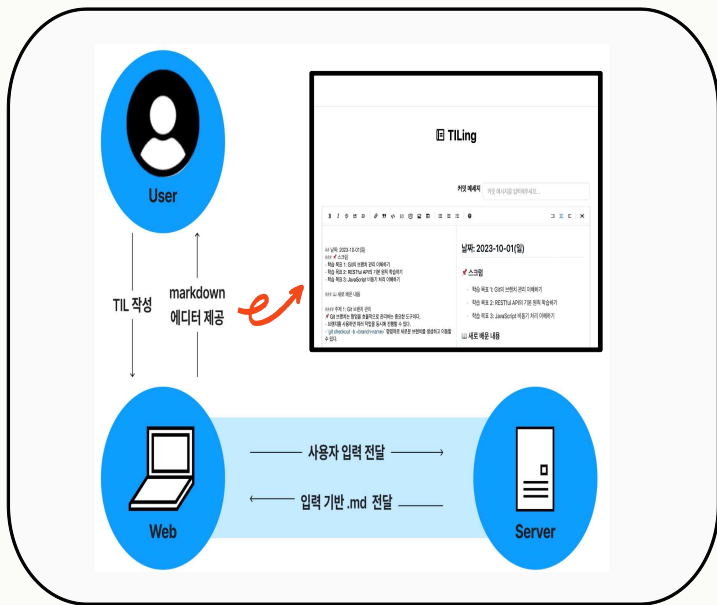


Design

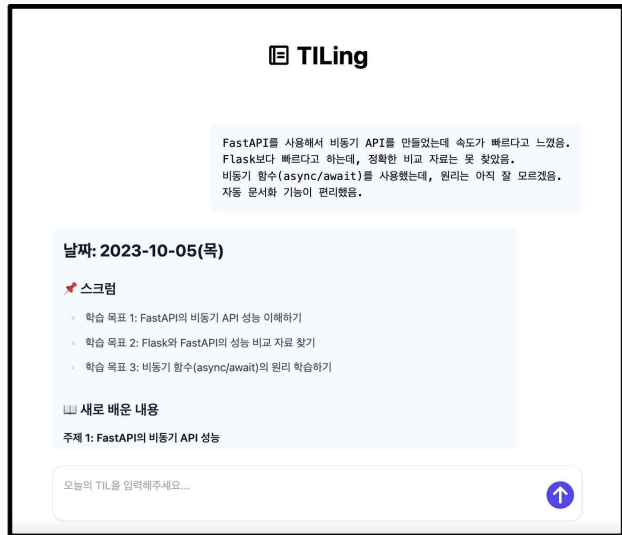
아키텍처 - Server & Client & Deploy



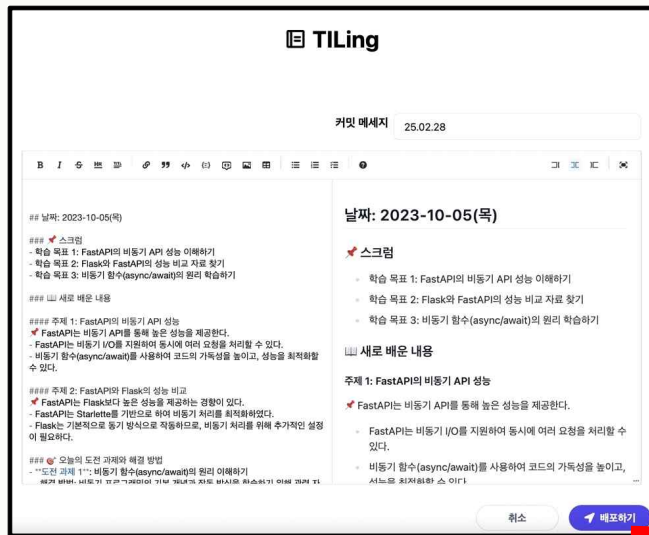
Frontend, AI Server



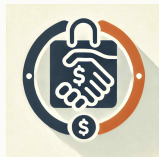
구현 화면



1. LLM을 통해 TIL을 작성



2. 최종 수정 후 배포



MUGU대학 캡스톤 디자인(4개월, 4명)

집에 필요 없는 물건을 쉽게 팔거나, 단기적인 알바를 구하기 위한 사이트

- 회원 가입 및 로그인 기능 제공.
- 회원 각각의 장바구니 기능 및, 검색 기능 제공.
- 업로드 후 업로드한 사람과의 채팅 구현.
- 원하는 만큼의 캐시를 입력하고, 충전 할수있는 기능.
- 클라우드기반으로 서비스의 자동배포 및 확장성을 지원



클라우드 아키텍처

Fargate, ECS로 프론트엔드.백엔드
컨테이너 배포, 블루/그린 배포, Aurora
사용.

ECR에 컨테이너 이미지 저장,
CloudWatch Logs, Fluent Bit로
로깅 및 모니터링 수행.

This diagram illustrates a multi-region, multi-availability zone AWS architecture. It features two identical regions (us-east-1 and us-west-2) for high availability. Each region contains two availability zones (us-east-1a/us-east-1b and us-west-2a/us-west-2b). The architecture includes an Internet Gateway for external access, an Amazon Route 53 for DNS, and an Amazon CloudFront for content delivery. The application is deployed on Amazon ECS using Fargate instances, with a multi-tier database structure (Amazon Aurora) and a Redis cache. The architecture also includes an Amazon S3 bucket for static content, an Amazon ECR repository for container images, and an Amazon IAM role for the ECS task role. The architecture is designed to be highly available and scalable across multiple regions and availability zones.

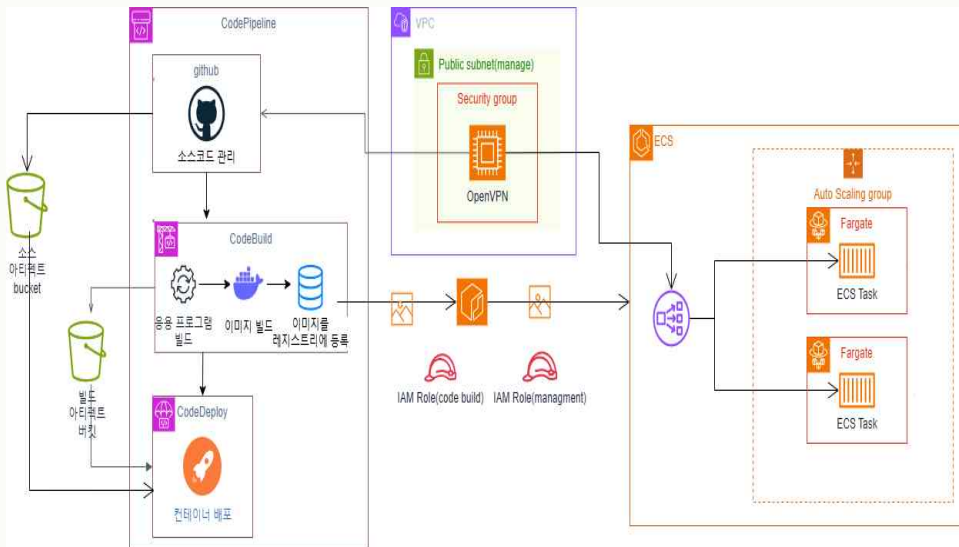
Design

CI/CD 파이프라인

에서 코드를 푸시하면
이 이를 감지하고
가 이미지를
빌드하여 컨테이너 레지스트리에
저장

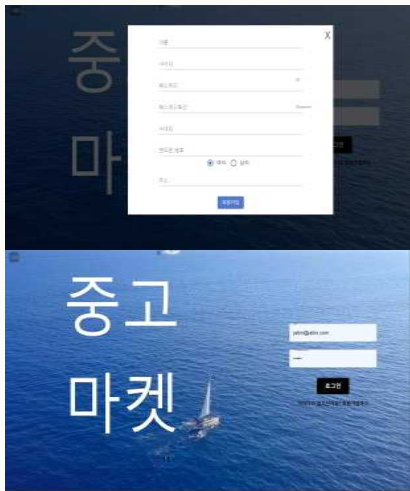
이후 가 이를 의
환경에 배포하며
을 통해 컨테이너를
동적으로 확장 축소

역할을 활용하여 권한을 관리
배포 및 모니터링



Implementation

주요 구현 화면



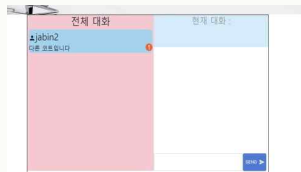
1. 로그인



2. 메인 페이지



3. 게시글 작성.



4. 채팅



5. 캐시 충전

감사합니다.
열심히 하겠습니다....!

Tech Blog

<https://velog.io/@luckyprice1103/posts>

클릭



Web Portfolio

portfolio.cloudwoon.com

클릭



GITHUB

github.com/luckyPrice

클릭

