

# 김성엽 Portfolio

**현재 주력 LANGUAGE :**

Python, Golang

[개발 경력]

**2022 년 8 월 1 일 ~ 현재**

**메가존 클라우드 근무**

**주 사용언어 : python, golang**

**주요 업무 :**

## **AIOPS platform on KT CLOUD**

주 사용언어 : python, golang

주요 프레임워크 : Django

주요 기술스택 : EKS, terraform, Prometheus, ELK, argocd, Jenkins, istio, Kubeflow, helm

주요 성과 :

KT CLOUD 의 AI SERV 와 연계하는 모델 서빙 기능 개발

KT CLOUD 인프라에 빠른 플랫폼 배포에 기여

아래에서 언급한 mlops platform 의 기능에 덧붙여 KT CLOUD 와 협업하여 mlops platform 을 구축 중에 있습니다. 저는 해당 프로젝트에서 AI SERV 관련 기능에 대한 개발을 맡고 있으며, 현재 개발이 끝난 상태입니다. 저희 플랫폼의 ASSET 관리 모듈인 ASSET-HUB 에 모델서버 파일을 올리고 해당 파일을 KT 의 모델 서버가 가져가도록 하는 일련의 business 로직을 개발하여 해당 서버의 관리를 저희 플랫폼에서 할 수 있도록 하였습니다.

해당 기능을 통해 triton, tensorflow, pytorch 등의 서버를 띄울 수 있도록 하였습니다.

또한 기존에 작성해두었던 terraform script 를 활용하여 kt 인프라에 저희 플랫폼을 빠른 속도로 배포할 수 있도록 하여 개발의 소요시간을 단축하였습니다.

## **mlops platform 개발(devops, software 개발)**

주 사용언어 : python, golang

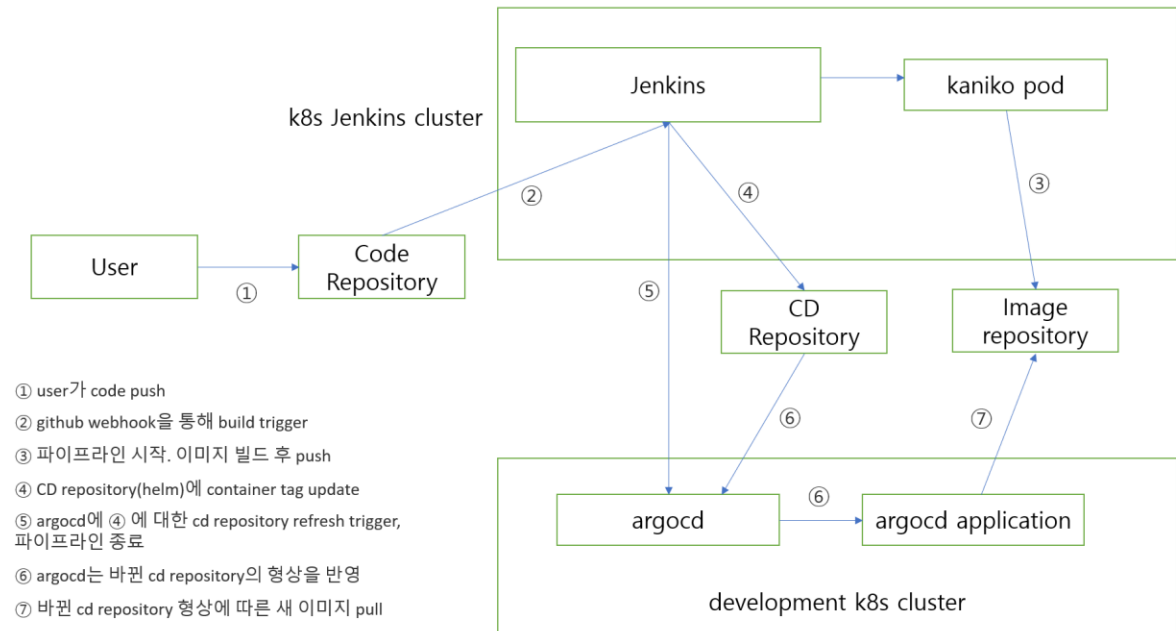
주요 프레임워크 : Django

주요 기술스택 : EKS, terraform, Prometheus, ELK, argocd, Jenkins, istio, Kubeflow, helm

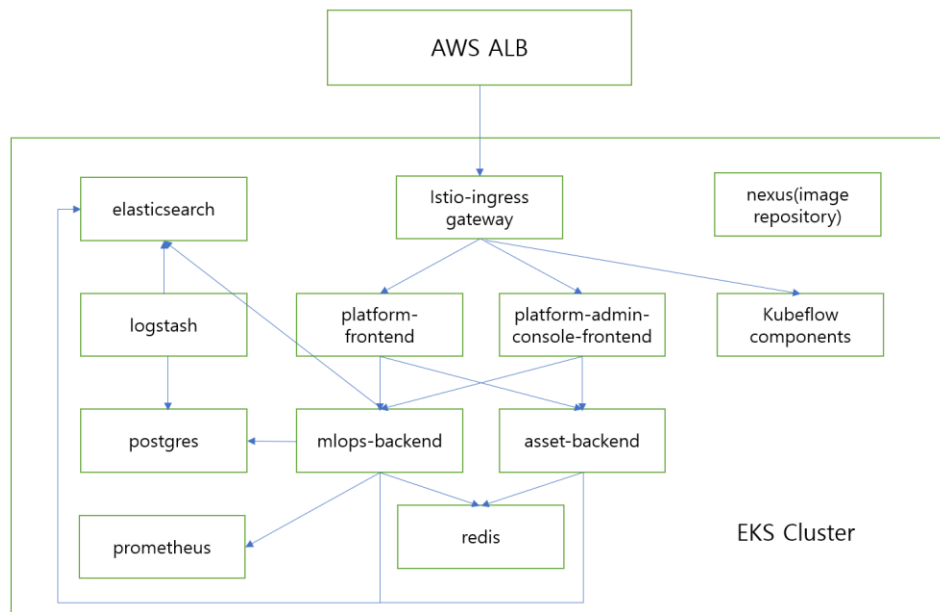
주요 성과 : 현대 사이트솔루션에 mlops 솔루션제공, 우수 사원 선정

기본적으로 개발에 대한 CI/CD 파이프라인을 직접 구축하였습니다. jenkins, argocd 를 연계하여 배포를 할 수 있도록 하였습니다. 빌드 과정에서 kaniko build agent 를 사용하도록 하여 따로 도커 데몬 서버를 띄우지 않더라도 빌드가 수행될 수 있도록 구성하였습니다. 또한 컨테이너 이미지 빌드시에 commit hash 가 container image 의 태그로 붙게 하여 어느 커밋에 대한 컨테이너 이미지인지를 빠르게 파악할 수 있도록 하여 롤백 등에 적극 활용될 수 있도록 하였습니다.

CD repository 를 따로 구성하여 argocd application 을 만든다음, 젠킨스에서 빌드 될 때 마다 CD repository 의 image tag value 를 바꿔서 형상을 바꾸는 전략을 취했습니다.



사내에 Kubeflow 기반 mlops 솔루션 개발에 기여 하였으며 Kubernetes 위에서 개발을 진행하였고 서비스 아키텍처 설계를 도맡아 하였습니다. Istio 를 활용해 서비스 메쉬를 구성하였고, gateway 에서 dex IDP 와 연계하여 인증체계를 구성하였습니다.



**mlops platform architecture**

현대 사이트솔루션의 경우 saml 을 통한 인증에 대한 요건이 있었는데 istio gateway 와 현대 사이트솔루션의 nets IDP 와 연동하는 인증 연동 작업을 수행하였습니다.

또한 유저 별로 역할을 부여하여 각 유저의 역할별 자원 사용을 제한하는 정책을 세우고 구현하였습니다. 또한 역할에 할당된 permission 에 따라 특정 메뉴나 admin\_console 의 접근 제한을 제어하는 정책을 세웠습니다.

또한 플랫폼 내부적으로 container 이미지를 새로 생성하는 기능이 있었는데, 기존에 도커 데몬 ec2 서버를 띄워서 진행하던 방식을 kaniko 를 사용하여 도커데몬 서버가 필요없도록 개선하여 한 클러스터 내에서 모든 작업이 이뤄질 수 있도록 구성형태를 개선하였습니다.

Open api 에 대한 요건이 있어 open api 에 대한 외부의 요청일 경우 인증을 우회하는 로직을 istio 인프라 단에 넣어서 백엔드에서 api key 에 대한 인증 확인을 구성하도록 하였습니다.

또한 플랫폼 내에 각종 스토리지 모니터링을 위해 prometheus 를 도입하였고, 스토리지 모니터링을 위한 exporter 를 go lang 으로 작성하였습니다.

스토리지 모니터링 뿐만 아니라 mlops jupyter 노트북 자원의 회수를 구현하기 위해 각 리소스에 대해서 prometheus 의 pod cpu 를 조회하는 쿼리를 날려 cpu 사용률이 낮다면 노트북을 회수(정지)하는 정책을 세웠습니다.

또, 노트북의 시간대별 사용시간을 체크하기 위해 각 노트북이 살아있는지를 주기적으로 조사하여, 사용시간을 누적하는 로직을 exporter 로 구현하여 시계열 형태로 대시보드에서 파악할 수 있는 api 를 만들었습니다.

노트북 사용시간 api 이외에도 dashboard backend api 구현을 하였는데, 이때 Prometheus query(cpu, memory)를 활용하여 일별, 주간, 월별 데이터에 대한 dashboard api 를 구현하였습니다,

또한 플랫폼 내의 통합 검색 기능, pod 로그 조회 기능을 구현하기 위해 ELK 를 배포하였으며, 라이선스 문제를 고려하여 helm chart 를 구성하였습니다.

Terraform 을 통해서 EKS 클러스터를 배포하고 플랫폼 배포까지 자동화하여 스크립트만 실행하면 바로 플랫폼을 구축할 수 있는 형상까지 만들었습니다.

개발 과정에서 총 4 개의 클러스터를 배포하였으며, 각 클러스터의 역할은

dev, qa, prod-dev, prod 이었습니다.

dev 클러스터에서 CI CD 를 통해 개발 결과를 즉각적으로 확인하며 검증하였고 qa 클러스터에 주기적으로 머지하여 배포 형상에 대한 qa 테스트를 진행하였습니다. 그 후 prod-dev 에 해당 이미지를 적용하여 확인하고, 실제 운영 클러스터인 prod 에 배포하였습니다.

플랫폼 구축에 있어 전반적으로 큰 기여를 하였으며, 해당 업무로 인해 지난 분기에 우수 사원으로 선정되었습니다.

## **2021 년 12 월 1 일 ~ 2022 년 7 월 31 일**

### **데브스택 근무**

주 사용언어 : JAVA, javascript

주요 프레임워크 : Spring Boot, vue.js

주요 업무 :

### **삼성 SDS 의 IDP 유지보수 및 신규 기능 개발**

삼성 SDS 의 IDP 의 유지보수 및 신규 개발 업무를 맡아 삼성 SDS 및 에스코어 내부 인원들과 의사소통을 진행하며 IDP 기능의 개발 및 인프라 구성에도 참여하였습니다.

삼성 SDS 의 IDP 는 CAS 를 베이스로 구성되어 있으며, 여기에 부가 기능을 더하여 kubernetes 상에 두개의 서비스로 띄워져 있습니다. Spring boot 를 통한 backend 개발 뿐만이 아니라 필요시 vue.js 를 통한 frontend 개발을 진행하여 전체적인 개발에 대한 시야를 넓힐 수 있는 계기가 되었습니다.

또한 해당 업무를 통해 jenkins, argocd 조합의 CI / CD 체계를 경험하여 추후에 메가존클라우드의 업무에서 적용할 수 있는 계기가 되었습니다.

## **2021 년 4 월 26 일 ~ 2021 년 9 월 -**

### **가온미디어 근무**

주 사용언어 : golang

주요 개발 스택 : microk8s, gke, istio, mariadb, mysql, redis

주요 업무 :

## 마이크로 서비스 기반 MQTT Broker 라이브러리 개발

KRMS(KAON REMOTE MANAGEMENT SOLUTION) 솔루션이란 SET-TOP BOX 를 remote 로 관리해주는 솔루션입니다.

해당 시스템에 대해서 특정 프로토콜의 경우 MQTT 를 통해서 device(Set-Top Box)와 통신을 해야하는 요건이 있었습니다. 그래서 해당 모듈을 KRMS 시스템에 추가하였습니다.

KRMS 시스템에서 트리거를 걸면 외부 device 가 MQTT BROKER 에 subscriber 로 붙는 구조였으며, subscriber 로 붙은 후 특정 토픽에 대해 요청을 publish 하면 device 는 요청한 JOB 을 수행합니다. 라이브러리 작성 뿐만이 아니라 해당 서비스를 gcp 인프라에 마이크로 서비스 형태로 추가하는 과정을 수행하였습니다. 시스템 외부에 존재하는 device 가 MQTT BROKER 에 접근해야 하기 때문에 istio-ingress-gateway 에 MQTT broker 의 포트를 추가하고, virtual service 를 작성하여 해당 포트가 실제 mqtt broker 서비스로 라우팅 되도록 구성하였습니다.

## 전반적인 krms 서비스에 대한 유지 보수 및 신규기능 개발

mqtt broker 뿐만 아니라 기존의 krms 시스템에 대한 신규 기능 개발 및 기존 시스템의 유지보수를 수행하였습니다. 시스템 아키텍처가 복잡하여 이해하는 데에 수 개월이 걸렸으나 결국 이해하는 데에 성공하여 전체적인 시스템을 관장할 수 있는 정도의 인재로 성장하였습니다. 신규 기능개발의 경우 디바이스에게 지금까지 KRMS 에서 서비스 되고 있지 않은 동작에 대한 요청을 수행하는 부분을 구현하고 그 결과를 받아 웹소켓에 전송하는 전반의 과정을 구현하였습니다.

기존 시스템의 유지보수 역시 전체적인 시스템의 구조를 파악하고 있기 때문에 시스템에 문제가 발생했을 시 어느 서비스에 문제가 생겼는지 바로 인지할 수 있었으며, 빠른 속도로 해당 문제를 수정할 수 있는 역량을 갖추게 되었습니다.

**2019 년 10 월 1 일 ~ 2021 년 2 월 -**

**뱅크비(bankbe)근무**

주 사용언어 : golang

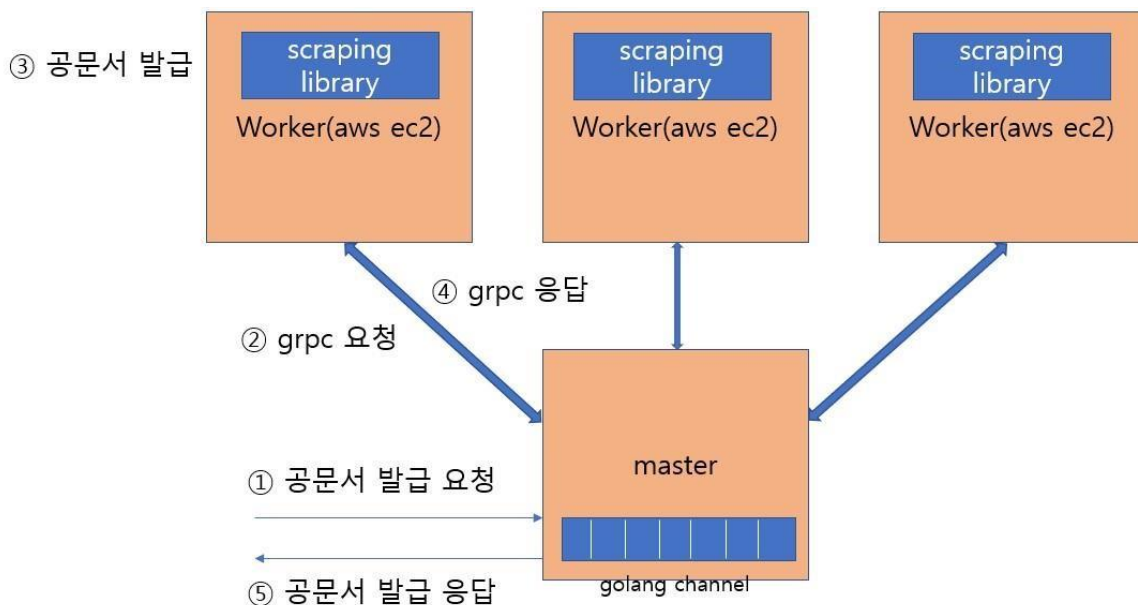
주요 업무 :

c 로 개발된 scraping 라이브러리를

구동하는 golang backend server 유지 및

보수(2019.10~2021.2)

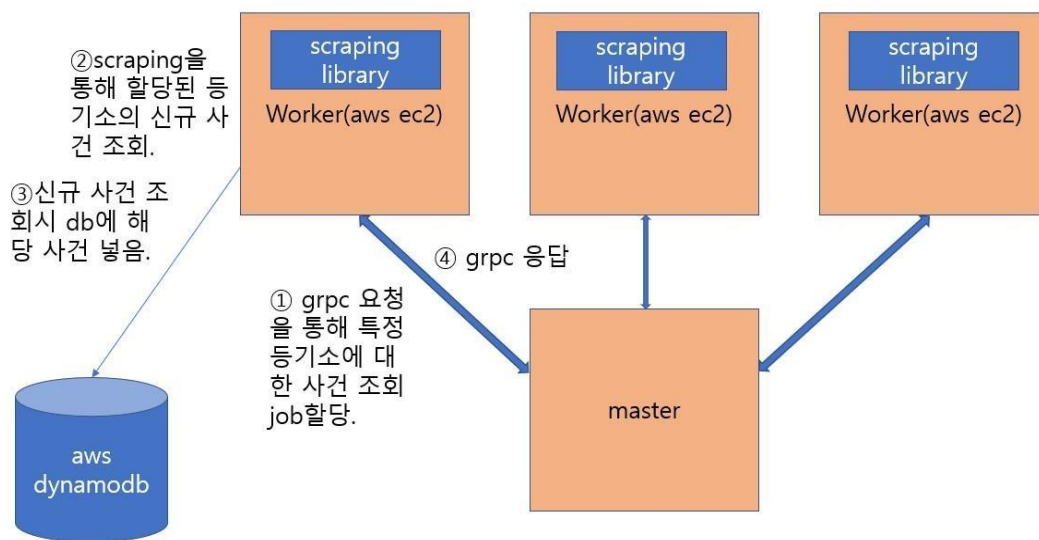
하나의 마스터와 여러개의 worker 서버(aws ec2)들이 grpc 로 통신하는 다중 인스턴스의 구조의 golang backend server 유지 보수 업무 수행하였습니다. 해당 서버에서는 여러가지 공문서에 대한 발급 요청을 외부로부터 받으면 해당 요청을 golang channel 에 쌓아두었다가 하나씩 worker instance 의 서버에 grpc 요청을 보내고 각 worker 서버는 scraping 라이브러리를 통해 공문서를 발급 후 해당 문서에 대한 응답을 받는 구조입니다. 또한 앞단에 **api gateway** 를 두어 api key 를 통한 모든 backend server 에 대한 공통 인증 방식을 적용하였습니다. 해당 서버에 대한 유지 및 보수 및 새로운 문서 발급 기능 추가를 지속적으로 수행하였습니다.



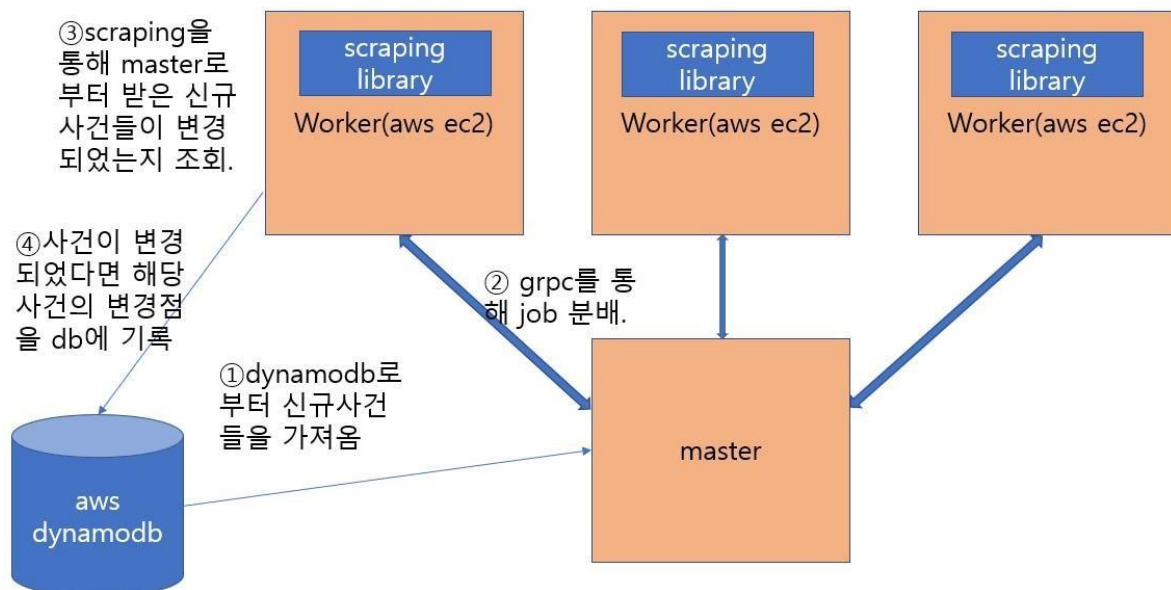
scraping 라이브러리를 통해 대법원 등기소의 등기사건을 조회하는 golang 기반 분산 서버 개발(2019.10~2020.04)

크게 두가지의 서버 모듈(신규 사건 조회, 사건 변경 조회)로 나뉘었으며, 각 서버에서 분산적으로 대법원등기소의 부동산 등기사건 변경 여부를 조회하였습니다. 신규 사건 조회

모듈에서는 신규 등기 사건에 대해 각 등기소별로 감지하여 **aws dynamodb**에 저장하였으며, 사건 변경 감지 모듈에서는 신규감지 모듈을 통해 db에 등록된 사건이 실제로 등기완료가 되었는지 지속적으로 스크래핑을 통해 감시하였습니다. 각 모듈은 하나의 master 서버에 여러개의 worker 서버로 구성되었습니다. master 와 worker 는 grpc 를 통해 통신하였고 master 가 worker 에 job 을 분배하는 구조로 이루어졌습니다.



[신규 사건 조회 모듈]



[사건 변경 조회 모듈]



## **kubernetes 를 활용한 스크래핑 모듈**

### **구동하는 golang backend 서버**

**개발(2020.07~2021.02)**

ec2 를 통해 backend 서버를 개발하면서 느꼈던 단점이 바이너리 교체등의 작업시 서버 구동이 중지된다는 점이었는데, 쿠버네티스에서 rolling update 를 통해 무중단 서비스를 제공해준다는 점을 인지하고 적용의 이점을 깨달아 직접 스터디를 통해 사용법을 익힌 후 회사내에서 독자적으로 쿠버네티스를 통해 backend server 개발하였습니다. 또한 backend server 에서 진행중인 http request 의 수를 알려주는 **metric** 을 export 하고 해당 metric 을 **prometheus** 를 사용해 수집 하도록 하여 해당 값이 크면 pod 의 수를 늘리는 **autoscaling** 정책을 도입하였습니다. 또한, **kubernetes ingress** 의 cookie 기능을 활용하여 단일 pod 에서 특정 클라이언트로부터의 요청을 계속 받아야 하는 요구사항을 해결한 경험이 있습니다. kubernetes 를 활용하여 두개의 프로젝트를 진행하였습니다.

프로젝트 1. 간단한 스크래핑 모듈을 구동하는 rest api 서버 개발    프로젝트 2. 베트남의 은행에 대한 계좌 이체의 전반적인 과정을 수행하는 스크래핑 모듈을 구동하는 rest api 서버 개발.

**2017 년 02 월 01 일 ~ 2019 년 09 월**

**20 일 - 티맥스소프트 근무**

주 사용언어 : C++, JAVA

### **C++ ORM 개발.**

주된 업무는 C++ ORACLE 전용 ORM 개발이었습니다. ORACLE 클라이언트 라이브러리인 OCCI 를 사용하여 ORM 개발하였습니다. 추가적으로 자사 DBMS 인 TIBERO 에 대한 ORM 을 추가적으로 개발하였습니다. 개발 자체는 FRAMEWORK 를 알아가는 측면에서 유익했지만 실제 고객들에게 노출되는 서비스 개발을 하면서 다양한 개발 경험을 쌓고 싶어 티맥스소프트를 퇴사하였습니다.