

경력이력서 및 포트폴리오

서우영

연락처 ➡ cerealtigerpower@gmail.com

개발 Blog ➡ <https://wooyoung85.tistory.com>

Github ➡ <https://github.com/wooyoung85>

자기소개

안녕하세요 LG CNS에서 Application Architect로 재직 중인 서우영입니다 😊

주요 보유 기술

- 클라우드 아키텍처 설계 및 구축 (Azure, GCP, AWS)
- Kubernetes 기반 플랫폼 개발 및 고도화
- CI/CD 파이프라인 설계 및 구현
- AI 플랫폼 및 데이터 파이프라인 설계 및 구축
- 통합 인증 및 모니터링 시스템 구축

업무상 강점

- Cloud Native 개발 : 컨테이너, Devops, MSA 등에 대해 꾸준히 학습하고 업무에 적용
- Trouble Shooting : 개발/운영 중 발생하는 문제는 영역 구분없이 시스템 전반에 걸쳐 원인 분석과 해결책을 찾고자 함
- 문서화 : 개발 관련 이슈나 특이사항들을 협업 툴에 정리 및 공유

| 주요 기술스택 : Python/FastAPI , Java/Spring , Postgresql , Docker/Kubernetes , Aws , Azure , React , Ansible , Terraform , Kubernetes , Kubeflow

학력사항

학교	전공	학점	기간	졸업상태
홍익대학교	정보산업공학과	3.41 / 4.50	2004.03 ~ 2012.02	졸업

병역사항

병역구분	군별	계급	제대구분	복무기간
군필	육군	병장	만기제대	2005.07 ~ 2007.07

경력사항

회사	부서	기간	직급	담당업무
SK 주식회사 C&C	에너지 Digital 그룹	2012.01.01 ~ 2021.06.04	수석	MIS 시스템 개발/운영
Seegene	디지털컨텐츠팀	2021.06.07 ~ 2023.04.07	과장	플랫폼 백엔드 개발/운영
LG CNS	AI 아키텍처팀	2023.04.17 ~	책임	Application Architect

자격/어학 사항

시험명	취득일	주최기관
CKA	2024.08.04	Cloud Native Computing Foundation (CNCF)
SQLD	2016.07.05	한국데이터진흥원
정보처리기사	2011.09.02	한국산업인력공단

주요 프로젝트 및 포트폴리오

(LG CNS)

(주)LG 경영정보 Agent AI 구축



- 목적 : 경영정보 연계 Agent AI 구축을 통한 업무효율성 증대
- 기간 : 2025년 5월 ~
- 역할 : Application Architect
- 기술 : Kubernetes, Docker, Exaone(LRM), NL2SQL(LLM), AgenticWorks, OpenSearch, Python/FastAPI, LangChain, LangGraph, FastMCP, MCP Inspector, Langfuse, LiteLLM, Loki, Prometheus, Grafana
- 구성원 : 11명 (AI 연구원 외)
- 성과 :
 - 전체 아키텍트 설계 및 구현 총괄
 - On-Prem Kubernetes 클러스터 신규 구축 (H100 GPU)
 - Object Storage, PV/PVC 등 저장소 설계 및 구축
 - Private 라이브러리, 컨테이너 이미지 저장소 구축
 - 오케스트레이터 및 MCP Tool 개발
 - Exaone LRM 모델 OpenAI Comptitable 관련 Troubleshooting
 - Langfuse 및 LiteLLM 을 활용한 Agent AI 답변 Tracing 체계 구현

KB 증권 정보계 차세대 - 내부분석환경 구축



- 목적 : 정보계 시스템 재구축
- 기간 : 2024년 11월 ~ 2025년 4월
- 역할 : Application Architect, Technical Architect
- 기술 : Kubeflow, On-prem Kubernetes, Cilium, NFS Provisioner, Harbor, Nexus, Keycloak, Java/Spring, Loki, Prometheus, Grafana
- 구성원 : 20명 (전체 프로젝트 구성원 200명 이상)
- 성과 :
 - Kubeflow 기반 내부분석환경 구축 리딩
 - 분석환경 거버넌스 정책 수립 및 Batch/Online모델 배포 가이드
 - On-Prem Kubernetes 클러스터 신규 구축 (H800 GPU Node 포함)
 - Object Storage, PV/PVC 등 저장소 설계 및 구축
 - Private 라이브러리, 컨테이너 이미지 저장소 구축
 - Keycloak 기반 SSO 구현 (LDAP 연동) **
 - 과제 관리 포털 시스템 설계
 - Jupyter Notebook Kernel dead 발생 시 담당자에게 알람 기능 개발

**Keycloak 기반 SSO 구현 상세 설명

- Kubeflow 1.9.1 버전부터 인증 체계가 크게 변경됨
(~~oidc-authservice~~ ➡ oauth2-proxy)
- 새롭게 적용된 인증방식 분석 후 Dex Upstream IDP 를 Keycloak으로 설정하는 방식으로 SSO 구현
- Istio Ingress Gateway ➡ OAuth2-Proxy ➡ Dex ➡ Keycloak 인증 흐름을 가지게 됨
- Kubeflow Slack채널에서 활동하던 중, Kubeflow 오픈소스 메인테이너로부터 Dex와 Keycloak 통합 가이드를 작성해 PR로 제출해 달라는 제안을 받아 프로젝트에 기여하게 되었음

[관련 링크]

<https://github.com/kubeflow/manifests/pull/3012>

<https://github.com/kubeflow/manifests/blob/master/common/dex/README.md>

-  **WOOYOUNG** 2월 24일 오후 6:24
@Julius von Kohout I used kubeflow 1.9.1 version and i'm not used oidc-authservice.
Interestingly, during the integration process between Dex and Keycloak, it only worked when the Client ID was set to `kubeflow-oidc-authservice`.
-  **Julius von Kohout (Kubeflow Consulting)** 2월 24일 오후 6:28
That is really interesting. Do you want to add this guide for Dex and keycloak to our repository? Maybe here in a readme <https://github.com/kubeflow/manifests/tree/master/common> that is liked from <https://github.com/kubeflow/manifests#dex> since we get too many requests regarding keycloak
You can just create a PR and tag me
-  **WOOYOUNG** 2월 24일 오후 6:38
Okay:) I'll work on it as soon as possible. (편집됨)
-  **Julius von Kohout (Kubeflow Consulting)** 2월 25일 오전 1:16
@WOOYOUNG thank you for the PR. can you sign your commit
<https://github.com/kubeflow/manifests/pull/3012/commits/2cb96f3c3d3051c4751491eae8ee8223ea6b9217> according to the DCO? (편집됨)
-  **WOOYOUNG** 2월 25일 오전 8:57
@Julius von Kohout Just finished!

신한카드 AI 플랫폼 구축



- 목적 : GenAI 플랫폼 구축 (RAG Pipeline, AI Service, LLMOps 등)
- 기간 : 2024년 09월 ~ 2024년 11월
- 역할 : Application Architect
- 기술 : Kubernetes , AWS(EKS) , DAP(AI Platform) , Java/Spring , Python/FastAPI , OpenTelemetry , AWS X-Ray , Amazon Cloud Watch , Prometheus , Grafana
- 구성원 : 50 ~ 60명
- 성과 :
 - 전체 시스템 아키텍처 설계
 - 통합인증, Data Pipeline, 통합 로깅/모니터링 상세 설계
 - OpenTelemetry 기반 Application 간 Tracing 체계 설계

LG U+ AI 플랫폼 고도화 및 포털 구축



- 목적 : 기존 AI 플랫폼(Violet) 통합 관리를 위한 포털 구축
- 기간 : 2024년 04월 ~ 2024년 09월
- 역할 : Application Architect, Technical Architect
- 기술 : Kubeflow, AWS(EKS), On-prem Kubernetes, Calico, MinIO, NFS Provisioner, Harbor, Nexus, Prometheus, Grafana, Keycloak, Java/Spring
- 구성원 : 10명
- 성과 :
 - ixi 도우미 모델 학습을 위한 On-Prem Kubernetes GPU(H100) 클러스터 신규 구축
 - Object Storage, PV/PVC 등 저장소 설계 및 구축
 - Private 라이브러리, 컨테이너 이미지 저장소 구축
 - 인증 통합을 위한 사내 LDAP 연동
 - 과제 관리 포털 시스템 설계

스타벅스 DW이관 및 CDP 포털 구축



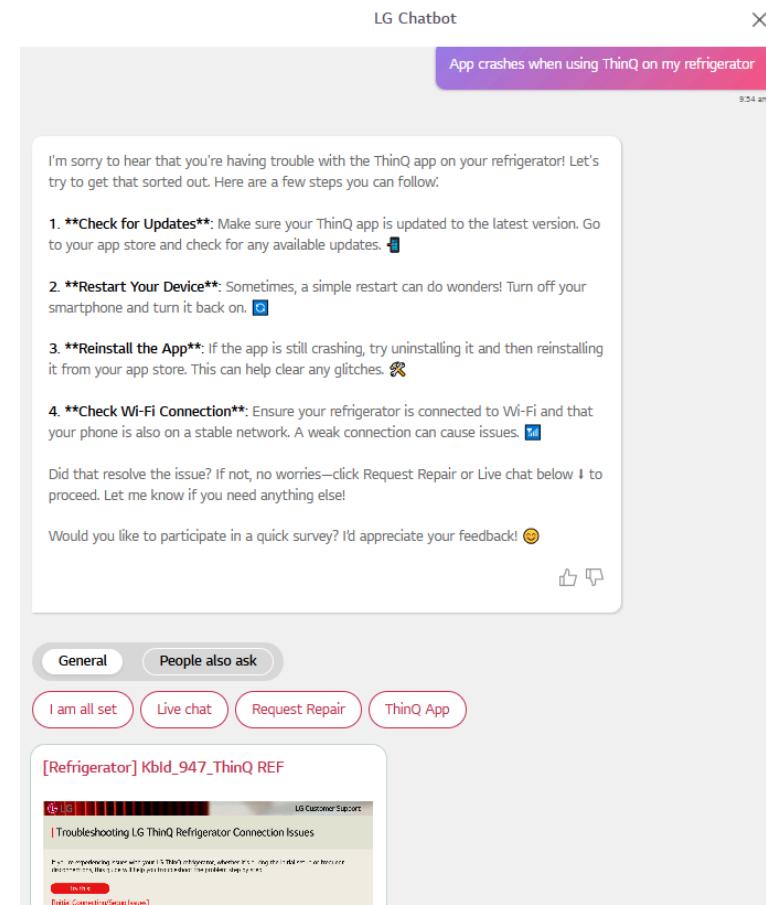
- 목적 : On-prem DW를 AWS 환경으로 이관하고 CDP 포털 신규 구축
- 기간 : 2024년02월 ~ 2024년04월
- 역할 : Application Architect
- 기술 : AWS Code Series
- 구성원 : 10명 (전체 프로젝트 구성원 : 90~100명)
- 성과 :
 - EC2, Fargate, EKS 환경에 배포되는 어플리케이션 CI/CD Pipeline 구축
 - Amazon MWAA**, Flink, Redshift에서 실행되는 코드들의 형상관리 및 승인 배포 체계 구축

**Amazon MWAA (Managed Workflows for Apache Airflow)

- 고객 요구사항에 맞게 Python 스크립트의 형상관리와 배포 시 승인 프로세스 절차가 반드시 필요했음
- MWAA의 경우 Airflow에서 지원하는 git sync를 사용할 수 없고 S3를 반드시 원천으로 사용해야 함
- CI/CD Pipeline에 git에 푸시된 코드를 변경 감지해서 변경분만 S3로 업로드하는 Job 추가하여 해결함

LG전자 미국법인 챗봇 ChatGPT 적용

- 목적 : 기존 Rule Base 챗봇의 한계를 극복하기 위해 RAG 기반 GPT 를 활용한 답변 시스템 구축
- 기간 : 2023년10월 ~ 2024년01월
- 역할 : Application Architect
- 기술 : Azure , AOAII ChatGPT , GCP , Docker , Python/FastAPI , Langchain , ElasticSearch(VectorDB) , MySQL 등
- 구성원 : 8명
- 성과 :
 - RAG 아키텍처 구성
 - App Framework 개발, Trouble Shooting, Dockerizing
 - GCP Cloud Build 를 활용하여 CI/CD Pipeline 구축
 - Load Balancing 및 Autoscale 설정
 - Azure와 GCP 간 HA Site-to-Site VPN 연결, DNS Forwarding 설정
 - Container-Optimized OS를 활용하여 심플하고 비용 효율적인 아키텍처 구성



담당 시스템 업무

(Seegene)

SG STATS



- 개요 : PCR 결과 데이터 분석 및 바이러스 변이 통계를 위한 시스템
- 목적 : 바이러스 및 변이 추세 모니터링 및 백신, 성별 등과 같이 다양한 요인에 의한 심층 데이터 분석 지원
- 기간 : 2021.08 ~
- 구성원 : 3명
- 담당업무

- 1 성능 고도화 프로젝트 진행 (중간집계 프로세스 도입, JPA 성능이슈 Trouble Shooting)
- 2 CI/CD Pipeline 구축
- 3 실험결과 데이터 대량 업로드 지원
- 4 설치형 시스템 개발 프로젝트 관리

• 사용기술

- 배포환경 : Azure VM (CentOS), Nginx, Tomcat 8.5
- Frontend/Backend : React / Java, Spring Boot
- DBMS : Postgresql
- 주요 라이브러리 : Spring Security, Spring Data JPA, MyBatis, GraphQL, Toastr Grid, SendGrid

중간집계 프로세스 도입

[Problem]

- 독일 고객의 실험데이터 대량 업로드 후 빈번한 시스템 장애 발생
 Frontend (차트 로딩 불가, JS Heap Memory Leak), BackEnd (Java Heap Memory Exception), DB (메모리/디스크 리소스 Full)
- 차트 쿼리 실행 시 테이블 Full Scan이 자주 발생했고, With절을 여러 개 선언한 후 With절 간 참조해 사용함으로써 심각한 성능저하 유발

[Solution]

- Backend : Old 영역으로 넘어가는 객체 수를 최소화하고 Full GC 시간을 줄이기 위한 적절한 GC 튜닝 옵션을 찾기 위해 모니터링 & 설정 반복
- DB
 - 설계 : 요구사항을 재정의해 각 메뉴에 맞는 중간집계 로직구현 및 데이터 스펙 정의
 - 검증 : Materialized View 를 활용해 중간집계 프로세스 검증
 (Concurrently 옵션으로 동시성 이슈를 해결했고, 조회조건에 맞게 Index를 생성해 성능 최적화)
 - 최종구현 : 실험결과 업로드 시 Message Queue에 작업을 할당하고 Spring Scheduler 를 사용해 중간집계 테이블 데이터를 생성하도록 개발

JPA 성능이슈 Trouble Shooting

[Problem]

- N+1 문제가 시스템 전반적으로 발생 (@Service 계층)
- API 데이터 탑재를 정의할 때 Entity 객체를 사용해 N+1 문제가 @Controller 계층에서 한번 더 발생
- JPA Native Query 사용해 비슷한 유형의 쿼리가 중복되는 경우 발생

[Solution]

- ToOne 관계는 Fetch Join,ToMany 관계는 Batch Size 를 설정해 최소한의 쿼리로 데이터를 조회하도록 했고, 필요에 따라 MyBatis SQL Mapper를 사용하기도 했음
- Open-Session-In-View 를 false 로 설정하고 @Service 계층에서 DB Connection Pool을 반납하기 위해 Entity 를 Dto 로 변환해 @Controller 계층에 반환하도록 수정했음
- MyBatis 동적쿼리 및 Include 를 활용해 권한처리나 공통 조건 같은 부분을 효율적으로 처리

CI/CD Pipeline 구축

[BackGround]

- 기존에는 운영자 PC에서 빌드 후 서버 SSH 접속해 수동배포하는 방식으로 운영

[Task]

- Git Branch 전략에 따라 dev 브랜치 업데이트 시 소스코드 빌드 후 개발 환경에 배포되고, prod 업데이트 시 담당자 승인 후 운영 환경에 배포되도록 구성
- Azure Devops에서 제공하는 에이전트를 사용해 개발/운영 서버에 배포(파일전송)를 하려면 서버의 방화벽을 모두 풀어야 했기 때문에 자체 호스팅 에이전트(서브PC)를 사용해 Pipeline 구성

실험결과 데이터 대량 업로드 지원

[BackGround]

- PCR 실험 결과를 확인하고 SG STATS 서버로 데이터를 올리기 위해서 실행해야 하는 프로그램은 총 3가지
CFX96 Manager 장비로부터 데이터 추출 ➡ Seegene Viewer 씨젠시약제품 실험결과 확인 ➡ Uploader SG STATS 서버로 데이터 업로드
- 프로그램 간 연동은 Excel Export 파일로 데이터를 전달하고 있어 다양한 통계 및 데이터 분석에 대한 니즈가 있는 고객들조차 데이터를 쉽게 업로드 할 수 없는 환경

[Task]

- 데이터를 올리기 위해 연구원이 수행해야 할 작업을 Python 자동화 스크립트로 구현
- PyAutoGUI 를 활용해 클릭하고자 하는 Target 이미지 찾기, 단축키, 화면 좌표값 등 다양한 방법을 활용
- 로직이나 구현방법 자체는 어렵지 않았지만 다양한 대형 검사소들의 데이터를 작업하면서 다양한 제품별 데이터 특성을 파악할 수 있었음

SG Archive



- **개요** : 제품별 인허가 문서, 논문, 소프트웨어, 홍보물 등 각종 산출물을 관리하는 시스템
- **목적** : 체계적인 산출물 관리와 대리점 및 KOL(Key Opinion Leader)들에게 자료를 효율적으로 공유
- **기간** : 2021.07 ~
- **구성원** : 2명
- **담당업무**

1 Azure Kubernetes Service 기반 인프라 구축 및 운영

2 CI/CD Pipeline 구축

3 Azure AD 연동해 임직원 SSO 로그인 기능 구현

- **사용기술**

- 배포환경 : Kubernetes(AKS)

- Frontend/Backend : React , React Native / Java , Spring Boot

- DBMS : MariaDB

- 주요 라이브러리 : Spring Security , MyBatis , SendGrid

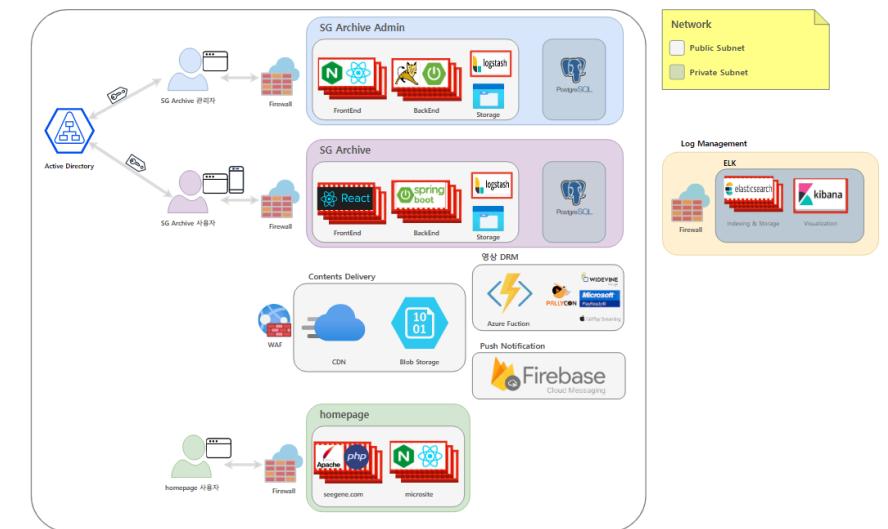
Azure Kubernetes Service 기반 인프라 구축 및 운영

[BackGround]

- 팀 내 요구사항을 만족하는 클라우드 서비스 Research
 - 중점 검토 사항** : 비용 효율, 안정적인 서비스, Scale Out 지원, CI/CD 구성 용이, Static Public IP 지원, Multi Port 지원
 - 대상 Application** : 사용자 시스템 Front/Back, 관리자 시스템 Front/Back, App API 등 (추후 홈페이지 이관 예정)
- Azure Container Instances > Azure Web App for Containers > Azure Kubernetes Service 순서로 검토 진행
- Azure Container Instances 는 간단하게 Container 를 실행시켜 테스트 해볼 수 있는 수준의 서비스였고,
- Azure Web App for Containers 사용 시 AKS를 사용하는 것보다 비용이 더 발생할 것으로 예상되어 최종적으로 **Azure Kubernetes Service (AKS)** 선택

[Task]

- Helm 차트를 활용해 Kubernetes Cluster 구축 후 각 Application 에 맞는 Manifest 파일 설정
- DNS 설정, 외부 서비스 연동을 위한 방화벽 설정
- Container 로그 수집 설정 (ELK Stack)



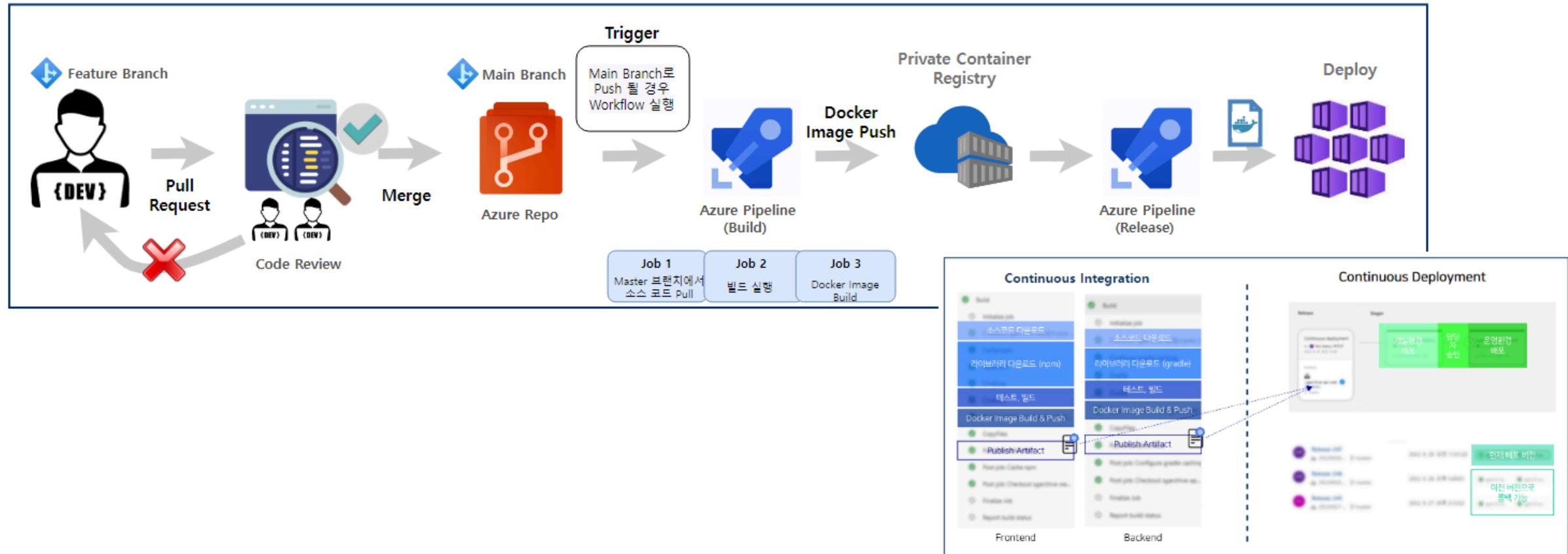
CI/CD Pipeline 구축

[BackGround]

- 기존에는 운영자 PC에서 빌드 후 서버 SSH 접속해 수동배포하는 방식으로 운영

[Task]

- Git Branch 전략에 따라 dev 브랜치 업데이트 시 소스코드 빌드 후 개발 환경에 배포되고, prod 업데이트 시 담당자 승인 후 운영 환경에 배포되도록 구성
- npm, gradle cache 설정



임직원 SSO 로그인 기능 구현 (Azure AD 연동)

[BackGround]

- 기존 로그인 방식 : ID/Password 입력 시 사용자 인증 후 JWT Token 발급
- 보안을 위한 Password 복잡도 규칙 및 주기적 교체 설정이 불편하다는 요청사항 접수 후 내부임직원을 대상으로 SSO 제공하기로 결정
- 임직원 정보를 관리 중인 Azure AD의 인증 서비스는 OIDC(Open ID Connect) 방식으로 제공됨
- Azure에서 Spring 환경에서 사용할 수 있는 라이브러리를 제공하지만 Spring 버전 이슈로 사용 불가
(`spring-cloud-azure-starter-active-directory` stable 버전 라이브러리는 Spring boot 2.5 버전 이상에서 유효했지만 해당 시스템은 2.2 버전 사용 중)

[Task]

- SG Archive 인증 관련 소스코드 분석
- Azure에서 제공하는 `spring-cloud-azure-starter-active-directory` 라이브러리 소스코드 분석 후 OIDC 방식 인증 구현
 - OAuth2UserService 인터페이스 구현 후 loadUser 재정의
 - `Multi Provider 적용` : 2가지 인증 방식(ID/Password, SSO)을 처리하기 위해 SSO 용 WebSecurityConfigurer 추가 정의 후 Order 정의
- 타 팀에서 운영중인 시스템에도 적용 가능하도록 BoilerPlate 코드 및 문서 작성

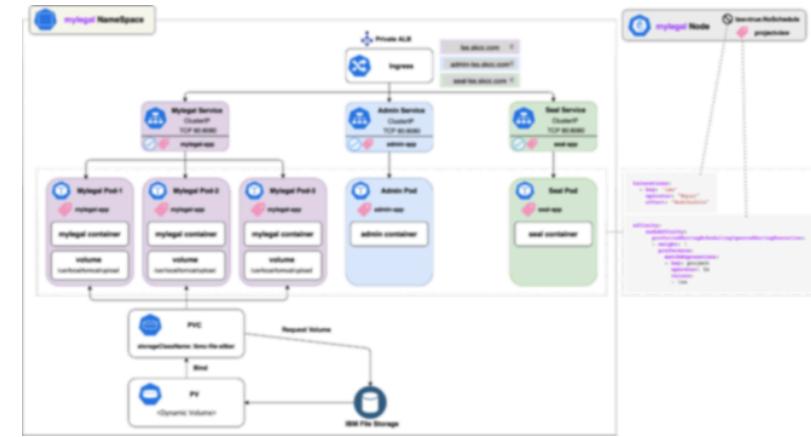
주요 프로젝트 및 포트폴리오

(SK 주식회사 C&C)

사내 법무지원 시스템 개발



- **개요** : 법무 및 IP(특허) 관련 업무 지원
- **목적** : 파편적으로 관리되던 법무 기능(계약검토, 자문, IPR)을 시스템으로 통합 구현
- **기간** : 2020.03 ~ 2020.06
- **구성원** : 8명
- **담당업무**
 - 공통 모듈 개발, 인증 및 인가 처리, Redis를 활용한 Session Clustering
 - 오픈소스, 솔루션 도입 검토 및 적용
 - Cloud 배포를 위한 Dockerfile, Kubernetes Configuration, Jenkins Script 작성
- **사용기술**
 - 배포환경 / 웹서버 : Cloud Z CP(Kubernetes) / Tomcat 8.5
 - 언어 및 프레임워크 : Java, Spring Boot
 - DBMS : Maria DB, Oracle, Redis(Session, Cache)
 - 주요 라이브러리 : Spring Security, Spring Session, MyBatis, Thymeleaf, Jquery, Toast Grid, CKEditor, KUpload, INISAFE, Nexess(SSO)



[Kubernetes]

- 법무시스템 외에 관리자 시스템, 인장수납 시스템도 서비스 하기 위해 각 시스템마다 Pod 생성 후 Service, Ingress 연결
- 사용량이 많은 것으로 예상되는 시스템에 HorizontalPodAutoscaler 설정
- Dynamic Provisioning 기능을 활용해 Pvc 만 선언하면 ibm file storage 와 Pv 가 연결될 수 있도록 설정
- Taint/Tolerance, NodeAffinity 를 통해 사내 클러스터 중 법무시스템 전용 노드에만 Pod 가 생성될 수 있도록 설정

[Backend]

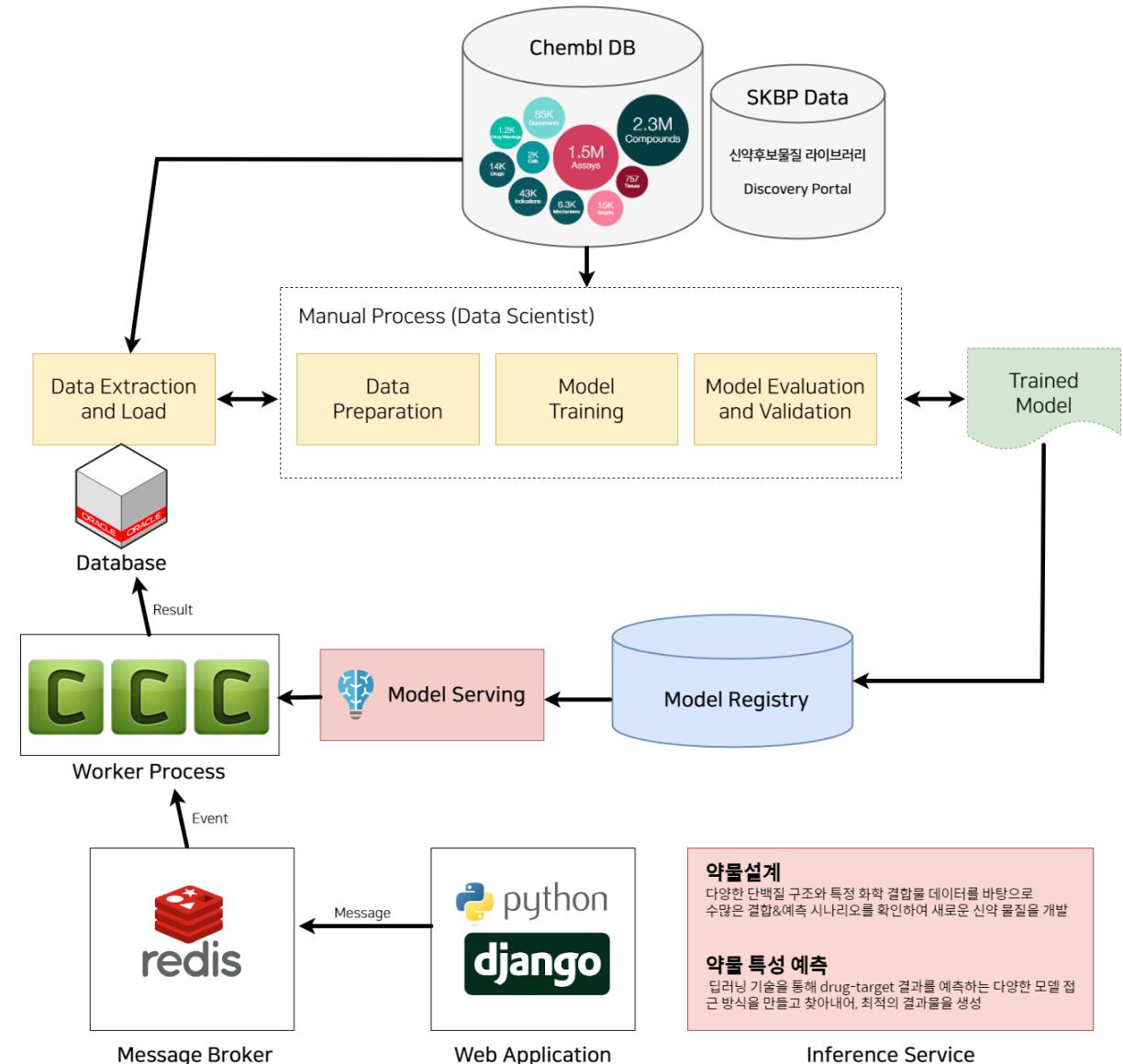
- 메이븐 멀티 모듈 구조를 사용해 공유할 수 있는 부분은 core 로 빼서 코드 중복을 줄이고자 했음
- 인증/인가 구현 시 사내 SSO와 연동하기 위해 Authentication Provider를 커스터마이징 했고, 다른 도메인 간 cookie 공유문제, 암복호화 모듈 관련 Trouble Shooting 활동 진행
- 오픈소스나 솔루션 적용 시 공통모듈화 해 개발
- Redis를 활용해 Session Clustering 을 적용했고, 공통코드 같은 데이터들에 대해 Cache를 적용해 조회 성능을 개선했음
- Dockerfile, Kubernetes manifest, 젠킨스 파이프라인 스크립트를 작성해 사내 CI/CD 툴을 통해 원클릭으로 롤링 업데이트 배포가 가능하도록 설정

[프로젝트 후]

- 소스코드 및 클라우드 관련 노하우를 Asset화 해 [타 관계사 법무지원 시스템 개발 프로젝트에 활용](#)
- Cloud ZCP 와 Trouble Shooting 내용 정리해 본부 세미나에서 공유
- Toast Grid 사용 중 버그를 발견해 이슈 제기한 건이 개선 사항으로 받아들여졌음  [관련 Github Issue](#)

AI 약물설계 플랫폼 개발/운영

- 개요 : AI 기반 신약 개발 시스템
- 목적 : 연구원들의 신약개발기간 단축
- 기간 :
 - 개발 : 2018.03 ~ 2018.08
 - 운영 : 2018.08 ~ 2020.06
- 구성원 : 10명
- 담당업무 : 화면개발, 데이터 수집 및 정제 작업
- 사용기술
 - OS / 웹서버 : CentOS 7 / NginX
 - 언어 및 프레임워크 : Python3.6.4 , Django , Jquery
 - DBMS : Oracle
 - 기타 : Celery , Redis , Tensorflow , Cuda , Pyenv



[주요 사항]

- 주기적으로 데이터를 수집하는 모듈과 AI 모델 Prediction 작업 기능 개발
 - 시간이 오래 걸리는 작업들이라 Celery를 활용해 비동기 처리
 - 대량으로 Prediction 작업을 할 경우에는 다수의 celery worker 프로세스를 활용해 수행되도록 구현
 [연관된 Blog 글](#)
- 복잡한 운영환경 정리
 - 아나콘다 사용 시 간단하게 설치해 사용하는 라이브러리들도 운영 서버에 세팅 시 충돌 및 설치 오류 발생
(특정 파이썬 라이브러리는 `pip install` 을 통해 설치가 안되어서 강제로 파일을 넣어주는 등의 설정 필요;;)
 - 복잡한 설정들을 DockerFile로 정의해서 배포
 - RDKit(Open-Source Cheminformatics Software) 관련 라이브러리는 소스를 직접 다운받아 빌드해 사용할 수 있도록 개선
 [연관된 Blog 글](#)

로그분석 시스템 개발



- 개요 : SK 주식회사 홀딩스 MIS 시스템 로그분석 시스템 개발
- 목적 : DB에 쌓여있는 로그들을 수집 및 분석해 운영자들의 운영 편의성 증대
- 기간 : 2019.06 ~ 2019.11
- 담당업무
 - 시스템 아키텍처 설계
 - ELK Stack Install
 - Alert 기능 개발
- 사용기술
 - 언어 및 프레임워크 : ELK(ElasticSearch, Logstash, Kibana) , ASP.NET
 - 기타 : Elastic NEST API
- 연관된 Blog 글

SK DMS 시스템 개발 / 운영



- **개요** : 그룹 연간 공시를 위해 데이터를 취합하고 작성하는 시스템
- **목적** : 공시자료 취합 및 작성 편의성 증대, 업무 자동화
- **기간** : 2018.02 ~ 2018.06
- **담당업무**
 - 시스템 아키텍처 설계, 계정관리 기능 개발, Pivot 형태의 화면 개발
 - 전자공시시스템(Dart)에 공시된 이전 공시자료를 Scraping 해 현재 취합중인 데이터와 비교 분석하는 기능 개발
(기존에 1~2일이 걸리던 작업을 자동화)
- **사용기술**
 - OS / 웹서버 : Windows Server 2012 / IIS
 - 언어 및 프레임워크 : ASP.NET MVC5, Dapper, Jquery
 - DBMS : MS-SQL
 - 기타 : IdentityFramework, Alopex(grid tool), HtmlAgilityPack

IFRS 가이드 시스템 개발



- **개요** : IFRS 가이드 문서 DB화
- **목적** : WORD로 된 IFRS 가이드 문서를 DB화 해 관리편의성 및 사용성 증대
- **기간** : 2018.10 ~2019.02
- **담당업무** : 문서 파싱 모듈 개발
- **사용기술**
 - OS / 웹서버 : Windows Server 2012 / IIS
 - 언어 및 프레임워크 : ASP.NET WebForm , Dapper
 - DBMS : MS-SQL
 - 기타 : OpenXml , HtmlAgilityPack
- **연관된 Blog 글**

운영 시스템

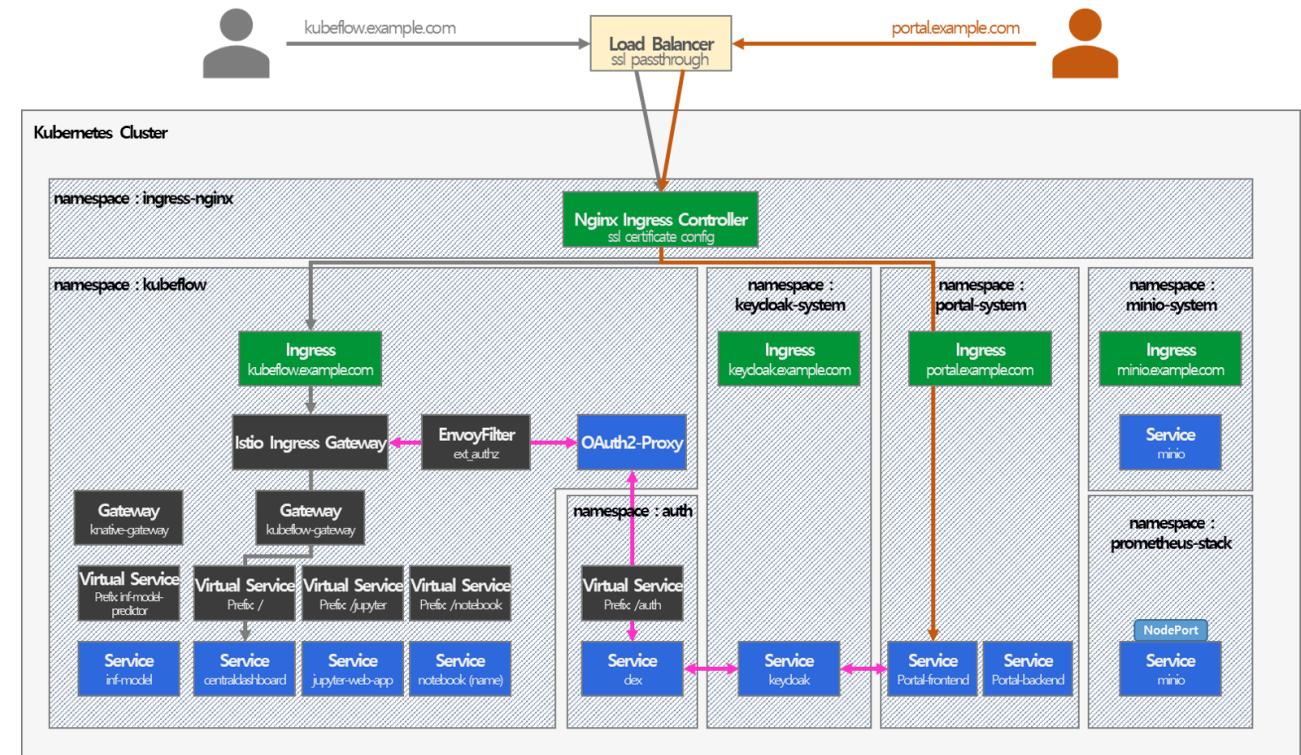
시스템명	기간	사용기술
SK바이오팜 AIDD	2018.09 ~ 2020.06	Python3.6.4, Django MTV, Django REST, Oracle
SK머티리얼즈 법무포털시스템	2019.01 ~	JSP, Servlet, MS-SQL, Tomcat
그룹채용시스템	2014.01 ~ 2018.06	ASP.NET, MS-SQL, IIS
고용디딤돌시스템	2015.07 ~ 2017.09	ASP.NET, MS-SQL, IIS

파일럿 프로젝트

Enterprise MLOps



- 개요 : Kubeflow 기반 Enterprise MLOps 환경 구축
- 목적 : 머신러닝 모델 개발, 배포, 관리의 효율적 운영 체계 구현
- 기간 : 2024.10 ~ 2025.03
- 담당업무 : MLOps 인프라 설계 및 실습 환경 구축
- 사용기술
 - 배포환경 : AWS, GCP, Terraform, Kubernetes, Docker
 - 언어 및 프레임워크 : React, Python/Flask, Kubeflow
 - 스토리지 : MinIO, PVC
 - 모니터링/로깅 : Prometheus, Loki
 - 인증 : Istio, OAuth2-Proxy, Keycloak, Dex



<https://github.com/wooyoung85/mllops>

MSA Project

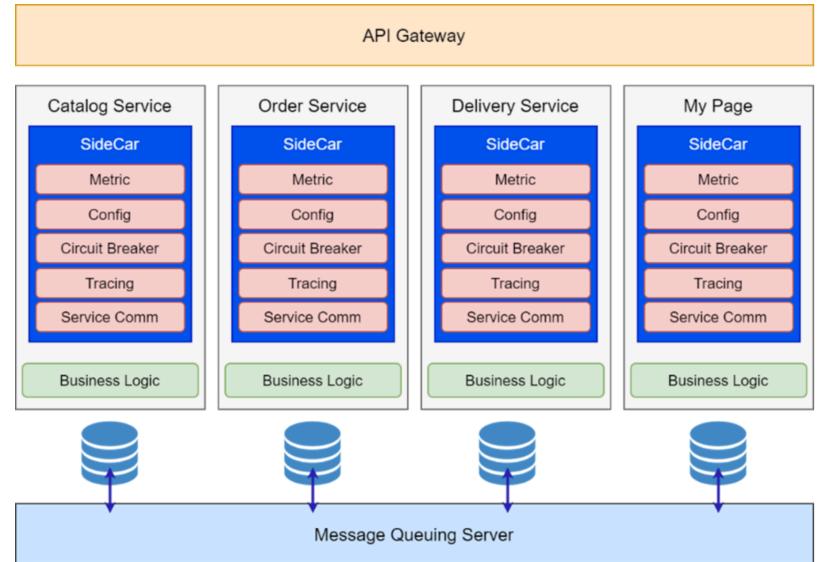


- **개요** : Microservice Architecture 기반 E-Commerce App 구축
- **목적** : Istio 활용한 효율적인 MSA Application 개발
- **기간** : 2022.08 ~
- **담당업무** : 시스템 아키텍처 설계 및 개발
- **사용기술**
 - 배포환경 : Kubernetes + Istio, AKS
 - 언어 및 프레임워크 : Java, Spring Boot, Spring Cloud
 - DBMS : MariaDB
 - AMQP : RabbitMQ, Kafka

[주요내용]

- **1단계** : Spring Cloud 를 활용해 MSA Application 구축
- **2단계** : Spring Cloud 설정 Code, Properties, Annotation 제거
- **3단계** : Kubernetes + Istio 환경에 배포

<https://github.com/wooyoung85/msa-project>



고려사항	Spring Cloud	Kubernetes + Istio
API Gateway	Netflix Zuul, Spring Cloud Gateway	Istio Envoy
Load Balancing	Netflix Ribbon	Istio Envoy
Service Discovery	Netflix Eureka	Kubernetes Service & Ingress
Service Invocation	Netflix Feign, RestTemplate	call Service Name
Security	Spring Security	Istio Envoy
Circuit Breaker	Netflix Hystrix	Istio Envoy
Tracing	Zipkin Server	Istio Jaeger
Metric	Hystrix Dashboard + Turbine Server	Prometheus, Grafana
Configuration	Spring Cloud Config	ConfigMap

Digital Play Ground



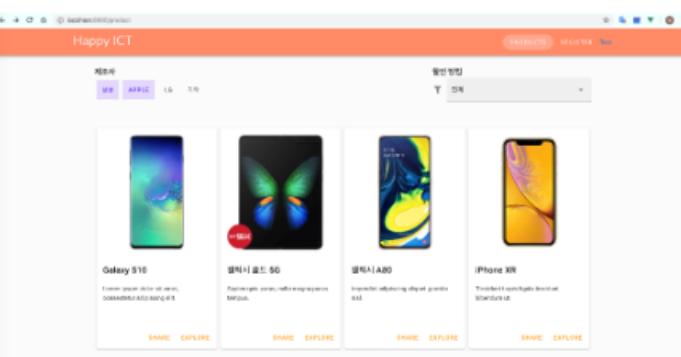
- 개요 : 누구나 언제나 Digital Play 할 수 있는 Quick VM 제공
- 목적 : 사내공모전 Pilot 시스템 개발
- 기간 : 2019.06 ~2019.09
- 담당업무 : 시스템 아키텍처 설계 및 개발
- 사용기술
 - OS / 웹서버 : CentOS 7 / NginX
 - 언어 및 프레임워크 : Vue.js , Node(express)
 - 기타 : OpenStack , Apache Guacamole
- 시연영상 (<https://player.vimeo.com/video/381607706>)



Frontend(Vue.js) 강의형 Study



- **개요 :** 행복 ICT 개발자들과 함께한 Frontend 강의형 Study
- **목적 :** Frontend(Vue.js)에 대한 기술을 익히고 사업에 적용
- **기간 :** 2019.10 ~ 2019.12
- **담당업무 :** 강의자료 작성, 샘플 프로젝트 개발
- **사용기술**
 - 언어 및 프레임워크 : [Vue.js](#), [Spring Boot](#)
 - DBMS : [Postgres](#)
- **관련자료**
 - [강의자료 - 블로그](#)
 - [샘플 프로젝트 - github](#)



Vue.js 실전개발 - Vue Router, Vuex, Axios

Vue Router Server-side Routing(MPA) VS Client-side Routing(SPA) MPA SPA 이미지 출처 : Vue Mastery-facebook Server-side Routing(MPA) client 가 URL을 변경 요청할 때마다 서버로 요청이 가고 서버로부터 페이지를 Return 받는 방식 Client-side Routing(SPA) SPA에서 컨텐츠를 탐색하기 위한 방식 (SPA는 하나의 페이지에서 모든 리소스들을 로드한 후 ...

Vue.js · 2019. 12. 10. 13:39

Vue.js 실전개발 - Vue CLI, Vuetify

Vue CLI Vue.js 개발 환경을 빠르고 쉽게 제공 공식 지원되는 도구 project scaffolding 제공 (👉 프로젝트 폴더 구조 및 라이브러리 설정) Vue CLI를 쓰지 않고 직접 설정도 가능하지만 추천 ❌ 현재 v4.0.0-rc.7 까지 Pre-release 된 상태이지만 3.x version을 사용할 예정입니다. 게으른 탓인지 글을 쓰는 중에 4.x version이 공식 릴리즈 되었습니다 😅 4.x version을 사용해야...

Vue.js · 2019. 12. 10. 13:31

Vue.js 기초 - Components, Vue Instance Life Cycle

Components 컴포넌트란? 재사용 가능한 코드 블럭 모듈화 및 관리가 용이한 코드베이스를 만드는데 도움이 될 컴포넌트들을 조합하여 어플리케이션 개발 컴포넌트들은 부모 자식 관계로 트리 구조 형성 이미지 출처 : Cracking Vue.js 컴포넌트는 SPA(Single Page Application) 개발 시 가장 기본적이고 중요한 요소임 전역 컴포넌트 Vue.component("product", {}); 첫번째...

Vue.js · 2019. 12. 10. 13:28

Vue.js 기초 - Directives, Vue Instance 속성

👉 사전 준비 쇼핑몰 예제 코드 작성 (Step_0) 예제 소스 내려받기 \$> git clone https://github.com/wooyoung85/vuejs-study.git \$> cd vuejs-study/example ## Visual Studio Code 실행 \$> code . 템플릿, 보간법, Mustache 구문, Interpolation ({{}}) I have a {{ product }} {{ pr...

Vue.js · 2019. 10. 18. 16:21

Vue.js 소개

Vue.js란? progressive framework for building user interfaces 2014년 11월 9일에 Vue.js v0.11 Release 후 현재 v2.6.10 까지 Release 될 vue - Roadmap | github Google Creative Lab에서 일하던 EVAN YOU가 개발 현재 가장 주목받고 있는 Frontend Framework 중 하나 이미지 출처 : risingstars.js.org npm-stat 확인 npm-stat.com 사용량으로 보면 Re...

Vue.js · 2019. 9. 24. 13:59

PC Sell



- **개요** : 실제 운영 중인 간단한 업무 시스템을 참고해 가상의 사이트를 개발
- **목적** : Spring Boot 웹 개발 및 통합테스트 자동화 스터디
- **기간** : 2017.08 ~ 2017.11
- **담당업무** : 시스템 아키텍처 설계 및 개발
- **사용기술**
 - 언어 및 프레임워크 : Spring Boot , Geb(Groovy기반 웹 통합 테스트 라이브러리)
 - DBMS : H2
 - 주요 라이브러리 : Thymeleaf , Mustache , Jquery
- **관련자료**
 - 프로젝트 : <https://github.com/wooyoung85/pcsell>
 - 통합 테스트 : <https://github.com/wooyoung85/pcsell-integration-test>

Multi Tenancy 평가 시스템

1

- **개요** : 관계사마다 필요한 평가 시스템을 Multi Tenancy 서비스 형태로 개발
- **목적** : Multi Tenancy 웹 어플리케이션 개발 방법 스터디
- **기간** : 2017.11 ~ 2018.03
- **담당업무** : 시스템 아키텍처 설계 및 개발
- **사용기술**
 - 언어 및 프레임워크 : Spring Boot
 - DBMS : H2 , MYSQL
 - 주요 라이브러리 : Thymeleaf , Mustache , Jquery
- **관련자료**
 - [프로젝트](https://bitbucket.org/cloudstudy) : <https://bitbucket.org/cloudstudy>

[주요내용]

- URI 로 Tenant를 식별하는 전략을 사용했고, DataBase는 같은 DB 내에서 Tenant별로 다른 Schema를 사용하는 전략 사용
- Spring의 HandlerInterceptor를 활용해 DispatcherServlet으로 가기 전에 Tenant를 식별
- Hibernate가 제공하는 CurrentTenantIdentifierResolver 인터페이스를 구현한 클래스에 Tenant정보를 전달해, AbstractMultiTenantConnectionProvider를 상속한 클래스가 Tenant 별로 DataSource를 다르게 처리하도록 구현

감사합니다