|  |  |
| --- | --- |
| **진기태**  **Backend Engineer** | **Phone |** 010-5570-9143  **Email |** msjinkitea@gmail.com  **Github** | <https://github.com/sensesis>  **Blog |** <https://kittae.tistory.com/> |

**Summary**

* **Spring Boot 및 Django 기반 서버를 구축하고, 클라우드 환경에서 서비스 배포 경험이 있습니다.**  
  **Docker와 Jenkins**를 활용한 **CI/CD 파이프라인을 구성하여 무중단 배포 환경을 구현한 경험이 있습니다.**
* **RabbitMQ와 Celery**를 활용한 **비동기 및 병렬처리**를 통해 사용자 경험을 크게 개선시킨 경험이 있습니다.
* **기획부터 배포까지 다양한 End-to-End 프로젝트를 수행하며 전체 개발 프로세스를 경험했습니다.**  
  **Backend 및 DevOps 개발을 주도**하며, **효율적인 협업과 팀원들과의 소통을 중요하게 생각합니다.**

**DevSketch** 2024.12 ~ 2025.02 [Medium](https://medium.com/@qkrb8019/%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-%EC%84%A4%EA%B3%84%EB%B6%80%ED%84%B0-%EC%83%9D%EC%84%B1%EA%B9%8C%EC%A7%80-ai%EA%B0%80-%EB%8F%84%EC%99%80%EC%A3%BC%EB%8A%94-devsketch-2024-siliconvalley-winter-bootcamp-8f69595ec7a1) [GitHub](https://github.com/sensesis/DevSketch)

사용자가 입력한 내용을 바탕으로 프로젝트의 설계부터 개발, 배포까지 자동화 해주는 AI기반 웹 서비스

**Backend | Django, RabbitMQ, Celery, Radis, MySQL**

**Infra | Docker, Traefik, GCP(GCE, Google Cloud), Prometheus, Grafana, Jenkins**

**기여 내용 |**

**[AI 설계 기능 구현]**

* **문제:** 프로젝트 설계 자동화 기능에서 ERD, 다이어그램, API 문서를 생성하는 과정에서 대량 문서 처리 시 성능 저하 발생. 기존 동기 방식에서는 평균 **62**초가 소요되어 사용자 경험 저하
* **해결:**
* **Celery + RabbitMQ**를 활용하여 대량 문서 생성을 **비동기 처리**하도록 변경
* 기존 방식 대비 **45% 속도 개선(62초 → 34초)**하여 사용자 대기 시간 단축
* **Redis를 도입해 메시지 큐 상태를 실시간 모니터링**, 트래픽 급증 시에도 안정적인 워크플로우 유지

**[가상 환경 자동화(Docker-in-Docker, DinD)**]

* **문제:** 
  + 개발자가 환경을 개별적으로 설정해야 하며, 테스트 환경이 일관되지 않아 배포 시 오류 발생.
* **해결:** 
  + **Docker-in-Docker(DinD) 기술 적용** → 컨테이너 내부에서 사용자별 가상 환경 자동 생성
  + 개발자가 직접 환경 설정할 필요 없이 **즉시 코드 실행 가능**하도록 변경
  + 실제 배포 환경과 동일한 환경에서 테스트 가능하여 **배포 과정에서 오류 발생률 감소**
* **문제:** 
  + 사용자의 가상 서버(Docker 컨테이너)가 계속 실행되면서 **서버 리소스가 과도하게 소비**됨.
  + 장기간 실행된 컨테이너가 **CPU와 메모리를 점유**하여 다른 요청 처리 속도 저하 발생.
* **해결:**
  + **컨테이너 실행 시간을 30분으로 제한**하여 일정 시간 이후 자동 삭제 정책 적용.
  + 불필요한 리소스 점유를 방지하여 **서버 부하를 줄이고 시스템의 안정성 유지**.
  + **Docker 컨테이너의 Lifecycle 관리 최적화**, 불필요한 컨테이너가 유지되지 않도록 정기적인 정리 프로세스 추가.
* **문제:** 
  + 각 서브 도메인(sub1.example.com, sub2.example.com 등)에서 HTTPS가 적용되지 않아 **보안 취약점 발생**.
  + 개별 인증서 적용은 유지보수 부담이 크고, 자동화되지 않아 효율성이 저하됨.
* **해결:**
  + **와일드카드 SSL 인증서(\*.example.com)를 적용**하여 모든 서브 도메인에서 HTTPS 활성화.불필요한 리소스 점유를 방지하여 **서버 부하를 줄이고 시스템의 안정성 유지**.
  + **Traefik과 Let's Encrypt를 활용해 인증서 자동 갱신**하여 유지보수 부담 최소화.
  + 클라이언트와 서버 간 **데이터 암호화로 보안성 강화**, Man-in-the-Middle 공격 방지.

**[실시간 Monitoring 서버 구축]**

* **문제:**
  + 장애 발생 시 원인 분석이 지연되면서 대응 시간이 늘어나는 문제점이 발생.
* **해결:**
  + **Prometheus로 서버 및 애플리케이션 메트릭 수집**, Grafana 대시보드를 구축하여 **실시간 모니터링 환경 조성Traefik과 Let's Encrypt를 활용해 인증서 자동 갱신**하여 유지보수 부담 최소화.
  + cAdvisor를 활용해 Docker 컨테이너별 리소스(CPU, 메모리) 사용량 최적화
  + **장애 발생 시 즉시 알람을 통해 대응 시간 1시간 → 10분 내로 단축**, 운영 효율 향상

**[CI/CD 파이프라인 구축]**

* **문제:** 
  + 코드 푸시 후 수동 빌드·배포로 인한 운영 부담 증가 및 서비스 중단 발생.
* **해결**
  + Jenkins + Docker 연동하여 Git 푸시 시 자동 빌드 및 배포 적용cAdvisor를 활용해 Docker 컨테이너별 리소스(CPU, 메모리) 사용량 최적화
  + **Traefik을 활용한 서비스 간 라우팅 및 로드 밸런싱 자동화**, 무중단 배포 환경 구성
  + 배포 자동화 적용 후 **서비스 가용성과 확장성 극대화**, 운영 비용 절감

**SML** 2024.06 ~ 2024.08 Medium [GitHub](https://github.com/Summer-Bootcamp-2024/SML)

현직자들과의 만남을 위해 직접 기회를 얻을 수 있는 웹 서비스

**Backend | FastAPI, Radis, PostgreSQL**

**Infra | Docker, NginX, AWS(EC2, S3, RDS), Prometheus, Grafana, ELK Stack**

**기여 내용 |**

**[이촌 검색 기능 구현]**

* **문제:** 
  + 검색 시 유사한 이름이나 직책이 포함된 경우 결과가 일관되지 않거나 부정확한 데이터가 포함됨.
  + 기존 단순 문자열 검색 방식으로는 네트워크(1촌, 2촌) 관계를 반영한 검색이 어려움.
* **해결:**
  + **Analysis-Token 시스템을 활용하여 검색 키워드를 여러 개의 토큰으로 분리**, 유사 검색어 대응 및 검색 정확도 개선.**raefik을 활용한 서비스 간 라우팅 및 로드 밸런싱 자동화**, 무중단 배포 환경 구성
  + **1촌과 2촌 데이터를 함께 스코핑하도록 토큰 매칭 로직 적용**, 네트워크 관계를 반영한 검색 구현.
  + **Elasticsearch 유사 형태소 분석(Search Analyzer) 적용**, 다양한 검색 패턴(이름, 직책, 키워드 등)에 대한 정밀한 검색 지원
  + **검색 정확도가 증가하고, 불필요한 검색 결과를 줄여 사용자 경험 개선**.

**[로그 관리]**

* **문제:** 
  + 기존 로그 시스템에서는 **서버 오류 및 검색 성능 저하 문제 발생 시 원인 분석이 어려움**.
  + 장애 발생 시 로그 데이터를 수작업으로 분석해야 해 **운영 부담 증가**.
* **해결:**
  + **ELK Stack(Elasticsearch, Logstash, Kibana, Beats)를 활용하여 실시간 로그 수집 및 시각화 시스템 구축**.
  + **Docker 컨테이너 로그를 Beats를 통해 수집 → Logstash로 전송 → Kibana 대시보드에서 분석 가능하도록 구성**.**결과적으로 검색 정확도가 증가하고, 불필요한 검색 결과를 줄여 사용자 경험 개선**.
  + **검색 성능 저하 및 서버 장애 발생 시 실시간 로그 분석을 통해 즉각적인 원인 파악 및 대응 가능**.
  + 운영 부담 감소 및 문제 해결 속도 향상으로 **서비스 가용성 및 안정성 강화**.

**Skills**

**Language |** Java, Python

**Library / Framework |** Spring Boot, Django, FastAPI

**Database |** MySQL, PostgreSQL, Redis

**Monitoring |** Prometheus, Grafana

**Development Tools** **|** Docker, Nginx, Traefik, Jenkins, ELK Stack

**ETC |** RabbitMQ, Celery

**Activities**

**Techeer [테커]** | 2024.12 ~ 현재

● 실리콘밸리 개발자의 SW 개발자 커리어 그룹

● 프로젝트, 스터디, 네트워킹 행사 등 다양한 활동 참여

**부트캠프**

● [티타임즈 x Techeer] 2024 하계 실리콘밸리 SW 부트캠프 2024.06 ~ 2024.08

● [티타임즈 x Techeer] 2024 동계 실리콘밸리 SW 부트캠프(1등 수상 🥇) 2024.12 ~ 2025.02

**Education**

한성대학교 컴퓨터공학과 | 2020.03 - 2026.02(졸업예정)