



## 프로젝트

파란색 테두리 카드는 클릭하면 자세히 볼 수 있어요.

### 내부통제 요구사항을 반영한 운영 체계 구축

#### Jenkins 배치 통제 체계 구현

분산 운영 중이던 Jenkins 환경을 중앙 집중화하고, 배치 작업 전 과정에 내부 통제를 적용한 운영 체계를 구축

- 직무 분리 원칙 기반 권한 차등(등록/수정/실행) 및 승인·변경 이력 관리로 통제 강화
- 실행 로그 실시간 수집 + 성공/실패 자동 리포트로 감사 증적 확보

Jenkins   Python   ITGC   Compliance

#### 감사/증적 대응 보고 자동화

변경·실패 이력 데이터와 승인 번호를 자동 매칭하고, 현황을 실시간으로 가시화해 감사 대응 속도와 운영 효율을 높였습니다.

- 승인 완료 데이터 API 연동으로 리포트 상태 실시간 반영
- 월 단위 보고서 작성 6h → 1h(약 80%↓)로 단축

Elasticsearch   Kibana   Automation   Audit

#### 이슈 대응 기능 구현

긴급 상황 시 최소 권한으로도 잡 상태를 제어할 수 있는 기능을 도입하여 대응 속도를 향상시키고 통제 증적을 확보했습니다.

- 최소 권한으로 잡 상태(활성화/비활성화) 제어 기능 도입
- 모든 사용 이력 자동 저장으로 통제 증적 확보

Jenkins   Automation   ITGC

### 데이터 기반 문제 분석을 통한 구조 개선

#### 전금법망 Sentry 플랫폼 구조 개선

외부 통신이 제한된 전금법망 환경에서, Sentry의 안정성·처리 성능·확장성을 개선하기 위한 구조 개선 프로젝트를 추진했습니다.

- 이슈 생성 시간 1시간+ → 0~1분으로 단축
- 초당 처리량 3,000 TPS → 12,000+ TPS, Kafka Lag 2,800만 → 162만(94%↓)

Kubernetes Helm Sentry Kafka PostgreSQL

## 플랫폼 장애 리스크 최소화 및 무중단 전환

### 운영 리스크 최소화를 위한 Jenkins 구조 재설계

사용 부서 증가로 단일 Jenkins 인스턴스에 부하가 집중될 것으로 예상됨에 따라, 작업 지연과 운영 불안정을 예방하기 위해 Jenkins 인프라를 사전에 분리·재설계했습니다.

- 업무 특성 기준 3개 마스터로 분리 구축, 리소스 충돌 방지 및 안정성 확보
- 스크립트+Jenkins API로 600+ 잡 이관 자동화(정확도 향상/시간 단축)

Jenkins DR Automation Operations

## 정책 변화에 따른 신규 업무 환경 전환

### 정책 변화에 따른 신규 업무 환경 전환

정책 변화에 따른 인프라 환경 전환에 선제 대응하여, 팀이 안정적으로 적응할 수 있도록 실무 중심의 가이드와 협업 구조를 설계한 프로젝트입니다.

- 전환 가이드/연동 매뉴얼/이슈 대응 자료를 실무 중심으로 제작·배포
- 지속 가능한 협업 흐름과 운영 채널 구축으로 전환 과정 혼란 최소화

Docker Kubernetes Github Naver Cloud Transition Documentation

## 업무 자동화 도구 설계·개발

### 개발부터 운영까지 문제 해결 중심의 DevOps

개발-배포-운영-모니터링까지 전체 흐름을 연결해 문제의 원인을 파악하고, 자동화·표준화로 재발 방지와 지속 가능한 운영 체계를 구축했습니다.

- Cursor 등 AI 도구를 활용해 빠르게 분석·구현하고, 동작 가능한 솔루션으로 연결
- 자동화·대시보드·대응 잡 등 '구현 항목'을 누적해 운영 효율을 지속 개선

DevOps

Automation

Monitoring

CI/CD

AI

### AI 브라우저 확장앱 구현

요청서/이슈 형식이 불규칙한 환경에서 LLM을 활용해 코멘트 생성·등록을 자동화했습니다.

- 코멘트 작성 3분 → 평균 10초(약 96%↓), 주 100건+ 반복 업무 자동화
- 규칙 기반 파싱 + LLM 파싱 병행으로 케이스별 예외 대응

LLM

Browser Extension

Python

FastAPI

### 운영 알림·대시보드 자동화(알람봇/업무 대시보드)

업무 인입과 운영 현황을 실시간으로 공유하기 위해 알람봇과 Kibana 기반 대시보드를 구축하고, 자동 갱신 파이프라인까지 연결했습니다.

- 업무 인입 시 메신저봇으로 실시간 알림 자동 전송(알람봇)
- 업무 로그 시각화로 팀 주간 업무량/처리 현황을 한눈에 파악(Kibana)

Python

Lambda

GitHub Webhook

Works API

Elasticsearch

Kibana

Jenkins

### 장애 대응 자동화

운영 이슈의 감지·대응 속도를 높이기 위해 Jenkins 기반 대응 잡과 Kafka Offset 에러 자동 대응 프로세스를 구현했습니다.

- Jenkins 노드 모니터링 잡으로 장애 대응 속도 향상
- 활성화/비활성화 자동화 잡으로 즉시 대응 가능하게 구성

Jenkins

Groovy

Kafka

Linux

Shell Script

Kafka CLI

### 운영 안정성 자동화(DR·부하테스트·CI/이관)

운영 안정성과 배포 효율을 높이기 위해 DR 오탐 방지, 부하 테스트, 대량 이관, CI 기반 빌드/라벨링 자동화를 묶어 운영 체계를 강화했습니다.

- Jenkins DR 오탐 전환 방지 자동화(리전 간 단절로 인한 Failover 오탐 방지)
- Sentry 부하 테스트 앱 구현(초당 2,000+ 트래픽, Kubernetes Job/Docker/Python)

Python

Lambda

Kubernetes

Docker

GitHub Actions

Jenkins API

Harbor

## 문서화·교육을 통한 운영 체계 강화

### 가이드북 기반 업무 교육 시스템 구축

Sentry 실무 중심 가이드북을 직접 설계하고 Next.js로 구현해 교육·공유·협업이 자연스럽게 이루어지는 표준 운영 체계를 만들었습니다.

- Sentry·Helm·Kafka·PostgreSQL 등 구성요소별 실무 가이드 제공
- 신규 인원도 즉시 대응 가능한 표준 대응/트러블슈팅 체계 구축

[Next.js](#)[Documentation](#)[Sentry](#)[Education](#)