

이재철 | Server Developer

Phone 010-5757-9592 **Email** qws941@kakao.com **GitHub** github.com/qws941 **Portfolio** resume.jclee.me
Live Demos blacklist.jclee.me | splunk.jclee.me | grafana.jclee.me

Summary

7년간 대규모 인프라를 운영하며 “반복 작업을 기술로 치환”하는 데 집중했습니다. 수동 배포 2주→3일 단축, 보안 점검 오탐 95% 감소 등 **측정 가능한 성과**를 만들었습니다.

현재 토스 커머스에서 요구하는 **Kotlin/Spring을 선제 학습** 중이며, Python/Node.js 기반의 7년 서버 개발 경험을 바탕으로 빠르게 기술 스택을 전환할 준비가 되어 있습니다.

토스 커머스가 찾는 개발자와의 매칭

“지금 잘 작동하는 것”에서 멈추지 않고 “시간이 지날수록 더 나아질 수 있는 구조”를 고민합니다.

- **문제 정의 능력**: EPP/DLP 충돌 문제를 데이터 기반으로 분석하고 프로세스 격리로 근본 해결
- **자동화 마인드**: Python으로 작업시간 50~75% 단축, Ansible로 1,000대 VPN 3일 배포
- **안정성 추구**: 금융권 24/7 운영 환경에서 보안사고 0건, 장애 자동 복구 30초 이내
- **빠른 실행**: 당일 프로토타입, 1주일 내 Production 투입 가능한 실용주의

차별화된 강점

1. 완벽주의 & 품질 집착

- 보안사고 0건이 아니라, 15종 솔루션 통합 운영으로 0건 달성
- 오탐률 95% 감소가 아니라, 월 500건→25건 감소 (구체적 지표)
- 3회 이상 검증 - 배포 전 자동화 테스트, Grafana 실시간 검증, 롤백 시나리오

2. 실용주의 & 빠른 실행

- 복잡한 아키텍처보다 작동하는 시스템 우선 - Ansible로 1,000대 VPN을 3일 내 배포
- 불필요한 과정 제거 - 수동 점검 75% 자동화, 배포 시간 80% 단축
- 즉시 적용 가능한 솔루션 - Python/Node.js로 당일 프로토타입, 1주일 내 Production 투입

3. 시스템 사고 & 자동화 마인드

- **Grafana 기반 Full Observability** - Prometheus, Loki, Tempo 통합 모니터링
- **AI 기반 장애 분석 자동화** - Claude Code로 로그 패턴 자동 분류, 오탐 95% 감소
- **GitHub Actions CI/CD** - 코드 푸시 → 자동 테스트 → Grafana 검증 → 배포 전체 자동화

4. 비판적 사고 & 본질 추구

- **표면적 성과가 아닌 본질적 가치** - “어떻게”를 중시, 재현 가능한 시스템 구축
- **과대평가 배제** - GenAI 2년이 아니라 1년 실무 경험, 검증 가능한 주장만 기재
- **문제의 근본 원인 해결** - EPP/DLP 충돌 시 프로세스 격리 구조로 근본 해결

핵심 역량 (토스 커머스 요구사항 매칭)

서버 개발 역량

- **Python, Node.js 기반 서버 개발 7년** (Production 환경 다수 운영)
- **Kotlin/Spring 선제 학습 중** (Udemy 강의 완료, REST API 프로젝트 진행 중)
- **대규모 트래픽 환경** (대용량 이벤트 처리 시스템 구축 및 검증)

시스템 설계 역량

- 실시간성과 정확성을 만족하는 시스템 설계 (API 응답시간 < 100ms, 무중단 운영)
- 복잡한 도메인의 단순화 및 구조화 (작업시간 50~75% 단축)
- 안정성과 예측 가능성을 유지하는 아키텍처 (자동 롤백, 장애 분석 자동화)

데이터 & 인프라 역량

- 데이터베이스 설계 및 최적화 (PostgreSQL, MySQL, MongoDB, Redis)
- Platform Engineering (Prometheus, Grafana, Docker, Kubernetes, GitHub Actions)
- GenAI 기반 자동화 (Claude Code, LLM 활용 1년 실무 경험)

주요 성과

- 자동화로 작업시간 50~75% 단축 (수동 2주→3일)
 - Ansible로 1,000대 규모 VPN 자동 배포 (3일 완료)
 - 보안사고 0건 유지 (금융권 운영 20개월+, 15종 솔루션 통합)
 - AI 기반 오탐 95% 감소 (월 500건→25건)
-

Work Experience

(주)아이티센 CTS

정보보안 운영 엔지니어 | 2025.03 ~ 현재 (7개월)

넥스트레이드 운영SM (정보보안팀)

담당 업무

- 금융위원회 본인가 이후 운영 보안 체계 안정화
- 15종 이상 보안 솔루션 통합 운영 및 정책 최적화
- 보안 이벤트 모니터링 및 장애 대응 (24/7)
- 정기 보안 점검 및 취약점 관리

주요 성과 (문제 정의 → 해결 → 성과) 1. 보안 솔루션 통합 운영 안정화 - 문제: 15종 보안 솔루션 간 정책 충돌, 오탐 알림 월 500건+ - 해결: 솔루션 간 정책 우선순위 재설계, 통합 모니터링 대시보드 구축 - 성과: 오탐 알림 500건→100건 (80% 감소), 금융감독원 정기 감사 지적사항 0건

2. 24/7 실시간 모니터링 체계 구축 - 문제: 야간 보안 이벤트 대응 지연 (평균 30분+) - 해결: Grafana 기반 실시간 알림 시스템, 자동 티켓 생성 - 성과: 대응 시간 30분→5분 (83% 단축), 보안사고 0건 유지

Tech Stack DDoS, IPS, WAF, NAC, DLP, EDR, APT, VPN, WIPS, Grafana

토스 커머스 연관성 → 많은 요청과 변화하는 조건 속에서도 안정성과 예측 가능성을 유지하는 경험. 실시간 금융거래 환경에서 24/7 모니터링 및 즉시 대응 체계 구축.

(주)가온누리정보시스템

프리랜서 인프라 엔지니어 | 2024.03 ~ 2025.02 (11개월)

넥스트레이드 구축 프로젝트

담당 업무

- 금융위원회 본인가 대비 **망분리 및 보안 체계 구축**
- **15종 이상 보안 솔루션 통합 운영**
- NAC, DLP, SSL 복호화, APT 솔루션 도입 및 최적화

주요 성과 (문제 정의 → 해결 → 성과) 1. **방화벽 정책 배포 자동화** - 문제: 수동 정책 배포 시 2주 소요, 휴먼에러로 서비스 장애 발생 위험 - 해결: Python + FortiManager API 자동화 파이프라인 구축 - 성과: 배포 시간 2주→3일 (85% 단축), 정책 오류 0건, 작업시간 50% 단축

2. **EPP/DLP 충돌 문제 근본 해결** - 문제: 백신(EPP)과 DLP 솔루션이 네트워크 드라이버 레벨에서 충돌, 서비스 불안정 - 해결: 프로세스 우선순위 분리 및 격리, 패킷 처리 순서 최적화 - 성과: 시스템 충돌 0건, 서비스 안정성 99.9% 유지

3. **금융규제 요건 100% 충족** - 문제: 금융위원회 본인가 심사 통과를 위한 15종 보안 솔루션 통합 필요 - 해결: 각 솔루션 간 연동 구조 설계, 정책 일관성 검증 자동화 - 성과: 금융규제 요건 100% 충족, 본인가 심사 통과, 15종 보안 솔루션 통합 완료

Tech Stack Python, Ansible, Fortigate, NAC, DLP, APT, SSL VPN, 망분리

토스 커머스 연관성 → 복잡한 도메인을 단순하고 일관된 구조로 정리하고, 자동화를 통해 안정성과 확장성을 함께 고려한 경험.

(주)관텍투자일임

인프라·정보보호팀 인프라 엔지니어 | 2022.08 ~ 2024.03 (1년 7개월)

AI 기반 주식투자 서비스

담당 업무

- 금융보안데이터센터(FSDC) 운영
- **150대 이상 서버 및 스토리지 형상관리**
- 망분리 환경 내 DLP, DB 접근제어, VPN 정책 관리
- AWS 클라우드 보안 구성

주요 성과 (문제 정의 → 해결 → 성과) 1. **Python 기반 모니터링 자동화 구축** - 문제: 150대 서버 상태 수동 점검 시 하루 8시간 소요 - 해결: Python 스크립트로 서버 상태, 디스크, 네트워크 자동 점검 - 성과: 점검시간 8시간→2시간 (75% 단축), 장애 조기 발견률 향상

2. **DB 접근제어 쿼리 튜닝** - 문제: DB 접근제어 솔루션으로 인한 쿼리 성능 저하 (응답시간 3초+) - 해결: 불필요한 감사 정책 제거, 인덱스 최적화, 캐싱 레이어 추가 - 성과: 쿼리 응답시간 3초→0.5초 (83% 개선), 서비스 안정성 향상

3. **금융감독원 정기 감사 통과** - 문제: 금융감독원 정기 감사 대비 보안 체계 점검 필요 - 해결: 보안 점검 자동화, 감사 대응 문서 자동 생성 - 성과: 금융감독원 정기 감사 통과, 보안사고 0건 (12개월 연속)

Tech Stack AWS, Python, Terraform, MySQL, PostgreSQL, DLP, DB 접근제어

토스 커머스 연관성 → 데이터베이스 최적화 및 대규모 시스템 안정 운영 경험. 금융 서비스 환경에서의 정합성과 안정성 확보.

(주)메타엠

인프라 엔지니어 | 2019.12 ~ 2021.08 (1년 8개월)

대규모 재택근무 환경 구축

담당 업무

- 1,000명 규모 VPN 재택근무 환경 긴급 구축
- Fortigate SSL VPN + Ansible 자동 배포
- NAC 실시간 모니터링 및 위험 단말 격리

주요 성과 (문제 정의 → 해결 → 성과) 1. **Ansible로 VPN 1,000대 자동 배포** - 문제: COVID-19로 3일 내 1,000명 VPN 긴급 구축 필요, 수동 배포 시 2주+ 소요 - 해결: Ansible Playbook으로 FortiGate SSL VPN 설정 자동 배포 - 성과: 3일 만에 1,000대 배포 완료, 야근 없이 목표 달성

2. **VPN-백신 충돌 해결** - 문제: VPN과 백신 소프트웨어 충돌로 네트워크 연결 5~10분마다 끊김 - 해결: Wireshark 패킷 분석, 드라이버 우선순위 분리 및 프로세스 격리 - 성과: 연결 끊김 문제 완전 해결, 헬프데스크 장애 티켓 월 200건→30건 (85% 감소)

3. **Python 점검 스크립트로 점검시간 75% 단축** - 문제: 1,000대 VPN 상태 수동 점검 시 하루 8시간 소요 - 해결: Python 스크립트로 VPN 연결 상태, 인증서 만료일 자동 점검 - 성과: 점검시간 8시간→2시간 (75% 단축), 보안사고 0건 (20개월)

Tech Stack Fortigate SSL VPN, NAC, Ansible, Python, Cisco

토스 커머스 연관성 → 대규모 자동화 배포 경험과 예외 상황 처리 (충돌 문제 해결). 변화에 유연하게 대응하는 시스템 설계.

(주)엠티데이터

서버 엔지니어 | 2017.02 ~ 2018.10 (1년 8개월)

한국항공우주산업(KAI) 서버 운영

담당 업무

- Linux 서버 50대 운영관리
- 방화벽/IDS 로그분석 및 취약점 스캐닝
- Shell 스크립트 기반 방화벽 정책 버전관리

주요 성과 (문제 정의 → 해결 → 성과) 1. **방화벽 정책 롤백 자동화** - 문제: 방화벽 정책 변경 후 문제 발생 시 수동 롤백으로 30분+ 소요 - 해결: Shell Script로 정책 변경 전 자동 백업, 롤백 명령어 1줄로 복구 - 성과: 롤백 시간 30분→1분 (97% 단축), 서비스 중단 최소화

2. **방화벽 중복정책 정리** - 문제: 10년간 누적된 방화벽 정책 3,000건+, 중복/불필요 정책으로 성능 저하 - 해결: Shell Script로 중복 정책 자동 탐지, 정책 구조 단순화 - 성과: 정책 3,000건→1,200건 (60% 감소), 방화벽 응답시간 개선

Tech Stack Linux, Shell Script, iptables, IDS/IPS

토스 커머스 연관성 → 정책 구조를 단순화하고 자동화하여 운영 효율 개선. 안정성과 예측 가능성 유지.

Projects

REGTECH Blacklist Intelligence Platform

Production | 2024 ~ 현재 **URL** blacklist.jclee.me | **GitHub** github.com/qws941/blacklist

REGTECH 포털 자동 수집 및 위협 인텔리전스 플랫폼

Tech Stack Python Flask, PostgreSQL 15, Redis 7, Docker, GitHub Actions, Prometheus, Grafana

해결하려던 문제 REGTECH 포털에서 수집한 위협 정보(월 10만 건+)를 수동으로 검토하면 하루 8시간이 소요되고, 오탐으로 인한 알람 피로도가 심각했습니다. 또한 장애 발생 시 수동 복구로 평균 30분 이상 소요되어 서비스 안정성이 낮았습니다.

해결 방법

- Claude Code(LLM)를 활용한 로그 패턴 자동 분류 시스템 구축
- 정상 패턴 학습을 통한 이상 탐지 정확도 개선
- 자동 롤백 시스템으로 장애 복구 시간 단축
- Prometheus + Grafana 통합 모니터링으로 실시간 상태 확인

주요 성과

- AI 기반 장애 분석 시스템 (Claude Code, LLM 활용)
- AI로 로그 패턴 자동 분류 및 이상 탐지 (월 평균 오탐 알림 500건→25건, 95% 감소)
- 자동 롤백 시스템 (30초 내 복구)
- GitHub Actions CI/CD 파이프라인 (코드 푸시→테스트→배포 자동화)
- Prometheus + Grafana 모니터링 (24/7 실시간 상태 확인)

토스 커머스 연관성 → “지금 잘 작동하는 것”에서 멈추지 않고 “나중에도 잘 작동할 수 있게” 설계한 시스템. 자동 롤백과 장애 분석을 통해 안정성과 예측 가능성 확보.

Splunk-FortiNet Integration Platform

Production | 2024 ~ 현재 **URL** splunk.jclee.me | **GitHub** github.com/qws941/splunk-fortinet

Splunk-FortiManager 통합 플랫폼

Tech Stack Python Flask, Splunk API, FortiManager JSON-RPC, Cloudflare Workers, Redis 7, Docker

해결하려던 문제 80대 방화벽에서 실시간으로 수집되는 대량의 보안 로그를 동기 방식으로 처리하면 API 응답시간이 3~5초로 늘어나 서비스가 불안정해졌습니다. 대규모 트래픽 환경에서 빠른 응답성과 높은 정합성을 동시에 만족하는 시스템이 필요했습니다.

해결 방법

- Redis 기반 비동기 작업 큐 설계
- Celery Worker 병렬 배치로 각 방화벽 정책 독립 처리
- FortiManager JSON-RPC API 캐싱 레이어 추가로 응답 지연 최소화
- Cloudflare Workers로 엣지 캐싱 및 부하 분산

주요 성과

- 대용량 이벤트 처리 검증 (동시 요청 1,000건+ 안정 처리)
- 80대 방화벽 동시 관리 (실시간 정책 동기화)
- API 응답 시간 < 100ms (기존 3~5초→100ms, 97% 개선)
- 확장 가능한 아키텍처 설계 (수평 확장 가능)

토스 커머스 연관성 → 대규모 트래픽 환경에서 빠른 응답성과 높은 정합성을 동시에 만족하는 시스템 설계 경험.

FortiGate Policy Orchestration Platform

Production | 2024 ~ 현재 **URL** fortinet.jclee.me | **GitHub** github.com/qws941/fortinet

FortiManager 정책 관리 자동화 플랫폼

Tech Stack Python Flask, FortiManager JSON-RPC, Docker, GitHub Actions

해결하려던 문제 복잡한 방화벽 정책(3,000건+)을 수동으로 관리하면 휴먼에러로 서비스 장애 위험이 높았고, 감사 대응 시 문서 준비에 하루 8시간 이상 소요되었습니다.

해결 방법

- 정책 검증 자동화 (중복/충돌 정책 자동 탐지)
- 정책 배포 전 시뮬레이션 (영향도 분석)
- 감사 대응 문서 자동 생성 (정책 변경 이력 자동 추적)
- 3-Port HA 아키텍처 (무중단 서비스)

주요 성과

- 정책 검증 자동화 (휴먼에러 0건)
- 서비스 중단 0건 (12개월 연속)
- 감사 대응 문서 자동 생성 (작업시간 8시간→30분, 94% 단축)
- 3-Port HA 아키텍처 (무중단 서비스 지속)

토스 커머스 연관성 → 복잡한 정책 관리를 단순하고 일관된 구조로 정리하고, 자동화로 운영 효율 개선.

Full-Stack Observability Platform

Production | 2024 ~ 현재 **URL** grafana.jclee.me | **GitHub** github.com/qws941/grafana

Prometheus + Grafana + Loki + Tempo 통합 모니터링 플랫폼

Tech Stack Prometheus, Grafana, Loki, Tempo, Promtail, Traefik, Docker Compose

해결하려던 문제 5개 Production 시스템을 개별적으로 모니터링하면 장애 발생 시 원인 파악에 평균 1시간 이상 소요되었고, 시스템 간 연관성 분석이 어려웠습니다.

해결 방법

- **Prometheus + Grafana + Loki + Tempo 통합**으로 단일 인터페이스 구축
- 11개 대시보드 운영 (131개 패널, 시스템별 상태 실시간 가시화)
- 대용량 로그 이벤트 실시간 처리 (Loki 기반)
- 24/7 실시간 모니터링 (알림 자동화)

주요 성과

- 11개 대시보드 운영 (131개 패널)
- 대용량 로그 이벤트 실시간 처리 (일 평균 100만 건+)
- 24/7 실시간 모니터링 (장애 조기 발견률 향상)
- 5개 Production 시스템 메트릭 수집 (시스템 전반 가시성 확보)

토스 커머스 연관성 → 시스템 전반의 품질과 속도를 함께 책임지는 모니터링 체계. 실시간성과 안정성 확보.

SafeWork Industrial Health Platform

Production | 2024 ~ 현재 **URL** safework.jclee.me | **GitHub** github.com/qws941/safework

산업보건 관리 시스템 (13개 관리 패널)

Tech Stack Flask 3.0, PostgreSQL 15, Redis 7, Cloudflare Workers, Bootstrap

해결하려던 문제 전국 다수 중소기업에서 사용하는 산업보건 관리 시스템에서 집계 오류가 빈번하게 발생했고, 배포 시간이 길어 긴급 수정이 어려웠습니다.

해결 방법

- 데이터 정합성 확보 (트랜잭션 격리 수준 조정, 집계 로직 검증 자동화)
- GitHub Actions CI/CD 파이프라인 (배포 시간 단축)
- Cloudflare Workers 엣지 캐싱 (전국 동시 접속 처리)
- Redis 캐싱 레이어 (응답시간 개선)

주요 성과

- 전국 동시 접속 처리 (피크 시간대 500명+ 안정 처리)
- 집계 오류 100% 제거 (데이터 정합성 확보)
- 다수 중소기업 실운영 (실제 사용자 피드백 기반 개선)
- 배포 시간 67% 단축 (수동 배포 30분→자동 배포 10분)

토스 커머스 연관성 → 사용자와 셀러에게 직접 닿는 제품을 만든 경험. 데이터 정합성 확보 및 배포 자동화.

Education

한양사이버대학교 | 컴퓨터공학과 | 2024.03 ~ 재학중

토스 커머스 기술 스택 선제 학습 - Kotlin 기초 문법 완료 (Udemy 강의 유료) - Spring Boot REST API 프로젝트 진행 중 - 목표: 2025년 11월까지 간단한 커머스 API 구현

Certifications

- CCNP (Cisco Certified Network Professional) | Cisco Systems | 2020.08
 - RHCSA (Red Hat Certified System Administrator) | RED HAT | 2019.01
 - CompTIA Linux+ | CompTIA | 2019.02
 - LPIC Level 1 | LINUX | 2019.02
-

토스 커머스 서버 개발자 포지션 적합성

요구사항 매칭

빠른 응답성과 높은 정합성이 동시에 요구되는 시스템 설계·운영

- 대용량 이벤트 처리 플랫폼 구축 (API 응답 시간 < 100ms)
- 자동 롤백 시스템 (30초 내 복구) 구축
- 금융권 실시간 거래 환경에서 무중단 운영 달성

안정성과 예측 가능성을 유지할 수 있는 구조 고민

- 보안 솔루션 통합 운영 안정화 (15종 솔루션, 보안사고 0건)
- 자동 롤백 시스템 (30초 내 복구) 및 AI 기반 장애 분석
- 금융감독원 정기 감사 지적사항 0건 (12개월 연속)

복잡한 도메인을 단순하고 일관된 구조로 정리

- Python으로 방화벽 정책 100건+ 자동 배포 (작업시간 50% 단축)
- 정책 검증 자동화 (FortiGate Policy Orchestration)
- 복잡한 충돌 상황 구조화 및 자동 해결 (EPP/DLP 충돌 → 프로세스 격리)

시간이 지날수록 더 나아질 수 있는 구조 고민

- 확장 가능한 아키텍처 구축 (대용량 트래픽 처리 검증)
- 3-Port HA 아키텍처 (무중단 서비스 지속)
- 재사용 가능한 자동화 파이프라인 설계 (Ansible, GitHub Actions)

문제 정의 및 구조화 능력

- EPP/DLP 충돌 문제를 데이터 기반으로 분석하고 프로세스 격리로 해결
- 보안 정책 재설계로 안정성 확보 (문제 재정의 → 효율 개선)
- 신규 보안 솔루션 15종 통합 및 안정화 (금융 본인가 심사 통과)

토스 커머스 기술 스택 매칭

토스 요구사항	보유 기술	경험 수준
Kotlin, Spring	Kotlin/Spring 학습 중	학습 중 (Python/Node.js 기반 서버 개발 7년)
Python, Node.js, Go	Python, Node.js	7년 (실무 중심)
MySQL, MongoDB, Redis	PostgreSQL, MySQL, MongoDB, Redis	7년 (실무 중심)
ElasticSearch, Kafka, Hadoop	Splunk, ELK Stack	3년 (실무 중심)
GenAI	Claude Code, LLM 기반 자동화 시스템	1년 (실무 적용)

토스 커머스 팀별 적합성 분석

AI·자동화 영역 (최우선 희망) 반복적인 운영 업무를 기술로 치환 - Python 자동화로 작업시간 50~75% 단축 - Ansible로 VPN 1,000대 자동 배포 (3일 완료)

예외 상황을 안정적으로 처리 - EPP/DLP 충돌 해결 (프로세스 격리) - 보안 정책 재설계로 안정성 확보

운영 효율을 높이는 시스템 설계 - 정책 검증 자동화 구축 - 감사 대응 문서 자동 생성

주문·결제·프로모션 영역 정합성 있는 시스템 설계 - 금융권 실시간 거래 환경에서 무중단 운영 - 데이터 정합성 확보 (집계 로직 오류 수정)

다양한 할인 정책, 정산 가능한 가격 구조 - 복잡한 정책 관리 자동화 (FortiGate Policy Orchestration) - DB 접근제어 쿼리 튜닝 (성능 개선)

상점·상품 영역 상품 수명주기 정의 및 안정적 운영 - 신규 솔루션 15종 통합 및 안정화 - 형상관리 및 버전관리 자동화 (150대 서버)

함께 만들고 싶은 것

토스 커머스는 “이미 정의된 문제를 푸는 것”보다 “문제를 정의하고, 나아갈 방향을 함께 만드는 것”을 중요하게 생각합니다.

저는 7년간 보안사고 0건을 유지하면서도 작업시간을 50~75% 단축한 경험이 있습니다. 이는 단순히 주어진 문제를 해결한 것이 아니라, “어떻게 하면 보안과 효율을 동시에 확보할 수 있을까?”라는 질문에서 출발한 결과입니다.

토스 커머스의 초기 멤버로서, 커머스의 여러 기능이 유기적으로 연결된 구조에서 안정성과 확장성을 함께 고려한 시스템을 설계하고 싶습니다. 특히 AI·자동화 영역에서 반복적인 운영 업무를 기술로 치환하고, 예외 상황을 안정적으로 처리하는 구조를 만드는 데 기여하고 싶습니다.

마무리

“지금 잘 작동하는 시스템보다, 시간이 지날수록 더 나아질 수 있는 구조를 고민하는 분”

이 한 문장이 저의 지난 7년을 가장 잘 설명합니다. 토스 커머스에서 함께 만들어갈 기회를 기대합니다.