

# 류지환

## 연락처

- 전화: 010-4008-0967
- 이메일: ryoo0504@gmail.com
- GitHub: <https://github.com/mangowhoiscloud>
- Blog: <https://rooftopsnow.tistory.com/>
- LinkedIn: <https://www.linkedin.com/in/jihwan-ryu-b6b04a202/>

## 지원 포지션

- 두나무 DevOps

## Summary & 지원 동기

- Distributed System & Low-level Optimization:** 글로벌 통신망을 위한 클라우드 스토리지(Robin Storage)를 개발하며 K8s / C 기반의 비동기 RPC 통신과 I/O 동시성 제어(Locking) 문제를 해결했습니다. 이러한 분산 시스템 운영 및 최적화 경험은 블록체인 노드의 안정적인 운영과 성능 개선에 기여할 수 있다고 확신합니다.
- Infrastructure as Code & Automation:** Terraform과 Ansible을 활용해 14-Node 규모의 Kubernetes 클러스터를 밑바닥부터 구축하고, GitOps(ArgoCD) 파이프라인을 통해 운영 자동화를 달성했습니다. AWS 및 On-premise 환경에서의 노드 오케스트레이션 역량을 바탕으로 뱌리데이터 노드를 안정적으로 운영하겠습니다.
- Passion for Blockchain:** 과거 Ethereum Geth를 활용해 Private Network를 구축하고 ERC-1155 기반의 DApp을 개발하며 블록체인 생태계에 입문했습니다. 현재는 Rust/Go 언어 학습을 통해 블록체인 코어 레벨의 기여자가 되고자 하는 목표를 가지고 있습니다.
- AI-Driven Development:** Claude/GPT 모델에 약 30.6억 토큰을 투자하며 인프라 구축과 트러블슈팅 효율을 극대화하는 개발 방식을 실천하고 있습니다. 단순 코딩을 넘어, AI를 Pair Programmer로 활용하여 생산성을 증진시키는 방법에 흥미를 가집니다. **기술적 해자(Moat)**의 필요성에 대해 고민하며, 코어 역량(아키텍처, 철린지한 문제 해결)에 집중해 가치를 찾는 데 주력하고 있습니다.

## 핵심 역량

- Blockchain:** Ethereum (Geth Private Network 구축), Solidity (ERC-1155 Smart Contract), Web3.js, IPFS
- Infrastructure:** Kubernetes, AWS (EC2, VPC, IAM), Terraform, Ansible, Docker
- Backend:** Python (FastAPI), Java (Spring Boot), C++, RPC/REST API, Concurrency Control
- DevOps & Security:** ArgoCD (GitOps), GitHub Actions, NGINX (Reverse Proxy/SSL), Linux (Kernel Tuning, io\_uring)

## Professional Experience

### Rakuten Symphony Korea - Cloud Engineer / Jr. Storage Dev

2024.12.09 - 2025.08.31 글로벌 리전(독일, 일본)에 배포된 통신용 클라우드 플랫폼의 스토리지 서버 모듈을 개발하고 인프라를 최적화했습니다.

1. **Robin Storage (Block Storage) 분산 처리 및 성능 최적화** 블록체인 노드의 I/O 처리 및 상태 동기화와 유사한 분산 스토리지의 정합성/성능 문제를 해결했습니다. 1. **Control Plane ↔ Data Plane 상태 동기화 RPC 개발:** - **상황:** 스냅샷 로딩 시 Control Plane과 Data Plane(RDVM) 간의 메타데이터 불일치로 인해 볼륨 마운트 실패 발생. - **액션:** 비동기 RPC 프로시저(RDVM\_VOL\_CFG\_UPDATE)를 신규 개발하여 노드 간 상태 업데이트를 강제 동기화하는 로직 구현. - **결과:** 데이터 무결성을 확보하고 스냅샷 추적 목표 달성.

#### 2. I/O Concurrency 성능 병목 개선:

- 상황:** 스냅샷 반복 읽기(Read-heavy) 작업 시 down\_write(Write Lock) 점유로 인한 병목 현상 확인.
- 액션:** 커널 레벨의 락 메커니즘을 분석하여, 데이터 변경이 없는 구간을 down\_read(Read Lock)로 전환.
- 결과:** 동시 읽기 처리량(Throughput) 대폭 개선. 향후 io\_uring 도입을 통한 비동기 I/O 전환 로드맵 캐치업.

#### 2. Rakuten OStore (Object Storage) 보안 및 인증 체계 개선

- Authentication 보안 취약점 개선:** 사용자 비활성화 시에도 Access Key가 잔존하는 보안 훔을 발견하고, Gateway-DB 간 인증 흐름을 재설계하여 키 생명주기를 사용자 상태와 동기화.
- Root User 권한 제어:** 설치 시점에만 Root 사용자가 생성되도록 부트스트래핑 로직을 수정하여 운영 중 불필요한 슈퍼유저 생성을 차단, 보안성 강화.

## Projects

### 1. Ethereum Multi-Token 기반 경매/운송 플랫폼

2022.03 - 2022.11 | Backend & Smart Contract Developer GitHub: mng990/ethereum\_FisheriesMarket

**Role:** 스마트 컨트랙트 개발 및 Private Blockchain 네트워크 구축

- **Smart Contract (Solidity):** 단일 컨트랙트로 다중 토큰을 관리할 수 있는 **ERC-1155** 표준을 채택하여 상품(NFT), 경매 권한, 화폐(FT)를 효율적으로 관리하는 스마트 컨트랙트 설계 및 배포.
- **Private Network 구축 (Geth):** Ethereum Go Client(Geth)를 활용해 Genesis Block을 생성하고 Private Network를 구축하여 트랜잭션 수수료(Gas) 비용 없이 테스트 가능한 환경 구성.
- **Decentralized Storage (IPFS):** 중앙화된 서버의 데이터 위변조 위험을 제거하기 위해, 경매 물품의 미디어 데이터를 IPFS(InterPlanetary File System)에 저장하고 해당 해시값을 온체인에 기록하여 무결성 보장.

### 2. Eco<sup>2</sup> (이코에코) - K8s 기반 인프라 구축 및 운영

2025.10 - Present | Backend & DevOps (1인 개발) GitHub: eco2-team/backend

**Role:** 14-Node Kubernetes 클러스터 구축 및 GitOps 파이프라인 운영

#### 1. Infrastructure as Code (Terraform/Ansible):

- AWS VPC, EC2, ALB 등 60여 개의 리소스를 Terraform으로 정의하여 인프라 프로비저닝/부트스트래핑 시간 단축 (수동 4시간 → 자동 15분).
- Ansible Playbook을 작성하여 14개의 노드에 대한 OS 설정(Kernel Parameter), Container Runtime, Kubeadm 조인 과정을 완전 자동화.

#### 2. GitOps Strategy (ArgoCD & Helm/Kustomize):

- Helm Charts로 패키징된 애플리케이션과 Kustomize를 결합(Base/Overlays)하여 환경별 설정을 유연하게 관리.
- ArgoCD Sync Wave를 적용하여 **Infra(DB/MQ) → App** 순서의 배포 의존성을 제어하고, 안정적인 롤아웃 전략 수립.
- Operator + CRD/CR을 활용한 선언적 관리 모델을 도입하여, 단순 배포를 넘어 애플리케이션의 라이프사이클(백업, 복구 등)까지 자동화하는 방향으로 인프라를 점진적으로 개선 중.

#### 3. Security & Network Policy:

- Zero Trust 원칙을 적용하여 네임스페이스 간 통신을 기본 차단(Deny-All)하고, 필요한 트래픽만 Network Policy로 허용하여 노드 간 보안 강화.
- Taints/Tolerations 및 Node Affinity를 활용해 중요 API 파드와 인프라 파드를 물리적으로 격리, 밸리데이터 노드 운영 시 필요한 리소스 격리 전략 습득.
- SSM 도입, Secret 주입 자동화

## Education

- 부산대학교 컴퓨터공학부 학사 (2017.03 - 2023.08)

## Certifications & Language

- 정보처리기사 (Engineer Information Processing)
- OPIc IH (Intermediate High) - 글로벌 팀과의 기술적 소통 가능