

# 백엔드 개발자 (Go/Microservices) - 경력 6년차

---

## 기본 정보

- 이름: 최도윤
  - 연락처: 010-9999-1111
  - 이메일: doyoon.choi@example.com
  - GitHub: [github.com/doyoonchoi](https://github.com/doyoonchoi)
- 

## 기술 스택

- 프로그래밍 언어: Go, Python, Java
  - 백엔드 프레임워크: Gin, Echo, Fiber, Spring Boot
  - 아키텍처: MSA(Microservices Architecture), RESTful API, gRPC, Protobuf
  - 데이터베이스: PostgreSQL, MySQL, Redis
  - 메시지 브로커: Kafka, RabbitMQ
  - 클라우드 및 DevOps: AWS (EC2, S3, Lambda, RDS), Kubernetes, Docker
  - 모니터링 및 로깅: Prometheus, Grafana, ELK Stack
- 

## 경력

### ◆ 백엔드 엔지니어 | 물류 플랫폼 기업 | 2018.03 ~ 현재

- Go 언어 기반의 고성능 API 서버 개발 및 운영
  - MSA 환경에서 수십 개의 마이크로서비스를 운영하며, gRPC 및 Protobuf을 활용한 서비스 간 통신 최적화
  - Kafka 기반 이벤트 스트리밍 도입으로 비동기 데이터 처리를 최적화
  - Prometheus, Grafana 기반의 실시간 모니터링 구축을 통해 서비스 장애 감지 및 대응 속도 개선
  - Kubernetes 기반의 컨테이너 오케스트레이션 최적화로 배포 및 스케일링 자동화
- 

## 프로젝트 경험

- ✓ 물류 추적 시스템 리뉴얼

- 기존 모놀리식 구조의 물류 추적 시스템을 **MSA로 전환**하여 확장성과 유지보수성을 개선
- **Kafka를 활용한 이벤트 스트리밍 아키텍처** 도입으로 실시간 데이터 처리를 최적화
- 장애 발생률 **30% 감소**, 서비스 가용성 **99.9% 유지**, SLA(서비스 수준 계약) 만족도 **95% 이상** 달성
- **Redis 기반 캐싱 시스템** 도입으로 API 응답 속도를 평균 **40% 개선**

#### ✅ 배송 최적화 알고리즘 개선

- 머신러닝 모델을 적용한 배송 경로 최적화 기능 개발
- 과거 데이터 분석을 통해 **배송 비용 15% 절감**, 평균 배송 시간 **20% 단축**
- Kubernetes 및 AWS Lambda를 활용한 **서버리스 아키텍처** 도입으로 비용 절감 및 운영 효율성 향상

#### ✅ CI/CD 파이프라인 구축

- **GitHub Actions** 및 **ArgoCD** 기반의 배포 자동화 구축
- Blue-Green 배포 전략을 도입하여 **서비스 다운타임 없이** 신규 기능 배포 가능



## 자기소개서

**"시스템 아키텍처는 곧 비즈니스 전략이다."**

6년간 물류 시스템을 개발하며 얻은 가장 큰 교훈입니다.

개발자로서 단순히 기능을 구현하는 것이 아니라, 서비스의 안정성과 운영 효율성이 곧 비즈니스 성과로 직결된다는 점을 몸소 경험했습니다.

저는 항상 **"어떻게 더 나은 구조로 만들 수 있을까?"** 라는 고민을 하며, 기능 개발을 넘어서 **시스템 아키텍처 전반에 기여하는 것**을 목표로 하고 있습니다.

## 1. 문제 해결을 위한 깊이 있는 접근

과거 모놀리식 시스템에서 운영되던 물류 추적 서비스를 MSA 구조로 전환하는 과정에서, **기존의 데이터 흐름을 분석하고 최적화하는 것이 핵심 과제**였습니다.

초기에는 서비스 간의 통신 지연과 데이터 일관성 문제로 인해 장애율이 높았지만,

**Kafka 기반의 이벤트 스트리밍을 도입하여 비동기 메시지 처리 방식으로** 개선하였습니다.

그 결과, 장애 발생률을 **30% 줄이고** 서비스 가용성을 **99.9%까지 유지**할 수 있었습니다.

이처럼 저는 **기술적 문제를 단순 해결하는 것이 아니라, 장기적으로 유지보수성과 확장성을 고려한 최적의 솔루션을 설계**하는 개발자입니다.

---

## 2. 클라우드 및 DevOps 환경에서의 성장

서비스를 운영하면서 클라우드 네이티브 환경에서 자동화와 운영 효율성이 얼마나 중요한지 깨달았습니다.

특히 Kubernetes와 AWS 기반의 인프라 환경에서 자동화된 배포(CI/CD) 및 스케일링 시스템을 구축하며,

서비스 확장과 운영 비용 최적화의 필요성을 더욱 체감할 수 있었습니다.

이를 위해 Prometheus, Grafana 기반의 모니터링 시스템을 구축하여 장애 감지 및 대응 속도를 대폭 개선하였으며,

GitHub Actions 및 ArgoCD를 활용하여 Blue-Green 배포 자동화를 구현함으로써 다운타임 없는 안정적인 배포 프로세스를 만들었습니다.

이제는 개발뿐만 아니라 DevOps 및 클라우드 환경에서의 운영 최적화 경험을 더욱 심화하고 싶습니다.

---

## 3. 입사 후 목표

### 1) 확장성 높은 아키텍처 설계

- 서비스 트래픽 증가에도 안정적인 성능을 유지할 수 있는 확장 가능한 시스템을 구축하고 싶습니다.
- 특히 gRPC 기반의 마이크로서비스 최적화 및 메시지 브로커(Kafka)를 활용한 고성능 데이터 처리 시스템 구축을 목표로 합니다.

### 2) 고성능 API 및 데이터 처리 최적화

- 현재 보유한 Go 기반 백엔드 최적화 경험을 더욱 발전시켜, 대규모 트래픽을 효과적으로 처리할 수 있는 아키텍처를 설계하고 싶습니다.
- Redis 및 데이터 캐싱 기법을 활용한 API 성능 개선을 통해, 빠른 응답 속도와 안정적인 서비스 제공을 목표로 합니다.

### 3) DevOps 및 클라우드 인프라 경험 확장

- 현재 보유한 Kubernetes 및 AWS 운영 경험을 바탕으로, 인프라 최적화 및 자동화 배포 환경을 더욱 고도화하고 싶습니다.
  - IaC(Infrastructure as Code) 도입 및 클라우드 네이티브 환경에서의 최적화 기술을 익혀, 운영 효율성을 높이는 데 기여하고 싶습니다.
- 

마무리하며...

백엔드 개발자로서, 저는 단순한 기능 구현을 넘어서 **서비스의 안정성과 확장성을 고민하는 개발자**입니다.

6년간 쌓아온 **MSA 아키텍처 설계 및 대규모 시스템 운영 경험**을 바탕으로,  
비즈니스 성장과 기술 혁신에 기여할 수 있는 개발자로 계속해서 성장하고 싶습니다.

저는 "**문제를 해결하는 개발자**"에서 나아가, **시스템을 설계하고 최적화하는 개발자**가 되고자 합니다.

입사 후에도 계속해서 새로운 기술을 탐색하고, 서비스의 가치를 높이는 데 기여하겠습니다.



---

## 최종 요약

- **MSA 및 분산 시스템 경험 풍부** (Kafka, gRPC, Kubernetes)
- **고성능 API 및 데이터 최적화 경험** (Redis, PostgreSQL, Go)
- **클라우드 및 DevOps 환경 최적화 가능** (AWS, CI/CD, ArgoCD)
- **입사 후 확장성 높은 아키텍처 설계 및 최적화 목표**