

# 챌린지를 즐기는 개발자, 홍석민입니다.



# **About Me**

### Introduction

- 1년차 백엔드 + Devops 개발자 홍석민입니다.
- Challenging한 태스크를 해결하는 것에 푹 빠져있습니다.
- 끊임없는 성장을 위하여 1일 1커밋을 지속하고, Tech Blog를 운영하고 있습니다.
- 더 나은 코드를 위한 고민을 멈추지 않습니다. (Blog)
  - Clean Code
  - TDD
  - Clean Architecture
  - Refactoring
- 음악과 봉사, 성장하기에 관심이 많습니다.



# Contact & Channel

- Email:
  - scarletgold0601@gmail.com
  - hongseoljin@korea.ac.kr
- Github : https://github.com/honghyeong
- Blog: <a href="https://5hongmin.tistory.com">https://5hongmin.tistory.com</a>



#### **Programming Language**

- C++
- JavaScript
- TypeScript
- Java
- Go
- Python

### Backend

- NodeJs, NestJS, ExpressJS
- Spring
- TypeORM
- gRPC

#### Devops

- Docker
- AWS EC2, ECS, RDS, Route 53, CloudWatch, SNS, Lambda, MSK, EventBridge, S3, ECR
- Postgres DB, MySQL
- Git, Git Actions (CI / CD)
- Pulumi
- Nx
- Locust

#### Frontend

- HTML5, CSS3(SCSS), JS(ES6)
- React
- NextJS

#### Collaboration

- Slack
- Git

# **Work Experience**

### 코르카 AdTech bidding price 추론 시스템

하루에 최대 18500/sec, 251M/day의 요청을 처리해야하는 시스템 Docker 컨테이너당 rps 6~700 를 처리해야하는 시스템

#### Dev

- 트래픽(RPS)가 높아짐에 따라 생긴 다양한 문제를 로컬에서 재현하고, CS에 기반하여 해결
  - Golang 으로 작성한 bidding price 추론부에서 Map에 동시에 여러 write 요청 발생하여 race condition 발생 ⇒ SyncMap, Mutex 패키지를 이용하여 해결
  - HTTP POST request가 쌓여 HTTP Connection(Hand-shaking) 이 증가함에 따라 too many open files 문제 발생 ⇒ HTTP Connection 재활용하도록 Http
    Request 수정
  - 추론부 CPU utilization이 증가함에 따라 scale out 이 과도하게 진행되는 문제 발생
    Go tracing tool: jaegar 를 이용하여 각 메소드별 처리시간을 트레이싱하고
    HTTP Request 의 처리 시간이 긴 것(80ms)을 파악 ⇒ TCP 에서 UDP으로 통신 프로토콜 바꾸는 것을 고려
  - Socket Connection 이 끊어졌다가 다시 연결될 때, 서버쪽 DB에 등록되지않은 이름을 부여받아 실패하는 상황 발생 ⇒ Connection close될 때, 추론부 id 삭제
- 외부 컴포넌트와 grpc 통신하는 과정에서 Network Latency 문제를 해결하기위해 기존의 Docker을 통해 추론하던 추론부를 executable file로 변경하여 Docker Contaienr network로 인한 grpc latency를 획기적으로 감소하는 아이디어 제안
  - Docker Container ( 12ms ) ⇒ executable ( 3.3 ms ): grpc + file io / stdio pipe
  - 。 이후 dll 을 이용한 커널 레벨 공유로 전개됨
- TDD를 통한 명세를 기반으로 개발, 협업
  - o git convention, gitflow

### **DevOps**

- 과도한 트래픽에 의해 AWS 환경에서 띄워진 아키텍쳐의 어디서도 문제가 생길 수 있기 때문에 Slack 웹혹을 이용한 모니터링 시스템 구축
  - 。 ECS Task Exited Alerting ⇒ AWS EventBridge + Lambda 를 이용한 슬랙 알람
  - ECS Resource Alerting ⇒ AWS CloudWatch + AWS SNS + AWS Lambda를
    이용한 슬랙 알람
  - 。 서버 내부 Slack 알림 서비스 개발
  - Pulumi 를 통한 AWS Resource 알람 설정 자동화
- 트래픽 증가로 인한 RDS Queue depth 문제 ⇒ Scale up을 통한 문제 해결
- Pulumi 를 이용한 Infrastructure managing
- 초당 2만개의 트래픽을 감당할 수 있는 아키텍처 설계
  - Distributed Load Testing 을 통하여 초당 2만개의 요청을 보내 아키텍처의 안정성 검증

# **Personal Experience & Projects**

# **Club Experience**

### **Devkor**

- 2021.07 ~
- devops, backend, frontend 세미나 진행
- devkor 공식 사이트

#### **Kweb**

- kweb 정회원
- backend + frontend 세미나
- 고려대학교 수강 리뷰 사이트 KLUE 개발 참여

### **ALPS**

### **KUBIG**

- CARLA Simulation 를 이용한 RL
- 수강신청 리뷰 요약

### **Lab Experience**

### **Distributed / Cloud Computing Lab**

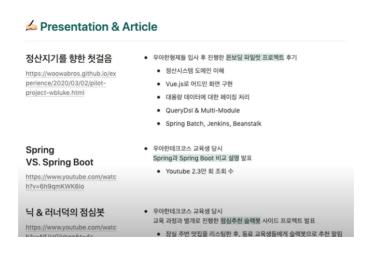
• 2021.12 ~ 2022.03

## **Project: Nangboo**

• 냉장고를 부탁해 어플 출시

•

# **Presentation & Article**



### **Education**

### 고려대학교 컴퓨터학과

• 2017.03 ~ 2023.02

• GPA: 3.7

CV

6