张刊

- Tel: 16602678467
- Wx: `kanlac`
- Email: ridethesnake@outlook.com

一句话介绍

5 年服务端开发经验, 深入理解面向对象、函数式等编程范式, 工作风格严谨、重视开发规范, 平时写写技术博客 https://kanlac.hashnode.dev, 具备英语自然阅读和交流能力。最近在深入探索 AI coding agent 的生产化实践。

技术栈

- **编程语言**: Go(5 年)、Python、C++
- **后端框架**: Gin、Gorm、Flask
- **数据库**: PostgreSQL、MongoDB、ZooKeeper、对象存储
- **开发实践**: TDD(testify/testcontainers)、DDD 分层架构、MVC
- **容器编排**: Kubernetes(CKA 认证)、Docker、Helm
- **团队协作**: 技术文档编写、Code Review、性能调优

工作经历

杭州心智拟合 | 技术负责人 | 2025.05 - 至今

- 从 0 到 1 搭建对象存储服务与交易系统, 使用 Go 实现订单流转、文件上传下载等核心功能, 通过分片上传、存储去重和冷热分离策略, 降低存储成本 40%, 上传速度提升 60%
- 带领 3 人团队深度实践 Claude Code 工程化, 在开发提效 3-5 倍的基础上保障可控的代码质量，实践复利工程
- 负责技术选型与架构设计 (PostgreSQL + Redis), 搭建监控告警体系与 Docker Compose 自动化部署流程, 实现代码提交到上线的全流程自动化
- 制定 Go 开发规范和 Code Review 流程, 推动 TDD 实践, 独立处理服务器故障排查与性能调优, 确保代码质量与系统稳定性

北京智维盈讯 | 后端开发工程师 | 2023.02 - 2024.10

- 使用 Go 开发配置中心、事件中心等平台功能, 为 100+ 微服务提供配置管理
- 主导 DevOps 平台建设, 实现 80% 业务自动化部署覆盖率
- 负责 SDK 技术文档的撰写与维护, 降低团队沟通成本
- 为开发团队提供容器运行时、网络连通性、构建部署等技术支持

北京启明星辰 | 后端开发工程师 | 2021.05 - 2022.11

- 使用 Go 开发 Kubernetes(K8s) 部署后端, 将人为配置错误率降低 90%
- 主导开发软件升级包方案, 使用 ECC 算法生成数字签名防止篡改
- 重构 GitLab CI/Jenkins 流水线, 通过多阶段 Docker 构建缓存、并行测试, 将全流程耗时从 40 分钟缩短至 15 分钟

天津卓筑汇 | 全栈开发工程师 | 2020.03 - 2020.12

- 深度参与建筑设计软件从 0 到 1 的研发与上线过程
- 同建筑专家完成领域建模, 独立负责 C++ 图形应用开发, 将地库出图效率提升 30%
- 使用 Python Flask 搭建用户和项目管理后端

核心项目经历

配置中心

背景: 微服务配置项散落在各服务本地文件, 紧急变更需登录 shell 操作繁琐易出错, 平均耗时 15 分钟/次; 默认配置与生产环境配置耦合, 多次因意外覆盖配置项导致事故。

任务: 设计统一配置管理方案, 支持文本、数字、枚举等类型, 允许 Web 界面、后端服务和脚本多端调用, 支持关联配置项与部署参数, 减少手动运维操作。

行动: 1. 开发 RESTful/gRPC 接口和 Go 语言 SDK, 通过 ZooKeeper 提供配置变更监听,PostgreSQL 持久化 2. 实现配置与资源联动, 自动触发服务滚动更新 3. 编写函数式 validator, 支持正则表达式、数值区间等验证模式, 拦截 40% 以上不规范配置提交

成果: 为 100+ 个微服务提供配置接口, 节省了 80% 以上配置相关的沟通协调工时, 紧急变更耗时缩短至 2 分钟内, 配置错误导致的生产事故归零; 通过自动联动机制, 容器编排相关工单减少 40%。

分布式迁移

背景: 随着业务流量的增长, 单体应用面临性能瓶颈和资源利用率低的问题。为充分利用多节点服务器资源, 我们决定将现有应用迁移至 Kubernetes 集群 (k3s)。

任务: 1. 推进现有应用适配分布式环境, 包括容器化改造、配置管理规范、服务迁移脚本以及服务发现机制 2. 改造现有的制包、部署工具和 CI 流程, 使其兼容和支持 Kubernetes 版本的应用部署 3. 设计兼容开发和生产环境的镜像管理方案

行动: 1. 编写 Helm Library Chart, 简化服务配置的编写成本, 降低开发者的认知负载, 通过 Git Submodule 集成到服务的代码仓库 2. 使用 kustomize 完成基础组件的部署, 包括 Ingress-Nginx、镜像 Registry 等 3. 使制包工具、部署工具和 CI 兼容新的部署模式, 支持 K8s 版本的应用部署

成果: 成功将核心交易链路上的 5 个关键应用迁移至 Kubernetes (k3s) 集群;Helm Library Chart 的引入使新服务的 Kubernetes 配置编写效率提升了 40%。

Docker Compose 自动化部署

背景: 公司产品需要一个简单易用的私有化部署方案, 支持在裸金属环境实现一键初始化, 满足客户管理规范, 支持主机配置 (DNS) 的动态修改。

行动: 开发 systemd 守护进程和二进制工具, 实现安装包制作、服务部署和启停等功能; 制定权限管理规范, 推动 20+ 微服务重构, 消除脚本中的 sudo 操作; 提出覆盖不同类型的容器网络的 DNS 同步方案。

成果: 支持了 6 个 Linux 发行版的产品部署, 安装成功率从 65% 提升至 95%; 修复 40+ 部署工具漏洞; 全线产品支持 non-root 降权部署。

教育背景

TJAU 天津农学院 | 全日制本科（学信网可查） | 软件工程专业 | 2016-2020

开源贡献

- **ko** (8.1k star, Go 应用部署): 支持部署更小的 go 镜像, 编写 e2e 测试
- **podman** (25k star, daemonless 容器运行时): 遵循 Docker Engine 文档完成 API 开发