

# Данила Манаков

email: ManakowDanila@yandex.ru | telegram: @CardinalSD

---

## Общее

---

- Защитил проект на стипендию от Selectel по backend разработке
- Решаю задачи на Codeforces, Acmp, Leetcode
- Разрабатываю собственные проекты: Система мониторинга VPN-сервиса, система создания бэкапов конфигурационной базы данных для VPN-сервиса.
- 2 года опыта разработки

## Навыки

---

- Языки программирования: Go, C/C++, Python, Kotlin
- Базы данных: mariadb, postgresql(mysql)
- Общие навыки: REST API, Git, Postman, Docker, testify, Kafka, Linux, Bash, S3, Grafana
- Языки: Английский B2, Немецкий C2

## Проекты

---

- Разработка и интеграция системы автоматического резервного копирования

(Проект доступен на [GitHub](#))

Задача: Разработать и интегрировать в open-source проект систему автоматического резервного копирования конфигурационной базы.

Как реализовывал: Ознакомился с архитектурой проекта и форматом хранения

конфигурационных данных. Реализовал модуль, который по расписанию создает резервные копии конфигурационной базы и сохраняет их в отдельном хранилище. Добавил проверку целостности бэкапов и логирование. Для гибкости настроек использовал YAML-конфигурации и системные переменные. Интеграция выполнена с учетом существующей структуры проекта и CI/CD-процессов.

**Результаты:** Система автоматически выполняет резервное копирование конфигурационной базы по заданному расписанию, обеспечивая сохранность данных и возможность быстрого восстановления при сбоях. Повысилась надежность работы сервиса и упростилось администрирование.

- Приложение по сокращению ссылок (Проект доступен на [GitHub](#))

**Задача:** Реализовать сервис по сокращению ссылок.

**Как реализовывал:** Создал схему базы данных для хранения ссылок и пользователей. Для ускорения работы реализовал кеш и прогрев при старте. Актуальность кеша контролирую структурой данных LRU. Также реализовал

авторизацию и регистрацию.

**Результаты:** Был создан сервис по сокращению ссылок с реализацией

кеширования ссылок для ускорения повторного сокращения ссылок, что позволило сократить скорость ответа пользователю с 1800ms до 5-15ms (x180).

- Исследовательская работа по теме “Преимущества параллелизма в Golang в сравнении с C++ и Java” (Проект доступен на [GitHub](#))

**Задача:** Исследовать и сравнить механизмы параллелизма в Golang, C++, и Java, оценив их производительность и эффективность в различных сценариях. **Результаты:** Golang обеспечивает высокую производительность и удобство разработки благодаря легковесным горутинам и кросс-коммуникациям между ними.

- Веб-сервис файлообмена Go, MariaDB, S3 (Проект доступен на [GitHub](#))

**Задача:** Реализовать сервис обмена файлами

**Как реализовывал:** Выстроил архитектуру проекта, разбил сервис на микросервисы - web-api, watchdog, cli(wip). Разработал Rest API сервис с веб-интерфейсом и Watchdog сервис для управления жизненным циклом файла.

**Результаты:** удобным способом web/cli загружается файл, а пользователю отдается прямая ссылка на этот файл.