

ФГБОУ «Рязанский государственный радиотехнический университет
имени В.Ф. Уткина»

РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ ТРЕНИРОВКИ В РЕШЕНИИ АЛОГРИТМИЧЕСКИХ ЗАДАЧ И АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ КАНДИДАТОВ

Подготовил:
Вашкулатов Н.А. 045

Руководитель:
Ефимов А.И.

Рязань 2024

Актуальность

Сейчас почти на всех собеседованиях в крупные компании для начинающих разработчиков обязательно присутствует решение каких-либо алгоритмических задач.

Постоянная практика в решении алгоритмических задач не только развивает умение быстро и эффективно находить оптимальные решения, но также формирует аналитическое мышление и готовность к промышленной разработке.

Так же компании могут быть заинтересованы в отборе лучших кандидатов и такая платформа может помочь в отборе.



Постановка задачи

- Выполнение кода пользователя на сервере;
- Автоматическая проверка решения пользователя при помощи набора тестовых данных и запуска кода;
- Сортировка задач по сложности и темам;
- Создание частных и публичных задач, в решении которых пользователь может тренироваться;
- Создание частных и публичных соревнований – контестов;
- Для решения задачи пользователю нужно лишь реализовать метод, без реализации ввода/вывода в консоль;
- Не является платформой для проведения олимпиадных соревнований.

Существующие решения

| Платформа | Русский язык | Для решения только реализовать метод | Удобный интерфейс | Быстрая обратная связь | Быстрое создание соревнований | Быстрое создание задач |
|-------------------|--------------|--|----------------------|---------------------------|-------------------------------------|---------------------------|
| LeetCode | Нет | Да | Да | Да | Нет | Нет |
| CodeWars | Нет | Да | Нет | Да | Да | Нет |
| Codeforces | Да | Нет | Нет | Нет | Нет | Нет |
| Яндекс Контест | Да | Нет | Да | Нет | Частично | Частично |
| Желаемое | Да | Да | Да | Да | Да | Да |

Используемые технологии

- Kotlin
- Kora
- Kafka
- PostgreSQL
- Docker
- Kubernetes
- JavaScript
- Vue
- Nginx

Компиляция и выполнение кода на сервере

Для автоматизации этих процессов используется отдельное приложение и Docker

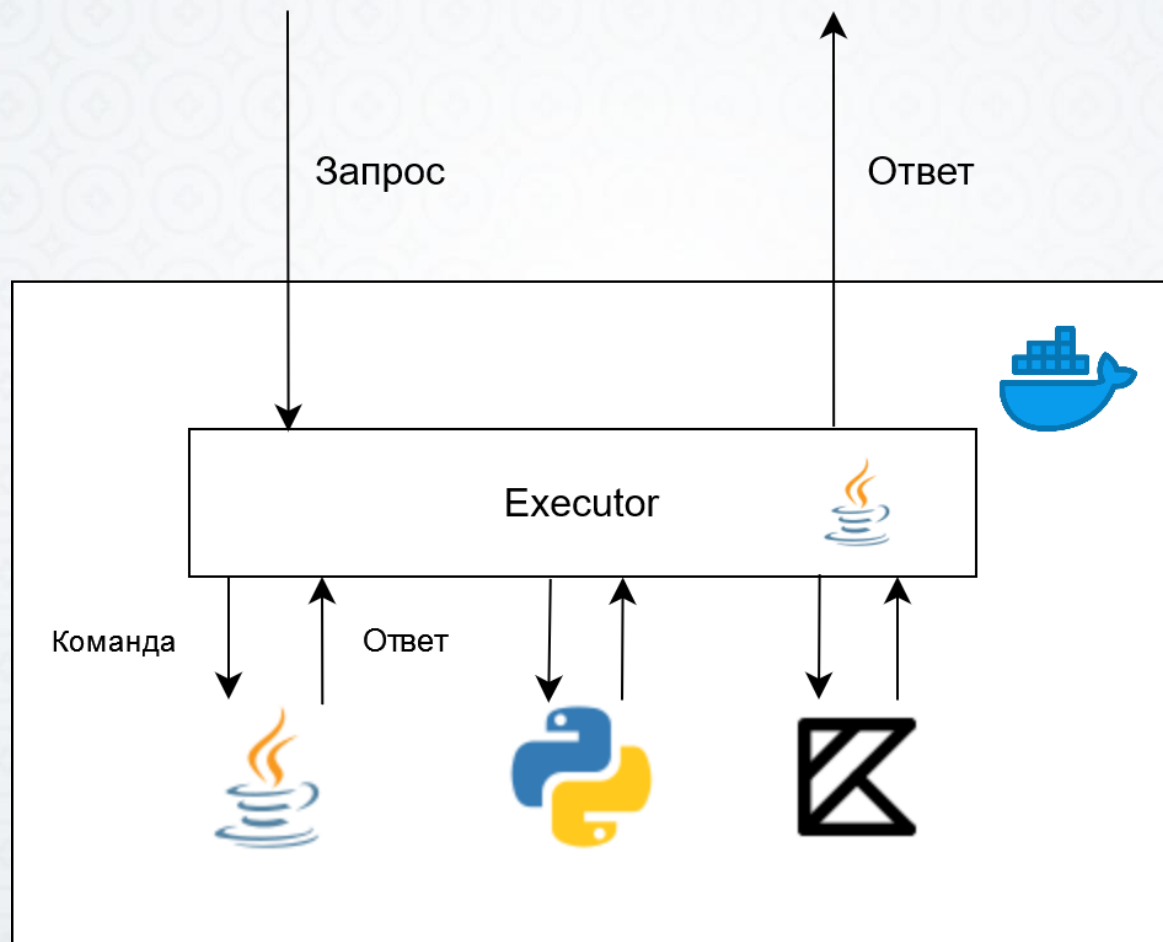
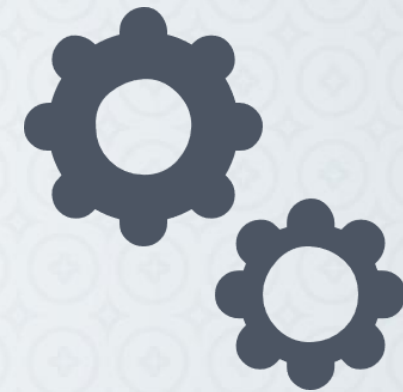


Рисунок 1 – Сервис выполнения



```
FROM openjdk:17-jdk  
  
RUN microdnf install -y python3  
RUN microdnf install gcc python3-devel  
RUN python3 -m pip install psutil
```

Рисунок 2 – Часть Dockerfile для образа

Структура задачи

- Типы данных для входных параметров и результата
- Тестовые данные, проверяющие правильность решения.
- Шаблон решения (рисунок 1 - 2) для каждого возможного языка



```
1 v class Solution{
2 v     public int add(int a, int b){
3
4     }
5 }
```

Рисунок 3 - Пример шаблона

```
1 v |class Solution:
2     def add(s,a,b):
```

Рисунок 4 - Пример шаблона

Проверка правильности решения

- Каждая задача содержит набор тестовых данных, который проверяет правильность решений, отправляемых пользователем.
- Тест состоит из массива входных данных и соответствующего ответа.
- Может быть несколько вариантов ответов для одних входных данных.
- Входные и выходные данные это просто строки, что упрощает тестирование.

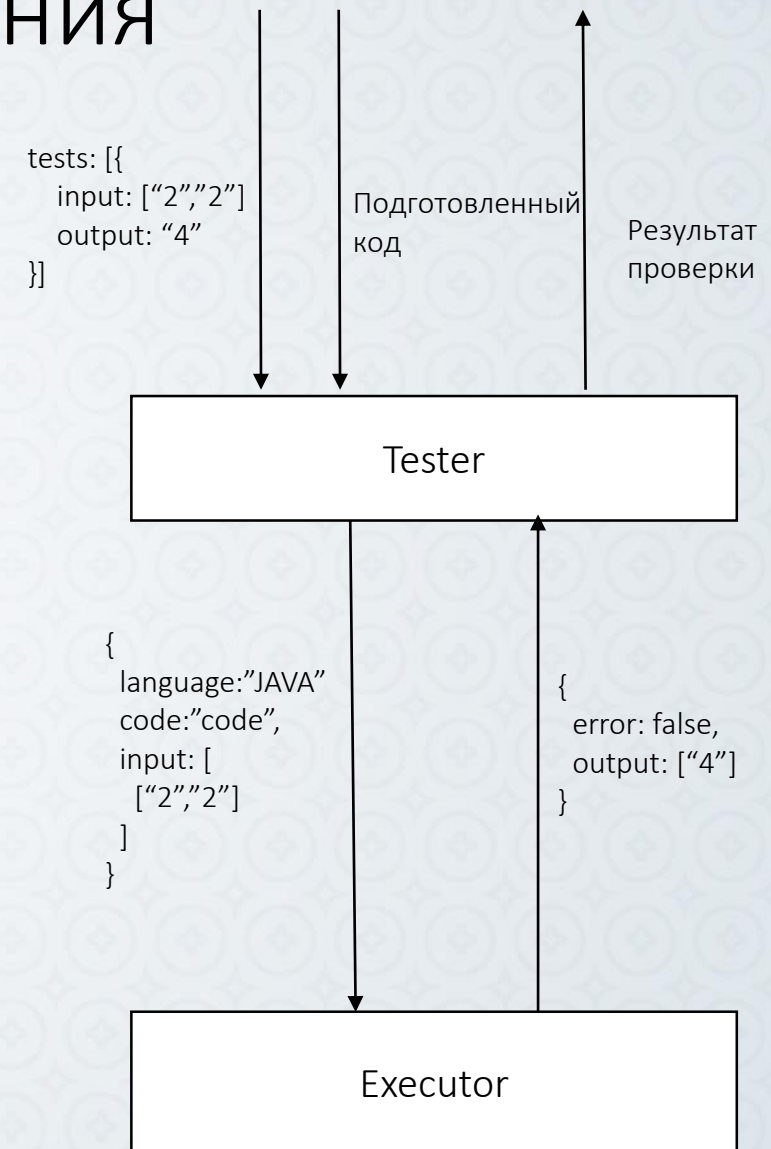


Рисунок 5 – Проверка решения

Процесс проверки решения

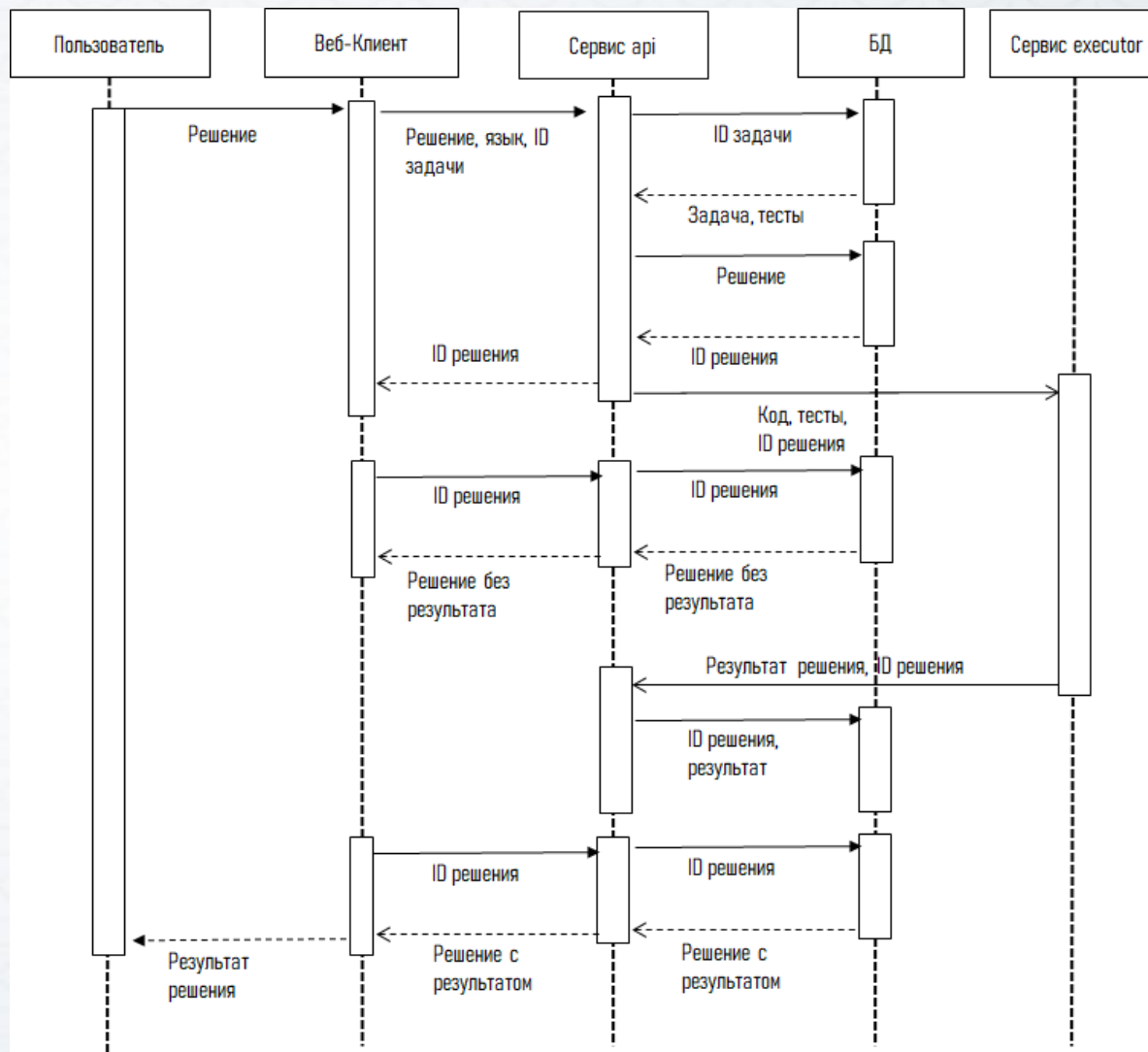


Рисунок 6 – Проверка решения

Архитектура платформы

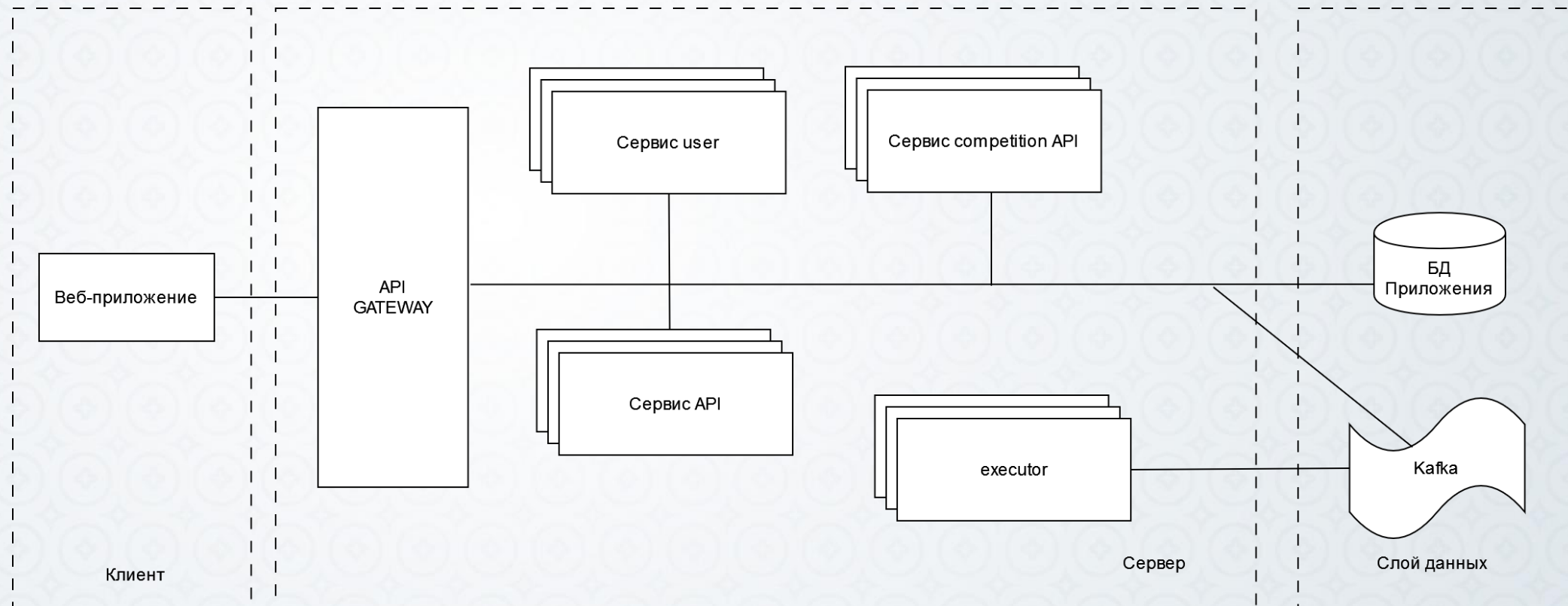


Рисунок 7 – Архитектура

Инвалидация in-memory кеша

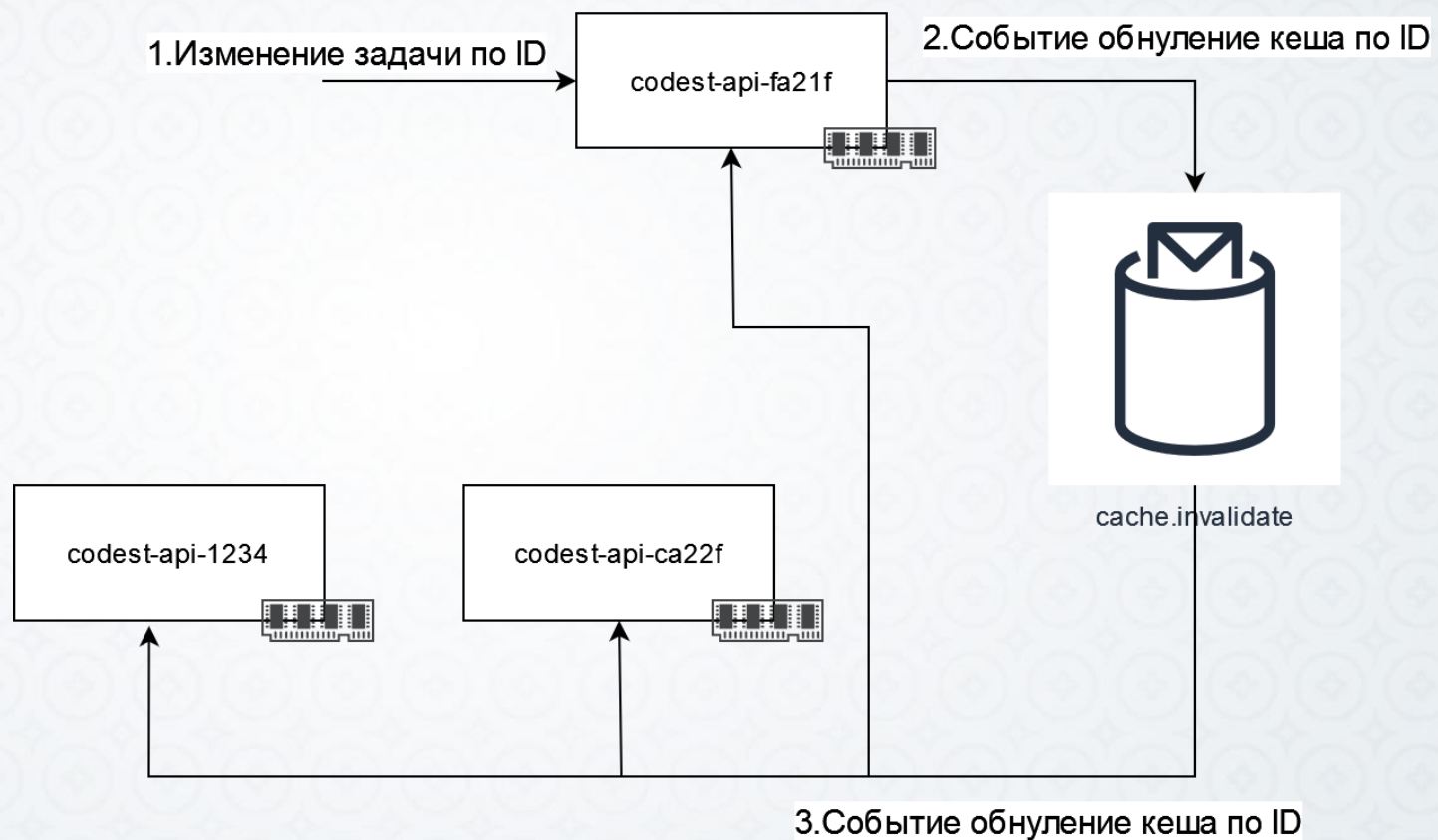


Рисунок 8 – Инвалидация кеша

Инфраструктура

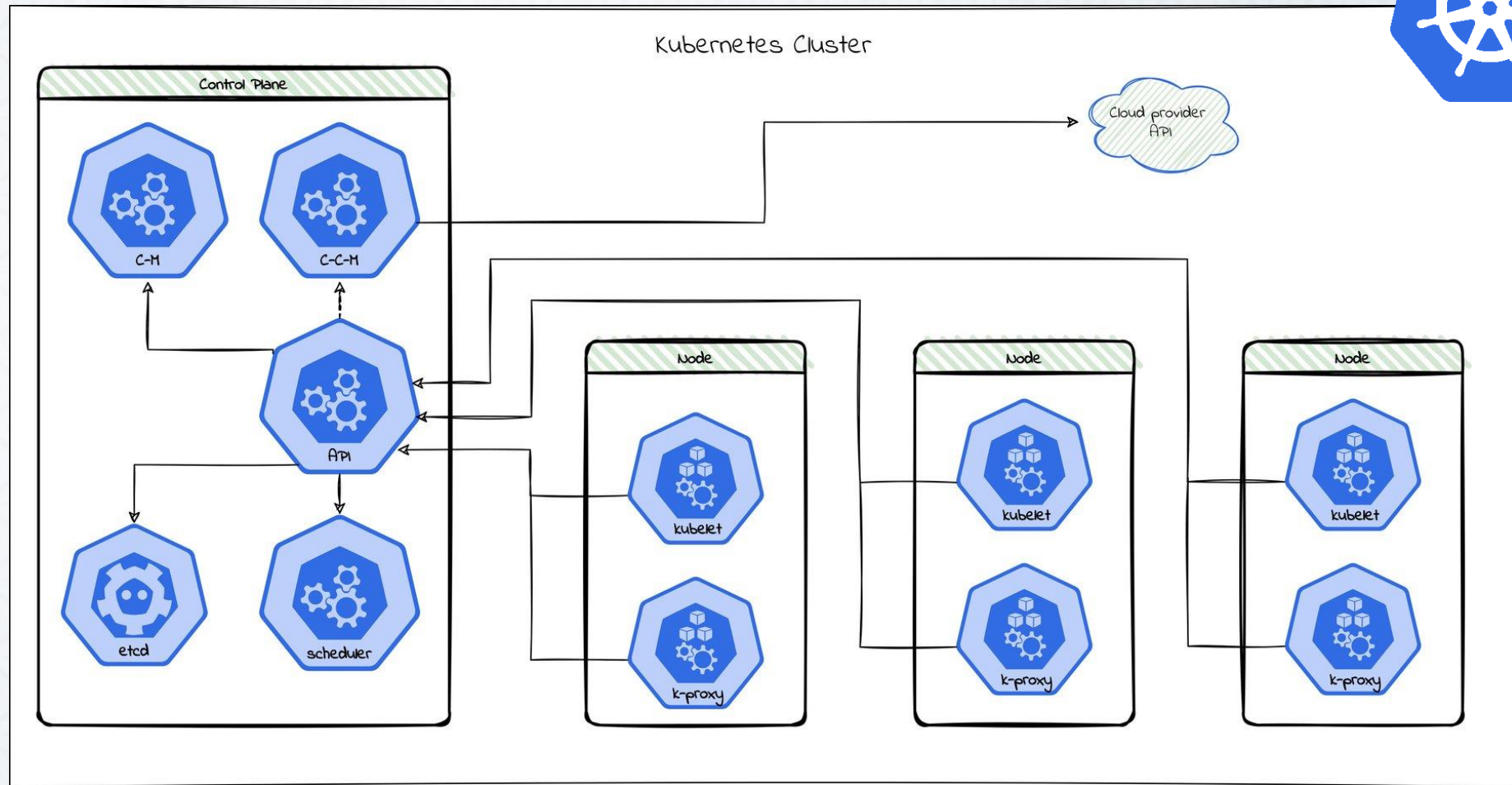


Рисунок 9 – Инфраструктура

Спасибо за внимание



<https://codestopen.r3nny.ru/task>