

Разработка визуального редактора политик сетевой безопасности для Kubernetes

Щанников Михаил Викторович

Группа: K0411-22

Специальность: 10.02.04 Обеспечение информационной
безопасности телекоммуникационных систем

Актуальность и проблема

Контекст: Kubernetes — стандарт индустрии.

Инструмент безопасности: Сетевые политики (Network Policies).

Главная проблема: Ручная настройка политик через **YAML** это:

- **Сложно:** Вложенная структура, строгие отступы.
- **Рискованно:** Опечатка в метке или порте — критическая уязвимость.
- **Непрозрачно:** Невозможно понять общую картину из десятков файлов.

Вывод: Существует острая потребность в инструменте, который делает управление политиками наглядным, интуитивным и безопасным.

Анализ существующих решений

Ручные и CLI-инструменты

kubectl:

Полностью ручная работа с YAML.

Calico / Cilium CLI:

Привязка к экосистеме (вендор-лок).

Визуальные инструменты (UI)

Lens / MKE UI:

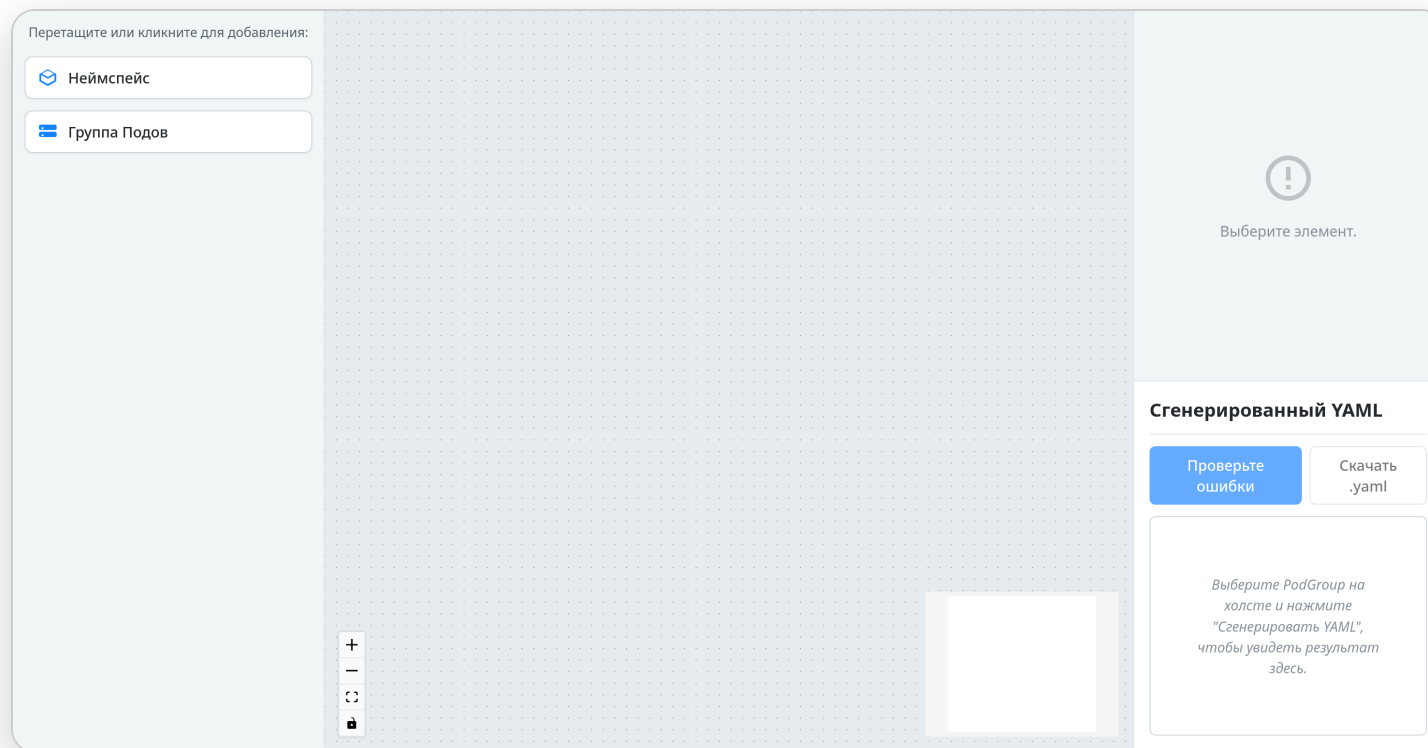
Часто лишь "оболочка" для YAML.

Cilium Editor / Sysdig:

Не решают проблему для стандартных политик.

Решение: Визуальный редактор

- **Интуитивное проектирование**
 - › Drag-and-drop
 - › Визуальные связи
- **Интерактивная настройка**
 - › Редактор в Инспекторе
 - › Валидация в реальном времени
- **Автоматическая генерация**
 - › Корректный YAML в один клик

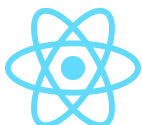


Технологический стек



TypeScript

Строгая типизация и надежность кода.



React

Компонентная архитектура UI.



ReactFlow

Создание интерактивного холста.



Vite

Современная и быстрая сборка проекта.



Zustand

Легковесное управление состоянием.



Jest & Playwright

Комплексное Unit и E2E тестирование.

Обеспечение качества: Комплексное тестирование

Модульное и Интеграционное тестирование (Jest)

Проверена корректность бизнес-логики (генерация YAML, валидация) и правильность взаимодействия UI-компонентов.

Сквозное тестирование (Playwright)

Проверены полные сценарии работы пользователя, эмулируя реальные действия в браузере от начала до конца.

Результат

Подтверждена стабильность работы прототипа и корректность генерируемых конфигураций для всех тестовых сценариев.

Заключение и перспективы

Достигнутые результаты

- ✓ **Цели проекта выполнены:** создан прототип.
- ✓ **Проблема решена:** процесс упрощен.
- ✓ **Качество подтверждено:** тесты пройдены.
- ✓ **Практическая значимость:** ГОТОВ К работе.

Перспективы развития

- › Реализация импорта YAML-файлов.
- › Прямая интеграция с Kubernetes API.
- › Поддержка расширенных политик.
- › Сохранение и загрузка проектов.

Спасибо за внимание!

**Готов ответить
на ваши вопросы.**