Секция «Цифровая трансформация образования и новые технологии обучения»

Проектирование и разработка системы сбора и анализа данных об утечках метана и углекислого газа

Научный руководитель – Демин Антон Юрьевич

Зарубин Владислав Александрович

Студент (бакалавр)

Национальный исследовательский Томский политехнический университет, Институт кибернетики, Томск, Россия

E-mail: vladislavzarubin113@gmail.com

Введение

Контроль выбросов парниковых газов является одной из ключевых задач для России в рамках выполнения международных обязательств по снижению воздействия на климат. В 2023 году подписан Указ Президента Российской Федерации [1], устанавливающий цель достижения углеродной нейтральности к 2060 году. В развитие этой инициативы принят Федеральный закон "Об ограничении выбросов парниковых газов" [2], обязывающий предприятия вести учет выбросов и предоставлять отчетность.

В целях повышения эффективности экологического мониторинга в данной статье рассматривается централизованная система автоматизированного сбора, обработки и анализа данных о выбросах метана и углекислого газа. Решение интегрируется с датчиками, БПЛА и другими средствами контроля утечек, обеспечивая объединение данных со всех источников в единой базе данных. Данная система позволит устранить проблему сбора данных[3] и повысит оперативность выявления превышений допустимых норм. Внедрение современных методов обработки данных и прогнозирования обеспечивает точное определение источников выбросов и оптимизацию процессов контроля.

Основная часть

Разработана микросервисная система для мониторинга и анализа выбросов парниковых газов, обеспечивающая автоматизированный сбор, обработку и прогнозирование данных. В качестве архитектурного подхода выбрана микросервисная модель, так как она позволяет разделить систему на независимые компоненты, что повышает её отказоустойчивость и гибкость. В отличие от монолитной архитектуры, микросервисы могут масштабироваться независимо, а обновление одного сервиса не требует изменения всей системы, что снижает риски и упрощает поддержку[4].

API-сервис реализован с использованием FastAPI, так как он обладает высокой производительностью, поддерживает асинхронное программирование и автоматическую генерацию документации. В отличие от Flask, который требует дополнительных библиотек для асинхронности, и Django, который может быть избыточным для задач, связанных с API, FastAPI предоставляет удобный баланс между скоростью разработки и производительностью[5]. Этот сервис отвечает за сбор данных из различных источников, включая датчики мониторинга и лабораторные системы, проводит их валидацию и передаёт в аналитический сервис.

Аналитический сервис выполняет анализ данных с использованием библиотек pandas, numpy и scikit-learn. Данный сервис выполняет прогнозирование выбросовов. Сервис выявляет аномалий и формирует отчёты, используя алгоритмы линейной регрессии Для хранения данных используется PostgreSQL, поскольку она сочетает высокую надёжность, поддержку сложных типов данных и хорошую масштабируемость.

Взаимодействие между сервисами организовано через очередь сообщений Apache Kafka, что позволяет минимизировать задержки и гарантировать доставку данных даже в усло-

виях временных сбоев. В отличие от традиционного REST API, Kafka обеспечивает асинхронное взаимодействие между компонентами, что критически важно для системы реального времени с высокой нагрузкой[6]. Для мониторинга системы интегрированы Prometheus и Grafana, позволяющие собирать метрики, анализировать динамику выбросов и визуализировать данные.

Развертывание системы осуществляется с использованием Docker и Kubernetes, что обеспечивает удобное управление контейнерами и их автоматическое масштабирование. Docker позволяет изолировать каждую службу, упрощая развертывание, а Kubernetes обеспечивает автоматическое восстановление сервисов и балансировку нагрузки. В отличие от традиционного подхода с виртуальными машинами, контейнеризация значительно снижает накладные расходы и ускоряет развертывание новых экземпляров сервисов[7].

Заключение

Разработанная система мониторинга и анализа выбросов парниковых газов обеспечивает точный сбор, обработку и прогнозирование данных, помогая оперативно выявлять утечки и снижать экологические риски. Микросервисная архитектура, контейнеризация и асинхронные API обеспечивают надежность, масштабируемость и высокую производительность. Интеграция с Prometheus и Grafana позволяет визуализировать данные и реагировать на аномалии в режиме реального времени. Решение упрощает контроль выбросов, автоматизирует анализ и поддерживает адаптацию к изменяющимся требованиям.

Источники и литература

1) Путин утвердил достижение углеродной нейтральности к 2060 году в новой доктрине // Коммерсантъ: сайт. – URL: https://www.kommersant.ru/doc/6298763 (дата обращения: 01.03.2025).
2) Федеральный закон "Об ограничении выбросов парниковых газов" от 02.07.2021 N 296-ФЗ (последняя редакция)// КонсультПлюс: сайт. – URL: https://www.consultant.ru/document/cons_doc_LAW_388992/ (дата обращения 01.03.2025)
3) Управление выбросами парниковых газов // Цифровая Энергетика: доклад «Центр 2М.» – URL: https://www.digital-energy.ru/wp-content/uploads/2021/07/Цифровизация-CO2_Центр2М.pdf (дата обращения 01.03.2025)
4) Монолитная и микросервисная архитектура. Сравнение// Хабр: сайт. – URL: https://habr.com/ru/companies/haulmont/articles/758780/ (дата обращения 02.03.2025)
5) Comparison of Flask, Django, and FastAPI: Advantages, Disadvantages, and Use Cases // Medium: site – URL: https://medium.com/@tubelwj/comparison-of-flask-django-and-fastapi-advantages-disadvantages-and-use-cases-63e7c692382a (дата обращения 02.03.2025)
6) What are event-driven APIs? Understanding event-driven vs REST API interactions // Axway: site – URL: https://blog.axway.com/learning-center/apis/basics/event-driven-vs-rest-api-interactions (дата обращения 02.03.2025)
7) Введение в Kubernetes: преимущества для разработки // Timeweb-cloud: site – URL: https://timeweb.cloud/tutorials/kubernetes/kubernetes-preimushchestva-dlya-razrabotki(дата обращения 02.03.2025)

Иллюстрации

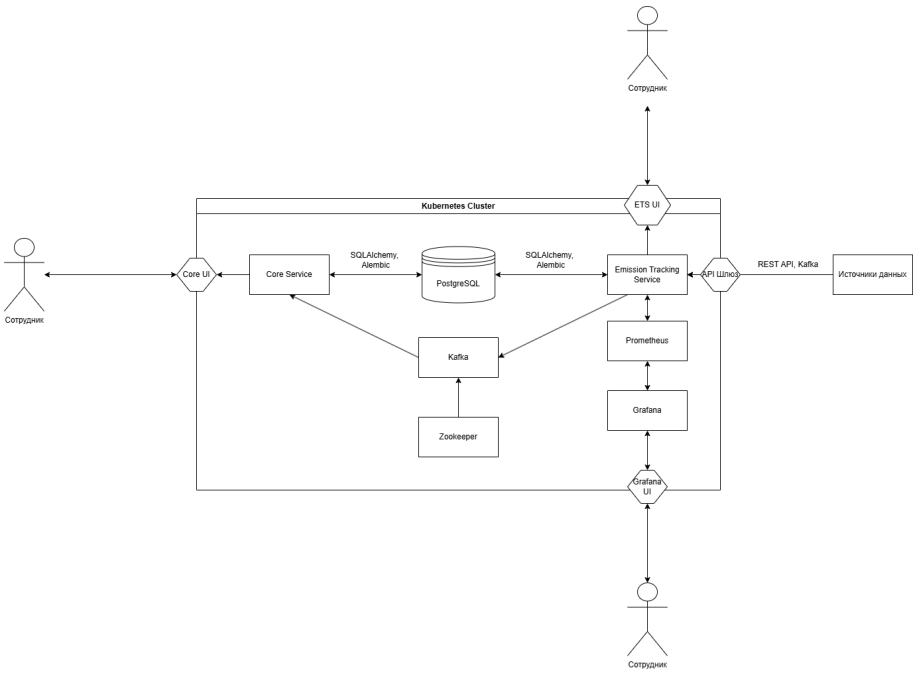

Рис. : Архитектура системы с использованием Kubernetes