

**Ерофеев Ермак Владимирович**,  
студент, Ангарский государственный технический университет,  
e-mail: ermak.080@mail.ru

**Кулакова Ирина Михайловна**,  
к.т.н., доцент, Ангарский государственный технический университет,  
e-mail: iyelkina@mail.ru

## **СИСТЕМА МОНИТОРИНГА DOCKER-КОНТЕЙНЕРОВ**

**Erofeev E.V., Kulakova I.M.**

### **DOCKER CONTAINER MONITORING SYSTEM**

**Аннотация.** Разработана система мониторинга Docker-контейнеров, которая представляет собой интегрированный сервис для мониторинга состояния контейнеров в реальном времени. Система использует передовые технологии, такие как Go, PostgreSQL, Nginx, RabbitMQ и React с TypeScript, для обеспечения надежности и масштабируемости. Взаимодействие сервисов через Docker-сеть позволяет легко управлять и адаптировать проект под различные эксплуатационные условия.

**Ключевые слова:** Docker-контейнеры, мониторинг, масштабируемость, интеграция.

**Abstract.** A docker - container monitoring system has been developed, which is an integrated service for monitoring the status of containers in real time. The system uses advanced technologies such as Go, PostgreSQL, Nginx, RabbitMQ, and React with TypeScript to ensure reliability and scalability. The interaction of services through the Docker network makes it easy to manage and adapt the project to various operational conditions.

**Keywords:** Docker-containers, monitoring, scalability, integration.

Docker-контейнер – это виртуальная среда, которая позволяет запускать приложения и их зависимости изолированно, независимо от операционной системы хоста. Мониторинг состояния Docker-контейнеров в реальном времени является критически важным для обеспечения надежности, производительности и безопасности приложений, работающих в контейнерах. Использование соответствующих инструментов мониторинга помогает командам DevOps и администраторам систем эффективно управлять своей инфраструктурой и быстро реагировать на возникающие проблемы.

Разработана система мониторинга docker-контейнеров, которая представляет собой интегрированный сервис для мониторинга состояния контейнеров в реальном времени. Система использует передовые технологии, такие как Go, PostgreSQL, Nginx, RabbitMQ и React с TypeScript, для обеспечения надежности и масштабируемости. Взаимодействие сервисов через Docker-сеть позволяет легко управлять и адаптировать проект под различные эксплуатационные условия. Это решение упрощает процесс управления контейнерной инфраструктурой, обеспечивая своевременную реакцию на изменения в состоянии контейнеров [1].

Все компоненты проекта интегрированы в единую Docker-сеть, управляемую с помощью Docker Compose, инструмента, который позволяет определять

и запускать многоконтейнерные Docker-приложения с помощью файла конфигурации.

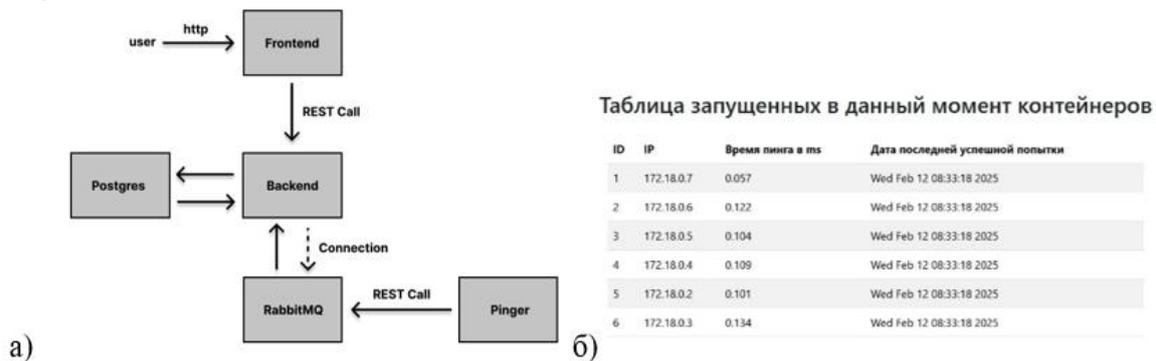


Рисунок 1 – Структура системы мониторинга docker-контейнеров (а) и интерфейс просмотра данных состояния контейнеров (б)

Структура проекта представлена на рисунке 1а и включает в себя:

- Backend (язык Go) – сервис, разработанный на языке Go, отвечает за обработку запросов к системе. Возвращает информацию о контейнерах с информацией о их текущем состоянии. Позволяет добавлять новые контейнеры.
- Асинхронный Pinger (язык Go) – сервис, регулярно проверяющий доступность контейнеров и отправляющий результаты в очередь RabbitMQ.
- RabbitMQ – сервис, который позволяет приложениям обмениваться сообщениями через очереди по протоколу AMQP.
- Frontend (React + TypeScript) – предоставляет пользователю интерфейс для просмотра данных о состоянии контейнеров (рисунок 1б).
- PostgreSQL служит надежным хранилищем данных, полученных от backend. Это обеспечивает долговременное хранение информации и возможность выполнения сложных запросов для анализа состояния контейнеров.

Взаимодействие между сервисами организовано таким образом, чтобы обеспечить максимальную эффективность и надежность передачи данных. Nginx управляет маршрутизацией запросов к backend, обеспечивая безопасность и балансировку нагрузки. Pinger асинхронно отправляет результаты опроса в backend через RabbitMQ, что позволяет backend оперативно обновлять данные в базе данных PostgreSQL. Frontend периодически запрашивает обновленные данные у backend и отображает их пользователю.

Таким образом, разработанное решение позволяет эффективно управлять контейнерной инфраструктурой, своевременно реагируя на изменения в состоянии Docker-контейнеров.

## ЛИТЕРАТУРА

1. Документация Docker [Электронный ресурс]. URL: <https://docs.docker.com/>. (Дата обращения: 01.02.2025).