

АВТОМАТИЗИРОВАННАЯ СИСТЕМА CI/CD ДЛЯ РАЗВЕРТЫВАНИЯ И МОНИТОРИНГА МИКРОСЕРВИСОВ

Валаханович В.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Цявловская Н.В. – магистр техн. наук, ст. преподаватель кафедры ИПиЭ

Аннотация. В статье описана система, реализующая современные подходы организации непрерывной интеграции и непрерывного развертывания (CI/CD) в архитектуре микросервисов. Анализируются ключевые технологии, используемые для автоматизации процессов сборки, тестирования и деплоя.

Ключевые слова: автоматизация, CI/CD, микросервисы, DevOps, контейнеризация, Kubernetes, Docker, GitHub Actions, мониторинг, Prometheus, Grafana, PostgreSQL.

Введение. Современная разработка программного обеспечения характеризуется высокой динамикой и требованиями к скорости развертывания и надежности сервисов. Традиционные монолитные архитектуры нередко оказываются неэффективными в условиях частых изменений и масштабируемости. В этой связи особую актуальность приобретает архитектура микросервисов, требующая автоматизированных процессов развертывания и сопровождения. В данных условиях ключевым инструментом является система CI/CD, обеспечивающая бесперебойную интеграцию, тестирование и развертывание микросервисов. Настоящая работа посвящена анализу подходов к построению эффективной системы CI/CD в микросервисной архитектуре.

Основная часть. В индустрии разработки программного обеспечения микросервисная архитектура становится все более востребованной. С другой стороны, традиционные подходы к интеграции и развертыванию сопровождаются высокими затратами времени, сложностями координации и рисками ошибок. Поэтому актуальной задачей является внедрение автоматизированных процессов CI/CD, обеспечивающих бесперебойную интеграцию, тестирование и развертывание микросервисов [1, 2].

Разработка эффективной системы CI/CD базируется на современных технологиях, позволяющих создать надежную, гибкую и масштабируемую инфраструктуру. В качестве инструмента автоматизации используется GitHub Actions, обеспечивающий выполнение сборочных, тестовых и развертываемых процессов. Контейнеризация реализована с помощью Docker, а управление развертыванием осуществляется через Kubernetes, что позволяет эффективно масштабировать микросервисные приложения. Для мониторинга используются Prometheus и Grafana, обеспечивающие сбор, анализ и визуализацию метрик, необходимых для контроля состояния системы [3]. Хранение данных реализовано в PostgreSQL, что обеспечивает целостность и доступность информации.

Разработка системы осуществляется на C# с использованием фреймворка .NET Core, что обеспечивает высокую производительность и кроссплатформенность. В качестве API-шлюза используется ASP.NET Core, обеспечивающий маршрутизацию запросов между микросервисами. Сборка и тестирование выполняются с использованием GitHub Actions, а для тестирования модулей применяется xUnit.

Каждый микросервис разворачивается в контейнере Docker, а управление развертыванием и масштабированием обеспечивается Kubernetes. База данных PostgreSQL подключается через Entity Framework Core, что позволяет эффективно управлять данными и поддерживать миграции схемы базы.

Для мониторинга и логирования интегрированы Prometheus и Grafana, а сбор логов осуществляется через EFK-стек (Elasticsearch, Fluentd, Kibana). Контроль за

отказоустойчивостью обеспечивается механизмами Kubernetes Health Checks и автоматическим перезапуском контейнеров при сбоях.

Внедрение системы CI/CD позволяет минимизировать время вывода изменений в эксплуатацию, повысить надежность развертывания и снизить количество ошибок, возникающих при интеграции кода. Автоматизация процессов тестирования и деплоя способствует ускорению выпуска обновлений, а мониторинг метрик обеспечивает оперативное реагирование на возникающие проблемы.

При разработке учитывались принципы эргономики и удобства администрирования, что позволило создать интуитивно понятную систему управления развертываниями. Оптимизированные процессы CI/CD обеспечивают удобство использования на различных платформах, минимизируют вероятность сбоев и повышают эффективность сопровождения микросервисов. Визуальный интерфейс мониторинга и логирования в Grafana упрощает анализ производительности и выявление проблем.

Функциональное и нагрузочное тестирование подтвердило стабильность работы системы при высокой нагрузке. Проверены корректность автоматического развертывания, мониторинга, управления ресурсами и интеграции с облачными сервисами. Архитектура, основанная на GitHub Actions, Kubernetes и PostgreSQL, гарантирует отказоустойчивость и безопасность данных [4].

Заключение. Разработанная система CI/CD для микросервисов обеспечивает автоматизацию процессов сборки, тестирования, развертывания и мониторинга, что снижает временные затраты и повышает стабильность работы программных продуктов. Использование контейнеризации на основе Docker и Kubernetes позволяет эффективно управлять развертыванием микросервисов, обеспечивая их масштабируемость и отказоустойчивость. Интеграция с GitHub Actions автоматизирует выполнение всех этапов CI/CD, а мониторинг с помощью Prometheus и Grafana обеспечивает контроль за состоянием системы и анализ ключевых метрик. Таким образом, внедрение данной системы способствует повышению эффективности процессов разработки и эксплуатации микросервисных приложений в условиях современных требований к надежности и скорости доставки программного обеспечения.

Список литературы

1. Маличенко, С. В. Проблемы перехода от монолитной к микросервисной архитектуре // Евразийский научный журнал. 2022. №5. С. 8–12.
2. Вальдивия Х. А., Лора-гонсалес А., Лимон К., Кортес-вердин К., Очаран-эрнандес Х. О. Паттерны микросервисной архитектуры: многопрофильный обзор литературы // Труды ИСП РАН. – 2021. – №1. – С. 81–95.
3. Маратканов А. С., Суханов А. А., Воробьева А. Ан. Средства анализа и визуализации метрик работы приложения // International scientific review. 2019. №LIX. С. 128–130.
4. Лазарев Н. А., Борисенко О. Д. Построение требований и архитектуры облачного оркестратора платформенных сервисов // Труды ИСП РАН. 2022. №4. С. 211–228.

UDC 004.418

CI/CD AUTOMATED SYSTEM FOR MICROSERVICE DEPLOYMENT AND MONITORING

Valakhanovich V.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Tsyavlovskaya N.V – Master of Sci., Senior Lecturer at the Department of EPE

Annotation. The paper describes a system that implements modern approaches to continuous integration and continuous deployment (CI/CD) in microservices architecture. The key technologies used to automate build, test and deployment processes are analyzed.

Keywords: automation, CI/CD, microservices, DevOps, containerization, Kubernetes, Docker, GitHub Actions, monitoring, Prometheus, Grafana, PostgreSQL.