Министерство образования Республики Беларусь
Учреждение образования

Белорусский государственный университет
информатики и радиоэлектроники

004.72, 004.75

Яцевич
Кирилл Витальевич

МЕТОДЫ РАЗВЕРТЫВАНИЯ МИКРОСЕРВИСНОГО ПРИЛОЖЕНИЯ В КЛАСТЕРЕ KUBERNETES

Автореферат
на соискание степени магистра
по специальности 7–06–0611–06 Системы и сети инфокоммуникаций

_____

Научный руководитель
канд. техн. наук,
доцент
ДАВЫДОВА Надежда
Сергеевна

_____

Минск 2026

ВВЕДЕНИЕ

Современные программные системы характеризуются высокой динамичностью, масштабируемостью и необходимостью быстрой адаптации к изменениям бизнес-требований. В этих условиях всё более широкое распространение получают микросервисные архитектуры, предполагающие декомпозицию приложения на независимые, четко изолированные сервисы. Такой подход обеспечивает гибкость разработки, упрощает обновление и сопровождение программных компонентов, а также позволяет эффективно масштабировать систему под нагрузкой. Однако вместе с тем микросервисная архитектура порождает новые технические вызовы, связанные с автоматизацией развертывания, управлением конфигурациями, мониторингом и обеспечением надежности.

Одним из наиболее распространённых решений для оркестрации контейнеризированных приложений является Kubernetes — платформа, предоставляющая развитые механизмы автоматического масштабирования, распределения нагрузки и управления конфигурациями. Kubernetes фактически стал индустриальным стандартом для развертывания микросервисов, однако разнообразие доступных стратегий и инструментов требует тщательного анализа и сравнения. Методы развертывания в рамках Kubernetes значительно различаются по сложности, обеспечиваемому уровню надежности и влиянию на доступность приложения. Важную роль играют такие механизмы, как rolling update, blue-green deployment, canary-развертывание, а также специализированные инструменты, включая Helm, Argo CD, Tekton и другие средства GitOps-подхода.

Настоящая работа посвящена исследованию и сравнительному анализу методов развертывания микросервисных приложений в кластерах Kubernetes. В рамках исследования рассматриваются ключевые стратегии доставки обновлений, их преимущества и ограничения, а также практические аспекты применения на реальных инфраструктурах. Результаты работы направлены на формирование рекомендаций по выбору оптимального подхода для обеспечения устойчивости, предсказуемости и эффективности процессов развертывания в микросервисных системах.

Объектом исследования в данной работе являются микросервисные программные системы, развертываемые в контейнеризованной среде и управляемые средствами оркестрации. Такие системы представляют собой совокупность автономных сервисов, взаимодействующих между собой и формирующих единое приложение, для которого критически важны устойчивость, гибкость обновления и способность к масштабированию.

Предметом исследования выступают методы и стратегии развертывания микросервисных приложений в кластере Kubernetes, включая механизмы доставки обновлений, особенности управления состоянием приложений и технологии автоматизации процессов CI/CD. В центре внимания находятся подходы, влияющие на доступность сервисов в процессе обновления, устойчивость к ошибкам, предсказуемость поведения системы и эффективность эксплуатации. Особое внимание уделяется сравнительному анализу различных стратегий развертывания и оценке их применимости в условиях реальных микросервисных архитектур.

Научная значимость данной работы заключается в систематизации и комплексном анализе методов развертывания микросервисных приложений в среде Kubernetes, что способствует более глубокому пониманию механизмов обеспечения надежности и устойчивости современных распределённых систем. Полученные в ходе исследования результаты расширяют теоретическую базу, связанную с архитектурой микросервисов и технологиями контейнерной оркестрации, а также позволяют уточнить критерии сравнения различных стратегий развертывания. Обобщение подходов и выявление их характерных особенностей создают основу для дальнейших научных разработок в области повышения отказоустойчивости и оптимизации процессов доставки программного обеспечения.

Практическая значимость работы определяется её ориентацией на решение задач, возникающих при эксплуатации микросервисных систем в условиях реальной инфраструктуры. Результаты исследования могут быть использованы специалистами DevOps, инженерами по эксплуатации и архитекторами при выборе оптимальных стратегий развертывания, построении процессов CI/CD и проектировании надежных систем на базе Kubernetes.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Связь работы с крупными научными программами

Тема диссертационной работы соответствует пункту 1 приоритетных направлений научной, научно-технической и инновационной деятельности Республики Беларусь на 2021–2025 гг., утвержденных Указом Президента Республики Беларусь №156 от 7 мая 2020 г. «Информационно-управляющие системы». Работа выполнялась в учреждении образования «Белорусский государственный университет информатики и радиоэлектроники»

Цель и задачи исследования

Целью данной работы является комплексный анализ и систематизация методов развертывания микросервисных приложений в кластере Kubernetes, а также формирование рекомендаций по выбору наиболее эффективных стратегий доставки обновлений с учётом архитектурных, требований.

Для достижения поставленной цели в диссертации решены следующие задачи:

1. выполнен анализ современных подходов к построению микросервисных приложений и использованию контейнерных технологий в распределённых информационных системах;
2. рассмотрены архитектурные и функциональные особенности платформы Kubernetes как среды развертывания и эксплуатации микросервисных приложений;
3. проведен анализ применимости различных методов развертывания в зависимости от сценариев эксплуатации микросервисных приложений;
4. обобщены результаты исследования и сформулированы выводы о целесообразности применения различных стратегий развертывания в Kubernetes-кластерах.

Личный вклад соискателя ученой степени

Содержание диссертации отображает личный вклад автора. Он заключается в самостоятельном выполнении теоретического анализа методов развертывания микросервисных приложений в Kubernetes-кластере, а также в обобщении и систематизации существующих подходов и практик. Соискателем была разработана система критериев для сравнения стратегий развертывания с учетом показателей реализации, требований к инфраструктуре и условий эксплуатации. Автором выполнен анализ применимости различных стратегий развертывания в зависимости от сценариев эксплуатации микросервисных приложений. Кроме того, соискателем подготовлено тестовое микросервисное приложение и разработаны конфигурационные файлы Kubernetes, использованные для иллюстрации и подтверждения полученных выводов.

Определение целей и задач исследований, интерпретация и обобщение полученных результатов проводилось совместно с научным руководителем к.т.н., доцентом Давыдовой Надеждой Сергеевной.

Апробация диссертации и информация об использовании ее результатов

Основные положения и результаты диссертационной работы докладывались и обсуждались на 60-ой и 61-ой научных конференциях аспирантов, магистрантов, и студентов БГУИР, а также на консультации с научным руководителем.

Опубликование результатов диссертации

По результатам исследования, представленных в диссертации, опубликованы 2 статьи в сборниках и материалах конференций общий объем 6 авторских листов.

Структура и объем диссертации

Общий объем диссертационной работы составляет страницы, из введения, общей характеристики работы, трех глав с выводами по каждой, заключения, библиографического списка, одного приложения.

Общий объем диссертационной работы составляет 78 страниц, из них 63 страниц текста, 5 рисунков на 5 страницах, 3 таблица на 3 страницах, список использованных библиографических источников (10 наименований на 17 страницах), список публикаций автора по теме диссертации (2 наименования на 1 странице), 4 приложения на 15 страницах.

Проверка на уникальность

Проведена экспертиза диссертации Яцевича Кирилл Витальевич «Методы развертывания микросервисного приложения в кластере Kubernetes» на корректность использования материалов с применением сетевого ресурса «Антиплагиат» (адрес доступа: https://antiplagiat.ru) в on-line режиме 20.12.2025 г. В результате проверки установлена корректность использования заимствованных материалов (оригинальность диссертационной работы составляет 87 %)

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении рассмотрены проблемы, связанные с необходимостью разработки и внедрения эффективных методов развертывания микросервисных приложений в кластере Kubernetes. Обоснована актуальность обеспечения высокой доступности, надежности и управляемости процессов обновления приложений в условиях динамичного развития распределённых и облачных информационных систем. Показано,

что использование современных стратегий развертывания позволяет минимизировать простои и риски при внедрении изменений.

В общей характеристике работы показана связь исследования с приоритетными направлениями развития информационных технологий и облачных вычислений, сформулированы цель и задачи исследования, отражён личный вклад соискателя ученой степени, а также представлены сведения об апробации результатов диссертационной работы.

В первом разделе выполнен обзор современных подходов и инструментов развертывания микросервисных приложений в Kubernetes-кластере. Проведён анализ базовых стратегий развертывания, таких как Recreate, Rolling Update, Canary Deployment и Blue-Green Deployment. Установлено, что каждая из рассмотренных стратегий обладает определёнными преимуществами и ограничениями, связанными с доступностью приложения, сложностью конфигурации и требованиями к вычислительным ресурсам, что обуславливает необходимость их осознанного выбора в зависимости от условий эксплуатации.

Во втором разделе выполнен анализ требований к тестовому микросервисному приложению и инфраструктуре Kubernetes-кластера. Определена цель работы и сформулированы задачи, направленные на разработку и сравнительный анализ методов развертывания, а также на количественную оценку их эксплуатационных характеристик. Стратегия Recreate Deployment является одной из базовых и наиболее простых стратегий развертывания приложений в Kubernetes. Данный подход предполагает полную остановку всех экземпляров приложения предыдущей версии перед запуском новой версии, что принципиально отличает его от более сложных стратегий, ориентированных на непрерывную доступность сервиса. Несмотря на наличие периода недоступности приложения, стратегия Recreate Deployment продолжает использоваться на практике, особенно в средах с невысокими требованиями к отказоустойчивости и непрерывности обслуживания (см. рисунок 1.1).

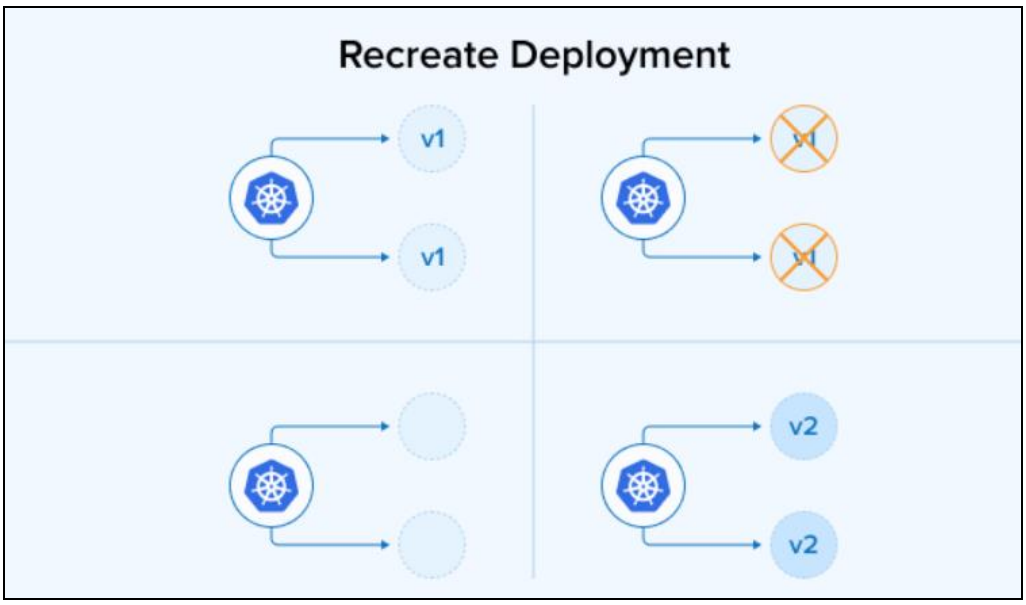

Рисунок 1.1 – Обзор метода recreate deployment

В основе стратегии Rolling Update лежит принцип поэтапной замены экземпляров приложения. В процессе обновления Kubernetes не останавливает все поды одновременно, а последовательно выводит из эксплуатации отдельные экземпляры старой версии, параллельно запуская поды с новой версией контейнерного образа. Таким образом, в кластере одновременно функционируют поды нескольких версий приложения, что позволяет системе продолжать обработку пользовательских запросов на протяжении всего процесса развертывания. Данный подход принципиально отличает Rolling Update от стратегии Recreate Deployment, при которой неизбежен период полного простоя сервиса (см. рисунок 1.2).

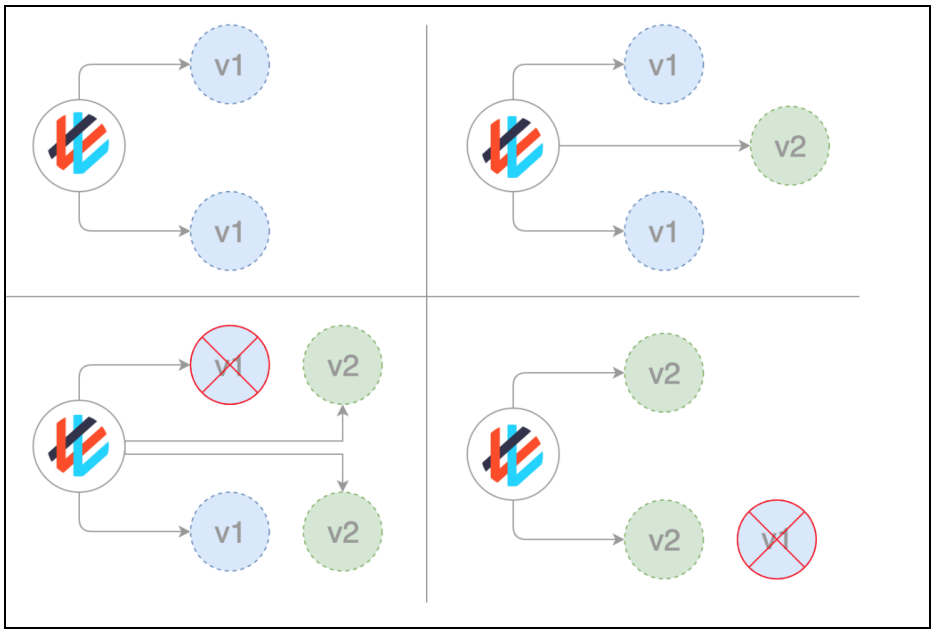

Рисунок 1.2 – Обзор метода rolling update

Стратегия Blue-Green Deployment относится к числу продвинутых методов развертывания приложений в Kubernetes и ориентирована на обеспечение максимально высокой доступности сервиса в процессе обновления. В отличие от стратегий Recreate Deployment и Rolling Update, данный подход предполагает одновременное существование двух полностью изолированных окружений приложения, представляющих разные версии одного и того же сервиса. Такая архитектура позволяет выполнять обновление практически мгновенно с точки зрения конечных пользователей и обеспечивает удобный механизм отката изменений (см. рисунок 2.3).

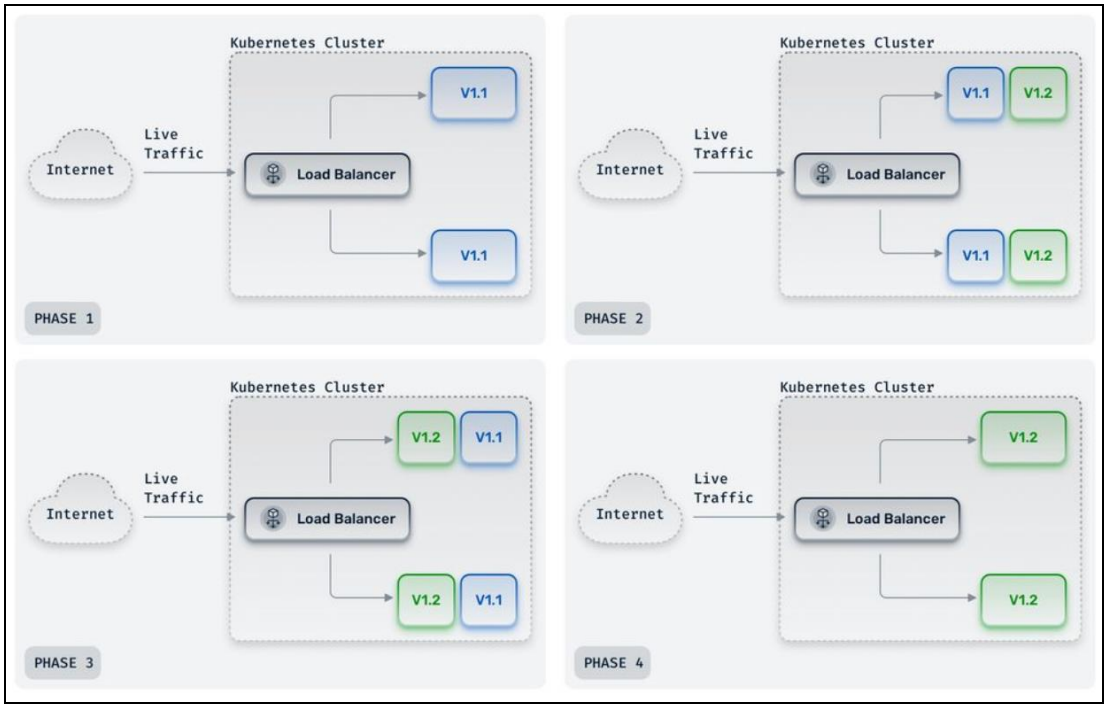

Рисунок 2.3 – Обзор метода blue-green deployment

Исторически термин «canary deployment» восходит к практике использования канареек в шахтах, где птицы служили индикатором опасных условий. Аналогичным образом в программной инженерии канареечное развертывание используется как механизм раннего обнаружения дефектов и отклонений в поведении новой версии приложения. В отличие от традиционных стратегий обновления, предполагающих одномоментный или поэтапный переход всей системы на новую версию, Canary Deployment ориентирован на ограниченное воздействие изменений на производственную среду (см. рисунок 2.4).

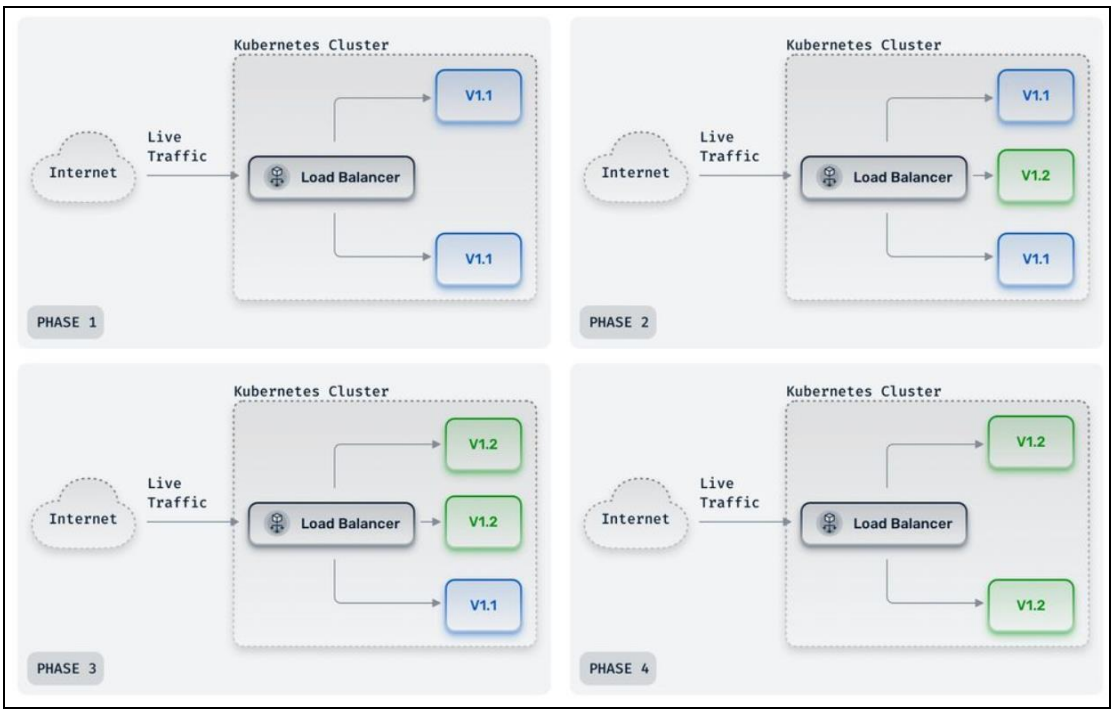

Рисунок 2.4 – Обзор метода canary deployment

Таким образом, Kubernetes предоставляет комплекс взаимосвязанных механизмов, которые в совокупности позволяют реализовывать широкий спектр стратегий развертывания приложений. Гибкость и расширяемость платформы делают возможным адаптацию этих механизмов под конкретные требования системы и уровень зрелости процессов разработки и эксплуатации. Рассмотренные механизмы образуют технологическую основу для реализации стратегий Recreate Deployment, Rolling Update, Blue-Green Deployment и Canary Deployment и обеспечивают их практическую применимость в современных микросервисных архитектурах. Использование данных механизмов позволяет не только автоматизировать процесс развертывания, но и повысить надёжность, управляемость и предсказуемость обновлений программного обеспечения. Это делает Kubernetes универсальной платформой для реализации стратегий развертывания в распределённых системах и обосновывает её широкое применение в современных облачных инфраструктурах.

В третьем разделе обоснован выбор архитектуры тестового микросервисного приложения и разработаны конфигурации Kubernetes-манифестов для реализации различных стратегий развертывания. Рассмотрены параметры управления жизненным циклом приложений, механизмы маршрутизации трафика и автоматического отката. Для продвинутых сценариев обновления использованы инструменты Argo CD и

Argo Rollouts, обеспечивающие реализацию подхода GitOps и формализацию процессов Canary и Blue-Green Deployment.

Для проведения объективного сравнительного анализа стратегий развертывания микросервисных приложений в Kubernetes-кластере в работе используется набор критериев, ориентированных на количественную оценку эксплуатационных и реализационных характеристик. Выбранные критерии позволяют сопоставить стратегии Recreate, Rolling Update, Canary Deployment и Blue-Green Deployment на основе измеряемых показателей, получаемых в ходе экспериментального развертывания тестового микросервисного приложения. К критериям относится следующее:

1. Время развертывания новой версии приложения — интервал времени от начала процесса обновления до момента, когда новая версия приложения становится доступной для пользовательского трафика. Данный показатель измеряется в секундах и отражает скорость внедрения изменений при использовании конкретной стратегии развертывания.
2. Доля недоступных экземпляров приложения в процессе обновления — процент подов, находящихся в недоступном состоянии в ходе развертывания новой версии. Критерий характеризует влияние стратегии развертывания на непрерывность обслуживания и измеряется в процентах от общего числа подов.
3. Количество ошибок HTTP-ответов — число ответов с кодами ошибок (4xx и 5xx), зафиксированных в процессе обновления приложения. Данный показатель используется для оценки стабильности системы при развертывании и измеряется в абсолютных значениях или в виде процента от общего числа запросов.
4. Среднее время отклика приложения — среднее значение времени обработки HTTP-запросов в период развертывания новой версии. Показатель измеряется в миллисекундах и позволяет оценить влияние стратегии развертывания на производительность приложения.
5. Возможность и скорость отката к предыдущей версии — время, необходимое для восстановления предыдущей стабильной версии приложения после выявления ошибок в новой версии. Данный критерий измеряется в секундах и отражает оперативность реакции системы на отказовые ситуации.
6. Избыточное потребление вычислительных ресурсов — дополнительное использование ресурсов процессора и оперативной памяти в процессе развертывания по сравнению с нормальным режимом работы. Критерий измеряется в процентах и позволяет

оценить нагрузку на инфраструктуру при применении различных стратегий.

7. Сложность конфигурации и управления развертыванием — количество конфигурационных объектов Kubernetes и дополнительных инструментов, необходимых для реализации стратегии. Показатель выражается в количественном виде и используется для оценки сложности внедрения и сопровождения стратегии.
8. Гранулярность управления пользовательским трафиком — минимальный шаг изменения доли трафика, направляемого на новую версию приложения. Критерий измеряется в процентах и отражает степень контроля над процессом внедрения изменений.

Использование указанных критериев обеспечивает возможность проведения формализованного и воспроизводимого сравнения стратегий развертывания микросервисных приложений. Каждый критерий допускает количественную оценку и может быть представлен в виде таблиц, графиков и диаграмм, что соответствует требованиям к аналитической части магистерской диссертации и позволяет обоснованно сопоставить рассматриваемые подходы. Сценарии эксплуатации микросервисных приложений в Kubernetes неизбежно эволюционируют по мере развития системы и изменения внешних условий. Стратегия развертывания, оптимальная на одном этапе, может утратить свою применимость на другом. В этой связи анализ применимости стратегий должен рассматриваться как непрерывный процесс, сопровождающий жизненный цикл системы, а не как разовое архитектурное решение.

Таким образом, анализ применимости стратегий развертывания в различных сценариях эксплуатации показывает, что эффективность той или иной стратегии определяется совокупностью технических, инфраструктурных, организационных и бизнес-факторов. Отсутствие универсального решения делает необходимым гибкий и контекстно-зависимый подход к выбору стратегии развертывания. Комплексное понимание эксплуатационных сценариев позволяет обоснованно соотносить возможности конкретных стратегий с реальными условиями функционирования микросервисных приложений в Kubernetes-кластере и формирует методологическую основу для принятия архитектурных и эксплуатационных решений.

По результатам экспериментальных испытаний установлено, что разработанные методы развертывания обеспечивают стабильную работу микросервисного приложения и соответствуют заявленным эксплуатационным характеристикам. Ошибок в конфигурации выявлено не было, а использование продвинутых стратегий развертывания позволило существенно повысить управляемость и надежность процесса обновления приложения.

ЗАКЛЮЧЕНИЕ

В рамках выполненной магистерской диссертации были рассмотрены и проанализированы методы развертывания микросервисных приложений в кластере Kubernetes с позиций современных требований к надежности, масштабируемости и эксплуатационной устойчивости распределенных информационных систем. Актуальность выбранной темы обусловлена широким распространением микросервисной архитектуры и контейнерных технологий, а также возрастающей сложностью процессов доставки и сопровождения программного обеспечения в условиях непрерывного развития и высокой динамики изменений.

В ходе исследования было показано, что Kubernetes, являясь стандартом оркестрации контейнеров, предоставляет универсальную и гибкую платформу для управления жизненным циклом микросервисных приложений, однако сам по себе не определяет оптимальный способ их развертывания. В теоретической части работы были рассмотрены основные принципы микросервисной архитектуры и контейнеризации, а также проанализированы базовые механизмы Kubernetes, используемые при развертывании приложений. Особое внимание было уделено рассмотрению стратегий развертывания как совокупности технических и организационных решений, влияющих на устойчивость системы в процессе обновления..

В работе была сформирована система критериев сравнения стратегий развертывания микросервисных приложений. Данные критерии охватывают показатели сложности реализации, требования к инфраструктурным ресурсам, влияние на доступность сервиса, возможности автоматизации и интеграции с CI/CD-процессами, а также эксплуатационные и организационные аспекты.

Важным результатом исследования, который представлен в таблице 1, стал анализ применимости стратегий развертывания в различных сценариях эксплуатации. Было показано, что эффективность стратегии определяется не только ее техническими характеристиками, но и соответствием конкретному эксплуатационному контексту, включающему масштаб системы, критичность предоставляемых сервисов, частоту обновлений, уровень автоматизации и зрелость процессов DevOps. Отсутствие универсальной

стратегии развертывания делает необходимым контекстно-зависимый подход к выбору решений, основанный на анализе реальных условий функционирования системы.

Таблица 1 – Сводная таблица сравнения методов развертывания

Критерий	Recreate	Rolling Update	Canary Deployment	Blue-green Deployment
Время развертывания, с	70	95	180	60
Время отката к предыдущей версии, с	65	80	45	20
Максимальная недоступность приложения, %	100	25	5	0
Доля HTTP-ошибок, %	4,5	1,8	0,6	0,2
Среднее время отклика, мс	260	230	210	200
Потребление вычислительных ресурсов, %	5	9	20	40
Сложность конфигурации	Низкая	Низкая	Высокая	Высокая
Минимальный шаг изменения трафика, %	100	100	5	

Особое внимание в диссертации было уделено сравнению стратегий развертывания по показателям реализации и требованиям к инфраструктуре. В ходе анализа было показано, что более сложные стратегии, обеспечивающие высокий уровень управляемости и отказоустойчивости, как правило, требуют наличия развитой инфраструктуры Kubernetes-кластера, дополнительных компонентов управления трафиком и мониторинга, а также высокой квалификации персонала. В то же время более простые стратегии обладают меньшими инфраструктурными требованиями, но могут быть ограничены в применимости для систем с высокими требованиями к непрерывности работы.

Полученные в ходе работы выводы подтверждают, что стратегия развертывания должна рассматриваться как динамический элемент архитектуры микросервисного приложения, подлежащий пересмотру и адаптации по мере развития системы и изменения требований. Выбор

стратегии на ранних этапах жизненного цикла может существенно отличаться от решений, оптимальных для стадии промышленной эксплуатации, что требует гибкости архитектурных и эксплуатационных подходов.

Практическая значимость выполненной работы заключается в возможности использования полученных результатов при проектировании и эксплуатации микросервисных приложений в Kubernetes-кластерах. Сформированная система критериев и проведенный анализ применимости стратегий могут быть использованы архитекторами и инженерами эксплуатации для обоснованного выбора стратегии развертывания с учетом конкретных технических и организационных ограничений. Результаты работы также могут быть использованы в образовательных целях при изучении современных подходов к управлению жизненным циклом микросервисных систем.

В ходе выполнения магистерской диссертации были достигнуты поставленные цели и решены основные задачи исследования. Полученные результаты подтверждают актуальность и практическую значимость выбранной темы, а также демонстрируют необходимость комплексного подхода к вопросам развертывания микросервисных приложений в Kubernetes. Направления дальнейших исследований могут включать экспериментальную оценку стратегий развертывания в реальных кластерах, анализ их влияния на показатели надежности и производительности, а также исследование новых подходов и инструментов, расширяющих возможности управления процессами развертывания в распределенных системах.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1–А. Яцевич, К.В. Методы развертывания микросервисных приложений в кластере Kubernetes/ Яцевич К.В. // 60-я научная конференция аспирантов, магистрантов и студентов БГУИР : тезисы докладов 60-ой научной конференции аспирантов, магистрантов и студентов БГУИР. Минск, 22-26 апреля 2024г. / редкол. : В. Ю. Цветков [и др.]. – Минск : БГУИР, 2024. – С. 170.

2–А. Яцевич, Построение эффективного CI/CD конвейера / Яцевич К.В. // 61-я научная конференция аспирантов, магистрантов и студентов БГУИР : тезисы докладов 61-ой научной конференции аспирантов, магистрантов и студентов БГУИР. Минск, 21-25 апреля 2025г. / редкол. : В. Ю. Цветков [и др.]. – Минск : БГУИР, 2024. – С. 125.