

## АВТОМАТИЗАЦИЯ КАНАРЕЕЧНОГО РАЗВЕРТЫВАНИЯ ПРИЛОЖЕНИЙ В KUBERNETES НА ОСНОВЕ АНАЛИЗА ЭКСПЛУАТАЦИОННЫХ МЕТРИК

Тимофеев Л. А.<sup>1</sup>, Чигвинцева Е. А.<sup>2</sup>

Научный руководитель – канд. техн. наук, доцент Марочкина А. В.<sup>1</sup>

<sup>1</sup>Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.

А. Бонч-Бруевича

<sup>2</sup>RDP.RU, Москва, Россия

[eliza.chia@ya.ru](mailto:eliza.chia@ya.ru), [other.saew@mail.ru](mailto:other.saew@mail.ru)

### Введение

Современная разработка программных продуктов - непрерывный процесс внедрения изменений. Критическим требованием при этом является обеспечение “бесшовного” обновления, не влияющего на пользовательский опыт. Эффективным способом решения данной задачи выступает стратегия постепенного развертывания (rollout deployment). Принцип работы базируется на поэтапном увеличении доли пользователей, взаимодействующих с обновленной версией, что дает возможность оценивать эффективность нововведений и выявлять возможные проблемы на раннем этапе, локализовать проблему в пределах малой части аудитории. Существует несколько подходов, применяющих данную стратегию, в работах [1, 2] сравниваются их сильные и слабые стороны.

Как показано в работе [2], наиболее подходящим методом безопасного обновления является канареечное развертывание (Canary Deployment). Подход предполагает дозированную подачу обновлений: сначала новая версия тестируется на 5% пользователей, затем, при подтверждении стабильности, доля возрастает до, например, 20%, потом до 50% - и так до полного замещения предыдущей версии. При возникновении ошибок на любом этапе обновление можно откатить, затронув при этом лишь небольшую часть пользователей. Таким образом риск крупных сбоев сводится к минимуму.

Однако канареечное развертывание сопряжено с рядом сложностей. Во-первых, оно требует постоянного мониторинга ключевых метрик - частоты ошибок, времени отклика, нагрузки на систему и так далее. Поскольку решение о продолжении или остановке работ принимается на основе анализа этих данных, значительную часть приходится выполнять инженерам вручную, что замедляет процесс и создает риск ошибок. Во-вторых, возникает естественный конфликт между скоростью и безопасностью: слишком быстрый процесс обновления повышает вероятность сбоев, а чрезмерно осторожный заставляет пользователей долго ждать новых функций.

Решений, способных компенсировать эти недостатки, пока немного. Например, в работе [3] описан способ автоматического анализа канареечного развертывания (Automated Canary Analysis, ACA) на основе оценки метрик, позволяющий автоматизировать принятие решений. Однако на практике он часто дает ложные срабатывания из-за шумных метрик, пропускает скрытые ошибки, плохо масштабируется на сложные сервисы и миграции баз данных. В статье [4] предлагается более перспективный подход - прогнозирование поведения приложения с помощью моделей на основе машинного обучения. Но и здесь есть нерешенные вопросы: отсутствует задокументированная интеграция с системами оркестрации (например, Kubernetes), неясны детали обучения, внедрения таких моделей и их точность. Таким образом, проблема создания эффективных и практичных методов анализа постепенного внедрения пока не имеет универсального решения.

## Основная часть

Предлагаемый метод изменяет логику постепенного внедрения: вместо заранее установленного сценария (5% - 20% - 50% - 100%) решение о дальнейшем обновлении принимается на основе оценки текущего состояния системы. Анализ метрик ведется с поправкой на естественный шум в данных и динамику инфраструктуры (например, автоматическое добавление ресурсов).

В его основе лежит механизм принятия решений - любое обновление рассматривается, как процесс с обратной связью: система непрерывно анализирует метрики обновления, сопоставляя их с пороговыми значениями и показателями предыдущей стабильной версии. В своей работе она учитывает естественный разброс параметров, чтобы кратковременные отклонения не интерпретировались, как сбои, а также оценивает достоверность данных - решение принимается только при достижении требуемого уровня уверенности.

Особенностью является учет контекста развернутой инфраструктуры. Например, при автоматическом масштабировании ресурсов система распознает, что изменение показателей связано не с ошибкой, а с адаптацией к нагрузке. Это исключает ложные срабатывания.

В результате формируется адаптивная стратегия управления - скорость распространения новой версии изменяется динамически: высокая стабильность ускоряет обновление, а признаки ухудшения - замедляют или останавливают его. Такой метод превращает формальную процедуру в самоадаптирующийся процесс.

## Выводы

Цель работы - разработка адаптивного метода канареечного развертывания приложений в Kubernetes, обеспечивающего автоматическое принятие решений о продолжении или откате обновления на основе анализа эксплуатационных метрик.

Результаты ориентированы на применение в рамках методологии DevOps (взаимодействие разработки и эксплуатации). Апробация в эксплуатируемом Kubernetes-кластере подтвердила эффективность подхода.

## Литература

1. Rakshit H., Banerjee S. Scalability evaluation on zero downtime deployment in Kubernetes cluster // Proc. 2024 IEEE Calcutta Conference (CALCON). Kolkata, India, 2024. P. 1–5. <https://doi.org/10.1109/ACCESS.2024.3416087>.
2. Malhotra A., Elsayed A., Torres R., Venkatraman S. Evaluate canary deployment techniques using Kubernetes, Istio, and Liquibase for cloud native enterprise applications to achieve zero downtime for continuous deployments // IEEE Access. 2024 Vol. 12 P. 87883–87899. <https://doi.org/10.1109/ACCESS.2024.3416087>.
3. Khande V. N. S., Balaji N. V. J. Automated canary analysis for Kubernetes deployments // Proc. 2025 IEEE Cloud Summit. Washington, DC, USA, 2025. P. 101–109. <https://doi.org/10.1109/Cloud-Summit64795.2025.00023>.
4. Rajakumar J. M. T., Solanki S. Automated canary deployments in continuous delivery: balancing speed and reliability [Электронный ресурс]. – Режим доступа: [https://www.researchgate.net/publication/394069823\\_Automated\\_Canary\\_Deployments\\_in\\_Continuous\\_Delivery\\_Balancing\\_Speed\\_and\\_Reliability](https://www.researchgate.net/publication/394069823_Automated_Canary_Deployments_in_Continuous_Delivery_Balancing_Speed_and_Reliability) (Дата обращения 23.02.2026).