

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ
СБОРКИ, НАСТРОЙКИ И РАЗВЁРТЫВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Приори Льва Дмитриевича

Научный руководитель
профессор, д. ф.-м. н.

В. А. Молчанов

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2023

ВВЕДЕНИЕ

В современном информационном обществе существует постоянная потребность в эффективных методах автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения. Быстрая и надёжная поставка программных продуктов на рынок стала ключевым фактором успеха для многих организаций, и автоматизация данных процессов позволяет существенно сократить время, затрачиваемое на разработку и внедрение новых продуктов, а также повысить качество и надёжность программного обеспечения.

Целью данной работы является исследование и разработка методов и инструментов для автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения. Работа направлена на создание эффективной и гибкой системы, которая позволит значительно ускорить и упростить процесс разработки и доставки программных продуктов, а также повысить уровень автоматизации в рамках различных проектов и организаций.

Материалы исследования для данной работы включают следующие компоненты:

1. Технические статьи: Анализ и изучение актуальных технических статей, посвященных автоматизации технологических процессов сборки, настройки и развертывания программного обеспечения. Эти статьи предоставляют важную информацию о существующих методах, инструментах, технологиях и подходах к автоматизации процессов разработки и доставки программных продуктов.
2. Книги и учебные пособия: Изучение специализированных книг и учебных пособий, которые рассматривают принципы и методы автоматизации процессов разработки программного обеспечения. Эти материалы предоставляют теоретическую базу и практические рекомендации по применению автоматизации в различных аспектах сборки, настройки и развертывания ПО.
3. Открытый исходный код: Изучение открытых проектов и их исходного кода, связанных с автоматизацией технологических процессов разработки программного обеспечения. Это позволяет ознакомиться с реализацией конкретных автоматизированных решений и понять их преимуще-

щества и ограничения.

Структура выпускной квалификационной работы:

1. Необходимость внедрения devops-технологий
2. Методология DevOps
3. Реализация DevOps с использованием непрерывной интеграции и доставки
4. Реализация конвейера автоматической доставки кода

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе "Необходимость внедрения DevOps-технологий" рассматривается актуальность исследуемой темы, а также формулируются цель и задачи работы.

В современном информационном обществе быстрая и эффективная разработка программного обеспечения становится все более важным фактором успеха для организаций. Рынок требует постоянных обновлений и улучшений продуктов, а пользователи становятся более требовательными к их качеству и функциональности. Однако, традиционные методы разработки и доставки ПО часто становятся преградой для достижения высокой производительности и гибкости. [1]

В этой главе исследуется необходимость внедрения DevOps-технологий, которые предлагают новый подход к организации и автоматизации технологических процессов сборки, настройки и развертывания программного обеспечения. Этот подход позволяет организациям более эффективно и гибко реагировать на изменения рынка и потребностей пользователей.

Актуальность темы обусловлена несколькими факторами. Во-первых, конкурентная борьба на рынке ПО требует сокращения времени от идеи до выпуска новых продуктов. DevOps-практики и инструменты позволяют автоматизировать и ускорить процессы разработки, тестирования и развертывания ПО, что способствует сокращению времени до его выпуска и повышает отзывчивость организации.

Во-вторых, сложность программных систем требует тесной интеграции разработки и операционной деятельности для обеспечения их надежности и стабильности. DevOps-подход помогает снять границы между разработкой и эксплуатацией, что способствует снижению рисков и повышению стабильности системы. [2]

Наконец, высокие требования пользователей к отзывчивости и частым обновлениям ПО делают необходимым быстрое и надежное внедрение изменений. DevOps-технологии позволяют автоматизировать процессы сборки, настройки и развертывания, что сокращает время и риски при внедрении новых функций и исправлении ошибок.

В заключение главы отмечается, что внедрение DevOps-технологий является неотъемлемой необходимостью для современных организаций разра-

ботки ПО. Результаты и рекомендации данной работы могут быть использованы для успешной реализации и применения DevOps-технологий, а также для улучшения конкурентоспособности организаций на рынке ПО.

Во второй главе "Методология DevOps" рассматривается сама методология DevOps, ее принципы и ключевые компоненты.

Методология представляет собой подход, объединяющий разработку (Development) и операционную деятельность (Operations) в единый процесс, направленный на достижение высокой производительности и качества в разработке, тестировании и доставке ПО. [3]

Главная цель методологии DevOps заключается в снятии границ и создании тесной интеграции между командами разработчиков и операционных специалистов. Это позволяет обеспечить более гладкое и эффективное взаимодействие между ними, ускорить процессы разработки и доставки ПО, а также повысить стабильность и надежность программных систем.

Основные принципы методологии DevOps включают автоматизацию, контроль версий, непрерывную интеграцию и развертывание, контейнеризацию и оркестрацию, а также непрерывную доставку и мониторинг. Автоматизация играет ключевую роль в DevOps, позволяя устранить рутинные и повторяющиеся задачи, минимизировать ошибки и сократить время, необходимое для разработки и доставки ПО.

Важной составляющей DevOps является контроль версий, который обеспечивает отслеживание изменений в коде и совместную работу над ним. Непрерывная интеграция и развертывание позволяют автоматически собирать, тестировать и развертывать ПО, обеспечивая быструю обратную связь и быстрое внедрение изменений. Контейнеризация и оркестрация позволяют упаковывать приложения и их зависимости в контейнеры для более эффективной доставки и масштабирования. Непрерывная доставка и мониторинг обеспечивают постоянный контроль над работой системы и быструю реакцию на изменения. [4]

В заключение главы подчеркивается, что методология DevOps представляет собой мощный инструмент для повышения производительности и качества в разработке программного обеспечения. Ее основные принципы и компоненты позволяют организациям достигать более гибкой и отзывчивой разработки, ускорять доставку новых функций и обеспечивать более стабиль-

ное и надежное функционирование программных систем.

В третьей главе "Реализация DevOps с использованием непрерывной интеграции и доставки" рассматривается роль и значение непрерывной интеграции (CI) и непрерывной доставки (CD) в контексте внедрения методологии DevOps.

Непрерывная интеграция (Continuous Integration) - это практика, которая заключается в автоматическом и регулярном слиянии изменений в коде, выполнении сборки и запуске автоматических тестов. Она способствует быстрой интеграции кода разработчиков, обнаружению ошибок и конфликтов на ранних этапах, а также обеспечивает постоянную обратную связь разработчикам.

Непрерывная доставка (Continuous Delivery) - это практика, которая предполагает автоматизированное развертывание приложения в любой среде, начиная от тестовой и заканчивая рабочей средой. Она позволяет организации быстро и безопасно доставлять изменения в продукцию, снижая риски и время, связанные с ручными процессами развертывания.

В главе рассматриваются основные принципы и компоненты непрерывной интеграции и доставки. Важными составляющими CI/CD являются автоматизированные сборки, автоматическое тестирование, контроль версий и инфраструктура как код.

Автоматизированные сборки позволяют автоматически собирать приложение из исходного кода, устанавливать зависимости и создавать исполняемый файл. Это позволяет упростить и ускорить процесс сборки и гарантировать, что приложение будет собрано с последними изменениями. [5]

Автоматическое тестирование включает в себя запуск различных видов тестов, таких как модульные, интеграционные и функциональные тесты. Оно позволяет автоматически проверять работоспособность и качество приложения, выявлять возможные ошибки и проблемы, а также обеспечивать стабильность при каждом изменении в коде.

Контроль версий является важной составляющей CI/CD, поскольку он обеспечивает отслеживание изменений в коде и возможность работать с разными версиями приложения. Это позволяет разработчикам совместно работать над проектом, вносить изменения и откатываться к предыдущим версиям в случае необходимости.

Инфраструктура как код представляет собой практику описания инфраструктуры и конфигурации среды развертывания в виде кода. Это позволяет автоматизировать процесс создания и настройки необходимой инфраструктуры, что способствует повторяемости, масштабируемости и надежности процесса развертывания.

В заключение главы подчеркивается, что непрерывная интеграция и доставка играют ключевую роль в успешной реализации методологии DevOps. Они позволяют организациям достигать высокой производительности, качества и отзывчивости в разработке и доставке программного обеспечения.

В четвертой главе "Реализация конвейера автоматической доставки кода" рассматривается процесс разработки и внедрения конвейера автоматической доставки кода в рамках методологии DevOps.

Конвейер автоматической доставки (CI/CD pipeline) является основным инструментом для автоматизации и управления процессом доставки ПО. Он включает в себя серию этапов и задач, выполняемых автоматически, начиная с момента внесения изменений в код до его развертывания и выпуска в продукцию.

Главная цель конвейера автоматической доставки состоит в создании непрерывного и надежного процесса доставки ПО, который позволяет быстро и безопасно внедрять новые функции и исправления в рабочую среду.

В главе рассматриваются основные этапы и компоненты конвейера автоматической доставки. Это включает этапы сборки, тестирования, развертывания и мониторинга.

Этап сборки представляет собой процесс сборки и упаковки кода в исполняемые компоненты или контейнеры. Он включает в себя компиляцию исходного кода, установку зависимостей и создание исполняемых файлов или образов. [6]

Этап тестирования включает запуск различных видов тестов для проверки работоспособности и качества ПО. Это может включать модульное тестирование, интеграционное тестирование, функциональное тестирование и другие виды тестирования. Цель этого этапа - обнаружить и исправить ошибки или проблемы в ПО на ранних этапах разработки.

Этап развертывания представляет собой процесс автоматического развертывания ПО в целевую среду, будь то тестовая среда, рабочая среда или

облачная инфраструктура. Здесь происходит автоматическое развертывание и настройка приложения в соответствии с требуемыми конфигурациями. [7]

Этап мониторинга обеспечивает постоянный контроль за работой развернутой системы, сбор и анализ данных о производительности, доступности и стабильности. Это позволяет быстро обнаруживать и реагировать на проблемы, связанные с развернутым приложением.

В заключение главы подчеркивается, что конвейер автоматической доставки является критическим элементом реализации методологии DevOps. Он позволяет организациям ускорить и автоматизировать процесс доставки ПО, обеспечивая гибкость, надежность и высокое качество разработки.

ЗАКЛЮЧЕНИЕ

В данном автореферате была рассмотрена актуальная тема автоматизации технологических процессов сборки, настройки и развертывания программного обеспечения в контексте современной разработки и внедрения методологии DevOps.

Целью работы было исследование и описание принципов, методов и инструментов автоматизации этих процессов, а также анализ их роли и значимости для повышения эффективности и качества разработки ПО.

В целом, исследование показало, что автоматизация технологических процессов сборки, настройки и развертывания программного обеспечения играет важную роль в современной разработке ПО. Она позволяет сократить время и ресурсы, повысить качество и надежность разработки, а также обеспечить более гибкую и эффективную работу команды.

Таким образом, автоматизация технологических процессов является неотъемлемой частью методологии DevOps и способствует достижению высокой производительности, качества и отзывчивости в разработке и доставке программного обеспечения. Дальнейшее исследование и практическое внедрение этих принципов и инструментов позволит организациям оставаться конкурентоспособными и успешными на рынке разработки ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 F. Millstein, DevOps Handbook: What Is DevOps, Why You Need It And How To Transform Your Business With DevOps Practices. Frank Millstein, 2020.
- 2 Kim, G., et al. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, 2016
- 3 Hering, Mirco. DevOps for the Modern Enterprise: Winning Practices to Transform Legacy IT Organizations. United States, IT Revolution Press, 2018.
- 4 Humble, Jez, and Farley, David. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. United Kingdom, Pearson Education, 2010.
- 5 Duvall, Paul M., et al. Continuous Integration: Improving Software Quality and Reducing Risk. United Kingdom, Pearson Education, 2007.
- 6 Geerling, Jeff. Ansible for DevOps: Server and Configuration Management for Humans. United States, Midwestern Mac, LLC, 2020.
- 7 Red Hat. (n.d.). What is orchestration? [Электронный ресурс], URL <https://www.redhat.com/en/topics/automation/what-is-orchestration> (дата обращения 25.05.2023)