

УДК 004.82

Коптенко Елизавета Викторовна
старший преподаватель кафедры «Информатика и программное
обеспечение»

Брянский Государственный Технический Университет
Россия, г. Брянск

Титарев Валентин Дмитриевич
Студент

Брянский Государственный Технический Университет
Россия, г. Брянск

СРАВНИТЕЛЬНЫЙ АНАЛИЗ DEVOPS ТЕХНОЛОГИЙ

Аннотация. В данной статье рассматривается применение подхода DevOps в разработке программного обеспечения. Автор анализирует различные инструменты и практики DevOps, включая контейнеризацию с помощью Docker, оркестрацию контейнеров с использованием Kubernetes, автоматизацию процессов с помощью Jenkins и GitLab CI/CD, а также интегрированные решения Azure DevOps. В статье анализируются современные подходы к использованию этих инструментов в DevOps и описываются перспективы их развития и внедрения в различных отраслях.

Annotation. This article discusses the application of the DevOps approach in software development. The author examines various DevOps tools and practices, including containerization using Docker, container orchestration using Kubernetes, process automation using Jenkins and GitLab CI/CD, and integrated Azure DevOps solutions. The article analyzes modern approaches to the use of these tools in DevOps and describes the prospects for their development and implementation in various industries.

Ключевые слова: DevOps, Docker, Kubernetes, Jenkins, GitLab CI/CD, Azure DevOps, контейнеризация, оркестрация, автоматизация, интеграция, разработка программного обеспечения.

Keywords: DevOps, Docker, Kubernetes, Jenkins, GitLab CI/CD, Azure DevOps, containerization, orchestration, automation, integration, software development.

DevOps – это подход к разработке программного обеспечения, который ставит своей целью упрощение и ускорение процесса доставки программных продуктов. Он использует автоматизированные процессы для быстрого развертывания и обновления программного обеспечения, что позволяет организациям быстрее реагировать на изменения в требованиях и улучшать свои программные продукты и услуги.

Одним из ключевых аспектов DevOps является совместная работа между разработчиками и операционными командами. DevOps стирает границы между этими командами, обеспечивая более тесное сотрудничество и обмен информацией. Это улучшает процесс разработки и снижает вероятность ошибок.

DevOps также внедряет инструменты и практики для автоматизации рутинных задач, таких как тестирование и развертывание. Это повышает производительность и сокращает время на разработку. Непрерывная интеграция и непрерывная доставка (CI/CD) – это ключевой компонент DevOps, который обеспечивает постоянное и быстрое обновление программного обеспечения без прерывания работы сервиса.

DevOps включает множество задач, таких как управление версиями, автоматизация тестирования, развертывание и мониторинг. Для каждой из этих задач существуют специализированные продукты, которые облегчают их выполнение:

Управление версиями: Использование Git для отслеживания изменений и совместной работы над исходным кодом.

Git использует систему контроля версий, которая позволяет разработчикам управлять и отслеживать изменения в исходном коде проекта. Git обеспечивает эффективное управление большими проектами и поддерживает непрерывную интеграцию. Git является основой для многих веб-сервисов, таких как GitHub, который предоставляет инструменты для хранения кода, совместной работы и отслеживания изменений.

Автоматизация тестирования: Jenkins или GitLab CI/CD для автоматического выполнения тестов при каждом изменении кода.

Jenkins и GitLab CI/CD – это инструменты автоматизации, которые играют ключевую роль в практиках DevOps, особенно в автоматизации тестирования. Они обеспечивают автоматическое выполнение тестов при каждом изменении кода, что ускоряет процесс разработки и помогает обнаруживать и исправлять ошибки на ранних стадиях.

Jenkins – это открытый сервер автоматизации, который позволяет автоматизировать различные стадии разработки программного обеспечения. Jenkins может быть настроен на выполнение тестов каждый раз, когда разработчик вносит изменения в код, обеспечивая непрерывную интеграцию.

GitLab CI/CD – это часть GitLab, которая предоставляет инструменты для непрерывной интеграции, доставки и развертывания. GitLab CI/CD автоматически запускает тесты на каждый новый коммит, который добавляется в репозиторий в GitLab, и предоставляет обратную связь разработчикам о состоянии этих тестов, что позволяет быстро обнаруживать и исправлять ошибки.

Развертывание: Docker для контейнеризации приложений и Kubernetes для управления контейнеризированными приложениями.

Docker – это платформа, предназначенная для разработки, доставки и запуска приложений с использованием контейнеров. Docker позволяет разработчикам упаковывать приложения вместе со всеми их зависимостями в контейнеры, что обеспечивает консистентность во всех средах. Docker облегчает процесс развертывания приложений, делая его более эффективным и безопасным. Он не только облегчает процесс развертывания приложений, но и способствует стандартизации окружений, что в свою очередь уменьшает вероятность ошибок, связанных с различиями в инфраструктуре. Это делает Docker особенно полезным в мире DevOps, где быстрота и надежность являются ключевыми. Docker использует клиент-серверную архитектуру (см. рис.1). Docker-клиент взаимодействует с Docker-демоном, который отвечает за создание, запуск и распределение Docker-контейнеров. Docker позволяет разработчикам упаковывать приложения вместе со всеми их зависимостями в контейнеры, что обеспечивает консистентность во всех средах. Docker облегчает процесс развертывания приложений, делая его более эффективным и безопасным. Он способствует стандартизации окружений, что уменьшает вероятность ошибок, связанных с различиями в инфраструктуре.

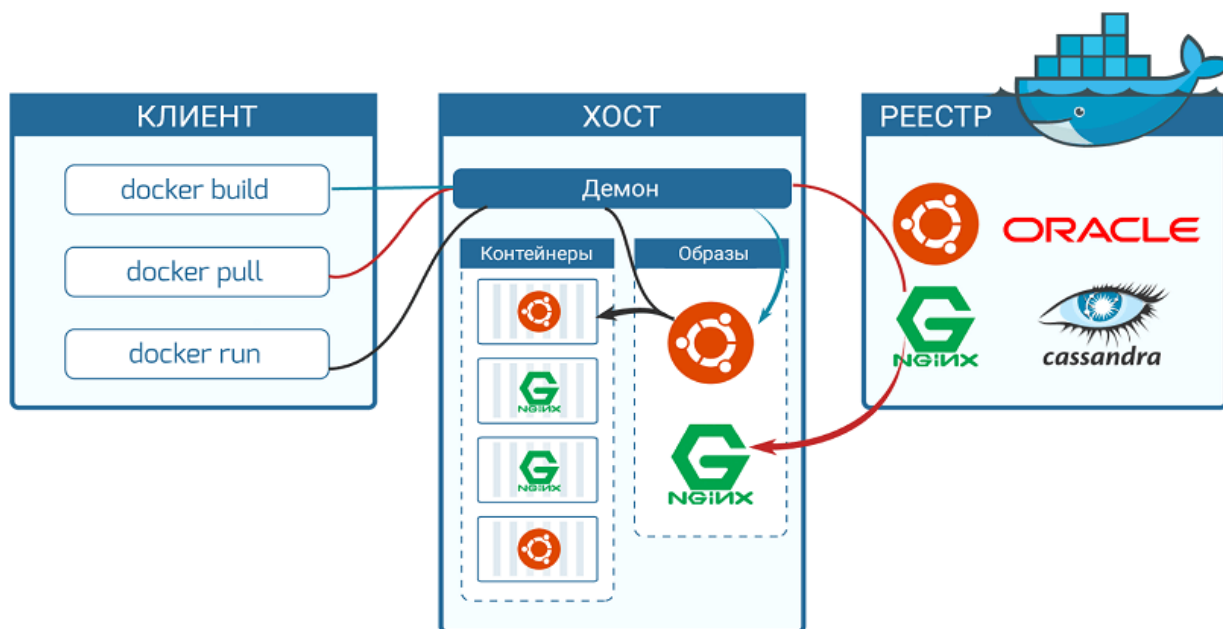


Рис. 1. Архитектура Docker

Kubernetes (часто сокращается до K8s) — это открытая система для автоматизации развертывания, масштабирования и управления контейнеризованными приложениями. Kubernetes группирует контейнеры, составляющие приложение, в логические единицы для упрощения управления и обнаружения. Kubernetes разработан на основе 15-летнего опыта Google в области работы с рабочими нагрузками в производственных условиях, а также лучших идей и практик сообщества. Kubernetes использует архитектуру (см. рис. 2), основанную на контроллере и рабочих узлах. Контроллер управляет кластером Kubernetes, а рабочие узлы запускают приложения в контейнерах. Kubernetes также использует API-сервер для обработки внутренних и внешних запросов.

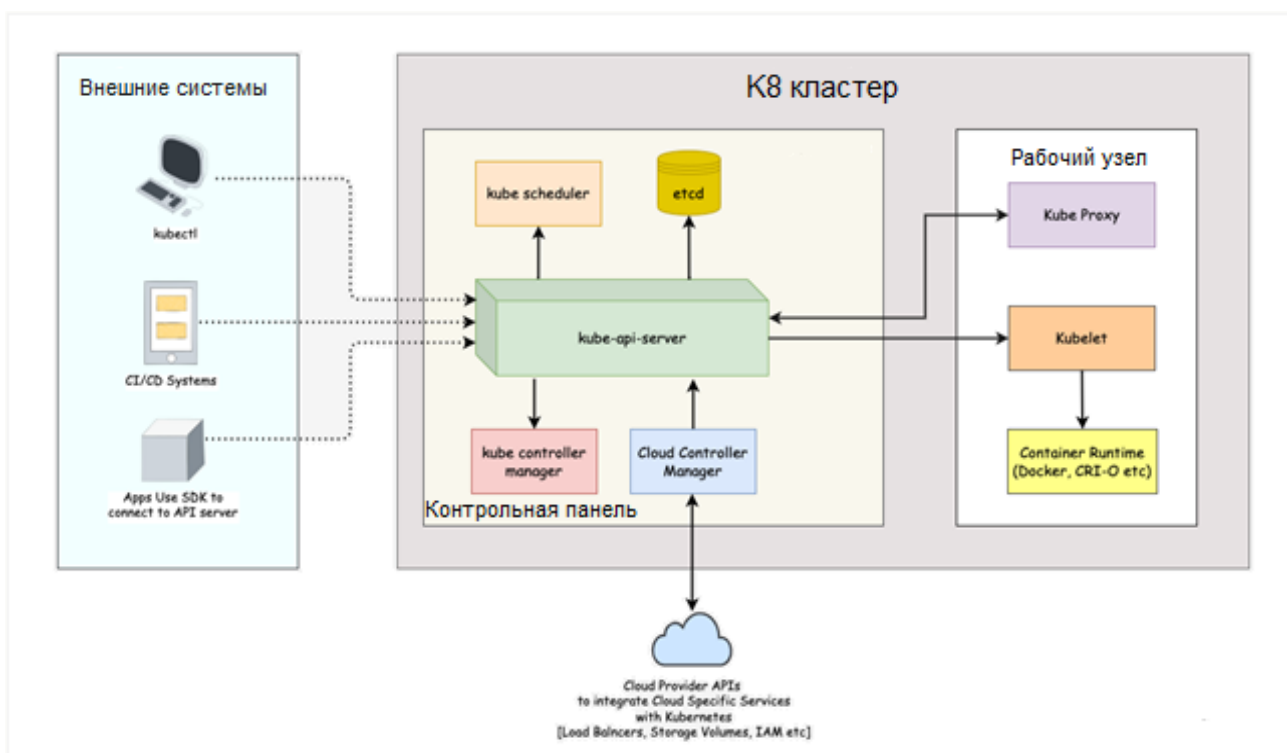


Рис. 2. Архитектура Kubernetes

Мониторинг: Prometheus и Grafana для сбора и визуализации метрик производительности.

Azure DevOps от Microsoft – это комплексный инструментарий, который предоставляет набор средств для управления жизненным циклом разработки

программного обеспечения. Он включает систему контроля версий Git, систему отслеживания ошибок, средства CI/CD, планирование проектов и управление задачами, а также множество других инструментов. В отличие от отдельных инструментов, Azure DevOps обеспечивает единое и интегрированное решение, которое позволяет командам разработки сотрудничать, управлять и отслеживать весь процесс разработки от идеи до выпуска продукта. Azure предлагает обширный набор инструментов и сервисов для разработки, развертывания и управления приложениями. Azure поддерживает различные архитектуры и технологии, включая микросервисы, серверные приложения, базы данных, AI и машинное обучение, DevOps и многое другое. Azure также предлагает решения для различных сценариев и рабочих нагрузок.

Для анализа текущих тенденций в области DevOps был проведен опрос среди различных компаний с целью определить, какие инструменты DevOps наиболее часто используются. Результаты показывают, что Docker является самым популярным инструментом, используемым в 120 компаниях. Это неудивительно, поскольку Docker предоставляет удобные возможности для контейнеризации приложений, упрощая их развертывание и масштабирование.

Jenkins также занимает значительное место среди компаний: 110 компаний используют этот инструмент. Jenkins является одним из самых популярных инструментов для автоматизации сборок и развертываний, что делает его ключевым элементом в инфраструктуре DevOps.

GitLab CI/CD используется в 85 компаниях, предоставляя комплексное решение для непрерывной интеграции и доставки (CI/CD). Этот инструмент позволяет автоматизировать процесс развертывания приложений и управления версиями, что делает его популярным выбором среди DevOps-команд.

Kubernetes широко используется для оркестрации контейнеров, с количеством пользователей 95 компаний. Он обеспечивает надежное управление

и масштабирование контейнеризованных приложений, что делает его важным компонентом в современной DevOps-практике.

Кроме того, Azure DevOps используется в 70 компаниях. Azure DevOps предлагает интегрированные инструменты для управления проектами, репозиториями кода, тестирования и развертывания, что позволяет командам эффективно сотрудничать и автоматизировать процессы разработки и доставки программного обеспечения.

Таким образом, данные показывают, что Docker, Jenkins, GitLab CI/CD и Kubernetes являются одними из самых популярных инструментов DevOps, широко используемыми в индустрии. Компании стремятся к использованию контейнеризации и оркестрации для упрощения развертывания приложений и управления инфраструктурой. Azure DevOps также играет важную роль в современных DevOps практиках, предоставляя комплексные решения для управления и автоматизации процессов разработки.

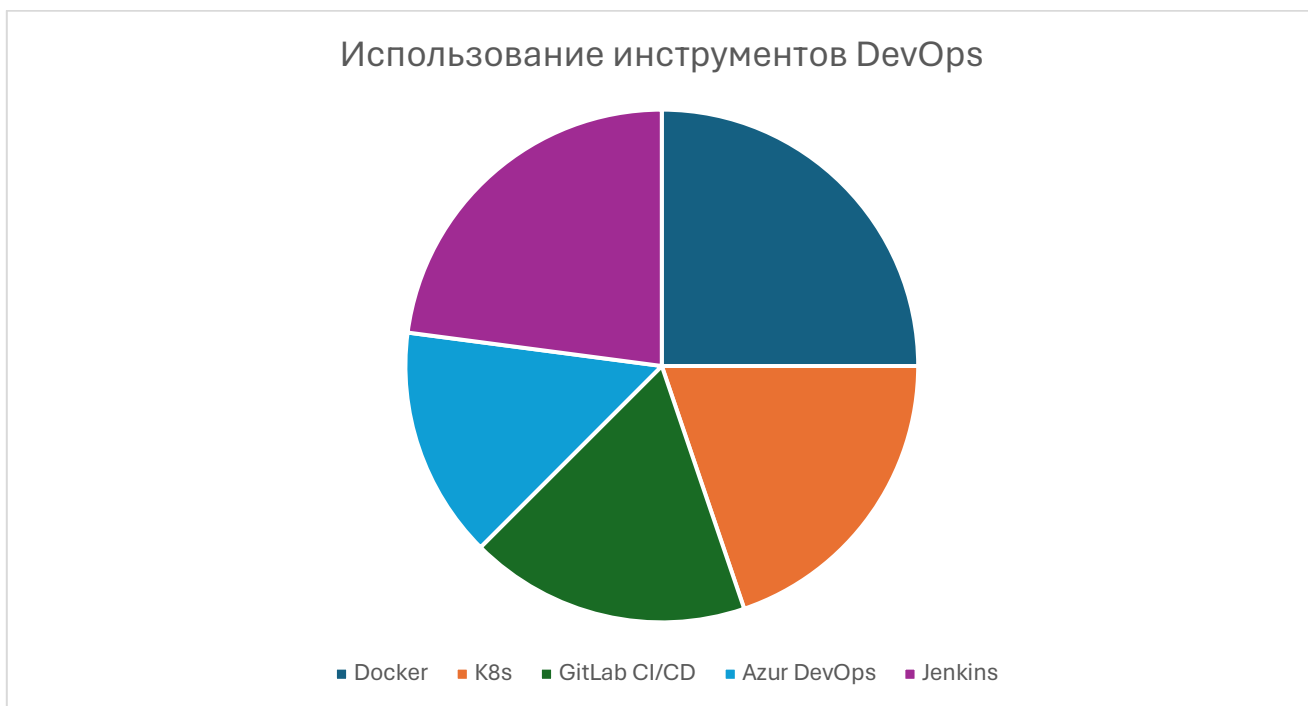


Рис. 3. Использование инструментов DevOps

Все эти инструменты вместе формируют мощную экосистему для поддержки DevOps-практик. Они обеспечивают автоматизацию, сотрудничество,

управление версиями и масштабируемость, которые являются ключевыми для эффективной и надежной разработки программного обеспечения. Однако выбор конкретных инструментов должен основываться на конкретных потребностях и целях команды или организации. Каждый инструмент имеет свои сильные и слабые стороны, и важно провести тщательное исследование перед принятием решения.

Литература

1. «State of DevOps Report 2023». Puppet Labs. / [Электронный ресурс] – Режим доступа: <https://puppet.com/resources/report/state-of-devops-report/>
2. «Docker Hub Quickstart». Docker Documentation. / [Электронный ресурс] – Режим доступа: <https://docs.docker.com/docker-hub/>
3. «Developer Survey Results 2023». Stack Overflow. / [Электронный ресурс] – Режим доступа: <https://insights.stackoverflow.com/survey/2023>
4. «CNCF Survey 2023». Cloud Native Computing Foundation. / [Электронный ресурс] – Режим доступа: <https://www.cncf.io/reports/cncf-survey-2023/>

Literature

1. "State of DevOps Report 2023." Puppet Labs. / [Electronic resource] – Access mode: <https://puppet.com/resources/report/state-of-devops-report/>
2. "Docker Hub Quickstart". Docker Documentation. / [Electronic resource] – Access mode: <https://docs.docker.com/docker-hub/>
3. "Developer Survey Results 2023." Stack Overflow. / [Electronic resource] – Access mode: <https://insights.stackoverflow.com/survey/2023>
4. "CNCF Survey 2023". Cloud Native Computing Foundation. / [Electronic resource] – Access mode: <https://www.cncf.io/reports/cncf-survey-2023/>