

ОБ АВТОРЕ

Баланов Антон Николаевич имеет большой опыт руководства и консультирования в сфере ИТ-технологий. Работал топ-менеджером в крупных компаниях — таких, как Industrial and Commercial Bank of China (КНР), Caravan portal (ОАЭ), Банк ВТБ, Сбербанк России, VK; руководил разработками сервиса Gosuslugi.ru. Имеет степень MBA IT (CIA) и сертификации Microsoft, CompTIA, ISACA, PMI, SHRM, ПВА, HRCI, ISO, Six Sigma (Master Black Belt). Преподавал в следующих вузах и учебных центрах: Российском университете дружбы народов, СберУниверситете, Институте бизнеса и делового администрирования и Центре подготовки руководителей и команд цифровой трансформации (на базе Высшей школы государственного управления РАНХиГС). Автор десятков книг и научно-практических публикаций в профессиональных изданиях. Является советником Российской академии естественных наук.

Широкая эрудиция и глубокие профессиональные компетенции автора в сфере ИТ-технологий позволили ему создать книжную серию «Айтишный университет», один из выпусков которой находится перед вами.

ОГЛАВЛЕНИЕ

Глава 1. Введение в DevOps	11
Введение	11
Определение DevOps и его цель.....	12
Значение DevOps в современной разработке программного обеспечения.....	14
Преимущества и вызовы, связанные с применением DevOps	17
Заключение	19
 Глава 2. Основные принципы и инструменты DevOps	20
Введение	20
Основные принципы DevOps, такие как непрерывная интеграция, непрерывная доставка и непрерывное развертывание.....	21
Распространенные инструменты и технологии, используемые в DevOps практиках	24
Автоматизация процессов разработки, тестирования и доставки с использованием инструментов DevOps	26
Заключение	28
 Глава 3. Внедрение DevOps в вашей команде	30
Введение	30
Шаги и рекомендации по внедрению DevOps в организации	31

Формирование культуры и командной работы, способствующей успешному применению DevOps	34
Обучение и развитие сотрудников для работы в DevOps-ориентированной среде	36
Заключение	38
Глава 4. Примеры успешного внедрения DevOps	40
Введение	40
Рассмотрение реальных примеров компаний, которые успешно применяют DevOps	41
Изучение их опыта, вызовов и достижений при использовании DevOps	42
Уроки, извлеченные из этих примеров, и рекомендации для других организаций и команд	44
Заключение	46
Глава 5. Управление конфигурацией и инфраструктурой	48
Введение	48
Использование инструментов для управления конфигурацией и автоматизации управления инфраструктурой	49
Создание и развертывание инфраструктуры как кода (Infrastructure as Code)	53
Обеспечение надежности, масштабируемости и гибкости в управлении конфигурацией и инфраструктурой	55
Заключение	57
Глава 6. Обеспечение безопасности и надежности в DevOps	59
Введение	59
Вопросы безопасности и надежности в DevOps-среде	60
Внедрение и автоматизация процессов	

безопасности, включая контроль доступа, мониторинг и обнаружение угроз	63
Обеспечение непрерывности работы и восстановления после сбоев в DevOps-процессах	64
Заключение	66
 Глава 7. Мониторинг и отладка в DevOps	 68
Введение	68
Разработка и применение эффективных методов мониторинга и отладки приложений	69
Использование инструментов для мониторинга производительности, доступности и управления ошибками	72
Анализ данных мониторинга для оптимизации производительности и качества приложения	74
Заключение	76
 Глава 8. Автоматизация тестирования в DevOps	 78
Введение	78
Виды тестирования, включая модульное тестирование, интеграционное тестирование и автоматизированное функциональное тестирование	79
Автоматизация процессов тестирования с использованием инструментов и практик DevOps	82
Управление наборами тестов и обеспечение высокого качества продукта в DevOps-циклах разработки	85
Заключение	87
 Глава 9. Управление изменениями и версионностью	 89
Введение	89
Управление изменениями в DevOps-процессах, включая контроль версий кода и управление конфигураций	90
Стратегии и инструменты для эффективного	

управления изменениями и версиями в разработке и доставке приложений.....	93
Организация и координация работы разработчиков, тестировщиков и операционных команд при внесении изменений в продукт	95
Заключение	99
 Глава 10. Будущее DevOps и инновации	101
Введение	101
Тенденции и перспективы развития DevOps	102
Влияние новых технологий, таких как искусственный интеллект, машинное обучение и автоматизация процессов, на развитие DevOps.....	103
Ожидаемые вызовы и возможности для DevOps в будущем.....	105
Заключение	107

ГЛАВА 1

ВВЕДЕНИЕ В DEVOPS

ВВЕДЕНИЕ

Глава 1 посвящена введению в DevOps — философии, методологии и набору практик, объединяющих разработку и операции в области разработки программного обеспечения. Мы рассмотрим определение DevOps и его цель, а также роль и значение DevOps в современной разработке программного обеспечения. Мы также обратим внимание на преимущества, которые могут быть получены с применением DevOps, а также вызовы и сложности, связанные с его внедрением и использованием.

В первой части главы мы определим понятие DevOps. DevOps — это методология и набор практик, направленных на объединение разработки (Development) и операций (Operations) в целях создания более эффективного и гибкого процесса разработки программного обеспечения. Он стремится к достижению высокой скорости развертывания, качества и надежности программных продуктов.

Затем мы рассмотрим значимость DevOps в современной разработке программного обеспечения. DevOps помогает сократить время цикла разработки, улучшить качество продукта и улучшить коммуникацию и сотрудничество между разработчиками и операционными командами. Это позволяет достичь более быстрой и надежной поставки программного обеспечения, улучшить удовлетворенность клиентов и конкурентоспособность компании на рынке.

В следующей части главы мы обратим внимание на преимущества и вызовы, связанные с применением DevOps. Преимущества DevOps включают улучшенную скорость разработки и доставки, повышенную надежность и стабильность системы,

автоматизацию процессов, а также более эффективное использование ресурсов и сокращение затрат. Однако, применение DevOps может также столкнуться с вызовами, такими как изменение организационной культуры, необходимость внедрения новых инструментов и процессов, и необходимость эффективного сотрудничества между различными командами и дисциплинами.

Изучение DevOps поможет нам понять его значимость и преимущества, а также вызовы, с которыми мы можем столкнуться при его внедрении. DevOps предлагает новый подход к разработке программного обеспечения, который способствует более эффективному и надежному процессу разработки, улучшает качество продукта и повышает удовлетворенность клиентов.

ОПРЕДЕЛЕНИЕ DEVOPS И ЕГО ЦЕЛЬ

DevOps — это методология разработки программного обеспечения, объединяющая процессы разработки и операционной эксплуатации (IT Operations). Она стремится к сокращению разрыва между разработчиками и операционной командой, чтобы создать более эффективный и автоматизированный процесс разработки, тестирования, доставки и поддержки приложений.

Цель *DevOps* заключается в улучшении сотрудничества и коммуникации между разработчиками, тестировщиками и операционной командой, а также в автоматизации и оптимизации процессов разработки и эксплуатации приложений. Основные принципы и практики *DevOps* включают непрерывную интеграцию (Continuous Integration), непрерывное развертывание (Continuous Deployment), автоматизацию управления конфигурацией и инфраструктурой (Infrastructure as Code), мониторинг и логирование приложений (см. Табл. 1.1).

Пример. Компания, разрабатывающая веб-приложение, реализовала *DevOps*-подход для улучшения своих процессов разработки и эксплуатации. Они использовали инструменты непрерывной интеграции, такие как Jenkins, для автоматиче-

Таблица 1.1

Основные практики и инструменты DevOps

<i>Практика</i>	<i>Описание</i>
Continuous Integration	Автоматическое интегрирование изменений в основную кодовую базу
Continuous Deployment	Автоматическое развертывание приложения после успешной интеграции
Infrastructure as Code	Описывание и управление инфраструктурой с использованием кода
Мониторинг и логирование	Сбор и анализ данных о работе приложения для обеспечения его надежности
Автоматизация тестирования	Автоматизация процесса тестирования приложения для быстрой обратной связи
Контейнеризация	Использование контейнеров для упаковки и развертывания приложений
Оркестрация контейнеров	Управление и масштабирование контейнеризованных приложений

ской сборки и тестирования кода при каждом коммите. После успешной интеграции, приложение автоматически развертывается на серверах с помощью инструмента для управления конфигурацией Ansible.

Они также использовали контейнеризацию с Docker и оркестрацию с Kubernetes для упаковки и масштабирования своих сервисов. Компания отметила значительное улучшение скорости и надежности своего процесса разработки и эксплуатации благодаря внедрению DevOps-подхода.

DevOps — это методология, которая объединяет процессы разработки и операционной эксплуатации для создания эффективного и автоматизированного процесса разработки, тестирования, доставки и поддержки приложений. Она стремится к улучшению сотрудничества и коммуникации между различными командами, а также к автоматизации и оптимизации процессов. Цель DevOps заключается в достижении более бы-

строй и надежной разработки приложений, обеспечении частых итераций, автоматизации процессов развертывания и управления инфраструктурой. Применение DevOps практик и инструментов позволяет организациям достигать более высокой производительности, сокращать время доставки продукта на рынок и повышать уровень надежности приложений.

ЗНАЧЕНИЕ DEVOPS В СОВРЕМЕННОЙ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Значение DevOps в современной разработке программного обеспечения сегодня невозможно переоценить. DevOps — это подход к разработке и доставке программного обеспечения, который объединяет разработчиков и операционную команду в единую команду с целью улучшения сотрудничества, эффективности и качества процесса разработки и доставки ПО. DevOps уделяет особое внимание автоматизации процессов, непрерывной интеграции и доставке, мониторингу и обратной связи для обеспечения более быстрой, стабильной и инновационной разработки.

Ниже приведена таблица, иллюстрирующая ключевые аспекты и преимущества DevOps в разработке программного обеспечения.

Таблица 1.2

<i>Аспект</i>	<i>Описание</i>
Культура	Внедрение культуры сотрудничества, коммуникации и общей ответственности между разработчиками и операционной командой.
Автоматизация процессов	Использование инструментов автоматизации для упрощения процессов разработки, тестирования и доставки ПО.
Непрерывная интеграция	Автоматическая интеграция изменений в код в общую кодовую базу для обеспечения более регулярного и стабильного процесса разработки.

<i>Аспект</i>	<i>Описание</i>
Непрерывная доставка	Автоматическая доставка ПО в производственную среду после успешной интеграции и проверки кода, что позволяет доставлять изменения быстрее и безопаснее.
Мониторинг	Реализация системы мониторинга для наблюдения за работоспособностью и производительностью ПО и быстрого выявления проблем.
Обратная связь	Обеспечение своевременной обратной связи между командами разработчиков и операционной командой для улучшения процесса разработки и доставки ПО.

Пример. Рассмотрим пример использования DevOps в разработке веб-приложения. Команда разработчиков и операционная команда работают совместно, используя инструменты автоматизации процессов, такие как система контроля версий Git, система автоматической сборки и развертывания Jenkins, и инструменты управления конфигурацией, такие как Ansible.

Команда разработчиков регулярно коммитит изменения в Git-репозиторий. Система непрерывной интеграции, настроенная на базе Jenkins, автоматически собирает и тестирует код после каждого коммита. Если тесты успешно пройдены, происходит автоматическое развертывание приложения на тестовом сервере.

После тестирования команда разработчиков проводит регулярные код-ревью и исправляет возможные ошибки. Когда команда удовлетворена результатом, происходит автоматическое развертывание приложения на производственном сервере с помощью Ansible.

Всюду применяется мониторинг для отслеживания производительности и доступности приложения. Если возникают проблемы, команда операционной команды быстро реагирует и принимает меры для исправления ситуации.

DevOps является неотъемлемой частью современной разработки программного обеспечения. Он позволяет создавать и

доставлять ПО более быстро, стабильно и эффективно. Значительные выгоды, которые может принести DevOps, включают повышение качества ПО, снижение времени цикла разработки, более эффективное использование ресурсов, улучшение коммуникации и сотрудничества между различными командами, а также улучшение обратной связи и реагирования на изменения и проблемы.

DevOps также способствует созданию более надежных и гибких систем, которые могут легко масштабироваться и адаптироваться к изменяющимся требованиям и условиям. Он способствует автоматизации рутинных задач, упрощению процессов развертывания и доставки ПО, а также повышению степени автоматизации тестирования и мониторинга.

Существует множество компаний, которые успешно применяют DevOps (Development and Operations) подход для улучшения скорости разработки, качества продукта и эффективности операций. Вот примеры таких компаний.

1. *Amazon*. Amazon является одним из основоположников DevOps и успешно применяет этот подход для своих облачных услуг (Amazon Web Services — AWS) и электронной коммерции. DevOps позволяет им автоматизировать развертывание и масштабирование инфраструктуры, обеспечивать высокую доступность и быструю доставку обновлений.

2. *Netflix*. Netflix также известен своим успешным применением DevOps. Они используют автоматизацию процессов и контейнеризацию для развертывания своих сервисов и быстрой доставки обновлений. Netflix активно применяет мониторинг и логирование для обеспечения высокой доступности и устранения проблем в реальном времени.

3. *Google*. Google является примером компании, которая успешно использует DevOps подход для своих облачных услуг (Google Cloud Platform) и разработки программного обеспечения. Они активно применяют автоматизацию процессов, контейнеризацию и инфраструктуру как код для быстрого развертывания и масштабирования своих приложений.

4. *Газпромбанк*. Газпромбанк является компанией, которая также применяет DevOps для своих сервисов социальных сетей

и инфраструктуры. Они используют автоматическое развертывание и инструменты непрерывной интеграции и доставки (CI/CD) для обеспечения быстрой доставки обновлений и обратной связи между разработчиками и операционной командой.

5. *Etsy*. Etsy, онлайн-платформа для продажи и покупки ручных изделий, является примером компании, которая успешно применяет DevOps. Они активно используют инфраструктуру как код, автоматизацию процессов и непрерывную доставку для быстрой и надежной доставки своих приложений.

6. *Spotify*. Spotify, музыкальная платформа для потокового воспроизведения, является еще одним примером компании, успешно применяющей DevOps. Они активно применяют микросервисную архитектуру, автоматизацию процессов и CI/CD для быстрой доставки новых функций и обновлений для своих миллионов пользователей.

В целом, DevOps играет важную роль в современной разработке программного обеспечения, обеспечивая быстрое развертывание, стабильность и высокое качество ПО. Он способствует сокращению времени доставки ПО, автоматизации процессов и улучшению коммуникации и сотрудничества внутри команды разработчиков и операционной команды.

ПРЕИМУЩЕСТВА И ВЫЗОВЫ, СВЯЗАННЫЕ С ПРИМЕНЕНИЕМ DEVOPS

DevOps — это подход к разработке программного обеспечения, который объединяет разработку (Development) и эксплуатацию (Operations) в единое целое. Он направлен на улучшение сотрудничества и коммуникации между разработчиками, тестировщиками и операционными специалистами для достижения быстрой и надежной поставки программного обеспечения. Применение DevOps в организации может принести множество преимуществ, но также может встретить вызовы, которые требуют внимания и решения (См. Табл. 1.3).

Пример. Одним из преимуществ DevOps является быстрая поставка программного обеспечения. Благодаря автоматизации и стандартизации процессов разработки, тестирования

Таблица 1.3

**Преимущества и вызовы, связанные
с применением DevOps**

Преимущества	Вызовы
Быстрая поставка программного обеспечения	Сложность внедрения DevOps в сложные и устаревшие инфраструктуры
Улучшенное качество продукта	Культурные и организационные изменения
Улучшенная коммуникация и сотрудничество	Отсутствие согласованности и стандартизации процессов
Автоматизация и стандартизация процессов	Сложности в обеспечении безопасности и соответствия требованиям
Более эффективное использование ресурсов	Недостаток навыков и знаний по DevOps

и развертывания, команда может выпускать новые версии приложения на регулярной основе. Например, компания XYZ внедрила DevOps и перешла от ежеквартальных релизов к релизам каждые две недели. Это позволило им оперативно внедрять новые функции и исправления ошибок, что привело к повышению уровня удовлетворенности клиентов и конкурентоспособности компании.

Однако внедрение DevOps может столкнуться с вызовами. Например, устаревшая и сложная инфраструктура может потребовать дополнительных усилий для автоматизации и оркестрации процессов. Культурные и организационные изменения также могут быть сложными, поскольку требуется изменить способ работы и взаимодействия различных команд. Решение вызовов, связанных с внедрением DevOps, требует тесного сотрудничества между разработчиками, операционными специалистами и другими участниками команды, а также поддержки со стороны руководства.

Итак, DevOps предлагает множество преимуществ, но также требует учета вызовов, связанных с его внедрением. Правиль-

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru