

# Список терминов

**Agile (Гибкая разработка программного обеспечения)** - методология доставки программного обеспечения, основанная на коротких итерационных этапах разработки, где каждый спринт должен приводить к эксплуатационному продукту. Это позволяет легко корректировать требования проекта в случае необходимости и дает возможность творчества и гибкости в командах разработчиков.

**CI / CD (Непрерывная интеграция / Непрерывная доставка)** — основа современной культуры DevOps. CI гарантирует, что новый код передается в централизованное хранилище кода несколько раз в день, чтобы пройти автоматические модульные тесты и ускорить сборку нового программного обеспечения. Если тесты пройдены успешно, CD гарантирует, что новая версия приложения будет автоматически отправлена в промежуточную и производственную среды без простоев службы. Рабочий процесс CI / CD гарантирует, что все ошибки будут найдены и исправлены на ранней стадии, а продукт будет доступен в любое время.

**Инфраструктура** — весь комплекс оборудования, программного обеспечения и процессов, необходимых для запуска приложений, а также для сбора, управления и хранения данных.

**IaC** — Инфраструктура как код — один из базовых принципов DevOps. Это означает, что конфигурация инфраструктуры выполняется с помощью машиночитаемых декларативных файлов, а не вручную или с использованием интерактивных инструментов. Эти файлы (например, манифесты Kubernetes или Terraform ) могут храниться в репозиториях GitHub, настраиваться и корректироваться так же, как и код, что обеспечивает эффективную автоматизацию обеспечения инфраструктуры.

**Bare-metal** — случай, когда программное обеспечение установлено на физических устройствах (жестких дисках), пропуская уровень виртуализации.

**Кластер** — это набор взаимосвязанных экземпляров (серверы без поддержки, виртуальные машины, модули Kubernetes и т. Д.), Которые рассматриваются как единое целое для обеспечения балансировки нагрузки, автоматического масштабирования и высокой доступности.

**Гипервизор** – это программа, которая управляет физическими ресурсами вычислительной машины и распределяет эти ресурсы между несколькими различными операционными системами, позволяя запускать их одновременно.

**Виртуальная машина** - программная система, эмулирующая аппаратное обеспечение компьютера.

**Контейнер** – это программная абстракция, внутри которой запускается необходимое приложение вместе со всем своим окружением и изолированная от остальной системы.

**Гостевая операционная система** – операционная система, установленная внутри виртуальной машины.

**Node (узел, нода)** — абстракция компонента инфраструктуры, представляющая из себя в зависимости от контекста – физический сервер, контейнер, виртуальную машину или сервис.

**Среда (Environment)** — все ресурсы сервера (ОС, библиотеки, API, инструменты и платформы и т. Д.), Необходимые для запуска программного обеспечения на различных этапах его жизненного цикла (разработка, тестирование, подготовка, производство).

**IOPS(Пропускная способность ввода / вывода)** — количество операций ввода / вывода в секунду, характеристика пропускной способности дисковых накопителей.

**Snapshot** — создания мгновенной статической копии содержимого вашего экземпляра в целях резервного копирования и восстановления.

**Technical debt (Технический долг)** — концепция нежелательного количества работы разработчиков, необходимого для исправления простого кода, используемого для быстрого получения результатов вместо того, чтобы тратить время на разработку и внедрение наилучшего решения.

**Репозиторий** – хранилище исходного кода, такое как git или SVN

**Commit (Комит)** — процесс отправки кода в репозиторий Git и полученный фрагмент кода.

**Bucket** — логическая единица в Amazon S3 (Simple Storage Service), используемая для хранения нескольких типов объектов (в основном, различных данных и метаданных, которые их описывают).

**Сборка** — это конкретная версия программного кода, в основном называемая этапом разработки новых функций.

**Back-end** — программный движок, к которому у пользователя нет прямого доступа. Этот механизм получает запросы от пользовательского интерфейса и выполняет определенные операции, включая загрузку и сохранение данных в базе данных и т. д.

**Регрессионное тестирование (Regression testing)** — комплексное тестирование обновленной версии продукта, чтобы убедиться, что последняя сборка не оказала негативного влияния на уже имеющуюся функциональность.

**Модульное тестирование (Unit testing)** — основа CI / CD, модульное тестирование — это практика тестирования кода приложения небольшими блоками на основе кода автоматизированного тестирования перед сборкой приложения, чтобы минимизировать

время, необходимое для обнаружения и исправления ошибок, сокращая время выхода на рынок.

**Staging environment (Stage, промежуточная среда)** — контролируемая копия вашей производственной среды, максимально напоминающая ее. Это позволяет тестировать новые версии программного обеспечения, чтобы находить ошибки перед выпуском в эксплуатацию.

**MTTR (среднее время до восстановления)** — среднее ожидаемое время, когда отказавший системный компонент снова заработает; основной параметр сценариев восстановления после сбоев, системного стресс тестирования и проверки производительности.

**Lead time (Время выполнения)** — время, необходимое для перемещения нового пакета кода из коммита в релиз.

**Rollback (Откат)** — ручное или автоматическое восстановление ранее сохраненного состояния программы или базы данных.

---

Версия #2

Артем Швидкий создал 20 октября 2023 11:30:57

Артем Швидкий обновил 20 октября 2023 11:39:22