

Рассмотрим пример DevOps конвейера для веб-приложения на основе Git, Jenkins, Docker и Kubernetes.

*далее мы рассмотрим инструменты, пока просто поймите процесс

1. Continuous Integration (Непрерывная Интеграция):

- **Шаг 1: Разработка и commit**
 - Разработчик пишет код на локальной машине.
 - Код коммитится в Git репозиторий(отправляет код в облачный репозиторий Git со своего компьютера).
- **Шаг 2: Сборка**
 - Jenkins, настроенный на обнаружение новых коммитов, запускает сборку проекта.
 - В этом процессе, Jenkins получает код из Git, компилирует его, и запускает автоматизированные тесты.

2. Continuous Delivery (Непрерывная Ддоставка):

- **Шаг 3: Сборка Docker образа**
 - Если сборка и тесты прошли успешно, Jenkins собирает Docker образ с веб-приложением.
- **Шаг 4: Развертывание в тестовой среде**
 - Docker образ развертывается в тестовой среде с использованием Kubernetes.
 - Проводится ручное тестирование приложения.

3. Continuous Deployment (Непрерывное развертывание):

- **Шаг 5: Развертывание в продакшен**
 - После успешного тестирования, Docker образ развертывается в продакшен среде через Kubernetes.
 - Этот процесс может быть автоматизирован или требовать ручного подтверждения, в зависимости от настроек конвейера.

- **Шаг 6: Мониторинг и обратная связь**

- После развертывания, производится мониторинг приложения и сбор логов для анализа.
- Ошибки и отклонения обратно сообщаются разработчикам для быстрого исправления.

В итоге:

1. Continuous Integration:

- Основной фокус на постоянной интеграции кода и его проверке.
- Позволяет обнаруживать проблемы на ранней стадии разработки.

2. Continuous Delivery:

- Расширяет CI, добавляя автоматизацию процесса доставки.
- Помогает обеспечивать стабильность и надежность развертывания.

3. Continuous Deployment:

- Автоматизирует последний этап, позволяя быстро и безопасно внедрять новые изменения в продакшен.
- Помогает сократить время от идеи до реализации.

