CI/CD пайплайн

Правдин Иван

24 сентября 2025 г.

Что мы имеем после прошлой лекции

Наш webhook сервер работает:

- Git push → автоматический деплой
- Разные ветки обновляют разные окружения

Но:

- Изменения вебхук сервера требуют ручного вмешательства
- Нет простого доступа к логам скриптов, нет статуса деплоя

Индустриальные решения

В реальных проектах используют готовые платформы:

- **Jenkins** — самый популярный, self hosted
- **GitHub Actions** — встроен в GitHub, есть бесплатный режим
- **GitLab CI** — встроен в GitLab, есть self hosted режим
- **Azure DevOps, TeamCity, CircleCI...**

Общая идея: вместо своего webhook сервера используем платформу (отдельное приложение), которая позволяет выполнять скрипты по событиям в Git репозитории.

Преимущества и недостатки платформ

Плюсы:

- Надежность
- Масштабируемость
- Удобство
- Интеграции
- Переиспользуемые скрипты

Минусы:

- Дополнительная сложность
- Свой язык автоматизации
- Ограниченные возможности

Концепция Pipeline

У всех этих платформ есть общая концепция — **Pipeline**

Pipeline — структурированная последовательность шагов автоматизации

Типичные шаги:

1. Получить код из Git
2. Собрать приложение (если нужно)
3. Запустить тесты
4. Задеплоить на сервер

GitHub Actions

Как GitHub Actions реализует концепцию Pipeline:

Базовые понятия:

- **Workflow** — описание pipeline в YAML файле
- **Job** — группа связанных шагов (например, "тестирование")
- **Step** — отдельная команда или действие
- **Action** — готовый переиспользуемый компонент
- **Runner** — сервер, который выполняет workflow

Структура: Workflow содержит Jobs, Job содержит Steps

Триггеры: push, pull request, расписание, ручной запуск

Демонстрация: Переносим webhook на GitHub Actions

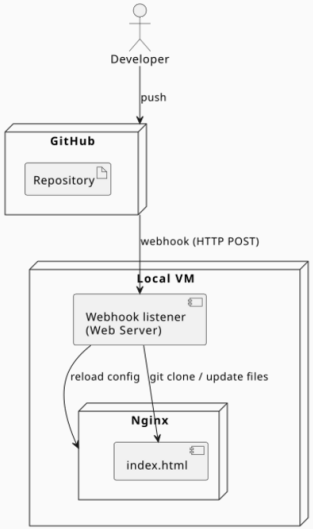
Задача: воспроизвести логику нашего webhook сервера в GitHub

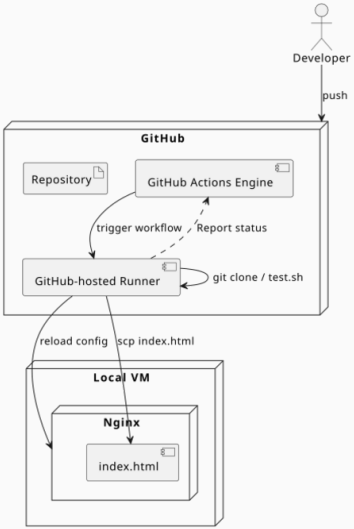
Что покажем:

- Как добавить workflow в GitHub репозиторий
- Описание workflow
- Работа с секретами в GitHub Actions
- Защита ветки от интеграции сломанного кода

Демо-репозиторий:
https://github.com/prafdin/devops-demo-website/tree/cicd-demo

Webhook vs GitHub Actions pipeline





Наш pipeline — это CI/CD pipeline

Поздравляем! Мы только что создали CI/CD pipeline

CI/CD — собирательный термин для автоматизации задач разработки:

- **CI** — Continuous Integration (непрерывная интеграция)
- **CD** — Continuous Deployment (непрерывное развертывание)

Что такое CI и CD

CI задачи это шлюзы:

- Линтинг
- Тесты
- Сканеры безопасности

Цель: убедиться что код соответствует требованиям

CD задачи это преобразования:

- Сборка приложения
- Публикация и обновление приложения

Цель: изменить состояние артефакта

Что запомнить

- CI/CD платформы решают проблемы самодельных решений
- CI - проверки, CD - преобразования

Вопросы?