Hастройка CI/CD Pipeline на GitHub Actions

Для выполнения задания был выбран проект WalletService. Это Java-приложение, которое использует Spring Boot для реализации функциональности и управляется с помощью Maven. Проект построен в архитектуре с несколькими модулями, такими как audit-starter, common, и application.

Код файла ci-cd.yml name: CI/CD Pipeline on: push: branches: - main pull request: branches: - main jobs: build-test: runs-on: \${{ matrix.os }} strategy: matrix: os: [ubuntu-latest, windows-latest, macos-latest] steps: # Шаг 1: Checkout кода из репозитория - name: Checkout Code uses: actions/checkout@v3

Шаг 2: Установка Java и Maven

```
- name: Set up Java
    uses: actions/setup-java@v3
    with:
     java-version: '21'
      distribution: 'temurin'
   # Шаг 3: Кэширование зависимостей Maven
   - name: Cache Maven dependencies
    uses: actions/cache@v3
    with:
      path: ~/.m2
      key: ${{ runner.os }}-maven-${{ hashFiles('**/pom.xml') }}
      restore-keys:
       ${{ runner.os }}-maven
   # Шаг 4: Сборка проекта
   - name: Build with Maven
    run: mvn clean package -DskipTests
 deploy:
  runs-on: ubuntu-latest
  steps:
   - name: Checkout code
    uses: actions/checkout@v3
   - name: Install Render CLI
    run: |
      curl -fsSL https://raw.githubusercontent.com/render-
oss/cli/refs/heads/main/bin/install.sh | sh
   - name: Deploy to Render
```

env:

```
RENDER_API_KEY: ${{ secrets.RENDER_API_KEY }}
```

run:

render deploys create srv-ctc2u4jtq21c73dj01g0 --wait --confirm

Описание настроенных шагов в джобе:

1. Checkout кода:

Используется GitHub Action actions/checkout@v3, чтобы клонировать репозиторий и получить доступ к исходному коду.

2. Установка Java:

С помощью actions/setup-java@v3 устанавливается версия Java 17, необходимая для сборки проекта.

3. Сборка проекта:

С помощью команды mvn clean package создается исполняемый JAR-файл, используя конфигурацию Maven.

4. Установка Render CLI:

Render CLI устанавливается с помощью команды:

curl -fsSL https://raw.githubusercontent.com/render-oss/cli/refs/heads/main/bin/install.sh | sh

5. Деплой на Render:

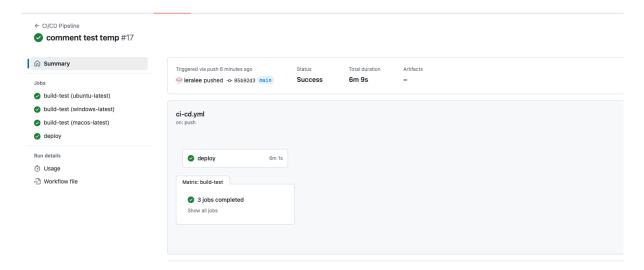
Используя CLI Render и заранее настроенный RENDER_API_KEY, проект деплоится с помощью команды:

render deploys create srv-ctc2u4jtq21c73dj01g0 --wait -confirm

Ссылка на репозиторий

https://github.com/leralee/WalletService

Результат деплоя



Скриншот приложения на Render

