DevOps

CI/CD z GitHub Actions — Symfony, testy i automatyczny deployment

Anna Nowak 09.10.2026 242 wyświetleń 0 komentarzy

Dostępne w językach: PL EN

Czym jest CI/CD?

- **CI** *(Continuous Integration)* — automatyczne testy i analiza kodu przy każdym pushu lub pull requeście
- **CD** *(Continuous Deployment)* — automatyczny deployment na serwer po przejściu wszystkich testów

> *Cel: żeby nie robić deployu „ręcznie przez SSH w piątek o 17:00".*

Podstawowy workflow

Utwórz plik `.github/workflows/ci.yml` :

```yaml
name: CI

on:
  push:
    branches: [main, develop]
  pull_request:
    branches: [main]

jobs:
  test:
    runs-on: ubuntu-latest

    services:
      postgres:
        image: postgres:16-alpine
        env:
          POSTGRES_DB:       test_db
          POSTGRES_USER:     test_user
          POSTGRES_PASSWORD: secret
        ports:
          - 5432:5432
        options: >-
          --health-cmd pg_isready
          --health-interval 10s
          --health-timeout 5s
          --health-retries 5

    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Setup PHP
        uses: shivammathur/setup-php@v2
        with:
          php-version: '8.3'
          extensions: pdo_pgsql, intl
          coverage: xdebug

      - name: Cache Composer
        uses: actions/cache@v4
```

```
        with:
          path: vendor
          key: composer-${{ hashFiles('composer.lo

      - name: Install dependencies
        run: composer install --no-progress

      - name: Setup test database
        run: |
          php bin/console doctrine:database:create
          php bin/console doctrine:migrations:migr

      - name: Run PHPUnit
        run: php bin/phpunit --coverage-text

      - name: Run PHPStan
        run: vendor/bin/phpstan analyse
```

Deployment po zielonych testach

```
deploy:
  needs: test
  runs-on: ubuntu-latest
  if: github.ref == 'refs/heads/main'

  steps:
    - name: Deploy via SSH
      uses: appleboy/ssh-action@v1.0.0
      with:
        host:     ${{ secrets.SSH_HOST }}
        username: ${{ secrets.SSH_USER }}
        key:      ${{ secrets.SSH_PRIVATE_KEY }}
        script: |
          cd /var/www/myapp
          git pull origin main
          composer install --no-dev --optimize-a
```

```
php bin/console cache:clear --env=prod
php bin/console doctrine:migrations:mi
```

Zmienne `SSH_HOST`, `SSH_USER`, `SSH_PRIVATE_KEY` dodaj w **GitHub → Settings → Secrets**.

Przydatne akcje

Akcja	Zastosowanie
`actions/checkout@v4`	Pobieranie kodu
`shivammathur/setup-php@v2`	Konfiguracja PHP
`actions/cache@v4`	Cache vendor/, node_modules/
`appleboy/ssh-action`	Deployment przez SSH

Statusy buildów w README

Dodaj badge do `README.md`:

```
![CI](https://github.com/USER/REPO/actions/workflo
```

→ 

Tagi: #Symfony #Docker #Linux #CI/CD

Udostępnij: Facebook X LinkedIn Kopiuj link

Komentarze (0)

Bądź pierwszy — zostaw komentarz!

Zaloguj się, aby dodać komentarz.

FactoryCode

Projekt demonstracyjny zbudowany na Symfony 8, PHP 8.4 i PostgreSQL 16.
Tłumaczenia zarządzane przez panel admina z AI.

NAWIGACJA

- Start
- Blog
- Projekty
- Kontakt
- Polityka prywatności

KONTAKT

- kontakt@factorycode.pl
- GitHub

© 2026 FactoryCode. Wszelkie prawa zastrzeżone.

Zbudowane z Symfony 8 & Claude AI