ТЕМА 16

CI/CD

Автоматизація збірки, тестування та розгортання

Бекенд-розробка, сервіси та мікросервіси

Комп'ютерна інженерія · 3 курс · Лекція 16 з 18

ПЛАН ЛЕКЦІЇ

7 питань сьогоднішньої теми

01

Що таке CI/CD

02

Continuous Integration

03

Етапи CI/CD пайплайну

04

GitHub Actions для PHP

05

Delivery vs Deployment

06

Rolling update, Blue-Green

07

Від коміту до production

Розшифровка абревіатури

Термін	Розшифровка
CI	Continuous Integration — автоматична збірка й тестування
CD (Delivery)	Continuous Delivery — готовність до розгортання, останній крок — вручну
CD (Deployment)	Continuous Deployment — автоматичне розгортання без підтвердження

Без автоматизації: запустити тести вручну, підключитись до сервера, скопіювати файли, перезапустити — кожен крок є ризиком людської помилки.

Автоматизація тестування

```
Розробник → git push → CI-система автоматично:

  1. Завантажує залежності (composer install)
  2. Запускає unit-тести (тема 11, PHPUnit)
  3. Запускає функціональні тести
  4. Перевіряє стиль коду
  5. Повідомляє результат: ✓ успішно чи ✗ помилка
```

Помилка виявляється протягом хвилин, поки контекст зміни ще свіжий — не через тиждень.

П'ять етапів пайплайну

Якщо будь-який етап завершується невдачею, пайплайн зупиняється — це запобігає потраплянню неперевіреного коду на production.

.github/workflows/ci.yml

```yaml
name: CI

on:
  push: { branches: [main] }
  pull_request: { branches: [main] }

jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Встановити PHP
        uses: shivammathur/setup-php@v2
        with: { php-version: '8.3' }

      - name: Встановити залежності
        run: composer install --no-progress

      - name: Запустити тести
        run: vendor/bin/phpunit
```

Ключові елементи YAML

Ключ	Призначення
`on`	Подія, що запускає workflow
`jobs`	Набори кроків, що можуть виконуватись паралельно
`runs-on`	Тип віртуальної машини для job
`steps`	Послідовність кроків усередині job
`uses / run`	Готовий крок з реєстру / довільна команда

Хто натискає "розгорнути"?

Continuous Delivery

Автоматично готово до розгортання, але фактичний запуск ініціює людина

Continuous Deployment

Розгортається на production повністю автоматично, без втручання людини

Це організаційне, а не технічне рішення — критичні системи частіше обирають Delivery.

Job розгортання (Continuous Deployment)

```
deploy:
  needs: test
  runs-on: ubuntu-latest
  if: github.ref == 'refs/heads/main'
  steps:
    - name: Розгорнути на сервер
      run: |
        docker build -t myapp:${{ github.sha }} .
        docker push myregistry/myapp:${{ github.sha }}
        ssh user@server 'docker compose up -d'
```

Rolling update: поступова заміна

← перший екземпляр уже оновлено, інші чекають своєї черги

Нові екземпляри запускаються поступово: один за одним, з перевіркою працездатності. У будь-який момент сервіс залишається доступним — без простою (downtime).

Blue-Green Deployment

СИНЯ (production)

ЗЕЛЕНА (нова версія)

балансувальник миттєво перемикає трафік

Нову версію розгортають і тестують у "зеленому" середовищі, не зачіпаючи "синє". Коли готова — трафік перемикається миттєво; при проблемі так само миттєво відкат назад.

Порівняння стратегій

Стратегія	Перевага	Недолік
Rolling update	Не потребує подвійних ресурсів	Дві версії тимчасово працюють одночасно
Blue-Green	Миттєве перемикання й відкат	Вдвічі більше ресурсів під час розгортання

Повний шлях однієї зміни коду

1. git push у feature-branch
2. Pull request → GitHub Actions запускає CI
3. Тести не пройшли? Розробник бачить помилку одразу
4. CI пройшов + code review → merge у main
5. Збірка Docker-образу й публікація в реєстр
6. Rolling update оновлює контейнери без простою

Підсумок теми 16

- CI/CD автоматизує шлях коду від коміту до production, усуваючи людські помилки
- CI виявляє проблему одразу після зміни, поки контекст ще свіжий
- Пайплайн зупиняється на першій-ліпшій невдачі — неперевірений код не потрапляє далі
- GitHub Actions описує workflow декларативно у файлі YAML прямо в репозиторії
- Delivery — розгортання підтверджує людина; Deployment — повністю автоматично
- Rolling update і Blue-Green дозволяють оновлювати застосунок без простою