

CI/CD: Automatizace životního cyklu softwaru

CI/CD je zkratka pro Kontinuální integraci (Continuous Integration) a Kontinuální nasazování (Continuous Deployment). Tento proces, často označovaný jako **CI/CD pipeline** (potrubí), odstraňuje lidskou chybovost při testování a nasazování aplikací.

1. CI: Kontinuální integrace (Continuous Integration)

CI je praxe pravidelného (i několikrát denně) začleňování změn kódu do hlavní větve v [repozitáři](#).

- **Proces:** Pokaždé, když vývojář provede `git push`, automatizovaný server kód stáhne, sestaví (build) a spustí testy.
- **Cíl:** Odhalit chyby a konflikty v kódu okamžitě po jejich vzniku, nikoliv až těsně před vydáním.

Klíčové kroky v CI:

- **Build:** Překlad zdrojového kódu do spustitelné podoby.
- **Unit Tests:** Testování jednotlivých funkcí.
- **Lintery:** Kontrola kvality a formátování kódu.

2. CD: Kontinuální doručování a nasazování

Na fázi CI navazuje část CD, která má dvě podoby:

Continuous Delivery (Kontinuální doručování)

Každá verze, která projde testy, je připravena k nasazení. Samotný krok „nasadit na produkci“ však stále vyžaduje **manuální schválení** člověkem (např. kliknutí na tlačítko).

Continuous Deployment (Kontinuální nasazování)

Nejpokročilejší fáze. Pokud kód projde všemi automatizovanými testy, je **automaticky nasazen** přímo koncovým uživatelům bez jakéhokoliv lidského zásahu.

3. Přínosy CI/CD

- **Rychlost (Time-to-Market):** Nové funkce se k uživatelům dostanou v řádu minut či hodin,

nikoliv měsíců.

- **Kvalita:** Automatizované testy zachytí většinu chyb (regresí) dříve, než ovlivní zákazníky.
- **Menší riziko:** Nasazují se malé změny. Pokud se něco pokazí, je snadné identifikovat chybu a vrátit se k předchozí verzi (Rollback).
- **Produktivita:** Vývojáři se nemusí starat o manuální nahrávání souborů na servery.

4. Nástroje pro CI/CD

Existuje mnoho platforem, které se integrují přímo s vaším repozitářem:

Nástroj	Charakteristika
—	—
GitHub Actions	Přímo integrované v GitHubu, využívá YAML konfiguraci.
GitLab CI/CD	Považováno za jedno z nejrobustnějších vestavěných řešení.
Jenkins	Open-source standard, extrémně flexibilní díky tisícům pluginů.
CircleCI / Travis CI	Populární cloudové služby pro rychlé nastavení.

5. Praktický příklad (Pipeline v YAML)

Většina moderních CI/CD systémů se konfiguruje pomocí souboru v repozitáři (např. `.github/workflows/main.yml`):

```
stages:
  - build
  - test
  - deploy

build_job:
  script: npm install && npm run build

test_job:
  script: npm test

deploy_job:
  script: scp -r dist/* server:/var/www/
  only:
    - main
```

Související články:

- [Repozitář: Základna pro CI/CD](#)

- [Git: Jak spouštět automatizaci pomocí commitů](#)
- [Docker: Jak CI/CD balí aplikace do kontejnerů](#)

Tagy: it cicd devops automation software-engineering deployment

From:

<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:

<https://serviceit.cz/doku.php?id=it:sw:cicd&rev=1767382186>

Last update: **2026/01/02 20:29**

