

CI/CD

Inhaltsverzeichnis

- [1 Funktionsweise](#)
- [2 Typische Einsatzbereiche](#)
- [3 Wichtige Hinweise](#)
- [4 Fazit](#)

CI/CD steht für Continuous Integration und Continuous Delivery bzw. Deployment. Automatisierte Pipelines bauen, testen und verteilen Softwareänderungen zuverlässig – von jedem Commit bis zur produktiven Freigabe.

1 Funktionsweise

Continuous Integration (CI) integriert Codeänderungen häufig in ein gemeinsames Repository und prüft sie automatisch (Build, Tests, Qualität). **Continuous Delivery/Deployment (CD)** überführt erfolgreiche Builds automatisiert oder halbautomatisiert in Zielumgebungen.

- **Trigger:** Commit, Merge Request oder Zeitplan startet die Pipeline.
- **Stages:** z. B. Build ? Test ? Artifact ? Deploy.
- **Artifacts:** bau- oder containerisierte Ergebnisse für Folge-Schritte.

2 Typische Einsatzbereiche

- Web- und API-Projekte mit häufigen Releases
- Infrastructure-as-Code und Config-Deployments
- Qualitätssicherung durch automatisierte Tests vor dem Go-live

3 Wichtige Hinweise

- Pipelines brauchen Secrets-Management und getrennte Umgebungen (Dev/Stage/Prod).
- Fehlgeschlagene Tests müssen Deployments zuverlässig stoppen.
- Rollback-Strategien und Monitoring gehören zur Pipeline-Denkweise dazu.

4 Fazit

CI/CD reduziert manuelle Fehler, verkürzt Release-Zyklen und macht Softwareänderungen nachvollziehbar – Voraussetzung für stabile, wiederholbare Deployments.