DevOps a automatizace

DevOps je metodika zaměřená na sjednocení vývoje softwaru (Development) a jeho provozu (Operations). Hlavním cílem je zkrácení systémového životního cyklu vývoje při zachování vysoké kvality softwaru. DevOps stojí na pilířích kultury, automatizace, měření a sdílení (metodika CAMS).

1. Životní cyklus CI/CD

Základním technickým nástrojem DevOps je potrubí (pipeline) **CI/CD**, které automatizuje cestu kódu od programátora až k zákazníkovi.

CI (Continuous Integration - Průběžná integrace)

Při každém nahrání kódu do verzovacího systému se automaticky spustí proces:

- **Build:** Sestavení aplikace a stažení závislostí.
- **Automated Testing:** Spuštění unit a integračních testů. Pokud testy selžou, proces se zastaví.
- **Statická analýza:** Kontrola kvality kódu a bezpečnostních zranitelností.

CD (Continuous Delivery / Deployment)

- **Continuous Delivery:** Kód je po testech připraven k nasazení, ale finální krok (stisknutí tlačítka) provádí člověk.
- **Continuous Deployment:** Kód, který projde všemi testy, je **automaticky** nasazen přímo do produkčního prostředí bez lidského zásahu.

—

2. Infrastructure as Code (IaC)

V DevOps světě se už servery neklikají v portálu. Celá infrastruktura je definována textovým kódem.

- **Výhody:** Verziovatelnost (můžete se vrátit k minulé verzi sítě), reprodukovatelnost (stejné prostředí pro testování i produkci).
- **Nástroje pro provizování: Terraform**, CloudFormation. Definují virtuální stroje, sítě a databáze.
- **Nástroje pro konfiguraci: Ansible**, Puppet, Chef. Definují, co má být na serveru nainstalováno (např. verze PHP, Nginx).

—

3. Kontejnery a Orchestrace

DevOps se dnes neobejde bez technologií, které izolují aplikace a usnadňují jejich škálování.

- **Docker:** Zabalí aplikaci a její prostředí do lehkého kontejneru.
- **Kubernetes (K8s):** Nástroj pro orchestraci, který automaticky spravuje tisíce kontejnerů – hlídá, aby běžely, restartuje je při pádu a automaticky je škáluje podle zátěže.

—

4. Monitoring a Observability

Automatizace nekončí nasazením. DevOps týmy musí mít přehled o stavu systému v reálném čase.

- **Logování:** Centrální sběr logů (např. ELK stack – Elasticsearch, Logstash, Kibana).
- **Metriky:** Sledování vytížení CPU, RAM a počtu chyb (např. Prometheus + Grafana).
- **Tracing:** Sledování cesty jednoho požadavku skrze mikroslužby (např. Jaeger).

—

5. Přínosy DevOps pro byznys

Metrika	Tradiční IT	DevOps přístup
Frekvence nasazení	Jednou za měsíc/kvartál	Mnohokrát denně
Doba na opravu (MTTR)	Hodiny až dny	Minuty
Chybovost změn	Vysoká (manuální chyby)	Nízká (otestovaná automatizace)

Související články:

- Verzování kódu (Git)
- Testování softwaru a QA
- Cloud computing (AWS, Azure, GCP)

Tagy: devops automation ci_cd docker kubernetes terraform ansible infrastructure