

Kubernetes (K8s)

Kubernetes je open-source platforma pro orchestraci kontejnerů. Umožňuje automatizovat nasazování, škálování a správu kontejnerizovaných aplikací. Původně byl vyvinut společností Google (založen na interním systému Borg) a nyní je spravován nadací CNCF.

Zatímco **Docker** definuje, jak kontejner vypadá, Kubernetes určuje, jak se kontejnery chovají v clusteru, jak spolu komunikují a jak se zotavují z chyb.

1. Architektura Clusteru Kubernetes cluster se skládá ze dvou hlavních částí:

- **Control Plane (Řídící rovina):** „Mozek“ clusteru. Rozhoduje o tom, kde se kontejnery spustí, sleduje stav clusteru a reaguje na události. Obsahuje komponenty jako *API Server*, *etcd* (databáze stavu), *Scheduler* a *Controller Manager*.
- **Worker Nodes (Pracovní uzly):** Stroje (fyzické nebo virtuální), kde skutečně běží vaše aplikace. Každý uzel obsahuje službu **Kubelet** (komunikuje s řídicí rovinou) a **Container Runtime** (např. Docker nebo containerd).

2. Základní objekty (Abstrakce) Kubernetes nepracuje přímo s kontejnery, ale s vyššími objekty:

- **Pod:** Nejmenší jednotka v K8s. Může obsahovat jeden nebo více úzce propojených kontejnerů, které sdílejí stejnou IP adresu a úložiště.
- **Deployment:** Definuje požadovaný stav aplikace (např. „chci, aby vždy běžely 3 repliky mého webu“). Pokud jeden Pod spadne, Deployment zajistí spuštění nového.
- **Service:** Zajišťuje stabilní síťovou adresu a balancování zátěže (Load Balancing) mezi skupinu Podů.
- **Namespace:** Virtuální oddělení zdrojů uvnitř jednoho clusteru (např. vyvoj, test, produkce).

3. Deklarativní konfigurace (YAML) V Kubernetes neříkáme systému „spuště toto“, ale definujeme **požadovaný stav** v YAML souboru. Kubernetes pak neustále porovnává aktuální stav s tímto zadáním a provádí opravy.

Příklad jednoduchého Deploymentu:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
```

```
image: nginx:1.21
ports:
- containerPort: 80
```

4. Klíčové výhody

- **Self-healing (Samooprava):** Pokud kontejner selže, K8s ho restartuje. Pokud uzel zemře, přesune kontejnery na jiný uzel.
- **Horizontální škálování:** Automatické navyšování nebo snižování počtu běžících instancí podle zátěže (CPU/RAM).
- **Service Discovery:** Aplikace se v síti najdou automaticky pomocí DNS jmen, bez nutnosti znát konkrétní IP adresy.
- **Automated Rollouts/Rollbacks:** Snadné nasazování nových verzí bez výpadku (Rolling Update).

5. Správa pomocí kubectl Hlavním nástrojem pro ovládání clusteru je CLI utilita kubectl.

Příkaz	Význam
kubectl get pods	Zobrazí seznam běžících Podů.
kubectl apply -f soubor.yaml	Nasadí nebo aktualizuje konfiguraci ze souboru.
kubectl logs <pod-name>	Zobrazí logy z konkrétního kontejneru.
kubectl scale deployment web --replicas=5	Změní počet instancí aplikace na pět.
kubectl exec -it <pod-name> -- bash	Otevře interaktivní terminál uvnitř kontejneru.

Zajímavost: Číslo **8** v názvu K8s nahrazuje osm písmen mezi „K“ a „s“ ve slově „K-ubernete-s“ (tzv. numeronym).

[Zpět na DevOps](#)

From:
<http://serviceit.cz/> - **IT ENCYKLOPEDI E**

Permanent link:
<http://serviceit.cz/doku.php?id=kubernetes>

Last update: **2025/12/31 14:21**

