

Kontejnerizace (Containerization)

Kontejnerizace je forma virtualizace na úrovni operačního systému. Umožňuje zabalit aplikaci a všechny její závislosti (knihovny, konfigurační soubory, binární soubory) do jednoho standardizovaného celku zvaného **kontejner**.

Hlavním heslem kontejnerizace je: „Sestav jednou, spouštěj kdekoli.“ To znamená, že aplikace v kontejneru se bude chovat naprosto stejně na notebooku vývojáře, v testovacím prostředí i na produkčním serveru v cloudu.

1. Kontejnery vs. Virtuální stroje (VM)

Ačkoliv oba přístupy slouží k izolaci aplikací, fungují odlišně:

- **Virtuální stroje:** Každý stroj obsahuje kompletní kopii operačního systému, virtuální hardware a aplikaci. Jsou náročné na paměť a pomalu startují.
- **Kontejnery:** Sdílejí jádro (kernel) hostitelského operačního systému. Jsou extrémně lehké, startují v řádu sekund a spotřebovávají minimum systémových prostředků.

2. Proč je kontejnerizace tak populární?

- **Izolace:** Aplikace v jednom kontejneru neovlivňuje aplikaci v jiném. Každá může mít jinou verzi stejné knihovny (např. jedna Python 3.8, druhá Python 3.11).
- **Přenositelnost:** Kontejner obsahuje vše, co aplikace potřebuje. Odpadá problém „u mě na počítači to fungovalo“.
- **Škálovatelnost:** Díky nízkým nárokům lze snadno spustit desítky nebo stovky instancí téže aplikace podle aktuální zátěže.
- **Agilita:** Ideální pro architekturu **Mikroslužeb**, kde je velká aplikace rozdělena na mnoho malých, nezávisle běžících částí.

3. Hlavní technologie

Docker

Nejnámější platforma pro vytváření a správu kontejnerů. Definice kontejneru se píše do tzv. **Dockerfile**, ze kterého se sestaví **Image** (obraz), a z něj pak běží samotný **Container**.

Kubernetes (K8s)

Když máte stovky kontejnerů, potřebujete nástroj na jejich řízení (orchestraci). Kubernetes

automatizuje nasazování, škálování a správu kontejnerů napříč celým serverovým clusterem.

4. Využití v oblasti AI a Data Science

V oblasti strojového učení je kontejnerizace nepostradatelná pro:

- **Reprodukovatelnost:** Zajištění, že trénovací skript poběží se stejnými verzemi knihoven (PyTorch, TensorFlow) i za dva roky.
- **Model Serving:** Snadné nasazení natrénovaného modelu jako [API](#) mikroslužby.
- **Využití GPU:** Kontejnery umožňují izolovaný přístup k ovladačům grafických karet (např. NVIDIA Docker).

5. Bezpečnostní aspekty

Kontejnery jsou sice izolované, ale sdílejí jádro systému. Pokud je v jádře bezpečnostní díra, může být ohrožen celý hostitel. Proto je důležité používat:

- **Minimalistické obrazy:** Obsahují jen to nejnutnější (např. Alpine Linux).
- **Skenování zranitelností:** Automatická kontrola, zda knihovny v kontejneru neobsahují známé chyby.

Zajímavost: První formy kontejnerizace (jako chroot nebo FreeBSD Jails) existovaly desítky let, ale teprve příchod Dockeru v roce 2013 udělal technologii tak jednoduchou, že se stala celosvětovým standardem.

[Zpět na IT architekturu](#)

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

Permanent link:
<https://serviceit.cz/doku.php?id=kontejnerizace>

Last update: **2025/12/31 14:32**

