

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**Docker a kontajnerizácia**

# Obsah

|                                       |   |
|---------------------------------------|---|
| Úvod.....                             | 4 |
| 1 Docker a kontajnerizácia.....       | 5 |
| 1.1 Virtuálny stroj a kontajner ..... | 6 |
| 1.2 Docker .....                      | 7 |
| Záver.....                            | 8 |
| Zdroje .....                          | 9 |

## **Zoznam obrázkov**

Obr. 1 Porovnanie virtuálneho prístroja a kontajnera ..... 7

## **Zoznam skratiek a značiek**

**API**            Rozhranie pre programovanie aplikácií (*Application programming interface*)

# Úvod

Súčasná doba vyžaduje rýchle nasadenie aplikácii, ktoré bude možné jednoducho škálovať a zároveň ich celková veľkosť bude menšia. Práca sa venuje porovnaniu momentálne dvoch najrozšírenejších možností ako aplikácie nasadiť. Pričom je primárne zameraná na kontajnerizáciu a platformu Docker, ktorá slúži na vývoj a nasadenie kontajnerov.

# 1 Docker a kontajnerizácia

Pokiaľ chceme na serveri nasadiť aplikáciu, budeme potrebovať na danom serveri operačný systém a prostredie v ktorom bude aplikácia bežať. Ak by na danom serveri bežala len jedna aplikácia, tak by pravdepodobne nevyužila plný poskytovaný výkon servera. To by malo za následok nasadenie viacerých aplikáciách na daný server. Avšak to by sme mohli naraziť na problém, že dané aplikácie využívajú rôzne verzie istej knižnice, prípadne by sa navzájom aplikácie ovplyvňovali. Jedným možným riešením by bolo vytvorenie virtuálnych strojov na danom serveri, čo by bolo neefektívne. Nakoľko každý virtuálny stroj by si ukrojil určitý výpočtový výkon zo servera, len na samostatný beh virtuálneho stroja, nie len na beh aplikácie. Tento problém by mohol vyriešiť kontajner. Tak ako v reálnom živote existuje kontajner na prepravu tovaru, tak vo virtuálnom svete existuje kontajner „na univerzálnu prepravu softvéru“. V reálnom živote je kontajner vyrobený tak, aby sa dal ľahko prepraviť na rôznych prepravných prostriedkoch, ako je loď, kamión, lietadlo alebo vlak. Cieľom kontajnera vo virtuálnom svete je zabaliť aplikáciu tak aby sa dalo čo najľahšie prepraviť na rôzne systémy. Jednoducho povedané, kontajner je samostatný balík softvéru, ktorý obsahuje všetky potrebné závislosti aby aplikácia mohla byť spustená a bežať, nezávisle od prostredia na ktorom bola vytvorená [1]. Hlavná myšlienka je oddeliť aplikáciu a jej závislosti od operačného systému. Medzi závislosti patrí:

- Kód
- Runtime
- Systémové nástroje
- Knižnice
- Nastavenia

Kontajner by mal obsahovať len jednu aplikáciu, vďaka čomu vieme získať mikroservisovo orientovanú architektúru, ktorá je v súčasnej dobe veľmi populárna [1]. Mikroservisová architektúra nám umožňuje jednoducho udržiavať a škálovať aplikácie.

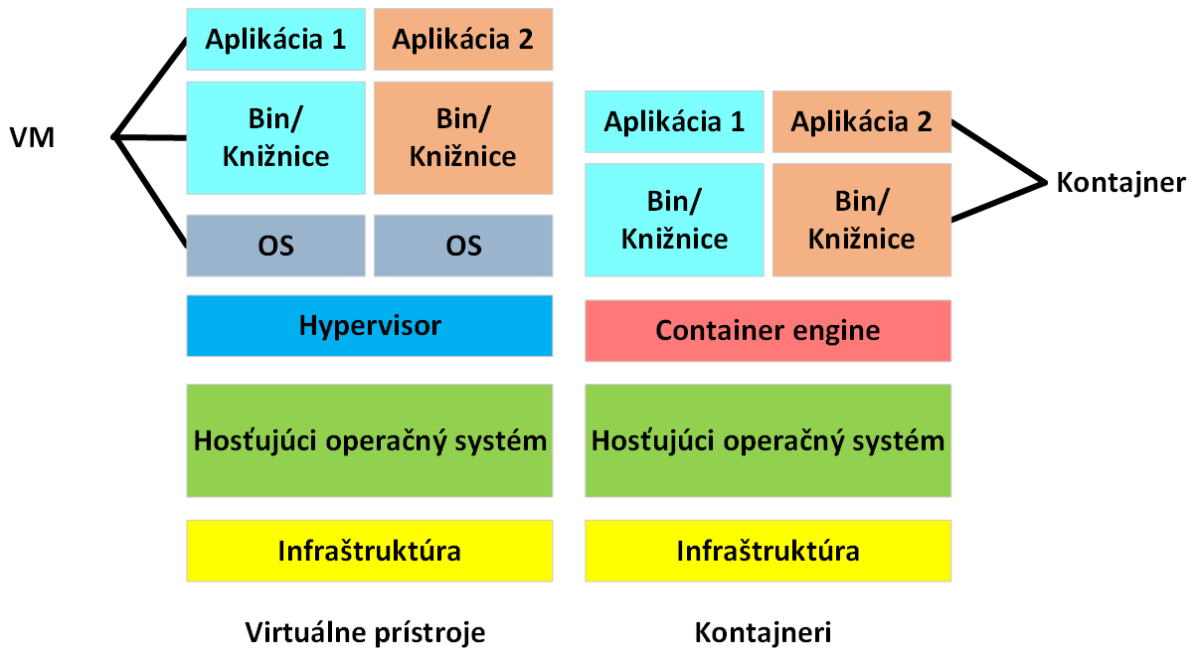
Kontajneri by mali dodržiavať tri princípy [1]:

1. **Štandardnosť** - Kontajneri sú vytvorené podľa štandardu, že sú kompletne oddelené od ich pôvodne systému, vďaka čomu sú prenositeľné na iné systémy. Všetky závislosti, ako programovací jazyk, knižnice a iné, sú súčasťou kontajnera.

2. **Jednoduchosť** - Kontajner by mal obsahovať iba jednu aplikáciu. Viac aplikácií v kontejneri by sa mohli navzájom ovplyvňovať, čo by malo v konečnom dôsledku prinieslo aj zložitosť pri identifikovaní problémov. Kontajneri zdieľajú jadro operačného systému a jeho zdroje, vďaka čomu nie je potrebný vlastný operačný systém pre každý kontajner.
3. **Izolovanosť** - Kontajnery virtualizujú centrálny procesor, pamäť, disk a sieť na úrovni operačného systému, vďaka čomu je kontajner izolovaný a nevznikajú žiadne konflikty medzi internými procesmi kontajnera a externými procesmi hostiteľského operačného systému. Hostiteľ môže komunikovať s kontajnerom iba prostredníctvom jadra, čo poskytuje vyššiu bezpečnosť.

## 1.1 Virtuálny stroj a kontajner

Virtuálny stroj (ďalej v text len VM, od anglického Virtual Machine) je obraz, ktorý sa správa ako reálny počítač a jeho vnútro tvorí operačný systém, vďaka čomu sa správa ako sandbox a je oddelený od hostiteľského operačného systému. Na jednom hostiteľskom stroji môžu bežať viaceré VM, o ktorých beh sa stará „Hypervisor“. VM funguje tak, že si vyhradí virtuálny hardvér, ako sú centrálny procesor, pamäť, úložiska a sieťové rozhrania, ktoré sú následne priradené na reálny hardvér. Prvý problém VM oproti kontajnerom spočíva v tom, že VM potrebuje vlastný operačný systém, čo spôsobuje vyššie nároky na hardvér. VM potrebuje až desiatky gigabajtov miesta namiesto niekoľkých megabajtov ako je tomu v prípade kontajnerov. Pri kontajneri sa spúšťa iba samostatná aplikácia, avšak pri VM sa spúšťa jadro a viacero systémových služieb. Druhým problémom je čas potrebný na inicializáciu VM, nakoľko VM si potrebuje vyhradiť vlastný hardvér a spustiť operačný systém. Kontajneri sa dokážu inicializovať v priebehu niekoľkých sekúnd [1].



Obr. 1 Porovnanie virtuálneho prístroja a kontajnera

## 1.2 Docker

Docker (Docker engine) patrí medzi najznámejšiu otvorenú platformu pre vývoj a kontajnerizáciu aplikácií. Docker používa klient-server architektúru. Vývojári využívajú klienta na komunikáciu s Docker daemon-om, ktorý zostavuje, štartuje a vykonáva rôzne ďalšie operácie nad kontajnermi. Klient a daemon môžu bežať na rovnakom systéme ale môžu bežať aj na rozdielnych systémoch. Komunikácia medzi daemon a klientom je väčšinou riešená cez REST API [1].

Dôležitou súčasťou architektúry je docker registry, ktorý tvorí úložisko pre docker obrazy. K dispozícii sú privátne a verejné registre. Ako už napovedá názov, privátne registry slúži na ukladanie obrazov, ktoré sú dostupné iba pre špecifických užívateľov. Docker obrazy sú read-only šablóny, ktoré slúžia na vytvorenie docker kontajnerov. Kontajner predstavuje špecifickú bežiacu inštanciu obrazu. Kontajner môžeme vytvoriť, zapnúť, vypnúť, odstrániť alebo pripojiť k sieťam a úložiskám. Docker ponúka aj takzvaný Docker swarm, ktorý slúži na jednoduché škálovanie kontajnerov podľa potreby [1].

## **Záver**

Súčasťou práce je opis technológie kontajnerizácie a platformy Docker, a následné porovnanie s virtuálnymi strojmi. Na základe analýzy oboch možností, ako je možné nasadiť aplikáciu, sa kontajner javí ako lepšia možnosť. Nakoľko kontajner predstavuje samostatný balík softvéru, ktorý obsahuje všetky potrebné závislosti aby aplikácia mohla byť spustená a bežať nezávisle od prostredia na ktorom bola vytvorená.



## Zdroje

- [1] Magazín, '*Docker a jeho použitie pri kontajnerizácii*'. [on-line]. [cit: 2020-5-5]. Dostupné na:  
<<https://magazin.kpi.fei.tuke.sk/2019/02/docker-a-jeho-pouzitie-pri-kontajnerizacii>>