



TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

TECHNOLOGIE PRO BIG DATA CVIČENÍ I.

Lukáš Matějů

30.8.2024 | TPB



ORGANIZACE

- základní informace
 - přednášející i cvičící
 - Lukáš Matějů
 - lukas.mateju@tul.cz
 - rozsah předmětu 2+2
 - veškeré materiály zveřejňovány na [elearningu FM](#)
- přednášky
 - každé pondělí od 10:40
 - budova A, místnost A-A0110
 - účast nepovinná, ale vítaná

ORGANIZACE

- cvičení
 - každé pondělí od 12:30 a 14:20
 - budova A, místnost A-A0304
 - samostatné práce volně doplňující přednášky
 - 10 povinných a 10 bonusových úloh
 - každá bonusová úloha je za **0,5** bodu
 - na vypracování a odevzdání úloh je **1** týden
 - odevzdává se **výhradně** na cvičeních (s osobním vysvětlením)
 - za každý týden opožděného odevzdání je **-0,5** bodu
 - finální počet úloh může být ovlivněn odpadnutím výuky
 - 2 povolené absence
 - každá další absence je za **-3** body (onemocnění lze snadno omluvit)
 - absence automaticky prodlužuje dobu odevzdání o **1** týden

ORGANIZACE

- zápočet
 - odevzdané a správně vyřešené povinné úlohy ze cvičení
- zkouška
 - prezenční
 - písemná
 - max 20 bodů
 - 10 otázek po 2 bodech
 - body ze cvičení jsou přenášeny ke zkoušce
 - zaměřená na základní koncepty probírané v rámci předmětu

ORGANIZACE

- hodnocení
 - dvě varianty
 - jen za bonusové body ze cvičení...
 - 5,0 bodů -> 1
 - 4,5 bodů -> 2
 - 4,0 body -> 3
 - povinná docházka na přednášky i cvičení (2 povolené absence)
 - absolvování písemné zkoušky...
 - maximum 25 bodů (20 + 5)
 - ≥ 21 bodů -> 1 ≥ 19 bodů -> 1-
 - ≥ 17 bodů -> 2 ≥ 15 bodů -> 2-
 - ≥ 13 bodů -> 3 < 13 bodů -> 4
 - v případě odpadnutí výuky budou potřebné body upraveny

CO BUDEME POUŽÍVAT?

- větší množství technologií pro velká data
 - Apache Spark, Apache Flink, Apache Kafka, ...
 - často komplexnější instalace
- Docker
 - v učebnách nainstalovaný
 - použití oficiálních image
- cloudová řešení
 - v případě použití zdarma
- obsluha
 - Python

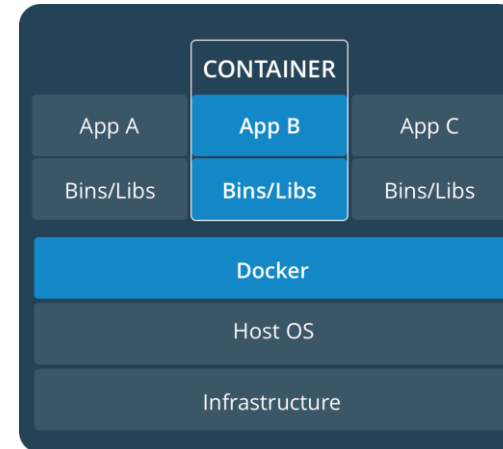
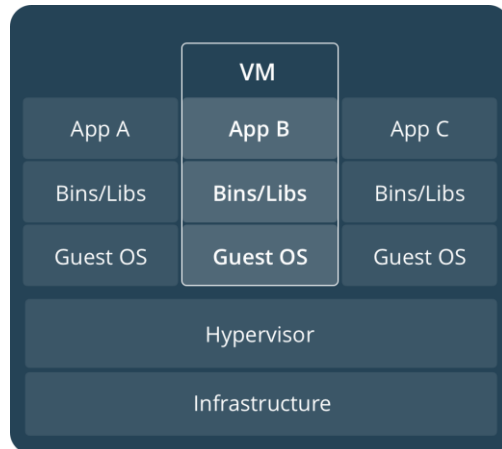


KONTEJNERIZACE

- řeší nedostatky klasické plné virtualizace
- plná virtualizace
 - na serveru je nainstalována softwarová komponenta hypervisor
 - umožňuje vytváření virtuálních strojů
 - každý virtuální stroj se chová jako samostatný server s vlastním OS
 - velké režijní náklady
- kontejnerizace
 - virtualizace jádra OS
 - kontejnery běží v rámci jednoho OS a sdílejí paměť, knihovny a další zdroje
 - kontejnery mohou být izolovány od okolního prostředí
 - a následně nasazeny v různých prostředích
 - snižuje režijní náklady
 - zdroje jsou také využívány efektivněji

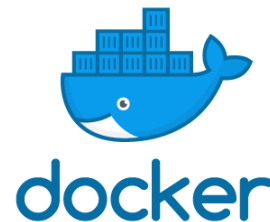


KONTEJNERIZACE



DOCKER

- nejpopulárnější kontejnerová technologie
- izolace aplikací se všemi knihovnami, configy a dalšími soubory
 - kontejnery zajišťují spuštění aplikace v jakémkoliv prostředí
- umožňuje tedy vývoj, sestavení, spuštění i sdílení aplikace uzavřené v kontejneru
- [web](#)

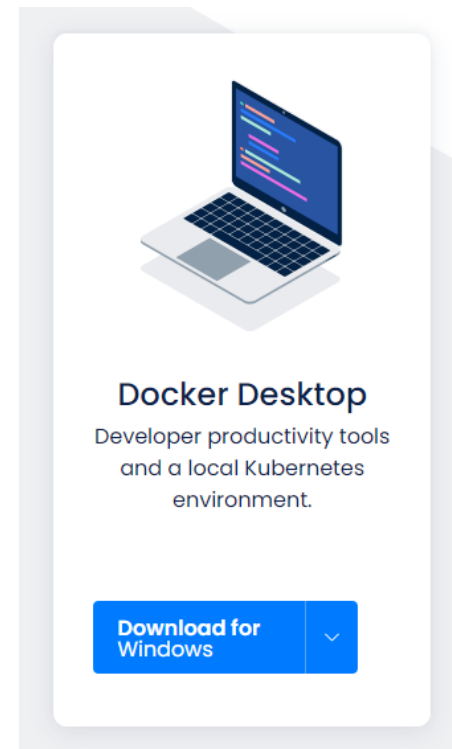


DOCKER

- image
 - obsahuje vše potřebné pro běh programu
 - připravené prostředí bez lokálních úprav
 - zdrojové kódy, závislosti, soubory potřebné pro běh, ...
- kontejner
 - běžící prostředí vytvořené z image
 - obsahuje data a lokální změny
 - přístup pouze ke svému souborovému systému, který spravuje Docker
 - výjimkou je použití Volumes pro ukládání dat – může být sdíleno mezi kontejnery

DOCKER

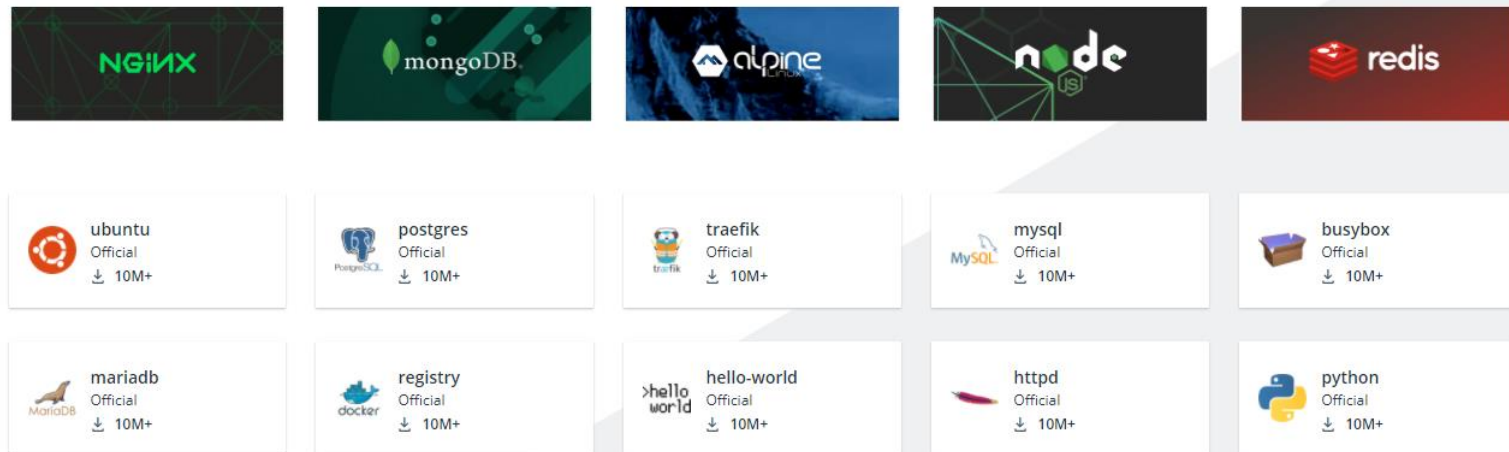
- instalace ([stažení](#))
 - verze pro [Windows](#), [MacOS](#), [Linux](#)
 - přímočará instalace
 - detailní návod na oficiálních stránkách
- ověření instalace
 - `docker --version`
 - `docker run hello-world`
- výpis všech kontejnerů (i zastavených)
 - `docker ps -a`
 - pouze běžící bez -a (--all)
- vytváření vlastní image
 - `docker build -f /path/to/a/Dockerfile .`



DOCKER

- Docker Hub
 - služba pro hledání a sdílení image
 - obsahuje spoustu oficiálních image
 - Ubuntu, MongoDB, Python, ...

Official Images



ZÁKLADNÍ PŘÍKAZY

- spuštění kontejneru z [MongoDB image](#)
 - `docker run -p 27017:27017 --name mongo_cv01 -d mongo:tag`
 - `-p` (`--publish`) pro možnost komunikace na daném portu (`port image:port host`)
 - `-d` (`--detach`) pro běh kontejneru na pozadí
 - `tag` slouží pro uvedení verze
 - bez tagu se stáhne poslední verze (`latest`)
 - `docker container exec -it mongo_cv01 bash`
 - `bash` dovnitř mongo kontejneru
 - ukončení přes příkaz `exit`

ZÁKLADNÍ PŘÍKAZY

- `docker ps`
 - výpis všech běžících kontejnerů
 - sloupec container id
- `docker stop container_id`
 - zastavení běžícího kontejneru
 - stačí uvést první 3 znaky container id
- `docker ps`
 - status exited u cvičného kontejneru
- `docker start container_id`
 - opětovné spuštění kontejneru

ZÁKLADNÍ PŘÍKAZY

- `docker stop container_id`
 - opětovné zastavení
- `docker container rm container_id`
 - odstranění kontejneru včetně provedených změn
- `docker image ls`
 - výpis stažených image
- `docker image rm image_id`
 - smazání image
 - nelze smazat, pokud existuje kontejner (i zastavený), který image používá

PYTHON



- [web](#)
- interpretovaný jazyk
- dynamická typová kontrola
- podpora různých programovacích paradigmat
 - objektové i funkcionální
- [stažení](#) a [návod](#) pro instalaci na různé OS
- možnost využít také např. [Anacondu](#)

PYTHON

- pip
- package manager pro Python
- instalace balíčku (v příkazové řádce)
 - pip install pymongo
- odinstalace
 - pip uninstall pymongo
- zobrazení nainstalovaných balíčků včetně verze
 - pip list



CVIČENÍ

- opakování – Python
- web scraper pro web [iDNES.cz](https://idnes.cz)
 - navrhnete algoritmus, který bude postupně procházet články webového portálu iDNES.cz
 - ke každému článku uloží název článku, obsah článku, kategorii, počet fotografií, datum publikace a počet komentářů
 - pozor na neúplné údaje
 - informace bude ukládat do textového souboru ve formátu JSON
 - doplňte i funkcionalitu pro načtení dat
 - bude potřebná pro další cvičení
- cílem je stáhnout alespoň 250 MB textových dat
- BONUS: stáhněte minimálně 1 GB textových dat

