

Úvod ku Dockeru

Kamil Janeček

4.2.2020

Kto som ja?

- študent v 2. ročníku inžinierskeho štúdia @ FIIT STUBA
- (vraj) programátor
- <https://kamko.dev>

Kto ste vy?

- študenti?
 - pracujúci študenti?
 - absolventi?
-
- Kto pozná Docker?
 - Kto bežne používa Docker?

Agenda workshopu

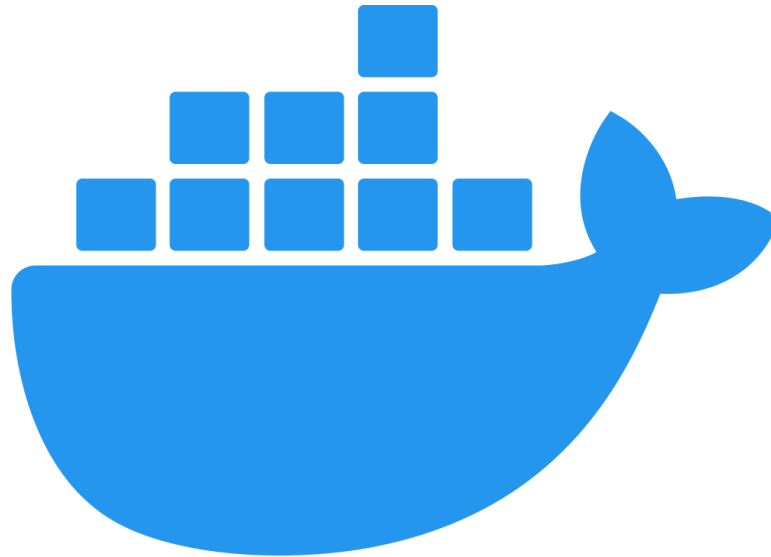
1. Rýchly úvod ku Dockeru
 - High-level prehľad
 - Základné pojmy
 - Kedy (ne)použiť
2. Ukážka ako kontajnerizovať jednoduchú server aplikáciu
 - Jednoduchý server
3. Ukážka ako pracovať s viacerými kontajnermi s Docker Compose
 - Server aj s databázou
4. Tipy a triky

Čo to dnes bude?

- Nevysvetlíme si detailne čo je to Docker
- Vysvetlíme a ukážeme si prečo je Docker fajn
- ak už Docker poznáte tak sa toho dnes asi veľa nového nedozviete

Čo je to ten Docker?

- Firma, ktorá spopularizovala kontajnerizované aplikácie
- Vznikla v 2013



docker®

OK, čo je to docker?

*Docker is a **set of platform as a service** (PaaS) products that use **OS-level virtualization** to deliver software in packages called **containers**.*

[https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

Ako to bolo predtým?

- Fyzický server
 - Pomalé nasadzovanie
 - Drahé
 - Zle škálovateľné
 - ...
- Virtuálne servery
 - Každá virtuálka je stále náročná na zdroje
 - Izolácia medzi aplikáciami
 - Ľahšie škálovateľné

A dñes?



Kontajner

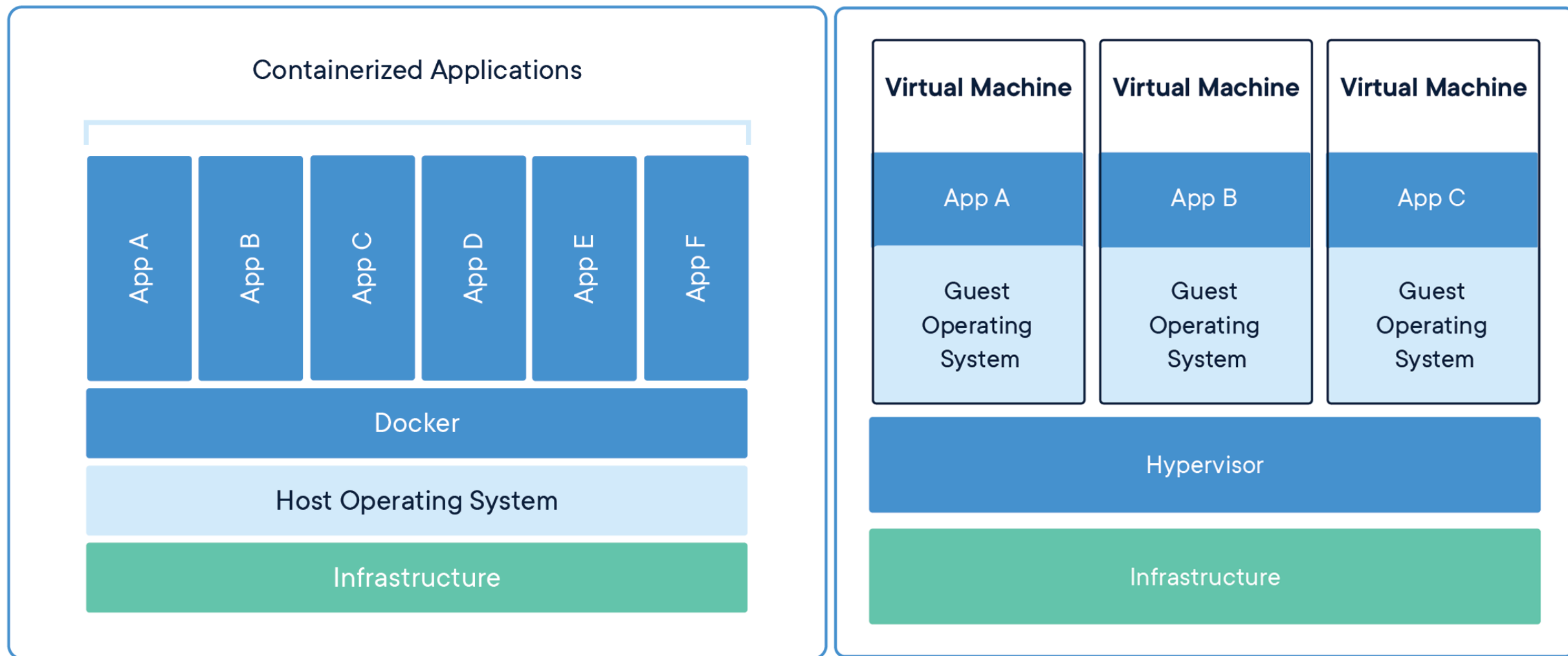
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another

<https://www.docker.com/resources/what-container>

Aké to má výhody?

- Kontajnery ponúkajú podobnú (nižšiu) úroveň izolácie ako VM
- Na rozdiel od VM sú ale oveľa menej náročné na prostriedky
- Oveľa viac aplikácií na jednom stroji
- Jednoduchšia škálovateľnosť (rýchlosť spustenia je neporovnateľná s VM)
- Kontajner beží (takmer) všade rovnako

Kontajnery vs Virtuálne stroje



Ideme po povrchu...

- Tieto porovnania sú zjednodušené
 - je ešte veľa technických detailov okolo kontajnerov
 - nebudeme vysvetľovať ako to funguje na úrovni OS
 - napr. Docker nie je jediný spôsob ako využívať kontajnery – iba najznámejší a najpoužívannejší
- My sa budeme na Docker a kontajnery pozerať z pohľadu **používateľa**
 - Ako programátor
 - Ako administrátor

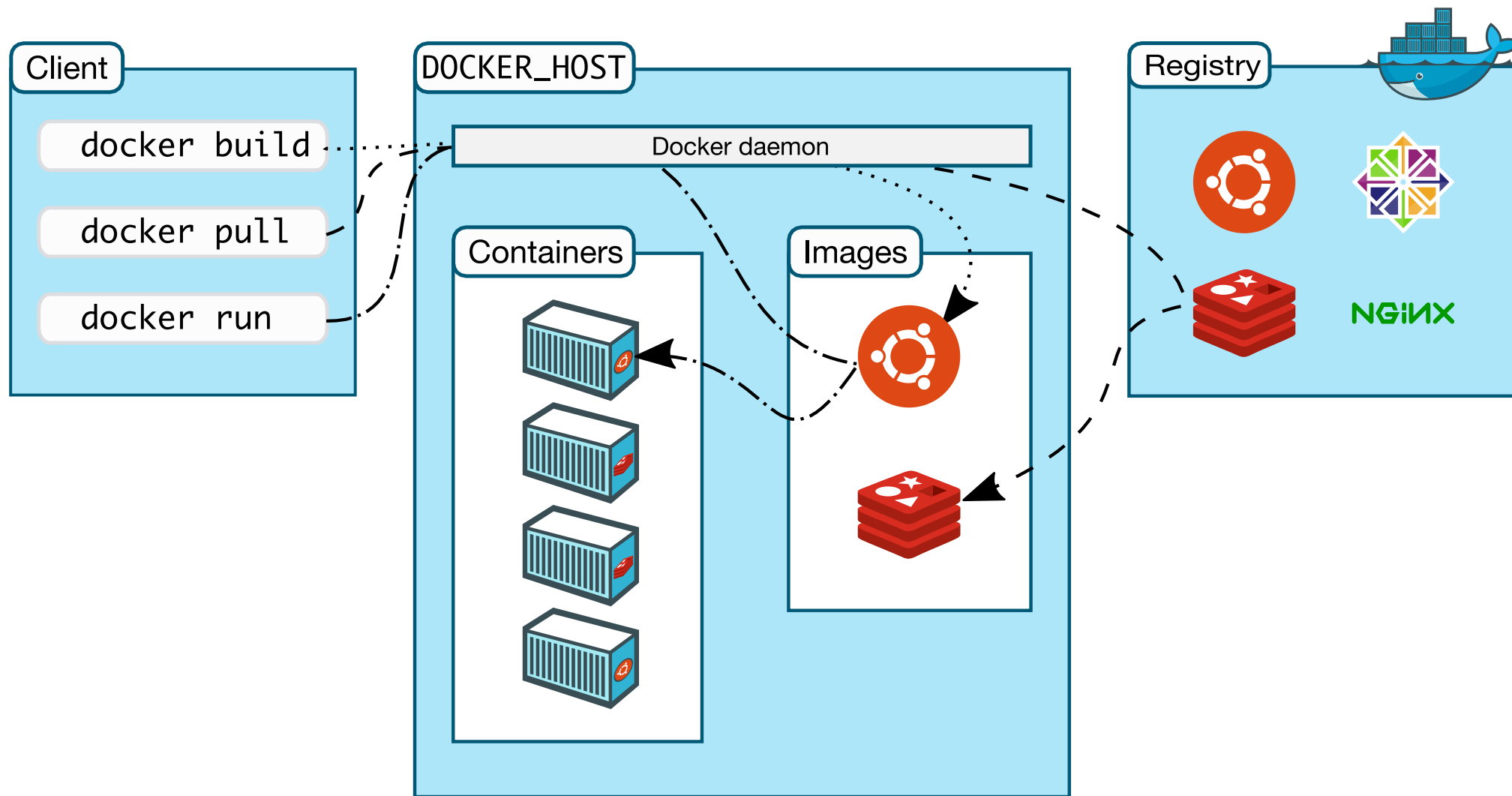
Kontajnery a programátor

- Prostredie pre beh aplikácií obsahujúci iba nutné minimum
- Žiadne konflikty medzi verziami knižníc
- Možnosť upgradovať aplikáciu bez strachu, že sa niečo rozsype
- Jednoduché spustenie rôznych aplikácií
 - Napr. lokálna databáza pre development alebo CI
- Aplikácia sa všade spúšťa a správa rovnako
- Jednotné zdieľanie balíkov

Kontajnery a administrátor

- Nasadzovanie je štandardizované pre akékoľvek aplikácie
- Aplikácia je nezávislá od prostredí
- Štandardizovaný prístup k správe kontajnerov
- Jednoduché spúšťanie nových verzií
- Viac aplikácií na rovnakom stroji oproti VM
- Ľahká škálovateľnosť

Docker architektúra



Základné pojmy

- Image
- Dockerfile
- Container
- Volume
- Registry

Image

- Nemenný vzor obsahujúci inštrukcie pre vytvorenie kontajnera
- Skladá sa z vrstiev
- Vrstvy sa zdieľajú medzi imagmi
- Vytvára sa pomocou *Dockerfile*
- Rozlišujú sa pomocou tagov

Container

- Spustiteľná inštancia konkrétneho imagu
- Možné zosieťovať viacero kontajnerov
- Zdieľať dáta z hosta
- Keď sa zmaže stratia sa všetky zmeny imagu

Dockerfile

- Súbor obsahujúci zoznam krokov potrebných pre vytvorenie nejakého Docker image
- Môže rozširovať už existujúci image

```
1  FROM nginx:1.17.6
2
3  WORKDIR /tmp
4
5  COPY template.nginx .
6  COPY entrypoint.sh .
7
8  CMD ["bash", "entrypoint.sh"]
```

Registry

- Úložisko Docker imagov
- Najväčší je **Docker Hub**
 - Ale je veľa alternatív
- Uploadnutím imagu do registry je možné sa k nemu dostať odkiaľkoľvek
- Napr. nginx
 - https://hub.docker.com/_/nginx

Volume

- Kontajnery nie sú stavané na ukladanie dát
- Volume slúži na ukladanie dát
- Manažované interne Dockerom
- Možné zdieľať dáta medzi kontajnermi
- Perzistencia dát

Bind mount

- Umožňuje zdieľať priečinok medzi hostom a kontajnerom

Kde môžete využiť kontajnery

- Nasadzovanie vašej aplikácie na server
- Lokálna inštancia napr. databázy
- Potrebujete nejakú utilitu (napr. db klient) a nechcete si ju inštalovať?
- Otestovať niečo na konkrétnej verzii systému alebo s konkrétnymi knižnicami

Kedy nepoužiť kontajnery

- GUI (desktop) aplikácie
- Ak je potrebná čo najvyššia rýchlosť
- Ak je existujúca aplikácia zložitá a ťažko dockerizovateľná (resp. to nemá kto robiť)
- Ak je bezpečnosť jedna z top priorít

Zdroje

- <https://docs.docker.com>
- <https://www.docker.com/resources>
- <https://docker-curriculum.com>
- <https://pointful.github.io/docker-intro>

Hands-on

Wifi:

Campus

2spaces1community

<https://github.com/zanasufiit/ws1-docker-hands-on>

Tipy a triky

- Používajte linter pre Dockerfile
 - Napr. <https://github.com/hadolint/hadolint>
- Pridajte si auto-complete do terminálu
- Ak máte radi GUI tak môžete použiť napr. portainer
- Minimalizujte počet a veľkosť jednotlivých vrstiev
 - Napr. ak inštalujete nové balíky tak zmažte cache
- Súbor, ktoré majú tendenciu sa meniť kopírujte čo najneskôr
- Využívajte restart policies
- Docker Compose

Tipy a triky

- Alpine base image – malá veľkosť ale za cenu inej libc
- Distroless – malá veľkosť a maximálna bezpečnosť
 - <https://github.com/GoogleContainerTools/distroless>
- **Vždy** používajte fixný tag
- Používajte .dockerignore
- Multi-stage build napr. pre SPA aplikácie
 - <https://docs.docker.com/develop/develop-images/multistage-build/>
- Ak to je možné zmeniť usera z roota
- 1 process per kontajner? (kedy ano?)