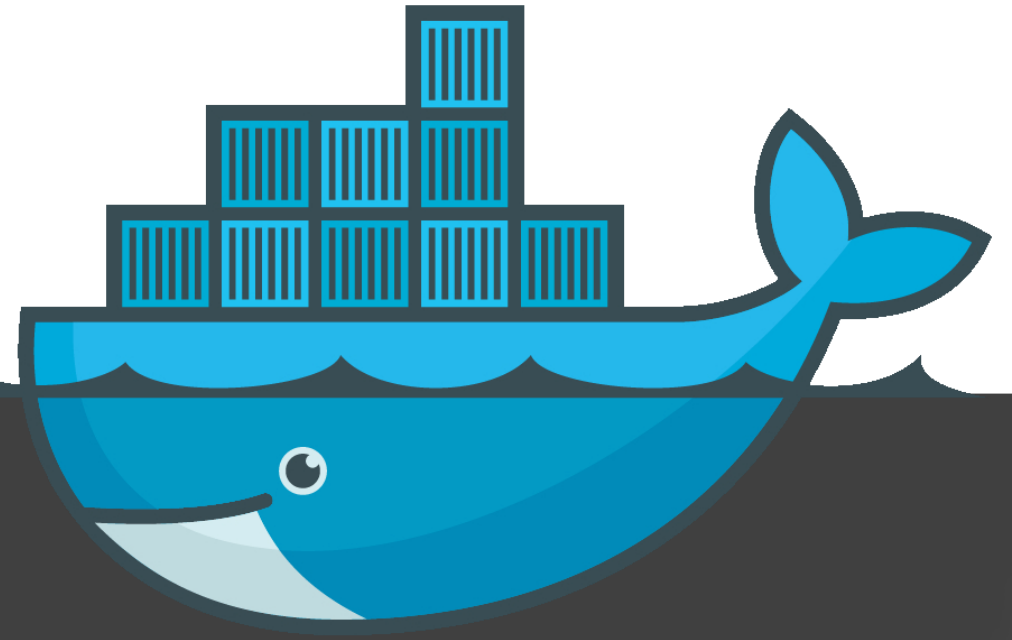


Urychlete svůj vývoj!



docker

Michal Macinka

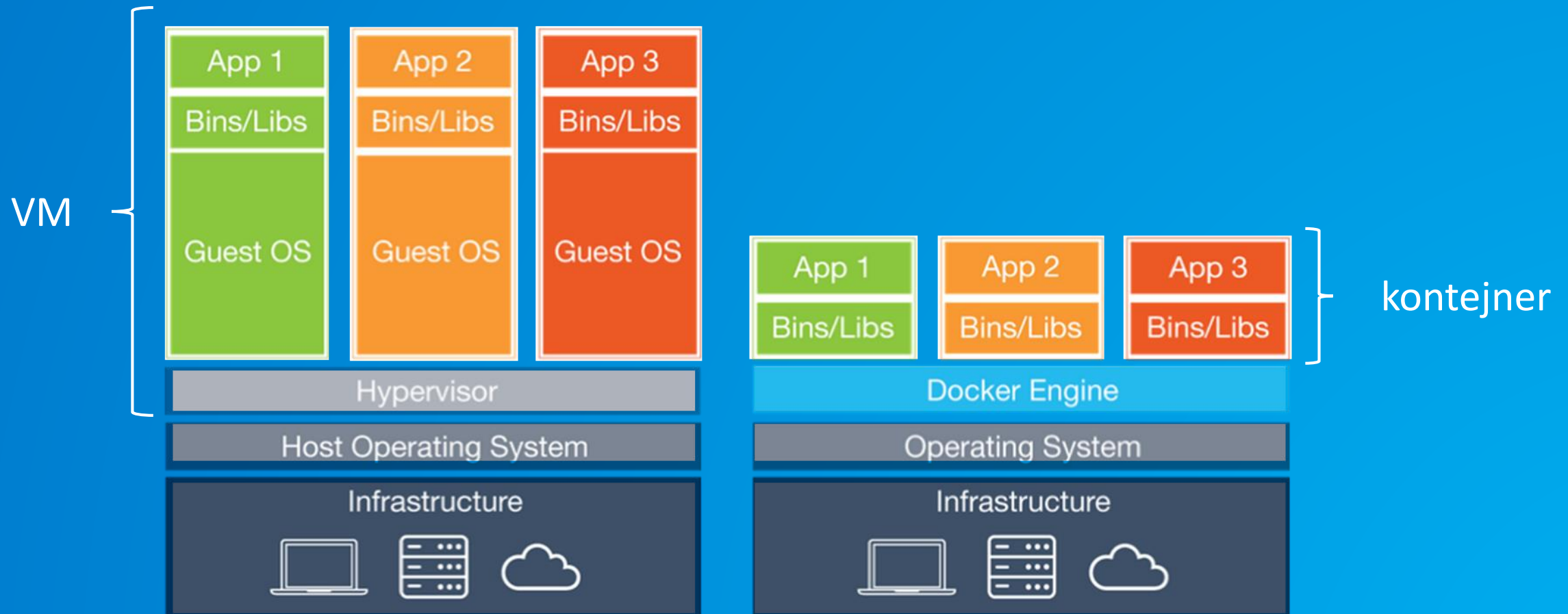
ai2

„U mě to fungovalo.“

Proč všichni chtějí Docker?

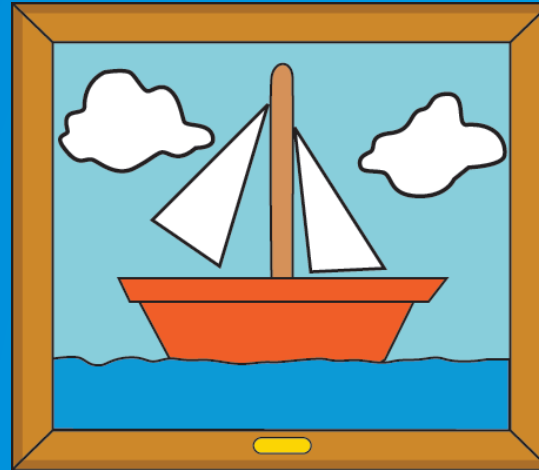
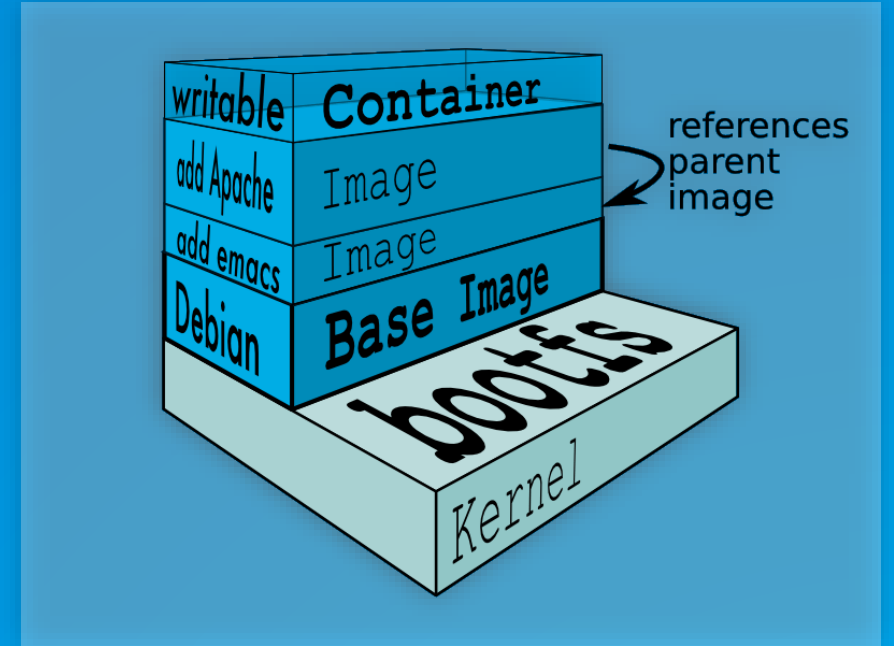
- jednoduchost
- nenáročnost
- je to přece moderní a mají to všichni

Kontejnerizace vs virtualizace



Trocha teorie...

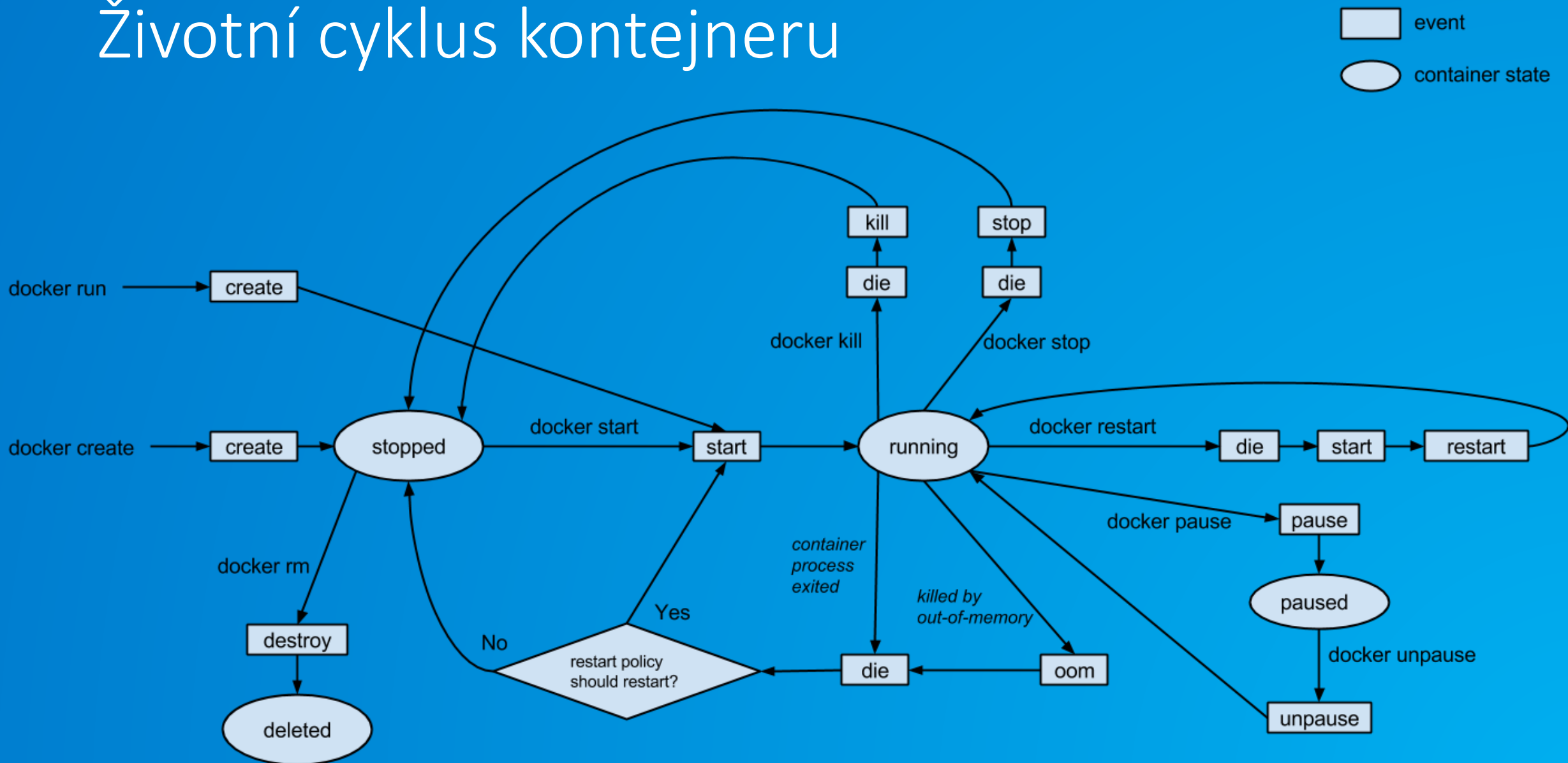
- vrstva
- obraz
- kontejner



Příkazů je vcelku hodně..

- `docker ps ...`
- `docker commit ...`
- `docker rm ...`
- `docker images ...`
- `docker stats ...`
- `docker logs ...`
- `docker swarm ...`
- `docker service ...`
- `docker exec ...`
- `docker cp ...`
- `docker attach ...`
- `docker rmi ...`
- `docker kill ...`
- `docker start`

Životní cyklus kontejneru



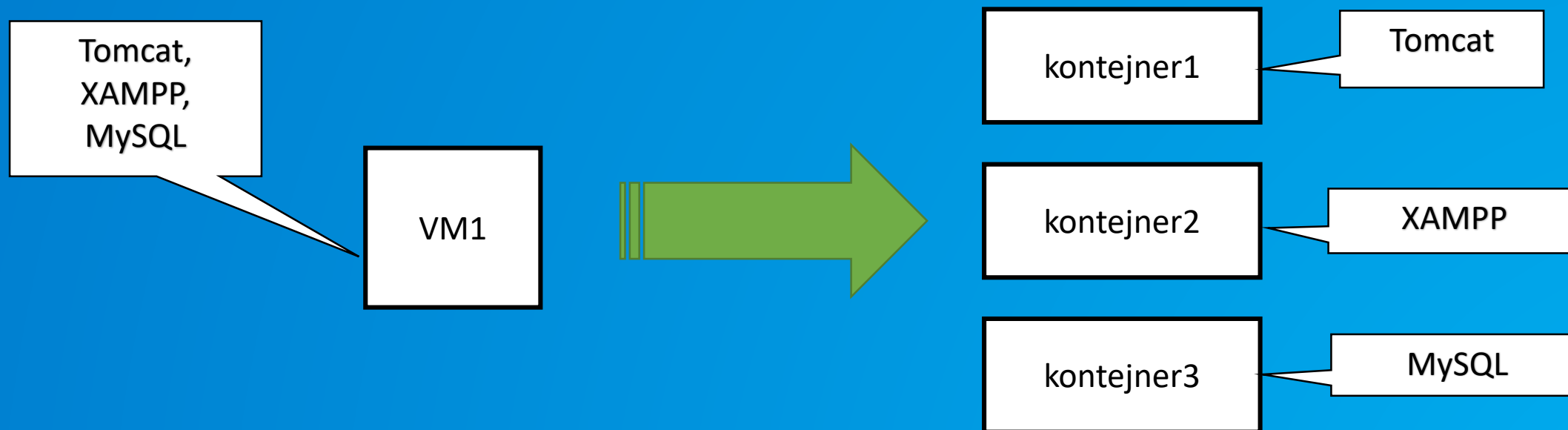
Co Docker rozhodně není

- nástupce virtualizace

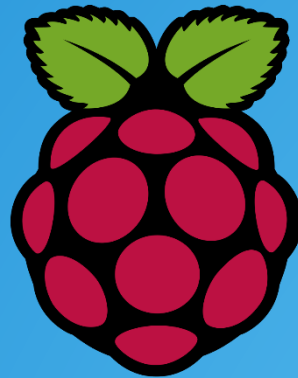


Jak by to mělo vypadat

- kontejner per proces



Kde provozovat?



Instalace Dockeru

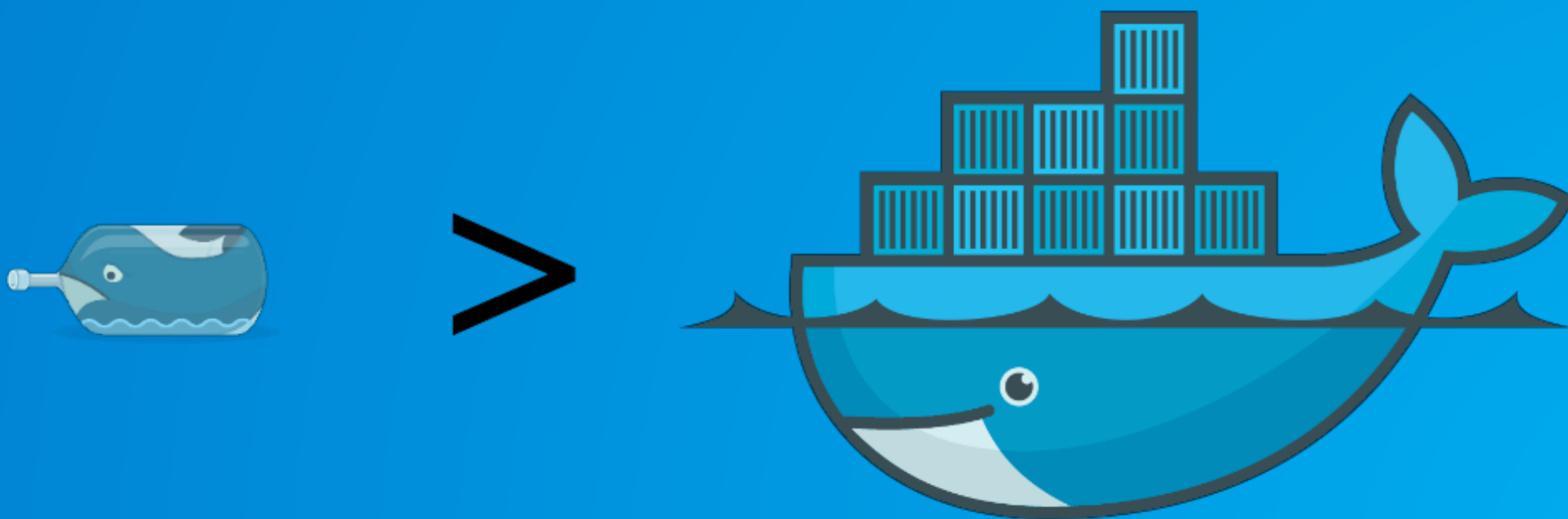
- <https://goo.gl/ERVXr4>

Dockerfile

- taková kuchařka pro vytvoření obrazu
- každý příkaz znamená jednu vrstvu

Proč je vytvářet?

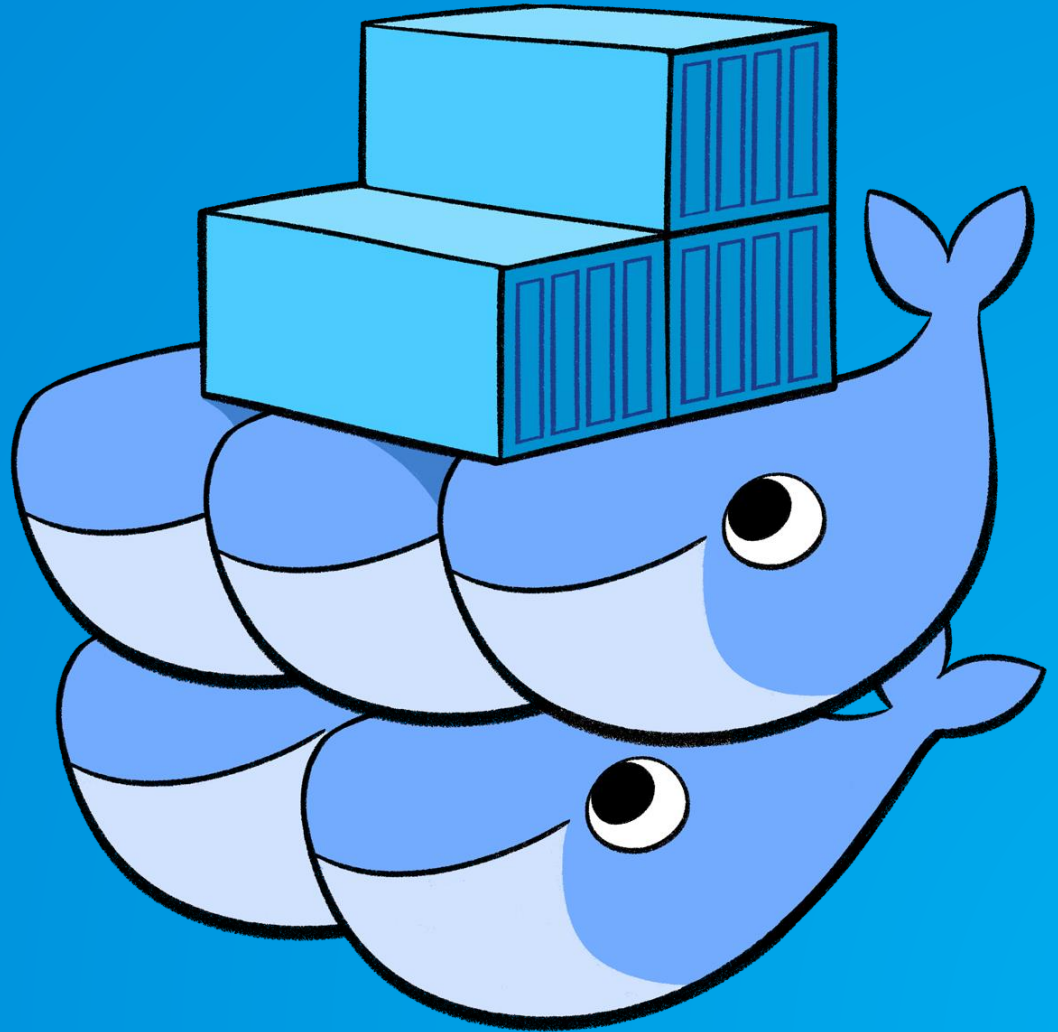
- máme obraz pod kontrolou
- výsledná velikost obrazu
- např. oficiální obraz pro Node.js = 643 MB





Velrybí cluster aneb Docker Swarm

- vytváříme služby
- škálování
- load balancing



Monitoring a řešení problémů

- docker logs
- docker events&
- docker exec
- docker cp

Správa pomocí API

- lokální i vzdálená správa pomocí HTTP Requestů
 - `--unix-socket /var/run/docker.sock`
 - povolení remote portu

Nástroje pro jednodušší správu

- Portainer - <http://portainer.io/>
- Shipyard - <https://shipyard-project.com/>
- Kubernetes - <https://kubernetes.io/>

A jak to máme na FIMu?

- kontejnery v kontejnerech
- 2 násobný port-forwarding

Něco nebylo jasné?