Kontejnerové technologie

Jindřich Káňa
ELOS Technologies s.r.o.

Agenda

Historie kontejnerových technologií

Docker

OpenShiftu

CloudForms

Historie

Nová funkce v UNIXu

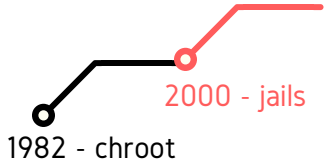
Systémové volání

Izolace na úrovni procesu

UNIX CHROOT

Historie

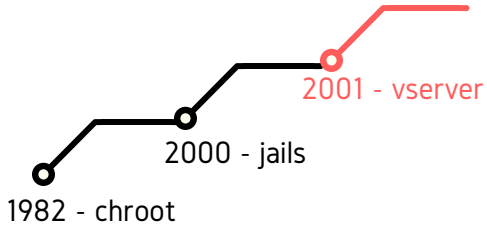
Představení izolace na úrovni souborového systému, paměti, sítě a izolace oprávnění pro uživatele root



FREEBSD JAIL

Historie

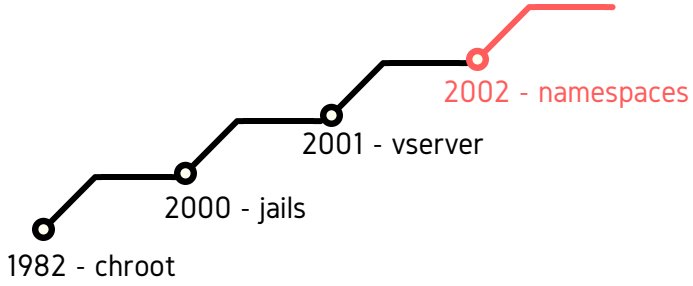
Linux VServer patch přinesl první, základní principy kontejnerů do světa Linuxu.
(Security Context pro paměť, CPU, síť)



LINUX VSERVER

Historie

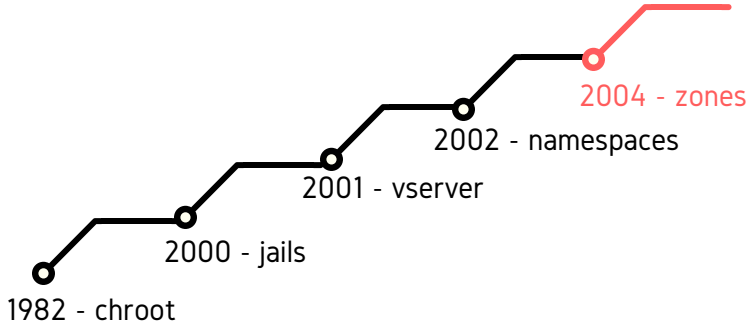
vzniká v kernelu 2.14.19 první namespace mount



LINUX NAMESPACE

Historie

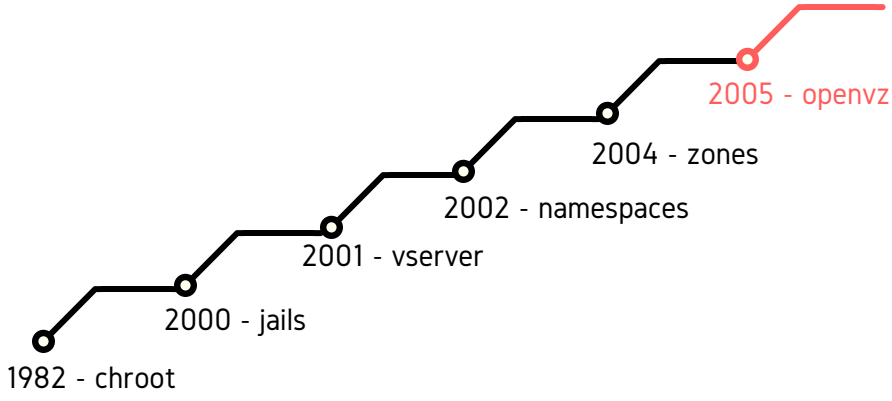
Sun Microsystems představuje kontejnery pro Solaris UNIX (zones)



SOLARIS CONTAINER

Historie

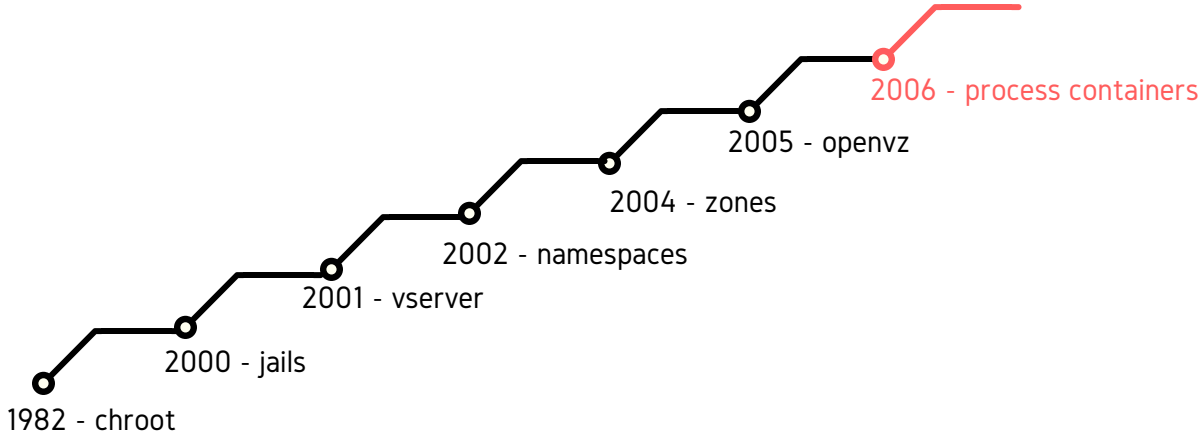
Linux kernel patch ke správě fyzických zdrojů
pro běh a izolaci aplikací v kontejnerech



OPEN VZ

Historie

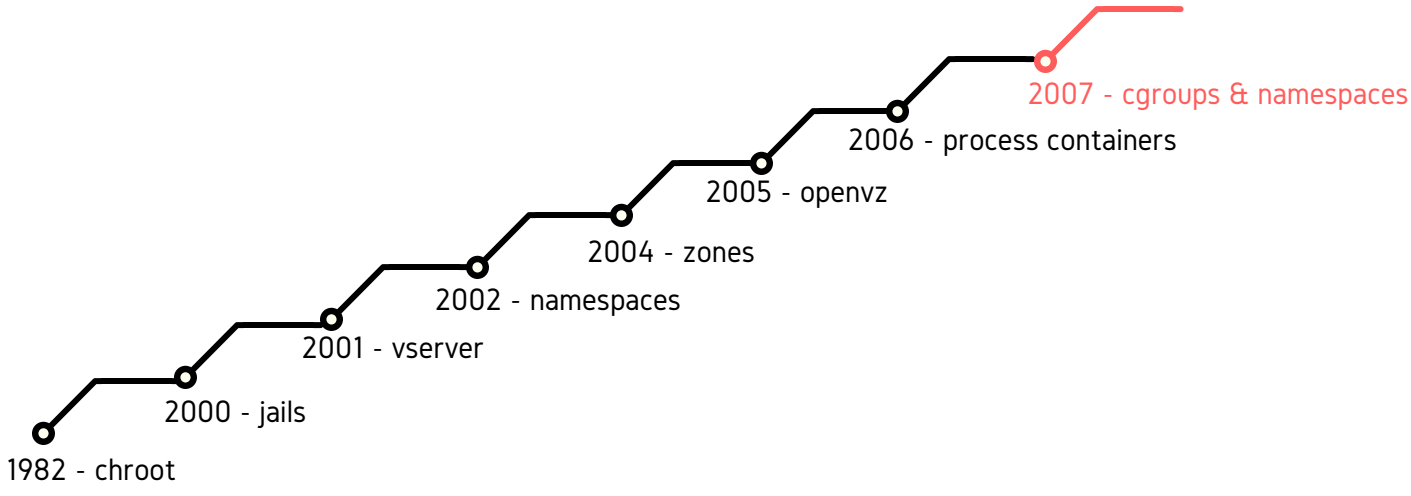
Google vyvíjí ‘Process Containers’ pro Linux,
který pak přechází v CGroups.



PROCESS CONTAINERS

Historie

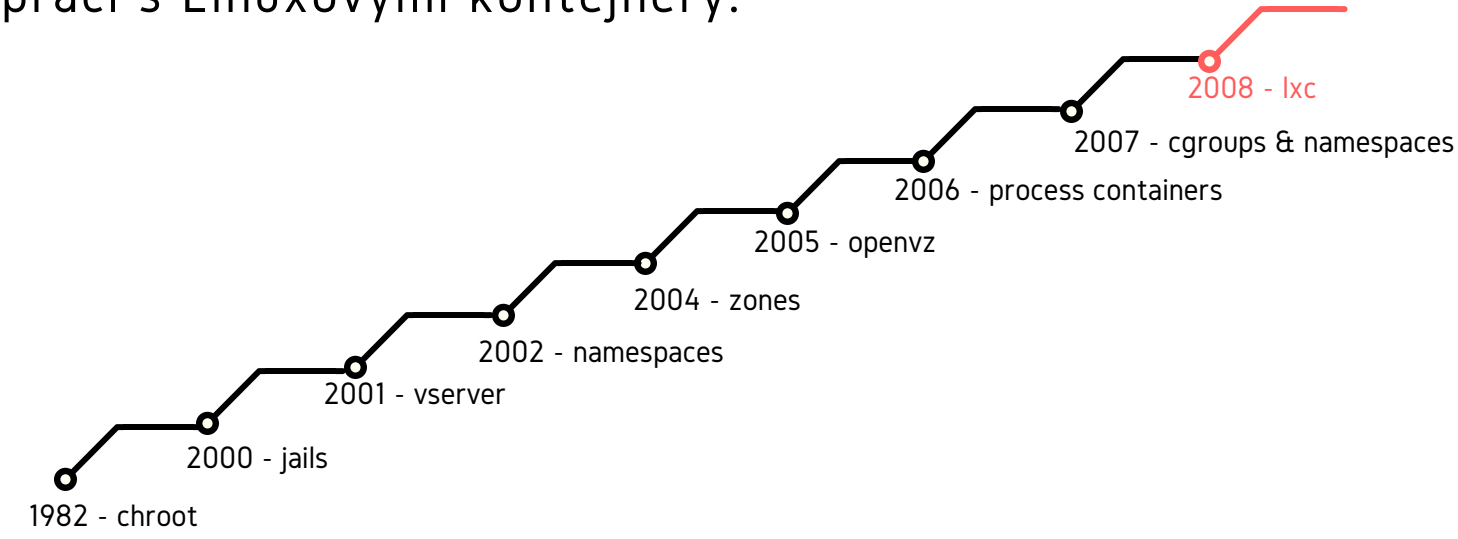
Control Groups a Namespaces byly portovány do Linux kernelu.



CGROUPS & NAMESPACES

Historie

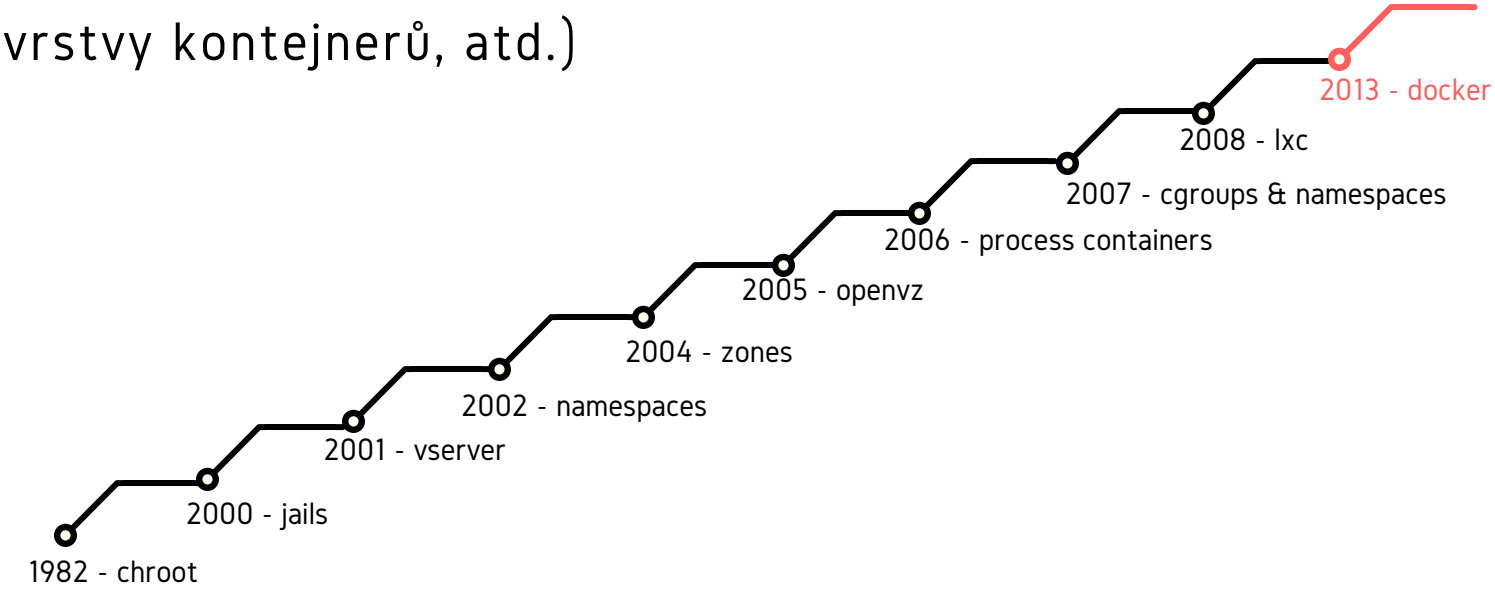
IBM představuje uživatelské nástroje pro práci s Linuxovými kontejnery.



L X C

Historie

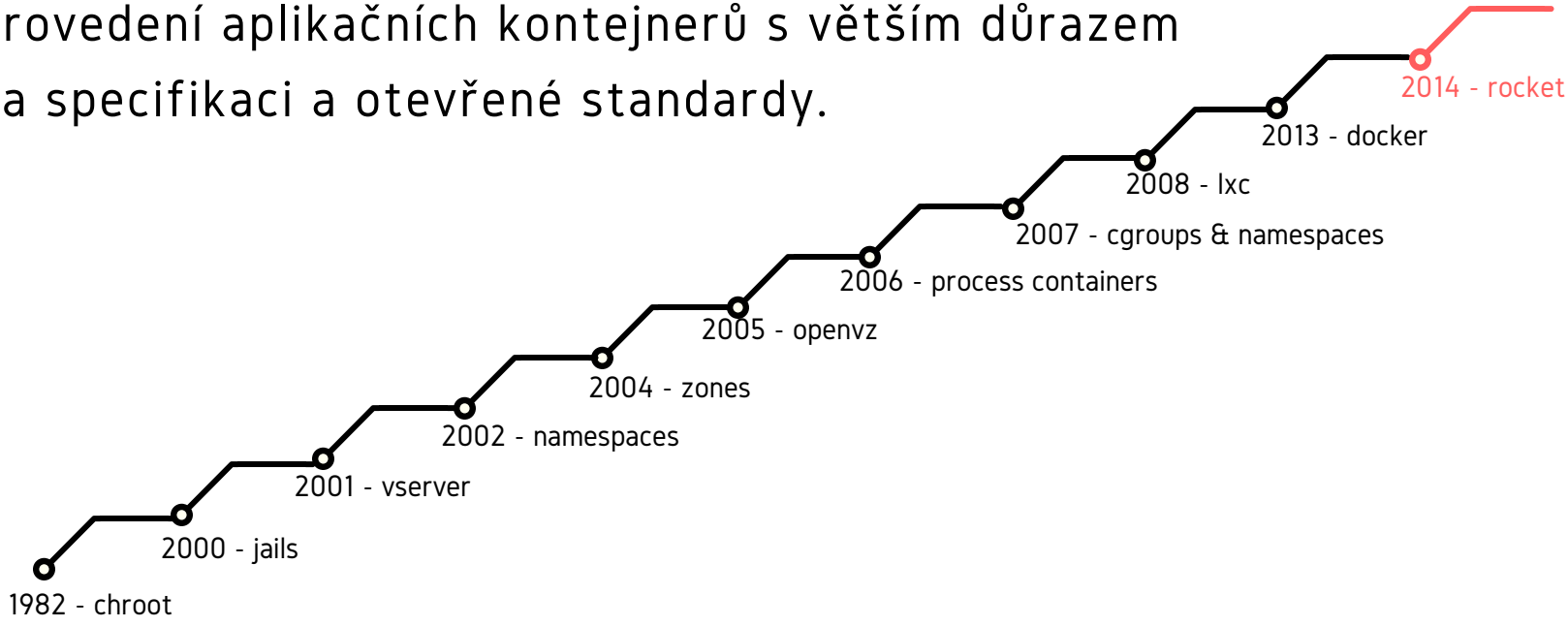
z počátku na bázi LXC, později představuje svou knihovnu libcontainer a postupně přibývá celý ekosystém (REST API, CLI, registry obrazy, vrstvy kontejnerů, atd.)



DOCKER

Historie

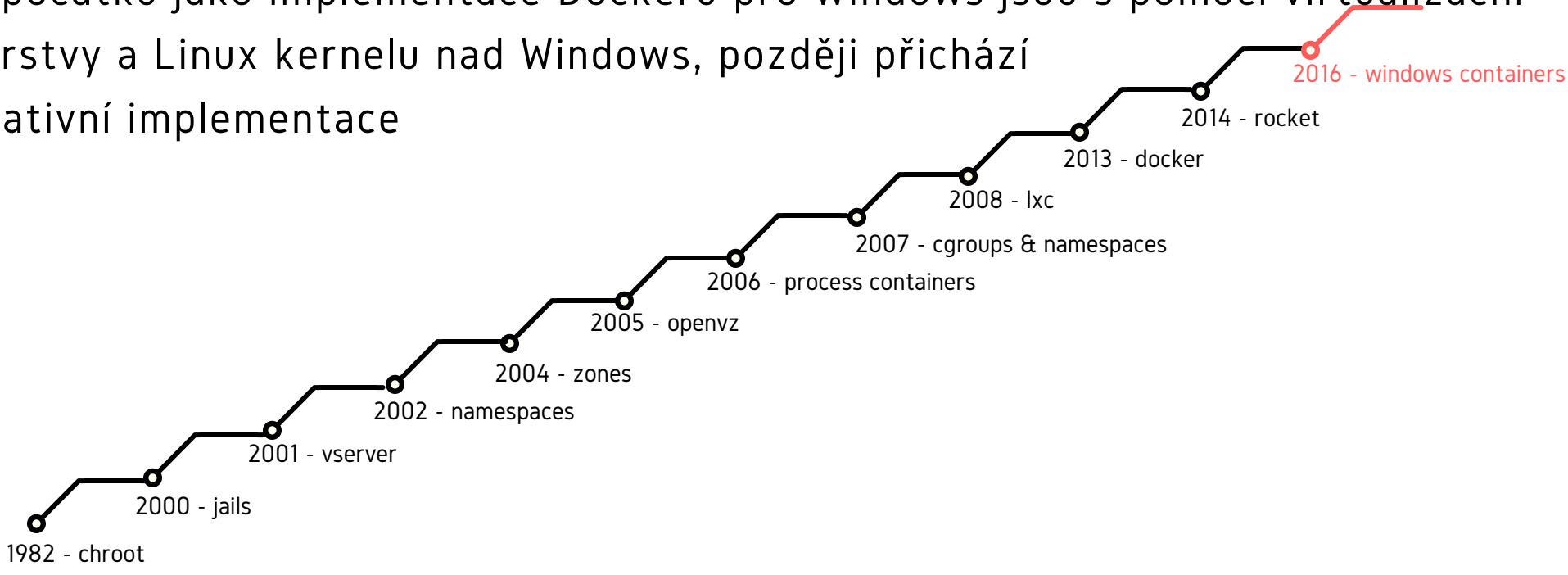
velmi podobná technologie jako Docker, soustředí se ale na čistější provedení aplikačních kontejnerů s větším důrazem na specifikaci a otevřené standardy.



ROCKET

Historie

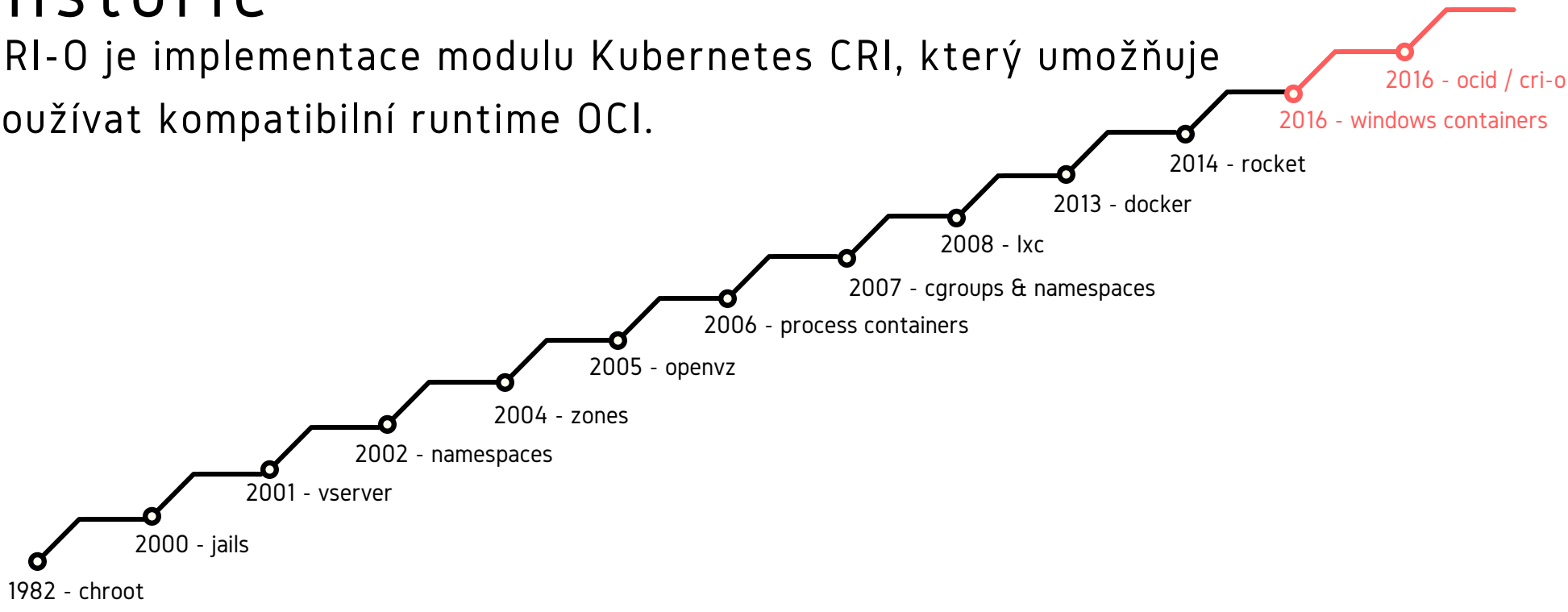
zpočátku jako implementace Dockeru pro Windows jsou s pomocí virtualizační vrstvy a Linux kernelu nad Windows, později přichází nativní implementace



WINDOWS CONTAINERS

Historie

CRI-O je implementace modulu Kubernetes CRI, který umožňuje používat kompatibilní runtime OCI.



C R I - O

Co je to dnešní Linuxový kontejner?

Obvyklý Linuxový proces běžící v operačním systému s omezeným pohledem na HW zdroje.

Linux namespaces
control groups
union filesystem
iptables
Linux capabilities

Linux namespaces - typy

namespace	kernel	vydání
mount	2.4.19	srpen 2002
ipc	2.6.19	listopad 2006
uts	2.6.19	listopad 2006
pid	2.6.24	leden 2008
network	2.6.24	leden 2008
user	3.8	únor 2013
cgroups	4.6	květen 2016

Linux namespaces

```
[jindrich.kana@jindrovo ~]$ ps axu|grep thunar
jindric+ 13840  0.5  0.1 507324 28248 ?        Sl   06:08   0:00 thunar
jindric+ 14052  0.0  0.0  10708  1004 pts/0    S+   06:10   0:00 grep --color=auto thunar
[jindrich.kana@jindrovo ~]$ ls -lah /proc/13840/ns/
total 0
dr-x--x--x. 2 jindrich.kana jindrich.kana 0 Jun  4 06:08 .
dr-xr-xr-x. 9 jindrich.kana jindrich.kana 0 Jun  4 06:08 ..
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 cgroup -> cgroup:[4026531835]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 ipc -> ipc:[4026531839]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 mnt -> mnt:[4026531840]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 net -> net:[4026532009]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 pid -> pid:[4026531836]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 pid_for_children -> pid:[4026531836]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 user -> user:[4026531837]
lrwxrwxrwx. 1 jindrich.kana jindrich.kana 0 Jun  4 06:08 uts -> uts:[4026531838]
[jindrich.kana@jindrovo ~]$
```

Linux namespaces

unshare - vytvoření namespace

vstup do namespace
sudo nsenter -t PID --mount --uts --ipc --net --pid

sudo nsenter -t $(docker inspect --format '{{ .State.Pid }}' $(docker ps -lq))
-m -u -i -n -p -w

man 7 NAMESPACES
man 1 NSENTER
man 1 unshare

Linux cgroups - resource control

```
mkdir /sys/fs/cgroups/cpuset/skupina
echo 2 > /sys/fs/cgroups/cpuset/skupina/cpuset.cpus
echo $$ > /sys/fs/cgroups/cpuset/skupina/cgroup.procs

mkdir /sys/fs/cgroups/memory/skupina
echo 10000000 > /sys/fs/cgroups/memory/skupina/memory.limit_in_bytes
echo $$ > /sys/fs/cgroups/memory/skupina/cgroup.procs
```

```
man CGROUPS
```

Linux capabilities

detailní kontrola oprávnění pro superuživatele

man CAPABILITIES

Docker úvod

Vzniká v dotCloud Inc (cloudControl), později Docker, Inc.

Používá LXC, později přechod na libcontainer

Později opencontainers/runc

Standardizace Linux kontejnerů pod Linux Foundation

Open Container Initiative (OCI)

Docker klíčové vlastnosti

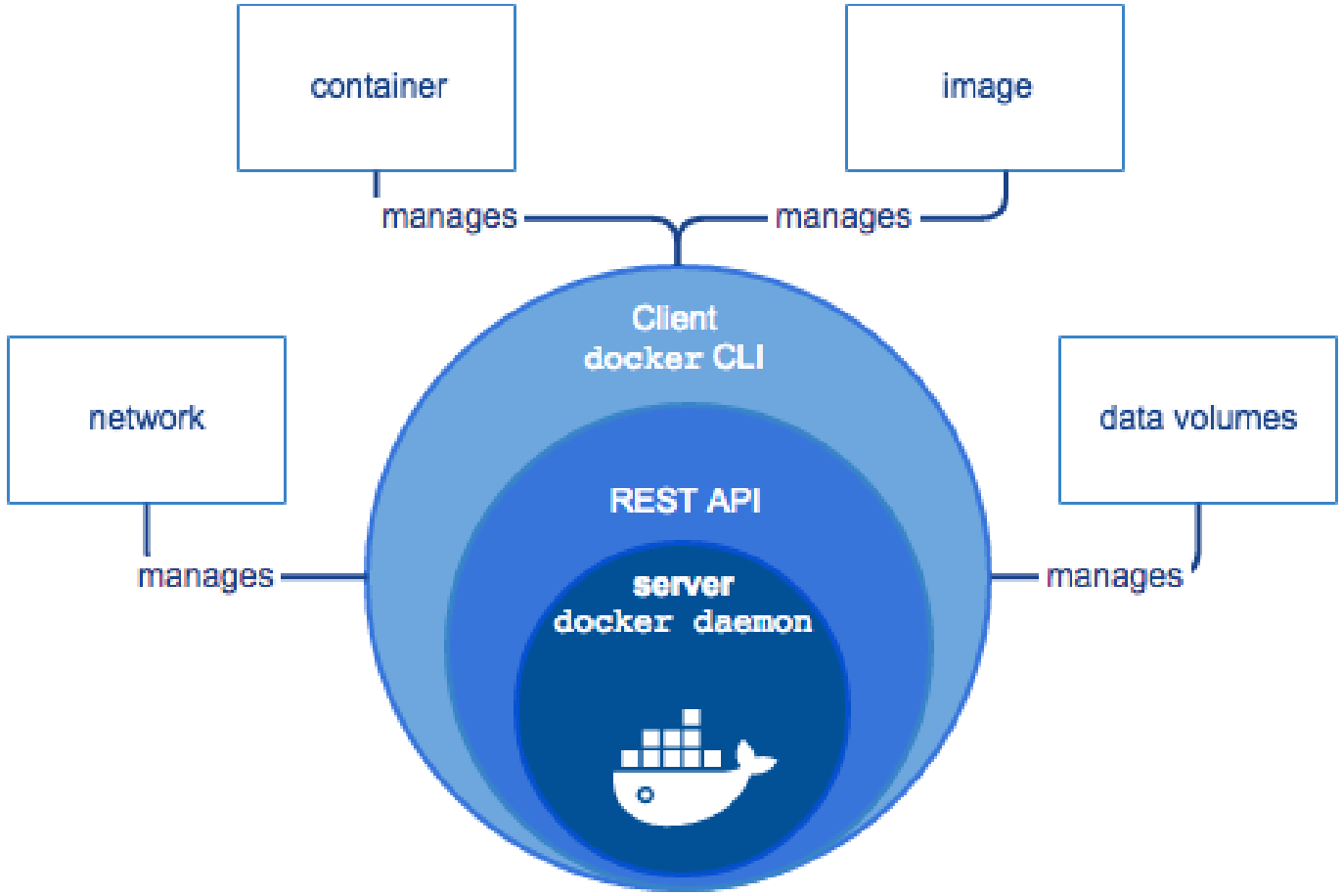
Lokální vývoj a test

Týmová spolupráce

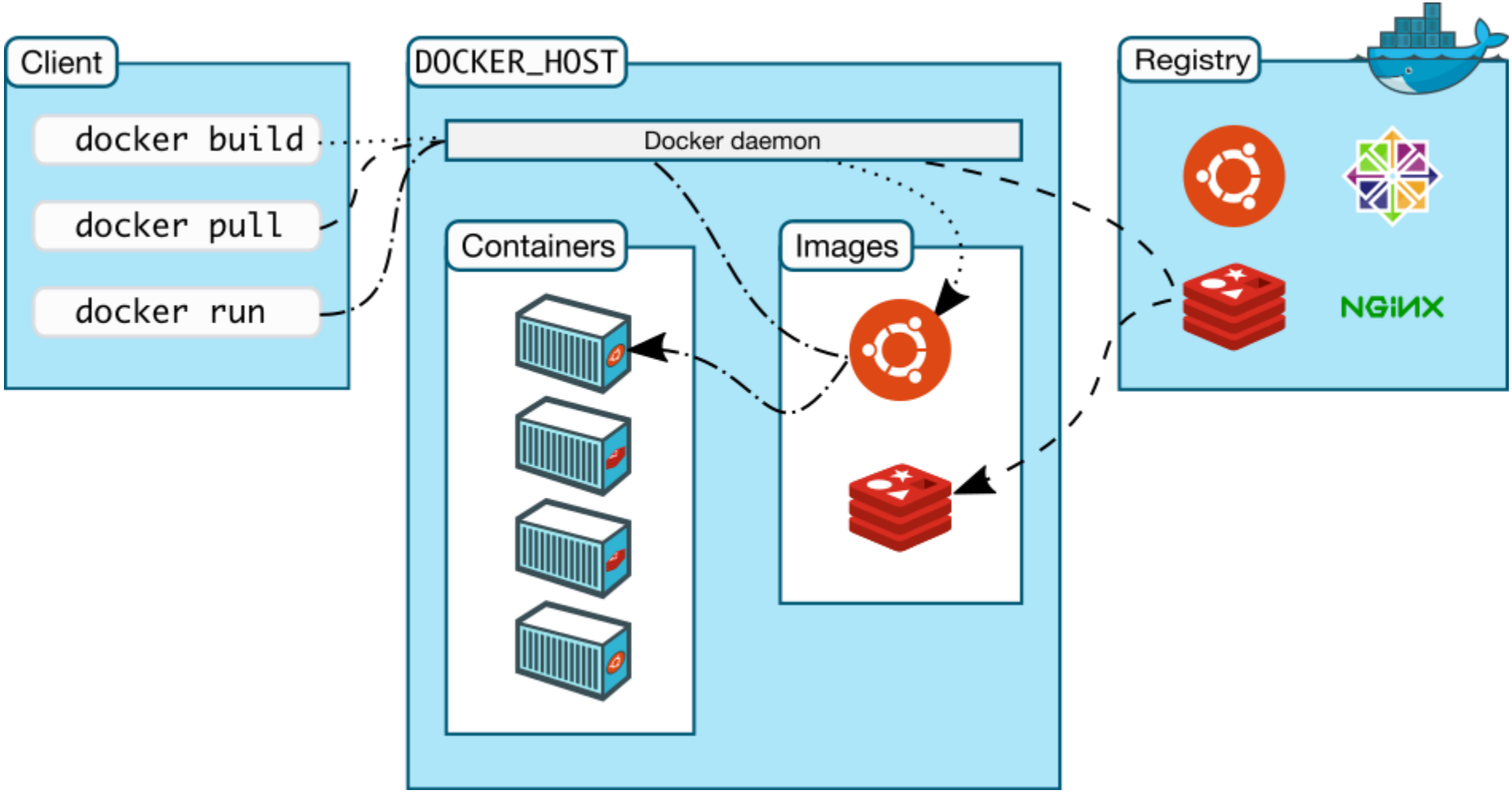
Continuous Integration

Docker -> PaaS

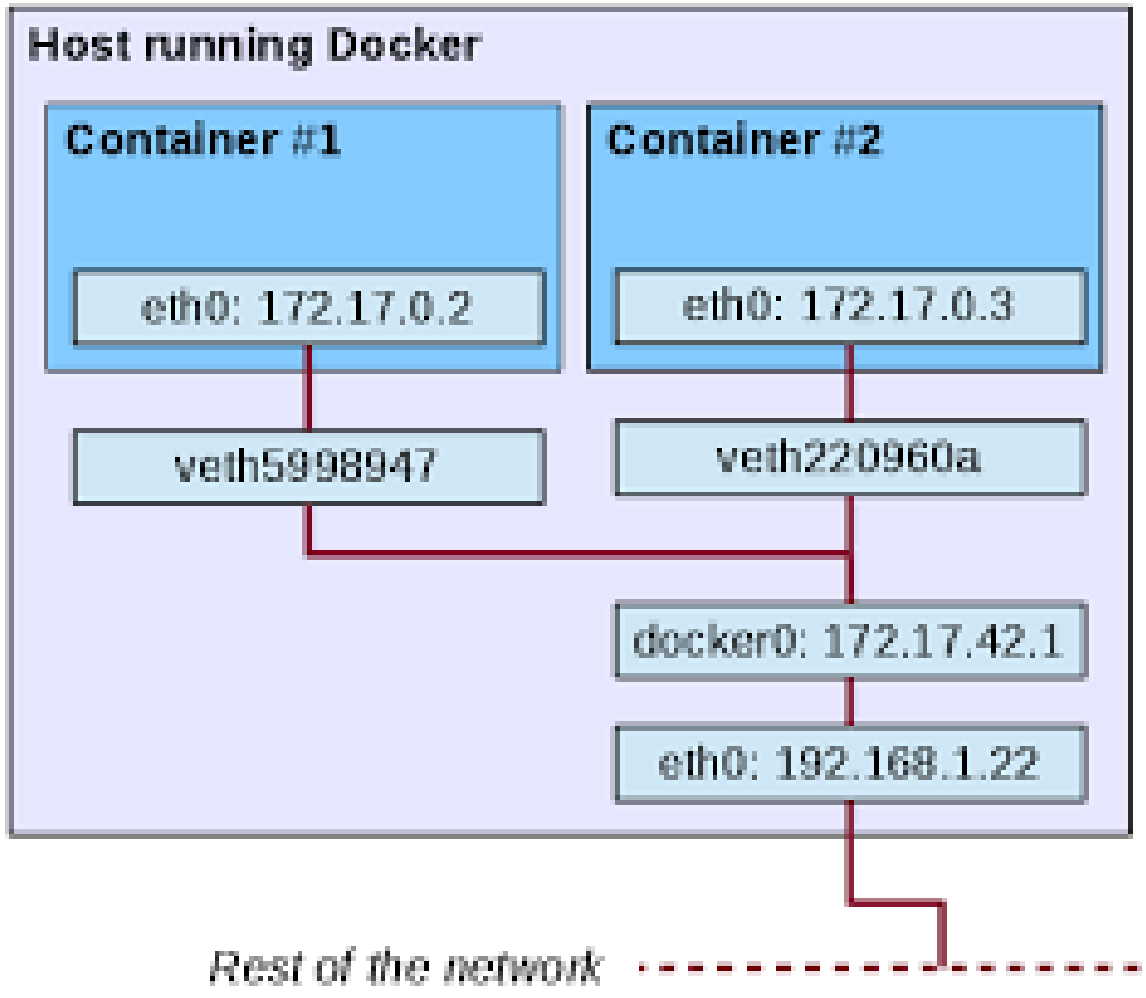
Docker engine



Docker architektura



Docker - network



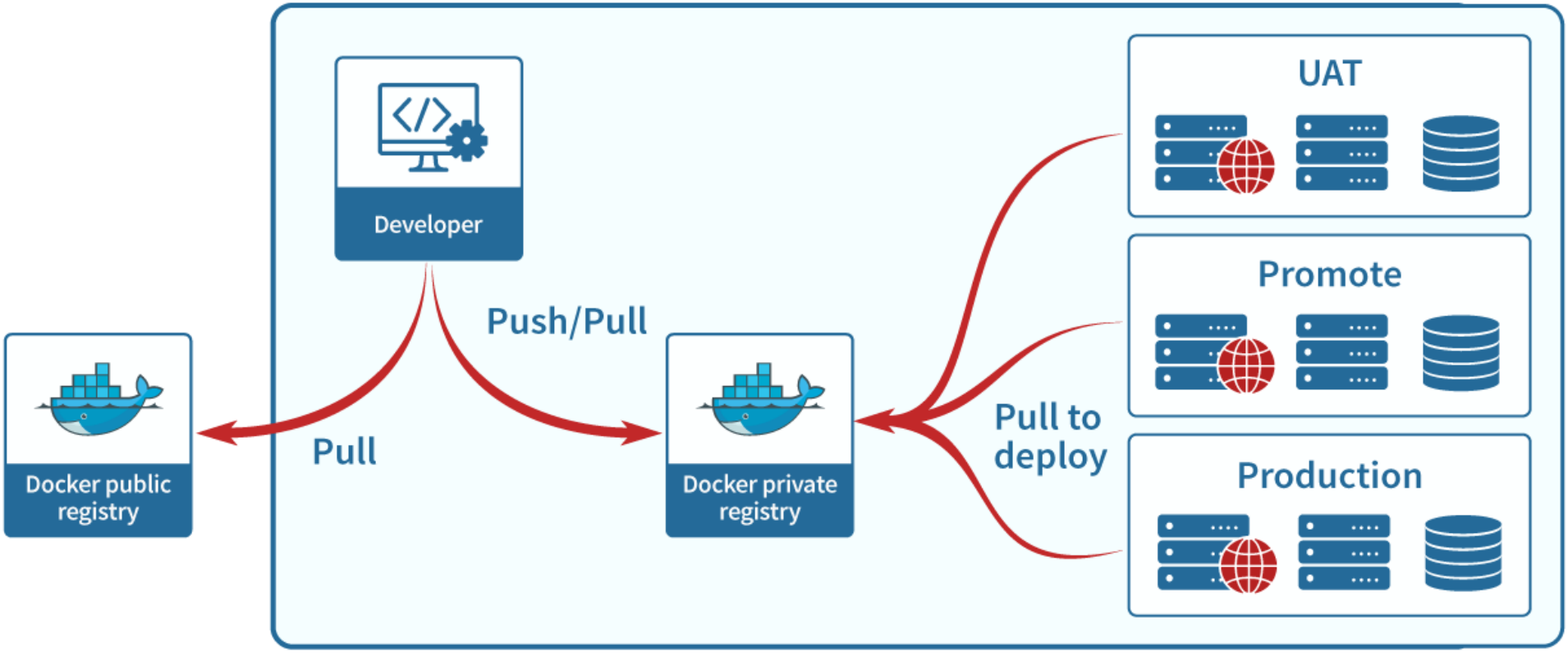
Docker - storage

Linux distribution	Recommended storage drivers
Docker CE on Ubuntu	`aufs`, `devicemapper`, `overlay2` (Ubuntu 14.04.4 or later, 16.04 or later), `overlay`, `zfs`, `vfs`
Docker CE on Debian	`aufs`, `devicemapper`, `overlay2` (Debian Stretch), `overlay`, `vfs`
Docker CE on CentOS	`devicemapper`, `vfs`
Docker CE on Fedora	`devicemapper`, `overlay2` (Fedora 26 or later, experimental), `overlay` (experimental), `vfs`

/etc/sysconfig/docker-storage-setup

/etc/sysconfig/docker-storage

Docker - Registry



docker info

/etc/containers/registries.conf

Docker - CLI

nejčastěji užívané příkazy

docker info
docker images
docker ps
docker run
docker exec
docker save

příklad vyhledávání specifické property, lze také grepovat
docker inspect ContainerName --format '{{ .NetworkSettings.Networks.bridge.IPAddress }}'

vyhledání manuálů pro docker
mandb - aktualizace manuálových stránek

man -k docker

stránky projektu
https://docs.docker.com/