

# Začínáme s kontejnery

Jak na vlastní Containerfile?  
Co je to podman?



Lukáš Kotek  
Senior Software Quality Engineer  
Red Hat

# Kdo jsem?

- Software Quality Engineer ve firmě Red Hat
  - Pracuji na Continuous Integration (CI) a poskytuji různorodou podporu pro automatizované testování v oblasti [virtualizace](#)
- Čemu se věnuje Red Hat?
  - Komerční linuxová distribuce [Red Hat Enterprise Linux](#) (RHEL)
  - [OpenShift](#) (pro orchestraci linuxových kontejnerů)
  - Automatizační nástroj [Ansible](#) (Configuration-as-Code)
- Nadšenec do Open Source
  - Uživatel [Fedora Linuxu](#) od verze 5 :-)

# Virtualizace vs. kontejnery

## ● Virtualizace

- Virtualizuje se celý operační systém včetně jádra
- Pracujeme s tzv. virtuální strojem, Virtual Machine (VM)
  - HW může být emulován ([QEMU](#)), nebo zpřístupněn přes [KVM](#)
  - Termíny host (hostitel) a guest (host)

## ● Kontejnerizace

- V kontejneru se nachází pouze to, co tam chceme mít
  - Aplikace a její knihovny (závislosti), operační systém (...)
- Kontejner **neobsahuje jádro OS** → **závisíme na jádru hostitele**

# Proč použít kontejnery

## ● Jednoduchost

- Kontejner může být definován jedním **textovým souborem**
- Snadná cesta pro distribuci aplikace

## ● Škálovatelnost

- Lze mít mnoho instancí jedné image

## ● Rychlosť

- Není třeba žádný virtuální stroj

## ● Bezpečnost (?)

- Izolace od hostitele (?)



# Pojmy pro dnešek

## ● Image vs. kontejner

- **Image:** předpřipravený obraz, může být založený na dalším obrazu
  - Dostupné z veřejných registrů, např. [quay.io](#) nebo [hub.docker.com](#)
- **Kontejner:** konkrétní spuštěná instance image

## ● Podman

- Nástroj pro práci s kontejnery (kompatibilní s [Dockerem](#))
  - Nepotřebuje běžící službu v rámci OS, ani rootovské oprávnění

## ● Containerfile

- Soubor popisující, co má kontejner obsahovat
  - Platí pro něj stejná pravidla jako pro Dockerfile

# První kontejner I.

1) Nainstalujte podman:

```
dnf install podman
```

2) Zkontrolujte konfiguraci Fedory pro přístup do registru:

```
/etc/containers/registries.d
```

3) Stáhněte si fedora image (který registr je použit?):

```
podman pull fedora:43
```

4) Ověřte, že image je přítomná na lokálním systému:

```
podman images [fedora]
```

```
podman inspect fedora:43
```

5) Spusťte kontejner používající tuto image:

```
podman run fedora:43
```

# První kontejner II.

- Ověřte, co se stalo, pomocí:

podman ps -a

- Čím se liší následující příkazy?

podman run fedora:43

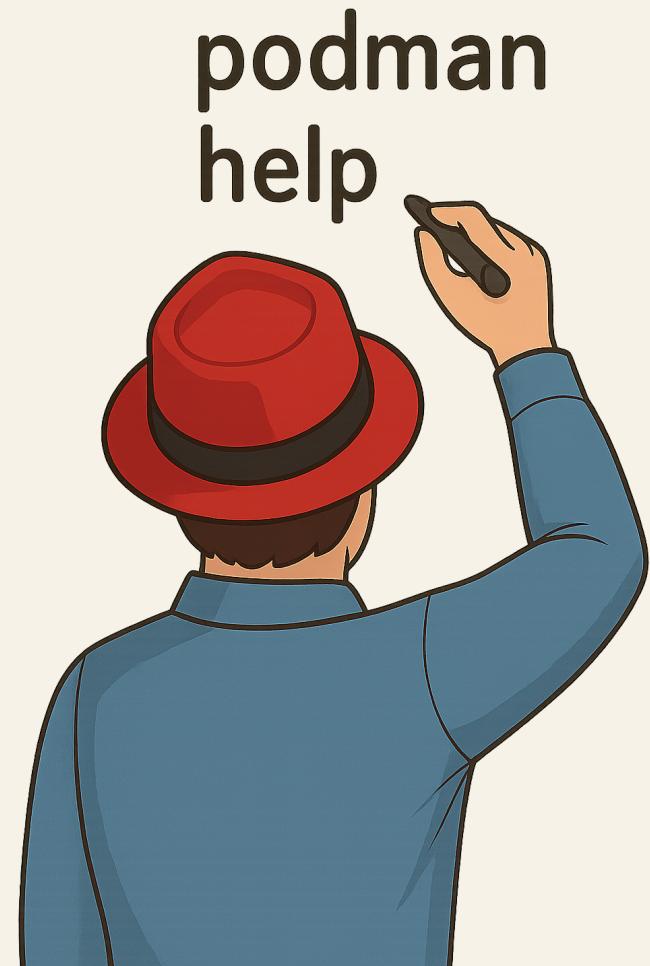
podman run -it fedora:43

podman run -d fedora:43

podman run -dit fedora:43

- Co je to ENTRYPPOINT / CMD?

- A jsou provedené změny trvalé?



# Tvoříme Containerfile

**1) Naklonujte gitový **repozitář** s podklady pro workshop:**

```
git clone https://github.com/lkotek/trebesin-workshop
```

**2) Přepněte se do adresáře obsahující Containerfile:**

```
cd trebesin-workshop/2025/app
```

**3) Měli byste vidět 2 soubory:**

Containerfile → definice pro sestavení image

web.py → aplikace, kterou “zabalíme do kontejneru”

# Sestavení a spuštění

## 1) Sestavme novou image (v adresáři app):

```
podman build .
```

## 2) Nalezneme naši novou image a otagujeme ji:

```
podman image → ukáže image s [IMAGE_ID]  
podman tag [IMAGE_ID] [NAZEV]
```

## 3) Spustíme kontejner používající naši image:

```
podman run -p 8080:8081 [NAZEV]
```

## 4) Ověříme, že aplikace v našem kontejneru je funkční

→ V browseru otevřeme adresu 127.0.0.1:8080

# Několik rychlých „tipů“

**1) Jak se připojit ke běžícímu kontejneru (jako v našem příkladu)?**

V jiném okně emulátoru terminálu spustíme:

```
podman exec -it [ID nebo JMENO kontejneru]
```

**2) Jak pojmenovat náš kontejner, abychom nemuseli použít ID?**

Spustíme náš kontejner s argumentem –name:

```
podman run --name moje_fedora -it fedora:43
```

**3) Jak uděláme změny v běžícím kontejneru perzistentními?**

```
podman commit [ID nebo JMENO kontejneru]
```

```
podman images # => Na prvním místě bude nově vzniklá image
```

**4) Jak zjistit informace o běžících procesech v kontejneru?**

```
podman top [ID nebo JMENO kontejneru]
```

# Práce s úložištěm

## 1) Vytvořte nový adresář:

```
mkdir ~/uloziste
```

## 2) Spusťte kontejner pomocí:

```
podman run \  
    -v ~/uloziste:/tmp/uloziste:Z \  
    -p 8080:8081 [NAZEV]  
    # :Z nastaví korektně selinux kontext (předejdeme tak  
    # chybám s oprávněními), alternativně můžeme přepnout  
    # selinux do režimu permissive přes ,sudo setenforce 0'
```

## 3) Přistupte několikrát k webové aplikaci a zastavte kontejner

## 4) Co se nachází v adresáři ~/uloziste?

## 5) Pozn. pro pokročilejší práci použijte podman **volumes**

Otázky?

