Docker + Portainer

Instalace dockeru a portaineru

- Instalace Dockeru
- Instalace Portaineru
- Instalace Portaineru NEW

Instalace Dockeru

Instalace Dockeru

Díky šikovnému instalačnímu skriptu vyvinutému týmem Docker je instalace softwaru kontejneru neuvěřitelně jednoduchá.

Následující kroky můžete dokonce dokončit pomocí připojení SSH k vašemu Raspberry Pi .

1. Naším prvním úkolem je aktualizovat všechny naše stávající balíčky, než přistoupíme k instalaci Dockeru.

Všechny stávající balíčky můžeme upgradovat spuštěním následujících dvou příkazů na Raspberry Pi.

```
sudo apt update
sudo apt upgrade
```

2. S naším Raspberry Pi zcela aktuálním můžeme nyní pokračovat a nainstalovat Docker na Raspberry Pi.

Naštěstí pro nás Docker tento proces neuvěřitelně zrychlil a zjednodušil tím, že poskytl bash skript, který vše nainstaluje za vás.

Můžete si stáhnout a spustit oficiální instalační skript Docker spuštěním následujícího příkazu.

```
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
```

Tento příkaz převede skript přímo do příkazového řádku. Obvykle by bylo nejlepší, kdybyste to neudělali; Docker je však důvěryhodný zdroj.

Pokud si nejste jisti, zda to spustit přímo bez předchozí kontroly, můžete přímo na get.docker.com . skript zobrazit

Dokončení tohoto skriptu může nějakou dobu trvat, protože automaticky detekuje a nainstaluje vše, co potřebuje ke spuštění Dockeru na Raspberry Pi.

Nastavení uživatele pro Docker

Než budeme moci začít používat Docker bez problémů, musíme provést mírné úpravy našeho uživatele.

Souvisí to se způsobem, jakým systém oprávnění Linux pracuje s Dockerem. Ve výchozím nastavení může s Dockerem komunikovat pouze uživatel Dockeru, ale existuje způsob, jak to obejít.

1. Jakmile Docker dokončí instalaci do vašeho Raspberry Pi, musíme udělat ještě několik věcí.

Aby mohl jiný uživatel komunikovat s Dockerem, musí být přidán do `docker` skupina.

Takže naším dalším krokem je přidání našeho aktuálního uživatele do `docker` skupinu pomocí příkazu usermod , jak je uvedeno níže. Používáním " `$USER` ” vkládáme proměnnou prostředí, která ukládá jméno aktuálního uživatele.

```
sudo usermod -aG docker $USER
```

Pokud do skupiny nepřidáme našeho uživatele, nebudeme moci komunikovat s Dockerem bez spuštění jako uživatel root.

Chcete-li se dozvědět více o oprávněních a skupinách v Linuxu, podívejte se na naše oprávnění k souborům v příručce Linux .

2. Protože jsme u našeho uživatele provedli nějaké změny, budeme se nyní muset odhlásit a znovu přihlásit, aby se změny projevily.

Můžete se odhlásit spuštěním následujícího příkazu v terminálu.

```
logout
```

3. Jakmile se znovu přihlásíte, můžete si ověřit, že skupina dockerů byla úspěšně přidána k vašemu uživateli spuštěním následujícího příkazu.

```
groups
```

Tento příkaz zobrazí seznam všech skupin, kterých je aktuální uživatel součástí. Pokud vše fungovalo, jak má, skupina `docker` by zde mělo být uvedeno.

Testování instalace Dockeru na Raspberry Pi

Když je Docker nyní nastaven na našem Raspberry Pi, měli bychom nyní pokračovat a testovat, abychom se ujistili, že funguje.

1. Chcete-li otestovat, zda Docker funguje, budeme pokračovat a spustíme následující příkaz na našem Pi.

Tento příkaz řekne Dockeru, aby si stáhl, nastavil a spustil kontejner dockeru s názvem „ **hello-world** .

```
docker run hello-world
```

2. Pokud jste úspěšně nainstalovali Docker do vašeho Raspberry Pi, měli byste vidět zprávu s následujícím textem.

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Záv?r

Nyní můžete bezpečně začít používat Docker pro svůj projekt, jako je nastavení Docker swarm s vaším Raspberry Pi.

Docker kontejnery jsou skvělý způsob, jak rychle nasadit software do vašeho zařízení.

Chcete-li si usnadnit život při správě kontejnerů Docker, můžete také nastavit Portainer na zařízení Pi . Portainer je úhledný kus softwaru, který vám umožňuje vytvářet a spravovat kontejnery ve webovém rozhraní.

Pokud jste našli nějaké problémy s instalací Dockeru, pak neváhejte napsat komentář níže.

Instalace Portaineru

Instalace a aktualizace Portaineru (Docker)

Tento návod slouží k instalaci Portainer CE pomocí Dockeru. Veškerá data a nastavení budou uložena v Docker volume, takže o ně při aktualizaci nepřijdete.

1. Instalace Portaineru

Nejdříve vytvoříme svazek pro data a následně spustíme samotný kontejner.

Krok 1: Vytvoření volume pro trvalá data

```
sudo docker volume create portainer_data
```

Ověření:

```
docker volume inspect portainer_data
```

Krok 2: Spuštění kontejneru Portainer

```
docker run -d \
  --name=portainer \
  --restart=always \
  -p 9000:9000 \
  -p 9443:9443 \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -v portainer_data:/data \
  portainer/portainer-ce:latest
```

- **-p 9000:9000**: Přístup přes HTTP (nezabezpečený).
- **-p 9443:9443**: Přístup přes HTTPS (zabezpečený – doporučeno).
- **--name=portainer**: Pojmenování kontejneru pro snadnou správu.
- **--restart=always**: Portainer se automaticky spustí po restartu serveru.
- **-v /var/run/docker.sock...**: Umožňuje Portaineru ovládat Docker na vašem serveru.
- **-v portainer_data:/data**: Propojení vytvořeného svazku s vnitřkem kontejneru (vaše data).

Co je kritické:

Parametr	Proč
`-v portainer_data:/data`	**ukládá stack YAML, uživatele, nastavení**
`--restart=always`	přežije reboot
`docker.sock`	správa Dockeru

3 Ověření správné instalace (POVINNÉ)

```
docker inspect portainer | grep -A10 Mounts
```

Správný výstup MUSÍ obsahovat:

"Type": "volume",
"Source": "/var/lib/docker/volumes/portainer_data/_data",
"Destination": "/data"

Pokud to tam není → **OKAMŽITĚ STOP** – instalace je špatně.

? BEZPE?NÁ AKTUALIZACE PORTAINERU

(bez ztráty YAML)

> Aktualizuje se **POUZE kontejner**, **NIKDY volume**

Tento postup použijte vždy, když chcete přejít na novější verzi. Data zůstanou zachována.

Krok 1: Stažení nejnovější verze obrazu z internetu

```
sudo docker pull portainer/portainer-ce:latest
```

Krok 2: Zastavení aktuálně běžícího Portaineru + Krok 3: Odstranění starého kontejneru *(Nebojte se, data jsou bezpečně v portainer_data volume)*

```
sudo docker stop portainer
sudo docker rm portainer
```

Krok 4: Spuštění Portaineru znovu (už z nové verze)

```
docker run -d \
  --name=portainer \
```

```
  --restart=always \
  -p 9000:9000 \
  -p 9443:9443 \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -v portainer_data:/data \
  portainer/portainer-ce:latest
```

➡ Po přihlášení:

- Stacky
- YAML
- Historie
- Credentials
  = VŠECHNO ZACHOVÁNO

P?ihlášení do rozhraní

Po instalaci nebo aktualizaci otevřete prohlížeč a zadejte:

- **HTTP:** `http://vase-ip-adresa:9000`
- **HTTPS:** `https://vase-ip-adresa:9443`

*Poznámka: Jak jsme si potvrdili, veškeré další programy už budete instalovat přes Portainer UI v sekci **Stacks** s využitím Environment Variables.*

?? 2) SMAZAT VŠECHNY IMAGE, KTERÉ NEPOUŽÍVÁ ŽÁDNÝ KONTEJNER

Tímto smažeš:

- **i staré image, které nejsou dangling, ale nemá je přiřazen žádný kontejner:**

```
docker image prune -a -f
```

App Templates:

```
https://raw.githubusercontent.com/Lissy93/portainer-templates/main/templates_v3.json
```

Konfigurace Porttaineru na vašem Raspberry Pi

Když se poprvé připojíte k webovému rozhraní Portaineru, budete muset provést některé počáteční kroky nastavení.

Nebojte se, protože tyto jsou neuvěřitelně přímočaré a pomáhají zajistit, že bude fungovat tak, jak chceme.

1. Při prvním spuštění webového rozhraní Portaineru si budete muset vytvořit účet správce.

Chcete-li vytvořit tento administrátorský účet, musíte mu dát uživatelské jméno (**1.**).

Dále budete muset tomuto novému účtu přiřadit heslo (**2.**). Portainer vyžaduje, abyste měli hesla dlouhá alespoň 12 znaků.

Jakmile nastavíte požadované uživatelské jméno a heslo, click" `Create user` ” pro dokončení jeho vytváření.

Vytvoření uživatele správce

2. Nyní musíme vybrat, jaký druh prostředí kontejneru chceme, aby Porttainer spravoval.

V našem případě musíme vybrat `Docker` Možnost **1.**).

Jakmile tuto možnost vyberete, clicka `Connect` tlačítko (**2.**).

Vyberte Prostředí Portainer Docker

3. Nyní byste měli mít Porttainer úspěšně spuštěný na vašem Raspberry Pi.

Nyní jej můžete použít ke správě kontejnerů Docker běžících na vašem zařízení.

Otevřeno webové uživatelské rozhraní Raspberry Pi Portainer

Základní použití uživatelského rozhraní Portainer

Tato část ukáže základy používání Portainer UI k vytvoření nového kontejneru Docker.

Podle těchto kroků si Portainer stáhne obrázek kontejneru do vašeho Raspberry Pi a nastaví jej.

1. Když vstoupíte do rozhraní Portainer, měli byste vidět obrazovku jako níže.

Můžete vidět, že naše instalace Raspberry Pi Docker je uvedena jako koncový bod Portainer. Clicktento koncový bod otevřete jeho nastavení.

Otevřete Docker Endpoint

2. Zde uvidíte rychlý přehled Dockeru spuštěného ve vašem systému.

Na této obrazovce se zobrazí počet kontejnerů, svazků, obrazů a sítí ve vašem systému.

Chcete-li zjistit, jaké kontejnery běží na vašem Raspberry Pi, click" `Containers` “ na postranním panelu.

Přejděte na kartu Kontejnery

3. Porttaineru se nyní zobrazí seznam kontejnerů aktuálně nastavených na vašem zařízení (**1.**).

Když vyberete kontejner, budete jej moci ovládat volbou, zda chcete kontejner spustit, zastavit, zabít (**2.**).

Pokud chcete přidat zcela nový kontejner, můžete click" `Add container` tlačítko “ (**3.**).

Přehled seznamu kontejnerů

3. A konečně, tato obrazovka je ta, která vám umožní vytvářet nové kontejnery v Porttaineru.

Pomocí této možnosti můžete zvolit různá nastavení.

Nejprve musíte definovat název, který chcete tomuto kontejneru přidělit (**1.**).

Poté musíte vybrat obrázek, který má Portainer natáhnout do vašeho Raspberry Pi (**2.**).

Je dokonce možné nastavit síťové porty, které má váš kontejner Docker používat (**3.**).

Kromě toho můžete také ovládat, kdo může tento kontejner spravovat ve webovém rozhraní Portaineru (**4.**).

Nakonec je zde řada pokročilých možností, které můžete nastavit pomocí pole v dolní části stránky (**5.**). Ten, který budete pravděpodobně konfigurovat nejvíce, je „ `Restart Policy` “.

Pomocí toho můžete zajistit, že se váš kontejner dockeru automaticky restartuje, když se Raspberry Pi restartuje.

Jakmile budete spokojeni se všemi nastaveními nového kontejneru Docker, můžete click" `Deploy the container` tlačítko “ (6.)

Vytvořte nové kontejnery v Porttaineru pro Raspberry Pi

Záv?r

V tuto chvíli byste měli mít úspěšně spuštěn Portainer na vašem Raspberry Pi.

Portainer je šikovný software, který vám umožní snadno spravovat vaše kontejnery Docker.

Pomocí webového rozhraní můžete vytvářet, upravovat a mazat své kontejnery.

Pokud máte nějaké problémy se spuštěním Porttaineru na vašem Raspberry Pi, zanechte prosím komentář níže.

Měli byste se také podívat na naše další projekty Raspberry Pi , abyste viděli, jak jinak můžete využít své Pi.

Instalace Portaineru NEW

Tady máš **CELÝ POSTUP OD NULY**, přesně **tak, jak chceš**:

- ✔ Portainer **instalovaný jako stack přes YML**
- ✔ **VŠECHNO v** `/opt/stack/`
- ✔ **file-based stack** (Portainer YAML **nevlastní**)
- ✔ **aktualizace / reinstalace Portaineru = BEZ** `LIMITED`
- ✔ odpovídá tomu, co chceš dlouhodobě provozovat

Žádné alternativy, žádné teorie. **Toto je správný a bezpečný způsob.**

? CÍLOVÝ STAV

```
/opt/stack/
├── portainer/
│   └── docker-compose.yml
├── portainer-data/
└── (ostatní stacky později stejným stylem)
```

0?? P?EDPOKLADY

- Docker **už máš**
- Používáš **docker compose plugin** (`docker compose`)
- Vše děláš jako `root` nebo přes `sudo`

Ověření (jen pro klid):

```
docker --version
docker compose version
```

1?? VYTVO? STRUKTURU (POVINNÉ)

```
mkdir -p /opt/stack/portainer
mkdir -p /opt/stack/portainer-data
```

□□ DŮLEŽITÉ

- `portainer-data` je **perzistence**
- **nikdy** nepoužívej Docker volume typu `portainer_data`
- jen **bind mount do** `/opt`

2?? VYTVO? YML PRO PORTAINER (FILE-BASED STACK)

Soubor:

```
nano /opt/stack/portainer/docker-compose.yml
```

Obsah **VLOŽ DOSLOVA**:

```
version: "3.8"

services:
  portainer:
    image: portainer/portainer-ce:latest
    container_name: portainer
    restart: unless-stopped

    ports:
      - "9000:9000"
      - "9443:9443"

    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /opt/stack/portainer-data:/data
```

✔ YAML je **soubor na disku**
✔ Portainer ho **nikdy nebude ukládat do DB**
✔ upgrade-safe

3?? NASAZENÍ STACKU (RU?N?, BEZ UI)

⚠ **Tohle je záměr** – první start vždy ručně.

```
cd /opt/stack/portainer
docker compose up -d
```

Ověření:

```
docker ps | grep portainer
```

4?? PRVNÍ P?IHLÁŠENÍ DO PORTAINERU (KRITICKÉ)

Otevři v prohlížeči:

```
https://IP_SERVERU:9443
```

Nastav:

- vytvoř **JEDEN admin účet**
- **neměň ho**
- **nemaž**
- **žádní další uživatelé** (pokud nejsou nutní)

□□ RBAC změny = častý zdroj LIMITED

5?? JAK TENHLE STACK ZAREGISTRUJEŠ DO PORTAINERU (SPRÁVN?)

Teď to důležité.

V Portainer UI:

1. **Stacks → Add stack**
2. **Build method: *File upload / Path***
3. **NE web editor**
4. Zadej **cestu k souboru**:

```
/opt/stack/portainer/docker-compose.yml
```

5. Deploy

□□ Portainer:

- uloží si **jen cestu**
- **NEOBSAH** YML
- **nevznikne LIMITED**
- upgrade je bezpečný

6?? AKTUALIZACE PORTAINERU (BEZ RIZIKA)

Kdykoliv později:

```
cd /opt/stack/portainer
docker compose pull
docker compose up -d
```

➡ NIC jiného se nedotkne
➡ ŽÁDNÝ stack se nepřepne do LIMITED

7?? REINSTALACE PORTAINERU (NUCLEAR SAFE)

Když se cokoliv posere:

```
docker stop portainer
docker rm portainer
```

 Stacky mimo Portainer běží dál

Pak znovu:

```
cd /opt/stack/portainer
docker compose up -d
```

A v Portainer UI:

- znovu **Stacks → Add stack**
- znovu **Path**
- stejný soubor
- hotovo

8?? PRAVIDLA, KTERÁ NESMÍŠ PORUŠIT (D?LEŽITÉ)

 nepoužívej **Web editor**
 nepoužívej **Git stack**
 nekopíruj YML do UI
 neupravuj stack v Portaineru

✔ YML **VŽDY** upravuj v:

```
/opt/stack/<nazev>/docker-compose.yml
```

✔ Pak:

`docker compose up -d`

nebo **Redeploy** (file-based je OK)

? PRO? TOHLE FUNGUJE (STRU?N?)

- Portainer **nevlastní YAML**
- nemá co porovnávat při upgradu
- nemá co označit jako „externí“
- ownership zůstává konzistentní
- **LIMITED NEMÁ JAK VZNIKNOUT**

? FINÁLNÍ CHECKLIST

✔ `/opt/stack/portainer/docker-compose.yml` existuje
✔ Portainer stack je **file-based**
✔ Data jsou v `/opt/stack/portainer-data`
✔ Žádné Docker volume
✔ Jeden admin
✔ Žádný Web editor

➡ MÁŠ HOTOVO A BEZPEČNO