

# Název projektu: Instalace a ověření Dockeru na Raspberry Pi 5

Škola: OAUH

Řešitelé projektu: Nela Slezáková, Kryštof Blažek

## Úvod

Cílem tohoto projektu bylo nainstalovat Docker na zařízení Raspberry Pi 5, ověřit jeho funkčnost pomocí oficiálního kontejneru nginx a popsat celý proces včetně testování. Docker umožňuje jednoduchou správu aplikací v kontejnerech a je hojně využíván pro vývoj i produkční nasazení softwaru. Prakticky tak můžeme rychle a efektivně spouštět různé služby bez nutnosti složité konfigurace operačního systému.

Projekt demonstruje základní práci s Dockerem na Raspberry Pi 5, což je aktuálně výkonná platforma pro domácí či vzdělávací nasazení.

## Cíle projektu

1. Instalovat Docker na Raspberry Pi 5.
2. Spustit základní testovací kontejner (hello-world).
3. Spustit nginx kontejner a otestovat připojení přes síť.
4. Zdokumentovat celý postup a výsledky.

Ověření úspěšné instalace bude provedeno podle následujících testů:

- Výpis správného běhu příkazu ``docker run hello-world``.
- Zobrazení výchozí stránky nginxu po spuštění kontejneru a připojení z Windows.

## Použité materiály

- Oficiální dokumentace Docker: <https://docs.docker.com/>
- Oficiální Docker image nginx: [https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)
- SSH přístup z Windows do Raspberry Pi (Windows Terminal)

## Postup řešení

1. Připojili jsme se k Raspberry Pi 5 přes SSH z Windows.
2. Aktualizovali jsme systémové balíčky příkazem ``sudo apt update && sudo apt upgrade``.
3. Nainstalovali jsme Docker příkazem:  
``curl -fsSL https://get.docker.com -o get-docker.sh && sudo sh get-docker.sh``
4. Ověřili instalaci pomocí:  
``docker run hello-world``  
- Zobrazila se potvrzující zpráva o úspěchu.
5. Spustili jsme nginx kontejner na portu 8080, protože port 80 byl obsazen:  
``docker run -d -p 8080:80 --name webserver nginx``
6. Připojili jsme se z Windows prohlížeče na adresu:  
``http://<IP_RPi>:8080``  
- Zobrazila se výchozí stránka nginxu.

Poznámka: Pokud by port 80 nebyl obsazený, je možné spustit nginx i na portu 80.  
Pokud je kontejner s názvem ``webserver`` již vytvořen, je třeba ho nejprve odstranit:  
``docker rm -f webserver``

```
krekry@raspberrypi:~ $ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c9c5fd25a1bd: Pull complete
Digest: sha256:940c619fbd418f9b2b1b63e25d8861f9cc1b46e3fc8b018ccfe8b78f19b8cc4f
Status: Downloaded newer image for hello-world:latest

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

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (arm64v8)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

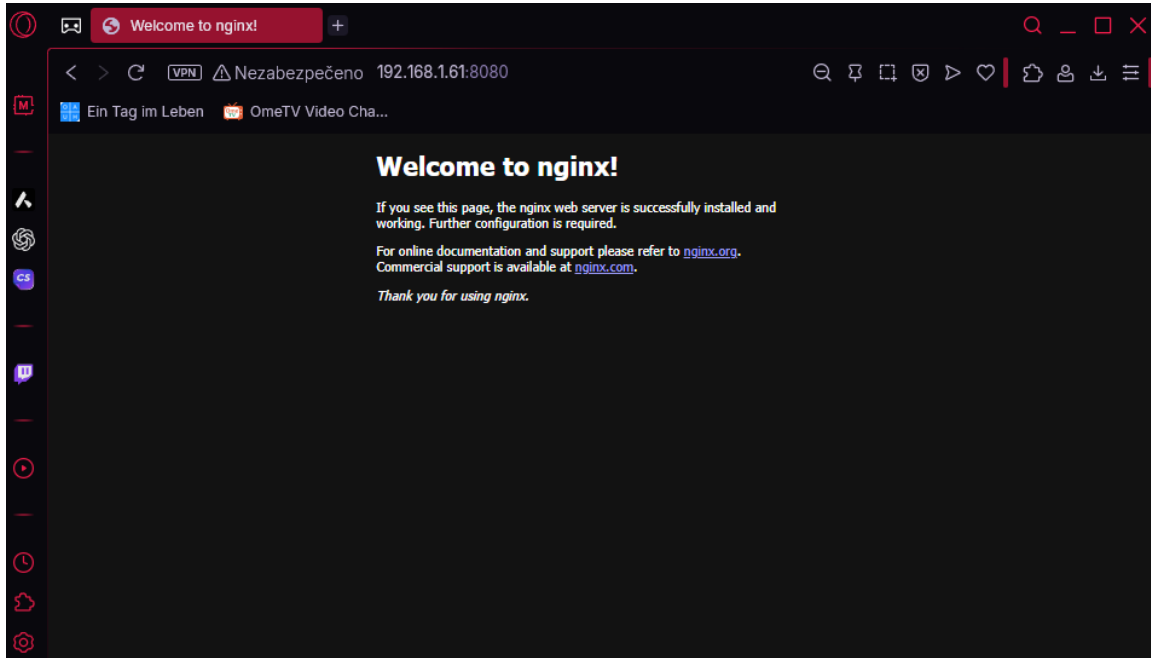
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```
krekry@raspberrypi:~ $ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
krekry@raspberrypi:~ $ docker run -d -p 80:80 --name webserver nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
34ef2a75627f: Pull complete
06cdc65323d: Pull complete
27fe6a09bce8: Pull complete
ca26eebd8366: Pull complete
9d29d0ca23f3: Pull complete
447b5a3f15c6: Pull complete
71eccce642b8: Pull complete
Digest: sha256:6784fb0834aa7dbbe12e3d7471e69c290df3e6ba810dc38b34ae33d3c1c05f7d
Status: Downloaded newer image for nginx:latest
fa6627c683836e6f447a49386e3cb0f8341c0a2ec553ae7f0a4b90415798b5c7
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint webserver (9c566d16a9a84a2720f4a1c3f38d4d902e135df27c35d0f57c85be21deb6cef8): failed to bind host port for 0.0.0.0:80:172.17.0.2:80/tcp: address already in use

Run 'docker run --help' for more information
krekry@raspberrypi:~ $ docker run -d -p 8080:80 --name webserver nginx
docker: Error response from daemon: Conflict. The container name "/webserver" is already in use by container "fa6627c683836e6f447a49386e3cb0f8341c0a2ec553ae7f0a4b90415798b5c7". You have to remove (or rename) that container to be able to reuse that name.

Run 'docker run --help' for more information
krekry@raspberrypi:~ $ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
fa6627c68383   nginx     "/docker-entrypoint..." 2 minutes ago   Created   webserver
13289534e2ea   hello-world  "/hello"   3 minutes ago   Exited (0) 3 minutes ago   zealous_proskuriakova
krekry@raspberrypi:~ $ docker rm webserver
webserver
krekry@raspberrypi:~ $ docker stop webserver
docker rm webserver
Error response from daemon: No such container: webserver
Error response from daemon: No such container: webserver
krekry@raspberrypi:~ $ docker run -d -p 8080:80 --name webserver nginx
64eb56622ce38bcd9bc95926ed7814fda92dea1d92e7ef4219e955dbef7d24d3
```



## **Dokumentace testování**

Úspěšné provedení příkazu ``docker run hello-world`` potvrdilo, že Docker je správně nainstalován a funguje.

Zobrazení výchozí stránky nginxu na portu 8080 potvrdilo funkčnost kontejneru a správné síťové nastavení.

## **Rozdělení práce**

### **Kryštof Blažek**

- Instalace Dockeru na Raspberry Pi 5
- Nastavení a spuštění testovacího Docker kontejneru (hello-world)
- Řešení problémů s porty a spuštěním Nginx kontejneru
- Příprava základní dokumentace a příkazů pro instalaci
- Testování funkčnosti Dockeru a kontejnerů na RPi
- Zpracování závěrečného dokumentu a formátování do Wordu

### **Nela Slezáková**

- Zajištění připojení přes SSH z Windows
- Kontrola přístupu a oprávnění uživatele na Raspberry Pi
- Testování přístupu k webovému serveru přes prohlížeč
- Vkládání screenshotů do dokumentace
- Kontrola a korektura textu dokumentace
- Finalizace dokumentu a příprava PDF verze

## Závěr

Projekt úspěšně prokázal základní funkčnost Dockeru na Raspberry Pi 5.

Běh nginx kontejneru a připojení z Windows bylo bez problémů.

V budoucnu lze rozšířit o nasazení dalších služeb v kontejnerech.

Pozor na možné konflikty portů při provozu nativních webových serverů.

Varianty řešení:

- Spuštění kontejnerů na jiných portech.
- Odstranění nativních služeb zabírajících port 80.
- Použití Docker Compose pro složitější konfigurace.

Upozornění:

- Ujistěte se, že porty nejsou blokovány firewallem.
- Pokud máte kontejner pojmenovaný stejně, nejdříve jej odstraňte.

## Použité zdroje

1. Docker Documentation: <https://docs.docker.com/>
2. Nginx Docker Hub: [https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)
3. SSH klient PuTTY: <https://www.putty.org/>

Poznámka: Některé části vznikly i na základě komunikace s AI ChatGPT.