

WSL (Windows Subsystem for Linux) – Praktický průvodce & tipy

 Moderní přehled instalace, nastavení a doporučení pro práci s WSL na Windows.

Co je WSL?

- **WSL** umožňuje instalaci a provoz linuxových distribucí přímo na Windows.
- Umožňuje spouštět linuxové aplikace bez potřeby virtuálního stroje.

Note


Ideální pro vývojáře, kteří potřebují Linux nástroje na Windows.

Instalace WSL na Windows

- ▶ ♦ Krok 1: Povolení virtualizace v BIOS/UEFI
- ▶ ♦ Krok 2: Povolení WSL v systému
- ▶ ♦ Krok 3: Povolení Virtual Machine Platform & WSL 2
- ▶ ♦ Krok 4: Stažení aktualizace Linux jádra
- ▶ ♦ Krok 5: Nastavení WSL 2 jako výchozí
- ▶ ♦ Krok 6: Instalace linuxové distribuce



Docker – Praktický průvodce & tipy

 Moderní přehled základních pojmů, příkazů, konfigurace a doporučení pro práci s Dockerem na Windows.



Co je Docker?

- **Platforma pro vývoj, doručování a běh aplikací pomocí kontejnerizace**
- Izoluje aplikace v kontejnerech se všemi závislostmi
- Kontejnery jsou rychlejší a efektivnější než klasická virtualizace

Note

Pro instalaci Dockeru na Windows je nutné mít nainstalovaný [WSL](#).



Základní pojmy

- ▶  Dockerfile
- ▶  Docker image
- ▶  Docker run
- ▶  Docker hub
- ▶  Docker engine
- ▶  Docker compose



Soubory Dockeru

- ▶ `dockerd.exe`
- ▶ `docker.exe`
- ▶ `docker-compose.exe`
- ▶ `docker-compose.yml`



Základní příkazy

 Kategorie	 Příkaz	 Popis
 Zobrazení	<code>docker ps</code>	Zobrazí běžící kontejnery
	<code>docker images</code>	Zobrazí všechny docker image
 Aktualizace nastavení	<code>docker update --restart=yes <jméno/id></code>	Zapne automatické spouštění kontejneru
	<code>docker update --restart=no <jméno/id></code>	Vypne automatické spouštění kontejneru
 Stažení	<code>docker pull <image></code>	Stáhne docker image
 Záloha & obnova	<code>docker save -o <cesta>.tar <image/id></code>	Zálohuje image do souboru
	<code>docker load -i <cesta>.tar</code>	Obnoví image ze souboru
 Vytvoření image	<code>docker build -t <jméno> .</code>	Sestaví image z Dockerfile
 Spuštění kontejneru	<code>docker run <image></code>	Spustí kontejner z image
	<code>docker run -p 70:80 <image></code>	Spustí kontejner na jiném portu
	<code>docker run --rm <image></code>	Spustí kontejner a smaže ho po ukončení
	<code>docker run -it <image></code>	Spustí kontejner v interaktivním módu
 Docker Compose	<code>docker-compose up</code>	Spustí služby z docker-compose (běží v terminálu)
	<code>docker compose up -d</code>	Spustí služby z docker-compose na pozadí
	<code>docker compose down</code>	Zastaví a odstraní všechny kontejnery z docker-compose
 Zastavení & odstranění	<code>docker stop <jméno/id></code>	Zastaví kontejner
	<code>docker rm <jméno/id></code>	Odstraní kontejner
	<code>docker rmi <image/id></code>	Odstraní image

Dockerfile – Příklady

- ▶ [.NET Core aplikace](#)
 - ▶ [C# aplikace s buildem](#)
 - ▶ [Lokální NuGet balíčky](#)
-

Volumes & data

- ▶ [Propojení složky z Windows s kontejnerem](#)
 - ▶ [Zachování dat z kontejneru na lokálním disku](#)
-

Řešení problémů

- ▶ [Port není dostupný](#)



Portainer – Praktický průvodce & tipy



Moderní přehled spuštění, parametrů a doporučení pro práci s Portainerem v Dockeru.



Co je Portainer?

- **Webové rozhraní pro správu Docker kontejnerů**
- Umožňuje snadnou správu, monitoring a konfiguraci kontejnerů, image, volume a sítí
- Podporuje Docker, Docker Swarm, Kubernetes



Note

Portainer výrazně zjednodušuje správu Docker prostředí.



Spuštění Portaineru

- ▶ [♦ Krok 1: Spuštění kontejneru](#)



Přístup k Portaineru

- ▶ [♦ Krok 2: Otevření webového rozhraní](#)



Řešení problémů

- ▶ [Port 9000 je obsazený](#)
- ▶ [Chyba při připojení k Docker daemonu](#)

Docker Duplicati – Praktický průvodce & tipy

 Moderní přehled nastavení složek, oprávnění a doporučení pro zálohování s Duplicati v Dockeru.



Co je Duplicati?

- **Open-source nástroj pro zálohování dat**
- Umožňuje šifrované, komprimované a inkrementální zálohy
- Snadná integrace s Dockerem

Note

Ideální pro automatizované zálohy v kontejnerizovaném prostředí.

Krok 1: Vytvoření a nastavení složky pro zálohy

- ▶  [Vytvoření složky](#)
- ▶  [Nastavení oprávnění](#)

Krok 2: Nastavení oprávnění pro Docker volumes

- ▶  [Rekurzivní nastavení složek a souborů](#)

Popis	Cesta
Host/volume	<code>/var/lib/docker/volumes</code> (cesta k diskům)
Path in container	<code>/docker_XX</code> (vlastní cesta v kontejneru pro Volume)


Important

V kontejneru Duplicati musí být typ svazku nastaven na **Bind**.

Obnovení výchozích oprávnění

- ▶ [Vrácení oprávnění na výchozí hodnoty](#)

Docker BusyBox – Praktický průvodce zálohováním volumes

 Moderní přehled zálohování dat z Docker volumes do lokální složky pomocí BusyBox.


Co je BusyBox?

- Minimalistický Linux image s základními Unix nástroji
- Ideální pro jednoduché operace v Docker kontejnerech (kopírování, skripty, testování)

Note

BusyBox je často využíván pro rychlé utility v kontejnerizovaném prostředí.

Krok 1: Zjištění dostupných Docker volumes

►  [Zobrazit seznam volumes](#)

Krok 2: Záloha dat z Docker volume do počítače

►  [Přenesení dat pomocí BusyBox](#)



Kubernetes – Praktický průvodce & tipy



Moderní přehled základních pojmů, principů a doporučení pro práci s Kubernetes.



Co je Kubernetes?

- **Orchestrátor kontejnerů** pro automatizované zavádění, správu, škálování a aktualizaci aplikací v kontejnerech.
- Umožňuje efektivně využívat kapacitu serverů, dynamicky spouštět nové verze služeb a vypínat staré.
- Klíčový nástroj pro správu microservices architektury.



Important

Pro práci s Kubernetes je zapotřebí znalost Dockeru.





Základní pojmy

- ▶  Kontejner
- ▶  Pod
- ▶  Service
- ▶  Deployment
- ▶  Cluster



Orchestrátory

- ▶  Co dělá orchestrátor?
- ▶  Ukázka orchestrátoru