



20. Virtualizace a kontejnery

Virtualizace a kontejnery – definice, základní pojmy, účel

Virtualizace

- **Virtualizace** vytváří simulované (virtuální) výpočetní prostředí namísto fyzického.
- Často zahrnuje počítačem vytvořené verze hardwaru, operačních systémů, úložných zařízení apod.
- Umožňuje rozdělit jeden fyzický počítač (tzv. *hostitelský systém*) na více nezávislých **virtuálních strojů (VM – Virtual Machines)**.
- Každý virtuální stroj funguje jako samostatný počítač, může mít vlastní **operační systém** a aplikace, a přitom sdílí fyzické prostředky (CPU, RAM, disk) hostitelského zařízení.

Účel virtualizace

- **Efektivnější využití hardwaru** – více systémů na jednom fyzickém stroji
- **Úspora nákladů** – méně fyzických zařízení = nižší náklady na elektřinu, chlazení, prostor a údržbu
- **Lepší škálovatelnost** – snadné přidávání nebo odebírání virtuálních strojů
- **Snadnější správa a zálohování** – rychlé klonování, snapshoty, obnova systémů

Výhody virtualizace

- Úspora nákladů na hardware a provoz
- Zvýšení dostupnosti a flexibility
- Rychlá obnova při selhání systému
- Možnost provozovat různé operační systémy současně

Základní pojmy

- **Hypervisor (virtualizační vrstva)** – software, který umožňuje provozovat více operačních systémů současně na jednom fyzickém stroji. Dva typy:
 - *Typ 1 (bare-metal)* – běží přímo na hardwaru (např. VMware ESXi, Microsoft Hyper-V)
 - *Typ 2 (hostovaný)* – běží nad hostitelským OS (např. VirtualBox, VMware Workstation)
- **Hostitelský systém (host machine)** – fyzický počítač, na kterém běží hypervisor
- **Host (virtuální počítač)** – vytvořený virtuální systém, který běží uvnitř hypervisoru
- **Snapshot** – snímek aktuálního stavu virtuálního počítače, slouží k záloze a návratu zpět

Hlavní typy virtualizace

1. Virtualizace plochy (desktop virtualization)

- Jeden server poskytuje vzdálený přístup k desktopovému prostředí (např. RDP)

2. Virtualizace sítě (network virtualization)

- Logické rozdělení sítě na menší části (např. VLANy), řízení síťového provozu

3. Virtualizace softwaru/aplikací

- Spouštění aplikací nezávisle na OS (např. Java, .NET, Docker)

4. Virtualizace úložiště (storage virtualization)

- Sjednocení více fyzických úložišť do jednoho logického celku

Kontejnery

Definice

- Kontejner je standardizovaná jednotka softwaru, která obsahuje veškerý kod, runtime, knihovny a závislosti potřebné k chodu aplikace
- Je to v podstatě taková krabička, která uchovává všechno, co aplikace potřebuje k tomu, aby fungovala
- Díky kontejnerizaci může aplikace běžet konzistentně v různých prostředích

Účel

- kontejnery umožňují vývojářům vytvářet a nasazovat aplikace, které jsou flexibilní(mohou běžet v různých prostředích)
- Tyto aplikace jsou lehce škálovatelné
- Zjednoduší také nasazování a správu aplikací
- Izolace aplikací: chyba v jedné aplikaci neovlivní chod ostatních
- Kontejnery také dále zvýší efektivitu využití zdrojů- na jednom serveru může běžet více kontejnerů

Výhody kontejnerů

- Přenositelnost
 - Běží na jakémkoliv systému, který je podporuje
- Rychlost nasazení a spuštění
- Efektivní využití zdrojů(sdílí jádro hostitelského OS, nepotřebují vlastní, takže spotřebují méně paměti a výkonu než virtuální počítače)

- Lepší škálovatelnost-snadné nasazení více instancí jedné aplikace

Porovnání s virtualizací

- Kontejnery sdílí jádro hostitelského OS, čímž šetří:
 - Využití hardware prostředků: RAM, CPU, atd.
 - Spouštějí se rychleji
 - Umožňují vyšší hustotu aplikací na stejném hardwaru

Základní pojmy

- Kontejner
 - Izolované prostředí obsahující aplikaci a vše potřebné k jejímu spuštění(kod, knihovny, závislosti)
- Docker
 - Nejrozšířenější platforma pro vytváření, správu a běh kontejnerů
- Dockerfile
 - Soubor obsahující instrukce k tomu, jak vytvořit image
- Image(obraz)
 - šablona pro vytvoření kontejneru, obsahuje aplikaci a její prostředí. Z jednoho image lze vytvořit více kontejnerů
- Container registry
 - Úložiště obrazů kontejnerů(docker hub, azure container registry)
- Orchestrace:
 - Možnost spravovat více kontejnerů zároveň automaticky(jejich nasazení, škálování, atd.)

Virtualizace v OS Windows a v UNIX-like OS

OS Windows

- Obvykle musíme nejdřív virtualizaci povolit:
 - Půjdeme do BIOS a vyhledáme rozšířené nastavení procesoru
 - V pokročilém nastavení procesoru povolíme možnost virtualizace
 - Ve windows 11 se to nastavuje v ovládacích panelech
- Virtualizaci jsem rozdělil podle typu hyper visoru. Typ hypervisoru nám určí, jaké nástroje pro virtualizaci používáme

1. Typ(bare metal)

- Použijeme nástroj jako Microsoft Hyper-V. Dostupný pouze na WS serveru a také na vyšších verzích WS(10, 11)
- Správa Hyper-V. Powershell nebo Hyper-V manager. Powershell využijeme, když chceme využít skriptování nebo automatizovat nějaké úlohy.
- Skrze Hyper-V poté spouštíme virtuály a to vše v rámci jednoho fyzického počítače

2. Typ(host hypervisor)

- Použijeme software/aplikaci jako Vmware workstation nebo VirtualBox
- Pro jejich správu využijeme jejich intuitivní rozhraní
- V těchto aplikacích vytvoříme virtuální počítače velmi jednoduše
- Microsoft Hyper-V je řešení pro podniky, zatímco Vmware workstation či VirtualBox je pro jednotlivce

UNIX-LIKE OS

1. Typ(bare metal)

- Využijeme nástroj KVM, který se často používá s nástrojem qemu
- Pro správu využijeme nástroje virsh nebo virtual manager

2. Typ(host hypervisor)

- Opět nástroje virtualbox nebo VMware workstation

Konfigurace a správa kontejnerů

- Nejznámější platformy:
 - Docker
 - Kubernetes

Docker

- Nejznámější platforma pro správu, nasazení a provoz kontejnerů
- Nabízí desítky kontejnerů, které jsou jako obrazy dostupné na Docker Hubu. Zde si tyto kontejnery můžeme stáhnout
- Je dostupný na Windows, Linux i Mac OS
- Ovládání: Rozhraní nebo příkazy
- Instalace: Web nebo repo na linuxu

Jak nakonfigurovat kontejner v Dockeru

1. Image(obraz)

- a. Kontejner vzniká z tzv. obrazu, který si buď sami vytvoříme nebo si ho stáhneme s docker hubu.
- b. Image je jakási šablona pro vytváření kontejnerů
- c. Image obsahuje nejmenší stavební jednotku(například operační systém, nebo samotnou aplikaci a všechny věci, co jsou k ní potřebné)

2. Dockerfile

- a. V docker file je specifikováno, jak má být image vytvořen a jaké soubory do něj mohou být kopírovány



Toto je vytvoření kontejneru, teď se podíváme na konfiguraci kontejneru

Porty kontejnerů

- kontejnery naslouchají na nějakých portech
- Když chceme tyto porty zpřístupnit zařízením z venku, musíme nastavit port forwarding
- Například náš kontejner má službu, co běží na portu 3000, ale to není volně dostupné
- Musíme nakonfigurovat port forwarding na nějaký dostupný port hostitele
- Například namapujeme port 3000 kontejneru na port 8080 hostitele
- Chceme poté přistoupit k tomu, tak zadáme například localhost a port 8080, tím se dostaneme na službu v kontejneru

Volumes

- Pokud chceme, aby se data ukládala i po vypnutí či smazání kontejneru, musíme nastavit volumes
- Propojíme souborový systém hostitele s kontejnerem

Sít

- Každý kontejner má vlastní IP adresu v rámci docker sítě
- Když chceme, aby kontejnery komunikovali mezi sebou v síti, například máme backend a fronted část aplikace, tak je propojíme skrze síťová rozhraní

Prostředí (Environment variables):

- Do kontejneru můžeš předávat proměnné prostředí – např. hesla, názvy databází nebo konfigurační volby.

Správa kontejnerů v dockeru

- Kontejner se spustí z image(jednorázově) nebo opakovaně
- Restart a automatické spuštění
- Zastavení
- Monitoring

Kubernetes

- Open source platforma pro orchestraci kontejnerů
- S dockerem pracují ruku v ruce, docker umí vytvářet, nasazovat a spravovat kontejnery, společně s kubernetes to může dělat ve velkém měřítku
- Kubernetes se dá použít pro lokální vývoj(minikube), v produkčním prostředí(kubeadm), ale taky v cloudu(aks-azure kubernetes services)
- Proč využívat kubernetes a co je vlastně orchestrace?
 - **Automaticky nasazovat** kontejnery
 - **Restartovat nebo nahradit** kontejnery, když selžou
 - **Škálovat** služby podle zátěže (nahoru/dolů)
 - **Vyvažovat zátěž** mezi kontejnery (load balancing)
 - **Spravovat konfiguraci a tajné údaje**
 - **Udržovat cílový stav** (declarative infrastructure – co má být, ne jak)

Pojmy v kubernetes

- Cluster
 - Skupina počítačů, na kterých běží kontejnery. Tyto počítače jsou navzájem propojené a používají se pro provoz a správu kontejnerizovaných aplikací
- Master Node
 - Hlavní počítač z clusteru, spravuje a řídí celý cluster
- Worker nodes
 - Virtuální počítače v clusteru, na kterých běží jednotlivé kontejnerizované aplikace