20. Virtualizace a kontejnery

**1. Virtualizace a kontejnery – definice, základní pojmy, účel**

**Virtualizace**

* **Definice:** Virtualizace je proces, který umožňuje oddělení fyzického hardwaru od softwarových prostředků. V podstatě se na jednom fyzickém serveru spouští více virtuálních strojů (VM), z nichž každý může běžet s vlastním operačním systémem.
* **Základní pojmy:**
  + **Hypervisor:** Softwarová vrstva, která spravuje a izoluje virtuální stroje. Existují dva typy:
    - **Typ 1 (bare-metal):** Přímý běh na hardwaru (např. VMware ESXi, Microsoft Hyper-V, Xen).
    - **Typ 2 (hostovaný):** Běží jako aplikace v hostitelském operačním systému (např. VirtualBox, VMware Workstation).
  + **Virtuální stroj (VM):** Emulovaný počítač, který má vlastní virtuální hardware, operační systém a aplikace.
* **Účel:**
  + Konsolidaci hardwaru a snížení nákladů.
  + Izolaci a bezpečné provozování více služeb na jednom fyzickém zařízení.
  + Dynamickou správu zdrojů a flexibilní nasazení systémů (např. při migraci mezi servery).
* Příkazy:
  + virsh (virsh --list all, virsh dominfo centos7vm, virsh edit centos7vm…)

**Kontejnery**

* **Definice:** Kontejnery poskytují lehkou virtualizační technologii na úrovni operačního systému, kde se aplikace a jejich závislosti balí do izolovaných prostředí. Na rozdíl od virtuálních strojů sdílí jádro hostitelského OS.
* **Základní pojmy:**
  + **Kontejnerový engine:** Software, který spravuje kontejnery (např. Docker, Podman).
  + **Obraz (Image):** Statická šablona obsahující aplikaci a její závislosti. Kontejnery se vytvářejí z obrazů.
  + **Registry:** Úložiště pro kontejnery (např. Docker Hub), odkud se obrazy stahují a spravují.
* **Účel:** Kontejnery slouží k:
  + Rychlému nasazení a škálování aplikací.
  + Izolaci prostředí, což zajišťuje, že aplikace běží konzistentně napříč různými prostředími (vývoj, testování, produkce).
  + Efektivní správě zdrojů díky nízké režii, jelikož kontejnery sdílí jádro hostitelského systému.

**2. Virtualizace v OS Windows a UNIX-like OS**

**Virtualizace v OS Windows**

* **Microsoft Hyper-V:**
  + Integrovaný hypervisor ve Windows Serveru i některých verzích Windows Pro.
  + Umožňuje vytváření a správu virtuálních strojů prostřednictvím grafického rozhraní (Hyper-V Manager) nebo PowerShellu.
* **Další řešení:**
  + **VMware Workstation/ESXi:** Komerční řešení, která jsou také k dispozici na platformě Windows.
  + **VirtualBox:** Open-source řešení, vhodné pro menší nasazení a testovací prostředí.

**Virtualizace v UNIX-like OS**

* **KVM (Kernel-based Virtual Machine):**
  + Integrovaná virtualizační technologie v linuxovém jádře, která umožňuje spouštění virtuálních strojů.
  + Správa se provádí pomocí nástrojů jako **libvirt**, **virt-manager** nebo příkazového řádku.
* **Xen:**
  + Další hypervisor pro Linux, využívaný zejména v datových centrech a cloudech.
* **Další řešení:**
  + **VMware ESXi:** Specializované hypervisorové řešení, běžící na serverovém hardwaru.
  + **LXC/LXD:** Technologie založené na kontejnerové virtualizaci, která poskytuje lehčí alternativu k plné virtualizaci.

**3. Konfigurace a správa kontejnerů**

**Kontejnery v UNIX-like OS a Windows**

* **Docker:**
  + Nejrozšířenější platforma pro kontejnery, která umožňuje vytváření, distribuci a správu kontejnerových obrazů.
  + **Konfigurace:**
    - Dockerfile: Textový soubor s instrukcemi pro sestavení obrazu.
    - Příkazy: docker build, docker run, docker pull, docker push pro správu obrazů a kontejnerů.
  + **Správa:**
    - Docker CLI a Docker Compose pro definici a orchestraci více kontejnerů.
    - Grafické nástroje jako Portainer umožňují správu kontejnerů přes webové rozhraní.
* **Podman a Buildah:**
  + Alternativy k Dockeru, zejména na Linuxu, které běží bez démonu a umožňují správu kontejnerů a obrazů.
* **Kubernetes:**
  + Orchestrace kontejnerizovaných aplikací ve velkém měřítku.
  + Umožňuje automatické nasazení, škálování, load balancing a správu životního cyklu kontejnerů.

**Specifika správy kontejnerů**

* **Izolace a networking:**
  + Kontejnery mohou sdílet síťové rozhraní nebo mít vlastní virtuální síť (bridge, host nebo overlay sítě).
* **Persistence dat:**
  + Datová svazky (volumes) a bind mounts umožňují uchování dat i po restartu kontejneru.
* **Bezpečnost:**
  + Izolace na úrovni procesů, omezení prostřednictvím profilů (např. AppArmor, SELinux) a řízení zdrojů (CPU, paměť).

Příklad runování:

* docker run -dit --name test-web -p 8080:80 -v /home/user/website/:/usr/local/apache2/htdocs/ httpd:2.4

**Shrnutí**

* **Virtualizace** umožňuje spouštět více izolovaných operačních systémů (virtuálních strojů) na jednom fyzickém hardwaru, zatímco **kontejnery** poskytují lehkou izolaci na úrovni operačního systému, kde se aplikace spouští společně s jádrem hostitelského systému.
* **V prostředí Windows** jsou nejčastěji využívána řešení jako Hyper-V, VMware a VirtualBox, zatímco v **UNIX-like OS** dominuje KVM, Xen a technologie jako LXC/LXD.
* **Kontejnerové technologie** jako Docker, Podman a Kubernetes umožňují rychlé a efektivní nasazení, škálování a správu aplikací. Konfigurace se provádí prostřednictvím Dockerfile, CLI příkazů, konfiguračních souborů a grafických nástrojů pro orchestraci.