

Docker je **open-source platforma** určená k vývoji, distribuci a provozu aplikací v tzv. **kontejnerech**. Kontejner je lehké a izolované prostředí, které obsahuje samotnou aplikaci i všechny její knihovny, konfigurace a další závislosti. Díky tomu lze aplikaci spustit na jakémkoliv počítači, kde je nainstalovaný Docker, a běží vždy stejně. Docker tak řeší dlouholetý problém softwarového vývoje, kdy se aplikace chovají jinak na různých strojích.

Na rozdíl od **virtuálních strojů**, které ke každé aplikaci vytvářejí celý další operační systém, používá Docker sdílený kernel hostitelského systému. To znamená, že kontejnery jsou mnohem menší, rychleji startují a mají výrazně nižší nároky na výkon. Kontejner je v podstatě jen izolovaný proces, nikoliv plnohodnotná instance systému. Díky této lehkosti lze spouštět i desítky kontejnerů zároveň, což umožňuje rychlé testování, jednoduché replikování prostředí a efektivní škálování aplikací.

Jednou z klíčových vlastností Dockeru je **izolace**. Každý kontejner běží odděleně a nemůže zasahovat do ostatních kontejnerů ani do samotného hostitelského systému. Izolace je zajištěna na úrovni procesů, jmenných prostorů a kontrolních skupin. To přináší bezpečnost, stabilitu a zároveň umožňuje provozovat různé verze stejných aplikací bez konfliktů. Docker navíc umožňuje omezit využití procesoru, paměti či diskového prostoru, takže vývojář může přesně řídit dostupné zdroje.

Základem Dockeru je **image**, což je neměnná šablona popisující, jak má aplikace vypadat. Z jednoho image lze vytvořit libovolný počet kontejnerů. Image se obvykle tvoří pomocí souboru **Dockerfile**, který obsahuje instrukce typu *FROM*, *RUN*, *COPY* či *CMD* a přesně popisuje kroky potřebné k sestavení prostředí. Samotné kontejnery pak představují běžící instanci tohoto image. Tato koncepce přímo podporuje moderní způsob vývoje, kde lze snadno verzovat prostředí aplikace a jednoduše ho sdílet mezi vývojáři.

Docker disponuje propracovaným systémem **síťování**, díky kterému mohou kontejnery spolu komunikovat napříč projekty nebo být izolované ve vlastních virtuálních sítích. Docker automaticky vytváří DNS názvy pro jednotlivé služby a vývojář tak může kontejner oslovit jednoduchým názvem místo IP adresy. Tato vlastnost výrazně usnadňuje provoz více propojených komponent, například aplikace, databáze a cache.

Velmi důležitým nástrojem je **Docker Compose**, který umožňuje definovat a spustit více služeb najednou prostřednictvím jediného YAML souboru. Vývojář v něm popíše celý systém – například backend, frontend a databázi – a jedním příkazem může sestavit a spustit kompletní prostředí. Compose také řeší propojení sítí, sdílení dat, načítání prostředí z *.env* a závislosti mezi službami. V praxi to zásadně zjednodušuje vývoj i testování složitějších aplikací.

Docker podporuje i **škálování**, tedy spouštění více instancí jedné služby. Bez zásahu do zdrojového kódu lze například vytvořit několik kontejnerů stejného backendu a rozdělovat mezi ně zátěž. Toto chování je základem distribuovaných systémů a cloudových služeb, které využívají horizontální škálování pro zpracování většího objemu požadavků.

Celý Docker funguje jako **client-server systém**: uživatel pracuje s Docker Klientem (CLI nebo Docker Desktop), který odesílá příkazy na **Docker Daemon**. Ten zajišťuje běh kontejnerů, správu image, sítí i disků. Mezi klientem a daemonelem probíhá komunikace přes **REST API**, a celý tento celek se označuje jako **Docker Engine**.

Docker se stal klíčovou technologií pro moderní vývoj, DevOps a cloudové služby. Běžně jej používají společnosti jako **Spotify**, **Netflix** či **PayPal**. V kombinaci s nástroji pro orchestraci, jako je **Kubernetes** nebo **Docker Swarm**, tvoří základ rozsáhlých a škálovatelných distribuovaných systémů.