Containerisierung

Inhaltsverzeichnis

- 1 Funktionsweise
- 2 Vorteile
- 3 Grenzen & Hinweise
- 4 Fazit

Containerisierung verpackt Anwendungen zusammen mit ihren Abhängigkeiten in portable Einheiten. So laufen Dienste isoliert und reproduzierbar – bei geteiltem Host-Kernel typischerweise mit Docker, Podman oder vergleichbaren Runtimes.

1 Funktionsweise

Ein Container teilt sich den Kernel des Host-Systems, isoliert aber Prozesse, Dateisysteme und Netzwerke. Images beschreiben den gewünschten Zustand; daraus starten laufende Container-Instanzen.

- **Image:** unveränderliche Vorlage einer Anwendung.
- **Container:** laufende Instanz aus einem Image.
- **Registry:** Speicherort für Images (z. B. Docker Hub).

2 Vorteile

- Gleiche Laufzeitumgebung in Entwicklung, Test und Produktion
- Schneller Start und geringerer Overhead als klassische VMs
- Einfachere Skalierung und Orchestrierung

3 Grenzen & Hinweise

- Keine vollständige Hardware-Virtualisierung – Kernel wird geteilt.
- Sicherheit hängt stark von Image-Qualität, Updates und Runtime-Härte ab.
- Persistente Daten gehören in Volumes, nicht in den Container selbst.

4 Fazit

Containerisierung ist der Standard für moderne Anwendungsbereitstellung: Anwendungen werden portabel, Umgebungen reproduzierbar und Betrieb sowie Skalierung deutlich planbarer.