

[zurück](#)

# Virtualisierung - Grundlagen (Hypervisor, VMs, Snapshots, Container)

Virtualisierung ermöglicht es, mehrere Betriebssysteme oder Dienste gleichzeitig auf derselben Hardware zu betreiben.

Sie trennt Hardware und Software logisch voneinander und sorgt für Flexibilität, Ausfallsicherheit und bessere Ressourcennutzung.

---

## 1. Warum Virtualisierung?

- mehrere Systeme auf einer Hardware
- bessere Auslastung (CPU, RAM)
- hohe Flexibilität
- schnelle Wiederherstellung (Snapshots)
- leichter Umzug (Migration)
- Isolation von Anwendungen
- Kostenersparnis (weniger Server)

Beispiele aus der Praxis:

- Proxmox-Cluster
- VMware ESXi
- Hyper-V
- Virtuelle Testumgebungen
- Docker-Container

---

## 2. Hypervisor - Typ 1 und Typ 2

Ein Hypervisor ist eine Software, die virtuelle Maschinen betreibt.

### Typ 1 - Bare-Metal Hypervisor

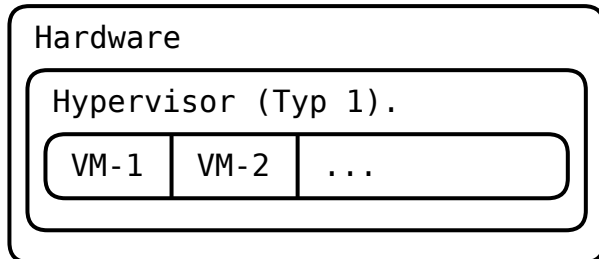
läuft direkt auf der Hardware:

- Proxmox VE
- VMware ESXi
- Microsoft Hyper-V (Server)
- XCP-ng

Vorteile:

- hohe Performance
- stabil und sicher
- ideal für Server

ASCII:



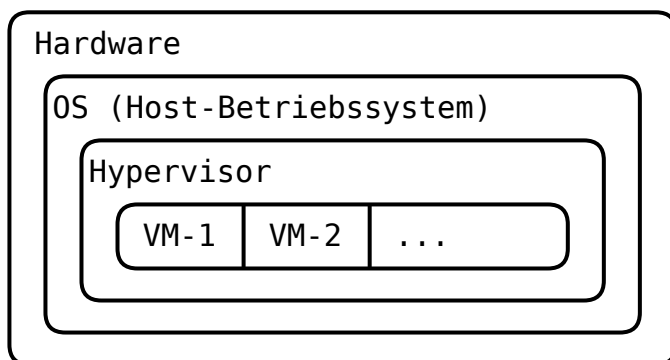
## Typ 2 - Hosted Hypervisor

läuft auf einem bestehenden Betriebssystem:

- VirtualBox
- VMware Workstation
- Parallels

Vorteile:

- gut für Tests & Schulung
- einfacher zu bedienen



---

## 3. Virtuelle Maschinen (VMs)

Eine VM ist ein vollständiges Betriebssystem innerhalb eines Hypervisors.

Enthält:

- virtuelle CPU(s)

- RAM
- virtuelle Festplatte (z. B. qcow2, vmdk, vhdx)
- virtuelle Netzwerkkarte
- BIOS/UEFI
- eigene Treiber

Beispiele:

- Windows Server 2022
- Debian / Ubuntu Linux
- Firewall-VM (OPNsense)
- Datenbankserver

Vorteile:

- komplett isoliert
  - \* beliebig duplizierbar
  - \* portabel

---

## 4. Snapshots

Snapshots speichern den Zustand einer VM zu einem Zeitpunkt.

Beinhalten:

- RAM (optional)
- Festplattenzustand
- Konfiguration

Nutzen:

- schnelle Rückkehr bei Fehlern
- ideal für Updates
- Test-Umgebungen

Wichtiger Hinweis:

- **Snapshots sind kein Backup**
- Snapshots können Speicher fressen, wenn sie lange existieren

ASCII:

```
Snapshot A → Snapshot B → Snapshot C
```

## 5. Live Migration / HA

Moderne Hypervisoren erlauben:

### Live Migration

VM läuft weiter, während sie auf einen anderen Host verschoben wird.

### HA - High Availability

Wenn ein Hypervisor ausfällt, übernimmt ein anderer.

Voraussetzung:

- Shared Storage
    - \* Cluster
    - \* funktionierendes Netzwerk
- 

## 6. Storage in Virtualisierung

Virtuelle Maschinen können verschiedene Datenträger nutzen:

- lokale SSD/NVMe
  - ZFS-Pools
    - \* Ceph (Cluster Storage)
    - \* NFS/SMB Storage
    - \* iSCSI-SAN
    - \* LVM-Thin

In Unternehmensumgebungen:

- Storage redundant
    - Snapshots & Backups automatisiert
-

# 7. Container vs Virtuelle Maschinen

Viele verwechseln Container mit VMs.

## Virtuelle Maschinen

- vollständiges Betriebssystem
  - hohe Isolation
    - \* Ressourcen sind „schwer“
    - \* ideal für Server, AD, Datenbanken

## Container

- teilen den Kernel mit dem Host
  - leichtgewichtig
    - \* sehr schnell
    - \* ideal für Microservices & Webapps

Beispiele:

- Docker
  - Podman
  - \* Kubernetes

ASCII:

```
VM:   Host → Hypervisor → vollständiges OS → App
Container: Host → Kernel → Container → App
```

## Vorteile von Containern

- starten in Sekunden
  - minimaler Ressourcenverbrauch
    - \* perfekt für DevOps und Microservices
    - \* ideal für Traefik, Nextcloud, Vaultwarden, Mailserver etc.

## Nachteile

- keine vollständige Isolation
  - Kernel wird geteilt
    - \* nicht ideal für klassische Windows-Dienste

—

## 8. Orchestrierung - Kubernetes, Docker Swarm

Bei vielen Containern wird ein Orchestrierer benötigt.

Kubernetes bietet:

- automatische Skalierung
  - \* Load Balancing
  - \* Self-Healing
  - \* Rolling Updates
  - \* Secret Management

Docker Swarm:

- einfachere Alternative
  - \* weniger komplex

—

## 9. Virtualisierung in der Praxis

Typische Einsatzszenarien:

- Proxmox-Cluster im Unternehmen
  - Windows AD Domain Controller in VMs
    - \* Web- & Datenbank-Server
    - \* Firewall-VM wie OPNsense
    - \* Entwicklungsumgebungen
    - \* Testumgebungen
    - \* Home-Lab (z. B. bei dir: Proxmox + Docker)

---

## 10. Sicherheit in der Virtualisierung

- Aktualisierte Hypervisor-Software
  - Netzwerktrennung (VLANs)
    - \* VM-Firewalls
    - \* Container nicht als root laufen lassen
    - \* Backups regelmäßig testen
    - \* Snapshots sparsam verwenden
    - \* Unnötige virtuelle Hardware entfernen

# Zusammenfassung

- Virtualisierung trennt Hardware von Software
  - Typ 1 Hypervisor = direkt auf Hardware (Proxmox, ESXi)
    - \* Typ 2 Hypervisor = auf Betriebssystem (VirtualBox)
    - \* VMs = vollständige Systeme
    - \* Container = leichtgewichtige Prozesse
    - \* Snapshots sind NICHT Backups
    - \* Orchestrierung (Kubernetes) für viele Container
    - \* Virtualisierung ist Grundlage moderner IT

From:

<http://wiki.nctl.de/dokuwiki/> - `Veni. Vidi. sudo rm -rf / vici.`

Permanent link:

<http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:grundlagen:netzwerkdienste:virtualisierung&rev=1764848592>

Last update: **04.12.2025 12:43**

