VPS - Einführung

Dieses Dokument dient als Nachschlagewerk und Ausgangslage für die Aufgaben zu Virtual Platforms. Es fasst die theoretischen Grundlagen zusammen, die für die Aufträge und Lernzielkontrollen benötigt werden.

Ausgangslage

Virtualisierung trennt ein IT-System von der direkt darunterliegenden physischen Hardware. Dadurch können mehrere virtuelle Systeme auf derselben Plattform betrieben werden. Eine Virtualisierungsplattform stellt virtuelle CPU, virtuellen Arbeitsspeicher, virtuelle Datenträger und virtuelle Netzwerke bereit.

Für Lernende ist wichtig, nicht nur einzelne Begriffe auswendig zu kennen, sondern die Zusammenhänge zu verstehen:

- Warum wird virtualisiert?
- Was wird virtualisiert?
- Welche Rolle spielt der Hypervisor?
- Welche Ressourcen brauchen virtuelle Maschinen?
- Wie werden Storage, Netzwerk, Sicherheit und Betrieb geplant?
- Warum ersetzen Snapshots keine Backups?
- Welche Kosten und Lizenzfragen entstehen?

Voraussetzungen

Für dieses Thema werden Grundlagen aus Betriebssystemen, Linux, Netzwerken und Storage benötigt.

Wichtige Begriffe:

- Prozess
- CPU und RAM
- Kernel und Userspace
- Treiber
- IP-Adresse, Subnetz und Gateway
- VLAN und Netzwerksegment
- Disk, Partition und Dateisystem
- CLI und einfache Diagnosebefehle

Warum virtualisieren?

Virtualisierung löst das Problem, IT-Systeme flexibel und isoliert auf gemeinsam genutzter Hardware zu betreiben.

Typische Gründe:

- bessere Auslastung vorhandener Hardware
- weniger physische Server
- schnellere Bereitstellung neuer Systeme
- isolierte Test- und Entwicklungsumgebungen
- einfacheres Zurückrollen vor riskanten Änderungen
- flexiblere Verschiebung von Workloads
- konsolidierter Betrieb und zentrale Verwaltung

Konsolidierung bedeutet, mehrere Systeme oder Server auf weniger physischer Hardware zu betreiben. Das ist möglich, weil viele Systeme CPU, RAM, Storage und Netzwerk nicht dauerhaft vollständig auslasten.

Virtualisierung ist nicht immer die beste Lösung. Direkter Hardwarezugriff, sehr niedrige Latenz, Spezialhardware oder besonders einfache Einzelinstallationen können gegen Virtualisierung sprechen.

Abstraktionsebenen

Virtualisierung kann auf unterschiedlichen Ebenen stattfinden.

Ebene	Beispiel	Beschreibung
Hardware-Virtualisierung	VM	Vollständige virtuelle Maschine mit eigener virtueller Hardware und eigenem Gastbetriebssystem.
OS-Level-Virtualisierung	Container	Isolierte Prozesse, die den Kernel des Hosts teilen.
Applikations-/Service-Virtualisierung	Dienst	Einzelne Anwendung oder Funktion wird abstrahiert bereitgestellt.

Eine VM startet ein eigenes Betriebssystem. Sie ist dadurch stärker getrennt, benötigt aber mehr Ressourcen und startet langsamer als ein Container.

Ein Container teilt sich den Kernel des Hosts. Er eignet sich gut für einzelne Dienste, portable Anwendungen und schnelle Entwicklungsumgebungen.

In diesem Bereich liegt der Fokus auf Hardware-Virtualisierung mit virtuellen Maschinen.

Hypervisor

Der Hypervisor ist die Virtualisierungsschicht. Er stellt VMs virtuelle CPU, RAM, Storage und Netzwerk bereit und vermittelt den Zugriff auf die physischen Ressourcen.

Es gibt zwei Grundtypen:

Typ	Beschreibung	Beispiele	Typischer Einsatz
Typ 1	läuft direkt auf der Hardware	Proxmox VE/KVM, VMware ESXi, Microsoft Hyper-V	Server, Rechenzentrum, zentrales Lab
Typ 2	läuft als Anwendung auf einem Host-Betriebssystem	VirtualBox, VMware Workstation	lokaler Client, Tests, Einzelarbeitsplatz

Bare Metal bedeutet, dass ein System direkt auf der physischen Hardware läuft. Ein Typ-1-Hypervisor ist deshalb für zentrale Virtualisierungsumgebungen meist besser geeignet als ein Typ-2-Hypervisor.

Mindestanforderungen

Eine Virtualisierungsplattform benötigt passende Hardware und aktivierte Funktionen.

Wichtige Voraussetzungen:

- CPU mit Virtualisierungserweiterungen wie Intel VT-x/EPT oder AMD-V/RVI
- aktivierte Virtualisierung im BIOS oder UEFI
- genügend RAM für Host und VMs
- genügend Storage für ISO-Dateien, virtuelle Disks, Templates, Snapshots und Backups
- passende Netzwerkanbindung
- Nested Virtualization, falls ein Hypervisor innerhalb einer VM betrieben wird

Der Host benötigt selbst ebenfalls Ressourcen. Deshalb dürfen CPU, RAM und Storage nicht vollständig für VMs verplant werden.

Eine schnelle CPU allein reicht nicht aus. RAM, Storage-I/O, Netzwerk und Füllstand des Datastores können ebenfalls begrenzen.

Ressourcen und Overcommitment

Ein Hypervisor verteilt physische Ressourcen auf virtuelle Maschinen.

Wichtige Begriffe:

- pCPU: physische CPU-Ressourcen des Hosts
- vCPU: virtuelle CPU, die einer VM zugewiesen wird
- RAM-Overcommitment: VMs erhalten zusammen mehr RAM, als physisch vorhanden ist
- Ballooning: Der Hypervisor fordert Speicher von VMs zurück
- Shares: Prioritäten bei Ressourcenknappheit
- Limits: maximale Ressourcennutzung
- Reservations: garantierte Mindestressourcen

Overcommitment funktioniert, weil nicht alle VMs gleichzeitig ihre Maximalressourcen nutzen. Es wird riskant, wenn viele VMs gleichzeitig hohe Last erzeugen oder kritische Systeme nicht genug Ressourcen erhalten.

Typische Symptome für Ressourcenprobleme:

- langsame VMs
- hohe CPU-Wartezeit
- Swapping
- hohe Disk-Latenz
- nicht reagierende Dienste
- volle Datastores

Monitoring ist deshalb ein zentraler Bestandteil des Betriebs.

Storage und virtuelle Disks

VMs verwenden virtuelle Disks. Diese liegen als Dateien oder Volumes auf einem Storage-System.

Wichtige Disk-Formate:

- VMDK
- QCOW2
- VHDX

Provisioning-Arten:

Art	Beschreibung	Vorteil	Risiko
Thin Provisioning	Speicher wird erst bei Nutzung belegt	spart Platz	Storage kann voll laufen
Thick Provisioning	Speicher wird direkt reserviert	planbarer	mehr Platzbedarf

Ein Datastore ist ein Speicherbereich für VM-Dateien, Images, Disks oder Snapshots.

Lokaler Storage reicht für Einzelhosts und kleine Labs oft aus. Geteilter Storage wie NFS, iSCSI oder Ceph ist sinnvoll, wenn mehrere Hosts, Live-Migration oder Hochverfügbarkeit benötigt werden.

Storage-I/O ist wichtig für die Performance. Hohe Latenz oder langsame Datenträger machen VMs spürbar langsam.

Lab-Setup

Beim Erstellen einer VM müssen Zweck, Ressourcen und Betriebssystem klar sein.

Typische Angaben:

- VM-Name
- Zweck
- Betriebssystem
- Installationsmedium
- vCPU
- RAM
- virtuelle Disk
- Netzwerk
- IP-Adresse oder DHCP
- Hostname
- Benutzerkonto
- Zugangsmethode

Ein ISO-Image wird als Installationsmedium verwendet. Die Boot-Reihenfolge entscheidet, ob zuerst vom ISO oder von der virtuellen Disk gestartet wird.

Nach der Installation gehören Grundkonfiguration und Dokumentation dazu:

- Hostname setzen
- Benutzer einrichten
- Updates installieren
- Netzwerk prüfen
- SSH oder Remote-Zugriff einrichten
- IP-Adresse und Zweck dokumentieren

Paravirtualisierung und Treiber

Virtuelle Maschinen verwenden virtuelle Geräte. Diese können emuliert oder paravirtualisiert sein.

Gerätetyp	Beschreibung	Vorteil	Nachteil
emuliert	bekannte Hardware wird nachgebildet	funktioniert oft ohne Zusatztreiber	weniger effizient
paravirtualisiert	für Virtualisierung optimierte Schnittstelle	bessere Performance	benötigt passende Treiber

VirtIO ist eine verbreitete paravirtualisierte Schnittstelle bei KVM und Proxmox. Sie wird häufig für Disk, Netzwerk und Ballooning verwendet.

Ein Guest-Agent verbessert die Kommunikation zwischen Gastbetriebssystem und Hypervisor. Er kann zum Beispiel IP-Adresse, Hostname, Statusinformationen und sauberes Herunterfahren unterstützen.

GPU-Passthrough bedeutet, dass eine physische GPU direkt an eine VM durchgereicht wird. Das ist ein fortgeschrittenes Thema und benötigt passende Hardware und Konfiguration.

Netzwerk und Isolation

VMs verwenden virtuelle Netzwerkkarten. Diese werden mit virtuellen Switches oder Bridges verbunden.

Wichtige Begriffe:

- Bridge: verbindet VMs mit einem Netzwerksegment
- NAT: VM kommuniziert über eine übersetzte Adresse
- Bridged Networking: VM ist direkt im gleichen Netzsegment wie der Host oder das physische Netzwerk sichtbar
- VLAN: logische Trennung von Netzwerken
- Segmentierung: Aufteilung in getrennte Bereiche mit kontrollierter Kommunikation

Netzwerkisolation ist wichtig, damit Testsysteme, produktionsnahe Systeme und sensible Dienste nicht unbeabsichtigt miteinander kommunizieren.

Erreichbarkeit kann geprüft werden mit:

```
ip addr
ip route
ping <ziel>
curl http://<ziel>
traceroute <ziel>
```

Eine Netzwerkdokumentation enthält VM, IP-Adresse, Subnetz, Gateway, VLAN, Bridge, erlaubte Kommunikation und Zweck.

Templates, Klone und Automatisierung

Templates und Klone beschleunigen die Bereitstellung neuer VMs.

Wichtige Begriffe:

- Golden Image: vorbereitete Referenzinstallation
- Template: Vorlage für neue VMs
- Full Clone: vollständige unabhängige Kopie
- Linked Clone: Klon, der von einer Basisdisk abhängt

- cloud-init: automatische Erstkonfiguration
- Infrastructure as Code: Infrastruktur wird über Dateien oder Code beschrieben

In ein Golden Image gehören:

- Betriebssystem
- Updates
- Basispakete
- Zeitzone
- Guest-Agent
- SSH-Server
- bereinigte Grundkonfiguration

Nicht ins Template gehören:

- Passwörter
- private Schlüssel
- persönliche Daten
- feste IP-Adressen
- systemspezifische IDs

Automatisierung spart Zeit, reduziert Fehler und macht Umgebungen reproduzierbar.

Snapshots

Ein Snapshot hält den Zustand einer VM zu einem bestimmten Zeitpunkt fest. Änderungen nach dem Snapshot werden separat gespeichert, zum Beispiel in Delta-Dateien oder über Copy-on-Write.

Typische Einsatzfälle:

- vor Updates
- vor Konfigurationsänderungen
- vor riskanten Tests
- als kurzfristige Rückfallmöglichkeit

Rollback bedeutet, zu einem früheren Zustand zurückzukehren.

Wichtig: Ein Snapshot ist kein Backup. Er liegt meist auf derselben Plattform und ist abhängig von VM und Storage. Lange Snapshot-Ketten benötigen Speicherplatz und können die Performance verschlechtern.

Snapshots sollten zeitnah gelöscht oder konsolidiert werden, wenn sie nicht mehr benötigt werden.

Backup und Restore

Ein Backup ist eine eigenständige Sicherung zur Wiederherstellung nach Datenverlust, Defekt, Fehlbedienung oder Angriff.

Unterschied Snapshot und Backup:

Kriterium	Snapshot	Backup
Zweck	kurzfristiger Rücksprung	Wiederherstellung nach Verlust
Abhängigkeit	meist gleiche Plattform	möglichst unabhängig
Dauer	kurzzeitig	nach Aufbewahrungskonzept

Kriterium	Snapshot	Backup
Prüfung	Rollback	Restore-Test

Ein konsistentes Backup enthält Daten in einem brauchbaren Zustand. Bei Datenbanken oder laufenden Anwendungen muss darauf geachtet werden, dass keine unvollständigen Schreibvorgänge gesichert werden.

Die 3-2-1-Regel beschreibt:

- 3 Kopien
- 2 unterschiedliche Medien oder Speicherarten
- 1 Kopie extern oder offline

Ein Restore-Test prüft, ob Backups tatsächlich wiederhergestellt werden können.

Wichtige Planungsbegriffe:

- Recovery Point Objective: maximal akzeptierter Datenverlust
- Recovery Time Objective: maximal akzeptierte Wiederherstellungszeit

Hochverfügbarkeit und Live-Migration

Virtualisierung entkoppelt eine VM teilweise von der physischen Hardware. Dadurch kann eine VM unter passenden Voraussetzungen auf einem anderen Host laufen.

Wichtige Begriffe:

- Cluster: mehrere Hosts werden gemeinsam verwaltet
- Live-Migration: eine laufende VM wird auf einen anderen Host verschoben
- Hochverfügbarkeit: VM wird bei Host-Ausfall auf einem anderen Host neu gestartet
- geteilter oder replizierter Storage: VM-Daten sind für mehrere Hosts erreichbar

Live-Migration benötigt kompatible Hosts, stabiles Netzwerk, erreichbaren Storage und genügend Ressourcen auf dem Zielhost.

HA ist nicht dasselbe wie Backup. HA erhöht die Verfügbarkeit, schützt aber nicht vor Datenverlust, Fehlbedienung, beschädigten Daten oder Angriffen.

Sicherheit und Isolation

Der Hypervisor ist sicherheitskritisch, weil er viele VMs und zentrale Ressourcen kontrolliert. Ein Angriff auf den Hypervisor kann grosse Auswirkungen haben.

Wichtige Risiken:

- kompromittierter Managementzugang
- ungepatchter Hypervisor
- zu breite Administratorrechte
- ungetrennte Netzwerke
- manipulierte oder gelöschte Backups
- Fehlkonfigurationen

Grundmassnahmen:

- Hypervisor regelmässig patchen
- Managementzugänge absichern
- starke Authentifizierung verwenden

- Rechte nach dem Least-Privilege-Prinzip vergeben
- Netzwerke segmentieren
- unnötige Dienste deaktivieren
- Logging und Monitoring aktivieren
- Backups getrennt und geschützt aufbewahren

VM-Escape beschreibt das Konzept, dass ein Angreifer aus einer VM auf den Hypervisor oder andere VMs ausbrechen könnte.

Least Privilege bedeutet, dass Benutzer nur die Rechte erhalten, die sie für ihre Aufgabe benötigen.

Betrieb, Monitoring und Troubleshooting

Day-2-Betrieb bezeichnet den laufenden Betrieb nach der Bereitstellung. Dazu gehören Wartung, Monitoring, Updates, Backups, Fehlerbehebung und Dokumentation.

Wichtige Metriken:

- CPU-Auslastung
- CPU-Ready
- RAM-Auslastung
- Swapping
- Disk-I/O
- Disk-Latenz
- Netzwerkdurchsatz
- Paketverlust
- Storage-Füllstand
- Logs von Hypervisor, VM und Applikation

Troubleshooting beginnt mit einer klaren Beschreibung:

- Was ist das Symptom?
- Seit wann tritt es auf?
- Wer oder was ist betroffen?
- Gab es Änderungen?
- Welche Metriken zeigen Auffälligkeiten?
- Welche Logs bestätigen oder widerlegen die Vermutung?

Bei der Fehlersuche sollte immer nur eine Änderung oder ein Test auf einmal durchgeführt werden. Sonst ist nicht klar, welche Massnahme die Wirkung verursacht hat.

Eine kurze Troubleshooting-Dokumentation enthält Symptom, Zeitpunkt, betroffene Systeme, geprüfte Metriken, Befund, Hypothese und nächsten Schritt.

Lizenzierung und Wirtschaftlichkeit

Virtualisierung beeinflusst Kosten und Lizenzierung. Neben der Hypervisor-Lizenz entstehen weitere Kosten.

Typische Lizenzmodelle:

- pro Core
- pro Socket
- pro Host
- pro VM
- Subscription

- Supportvertrag

Kostenfaktoren:

- Hardware
- Storage
- Netzwerk
- Strom und Kühlung
- Backup
- Monitoring
- Betrieb
- Schulung
- Support
- Lizenzen für Betriebssysteme und Anwendungen

Konsolidierung kann Kosten senken, weil weniger physische Server benötigt werden. Sie kann Lizenzkosten aber auch erhöhen, wenn nach Cores, Hosts, Sockets oder VMs lizenziert wird.

Open-Source-Lösungen sind nicht automatisch kostenlos im Betrieb. Know-how, Wartung, Support, Hardware und Ausfallrisiken verursachen trotzdem Aufwand und Kosten.

Vor einer Beschaffung sollten mindestens Workloads, Anzahl Hosts, Anzahl VMs, CPU-Kerne, Supportbedarf, Backup, HA, Wachstum, Budget und rechtliche Lizenzbedingungen geklärt werden.

Mini-Check

Die folgenden Fragen helfen bei der Selbstkontrolle:

- Kann ich erklären, welches Problem Virtualisierung löst?
- Kann ich VM, Container und Service-Virtualisierung unterscheiden?
- Kann ich Typ-1- und Typ-2-Hypervisoren einordnen?
- Kann ich CPU, RAM, Storage und Netzwerk für eine kleine VM grob planen?
- Kann ich Thin und Thick Provisioning erklären?
- Kann ich eine VM-Grundkonfiguration dokumentieren?
- Kann ich erklären, warum VirtIO oder Guest-Agent nützlich sind?
- Kann ich VM-Netzwerke verbinden oder isolieren?
- Kann ich Snapshot und Backup klar unterscheiden?
- Kann ich erklären, wofür HA und Live-Migration genutzt werden?
- Kann ich Sicherheitsrisiken des Hypervisors benennen?
- Kann ich eine langsame VM strukturiert eingrenzen?
- Kann ich grundlegende Kosten- und Lizenzfaktoren nennen?