

yoTech24

IT-Grundlagen

Lernportal-Zusammenfassung · 89 Kapitel · LB-01 bis LB-10 & LB-13

Netzwerk, Betriebssysteme, Hardware, Server & Dienste, Virtualisierung, Storage & Backup, IT-Security, Cloud, Monitoring & Management, Scripting & Automatisierung sowie Ausbildung & Soft Skills als kompakte Nachschlage-Zusammenfassung.

Diese PDF fasst die Kernpunkte zusammen. Die vollständigen interaktiven Folien inkl. Audio-Dialogen (Podcast) finden sich unter yotech24.com/akademie/



LB-01 · Netzwerk

12 Kapitel

Netzwerk-Grundlagen & OSI-Modell

Die sieben Schichten als roter Faden jeder Fehlersuche

Warum das OSI-Modell?

- Weißt du die Schicht, kennst du das zu prüfende Gerät/den Dienst
- Fehlersuche wird systematisch statt Raten
- In der Prüfung garantiert relevant

Es ist kein trockener Stoff, sondern dein wichtigstes Diagnosewerkzeug.

Sieben Schichten

OSI = Open Systems Interconnection: sieben Schichten, jede mit klarer Aufgabe, jede spricht mit ihren Nachbarn.

- Herstellerunabhängige Standardisierung
- Fehler lassen sich einer Schicht zuordnen

Schichten 1–4

- L1 Bitübertragung: Kabel, Funk, Glasfaser
- L2 Sicherung: Switch, MAC-Adressen, Ethernet
- L3 Vermittlung: Router, IP-Adressen, Subnetting
- L4 Transport: TCP und UDP

Schichten 5–7

- L5 Sitzung: Auf-/Abbau von Verbindungen
- L6 Darstellung: Kodierung, Verschlüsselung
- L7 Anwendung: HTTP, DNS, SMTP, SSH
- In der Praxis oft als „Anwendungsschichten“ zusammengefasst

Geräte pro Schicht

Switch (L2)

Leitet anhand der MAC-Adresse im lokalen Netz.

Router (L3)

Verbindet Netze anhand der IP-Adresse.

- Alter Hub arbeitete auf L1 (heute abgelöst)
- Firewalls/NGFW prüfen bis in die oberen Schichten

Kollisions- vs. Broadcast-Domäne

Jeder Switch-Port ist eine eigene Kollisionsdomäne – Kollisionen gibt es praktisch nicht mehr.

- Ohne VLANs ist der ganze Switch eine Broadcast-Domäne
- Ein VLAN = eine eigene Broadcast-Domäne
- Kleinere Domänen = weniger Last

TCP vs. UDP

TCP

Zuverlässig, verbindungsorientiert, bestätigt jede Lieferung.

UDP

Schnell, verbindungslos – für Streaming, VoIP, DNS.

- TCP für Korrektheit, UDP für Tempo

Ports: die Türnummern der Anwendungen

Bekannte Ports

80 HTTP · 443 HTTPS · 22 SSH · 53 DNS · 25 SMTP · 3389 RDP

Praxisnutzen

Blockiert die Firewall Port 443, scheitern alle HTTPS-Verbindungen – gezielt statt geraten.

- Ports 0–1023: „Well-Known Ports“, meist Systemdienste
- Ein offener Port ohne laufenden Dienst dahinter bringt nichts

Eine IP-Adresse findet das Gerät, ein Port findet den Dienst darauf.

Fehlersuche von unten nach oben

Kabel → IP-Konfiguration → Namensauflösung → Anwendung – genau in dieser Reihenfolge.

- L1: Kabel/Link prüfen
- L3: IP und Gateway prüfen
- L7: DNS und Anwendung prüfen

Das Wichtigste

- Sieben Schichten, jede mit klarer Aufgabe
- Switch auf L2, Router auf L3
- VLAN = eigene Broadcast-Domäne
- Fehlersuche immer von unten nach oben

IP-Adressierung & Subnetting

IPv4, CIDR, private Bereiche, Subnetzrechnung und IPv6

Wozu IP-Adressen?

- Ermöglicht Routing über Netzgrenzen hinweg
- Besteht aus Netz-Anteil und Host-Anteil
- Ergänzt die feste MAC-Adresse (Layer 2)

Jedes Gerät im Netz braucht eine eindeutige logische Adresse (Layer 3).

Aufbau einer IPv4-Adresse

32 Bit, geschrieben als vier Oktette: 192 . 168 . 10 . 25 – jedes Oktett 0–255.

- Die Subnetzmaske trennt Netz von Host
- 255 . 255 . 255 . 0 = die ersten 24 Bit sind Netz
- Netzadresse (alle Host-Bits 0) und Broadcast (alle 1) sind reserviert

CIDR-Notation

192 . 168 . 10 . 0 / 24 ist gleichbedeutend mit Maske 255 . 255 . 255 . 0 .

- /24 → 256 Adressen, davon 254 nutzbar
- /25 → 128, /26 → 64, /27 → 32
- Je größer der Präfix, desto kleiner das Netz

Statt langer Masken schreibt man die Anzahl der Netz-Bits.

Private Adressbereiche (RFC 1918)

Klasse A

10.0.0.0/8 – große Netze.

Klasse B/C

172.16.0.0/12 · 192.168.0.0/16 – Heim/KMU.

- Nicht im Internet routbar → Übergang per NAT
- Frei intern verwendbar

Subnetting – ein Beispiel

/24 → /26: 2 zusätzliche Netz-Bits = 4 Subnetze à 64 Adressen (62 nutzbar).

- .0–.63 · .64–.127 · .128–.191 · .192–.255
- Jeweils erste = Netz, letzte = Broadcast
- Merke: Blockgröße = 256 – Maskenwert des letzten Oktetts

192.168.1.0/24 in vier gleiche Subnetze teilen.

VLSM

- Große Netze für viele Hosts, kleine (/30) für Punkt-zu-Punkt-Links
- Spart Adressen gegenüber gleich großen Subnetzen
- Von groß nach klein planen, um Überschneidungen zu vermeiden

Variable Length Subnet Mask – Subnetze unterschiedlicher Größe.

IPv6 in Kürze

128 Bit, hexadezimal: 2001:db8::1 – genug Adressen für praktisch alles.

- Kein NAT nötig – globale Adressen für alle Geräte
- Führende Nullen kürzbar, :: ersetzt einen Null-Block
- Link-Local fe80::/10 immer vorhanden

Subnetting in der Praxis

Blockgröße = 256 minus Maskenwert im letzten Oktett – damit findest du Netz, erste/letzte Host-Adresse und Broadcast.

- Beispiel /27: Blockgröße 32 → .0, .32, .64 ...
- Netzadresse ist die erste, Broadcast die letzte im Block
- Nutzbare Hosts = Adressen minus 2

IPv6 im Alltag

- Standard-Präfix /64 pro Subnetz
- Adressbezug per SLAAC oder DHCPv6
- Link-Local (fe80::) immer vorhanden
- Dual-Stack: IPv4 und IPv6 parallel

Das Wichtigste

- IP = Netz-Anteil + Host-Anteil, getrennt durch die Maske
- CIDR /n zählt die Netz-Bits
- RFC-1918-Adressen sind privat → NAT ins Internet
- Subnetting: Blockgröße = 256 – Maskenoktett

Switching & VLANs

MAC-Tabelle, Broadcast-Domänen, 802.1Q und Inter-VLAN-Routing

Was ein Switch tut

- Leitet Frames anhand der Ziel-MAC gezielt weiter
- Lernt MAC-Adressen automatisch aus eingehenden Frames
- Ersetzt den alten Hub (der stumpf an alle sendete)

Ein Switch arbeitet auf Layer 2 und verbindet Geräte im selben Netz.

Die MAC-Adresstabelle

Der Switch merkt sich: welche MAC-Adresse hängt an welchem Port.

- Unbekanntes Ziel → Frame wird geflutet (an alle Ports)
- Einträge altern aus (Aging Timer)
- Grundlage für effizientes, gezieltes Weiterleiten

Kollisions- vs. Broadcast-Domäne

Kollisionsdomäne

Jeder Switch-Port ist eine eigene – keine Kollisionen mehr (Full Duplex).

Broadcast-Domäne

Ohne VLANs erstreckt sie sich über den ganzen Switch.

- Broadcasts erreichen alle Geräte der Domäne
- Große Broadcast-Domänen = mehr Last und Risiko

VLANs

Ein VLAN = eine eigene Broadcast-Domäne, unabhängig vom physischen Standort.

- Segmentierung nach Abteilung, Funktion oder Sicherheit
- Weniger Broadcast-Last, bessere Isolation
- VLAN-übergreifend geht nur über einen Router (Layer 3)

Virtuelle LANs unterteilen einen Switch in mehrere logische Netze.

Access- vs. Trunk-Port

Access-Port

Gehört zu genau einem VLAN – für Endgeräte.

Trunk-Port

Transportiert mehrere VLANs – zwischen Switches/zum Router.

- Trunks kennzeichnen Frames per VLAN-Tag
- Endgerät bemerkt vom VLAN nichts

802.1Q-Tagging

802.1Q fügt ein 4-Byte-Tag mit der VLAN-ID (1–4094) in den Frame ein.

- Nur auf Trunks – Access-Ports senden ungetaggt
- Native VLAN wird ungetaggt übertragen (Standard: VLAN 1)
- Native VLAN auf beiden Trunk-Enden gleich setzen

Inter-VLAN-Routing

- Router-on-a-Stick: ein Trunk zum Router mit Sub-Interfaces
- Layer-3-Switch: routet VLANs intern, deutlich schneller
- Ohne Routing bleiben VLANs vollständig getrennt

Damit VLANs miteinander reden, braucht es Routing.

Managed L3-Switches in der Praxis

Zentrale Verwaltung: on-prem per eigenem Controller/Stack, oder cloudbasiert wie Cisco Meraki, Aruba Central/Instant On, Ubiquiti UniFi, TP-Link Omada – bequemer, aber mit laufenden Lizenzkosten und Herstellerbindung.

- SNMP fürs Monitoring, PoE für Access Points/Telefone, QoS für priorisierten Voice-/Videoverkehr
- ACLs filtern Verkehr direkt auf dem Switch
- LACP (mehrere Uplinks bündeln) und Stacking (mehrere Switches als eine logische Einheit)

Ein Managed Switch kann weit mehr als VLANs und Routing.

VLAN-Design

- Ein VLAN je Funktion (Nutzer, Voice, Management, Gäste)
- Management-VLAN vom Nutzerverkehr trennen
- Default-VLAN 1 möglichst nicht produktiv nutzen
- Konsistente VLAN-IDs im ganzen Netz

Häufige Fehler

Der Klassiker: Ein Gerät hängt am falschen Port-Typ oder das VLAN ist auf dem Trunk nicht erlaubt.

- Access- und Trunk-Port verwechselt
- Native-VLAN auf beiden Trunk-Enden unterschiedlich
- VLAN nicht in der erlaubten Trunk-Liste

Das Wichtigste

- Switch lernt MACs und leitet gezielt weiter
- VLAN = eigene Broadcast-Domäne
- Access = ein VLAN, Trunk = viele (802.1Q-Tag)
- VLAN-übergreifend nur per Layer-3-Routing

STP, RSTP & Link Aggregation

Schleifen verhindern und Leitungen bündeln

Das Problem: Layer-2-Schleifen

Ein Broadcast-Storm kann ein ganzes Netz in Sekunden lahmlegen.

- Frames kreisen endlos (kein TTL auf Layer 2)
- MAC-Tabellen werden instabil
- Lösung: Spanning Tree Protocol

Redundante Switch-Verbindungen erzeugen ohne Schutz Endlosschleifen.

Spanning Tree (802.1D)

- Redundante Pfade werden logisch blockiert
- Fällt ein Pfad aus, wird ein Backup aktiviert
- Kommunikation über BPDUs zwischen Switches

STP baut aus dem vermaschten Netz einen schleifenfreien Baum.

Root-Bridge-Wahl

Der Switch mit der niedrigsten Bridge-ID (Priority + MAC) wird Root-Bridge.

- Alle Pfade werden relativ zur Root berechnet
- Priorität gezielt setzen, um die Root zu bestimmen
- Root sollte ein zentraler, leistungsfähiger Switch sein

Port-Rollen & Zustände

Rollen

Root-Port, Designated-Port, Blocking-Port.

Zustände (STP)

Blocking → Listening → Learning → Forwarding.

- Klassisches STP konvergiert langsam (30–50 s)
- Deshalb der Nachfolger RSTP

RSTP (802.1w)

Rapid STP konvergiert in Sekunden statt in fast einer Minute.

- Zusätzliche Portrollen (Alternate, Backup)
- Schnelle Umschaltung über Handshakes
- Heute Standard; abwärtskompatibel zu STP

Link Aggregation (LAG)

- Mehr Bandbreite und Ausfallsicherheit
- Kein STP-Blocking – alle Links aktiv
- Aushandlung per LACP (802.3ad)

Mehrere physische Links werden zu einem logischen gebündelt.

STP absichern

An Endgeräte-Ports gehören PortFast plus BPDU Guard – so bleibt der Baum stabil.

- PortFast: schneller Link-Up für Endgeräte
- BPDU Guard: sperrt Port, wenn dort ein Switch auftaucht
- Root Guard schützt die gewählte Root-Bridge

LAG in der Praxis

- Beide Seiten identisch konfigurieren
- LACP active/passive richtig kombinieren
- Lastverteilung erfolgt pro Fluss (Hash), nicht pro Paket
- Nur gleichartige Links bündeln

Das Wichtigste

- STP verhindert Layer-2-Schleifen durch Blockieren
- Root-Bridge = niedrigste Bridge-ID
- RSTP konvergiert schnell (Standard)
- LAG/LACP bündelt Links für Bandbreite + Redundanz

Routing-Grundlagen

Wie Pakete den Weg zwischen Netzen finden

Was ist Routing?

- Entscheidung anhand der Ziel-IP-Adresse
- Jedes durchquerte Gerät ist ein Hop
- Grundlage für das gesamte Internet

Router verbinden verschiedene Netze und leiten Pakete auf Layer 3 weiter.

Die Routing-Tabelle

Für jedes Paket sucht der Router den passenden Eintrag – das beste Ziel-Netz.

- Einträge: Zielnetz, nächster Hop, Interface
- Direkt verbundene Netze sind automatisch bekannt
- Der Rest kommt über statische oder dynamische Routen

Statisches Routing

- Voll kontrollierbar, kein Protokoll-Overhead
- Aufwändig und fehleranfällig bei vielen Netzen
- Ideal für kleine, stabile Umgebungen

Der Administrator trägt Routen von Hand ein.

Die Default-Route

`0.0.0.0/0` – „alles, was ich sonst nicht kenne, hierhin“ (Richtung Internet).

- Meist zum Provider/Gateway
- Spart unzählige Einzelrouten
- Longest-Prefix-Match: spezifischere Route gewinnt immer

Longest-Prefix-Match

Ziel `10.1.1.5` → `10.1.1.0/24` schlägt `10.0.0.0/8`.

- Spezifischer = genauer = bevorzugt
- Default-Route ist die unspezifischste (Länge 0)

Passen mehrere Routen, gewinnt die mit dem längsten Präfix.

Administrative Distanz

Niedrig = vertrauenswürdig

Direkt verbunden (0), statisch (1).

Höher

OSPF (110), RIP (120) – wird nachrangig genutzt.

- Erst Longest-Prefix, dann bei Gleichstand die AD

Kommen Routen aus verschiedenen Quellen, entscheidet die Vertrauenswürdigkeit.

Statisch vs. dynamisch

- Statisch: einfach, kontrolliert, wartungsintensiv
- Dynamisch: passt sich automatisch an Ausfälle an
- Große Netze brauchen dynamische Protokolle (nächstes Kapitel)

Routing-Tabelle lesen

- Spalten: Zielnetz, nächster Hop, Interface, Metrik
- Herkunft: connected, statisch oder dynamisch
- Bei Gleichstand entscheidet die administrative Distanz
- Default-Route als Auffangnetz

Routing-Troubleshooting

Ein häufiger Fehler: die Hinroute existiert, aber die Rückroute fehlt – Routing braucht beide Richtungen.

- Mit ping und traceroute eingrenzen
- Prüfen, ob das Zielnetz überhaupt bekannt ist
- Default-Route und Gateway kontrollieren

Das Wichtigste

- Router leiten anhand der Ziel-IP weiter
- Longest-Prefix-Match bestimmt die Route
- Default-Route `0.0.0.0/0` fängt alles Unbekannte
- Administrative Distanz entscheidet zwischen Quellen

Dynamisches Routing

OSPF, BGP und EIGRP – Routen automatisch lernen

Warum dynamisch?

- Reagiert selbstständig auf Ausfälle und neue Netze
- Skaliert dort, wo statische Routen unpraktikabel werden
- Kostet etwas CPU, RAM und Bandbreite

Router tauschen Routing-Informationen automatisch aus.

Zwei Grundprinzipien

Distance-Vector

Kennt nur Richtung + Distanz („Hörensagen“).
Beispiel: RIP.

Link-State

Jeder kennt die komplette Topologie und rechnet selbst. Beispiel: OSPF.

- Link-State konvergiert schneller und stabiler
- Distance-Vector ist einfacher, aber langsamer

OSPF

Link-State-Protokoll, berechnet kürzeste Pfade nach Kosten (Bandbreite).

- Gliederung in Areas für Skalierbarkeit
- Schnelle Konvergenz bei Änderungen
- Offener Standard, herstellerübergreifend

Open Shortest Path First – der Standard im Unternehmensnetz.

BGP

- Verbindet autonome Systeme (AS) verschiedener Provider
- Pfadvektor: trifft Entscheidungen nach Richtlinien, nicht nur Distanz
- Extrem skalierbar, aber langsam konvergierend

Border Gateway Protocol – das Routing-Protokoll des Internets.

EIGRP

- Schnelle Konvergenz durch das DUAL-Verfahren
- Lange proprietär, inzwischen teils offengelegt
- Vor allem in reinen Cisco-Umgebungen

Cisco-Protokoll, Mischform (Advanced Distance-Vector).

Vergleich

Intern (IGP)

OSPF, EIGRP – innerhalb einer Organisation.

Extern (EGP)

BGP – zwischen Organisationen/Providern.

- Faustregel: OSPF im Haus, BGP nach draußen

OSPF-Begriffe

- Area 0 ist das Backbone, alle Areas hängen daran
- Router-ID identifiziert jeden Router eindeutig
- Kosten richten sich nach der Bandbreite
- Nachbarschaften (Adjacencies) müssen zustande kommen

Konvergenz & Stabilität

- Schnelle Neuberechnung bei Topologie-Änderungen
- Route Summarization reduziert die Tabellengröße
- Passive Interfaces dort, wo keine Nachbarn sind
- Authentifizierung der Routing-Updates

Das Wichtigste

- Dynamisch = Router lernen Routen selbst
- OSPF: Link-State, Standard im LAN/WAN
- BGP: hält das Internet zusammen (AS-übergreifend)
- Link-State konvergiert schneller als Distance-Vector

NAT & PAT

Adressumsetzung zwischen privaten und öffentlichen Netzen

Warum NAT?

IPv4 ist knapp – NAT lässt viele private Geräte eine öffentliche Adresse teilen.

- Verbirgt die interne Struktur nach außen
- Kein direkter Zugriff von außen (Nebeneffekt: etwas Schutz)
- Findet meist am Router/Firewall statt

Network Address Translation übersetzt private in öffentliche IP-Adressen.

Static NAT

- Für Server, die dauerhaft erreichbar sein sollen
- Vorhersehbar, aber verbraucht je eine öffentliche IP
- Beide Richtungen möglich

Feste 1:1-Zuordnung einer privaten zu einer öffentlichen Adresse.

Dynamic NAT

- Erste-komme-erste-Adresse aus dem Pool
- Sind alle vergeben, muss gewartet werden
- Immer noch 1:1 pro aktiver Verbindung

Zuordnung aus einem Pool öffentlicher Adressen – je nach Bedarf.

PAT / NAT-Overload

Viele interne Geräte teilen sich EINE öffentliche IP – unterschieden über Portnummern.

- Der häufigste Fall zu Hause und im KMU
- Der Router führt eine Übersetzungstabelle (IP:Port)
- Auch „Masquerading“ genannt (Linux)

Port-Forwarding

Öffentlich :443 → intern 192.168.1.10:443 (Webserver).

- Nötig, damit interne Server von außen erreichbar sind
- Nur die freigegebenen Ports öffnen
- Sicherheitsrelevant – sparsam einsetzen

Destination-NAT – Zugriffe von außen auf einen internen Dienst leiten.

Grenzen von NAT

- Erschwert Ende-zu-Ende-Verbindungen (z. B. VoIP, P2P)
- Ist KEINE vollwertige Firewall
- IPv6 macht NAT weitgehend überflüssig

NAT-Varianten im Überblick

Source-NAT

Ändert die Quelladresse – der typische Weg nach außen.

Destination-NAT

Ändert die Zieladresse – Port-Forwarding von außen nach innen.

- Hairpinning: interner Zugriff auf den eigenen öffentlichen Dienst
- Carrier-Grade-NAT beim Provider

NAT-Troubleshooting

- Übersetzungstabelle des Routers prüfen
- Port-Forward gezielt von außen testen
- Doppelte Portzuordnungen vermeiden
- Protokolle wie VoIP brauchen ggf. ALG/Helper

Das Wichtigste

- NAT übersetzt privat ↔ öffentlich
- PAT teilt eine IP über Ports – der Alltag
- Port-Forwarding holt Dienste von außen erreichbar
- IPv6 braucht kein NAT mehr

WLAN-Technologien

Standards, Frequenzen, Sicherheit und Architektur

WLAN-Grundlagen

- Geteiltes Medium – Geräte funken auf denselben Kanälen
- Half-Duplex: senden und empfangen nicht gleichzeitig
- Reichweite und Tempo hängen stark von der Umgebung ab

Drahtlose Netze nach der Standardfamilie IEEE 802.11.

Die Standards

Klassisch

a/b/g/n – bis Wi-Fi 4 (802.11n).

Modern

ac (Wi-Fi 5), ax (Wi-Fi 6/6E), be (Wi-Fi 7).

- Höhere Standards = mehr Durchsatz und Effizienz
- Abwärtskompatibel, aber langsamste Clients bremsen

Frequenzbänder

2,4 GHz: mehr Reichweite, mehr Störungen · 5 GHz: schneller, kürzere Reichweite.

- 2,4 GHz nur 3 überlappungsfreie Kanäle (1, 6, 11)
- 5 GHz viele Kanäle, weniger Störung
- 6 GHz (Wi-Fi 6E) bringt zusätzlichen Raum

SSID & Kanäle

- SSID = der Netzname
- Benachbarte APs auf überlappungsfreie Kanäle legen
- Zu viele APs auf gleichem Kanal → gegenseitige Störung
- Sendeleistung sinnvoll begrenzen

WLAN-Sicherheit

WPA2

AES-Verschlüsselung, lange Standard – noch verbreitet.

WPA3

Aktuell, stärkerer Schutz (SAE) gegen Angriffe.

- WEP und WPA (TKIP) gelten als unsicher – nicht mehr nutzen
- Enterprise-Modus (802.1X) statt gemeinsamem Passwort

Roaming & Architektur

Standalone

Einzelne APs, manuell konfiguriert – kleine Umgebungen.

Controller / Cloud

Zentrale Verwaltung, nahtloses Roaming, Skalierung.

- Roaming: Client wechselt beim Gehen den AP
- Standards wie 802.11r beschleunigen den Wechsel

Ausleuchtung & Planung

Ein Site Survey vor dem Rollout spart später viel Frust – Kanäle und Sendeleistung planen.

- Überlappung der Zellen ~15–20 % für Roaming
- Kanalbreite bewusst wählen (mehr Breite = mehr Störung)
- Sendeleistung nicht maximal, sondern passend

Enterprise-WLAN

- 802.1X mit RADIUS statt gemeinsamem Passwort
- Mehrere SSIDs auf getrennte VLANs mappen
- Gäste-Netz isolieren (Client-Isolation, Captive Portal)
- Zentrale Verwaltung über Controller/Cloud

Das Wichtigste

- 802.11-Familie, Wi-Fi 5/6/6E heute relevant
- 2,4 GHz Reichweite vs. 5 GHz Tempo
- WPA3 (mind. WPA2), niemals WEP
- Ab mehreren APs: Controller/Cloud-Verwaltung

VPN-Technologien

Sichere Tunnel über unsichere Netze

Was ist ein VPN?

- Vertraulichkeit, Integrität und Authentizität
- Verbindet Standorte oder mobile Nutzer sicher
- Als wären entfernte Geräte im selben lokalen Netz

Ein Virtual Private Network baut einen verschlüsselten Tunnel über das Internet.

Zwei Einsatzformen

Site-to-Site

Verbindet ganze Standorte dauerhaft
(Router/Firewall zu Firewall).

Remote-Access

Einzelner Nutzer/Client verbindet sich ins Firmennetz.

- Site-to-Site: transparent für die Nutzer
- Remote-Access: Client-Software auf dem Endgerät

IPsec

- Arbeitet auf Layer 3, herstellerübergreifend
- IKE für Schlüsselaustausch, ESP für Verschlüsselung
- Robust, aber komplex in der Konfiguration

Der klassische Standard, besonders für Site-to-Site.

OpenVPN

- Nutzt TLS/SSL, flexibel bei Ports (auch 443/TCP)
- Kommt gut durch Firewalls und NAT
- Zertifikats- oder Passwort-basiert

Weit verbreitete Open-Source-Lösung.

WireGuard

Moderner, schlanker Ansatz – wenig Code, hohe Geschwindigkeit.

- Einfache Konfiguration über Schlüsselpaare
- Sehr performant, auch auf schwacher Hardware
- Zunehmend Standard für neue Deployments

Split- vs. Full-Tunnel

Split-Tunnel

Nur Firmenverkehr durch den Tunnel, Rest direkt ins Internet.

Full-Tunnel

Gesamter Verkehr läuft durchs Firmennetz.

- Split: performant, weniger Kontrolle
- Full: mehr Sicherheit/Filterung, mehr Last

Zertifikate & Schlüssel

- Client-Zertifikate über eine interne PKI
- Pre-Shared Key ist einfach, Zertifikate sind sicherer
- Perfect Forward Secrecy schützt alte Sitzungen
- Ablauf von Zertifikaten überwachen

VPN-Fehlersuche

Bei IPsec zuerst klären: scheitert es in Phase 1 (Aufbau) oder Phase 2 (Datenkanal)?

- MTU/Fragmentierung als häufige Ursache
- Zeit-Synchronisation und Zertifikatsablauf prüfen
- NAT-Traversal bei NAT auf dem Weg

Das Wichtigste

- VPN = verschlüsselter Tunnel übers Internet
- Site-to-Site vs. Remote-Access
- IPsec (klassisch), OpenVPN (flexibel), WireGuard (modern)
- Full-Tunnel für Kontrolle, Split für Tempo

Netzwerk-Diagnose & Tools

Systematische Fehlersuche mit den richtigen Werkzeugen

Systematisch vorgehen

Kabel/Link → IP-Konfiguration → Erreichbarkeit → Namensauflösung → Anwendung.

- Erst prüfen, was am wahrscheinlichsten ist
- Ein Schritt nach dem anderen, nicht raten
- Ergebnisse dokumentieren

Fehlersuche am besten von unten nach oben durch die Schichten.

IP-Konfiguration prüfen

`ipconfig /all` (Windows) · `ip a` (Linux)

- Hat das Gerät überhaupt eine gültige IP?
- 169.254.x.x = kein DHCP erreicht (APIPA)
- Gateway und DNS-Server korrekt?

Erreichbarkeit: ping

- Erst Gateway, dann externes Ziel (z. B. 1.1.1.1)
- Zeigt Paketverlust und Laufzeit
- Manche Geräte blocken ICMP – kein Ping ≠ immer offline

Prüft, ob ein Ziel antwortet (ICMP).

Pfad verfolgen: traceroute

`tracert` (Windows) · `traceroute` (Linux) zeigt jeden Hop.

- Findet heraus, wo der Weg abbricht
- Zeigt Latenz pro Zwischenstation
- Gut, um Provider- vs. interne Probleme zu trennen

Namensauflösung: nslookup/dig

- `nslookup name` / `dig name`
- Ping-IP klappt, Ping-Name nicht → DNS-Problem
- Anderen DNS-Server testen zum Eingrenzen

Prüft, ob DNS funktioniert.

Verbindungen & Nachbarn

- `netstat -ano` / `ss -tulpn` – offene Ports + Prozess
- `arp -a` – MAC-zu-IP-Zuordnung im lokalen Netz
- Hilft bei „Port belegt“ und Layer-2-Fragen

Tiefer analysieren

Wireshark

Paket-Mitschnitt – zeigt exakt, was über die Leitung geht.

nmap

Port-/Netzwerk-Scan – welche Hosts/Dienste sind erreichbar?

- Wireshark für schwierige Protokoll-Fehler
- nmap nur in eigenen/autorisierten Netzen

Checkliste je Schicht

- Schicht 1: Kabel, Link-LED, Port
- Schicht 2/3: `ipconfig/ip`, `arp`, richtiges VLAN
- Schicht 3: ping Gateway, dann extern
- Schicht 4–7: Port offen (`ss/netstat`), DNS (`nslookup`)

Wireshark-Basics

Capture-Filter begrenzen schon die Aufnahme, Display-Filter die Anzeige – nicht verwechseln.

- Nach IP oder Port filtern
- Den TCP-Handshake (SYN, SYN-ACK, ACK) erkennen
- Retransmissions deuten auf Verluste hin

Das Wichtigste

- Schichtweise eingrenzen statt raten
- `ipconfig/ip` → ping → `tracert` → `nslookup`
- `netstat/ss` + `arp` für Ports und Nachbarn
- Wireshark/nmap für die Tiefenanalyse

Netzwerk-Sicherheit

Angriffe auf Layer 2 erkennen und abwehren

Der oft vergessene Layer 2

Wer physischen oder WLAN-Zugang hat, kann Layer 2 angreifen.

- Switching-Ebene braucht eigene Schutzmaßnahmen
- Interner Angreifer ist ein reales Szenario
- Grundhärtung der Access-Switches

Viele Angriffe setzen im lokalen Netz an – nicht am Perimeter.

802.1X / NAC

- Port bleibt gesperrt bis zur erfolgreichen Anmeldung
- RADIUS-Server prüft die Identität
- NAC kann zusätzlich den Gerätezustand bewerten

Zugangskontrolle: nur authentifizierte Geräte kommen ins Netz.

DHCP-Snooping

Der Switch erlaubt DHCP-Antworten nur von vertrauenswürdigen Ports.

- Blockt Rogue-DHCP-Server (falsche Gateways/DNS)
- Baut eine Binding-Tabelle (IP ↔ MAC ↔ Port) auf
- Grundlage für weitere Schutzfunktionen

Dynamic ARP Inspection

- Verwirft gefälschte ARP-Antworten
- Wehrt ARP-Spoofing / Man-in-the-Middle ab
- Setzt aktiviertes DHCP-Snooping voraus

DAI prüft ARP-Pakete gegen die DHCP-Snooping-Tabelle.

Port-Security

- Schützt vor angeschlossenen Fremdgeräten/Hubs
- Verletzung → Port sperren oder alarmieren
- Sinnvoll an Endgeräte-Ports

Begrenzt, welche und wie viele MAC-Adressen pro Port erlaubt sind.

Typische Angriffe

VLAN-Hopping

Zugriff auf fremde VLANs – Gegenmittel: Native VLAN härten, kein Auto-Trunk.

ARP-Spoofing

Umleiten des Verkehrs – Gegenmittel: DAI + DHCP-Snooping.

- Unbenutzte Ports abschalten
- Management-VLAN vom Nutzerverkehr trennen

Weitere Layer-2-Angriffe

- MAC-Flooding füllt die MAC-Tabelle → Switch flutet
- DHCP-Starvation erschöpft den Adresspool
- STP-Manipulation, um Root-Bridge zu werden
- Gegenmittel: Port-Security, DHCP-Snooping, Root/BPDU Guard

Segmentierung als Sicherheitsstrategie

Segmentierung begrenzt den Schaden: Ein kompromittiertes Gerät darf nicht automatisch das ganze Netz gefährden.

- IoT- und Gäste-Geräte in eigene, isolierte VLANs
- Produktions- und Büronetz strikt trennen
- Zwischen Segmenten: Firewall-Regeln statt freier Durchgang

Netzwerk-Härtung

- Management-Zugänge in ein eigenes Netz
- Unbenutzte Ports und Dienste abschalten
- Konfigurations-Backups und Versionierung
- Änderungen über Change-Kontrolle

Das Wichtigste

- Layer 2 aktiv absichern, nicht nur den Perimeter
- 802.1X/NAC kontrolliert den Zugang
- DHCP-Snooping + DAI gegen Rogue-DHCP und ARP-Spoofing
- Port-Security + unbenutzte Ports deaktivieren

SD-WAN & moderne Netzarchitekturen

Software-definierte Netze, Zero Trust und SASE

Das klassische WAN

- Aufwändige Konfiguration pro Standort
- Wenig flexibel bei Cloud-Anbindung
- Ganzer Verkehr oft über die Zentrale (Umweg)

Standortvernetzung traditionell über teure, starre MPLS-Leitungen.

SD-WAN-Konzept

Software-Defined WAN trennt Steuerung (zentral) von der Weiterleitung (Geräte).

- Zentrales Management aller Standorte
- Nutzt beliebige Leitungen (Internet, LTE, MPLS) als Overlay
- Wählt automatisch den besten Pfad je Anwendung

Vorteile von SD-WAN

- Günstiger durch normale Internetleitungen
- Direkter, schneller Weg in die Cloud (Local Breakout)
- Zentrale Richtlinien statt Gerät-für-Gerät
- Schnelle Inbetriebnahme neuer Standorte

Zero-Trust-Networking

„Never trust, always verify“ – kein Gerät ist allein durch seinen Standort vertrauenswürdig.

- Jeder Zugriff wird geprüft (Identität + Kontext)
- Mikrosegmentierung statt eines großen flachen Netzes
- Least Privilege als Grundprinzip

SASE

- Kombiniert SD-WAN mit Sicherheitsfunktionen (SWG, CASB, ZTNA, FWaaS)
- Schutz nah am Nutzer, egal wo er arbeitet
- Passt zu Cloud- und Homeoffice-Realität

Secure Access Service Edge – Netzwerk und Sicherheit als Cloud-Dienst.

Netzwerkvirtualisierung

- Overlay-Netze (z. B. VXLAN) über physische Infrastruktur
- Beispiel VMware NSX: Mikrosegmentierung im Rechenzentrum
- Automatisierbar wie Software (Infrastructure as Code)

Netze werden in Software abgebildet – unabhängig von der Hardware.

SD-WAN-Bausteine

- Underlay = physische Leitungen, Overlay = logische Tunnel
- Zentraler Controller orchestriert die Standorte
- Application-aware Routing wählt den besten Pfad je App
- Local Breakout für direkten Cloud-Zugang

Migration & Betrieb

SD-WAN führt man meist schrittweise ein – hybrid aus vorhandenem MPLS und Internet.

- Zero-Touch-Provisioning für neue Standorte
- Zentrale Sichtbarkeit und Monitoring
- Sicherheit gleich mitdenken (Richtung SASE)

Das Wichtigste

- SD-WAN: zentrale Steuerung, flexible Leitungen, Cloud-freundlich
- Zero Trust: prüfen statt vertrauen, segmentieren
- SASE bündelt Netz + Security aus der Cloud
- Virtualisierung macht Netze software-steuerbar



LB-02 · Betriebssysteme

12 Kapitel

OS-Geschichte & Hersteller

Von DOS und Unix bis Windows 11, Linux und macOS

Warum Geschichte kennen?

- Kommandozeile, Dateisysteme, Benutzerrechte – alt, aber lebendig
- Wer die Herkunft kennt, versteht Design-Entscheidungen
- Prüfungsrelevant: Generationen und Marktanteile

Viele Konzepte von heute stammen aus den 70ern und 80ern.

Die DOS-Ära

Kein Multitasking, kein Speicherschutz – aber die Basis für den PC-Siegeszug.

- Kommandozeile `C:\>` als einzige Oberfläche
- 640 KB RAM-Grenze prägte eine Generation
- Lebt bis heute in der `cmd.exe` weiter

MS-DOS/PC-DOS ab 1981 – reine Textkonsole, ein Nutzer, eine Aufgabe.

Windows-Generationen

9x-Linie

Windows 1.0 → 3.11 → 95/98/ME – grafisch, aber auf DOS aufgesetzt, instabil.

NT-Linie

Windows NT → 2000 → XP → 7 → 10 → 11 – eigener Kernel, Speicherschutz, Mehrbenutzer.

- Seit XP basieren alle Consumer-Versionen auf NT
- Windows 11: nur noch 64-Bit, TPM 2.0 Pflicht

Unix & die Geburt von Linux

1991 veröffentlichte Linus Torvalds den ersten Linux-Kernel – frei und quelloffen.

- POSIX-Standard sorgt für Unix-Kompatibilität
- GNU-Werkzeuge + Linux-Kernel = vollständiges freies OS
- BSD als zweiter großer Unix-Zweig

Unix (1969, Bell Labs) prägte fast alles, was danach kam.

Linux-Distributionen

Debian-Zweig

Debian, Ubuntu, Mint – apt , weit verbreitet auf Servern.

RHEL-Zweig

RHEL, Rocky, Alma, Fedora – dnf , im Enterprise-Umfeld verbreitet.

- Dritter großer Zweig: SUSE/openSUSE (zypper)
- Distribution wählt Init-System, Paketformat, Support-Zyklus

Ein Kernel, viele Distributionen – gebündelt mit Paketverwaltung und Tools.

macOS

- Wurzeln in NeXTSTEP (1989), gekauft mit Rückkehr von Steve Jobs
- Unterbau ist BSD-Unix + Mach-Kernel (Darwin)
- Seit 2020 auf eigener ARM-Architektur (Apple Silicon M-Chips)

Apples Desktop-OS mit Unix-Kern.

Marktanteile heute

Desktop

Windows dominiert (~70 %), macOS ~15 %, Linux wächst im einstelligen Bereich.

Server & Cloud

Linux dominiert klar – Basis fast aller Webserver, Container und Cloud-VMs.

- Mobil: Android (Linux-Kernel) vor iOS
- Für Admins gilt: beide Welten beherrschen

Mobile & eingebettete Systeme

- Android basiert auf einem Linux-Kernel
- iOS stammt aus der Unix-Familie (wie macOS)
- Eingebettete Geräte nutzen oft ein Echtzeit-OS (RTOS)
- Mobil dominieren Android und iOS

Lizenzmodelle

Proprietär

Windows, macOS – geschlossen, lizenzpflichtig.

Open Source

Linux & Co. – quelloffen; Copyleft (GPL) vs. permissiv.

- Open Source heißt nicht automatisch kostenlos im Betrieb
- Support und Wartung oft als Subscription

Das Wichtigste

Windows-NT-Linie und Unix/Linux sind die zwei prägenden Stammbäume – macOS gehört zur Unix-Familie.

- DOS lebt in der CMD weiter
- Linux = ein Kernel, viele Distributionen
- Server-Welt ist überwiegend Linux

OS-Kernkomponenten & Architektur

Kernel, Prozesse, Scheduling und Speicherverwaltung

Was ein OS leistet

- Abstraktion: Programme müssen die Hardware nicht kennen
- Ressourcenverwaltung: CPU, RAM, Datenträger, Geräte
- Schutz & Isolation zwischen Prozessen und Nutzern

Das Betriebssystem vermittelt zwischen Hardware und Anwendungen.

Kernel vs. Userspace

Der Kernel läuft privilegiert (Ring 0), Anwendungen laufen isoliert im Userspace (Ring 3).

- Übergang über System Calls (kontrollierte Schnittstelle)
- Absturz einer App reißt nicht das System mit
- Treiber laufen meist im Kernel – daher kritisch für Stabilität

Kernel-Typen

Monolithisch

Alles im Kernel (Linux). Schnell, aber Treiberfehler treffen den Kern.

Mikrokern

Nur das Nötigste im Kern, Dienste im Userspace. Robust, mehr Overhead.

- Hybrid: Mischform (Windows NT, macOS/XNU)
- Trade-off zwischen Geschwindigkeit und Isolation

Prozesse & Threads

Prozess

Eigener Speicherraum, isoliert. Eigene Ressourcen.

Thread

Läuft innerhalb eines Prozesses, teilt sich dessen Speicher.

- Mehrere Threads ermöglichen Parallelität
- Kontextwechsel kostet CPU-Zeit

Scheduling

Bei einer CPU läuft nie wirklich alles gleichzeitig – der Scheduler wechselt nur sehr schnell.

- Zeitscheiben (Time Slices) und Prioritäten
- Preemptives Multitasking: OS kann Prozesse unterbrechen
- Ziel: Fairness + Reaktionsfähigkeit

Der Scheduler verteilt CPU-Zeit auf die lauffähigen Prozesse.

Speicherverwaltung

- Virtueller Speicher entkoppelt Programm von physischem RAM
- Paging: Speicher in Seiten (Pages) verwaltet
- Auslagern auf Datenträger: Swap (Linux) / Auslagerungsdatei (Windows)

Jeder Prozess sieht seinen eigenen virtuellen Adressraum.

Dateisysteme

Windows

NTFS (Rechte, Journaling), ReFS für große Volumes.

Linux / macOS

ext4, XFS, Btrfs (Linux) · APFS (macOS).

- Journaling schützt vor Inkonsistenzen bei Abstürzen
- Berechtigungen sind Teil des Dateisystems
- Praxis: Linux liest NTFS meist problemlos (Treiber ntfs-3g) – umgekehrt braucht Windows für ext4 Zusatzsoftware

Ein-/Ausgabe & Treiber

- Der Kernel spricht die Hardware über Treiber an
- Interrupts melden Ereignisse an die CPU
- Fehlerhafte Treiber sind eine häufige Absturzursache
- Unter Linux erscheinen Geräte als Dateien unter /dev

Der Bootvorgang

Firmware → Bootloader → Kernel → Init-System – diese Kette solltest du kennen.

- UEFI/BIOS startet den Bootloader
- Der Bootloader lädt den Kernel
- systemd bzw. Init startet die Dienste

Das Wichtigste

Kernel verwaltet Hardware und schützt Prozesse; Scheduling und virtueller Speicher schaffen die Illusion von Gleichzeitigkeit und unbegrenztem RAM.

- Kernel privilegiert, Apps isoliert
- Prozess = Isolation, Thread = Parallelität
- Virtueller Speicher + Paging als Fundament

Windows Desktop – Grundlagen & Administration

Editionen, Registry, Dienste, Ereignisprotokoll & Benutzer

Editionen

Pro / Enterprise

Domänenfähig, per Gruppenrichtlinie verwaltbar – fürs Unternehmen.

Home

Nicht domänenfähig – eher privat.

- Windows 10/11 im Unternehmen
- Windows 11: TPM 2.0 und Secure Boot vorausgesetzt

Die Registry

Die Registrierungsdatenbank speichert fast alle Einstellungen – ein falscher Eingriff kann das System lahmlegen.

- Bearbeiten mit regedit
- Vor Änderungen den Zweig exportieren
- Nur anfassen, wenn es keinen anderen Weg gibt

Dienste

- Hintergrundprogramme, oft schon vor der Anmeldung
- Starttyp: automatisch, manuell, deaktiviert
- Über die Dienste-Verwaltung steuern
- Abhängigkeiten beachten

Ereignisprotokoll

Das Ereignisprotokoll ist das Tagebuch von Windows – oft der schnellste Weg zur Ursache.

- Fehler, Warnungen, Anmeldungen
- Nach Quelle und Zeit filtern
- Ereignis-IDs gezielt nachschlagen

Task-Manager

- Schneller Blick auf CPU-, Speicher-, Datenträgerlast
- Prozesse und Autostart verwalten
- Ressourcenfresser identifizieren

Benutzerverwaltung

- Lokale Konten und Gruppen
- Im Unternehmen meist zentral über Active Directory
- Kennwörter zurücksetzen, Gruppen zuordnen

Least Privilege

Die wirksamste einzelne Schutzmaßnahme: nicht dauerhaft als Administrator arbeiten.

- Als normaler Benutzer anmelden
- Rechte nur bei Bedarf anheben

Autostart im Griff

- Task-Manager, Reiter Autostart, zeigt und deaktiviert Einträge
- Sysinternals Autoruns findet auch versteckte Starts
- Weniger Autostart = schnellerer, stabilerer Rechner

Zu viele Autostart-Programme bremsen den Start und fressen Ressourcen.

Wiederherstellung & abgesicherter Modus

Vor größeren Änderungen einen Wiederherstellungspunkt setzen – das ist der schnelle Rückweg.

Abgesicherter Modus

Startet mit minimalen Treibern – ideal zur Fehlersuche.

Systemwiederherstellung

Setzt Systemdateien auf einen früheren Punkt zurück.

Das Wichtigste

- Pro/Enterprise fürs Unternehmen
- Registry, Dienste, Ereignisprotokoll, Task-Manager
- AD für zentrale Benutzerverwaltung
- Immer mit minimalen Rechten arbeiten

Windows Server – Grundlagen

Editionen, Rollen, Installationsvarianten und Verwaltung

Windows Server im Überblick

- Basis für Active Directory, DNS, DHCP, Dateidienste
- Optimiert auf Stabilität und Langzeitbetrieb
- Aktuelle Version: Windows Server 2022/2025

Das Server-Betriebssystem von Microsoft für zentrale IT-Dienste.

Editionen

Standard

Für physische Server / wenige VMs. Lizenz für 2 VMs inklusive.

Datacenter

Für hochvirtualisierte Umgebungen – unbegrenzte VMs, Storage Spaces Direct.

- Lizenzierung nach physischen CPU-Kernen (Core-Lizenzen)
- Zusätzlich CALs (Client Access Licenses) für Zugriffe

Rollen & Features

Ein Server sollte möglichst wenige Rollen tragen – kleinere Angriffsfläche, klarere Zuständigkeit.

- Typische Rollen: AD DS, DNS, DHCP, Datei-/Druckdienste, IIS, Hyper-V
- Features ergänzen (z. B. .NET, Failover-Clustering)
- Installation über Server-Manager oder PowerShell

Funktionen werden als Rollen installiert – der Server macht nur, was er soll.

Installationsvarianten

Desktop Experience

Mit grafischer Oberfläche – einfacher, aber mehr Angriffsfläche und Updates.

Server Core

Ohne GUI, nur Konsole/Remote – schlanker, sicherer, weniger Reboots.

- Server Core empfohlen für Infrastruktur-Rollen
- Verwaltung dann per PowerShell / Windows Admin Center

Verwaltung

- Server-Manager: lokale und Remote-Server im Blick
- Windows Admin Center: modernes Web-Interface
- PowerShell + Remoting: skalierbare Automatisierung
- Gruppenrichtlinien (GPO) für Konfiguration im Verbund

Release-Modelle

LTSC

Long-Term Servicing Channel – 5+5 Jahre Support, stabil (2019/2022/2025).

Hotpatch / Azure Edition

Weniger Reboots, engere Cloud-Integration.

- Für klassische Server: LTSC ist der Standard

Best Practices

Least Privilege, Updates, Backup und Monitoring sind Pflicht – kein Server ohne Wiederherstellungsplan.

- Dedizierte Admin-Konten, kein Alltag als Domänen-Admin
- Regelmäßige Updates in Wartungsfenstern
- Getestete Backups und dokumentierte Konfiguration

Lizenzierung vertieft

- Core-Lizenzen: mindestens 16 Kerne pro Server
- Zusätzlich CALs: pro Benutzer oder pro Gerät
- Datacenter für unbegrenzte VMs
- Software Assurance für Upgrades/Rechte

Device-CAL vs. User-CAL – ein Beispiel

Device-CAL

Lohnt sich bei Schichtbetrieb: ein Kiosk-PC im Lager, mehrere Mitarbeiter pro Tag – eine Lizenz statt vielen.

User-CAL

Lohnt sich bei Homeoffice: ein Mitarbeiter mit Laptop, Tablet und Diensthandy – eine Lizenz statt pro Gerät.

- Mischbetrieb ist erlaubt: User-CAL fürs Außendienstteam, Device-CAL für gemeinsam genutzte PCs
- RDS-CAL ist eine **separate** Lizenz zusätzlich zur normalen CAL (siehe LB-04 · RDS)

SQL Server: Per-Core vs. Server+CAL

Faustregel: Server+CAL ist meist nur bis rund 30–130 Nutzer günstiger (je nach Szenario) – darüber gewinnt Per-Core.

Per-Core

Lizenz nach CPU-Kernen, unbegrenzt viele Nutzer/Geräte – planbar bei großer oder schwankender Nutzerzahl.

Server + CAL

Server-Lizenz + CAL pro Nutzer/Gerät – nur bei Standard Edition verfügbar.

Rollen im Zusammenspiel

AD, DNS und DHCP bilden zusammen das Fundament einer Windows-Umgebung.

- Datei- und Druckdienste bauen darauf auf
- Jede Rolle erhöht die Angriffsfläche
- Rollen auf mehrere Server verteilen

Das Wichtigste

- Standard vs. Datacenter je nach Virtualisierungsgrad
- Nur benötigte Rollen installieren
- Server Core + Remote-Verwaltung als sichere Basis
- LTSC für planbaren Langzeitbetrieb

DOS, CMD & klassische Befehle

Die Kommandozeile für schnelle Diagnose und Wartung

Von DOS zur cmd.exe

- Auch heute schnellster Weg für viele Checks
- Läuft ohne GUI – ideal auf Server Core
- Grundlage für Batch-Automatisierung (.bat/.cmd)

Die Eingabeaufforderung ist der direkte Nachfahre von MS-DOS.

Navigation

`dir` listet Verzeichnisse, `cd` wechselt, `cd ..` geht eine Ebene hoch.

- `dir /a` zeigt auch versteckte Dateien
- `tree` zeigt die Ordnerstruktur
- `cls` leert den Bildschirm

Dateien verwalten

- `copy` / `move` / `del` / `ren` – Basics
- `xcopy` für Ordnerbäume
- `robocopy` – robustes Kopieren mit Wiederaufnahme und Spiegelung (`/MIR`)
- `mkdir` / `rmdir` für Verzeichnisse

Netzwerk-Diagnose

Der Klassiker bei Netzwerkproblemen: `ipconfig /all` → `ping` → `tracert` → `nslookup`.

- `ipconfig /release` und `/renew` für DHCP
- `netstat -ano` zeigt Verbindungen + PID
- `arp -a` für die ARP-Tabelle

System & Dienste

- `tasklist` / `taskkill /PID` – Prozesse
- `sc query` / `sc start|stop` – Dienste
- `net user` / `net use` – Benutzer & Laufwerke
- `systeminfo` für Systemüberblick

Wartung & Reparatur

- `sfc /scannow` – Systemdateien prüfen/reparieren
- `DISM /Online /Cleanup-Image /RestoreHealth` – Abbild reparieren
- `chkdsk /f` – Dateisystem prüfen
- `gpupdate /force` – Gruppenrichtlinien neu anwenden

Batch-Grundlagen

`@echo off` unterdrückt die Ausgabe der Befehle selbst.

- Variablen mit `%name%`, Argumente mit `%1`
- Steuerung mit `if`, `for`, `goto`
- Für komplexere Aufgaben besser PowerShell nutzen

Befehle lassen sich in `.bat/.cmd`-Dateien bündeln.

Umgebungsvariablen & Pfade

- `set` zeigt und setzt Variablen
- Nutzung mit Prozentzeichen: `%PATH%`
- `PATH` bestimmt, wo Befehle gesucht werden
- `where befehl` findet den Speicherort

Pipes & Umleitung

Mit `|` verkettest du Befehle, mit `>` und `>>` schreibst du in Dateien.

- `befehl > datei.txt` überschreibt
- `befehl >> datei.txt` hängt an
- Mit `findstr` in der Ausgabe suchen

Das Wichtigste

CMD ist nicht tot – für schnelle Diagnose und einfache Skripte oft der kürzeste Weg.

- Netzwerk-Kette: `ipconfig` → `ping` → `tracert` → `nslookup`
- `robocopy` statt `copy` für ernsthafte Kopierjobs
- `sfc/DISM/chkdsk` als Reparatur-Trio

PowerShell – Grundlagen & Praxis

Objektorientierte Automatisierung für Windows und mehr

Was ist PowerShell?

Der entscheidende Unterschied zur CMD: durch die Pipeline fließen echte Objekte, keine Textzeilen.

- Plattformübergreifend (PowerShell 7 auch auf Linux/macOS)
- Tief in Windows und Microsoft-Produkte integriert
- Standardwerkzeug für Admin-Automatisierung

Eine Shell und Skriptsprache, die mit Objekten statt reinem Text arbeitet.

Cmdlet-Schema

Verben

Get, Set, New, Remove, Start, Stop – vorhersehbar.

Beispiele

Get-Service, Get-Process, New-Item, Set-Content.

- `Get-Command` findet Cmdlets
- `Get-Help` liefert Hilfe und Beispiele

Befehle folgen dem Muster `Verb-Substantiv`.

Die Pipeline

```
Get-Service | Where-Object Status -eq 'Running' | Sort-Object Name
```

- Objekte werden an das nächste Cmdlet weitergereicht
- Eigenschaften bleiben erhalten (kein Text-Parsing)
- `Get-Member` zeigt Eigenschaften und Methoden

Variablen & Operatoren

- Variablen beginnen mit `$`: `$name = 'Server01'`
- Vergleich: `-eq -ne -gt -lt -like -match`
- Logik: `-and -or -not`
- Arrays: `$liste = 1,2,3`

Filtern & Auswählen

- `Where-Object` filtert Objekte nach Bedingung
- `ForEach-Object` verarbeitet jedes Objekt
- `Select-Object` wählt Eigenschaften / erste N
- `Sort-Object`, `Group-Object`, `Measure-Object`

Fehlerbehandlung

```
try { ... } catch { ... } finally { ... }
```

- `-ErrorAction Stop` macht Fehler abfangbar
- `$Error` enthält die letzten Fehler
- Ausgaben mit `Write-Verbose` / `Write-Warning`

Robuste Skripte fangen Fehler kontrolliert ab.

Wichtige Bausteine

- Module: `ActiveDirectory`, `ImportExcel`, `PSWindowsUpdate`
- Remoting über WinRM: `Enter-PSSession`, `Invoke-Command`
- Ausführungsrichtlinie: `Get-/Set-ExecutionPolicy`
- Skripte als `.ps1`, Funktionen für Wiederverwendung

Praxisbeispiel

```
Get-Service | Where-Object {$_.StartType -eq 'Automatic' -and $_.Status -eq 'Stopped'} | Start-Service
```

- Erst mit `-WhatIf` testen, dann ausführen
- So werden aus 20 Klicks eine Zeile

Alle gestoppten automatischen Dienste finden und neu starten:

Ausgabe formatieren

- `Format-Table` und `Format-List` für die Anzeige
- `Select-Object` wählt Eigenschaften
- `Sort-Object` / `Group-Object` zum Ordnen
- `Out-File` oder `Export-Csv` zum Speichern

Skripte strukturieren

- Sprechende Namen und Kommentare
- Parameter typisiert im `param()`-Block
- Funktionen für Wiederverwendung
- Häufiges in eigene Module auslagern

Das Wichtigste

PowerShell denkt in Objekten – das macht Filtern, Sortieren und Weiterverarbeiten zuverlässig und lesbar.

- Verb-Substantiv + Pipeline als Kern
- Get-Help und Get-Member sind deine Freunde
- -WhatIf schützt vor Unfällen

Linux-Administration

Dateisystem, Rechte, Pakete, systemd und Automatisierung

Dateisystem-Hierarchie

- `/etc` Konfiguration · `/var` variable Daten & Logs
- `/home` Nutzerdaten · `/usr` Programme
- `/proc` und `/sys` : virtuelle Kernel-Schnittstellen

Linux ordnet alles unter einem einzigen Wurzelverzeichnis `/` an (FHS).

Berechtigungen

Jede Datei hat Rechte für **user**, **group**, **others** – jeweils **r**(ead), **w**(rite), **x**(ecute).

- `chmod 750 datei` setzt Rechte numerisch
- `chown user:group datei` ändert Eigentümer
- `sudo` für administrative Aktionen

Paketverwaltung

Debian/Ubuntu

`apt update`, `apt install`, `apt upgrade` ·
Basis `dpkg`.

RHEL/Rocky/Fedora

`dnf install`, `dnf update` · Basis `rpm`.

- Repositories liefern signierte Pakete
- Abhängigkeiten werden automatisch aufgelöst

systemd

`systemctl status|start|stop|enable|restart dienst`

- Dienste werden als Units beschrieben
- `enable` = Autostart beim Booten
- `systemctl list-units` zeigt laufende Units

Das moderne Init- und Dienstverwaltungssystem der meisten Distributionen.

Logs analysieren

- `journalctl -u dienst` – Logs eines Dienstes
- `journalctl -f` – live mitlesen
- `journalctl -p err -b` – Fehler des aktuellen Boots
- Klassische Logs weiterhin unter `/var/log`

Automatisierung mit cron

```
0 2 * * * /pfad/backup.sh – täglich um 02:00 Uhr.
```

- `crontab -e` bearbeitet die eigene Crontab
- Fünf Felder: Minute Stunde Tag Monat Wochentag
- Alternative: systemd-Timer

Wiederkehrende Aufgaben zeitgesteuert ausführen.

Netzwerk & SSH

- `ip a` zeigt Adressen, `ip r` die Routen
- `ss -tulpn` zeigt offene Ports + Prozesse
- `ssh user@host` – Fernzugriff
- Schlüsselbasierte Anmeldung statt Passwort bevorzugen

Prozesse & Ressourcen

- `ps` und `top/htop` zeigen Prozesse
- `kill` beendet über die PID
- `df` und `du` für Speicherplatz
- `free` für den Arbeitsspeicher

Benutzer & Rechte vertieft

- Benutzer in `/etc/passwd`, Gruppen in `/etc/group`
- Administrative Rechte über `sudo` (sudoers)
- `umask` steuert Standard-Rechte neuer Dateien
- SUID-Bit mit Bedacht einsetzen

Das Wichtigste

Rechte, Paketmanager und systemd sind das Fundament der Linux-Administration – dazu Logs mit `journalctl` und Automatisierung per cron.

- `apt` (Debian) vs. `dnf` (RHEL) je nach Distribution
- `systemctl` steuert Dienste, `journalctl` liest Logs
- SSH mit Schlüsseln absichern

Software-Deployment & Paketsysteme

Software zuverlässig verteilen – Windows und Linux

Warum standardisieren?

- Gleiche Version auf allen Geräten
- Nachvollziehbar, wiederholbar, automatisierbar
- Schnelles Ausrollen und Zurückrollen

Manuelle Installation skaliert nicht und ist fehleranfällig.

Windows-Paketformate

MSI

Klassisches Installer-Format, unterstützt Silent-Install und Rollback.

MSIX

Modernes, sandboxed Paketformat – saubere Deinstallation.

- Silent-Install z. B. `msiexec /i app.msi /qn`
- Auch klassische EXE-Setups mit Silent-Schaltern

Winget & Chocolatey

winget

Microsofts eingebauter Paketmanager für Windows.

Chocolatey

Community-Paketquelle, in Unternehmen etabliert.

- `winget install <id>` installiert per Kommandozeile
- Ideal für Skripte und Erstinstallation neuer Geräte

Intune-Deployment

Geräte müssen nicht im lokalen Netz sein – Verteilung über das Internet. Voraussetzung: Enrollment, passende Lizenz (z. B. Business Premium/EMS) und laufende Internetverbindung – kein Automatismus für jedes Unternehmen.

- Apps, Konfigurationen und Richtlinien zentral
- Win32-Apps als `.intunewin` paketieren
- Zuweisung an Benutzer- oder Gerätegruppen

Cloud-basiertes Ausrollen über Microsoft Intune (MDM).

Linux-Paketverteilung

Klassisch

apt / dnf aus Repositories – integriert und schlank.

Universell

Snap, Flatpak, AppImage – distributionsunabhängig, mehr Speicher.

- Repos für System- und Serverpakete
- Flatpak/Snap eher für Desktop-Anwendungen

Enterprise-Werkzeuge

- MECM/SCCM: Verteilung im klassischen On-Prem-Umfeld
- Intune: Cloud-first, moderne Verwaltung
- Kombination (Co-Management) ist verbreitet
- In KMU/MSP-Praxis oft pragmatischer: RMM-Tools (NinjaOne, ManageEngine u. a., siehe LB-09) – Deployment, Patching und Fernzugriff aus einer Konsole, ohne volles Intune-/Entra-Setup
- Ansible/Skripte für serverseitiges Deployment

Best Practices

Erst in einem Testring ausrollen, dann breit – und immer einen Rückweg planen.

- Pilotgruppe vor flächigem Rollout
- Versionen und Silent-Schalter dokumentieren
- Deinstallation/Rollback vorab testen

Paketquellen & Signaturen

Pakete kommen aus signierten Quellen – die Signatur schützt vor Manipulation.

- Repositories als vertrauenswürdige Quelle
- GPG-Signaturen prüfen die Echtheit
- Interne Spiegel/Repos für Unternehmen

Rollout-Strategien

- In Ringen ausrollen: Pilot → breit → alle
- Wartungsfenster einplanen
- Deinstallation/Rollback vorher testen
- Installierte Software inventarisieren

Das Wichtigste

- MSI/MSIX + winget/Chocolatey für Windows
- Intune für Cloud-Verteilung
- Repos vs. universelle Pakete unter Linux
- Testring + Rollback als Sicherheitsnetz

Update- & Patch-Management

Systeme sicher und stabil aktuell halten

Warum patchen?

Die meisten erfolgreichen Angriffe nutzen längst bekannte, aber ungepatchte Lücken.

- Sicherheit: bekannte Schwachstellen schließen
- Stabilität: Fehler beheben
- Compliance: Nachweis regelmäßiger Wartung

Ungepatchte Systeme sind das häufigste Einfallstor für Angriffe.

Windows Update & WSUS

Ehrlich betrachtet: WSUS setzt ständige LAN-Erreichbarkeit voraus. Bei verteilten Standorten und viel Homeoffice ohne VPN ist das in der Praxis kaum noch zeitgemäß.

Windows Update

Direkt von Microsoft – einfach, wenig Kontrolle.

WSUS

Lokaler Update-Server – von Microsoft aber seit 2024 im Erhaltungsmodus (keine neuen Funktionen mehr).

- Läuft technisch weiter, wird aber nicht mehr weiterentwickelt
- Nur für Microsoft-Produkte – keine Drittanbieter-Software

Moderne Verwaltung

- Intune / Windows Update for Business: cloudbasierte Update-Ringe
- MECM für detaillierte On-Prem-Steuerung – aber Setup-/Lizenzaufwand für kleine Umgebungen oft zu hoch
- Automatische Genehmigungen mit Verzögerung (Deferral)
- In der Praxis übernehmen bei KMU/MSP oft RMM-Systeme (NinjaOne, ManageEngine u. a., siehe LB-09) das Patchen – ein Werkzeug für Windows, macOS, Linux und Drittanbieter-Software zugleich

Der Patch-Zyklus

„Patch Tuesday“ – der zweite Dienstag im Monat.

- Out-of-Band-Updates bei kritischen Lücken
- Test-Ringe: Pilot → Breit → Alle
- Wartungsfenster für Server einplanen

Microsoft veröffentlicht Sicherheitsupdates gebündelt.

Linux-Updates

- `apt upgrade` / `dnf update` regelmäßig einplanen
- `unattended-upgrades` (Debian) für Sicherheitsupdates
- `dnf-automatic` (RHEL) als Pendant
- Kernel-Updates brauchen meist einen Reboot (Livepatch als Option)

Rollback & Wartungsfenster

Kein Patch ohne Rückweg – Snapshot oder Backup vor größeren Updates.

- VM-Snapshot vor kritischen Updates
- Wartungsfenster + Kommunikation an Nutzer
- Fehlerhafte Updates gezielt deinstallieren

Priorisierung

- Kritische, aktiv ausgenutzte Lücken zuerst
- Exponierte Systeme (Internet-facing) bevorzugt
- Abwägung: Risiko der Lücke vs. Risiko des Updates

Nicht jeder Patch ist gleich dringend.

Test & Freigabe

- Updates zuerst in einem Testring prüfen
- Known Issues der Hersteller lesen
- Kompatibilität mit Anwendungen sicherstellen
- Gestaffelt freigeben, nicht alles auf einmal

Compliance & Reporting

Nachweisbarer Patch-Stand ist oft eine Compliance-Anforderung.

- Patch-Stand regelmäßig messen
- Ausnahmen dokumentieren und begründen
- Berichte für Audits bereithalten

Das Wichtigste

- Ungepatcht = Hauptrisiko
- WSUS ist ein Auslaufmodell – Intune/Windows Update for Business oder RMM sind der praktische Standard
- Test vor breitem Rollout, Rollback bereithalten
- Kritische & exponierte Systeme priorisieren

CVE, Zero-Day & Schwachstellen-Management

Schwachstellen erkennen, bewerten und priorisieren

Was ist eine Schwachstelle?

- Software-Bug, Fehlkonfiguration oder Design-Fehler
- Ausnutzung durch einen Exploit
- Ziel des Managements: schneller schließen als Angreifer nutzen

Ein Fehler oder eine Schwäche, die Angreifer ausnutzen können.

CVE

Format: CVE-2024-12345 (Präfix – Jahr – laufende Nummer).

- Vergeben von MITRE und CNAs (CVE Numbering Authorities)
- Eine ID pro Schwachstelle – herstellerübergreifend eindeutig
- Grundlage für Advisories und Scanner

Common Vulnerabilities and Exposures – der weltweite Katalog eindeutiger Kennungen.

CVSS-Score

Niedrig / Mittel

0,1–3,9 / 4,0–6,9 – geringeres Risiko.

Hoch / Kritisch

7,0–8,9 / 9,0–10,0 – dringend handeln.

- Base-, Temporal- und Environmental-Metriken
- Berücksichtigt Angriffsweg, Komplexität, Auswirkung
- Hilft bei der Priorisierung – ersetzt aber keinen Kontext

Common Vulnerability Scoring System – bewertet den Schweregrad von 0 bis 10.

Wichtige Quellen

- NVD (National Vulnerability Database): CVEs + CVSS
- Exploit-DB: bekannte Exploit-Codes
- Hersteller-Advisories (Microsoft, Cisco, ...)
- CISA KEV: Katalog aktiv ausgenutzter Lücken

Zero-Day

Eine Lücke, für die es noch keinen Patch gibt – der Hersteller hatte „null Tage“ Zeit.

- Besonders gefährlich, da keine Standardabwehr
- Zwischenlösung: Workarounds, Härtung, Monitoring
- Wird oft teuer gehandelt und gezielt eingesetzt

Der Management-Prozess

Scannen → Priorisieren → Beheben → Verifizieren → wiederholen.

- Regelmäßige Scans (z. B. Nessus, OpenVAS, Qualys)
- Priorisieren nach Score + Exposition + Ausnutzung
- Nach dem Patch erneut prüfen (Verifizierung)

Schwachstellen-Management ist ein Kreislauf, kein Einmal-Projekt.

Priorisierung in der Praxis

- Aktiv ausgenutzt (KEV) → sofort
- Kritisch + Internet-facing → sehr hoch
- Hoch, aber intern/isoliert → geplant
- Niedrig → im normalen Zyklus

Den CVSS-Vektor lesen

- Angriffsvektor: Netzwerk, angrenzend, lokal, physisch
- Komplexität und benötigte Rechte
- Nutzer-Interaktion erforderlich?
- Environmental-Score gewichtet für die eigene Umgebung

Vom Advisory zur Aktion

Zwei Fragen entscheiden über die Dringlichkeit: Bin ich betroffen? Ist das System erreichbar?

- Betroffene Versionen abgleichen
- Workaround oder Patch einspielen
- Nachverfolgen bis zur Verifizierung

Das Wichtigste

CVE benennt, CVSS bewertet, der Prozess priorisiert – entscheidend ist, ob eine Lücke aktiv ausgenutzt und erreichbar ist.

- CVE = eindeutige ID, CVSS = Schweregrad 0–10
- Zero-Day = Lücke ohne Patch
- Management ist ein Kreislauf mit Verifizierung

macOS-Administration

Das dritte große Betriebssystem im Unternehmen – Unix-Basis, MDM und Praxis

Wo begegnet einem macOS im Unternehmen?

- Kreativabteilungen (Design, Video, Audio)
- Software-Entwicklung
- Unternehmen, die bewusst auf Apple setzen

Neben Windows und Linux das dritte relevante Betriebssystem im IT-Alltag.

Apple Silicon

Der Wandel zu eigenen M-Chips (M1, M2 und Nachfolger) brachte konkurrenzlose Effizienz pro Watt.

- Lange Akkulaufzeit bei hoher Leistung
- Kaum Lüftergeräusch, auch unter Last
- ARM-Architektur statt klassischem x86

Unix-Basis & Terminal

- Standard-Shell: `zsh`
- Viele Befehle identisch zu Linux: `ls`, `grep`, `ssh`
- Kein Paketmanager ab Werk – `Homebrew` füllt die Lücke

macOS läuft auf einer Unix-Basis – vertraut für alle mit Linux-Erfahrung.

Geräteverwaltung im Unternehmen

Verwaltung läuft über **MDM** (Mobile Device Management) – etwa Jamf oder Microsoft Intune.

- Verteilung von Konfigurationsprofilen: WLAN, VPN, Sicherheitsrichtlinien
- Zentrale Softwareverteilung ohne Nutzereingriff
- Volle Kontrolle (Supervised Mode) braucht Apple Business Manager + automatische Geräteregistrierung – bei privat/gebraucht gekauften Geräten ohne ABM-Zuordnung geht das nicht
- Dieser Registrierungs-Zwang ist ein Grund, warum Jamf oder RMM-Tools mit Mac-Support neben Intune verbreitet sind

Active Directory und klassische GPOs greifen bei macOS nicht direkt.

Netzwerkeinbindung

- Standard-WLAN und VLAN-getaggte Netze funktionieren ganz normal
- Dateifreigaben mit Windows-Servern über SMB
- Eigene Mechanismen: AirDrop (Dateiaustausch), Time Machine (Backup)

Backup mit Time Machine

Einzelplatz

Externe Platte oder Netzwerkspeicher einrichten – stündliches, automatisches Backup im Hintergrund.

Unternehmen

Ersetzt keine zentrale Backup-Strategie mit Versionierung und Monitoring.

Die eingebaute Backup-Lösung von macOS.

Praxis-Relevanz für Azubis

- Sobald Entwickler, Design oder Geschäftsführung im Haus sind, taucht macOS im Support auf
- Unix-Grundlagen aus der Linux-Administration helfen direkt weiter
- MDM-Kenntnisse (z. B. Jamf) sind bei IT-Admins zunehmend gefragt

Das Wichtigste

macOS bringt Unix-Komfort mit Apple-Hardware zusammen, wird aber nicht über Windows-Bordmittel, sondern über MDM verwaltet.

- Netzwerkeinbindung: Standardtechnik (WLAN, VLAN, SMB)
- Verwaltung: MDM (Jamf, Intune) statt AD/GPO – volle Kontrolle nur mit Apple Business Manager
- Backup & Dateiaustausch: Time Machine, AirDrop

Dateisysteme

NTFS, ext4, FAT32 und exFAT – was du im Alltag wirklich wissen musst

Die Ausgangssituation

Lesen geht fast immer. Schreiben ist komplizierter – und genau da entstehen die häufigsten Support-Fälle.

Klassischer Fall: Ein Kunde bringt eine externe Festplatte mit Windows-Daten. Sie wird an einen Linux-Server angeschlossen.

NTFS unter Linux

- Linux kann NTFS ohne Zusatzpakete **lesen**
- Für zuverlässiges **Schreiben** wird `ntfs-3g` benötigt
- Fehlt das Paket, wundert man sich, warum Dateien nicht geschrieben werden können

NTFS versus ext4

NTFS

Windows-Standard. Komplexe Zugriffsrechte über ACLs, Dateiverschlüsselung direkt mit EFS. Journaling gegen Datenkorruption bei Absturz.

ext4

Linux-Standard. Klassisches Unix-Rechtesystem (Benutzer, Gruppe, andere). Ebenfalls Journaling. Beide erlauben Dateien bis 16 TB.

Merksatz

Für Windows-Server ist NTFS gesetzt. Für Linux-Server ist ext4 der Standard.

Wechselmedien: FAT32

- Kompatibel mit fast allem: Windows, Mac, Linux, Drucker, Kameras
- Begrenzt auf maximal 4 GiB pro Datei – exakt 4.294.967.295 Byte
- Größere Datei kopieren → kryptische Fehlermeldung

Der Nachfolger: exFAT

- Keine praktische Dateigrößen-Grenze
- Wird von allen modernen Betriebssystemen unterstützt
- Erste Wahl für USB-Sticks und externe Platten mit großen Dateien

Löst die Größenbegrenzung von FAT32 auf, bei weiterhin guter Kompatibilität.

Das Wichtigste

Windows-Systeme → NTFS. Linux → ext4. Maximale Kompatibilität auf Wechselmedien → exFAT.

Und: Wer FAT32 auf einem USB-Stick hat und eine große Filmdatei kopieren will – formatiert neu.



LB-03 · Hardware

7 Kapitel

CPU – Geschichte, Architektur & Markt

x86 und ARM, Kerne, Takt, Cache, TDP & Sockel

Zwei große Architekturen

x86

Intel und AMD – klassisch auf Desktop und Server.

ARM

Energieeffizient – Apple M-Chips, zunehmend im Server.

- x86 dominiert historisch den PC/Server-Markt
- ARM wächst durch Effizienz

Hersteller & Reihen

- Intel: Core (Desktop), Xeon (Server)
- AMD: Ryzen (Desktop), EPYC (Server)
- ARM: u. a. Apple Silicon (M-Reihe)

Wichtige Kennzahlen

- Kerne: parallele Aufgaben
- Takt (GHz): Geschwindigkeit je Kern
- Cache: schneller Zwischenspeicher
- TDP: zu kühlende Verlustleistung

Threads

Per Hyper-Threading/SMT bearbeitet ein Kern zwei Threads – 8 Kerne zeigen dann 16 logische Prozessoren.

- Mehr Threads helfen bei paralleler Last
- Nicht dasselbe wie echte Kerne

Herstellertricks: Binning & Kernsegmentierung

Beim Binning werden Chips nach Qualität sortiert: ein Die mit defekten oder schwächeren Kernen wird per eFuse dauerhaft beschnitten und als günstigeres Modell verkauft – z. B. wird aus einem 8-Kern-Die mit zwei fehlerhaften Kernen ein 6-Kern-i5 statt eines i7.

- Intel-K-Modelle sind die besten Exemplare eines Wafers – deshalb übertaktbar
- Intels P-Core/E-Core-Hybrid-Design ist zusätzlich eine Marketing-Differenzierung nach Kernanzahl
- AMD verzichtet bei Ryzen bewusst darauf – Kernzahl allein sagt nichts über echte Mehrleistung

Nicht jede Modellreihe ist ein eigenes Design – oft steckt derselbe Wafer dahinter.

Sockel & Kompatibilität

- Die CPU muss zum Sockel des Mainboards passen
- Ryzen passt nicht in einen Intel-Sockel
- Chipsatz bestimmt Funktionsumfang
- Neue CPUs brauchen oft ein UEFI-Update

Auswahl für Server

Für Virtualisierung und Datenbanken zählen viele Kerne oft mehr als hoher Takt.

- Kernzahl an die Last anpassen
- ECC-RAM-Unterstützung beachten

Cache-Stufen

- L1 winzig und extrem schnell, pro Kern
- L2 größer, meist pro Kern
- L3 groß und von allen Kernen geteilt
- Mehr Cache hilft besonders bei Datenbanken und Servern

Schneller Zwischenspeicher direkt in der CPU – überbrückt den langsamen Arbeitsspeicher.

Stromverbrauch & Kühlung

Im Server zählt Leistung pro Watt: effiziente CPUs senken Strom- und Klimakosten spürbar.

TDP

Verlustleistung in Watt – Richtwert für die nötige Kühlung.

Kühlung

Luft oder Wasser – muss zur TDP und zum Sockel passen.

Das Wichtigste

- x86 (Intel/AMD) und ARM als große Architekturen
- Kerne, Takt, Cache, TDP als Kennzahlen
- CPU muss zum Sockel passen
- Server: viele Kerne für Virtualisierung/DB

RAM – Typen, Specs & ECC

Arbeitsspeicher richtig verstehen und auswählen

Was RAM leistet

- Flüchtig: Inhalt geht beim Ausschalten verloren
- Viel schneller als SSD/HDD
- Zu wenig RAM → Auslagern → System wird langsam

Arbeitsspeicher hält Daten und Programme bereit, mit denen die CPU gerade arbeitet.

DDR4 vs. DDR5

DDR4

Etabliert, günstig, weit verbreitet.

DDR5

Höhere Bandbreite und Kapazität, neuere Plattformen.

- Generationen sind nicht mischbar (andere Slots)
- Board und CPU müssen den Typ unterstützen

Wichtige Kennzahlen

- Kapazität (GB) – wichtigster Faktor
- Takt (MHz/MT/s) – Geschwindigkeit
- Timings (CL) – Latenz
- Kanäle: Dual-/Quad-Channel steigert Bandbreite

ECC-RAM

ECC (Error-Correcting Code) erkennt und korrigiert einzelne Bitfehler – Pflicht für Server.

- Verhindert stille Datenverfälschung
- Braucht Unterstützung von CPU und Board
- Im Desktop meist nicht vorhanden

Weitere Formen

LPDDR

Stromsparend, für Notebooks/Mobilgeräte – oft fest verlötet.

Registered/RDIMM

Für Server mit sehr viel RAM, stabiler bei hoher Bestückung.

- DIMM (Desktop/Server) vs. SO-DIMM (Notebook)

Praxis-Tipps

- Immer im passenden Kanal-Layout bestücken (Handbuch!)
- Module möglichst identisch (Kit kaufen)
- Server: ECC nicht weglassen
- Nach Aufrüstung Speichertest laufen lassen

RAM richtig bestücken

- Steckplatz-Layout laut Handbuch (Dual-/Quad-Channel)
- Möglichst als Kit kaufen (gleicher Typ/Timing)
- Module verschiedener Hersteller nur ungern mischen
- Takt per XMP/EXPO im UEFI aktivieren

Fehler erkennen

Zufällige Abstürze und Bluescreens ohne Muster deuten oft auf RAM-Fehler hin.

- Mit Memtest86 gründlich testen
- Bei Servern: ECC-Fehler stehen im BMC-Log
- Einzelne Riegel isoliert testen

Das Wichtigste

- Kapazität vor Takt für die meisten Aufgaben
- DDR4/DDR5 nicht mischbar
- ECC ist Server-Pflicht
- Dual-Channel für mehr Bandbreite nutzen

Storage – HDD, SSD, NVMe

Datenträgertechnik von der Festplatte bis NVMe

Die Speicherhierarchie

- HDD: mechanisch, günstig pro GB
- SSD: elektronisch, schnell, robust
- NVMe: SSD über PCIe – am schnellsten

Massenspeicher ist langsamer als RAM, aber dauerhaft.

HDD

Stärken

Viel Kapazität für wenig Geld – Archiv/Backup.

Schwächen

Langsam, empfindlich, mechanischer Verschleiß.

- Drehzahl (5400/7200 rpm) beeinflusst Tempo
- NAS-Platten sind auf Dauerbetrieb ausgelegt

SSD & NVMe

NVMe-SSDs nutzen PCIe statt SATA und sind um ein Vielfaches schneller.

- SATA-SSD: schnell, überall einsetzbar
- NVMe (M.2/U.2): höchste Geschwindigkeit
- Kein Vorteil durch Defragmentieren (nicht nötig)

Haltbarkeit & Leistung

Ehrlicher Vergleich: 1 TB TLC hält oft ca. 600 TBW aus, eine vergleichbare QLC-SSD nur ca. 200–400 TBW. Für Datenbanken/VDI gehört TLC hin, QLC eignet sich nur für kapazitätsstarke, lesestarke Ablagen.

- TBW: geschriebene Terabyte über die Lebensdauer
- DWPD (Drive Writes Per Day): wie oft die volle Kapazität täglich über die Garantiezeit überschrieben werden darf
- P/E-Zyklen: Schreib-/Löschvorgänge pro Zelle
- IOPS: Ein-/Ausgabeoperationen pro Sekunde

Auswahl nach Einsatz

Consumer

Desktop/Notebook, moderate Schreiblast.

Enterprise/NAS

Dauerbetrieb, hohe TBW, oft mit Power-Loss-Protection.

- Datenbanken/Virtualisierung profitieren stark von NVMe

Praxis

- SSD/NVMe fürs System, HDD fürs Archiv
- SMART-Werte im Blick behalten
- RAID ersetzt kein Backup (siehe LB-06)
- Firmware aktuell halten

Anbindung & Formfaktor

M.2 / U.2

NVMe direkt am PCIe-Bus – höchste Geschwindigkeit.

SATA

2,5" oder 3,5", einfach und überall einsetzbar.

- PCIe-Lanes des Boards beachten
- Enterprise-Laufwerke mit Power-Loss-Protection

SMART & Gesundheit

- SMART-Werte regelmäßig auslesen
- Wear-Leveling verteilt Schreibzugriffe
- Ersatzsektoren/Reserve als Frühwarnung
- Auffällige Datenträger rechtzeitig tauschen

Das Wichtigste

- HDD = billig & groß, SSD = schnell, NVMe = am schnellsten
- TBW/IOPS/P-E-Zyklen als Kennzahlen
- Enterprise-SSD für Dauerlast
- System auf SSD, Archiv auf HDD

Mainboard & Formfaktoren

Die Platine, die alles verbindet

Rolle des Mainboards

- Chipsatz bestimmt Funktionsumfang
- Sockel muss zur CPU passen
- Bestimmt Aufrüstbarkeit

Verbindet CPU, RAM, Storage und Erweiterungskarten.

Formfaktoren

ATX / microATX

Standard bzw. kompakter – Desktops/Workstations.

Mini-ITX

Sehr klein – Mini-PCs, Homelabs.

- Größe bestimmt Slots und Anschlüsse
- Server-Boards nach eigenen Standards

PCIe

PCI Express verbindet Grafikkarten, NVMe und Netzwerkkarten – neuere Generationen sind schneller.

- Lanes (x1, x4, x8, x16) = Bandbreite
- Abwärtskompatibel zwischen Generationen
- Aufteilung der Lanes im Handbuch prüfen

BIOS vs. UEFI

BIOS

Alt, Textoberfläche, MBR-Partitionierung.

UEFI

Modern, GPT, große Datenträger, grafisch.

- UEFI ist heute Standard
- Firmware-Updates schließen Sicherheitslücken

Secure Boot

Secure Boot lässt nur signierte Bootloader zu und erschwert Boot-Malware.

- Für Windows 11 vorausgesetzt (mit TPM 2.0)
- Bei manchen Linux-Setups Anpassung nötig
- Teil der UEFI-Konfiguration

Praxis

- Board nach CPU-Sockel und Chipsatz wählen
- Genug PCIe/M.2-Slots für die Zukunft
- Firmware vor Inbetriebnahme aktualisieren
- Kompatibilitätslisten (QVL) beachten

Anschlüsse & Header

- SATA- und M.2-Slots für Datenträger
- USB-, Front-Panel- und Lüfter-Header
- PWM-Lüftersteuerung für leisen Betrieb
- Genug Anschlüsse für die Zukunft einplanen

Aufrüsten & Kompatibilität

Neue CPU-Generationen brauchen oft erst ein UEFI-Update auf dem Board.

- Kompatibilitätsliste (QVL) prüfen
- PCIe-Lanes werden teils geteilt (Slot vs. M.2)
- BIOS/UEFI vor CPU-Wechsel aktualisieren

Das Wichtigste

- Sockel + Chipsatz bestimmen die CPU-Wahl
- Formfaktor legt Slots/Größe fest
- UEFI + GPT statt BIOS + MBR
- Secure Boot + TPM für moderne Sicherheit

Server-Hardware

Was Server von Desktops unterscheidet

Server ≠ großer PC

- 24/7-Betrieb über Jahre
- Redundante Komponenten
- Zentrale Verwaltung vieler Systeme

Server sind auf Dauerbetrieb, Redundanz und Fernwartung ausgelegt.

Bauformen

Rack (1U/2U)

Standard im Rechenzentrum, im 19-Zoll-Schrank.

Tower / Blade

Tower für kleine Umgebungen; Blade für hohe Dichte.

- Höheneinheit U = 44,45 mm
- Rack spart Platz und erleichtert Verkabelung

Redundanz

Kritische Komponenten sind doppelt vorhanden – fällt eine aus, läuft der Server weiter.

- Redundante Netzteile (Hot-Swap)
- RAID für Datenträger
- Mehrere Netzwerkpfade

Out-of-Band-Management

iLO / iDRAC

HPE bzw. Dell – eigene Management-Controller.

IPMI

Herstellerübergreifender Standard.

- Ein-/Ausschalten, Konsole, Sensoren aus der Ferne
- Eigener Netzwerkport, streng absichern

Fernwartung unabhängig vom Betriebssystem.

Marktführer bei Server-Hardware

Dell & HPE

Klassische Marktführer im Channel/SMB-Geschäft.

Supermicro

Durch den GPU-/KI-Server-Boom stark gewachsen.

- Lenovo (breit aufgestellt), Cisco UCS, Fujitsu (stark in Europa) als weitere Größen
- Hyperscaler (große Cloud-Anbieter) kaufen oft direkt bei ODMs statt bei Markenherstellern – für KMU/Channel bleiben aber Dell/HPE/Lenovo/Supermicro die relevante Auswahl

Wartung im Betrieb

- Hot-Swap für Platten und Netzteile
- Werkzeuglose Gehäuse
- Sensoren für Temperatur/Lüfter
- Ersatzteile bevorraten

Praxis

- Management-Netz vom Produktivnetz trennen
- Firmware/BMC aktuell halten und absichern
- Redundanz regelmäßig testen
- Auf ausreichende Kühlung achten

Kühlung & Stromversorgung

- Klare Trennung von Kalt- und Warmgang
- Redundante, Hot-Swap-fähige Netzteile
- USV gegen Stromausfälle
- PDU im Rack mit Monitoring

Bauform wählen

Die Bauform folgt Platz, Dichte und Budget – nicht dem Prospekt.

- Rack für Rechenzentren (Standard)
- Tower für kleine Standorte ohne Rack
- Blade für hohe Dichte bei vielen Servern

Das Wichtigste

- Server = Dauerbetrieb + Redundanz + Fernwartung
- Rack/Tower/Blade je nach Umgebung
- iLO/iDRAC/IPMI für Out-of-Band
- Management-Zugang streng absichern

Netzwerkkarten & Verkabelung

Vom NIC bis zum Patchpanel

Netzwerkkarten (NICs)

- Geschwindigkeiten: 1G, 10G, 25G, 40G und mehr
- Server oft mit mehreren Ports
- Teaming/Bonding für Redundanz & Bandbreite

Die Schnittstelle zwischen Gerät und Netz.

Kupfer vs. Glasfaser

Kupfer (RJ45)

Günstig, einfach, begrenzte Distanz.

Glasfaser (SFP/SFP+)

Große Distanzen, hohe Geschwindigkeit, störungsfrei.

- SFP-Module je nach Reichweite/Typ
- Glasfaser im RZ und für Backbone

Kabelkategorien

Cat5e reicht für 1G; für 10G eher Cat6/Cat6a; Cat7/Cat8 für höchste Ansprüche.

- Höhere Kategorie = mehr Bandbreite/Schirmung
- Länge begrenzt (Kupfer meist 100 m)
- Qualität und Verlegung beeinflussen die Leistung

Strukturierte Verkabelung

- Patchpanel als zentraler Rangierpunkt
- Kurze Patchkabel zum Switch
- Beschriftung ist Pflicht
- Saubere Kabelführung im Rack

Häufige Fehler

- Zu lange oder minderwertige Kabel
- Knicke und enge Radien
- Fehlende Dokumentation
- Duplex-/Speed-Mismatch

Praxis

- Kabel testen (Zertifizierung) nach Verlegung
- Reserve-Adern/Ports einplanen
- Glasfaser sauber halten (Verschmutzung!)
- Konsistente Beschriftung im gesamten Rack

Glasfaser-Typen

Single-Mode

Große Distanzen (Backbone, Standortverbindung).

Multi-Mode

Kürzere Strecken im Gebäude/Rechenzentrum.

- Passende SFP-Module zum Fasertyp
- Stecker sauber halten (LC/SC)

Dokumentation & Prüfung

- Jede Leitung eindeutig beschriften
- Nach dem Verlegen zertifizieren/messen
- Sauberes Kabelmanagement im Rack
- Reserve-Adern und -Ports einplanen

Das Wichtigste

- NIC-Geschwindigkeit passend zur Last wählen
- Kupfer kurz/günstig, Glasfaser weit/schnell
- Kabelkategorie bestimmt die Bandbreite
- Struktur + Beschriftung sparen später Zeit

Hardware-Diagnose & Lifecycle

Fehler finden und Hardware planvoll ablösen

POST & Boot

- Piepcodes/LEDs zeigen Fehler
- Kein Bild? → CPU/RAM/GPU prüfen
- BMC-Logs bei Servern auswerten

Der Power-On Self-Test prüft die Hardware beim Start.

Diagnose-Werkzeuge

Memtest86

Testet RAM auf Fehler.

CrystalDiskInfo / smartctl

Liest SMART-Werte von Datenträgern.

- Herstellerdiagnose (Dell/HPE/Lenovo)
- Live-Systeme für Tests nutzen

Typische Symptome

- Zufällige Abstürze → RAM/Netzteil/Temperatur
- Langsame Zugriffe → Datenträger (SMART)
- Bluescreens → Treiber/Hardware
- Thermal Throttling → Kühlung/Staub

Monitoring

Sensoren früh überwachen – warme/lärmende Systeme kündigen oft Ausfälle an.

- Temperatur, Lüfter, Spannungen
- SMART-Trends statt Momentaufnahme
- Alarme bei Grenzwerten (siehe LB-09)

Lifecycle-Management

- EOL/EOSL: Ende von Verkauf und Support
- Refresh-Zyklen (z. B. Server ~5 Jahre)
- Garantie/Support-Verträge im Blick behalten

Hardware hat eine geplante Lebensdauer.

Außerbetriebnahme

- Datenträger sicher löschen oder vernichten
- Assets dokumentieren und abmelden
- Umweltgerecht entsorgen/recyclen
- Ersatz rechtzeitig beschaffen

Vorbeugende Wartung

- Staub entfernen, Luftwege frei halten
- Wärmeleitpaste bei älteren Systemen erneuern
- Firmware/BIOS aktuell halten
- Kritische Ersatzteile bevorraten

Fehler eingrenzen

Systematisch tauschen: mit minimaler Bestückung booten, dann Komponenten einzeln ergänzen.

- Logs (BMC/Ereignisanzeige) zuerst lesen
- Eine Variable nach der anderen ändern
- Verdächtige Komponente isolieren

Das Wichtigste

- POST/LEDs und Logs als erste Anlaufstelle
- Memtest & SMART für RAM/Storage
- Sensoren proaktiv überwachen
- Lifecycle & sichere Entsorgung einplanen



LB-04 · Server & Dienste

9 Kapitel

Active Directory – Grundlagen

Domäne, Forest, Trust, DC, Kerberos & LDAP

Was ist Active Directory?

- Verwaltet Benutzer, Computer und Gruppen
- Zentrale Anmeldung im ganzen Netz
- Alles an einer Stelle statt an jedem Rechner

Der zentrale Verzeichnisdienst von Microsoft.

Grundbegriffe

- Domäne: die Verwaltungseinheit
- Forest: mehrere Domänen
- Trust: Vertrauensstellung zwischen Domänen
- DC: Domänencontroller stellt AD bereit

Kerberos & LDAP

Kerberos

Anmeldung per Ticket – Passwort wird nicht ständig gesendet.

LDAP

Protokoll für Abfragen des Verzeichnisses.

- Zeit-Synchronisation ist für Kerberos kritisch

Objekte & OUs

- Objekte: Benutzer, Computer, Gruppen
- Organisationseinheiten (OUs) strukturieren
- Nach Verwaltung/Standort gliedern
- Basis für Rechte-Delegation und Gruppenrichtlinien

AD braucht DNS

Ohne funktionierendes DNS findet ein Client keinen Domänencontroller – dann geht fast nichts.

- AD ist eng mit DNS verzahnt
- Bei AD-Problemen zuerst DNS prüfen

FSMO-Rollen

Forest-weit

Schema-Master und Domänennamen-Master – je einmal im Forest.

Domänen-weit

RID-Master, PDC-Emulator und Infrastruktur-Master – je Domäne.

- PDC-Emulator ist Zeitquelle und wichtig bei Passwort-Änderungen
- Bei DC-Ausfall lassen sich Rollen übertragen (transfer/seize)

Fünf Sonderaufgaben, die jeweils nur ein DC übernimmt (Single Master).

Gruppenrichtlinien (GPO)

Ehrlicher Haken: GPOs wirken nur bei regelmäßigem Kontakt zum Domänencontroller. Homeoffice/Außenstandorte ohne VPN oder Entra-Hybrid-Join bleiben dadurch oft tagelang auf altem Stand – genau diese Lücke füllen in der Praxis häufig RMM-Systeme oder Cloud-Policy-Tools (siehe LB-09).

- Verknüpft an Standort, Domäne oder OU
- Steuern Sicherheit, Software, Laufwerke, Desktop
- Wirken von oben nach unten – untere gewinnen bei Konflikt
- gpupdate erzwingt die Aktualisierung am Client

Zentrale Regeln für Benutzer und Computer im gesamten Verzeichnis.

Standorte, Replikation & Global Catalog

Faustregel: mindestens zwei Domänencontroller je Domäne – fällt einer aus, läuft die Anmeldung weiter.

- Mehrere DCs gleichen ihre Daten per Replikation ab
- Sites bilden physische Standorte ab und steuern den Replikationsverkehr
- Der Global Catalog beantwortet Forest-weite Suchen

Das Wichtigste

- Domäne, Forest, Trust, DC als Grundbegriffe
- Anmeldung über Kerberos, Abfragen über LDAP
- Objekte in OUs strukturieren
- Ohne DNS kein Active Directory

Active Directory – Administration

OUs, Gruppenrichtlinien und FSMO im Griff

OU-Design

- Nach Verwaltung/Standort gliedern, nicht nach Optik
- Delegation von Rechten pro OU möglich
- Basis für gezielte GPO-Zuweisung

Organisationseinheiten strukturieren Objekte und tragen Richtlinien.

Gruppenrichtlinien (GPO)

GPOs verteilen Einstellungen zentral an Benutzer und Computer.

- Verknüpfung an Standort, Domäne oder OU
- Reihenfolge: Local → Site → Domain → OU
- `gpupdate /force` erzwingt Aktualisierung

FSMO-Rollen

- Schema-, Domänennamen-Master (Forest-weit)
- PDC-Emulator, RID-, Infrastruktur-Master (je Domäne)
- Ausfall planvoll übertragen (seize/transfer)

Fünf Sonderrollen, die nicht jeder DC gleichzeitig hat.

Replikation

- DCs gleichen Änderungen untereinander ab
- Standorte (Sites) steuern Replikationswege
- `repadmin` zur Diagnose
- Zeit-Synchronisation ist kritisch

Diagnose & Backup

dcdiag

Prüft die Gesundheit der Domänencontroller.

System State Backup

Sichert AD für die Wiederherstellung.

- Regelmäßig testen, nicht nur sichern
- Mind. zwei DCs für Redundanz

PowerShell-Automatisierung

- Massen-Anlage/-Änderung von Konten
- Berichte über Gruppen und Rechte
- Verwaiste Konten aufspüren
- Wiederkehrende Aufgaben skripten

Sichern & Wiederherstellen

- System-State-Backup enthält das AD
- Nicht-autoritativ: DC holt sich Daten von anderen
- Autoritativ: gezielt gelöschte Objekte zurückholen
- AD-Papierkorb (Recycle Bin) für einzelne Objekte

AD härten

Getrennte Admin-Ebenen (Tiering) verhindern, dass ein kompromittierter Arbeitsplatz zur Domänenübernahme führt.

- Keine Dauer-Nutzung von Domänen-Admin-Konten
- LAPS für einzigartige lokale Admin-Passwörter
- Privilegierte Konten besonders schützen

Das Wichtigste

- OUs sauber nach Verwaltung strukturieren
- GPO-Reihenfolge LSDOU merken
- FSMO kennen und Ausfälle planen
- dcdiag + getestetes AD-Backup

DNS – Deep Dive

Namensauflösung im Detail

Wozu DNS?

- Ohne DNS läuft im Unternehmen fast nichts
- AD ist stark von DNS abhängig
- Fehler äußern sich als „geht nicht“ überall

Übersetzt Namen in IP-Adressen – das Telefonbuch des Netzes.

Record-Typen

A / AAAA

Name → IPv4 bzw. IPv6.

CNAME / MX / PTR

Alias · Mailserver · Reverse-Lookup.

- SRV-Records für Dienste (z. B. AD)
- TXT für SPF/DKIM/DMARC

Zonen

Eine Zone ist ein verwalteter Namensbereich; der autoritative Server kennt die „Wahrheit“.

- Forward- vs. Reverse-Zone
- Primär/Sekundär für Redundanz
- AD-integrierte Zonen replizieren mit AD

Auflösung & Caching

- Rekursiv vs. iterativ
- Forwarder leiten unbekannte Anfragen weiter
- TTL steuert Caching-Dauer
- Cache leeren bei hartnäckigen Fehlern

Debugging

nslookup / dig sind die wichtigsten Werkzeuge.

- Gegen konkreten Server testen
- Reverse-Lookups prüfen
- Split-Brain-DNS: intern/extern andere Antworten

Sicherheit

- DNSSEC signiert Antworten (Manipulationsschutz)
- Zugriff auf Zonentransfers einschränken
- Interne Namen nicht nach außen leaken
- DNS-Logs für Auffälligkeiten nutzen

Weiterleitung & Cache

- Forwarder für externe Auflösung
- Conditional Forwarder für bestimmte Domänen
- Root Hints als Fallback
- Cache leeren bei hartnäckigen Fehlern

Wartung & Sicherheit

- Scavenging entfernt veraltete Einträge
- Zonentransfers einschränken
- DNSSEC signiert Antworten
- Interne Namen nicht nach außen leaken

Das Wichtigste

- DNS ist das Fundament – zuerst hier suchen
- Record-Typen und Zonen verstehen
- nslookup/dig zum Eingrenzen
- DNSSEC + eingeschränkte Transfers

DHCP – Deep Dive

Automatische IP-Vergabe zuverlässig betreiben

Wozu DHCP?

- Kein manuelles Konfigurieren jedes Geräts
- Vergibt IP, Gateway, DNS, Optionen
- Der DORA-Ablauf: Discover, Offer, Request, Ack

Vergibt IP-Adressen und Netzwerkparameter automatisch.

Scopes & Reservierungen

Scope

Adressbereich für ein Subnetz.

Reservierung

Feste IP für eine bestimmte MAC.

- Ausschlüsse für statisch vergebene IPs
- Lease-Dauer je nach Umgebung wählen

DHCP-Optionen

- Option 3: Gateway
- Option 6: DNS-Server
- Option 15: DNS-Suffix
- Weitere für VoIP, PXE-Boot etc.

Hochverfügbarkeit

DHCP-Failover verteilt die Last auf zwei Server und verhindert Ausfälle.

- Lastausgleich oder Hot-Standby
- Ohne DHCP bekommen neue Geräte keine IP
- Redundanz einplanen

Sicherheit

- Rogue-DHCP erkennen und verhindern
- DHCP-Snooping auf Switches (siehe LB-01)
- Nur autorisierte Server im AD
- Ungewöhnliche Vergaben überwachen

Troubleshooting

- 169.254.x.x → kein DHCP erreichbar (APIPA)
- Scope erschöpft? Bereich/Lease prüfen
- Relay (IP-Helper) über Subnetze hinweg
- Leases und Konflikte kontrollieren

Optionen & Klassen

- Vendor-/User-Classes für differenzierte Optionen
- PXE-Boot-Optionen für Betriebssystem-Verteilung
- Reservierungen sauber dokumentieren
- Optionen auf Server- vs. Scope-Ebene

Monitoring & Wartung

Ein voller Scope bedeutet: neue Geräte bekommen keine Adresse mehr.

- Scope-Auslastung überwachen
- Lease-Konflikte erkennen
- DHCP-Datenbank sichern

Das Wichtigste

- DORA-Ablauf und Optionen kennen
- Scopes + Reservierungen sauber planen
- Failover für Ausfallsicherheit
- APIPA-Adresse = DHCP-Problem

Fileserver & DFS

Dateidienste, Berechtigungen und verteilte Namespaces

Der Fileserver

- Freigaben (Shares) über SMB
- Zentrale Sicherung statt lokaler Inseln
- Grundlage für Zusammenarbeit

Zentrale Ablage für Dateien mit geregelter Zugriff.

NTFS- vs. Share-Rechte

Beim Zugriff gilt die restriktivere der beiden Berechtigungen.

- Share-Rechte gelten nur über das Netz
- NTFS-Rechte gelten immer
- Best Practice: Share großzügig, NTFS präzise

Rechte sauber vergeben

- Rechte über Gruppen, nicht über Einzelnutzer
- Vererbung bewusst nutzen
- Prinzip der minimalen Rechte
- Regelmäßig auf verwaiste Rechte prüfen

Quotas & FSRM

- Kontingente pro Ordner/Nutzer
- Dateiscreening (z. B. keine .exe)
- Berichte über Speicherbelegung

Speicherplatz steuern und Dateitypen filtern.

DFS

Namespace

Ein einheitlicher Pfad über mehrere Server hinweg.

Replikation

Hält Ordner zwischen Servern synchron.

- Nutzer sehen einen Pfad, egal wo die Daten liegen
- Standortübergreifende Ausfallsicherheit

Praxis

- Schattenkopien (VSS) für Vorversionen
- Zugriffe protokollieren (Auditing)
- Backup nicht vergessen (RAID ≠ Backup)
- Klare Ordnerstruktur vorgeben

Zugriffe auditieren

- Auditing protokolliert Zugriffe und Änderungen
- Access-Based Enumeration blendet Unsichtbares aus
- Regelmäßige Berechtigungsberichte
- Verwaiste Rechte aufräumen

Migration & DFS-R

Für Datei-Umzüge ist Robocopy mit Rechte-Übernahme das Werkzeug der Wahl.

- Robocopy überträgt Daten samt NTFS-Rechten
- DFS-Replikation hat Grenzen bei offenen Dateien
- Staging-Speicher ausreichend dimensionieren

Das Wichtigste

- Restriktivere von NTFS/Share gewinnt
- Rechte über Gruppen + Least Privilege
- DFS-Namespace vereinheitlicht Pfade
- Quotas/Screening + Backup

Printserver & Druckverwaltung

Drucken zentral bereitstellen

Warum ein Printserver?

- Ein Ort für Treiber und Konfiguration
- Warteschlangen zentral steuerbar
- Weniger Aufwand pro Client

Zentralisiert Warteschlangen, Treiber und Freigaben.

Treiber

Universelle/herstellerübergreifende Treiber reduzieren die Vielfalt.

- Treiber zentral bereitstellen
- Typ-4-Treiber für einfachere Verteilung
- Treiberkonflikte vermeiden

Verteilung per GPO

- Drucker automatisch zuweisen
- Nach Standort/Abteilung (OU)
- Standarddrucker setzen
- Selbstbedienung über Portal möglich

Protokolle

IPP

Internet Printing Protocol – modern, netzwerkfähig.

Direkt/TCP-Port

Direktes Drucken auf den Netzwerkdrucker.

- Warteschlange vs. Direktdruck abwägen

Troubleshooting

- Spooler-Dienst neu starten
- Warteschlange leeren
- Netzwerk/IP des Druckers prüfen
- Treiberprobleme isolieren

Trends

- Cloud-/Serverless-Printing im Kommen
- Sicheres Drucken (PIN am Gerät)
- Weniger On-Prem-Printserver
- Kosten über Print-Management steuern

Sicherheit & Treiber

Nach PrintNightmare gilt: Point-and-Print härten und Treiberinstallation einschränken.

- Treiberisolierung aktivieren
- Nur signierte Treiber zulassen
- Server zeitnah patchen

Moderne Alternativen

- Universal Print als Cloud-Dienst
- Direktdruck reduziert Serverabhängigkeit
- Sicheres Drucken (PIN/Badge am Gerät)
- Weniger On-Prem-Printserver nötig

Das Wichtigste

- Printserver zentralisiert Treiber & Queues
- Universaltreiber reduzieren Aufwand
- GPO verteilt Drucker automatisch
- Spooler/Queue als erste Fehlerquelle

Remote Desktop Services (RDS)

Zentrale Desktops und Anwendungen bereitstellen

Was ist RDS?

- Rechenleistung im Rechenzentrum
- Thin Clients oder beliebige Geräte
- Zentrale Wartung und Sicherheit

Nutzer arbeiten auf zentralen Servern statt auf lokalen Ressourcen.

Die Rollen

- RD Session Host: führt Sitzungen/Apps aus
- RD Gateway: sicherer Zugang von außen
- RD Web Access: Zugriff per Browser
- RD Connection Broker: verteilt Sitzungen

RemoteApp

Einzelne Anwendungen erscheinen wie lokal – ohne kompletten Remote-Desktop.

- Nur die App, nicht der ganze Desktop
- Gut für einzelne Fachanwendungen
- Nahtlose Integration ins lokale Startmenü

Lizenzierung (CALs)

Per User

Lizenz pro benanntem Benutzer.

Per Device

Lizenz pro Gerät.

- RD-Lizenzserver erforderlich
- Ohne CALs endet die Testphase

Zugang absichern

- RD Gateway + MFA statt RDP offen ins Internet
- Netzwerkebenen-Authentifizierung (NLA)
- Least Privilege auf dem Session Host
- Patches zeitnah einspielen

Praxis

- Ressourcen pro Nutzer dimensionieren
- Profile/Ordnerumleitung planen
- Alternativen: Azure Virtual Desktop
- Performance überwachen

Skalierung & Profile

- Mehrere Session Hosts als Farm
- Benutzerprofile mit FSLogix zentralisieren
- Ressourcen pro Nutzer dimensionieren
- Connection Broker verteilt die Last

Alternativen & Lizenz

Cloud

Azure Virtual Desktop, Windows 365 als moderne Optionen.

Lizenz

RD-Lizenzserver, User- oder Device-CAL.

- Cloud spart eigene Infrastruktur
- CAL-Modell zur Nutzung passend wählen

Das Wichtigste

- RDS zentralisiert Desktops/Apps
- Rollen: Session Host, Gateway, Web, Broker
- RemoteApp für einzelne Programme
- CALs + abgesicherter Zugang (Gateway/MFA)

IIS & Webserver-Grundlagen

Webinhalte und -anwendungen bereitstellen

Webserver im Überblick

IIS

Microsofts Webserver, tief in Windows integriert.

nginx / Apache

Verbreitet unter Linux; nginx auch als Reverse Proxy.

- Alle liefern Webinhalte per HTTP(S)
- Wahl nach Plattform und Anwendung

Sites & Bindings

Ein Binding verknüpft Site + IP + Port + Hostname (und Zertifikat bei HTTPS).

- Mehrere Sites pro Server möglich
- Host-Header trennen Websites
- Ports 80/443 als Standard

Application Pools (IIS)

- Isolieren Anwendungen voneinander
- Eigene Identität/Rechte pro Pool
- Absturz einer App reißt andere nicht mit
- Recycling gegen Speicherlecks

HTTPS & Zertifikate

- TLS-Zertifikat pro Site binden
- Weiterleitung HTTP → HTTPS
- Ablauf überwachen (siehe LB-07)
- Alte TLS-Versionen deaktivieren

Reverse Proxy & Rewrite

- URL-Rewrite für saubere/umgeleitete Adressen
- ARR / nginx als Reverse Proxy
- Lastverteilung auf mehrere Backends

Anfragen weiterleiten und URLs umschreiben.

Basis-Hardening

- Nicht benötigte Module deaktivieren
- Verzeichnis-Browsing aus
- Fehlermeldungen nicht zu gesprächig
- Security-Header setzen

Logs & Troubleshooting

- IIS-Logs zeigen jede Anfrage samt Status
- HTTP-Codes deuten: 404 (fehlt), 500 (Serverfehler)
- Failed Request Tracing für schwierige Fälle
- Application-Pool-Zustand prüfen

Performance & Skalierung

- Output-Caching und Kompression aktivieren
- Statische Inhalte effizient ausliefern
- Web-Farm hinter einem Load Balancer
- Ressourcen pro App-Pool begrenzen

Das Wichtigste

- IIS vs. nginx/Apache nach Plattform
- Bindings verknüpfen Site/IP/Port/Host
- App Pools isolieren Anwendungen
- HTTPS + Hardening als Pflicht

Datenbankserver

Grundlagen für IT-Admins

Rolle des DB-Servers

- Herzstück vieler Business-Anwendungen
- Leistung und Verfügbarkeit sind kritisch
- Backups sind unverzichtbar

Speichert und verwaltet strukturierte Daten für Anwendungen.

Die gängigen Systeme

SQL Server

Microsoft, tief in Windows/AD integriert.

MySQL/MariaDB · PostgreSQL

Weit verbreitet, oft unter Linux; PostgreSQL sehr funktionsreich.

- Editionen unterscheiden sich stark (Preis/Features)
- Wahl richtet sich nach der Anwendung

Admin-Grundlagen

- Instanzen, Datenbanken, Schemata
- Benutzer und Rechte (Least Privilege)
- Dienste und Ports im Blick
- Wartungspläne einrichten

Backup (Überblick)

DB-Backups laufen anders als Datei-Backups – Details in LB-06.

- Voll-, Differenz- und Transaktionslog-Sicherung
- Point-in-Time-Wiederherstellung
- Restore regelmäßig testen

Hochverfügbarkeit

- Replikation für Lesekopien/Ausfallschutz
- Failover-Cluster / Always On
- Wartungsfenster einplanen
- Monitoring von Sperren und Last

Performance-Grundlagen

- Indizes beschleunigen Abfragen
- Genug RAM für den Cache
- Schnelle Datenträger (NVMe)
- Langsame Abfragen identifizieren

Datenbank-Sicherheit

- Least Privilege für DB-Konten
- Verbindungen per TLS verschlüsseln
- Verschlüsselung im Ruhezustand (z. B. TDE)
- Zugriffe auditieren

Wartung & Performance

Indizes und aktuelle Statistiken sind der größte Hebel für schnelle Abfragen.

- Index-Pflege und Reorganisation
- Langsame Abfragen identifizieren
- Wartungspläne für Backup und Integrität

Das Wichtigste

- SQL Server / MySQL / PostgreSQL kennen
- Rechte sparsam, Wartung geplant
- DB-Backup + getesteter Restore
- HA + Monitoring für kritische DBs



LB-05 · Virtualisierung & Container

6 Kapitel

Virtualisierung – Grundlagen

Mehrere Systeme auf einer physischen Maschine

Was ist Virtualisierung?

- Bessere Auslastung teurer Hardware
- Isolierte, unabhängige VMs
- Schnelles Bereitstellen und Klonen

Ein Hypervisor teilt physische Hardware in mehrere virtuelle Maschinen auf.

Typ-1 vs. Typ-2

Typ-1 (Bare-Metal)

Läuft direkt auf der Hardware (ESXi, Hyper-V, Proxmox) – für Server.

Typ-2 (Hosted)

Läuft auf einem Betriebssystem (VirtualBox, Workstation) – für Desktops/Tests.

- Typ-1 ist effizienter und stabiler
- Typ-2 ist einfach für lokale Tests

Komponenten einer VM

- Virtuelle CPU, RAM, Disk, NIC
- Virtuelle Festplatten als Dateien
- Snapshots festhalten von Zuständen
- Gasterweiterungen für bessere Integration

Vorteile

Konsolidierung, Flexibilität und schnelle Wiederherstellung sind die großen Gewinne.

- Weniger physische Server
- Live-Migration ohne Ausfall
- Einfaches Backup ganzer VMs

Grenzen & Risiken

- Host-Ausfall trifft viele VMs (Redundanz!)
- Ressourcen-Überbuchung mit Bedacht
- Lizenzierung kann komplex sein
- Snapshots sind kein Backup

Ressourcen zuweisen

- Jeder VM weist du virtuelle CPUs (vCPU) und RAM zu
- Überbuchung (Overcommitment) ist möglich, aber mit Bedacht
- Bei Festplatten: Thin Provisioning (wächst mit) vs. Thick (fest reserviert)
- RAM-Ballooning gibt ungenutzten Speicher an andere VMs zurück

Die VMs teilen sich die physischen Ressourcen des Hosts.

Typische Einsatzfelder

Serverkonsolidierung

Viele Dienste laufen auf wenigen leistungsstarken Hosts.

Test & Entwicklung

Schnelle Wegwerf-Umgebungen, per Snapshot zurücksetzbar.

- Legacy-Systeme sicher gekapselt weiterbetreiben
- Desktop-Virtualisierung (VDI) für zentrale Arbeitsplätze

VMs sichern & klonen

Eine ganze VM ist letztlich eine Datei – das macht Sichern und Klonen einfach.

- Agentenloses Backup direkt über den Hypervisor
- Aus einer Vorlage (Template) neue VMs in Minuten ausrollen
- Aber: Ein Snapshot ist kein Backup (liegt auf demselben Storage)

Das Wichtigste

- Hypervisor teilt Hardware in VMs
- Typ-1 für Server, Typ-2 für Desktop/Tests
- Konsolidierung + Flexibilität
- Host-Redundanz einplanen

VMware & ESXi

Der etablierte Enterprise-Hypervisor

ESXi & vSphere

- ESXi läuft direkt auf dem Server
- vCenter verwaltet viele Hosts zentral
- Marktführer im Enterprise-Umfeld

ESXi ist der Typ-1-Hypervisor; vSphere ist die Produktsuite darum herum.

vCenter

vCenter bündelt Verwaltung, Rechte, Vorlagen und Automatisierung an einem Ort.

- Zentrale Konsole für alle Hosts
- Rollen- und Rechteverwaltung
- Vorlagen für schnelles Ausrollen

Kernfunktionen

vMotion

Live-Migration laufender VMs zwischen Hosts.

HA / DRS

Automatischer Neustart bei Ausfall · Lastverteilung.

- Cluster fassen Hosts zusammen
- Wartung ohne VM-Ausfall möglich

Storage & Netzwerk

- Datastores (VMFS/NFS) für VM-Dateien
- Shared Storage für vMotion/HA
- Virtuelle Switches (vSwitch/Distributed)
- Redundante Pfade (Multipathing)

Broadcom-Situation

- Umstellung auf Abo/Bundles
- Kosten für viele gestiegen
- Alternativen werden stärker geprüft (Proxmox, Hyper-V)

Nach der Übernahme durch Broadcom haben sich Lizenzmodelle geändert.

Vorlagen & Klone

- Templates als goldenes Abbild für neue VMs
- Full Clone = eigenständige Kopie; Linked Clone spart Speicher
- Anpassung per Customization Specification (Name, IP, Domäne)

Standardisierte VMs schnell und einheitlich ausrollen.

Storage vMotion

Storage vMotion verschiebt die VM-Dateien im laufenden Betrieb auf einen anderen Datastore.

- Wartung oder Umzug von Speicher ohne VM-Ausfall
- Gut kombinierbar mit Thin Provisioning
- Ergänzt das normale vMotion (das nur die Ausführung verschiebt)

Praxis & Wartung

- VMware Tools in jeder VM installieren (Treiber, Zeitsync, sauberes Herunterfahren)
- Vor Host-Wartung: Maintenance Mode (verschiebt VMs per DRS weg)
- Patchen zentral über den vSphere Lifecycle Manager
- Gesundheit und Warnungen im vCenter im Blick behalten

Das Wichtigste

- ESXi = Hypervisor, vCenter = zentrale Verwaltung
- vMotion/HA/DRS als Kernvorteile
- Shared Storage ermöglicht Live-Migration
- Lizenzlage seit Broadcom neu bewerten

Hyper-V

Microsofts Virtualisierung in Windows Server

Hyper-V im Überblick

- Als Server-Rolle aktivierbar
- Auch in Windows Pro enthalten (für Tests)
- Gut integriert in die Microsoft-Welt

Der in Windows Server integrierte Typ-1-Hypervisor.

Virtuelle Switches

Extern

VMs erreichen das physische Netz.

Intern / Privat

Nur Host ↔ VM bzw. nur VM ↔ VM.

- Switch-Typ je nach Zweck wählen
- VLAN-Tagging möglich

Gen 1 vs. Gen 2

Generation 2 nutzt UEFI, Secure Boot und PXE über Standard-Netzwerkkarten.

- Gen 2 für moderne 64-Bit-Gäste
- Gen 1 für ältere/kompatible Systeme
- Generation wird beim Erstellen festgelegt

Live Migration & Replica

Live Migration

VM ohne Ausfall auf anderen Host verschieben.

Hyper-V Replica

VM asynchron an einen Zweitstandort replizieren.

- Failover-Cluster für Hochverfügbarkeit
- Replica als einfache DR-Option

Verwaltung

- Hyper-V-Manager für einzelne Hosts
- SCVMM für große Umgebungen
- PowerShell-Modul für Automatisierung
- Windows Admin Center als Web-Oberfläche

Checkpoints

Standard

Erfasst auch den Arbeitsspeicher – exakt, aber nicht app-konsistent.

Production

Nutzt VSS in der VM – anwendungskonsistent, empfohlen.

- Praktisch vor Updates – aber nicht als Dauer-Backup nutzen
- Lange offene Checkpoints kosten Leistung und Speicher

Checkpoints sind Hyper-Vs Momentaufnahmen einer VM.

Dynamischer Speicher

Dynamic Memory weist einer VM RAM nach tatsächlichem Bedarf zu.

- Start-, Minimal- und Maximalwert konfigurierbar
- Erhöht die VM-Dichte auf dem Host
- Für manche Workloads (z. B. bestimmte Datenbanken) besser fester RAM

Integration & Verschachtelung

- Integration Services liefern Treiber und Dienste in der VM
- Nested Virtualization: Hyper-V in einer VM – ideal für Labore
- Ressourcenmessung (Resource Metering) für Verrechnung
- Generation 2 nutzen, außer bei alten Gast-Systemen

Das Wichtigste

- In Windows Server integriert
- Gen 2 (UEFI/Secure Boot) bevorzugen
- Live Migration + Replica
- SCVMM/PowerShell zur Verwaltung

Proxmox VE

Open-Source-Virtualisierung mit VMs und Containern

Was ist Proxmox VE?

- Kostenlos nutzbar (optionales Support-Abo)
- Web-Oberfläche zur Verwaltung
- Beliebt in KMU und Homelabs

Eine offene Plattform, die KVM-VMs und LXC-Container vereint.

KVM + LXC

KVM

Vollwertige VMs mit eigenem Kernel.

LXC

Leichtgewichtige Linux-Container.

- VMs für beliebige OS, Container für schlanke Linux-Dienste
- Beides in einer Oberfläche

Clustering

Mehrere Proxmox-Knoten bilden einen Cluster mit zentraler Verwaltung.

- Migration von VMs zwischen Knoten
- Quorum für Konsistenz
- HA für automatischen Neustart

Storage & Ceph

- Lokaler Storage, NFS, iSCSI, ZFS
- Ceph als verteilter, redundanter Speicher
- Shared Storage für Live-Migration
- Snapshots und Backups integriert

Migration von VMware

- Import bestehender VMs möglich
- Kostenargument nach Lizenzänderungen
- Umschulung des Teams einplanen

Proxmox ist eine häufig genannte VMware-Alternative.

Backup mit vmdump & PBS

- `vmdump` sichert ganze VMs und Container
- Proxmox Backup Server (PBS): dedupliziert und inkrementell
- Zeitpläne pro VM/Container definierbar
- Restore regelmäßig testen – wie überall

Proxmox bringt Backup direkt mit.

Vorlagen & Linked Clones

Aus einer VM oder einem Container wird per Vorlage schnell eine ganze Flotte.

- Template aus einer fertig eingerichteten VM erstellen
- Linked Clone spart Speicher (teilt die Basis)
- Full Clone für eigenständige Kopien

Sicherheit & Rechte

- Integrierte Firewall auf Host-, VM- und Cluster-Ebene
- Rollenbasierte Rechteverwaltung für Nutzer
- Zwei-Faktor-Authentifizierung für die Web-Oberfläche
- Updates aus dem passenden Repository (Enterprise oder No-Subscription)

Das Wichtigste

- KVM (VMs) + LXC (Container) vereint
- Web-GUI, Cluster, HA
- Ceph für verteilten Speicher
- Ernsthafte, günstige VMware-Alternative

Docker & Container-Grundlagen

Anwendungen leichtgewichtig verpacken

Container vs. VM

Container teilen sich den Kernel des Hosts – dadurch viel leichter als VMs.

- Sekundenschnelles Starten
- Weniger Overhead als eine VM
- Isoliert auf Prozessebene, nicht auf Hardware-Ebene

Docker-Architektur

- Docker Engine (Daemon) führt Container aus
- CLI/API zur Steuerung
- Läuft unter Linux nativ, sonst über eine VM
- Registry als Bildquelle

Images & Container

Image

Unveränderliche Vorlage (aus einem Dockerfile gebaut).

Container

Laufende Instanz eines Images.

- Images in Schichten (Layers)
- Container sind kurzlebig/ersetzbar

Daten & Netzwerk

- Volumes für persistente Daten
- Ports vom Host auf Container mappen
- Eigene Netzwerke für Container-Kommunikation
- Umgebungsvariablen für Konfiguration

Docker Compose

Mehrere Container als ein Stack in einer `docker-compose.yml` beschreiben.

- Ein Befehl startet die ganze Anwendung
- Ideal für App + Datenbank + Cache
- Versionierbar und reproduzierbar

Sicherheit

- Vertrauenswürdige Images verwenden
- Nicht als root im Container laufen
- Images aktuell halten (CVEs)
- Secrets nicht ins Image backen

Das Dockerfile

Wichtige Befehle: FROM , RUN , COPY , CMD .

- Jede Anweisung erzeugt eine Schicht (Layer) – gut für Caching
- Kleine Basis-Images wählen (z. B. Alpine)
- Multi-Stage-Builds halten das Endergebnis schlank

Ein Dockerfile beschreibt Schritt für Schritt, wie ein Image gebaut wird.

Registry & Tags

- Images liegen in einer Registry – Docker Hub oder privat
- Tags kennzeichnen Versionen (z. B. `app:1.2`)
- Nur vertrauenswürdige Images verwenden
- Images auf Schwachstellen scannen (siehe LB-07)

Betrieb & Grenzen

- `docker logs` und `docker stats` zur Beobachtung
- Ressourcenlimits (CPU/RAM) und Restart-Policies setzen
- Docker allein orchestriert nicht über viele Hosts
- Dafür kommt Kubernetes (nächstes Kapitel)

Das Wichtigste

- Container teilen den Kernel → leicht & schnell
- Image = Vorlage, Container = Instanz
- Volumes für Daten, Ports für Zugriff
- Compose bündelt Multi-Container-Apps

Kubernetes-Einstieg

Container im großen Maßstab orchestrieren

Warum Kubernetes?

- Skalierung, Selbstheilung, Rollouts
- Wenn Docker allein nicht mehr reicht
- De-facto-Standard für Container-Orchestrierung

Verwaltet viele Container automatisch über mehrere Hosts.

Grundbegriffe

- Pod: kleinste Einheit (ein/mehrere Container)
- Deployment: verwaltet Pods und Rollouts
- Service: stabile Adresse für Pods
- Namespace: logische Trennung

Selbstheilung & Skalierung

K8s startet abgestürzte Pods neu und skaliert nach Bedarf.

- Gewünschter Zustand wird durchgesetzt
- Rolling Updates ohne Ausfall
- Horizontale Skalierung nach Last

kubectl

- `kubectl get pods` zeigt Zustände
- `kubectl apply -f` rollt YAML aus
- `kubectl logs / describe` zur Diagnose

Das zentrale Kommandozeilen-Werkzeug.

Helm & Einstieg

- Helm-Charts paketieren Anwendungen
- K3s/Minikube für erste Schritte
- Managed K8s (AKS/EKS/GKE) nimmt Arbeit ab
- Klein anfangen, Konzepte üben

Konfiguration & Secrets

- ConfigMaps für nicht-geheime Einstellungen
- Secrets für Passwörter und Schlüssel
- So bleibt dasselbe Image in Test und Produktion nutzbar
- Secrets zusätzlich absichern (RBAC, Verschlüsselung)

Konfiguration gehört nicht ins Image.

Ingress & Skalierung

Ein Ingress regelt den externen Zugriff und das Routing zu den Services.

- Ingress-Controller als zentraler Eingang (mit TLS)
- Horizontal Pod Autoscaler skaliert nach Last
- Ressourcen-Requests und -Limits pro Pod festlegen

Betrieb & Troubleshooting

- `kubectl describe` und `kubectl get events` zur Diagnose
- Health-Probes: Liveness (läuft noch?) und Readiness (bereit?)
- Rolling Update mit automatischem Rollback bei Fehlern
- Logs zentral einsammeln (siehe LB-09)

Das Wichtigste

- K8s orchestriert Container über viele Hosts
- Pod/Deployment/Service/Namespace
- Selbstheilung + Rolling Updates
- `kubectl` + Helm; K3s zum Lernen



LB-06 · Storage & Backup

6 Kapitel

RAID-Systeme

Datenträger zu Verbünden zusammenschalten

Was ist RAID?

- Redundant Array of Independent Disks
- Schützt vor Plattenausfall (je nach Level)
- Kein Schutz vor Löschen/Ransomware

Mehrere Datenträger wirken als ein Verbund – für Tempo und/oder Sicherheit.

Die wichtigsten Level

RAID 0 / 1

Striping (Tempo, keine Sicherheit) · Spiegelung (Sicherheit).

RAID 5 / 6

Parität über 3+ bzw. 4+ Platten – verträgt 1 bzw. 2 Ausfälle.

- RAID 10: Spiegel + Stripe (schnell + sicher)
- Level nach Anforderung wählen

Hardware vs. Software

Hardware-RAID

Eigener Controller, oft mit Cache/BBU.

Software-RAID

Vom OS verwaltet (z. B. ZFS, mdadm, Storage Spaces).

- ZFS bietet zusätzlich Prüfsummen
- Controller-Ausfall einkalkulieren

Rebuild-Risiko

Beim Wiederherstellen (Rebuild) sind die übrigen Platten stark belastet – ein zweiter Ausfall droht.

- Große Platten → langer Rebuild
- RAID 6 puffert einen zweiten Ausfall
- Hot-Spare hält Ersatz bereit

RAID ≠ Backup

- Schützt nur vor Hardware-Ausfall
- Gelöschtes bleibt gelöscht
- Ransomware verschlüsselt den Verbund mit
- Backup ist zusätzlich zwingend

Nutzbare Kapazität

RAID 1 / 10

Nur die Hälfte der Rohkapazität nutzbar (Spiegelung).

RAID 5 / 6

n minus 1 bzw. n minus 2 Platten nutzbar (Parität).

- Faustregel: mehr Sicherheit = weniger nutzbarer Platz
- Kapazität, Tempo und Sicherheit gegeneinander abwägen

Sicherheit kostet Kapazität – je nach Level unterschiedlich viel.

Hot-Spare & Prüfläufe

Ein Hot-Spare springt beim Ausfall automatisch ein und verkürzt die Zeit im ungeschützten Zustand.

- Controller- und SMART-Alarme aktivieren
- Regelmäßiges Scrubbing / Patrol Read gegen stille Fehler
- Ausgefallene Platten zeitnah tauschen

Praxis-Tipps

- Möglichst identische Platten (Größe, Typ, Firmware)
- Controller-Firmware aktuell halten
- Bei Hardware-RAID: Cache-Batterie (BBU) prüfen
- Rebuild-Dauer bei großen Platten einplanen

Das Wichtigste

- Level nach Tempo/Sicherheit wählen
- RAID 10/6 für wichtige Systeme
- Rebuild-Risiko bei großen Platten
- RAID ersetzt niemals ein Backup

SAN & NAS

Zentraler Speicher im Netzwerk

NAS vs. SAN

NAS

Datei-basiert (SMB/NFS) – einfach, für Freigaben.

SAN

Block-basiert – erscheint wie eine lokale Platte, für Server/VMs.

- NAS: Dateien teilen
- SAN: hochperformanter Block-Speicher

NAS im Einsatz

- Freigaben per SMB (Windows) / NFS (Linux)
- Backups, Fileservices, Medien
- Hersteller: QNAP, Synology u. a.
- Einfach zu betreiben

SAN-Konzept

Ein SAN stellt Server über ein eigenes Speichernetz Block-Speicher (LUNs) bereit.

- Getrenntes, schnelles Speichernetz
- Ideal für VM-Datastores und Datenbanken
- Hohe Verfügbarkeit über mehrere Pfade

Anbindungen

iSCSI

Block-Speicher über normales Ethernet.

Fibre Channel

Dediziertes, sehr schnelles Speichernetz.

- iSCSI günstiger, FC leistungsstark
- Multipathing gegen Pfadausfälle

Auswahlkriterien

- Datei- vs. Block-Bedarf klären
- Leistung (IOPS/Durchsatz)
- Redundanz und Skalierbarkeit
- Budget und Betriebsaufwand

Protokolle im Detail

SMB / NFS

Datei-Zugriff: SMB für Windows, NFS für Linux/Unix.

iSCSI

Initiator (Server) verbindet sich mit dem Target (Storage).

- Eigenes, schnelles Speichernetz einplanen
- Jumbo Frames können den Durchsatz erhöhen

Verfügbarkeit

- Redundante Controller im Storage-System
- Multipathing gegen Pfadausfälle
- Snapshots auf Storage-Ebene für schnelle Wiederherstellung
- Replikation zwischen Standorten für Disaster-Fälle

Dimensionierung

Nicht nur Kapazität zählt – auch IOPS und Durchsatz müssen zur Last passen.

- Kapazität plus Reserve für Wachstum
- IOPS für Datenbanken/Virtualisierung beachten
- Herstellersupport und Ersatzteil-Verfügbarkeit prüfen

Das Wichtigste

- NAS = Dateien, SAN = Blöcke
- SMB/NFS vs. iSCSI/Fibre Channel
- SAN für VMs/DBs mit hoher Last
- Multipathing für Ausfallsicherheit

Backup-Grundlagen & Strategien

Daten zuverlässig sichern und wiederherstellen

Warum Backups?

- Backup ist die letzte Verteidigungslinie
- Nur ein getesteter Restore zählt
- Datenverlust kann Existenzen kosten

Hardware fällt aus, Menschen löschen, Ransomware verschlüsselt.

Die 3-2-1-1-0-Regel

3 Kopien · 2 Medien · 1 extern · 1 offline/unveränderlich · 0 Fehler beim Restore-Test.

- Mehrere Kopien gegen Einzelfehler
- Offline/immutable gegen Ransomware
- Regelmäßig verifizieren

Backup-Arten

Voll / Differenziell

Alles · Änderungen seit letzter Vollsicherung.

Inkrementell / Synthetisch

Änderungen seit letztem Backup · nachträglich zusammengesetzte Vollsicherung.

- Voll = einfach, aber groß
- Inkrementell = sparsam, längere Kette

RPO & RTO

RPO

Wie viel Datenverlust ist akzeptabel? (Backup-Intervall)

RTO

Wie schnell muss es wieder laufen? (Wiederherstellzeit)

- Beide bestimmen Strategie und Kosten
- Mit dem Business abstimmen

Aufbewahrung

- Aufbewahrungsfristen definieren
- Großvater-Vater-Sohn-Rotation
- Gesetzliche Vorgaben beachten
- Ältere Sicherungen auslagern

Wohin sichern?

- Disk-to-Disk für schnelle Wiederherstellung
- Band für günstige, langfristige Archivierung
- Cloud als externes Ziel (das „ausser Haus“-Exemplar)
- Air-Gap oder unveränderlich (immutable) gegen Ransomware

Was sichern?

- Nutzdaten, System-State und Konfigurationen
- Ganze VMs und Datenbanken (mit eigenen Methoden)
- Kritische Systeme zuerst und häufiger
- Auch die Dokumentation einbeziehen

Nicht alles ist gleich wichtig – priorisieren.

Notfallplan (DR)

Backups sind nur die halbe Miete – die Wiederanlauf-Reihenfolge muss geübt sein.

- Runbooks: Wer stellt was in welcher Reihenfolge wieder her?
- Ausweichstandort oder Cloud-Recovery vorbereiten
- Regelmäßige Notfallübungen durchführen

Das Wichtigste

- 3-2-1-1-0 als Leitlinie
- Voll/Diff/Inkrementell bewusst kombinieren
- RPO/RTO mit dem Business festlegen
- Restore testen – sonst kein Backup

VSS, Snapshots & agentenloses Backup

Konsistente Sicherungen im laufenden Betrieb

Das Konsistenz-Problem

- Einfaches Kopieren erfasst nur einen Moment
- Anwendungen brauchen einen sauberen Stand
- Sonst droht ein unbrauchbares Backup

Offene Dateien und Datenbanken müssen konsistent gesichert werden.

VSS (Windows)

Der Volume Shadow Copy Service erzeugt konsistente Momentaufnahmen im Betrieb.

- VSS-Writer bringen Apps in einen sicheren Zustand
- VSS-Provider erstellen den Snapshot
- Grundlage für Windows-Backups und Vorversionen

VM-Snapshots

VMware VADP

Snapshot-basiertes Backup der VM.

Hyper-V CBT

Changed Block Tracking sichert nur geänderte Blöcke.

- Nur geänderte Blöcke → schnelle Backups
- Snapshots nicht dauerhaft offen lassen

Agentenlos

- Weniger Aufwand, keine Agents pflegen
- Sichert die ganze VM auf einmal
- Application-aware für DB-Konsistenz

Backup über den Hypervisor statt per Agent in jeder VM.

Wichtige Warnung

Snapshots sind KEIN Backup – sie liegen auf demselben Storage.

- Nur kurzfristige Zwischenstände
- Wachsen und kosten Leistung
- Immer in ein echtes Backup überführen

Crash- vs. app-konsistent

App-konsistente Sicherungen bringen Anwendungen vorher in einen sauberen Zustand (Quiescing).

- Crash-konsistent = wie nach einem Stromausfall
- App-konsistent = Datenbanken sind sauber geschlossen
- VSS-Writer bzw. Pre-/Post-Skripte sorgen dafür

Snapshot-Lebenszyklus

- Nach dem Backup automatisch wieder entfernen
- Delta-Dateien wachsen und kosten Leistung
- Nie als dauerhaften Zustand offen lassen
- Speicherplatz für Snapshots überwachen

Grenzen

- Liegt auf demselben Storage wie das Original
- Kein Schutz bei Storage- oder Ransomware-Ausfall
- Immer in ein echtes, ausgelagertes Backup überführen

Ein Snapshot fühlt sich an wie ein Backup – ist aber keines.

Das Wichtigste

- VSS sorgt für konsistente Windows-Sicherungen
- CBT/VADP sichern nur geänderte Blöcke
- Agentenlos = weniger Aufwand
- Snapshot ≠ Backup

Datenbank-Backup

Datenbanken korrekt sichern und wiederherstellen

Warum speziell?

- Transaktionen dürfen nicht zerrissen werden
- Eigene Backup-Mechanismen nutzen
- Point-in-Time-Wiederherstellung als Ziel

Datenbanken sind ständig in Bewegung – Dateikopien allein reichen nicht.

SQL Server

Full / Differential

Ganze DB · Änderungen seit Voll-Backup.

Transaction Log

Ermöglicht Wiederherstellung auf die Sekunde.

- Recovery-Modell bestimmt die Möglichkeiten
- Log-Sicherung hält das Log klein

MySQL / MariaDB

- `mysqldump` für logische Backups
- Percona XtraBackup für heiße physische Backups
- Binlogs für Point-in-Time
- Konsistenz bei laufendem Betrieb beachten

PostgreSQL

- `pg_dump` / `pg_dumpall` logisch
- Basis-Backup + WAL-Archivierung physisch
- Point-in-Time-Recovery über WAL
- Regelmäßig gegen Restore prüfen

Gute Praxis

Ohne getesteten Restore ist ein DB-Backup nur eine Hoffnung.

- Restore regelmäßig üben
- Backups außerhalb der DB aufbewahren
- Aufbewahrung/Compliance beachten
- Backups überwachen (Erfolg/Fehler)

Recovery-Modelle

Simple

Kein Transaktionslog-Backup – nur Voll/Diff, kein Point-in-Time.

Full

Log-Backups möglich – Wiederherstellung auf die Sekunde.

- Das Modell bestimmt, was überhaupt möglich ist
- Bei Full das Log regelmäßig sichern (sonst wächst es)

Heiße Backups

- Sicherung im laufenden Betrieb ohne langes Sperren
- Wartungsfenster für aufwändige Operationen
- Auswirkungen auf die Leistung im Blick behalten
- Konsistenz über die DB-eigenen Mechanismen sicherstellen

Aufbewahrung & Test

Ein DB-Backup zählt erst, wenn der Restore in einer Testinstanz nachweislich klappt.

- Aufbewahrungsketten definieren (täglich/wöchentlich/monatlich)
- Backups ausgelagert und geschützt speichern
- Restore regelmäßig üben und dokumentieren

Das Wichtigste

- DB-eigene Mechanismen statt Dateikopie
- Log/WAL ermöglichen Point-in-Time
- Je System eigene Werkzeuge
- Restore testen ist Pflicht

Backup-Software & Hersteller

Werkzeuge für die Datensicherung

Warum dedizierte Software?

- Application-aware Backups
- Berichte und Alarmer
- Verschlüsselung und Deduplizierung

Zentrale Steuerung, Zeitpläne, Überwachung und schnelle Restores.

Bekannte Anbieter

Veeam / Nakivo

Stark bei Virtualisierung und modernen Umgebungen.

Acronis / Barracuda

Breite Lösungen inkl. Cyber-Protection.

- Auswahl nach Umgebung und Budget
- Immutable/Offsite-Optionen prüfen

Cloud-Backup

- Azure Backup / AWS Backup
- Als externes „1“ in 3-2-1-1-0
- Kosten für Speicher und Zugriff beachten
- Wiederherstellzeiten aus der Cloud einplanen

Wichtige Funktionen

- Deduplizierung und Kompression
- Unveränderliche (immutable) Backups
- Verschlüsselung in Ruhe und Transport
- Granularer Restore (einzelne Dateien/Mails)

Restore-Tests als Pflicht

Ein Backup ist erst dann eines, wenn der Restore nachweislich funktioniert.

- Regelmäßige, dokumentierte Tests
- Auch komplette Notfall-Wiederherstellung üben
- Wiederanlaufzeiten (RTO) messen

Auswahlkriterien

- Unterstützte Workloads: VMware, Hyper-V, physisch, M365, Datenbanken
- Lizenzmodell und Skalierbarkeit
- Immutable- und Offsite-Optionen
- Herstellersupport und Reifegrad

Ransomware-Resilienz

Backups sind das Hauptziel von Ransomware – sie müssen besonders geschützt sein.

- Unveränderliche (immutable) Repositories
- Air-Gap oder getrennte Netzsegmente
- Eigene Backup-Zugangsdaten, nicht die Domänen-Admins
- MFA für die Backup-Konsole

Betrieb

- Backup-Jobs überwachen, Alarme bei Fehlern
- Kapazität planen (Dedup/Kompression)
- Auch Backup-Server und -Konfiguration sichern
- Regelmäßige Restore-Tests als fester Prozess

Das Wichtigste

- Backup-Software zentralisiert und automatisiert
- Veeam/Nakivo/Acronis u. a. je nach Umgebung
- Immutable + Offsite gegen Ransomware
- Restore-Tests sind nicht verhandelbar



LB-07 · IT-Security

10 Kapitel

Schutzziele & Grundlagen

CIA-Triade, Risiko und Defense in Depth

Die CIA-Triade

- Vertraulichkeit: nur Befugte sehen Daten
- Integrität: Daten werden nicht unbemerkt verändert
- Verfügbarkeit: Systeme laufen, wenn man sie braucht
- Das Grundgerüst jeder Sicherheitsbetrachtung

Plus Authentizität

Oft ergänzt: Authentizität – die Sicherheit, dass jemand/etwas wirklich das ist, was es vorgibt.

- Basis für Vertrauen und Nachweisbarkeit

Zielkonflikte

- Maximale Vertraulichkeit kann Verfügbarkeit einschränken
- Kein Ziel „gewinnt“ absolut
- Balance je nach Schutzbedarf

Sicherheit ist immer ein Abwägen.

Risiko & Risk-Management

Risiko = Bedrohung trifft Schwachstelle.

- Risiken erkennen und bewerten
- Optionen: vermeiden, vermindern, übertragen, akzeptieren
- Nicht jedes Risiko ist wirtschaftlich ausschließbar

Angriffsfläche

- Jeder offene Port/Dienst/zu weites Recht vergrößert sie
- Grundprinzip: minimale Rechte
- Grundprinzip: minimale Angriffsfläche

Defense in Depth

Keine einzelne Maßnahme ist perfekt – erst das Zusammenspiel schützt wirklich.

- Technik, Prozesse und geschulte Menschen
- Mehrere Schichten hintereinander

Nichtabstreitbarkeit

- Umgesetzt über Protokolle (Logs) und digitale Signaturen
- Wichtig bei Verträgen, Überweisungen und Admin-Aktionen
- Ergänzt Vertraulichkeit, Integrität und Verfügbarkeit

Eine Handlung lässt sich nachweislich einer Person zuordnen – sie kann nicht geleugnet werden.

Schutzbedarf feststellen

normal

Überschaubarer Schaden – Standard-Maßnahmen genügen.

hoch / sehr hoch

Existenzbedrohend – zusätzliche, strengere Maßnahmen.

- Grundlage für Priorisierung nach BSI IT-Grundschutz

Nicht alles ist gleich schützenswert – der Aufwand richtet sich nach dem Schaden.

Das Wichtigste

- CIA: Vertraulichkeit, Integrität, Verfügbarkeit (+ Authentizität)
- Risiko = Bedrohung trifft Schwachstelle
- Minimale Rechte und Angriffsfläche
- Sicherheit immer mehrschichtig denken

Malware & Angriffstechniken

Von Viren und Ransomware bis Social Engineering

Was ist Malware?

- Datendiebstahl, Erpressung, Sabotage oder Spionage
- Verbreitung über Mail, Web, USB oder Schwachstellen
- Der Mensch ist oft das schwächste Glied

Sammelbegriff für schädliche Software mit unterschiedlichen Zielen.

Klassische Typen

Virus / Wurm

Virus hängt sich an Dateien; Wurm verbreitet sich selbst über das Netz.

Trojaner

Tarnt sich als nützliches Programm und öffnet die Hintertür.

- Würmer brauchen keinen Nutzer zum Weiterverbreiten
- Trojaner laden oft weitere Schadsoftware nach

Ransomware

Verschlüsselt Daten und erpresst Lösegeld – die derzeit größte Bedrohung für Unternehmen.

- Häufig „Double Extortion“: Verschlüsseln + Daten veröffentlichen
- Bester Schutz: getestete, offline gehaltene Backups
- Schnelle Ausbreitung im flachen Netz → Segmentierung hilft

Spyware & Rootkits

Spyware

Sammelt heimlich Daten (Tastatureingaben, Zugänge).

Rootkit

Nistet sich tief ein und versteckt seine Präsenz.

- Rootkits sind schwer zu entdecken und zu entfernen
- Oft ist Neuaufsetzen die sicherste Lösung

APT

- Meist gut ausgestattete Gruppen mit klarem Ziel
- Lange unbemerkt im Netz (Monate)
- Kombinieren mehrere Techniken und Zero-Days

Advanced Persistent Threat – gezielte, langfristige Angriffe.

Social Engineering

Der Angriff auf den Menschen – Phishing ist der häufigste Einstieg.

- Phishing: gefälschte Mails locken zu Klicks/Eingaben
- Spear-Phishing: gezielt auf Einzelpersonen
- BEC (CEO-Fraud): gefälschte Chef-Anweisung zur Überweisung

Schutz auf mehreren Ebenen

- Awareness-Schulungen für Mitarbeitende
- Patchen, MFA und minimale Rechte
- Backups, Segmentierung und Monitoring
- Mail- und Endpoint-Schutz kombinieren

Die Angriffskette

- Erstzugang (Phishing, Schwachstelle)
- Ausführung und Persistenz
- Laterale Bewegung im Netz
- Datenabfluss (Exfiltration) oder Verschlüsselung

Angriffe laufen in Phasen ab – die Kill Chain hilft, sie zu verstehen.

Erstmaßnahmen im Vorfall

Bei Ransomware-Verdacht: System isolieren, aber für die Forensik nicht einfach ausschalten.

- Betroffene Systeme vom Netz trennen
- Zugangsdaten zurücksetzen
- IT-Leitung und ggf. Behörden informieren

Das Wichtigste

- Malware = Viren, Würmer, Trojaner, Ransomware, Spyware, Rootkits
- Ransomware ist die Top-Bedrohung – Backups sind Pflicht
- Der Mensch ist Hauptangriffsziel (Phishing/BEC)
- Schutz immer mehrschichtig denken

Antivirus & NGAV

Von Signaturen zu verhaltensbasierter Erkennung

Klassisches Antivirus

- Sehr zuverlässig bei bekannter Malware
- Braucht ständige Signatur-Updates
- Blind gegenüber brandneuen Varianten

Erkennt Schadsoftware anhand bekannter Signaturen (Fingerabdrücke).

Heuristik & Verhalten

Statt nur „Kenne ich das?“ fragt moderne Erkennung „Verhält sich das verdächtig?“.

- Analyse verdächtiger Aktionen (z. B. Massen-Verschlüsselung)
- Erkennt auch unbekannte Varianten
- Höheres Risiko für Fehlalarme

NGAV

- Maschinelles Lernen statt reiner Signaturen
- Cloud-Lookups für aktuelle Bedrohungsdaten
- Fokus auf Techniken, nicht auf einzelne Dateien

Next-Generation Antivirus kombiniert mehrere Techniken.

False Positives

- Legitime Software wird fälschlich blockiert
- Ausnahmen mit Bedacht und dokumentiert setzen
- Zu viele Fehlalarme führen zu „Abstumpfung“

Fehlalarme sind die Kehrseite aggressiver Erkennung.

AV-Evasion

- Packen/Verschlüsseln des Schadcodes
- Dateilose Angriffe direkt im Arbeitsspeicher
- „Living off the Land“: missbraucht legitime Tools

Angreifer versuchen, Erkennung zu umgehen.

Grenzen → EDR

Reines AV sieht oft nur die Datei, nicht die gesamte Angriffskette.

- Dateilose und mehrstufige Angriffe entgehen ihm
- Keine Rückschau, wenn etwas durchrutscht
- Deshalb der Schritt zu EDR/XDR (nächstes Kapitel)

Betrieb im Unternehmen

- Zentrale Verwaltung aller Endpunkte
- Ausnahmen begründet und dokumentiert
- Definitionen/Engine aktuell halten
- Regelmäßige und Echtzeit-Scans

Grenzen & Ergänzung

- Dateilose und LotL-Angriffe entgehen reinem AV
- Ergänzung durch EDR (nächstes Kapitel)
- Application Control gegen unerwünschte Programme
- Awareness gegen Social Engineering

Das Wichtigste

- Signaturen: stark bei Bekanntem, blind bei Neuem
- Verhaltensbasiert/NGAV erkennt auch Unbekanntes
- Fehlalarme bewusst managen
- AV allein reicht heute nicht mehr

EDR & XDR

Endpunkte überwachen, Angriffe erkennen und reagieren

Warum EDR?

Annahme: Ein Angreifer kommt irgendwann durch – man muss ihn erkennen und stoppen können.

- Kontinuierliche Aufzeichnung von Endpunkt-Aktivität
- Erkennt verdächtige Ketten, nicht nur einzelne Dateien
- Ermöglicht gezielte Reaktion

Endpoint Detection and Response geht über das reine Blockieren hinaus.

Wie EDR arbeitet

- Sammelt Telemetrie (Prozesse, Netzwerk, Registry, Dateien)
- Korreliert Ereignisse zu einer Angriffsgeschichte
- Alarmiert das Security-Team mit Kontext
- Reaktion: Host isolieren, Prozess beenden, zurückrollen

Forensik & Threat Hunting

- Nachvollziehen, wie ein Angriff ablief
- Proaktiv nach Spuren suchen (Threat Hunting)
- Indikatoren (IOCs) im ganzen Bestand suchen

EDR liefert die Datenbasis für Analyse und aktive Suche.

Von EDR zu XDR

XDR erweitert die Sicht vom Endpunkt auf Netzwerk, Mail, Identität und Cloud.

EDR

Fokus auf den Endpunkt.

XDR

Korreliert Signale über mehrere Ebenen hinweg.

- Mehr Kontext = weniger blinde Flecken
- Zusammenhänge über Systeme hinweg sichtbar

SOAR

- Playbooks für wiederkehrende Vorfälle
- Automatische Erstmaßnahmen (z. B. Konto sperren)
- Entlastet das Team von Routine

Security Orchestration, Automation and Response automatisiert die Reaktion.

Bekannte Lösungen

- CrowdStrike Falcon
- SentinelOne
- Microsoft Defender for Endpoint
- Auswahl nach Umgebung, Integration und Budget

Auswahl & Betrieb

- Möglichst lückenlose Sensor-Abdeckung
- Managed EDR / SOC bei wenig Personal
- Regeln tunen, um Fehlalarme zu senken
- Reaktions-Playbooks vorbereiten

MITRE ATT&CK

ATT&CK ist ein Katalog realer Angriffstechniken – ideal, um Erkennung und Lücken einzuordnen.

- Techniken statt nur Signaturen betrachten
- Eigene Abdeckung gegen ATT&CK mappen
- Gemeinsame Sprache im Team

Das Wichtigste

- EDR: Endpunkte aufzeichnen, erkennen, reagieren
- XDR: Sicht über Endpoint + Netzwerk + Cloud + Identität
- SOAR automatisiert die Reaktion
- Grundhaltung: Erkennung statt reiner Prävention

Firewall & Netzwerksicherheit

Vom Paketfilter zur Next-Generation Firewall

Die Firewall als Torwächter

- Trennt vertrauenswürdige von unsicheren Zonen
- Grundprinzip: alles verbieten, gezielt erlauben
- Zentraler Baustein der Netzwerksicherheit

Kontrolliert den Verkehr zwischen Netzen anhand von Regeln.

Entwicklungsstufen

Paketfilter

Entscheidet nur nach IP/Port – zustandslos, simpel.

Stateful

Verfolgt Verbindungszustände – Antworten werden automatisch erlaubt.

- Stateful ist heute der Mindeststandard
- Reine Paketfilter sind zu grob

Next-Generation Firewall

Die NGFW erkennt Anwendungen und Inhalte, nicht nur Ports.

- Application Control (z. B. gezielt Apps sperren)
- Deep Packet Inspection (DPI)
- Integriertes IPS und oft Malware-Schutz

Proxy als Ergänzung zur Firewall

Forward-Proxy

Filtert ausgehenden Verkehr: URL-/Kategorie-Filter, Caching, zentrale Protokollierung.

Reverse-Proxy

Steht vor einem Server, verbirgt dessen echte Adresse, verteilt Last, terminiert TLS.

Firewall und Proxy lösen unterschiedliche Aufgaben – oft im selben Gerät kombiniert.

SSL-Inspection

- Firewall entschlüsselt, prüft und verschlüsselt neu
- Nötig, weil fast alles TLS-verschlüsselt ist
- Datenschutz und Ausnahmen (Banking) beachten

Verschlüsselter Verkehr lässt sich sonst nicht prüfen.

IDS & IPS

IDS

Erkennt und meldet Angriffe (passiv).

IPS

Erkennt und blockiert aktiv (inline).

- Signatur- und anomaliebasiert
- Teil moderner NGFWs

Segmentierung & DMZ

- DMZ für von außen erreichbare Dienste
- Segmente begrenzen die Ausbreitung von Angriffen
- Regeln zwischen Segmenten bewusst setzen

Nicht alles gehört in dasselbe Netz.

Regelwerk pflegen

Kein „Any-Any“ – jede Regel so eng wie möglich und dokumentiert.

- Regeln kommentieren (Zweck, Ersteller, Datum)
- Nicht mehr benötigte Regeln entfernen
- Regelmäßiges Review des Regelwerks

Zonen & Segmentierung

- Trust-Zonen mit klaren Übergängen
- Ost-West-Verkehr (intern) kontrollieren
- Mikrosegmentierung für kritische Systeme
- DMZ für öffentlich Erreichbares

Das Wichtigste

- Default Deny – nur gezielt erlauben
- Stateful ist Minimum, NGFW ist Standard
- SSL-Inspection, IPS und Segmentierung ergänzen
- DMZ trennt öffentliche Dienste vom internen Netz

Mail-Security & Anti-Spam

SPF, DKIM, DMARC und Schutz vor Phishing

E-Mail als Hauptangriffsweg

- Phishing, Malware-Anhänge, BEC
- Absender lassen sich leicht fälschen
- Schutz auf Protokoll- und Inhaltsebene nötig

Die meisten Angriffe beginnen mit einer E-Mail.

SPF

Sender Policy Framework: legt per DNS fest, welche Server für eine Domain senden dürfen.

- Empfänger prüft, ob der sendende Server berechtigt ist
- Schützt vor Absender-Fälschung der Envelope-Domain
- Als DNS-TXT-Eintrag hinterlegt

DKIM

- Signatur belegt Echtheit und Unverändertheit
- Öffentlicher Schlüssel liegt im DNS
- Empfänger verifiziert die Signatur

DomainKeys Identified Mail signiert ausgehende Mails kryptografisch.

DMARC

DMARC verknüpft SPF und DKIM mit einer Richtlinie: was tun, wenn die Prüfung scheitert?

- Aktionen: none / quarantine / reject
- Reports zeigen Missbrauch der eigenen Domain
- Erst beobachten (none), dann verschärfen

Inhaltsprüfung & Sandbox

- Anti-Spam- und Anti-Malware-Filter
- Anhänge in einer Sandbox detonieren lassen
- Links umschreiben und beim Klick prüfen (Time-of-Click)
- Warnbanner bei externen Absendern

Lösungen am Markt

- Hornetsecurity, Proofpoint, Mimecast
- Microsoft Defender for Office 365
- Vorgeschaltete Gateways oder cloud-integriert

Weitere Schutzschichten

- Anhänge nach Typ filtern
- URL-Rewrite mit Prüfung beim Klick
- Impersonation-Schutz (gefälschte Chefs)
- Externe Absender markieren

Nutzer einbeziehen

Die beste Technik nützt wenig ohne aufmerksame Nutzer.

- Regelmäßige Awareness-Schulungen
- Phishing-Simulationen
- Einfacher Melde-Button für Verdachtsfälle

Das Wichtigste

- E-Mail ist der Angriffsweg Nummer eins
- SPF + DKIM + DMARC gegen Absender-Fälschung
- Sandbox und Link-Schutz gegen Inhalte
- DMARC schrittweise scharf schalten

SSL/TLS & PKI

Verschlüsselte Verbindungen und Zertifikate verstehen

Verschlüsselung – die Grundlagen

Symmetrisch

Ein gemeinsamer Schlüssel für Ver- und Entschlüsselung – schnell, z. B. AES. Problem: sicherer Schlüsselaustausch.

Asymmetrisch

Schlüsselpaar aus öffentlich und privat, z. B. RSA. Löst den Austausch, ist aber rechenintensiver.

- TLS kombiniert beides: asymmetrisch für den Handshake, symmetrisch für die Nutzdaten
- Hashing (z. B. SHA-256) sichert zusätzlich die Integrität – keine Verschlüsselung, aber verwandtes Werkzeug

Zwei grundsätzliche Verfahren, auf denen TLS und viele andere Techniken aufbauen.

Wozu TLS?

- Vertraulichkeit: niemand liest mit
- Integrität: Daten werden nicht verändert
- Authentizität: der Server ist echt (Zertifikat)

TLS verschlüsselt Verbindungen und stellt Echtheit sicher.

Der TLS-Handshake

Aushandeln der Verschlüsselung + Prüfen des Serverzertifikats vor dem eigentlichen Datentransfer.

- Client und Server einigen sich auf Verfahren
- Schlüsselaustausch etabliert einen Sitzungsschlüssel
- Danach symmetrisch verschlüsselte Übertragung

Zertifikatskette

Server-Zertifikat

Für die konkrete Domain ausgestellt.

CA / Root

Vertrauenswürdige Stelle, im System hinterlegt.

- Zwischenzertifikate verbinden Server und Root
- Browser vertrauen den bekannten Root-CAs

Vertrauen entsteht über eine Kette bis zu einer Root-CA.

Let's Encrypt

Kostenlose, automatisierte Zertifikate – heute weit verbreitet.

- Automatische Ausstellung/Erneuerung (ACME)
- Kurze Laufzeit (90 Tage) → Automatisierung Pflicht
- Domain-validiert, ideal für Webserver

Interne PKI

- Windows: Active Directory Certificate Services (AD CS)
- Root-CA absichern (am besten offline)
- Interne Root muss auf Clients vertraut sein

Für interne Dienste eine eigene Zertifizierungsstelle betreiben.

Zertifikate im Blick behalten

- Ablaufdaten überwachen und rechtzeitig erneuern
- Automatisierung reduziert Fehler
- Schwache Verfahren/alte TLS-Versionen abschalten

Abgelaufene Zertifikate sind eine häufige Ausfallursache.

Zertifikatstypen

- DV, OV, EV nach Prüftiefe
- Wildcard für ganze Subdomänen
- SAN-Zertifikate für mehrere Namen
- Interne vs. öffentliche CA je Einsatz

Betrieb & Fehlerquellen

Die häufigsten Fehler: abgelaufene Zertifikate und fehlende Zwischenzertifikate.

- Ablauf zentral überwachen
- Vollständige Kette ausliefern
- Widerruf über CRL/OCSP
- Private Schlüssel streng schützen

Das Wichtigste

- TLS = Vertraulichkeit + Integrität + Authentizität
- Vertrauen über die Zertifikatskette zur Root-CA
- Let's Encrypt automatisiert öffentliche Zertifikate
- Interne PKI via AD CS – Ablauf-Monitoring nicht vergessen

Identity & Access Management

MFA, Passkeys, SSO und Zero-Trust-Zugriff

AuthN vs. AuthZ

Authentifizierung

Wer bist du? (Identität nachweisen).

Autorisierung

Was darfst du? (Rechte/Rollen).

- Least Privilege: nur so viel Rechte wie nötig
- Konten sauber verwalten (Joiner/Mover/Leaver)

Identität ist die neue Perimeter-Grenze.

Multi-Faktor-Authentifizierung

MFA kombiniert mehrere Faktoren – Wissen, Besitz, Biometrie.

- Deutlich stärker als reines Passwort
- App-basiert statt SMS (SMS ist angreifbar)
- Größter Einzel-Hebel gegen Kontenübernahme

Passkeys / FIDO2

Passkeys sind phishing-resistent – es gibt kein abgreifbares Passwort mehr.

- Kryptografisches Schlüsselpaar statt Passwort
- An die echte Domain gebunden → Phishing läuft ins Leere
- Anmeldung per Biometrie oder Sicherheitsschlüssel

SSO & Standards

- OAuth2 für Autorisierung, OIDC für Anmeldung
- SAML im klassischen Enterprise-Umfeld
- Weniger Passwörter = weniger Angriffsfläche

Single Sign-On: einmal anmelden, viele Dienste nutzen.

Privileged Access Management

- Getrennte Admin-Konten, kein Alltagsgebrauch
- Just-in-Time-Rechte statt Dauerberechtigung
- Sitzungen protokollieren, Passwörter im Tresor

Administrative Konten brauchen besonderen Schutz.

Zero-Trust-Grundsätze

Jeder Zugriff wird geprüft – unabhängig von Standort oder Netz.

- Identität + Gerätezustand + Kontext entscheiden
- Conditional Access setzt Richtlinien durch
- „Assume breach“ als Grundhaltung

Lebenszyklus (Joiner-Mover-Leaver)

- Automatisches Provisioning beim Eintritt
- Rechte anpassen bei Rollenwechsel
- Sofortiges Deprovisioning beim Austritt
- Regelmäßige Rezertifizierung der Rechte

Verzeichnisse & Föderation

Ein zentraler Identity Provider (IdP) föderiert die Anmeldung an viele Dienste.

- SAML und OIDC für Föderation
- SCIM für automatische Bereitstellung
- Weniger Passwörter, mehr Kontrolle

Das Wichtigste

- AuthN (wer) vs. AuthZ (was) – Least Privilege
- MFA ist Pflicht, Passkeys sind phishing-resistent
- SSO über OAuth2/OIDC reduziert Passwörter
- PAM schützt privilegierte Konten, Zero Trust prüft jeden Zugriff

SIEM & SOC-Grundlagen

Logs zentral auswerten und auf Vorfälle reagieren

Warum zentrale Logs?

- Angriffe hinterlassen Spuren über viele Systeme
- Verstreute Logs sind praktisch unauswertbar
- Zentrale Sammlung ist die Voraussetzung für Erkennung

Einzelne Logs sagen wenig – erst die Zusammenschau zeigt Angriffe.

Was ein SIEM tut

Security Information and Event Management: sammeln, korrelieren, alarmieren.

- Aggregation von Logs aus vielen Quellen
- Korrelation zu aussagekräftigen Ereignissen
- Alarmierung bei verdächtigen Mustern

Gängige SIEM-Tools

- Splunk
- Microsoft Sentinel (cloud-nativ)
- Graylog / Elastic
- Wazuh als Open-Source-Option

Das SOC

Tier 1

Sichtung und Triage von Alarmen.

Tier 2/3

Analyse, Threat Hunting, Incident Response.

- Rund-um-die-Uhr-Überwachung möglich
- Intern oder als Managed Service (MSSP)

Security Operations Center – das Team hinter den Werkzeugen.

Incident-Response-Prozess

Vorbereiten → Erkennen → Eindämmen → Beseitigen → Wiederherstellen → Lernen.

- Klare Rollen und ein Notfallplan vorab
- Eindämmung vor Beseitigung (Ausbreitung stoppen)
- Nachbereitung: was war die Ursache, was verbessern?

Alert-Fatigue vermeiden

- Regeln pflegen, Fehlalarme reduzieren
- Priorisieren nach Risiko
- Wichtiges darf nicht im Rauschen untergehen

Zu viele Alarme sind so gefährlich wie zu wenige.

Use Cases & Korrelation

- Detektionsregeln für konkrete Szenarien
- False Positives gezielt tunen
- Threat Intelligence einbinden
- Playbooks je Alarmtyp

Log-Quellen & Aufbewahrung

Ein SIEM ist nur so gut wie seine Quellen – und die Uhren müssen synchron laufen.

- AD, Firewall, EDR, Proxy als Kernquellen
- Zeit-Synchronisation (NTP) überall
- Gesetzliche Aufbewahrungsfristen beachten

Das Wichtigste

- SIEM: sammeln, korrelieren, alarmieren
- SOC ist das Team, SIEM das Werkzeug
- Incident Response als geübter Prozess
- Alarmqualität schlägt Alarmmenge

BSI IT-Grundschutz & Compliance

Methodik, Schutzbedarf und der Bezug zu ISO 27001 & DSGVO

BSI IT-Grundschutz

- Herausgegeben vom BSI (Bundesamt für Sicherheit in der Informationstechnik)
- Praxisnahe Bausteine mit konkreten Anforderungen
- Basis für ein ISMS (Managementsystem)

Ein bewährtes deutsches Rahmenwerk für Informationssicherheit.

Die Methodik

Bausteine beschreiben typische Objekte (z. B. Server, Netz, Anwendung) und ihre Anforderungen.

- Strukturanalyse: Was ist überhaupt vorhanden?
- Modellierung mit passenden Bausteinen
- Soll-Ist-Abgleich deckt Lücken auf

Schutzbedarfsfeststellung

normal / hoch

Überschaubarer bzw. beträchtlicher Schaden.

sehr hoch

Existenzbedrohende Auswirkungen.

- Bewertung je Vertraulichkeit, Integrität, Verfügbarkeit
- Höherer Bedarf → strengere Maßnahmen

Wie schützenswert ist ein Objekt hinsichtlich CIA?

Bezug zu ISO 27001

- IT-Grundschutz kann zur ISO-27001-Zertifizierung führen
- ISO ist generischer, Grundschutz konkreter
- Beide zielen auf einen kontinuierlichen Verbesserungsprozess

International anerkannter Standard für ISMS.

DSGVO-Bezug

Datenschutz verlangt „geeignete technische und organisatorische Maßnahmen“ (TOM).

- Sicherheitsmaßnahmen belegen die TOM-Pflicht
- Verfügbarkeit und Vertraulichkeit personenbezogener Daten
- Dokumentation ist im Ernstfall entscheidend

Prüfungsrelevanz

- CIA-Triade (Vertraulichkeit, Integrität, Verfügbarkeit)
- Grundbegriffe: ISMS, Schutzbedarf, Baustein
- Zusammenspiel BSI-Grundschutz ↔ ISO 27001 ↔ DSGVO

Vorgehen im Detail

- Basis-, Standard- und Kern-Absicherung
- Bausteine → Anforderungen → Maßnahmen
- Strukturanalyse und Modellierung
- Soll-Ist-Abgleich deckt Lücken auf

Rollen & Nachweise

Die Leitung trägt die Verantwortung – Restrisiken werden bewusst akzeptiert und dokumentiert.

- Informationssicherheitsbeauftragter (ISB)
- Dokumentation als Audit-Nachweis
- Kontinuierlicher Verbesserungsprozess

Das Wichtigste

- IT-Grundschutz = praxisnaher Weg zu einem ISMS
- Schutzbedarf normal / hoch / sehr hoch je CIA
- Grundschutz kann zur ISO-27001-Zertifizierung führen
- Sicherheitsmaßnahmen erfüllen zugleich DSGVO-TOM



LB-08 · Cloud

8 Kapitel

Cloud-Grundlagen & Modelle

IaaS/PaaS/SaaS, Bereitstellung & geteilte Verantwortung

Was ist Cloud?

- Zahlung nach Verbrauch
- Schnell hoch- und runterskalieren
- Aber: nicht alles gehört in die Cloud

IT-Ressourcen als Dienst über das Internet – statt eigener Hardware.

Service-Modelle

IaaS / PaaS

Virtuelle Maschinen bzw. fertige Plattform.

SaaS

Fertige Software wie Microsoft 365.

- IaaS: du verwaltest das Betriebssystem
- SaaS: du nutzt nur die Anwendung

Bereitstellung

- Public: beim Anbieter, geteilt
- Private: exklusiv für eine Organisation
- Hybrid: Kombination aus beidem

Geteilte Verantwortung

Der Anbieter sichert die Cloud – du sicherst, was du darin betreibst.

- Konfiguration ist Kundensache
- Fehlkonfiguration ist das Hauptrisiko

Datenschutz

- DSGVO bei personenbezogenen Daten
- Datenstandort und Auftragsverarbeitung prüfen
- Verschlüsselung und Zugriffskontrolle

Skalierung & Elastizität

Vertikal (scale up)

Eine Maschine bekommt mehr CPU/RAM.

Horizontal (scale out)

Mehr Maschinen parallel – ideal für Lastspitzen.

- Auto-Scaling startet Instanzen automatisch nach Last
- Nach der Spitze wird wieder heruntergefahren – das spart Geld

Ressourcen wachsen und schrumpfen mit dem tatsächlichen Bedarf.

Kostenmodell

Kostenfallen vermeiden: Budgets, Alarmer und ein Tagging-Konzept von Anfang an einrichten.

- Pay-as-you-go: nur genutzte Ressourcen werden bezahlt
- OpEx statt CapEx – laufende Kosten statt großer Anschaffung
- Reservierungen und Spot-Instanzen senken den Preis
- Achtung: Datenabfluss (Egress) und ungenutzte Ressourcen kosten schnell

Die großen Anbieter

Hyperscaler

AWS, Microsoft Azure und Google Cloud – riesiger Dienst-Katalog.

Regional/EU

Anbieter mit EU-Rechenzentren für strengere Datenschutz-Anforderungen.

- Multi-Cloud verteilt auf mehrere Anbieter – flexibel, aber komplexer
- Vendor-Lock-in: Abhängigkeit von einem Anbieter bewusst abwägen

Das Wichtigste

- Cloud = Ressourcen als Dienst, nach Verbrauch
- IaaS/PaaS/SaaS je nach Tiefe
- Public/Private/Hybrid als Bereitstellung
- Geteilte Verantwortung + DSGVO beachten

Microsoft 365 – Administration

Tenant, Dienste und Sicherheit verwalten

Was ist Microsoft 365?

- Exchange Online, SharePoint, Teams, OneDrive
- Identitäten über Entra ID (früher Azure AD)
- Lizenzen pro Nutzer

Cloud-Suite aus Office-Apps und Diensten – als Tenant pro Organisation.

Tenant & Admin Center

- Der Tenant ist die Organisation in der Cloud
- Verwaltung über die Admin Center
- Rollen mit minimalen Rechten vergeben
- Mandanteneinstellungen zentral

Die Kerndienste

Exchange Online

E-Mail und Kalender.

SharePoint / OneDrive / Teams

Dokumente, Zusammenarbeit, Kommunikation.

- Teams sitzt auf SharePoint/Exchange auf
- OneDrive für persönliche Dateien

Identität & Sicherheit

MFA und Conditional Access sind die wichtigsten Schutzmaßnahmen.

- Entra ID verwaltet Konten und Anmeldung
- Conditional Access: Zugriff nach Bedingungen
- MFA möglichst für alle erzwingen

Betrieb

- Lizenzen und Zuweisungen im Blick
- Backup von M365-Daten mitdenken
- Compliance-/Aufbewahrungsrichtlinien
- Service Health überwachen

Rollen & Delegation

Nicht jeder braucht globalen Admin – Rechte fein und zeitlich begrenzt vergeben.

- Spezifische Admin-Rollen (Exchange, Teams, User)
- Privileged Identity Management: Rechte nur bei Bedarf (Just-in-Time)
- Notfall-Konten (Break-Glass) separat absichern

Datensicherheit & Compliance

- Data Loss Prevention (DLP) gegen Datenabfluss
- Vertraulichkeitskennzeichnungen (Sensitivity Labels)
- Aufbewahrungsrichtlinien und eDiscovery
- Alerts im Compliance-Portal

Lizenz-Praxis

- Pläne von Business bis E3/E5 je nach Bedarf
- Lizenzen über Gruppen zuweisen (gruppenbasiert)
- Ungenutzte Lizenzen erkennen und sparen
- Add-ons gezielt statt pauschal

Das Wichtigste

- M365 = Tenant + Cloud-Dienste
- Exchange/SharePoint/Teams/OneDrive
- Entra ID + MFA + Conditional Access
- Backup und Compliance nicht vergessen

Microsoft Azure – Kernprodukte

Die wichtigsten Bausteine der Azure-Cloud

Azure im Überblick

- Ressourcen in Regionen weltweit
- Abrechnung nach Verbrauch
- Enge Integration mit Microsoft 365/Entra

Microsofts Public Cloud für IaaS, PaaS und mehr.

Compute & Netzwerk

Azure VMs

Virtuelle Maschinen (IaaS).

VNet

Virtuelles Netzwerk mit Subnetzen und Regeln.

- NSGs als virtuelle Firewalls
- VPN/ExpressRoute zur Anbindung

Storage

- Blob Storage für Objekte/Dateien
- Managed Disks für VMs
- Tiers (Hot/Cool/Archive) für Kosten
- Redundanzoptionen (LRS/GRS)

Sicherheit & Betrieb

- Key Vault für Secrets/Zertifikate
- Azure Backup & Site Recovery (DR)
- Azure Monitor für Metriken/Logs
- RBAC für feingranulare Rechte

Kosten im Griff

Cloud-Kosten laufen ohne Kontrolle schnell davon – Cost Management nutzen.

- Budgets und Alarmer setzen
- Ungenutzte Ressourcen abschalten
- Reservierungen für Dauerlast

Ressourcen organisieren

- Management Groups → Subscriptions → Resource Groups
- Tags für Kostenstellen und Zuordnung
- Einheitliche Namenskonventionen

Struktur von oben nach unten schafft Übersicht und Governance.

Governance

Azure Policy erzwingt Regeln automatisch – z. B. erlaubte Regionen oder Pflicht-Tags.

- Policy und Blueprints für Vorgaben
- Resource Locks gegen versehentliches Löschen
- RBAC vererbt sich entlang der Hierarchie

Netzwerk-Sicherheit

- NSGs für Subnetze, Azure Firewall für zentrale Kontrolle
- Private Endpoints statt öffentlicher Dienste
- Azure Bastion statt RDP/SSH offen ins Internet
- DDoS-Schutz für exponierte Dienste

Das Wichtigste

- VMs + VNet als Basis
- Blob/Managed Disks für Speicher
- Key Vault, Backup, Monitor, RBAC
- Kosten aktiv steuern

AWS – Kernprodukte

Die zentralen Dienste von Amazon Web Services

AWS im Überblick

- Regionen und Availability Zones
- Pay-as-you-go
- Sehr großes Dienstangebot

Der größte Public-Cloud-Anbieter mit sehr breitem Portfolio.

Compute & Netzwerk

EC2

Virtuelle Server (IaaS).

VPC

Virtuelles Netzwerk mit Subnetzen und Security Groups.

- Auto Scaling für Elastizität
- Load Balancer für Verteilung

Storage & Datenbank

S3

Objektspeicher – extrem verbreitet.

RDS

Verwaltete relationale Datenbanken.

- S3-Storage-Klassen für Kosten
- Backups/Snapshots nutzen

Weitere Kerndienste

- IAM: Identitäten und Rechte
- Lambda: Serverless-Funktionen
- CloudWatch: Monitoring/Logs
- Route 53: DNS

Einstieg & Zertifizierung

- Well-Architected Framework als Leitlinie
- Zertifizierungspfade (Cloud Practitioner → Associate)
- Kosten mit Budgets/Alarmen kontrollieren
- Least-Privilege in IAM

Konten organisieren

- AWS Organizations bündelt mehrere Konten
- Service Control Policies (SCPs) als Leitplanken
- Getrennte Konten für Prod, Test, Sandbox
- Eine Landing Zone als sauberes Fundament

Netzwerk & Zugriff

IAM Roles statt langlebiger Access Keys – Rechte temporär und rollenbasiert.

- Public vs. Private Subnets, NAT Gateway für Ausgang
- Security Groups (zustandsbehaftet) vs. NACLs
- Least Privilege in IAM konsequent

Kosten & Audit

- Cost Explorer und Budgets mit Alarmen
- Durchgängiges Tagging für Kostenzuordnung
- CloudTrail protokolliert alle API-Aufrufe
- GuardDuty für Bedrohungserkennung

Das Wichtigste

- EC2 + VPC als Fundament
- S3 (Objekte) und RDS (DB)
- IAM/Lambda/CloudWatch/Route 53
- Kosten und Rechte diszipliniert managen

Google Cloud & Workspace

Googles Cloud-Angebote im Überblick

Zwei Welten

Google Workspace

Produktivität: Gmail, Docs, Drive, Meet.

Google Cloud (GCP)

Infrastruktur und Plattform wie Azure/AWS.

- Workspace ≈ Gegenstück zu Microsoft 365
- GCP ≈ Gegenstück zu Azure/AWS

Workspace vs. M365

- Beide: Mail, Dateien, Zusammenarbeit
- Workspace stark im Browser/kollaborativ
- Wahl oft strategisch/ökosystembedingt
- Migration zwischen beiden ist Aufwand

GCP-Kerndienste

- Compute Engine (VMs)
- Cloud Storage (Objekte)
- VPC-Netzwerke
- IAM für Rechte

Googles Stärken

Besonders stark bei Daten, KI/ML und Analytics.

- BigQuery für große Datenanalysen
- GKE (veraltetes Kubernetes)
- KI-/ML-Dienste out of the box

Betrieb & Kosten

- Projekte gliedern Ressourcen
- Budgets und Alarmer setzen
- Least Privilege in IAM
- Regionen/Compliance beachten

Ressourcen-Hierarchie

- Organization → Folders → Projects
- IAM-Rechte vererben sich nach unten
- Projekte trennen Umgebungen und Kosten
- Labels zur Zuordnung

Sicherheit

Service Accounts sind mächtig – sparsam und mit minimalen Rechten einsetzen.

- Least Privilege in IAM
- VPC Service Controls gegen Datenabfluss
- Kundenverwaltete Schlüssel für sensible Daten

Wann Google?

- Starke Daten- und ML-Werkzeuge (BigQuery, Vertex AI)
- Nahtlose Kollaboration über Workspace
- Anthos für Hybrid- und Multi-Cloud
- Ökosystem-Entscheidung mitdenken

Das Wichtigste

- Workspace (Produktivität) vs. GCP (Infra)
- Workspace als M365-Alternative
- GKE/BigQuery als GCP-Stärken
- Google glänzt bei Daten & KI

Cloud-Migration

Von On-Premises in die Cloud

Warum migrieren?

- Weg von Investitionen hin zu laufenden Kosten
- Schnellere Bereitstellung
- Aber: nicht alles gehört in die Cloud

Skalierbarkeit, Flexibilität und weniger eigene Hardware.

Die Migrationsstrategien

Lift & Shift

1:1 verschieben – schnell, wenig Optimierung.

Replatform / Refactor

Anpassen bzw. neu bauen – mehr Nutzen, mehr Aufwand.

- Oft als „6 R“ bekannt (u. a. Rehost, Retire)
- Pro Anwendung entscheiden

Phasen

- Assessment: Bestand und Abhängigkeiten erfassen
- Planung: Reihenfolge und Zielarchitektur
- Migration: schrittweise umsetzen
- Optimierung: Kosten und Betrieb feinjustieren

Werkzeuge

- Azure Migrate / AWS Migration Services
- Abhängigkeiten automatisiert erkennen
- Testmigrationen vor dem Umzug
- Rückfallplan bereithalten

Häufige Fehler

Reines Lift & Shift ohne Optimierung führt oft zu hohen Kosten.

- Kosten unterschätzt
- Abhängigkeiten übersehen
- Sicherheit/Compliance zu spät bedacht
- Kein Rollback geplant

Landing Zone zuerst

Vor der ersten Migration steht das Fundament: Konten, Netzwerk, Identität, Guardrails.

- Identität und Zugriff (SSO) einrichten
- Netzwerk-Grundstruktur und Sicherheit
- Governance und Kostenkontrolle von Anfang an

Datenmigration

- Online (über die Leitung) vs. offline (Transfer-Appliance)
- Bandbreite und Zeitfenster realistisch planen
- Downtime minimieren (Sync + kurzer Umschalt)
- Integrität nach der Übertragung prüfen

Nach der Migration

- Rightsizing – überdimensionierte Ressourcen verkleinern
- Monitoring und Alarmer etablieren
- Kosten kontinuierlich optimieren (FinOps)
- Alte On-Prem-Umgebung geordnet abbauen

Das Wichtigste

- Strategie je Anwendung (6 R)
- Assessment → Planung → Migration → Optimierung
- Migrationswerkzeuge nutzen
- Kosten und Rückweg von Anfang an mitdenken

Infrastructure as Code

Infrastruktur automatisiert und reproduzierbar

Was ist IaC?

Der Code ist die einzige Wahrheit – reproduzierbar, versionierbar, überprüfbar.

- Gleiche Umgebung immer wieder aufbaubar
- Änderungen nachvollziehbar (Git)
- Weniger manuelle Fehler

Infrastruktur wird als Code beschrieben statt manuell geklickt.

Deklarativ vs. imperativ

Deklarativ

Beschreibt den Zielzustand (Terraform, Bicep).

Imperativ

Beschreibt die Schritte dorthin (klassische Skripte).

- Deklarativ ist bei IaC der Regelfall
- Werkzeug gleicht Ist- und Sollzustand ab

Die Werkzeuge

- Terraform: cloud-übergreifend
- Bicep/ARM: Azure-nativ
- CloudFormation: AWS-nativ
- Ansible: Konfiguration/Provisionierung

Ansible-Besonderheit

- Fokus auf Konfigurationsmanagement
- Agentenlos über SSH/WinRM
- Playbooks in YAML
- Gut für Server-Setup und Software

GitOps

Der gewünschte Zustand liegt in Git; Änderungen laufen über Commits und Pipelines.

- Nachvollziehbare, geprüfte Änderungen
- State/Secrets sicher verwalten
- Erst planen (plan), dann anwenden (apply)

State & Module

- State remote speichern und sperren (Locking)
- Wiederverwendbare Module gegen Copy-Paste
- Variablen und Umgebungen sauber trennen

Der State bildet ab, was tatsächlich existiert – er ist kritisch.

In die Pipeline

plan im Pull-Request zeigt die Änderung; apply erst nach Freigabe.

- Automatische Prüfung und Vorschau vor dem Anwenden
- Policy-as-Code prüft Regeln automatisch
- Nur geprüfter Code verändert die Infrastruktur

Gute Praxis

- Kleine, nachvollziehbare Änderungen
- Drift vermeiden – nichts von Hand nachklicken
- Secrets aus einem Tresor beziehen
- State und Secrets niemals ins Repository

Das Wichtigste

- IaC = Infrastruktur als versionierter Code
- Deklarativ (Terraform/Bicep/CloudFormation)
- Ansible fürs Konfigurationsmanagement
- GitOps: Git als Quelle der Wahrheit

Cloud-Security & FinOps

Sicher und kosteneffizient in der Cloud

Geteilte Verantwortung

Der Anbieter sichert die Cloud, DU sicherst, was DU in der Cloud betreibst.

- Konfiguration ist Kundensache
- Fehlkonfiguration ist Hauptrisiko
- Klar abgrenzen, wer wofür haftet

IAM-Best-Practices

- Least Privilege konsequent
- MFA für alle, besonders Admins
- Keine Dauer-Root-/Owner-Nutzung
- Rechte regelmäßig überprüfen

Cloud-Security-Werkzeuge

CSPM

Erkennt Fehlkonfigurationen in der Cloud.

CASB

Kontrolliert Nutzung von Cloud-Diensten.

- Verschlüsselung in Ruhe und Transport
- Logging/Monitoring aktivieren

Was ist FinOps?

- Kosten sichtbar machen (Tags/Reports)
- Verantwortung in die Teams geben
- Kontinuierlich optimieren

Kostenkultur für die Cloud – Transparenz und Verantwortung.

Kosten senken

- Ungenutzte Ressourcen abschalten
- Rightsizing (passende Größe wählen)
- Reserved Instances / Savings Plans für Dauerlast
- Storage-Tiers und Autoscaling nutzen

Sichtbarkeit & Erkennung

- Zentrale Log-Sammlung über alle Konten
- Cloud-native Security-Center nutzen
- CSPM-Findings nach Risiko priorisieren
- Alarme in Ticket/Chat routen

Datenschutz in der Cloud

Verschlüsselung ist Standard – entscheidend ist die Kontrolle über die Schlüssel.

- Datenstandort und Rechtsraum beachten
- Schlüsselverwaltung über KMS/HSM
- Zugriffe protokollieren und auswerten

FinOps in der Praxis

- Kosten je Team sichtbar machen (Showback/Chargeback)
- Anomalie-Alarme bei Kostensprüngen
- Regelmäßige Optimierungs-Reviews
- Ungenutztes konsequent abschalten

Das Wichtigste

- Shared Responsibility verstehen
- Fehlkonfiguration = Hauptrisiko → CSPM/IAM
- FinOps macht Kosten transparent
- Rightsizing + Reservierungen sparen viel



LB-09 · Monitoring & Management

6 Kapitel

Monitoring-Grundlagen

Systeme im Blick behalten, bevor Nutzer klagen

Warum Monitoring?

- Verfügbarkeit und Leistung sichern
- Trends und Kapazität planen
- Fehler schneller eingrenzen

Probleme erkennen, bevor sie zum Ausfall werden.

Monitoring vs. Observability

Monitoring

Bekannte Metriken überwachen („ist X ok?“).

Observability

Aus Daten unbekannte Probleme verstehen („warum?“).

- Monitoring beantwortet erwartete Fragen
- Observability hilft bei Unerwartetem

Die drei Säulen

Metriken, Logs und Traces ergeben zusammen das Gesamtbild.

- Metriken: Zahlen über die Zeit
- Logs: Ereignisse im Detail
- Traces: Weg einer Anfrage durch Systeme

Was überwachen?

- CPU, RAM, Datenträger, Netzwerk
- Dienste und Anwendungen
- Zertifikate und Ablaufdaten
- Endnutzer-Erlebnis (synthetisch)

Alert-Fatigue vermeiden

Zu viele Alarme sind so schädlich wie zu wenige – wichtige gehen unter.

- Sinnvolle Schwellwerte statt Rauschen
- Alarme priorisieren und gruppieren
- Eskalationswege definieren
- Regelmäßig aufräumen

Schwellwerte & Baselines

- Statische Schwellwerte für klare Grenzen
- Dynamische Baselines lernen das Normalverhalten
- Saisonale Muster berücksichtigen (Tag/Nacht, Monatsende)
- Zu enge Schwellen erzeugen Rauschen

SLA, SLO & Fehlerbudget

Ein Fehlerbudget macht Verfügbarkeit messbar: erlaubte Ausfallzeit vor Konsequenzen.

- SLA ist die Zusage, SLO das interne Ziel
- Verfügbarkeit sauber messen und berichten
- Fehlerbudget steuert das Tempo von Änderungen

Dashboards & Berichte

- Rollengerechte Sichten (Technik vs. Management)
- Trends für die Kapazitätsplanung
- Klar statt überladen – die wichtigen Signale zuerst

Das Wichtigste

- Monitoring = proaktiv statt reaktiv
- Metriken + Logs + Traces
- Das Richtige überwachen, nicht alles
- Alarmqualität schlägt Alarmmenge

Monitoring-Tools Open Source

Bewährte freie Werkzeuge

Warum Open Source?

- Keine Lizenzgebühren
- Große Communities
- Betriebsaufwand einkalkulieren

Mächtig, flexibel und ohne Lizenzkosten – dafür Betrieb selbst.

Prometheus & Grafana

Prometheus sammelt Metriken, Grafana visualisiert sie.

- Pull-Modell mit Exportern
- Alerting über Alertmanager
- Grafana für Dashboards
- Standard im Cloud-Native-Umfeld

Klassische Allrounder

Zabbix / Checkmk

Umfassendes Infrastruktur-Monitoring.

Nagios / Icinga

Bewährter Klassiker für Verfügbarkeit.

- Agenten und agentenlose Checks
- Viele fertige Vorlagen

Logs

- ELK-Stack (Elasticsearch, Logstash, Kibana)
- Graylog als Alternative
- Zentrale Logsammlung und Suche
- Speicher/Retention einplanen

Auswahl

- Metriken vs. Logs vs. beides klären
- Aufwand für Betrieb bewerten
- Skalierbarkeit prüfen
- Mit kleinem Setup starten

Pull, Push & Datenmodell

- Exporter stellen Metriken bereit, Prometheus holt sie ab
- Push Gateway für kurzlebige Batch-Jobs
- Labels machen Metriken auswertbar

Prometheus arbeitet nach dem Pull-Modell mit Zeitreihen.

Skalierung & Speicher

- Aufbewahrung (Retention) bewusst festlegen
- Langzeit-/Remote-Storage (z. B. Thanos, Mimir)
- Hochverfügbarkeit durch Redundanz
- Speicherbedarf wächst schnell – planen

Alerting-Praxis

Der Alertmanager gruppiert, unterdrückt und routet Alarmer gezielt.

- Gruppierung verhindert Alarm-Fluten
- Silences für geplante Wartung
- Routing an Chat, Ticket oder On-Call

Das Wichtigste

- Prometheus+Grafana für Metriken
- Zabbix/Checkmk/Icinga als Allrounder
- ELK/Graylog für Logs
- Kostenfrei, aber Betrieb selbst

Monitoring-Tools Commercial

Kommerzielle Lösungen mit Support

Warum kommerziell?

- Schnellere Einführung
- Herstellersupport und SLAs
- Dafür laufende Lizenzkosten

Weniger Eigenbetrieb, mehr fertige Funktionen und Support.

Bekannte Produkte

PRTG

Einfaches, sensorbasiertes Monitoring (v. a. KMU).

Datadog / New Relic

Cloud-SaaS, stark bei Observability/APM.

- SolarWinds als breite Suite
- Preis oft nach Sensoren/Hosts/Datenvolumen

Cloud-nativ

- Azure Monitor für Azure-Ressourcen
- AWS CloudWatch für AWS
- Tief in die jeweilige Cloud integriert
- Kosten nach Metriken/Logs beachten

APM & Observability

APM zeigt, wo in einer Anwendung Zeit verloren geht – nicht nur ob der Server läuft.

- Traces über verteilte Systeme
- Fehler- und Performance-Analyse
- Gut für komplexe Anwendungen

Auswahlkriterien

- On-Prem vs. SaaS
- Preismodell und Skalierung
- Integrationen (Cloud, Ticketing)
- Datenschutz/Datenstandort

Agent vs. agentenlos

Mit Agent

Mehr Tiefe: Prozesse, Logs, Anwendungsmetriken.

Agentenlos

Breite Abdeckung über SNMP oder APIs.

- Oft eine Mischung aus beidem
- Agent-Ausrollung und -Pflege einplanen

Integrationen

- Anbindung an Ticketsysteme (ITSM)
- ChatOps für Alarme im Team-Chat
- On-Call und Eskalation (z. B. PagerDuty, Opsgenie)
- Automatische Reaktionen (Runbooks)

Kosten & Datenhaltung

- Preis oft nach Hosts oder Datenvolumen
- Datenstandort und Datenschutz prüfen
- Aufbewahrungsdauer beeinflusst die Kosten
- Vendor-Lock-in bedenken

Das Wichtigste

- Kommerziell = weniger Betrieb, mehr Kosten
- PRTG (KMU), Datadog/New Relic (Observability)
- Cloud-nativ: Azure Monitor / CloudWatch
- APM für Anwendungs-Tiefenblick

RMM-Systeme

Remote Monitoring & Management

Was ist RMM?

- Agent auf jedem verwalteten Gerät
- Überwachen, patchen, fernsteuern
- Vor allem bei MSPs verbreitet

Zentrale Fernüberwachung und -verwaltung vieler Endpunkte.

Kernfunktionen

- Monitoring von Zustand und Leistung
- Patch-Management zentral
- Fernzugriff/-wartung
- Automatisierungen und Skripte ausrollen

Bekannte Anbieter

NinjaOne / Datto RMM

Moderne, verbreitete Plattformen.

N-able / Atera / Syncro

Weitere gängige Lösungen.

- ConnectWise im MSP-Umfeld
- ManageEngine (Endpoint Central/RMM Central): stark bei Firmen, die schon andere ManageEngine-Tools einsetzen
- Auswahl nach Größe/Integrationen

Warum RMM statt WSUS/Intune/MDM?

WSUS und GPOs brauchen Domänen-/LAN-Kontakt, Intune und Apple-MDM brauchen volles Enrollment samt Lizenz. Bei verteilten Standorten und viel Homeoffice greift das zu oft nicht zuverlässig.

- Ein Agent patcht Windows, macOS, Linux **und** Drittanbieter-Software – WSUS kann nur Microsoft
- Funktioniert über Internet, unabhängig von Domänen-Mitgliedschaft oder Site-Topologie
- Für KMU oft günstiger und schneller aufgesetzt als volles Intune + Autopilot + Entra
- Ein Werkzeug für Monitoring, Patching und Fernzugriff statt mehrerer getrennter Silos

Kein Ersatz, sondern oft die praktischere Ergänzung – oder Alternative.

MSP vs. interne IT

MSP

Viele Kunden/Mandanten zentral betreuen.

Interne IT

Eigene Geräteflotte effizient verwalten.

- Mandantenfähigkeit ist für MSPs zentral
- Auch intern großer Effizienzgewinn

Sicherheit & Grenzen

RMM ist mächtig – wird es kompromittiert, sind alle Endpunkte gefährdet. Ein bekanntes, hochwertiges Angriffsziel.

- MFA und starke Zugänge zwingend
- Least Privilege für Techniker
- Alle Aktivitäten protokollieren
- Nur geprüfte Skripte ausrollen

Onboarding & Automatisierung

- Agent per Massen-Rollout ausbringen
- Skript-Bibliothek für wiederkehrende Aufgaben
- Automatisierungs-Policies (self-healing)
- Gerätegruppen für gezielte Aktionen

Patch- & Software-Management

Patches in Ringen und Zeitfenstern ausrollen – nicht alles auf einmal.

- Betriebssystem- und Drittanbieter-Updates
- Test- vor Produktivring
- Reporting über den Patch-Stand

Das Wichtigste

- RMM = Fernüberwachung + -verwaltung per Agent
- Monitoring, Patching, Fernzugriff, Automatisierung
- NinjaOne/Datto/ConnectWise/ManageEngine u. a.
- Praktikable Antwort auf verteilte Standorte & Homeoffice, wo WSUS/GPO/MDM an Grenzen stoßen
- Streng absichern (MFA, Least Privilege)

ITSM & Ticketsysteme

IT-Service-Management mit System

Was ist ITSM?

- Nachvollziehbare Bearbeitung
- Messbare Servicequalität
- ITIL als bekanntes Rahmenwerk

Der geordnete Betrieb von IT-Services – Prozesse statt Zuruf.

Incident vs. Problem

Incident

Störung schnell beheben (Service wieder herstellen).

Problem

Ursache dauerhaft beseitigen.

- Erst Betrieb sichern, dann Ursache lösen
- Bekannte Fehler dokumentieren

Change Management

Änderungen geplant, geprüft und mit Rückfallplan – nicht spontan auf Produktion.

- Risiko und Auswirkung bewerten
- Freigaben und Wartungsfenster
- Dokumentation der Änderung

Ticketsysteme

- TANSS, Halo, Freshdesk
- Jira Service Management
- SLAs und Prioritäten abbilden
- Wissensdatenbank integrieren

Gute Praxis

- Klare Kategorien und Prioritäten
- SLAs realistisch definieren
- Self-Service für Standardanfragen
- Kennzahlen auswerten und verbessern

Weitere ITIL-Bausteine

- Service Request vs. Incident sauber trennen
- Service-Katalog und Self-Service-Portal
- CMDB: Überblick über Assets und Beziehungen
- Wissensdatenbank für bekannte Lösungen

SLAs & Priorisierung

Priorität ergibt sich aus Auswirkung mal Dringlichkeit.

- Reaktions- und Lösungszeiten je Priorität
- Eskalationswege klar definieren
- Realistische, messbare SLAs

Kennzahlen

- Erstlösungsquote und MTTR
- Backlog und Wiedereröffnungsrate
- Kundenzufriedenheit (CSAT)
- Kennzahlen für kontinuierliche Verbesserung nutzen

Das Wichtigste

- ITSM ordnet den IT-Betrieb (ITIL)
- Incident = beheben, Problem = Ursache
- Changes geplant + mit Rollback
- Ticketsystem + SLAs + Wissensbasis

IT-Dokumentation

Was, womit und warum dokumentieren

Warum dokumentieren?

- Schnellere Wiederherstellung bei Ausfällen
- Unabhängigkeit von einzelnen Köpfen
- Unbezahlbar für die Einarbeitung

Im Ernstfall fehlt die Zeit, alles neu herauszufinden.

Was gehört hinein?

- Netzwerkpläne und IP-Konzepte
- Serverlandschaft und Abhängigkeiten
- Notfallkontakte und Wiederanlaufpläne
- Zugänge – aber sicher, nicht in Excel

Werkzeuge

IT Glue / Hudu

Auf IT-Dokumentation zugeschnitten (v. a. MSP).

Confluence

Allgemeiner für Wissen und Wikis.

- Passwörter in einen Manager (Bitwarden, 1Password)

Aktualität

Eine veraltete Dokumentation ist schlimmer als gar keine – weil man sich auf sie verlässt.

- Beim Ändern direkt mitpflegen
- Regelmäßig überprüfen
- So knapp wie möglich, so ausführlich wie nötig

Doku absichern

- Enthält sensible Infos über die ganze Infrastruktur
- Zugriff nur für Befugte
- Verschlüsselt speichern
- Ins Backup einbeziehen

Netzplan & Inventar

Ein aktueller Netzplan spart im Störfall die meiste Zeit – er zeigt sofort, was mit was zusammenhängt.

- Netzplan zeigt Geräte, Verbindungen und IP-Bereiche
- Inventar (CMDB) listet Hard- und Software mit Standort und Besitzer
- Lizenzen und Wartungsverträge mitführen

Passwörter & Geheimnisse

- Passwort-Manager oder Vault mit Zugriffsprotokoll
- Vier-Augen-Prinzip für kritische Konten
- Notfall-Zugänge versiegelt und getrennt aufbewahren

Zugangsdaten gehören nicht in Textdateien oder Wikis.

Runbooks & Notfall-Handbuch

Runbook

Schritt-für-Schritt-Anleitung für wiederkehrende Aufgaben.

Notfall-Handbuch

Wer tut was bei Ausfall? Offline verfügbar halten.

- Wiederherstellungs-Schritte regelmäßig durchspielen
- Kontaktlisten und Eskalationswege aktuell halten

Das Wichtigste

- Dokumentation spart im Ernstfall enorm Zeit
- Netzpläne, Configs, Notfallpläne, Passwörter (im Manager)
- Aktuell halten ist entscheidend
- Sensibel behandeln und absichern



LB-10 · Scripting & Automatisierung

7 Kapitel

PowerShell – Grundlagen

Sprachbausteine: Variablen, Logik, Schleifen, Funktionen

PowerShell als Sprache

- Skripte als `.ps1`-Dateien
- Objekt-Pipeline statt reinem Text
- Auf Windows Standard, plattformübergreifend als PowerShell 7

Mehr als eine Shell – eine vollwertige Skriptsprache mit Objekten.

Variablen & Datentypen

Variablen beginnen mit `$`: `$name = 'Server01'`

- Zahlen, Strings, Arrays, Hashtables
- Typisierung optional: `[int]$anzahl = 5`
- Automatische Variablen wie `$_` in der Pipeline

Bedingungen

```
if ($x -gt 10) { ... } elseif ($x -eq 10) { ... } else { ... }
```

- Vergleiche: `-eq`, `-ne`, `-gt`, `-lt`, `-like`
- `switch` für viele Fälle
- Logik mit `-and`, `-or`, `-not`

Verzweigungen steuern den Ablauf.

Schleifen

- `foreach ($i in $liste) { ... }`
- `for ($i=0; $i -lt 10; $i++) { ... }`
- `while` / `do..while` für Bedingungsschleifen
- `ForEach-Object` in der Pipeline

Funktionen

```
function Get-Status { param([string]$Name) ... }
```

- Parameter mit `param(...)`
- Rückgabe über die Ausgabe (kein zwingendes `return`)
- Sprechende Verb-Substantiv-Namen wählen

Wiederverwendbare Bausteine mit Parametern.

Fehlerbehandlung

```
try { ... } catch { ... } finally { ... }
```

- `-ErrorAction Stop` macht Fehler abfangbar
- `$Error` enthält die letzten Fehler
- Aussagekräftige Meldungen ausgeben

Skripte ausführen

- `Get-ExecutionPolicy` / `Set-ExecutionPolicy`
- Signierte Skripte in sensiblen Umgebungen
- Erst mit `-WhatIf` testen, wo möglich

Die Ausführungsrichtlinie schützt vor ungewollten Skripten.

Nützliche Cmdlets

- `Get-Help` und `Get-Command` zum Nachschlagen
- `Get-Member` zeigt Eigenschaften/Methoden
- `Get-Content` / `Set-Content` für Dateien
- `Test-Path` prüft, ob etwas existiert

Skript-Bausteine

- Kommentare mit `#` oder Block `<# ... #>`
- Parameter typisiert im `param()`-Block
- Splatting für viele Parameter übersichtlich
- Ausgabe ist die Rückgabe – kein zwingendes `return`

Das Wichtigste

- PowerShell ist eine objektorientierte Skriptsprache
- Variablen, Bedingungen, Schleifen, Funktionen als Basis
- `try/catch` für robuste Skripte
- `ExecutionPolicy` und `-WhatIf` für Sicherheit

PowerShell – Praxis für Admins

Automatisierung im echten Admin-Alltag

Vom Skript zur Automatisierung

- Massenoperationen statt Einzelclicks
- Reproduzierbar und dokumentierbar
- Als geplante Aufgabe automatisierbar

PowerShell glänzt bei wiederkehrenden Verwaltungsaufgaben.

Active Directory automatisieren

Das ActiveDirectory -Modul steuert Benutzer, Gruppen und Computer per Skript.

- `Get-ADUser` , `New-ADUser` , `Set-ADUser`
- Massen-Anlage aus CSV
- Gruppenmitgliedschaften pflegen

CSV Import & Export

```
Import-Csv nutzer.csv | ForEach-Object { ... }
```

- `Export-Csv` für Berichte
- Kombinierbar mit `Select-Object`
- Für Excel: Modul `ImportExcel`

Der Standardweg für Ein- und Ausgabe von Tabellendaten.

Remoting (WinRM)

- `Enter-PSSession` für interaktive Fernsteuerung
- `Invoke-Command` für Befehle auf vielen Hosts
- WinRM muss aktiviert und abgesichert sein
- Ideal für Server-Flotten

Geplante Aufgaben

- Windows Task Scheduler per PowerShell konfigurieren
- `Register-ScheduledTask`
- Dienstkonto mit minimalen Rechten verwenden

Skripte zeitgesteuert ausführen.

Wichtige Module

ImportExcel

Excel-Dateien ohne installiertes Excel lesen/schreiben.

PSWindowsUpdate

Windows-Updates per Skript verwalten.

- Module aus der PowerShell Gallery installieren
- Nur vertrauenswürdige Module verwenden

Fehler & Logging

Mit Start-Transcript protokollierst du einen ganzen Lauf mit.

- `try/catch` plus `-ErrorAction Stop`
- `$ErrorActionPreference` global steuern
- Aussagekräftige Meldungen ausgeben

Geplant & signiert

- Skripte als geplante Aufgabe ausführen
- ExecutionPolicy passend setzen
- Skripte in sensiblen Umgebungen signieren
- Dienstkonto mit minimalen Rechten

Das Wichtigste

- AD-Verwaltung + CSV = mächtige Kombination
- Remoting skaliert auf viele Server
- Geplante Aufgaben für echte Automatisierung
- Nützliche Module gezielt einsetzen

Bash & Linux-Scripting

Robuste Shell-Skripte für Linux-Systeme

Bash-Grundlagen

Jedes Skript beginnt mit einer Shebang-Zeile: `#!/usr/bin/env bash`

- Ausführbar machen: `chmod +x skript.sh`
- Starten mit `./skript.sh`
- Kommentare mit `#`

Bash ist die Standard-Shell der meisten Linux-Systeme.

Variablen & Arrays

- Zuweisung ohne Leerzeichen: `name="Server01"`
- Nutzung: `$name` bzw. `${name}`
- Arrays: `liste=(a b c)`, Zugriff `${liste[0]}`
- Immer in doppelte Anführungszeichen setzen: `"$var"`

Bedingungen & Schleifen

```
if [ "$x" -gt 10 ]; then ... fi · for f in *.log; do ... done
```

- Test mit `[ ... ]` bzw. `[[ ... ]]`
- `while`-Schleifen für Bedingungen
- `case` für viele Fälle

Funktionen

```
backup() { tar -czf "$1" "$2"; }
```

- Argumente als `$1`, `$2`, ...
- `local` für lokale Variablen
- Rückgabewert über `return` (Exit-Code)

Wiederverwendbare Codeblöcke.

Textwerkzeuge

- `grep` – Zeilen filtern
- `sed` – ersetzen/bearbeiten
- `awk` – spaltenweise verarbeiten
- `cut`, `sort`, `uniq`, `wc`

Robuste Skripte

`set -euo pipefail` lässt Skripte bei Fehlern früh und laut scheitern.

- `-e` bricht bei Fehler ab, `-u` meldet leere Variablen
- Logging mit Zeitstempel
- Rückgabecodes prüfen (`$?`)

Nützliche Werkzeuge

- `find` mit `xargs` für Massenaktionen
- `tee` schreibt in Datei und Ausgabe zugleich
- Exit-Codes prüfen: `befehl && ok || fehler`
- `watch` für wiederholte Beobachtung

Argumente & Sicherheit

Immer quoten – und niemals `eval` auf ungeprüfte Eingaben.

- Argumente über `$1`, `$@`, `getopts`
- `read` für interaktive Eingaben
- Standardwerte mit `${VAR:-default}`

Das Wichtigste

- Shebang + ausführbar machen
- Variablen immer quoten
- `sed/awk/grep` für Textverarbeitung
- `set -euo pipefail` für Robustheit

Cron & Task-Scheduling

Aufgaben zeitgesteuert und zuverlässig ausführen

Warum Scheduling?

- Backups, Bereinigungen, Reports, Updates
- Läuft auch ohne angemeldeten Nutzer
- Nachvollziehbar und zuverlässig

Wiederkehrende Aufgaben automatisch statt manuell ausführen.

Cron-Syntax

Fünf Felder: Minute · Stunde · Tag(Monat) · Monat · Wochentag

- `0 2 * * *` = täglich 02:00 Uhr
- `*/15 * * * *` = alle 15 Minuten
- `0 3 * * 1` = montags 03:00 Uhr

Crontab verwalten

- `crontab -e` bearbeitet die eigene Crontab
- `crontab -l` listet Einträge
- System-Crontab unter `/etc/cron.*`
- Ausgaben in eine Logdatei umleiten

Umgebungsvariablen in cron

Cron läuft mit minimaler Umgebung – `$PATH` ist oft anders als in der Shell.

- Immer absolute Pfade verwenden
- Benötigte Variablen im Skript oder in der Crontab setzen
- Häufigste Fehlerquelle bei „läuft manuell, aber nicht per cron“

Windows Task Scheduler

- `Register-ScheduledTask` zum Anlegen
- Trigger (Zeit, Anmeldung, Ereignis)
- Dienstkonto mit minimalen Rechten

Das Windows-Pendant, gut per PowerShell steuerbar.

systemd-Timer

- Timer-Unit + Service-Unit
- Bessere Logs über journalctl
- OnCalendar= für Zeitpläne

Moderne Alternative zu cron unter Linux.

Cron-Spezialfälle

- @reboot für Start beim Booten
- System-Jobs in /etc/cron.d
- MAILTO für Fehlermeldungen
- Ausgabe in eine Logdatei umleiten

Zuverlässigkeit

flock verhindert, dass sich lange Jobs überlappen.

- Doppelte Läufe vermeiden
- Bei Fehler benachrichtigen
- Zeitzone des Systems beachten

Das Wichtigste

- Cron: fünf Felder, absolute Pfade
- cron-Umgebung ist minimal – daran scheitern viele Jobs
- Windows: Task Scheduler per PowerShell
- systemd-Timer als moderne Linux-Option

Python für IT-Admins

Wenn Shell-Skripte an ihre Grenzen stoßen

Warum Python?

- Plattformübergreifend (Windows, Linux, macOS)
- Ideal für komplexere Logik als Bash/PowerShell
- Große Auswahl an Modulen für fast alles

Lesbar, mächtig und mit riesigem Ökosystem an Bibliotheken.

Grundsyntax

Einrückung ist Teil der Syntax – Blöcke werden durch Einrücken gebildet.

- Variablen ohne Typdeklaration
- `if/elif/else`, `for`, `while`
- Funktionen mit `def`

Umgebungen & Pakete

- Virtuelle Umgebung: `python -m venv .venv`
- Pakete mit `pip install`
- `requirements.txt` für Reproduzierbarkeit

Saubere Trennung von Projektabhängigkeiten.

Nützliche Bibliotheken

[requests](#) / [paramiko](#)

HTTP-APIs ansprechen · SSH-Automatisierung.

[pandas](#) / [psutil](#)

Daten auswerten · System- und Prozess-Infos.

- `openpyxl` für Excel
- `subprocess` für Systembefehle

Praxisbeispiele

- Logdateien auswerten und Berichte erzeugen
- REST-APIs abfragen (Monitoring, Ticketsysteme)
- Server per SSH konfigurieren (paramiko)
- Wiederkehrende Reports als CSV/Excel

Gute Praxis

Keine Zugangsdaten im Code – aus Umgebungsvariablen oder einem Tresor lesen.

- Fehler mit `try/except` abfangen
- Skripte klein und testbar halten
- Mit Versionskontrolle (Git) arbeiten

Struktur & Umgebungen

- Virtuelle Umgebung mit `venv`
- Abhängigkeiten in `requirements.txt`
- Module und Imports sauber trennen
- `if __name__ == "__main__":` als Einstieg

Robuste Skripte

- Fehler mit `try/except` abfangen
- Das `logging`-Modul statt `print`
- `argparse` für Kommandozeilen-Argumente
- Type Hints für Lesbarkeit

Das Wichtigste

- Python für komplexere Automatisierung
- `venv` + `pip` + `requirements.txt`
- `requests/paramiko/pandas/psutil` als Werkzeugkasten
- Secrets nie hart codieren

Git & Versionskontrolle

Änderungen nachvollziehbar verwalten – auch für Configs

Warum Versionskontrolle?

- Historie und Nachvollziehbarkeit
- Zusammenarbeit ohne Datei-Chaos
- Auch für Skripte und Konfigurationen wertvoll

Git protokolliert, wer wann was geändert hat – und macht Änderungen rückgängig machbar.

Der Grund-Workflow

```
git add → git commit → git push
```

- `git init` / `git clone` zum Start
- `git status` zeigt den Zustand
- `git pull` holt Änderungen anderer
- Aussagekräftige Commit-Nachrichten

Branches

- `git branch` / `git switch`
- Feature in eigenem Branch entwickeln
- Später in den Hauptzweig zusammenführen

Parallele Entwicklungslinien ohne sich zu stören.

Merge vs. Rebase

Merge

Führt Branches zusammen, behält die Historie beider.

Rebase

Setzt Commits neu auf – lineare, aufgeräumte Historie.

- Merge ist sicherer für Einsteiger
- Rebase nicht auf bereits geteilten Branches

Plattformen & .gitignore

- GitHub, GitLab, Azure DevOps als Hosting
- `.gitignore` hält Unnötiges/Geheimes draußen
- Niemals Secrets committen
- Repositories sinnvoll strukturieren

Git für Konfigurationen

Configs unter Git = Änderungen dokumentiert und rückrollbar.

- nginx-, systemd-, Ansible-Dateien versionieren
- Vor riskanten Änderungen committen
- Bei Problemen gezielt zurückgehen

Konflikte & Rückgängig

`git revert` macht einen Commit sauber rückgängig – ohne die Historie umzuschreiben.

- Merge-Konflikte manuell auflösen
- `git reset` nur lokal/mit Bedacht
- `git stash` parkt Änderungen kurz

Zusammenarbeit

- Pull Requests für Review
- `.gitignore` hält Unnötiges/Geheimes draußen
- Tags für Releases
- Kleine, häufige Commits

Das Wichtigste

- `add` → `commit` → `push` als Kern
- Branches für paralleles Arbeiten
- Merge (sicher) vs. Rebase (aufgeräumt)
- Secrets via `.gitignore` fernhalten

CI/CD-Grundlagen

Automatisiert testen und ausrollen

Was ist CI/CD?

CI

Continuous Integration: Änderungen laufend zusammenführen und automatisch testen.

CD

Continuous Delivery/Deployment: automatisch bereitstellen bzw. ausrollen.

- Weniger manuelle Fehler
- Schnelleres, verlässlicheres Ausrollen

Die Pipeline

Typisch: Build → Test → (Staging) → Deploy

- Jede Phase kann die Pipeline stoppen
- Nur getesteter Code geht weiter
- Ergebnisse sind einsehbar und protokolliert

Ein automatisierter Ablauf, ausgelöst durch einen Git-Push.

GitHub Actions

- Workflows in `.github/workflows/`
- Ausgelöst durch Push, Pull Request oder Zeitplan
- Große Auswahl fertiger Actions

CI/CD direkt in GitHub, per YAML beschrieben.

Weitere Werkzeuge

- GitLab CI/CD (`.gitlab-ci.yml`)
- Azure DevOps Pipelines
- Jenkins im klassischen On-Prem-Umfeld
- Alle folgen demselben Grundprinzip

Anwendung: Deployment

- Push löst die Pipeline aus
- Automatischer Build und Test
- Bei Erfolg: Ausrollen auf den Zielserver
- Secrets sicher im CI-Tresor hinterlegen

Beispiel für einen einfachen Ablauf.

Gute Praxis

Erst nach Staging in Produktion – und immer mit einem Rückweg (Rollback).

- Kleine, häufige Änderungen statt großer Sprünge
- Pipelines schnell und aussagekräftig halten
- Zugangsdaten niemals im Repository

Pipeline in der Praxis

- Stages: Build → Test → Deploy
- Artefakte zwischen Stages weitergeben
- Caching beschleunigt Läufe
- Secrets nur aus dem CI-Tresor

Qualität & Sicherheit

Tests und Linting als Gate: schlägt etwas fehl, geht der Code nicht weiter.

- Automatische Tests bei jedem Push
- Abhängigkeits- und Sicherheits-Scans (SAST)
- Rollback- oder Blue-Green-Deployment

Das Wichtigste

- CI = testen, CD = ausrollen
- Pipeline: Build → Test → Deploy
- GitHub Actions / GitLab CI / Azure DevOps
- Staging + Rollback + geschützte Secrets



LB-13 · Ausbildung & Soft Skills

6 Kapitel

Kundenkommunikation

Als Fachinformatiker bist du auch Übersetzer zwischen Technik und Mensch

Technik allein reicht nicht

Ein perfekt gelöster technischer Fehler mit schlechter Kommunikation hinterlässt einen unzufriedenen Kunden. Eine suboptimale Lösung mit exzellenter Kommunikation führt oft zu mehr Zufriedenheit als das Gegenteil.

Beispiel: Buchhalterin Renate kann nicht drucken

- „Ihr Spool-Service hat sich aufgehängt, führen Sie `net stop spooler` aus, dann `net start spooler`.“
- Renate weiß nicht, was ein Spool-Service ist, und fühlt sich überfordert

Die richtige Antwort

„Ich schaue mir das sofort an. Ich sehe das Problem auf Ihrer Seite und löse es kurz remote – dauert zwei Minuten. Sie müssen nichts tun, ich gebe kurz Bescheid, wenn alles läuft.“

- Problem wird übernommen, keine Fachbegriffe
- Realistische Zeitangabe
- Kunde muss sich um nichts kümmern

Aktives Zuhören & Empathie

- Vollständig zuhören, bevor man antwortet – nicht nach dem ersten Satz schon die Lösung denken
- Der Kunde *hat* ein Problem – er *ist* nicht das Problem

Technisch übersetzen

Fachjargon

„Out of Memory Exception“

Kundengerecht

„Der Arbeitsspeicher ist voll“

Zeitangaben & Dokumentation

- Lieber 30 Minuten ansagen und in 15 fertig sein als umgekehrt
- Tickets vollständig dokumentieren: gemeldet, geprüft, gelöst, Dauer

Das Wichtigste

Ein Satz, den man nie sagen sollte: „Das ist eigentlich ganz einfach.“ Für den Kunden ist es nicht einfach – sonst hätte er nicht angerufen.

Projektarbeit & Zeitmanagement

Das IHK-Abschlussprojekt: 50 % der Abschlussnote, 35 Stunden Arbeitszeit

Das IHK-Abschlussprojekt

50 % der Abschlussnote. 35 Stunden Arbeitszeit, vollständige Dokumentation, Abnahme im Betrieb, Präsentation bei der IHK.

Wer kein Projektmanagement-Verständnis hat, läuft in Zeitprobleme.

Fehler 1: Zu breites Thema

Das Projekt muss realistisch in 35 Stunden abgeschlossen sein – nicht theoretisch, sondern tatsächlich.

35 Stunden klingen viel – sind es aber nicht, wenn das Thema schlecht abgegrenzt ist.

Fehler 2 & 3

- **Dokumentation am Ende schreiben:** zeitlich fast unmöglich, qualitativ meist schlechter als parallel dokumentiert
- **Keine Pufferzeit einplanen:** irgendetwas dauert immer länger als gedacht – ohne Puffer rutscht der ganze Zeitplan

Projektphasen nach Wasserfall

- Analyse
- Planung – inklusive Wirtschaftlichkeitsbetrachtung (Kosten/Nutzen)
- Durchführung
- Test
- Übergabe & Dokumentation

Die Eisenhower-Matrix

Dringend + wichtig

Sofort erledigen

Wichtig, nicht dringend

Terminieren und planen – z. B.
Prüfungsvorbereitung

Dringend, nicht wichtig

Möglichst delegieren

Weder noch

Weglassen

Das Berichtsheft

Wer täglich schreibt, weint einmal. Wer es vergisst, weint jeden Freitag.

IHK-Pflicht, täglich zu führen.

Das Wichtigste

Als angehender Fachinformatiker bist du Techniker, Kommunikator, Dokumentator und Projektmanager in einem. Die technischen Skills bringen dich in den Job – die Soft Skills halten dich dort und bringen dich weiter.

Teamarbeit & Konfliktmanagement

IT-Arbeit ist selten Einzelkämpfertum – wie Teams gut zusammenarbeiten und Konflikte lösen

Konflikte sind normal

Unterschiedliche Meinungen sind gesund – sie zeigen, dass beide Seiten mitdenken. Problematisch wird es erst, wenn aus der Sachebene eine persönliche Ebene wird.

Beispiel: Tool-Streit im Team

- Erst trennen: Geht es um das Tool – oder darum, wer recht behält?
- Kriterien vorher festlegen: Kosten, Lernaufwand, bestehende Integration
- Bei Stillstand: fester Zeitrahmen, danach entscheidet die Teamleitung

Zwei Kollegen sind sich uneinig, welches Monitoring-Tool eingeführt werden soll. Die Diskussion wird laut.

Feedback-Regeln

Vorwurf

„Du machst immer...“

Ich-Botschaft

„Mir ist aufgefallen, dass...“

Feedback zeitnah geben – nicht erst drei Wochen später in der Retro aufwärmen.

Standups & Retrospektiven

- Kurz, konkret, ergebnisorientiert
- Was lief gut, was lief schlecht, was ändern wir konkret
- Nicht nur Frust ablassen ohne Konsequenz

Das Wichtigste

Sachkonflikte sachlich lösen, mit klaren Kriterien statt Bauchgefühl. Echte Spannungen früh ansprechen, nicht aussitzen – ausgesessene Konflikte werden fast nie kleiner.

Präsentation & Fachgespräch

Teil 2 der IHK-Abschlussprüfung: das Projekt vorstellen und Rückfragen souverän beantworten

Der Ablauf

Rund 15 Minuten Präsentation, danach etwa 15–20 Minuten Fachgespräch mit dem Prüfungsausschuss.

Im Fachgespräch stellen die Prüfer Rückfragen – auch zu Alternativen, die nicht gewählt wurden.

Die Storyline der Präsentation

- **Ausgangslage** – was war das Problem
- **Lösung** – was wurde gebaut und warum genau so
- **Ergebnis** – hat es funktioniert, was wäre im Nachhinein anders

Folien gestalten

Der Vortrag lebt von dir, nicht vom Text auf der Leinwand.

Wenige Stichpunkte pro Folie – keine Textwüste.

Das Fachgespräch

Typische Frage: „Warum haben Sie sich für X entschieden und nicht für Y?“ Eine ehrliche Abwägung schlägt eine auswendig gelernte Antwort.

Die Prüfer wollen die Lösung nicht zerreißen – sie wollen verstehen, ob sie wirklich durchdacht ist.

Das Wichtigste

- Laut üben – vor Kollegen, Familie oder dem Spiegel
- Bei Nervosität: kurze Pause, tief durchatmen, weiterreden ist erlaubt

Lernstrategien & Prüfungsvorbereitung

Für Zwischenprüfung (Teil 1) und Abschlussprüfung – was wirklich hilft

Der größte Hebel: alte Prüfungen üben

Der IHK-Prüfungskatalog wiederholt sich in Struktur und Aufgabentypen stark von Jahr zu Jahr. Wer zehn alte Prüfungen durchgerechnet hat, kennt die Fallen, bevor er sie in der echten Prüfung sieht.

Spaced Repetition

- Das Gehirn merkt sich Dinge besser, die es mehrmals über Tage verteilt abrufen
- Ein Abruf am Stück durchgelesen bleibt kaum hängen

Karteikarten, die in wachsenden Abständen wiederholt werden, statt einmal stundenlang zu pauken.

Ein realistischer Lernplan

- Rückwärts vom Prüfungstermin planen – mit echtem Puffer
- Themen nach Prüfungsgewicht priorisieren, nicht alles gleich behandeln
- Kurze, fokussierte Lernblöcke (25–30 Min.) mit Pause schlagen stundenlanges Lesen

Gruppenlernen oder allein?

Allein

Wissen aufbauen, Grundlagen verstehen

In der Gruppe

Verständnislücken aufdecken – wer es erklären kann, hat es verstanden

Das Wichtigste

Alte Prüfungen rechnen schlägt Theorie lesen. Lernen in Etappen mit Wiederholung schlägt Pauken in letzter Minute.

Ausbildungsrecht & Berufseinstieg

Das juristische Grundgerüst rund um die Ausbildung – kurz und praxisnah

Die Probezeit

Laut Berufsbildungsgesetz mindestens 1, höchstens 4 Monate. In dieser Zeit können beide Seiten jederzeit ohne Frist und ohne Angabe von Gründen kündigen.

Kündigung nach der Probezeit

- Azubi: nur mit 4 Wochen Frist, nur bei Ausbildungsaufgabe oder Berufswechsel
- Betrieb: nur noch aus wichtigem Grund fristlos

Berufsschule & Jugendarbeitsschutz

- Berufsschulzeit ist Ausbildungszeit – Freistellung und Bezahlung durch den Betrieb
- Unter 18: Jugendarbeitsschutzgesetz greift – max. 8 Std./Tag, feste Pausenregelungen
- Ein Berufsschultag mit viel Unterricht zählt bereits als vollständiger Arbeitstag

Übernahme nach der Ausbildung

Deshalb: einige Monate vor Prüfungsende aktiv nach der Übernahme fragen – nicht einfach abwarten.

Einen generellen gesetzlichen Anspruch gibt es normalerweise nicht – außer er steht im Tarifvertrag oder in einer Betriebsvereinbarung.

Das Wichtigste

Am Ende steht dir ein Ausbildungszeugnis zu – auf Wunsch ein qualifiziertes mit Bewertung von Leistung und Verhalten. Das ist die Grundlage für jede Bewerbung danach.