17.5 Netzwerk- und Systembefehle in der Praxis

Administratoren nutzen Befehle, um Systeme, Netzwerke, Dienste, Verbindungen, IP-Konfigurationen und Fehlerzustände zu prüfen.

Typische Ziele sind:

```
- IP-Adresse prüfen
- Gateway prüfen
- DNS prüfen
- Erreichbarkeit testen
- Routing prüfen
- offene Ports prüfen
- aktive Verbindungen anzeigen
- Dienste prüfen
- Logs anzeigen
- Benutzerrechte prüfen
- Systemressourcen prüfen
```

Merksatz:

```
Befehle helfen,
Vermutungen durch konkrete Informationen zu ersetzen.
```

Warum Befehle wichtig sind

Grafische Oberflächen zeigen oft nur einen Teil der Informationen.

Befehle sind wichtig, weil sie:

```
schnell sind
genaue Ausgaben liefern
automatisierbar sind
auch per SSH funktionieren
bei Servern ohne GUI nutzbar sind
Fehlersuche systematisch machen
gut dokumentierbare Ergebnisse liefern
```

Merksatz:

> Kommandozeilenwerkzeuge sind praktische Diagnosewerkzeuge.

Befehle richtig einordnen

Ein einzelner Befehl löst selten das ganze Problem.

Wichtig ist die Reihenfolge:

> 1. IP-Konfiguration prüfen.
> 2. Gateway prüfen.
> 3. DNS prüfen.
> 4. Ziel erreichen.
> 5. Port prüfen.
> 6. Dienst prüfen.
> 7. Logs prüfen.
> 8. Änderungen prüfen.
> 9. Ursache dokumentieren.

Merksatz:

> Nicht blind Befehle ausführen,
> sondern systematisch prüfen.

Windows und Linux unterscheiden

Viele Befehle unterscheiden sich je nach Betriebssystem.

Beispiele:

Aufgabe	Windows	Linux/macOS
IP-Konfiguration anzeigen	ipconfig	ip addr oder ifconfig
Routing anzeigen	route print	ip route oder netstat -rn
DNS testen	nslookup	dig oder nslookup
Erreichbarkeit testen	ping	ping
Route verfolgen	tracert	traceroute
Verbindungen anzeigen	netstat	ss oder netstat
Dienste prüfen	services.msc, sc	systemctl

Aufgabe	Windows	Linux/macOS
Logs anzeigen	Ereignisanzeige	journalctl, Logdateien

Merksatz:

> Gleiche Aufgabe,
> anderes Werkzeug je nach System.

IP-Konfiguration prüfen

Die IP-Konfiguration zeigt, wie ein System im Netzwerk eingebunden ist.

Zu prüfen:

> IP-Adresse
> Subnetzmaske oder Präfix
> Standardgateway
> DNS-Server
> DHCP oder statische Adresse
> Netzwerkschnittstelle
> IPv4 und IPv6
> Verbindung aktiv oder getrennt

Merksatz:

> Ohne korrekte IP-Konfiguration funktioniert keine saubere Netzwerkkommunikation.

---

ipconfig unter Windows

Unter Windows zeigt ipconfig die IP-Konfiguration an.

Typische Informationen:

> IPv4-Adresse
> IPv6-Adresse
> Subnetzmaske
> Standardgateway
> DNS-Suffix

Adaptername

Typische Nutzung:

IP-Adresse prüfen
Gateway prüfen
DHCP-Zuweisung prüfen
DNS-Server prüfen

Merksatz:

ipconfig zeigt die Netzwerkkonfiguration unter Windows.

---

ipconfig /all

ipconfig /all zeigt detailliertere Informationen.

Zusätzlich sichtbar:

MAC-Adresse
DHCP aktiviert oder nicht
DHCP-Server
DNS-Server
Lease-Zeit
Hostname
DNS-Suffix
Adapterbeschreibung

Merksatz:

ipconfig /all ist genauer als einfaches ipconfig.

---

ip addr unter Linux

Unter Linux zeigt ip addr IP-Adressen und Schnittstellen an.

Typische Informationen:

```
Interface-Name
IPv4-Adresse
IPv6-Adresse
MAC-Adresse
Status der Schnittstelle
Präfixlänge
```

Beispiele für Interface-Namen:

```
eth0
ens18
enp0s3
wlan0
lo
```

Merksatz:

```
ip addr zeigt IP-Adressen und Netzwerkinterfaces unter Linux.
```

---

ifconfig

ifconfig ist ein älteres Werkzeug, das auf vielen Systemen noch bekannt ist.

Es zeigt ebenfalls Netzwerkschnittstellen und IP-Adressen.

Wichtig:

```
Auf modernen Linux-Systemen wird häufig ip aus dem iproute2-Paket bevorzugt.
```

Merksatz:

```
ifconfig ist bekannt,
ip ist auf modernen Linux-Systemen oft aktueller.
```

MAC-Adresse prüfen

Die MAC-Adresse ist die Hardwareadresse einer Netzwerkschnittstelle.

Sie ist wichtig für:

```
DHCP-Zuordnung
Switch-MAC-Tabellen
Wake-on-LAN
Netzwerkzugangskontrolle
Fehlersuche auf Schicht 2
```

Merksatz:

```
MAC-Adresse hilft bei Layer-2-Analyse.
```

Gateway prüfen

Das Standardgateway wird benötigt, wenn ein Ziel außerhalb des eigenen Netzes liegt.

Fehlt das Gateway, funktioniert meist nur Kommunikation im lokalen Netz.

Typische Symptome:

```
lokale Geräte erreichbar
Internet nicht erreichbar
andere Netze nicht erreichbar
DNS-Server außerhalb des Netzes nicht erreichbar
```

Merksatz:

```
Ohne Gateway kein Weg in andere Netze.
```

Routing anzeigen unter Windows

Unter Windows zeigt route print die Routing-Tabelle.

Zu prüfen:

```
Standardroute
Zielnetze
Gateway
```

```
Interface
Metrik
IPv4-Routen
IPv6-Routen
```

Merksatz:

```
route print zeigt,
wohin Windows Pakete senden will.
```

Routing anzeigen unter Linux

Unter Linux zeigt ip route die Routing-Tabelle.

Wichtig sind:

```
default route
lokale Netzroute
Gateway
Interface
spezielle Routen
Metrik
```

Merksatz:

```
ip route zeigt Routingentscheidungen unter Linux.
```

Standardroute

Die Standardroute wird genutzt, wenn keine spezifischere Route passt.

Sie wird oft angezeigt als:

```
0.0.0.0/0
```

oder:

```
default
```

Merksatz:

```
Standardroute ist der allgemeine Weg zu unbekannten Netzen.
```

Spezifischste Route gewinnt

Wenn mehrere Routen passen, wird die spezifischste Route verwendet.

Beispiel:

```
Route zu 192.168.1.0/24 ist spezifischer als 0.0.0.0/0.
```

Merksatz:

```
Die genaueste passende Route gewinnt.
```

---

ping

ping testet, ob ein Ziel per ICMP erreichbar ist.

Es prüft:

```
Ziel antwortet?
Paketlaufzeit?
Paketverlust?
Namensauflösung indirekt,
wenn ein Name genutzt wird
```

Merksatz:

```
ping prüft grundlegende Erreichbarkeit per ICMP.
```

---

ping richtig interpretieren

ping erfolgreich:

```
Ziel antwortet per ICMP.
```

ping fehlgeschlagen bedeutet nicht automatisch:

```
Ziel ist aus.
```

Mögliche Ursachen:

```
Firewall blockiert ICMP
Ziel blockiert ICMP
Routingproblem
DNS-Problem
Ziel offline
Gatewayproblem
Paketverlust
```

Merksatz:

```
Kein ping ist ein Hinweis,
aber kein endgültiger Beweis.
```

---

ping mit IP-Adresse

Ping auf IP-Adresse prüft vor allem:

```
Netzwerkweg
Routing
ICMP-Erreichbarkeit
```

Beispielprinzip:

```
ping 192.168.1.1
```

Wenn IP-Ping funktioniert, aber Name nicht, liegt der Fehler häufig bei DNS.

Merksatz:

```
IP-Ping trennt Netzwerkprüfung von Namensauflösung.
```

---

ping mit Hostname

Ping auf Hostname prüft zusätzlich DNS.

Beispielprinzip:

```
ping server.example.local
```

Wenn Hostname nicht aufgelöst wird, kann ping gar nicht zum eigentlichen Ziel starten.

Merksatz:

```
Hostname-Ping prüft DNS und Erreichbarkeit zusammen.
```

DNS prüfen

DNS-Probleme sind häufige Ursachen für Netzwerkfehler.

Zu prüfen:

```
richtiger DNS-Server eingetragen?
DNS-Server erreichbar?
Name existiert?
richtige IP wird geliefert?
interner oder externer DNS?
Split-DNS bei VPN?
DNS-Cache veraltet?
```

Merksatz:

```
Wenn IP funktioniert,
aber Name nicht,
ist DNS verdächtig.
```

---

nslookup

nslookup fragt DNS-Einträge ab.

Es kann genutzt werden für:

Name zu IP-Adresse
IP-Adresse zu Name
bestimmten DNS-Server testen
DNS-Fehler eingrenzen

Merksatz:

nslookup prüft Namensauflösung.

---

dig

dig ist ein häufiges DNS-Werkzeug unter Linux und macOS.

Es zeigt detaillierte DNS-Antworten.

Nützlich für:

A-Record
AAAA-Record
MX-Record
TXT-Record
CNAME
DNS-Serververgleich
Antwortzeiten
TTL

Merksatz:

dig zeigt DNS-Antworten sehr detailliert.

DNS-Recordtypen

Wichtige DNS-Recordtypen:

Record	Bedeutung
A	Name zu IPv4-Adresse
AAAA	Name zu IPv6-Adresse
CNAME	Alias auf anderen Namen

Record	Bedeutung
MX	Mailserver einer Domain
TXT	Textinformationen, oft SPF oder Verifikation
NS	zuständige Nameserver
PTR	Reverse DNS, IP zu Name

Merksatz:

> DNS besteht aus unterschiedlichen Recordtypen für unterschiedliche Aufgaben.

DNS-Cache

Betriebssysteme und Anwendungen können DNS-Antworten zwischenspeichern.

Vorteil:

> schnellere Namensauflösung

Nachteil:

> alte Einträge können Fehler verursachen.

Merksatz:

> DNS-Cache kann veraltete Antworten liefern.

---

tracert unter Windows

tracert zeigt den Weg zu einem Ziel über mehrere Router.

Es hilft bei:

> Routingproblemen
> Paketverlust unterwegs
> falschem Weg
> Abbruch an bestimmtem Hop
> Erkennung von Übergabepunkten

Merksatz:

```
tracert zeigt Zwischenstationen auf dem Weg zum Ziel.
```

---

traceroute unter Linux und macOS

traceroute erfüllt ähnlich wie tracert die Aufgabe, den Weg zum Ziel sichtbar zu machen.

Je nach System können unterschiedliche Protokolle genutzt werden.

Merksatz:

```
traceroute hilft,
den Netzwerkpfad einzugrenzen.
```

Traceroute richtig interpretieren

Ein Sternchen oder fehlende Antwort bedeutet nicht automatisch, dass dort ein Fehler ist.

Mögliche Ursachen:

```
Router antwortet nicht auf Traceroute
ICMP wird blockiert
Rate Limiting
Firewall-Regel
Paket geht trotzdem weiter
```

Wichtig ist:

```
Kommt das Ziel am Ende an?
Wo beginnt der Ausfall?
Sind mehrere Ziele betroffen?
```

Merksatz:

```
Traceroute-Ausgaben brauchen Interpretation.
```

---

netstat

netstat zeigt Netzwerkverbindungen, lauschende Ports und Routinginformationen.

Typische Nutzung:

```
aktive Verbindungen prüfen
offene Ports prüfen
lokale Dienste finden
Verbindungszustände anzeigen
```

Merksatz:

```
netstat zeigt Netzwerkstatus und Verbindungen.
```

---

ss unter Linux

ss ist ein modernes Werkzeug unter Linux, um Sockets und Netzwerkverbindungen anzuzeigen.

Es ersetzt häufig netstat.

Typische Informationen:

```
lauschende Ports
aktive TCP-Verbindungen
UDP-Sockets
Prozesszuordnung je nach Option
```

Merksatz:

```
ss zeigt schnell offene Ports und Verbindungen unter Linux.
```

Offene Ports prüfen

Offene Ports zeigen, welche Dienste erreichbar sein könnten.

Zu prüfen:

```
lauscht ein Dienst?
auf welcher IP lauscht er?
TCP oder UDP?
```

nur localhost oder extern?

welcher Prozess nutzt den Port?

sollte der Port offen sein?

Merksatz:

Ein offener Port ist ein möglicher Zugangspunkt.

---

localhost und 0.0.0.0 unterscheiden

127.0.0.1:

Dienst ist nur lokal erreichbar.

0.0.0.0:

Dienst lauscht auf allen IPv4-Schnittstellen.

Konkrete IP-Adresse:

Dienst lauscht nur auf dieser Schnittstelle.

Merksatz:

127.0.0.1 ist lokal.

0.0.0.0 ist auf allen Schnittstellen.

TCP-Verbindungszustände

Typische TCP-Zustände:

Zustand	Bedeutung
LISTEN	Dienst wartet auf Verbindungen
ESTABLISHED	Verbindung besteht
TIME_WAIT	Verbindung wurde beendet, Zustand läuft aus
SYN_SENT	Verbindungsaufbau gestartet

Zustand	Bedeutung
SYN_RECEIVED	Verbindungsaufbau teilweise angekommen
CLOSE_WAIT	Gegenseite hat geschlossen, lokale Seite noch nicht

Merksatz:

```
TCP-Zustände helfen,
Verbindungsprobleme einzugrenzen.
```

Port erreichbar testen

Um zu prüfen, ob ein Port erreichbar ist, kann je nach System ein Werkzeug genutzt werden.

Beispiele für Prinzipien:

```
TCP-Verbindung zu Zielport testen

HTTP-Antwort testen

Dienstprotokoll prüfen
```

Zu beachten:

```
Port offen bedeutet nicht automatisch,
dass die Anwendung korrekt funktioniert.
```

Merksatz:

```
Porttest prüft Erreichbarkeit,
nicht vollständige Anwendungsfunktion.
```

---

telnet als Porttest

telnet kann technisch genutzt werden, um einfache TCP-Verbindungen zu testen.

Wichtig:

```
Telnet als Verwaltungsprotokoll ist unsicher,
weil es unverschlüsselt ist.
```

Einordnung:

```
Telnet zur Administration vermeiden.
Als einfacher Porttest historisch bekannt,
aber moderne Alternativen bevorzugen.
```

Merksatz:

```
Telnet nicht für sichere Administration verwenden.
```

---

curl

curl ruft URLs und APIs über die Kommandozeile ab.

Nützlich für:

```
HTTP-Status prüfen
Header prüfen
API testen
Zertifikatsprobleme erkennen
Weiterleitungen prüfen
Antwortzeiten grob prüfen
Webserver-Erreichbarkeit testen
```

Merksatz:

```
curl ist ein wichtiges Werkzeug für Web- und API-Tests.
```

HTTP mit curl prüfen

Mit curl kann geprüft werden:

```
Antwortet der Webserver?
Welcher Statuscode kommt zurück?
```

Gibt es Weiterleitungen?
Wird HTTPS genutzt?
Stimmen Header?
Kommt eine Fehlermeldung?

Merksatz:

curl prüft Webdienste direkter als ein Browser.

---

wget

wget kann Dateien oder Webseiten über HTTP, HTTPS oder FTP herunterladen.

Nützlich für:

Download testen
Erreichbarkeit prüfen
automatisierte Abrufe
Spiegeln einfacher Inhalte

Merksatz:

wget dient vor allem zum Abrufen und Herunterladen.

Dienststatus unter Linux prüfen

Auf vielen Linux-Systemen wird systemctl genutzt.

Typische Prüfungen:

läuft der Dienst?
ist der Dienst aktiviert?
ist der Dienst fehlgeschlagen?
wann wurde er gestartet?
welche Fehlermeldung gibt es?

Merksatz:

systemctl prüft und verwaltet systemd-Dienste.

---

journalctl für Dienstlogs

journalctl zeigt Logs des systemd-Journals.

Nützlich für:

Fehler eines Dienstes prüfen
Logs seit Start ansehen
aktuelle Meldungen verfolgen
Bootprobleme analysieren

Merksatz:

journalctl und systemctl gehören bei Linux-Dienstfehlern zusammen.

Dienste unter Windows prüfen

Unter Windows können Dienste geprüft werden über:

Dienste-Verwaltung
Ereignisanzeige
Task-Manager
PowerShell
sc

Zu prüfen:

Dienst gestartet?
Starttyp korrekt?
Abhängigkeiten erfüllt?
Fehler im Ereignisprotokoll?
Dienstkonto korrekt?
Berechtigungen vorhanden?

Merksatz:

Windows-Dienstfehler stehen oft zusätzlich in der Ereignisanzeige.

Prozessprüfung

Prozesse zeigen, welche Programme gerade laufen.

Zu prüfen:

läuft der Prozess?
verbraucht er viel CPU?
verbraucht er viel RAM?
startet er immer wieder neu?
gehört er zum erwarteten Dienst?
läuft er mit welchem Benutzer?

Merksatz:

Dienstprobleme zeigen sich oft auch in Prozessen.

Ressourcen prüfen

Systemressourcen beeinflussen Dienste.

Wichtige Ressourcen:

CPU
RAM
Festplatte
I/O
Netzwerk
offene Dateien
Prozesslimits

Merksatz:

Ein Dienst kann fehlschlagen,
obwohl Netzwerk und Konfiguration richtig sind,
wenn Ressourcen fehlen.

Speicherplatz prüfen

Volle Datenträger sind häufige Fehlerursachen.

Folgen:

```
Logs können nicht geschrieben werden.
Datenbank stoppt.
Updates schlagen fehl.
Anwendung kann keine Dateien speichern.
Backup bricht ab.
System wird instabil.
```

Merksatz:

```
Bei Dienstproblemen immer Speicherplatz prüfen.
```

---

df unter Linux

df zeigt die Belegung von Dateisystemen.

Nützlich für:

```
freie Kapazität prüfen
volle Partitionen erkennen
Mountpunkte prüfen
```

Merksatz:

```
df zeigt,
wie voll Dateisysteme sind.
```

---

du unter Linux

du zeigt, wie viel Speicher bestimmte Ordner oder Dateien belegen.

Nützlich für:

```
große Ordner finden
Logwachstum prüfen
Speicherfresser eingrenzen
```

Merksatz:

```
df zeigt Dateisysteme.
du zeigt Ordner- und Dateigrößen.
```

---

top und htop

top zeigt laufende Prozesse und Ressourcennutzung.

htop ist eine komfortablere Variante, wenn installiert.

Nützlich für:

```
CPU-lastige Prozesse finden
RAM-Verbrauch prüfen
Prozesszustände ansehen
Systemlast beurteilen
```

Merksatz:

```
top und htop zeigen laufende Prozesse und Last.
```

Task-Manager unter Windows

Der Task-Manager zeigt:

```
laufende Prozesse
CPU-Auslastung
Arbeitsspeicher
Datenträgerlast
Netzwerk
Autostart
Benutzer
Dienste
```

Merksatz:

```
Task-Manager ist ein schneller Überblick über Windows-Systemlast.
```

PowerShell in der Administration

PowerShell ist eine Verwaltungs- und Automatisierungsumgebung unter Windows.

Sie wird genutzt für:

```
Benutzerverwaltung
Dienste prüfen
Prozesse prüfen
Dateien verwalten
Netzwerk prüfen
Ereignislogs auswerten
Microsoft-365-Administration
Automatisierung
```

Merksatz:

```
PowerShell ist ein zentrales Administrationswerkzeug unter Windows.
```

Ereignisanzeige unter Windows

Die Ereignisanzeige zeigt Windows-Logs.

Wichtige Bereiche:

```
Anwendung
Sicherheit
System
```

Typische Nutzung:

```
Dienstfehler prüfen
Anmeldeereignisse prüfen
Treiberfehler prüfen
```

Systemfehler analysieren
Sicherheitsereignisse ansehen

Merksatz:

Windows-Fehleranalyse führt häufig zur Ereignisanzeige.

---

whoami

whoami zeigt, unter welchem Benutzer man angemeldet ist.

Nützlich bei:

Rechteproblemen
falscher Anmeldung
Skriptausführung
Dienstkontenprüfung
Remote-Sitzungen

Merksatz:

Erst prüfen,
wer man auf dem System wirklich ist.

Gruppenmitgliedschaften prüfen

Gruppenmitgliedschaften bestimmen oft Rechte.

Zu prüfen:

ist Benutzer in richtiger Gruppe?
fehlt eine Gruppe?
ist eine alte Gruppe noch aktiv?
greifen Gruppenrichtlinien?
wurde Anmeldung neu gestartet?

Merksatz:

Viele Rechteprobleme sind Gruppenprobleme.

Hostname prüfen

Der Hostname identifiziert ein System im Netzwerk.

Wichtig für:

DNS
Dokumentation
Zertifikate
Monitoring
Logs
Inventarisierung

Merksatz:

Hostname,
DNS-Name
Zertifikat
und Dokumentation sollten zusammenpassen.

Datum und Uhrzeit prüfen

Falsche Uhrzeit kann Probleme verursachen.

Beispiele:

Zertifikatsfehler
Kerberos-Probleme
Logauswertung falsch
MFA-Probleme
geplante Aufgaben laufen falsch
Token ungültig

Merksatz:

Zeitabweichungen verursachen viele schwer erkennbare Fehler.

Umgebungsvariablen

Umgebungsvariablen beeinflussen Programme und Skripte.

Beispiele:

```
PATH
HOME
USER
TEMP
Proxy-Einstellungen
Anwendungskonfiguration
Zugangspfade
```

Merksatz:

```
Falsche Umgebungsvariablen können Programme anders verhalten lassen.
```

Konfigurationsdateien prüfen

Viele Dienste werden über Konfigurationsdateien gesteuert.

Zu prüfen:

```
Syntax korrekt?
Pfade richtig?
Ports richtig?
Zugangsdaten korrekt?
Zertifikate richtig eingetragen?
Änderungen dokumentiert?
Dienst nach Änderung neu geladen?
```

Merksatz:

```
Kleine Fehler in Konfigurationen können ganze Dienste stoppen.
```

Änderungen prüfen

Bei neuen Fehlern ist die wichtigste Frage oft:

```
Was wurde zuletzt geändert?
```

Mögliche Änderungen:

```
Update
Firewall-Regel
Zertifikat
Passwort
DNS-Eintrag
DHCP-Änderung
Routing
Dienstkonfiguration
Berechtigung
Netzwerkverkabelung
Cloud-Regel
```

Merksatz:

```
Neue Fehler hängen oft mit letzten Änderungen zusammen.
```

Systematische Fehlersuche mit Befehlen

Sinnvolle Reihenfolge:

```
1. Eigenes System prüfen.
2. IP-Konfiguration prüfen.
3. Gateway prüfen.
4. DNS prüfen.
5. Ziel per IP testen.
6. Ziel per Name testen.
7. Route verfolgen.
8. Port prüfen.
9. Dienststatus prüfen.
10. Logs prüfen.
11. Ressourcen prüfen.
12. Berechtigungen prüfen.
```

13. Änderungen prüfen.
14. Ergebnis dokumentieren.

Merksatz:

Systematische Befehlsnutzung spart Zeit.

Typische Fehlerbilder und passende Prüfung

Fehlerbild	Erste Prüfungen
Kein Internet	IP, Gateway, DNS, Router
Name geht nicht	DNS, DNS-Server, Cache
IP geht, Name nicht	DNS prüfen
Dienst nicht erreichbar	Port, Firewall, Dienststatus
Webseite 404	URL, Webserver, Pfad
Webseite 500	Anwendung, Backend, Logs
Login schlägt fehl	Benutzer, Passwort, MFA, Logs
Zugriff verweigert	Gruppen, Rechte, ACL
Langsame Verbindung	Latenz, Paketverlust, Last
Zertifikatsfehler	Uhrzeit, Zertifikat, Hostname

Merksatz:

Fehlerbild bestimmt die nächste sinnvolle Prüfung.

Typische Praxisfehler

Häufige Fehler:

DNS wird zu spät geprüft
nur ping genutzt
Firewall vergessen
Gateway nicht geprüft
IPv4 und IPv6 verwechselt
falsches Interface betrachtet
Dienststatus geprüft,

aber Port nicht

Port offen,

aber Anwendung nicht getestet

Logs nicht gelesen

letzte Änderungen ignoriert

Rechteproblem als Netzwerkproblem behandelt

Merksatz:

Fehlersuche braucht mehrere Blickwinkel.

Checkliste: Netzwerk- und Systemprüfung

IP-Adresse prüfen.

Subnetz prüfen.

Gateway prüfen.

DNS-Server prüfen.

Ping zur Gateway-IP testen.

Ping zur Ziel-IP testen.

Namensauflösung testen.

Route prüfen.

Port prüfen.

Dienststatus prüfen.

Prozess prüfen.

Logs prüfen.

Firewall prüfen.

Ressourcen prüfen.

Benutzer und Rechte prüfen.

Zertifikate und Uhrzeit prüfen.

letzte Änderungen prüfen.

Ergebnis dokumentieren.

Merksatz:

Erst Grundlagen prüfen,

dann Spezialfälle.

Typische IHK-Fragen

In AP1 und AP2 kann zum Beispiel gefragt werden:

- Wofür nutzt man ipconfig?
- Wofür nutzt man ipconfig /all?
- Wofür nutzt man ip addr?
- Was zeigt eine Routing-Tabelle?
- Was ist eine Standardroute?
- Wofür nutzt man ping?
- Warum beweist ein fehlgeschlagener ping nicht immer,
  dass ein Ziel offline ist?
- Wie prüft man DNS?
- Was macht nslookup?
- Was macht dig?
- Wofür nutzt man tracert oder traceroute?
- Was zeigt netstat?
- Was zeigt ss?
- Was bedeutet LISTEN?
- Was bedeutet ESTABLISHED?
- Warum ist 127.0.0.1 besonders?
- Wofür nutzt man curl?
- Was prüft systemctl?
- Was zeigt journalctl?
- Warum sollte man Logs prüfen?
- Warum ist Speicherplatzprüfung wichtig?
- Warum ist die Frage nach letzten Änderungen wichtig?

Typische Prüfungsfallen

ping prüft ICMP,
nicht automatisch jeden Dienst.

Kein ping kann durch Firewall entstehen.

IP funktioniert,
Name nicht:
DNS prüfen.

Hostname funktioniert nicht automatisch,
wenn IP funktioniert.

Standardroute ist wichtig für fremde Netze.

Spezifischste Route gewinnt.

tracert und traceroute zeigen Zwischenstationen,
aber nicht jeder Hop antwortet.

Offener Port bedeutet nicht,
dass Anwendung vollständig funktioniert.

127.0.0.1 ist nur lokal.

0.0.0.0 bedeutet alle IPv4-Schnittstellen.

LISTEN bedeutet Dienst wartet auf Verbindung.

ESTABLISHED bedeutet Verbindung besteht.

Telnet ist als Administrationsprotokoll unsicher.

curl ist nützlich für Webtests.

Dienst läuft nicht automatisch gleich Dienst funktioniert.

Logs liefern oft den entscheidenden Hinweis.

Volle Festplatte ist eine häufige Fehlerursache.

Uhrzeitfehler können Zertifikate und Anmeldung stören.

Letzte Änderungen immer prüfen.

Wichtige Begriffe kurz erklärt

Begriff	Kurze Erklärung
ipconfig	Windows-IP-Konfiguration anzeigen

Begriff	Kurze Erklärung
ipconfig /all	detaillierte Windows-Netzwerkinformationen
ip addr	Linux-IP-Adressen anzeigen
ifconfig	älteres Werkzeug für Netzwerkschnittstellen
MAC-Adresse	Hardwareadresse einer Netzwerkschnittstelle
Gateway	Übergang in andere Netze
Routing-Tabelle	Liste möglicher Netzwerkwege
Standardroute	allgemeiner Weg zu unbekannten Netzen
ping	ICMP-Erreichbarkeit testen
ICMP	Kontrollprotokoll für Netzwerkmeldungen
nslookup	DNS-Abfragewerkzeug
dig	detailliertes DNS-Abfragewerkzeug
DNS-Cache	zwischengespeicherte DNS-Antworten
tracert	Windows-Wegverfolgung zum Ziel
traceroute	Linux/macOS-Wegverfolgung zum Ziel
netstat	Netzwerkverbindungen anzeigen
ss	moderne Socketanzeige unter Linux
Port	logischer Dienstzugang
LISTEN	Dienst wartet auf Verbindungen
ESTABLISHED	Verbindung besteht
localhost	lokaler Rechner
127.0.0.1	lokale Loopback-Adresse
0.0.0.0	alle IPv4-Schnittstellen
curl	Web- und API-Anfragen testen
wget	Dateien oder Webseiten abrufen
systemctl	systemd-Dienste verwalten
journalctl	systemd-Logs anzeigen
Dienststatus	Zustand eines Dienstes
Prozess	laufendes Programm
df	Dateisystembelegung anzeigen
du	Ordner- und Dateigrößen anzeigen
top	Prozess- und Lastanzeige
htop	komfortablere Prozessanzeige

Begriff	Kurze Erklärung
Task-Manager	Windows-Prozess- und Leistungsanzeige
PowerShell	Windows-Verwaltung und Automatisierung
Ereignisanzeige	Windows-Loganzeige
whoami	aktueller Benutzer
Hostname	Name eines Systems
Umgebungsvariable	System- oder Programmeinstellung

IHK-sichere Kurzformulierung

Netzwerk- und Systembefehle dienen dazu, Fehler systematisch einzugrenzen und Zustände zu prüfen. Mit ipconfig oder ip addr wird die IP-Konfiguration geprüft, mit route print oder ip route die Routing-Tabelle. ping testet grundlegende Erreichbarkeit per ICMP, nslookup oder dig prüfen DNS, tracert oder traceroute zeigen den Weg zum Ziel. netstat oder ss zeigen offene Ports und Verbindungen. curl prüft Webdienste und APIs. systemctl und journalctl helfen bei der Prüfung von Diensten und Logs unter Linux, während unter Windows Task-Manager, Ereignisanzeige und PowerShell wichtige Werkzeuge sind. Bei der Fehlersuche sollten IP-Konfiguration, Gateway, DNS, Route, Port, Dienststatus, Logs, Ressourcen, Rechte und letzte Änderungen systematisch geprüft werden.

Merksätze

Befehle ersetzen Vermutungen durch Fakten.

Erst Grundlagen prüfen,
dann Spezialfälle.

IP-Konfiguration ist die Basis.

Gateway führt in andere Netze.

Ohne Gateway meist kein Zugriff auf fremde Netze.

Routing-Tabelle zeigt Netzwerkwege.

Standardroute ist Weg für unbekannte Ziele.

Spezifischste Route gewinnt.

ping prüft ICMP.

Kein ping ist kein endgültiger Beweis.

IP-Ping trennt DNS von Netzwerkprüfung.

Hostname-Ping prüft DNS mit.

Wenn IP geht,
aber Name nicht:
DNS prüfen.

nslookup prüft DNS.

dig zeigt DNS detailliert.

DNS-Cache kann veraltet sein.

tracert und traceroute zeigen den Weg.

Nicht jeder Hop muss antworten.

netstat zeigt Verbindungen.

ss zeigt Sockets unter Linux.

Offene Ports prüfen.

127.0.0.1 ist lokal.

0.0.0.0 bedeutet alle Schnittstellen.

LISTEN wartet auf Verbindungen.

ESTABLISHED ist aktive Verbindung.

Port offen heißt nicht Anwendung gesund.

Telnet nicht zur sicheren Administration nutzen.

curl prüft Webdienste.

wget lädt Inhalte ab.

systemctl prüft Dienste.

journalctl zeigt Dienstlogs.

Windows nutzt Ereignisanzeige.

Prozesse und Ressourcen mitprüfen.

Volle Festplatte verursacht viele Fehler.

df zeigt Dateisystembelegung.

du findet große Ordner.

top zeigt Prozesslast.

Task-Manager zeigt Windows-Systemlast.

whoami prüft aktuellen Benutzer.

Gruppenmitgliedschaften beeinflussen Rechte.

Hostname muss zur Dokumentation passen.

Falsche Uhrzeit verursacht Zertifikats- und Loginprobleme.

Konfigurationen sorgfältig prüfen.

Letzte Änderungen sind oft der Schlüssel.

Fehlersuche systematisch dokumentieren.