19.10. Admin-Befehle und Fehlersuche Spickzettel

Dieser Spickzettel fasst wichtige Befehle und Werkzeuge für Netzwerkadministration, Systemprüfung und Fehlersuche kompakt zusammen.

Er ist besonders wichtig für:

```
- IP-Konfiguration prüfen
- DNS prüfen
- Routing prüfen
- Ports prüfen
- Dienste prüfen
- Logs prüfen
- Erreichbarkeit testen
- Fehler systematisch eingrenzen
```

Merksatz:

```
Befehle liefern Fakten,
keine Vermutungen.
```

Grundidee der Fehlersuche

Bei Netzwerkproblemen immer systematisch prüfen.

Sinnvolle Reihenfolge:

```
1. physische Verbindung

2. IP-Konfiguration

3. Gateway

4. DNS

5. Routing
```

6. Firewall

7. Port

8. Dienst

9. Anwendung

10. Logs

Merksatz:

Erst Verbindung,
dann Adresse,
dann Dienst,
dann Anwendung.

Windows- und Linux-Werkzeuge im Überblick

Aufgabe	Windows	Linux/macOS
IP-Konfiguration anzeigen	ipconfig	ip addr
detaillierte IP-Konfiguration	ipconfig /all	ip addr, ip link
Routing anzeigen	route print	ip route
DNS prüfen	nslookup	dig, nslookup
Erreichbarkeit prüfen	ping	ping
Weg zum Ziel prüfen	tracert	traceroute
Verbindungen anzeigen	netstat	ss, netstat
Ports prüfen	netstat, PowerShell	ss, netstat
Webdienst prüfen	PowerShell, Browser	curl
Dienste prüfen	services.msc, sc, PowerShell	systemctl
Logs prüfen	Ereignisanzeige	journalctl
Prozesse prüfen	Task-Manager, tasklist	ps, top
Speicherplatz prüfen	Explorer, PowerShell	df, du
Benutzer prüfen	whoami	whoami, id

Merksatz:

Gleiche Aufgabe,
anderes Werkzeug je nach Betriebssystem.

IP-Konfiguration prüfen

Windows:

```
ipconfig

ipconfig /all
```

Linux/macOS:

```
ip addr

ip link
```

Wichtige Angaben:

IP-Adresse
Subnetzmaske oder Präfix
Standardgateway
DNS-Server
MAC-Adresse
DHCP aktiviert ja oder nein
Lease-Informationen
Interface-Status

Merksatz:

IP,
Maske,
Gateway
und DNS immer zusammen prüfen.

---

ipconfig

ipconfig zeigt unter Windows die IP-Konfiguration.

Wichtige Ausgaben:

```
IPv4-Adresse
Subnetzmaske
Standardgateway
```

Mit:

```
ipconfig /all
```

sieht man zusätzlich:

```
DNS-Server
DHCP-Status
MAC-Adresse
Lease-Zeiten
Adapterdetails
```

Merksatz:

```
ipconfig /all zeigt die wichtigen Details.
```

---

ip addr

ip addr zeigt unter Linux die IP-Adressen der Netzwerkschnittstellen.

Wichtige Punkte:

```
Interface-Name
IPv4-Adresse
IPv6-Adresse
Präfix
Status
MAC-Adresse
```

Beispiele für Interface-Namen:

```
eth0
ens18
enp0s3
wlan0
```

Merksatz:

```
ip addr ist unter Linux der Standard für IP-Adressen.
```

---

ip link

ip link zeigt den Zustand von Netzwerkschnittstellen.

Wichtige Angaben:

```
Interface aktiv oder inaktiv
MAC-Adresse
MTU
Link-Status
```

Typische Begriffe:

```
UP

DOWN
```

Merksatz:

```
ip link hilft,
den Interface-Status zu prüfen.
```

Gateway prüfen

Windows:

```
ipconfig
```

```
route print
```

Linux/macOS:

```
ip route
```

Wichtig:

```
Gateway muss im lokalen Subnetz erreichbar sein.
```

Typisches Fehlerbild:

```
lokale Geräte erreichbar,
Internet nicht erreichbar
```

Merksatz:

```
Lokal ja,
extern nein:
Gateway prüfen.
```

Routing prüfen

Windows:

```
route print
```

Linux/macOS:

```
ip route
```

Wichtige Angaben:

```
Zielnetz
Gateway oder Next Hop
Interface
Metrik
```

```
Standardroute
```

Standardroute IPv4:

```
0.0.0.0/0
```

oder unter Linux oft:

```
default
```

Merksatz:

```
Routing-Tabelle zeigt,
wohin Pakete gesendet werden.
```

---

route print

route print zeigt unter Windows die Routing-Tabelle.

Wichtig sind:

```
Netzwerkziel
Netzmaske
Gateway
Schnittstelle
Metrik
```

Typische Prüfung:

```
Gibt es eine Standardroute?

Ist das Gateway korrekt?

Gibt es eine spezifische Route zum Zielnetz?
```

Merksatz:

```
route print zeigt den Weg aus Sicht des Windows-Clients.
```

---

ip route

ip route zeigt unter Linux die Routing-Tabelle.

Typische Ausgabe enthält:

```
default via 192.168.1.1 dev eth0
```

Bedeutung:

```
Standardroute über Gateway 192.168.1.1 über Interface eth0
```

Merksatz:

```
default ist die Standardroute.
```

---

ping

ping prüft Erreichbarkeit mit ICMP.

Typische Nutzung:

```
Ziel-IP testen

Gateway testen

DNS-Namen testen
```

Wichtig:

```
ping prüft ICMP,
nicht automatisch den Dienst.
```

Beispiel:

```
ping funktioniert,
aber HTTPS kann trotzdem blockiert sein.
```

Merksatz:

```
ping ist ein Erreichbarkeitstest,
kein vollständiger Diensttest.
```

---

ping richtig interpretieren

Ergebnis	Mögliche Bedeutung
Antwort kommt	Ziel antwortet auf ICMP
keine Antwort	Ziel offline, ICMP blockiert, Routingproblem, Firewall
hohe Zeit	Latenz, Last, lange Strecke
Paketverlust	Funkproblem, Überlastung, Leitungsproblem
Name wird nicht gefunden	DNS-Problem
IP geht, Name nicht	DNS prüfen

Merksatz:

```
Kein ping heißt nicht automatisch,
dass das Ziel offline ist.
```

DNS prüfen

Windows:

```
nslookup
```

Linux/macOS:

```
dig

nslookup
```

Prüfen:

```
Wird der Name aufgelöst?

Welche IP wird geliefert?

Welcher DNS-Server antwortet?

Ist der Record korrekt?

Gibt es interne und externe Unterschiede?
```

Merksatz:

```
DNS gezielt mit nslookup oder dig prüfen.
```

---

nslookup

nslookup prüft Namensauflösung.

Nützlich für:

```
A-Record prüfen
AAAA-Record prüfen
DNS-Server testen
Fehler bei Namensauflösung erkennen
```

Typisches Fehlerbild:

```
IP-Adresse funktioniert,
Name funktioniert nicht
```

Wahrscheinlich:

```
DNS-Problem
```

Merksatz:

```
nslookup prüft DNS,
nicht den eigentlichen Dienst.
```

---

dig

dig zeigt DNS-Abfragen detailliert an.

Nützlich für:

```
genaue DNS-Antworten
Recordtypen
TTL
Antwortserver
Debugging
Vergleich verschiedener DNS-Server
```

Typische Recordtypen:

```
A
AAAA
MX
CNAME
TXT
NS
PTR
```

Merksatz:

```
dig ist besonders hilfreich für detaillierte DNS-Prüfung.
```

---

tracert und traceroute

Windows:

```
tracert
```

Linux/macOS:

```
traceroute
```

Aufgabe:

```
Weg zum Ziel über Router anzeigen
```

Nützlich für:

```
Routingprobleme eingrenzen
Hops sichtbar machen
Latenzen grob erkennen
Unterbrechungen lokalisieren
```

Merksatz:

```
traceroute zeigt den Weg,
aber nicht jeder Hop muss antworten.
```

---

traceroute richtig interpretieren

Beobachtung	Mögliche Bedeutung
erster Hop fehlt	Gateway oder lokales Netz prüfen
Abbruch nach bestimmtem Hop	Firewall, Routing, Provider
einzelne Sternchen	Hop antwortet nicht, Weg kann trotzdem funktionieren
hohe Latenz ab bestimmtem Hop	Engstelle oder lange Strecke
Ziel antwortet nicht	Ziel blockiert oder nicht erreichbar

Merksatz:

```
Sternchen bedeuten nicht automatisch Totalausfall.
```

---

netstat

netstat zeigt Netzwerkverbindungen und lauschende Ports.

Nützlich für:

```
offene Ports anzeigen
bestehende Verbindungen prüfen
Dienst lauscht ja oder nein
Quell- und Zielports sehen
TCP-Status prüfen
```

Typische Zustände:

```
LISTEN

ESTABLISHED

TIME_WAIT

CLOSE_WAIT
```

Merksatz:

```
netstat zeigt,
welche Verbindungen und Ports aktiv sind.
```

---

ss

ss ist unter Linux ein modernes Werkzeug für Sockets und Ports.

Nützlich für:

```
lauschende Ports
TCP-Verbindungen
UDP-Sockets
Prozesszuordnung je nach Berechtigung
schnelle Portprüfung lokal
```

Merksatz:

```
ss ersetzt auf vielen Linux-Systemen netstat.
```

TCP-Status kurz erklärt

Status	Bedeutung
LISTEN	Dienst wartet auf Verbindungen
ESTABLISHED	Verbindung besteht
SYN_SENT	Verbindungsaufbau gestartet
SYN_RECEIVED	SYN empfangen und beantwortet
TIME_WAIT	Verbindung beendet, Wartephase
CLOSE_WAIT	Gegenseite hat beendet, lokale Anwendung noch nicht

Merksatz:

> LISTEN bedeutet:
> Dienst lauscht.

Portprüfung

Bei Dienstproblemen prüfen:

> Läuft der Dienst?
>
> Lauscht der Dienst auf dem richtigen Port?
>
> Lauscht er auf der richtigen Schnittstelle?
>
> Blockiert die lokale Firewall?
>
> Blockiert eine Netzwerkfirewall?
>
> Stimmt TCP oder UDP?
>
> Ist NAT oder Portweiterleitung korrekt?

Merksatz:

> Host erreichbar heißt nicht,
> dass der Port erreichbar ist.

---

localhost, 127.0.0.1 und 0.0.0.0

127.0.0.1:

```
Loopback-Adresse

nur lokal auf demselben System erreichbar
```

localhost:

```
Name für lokalen Rechner
```

0.0.0.0:

```
Dienst lauscht auf allen IPv4-Schnittstellen des Hosts
```

Merksatz:

```
127.0.0.1 ist lokal.
0.0.0.0 bedeutet alle Interfaces.
```

---

curl

curl prüft Webdienste, APIs und HTTP-Antworten.

Nützlich für:

```
HTTP-Statuscode
Weiterleitungen
Header
TLS-Verbindung
API-Antwort
Erreichbarkeit ohne Browser
Proxy-Fehler
Backend-Fehler
```

Merksatz:

curl prüft Webdienste direkter als ein Browser.

HTTP-Statuscodes

Statuscode	Bedeutung
200	erfolgreich
301 / 302	Weiterleitung
400	fehlerhafte Anfrage
401	nicht authentifiziert
403	verboten
404	nicht gefunden
500	interner Serverfehler
502	Gateway oder Backend-Fehler
503	Dienst nicht verfügbar
504	Gateway Timeout

Merksatz:

```
4xx deutet eher auf Client,
Anfrage
oder Rechte.

5xx deutet eher auf Server,
Anwendung
oder Backend.
```

Dienste prüfen unter Linux

Wichtige Werkzeuge:

```
systemctl

journalctl
```

systemctl prüft und verwaltet Dienste.

journalctl zeigt Logs von systemd-Diensten.

Merksatz:

```
systemctl zeigt den Dienststatus.
journalctl zeigt die Logs.
```

---

systemctl

systemctl wird genutzt für:

```
Dienststatus prüfen
Dienst starten
Dienst stoppen
Dienst neu starten
Dienst aktivieren
Dienst deaktivieren
```

Typische Prüfung:

```
Läuft der Dienst?

Ist der Dienst beim Booten aktiviert?

Gibt es Fehler beim Start?
```

Merksatz:

```
systemctl beantwortet:
Läuft der Dienst?
```

---

journalctl

journalctl zeigt Logs unter systemd.

Nützlich für:

Dienstfehler
Startprobleme
Abstürze
Berechtigungsfehler
Konfigurationsfehler
Zeitliche Eingrenzung

Merksatz:

journalctl beantwortet:
Warum hat der Dienst Probleme?

Dienste prüfen unter Windows

Wichtige Werkzeuge:

services.msc

Task-Manager

Ereignisanzeige

PowerShell

sc

Typische Fragen:

Läuft der Dienst?

Ist der Starttyp korrekt?

Gibt es Fehlermeldungen?

Gibt es Abhängigkeiten?

Merksatz:

```
Unter Windows helfen Diensteverwaltung und Ereignisanzeige.
```

Windows-Ereignisanzeige

Die Ereignisanzeige zeigt Windows-Logs.

Wichtige Bereiche:

```
Anwendung

Sicherheit

System

Setup
```

Nützlich für:

```
Anmeldefehler
Dienstfehler
Systemfehler
Treiberprobleme
Sicherheitsereignisse
```

Merksatz:

```
Windows-Logs stehen in der Ereignisanzeige.
```

Prozesse prüfen

Windows:

```
Task-Manager

tasklist
```

Linux/macOS:

```
ps

top

htop
```

Nützlich für:

```
hohe CPU-Last
hoher RAM-Verbrauch
hängende Prozesse
Prozess läuft ja oder nein
Prozess-ID finden
```

Merksatz:

```
Prozess läuft nicht automatisch gleich Dienst erreichbar.
```

Speicherplatz prüfen

Linux/macOS:

```
df

du
```

Windows:

```
Explorer

Datenträgerverwaltung

PowerShell
```

Wichtige Fragen:

Ist das Dateisystem voll?

Ist die Logpartition voll?

Ist genug Speicher für Backup oder Update vorhanden?

Wächst eine Datei ungewöhnlich stark?

Merksatz:

Volle Datenträger verursachen viele Dienstprobleme.

---

df und du

df zeigt freien Speicher auf Dateisystemen.

du zeigt Speicherverbrauch von Dateien und Ordnern.

Unterschied:

df:
Wie voll ist das Dateisystem?

du:
Welcher Ordner verbraucht Speicher?

Merksatz:

df zeigt Gesamtfüllstand.
du zeigt Verursacher.

Benutzer und Rechte prüfen

Windows:

whoami

whoami /groups

Linux/macOS:

```
whoami

id

groups
```

Prüfen:

```
Welcher Benutzer bin ich?

In welchen Gruppen bin ich?

Habe ich die nötigen Rechte?

Wird ein anderes Konto verwendet?
```

Merksatz:

```
Anmeldung erfolgreich heißt nicht automatisch Zugriff erlaubt.
```

Dateirechte prüfen unter Linux

Wichtige Begriffe:

```
Besitzer

Gruppe

Rechte

Lesen

Schreiben

Ausführen
```

Typische Werkzeuge:

```
ls -l

chmod

chown
```

Merksatz:

```
Linux-Rechte bestehen aus Besitzer,
Gruppe
und anderen.
```

Freigaberechte prüfen

Bei Dateifreigaben prüfen:

```
Freigaberechte

Dateisystemrechte

Gruppenmitgliedschaft

Deny-Regeln

Vererbung

Benutzerkontext
```

Merksatz:

```
Bei Freigaben zählen oft Freigaberechte und Dateisystemrechte zusammen.
```

Logs allgemein prüfen

Wichtige Fragen:

```
Wann trat der Fehler auf?

Welches System meldet den Fehler?

Welcher Benutzer ist betroffen?

Welche Quelle und welches Ziel?

Welche Fehlermeldung?

Wiederholt sich das Muster?

Gab es kurz vorher Änderungen?
```

Merksatz:

```
Logs beantworten:
wer,
wann,
was,
von wo,
mit welchem Ergebnis.
```

Zeit und NTP prüfen

Korrekte Zeit ist wichtig für:

```
Logs
Zertifikate
Kerberos
MFA
Tokens
geplante Aufgaben
Forensik
```

Typische Fehler:

Zertifikatswarnung
Kerberos-Fehler
ungültige Tokens
falsche Log-Reihenfolge

Merksatz:

Bei Zertifikaten,
Kerberos
und Logs immer auch Zeit prüfen.

Firewall prüfen

Bei Firewall-Verdacht prüfen:

Quelle

Ziel

Port

Protokoll

Richtung

Zone

Regelreihenfolge

NAT

Logs

lokale Firewall

Netzwerkfirewall

Merksatz:

Firewallprüfung braucht Quelle,
Ziel,
Port
und Protokoll.

NAT und Portweiterleitung prüfen

Prüfen:

öffentliche IP korrekt?

DNS zeigt auf richtige öffentliche IP?

externer Port korrekt?

interne Ziel-IP korrekt?

interner Port korrekt?

Dienst läuft intern?

lokale Firewall erlaubt?

Router-Firewall erlaubt?

doppeltes NAT?

CGNAT?

Merksatz:

Portweiterleitung braucht DNS,
NAT,
Firewall
und laufenden Dienst.

Typische Fehlerbilder und erste Prüfung

Fehlerbild	Erste Prüfung
keine IP-Adresse	DHCP, Kabel, WLAN, VLAN
169.254.x.x	DHCP nicht erreicht
lokales Netz geht, Internet nicht	Gateway, Routing, NAT
IP geht, Name nicht	DNS
ping geht, Webseite nicht	Port, Dienst, Firewall
Login geht, Zugriff nicht	Rechte, Gruppe, ACL
Anmeldung scheitert	Konto, Passwort, MFA, Verzeichnisdienst
Zertifikatswarnung	Zertifikat, Hostname, Uhrzeit
Dienst startet nicht	Logs, Konfiguration, Abhängigkeiten
alles langsam	Last, DNS, Netzwerk, Speicher

Merksatz:

> Fehlerbild bestimmt den ersten sinnvollen Test.

Mini-Workflow: Client kommt nicht ins Internet

Prüfen:

1. IP-Adresse vorhanden?
2. APIPA-Adresse 169.254.x.x?
3. Subnetzmaske korrekt?
4. Gateway eingetragen?
5. Gateway erreichbar?
6. externe IP erreichbar?
7. DNS-Auflösung funktioniert?
8. Firewall oder Proxy?

9. NAT oder Provider?

Merksatz:

Erst IP,
dann Gateway,
dann DNS,
dann Firewall und NAT.

Mini-Workflow: Webseite nicht erreichbar

Prüfen:

1. Name wird aufgelöst?

2. Ziel-IP erreichbar?

3. TCP 80 oder 443 erreichbar?

4. Webserver läuft?

5. TLS-Zertifikat gültig?

6. Reverse Proxy korrekt?

7. Anwendung liefert Fehler?

8. Logs prüfen.

Merksatz:

Webfehler können DNS,
Port,
TLS,
Proxy
oder Anwendung betreffen.

Mini-Workflow: Datei-Freigabe nicht erreichbar

Prüfen:

```
1. Netzwerkverbindung?

2. Namensauflösung?

3. Server erreichbar?

4. SMB-Port TCP 445 erreichbar?

5. Benutzer angemeldet?

6. Freigaberechte?

7. Dateisystemrechte?

8. Gruppenmitgliedschaft?

9. Firewall?

10. Logs?
```

Merksatz:

```
Freigabeprobleme können Netzwerk oder Rechte sein.
```

Mini-Workflow: SSH nicht erreichbar

Prüfen:

```
1. Ziel-IP erreichbar?

2. TCP 22 erreichbar?

3. SSH-Dienst läuft?

4. Firewall erlaubt Zugriff?
```

5. richtige Quell-IP erlaubt?

6. Benutzer erlaubt?

7. Schlüssel korrekt?

8. Host-Key-Warnung prüfen.

9. Logs prüfen.

Merksatz:

SSH-Probleme können Port,
Dienst,
Schlüssel,
Benutzer
oder Firewall sein.

Mini-Workflow: DNS-Problem

Prüfen:

1. DNS-Server eingetragen?

2. DNS-Server erreichbar?

3. richtiger Record vorhanden?

4. interner oder externer DNS?

5. DNS-Suffix korrekt?

6. Split-DNS relevant?

7. DNS-Cache veraltet?

8. Firewall blockiert UDP/TCP 53?

Merksatz:

IP ja,
Name nein:
DNS prüfen.

Mini-Workflow: DHCP-Problem

Prüfen:

1. Client im richtigen VLAN?

2. Switchport korrekt?

3. DHCP-Server aktiv?

4. DHCP-Scope frei?

5. DHCP-Relay korrekt?

6. Firewall erlaubt DHCP?

7. Reservierung oder Ausschluss?

8. Lease erneuern.

Merksatz:

169.254.x.x:
DHCP,
VLAN
oder Relay prüfen.

Mini-Workflow: Dienst startet nicht

Prüfen:

1. Dienststatus

2. Logs

3. Konfigurationsdatei

4. Berechtigungen

5. Abhängigkeiten

6. belegter Port

7. Speicherplatz

8. Zertifikate oder Schlüssel

9. letzte Änderung

Merksatz:

Dienstfehler stehen oft in Logs.

Mini-Workflow: langsames Netzwerk

Prüfen:

Latenz

Paketverlust

DNS-Auflösung

WLAN-Signal

Bandbreite

Duplex oder Link-Geschwindigkeit

CPU oder RAM

```
Speicher-I/O

Firewall oder Proxy

Serverlast
```

Merksatz:

```
Langsamkeit ist kein einzelner Fehler,
sondern muss eingegrenzt werden.
```

Prüfungsfalle: Befehl ohne Interpretation

Ein Befehl allein löst noch keine Aufgabe.

Schlecht:

```
ping ausführen.
```

Besser:

```
Mit ping wird geprüft,
ob das Ziel per ICMP erreichbar ist.
Wenn die IP erreichbar ist,
der Name aber nicht,
sollte DNS geprüft werden.
```

Merksatz:

```
In der Prüfung nicht nur Befehl nennen,
sondern Zweck erklären.
```

Prüfungsfalle: ping als Diensttest

ping prüft ICMP.

ping prüft nicht:

```
HTTPS
SSH
SMB
Datenbank
Anmeldung
Rechte
Anwendung
```

Merksatz:

```
ping erfolgreich heißt nicht,
dass der Dienst funktioniert.
```

Prüfungsfalle: DNS und Dienst verwechseln

Wenn ein Name nicht aufgelöst wird, ist zuerst DNS zu prüfen.

Wenn der Name aufgelöst wird, aber die Anwendung nicht funktioniert, sind eher Port, Dienst, Firewall, TLS oder Anwendung zu prüfen.

Merksatz:

```
DNS findet die Adresse.
Der Dienst muss danach trotzdem funktionieren.
```

Prüfungsfalle: Rechteproblem als Netzwerkproblem deuten

Fehlerbild:

```
Benutzer kann sich anmelden,
aber Datei nicht öffnen.
```

Wahrscheinlich:

```
Autorisierung,
Gruppenmitgliedschaft,
ACL
oder Freigaberecht
```

Nicht zuerst:

```
DNS
Gateway
NAT
```

Merksatz:

```
Login ja,
Zugriff nein:
Rechte prüfen.
```

Typische IHK-Fragen

In AP1 und AP2 kann zum Beispiel gefragt werden:

```
- Welcher Befehl zeigt die IP-Konfiguration unter Windows?
- Welcher Befehl zeigt IP-Adressen unter Linux?
- Wie prüft man DNS?
- Was prüft ping?
- Warum ersetzt ping keinen Diensttest?
- Was zeigt tracert oder traceroute?
- Wie prüft man Routing?
- Wie erkennt man,
  ob ein Dienst auf einem Port lauscht?
- Was bedeutet LISTEN?
- Wie prüft man Linux-Dienste?
- Wie prüft man Windows-Logs?
- Warum ist NTP wichtig für Logs?
- Welche Schritte sind sinnvoll,
  wenn ein Client nicht ins Internet kommt?
- Wie grenzt man ein Firewallproblem ein?
```

IHK-sichere Kurzformulierung

Bei der Netzwerk- und Systemfehlersuche sollten zuerst Fakten gesammelt werden. Unter Windows zeigt ipconfig die IP-Konfiguration, unter Linux ip addr. Routing wird mit route print oder ip route geprüft. DNS wird gezielt mit nslookup oder dig geprüft. ping testet ICMP-Erreichbarkeit, ersetzt aber keinen Port- oder Diensttest. tracert oder traceroute zeigen den Weg über Router. netstat

oder ss zeigen Verbindungen und lauschende Ports. systemctl prüft Linux-Dienste, journalctl zeigt deren Logs. Unter Windows helfen Diensteverwaltung und Ereignisanzeige. Gute Fehlersuche erfolgt systematisch: Verbindung, IP, Gateway, DNS, Routing, Firewall, Port, Dienst, Anwendung und Logs.

Merksätze

Befehle liefern Fakten.

Erst prüfen,
dann vermuten.

IP,
Maske,
Gateway
und DNS zusammen prüfen.

ipconfig zeigt Windows-IP-Konfiguration.

ip addr zeigt Linux-IP-Konfiguration.

route print zeigt Windows-Routing.

ip route zeigt Linux-Routing.

ping prüft ICMP.

ping ist kein Diensttest.

nslookup prüft DNS.

dig prüft DNS detailliert.

tracert und traceroute zeigen den Weg.

netstat zeigt Verbindungen und Ports.

ss zeigt Sockets und Ports unter Linux.

LISTEN heißt:

Dienst wartet auf Verbindungen.

ESTABLISHED heißt:
Verbindung besteht.

127.0.0.1 ist lokal.

0.0.0.0 bedeutet alle Interfaces.

curl prüft Webdienste.

systemctl prüft Dienststatus.

journalctl zeigt Logs.

Windows-Logs stehen in der Ereignisanzeige.

df zeigt Dateisystemfüllstand.

du zeigt Ordnerverbrauch.

whoami zeigt den Benutzerkontext.

Logs beantworten:
wer,
wann,
was,
von wo,
mit welchem Ergebnis.

NTP macht Logzeiten vergleichbar.

Firewallprüfung braucht Quelle,
Ziel,
Port
und Protokoll.

Portweiterleitung braucht DNS,
NAT,
Firewall

und Dienst.

IP ja,
Name nein:
DNS.

Lokal ja,
extern nein:
Gateway.

Host ja,
Dienst nein:
Port,
Dienst,
Firewall.

Login ja,
Zugriff nein:
Rechte.

Dienst startet nicht:
Logs prüfen.