

# corewire

## Linux

### Logs & Fehleranalyse

# Folien-Hinweis

- Space, Page down: Nächste Folie
- Page up: Vorherige Folie
- ESC, o: Übersicht

[Zur Kapitelübersicht](#)

# Einführung in Linux-Logs

# Was sind Logs?

**Logs sind strukturierte Aufzeichnungen von:**

- **Systemaktivitäten und Zustandsänderungen**
- **Servicemeldungen und Fehlermeldungen**
- **Benutzeraktionen und Anmeldungen**
- **Sicherheitsereignisse und Warnungen**

# Warum sind Logs wichtig?

- Fehlerdiagnose und Problemlösung
- Systemüberwachung und Performance-Analyse
- Sicherheitsauswertung und Compliance
- Kapazitätsplanung und Vorhersagen

# Log-Arten in Linux

- **System-Logs:** Kernel, Boot, Authentication
- **Service-Logs:** Webserver, Datenbanken, Anwendungen
- **Security-Logs:** Failed Logins, sudo, Firewall

# Das systemd Journal (journalctl)

# journalctl - Das moderne Log-System

systemd journal sammelt alle Logs zentral:

- **Binäres Format** für bessere Performance
- **Strukturierte Metadaten** für erweiterte Filterung
- **Automatische Rotation** und Komprimierung
- **Integration** mit allen systemd Services

```
# Alle Journal-Einträge anzeigen
journalctl

# Journal in Echtzeit verfolgen
journalctl -f

# Nur die letzten 50 Einträge
journalctl -n 50
```



# journalctl - Grundlegende Befehle

```
# Journal rückwärts durchblättern (neueste zuerst)  
journalctl -r
```

```
# Nur bestimmte Prioritätsstufen anzeigen  
journalctl -p err          # Nur Fehler  
journalctl -p warning      # Warnungen und höher
```

```
# Nach Service filtern  
journalctl -u sshd.service  
journalctl -u nginx
```

# Traditionelles syslog-System

# rsyslog - Der klassische Logger

rsyslog ist das traditionelle Logging-System:

- Text-basierte Logs in `/var/log/`
- Konfigurierbare Regeln für Log-Routing
- Remote Logging über Netzwerk möglich
- Längere Historie und etablierte Tools

```
# Zentrale Log-Datei anzeigen  
tail -f /var/log/syslog  
  
# System-Messages  
tail -f /var/log/messages  
  
# Kernel-Nachrichten  
tail -f /var/log/kern.log
```

# Wichtige Log-Dateien in /var/log/

Datei	Inhalt
<code>/var/log/syslog</code>	Allgemeine System-Logs
<code>/var/log/auth.log</code>	Authentifizierung und Autorisierung
<code>/var/log/kernel.log</code>	Kernel-Nachrichten
<code>/var/log/mail.log</code>	E-Mail System

# Log-Rotation

# logrotate - Automatische Log-Verwaltung

## Warum Log-Rotation?

- **Festplattenspeicher** begrenzt verhindern
- **Performance** bei sehr großen Log-Dateien
- **Archivierung** alter Logs für Compliance
- **Automatisierte Bereinigung** nach Zeitraum

```
# logrotate Konfiguration anzeigen
cat /etc/logrotate.conf

# Service-spezifische Konfigurationen
ls /etc/logrotate.d/

# logrotate manuell testen
sudo logrotate -d /etc/logrotate.conf # Dry-run
sudo logrotate -f /etc/logrotate.conf # Force rotation
```

# Logs als Werkzeug zur Fehlersuche

# Systematische Fehlerdiagnose mit Logs

## Der strukturierte Ansatz:

1. **Problem identifizieren:** Wann trat der Fehler auf?
2. **Zeitraum eingrenzen:** Logs für relevanten Zeitabschnitt
3. **Services identifizieren:** Welche Dienste sind betroffen?
4. **Kausaler Zusammenhang:** Was passierte vor dem Fehler?
5. **Root Cause:** Grundursache in den Logs finden



# Häufige Fehlerszenarien erkennen

## Service startet nicht:

```
# Service-Status prüfen
systemctl status nginx

# Service-spezifische Logs prüfen
journalctl -u nginx --since "5 minutes ago"

# Konfigurationsfehler finden
nginx -t # Syntax-Check
```

## Performance-Probleme:

```
# System-Ressourcen in Logs
journalctl -p warning --since "1 hour ago"

# Out-of-Memory Ereignisse
journalctl -k | grep -i "killed process"

# Disk-Space Probleme
journalctl | grep -i "no space left"
```

# Netzwerk- und Verbindungsfehler

## SSH-Verbindungsprobleme:

```
# SSH-Authentifizierungs-Logs
journalctl -u ssh --since "today"
grep "Failed password" /var/log/auth.log

# Netzwerk-Interface Probleme
journalctl -k | grep -i "network\|interface"
```

# Log-Analyse mit Standard-Tools

## grep für Pattern-Matching:

```
# Fehler-Level finden
grep -i "error\|critical\|failed" /var/log/syslog

# IP-Adressen extrahieren
grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /var/log/auth.log

# Zeitraum-spezifische Suche
grep "2024-02-24 14:" /var/log/syslog
```

# Praktische Troubleshooting- Beispiele

# Beispiel 1: Server startet nicht richtig

**Problemszenario:** System startet, aber Services funktionieren nicht

## Lösungsansatz:

```
# 1. Boot-Logs analysieren
journalctl -b

# 2. Failed Services identifizieren
systemctl --failed

# 3. Service-spezifische Probleme
journalctl -u apache2
```

# Beispiel 2: Hohe Systemlast

**Problemszenario:** System reagiert langsam, hohe Load Average

## Lösungsansatz:

```
# 1. Ressourcen mit htop prüfen
htop

# 2. Out-of-Memory Events prüfen
journalctl --since "1 hour ago" | grep -i "killed process"

# 3. Disk I/O Probleme
journalctl -k | grep -i "blocked\|hung_task"

# 4. CPU-intensive Prozesse in Logs
journalctl | grep -i "cpu\|load"
```

# Beispiel 3: Security Incident

**Problemszenario:** Verdächtige Aktivitäten auf dem Server

## Lösungsansatz:

```
# 1. Failed Login Attempts
grep "Failed password" /var/log/auth.log | tail -20

# 2. Successful Logins analysieren
grep "Accepted password" /var/log/auth.log | tail -10

# 3. sudo Aktivitäten prüfen
grep "sudo" /var/log/auth.log | tail -15

# 4. Ungewöhnliche Prozesse
journalctl | grep -i "started session\|ended session"
```

## Zur Kapitelübersicht

- Vorheriges Kapitel: [SSH Grundlagen](#)