

corewire

Linux

Paketmanager

Folien-Hinweis

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

[Zur Kapitelübersicht](#)

Was ist ein Paketmanager?

- **Zentrales Tool zur Verwaltung von Software-Paketen**
- **Automatische Abhängigkeitsverwaltung**
- **Einheitliche Installation/Deinstallation**
- **Integrität und Sicherheit der Software gewährleistet**

Warum Paketmanager verwenden?

- **Einfache Installation:** Ein Befehl installiert Software
- **Abhängigkeiten:** Automatische Verwaltung von Abhängigkeiten
- **Sicherheit:** Digitale Signaturen und Checksummen
- **Updates:** Zentrale Update-Verwaltung
- **Deinstallation:** Saubere Entfernung inkl. Abhängigkeiten

Vergleich verschiedener Paketmanager

APT (Advanced Package Tool)

Verwendet von: Debian, Ubuntu, Linux Mint

```
# Paketlisten aktualisieren  
sudo apt update  
  
# Paket installieren  
sudo apt install nginx  
  
# System upgraden  
sudo apt upgrade
```

YUM/DNF

- **Verwendet von:** RHEL, CentOS, Fedora, Rocky Linux
- DNF ist der Nachfolger von YUM.

```
# Paket installieren (DNF)
sudo dnf install nginx

# System updaten
sudo dnf update
```

Snap

- **Universal Package Manager** von Canonical
- Containerisierte Anwendungen

```
# Snap installieren  
sudo snap install code  
  
# Installierte Snaps anzeigen  
snap list  
  
# Snap updaten  
sudo snap refresh code
```


Abhängigkeiten verstehen

Was sind Abhängigkeiten?

- **Bibliotheken und Programme**, die eine Software benötigt
- **Shared Libraries**: Gemeinsam genutzte Systemkomponenten
- **Versionskompatibilität**: Bestimmte Versionen erforderlich

Abhängigkeitsprobleme

Dependency Hell

- Konflikte zwischen verschiedenen Versionsanforderungen
- Unaufgelöste Abhängigkeiten
- Circular Dependencies (zirkuläre Abhängigkeiten)

Abhängigkeitsprobleme

Lösungsansätze

- Moderne Paketmanager lösen dies automatisch
- Virtuelle Umgebungen (z.B. Python venv, Docker)
- Statische Kompilierung

Abhängigkeiten anzeigen

```
# APT - Abhängigkeiten anzeigen  
apt depends nginx  
  
# RPM - Abhängigkeiten anzeigen  
rpm -qR nginx
```

Pakete installieren, updaten, deinstallieren

Installation von Paketen

APT (Debian/Ubuntu)

```
# Einzelnes Paket
sudo apt install nginx

# Mehrere Pakete
sudo apt install nginx mysql-server php

# Aus lokaler Datei
sudo dpkg -i paket.deb
```

Installation von Paketen

YUM/DNF (RHEL/Fedora)

```
# Einzelnes Paket
sudo dnf install nginx

# Gruppe von Paketen
sudo dnf groupinstall "Development Tools"

# Aus lokaler Datei
sudo rpm -i paket.rpm
```


Updates durchführen

APT

```
# Paketlisten aktualisieren
sudo apt update

# Verfügbare Updates anzeigen
apt list --upgradable

# Alle Pakete updaten
sudo apt upgrade

# System-Upgrade (neue Version)
sudo apt dist-upgrade
```

DNF

```
# Einzelnes Paket updaten
sudo dnf update nginx

# Alle Pakete updaten
sudo dnf update
```

Deinstallation von Paketen

APT

```
# Paket entfernen (Konfiguration bleibt)
sudo apt remove nginx

# Paket komplett entfernen
sudo apt purge nginx

# Nicht mehr benötigte Abhängigkeiten entfernen
sudo apt autoremove
```

DNF

```
# Paket entfernen
sudo dnf remove nginx

# Nicht mehr benötigte Abhängigkeiten entfernen
sudo dnf autoremove
```

Repositories

Was sind Repositories?

- **Zentrale Sammlungen** von Software-Paketen
- **Vertrauenswürdige Quellen** mit signierten Paketen
- **Verschiedene Kategorien:** Main, Universe, Contrib, Non-free
- **Spiegelserver** für bessere Performance

Repository-Konfiguration

APT - /etc/apt/sources.list

```
# Hauptrepository
deb http://archive.ubuntu.com/ubuntu/ jammy main restricted
deb http://archive.ubuntu.com/ubuntu/ jammy-updates main restricted
deb http://security.ubuntu.com/ubuntu/ jammy-security main restricted

# Universe und Multiverse
deb http://archive.ubuntu.com/ubuntu/ jammy universe multiverse
```

DNF - /etc/yum.repos.d/

```
# Repository hinzufügen
sudo dnf config-manager --add-repo https://example.com/repo.repo

# Repository aktivieren/deaktivieren
sudo dnf config-manager --enable repo-name
sudo dnf config-manager --disable repo-name
```

Externe Repositories hinzufügen

Ubuntu - PPA (Personal Package Archive)

```
# PPA hinzufügen
sudo add-apt-repository ppa:nginx/stable
sudo apt update
sudo apt install nginx

# PPA entfernen
sudo add-apt-repository --remove ppa:nginx/stable
```

DNF - EPEL Repository

```
# EPEL für RHEL/CentOS
sudo dnf install epel-release

# RPM Fusion für Fedora
sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release
```

Repository-Sicherheit

GPG-Schlüssel

- **Digitale Signaturen** zur Verifikation der Pakete
- **Schutz vor manipulierten Paketen**

```
# GPG-Schlüssel hinzufügen (APT)
wget -qO - https://example.com/key.gpg | sudo apt-key add -

# GPG-Schlüssel hinzufügen (DNF)
sudo rpm --import https://example.com/RPM-GPG-KEY
```

Autoupdates

Automatische Updates konfigurieren

Ubuntu - unattended-upgrades

```
# Installation
sudo apt install unattended-upgrades

# Konfiguration aktivieren
sudo dpkg-reconfigure unattended-upgrades

# Konfiguration bearbeiten
sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
```

Automatische Updates konfigurieren

RHEL/CentOS - dnf-automatic

```
# Installation
sudo dnf install dnf-automatic

# Service aktivieren
sudo systemctl enable --now dnf-automatic.timer

# Konfiguration
sudo nano /etc/dnf/automatic.conf
```

Update-Strategien

Sicherheitsupdates automatisch

- Nur kritische Sicherheitspatches
- Minimales Risiko, maximale Sicherheit
- Empfohlen für Produktionssysteme

Update-Strategien

Alle Updates automatisch

- Alle verfügbaren Updates
- Höhere Verfügbarkeit neuer Features
- Risiko von Kompatibilitätsproblemen
- Nur für nicht-kritische Systeme empfohlen

Update-Strategien

Manuelle Updates

- Vollständige Kontrolle über Updates
- Aufwändiger in der Wartung
- Nur in speziellen Fällen empfohlen

Update-Benachrichtigungen

Log-Überwachung

```
# Update-Logs anzeigen
sudo journalctl -u unattended-upgrades
sudo journalctl -u dnf-automatic

# Log-Dateien
tail -f /var/log/unattended-upgrades/unattended-upgrades.log
tail -f /var/log/dnf.log
```

Zur Kapitelübersicht

- Vorheriges Kapitel: [Shell](#)
- Nächstes Kapitel: [File System](#)