Ubuntu Server 22.04 LTS auf 24.04 LTS upgraden

Diese Anleitung beschreibt das Upgrade eines Ubuntu Servers von Ubuntu 22.04 LTS auf Ubuntu 24.04 LTS.

Stand: 2026-05-09

Ziel

Upgrade eines bestehenden Ubuntu Servers von Ubuntu 22.04 LTS auf Ubuntu 24.04 LTS.

Besonderheiten dieses Servers:

- Ubuntu Server
- SSH Zugriff
- UFW Firewall
- Apache als Reverse Proxy
- Let's Encrypt und Certbot via Snap
- Docker
- Docker Compose
- Nextcloud unter Docker

Wichtige Hinweise

Vor dem Upgrade muss ein funktionierendes Backup vorhanden sein.

Ein Release Upgrade kann Paketquellen, Dienste, Konfigurationsdateien und Abhängigkeiten verändern. Besonders bei Docker, Nextcloud, Apache, SSH und Firewall Regeln muss vorsichtig gearbeitet werden.

Das Upgrade sollte möglichst in einem Wartungsfenster durchgeführt werden.

Wenn möglich, zusätzlich zur SSH Verbindung eine Konsole des Providers, eine VM Konsole, IPMI, iLO oder eine andere Notfallkonsole bereithalten.

Das Upgrade sollte in **tmux** oder **screen** gestartet werden, damit es bei einer unterbrochenen SSH Verbindung weiterläuft.

Nicht verwenden:

```
sudo do-release-upgrade -d
```

Der Parameter -d ist für Entwicklungs oder vorgezogene Upgrade Pfade gedacht und sollte auf produktiven Servern nicht verwendet werden.

Offizielle Quellen

- Ubuntu Server Upgrade Guide: https://ubuntu.com/server/docs/how-to/software/upgrade-your-release/
- Ubuntu 24.04 LTS Release Notes: https://documentation.ubuntu.com/release-notes/24.04/
- Ubuntu 24.04 Community Release Notes: https://discourse.ubuntu.com/t/ubuntu-24-04-lts-noble-numbat-release-notes/39890

1. Aktuelle Systemversion prüfen

```
lsb_release -a
uname -r
hostnamectl
```

Erwartet vor dem Upgrade:

```
Ubuntu 22.04.x LTS Codename: jammy
```

2. Backup erstellen

Vor dem Upgrade muss ein vollständiges und geprüftes Backup vorhanden sein.

Mindestens sichern:

- /etc
- Docker Compose Verzeichnisse
- .env Dateien
- Docker Volumes
- Datenbanken
- Nextcloud Datenverzeichnis
- Apache Konfiguration
- Certbot und Let's Encrypt Konfiguration
- wichtige eigene Skripte
- Cronjobs
- UFW Regeln

2.1 /etc sichern

```
sudo tar -czf /root/backup-etc-$(date +%F).tar.gz /etc
```

2.2 Docker Compose Verzeichnisse sichern

Pfad zum Docker Compose Projekt prüfen:

```
cd /pfad/zum/docker-compose-verzeichnis pwd ls -la
```

Beispiel Backup:

```
sudo tar -czf /root/backup-docker-compose-$(date +%F).tar.gz
/pfad/zum/docker-compose-verzeichnis
```

2.3 Docker Status prüfen

```
docker ps
docker ps -a
docker volume ls
docker network ls
docker compose version
docker version
```

2.4 Nextcloud Backup prüfen

Bei Nextcloud unter Docker müssen besonders gesichert werden:

- Nextcloud Datenverzeichnis
- Nextcloud config.php
- Datenbank
- Docker Compose Datei
- .env Datei
- eventuell externe App Daten

Datenbankcontainer anzeigen:

```
docker ps
```

Beispiel für einen Datenbankdump bei MariaDB oder MySQL:

```
docker exec -it DATENBANK_CONTAINER_NAME mysqldump -u root -p --all-
databases > /root/backup-mysql-$(date +%F).sql
```

Hinweis: Containername, Datenbankbenutzer und Datenbankname müssen zur eigenen Installation passen.

3. SSH Zugriff absichern

3.1 SSH-Key-Authentifizierung prüfen:

```
grep -E "PubkeyAuthentication|AuthorizedKeysFile|PasswordAuthentication"
/etc/ssh/sshd_config
```

Wenn vorhanden:

3.1.1. SSH Konfiguration sichern:

```
sudo cp /etc/ssh/sshd_config /root/backup-sshd_config-$(date +%F)
```

3.2. SSH Dienst prüfen

```
systemctl status ssh
systemctl status ssh.socket
```

Während des Upgrades kann ein zusätzlicher SSH Port 1022 verwendet werden. Der Ubuntu Release Upgrader bietet diesen Port häufig als Notfallzugang an.

3.3. Port 1022 in UFW erlauben:

```
sudo ufw allow 1022/tcp
sudo ufw status numbered
```

SSH Status prüfen:

```
systemctl status ssh --no-pager
ss -tlnp | grep -E ':22|:1022'
```

3,4, tmux installieren bzw. prüfen ob installiert:

```
sudo apt install tmux
```

tmux starten:

```
tmux
```

Das Upgrade danach innerhalb der tmux Sitzung starten.

4. System vollständig aktualisieren

4.1. Paketlisten aktualisieren:

```
sudo apt update
```

Bestehende Ubuntu 22.04 Installation vollständig aktualisieren:

```
sudo apt dist-upgrade -o APT::Get::Always-Include-Phased-Updates=true
```

Nicht mehr benötigte Pakete entfernen:

```
sudo apt autoremove --purge
```

Falls Pakete zurückgehalten werden:

```
apt list --upgradable apt-mark showhold
```

Paketstatus prüfen:

```
sudo apt --fix-broken install sudo dpkg --audit apt-mark showhold
```

4.2. Freien Speicherplatz prüfen:

```
df -h df -h / /boot /var
```

5. Reboot vor dem Release Upgrade

5.1. Prüfen, ob ein Reboot erforderlich ist:

```
test -f /run/reboot-required && echo "Reboot erforderlich"
```

5.2. Server neu starten:

```
sudo reboot
```

Nach dem Reboot erneut anmelden und prüfen:

```
lsb_release -a uname -r systemctl --failed
```

6. Upgrade Tool vorbereiten

6.1. update-manager-core installieren bzw. prüfen ob bereits vorhanden:

```
sudo apt install update-manager-core
```

6.2. Konfiguration prüfen:

```
cat /etc/update-manager/release-upgrades
```

Dort muss stehen:

```
Prompt=lts
```

Falls nicht, Datei bearbeiten:

```
sudo nano /etc/update-manager/release-upgrades
```

Oder per Befehl setzen:

```
sudo sed -i 's/^Prompt=.*/Prompt=lts/' /etc/update-manager/release-upgrades
```

7. Drittanbieter Paketquellen prüfen

7.1. Vor dem Upgrade alle zusätzlichen Paketquellen prüfen:

```
ls -la /etc/apt/sources.list.d/ grep -R "^deb " /etc/apt/sources.list
/etc/apt/sources.list.d/
```

Typische Drittquellen:

- Docker
- Google
- Microsoft
- NodeSource
- PostgreSQL
- MariaDB
- Grafana
- sonstige PPAs

Der Release Upgrader deaktiviert Drittanbieterquellen häufig automatisch. Nach dem Upgrade müssen sie für noble wieder korrekt aktiviert werden.

7.2. Aktuellen Ubuntu Codename prüfen:

```
lsb_release -cs
```

Vor dem Upgrade sollte hier stehen:

```
jammy
```

Nach dem Upgrade sollte hier stehen:

```
noble
```

8. Optional Docker und Nextcloud vor dem Upgrade stoppen

8.1. Bei produktiven Diensten Wartungsfenster einplanen.

Docker Compose Verzeichnis öffnen:

```
cd /pfad/zum/docker-compose-verzeichnis
```

Status prüfen:

```
docker compose ps
```

Optional Container stoppen:

```
docker compose down
```

Nur stoppen, wenn ein Backup vorhanden ist und klar ist, wie die Dienste wieder gestartet werden.

9. Release Upgrade starten

9.1. Upgrade starten:

```
sudo do-release-upgrade
```

9.2. Während des Upgrades:

- Fragen aufmerksam lesen
- keine Paketentfernungen blind bestätigen
- bei Konfigurationsdateien bewusst entscheiden
- Unterschiede anzeigen lassen, wenn lokale Änderungen vorhanden sind
- bei SSH Hinweis Port 1022 offen lassen
- bei Aufforderung zum Neustart bestätigen

9.3. Typische Frage bei Konfigurationsdateien:

- lokale Version behalten
- Paketbetreuer Version installieren
- Unterschiede anzeigen

Bei manuell angepassten Diensten wie Apache, SSH, UFW, Docker oder Fail2Ban im Zweifel zuerst Unterschiede anzeigen lassen.

10. Reboot nach dem Upgrade

Nach Abschluss des Upgrades neu starten:

```
sudo reboot
```

Danach Version prüfen:

```
lsb_release -a uname -r hostnamectl
```

Erwartet:

```
Ubuntu 24.04.x LTS Codename: noble
```

11. Paketquellen nach dem Upgrade prüfen

11.1. Paketquellen prüfen:

```
sudo apt update
```

11.2. Alle Quellen kontrollieren:

```
grep -R "^deb " /etc/apt/sources.list /etc/apt/sources.list.d/
```

Es sollten keine alten jammy Quellen aktiv sein, außer bewusst und begründet.

11.3. Nach alten Einträgen suchen:

```
grep -R "jammy" /etc/apt/sources.list /etc/apt/sources.list.d/
```

11.4. Falls Docker Quelle deaktiviert wurde:

```
ls -la /etc/apt/sources.list.d/ cat /etc/apt/sources.list.d/docker.list
2>/dev/null
```

Docker Quelle muss nach dem Upgrade auf noble zeigen, nicht auf jammy.

12. System nachbereinigen

12.1. Paketstatus prüfen:

```
sudo apt update sudo apt upgrade sudo apt autoremove --purge sudo apt clean
```

12.2. Fehlerhafte Dienste anzeigen:

```
systemctl --failed
```

12.3. Fehler im aktuellen Boot anzeigen:

```
journalctl -p 3 -xb
```

12.4. Pakete mit Problemen prüfen:

```
sudo dpkg --audit sudo apt --fix-broken install
```

13. SSH nach dem Upgrade prüfen

13.1. Status prüfen:

```
systemctl status ssh --no-pager systemctl status ssh.socket --no-pager ss -tlnp | grep -E ':22|:1022'
```

Neue SSH-Verbindung in separatem Terminal testen BEVOR Port 1022 geschlossen wird

```
diff /root/backup-sshd_config-$(date +%F) /etc/ssh/sshd_config
```

Ab Ubuntu 24.04 kann SSH über systemd Socket Activation laufen. Deshalb zusätzlich ssh.socket prüfen.

13.2. Falls SSH funktioniert und Port 1022 nicht mehr benötigt wird:

```
sudo ufw delete allow 1022/tcp sudo ufw status numbered
```

14. UFW Firewall prüfen

14.1. Status anzeigen:

```
sudo ufw status verbose
```

Typische erlaubte Ports:

```
22/tcp 80/tcp 443/tcp
```

Falls temporär verwendet:

```
1022/tcp
```

Regeln prüfen:

```
sudo ufw status numbered
```

15. Apache Reverse Proxy prüfen

15.1. Apache Konfiguration testen:

```
sudo apache2ctl configtest
```

Apache Status prüfen:

```
systemctl status apache2 --no-pager
```

Apache neu laden:

```
sudo systemctl reload apache2
```

15.2. Aktive Virtual Hosts prüfen:

```
sudo apache2ctl -S
```

15.3. Aktive Module prüfen:

```
apache2ctl -M
```

Wichtige Module für Reverse Proxy:

- proxy
- proxy_http
- headers
- ssl
- rewrite

Prüfung:

```
apache2ctl -M | grep -E "proxy|ssl|rewrite|headers"
```

16. Let's Encrypt und Certbot prüfen

16.1. Snap prüfen:

```
snap list
```

16.2. Certbot Zertifikate prüfen:

```
sudo certbot certificates
```

16.3. Renewal Test durchführen:

```
sudo certbot renew --dry-run
```

16.4. Timer prüfen:

```
systemctl list-timers | grep -i certbot systemctl list-timers | grep -i
snap
```

17. Docker nach dem Upgrade prüfen

17.1. Docker Version prüfen:

```
docker version docker compose version
```

17.2. Docker Dienst prüfen:

```
systemctl status docker --no-pager
```

17.3. Container prüfen:

```
docker ps docker ps -a
```

Falls Container vor dem Upgrade gestoppt wurden:

```
cd /pfad/zum/docker-compose-verzeichnis docker compose up -d
```

17.4. Logs prüfen:

```
docker compose logs --tail=100
```

18. Nextcloud nach dem Upgrade prüfen

18.1. Docker Container prüfen:

```
docker ps
```

18.2. Nextcloud Logs prüfen:

```
docker logs --tail=100 NEXTCLOUD_CONTAINER_NAME
```

18.3. Nextcloud Status per occ prüfen:

```
docker exec -u www-data -it NEXTCLOUD_CONTAINER_NAME php occ status
```

18.4. Maintenance Mode prüfen:

```
docker exec -u www-data -it NEXTCLOUD_CONTAINER_NAME php occ
maintenance:mode
```

18.5. Core Integrität prüfen:

```
docker exec -u www-data -it NEXTCLOUD_CONTAINER_NAME php occ
integrity:check-core
```

Hinweis: Containername und Benutzer können je nach Image abweichen.

19. Netzwerk prüfen

19.1. IP Adressen prüfen:

```
ip addr ip route
```

19.2. DNS prüfen:

```
resolvectl status
```

19.3. Erreichbarkeit prüfen:

```
ping -c 4 1.1.1.1 ping -c 4 ubuntu.com
```

19.4. Offene Ports prüfen:

```
ss -tulpen
```

20. Dienste prüfen

Alle fehlgeschlagenen Dienste anzeigen:

```
systemctl --failed
```

Wichtige Dienste einzeln prüfen:

```
systemctl status ssh --no-pager systemctl status apache2 --no-pager
systemctl status docker --no-pager systemctl status ufw --no-pager
```

Logs prüfen:

```
journalctl -p 3 -xb journalctl -u apache2 -n 100 --no-pager journalctl -u
docker -n 100 --no-pager journalctl -u ssh -n 100 --no-pager
```

21. Alte Kernel und Pakete prüfen

Installierte Kernel anzeigen:

```
dpkg -l | grep linux-image
```

Nicht mehr benötigte Pakete entfernen:

```
sudo apt autoremove --purge
```

System bereinigen:

```
sudo apt clean
```

22. Abschlussprüfung

Folgende Punkte müssen erfolgreich sein:

- Server startet ohne Fehler
- SSH funktioniert
- UFW Regeln passen
- Apache Konfiguration ist gültig
- HTTPS Zertifikate funktionieren
- Docker läuft
- Nextcloud ist erreichbar
- keine fehlgeschlagenen systemd Dienste
- keine alten jammy Paketquellen aktiv
- apt update läuft fehlerfrei
- journalctl zeigt keine kritischen neuen Fehler

Befehle:

```
 lsb_release -a uname -r sudo apt update systemctl --failed journalctl -p 3 -xb sudo apache2ctl configtest sudo certbot renew --dry-run docker ps sudo ufw status verbose
```

23. Temporäre Freigaben entfernen

Falls Port 1022 nur für das Upgrade geöffnet wurde:

```
sudo ufw delete allow 1022/tcp sudo ufw status numbered
```

24. Rollback

Falls das Upgrade fehlschlägt:

- nicht weiter blind reparieren
- Fehlermeldungen sichern
- Logs sichern
- Backup prüfen
- bei VM Snapshot zurückspielen
- bei Bare Metal Restore aus Backup planen

Wichtige Logs:

```
/var/log/dist-upgrade/ /var/log/apt/ /var/log/syslog
```

Anzeigen:

```
ls -la /var/log/dist-upgrade/ less /var/log/dist-upgrade/main.log less
```

```
/var/log/dist-upgrade/apt.log
```

25. Kurzfassung

```
 sudo ufw allow 1022/tcp sudo apt update sudo apt dist-upgrade -o
APT::Get::Always-Include-Phased-Updates=true sudo apt autoremove --purge
sudo reboot

sudo apt install update-manager-core \\ cat /etc/update-manager/release-
upgrades

sudo do-release-upgrade

sudo reboot

lsb_release -a \\ sudo apt update \\ systemctl --failed \\ journalctl -p 3 -
xb \\ sudo apache2ctl configtest \\ sudo certbot renew --dry-run \\ docker
ps \\ sudo ufw status verbose \\
```

Ergebnis

Nach erfolgreichem Abschluss läuft der Server auf:

```
Ubuntu 24.04 LTS Noble Numbat
```

Alle produktiven Dienste wurden geprüft.

From:
http://wiki.waldhofer.at/ - **Wiki von Franz**

Permanent link:
http://wiki.waldhofer.at/doku.php?id=ubuntu:ubuntu_24.04_1&rev=1782647271

Last update: **2026/06/28 13:47**