

Ubuntu Server 22.04 LTS auf 24.04 LTS upgraden

Diese Anleitung beschreibt das Upgrade eines Ubuntu Servers von **Ubuntu 22.04 LTS (Jammy Jellyfish)** auf **Ubuntu 24.04 LTS (Noble Numbat)**.

Stand: 2026-05-09

Ziel

Upgrade eines bestehenden Ubuntu Servers von:

- * Ubuntu 22.04 LTS
- * auf Ubuntu 24.04 LTS

mit besonderer Beachtung von:

- * SSH-Zugriff
- * UFW Firewall
- * Apache Reverse Proxy
- * Let's Encrypt / Certbot via Snap
- * Docker / Docker Compose
- * Nextcloud unter Docker
- * Paketquellen und Drittanbieter-Repositories

Wichtige Hinweise

Nicht ohne Backup starten.

Ein Release-Upgrade kann Dienste, Paketquellen, Konfigurationsdateien und Abhängigkeiten verändern. Besonders bei Servern mit Docker, Reverse Proxy, Datenbanken oder Nextcloud muss vorher ein vollständiges Backup vorhanden sein.

Nicht über eine einzige unsichere SSH-Verbindung arbeiten.

Am besten:

- * über Konsole / VM-Konsole / iLO / IPMI / Provider-Konsole arbeiten
- * zusätzlich SSH absichern
- * Upgrade in tmux oder screen starten

Nicht verwenden:

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

Der Parameter -d ist für Entwicklungs- bzw. 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\\
```

2. Backup erstellen

Vor dem Upgrade muss ein funktionierendes Backup vorhanden sein.

2.1 Wichtige Verzeichnisse sichern

Mindestens sichern:

- * /etc
- * Docker-Compose-Verzeichnisse
- * .env Dateien
- * Docker Volumes
- * Datenbanken
- * Nextcloud-Datenverzeichnis
- * Apache-Konfiguration
- * Certbot-/Let's-Encrypt-Konfiguration

Beispiele:

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

Falls Docker Compose verwendet wird, das jeweilige Compose-Verzeichnis sichern:

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

Beispiel:

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

2.2 Docker / Nextcloud Backup prüfen

Laufende Container anzeigen:

```
\\  
docker ps\\  
docker compose version\\  
docker version\\
```

Docker Volumes anzeigen:

```
\\  
docker volume ls\\
```

Bei Nextcloud besonders wichtig:

- * Nextcloud-Konfigurationsdateien
- * Nextcloud-Datenverzeichnis
- * Datenbankdump
- * Docker Compose Datei
- * .env Datei

Beispiel für MariaDB/MySQL-Dump im Container:

```
\\  
docker ps\\
```

Dann passenden Datenbankcontainer ermitteln und Dump erstellen, Beispiel:

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

Achtung: Containername, Benutzer und Datenbankname müssen zur eigenen Umgebung passen.

3. SSH-Zugriff absichern

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

Firewall vorher öffnen:

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

Aktuellen SSH-Zugriff prüfen:

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

Zusätzlich empfohlen:

```
\\  
sudo apt install tmux\\  
tmux\\
```

Das Upgrade anschließend innerhalb der tmux-Sitzung starten.

4. System vollständig aktualisieren

Paketlisten aktualisieren:

```
\\  
sudo apt update\\
```

Vollständiges Upgrade der bestehenden Ubuntu-22.04-Installation durchführen:

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

Nicht mehr benötigte Pakete entfernen:

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

Paketstatus prüfen:

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

```
apt-mark showhold\\
```

Freien Speicherplatz prüfen:

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

Falls Pakete zurückgehalten werden, zuerst prüfen und lösen:

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

5. Reboot vor dem Release-Upgrade

Prüfen, ob ein Reboot erforderlich ist:

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

Server neu starten:

```
\\  
sudo reboot\\
```

Nach dem Reboot erneut anmelden und prüfen:

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

6. Upgrade-Tool vorbereiten

Paket installieren:

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

Konfiguration prüfen:

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

Dort muss stehen:

```
\\  
Prompt=lts\\
```

Falls nicht, ändern:

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

Oder per Befehl:

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

7. Drittanbieter-Quellen prüfen

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.

Aktuellen Ubuntu-Codename prüfen:

```
\\  
lsb_release -cs\\
```

Vor dem Upgrade sollte hier stehen:

```
\\  
jammy\\
```

Nach dem Upgrade sollte hier stehen:

```
\\
```

```
noble\\
```

8. Optional: Docker/Nextcloud vor dem Upgrade stoppen

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\\
```

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

9. Release-Upgrade starten

Upgrade starten:

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

Während des Upgrades:

- * Fragen aufmerksam lesen
- * keine Paketentfernungen blind bestätigen
- * bei Konfigurationsdateien bewusst entscheiden
- * bei SSH-Hinweis Port 1022 offen lassen
- * bei Aufforderung zum Neustart bestätigen

Typische Frage bei Konfigurationsdateien:

- * lokale Version behalten
- * Paketbetreuer-Version installieren
- * Unterschiede anzeigen

Bei manuell angepassten Serverdiensten 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

Paketquellen prüfen:

```
\\  
sudo apt update\\
```

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.

Suchen nach alten Einträgen:

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

Falls Docker-Quelle deaktiviert wurde, prüfen:

```
\\  
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

Paketstatus prüfen:

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

Fehlerhafte Dienste anzeigen:

```
\  
systemctl --failed
```

Fehler im aktuellen Boot anzeigen:

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

Pakete mit Problemen prüfen:

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

13. SSH nach dem Upgrade prüfen

Status prüfen:

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

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

Falls SSH funktioniert und Port 1022 nicht mehr benötigt wird:

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

14. UFW Firewall prüfen

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

Apache-Konfiguration testen:

```
\\  
sudo apache2ctl configtest\\
```

Apache-Status prüfen:

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

Apache neu laden:

```
\\  
sudo systemctl reload apache2\\
```

Aktive VirtualHosts prüfen:

```
\\  
sudo apache2ctl -S\\
```

Aktive Module prüfen:

```
\\  
apache2ctl -M\\
```

Wichtige Module für Reverse Proxy:

```
* proxy  
* proxy_http  
* headers  
* ssl  
* rewrite
```

Beispielprüfung:

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

16. Let's Encrypt / Certbot prüfen

Snap prüfen:

```
\\  
snap list\\
```

Certbot-Zertifikate prüfen:

```
\\  
sudo certbot certificates\\
```

Renewal-Test durchführen:

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

Timer prüfen:

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

17. Docker nach dem Upgrade prüfen

Docker-Version prüfen:

```
\\  
docker version\\  
docker compose version\\
```

Docker-Dienst prüfen:

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

Container prüfen:

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

Falls Container vor dem Upgrade gestoppt wurden:

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

Logs prüfen:

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

18. Nextcloud nach dem Upgrade prüfen

Docker-Container prüfen:

```
\\  
docker ps\\
```

Nextcloud-Logs prüfen, Beispiel:

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

Falls Nextcloud OCC benötigt wird, Beispiel:

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

Weitere sinnvolle Prüfungen:

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

IP-Adressen prüfen:

```
\\  
ip addr\\  
ip route\\
```

DNS prüfen:

```
<code bash>  
resolvectl status
```

From:

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

Permanent link:

http://wiki.waldhofer.at/doku.php?id=ubuntu:ubuntu_24.04_1&rev=1778312985

Last update: **2026/05/09 09:49**

