Nextcloud Talk – Hochleistungs-Backend mit Docker installieren und aktualisieren

1. Zielsetzung

Diese Anleitung beschreibt die Installation, Konfiguration, Wartung und Aktualisierung eines eigenständigen Hochleistungs-Backends (HPB) für Nextcloud Talk.

Das Backend wird unabhängig von Nextcloud All-in-One als eigenes Docker-Compose-Projekt betrieben. Verwendet wird das offizielle AIO-Talk-Containerimage.

Technische Komponenten:

- Nextcloud Server mit Talk-App
- Docker Engine und Docker Compose v2
- Nextcloud Talk Hochleistungs-Backend
- Signaling-Server
- Janus WebRTC-Medienserver
- Integrierter TURN-Server
- Apache2 als HTTPS-Reverse-Proxy
- Let's Encrypt für TLS-Zertifikate

Wichtig: Die Nextcloud-App `notify_push` ist nicht das Talk-Hochleistungs-Backend. Sie beschleunigt Benachrichtigungen, ersetzt jedoch weder den Signaling-Server noch den Medienserver.

2. Beispielumgebung

Komponente	Beispielwert
Betriebssystem	Ubuntu Server LTS
Docker	Docker Engine mit Compose v2
Nextcloud-Domain	cloud.example.org
Talk-Domain	talk.example.org
Nextcloud-Projekt	/opt/docker/nextcloud
HPB-Projekt	/opt/docker/talk-hpb
Backup-Verzeichnis	/opt/backups
Reverse-Proxy	Apache2
HTTPS	443/TCP
Signaling intern	8081/TCP
TURN	3478/TCP und UDP

Alle Domains und Verzeichnisse müssen an die eigene Umgebung angepasst werden.

Der Signaling-Port wird ausschließlich an die lokale Loopback-Adresse gebunden. Der TURN-Port muss für externe Teilnehmer erreichbar sein.

3. Voraussetzungen

3.1 Software

Erforderlich sind:

- Funktionierende Nextcloud-Installation
- Installierte und aktivierte Nextcloud Talk-App
- Docker Engine
- Docker Compose v2
- Apache2 mit HTTPS-Unterstützung
- Gültiges TLS-Zertifikat
- DNS-Eintrag für die Talk-Domain

3.2 Docker prüfen

```
docker --version
docker compose version
docker info
```

3.3 Apache-Module prüfen

```
sudo apache2ctl -M | grep -E 'proxy|ssl|headers|rewrite'
```

Erforderliche Module aktivieren:

```
sudo a2enmod proxy proxy_http proxy_wstunnel ssl headers rewrite
sudo apache2ctl configtest
```

3.4 DNS prüfen

```
dig +short cloud.example.org
dig +short talk.example.org
```

Beide Domains müssen korrekt aufgelöst werden. Die Talk-Domain muss auf den öffentlich erreichbaren Reverse-Proxy zeigen.

3.5 Ports prüfen

```
sudo ss -tuln | grep -E ':(3478|8081)\b'
```

Vor der Installation sicherstellen, dass die benötigten Hostports nicht bereits belegt sind.

4. Verzeichnisstruktur erstellen

```
sudo mkdir -p /opt/docker/talk-hpb/secrets
sudo mkdir -p /opt/backups

sudo chmod 700 /opt/docker/talk-hpb/secrets
sudo chmod 700 /opt/backups

cd /opt/docker/talk-hpb
```

Die weiteren Befehle werden mit einem Benutzer ausgeführt, der die erforderlichen Berechtigungen besitzt.

5. Zugangsschlüssel erzeugen

Das Hochleistungs-Backend benötigt drei unabhängige Schlüssel.

Schlüssel	Zweck
TURN_SECRET	Authentifizierung der TURN-Verbindungen
SIGNALING_SECRET	Kommunikation zwischen Nextcloud und Signaling
INTERNAL_SECRET	Interne Authentifizierung der HPB-Komponenten

Die Schlüssel werden bei der Erstinstallation erzeugt.

```
cd /opt/docker/talk-hpb

umask 077

for name in turn signaling internal; do
    file="secrets/${name}_secret"
    if [ ! -e "$file" ]; then
        openssl rand -hex 32 > "$file"
    fi
done

chmod 700 secrets
chmod 600 secrets/*
```

Kontrolle:

```
ls -l secrets/
```

Wichtig:

- Bei Updates keine neuen Secrets generieren.
- Vorhandene Secrets unverändert weiterverwenden.
- Secrets nicht in Repositories oder öffentlichen Dokumentationen veröffentlichen.

6. Umgebungsdatei erstellen

Die Datei `.env` enthält die Konfigurationsvariablen.

```
cd /opt/docker/talk-hpb

umask 077

{
    echo "NC_DOMAIN=cloud.example.org"
    echo "TALK_HOST=talk.example.org"
    echo "TALK_PORT=3478"
    echo "TZ=Europe/Vienna"
    echo "TURN_SECRET=$(cat secrets/turn_secret)"
    echo "SIGNALING_SECRET=$(cat secrets/signaling_secret)"
    echo "INTERNAL_SECRET=$(cat secrets/internal_secret)"
} > .env

chmod 600 .env
```

Achtung: Diesen Block ausschließlich bei einer Erstinstallation verwenden.

Bei bestehenden Installationen die `.env` nicht überschreiben. Zuerst eine Sicherung erstellen.

7. Docker Compose konfigurieren

Datei: `/opt/docker/talk-hpb/compose.yaml`

```
services:
  talk-hpb:
    image: ghcr.io/nextcloud-releases/aio-talk:latest
    container_name: nextcloud-talk-hpb
    restart: unless-stopped
    init: true

    env_file:
      - .env

    ports:
      - "127.0.0.1:8081:8081/tcp"
      - "3478:3478/tcp"
      - "3478:3478/udp"
```

Erläuterung:

- 8081/TCP ist ausschließlich über localhost erreichbar.
- 3478/TCP und UDP werden für den integrierten TURN-Server bereitgestellt.
- Die Secrets werden aus der geschützten `.env` geladen.

- Der Container startet nach einem Serverneustart automatisch neu.

Der Image-Tag `latest` wird hier für die Erstinstallation verwendet.

Für den langfristigen Produktionsbetrieb sollte nach erfolgreicher Prüfung ein festgelegter Image-Digest verwendet werden. Damit lassen sich Updates kontrolliert durchführen.

7.1 Docker-Konfiguration prüfen

```
cd /opt/docker/talk-hpb

docker compose config --quiet
```

Bei korrekter Syntax erscheint keine Fehlermeldung.

Die normale Ausgabe von `docker compose config` kann Secrets enthalten. Für die Syntaxprüfung daher `–quiet` verwenden.

7.2 Container starten

```
docker compose pull
docker compose up -d
```

7.3 Containerstatus prüfen

```
docker compose ps
```

Der Container muss laufen.

Falls das verwendete Image einen Healthcheck bereitstellt, sollte der Zustand `healthy` erreicht werden.

8. Signaling-Server lokal testen

```
curl -i http://127.0.0.1:8081/api/v1/welcome
```

Erwartet wird HTTP 200 und eine JSON-Antwort ähnlich:

```
{
  "nextcloud-spreed-signaling": "Welcome",
  "version": "..."
}
```

Bei Problemen:

```
docker logs --tail 100 nextcloud-talk-hpb
```

9. Apache-Reverse-Proxy einrichten

Der Signaling-Server wird unter einer eigenen HTTPS-Domain veröffentlicht.

Beispiel:

https://talk.example.org

9.1 TLS-Zertifikat

Mit Certbot kann ein Zertifikat beispielsweise so beantragt werden:

```
sudo certbot certonly --apache -d talk.example.org
```

Alternativ kann ein bereits vorhandenes gültiges Wildcard-Zertifikat verwendet werden.

9.2 VirtualHost erstellen

Datei: `/etc/apache2/sites-available/talk.example.org.conf`

```
<VirtualHost *:80>
    ServerName talk.example.org
    Redirect permanent / https://talk.example.org/
</VirtualHost>

<VirtualHost *:443>
    ServerName talk.example.org

    SSLEngine on

    SSLCertificateFile /etc/letsencrypt/live/talk.example.org/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/talk.example.org/privkey.pem

    ProxyPreserveHost On
    ProxyRequests Off

    RequestHeader set X-Forwarded-Proto "https"

    ProxyPass / http://127.0.0.1:8081/ upgrade=websocket
    ProxyPassReverse / http://127.0.0.1:8081/

    ErrorLog ${APACHE_LOG_DIR}/talk_error.log
    CustomLog ${APACHE_LOG_DIR}/talk_access.log combined
</VirtualHost>
```

Diese Konfiguration setzt eine Apache-Version mit entsprechender WebSocket-Upgrade-Unterstützung über `mod_proxy_http` voraus.

Die Zertifikatspfade müssen an die tatsächliche Installation angepasst werden.

9.3 Website aktivieren

```
sudo a2ensite talk.example.org.conf
sudo apache2ctl configtest
sudo systemctl reload apache2
```

9.4 HTTPS testen

```
curl -i https://talk.example.org/api/v1/welcome
```

Erwartetes Ergebnis: HTTP 200.

10. WebSocket-Verbindung testen

```
curl --http1.1 -i -N --max-time 5 \
  -H "Connection: Upgrade" \
  -H "Upgrade: websocket" \
  -H "Sec-WebSocket-Version: 13" \
  -H "Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==" \
  https://talk.example.org/spreed
```

Erwartetes Ergebnis:

```
HTTP/1.1 101 Switching Protocols
```

Eine später auftretende Meldung `hello_timeout` ist bei diesem einfachen Test normalerweise unkritisch, da curl nicht das vollständige Signaling-Protokoll ausführt.

11. Hochleistungs-Backend in Nextcloud eintragen

Als Administrator bei Nextcloud anmelden.

https://cloud.example.org/settings/admin/talk

Den Abschnitt **Hochleistungs-Backend** öffnen.

Einen neuen Backend-Server hinzufügen.

Einstellung	Wert
Backend-URL	https://talk.example.org

Einstellung	Wert
Gemeinsames Geheimnis	Inhalt von signaling_secret
SSL-Zertifikat überprüfen	Aktiviert

Den Signaling-Schlüssel lokal anzeigen:

```
cat /opt/docker/talk-hpb/secrets/signaling_secret
```

Den Schlüssel in Nextcloud eintragen und die Verbindung prüfen.

Erwartetes Ergebnis:

- Nextcloud erkennt das Hochleistungs-Backend.
- Die Verbindung ist erfolgreich.
- Die verfügbaren Signaling-Funktionen werden angezeigt.

Bei Warnungen über fehlende optionale Funktionen zuerst die Versionskompatibilität prüfen.

Ein funktionierendes Backend nicht ohne vorherige Prüfung durch eine ungeprüfte Entwicklungsversion ersetzen.

12. TURN-Server einrichten

Das verwendete AIO-Talk-Image enthält bereits einen TURN-Server.

Ein zusätzlicher coturn-Container ist für diese Installationsvariante nicht erforderlich.

12.1 Firewall konfigurieren

Bei Verwendung von UFW:

```
sudo ufw allow 3478/tcp
sudo ufw allow 3478/udp
sudo ufw status
```

Die HTTPS-Verbindung auf Port 443 muss ebenfalls erreichbar sein.

12.2 Router-Portweiterleitung

Bei einem Server hinter NAT müssen folgende Freigaben eingerichtet werden:

Protokoll	Externer Port	Internes Ziel
TCP	3478	Docker-Host, Port 3478
UDP	3478	Docker-Host, Port 3478

Beispiel FRITZ!Box:

1. Benutzeroberfläche öffnen.
2. Internet → Freigaben → Portfreigaben aufrufen.
3. Den Docker-Host auswählen.
4. Neue Portfreigabe für UDP 3478 erstellen.
5. Zweite Portfreigabe für TCP 3478 erstellen.
6. Einstellungen übernehmen.

Die vorhandene HTTPS-Portfreigabe für 443 bleibt unverändert.

Port 8081 darf nicht öffentlich freigegeben werden.

Bei vorhandenem IPv6 müssen die entsprechenden Firewallfreigaben separat geprüft werden.

12.3 TURN in Nextcloud konfigurieren

In Nextcloud:

Administration → Talk → TURN-Server

Einstellung	Wert
Protokoll	turn:
Server	talk.example.org:3478
Gemeinsames Geheimnis	Inhalt von turn_secret
Transport	UDP und TCP

TURN-Schlüssel anzeigen:

```
cat /opt/docker/talk-hpb/secrets/turn_secret
```

Konfiguration speichern und über die Testfunktion prüfen.

Ein grüner Haken bestätigt den erfolgreichen Konfigurationstest. Er ersetzt jedoch keinen praktischen Verbindungstest.

12.4 STUN konfigurieren

Optional kann der integrierte STUN-Dienst verwendet werden.

Beispiel:

`talk.example.org:3478`

Ein bereits funktionierender STUN-Server muss nicht zwingend ersetzt werden.

13. Gesamtsystem prüfen

13.1 Containerstatus

```
cd /opt/docker/talk-hpb

docker compose ps
```

13.2 Öffentlichen Signaling-Endpunkt prüfen

```
curl -fsS https://talk.example.org/api/v1/welcome
```

13.3 Verbindung aus Nextcloud prüfen

Bei einem Nextcloud-Docker-Projekt mit dem Service app:

```
cd /opt/docker/nextcloud

docker compose exec -T app \
  curl -fsS https://talk.example.org/api/v1/welcome
```

13.4 TURN-Port prüfen

```
sudo ss -tuln | grep ':3478'
```

Auch die Docker-Portzuordnung prüfen:

```
docker ps --format 'table {{.Names}}\t{{.Ports}}'
```

13.5 Praxistest

1. Eine Testunterhaltung in Nextcloud Talk erstellen.
2. Einen Videoanruf mit zwei Endgeräten starten.
3. Mindestens ein Gerät über ein externes Netzwerk verbinden.
4. Audio in beide Richtungen prüfen.
5. Video in beide Richtungen prüfen.
6. Anschließend einen Anruf mit mindestens drei Teilnehmern durchführen.

Der Test sollte auch mit unterschiedlichen Netzwerken erfolgen, beispielsweise LAN und Mobilfunk.

14. Updatekonzept

Nextcloud Talk besteht aus mehreren unabhängig aktualisierbaren Komponenten.

Komponente	Aktualisierung
Nextcloud Server	Nextcloud-Docker-Image
Nextcloud Talk-App	App-Update in Nextcloud
Talk HPB	AIO-Talk-Docker-Image
TURN	Bestandteil des HPB-Images
Apache2	Paketverwaltung des Betriebssystems

Empfohlene Grundregeln:

- Sicherheitsupdates priorisieren.
- Vor Updates die Release Notes lesen.
- Versionskompatibilität kontrollieren.
- Vor Updates Backups erstellen.
- Während laufender Konferenzen keine Updates durchführen.
- Änderungen zuerst in einer Testumgebung prüfen.
- Nach jedem Update einen Funktionstest durchführen.

14.1 Kompatibilität prüfen

Vor einem Upgrade kontrollieren:

- Unterstützt die Talk-App die installierte Nextcloud-Version?
- Unterstützt das HPB die benötigten Signaling-Funktionen?
- Gibt es bekannte Probleme mit dem neuen Image?
- Sind die erforderlichen Docker- und Apache-Versionen vorhanden?
- Sind weitere Nextcloud-Apps mit dem geplanten Serverupgrade kompatibel?

Nextcloud-Major-Versionen nicht überspringen.

15. HPB vor einem Update sichern

15.1 Konfiguration sichern

```
cd /opt/docker/talk-hpb

umask 077

BACKUP="/opt/backups/talk-hpb-$(date +%Y%m%d-%H%M).tar.gz"

tar -czf "$BACKUP" \
  compose.yaml \
  .env \
  secrets

chmod 600 "$BACKUP"
```

Diese Sicherung enthält vertrauliche Schlüssel und muss besonders geschützt aufbewahrt werden.

15.2 Aktuelles Docker-Image dokumentieren

```
docker inspect nextcloud-talk-hpb \
  --format '{{.Image}}'
```

Die Ausgabe enthält die ID des tatsächlich verwendeten Images.

Zusätzlich das konfigurierte Image anzeigen:

```
docker inspect nextcloud-talk-hpb \
  --format '{{.Config.Image}}'
```

15.3 Aktuelles Image für Rollback sichern

```
cd /opt/docker/talk-hpb

umask 077

docker image save \
  -o /opt/backups/talk-hpb-image-before-update.tar \
  ghcr.io/nextcloud-releases/aio-talk:latest

chmod 600 /opt/backups/talk-hpb-image-before-update.tar
```

Dieser Befehl setzt voraus, dass der bisherige Container tatsächlich mit dem lokal vorhandenen `latest`-Image betrieben wird. Bei festgelegten Tags oder Digests muss stattdessen genau dieses Image gesichert werden.

16. Talk-HPB aktualisieren

16.1 Neue Version prüfen

Vor dem Update die offiziellen Veröffentlichungen kontrollieren.

- https://github.com/nextcloud/all-in-one
- https://github.com/nextcloud/spreed
- https://nextcloud-talk.readthedocs.io/

16.2 Image herunterladen

```
cd /opt/docker/talk-hpb

docker compose pull
```

16.3 Container aktualisieren

```
docker compose up -d
```

Docker Compose erstellt den Container bei geändertem Image neu.

Die Dateien `.env` und `compose.yaml` bleiben dabei unverändert.

16.4 Update prüfen

```
docker compose ps
```

Anschließend:

```
docker logs --tail 100 nextcloud-talk-hpb
```

Den öffentlichen Endpunkt testen:

```
curl -fsS https://talk.example.org/api/v1/welcome
```

Abschließend in Nextcloud Administration → Talk die Verbindung prüfen und einen Videoanruf durchführen.

17. Rollback bei Problemen

Wenn ein neues Image Probleme verursacht, kann auf das vorherige Image zurückgewechselt werden.

Voraussetzung ist, dass das vorherige Image beziehungsweise dessen Digest bekannt und verfügbar ist.

17.1 Vorheriges Image laden

Wenn zuvor eine Image-Sicherung angelegt wurde:

```
docker image load \
  -i /opt/backups/talk-hpb-image-before-update.tar
```

17.2 Compose-Konfiguration zurücksetzen

In `compose.yaml` den vorher verwendeten Image-Tag beziehungsweise Digest eintragen.

Keinen unveränderten `latest`-Tag verwenden, wenn dieser inzwischen auf das neue, fehlerhafte Image verweist.

17.3 Container neu erstellen

```
cd /opt/docker/talk-hpb

docker compose up -d --force-recreate
```

17.4 Funktion prüfen

```
docker compose ps

curl -fsS https://talk.example.org/api/v1/welcome
```

Danach die Verbindung in Nextcloud Talk erneut testen.

18. Nextcloud Server und Talk-App aktualisieren

Ein Nextcloud-Serverupgrade ist unabhängig vom HPB-Update durchzuführen.

Empfohlene Reihenfolge:

1. Kompatibilität und Release Notes prüfen.
2. Datenbank und Nextcloud-Daten sichern.
3. Nextcloud-Serverimage aktualisieren.
4. Datenbankmigration und Nextcloud-Upgrade durchführen.
5. Nextcloud-Apps einschließlich Talk aktualisieren.
6. Administrationsübersicht prüfen.
7. HPB-Verbindung kontrollieren.
8. TURN-Konfiguration prüfen.
9. Testanruf durchführen.

Bei einer Docker-Installation erfolgt das eigentliche Serverupgrade nach dem Update des Images entsprechend der verwendeten Nextcloud-Docker-Konfiguration.

Die erforderlichen `occ`-Befehle hängen vom konkreten Deployment ab.

19. Fehleranalyse

19.1 Containerprotokolle

```
docker logs --tail 200 nextcloud-talk-hpb
```

19.2 Nur Fehler und Warnungen anzeigen

```
docker logs --since 30m nextcloud-talk-hpb 2>&1 \
```

```
  | grep -Ei 'error|warn|failed|timeout|unauthorized|forbidden'
```

19.3 Apache prüfen

```
sudo apache2ctl configtest

sudo tail -n 100 /var/log/apache2/talk_error.log
```

19.4 Docker-Portzuordnung prüfen

```
docker port nextcloud-talk-hpb
```

19.5 Häufige Fehler

Symptom	Mögliche Ursache
HTTP 502	Container nicht erreichbar oder falscher Proxy-Port
HTTP 404	Falscher Proxy-Pfad
WebSocket-Verbindung schlägt fehl	Proxy-Konfiguration oder Upgrade-Unterstützung fehlerhaft
Nextcloud erkennt HPB nicht	Backend-URL, DNS, TLS oder Signaling-Secret falsch
TURN-Test schlägt fehl	Portweiterleitung, Firewall oder TURN-Secret falsch
Video funktioniert nur intern	TURN-Erreichbarkeit oder NAT-Konfiguration fehlerhaft
Container startet nicht	Fehlerhafte Umgebungsvariablen oder Portkonflikt

20. Sicherheitsempfehlungen

- Port 8081 ausschließlich lokal verfügbar machen.
- Port 3478 nur für die erforderlichen Protokolle freigeben.
- HTTPS mit gültigen Zertifikaten verwenden.
- Secrets mit restriktiven Dateiberechtigungen schützen.
- Docker-Images regelmäßig auf Sicherheitsupdates prüfen.
- Vor Änderungen Sicherungen erstellen.
- Nicht mehr benötigte Container und Images kontrolliert entfernen.
- Apache- und Docker-Protokolle regelmäßig prüfen.
- Keine Secrets oder vollständigen Umgebungsdateien in Tickets oder Wikis veröffentlichen.

21. Regelmäßige Wartung

Intervall	Tätigkeit
Wöchentlich	Containerstatus und relevante Logs prüfen
Wöchentlich	Sicherheitsmeldungen kontrollieren
Monatlich	Release Notes und verfügbare HPB-Updates prüfen
Monatlich	Backup und Wiederherstellbarkeit kontrollieren
Nach jedem Update	Signaling-Verbindung und Testanruf prüfen
Nach Netzwerkänderungen	TURN und externe Erreichbarkeit testen

22. Offizielle Dokumentation

- Nextcloud Talk – Quick Install
- Nextcloud All-in-One Repository
- Nextcloud Talk Repository
- Nextcloud Signaling Server
- Docker Compose Dokumentation

23. Zusammenfassung

Das Nextcloud Talk Hochleistungs-Backend wird als eigenständiges Docker-Compose-Projekt betrieben.

Die Trennung von Nextcloud Server, Talk-App und HPB ermöglicht eine kontrollierte Wartung und Aktualisierung der Komponenten.

Für einen zuverlässigen Betrieb sind insbesondere folgende Punkte entscheidend:

- Korrekte Signaling-Konfiguration
- Funktionierender Apache-Reverse-Proxy
- Erreichbarer TURN-Server
- Korrekte Portfreigaben
- Sichere Aufbewahrung der Zugangsschlüssel
- Regelmäßige Updates und Backups
- Praktische Tests mit mehreren Teilnehmern

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

Permanent link:
http://wiki.waldhofer.at/doku.php?id=nextcloud:talk

Last update: **2026/10/30 09:38**