= Nextcloud Talk – Hochleistungs-Backend mit Docker installieren und aktualisieren ====== ===== 1. Zielsetzung ===== Diese Anleitung beschreibt die Installation und Wartung eines eigenständigen Hochleistungs-Backends (HPB) für Nextcloud Talk. Das HPB verbessert die Skalierbarkeit von Audio- und Videokonferenzen mit mehreren Teilnehmern. Die Installation erfolgt unabhängig von Nextcloud All-in-One mit dem offiziellen AIO-Talk-Docker-Image. **Technische Komponenten: * Nextcloud Server mit installierter Talk-App * Docker Engine und Docker Compose * Nextcloud Talk Hochleistungs-Backend * Signaling-Server * Janus WebRTC-Medienserver * TURN-Server * Apache2 als HTTPS-Reverse-Proxy * Let's Encrypt für TLS-Zertifikate Das Backend wird als eigenständiges Docker-Compose-Projekt betrieben. Das Nextcloud-Programm ''notify_push'' ist nicht das Talk-Hochleistungs-Backend. Es beschleunigt Benachrichtigungen, ersetzt aber weder den Signaling-Server noch den WebRTC-Medienserver. ===== 2. Beispielumgebung ===== ^ Komponente ^ Beispielwert ^ | Betriebssystem | Ubuntu Server LTS | | Docker | Docker Engine mit Compose v2 | | Nextcloud | https://cloud.example.org | | Talk-HPB | https://talk.example.org | | Nextcloud-Projekt | /opt/docker/nextcloud | | HPB-Projekt | /opt/docker/talk-hpb | | Backup-Verzeichnis | /opt/backups | | Reverse-Proxy | Apache2 | | HTTPS-Port | 443/TCP | | Signaling-Port intern | 8081/TCP | | TURN-Port | 3478/TCP und UDP | Die Domains, Verzeichnisse und Serveradressen sind an die eigene Umgebung anzupassen. Der lokale Signaling-Port wird ausschließlich an 127.0.0.1 gebunden. ===== 3. Voraussetzungen ===== ==== 3.1 Software ====**

Erforderlich sind: * Funktionierende Nextcloud-Installation * Nextcloud Talk App * Docker Engine * Docker Compose v2 * Apache2 mit TLS-Unterstützung * Gültiges TLS-Zertifikat * DNS-Eintrag für den Talk-Server Bei einer Installation mit mehreren Teilnehmern müssen zusätzlich ausreichende CPU-Leistung, RAM und Netzwerkbandbreite verfügbar sein. ==== 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' '' Benötigt werden insbesondere: * proxy * proxy_http * ssl * headers Für ältere Apache-Konfigurationen kann auch proxy_wstunnel erforderlich sein. Fehlende 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 '' Die Talk-Domain muss auf die öffentliche Adresse des entsprechenden Reverse-Proxys zeigen. ==== 3.5 Ports prüfen ==== '' sudo ss -tuln | grep -E ':(3478|8081)\b' '' Vor der Installation sollten die vorgesehenen Hostports nicht bereits durch einen anderen Dienst belegt sein. ===== 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 folgenden Arbeiten werden mit einem administrativen Benutzer durchgeführt. ===== 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 nur bei der Erstinstallation neu generiert. '' 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 > "$file" fi done chmod 700 secrets chmod 600 secrets/* '' Kontrolle: '' ls -l secrets/ '' Bei Updates keine neuen Secrets erzeugen! Die bereits verwendeten Schlüssel müssen unverändert bleiben. Secrets niemals in öffentlich zugänglichen Repositories, Logdateien oder Dokumentationen veröffentlichen. ===== 6. Umgebungsdatei erstellen ===== Die Datei ''.env'' enthält die Konfigurationsvariablen des Containers. '' cd /opt/docker/talk-hpb umask 077 { echo "NC_DOMAIN=cloud.example.org" echo "TALK_HOST=talk.example.org" echo "TALK_PORT=3478" echo "TZ=Europe/Vienna" echo "TURN_SECRET=$(cat secrets/turn_secret)" echo "SIGNALING_SECRET=$(cat secrets/signaling_secret)" echo "INTERNAL_SECRET=$(cat secrets/internal_secret)" } > .env chmod 600 .env '' Bei einer vorhandenen Installation die ''.env'' nicht erneut mit diesem Block überschreiben. Vor Änderungen immer die aktuelle Datei sichern. Die ''.env'' enthält sensible Informationen und darf nicht veröffentlicht werden. ===== 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 user: "1000" init: true env_file: - .env ports: - "127.0.0.1:8081:8081/tcp" - "3478:3478/tcp" - "3478:3478/udp" read_only: true tmpfs: - /run - /opt/eturnal/run - /conf - /tmp cap_drop: - NET_RAW healthcheck: test: ["CMD", "/healthcheck.sh"] interval: 30s

timeout: 30s retries: 3 '' **Erläuterung: * 8081/TCP ist ausschließlich lokal für Apache verfügbar. * 3478/TCP und UDP werden für TURN bereitgestellt. * Die Secrets werden über die geschützte ''.env'' übergeben. * Der Container wird nach einem Serverneustart automatisch wieder gestartet. * Das Dateisystem des Containers ist weitgehend schreibgeschützt. Der Tag ''latest'' vereinfacht die Erstinstallation. Für den langfristigen Produktionsbetrieb ist ein ausdrücklich geprüfter und festgelegter Image-Tag oder Digest empfehlenswert. Bei Verwendung von ''latest'' wird durch ''docker compose pull'' gegebenenfalls eine neuere Version geladen. ==== 7.1 Konfiguration prüfen ==== '' cd /opt/docker/talk-hpb docker compose config --quiet '' Bei einer korrekten Syntax erscheint keine Fehlermeldung. Achtung: Die normale Ausgabe von ''docker compose config'' kann Geheimnisse enthalten. Deshalb für die reine Syntaxprüfung ''--quiet'' verwenden. ==== 7.2 Container starten ==== '' docker compose pull docker compose up -d '' ==== 7.3 Containerstatus prüfen ==== '' docker compose ps '' Erwarteter Zustand: ''Up ... (healthy)'' ===== 8. Signaling-Server lokal testen ===== '' curl -i http://127.0.0.1:8081/api/v1/welcome '' Erwartetes Ergebnis: '' HTTP/1.1 200 OK {"nextcloud-spreed-signaling":"Welcome","version":"..."} '' Die Versionsnummer hängt vom installierten Containerimage ab. Bei Problemen: '' docker logs --tail 100 nextcloud-talk-hpb '' ===== 9. Apache-Reverse-Proxy einrichten ===== Der Signaling-Server wird über HTTPS unter einer eigenen Subdomain erreichbar gemacht. Beispiel: ''https://talk.example.org'' ==== 9.1 TLS-Zertifikat ==== Bei Verwendung von 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'' '' ServerName talk.example.org Redirect permanent / https://talk.example.org/ 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 '' Diese Variante verwendet Apache mit WebSocket-Upgrade-Unterstützung über mod_proxy_http. Bei älteren Apache-Versionen ist gegebenenfalls eine gesonderte WebSocket-Proxy-Konfiguration erforderlich. Die Zertifikatspfade müssen an die tatsächlich verwendeten Zertifikate 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 '' Erwartet wird: ''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 Connection: Upgrade Upgrade: websocket '' Zusätzlich kann eine JSON-Nachricht mit den verfügbaren Signaling-Funktionen erscheinen. Eine spätere Meldung wie: '' "reason":"hello_timeout" '' ist bei diesem einfachen Test normalerweise unkritisch, weil curl nicht das vollständige Signaling-Protokoll ausführt. ===== 11. Hochleistungs-Backend in Nextcloud eintragen ===== Als Administrator anmelden: ''https://cloud.example.org/settings/admin/talk'' Den Abschnitt ''Hochleistungs-Backend'' öffnen. Auf ''Hochleistungs-Backend-Server hinzufügen'' klicken. Folgende Werte eintragen: ^ Einstellung ^ Wert ^ | Backend-URL | https://talk.example.org | | 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. Verbindung speichern beziehungsweise erneut prüfen. Erwartetes Ergebnis: * Nextcloud erkennt den Signaling-Server. * Die WebSocket-Verbindung funktioniert. * Die unterstützten Funktionen werden angezeigt. Falls Nextcloud meldet, dass einzelne Signaling-Funktionen fehlen, zunächst die Versionskompatibilität prüfen. Eine solche Warnung bedeutet nicht zwangsläufig, dass alle Talk-Anrufe fehlschlagen. Ein vorhandenes und funktionierendes Backend nicht allein wegen einer optionalen Funktionswarnung durch eine ungeprüfte Entwicklungsversion ersetzen. ===== 12. TURN-Server einrichten ===== Der offizielle AIO-Talk-Container 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 '' ==== 12.2 Router-Portweiterleitung ==== Bei einem Server hinter NAT müssen zwei Portfreigaben eingerichtet werden. ^ Protokoll ^ Externer Port ^ Ziel ^ | TCP | 3478 | Interne IPv4-Adresse des Docker-Hosts, Port 3478 | | UDP | 3478 | Interne IPv4-Adresse des Docker-Hosts, Port 3478 | Bei einer FRITZ!Box: - Benutzeroberfläche öffnen. - Internet → Freigaben → Portfreigaben. - Server auswählen. - Neue Portfreigabe hinzufügen. - UDP 3478 konfigurieren. - Zweite Freigabe für TCP 3478 hinzufügen. - Einstellungen übernehmen. Die bestehende HTTPS-Freigabe für Port 443 bleibt unverändert. Port 8081 darf nicht öffentlich freigegeben werden. ==== 12.3 TURN in Nextcloud konfigurieren ==== In Nextcloud: ''Administration → Talk → TURN-Server'' Folgende Einstellungen verwenden: ^ Feld ^ Wert ^ | Protokoll | nur turn: | | TURN-Server-URL | talk.example.org:3478 | | TURN-Server-Secret | Inhalt von turn_secret | | Transport | UDP und TCP | Secret anzeigen: '' cat /opt/docker/talk-hpb/secrets/turn_secret '' Die Konfiguration speichern und über das Testsymbol überprüfen. Ein grüner Haken bestätigt den erfolgreichen Konfigurationstest. ==== 12.4 STUN ==== Optional kann ein eigener STUN-Server konfiguriert werden. Beispiel: ''talk.example.org:3478'' Ein bereits funktionierender STUN-Eintrag muss nicht zwingend ersetzt werden. ===== 13. Gesamtsystem prüfen ===== ==== 13.1 HPB-Container ==== '' 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 Erreichbarkeit aus Nextcloud ==== Bei einer Nextcloud-Docker-Installation 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' '' ==== 13.5 Praxistest ==== - Testunterhaltung in Nextcloud Talk erstellen. - Videoanruf zwischen zwei Geräten starten. - Mindestens ein Gerät über ein externes Netzwerk verbinden. - Audio in beide Richtungen prüfen. - Video in beide Richtungen prüfen. - Anschließend einen Anruf mit mindestens drei Teilnehmern durchführen. Ein erfolgreicher HTTPS- oder TURN-Konfigurationstest ersetzt keinen praktischen Test der Audio- und Videoverbindungen. Die tatsächliche Qualität hängt auch von Internetanbindung, Netzwerkumgebung und Endgeräten ab. ===== 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 | Im HPB-Image enthalten | | Apache | Paketverwaltung des Betriebssystems | **Empfohlene Grundregel: * Sicherheitsupdates priorisieren. * Vor jedem Update Kompatibilität und Release Notes prüfen. * Vor jedem Update Backups erstellen. * Updates möglichst außerhalb laufender Konferenzen durchführen. * Neue Versionen zuerst in einer Testumgebung prüfen. * Nach jedem Update einen Funktionstest durchführen. ==== 14.1 Kompatibilität prüfen ==== Vor einem Upgrade prüfen: * Unterstützt die Talk-App die eingesetzte Nextcloud-Version? * Unterstützt das HPB die benötigten Signaling-Funktionen? * Gibt es bekannte Fehler im neuen HPB-Image? * Ist die verwendete PHP-Version mit Nextcloud kompatibel? * Sind vorhandene weitere Nextcloud-Apps kompatibel? Bei**

Nextcloud-Major-Upgrades dürfen Hauptversionen nicht übersprungen werden. ===== 15. HPB vor dem 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 sensible Schlüssel und muss geschützt aufbewahrt werden. ==== 15.2 Aktuelles Image dokumentieren ==== '' docker inspect nextcloud-talk-hpb \ --format ' .Config.Image ' '' Zusätzlich die aktuelle Signaling-Version dokumentieren: '' curl -fsS \ https://talk.example.org/api/v1/welcome ''

image 15.3 Image lokal sichern (optional) ==== Für eine schnellere Rückkehr zu einem bereits getesteten Image kann zusätzlich eine lokale Image-Sicherung erstellt werden. '' docker image save \ ghcr.io/nextcloud-releases/aio-talk:latest \ -o /opt/backups/talk-hpb-image.tar '' Diese Sicherung kann sehr groß werden. Bei produktiven Installationen ist ein festgelegter Image-Tag oder Digest und dessen Dokumentation normalerweise zweckmäßiger. ===== 16. HPB aktualisieren ===== ==== 16.1 Neue Version prüfen ==== Vor dem Update: * Release Notes lesen. * Bekannte Fehler prüfen. * Kompatibilität zur Talk-App prüfen. * Wartungsfenster festlegen. Bei Verwendung eines festgelegten Image-Tags zuerst den gewünschten neuen Tag in `compose.yaml` eintragen. ==== 16.2 Neues Image herunterladen ==== '' cd /opt/docker/talk-hpb docker compose pull talk-hpb '' ==== 16.3 Container aktualisieren ==== '' docker compose up -d –no-deps talk-hpb '' Dabei wird der Container mit dem neuen Image neu erstellt, sofern sich das Image geändert hat. Die bestehende `.env` und die Secrets bleiben erhalten. ==== 16.4 Update kontrollieren ==== '' docker compose ps curl -fsS \ https://talk.example.org/api/v1/welcome docker logs –tail 100 nextcloud-talk-hpb '' Danach: - HPB-Verbindung in Nextcloud prüfen. - TURN-Konfigurationstest ausführen. - Videoanruf durchführen. - Gegebenenfalls die angebotenen Signaling-Funktionen vergleichen. Bei einem normalen HPB-Update keine neuen Secrets erzeugen. Auch DNS, Apache-VirtualHost und Router-Portfreigaben müssen normalerweise nicht verändert werden. ===== 17. Talk-App aktualisieren ===== Die Talk-App besitzt die technische App-ID `spreed`. Zuerst den aktuellen Stand prüfen: '' cd /opt/docker/nextcloud docker compose exec -T -u www-data app \ php occ app:list | grep spreed '' Wenn ein kompatibles Update verfügbar ist: '' docker compose exec -T -u www-data app \ php occ app:update spreed '' Anschließend: '' docker compose exec -T -u www-data app \ php occ app:list | grep spreed '' Nach dem Update: * Talk-Administrationsseite öffnen. * HPB-Verbindung prüfen. * TURN testen. * Anruf mit mehreren Teilnehmern testen. Vor dem App-Update müssen ein konsistentes Nextcloud-Backup und ein geeigneter Wiederherstellungsweg vorhanden sein. ===== 18. Nextcloud Server aktualisieren ===== Das konkrete Verfahren hängt von der jeweiligen Nextcloud-Installation ab. Bei Docker Compose wird normalerweise ein neues Nextcloud-Image bereitgestellt. **Empfohlenes Vorgehen:** - Aktuelle Version und App-Kompatibilität prüfen. - Datenbank, Datenverzeichnis und Konfiguration sichern. - Neues Nextcloud-Image vorbereiten. - Update zuerst in einer Testinstanz durchführen. - Datenbankmigrationen abschließen. - Talk-App-Kompatibilität prüfen. - Produktivsystem im Wartungsfenster aktualisieren. - HPB und TURN erneut testen. Status prüfen: '' cd /opt/docker/nextcloud docker compose exec -T -u www-data app \ php occ status '' Nach dem

Imagewechsel prüfen, ob die Aktualisierung bereits automatisch erfolgt ist. Falls laut offizieller Upgrade-Anleitung erforderlich: '' docker compose exec -T -u www-data app \ php occ upgrade '' Der genaue Ablauf kann bei Standardimages, selbst gebauten Images und anderen Deploymentformen unterschiedlich sein. Bei einem Nextcloud-Major-Upgrade müssen App-Kompatibilität und PHP-Anforderungen vor dem produktiven Upgrade geklärt sein. Ein Datenbank-Upgrade kann nicht allein durch die Rückkehr zum alten Docker-Image rückgängig gemacht werden. ===== 19. Wiederherstellung nach fehlerhaftem HPB-Update ===== Ein HPB-Container-Update lässt sich normalerweise durch Wiederherstellung des zuvor eingesetzten Images und der Konfiguration rückgängig machen. ==== 19.1 Aktuellen Zustand dokumentieren ==== '' cd /opt/docker/talk-hpb docker compose ps docker logs –tail 100 nextcloud-talk-hpb '' ==== 19.2 Konfiguration wiederherstellen ==== Dazu den gesicherten Stand von `compose.yaml`, `.env` und `secrets` aus dem Backup zurückspielen. Das Backup muss zur vorherigen Containerkonfiguration passen. ==== 19.3 Vorheriges Image verwenden ==== Wenn das alte Image lokal verfügbar ist oder ein unveränderlicher Image-Tag beziehungsweise Digest dokumentiert wurde, diesen wieder in der Compose-Datei verwenden. Wurde das alte Image mit `docker image save` gesichert: '' docker image load \ -i /opt/backups/talk-hpb-image.tar '' Anschließend sicherstellen, dass `compose.yaml` tatsächlich auf das wiederhergestellte Image verweist. '' docker compose up -d –no-deps –force-recreate talk-hpb '' ==== 19.4 Wiederherstellung prüfen ==== '' docker compose ps curl -fsS \ https://talk.example.org/api/v1/welcome '' Danach Nextcloud-Verbindung und Videoanruf testen. Ein Rollback des HPB ersetzt nicht die Wiederherstellung einer Nextcloud-Datenbank nach einem fehlgeschlagenen Nextcloud-Server-Upgrade. Für Nextcloud sind gesonderte, konsistente Backups erforderlich. ===== 20. Betrieb und Monitoring ===== ==== 20.1 Containerstatus ==== '' cd /opt/docker/talk-hpb docker compose ps '' ==== 20.2 Logs ==== '' docker logs –tail 100 nextcloud-talk-hpb '' Live-Logs: '' docker logs -f nextcloud-talk-hpb '' ==== 20.3 Signaling prüfen ==== '' curl -fsS \ https://talk.example.org/api/v1/welcome '' ==== 20.4 Versionskontrolle ==== '' docker inspect nextcloud-talk-hpb \ –format ' .Config.Image ' '' ====

image 20.5 Empfohlene regelmäßige Prüfungen ==== ^ Intervall ^ Maßnahme ^ | Wöchentlich | Containerstatus und Fehlerlogs prüfen | | Wöchentlich | Sicherheitsmeldungen prüfen | | Monatlich | Neue Image-Versionen und Release Notes prüfen | | Vor jedem Upgrade | Konfiguration und Secrets sichern | | Nach jedem Upgrade | HPB-, TURN- und Videotest | | Regelmäßig | Wiederherstellbarkeit der Backups testen | ===== 21. Typische Fehler ===== ==== Kein Hochleistungs-Backend eingerichtet ==== Mögliche Ursachen: * HPB nicht in Nextcloud eingetragen * Falsche Backend-URL * Falsches Signaling-Secret * Container nicht gestartet Prüfen: '' docker compose ps curl -i \ https://talk.example.org/api/v1/welcome '' ==== WebSocket-Verbindung fehlgeschlagen ==== Mögliche Ursachen: * Fehlerhafter Apache-Reverse-Proxy * WebSocket-Upgrade wird nicht weitergeleitet * Zertifikatsproblem * Browser- oder Authentifizierungsproblem Prüfen: '' 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 '' `101 Switching Protocols` bestätigt das erfolgreiche WebSocket-Upgrade, jedoch noch keine vollständige Anmeldung. ==== TURN-Test fehlgeschlagen ==== Prüfen: * TURN_SECRET korrekt? * Port 3478/TCP freigegeben? * Port 3478/UDP freigegeben? * Router-Portweiterleitung richtig? * DNS-Adresse korrekt? * Container läuft? ==== Video funktioniert nur im lokalen Netzwerk ==== Mögliche Ursachen: * TURN nicht erreichbar * Router-Portweiterleitung fehlt * Firewall blockiert Pakete * NAT- oder Client-Netzwerkproblem Einen weiteren Test über ein externes Netzwerk durchführen. ==== HPB meldet fehlende Funktionen ==== Die Talk-App kann zusätzliche Signaling-Funktionen erwarten. In diesem Fall: * Signaling-Version feststellen. * Unterstützte Funktionen auslesen. * Kompatibilität zur Talk-App prüfen. * Release Notes und bekannte Fehler prüfen. * Erst danach ein gezieltes Update durchführen. ===== 22. Sicherheitsempfehlungen ===== * Signaling ausschließlich über HTTPS bereitstellen. * Zertifikatsprüfung aktiviert lassen. * Port 8081 nicht öffentlich freigeben. * TURN nur über erforderliche Ports veröffentlichen. * Sichere, zufällig generierte Secrets einsetzen. * Secret-

Dateien vor fremdem Zugriff schützen. * Backups verschlüsselt beziehungsweise zugriffsgeschützt speichern. * Containerimages aus vertrauenswürdigen Quellen beziehen. * Produktionsupdates vorab testen. * Docker und Betriebssystem regelmäßig mit Sicherheitsupdates versorgen. * Nicht benötigte Router- und Firewall-Freigaben vermeiden. ===== 23. Zusammenfassung ===== Die Installation besteht aus: - Eigenem Docker-Compose-Projekt für das Talk-HPB. - Drei Zugangsschlüsseln für TURN, Signaling und interne Kommunikation. - Apache-Reverse-Proxy mit HTTPS. - WebSocket-Unterstützung. - TURN-Portfreigabe für TCP und UDP. - Eintragung des HPB in Nextcloud Talk. - Einrichtung des TURN-Servers. - Prüfung mit externen Videokonferenzteilnehmern. Für Updates gilt: * Nextcloud Server und Talk-App aufeinander abstimmen. * HPB getrennt aktualisieren. * Secrets beibehalten. * Vorherige funktionierende Version dokumentieren. * Nach jedem Update einen vollständigen Funktionstest durchführen. ===== 24. Offizielle Dokumentation ===== * Nextcloud Talk – Quick Install * Nextcloud All-in-One * Offizielle Docker-Compose-Beispielkonfiguration * Nextcloud Spreed Signaling * Nextcloud Administrationshandbuch * Docker Compose * Apache2-Dokumentation * Certbot / Let's Encrypt —- Stand: Oktober 2026 Dokumentation für eigenständige Nextcloud-Talk-HPB-Installationen mit Docker Compose und Apache2.

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

Permanent link:
http://wiki.waldhofer.at/doku.php?id=nextcloud:talk&rev=1793348432

Last update: **2026/10/30 09:20**