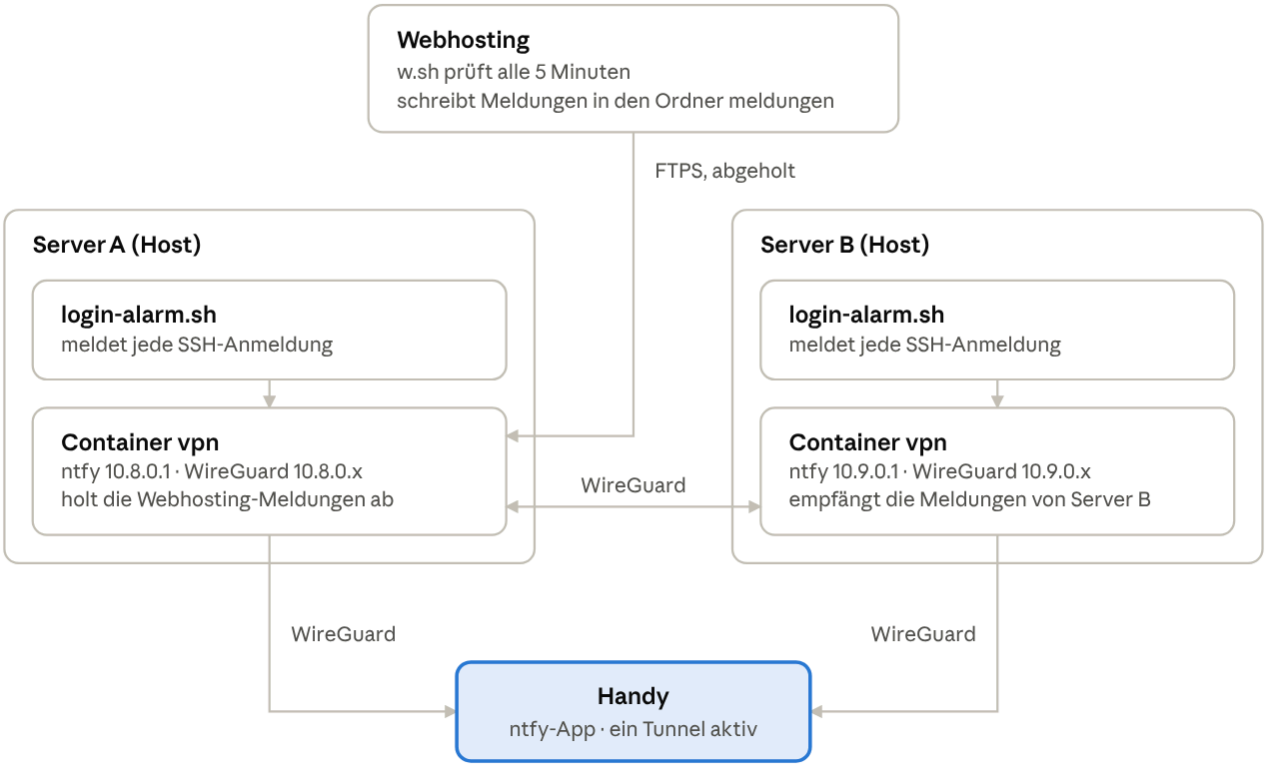
Eigene Server-Benachrichtigungen mit WireGuard, LXD und ntfy

Oct 23, 2026 · @Sebastian Ebner · Quelle: KI Werkzeug Cloude

Überblick

Am Ende melden zwei eigene V-Server jede SSH-Anmeldung und ein Webhosting jede Dateiänderung aufs Handy, ohne fremde Dienste wie ntfy.sh. Jeder Server bekommt einen WireGuard-VPN und einen eigenen ntfy-Server in einem LXD-Container. Die beiden Container sind miteinander verbunden, dadurch erreicht das Handy beide Benachrichtigungsserver, egal mit welchem Tunnel es gerade verbunden ist.

Alle Meldungen laufen nur über eigene Server zum Handy



Aufbau · 2 Server, 1 Webhosting, 1 Handy

Jeder Host meldet an den ntfy im eigenen Container. Container A holt zusätzlich die Webhosting-Meldungen ab, und die Verbindung zwischen den Containern macht beide ntfy-Server aus jedem Tunnel erreichbar.

Voraussetzungen

- Zwei Linux-Server (Ubuntu) mit root-Zugang, je einer eigenen öffentlichen IPv4-Adresse und eingerichtetem LXD (`snap install lxd`, dann `lxd init`).
- Ein Webhosting mit DirectAdmin, Cronjobs und FTPS (bei anderen Panels heißen die Menüs anders, der Ablauf ist gleich).
- Ein Android-Handy mit den Apps **WireGuard**, **ntfy** und **Termux**. In Termux einmalig `pkg install openssh` ausführen.

Platzhalter in dieser Anleitung — alles in GROSSBUCHSTABEN durch die eigenen Werte ersetzen:

Platzhalter	Bedeutung	Beispiel
SERVER-A-IP	öffentliche Adresse von Server A (Hauptserver)	203.0.113.10
SERVER-B-IP	öffentliche Adresse von Server B	203.0.113.20
CONTAINER-IP	interne Adresse des Containers auf dem jeweiligen Server, zeigt Teil 1 an	10.x.x.x
BENUTZER	Benutzername auf dem Webhosting	webuser
DOMAIN	Domain auf dem Webhosting	beispiel.de
FTP-SERVER	Servername, den auch die eigene FTP-App benutzt	ftp.beispiel.de
PASSWORT	Passwort des eigenen FTP-Kontos aus Teil 5	–

Die Netze im Tunnel: Server A nutzt `10.8.0.x`, Server B `10.9.0.x`. Der Container ist jeweils `.1`, das Handy `.2`. Verschiedene Netze sind nötig, sonst hätten beide ntfy-Server dieselbe Adresse und die ntfy-App könnte sie nicht unterscheiden.

Wo man gerade ist, zeigt die Eingabezeile:

Eingabezeile	Ort
`root@SERVERNAME:~#`	Host (der Server selbst)
`root@vpn:~#`	Container `vpn`
`~ $`	Termux auf dem Handy

Jeden Befehlskasten einzeln eingeben. Blöcke, die mit `cat … <<EOF` beginnen, komplett auf einmal einfügen; die `>` am Zeilenanfang dabei sind normal.

Teil 1: VPN und ntfy im Container (auf beiden Servern)

Teil 1 läuft auf beiden Servern mit denselben Befehlen. Der einzige Unterschied ist die Zahl `N`: **8 auf Server A, 9 auf Server B.** Alle Befehle setzen sie selbst ein.

1.1 Prüfen (Host)

```
modprobe wireguard && lsmod | grep wireguard
```

Es muss eine Zeile mit `wireguard` erscheinen. `lxc list` zeigt die vorhandenen Container; der Name `vpn` darf noch nicht vergeben sein.

1.2 Werte festlegen (Host)

```
CT=vpn; N=8
```

Auf Server B stattdessen `CT=vpn; N=9`.

```
HOST_IP=$(curl -4 -s ifconfig.me); echo $HOST_IP
```

Angezeigt werden muss die öffentliche Adresse des Servers. Die Werte gelten nur in dieser Sitzung; nach einer neuen Anmeldung 1.2 wiederholen.

1.3 Container anlegen (Host)

```
lxc init ubuntu:24.04 $CT -c linux.kernel_modules=wireguard -c
boot.autostart=true
```

```
lxc start $CT
```

```
until lxc exec $CT -- ip -4 -o addr show eth0 | grep -q inet; do sleep 2;
done
```

```
CIP=$(lxc exec $CT -- ip -4 -o addr show eth0 | awk '{print $4}' | cut -d/ -
f1); echo $CIP
```

Diese Adresse aufschreiben: Das ist die **CONTAINER-IP** dieses Servers, sie wird in Teil 4 gebraucht.

1.4 Feste Adresse und Weiterleitung (Host)

```
lxc config device override $CT eth0 ipv4.address=$CIP
```

```
lxc config device add $CT wgport proxy listen=udp:$HOST_IP:51820
connect=udp:$CIP:51820 nat=true
```

```
lxc config device show $CT
```

In der Ausgabe müssen `ipv4.address` und `wgport` mit `nat: "true"` stehen. Ist eine Firewall aktiv (`ufw status`), zusätzlich `ufw allow 51820/udp` ; eine Firewall im Kundenmenü des Anbieters muss UDP 51820 ebenfalls durchlassen.

1.5 In den Container wechseln (Host)

```
lxc exec $CT --env EP=$HOST_IP --env N=$N -- bash
```

Die Eingabezeile muss jetzt `root@vpn` zeigen. Die Werte `EP` (Server-Adresse) und `N` sind im Container verfügbar.

1.6 WireGuard einrichten (Container)

```
apt update && apt install -y wireguard iptables unattended-upgrades
```

Schlüssel erzeugen. Die runden Klammern sorgen dafür, dass nur diese Dateien geschützt angelegt werden und spätere Installationen nicht gestört werden:

```
mkdir -p /etc/wireguard && (cd /etc/wireguard && umask 077 && wg genkey | tee
server.key | wg pubkey > server.pub && wg genkey | tee handy.key | wg pubkey
> handy.pub)
```

```
echo 'net.ipv4.ip_forward=1' > /etc/sysctl.d/99-wg.conf && sysctl -w
net.ipv4.ip_forward=1
```

Diesen Block komplett einfügen:

```
cat > /etc/wireguard/wg0.conf <<EOF
[Interface]
Address = 10.$N.0.1/24
ListenPort = 51820
PrivateKey = $(cat /etc/wireguard/server.key)
PostUp = iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
# Handy
PublicKey = $(cat /etc/wireguard/handy.pub)
AllowedIPs = 10.$N.0.2/32
EOF
```

```
chmod 600 /etc/wireguard/wg0.conf && systemctl enable --now wg-quick@wg0 &&
wg show
```

`wg show` muss `interface: wg0` und einen `peer` zeigen.

1.7 ntfy einrichten (Container)

```
mkdir -p /etc/apt/keyrings && curl -L -o /etc/apt/keyrings/ntfy.gpg
https://archive.ntfy.sh/apt/keyring.gpg
```

```
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/ntfy.gpg]
https://archive.ntfy.sh/apt stable main" > /etc/apt/sources.list.d/ntfy.list
```

```
apt update && apt install -y ntfy
```

Diesen Block komplett einfügen:

```
cat > /etc/ntfy/server.yml <<EOF
base-url: "http://10.$N.0.1"
listen-http: ":80"
cache-file: "/var/cache/ntfy/cache.db"
cache-duration: "24h"
EOF
```

```
mkdir -p /var/cache/ntfy && chown ntfy:ntfy /var/cache/ntfy && systemctl
enable --now ntfy
```

```
curl -d "Test aus dem Container" http://localhost/ssh
```

Die Antwort muss `"event":"message"` enthalten. Der Port 80 ist von außen nicht erreichbar, weil LXD nur UDP 51820 weiterleitet. Meldungen werden 24 Stunden aufbewahrt und beim nächsten Verbinden nachgeliefert.

1.8 Handy-Konfiguration erzeugen (Container)

Diesen Block komplett einfügen:

```
(umask 077; cat > /root/handy.conf <<EOF
[Interface]
PrivateKey = $(cat /etc/wireguard/handy.key)
Address = 10.$N.0.2/32
DNS = 9.9.9.9

[Peer]
PublicKey = $(cat /etc/wireguard/server.pub)
Endpoint = $EP:51820
AllowedIPs = 0.0.0.0/0, ::/0
PersistentKeepalive = 25
EOF
)
```

```
grep -e Address -e Endpoint /root/handy.conf
```

Hinter `Endpoint =` muss die Server-Adresse mit `:51820` stehen. Dann zurück auf den Host und die Datei aus dem Container holen:

```
exit
```

```
lxc file pull vpn/root/handy.conf /root/handy.conf
```

Danach Teil 1 auf dem zweiten Server wiederholen, dort mit `N=9`.

Teil 2: Handy einrichten

Das Handy bekommt je Server einen Tunnel und in der ntfy-App je Server ein Abo. Android nutzt immer nur einen Tunnel gleichzeitig; nach Teil 3 erreicht aber jeder Tunnel beide ntfy-Server.

2.1 Konfigurationen holen (Termux)

Einmalig, den Befehl allein eingeben und die Frage nach dem Speicherzugriff beantworten:

```
termux-setup-storage
```

Dann je Server die Datei holen. Der Dateiname wird beim Import zum Tunnelnamen (höchstens 15 Zeichen, keine Leerzeichen):

```
scp root@SERVER-A-IP:/root/handy.conf ~/storage/downloads/server-a.conf
```

```
scp root@SERVER-B-IP:/root/handy.conf ~/storage/downloads/server-b.conf
```

2.2 WireGuard-App

1. „+“ → „Aus Datei oder Archiv importieren“ → `server-a.conf`, danach genauso `server-b.conf`.
2. Bei jedem Tunnel auf die Zeile mit den Anwendungen tippen: „Alle Anwendungen“ lassen oder, wenn nur bestimmte Apps eingeschlossen sind, **ntfy** mit einschließen. Sonst erreicht die ntfy-App ihre Server nicht.
3. Den Tunnel von Server A einschalten.
4. Android-Einstellungen → Netzwerk → VPN → Zahnrad bei WireGuard → „Durchgehend aktives VPN“.
5. Test: https://ifconfig.me im Browser zeigt SERVER-A-IP.

2.3 ntfy-App

Je Abo: „+“ → Thema eingeben → „Anderen Server verwenden“ anhaken → Server eintragen → abonnieren.

Thema	Server	Meldet
`ssh`	`http://10.8.0.1`	Anmeldungen auf Server A
`ssh`	`http://10.9.0.1`	Anmeldungen auf Server B
`webhosting`	`http://10.8.0.1`	Dateiänderungen auf dem Webhosting

Das Thema `login` funktioniert nicht, es ist in ntfy reserviert. Zum Schluss in den Android-Einstellungen bei der ntfy-App den Akku auf **„Nicht eingeschränkt“** stellen, sonst beendet Android die Verbindung im Hintergrund. Die Testmeldung aus 1.7 erscheint beim Abo von Server A sofort; das Abo von Server B zeigt bis nach Teil 3 ein orangenes Dreieck.

2.4 Aufräumen

Der private Handy-Schlüssel soll nur noch in der App liegen. Die Dateien in „Downloads“ löschen und auf beiden Servern (Host):

```
rm /root/handy.conf && lxc exec vpn -- rm /root/handy.conf
/etc/wireguard/handy.key
```

Teil 3: Die beiden Container verbinden

Nach diesem Teil leitet jeder Container Anfragen an das andere Netz weiter. Das Handy erreicht dann `10.8.0.1` und `10.9.0.1` in jedem der beiden Tunnel, und kein zusätzlicher Port muss geöffnet werden. Am einfachsten mit zwei Termux-Sitzungen, eine zu jedem Server.

3.1 Öffentliche Schlüssel anzeigen (Host, auf beiden Servern)

```
lxc exec vpn -- wg show wg0 public-key
```

Die Zeile jeweils kopieren. Öffentliche Schlüssel sind keine Geheimnisse.

3.2 Server B bei Server A eintragen (Server A)

```
lxc exec vpn -- bash
```

Den Schlüssel von **Server B** zwischen die beiden `'` einfügen und mit `echo $K` prüfen:

```
K='SCHLÜSSEL-VON-SERVER-B'
```

Dann diese eine Zeile (`SERVER-B-IP` ersetzen):

```
printf '\n[Peer]\n# Server B\nPublicKey = %s\nEndpoint = SERVER-B-
IP:51820\nAllowedIPs = 10.9.0.0/24\nPersistentKeepalive = 25\n' "$K" >>
/etc/wireguard/wg0.conf
```

```
tail -6 /etc/wireguard/wg0.conf && systemctl restart wg-quick@wg0 && exit
```

3.3 Server A bei Server B eintragen (Server B)

```
lxc exec vpn -- bash
```

```
K='SCHLÜSSEL-VON-SERVER-A'
```

```
printf '\n[Peer]\n# Server A\nPublicKey = %s\nEndpoint = SERVER-A-
IP:51820\nAllowedIPs = 10.8.0.0/24\nPersistentKeepalive = 25\n' "$K" >>
/etc/wireguard/wg0.conf
```

```
tail -6 /etc/wireguard/wg0.conf && systemctl restart wg-quick@wg0
```

3.4 Testen (Container auf Server B)

```
ping -c 3 10.8.0.1
```

```
curl -d "Test über die Verbindung" http://10.8.0.1/ssh
```

Drei Antworten beim `ping` und die Meldung beim Abo `10.8.0.1/ssh` = die Verbindung steht. Beim Neustart von `wg-quick@wg0` ist der Handy-Tunnel kurz unterbrochen und verbindet sich von selbst neu. In der ntfy-App darf danach bei keinem Abo mehr ein orangenes Dreieck stehen.

Teil 4: SSH-Login-Alarm (auf beiden Servern)

Jede SSH-Anmeldung am Host schickt eine Meldung an den ntfy im eigenen Container: leise bei bekannten Adressen, als Alarm bei allen anderen. Das Skript läuft als root, deshalb liegt es in `/root`, wo kein anderer Benutzer es ändern kann.

4.1 Skript anlegen (Host)

Diesen Block unverändert einfügen:

```
cat > /root/login-alarm.sh <<'EOF'
#!/bin/sh
# meldet jede SSH-Anmeldung an den ntfy im Container
PATH=/usr/sbin:/usr/bin:/sbin:/bin
[ "$PAM_TYPE" = "open_session" ] || exit 0
NTFY="http://CONTAINER-IP/ssh"
IP="${PAM_RHOST:-lokal}"
ZEIT=$(TZ=Europe/Berlin date '+%d.%m.%Y %H:%M:%S')
case "$IP" in
  10.*|EIGENE-ADRESSEN)
    PRIO=low;    TITEL="SSH-Anmeldung (bekannte Adresse)" ;;
  *)
    PRIO=urgent; TITEL="ACHTUNG: SSH-Anmeldung von unbekannter Adresse" ;;
esac
curl -s -m 10 -H "Title: $TITEL" -H "Priority: $PRIO" -H "Tags: key" \
  -d "Benutzer $PAM_USER auf $(hostname) von $IP am $ZEIT" "$NTFY" >/dev/null
2>&1 &
exit 0
EOF
```

Die Container-Adresse automatisch eintragen und die Rechte setzen:

```
sed -i "s#CONTAINER-IP#$(lxc config device get vpn eth0 ipv4.address)#"
/root/login-alarm.sh && chmod 700 /root/login-alarm.sh && grep NTFY=
/root/login-alarm.sh
```

Die letzte Zeile muss `NTFY="http://10.x.x.x/ssh"` mit der echten Adresse zeigen.

4.2 Bekannte Adressen eintragen

In der Zeile `10.*|EIGENE-ADRESSEN)` den Platzhalter ersetzen, z. B. mit `nano /root/login-alarm.sh`. Mehrere Adressen mit `|` trennen, `*` steht für beliebige Ziffern:

Eintrag	Wann bekannt
`10.*`	Anmeldung mit eingeschaltetem VPN dieses Servers (sie kommt von der Container-Adresse)

Eintrag	Wann bekannt
`SERVER-A-IP` / `SERVER-B-IP`	Anmeldung im Tunnel des jeweils anderen Servers
z. B. `176.*`	eigener Mobilfunk- oder Heimanschluss ohne VPN, wenn der Adressbereich bekannt ist

Wer gar keine weiteren Adressen eintragen will, schreibt nur `10.*)`. Eine Zeitzone außer `Europe/Berlin` in der Zeile `ZEIT=` anpassen.

4.3 Testen und einhängen (Host)

```
PAM_TYPE=open_session PAM_USER=root PAM_RHOST=1.2.3.4 /root/login-alarm.sh
```

Am Handy muss „ACHTUNG: SSH-Anmeldung von unbekannter Adresse“ mit `1.2.3.4` erscheinen — das ist richtig, die Adresse ist ausgedacht. Dann einhängen:

```
grep -q login-alarm /etc/pam.d/sshd || echo 'session optional pam_exec.so
/root/login-alarm.sh' >> /etc/pam.d/sshd
```

`optional` ist wichtig: Schlägt das Skript fehl, kommt man trotzdem hinein. Die laufende Sitzung offen lassen und in einer zweiten Termux-Sitzung neu anmelden; die Meldung muss mit der echten Adresse kommen.

Teil 5: Datei-Wächter auf dem Webhosting

Das Webhosting kann keinen VPN aufbauen. Der Wächter schreibt seine Meldungen deshalb als kleine Textdateien in einen eigenen Ordner, und Server A holt sie in Teil 6 verschlüsselt per FTPS ab. Alles in diesem Teil geschieht im DirectAdmin-Panel und seinem Dateimanager.

5.1 Ordner anlegen

Im Dateimanager auf der obersten Ebene (`/home/BENUTZER`, neben `domains`) den Ordner **`meldungen`** anlegen. Er liegt außerhalb von `public_html` und ist nicht aus dem Internet abrufbar.

5.2 Eigenes FTP-Konto

FTP-Verwaltung → FTP-Konto anlegen:

- Benutzer: `meldung` (daraus wird `meldung@DOMAIN`), eigenes langes Passwort ohne `'`.
- Verzeichnis: **Benutzerdefiniert**, eingetragen wird nur **`/meldungen`**. DirectAdmin setzt `/home/BENUTZER` selbst davor; wer den ganzen Pfad einträgt, bekommt ihn doppelt.

In der Übersicht muss beim Konto `home/BENUTZER/meldungen/` stehen. Dieses Konto sieht nur den Ordner `meldungen`, nicht die Websites.

5.3 Die w.sh anlegen

Im Dateimanager auf der obersten Ebene die Datei `w.sh` mit diesem Inhalt anlegen. In der Zeile `BASIS=` den Benutzernamen eintragen, in der `pruefe`-Zeile die Domain und einen Ordner, der sich ständig von selbst ändert (Cache, Zählerdaten); ohne solchen Ordner dort `""` eintragen.

```
#!/bin/sh
# Datei-Waechter: meldet neue, geaenderte und geloeschte Dateien
BASIS="/home/BENUTZER"
MARKE="$BASIS/.waechter-marke"
export LC_ALL=C

melde() {   # $1 Titel  $2 Liste  $3 Ordner
  [ -z "$2" ] && return
  M="$BASIS/meldungen"; mkdir -p "$M"
  N=$(printf '%s\n' "$2" | wc -l)
  D="$M/$(date +%Y%m%d-%H%M%S-%N)"
  { echo "$1 ($N)"; printf '%s\n' "$2" | sed "s#^$3##" | head -20; } >
"$D.tmp"
  mv "$D.tmp" "$D.txt"
}

pruefe() {  # $1 Name  $2 Ordner  $3 Ausnahme  $4 zweite Ausnahme (optional)
  A2="${4:-$3}"
  BESTAND="$BASIS/.waechter-liste-$1"
  find "$2" -type f ! -path "$3" ! -path "$A2" 2>/dev/null | sort >
"$BESTAND.neu"
  GEAENDERT=$( {
    [ -f "$MARKE" ]   && find "$2" -type f -cnewer "$MARKE" ! -path "$3" ! -
path "$A2" 2>/dev/null
    [ -f "$BESTAND" ] && comm -13 "$BESTAND" "$BESTAND.neu"
  } | sort -u )
  GELOESCHT=""
  [ -f "$BESTAND" ] && GELOESCHT=$(comm -23 "$BESTAND" "$BESTAND.neu")
  mv "$BESTAND.neu" "$BESTAND"
  melde "Neu/geaendert auf $1" "$GEAENDERT" "$2"
  melde "Geloescht auf $1" "$GELOESCHT" "$2"
}
```

```
touch "$MARKE.neu"
# je Website eine Zeile:
pruefe "DOMAIN" "$BASIS/domains/DOMAIN/public_html" "*/cache/*"
mv "$MARKE.neu" "$MARKE"
```

Rechte der Datei: 700. Mit Linux-Zeilenenden (LF) speichern; der Dateimanager von DirectAdmin macht das von selbst.

So arbeitet die w.sh:

- Sie erkennt Änderungen über die Statusänderungszeit der Dateien (`-cnewer`). Die lässt sich nicht zurückdatieren, auch nicht von einer FTP-App, die das Originaldatum mit hochlädt.
- Neue und gelöschte Dateien findet sie durch den Vergleich mit der Dateiliste vom letzten Lauf.
- Pro Meldung schreibt sie bis zu 20 Dateinamen und die Gesamtzahl. Die Datei entsteht erst als `.tmp` und wird dann umbenannt, damit nie eine halbfertige Meldung abgeholt wird.

5.4 Cronjob

DirectAdmin → Cronjobs → Minute `*/5`, alle anderen Felder `*`, Befehl:

```
/bin/sh /home/BENUTZER/w.sh >/dev/null 2>&1
```

Der erste Lauf legt nur Marke und Dateiliste an und meldet nichts. Zum Testen kann die Minute kurz auf `*` stehen, danach wieder auf `*/5` stellen.

5.5 Testen

Eine Datei nach `public_html` hochladen und bis nach dem nächsten Lauf warten. Dann muss im Ordner `meldungen` eine `.txt` liegen, die erste Zeile ist der Titel. Liegen lassen, sie dient in Teil 6 als Test.

Teil 6: Meldungen abholen (Container auf Server A)

Der Container auf Server A holt jede Minute neue Meldungsdateien per FTPS ab, schickt sie an seinen ntfy und löscht sie auf beiden Seiten. Er läuft im Container, weil dort auch ntfy läuft; das FTP-Passwort liegt so nur im Container.

6.1 In den Container und lftp installieren (Server A)

```
lxc exec vpn -- bash
```

```
apt install -y lftp cron
```

6.2 Zugangsdaten speichern (Container)

Vor dem Einfügen `FTP-SERVER`, `DOMAIN` und `PASSWORT` ersetzen; die `'` um das Passwort bleiben stehen:

```
(umask 077; cat > /root/.meldungen-ftp <<'EOF'
FTP_HOST=FTP-SERVER
FTP_USER=meldung@DOMAIN
LFTP_PASSWORD='PASSWORT'
EOF
)
```

Verbindung testen — die Test-`.txt` aus 5.5 muss aufgelistet werden:

```
. /root/.meldungen-ftp; export LFTP_PASSWORD; lftp --env-password -u
"$FTP_USER" "$FTP_HOST" -e "set ftp:ssl-force true; set net:max-retries 1;
ls; quit"
```

6.3 Abhol-Skript anlegen (Container)

Diesen Block unverändert einfügen:

```
cat > /usr/local/bin/meldungen-holen.sh <<'EOF'
#!/bin/sh
# holt Meldungen vom Webhosting (FTPS) und schickt sie an ntfy
. /root/.meldungen-ftp
export LFTP_PASSWORD
NTFY="http://localhost/webhosting"
EINGANG=/var/lib/meldungen
mkdir -p "$EINGANG"
lftp --env-password -u "$FTP_USER" "$FTP_HOST" -e "set ftp:ssl-force true;
set net:timeout 20; set net:max-retries 2; mirror --Remove-source-files --
include-glob=*.txt / $EINGANG; quit" >/dev/null 2>&1
for f in "$EINGANG"/*.txt; do
  [ -f "$f" ] || continue
  TITEL=$(head -1 "$f")
  tail -n +2 "$f" | curl -s -m 15 -H "Title: $TITEL" -H "Tags: warning" --
data-binary @- "$NTFY" >/dev/null && rm -f "$f"
done
EOF
```

```
chmod 700 /usr/local/bin/meldungen-holen.sh && sh -x
/usr/local/bin/meldungen-holen.sh
```

Die Meldung zur Testdatei muss beim Abo `webhosting` ankommen, und der Ordner `meldungen` auf dem Webhosting ist danach leer. Eine Datei wird erst gelöscht, wenn das Senden geklappt hat; ist ntfy kurz nicht erreichbar, versucht der nächste Lauf es erneut.

6.4 Jede Minute automatisch (Container)

```
echo '* * * * * root sleep 30; flock -n /run/meldungen.lock /bin/sh /usr/local/bin/meldungen-holen.sh' > /etc/cron.d/meldungen && systemctl enable --now cron
```

Das `sleep 30` lässt den Container eine halbe Minute nach dem Webhosting abholen, `flock` verhindert zwei gleichzeitige Läufe. Eine Meldung kommt damit bis zu etwa 5,5 Minuten nach der Änderung an (5 Minuten Prüfabstand plus Abholen). Den Eintrag in **diesem Container** anlegen, nicht auf dem Host.

Fehlerbehebung

Fast alle Fehler bei der Einrichtung haben eine von wenigen Ursachen: falscher Ort (Host statt Container oder der falsche Server), ein nicht ersetzter Platzhalter oder ein beim Einfügen zerbrochener Befehl.

Meldung oder Beobachtung	Ursache	Lösung
`>` am Zeilenanfang	Die Konsole sammelt einen mehrzeiligen Block bis `EOF`	normal; hängt sie dauerhaft, fehlt `EOF` oder ein schließendes `'` → Strg + C, neu einfügen
`Instance is not running` / Container fehlt	Befehl auf dem falschen Server eingegeben	Eingabezeile prüfen, dort steht der Servername
`wg show` zeigt nichts	wg0.conf leer oder Dienst nicht gestartet	`cat /etc/wireguard/wg0.conf`, dann `systemctl restart wg-quick@wg0`
`Endpoint = :51820` ohne Adresse	Container ohne `--env EP=$HOST_IP` betreten	`exit`, Schritt 1.2 und 1.5 wiederholen
Befehle mit `$CT` oder `$CIP` scheitern	Nach neuer Anmeldung sind die Werte weg	1.2 wiederholen, `CIP=$(lxc config device get vpn eth0 ipv4.address)`

Meldung oder Beobachtung	Ursache	Lösung
`NO_PUBKEY` beim Installieren von ntfy	Schlüsseldatei nicht lesbar	`chmod 644 /etc/apt/keyrings/ntfy.gpg`, dann `apt update`
`topic name is not allowed`	Thema wie `login` ist in ntfy reserviert	anderes Thema, z. B. `ssh`
Orangenes Dreieck in der ntfy-App	Server gerade nicht erreichbar	VPN an? ntfy im Tunnel eingeschlossen? Teil 3 erledigt?
Keine Meldung, obwohl der Test im Container klappt	ntfy-App fehlt in den eingeschlossenen Anwendungen des Tunnels	in der WireGuard-App ntfy einschließen oder „Alle Anwendungen“
`421 Home directory not available`	Pfad des FTP-Kontos doppelt eingetragen	in DirectAdmin nur `/meldungen` eintragen
lftp hängt bei `Resolving host address`	Platzhalter `FTP-SERVER` nicht ersetzt	`/root/.meldungen-ftp` mit echten Werten neu schreiben
Fehler mit `certificate`	Servername passt nicht zum Zertifikat	den Servernamen der eigenen FTP-App nehmen
Webhosting-Meldungen kommen nur von Hand	Cron-Eintrag fehlt im Container	`cat /etc/cron.d/meldungen` im Container, nicht auf dem Host
Zwei Container mit Weiterleitung auf Port 51820	alter Container hat noch `wgport`	`lxc config device remove ALTER-NAME wgport`
Aus einer Webseite kopierter Befehl kommt in einer Zeile an	Zeilenumbrüche gingen beim Kopieren verloren	Befehl abtippen oder aus einer Quelle mit erhaltenen Zeilen kopieren

Hinweise

Von außen ist auf jedem Server nur SSH und der WireGuard-Port UDP 51820 erreichbar; die ntfy-Server antworten nur Geräten im Tunnel.

Wo die Meldungen entlanglaufen

Strecke	Weg	Verschlüsselt
Webhosting → Container A	FTPS, Container A holt ab	ja (TLS)
Host → eigener Container	LXD-Netz, verlässt den Server nicht	–
Container A ↔ Container B	WireGuard	ja
Container → Handy	WireGuard	ja

Sicherheit

- Private Schlüssel (`server.key` , `handy.key` , die `.conf` der App) nie weitergeben oder fotografieren. Mit einer Handy-Konfiguration kann jeder über den Server ins Internet.
- Der WireGuard-Port antwortet nur auf Pakete mit gültigem Schlüssel und ist für Portscanner praktisch unsichtbar. Die eigentliche Angriffsfläche bleibt SSH.
- Das FTP-Konto `meldung` sieht nur den Ordner `meldungen` . Seine Zugangsdaten liegen in einer Datei, die nur root im Container lesen darf.
- Der Kanalname eines Themas ist hier kein Geheimnis mehr, weil nur Geräte im Tunnel den Server erreichen.

Wenn etwas ausfällt

- VPN am Handy aus: Meldungen kommen nicht an, werden 24 Stunden aufbewahrt und beim nächsten Verbinden nachgeliefert.
- Server A aus: Webhosting-Meldungen sammeln sich im Ordner `meldungen` und werden später abgeholt. Server B meldet weiter, erreichbar über seinen eigenen Tunnel.
- Server B aus: Server A und das Webhosting laufen normal weiter.

Erweitern

- **Weiteres Handy oder Gerät:** im Container ein Schlüsselpaar erzeugen, einen `[Peer]` mit der nächsten freien Adresse anhängen (`10.N.0.3/32` , dann `.4`), `systemctl restart wg-quick@wg0` , die Konfiguration wie in 1.8 mit dieser Adresse erzeugen.
- **Nur ein Server:** Teil 3 entfällt, Teil 1, 2, 4, 5 und 6 bleiben gleich.
- **Gerät entfernen:** seinen `[Peer]` -Block aus `/etc/wireguard/wg0.conf` löschen und `systemctl restart wg-quick@wg0` .
- **Aufbewahrung ändern:** in `/etc/ntfy/server.yml` `cache-duration` anpassen, dann `systemctl restart ntfy` .

Quellen: ntfy – Installation, WireGuard, LXD.