

# Libre nms smnp

## Ändert librenms bei smnp scans die ip adressen zuordnung von smnp geräten wenn sie sich ändern?

Das kommt ganz darauf an, **wie** du das Gerät ursprünglich zu LibreNMS hinzugefügt hast.

Die kurze Antwort lautet: **Automatisch passiert das in der Regel nur, wenn du DNS nutzt.** Wenn du Geräte per IP hinzugefügt hast, wird LibreNMS das Gerät als "Down" markieren, wenn sich die IP ändert, anstatt die Adresse automatisch zu aktualisieren.

Hier ist die Aufschlüsselung, wie LibreNMS auf deinem Proxmox-Cluster damit umgeht:

### 1. Szenario: Gerät wurde per DNS-Hostname hinzugefügt (Empfohlen)

Wenn du das Gerät z. B. als `server01.deine-domain.local` hinzugefügt hast:

- **Ja, es funktioniert.** LibreNMS löst bei jedem Polling- oder Discovery-Lauf den Hostnamen neu auf.
- Ändert sich die IP-Adresse im DNS, bemerkt LibreNMS dies beim nächsten Lauf, aktualisiert die zugeordnete IP in der Datenbank und das Monitoring läuft nahtlos weiter.

### 2. Szenario: Gerät wurde per IP-Adresse hinzugefügt

Wenn du das Gerät z. B. als `192.168.1.50` hinzugefügt hast:

- **Nein, es wird nicht automatisch aktualisiert.** LibreNMS pollt stur diese spezifische IP.
- Wechselt das Gerät auf `192.168.1.51`, erhält LibreNMS keine Antwort mehr von der `.50` und meldet das Gerät als **Down**.

- Selbst wenn ein "SNMP Scan" (Auto-Discovery) über das Subnetz läuft und das Gerät unter der neuen IP `.51` findet, wird LibreNMS es meist als **neues Gerät** erkennen (oder wegen Duplikat-Schutz meckern), aber nicht den alten Eintrag "verschieben".

## Besonderheit bei Net-SNMP (Windows)

Da du Net-SNMP unter Windows nutzt, hast du einen Vorteil bei der Identifikation. Net-SNMP liefert in der Regel einen sauberen `sysName` (System Name) zurück.

- LibreNMS nutzt diesen `sysName`, um Duplikate zu erkennen.
- Wenn du **Auto-Discovery** für dein Subnetz aktiviert hast, wird LibreNMS das Gerät unter der neuen IP finden. Es wird dann aber wahrscheinlich warnen, dass ein Gerät mit diesem `sysName` bereits existiert (unter der alten IP), anstatt die alte IP einfach zu überschreiben. Das dient dem Schutz vor Datenkorruption (damit nicht RRD-Graphen von zwei verschiedenen Geräten vermischt werden).

## Lösungsmöglichkeiten

Wenn sich eine IP geändert hat und das Gerät in LibreNMS "Down" ist, hast du zwei saubere Wege, das zu korrigieren, ohne die Historie (Graphen) zu verlieren:

**A. Über die Web-GUI (Edit Device):** Du gehst auf das Gerät -> Einstellungen (Zahnrad) -> **Edit**. Dort kannst du das Feld "Hostname" bzw. "IP" manuell auf die neue IP ändern.

**B. Über die Konsole (Proxmox/LXC):** Das ist oft schneller und sauberer. Du kannst das eingebaute Rename-Tool nutzen:

Bash

```
cd /opt/librenms
./lnms device:rename <ALTE_IP> <NEUE_IP>
```

(bzw. `./lnms device:rename <ALTE_IP> <NEUER_DNS_NAME>` - was langfristig besser ist).

## Zusammenfassung

Verlasse dich bei IP-Änderungen **nicht** auf den automatischen SNMP-Scan. LibreNMS ist hier konservativ, um Datenfehler zu vermeiden. Der sicherste Weg für eine dynamische Umgebung ist immer:

1. **DNS nutzen** (und TTL niedrig halten).

2. Geräte in LibreNMS per **FQDN** hinzufügen.

---

**Möchtest du wissen, wie man ein Skript aufsetzt, das IP-Änderungen über die Net-SNMP `sysName` Identifizierung automatisch in LibreNMS nachpflegt (via API)?**

Kann ich erkannte client geräte mit net-snmp automatisch mit fqdn hinzufügen?

Die direkte Antwort ist: **Ja, aber nur indirekt über DNS.**

LibreNMS hat keine eingebaute Funktion "Nutze `sysName` als Hostnamen beim Hinzufügen". Es verlässt sich beim Auto-Discovery (SNMP Scan) fast vollständig auf die Namensauflösung (DNS).

Hier sind die drei Wege, wie du das Ziel erreichst, dass deine Net-SNMP Clients mit FQDN statt IP erscheinen:

## 1. Der "Königsweg": Reverse DNS (PTR Records)

Das ist der Weg, den LibreNMS standardmäßig erwartet.

- **Wie es funktioniert:** Wenn der Discovery-Scanner eine IP (z.B. `192.168.1.50`) findet, fragt er deinen DNS-Server: "Wer ist `.50`?".
- **Voraussetzung:** Du musst im DNS-Server (in deinem Fall vermutlich auf dem Windows Domain Controller oder dem Proxmox-Umfeld) einen **Reverse-Lookup-Zone (PTR)** Eintrag für die Clients haben.
- **Ergebnis:** Wenn der DNS-Server mit `client01.deine-domain.local` antwortet, fügt LibreNMS das Gerät automatisch mit diesem FQDN hinzu. Antwortet DNS nicht, wird die IP genommen.

## 2. Die "Visuelle" Lösung (Empfohlen, wenn kein Reverse DNS möglich)

Wenn du keinen sauberen Reverse-DNS für alle Clients pflegen willst, kannst du LibreNMS anweisen, die IP in der Datenbank zu behalten, aber in der Oberfläche **immer** den `sysName` (den Net-SNMP liefert) anzuzeigen.

Füge dazu folgende Zeile in deine `/opt/librenms/config.php` ein:

PHP

```
$config['force_ip_to_sysname'] = true;
```

- **Vorteil:** Die Geräte werden technisch per IP überwacht (keine DNS-Probleme), aber in Listen, Graphen und Warnungen siehst du den echten Namen (z.B. `DESKTOP-H3K9L`).
- **Nachteil:** In der Datenbank ist der "Hostname" weiterhin die IP.

## 3. Die "Harte" Methode (Automatisiertes Umbenennen)

Da du Net-SNMP unter Windows nutzt, liefert das Gerät zuverlässig seinen Namen. Du kannst ein kleines Skript (Cronjob) laufen lassen, das nachträglich IPs in echte Hostnamen umwandelt, indem es die LibreNMS API oder CLI nutzt.

### Beispiel-Logik für ein Skript:

1. Hole Liste aller Geräte, die als IP hinzugefügt sind (Regex auf IP-Muster).
2. Frage per SNMP den `sysName` ab.
3. Führe `./lnms device:rename <IP> <sysName>` aus.

## Meine Empfehlung für deine Umgebung

Da du Net-SNMP auf Windows nutzt und das Monitoring auf Proxmox läuft:

1. Aktiviere `$config['force_ip_to_sysname'] = true;`. Das ist sofort wirksam und löst das "Hässliche-IP-Problem" in der Ansicht.
2. Wenn du echte FQDNs in der Datenbank brauchst (z.B. für externe Reports), Sorge dafür, dass dein DHCP-Server die Hostnamen der Windows-Clients automatisch im DNS registriert (Dynamic DNS Updates). Dann macht LibreNMS den Rest beim nächsten Scan von alleine.

erstelle mir ein script das dies 1 mal um 01.00Uhr durchführt für librenms im docker container

Hier ist ein robustes **Bash-Skript**, das auf deinem Host-System (z. B. der VM oder dem LXC-Container in deinem Proxmox-Cluster, in dem Docker läuft) ausgeführt wird.

Es nutzt die LibreNMS-internen Tools (`lnms tinker` und `lnms device:rename`) im Docker-Container, um sicherzustellen, dass die Datenbank und die RRD-Dateien (Graphen) korrekt migriert werden.

Das Skript: `auto_rename_ips.sh`

Erstelle diese Datei auf dem Host, auf dem der Docker-Container läuft (z.B. unter `/root/scripts/auto_rename_ips.sh`).

## Bash

```
#!/bin/bash

# --- KONFIGURATION ---
# Name des LibreNMS Docker Containers (prüfen mit 'docker ps')
CONTAINER_NAME="librenms"

# Domain-Suffix (Optional)
# Falls Net-SNMP nur "server1" liefert, du aber "server1.local" willst.
# Leer lassen "", wenn der sysName von Net-SNMP direkt übernommen werden soll.
DOMAIN_SUFFIX=""
# Beispiel: DOMAIN_SUFFIX=".deine-domain.local"

# --- LOGIK ---

echo "[$(date)] Starte IP-zu-sysName Umbenennung..."

# 1. Liste alle Geräte, die als IP gespeichert sind, aber einen sysName haben
# Wir nutzen 'Inms tinker' (PHP Shell), um direkt die DB zu fragen. Das ist sicherer als SQL von außen.
# Regex sucht nach IP-Muster (x.x.x.x)
DEVICES_TO_RENAME=$(docker exec -i $CONTAINER_NAME Inms tinker --execute
"\App\Models\Device::whereRaw(\\"hostname REGEXP '^([0-9]{1,3})\\.([0-9]{1,3})\\.([0-9]{1,3})\\.([0-9]{1,3})\$\\")-
>whereNotNull('sysName')->where('sysName', '!=', '')->get()->each(function(\$d) { echo \$d->hostname . '|' .
\$d->sysName . PHP_EOL; });" | grep "|")

# Prüfen, ob Geräte gefunden wurden
if [ -z "$DEVICES_TO_RENAME" ]; then
    echo "Keine Geräte gefunden, die umbenannt werden müssen."
    exit 0
fi

# 2. Loop durch die Ergebnisse und benenne um
while IFS='|' read -r OLD_IP SYS_NAME; do
    # Bereinigen von möglichen Leerzeichen
    SYS_NAME=$(echo "$SYS_NAME" | tr -d '[:space:]')
```

```
# Zieldomain bauen
NEW_HOSTNAME="${SYS_NAME}${DOMAIN_SUFFIX}"

# Sicherheitscheck: Nicht umbenennen, wenn Name == IP (passiert manchmal bei Fehlkonfiguration)
if [[ "$NEW_HOSTNAME" == "$OLD_IP" ]]; then
    echo "Überprünge $OLD_IP: sysName ist identisch mit IP."
    continue
fi

echo "Benenne um: $OLD_IP -> $NEW_HOSTNAME"

# Führe den offiziellen Rename-Befehl im Container aus
docker exec $CONTAINER_NAME Inms device:rename "$OLD_IP" "$NEW_HOSTNAME"

done <<< "$DEVICES_TO_RENAME"

echo "[$(date)] Fertig."
```

# Installation & Einrichtung

## 1. Skript speichern und ausführbar machen

Führe auf deinem Docker-Host folgende Befehle aus:

Bash

```
mkdir -p /root/scripts
nano /root/scripts/auto_rename_ips.sh
# -> Inhalt von oben einfügen und speichern (Strg+O, Enter, Strg+X)

chmod +x /root/scripts/auto_rename_ips.sh
```

## 2. Testlauf (Trockenübung)

Führe das Skript einmal manuell aus, um zu sehen, ob es funktioniert:

Bash

```
/root/scripts/auto_rename_ips.sh
```

Wenn deine Net-SNMP Clients sauber konfiguriert sind, solltest du jetzt sehen, wie die IPs in Hostnamen umgewandelt werden.

### 3. Cronjob einrichten (01:00 Uhr nachts)

Öffne die Crontab des Root-Users auf dem Host:

Bash

```
crontab -e
```

Füge folgende Zeile am Ende ein:

Code-Snippet

```
0 1 * * * /root/scripts/auto_rename_ips.sh >> /var/log/librenms_rename.log 2>&1
```

## Wichtige Hinweise für deine Umgebung

- Net-SNMP sysName:** Da du Windows mit Net-SNMP nutzt, wird der Name meist als einfacher Hostname (z.B. `TERMINALSERVER01`) zurückgegeben, oft ohne `.domain.local`.
  - Wenn du volle FQDNs willst, trage deine Domain oben im Skript bei `DOMAIN_SUFFIX=".local"` (oder was dein Cluster nutzt) ein.
- Docker Container Name:** Das Skript geht davon aus, dass dein Container `librenms` heißt. Wenn du `docker-compose` nutzt, heißt er oft `librenms_librenms_1` oder ähnlich. Prüfe das kurz mit `docker ps` und passe die Variable `CONTAINER_NAME` im Skript an.
- RRD Daten:** Da wir `Inms device:rename` nutzen, bleiben alle deine historischen Graphen erhalten. Die RRD-Dateien werden auf Dateiebene vom System automatisch umbenannt.

Möchtest du, dass ich das Skript anpasse, um automatisch zu erkennen, ob es sich um einen deiner **Terrascaler Datacore Cluster** Server handelt (z.B. anhand der `sysDescr`), um diese speziell zu behandeln?

---

Revision #1

Created 30 December 2025 09:48:19 by Ahorn

Updated 30 December 2025 09:56:19 by Ahorn