Navidrome Musik-Streaming auf Ubuntu Server

Navidrome Server Software

> https://www.navidrome.org/
>
> https://www.navidrome.org/docs/installation/linux/
>
> https://github.com/navidrome/navidrome/releases

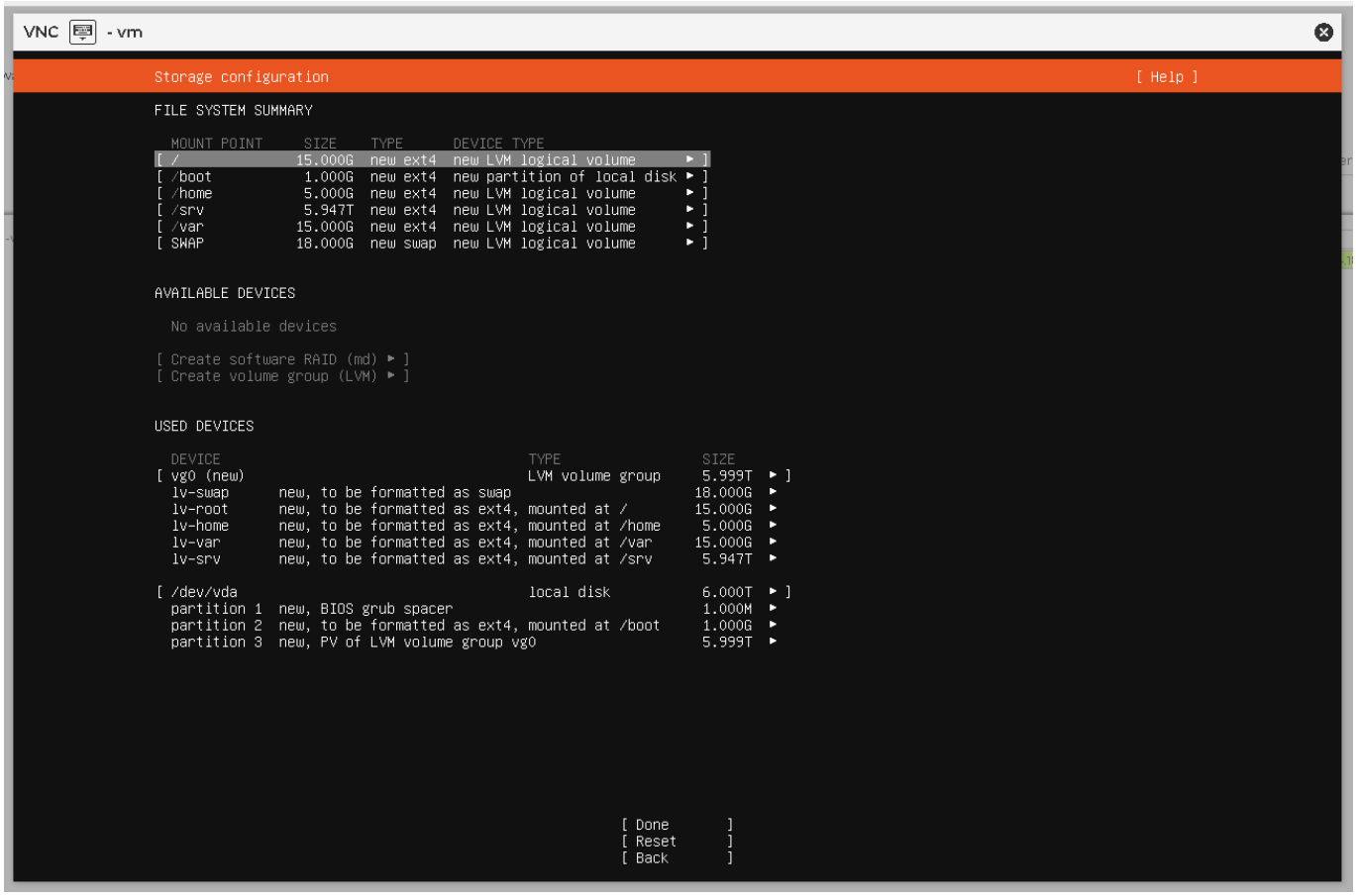
Installation

Volumes und Partitionierung

Um alles unterzubekommen, sollte man sich bereits bei der Installation des Servers Gedanken dazu machen, wie das Partitions- bzw. Festplattenlayout aussehen soll.

Folgende Fragen sind zu berücksichtigen:

- Wo soll die music library liegen? Soll es mehrere geben?
- Wo soll das Arbeitsverzeichnis von Navidrome gespeichert werden? –> Datenbank
- Wohin mit dem Cache der Applikation? –> Transcoding, Bilddaten (cover arts, album arts)




```
/dev/sda1   EFI     /boot/efi   ext4
/dev/sda2   /boot       ext4
/dev/sda3   LVM PV
  └─ vg0
     ├─ lv-swap   swap
     ├─ lv-root   /
     ├─ lv-home   /home
     ├─ lv-var    /var
     └─ lv-srv    /srv
```

Installation der Anwendung

Service-User anlegen

```
useradd -r -s /usr/sbin/nologin navidrome
```

Anwendung installieren

apt nutzen, um Abhängigkeiten aufzulösen!

```
apt install ./navidrome_0.61.2_linux_amd64.deb
```

Daten wiederherstellen (bei Migration)

```
rm -rf /var/lib/navidrome
tar -xzvf /tmp/navidrome-backup.tar.gz -C /
chown -R navidrome:navidrome /var/lib/navidrome
```

Config erstellen oder übernehmen

> https://www.navidrome.org/docs/usage/configuration/options/

```
mv navidrome.toml /etc/navidrome/navidrome.toml
chown -R navidrome:navidrome /etc/navidrome/navidrome.toml
```

Verzeichnisse übernehmen / anlegen

```
cat /etc/navidrome/navidrome.toml |grep '\"/'
```

```
mkdir -p /srv/LOSSLESS_AUDIO
mkdir -p /srv/LOSSY_AUDIO
mkdir -p /srv/PLAYLISTS_NAVIDROME
mkdir -p /srv/BACKUP_NAVIDROME
chown -R navidrome:navidrome /srv/BACKUP_NAVIDROME
chown -R navidrome:navidrome /srv/PLAYLISTS_NAVIDROME
```

Spezialfall music library

Um die Bibliothek automatisiert auf dem Laufenden halten zu können, lege ich einen weiteren Service-User an und vergebe entsprechende Berechtigungen:

- navidrome –> lesen
- audioadmin –> lesen+schreiben

```
adduser audioadmin
groupadd audio
usermod -aG audio audioadmin
usermod -aG audio navidrome
```

Aktualisierung / Update der Server-Applikation

Welche Version läuft?

```
navidrome --version
```

Ist das Backup gelaufen?

Den Link zur aktuellen Version holen und das Release laden:

> https://github.com/navidrome/navidrome/releases

```
wget -c
https://github.com/navidrome/navidrome/releases/download/v0.60.0/navidrome_0
.60.0_linux_amd64.deb
```

```
systemctl stop navidrome

dpkg -i navidrome_0.60.0_linux_amd64.deb
(Reading database ... 153778 files and directories currently installed.)
Preparing to unpack navidrome_0.60.0_linux_amd64.deb ...
Unpacking navidrome (0.60.0) over (0.59.0) ...
Setting up navidrome (0.60.0) ...

apt -f install
```

Gibt es Neuerungen, die Änderungen an der Konfiguration mit sich ziehen?

```
vi /etc/navidrome/navidrome.toml
```

Server wieder starten:

```
systemctl start navidrome

systemctl status navidrome
● navidrome.service - Your Personal Streaming Service
     Loaded: loaded (/etc/systemd/system/navidrome.service; enabled; preset:
enabled)
     Active: active (running) since Fri 2026-02-06 20:28:44 CET; 6s ago
   Main PID: 5985 (navidrome)
      Tasks: 14 (limit: 19083)
     Memory: 159.4M (peak: 293.1M)
        CPU: 1.723s
     CGroup: /system.slice/navidrome.service
             └─5985 /usr/bin/navidrome service execute -c
/etc/navidrome/navidrome.toml

Feb 06 20:28:44 server systemd[1]: Started navidrome.service - Your Personal
Streaming Service.
Feb 06 20:28:44 server navidrome[5985]:  _   _                 _     _
Feb 06 20:28:44 server navidrome[5985]: | \ | |               (_)   | |
Feb 06 20:28:44 server navidrome[5985]: |  \| | __ ___   ___  __| |_ __ ___
_ __ ___   ___
Feb 06 20:28:44 server navidrome[5985]: | . ` |/ _` \ \ / / |/ _` | '__/ _
\| '_ ` _ \ / _ \
Feb 06 20:28:44 server navidrome[5985]: | |\  | (_| |\ V /| | (_| | | | (_)
| | | | | |  __/
Feb 06 20:28:44 server navidrome[5985]: \_| \_/\__,_| \_/ |_|\__,_|_|
```

```
\___/|_| |_| |_|\___|
Feb 06 20:28:44 server navidrome[5985]:                                Version:
0.60.0 (0c8f2a55
```

Apache2 als Reverse-Proxy

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName music.example.com
    DocumentRoot /var/www/navidrome/html

    # Logs
    ErrorLog /var/www/navidrome/logs/error.log
    CustomLog /var/www/navidrome/logs/access.log combined

    # Proxy Settings
    ProxyPreserveHost On
    Protocols http/1.1   # HTTP/1.1 erzwingen für stabile Streaming-
Verbindungen

    # WebSocket Support (Rewrites nur für Upgrade)
    RewriteEngine On
    RewriteCond %{HTTP:Upgrade} =websocket [NC]
    RewriteRule /(.*) ws://127.0.0.1:4533/$1 [P,L]

    # Normaler Proxy für alle anderen Requests
    ProxyPass / http://127.0.0.1:4533/ nocanon
    ProxyPassReverse / http://127.0.0.1:4533/

    # Forwarded Headers
    RequestHeader set X-Forwarded-Proto "https"
    RequestHeader set X-Forwarded-Port "443"
    RequestHeader set X-Forwarded-For "%{REMOTE_ADDR}s"

    # Connection / Timeout Optimierungen
    KeepAlive On
    MaxKeepAliveRequests 100
    KeepAliveTimeout 5
    ProxyTimeout 300

    # Security Headers
    # X-XSS-Protection ist veraltet, kann optional drinbleiben oder entfernt
werden
    # Header always set X-XSS-Protection "1; mode=block"
    Header always set X-Content-Type-Options "nosniff"
    Header always set Strict-Transport-Security "max-age=31536000"

    # SSL Certs
    Include /etc/letsencrypt/options-ssl-apache.conf
```

```
    SSLCertificateFile /etc/letsencrypt/live/music.example.com/fullchain.pem
    SSLCertificateKeyFile
/etc/letsencrypt/live/music.example.com/privkey.pem

</VirtualHost>
</IfModule>
```

ModSecurity-Konfiguration für Apache2 Reverse Proxy vor Navidrome

Ziel der Konfiguration

- Schutz vor typischen Webangriffen (SQLi, XSS, RCE etc.)
- Keine unnötigen False Positives für Navidrome
- API-/Streaming-kompatibel
- Sinnvolles Logging

1. Basis: ModSecurity aktivieren

```
a2enmod security2
systemctl restart apache2
```

2. Hauptkonfiguration

Datei:

```
/etc/modsecurity/modsecurity.conf
```

Wichtige Anpassungen:

```
SecRuleEngine On

# Body handling (wichtig für API!)
SecRequestBodyAccess On
SecResponseBodyAccess Off

# Limits (Navidrome kann größere Requests haben, z.B. Uploads)
SecRequestBodyLimit 104857600
SecRequestBodyNoFilesLimit 1048576

# JSON Support (sehr wichtig für Navidrome API)
SecRequestBodyProcessor JSON

# Logging
SecAuditEngine RelevantOnly
SecAuditLog /var/log/apache2/modsec_audit.log
```

```
SecAuditLogParts ABIJDEFHZ

# Weniger aggressiv bei Streaming
SecResponseBodyMimeType text/plain text/html text/xml application/json

# Encoding
SecDefaultAction "phase:2,log,auditlog,deny,status:403"
```

3. OWASP Core Rule Set (empfohlen)

Installieren:

```
 apt install modsecurity-crs
```

Dann aktivieren:

```
cp /usr/share/modsecurity-crs/crs-setup.conf.example /etc/modsecurity/crs-setup.conf
```

Apache Config ergänzen (z. B. in /etc/apache2/mods-enabled/security2.conf):

```
IncludeOptional /etc/modsecurity/crs-setup.conf
IncludeOptional /usr/share/modsecurity-crs/rules/*.conf
```

3a. Option: Core Ruleset von Github (aktueller)

Benötigte Version klonen. In der Kombi Apache2 + Modsec2 unter Ubuntu 24.04 ist die v3.3.5 am sinnvollsten.

```
git clone https://github.com/coreruleset/coreruleset.git owasp-crs
cd owasp-crs
git fetch --tags
git checkout -b v3.3.5 tags/v3.3.5
cp crs-setup.conf.example crs-setup.conf
vi crs-setup.conf
```

Nun kann die `crs-setup.conf` auf die eigenen Bedürfnisse angepasst werden. Ich ändere am Default allerdings nichts.

Wichtiger ist es, das Ruleset auch korrekt einzubinden.

```
vi /etc/apache2/mods-enabled/security2.conf
vi /etc/modsecurity/modsecurity.conf
```

4. WICHTIG: Ausnahmen für Navidrome

Navidrome nutzt:

- /rest/* API
- JSON Requests
- Query-Parameter intensiv
- Streaming (Range Requests)

Ohne Anpassung gibt es False Positives.

5. Custom Rules für Navidrome

Neue Datei:

```
/etc/modsecurity/navidrome-exclusions.conf
```

```
# Navidrome API weniger restriktiv behandeln
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1000001,phase:1,pass,nolog,ctl:ruleEngine=DetectionOnly"

# JSON API toleranter machen
SecRule REQUEST_HEADERS:Content-Type "application/json" \
    "id:1000002,phase:1,pass,nolog,ctl:requestBodyProcessor=JSON"

# Range Requests erlauben (für Streaming)
SecRule REQUEST_HEADERS:Range ".*" \
    "id:1000003,phase:1,pass,nolog,ctl:ruleRemoveByTag=attack-protocol"

# False positives reduzieren (SQLi / XSS bei API)
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1000004,phase:1,pass,nolog,ctl:ruleRemoveByTag=attack-
sqli,ctl:ruleRemoveByTag=attack-xss"

# Große Downloads erlauben
SecRule REQUEST_URI "@beginsWith /rest/stream" \
    "id:1000005,phase:1,pass,nolog,ctl:responseBodyAccess=Off"
```

oder

```
# ============================================================
# Navidrome CRS Minimal Exclusions - Optimiert 2026
# ============================================================

# (1) SQL Injection False Positives
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1100001,phase:1,pass,nolog,\
    ctl:ruleRemoveById=942100,\
    ctl:ruleRemoveById=942110,\
    ctl:ruleRemoveById=942150"

# (2) XSS False Positives
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1100002,phase:1,pass,nolog,\
```

```
    ctl:ruleRemoveById=941100,\
    ctl:ruleRemoveById=941160"

# (3) Protocol Enforcement
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1100003,phase:1,pass,nolog,\
    ctl:ruleRemoveById=920230,\
    ctl:ruleRemoveById=920240"

# (4) Streaming (Range Requests)
SecRule REQUEST_URI "@beginsWith /rest/stream" \
    "id:1100004,phase:1,pass,nolog,\
    ctl:ruleRemoveById=920200"

# (5) Typische Parameter entschärfen
SecRule REQUEST_URI "@beginsWith /rest/" \
    "id:1100005,phase:1,pass,nolog,chain"
    SecRule ARGS_NAMES "^(u|t|s|f|c)$" \
        "ctl:ruleRemoveById=942100"

# (6) JSON Requests korrekt parsen
SecRule REQUEST_HEADERS:Content-Type "application/json" \
    "id:1100006,phase:1,pass,nolog,\
    ctl:requestBodyProcessor=JSON"

# (7) CoverArt / Artist Images
SecRule REQUEST_URI "@beginsWith /rest/getCoverArt" \
    "id:1100010,phase:1,pass,nolog,ctl:responseBodyAccess=Off"

SecRule REQUEST_URI "@beginsWith /rest/getCoverArt" \
    "id:1100011,phase:3,pass,nolog,ctl:responseBodyAccess=Off"

SecRule REQUEST_URI "@beginsWith /rest/getCoverArt" \
    "id:1100012,phase:1,pass,nolog,\
    ctl:ruleRemoveById=941100,\
    ctl:ruleRemoveById=920200,\
    ctl:ruleRemoveById=920420,\
    ctl:ruleRemoveById=920430,\
    ctl:ruleRemoveById=933100"

# (8) Artist API POST/DELETE WebUI
SecRule REQUEST_URI "@beginsWith /api/artist" \
    "id:1100013,phase:1,pass,nolog,chain"
    SecRule REQUEST_METHOD "@rx ^(POST|DELETE)$" \
        "ctl:ruleRemoveById=941100"

# (9) Anomaly Scoring feinjustieren (Warnung vermeiden)
SecAction "id:1100014,phase:1,pass,nolog,t:none,\
    setvar:tx.inbound_anomaly_score_threshold=10,\
    setvar:tx.outbound_anomaly_score_threshold=5,\
    tag:'navidrome-exclusions'"
```

Einbinden:

```
Include /etc/modsecurity/navidrome-exclusions.conf
```

6. Apache VirtualHost (Reverse Proxy)

Beispiel:

```
<VirtualHost *:80>
    ServerName music.example.com

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:4533/
    ProxyPassReverse / http://127.0.0.1:4533/

    # Sicherheit
    Header always set X-Frame-Options SAMEORIGIN
    Header always set X-Content-Type-Options nosniff
    Header always set X-XSS-Protection "1; mode=block"

    # Logging
    ErrorLog ${APACHE_LOG_DIR}/navidrome_error.log
    CustomLog ${APACHE_LOG_DIR}/navidrome_access.log combined
</VirtualHost>
```

7. Test & Debug

Logs prüfen:

```
tail -f /var/log/apache2/modsec_audit.log
```

Typische Probleme:

- 403 bei API Calls → Rule greift zu aggressiv
- Streaming bricht → Range Requests blockiert
- Login geht nicht → JSON / POST blockiert

8. Empfohlene Betriebsmodi

Phase	Einstellung
Test	DetectionOnly
Produktion	On + gezielte Exclusions