Airsonic (Advanced) Streamingserver

Ich nutze die Stand-alone-WAR.

> https://airsonic.github.io/docs/install/war-standalone/
>
> https://github.com/airsonic-advanced/airsonic-advanced/releases

Installation

Java JRE

Wie auch Subsonic ist Airsonic in Java geschrieben. Daher wird für den Betrieb das Java Runtime Environment in der Version 8 benötigt:

```
# aptitude install openjdk-8-jre
```

Für Airsonic Advanced wird Java 11 benötigt:

```
# aptitude install openjdk-11-jre
```

Falls bereits eine andere Java-Version installiert worden ist, kann über den folgenden Befehl Java 8 zum Systemstandard gemacht werden.

```
# update-alternatives --config java
```

Um zu erfahren, welche Java-Version Systemstandard ist, kann folgender Aufruf getätigt werden:

```
# java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

Transcoder

```
# aptitude install ffmpeg
```

Falls der Ordner noch nicht vorhanden sein sollte, bitte anlegen:

```
# mkdir /srv/airsonic/transcode
```

Nachsehen, wo die ffmpeg liegt:

```
#  whereis ffmpeg
```

```
ffmpeg: /usr/bin/ffmpeg /usr/share/ffmpeg /usr/share/man/man1/ffmpeg.1.gz
```

Nun geben wir der Applikation Zugriff auf den Transcoder:

```
# cd /srv/airsonic/transcode
# ln -s /usr/bin/ffmpeg
# chown -h airsonic.airsonic ffmpeg
```

Optionale Tools

Optional können noch folgende Tools zur Bearbeitung der Musiksammlung installiert werden:

```
aptitude install id3 id3v2 eyed3
```

Das System vorbereiten

Umlegen des Arbeitsverzeichnises auf eine andere Platte:

```
# mkdir -p /srv/airsonic
# cd /var
# ln -s  /srv/airsonic airsonic
```

Nun legen wir das Dienstkonto an:

```
# useradd -M -r -s /usr/sbin/nologin -d /var/airsonic airsonic
# chown -R airsonic:airsonic /srv/airsonic
```

Nun die Verzeichnisse für die Musiksammlung:

```
# cd /srv
# mkdir PLAYLISTS
# chown -R airsonic.airsonic PLAYLISTS/
# mkdir ALBEN
# chown -R airsonic.airsonic ALBEN/
```

Die Anwendung als Dienst implementieren

Bei Airsonic:

```
# wget
https://raw.githubusercontent.com/airsonic/airsonic/master/contrib/airsonic.
service -O /etc/systemd/system/airsonic.service
# wget
https://raw.githubusercontent.com/airsonic/airsonic/master/contrib/airsonic-
systemd-env -O /etc/default/airsonic
# wget -c
```

```
https://github.com/airsonic/airsonic/releases/download/v10.6.2/airsonic.war
# cp airsonic.war /srv/airsonic
# systemctl daemon-reload
# systemctl start airsonic.service
# systemctl enable airsonic.service
```

Bei Airsonic Advanced:

```
# wget
https://raw.githubusercontent.com/airsonic-advanced/airsonic-advanced/master
/contrib/airsonic.service -O /etc/systemd/system/airsonic.service
# wget
https://raw.githubusercontent.com/airsonic/airsonic/master/contrib/airsonic-
systemd-env -O /etc/default/airsonic
# wget -c
https://github.com/airsonic-advanced/airsonic-advanced/releases/download/11.
0.0-SNAPSHOT.20211007145752/airsonic.war
# cp airsonic.war /srv/airsonic
# systemctl daemon-reload
# systemctl start airsonic.service
# systemctl enable airsonic.service
```

Innerhalb der airsonic.service muss der Pfad für `EnvironmentFile` an die Gegebenheiten in Ubuntu angepasst werden.

/etc/systemd/system/airsonic.service

```
[Unit]
Description=Airsonic Media Server
After=remote-fs.target network.target
AssertPathExists=/var/airsonic

[Service]
Type=simple
Environment="JAVA_JAR=/var/airsonic/airsonic.war"
Environment="JAVA_OPTS=-Xmx700m"
Environment="AIRSONIC_HOME=/var/airsonic"
Environment="PORT=8080"
Environment="CONTEXT_PATH=/airsonic"
Environment="JAVA_ARGS="
EnvironmentFile=-/etc/default/airsonic
ExecStart=/usr/bin/java \
          $JAVA_OPTS \
          -Dairsonic.home=${AIRSONIC_HOME} \
          -Dserver.servlet.context-path=${CONTEXT_PATH} \
          -Dserver.port=${PORT} \
          -jar ${JAVA_JAR} $JAVA_ARGS
User=airsonic
Group=airsonic

# See
```

```
https://www.freedesktop.org/software/systemd/man/systemd.exec.html
# for details
DevicePolicy=closed
NoNewPrivileges=yes
PrivateDevices=yes
PrivateTmp=yes
PrivateUsers=yes
ProtectControlGroups=yes
ProtectKernelModules=yes
ProtectKernelTunables=yes
RestrictAddressFamilies=AF_UNIX AF_INET AF_INET6
RestrictNamespaces=yes
RestrictRealtime=yes
SystemCallFilter=~@clock @debug @module @mount @obsolete @privileged
@reboot @setuid @swap
ReadWritePaths=/var/airsonic

# You can change the following line to `strict` instead of `full`
# if you don't want airsonic to be able to
# write anything on your filesystem outside of AIRSONIC_HOME.
ProtectSystem=full

# You can uncomment the following line if you don't have any media
# in /home/…. This will prevent airsonic from ever reading/writing
anything there.
#ProtectHome=true

# You can uncomment the following line if you're not using the OpenJDK.
# This will prevent processes from having a memory zone that is both
writeable
# and executeable, making hacker's lifes a bit harder.
#MemoryDenyWriteExecute=yes


[Install]
WantedBy=multi-user.target
```

Reverse Proxy

Airsonic Advanced nutzt Websockets. Entsprechend muss der Proxy dies unterstützen!

Apache

Installation

```
# aptitude install apache2
```

```
# a2enmod proxy proxy_http headers proxy_wstunnel rewrite ssl
# a2dismod status
```

Virtuelle Hosts anlegen

```
# cd /var/www
# mkdir airsonic
# mkdir -p airsonic/html
# mkdir -p airsonic/logs
```

/etc/apache2/sites-available/airsonic.conf

```
<VirtualHost *:80>
    ServerName domain.de
    ServerAlias www.domain.de v3.domain.de
    DocumentRoot /var/www/airsonic/html
    Redirect 301 / https://v3.domain.de/

    ErrorLog /var/www/airsonic/logs/error.log
    CustomLog /var/www/airsonic/logs/access.log combined
</VirtualHost>

<VirtualHost *:80>
    ServerName domain.rocks
    ServerAlias www.domain.rocks v3.domain.rocks
    DocumentRoot /var/www/airsonic/html
    Redirect 301 / https://v3.domain.rocks/

    ErrorLog /var/www/airsonic/logs/error.log
    CustomLog /var/www/airsonic/logs/access.log combined
</VirtualHost>

<VirtualHost *:80>
    ServerName domain.net
    ServerAlias www.domain.net v3.domain.net
    DocumentRoot /var/www/airsonic/html
    Redirect 301 / https://v3.domain.net/

    ErrorLog /var/www/airsonic/logs/error.log
    CustomLog /var/www/airsonic/logs/access.log combined
</VirtualHost>
```

Nun für HTTPS. Ich bereite die Konfiguration vor, noch bevor LE diese erstellt.

/etc/apache2/sites-enabled/airsonic-le-ssl.conf

```
<IfModule mod_ssl.c>
SSLStaplingCache shmcb:/var/run/apache2/stapling_cache(128000)
```

```
<VirtualHost *:443>
        ServerName domain.de
        ServerAlias v3.domain.de www.domain.de
        DocumentRoot /var/www/airsonic/html

        ErrorLog /var/www/airsonic/logs/error.log
        CustomLog /var/www/airsonic/logs/access.log combined

        ProxyPass / http://127.0.0.1:4040/
        ProxyPassReverse / http://127.0.0.1:4040/
        RequestHeader     set       X-Forwarded-Proto "https"

        #websocket proxying
        RewriteEngine on
        RewriteCond %{HTTP:Upgrade} websocket [NC]
        RewriteCond %{HTTP:Connection} upgrade [NC]
        RewriteRule ^/?(.*) "ws://127.0.0.1:4040/$1" [P,L]


        Header always unset X-Frame-Options

        Include /etc/letsencrypt/options-ssl-apache.conf
        Header always set Strict-Transport-Security "max-age=31536000"
        SSLUseStapling on
        SSLCertificateFile
/etc/letsencrypt/live/v3.domain.de/fullchain.pem
        SSLCertificateKeyFile
/etc/letsencrypt/live/v3.domain.de/privkey.pem
</VirtualHost>
</IfModule>
```

Virtuellen Host aktivieren:

```
# a2ensite airsonic
```

Damit der Server SSL sprechen kann, muss der Cerbot von Let's Encrypt installiert werden. Eine Anleitung habe ich hier bereits hinterlegt:

> Installation Let's Encrypt

From:
https://wiki.sebastianhetzel.net/ - **Sebastians IT-Wiki**

Permanent link:
https://wiki.sebastianhetzel.net/ubuntu:airsonic

Last update: **2021/10/30 14:50**