
Audalaxy Listener Logs Observation

Release 1.5.1.1

Audalaxy

Jan 19, 2025

DEUTSCHE DOKUMENTATION

1	Erste Schritte	3
1.1	Erklärungen zu den Dashboards	3
1.1.1	Menü	3
1.1.2	Dashboards	4
2	Technische Dokumentation	9
2.1	Logfileimport per Logimporter	9
2.1.1	Input Plugins für Logfiles	10
2.1.2	Parser Plugins für Streaming-Server	10
2.1.3	Output Plugins für die Datenübermittlung	11
2.1.4	Nachliefern von Daten im Fehlerfall	12
2.1.5	Healthcheck	13
2.2	Logfileimport per SFTP	13

Listener Logs Observation is a big data solution to monitoring listener Logs using dashboards.
Find user guides, technical spec, developer guides and API references.

ERSTE SCHRITTE

1.1 Erklärungen zu den Dashboards

Nach der Anmeldung gelangt man auf die Startseite – hier bekommt man eine Liste aller verfügbaren Dashboards, Links zu den entsprechenden Dokumentationen und Kontaktinformationen.

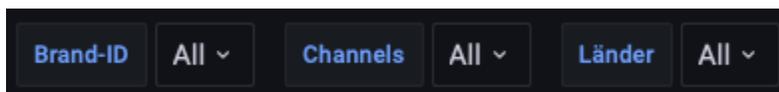


1.1.1 Menü

Über ein Auswahlménü können verschiedene Filter auf die einzelnen Dashboards gelegt werden – so Daten gezielter untersuchen zu können.

Auswahlmöglichkeiten:

- Brand-ID
- Channels
- Länder



Es lassen sich auch mehrere Optionen gleichzeitig anwählen. Wer alle Felder gleichzeitig markieren will, nutzt „All“.

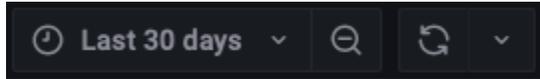
Im „Advanced Dashboard“ können zusätzlich folgende Filter genutzt werden:

- Länge der Session (in Millisekunden)

- Intervall (Einstellung, ob zeitliche Verläufe nach Stunden (h), Tagen (d) oder Wochen (7d oder 14d) ausgewertet werden)

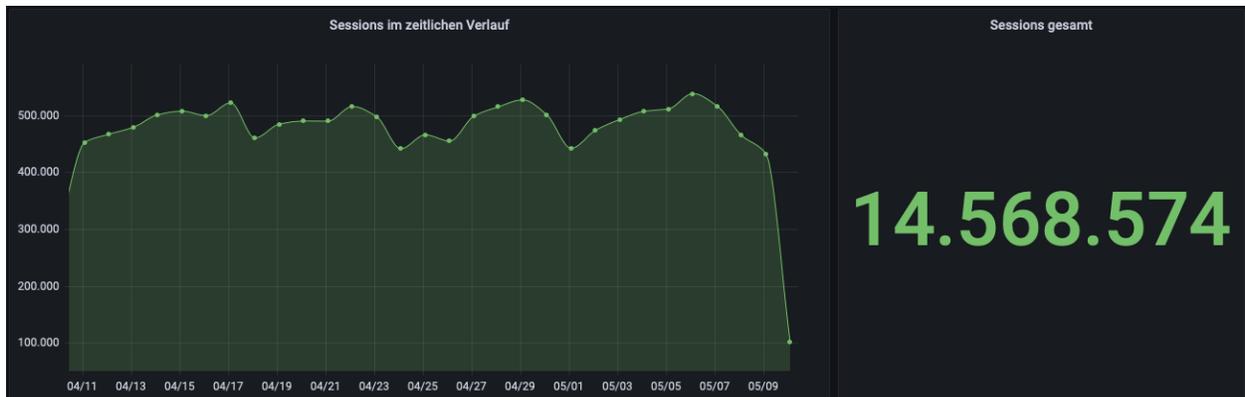


Über ein weiteres Auswahlménü lässt sich zusätzlich der Zeitraum verändern.



1.1.2 Dashboards

Sessions



Alle Hörerverbindungen, die länger als 60 Sekunden und kürzer als 24 Stunden bestehen.

Auswertungen:

- Sessions im zeitlichen Verlauf: Anzeige der Sessions pro Tag über den im Auswahlménü ausgewählten Zeitbereich
- Sessions gesamt: Anzeige der Summe aller Sessions im eingestellten Zeitbereich

Dauer pro Session



Die durchschnittliche Dauer einer Session, wobei als Session alle Verbindungen länger als 60 Sekunden und kürzer als 24 Stunden gezählt werden.

Auswertungen:

- Durchschnittliche Dauer pro Sessions im zeitlichen Verlauf: Anzeige der durchschnittliche Verweildauer pro Hörsession pro Tag über den im Auswahlmenü ausgewählten Zeibereich
- Sessions gesamt: Anzeige der durchschnittliche Verweildauer aller Sessions im eingestellten Zeitbereich

Hördauer

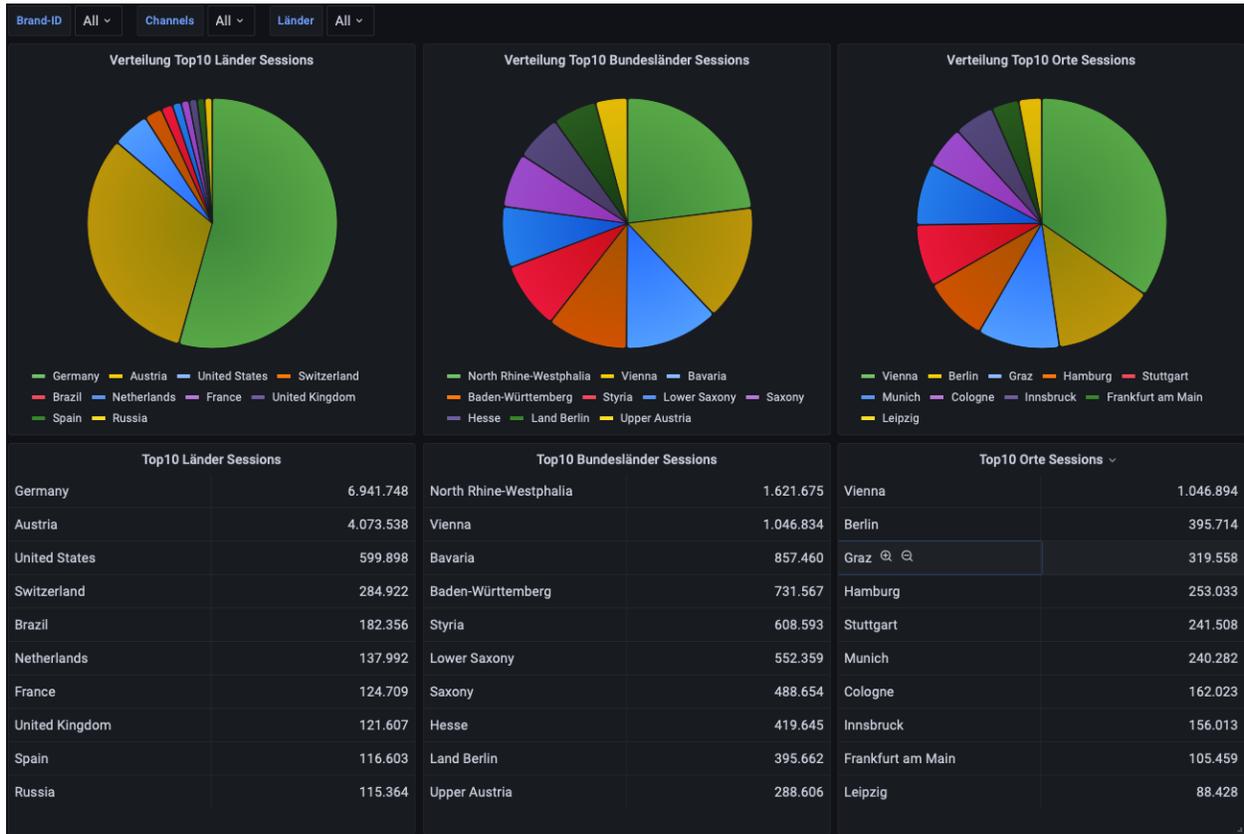


Die Hördauer (in Stunden) gibt an, wie lange die Hörer pro Tag den Channel hören.

Auswertungen:

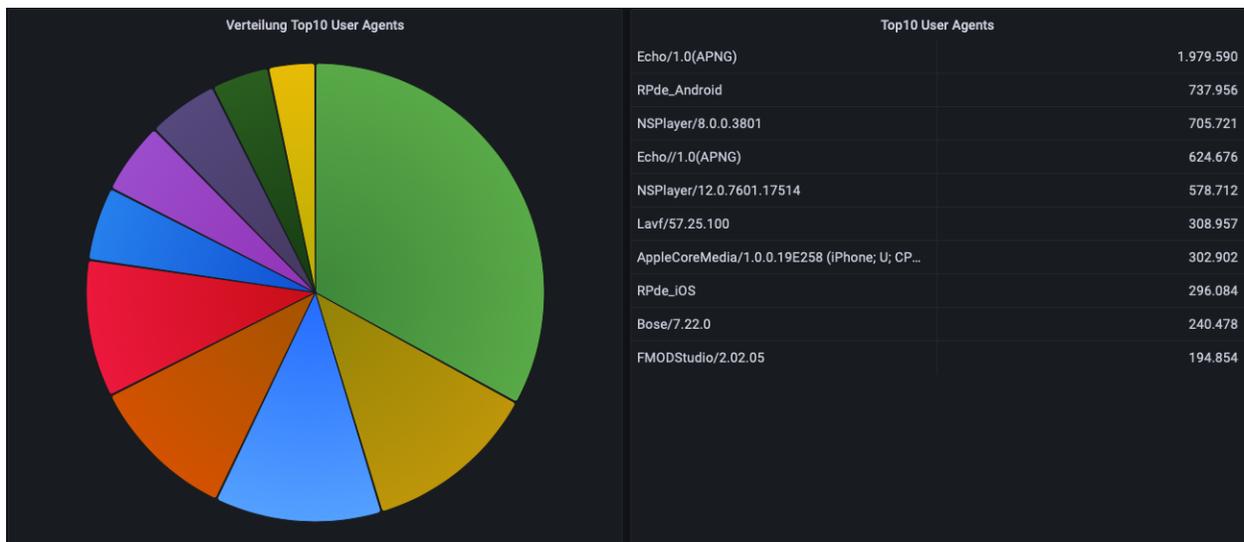
- Hördauer im zeitlichen Verlauf: Anzeige der Gesamthördauer pro Tag über den im Auswahlmenü ausgewählten Zeitraum
- Hördauer im Gesamtzeitraum: Anzeige der Summe aller Hörerstunden im eingestellten Zeitbereich

Top10 Länder, Bundesländer, Städte



Ein Überblick über die Verteilung der Hörsersessions nach Ländern, Bundesländern und Orten.

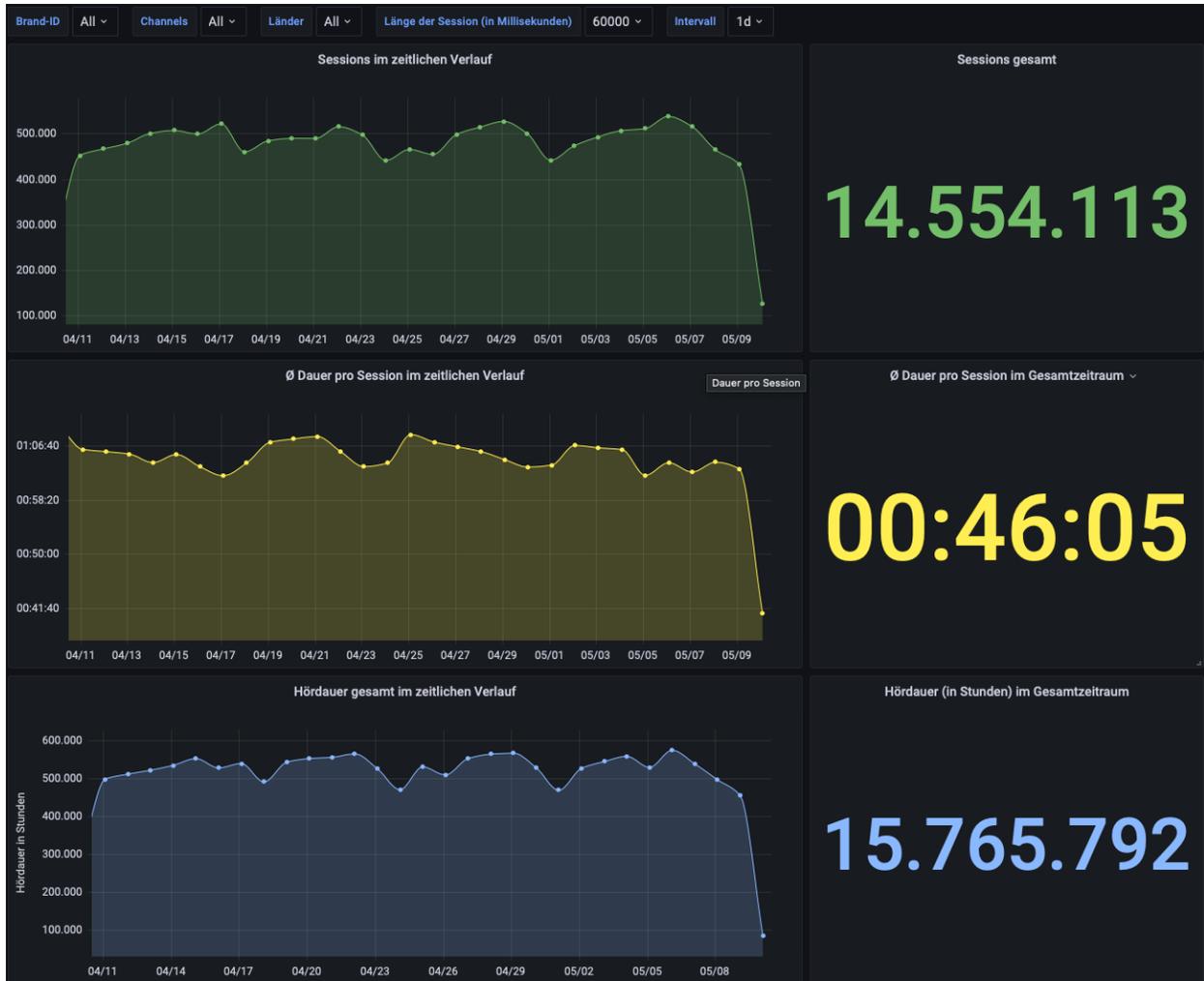
User Agents



Ein User Agent ist die Kennung, mit der sich ein Programm bei einem Webserver anmeldet. Verschiedene Geräte, wie Computer, Smartphones, Sprachassistenten, haben dabei verschiedene Kennungen. Das Dashboard „User Agents“

zeigt die Verteilung der Hörsessions auf die verschiedenen Geräte. https://de.wikipedia.org/wiki/User_Agent

Advanced Dashboard



Im Advanced Dashboard sind noch einmal die Einzel-Dashboards „Sessions“, „Dauer pro Session“ und „Hördauer“ zusammengefasst. Allerdings lässt sich hier die Anzeige noch weiter individualisieren, da man neben „Brand-ID“, „Channel“ und „Länder“ zusätzlich die „Länger der Session“ und das angezeigte „Intervall“ verändern kann.

TECHNISCHE DOKUMENTATION

2.1 Logfileimport per Logimporter

Mit Hilfe des Audalaxy-Logimporter ist es möglich die Logfiles aller gängigen Streaming-Server an das Auswertungssystem zu übermitteln. Die Daten werden vor der Übermittlung anonymisiert und bereits so aufgearbeit, dass alle Datenschutzregelungen der DSGVO eingehalten werden.

Der Audalaxy-Logimporter ist eine Software, welche als eigenständiger Systemdienst auf selbstverwalteten Streaming-Servern installiert werden kann.

Hier steht jeweils die neueste Version zum Download bereit:

- Linux 64Bit: <https://streamabc-sw.s3.eu-central-1.amazonaws.com/logimporter/logimporter-linux-amd64>
- Windows 64Bit: <https://streamabc-sw.s3.eu-central-1.amazonaws.com/logimporter/logimporter.exe>
- MacOS: <https://streamabc-sw.s3.eu-central-1.amazonaws.com/logimporter/logimporter-mac>

Der Logimporter unterstützt verschiedene Eingangsformen, Logformate und Ausgabemöglichkeiten. Die Konfiguration erfolgt über Parameter oder Environment-Variablen, die beim Programmaufruf angegeben werden müssen.

Dabei sind immer drei Bereiche nötig: “input” definiert, woher die Logs kommen (z.B. “tail” auf eine aktive Logdatei, eine einzelne vorhandene Logdatei einlesen oder mehrere), “parser” konfiguriert das zu parsende Logformat und “output” definiert wie die Daten ausgegeben werden (z.B. Ingest bei Audalaxy, Testausgabe auf Konsole).

Input, Parser und Output müssen zwingend immer beim Aufruf angegeben werden. Es muss immer ein “Origin” angegeben werden. Dieser ist eine individuelle Kennung, die Ihnen von Audalaxy mitgeteilt wird.

Der allgemeine Aufruf folgt diesem Muster:

```
logimporter input:file -path="./logs/access.log" parser:icecast output:amqp -origin=xxx
```

Hint: In den Beispielen das xxx bei `-origin=xxx` durch den jeweiligen Origin-Wert ersetzen.

Mit `-help` kann nach jedem Plugin eine kurze Hilfe über mögliche Parameter angezeigt werden.

```
./logimporter -help
```

2.1.1 Input Plugins für Logfiles

Das Input-Plugin wird mit dem Prefix “input:” angegeben.

Es gibt folgende Inputs, von denen immer nur eines verwendet werden kann:

```
input:file -path=/pfad/zur/Datei
```

Vollständiges Einlesen einer einzelnen Datei. Es muss der Pfad zur Datei angegeben werden. Es können GZ-komprimierte Dateien verwendet werden. Diese müssen die Endung .gz haben.

```
input:fileglob -path=/pfad/zu/Dateien*.log
```

Vollständiges Einlesen mehrerer Dateien. Es muss ein Pfad mit Glob-Pattern angegeben werden. Es können GZ-komprimierte Dateien verwendet werden. Diese müssen die Endung .gz haben.

```
input:tail -path=/pfad/zur/Datei.log
```

Einlesen von Logzeilen einer Datei während sie geschrieben wird. Das Programm bleibt so lange aktiv, bis es manuell gestoppt wird und sendet automatisch neue Zeilen.

In der Regel werden rotierte Logfiles erkannt, wenn der Name gleich bleibt.

Weitere Flags für *input:tail*:

```
-whence=end oder start
```

Einlesen der Datei am Anfang oder Ende beginnen (Default ist Ende, um nur neue Zeilen einzulesen).

```
-polling
```

Polling statt fsnotify benutzen, um neue Daten zu ermitteln.

```
-scanheader
```

Sollte beim Tail von AIS-Logs verwendet werden, um Felderkonfiguration zu ermitteln, die in den ersten Zeilen der Datei angegeben werden.

Hint: Zur Überwachung des Inputs bietet sich die Nutzung des *Healthcheck* an.

2.1.2 Parser Plugins für Streaming-Server

Das Parser-Plugin wird mit dem Prefix “parser:” angegeben.

Es gibt folgende Parser, es kann immer nur einer verwendet werden:

```
parser:icecast
```

Logs haben Icecast-Format.

```
parser:ais
```

Logs haben AIS Sessionlog-Format.

```
-version=ais8
```

Es gibt verschiedene Formate, die normalerweise anhand der Felder am Anfang der Datei erkannt werden. Falls es Probleme bei der Erkennung gibt, kann hier ein Format vorgegeben werden.

2.1.3 Output Plugins für die Datenübermittlung

Das Output-Plugin wird mit dem Prefix “output:” angegeben.

Es gibt folgende Outputs, von denen immer nur eines verwendet werden kann:

```
output:amqp
```

Logs werden über AMQP Messages an Audalaxy übertragen. Dies ist der Standard und sollte nur geändert werden, wenn explizit von Audalaxy gefordert.

Weitere Flags für *output:amqp*:

```
-origin=xx
```

Pflichtfeld. Diese Kennung wird von Audalaxy mitgeteilt und wird benutzt, um Zugriffe über den Logimporter korrekt zuzuordnen.

```
-streamwatch
```

Optional. Damit wird eine zusätzliche Belieferung der Audalaxy Streamwatch aktiviert. Nur verwenden, wenn von Audalaxy expliziert angefordert.

```
output:noop
```

Testausgabe “dry run”. Es werden keine Daten gesendet, es kann aber geprüft werden, ob der Import funktioniert.

Weitere Flags für *output:noop*:

```
-output
```

Logzeilen werden testweise ausgegeben.

```
output:http
```

Logs werden über HTTP an einen Ingest-Endpunkt gesendet. Alternative, nur wenn AMQP nicht funktioniert und von Audalaxy angefordert wird.

Weitere Flags für *output:http*:

```
-origin=xx
```

Pflichtfeld. Diese Kennung wird von Audalaxy mitgeteilt und wird benutzt um Zugriffe über den Logimporter korrekt zuzuordnen.

```
-streamwatch
```

Optional. Damit wird eine zusätzliche Belieferung der Audalaxy Streamwatch aktiviert. Nur verwenden, wenn von Audalaxy expliziert angefordert.

Es empfiehlt sich, den Logimporter über ein Start-Script zu verwenden, z.B. ein Systemd Unit-File. Das Programm läuft dauerhaft und versucht selbständig Verbindungen neu aufzubauen. Log-Rotation wird automatisch erkannt, wenn der Dateiname gleichbleibt.

Beim AIS unbedingt das Session-Log verwenden und nicht das Access-Log. Nur das enthält die Dauer der Verbindung. Beim Icecast dagegen das Access-Log.

Beim Logimporter dann entsprechend immer den Typ der Datei angeben.

Hier ist ein Beispiel für eine systemd Unit-Datei logimporter.service

```
[Unit]
Description=Audalaxy Logimporter
Wants=network-online.target
After=network-online.target

[Service]
ExecReload=/bin/kill -HUP $MAINPID
ExecStart=/usr/local/bin/logimporter-linux-amd64 input:tail -path=/var/log/icecast/
↳access.log parser:icecast output:amqp -origin=xxx
User=icecast
KillMode=process
KillSignal=SIGINT
LimitNOFILE=infinity
LimitNPROC=infinity
Restart=on-failure
RestartSec=2
StartLimitBurst=3
StartLimitIntervalSec=10
TasksMax=infinity

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

Angepasst an die jeweiligen Bedingungen werden muss:

- Pfad zum Programm (hier /usr/local/bin/logimporter-linux-amd64)
 - Pfad zum Log (hier /var/log/icecast/access.log)
 - User (hier icecast)
-

2.1.4 Nachliefern von Daten im Fehlerfall

Es gibt seit Version v3.1.0 zwei neue Flags `-after="YYYY-MM-DD HH:mm:ss` und `-before="YYYY-MM-DD HH:mm:ss` um nur Logs vor oder nach einem angegebenen Datum zu senden. Beide Flags können auch kombiniert werden (also nur nach einem Zeitpunkt aber zwingend vor einem weiteren). Beides sind globale Flags und müssen vor dem Input angegeben werden.

Wenn der Zeitpunkt eines Fehlers ermittelt wurde (z.B. anhand lokaler Fehlermeldungen oder im Dashboard, wenn Graphen enden), muss die Datei des Tages mit dem `-after` Flag verwendet werden, um die fehlenden Daten nachzuliefern.

Wenn man weiß, ab wann es wieder funktioniert hat (z.B. weil man da neu gestartet hat), sollte dieser Zeitpunkt als `-before` angegeben werden.

Ein vollständiger Aufruf sieht so aus:

```
logimporter -after="2022-03-04 19:15:25" input:file -path="./logs/access-2022-03-04.log.  
↪gz" parser:icecast output:amqp -origin=xxx
```

2.1.5 Healthcheck

Wenn über das „-listen“-Flag eine Binding-Adresse/Port angegeben wird, startet der Importer einen HTTP-Server und aktiviert dort einen Endpunkt „/health“.

Beispiel:

`-listen=127.0.0.1:8080` startet den HTTP-Server und bindet ihn an die IP 127.0.0.1 und den Port 8080. Mit <http://127.0.0.1:8080/health> kann der Healthcheck aufgerufen werden. Er liefert HTTP-Status 200 wenn alles ok ist. Wurden nicht genügend neue Logzeilen per *Input Plugins für Logfiles* im Interval „-alertInterval 300“ (default 300s) verarbeitet, wird Status 500 zurückgegeben.

Wird für `-listen` die IP weggelassen, hört der Healthcheck auf alle IPs, die konfiguriert sind `-listen :8080`

2.2 Logfileimport per SFTP

Hier kann der Upload der Logfiles erfolgen:

```
SFTP-Host: logimport.quantumcast.cloud  
Port: 2022
```

Benutzername und Passwort sind individuell pro Kunden und werden von Audalaxy zur Verfügung gestellt.

Die Logdateien müssen in den Hauptordner hochgeladen werden. Unterordner werden nicht unterstützt.

Die Dateien können unkomprimiert oder komprimiert (GZIP mit Dateiendung `.gz`) verarbeitet werden.

Sobald eine Datei verarbeitet wurde, wird sie automatisch in den Ordner “processed” verschoben.