SPEECHMATICS

Usage Container 0.2.0-early-access

Table of Contents

Release Notes

0.2.0

Initial release

ASR Usage Container

This document describe the details about how to download, and run the Speechmatics Usage Container to collect the transcription usage data from Speechmatics' ASR Batch and Real-time Containers.

Compatibility

To use the Usage Container, you must be running the following ASR Container versions:

- Batch Container 8.2.0 onwards
- Real-time Container 1.4.1 onwards

The ASR Usage Container has been tested using Docker Version 20. Compatibility with previous versions of Docker has not been tested.

Introduction

The Speechmatics ASR Usage Container is used to collect usage data from ASR Containers. Like the ASR Containers, these must run in customer's own environment, but require no connection to external internet.

The ASR Usage Container only collates data that is required for Speechmatics to calculate accurate financial billing and measure product usage and system performance. This data is made up of a series of events that correspond to the various stages of an Speechmatics Batch or Real-time Container as it processes a media file.

No personal customer data, transcripts or media data is captured or stored at any point.

The customer is responsible for assigning storage to the Usage Container in order to capture all usage information, and sending data to Speechmatics at regular intervals.

Terminology

Throughout this document there are references to different types of containers:

- ASR Containers - Speechmatics containers that transcribe media or audio files into a transcript. Two types are available - those can process media in batch, and those that can process media in real-time. When these are specifically referred to they are called the Batch or Real-time Containers
- Usage Containers - a new container that stores event-specific data from ASR Containers

Early Access

The Usage Container has been released as an early access product that any customer using either Speechmatics' Batch or Real-time ASR Containers is entitled to use. Speechmatics encourages customers to try this solution, in order to simplify their usage logging and reporting processes.

Speechmatics encourages feedback on the Usage Container, and the raising of any bugs or usability issues. These will be subject to our normal bug triage process, and should be submitted to support@speechmatics.com .

Workflow

The following workflow is recommended:

- The user downloads the Usage Container from Speechmatics' Docker registry using their existing credentials
    - The user must cache a copy of each container they download within their own environment
- The user will run one or multiple Usage Container(s) depending on their requirements.
    - Any Usage Container must be assigned its own persistent data volume. The user is responsible for allocating and backing up this persistent volume.
    - Any Usage Container must also have all relevant ports opened to allow data exchange and export where necessary
- When requesting transcription from a Batch or Real-time Container, the user must specify the hostname or IP address of the Usage container via a new environment variable
    - Data will then be stored in the Usage Container for up to 90 days
- At intervals of no more than a calendar month, the user will extract usage data processed in that interval from the ASR Usage Container via the RESTful API.
    - The user will then send this data to a designated Speechmatics email address (usage@speechmatics.com). Examples of automating this via scripting are present in the document

> ***[info]** Although the use of ASR Usage Container is currently optional it will become mandatory in the future.*

Reporting Cadence

Speechmatics requires customers to send all usage data by the last working day of each calendar month. You should send data for each Usage Container you have running in your environments. For customers with very large transcription volumes, more regular reporting may be recommended. Large transcription volumes can mean:

- Large number of jobs.
    - This means any Usage Container that will store data from more than 10,000 Batch jobs in a calendar month, or 1250 Real-time jobs of more than an hour
- Many jobs of long duration (>60 minutes), especially when using the Real-time Container in a 'streaming' mode where it persists between sessions

The ASR Usage Container stores the collected events for up to 90 days before loss.

Getting Started

The ASR Usage Container can be retrieved from Speechmatics docker registry as a Docker image. To access the Usage Container, you should use the same credentials that you use to access Speechmatics' ASR Containers from its Docker Registry. This information should already be provided to you by the Speechmatics support team support@speechmatics.com when you are onboarded.

You will also need to know the following information:

- Docker registry URL, e.g. `https://speechmatics-docker-public.jfrog.io`
- Image name, e.g. `asr-usage`
- Image tag, e.g. `0.2.0`

The image can be downloaded by using the standard Docker workflow:

```
# Login
docker login https://speechmatics-docker-public.jfrog.io

# Download image
docker pull speechmatics-docker-public.jfrog.io/asr-usage:0.2.0
```

> ***[info]** Speechmatics require all customers to cache a copy of the Docker images within their own environment. Please do not pull directly from the Speechmatics docker registry for each deployment.*

System Requirements

The ASR Usage Container requires the following resources:

- 1 vCPU
- 1 GB memory
- At least 1 GB of persistent storage per Usage Container deployed. Every 25 MB can store data for up to 13,000 batch jobs or up to 1250 (60 minute) realtime sessions.

Persisting storage to temporary locations (e.g. `tmpfs`) is supported where this is necessary as part of a user's workflow, but is not recommended. If you are required to use `tmpfs` or other such directories as a storage solution, Speechmatics recommends increasing the frequency of how often usage reports are sent to avoid any potential data loss

Configuration

The following section will show you how to set up an environment where you have a running ASR Usage Container that can accept data from one or multiple ASR Containers. It will show in order:

- How to set up and run a ASR Usage Container
- How to ensure an ASR Batch or Real-time Container can send all required data to a ASR Usage Container during transcription.

You must set up a Usage Container before running Speechmatics' Batch or Real-time ASR Containers in order to ensure that all usage data is captured. A Usage Container is persistent, which means it does not shut down after receiving transcription data.

Pre-Requisites

When setting up an environment with one or multiple Speechmatics ASR Container(s) and one or multiple Usage Container(s) please ensure:

- That all Batch or Real-time Containers you require to send data to the Usage Container can exchange communication with each other in their environment
- That all communication between Docker containers is via HTTPS
- That when running Usage Containers, you enable the required ports when necessary to send and extract data. More detail is below

ASR Usage Container

The ASR Usage Container always requires a persistent storage volume to store the data. This volume must be mounted inside the container at `/data`.

Endpoints

The ASR Usage Container has 2 endpoints:

endpoint	use	port	how to set
v1/log	Receives transcription event data from Batch and Real-time Containers	9090	use the `SM_EATS_URL` environment variable
v1/export	All event data, or time-specific event data, can be extracted	8000	use the `docker -p $PORT:$PORT` command. If you need to change the default port use `-e`

	from this endpoint as a compressed file		`PANDAS_PORT` environment variable as well as `docker -p` with your required port

By default all Docker containers do not expose any ports. You must specifically request these ports to be open to ensure transcription events are captured, or that data can be extracted.

The example below starts a Usage Container with

- A persistent volume mounted to `/data`
- Port 9090 open via the `EATS_PORT` environment variable to allow the Container to accept transcription event data
- Port 8000 open via the `docker -p` command to allow data to be exported from the Usage Container

```
# Create volume
docker volume create volume-1

# Mount volume
docker run -it  \
    -v volume-1:/data \
    -e EATS_PORT=9090  \
    -p 8000:8000 \
    speechmatics-docker-public.jfrog.io/asr-usage:0.2.0
```

Further documentation on using persistent storage volumes on popular container orchestration engines:

- Kubernetes https://kubernetes.io/docs/concepts/storage/persistent-volumes/
- Nomad https://www.nomadproject.io/docs/job-specification/volume

> ***[info]*** *Speechmatics recommends to set up back-up policies for the persistent volume. The ASR Usage Container cannot perform recovery by itself if the data file or volume is corrupted.*

The ASR Usage Container accepts the following configuration option, which can be set via environment variables.

Key	Default	Type	Description
`EATS_PORT`	`9090`	int	Listening port for incoming data from transcribers. Must be set to accept usage data from Batch or Real-time ASR Containers

Sending transcription data from ASR Container to ASR Usage Container

An ASR Container must be explicitly configured to send data to the ASR Usage Container when starting. By default this is via HTTPS.

The following configuration options **must** be specified when running the ASR Container to send usage data:

Key	Default	Type	Description
`SM_EATS_URL`	none	string	Address and listening port of the ASR Usage Container you wish to send data to

To correctly configure the transcriber, set `SM_EATS_URL` environment variable to point to ASR Usage Container. eg., `SM_EATS_URL=asr-usage.example.net:9090` or `SM_EATS_URL=10.244.8.32:9090`, where `asr-usage.example.net` and `10.244.8.32` correspond to the relevant ASR Usage Container instance. The port `9090` is the default listening port for incoming data from transcribers. The port number is alterable by using the `EATS_PORT` environment variable.

Below is a working example of running an ASR Batch Container that will then send transcription event data to a running ASR Usage Container:

```
docker run -i -v $AUDIO_FILE:/input.audio \
  -v $CONFIG_FILE:/config.json:ro \
  -e LICENSE_TOKEN=$TOKEN_VALUE \
  -e SM_EATS_URL=-asr-usage.example.net:9090
  batch-asr-transcriber-en:8.2.0
```

Below is a similar example of a Real-time Container that will send transcription event data to a running ASR Usage Container:

```
docker run -p 9000:9000 \
-e LICENSE_TOKEN=$TOKEN_VALUE \
-e SM_EATS_URL=asr-usage.example.net:9090 \
rt-asr-transcriber-en:1.4.0
```

Logging

The Usage Container will log event data sent by an ASR Batch or Real-time Container:

- during transcription
- when transcription has finished
    - For the Batch Container this is when transcription finishes as it is not a persistent container.
    - For the Real-time Container this is both after a endOfTranscription websocket message. The Real-time Container will send a `SESSION_ENDED` message to the Usage Container
- when the container is shut down or terminated by the user or due to system error during transcription itself (e.g. SIGTERM)

Example Logs - Success

The following is an example of a log from by a Batch or Real-time ASR Container when they successfully send data to the ASR Usage Container:

```
2021-07-19 11:24:31.314 INFO sentryserver Transcription usage registered with EATS
```

The following is an example of a log from the Usage Container when it successfully receives data from a Batch or Real-time ASR Container:

```
[2021-08-25T10:45:37Z INFO  actix_web::middleware::logger] 172.19.0.3:39068 "POST /v1/log
HTTP/1.1" 201 0 "-" "Go-http-client/1.1" 0.009459
```

The following is an example of a log from the Usage Container when a customer successfully exports data:

```
[2021-09-03T14:54:00Z INFO  actix_web::middleware::logger] 172.19.0.1:55820 "GET /v1/export
HTTP/1.1" 200 12912 "-" "curl/7.64.1" 0.006313
```

Example Logs - Failure

If data cannot be sent from the ASR Container to the ASR Usage Container, the following error message is shown in the ASR Container:

```
2021-07-19 11:27:43.158 ERROR sentryserver Error 'Post "https://asr-usage.net:9090/v1/log":
dial tcp 172.25.0.2:909: connect: connection refused' occurred when logging EATS data:
retrying
```

Example Logs - Failure upon container termination

If a container is shut down or terminated, both the Batch and the Real-time Container will attempt retries for up to 1 minute after receiving `SIGTERM`. For Batch, the Container will attempt to send data when transcription finishes. For the Real-time Container, this is when container termination is requested. After this point, any unsent data is lost with following message.

```
2021-07-19 11:28:55.288 WARNING sentryserver Some activity events could not be sent to
EATS: count: 4
```

Orchestrating multiple ASR Usage Containers

It is up to the customer's level of risk tolerance and their internal topology and orchestration how many ASR Usage Containers they need to deploy in ratio to their number of ASR Containers. Speechmatics recommends that each environment in which Batch or Real-time Containers are deployed requires at least one Usage Container. Customers can implement multiple Usage Containers in each environment for redundancy and to reduce the risk of failure

If a customer has ASR containers in multiple availability zones or clusters, assigning Usage Containers per environment or cluster reduces latency and the requirement to send messages cross cluster

Orchestrating multiple ASR Usage Containers allows redundancy in the event of network or storage failure. It is possible to deploy multiple ASR Usage Containers in a single environment and have usage data distributed to those containers. A basic scenario example is below,

The `docker-compose` example below illustrate this scenario with:

- Two Speechmatics ASR Usage Containers
- One proxy container, to route telemetry data
- One Speechmatics ASR Batch Container

```
---
#
# Example docker-compose file using multiple telemeter containers.
#
version: '3.4'

# Common setup for ASR Usage Containers
x-usage-template:
  &usage-template
  image: asr-usage:x.y.z
  labels:
    - "traefik.enable=true"
    - "traefik.tcp.routers.usage.rule=HostSNI(`*`)"
    - "traefik.tcp.routers.usage.entrypoints=custom"
    - "traefik.tcp.routers.usage.tls=true"
    - "traefik.tcp.routers.usage.tls.passthrough=true"
    - "traefik.tcp.routers.usage.service=telemeter"
    - "traefik.tcp.services.usage.loadbalancer.server.port=9090"
  depends_on:
    - proxy

services:
  # Traefik reverse proxy, to route telemetry events to multiple ASR Usage Container
  # containers
  proxy:
    image: traefik:v2.4
    command: --providers.docker --providers.docker.exposedByDefault=false --
entrypoints.custom.address=:9090
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock

  usagecontainer1:
    <<: *usage-template
    ports:
      - "8001:8000"

  usagecontainer2:
    <<: *usage-template
    ports:
      - "8002:8000"

  transcriber-batch:
    image: batch-asr-transcriber-en:x.y.z
    environment:
      SM_EATS_URL: proxy:9090
    volumes:
      - ./input/10_sec_news.wav:/input.audio
      - ./input/license.json:/license.json
    depends_on:
      - proxy
      - usagecontainer1
      - usagecontainer2
```

The example configures the ASR Batch Container to send data, using `SM_EATS_URL`, to the proxy container instead of a specific ASR Usage Container. When receiving usage data, the proxy will forward it to one ASR Usage Container, using round-robin balancing.

Each ASR Usage Container will need its own persistent storage volume to store usage data. This means that when generating reports to send to Speechmatics, an export request must be made for each ASR Usage Container the user has in operation. There will be as many reports as there are ASR Usage Containers deployed.

Sending data to Speechmatics

> ***[warning]*** *The exported data must not be modified in any way before sending to Speechmatics. Speechmatics will request a new unmodified data export if it is found that data has been altered.*

Data is retained in the Usage Container for **90 days**, after which point it is purged.

By Email

Currently Speechmatics requires data to be sent via email to usage@speechmatics.com.

The data must be exported from each ASR Usage Container you have used, and then sent to Speechmatics for calculation. The ASR Usage Container has a REST API to export transcription data. You will need to send at least as many reports as you have from ASR Usage Containers. Based on heavy transcription usage, you may have to provide multiple reports per single ASR Usage container. You can send multiple attachments per email, or each email as a separate attachment, so long as you are under email provider limits for sending files. Speechmatics recommends file sizes to not exceed 25MB. This is the default limit for sending emails for many popular providers like Microsoft Office 365. Files in excess of this size may trigger an error when sending by your email provider.

To remain under this 25MB limit, we recommend compressed files with no more than 10,000 batch jobs or 1250 Real-time sessions of one hour.

Data is exported in compressed `json.gz` format. All files must be sent in this format to Speechmatics. The name of the file does not matter.

The complete API reference for extracting usage data can be found in the API Reference section.

```
# To export all data
curl 'asr-usage.net:8000/v1/export' > ExportExampleFile.json.gz

# To export data between a date window, eg. 1-Jan-2020 to 1-Feb-2020
curl 'asr-usage.net:8000/v1/export?since=2020-01-01T00:00:00.000000Z&until=2020-02-
01T00:00:00Z' > ExportExampleFile-01-01_2020-02-01.json.gz
```

If the number of jobs extracted is too large a 4XX response may be returned. Generally this has been shown in testing to be circa. 25,000 Batch jobs or 5,000 Real-time jobs of an hour long.

In such cases, please select a smaller time window with `since` and `until` parameters.

> ***[info]** It is fine to have overlapping reports with duplicate data. Transcriptions will always be billed once; the billing cycle will be determined by their time of completion.*

The following example script exports reports by each week for the whole month:

```
#!/bin/bash

# Use ISO-8601 format
START="2021-11-01T00:00:00Z"
END="2021-12-01T00:00:00Z"
CHUNK="7 day"

d=$(date -d "$START" -I)
while [ $(date -d "$d" +%s) -le $(date -d "$END" +"%s") ]; do
  SINCE=$(date -d "$d" +"%Y-%m-%dT%H:%M:%SZ")
  d=$(date -I -d "$d + $CHUNK")
  UNTIL=$(date -d "$d" +"%Y-%m-%dT%H:%M:%SZ")

  curl "asr-usage.net:8000/v1/export?since=${SINCE}&until=${UNTIL}" > exported_$(date -d
"${SINCE}" -I)_$(date -d "${UNTIL}" -I).json.gz
done
```

Once the data has been exported, it must be emailed as attachment(s) to mailto:billing-reporting@speechmatics.com. You will receive a confirmation email within 15 minutes if the report(s) get accepted by our billing system.

> ***[danger]** Any attachment sent to Speechmatics must have the correct file name extension:* `.json.gz`.

The exported usage data is a compressed JSON file; it is possible to inspect the contents by unpacking it and opening the text file. The following example uses the jq JSON parser.

```
$ cat exported_2020-01-01_2020-02-01.json.gz | gunzip | jq .
{
  "header": {
    "alg": "HS512"
  },
  "payload": {
```

```
  "events": [
    {
      ...
```

Appendix

Data stored in ASR Usage Container

This section describes the data that is stored in ASR Usage Container:

- Audio duration information to accurately calculate billing
- Customer information including: Contract ID, Customer name and License ID
- Configuration information used by the customer to request transcription
- Audio information such as codec and format type
- System information, including certain error events

ASR Usage Container REST API

The ASR Usage Container collects usage data from ASR Containers. This data contains no sensitive information and is used by Speechmatics only for billing and improving product functionality. The following REST API can be used to export this data to be forwarded to Speechmatics.

Version: 0.2.0

Contact information:
support@speechmatics.com

Base URL: /v1

Scheme: HTTP

Port: 8000

/export

GET

Summary:
Export usage data. By default, all data is included. Use the query parameters to enforce a date range. It is possible to examine the contents of the exported data, but please do not modify it or the data will be rejected by Speechmatics, and will need to be regenerated The format of exported data can be specified in the `Accept` header, which defaults to `application/gzip`. An uncompressed version can be requested with `application/json`. You must send data to Speechmatics in `.gz` format

Parameters

Name	Located in	Description	Required	Schema
since	query	Only include data dated after (inclusive) the given time. This is a timestamp in ISO-8601 format: `YYYY-MM-DDTHH:MM:SSZ`.	No	dateTime
until	query	Only include data dated before (inclusive) the given time. This is a timestamp in ISO-8601 format: `YYYY-MM-DDTHH:MM:SSZ`.	No	dateTime

Responses

Code	Description
200	OK
400	Bad Request
500	Internal Server Error