



**SPEECHMATICS**

Batch Virtual Appliance 4.0.0

## Table of Contents

- [Batch Virtual Appliance](#)
  - [Important Notices](#)
  - [What's New](#)
    - [4.1.0](#)
    - [Known Limitations](#)
  - [Supported Platforms](#)
  - [Form Factors](#)
  - [Upgrade Path](#)
  - [Installation](#)
    - [Performance at Scale](#)
- [Batch Virtual Appliance Installation and Admin Guide](#)
- [System requirements](#)
  - [Host requirements](#)
    - [AVX flags](#)
    - [Useful links](#)
  - [Virtual Appliance system requirements](#)
    - [Real-time Virtual Appliance](#)
    - [Batch Virtual Appliance](#)
    - [Important Message on IOPS](#)
- [Downloading the appliance](#)
- [Importing the appliance](#)
  - [Note on Performance Benefits on VMWare and VirtualBox](#)
  - [VMware ESXi](#)
  - [VMware Workstation Player](#)
  - [VirtualBox](#)
  - [Amazon Web Services](#)
    - [Prerequisites](#)
    - [Uploading the OVA file to S3](#)
    - [Importing the OVA as AMI instance](#)
      - [Creating an Import Service Role](#)
      - [Creating a Role Policy](#)
      - [Importing the OVA](#)
    - [Security](#)
      - [Real-time Virtual Appliance](#)
      - [Batch Virtual Appliance](#)
    - [Launching a Virtual Appliance](#)
- [Network Configuration](#)
  - [Network interface mapping](#)
    - [VMware ESXi](#)
    - [VMware Workstation Player](#)
    - [VirtualBox](#)
  - [IP Configuration](#)
    - [Configure static IP](#)
    - [Configure DHCP IP](#)
- [Licensing](#)
  - [Licensing with the enhanced model](#)
  - [Applying an Online License](#)
  - [Checking an Appliance License](#)
    - [Example Response \(unlicensed\)](#)
    - [Example Response \(licensed\)](#)

- [Removing a License](#)
- [Using a Proxy Server](#)
- [Offline License Activation](#)
  - [Generating an Activation Certificate](#)
  - [Sending the Activation Certificate to Speechmatics](#)
  - [Applying the License Certificate](#)
- [Running an Appliance Offline](#)
- [Licensing Troubleshooting](#)
  - [Receiving Updates to a License](#)
  - [Invalid License](#)
  - [Appliance Offline](#)
  - [Offline Activation Error](#)
  - [Unable to Delete License when Offline](#)
  - [Virtual appliance is offline message when port 80 is blocked](#)
- [Verify and Go \(Batch\)](#)
- [SSL Configuration](#)
  - [Default behaviour](#)
    - [Management API Examples](#)
    - [Monitoring API Example](#)
    - [Speech API Example](#)
  - [Using your own SSL certificate and private key](#)
    - [Uploading the certificate and key to the appliance](#)
    - [Disabling HTTP access](#)
    - [Enable Basic Authentication for Admin](#)
  - [FAQs](#)
    - [How do I reset the SSL settings?](#)
    - [What if I forget the admin password?](#)
    - [What versions of SSL/TLS do you support?](#)
      - [What cipher suites do you support?](#)
- [Networking](#)
  - [Network Requirements](#)
  - [Configure Static IP](#)
  - [Configure DHCP](#)
  - [Firewall Ports](#)
  - [Using Proxies](#)
- [Virtual Appliance Scaling](#)
  - [Real-time Virtual Appliance Scaling](#)
    - [Worker Limits](#)
    - [View Maximum Workers](#)
    - [Setting Maximum Workers](#)
  - [Batch Virtual Appliance Scaling](#)
    - [Worker Limits](#)
    - [View Maximum Workers](#)
    - [Setting Maximum Workers](#)
- [Monitoring](#)
- [Services](#)
  - [Batch Virtual Appliance](#)
  - [Service status](#)
  - [Real-time Virtual Appliance](#)
  - [Service status](#)
  - [Service restart](#)
  - [Access Logs](#)
  - [System restart](#)
  - [System shutdown](#)

- [Troubleshooting](#)
  - [Transcription job failure](#)
  - [Illegal instruction errors](#)
  - [AVX2 Warning](#)
- [Console for Advanced Troubleshooting](#)
  - [License](#)
  - [Networking](#)
  - [Reboot and Shutdown](#)
  - [Security](#)
  - [Services](#)
  - [Tools](#)
  - [Workers](#)
- [Security](#)
  - [Overview](#)
  - [Ports and Protocols](#)
- [Batch Virtual Appliance](#)
  - [Overview](#)
    - [Terms](#)
    - [Getting Started](#)
    - [Audio Formats](#)
  - [Accessing the API](#)
    - [V2 API](#)
    - [File Size Limits](#)
    - [Transcription Formats](#)
    - [Troubleshooting](#)
    - [Tools](#)
      - [Windows](#)
      - [Linux](#)
      - [Mac OS X](#)
    - [Language Pack Codes](#)
- [How To Use the V2 API](#)
  - [Quick Start](#)
  - [Examples](#)
  - [Submitting a Job](#)
    - [Requesting an enhanced model](#)
  - [Checking on a Job Status](#)
    - [Checking the status of multiple submitted jobs](#)
  - [Retrieving a Transcript](#)
    - [Deleting a Job](#)
    - [Cancelling a Job](#)
  - [Configuring the V2 Speech API Request](#)
    - [Fetch URL](#)
    - [Speaker Separation \(Diarization\)](#)
      - [Speaker Diarization](#)
      - [Speaker diarization tuning](#)
    - [Speaker diarization post-processing](#)
    - [Speaker diarization timeout](#)
  - [Channel Diarization](#)
  - [Speaker Change Detection \(beta feature\)](#)
  - [Speaker Change Detection With Channel Diarization](#)
  - [Custom dictionary](#)
  - [Output Locale](#)
  - [Advanced punctuation](#)

- [Notifications](#)
  - [Configuring the Callback](#)
  - [Accepting the Callback](#)
  - [Configuring your webserver to accept the Callback](#)
- [Metadata and Job Tracking](#)
- [SubRip Subtitling Format](#)
- [Word Tagging](#)
  - [Profanity Tagging](#)
  - [Disfluency Tagging](#)
- [Getting a Job log file](#)
- [V2 API Reference](#)
  - [Version: 2.7.0](#)
  - [Terms of service](#)
  - [Contact information](#)
  - [URI scheme](#)
  - [Paths](#)
    - [/jobs](#)
      - [POST](#)
      - [GET](#)
        - [\*\*HTTP Method GET\*\*](#)
    - [/jobs/{jobid}](#)
      - [\*\*HTTP Method DELETE\*\*](#)
      - [HTTP Method GET](#)
    - [/jobs/{jobid}/transcript](#)
      - [\*\*HTTP Method GET\*\*](#)
    - [/jobs/{jobid}/log](#)
    - [Models](#)
      - [ErrorResponse](#)
      - [TrackingData](#)
      - [DataFetchConfig](#)
      - [TranscriptionConfig](#)
      - [SpeakerDiarizationConfig](#)
      - [NotificationConfig](#)
      - [OutputConfig](#)
      - [JobConfig](#)
      - [CreateJobResponse](#)
      - [JobDetails](#)
      - [RetrieveJobsResponse](#)
      - [RetrieveJobResponse](#)
      - [DeleteJobResponse](#)
      - [JobInfo](#)
      - [RecognitionMetadata](#)
      - [RecognitionDisplay](#)
      - [RecognitionAlternative](#)
      - [RecognitionResult](#)
      - [RetrieveTranscriptResponse](#)
- [Error Codes](#)
  - [4XX Errors](#)
  - [5XX Errors](#)
- [Formatting Common Entities](#)
  - [Overview](#)
  - [Supported Languages](#)
  - [Using the enable\\_entities parameter](#)
  - [Configuration example](#)

- [Different entity classes](#)
- [Output locale styling](#)
- [Example output](#)

# Batch Virtual Appliance

## Important Notices

The new enhanced model requires increased compute requirements and new recommended AVX flags. Please check the updated system requirements in the installation guide and ensure your hardware meets Speechmatics' recommendations. Otherwise you may see a slow down in processing speed when using the enhanced model. It is also now necessary to run the appliance on processors that support AVX2 in order to take advantage of latest performance optimisations for both the standard and enhanced model for all language packs.

If you are importing an appliance through VirtualBox, and AVX flags are not automatically enabled, you can also take advantage of the the performance benefits from AVX2 following [these guidelines](#).

## What's New

### 4.1.0

- Deprecation of the V1 API for Batch Virtual Appliance
- 16 Languages updated with additional punctuation marks for improved readability
  - The following languages now support ( . ? , ! ): Bulgarian, Catalan, Czech, Greek, Finnish, Croatian, Hungarian, Lithuanian, Latvian, Norwegian, Polish, Romanian, Slovak, Slovenian, Korean
- Improved accuracy for French, including more data for Canadian French (fr-ca)
- Improved accuracy for Portuguese, including more data for Brazilian Portuguese (pt-br)
- Standard operating point improved accuracy for Romanian, Hungarian, Danish, Slovakian, Croatian, Bulgarian, Finnish, Slovenian, Lithuanian
- Updated Danish, Norwegian and Swedish to remove undesired character sets
- Improved accuracy in localised spelling for English output locale feature
- Fixes for English and Italian written form numeric entities
- Improved accuracy of percentage symbol recognition in French
- New parameter added for controlling Speaker Diarization sensitivity: `speaker_sensitivity` . Refer to our [documentation here](#) for more details
- Resolved an issue where a small number of files with multiple audio channels were mistakenly detected as containing inverted audio, which lead to no transcription being returned. The check for inverted audio is now more robust

## Known Limitations

The following are known issues in this release:

Issue ID	Summary	Detailed Description and Possible Workarounds
REQ-1409	Proteus HCL with <unk> causes out of memory error	A custom dictionary list that contains the word '' causes the worker to crash.
REQ-7549	Memory leak affecting gRPC	There is a small memory leak in the gRPC Python server <a href="https://github.com/grpc/grpc/issues/5913">https://github.com/grpc/grpc/issues/5913</a> .
REQ-10160	Advanced punctuation for Spanish (es) does not contain inverted marks.	Inverted marks [ ¿ ; ] are not currently available for Spanish advanced punctuation.
REQ-10627	Double full stops when acronym is at the end of the sentence	If there is an acronym at the end of the sentence, then a double full stop will be output, for example: "team G.B."
REQ-10634	Putting "-" as an item in <code>additional_vocab</code> configuration will cause the container to fail	Do not enter just a "-" on its own in Custom Dictionary either as an additional vocab item or in the <code>sounds_like</code> property. Hyphens are still supported when entered as part of phrases or words

REQ-14402	When running very large numbers of small jobs (less than 10 seconds) offline, this may cause some of the jobs to be rejected	If you encounter this issue, please ensure licensing is in offline mode when running the appliance offline
-----------	--	--

## Supported Platforms

Virtual Appliance image (OVA) for installation on:

- VMware ESXi 6.5+ or VMware Workstation Player.
- VirtualBox 5.2+
- Amazon EC2

See the Installation and Admin Guide for details on the minimum specifications for the VM. The maximum number of concurrent jobs (maxworkers) that you can run on a single appliance is 30.

## Form Factors

Variant	Image Size	Max. Disk Space	Languages
nano	10GB	40GB	en
mini	15GB	40GB	en, de, es
midi	30GB	60GB	en, de, es, fr, ko, ja, nl, pt
maxi	52GB	80GB	en, de, es, fr, ko, ja, nl, pt, it, da, pl, ca, hi, ru, sv
plus	61GB	80GB	en, cmn, no, ar, bg, cs, el, fi, hu, hr, lt, lv, ro, sk, sl, tr, ms, id, yue

## Upgrade Path

Remove the license from your old appliance (see the Admin Guide), then re-import the new OVA and configure networking as per the Installation and Admin guide. You will need to re-apply the license code you have once the OVA has imported.

## Installation

Upload the OVA to VMWare ESX, VMWare Workstation Player, or VirtualBox. See the Installation and Admin Guide for more information.

## Performance at Scale

Further notes on IOPS requirements under heavy usage of the appliance are now provided in the System Requirements section of the Installation Guide.

## Batch Virtual Appliance Installation and Admin Guide

This guide explains how to install and administer the Batch Virtual Appliance using the Management REST API.

The Speechmatics virtual appliance is available in two modes: real-time and batch. For the most part installation and administration are identical for both modes. Where differences exist this is explicitly noted in this guide.

**Note:** Most of the code examples in these docs use HTTP rather than HTTPS to communicate with the appliance. However, we recommend using HTTPS for your production deployments. For information on SSL/HTTPS configuration, see the 'SSL Configuration' section of the docs.

## System requirements



The Speechmatics Batch Virtual Appliance operates on a hypervisor host system. For this version of the appliance, the following hypervisors are supported:

- VMware®
- VirtualBox
- AWS EC2

For the virtual appliance to operate as required, the host must meet the requirements and have the resources available as defined below.

## Host requirements

The host machine requires a processor with following microarchitecture specification:

- If using the standard model offering at least the Broadwell Class is required
- If using the enhanced model a chip that is at least the CascadeLake class is required, as this is the minimum spec that will support AVX512\_VNNI - for more information see below.
- It is also recommended if using the enhanced model that the hardware supports the AVX512\_VNNI flag, as this will greatly improve transcription processing speed
  - Examples of this among popular hosting providers include the Microsoft Azure DSV-4 class, and the Amazon M5n EC2 server class
  - If you are using the enhanced model and running on VMWare, you will have to upgrade to `hardware_version 18` to take advantage of the AVX512\_VNNI flag. Please note this is only supported by ESXi version 7.0 onwards
  - If you are using VMWare and the enhanced model, and encounter performance issues, we recommend allocating dedicated memory and/or processors to the appliance. How to apply dedicated processors in VMWare is documented [here](#), setting memory is documented [here](#)
- If you encounter performance issues when running the enhanced model, disabling hyperthreading when running the enhanced model can also improve transcription speed. How to do so when running on Amazon Web Services is shown [here](#), and for Microsoft Azure please see [here](#)

## AVX flags

The hardware you run the appliance on must support Advanced Vector Extensions (AVX). Advanced Vector Extensions are necessary to allow Speechmatics to carry out transcription:

- For the standard model, it is necessary to use at least a processor that supports at least Advanced Vector Extensions 2 (AVX2).
  - You should also ensure your hypervisor is enabled to use AVX2.
- For the enhanced model, it is recommended to run the appliance on hardware that supports the AVX512\_VNNI flag in addition to AVX2, which will substantially improve transcription processing speed.

To see what AVX flags are supported by the CPU of your host system, you can run the following query via the Management API of the appliance:

```
curl -X GET "https://{HOSTAPPLIANCE}/v1/management/cpuinfo" -H "accept: application/json"
```

You will receive information about the host CPU. Supported AVX flags will be returned as flags in the Management API response. An example is below:

```
{
  "usage_percentage": 2.5,
  "architecture": "X86_64",
  "model_name": "Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz",
  "cpus": "2",
  "vendor": "GenuineIntel",
  "hyperthreading": false,
  "flags": "3dnowprefetch abm adx aes apic arat arch_capabilities arch_perfmon avx avx2
avx512_vnni bmi1 bmi2 clflush cmov constant_tsc cpuid cpuid_fault cx16 cx8 de f16c flush_l1d fma
fpu fsgsbase fxsr hypervisor ibpb ibrs invpcid invpcid_single lahf_lm lm mca mce md_clear mmx"
```

```
movbe msr mtrr nonstop_tsc nopl nx pae pat pcid pclmulqdq pdpe1gb pge pni popcnt pse pse36 pti
rdrand rdseed rdtscp sep smap smep ss ssbd sse sse2 sse4_1 sse4_2 sse3 stibp syscall tsc
tsc_adjust tsc_deadline_timer tsc_reliable vme x2apic xsave xsaveopt xtopology"
}
```

## Useful links

See below for minimum Batch Virtual Appliance VM (guest) specifications; the host machine must have enough resources (processor, memory and storage) to run the hypervisor, the guest VMs you intend to host on it, plus any other processes you expect to run on it. Vendor guidelines should be followed for other host requirements and installation process.

For VMWare, the document Performance Best Practices for VMware vSphere® 6.0 contains a comprehensive overview of hardware considerations and recommendations on how to optimize your host platform. See <https://www.vmware.com/support.html> for up-to-date technical information on VMWare.

For VirtualBox, please consult the online documentation: <https://www.virtualbox.org/wiki/Documentation>

For Amazon EC2, the following link explains how to setup a VM using an Amazon S3 to store the OVA file: <https://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html>.

## Virtual Appliance system requirements

### Real-time Virtual Appliance

The Speechmatics Real-time Batch Virtual Appliance must be allocated the following *minimum* specification:

- 2 vCPU
- 8GB RAM
- Up to 38GB hard disk space

For each concurrent input stream using the standard model the appliance requires an additional 1 vCPU and at least 1.5GB RAM.

If you are using the custom dictionary (additional words) feature then each concurrent input stream that is configured to use it will require up to 3GB RAM.

If you are using the enhanced model, then each concurrent input stream that is configured to use it will require up to 3GB RAM. If the enhanced model is used in conjunction with other features like Custom Dictionary and encountering performance issues, then up to 5GB may be required.

### Batch Virtual Appliance

For operation in batch mode, the following *minimum* specifications are required:

- 2 vCPUs
- 8GB RAM
- Up to 44GB hard disk space

### Important Message on IOPS

Heavy usage of the appliance at scale can sometimes result in very high percentage usage of volume throughput. If this is the case, we recommend increasing the maximum IOPs supported by your hardware to a value between 8,000-12,000. This is not necessary in all circumstances, but may result in better performance if you are running more than 10 concurrent workers. Increasing the IOPS also will result in an increase in cost for resource usage. If you use AWS, setting the `volume type` to `io2` is also recommended in this scenario. How to change the maximum IOPS supported by your hardware is documented [here for AWS](#), [here for Microsoft Azure](#), and [here for VMWare](#). You may need to do this if:

- You are using close to or the maximum number of workers supported by that appliance size
- The jobs being processed are all long files, and diarization is requested

## Downloading the appliance

A download link will be provided by Speechmatics through the solutions section of the support portal (<https://support.speechmatics.com>). The latest version of the appliance can be located within the solutions section. Select the required version number within the "Batch Virtual Appliance" area that you wish to download to view the download link and all associated documentation for the virtual appliance. Once the download link is selected the download will begin, or a save file prompt will appear, enabling the file to be saved (the exact method will depend on the web browser being used). After the download a file with an ".ova" extension will be stored on the computer.

An account is required to access the documents and download link in the support portal. If an account is not available or the "Batch Virtual Appliance" section is not visible in the support portal, please contact Speechmatics Support [support@speechmatics.com](mailto:support@speechmatics.com) for help.

## Importing the appliance

Once the .ova file has been downloaded, it is ready to be imported into the host you have already prepared. Please ensure that the host meets the requirements stated earlier in this guide, then based on the hypervisor environment follow the instructions below.

### Note on Performance Benefits on VMWare and VirtualBox

To take advantage of recent Speechmatics improvements in performance using AVX2, the `hardware_version` of the Appliance has been upgraded from 9 to 11. If you are running VMWare ESXi host 6.5 or later, this should not affect any system behaviour. If you are on an earlier version, you can downgrade the hardware version as documented [here](#); however please note that this will mean you cannot take advantage of more recent optimisations in performance from using Advanced Vector Extensions 2 (AVX2).

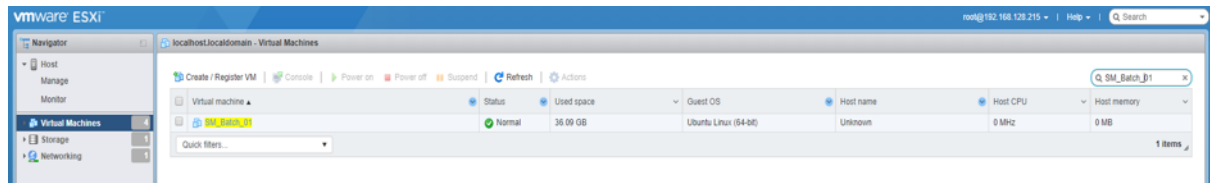
If you are running your Appliance in VirtualBox please follow [the following guidelines](#) to enable AVX2 if it is not done automatically during the importing process.

### VMware ESXi

The following steps can be used to import the virtual appliance into VMWare ESXi 6.5:

- Open the vSphere web console on the host
- Choose "Virtual Machines" from the Navigator
- Select "Create/Register VM" option
- A wizard will appear:
  - Choose "Deploy a virtual machine from an OVF or OVA file" and click "Next"
  - Enter a VM name e.g. "SM\_Batch\_01", and drag the downloaded .ova file onto the window and click "Next"
  - Select a datastore that has enough capacity to store the virtual appliance and click "Next"
  - From the "VM network" dropdown box, select a network
  - Choose Thin or Thick disk provisioning (the Speechmatics Batch Virtual Appliance supports either. Choose the options that is right for the hosting environment refer to VMWare documentation for help and click "Next"
  - Check the details are correct and click "Finish"
- The virtual appliance will import. This can take a few minutes depending on the datastore chosen.

Once the VM has imported it should be visible on the vSphere web console:



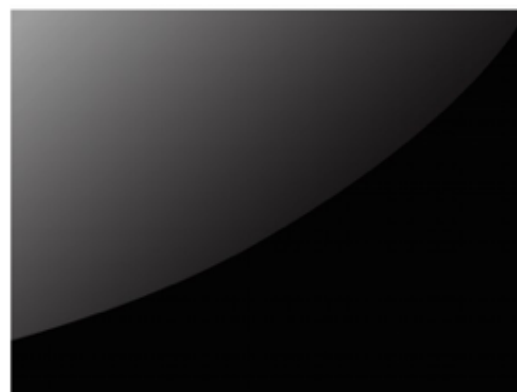
### Important Notice

If you are running VMWare ESXi version 6.5 version or above, change the `hardware_version` of the appliance to 11 to take advantage of recently implemented Speechmatics performance improvements. How to do so is documented [here](#)

## VMware Workstation Player

- Open VMware Workstation Player
- From the top options bar select "Player", then "File" and "Open..."
- The "Open Virtual Machine" window will appear. Navigate to the ".ova" file you downloaded earlier, select it, click "Open"
- Enter a VM name e.g. "SM\_Batch\_01"
- A default storage location for the virtual appliance will be shown, the can be changed if required. Click "Import".
- Dropdown box from the top options bar, click on "File"
- The virtual appliance will import. This can take a few minutes depending on the hard disk chosen

Once the VM has imported it should be visible on the Workstation player:



### SM\_Batch\_01

**State:** Powered Off

**OS:** Ubuntu 64-bit

**Version:** Workstation 9.x virtual machine

**RAM:** 5.3 GB

 Play virtual machine

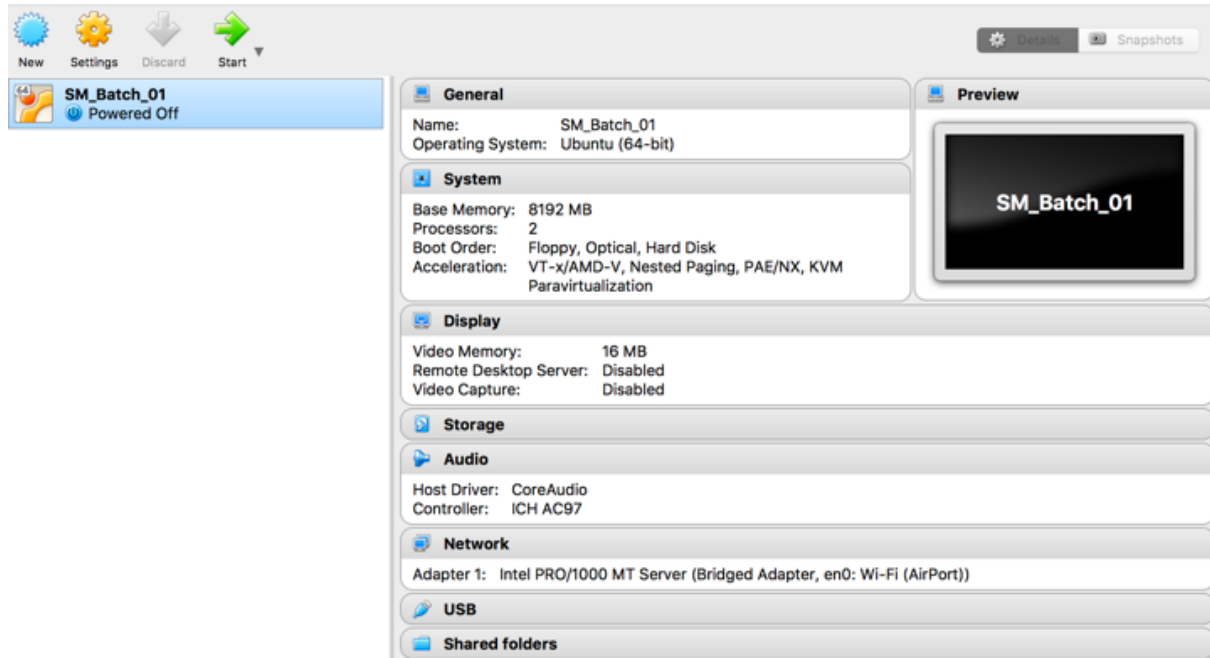
 Edit virtual machine settings

## VirtualBox

The following steps can be used to import the virtual appliance into VirtualBox 5.2 or above.

- Open VirtualBox
- From the Manager window select "File", then "Import Appliance..."
- In the Name field, name the Batch Virtual Appliance e.g. "SM\_Batch\_01"
- Browse to the OVA file and click on the "Import" button

Once the VM has imported it should be visible on the VirtualBox Manager:



## Amazon Web Services

This section explains how to create a Batch Virtual Appliance EC2 instance on the Amazon Web Services (AWS) platform by using the AWS VM Import/Export tool. This tool is designed for importing VM images from the OVA file format provided by Speechmatics. You will import the image as an Amazon Machine Image (AMI), from which you can then launch machine instances.

The information in this section is taken from the official AWS documentation and parts of it have been extracted to focus more on the particulars of the Speechmatics Batch Virtual Appliance. For more details of the Amazon VM image import process, please refer to <https://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html>

### Prerequisites

There are a few pre-requisites that you will need to have setup before you can follow the instructions in this section:

- [AWS Command Line Interface](#) (CLI)
- Python 2.6.5 or higher

Please follow the recommendations on configuration of the AWS CLI by referring to the [Getting Started](#) guide.

### Uploading the OVA file to S3

This section describes the process of uploading the Speechmatics OVA file to an Amazon S3 bucket from where it can be imported as an AMI instance. We recommend using a bucket in the same region where you want the AMI to be created and made available.

Once you've identified or created the S3 bucket on your account where the Speechmatics Batch Virtual Appliance OVA will be uploaded to, you can use any of the tools below to help with the upload of the OVA file.

- The following [AWS SDK libraries](#) support S3 multipart upload (which is the recommended method given the large size of the OVA file):
  - AWS SDK for Java
  - AWS SDK for .NET
  - AWS SDK for PHP
  - AWS SDK for Python (Boto)
  - AWS SDK for Ruby
  - You can also use the [Multipart Upload API](#) directly
- User interface tools, for instance:
  - [S3 Browser](#)
  - [CloudBerry S3 Explorer](#)

For more information about the multipart uploads, see the [AWS documentation](#).

## Importing the OVA as AMI instance

After the Virtual Appliance OVA file has been successfully uploaded to an S3 bucket, it's time to import the image.

See the AWS documentation that covers [uploading an image](#) for full details.

The steps that you will perform in this section include (in order):

- Creating a Service Role on your AWS account
- Assigning a Role Policy to this Service Role
- Importing the OVA for the Batch Virtual Appliance from the S3 bucket file

### Creating an Import Service Role

First of all, a **service role** needs to be created on your AWS account. This allows certain operations, including downloading images from an S3 bucket.

Create a file named `trust-policy.json` with the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "vmie.amazonaws.com" },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:Externalid": "vmimport"
        }
      }
    }
  ]
}
```

Then use the `create-role` command from the AWS CLI to create a role named `vmimport`. You need to specify the full path of the `trust-policy.json` file:

```
aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
```

You need to ensure that the `file://` prefix is prepended to the filename.

### Creating a Role Policy

Create a file named `role-policy.json` with the following policy. Where you see `ova-bucket` it will need to be replaced with the name of the S3 bucket where the OVA file is stored.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::ova-bucket",
        "arn:aws:s3:::ova-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:ModifySnapshotAttribute",
        "ec2:CopySnapshot",
        "ec2:RegisterImage",
        "ec2:Describe*"
      ],
      "Resource": "*"
    }
  ]
}

```

Use the `put-role-policy` command to attach the policy to the role. You must specify the full path to the location of the `role-policy.json`:

```

aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document
file://role-policy.json

```

### Importing the OVA

Importing the virtual appliance image (OVA) to Amazon EC2 as an Amazon Machine Image (AMI) is the next step.

Create a file named `containers.json` with the following content. Where you see `ova-bucket` it will need to be replaced with the name of the S3 bucket where the OVA file is stored and where you see `example-virtual-appliance.ova` it will need to be replaced with the name of the OVA file to be imported (e.g. `batch-appliance-<version>-maxi-<build-number>.ova` or `rt-appliance-<version>-maxi-<build-number>.ova`).

```

[
  {
    "Description": "Virtual Appliance OVA",
    "Format": "ova",
    "UserBucket": {
      "S3Bucket": "ova-bucket",
      "S3Key": "example-virtual-appliance.ova"
    }
  }
]

```

Use the `import-image` command to create an import task (Specify the full path to the location of the `containers.json`):

```
aws ec2 import-image --description "Virtual Appliance OVA" --disk-containers
file://containers.json
```

The resulting JSON output will show an `ImportTaskId` which you can use to check the status of the import task. You do this by running the `describe-import-image-tasks` command:

```
aws ec2 describe-import-image-tasks --import-task-ids import-ami-abcd1234
```

You need to replace the task identifier with the `ImportTaskId` for your import task ( `import-ami-abcd1234` in this example).

When the status is in the `completed` state the AMI is ready to use.

## Security

For more background on creating security groups refer to the official [AWS documentation](#). See the [Ports and Protocols](#) section for a list of the ports that are used. These ports should be opened so that you can submit jobs and manage and monitor the Speechmatics Virtual Appliance.

### Real-time Virtual Appliance

If you setup HTTPS as described in the 'SSL Configuration' section of these docs then you only need to expose port 443, **unless** you require use of the v1 WebSockets API, which requires use of port 9000. We recommend use of our updated v2 API unless you are a customer who has already implemented code against our v1 API.

### Batch Virtual Appliance

If you setup HTTPS as described in the 'SSL Configuration' section of these docs then you only need to expose port 443.

## Launching a Virtual Appliance

Now that the Virtual Appliance has been imported, it will be available as an AMI which can be launched as an instance. To launch a Speechmatics Virtual Appliance, do the following:

- Login to the AWS console and find your image under **EC2 Service | Images**
- Right-click the image and choose **Launch**
- Refer to the **System requirements** section of the Speechmatics Quick Start Guide or Admin Guide to identify how much system resources is required for your set up. Choose the instance type that meets your requirement
- Choose **Review and Launch** from the console. Setup the Key Pair if required and choose **Launch** again.

Full instructions for launching instances can be found here:

<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/launching-instance.html>

## Network Configuration

Before starting the virtual appliance for the first time, it is important to consider the network settings that will be used. The section below describes the options.

### Network interface mapping

Whilst the virtual appliance is powered off, the virtual network adaptor should be mapped to the correct physical adaptor on the host. The virtual interface must be mapped to a physical adaptor on which the Speechmatics Batch Virtual Appliance will be contacted. Steps are provided below for the supported hypervisors.

#### VMware ESXi

There is nothing to configure here. The network as specified during the import stage described above will be used.

#### VMware Workstation Player

Speechmatics recommends using bridged network mode. To ensure bridged networking is selected:



- Open VMware Workstation Player
- Right click on the virtual appliance e.g. "SM\_App\_01", and select "Settings..."
- Select the "Network Adapter" in the devices list
  - Select "Bridged: Connected directly to the physical network"
- Click "OK"

This will result in the VM using an IP Address for its use that is independent from that of the host.

## VirtualBox

Speechmatics recommends using bridged network mode. To ensure bridged networking is selected:

- Open VirtualBox
- Right click on the virtual appliance and select "Settings..."
- Select "Network" and from the "Attached to:" dropdown box, select "Bridged Adaptor"
- Click "OK"

This will result in the VM using an IP Address for its use that is independent from that of the host.

## IP Configuration

When the Speechmatics Batch Virtual Appliance is started, the default behavior will be to dynamically acquire an IP address. If there is no DHCP service available on the network, it will fall back to an IP address automatically assigned.

The IP address information can be viewed by opening the virtual appliance console once it has booted, as shown below.

The screenshot shows a terminal window titled "rt-appliance-3.1.0-mini-46373 [Running]". The terminal output is as follows:

```

Welcome
=====
This host is only accessible via the management (port 8080) and speech APIs.

Version : 3.1.0

192.168.128.17
10.10.10.2
fe80::a00:27ff:fed7:bddf

ubuntu login: _
  
```

The terminal window also shows a standard Ubuntu desktop environment at the bottom with various system icons and a taskbar.

The screen shot above shows the 10.10.10.2 IP address as the fallback address. The other address shown was allocated by DHCP and should be used for all communication.

If DHCP cannot be used, a static IP address can be configured as described below.

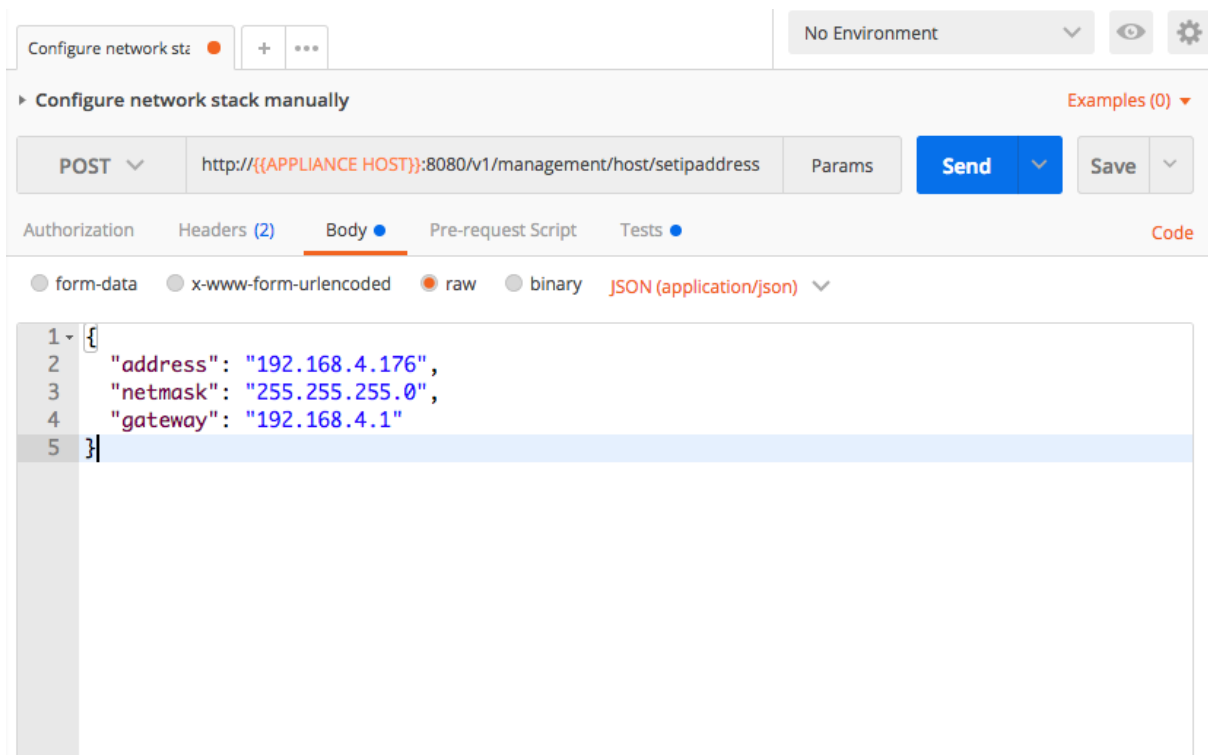
## Configure static IP

To configure a static IP address, the Management REST API for the virtual appliance is used. The following information is required:

- **Method:** POST
- **URL:**  
http://\${APPLIANCE\_HOST}:8080/v1/management/host/setipaddress
- **Body Format:** JSON
- **Body:** address, netmask, gateway, nameservers

Where \${APPLIANCE\_HOST} is the hostname or IP address of your Batch Virtual Appliance.

The example below shows use of [Postman](#) (available for free from the Chrome web store) to POST new IP settings.



You can optionally specify a list of nameservers to use (if none are specified then, 8.8.8.8 is used), for example this time using [curl](#) from the command-line to make the POST request:

```
curl -L -X POST 'http://${APPLIANCE_HOST}:8080/v1/management/host/setipaddress' \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-d '@network-config.json'
```

In this example, a local file network-config.json is used for the JSON configuration:

```
{
  "address": "192.168.128.96",
  "netmask": "255.255.255.0",
  "gateway": "192.168.4.1",
  "nameservers": ["208.67.222.222", "208.67.220.220"]
}
```

**NOTE:** once the POST is sent, the virtual appliance will automatically reboot. Check the console to verify the new IP address has been applied.

## Configure DHCP IP

To configure a dynamic IP address using DHCP, the admin REST API is used as follows:

- **Method:** POST
- **URL:**  
http://\${APPLIANCE\_HOST}:8080/v1/management/host/setdhcp
- **Body format:** JSON

The example below shows how to use Postman to POST to the REST API in order to configure a DHCP address.

```
curl -L -X POST 'http://${APPLIANCE_HOST}:8080/v1/management/host/setdhcp' \
-H 'Accept: application/json'
```

**NOTE:** once submitted, the virtual appliance will automatically reboot. Check the console to verify the new IP address has taken affect.

## Licensing

The Speechmatics Batch Virtual Appliance uses two licensing options: an online cloud-based licensing mechanism, and offline licensing. The online license requires that the appliance must be connected to the Internet in order to activate the license, and then while running will need connection to the external internet via Port 80.

An offline license allows a user to apply a license without ever connecting an appliance to the external internet, even to initially license the appliance. Users generate an Activation Certificate via the Management API, which is then sent to Speechmatics Support to generate a separate license certificate. The license certificate can be used to then successfully generate transcription from the offline appliance. How to do so is described in more detail below.

Your appliance must have been activated with a valid license before the Speech API can be used. Use of the Management API does not require a license. Please contact Speechmatics support [support@speechmatics.com](mailto:support@speechmatics.com) if you do not have a license.

You can only apply to one license to an appliance at a time. If you want to apply a new license, you must first remove the old license, and then apply a new license as shown [here](#).

## Licensing with the enhanced model

If you are using both the standard and the enhanced model interchangeably, please note that you will need two separate licenses, one for standard appliances which will be entitled to use a standard model, and one for appliances that will be entitled to use a standard and enhanced model. You should ensure in your routing logic that jobs using the standard model are sent to the appliance which only uses the standard model. Otherwise, there is a risk of overbilling. You will also require a new license to use the enhanced model.

If you are not certain whether you are entitled to use the enhanced model, please check with your account manager.

## Applying an Online License

To apply a license that you have received from Speechmatics you use a POST request to the Management API. If your license supports fully offline activation and your appliance has no route to the Internet, then you should follow the instructions in the section on [Offline License Activation](#) later on. Otherwise keep reading this section.

Assuming your appliance is deployed in a network that has a route to the Internet you can make the activations request to the `/v1/management/license` endpoint as follows:

- **Method:** POST
- **URL:**  
`http://${APPLIANCE_HOST}:8080/v1/management/license`
- **Body format:** JSON
- **Body:** `license`, `username`, `email_address`, `company_name`

You must supply the `license` value. The other fields (`username`, `email_address` and `company_name`) are optional, but we recommend that you fill them in with your details to help in case of support issues.

**Note:** make sure when applying the license, that all the appliance services are running; otherwise the activation will fail.

The example below shows how to make an activation call using the Management REST API:

```
curl -L -X POST "http://${APPLIANCE_HOST}:8080/v1/management/license" \  
-H 'Accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
  "license": "494953679762904933",  
  "username": "Amy Liu",  
  "email_address": "a-liu@example.com",  
  "company_name": "Example Pty"  
}' \  
| jq
```

The response should indicate that the licensed status is true. The licensing activation requires a connection to the Internet (using TCP port 80). Blocking this port with an online license can cause transcription issues. If you are behind a corporate firewall that does not allow a direct connection to the Internet then you can configure the appliance to use a proxy server to allow you to license the appliance as shown in the documentation [here](#).

## Checking an Appliance License

You can check whether the appliance is licensed by using a GET request to the `/license` endpoint on the Management API. For example:

```
curl -L -X GET "http://${APPLIANCE_HOST}:8080/v1/management/license" \  
-H 'Accept: application/json' \  
| jq
```

### Example Response (unlicensed)

If the appliance has *not* been licensed then you will see something like this:

```
{  
  "licensed": false,  
  "product": "103",  
  "subscription_expiry": "1970-01-01T00:00:00Z",  
  "status": "-113 License status server trial expired",  
  "message": "",  
  "transcription_minutes_allowed": "0",  
  "license_type": "",  
}
```

```
"activation_type": "",
"transcription_secs_allowed": 0,
"transcription_secs_allocated": 1800000,
"connected": true,
"customer_id": 4920,
"license_code": ""
}
```

The `licensed` property is `false`, and the `license_code` property is empty, indicating that the appliance has not yet been activated with a valid license code. The appliance can be managed through the Management API whilst in this state, but any attempt to use the Speech API to transcribe speech will return a `not authorised` error with the reason "You are not authorised to perform this action: License is invalid".

### Example Response (licensed)

If the appliance *has* been licensed then you will see a return like this:

```
{
  "licensed": true,
  "product": "103",
  "subscription_expiry": "2021-10-16T12:26:37Z",
  "status": "3 License status concurrent license",
  "message": "",
  "transcription_minutes_allowed": "60",
  "license_type": "6 License is concurrent subscription",
  "activation_type": "1 License was activated online",
  "transcription_secs_allowed": 3600,
  "transcription_secs_allocated": 1800000,
  "connected": true,
  "customer_id": 4949,
  "license_code": "494913586168289666"
}
```

This shows that the appliance has been licensed with code 494913586168289666. The license is due to expire on the 16th October 2021. 500 hours (1800000 seconds) have been allocated for use. A local cache of 1 hour (3600 seconds) is maintained on the appliance to allow it to operate offline for limited periods of time; both of these values will be defined depending on your license requirements.

## Removing a License

If you no longer wish to use the appliance, or you need to perform an upgrade to a newer version then you should first remove the license before powering down the virtual appliance. This will return any unused audio hours that you may have cached on the appliance. You must be online to perform this action. Once the license is removed the appliance will no longer be able to transcribe speech. You use an HTTP DELETE to remove the license:

```
curl -L -X DELETE "http://${APPLIANCE_HOST}:8080/v1/management/license" \
  -H 'Accept: application/json'
  | jq
```

The response shows the number of unused seconds that were returned for use by future activations, for instance:

```
{
  "transcription_secs_returned": 60780
}
```

## Using a Proxy Server

The appliance needs to talk to the cloud licensing service using HTTP (TCP port 80). The license credentials are encrypted over this link. If the network the appliance is installed on uses a proxy server to access the Internet, then you will need to configure the appliance to use that proxy. This is a pre-requisite before attempting to apply the license.

To configure the appliance, use a POST to the `/v1/management/license/network` endpoint:

```
curl -X POST "http://${APPLIANCE_HOST}:8080/v1/management/license/network" \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-d '{ "http_configuration": {
  "ip": "${PROXY_HOST}",
  "port": "${PROXY_PORT}",
  "user": "${PROXY_USERNAME}",
  "password": "${PROXY_PASSWORD}"
} }' \
| jq
```

Where `${PROXY_HOST}` is the IP address or hostname of your proxy server, and `${PROXY_PORT}` is the port number it uses. If you use username and password authentication for the proxy server, then these also need to be specified using the `${PROXY_USERNAME}` and `${PROXY_PASSWORD}` options. If the proxy server does not require authentication then they should be left out.

Once you have configured the Batch Virtual Appliance to use your proxy server you will be able to activate the appliance – see the [section above](#) on how to apply a new license.

## Offline License Activation

This section explains how to license your appliance if it is in a completely offline environment (ie. there is no route to the Internet), and you are not able to connect to the Internet even during initial activation of the license. If this is the case then you need to generate an *activation certificate*, send this to [support@speechmatics.com](mailto:support@speechmatics.com), and then apply the *license certificate* that is sent back. Follow the steps in this section to do this.

It is recommended as part of best practice where the appliance cannot connect to the internet to activate the license offline using the method below.

### Generating an Activation Certificate

The process is similar to online activation: you will receive a license code from Speechmatics support. However, additional steps are required, and different endpoints on the Management API are used.

Offline activations require a POST request to the `/v1/management/license/offlineactivation` endpoint:

- **Method:** POST
- **URL:**  
`http://${APPLIANCE_HOST}:8080/v1/management/license/offlineactivation`
- **Body format:** JSON
- **Body:** `license`, `username`, `email_address`, `company_name`

**Note:** make sure when applying the license, that all the appliance services are running; otherwise the activation will fail.

The example below shows how to make an offline activation call using the a POST request to the Management REST API:

```
curl -L -X POST "http://${APPLIANCE_HOST}:8080/v1/management/license/offlineactivation" \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "license": "494853989762904933",
  "username": "Fiona Kelly",
  "email_address": "fjk@example.com"
}
```

```
}' \  
| jq
```

## Sending the Activation Certificate to Speechmatics

The response contains a long string of alphanumeric characters. This is the activation certificate. You should save this as a text file and send to Speechmatics support [support@speechmatics.com](mailto:support@speechmatics.com), along with the license code that you used.

Once this has been done, the support team will use the activation certificate to generate a license certificate. They will then send this back to you by reply of email.

**Note:** You should make sure, when in the process of applying an offline license, that you do not reboot the appliance between generating the activation certificate and applying the license certificate. The reason being that the computer identifier which is used as a component of the certificates will be different between reboots.

## Applying the License Certificate

Once you have been sent the license certificate by Speechmatics support you can use this to activate the appliance by making a **PUT** request to the `/v1/management/license/offlineactivation` endpoint:

- **Method:** PUT
- **URL:**  
`http://${APPLIANCE_HOST}:8080/v1/management/license/offlineactivation`
- **Body format:** JSON
- **Body:** `license`, `certificate`

Where `certificate` is the license certificate that Speechmatics support have provided to you, and `license` is the original activation license code you received.

**Note:** It is important that you use the **PUT** method to send the license certificate to the appliance. If you use the **POST** method you will end up with an error and will need to repeat the license activation process.

The example below shows how to make this activation call:

```
curl -L -X PUT "http://${APPLIANCE_HOST}:8080/v1/management/license/offlineactivation" \  
-H 'Accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
  "license": "494853989762904933",  
  "certificate": "7bc3c6684...f46d9781cfbae3c8129505e"  
}' \  
| jq
```

**Note:** The certificate is a very long string of alphanumeric characters. We shorten it here for brevity.

Once the appliance has been licensed in this way you will see a return like this:

```
{  
  "licensed": true,  
  "product": "100",  
  "subscription_expiry": "2019-04-07T14:35:09Z",  
  "status": "3 License status concurrent license",  
  "message": "",  
  "transcription_minutes_allowed": "2999",  
  "license_type": "6 License is concurrent subscription",  
  "activation_type": "1 License was activated online",  
  "transcription_secs_allowed": 179998,  
  "transcription_secs_allocated": 2147483647,  
  "connected": false,  
  "customer_id": 4948,  
  "license_code": "494853989762904933",  
}
```

```
"computer_id": "bc8e7e32-7cc6-4292-83e5-555c726ae8d8",
"error_message": ""
}
```

## Running an Appliance Offline

If you have activated your license offline, and have already processed the activation certificate using the steps described above, your virtual appliance will go into offline mode automatically, and will no longer need to talk to an external server.

If you have activated the appliance online, but are then expecting to run it in a completely 'dark' network (that is, your appliance will not have any route to the Internet), then after you have activated the license online, then, *after* licensing you should put it into an *offline* mode. This is a quicker way of activating the license where the appliance can be online for license activation only. Running the appliance without an internet connection Activating and deactivating your license must still be done when your appliance is online. We recommend that you run licensing in offline mode if the appliance will not have any route to the internet as best practice.

Enabling offline mode can be done through the Management API by setting the `offline` state to `true`, like this:

```
curl -X POST "http://${APPLIANCE_HOST}:8080/v1/management/license/offlinemode" \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-d '{ "offline": true}' \
| jq
```

**Note:** You can only use this mode if your license allows offline mode to be used. If offline mode is not supported by your license you will see an error message returned:

```
{
  "error": "Error happened in the license_management service: ModelException: Unknown Rpc Error.
Status code=StatusCode.UNKNOWN: <_Rendezvous of RPC that terminated with:\n\tstatus =
StatusCode.UNKNOWN\n\tetails = \"Offline mode not enabled for this
license\"\n\tdebug_error_string = \"
{\\\"created\\\":\\\"@1544116759.516901870\\\",\\\"description\\\":\\\"Error received from
peer\\\",\\\"file\\\":\\\"src/core/lib/surface/call.cc\\\",\\\"file_line\\\":1099,\\\"grpc_message\\\":\\\"Offline
mode not enabled for this license\\\",\\\"grpc_status\\\":2}\\\"\\n>\",
  \"code\": 13
}
```

**Note:** Putting the appliance into offline mode like this stops any online license checking, which means that any changes to your license will not be picked up until you disable offline mode and go back online.

If you think that you may need to run your appliance in offline mode then please contact [support@speechmatics.com](mailto:support@speechmatics.com).

## Licensing Troubleshooting

### Receiving Updates to a License

In the case where Speechmatics has provided an update to an already activated license (for example, to increase the number of license activations, or the amount of audio hours allowed), then you will need to restart the services on your appliance, and ensure that the appliance is online when you do so, in order for the license updates to take effect.

### Invalid License

If you attempt to activate your virtual appliance by applying a license code and you see this error message, then it means that the license code is invalid.

```
Input exception: Not activated
```



If this occurs please contact [support@speechmatics.com](mailto:support@speechmatics.com), sending back the full output from the activate license POST request.

## Appliance Offline

If you see the following error when attempting to activate the appliance:

```
Input exception: Not activated - Cannot activate when offline
```

Then the appliance is unable to contact the cloud license service. Make sure that you are able to reach the licensing service by pinging the following hostname: my.nalpeiron.com. If you use a proxy server to connect to the Internet, ensure that the appliance has been configured to use the proxy (making sure that you specify at least the IP address or hostname of the proxy, and the correct port number). Look in the logs of your proxy server to check that the appliance is using the correct proxy server. To check whether you are running online or not you can run the following:

```
curl -L -X GET "http://${APPLIANCE_HOST}:8080/v1/management/license" \
  -H 'Accept: application/json' \
  | jq '.connected'
```

## Offline Activation Error

If you are carrying out activation offline and you use the **POST** method rather than a **PUT** to send the license certificate to the appliance you will see a `"code": 13` error with a description message of `"Unable to request activation certificate: -1121"`. The full error response will look like this:

```
{
  "error": "Error happened in the license_management service: ModelException: Unknown Rpc Error.
  Status code=StatusCode.UNKNOWN: <_Rendezvous of RPC that terminated with:\n\tstatus =
  StatusCode.UNKNOWN\n\tetails = \"Unable to request activation certificate:
  -1121\"\n\tdebug_error_string = \"{\n\t\t\"created\": \"@1738245024.927287214\", \"description\": \"Error
  received from
  peer\", \"file\": \"src/core/lib/surface/call.cc\", \"file_line\": 1036, \"grpc_message\": \"Unable to
  request activation certificate: -1121\", \"grpc_status\": 2}\n\t\",
  \"code\": 13
}
```

## Unable to Delete License when Offline

If you have activated a license online, but you then go into offline mode, you will get an error if you attempt to remove the license (`DELETE /v1/management/license`).

```
{
  "error": "Error happened in the license_management service: ModelException: Unknown Rpc Error.
  Status code=StatusCode.UNKNOWN: <_Rendezvous of RPC that terminated with:\n\tstatus =
  StatusCode.UNKNOWN\n\tetails = \"Cannot remove license in offline mode\"\n\tdebug_error_string =
  \"{\n\t\t\"created\": \"@1738245024.927287214\", \"description\": \"Error received from
  peer\", \"file\": \"src/core/lib/surface/call.cc\", \"file_line\": 1036, \"grpc_message\": \"Cannot
  remove license in offline mode\", \"grpc_status\": 2}\n\t\",
  \"code\": 13
}
```

In order to remove the license you will need to exit offline mode, and make sure that there is a route to the Internet before trying to remove the license.

## Virtual appliance is offline message when port 80 is blocked

Communication with the cloud license service relies on port 80 being open. If there is a firewall in your network that blocks port 80 then you will see error messages like this when attempting to make a licensing call:

```
{
  "error": "Error happened in the license_management service: ModelException: Unknown Rpc Error.
Status code=StatusCode.CANCELLED: <_Rendezvous of RPC that terminated with:\n\tstatus =
StatusCode.CANCELLED\n\tetails = \"Received RST_STREAM with error code 8\"\n\tdebug_error_string
= \"{\n\t\"created\": \"@1546953904.251876385\", \"description\": \"Error received from
peer\", \"file\": \"src/core/lib/surface/call.cc\", \"file_line\": 1099, \"grpc_message\": \"Received
RST_STREAM with error code 8\", \"grpc_status\": 1}\n\t\">\",
  "code": 13
}
```

In such cases make sure that port 80 is open, or use configure the appliance to use a proxy server.

## Verify and Go (Batch)

This section explains how to verify the correct operation of the Batch Virtual Appliance using the REST Speech API.

Check that all the Speechmatics services within the appliance are up and running before passing the audio file. The Management REST API can be used for this.

- **Method:** GET
- **URL:**  
http://\${APPLIANCE\_HOST}:8080/v1/management/services

To run a simple transcription job to test that everything is working use the Batch Virtual Appliance Speech API (on port 8082)

- **Method:** POST
- **URL:**  
http://\${APPLIANCE\_HOST}:8082/v2/jobs

For example, you can use the following Speech API request using the curl command-line tool to transcribe an audio file 'sample.wav' and return the Job ID:

```
curl -s -L -X POST 'https://${APPLIANCE_HOST}/v2/jobs/' \
-F data_file=@sample.wav \
-d 'config={ "type": "transcription",
  "transcription_config": { "language": "en" }
}' \
| jq
```

Where \${APPLIANCE\_HOST} is the hostname or IP address of your virtual appliance. The above assumes that sample.wav contains English speech; modify the language identifier in the job config to match the language you want to transcribe.

You can use the Job ID to get the status of the job:

```
curl -s -L -X GET 'https://${APPLIANCE_HOST}/v2/jobs/${JOB_ID}/' \
| jq
```

Where \${JOB\_ID} is the Job ID (id field) that was returned when you submitted the job. Once the job is done, you use the Job ID to return the transcription:

```
curl -s -L -X GET "https://${APPLIANCE_HOST}/v2/jobs/transcript" \
| jq
```

Under normal conditions, the job should take less than half the duration of the media file to process. So for example if you submit a MP3 file that is 60 minutes long, its transcription should be processed in less than 30 minutes. See the REST Speech API Guide for the list of language codes, how to use features of the API, the output formats that are supported, as well as more usage examples.

The Speechmatics Batch Virtual Appliance is now ready to use.

## SSL Configuration

When the appliance is imported it contains a default self-signed certificate, so you can use HTTPS to access the appliance via the Management, Monitoring and Speech APIs. However, we recommend replacing this default SSL certificate with your own certificate, signed by your organisation or a trusted third-party certificate authority (CA).

### Default behaviour

By default, our appliances allow connections over HTTP. The services on the appliance expose several ports for HTTP access, such as 8080 for the management API and 3000 for the monitoring API.

Since version 3.4.0 of the appliances, we also support HTTPS access to these services over port 443. To use HTTPS simply change the protocol used for API calls from `'http'` to `'https'`, and remove the port from the URL. If you are copying the examples from this document you can set the `$APPLIANCE_HOST` environment variable like this: `export APPLIANCE_HOST=localhost`.

### Management API Examples

```
curl -L -X GET "http://${APPLIANCE_HOST}:8080/v1/management/services" \  
-H 'Accept: application/json'
```

To modify this to use a secure API call, change `http://` to `https://` and remove the port number `:8080` from the URL:

```
curl -L -X GET "https://${APPLIANCE_HOST}/v1/management/services" \  
-H 'Accept: application/json'
```

**Note:** If you are using a self-signed certificate (your own, or the Speechmatics certificate that is used by default), then you will see a warning like this when using the above curl command:

```
curl: (60) SSL certificate problem: self signed certificate
```

**Warning:** The default SSL certificate on the appliance is a self-signed certificate created by Speechmatics, which is not signed by any certificate authority. Your HTTP client or web browser may warn that this is insecure. This warning can be suppressed, for example with cURL by adding the `--insecure` flag, however customers who are serious about security should not be using the self-signed certificate. We recommend uploading your own SSL certificate to the appliance. Instructions for doing this can be found below.

**Important:** We have added `--insecure` to some of them cURL examples in this document so that the command trusts the self signed certificate. You won't need this option once you've uploaded your own certificate and configured your own system to trust it.

### Monitoring API Example

With access to the Monitoring API (available on port 3000 if you are using HTTP) you will need to prefix the endpoint with `/monitor`. For example:

```
curl --insecure -L -X GET "https://${APPLIANCE_HOST}/monitor/api/3/mem"
```

### Speech API Example

Access to the REST Speech API (available on port 8082 using HTTP), is also possible via HTTPS:

```
curl --insecure -L -X GET "https://${APPLIANCE_HOST}/v1.0/user/1/jobs/"
```

## Using your own SSL certificate and private key

To use your own SSL certificate you'll need to upload your *certificate* file as well as the associated *private key* file.

- The **private key** file normally has a '.key' extension and should look similar to the example below.

```
-----BEGIN RSA PRIVATE KEY-----
xqgLwi4gJ9+9Qkavpk3WpPFTTYUfVrCJNviKEn5wAltutqLQkRTcxJtrEk8trKI
fCxeZo35yVhYmDGUIuAdAcPRTPj0XZkXQRhkITmD8TYMc/sVlJpFr+TAssGzute8
... 21 lines redacted ...
+bLv4aqI9tZrwpyeziaOuyQRhYodpAjhCyCFMkJjY59BKv/cqMHx8FPDQmaZ9Xs0
SmE9JAKnDgF5yLHm1Q6WZ1/L/M4SkgIqEglF7ifLd5M3wskpmHia6/f8Fa2KwbBJ
-----END RSA PRIVATE KEY-----
```

**Note:** We do not currently support encrypted/password protected private key files.

- The **certificate** file should be PEM encoded and normally has a '.crt' or '.pem' extension. It should look similar to this:

```
-----BEGIN CERTIFICATE-----
MIIGuzCCBaOgAwIBAgIIIIHlfyznYUA8wDQYJKoZIhvcNAQELBQAwbQxwCzAJBgNV
BAYTA1VTMRAwDgYDVQQIEwdBcm16b25hMRRMwEQYDVQQHEwptY290dHNkYXxlMR0w
... 32 lines redacted ...
P4LMbjCA4mqQvlipeSANIE40rFL47zLcy+H9M0+Rw2CUiwL8QZFq+TAiIZ34tC
UVCh52xpB9/BhO++QbGdlzObqDhcGEg8pJpJIycej9t4GN1eqNSudn0ibsQWew8=
-----END CERTIFICATE-----
```

Both files should be in [PKCS8](#) format. If you have to upload a certificate chain, then the file you upload should contain the individual certificates concatenated, with your organisation's certificate first.

### Uploading the certificate and key to the appliance

To upload your own certificate to the appliance you will need to make a POST request to the `/v1/security/sslcertificate` endpoint. This can be done using an HTTP client on the command line or with the management interface in a browser.

With the example shown here set `APPLIANCE_HOST` as appropriate (e.g. `export APPLIANCE_HOST=localhost` if your appliance is running locally):

```
curl --insecure -X POST "https://${APPLIANCE_HOST}/v1/security/sslcertificate" \
-F "keyfile=@appliance.key" -F "certfile=@appliance.crt"
```

**Warning:** Do not upload these files over HTTP, or you risk leaking the private key for your certificate.

If the upload succeeds then you should receive an HTTP 200 response with a success message:

```
{
  "success": true,
  "message": "certificate and private_key applied successfully"
}
```

Be aware that setting a new certificate will cause the web server in the appliance to restart which can take around five seconds. During this period, requests will still be served, however the old certificate will be used. Existing connections such as job uploads or WebSocket streams will not be interrupted.

You can check the certificate on the appliance by using the `openssl` tool:

```
$ openssl s_client -connect ${APPLIANCE_HOST}:443
```

## Disabling HTTP access

If desired, HTTP access may be disabled, which will cause any requests to the appliance using HTTP to fail. To do this, make a POST request to the `/v1/security/insecureports` endpoint, with a JSON body containing

```
{"enable_insecure_ports": false} :
```

```
curl -X POST "https://${APPLIANCE_HOST}/v1/security/insecureports" \
  -H "Content-Type: application/json" \
  -d '{"enable_insecure_ports": false}'
```

If the request succeeded then you should receive an HTTP 200 response. The web server in the appliance will take around five seconds to restart. Now, when attempting to make an HTTP request to the appliance you should see that no response is returned:

```
curl -X GET "http://${APPLIANCE_HOST}:8080/v1/management/services"

curl: (52) Empty reply from server
```

## Enable Basic Authentication for Admin

An admin password can be set to enable [HTTP basic authentication](#) for an `admin` user. Note that **authentication is only enforced when using HTTPS**. If you set an admin password then you **must** also disable HTTP access as described in the previous section. If you do not do this then it will be possible for someone else to override the admin password by making an unauthorized HTTP request.

To set a password, make a POST request to the `/v1/security/adminpassword` endpoint. The username for basic auth is always `admin`.

```
curl -X POST "https://${APPLIANCE_HOST}/v1/security/adminpassword" \
  -H "Content-Type: application/json" \
  -d '{"password": "example"}'

{"success":true,"message":"nginx_restart"}
```

If this request was successful then you should receive an HTTP 200 response with a success message. The web server in the appliance will take around five seconds to restart. All requests to HTTPS endpoints will now require a valid `Authorization` header as specified by [RFC7617](#). Unauthenticated requests will fail, for example:

```
$ curl -X GET "https://${APPLIANCE_HOST}/v1/management/services"
<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.17.6</center>
</body>
</html>
```

Authenticated requests should succeed. If you are using `curl` then the `--user` flag can be used to set the username and password (separated with a colon). If you're using the Management UI in a browser then a prompt will appear for a username and password.

```
$ curl --insecure -X GET --user "admin:example"
"https://${APPLIANCE_HOST}/v1/management/services"
```

If you have disabled HTTP access then it should now be impossible to make requests to the appliance without knowing the admin password. Please be aware that plain HTTP access does **not** require the admin password, and should be disabled if you are using a password.

## FAQs

### How do I reset the SSL settings?

If you have made a mistake in your SSL configuration, it is possible to reset the appliance to its default settings. This will return it to using the self-signed certificate from Speechmatics, and will delete any configured admin password. If you have disabled HTTP access then you need to know the existing admin password in order to do this.

To do this, make a DELETE request to the `/v1/security/reset` endpoint:

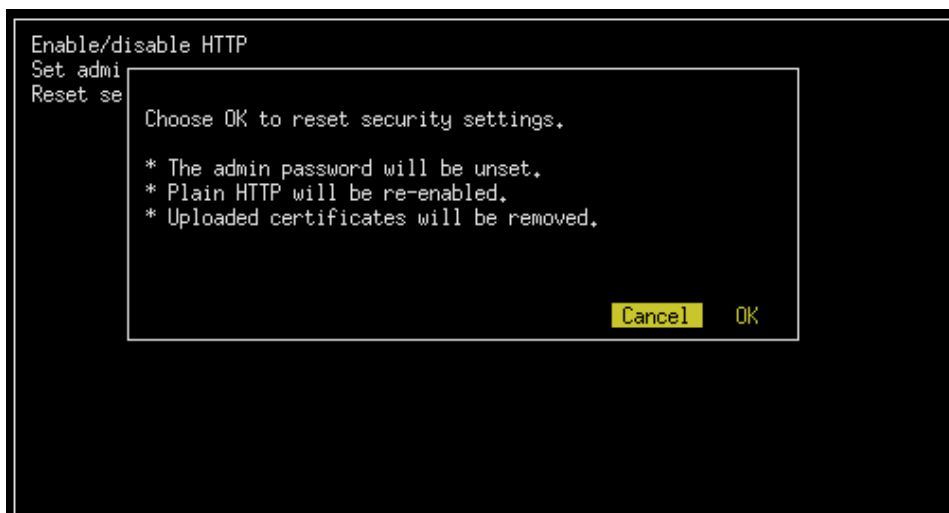
```
$ curl -X DELETE --user "admin:$PWD" "https://${APPLIANCE_HOST}/v1/security/reset"
{"success": true, "message": "nginx_restart"}
```

### What if I forget the admin password?

If you have forgotten the admin password you have set, and have disabled HTTP access to the appliance then it will not be possible to interact with the appliance over HTTP/HTTPS. Fortunately there is a way to reset the SSL configuration if you have direct access to the appliance's console (through the hypervisor that you use).

See the 'Administration -> Services -> Console for Advanced Troubleshooting' section for instructions on how to access the console.

Once you have opened the console open the 'Security' menu and select the 'Reset security' option to reset all security settings. It is also possible to toggle HTTP access and set the admin password using this interface.



### What versions of SSL/TLS do you support?

We support TLS 1.2 and TLS 1.3. We do not support earlier versions of TLS/SSL as these are considered weak. In general we would recommend you keep your client frameworks up to date with the latest security patches and try to use the strictest TLS configuration that you can.

### What cipher suites do you support?

For TLS 1.3 we support the following cipher suites that are considered strong (in server-preferred order):

- TLS\_AES\_256\_GCM\_SHA384
- TLS\_CHACHA20\_POLY1305\_SHA256
- TLS\_AES\_128\_GCM\_SHA256

For TLS 1.2 we support the following cipher suites that are considered strong (in server-preferred order):

- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_CHACHA20\_POLY1305\_SHA256

- TLS\_ECDHE\_RSA\_WITH\_ARIA\_256\_GCM\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_ARIA\_128\_GCM\_SHA256

Other cipher suites are available for TLS 1.2, but they are considered to be weak. Our recommendation is that you select one of the above cipher suites.

## Networking

### Network Requirements

When the virtual appliance is started for the first time it will automatically try to acquire an IP address using DHCP. If it is able to successfully acquire an address, it will be displayed on the VM console along with the fallback IP address: 10.10.10.2. However, if there is no DHCP server available on the network only the 10.10.10.2 IP address will be displayed.

The 10.10.10.2 address is a fallback address enabling communication with the virtual appliance when no DHCP services are available. This address should be used temporarily to set a static IP address if no DHCP is available. To do this, ensure that the client connecting to this address is on the same network by assigning it a suitable IP address (e.g. 10.10.10.3/24).

**Note:** The appliance uses three internal networks:

- docker\_gwbridge - 10.254.0.0/22
- ingress - 10.254.4.0/25
- docker0 - 10.254.4.128/25

You need to ensure that any network you use does not have an IP address conflict with anything in the range: 10.254.0.0 to 10.254.4.255.

### Configure Static IP

The virtual appliance can be configured to work on any IP network.

Setting a static IP requires three parameters: the IP address, subnet mask and default gateway. You set the static IP address like this:

```
curl -L -X POST "http://${APPLIANCE_HOST}:8080/v1/management/host/setipaddress" \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "address": "192.168.128.160",
    "netmask": "255.255.255.0",
    "gateway": "192.168.128.1"
  }' \
  | jq
```

**Note:** Once the POST is sent, the virtual appliance will automatically reboot. Check the console (or make an API call) to verify the new IP address has taken affect.

### Configure DHCP

You can also change back to using DHCP. Before undertaking this, ensure the network the virtual appliance is on has DHCP enabled.

```
curl -L -X POST "http://${APPLIANCE_HOST}:8080/v1/management/host/setdhcp" \
  -H 'Accept: application/json'
```

**NOTE:** once submitted, the virtual appliance will automatically reboot. Check the console to verify the new IP address has taken affect.

## Firewall Ports

There are several firewall rules that may need to be enabled to ensure the communication can be made to the virtual appliance:

- 8080/TCP - Used for the Management API to manage the virtual appliance
- 3000/TCP - Monitoring API
- 8082/TCP - Speech API for submitting jobs (Batch Appliance only)
- 9000/TCP - WebSockets Speech API for submitting jobs (Realtime Appliance only)
- 443/TCP - HTTPS access to the above APIs

## Using Proxies

If the network that you are deploying your appliance into does not have a direct route to the Internet, you may need to use a proxy server in order to talk to the cloud-based license service. See the relevant section in Licensing (below) for details on how to set this up.

# Virtual Appliance Scaling

## Real-time Virtual Appliance Scaling

This section explains how to scale the Real-time Virtual Appliance, and gives advice on how to make sure you've allocated enough resources for your workload.

### Worker Limits

The number of concurrent workers can be restricted using the Management API. This can be used to ensure that the system resources do not get exhausted by clients starting more sessions than expected. The maximum number of concurrent workers is set for the entire system, irrespective of which language packs are being used. The default number of maximum concurrent workers is 1.

### View Maximum Workers

Use a GET request to the `maxworkers` endpoint to view the maximum number of workers:

```
curl -L -X GET 'http://${APPLIANCE_HOST}:8080/v1/management/maxworkers' \
  -H 'Accept: application/json' \
  | jq
```

This shows the maximum number of workers that can run concurrently on the appliance. If more sessions are opened by clients using the Speech API then you will receive the job error: `No worker can be scheduled because the service is at capacity.`

### Setting Maximum Workers

Before changing the maximum number of concurrent workers for real-time transcription, it is important that the virtual appliance has enough system resources (CPU and RAM) to support the new requirement (see the Batch Virtual Appliance system requirements). This example shows how to set the maximum number of concurrent workers to 5:

```
curl -L -X POST 'http://${APPLIANCE_HOST}:8080/v1/management/maxworkers' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{ "count": "5" }'
```

As a rule of thumb, each concurrent worker will require 1 vCPU and up to 2GB RAM.

## Batch Virtual Appliance Scaling



This section explains how to scale the Batch Virtual Appliance, and gives advice on how to make sure you've allocated enough resources for your workload.

## Worker Limits

The number of concurrent workers (jobs) can be restricted using the Management API. This can be used to ensure that the system resources do not get exhausted by clients starting more transcriptions than expected. The maximum number of concurrent workers is set for the entire system, irrespective of which language packs are being used. The default number of maximum concurrent workers is 1.

## View Maximum Workers

Use a GET request to the maxworkers endpoint to view the maximum number of workers:

```
curl -L -X GET 'http://${APPLIANCE_HOST}:8080/v1/management/maxworkers' \
  -H 'Accept: application/json' \
  | jq
```

The response will indicate the maximum number of workers that can run concurrently on the appliance. If more jobs are submitted by clients using the Speech API then these will be queued up and processed once there is spare capacity on the appliance.

## Setting Maximum Workers

Before changing the maximum number of concurrent workers, it is important that the virtual appliance has enough system resources (CPU and RAM) to support the new requirement (see the Batch Virtual Appliance system requirements).

This example shows how to set the maximum number of concurrent workers to 5:

```
curl -L -X POST 'http://${APPLIANCE_HOST}:8080/v1/management/maxworkers' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{ "count": "5" }'
```

As a rule of thumb, each concurrent worker will require 1 vCPU and up to 5GB of RAM (depending on the quality of the audio).

If the number of jobs submitted exceeds the maximum number of concurrent workers then jobs will start to be queued, and the real-time factor (RTF) will increase, meaning you will wait longer for your transcripts to be made available.

# Monitoring

Appliance resources can be monitored at a system-wide level. Exhaustion of any of the resources can have a negative impact on the speed of the transcription.

The following resources that can be monitored:

Resource ID (rID)	Description
cpu	Provides the CPU usage across all the vCPU assigned
mem	Provides the total RAM usage of the appliance

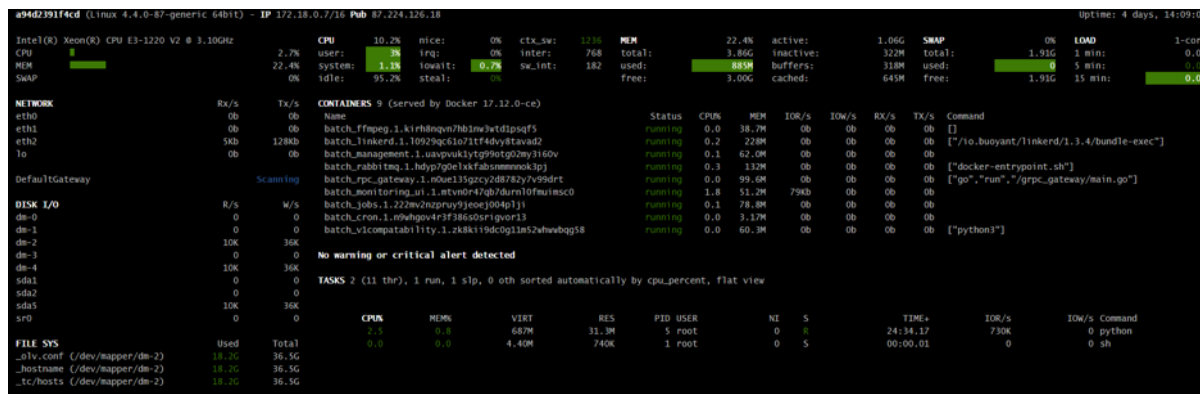
Here is an example GET request for the `mem` (RAM) resource:

```
curl -L -X GET "http://${APPLIANCE_HOST}:8080/v1/management/resource/mem" \
  -H 'Accept: application/json' \
  | jq
```

Here is an example response:

```
{
  "rId": "mem",
  "percentage": 13.4
}
```

For advanced monitoring, a utility called [Glances](#) is available that runs on TCP port 3000. It allows real-time resource stats to be monitored on the Batch Virtual Appliance. The easiest way to access this is via a web browser using the link `http://${APPLIANCE_HOST}:3000/` in the address bar.



It is also possible to access the Glances API using XML-RPC or HTTP REST (for JSON output), for example:

```
curl -L -X GET "http://${APPLIANCE_HOST}:3000/api/3/mem/percent" \
  -H 'Accept: application/json' \
  | jq
```

For more information on the HTTP REST interface, consult the [Glances documentation](#).

## Services

The virtual appliance has internal services that are required for operation.

There are system-wide services, and services specific to transcription workers for a given language.

### Batch Virtual Appliance

For the Batch Virtual Appliance, this table lists the services:

Service Name (Begins with)	Description	Required Status
batch_bja...	V2 REST API	Running.
batch_rpc_gateway...	RPC endpoint	Running
batch_license...	Licensing service	Running
batch_linkerd...	Internal Networking	Running
batch_management...	Management functions	Running
batch_ba_worker...	Job Queue management	Running
batch_monitoring_ui...	Monitoring Web GUI	Running
batch_batch-cron...	Completed job clean-up	Running

batch_v1compatibility...	V1 REST API	Running
jobs...	Used to perform ASR and transcription	Running
batch_swaggerui...	Swagger UI for certain APIs	Running
batch_nginxlb...	HTTP gateway	Running
batch_postgres...	Jobs Database	Running

The service will always have a current state, these states include:

Service Status	Description
running	Service has started and is running
created	Service is in the process of starting
exited	Service has stopped and is no longer running

## Service status

This can be used to ensure all services have the required status to operate (see table above). Example: GET to list services and corresponding status:

```
curl -L -X GET 'http://${APPLIANCE_HOST}:8080/v1/management/services' \
-H 'Accept: application/json' \
| jq
```

If the appliance has been licensed then you will see a return like this (for the Batch Virtual Appliance):

```
{
  "service_status": [
    {
      "service": "job-50",
      "status": "running"
    },
    {
      "service": "batch_bja.1.qegys910pamsduryf9tujm2db",
      "status": "running"
    },
    {
      "service": "batch_swaggerui.1.01imj506dokksku4mvy00gt70",
      "status": "running"
    },
    {
      "service": "batch_rpc_gateway.1.10aoi8f9cvkcko8s5jhrio8b6",
      "status": "running"
    },
    {
      "service": "batch_batch-cron.1.uahr5xz4edjx11fm06bflhthx",
      "status": "running"
    },
    {
      "service": "batch_v1compatibility.1.5t9hbwk30zqt2cnx5xzjf9zkt",
      "status": "running"
    },
    {
      "service": "batch_nginxlb.1.p2mq6ho4k5hho180zkog2maej",
      "status": "running"
    }
  ]
}
```

```

    },
    {
      "service": "batch_license.1.urx4qlzru7430lhv9669h9xxy",
      "status": "running"
    },
    {
      "service": "batch_management.1.5r92dvzwu0021g7mc9pb7qtg0",
      "status": "running"
    },
    {
      "service": "batch_postgres.1.yvef8y8gtq8nt62bc6ow987z",
      "status": "running"
    },
    {
      "service": "batch_monitoring_ui.1.m29c6ne7621y6dapq5fjojxj3",
      "status": "running"
    },
    {
      "service": "batch_linkerd.1.30ng6rrqiar7fqgkb9tesn9uw",
      "status": "running"
    },
    {
      "service": "batch_ba_worker.1.yliwg0uynenv2jcno9x423brc",
      "status": "running"
    }
  ]
}

```

## Real-time Virtual Appliance

For the Real-time Virtual Appliance, this table lists the services:

Service Name (Begins with)	Description	Required Status
rt_rt-server...	Load-balancing handling job requests	Running
rt_linkerd...	Proxy	Running
rt_management...	MGMT API Calls	Running
appliance_autoscaler...	required only during OVA build	Exited
rt_redis...	Handles worker availability	Running
rt_rpc_gateway...	Internal service management	Running
rt_monitoring_ui...	Monitoring Web GUI	Running
rt_nginx...	Proxying requests	Running
rt_rt-janitor...	Completed job clean-up	Running
rt_license...	Licensing	Running
rt_autoscaler...	Used to perform ASR and transcription	Running

The service will always have a current state, these states include:

Service Status	Description
running	Service has started and is running

created	Service is in the process of starting
exited	Service has stopped and is no longer running

## Service status

```
curl -L -X GET 'http://${APPLIANCE_HOST}:8080/v1/management/services' \  
-H 'Accept: application/json' \  
| jq
```

This can be used to ensure all services have the required status. If successful you will see the following response

```
{  
  "service_status": [  
    {  
      "service": "rt_rt-server.1.jgwwfsybbxmdq8205dqdz2r4",  
      "status": "running"  
    },  
    {  
      "service": "rt_linkerd.1.tetkum9u3iowqn2w71ok2nfp",  
      "status": "running"  
    },  
    {  
      "service": "rt_management.1.wk2kse9inpaie5nnby57zgjc",  
      "status": "running"  
    },  
    {  
      "service": "appliance_autoscaler-bootstrap-task_run_f92039b26280",  
      "status": "exited"  
    },  
    {  
      "service": "rt_redis.1.osd52r5esip3cvpsa3bsyfa3o",  
      "status": "running"  
    },  
    {  
      "service": "rt_rpc_gateway.1.mhb1yk8i50qxqs50jmu573u2o",  
      "status": "running"  
    },  
    {  
      "service": "rt_monitoring_ui.1.qzir2168b01zroej5khlgac0x",  
      "status": "running"  
    },  
    {  
      "service": "rt_nginxlb.1.z9uwrh458ttct6mg2iilcp427",  
      "status": "running"  
    },  
    {  
      "service": "rt_rt-janitor.1.leqrp4vre3eqg213uceye41zm",  
      "status": "running"  
    },  
    {  
      "service": "rt_license.1.jeop3k5hscque3vw9qo24jmtu",  
      "status": "running"  
    },  
    {  
      "service": "rt_autoscaler.1.jbpngclrokzf7zs7i7r97uxij",  
      "status": "running"  
    }  
  ]  
}
```

```
}
]
}
```

## Service restart

**Note:** After a service is restarted it will have a random string identifier post fixed to its name.

If required for troubleshooting you may need to restart all the services. During the restart, all transcription will stop. The following command performs a service restart:

```
$ curl -X DELETE 'http://<APPLIANCE_HOST>:8080/v1/management/services' \
-H 'Accept: application/json'
```

## Access Logs

The individual services on the system provide log files that can be collected to help with troubleshooting. The service name will need to be provided when retrieving logs. See above for instructions on how to view the names of the running services

The following parameters are available when accessing logs:

Name	Description	Required Status
name	Name of the service to collect the logs for	Required
count	Number of log lines wanted, defaults to 100; if all lines are to be returned set to -1	Optional

Example: GET to retrieve logs for batch\_monitoring\_ui service:

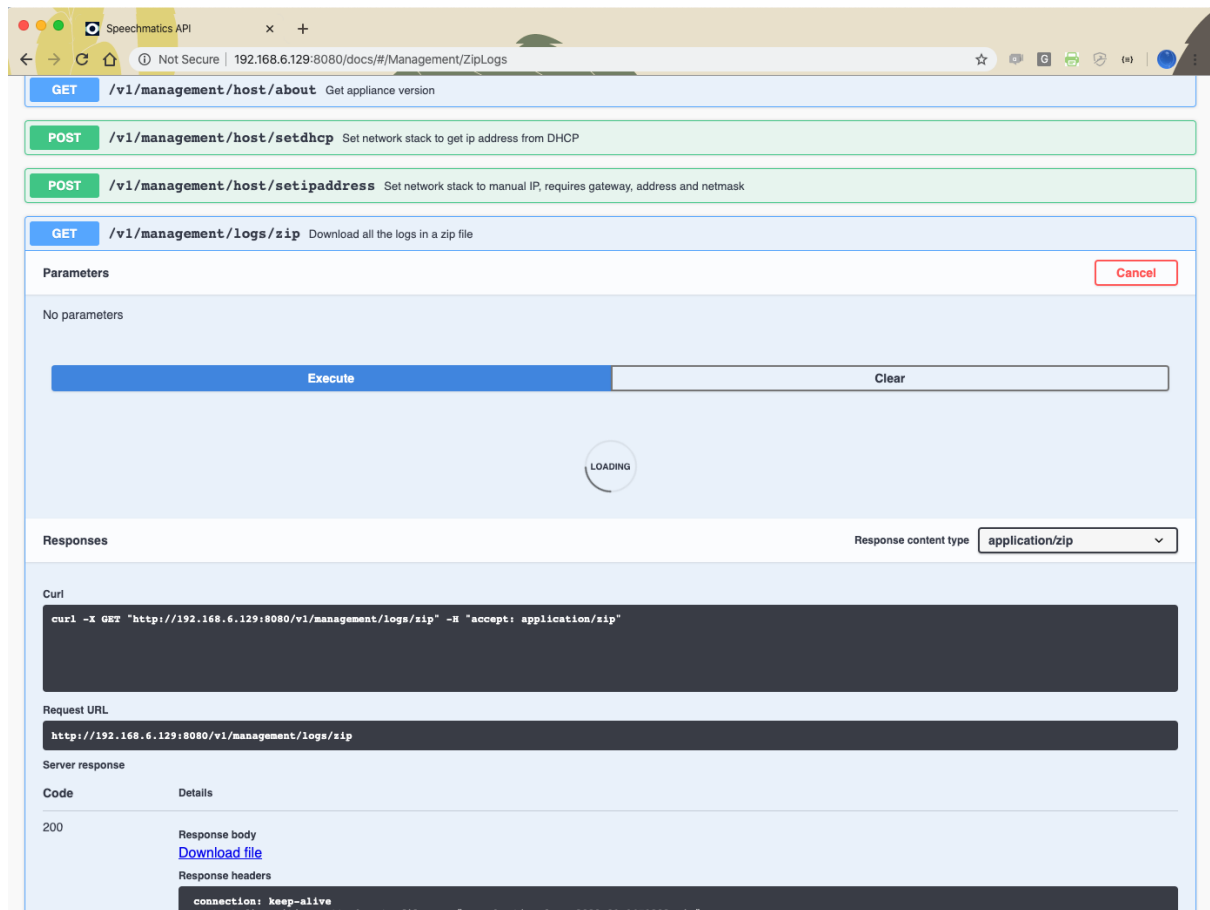
```
curl -L -X GET
'http://${APPLIANCE_HOST}:8080/v1/management/logs/batch_monitoring_ui.1.mtvn0r47qb7durnl0fmuimsc0'
\
-H 'Accept: application/json' \
| jq -r '.log_lines'
```

If you want to download *all* the logs (in order to provide information for a support ticket for instance) as a ZIP file, then it is possible to do this using the following command:

```
curl -L -X GET 'http://${APPLIANCE_HOST}:8080/v1/management/logs/zip' \
-H 'Accept: application/json' \
-o ./speechmatics.zip
```

It is also possible to do this directly from the Swagger UI by entering in the following URL to your browser:

[http://\\${APPLIANCE\\_HOST}:8080/docs/#/Management/ZipLogs](http://${APPLIANCE_HOST}:8080/docs/#/Management/ZipLogs), and then clicking on the download link when the ZIP file is ready.



## System restart

If the virtual appliance becomes unresponsive, there might be a need to restart it. If this is the case, it's recommended that the system is restarted through the management API, like this:

```
curl -L -X DELETE 'http://${APPLIANCE_HOST}:8080/v1/management/reboot'
```

If the Management API is not available, then you should reboot the appliance from the hypervisor console. For further information on how to restart the virtual machine via the console, please follow the manufacturers advice.

## System shutdown

You may wish to shut down the appliance. If so, it's recommended that the system is shut down through the management API, like this:

```
curl -L -X DELETE 'http://${APPLIANCE_HOST}:8080/v1/management/shutdown'
```

If the Management API is not available, then you should shut down the appliance from the hypervisor console. For further information on how to shut down the virtual machine via the console, please follow the manufacturers advice.

## Troubleshooting

There may be times unexpected behavior is observed with the Batch Virtual Appliance. If this is the case the following should be performed/checked:

- Check the license is valid (see licensing)
- Check the worker services are running

- Check the resources (CPU, memory & disk) to ensure they are not exhausted
- Restart all the services
- Restart the virtual appliance
- Collect logs and contact Speechmatics support: [support@speechmatics.com](mailto:support@speechmatics.com).

### Transcription job failure

If your transcription job fails with an `error` job status, more information can be found by looking at the logs from the `jobs` container (using the Management API, as previously described). Search the logs for the job id corresponding with your failure. If you see a `SoftTimeLimitExceeded` exception, this indicates that the job took longer than anticipated and as such was terminated. This is typically caused by poor VM performance, in particular slow disk IO operations (IOPS). If issues persist it may be necessary to improve the disk IO performance on the underlying host, or you may need to increase the RAM available to the VM such that memory caches can be taken advantage of. Please consult the section above on Host requirements, and the optimization advice specific to your hypervisor to ensure that you are not over-committing your compute resources.

### Illegal instruction errors

If jobs fail repeatedly and you see `Illegal instruction` errors in the log information for these jobs then it is likely that the host hardware you are running on does not support AVX. The host machine requirements for the Batch Virtual Appliance must meet the following minimum specification: Intel® Xeon® CPU E5-2630 v4 (Sandy Bridge) 2.20GHz (or equivalent). This is important because these chipsets (and later ones) support Advanced Vector Extensions (AVX). The machine learning algorithms used by Speechmatics ASR require the performance optimizations that AVX provides.

You can check this by looking in the management log when the appliance starts up. If you see a message like this:

```
2019-03-26 16:53:07,136    sm_management.app    ERROR    Processor not AVX capable. Tensorflow
language models cannot run.
```

Then it means that your host's CPU does not support AVX, or that your hypervisor does not have AVX support.

A console is available to help with advanced troubleshooting in the event that the Management API is unavailable. It is described in the next section.

### AVX2 Warning

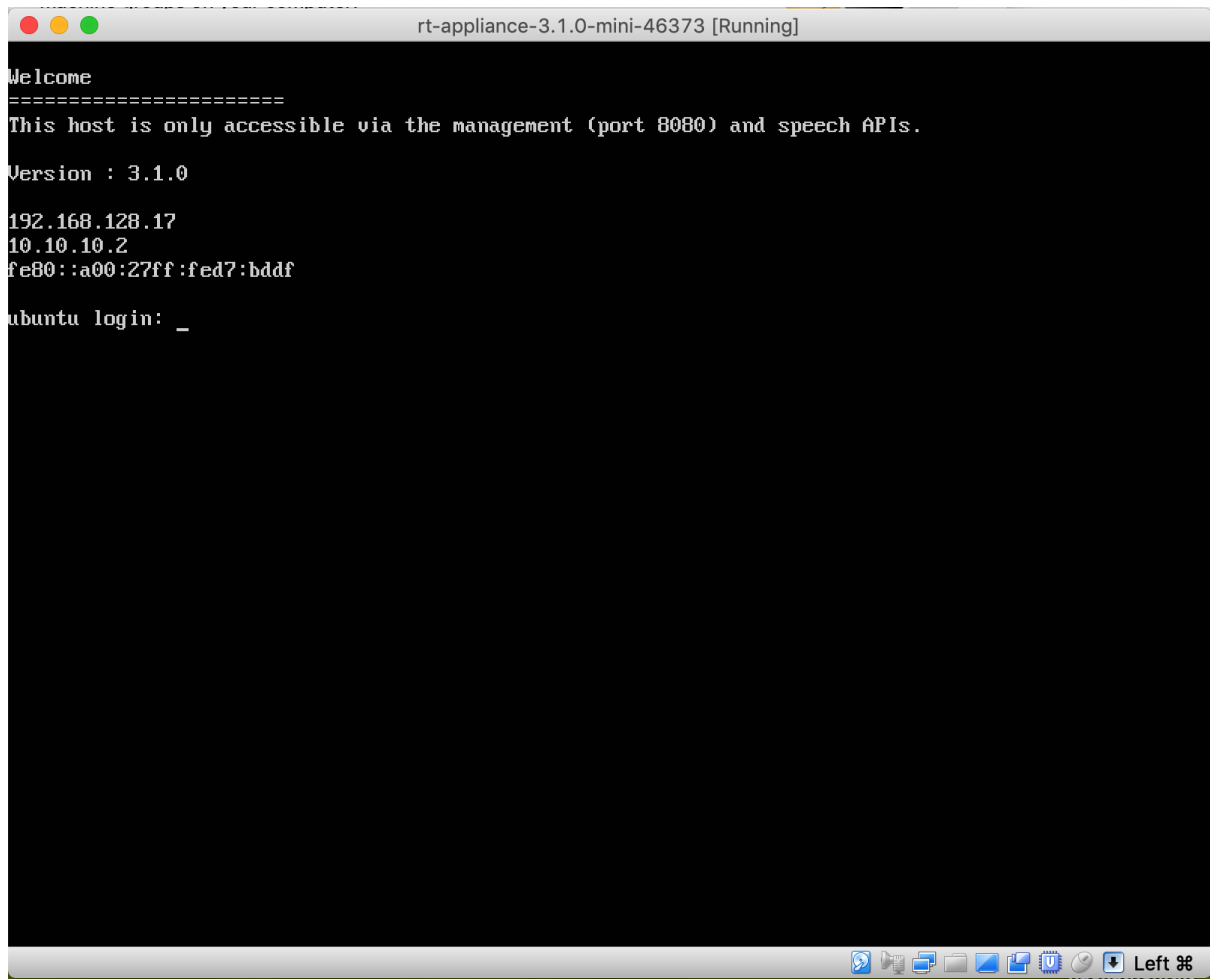
Speechmatics Appliance is optimised for running on hardware that supports the AVX2 flag. If you see the below message, your hardware is not optimised, and you may see slower performance of jobs

```
WARNING ([5.5.675~1-0c22]:SetupMathLibrary():asengine/asengine.cc:356) Unable to set CNR mode
to 10 (AVX2); falling back to 9. The transcription might be slower and/or use more CPU resource.
```

## Console for Advanced Troubleshooting

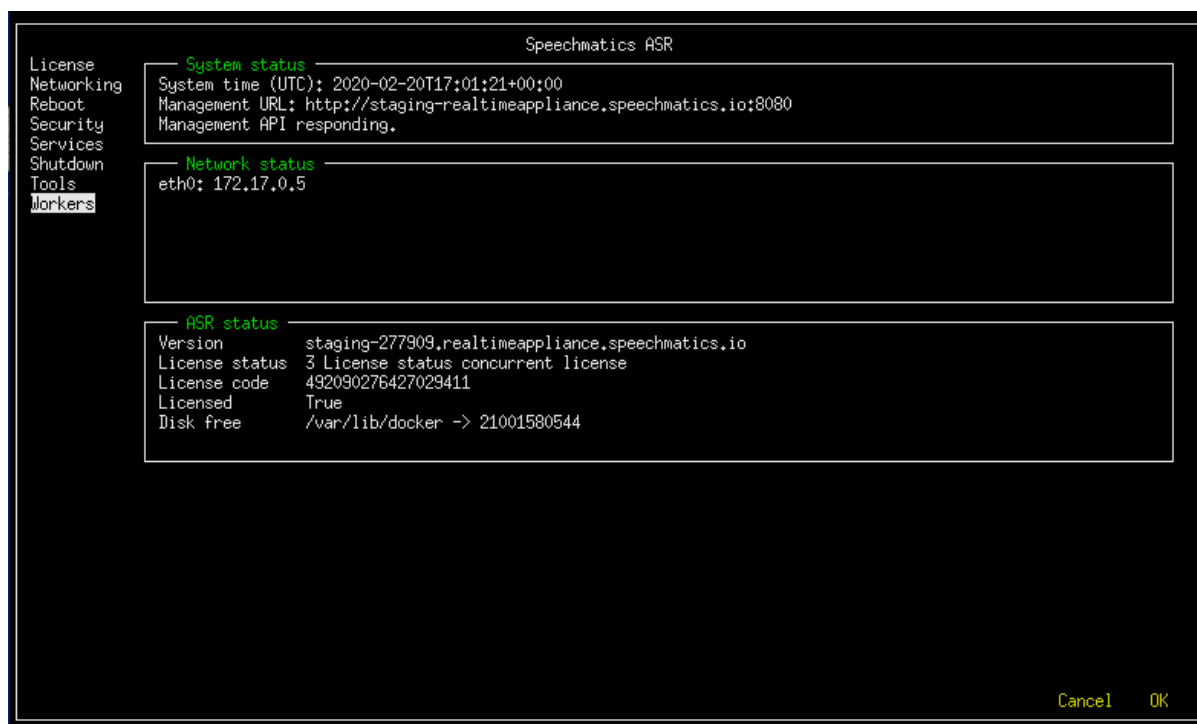
In the event that the Management API is unavailable (it is unresponsive, or there is no network connectivity) you can use the console to restore network connectivity, restart the appliance, or view information about services. To use this you need to use your hypervisor's GUI to access the logon screen for the appliance.





From this screen use the CTRL+ALT+F5 key combination to get to the console. Once you are in the console you have the following menu options available:

- License
- Networking
- Reboot
- Services
- Shutdown
- Tools
- Workers



The home screen shows high-level information about the appliance: IP addressing, software version and license status.

In the **System status** panel the **API responding** indicator shows the state of the Management API. **Network status** shows the IP address the appliance is currently configured with, and **ASR status** shows the license state and available storage space on the appliance.

In the event that you need to provide information to Speechmatics support you may be asked to connect to the console and provide this information. This section provides some tips on how to use the console to perform basic troubleshooting yourself.

**Note:** We recommend that you use the Management API for most troubleshooting tasks as it is easier to use. The console can be used in the event that the Management API is unavailable, but it does not provide all the features of the Management API.

## License

The [Licensing Troubleshooting](#) section provides detailed instructions on how to use the Management API to resolve common licensing issues. If you cannot use the Management API then you can still use console to check the license status and perform basic licensing steps.

## Networking

You can use the networking option to configure a static IP address, or use DHCP.

## Reboot and Shutdown

Reboot and Shutdown options exist to allow you to restart or shutdown the appliance from the console. You will be asked to select OK to confirm.

## Security

From this menu you can manage the security settings on the appliance, such as disabling HTTP access, changing the admin password for HTTP basic authentication, and resetting the SSL configuration.

## Services

From this menu you can access the list of services that are running on the appliance. Selecting a service shows the log entries for that service.

## Tools

This menu allows you to access a number of useful Unix utilities that can be used for advanced troubleshooting. In order to help progress a support ticket you may be asked to provide the output (ie. a screenshot) from running one of these commands.

## Workers

This allows you to view and change the maximum number of workers allowed to run concurrently.

# Security

The appliance is designed to be installed within your own security perimeter. It has its own firewall installed to only allow ingress to ports that are required for its management, monitoring and Speech APIs.

## Overview

The appliance uses a microservices architecture running on a customized Ubuntu machine. [AppArmor](#) default security policies are used to protect the OS and running applications on the appliance.

Data on the appliance (including audio and video data that is submitted via the Speech API, logs, and output transcripts) are encrypted on disk.

## Ports and Protocols

There are several firewall rules that may need to be enabled to ensure the communication can be made to the virtual appliance. If you setup HTTPS as described in the 'SSL Configuration' section of these docs then you only need to expose port 443.

Port/Protocol	Description
8080/TCP	Used for the Management API to manage the virtual appliance
3000/TCP	Monitoring (Glances)
8082/TCP	REST Speech API for batch ASR
9000/TCP	Websocket Speech API for real-time ASR
443/TCP	Used for HTTPS communication with all of the above services

# Batch Virtual Appliance

:::warning Removal Note The legacy V1 API that the Batch Virtual Appliance currently supports is **now removed**. The V2 API is the only supported API in the Batch Virtual Appliance following this release. How to use the V2 API can be seen in the docs (<https://docs.speechmatics.com/en/batch-appliance/api-guide/api-howto/>). :::

## Overview

The Speechmatics Batch Virtual Appliance exposes a REST Speech API to enable communication between a client application and the appliance over a HTTP or HTTPS connection. This provides the ability to convert a media file into a text transcript, providing words, speaker, and timing information.

## Terms

For the purposes of this guide the following terms are used.

Term	Description
Client	An application connecting to the Batch Virtual Appliance using the Transcription API. The client will provide audio containing speech, and process the transcripts received as a result.
Management API	The REST API that allows administrators to manage the virtual appliance over port 8080 (or 443 for secure access). To access the documentation you can use the following endpoints: <code>http://\${APPLIANCE_HOST}:8080/docs/</code> OR <code>https://\${APPLIANCE_HOST}/docs/</code> , where <code>\${APPLIANCE_HOST}</code> is the IP address or hostname of your appliance.
Speech V2 API	The REST API that allows users of the appliance to submit ASR jobs over port 8082 (or 443 for secure access). The endpoints <code>https://\${APPLIANCE_HOST}/v2/jobs/</code> OR <code>http://\${APPLIANCE_HOST}:8082/v2/jobs/</code> can be used.
Batch Virtual Appliance	The appliance (VM) that provides ASR transcription capability.

## Getting Started

In order to use the REST Speech API you need access to a Batch Virtual Appliance. See the Speechmatics Virtual Appliance Installation and Admin Guide on how to install, configure, and license the appliance.

You do not need user credentials (such as an authorization token) to use the Speech API with the Batch Virtual Appliance.

## Audio Formats

A variety of audio formats for input are supported; there is no need to specify the audio format when it is submitted for transcription; the Batch Virtual Appliance automatically detects the format and handles it using the correct decoder. The current audio formats are supported:

- aac
- amr
- flac
- m4a
- mp3
- mp4
- mpeg
- ogg
- wav

**Note:** the native formats are 16KHz or 8KHz (PCM32 LE) WAV; for the best results and performance we recommend that you submit files in that format.

## Accessing the API

### V2 API

The V2 API is the primary way via which all customers should submit media and retrieve transcripts on the Batch Virtual Appliance.

- HTTP and HTTPS are supported. We recommend using HTTPS wherever possible. How to set up SSL configuration is documented in the Installation Guide
- The port used for HTTP connection is port 8082 only
- All V2 features are supported using this API version

### File Size Limits

The maximum file size supported is 4GB, or up to 2 hours in length. Anything larger must be chunked into smaller sections in order to be successfully transcribed.

## Transcription Formats

In the V2 API, three output formats are available: `json-v2` (the default), `txt`, and `srt`. The current version of this output is 2.7. If the output format is set to `txt`, the file is returned in plain text rather than JSON format. If the output format is set to `srt`, the file is returned in the SubRip subtitle format instead.

In the V1 API, four output formats for transcription are available: `json` (the default), `json-v2`, `txt`, and `srt`. If you want JSON output it is recommended to use `json-v2`.

## Troubleshooting

If you have problems making a call, ensure that you are using exactly the same URI format as shown in this document. For instance, not including the trailing '/' character on the URIs will cause a 302 redirect response to be sent – if your client does not handle redirects then this may cause problems.

## Tools

Code samples in this guide expect you to use [curl](#) for making HTTP requests to the Management API, and the [jq](#) tool to parse and display JSON responses.

The easiest way to access the APIs and online help is via the following URL on the appliance:

```
http://${APPLIANCE_HOST}:8080/help/
```

This page allows you to access the documentation from the browser as well as providing links to the APIs.

## Windows

On a Windows PC you can use these download and installation links to get these tools:

```
https://curl.haxx.se/download.html  
https://stedolan.github.io/jq/download/
```

## Linux

Use the relevant package manager for your flavor of Linux, which will either be:

```
$ apt install curl jq
```

or

```
$ yum install curl jq
```

## Mac OS X

On the Mac, the easiest way to install these utilities is using [Homebrew](#):

```
$ brew install curl jq
```

## Language Pack Codes

Language	ISO Code
Arabic	ar
Bulgarian	bg
Catalan	ca
Mandarin	cmn
Czech	cs

Danish	da
German	de
Greek	el
Global English	en
Global Spanish	es
Finnish	fi
French	fr
Hindi	hi
Croatian	hr
Hungarian	hu
Indonesian	id
Italian	it
Japanese	ja
Korean	ko
Lithuanian	lt
Latvian	lv
Malay	ms
Dutch	nl
Norwegian	no
Polish	pl
Portuguese	pt
Romanian	ro
Russian	ru
Slovakian	sk
Slovenian	sl
Swedish	sv
Turkish	tr
Cantonese	yue

## How To Use the V2 API

This section will take you through how to send a file to the V2 Speech API in the Batch Virtual Appliance and receive a finished transcript. It will also show you how to configure the transcription to use supported speech features.

### Quick Start

This quick start guide will show you how to submit a media file for processing and then retrieve a transcript in the format of your choice via the V2 API, the recommended method of using the Batch Virtual Appliance. It will also show you optionally how to check the status of a job and to delete it once it has completed.

## Pre Requisites

- You have successfully imported, installed, and licensed the appliance of your choice as shown in the Installation Guide

## Examples

All examples in this document use `curl` to make the REST API call from a command line. We recommend using retry parameters, so that retry attempts can be made for at least one minute. With the curl command this is done with the `--retry 5 --retry-delay 10` parameters. This has been omitted from the examples in this document for brevity.

**Note:** If you are using a self-signed certificate (your own, or the Speechmatics certificate that is used by default), then you will see a warning like this when using the curl command to access the Speech API using HTTPS:

```
curl: (60) SSL certificate problem: self signed certificate
```

We recommend, if you are going to use the secure Speech API, that you upload your own SSL certificate (signed by a CA) to the appliance, to avoid this problem. See the Installation and Admin Guide for details of how to do this.

## Submitting a Job

To successfully submit a job you must send a HTTP POST request to your chosen endpoint with:

- The request `type`. This is always `transcription`
- The language you want your transcript in. This is submitted within configuration object as part of the transcription config as a two-letter ISO 639-1 code, and is mandatory
  - For more details about the configuration object, please see sections below in the section on Configuring the Transcript
- A media file in a supported format, or a URL address of a file location the appliance is authorised to fetch

An example is below for a transcript request in English:

```
curl -X POST 'https://${APPLIANCE_HOST}/v2/jobs/' \
-F data_file=@example.wav \
-F config='{
  "type": "transcription",
  "transcription_config": { "language": "en" }
}' \
```

If you are successful, you will receive a HTTP 201 request and a Job ID. A Job ID is a unique sequential numeric string. You will need this job ID to retrieve any transcript generated.

## Requesting an enhanced model

Speechmatics supports two different models within each language pack; a standard or an enhanced model. The standard model is the faster of the two, whilst the enhanced model provides a higher accuracy, but a slower turnaround time.

The enhanced model is a premium model. Please contact your account manager or Speechmatics if you would like access to this feature.

An example of requesting the enhanced model is below

```
{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "operating_point": "enhanced"
  }
}
```

Please note: `standard`, as well as being the default option, can also be explicitly requested with the `operating_point` parameter.

## Checking on a Job Status

If you want to see the progress of an individual job you can make a GET request. You must include the Job ID you want to check in the GET request.

To retrieve a job run the following request:

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/${JOBID}'
```

Here is an example of a successful response for a completed job:

```
{
  "jobs": [
    {
      "config": {
        "transcription_config": {
          "language": "en"
        },
        "type": "transcription"
      },
      "created_at": "2020-12-08T09:49:39.907Z",
      "data_name": "Can robots care for us_.mp3",
      "duration": 379,
      "id": "1",
      "status": "done"
    }
  ]
}
```

In the response you will receive:

- The configuration information used to submit that job
- The time the job was created
- The duration of the audio file measured in seconds
- The status of the job. If it is finished, the job status should return `done`. If the job is still being processed it will return `running`.
- The ID of the job you requested

## Checking the status of multiple submitted jobs

If you wish you can retrieve all jobs submitted to the appliance within the last 24 hours by not including the job ID in the GET request. An example is below

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/'
```

If successful you will receive a 200 response and all available jobs:

```
{
  "jobs": [
    {
      "created_at": "2021-01-08T11:58:04.124Z",
      "data_name": "IsTheRecyclingSystemBroken.mp3",
      "duration": 377,
      "id": "2",
      "status": "running"
    },
  ]
}
```



```

    {
      "created_at": "2021-01-08T11:57:48.945Z",
      "data_name": "Can robots care for us_.mp3",
      "duration": 379,
      "id": "1",
      "status": "running"
    }
  ]
}

```

Please note if you request to see all jobs, you will not see the configuration for each job. Configuration information can only be retrieved for a request for an individual job. If you have changed the clean up job on the appliance to run at more frequent intervals than the default 24 hours you will only see jobs posted after that clean-up job ran.

You can now retrieve a transcript from the appliance.

## Retrieving a Transcript

Here is an example request to retrieve a transcript from a completed job:

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/$JOBID/transcript'
```

You must put the job ID within the URL path that you received upon successfully requesting the transcription job.

If you request a transcript before it has finished processing, you will receive a HTTP 404 message. To avoid this, you can configure notifications so that you can retrieve transcripts via callback when completed. For details of setting up notifications, please see the section on 'Configuring the Job Request'.

The default format for any transcript is `json-v2`. Speechmatics also supports transcripts in plain text (TXT) and SubRip Title (SRT) formats. To do so you must explicitly request these.

An example of a successful retrieval of a transcript in plain TXT format:

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/$JOBID/transcript?format=txt'
```

Here is an example of a successful request of a transcript in SRT format:

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/$JOBID/transcript?format=srt'
```

You can receive transcripts in multiple output formats simultaneously via notifications requested in the initial POST submission.

You should now have been able to submit a file and retrieve a transcript.

## Deleting a Job

In addition, you can delete a transcript using a HTTP DELETE request only once it has finished processing. The default retention period for a transcript on the Batch Virtual Appliance is **24 hours**. You can alter the configuration of the appliance to shorten this retention period via the Management API; how to do so is documented in the installation guide

You must include in the request the Job ID you wish to delete

```
curl -X DELETE 'https://${APPLIANCE_HOST}/v2/jobs/$JOBID/'
```

If you have successfully deleted the transcript, you will receive a HTTP 200 response, and a summary of the job you have just deleted. An example is below

```

{
  "job": {
    "config": {
      "transcription_config": {

```

```

        "language": "en"
      },
      "type": "transcription"
    },
    "created_at": "2020-12-10T15:38:33.866Z",
    "data_name": "Can robots care for us_.mp3",
    "duration": 379,
    "id": "5",
    "status": "deleted"
  }
}

```

You cannot delete multiple jobs at once.

## Cancelling a Job

Via the V2 API, you are now able to delete a running job. In this case, no transcript will be returned, and any seconds deducted for processing the transcript will be returned to the license.

To cancel a running job, use the query parameter `force=true` when sending a DELETE request. An example is below

```
curl -X DELETE 'https://${APPLIANCE_HOST}/v2/jobs/${JOBID}?force=true'
```

The response will show the job, and a status of `deleted` : an example is below:

```

{
  "job": {
    "config": {
      "transcription_config": {
        "language": "en"
      },
      "type": "transcription"
    },
    "created_at": "2021-02-02T13:45:37.074Z",
    "data_name": "6MinuteEnglish-20200528-IsTheRecyclingSystemBroken.mp3",
    "duration": 378,
    "id": "9",
    "status": "deleted"
  }
}

```

By default the `force` flag is **false**. This means a DELETE request without the `force` or `force=false` flag for a running job will return HTTP 423 `Resource Locked` and the transcript will continue to be processed. If the job has already finished, the request will be handled as a normal DELETE request (e.g. the transcript will be deleted, but no time will be returned to the appliance license)

## Configuring the V2 Speech API Request

The following sections will show how to use the configuration object when submitting a request in order to use various Speechmatics features in the Batch Virtual Appliance. Where features are only supported in the V2 API this will be made explicit.

To configure any transcription request you must alter the relevant part of the configuration object:

- `fetch_data` config: If you want to fetch a file stored in an online location
- `transcription_config`: For any of the following features:
  - Diarization
  - Custom Dictionary/additional vocabulary
  - Output Locale

- o Advanced Punctuation
- notification\_config: For receiving any notifications from the appliance. You can receive updates as to the job's status, or the transcript once completed
- output\_config: For altering the presentation of transcripts, only valid in SRT format

## Fetch URL

The previous example showed how to create a job from a locally uploaded audio file. If you store your digital media in cloud storage (for example AWS S3 or Azure Blob Storage) you can also submit a job by providing the URL of the audio file. The configuration uses a `fetch_data` section, which looks like this:

```
curl -X POST 'https://${APPLIANCE_HOST}/v2/jobs' \
  -F config='{
    "type": "transcription",
    "transcription_config": { "language": "en" },
    "fetch_data": { "url": "https://s3.us-east-
2.amazonaws.com/bucketname/jqld_/20180804102000/profile.m4v" }
  }' \
```

**\*\* A note on best practice \*\***

If you are using pre-signed URLs, please ensure these have not expired before sending them to the appliance, as the job will fail.

If you need to additional authentication or authorization the appliance supports an optional `auth_headers` parameter where these can be supplied: e.g. when using an OAuth2 Bearer token.

Please note: when submitting many jobs at once, please note that the audio will be fetched after a job ID is returned and before a job can be processed by an ASR worker. Please ensure when submitting large numbers of jobs in a small space of time that there is sufficient space on the appliance for the number of files you wish to submit.

## Speaker Separation (Diarization)

Speechmatics offers four different modes for separating out different speakers in the audio:

Type	Description	Use Case
speaker diarization	Aggregates all audio channels into a single stream for processing and picks out unique speakers based on acoustic matching.	Used in cases where there are multiple speakers embedded in the same audio recording and it's required to understand what each unique speaker said.
channel diarization	Transcribes each audio channel separately and treats each channel as a unique speaker.	Used when it's possible to record each speaker on separate audio channels.
speaker change (beta)	Provides the point in transcription when there is believed to be a new speaker.	Used for when you just need to know the speaker has changed usually in a real-time application.
channel diarization & speaker change	Transcribes each audio channel separately and within each channel provides the point when there is believed to be a new speaker.	Used when it's possible to record some speakers on a separate audio channel, but some channels there are multiple speakers.

Each of these modes can be enabled by using the `diarization` config. The following are valid values:

The default value is `none` - e.g. the transcript will not be diarized.

Type	Config Value

speaker diarization	speaker
channel diarization	channel
speaker change	speaker_change
channel diarization & speaker change	channel_and_speaker_change

### Speaker Diarization

Speaker diarization aggregates all audio channels into a single stream for processing, and picks out different speakers based on acoustic matching.

By default the feature is disabled. To enable speaker diarization the following must be set when you are using the config object:

```
{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "diarization": "speaker"
  }
}
```

When enabled, every `word` and `punctuation` object in the output results will be a given "speaker" property which is a label indicating who said that word. There are two kinds of labels you will see:

- `S#` - S stands for speaker and the # will be an incrementing integer identifying an individual speaker. S1 will appear first in the results, followed by S2 and S3 etc.
- `UU` - Diarization is disabled or individual speakers cannot be identified. `UU` can appear for example if some background noise is transcribed as speech, but the diarization system does not recognise it as a speaker.

**Note:** Enabling diarization increases the amount of time taken to transcribe an audio file. In general we expect diarization to take roughly the same amount of time as transcription does, therefore expect the use of diarization to roughly double the overall processing time.

The example below shows relevant parts of a transcript with 3 speakers. The output shows the configuration information passed in the `config.json` object and relevant segments with the different speakers in the JSON output. Only part of the transcript is shown here to highlight how different speakers are displayed in the output.

```
"format": "2.7",
"metadata": {
  "created_at": "2020-07-01T13:26:48.467Z",
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "diarization": "speaker"
  }
},
"results": [
  {
    "alternatives": [
      {
        "confidence": 0.93,
        "content": "hello",
        "language": "en",
        "speaker": "S1"
      }
    ]
  },
  ]
```

```

    "end_time": 0.51,
    "start_time": 0.36,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 1.0,
        "content": "hi",
        "language": "en",
        "speaker": "S2"
      }
    ],
    "end_time": 12.6,
    "start_time": 12.27,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 1.0,
        "content": "good",
        "language": "en",
        "speaker": "S3"
      }
    ],
    "end_time": 80.63,
    "start_time": 80.48,
    "type": "word"
  }
}

```

In our JSON output, `start_time` identifies when a person starts speaking each utterance and `end_time` identifies when they finish speaking.

### Speaker diarization tuning

The sensitivity of the speaker detection is set to a sensible default that gives the optimum performance under most circumstances. However, you can change this value based on your specific requirements by using the `speaker_sensitivity` setting in the `speaker_diarization_config` section of the job config object, which takes a value between 0 and 1 (the default is 0.5). A higher sensitivity will increase the likelihood of more unique speakers returning. For example, if you see fewer speakers returned than expected, you can try increasing the sensitivity value, or if too many speakers are returned try reducing this value. It's not guaranteed to change since several factors can affect the number of speakers detected. Here's an example of how to set the value:

```

{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "diarization": "speaker",
    "speaker_diarization_config": {
      "speaker_sensitivity": 0.6
    }
  }
}

```

### Speaker diarization post-processing

To enhance the accuracy of our speaker diarization, we make small corrections to the speaker labels based on the punctuation in the transcript. For example if our system originally thought that 9 words in a sentence were spoken by speaker S1, and only 1 word by speaker S2, we will correct the incongruous S2 label to be S1. This only works if punctuation is enabled in the transcript.

Therefore if you disable punctuation, for example by removing all `permitted_marks` in the `punctuation_overrides` section of the `config.json` then expect the accuracy of speaker diarization to vary slightly.

## Speaker diarization timeout

Speaker diarization will timeout if it takes too long to run for a particular audio file. Currently the timeout is set to 5 minutes or  $0.5 * \text{the audio duration}$ ; whichever is longer. For example, with a 2 hour audio file the timeout is 1 hour. If a timeout happens the transcript will still be returned but without the speaker labels set.

If the diarization does timeout you will see an ERROR message in the logs that looks like this:

```
Speaker diarization took too long and timed out (X seconds).
```

If a timeout occurs then all speaker labels in the output will be labelled as UU.

Under normal operation we do not expect diarization to timeout, but diarization can be affected by a number of factors including audio quality and the number of speakers. If you do encounter timeouts frequently then please get in contact with Speechmatics support.

## Channel Diarization

Channel diarization allows individual channels in an audio file to be labelled. This is ideal for audio files with multiple channels (up to 6) where each channel is a unique speaker.

By default the feature is disabled. To enable channel diarization the following must be set when you are using the config object:

```
{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "diarization": "channel"
  }
}
```

The following illustrates an example configuration to enable channel diarization on a 2-channel file that will use labels `Customer` for channel 1 and `Agent` for channel 2:

```
{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "diarization": "channel",
    "channel_diarization_labels": ["Customer", "Agent"]
  }
}
```

For each named channel, the words will be listed in its own labelled block, for example:

```
{
  "format": "2.7",
  "metadata": {
    "created_at": "2020-07-01T14:11:43.534Z",
    "type": "transcription",

```

```

"transcription_config": {
  "language": "en",
  "diarization": "channel",
  "channel_diarization_labels": ["Customer", "Agent"]
}
},
"results": [
  {
    "alternatives": [
      {
        "confidence": 0.87,
        "content": "Hello",
        "language": "en"
      }
    ],
    "channel": "Customer",
    "end_time": 14.34,
    "start_time": 14.21,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.87,
        "content": "how",
        "language": "en"
      }
    ],
    "channel": "Agent",
    "end_time": 14.62,
    "start_time": 14.42,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.87,
        "content": "can",
        "language": "en"
      }
    ],
    "channel": "Agent",
    "end_time": 15.14,
    "start_time": 14.71,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.79,
        "content": "I",
        "language": "en"
      }
    ],
    "channel": "Agent",
    "end_time": 16.71,
    "start_time": 16.3,

```

```

    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.67,
        "content": "help",
        "language": "en"
      }
    ],
    "channel": "Agent",
    "end_time": 10.39,
    "start_time": 10.17,
    "type": "word"
  }
}

```

**Note:**

- Transcript output is provided sequentially **by channel**. So if you have two channels, all of channel 1 would be output first, followed by all of channel 2, and so on
- If you specify `channel` as a diarization option, and do not assign `channel_diarization_labels` then default labels will be used (`channel_1`, `channel_2` etc)
- Spaces cannot be used in the channel labels

## Speaker Change Detection (beta feature)

This feature allows changes in the speaker to be detected and then marked in the transcript. It does not provide information about whether the speaker is the same as one earlier in the audio.

By default the feature is disabled. The config used to request speaker change detection looks like this:

```

{
  "type": "transcription",
  "transcription_config": {
    "diarization": "speaker_change",
    "speaker_change_sensitivity": 0.8
  }
}

```

**Note:** Speaker change is only visible in the JSON V2 output, so make sure you use the `json-v2` format when you retrieve the transcript.

The `speaker_change_sensitivity` property, if used, must be a numeric value between 0 and 1. It indicates to the algorithm how sensitive to speaker change events you want to make it. A low value will mean that very few changes will be signalled (with higher possibility of false negatives), whilst a high value will mean you will see more changes in the output (with higher possibility of false positives). If this property is not specified, a default of 0.4 is used.

Speaker change elements appear in resulting JSON transcript `results` array look like this:

```

{
  "type": "speaker_change",
  "start_time": 0.55,
  "end_time": 0.55,
  "alternatives": []
}

```

**Note:** Although there is an `alternatives` property in the speaker change element it is always empty, and can be ignored. The `start_time` and `end_time` properties are always identical, and provide the time when the change was detected.



A speaker change indicates where we think a different person has started talking. For example, if one person says "Hello James" and the other responds with "Hi", there should be a `speaker_change` element between "James" and "Hi", for example:

```
{
  "format": "2.7",
  "job": {
    ....
    "results": [
      {
        "start_time": 0.1,
        "end_time": 0.22,
        "type": "word",
        "alternatives": [
          {
            "confidence": 0.71,
            "content": "Hello",
            "language": "en",
            "speaker": "UU"
          }
        ]
      },
      {
        "start_time": 0.22,
        "end_time": 0.55,
        "type": "word",
        "alternatives": [
          {
            "confidence": 0.71,
            "content": "James",
            "language": "en",
            "speaker": "UU"
          }
        ]
      },
      {
        "start_time": 0.55,
        "end_time": 0.55,
        "type": "speaker_change",
        "alternatives": []
      },
      {
        "start_time": 0.56,
        "end_time": 0.61,
        "type": "word",
        "alternatives": [
          {
            "confidence": 0.71,
            "content": "Hi",
            "language": "en",
            "speaker": "UU"
          }
        ]
      }
    ]
  }
}
```

- Note: You can only choose **speaker\_change** as an alternative to **speaker** or **channel** diarization.

## Speaker Change Detection With Channel Diarization

Speaker change can be combined with channel diarization. It will transcribe each channel separately and indicate in the output each channel (with labels if set) and the speaker changes on each of the channels. For example, if a two-channel audio contains three people greeting each other (with a single speaker on channel 1 and two speakers on channel 2), the config submitted with the audio to request the speaker change detection is:

```
{
  "type": "transcription",
  "transcription_config": {
    "diarization": "channel_and_speaker_change",
    "speaker_change_sensitivity": 0.8
  }
}
```

The output will have special elements in the `results` array between two words where a different person starts talking on the same channel.

```
{
  "format": "2.7",
  "job": {
    ....
  },
  "metadata": {
    ....
  },
  "results": [
    {
      "channel": "channel_2",
      "start_time": 0.1,
      "end_time": 0.22,
      "type": "word",
      "alternatives": [
        {
          "confidence": 0.71,
          "content": "Hello",
          "language": "en",
          "speaker": "UU"
        }
      ]
    },
    {
      "channel": "channel_2",
      "start_time": 0.22,
      "end_time": 0.55,
      "type": "word",
      "alternatives": [
        {
          "confidence": 0.71,
          "content": "James",
          "language": "en",
          "speaker": "UU"
        }
      ]
    },
    {
      "channel": "channel_1",
```

```

    "start_time": 0.55,
    "end_time": 0.55,
    "type": "speaker_change",
    "alternatives": []
  },
  {
    "channel": "channel_2",
    "start_time": 0.56,
    "end_time": 0.61,
    "type": "word",
    "alternatives": [
      {
        "confidence": 0.71,
        "content": "Hi",
        "language": "en",
        "speaker": "UU"
      }
    ]
  },
  {
    "channel": "channel_1",
    "start_time": 0.56,
    "end_time": 0.61,
    "type": "word",
    "alternatives": [
      {
        "confidence": 0.71,
        "content": "Hi",
        "language": "en",
        "speaker": "UU"
      }
    ]
  }
]
}

```

- Note: Do not try to request **speaker\_change** and **channel diarization** as multiple options: only **channel\_and\_speaker\_change** is an accepted parameter for this configuration.

## Custom dictionary

The Custom Dictionary feature allows a list of custom words to be added for each transcription job. This helps when a specific word is not recognised during transcription. It could be that it's not in the vocabulary for that language, for example a company or person's name. Adding custom words can improve the likelihood they will be output.

The `sounds_like` feature is an extension to this to allow alternative pronunciations to be specified to aid recognition when the pronunciation is not obvious.

The Custom Dictionary feature can be accessed through the `additional_vocab` property.

Prior to using this feature, consider the following:

- `sounds_like` is an optional setting recommended when the pronunciation is not obvious for the word or it can be pronounced in multiple ways; it is valid just to provide the `content` value
- `sounds_like` only works with the main script for that language
  - Japanese (ja) `sounds_like` only supports full width Hiragana or Katakana
- You can specify up to 1000 words or phrases (per job) in your custom dictionary

```

"transcription_config": {
  "language": "en",
  "additional_vocab": [
    {
      "content": "gnocchi",
      "sounds_like": [
        "nyohki",
        "nokey",
        "nochi"
      ]
    },
    {
      "content": "CEO",
      "sounds_like": [
        "C.E.O."
      ]
    },
    {
      "content": "financial crisis"
    }
  ]
}

```

In the above example, the words *gnocchi* and *CEO* have pronunciations applied to them; the phrase *financial crisis* does not require a pronunciation. The `content` property represents how you want the word to be output in the transcript.

## Output Locale

It is possible to specify the spelling rules to be used when generating the transcription, based on locale. The `output_locale` configuration setting is used for this. As an example, the following configuration uses the Global English (en) language pack with an output locale of British English (en-GB):

```

{ "type": "transcription",
  "transcription_config": {
    "language": "en",
    "output_locale": "en-GB"
  }
}

```

The following locales are supported in the Global English language pack, if no locale is specified then the ASR engine will use whatever spelling it has learnt as part of our language model training (in other words it will be based on the training data used).

- British English (en-GB)
- US English (en-US)
- Australian English (en-AU)

The following locales are supported for Chinese Mandarin. The default is simplified Mandarin.

- Simplified Mandarin (cmn-Hans)
- Traditional Mandarin (cmn-Hant)

## Advanced punctuation

All Speechmatics language packs support Advanced Punctuation. This uses machine learning techniques to add in more naturalistic punctuation, improving the readability of your transcripts.

The following punctuation marks are supported for each language:

--	--	--

Language(s)	Supported Punctuation	Comment
Cantonese, Mandarin	, 。 ? ! 、	Full-width punctuation supported
Japanese	。 、	Full-width punctuation supported
Hindi	! ? , !	
All other languages	. , ! ?	

If you do not want to see any of the supported punctuation marks in the output, then you can explicitly control this through the `punctuation_overrides` settings, for example:

```
"transcription_config": {
  "language": "en",
  "punctuation_overrides": {
    "permitted_marks": [ ".", ", " ]
  }
}
```

This will exclude exclamation and question marks from the returned transcript.

All Speechmatics output formats support Advanced Punctuation. JSON output places punctuation marks in the results list marked with a `type` of `"punctuation"`.

**Note:** Disabling punctuation may slightly harm the accuracy of speaker diarization. Please see the ["Speaker diarization post-processing"](#) section in these docs for more information.

## Notifications

Customers can poll the appliance to check on the status of the job, before making the call to retrieve the transcript. Where many jobs are being done at scale, this may not be sustainable. A more convenient method - and recommended approach - is to use notifications. This involves a callback to a web service that you control once a job is complete. An HTTP POST request is then made from the Speechmatics appliance once the transcript is available. PUT is also supported where specified

The notification support offered in V1 has been extended and generalized in V2 to support a wider range of customer integration scenarios:

- A callback does not need to include any data attachments, and can be just a signal that the transcript is ready to be fetched through the normal API.
- Multiple pieces of content can be sent as multiple attachments in one request, allowing multiple combination of the input(s) and output(s) of the job to be forwarded to another processing stage. The exception is the audio file, which is deleted upon completion of the transcript. Formatting options for outputs can be specified per attachment.
- You can setup multiple notifications to up to 3 different endpoints: for instance you can send a `jobinfo` notification to one service, and the `transcript` notification to another.
- Callbacks with a single attachment will send the content item as the HTTP request body, rather than using multipart mode. This allows writing an individual item to an object store like Amazon S3.
- HTTP PUT methods are now supported to allow uploading of content directly to an object store such as S3.
  - A set of additional HTTP request headers can be specified in order:
    - To satisfy authentication / authorization requirements for systems that do not support auth tokens in query parameters.
    - To control behaviour of an object store or another existing service endpoint.
- Multiple callbacks can be specified per job.
  - This allows sending individual pieces of content to different URLs.
  - It allows sending combinations of the inputs/outputs to multiple destinations, to support a fanout workflow.

- Callbacks will be invoked in parallel and so may complete in any order. If a downstream workflow depends on getting several items of content delivered as separate callbacks (eg. uploaded as separate items to S3), then the downstream processing logic will need to be robust to the ordering of upload content, and the possibility that only some might succeed.

**\*\* Important Notice \*\***

In the Batch Virtual Appliance, a user **cannot** request the audio file that was part of the original job submission.

## Configuring the Callback

The callback is specified by using the `notification_config` within the config object. For example:

```
curl -X POST 'https://{APPLIANCE_HOST}/v2/jobs' \  
  --form data_file=@example.wav \  
  --form config='{  
    "type": "transcription",  
    "transcription_config": { "language": "en" },  
    "notification_config": [  
      {  
        "url": "https://collector.example.org/callback",  
        "contents": [ "transcript" ],  
        "auth_headers": [  
          "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb"  
        ]  
      }  
    ]  
  }  
}
```

## Accepting the Callback

You need to ensure that the service that you implement to receive the callback notification is capable of processing the Speechmatics transcript using the format that has been specified in the config JSON. When testing your integration you should check the error logs on your web service to ensure that notifications are being accepted and processed correctly.

The callback appends the job ID as a query string parameter with name `id`, as well as the status of the job. As an example, if the job ID is 100, you'd see the following POST request:

```
POST /callback?id=100&status=success HTTP/1.1  
Host: collector.example.org
```

The user agent is `Speechmatics-API/2.0`.

## Configuring your webserver to accept the Callback

Once transcription is complete and the transcript file is available, the Speechmatics Batch Virtual Appliance will send the transcript file in a HTTP POST request (unless otherwise specified) to the client web server specified in the `notification_config` config object. If the appliance does not receive successful 2xx response it will keep trying to send the file until it reaches the set timeout threshold.

If the clients webserver cannot accept the file(s) because it is not configured with a large enough size limit, it will generate a 413 (Request Entity Too Large) response. If the appliance does not receive a 2xx response it will continue to retry sending the file. Users are recommended to check their webserver size limits to ensure they are adequate for the files that will be sent.

## Metadata and Job Tracking

It is now possible to attach richer metadata to a job using the tracking configuration. This metadata can be used to identify transcripts for appropriate data storage and classification, especially where they may have passed through multiple systems using whatever information is relevant to you. The tracking object contains the following properties:

Name	Type	Description	Notes
title	str	The title of the job.	[optional]
reference	str	External system reference.	[optional]
tags	list[str]	Customer-defined tags	[optional]
details	object	Customer-defined JSON structure.	[optional]

Here is an example

```
curl -X POST 'https://${APPLIANCE_HOST}/v2/jobs' \
  -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
  --form data_file=@example.wav \
  --form config='{
    "type": "transcription",
    "transcription_config": { "language": "en" },
    "tracking": {
      "title": "ACME Q12018 Statement",
      "reference": "/data/clients/ACME/statements/segs/2018Q1-seg8",
      "tags": [ "quick-review", "segment" ],
      "details": {
        "client": "ACME Corp",
        "segment": 8,
        "seg_start": 963.201,
        "seg_end": 1091.481
      }
    }
  }
}
```

## SubRip Subtitling Format

SubRip (SRT) is a subtitling format that can be used in to generate subtitles for video content or other workflows. Our SRT output will generate a transcript together with corresponding alignment timestamps. We follow best practice as recommended by major broadcasters in our default line length and number of lines output.

Speechmatics provides a default configuration output for SRT files for both number of lines and line length in characters. You can change these parameters, by passing configuration options described below. To alter default parameters, you must make parameter changes within the configuration file:

```
{
  "type": "transcription",
  "transcription_config": {
    ...
  },
  "output_config": {
    "srt_overrides": {
      "max_line_length": 37,
      "max_lines": 2
    }
  }
}
```

- `max_line_length` : sets maximum count of characters per subtitle line including white space (default: 37 ).
- `max_lines` : sets maximum count of lines in a subtitle section (default: 2 ).

## Word Tagging

## Profanity Tagging

Speechmatics now outputs in JSON transcript only a metadata tag to indicate whether a word is a profanity or not. This is for the following languages:

- English (EN)
- Italian (IT)
- Spanish (ES)

The list of profanities is not alterable. Users do not have to take any action to access this - it is provided in our JSON output as standard Customers can use this tag for their own post-processing in order to identify, redact, or obfuscate profanities and integrate this data into their own workflows. An example of how this looks is below.

```
"results": [
{
  "alternatives": [
    {
      "confidence": 1.0,
      "content": "$PROFANITY",
      "language": "en",
      "speaker": "UU",
      "tags": [
        "profanity"
      ]
    }
  ],
  "end_time": 18.03,
  "start_time": 17.61,
  "type": "word"
}
]
```

## Disfluency Tagging

Speechmatics now outputs in JSON transcript only a metadata tag to indicate whether a word is a disfluency or not in the English language only. A disfluency here refers to a set list of words in English that imply hesitation or indecision. Please note while disfluency can cover a range of items like stuttering and interjections, here it is only used to tag words such as 'hmm' or 'umm'. An example of how this looks is below:

```
"results": [
{
  "alternatives": [
    {
      "confidence": 1.0,
      "content": "hmm",
      "language": "en",
      "speaker": "UU",
      "tags": [
        "disfluency"
      ]
    }
  ],
  "end_time": 18.03,
  "start_time": 17.61,
  "type": "word"
}
]
```



## Getting a Job log file

In case something unexpected happens with your transcription job, you can use the V2 API to retrieve logging for any job. This can be used for internal debugging and troubleshooting, or for providing more information to Speechmatics Support in the event of continued failure.

This feature is available only when a Job ID is generated and returned at the audio submission time. If the audio upload fails and no Job ID is returned, the log will not be available. If a user submits a job and gets a 401 error back (for example) rather than a Job ID, we won't provide logs via this endpoint. The transcription job log is available when the job finishes successfully, but there was then an error with the file processing or the transcript retrieval failed (e.g. an HTTP 500 error when retrieving the transcript).

You must include the Job ID in the request to retrieve logs for any job. You can only request logs from one job ID at a time. Here is a simple example URL:

```
curl -X GET 'https://${APPLIANCE_HOST}/v2/jobs/${JOBID}/log'
```

## V2 API Reference

The Speechmatics Automatic Speech Recognition REST API is used to submit ASR jobs and receive the results. The supported job type is transcription of audio files.

### Version: 2.7.0

### Terms of service

<https://www.speechmatics.com/terms-and-conditions/>

### Contact information

[support@speechmatics.com](mailto:support@speechmatics.com)

### URI scheme

**BasePath:** /v2/jobs/

**Schemes:** HTTPS, HTTP

## Paths

The base URL `https://${APPLIANCE_HOST}/v2/jobs/` is used for REST Speech API requests. If you are using HTTP, the base URL is: `http://${APPLIANCE_HOST}:8082/v2/jobs/`.

### /jobs

Requests without a job ID component are used to create a new job, or to return a list of all submitted jobs

### POST

**Summary:** Create a new job.

### Parameters

Name	Located in	Description	Required	Schema
config	formData	JSON containing a <code>JobConfig</code> model indicating the type and parameters for the recognition job.	Yes	string
data_file	formData	The data file to be processed. Alternatively the data file can be	No	file

fetched from a url specified in `JobConfig`.

## Responses

Code	Description	Schema
201	OK	<a href="#">CreateJobResponse</a>
400	Bad request	<a href="#">ErrorResponse</a>
401	Unauthorized	<a href="#">ErrorResponse</a>
403	Forbidden	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>

## GET

**Summary:** List all jobs.

## Responses

Code	Description	Schema
200	OK	<a href="#">RetrieveJobsResponse</a>
401	Unauthorized	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>

## HTTP Method GET

**Summary:** Get job details, including progress and any error reports.

## /jobs/{jobid}

Requests with a job ID component are used to view the status, transcript or audio data for a job, or remove a given job from the system.

## Parameters

Name	Located in	Description	Required	Schema
jobid	path	ID of the job.	Yes	string

## Responses

Code	Description	Schema
200	OK	<a href="#">RetrieveJobResponse</a>
401	Unauthorized	<a href="#">ErrorResponse</a>
404	Not found	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>

## HTTP Method DELETE

**Summary:** Delete a job and remove all associated resources.

## Parameters

Name	Located in	Description	Required	Schema
------	------------	-------------	----------	--------

jobid	path	ID of the job to delete.	Yes	string
force	query	When set, a running job will be force terminated. When unset (default), a running job will not be terminated and request will return HTTP 423 Locked.	No	boolean

### Responses

Code	Description	Schema
200	The job that was deleted.	<a href="#">DeleteJobResponse</a>
401	Unauthorized	<a href="#">ErrorResponse</a>
404	Not found	<a href="#">ErrorResponse</a>
423	Locked	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>

### HTTP Method GET

#### **/jobs/{jobid}/transcript**

**Summary:** Get the transcript for a transcription job.

#### Parameters

Name	Located in	Description	Required	Schema
jobid	path	ID of the job.	Yes	string
format	query	The transcription format (by default the <code>json-v2</code> format is returned). <code>txt</code> and <code>srt</code> are also supported/	No	string

### Responses

Code	Description	Schema
200	OK	<a href="#">RetrieveTranscriptResponse</a>
401	Unauthorized	<a href="#">ErrorResponse</a>
404	Not found	<a href="#">ErrorResponse</a>
410	Gone	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>

### HTTP Method GET

#### **/jobs/{jobid}/log**

**Summary:** Get the log file for a transcription job.

#### Parameters

Name	Located in	Description	Required	Schema
jobid	path	ID of the job.	Yes	string

### Responses

Code	Description	Schema
------	-------------	--------

200	OK	file
401	Unauthorized	<a href="#">ErrorResponse</a>
404	Not Found	<a href="#">ErrorResponse</a>
410	Gone	<a href="#">ErrorResponse</a>
500	Internal Server Error	<a href="#">ErrorResponse</a>
501	Not Implemented	<a href="#">ErrorResponse</a>

## Models

### ErrorResponse

Name	Type	Description	Required
code	integer	The HTTP status code.	Yes
error	string	The error message.	Yes
detail	string	The details of the error.	No

### TrackingData

Name	Type	Description	Required
title	string	The title of the job.	No
reference	string	External system reference.	No
tags	[ string ]		No
details	object	Customer-defined JSON structure.	No

### DataFetchConfig

Name	Type	Description	Required
url	string	A URL where a file is stored	Yes
auth_headers	[ string ]	A list of additional headers to be added to the input fetch request when using http or https. This is intended to support authentication or authorization, for example by supplying an OAuth2 bearer token.	No

### TranscriptionConfig

Name	Type	Description	Required
language	string	Language model to process the audio input, normally specified as an ISO language code	Yes
output_locale	string	Language locale to be used when generating the transcription output, normally specified as an ISO language code	No
additional_vocab	[ object ]	List of custom words or phrases that should be recognized. Alternative pronunciations can be specified to aid recognition.	No
punctuation_overrides		Control punctuation settings.	No

diarization	string	Specify whether speaker or channel labels are added to the transcript. The default is <code>none</code> . - <b>none</b> : no speaker or channel labels are added. - <b>speaker</b> : speaker attribution is performed based on acoustic matching; all input channels are mixed into a single stream for processing. - <b>channel</b> : multiple input channels are processed individually and collated into a single transcript. - <b>speaker_change</b> : the output indicates when the speaker in the audio changes. No speaker attribution is performed. This is a faster method than speaker. The reported speaker changes may not agree with speaker. - <b>channel_and_speaker_change</b> : both channel and speaker_change are switched on. The speaker change is indicated if more than one speaker are recorded in one channel.	No
speaker_diarization_config	<a href="#">SpeakerDiarizationConfig</a>	Configuration for speaker diarization. Includes <code>speaker_sensitivity</code> : Range between 0 and 1. A higher sensitivity will increase the likelihood of more unique speakers returning. For example, if you see fewer speakers returned than expected, you can try increasing the sensitivity value or if too many speakers are returned try reducing this value. The default is 0.5.	No
speaker_change_sensitivity	float	Ranges between zero and one. Controls how responsive the system is for potential speaker changes. High value indicates high sensitivity. Defaults to 0.4.	No
channel_diarization_labels	[ string ]	Transcript labels to use when using collating separate input channels.	No
speaker_diarization_params		(Deprecated, Ignored) Configuration for speaker diarization	No
operating_point	string	Specify whether to use a <code>standard</code> or <code>enhanced</code> model for transcription. By default the model used is <code>standard</code>	No
enable_entities	Boolean	Specify whether to enable <code>entity</code> types within JSON output, as well as additional <code>spoken_form</code> and <code>written_form</code> metadata. By default <code>false</code>	No

For the diarization parameter, the following values are valid:

Value	Description
<b>none</b>	no speaker or channel labels are added.

<b>speaker</b>	speaker attribution is performed based on acoustic matching; all input channels are mixed into a single stream for processing.
<b>channel</b>	multiple input channels are processed individually and collated into a single transcript.
<b>speaker_change</b>	the output indicates when the speaker in the audio changes. No speaker attribution is performed. This is a faster method than speaker. The reported speaker changes may not agree with speaker.
<b>channel_and_speaker_change</b>	both channel and speaker_change are switched on. The speaker change is indicated if more than one speaker are recorded in one channel.

### SpeakerDiarizationConfig

Additional configuration for the Speaker Diarization feature.

Name	Type	Description	Required
speaker_sensitivity	float	Used for <code>speaker_diarization</code> feature. Range between 0 and 1. A higher sensitivity will increase the likelihood of more unique speakers returning. For example, if you see fewer speakers returned than expected, you can try increasing the sensitivity value, or if too many speakers are returned try reducing this value. The default is 0.5.	No

### NotificationConfig

Name	Type	Description	Required
url	string	The url to which a notification message will be sent upon completion of the job. The <code>job_id</code> and <code>status</code> are added as query parameters, and any combination of the job inputs and outputs can be included by listing them in <code>contents</code> . If <code>contents</code> is empty, the body of the request will be empty. If only one item is listed, it will be sent as the body of the request with <code>Content-Type</code> set to an appropriate value such as <code>application/octet-stream</code> or <code>application/json</code> . If multiple items are listed they will be sent as named file attachments using the multipart content type. If <code>contents</code> is not specified, the <code>transcript</code> item will be sent within the body of the POST request in <code>json-v2</code> format. If the job was rejected or failed during processing, that will be indicated by the <code>status</code> , and any output items that are not available as a result will be omitted. The body formatting rules will still be followed as if all items were available. The user-agent header is set to <code>Speechmatics-API/2.0</code> , or <code>Speechmatics API V2</code> in older API versions.	Yes
contents	[ string ]	Specifies a list of items to be attached to the notification message. When multiple items are requested, they are included as named file attachments.	No
method	string	The method to be used with http and https urls. The default is post.	No
auth_headers	[ string ]	A list of additional headers to be added to the notification request when using http or https. This is intended to support authentication or authorization, for example by supplying an OAuth2 bearer token.	No

### OutputConfig

If you want the transcription output to be in the SubRip Title (SRT) format, **and** you want to alter the default parameters Speechmatics provides you must provide the `output_config` within the config object

Name	Type	Description	Required

srt_overrides	object	Parameters that override default values of srt conversion. max_line_length: sets maximum count of characters per subtitle line including white space. max_lines: sets maximum count of lines in a subtitle section.	No
---------------	--------	--	----

### JobConfig

JSON object that contains various groups of job configuration parameters. Based on the value of `type`, a type-specific object such as `transcription_config` is required to be present to specify all configuration settings or parameters needed to process the job inputs as expected.

If the results of the job are to be forwarded on completion, `notification_config` can be provided with a list of callbacks to be made; no assumptions should be made about the order in which they will occur.

Customer specific job details or metadata can be supplied in `tracking`, and this information will be available where possible in the job results and in callbacks.

Name	Type	Description	Required
type	string		Yes
fetch_data	<a href="#">DataFetchConfig</a>		No
fetch_text	<a href="#">DataFetchConfig</a>		No
transcription_config	<a href="#">TranscriptionConfig</a>		No
notification_config	[ <a href="#">NotificationConfig</a> ]		No
tracking	<a href="#">TrackingData</a>		No
output_config	<a href="#">OutputConfig</a>		No

### CreateJobResponse

In the job response you will see `balance` and `cost` values returned, but these are not used by the appliance; they are only maintained for backwards compatibility with the legacy V1 Cloud Offering, and should be ignored by clients.

Name	Type	Description	Required
id	string	The unique ID assigned to the job. Keep a record of this for later retrieval of your completed job.	Yes

### JobDetails

Document describing a job. JobConfig will be present in JobDetails returned for GET jobs/ request in the Cloud Offering and in Batch Appliance, but it will not be present in JobDetails returned as item in RetrieveJobsResponse in case of Batch Appliance.

Name	Type	Description	Required
created_at	dateTime	The UTC date time the job was created.	Yes
data_name	string	Name of the data file submitted for job.	Yes
duration	integer	The file duration (in seconds). May be missing for fetch URL jobs.	No
id	string	The unique id assigned to the job.	Yes
status	string	The status of the job. * <code>running</code> - The job is actively running. * <code>done</code> - The job completed successfully. * <code>rejected</code> - The job was accepted at first, but later could not be processed by the transcriber. * <code>deleted</code> - The user	Yes

		deleted the job. * <i>expired</i> - The system deleted the job. Usually because the job was in the <i>done</i> state for a very long time.	
config	<a href="#">JobConfig</a>		No

#### RetrieveJobsResponse

Name	Type	Description	Required
jobs	[ <a href="#">JobDetails</a> ]		Yes

#### RetrieveJobResponse

Name	Type	Description	Required
job	<a href="#">JobDetails</a>		Yes

#### DeleteJobResponse

Name	Type	Description	Required
job	<a href="#">JobDetails</a>		Yes

#### JobInfo

Summary information about an ASR job, to support identification and tracking.

Name	Type	Description	Required
created_at	dateTime	The UTC date time the job was created.	Yes
data_name	string	Name of data file submitted for job.	Yes
duration	integer	The data file audio duration (in seconds).	Yes
id	string	The unique id assigned to the job.	Yes
tracking	<a href="#">TrackingData</a>	customer-supplied data	No

#### RecognitionMetadata

Summary information about the output from an ASR job, comprising the job type and configuration parameters used when generating the output.

Name	Type	Description	Required
created_at	dateTime	The UTC date time the transcription output was created.	Yes
type	string		Yes
transcription_config	<a href="#">TranscriptionConfig</a>		No
output_config	<a href="#">OutputConfig</a>		No

#### RecognitionDisplay

Name	Type	Description	Required
direction	string		Yes

#### RecognitionAlternative

List of possible job output item values, ordered by likelihood.



Name	Type	Description	Required
content	string		Yes
confidence	float		Yes
language	string		Yes
display	<a href="#">RecognitionDisplay</a>		No
speaker	string		No
tags	[ string ]		No

### RecognitionResult

An ASR job output item. The primary item types are `word` and `punctuation`. Other item types may be present, for example to provide semantic information of different forms.

Name	Type	Description	Required
channel	string		No
start_time	float		Yes
end_time	float		Yes
entity_class	string	If an entity has been recognised, what type of entity it is. Displayed even if <code>enable_entities</code> is false	Yes
spoken_form	array	For <code>entity</code> results only, the <code>spoken_form</code> is the transcript of the words directly spoken. Only valid if <code>enable_entities</code> is <code>true</code>	No
written_form	array	For <code>entity</code> results only, the <code>written_form</code> is a standardized form of the spoken words. Only valid if <code>enable_entities</code> is <code>true</code>	No
is_eos	boolean	Whether the punctuation mark is an end of sentence character. Only applies to punctuation marks.	No
type	string	New types of items may appear without being requested; unrecognized item types can be ignored. Current types are <code>word</code> , <code>punctuation</code> , <code>speaker_change</code> , and <code>entity</code>	Yes
alternatives	[ <a href="#">RecognitionAlternative</a> ]		No

### RetrieveTranscriptResponse

Name	Type	Description	Required
format	string	Speechmatics JSON transcript format version number.	Yes
job	<a href="#">JobInfo</a>		Yes
metadata	<a href="#">RecognitionMetadata</a>		Yes

## Error Codes

If the Batch Appliance is unable to process a request, then it will typically return one of the error codes listed below. We suggest that your API integration be created to robustly handle these errors.

## 4XX Errors

The 4xx class of status codes is intended for situations in which the request seems is in error.

All 4xx HTTP errors return in JSON format. Users of curl will see this displayed in their terminal, other interfaces may need a JSON parser. The JSON object contains two components: a return code and an error message. More details on these common errors for each endpoint are below. An example error object (from curl) would look like this:

```
curl "http://${APPLIANCE_HOST}:8082/v1.0/user/1/jobs/4087/transcript/"
{
  "detail": "The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.",
  "status": 404,
  "title": "Not Found",
  "type": "about:blank"
}
```

In this example, the URL was malformed (inclusion of trailing '/' character where none is required), giving a 404 status code.

Status Code	Status Message	Detailed Notes
400	Malformed request	Your request contained something unexpected - perhaps a string as a parameter where an integer was expected.
400	Missing data_file	Your request did not contain a data file in the data_file field.
400	No language selected	You did not supply a language code value in the model field.
400	Requested product not available	You requested an unsupported language.
403	Job rejected due to invalid audio	The audio file you submitted was in a format we do not support.
404	Job not found	We could not find a job with the specified id associated.
404	Job was rejected on submission	The job with this id was rejected on submission, probably due to an unsupported file format.
404	Job In Progress	The requested job is still being processed - please wait.
404	Output format Not Supported <b>(V1 API only)</b>	You have requested the transcription output in an unsupported format in the V1 API
422	Unprocessable Entity <b>(V2 API only)</b>	The transcription format should be one of [json-v2 txt srt]. You have requested the transcription output in an unsupported format in the V2 API

## 5XX Errors

The 5xx class of status codes is intended for situations in which your request appears valid but nonetheless the server failed to fulfil an apparently valid request.

All 5xx HTTP errors return a HTML snippet.

Status	Status Message	Detailed Notes
--------	----------------	----------------

Code		
500	Internal Server Error	Our service suffered an internal error. Please please contact us with as much information as possible for us to debug the problem.
502	Bad Gateway	The service is not active at present.
503	Service Temporarily Unavailable	Our service is temporarily overloaded and unable to process your request.

## Formatting Common Entities

### Overview

Entities are commonly recognisable classes of information that appear in languages, for example numbers and dates. Formatting these entities is commonly referred to as Inverse Text Normalisation (ITN). Speechmatics will output entities in a predictable, consistent written form, reducing post-processing work required aiming to make the transcript more readable.

The language pack will use these formatted entities by default in the transcription for all outputs (JSON, text and srt). Additional metadata about these entities can be requested via the API including the spoken words without formatting and the entity class that was used to format it.

### Supported Languages

Entities are supported in the following languages:

- Cantonese
- Chinese Mandarin (Simplified and Traditional)
- English
- French
- German
- Hindi
- Italian
- Japanese
- Portuguese
- Russian
- Spanish

### Using the `enable_entities` parameter

Speechmatics now includes an `enable_entities` parameter. This can be requested via the API. By default this is `false`.

Changing `enable_entities` to `true` will enable a richer set of metadata in the JSON output only. Customers can choose between the default written form, spoken form, or a mixture, for their own workflows.

The changes are as following:

- A new `type - entity` in the JSON output in addition to `word` and `punctuation`. For example: "1.99" would have a `type` of `entity` and a corresponding `entity_class` of `decimal`
- The `entity` will contain the formatted text in the `content` section, like other words and punctuation
  - The `content` can include spaces, non-breaking spaces, and symbols (e.g. \$/£/%)
- A new output element, `entity_class` has been introduced. This provides more detail about how the entity has been formatted. A full list of entity classes is provided below.
- The start and end time of the entity will span all the words that make up that entity
- The entity also contains two ways that the content will be output:

- `spoken_form` - Each individual `word` within the entity, written out in words as it was spoken. Each individual word has its own start time, end time, and confidence score. For example: "one", "million", "dollars"
- `written_form` - The same output as within `entity` content, with a type of `word` instead. If there are spaces in the content it will be split into individual words. For example: "\$1", "million"

## Configuration example

Please see an example configuration file that would request entities:

```
{
  "type": "transcription",
  "transcription_config": {
    "language": "en",
    "enable_entities": true
  }
}
```

## Different entity classes

The following `entity_classes` can be returned. Entity classes indicate how the numerals are formatted. In some cases, the choice of class can be contextual and the class may not be what was expected (for example "2001" may be a "cardinal" instead of "date"). The number of `entity_classes` may grow or shrink in the future.

N.B. Please note existing behaviour for English where numbers from zero to 10 (excluding where they are output as a decimal/money/percentage) are output as **words** is unchanged.

Entity Class	Formatting Behaviour	Spoken Word Form Example	Written Form Example
alphanum	A series of three or more alphanumerics, where an alphanumeric is a digit less than 10, a character or symbol	triple seven five four	77754
cardinal	Any number greater than ten is converted to numbers. Numbers ten or below remain as words. Includes negative numbers	nineteen	19
credit card	A long series of spoken digits less than 10 are converted to numbers. Support for common credit cards	one one one one two two two two three three three three four four four four	1111222233334444
date	Day, month and year, or a year on its own. Any words spoken in the date are maintained (including "the" and "of")	fifteenth of January twenty twenty two	15th of January 2022
decimal	A series of numbers divided by a separator	eighteen point one two	18.12
fraction	Small fractions are kept as words ("half"), complex fractions are converted to numbers separated by "/"	three sixteenths	3/16
money	Currency words are converted to symbols before or after the number (depending on the language)	twenty dollars	\$20
ordinal	Ordinals greater than 10 are output as numbers	forty second	42nd
percentage	Numbers with a per cent have the per cent	duecento percento	200%

	converted to a % symbol		
span	A range expressed as "x to y" where x and y correspond to another entity class	one hundred to two hundred million pounds	100 to £200 million
time	Times are converted to numbers	eleven forty a m	11:40 a.m.
word	Entities that do not match a specific class	hundreds	hundreds

## Output locale styling

Each language has a specific style applied to it for thousands, decimals and where the symbol is positioned for money or percentages.

For example

- English contains commas as separators for numbers above 9999 (example: "20,000"), the money symbol at the start (example: "\$10") and full stops for decimals (example: "10.5")
- German contains full stops as separators for numbers above 9999 (example: "20.000"), the money symbol comes after with a non-breaking space (example: "10 \$") and commas for decimals (example: "10,5")
- French contains non-breaking spaces as separators for numbers above 9999 (example: "20 000"), the money symbol comes after with a non-breaking space (example: "10 \$") and commas for decimals (example: "10,5")

## Example output

Here is an example of a transcript requested with `enable_entities` set to true:

- An `entity` that is "17th of January 2022", including spaces
  - The start and end times span the entire entity
  - An `entity_class` of `date`
  - The `spoken_form` is split into the following individual words: "seventeenth", "of", "January", "twenty", "twenty", "two". Each word has its own start and end time
  - the `written_form` split into the following individual words: "17th", "of", "January", "2022". Each word has its own start and end time

Note:

- By default and when speaker diarization is enabled, `speaker` parameter is added per word within the entity, spoken and written form
- When channel diarization is enabled, `channel` parameter is only added on the `results` parent within the entity and not included in spoken and written form

```
"results": [
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "17th of January 2022",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 3.14,
    "entity_class": "date",
    "spoken_form": [
      {
        "alternatives": [
          {
            "confidence": 1.0,
```

```

        "content": "seventeenth",
        "language": "en",
        "speaker": "UU"
    }
],
"end_time": 1.41,
"start_time": 0.72,
"type": "word"
},
{
    "alternatives": [
        {
            "confidence": 1.0,
            "content": "of",
            "language": "en",
            "speaker": "UU"
        }
    ],
    "end_time": 1.53,
    "start_time": 1.41,
    "type": "word"
},
{
    "alternatives": [
        {
            "confidence": 1.0,
            "content": "January",
            "language": "en",
            "speaker": "UU"
        }
    ],
    "end_time": 2.04,
    "start_time": 1.53,
    "type": "word"
},
{
    "alternatives": [
        {
            "confidence": 1.0,
            "content": "twenty",
            "language": "en",
            "speaker": "UU"
        }
    ],
    "end_time": 2.46,
    "start_time": 2.04,
    "type": "word"
},
{
    "alternatives": [
        {
            "confidence": 1.0,
            "content": "twenty",
            "language": "en",
            "speaker": "UU"
        }
    ],
    "end_time": 2.46,
    "start_time": 2.04,
    "type": "word"
},
{
    "alternatives": [
        {
            "confidence": 1.0,
            "content": "twenty",
            "language": "en",
            "speaker": "UU"
        }
    ],
    "end_time": 2.46,
    "start_time": 2.04,
    "type": "word"
},

```

```
"end_time": 2.79,
"start_time": 2.46,
"type": "word"
},
{
  "alternatives": [
    {
      "confidence": 0.97,
      "content": "two",
      "language": "en",
      "speaker": "UU"
    }
  ],
  "end_time": 3.14,
  "start_time": 2.79,
  "type": "word"
}
],
"start_time": 0.72,
"type": "entity",
"written_form": [
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "17th",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 1.33,
    "start_time": 0.72,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "of",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 1.93,
    "start_time": 1.33,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "January",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 2.54,
```

```

        "start_time": 1.93,
        "type": "word"
    },
    {
        "alternatives": [
            {
                "confidence": 0.99,
                "content": "2022",
                "language": "en",
                "speaker": "UU"
            }
        ],
        "end_time": 3.14,
        "start_time": 2.54,
        "type": "word"
    }
]
}
]

```

If `enable_entities` is set to `false`, the output is as below:

```

"results": [
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "17th",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 1.33,
    "start_time": 0.72,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "of",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 1.93,
    "start_time": 1.33,
    "type": "word"
  },
  {
    "alternatives": [
      {
        "confidence": 0.99,
        "content": "January",
        "language": "en",
        "speaker": "UU"
      }
    ],
    "end_time": 1.93,
    "start_time": 1.33,
    "type": "word"
  }
]

```



```
"end_time": 2.54,  
"start_time": 1.93,  
"type": "word"  
},  
{  
  "alternatives": [  
    {  
      "confidence": 0.99,  
      "content": "2022",  
      "language": "en",  
      "speaker": "UU"  
    }  
  ],  
  "end_time": 3.14,  
  "start_time": 2.54,  
  "type": "word"  
}  
]  
}
```