
Amazon Transcribe

Developer Guide



Amazon Transcribe: Developer Guide

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Transcribe?	1
Recognizing Voices	2
Transcribing Separate Audio Channels	2
Transcribing Streaming Audio	2
Custom Vocabulary	3
Are You a First-time User of Amazon Transcribe ?	3
How It Works	4
Speech Input	6
Transcribing Numbers	6
Alternative Transcriptions	7
Identifying Speakers	10
Transcribing Streaming Audio	11
Channel Identification	11
Custom Vocabularies	13
Create a Custom Vocabulary Using a List	14
Create a Custom Vocabulary Using a Table	14
Character Sets for Custom Vocabularies	16
Vocabulary Filtering	45
Automatic Content Redaction	46
Job Queuing	49
IAM Policies for Job Queuing	49
Getting Started	52
Step 1: Set Up an Account	52
Sign up for AWS	52
Create an IAM User	52
Next Step	53
Step 2: Set Up the AWS CLI	53
Next Step	54
Step 3: Getting Started Using the Console	54
Create a Transcription Job	54
View a Transcription Job	55
Step 4: Getting Started Using the API	58
Getting Started (AWS CLI)	58
Getting Started (SDK for Python)	60
Step 5: Getting Started With Streaming Audio	62
Streaming Transcription	66
Event Stream Encoding	68
Using WebSocket Streaming	69
Adding a Policy for WebSocket Requests to Your IAM Role	69
Creating a Pre-Signed URL	70
Handling the WebSocket Upgrade Response	73
Making a WebSocket Streaming Request	74
Handling a WebSocket Streaming Response	75
Handling WebSocket Streaming Errors	75
Using HTTP/2 Streaming	76
Streaming Request	76
Streaming Response	78
Example Request and Response	79
HTTP/2 Retry Client	81
Using the HTTP/2 Retry Client	88
Amazon Transcribe Medical	91
What Is Amazon Transcribe Medical?	91
Important Notice	91
Transcribing Streaming Audio	91

Supported Specialties	91
Are You a First-time User of Amazon Transcribe Medical?	92
How It Works	92
Amazon Transcribe Medical Features	92
Transcribing Streaming Audio	92
Transcribing Numbers	92
Getting Started	94
Set Up an Account	94
Getting Started (Console)	95
Using WebSocket Streaming	95
Adding a Policy for WebSocket Requests to Your IAM Role	96
Creating a Pre-Signed URL	96
Handling the WebSocket Upgrade Response	100
Event Stream Encoding	101
Making a WebSocket Streaming Request	102
Handling a WebSocket Streaming Response	102
Handling WebSocket Streaming Errors	103
Security	104
Data Protection	104
Encryption at Rest	105
Encryption in Transit	105
Key Management	105
Identity and Access Management	106
Audience	107
Authenticating with Identities	107
Managing Access Using Policies	109
How Amazon Transcribe Works with IAM	110
Identity-Based Policy Examples	112
Troubleshooting	115
Monitoring Amazon Transcribe	117
Monitoring Amazon Transcribe with CloudTrail	117
CloudWatch Events	120
Compliance Validation	122
Resilience	122
Infrastructure Security	122
Guidelines and Quotas	124
Supported Regions	124
Throttling	124
Guidelines	125
Quotas	125
Document History	126
API Reference	130
Actions	130
Amazon Transcribe Service	130
Amazon Transcribe Streaming Service	174
Data Types	178
Amazon Transcribe Service	179
Amazon Transcribe Streaming Service	193
Common Errors	202
Common Parameters	203
AWS Glossary	206

What Is Amazon Transcribe?

Amazon Transcribe uses advanced machine learning technologies to recognize speech in audio files and transcribe them into text. You can use Amazon Transcribe to convert audio to text and to create applications that incorporate the content of audio files. For example, you can transcribe the audio track from a video recording to create closed captioning for the video.

You can use Amazon Transcribe to transcribe the following languages to text:

- Gulf Arabic (ar-AE)
- Modern Standard Arabic (ar-SA)
- Chinese Mandarin - Mainland (zh-CN)
- Dutch (nl-NL)
- Australian English (en-AU)
- British English (en-GB)
- Indian English (en-IN)
- Irish English (en-IE)
- Scottish English (en-AB)
- US English (en-US)
- Welsh English (en-WL)
- Spanish (es-ES)
- US Spanish (es-US)
- French (fr-FR)
- Canadian French (fr-CA)
- Farsi (fa-IR)
- German (de-DE)
- Swiss German (de-CH)
- Hebrew (he-IL)
- Indian Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Malay (ms-MY)
- Portuguese (pt-PT)
- Brazilian Portuguese (pt-BR)
- Russian (ru-RU)
- Tamil (ta-IN)
- Telugu (te-IN)
- Turkish (tr-TR)

You can transcribe streaming audio in the following languages:

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

- French (fr-FR)
- Canadian French (fr-CA)
- US Spanish (es-US)

You can use Amazon Transcribe with other AWS services to create applications. For example, you can:

- Use Amazon Transcribe to convert voice to text, send the text to Amazon Translate to translate it into another language, and send the translated text to Amazon Polly to speak the translated text.
- Use Amazon Transcribe to transcribe recordings of customer service calls for analysis. After transcribing a recording, send the transcription to Amazon Comprehend to identify keywords, topics, or sentiments.
- Use Amazon Transcribe to transcribe live broadcasts such as television to provide real-time subtitles.

To use Amazon Transcribe you store your audio file in an Amazon S3 bucket. The output from the transcription job is also stored in an S3 bucket. Content delivered to Amazon S3 buckets might contain customer content. For more information about removing sensitive data, see [How Do I Empty an S3 Bucket?](#) or [How Do I Delete an S3 Bucket?](#).

Recognizing Voices

Amazon Transcribe can identify the individual speakers in an audio clip, a technique known as *diarization* or *speaker identification*. When you activate speaker identification, Amazon Transcribe includes an attribute that identifies each speaker in the audio clip. You can use speaker identification to:

- identify the customer and the support representative in a recorded customer support call
- identify characters for closed captions
- identify the speaker and questioners in a recorded press conference or lecture

You can specify the number of voices that you want Amazon Transcribe to recognize in an audio clip.

Transcribing Separate Audio Channels

To create a transcript for each channel, or single stream of recorded sound, in an audio file, use *channel identification*. With channel identification, Amazon Transcribe returns two or more transcriptions: a combined transcription of all of the audio channels and a transcription of each audio channel.

Use channel identification when your audio is on multiple channels. For example, use channel identification:

- When your recording has a customer service representative on one channel and a customer on another
- When you transcribe a podcast where the host is recorded on one channel and the guest on another

For more information about channel identification, see [Channel Identification \(p. 11\)](#).

Transcribing Streaming Audio

You can use Amazon Transcribe to transcribe streaming audio in real-time. You send Amazon Transcribe a stream of audio and Amazon Transcribe returns a stream of JSON objects containing the transcription of the audio.

For more information about processing audio streams, see [Streaming Transcription \(p. 66\)](#).

Custom Vocabulary

Create a custom vocabulary to help Amazon Transcribe recognize words that are specific to your use case and improve its accuracy in converting speech to text. For example, you might create a custom vocabulary that includes industry-specific words and phrases.

Use a custom vocabulary to help Amazon Transcribe recognize:

- words that are not being recognized
- unfamiliar words that are specific to your domain

For more information about creating a custom vocabulary, see [Custom Vocabularies \(p. 13\)](#).

Are You a First-time User of Amazon Transcribe ?

If you are a first-time user, we recommend that you read the following sections in order:

1. [How Amazon Transcribe Works \(p. 4\)](#)—Introduces Amazon Transcribe.
2. [Getting Started with Amazon Transcribe \(p. 52\)](#)—Explains how to set up your AWS account and use Amazon Transcribe.
3. [API Reference \(p. 130\)](#)—Contains reference documentation for Amazon Transcribe operations.

How Amazon Transcribe Works

Amazon Transcribe analyzes audio files that contain speech and uses advanced machine learning techniques to transcribe the voice data into text. You can then use the transcription as you would any text document.

To transcribe an audio file, Amazon Transcribe uses three operations:

- [StartTranscriptionJob \(p. 163\)](#) – Starts an asynchronous job to transcribe the speech in an audio file to text.
- [ListTranscriptionJobs \(p. 154\)](#) – Returns a list of transcription jobs that have been started. You can specify the status of the jobs that you want the operation to return. For example, you can get a list of all pending jobs, or a list of completed jobs.
- [GetTranscriptionJob \(p. 145\)](#) – Returns the result of a transcription job. The response contains a link to a JSON file containing the results.

To transcribe streaming audio to text, Amazon Transcribe provides one operation:

- [StartStreamTranscription \(p. 175\)](#) – Starts a bi-directional HTTP/2 stream where audio is streamed to Amazon Transcribe and the transcription results are streamed to your application.

You can also start a WebSocket protocol stream to send audio the Amazon Transcribe. For more information, see [Using Amazon Transcribe Streaming with WebSockets \(p. 69\)](#).

You can use Amazon Transcribe to create and manage custom vocabularies for your solution. A custom vocabulary gives Amazon Transcribe more information about how to process speech in an audio clip.

- [CreateVocabulary \(p. 132\)](#) – Creates a custom vocabulary that you can use in your transcription jobs.
- [DeleteVocabulary \(p. 141\)](#) – Deletes a custom vocabulary from your account.
- [GetVocabulary \(p. 148\)](#) – Gets information about a custom vocabulary and a URL that you can use to download the contents of a vocabulary.
- [ListVocabularies \(p. 157\)](#) – Gets a list of custom vocabularies in your account.
- [UpdateVocabulary \(p. 168\)](#) – Updates an existing vocabulary.

You can transcribe speech in any of the following languages:

- Gulf Arabic (ar-AE)
- Modern Standard Arabic (ar-SA)
- Chinese Mandarin-Mainland (zh-CN)
- Dutch (nl-NL)
- Australian English (en-AU)
- British English (en-GB)
- Indian English (en-IN)
- Irish English (en-IE)
- Scottish English (en-AB)

- US English (en-US)
- Welsh English (en-WL)
- French (fr-FR)
- Canadian French (fr-CA)
- Farsi (fa-IR)
- German (de-DE)
- Swiss German (de-CH)
- Hebrew (he-IL)
- Indian Hindi (hi-IN)
- Indonesian (id-ID)
- Italian (it-IT)
- Japanese (ja-JP)
- Korean (ko-KR)
- Malay (ms-MY)
- Portuguese (pt-PT)
- Brazilian Portuguese (pt-BR)
- Russian (ru-RU)
- Spanish (es-ES)
- US Spanish (es-US)
- Tamil (ta-IN)
- Telugu (te-IN)
- Turkish (tr-TR)

You can use streaming transcription for the following languages:

- Australian English (en-AU)
- British English (en-GB)
- US English (en-US)
- French (fr-FR)
- Canadian French (fr-CA)
- US Spanish (es-US)

Topics

- [Speech Input \(p. 6\)](#)
- [Transcribing Numbers \(p. 6\)](#)
- [Alternative Transcriptions \(p. 7\)](#)
- [Identifying Speakers \(p. 10\)](#)
- [Transcribing Streaming Audio \(p. 11\)](#)
- [Channel Identification \(p. 11\)](#)
- [Custom Vocabularies \(p. 13\)](#)
- [Vocabulary Filtering \(p. 45\)](#)
- [Automatic Content Redaction \(p. 46\)](#)

- [Job Queuing \(p. 49\)](#)

Speech Input

To transcribe an audio file, you use a transcription job. You store the file as an object in an Amazon S3 bucket. The input file must be:

- In FLAC, MP3, MP4, or WAV file format
- Less than 4 hours in length or less than 2 Gb of audio data

Specify the language and format of the input file.

For best results:

- Use a lossless format, such as FLAC or WAV, with PCM 16-bit encoding.
- Use a sample rate of 8000 Hz for telephone audio.

Transcribing Numbers

When you are transcribing US English, Australian English, British English, or Indian English audio using the [StartTranscriptionJob \(p. 163\)](#) operation, numbers are transcribed as digits instead of words. For streaming transcription, numbers are transcribed as digits for US English and British English only. For example, the spoken number "one thousand two hundred forty-two" is transcribed as 1242.

In all other languages numbers are transcribed into their word forms.

Numbers are transcribed according to the following rules:

Rule	Description
Convert cardinal numbers greater than 10 to numbers.	<ul style="list-style-type: none">• "Fifty five" > 55• "a hundred" > 100• "One thousand and thirty one" > 1031• "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" > 123,456,789
Convert cardinal numbers followed by "million" or "billion" to numerals followed by a word when "million" or "billion" is not followed by a number.	<ul style="list-style-type: none">• "one hundred million" > 100 million• "one billion" > 1 billion• "two point three million" > 2.3 million
Convert ordinal numbers greater than 10 to numbers.	<ul style="list-style-type: none">• "Forty third" > 43rd• "twenty sixth avenue" > 26th avenue
Convert fractions to their numeric format.	<ul style="list-style-type: none">• "a quarter" > 1/4• "three sixteenths" > 3/16• "a half" > 1/2• "a hundredth" > 1/100
Convert numbers less than 10 to digits if there are more than one in a row.	<ul style="list-style-type: none">• "three four five" > 345

Rule	Description
	<ul style="list-style-type: none"> "My phone number is four two five five five five one two one two" > 4255551212
Decimals are indicated by "dot" or "point."	<ul style="list-style-type: none"> "three hundred and three dot five" > 303.5 "three point twenty three" > 3.23 "zero point four" > 0.4 "point three" > 0.3
Convert the word "percent" after a number to a percent symbol (%).	<ul style="list-style-type: none"> "twenty three percent" > 23% "twenty three point four five percent" > 23.45%
Convert the words "dollar," "US dollar," "Australian dollar," "AUD," or "USD" after a number to a dollar sign (\$) before the number.	<ul style="list-style-type: none"> "one dollar and fifteen cents" > \$1.15 "twenty three USD" > \$23 "twenty three Australian dollars" > \$23
Convert the words "pounds," "British pounds," or "GDB" after a number to pound sign (£) before the number.	<ul style="list-style-type: none"> "twenty three pounds" > £23 "I have two thousand pounds" > I have £2,000 "five pounds thirty three pence" > £5.33
Convert the words "rupees," "Indian rupees," or "INR" after a number to rupee sign (₹) before the number.	<ul style="list-style-type: none"> "twenty three rupees" > ₹23 "fifty rupees thirty paise" > ₹50.30
Convert times to numbers.	<ul style="list-style-type: none"> "seven a m eastern standard time" > 7 a.m. eastern standard time "twelve thirty p m" > 12:30 p.m.
Combine years expressed as two digits into four. Only valid for the 20th, 21st, and 22nd centuries.	<ul style="list-style-type: none"> "nineteen sixty two" > 1962 "the year is twenty twelve" > the year is 2012 "twenty nineteen" > 2019 "twenty one thirty" > 2130
Convert dates to numbers.	<ul style="list-style-type: none"> "May fifth twenty twelve" > May 5th 2012 "May five twenty twelve" > May 5 2012 "five May twenty twelve" > 5 May 2012
Separate spans of numbers by the word "to."	<ul style="list-style-type: none"> "twenty three to thirty seven" > 23 to 37

Alternative Transcriptions

When Amazon Transcribe transcribes an audio file, it returns the transcription with the highest confidence level. You can specify that Amazon Transcribe return additional transcriptions with lower confidence levels. Use alternative transcriptions to see different interpretations of the transcribed audio. For example, in an application that enables a person to review the transcription, you can present the alternative transcriptions for the person to choose from. Alternative transcriptions are only available for the [StartTranscriptionJob \(p. 163\)](#) operation.

You can configure Amazon Transcribe to return alternative transcription using the console or by using the Amazon Transcribe API. To get alternative transcriptions using the API, set the `ShowAlternatives` field to `true` and set the `MaxAlternatives` field to the number of alternatives to return when you call the [StartTranscriptionJob \(p. 163\)](#) operation. You can specify that Amazon Transcribe return up to 10 alternative transcriptions.

You can combine alternative transcriptions with speaker identification and channel identification. Alternative transcriptions are available in all supported languages.

Alternatives are presented at the segment level of the transcription. Segments are defined by natural pauses in speech, such as a change in speaker or pause in the audio. For example, the spoken phrase "It is raining today in Seattle, but not in Portland" is separated into two segments: "It is raining today in Seattle" and "but not in Portland."

Amazon Transcribe returns an overall transcription of your audio file in the response. When you have configured Amazon Transcribe to return alternatives, the overall transcription is built from the segment alternatives with the highest confidence level. Alternative transcriptions are returned in the `segments` structure in the output JSON. If Amazon Transcribe doesn't find alternatives, it returns fewer than the number of alternatives specified in the `MaxAlternatives` field.

The following is the JSON output from Amazon Transcribe. It is the transcription output for this input: **Uh, you can just call this number if I don't pick up, just leave a voicemail and I'll get back to you. Okay. And that's the number. The 1166 number, you mean?**

The following is the JSON output with `ShowAlternatives` set to `false`.

```
{
  "results": {
    "transcripts": [
      "Uh, you can just call this number if I don't pick up and leave a voicemail and I'll get back to you. Okay. And that's the number. The 1166 number, you mean"
    ],
    "items": [
      {
        "start_time": 12.35,
        "end_time": 12.57,
        "alternatives": [
          {
            "confidence": 0.9989,
            "content": "Uh"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity.
    ]
  }
}
```

The following is the JSON output for the same input with `ShowAlternatives` set to `true` and `MaxAlternatives` set to 2.

```
{
  "results": {
    "transcripts": [
      "Uh, you can just call this number if I don't pick up and leave a voicemail and I'll get back to you. Okay. And that's the number. The 1166 number, you mean"
    ],
    "items": [
      {
        "start_time": 12.35,
        "end_time": 12.57,
        "alternatives": [
          {
            "confidence": 0.9989,
            "content": "Uh"
          }
        ]
      }
    ]
  }
}
```

```

        ],
        "type": "pronunciation"
    },
    Items removed for brevity..
],
"segments": [
    {
        "start_time": 11.84,
        "end_time": 19.665,
        "alternatives": [
            {
                "transcript": "Uh, you can just call this number if I don't pick up
and leave a voicemail and I'll get back to you.",
                "items": [
                    {
                        "start_time": 12.35,
                        "end_time": 12.57,
                        "confidence": 0.9989,
                        "content": "Uh",
                        "type": "pronunciation"
                    },
                    Items removed for brevity.
                    {
                        "start_time": 16.42,
                        "end_time": 16.52,
                        "confidence": 0.7572,
                        "content": "and",
                        "type": "pronunciation"
                    },
                    Items removed for brevity.
                ]
            },
            {
                "transcript": "Uh, you can just call this number if I don't pick
up, just leave a voicemail and I'll get back to you.",
                "items": [
                    {
                        "start_time": 12.35,
                        "end_time": 12.57,
                        "confidence": 0.9989,
                        "content": "Uh",
                        "type": "pronunciation"
                    },
                    Items removed for brevity..
                    {
                        "start_time": 16.42,
                        "end_time": 16.52,
                        "content": ",",
                        "type": "punctuation"
                    },
                    {
                        "start_time": 16.42,
                        "end_time": 16.52,
                        "confidence": 0.8934,
                        "content": "just",
                        "type": "punctuation"
                    },
                    Items removed for brevity..
                ]
            },
            Alternatives removed for brevity.
        ]
    },
    Segments removed for brevity..
]
}

```

```
}
```

Identifying Speakers

You can have Amazon Transcribe identify the different speakers in an audio clip, a process known as *diarization* or *speaker identification*. When you enable speaker identification, Amazon Transcribe labels each fragment with the speaker that it identified.

You can specify that Amazon Transcribe identify between 2 and 10 speakers in the audio clip. You get the best performance when the number of speakers that you ask to identify matches the number of speakers in the input audio.

To turn on speaker identification, set the `MaxSpeakerLabels` and `ShowSpeakerLabels` field of the `Settings` field when you call the [StartTranscriptionJob](#) (p. 163) operation. You must set both fields or else Amazon Transcribe returns an exception.

When Amazon Transcribe completes a transcription job, it creates a JSON file that contains the results and saves the file in an S3 bucket. The file is identified by a user-specific URI. Use the URI to get the results.

The following is the JSON file for a short audio file:

```
{
  "jobName": "job ID",
  "accountId": "account ID",
  "results": {
    "transcripts": [{
      "transcript": "Professional answer."
    }],
    "speaker_labels": {
      "speakers": 1,
      "segments": [{
        "start_time": "0.000000",
        "speaker_label": "spk_0",
        "end_time": "1.430",
        "items": [{
          "start_time": "0.100",
          "speaker_label": "spk_0",
          "end_time": "0.690"
        }, {
          "start_time": "0.690",
          "speaker_label": "spk_0",
          "end_time": "1.210"
        }
      ]
    }
  ],
  "items": [{
    "start_time": "0.100",
    "end_time": "0.690",
    "alternatives": [{
      "confidence": "0.8162",
      "content": "Professional"
    }],
    "type": "pronunciation"
  }, {
    "start_time": "0.690",
    "end_time": "1.210",
    "alternatives": [{
      "confidence": "0.9939",
      "content": "answer"
    }],
    "type": "pronunciation"
  }
]
```

```
    }],  
    "type": "pronunciation"  
  }, {  
    "alternatives": [{  
      "content": "."  
    }],  
    "type": "punctuation"  
  }]  
},  
"status": "COMPLETED"  
}
```

Transcribing Streaming Audio

Streaming transcription takes a stream of your audio data and transcribes it in real time. Streaming uses HTTP/2 or WebSocket streams so that the results of the transcription are returned to your application while you send more audio to Amazon Transcribe. Use streaming transcription when you want to make the results of live audio transcription available immediately or when you have an audio file that you want to process as it is transcribed.

You can use streaming transcription with the following languages:

- Australian English (en-AU)
- British English (en-GB)
- US English (en-US)
- French (fr-FR)
- Canadian French (fr-CA)
- US Spanish (es-US)

To learn more, see [Streaming Transcription \(p. 66\)](#).

Channel Identification

When an audio file has multiple channels that you want to transcribe into separate transcriptions that identify the channel that contains the speech, use *channel identification*. For example, if you have a customer support representative on one channel and a customer on another, use channel identification to create a transcription that is identified by each channel and a single transcription that combines them.

Amazon Transcribe splits your audio file into multiple channels and transcribes the channels separately. After transcribing all channels, Amazon Transcribe also merges the transcriptions to create a single transcription. It returns all of the transcriptions in a single result file.

Speakers' utterances are ordered by their start time. An *utterance* is a unit of speech on the audio channel that is typically separated from other utterances by silence. If an utterance on one channel overlaps one on another channel, Amazon Transcribe orders them in the transcription by their start times. Utterances that overlap in the input audio don't overlap in the transcription output.

You can enable channel identification in the Amazon Transcribe console or with the API. In the console, choose **Channel identification** when you create the transcription job. When you use the API, set the `ChannelIdentification` flag when you call the [StartTranscriptionJob \(p. 163\)](#) operation.

The following is the abbreviated output for a conversation on two channels:

```
{
```

```

"jobName": "job id",
"accountId": "account id",
"results": {
  "transcripts": [
    {
      "transcript": "When you try ... It seems to ..."
    }
  ],
  "channel_labels": {
    "channels": [
      {
        "channel_label": "ch_0",
        "items": [
          {
            "start_time": "12.282",
            "end_time": "12.592",
            "alternatives": [
              {
                "confidence": "1.0000",
                "content": "When"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.592",
            "end_time": "12.692",
            "alternatives": [
              {
                "confidence": "0.8787",
                "content": "you"
              }
            ],
            "type": "pronunciation"
          }
        ],
        "start_time": "12.702",
        "end_time": "13.252",
        "alternatives": [
          {
            "confidence": "0.8318",
            "content": "try"
          }
        ],
        "type": "pronunciation"
      },
      Transcription abbreviated
    ]
  },
  {
    "channel_label": "ch_1",
    "items": [
      {
        "start_time": "12.379",
        "end_time": "12.589",
        "alternatives": [
          {
            "confidence": "0.5645",
            "content": "It"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "12.599",
        "end_time": "12.659",

```



```
        "alternatives": [
          {
            "confidence": "0.2907",
            "content": "seems"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "12.669",
        "end_time": "13.029",
        "alternatives": [
          {
            "confidence": "0.2497",
            "content": "to"
          }
        ],
        "type": "pronunciation"
      },
      Transcription abbreviated
    ]
  }
}
```

Custom Vocabularies

Topics

- [Create a Custom Vocabulary Using a List \(p. 14\)](#)
- [Create a Custom Vocabulary Using a Table \(p. 14\)](#)
- [Character Sets for Custom Vocabularies \(p. 16\)](#)

You can give Amazon Transcribe more information about how to process speech in your input file by creating a custom vocabulary. A *custom vocabulary* is a list of specific words that you want Amazon Transcribe to recognize in your audio input. These are generally domain-specific words and phrases, words that Amazon Transcribe isn't recognizing, or proper nouns.

Custom vocabularies work best when used to target specific words or phrases. We recommend that you create separate small vocabularies tailored to specific audio recordings instead of creating a single vocabulary with many terms to use for all of your recordings. You can have up to 100 vocabularies in your account. The size limit for a custom vocabulary is 50 Kb.

You specify the custom vocabulary in a text file. You can specify either a list of words in the vocabulary, or a four-column table that gives you more control over the input and output of the words in the custom vocabulary.

For more information about creating a custom vocabulary, see [Create a Custom Vocabulary Using a List \(p. 14\)](#) and [Create a Custom Vocabulary Using a Table \(p. 14\)](#).

To create a custom vocabulary, use the [CreateVocabulary \(p. 132\)](#) operation or the [Amazon Transcribe console](#). After you submit the `CreateVocabulary` request, Amazon Transcribe processes the vocabulary. To see the processing status of the vocabulary, use the console or the [GetVocabulary \(p. 148\)](#) operation.

Note

If you are uploading the custom vocabulary using the Amazon Transcribe console, you must use vocabulary list instead of a vocabulary table. To use the console to create a custom vocabulary using a vocabulary table, the source file must be in an Amazon S3 bucket.

To use the custom vocabulary, set the `VocabularyName` field of the `Settings` field when you call the [StartTranscriptionJob](#) (p. 163) operation or choose the vocabulary in the console when you create the transcription job.

Create a Custom Vocabulary Using a List

You can create a custom vocabulary using a list of words or phrases in a text file. You can place each word on its own line, or you can put multiple words on a single line, separating the words or phrases from each other with a comma.

Each entry must contain:

- Fewer than 256 characters, including hyphens
- Only characters from the allowed character set

For valid character sets, see [Character Sets for Custom Vocabularies](#) (p. 16).

If an entry in the list is a phrase, separate the words of the phrase with a hyphen. For example, if the phrase is **Los Angeles**, you would enter it in the file as **Los-Angeles**.

Enter acronyms or other words whose letters should be pronounced individually as single letters separated by dots, such as **A.B.C.** or **F.B.I.**. To enter the plural form of an acronym, such as "ABCs", separate the "s" from the acronym with a hyphen: **A.B.C.-s**. You can use either upper or lower case letters to enter an acronym. Acronyms are supported in the following languages:

- Dutch
- All English variants
- All French variants
- All German variants
- Hindi
- Indonesian
- Italian
- Malay
- All Portuguese variants
- All Spanish variants
- Turkish

The following example shows an input file with the vocabulary words and phrases on separate lines:

```
Los-Angeles
F.B.I.
Etienne
```

The following example shows an input file with the vocabulary words and phrases on a single line, separated by commas:

```
Los-Angeles,F.B.I.,Etienne
```

Create a Custom Vocabulary Using a Table

You can create a custom vocabulary by creating a table in a text file. Each row in the table is a word or phrase followed by the optional `IPA`, `SoundsLike`, and `DisplayAs` fields. Each field must contain:

- Fewer than 256 characters, including hyphens
- Only characters from the allowed character set

For valid character sets, see [Character Sets for Custom Vocabularies \(p. 16\)](#)

Place each word or phrase in your text file on a separate line. Separate the fields with TAB characters. Save the file with the extension `.txt` in an Amazon S3 bucket in the same region that you are calling the API.

The following examples are input files in text format. The examples use spaces to align the columns. Your input files should use TAB characters to separate the columns. Include spaces only in the `IPA` and `DisplayAs` columns. If you copy these examples, remove the extra spaces between columns and replace "[TAB]" with a TAB character.

```
Phrase      [TAB]IPA      [TAB]SoundsLike[TAB]DisplayAs
Los-Angeles[TAB]      [TAB]      [TAB]Los Angeles
F.B.I.      [TAB]# f b i a#[TAB]      [TAB]FBI
Etienne     [TAB]      [TAB]eh-tee-en [TAB]
```

Columns can be entered in any order. The following are also valid structures for the custom vocabulary input file.

```
Phrase      [TAB]SoundsLike[TAB]IPA      [TAB]DisplayAs
Los-Angeles[TAB]      [TAB]      [TAB]Los Angeles
F.B.I       [TAB]      [TAB]# f b i a#[TAB]FBI
Etienne     [TAB]eh-tee-en [TAB]      [TAB]
```

```
DisplayAs   [TAB]SoundsLike[TAB]IPA      [TAB]Phrase
Los Angeles[TAB]      [TAB]      [TAB]Los-Angeles
FBI         [TAB]      [TAB]# f b i a#[TAB]F.B.I.
            [TAB]eh-tee-en [TAB]      [TAB]Etienne
```

- **Phrase** – The word or phrase that should be recognized.

If the entry is a phrase, separate the words with a hyphen (-). For example, you type **Los Angeles** as **Los-Angeles**.

Enter acronyms or other words whose letters should be pronounced individually as single letters followed by dots, such **A.B.C.** or **F.B.I.**. To enter the plural form of an acronym, such as "ABCs," separate the "s" from the acronym with a hyphen: **"A.B.C.-s"**. You can use either upper- or lower-case letters to enter an acronym. For a list of languages that support acronyms, see [Create a Custom Vocabulary Using a List \(p. 14\)](#).

The `Phrase` field is required. You can use any of the allowed characters for the input language. For the list of allowed characters, see the individual languages. If you do not specify the `DisplayAs` field, Amazon Transcribe uses the contents of the `Phrase` field in the output file.

- **IPA** – To specify the pronunciation of your word or phrase, you can include characters in the [International Phonetic Alphabet \(IPA\)](#) in this field. The `IPA` field can't contain leading or trailing spaces, and you must use a single space to separate each phoneme in the input. For example, in English you would enter the phrase **Los-Angeles** as **l # s æ n # # l # s**. You would enter the phrase **F.B.I.** as **# f b i a#**.

If you don't specify the contents of the `IPA` field, you must include a blank `IPA` field. If you specify the `IPA` field, you can't specify the `SoundsLike` field.

For a list of allowed IPA characters for a specific language, see the table for individual languages.

- **SoundsLike** – You can break a word or phrase down into smaller pieces and provide a pronunciation for each piece using the standard orthography of the language to mimic the way that the word sounds. For example, in English you can provide pronunciation hints for the phrase **Los-Angeles** like this: **loss-ann-gel-es**. The hint for the word **Etienne** would look like this: **eh-tee-en**. You separate each part of the hint with a hyphen (-).

If you don't specify the `SoundsLike` field you must include a blank `SoundsLike` field. If you specify the `SoundsLike` field you can't specify the `IPA` field.

You can use any of the allowed characters for the input language. For the list of allowed characters, see the individual languages.

- **DisplayAs** – Defines the how the word or phrase looks when it's output. For example, if the word or phrase is **Los-Angeles**, you can specify the display form as "Los Angeles" so that the hyphen is not present in the output.

If you don't specify the `DisplayAs` field, Amazon Transcribe uses the `Phrase` field from the input file in the output.

You can use any UTF-8 character in the `DisplayAs` field.

Character Sets for Custom Vocabularies

Amazon Transcribe limits the characters that you can use to create custom vocabularies. You can use the following character sets for each language.

Topics

- [Arabic Character Set \(p. 17\)](#)
- [Chinese Character Set \(p. 18\)](#)
- [Dutch Character Set \(p. 19\)](#)
- [English Character Set \(p. 20\)](#)
- [Farsi Character Set \(p. 21\)](#)
- [French Character Set \(p. 23\)](#)
- [German Character Set \(p. 24\)](#)
- [Hebrew Character Set \(p. 26\)](#)
- [Hindi Character Set \(p. 27\)](#)
- [Indonesian Character Set \(p. 29\)](#)
- [Italian Character Set \(p. 30\)](#)
- [Japanese Character Set \(p. 32\)](#)
- [Korean Character Set \(p. 33\)](#)
- [Malay Character Set \(p. 34\)](#)
- [Portuguese Character Set \(p. 35\)](#)
- [Russian Character Set \(p. 36\)](#)
- [Spanish Character Set \(p. 38\)](#)
- [Tamil Character Set \(p. 39\)](#)
- [Telugu Character Set \(p. 41\)](#)
- [Turkish Character Set \(p. 43\)](#)

Arabic Character Set

For Arabic custom vocabularies, you can use the following Unicode characters in the `Phrase` and `SoundsLike` fields. You can also use the hyphen (-) character to separate words.

Character	Code	Character	Code
ء	0621	س	0633
آ	0622	ش	0634
أ	0623	ص	0635
ؤ	0624	ض	0636
إ	0625	ط	0637
ئ	0626	ظ	0638
ا	0627	ع	0639
ب	0628	غ	063A
ة	0629	ف	0641
ت	062A	ق	0642
ث	062B	ك	0643
ج	062C	ل	0644
ح	062D	م	0645
خ	062E	ن	0646
د	062F	ه	0647
ذ	0630	و	0648
ر	0631	ى	0649
ز	0632	ي	064A

You can use the following International Phonetic Alphabet characters in the `IPA` field of the vocabulary input file:

Character	Code	Character	Code
a	0061	tʃ	0074 02E4
a:	0061 02D0	u	0075
b	0062	u:	0075 02D0
d	0064	v	0076
dʃ	0064 02E4	w	0077
f	0066	x	0078

Character	Code	Character	Code
h	0068	z	007A
i	0069	z ^ɸ	007A 02E4
i:	0069 02D0	ð	00F0
j	006A	ð ^ɸ	00F0 02E4
k	006B	ħ	0127
l	006C	ʎ	0263
m	006D	ɪ	026A
n	006E	†	026B
p	0070	ƒ	0283
q	0071	Ʒ	0292
r	0072	ʔ	0294
s	0073	ɸ	0295
s ^ɸ	0073 02E4	θ	03B8
t	0074	χ	03C7

Chinese Character Set

For Chinese custom vocabularies, the `Phrase` field can use any of the characters listed in the following file on GitHub.

- [chinese-character-set.txt](#)

The `SoundsLike` field can contain the pinyin syllables listed in the following file on GitHub.

- [pinyin-set.txt](#)

When you use pinyin syllables in the `SoundsLike` field, separate the syllables with a hyphen (-).

Amazon Transcribe represents the four tones in Mandarin Chinese using numbers. The following table shows how tone marks are mapped for the word "ma."

Tone	Tone Mark	Tone Number
Tone 1	mā	ma1
Tone 2	má	ma2
Tone 3	mǎ	ma3
Tone 4	mà	ma4

Chinese custom vocabularies don't use the `IPA` field, but you must still include the `IPA` header in the vocabulary table.

The following example is an input file in text format. The example uses spaces to align the columns. Your input files should use TAB characters to separate the columns. Include spaces only in the `DisplayAs` column.

Phrase	SoundsLike	IPA	DisplayAs
##	kang1-jian4		
##	qian3-ze2		
####	guo2-fang2-da4-chen2		
#####	shi4-jie4-bo4-lan3-hui4	###	

Dutch Character Set

For Dutch custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
à	00E0	î	00EE
á	00E1	ï	00EF
â	00E2	ñ	00F1
ä	00E4	ò	00F2
ç	00E7	ó	00F3
è	00E8	ô	00F4
é	00E9	ö	00F6
ê	00EA	ù	00F9
ë	00EB	ú	00FA
ì	00EC	û	00FB
í	00ED	ü	00FC

You can use the following International Phonetic Alphabet characters in the `IPA` field of the vocabulary input file:

Character	Code	Character	Code
a:	0061 003A	z	007A

Character	Code	Character	Code
b:	0062 02D0	ø:	00F8 003A
b	0062	ŋ	014B
d	0064	œy	0153 0079
e:	0065 02D0	œ:	0153 02D0
f	0066	ɑ	0251
g	0067	ɔ	0254
i	0069	ɔu	0254 0075
j	006A	ɔ:	0254 02D0
k	006B	ə	0259
l	006C	ɛ	025B
m	006D	ɛ:	025B 003A
n	006E	ɛi	025B 0069
o:	006F 02D0	ɦ	0266
p	0070	ɪ	026A
s	0073	ɹ	0272
t	0074	ɹ	027E
u	0075	ʃ	0283
v	0076	ɣ	028F
w	0077	ʒ	0292
y	0079	χ	03C7

English Character Set

For English custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can use the following International Phonetic Alphabet characters in the `IPA` field of the vocabulary input file:

Character	Code	Character	Code
au	0061 028A	w	0077
ai	0061 026A	z	007A
b	0062	æ	00E6
d	0064	ð	00F0
ei	0065 026A	ŋ	014B
f	0066	ɑ	0251
g	0067	ɔ	0254
h	0068	ɔɪ	0254 026A
i	0069	ə	0259
j	006A	ɛ	025B
k	006B	ʒ	025D
l	006C	g	0261
ɹ	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
ou	006F 028A	ʌ	028C
p	0070	ʌ	028D
s	0073	ʒ	0292
t	0074	ɔʒ	02A4
u	0075	ʈʂ	02A7
v	0076	θ	03B8

Farsi Character Set

For Farsi custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields.

Character	Code	Character	Code
ء	0621	ط	0638
آ	0622	ع	0639
إ	0623	غ	063A
ؤ	0624	ف	0641

Character	Code	Character	Code
ئ	0626	ق	0642
ا	0627	ل	0644
ب	0628	م	0645
ت	062A	ن	0646
ث	062B	ه	0647
ج	062C	و	0648
ح	062D	ـ	064E
خ	062E	ـ	064F
د	062F	ـ	0650
ذ	0630	ـ	0651
ر	0631	پ	067E
ز	0632	چ	0686
س	0633	ژ	0698
ش	0634	ک	06A9
ص	0635	گ	06AF
ض	0636	ی	06CC
ط	0637		

You can use the following International Phonetic Alphabet in the IPA field of your vocabulary file:

Character	Code	Character	Code
b	0062	u	0075
d	0064	v	0076
f	0066	z	007A
g	0067	æ	00E6
h	0068	ɒ	0252
i	0069	ɛ	025B
j	006A	ɹ	027E
k	006B	ɸ	0281
l	006C	ʃ	0283
m	006D	ʒ	0292
n	006E	ʔ	0294

Character	Code	Character	Code
o	006F	ʔ	0294
p	0070	ɔ̃	02A4
s	0073	ʃ	02A7
t	0074	χ	03C7

French Character Set

For French custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
À	00C0	à	00E0
Â	00C2	â	00E2
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Î	00CE	î	00EE
Ï	00CF	ï	00EF
Ô	00D4	ô	00F4
Ö	00D6	ö	00F6
Ù	00D9	ù	00F9
Û	00DB	û	00FB
Ü	00DC	ü	00FC

You can use the following International Phonetic Alphabet in the `IPA` field of your vocabulary file:

Character	Code	Character	Code
a	0061	z	007A
b	0062	ã	00E3
d	0064	ö	00F5
e	0065	ø	00F8
f	0066	ŋ	014B
i	0069	œ	0153
j	006A	œ~	0153 0303
k	006B	Ƶ	0250
l	006C	ɔ	0254
m	006D	ə	0259
n	006E	ε	025B
o	006F	g	0261
p	0070	ɥ	0265
s	0073	ɹ	0272
t	0074	ɻ	0281
u	0075	ɟ	0283
v	0076	ʒ	0292
w	0077	ě	1EBD
y	0079		

German Character Set

For German custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
ä	00E4	Ä	00C4

Character	Code	Character	Code
ö	00F6	Ö	00D6
ü	00FC	Ü	00DC
ß	00DF		

You can use the following International Phonetic Alphabet characters in the IPA field of the vocabulary input file:

Character	Code	Character	Code
a	0061	ts	0074 0073
aɪ	0061 026A	u:	0075 02D0
aʊ	0061 028A	v	0076
a:	0061 02D0	x	0078
b	0062	z	007A
d	0064	y:	0079 02D0
e:	0065 02D0	ã	00E3
f	0066	ç	00E7
g	0067	ø:	00F8 02D0
h	0068	ŋ	014B
i:	0069 02D0	œ	0153
j	006A	ɐ̯	0250 032F
k	006B	ɔ	0254
l	006C	ɔʏ	0254 028F
ɭ	006C 0329	ə	0259
m	006D	ɛ	025B
ɱ	006D 0329	ɛ:	025B 02D0
n	006E	ɪ	026A
ɳ	006E 0329	ɸ	0281
o:	006F 02D0	ʃ	0283
p	0070	ʊ	028A
pf	0070 0066	ɣ	028F
s	0073	ʈʂ	02A7
t	0074		

Hebrew Character Set

For Hebrew custom vocabularies, you can use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
-	002D	א	05DD
א	05D0	ב	05DE
ב	05D1	ג	05DF
ג	05D2	ד	05E0
ד	05D3	ה	05E1
ה	05D4	ו	05E2
ו	05D5	ז	05E3
ז	05D6	ח	05E4
ח	05D7	ט	05E5
ט	05D8	י	05E6
י	05D9	ך	05E7
ך	05DA	ס	05E8
ס	05DB	ע	05E9
ע	05DC	פ	05EA
		ק	
		ר	
		ש	
		ת	

You can use the following International Phonetic Alphabet characters in the `IPA` field of the vocabulary input file:

Character	Code	Character	Code
a	0061	p	0070
b	0062	s	0073
d	0064	t	0074
e	0065	u	0075
f	0066	v	0076
g	0067	w	0077
h	0068	z	007A
i	0069	ŋ	014B
j	006A	ʎ	0263
k	006B	ʃ	0283

Character	Code	Character	Code
l	006C	3	0292
m	006D	?	0294
n	006E	χ	03C7
o	006F		

Hindi Character Set

For Hindi custom vocabularies, you can use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
-	002D	थ	0925
.	002E	द	0926
ॢ	0901	ध	0927
.	0902	न	0928
:	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D
ई	0908	म	092E
उ	0909	य	092F
ऊ	090A	र	0930
ऋ	090B	ल	0932
ए	090F	व	0935
ऐ	0910	श	0936
औ	0911	ष	0937
ओ	0913	स	0938
औ	0914	ह	0939
क	0915	ट	093E
ख	0916	ठ	093F
ग	0917	ड	0940
घ	0918	ॢ	0941
ङ	0919	ॣ	0942

Character	Code	Character	Code
च	091A	८	0943
छ	091B	९	0945
ज	091C	॰	0947
झ	091D	ॱ	0948
ञ	091E	ॲ	0949
ट	091F	ॳ	094B
ठ	0920	ॴ	094C
ड	0921	ॵ	094D
ढ	0922	ॶ	095B
ण	0923	ॷ	095C
त	0924	ॸ	095D

Amazon Transcribe maps the following characters:

Character	Mapped to
न (0929)	न (0928)
र (0931)	र (0930)
क (0958)	क (0915)
ख (0959)	ख (0916)
ग (095A)	ग (0917)
फ (095E)	फ (092B)
य (095F)	य (092F)

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
a:	0097 0720	ŋ	0331
b	0098	ɖ	0598
b ^h	0098 0689	ɔ:	0596 0720
d	0100	ɖ ^h	0598 0689
d ^h	0100 0689	ə	0601
e:	0101 0720	ɛ:	0603 0720
f	0102	g	0609

Character	Code	Character	Code
i:	0105 0720	g ^h	0609 0689
j	0106	h	0614
k	0107	ɪ	0618
k ^h	0107 0688	ɲ	0626
l	0108	ŋ	0627
m	0109	r	0638
n	0110	ʂ	0642
o:	0111 0720	ʃ	0643
p	0112	t	0648
p ^h	0112 0688	t ^h	0648 0688
r	0114	ʊ	0650
s	0115	u	0651
t	0116	ɖʒ	0676
t ^h	0116 0688	ɖʒ ^h	0676 0689
u:	0117 0720	ʈʂ	0679
z	0122	ʈʂ ^h	0679 0688

Indonesian Character Set

For Indonesian custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can use the following International Phonetic Alphabet characters in the `IPA` field of your input file:

Character	Code	Character	Code
a	0061	r	0072
ai	0061 0069	s	0073
au	0061 0075	t	0074
b	0062	ʈʂ	0074 0283
d	0064	u	0075

Character	Code	Character	Code
d	0064	v	0076
e	0065	w	0077
f	0066	x	0078
h	0068	y	0079
i	0069	ŋ	014B
j	006A	ɔ	0254
k	006B	ə	0259
l	006C	ɛ	025B
m	006D	g	0261
n	006E	ɣ	0263
o	006F	ɪ	026A
ɔj	006F 0069 032F	ɲ	0272
p	0070	ʃ	0283
q	0071	ʊ	028A

Italian Character Set

For Italian custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
À	00C0	à	00E0
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB

Character	Code	Character	Code
ì	00CC	ì	00EC
Ò	00D2	ò	00F2
Ù	00D9	ù	00F9
Ü	00DC	ü	00FC

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
a	0061	ss	0073 0073
b	0062	t	0074
bb	0062 0062	tt	0074 0074
d	0064	u	0075
dd	0064 0064	v	0076
e	0065	vv	0076 0076
f	0066	w	0077
ff	0066 0066	z	007A
gg	0067 0067	ɔ	0254
i	0069	ɛ	025B
j	006A	g	0261
k	006B	ɲ	0272
kk	006B 006B	ɲɲ	0272 0272
l	006C	ʃ	0283
ll	006C 006C	ʃʃ	0283 0283
m	006D	ʌ	028E
mm	006D 006D	ʌʌ	028E 028E
n	006E	ɖ	02A3
nn	006E 006E	ɖɖ	02A3 02A3
o	006F	ɖɜ	02A4
p	0070	ɖɜɜ	02A4 02A4
pp	0070 0070	ts	02A6
r	0072	tsɬ	02A6 02A6
rr	0072 0072	tʃ	02A7

Character	Code	Character	Code
s	0073	ʃʒ	02A7 02A7

Japanese Character Set

For Japanese custom vocabularies, the `Phrase` and `DisplayAs` fields can use any of the characters listed in the following file on GitHub.

- [japanese-character-set.txt](#)

Amazon Transcribe supports Romaji characters in the `SoundsLike` field. You can use the following lower-case characters:

- a - k
- m - p
- r - w
- y - z

Represent long vowels by doubling the vowel:

Vowel	Representation
ā	aa
ē	ee
ī	ii
ō	oo
ū	uu

You can use the following International Phonetic Alphabet characters in the `IPA` field of your input file:

Character	Code	Character	Code
a	0061	p	0070
a:	0061 02D0	s	0073
b	0062	t	0074
d	0064	ts	0074 0073
dz	0064 007A	tʃ	0074 0255
dʒ	0064 0291	w	0077
e	0065	z	007A
e:	0065 02D0	ʒ	00E7
g	0067	ŋ	014B

Character	Code	Character	Code
h	0068	ɸ	0255
i	0069	ʊ	026F
i:	0069 02D0	ʊ:	026F 02D0
j	006A	ɳ	0274
k	006B	ϕ	0278
m	006D	ɾ	027E
n	006E	ʐ	0291
o	006F	ʔ	0294
o:	006F 02D0		

Korean Character Set

For Korean custom vocabularies, you can use any of the Hangul syllables in the `Phrase` and `SoundsLike` fields. For more information, see [Hangul Syllables](#) on Wikipedia.

You can use the following International Phonetic Alphabet characters in the `IPA` field of your input file:

Character	Code	Character	Code
a	00061	s#	0073 0348
e	00065	t	0074
h	00068	tɕ	0074 0255
i	00069	tɕʰ	0074 0255 02B0
je	006A 0065	tʰ	0074 02B0
jo	006A 006F	t#	0074 0348
ju	006A 0075	t#ɕ	0074 0348 0255
jɛ	006A 025B	u	0075
jʌ	006A 028C	wɛ	0077 0065
ja	006A 0061	wi	0077 0069
k	006B	wɛ	0077 025B
kʰ	006B 02B0	wʌ	0077 028C
k#	006B 0348	wɑ	0077 0061
l	006C	ø	00F8
m	006D	ɲ	0014B
n	006E	ɛ	0025B

Character	Code	Character	Code
o	006F	u	026F
p	0070	ui	006F 0069
p ^h	0070 02B0	r	027E
p#	0070 0348	ʌ	028C
s	0073		

Malay Character Set

For Malay custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can use the following International Phonetic Alphabet characters in the `IPA` field of your input file:

Character	Code	Character	Code
F	0046	r	0072
a	0061	s	0073
ai	0061 0069	t	0074
au	0061 0075	tʃ	0074 0283
b	0062	v	0076
d	0064	w	0077
dʒ	0064 0292	x	0078
e	0065	y	0079
h	0068	ŋ	014B
i	0069	ɔ	0254
j	006A	ə	0259
k	006B	ɛ	025B
l	006C	g	0261
m	006D	ɣ	0263
n	006E	ɪ	026A
o	006F	ɲ	0272

Character	Code	Character	Code
q̇	006F 0069 32F	ſ	0283
p	0070	ʊ	028A
q	0071	ui	028A 0069

Portuguese Character Set

For Portuguese custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
À	00C0	à	00E0
Á	00C1	á	00E1
Â	00C2	â	00E2
Ã	00C3	ã	00E3
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Í	00CD	í	00ED
Ñ	00D1	ñ	00F1
Ó	00D3	ó	00F3
Ô	00D4	ô	00F4
Õ	00D5	õ	00F5
Ö	00D6	ö	00F6
Ú	00DA	ú	00FA
Ü	00DC	ü	00FC

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
a	0061	v	0076
b	0062	w	0077
d	0064	w [~]	0077 0303
e	0065	z	007A
f	0066	õ	00F5
g	0067	ĩ	00129
i	0069	ũ	00169
j	006A	ě	0250 0303
k	006B	ɔ	0254
l	006C	ε	025B
m	006D	ɲ	0272
n	006E	ɾ	027E
o	006F	ʁ	0281
p	0070	ʃ	0283
s	0073	ʌ	028E
t	0074	ʒ	0292
tʃ	0074 0283	ɖʒ	02A4
u	0075	ě	1EBD

Russian Character Set

For Russian custom vocabularies, you can use the following characters in the Phrase and SoundsLike fields:

Character	Code	Character	Code
'	0027	п	043F
-	002D	р	0440
.	002E	с	0441
а	0430	т	0442
б	0431	у	0443
в	0432	ф	0444
г	0433	х	0445

Character	Code	Character	Code
д	0434	ц	0446
е	0435	ч	0447
ж	0436	ш	0448
з	0437	щ	0449
и	0438	ъ	044A
й	0439	ы	044B
к	043A	ь	044C
л	043B	э	044D
м	043C	ю	044E
н	043D	я	044F
о	043E	ё	0451

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
b	0062	t	0074
b ^j	0062 02B2	tʃ	0074 0283
d	0064	tʰ	0074 02B2
d ^j	0064 02B2	u	0075
f	0066	v	0076
f ^j	0066 02B2	v ^j	0076 02B2
g	0067	x	0078
g ^j	0067 02B2	x ^j	0078 02B2
i	0069	z	007A
j	006A	z ^j	007A 02B2
k	006B	æ	00E6
k ^j	006B 02B2	ə	0259
l	006C	ɛ	025B
l ^j	006C 02B2	ɪ	0268
m	006D	ʃ	0283
m ^j	006D 02B2	ʃ ^j	0283 02B2
n	006E	ʊ	028A

Character	Code	Character	Code
n ⁱ	006E 02B2	ʌ	028C
p	0070	ʒ	0292
p ⁱ	0070 02B2	'i	02C8 0069
r	0072	'o	02C8 006F
r ⁱ	0072 02B2	'v	02C8 0075
s	0073	'ɛ	02C8 025B
s ⁱ	0073 02B2	'i	02C8 0268
ts	0074 0073	'a	02C8 0061

Spanish Character Set

For Spanish custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
Á	00C1	á	00E1
É	00C9	é	00E9
Í	00CD	ë	00ED
Ó	00D3	ó	00F3
Ú	00DA	ú	00FA
Ñ	00D1	ñ	00F1
ü	00FC		

You can use the following International Phonetic Alphabet characters in the `IPA` field of your input file:

Character	Code	Character	Code
a	0061	r	0072
b	0062	s	0073

Character	Code	Character	Code
d	0064	t	0074
e	0065	u	0075
f	0066	v	0076
g	0067	w	0077
h	0068	x	0078
i	0069	z	007A
j	006A	ŋ	014B
k	006B	ɲ	0272
l	006C	r	027E
m	006D	ʃ	0283
n	006E	ʒ	029D
o	006F	ʈʂ	02A7
p	0070	θ	03B8

Tamil Character Set

For Tamil custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
அ	0B85	ர	0BB0
ஆ	0B86	ல	0BB2
இ	0B87	வ	0BB5
ஈ	0B88	ழ	0BB4
உ	0B89	ள	0BB3
ஊ	0B8A	ற	0BB1
எ	0B8E	ன	0BA9
ஏ	0B8F	ஜ	0B9C
ஐ	0B90	#	0BB6
ஒ	0B92	ஷ	0BB7
ஔ	0B93	ஸ	0BB8
ஓள	0B94	ஹ	0BB9
ஃ	0B83	.	0BCD

Character	Code	Character	Code
ஃ	0B95	ஈ	0BBE
ஊ	0B99	ஊ	0BBF
ஈ	0B9A	ஊ	0BC0
ஊ	0B9E	ஊ	0BC1
ஊ	0B9F	ஊ	0BC2
ஊ	0BA3	ஊ	0BC6
ஊ	0BA4	ஊ	0BC7
ஊ	0BA8	ஊ	0BC8
ஊ	0BAA	ஊ	0BCA
ஊ	0BAE	ஊ	0BCB
ஊ	0BAF	ஊ	0BCC

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
a	0061	v	0076
a:	0061 02D0	w	0077
b	0062	z	007A
d	0064	æ	00E6
dʒ	0064 0292	ð	00F0
e	0065	ŋ	014B
f	0066	ɑ	0251
g	0067	ɔ	0254
h	0068	ə	0259
i	0069	ɛ	025B
i:	0069 02D0	g	0261
j	006A	ɪ	026A
k	006B	l	026D
l	006C	ɹ	0272
m	006D	ŋ	0273
n	006E	ɹ	0279
ŋ	006E 032A	ɹ	0279

Character	Code	Character	Code
o	006F	౧	0279 0329
o:	006F 02D0	ర	027E
p	0070	స	0282
r	0072	ఙ	0283
s	0073	త	0288
t	0074	ు	028A
ṭ	0074 032A	ు	028B
ṭf	0074 0283	ఱ	028C
u	0075	౩	0292
u:	0075 02D0	ఠ	03B8

Telugu Character Set

For Telugu custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
-	002D	త	0C24
ఁ	0C01	థ	0C25
ం	0C02	ద	0C26
ః	0C03	ధ	0C27
అ	0C05	న	0C28
ఆ	0C06	ప	0C2A
ఇ	0C07	ళ	0C2B
ఈ	0C08	బ	0C2C
ఊ	0C09	భ	0C2D
ఋ	0C0A	మ	0C2E
ఋ	0C0B	య	0C2F
ౠ	0C0C	ర	0C30
ఎ	0C0E	ల	0C31
ఏ	0C0F	ల	0C32
ఐ	0C10	ళ	0C33
ఒ	0C12	న	0C35

Character	Code	Character	Code
ఓ	0C13	ఔ	0C36
ఔ	0C14	ప	0C37
క	0C15	ప	0C38
ఖ	0C16	భ	0C39
గ	0C17	ి	0C3E
ఘ	0C18	ి	0C3F
ఙ	0C19	ఱ	0C40
చ	0C1A	ృ	0C41
ఛ	0C1B	ౄ	0C42
జ	0C1C	౅	0C43
ఝ	0C1D	ె	0C44
ఞ	0C1E	ే	0C47
ట	0C1F	ై	0C48
ఠ	0C20	౉	0C4A
డ	0C21	ొ	0C4B
ఢ	0C22	ో	0C4C
ణ	0C23	ౌ	0C4D

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
ɖ	0064 032A	ð	00F0
ɗ	0064 032A 0324	ɲ	014B
dʒ	0064 0292	ɑ	0251
dʒ	0064 0292 0324	ɔ	0254
e	0065	ɔ̃	0256
e:	0065 02D0	ɔ̃	0256 0324
f	0066	ə	0259
h	0068	ɛ	025B
i	0069	g	0261
iz̥	0069 0290	g	0261 0324
j	006A	ɪ	026A

Character	Code	Character	Code
k	006B	l	026D
k ^h	006B 02B0	ɲ	0272
l	006C	ŋ	0273
m	006D	ɹ	0279
n	006E	ɻ	0279 0329
o	006F	ɿ	027D
o:	006F 02D0	ʂ	0282
p	0070	ʃ	0283
p ^h	0070 02B0	ʈ	0288
r	0072	ʈ ^h	0288 02B0
s	0073	ʊ	028A
t	0074	ʮ	028B
ṭ	0074 032A	ʮ	028C
ṭ ^h	0074 032A 02B0	ʼ	0292
t	0074	θ	03B8
tʃ ^h	0074 0283 02B0		

Turkish Character Set

For Turkish custom vocabularies, you can use the following characters in the `Phrase` and `SoundsLike` fields:

- a - z
- A - Z
- ' (apostrophe)
- - (hyphen)
- . (period)

You can also use the following Unicode characters in the `Phrase` and `SoundsLike` fields:

Character	Code	Character	Code
Ç	00C7	ö	00F6
Ö	00D6	û	00FB
Ü	00DC	ü	00FC
â	00E2	ğ	011E
ä	00E4	ğ	011F

Character	Code	Character	Code
ç	00E7	ï	0130
è	00E8	ı	0131
é	00E9	Ş	015E
ê	00EA	ş	015F
í	00ED	Š	0161
î	00EE	ž	017E
ó	00F3		

You can use the following International Phonetic Alphabet characters in the IPA field of your input file:

Character	Code	Character	Code
a	0061	u	0075
a:	0061 02D0	u:	0075 02D0
b	0062	v	0076
c	0063	w	0077
d	0064	y	0079
e	0065	y:	0079 02D0
e:	0065 02D0	z	007A
f	0066	ø	00F8
g	0067	ø:	00F8 02D0
h	0068	ŋ	014B
i	0069	ɟ	025F
i:	0069 02D0	ɣ	0263
j	006A	ɮ	026B
k	006B	ɹ	026F
l	006C	ɹ:	026F 02D0
m	006D	r	027E
n	006E	ʃ	0283
o	006F	ʒ	0292
o:	006F 02D0	ʔ	0294
p	0070	ɬ	02A4
s	0073	ɮ	02A7

Character	Code	Character	Code
t	0074		

Vocabulary Filtering

While transcribing a job, Amazon Transcribe can check the transcript for particular words, such as profane words, that you don't want in the transcript. During transcription Amazon Transcribe filters these words from the transcript. You can choose to remove words entirely or you can choose to replace the filtered words with placeholder text.

To omit or redact specific words from the transcript, create a collection of words that Amazon Transcribe filters from the transcript. You specify the list of words in a text file that you store in an S3 bucket or provide a list of words using the [CreateVocabularyFilter \(p. 136\)](#) operation. The following is an example of a list of words to filter from the transcript.

```
word1
word2
word3
...
wordn
```

Each line contains a single word and ends with a newline (\n) character. The words are not case sensitive, "word1" and "WORD1" are considered the same word.

To use a filter when you are transcribing an audio file, specify the name of the filter to use when you create the job with either the console or the API. The output of the transcription is modified to remove the filtered words. You can choose to mask the word, or you can completely remove the word from the transcript. If you choose to mask words Amazon Transcribe replaces the word with three asterisks ("***").

For example, consider the transcription of the sentence "I said, I don't like word1!" If you choose to mask words, the output is

```
I said, I don't like ***!
```

If you choose to remove words, the output is

```
I said, I don't like!
```

Amazon Transcribe does not filter words that appear inside other words. For example, the word "upword1" is not filtered, nor is "up-word1."

You create a filter by using the console or by using the Amazon Transcribe API. For both, you provide a list of words to remove from the transcription.

To use the API, call the [CreateVocabularyFilter \(p. 136\)](#) operation and either pass it the name of the filter and the S3 bucket location where the input file is stored, or pass it the name of the filter and a list of words to add. Use the [GetVocabularyFilter \(p. 151\)](#) to monitor the progress of creating the filter.

To use the console, choose **Vocabulary filtering** and then choose **Create vocabulary filter**. Give the filter a name and choose the source of words for the filter. When you use the console you can either store the text file for the filter in an S3 bucket or you can upload the file from your local computer. Choose **Create** to create the collection. You can monitor the creation progress of the collection using the vocabulary filters list.

You can manage your vocabulary filter using the console or the API. You can use both to list the vocabulary filters in your account, get details of individual filters, and delete filters from your account.

Automatic Content Redaction

Amazon Transcribe's automatic content redaction feature automatically redacts sensitive personally identifiable information (PII) from your transcription results. It replaces each identified instance of PII with a [PII] tag in the transcript. You can use this feature to protect privacy and comply with local laws and regulations. Automatic content redaction enables you to easily review and share transcripts to improve the customer service experience, coach agents, and discover new business opportunities while protecting sensitive personal information. You can use this feature for source audio in US English (en-US) with asynchronous API calls.

Personally Identifiable Information includes:

PII Entity	Definition
Bank Account Number	A number that uniquely identifies a bank account.
Bank Routing Number	A number that identifies the location of a bank account.
Credit Card Number or Debit Card Number	A value that uniquely defines a payment card issued by a bank.
Credit Card or Debit Card CVV Code	A 3 digit or 4 digit security code on each credit card.
Credit Card or Debit Card Expiration Date	The month and year a card expires.
Debit Card PIN or Credit Card PIN	A security code issued by a bank or credit union. This number is used for bank accounts and payment cards.
Email address	The unique identifier of an email box where messages are delivered.
US Mailing address	The U.S. mailing address of an individual.
Name	The first name and last name of a person.
US phone number	A 10-digit phone number within the United States.
Social Security Number	A 9 digit number or the last 4 digits of that number. Issued to U.S. citizens, permanent residents, and temporary residents with employment.

For each transcription job with automatic content redaction enabled, you generate either:

- Only the redacted transcript.
- Both the redacted transcript and the unredacted transcript.

Both redacted and unredacted transcripts are stored in the same output S3 bucket. Amazon Transcribe stores them in either a bucket you specify or in the default S3 bucket managed by the service.

To enable content redaction, use the console or the API. In the console, you enable **Automatic content redaction** in the **Content removal** section.

To enable content redaction using the API, complete the request parameters of the ContentRedaction object in the **StartTranscriptionJob** operation. See the request syntax for the [StartTranscriptionJob](#) (p. 163) action for more information. To see if content redaction has been enabled for a particular transcription job, use [GetTranscriptionJob](#) (p. 145). To see which jobs have content redaction enabled, use [ListTranscriptionJobs](#) (p. 154).

A redacted transcript has sensitive PII replaced with the [PII] tag, shown in the first truncated JSON output on this page. Because Amazon Transcribe has redacted this transcript, the `isRedacted` field of this JSON output is `true`. Each JSON output of a transcription job has a `results` section that contains the transcription results. Every word, number, punctuation mark, or redaction has a confidence value.

Transcription jobs using automatic content redaction generate two types of confidence values. The Automatic Speech Recognition (ASR) confidence indicates the items that have the type of pronunciation or punctuation is a specific utterance. In the transcript output below, the word Good has a confidence of 1.0. This confidence value indicates that Amazon Transcribe is 100 percent confident that the word uttered in this transcript is Good. The confidence value for a PII tag is the confidence that the speech it flagged for redaction is truly PII. In the transcript output below, the confidence of 0.9999 indicates that Amazon Transcribe is 99.99 percent confident that the entity it redacted in the transcript is PII.

The following is the redacted JSON output:

```
{
  "jobName": "job id",
  "accountId": "account id",
  "isRedacted": true,
  "results": {
    "transcripts": [
      {
        "transcript": "Good morning, everybody. My name is [PII], and today I feel like sharing a whole lot of personal information with you. Let's start with my Social Security number [PII]. My credit card number is [PII] and my C V V code is [PII]. I hope that Amazon Transcribe is doing a good job at redacting that personal information away. Let's check."
      }
    ],
    "items": [
      {
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Good"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
      {
        "start_time": "5.56",
        "end_time": "6.25",
        "alternatives": [
          {
            "content": "[PII]",
            "redactions": [
              {
                "confidence": "0.9999"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
        }
      ],
      "type": "pronunciation"
    },
    Items removed for brevity
  ],
},
"status": "COMPLETED"
}
```

If you generate an additional unredacted transcript, its JSON output looks similar to the output of a transcription job with content redaction disabled. The only difference is the additional `isRedacted` field being set to `false`.

The following is the unredacted JSON output:

```
{
  "jobName": "job id",
  "accountId": "account id",
  "isRedacted": false,
  "results": {
    "transcripts": [
      {
        "transcript": "Good morning, everybody. My name is Mike, and today I feel like sharing a whole lot of personal information with you. Let's start with my Social Security number 000000000. My credit card number 5555555555555555 is and my C V V code is 000. I hope that Amazon Transcribe is doing a good job at redacting that personal information away. Let's check."
      }
    ],
    "items": [
      {
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Good"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
      {
        "start_time": "5.56",
        "end_time": "6.25",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Mike"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
    ],
  },
  "status": "COMPLETED"
}
```

Amazon Transcribe throws an error message if you use automatic content redaction in an unsupported region. Similarly, you receive an error message if you use content redaction with an unsupported language.

Job Queuing

When you send transcription jobs to Amazon Transcribe, there is a limit to the total number of jobs that can run at one time. By default, there are 100 slots for jobs. When the limit is reached, you must wait until one or more jobs have finished and freed up a slot before you can send your next job.

To queue jobs so that they run as soon as a slot becomes available, you can use *job queuing*. Job queuing creates a queue on your behalf that contains your jobs. When a slot is available, Amazon Transcribe takes the next job from the queue and immediately starts processing it. To allow resources for new jobs to be submitted and processed, Amazon Transcribe uses at most 90 percent of your slots to process jobs in the queue.

You can turn on job queuing with the console, or you can set the `AllowDeferredExecution` field in the `JobExecutionSettings` parameter to `true` when you call the [StartTranscriptionJob \(p. 163\)](#) operation.

When you submit a job with job queuing turned on, one of the following things happens.

- If slots are available, the job is processed immediately.
- If no slots are available, the job is sent into a queue. When slots become available, jobs are removed from the queue in FIFO order (first in, first out).

You can see the progress of a queued job using the console or by using the [GetTranscriptionJob \(p. 145\)](#) operation. When a job is queued, the `Status` field of the `TranscriptionJob` object returned by the `StartTranscriptionJob` operation is set to `QUEUED`. The status changes to `IN_PROGRESS` when Amazon Transcribe starts processing the audio, and then changes to either `COMPLETED` or `FAILED` when processing is finished. You can use the `TranscriptionJobName` field with the `GetTranscriptionJob` operation to monitor the status of a job.

You can submit up to 10,000 jobs to the queue. If you exceed 10,000 jobs, you get a `LimitExceededConcurrentJobException` exception.

IAM Policies for Job Queuing

To use job queuing, you need to provide Amazon Transcribe with a data access role that permits access to your audio file to be transcribed. You can choose the data access role using the console, or you use the `DataAccessRoleArn` field of the `JobExecutionSettings` parameter of the `StartTranscriptionJob` operation to specify the role to use.

The role policies that you use depends on where you are storing your input files, where you are storing your output files, and whether or not you are encrypting the output with an AWS KMS customer master key (CMK). The IAM policies in this section are required for the role. The policies enable Amazon Transcribe to work on your behalf, allow access to the input and output locations for your jobs, and enable Amazon Transcribe to use a AWS KMS CMK to encrypt your transcriptions.

Trust Policy

The data access role that you use for transcription must have a trust policy that enables Amazon Transcribe to assume the role. Use the following trust policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
```

```
    "Principal": {
      "Service": [
        "transcribe.amazonaws.com",
      ],
    },
    "Action": "sts:AssumeRole"
  }
]
```

Input Bucket Policy

The following IAM policy gives the data access role permission to read files from your input bucket.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::input-bucket-name",
      "arn:aws:s3:::input-bucket-name/*"
    ]
  }
}
```

Output Bucket Policy

The following IAM policy gives the data access role permission to write files to your output bucket.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::output-bucket-name/*"
    ]
  }
}
```

AWS KMS Key Policy for Input Buckets

If you have encrypted your input files, the data access role needs permission to use the AWS KMS key to decrypt the files. The following policy provides that permission.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt"
    ],
    "Resource": [
```

```
        "arn:aws:kms::input-bucket-cmk-name"  
    ]  
  }  
}
```

AWS KMS Key Policy for Output Buckets

To encrypt your output transcriptions, the data access role needs permission to use the AWS KMS key. The following policy provides that permission.

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Action": [  
      "kms:GenerateDataKey*"  
    ],  
    "Resource": [  
      "arn:aws:kms::output-bucket-cmk-name"  
    ]  
  }  
}
```

Getting Started with Amazon Transcribe

To get started using Amazon Transcribe, set up an AWS account and create an AWS Identity and Access Management (IAM) user. To use the AWS Command Line Interface (AWS CLI), download and configure it.

Topics

- [Step 1: Set up an AWS Account and Create an Administrator User \(p. 52\)](#)
- [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 53\)](#)
- [Step 3: Getting Started Using the Console \(p. 54\)](#)
- [Step 4: Getting Started Using the API \(p. 58\)](#)
- [Step 5: Getting Started With Streaming Audio \(p. 62\)](#)

Step 1: Set up an AWS Account and Create an Administrator User

Before you use Amazon Transcribe for the first time, complete the following tasks:

1. [Sign up for AWS \(p. 52\)](#)
2. [Create an IAM User \(p. 52\)](#)

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including Amazon Transcribe. You are charged only for the services that you use.

With Amazon Transcribe, you pay only for the resources that you use. If you are a new AWS customer, you can get started with Amazon Transcribe for free. For more information, see [AWS Free Usage Tier](#).

If you already have an AWS account, skip to the next section.

To create an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Record your AWS account ID because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon Transcribe, require that you provide credentials when you access them. This allows the service to determine whether you have permissions to access the service's resources.

We strongly recommend that you access AWS using AWS Identity and Access Management (IAM), not the credentials for your AWS account. To use IAM to access AWS, create an IAM user, add the user to an IAM

group with administrative permissions, and then grant administrative permissions to the IAM user. You can then access AWS using a special URL and the IAM user's credentials.

The Getting Started exercises in this guide assume that you have a user with administrator privileges, `adminuser`.

To create an administrator user and sign in to the console

1. Create an administrator user called `adminuser` in your AWS account. For instructions, see [Creating Your First IAM User and Administrators Group](#) in the *IAM User Guide*.
2. Sign in to the AWS Management Console using a special URL. For more information, see [How Users Sign In to Your Account](#) in the *IAM User Guide*.

For more information about IAM, see the following:

- [AWS Identity and Access Management \(IAM\)](#)
- [Getting Started](#)
- [IAM User Guide](#)

Next Step

[Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 53\)](#)

Step 2: Set up the AWS Command Line Interface (AWS CLI)

You don't need the AWS CLI to perform the steps in the Getting Started exercises. However, some of the other exercises in this guide do require it. If you prefer, you can skip this step and set up the AWS CLI later.

To set up the AWS CLI

1. Download and configure the AWS CLI. For instructions, see the following topics in the *AWS Command Line Interface User Guide*:
 - [Getting Set Up with the AWS Command Line Interface](#)
 - [Configuring the AWS Command Line Interface](#)
2. In the AWS CLI `config` file, add a named profile for the administrator user:

```
[profile adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

You use this profile when executing the AWS CLI commands. For more information about named profiles, see [Named Profiles](#) in the *AWS Command Line Interface User Guide*. For a list of AWS Regions, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

3. Verify the setup by typing the following help command at the command prompt:

```
aws help
```

Next Step

[Step 3: Getting Started Using the Console \(p. 54\)](#)

Step 3: Getting Started Using the Console

The easiest way to get started with Amazon Transcribe is to submit a job using the console to transcribe an audio file. If you haven't reviewed the concepts and terminology in [How Amazon Transcribe Works \(p. 4\)](#), we recommend that you do that before proceeding.

Topics

- [Create a Transcription Job \(p. 54\)](#)
- [View a Transcription Job \(p. 55\)](#)

Create a Transcription Job

Use the Amazon Transcribe console to create a transcription job for your audio files.

1. Provide the following information:

- **Transcription job name**—A name for the job. The name must be unique within your AWS account.
- **Amazon S3 input URL**—The Amazon S3 location of your input audio file. The location must be in the same region as the endpoint that you are calling.
- **Language**—Choose the language of your input file.
- **Format**—The format of the audio file. For best results you should use a lossless format such as FLAC or WAV with PCM 16-bit encoding.
- **Media sampling rate (Hz)**—Optional. The bit sampling rate of the audio file. Amazon Transcribe accepts sample rates between 8000 Hz and 48000 Hz. For best results, you should use 8000 Hz for low-fidelity audio and 16000 for high-fidelity audio.

The following shows the **Create Transcription Job** filled out for a sample job.

Create transcription job

Input [Info](#)

Name

The name can be up to 200 characters long. Valid characters are a-z, A-Z, 0-9 and - (hyphen).

S3 input URL
Type or paste the URL of your input audio file in S3.

Valid formats for audio files are mp3, mp4, wav, and flac.

Language
Choose the language of the input audio.

English ▼

Format
Choose the format of your audio file.

mp3 ▼

Valid formats for the audio are mp3, mp4, wav and flac.

Audio sampling rate (Hz)
Type the sampling rate of the input audio file.

Must be an integer between 8000 and 48000

Apply custom vocabulary [Info](#)
A custom vocabulary improves the accuracy of recognizing words, phrases, and commands.

▼

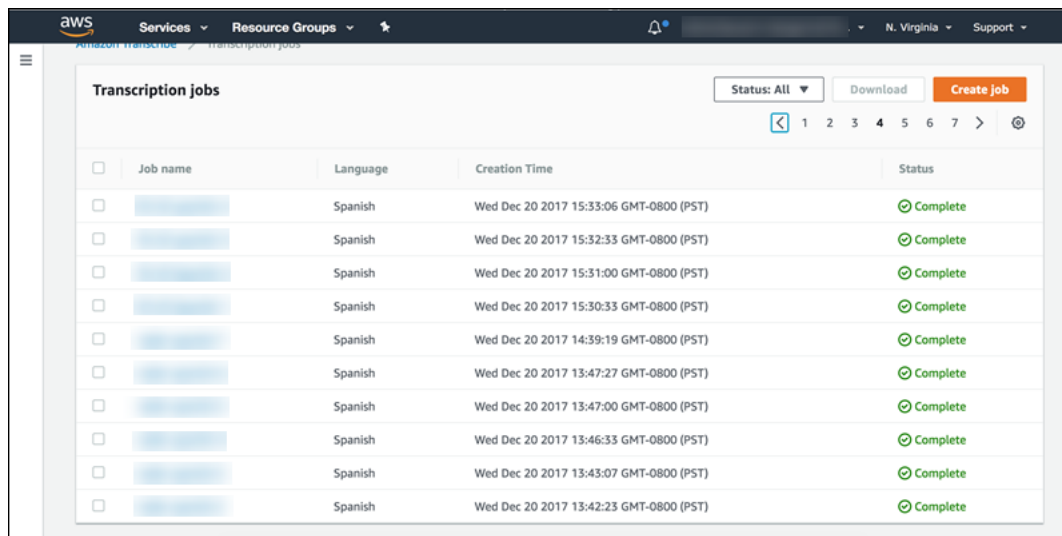
Speaker identification [Info](#)
Identifies speakers in the input audio file.

☒ Disabled
☐ Enabled

2. Choose **Create** to submit the job for processing.

View a Transcription Job

Completed transcription jobs are displayed in a list that contains a brief description of the job. The **Availability** column shows the remaining time that the job results will be kept on the server. Jobs are kept for 90 days and then deleted from the system.



The screenshot shows the Amazon Transcribe console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a search bar. Below this, the 'Transcription jobs' page is displayed. It features a table with columns for 'Job name', 'Language', 'Creation Time', and 'Status'. There are 10 jobs listed, all with a status of 'Complete'. The table is paginated, showing jobs 1 through 7. To the right of the table, there are buttons for 'Status: All', 'Download', and 'Create job'.

Job name	Language	Creation Time	Status
[Redacted]	Spanish	Wed Dec 20 2017 15:33:06 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:32:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:31:00 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 15:30:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 14:39:19 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:47:27 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:47:00 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:46:33 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:43:07 GMT-0800 (PST)	Complete
[Redacted]	Spanish	Wed Dec 20 2017 13:42:23 GMT-0800 (PST)	Complete

Choose a job in the list to see information about the job.

The information page about the transcription job has three sections. The **Detail** section provides details about the transcription job, including the name, information about when the job will be deleted from the server, and the input and output URLs. Use the output URL to download the output from your transcription job.

The **Output** section contains the transcription of the audio submitted to Amazon Transcribe. You can download the transcription by choosing the **Download transcription** button.



The screenshot shows the 'Details' page for a specific transcription job named 'ES-US-spanish-4'. The page has a breadcrumb trail: 'Amazon Transcribe > Transcription jobs > ES-US-spanish-4'. The 'Details' section includes a 'Download transcription' button. Below this, the 'Job name' is shown as '[Redacted]'. The 'Status' is 'Complete' with a green checkmark icon. The 'Expiration date' section states 'The transcription will be available for another 9 days'. The 'Input URL' is 'https://s3.amazonaws.com/[Redacted].wav'. The 'Output URL' is 'https://s3.amazonaws.com/[Redacted]/asrOutput.json'. There are external link icons next to the URLs.

Amazon Transcribe > Transcription jobs > ES-US-spanish-4

ES-US-spanish-4

Details Download transcription

Job name
[Redacted]

Status
Complete

Expiration date
The transcription will be available for another 9 days

Input URL
https://s3.amazonaws.com/[Redacted].wav

Output URL
https://s3.amazonaws.com/[Redacted]/asrOutput.json

The **Code samples** section contains the JSON input for the [StartTranscriptionJob](#) (p. 163) operation and the output from the [GetTranscriptionJob](#) (p. 145) operation.

Code Samples

Audio conversion

JSON Request

```
{
  "TranscriptionJobName": "██████████",
  "LanguageCode": "en-US",
  "MediaSampleRateHertz": 8000,
  "MediaFormat": "wav",
  "Media": {
    "MediaFileUri": "https://██████████.s3.amazonaws.com/answer2.wav"
  }
}
```

JSON Response

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "[REDACTED]",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "en-US",
    "MediaSampleRateHertz": 8000,
    "MediaFormat": "wav",
    "Media": {
      "MediaFileUri": "https://[REDACTED]
[REDACTED]/answer2.wav?X-Amz-Security-
Token=FQoDYXdzEOD%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDJVU%2Fub%2B5SJstTh3YXiK3AzA%2FWgl"
```

Next Step

Step 4: Getting Started Using the API (p. 58)

Step 4: Getting Started Using the API (p. 58)

Step 4: Getting Started Using the API

This section contains examples that demonstrate using the Amazon Transcribe API. You can use these samples to learn about the API or as building blocks in your own applications.

Topics

- [Getting Started \(AWS Command Line Interface\) \(p. 58\)](#)
- [Getting Started \(AWS SDK for Python \(Boto\)\) \(p. 60\)](#)

Getting Started (AWS Command Line Interface)

In the following exercise, you use the AWS Command Line Interface (AWS CLI) to transcribe speech into text. To complete this exercise, you need to:

- Have a text editor.
- Be familiar with the AWS CLI. For more information, see [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 53\)](#).
- Have a speech file in .WAV or .MP4 format that is stored in an S3 bucket that has the proper permissions. For more information about the permissions needed for Amazon Transcribe, see [Permissions Required for IAM User Roles \(p. 114\)](#).

To transcribe text, you have to provide the input parameters in a JSON file.

To transcribe text

1. Copy your input speech to an S3 bucket. The location must be in the same region as the endpoint that you are calling. This example assumes that the file is in an S3 bucket named `test-transcribe` and that the file name is `answer2.wav`.
2. Create a JSON file named `test-start-command.json` that contains the input parameters for the [StartTranscriptionJob \(p. 163\)](#) operation. Enter a unique name for your transcription job under "TranscriptionJobName".

```
{
  "TranscriptionJobName": "unique job name",
  "LanguageCode": "en-US",
  "MediaFormat": "wav",
  "Media": {
    "MediaFileUri": "https://S3 endpoint/test-transcribe/answer2.wav"
  }
}
```

3. In the AWS CLI, run the following command. The example is formatted for Unix, Linux, and macOS. For Windows, replace the backslash (\) Unix continuation character at the end of each line with a caret (^).

```
aws transcribe start-transcription-job \
  --region region \
  --cli-input-json file://test-start-command.json
```

Amazon Transcribe responds with the following:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "unique job name",
```

```
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "Media": {
      "MediaFileUri": "https://S3 endpoint/test-transcribe/answer2.wav"
    },
    "CreationTime": timestamp,
    "MediaFormat": "wav"
  }
}
```

To list transcription jobs

- Run the following command:

```
aws transcribe list-transcription-jobs \
  --region region \
  --status IN_PROGRESS
```

Amazon Transcribe responds with the following:

```
{
  "Status": "IN_PROGRESS",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "unique job name",
      "LanguageCode": "en-US",
      "CreationTime": timestamp,
      "TranscriptionJobStatus": "IN_PROGRESS"
    }
  ]
}
```

To get the results of a transcription job

- When the job has the status COMPLETED, get the results of the job. Type the following command:

```
aws transcribe get-transcription-job \
  --region region \
  --transcription-job-name "unique job name"
```

Amazon Transcribe responds with the following:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "unique job name",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "Media": {
      "MediaFileUri": "input URI"
    },
    "CreationTime": timestamp,
    "CompletionTime": timestamp,
    "Transcript": {
      "TranscriptFileUri": "output URI"
    }
  }
}
```

2. Use the output URI to get the transcribed text from the audio file. The following is the output from transcribing a short audio clip:

```
{
  "jobName": "job ID",
  "accountId": "account ID",
  "results": {
    "transcripts": [
      {
        "transcript": " that's no answer"
      }
    ],
    "items": [
      {
        "start_time": "0.180",
        "end_time": "0.470",
        "alternatives": [
          {
            "confidence": 0.84,
            "content": "that's"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "0.470",
        "end_time": "0.710",
        "alternatives": [
          {
            "confidence": 0.99,
            "content": "no"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "0.710",
        "end_time": "1.080",
        "alternatives": [
          {
            "confidence": 0.874,
            "content": "answer"
          }
        ],
        "type": "pronunciation"
      }
    ]
  },
  "status": "COMPLETED"
}
```

Getting Started (AWS SDK for Python (Boto))

In this exercise you create script that uses the SDK for Python to transcribe speech into text. To complete this exercise, you need to:

- Install the AWS CLI. For more information, see [Step 2: Set up the AWS Command Line Interface \(AWS CLI\) \(p. 53\)](#). This installs the AWS SDK for Python (Boto).
- Have a speech file in .WAV or .MP4 format that is stored in an S3 bucket that has the proper permissions. For more information about the permissions needed for Amazon Transcribe, see [Permissions Required for IAM User Roles \(p. 114\)](#). The location must be in the same region as the

endpoint that you are calling. This example assumes that the file is in an Amazon S3 bucket named `test-transcribe` and that the file name is `answer2.wav`.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "job name"
job_uri = "https://S3 endpoint/test-transcribe/answer2.wav"
transcribe.start_transcription_job(
    TranscriptionJobName=job_name,
    Media={'MediaFileUri': job_uri},
    MediaFormat='wav',
    LanguageCode='en-US'
)
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName=job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

When the transcription job is complete, the result links to an Amazon S3 presigned URL that contains the transcription in JSON format:

```
{
  "jobName": "job ID",
  "accountId": "account ID",
  "results": {
    "transcripts": [
      {
        "transcript": " that's no answer"
      }
    ],
    "items": [
      {
        "start_time": "0.180",
        "end_time": "0.470",
        "alternatives": [
          {
            "confidence": 0.84,
            "word": "that's"
          }
        ]
      },
      {
        "start_time": "0.470",
        "end_time": "0.710",
        "alternatives": [
          {
            "confidence": 0.99,
            "word": "no"
          }
        ]
      },
      {
        "start_time": "0.710",
        "end_time": "1.080",
        "alternatives": [
          {
            "confidence": 0.87,
            "word": "answer"
          }
        ]
      }
    ]
  }
}
```

```
    }  
  ]  
}  
],  
  "status": "COMPLETED"  
}
```

Step 5: Getting Started With Streaming Audio

The following example is a Java program that transcribes streaming audio. The input comes from your computer's microphone or a file upload and the output is presented on your computer's standard output.

To run this example, you need the following:

- You must use the [AWS SDK for Java 2.x](#)
- Clients must use Java 1.8 to be compatible with the AWS SDK for Java 2.x.

```
/**  
 * COPYRIGHT:  
 * <p>  
 * Copyright 2018-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
 * <p>  
 * Licensed under the Apache License, Version 2.0 (the "License").  
 * You may not use this file except in compliance with the License.  
 * A copy of the License is located at  
 * <p>  
 * http://www.apache.org/licenses/LICENSE-2.0  
 * <p>  
 * or in the "license" file accompanying this file. This file is distributed  
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
 * express or implied. See the License for the specific language governing  
 * permissions and limitations under the License.  
 */  
package com.amazonaws.transcribestreaming;  
  
import org.reactivestreams.Publisher;  
import org.reactivestreams.Subscriber;  
import org.reactivestreams.Subscription;  
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;  
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;  
import software.amazon.awssdk.core.SdkBytes;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.transcribestreaming.TranscribeStreamingAsyncClient;  
import software.amazon.awssdk.services.transcribestreaming.model.*;  
  
import javax.sound.sampled.*;  
import java.io.*;  
import java.net.URISyntaxException;  
import java.nio.ByteBuffer;  
import java.util.List;  
import java.util.concurrent.CompletableFuture;  
import java.util.concurrent.ExecutionException;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
import java.util.concurrent.atomic.AtomicLong;  
  
public class TranscribeStreamingDemoApp {  
    private static final Region REGION = Region.US_EAST_1;
```

```
private static Subscription currentSubscription;

private static TranscribeStreamingAsyncClient client;

public static void main(String args[]) throws URISyntaxException, ExecutionException,
InterruptedException, LineUnavailableException {

    client = TranscribeStreamingAsyncClient.builder()
        .credentialsProvider(getCredentials())
        .region(REGION)
        .build();

    CompletableFuture<Void> result =
client.startStreamTranscription(getRequest(16_000),
    new AudioStreamPublisher(getStreamFromMic()),
    getResponseHandler());

    result.get();
    client.close();
}

private static InputStream getStreamFromMic() throws LineUnavailableException {

    // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
    int sampleRate = 16000;
    AudioFormat format = new AudioFormat(sampleRate, 16, 1, true, false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

    if (!AudioSystem.isLineSupported(info)) {
        System.out.println("Line not supported");
        System.exit(0);
    }

    TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
    line.open(format);
    line.start();

    InputStream audioStream = new AudioInputStream(line);
    return audioStream;
}

private static AwsCredentialsProvider getCredentials() {
    return DefaultCredentialsProvider.create();
}

private static StartStreamTranscriptionRequest getRequest(Integer mediaSampleRateHertz)
{
    return StartStreamTranscriptionRequest.builder()
        .languageCode(LanguageCode.EN_US.toString())
        .mediaEncoding(MediaEncoding.PCM)
        .mediaSampleRateHertz(mediaSampleRateHertz)
        .build();
}

private static StartStreamTranscriptionResponseHandler getResponseHandler() {
    return StartStreamTranscriptionResponseHandler.builder()
        .onResponse(r -> {
            System.out.println("Received Initial response");
        })
        .onError(e -> {
            System.out.println(e.getMessage());
            StringWriter sw = new StringWriter();
            e.printStackTrace(new PrintWriter(sw));
            System.out.println("Error Occurred: " + sw.toString());
        })
        .onComplete(() -> {
```

```
        System.out.println("=== All records stream successfully ===");
    })
    .subscriber(event -> {
        List<Result> results = ((TranscriptEvent)
event).transcript().results();
        if (results.size() > 0) {
            if (!results.get(0).alternatives().get(0).transcript().isEmpty()) {
System.out.println(results.get(0).alternatives().get(0).transcript());
            }
        }
    })
    .build();
}

private InputStream getStreamFromFile(String audioFileName) {
    try {
        File inputFile = new
File(getClass().getClassLoader().getResource(audioFileName).getFile());
        InputStream audioStream = new FileInputStream(inputFile);
        return audioStream;
    } catch (FileNotFoundException e) {
        throw new RuntimeException(e);
    }
}

private static class AudioStreamPublisher implements Publisher<AudioStream> {
    private final InputStream inputStream;

    private AudioStreamPublisher(InputStream inputStream) {
        this.inputStream = inputStream;
    }

    @Override
    public void subscribe(Subscriber<? super AudioStream> s) {

        if (this.currentSubscription == null) {
            this.currentSubscription = new SubscriptionImpl(s, inputStream);
        } else {
            this.currentSubscription.cancel();
            this.currentSubscription = new SubscriptionImpl(s, inputStream);
        }
        s.onSubscribe(currentSubscription);
    }
}

public static class SubscriptionImpl implements Subscription {
    private static final int CHUNK_SIZE_IN_BYTES = 1024 * 1;
    private final Subscriber<? super AudioStream> subscriber;
    private final InputStream inputStream;
    private ExecutorService executor = Executors.newFixedThreadPool(1);
    private AtomicLong demand = new AtomicLong(0);

    private SubscriptionImpl(Subscriber<? super AudioStream> s, InputStream
inputStream) {
        this.subscriber = s;
        this.inputStream = inputStream;
    }

    @Override
    public void request(long n) {
        if (n <= 0) {
            subscriber.onError(new IllegalArgumentException("Demand must be
positive"));
        }
    }
}
```

```
demand.getAndAdd(n);

executor.submit(() -> {
    try {
        do {
            ByteBuffer audioBuffer = getNextEvent();
            if (audioBuffer.remaining() > 0) {
                AudioEvent audioEvent = audioEventFromBuffer(audioBuffer);
                subscriber.onNext(audioEvent);
            } else {
                subscriber.onComplete();
                break;
            }
        } while (demand.decrementAndGet() > 0);
    } catch (Exception e) {
        subscriber.onError(e);
    }
});
}

@Override
public void cancel() {
    executor.shutdown();
}

private ByteBuffer getNextEvent() {
    ByteBuffer audioBuffer = null;
    byte[] audioBytes = new byte[CHUNK_SIZE_IN_BYTES];

    int len = 0;
    try {
        len = inputStream.read(audioBytes);

        if (len <= 0) {
            audioBuffer = ByteBuffer.allocate(0);
        } else {
            audioBuffer = ByteBuffer.wrap(audioBytes, 0, len);
        }
    } catch (IOException e) {
        throw new UncheckedIOException(e);
    }

    return audioBuffer;
}

private AudioEvent audioEventFromBuffer(ByteBuffer bb) {
    return AudioEvent.builder()
        .audioChunk(SdkBytes.fromByteBuffer(bb))
        .build();
}
}
```

Streaming Transcription

Amazon Transcribe streaming transcription enables you to send an audio stream and receive a stream of text in real time. The API makes it easy for developers to add real-time speech-to-text capability to their applications.

You can use streaming transcription in the following languages:

- 8 KHz and 16 KHz
- • US English (en-US)
 - US Spanish (es-US)
- 8 KHz only
- • Australian English (en-AU)
 - British English (en-GB)
 - French (fr-FR)
 - Canadian French (fr-CA)

Amazon Transcribe streaming transcription can be used for a variety of purposes. For example:

- Streaming transcriptions can generate real-time subtitles for live broadcast media.
- Lawyers can make real-time annotations on top of streaming transcriptions during courtroom depositions.
- Video game chat can be transcribed in real time so that hosts can moderate content or run real-time analysis.
- Streaming transcriptions can provide assistance to the hearing impaired.

Streaming transcription does not support channel identification or speaker identification. Use the [StartTranscriptionJob \(p. 163\)](#) operation if you need these features.

If you are using HTTP/2, we provide an HTTP/2 streaming client that handles retrying the connection when there are transient problems on the network. You can use this client as a starting point for your own applications. To use Amazon Transcribe streaming with the WebSocket protocol, you can create your own client.

Streaming transcription takes a stream of your audio data and transcribes it in real time. The transcription is returned to your application in a stream of transcription events.

Amazon Transcribe breaks your incoming audio stream based on natural speech segments, such as a change in speaker or a pause in the audio. The transcription is returned progressively to your application, with each response containing more transcribed speech until the entire segment is transcribed.

In the following example, each line is a partial result transcription output of an audio segment being streamed:

```
the amazon is the largest
the amazon is the largest
the amazon is the largest
the amazon is the largest rainforest
the amazon is the largest rainforest
the amazon is the largest rainforest
the amazon is the largest rainforest on the
the amazon is the largest rainforest on the
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet
```

```

the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet
the amazon is the largest rainforest on the planet covering over
the amazon is the largest rainforest on the planet covering over
the amazon is the largest rainforest on the planet covering over two million

```

Each [Result](#) object in the response contains a field called `IsPartial` that indicates whether the response is a partial response containing the transcription results so far or if it is a complete transcription of the audio segment.

Each [Result](#) object also contains the start time and end time of the term from the audio stream so that you can, for example, synchronize the transcription with the video.

The following example is a partial transcription response.

```

{
  "TranscriptResultStream": {
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
              {
                "Items": [
                  {
                    "Content": "the",
                    "EndTime": 0.3799375,
                    "StartTime": 0.0299375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "amazon",
                    "EndTime": 0.5899375,
                    "StartTime": 0.3899375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "is",
                    "EndTime": 0.7899375,
                    "StartTime": 0.5999375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "the",
                    "EndTime": 0.9199375,
                    "StartTime": 0.7999375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "largest",
                    "EndTime": 1.0199375,
                    "StartTime": 0.9299375,
                    "Type": "pronunciation"
                  }
                ],
                "Transcript": "the amazon is the largest"
              }
            ],
            "EndTime": 1.02,
            "IsPartial": true,
            "ResultId": "2db76dc8-d728-11e8-9f8b-f2801f1b9fd1",
            "StartTime": 0.0199375
          }
        ]
      }
    }
  }
}

```

```
}
  }
}
}
```

Topics

- [Event Stream Encoding \(p. 68\)](#)
- [Using Amazon Transcribe Streaming with WebSockets \(p. 69\)](#)
- [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#)

Event Stream Encoding

Event stream encoding provides bidirectional communication using messages between a client and a server. Data frames sent to the Amazon Transcribe streaming service are encoded in this format. The response from Amazon Transcribe also uses this encoding.

Each message consists of two sections: the prelude and the data. The prelude consists of:

1. The total byte length of the message
2. The combined byte length of all of the headers

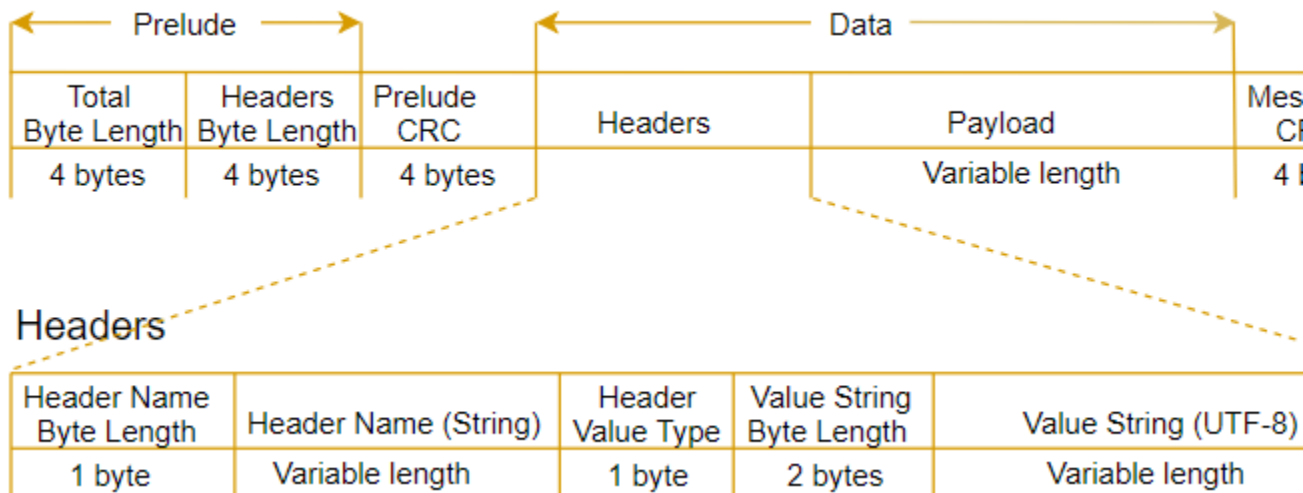
The data section consists of:

1. The headers
2. A payload

Each section ends with a 4-byte big-endian integer CRC checksum. The message CRC checksum is for both the prelude section and the data section. Amazon Transcribe uses CRC32 (often referred to as GZIP CRC32) to calculate both CRCs. For more information about CRC32, see [GZIP file format specification version 4.3](#).

Total message overhead, including the prelude and both checksums, is 16 bytes.

The following diagram shows the components that make up a message and a header. There are multiple headers per message.



Each message contains the following components:

- **Prelude:** Always a fixed size of 8 bytes, two fields of 4 bytes each.
 - *First 4 bytes:* The total byte-length. This is the big-endian integer byte-length of the entire message, including the 4-byte length field itself.
 - *Second 4 bytes:* The headers byte-length. This is the big-endian integer byte-length of the headers portion of the message, excluding the headers length field itself.
- **Prelude CRC:** The 4-byte CRC checksum for the prelude portion of the message, excluding the CRC itself. The prelude has a separate CRC from the message CRC to ensure that Amazon Transcribe can detect corrupted byte-length information immediately without causing errors such as buffer overruns.
- **Headers:** Metadata annotating the message, such as the message type, content type, and so on. Messages have multiple headers. Headers are key-value pairs where the key is a UTF-8 string. Headers can appear in any order in the headers portion of the message and any given header can appear only once. For the required header types, see the following sections.
- **Payload:** The audio content to be transcribed.
- **Message CRC:** The 4-byte CRC checksum from the start of the message to the start of the checksum. That is, everything in the message except the CRC itself.

Each header contains the following components. There are multiple headers per frame.

- **Header name byte-length:** The byte-length of the header name.
- **Header name:** The name of the header indicating the header type. For valid values, see the following frame descriptions.
- **Header value type:** An enumeration indicating the header value type.
- **Value string byte length:** The byte-length of the header value string.
- **Header value:** The value of the header string. Valid values for this field depend on the type of header. For valid values, see the following frame descriptions.

Using Amazon Transcribe Streaming with WebSockets

When you use the [WebSocket protocol](#) to stream audio, Amazon Transcribe transcribes the stream in real time. You encode the audio with event stream encoding, Amazon Transcribe responds with a JSON structure that is also encoded using event stream encoding. For more information, see [Event Stream Encoding \(p. 68\)](#). You can use the information in this section to create applications using the WebSocket library of your choice.

Topics

- [Adding a Policy for WebSocket Requests to Your IAM Role \(p. 69\)](#)
- [Creating a Pre-Signed URL \(p. 70\)](#)
- [Handling the WebSocket Upgrade Response \(p. 73\)](#)
- [Making a WebSocket Streaming Request \(p. 74\)](#)
- [Handling a WebSocket Streaming Response \(p. 75\)](#)
- [Handling WebSocket Streaming Errors \(p. 75\)](#)

Adding a Policy for WebSocket Requests to Your IAM Role

To use the WebSocket protocol to call Amazon Transcribe, you need to attach the following policy to the AWS Identity and Access Management (IAM) role that makes the request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "transcribestreaming",
      "Effect": "Allow",
      "Action": "transcribe:StartStreamTranscriptionWebSocket",
      "Resource": "*"
    }
  ]
}
```

Creating a Pre-Signed URL

Construct a URL for your WebSocket request that contains the information needed to set up communication between your application and Amazon Transcribe. WebSocket streaming uses the Amazon Signature Version 4 process for signing requests. Signing the request helps to verify the identity of the requester and to protect your audio data in transit. It also protects against potential replay attacks. For more information about Signature Version 4, see [Signing AWS API Requests](#) in the *Amazon Web Services General Reference*.

The URL has the following format. Line breaks have been added for readability.

```
GET wss://transcribestreaming.region.amazonaws.com:8443/stream-transcription-websocket
?language-code=languageCode
  &X-Amz-Algorithm=AWS4-HMAC-SHA256
  &X-Amz-Credential=Signature Version 4 credential scope
  &X-Amz-Date=date
  &X-Amz-Expires=time in seconds until expiration
  &X-Amz-Security-Token=security-token
  &X-Amz-Signature=Signature Version 4 signature
  &X-Amz-SignedHeaders=host
  &media-encoding=mediaEncoding
  &sample-rate=mediaSampleRateHertz
  &session-id=sessionId
  &vocabulary-name=vocabularyName
```

Use the following values for the URL parameters:

- **language-code** – The language code for the input audio. Valid values are `en-GB`, `en-US`, `es-US`, `fr-CA`, and `fr-FR`.
- **media-encoding** – The encoding used for the input audio. The only valid value is `pcm`.
- **sample-rate** – The sample rate of the input audio in Hertz. We suggest that you use 8000 Hz for low-quality audio and 16000 Hz for high-quality audio. The sample rate must match the sample rate in the audio file.
- **sessionId** – Optional. An identifier for the transcription session. If you don't provide a session ID, Amazon Transcribe generates one for you and returns it in the response.
- **vocabulary-name** – Optional. The name of the vocabulary to use when processing the transcription job, if any.

The remaining parameters are Signature Version 4 parameters:

- **X-Amz-Algorithm** – The algorithm you're using in the signing process. The only valid value is `AWS4-HMAC-SHA256`.
- **X-Amz-Credential** – A string separated by slashes ("/") that is formed by concatenating your access key ID and your credential scope components. Credential scope includes the date in YYYYMMDD format, the AWS Region, the service name, and a special termination string (`aws4_request`).

- **X-Amz-Date** – The date and time that the signature was created. Generate the date and time by following the instructions in [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.
- **X-Amz-Expires** – The length of time in seconds until the credentials expire. The maximum value is 300 seconds (5 minutes).
- **X-Amz-Security-Token** – Optional. A Signature Version 4 token for temporary credentials. If you specify this parameter, include it in the canonical request. For more information, see [Requesting Temporary Security Credentials](#) in the *AWS Identity and Access Management User Guide*.
- **X-Amz-Signature** – The Signature Version 4 signature that you generated for the request.
- **X-Amz-SignedHeaders** – The headers that are signed when creating the signature for the request. The only valid value is `host`.

To construct the URL for the request and create the Signature Version 4 signature, use the following steps. The examples are in pseudocode.

Task 1: Create a Canonical Request

Create a string that includes information from your request in a standardized format. This ensures that when AWS receives the request, it can calculate the same signature that you calculate in Task 3. For more information, see [Create a Canonical Request for Signature Version 4](#) in the *Amazon Web Services General Reference*.

1. Define variables for the request in your application.

```
# HTTP verb
method = "GET"
# Service name
service = "transcribe"
# AWS Region
region = "AWS Region"
# Amazon Transcribe streaming endpoint
endpoint = "wss://transcribestreaming.region.amazonaws.com:8443"
# Host
host = "transcribestreaming.region.amazonaws.com:8443"
# Date and time of request
amz-date = "YYMMDD'T'HHMMSS'Z'"
# Date without time for credential scope
datestamp = "YYMMDD"
```

2. Create a canonical URI. The canonical URI is the part of the URI between the domain and the query string.

```
canonical_uri = "/stream-transcription-websocket"
```

3. Create the canonical headers and signed headers. Note the trailing "\n" in the canonical headers.

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. Match the algorithm to the hashing algorithm. You must use SHA-256.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. Create the credential scope, which scopes the derived key to the date, Region, and service to which the request is made.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. Create the canonical query string. Query string values must be URL-encoded and sorted by name.

```
canonical_querystring = "X-Amz-Algorithm=" + algorithm  
canonical_querystring += "&X-Amz-Credentials=" + access_key + "/" + credential_scope  
canonical_querystring += "&X-Amz-Date=" + amz_date  
canonical_querystring += "&X-Amz-Expires=300"  
canonical_querystring += "&X-Amz-Security-Token=" + token  
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers  
canonical_querystring += "&language-code=en-US&media-encoding=pcm&sample-rate=16000"
```

7. Create a hash of the payload. For a GET request, the payload is an empty string.

```
payload_hash = HashSHA256("").Encode("utf-8").HexDigest()
```

8. Combine all of the elements to create the canonical request.

```
canonical_request = method + '\n'  
+ canonical_uri + '\n'  
+ canonical_querystring + '\n'  
+ canonical_headers + '\n'  
+ signed_headers + '\n'  
+ payload_hash
```

Task 2: Create the String to Sign

The string to sign contains meta information about your request. You use the string to sign in the next step when you calculate the request signature. For more information, see [Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

- Create the string.

```
string_to_sign=algorithm + "\n"  
+ amz_date + "\n"  
+ credential_scope + "\n"  
+ HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

Task 3: Calculate the Signature

You derive a signing key from your AWS secret access key. For a greater degree of protection, the derived key is specific to the date, service, and AWS Region. You use the derived key to sign the request. For more information, see [Calculate the Signature for AWS Signature Version 4](#) in the *Amazon Web Services General Reference*.

The code assumes that you have implemented the `GetSignatureKey` function to derive a signing key. For more information and example functions, see [Examples of How to Derive a Signing Key for Signature Version 4](#) in the *Amazon Web Services General Reference*.

The function `HMAC(key, data)` represents an HMAC-SHA256 function that returns the results in binary format.

- Create the signing key and sign the string to sign.

```
#Create the signing key
signing_key = GetSignatureKey(secret_key, timestamp, region, service)

# Sign the string_to_sign using the signing key
signature = HMAC.new(signing_key, (string_to_sign).Encode("utf-8"), Sha256()).hexdigest
```

Task 4: Add Signing Information to the Request and Create the Request URL

After you calculate the signature, add it to the query string. For more information, see [Add the Signature to the Request](#) in the *Amazon Web Services General Reference*.

1. Add the authentication information to the query string.

```
canonical_querystring += "&X-Amz-Signature=" + signature
```

2. Create the URL for the request.

```
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

You use the request URL with your WebSocket library to make the request to the Amazon Transcribe service.

Including WebSocket Request Headers

The request to Amazon Transcribe must include the following headers. Typically these headers are managed by your WebSocket client library.

```
Host: transcribestreaming.region.amazonaws.com:8443
Connection: Upgrade
Upgrade: websocket
Origin: request source
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: random key
```

Use the following values for the headers:

- **Connection** – Always Upgrade.
- **Upgrade** – Always websocket.
- **Origin** – The URI of the WebSocket client.
- **Sec-WebSocket-Version** – The version of the WebSocket protocol to use.
- **Sec-WebSocket-Key** – A base-64 encoded randomly generated string that identifies the request.

Handling the WebSocket Upgrade Response

When Amazon Transcribe receives your WebSocket request, it responds with a WebSocket upgrade response. Typically your WebSocket library manages this response and sets up a socket for communications with Amazon Transcribe.

The following is the response from Amazon Transcribe. Line breaks have been added to the `websocket-location` header for readability.

```
HTTP/1.1 101 Web Socket Protocol Handshake
```

```
Connection: upgrade
Upgrade: websocket
websocket-origin: wss://transcribestreaming.region.amazonaws.com:8443
websocket-location: transcribestreaming.region.amazonaws.com:8443/stream-transcription-
websocket?
                                X-Amz-Algorithm=AWS4-HMAC-SHA256
                                &X-Amz-Credential=AKIDEXAMPLE%2F20190117%2Fregion%2Ftranscribe
%2Faws4_request
                                &X-Amz-Date=date and time
                                &X-Amz-Expires=expiration length
                                &X-Amz-SignedHeaders=host
                                &language-code=language code
                                &media-encoding=media encoding
                                &sample-rate=media sample rate
                                &X-Amz-Signature=signature
x-amzn-RequestId: RequestId
x-amzn-SessionId: SessionId
Strict-Transport-Security: max-age=31536000
sec-websocket-accept: token
```

The response has the following values:

- **Connection** – Always Upgrade.
- **Upgrade** – Always websocket.
- **websocket-origin** – The URI of the WebSocket server that responded to the request.
- **websocket-location** – The contents of the request URI that was sent to the server. For a description of the contents, see [Creating a Pre-Signed URL \(p. 70\)](#).
- **x-amzn-RequestId** – An identifier for the request.
- **x-amzn-SessionId** – An identifier for a transcription session.
- **Strict-Transport-Security** – A header that informs browsers to access the endpoint using only HTTPS.
- **sec-websocket-accept** – The hash of the Sec-WebSocket-Key header sent in the request.

Making a WebSocket Streaming Request

After the WebSocket connection is established, the client can start sending a sequence of audio frames. Each frame contains one data frame that is encoded in event stream encoding. For more information, see [Event Stream Encoding \(p. 68\)](#).

Each data frame contains three headers combined with a chunk of raw audio bytes. The following table lists and describes the headers.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/ octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

To end the audio data stream, send an empty audio chunk in an event-stream-encoded message.

Handling a WebSocket Streaming Response

The response contains event-stream-encoded raw bytes in the payload. It contains the standard prelude and the following headers.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

When you decode the binary response, you end up with a JSON structure with the results of the transcription. For an example of the JSON response, see [Streaming Transcription \(p. 66\)](#).

Handling WebSocket Streaming Errors

If an exception occurs while processing your request, Amazon Transcribe responds with a terminal WebSocket frame containing an event-stream-encoded response. The response has the headers described in the following table, and the body of the response contains a descriptive error message. After sending the exception response, Amazon Transcribe sends a close frame.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
15	:exception-type	7	varies	varies, see below
13	:message-type	7	9	exception

The exception-type header contains one of the following values.

- **BadRequestException** – There was a client error when the stream was created, or an error occurred while streaming data. Make sure that your client is ready to accept data and try your request again.
- **InternalFailureException** – Amazon Transcribe had a problem during the handshake with the client. Try your request again.
- **LimitExceededException** – The client exceeded the concurrent stream limit. For more information, see [Amazon Transcribe Limits](#) in the *Amazon Web Services General Reference*. Reduce the number of streams that you are transcribing.
- **UnrecognizedClientException** – The WebSocket upgrade request was signed with an incorrect access key or secret key. Make sure that you are correctly creating the access key and try your request again.

In addition, Amazon Transcribe can return any of the common service errors. For a list, see [Common Errors](#).

Using Amazon Transcribe Streaming With HTTP/2

Amazon Transcribe uses a format called *event stream encoding* for streaming transcription. This format encoded binary data with header information that describes the contents of each event. For more information, see [Event Stream Encoding](#) (p. 68). You can use this information for applications that call the Amazon Transcribe endpoint without using the Amazon Transcribe SDK.

When Amazon Transcribe uses the [HTTP/2 protocol](#) for streaming transcriptions, the key components for a streaming request are:

- A header frame. This contains the HTTP/2 headers for the request, and a signature in the `authorization` header that Amazon Transcribe uses as a seed signature to sign the following data frames.
- One or more message frames in event stream encoding. The frame contains metadata and the raw audio bytes.
- An end frame. This is a signed message in event stream encoding with an empty body.

Streaming Request

To make a streaming request, you use the [StartStreamTranscription](#) (p. 175) operation.

Header Frame

The header frame is the authorization frame for the streaming transcription. Amazon Transcribe uses the value of the `authorization` header as the seed for generating a chain of authorization headers for the data frames in the request.

Required Headers

The header frame of a request to Amazon Transcribe requires the following HTTP/2 headers:

```
POST /stream-transcription HTTP/2.0
host: transcribestreaming.region.amazonaws.com
authorization: Generated value
content-type: application/vnd.amazon.eventstream
x-amz-target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-EVENTS
x-amz-date: Date
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: pcm
x-amzn-transcribe-sample-rate: Sample rate
transfer-encoding: chunked
```

In the request, use the following values for the `host`, `authorization`, and `x-amz-date` headers:

- **host:** Use the AWS Region where you are calling Amazon Transcribe. For a list of valid regions, see [AWS Regions and Endpoints](#) in the *Amazon Web Services General Reference*.
- **authorization:** The Signature Version 4 signature for the request. For more information about creating a signature, see [Signing AWS Requests with Signature Version 4](#) in the *Amazon Web Services General Reference*.
- **x-amz-date:** Generate a date and time for the request following the instructions in [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

For more information about the headers specific to Amazon Transcribe, see the [StartStreamTranscription \(p. 175\)](#) operation.

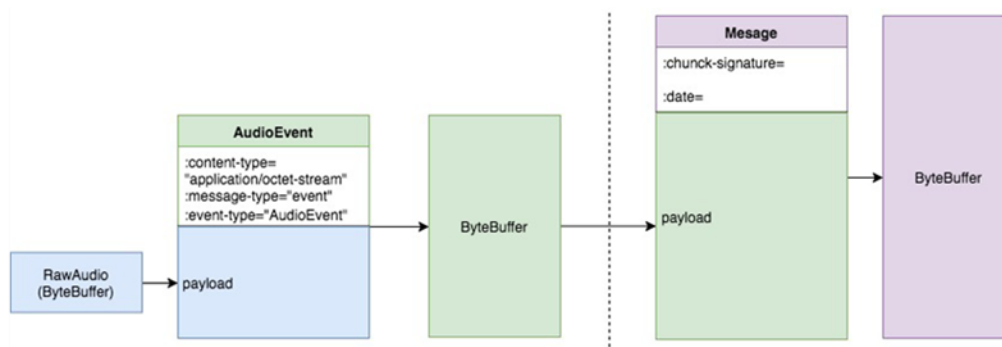
Data Frames

Each request contains one or more data frames. The data frames use event stream encoding. The encoding supports bidirectional data transmission between a client and a server.

There are two steps to creating a data frame:

1. Combine the raw audio data with metadata to create the payload of the request.
2. Combine the payload with a signature to form the event message that is sent to Amazon Transcribe.

The following diagram shows how this works.



Create the Audio Event

To create the message to send to Amazon Transcribe, create the audio event. Combine the headers described in the following table with a chunk of audio bytes into an event-encoded message.

Header Name Byte Length	Header Name (string)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

To create the payload for the event message, use a buffer in raw-byte format.

Create the Message

Create a data frame using the audio event payload to send to Amazon Transcribe. The data frame contains event-encoding headers that include the current date and a signature for the audio chunk and the audio event. To indicate to Amazon Transcribe that the audio stream is complete, send an empty data frame that contains only the date and signature.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value
16	:chunk-signature	6	varies	Generated signature
5	:date	8	8	Timestamp

To create the signature for the data frame, first create a string to sign, and then calculate the signature for the event. Construct the string to sign as follows:

```
String stringToSign =
    "AWS4-HMAC-SHA256-PAYLOAD" +
    "\n" +
    DATE +
    "\n" +
    KEYPATH +
    "\n" +
    Hex(priorSignature) +
    "\n" +
    HexHash(nonSignatureHeaders) +
    "\n" +
    HexHash(payload);
```

- **DATE:** The current date and time in Universal Time Coordinated (UTC) and using the [ISO 8601 format](#). Don't include milliseconds in the date. For example, 20190127T223754Z is 22:37:54 on 1/27/2019.
- **KEYPATH:** The signature scope in the format `date/region/service/aws4_request`. For example, 20190127/us-east-1/transcribe/aws4_request.
- **priorSignature:** The signature for the previous frame. For the first data frame, use the signature of the header frame.
- **nonSignatureHeaders:** The `DATE` header encoded as a string.
- **payload:** The byte buffer containing the audio event data.
- **Hex:** A function that encodes its input into a hexadecimal representation.
- **HexHash:** A function that first creates a SHA-256 hash of its input and then uses the `Hex` function to encode the hash.

After you have constructed the string to sign, sign it using the key that you derived for Signature Version 4, as follows. For details, see [Examples of How to Derive a Signing Key for Signature Version 4](#) in the *Amazon Web Services General Reference*.

```
String signature = HMACSHA256(derivedSigningKey, stringToSign);
```

- **HMACSHA256:** A function that creates a signature using the SHA-256 hash function.
- **derivedSigningKey:** The Signature Version 4 signing key.
- **stringToSign:** The string that you calculated for the data frame.

After you have calculated the signature for the data frame, construct a byte buffer containing the date, the signature, and the audio event payload. Send the byte array to Amazon Transcribe for transcription.

End Frame

To indicate that the audio stream is complete, send an end frame to Amazon Transcribe. The *end frame* is a data frame with an empty payload. You construct the end frame the same way that you construct a data frame.

Streaming Response

Responses from Amazon Transcribe are also sent using event stream encoding. Use this information to decode a response from the [StartStreamTranscription \(p. 175\)](#) operation.

Transcription Response

A transcription response is event stream encoded. It contains the standard prelude and the following headers:

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

For details, see [Event Stream Encoding \(p. 68\)](#).

When the response is decoded, it contains the following information:

```
:content-type: "application/json"
:event-type: "TranscriptEvent"
:message-type: "event"
```

JSON transcription information

For an example of the JSON structure returned by Amazon Transcribe, see [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#).

Exception Response

If there is an error in processing your transcription stream, Amazon Transcribe sends an exception response. The response is event stream encoded. For details, see [Event Stream Encoding \(p. 68\)](#).

The response contains the standard prelude and the following headers:

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	19	BadRequestException
13	:message-type	7	9	exception

When the exception response is decoded, it contains the following information:

```
:content-type: "application/json"
:event-type: "BadRequestException"
:message-type: "exception"
```

Exception message

Example Request and Response

The following is an end-to-end example of a streaming transcription request. In this example, binary data is represented as base64-encoded strings. In an actual response, the data are raw bytes.

Step 1 - Start the Session With Amazon Transcribe

To start the session, send an HTTP/2 request to Amazon Transcribe:

```
POST /stream-transcription HTTP/2.0
host: transcribestreaming.region.amazonaws.com
authorization: Generated value
content-type: application/vnd.amazon.eventstream
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-EVENTS
x-amz-date: Date
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: pcm
x-amzn-transcribe-sample-rate: Sample rate
transfer-encoding: chunked
```

Step 2 - Send Authentication Information to Amazon Transcribe

Amazon Transcribe sends the following response:

```
HTTP/2.0 200
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-sample-rate: Sample rate
x-amzn-request-id: 8a08df7d-5998-48bf-a303-484355b4ab4e
x-amzn-transcribe-session-id: b4526fcf-5eee-4361-8192-d1cb9e9d6887
x-amzn-transcribe-media-encoding: pcm
x-amzn-RequestId: 8a08df7d-5998-48bf-a303-484355b4ab4e
content-type: application/vnd.amazon.eventstream
```

Step 3 - Create an Audio Event

Create an audio event containing the audio data to send. For details, see [Event Stream Encoding \(p. 68\)](#). The binary data in this request is base64-encoded. In an actual request, the data is raw bytes.

```
:content-type: "application/octet-stream"
:event-type: "AudioEvent"
:message-type: "event"

UklGRjzxPQBXQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVTPQAAAAAAAAAAAAAAAAAD//wIA/f8EAA==
```

Step 4 - Create an Audio Event Message

Create an audio message that contains the audio data to send to Amazon Transcribe. For details, see [Event Stream Encoding \(p. 68\)](#). The audio event data in this example is base64-encoded. In an actual request, the data is raw bytes.

```
:date: 2019-01-29T01:56:17.291Z
:chunk-signature: signature
```

```
AAAA0gAAAIKvORFcTtcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0LOmV2ZW50LXR5cGUHAApBdWRpb0V2ZW50TptZXXNzYWdlLXR5cGUHAAVldmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFteilqc29uLTEuMVJJRky88T0AV0FWRWZtdCAQAAAAQABAIA+AAAAfQAAAgAQAGRhdGFU8D0AAAAA  
AAAAAAAAAAAA//8CAP3/BAC7QLFf
```

Step 5 - Use the Response from Amazon Transcribe

Amazon Transcribe creates a stream of transcription events that it sends to your application. The events are sent in raw-byte format. In this example, the bytes are base64-encoded.

The response from Amazon Transcribe is:

```
AAAAUwAAAEPlRHpYBTpkYXRlCAAAAWiXUkMLEdpjaHVuay1zaWduYXR1cmUGACct6Zy+uymwEK2SrLp/zVBI  
5eGn83jdBwCaRUBJA+eaDafqjI=
```

To see the transcription results, decode the raw bytes using event-stream encoding:

```
:event-type: "TranscriptEvent"  
:content-type: "application/json"  
:message-type: "event"  
  
{ "Transcript": { "Results": [ results ] }}
```

For an example of the JSON structure returned by Amazon Transcribe, see [Event Stream Encoding \(p. 68\)](#).

Step 6 - End the Transcription Stream

Finally, send an empty audio event to Amazon Transcribe to end the transcription stream. Create the audio event exactly like any other, except with an empty payload. Sign the event and include the signature in the `:chunk-signature` header, as follows:

```
:date: 2019-01-29T01:56:17.291Z  
:chunk-signature: signature
```

HTTP/2 Streaming Retry Client

You can use the following code in your applications to handle retry logic for Amazon Transcribe streaming transcription. The code provides tolerance for intermittent failures in the connection to Amazon Transcribe. There are two parts of the client: an interface that you implement for your application, and the retry client itself.

Streaming Retry Client Code

This code implements a streaming retry client. It manages the connection to Amazon Transcribe and retries sending data when there are errors on the connection. For example, if there is a transient error on the network, this client resends the request that failed.

The retry client has two properties that control the behavior of the client. You can set:

- The maximum number of times that the client should attempt before failing. Reduce this value to make your application stop retrying sooner when there are network issues. The default is 10.
- The time in milliseconds that the client should wait between retries. Longer times raise the risk of losing data, shorter times raise the risk of your application being throttled. The default is 100 milliseconds.

The following is the client. You can copy this code to your application or use it as a starting point for your own client.

```
/**
 * COPYRIGHT:
 * <p>
 * Copyright 2018-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
package com.amazonaws.transcribestreaming.retryclient;

import org.reactivestreams.Publisher;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.signer.EventStreamAws4Signer;
import software.amazon.awssdk.core.client.config.SdkAdvancedClientOption;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.transcribestreaming.TranscribeStreamingAsyncClient;
import software.amazon.awssdk.services.transcribestreaming.model.AudioStream;
import software.amazon.awssdk.services.transcribestreaming.model.BadRequestException;
import
    software.amazon.awssdk.services.transcribestreaming.model.StartStreamTranscriptionRequest;
import
    software.amazon.awssdk.services.transcribestreaming.model.StartStreamTranscriptionResponseHandler;

import java.net.URI;
import java.net.URISyntaxException;
import java.util.Arrays;
import java.util.List;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

/**
 * Build a client wrapper around the Amazon Transcribe client to retry
 * on an exception that can be retried.
 */
public class TranscribeStreamingRetryClient {

    private static final int DEFAULT_MAX_RETRIES = 10;
    private static final int DEFAULT_MAX_SLEEP_TIME_MILLS = 100;
    private static final Logger log =
        LoggerFactory.getLogger(TranscribeStreamingRetryClient.class);
    private final TranscribeStreamingAsyncClient client;
    List<Class<?>> nonRetriableExceptions = Arrays.asList(BadRequestException.class);
    private int maxRetries = DEFAULT_MAX_RETRIES;
    private int sleepTime = DEFAULT_MAX_SLEEP_TIME_MILLS;
```

```
/**
 * Create a TranscribeStreamingRetryClient with given credential and configuration
 *
 * @param creds    Creds to use for transcription
 * @param endpoint Endpoint to use for transcription
 * @param region   Region to use for transcriptions
 * @throws URISyntaxException if the endpoint is not a URI
 */
public TranscribeStreamingRetryClient(AwsCredentialsProvider creds,
                                     String endpoint, Region region) throws
URISyntaxException {
    this(TranscribeStreamingAsyncClient.builder()
        .overrideConfiguration(
            c -> c.putAdvancedOption(
                SdkAdvancedClientOption.SIGNER,
                EventStreamAws4Signer.create()))
        .credentialsProvider(creds)
        .endpointOverride(new URI(endpoint))
        .region(region)
        .build());
}

/**
 * Initiate TranscribeStreamingRetryClient with TranscribeStreamingAsyncClient
 *
 * @param client TranscribeStreamingAsyncClient
 */
public TranscribeStreamingRetryClient(TranscribeStreamingAsyncClient client) {
    this.client = client;
}

/**
 * Get Max retries
 *
 * @return Max retries
 */
public int getMaxRetries() {
    return maxRetries;
}

/**
 * Set Max retries
 *
 * @param maxRetries Max retries
 */
public void setMaxRetries(int maxRetries) {
    this.maxRetries = maxRetries;
}

/**
 * Get sleep time
 *
 * @return sleep time between retries
 */
public int getSleepTime() {
    return sleepTime;
}

/**
 * Set sleep time between retries
 *
 * @param sleepTime sleep time
 */
public void setSleepTime(int sleepTime) {
```

```
        this.sleepTime = sleepTime;
    }

    /**
     * Initiate a Stream Transcription with retry.
     *
     * @param request      StartStreamTranscriptionRequest to use to start transcription
     * @param publisher    The source audio stream as Publisher
     * @param responseHandler StreamTranscriptionBehavior object that defines how the
     response needs to be handled.
     * @return CompletableFuture future to handle stream response.
     */

    public CompletableFuture<Void> startStreamTranscription(final
    StartStreamTranscriptionRequest request,
                                                                final Publisher<AudioStream>
    publisher,
                                                                final
    StreamTranscriptionBehavior responseHandler) {

        CompletableFuture<Void> finalFuture = new CompletableFuture<>();

        recursiveStartStream(rebuildRequestWithSession(request), publisher,
    responseHandler, finalFuture, 0);

        return finalFuture;
    }

    /**
     * Recursively call startStreamTranscription() to be called till the request is
     completed or till we run out of retries.
     *
     * @param request      StartStreamTranscriptionRequest
     * @param publisher    The source audio stream as Publisher
     * @param responseHandler StreamTranscriptionBehavior object that defines how the
     response needs to be handled.
     * @param finalFuture  final future to finish on completing the chained futures.
     * @param retryAttempt Current attempt number
     */
    private void recursiveStartStream(final StartStreamTranscriptionRequest request,
                                                                final Publisher<AudioStream> publisher,
                                                                final StreamTranscriptionBehavior responseHandler,
                                                                final CompletableFuture<Void> finalFuture,
                                                                final int retryAttempt) {
        CompletableFuture<Void> result = client.startStreamTranscription(request,
    publisher,
        getResponseHandler(responseHandler));
        result.whenComplete((r, e) -> {
            if (e != null) {
                log.debug("Error occurred:", e);

                if (retryAttempt <= maxRetries && isExceptionRetriable(e)) {
                    log.debug("Retriable error occurred and will be retried.");
                    log.debug("Sleeping for sometime before retrying...");
                    try {
                        Thread.sleep(sleepTime);
                    } catch (InterruptedException e1) {
                        log.debug("Unable to sleep. Failed with exception: ", e);
                        e1.printStackTrace();
                    }
                    log.debug("Making retry attempt: " + (retryAttempt + 1));
                    recursiveStartStream(request, publisher, responseHandler, finalFuture,
    retryAttempt + 1);
                } else {
                    log.error("Encountered unretriable exception or ran out of retries. ");
                    responseHandler.onError(e);
                }
            }
        });
    }
}
```



```

        finalFuture.completeExceptionally(e);
    }
    } else {
        responseHandler.onComplete();
        finalFuture.complete(null);
    }
    });
}

private StartStreamTranscriptionRequest
rebuildRequestWithSession(StartStreamTranscriptionRequest request) {
    return StartStreamTranscriptionRequest.builder()
        .languageCode(request.languageCode())
        .mediaEncoding(request.mediaEncoding())
        .mediaSampleRateHertz(request.mediaSampleRateHertz())
        .sessionId(UUID.randomUUID().toString())
        .build();
}

/**
 * StartStreamTranscriptionResponseHandler implements subscriber of transcript stream
 * Output is printed to standard output
 */
private StartStreamTranscriptionResponseHandler getResponseHandler(
    StreamTranscriptionBehavior transcriptionBehavior) {
    final StartStreamTranscriptionResponseHandler build =
    StartStreamTranscriptionResponseHandler.builder()
        .onResponse(r -> {
            transcriptionBehavior.onResponse(r);
        })
        .onError(e -> {
            //Do nothing here. Don't close any streams that shouldn't be cleaned up
yet.

        })
        .onComplete(() -> {
            //Do nothing here. Don't close any streams that shouldn't be cleaned up
yet.

        })
        .subscriber(event -> transcriptionBehavior.onStream(event))
        .build();
    return build;
}

/**
 * Check if the exception can be retried.
 *
 * @param e Exception that occurred
 * @return True if the exception is retrieable
 */
private boolean isExceptionRetriable(Throwable e) {
    e.printStackTrace();

    return nonRetriableExceptions.contains(e.getClass());
}

public void close() {
    this.client.close();
}
}

```

Streaming Retry Client Interface Code

This interface is similar to the response handler used in the getting started example. It implements the same event handlers. Implement this interface to use the streaming retry client.

```
/**
 * COPYRIGHT:
 * <p>
 * Copyright 2018-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * <p>
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 * <p>
 * http://www.apache.org/licenses/LICENSE-2.0
 * <p>
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 */
package com.amazonaws.transcribestreaming.retryclient;

import
    software.amazon.awssdk.services.transcribestreaming.model.StartStreamTranscriptionResponse;
import software.amazon.awssdk.services.transcribestreaming.model.TranscriptResultStream;

/**
 * Defines how a stream response should be handled.
 * You should build a class implementing this interface to define the behavior.
 */
public interface StreamTranscriptionBehavior {
    /**
     * Defines how to respond when encountering an error on the stream transcription.
     *
     * @param e The exception
     */
    void onError(Throwable e);

    /**
     * Defines how to respond to the Transcript result stream.
     *
     * @param e The TranscriptResultStream event
     */
    void onStream(TranscriptResultStream e);

    /**
     * Defines what to do on initiating a stream connection with the service.
     *
     * @param r StartStreamTranscriptionResponse
     */
    void onResponse(StartStreamTranscriptionResponse r);

    /**
     * Defines what to do on stream completion
     */
    void onComplete();
}
```

The following is an example implementation of the `StreamTranscriptionBehavior` interface. You can use this implementation or use it as a starting point for your own implementation.

```
package com.amazonaws.transcribestreaming.retryclient;

import com.amazonaws.transcribestreaming.retryclient.StreamTranscriptionBehavior;
import software.amazon.awssdk.services.transcribestreaming.model.Result;
import software.amazon.awssdk.services.transcribestreaming.model.StartStreamTranscriptionResponse;
import software.amazon.awssdk.services.transcribestreaming.model.TranscriptEvent;
import software.amazon.awssdk.services.transcribestreaming.model.TranscriptResultStream;

import java.util.List;

/**
 * Implementation of StreamTranscriptionBehavior to define how a stream response should be
 * handled.
 *
 * * COPYRIGHT:
 *
 * * Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * * Licensed under the Apache License, Version 2.0 (the "License").
 * * You may not use this file except in compliance with the License.
 * * A copy of the License is located at
 *
 * * http://www.apache.org/licenses/LICENSE-2.0
 *
 * * or in the "license" file accompanying this file. This file is distributed
 * * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * * express or implied. See the License for the specific language governing
 * * permissions and limitations under the License.
 */
public class StreamTranscriptionBehaviorImpl implements StreamTranscriptionBehavior {

    @Override
    public void onError(Throwable e) {
        System.out.println("=== Failure Encountered ===");
        e.printStackTrace();
    }

    @Override
    public void onStream(TranscriptResultStream e) {
        // EventResultStream has other fields related to the the timestamp of the
        transcripts in it.
        // Please refer to the javadoc of TranscriptResultStream for more details
        List<Result> results = ((TranscriptEvent) e).transcript().results();
        if (results.size() > 0) {
            if (results.get(0).alternatives().size() > 0)
                if (!results.get(0).alternatives().get(0).transcript().isEmpty()) {
                    System.out.println(results.get(0).alternatives().get(0).transcript());
                }
        }
    }

    @Override
    public void onResponse(StartStreamTranscriptionResponse r) {
        System.out.println(String.format("=== Received Initial response. Request Id: %s",
            r.requestId()));
    }

    @Override
```

```
public void onComplete() {  
    System.out.println("=== All records stream successfully ===");  
}
```

Next step

[Using the HTTP/2 Retry Client \(p. 88\)](#)

Using the HTTP/2 Retry Client

The following is a sample application that uses the retry client to transcribe audio from either a file or your microphone. You can use this application to test the client, or you can use it as a starting point for your own applications.

To run the sample, do the following:

- Copy the retry client to your workspace. See [Streaming Retry Client Code \(p. 81\)](#).
- Copy the retry client interface to your workspace. Implement the interface, or you can use the sample implementation. See [Streaming Retry Client Interface Code \(p. 86\)](#).
- Copy the sample application to your workspace. Build and run the application.

```
/**  
 * COPYRIGHT:  
 * <p>  
 * Copyright 2018-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.  
 * <p>  
 * Licensed under the Apache License, Version 2.0 (the "License").  
 * You may not use this file except in compliance with the License.  
 * A copy of the License is located at  
 * <p>  
 * http://www.apache.org/licenses/LICENSE-2.0  
 * <p>  
 * or in the "license" file accompanying this file. This file is distributed  
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either  
 * express or implied. See the License for the specific language governing  
 * permissions and limitations under the License.  
 */  
package com.amazonaws.transcribestreaming.retryclient;  
  
import com.amazonaws.transcribestreaming.TranscribeStreamingDemoApp;  
import org.reactivestreams.Publisher;  
import org.reactivestreams.Subscriber;  
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;  
import software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;  
import software.amazon.awssdk.regions.Region;  
import software.amazon.awssdk.services.transcribestreaming.model.AudioStream;  
import software.amazon.awssdk.services.transcribestreaming.model.LanguageCode;  
import  
    software.amazon.awssdk.services.transcribestreaming.model.StartStreamTranscriptionRequest;  
  
import javax.sound.sampled.LineUnavailableException;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.InputStream;  
import java.net.URISyntaxException;  
import java.util.concurrent.CompletableFuture;  
import java.util.concurrent.ExecutionException;  
  
import static com.amazonaws.transcribestreaming.TranscribeStreamingDemoApp.getCredentials;
```

```
public class StreamingRetryApp {
    private static final String endpoint = "endpoint";
    private static final Region region = Region.US_EAST_1;
    private static final int sample_rate = 28800;
    private static final String encoding = " ";
    private static final String language = LanguageCode.EN_US.toString();

    public static void main(String args[]) throws URISyntaxException, ExecutionException,
    InterruptedException, LineUnavailableException, FileNotFoundException {
        /**
         * Create Transcribe streaming retry client using AWS credentials.
         */

        TranscribeStreamingRetryClient client = new
        TranscribeStreamingRetryClient(EnvironmentVariableCredentialsProvider.create(), endpoint,
        region);

        StartStreamTranscriptionRequest request = StartStreamTranscriptionRequest.builder()
            .languageCode(language)
            .mediaEncoding(encoding)
            .mediaSampleRateHertz(sample_rate)
            .build();

        /**
         * Start real-time speech recognition. The Transcribe streaming java client uses
         the Reactive-streams
         * interface. For reference on Reactive-streams:
         * https://github.com/reactive-streams/reactive-streams-jvm
         */
        CompletableFuture<Void> result = client.startStreamTranscription(
            /**
             * Request parameters. Refer to API documentation for details.
             */
            request,
            /**
             * Provide an input audio stream.
             * For input from a microphone, use getStreamFromMic().
             * For input from a file, use getStreamFromFile().
             */
            new AudioStreamPublisher(
                new FileInputStream(new File("FileName"))),
            /**
             * Object that defines the behavior on how to handle the stream
             */
            new StreamTranscriptionBehaviorImpl());

        /**
         * Synchronous wait for stream to close, and close client connection
         */
        result.get();
        client.close();
    }

    private static class AudioStreamPublisher implements Publisher<AudioStream> {
        private final InputStream inputStream;

        private AudioStreamPublisher(InputStream inputStream) {
            this.inputStream = inputStream;
        }

        @Override
        public void subscribe(Subscriber<? super AudioStream> s) {
            if (s.currentSubscription == null) {
                this.currentSubscription = new
                TranscribeStreamingDemoApp.SubscriptionImpl(s, inputStream);
            } else {

```

```
        this.currentSubscription.cancel();
        this.currentSubscription = new
TranscribeStreamingDemoApp.SubscriptionImpl(s, inputStream);
    }
    s.onSubscribe(currentSubscription);
}
}
```

Amazon Transcribe Medical

Contents

- [What Is Amazon Transcribe Medical? \(p. 91\)](#)
- [How Amazon Transcribe Medical Works \(p. 92\)](#)
- [Getting Started with Amazon Transcribe Medical \(p. 94\)](#)
- [Establish a Bi-Directional Connection Using the WebSocket Protocol \(p. 95\)](#)

What Is Amazon Transcribe Medical?

Amazon Transcribe Medical is an automatic speech recognition (ASR) service that you can use to add medical speech-to-text capabilities to an application. Amazon Transcribe Medical is available as a real-time application programming interface (API) that you call to connect using the WebSocket protocol, which helps secure the bi-directional connection. When you pass an audio stream to Amazon Transcribe Medical, the service returns a stream of JSON objects containing the results of the transcription.

You can use Amazon Transcribe Medical to transcribe medical-related speech found in a variety of use cases. For example, physician-dictated notes, drug safety monitoring, telemedicine, or even physician-patient conversations.

You can use [Amazon Comprehend Medical](#) for further analysis of your transcript.

Important Notice

Amazon Transcribe Medical is not a substitute for professional medical advice, diagnosis, or treatment. Identify the right confidence threshold for your use case, and use high confidence thresholds in situations that require high accuracy. For certain use cases, results should be reviewed and verified by appropriately trained human reviewers. All operations of Amazon Transcribe Medical should only be used in patient care scenarios after review for accuracy and sound medical judgement by trained medical professionals.

Transcribing Streaming Audio

You can use Amazon Transcribe Medical to transcribe medical speech in real time. When you send Amazon Transcribe Medical a stream of audio, it returns a stream of JSON objects containing the transcription of the audio.

For more information about processing audio streams, see [Establish a Bi-Directional Connection Using the WebSocket Protocol \(p. 95\)](#).

Supported Specialties

Amazon Transcribe Medical currently supports transcription for primary care. The following specialties are included:

- Family Medicine
- Internal Medicine
- Obstetrics and Gynecology (OB-GYN)
- Pediatrics

Are You a First-time User of Amazon Transcribe Medical?

If you are a first-time user, we recommend that you read the following topic to get an introduction to Amazon Transcribe Medical: [How Amazon Transcribe Medical Works \(p. 92\)](#).

How Amazon Transcribe Medical Works

To transcribe streaming audio to text, Amazon Transcribe Medical uses a WebSocket protocol. For more information, see [Establish a Bi-Directional Connection Using the WebSocket Protocol \(p. 95\)](#).

Amazon Transcribe Medical can transcribe 16,000 Hz (16 kHz) audio in real time. Amazon Transcribe Medical currently supports US English (en-US).

Amazon Transcribe Medical Features

- Transcribe 16 kHz audio in real time
- Transcribe up to 4 hours of audio streams
- Transcribe audio in US English
- Word-level timestamp in transcripts
- Word-level confidence in transcripts
- [Number normalization \(p. 92\)](#)
- Punctuation and true casing in transcripts
- Supports both dictation and conversation speech types

Transcribing Streaming Audio

Streaming transcription takes a stream of your audio data and transcribes it in real time. It uses a bidirectional WebSocket connection so that the results of the transcription are returned to your application while you send more audio to Amazon Transcribe Medical. You can also use it when you have an audio file that you want to process as it is transcribed.

You can use streaming transcription in US English (en-US).

For more information, see [Establish a Bi-Directional Connection Using the WebSocket Protocol \(p. 95\)](#).

Transcribing Numbers

When you are transcribing, numbers are transcribed as digits instead of words. For streaming transcription, numbers are transcribed as digits. For example, the spoken number "one thousand two hundred forty-two" is transcribed as "1242."

Numbers are transcribed according to the following rules:

Rule	Description
Convert cardinal numbers greater than 10 to numerals	<ul style="list-style-type: none"> "Fifty five" > 55 "a hundred" > 100 "One thousand and thirty one" > 1031 "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" > 123,456,789
Convert cardinal numbers followed by "million" or "billion" to numerals followed by a word when "million" or "billion" is not followed by a number.	<ul style="list-style-type: none"> "one hundred million" > 100 million "one billion" > 1 billion "two point three million" > 2.3 million
Convert ordinal numbers greater than 10 to numbers	<ul style="list-style-type: none"> "Forty third" > 43rd "twenty sixth avenue" > 26th avenue
Convert fractions to their numeric format.	<ul style="list-style-type: none"> "a quarter" > 1/4 "three sixteenths" > 3/16 "a half" > 1/2 "a hundredth" > 1/100
Convert numbers less than 10 to digits if there are more than one in a row.	<ul style="list-style-type: none"> "three four five" > 345 "My phone number is four two five five five five one two one two" > 4255551212
Decimals are indicated by "dot" or "point."	<ul style="list-style-type: none"> "three hundred and three dot five" > 303.5 "three point twenty three" > 3.23 "zero point four" > 0.4 "point three" > 0.3
Convert the word "percent" after a number to a percent sign (%).	<ul style="list-style-type: none"> "twenty three percent" > 23% "twenty three point four five percent" > 23.45%
Convert the words "dollar," "US dollar," "Australian dollar," "AUD," or "USD" after a number to a dollar sign (\$) before the number.	<ul style="list-style-type: none"> "one dollar and fifteen cents" > \$1.15 "twenty three USD" > \$23 "twenty three Australian dollars" > \$23
Convert the words "pounds," or "milligrams" to "lbs" or "mg",	<ul style="list-style-type: none"> "twenty three pounds" > 23 lbs "forty-five milligrams" > 45 mg
Convert the words "rupees," "Indian rupees," or "INR" after a number to rupee sign (₹) before the number.	<ul style="list-style-type: none"> "twenty three rupees" > ₹23 "fifty rupees thirty paise" > ₹50.30
Convert times to numbers.	<ul style="list-style-type: none"> "seven a m eastern standard time" > 7 a.m. eastern standard time "twelve thirty p m" > 12:30 p.m.
Combine years expressed as two digits into four. Only valid for the 20th, 21st, and 22nd centuries.	<ul style="list-style-type: none"> "nineteen sixty two" > 1962 "the year is twenty twelve" > the year is 2012 "twenty nineteen" > 2019 "twenty one thirty" > 2130

Rule	Description
Convert dates to numbers.	<ul style="list-style-type: none">• "May fifth twenty twelve" > May 5th 2012• "May five twenty twelve" > May 5 2012• "five May twenty twelve" > 5 May 2012
Separate spans of numbers by the word "to."	<ul style="list-style-type: none">• "twenty three to thirty seven" > 23 to 37

Getting Started with Amazon Transcribe Medical

To get started using Amazon Transcribe Medical, set up an AWS account and create an AWS Identity and Access Management (IAM) user.

Topics

- [Set Up an AWS Account and Create an Administrator User \(p. 94\)](#)
- [Getting Started \(Console\) \(p. 95\)](#)

Set Up an AWS Account and Create an Administrator User

Before you use Amazon Transcribe Medical for the first time, complete the following tasks:

1. [Sign up for AWS \(p. 94\)](#)
2. [Create an IAM User \(p. 94\)](#)

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all AWS services, including Amazon Transcribe Medical. You are charged only for the services that you use.

With Amazon Transcribe Medical, you pay only for the resources that you use. If you are a new AWS customer, you can get started with Amazon Transcribe Medical for free. For more information, see [AWS Free Usage Tier](#).

If you already have an AWS account, skip to the next section.

To create an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Record your AWS account ID because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon Transcribe Medical, require that you provide credentials when you access them. This allows the service to determine whether you have permissions to access the service's resources.

We strongly recommend that you access AWS using AWS Identity and Access Management (IAM), not the credentials for your AWS account. To use IAM to access AWS, create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user. You can then access AWS using a special URL and the IAM user's credentials.

The Getting Started exercises in this guide assume that you have a user with administrator privileges, `adminuser`.

To create an administrator user and sign in to the console

1. Create an administrator user called `adminuser` in your AWS account. For instructions, see [Creating Your First IAM User and Administrators Group](#) in the *IAM User Guide*.
2. Sign in to the AWS Management Console using a special URL. For more information, see [How Users Sign In to Your Account](#) in the *IAM User Guide*.

For more information about IAM, see the following:

- [AWS Identity and Access Management \(IAM\)](#)
- [Getting Started](#)
- [IAM User Guide](#)

Next Step

To get started with medical transcription using the console, see [Getting Started \(Console\)](#) (p. 95).

Getting Started (Console)

Get started with Amazon Transcribe Medical by using the console to transcribe up to 15 minutes of medical speech into text. You use the microphone on your computer for the audio source.

To start a medical transcription job

1. Open the Amazon Transcribe console at <https://console.aws.amazon.com/transcribe>.
2. From the left menu, choose **Amazon Transcribe Medical** and then choose **Real-time transcription**.
3. Choose **Start streaming** and then speak into your microphone. Amazon Transcribe Medical will transcribe the speech and display the results on the console.

Next Step

To learn more about WebSocket streaming, go to [WebSocket Streaming](#) (p. 95).

Establish a Bi-Directional Connection Using the WebSocket Protocol

You can use the [WebSocket protocol](#) to open a bi-directional connection that is designed to be secure with Amazon Transcribe Medical. You encode the audio stream with event stream encoding, and Amazon Transcribe Medical returns a stream of text in JSON blobs that are also encoded using event stream encoding. For more information, see [Event Stream Encoding](#) (p. 101). You can use the information in this section to create applications using the WebSocket library of your choice.

Topics

- [Adding a Policy for WebSocket Requests to Your IAM Role \(p. 96\)](#)
- [Creating a Pre-Signed URL \(p. 96\)](#)
- [Handling the WebSocket Upgrade Response \(p. 100\)](#)
- [Event Stream Encoding \(p. 101\)](#)
- [Making a WebSocket Streaming Request \(p. 102\)](#)
- [Handling a WebSocket Streaming Response \(p. 102\)](#)
- [Handling WebSocket Streaming Errors \(p. 103\)](#)

Adding a Policy for WebSocket Requests to Your IAM Role

To use the WebSocket protocol to call Amazon Transcribe Medical, you need to attach the following policy to the AWS Identity and Access Management (IAM) role that makes the request.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "transcribemedicalstreaming",
      "Effect": "Allow",
      "Action": "transcribe:StartMedicalStreamTranscription",
      "Resource": "*"
    }
  ]
}
```

Creating a Pre-Signed URL

You need to construct a URL for your WebSocket request that contains the information needed to set up communication between your application and Amazon Transcribe Medical. WebSocket streaming uses the Amazon Signature Version 4 process for signing requests. Signing the request helps to verify the identity of the requester and to protect your audio data in transit. It also protects against potential replay attacks. For more information about Signature Version 4, see [Signing AWS API Requests](#) in the *Amazon Web Services General Reference*.

The URL has the following format. Line breaks have been added for readability.

```
GET https://transcribestreaming.region.amazonaws.com:8443/medical-stream-transcription-
websocket
?language-code=languageCode
  &X-Amz-Algorithm=AWS4-HMAC-SHA256
  &X-Amz-Credential=Signature Version 4 credential scope
  &X-Amz-Date=date
  &X-Amz-Expires=time in seconds until expiration
  &X-Amz-Security-Token=security-token
  &X-Amz-Signature=Signature Version 4 signature
  &X-Amz-SignedHeaders=host
  &media-encoding=mediaEncoding
  &sample-rate=mediaSampleRateHertz
  &session-id=sessionId
  &specialty=PRIMARYCARE
  &type=DICTATION
```

For TYPE, it is best to select CONVERSATION if your use case involves multiple speakers engaged in a discussion. An example would be a conversation between a clinician and a patient. Select DICTATION if your use case involves a single speaker where a person is dictating speech. An example would be if a physician is dictating medical notes for data entry purposes after a patient encounter.

Use the following values for the URL parameters:

- **language-code** – The language code for the input audio. The valid value is en-US.
- **media-encoding** – The encoding used for the input audio. The valid value is pcm.
- **sample-rate** – The sample rate of the input audio in hertz. 16,000 Hz or higher sample rates are accepted.
- **sessionId** – Optional. An identifier for the transcription session. If you don't provide a session ID, Amazon Transcribe Medical generates one for you and returns it in the response.
- **specialty** – The specialty in the medical domain. Must be PRIMARYCARE.
- **type** – The type of audio. Must be DICTATION or CONVERSATION.

The remaining parameters are Signature Version 4 parameters:

- **X-Amz-Algorithm** – The algorithm you're using in the signing process. The only valid value is AWS4-HMAC-SHA256.
- **X-Amz-Credential** – A string separated by slashes ("/") that is formed by concatenating your access key ID and your credential scope components. Credential scope includes the date in YYYYMMDD format, the AWS Region, the service name, and a special termination string (aws4_request).
- **X-Amz-Date** – The date and time that the signature was created. Generate the date and time by following the instructions in [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.
- **X-Amz-Expires** – The length of time in seconds until the credentials expire. The maximum value is 300 seconds (5 minutes).
- **X-Amz-Security-Token** – Optional. A Signature Version 4 token for temporary credentials. If you specify this parameter, include it in the canonical request. For more information, see [Requesting Temporary Security Credentials](#) in the *AWS Identity and Access Management User Guide*.
- **X-Amz-Signature** – The Signature Version 4 signature that you generated for the request.
- **X-Amz-SignedHeaders** – The headers that are signed when creating the signature for the request. The only valid value is host.

To construct the URL for the request and create the Signature Version 4 signature, use the following steps. The examples are in pseudocode.

Task 1: Create a Canonical Request

Create a string that includes information from your request in a standardized format. This ensures that when AWS receives the request, it can calculate the same signature that you calculate in Task 3. For more information, see [Create a Canonical Request for Signature Version 4](#) in the *Amazon Web Services General Reference*.

1. Define variables for the request in your application.

```
# HTTP verb
method = "GET"
# Service name
service = "transcribe"
# AWS Region
region = "AWS Region"
# Amazon Transcribe streaming endpoint
endpoint = "https://transcribestreaming.region.amazonaws.com:8443"
```

```
# Host
host = "transcribestreaming.region.amazonaws.com:8443"
# Date and time of request
amz-date = YYMMDD'T'HMMSS'Z'
# Date without time for credential scope
datestamp = YYYYMMDD
```

2. Create a canonical URI. The canonical URI is the part of the URI between the domain and the query string.

```
canonical_uri = "/medical-stream-transcription-websocket"
```

3. Create the canonical headers and signed headers. Note the trailing "\n" in the canonical headers.

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. Match the algorithm to the hashing algorithm. You must use SHA-256.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. Create the credential scope, which scopes the derived key to the date, AWS Region, and service to which the request is made.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. Create the canonical query string. Query string values must be URL-encoded and sorted by name.

```
canonical_querystring = "X-Amz-Algorithm=" + algorithm
canonical_querystring += "&X-Amz-Credentials=" + access key + "/" + credential_scope
canonical_querystring += "&X-Amz-Date=" + amz_date
canonical_querystring += "&X-Amz-Expires=300"
canonical_querystring += "&X-Amz-Security-Token=" + token
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers
canonical_querystring += "&language-code=en-US&media-encoding=pcm&sample-rate=16000"
canonical_querystring += "&specialty=PRIMARYCARE;"
canonical_querystring += "&type=DICTATION"
```

7. Create a hash of the payload. For a GET request, the payload is an empty string.

```
payload_hash = HashSHA256("").Encode("utf-8").HexDigest()
```

8. Combine all of the elements to create the canonical request.

```
canonical_request = method + '\n'
                  + canonical_uri + '\n'
                  + canonical_querystring + '\n'
                  + canonical_headers + '\n'
                  + signed_headers + '\n'
                  + payload_hash
```

Task 2: Create the String to Sign

The string to sign contains meta information about your request. You use the string to sign in the next step when you calculate the request signature. For more information, see [Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

- Create the string.

```
string_to_sign=algorithm + "\n"  
+ amz_date + "\n"  
+ credential_scope + "\n"  
+ HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

Task 3: Calculate the Signature

You derive a signing key from your AWS secret access key. For a greater degree of protection, the derived key is specific to the date, service, and AWS Region. You use the derived key to sign the request. For more information, see [Calculate the Signature for AWS Signature Version 4](#) in the *Amazon Web Services General Reference*.

The code assumes that you have implemented the `GetSignatureKey` function to derive a signing key. For more information and example functions, see [Examples of How to Derive a Signing Key for Signature Version 4](#) in the *Amazon Web Services General Reference*.

The function `HMAC(key, data)` represents an HMAC-SHA256 function that returns the results in binary format.

- Create the signing key and sign the string to sign.

```
#Create the signing key  
signing_key = GetSignatureKey(secret_key, timestamp, region, service)  
  
# Sign the string_to_sign using the signing key  
signature = HMAC.new(signing_key, (string_to_sign).Encode("utf-8"), Sha256()).HexDigest
```

Task 4: Add Signing Information to the Request and Create the Request URL

After you calculate the signature, add it to the query string. For more information, see [Add the Signature to the HTTP Request](#) in the *Amazon Web Services General Reference*.

1. Add the authentication information to the query string.

```
canonical_querystring += "&X-Amz-Signature=" + signature
```

2. Create the URL for the request.

```
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

You use the request URL with your WebSocket library to make the request to the Amazon Transcribe Medical service.

Including WebSocket Request Headers

The request to Amazon Transcribe Medical must include the following headers. Typically these headers are managed by your WebSocket client library.

```
Host: transcribestreaming.region.amazonaws.com:8443  
Connection: Upgrade  
Upgrade: websocket  
Origin: request source
```

```
Sec-WebSocket-Version: 13  
Sec-WebSocket-Key: random key
```

Use the following values for the headers:

- **Connection** – Always Upgrade.
- **Upgrade** – Always websocket.
- **Origin** – The URI of the WebSocket client.
- **Sec-WebSocket-Version** – The version of the WebSocket protocol to use.
- **Sec-WebSocket-Key** – A base-64 encoded randomly generated string that identifies the request.

Handling the WebSocket Upgrade Response

When Amazon Transcribe Medical receives your WebSocket request, it responds with a WebSocket upgrade response. Typically, your WebSocket library manages this response and sets up a socket for communications with Amazon Transcribe Medical.

The following is the response from Amazon Transcribe Medical. Line breaks have been added to the websocket-location header for readability.

```
HTTP/1.1 101 Web Socket Protocol Handshake  
  
Connection: upgrade  
Upgrade: websocket  
websocket-origin: https://transcribestreaming.region.amazonaws.com:8443  
websocket-location: transcribestreaming.region.amazonaws.com:8443/stream-transcription-  
websocket?  
                                X-Amz-Algorithm=AWS4-HMAC-SHA256  
                                &X-Amz-Credential=AKIDEXAMPLE%2F20190117%2Fregion%2Ftranscribe  
%2Faws4_request  
                                &X-Amz-Date=date and time  
                                &X-Amz-Expires=expiration length  
                                &X-Amz-SignedHeaders=host  
                                &language-code=language code  
                                &media-encoding=media encoding  
                                &sample-rate=media sample rate  
                                &type=dictation or conversation  
                                &specialty=medical specialty- must be Primary Care  
                                &X-Amz-Signature=signature  
x-amzn-RequestId: RequestId  
x-amzn-SessionId: SessionId  
Strict-Transport-Security: max-age=31536000  
sec-websocket-accept: token
```

The response has the following values:

- **Connection** – Always Upgrade.
- **Upgrade** – Always websocket.
- **websocket-origin** – The URI of the WebSocket server that responded to the request.
- **websocket-location** – The contents of the request URI that was sent to the server. For a description of the contents, see [Creating a Pre-Signed URL \(p. 70\)](#).
- **x-amzn-RequestId** – An identifier for the request.
- **x-amzn-SessionId** – An identifier for a transcription session.
- **Strict-Transport-Security** – A header that informs browsers to access the endpoint using only HTTPS.

- **sec-websocket-accept** – The hash of the Sec-WebSocket-Key header sent in the request.

Event Stream Encoding

Event stream encoding provides bidirectional communication using messages between a client and a server. Data frames sent to the Amazon Transcribe Medical streaming service are encoded in this format. The response from Amazon Transcribe Medical also uses this encoding.

Each message consists of two sections: the prelude and the data. The prelude consists of:

1. The total byte length of the message
2. The combined byte length of all of the headers

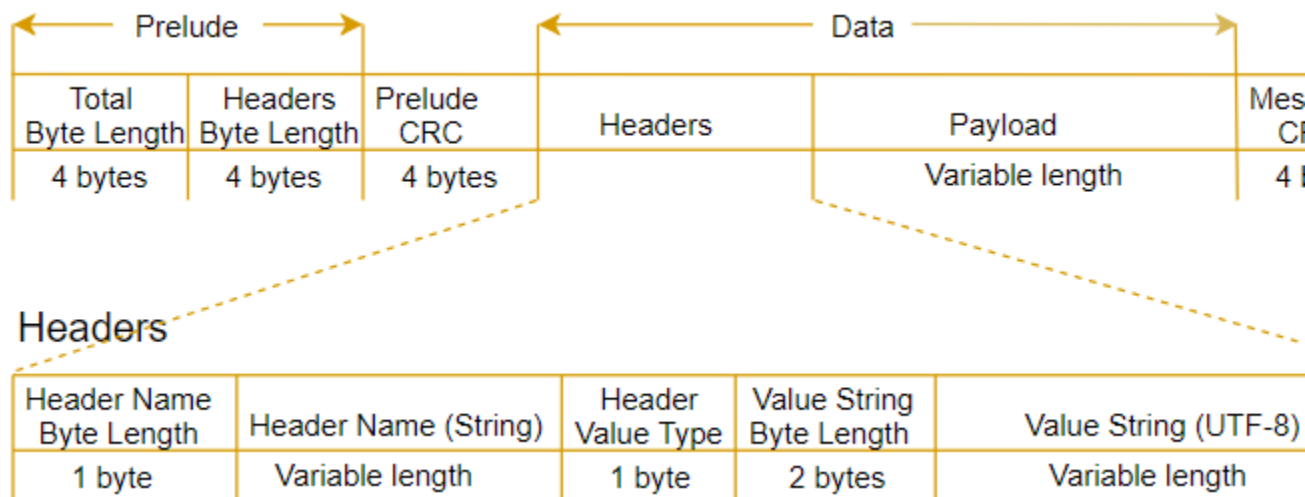
The data section consists of:

1. The headers
2. A payload

Each section ends with a 4-byte big-endian integer CRC checksum. Amazon Transcribe Medical uses CRC32 (often referred to as GZIP CRC32) to calculate both CRCs. For more information about CRC32, see [GZIP file format specification version 4.3](#).

Total message overhead, including the prelude and both checksums, is 16 bytes.

The following diagram shows the components that make up a message and a header. There are multiple headers per message.



Each message contains the following components:

- **Prelude** – Always a fixed size of 8 bytes, two fields of 4 bytes each.
 - *First 4 bytes* – The total byte length. This is the big-endian integer byte length of the entire message, including the 4-byte length field itself.
 - *Second 4 bytes* – The headers byte-length. This is the big-endian integer byte-length of the headers portion of the message, excluding the headers length field itself.
- **Prelude CRC** – The 4-byte CRC checksum for the prelude portion of the message, excluding the CRC itself. The prelude has a separate CRC from the message CRC to ensure that Amazon Transcribe

Medical can detect corrupted byte-length information immediately without causing errors such as buffer overruns.

- **Headers** – Metadata annotating the message, such as the message type, content type, and so on. Messages have multiple headers. Headers are key-value pairs where the key is a UTF-8 string. Headers can appear in any order in the headers portion of the message and any given header can appear only once. For the required header types, see the following sections.
- **Payload** – The audio content to be transcribed.
- **Message CRC** – The 4-byte CRC checksum from the start of the message to the start of the checksum. That is, everything in the message except the CRC itself.

Each header contains the following components. There are multiple headers per frame.

- **Header name byte-length** – The byte-length of the header name.
- **Header name** – The name of the header indicating the header type. For valid values, see the following frame descriptions.
- **Header value type** – An enumeration indicating the header value type.
- **Value string byte length** – The byte-length of the header value string.
- **Header value** – The value of the header string. Valid values for this field depend on the type of header. For valid values, see the following frame descriptions.

Making a WebSocket Streaming Request

After the WebSocket connection is established, the client can start sending a sequence of audio frames. Each frame contains one data frame that is encoded in event stream encoding. For more information, see [Event Stream Encoding \(p. 68\)](#).

Each data frame contains three headers combined with a chunk of raw audio bytes. The following table lists and describes the headers.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

To end the audio data stream, send an empty audio chunk in an event-stream-encoded message.

Handling a WebSocket Streaming Response

The response contains event-stream-encoded raw bytes in the payload. It contains the standard prelude and the following headers.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

When you decode the binary response, you end up with a JSON structure with the results of the transcription. For an example of the JSON response, see [Streaming Transcription](#) (p. 66).

Handling WebSocket Streaming Errors

If an exception occurs while processing your request, Amazon Transcribe Medical responds with a terminal WebSocket frame containing an event-stream-encoded response. The response has the headers described in the following table, and the body of the response contains a descriptive error message. After sending the exception response, Amazon Transcribe Medical sends a close frame.

Header Name Byte Length	Header Name (String)	Header Value Type	Value String Byte Length	Value String (UTF-8)
13	:content-type	7	24	application/octet-stream
15	:exception-type	7	varies	varies, see below
13	:message-type	7	9	exception

The exception-type header contains one of the following values:

- **BadRequestException** – There was a client error when the stream was created, or an error occurred while streaming data. Make sure that your client is ready to accept data and try your request again.
- **InternalFailureException** – Amazon Transcribe Medical had a problem during the handshake with the client. Try your request again.
- **LimitExceededException** – The client exceeded the concurrent stream limit. For more information, see [Amazon Transcribe Service Quotas](#) in the *Amazon Web Services General Reference*. Reduce the number of streams that you are transcribing.
- **UnrecognizedClientException** – The WebSocket upgrade request was signed with an incorrect access key or secret key. Make sure that you are correctly creating the access key and try your request again.

In addition, Transcribe Medical can return any of the common service errors. For a list, see [Common Errors](#).

Security in Amazon Transcribe

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS compliance programs](#). To learn about the compliance programs that apply to Amazon Transcribe, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This section helps you understand how to apply the shared responsibility model when using Amazon Transcribe. The following topics show you how to configure Amazon Transcribe to meet your security and compliance objectives. You also learn how to use other AWS services to monitor and secure your Amazon Transcribe resources.

Topics

- [Data Protection in Amazon Transcribe \(p. 104\)](#)
- [Identity and Access Management for Amazon Transcribe \(p. 106\)](#)
- [Logging and Monitoring in Amazon Transcribe \(p. 117\)](#)
- [Compliance Validation for Amazon Transcribe \(p. 122\)](#)
- [Resilience in Amazon Transcribe \(p. 122\)](#)
- [Infrastructure Security in Amazon Transcribe \(p. 122\)](#)

Data Protection in Amazon Transcribe

Amazon Transcribe conforms to the AWS [shared responsibility model](#), which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all of the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources.

- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon Simple Storage Service (Amazon S3).

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with Amazon Transcribe or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon Transcribe or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

For more information about data protection, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the *AWS Security Blog*.

Topics

- [Encryption at Rest \(p. 105\)](#)
- [Encryption in Transit \(p. 105\)](#)
- [Key Management \(p. 105\)](#)

Encryption at Rest

Amazon Transcribe uses the default Amazon S3 key (SSE-S3) for server-side encryption of transcripts placed in your S3 bucket.

When you use the [StartTranscriptionJob \(p. 163\)](#) operation, you can specify your own AWS Key Management Service key to encrypt the output from a transcription job.

Amazon Transcribe uses an Amazon EBS volume encrypted with the default key.

Encryption in Transit

Amazon Transcribe uses TLS 1.2 with AWS certificates to encrypt data in transit. This includes streaming transcription.

Key Management

Amazon Transcribe works with AWS Key Management Service (KMS) to provide enhanced encryption for your data. Amazon S3 already enables you to encrypt your input audio when creating a transcription job. Integration with KMS enables you to encrypt the output of the [StartTranscriptionJob \(p. 163\)](#) operation.

If you don't specify an encryption key, the output of the transcription job is encrypted with the default Amazon S3 key (SSE-S3).

KMS Encryption with the AWS Console

To encrypt the output of your transcription job, you can choose between using an encryption key for the account that is making the request, or you can use an encryption key from another account.

If you don't specify an encryption key, the output of the transcription job is encrypted with the default Amazon S3 key (SSE-S3).

To enable output result encryption

1. Under **Output data** choose **Encryption**.

☒ Encryption [Info](#)

☒ Use key from current account

☐ Use key from different account

KMS key ID

[Choose key](#)

2. Choose whether the KMS customer-managed key (CMK) is from the account you're currently using or from a different account. If you want to use a key from the current account, choose the key from **KMS key ID**. If you're using a key from a different account, you need to enter the key's ARN. To use a key from a different account, the caller must have `kms:Encrypt` permissions for the KMS key.

KMS Encryption with the API

To use output encryption with the API, you set the `OutputEncryptionKMSKeyId` parameter of the [StartTranscriptionJob](#) (p. 163) operation. You can use an encryption key from the current account, or you can use a key from another account. The account that you are using to create the job must have `kms:Encrypt` permissions for the KMS key.

You can use either of the following to identify a KMS key in the current account:

- KMS Key ID: "1234abcd-12ab-34cd-56ef-1234567890ab"
- KMS Key Alias: "alias/ExampleAlias"

You can use either of the following to identify a KMS key in the current account or another account:

- Amazon Resource Name (ARN) of a KMS Key: "arn:aws:kms:region:account ID:key/1234abcd-12ab-34cd-56ef-1234567890ab"
- ARN of a KMS Key Alias: "arn:aws:kms:region:account ID:alias/ExampleAlias"

Identity and Access Management for Amazon Transcribe

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon Transcribe resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience](#) (p. 107)
- [Authenticating with Identities](#) (p. 107)
- [Managing Access Using Policies](#) (p. 109)
- [How Amazon Transcribe Works with IAM](#) (p. 110)

- [Amazon Transcribe Identity-Based Policy Examples \(p. 112\)](#)
- [Troubleshooting Amazon Transcribe Identity and Access \(p. 115\)](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work you do in Amazon Transcribe.

Service user – If you use the Amazon Transcribe service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Transcribe features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon Transcribe, see [Troubleshooting Amazon Transcribe Identity and Access \(p. 115\)](#).

Service administrator – If you're in charge of Amazon Transcribe resources at your company, you probably have full access to Amazon Transcribe. It's your job to determine which Amazon Transcribe features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Transcribe, see [How Amazon Transcribe Works with IAM \(p. 110\)](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Transcribe. To view example Amazon Transcribe identity-based policies that you can use in IAM, see [Amazon Transcribe Identity-Based Policy Examples \(p. 112\)](#).

Authenticating with Identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [The IAM Console and Sign-in Page](#) in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 Signing Process](#) in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using Multi-Factor Authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS Account Root User

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then

securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM Users and Groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see [Managing Access Keys for IAM Users](#) in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to Create an IAM User \(Instead of a Role\)](#) in the *IAM User Guide*.

IAM Roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM Roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an [identity provider](#). For more information about federated users, see [Federated Users and Roles](#) in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM Roles Differ from Resource-based Policies](#) in the *IAM User Guide*.
- **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see [Creating a Role to Delegate Permissions to an AWS Service](#) in the *IAM User Guide*.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests.

This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles, see [When to Create an IAM Role \(Instead of a User\)](#) in the *IAM User Guide*.

Managing Access Using Policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an entity (root user, IAM user, or IAM role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON Policies](#) in the *IAM User Guide*.

An IAM administrator can use policies to specify who has access to AWS resources, and what actions they can perform on those resources. Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-Based Policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions that identity can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM Policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing Between Managed Policies and Inline Policies](#) in the *IAM User Guide*.

Resource-Based Policies

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket. Service administrators can use these policies to define what actions a specified principal (account member, user, or role) can perform on that resource and under what conditions. Resource-based policies are inline policies. There are no managed resource-based policies.

Access Control Lists (ACLs)

Access control lists (ACLs) are a type of policy that controls which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format. Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see [Access Control List \(ACL\) Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

Other Policy Types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions Boundaries for IAM Entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs Work](#) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session Policies](#) in the *IAM User Guide*.

Multiple Policy Types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy Evaluation Logic](#) in the *IAM User Guide*.

For more information about identity and access management for Amazon Transcribe, continue to the following pages:

- [How Amazon Transcribe Works with IAM](#) (p. 110)
- [Troubleshooting Amazon Transcribe Identity and Access](#) (p. 115)

How Amazon Transcribe Works with IAM

Before you use IAM to manage access to Amazon Transcribe, you should understand which IAM features are available to use with Amazon Transcribe. To get a high-level view of how Amazon Transcribe and other AWS services work with IAM, see [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Topics

- [Amazon Transcribe Identity-Based Policies](#) (p. 110)
- [Amazon Transcribe IAM Roles](#) (p. 111)

Amazon Transcribe Identity-Based Policies

With IAM identity-based policies, you can specify allowed or denied actions and resources and the conditions under which actions are allowed or denied. Amazon Transcribe supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON Policy Elements Reference](#) in the *IAM User Guide*.

Actions

The `Action` element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used in a policy to grant permissions to perform the associated operation.

Policy actions in Amazon Transcribe use the following prefix before the action: `transcribe:`. For example, to grant someone permission to run an Amazon EC2 instance with the Amazon Transcribe `StartTranscriptionJob` API operation, you include the `transcribe:StartTranscriptionJob` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Amazon Transcribe defines actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows.

```
"Action": [
    "transcribe:action1",
    "transcribe:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action.

```
"Action": "transcribe:List*"
```

To see a list of Amazon Transcribe actions, see [Actions Defined by Amazon Transcribe](#) in the *IAM User Guide*.

Resources

Amazon Transcribe doesn't support specifying resource ARNs in a policy.

Condition Keys

Amazon Transcribe doesn't provide service-specific condition keys, but it does support using some global condition keys. For all AWS global condition keys, see [AWS Global Condition Context Keys](#) in the *IAM User Guide*.

Examples

For examples of Amazon Transcribe identity-based policies, see [Amazon Transcribe Identity-Based Policy Examples](#) (p. 112).

Amazon Transcribe Resource-Based Policies

Amazon Transcribe doesn't support resource-based policies.

Amazon Transcribe IAM Roles

An [IAM role](#) is an entity within your AWS account that has specific permissions.

Using Temporary Credentials with Amazon Transcribe

You can use temporary credentials to sign in with federation, to assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS Security Token Service (AWS STS) API operations, such as [AssumeRole](#) or [GetFederationToken](#).

Amazon Transcribe supports using temporary credentials.

Service-Linked Roles

[Service-linked roles](#) allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but can't edit, the permissions for service-linked roles.

Amazon Transcribe doesn't support service-linked roles.

Service Roles

You can allow a service to assume a [service role](#) on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might prevent the service from functioning as expected.

Amazon Transcribe doesn't support service roles.

Amazon Transcribe Identity-Based Policy Examples

By default, IAM users and roles don't have permission to create or modify Amazon Transcribe resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the resources that they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating Policies on the JSON Tab](#) in the *IAM User Guide*.

Topics

- [Policy Best Practices](#) (p. 112)
- [Using the Amazon Transcribe Console](#) (p. 113)
- [AWS Managed \(Predefined\) Policies for Amazon Transcribe](#) (p. 113)
- [Permissions Required for IAM User Roles](#) (p. 114)
- [Permissions Required for Amazon S3 Encryption Keys](#) (p. 114)
- [Allow Users to View Their Own Permissions](#) (p. 115)

Policy Best Practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Amazon Transcribe resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get Started Using AWS Managed Policies** – To start using Amazon Transcribe quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see [Get Started Using Permissions With AWS Managed Policies](#) in the *IAM User Guide*.
- **Grant Least Privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see [Grant Least Privilege](#) in the *IAM User Guide*.
- **Enable MFA for Sensitive Operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see [Using Multi-Factor Authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

- **Use Policy Conditions for Extra Security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see [IAM JSON Policy Elements: Condition](#) in the *IAM User Guide*.

Using the Amazon Transcribe Console

To access the Amazon Transcribe console, you must have a minimum set of permissions for the console. These permissions must allow you to list and view details about the Amazon Transcribe resources in your AWS account. If you create an identity-based policy that applies permissions that are more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can use the Amazon Transcribe console, attach the following AWS managed policy to them.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "transcribe:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

For more information, see [Adding Permissions to a User](#) in the *IAM User Guide*.

AWS Managed (Predefined) Policies for Amazon Transcribe

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. These policies are called AWS managed policies. Managed policies make it easier for you to assign appropriate permissions to users, groups, and roles than if you had to write the policies yourself. For more information, see [AWS Managed Policies](#) in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users, roles, and groups in your account, are specific to Amazon Transcribe:

- **ReadOnly** – Grants read-only access to Amazon Transcribe resources so that you can get and list transcription jobs and custom vocabularies.
- **FullAccess** – Grants full access to create, read, update, delete, and run all Amazon Transcribe resources. It also allows access to Amazon Simple Storage Service (Amazon S3) buckets with `transcribe` in the bucket name.

Note

You can review the managed permission policies by signing in to the IAM console and searching by policy name.

You can also create your own custom IAM policies to allow permissions for Amazon Transcribe API actions. You can attach these custom policies to the IAM users, roles, or groups that require those permissions.

Permissions Required for IAM User Roles

When you create an IAM user to call Amazon Transcribe, the identity must have permission to access the S3 bucket and the AWS Key Management Service (AWS KMS) key used to encrypt the contents of the bucket, if you provided one.

The user must have the following IAM policy for decrypt permissions on the KMS Amazon Resource Name (ARN.)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "KMS key ARN",
      "Effect": "Allow"
    }
  ]
}
```

The user's IAM policy must have Amazon S3 permissions to access the S3 bucket where audio files are stored and transcriptions are saved.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
      ],
      "Resource": "S3 bucket location"
    }
  ]
}
```

Permissions Required for Amazon S3 Encryption Keys

If you are using an AWS KMS key to encrypt an Amazon S3 bucket, include the following in the AWS KMS key policy. This gives Amazon Transcribe access to the contents of the bucket.

```
{
  "Sid": "Allow-Transcribe",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account id:root",
  },
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "KMS key ARN"
}
```

For more information about allowing access to customer master keys, see [Allowing External AWS Accounts to Access a CMK](#) in the *AWS KMS Developer Guide*.

Allow Users to View Their Own Permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": [
        "arn:aws:iam::*:user/${aws:username}"
      ]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Troubleshooting Amazon Transcribe Identity and Access

Use the following information to diagnose and fix common issues that you might encounter when working with Amazon Transcribe and AWS Identity and Access Management (IAM).

Topics

- [I Am Not Authorized to Perform an Action in Amazon Transcribe](#) (p. 116)
- [I Am Not Authorized to Perform iam:PassRole](#) (p. 116)
- [I Want to View My Access Keys](#) (p. 116)
- [I'm an Administrator and Want to Allow Others to Access Amazon Transcribe](#) (p. 117)
- [I Want to Allow People Outside of My AWS Account to Access My Amazon Transcribe Resources](#) (p. 117)

I Am Not Authorized to Perform an Action in Amazon Transcribe

If you are using the AWS Management Console, and you get a message that you're not authorized to perform an action, contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

For example, the following error occurs when an IAM user named `mateojackson` IAM tries to use the console to view details about a transcription job but doesn't have `transcribe:GetTranscriptionJob` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
transcribe:GetTranscriptionJob
```

Mateo must ask his administrator to update his policies to allow him to access the `GetTranscriptionJob` operation using the `transcribe:GetTrascriptionJob` action.

I Am Not Authorized to Perform `iam:PassRole`

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Amazon Transcribe.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon Transcribe. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I Want to View My Access Keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important

Do not provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see [Managing Access Keys](#) in the *IAM User Guide*.

I'm an Administrator and Want to Allow Others to Access Amazon Transcribe

To allow others to access Amazon Transcribe, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Amazon Transcribe.

To get started right away, see [Creating Your First IAM Delegated User and Group](#) in the *IAM User Guide*.

I Want to Allow People Outside of My AWS Account to Access My Amazon Transcribe Resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon Transcribe supports these features, see [How Amazon Transcribe Works with IAM](#) (p. 110).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing Access to an IAM User in Another AWS Account That You Own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing Access to AWS Accounts Owned by Third Parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing Access to Externally Authenticated Users \(Identity Federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM Roles Differ from Resource-based Policies](#) in the *IAM User Guide*.

Logging and Monitoring in Amazon Transcribe

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Transcribe applications. To monitor Amazon Transcribe API calls, you can use AWS CloudTrail. To monitor the status of your jobs, use Amazon CloudWatch Events.

Topics

- [Monitoring Amazon Transcribe API Calls with AWS CloudTrail](#) (p. 117)
- [Using Amazon CloudWatch Events with Amazon Transcribe](#) (p. 120)

Monitoring Amazon Transcribe API Calls with AWS CloudTrail

Amazon Transcribe is integrated with AWS CloudTrail, a service that provides a record of actions taken in Amazon Transcribe by an AWS Identity and Access Management (IAM) user or role, or by an AWS service. CloudTrail captures all API calls for Amazon Transcribe, including calls from the Amazon Transcribe console and from code calls to the Amazon Transcribe APIs, as events. By creating a trail, you can enable continuous delivery of CloudTrail events, including events for Amazon Transcribe, to an Amazon Simple Storage Service (Amazon S3) bucket. If you don't create a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can see

each request that was made to Amazon Transcribe, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon Transcribe Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Transcribe, that activity is recorded in a CloudTrail event along with other AWS service events in the CloudTrail **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon Transcribe, create a trail. A *trail* is a configuration that enables CloudTrail to deliver events as log files to a specified S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

CloudTrail logs all Amazon Transcribe actions, which are documented in the [API Reference \(p. 130\)](#). For example, calls to the `CreateVocabulary`, `GetTranscriptionJob`, and `StartTranscriptionJob` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. This information helps you determine the following:

- Whether the request was made with root or IAM user credentials
- Whether the request was made with temporary security credentials for an IAM role or federated user
- Whether the request was made by another AWS service

For more information, see the [CloudTrail `userIdentity` Element](#).

You can also aggregate Amazon Transcribe log files from multiple AWS Regions and multiple AWS accounts into a single S3 bucket. For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#).

Example: Amazon Transcribe Log File Entries

A *trail* is a configuration that enables delivery of events as log files to a specified S3 bucket. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Calls to the `StartTranscriptionJob` and `GetTranscriptionJob` operations create the following entry.

```
{  
  "Records": [  

```

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser |
WebIdentityUser",
    "principalId": "principal ID",
    "arn": "ARN",
    "accountId": "account ID",
    "accessKeyId": "access key",
    "userName": "user name"
  },
  "eventTime": "timestamp",
  "eventSource": "transcribe.amazonaws.com",
  "eventName": "StartTranscriptionJob",
  "awsRegion": "region",
  "sourceIPAddress": "source IP address",
  "userAgent": "user agent",
  "requestParameters": {
    "mediaFormat": "flac | mp3 | mp4 | wav",
    "languageCode": "en-US | es-US",
    "transcriptionJobName": "unique job name",
    "media": {
      "mediaFileUri": ""
    }
  },
  "responseElements": {
    "transcriptionJob": {
      "transcriptionJobStatus": "IN_PROGRESS",
      "mediaFormat": "flac | mp3 | mp4 | wav",
      "creationTime": "timestamp",
      "transcriptionJobName": "unique job name",
      "languageCode": "en-US | es-US",
      "media": {
        "mediaFileUri": ""
      }
    }
  },
  "requestID": "request ID",
  "eventID": "event ID",
  "eventType": "AwsApiCall",
  "recipientAccountId": "account id"
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser |
WebIdentityUser",
    "principalId": "principal ID",
    "arn": "ARN",
    "accountId": "account ID",
    "accessKeyId": "access key",
    "userName": "user name"
  },
  "eventTime": "timestamp",
  "eventSource": "transcribe.amazonaws.com",
  "eventName": "GetTranscriptionJob",
  "awsRegion": "region",
  "sourceIPAddress": "source IP address",
  "userAgent": "user agent",
  "requestParameters": {
    "transcriptionJobName": "unique job name"
  },
  "responseElements": {
    "transcriptionJob": {
      "settings": {
```

```

    },
    "transcriptionJobStatus": "COMPLETED | FAILED | IN_PROGRESS",
    "mediaFormat": "flac | mp3 | mp4 | wav",
    "creationTime": "timestamp",
    "transcriptionJobName": "unique job name",
    "languageCode": "en-US | es-US",
    "media": {
      "mediaFileUri": ""
    },
    "transcript": {
      "transcriptFileUri": ""
    }
  },
  "requestID": "request ID",
  "eventID": "event ID",
  "eventType": "AwsApiCall",
  "recipientAccountId": "account id"
}
]
}

```

Using Amazon CloudWatch Events with Amazon Transcribe

With Amazon CloudWatch Events, you can respond to state changes in your Amazon Transcribe jobs by triggering events in other AWS services. When a transcription job changes state, CloudWatch Events automatically sends an event to an event stream. You create rules that define the events that you want to monitor in the event stream and the action that CloudWatch Events should take when those events occur. For example, routing the event to another service (or target), which can then take an action. You could, for example, configure a rule to route an event to an AWS Lambda function when a transcription job has completed successfully.

Before using CloudWatch Events, you should understand the following concepts:

- **Event** – An event indicates a change in the state of one of your transcription jobs. For example, when the `TranscriptionJobStatus` of a job changes from `IN_PROGRESS` to `COMPLETED`.
- **Target** – A target is another AWS service that processes an event. For example, AWS Lambda or Amazon Simple Notification Service (Amazon SNS). A target receives events in JSON format.
- **Rule** – A rule matches incoming events that you want CloudWatch Events to watch for and routes them to a target or targets for processing. If a rule routes an event to multiple targets, all of the targets process the event in parallel. A rule can customize the JSON sent to the target.

For more information about creating and managing CloudWatch Events events, see [What is Amazon CloudWatch Events](#) in the *Amazon CloudWatch User Guide*.

Defining CloudWatch Events Rules

To define CloudWatch Events rules, use the [CloudWatch Events console](#). When you define a rule, use Amazon Transcribe as the service name. For an example of how to create a CloudWatch Events rule, see [Creating a CloudWatch Events Rule That Triggers on an Event](#) in the *Amazon CloudWatch User Guide*.

The following is an example of a CloudWatch Events rule for Amazon Transcribe. It's triggered when a transcription job's status changes to `COMPLETED` or `FAILED`.

```

{
  "source": [

```

```
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

The rule contains the following fields:

- **source** – The source of the event. For Amazon Transcribe, this is always `aws.transcribe`.
- **detail-type** – An identifier for the details of the event. For Amazon Transcribe, this is always `Transcribe Job State Change`.
- **detail** – The new job status of the transcription job. In this example, the rule triggers an event when the job status changes to `COMPLETED` or `FAILED`. For a list of status values, see the `TranscriptionJobStatus` field of the [TranscriptionJob](#) (p. 186) data type.

Amazon Transcribe Event

When a job's state changes from `IN_PROGRESS` to either `COMPLETED` or `FAILED`, Amazon Transcribe generates an event. To identify the job that changed state and triggered the event in your target, use the event's `TranscriptionJobName` field. An Amazon Transcribe event contains the following information.

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Transcribe Job State Change",
  "source": "aws.transcribe",
  "account": "account ID",
  "time": "timestamp",
  "region": "region",
  "resources": [],
  "detail": {
    "TranscriptionJobName": "unique job name",
    "TranscriptionJobStatus": "status"
  }
}
```

The event passed to the target contains the following information:

- **version** – The version of the event data. This value is always 0.
- **id** – A unique identifier generated by CloudWatch Events for the event.
- **detail-type** – An identifier for the details of the event. For Amazon Transcribe, this is always `Transcribe Job State Change`.
- **source** – The source of the event. For Amazon Transcribe this is always `Transcribe Job State Change`.
- **account ID** – The AWS account ID of the account that generated the API call.
- **timestamp** – The date and time that the API call was made.
- **region** – The AWS Region where the API call was made.
- **resources** – Resources used by the API call. For Amazon Transcribe, this field is always empty.
- **detail** – Details about the event. It contains the following fields:

- `TranscriptionJobName` – The unique name that you gave the job.
- `TranscriptionJobStatus` – The new status of the transcription job. For a list of status values, see the `TranscriptionJobStatus` field of the [TranscriptionJob](#) (p. 186) data type.

Compliance Validation for Amazon Transcribe

Third-party auditors assess the security and compliance of Amazon Transcribe as part of multiple AWS compliance programs. These include PCI, FedRAMP, HIPAA, and others. You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using Amazon Transcribe is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [Architecting for HIPAA Security and Compliance Whitepaper](#) – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [AWS Config](#) – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

For a list of AWS services in scope of specific compliance programs, see [AWS Services in Scope by Compliance Program](#). For general information, see [AWS Compliance Programs](#).

Resilience in Amazon Transcribe

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

Infrastructure Security in Amazon Transcribe

As a managed service, Amazon Transcribe is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

To access Amazon Transcribe through the network, you use AWS published API calls. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS), such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems, such as Java 7 and later, support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an AWS Identity and Access Management (IAM) principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Guidelines and Quotas

Supported Regions

For a list of AWS Regions where Amazon Transcribe is available, see [Amazon Transcribe Endpoints and Quotas](#) in the *Amazon Web Services General Reference*.

Throttling

You can request a quota increase for the following resources:

Resource	Default
Number of concurrent transcription jobs	100
Transactions per second, <code>StartTranscriptionJob</code> operation	10
Number of concurrent streams for streaming transcription	5
Number of <code>StartStreamTranscription</code> Websocket requests	5
Transactions per second, <code>StartStreamTranscription</code> operation	5
Total number of vocabularies per account	100
Number of pending vocabularies	10
Transactions per second, <code>GetTranscriptionJob</code> operation	20
Transactions per second, <code>ListTranscriptionJobs</code> operation	5
Transactions per second, <code>CreateVocabulary</code> and <code>UpdateVocabulary</code> operations	10

Resource	Default
Transactions per second, DeleteVocabulary operation	5
Transactions per second, GetVocabulary operation	20
Transactions per second, ListVocabularies operation	5

For information about requesting a quota increase, see [AWS Service Quotas](#) in the *Amazon Web Services General Reference*.

Guidelines

For best results:

- Use a lossless format, such as FLAC or WAV, with PCM 16-bit encoding.
- Use a sample rate of 8000 Hz for low-fidelity audio and 16000 Hz for high-fidelity audio.

Amazon Transcribe may store your content to continuously improve the quality of its analysis models. See the [Amazon Transcribe FAQ](#) to learn more. To request that we delete content that may have been stored by Amazon Transcribe, open a case with AWS Support.

Quotas

Amazon Transcribe has the following quotas that are not alterable:

Description	Quotas
Maximum audio file length	14,400 seconds
Maximum audio file size	2 GB
Maximum size of a custom vocabulary	50 KB
Maximum length of a custom vocabulary phrase	256 characters
Maximum size of a vocabulary filter	5 MB
Maximum number of vocabulary filters	100
Number of days that job records are retained	90
Number of channels for channel identification	2

Document History for Amazon Transcribe

- **Latest documentation update:** December 20, 2019

The following table describes important changes in each release of Amazon Transcribe. For notification about updates to this documentation, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
New feature	Amazon Transcribe adds support for creating a vocabulary of words to filter from a transcription. For more information, see Vocabulary Filtering .	December 20, 2019
New feature	Amazon Transcribe adds support for queuing transcription jobs. For more information, see Job Queuing .	December 19, 2019
Region expansion	Amazon Transcribe is now available in the Asia Pacific (Tokyo) (ap-northeast-1) region.	November 21, 2019
New languages	Amazon Transcribe adds support for Gulf Arabic, Hebrew, Japanese, Malay, Swiss German, Telugu, and Turkish.	November 21, 2019
New feature	Amazon Transcribe adds support for alternative transcriptions. For more information, see Alternative Transcriptions .	November 20, 2019
New languages	Amazon Transcribe adds support for Dutch, Farsi, Indonesian, Irish English, Portuguese, Scottish English, Tamil, and Welsh English.	November 12, 2019
New language	Amazon Transcribe now supports streaming transcription for Australian English (en-AU).	October 25, 2019
Region expansion	Amazon Transcribe is now available in the China (Beijing) (cn-north-1) and China (Ningxia) (cn-northwest-1) regions.	October 9, 2019
New feature	Amazon Transcribe enables you to provide your own	September 24, 2019

	AWS Key Management Service key to encrypt your transcription output files. For more information, see the OutputEncryptionKMSKeyId parameter of the StartStreamTranscription operation.	
New languages	Amazon Transcribe adds support for Chinese Mandarin-Mainland and Russian.	August 23, 2019
New feature	Amazon Transcribe adds support for streaming audio transcription using the WebSocket protocol. For more information, see Streaming Transcription .	July 19, 2019
New feature	AWS CloudTrail now records events for the StartStreamTranscription operation.	July 19, 2019
Region expansion	Amazon Transcribe is now available in the US West (N. California) (us-west-1) region.	June 27, 2019
New language	Amazon Transcribe adds support for Modern Standard Arabic.	May 28, 2019
New feature	Amazon Transcribe now transcribes numeric words into numbers for US English. For example, "forty-two" is transcribed as "42". For more information, see Transcribing Numbers .	May 23, 2019
New language	Amazon Transcribe adds support for Hindi and Indian English.	May 15, 2019
New SDK	The AWS SDK for C++ now supports Amazon Transcribe.	May 8, 2019
New language	Amazon Transcribe adds support for Spanish.	April 19, 2019
Region expansion	Amazon Transcribe is now available in the EU (Frankfurt) (eu-central-1) and Asia Pacific (Seoul) (ap-northeast-2) regions.	April 18, 2019
New language	Amazon Transcribe adds support for streaming transcription in British English, French, and Canadian French.	April 5, 2019
New feature	The AWS SDK for Ruby V3 now supports Amazon Transcribe	March 25, 2019

New feature	Amazon Transcribe now enables you to create custom vocabularies, lists of specific words that you want Amazon Transcribe to recognize in your audio input. For more information, see Custom Vocabularies .	March 25, 2019
New languages	Amazon Transcribe adds support for German and Korean.	March 22, 2019
Region expansion	Amazon Transcribe is now available in the South America (Sao Paulo) (sa-east-1) region.	February 7, 2019
New language	Amazon Transcribe now supports streaming transcription for US Spanish (es-US).	February 7, 2019
Region expansion	Amazon Transcribe is now available in the Asia Pacific (Mumbai) (ap-south-1), Asia Pacific (Singapore) (ap-southeast-1), EU (London) (eu-west-2), and EU (Paris) (eu-west3) regions.	January 24, 2019
New languages	Amazon Transcribe adds support for French, Italian, and Brazilian Portuguese.	December 20, 2018
New feature	Amazon Transcribe now supports transcription of audio streams. For more information, see Streaming Transcription .	November 19, 2018
New languages	Amazon Transcribe adds support for Australian English, British English, and Canadian French.	November 15, 2018
Region expansion	Amazon Transcribe is now available in Canada (Central) (ca-central-1) and Asia Pacific (Sydney) (ap-southeast-2).	July 17, 2018
New feature	You can now specify your own location to store the output from a transcription job. For more information, see the TranscriptionJobSummary data type.	July 11, 2018
New feature	Added AWS CloudTrail and Amazon CloudWatch Events integration. For more information, see Monitoring Amazon Transcribe .	June 28, 2018

New feature (p. 126)	Amazon Transcribe adds support for custom vocabularies. For more information, see Create a Custom Vocabulary .	April 4, 2018
New guide (p. 126)	This is the first release of the <i>Amazon Transcribe Developer Guide</i> .	November 29, 2017

API Reference

This section contains the API Reference documentation.

Topics

- [Actions \(p. 130\)](#)
- [Data Types \(p. 178\)](#)

Actions

The following actions are supported by Amazon Transcribe Service:

- [CreateVocabulary \(p. 132\)](#)
- [CreateVocabularyFilter \(p. 136\)](#)
- [DeleteTranscriptionJob \(p. 139\)](#)
- [DeleteVocabulary \(p. 141\)](#)
- [DeleteVocabularyFilter \(p. 143\)](#)
- [GetTranscriptionJob \(p. 145\)](#)
- [GetVocabulary \(p. 148\)](#)
- [GetVocabularyFilter \(p. 151\)](#)
- [ListTranscriptionJobs \(p. 154\)](#)
- [ListVocabularies \(p. 157\)](#)
- [ListVocabularyFilters \(p. 160\)](#)
- [StartTranscriptionJob \(p. 163\)](#)
- [UpdateVocabulary \(p. 168\)](#)
- [UpdateVocabularyFilter \(p. 172\)](#)

The following actions are supported by Amazon Transcribe Streaming Service:

- [StartStreamTranscription \(p. 175\)](#)

Amazon Transcribe Service

The following actions are supported by Amazon Transcribe Service:

- [CreateVocabulary \(p. 132\)](#)
- [CreateVocabularyFilter \(p. 136\)](#)
- [DeleteTranscriptionJob \(p. 139\)](#)
- [DeleteVocabulary \(p. 141\)](#)
- [DeleteVocabularyFilter \(p. 143\)](#)
- [GetTranscriptionJob \(p. 145\)](#)
- [GetVocabulary \(p. 148\)](#)
- [GetVocabularyFilter \(p. 151\)](#)
- [ListTranscriptionJobs \(p. 154\)](#)

- [ListVocabularies](#) (p. 157)
- [ListVocabularyFilters](#) (p. 160)
- [StartTranscriptionJob](#) (p. 163)
- [UpdateVocabulary](#) (p. 168)
- [UpdateVocabularyFilter](#) (p. 172)

CreateVocabulary

Service: Amazon Transcribe Service

Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription of an audio file.

Request Syntax

```
{  
  "LanguageCode": "string",  
  "Phrases": [ "string" ],  
  "VocabularyFileUri": "string",  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

LanguageCode (p. 132)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: Yes

Phrases (p. 132)

An array of strings that contains the vocabulary entries.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: . +

Required: No

VocabularyFileUri (p. 132)

The S3 location of the text file that contains the definition of the custom vocabulary. The URI must be in the same region as the API endpoint that you are calling. The general form is

`https://s3.<aws-region>.amazonaws.com/<bucket-name>/<keyprefix>/<objectkey>`

For example:

`https://s3.us-east-1.amazonaws.com/examplebucket/vocab.txt`

For more information about S3 object names, see [Object Keys](#) in the *Amazon S3 Developer Guide*.

For more information about custom vocabularies, see [Custom Vocabularies](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: (s3://|http(s*)://).+

Required: No

VocabularyName (p. 132)

The name of the vocabulary. The name must be unique within an AWS account. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^[0-9a-zA-Z._-]+

Required: Yes

Response Syntax

```
{
  "FailureReason": "string",
  "LanguageCode": "string",
  "LastModifiedTime": number,
  "VocabularyName": "string",
  "VocabularyState": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FailureReason (p. 133)

If the VocabularyState field is FAILED, this field contains information about why the job failed.

Type: String

LanguageCode (p. 133)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

LastModifiedTime (p. 133)

The date and time that the vocabulary was created.

Type: Timestamp

VocabularyName (p. 133)

The name of the vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

VocabularyState (p. 133)

The processing state of the vocabulary. When the `VocabularyState` field contains `READY` the vocabulary is ready to be used in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `CreateVocabulary` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateVocabularyFilter

Service: Amazon Transcribe Service

Creates a new vocabulary filter that you can use to filter words, such as profane words, from the output of a transcription job.

Request Syntax

```
{
  "LanguageCode": "string",
  "VocabularyFilterFileUri": "string",
  "VocabularyFilterName": "string",
  "Words": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

LanguageCode (p. 136)

The language code of the words in the vocabulary filter. All words in the filter must be in the same language. The vocabulary filter can only be used with transcription jobs in the specified language.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: Yes

VocabularyFilterFileUri (p. 136)

The Amazon S3 location of a text file used as input to create the vocabulary filter. Only use characters from the character set defined for custom vocabularies. For a list of character sets, see [Character Sets for Custom Vocabularies](#).

The specified file must be less than 50 KB of UTF-8 characters.

If you provide the location of a list of words in the VocabularyFilterFileUri parameter, you can't use the Words parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: (s3://|http(s*)://).+

Required: No

VocabularyFilterName (p. 136)

The vocabulary filter name. The name must be unique within the account that contains it.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Words (p. 136)

The words to use in the vocabulary filter. Only use characters from the character set defined for custom vocabularies. For a list of character sets, see [Character Sets for Custom Vocabularies](#).

If you provide a list of words in the `Words` parameter, you can't use the `VocabularyFilterFileUri` parameter.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Syntax

```
{
  "LanguageCode": "string",
  "LastModifiedTime": number,
  "VocabularyFilterName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LanguageCode (p. 137)

The language code of the words in the collection.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

LastModifiedTime (p. 137)

The date and time that the vocabulary filter was modified.

Type: Timestamp

VocabularyFilterName (p. 137)

The name of the vocabulary filter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `CreateVocabulary` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteTranscriptionJob

Service: Amazon Transcribe Service

Deletes a previously submitted transcription job along with any other generated results such as the transcription, models, and so on.

Request Syntax

```
{  
  "TranscriptionJobName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

TranscriptionJobName (p. 139)

The name of the transcription job to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteVocabulary

Service: Amazon Transcribe Service

Deletes a vocabulary from Amazon Transcribe.

Request Syntax

```
{  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

VocabularyName (p. 141)

The name of the vocabulary to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteVocabularyFilter

Service: Amazon Transcribe Service

Removes a vocabulary filter.

Request Syntax

```
{  
  "VocabularyFilterName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

VocabularyFilterName (p. 143)

The name of the vocabulary filter to remove.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetTranscriptionJob

Service: Amazon Transcribe Service

Returns information about a transcription job. To see the status of the job, check the `TranscriptionJobStatus` field. If the status is `COMPLETED`, the job is finished and you can find the results at the location specified in the `TranscriptFileUri` field. If you enable content redaction, the redacted transcript appears in `RedactedTranscriptFileUri`.

Request Syntax

```
{  
  "TranscriptionJobName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

TranscriptionJobName (p. 145)

The name of the job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "TranscriptionJob": {  
    "CompletionTime": number,  
    "ContentRedaction": {  
      "RedactionOutput": "string",  
      "RedactionType": "string"  
    },  
    "CreationTime": number,  
    "FailureReason": "string",  
    "JobExecutionSettings": {  
      "AllowDeferredExecution": boolean,  
      "DataAccessRoleArn": "string"  
    },  
    "LanguageCode": "string",  
    "Media": {  
      "MediaFileUri": "string"  
    },  
    "MediaFormat": "string",  
    "MediaSampleRateHertz": number,  
    "Settings": {  
      "ChannelIdentification": boolean,  
      "MaxAlternatives": number,  
      "MaxSpeakerLabels": number,  
      "ShowAlternatives": boolean,  

```

```
    "ShowSpeakerLabels": boolean,
    "VocabularyFilterMethod": "string",
    "VocabularyFilterName": "string",
    "VocabularyName": "string"
  },
  "StartTime": number,
  "Transcript": {
    "RedactedTranscriptFileUri": "string",
    "TranscriptFileUri": "string"
  },
  "TranscriptionJobName": "string",
  "TranscriptionJobStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TranscriptionJob (p. 145)

An object that contains the results of the transcription job.

Type: [TranscriptionJob \(p. 186\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception Message field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetVocabulary

Service: Amazon Transcribe Service

Gets information about a vocabulary.

Request Syntax

```
{  
  "VocabularyName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

VocabularyName (p. 148)

The name of the vocabulary to return information about. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "DownloadUri": "string",  
  "FailureReason": "string",  
  "LanguageCode": "string",  
  "LastModifiedTime": number,  
  "VocabularyName": "string",  
  "VocabularyState": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DownloadUri (p. 148)

The S3 location where the vocabulary is stored. Use this URI to get the contents of the vocabulary. The URI is available for a limited time.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `(s3://|http(s*)://).+`

FailureReason (p. 148)

If the `VocabularyState` field is `FAILED`, this field contains information about why the job failed.

Type: String

LanguageCode (p. 148)

The language code of the vocabulary entries.

Type: String

Valid Values: `en-US` | `es-US` | `en-AU` | `fr-CA` | `en-GB` | `de-DE` | `pt-BR` | `fr-FR` | `it-IT` | `ko-KR` | `es-ES` | `en-IN` | `hi-IN` | `ar-SA` | `ru-RU` | `zh-CN` | `nl-NL` | `id-ID` | `ta-IN` | `fa-IR` | `en-IE` | `en-AB` | `en-WL` | `pt-PT` | `te-IN` | `tr-TR` | `de-CH` | `he-IL` | `ms-MY` | `ja-JP` | `ar-AE`

LastModifiedTime (p. 148)

The date and time that the vocabulary was last modified.

Type: Timestamp

VocabularyName (p. 148)

The name of the vocabulary to return.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

VocabularyState (p. 148)

The processing state of the vocabulary.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetVocabularyFilter

Service: Amazon Transcribe Service

Returns information about a vocabulary filter.

Request Syntax

```
{  
  "VocabularyFilterName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

VocabularyFilterName (p. 151)

The name of the vocabulary filter for which to return information.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{  
  "DownloadUri": "string",  
  "LanguageCode": "string",  
  "LastModifiedTime": number,  
  "VocabularyFilterName": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DownloadUri (p. 151)

The URI of the list of words in the vocabulary filter. You can use this URI to get the list of words.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `(s3://|http(s*)://).+`

LanguageCode (p. 151)

The language code of the words in the vocabulary filter.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

LastModifiedTime (p. 151)

The date and time that the contents of the vocabulary filter were updated.

Type: Timestamp

VocabularyFilterName (p. 151)

The name of the vocabulary filter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTranscriptionJobs

Service: Amazon Transcribe Service

Lists transcription jobs with the specified status.

Request Syntax

```
{  
  "JobNameContains": "string",  
  "MaxResults": number,  
  "NextToken": "string",  
  "Status": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

JobNameContains (p. 154)

When specified, the jobs returned in the list are limited to jobs whose name contains the specified string.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

MaxResults (p. 154)

The maximum number of jobs to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 154)

If the result of the previous request to `ListTranscriptionJobs` was truncated, include the `NextToken` to fetch the next set of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `.+`

Required: No

Status (p. 154)

When specified, returns only transcription jobs with the specified status. Jobs are ordered by creation date, with the newest jobs returned first. If you don't specify a status, Amazon Transcribe returns all transcription jobs ordered by creation date.

Type: String

Valid Values: QUEUED | IN_PROGRESS | FAILED | COMPLETED

Required: No

Response Syntax

```
{
  "NextToken": "string",
  "Status": "string",
  "TranscriptionJobSummaries": [
    {
      "CompletionTime": number,
      "ContentRedaction": {
        "RedactionOutput": "string",
        "RedactionType": "string"
      },
      "CreationTime": number,
      "FailureReason": "string",
      "LanguageCode": "string",
      "OutputLocationType": "string",
      "StartTime": number,
      "TranscriptionJobName": "string",
      "TranscriptionJobStatus": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 155)

The `ListTranscriptionJobs` operation returns a page of jobs at a time. The maximum size of the page is set by the `MaxResults` parameter. If there are more jobs in the list than the page size, Amazon Transcribe returns the `NextPage` token. Include the token in the next request to the `ListTranscriptionJobs` operation to return in the next page of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: . +

Status (p. 155)

The requested status of the jobs returned.

Type: String

Valid Values: QUEUED | IN_PROGRESS | FAILED | COMPLETED

TranscriptionJobSummaries (p. 155)

A list of objects containing summary information for a transcription job.

Type: Array of [TranscriptionJobSummary \(p. 189\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListVocabularies

Service: Amazon Transcribe Service

Returns a list of vocabularies that match the specified criteria. If no criteria are specified, returns the entire list of vocabularies.

Request Syntax

```
{  
  "MaxResults": number,  
  "NameContains": "string",  
  "NextToken": "string",  
  "StateEquals": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 203).

The request accepts the following data in JSON format.

MaxResults (p. 157)

The maximum number of vocabularies to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NameContains (p. 157)

When specified, the vocabularies returned in the list are limited to vocabularies whose name contains the specified string. The search is case-insensitive, `ListVocabularies` will return both "vocabularyname" and "VocabularyName" in the response list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

NextToken (p. 157)

If the result of the previous request to `ListVocabularies` was truncated, include the `NextToken` to fetch the next set of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `.+`

Required: No

StateEquals (p. 157)

When specified, only returns vocabularies with the `VocabularyState` field equal to the specified state.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Required: No

Response Syntax

```
{
  "NextToken": "string",
  "Status": "string",
  "Vocabularies": [
    {
      "LanguageCode": "string",
      "LastModifiedTime": number,
      "VocabularyName": "string",
      "VocabularyState": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 158)

The `ListVocabularies` operation returns a page of vocabularies at a time. The maximum size of the page is set by the `MaxResults` parameter. If there are more jobs in the list than the page size, Amazon Transcribe returns the `NextPage` token. Include the token in the next request to the `ListVocabularies` operation to return in the next page of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `.+`

Status (p. 158)

The requested vocabulary state.

Type: String

Valid Values: `QUEUED` | `IN_PROGRESS` | `FAILED` | `COMPLETED`

Vocabularies (p. 158)

A list of objects that describe the vocabularies that match the search criteria in the request.

Type: Array of [VocabularyInfo \(p. 192\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListVocabularyFilters

Service: Amazon Transcribe Service

Gets information about vocabulary filters.

Request Syntax

```
{  
  "MaxResults": number,  
  "NameContains": "string",  
  "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 203).

The request accepts the following data in JSON format.

MaxResults (p. 160)

The maximum number of filters to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NameContains (p. 160)

Filters the response so that it only contains vocabulary filters whose name contains the specified string.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

NextToken (p. 160)

If the result of the previous request to `ListVocabularyFilters` was truncated, include the `NextToken` to fetch the next set of collections.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `.+`

Required: No

Response Syntax

```
{
```

```
"NextToken": "string",
"VocabularyFilters": [
  {
    "LanguageCode": "string",
    "LastModifiedTime": number,
    "VocabularyFilterName": "string"
  }
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken (p. 160)

The `ListVocabularyFilters` operation returns a page of collections at a time. The maximum size of the page is set by the `MaxResults` parameter. If there are more jobs in the list than the page size, Amazon Transcribe returns the `NextPage` token. Include the token in the next request to the `ListVocabularyFilters` operation to return in the next page of jobs.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: . +

VocabularyFilters (p. 160)

The list of vocabulary filters. It will contain at most `MaxResults` number of filters. If there are more filters, call the `ListVocabularyFilters` operation again with the `NextToken` parameter in the request set to the value of the `NextToken` field in the response.

Type: Array of [VocabularyFilterInfo \(p. 191\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

StartTranscriptionJob

Service: Amazon Transcribe Service

Starts an asynchronous job to transcribe speech to text.

Request Syntax

```
{
  "ContentRedaction": {
    "RedactionOutput": "string",
    "RedactionType": "string"
  },
  "JobExecutionSettings": {
    "AllowDeferredExecution": boolean,
    "DataAccessRoleArn": "string"
  },
  "LanguageCode": "string",
  "Media": {
    "MediaFileUri": "string"
  },
  "MediaFormat": "string",
  "MediaSampleRateHertz": number,
  "OutputBucketName": "string",
  "OutputEncryptionKMSKeyId": "string",
  "Settings": {
    "ChannelIdentification": boolean,
    "MaxAlternatives": number,
    "MaxSpeakerLabels": number,
    "ShowAlternatives": boolean,
    "ShowSpeakerLabels": boolean,
    "VocabularyFilterMethod": "string",
    "VocabularyFilterName": "string",
    "VocabularyName": "string"
  },
  "TranscriptionJobName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 203).

The request accepts the following data in JSON format.

ContentRedaction (p. 163)

An object that contains the request parameters for content redaction.

Type: [ContentRedaction](#) (p. 180) object

Required: No

JobExecutionSettings (p. 163)

Provides information about how a transcription job is executed. Use this field to indicate that the job can be queued for deferred execution if the concurrency limit is reached and there are no slots available to immediately run the job.

Type: [JobExecutionSettings](#) (p. 181) object

Required: No

LanguageCode (p. 163)

The language code for the language used in the input media file.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: Yes

Media (p. 163)

An object that describes the input media for a transcription job.

Type: [Media \(p. 182\)](#) object

Required: Yes

MediaFormat (p. 163)

The format of the input media file.

Type: String

Valid Values: mp3 | mp4 | wav | flac

Required: No

MediaSampleRateHertz (p. 163)

The sample rate, in Hertz, of the audio track in the input media file.

If you do not specify the media sample rate, Amazon Transcribe determines the sample rate. If you specify the sample rate, it must match the sample rate detected by Amazon Transcribe. In most cases, you should leave the `MediaSampleRateHertz` field blank and let Amazon Transcribe determine the sample rate.

Type: Integer

Valid Range: Minimum value of 8000. Maximum value of 48000.

Required: No

OutputBucketName (p. 163)

The location where the transcription is stored.

If you set the `OutputBucketName`, Amazon Transcribe puts the transcript in the specified S3 bucket. When you call the [GetTranscriptionJob \(p. 145\)](#) operation, the operation returns this location in the `TranscriptFileUri` field. If you enable content redaction, the redacted transcript appears in `RedactedTranscriptFileUri`. If you enable content redaction and choose to output an unredacted transcript, that transcript's location still appears in the `TranscriptFileUri`. The S3 bucket must have permissions that allow Amazon Transcribe to put files in the bucket. For more information, see [Permissions Required for IAM User Roles](#).

You can specify an AWS Key Management Service (KMS) key to encrypt the output of your transcription using the `OutputEncryptionKMSKeyId` parameter. If you don't specify a KMS key, Amazon Transcribe uses the default Amazon S3 key for server-side encryption of transcripts that are placed in your S3 bucket.

If you don't set the `OutputBucketName`, Amazon Transcribe generates a pre-signed URL, a shareable URL that provides secure access to your transcription, and returns it in the `TranscriptFileUri` field. Use this URL to download the transcription.

Type: String

Length Constraints: Maximum length of 64.

Pattern: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

Required: No

OutputEncryptionKMSKeyId (p. 163)

The Amazon Resource Name (ARN) of the AWS Key Management Service (KMS) key used to encrypt the output of the transcription job. The user calling the `StartTranscriptionJob` operation must have permission to use the specified KMS key.

You can use either of the following to identify a KMS key in the current account:

- KMS Key ID: "1234abcd-12ab-34cd-56ef-1234567890ab"
- KMS Key Alias: "alias/ExampleAlias"

You can use either of the following to identify a KMS key in the current account or another account:

- Amazon Resource Name (ARN) of a KMS Key: "arn:aws:kms:region:account ID:key/1234abcd-12ab-34cd-56ef-1234567890ab"
- ARN of a KMS Key Alias: "arn:aws:kms:region:account ID:alias/ExampleAlias"

If you don't specify an encryption key, the output of the transcription job is encrypted with the default Amazon S3 key (SSE-S3).

If you specify a KMS key to encrypt your output, you must also specify an output location in the `OutputBucketName` parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=@.-]{0,2048}$`

Required: No

Settings (p. 163)

A `Settings` object that provides optional settings for a transcription job.

Type: [Settings \(p. 183\)](#) object

Required: No

TranscriptionJobName (p. 163)

The name of the job. Note that you can't use the strings "." or ".." by themselves as the job name. The name must also be unique within an AWS account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: Yes

Response Syntax

```
{
  "TranscriptionJob": {
    "CompletionTime": number,
    "ContentRedaction": {
      "RedactionOutput": "string",
      "RedactionType": "string"
    },
    "CreationTime": number,
    "FailureReason": "string",
    "JobExecutionSettings": {
      "AllowDeferredExecution": boolean,
      "DataAccessRoleArn": "string"
    },
    "LanguageCode": "string",
    "Media": {
      "MediaFileUri": "string"
    },
    "MediaFormat": "string",
    "MediaSampleRateHertz": number,
    "Settings": {
      "ChannelIdentification": boolean,
      "MaxAlternatives": number,
      "MaxSpeakerLabels": number,
      "ShowAlternatives": boolean,
      "ShowSpeakerLabels": boolean,
      "VocabularyFilterMethod": "string",
      "VocabularyFilterName": "string",
      "VocabularyName": "string"
    },
    "StartTime": number,
    "Transcript": {
      "RedactedTranscriptFileUri": "string",
      "TranscriptFileUri": "string"
    },
    "TranscriptionJobName": "string",
    "TranscriptionJobStatus": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TranscriptionJob (p. 166)

An object containing details of the asynchronous transcription job.

Type: [TranscriptionJob \(p. 186\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception Message field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `CreateVocabulary` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateVocabulary

Service: Amazon Transcribe Service

Updates an existing vocabulary with new values. The `UpdateVocabulary` operation overwrites all of the existing information with the values that you provide in the request.

Request Syntax

```
{
  "LanguageCode": "string",
  "Phrases": [ "string" ],
  "VocabularyFileUri": "string",
  "VocabularyName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 203\)](#).

The request accepts the following data in JSON format.

LanguageCode (p. 168)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: Yes

Phrases (p. 168)

An array of strings containing the vocabulary entries.

Type: Array of strings

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: . +

Required: No

VocabularyFileUri (p. 168)

The S3 location of the text file that contains the definition of the custom vocabulary. The URI must be in the same region as the API endpoint that you are calling. The general form is

`https://s3.<aws-region>.amazonaws.com/<bucket-name>/<keyprefix>/<objectkey>`

For example:

`https://s3.us-east-1.amazonaws.com/examplebucket/vocab.txt`

For more information about S3 object names, see [Object Keys](#) in the *Amazon S3 Developer Guide*.

For more information about custom vocabularies, see [Custom Vocabularies](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: (s3://|http(s*)://).+

Required: No

VocabularyName (p. 168)

The name of the vocabulary to update. The name is case-sensitive.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^[0-9a-zA-Z._-]+

Required: Yes

Response Syntax

```
{
  "LanguageCode": "string",
  "LastModifiedTime": number,
  "VocabularyName": "string",
  "VocabularyState": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LanguageCode (p. 169)

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

LastModifiedTime (p. 169)

The date and time that the vocabulary was updated.

Type: Timestamp

VocabularyName (p. 169)

The name of the vocabulary that was updated.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^[0-9a-zA-Z._-]+

VocabularyState (p. 169)

The processing state of the vocabulary. When the `VocabularyState` field contains `READY` the vocabulary is ready to be used in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

ConflictException

When you are using the `CreateVocabulary` operation, the `JobName` field is a duplicate of a previously entered job name. Resend your request with a different name.

When you are using the `UpdateVocabulary` operation, there are two jobs running at the same time. Resend the second request later.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateVocabularyFilter

Service: Amazon Transcribe Service

Updates a vocabulary filter with a new list of filtered words.

Request Syntax

```
{  
  "VocabularyFilterFileUri": "string",  
  "VocabularyFilterName": "string",  
  "Words": [ "string" ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 203).

The request accepts the following data in JSON format.

VocabularyFilterFileUri (p. 172)

The Amazon S3 location of a text file used as input to create the vocabulary filter. Only use characters from the character set defined for custom vocabularies. For a list of character sets, see [Character Sets for Custom Vocabularies](#).

The specified file must be less than 50 KB of UTF-8 characters.

If you provide the location of a list of words in the `VocabularyFilterFileUri` parameter, you can't use the `Words` parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: (s3://|http(s*)://).+

Required: No

VocabularyFilterName (p. 172)

The name of the vocabulary filter to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^[0-9a-zA-Z._-]+

Required: Yes

Words (p. 172)

The words to use in the vocabulary filter. Only use characters from the character set defined for custom vocabularies. For a list of character sets, see [Character Sets for Custom Vocabularies](#).

If you provide a list of words in the `Words` parameter, you can't use the `VocabularyFilterFileUri` parameter.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

Response Syntax

```
{
  "LanguageCode": "string",
  "LastModifiedTime": number,
  "VocabularyFilterName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LanguageCode (p. 173)

The language code of the words in the vocabulary filter.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

LastModifiedTime (p. 173)

The date and time that the vocabulary filter was updated.

Type: Timestamp

VocabularyFilterName (p. 173)

The name of the updated vocabulary filter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

Your request didn't pass one or more validation tests. For example, if the transcription you're trying to delete doesn't exist or if it is in a non-terminal state (for example, it's "in progress"). See the exception `Message` field for more information.

HTTP Status Code: 400

InternalFailureException

There was an internal error. Check the error message and try your request again.

HTTP Status Code: 500

LimitExceededException

Either you have sent too many requests or your input file is too long. Wait before you resend your request, or use a smaller file and resend the request.

HTTP Status Code: 400

NotFoundException

We can't find the requested resource. Check the name and try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Amazon Transcribe Streaming Service

The following actions are supported by Amazon Transcribe Streaming Service:

- [StartStreamTranscription](#) (p. 175)

StartStreamTranscription

Service: Amazon Transcribe Streaming Service

Starts a bidirectional HTTP/2 stream where audio is streamed to Amazon Transcribe and the transcription results are streamed to your application.

The following are encoded as HTTP/2 headers:

- x-amzn-transcribe-language-code
- x-amzn-transcribe-media-encoding
- x-amzn-transcribe-sample-rate
- x-amzn-transcribe-session-id

For more information about using the StartStreamTranscription operation, see [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#).

Request Syntax

The following is a JSON representation of the StartStreamTranscription request. The actual request is binary data encoded using event stream encoding. For more information, see [Event Stream Encoding \(p. 68\)](#).

```
POST /stream-transcription HTTP/2
x-amzn-transcribe-language-code: LanguageCode
x-amzn-transcribe-sample-rate: MediaSampleRateHertz
x-amzn-transcribe-media-encoding: MediaEncoding
x-amzn-transcribe-vocabulary-name: VocabularyName
x-amzn-transcribe-session-id: SessionId
Content-type: application/json

{
  "AudioStream": {
    "AudioEvent": {
      "AudioChunk": blob
    }
  }
}
```

URI Request Parameters

The request requires the following URI parameters.

LanguageCode (p. 175)

Indicates the language used in the input audio stream.

Valid Values: en-AU en-GB en-US es-US fr-CA fr-FR

MediaEncoding (p. 175)

The encoding used for the input audio.

Valid Values: pcm

MediaSampleRateHertz (p. 175)

The sample rate, in Hertz, of the input audio. We suggest that you use 8000 Hz for low quality audio and 16000 Hz for high quality audio.

Valid Range: Minimum value of 8000. Maximum value of 48000.

SessionId (p. 175)

A identifier for the transcription session. Use this parameter when you want to retry a session. If you don't provide a session ID, Amazon Transcribe will generate one for you and return it in the response.

Pattern: `[a-zA-Z0-9]{8}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{12}`

VocabularyName (p. 175)

The name of the vocabulary to use when processing the transcription job.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Request Body

The request accepts the following data as a binary stream. For more information, see [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#).

AudioStream (p. 175)

PCM-encoded stream of audio blobs. The audio stream is encoded as an HTTP/2 data frame. For details of the encoding, see [Event Stream Encoding \(p. 68\)](#).

Type: [AudioStream \(p. 196\)](#) object

Required: Yes

Response Syntax

The response from Amazon Transcribe contains two parts: HTTP/2 headers with metadata about the response and binary data containing the response. The binary data is encoded using event stream encoding. Once the binary data is decoded, the content is a JSON object containing the results of the transcription. For more information, see [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#).

```
HTTP/2 200
x-amzn-request-id: RequestId
x-amzn-transcribe-language-code: LanguageCode
x-amzn-transcribe-sample-rate: MediaSampleRateHertz
x-amzn-transcribe-media-encoding: MediaEncoding
x-amzn-transcribe-vocabulary-name: VocabularyName
x-amzn-transcribe-session-id: SessionId
Content-type: application/json

{
  "TranscriptResultStream": {
    "BadRequestException": {
    },
    "ConflictException": {
    },
    "InternalFailureException": {
    },
    "LimitExceededException": {
    },
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
```

```
{
  "Items": [
    {
      "Content": "string",
      "EndTime": number,
      "StartTime": number,
      "Type": "string"
    }
  ],
  "Transcript": "string"
},
"EndTime": number,
"IsPartial": boolean,
"ResultId": "string",
"StartTime": number
}
]
}
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

LanguageCode (p. 176)

The language code for the input audio stream.

Valid Values: en-GB en-US fr-CA fr-FR es-US

MediaEncoding (p. 176)

The encoding used for the input audio stream.

Valid Values: pcm

MediaSampleRateHertz (p. 176)

The sample rate for the input audio stream. Use 8000 Hz for low quality audio and 16000 Hz for high quality audio.

Valid Range: Minimum value of 8000. Maximum value of 48000.

RequestId (p. 176)

An identifier for the streaming transcription.

SessionId (p. 176)

An identifier for a specific transcription session.

Pattern: [a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}

VocabularyName (p. 176)

The name of the vocabulary used when processing the job.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^[0-9a-zA-Z._-]+

The body of the response from Amazon Transcribe is binary data in event stream encoding. Once the binary data is decoded, the content is a JSON object that contains the following information.

TranscriptResultStream (p. 176)

Represents the stream of transcription events from Amazon Transcribe to your application. For more information, see [Using Amazon Transcribe Streaming With HTTP/2 \(p. 76\)](#).

Type: [TranscriptResultStream \(p. 201\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 202\)](#).

BadRequestException

One or more arguments to the `StartStreamTranscription` operation was invalid. For example, `MediaEncoding` was not set to `pcm` or `LanguageCode` was not set to a valid code. Check the parameters and try your request again.

HTTP Status Code: 400

ConflictException

A new stream started with the same session ID. The current stream has been terminated.

HTTP Status Code: 409

InternalFailureException

A problem occurred while processing the audio. Amazon Transcribe terminated processing. Try your request again.

HTTP Status Code: 500

LimitExceededException

You have exceeded the maximum number of concurrent transcription streams, are starting transcription streams too quickly, or the maximum audio length of 4 hours. Wait until a stream has finished processing, or break your audio stream into smaller chunks and try your request again.

HTTP Status Code: 429

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Data Types

The following data types are supported by Amazon Transcribe Service:

- [ContentRedaction \(p. 180\)](#)
- [JobExecutionSettings \(p. 181\)](#)

- [Media](#) (p. 182)
- [Settings](#) (p. 183)
- [Transcript](#) (p. 185)
- [TranscriptionJob](#) (p. 186)
- [TranscriptionJobSummary](#) (p. 189)
- [VocabularyFilterInfo](#) (p. 191)
- [VocabularyInfo](#) (p. 192)

The following data types are supported by Amazon Transcribe Streaming Service:

- [Alternative](#) (p. 194)
- [AudioEvent](#) (p. 195)
- [AudioStream](#) (p. 196)
- [Item](#) (p. 197)
- [Result](#) (p. 198)
- [Transcript](#) (p. 199)
- [TranscriptEvent](#) (p. 200)
- [TranscriptResultStream](#) (p. 201)

Amazon Transcribe Service

The following data types are supported by Amazon Transcribe Service:

- [ContentRedaction](#) (p. 180)
- [JobExecutionSettings](#) (p. 181)
- [Media](#) (p. 182)
- [Settings](#) (p. 183)
- [Transcript](#) (p. 185)
- [TranscriptionJob](#) (p. 186)
- [TranscriptionJobSummary](#) (p. 189)
- [VocabularyFilterInfo](#) (p. 191)
- [VocabularyInfo](#) (p. 192)

ContentRedaction

Service: Amazon Transcribe Service

Settings for content redaction within a transcription job.

Contents

RedactionOutput

The output transcript file stored in either the default S3 bucket or in a bucket you specify.

When you choose `redacted` Amazon Transcribe outputs only the redacted transcript.

When you choose `redacted_and_unredacted` Amazon Transcribe outputs both the redacted and unredacted transcripts.

Type: String

Valid Values: `redacted` | `redacted_and_unredacted`

Required: Yes

RedactionType

Request parameter that defines the entities to be redacted. The only accepted value is `PII`.

Type: String

Valid Values: `PII`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

JobExecutionSettings

Service: Amazon Transcribe Service

Provides information about when a transcription job should be executed.

Contents

AllowDeferredExecution

Indicates whether a job should be queued by Amazon Transcribe when the concurrent execution limit is exceeded. When the `AllowDeferredExecution` field is true, jobs are queued and will be executed when the number of executing jobs falls below the concurrent execution limit. If the field is false, Amazon Transcribe returns a `LimitExceededException` exception.

If you specify the `AllowDeferredExecution` field, you must specify the `DataAccessRoleArn` field.

Type: Boolean

Required: No

DataAccessRoleArn

The Amazon Resource Name (ARN) of a role that has access to the S3 bucket that contains the input files. Amazon Transcribe will assume this role to read queued media files. If you have specified an output S3 bucket for the transcription results, this role should have access to the output bucket as well.

If you specify the `AllowDeferredExecution` field, you must specify the `DataAccessRoleArn` field.

Type: String

Pattern: `^arn:aws:iam:[0-9]{0,63}:role/[A-Za-z0-9:_/+=@.-]{0,1023}$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Media

Service: Amazon Transcribe Service

Describes the input media file in a transcription request.

Contents

MediaFileUri

The S3 object location of the input media file. The URI must be in the same region as the API endpoint that you are calling. The general form is:

```
s3://<bucket-name>/<keyprefix>/<objectkey>
```

For example:

```
s3://examplebucket/example.mp4
```

```
s3://examplebucket/mediadocs/example.mp4
```

For more information about S3 object names, see [Object Keys](#) in the *Amazon S3 Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: (s3://|http(s*)://).+

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Settings

Service: Amazon Transcribe Service

Provides optional settings for the `StartTranscriptionJob` operation.

Contents

ChannelIdentification

Instructs Amazon Transcribe to process each audio channel separately and then merge the transcription output of each channel into a single transcription.

Amazon Transcribe also produces a transcription of each item detected on an audio channel, including the start time and end time of the item and alternative transcriptions of the item including the confidence that Amazon Transcribe has in the transcription.

You can't set both `ShowSpeakerLabels` and `ChannelIdentification` in the same request. If you set both, your request returns a `BadRequestException`.

Type: Boolean

Required: No

MaxAlternatives

The number of alternative transcriptions that the service should return. If you specify the `MaxAlternatives` field, you must set the `ShowAlternatives` field to true.

Type: Integer

Valid Range: Minimum value of 2. Maximum value of 10.

Required: No

MaxSpeakerLabels

The maximum number of speakers to identify in the input audio. If there are more speakers in the audio than this number, multiple speakers will be identified as a single speaker. If you specify the `MaxSpeakerLabels` field, you must set the `ShowSpeakerLabels` field to true.

Type: Integer

Valid Range: Minimum value of 2. Maximum value of 10.

Required: No

ShowAlternatives

Determines whether the transcription contains alternative transcriptions. If you set the `ShowAlternatives` field to true, you must also set the maximum number of alternatives to return in the `MaxAlternatives` field.

Type: Boolean

Required: No

ShowSpeakerLabels

Determines whether the transcription job uses speaker recognition to identify different speakers in the input audio. Speaker recognition labels individual speakers in the audio file. If you set the `ShowSpeakerLabels` field to true, you must also set the maximum number of speaker labels `MaxSpeakerLabels` field.

You can't set both `ShowSpeakerLabels` and `ChannelIdentification` in the same request. If you set both, your request returns a `BadRequestException`.

Type: Boolean

Required: No

VocabularyFilterMethod

Set to `mask` to remove filtered text from the transcript and replace it with three asterisks ("***") as placeholder text. Set to `remove` to remove filtered text from the transcript without using placeholder text.

Type: String

Valid Values: `remove` | `mask`

Required: No

VocabularyFilterName

The name of the vocabulary filter to use when transcribing the audio. The filter that you specify must have the same language code as the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

VocabularyName

The name of a vocabulary to use when processing the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Transcript

Service: Amazon Transcribe Service

Identifies the location of a transcription.

Contents

RedactedTranscriptFileUri

The S3 object location of the redacted transcript.

Use this URI to access the redacted transcript. If you specified an S3 bucket in the `OutputBucketName` field when you created the job, this is the URI of that bucket. If you chose to store the transcript in Amazon Transcribe, this is a shareable URL that provides secure access to that location.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `(s3://|http(s*)://).+`

Required: No

TranscriptFileUri

The S3 object location of the transcript.

Use this URI to access the transcript. If you specified an S3 bucket in the `OutputBucketName` field when you created the job, this is the URI of that bucket. If you chose to store the transcript in Amazon Transcribe, this is a shareable URL that provides secure access to that location.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `(s3://|http(s*)://).+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TranscriptionJob

Service: Amazon Transcribe Service

Describes an asynchronous transcription job that was created with the `StartTranscriptionJob` operation.

Contents

CompletionTime

A timestamp that shows when the job was completed.

Type: Timestamp

Required: No

ContentRedaction

An object that describes content redaction settings for the transcription job.

Type: [ContentRedaction](#) (p. 180) object

Required: No

CreationTime

A timestamp that shows when the job was created.

Type: Timestamp

Required: No

FailureReason

If the `TranscriptionJobStatus` field is `FAILED`, this field contains information about why the job failed.

The `FailureReason` field can contain one of the following values:

- `Unsupported media format` - The media format specified in the `MediaFormat` field of the request isn't valid. See the description of the `MediaFormat` field for a list of valid values.
- `The media format provided does not match the detected media format` - The media format of the audio file doesn't match the format specified in the `MediaFormat` field in the request. Check the media format of your media file and make sure that the two values match.
- `Invalid sample rate for audio file` - The sample rate specified in the `MediaSampleRateHertz` of the request isn't valid. The sample rate must be between 8000 and 48000 Hertz.
- `The sample rate provided does not match the detected sample rate` - The sample rate in the audio file doesn't match the sample rate specified in the `MediaSampleRateHertz` field in the request. Check the sample rate of your media file and make sure that the two values match.
- `Invalid file size: file size too large` - The size of your audio file is larger than Amazon Transcribe can process. For more information, see [Limits](#) in the *Amazon Transcribe Developer Guide*.
- `Invalid number of channels: number of channels too large` - Your audio contains more channels than Amazon Transcribe is configured to process. To request additional channels, see [Amazon Transcribe Limits](#) in the *Amazon Web Services General Reference*.

Type: String

Required: No

JobExecutionSettings

Provides information about how a transcription job is executed.

Type: [JobExecutionSettings \(p. 181\)](#) object

Required: No

LanguageCode

The language code for the input speech.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: No

Media

An object that describes the input media for the transcription job.

Type: [Media \(p. 182\)](#) object

Required: No

MediaFormat

The format of the input media file.

Type: String

Valid Values: mp3 | mp4 | wav | flac

Required: No

MediaSampleRateHertz

The sample rate, in Hertz, of the audio track in the input media file.

Type: Integer

Valid Range: Minimum value of 8000. Maximum value of 48000.

Required: No

Settings

Optional settings for the transcription job. Use these settings to turn on speaker recognition, to set the maximum number of speakers that should be identified and to specify a custom vocabulary to use when processing the transcription job.

Type: [Settings \(p. 183\)](#) object

Required: No

StartTime

A timestamp that shows with the job was started processing.

Type: Timestamp

Required: No

Transcript

An object that describes the output of the transcription job.

Type: [Transcript \(p. 185\)](#) object

Required: No

TranscriptionJobName

The name of the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

TranscriptionJobStatus

The status of the transcription job.

Type: String

Valid Values: `QUEUED` | `IN_PROGRESS` | `FAILED` | `COMPLETED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TranscriptionJobSummary

Service: Amazon Transcribe Service

Provides a summary of information about a transcription job.

Contents

CompletionTime

A timestamp that shows when the job was completed.

Type: Timestamp

Required: No

ContentRedaction

The content redaction settings of the transcription job.

Type: [ContentRedaction \(p. 180\)](#) object

Required: No

CreationTime

A timestamp that shows when the job was created.

Type: Timestamp

Required: No

FailureReason

If the `TranscriptionJobStatus` field is `FAILED`, a description of the error.

Type: String

Required: No

LanguageCode

The language code for the input speech.

Type: String

Valid Values: `en-US` | `es-US` | `en-AU` | `fr-CA` | `en-GB` | `de-DE` | `pt-BR` | `fr-FR` | `it-IT` | `ko-KR` | `es-ES` | `en-IN` | `hi-IN` | `ar-SA` | `ru-RU` | `zh-CN` | `nl-NL` | `id-ID` | `ta-IN` | `fa-IR` | `en-IE` | `en-AB` | `en-WL` | `pt-PT` | `te-IN` | `tr-TR` | `de-CH` | `he-IL` | `ms-MY` | `ja-JP` | `ar-AE`

Required: No

OutputLocationType

Indicates the location of the output of the transcription job.

If the value is `CUSTOMER_BUCKET` then the location is the S3 bucket specified in the `outputBucketName` field when the transcription job was started with the `StartTranscriptionJob` operation.

If the value is `SERVICE_BUCKET` then the output is stored by Amazon Transcribe and can be retrieved using the URI in the `GetTranscriptionJob` response's `TranscriptFileUri` field.

Type: String

Valid Values: `CUSTOMER_BUCKET` | `SERVICE_BUCKET`

Required: No

StartTime

A timestamp that shows when the job started processing.

Type: Timestamp

Required: No

TranscriptionJobName

The name of the transcription job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

TranscriptionJobStatus

The status of the transcription job. When the status is `COMPLETED`, use the `GetTranscriptionJob` operation to get the results of the transcription.

Type: String

Valid Values: `QUEUED` | `IN_PROGRESS` | `FAILED` | `COMPLETED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

VocabularyFilterInfo

Service: Amazon Transcribe Service

Provides information about a vocabulary filter.

Contents

LanguageCode

The language code of the words in the vocabulary filter.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: No

LastModifiedTime

The date and time that the vocabulary was last updated.

Type: Timestamp

Required: No

VocabularyFilterName

The name of the vocabulary filter. The name must be unique in the account that holds the filter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

VocabularyInfo

Service: Amazon Transcribe Service

Provides information about a custom vocabulary.

Contents

LanguageCode

The language code of the vocabulary entries.

Type: String

Valid Values: en-US | es-US | en-AU | fr-CA | en-GB | de-DE | pt-BR | fr-FR | it-IT | ko-KR | es-ES | en-IN | hi-IN | ar-SA | ru-RU | zh-CN | nl-NL | id-ID | ta-IN | fa-IR | en-IE | en-AB | en-WL | pt-PT | te-IN | tr-TR | de-CH | he-IL | ms-MY | ja-JP | ar-AE

Required: No

LastModifiedTime

The date and time that the vocabulary was last modified.

Type: Timestamp

Required: No

VocabularyName

The name of the vocabulary.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `^[0-9a-zA-Z._-]+`

Required: No

VocabularyState

The processing state of the vocabulary. If the state is `READY` you can use the vocabulary in a `StartTranscriptionJob` request.

Type: String

Valid Values: `PENDING` | `READY` | `FAILED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Amazon Transcribe Streaming Service

The following data types are supported by Amazon Transcribe Streaming Service:

- [Alternative](#) (p. 194)
- [AudioEvent](#) (p. 195)
- [AudioStream](#) (p. 196)
- [Item](#) (p. 197)
- [Result](#) (p. 198)
- [Transcript](#) (p. 199)
- [TranscriptEvent](#) (p. 200)
- [TranscriptResultStream](#) (p. 201)

Alternative

Service: Amazon Transcribe Streaming Service

A list of possible transcriptions for the audio.

Contents

Items

One or more alternative interpretations of the input audio.

Type: Array of [Item \(p. 197\)](#) objects

Required: No

Transcript

The text that was transcribed from the audio.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

AudioEvent

Service: Amazon Transcribe Streaming Service

Provides a wrapper for the audio chunks that you are sending.

Contents

AudioChunk

An audio blob that contains the next part of the audio that you want to transcribe.

Type: Base64-encoded binary data object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

AudioStream

Service: Amazon Transcribe Streaming Service

Represents the audio stream from your application to Amazon Transcribe.

Contents

AudioEvent

A blob of audio from your application. Your audio stream consists of one or more audio events.

Type: [AudioEvent](#) (p. 195) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Item

Service: Amazon Transcribe Streaming Service

A word or phrase transcribed from the input audio.

Contents

Confidence

a numeric score between 0-1.0 representing the confidence of the model in correct identification for each word.

Type: Double

Required: No

Content

The word or punctuation that was recognized in the input audio.

Type: String

Required: No

EndTime

The offset from the beginning of the audio stream to the end of the audio that resulted in the item.

Type: Double

Required: No

StartTime

The offset from the beginning of the audio stream to the beginning of the audio that resulted in the item.

Type: Double

Required: No

Type

The type of the item. `PRONUNCIATION` indicates that the item is a word that was recognized in the input audio. `PUNCTUATION` indicates that the item was interpreted as a pause in the input audio.

Type: String

Valid Values: `pronunciation` | `punctuation`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Result

Service: Amazon Transcribe Streaming Service

The result of transcribing a portion of the input audio stream.

Contents

Alternatives

A list of possible transcriptions for the audio. Each alternative typically contains one item that contains the result of the transcription.

Type: Array of [Alternative \(p. 194\)](#) objects

Required: No

EndTime

The offset in seconds from the beginning of the audio stream to the end of the result.

Type: Double

Required: No

IsPartial

Amazon Transcribe divides the incoming audio stream into segments at natural points in the audio. Transcription results are returned based on these segments.

The `IsPartial` field is `true` to indicate that Amazon Transcribe has additional transcription data to send, `false` to indicate that this is the last transcription result for the segment.

Type: Boolean

Required: No

ResultId

A unique identifier for the result.

Type: String

Required: No

StartTime

The offset in seconds from the beginning of the audio stream to the beginning of the result.

Type: Double

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Transcript

Service: Amazon Transcribe Streaming Service

The transcription in a [TranscriptEvent](#) (p. 200).

Contents

Results

[Result](#) (p. 198) objects that contain the results of transcribing a portion of the input audio stream. The array can be empty.

Type: Array of [Result](#) (p. 198) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TranscriptEvent

Service: Amazon Transcribe Streaming Service

Represents a set of transcription results from the server to the client. It contains one or more segments of the transcription.

Contents

Transcript

The transcription of the audio stream. The transcription is composed of all of the items in the results list.

Type: [Transcript \(p. 199\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TranscriptResultStream

Service: Amazon Transcribe Streaming Service

Represents the transcription result stream from Amazon Transcribe to your application.

Contents

BadRequestException

A client error occurred when the stream was created. Check the parameters of the request and try your request again.

Type: Exception
HTTP Status Code: 400

Required: No

ConflictException

A new stream started with the same session ID. The current stream has been terminated.

Type: Exception
HTTP Status Code: 409

Required: No

InternalFailureException

A problem occurred while processing the audio. Amazon Transcribe terminated processing.

Type: Exception
HTTP Status Code: 500

Required: No

LimitExceededException

Your client has exceeded one of the Amazon Transcribe limits, typically the limit on audio length. Break your audio stream into smaller chunks and try your request again.

Type: Exception
HTTP Status Code: 429

Required: No

TranscriptEvent

A portion of the transcription of the audio stream. Events are sent periodically from Amazon Transcribe to your application. The event can be a partial transcription of a section of the audio stream, or it can be the entire transcription of that portion of the audio stream.

Type: [TranscriptEvent](#) (p. 200) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)

- [AWS SDK for Ruby V3](#)

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: `access_key/YYYYMMDD/region/service/aws4_request`.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.